OXFORD

[R] Companion for Experimental Design and Analysis for Psychology

Lynne J. Williams,
Anjali Krishnan & Hervé Abdi

OXFORD UNIVERSITY
PRESS

Oxford University Press is a department of the University of Oxford.
It furthers the University's objective of excellence in research, scholarship
and education by publishing worldwide in

Oxford New York

Auckland Cape Town Dar es Salaam Hong Kong Karachi
Kuala Lumpur Madrid Melbourne Mexico City Nairobi
New Delhi Shanghai Taipei Toronto

With offices in

Argentina Austria Brazil Chile Czech Republic France Greece
Guatemala Hungary Italy Japan Poland Portugal Singapore
South Korea Switzerland Thailand Turkey Ukraine Vietnam

Published in the United States
by Oxford University Press, Inc., New York

First published 2009

Typeset by
Lynne J. Williams, Toronto, Canada

1 3 5 7 9 10 8 6 4 2

Preface

You have successfully designed your first experiment, run the subjects, and you are faced with a mountain of data. What's next?[1] Does computing an analysis of variance by hand suddenly appear mysteriously attractive? Granted, writing an [R] program and actually getting it to run may appear to be quite an intimidating task for the novice, but fear not! There is no time like the present to overcome your phobias. Welcome to the wonderful world of [R]

The purpose of this book is to introduce you to relatively simple [R] programs. Each of the experimental designs introduced in *Experimental Design and Analysis for Psychology* by Abdi, *et al.* are reprinted herein, followed by their [R] code and output. The first chapter covers correlation, followed by regression, multiple regression, and various analysis of variance designs. We urge you to familiarize yourself with the [R] codes and [R] output, as they in their relative simplicity should alleviate many of your anxieties.

We would like to emphasize that this book is not written as *the* tutorial in the [R] programming language. For that there are several excellent books on the market. Rather, use this manual as your own cook book of basic recipies. As you become more comfortable with [R], you may want to add some additional flavors to enhance your programs beyond what we have suggested herein.

[1] Panic is *not* the answer!

Contents

1 Correlation

1.1 Example: Word Length and Number of Meanings

If you are in the habit of perusing dictionaries as a way of leisurely passing time, you may have come to the conclusion that longer words apparently have fewer meanings attributed to them. Now, finally, through the miracle of statistics, or more precisely, the Pearson Correlation Coefficient, you need no longer ponder this question.

We decided to run a small experiment. The data come from a sample of 20 words taken randomly from the *Oxford English Dictionary*. Table 1.1 on the following page gives the results of this survey.

A quick look at Table 1.1 on the next page does indeed give the impression that longer words tend to have fewer meanings than shorter words (*e.g.*, compare "by" with "tarantula".) Correlation, or more specifically the Pearson coefficient of correlation, is a tool used to evaluate the similarity of two sets of measurements (or dependent variables) obtained on the same observations. In this example, the goal of the coefficient of correlation is to express in a *quantitative* way the relationship between length and number of meanings of words.

For a more detailed description, please refer to Chapter 2 on Correlation in the textbook.

1.1.1 [R] code

```
# Correlation Example: Word Length and Number of Meanings

# We first enter the data under two different variables names
Length=c(3,6,2,6,2,9,6,5,9,4,7,11,5,4,3,9,10,5,4,10)
Meanings=c(8,4,10,1,11,1,4,3,1,6,2,1,9,3,4,1,3,3,3,2)
data=data.frame(Length,Meanings)

Mean=mean(data)
Std_Dev=sd(data)

# We now plot the points and SAVE it as a PDF
# Make sure to add the PATH to the location where the plot is
```

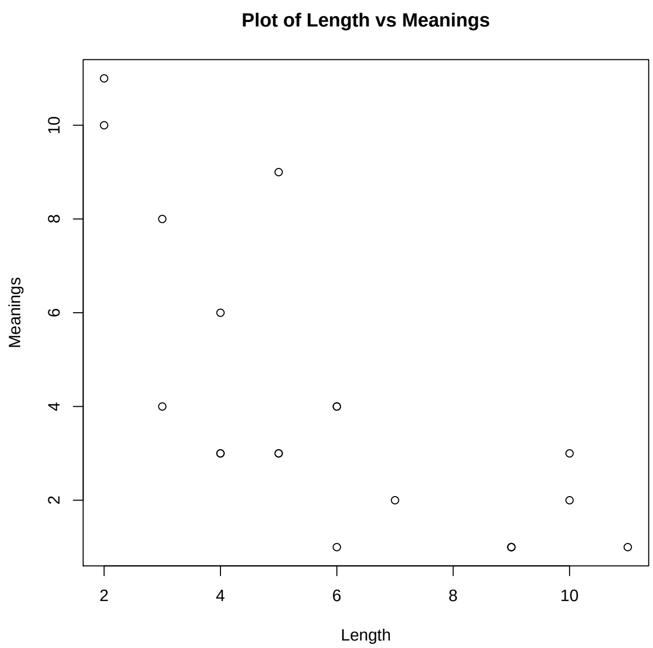
Word	Length	Number of Meanings
bag	3	8
buckle	6	4
on	2	10
insane	6	1
by	2	11
monastery	9	1
relief	6	4
slope	5	3
scoundrel	9	1
loss	4	6
holiday	7	2
pretentious	11	1
solid	5	9
time	4	3
gut	3	4
tarantula	9	1
generality	10	3
arise	5	3
blot	4	3
infectious	10	2

TABLE 1.1 Length (*i.e.,* number of letters) and number of meanings of a random sample of 20 words taken from the Oxford English Dictionary.

```
# to be saved
pdf('/home/anjali/Desktop/R_scripts/01_Correlation/corr_plot.pdf')
plot(Length,Meanings,main="Plot of Length vs Meanings")
dev.off()

# We now perform a correlation and a test on the data which gives
# confidence intervals
cor1=cor.test(Length, Meanings,method = c("pearson"))

# We now perform a regression analysis on the data
reg1=lm(Length~Meanings)

# We now perform an ANOVA on the data
aov1=aov(Length~Meanings)

# We now print the data and all the results
print(data)
print(Mean)
print(Std_Dev)
print(cor1)
summary(reg1)
summary(aov1)
```

1.1.2 [R] output

```
> # Correlation Example: Word Length and Number of Meanings

> # We first enter the data under two different variables names
> Length=c(3,6,2,6,2,9,6,5,9,4,7,11,5,4,3,9,10,5,4,10)
> Meanings=c(8,4,10,1,11,1,4,3,1,6,2,1,9,3,4,1,3,3,3,2)
> data=data.frame(Length,Meanings)

> Mean=mean(data)
> Std_Dev=sd(data)

> # We now plot the points and SAVE it as a PDF
> # Make sure to add the PATH to the location where the plot is
> # to be saved
> pdf('/home/anjali/Desktop/R_scripts/01_Correlation/corr_plot.pdf')
> plot(Length,Meanings,main="Plot of Length vs Meanings")
> dev.off()
```




```
> # We now perform a correlation and a test on the data which gives
> # confidence intervals
> cor1=cor.test(Length, Meanings,method = c("pearson"))

> # We now perform a regression analysis on the data
> reg1=lm(Length~Meanings)

> # We now perform an ANOVA on the data
> aov1=aov(Length~Meanings)

> # We now print the data and all the results
> print(data)
```

```
-------------------
   Length  Meanings
-------------------
1       3         8
2       6         4
3       2        10
4       6         1
5       2        11
6       9         1
7       6         4
8       5         3
9       9         1
10      4         6
11      7         2
12      11        1
13      5         9
14      4         3
15      3         4
16      9         1
17      10        3
18      5         3
19      4         3
20      10        2
-------------------
```

```
> print(Mean)

        ----------------
        Length  Meanings
        ----------------
             6         4
        ----------------

> print(Std_Dev)

        -------------------
          Length   Meanings
        -------------------
        2.809757   3.145590
        -------------------

> print(cor1)

        Pearson's product-moment correlation
        data:  Length and Meanings

        t = -4.5644,  df = 18,  p-value = 0.0002403

        alternative hypothesis: true correlation is not equal to 0

        95 percent confidence interval:  -0.8873588 -0.4289759

        sample estimates:
```

```
----------
       cor
----------
-0.7324543
----------
```

```
> summary(reg1)

       Call:
       lm(formula = Length ~ Meanings)

       Residuals:
       --------------------------------------------------
            Min         1Q     Median        3Q       Max
       --------------------------------------------------
       -3.00000   -1.65426   -0.03723   1.03723   3.34574
       --------------------------------------------------

       Coefficients:
       ---------------------------------------------------
                   Estimate  Std. Error  t value  Pr(>|t|)
       ---------------------------------------------------
       (Intercept)   8.6170      0.7224   11.928  5.56e-10 ***
       Meanings     -0.6543      0.1433   -4.564  0.000240 ***
       ---------------------------------------------------
       ---
       Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

       Residual standard error: 1.965 on 18 degrees of freedom

       Multiple R-squared: 0.5365,Adjusted R-squared: 0.5107
       F-statistic: 20.83 on 1 and 18 DF,  p-value: 0.0002403
```

```
> summary(aov1)

       ----------------------------------------------------
                 d.f.  Sum Sq   Mean Sq  F value     Pr(>F)
       ----------------------------------------------------
       Meanings     1  80.473   80.473    20.834  0.0002403 ***
       Residuals   18  69.527    3.863
       ----------------------------------------------------
       ---
       Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2 Simple Regression Analysis

2.1 Example: Memory Set and Reaction Time

In an experiment originally designed by Sternberg (1969), subjects were asked to memorize a set of random letters (like *lqwh*) called the *memory set*. The number of letters in the set was called the *memory set size*. The subjects were then presented with a *probe* letter (say *q*). Subjects then gave the answer *Yes* if the probe is present in the memory set and *No* if the probe was not present in the memory set (here the answer should be *Yes*). The time it took the subjects to answer was recorded. The goal of this experiment was to find out if subjects were "scanning" material stored in short term memory.

In this replication, each subject was tested one hundred times with a constant memory set size. For half of the trials, the probe is present, whereas for the other half the probe is absent. Four different set sizes are used: 1, 3, 5, and 7 letters. Twenty (fictitious) subjects are tested (five per condition). For each subject we used the mean reaction time for the correct *Yes* answers as the dependent variable. The research hypothesis was that subjects need to serially scan the letters in the memory set and that they need to compare each letter in turn with the probe. If this is the case, then each letter would add a given time to the reaction time. Hence the slope of the line would correspond to the time needed to process one letter of the memory set. The time needed to produce the answer and encode the probe should be constant for all conditions of the memory set size. Hence it should correspond to the intercept. The results of this experiment are given in Table 2.1 on the following page.

2.1.1 [R] code

```
# Regression Example: Memory Set and Reaction time

# We first arrange the data into the Predictors (X) and Regressor (Y)
# In this example the predictors are the sizes of the memory set and
# the regressors are the reaction time of the participants.
```

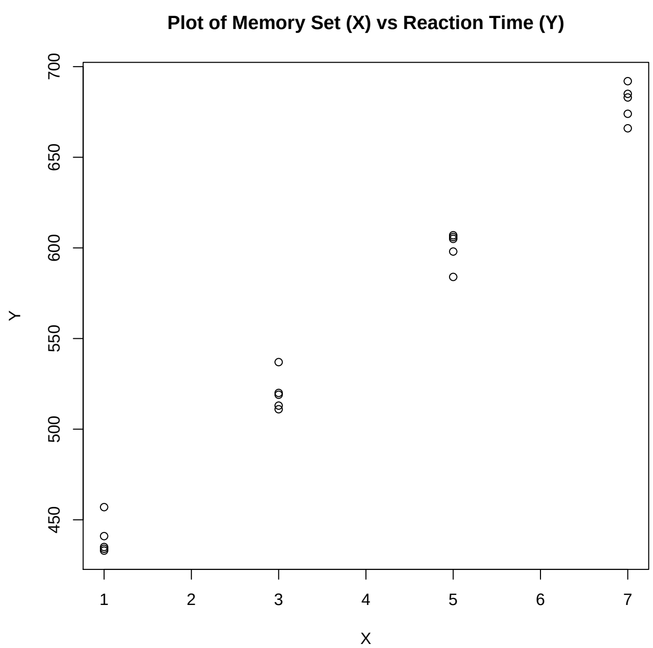
Memory Set Size			
$X = 1$	$X = 3$	$X = 5$	$X = 7$
433	519	598	666
435	511	584	674
434	513	606	683
441	520	605	685
457	537	607	692

TABLE 2.1 Data from a replication of a Sternberg (1969) experiment. Each data point represents the mean reaction time for the *Yes* answers of a given subject. Subjects are tested in only one condition. Twenty (fictitious) subjects participated in this experiment. For example the mean reaction time of subject one who was tested with a memory set of 1 was 433 ($Y_1 = 433$, $X_1 = 1$.)

```
X=c(1,1,1,1,1,3,3,3,3,3,5,5,5,5,5,7,7,7,7,7)
Y=c(433,435,434,441,457,519,511,513,520,537,598,584,606,
   605,607, 666,674,683,685,692)

# We now get a summary of simple statistics for the data

Mean=mean(data)
Std_Dev=sd(data)
r=cor(X,Y)

# We now plot the points and the regression line and SAVE as a pdf
# Make sure to add the PATH to the location where the plot is to be saved

pdf('/home/anjali/Desktop/R_scripts/02_Regression/reg_plot.pdf')
plot(X,Y,main="Plot of Memory Set (X) vs Reaction Time (Y)")
reg.line(reg1)
dev.off()

# We now perform the regression analysis on the data

reg1=lm(Y~X)

# We now perform an ANOVA on the data

aov1=aov(Y~X)

# We now print the data and all the results

print(data)
print(Mean)
print(Std_Dev)
print(r)
summary(reg1)
summary(aov1)
```

2.1.2 [R] output

```
> # Regression Example: Memory Set and Reaction time

> # We first arrange the data into the Predictors (X) and Regressor (Y)
> # In this example the predictors are the sizes of the memory set and
> # the regressors are the reaction time of the participants.

> X=c(1,1,1,1,1,3,3,3,3,3,5,5,5,5,5,7,7,7,7,7)
> Y=c(433,435,434,441,457,519,511,513,520,537,598,584,606,
   605,607, 666,674,683,685,692)

> # We now get a summary of simple statistics for the data

> Mean=mean(data)
> Std_Dev=sd(data)
> r=cor(X,Y)

> # We now plot the points and the regression line and SAVE as a pdf
> # Make sure to add the PATH to the location where the plot is to be saved

> pdf('/home/anjali/Desktop/R_scripts/02_Regression/reg_plot.pdf')
> plot(X,Y,main="Plot of Memory Set (X) vs Reaction Time (Y)")
> reg.line(reg1)
> dev.off()
```




```
> # We now perform the regression analysis on the data

> reg1=lm(Y~X)

> # We now perform an ANOVA on the data
```

```
> aov1=aov(Y~X)

> # We now print the data and all the results

> print(data)

      -------------------
         Length  Meanings
      -------------------
      1       3         8
      2       6         4
      3       2        10
      4       6         1
      5       2        11
      6       9         1
      7       6         4
      8       5         3
      9       9         1
      10      4         6
      11      7         2
      12     11         1
      13      5         9
      14      4         3
      15      3         4
      16      9         1
      17     10         3
      18      5         3
      19      4         3
      20     10         2
      -------------------

> print(Mean)

      ---------------
      Length Meanings
      ---------------
           6        4
      ---------------

> print(Std_Dev)

      ------------------
        Length  Meanings
      ------------------
      2.809757  3.145590
      ------------------

> print(r)

      [1] 0.9950372

> summary(reg1)

      Call:
```

```
lm(formula = Y ~ X)

Residuals:
---------------------------------
   Min     1Q  Median    3Q    Max
---------------------------------
-16.00  -6.25   -0.50  5.25  17.00
---------------------------------

Coefficients:
-----------------------------------------------------
              Estimate  Std. Error  t value  Pr(>|t|)
-----------------------------------------------------
(Intercept)   400.0000      4.3205    92.58    <2e-16 ***
X              40.0000      0.9428    42.43    <2e-16 ***
-----------------------------------------------------
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.428 on 18 degrees of freedom

Multiple R-squared: 0.9901,Adjusted R-squared: 0.9895
F-statistic:  1800 on 1 and 18 DF,  p-value: < 2.2e-16
```

```
> summary(aov1)
```

```
--------------------------------------------------
          d.f.  Sum Sq  Mean Sq  F value     Pr(>F)
--------------------------------------------------
X            1  160000   160000     1800  < 2.2e-16 ***
Residuals   18    1600       89
--------------------------------------------------
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3 Multiple Regression Analysis: Orthogonal Independent Variables

3.1 Example: Retroactive Interference

To illustrate the use of Multiple Regression Analysis, we present a replication of Slamecka's (1960) experiment on retroactive interference. The term retroactive interference refers to the interfering effect of later learning on recall. The general paradigm used to test the effect of retroactive interference is as follows. Subjects in the experimental group are first presented with a list of words to memorize. After the subjects have memorized this list, they are asked to learn a second list of words. When they have learned the second list, they are asked to recall the first list they learned. The number of words recalled by the experimental subjects is then compared with the number of words recalled by control subjects who learned only the first list of words. Results, in general, show that having to learn a second list impairs the recall of the first list (*i.e.,* experimental subjects recall fewer words than control subjects.)

In Slamecka's experiment subjects had to learn complex sentences. The sentences were presented to the subjects two, four, or eight times (this is the first independent variable.) We will refer to this variable as the *number of learning trials* or X. The subjects were then asked to learn a second series of sentences. This second series was again presented two, four, or eight times (this is the second independent variable.) We will refer to this variable as the *number of interpolated lists* or T. After the second learning session, the subjects were asked to recall the first sentences presented. For each subject, the number of words correctly recalled was recorded (this is the dependent variable.) We will refer to the dependent variable as Y.

In this example, a total of 18 subjects (two in each of the nine experimental conditions), were used. How well do the two independent variables "number of learning trials" and "number of interpolated lists" predict the dependent variable "number of words correctly recalled"? The results of this hypothetical replication are presented in Table 3.1.

3.1.1 [R] code

```
# Regression Example: Retroactive Interference

# NOTE: Install and load package "Design" in order to use the "ols"
# function.

# We first arrange the data into the Predictors (X and T) and
# Regressor (Y)

# In this example the predictors are Number of Learning Trials (X)
# and Number   of interpolated lists (T)
X=c(2,2,2,4,4,4,8,8,8,2,2,2,4,4,4,8,8,8)
T=c(2,4,8,2,4,8,2,4,8,2,4,8,2,4,8,2,4,8)

# The Regressors are the number of words correctly recalled (Y).
Y=c(35,21,6,40,34,18,61,58,46,39,31,8,52,42,26,73,66,52)

# Create data frame
data=data.frame(X,T,Y)

Mean=mean(data)
print(Mean)

Std_Dev=sd(data)
print(Std_Dev)


# We now perform an orthogonal multiple regression analysis on the data

multi_reg1=ols(Y~X+T)
print(multi_reg1)

# We now compute the predicted values and the residuals

Y_hat=predict(ols(Y~X+T))
Residual=round(residuals(multi_reg1),2)
print(data.frame(Y,Y_hat,Residual))

# We now compute the sum of squares of the residuals

SS_residual=sum(Residual^2)
print(SS_residual)


# We now compute the correlation matrix between the variables

r_mat=cor(data)
Corr=round(r_mat,4)
print(Corr)
```

Number of learning trials (X)	Number of interpolated lists (T)		
	2	**4**	**8**
2	35	21	6
	39	31	8
4	40	34	18
	52	42	26
8	61	58	46
	73	66	52

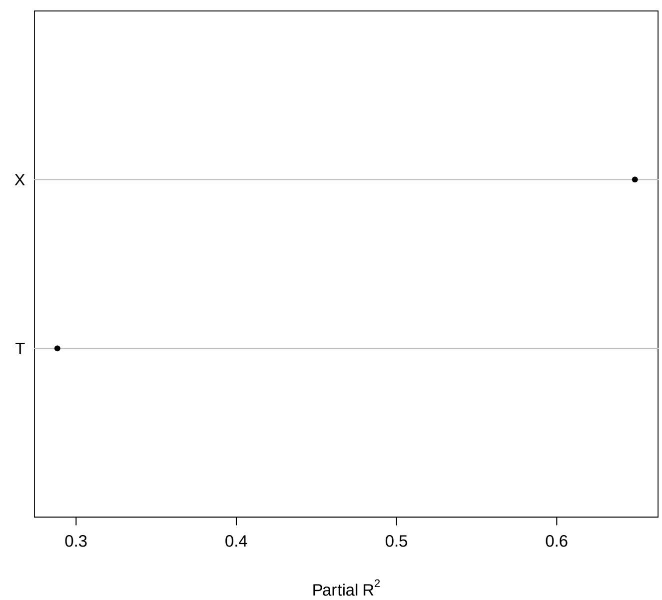
TABLE 3.1 Results of an hypothetical replication of Slamecka (1960)'s retroactive interference experiment.

```
# We now compute the semi-partial coefficients and create a plot
# Make sure to add the PATH to the location where the plot is to be saved

pdf('/Desktop/R_scripts/03_Ortho_Multi_Reg/semi_part_corr.pdf')
semi_part=plot(anova(multi_reg1),what='partial R2')
dev.off()
print(semi_part)

# We now perform an ANOVA on the data that shows the semi-partial
# sums of squares

aov1=anova(ols(Y~X+T))
print(aov1)
```

3.1.2 [R] output

```
> # Regression Example: Retroactive Interference

> # NOTE: Install and load package "Design" in order to use the "ols"
> # function.

> # We first arrange the data into the Predictors (X and T) and
> # Regressor (Y)

> # In this example the predictors are Number of Learning Trials (X)
> # and Number   of interpolated lists (T)
> X=c(2,2,2,4,4,4,8,8,8,2,2,2,4,4,4,8,8,8)
> T=c(2,4,8,2,4,8,2,4,8,2,4,8,2,4,8,2,4,8)

> # The Regressors are the number of words correctly recalled (Y).
> Y=c(35,21,6,40,34,18,61,58,46,39,31,8,52,42,26,73,66,52)

> # Create data frame
> data=data.frame(X,T,Y)

> Mean=mean(data)
```

```
> Std_Dev=sd(data)

> # We now perform an orthogonal multiple regression analysis on the data
> multi_reg1=ols(Y~X+T)

> # We now compute the predicted values and the residuals
> Y_hat=predict(ols(Y~X+T))
> Residual=round(residuals(multi_reg1),2)

> We now compute the sum of squares of the residuals
> SS_residual=sum(Residual^2)

> # We now compute the correlation matrix between the variables
> r_mat=cor(data)
> Corr=round(r_mat,4)

> # We now compute the semi-partial coefficients and create a plot
> # Make sure to add the PATH to the location where the plot is to be saved
> pdf('/Desktop/R_scripts/03_Ortho_Multi_Reg/semi_part_corr.pdf')
> semi_part=plot(anova(multi_reg1),what='partial R2')
> dev.off()
```




```
> # We now perform an ANOVA on the data that shows the semi-partial
> # sums of squares
> aov1=anova(ols(Y~X+T))

> # We now print the data and all the results
> print(data)

            --------------
                X   T    Y
```

```
        --------------
        1   2   2   35
        2   2   4   21
        3   2   8    6
        4   4   2   40
        5   4   4   34
        6   4   8   18
        7   8   2   61
        8   8   4   58
        9   8   8   46
        10  2   2   39
        11  2   4   31
        12  2   8    8
        13  4   2   52
        14  4   4   42
        15  4   8   26
        16  8   2   73
        17  8   4   66
        18  8   8   52
        --------------

> print(Mean)

        --------------------------------
               X           T           Y
        --------------------------------
        4.666667    4.666667   39.333333
        --------------------------------

> print(Std_Dev)

        --------------------------------
               X           T           Y
        --------------------------------
        2.566756    2.566756   19.118823
        --------------------------------

> print(multi_reg1)

        Linear Regression Model

        ols(formula = Y ~ X + T)


        -----------------------------------------------
         n  Model L.R.       d.f.         R2      Sigma
        -----------------------------------------------
        18       49.83          2     0.9372      5.099
        -----------------------------------------------

        Residuals:
        ---------------------------------
         Min      1Q  Median      3Q     Max
        ---------------------------------
```

```
-9.0   -4.0     0.5    4.0    6.0
---------------------------------

Coefficients:
----------------------------------------------
          Value  Std. Error       t   Pr(>|t|)
----------------------------------------------
Intercept    30      3.3993   8.825  2.519e-07
X             6      0.4818  12.453  2.601e-09
T            -4      0.4818  -8.302  5.440e-07
----------------------------------------------

Residual standard error: 5.099 on 15 degrees of freedom
Adjusted R-Squared: 0.9289
```

```
> print(data.frame(Y,Y_hat,Residual))
```

```
-----------------------
     Y  Y_hat  Residual
-----------------------
1   35     34         1
2   21     26        -5
3    6     10        -4
4   40     46        -6
5   34     38        -4
6   18     22        -4
7   61     70        -9
8   58     62        -4
9   46     46         0
10  39     34         5
11  31     26         5
12   8     10        -2
13  52     46         6
14  42     38         4
15  26     22         4
16  73     70         3
17  66     62         4
18  52     46         6
-----------------------
```

```
> print(SS_residual)
```

```
[1] 390
```

```
> print(Corr)
```

```
---------------------------
         X       T        Y
---------------------------
X   1.0000   0.000   0.8055
T   0.0000   1.000  -0.5370
Y   0.8055  -0.537   1.0000
---------------------------
```

```
> print(semi_part)

          --------------------
                  X          T
          --------------------
          0.6488574  0.2883811
          --------------------

> print(aov1)


                         Analysis of Variance
                              Response: Y
          -----------------------------------------------------
           Factor     d.f.  Partial SS     MS        F        P
          -----------------------------------------------------
           X            1         4032   4032   155.08   <.0001
           T            1         1792   1792    68.92   <.0001
           REGRESSION   2         5824   2912   112.00   <.0001
            ERROR       15          390     26
          -----------------------------------------------------
```

4 Multiple Regression Analysis: Non-orthogonal Independent Variables

4.1 Example: Age, Speech Rate and Memory Span

To illustrate an experiment with two quantitative independent variables, we replicated an experiment originally designed by Hulme, Thomson, Muir, and Lawrence (1984, as reported by Baddeley, 1990, p.78 *ff.*). Children aged 4, 7, or 10 years (hence *"age"* is the first independent variable in this experiment, denoted X), were tested in 10 series of immediate serial recall of 15 items. The dependent variable is the total number of words correctly recalled (*i.e.,* in the correct order). In addition to age, the speech rate of each child was obtained by asking the child to read aloud a list of words. Dividing the number of words read by the time needed to read them gave the *speech rate* (expressed in words per second) of the child. Speech rate is the second independent variable in this experiment (we will denote it T).

The research hypothesis states that the age and the speech rate of the children are determinants of their memory performance. Because the independent variable speech rate cannot be *manipulated,* the two independent variables are not orthogonal. In other words, one can expect speech rate to be partly correlated with age (on average, older children tend to speak faster than younger children.) Speech rate should be the major determinant of performance and the effect of age reflects more the confounded effect of speech rate rather than age, per se.

The data obtained from a sample of 6 subjects are given in the Table 4.1 on the next page.

4.1.1 [R] code

```
# Regression Example: Age, Speech Rate and Memory Span

# Install and load package "Design" in order to use the "ols"
# function.
```

The Independent Variables		The Dependent Variable
X Age (in years)	T Speech Rate (words per second)	Y Memory Span (number of words recalled)
4	1	14
4	2	23
7	2	30
7	4	50
10	3	39
10	6	67

TABLE 4.1 Data from a (fictitious) replication of an experiment of Hulme *et al.* (1984). The dependent variable is the total number of words recalled in 10 series of immediate recall of items, it is a measure of the *memory span*. The first independent variable is the *age* of the child, the second independent variable is the *speech rate* of the child.

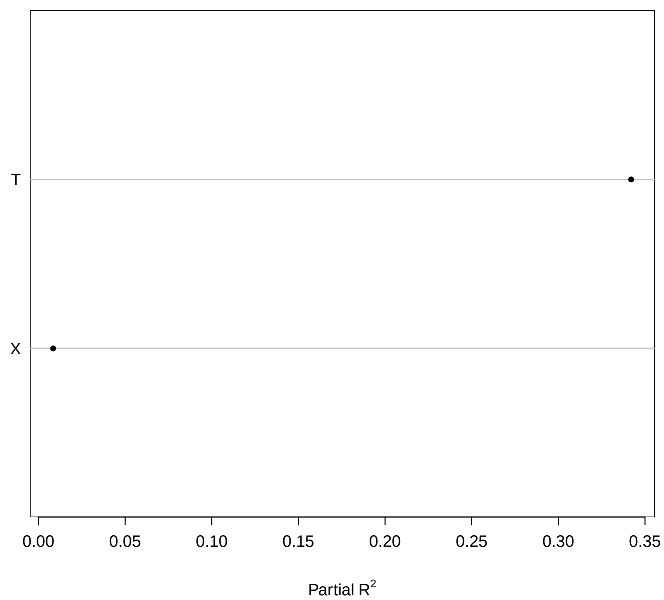
```
# We first arrange the data into the Predictors (X and T) and
# Regressor (Y)

# In this example the predictors are Age (X) and Speech Rate (T)
X=c(4,4,7,7,10,10)
T=c(1,2,2,4,3,6)

# The Regressors are the number of words correctly recalled (Y).
Y=c(14,23,30,50,39,67)

data=data.frame(X,T,Y)

Mean=mean(data)
Std_Dev=sd(data)

# Now we perform an orthogonal multiple regression analysis on the data
multi_reg1=ols(Y~X+T)

# Now we compute the predicted values and the residuals
Y_hat=round(predict(ols(Y~X+T)),2)

Residual=round(residuals(multi_reg1),2)
SS_residual=sum(Residual^2)

# Now we compute the correlation matrix between the variables
r_mat=cor(data)
Corr=round(r_mat,4)

# Now we compute the semi-partial coefficients
# Make sure to add the PATH to the location where the plot is to be saved
pdf('/home/anjali/Desktop/R_scripts/04_Non_Ortho_Multi_Reg/
semi_part_corr.pdf')
```

```
semi_part=plot(anova(multi_reg1),what='partial R2')
dev.off()

# Now we perfom an ANOVA on the data that shows the semi-partial
# sums of squares
aov1=anova(ols(Y~X+T))

# We now print the data and all the results
print(data)
print(Mean)
print(Std_Dev)
print(multi_reg1)
print(data.frame(Y,Y_hat,Residual))
print(SS_residual)
print(Corr)
print(semi_part)
print(aov1)
```

4.1.2 [R] output

```
> # Regression Example: Age, Speech Rate and Memory Span

> # Install and load package "Design" in order to use the "ols"
> # function.

> # We first arrange the data into the Predictors (X and T) and
> # Regressor (Y)

> # In this example the predictors are Age (X) and Speech Rate (T)
> X=c(4,4,7,7,10,10)
> T=c(1,2,2,4,3,6)

> # The Regressors are the number of words correctly recalled (Y).
> Y=c(14,23,30,50,39,67)

> data=data.frame(X,T,Y)

> Mean=mean(data)
> Std_Dev=sd(data)

> # Now we perform an orthogonal multiple regression analysis on the data
> multi_reg1=ols(Y~X+T)

> # Now we compute the predicted values and the residuals
> Y_hat=round(predict(ols(Y~X+T)),2)

> Residual=round(residuals(multi_reg1),2)
> SS_residual=sum(Residual^2)

> # Now we compute the correlation matrix between the variables
> r_mat=cor(data)
> Corr=round(r_mat,4)
```

```
> # Now we compute the semi-partial coefficients
> # Make sure to add the PATH to the location where the plot is to be saved
> pdf('/home/anjali/Desktop/R_scripts/04_Non_Ortho_Multi_Reg/
> semi_part_corr.pdf')
> semi_part=plot(anova(multi_reg1),what='partial R2')
> dev.off()
```




```
> # Now we perfom an ANOVA on the data that shows the semi-partial
> # sums of squares
> aov1=anova(ols(Y~X+T))

> # We now print the data and all the results
> print(data)
```

	X	T	Y
1	4	1	14
2	4	2	23
3	7	2	30
4	7	4	50
5	10	3	39
6	10	6	67

```
> print(Mean)
```

X	T	Y

```
    7.00000  3.00000  37.16667
    --------------------------

> print(Std_Dev)

    ----------------------------
           X         T         Y
    ----------------------------
    2.683282  1.788854  19.218914
    ----------------------------

> print(multi_reg1)

    Linear Regression Model

    ols(formula = Y ~ X + T)

    ------------------------------------
    n  Model L.R.  d.f.       R2    Sigma
    ------------------------------------
    6       25.85     2   0.9866    2.877
    ------------------------------------

    Residuals:
    --------------------------------------------
         1       2      3      4       5       6
    --------------------------------------------
    -1.167  -1.667  2.333  3.333  -1.167  -1.667
    --------------------------------------------

    Coefficients:
    ---------------------------------------------
              Value  Std. Error       t  Pr(>|t|)
    ---------------------------------------------
    Intercept 1.667       3.598  0.4633  0.674704
    X         1.000       0.725  1.3794  0.261618
    T         9.500       1.087  8.7361  0.003158
    ---------------------------------------------

    Residual standard error: 2.877 on 3 degrees of freedom
    Adjusted R-Squared: 0.9776


> print(data.frame(Y,Y_hat,Residual))

    ----------------------
        Y  Y_hat  Residual
    ----------------------
    1  14  15.17     -1.17
    2  23  24.67     -1.67
    3  30  27.67      2.33
    4  50  46.67      3.33
    5  39  40.17     -1.17
    6  67  68.67     -1.67
```

```
      ----------------------

> print(SS_residual)

      [1] 24.8334

> print(Corr)

      ------------------------
              X      T       Y
      ------------------------
      X  1.0000  0.750  0.8028
      T  0.7500  1.000  0.9890
      Y  0.8028  0.989  1.0000
      ------------------------

> print(semi_part)

      ------------------------
                T            X
      ------------------------
      0.342072015  0.008528111
      ------------------------

> print(aov1)
                          Analysis of Variance
                              Response: Y
      --------------------------------------------------------
      Factor      d.f.  Partial SS          MS       F       P
      --------------------------------------------------------
       X             1    15.75000   15.750000    1.90  0.2616
       T             1   631.75000  631.750000   76.32  0.0032
       REGRESSION    2  1822.00000  911.000000  110.05  0.0016
       ERROR         3    24.83333    8.277778
      --------------------------------------------------------
```

5

ANOVA One Factor Between-Subjects, $\mathcal{S}(\mathcal{A})$

5.1 Example: Imagery and Memory

Our research hypothesis is that material processed with imagery will be more resistant to forgetting than material processed without imagery. In our experiment, we ask subjects to learn pairs of words (e.g., "beauty-carrots"). Then, after some delay, the subjects are asked to give the second word of the pair (e.g., "carrot") when prompted with the first word of the pair (e.g., "beauty"). Two groups took part in the experiment: the *experimental* group (in which the subjects learn the word pairs using imagery), and the *control* group (in which the subjects learn without using imagery). The dependent variable is the number of word pairs correctly recalled by each subject. The performance of the subjects is measured by testing their memory for 20 word pairs, 24 hours after learning.

The results of the experiment are listed in the following table:

Experimental group	Control group
1	8
2	8
5	9
6	11
6	14

5.1.1 [R] code

```
# ANOVA One-factor between subjects, S(A)
# Imagery and Memory

# We have 1 Factor, A, with 2 levels: Experimental Group and Control
# Group.

# We have 5 subjects per group. Therefore 5 x 2 = 10 subjects total.
```

```
# We collect the data for each level of Factor A
Expt=c(1,2,5,6,6)
Control=c(8,8,9,11,14)

# We now combine the observations into one long column (score).
score=c(Expt,Control)

# We generate a second column (group), that identifies the group for
# each score.

levels=factor(c(rep("Expt",5),rep("Control",5)))

# We now form a data frame with the dependent variable and the factors.
data=data.frame(score=score,group=levels)

# We now generate the ANOVA table based on the linear model
aov1=aov(score~levels)
print(aov1)

# We now print the data and all the results
print(data)
print(model.tables(aov(score~levels),type = "means"),digits=3)
summary(aov1)
```

5.1.2 [R] output

```
> # ANOVA One-factor between subjects, S(A)
> # Imagery and Memory

> # We have 1 Factor, A, with 2 levels: Experimental Group and Control
> # Group.

> # We have 5 subjects per group. Therefore 5 x 2 = 10 subjects total.

> # We collect the data for each level of Factor A
> Expt=c(1,2,5,6,6)
> Control=c(8,8,9,11,14)

> # We now combine the observations into one long column (score).
> score=c(Expt,Control)

> # We generate a second column (group), that identifies the group for
> # each score.

> levels=factor(c(rep("Expt",5),rep("Control",5)))

> # We now form a data frame with the dependent variable and the factors.
> data=data.frame(score=score,group=levels)

> # We now generate the ANOVA table based on the linear model
> aov1=aov(score~levels)
> print(aov1)
```

```
       Call:
       aov(formula = score ~ levels)
              Terms:
       ---------------------------------
                       Levels  Residuals
       ---------------------------------
       Sum of Squares      90         48
       Deg. of Freedom      1          8
       ---------------------------------
              Residual standard error: 2.449490
       Estimated effects may be unbalanced

> # We now print the data and all the results
> print(data)

       -----------------
          Score    Group
       -----------------
       1      1     Expt
       2      2     Expt
       3      5     Expt
       4      6     Expt
       5      6     Expt
       6      8  Control
       7      8  Control
       8      9  Control
       9     11  Control
       10    14  Control
       -----------------

> print(model.tables(aov(score~levels),type = "means"),digits=3)

       Tables of means

       Grand mean
       ----------
                7
       ----------

       Levels
       ---------------
       Control    Expt
       ---------------
            10       4
       ---------------

> summary(aov1)

       ----------------------------------------------------
                   d.f.  Sum Sq  Mean Sq  F value    Pr(>F)
       ----------------------------------------------------
       levels         1      90       90       15  0.004721 **
       Residuals      8      48        6
       ----------------------------------------------------
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.1.3 ANOVA table

The results from our experiment can be condensed in an *analysis of variance table*.

Source	*df*	*SS*	*MS*	*F*
Between	1	90.00	90.00	15.00
Within $\mathcal{S}$	8	48.00	6.00	
Total	9	138.00		

5.2 Example: Romeo and Juliet

In an experiment on the effect of context on memory, Bransford and Johnson (1972) read the following passage to their subjects:

> "If the balloons popped, the sound would not be able to carry since everything would be too far away from the correct floor. A closed window would also prevent the sound from carrying since most buildings tend to be well insulated. Since the whole operation depends on a steady flow of electricity, a break in the middle of the wire would also cause problems. Of course the fellow could shout, but the human voice is not loud enough to carry that far. An additional problem is that a string could break on the instrument. Then there could be no accompaniment to the message. It is clear that the best situation would involve less distance. Then there would be fewer potential problems. With face to face contact, the least number of things could go wrong."

To show the importance of the context on the memorization of texts, the authors assigned subjects to one of four experimental conditions:

- **1.** *"No context" condition:* subjects listened to the passage and tried to remember it.

- **2.** *"Appropriate context before" condition:* subjects were provided with an appropriate context in the form of a picture and then listened to the passage.

- **3.** *"Appropriate context after" condition:* subjects first listened to the passage and then were provided with an appropriate context in the form of a picture.

- **4.** *"Partial context" condition:* subjects are provided with a context that does not allow them to make sense of the text at the same time that they listened to the passage.

Strictly speaking this experiment involves one experimental group (group 2: "appropriate context before"), and three control groups (groups 1, 3, and 4). The *raison d'être* of the control groups is to eliminate rival theoretical hypotheses (*i.e.,* rival theories that would give the same experimental predictions as the theory advocated by the authors).

For the (fictitious) replication of this experiment, we have chosen to have 20 subjects assigned randomly to 4 groups. Hence there is $S = 5$ subjects *per* group. The dependent variable is the "number of ideas" recalled (of a maximum of 14). The results are presented below.

	No Context	Context Before	Context After	Partial Context
	3	5	2	5
	3	9	4	4
	2	8	5	3
	4	4	4	5
	3ı	9	1	4
$Y_{a.}$	15	35	16	21
$M_{a.}$	3	7	3.2	4.2

The figures taken from our SAS listing can be presented in an analysis of variance table:

Source	*df*	*SS*	*MS*	*F*	$Pr(F)$
$\mathcal{A}$	3	50.90	10.97	7.22**	.00288
$\mathcal{S}(\mathcal{A})$	16	37.60	2.35		
Total	19	88.50			

For more details on this experiment, please consult your textbook.

5.2.1 [R] code

```
# ANOVA One-factor between subjects, S(A)
# Romeo and Juliet

# We have 1 Factor, A, with 4 levels: No Context, Context Before,
# Context After, Partial Context

# We have 5 subjects per group. Therefore 5 x 4 = 20 subjects total.

# We collect the data for each level of Factor A
No_cont=c(3,3,2,4,3)
Cont_before=c(5,9,8,4,9)
Cont_after=c(2,4,5,4,1)
Part_cont=c(5,4,3,5,4)

# We now combine the observations into one long column (score).
score=c(No_cont,Cont_before, Cont_after, Part_cont)

# We generate a second column (levels), that identifies the group for
# each score.

levels=factor(c(rep("No_cont",5),rep("Cont_before",5),
  rep("Cont_after",5),rep("Part_cont",5)))

# We now form a data frame with the dependent variable and the
# factors.
data=data.frame(score=score,group=levels)

# We now generate the ANOVA table based on the linear model
aov1=aov(score~levels)

# We now print the data and all the results
print(data)
print(model.tables(aov(score~levels),"means"),digits=3)
summary(aov1)
```

5.2.2 [R] output

```
> # ANOVA One-factor between subjects, S(A)
> # Romeo and Juliet

> # We have 1 Factor, A, with 4 levels: No Context, Context Before,
> # Context After, Partial Context

> # We have 5 subjects per group. Therefore 5 x 4 = 20 subjects total.

> # We collect the data for each level of Factor A
> No_cont=c(3,3,2,4,3)
> Cont_before=c(5,9,8,4,9)
> Cont_after=c(2,4,5,4,1)
> Part_cont=c(5,4,3,5,4)

> # We now combine the observations into one long column (score).
```

```
> score=c(No_cont,Cont_before, Cont_after, Part_cont)

> # We generate a second column (levels), that identifies the group for
> # each score.

> levels=factor(c(rep("No_cont",5),rep("Cont_before",5),
  rep("Cont_after",5),rep("Part_cont",5)))

> # We now form a data frame with the dependent variable and the
> # factors.
> data=data.frame(score=score,group=levels)

> # We now generate the ANOVA table based on the linear model
> aov1=aov(score~levels)

> # We now print the data and all the results
> print(data)
```

```
       --------------------
          Score       Group
       --------------------
       1      3     No_cont
       2      3     No_cont
       3      2     No_cont
       4      4     No_cont
       5      3     No_cont
       6      5 Cont_before
       7      9 Cont_before
       8      8 Cont_before
       9      4 Cont_before
       10     9 Cont_before
       11     2  Cont_after
       12     4  Cont_after
       13     5  Cont_after
       14     4  Cont_after
       15     1  Cont_after
       16     5   Part_cont
       17     4   Part_cont
       18     3   Part_cont
       19     5   Part_cont
       20     4   Part_cont
       --------------------
```

```
> print(model.tables(aov(score~levels),"means"),digits=3)
```

```
       Tables of means

       Grand mean
       ----------
             4.35
       ----------
```

```
Levels
-----------------------------------------------
 Cont_after Cont_before     No_cont   Part_cont
-----------------------------------------------
        3.2         7.0         3.0         4.2
-----------------------------------------------
```

```
> summary(aov1)

--------------------------------------------------
            Df  Sum Sq  Mean Sq  F value    Pr(>F)
--------------------------------------------------
levels       3  50.950   16.983    7.227  0.002782 **
Residuals   16  37.600    2.350
--------------------------------------------------
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.3 Example: Face Perception, $\mathcal{S}(\mathcal{A})$ with $\mathcal{A}$ random

In a series of experiments on face perception we set out to see whether the degree of attention devoted to each face varies across faces. In order to verify this hypothesis, we assigned 40 undergraduate students to five experimental conditions. For each condition we have a man's face drawn at random from a collection of several thousand faces. We use the subjects' pupil dilation when viewing the face as an index of the attentional interest evoked by the face. The results are presented in Table 5.1 (with pupil dilation expressed in arbitrary units).

	Experimental Groups				
	Group 1	Group 2	Group 3	Group 4	Group 5
	40	53	46	52	52
	44	46	45	50	49
	45	50	48	53	49
	46	45	48	49	45
	39	55	51	47	52
	46	52	45	53	45
	42	50	44	55	52
	42	49	49	49	48
$M_{a.}$	43	50	47	51	49

TABLE 5.1 Results of a (fictitious) experiment on face perception.

5.3.1 [R] code

```
# ANOVA One-factor between subjects, S(A)
# Face Perception

# We have 1 Factor, A, with 5 levels: Group 1, Group 2, Group 3,
# Group 4, Group 5

# We have 8 subjects per group. Therefore 5 x 8 = 40 subjects total.

# We collect the data for each level of Factor A
G_1=c(40,44,45,46,39,46,42,42)
G_2=c(53,46,50,45,55,52,50,49)
G_3=c(46,45,48,48,51,45,44,49)
G_4=c(52,50,53,49,47,53,55,49)
G_5=c(52,49,49,45,52,45,52,48)

# We now combine the observations into one long column (score).
score=c(G_1,G_2,G_3,G_4,G_5)

# We generate a second column (levels), that identifies the group for each score.
levels=factor(c(rep("G_1",8),rep("G_2",8),rep("G_3",8),
rep("G_4",8),rep("G_5",8)))

# We now form a data frame with the dependent variable and
# the factors.
data=data.frame(score=score,group=levels)

# We now generate the ANOVA table based on the linear model
aov1=aov(score~levels)

# We now print the data and all the results
print(data)
print(model.tables(aov(score~levels),"means"),digits=3)
summary(aov1)
```

5.3.2 [R] output

```
> # ANOVA One-factor between subjects, S(A)
> # Face Perception

> # We have 1 Factor, A, with 5 levels: Group 1, Group 2, Group 3,
> # Group 4, Group 5

> # We have 8 subjects per group. Therefore 5 x 8 = 40 subjects total.

> # We collect the data for each level of Factor A
> G_1=c(40,44,45,46,39,46,42,42)
> G_2=c(53,46,50,45,55,52,50,49)
> G_3=c(46,45,48,48,51,45,44,49)
> G_4=c(52,50,53,49,47,53,55,49)
> G_5=c(52,49,49,45,52,45,52,48)
```

```
> # We now combine the observations into one long column (score).
> score=c(G_1,G_2,G_3,G_4,G_5)

> # We generate a second column (levels), that identifies the group for each score.
> levels=factor(c(rep("G_1",8),rep("G_2",8),rep("G_3",8),
rep("G_4",8),rep("G_5",8)))

> # We now form a data frame with the dependent variable and
> # the factors.
> data=data.frame(score=score,group=levels)

> # We now generate the ANOVA table based on the linear model
> aov1=aov(score~levels)

> # We now print the data and all the results

> print(data)
```

	Score	Group
1	40	G_1
2	44	G_1
3	45	G_1
4	46	G_1
5	39	G_1
6	46	G_1
7	42	G_1
8	42	G_1
9	53	G_2
10	46	G_2
11	50	G_2
12	45	G_2
13	55	G_2
14	52	G_2
15	50	G_2
16	49	G_2
17	46	G_3
18	45	G_3
19	48	G_3
20	48	G_3
21	51	G_3
22	45	G_3
23	44	G_3
24	49	G_3
25	52	G_4
26	50	G_4
27	53	G_4
28	49	G_4
29	47	G_4
30	53	G_4
31	55	G_4
32	49	G_4
33	52	G_5

```
34    49   G_5
35    49   G_5
36    45   G_5
37    52   G_5
38    45   G_5
39    52   G_5
40    48   G_5
--------------

> print(model.tables(aov(score~levels),"means"),digits=3)

       Tables of means

       Grand mean
       ----------
               48
       ----------

       Levels
       -----------------------
       G_1  G_2  G_3  G_4  G_5
       -----------------------
        43   50   47   51   49
       -----------------------

> summary(aov1)

       ---------------------------------------------------
                   Df  Sum Sq  Mean Sq  F value     Pr(>F)
       ---------------------------------------------------
       Levels       4     320       80       10  1.667e-05 ***
       Residuals   35     280        8
       ---------------------------------------------------
       ---
        Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.3.3 ANOVA **table**

The results of our fictitious face perception experiment are presented in the following ANOVA Table:

Source	*df*	*SS*	*MS*	*F*	Pr(*F*)
$\mathcal{A}$	4	320.00	80.00	10.00	.000020
$\mathcal{S}(\mathcal{A})$	35	280.00	8.00		
Total	39	600.00			

From this table it is clear that the research hypothesis is supported by the experimental results: All faces do not attract the same amount of attention.

5.4 Example: Images ...

In another experiment on mental imagery, we have three groups of 5 students each (psychology majors for a change!) learn a list of 40 concrete nouns and recall them one hour later. The first group learns each word with its definition, and draws the object denoted by the word (the *built image condition*). The second group was treated just like the first, but had simply to *copy* a drawing of the object instead of making it up themselves (the *given image condition*). The third group simply read the words and their definitions (the *control condition*.) Table 5.2 shows the number of words recalled 1 hour later by each subject. The experimental design is $\mathcal{S}(\mathcal{A})$, with $S = 5$, $A = 3$, and $\mathcal{A}$ as a fixed factor.

	Experimental Condition		
	Built Image	Given Image	Control
	22	13	9
	17	9	7
	24	14	10
	23	18	13
	24	21	16
$\sum$	110	75	55
$M_{a.}$	22	15	11

TABLE 5.2 Results of the mental imagery experiment.

5.4.1 [R] code

```
# ANOVA One-factor between subjects, S(A)
# Another example: Images...

# We have 1 Factor, A, with 3 levels: Built Image, Given
#Image and Control.

# We have 5 subjects per group. Therefore 5 x 3 = 15 subjects total.
# We collect the data for each level of Factor A
Built=c(22,17,24,23,24)
Given=c(13,9,14,18,21)
Control=c(9,7,10,13,16)

# We now combine the observations into one long column (score).
score=c(Built,Given,Control)

# We generate a second column (group), that identifies the group
# for each score.
```

```
levels=factor(c(rep("Built",5),rep("Given",5),rep("Control",5)))

# We now form a data frame with the dependent variable and
# the factors.
data=data.frame(score=score,group=levels)

# We now generate the ANOVA table based on the linear model
aov1=aov(score~levels)

# We now print the data and all the results
print(data)
print(model.tables(aov(score~levels),"means"),digits=3)
summary(aov1)
```

5.4.2 [R] output

```
> # ANOVA One-factor between subjects, S(A)
> # Another example: Images...

> # We have 1 Factor, A, with 3 levels: Built Image, Given
> #Image and Control.

> # We have 5 subjects per group. Therefore 5 x 3 = 15 subjects total.
> # We collect the data for each level of Factor A
> Built=c(22,17,24,23,24)
> Given=c(13,9,14,18,21)
> Control=c(9,7,10,13,16)

> # We now combine the observations into one long column (score).
> score=c(Built,Given,Control)

> # We generate a second column (group), that identifies the group
> # for each score.
> levels=factor(c(rep("Built",5),rep("Given",5),rep("Control",5)))

> # We now form a data frame with the dependent variable and
> # the factors.
> data=data.frame(score=score,group=levels)

> # We now generate the ANOVA table based on the linear model
> aov1=aov(score~levels)

> # We now print the data and all the results
> print(data)
```

```
 -----------------
    Score    Group
 -----------------
 1     22    Built
 2     17    Built
 3     24    Built
 4     23    Built
 5     24    Built
```

```
6     13    Given
7      9    Given
8     14    Given
9     18    Given
10    21    Given
11     9  Control
12     7  Control
13    10  Control
14    13  Control
15    16  Control
-----------------
```

```
> print(model.tables(aov(score~levels),"means"),digits=3)

       Tables of means

       Grand mean
       ----------
               16
       ----------
              Levels
       ---------------------
       Built  Control  Given
       ---------------------
          22       11     15
       ---------------------

> summary(aov1)

       ---------------------------------------------------
                   Df   Sum Sq  Mean Sq  F value    Pr(>F)
       ---------------------------------------------------
       levels       2  310.000  155.000   10.941  0.001974 **
       Residuals   12  170.000   14.167
       ---------------------------------------------------
       ---
       Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

5.4.3 ANOVA table

Source	*df*	*SS*	*MS*	*F*	Pr(*F*)
$\mathcal{A}$	2	310.00	155.00	10.33^{**}	.0026
$\mathcal{S}(\mathcal{A})$	12	180.00	15.00		
Total	14	490.00			

We can conclude that instructions had an effect on memorization. Using APA style (*cf.* APA manual, 1994, p. 68), to write our conclusion: “The type of instructions has an effect on memorization, $F(2, 12) = 14.10$, $MS_e = 13.07$, $p < .01$”.

6

ANOVA One Factor Between-Subjects: Regression Approach

In order to use regression to analyze data from an analysis of variance design, we use a trick that has a lot of interesting consequences. The main idea is to find a way of replacing the nominal independent variable (*i.e.,* the experimental factor) by a numerical independent variable (remember that the independent variable should be numerical to run a regression). One way of looking at analysis of variance is as a technique predicting subjects' behavior from the experimental group in which they were. The trick is to find a way of coding those groups. Several choices are possible, an easy one is to *represent a given experimental group by its mean for the dependent variable*. Remember from Chapter 4 in the textbook (on regression), that the rationale behind regression analysis implies that the independent variable is under the *control* of the experimenter. Using the group mean seems to go against this requirement, because we need to wait until *after* the experiment to know the values of the independent variable. This is why we call our procedure a *trick*. It works because it is equivalent to more elaborate coding schemes using *multiple* regression analysis. It has the advantage of being simpler both from a conceptual and computational point of view.

In this framework, the general idea is to try to predict the subjects' scores from the mean of the group to which they belong. The rationale is that, if there is an experimental effect, then the mean of a subject's group should predict the subject's score better than the grand mean. In other words, the larger the experimental effect, the better the predictive quality of the group mean. Using the group mean to predict the subjects' performance has an interesting consequence that makes regression and analysis of variance identical: *When we predict the performance of subjects from the mean of their group, the predicted value turns out to be the group mean too!*

6.1 Example: Imagery and Memory revisited

As a first illustration of the relationship between ANOVA and regression we reintroduce the experiment on Imagery and Memory detailed in Chapter 9 of your textbook. Remember that in this experiment two groups of subjects were asked to learn pairs of words (*e.g.,* "beauty-carrot"). Subjects in the first group (control group) were simply asked to learn the pairs of words the best they could. Subjects in the second group (experimental group) were asked to picture each word in a pair and to make an image of the interaction between the two objects. After, some delay, subjects in both groups were asked to give the second word (*e.g.,* "carrot") when prompted with with the first word in the pair (*e.g.,* "beauty"). For each subject, the number of words correctly recalled was recorded. The purpose of this experiment was to demonstrate an effect of the independent variable (*i.e.,* learning with imagery *versus* learning without imagery) on the dependent variable (*i.e.,* number of words correctly recalled). The results of the scaled-down version of the experiment are presented in Table 6.1.

In order to use the regression approach, we use the respective group means as predictor. See Table 6.2.

6.1.1 [R] code

```
# Regression Approach: ANOVA One-factor between subjects, S(A)
# Imagery and Memory

# We have 1 Factor, A, with 2 levels: Experimental Group and
# Control Group.

# We have 5 subjects per group. Therefore 5 x 2 = 10 subjects total.
# We collect the data for each level of Factor A
Expt=c(1,2,5,6,6)
Control=c(8,8,9,11,14)
```

Control		Experimental	
Subject 1:	1	Subject 1:	8
Subject 2:	2	Subject 2:	8
Subject 3:	5	Subject 3:	9
Subject 4:	6	Subject 4:	11
Subject 5:	6	Subject 5:	14
$M_{1.} = M_{\text{Control}} = 4$		$M_{2.} = M_{\text{Experimental}} = 10$	
Grand Mean $= M_Y = M_{..} = 7$			

TABLE 6.1 Results of the "Memory and Imagery" experiment.

$X = M_{a.}$ Predictor	4	4	4	4	4	10	10	10	10	10
Y (Value to be predicted)	1	2	5	6	6	8	8	9	11	14

TABLE 6.2 The data from Table 6.1 presented as a regression problem. The predictor X is the value of the mean of the subject's group.

```
# We now combine the observations into one long column (score).
score=c(Expt,Control)

# We generate a second column (group), that identifies the group
# for each score.
levels=factor(c(rep("Expt",5),rep("Control",5)))

# We now use the means of the respective groups as the predictors
Predictors=c(rep(mean(Expt),5),rep(mean(Control),5))

# We now form a data frame for the Regression approach
data_reg=data.frame(Predictors,score)
r=cor(Predictors,score)

# Now we perform the regression analysis on the data
reg1=lm(score~Predictors)

# We now form a data frame with the dependent variable and the
# factors for the ANOVA.
data=data.frame(score=score,group=levels)

# We now perform an ANOVA
aov1=aov(score~levels)

# We now print the data and all the results
print(data_reg)
print(model.tables(aov(score~levels),"means"),digits=3)
print(r)
summary(reg1)
summary(aov1)
```

6.1.2 [R] output

```
> # Regression Approach: ANOVA One-factor between subjects, S(A)
> # Imagery and Memory

> # We have 1 Factor, A, with 2 levels: Experimental Group and
> # Control Group.

> # We have 5 subjects per group. Therefore 5 x 2 = 10 subjects total.
> # We collect the data for each level of Factor A
> Expt=c(1,2,5,6,6)
> Control=c(8,8,9,11,14)

> # We now combine the observations into one long column (score).
> score=c(Expt,Control)
```

```
> # We generate a second column (group), that identifies the group
> # for each score.
> levels=factor(c(rep("Expt",5),rep("Control",5)))

> # We now use the means of the respective groups as the predictors
> Predictors=c(rep(mean(Expt),5),rep(mean(Control),5))

> # We now form a data frame for the Regression approach
> data_reg=data.frame(Predictors,score)
> r=cor(Predictors,score)

> # Now we perform the regression analysis on the data
> reg1=lm(score~Predictors)

> # We now form a data frame with the dependent variable and the
> # factors for the ANOVA.
> data=data.frame(score=score,group=levels)

> # We now perform an ANOVA
> aov1=aov(score~levels)

> # We now print the data and all the results
> print(data_reg)
```

	Predictors	Score
1	4	1
2	4	2
3	4	5
4	4	6
5	4	6
6	10	8
7	10	8
8	10	9
9	10	11
10	10	14

```
> print(model.tables(aov(score~levels),"means"),digits=3)
```

Tables of means
Grand mean

Grand mean
7

Levels

Control	Expt
10	4

```
> print(r)

      [1] 0.8075729

> summary(reg1)

      Call:
      lm(formula = score ~ Predictors)
             Residuals:
      -------------------------------------------------------
             Min         1Q      Median          3Q        Max
      -------------------------------------------------------
      -3.000e+00  -2.000e+00  -5.551e-17  1.750e+00  4.000e+00
      -------------------------------------------------------
             Coefficients:
      -------------------------------------------------------
                     Estimate  Std. Error    t value  Pr(>|t|)
      -------------------------------------------------------
      (Intercept)  -1.123e-15   1.966e+00  -5.71e-16   1.00000
      Predictors    1.000e+00   2.582e-01      3.873   0.00472 **
      -------------------------------------------------------
      ---
      Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
             Residual standard error: 2.449 on 8 degrees of freedom
             Multiple R-squared: 0.6522,Adjusted R-squared: 0.6087
      F-statistic:    15 on 1 and 8 DF,  p-value: 0.004721
             > summary(aov1)

      -------------------------------------------------
                  Df Sum  Sq  Mean Sq  F value    Pr(>F)
      -------------------------------------------------
      levels       1      90       90       15  0.004721 **
      Residuals    8      48        6
      -------------------------------------------------
      ---
      Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

6.2 Example: Restaging Romeo and Juliet

This second example is again the *"Romeo and Juliet"* example from a replication of Bransford *et al.*'s (1972) experiment. The rationale and details of the experiment are given in Chapter 9 in the textbook.

To refresh your memory: The general idea when using the regression approach for an analysis of variance problem is to predict subject scores from the mean of the group to which they belong. The rationale for doing so is to consider the group mean as representing the experimental effect, and hence as a predictor of the subjects' behavior. If the independent variable has an effect, the group mean should be a better predictor of the

X	3	3	3	3	3	7	7	7	7	7	3.2	3.2	3.2	3.2	3.2	4.2	4.2	4.2	4.2	4.2
Y	3	3	2	4	3	5	9	8	4	9	2	4	5	4	1	5	4	3	5	4

TABLE 6.3 The data from the Romeo and Juliet experiment presented as a regression problem. The predictor X is the value of the mean of the subject's group.

subjects behavior than the grand mean. Formally, we want to predict the score Y_{as} of subject s in condition a from a quantitative variable X that will be equal to the mean of the group a in which the s observation was collected. With an equation, we want to predict the observation by:

$$\widehat{Y} = a + bX \tag{6.1}$$

with X being equal to $M_{a.}$. The particular choice of X has several interesting consequences. A first important one, is that the mean of the predictor M_X is also the mean of the dependent variable M_Y. These two means are also equal to the grand mean of the analysis of variance. With an equation:

$$M_{..} = M_X = M_Y \,. \tag{6.2}$$

Table VI.3 gives the values needed to do the computation using the regression approach.

6.2.1 [R] code

```
# Regression Approach: ANOVA One-factor between subjects, S(A)
# Romeo and Juliet

# We have 1 Factor, A, with 4 levels: No Context, Context Before,
# Context After, Partial Context

# We have 5 subjects per group. Therefore 5 x 4 = 20 subjects total.
# We collect the data for each level of Factor A
No_cont=c(3,3,2,4,3)
Cont_before=c(5,9,8,4,9)
Cont_after=c(2,4,5,4,1)
Part_cont=c(5,4,3,5,4)

# We now combine the observations into one long column (score).
score=c(No_cont,Cont_before, Cont_after, Part_cont)

# We generate a second column (levels), that identifies the group
# for each score.
levels=factor(c(rep("No_cont",5),rep("Cont_before",5),
  rep("Cont_after",5),rep("Part_cont",5)))

# We now use the means of the respective groups as the predictors
Predictors=c(rep(mean(No_cont),5),rep(mean(Cont_before),5),
  rep(mean(Cont_after),5),rep(mean(Part_cont),5))
```

```
# We now form a data frame for the Regression approach
data_reg=data.frame(Predictors,score)
r=cor(Predictors,score)

# We now perform the regression analysis on the data
reg1=lm(score~Predictors)

# We now form a data frame with the dependent variable and the factors
# for the ANOVA.
data=data.frame(score=score,group=levels)

# We now perform an ANOVA
aov1=aov(score~levels)

# We now print the data and all the results
print(data_reg)
print(model.tables(aov(score~levels),"means"),digits=3)
print(r)
summary(reg1)
summary(aov1)
```

6.2.2 [R] output

```
> # Regression Approach: ANOVA One-factor between subjects, S(A)
> # Romeo and Juliet

> # We have 1 Factor, A, with 4 levels: No Context, Context Before,
> # Context After, Partial Context

> # We have 5 subjects per group. Therefore 5 x 4 = 20 subjects total.
> # We collect the data for each level of Factor A
> No_cont=c(3,3,2,4,3)
> Cont_before=c(5,9,8,4,9)
> Cont_after=c(2,4,5,4,1)
> Part_cont=c(5,4,3,5,4)

> # We now combine the observations into one long column (score).
> score=c(No_cont,Cont_before, Cont_after, Part_cont)

> # We generate a second column (levels), that identifies the group
> # for each score.
> levels=factor(c(rep("No_cont",5),rep("Cont_before",5),
  rep("Cont_after",5),rep("Part_cont",5)))

> # We now use the means of the respective groups as the predictors
> Predictors=c(rep(mean(No_cont),5),rep(mean(Cont_before),5),
  rep(mean(Cont_after),5),rep(mean(Part_cont),5))

> # We now form a data frame for the Regression approach
> data_reg=data.frame(Predictors,score)
> r=cor(Predictors,score)

> # We now perform the regression analysis on the data
```

```
> reg1=lm(score~Predictors)

> # We now form a data frame with the dependent variable and the factors
> # for the ANOVA.
> data=data.frame(score=score,group=levels)

> # We now perform an ANOVA
> aov1=aov(score~levels)

> # We now print the data and all the results
> print(data_reg)

      --------------------
         Predictors  Score
      --------------------
      1         3.0      3
      2         3.0      3
      3         3.0      2
      4         3.0      4
      5         3.0      3
      6         7.0      5
      7         7.0      9
      8         7.0      8
      9         7.0      4
      10        7.0      9
      11        3.2      2
      12        3.2      4
      13        3.2      5
      14        3.2      4
      15        3.2      1
      16        4.2      5
      17        4.2      4
      18        4.2      3
      19        4.2      5
      20        4.2      4
      --------------------

> print(model.tables(aov(score~levels),"means"),digits=3)

      Tables of means

      Grand mean
      ----------
            4.35
      ----------


      Levels
      -------------------------------------------
      Cont_after  Cont_before  No_cont  Part_cont
      -------------------------------------------
             3.2          7.0      3.0        4.2
      -------------------------------------------
```

```
> print(r)

      [1] 0.7585388

> summary(reg1)

      Call:
      lm(formula = score ~ Predictors)

      Residuals:
      --------------------------------------------------
            Min         1Q    Median        3Q       Max
      --------------------------------------------------
      -3.00e+00  -1.05e+00  1.04e-15  8.50e-01  2.00e+00
      --------------------------------------------------

      Coefficients:
      --------------------------------------------------------
                     Estimate  Std. Error    t value  Pr(>|t|)
      --------------------------------------------------------
      (Intercept)  -7.944e-16   9.382e-01  -8.47e-16  1.000000
      Predictors    1.000e+00   2.025e-01      4.939  0.000106 ***
      --------------------------------------------------------
      ---
      Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
      Residual standard error: 1.445 on 18 degrees of freedom

      Multiple R-squared: 0.5754,Adjusted R-squared: 0.5518
      F-statistic: 24.39 on 1 and 18 DF,  p-value: 0.0001060

> summary(aov1)

      --------------------------------------------------
                  Df  Sum Sq  Mean Sq  F value    Pr(>F)
      --------------------------------------------------
      Levels       3  50.950   16.983    7.227  0.002782 **
      Residuals   16  37.600    2.350
      --------------------------------------------------
      ---
      Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

7

ANOVA **One factor: Planned Orthogonal Comparisons**

The *planned* comparisons (also called *a priori* comparisons) are selected *before* running the experiment. In general, they correspond to the research hypothesis that is being tested. If the experiment has been designed to confront two or more alternative theories (*e.g.*, with the use of *rival* hypotheses), the comparisons are derived from those theories. When the experiment is actually run, it is possible to see if the results support or eliminate one of the theories. Because these comparisons are planned they are usually *few* in number.

A set of comparisons is composed of orthogonal comparisons if the hypotheses corresponding to each comparison are independent of each other. The maximum number of possible orthogonal comparisons one can perform, is one less than the number of levels of the independent variable (*i.e.,* $A-1$). $A-1$ is also the number of degrees of freedom of the sum of squares for $\mathcal{A}$.

All different types of comparisons can be performed following the same procedure:

- *First.* Formalization of the comparison, and expression of the comparison as a set of weights for the means.
- *Second.* Computation of the $F_{\text{comp.}}$ ratio (this is the usual F ratio adapted for the case of testing a comparison).
- *Third.* Evaluation of the probability associated with $F_{\text{comp.}}$.

7.1 Context and Memory

This example is inspired by an experiment by Smith (1979). The main purpose in this experiment was to show that to be in the same context for learning and for test can give a better performance than being in different contexts. More specifically, Smith wants to explore the effect of putting oneself *mentally* in the same context. The experiment is organized as follow. During the learning phase, subjects learn a list made of 80 words in a room painted with an orange color, decorated with posters, paintings and a decent amount of paraphernalia. A first test of learning is given then, essentially to give subjects the impression that the experiment is over. One day after, subjects are unexpectedly re-tested for their memory. An experimenter will ask them to write down all the words of the list they can remember. The test takes place in 5 different experimental conditions. Fifty subjects (10 *per* group) are randomly assigned to the experimental groups. The formula of the experimental design is $\mathcal{S}(\mathcal{A})$ or $\mathcal{S}_{10}(\mathcal{A}_5)$. The dependent variable measured is the number of words correctly recalled. The five experimental conditions are:

- **1.** *Same context.* Subjects are tested in the same room in which they learned the list.
- **2.** *Different context.* Subjects are tested in a room very different from the one in which they learned the list. The new room is located in a different part of the Campus, is painted grey, and looks very austere.
- **3.** *Imaginary context.* Subjects are tested in the same room as subjects from group 2. In addition, they are told to try to remember the room in which they learned the list. In order to help them, the experimenter asks them several questions about the room and the objects in it.
- **4.** *Photographed context.* Subjects are placed in the same condition as group 3, and, in addition, they are shown photos of the orange room in which they learned the list.
- **5.** *Placebo context.* Subjects are in the same condition as subjects in group 2. In addition, before starting to try to recall the words, they are asked first to perform a warm-up task, namely to try to remember their living room.

Several research hypotheses can be tested with those groups. Let us accept that the experiment was designed to test the following research hypotheses:

- *Research Hypothesis 1.* Groups for which the context at test matches the context during learning (*i.e.,* is the same or is simulated by imaging or photography) will perform differently (precisely they are expected to do better) than groups with a different context or than groups with a Placebo context.
- *Research Hypothesis 2.* The group with the same context will differ from the group with imaginary or photographed context.
- *Research Hypothesis 3.* The imaginary context group differs from the photographed context group

- *Research Hypothesis 4.* The different context group differs from the placebo group.

The following Table gives the set of the four contrasts specified in the SAS program.

Comparison	Gr.1	Gr.2	Gr.3	Gr.4	Gr.5
ψ_1	+2	−3	+2	+2	−3
ψ_2	+2	0	−1	−1	0
ψ_3	0	0	+1	−1	0
ψ_4	0	+1	0	0	−1

The data and results of the replication of Smith's experiment are given in the two following Tables (Tables 7.1, and 7.2).

7.1.1 [R] code

```
# ANOVA One-factor between subjects S(A)
# Smith's experiment on context effects

# NOTE 1: Install package 'gregmisc' and 'Design' in order to
# use make.contrasts and 'ols'
```

	Experimental Context				
	Group 1 Same	Group 2 Different	Group 3 Imagery	Group 4 Photo	Group 5 Placebo
	25	11	14	25	8
	26	21	15	15	20
	17	9	29	23	10
	15	6	10	21	7
	14	7	12	18	15
	17	14	22	24	7
	14	12	14	14	1
	20	4	20	27	17
	11	7	22	12	11
	21	19	12	11	4
$Y_{a.}$	180	110	170	190	100
$M_{a.}$	18	11	17	19	10
$M_{a.} - M_{..}$	3	−4	2	4	−5
$\sum(Y_{as} - Ma.)^2$	218	284	324	300	314

TABLE 7.1 Results of a replication of an experiment by Smith (1979). The dependent variable is the number of words recalled.

```
# NOTE 2: make.contrasts will augment an incomplete set of
# orthogonal contrasts with "filler" contrasts

# NOTE 3: Arrange your levels in alphabetical order, else R will
# do it for you

# We have 1 Factor, A, with 5 levels: Same, Different, Imagery,
# Photo, Placebo

# We have 10 subjects per group. Therefore 10 x 5 = 50 subjects total.

# We collect the data for each level of Factor A (in alphabetical
# order!!!)
a1_Same=c(25,26,17,15,14,17,14,20,11,21)
a2_Different=c(11,21,9,6,7,14,12,4,7,19)
a3_Imagery=c(14,15,29,10,12,22,14,20,22,12)
a4_Photo=c(25,15,23,21,18,24,14,27,12,11)
a5_Placebo=c(8,20,10,7,15,7,1,17,11,4)

# We now combine the observations into one long column (score).
score=c(a1_Same,a2_Different,a3_Imagery,a4_Photo,a5_Placebo)

# We generate a second column (group), that identifies the group
# for each score.
levels=factor(c(rep("a1_Same",10),rep("a2_Different",10),
  rep("a3_Imagery",10),rep("a4_Photo",10),rep("a5_Placebo",10)))
data=data.frame(score=score,group=levels)

# We now form the set of orthogonal contrasts
psi_1=c(2,-3,2,2,-3)
psi_2=c(2,0,-1,-1,0)
psi_3=c(0,0,1,-1,0)
psi_4=c(0,1,0,0,-1)

# We now form a matrix of contrast coefficients
cont_coeff=(rbind(psi_1,psi_2,psi_3,psi_4))

# We create a model matrix and include the contrasts as separate
# exploratory variables.
Contrasts=cbind(psi_1,psi_2,psi_3,psi_4)
model_matrix=model.matrix(~C(levels,Contrasts,base=1))
data_reg=data.frame(score,Psi_1=model_matrix[,2],Psi_2=model_
  matrix[,3],Psi_3=model_matrix[,4],Psi_4=model_matrix[,5])
Psi_1=model_matrix[,2]
Psi_2=model_matrix[,3]
Psi_3=model_matrix[,4]
Psi_4=model_matrix[,5]

# Now we perform an orthogonal multiple regression analysis on
# the data
multi_reg1=ols(score~Psi_1+Psi_2+Psi_3+Psi_4)
```

```
# We now perform an ANOVA on the contrasts
aov2=aov(score~levels,contrasts=list(levels=make.
  contrasts(cont_coeff)))

# We now perfom on ANOVA on the data
aov1=aov(score~levels)

# We now print the data and all the results
print(data)
print(data_reg)
print(multi_reg1)
summary(aov2, split = list(levels = list("psi_1" = 1,
summary(aov1)
```

7.1.2 [R] output

```
> # ANOVA One-factor between subjects S(A)
> # Smith's experiment on context effects

> # NOTE 1: Install package 'gregmisc' and 'Design' in order to
> # use make.contrasts and 'ols'

> # NOTE 2: make.contrasts will augment an incomplete set of
> # orthogonal contrasts with "filler" contrasts

> # NOTE 3: Arrange your levels in alphabetical order, else R will
> # do it for you

> # We have 1 Factor, A, with 5 levels: Same, Different, Imagery,
> # Photo, Placebo

> # We have 10 subjects per group. Therefore 10 x 5 = 50 subjects total.

> # We collect the data for each level of Factor A (in alphabetical
> # order!!!)
> a1_Same=c(25,26,17,15,14,17,14,20,11,21)
> a2_Different=c(11,21,9,6,7,14,12,4,7,19)
> a3_Imagery=c(14,15,29,10,12,22,14,20,22,12)
> a4_Photo=c(25,15,23,21,18,24,14,27,12,11)
> a5_Placebo=c(8,20,10,7,15,7,1,17,11,4)

> # We now combine the observations into one long column (score).
> score=c(a1_Same,a2_Different,a3_Imagery,a4_Photo,a5_Placebo)

> # We generate a second column (group), that identifies the group
> # for each score.
> levels=factor(c(rep("a1_Same",10),rep("a2_Different",10),
  rep("a3_Imagery",10),rep("a4_Photo",10),rep("a5_Placebo",10)))
> data=data.frame(score=score,group=levels)

> # We now form the set of orthogonal contrasts
> psi_1=c(2,-3,2,2,-3)
> psi_2=c(2,0,-1,-1,0)
```

```
> psi_3=c(0,0,1,-1,0)
> psi_4=c(0,1,0,0,-1)

> # We now form a matrix of contrast coefficients
> cont_coeff=(rbind(psi_1,psi_2,psi_3,psi_4))

> # We create a model matrix and include the contrasts as separate
> # exploratory variables.
> Contrasts=cbind(psi_1,psi_2,psi_3,psi_4)
> model_matrix=model.matrix(~C(levels,Contrasts,base=1))
> data_reg=data.frame(score,Psi_1=model_matrix[,2],Psi_2=model_
  matrix[,3],Psi_3=model_matrix[,4],Psi_4=model_matrix[,5])
> Psi_1=model_matrix[,2]
> Psi_2=model_matrix[,3]
> Psi_3=model_matrix[,4]
> Psi_4=model_matrix[,5]

> # Now we perform an orthogonal multiple regression analysis on
> # the data
> multi_reg1=ols(score~Psi_1+Psi_2+Psi_3+Psi_4)

> # We now perform an ANOVA on the contrasts
> aov2=aov(score~levels,contrasts=list(levels=make.
  contrasts(cont_coeff)))

> # We now perfom on ANOVA on the data
> aov1=aov(score~levels)

> # We now print the data and all the results
> print(data)
```

```
---------------------
   Score        Group
---------------------
1     25      a1_Same
2     26      a1_Same
3     17      a1_Same
4     15      a1_Same
5     14      a1_Same
6     17      a1_Same
7     14      a1_Same
8     20      a1_Same
9     11      a1_Same
10    21      a1_Same
11    11 a2_Different
12    21 a2_Different
13     9 a2_Different
14     6 a2_Different
15     7 a2_Different
16    14 a2_Different
17    12 a2_Different
18     4 a2_Different
19     7 a2_Different
20    19 a2_Different
```

```
21    14   a3_Imagery
22    15   a3_Imagery
23    29   a3_Imagery
24    10   a3_Imagery
25    12   a3_Imagery
26    22   a3_Imagery
27    14   a3_Imagery
28    20   a3_Imagery
29    22   a3_Imagery
30    12   a3_Imagery
31    25     a4_Photo
32    15     a4_Photo
33    23     a4_Photo
34    21     a4_Photo
35    18     a4_Photo
36    24     a4_Photo
37    14     a4_Photo
38    27     a4_Photo
39    12     a4_Photo
40    11     a4_Photo
41     8   a5_Placebo
42    20   a5_Placebo
43    10   a5_Placebo
44     7   a5_Placebo
45    15   a5_Placebo
46     7   a5_Placebo
47     1   a5_Placebo
48    17   a5_Placebo
49    11   a5_Placebo
50     4   a5_Placebo
---------------------
```

```
> print(data_reg)

   ------------------------------------
      Score  Psi_1  Psi_2  Psi_3  Psi_4
   ------------------------------------
   1     25      2      2      0      0
   2     26      2      2      0      0
   3     17      2      2      0      0
   4     15      2      2      0      0
   5     14      2      2      0      0
   6     17      2      2      0      0
   7     14      2      2      0      0
   8     20      2      2      0      0
   9     11      2      2      0      0
   10    21      2      2      0      0
   11    11     -3      0      0      1
   12    21     -3      0      0      1
   13     9     -3      0      0      1
   14     6     -3      0      0      1
   15     7     -3      0      0      1
   16    14     -3      0      0      1
   17    12     -3      0      0      1
```

```
18   4  -3   0   0   1
19   7  -3   0   0   1
20  19  -3   0   0   1
21  14   2  -1   1   0
22  15   2  -1   1   0
23  29   2  -1   1   0
24  10   2  -1   1   0
25  12   2  -1   1   0
26  22   2  -1   1   0
27  14   2  -1   1   0
28  20   2  -1   1   0
29  22   2  -1   1   0
30  12   2  -1   1   0
31  25   2  -1  -1   0
32  15   2  -1  -1   0
33  23   2  -1  -1   0
34  21   2  -1  -1   0
35  18   2  -1  -1   0
36  24   2  -1  -1   0
37  14   2  -1  -1   0
38  27   2  -1  -1   0
39  12   2  -1  -1   0
40  11   2  -1  -1   0
41   8  -3   0   0  -1
42  20  -3   0   0  -1
43  10  -3   0   0  -1
44   7  -3   0   0  -1
45  15  -3   0   0  -1
46   7  -3   0   0  -1
47   1  -3   0   0  -1
48  17  -3   0   0  -1
49  11  -3   0   0  -1
50   4  -3   0   0  -1
------------------------------------
```

```
> print(multi_reg1)

       Linear Regression Model

      ols(formula = score ~ Psi_1 + Psi_2 + Psi_3 + Psi_4)
       ----------------------------------
        n  Model L.R. d.f.      R2  Sigma
       ----------------------------------
       50       19.81    4  0.3271  5.657
       ----------------------------------

       Residuals:
       ----------------------------------
          Min     1Q Median     3Q    Max
       ----------------------------------
        -9.00  -4.00  -1.00   4.75  12.00
       ----------------------------------
```

```
      Coefficients:
      ------------------------------------------------------
                     Value  Std. Error         t   Pr(>|t|)
      ------------------------------------------------------
      Intercept  1.500e+01      0.8000  1.875e+01  0.000e+00
      Psi_1      1.500e+00      0.3266  4.593e+00  3.521e-05
      Psi_2      1.174e-16      0.7303  1.608e-16  1.000e+00
      Psi_3     -1.000e+00      1.2649 -7.906e-01  4.333e-01
      Psi_4      5.000e-01      1.2649  3.953e-01  6.945e-01
      ------------------------------------------------------

      Residual standard error: 5.657 on 45 degrees of freedom
      Adjusted R-Squared: 0.2673

> summary(aov2, split = list(levels = list("psi_1" = 1,
  "psi_2" = 2, "psi_3" =3 , "psi_4" = 4)))

      --------------------------------------------------------------
                      Df     Sum Sq    Mean Sq    F value     Pr(>F)
      --------------------------------------------------------------
      levels           4        700        175     5.4687   0.001125 **
        levels: psi_1  1        675        675    21.0937  3.521e-05 ***
        levels: psi_2  1  4.127e-29  4.127e-29  1.290e-30   1.000000
        levels: psi_3  1         20         20     0.6250   0.433342
        levels: psi_4  1          5          5     0.1563   0.694500
      Residuals       45       1440         32
      --------------------------------------------------------------
      ---      Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> summary(aov1)

      ------------------------------------------------
                  Df  Sum Sq  Mean Sq  F value    Pr(>F)
      ------------------------------------------------
      Levels       4     700      175   5.4688  0.001125 **
      Residuals   45    1440       32
      ------------------------------------------------
      ---      Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

7.1.3 ANOVA table

Source	*df*	*SS*	*MS*	*F*	Pr(*F*)
$\mathcal{A}$	4	700.00	175.00	5.469^{**}	.00119
$\mathcal{S}(\mathcal{A})$	45	1,440.00	32.00		
Total	49	2,140.00			

TABLE 7.2 ANOVA Table for a replication of Smith's (1979) experiment.

8
Planned Non-orthogonal Comparisons

Non-orthogonal comparisons are more complex than orthogonal comparisons. The main problem lies in assessing the importance of a given comparison independently of the other comparisons of the set. There are currently two (main) approaches to this problem. The classical approach corrects for multiple statistical tests (*i.e.,* using a Šidàk or Bonferonni correction), but essentially evaluates each contrast as if it were coming from a set of orthogonal contrasts. The multiple regression (or modern) approach evaluates each contrast as a predictor from a set of non-orthogonal predictors and estimates its *specific* contribution to the explanation of the dependent variable.

8.1 Classical approach: Tests for non-orthogonal comparisons

The Šidàk or the Bonferonni, Boole, Dunn inequalityare used to find a correction on $\alpha[PC]$ in order to keep $\alpha[PF]$ fixed. The general idea of the procedure is to correct $\alpha[PC]$ in order to obtain the overall $\alpha[PF]$ for the experiment. By deciding that the family is the unit for evaluating Type I error, the inequalities give an approximation for each $\alpha[PC]$. The formula used to evaluate the alpha level for each comparison using the Šidàk inequality is:

$$\alpha[PC] \approx 1 - (1 - \alpha[PF])^{1/C} \ .$$

This is a conservative approximation, because the following inequality holds:

$$\alpha[PC] \geq 1 - (1 - \alpha[PF])^{1/C} \ .$$

The formula used to evaluate the alpha level for each comparison using Bonferonni, Boole, Dunn inequality would be:

$$\alpha[PC] \approx \frac{\alpha[PF]}{C} \ .$$

By using these approximations, the statistical test will be a *conservative* one. That is to say, the real value of $\alpha[PF]$ will always be smaller than the approximation we use. For example, suppose you want to perform four non-orthogonal comparisons, and that you want to limit the risk of making at least one Type I error to an overall value of $\alpha[PF] = .05$. Using the Šidàk correction you will consider that any comparison of the family reaches significance if the probability associated with it is smaller than:

$$\alpha[PC] = 1 - (1 - \alpha[PF])^{1/C} = 1 - (1 - .05)^{1/4} = .0127$$

Note, this is a change from the usual $.05$ and $.01$.

8.2 Romeo and Juliet, non-orthogonal contrasts

An example will help to review this section. Again, let us return to Bransford's "Romeo and Juliet". The following Table gives the different experimental conditions:

Context Before	Partial Context	Context After	Without Context

Suppose that Bransford had build his experiment to test *a priori* four research hypotheses:

- **1.** The presence of any context has an effect.
- **2.** The context given after the story has an effect.
- **3.** The context given before has a stronger effect than any other condition.
- **4.** The partial context condition differs from the "context before" condition.

These hypotheses can easily be translated into a set of contrasts given in the following Table.

	Context Before	Partial Context	Context After	Without Context
ψ_1	1	1	1	-3
ψ_2	0	0	1	-1
ψ_3	3	-1	-1	-1
ψ_4	1	-1	0	0

If $\alpha[PF]$ is set to the value $.05$, this will lead to testing each contrast with the $\alpha[PC]$ level:

$$\alpha[PC] = 1 - .95^{1/4} = .0127 \,.$$

If you want to use the critical values method, the Table gives for $\nu_2 = 16$ (this is the number of degrees of freedom of $MS_{S(A)}$), $\alpha[PF] = .05$, and $C = 4$ the value $F_{\text{critical Šidàk}} = 7.91$ (This is simply the critical value of the standard Fisher F with 1 and 16 degrees of freedom and with $\alpha = \alpha[PC] = .0127$).

8.2.1 [R] code

```
# ANOVA One-factor between subjects S(A)
# Romeo and Juliet - Non-Orthogonal Contrasts with Sidak
#      correction

# NOTE 1: Install and load package 'gregmisc' in order to use
#   make.contrasts
# NOTE 2: make.contrasts will augment an incomplete set of
#   orthogonal contrasts with "filler" contrasts
# NOTE 3: Arrange your levels in alphabetical order, else R
#   will do it for you

# We have 1 Factor, A, with 4 levels:
# Context Before,
# Partial Context,
# Context After,
# No Context

# We have 5 subjects per group. Therefore 5 x 4 = 20 subjects
#   total.

# We collect the data for each level of Factor A
a1_Cont_before=c(5,9,8,4,9)
a2_Part_cont=c(5,4,3,5,4)
a3_Cont_after=c(2,4,5,4,1)
a4_No_cont=c(3,3,2,4,3)

# We now combine the observations into one long column (score).
score=c(a1_Cont_before,a2_Part_cont, a3_Cont_after, a4_No_cont)

# We generate a second column (levels) that identifies the group
# for each score.

levels=factor(c(rep("a1_Cont_before",5),rep("a2_Part_cont",5),
       rep("a3_Cont_after",5),rep("a4_No_cont",5)))

# We now form a data frame with the dependent variable and the
#   factors.
data=data.frame(score=score,group=levels)

# We now define the non-orthogonal contrasts
C_1=c(1,1,1,-3)
```

```
C_2=c(0,0,1,-1)
C_3=c(3,-1,-1,-1)
C_4=c(1,-1,0,0)

# We now perform the test for multiple comparisons using "Sidak"
#   correction.
# The means with different letters are significantly different.

# NOTE: The source for R script "mulcomp" has to be specified.

means=tapply(score,levels,mean)
source("?/R_scripts/08_Planned_Non_Ortho_Cont/
       mulcomp.R")
multi_comp=mulcomp(as.vector(means),5,16,2.350,conf.level=
.05,type= "Sidak",decreasing=TRUE)

# We now perfom on ANOVA on the data
aov5=aov(score~levels)

# We now organize the results
Df_psi_1=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_1))),split = list(levels =
       list("C_1" = 1)))[[1]]$Df
Df_psi_2=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_2))),split = list(levels =
       list("C_2" = 1)))[[1]]$Df
Df_psi_3=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_3))),split = list(levels =
       list("C_3" = 1)))[[1]]$Df
Df_psi_4=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_4))), split = list(levels =
       list("C_4" = 1)))[[1]]$Df
Df_psi=data.frame(rbind(Df_psi_1,Df_psi_2,Df_psi_3,
       Df_psi_4))
SS_psi_1=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_1))),split = list(levels =
       list("C_1" = 1)))[[1]]$Sum
SS_psi_2=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_2))),split = list(levels =
       list("C_2" = 1)))[[1]]$Sum
SS_psi_3=summary(aov(score~levels,contrasts=
       list(levels=make.contrasts(C_3))),split =
       list(levels = list("C_3" = 1)))[[1]]$Sum
SS_psi_4=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_4))),split = list(levels =
       list("C_4" = 1)))[[1]]$Sum
SS_psi=data.frame(rbind(SS_psi_1,SS_psi_2,SS_psi_3,SS_psi_4))
MS_psi_1=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_1))),split = list(levels =
       list("C_1" = 1)))[[1]]$Mean
MS_psi_2=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_2))),split = list(levels =
       list("C_2" = 1)))[[1]]$Mean
MS_psi_3=summary(aov(score~levels,contrasts=list(levels=
```

```
       make.contrasts(C_3))),split = list(levels =
       list("C_3" = 1)))[[1]]$Mean
MS_psi_4=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_4))),split = list(levels =
       list("C_4" = 1)))[[1]]$Mean
MS_psi=data.frame(rbind(MS_psi_1,MS_psi_2,MS_psi_3,MS_psi_4))
F_psi_1=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_1))),split = list(levels =
       list("C_1" = 1)))[[1]]$F
F_psi_2=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_2))),split = list(levels =
       list("C_2" = 1)))[[1]]$F
F_psi_3=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_3))),split = list(levels =
       list("C_3" = 1)))[[1]]$F
F_psi_4=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_4))),split = list(levels =
       list("C_4" = 1)))[[1]]$F
F_psi=data.frame(rbind(F_psi_1,F_psi_2,F_psi_3,F_psi_4))
Pr_psi_1=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_1))),split = list(levels =
       list("C_1" = 1)))[[1]]$Pr
Pr_psi_2=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_2))),split = list(levels =
       list("C_2" = 1)))[[1]]$Pr
Pr_psi_3=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_3))),split = list(levels =
       list("C_3" = 1)))[[1]]$Pr
Pr_psi_4=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_4))),split = list(levels =
       list("C_4" = 1)))[[1]]$Pr
Pr_psi=data.frame(rbind(Pr_psi_1,Pr_psi_2,Pr_psi_3,Pr_psi_4))
Contrast_names=c("Psi_1","Psi_2","Psi_3","Psi_4")
Cont_mat=rbind("Psi_1"=C_1,"Psi_2"=C_2,"Psi_3"=C_3,"Psi_4"=C_4)
Contrasts=data.frame(G1=Cont_mat[,1],G2=Cont_mat[,2],
       G3=Cont_mat[,3],G4=Cont_mat[,4])
Contrast_Summary=data.frame(Contrast=Contrast_names,DF=Df_psi[,2],
       Contrast_SS=SS_psi[,2],Mean_Square=MS_psi[,2],F_Value=F_psi[,2],
       Pr=Pr_psi[,2])

# We now print the data and all the results
print(data)
summary(aov5)
print(Contrasts)
print(Contrast_Summary)
print(multi_comp)
print('Means with the same letter are not significantly different')
```

8.2.2 [R] output

```
> # ANOVA One-factor between subjects S(A)
> # Romeo and Juliet - Non-Orthogonal Contrasts with Sidak
> #      correction
```

```
> # NOTE 1: Install and load package 'gregmisc' in order to use
> #   make.contrasts
> # NOTE 2: make.contrasts will augment an incomplete set of
> #   orthogonal contrasts with "filler" contrasts
> # NOTE 3: Arrange your levels in alphabetical order, else R
> #   will do it for you

> # We have 1 Factor, A, with 4 levels:
> # Context Before,
> # Partial Context,
> # Context After,
> # No Context

> # We have 5 subjects per group. Therefore 5 x 4 = 20 subjects
> #   total.

> # We collect the data for each level of Factor A
> a1_Cont_before=c(5,9,8,4,9)
> a2_Part_cont=c(5,4,3,5,4)
> a3_Cont_after=c(2,4,5,4,1)
> a4_No_cont=c(3,3,2,4,3)

> # We now combine the observations into one long column (score).
> score=c(a1_Cont_before,a2_Part_cont, a3_Cont_after, a4_No_cont)

> # We generate a second column (levels) that identifies the group
> # for each score.

> levels=factor(c(rep("a1_Cont_before",5),rep("a2_Part_cont",5),
     rep("a3_Cont_after",5),rep("a4_No_cont",5)))

> # We now form a data frame with the dependent variable and the
> #   factors.
> data=data.frame(score=score,group=levels)

> # We now define the non-orthogonal contrasts
> C_1=c(1,1,1,-3)
> C_2=c(0,0,1,-1)
> C_3=c(3,-1,-1,-1)
> C_4=c(1,-1,0,0)

> # We now perform the test for multiple comparisons using "Sidak"
> #   correction.
> # The means with different letters are significantly different.

> # NOTE: The source for R script "mulcomp" has to be specified.

> means=tapply(score,levels,mean)
> source("?/R_scripts/08_Planned_Non_Ortho_Cont/
     mulcomp.R")
> multi_comp=mulcomp(as.vector(means),5,16,2.350,conf.level=
.05,type= "Sidak",decreasing=TRUE)
```

```
> # We now perfom on ANOVA on the data
> aov5=aov(score~levels)

> # We now organize the results
> Df_psi_1=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_1))),split = list(levels =
      list("C_1" = 1)))[[1]]$Df
> Df_psi_2=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_2))),split = list(levels =
      list("C_2" = 1)))[[1]]$Df
> Df_psi_3=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_3))),split = list(levels =
      list("C_3" = 1)))[[1]]$Df
> Df_psi_4=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_4))), split = list(levels =
      list("C_4" = 1)))[[1]]$Df
> Df_psi=data.frame(rbind(Df_psi_1,Df_psi_2,Df_psi_3,
      Df_psi_4))
> SS_psi_1=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_1))),split = list(levels =
      list("C_1" = 1)))[[1]]$Sum
> SS_psi_2=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_2))),split = list(levels =
      list("C_2" = 1)))[[1]]$Sum
> SS_psi_3=summary(aov(score~levels,contrasts=
      list(levels=make.contrasts(C_3))),split =
      list(levels = list("C_3" = 1)))[[1]]$Sum
> SS_psi_4=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_4))),split = list(levels =
      list("C_4" = 1)))[[1]]$Sum
> SS_psi=data.frame(rbind(SS_psi_1,SS_psi_2,SS_psi_3,SS_psi_4))
> MS_psi_1=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_1))),split = list(levels =
      list("C_1" = 1)))[[1]]$Mean
> MS_psi_2=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_2))),split = list(levels =
      list("C_2" = 1)))[[1]]$Mean
> MS_psi_3=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_3))),split = list(levels =
      list("C_3" = 1)))[[1]]$Mean
> MS_psi_4=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_4))),split = list(levels =
      list("C_4" = 1)))[[1]]$Mean
> MS_psi=data.frame(rbind(MS_psi_1,MS_psi_2,MS_psi_3,MS_psi_4))
> F_psi_1=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_1))),split = list(levels =
      list("C_1" = 1)))[[1]]$F
> F_psi_2=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_2))),split = list(levels =
      list("C_2" = 1)))[[1]]$F
> F_psi_3=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_3))),split = list(levels =
      list("C_3" = 1)))[[1]]$F
> F_psi_4=summary(aov(score~levels,contrasts=list(levels=
```

```
      make.contrasts(C_4))),split = list(levels =
      list("C_4" = 1)))[[1]]$F
> F_psi=data.frame(rbind(F_psi_1,F_psi_2,F_psi_3,F_psi_4))
> Pr_psi_1=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_1))),split = list(levels =
      list("C_1" = 1)))[[1]]$Pr
> Pr_psi_2=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_2))),split = list(levels =
      list("C_2" = 1)))[[1]]$Pr
> Pr_psi_3=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_3))),split = list(levels =
      list("C_3" = 1)))[[1]]$Pr
> Pr_psi_4=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_4))),split = list(levels =
      list("C_4" = 1)))[[1]]$Pr
> Pr_psi=data.frame(rbind(Pr_psi_1,Pr_psi_2,Pr_psi_3,Pr_psi_4))
> Contrast_names=c("Psi_1","Psi_2","Psi_3","Psi_4")
> Cont_mat=rbind("Psi_1"=C_1,"Psi_2"=C_2,"Psi_3"=C_3,"Psi_4"=C_4)
> Contrasts=data.frame(G1=Cont_mat[,1],G2=Cont_mat[,2],
      G3=Cont_mat[,3],G4=Cont_mat[,4])
> Contrast_Summary=data.frame(Contrast=Contrast_names,DF=Df_psi[,2],
      Contrast_SS=SS_psi[,2],Mean_Square=MS_psi[,2],F_Value=F_psi[,2],
      Pr=Pr_psi[,2])

> # We now print the data and all the results
> print(data)

      ----------------------------
         Score              Group
      ----------------------------
      1      5      a1_Cont_before
      2      9      a1_Cont_before
      3      8      a1_Cont_before
      4      4      a1_Cont_before
      5      9      a1_Cont_before
      6      5      a2_Part_cont
      7      4      a2_Part_cont
      8      3      a2_Part_cont
      9      5      a2_Part_cont
      10     4      a2_Part_cont
      11     2      a3_Cont_after
      12     4      a3_Cont_after
      13     5      a3_Cont_after
      14     4      a3_Cont_after
      15     1      a3_Cont_after
      16     3      a4_No_cont
      17     3      a4_No_cont
      18     2      a4_No_cont
      19     4      a4_No_cont
      20     3      a4_No_cont
      ----------------------------

> summary(aov5)
```

```
      -------------------------------------------------
                 Df  Sum Sq  Mean Sq  F value    Pr(>F)
      -------------------------------------------------
      Levels       3  50.950   16.983    7.227  0.002782 **
      Residuals   16  37.600    2.350
      -------------------------------------------------
      ---
      Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> print(Contrasts)

      ---------------------
             G1  G2  G3  G4
      ---------------------
      Psi_1   1   1   1  -3
      Psi_2   0   0   1  -1
      Psi_3   3  -1  -1  -1
      Psi_4   1  -1   0   0
      ---------------------

> print(Contrast_Summary)

      ---------------------------------------------------------------
        Contrast  DF Contrast_SS Mean_Square     F_Value           Pr
      ---------------------------------------------------------------
      1    Psi_1   1    12.15000    12.15000  5.17021277 0.0371045233
      2    Psi_2   1     0.10000     0.10000  0.04255319 0.8391709477
      3    Psi_3   1    46.81667    46.81667 19.92198582 0.0003921401
      4    Psi_4   1    19.60000    19.60000  8.34042553 0.0107046712
      ---------------------------------------------------------------

> print(multi_comp)

      $sem
      [1] 0.6855655

      $sed
      [1] 0.969536

      $MCT
      --------------
         medie  test
      --------------
      M1   7.0     a
      M2   4.2    ab
      M3   3.2     b
      M4   3.0     b
      --------------

      $Critical.Differences
      [1] 0.000000 2.911798 2.911798 2.911798


> print('Means with the same letter are not significantly different')
```

```
[1] "Means with the same letter are not significantly different"
```

8.3 Multiple Regression and Orthogonal Contrasts

ANOVA and multiple regression are equivalent if we use as many predictors for the multiple regression analysis as the number of degrees of freedom of the independent variable. An obvious choice for the predictors is to use a set of contrasts. Doing so makes contrast analysis a particular case of multiple regression analysis. Regression analysis, in return, helps solving some of the problems associated with the use of non-orthogonal contrasts: It suffices to use multiple regression and semi-partial coefficients of correlation to analyze non-orthogonal contrasts.

In this section, we illustrate the multiple regression analysis approach of the Bransford experiment (*i.e.,* "Romeo and Juliet"). We will first look at a set of orthogonal contrasts and then a set of non-orthogonal contrasts.

A set of data from a replication of this experiment is given in Table 8.1.

Experimental Condition			
Context before	Partial context	Context after	Without context
5	5	2	3
9	4	4	3
8	3	5	2
4	5	4	4
9	4	1	3

TABLE 8.1 The data from a replication of Bransford's "Romeo and Juliet" experiment. $M_{..} = 4.35$.

In order to analyze these data with a multiple regression approach we can use any arbitrary set of contrasts as long as they satisfy the following constraints:

1. there are as many contrasts as the independent variable has degrees of freedom,
2. the set of contrasts is not multicolinear. That is, no contrast can be obtained by combining the other contrasts[1].

[1]The technical synonym for non-multicolinear is *linearly independent*.

	Groups			
Contrast	1	2	3	4
ψ_1	1	1	−1	−1
ψ_2	1	−1	0	0
ψ_3	0	0	1	−1

TABLE 8.2 An arbitrary set of orthogonal contrasts for analyzing "Romeo and Juliet."

8.3.1 [R] code

```
# ANOVA One-factor between subjects S(A)
# Romeo and Juliet: Orthogonal Multiple Regression

# Install and load package "Design" and "gregmisc".
# NOTE 1: Arrange your levels in alphabetical order, else R
#   will do it for you


# We have 1 Factor, A, with 4 levels:
# Context Before,
# Partial Context,
# Context After,
# No Context

# We have 5 subjects per group. Therefore 5 x 4 = 20 subjects
#   total.

# We collect the data for each level of Factor A
a1_Cont_before=c(5,9,8,4,9)
a2_Part_cont=c(5,4,3,5,4)
a3_Cont_after=c(2,4,5,4,1)
a4_No_cont=c(3,3,2,4,3)

# We now combine the observations into one long column (score).
score=c(a1_Cont_before,a2_Part_cont, a3_Cont_after, a4_No_cont)

# We generate a second column (levels), that identifies the group
#   for each score.
levels=factor(c(rep("a1_Cont_before",5),rep("a2_Part_cont",5),
       rep("a3_Cont_after",5),rep("a4_No_cont",5)))

# We now form a data frame with the dependent variable and
#   the factors.
data=data.frame(score=score,group=levels)

# We now perfom an S(A) anova on the data
aov1=aov(score~levels)

# We now define the Orthogonal contrasts
C_1=c(1,1,-1,-1)
```

```
C_2=c(1,-1,0,0)
C_3=c(0,0,1,-1)

# We create a model matrix and include the contrasts as separate
#   exploratory variables.
cont_coeff=cbind(C_1,C_2,C_3)
model_matrix=model.matrix(~C(levels,cont_coeff,base=1))
data_reg=data.frame(score,psi_1=model_matrix[,2],psi_2=
       model_matrix[,3],psi_3=model_matrix[,4])
psi_1=model_matrix[,2]
psi_2=model_matrix[,3]
psi_3=model_matrix[,4]

# Now we perform an orthogonal multiple regression analysis on the data
multi_reg1=ols(score~psi_1+psi_2+psi_3)

# We now organize the results
Df_psi_1=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_1))),split = list(levels =
       list("C_1" = 1)))[[1]]$Df
Df_psi_2=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_2))),split = list(levels =
       list("C_2" = 1)))[[1]]$Df
Df_psi_3=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_3))),split = list(levels =
       list("C_3" = 1)))[[1]]$Df
Df_psi=data.frame(rbind(Df_psi_1,Df_psi_2,Df_psi_3))
SS_psi_1=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_1))),split = list(levels =
       list("C_1" = 1)))[[1]]$Sum
SS_psi_2=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_2))),split = list(levels =
       list("C_2" = 1)))[[1]]$Sum
SS_psi_3=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_3))),split = list(levels =
       list("C_3" = 1)))[[1]]$Sum
SS_psi=data.frame(rbind(SS_psi_1,SS_psi_2,SS_psi_3))
MS_psi_1=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_1))),split = list(levels =
       list("C_1" = 1)))[[1]]$Mean
MS_psi_2=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_2))),split = list(levels =
       list("C_2" = 1)))[[1]]$Mean
MS_psi_3=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_3))),split = list(levels =
       list("C_3" = 1)))[[1]]$Mean
MS_psi=data.frame(rbind(MS_psi_1,MS_psi_2,MS_psi_3))
F_psi_1=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_1))),split = list(levels =
       list("C_1" = 1)))[[1]]$F
F_psi_2=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_2))),split = list(levels =
       list("C_2" = 1)))[[1]]$F
F_psi_3=summary(aov(score~levels,contrasts=list(levels=
```

```
        make.contrasts(C_3))),split = list(levels =
        list("C_3" = 1)))[[1]]$F
F_psi=data.frame(rbind(F_psi_1,F_psi_2,F_psi_3))
Pr_psi_1=summary(aov(score~levels,contrasts=list(levels=
        make.contrasts(C_1))),split = list(levels =
        list("C_1" = 1)))[[1]]$Pr
Pr_psi_2=summary(aov(score~levels,contrasts=list(levels=
        make.contrasts(C_2))),split = list(levels =
        list("C_2" = 1)))[[1]]$Pr
Pr_psi_3=summary(aov(score~levels,contrasts=list(levels=
        make.contrasts(C_3))),split = list(levels =
        list("C_3" = 1)))[[1]]$Pr
Pr_psi=data.frame(rbind(Pr_psi_1,Pr_psi_2,Pr_psi_3))
Contrast_names=c("Psi_1","Psi_2","Psi_3")
Cont_mat=rbind("Psi_1"=C_1,"Psi_2"=C_2,"Psi_3"=C_3)
Contrasts=data.frame(G1=Cont_mat[,1],G2=Cont_mat[,2],
        G3=Cont_mat[,3],G4=Cont_mat[,4])
Contrast_Summary=data.frame(Contrast=Contrast_names,DF=
        Df_psi[,2],Contrast_SS=SS_psi[,2],Mean_Square=MS_psi[,2],
        F_Value=F_psi[,2],Pr=Pr_psi[,2])
Contrast=c("Psi_1","Psi_2","Psi_3")
Contrast_Summary=data.frame(Contrast=Contrast,DF=Df_psi[,2],
        Contrast_SS=SS_psi[,2],Mean_Square=MS_psi[,2],F_Value=
        F_psi[,2],Pr=Pr_psi[,2])

# Now we compute the semi-partial coefficients
# Make sure to add the PATH to the location where the plot is to be saved
pdf('/R_scripts/08_Planned_Non_Ortho_Cont/semi_part_corr.pdf')
semi_part=plot(anova(multi_reg1),what='partial R2')
dev.off()

# Now we print the data and all the results
print(data)
print(data_reg)
summary.aov(aov1)
print(Contrast_Summary)
print(multi_reg1)
print(semi_part)
```

8.3.2 [R] output

```
> # ANOVA One-factor between subjects S(A)
> # Romeo and Juliet: Orthogonal Multiple Regression

> # Install and load package "Design" and "gregmisc".
> # NOTE 1: Arrange your levels in alphabetical order, else R
> #   will do it for you


> # We have 1 Factor, A, with 4 levels:
> # Context Before,
> # Partial Context,
> # Context After,
```

```
> # No Context

> # We have 5 subjects per group. Therefore 5 x 4 = 20 subjects
> #   total.

> # We collect the data for each level of Factor A
> a1_Cont_before=c(5,9,8,4,9)
> a2_Part_cont=c(5,4,3,5,4)
> a3_Cont_after=c(2,4,5,4,1)
> a4_No_cont=c(3,3,2,4,3)

> # We now combine the observations into one long column (score).
> score=c(a1_Cont_before,a2_Part_cont, a3_Cont_after, a4_No_cont)

> # We generate a second column (levels), that identifies the group
> #   for each score.
> levels=factor(c(rep("a1_Cont_before",5),rep("a2_Part_cont",5),
      rep("a3_Cont_after",5),rep("a4_No_cont",5)))

> # We now form a data frame with the dependent variable and
> #   the factors.
> data=data.frame(score=score,group=levels)

> # We now perfom an S(A) anova on the data
> aov1=aov(score~levels)

> # We now define the Orthogonal contrasts
> C_1=c(1,1,-1,-1)
> C_2=c(1,-1,0,0)
> C_3=c(0,0,1,-1)

> # We create a model matrix and include the contrasts as separate
> #   exploratory variables.
> cont_coeff=cbind(C_1,C_2,C_3)
> model_matrix=model.matrix(~C(levels,cont_coeff,base=1))
> data_reg=data.frame(score,psi_1=model_matrix[,2],psi_2=
      model_matrix[,3],psi_3=model_matrix[,4])
> psi_1=model_matrix[,2]
> psi_2=model_matrix[,3]
> psi_3=model_matrix[,4]

> # Now we perform an orthogonal multiple regression analysis on the data
> multi_reg1=ols(score~psi_1+psi_2+psi_3)

> # We now organize the results
> Df_psi_1=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_1))),split = list(levels =
      list("C_1" = 1)))[[1]]$Df
> Df_psi_2=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_2))),split = list(levels =
      list("C_2" = 1)))[[1]]$Df
> Df_psi_3=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_3))),split = list(levels =
      list("C_3" = 1)))[[1]]$Df
```

```
> Df_psi=data.frame(rbind(Df_psi_1,Df_psi_2,Df_psi_3))
> SS_psi_1=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_1))),split = list(levels =
      list("C_1" = 1)))[[1]]$Sum
> SS_psi_2=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_2))),split = list(levels =
      list("C_2" = 1)))[[1]]$Sum
> SS_psi_3=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_3))),split = list(levels =
      list("C_3" = 1)))[[1]]$Sum
> SS_psi=data.frame(rbind(SS_psi_1,SS_psi_2,SS_psi_3))
> MS_psi_1=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_1))),split = list(levels =
      list("C_1" = 1)))[[1]]$Mean
> MS_psi_2=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_2))),split = list(levels =
      list("C_2" = 1)))[[1]]$Mean
> MS_psi_3=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_3))),split = list(levels =
      list("C_3" = 1)))[[1]]$Mean
> MS_psi=data.frame(rbind(MS_psi_1,MS_psi_2,MS_psi_3))
> F_psi_1=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_1))),split = list(levels =
      list("C_1" = 1)))[[1]]$F
> F_psi_2=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_2))),split = list(levels =
      list("C_2" = 1)))[[1]]$F
> F_psi_3=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_3))),split = list(levels =
      list("C_3" = 1)))[[1]]$F
> F_psi=data.frame(rbind(F_psi_1,F_psi_2,F_psi_3))
> Pr_psi_1=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_1))),split = list(levels =
      list("C_1" = 1)))[[1]]$Pr
> Pr_psi_2=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_2))),split = list(levels =
      list("C_2" = 1)))[[1]]$Pr
> Pr_psi_3=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_3))),split = list(levels =
      list("C_3" = 1)))[[1]]$Pr
> Pr_psi=data.frame(rbind(Pr_psi_1,Pr_psi_2,Pr_psi_3))
> Contrast_names=c("Psi_1","Psi_2","Psi_3")
> Cont_mat=rbind("Psi_1"=C_1,"Psi_2"=C_2,"Psi_3"=C_3)
> Contrasts=data.frame(G1=Cont_mat[,1],G2=Cont_mat[,2],
      G3=Cont_mat[,3],G4=Cont_mat[,4])
> Contrast_Summary=data.frame(Contrast=Contrast_names,DF=
      Df_psi[,2],Contrast_SS=SS_psi[,2],Mean_Square=MS_psi[,2],
      F_Value=F_psi[,2],Pr=Pr_psi[,2])
> Contrast=c("Psi_1","Psi_2","Psi_3")
> Contrast_Summary=data.frame(Contrast=Contrast,DF=Df_psi[,2],
      Contrast_SS=SS_psi[,2],Mean_Square=MS_psi[,2],F_Value=
      F_psi[,2],Pr=Pr_psi[,2])

> # Now we compute the semi-partial coefficients
```

```
> # Make sure to add the PATH to the location where the plot is to be saved
> pdf('/R_scripts/08_Planned_Non_Ortho_Cont/semi_part_corr.pdf')
> semi_part=plot(anova(multi_reg1),what='partial R2')
> dev.off()

      null device
                1

> # Now we print the data and all the results

> print(data)

      ------------------------
         Score           Group
      ------------------------
      1      5  a1_Cont_before
      2      9  a1_Cont_before
      3      8  a1_Cont_before
      4      4  a1_Cont_before
      5      9  a1_Cont_before
      6      5  a2_Part_cont
      7      4  a2_Part_cont
      8      3  a2_Part_cont
      9      5  a2_Part_cont
      10     4  a2_Part_cont
      11     2  a3_Cont_after
      12     4  a3_Cont_after
      13     5  a3_Cont_after
      14     4  a3_Cont_after
      15     1  a3_Cont_after
      16     3  a4_No_cont
      17     3  a4_No_cont
      18     2  a4_No_cont
      19     4  a4_No_cont
      20     3  a4_No_cont
      ------------------------

> print(data_reg)

      -----------------------------
         score  Psi_1  Psi_2  Psi_3
      -----------------------------
      1      5      1      1      0
      2      9      1      1      0
      3      8      1      1      0
      4      4      1      1      0
      5      9      1      1      0
      6      5      1     -1      0
      7      4      1     -1      0
      8      3      1     -1      0
      9      5      1     -1      0
      10     4      1     -1      0
      11     2     -1      0      1
      12     4     -1      0      1
```

```
   13     5     -1      0      1
   14     4     -1      0      1
   15     1     -1      0      1
   16     3     -1      0     -1
   17     3     -1      0     -1
   18     2     -1      0     -1
   19     4     -1      0     -1
   20     3     -1      0     -1
   -----------------------------
```

```
> summary.aov(aov1)


       ----------------------------------------------
                  Df Sum Sq Mean Sq F value   Pr(>F)
       ----------------------------------------------
       Levels       3 50.950  16.983   7.227 0.002782 **
       Residuals   16 37.600   2.350
       ----------------------------------------------
       ---
       Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> print(Contrast_Summary)

       -------------------------------------------------------------
         Contrast DF Contrast_SS Mean_Square     F_Value          Pr
       -------------------------------------------------------------
       1    Psi_1  1       31.25       31.25 13.29787234 0.002174163
       2    Psi_2  1       19.60       19.60  8.34042553 0.010704671
       3    Psi_3  1        0.10        0.10  0.04255319 0.839170948
       -------------------------------------------------------------

> print(multi_reg1)

       Linear Regression Model
       ols(formula = score ~ psi_1 + psi_2 + psi_3)

       ------------------------------------------------------
                n Model L.R.       d.f.         R2      Sigma
       ------------------------------------------------------
               20      17.13          3     0.5754      1.533
       ------------------------------------------------------


       Residuals:
       ------------------------------------------------------
              Min         1Q     Median         3Q        Max
       ------------------------------------------------------
       -3.000e+00 -1.050e+00  9.541e-18  8.500e-01  2.000e+00
       ------------------------------------------------------
```

```
Coefficients:
--------------------------------------------
           Value Std. Error       t  Pr(>|t|)
--------------------------------------------
Intercept  4.35     0.3428 12.6903 9.112e-10
Psi_1      1.25     0.3428  3.6466 2.174e-03
Psi_2      1.40     0.4848  2.8880 1.070e-02
Psi_3      0.10     0.4848  0.2063 8.392e-01
--------------------------------------------

Residual standard error: 1.533 on 16 degrees of freedom
Adjusted R-Squared: 0.4958
```

```
> print(semi_part)

      -----------------------------------
            Psi_1       Psi_2       Psi_3
      -----------------------------------
      0.352907962 0.221343874 0.001129305
      -----------------------------------
```

8.4 Multiple Regression and Non-orthogonal Contrasts

Most of the time, when experimentalists are concerned with *a priori* non-orthogonal comparisons, each comparison represents a prediction from a given theory. The goal of the experiment is, in general, to decide which one (or which ones) of the theories can explain the data best. In other words, the experiment is designed to eliminate some theories by showing that they cannot predict what the other theory (or theories) can predict. Therefore experimenters are interested in what each theory can *specifically* explain. In other words, when dealing with *a priori* non-orthogonal comparisons what the experimenter wants to evaluate are *semi-partial coefficients of correlation* because they express the *specific* effect of a variable. Within this framework, the multiple regression approach for non-orthogonal predictors fits naturally. The main idea, when analyzing non-orthogonal contrasts is simply to consider each contrast as an independent variable in a non-orthogonal multiple regression analyzing the dependent variable.

Suppose (for the beauty of the argument) that the "Romeo and Juliet" experiment was, in fact, designed to test three theories. Each of these theories is expressed as a contrast.

1. *Bransford's* theory implies that only the subjects from the context before group should be able to integrate the story with their long term knowledge. Therefore this group should do better than all the other

groups, which should perform equivalently. This is equivalent to the following contrast:

$$\psi_1 = 3 \times \mu_1 \quad - 1 \times \mu_2 \quad - 1 \times \mu_3 \quad - 1 \times \mu_4$$

2. the *imagery* theory would predict (at least at the time the experiment was designed) that any concrete context presented *during* learning will improve learning. Therefore groups 1 and 2 should do better than the other groups. This is equivalent to the following contrast:

$$\psi_2 = 1 \times \mu_1 \quad 1 \times \mu_2 \quad - 1 \times \mu_3 \quad - 1 \times \mu_4$$

3. The *retrieval cue* theory would predict that the context acts during the retrieval phase (as opposed to Bransford's theory which states that the context acts during the *encoding* phase). Therefore group 1 and 3 should do better than the other groups. This is equivalent to the following contrast:

$$\psi_3 = 1 \times \mu_1 \quad - 1 \times \mu_2 \quad 1 \times \mu_3 \quad - 1 \times \mu_4$$

Contrast	Groups			
	1	2	3	4
ψ_1	3	−1	−1	−1
ψ_2	1	1	−1	−1
ψ_3	1	−1	1	−1

TABLE 8.3 A set of non-orthogonal contrasts for analyzing "Romeo and Juliet." The first contrast corresponds to Bransford's theory. The second contrast corresponds to the imagery theory. The third contrast corresponds to the retrieval cue theory.

8.4.1 [R] code

```
# ANOVA One-factor between subjects S(A)
# Romeo and Juliet: Non-Orthogonal Multiple Regression


# Install and load package "Design" and "gregmisc".
# NOTE 1: Arrange your levels in alphabetical order, else R
#   will do it for you

# We have 1 Factor, A, with 4 levels:
# Context Before,
# Partial Context,
# Context After,
```

```
# No Context

# We have 5 subjects per group. Therefore 5 x 4 = 20 subjects
#   total.

# We collect the data for each level of Factor A
a1_Cont_before=c(5,9,8,4,9)
a2_Part_cont=c(5,4,3,5,4)
a3_Cont_after=c(2,4,5,4,1)
a4_No_cont=c(3,3,2,4,3)

# We now combine the observations into one long column (score).
score=c(a1_Cont_before,a2_Part_cont, a3_Cont_after, a4_No_cont)

# We generate a second column (levels), that identifies the group
#   for each score.
levels=factor(c(rep("a1_Cont_before",5),rep("a2_Part_cont",5),
       rep("a3_Cont_after",5),rep("a4_No_cont",5)))

# We now form a data frame with the dependent variable and
#   the factors.
data=data.frame(score=score,group=levels)

# We now define the non-orthogonal contrasts
C_1=c(3,-1,-1,-1)
C_2=c(1,1,-1,-1)
C_3=c(1,-1,1,-1)

# We now perfom an S(A) anova on the data
aov1=aov(score~levels)

# We create a model matrix and include the contrasts as separate
#   exploratory variables.
cont_coeff=cbind(C_1,C_2,C_3)
model_matrix=model.matrix(~C(levels,cont_coeff,base=1))
data_reg=data.frame(score,psi_1=model_matrix[,2],psi_2=
       model_matrix[,3],psi_3=model_matrix[,4])
psi_1=model_matrix[,2]
psi_2=model_matrix[,3]
psi_3=model_matrix[,4]

# Now we perform an orthogonal multiple regression analysis on
#   the data
multi_reg1=ols(score~psi_1+psi_2+psi_3)

# We now organize the data
Df_psi_1=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_1))),split = list(levels =
       list("C_1" = 1)))[[1]]$Df
Df_psi_2=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_2))),split = list(levels =
       list("C_2" = 1)))[[1]]$Df
Df_psi_3=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_3))),split = list(levels =
```

```
       list("C_3" = 1)))[[1]]$Df
Df_psi=data.frame(rbind(Df_psi_1,Df_psi_2,Df_psi_3))
SS_psi_1=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_1))),split = list(levels =
       list("C_1" = 1)))[[1]]$Sum
SS_psi_2=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_2))),split = list(levels =
       list("C_2" = 1)))[[1]]$Sum
SS_psi_3=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_3))),split = list(levels = l
       ist("C_3" = 1)))[[1]]$Sum
SS_psi=data.frame(rbind(SS_psi_1,SS_psi_2,SS_psi_3))
MS_psi_1=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_1))),split = list(levels =
       list("C_1" = 1)))[[1]]$Mean
MS_psi_2=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_2))),split = list(levels =
       list("C_2" = 1)))[[1]]$Mean
MS_psi_3=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_3))),split = list(levels =
       list("C_3" = 1)))[[1]]$Mean
MS_psi=data.frame(rbind(MS_psi_1,MS_psi_2,MS_psi_3))
F_psi_1=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_1))),split = list(levels =
       list("C_1" = 1)))[[1]]$F
F_psi_2=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_2))),split = list(levels =
       list("C_2" = 1)))[[1]]$F
F_psi_3=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_3))),split = list(levels =
       list("C_3" = 1)))[[1]]$F
F_psi=data.frame(rbind(F_psi_1,F_psi_2,F_psi_3))
Pr_psi_1=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_1))),split = list(levels =
       list("C_1" = 1)))[[1]]$Pr
Pr_psi_2=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_2))),split = list(levels =
       list("C_2" = 1)))[[1]]$Pr
Pr_psi_3=summary(aov(score~levels,contrasts=list(levels=
       make.contrasts(C_3))),split = list(levels =
       list("C_3" = 1)))[[1]]$Pr
Pr_psi=data.frame(rbind(Pr_psi_1,Pr_psi_2,Pr_psi_3))
Contrast_names=c("Psi_1","Psi_2","Psi_3")
Cont_mat=rbind("Psi_1"=C_1,"Psi_2"=C_2,"Psi_3"=C_3)
Contrasts=data.frame(G1=Cont_mat[,1],G2=Cont_mat[,2],
       G3=Cont_mat[,3],G4=Cont_mat[,4])
Contrast_Summary=data.frame(Contrast=Contrast_names,DF=
       Df_psi[,2],Contrast_SS=SS_psi[,2],Mean_Square=MS_psi[,2],
       F_Value=F_psi[,2],Pr=Pr_psi[,2])
Contrast=c("Psi_1","Psi_2","Psi_3")
Contrast_Summary=data.frame(Contrast=Contrast,DF=Df_psi[,2],
       Contrast_SS=SS_psi[,2],Mean_Square=MS_psi[,2],F_Value=
       F_psi[,2],Pr=Pr_psi[,2])
```

```
# Now we compute the semi-partial coefficients
# Make sure to add the PATH to the location where the plot is to
#   be saved
pdf('/R_scripts/08_Planned_Non_Ortho_Cont/
       semi_part_corr.pdf')
semi_part=plot(anova(multi_reg1),what='partial R2')
dev.off()

# Now we print the data and all the results
print(data)
print(data_reg)
summary.aov(aov1)
print(Contrast_Summary)
print(multi_reg1)
print(semi_part)
```

8.4.2 [R] output

```
> # ANOVA One-factor between subjects S(A)
> # Romeo and Juliet: Non-Orthogonal Multiple Regression


> # Install and load package "Design" and "gregmisc".
> # NOTE 1: Arrange your levels in alphabetical order, else R
> #   will do it for you

> # We have 1 Factor, A, with 4 levels:
> # Context Before,
> # Partial Context,
> # Context After,
> # No Context

> # We have 5 subjects per group. Therefore 5 x 4 = 20 subjects
> #   total.

> # We collect the data for each level of Factor A
> a1_Cont_before=c(5,9,8,4,9)
> a2_Part_cont=c(5,4,3,5,4)
> a3_Cont_after=c(2,4,5,4,1)
> a4_No_cont=c(3,3,2,4,3)

> # We now combine the observations into one long column (score).
> score=c(a1_Cont_before,a2_Part_cont, a3_Cont_after, a4_No_cont)

> # We generate a second column (levels), that identifies the group
> #   for each score.
> levels=factor(c(rep("a1_Cont_before",5),rep("a2_Part_cont",5),
       rep("a3_Cont_after",5),rep("a4_No_cont",5)))

> # We now form a data frame with the dependent variable and
> #   the factors.
> data=data.frame(score=score,group=levels)
```

```
> # We now define the non-orthogonal contrasts
> C_1=c(3,-1,-1,-1)
> C_2=c(1,1,-1,-1)
> C_3=c(1,-1,1,-1)

> # We now perfom an S(A) anova on the data
> aov1=aov(score~levels)

> # We create a model matrix and include the contrasts as separate
> #   exploratory variables.
> cont_coeff=cbind(C_1,C_2,C_3)
> model_matrix=model.matrix(~C(levels,cont_coeff,base=1))
> data_reg=data.frame(score,psi_1=model_matrix[,2],psi_2=
      model_matrix[,3],psi_3=model_matrix[,4])
> psi_1=model_matrix[,2]
> psi_2=model_matrix[,3]
> psi_3=model_matrix[,4]

> # Now we perform an orthogonal multiple regression analysis on
> #   the data
> multi_reg1=ols(score~psi_1+psi_2+psi_3)

> # We now organize the data
> Df_psi_1=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_1))),split = list(levels =
      list("C_1" = 1)))[[1]]$Df
> Df_psi_2=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_2))),split = list(levels =
      list("C_2" = 1)))[[1]]$Df
> Df_psi_3=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_3))),split = list(levels =
      list("C_3" = 1)))[[1]]$Df
> Df_psi=data.frame(rbind(Df_psi_1,Df_psi_2,Df_psi_3))
> SS_psi_1=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_1))),split = list(levels =
      list("C_1" = 1)))[[1]]$Sum
> SS_psi_2=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_2))),split = list(levels =
      list("C_2" = 1)))[[1]]$Sum
> SS_psi_3=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_3))),split = list(levels = l
      ist("C_3" = 1)))[[1]]$Sum
> SS_psi=data.frame(rbind(SS_psi_1,SS_psi_2,SS_psi_3))
> MS_psi_1=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_1))),split = list(levels =
      list("C_1" = 1)))[[1]]$Mean
> MS_psi_2=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_2))),split = list(levels =
      list("C_2" = 1)))[[1]]$Mean
> MS_psi_3=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_3))),split = list(levels =
      list("C_3" = 1)))[[1]]$Mean
> MS_psi=data.frame(rbind(MS_psi_1,MS_psi_2,MS_psi_3))
> F_psi_1=summary(aov(score~levels,contrasts=list(levels=
```

```
      make.contrasts(C_1))),split = list(levels =
      list("C_1" = 1)))[[1]]$F
> F_psi_2=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_2))),split = list(levels =
      list("C_2" = 1)))[[1]]$F
> F_psi_3=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_3))),split = list(levels =
      list("C_3" = 1)))[[1]]$F
> F_psi=data.frame(rbind(F_psi_1,F_psi_2,F_psi_3))
> Pr_psi_1=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_1))),split = list(levels =
      list("C_1" = 1)))[[1]]$Pr
> Pr_psi_2=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_2))),split = list(levels =
      list("C_2" = 1)))[[1]]$Pr
> Pr_psi_3=summary(aov(score~levels,contrasts=list(levels=
      make.contrasts(C_3))),split = list(levels =
      list("C_3" = 1)))[[1]]$Pr
> Pr_psi=data.frame(rbind(Pr_psi_1,Pr_psi_2,Pr_psi_3))
> Contrast_names=c("Psi_1","Psi_2","Psi_3")
> Cont_mat=rbind("Psi_1"=C_1,"Psi_2"=C_2,"Psi_3"=C_3)
> Contrasts=data.frame(G1=Cont_mat[,1],G2=Cont_mat[,2],
      G3=Cont_mat[,3],G4=Cont_mat[,4])
> Contrast_Summary=data.frame(Contrast=Contrast_names,DF=
      Df_psi[,2],Contrast_SS=SS_psi[,2],Mean_Square=MS_psi[,2],
      F_Value=F_psi[,2],Pr=Pr_psi[,2])
> Contrast=c("Psi_1","Psi_2","Psi_3")
> Contrast_Summary=data.frame(Contrast=Contrast,DF=Df_psi[,2],
      Contrast_SS=SS_psi[,2],Mean_Square=MS_psi[,2],F_Value=
      F_psi[,2],Pr=Pr_psi[,2])

> # Now we compute the semi-partial coefficients
> # Make sure to add the PATH to the location where the plot is to
> #   be saved
> pdf('?/R_scripts/08_Planned_Non_Ortho_Cont/
      semi_part_corr.pdf')
> semi_part=plot(anova(multi_reg1),what='partial R2')
> dev.off()

      null device
                1

> # Now we print the data and all the results

> print(data)
      ------------------------
         Score          Group
      ------------------------
      1      5  a1_Cont_before
      2      9  a1_Cont_before
      3      8  a1_Cont_before
      4      4  a1_Cont_before
      5      9  a1_Cont_before
      6      5  a2_Part_cont
```

```
7      4  a2_Part_cont
8      3  a2_Part_cont
9      5  a2_Part_cont
10     4  a2_Part_cont
11     2  a3_Cont_after
12     4  a3_Cont_after
13     5  a3_Cont_after
14     4  a3_Cont_after
15     1  a3_Cont_after
16     3  a4_No_cont
17     3  a4_No_cont
18     2  a4_No_cont
19     4  a4_No_cont
20     3  a4_No_cont
------------------------
```

```
> print(data_reg)

        -----------------------------
           Score  Psi_1  Psi_2  Psi_3
        -----------------------------
        1      5      3      1      1
        2      9      3      1      1
        3      8      3      1      1
        4      4      3      1      1
        5      9      3      1      1
        6      5     -1      1     -1
        7      4     -1      1     -1
        8      3     -1      1     -1
        9      5     -1      1     -1
        10     4     -1      1     -1
        11     2     -1     -1      1
        12     4     -1     -1      1
        13     5     -1     -1      1
        14     4     -1     -1      1
        15     1     -1     -1      1
        16     3     -1     -1     -1
        17     3     -1     -1     -1
        18     2     -1     -1     -1
        19     4     -1     -1     -1
        20     3     -1     -1     -1
        -----------------------------

> summary.aov(aov1)

        --------------------------------------------------
                    Df  Sum Sq  Mean Sq  F value    Pr(>F)
        --------------------------------------------------
        Levels       3  50.950   16.983    7.227  0.002782 **
        Residuals   16  37.600    2.350
        --------------------------------------------------
        ---
        Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
> print(Contrast_Summary)

        -----------------------------------------------------------
          Contrast DF Contrast_SS Mean_Square   F_Value           Pr
        -----------------------------------------------------------
        1    Psi_1  1    46.81667    46.81667 19.921986 0.0003921401
        2    Psi_2  1    31.25000    31.25000 13.297872 0.0021741633
        3    Psi_3  1    11.25000    11.25000  4.787234 0.0438564148
        -----------------------------------------------------------

> print(multi_reg1)

        Linear Regression Model
        ols(formula = score ~ psi_1 + psi_2 + psi_3)
        -----------------------------------------------------
                 n Model L.R.       d.f.         R2      Sigma
        -----------------------------------------------------
                20      17.13          3     0.5754      1.533
        -----------------------------------------------------

        Residuals:
        -------------------------------------------------------
               Min          1Q      Median         3Q        Max
        -------------------------------------------------------
        -3.000e+00  -1.050e+00  -2.797e-16  8.500e-01  2.000e+00
        -------------------------------------------------------

        Coefficients:
        -----------------------------------------------
                  Value  Std. Error        t   Pr(>|t|)
        -----------------------------------------------
        Intercept  4.35      0.3428  12.6903  9.112e-10
        psi_1      0.65      0.3428   1.8962  7.613e-02
        psi_2      0.60      0.4848   1.2377  2.337e-01
        psi_3      0.10      0.4848   0.2063  8.392e-01
        -----------------------------------------------
               Residual standard error: 1.533 on 16 degrees of freedom
        Adjusted R-Squared: 0.4958

> print(semi_part)

        -------------------------------------
              Psi_1        Psi_2        Psi_3
        -------------------------------------
        0.095426313  0.040654997  0.001129305
        -------------------------------------
```

9 Post hoc or *a-posteriori* analyses

Post hoc analyses are performed after the data have been collected, or in other words, after the fact. When looking at the results, you may find an unexpected pattern. If that pattern of results suggests some interesting hypothesis, then you want to be sure that it is not a fluke. This is the aim of *post hoc* (also called *a posteriori*) comparisons.

The main problem with *post hoc* comparisons involves the size of the family of possible comparisons. The number of possible comparisons, grows very quickly as a function of A (the number of levels of the independent variable), making the use of procedures such as Šidàk or Bonferonni, Boole, Dunn inequalities unrealistic.

Two main approaches will be examined:

- Evaluating all the possible contrasts; this is known as *Scheffé*'s test.
- The specific problem of pairwise comparisons. Here we will see three different tests: *Tukey, Newman-Keuls,* and *Duncan.*

Note that by default, SAS evaluates the contrasts with the α level set at $.05$. If a lower α is desired, this must be specified by following the post hoc option name with ALPHA=.01. For example, to specify an alpha level of .01 for a Scheffé's test, you would give the following command:

```
MEANS GROUP / SCHEFFE ALPHA=.01.
```

9.1 Scheffé's test

Scheffé's test was devised in order to be able to test all the possible contrasts *a posteriori* while maintaining the overall Type I error for the family at a reasonable level, as well as trying to have a relatively powerful test. Specifically, the Scheffé test is a conservative test. The critical value for the Scheffé test is larger than the critical value for other, more powerful, tests. In every case where the Scheffé test rejects the null hypothesis, more powerful tests also reject the null hypothesis.

9.1.1 Romeo and Juliet

We will use, once again, Bransford *et al.*'s "Romeo and Juliet" experiment. The following Table gives the different experimental conditions:

Context Before	Partial Context	Context After	Without Context

The "error mean square" is $MS_{S(A)} = 2.35$; and $S = 5$. Here are the values of the experimental means (note that the means have been reordered from the largest to the smallest):

	Context Before	Partial Context	Context After	Without Context
$M_{a.}$	7.00	4.20	3.20	3.00

Suppose now that the experimenters wanted to test the following contrasts *after* having collected the data.

	Context Before	Partial Context	Context After	Without Context
ψ_1	1	1	1	-3
ψ_2	0	0	1	-1
ψ_3	3	-1	-1	-1
ψ_4	1	-1	0	0

The critical value for $\alpha[PF] = .05$ is given by:

$$F_{\text{critical, Scheffé}} = (A-1)F_{\text{critical,omnibus}} = (4-1) \times 3.24 = 9.72$$

with $\nu_1 = A - 1 = 3$ and $\nu_2 = A(S-1) = 16$.

9.1.2 [R] code

```
# Romeo and Juliet: Post Hoc Comparisons - Scheffe's Test
# ANOVA One-factor between subjects S(A)

# NOTE 1: Install package 'gregmisc' in order to use
#    make.contrasts
# NOTE 2: make.contrasts will augment an incomplete set of
#    orthogonal contrasts with "filler" contrasts
# NOTE 3: Arrange your levels in alphabetical order, else R
#    will do it for you

# We have 1 Factor, A, with 4 levels: Context Before,
```

```
#    Partial Context, Context After,No Context
# We have 5 subjects per group. Therefore 5 x 4 = 20
#   subjects total.

# We collect the data for each level of Factor A
a1_Cont_before=c(5,9,8,4,9)
a2_Part_cont=c(5,4,3,5,4)
a3_Cont_after=c(2,4,5,4,1)
a4_No_cont=c(3,3,2,4,3)

# We now combine the observations into one long column (score).
score=c(a1_Cont_before,a2_Part_cont, a3_Cont_after, a4_No_cont)
# We generate a second column (levels), that identifies the
#    group for each score.
levels=factor(c(rep("a1_Cont_before",5),rep("a2_Part_cont",5),
rep("a3_Cont_after",5),rep("a4_No_cont",5)))

# We now form a data frame with the dependent variable and
#   the factors.
data=data.frame(score=score,group=levels)

# We now define the post hoc contrasts
C_1=c(1,1,1,-3)
C_2=c(0,0,1,-1)
C_3=c(3,-1,-1,-1)
C_4=c(1,-1,0,0)

# We now perform the test for multiple comparisons using
#   "Scheffe" correction.
# The means with different letters are significantly different.
# NOTE: The source for R script "mulcomp" has to be specified.
means=tapply(score,levels,mean)
source("/Desktop/R_scripts/09_Post_Hoc_Comp/mulcomp.R")
multi_comp=mulcomp(as.vector(means),5,16,2.350,
conf.level=.05,type= "Scheffe",decreasing=TRUE)

# We now perform an ANOVA on the data
aov5=aov(score~levels)

# We now organize the results
Df_psi_1=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_1))),split = list(levels = list("C_1"
    = 1)))[[1]]$Df
Df_psi_2=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_2))),split = list(levels = list("C_2"
    = 1)))[[1]]$Df
Df_psi_3=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_3))),split = list(levels = list("C_3"
    = 1)))[[1]]$Df
Df_psi_4=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_4))),split = list(levels = list("C_4"
    = 1)))[[1]]$Df
Df_psi=data.frame(rbind(Df_psi_1,Df_psi_2,Df_psi_3,Df_psi_4))
```

```
SS_psi_1=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_1))),split = list(levels = list("C_1"
    = 1)))[[1]]$Sum
SS_psi_2=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_2))),split = list(levels = list("C_2"
    = 1)))[[1]]$Sum
SS_psi_3=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_3))),split = list(levels = list("C_3"
    = 1)))[[1]]$Sum
SS_psi_4=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_4))),split = list(levels = list("C_4"
    = 1)))[[1]]$Sum
SS_psi=data.frame(rbind(SS_psi_1,SS_psi_2,SS_psi_3,SS_psi_4))

MS_psi_1=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_1))),split = list(levels = list("C_1"
    = 1)))[[1]]$Mean
MS_psi_2=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_2))),split = list(levels = list("C_2"
    = 1)))[[1]]$Mean
MS_psi_3=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_3))),split = list(levels = list("C_3"
    = 1)))[[1]]$Mean
MS_psi_4=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_4))),split = list(levels = list("C_4"
    = 1)))[[1]]$Mean
MS_psi=data.frame(rbind(MS_psi_1,MS_psi_2,MS_psi_3,MS_psi_4))

F_psi_1=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_1))),split = list(levels = list("C_1"
    = 1)))[[1]]$F
F_psi_2=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_2))),split = list(levels = list("C_2"
    = 1)))[[1]]$F
F_psi_3=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_3))),split = list(levels = list("C_3"
    = 1)))[[1]]$F
F_psi_4=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_4))),split = list(levels = list("C_4"
    = 1)))[[1]]$F
F_psi=data.frame(rbind(F_psi_1,F_psi_2,F_psi_3,F_psi_4))


F_scheffe=F_psi[,2]/3
Pr_scheffe=1-pf(F_scheffe,Df_psi[,1],Df_psi[,3])

Pr_psi_1=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_1))),split = list(levels = list("C_1"
    = 1)))[[1]]$Pr
Pr_psi_2=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_2))),split = list(levels = list("C_2"
    = 1)))[[1]]$Pr
Pr_psi_3=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_3))),split = list(levels = list("C_3"
```

```
    = 1)))[[1]]$Pr
Pr_psi_4=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_4))),split = list(levels = list("C_4"
    = 1)))[[1]]$Pr
Pr_psi=data.frame(rbind(Pr_psi_1,Pr_psi_2,Pr_psi_3,Pr_psi_4))

Contrast_names=c("Psi_1","Psi_2","Psi_3","Psi_4")
Cont_mat=rbind("Psi_1"=C_1,"Psi_2"=C_2,"Psi_3"=C_3,"Psi_4"=C_4)
Contrasts=data.frame(G1=Cont_mat[,1], G2=Cont_mat[,2],
    G3=Cont_mat[,3], G4=Cont_mat[,4])
Contrast_Summary=data.frame(Contrast= Contrast_names,
    DF=Df_psi[,2], Contrast_SS=SS_psi[,2],
    Mean_Square=MS_psi[,2], F_Value=F_psi[,2],Pr=Pr_psi[,2],
    F_scheffe=F_scheffe, Pr_scheffe=Pr_scheffe)

# We now print the data and all the results
print(data)
print(multi_comp)
print('Means with the same letter are not significant')
summary(aov5)
print(Contrasts)
print(Contrast_Summary)
```

9.1.3 [R] output

```
> # Romeo and Juliet: Post Hoc Comparisons - Scheffe's Test
> # ANOVA One-factor between subjects S(A)

> # NOTE 1: Install package 'gregmisc' in order to use
> #    make.contrasts
> # NOTE 2: make.contrasts will augment an incomplete set of
> #    orthogonal contrasts with "filler" contrasts
> # NOTE 3: Arrange your levels in alphabetical order, else R
> #    will do it for you

> # We have 1 Factor, A, with 4 levels: Context Before,
> #    Partial Context, Context After,No Context
> # We have 5 subjects per group. Therefore 5 x 4 = 20
> #   subjects total.

> # We collect the data for each level of Factor A
> a1_Cont_before=c(5,9,8,4,9)
> a2_Part_cont=c(5,4,3,5,4)
> a3_Cont_after=c(2,4,5,4,1)
> a4_No_cont=c(3,3,2,4,3)

> # We now combine the observations into one long column (score).
> score=c(a1_Cont_before,a2_Part_cont, a3_Cont_after, a4_No_cont)
> # We generate a second column (levels), that identifies the
> #    group for each score.
> levels=factor(c(rep("a1_Cont_before",5),rep("a2_Part_cont",5),
rep("a3_Cont_after",5),rep("a4_No_cont",5)))
```

```
> # We now form a data frame with the dependent variable and
> #   the factors.
> data=data.frame(score=score,group=levels)

> # We now define the post hoc contrasts
> C_1=c(1,1,1,-3)
> C_2=c(0,0,1,-1)
> C_3=c(3,-1,-1,-1)
> C_4=c(1,-1,0,0)

> # We now perform the test for multiple comparisons using
> #   "Scheffe" correction.
> # The means with different letters are significantly different.
> # NOTE: The source for R script "mulcomp" has to be specified.
> means=tapply(score,levels,mean)
> source("/Desktop/R_scripts/09_Post_Hoc_Comp/mulcomp.R")
> multi_comp=mulcomp(as.vector(means),5,16,2.350,
conf.level=.05,type= "Scheffe",decreasing=TRUE)

> # We now perform an ANOVA on the data
> aov5=aov(score~levels)

> # We now organize the results
> Df_psi_1=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_1))),split = list(levels = list("C_1"
>     = 1)))[[1]]$Df
> Df_psi_2=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_2))),split = list(levels = list("C_2"
>     = 1)))[[1]]$Df
> Df_psi_3=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_3))),split = list(levels = list("C_3"
>     = 1)))[[1]]$Df
> Df_psi_4=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_4))),split = list(levels = list("C_4"
>     = 1)))[[1]]$Df
> Df_psi=data.frame(rbind(Df_psi_1,Df_psi_2,Df_psi_3,Df_psi_4))

> SS_psi_1=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_1))),split = list(levels = list("C_1"
>     = 1)))[[1]]$Sum
> SS_psi_2=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_2))),split = list(levels = list("C_2"
>     = 1)))[[1]]$Sum
> SS_psi_3=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_3))),split = list(levels = list("C_3"
>     = 1)))[[1]]$Sum
> SS_psi_4=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_4))),split = list(levels = list("C_4"
>     = 1)))[[1]]$Sum
> SS_psi=data.frame(rbind(SS_psi_1,SS_psi_2,SS_psi_3,SS_psi_4))

> MS_psi_1=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_1))),split = list(levels = list("C_1"
```

```
>     = 1)))[[1]]$Mean
> MS_psi_2=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_2))),split = list(levels = list("C_2"
>     = 1)))[[1]]$Mean
> MS_psi_3=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_3))),split = list(levels = list("C_3"
>     = 1)))[[1]]$Mean
> MS_psi_4=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_4))),split = list(levels = list("C_4"
>     = 1)))[[1]]$Mean
> MS_psi=data.frame(rbind(MS_psi_1,MS_psi_2,MS_psi_3,MS_psi_4))

> F_psi_1=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_1))),split = list(levels = list("C_1"
>     = 1)))[[1]]$F
> F_psi_2=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_2))),split = list(levels = list("C_2"
>     = 1)))[[1]]$F
> F_psi_3=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_3))),split = list(levels = list("C_3"
>     = 1)))[[1]]$F
> F_psi_4=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_4))),split = list(levels = list("C_4"
>     = 1)))[[1]]$F
> F_psi=data.frame(rbind(F_psi_1,F_psi_2,F_psi_3,F_psi_4))


> F_scheffe=F_psi[,2]/3
> Pr_scheffe=1-pf(F_scheffe,Df_psi[,1],Df_psi[,3])

> Pr_psi_1=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_1))),split = list(levels = list("C_1"
>     = 1)))[[1]]$Pr
> Pr_psi_2=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_2))),split = list(levels = list("C_2"
>     = 1)))[[1]]$Pr
> Pr_psi_3=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_3))),split = list(levels = list("C_3"
>     = 1)))[[1]]$Pr
> Pr_psi_4=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_4))),split = list(levels = list("C_4"
>     = 1)))[[1]]$Pr
> Pr_psi=data.frame(rbind(Pr_psi_1,Pr_psi_2,Pr_psi_3,Pr_psi_4))

> Contrast_names=c("Psi_1","Psi_2","Psi_3","Psi_4")
> Cont_mat=rbind("Psi_1"=C_1,"Psi_2"=C_2,"Psi_3"=C_3,"Psi_4"=C_4)
> Contrasts=data.frame(G1=Cont_mat[,1], G2=Cont_mat[,2],
>     G3=Cont_mat[,3], G4=Cont_mat[,4])
> Contrast_Summary=data.frame(Contrast= Contrast_names,
>     DF=Df_psi[,2], Contrast_SS=SS_psi[,2],
>     Mean_Square=MS_psi[,2], F_Value=F_psi[,2],Pr=Pr_psi[,2],
>     F_scheffe=F_scheffe, Pr_scheffe=Pr_scheffe)
```

```
> # We now print the data and all the results

> print(data)

       -----------------------
          Score         Group
       -----------------------
       1      5 a1_Cont_before
       2      9 a1_Cont_before
       3      8 a1_Cont_before
       4      4 a1_Cont_before
       5      9 a1_Cont_before
       6      5   a2_Part_cont
       7      4   a2_Part_cont
       8      3   a2_Part_cont
       9      5   a2_Part_cont
       10     4   a2_Part_cont
       11     2  a3_Cont_after
       12     4  a3_Cont_after
       13     5  a3_Cont_after
       14     4  a3_Cont_after
       15     1  a3_Cont_after
       16     3     a4_No_cont
       17     3     a4_No_cont
       18     2     a4_No_cont
       19     4     a4_No_cont
       20     3     a4_No_cont
       -----------------------

> print(multi_comp)

       $sem
       [1] 0.6855655

       $sed
       [1] 0.969536

       $MCT
       -------------
          medie test
       -------------
       M1   7.0    a
       M2   4.2   ab
       M3   3.2    b
       M4   3.0    b
       -------------

       $Critical.Differences
       [1] 0.000000 3.022188 3.022188 3.022188


> print('Means with the same letter are not significant')
[1] "Means with the same letter are not significant"
```

```
> summary(aov5)

      ---------------------------------------------
                 Df Sum Sq Mean Sq F value   Pr(>F)
      ---------------------------------------------
      levels       3 50.950  16.983   7.227 0.002782 **
      Residuals   16 37.600   2.350
      ---------------------------------------------
      ---
      Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> print(Contrasts)

      -----------------
            G1 G2 G3 G4
      -----------------
      Psi_1  1  1  1 -3
      Psi_2  0  0  1 -1
      Psi_3  3 -1 -1 -1
      Psi_4  1 -1  0  0
      -----------------

> print(Contrast_Summary)

      -------------------------------------------------
        Contrast DF Contrast_SS Mean_Square     F_Value
      -------------------------------------------------
      1    Psi_1  1    12.15000    12.15000  5.17021277
      2    Psi_2  1     0.10000     0.10000  0.04255319
      3    Psi_3  1    46.81667    46.81667 19.92198582
      4    Psi_4  1    19.60000    19.60000  8.34042553
      -------------------------------------------------

      -------------------------------------------------
        Contrast            Pr   F_scheffe   Pr_scheffe
       -------------------------------------------------
      1    Psi_1  0.0371045233  1.72340426  0.202402780
      2    Psi_2  0.8391709477  0.01418440  0.997594549
      3    Psi_3  0.0003921401  6.64066194  0.004016922
      4    Psi_4  0.0107046712  2.78014184  0.074823141
      -------------------------------------------------
```

The results of the Scheffé procedure for the family can be summarized in the following Table:

Comparison	$SS_{\text{comp.}}$	$F_{\text{comp.}}$	Decision	$\Pr(F_{\text{Scheffé}})$
ψ_1	12.15	5.17	*ns*	.201599
ψ_2	0.10	0.04	*ns*	$F < 1$
ψ_3	46.82	19.92	reject H_0	.004019
ψ_4	19.60	8.34	*ns*	.074800

9.2 Tukey's test

The Tukey test uses a distribution derived by Gosset, who is better known under his pen name of Student (yes, like Student-t!). Gosset/Student derived a distribution called Student's q or the *Studentized range*, which is the value reported by SAS. The value reported in your textbook is a slightly modified version of his distribution called F-range or F_{range}. F-range is derived from q by the transformation:

$$F_{\text{range}} = \frac{q^2}{2} .$$

9.2.1 The return of Romeo and Juliet

For an example, we will use, once again, Bransford *et al.*'s "Romeo and Juliet." Recall that

$$MS_{S(A)} = 2.35; \quad S = 5$$

and that the experimental results were:

	Context Before	Partial Context	Context After	Without Context
$M_{a.}$	7.00	4.20	3.20	3.00

The pairwise difference between means can be given in a Table:

	$M_{1.}$	$M_{2.}$	$M_{3.}$	$M_{4.}$
$M_{1.}$		2.80	3.80	4.00
$M_{2.}$			1.00	1.20
$M_{3.}$				0.20

The values for $F_{\text{critical,Tukey}}$ given by the table are

$$8.20 \quad \text{for} \quad \alpha[PF] = .05$$

$$13.47 \quad \text{for} \quad \alpha[PF] = .01$$

The results of the computation of the different F ratios for the pairwise comparisons are given in the following Table, the sign $*$ indicates a difference significant at the $.05$ level, and $**$ indicates a difference significant at the $.01$ level.

Note in the SAS output, that the "Critical Value of Studentized Range = 4.046". Remember the formula to derive the F_{range} from q

$$F_{\text{range}} = \frac{q^2}{2}\,.$$

For our example,

$$F_{\text{range}} = 8.185 = \frac{4.046^2}{2}\,.$$

The difference observed between the value reported in the book and that obtained using SAS's value is due to rounding errors.

	$M_{1.}$	$M_{2.}$	$M_{3.}$	$M_{4.}$
$M_{1.}$		8.34^{*}	15.36^{**}	17.02^{**}
$M_{2.}$			1.06	1.53
$M_{3.}$				0.40

Tukey test is clearly a conservative test. Several approaches have been devised in order to have a more sensitive test. The most popular alternative (but not the "safest") is the Newman-Keuls test.

9.2.1.1 [R] code

```
# Romeo and Juliet: Post Hoc Comparisons - Tukey's Test
# ANOVA One-factor between subjects S(A)

# NOTE 1: Install package 'gregmisc' in order to use
#    make.contrasts
# NOTE 2: make.contrasts will augment an incomplete set of
#    orthogonal contrasts with "filler" contrasts
# NOTE 3: Arrange your levels in alphabetical order, else R
#    will do it for you


# We have 1 Factor, A, with 4 levels: Context Before,
#    Partial Context, Context After,No Context

# We have 5 subjects per group. Therefore 5 x 4 = 20 subjects
#    total.

# We collect the data for each level of Factor A
a1_Cont_before=c(5,9,8,4,9)
a2_Part_cont=c(5,4,3,5,4)
a3_Cont_after=c(2,4,5,4,1)
a4_No_cont=c(3,3,2,4,3)

# We now combine the observations into one long column (score).
score=c(a1_Cont_before,a2_Part_cont, a3_Cont_after, a4_No_cont)
```

```
# We generate a second column (levels), that identifies the
#    group for each score.

levels=factor(c(rep("a1_Cont_before",5),rep("a2_Part_cont",5),
#    rep("a3_Cont_after",5),rep("a4_No_cont",5)))

# We now form a data frame with the dependent variable and the
#    factors.
data=data.frame(score=score,group=levels)

# We now define the pairwise comparisons
C_1=c(1,-1,0,0)
C_2=c(1,0,-1,0)
C_3=c(1,0,0,-1)
C_4=c(0,1,-1,0)
C_5=c(0,1,0,-1)
C_6=c(0,0,1,-1)

# We now perform the test for multiple comparisons using
#    "Tukey" correction.
# The means with different letters are significantly different.
# NOTE: The source for R script "mulcomp" has to be specified.
means=tapply(score,levels,mean)
source("/home/anjali/Desktop/R_scripts/09_Post_Hoc_Comp/
    mulcomp.R")
multi_comp=mulcomp(as.vector(means),5,16,2.350,conf.level=
    .05,type= "TukeyHSD",decreasing=TRUE)

# We now perfom on ANOVA on the data
aov5=aov(score~levels)

# We now organize the results
Df_psi_1=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_1))),split = list(levels = list("C_1"
    = 1)))[[1]]$Df
Df_psi_2=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_2))),split = list(levels = list("C_2"
    = 1)))[[1]]$Df
Df_psi_3=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_3))),split = list(levels = list("C_3"
    = 1)))[[1]]$Df
Df_psi_4=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_4))),split = list(levels = list("C_4"
    = 1)))[[1]]$Df
Df_psi_5=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_5))),split = list(levels = list("C_5"
    = 1)))[[1]]$Df
Df_psi_6=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_5))),split = list(levels = list("C_6"
    = 1)))[[1]]$Df
Df_psi=data.frame(rbind(Df_psi_1,Df_psi_2,Df_psi_3,Df_psi_4,
    Df_psi_5,Df_psi_6))

SS_psi_1=summary(aov(score~levels,contrasts=list(levels=
```

```
make.contrasts(C_1))),split = list(levels = list("C_1"
= 1)))[[1]]$Sum

SS_psi_2=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_2))),split = list(levels = list("C_2"
    = 1)))[[1]]$Sum
SS_psi_3=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_3))),split = list(levels = list("C_3"
    = 1)))[[1]]$Sum
SS_psi_4=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_4))),split = list(levels = list("C_4"
    = 1)))[[1]]$Sum
SS_psi_5=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_5))),split = list(levels = list("C_5"
    = 1)))[[1]]$Sum
SS_psi_6=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_6))),split = list(levels = list("C_6"
    = 1)))[[1]]$Sum
SS_psi=data.frame(rbind(SS_psi_1,SS_psi_2,SS_psi_3,SS_psi_4,
    SS_psi_5,SS_psi_6))

MS_psi_1=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_1))),split = list(levels = list("C_1"
    = 1)))[[1]]$Mean
MS_psi_2=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_2))),split = list(levels = list("C_2"
    = 1)))[[1]]$Mean
MS_psi_3=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_3))),split = list(levels = list("C_3"
    = 1)))[[1]]$Mean
MS_psi_4=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_4))),split = list(levels = list("C_4"
    = 1)))[[1]]$Mean
MS_psi_5=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_5))),split = list(levels = list("C_5"
    = 1)))[[1]]$Mean
MS_psi_6=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_6))),split = list(levels = list("C_6"
    = 1)))[[1]]$Mean
MS_psi=data.frame(rbind(MS_psi_1,MS_psi_2,MS_psi_3,MS_psi_4,
    MS_psi_5,MS_psi_6))

F_psi_1=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_1))),split = list(levels = list("C_1"
    = 1)))[[1]]$F
F_psi_2=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_2))),split = list(levels = list("C_2"
    = 1)))[[1]]$F
F_psi_3=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_3))),split = list(levels = list("C_3"
    = 1)))[[1]]$F
F_psi_4=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_4))),split = list(levels = list("C_4"
    = 1)))[[1]]$F
```

```
F_psi_5=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_5))),split = list(levels = list("C_5"
    = 1)))[[1]]$F
F_psi_6=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_6))),split = list(levels = list("C_6"
    = 1)))[[1]]$F
F_psi=data.frame(rbind(F_psi_1,F_psi_2,F_psi_3,F_psi_4,F_psi_5,
    F_psi_6))

Pr_psi_1=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_1))),split = list(levels = list("C_1"
    = 1)))[[1]]$Pr
Pr_psi_2=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_2))),split = list(levels = list("C_2"
    = 1)))[[1]]$Pr
Pr_psi_3=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_3))),split = list(levels = list("C_3"
= 1)))[[1]]$Pr
Pr_psi_4=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_4))),split = list(levels = list("C_4"
    = 1)))[[1]]$Pr
Pr_psi_5=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_5))),split = list(levels = list("C_5"
    = 1)))[[1]]$Pr
Pr_psi_6=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_6))),split = list(levels = list("C_6"
    = 1)))[[1]]$Pr
Pr_psi=data.frame(rbind(Pr_psi_1,Pr_psi_2,Pr_psi_3,Pr_psi_4,
    Pr_psi_5,Pr_psi_6))

Contrast_names=c("Psi_1","Psi_2","Psi_3","Psi_4","Psi_5",
    "Psi_6")
Cont_mat=rbind("Psi_1"=C_1,"Psi_2"=C_2,"Psi_3"=C_3,"Psi_4"=C_4,
    "Psi_5"=C_5,"Psi_6"=C_6)
Contrasts=data.frame(G1=Cont_mat[,1], G2=Cont_mat[,2],
    G3=Cont_mat[,3], G4=Cont_mat[,4])

Contrast_Summary=data.frame(Contrast= Contrast_names,
    DF=Df_psi[,2], Contrast_SS=SS_psi[,2],
    Mean_Square=MS_psi[,2], F_Value=F_psi[,2], Pr=Pr_psi[,2])

# We now print the data and all the results

print(data)
print(multi_comp)
print('Means with the same letter are not significant')
summary(aov5)
print(Contrasts)
print(Contrast_Summary)
```

9.2.1.2 **[R] output**

```
> # Romeo and Juliet: Post Hoc Comparisons - Tukey's Test
> # ANOVA One-factor between subjects S(A)
```

```
> # NOTE 1: Install package 'gregmisc' in order to use
> #    make.contrasts
> # NOTE 2: make.contrasts will augment an incomplete set of
> #    orthogonal contrasts with "filler" contrasts
> # NOTE 3: Arrange your levels in alphabetical order, else R
> #    will do it for you


> # We have 1 Factor, A, with 4 levels: Context Before,
> #    Partial Context, Context After,No Context

> # We have 5 subjects per group. Therefore 5 x 4 = 20 subjects
> #    total.

> # We collect the data for each level of Factor A
> a1_Cont_before=c(5,9,8,4,9)
> a2_Part_cont=c(5,4,3,5,4)
> a3_Cont_after=c(2,4,5,4,1)
> a4_No_cont=c(3,3,2,4,3)

> # We now combine the observations into one long column (score).
> score=c(a1_Cont_before,a2_Part_cont, a3_Cont_after, a4_No_cont)

> # We generate a second column (levels), that identifies the
> #    group for each score.

> levels=factor(c(rep("a1_Cont_before",5),rep("a2_Part_cont",5),
> #    rep("a3_Cont_after",5),rep("a4_No_cont",5)))

> # We now form a data frame with the dependent variable and the
> #    factors.
> data=data.frame(score=score,group=levels)

> # We now define the pairwise comparisons
> C_1=c(1,-1,0,0)
> C_2=c(1,0,-1,0)
> C_3=c(1,0,0,-1)
> C_4=c(0,1,-1,0)
> C_5=c(0,1,0,-1)
> C_6=c(0,0,1,-1)

> # We now perform the test for multiple comparisons using
> #    "Tukey" correction.
> # The means with different letters are significantly different.
> # NOTE: The source for R script "mulcomp" has to be specified.
> means=tapply(score,levels,mean)
> source("/home/anjali/Desktop/R_scripts/09_Post_Hoc_Comp/
>     mulcomp.R")
> multi_comp=mulcomp(as.vector(means),5,16,2.350,conf.level=
>     .05,type= "TukeyHSD",decreasing=TRUE)

> # We now perfom on ANOVA on the data
> aov5=aov(score~levels)
```

```
> # We now organize the results
> Df_psi_1=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_1))),split = list(levels = list("C_1"
>     = 1)))[[1]]$Df
> Df_psi_2=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_2))),split = list(levels = list("C_2"
>     = 1)))[[1]]$Df
> Df_psi_3=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_3))),split = list(levels = list("C_3"
>     = 1)))[[1]]$Df
> Df_psi_4=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_4))),split = list(levels = list("C_4"
>     = 1)))[[1]]$Df
> Df_psi_5=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_5))),split = list(levels = list("C_5"
>     = 1)))[[1]]$Df
> Df_psi_6=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_5))),split = list(levels = list("C_6"
>     = 1)))[[1]]$Df
> Df_psi=data.frame(rbind(Df_psi_1,Df_psi_2,Df_psi_3,Df_psi_4,
>     Df_psi_5,Df_psi_6))

> SS_psi_1=summary(aov(score~levels,contrasts=list(levels=
make.contrasts(C_1))),split = list(levels = list("C_1"
= 1)))[[1]]$Sum

> SS_psi_2=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_2))),split = list(levels = list("C_2"
>     = 1)))[[1]]$Sum
> SS_psi_3=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_3))),split = list(levels = list("C_3"
>     = 1)))[[1]]$Sum
> SS_psi_4=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_4))),split = list(levels = list("C_4"
>     = 1)))[[1]]$Sum
> SS_psi_5=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_5))),split = list(levels = list("C_5"
>     = 1)))[[1]]$Sum
> SS_psi_6=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_6))),split = list(levels = list("C_6"
>     = 1)))[[1]]$Sum
> SS_psi=data.frame(rbind(SS_psi_1,SS_psi_2,SS_psi_3,SS_psi_4,
>     SS_psi_5,SS_psi_6))

> MS_psi_1=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_1))),split = list(levels = list("C_1"
>     = 1)))[[1]]$Mean
> MS_psi_2=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_2))),split = list(levels = list("C_2"
>     = 1)))[[1]]$Mean
> MS_psi_3=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_3))),split = list(levels = list("C_3"
>     = 1)))[[1]]$Mean
```

```
> MS_psi_4=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_4))),split = list(levels = list("C_4"
>     = 1)))[[1]]$Mean
> MS_psi_5=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_5))),split = list(levels = list("C_5"
>     = 1)))[[1]]$Mean
> MS_psi_6=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_6))),split = list(levels = list("C_6"
>     = 1)))[[1]]$Mean
> MS_psi=data.frame(rbind(MS_psi_1,MS_psi_2,MS_psi_3,MS_psi_4,
>     MS_psi_5,MS_psi_6))

> F_psi_1=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_1))),split = list(levels = list("C_1"
>     = 1)))[[1]]$F
> F_psi_2=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_2))),split = list(levels = list("C_2"
>     = 1)))[[1]]$F
> F_psi_3=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_3))),split = list(levels = list("C_3"
>     = 1)))[[1]]$F
> F_psi_4=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_4))),split = list(levels = list("C_4"
>     = 1)))[[1]]$F
> F_psi_5=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_5))),split = list(levels = list("C_5"
>     = 1)))[[1]]$F
> F_psi_6=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_6))),split = list(levels = list("C_6"
>     = 1)))[[1]]$F
> F_psi=data.frame(rbind(F_psi_1,F_psi_2,F_psi_3,F_psi_4,F_psi_5,
>     F_psi_6))

> Pr_psi_1=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_1))),split = list(levels = list("C_1"
>     = 1)))[[1]]$Pr
> Pr_psi_2=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_2))),split = list(levels = list("C_2"
>     = 1)))[[1]]$Pr
> Pr_psi_3=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_3))),split = list(levels = list("C_3"
= 1)))[[1]]$Pr
> Pr_psi_4=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_4))),split = list(levels = list("C_4"
>     = 1)))[[1]]$Pr
> Pr_psi_5=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_5))),split = list(levels = list("C_5"
>     = 1)))[[1]]$Pr
> Pr_psi_6=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_6))),split = list(levels = list("C_6"
>     = 1)))[[1]]$Pr
> Pr_psi=data.frame(rbind(Pr_psi_1,Pr_psi_2,Pr_psi_3,Pr_psi_4,
>     Pr_psi_5,Pr_psi_6))
```

```
> Contrast_names=c("Psi_1","Psi_2","Psi_3","Psi_4","Psi_5",
>     "Psi_6")
> Cont_mat=rbind("Psi_1"=C_1,"Psi_2"=C_2,"Psi_3"=C_3,"Psi_4"=C_4,
>     "Psi_5"=C_5,"Psi_6"=C_6)
> Contrasts=data.frame(G1=Cont_mat[,1], G2=Cont_mat[,2],
>     G3=Cont_mat[,3], G4=Cont_mat[,4])

> Contrast_Summary=data.frame(Contrast= Contrast_names,
>     DF=Df_psi[,2], Contrast_SS=SS_psi[,2],
>     Mean_Square=MS_psi[,2], F_Value=F_psi[,2], Pr=Pr_psi[,2])

> # We now print the data and all the results

> print(data)
       -----------------------
          Score          Group
       -----------------------
       1      5 a1_Cont_before
       2      9 a1_Cont_before
       3      8 a1_Cont_before
       4      4 a1_Cont_before
       5      9 a1_Cont_before
       6      5   a2_Part_cont
       7      4   a2_Part_cont
       8      3   a2_Part_cont
       9      5   a2_Part_cont
       10     4   a2_Part_cont
       11     2  a3_Cont_after
       12     4  a3_Cont_after
       13     5  a3_Cont_after
       14     4  a3_Cont_after
       15     1  a3_Cont_after
       16     3     a4_No_cont
       17     3     a4_No_cont
       18     2     a4_No_cont
       19     4     a4_No_cont
       20     3     a4_No_cont
       -----------------------

> print(multi_comp)

       $sem
       [1] 0.6855655

       $sed
       [1] 0.969536

       $MCT
       -------------
          medie test
       -------------
       M1   7.0    a
       M2   4.2    b
       M3   3.2    b
```

```
       M4   3.0    b
       -------------

       $Critical.Differences
       [1] 0.000000 2.773862 2.773862 2.773862


> print('Means with the same letter are not significant')
       [1] "Means with the same letter are not significant"

> summary(aov5)
       ----------------------------------------------
                   Df Sum Sq Mean Sq F value   Pr(>F)
       ----------------------------------------------
       levels       3 50.950  16.983   7.227 0.002782 **
       Residuals   16 37.600   2.350
       ----------------------------------------------
       ---
       Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'
       0.1 ' ' 1

> print(Contrasts)

       -----------------
             G1 G2 G3 G4
       -----------------
       Psi_1  1 -1  0  0
       Psi_2  1  0 -1  0
       Psi_3  1  0  0 -1
       Psi_4  0  1 -1  0
       Psi_5  0  1  0 -1
       Psi_6  0  0  1 -1
       -----------------

> print(Contrast_Summary)

       -------------------------------------------------
         Contrast DF Contrast_SS Mean_Square     F_Value
       -------------------------------------------------
       1    Psi_1  1        19.6        19.6  8.34042553
       2    Psi_2  1        36.1        36.1 15.36170213
       3    Psi_3  1        40.0        40.0 17.02127660
       4    Psi_4  1         2.5         2.5  1.06382979
       5    Psi_5  1         3.6         3.6  1.53191489
       6    Psi_6  1         0.1         0.1  0.04255319
       -------------------------------------------------

       ----------------
                     Pr
       1   0.0107046712
       2   0.0012227709
       3   0.0007927012
       4   0.3176832781
       5   0.2336812046
```

```
6   0.8391709477
----------------
```

9.3 Newman-Keuls' test

Essentially, the Newman-Keuls test consists of a sequential test in which the critical value depends on the range of each pair of means. To make the explanation easier, we will suppose that the means are ordered from the smallest to the largest. Hence $M_{1.}$ is the smallest mean, and $M_{A.}$ is the largest mean.

The Newman-Keuls test starts like the Tukey test. The largest difference between two means is selected. The range of the difference is A. The null hypothesis is tested for that mean using F_{range} following exactly the same procedure as for the Tukey test. If the null hypothesis cannot be rejected the test stops here, because not rejecting the null hypothesis for the largest difference implies not rejecting the null hypothesis for any other difference.

If the null hypothesis is rejected for the largest difference, then the two differences with a range of $A - 1$ are examined. They will be tested with a critical value of F_{range} selected for a range of $A-1$. When the null hypothesis cannot be rejected for a given difference, none of the differences included in that difference will be tested. If the null hypothesis can be rejected for a difference, then the procedure is re-iterated for a range of $A - 2$. The procedure is used until all the differences have been tested or declared nonsignificant by implication.

9.3.1 Taking off with Loftus...

In an experiment on eyewitness testimony, Loftus and Palmer (1974) tested the influence of the wording of a question on the answers given by eyewitnesses. They presented a film of a multiple car crash to 20 subjects. After seeing the film, subjects were asked to answer a number of specific questions. Among these questions, one question about the speed of the car was presented with five different versions:

- "HIT": About how fast were the cars going when they *hit* each other?
- "SMASH": About how fast were the cars going when they *smashed* into each other?
- "COLLIDE": About how fast were the cars going when they *collided* with each other?
- "BUMP": About how fast were the cars going when they *bumped* into each other?

- "CONTACT": About how fast were the cars going when they *contacted* each other?

The mean speed estimation by subjects for each version is given in the following Table:

	Experimental Group				
	Contact	Hit	Bump	Collide	Smash
	$M_{1.}$	$M_{2.}$	$M_{3.}$	$M_{4.}$	$M_{5.}$
$M_{a.}$	30.00	35.00	38.00	41.00	46.00

$S = 10$; $MS_{S(A)} = 80.00$
The obtained F ratios are given in the following Table.

	Experimental Group				
	Contact	Hit	Bump	Collide	Smash
Contact	—	1.56	4.00	7.56^{*}	16.00^{**}
Hit		—	0.56	2.25	7.56^{*}
Bump			—	0.56	4.00
Collide				—	1.56
Smash					—

9.3.1.1 [R] code

```
# Taking off with Loftus: Post Hoc Comparisons
#    Newman_Keul's Test
# ANOVA One-factor between subjects S(A)

# NOTE 1: Install package 'gregmisc' in order to use
#    make.contrasts
# NOTE 2: make.contrasts will augment an incomplete set of
#    orthogonal contrasts with "filler" contrasts
# NOTE 3: Arrange your levels in alphabetical order, else R
#    will do it for you


# We have 1 Factor, A, with 5 levels: Hit, Smash, Collide,
#    Bump, Contact

# We have 10 subjects per group. Therefore 10 x 5 = 50 subjects
#    total.

# We collect the data for each level of Factor A
a1_Contact=c(21,20,26,46,35,13,41,30,42,26)
a2_Hit=c(23,30,34,51,20,38,34,44,41,35)
a3_Bump=c(35,35,52,29,54,32,30,42,50,21)
a4_Collide=c(44,40,33,45,45,30,46,34,49,44)
```

```
a5_Smash=c(39,44,51,47,50,45,39,51,39,55)

# We now combine the observations into one long column (score).
score=c(a1_Contact,a2_Hit, a3_Bump, a4_Collide,a5_Smash)

# We generate a second column (levels), that identifies the
#    group for each score.

levels=factor(c(rep("a1_Contact",10),rep("a2_Hit",10),
    rep("a3_Bump",10),rep("a4_Collide",10),rep("a5_Smash",10)))

# We now form a data frame with the dependent variable and the
#    factors.
data=data.frame(score=score,group=levels)

# We now define the pairwise comparisons
C_1=c(1,-1,0,0,0)
C_2=c(1,0,-1,0,0)
C_3=c(1,0,0,-1,0)
C_4=c(1,0,0,0,-1)
C_5=c(0,1,-1,0,0)
C_6=c(0,1,0,-1,0)
C_7=c(0,1,0,0,-1)
C_8=c(0,0,1,-1,0)
C_9=c(0,0,1,0,-1)
C_10=c(0,0,0,1,-1)

# We now perform the test for multiple comparisons using
#    "Newman Keul's" correction.

# The means with different letters are significantly different.
# NOTE: The source for R script "mulcomp" has to be specified.

means=tapply(score,levels,mean)

source("/home/anjali/Desktop/R_scripts/09_Post_Hoc_Comp/
    mulcomp.R")
multi_comp=mulcomp(as.vector(means),10,45,80,conf.level=.05,
    type= "NewmanKeuls",decreasing=TRUE)

# We now perfom on ANOVA on the data
aov5=aov(score~levels)

# We now organize the results
Df_psi_1=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_1))),split = list(levels = list("C_1"
    = 1)))[[1]]$Df
Df_psi_2=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_2))),split = list(levels = list("C_2"
    = 1)))[[1]]$Df
Df_psi_3=summary(aov(score~levels,contrasts=list(levels=make.contrasts(C_3))),split = list(le
Df_psi_4=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_4))),split = list(levels = list("C_4"
    = 1)))[[1]]$Df
```

```
Df_psi_5=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_5))),split = list(levels = list("C_5"
    = 1)))[[1]]$Df
Df_psi_6=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_6))),split = list(levels = list("C_6"
    = 1)))[[1]]$Df
Df_psi_7=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_7))),split = list(levels = list("C_7"
    = 1)))[[1]]$Df
Df_psi_8=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_8))),split = list(levels = list("C_8"
    = 1)))[[1]]$Df
Df_psi_9=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_9))),split = list(levels = list("C_9"
    = 1)))[[1]]$Df
Df_psi_10=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_10))),split = list(levels = list("C_10"
    = 1)))[[1]]$Df
Df_psi=data.frame(rbind(Df_psi_1,Df_psi_2,Df_psi_3,Df_psi_4,
    Df_psi_5,Df_psi_6,Df_psi_7,Df_psi_8,Df_psi_9,Df_psi_10))

SS_psi_1=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_1))),split = list(levels = list("C_1"
    = 1)))[[1]]$Sum
SS_psi_2=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_2))),split = list(levels = list("C_2"
    = 1)))[[1]]$Sum
SS_psi_3=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_3))),split = list(levels = list("C_3"
    = 1)))[[1]]$Sum
SS_psi_4=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_4))),split = list(levels = list("C_4"
    = 1)))[[1]]$Sum
SS_psi_5=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_5))),split = list(levels = list("C_5"
    = 1)))[[1]]$Sum
SS_psi_6=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_6))),split = list(levels = list("C_6"
    = 1)))[[1]]$Sum

SS_psi_7=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_7))),split = list(levels = list("C_7"
    = 1)))[[1]]$Sum
SS_psi_8=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_8))),split = list(levels = list("C_8"
    = 1)))[[1]]$Sum
SS_psi_9=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_9))),split = list(levels = list("C_9"
    = 1)))[[1]]$Sum
SS_psi_10=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_10))),split = list(levels = list("C_10"
    = 1)))[[1]]$Sum
SS_psi=data.frame(rbind(SS_psi_1,SS_psi_2,SS_psi_3,SS_psi_4,
    SS_psi_5,SS_psi_6,SS_psi_7,SS_psi_8,SS_psi_9,SS_psi_10))
```

```
MS_psi_1=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_1))),split = list(levels = list("C_1"
    = 1)))[[1]]$Mean
MS_psi_2=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_2))),split = list(levels = list("C_2"
    = 1)))[[1]]$Mean
MS_psi_3=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_3))),split = list(levels = list("C_3"
    = 1)))[[1]]$Mean
MS_psi_4=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_4))),split = list(levels = list("C_4"
    = 1)))[[1]]$Mean
MS_psi_5=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_5))),split = list(levels = list("C_5"
    = 1)))[[1]]$Mean
MS_psi_6=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_6))),split = list(levels = list("C_6"
    = 1)))[[1]]$Mean
MS_psi_7=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_7))),split = list(levels = list("C_7"
    = 1)))[[1]]$Mean
MS_psi_8=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_8))),split = list(levels = list("C_8"
    = 1)))[[1]]$Mean
MS_psi_9=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_9))),split = list(levels = list("C_9"
    = 1)))[[1]]$Mean
MS_psi_10=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_10))),split = list(levels = list("C_10"
    = 1)))[[1]]$Mean
MS_psi=data.frame(rbind(MS_psi_1,MS_psi_2,MS_psi_3,MS_psi_4,
    MS_psi_5,MS_psi_6,MS_psi_7,MS_psi_8,MS_psi_9,MS_psi_10))

F_psi_1=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_1))),split = list(levels = list("C_1"
    = 1)))[[1]]$F
F_psi_2=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_2))),split = list(levels = list("C_2"
    = 1)))[[1]]$F
F_psi_3=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_3))),split = list(levels = list("C_3"
    = 1)))[[1]]$F
F_psi_4=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_4))),split = list(levels = list("C_4"
    = 1)))[[1]]$F
F_psi_5=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_5))),split = list(levels = list("C_5"
    = 1)))[[1]]$F
F_psi_6=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_6))),split = list(levels = list("C_6"
    = 1)))[[1]]$F
F_psi_7=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_7))),split = list(levels = list("C_7"
```

```
    = 1)))[[1]]$F
F_psi_8=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_8))),split = list(levels = list("C_8"
    = 1)))[[1]]$F
F_psi_9=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_9))),split = list(levels = list("C_9"
    = 1)))[[1]]$F
F_psi_10=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_10))),split = list(levels = list("C_10"
    = 1)))[[1]]$F
F_psi=data.frame(rbind(F_psi_1,F_psi_2,F_psi_3,F_psi_4,
    F_psi_5,F_psi_6,F_psi_7,F_psi_8,F_psi_9,F_psi_10))

Pr_psi_1=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_1))),split = list(levels = list("C_1"
    = 1)))[[1]]$Pr
Pr_psi_2=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_2))),split = list(levels = list("C_2"
    = 1)))[[1]]$Pr
Pr_psi_3=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_3))),split = list(levels = list("C_3"
    = 1)))[[1]]$Pr
Pr_psi_4=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_4))),split = list(levels = list("C_4"
    = 1)))[[1]]$Pr
Pr_psi_5=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_5))),split = list(levels = list("C_5"
    = 1)))[[1]]$Pr
Pr_psi_6=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_6))),split = list(levels = list("C_6"
    = 1)))[[1]]$Pr
Pr_psi_7=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_7))),split = list(levels = list("C_7"
    = 1)))[[1]]$Pr
Pr_psi_8=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_8))),split = list(levels = list("C_8"
    = 1)))[[1]]$Pr
Pr_psi_9=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_9))),split = list(levels = list("C_9"
    = 1)))[[1]]$Pr
Pr_psi_10=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_10))),split = list(levels = list("C_10"
    = 1)))[[1]]$Pr
Pr_psi=data.frame(rbind(Pr_psi_1,Pr_psi_2,Pr_psi_3,Pr_psi_4,
    Pr_psi_5,Pr_psi_6,Pr_psi_7,Pr_psi_8,Pr_psi_9,Pr_psi_10))

Contrast_names=c("Psi_1","Psi_2","Psi_3","Psi_4","Psi_5","Psi_6",
    "Psi_7","Psi_8","Psi_9","Psi_10")
Cont_mat=rbind("Psi_1"=C_1,"Psi_2"=C_2,"Psi_3"=C_3,"Psi_4"=C_4,
    "Psi_5"=C_5,"Psi_6"=C_6,"Psi_7"=C_7,"Psi_8"=C_8,"Psi_9"=C_9,
    "Psi_10"=C_10)

Contrasts=data.frame(G1=Cont_mat[,1], G2=Cont_mat[,2],
    G3=Cont_mat[,3], G4=Cont_mat[,4], G5=Cont_mat[,5])
```

```
Contrast_Summary=data.frame(Contrast= Contrast_names,
    DF=Df_psi[,2], Contrast_SS=SS_psi[,2],
    Mean_Square=MS_psi[,2], F_Value=F_psi[,2], Pr=Pr_psi[,2])

# We now print the data and all the results
print(data)
print(multi_comp)
print('Means with the same letter are not significant')
summary(aov5)
print(Contrasts)
print(Contrast_Summary)
```

9.3.1.2 [R] output

```
> # Taking off with Loftus: Post Hoc Comparisons
> #    Newman_Keul's Test
> # ANOVA One-factor between subjects S(A)

> # NOTE 1: Install package 'gregmisc' in order to use
> #    make.contrasts
> # NOTE 2: make.contrasts will augment an incomplete set of
> #    orthogonal contrasts with "filler" contrasts
> # NOTE 3: Arrange your levels in alphabetical order, else R
> #    will do it for you


> # We have 1 Factor, A, with 5 levels: Hit, Smash, Collide,
> #    Bump, Contact

> # We have 10 subjects per group. Therefore 10 x 5 = 50 subjects
> #    total.

> # We collect the data for each level of Factor A
> a1_Contact=c(21,20,26,46,35,13,41,30,42,26)
> a2_Hit=c(23,30,34,51,20,38,34,44,41,35)
> a3_Bump=c(35,35,52,29,54,32,30,42,50,21)
> a4_Collide=c(44,40,33,45,45,30,46,34,49,44)
> a5_Smash=c(39,44,51,47,50,45,39,51,39,55)

> # We now combine the observations into one long column (score).
> score=c(a1_Contact,a2_Hit, a3_Bump, a4_Collide,a5_Smash)

> # We generate a second column (levels), that identifies the
> #    group for each score.

> levels=factor(c(rep("a1_Contact",10),rep("a2_Hit",10),
>     rep("a3_Bump",10),rep("a4_Collide",10),rep("a5_Smash",10)))

> # We now form a data frame with the dependent variable and the
> #    factors.
> data=data.frame(score=score,group=levels)

> # We now define the pairwise comparisons
```

```
> C_1=c(1,-1,0,0,0)
> C_2=c(1,0,-1,0,0)
> C_3=c(1,0,0,-1,0)
> C_4=c(1,0,0,0,-1)
> C_5=c(0,1,-1,0,0)
> C_6=c(0,1,0,-1,0)
> C_7=c(0,1,0,0,-1)
> C_8=c(0,0,1,-1,0)
> C_9=c(0,0,1,0,-1)
> C_10=c(0,0,0,1,-1)

> # We now perform the test for multiple comparisons using
> #    "Newman Keul's" correction.

> # The means with different letters are significantly different.
> # NOTE: The source for R script "mulcomp" has to be specified.

> means=tapply(score,levels,mean)

> source("/home/anjali/Desktop/R_scripts/09_Post_Hoc_Comp/
>     mulcomp.R")
> multi_comp=mulcomp(as.vector(means),10,45,80,conf.level=.05,
>     type= "NewmanKeuls",decreasing=TRUE)

> # We now perfom on ANOVA on the data
> aov5=aov(score~levels)

> # We now organize the results
> Df_psi_1=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_1))),split = list(levels = list("C_1"
>     = 1)))[[1]]$Df
> Df_psi_2=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_2))),split = list(levels = list("C_2"
>     = 1)))[[1]]$Df
> Df_psi_3=summary(aov(score~levels,contrasts=list(levels=make.contrasts(C_3))),split = list(levels
> Df_psi_4=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_4))),split = list(levels = list("C_4"
>     = 1)))[[1]]$Df
> Df_psi_5=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_5))),split = list(levels = list("C_5"
>     = 1)))[[1]]$Df
> Df_psi_6=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_6))),split = list(levels = list("C_6"
>     = 1)))[[1]]$Df
> Df_psi_7=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_7))),split = list(levels = list("C_7"
>     = 1)))[[1]]$Df
> Df_psi_8=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_8))),split = list(levels = list("C_8"
>     = 1)))[[1]]$Df
> Df_psi_9=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_9))),split = list(levels = list("C_9"
>     = 1)))[[1]]$Df
> Df_psi_10=summary(aov(score~levels,contrasts=list(levels=
```

```
>     make.contrasts(C_10))),split = list(levels = list("C_10"
>     = 1)))[[1]]$Df
> Df_psi=data.frame(rbind(Df_psi_1,Df_psi_2,Df_psi_3,Df_psi_4,
>     Df_psi_5,Df_psi_6,Df_psi_7,Df_psi_8,Df_psi_9,Df_psi_10))

> SS_psi_1=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_1))),split = list(levels = list("C_1"
>     = 1)))[[1]]$Sum
> SS_psi_2=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_2))),split = list(levels = list("C_2"
>     = 1)))[[1]]$Sum
> SS_psi_3=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_3))),split = list(levels = list("C_3"
>     = 1)))[[1]]$Sum
> SS_psi_4=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_4))),split = list(levels = list("C_4"
>     = 1)))[[1]]$Sum
> SS_psi_5=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_5))),split = list(levels = list("C_5"
>     = 1)))[[1]]$Sum
> SS_psi_6=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_6))),split = list(levels = list("C_6"
>     = 1)))[[1]]$Sum

> SS_psi_7=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_7))),split = list(levels = list("C_7"
>     = 1)))[[1]]$Sum
> SS_psi_8=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_8))),split = list(levels = list("C_8"
>     = 1)))[[1]]$Sum
> SS_psi_9=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_9))),split = list(levels = list("C_9"
>     = 1)))[[1]]$Sum
> SS_psi_10=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_10))),split = list(levels = list("C_10"
>     = 1)))[[1]]$Sum
> SS_psi=data.frame(rbind(SS_psi_1,SS_psi_2,SS_psi_3,SS_psi_4,
>     SS_psi_5,SS_psi_6,SS_psi_7,SS_psi_8,SS_psi_9,SS_psi_10))

> MS_psi_1=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_1))),split = list(levels = list("C_1"
>     = 1)))[[1]]$Mean
> MS_psi_2=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_2))),split = list(levels = list("C_2"
>     = 1)))[[1]]$Mean
> MS_psi_3=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_3))),split = list(levels = list("C_3"
>     = 1)))[[1]]$Mean
> MS_psi_4=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_4))),split = list(levels = list("C_4"
>     = 1)))[[1]]$Mean
> MS_psi_5=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_5))),split = list(levels = list("C_5"
>     = 1)))[[1]]$Mean
```

```
> MS_psi_6=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_6))),split = list(levels = list("C_6"
>     = 1)))[[1]]$Mean
> MS_psi_7=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_7))),split = list(levels = list("C_7"
>     = 1)))[[1]]$Mean
> MS_psi_8=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_8))),split = list(levels = list("C_8"
>     = 1)))[[1]]$Mean
> MS_psi_9=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_9))),split = list(levels = list("C_9"
>     = 1)))[[1]]$Mean
> MS_psi_10=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_10))),split = list(levels = list("C_10"
>     = 1)))[[1]]$Mean
> MS_psi=data.frame(rbind(MS_psi_1,MS_psi_2,MS_psi_3,MS_psi_4,
>     MS_psi_5,MS_psi_6,MS_psi_7,MS_psi_8,MS_psi_9,MS_psi_10))

> F_psi_1=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_1))),split = list(levels = list("C_1"
>     = 1)))[[1]]$F
> F_psi_2=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_2))),split = list(levels = list("C_2"
>     = 1)))[[1]]$F
> F_psi_3=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_3))),split = list(levels = list("C_3"
>     = 1)))[[1]]$F
> F_psi_4=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_4))),split = list(levels = list("C_4"
>     = 1)))[[1]]$F
> F_psi_5=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_5))),split = list(levels = list("C_5"
>     = 1)))[[1]]$F
> F_psi_6=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_6))),split = list(levels = list("C_6"
>     = 1)))[[1]]$F
> F_psi_7=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_7))),split = list(levels = list("C_7"
>     = 1)))[[1]]$F
> F_psi_8=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_8))),split = list(levels = list("C_8"
>     = 1)))[[1]]$F
> F_psi_9=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_9))),split = list(levels = list("C_9"
>     = 1)))[[1]]$F
> F_psi_10=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_10))),split = list(levels = list("C_10"
>     = 1)))[[1]]$F
> F_psi=data.frame(rbind(F_psi_1,F_psi_2,F_psi_3,F_psi_4,
>     F_psi_5,F_psi_6,F_psi_7,F_psi_8,F_psi_9,F_psi_10))

> Pr_psi_1=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_1))),split = list(levels = list("C_1"
>     = 1)))[[1]]$Pr
```

```
> Pr_psi_2=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_2))),split = list(levels = list("C_2"
>     = 1)))[[1]]$Pr
> Pr_psi_3=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_3))),split = list(levels = list("C_3"
>     = 1)))[[1]]$Pr
> Pr_psi_4=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_4))),split = list(levels = list("C_4"
>     = 1)))[[1]]$Pr
> Pr_psi_5=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_5))),split = list(levels = list("C_5"
>     = 1)))[[1]]$Pr
> Pr_psi_6=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_6))),split = list(levels = list("C_6"
>     = 1)))[[1]]$Pr
> Pr_psi_7=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_7))),split = list(levels = list("C_7"
>     = 1)))[[1]]$Pr
> Pr_psi_8=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_8))),split = list(levels = list("C_8"
>     = 1)))[[1]]$Pr
> Pr_psi_9=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_9))),split = list(levels = list("C_9"
>     = 1)))[[1]]$Pr
> Pr_psi_10=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_10))),split = list(levels = list("C_10"
>     = 1)))[[1]]$Pr
> Pr_psi=data.frame(rbind(Pr_psi_1,Pr_psi_2,Pr_psi_3,Pr_psi_4,
>     Pr_psi_5,Pr_psi_6,Pr_psi_7,Pr_psi_8,Pr_psi_9,Pr_psi_10))

> Contrast_names=c("Psi_1","Psi_2","Psi_3","Psi_4","Psi_5","Psi_6",
>     "Psi_7","Psi_8","Psi_9","Psi_10")
> Cont_mat=rbind("Psi_1"=C_1,"Psi_2"=C_2,"Psi_3"=C_3,"Psi_4"=C_4,
>     "Psi_5"=C_5,"Psi_6"=C_6,"Psi_7"=C_7,"Psi_8"=C_8,"Psi_9"=C_9,
>     "Psi_10"=C_10)

> Contrasts=data.frame(G1=Cont_mat[,1], G2=Cont_mat[,2],
>     G3=Cont_mat[,3], G4=Cont_mat[,4], G5=Cont_mat[,5])

> Contrast_Summary=data.frame(Contrast= Contrast_names,
>     DF=Df_psi[,2], Contrast_SS=SS_psi[,2],
>     Mean_Square=MS_psi[,2], F_Value=F_psi[,2], Pr=Pr_psi[,2])

> # We now print the data and all the results

> print(data)

       -------------------
          Score      Group
       -------------------
       1     21 a1_Contact
       2     20 a1_Contact
       3     26 a1_Contact
       4     46 a1_Contact
```

```
5      35 a1_Contact
6      13 a1_Contact
7      41 a1_Contact
8      30 a1_Contact
9      42 a1_Contact
10     26 a1_Contact
11     23     a2_Hit
12     30     a2_Hit
13     34     a2_Hit
14     51     a2_Hit
15     20     a2_Hit
16     38     a2_Hit
17     34     a2_Hit
18     44     a2_Hit
19     41     a2_Hit
20     35     a2_Hit
21     35    a3_Bump
22     35    a3_Bump
23     52    a3_Bump
24     29    a3_Bump
25     54    a3_Bump
26     32    a3_Bump
27     30    a3_Bump
28     42    a3_Bump
29     50    a3_Bump
30     21    a3_Bump
31     44 a4_Collide
32     40 a4_Collide
33     33 a4_Collide
34     45 a4_Collide
35     45 a4_Collide
36     30 a4_Collide
37     46 a4_Collide
38     34 a4_Collide
39     49 a4_Collide
40     44 a4_Collide
41     39   a5_Smash
42     44   a5_Smash
43     51   a5_Smash
44     47   a5_Smash
45     50   a5_Smash
46     45   a5_Smash
47     39   a5_Smash
48     51   a5_Smash
49     39   a5_Smash
50     55   a5_Smash
-------------------
```

```
> print(multi_comp)

       $sem
       [1] 2.828427

       $sed
```

```
[1] 4

$MCT
-------------
   medie test
-------------
M5    46    a
M4    41   ab
M3    38  abc
M2    35   bc
M1    30    c
--------------

$Critical.Differences
[1]  0.000000  8.056414  9.694454 10.670798 11.365800


> print('Means with the same letter are not significant')

       [1] "Means with the same letter are not significant"

> summary(aov5)

       ----------------------------------------------
                   Df Sum Sq Mean Sq F value   Pr(>F)
       ----------------------------------------------
       levels       4   1460     365  4.5625 0.003532 **
       Residuals   45   3600      80
       ----------------------------------------------
       ---
       Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'
       0.1 ' ' 1

> print(Contrasts)

       ---------------------
              G1 G2 G3 G4 G5
       ---------------------
       Psi_1   1 -1  0  0  0
       Psi_2   1  0 -1  0  0
       Psi_3   1  0  0 -1  0
       Psi_4   1  0  0  0 -1
       Psi_5   0  1 -1  0  0
       Psi_6   0  1  0 -1  0
       Psi_7   0  1  0  0 -1
       Psi_8   0  0  1 -1  0
       Psi_9   0  0  1  0 -1
       Psi_10  0  0  0  1 -1
       ---------------------

> print(Contrast_Summary)

       -----------------------------------------------------------
          Contrast DF Contrast_SS Mean_Square F_Value           Pr
```

```
------------------------------------------------------------
1     Psi_1  1        125        125  1.5625 0.2177613192
2     Psi_2  1        320        320  4.0000 0.0515576878
3     Psi_3  1        605        605  7.5625 0.0085519105
4     Psi_4  1       1280       1280 16.0000 0.0002332699
5     Psi_5  1         45         45  0.5625 0.4571581827
6     Psi_6  1        180        180  2.2500 0.1405983243
7     Psi_7  1        605        605  7.5625 0.0085519105
8     Psi_8  1         45         45  0.5625 0.4571581827
9     Psi_9  1        320        320  4.0000 0.0515576878
10   Psi_10  1        125        125  1.5625 0.2177613192
------------------------------------------------------------
```

9.3.2 Guess who?

Using the data from Bransford's Romeo and Juliet, we ran the *post hoc* contrasts with the Newman-Keuls test.

9.3.2.1 [R] code

```
# Romeo and Juliet: Post Hoc Comparisons - Newman-Keuls's Test
# ANOVA One-factor between subjects S(A)

# NOTE 1: Install package 'gregmisc' in order to use
#   make.contrasts
# NOTE 2: make.contrasts will augment an incomplete set of
#    orthogonal contrasts with "filler" contrasts
# NOTE 3: Arrange your levels in alphabetical order, else R
#    will do it for you


# We have 1 Factor, A, with 4 levels: No Context,
#   Context Before, Context After, Partial Context

# We have 5 subjects per group. Therefore 5 x 4 = 20 subjects
#   total.

# We collect the data for each level of Factor A
a1_Cont_before=c(5,9,8,4,9)
a2_Part_cont=c(5,4,3,5,4)
a3_Cont_after=c(2,4,5,4,1)
a4_No_cont=c(3,3,2,4,3)

# We now combine the observations into one long column (score).
score=c(a1_Cont_before,a2_Part_cont, a3_Cont_after, a4_No_cont)

# We generate a second column (levels), that identifies the
#   group for each score.
levels=factor(c(rep("a1_Cont_before",5),rep("a2_Part_cont",5),
    rep("a3_Cont_after",5),rep("a4_No_cont",5)))

# We now form a data frame with the dependent variable and the
    factors.
```

```
data=data.frame(score=score,group=levels)

# We now define the pairwise comparisons
C_1=c(1,-1,0,0)
C_2=c(1,0,-1,0)
C_3=c(1,0,0,-1)
C_4=c(0,1,-1,0)
C_5=c(0,1,0,-1)
C_6=c(0,0,1,-1)

# We now perform the test for multiple comparisons using
    "Newman Keul's" correction.
# The means with different letters are significantly different.
# NOTE: The source for R script "mulcomp" has to be specified.
means=tapply(score,levels,mean)
source("/home/anjali/Desktop/R_scripts/09_Post_Hoc_Comp/
    mulcomp.R")
multi_comp=mulcomp(as.vector(means),5,16,2.350,conf.level=.05,
    type= "NewmanKeuls",decreasing=TRUE)

# We now perfom on ANOVA on the data
aov5=aov(score~levels)

# We now organize the results
Df_psi_1=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_1))),split = list(levels = list("C_1"
    = 1)))[[1]]$Df
Df_psi_2=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_2))),split = list(levels = list("C_2"
    = 1)))[[1]]$Df
Df_psi_3=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_3))),split = list(levels = list("C_3"
    = 1)))[[1]]$Df
Df_psi_4=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_4))),split = list(levels = list("C_4"
    = 1)))[[1]]$Df
Df_psi_5=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_5))),split = list(levels = list("C_5"
    = 1)))[[1]]$Df
Df_psi_6=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_5))),split = list(levels = list("C_6"
    = 1)))[[1]]$Df
Df_psi=data.frame(rbind(Df_psi_1,Df_psi_2,Df_psi_3,Df_psi_4,
    Df_psi_5,Df_psi_6))

SS_psi_1=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_1))),split = list(levels = list("C_1"
    = 1)))[[1]]$Sum
SS_psi_2=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_2))),split = list(levels = list("C_2"
    = 1)))[[1]]$Sum
SS_psi_3=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_3))),split = list(levels = list("C_3"
    = 1)))[[1]]$Sum
```

```
SS_psi_4=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_4))),split = list(levels = list("C_4"
    = 1)))[[1]]$Sum
SS_psi_5=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_5))),split = list(levels = list("C_5"
    = 1)))[[1]]$Sum
SS_psi_6=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_6))),split = list(levels = list("C_6"
    = 1)))[[1]]$Sum
SS_psi=data.frame(rbind(SS_psi_1,SS_psi_2,SS_psi_3,SS_psi_4,
    SS_psi_5,SS_psi_6))

MS_psi_1=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_1))),split = list(levels = list("C_1"
    = 1)))[[1]]$Mean
MS_psi_2=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_2))),split = list(levels = list("C_2"
    = 1)))[[1]]$Mean
MS_psi_3=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_3))),split = list(levels = list("C_3"
    = 1)))[[1]]$Mean
MS_psi_4=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_4))),split = list(levels = list("C_4"
    = 1)))[[1]]$Mean
MS_psi_5=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_5))),split = list(levels = list("C_5"
    = 1)))[[1]]$Mean
MS_psi_6=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_6))),split = list(levels = list("C_6"
    = 1)))[[1]]$Mean
MS_psi=data.frame(rbind(MS_psi_1,MS_psi_2,MS_psi_3,MS_psi_4,
    MS_psi_5,MS_psi_6))

F_psi_1=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_1))),split = list(levels = list("C_1"
    = 1)))[[1]]$F
F_psi_2=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_2))),split = list(levels = list("C_2"
    = 1)))[[1]]$F
F_psi_3=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_3))),split = list(levels = list("C_3"
    = 1)))[[1]]$F
F_psi_4=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_4))),split = list(levels = list("C_4"
    = 1)))[[1]]$F
F_psi_5=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_5))),split = list(levels = list("C_5"
    = 1)))[[1]]$F
F_psi_6=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_6))),split = list(levels = list("C_6"
    = 1)))[[1]]$F
F_psi=data.frame(rbind(F_psi_1,F_psi_2,F_psi_3,F_psi_4,F_psi_5,
    F_psi_6))
```

```
Pr_psi_1=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_1))),split = list(levels = list("C_1"
    = 1)))[[1]]$Pr
Pr_psi_2=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_2))),split = list(levels = list("C_2"
    = 1)))[[1]]$Pr
Pr_psi_3=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_3))),split = list(levels = list("C_3"
    = 1)))[[1]]$Pr
Pr_psi_4=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_4))),split = list(levels = list("C_4"
    = 1)))[[1]]$Pr
Pr_psi_5=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_5))),split = list(levels = list("C_5"
    = 1)))[[1]]$Pr
Pr_psi_6=summary(aov(score~levels,contrasts=list(levels=
    make.contrasts(C_6))),split = list(levels = list("C_6"
    = 1)))[[1]]$Pr
Pr_psi=data.frame(rbind(Pr_psi_1,Pr_psi_2,Pr_psi_3,Pr_psi_4,
    Pr_psi_5,Pr_psi_6))

Contrast_names=c("Psi_1","Psi_2","Psi_3","Psi_4","Psi_5",
    "Psi_6")
Cont_mat=rbind("Psi_1"=C_1,"Psi_2"=C_2,"Psi_3"=C_3,"Psi_4"=C_4,
    "Psi_5"=C_5,"Psi_6"=C_6)
Contrasts=data.frame(G1=Cont_mat[,1], G2=Cont_mat[,2],
    G3=Cont_mat[,3], G4=Cont_mat[,4])
Contrast_Summary=data.frame(Contrast= Contrast_names,
    DF=Df_psi[,2], Contrast_SS=SS_psi[,2],
    Mean_Square=MS_psi[,2], F_Value=F_psi[,2], Pr=Pr_psi[,2])

# We now print the data and all the results
print(data)
print(multi_comp)
print('Means with the same letter are not significant')
summary(aov5)
print(Contrasts)
print(Contrast_Summary)
```

9.3.2.2 [R] output

```
> # Romeo and Juliet: Post Hoc Comparisons - Newman-Keuls's Test
> # ANOVA One-factor between subjects S(A)

> # NOTE 1: Install package 'gregmisc' in order to use
> #   make.contrasts
> # NOTE 2: make.contrasts will augment an incomplete set of
> #    orthogonal contrasts with "filler" contrasts
> # NOTE 3: Arrange your levels in alphabetical order, else R
> #    will do it for you


> # We have 1 Factor, A, with 4 levels: No Context,
> #   Context Before, Context After, Partial Context
```

```
> # We have 5 subjects per group. Therefore 5 x 4 = 20 subjects
> #   total.

> # We collect the data for each level of Factor A
> a1_Cont_before=c(5,9,8,4,9)
> a2_Part_cont=c(5,4,3,5,4)
> a3_Cont_after=c(2,4,5,4,1)
> a4_No_cont=c(3,3,2,4,3)

> # We now combine the observations into one long column (score).
> score=c(a1_Cont_before,a2_Part_cont, a3_Cont_after, a4_No_cont)

> # We generate a second column (levels), that identifies the
> #   group for each score.
> levels=factor(c(rep("a1_Cont_before",5),rep("a2_Part_cont",5),
>     rep("a3_Cont_after",5),rep("a4_No_cont",5)))

> # We now form a data frame with the dependent variable and the
>     factors.
> data=data.frame(score=score,group=levels)

> # We now define the pairwise comparisons
> C_1=c(1,-1,0,0)
> C_2=c(1,0,-1,0)
> C_3=c(1,0,0,-1)
> C_4=c(0,1,-1,0)
> C_5=c(0,1,0,-1)
> C_6=c(0,0,1,-1)

> # We now perform the test for multiple comparisons using
>     "Newman Keul's" correction.
> # The means with different letters are significantly different.
> # NOTE: The source for R script "mulcomp" has to be specified.
> means=tapply(score,levels,mean)
> source("/home/anjali/Desktop/R_scripts/09_Post_Hoc_Comp/
>     mulcomp.R")
> multi_comp=mulcomp(as.vector(means),5,16,2.350,conf.level=.05,
>     type= "NewmanKeuls",decreasing=TRUE)

> # We now perfom on ANOVA on the data
> aov5=aov(score~levels)

> # We now organize the results
> Df_psi_1=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_1))),split = list(levels = list("C_1"
>     = 1)))[[1]]$Df
> Df_psi_2=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_2))),split = list(levels = list("C_2"
>     = 1)))[[1]]$Df
> Df_psi_3=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_3))),split = list(levels = list("C_3"
>     = 1)))[[1]]$Df
> Df_psi_4=summary(aov(score~levels,contrasts=list(levels=
```

```
>     make.contrasts(C_4))),split = list(levels = list("C_4"
>     = 1)))[[1]]$Df
> Df_psi_5=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_5))),split = list(levels = list("C_5"
>     = 1)))[[1]]$Df
> Df_psi_6=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_5))),split = list(levels = list("C_6"
>     = 1)))[[1]]$Df
> Df_psi=data.frame(rbind(Df_psi_1,Df_psi_2,Df_psi_3,Df_psi_4,
>     Df_psi_5,Df_psi_6))

> SS_psi_1=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_1))),split = list(levels = list("C_1"
>     = 1)))[[1]]$Sum
> SS_psi_2=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_2))),split = list(levels = list("C_2"
>     = 1)))[[1]]$Sum
> SS_psi_3=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_3))),split = list(levels = list("C_3"
>     = 1)))[[1]]$Sum
> SS_psi_4=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_4))),split = list(levels = list("C_4"
>     = 1)))[[1]]$Sum
> SS_psi_5=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_5))),split = list(levels = list("C_5"
>     = 1)))[[1]]$Sum
> SS_psi_6=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_6))),split = list(levels = list("C_6"
>     = 1)))[[1]]$Sum
> SS_psi=data.frame(rbind(SS_psi_1,SS_psi_2,SS_psi_3,SS_psi_4,
>     SS_psi_5,SS_psi_6))

> MS_psi_1=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_1))),split = list(levels = list("C_1"
>     = 1)))[[1]]$Mean
> MS_psi_2=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_2))),split = list(levels = list("C_2"
>     = 1)))[[1]]$Mean
> MS_psi_3=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_3))),split = list(levels = list("C_3"
>     = 1)))[[1]]$Mean
> MS_psi_4=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_4))),split = list(levels = list("C_4"
>     = 1)))[[1]]$Mean
> MS_psi_5=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_5))),split = list(levels = list("C_5"
>     = 1)))[[1]]$Mean
> MS_psi_6=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_6))),split = list(levels = list("C_6"
>     = 1)))[[1]]$Mean
> MS_psi=data.frame(rbind(MS_psi_1,MS_psi_2,MS_psi_3,MS_psi_4,
>     MS_psi_5,MS_psi_6))

> F_psi_1=summary(aov(score~levels,contrasts=list(levels=
```

```
>     make.contrasts(C_1))),split = list(levels = list("C_1"
>     = 1)))[[1]]$F
> F_psi_2=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_2))),split = list(levels = list("C_2"
>     = 1)))[[1]]$F
> F_psi_3=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_3))),split = list(levels = list("C_3"
>     = 1)))[[1]]$F
> F_psi_4=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_4))),split = list(levels = list("C_4"
>     = 1)))[[1]]$F
> F_psi_5=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_5))),split = list(levels = list("C_5"
>     = 1)))[[1]]$F
> F_psi_6=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_6))),split = list(levels = list("C_6"
>     = 1)))[[1]]$F
> F_psi=data.frame(rbind(F_psi_1,F_psi_2,F_psi_3,F_psi_4,F_psi_5,
>     F_psi_6))

> Pr_psi_1=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_1))),split = list(levels = list("C_1"
>     = 1)))[[1]]$Pr
> Pr_psi_2=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_2))),split = list(levels = list("C_2"
>     = 1)))[[1]]$Pr
> Pr_psi_3=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_3))),split = list(levels = list("C_3"
>     = 1)))[[1]]$Pr
> Pr_psi_4=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_4))),split = list(levels = list("C_4"
>     = 1)))[[1]]$Pr
> Pr_psi_5=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_5))),split = list(levels = list("C_5"
>     = 1)))[[1]]$Pr
> Pr_psi_6=summary(aov(score~levels,contrasts=list(levels=
>     make.contrasts(C_6))),split = list(levels = list("C_6"
>     = 1)))[[1]]$Pr
> Pr_psi=data.frame(rbind(Pr_psi_1,Pr_psi_2,Pr_psi_3,Pr_psi_4,
>     Pr_psi_5,Pr_psi_6))

> Contrast_names=c("Psi_1","Psi_2","Psi_3","Psi_4","Psi_5",
>     "Psi_6")
> Cont_mat=rbind("Psi_1"=C_1,"Psi_2"=C_2,"Psi_3"=C_3,"Psi_4"=C_4,
>     "Psi_5"=C_5,"Psi_6"=C_6)
> Contrasts=data.frame(G1=Cont_mat[,1], G2=Cont_mat[,2],
>     G3=Cont_mat[,3], G4=Cont_mat[,4])
> Contrast_Summary=data.frame(Contrast= Contrast_names,
>     DF=Df_psi[,2], Contrast_SS=SS_psi[,2],
>     Mean_Square=MS_psi[,2], F_Value=F_psi[,2], Pr=Pr_psi[,2])

> # We now print the data and all the results

> print(data)
```

```
-----------------------
   Score         Group
-----------------------
1      5 a1_Cont_before
2      9 a1_Cont_before
3      8 a1_Cont_before
4      4 a1_Cont_before
5      9 a1_Cont_before
6      5   a2_Part_cont
7      4   a2_Part_cont
8      3   a2_Part_cont
9      5   a2_Part_cont
10     4   a2_Part_cont
11     2  a3_Cont_after
12     4  a3_Cont_after
13     5  a3_Cont_after
14     4  a3_Cont_after
15     1  a3_Cont_after
16     3     a4_No_cont
17     3     a4_No_cont
18     2     a4_No_cont
19     4     a4_No_cont
20     3     a4_No_cont
-----------------------
```

```
> print(multi_comp)

       $sem
       [1] 0.6855655

       $sed
       [1] 0.969536

       $MCT
       -------------
          medie test
       -------------
       M1   7.0    a
       M2   4.2    b
       M3   3.2    b
       M4   3.0    b
       -------------

       $Critical.Differences
       [1] 0.000000 2.055324 2.501724 2.773862

> print('Means with the same letter are not significant')

       [1] "Means with the same letter are not significant"


> summary(aov5)
```

```
        ---------------------------------------------
                    Df Sum Sq Mean Sq F value   Pr(>F)
        levels       3 50.950  16.983   7.227 0.002782 **
        Residuals   16 37.600   2.350
        ---------------------------------------------
        ---
        Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'
        0.1 ' ' 1

> print(Contrasts)

        -----------------
              G1 G2 G3 G4
        -----------------
        Psi_1  1 -1  0  0
        Psi_2  1  0 -1  0
        Psi_3  1  0  0 -1
        Psi_4  0  1 -1  0
        Psi_5  0  1  0 -1
        Psi_6  0  0  1 -1
        -----------------

> print(Contrast_Summary)

        -------------------------------------------------
          Contrast DF Contrast_SS Mean_Square     F_Value
        -------------------------------------------------
        1    Psi_1  1        19.6        19.6  8.34042553
        2    Psi_2  1        36.1        36.1 15.36170213
        3    Psi_3  1        40.0        40.0 17.02127660
        4    Psi_4  1         2.5         2.5  1.06382979
        5    Psi_5  1         3.6         3.6  1.53191489
        6    Psi_6  1         0.1         0.1  0.04255319
        -------------------------------------------------

        ---------------
                     Pr
        ---------------
        1  0.0107046712
        2  0.0012227709
        3  0.0007927012
        4  0.3176832781
        5  0.2336812046
        6  0.8391709477
        ---------------
```

10

ANOVA Two factor design: $\mathcal{A} \times \mathcal{B}$ or $\mathcal{S}(\mathcal{A} \times \mathcal{B})$

10.1 Cute Cued Recall

To illustrate the use of a two-factor design, consider a replication of an experiment by Tulving & Pearlstone (1966), in which 60 subjects were asked to learn lists of 12, 24 or 48 words (factor $\mathcal{A}$ with 3 levels). These words can be put in pairs by categories (for example, apple and orange can be grouped as “fruits”). Subjects were asked to learn these words, and the category name was shown at the same time as the words were presented. Subjects were told that they did not have to learn the category names. After a very short time, subjects were asked to recall the words. At that time half of the subjects were given the list of the category names, and the other half had to recall the words without the list of categories (factor $\mathcal{B}$ with 2 levels). The dependent variable is the number of words recalled by each subject. Note that both factors are fixed. The results obtained in the six experimental conditions used in this experiment are presented in Table 10.1.

	Factor $\mathcal{A}$					
Factor $\mathcal{B}$	a_1: 12 words		a_2: 24 words		a_3: 48 words	
	11	07	13	15	17	16
	09	12	18	13	20	23
b_1	13	11	19	09	22	19
Free Recall	09	10	13	08	13	20
	08	10	08	14	21	19
	12	10	13	14	32	30
	12	12	21	13	31	33
b_2	07	10	20	14	27	25
Cued Recall	09	07	15	16	30	25
	09	12	17	07	29	28

TABLE 10.1 Results of a replication of Tulving and Pearlstone’s experiment (1966). The dependent variable is the number of words recalled (see text for explanation).

10.1.1 [R] code

```
# ANOVA Two-factor between subjects, S(AxB)
# Cute Cued Recall; Tulving and Pearlstone

# We have 2 Factors, A & B, with 3 levels in Factor A and 2 levels
#    in Factor B.  Therefore there are 3 x 2 = 6 groups with 10
#   observations (subjects) per group.

# We collect the data for each level of Factor B across all the
#   levels of Factor A.

# Factor A: Length of Word List
#   (a1=12 words, a2=24 words, a3=48 words)
# Factor B: Type of Recall
#   (b1=Free Recall, b2= Cued Recall)

free_recall=c(11,13,17, 9,18,20, 13,19,22, 9,13,13, 8,8,21,
     7,15,16, 12,13,23, 11,9,19, 10,8,20, 10,14,19)
cued_recall=c(12,13,32, 12,21,31, 7,20,27, 9,15,30, 9,17,29,
     10,14,30, 12,13,33, 10,14,25, 7,16,25, 12,7,28)

# We now combine the observations into one long column (score).
score=c(free_recall,cued_recall)

# We now prepare the labels for the 3 x 2 x 10 scores according to
#   the factor levels:
#       Factor A --- 12 words 24 words 48 words, 12 words 24 words
#                    48 words, ... etc.
list_length=gl(3,1,2*3*10, labels=c("12 Words","24 Words","48
     Words"))

#       Factor B --- Free Recall Free Recall , Cued Recall Cued
#                    Recall  etc.
recall_type=gl(2,3*10,2*3*10, labels=c("Free Recall","Cued
      Recall"))

# We generate a second column (group), that identifies the group
#   for each score.
group=gl(2*3,10,2*3*10, labels=c("a1b1", "a2b1", "a3b1", "a1b2",
     "a2b2", "a3b2"))

# We now form a data frame with the dependent variable and the factors.
data=data.frame(score = score, Factor_A = factor(list_length),
     Factor_B = factor(recall_type), Group = group)

# We now define the contrasts
Linear=c(-1,0,1)
Quadratic=c(1,-2,1)
a1_vs_a2_a3=c(-2,1,1)
a2_vs_a3=c(0,1,-1)
AB_contrast=c(-2,2,1,-1,1,-1)

# We now perform the ANOVA on the data.
```

```
aov1=aov(score~list_length*recall_type, data=data)
interaction=list_length:recall_type

# We now organize the results
Df_Linear=summary(aov(score~list_length+recall_type+interaction,
     contrasts=list(list_length=make.contrasts(Linear))),split =
     list(list_length = list("Linear" = 1)))[[1]]$Df
Df_Quadratic=summary(aov(score~list_length+recall_type+
    interaction,contrasts= list(list_length=  make.contrasts(
     Quadratic))), split = list(list_length = list("Quadratic" =
     1)))[[1]]$Df
Df_a1_vs_a2_a3=summary(aov(score~list_length+recall_type+
     interaction, contrasts=list(list_length = make.contrasts(
     a1_vs_a2_a3))),split = list(list_length = list("a1_vs_a2_a3" =
     1)))[[1]]$Df
Df_a2_vs_a3=summary(aov(score~list_length+recall_type+
     interaction,contrasts = list(list_length = make.contrasts(
     a2_vs_a3))), split = list(list_length = list("a2_vs_a3" =
     1)))[[1]]$Df
Df_AB_contrast=summary(aov(score~list_length+recall_type+
     interaction,contrasts = list(interaction = make.contrasts(
     AB_contrast))),split = list(interaction = list("AB_contrast" =
     1)))[[1]]$Df
Df_Cont = data.frame(rbind(Df_Linear, Df_Quadratic,
     Df_a1_vs_a2_a3, Df_a2_vs_a3, Df_AB_contrast))

SS_Linear=summary(aov(score~list_length+recall_type+interaction,
     contrasts=list(list_length=make.contrasts(Linear))),split =
     list(list_length = list("Linear" = 1)))[[1]]$Sum
SS_Quadratic=summary(aov(score~list_length+recall_type+
     interaction, contrasts=list(list_length =
     make.contrasts(Quadratic))), split = list(list_length =
     list("Quadratic" = 1)))[[1]]$Sum
SS_a1_vs_a2_a3=summary(aov(score~list_length+recall_type+
     interaction,contrasts = list(list_length =
     make.contrasts(a1_vs_a2_a3))), split = list(list_length =
     list("a1_vs_a2_a3" = 1)))[[1]]$Sum
SS_a2_vs_a3 = summary(aov(score~list_length + recall_type +
     interaction,contrasts = list(list_length =
     make.contrasts(a2_vs_a3))), split = list(list_length =
     list("a2_vs_a3" = 1)))[[1]]$Sum
SS_AB_contrast = summary(aov(score~list_length + recall_type +
     interaction, contrasts = list(interaction =
     make.contrasts(AB_contrast))), split = list(interaction =
     list("AB_contrast" = 1)))[[1]]$Sum
SS_Cont = data.frame(rbind(SS_Linear, SS_Quadratic,
     SS_a1_vs_a2_a3, SS_a2_vs_a3, SS_AB_contrast))

MS_Linear = summary(aov(score~list_length + recall_type +
     interaction, contrasts = list(list_length =
     make.contrasts(Linear))), split = list(list_length =
     list("Linear" = 1)))[[1]]$Mean
MS_Quadratic = summary(aov(score~list_length + recall_type +
     interaction, contrasts = list(list_length =
```

```
    make.contrasts(Quadratic))), split = list(list_length =
    list("Quadratic" = 1)))[[1]]$Mean
MS_a1_vs_a2_a3 = summary(aov(score~list_length + recall_type +
    interaction, contrasts = list(list_length =
    make.contrasts(a1_vs_a2_a3))), split = list(list_length =
    list("a1_vs_a2_a3" = 1)))[[1]]$Mean
MS_a2_vs_a3 = summary(aov(score~list_length + recall_type +
    interaction, contrasts = list(list_length =
    make.contrasts(a2_vs_a3))), split = list(list_length =
    list("a2_vs_a3" = 1)))[[1]]$Mean
MS_AB_contrast = summary(aov(score~list_length + recall_type +
    interaction, contrasts = list(interaction =
    make.contrasts(AB_contrast))),split = list(interaction =
    list("AB_contrast" = 1)))[[1]]$Mean
MS_Cont=data.frame(rbind(MS_Linear, MS_Quadratic, MS_a1_vs_a2_a3,
    MS_a2_vs_a3, MS_AB_contrast))
F_Linear = summary(aov(score~list_length + recall_type +
    interaction, contrasts = list(list_length =
    make.contrasts(Linear))), split = list(list_length =
    list("Linear" = 1)))[[1]]$F
F_Quadratic = summary(aov(score~list_length + recall_type +
    interaction, contrasts = list(list_length =
    make.contrasts(Quadratic))), split = list(list_length =
    list("Quadratic" = 1)))[[1]]$F

F_a1_vs_a2_a3 = summary(aov(score~list_length + recall_type +
    interaction, contrasts = list(list_length =
    make.contrasts(a1_vs_a2_a3))), split = list(list_length =
    list("a1_vs_a2_a3" = 1)))[[1]]$F
F_a2_vs_a3 = summary(aov(score~list_length+recall_type +
    interaction,contrasts = list(list_length =
    make.contrasts(a2_vs_a3))), split = list(list_length =
    list("a2_vs_a3" = 1)))[[1]]$F
F_AB_contrast = summary(aov(score~list_length + recall_type +
    interaction, contrasts = list(interaction =
    make.contrasts(AB_contrast))), split = list(interaction =
    list("AB_contrast" = 1)))[[1]]$F
F_Cont = data.frame(rbind(F_Linear, F_Quadratic, F_a1_vs_a2_a3,
    F_a2_vs_a3, F_AB_contrast))

Pr_Linear = summary(aov(score~list_length + recall_type +
    interaction, contrasts = list(list_length =
    make.contrasts(Linear))), split = list(list_length =
    list("Linear" = 1)))[[1]]$Pr
Pr_Quadratic = summary(aov(score~list_length + recall_type +
    interaction, contrasts = list(list_length =
    make.contrasts(Quadratic))), split = list(list_length =
    list("Quadratic" = 1)))[[1]]$Pr
Pr_a1_vs_a2_a3 = summary(aov(score~list_length + recall_type +
    interaction, contrasts = list(list_length =
    make.contrasts(a1_vs_a2_a3))), split = list(list_length =
    list("a1_vs_a2_a3" = 1)))[[1]]$Pr
Pr_a2_vs_a3 = summary(aov(score~list_length + recall_type +
    interaction, contrasts = list(list_length =
```

```
     make.contrasts(a2_vs_a3))), split = list(list_length =
     list("a2_vs_a3" = 1)))[[1]]$Pr
Pr_AB_contrast = summary(aov(score~list_length + recall_type +
     interaction, contrasts = list(interaction =
     make.contrasts(AB_contrast))),split = list(interaction =
     list("AB_contrast" = 1)))[[1]]$Pr
Pr_Cont = data.frame(rbind(Pr_Linear, Pr_Quadratic,
     Pr_a1_vs_a2_a3, Pr_a2_vs_a3, Pr_AB_contrast))
Contrast_names=c("Linear", "Quadratic", "a1 vs a2 &a3", "a2 vs
     a3","AB")

Cont_mat=rbind("Linear"=Linear,"Quadratic"=Quadratic,"a1 vs a2
     &a3"=a1_vs_a2_a3,"a2 vs a3"=a2_vs_a3,"AB"=AB_contrast)
Contrasts=data.frame(G1=Cont_mat[,1], G2 = Cont_mat[,2], G3 =
     Cont_mat[,3], G4 = Cont_mat[,4], G5 = Cont_mat[,5], G6 =
     Cont_mat[,6])
Contrast_Summary=data.frame(Contrast = Contrast_names, DF =
     c(Df_Cont[1:4,2], Df_Cont[5,4]), Contrast_SS =
     c(SS_Cont[1:4,2], SS_Cont[5,4]), Mean_Square =
     c(MS_Cont[1:4,2], MS_Cont[5,4]), F_Value = c(F_Cont[1:4,2],
     F_Cont[5,4]), Pr=c(Pr_Cont[1:4,2], Pr_Cont[5,4]))

# Now we print the data and all the results
print(data)
print(model.tables(aov1,"means"),digits=3)
summary(aov1)
print(Contrasts)
print(Contrast_Summary)
```

10.1.2 [R] output

```
> # ANOVA Two-factor between subjects, S(AxB)
> # Cute Cued Recall; Tulving and Pearlstone

> # We have 2 Factors, A & B, with 3 levels in Factor A and 2 levels
> #    in Factor B.  Therefore there are 3 x 2 = 6 groups with 10
> #   observations (subjects) per group.

> # We collect the data for each level of Factor B across all the
> #   levels of Factor A.

> # Factor A: Length of Word List
> #   (a1=12 words, a2=24 words, a3=48 words)
> # Factor B: Type of Recall
> #   (b1=Free Recall, b2= Cued Recall)

> free_recall=c(11,13,17, 9,18,20, 13,19,22, 9,13,13, 8,8,21,
     7,15,16, 12,13,23, 11,9,19, 10,8,20, 10,14,19)
> cued_recall=c(12,13,32, 12,21,31, 7,20,27, 9,15,30, 9,17,29,
     10,14,30, 12,13,33, 10,14,25, 7,16,25, 12,7,28)

> # We now combine the observations into one long column (score).
> score=c(free_recall,cued_recall)
```

```
> # We now prepare the labels for the 3 x 2 x 10 scores according to
> #   the factor levels:
> #       Factor A --- 12 words 24 words 48 words, 12 words 24 words
> #                    48 words, ... etc.
> list_length=gl(3,1,2*3*10, labels=c("12 Words","24 Words","48
    Words"))

> #       Factor B --- Free Recall Free Recall , Cued Recall Cued
> #                    Recall  etc.
> recall_type=gl(2,3*10,2*3*10, labels=c("Free Recall","Cued
     Recall"))

> # We generate a second column (group), that identifies the group
> #   for each score.
> group=gl(2*3,10,2*3*10, labels=c("a1b1", "a2b1", "a3b1", "a1b2",
    "a2b2", "a3b2"))

> # We now form a data frame with the dependent variable and the factors.
> data=data.frame(score = score, Factor_A = factor(list_length),
    Factor_B = factor(recall_type), Group = group)

> # We now define the contrasts
> Linear=c(-1,0,1)
> Quadratic=c(1,-2,1)
> a1_vs_a2_a3=c(-2,1,1)
> a2_vs_a3=c(0,1,-1)
> AB_contrast=c(-2,2,1,-1,1,-1)

> # We now perform the ANOVA on the data.
> aov1=aov(score~list_length*recall_type, data=data)
> interaction=list_length:recall_type

> # We now organize the results
> Df_Linear=summary(aov(score~list_length+recall_type+interaction,
    contrasts=list(list_length=make.contrasts(Linear))),split =
    list(list_length = list("Linear" = 1)))[[1]]$Df
> Df_Quadratic=summary(aov(score~list_length+recall_type+
   interaction,contrasts= list(list_length=  make.contrasts(
    Quadratic))), split = list(list_length = list("Quadratic" =
    1)))[[1]]$Df
> Df_a1_vs_a2_a3=summary(aov(score~list_length+recall_type+
    interaction, contrasts=list(list_length = make.contrasts(
    a1_vs_a2_a3))),split = list(list_length = list("a1_vs_a2_a3" =
    1)))[[1]]$Df
> Df_a2_vs_a3=summary(aov(score~list_length+recall_type+
    interaction,contrasts = list(list_length = make.contrasts(
    a2_vs_a3))), split = list(list_length = list("a2_vs_a3" =
    1)))[[1]]$Df
> Df_AB_contrast=summary(aov(score~list_length+recall_type+
    interaction,contrasts = list(interaction = make.contrasts(
    AB_contrast))),split = list(interaction = list("AB_contrast" =
    1)))[[1]]$Df
> Df_Cont = data.frame(rbind(Df_Linear, Df_Quadratic,
```

```
   Df_a1_vs_a2_a3, Df_a2_vs_a3, Df_AB_contrast))

> SS_Linear=summary(aov(score~list_length+recall_type+interaction,
   contrasts=list(list_length=make.contrasts(Linear))),split =
   list(list_length = list("Linear" = 1)))[[1]]$Sum
> SS_Quadratic=summary(aov(score~list_length+recall_type+
   interaction, contrasts=list(list_length =
   make.contrasts(Quadratic))), split = list(list_length =
   list("Quadratic" = 1)))[[1]]$Sum
> SS_a1_vs_a2_a3=summary(aov(score~list_length+recall_type+
   interaction,contrasts = list(list_length =
   make.contrasts(a1_vs_a2_a3))), split = list(list_length =
   list("a1_vs_a2_a3" = 1)))[[1]]$Sum
> SS_a2_vs_a3 = summary(aov(score~list_length + recall_type +
   interaction,contrasts = list(list_length =
   make.contrasts(a2_vs_a3))), split = list(list_length =
   list("a2_vs_a3" = 1)))[[1]]$Sum
> SS_AB_contrast = summary(aov(score~list_length + recall_type +
   interaction, contrasts = list(interaction =
   make.contrasts(AB_contrast))), split = list(interaction =
   list("AB_contrast" = 1)))[[1]]$Sum
> SS_Cont = data.frame(rbind(SS_Linear, SS_Quadratic,
   SS_a1_vs_a2_a3, SS_a2_vs_a3, SS_AB_contrast))

> MS_Linear = summary(aov(score~list_length + recall_type +
   interaction, contrasts = list(list_length =
   make.contrasts(Linear))), split = list(list_length =
   list("Linear" = 1)))[[1]]$Mean
> MS_Quadratic = summary(aov(score~list_length + recall_type +
   interaction, contrasts = list(list_length =
   make.contrasts(Quadratic))), split = list(list_length =
   list("Quadratic" = 1)))[[1]]$Mean
> MS_a1_vs_a2_a3 = summary(aov(score~list_length + recall_type +
   interaction, contrasts = list(list_length =
   make.contrasts(a1_vs_a2_a3))), split = list(list_length =
   list("a1_vs_a2_a3" = 1)))[[1]]$Mean
> MS_a2_vs_a3 = summary(aov(score~list_length + recall_type +
   interaction, contrasts = list(list_length =
   make.contrasts(a2_vs_a3))), split = list(list_length =
   list("a2_vs_a3" = 1)))[[1]]$Mean
> MS_AB_contrast = summary(aov(score~list_length + recall_type +
   interaction, contrasts = list(interaction =
   make.contrasts(AB_contrast))),split = list(interaction =
   list("AB_contrast" = 1)))[[1]]$Mean
> MS_Cont=data.frame(rbind(MS_Linear, MS_Quadratic, MS_a1_vs_a2_a3,
   MS_a2_vs_a3, MS_AB_contrast))
> F_Linear = summary(aov(score~list_length + recall_type +
   interaction, contrasts = list(list_length =
   make.contrasts(Linear))), split = list(list_length =
   list("Linear" = 1)))[[1]]$F
> F_Quadratic = summary(aov(score~list_length + recall_type +
   interaction, contrasts = list(list_length =
   make.contrasts(Quadratic))), split = list(list_length =
   list("Quadratic" = 1)))[[1]]$F
```

```
> F_a1_vs_a2_a3 = summary(aov(score~list_length + recall_type +
    interaction, contrasts = list(list_length =
    make.contrasts(a1_vs_a2_a3))), split = list(list_length =
    list("a1_vs_a2_a3" = 1)))[[1]]$F
> F_a2_vs_a3 = summary(aov(score~list_length+recall_type +
    interaction,contrasts = list(list_length =
    make.contrasts(a2_vs_a3))), split = list(list_length =
    list("a2_vs_a3" = 1)))[[1]]$F
> F_AB_contrast = summary(aov(score~list_length + recall_type +
    interaction, contrasts = list(interaction =
    make.contrasts(AB_contrast))), split = list(interaction =
    list("AB_contrast" = 1)))[[1]]$F
> F_Cont = data.frame(rbind(F_Linear, F_Quadratic, F_a1_vs_a2_a3,
    F_a2_vs_a3, F_AB_contrast))

> Pr_Linear = summary(aov(score~list_length + recall_type +
    interaction, contrasts = list(list_length =
    make.contrasts(Linear))), split = list(list_length =
    list("Linear" = 1)))[[1]]$Pr
> Pr_Quadratic = summary(aov(score~list_length + recall_type +
    interaction, contrasts = list(list_length =
    make.contrasts(Quadratic))), split = list(list_length =
    list("Quadratic" = 1)))[[1]]$Pr
> Pr_a1_vs_a2_a3 = summary(aov(score~list_length + recall_type +
    interaction, contrasts = list(list_length =
    make.contrasts(a1_vs_a2_a3))), split = list(list_length =
    list("a1_vs_a2_a3" = 1)))[[1]]$Pr
> Pr_a2_vs_a3 = summary(aov(score~list_length + recall_type +
    interaction, contrasts = list(list_length =
    make.contrasts(a2_vs_a3))), split = list(list_length =
    list("a2_vs_a3" = 1)))[[1]]$Pr
> Pr_AB_contrast = summary(aov(score~list_length + recall_type +
    interaction, contrasts = list(interaction =
    make.contrasts(AB_contrast))),split = list(interaction =
    list("AB_contrast" = 1)))[[1]]$Pr
> Pr_Cont = data.frame(rbind(Pr_Linear, Pr_Quadratic,
    Pr_a1_vs_a2_a3, Pr_a2_vs_a3, Pr_AB_contrast))
> Contrast_names=c("Linear", "Quadratic", "a1 vs a2 &a3", "a2 vs
    a3","AB")

> Cont_mat=rbind("Linear"=Linear,"Quadratic"=Quadratic,"a1 vs a2
    &a3"=a1_vs_a2_a3,"a2 vs a3"=a2_vs_a3,"AB"=AB_contrast)
> Contrasts=data.frame(G1=Cont_mat[,1], G2 = Cont_mat[,2], G3 =
    Cont_mat[,3], G4 = Cont_mat[,4], G5 = Cont_mat[,5], G6 =
    Cont_mat[,6])
> Contrast_Summary=data.frame(Contrast = Contrast_names, DF =
    c(Df_Cont[1:4,2], Df_Cont[5,4]), Contrast_SS =
    c(SS_Cont[1:4,2], SS_Cont[5,4]), Mean_Square =
    c(MS_Cont[1:4,2], MS_Cont[5,4]), F_Value = c(F_Cont[1:4,2],
    F_Cont[5,4]), Pr=c(Pr_Cont[1:4,2], Pr_Cont[5,4]))

> # Now we print the data and all the results
> print(data)
```

	Score	Factor_A	Factor_B	Group
1	11	12 Words	Free Recall	a1b1
2	13	24 Words	Free Recall	a1b1
3	17	48 Words	Free Recall	a1b1
4	9	12 Words	Free Recall	a1b1
5	18	24 Words	Free Recall	a1b1
6	20	48 Words	Free Recall	a1b1
7	13	12 Words	Free Recall	a1b1
8	19	24 Words	Free Recall	a1b1
9	22	48 Words	Free Recall	a1b1
10	9	12 Words	Free Recall	a1b1
11	13	24 Words	Free Recall	a2b1
12	13	48 Words	Free Recall	a2b1
13	8	12 Words	Free Recall	a2b1
14	8	24 Words	Free Recall	a2b1
15	21	48 Words	Free Recall	a2b1
16	7	12 Words	Free Recall	a2b1
17	15	24 Words	Free Recall	a2b1
18	16	48 Words	Free Recall	a2b1
19	12	12 Words	Free Recall	a2b1
20	13	24 Words	Free Recall	a2b1
21	23	48 Words	Free Recall	a3b1
22	11	12 Words	Free Recall	a3b1
23	9	24 Words	Free Recall	a3b1
24	19	48 Words	Free Recall	a3b1
25	10	12 Words	Free Recall	a3b1
26	8	24 Words	Free Recall	a3b1
27	20	48 Words	Free Recall	a3b1
28	10	12 Words	Free Recall	a3b1
29	14	24 Words	Free Recall	a3b1
30	19	48 Words	Free Recall	a3b1
31	12	12 Words	Cued Recall	a1b2
32	13	24 Words	Cued Recall	a1b2
33	32	48 Words	Cued Recall	a1b2
34	12	12 Words	Cued Recall	a1b2
35	21	24 Words	Cued Recall	a1b2
36	31	48 Words	Cued Recall	a1b2
37	7	12 Words	Cued Recall	a1b2
38	20	24 Words	Cued Recall	a1b2
39	27	48 Words	Cued Recall	a1b2
40	9	12 Words	Cued Recall	a1b2
41	15	24 Words	Cued Recall	a2b2
42	30	48 Words	Cued Recall	a2b2
43	9	12 Words	Cued Recall	a2b2
44	17	24 Words	Cued Recall	a2b2
45	29	48 Words	Cued Recall	a2b2
46	10	12 Words	Cued Recall	a2b2
47	14	24 Words	Cued Recall	a2b2
48	30	48 Words	Cued Recall	a2b2
49	12	12 Words	Cued Recall	a2b2
50	13	24 Words	Cued Recall	a2b2

```
51    33  48 Words  Cued Recall  a3b2
52    10  12 Words  Cued Recall  a3b2
53    14  24 Words  Cued Recall  a3b2
54    25  48 Words  Cued Recall  a3b2
55     7  12 Words  Cued Recall  a3b2
56    16  24 Words  Cued Recall  a3b2
57    25  48 Words  Cued Recall  a3b2
58    12  12 Words  Cued Recall  a3b2
59     7  24 Words  Cued Recall  a3b2
60    28  48 Words  Cued Recall  a3b2
-------------------------------------
```

```
> print(model.tables(aov1,"means"),digits=3)
```

```
Tables of means
Grand mean
16

List_length
-----------------------------
12 Words  24 Words  48 Words
-----------------------------
      10        14        24
-----------------------------

Recall_type
------------------------
Free Recall Cued Recall
------------------------
         14          18
------------------------

List_length:Recall_type
-----------------------------------
                 Recall_type
            -----------------------
List_length Free Recall Cued Recall
-----------------------------------
   12 Words          10          10
   24 Words          13          15
   48 Words          19          29
-----------------------------------
```

```
> summary(aov1)
```

```
-----------------------------------------------------------
                        Df Sum Sq Mean Sq F value    Pr(>F)
-----------------------------------------------------------
List_length              2   2080    1040 115.556 < 2.2e-16 ***
Recall_type              1    240     240  26.667 3.577e-06 ***
List_length:Recall_type  2    280     140  15.556 4.624e-06 ***
Residuals               54    486       9
-----------------------------------------------------------
---
```

```
    Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

> print(Contrasts)

      -----------------------------
                   G1 G2 G3 G4 G5 G6
      -----------------------------
      Linear       -1  0  1 -1  0  1
      Quadratic     1 -2  1  1 -2  1
      a1 vs a2 &a3 -2  1  1 -2  1  1
      a2 vs a3      0  1 -1  0  1 -1
      AB           -2  2  1 -1  1 -1
      -----------------------------

> print(Contrast_Summary)

    ----------------------------------------------------------------
          Contrast DF Contrast_SS Mean_Square   F_Value           Pr
    ----------------------------------------------------------------
    1       Linear  1        1960        1960 217.77778 1.351403e-20
    2    Quadratic  1         120         120  13.33333 5.896216e-04
    3 a1 vs a2 &a3  1        1080        1080 120.00000 2.459198e-15
    4     a2 vs a3  1        1000        1000 111.11111 1.024991e-14
    5           AB  1         120         120  13.33333 5.896216e-04
    ----------------------------------------------------------------
```

10.1.3 ANOVA table

Source	*df*	*SS*	*MS*	*F*	Pr(*F*)
$\mathcal{A}$	2	$2,080.00$	$1,040.00$	115.56	$< .000,001$
$\mathcal{B}$	1	240.00	240.00	26.67	$.000,007$
$\mathcal{AB}$	2	280.00	140.00	15.56	$.000,008$
$\mathcal{S}(\mathcal{AB})$	54	486.00	9.00		
Total	59	$3,086.00$			

TABLE 10.2 ANOVA Table for Tulving and Pearlstone’s (1966) experiment.

10.2 Projective Tests and Test Administrators

For purposes of illustrating the [R] code for an $\mathcal{A} \times \mathcal{B}$ design with both factors random (Model II), consider the following results from a hypothetical experiment on projective testing. The researchers were interested in the effects of using different test administrators.

Note that only the *F* values obtained using the random option in the [R] code are valid.

Order ($\mathcal{B}$)	Test Administrators ($\mathcal{A}$) 1	2	3	4	Means
I	127	117	111	108	
	121	109	111	100	113
II	117	113	111	100	
	109	113	101	92	107
III	107	108	99	92	
	101	104	91	90	99
IV	98	95	95	87	
	94	93	89	77	91
V	97	96	89	89	
	89	92	83	85	90
Means	106	104	98	92	100

10.2.1 [R] code

10.2.2 [R] output

```
> # ANOVA Two-factor between subjects, S(AxB)
> # Projective Tests and Test Administrators

> # We have 2 Factors, A & B, with 4 levels in Factor A and
> #    5 levels in Factor B.
> #    Therefore there are 4 x 5 = 20 groups with 2 observations
> #    (subjects) per group.

> # Factor A: Test Administrator
> #   (a1=1,a2=2,a3=3,a4=4)

> # Factor B: Order
> #   (b1=I, b2= II, b3= III, b4=IV, b5=V)

> # We collect the data for each level of Factor B across all the
> # levels of Factor A.
> I=c(127,117,111,108, 121,109,111,100)
> II=c(117,113, 111, 100, 109,113,101,92)
> III=c(107,108,99,92, 101,104,91,90)
> IV=c(98,95,95,87, 94,93,89,77)
> V=c(97,96,89,89, 89,92,83,85)

> # We now combine the observations into one long column (score).
> score=c(I,II,III,IV,V)

> # We now prepare the labels for the 4x5x2 scores according to
> #   the factor levels:
```

```
> # Admin_1 Admin_2 Admin_3 Admin_4, Admin_1 Admin_2 Admin_3
> #   Admin_4.......etc for Factor A
> Test_Admin=gl(4,1,5*4*2, labels = c("Admin_1", "Admin_2",
    "Admin_3", "Admin_4"))

> # I I I......., II II ......,III III ....,IV IV ....., V V.....
> #   etc for Factor B.

> Order=gl(5,4*2,5*4*2, labels=c("I","II","III","IV","V"))

> # We now form a data frame with the dependent variable and the
> # factors.
> data = data.frame(score = score, Factor_A = factor(Test_Admin),
    Factor_B=factor(Order))

> # We now perform the ANOVA on the data.
> aov1=aov(score~Test_Admin*Order, data=data)

> # Model III when both A and B are random
> aov2 = aov(score~Test_Admin + Order + Error(Test_Admin:Order),
    data = data)

> # We now print the data and all the results
> print(data)
```

```
 -------------------------
    Score Factor_A Factor_B
 -------------------------
 1    127  Admin_1        I
 2    117  Admin_2        I
 3    111  Admin_3        I
 4    108  Admin_4        I
 5    121  Admin_1        I
 6    109  Admin_2        I
 7    111  Admin_3        I
 8    100  Admin_4        I
 9    117  Admin_1       II
 10   113  Admin_2       II
 11   111  Admin_3       II
 12   100  Admin_4       II
 13   109  Admin_1       II
 14   113  Admin_2       II
 15   101  Admin_3       II
 16    92  Admin_4       II
 17   107  Admin_1      III
 18   108  Admin_2      III
 19    99  Admin_3      III
 20    92  Admin_4      III
 21   101  Admin_1      III
 22   104  Admin_2      III
 23    91  Admin_3      III
 24    90  Admin_4      III
 25    98  Admin_1       IV
 26    95  Admin_2       IV
```

```
27    95  Admin_3       IV
28    87  Admin_4       IV
29    94  Admin_1       IV
30    93  Admin_2       IV
31    89  Admin_3       IV
32    77  Admin_4       IV
33    97  Admin_1        V
34    96  Admin_2        V
35    89  Admin_3        V
36    89  Admin_4        V
37    89  Admin_1        V
38    92  Admin_2        V
39    83  Admin_3        V
40    85  Admin_4        V
--------------------------
```

```
> summary(aov1)
```

```
----------------------------------------------------
                 Df Sum Sq Mean Sq F value    Pr(>F)
----------------------------------------------------
Test_Admin        3   1200     400      20 3.102e-06 ***
Order             4   3200     800      40 2.836e-09 ***
Test_Admin:Order 12    240      20       1    0.4827
Residuals        20    400      20
----------------------------------------------------
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’
0.1 ’ ’ 1
```

```
> print(model.tables(aov1,"means"),digits=3)
```

```
Tables of means
Grand mean
       100

Test_Admin
-------------------------------
Admin_1 Admin_2 Admin_3 Admin_4
-------------------------------
    106     104      98      92
-------------------------------

Order
-------------------
  I  II III  IV   V
-------------------
113 107  99  91  90
-------------------

Test_Admin:Order
---------------------------------
                     Order
            ---------------------
```

```
Test_Admin    I   II  III  IV   V
---------------------------------
   Admin_1  124  113  104  96  93
   Admin_2  113  113  106  94  94
   Admin_3  111  106   95  92  86
   Admin_4  104   96   91  82  87
---------------------------------
```

```
> summary(aov2)

       Error: Test_Admin:Order
       ----------------------------------------------
                  Df Sum Sq Mean Sq F value    Pr(>F)
       ----------------------------------------------
       Test_Admin  3   1200     400      20 5.819e-05 ***
       Order       4   3200     800      40 7.590e-07 ***
       Residuals  12    240      20
       ----------------------------------------------
       ---
       Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’
       0.1 ’ ’ 1

       Error: Within
       ------------------------------------------
                 Df Sum Sq Mean Sq F value Pr(>F)
       ------------------------------------------
       Residuals 20    400      20
       ------------------------------------------
```

10.2.3 ANOVA table

Source	*df*	*SS*	*MS*	*F*	Pr(*F*)
$\mathcal{A}$	3	$1,200.00$	400.00	20.00**	.00008
$\mathcal{B}$	4	$3,200.00$	800.00	40.00**	.00000
$\mathcal{AB}$	12	240.00	20.00	1.00 *ns*	.48284
$\mathcal{S}(\mathcal{AB})$	20	400.00	20.00		
Total	39	$5,040.00$			

11 ANOVA One Factor Repeated Measures, $\mathcal{S} \times \mathcal{A}$

11.1 $\mathcal{S} \times \mathcal{A}$ design

For illustrative purposes we designed a hypothetical experiment using a within-subjects design. The independent variable consists of 4 levels and the size of the experimental group is 5 (*i.e.,* 5 subjects participated in the experiment, the results are presented in Table 11.1:

11.1.1 [R] code

```
# ANOVA One-factor within subjects, SxA
# Numerical Example

# We have 1 Factor, A, with 4 levels in Factor A and 4
#    subjects.
# The 4 levels of Factor A are: a1, a2, a3, a4.
# Therefore there are 4 groups with the same 5 observations
# (subjects) per group.

# We collect the data for each subject for all levels of
# Factor A.
sub_1=c(5,4,1,8)
sub_2=c(7,4,1,10)
```

	Levels of the independent variable				
Subjects	$\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{a}_3$	$\mathbf{a}_4$	$M_{.s}$
s_1	5	4	1	8	4.50
s_2	7	4	1	10	5.50
s_3	12	9	8	16	11.25
s_4	4	9	6	9	7.00
s_5	8	9	5	13	8.75
$M_{a.}$	7.20	7.00	4.20	11.20	$M_{..} = 7.40$

TABLE 11.1 A numerical example of an $\mathcal{S} \times \mathcal{A}$ design.

```
sub_3=c(12,9,8,16)
sub_4=c(4,9,6,9)
sub_5=c(8,9,5,13)

# We now combine the observations into one long column (score).
score=c(sub_1,sub_2,sub_3,sub_4,sub_5)

# We now prepare the labels for the 4x5 scores according to the
# factor levels:
# a1 a2 a3 a4, a1 a2 a3 a4........etc for Factor A
Fact_A=gl(4,1,5*4*1, labels=c("a1","a2","a3","a4"))

# sub_1 sub_1......., sub_2 sub_2......,sub_3 sub_3 ....,sub_4
#     sub_4 ....., sub_5 sub_5.....etc for Subjects
Subject=gl(5,4*1,5*4*1,  labels=c("sub_1", "sub_2", "sub_3",
     "sub_4", "sub_5"))

# We now form a data frame with the dependent variable and the factors.
data=data.frame(score = score,Factor_A = factor(Fact_A),
     Subject = factor(Subject))

# Anova when "Subject" is considered as a random factor.
aov1=aov(score~Fact_A+Error(Subject),data=data)

# We now print the data and all the results

print(data)
summary(aov1)
print(model.tables(aov(score~Fact_A+Subject),"means"),
```

11.1.2 [R] output

```
> # ANOVA One-factor within subjects, SxA
> # Numerical Example

> # We have 1 Factor, A, with 4 levels in Factor A and 4
> #    subjects.
> # The 4 levels of Factor A are: a1, a2, a3, a4.
> # Therefore there are 4 groups with the same 5 observations
> # (subjects) per group.

> # We collect the data for each subject for all levels of
> # Factor A.
> sub_1=c(5,4,1,8)
> sub_2=c(7,4,1,10)
> sub_3=c(12,9,8,16)
> sub_4=c(4,9,6,9)
> sub_5=c(8,9,5,13)

> # We now combine the observations into one long column (score).
> score=c(sub_1,sub_2,sub_3,sub_4,sub_5)

> # We now prepare the labels for the 4x5 scores according to the
```

```
> # factor levels:
> # a1 a2 a3 a4, a1 a2 a3 a4........etc for Factor A
> Fact_A=gl(4,1,5*4*1, labels=c("a1","a2","a3","a4"))

> # sub_1 sub_1......., sub_2 sub_2......,sub_3 sub_3 ....,sub_4
> #     sub_4 ....., sub_5 sub_5.....etc for Subjects
> Subject=gl(5,4*1,5*4*1,  labels=c("sub_1", "sub_2", "sub_3",
    "sub_4", "sub_5"))

> # We now form a data frame with the dependent variable and the factors.
> data=data.frame(score = score,Factor_A = factor(Fact_A),
    Subject = factor(Subject))

> # Anova when "Subject" is considered as a random factor.
> aov1=aov(score~Fact_A+Error(Subject),data=data)

> # We now print the data and all the results

> print(data)

       -------------------------
          score Factor_A Subject
       -------------------------
       1      5       a1   sub_1
       2      4       a2   sub_1
       3      1       a3   sub_1
       4      8       a4   sub_1
       5      7       a1   sub_2
       6      4       a2   sub_2
       7      1       a3   sub_2
       8     10       a4   sub_2
       9     12       a1   sub_3
       10     9       a2   sub_3
       11     8       a3   sub_3
       12    16       a4   sub_3
       13     4       a1   sub_4
       14     9       a2   sub_4
       15     6       a3   sub_4
       16     9       a4   sub_4
       17     8       a1   sub_5
       18     9       a2   sub_5
       19     5       a3   sub_5
       20    13       a4   sub_5
       -------------------------

> summary(aov1)

       Error: Subject
       -------------------------------------------
                 Df  Sum Sq Mean Sq F value Pr(>F)
       -------------------------------------------
       Residuals  4 115.300  28.825
       -------------------------------------------
              Error: Within
```

```
    -------------------------------------------
              Df  Sum Sq Mean Sq F value Pr(>F)
    -------------------------------------------
    Fact_A     3 124.400  41.467  14.177  3e-04 ***
    Residuals 12  35.100   2.925
    -------------------------------------------
    ---
    Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05
    '.' 0.1 ' ' 1

> print(model.tables(aov(score~Fact_A+Subject),"means"),
      digits=3)

    Tables of means
    Grand mean
           7.4

    Fact_A
    -------------------
      a1   a2   a3   a4
    -------------------
     7.2  7.0  4.2 11.2
    -------------------

    Subject
    -----------------------------
    sub_1 sub_2 sub_3 sub_4 sub_5
    -----------------------------
     4.50  5.50 11.25  7.00  8.75
    ----------------------------
```

11.2 Drugs and reaction time

In a psychopharmalogical experiment, we want to test the effect of two types of amphetamine-like drugs on latency performing a motor task. In order to control for any potential sources of variation due to individual reactions to amphetamines, the same six subjects were used in the three conditions of the experiment: Drug A, Drug B, and Placebo. The dependent variable is the reaction time measured in msec. The data from the experiment are presented in Table 11.2 on the facing page:

11.2.1 [R] code

```
# ANOVA One-factor within subjects, SxA
# Drugs and Reaction Time

# We have 1 Factor, A, with 3 levels in Factor A and 6
# subjects.
```

Subject	Drug A	Placebo	Drug B	Total
	Experimental Conditions			
s_1	124.00	108.00	104.00	336.00
s_2	105.00	107.00	100.00	312.00
s_3	107.00	90.00	100.00	297.00
s_4	109.00	89.00	93.00	291.00
s_5	94.00	105.00	89.00	288.00
s_6	121.00	71.00	84.00	276.00
Total	660.00	570.00	570.00	1,800.00

TABLE 11.2 Results of a fictitious hypothetical experiment illustrating the computational routine for a $S \times A$ design.

```
# The 4 levels of Factor A are: Drug A, Placebo, Drug B.
# Therefore there are 3 groups with the same 5 observations
# (subjects) per group.

# We collect the data for each subjects for all levels of
# Factor A.
sub_1=c(124,108,104)
sub_2=c(105,107,100)
sub_3=c(107,90,100)
sub_4=c(109,89,93)
sub_5=c(94,105,89)
sub_6=c(121,71,84)

# We now combine the observations into one long column (score).
score=c(sub_1,sub_2,sub_3,sub_4,sub_5,sub_6)

# We now prepare the labels for the 4x5 scores according to the factor levels:
# Drug_A Placebo Drub_B, Drug_A Placebo Drug_B........etc for
# Factor A
Drug=gl(3,1,6*3*1, labels=c("Drug_A","Placebo","Drug_B"))

# sub_1 sub_1......., sub_2 sub_2......,sub_3 sub_3 ....,sub_4
# sub_4 ....., sub_5 sub_5....., sub_6 sub_6 etc for Factor B.
Subject=gl(6,3*1,6*3*1, labels=c("sub _1", "sub_2", "sub_3",
     "sub_4", "sub_5", "sub_6"))

# We now form a data frame with the dependent variable and the
# factors.
data = data.frame(score = score, Drug = factor(Drug), Subject =
     factor(Subject))

# Anova when "Subject" is considered as a random factor.
aov1=aov(score~Drug+Error(Subject),data=data)

# We now print the data and all the results
print(data)
summary(aov1)
```

```
print(model.tables(aov(score~Drug+Subject),"means"),digits=3)
```

11.2.2 [R] output

```
> # ANOVA One-factor within subjects, SxA
> # Drugs and Reaction Time

> # We have 1 Factor, A, with 3 levels in Factor A and 6
> # subjects.

> # The 4 levels of Factor A are: Drug A, Placebo, Drug B.
> # Therefore there are 3 groups with the same 5 observations
> # (subjects) per group.

> # We collect the data for each subjects for all levels of
> # Factor A.
> sub_1=c(124,108,104)
> sub_2=c(105,107,100)
> sub_3=c(107,90,100)
> sub_4=c(109,89,93)
> sub_5=c(94,105,89)
> sub_6=c(121,71,84)

> # We now combine the observations into one long column (score).
> score=c(sub_1,sub_2,sub_3,sub_4,sub_5,sub_6)

> # We now prepare the labels for the 4x5 scores according to the factor levels:
> # Drug_A Placebo Drub_B, Drug_A Placebo Drug_B........etc for
> # Factor A
> Drug=gl(3,1,6*3*1, labels=c("Drug_A","Placebo","Drug_B"))

> # sub_1 sub_1......., sub_2 sub_2......,sub_3 sub_3 ....,sub_4
> # sub_4 ....., sub_5 sub_5....., sub_6 sub_6 etc for Factor B.
> Subject=gl(6,3*1,6*3*1, labels=c("sub _1", "sub_2", "sub_3",
    "sub_4", "sub_5", "sub_6"))

> # We now form a data frame with the dependent variable and the
> # factors.
> data = data.frame(score = score, Drug = factor(Drug), Subject =
    factor(Subject))

> # Anova when "Subject" is considered as a random factor.
> aov1=aov(score~Drug+Error(Subject),data=data)

> # We now print the data and all the results
> print(data)
```

	score	Drug	Subject
1	124	Drug_A	sub _1
2	108	Placebo	sub _1
3	104	Drug_B	sub _1

```
4    105  Drug_A   sub_2
5    107 Placebo   sub_2
6    100  Drug_B   sub_2
7    107  Drug_A   sub_3
8     90 Placebo   sub_3
9    100  Drug_B   sub_3
10   109  Drug_A   sub_4
11    89 Placebo   sub_4
12    93  Drug_B   sub_4
13    94  Drug_A   sub_5
14   105 Placebo   sub_5
15    89  Drug_B   sub_5
16   121  Drug_A   sub_6
17    71 Placebo   sub_6
18    84  Drug_B   sub_6
------------------------

> summary(aov1)

       Error: Subject
       ------------------------------------------
                 Df Sum Sq Mean Sq F value Pr(>F)
       ------------------------------------------
       Residuals  5    750     150
       ------------------------------------------

       Error: Within
       -------------------------------------------
                 Df Sum Sq Mean Sq F value  Pr(>F)
       -------------------------------------------
       Drug       2    900     450    3.75 0.06093 .
       Residuals 10   1200     120
       -------------------------------------------
       ---
       Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
        ' ' 1
       > print(model.tables(aov(score~Drug+Subject),"means"),digits=3)


       Tables of means
       Grand mean
              100

       Drug
       -----------------------
        Drug_A Placebo  Drug_B
       -----------------------
           110      95      95
       -----------------------

       Subject
       -----------------------------------------
       sub _1  sub_2  sub_3  sub_4  sub_5  sub_6
       -----------------------------------------
```

```
    112    104     99     97     96     92
   ----------------------------------------
```

11.2.3 ANOVA table

The final results are presented in the analysis of variance table:

Source	*df*	*SS*	*MS*	*F*	Pr(*F*)
$\mathcal{A}$	2	900.00	450.00	3.75	.060
$\mathcal{S}$	5	750.00	150.00		
$\mathcal{AS}$	10	1,200.00	120.00		
Total	17	2,850.00			

TABLE 11.3 ANOVA Table for the Drugs and Reaction Time experiment.

11.3 Proactive Interference

In an experiment on proactive interference, subjects were asked to learn a list of ten pairs of words. Two days later they were asked to recall these words. Once they finished recalling this first list, they were asked to learn a second list of ten pairs of words which they will be asked to recall after a new delay of two days. Recall of the second list was followed by a third list and so on until they learned and recalled six lists. The independent variable is the rank of the list in the learning sequence (first list, second list, ... , sixth list). The dependent variable is the number of words correctly recalled. The authors of this experiment predict that recall performance will decrease as a function of the rank of the lists (this effect is called "proactive interference"). The data are presented in Table 11.4.

11.3.1 [R] code

```
# ANOVA One-factor within subjects, SxA
# Proactive Interference

# We have 1 Factor, A (Rank), with 6 levels in Factor A and 8
# subjects.
# The 6 levels of Factor A are: rank_1, rank_2, rank_3, rank_4,
# rank_5, rank_6
# Therefore there are 6 groups with the same 8 observations
# (subjects) per group.

# We collect the data for each subjects for all levels of
```

	Rank of the list						
Subjects	1	2	3	4	5	6	Total
s_1	17	13	12	12	11	11	76
s_2	14	18	13	18	11	12	86
s_3	17	16	13	11	15	14	86
s_4	18	16	11	10	12	10	77
s_5	17	12	13	10	11	13	76
s_6	16	13	13	11	11	11	75
s_7	14	12	10	10	10	10	66
s_8	16	17	15	11	13	11	83
Total	129	117	100	93	94	92	625

TABLE 11.4 Results of an experiment on the effects of proactive interference on memory.

```
# Factor A.
sub_1=c(17,13,12,12,11,11)
sub_2=c(14,18,13,18,11,12)
sub_3=c(17,16,13,11,15,14)
sub_4=c(18,16,11,10,12,10)
sub_5=c(17,12,13,10,11,13)
sub_6=c(16,13,13,11,11,11)
sub_7=c(14,12,10,10,10,10)
sub_8=c(16,17,15,11,13,11)

# We now combine the observations into one long column (score).
score=c(sub_1,sub_2,sub_3,sub_4,sub_5,sub_6,sub_7,sub_8)

# We now prepare the labels for the 6x8 scores according to the
# factor levels:
# rank_1 rank_2 rank_3 rank_4 rank_5 rank_6........etc for
# Factor A
Rank=gl(6,1,8*6*1, labels = c("rank_1", "rank_2", "rank_3", "rank4", "rank_5", "rank_6"))

# sub_1 sub_1......., sub_2 sub_2......,sub_3 sub_3 ....,sub_4
# sub_4 ....., sub_5 sub_5.....etc for Subjects

Subject=gl(8,6*1,8*6*1, labels=c("sub _1", "sub_2", "sub_3",
     "sub_4", "sub_5", "sub_6", "sub_7", "sub_8"))

# We now form a data frame with the dependent variable and the
# factors.
data = data.frame(score = score, Rank = factor(Rank), Subject =
     factor(Subject))

# Anova when "Subject" is considered as a random factor.
aov1=aov(score~Rank+Error(Subject),data=data)

# We now print the data and all the results
print(data)
```

```
summary(aov1)
print(model.tables(aov(score ~ Rank + Subject, data =
     data),"means"),digits=3)
```

11.3.2 [R] output

```
> # ANOVA One-factor within subjects, SxA
> # Proactive Interference

> # We have 1 Factor, A (Rank), with 6 levels in Factor A and 8
> # subjects.
> # The 6 levels of Factor A are: rank_1, rank_2, rank_3, rank_4,
> # rank_5, rank_6
> # Therefore there are 6 groups with the same 8 observations
> # (subjects) per group.

> # We collect the data for each subjects for all levels of
> # Factor A.
> sub_1=c(17,13,12,12,11,11)
> sub_2=c(14,18,13,18,11,12)
> sub_3=c(17,16,13,11,15,14)
> sub_4=c(18,16,11,10,12,10)
> sub_5=c(17,12,13,10,11,13)
> sub_6=c(16,13,13,11,11,11)
> sub_7=c(14,12,10,10,10,10)
> sub_8=c(16,17,15,11,13,11)

> # We now combine the observations into one long column (score).
> score=c(sub_1,sub_2,sub_3,sub_4,sub_5,sub_6,sub_7,sub_8)

> # We now prepare the labels for the 6x8 scores according to the
> # factor levels:
> # rank_1 rank_2 rank_3 rank_4 rank_5 rank_6........etc for
> # Factor A
> Rank=gl(6,1,8*6*1, labels = c("rank_1", "rank_2", "rank_3", "rank4", "rank_5", "rank_6"))

> # sub_1 sub_1......., sub_2 sub_2......,sub_3 sub_3 ....,sub_4
> # sub_4 ....., sub_5 sub_5.....etc for Subjects

> Subject=gl(8,6*1,8*6*1, labels=c("sub _1", "sub_2", "sub_3",
     "sub_4", "sub_5", "sub_6", "sub_7", "sub_8"))

> # We now form a data frame with the dependent variable and the
> # factors.
> data = data.frame(score = score, Rank = factor(Rank), Subject =
     factor(Subject))

> # Anova when "Subject" is considered as a random factor.
> aov1=aov(score~Rank+Error(Subject),data=data)

> # We now print the data and all the results
> print(data)
```

```
-----------------------
   score   Rank Subject
-----------------------
1      17 rank_1  sub _1
2      13 rank_2  sub _1
3      12 rank_3  sub _1
4      12  rank4  sub _1
5      11 rank_5  sub _1
6      11 rank_6  sub _1
7      14 rank_1   sub_2
8      18 rank_2   sub_2
9      13 rank_3   sub_2
10     18  rank4   sub_2
11     11 rank_5   sub_2
12     12 rank_6   sub_2
13     17 rank_1   sub_3
14     16 rank_2   sub_3
15     13 rank_3   sub_3
16     11  rank4   sub_3
17     15 rank_5   sub_3
18     14 rank_6   sub_3
19     18 rank_1   sub_4
20     16 rank_2   sub_4
21     11 rank_3   sub_4
22     10  rank4   sub_4
23     12 rank_5   sub_4
24     10 rank_6   sub_4
25     17 rank_1   sub_5
26     12 rank_2   sub_5
27     13 rank_3   sub_5
28     10  rank4   sub_5
29     11 rank_5   sub_5
30     13 rank_6   sub_5
31     16 rank_1   sub_6
32     13 rank_2   sub_6
33     13 rank_3   sub_6
34     11  rank4   sub_6
35     11 rank_5   sub_6
36     11 rank_6   sub_6
37     14 rank_1   sub_7
38     12 rank_2   sub_7
39     10 rank_3   sub_7
40     10  rank4   sub_7
41     10 rank_5   sub_7
42     10 rank_6   sub_7
43     16 rank_1   sub_8
44     17 rank_2   sub_8
45     15 rank_3   sub_8
46     11  rank4   sub_8
47     13 rank_5   sub_8
48     11 rank_6   sub_8
-----------------------
```

```
> summary(aov1)
```

```
Error: Subject
------------------------------------------
          Df Sum Sq Mean Sq F value Pr(>F)
------------------------------------------
Residuals  7 52.479   7.497
------------------------------------------
       Error: Within
----------------------------------------------
          Df  Sum Sq Mean Sq F value    Pr(>F)
----------------------------------------------
Rank       5 146.854  29.371  10.316 3.875e-06 ***
Residuals 35  99.646   2.847
----------------------------------------------
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
```

```
> print(model.tables(aov(score ~ Rank + Subject, data =
     data),"means"),digits=3)
```

```
Tables of means
Grand mean
  13.02083

Rank
-----------------------------------------
rank_1 rank_2 rank_3  rank4 rank_5 rank_6
-----------------------------------------
 16.13  14.63  12.50  11.63  11.75  11.50
-----------------------------------------

Subject
-------------------------------------------------------
sub _1  sub_2  sub_3  sub_4  sub_5  sub_6  sub_7  sub_8
-------------------------------------------------------
 12.67  14.33  14.33  12.83  12.67  12.50  11.00  13.83
-------------------------------------------------------
```

11.3.3 ANOVA table

The results from this experiment are presented in the analysis of variance table.

Source	*df*	*SS*	*MS*	*F*	P(*F*)
$\mathcal{A}$	5	146.85	29.37	10.32^{**}	.000, 005
$\mathcal{S}$	7	52.48	7.50		
$\mathcal{AS}$	35	99.65	2.85		
Total	47	21, 806.50			

12
Two factors repeated measures, $\mathcal{S} \times \mathcal{A} \times \mathcal{B}$

12.1 Plungin'

What follows is a replication of Godden and Baddeley's (1975) experiment to show the effects of context on memory. Godden and Baddeley's hypothesis was that memory should be better when the conditions at test are more similar to the conditions experienced during learning. To operationalize this idea, Godden and Baddeley decided to use a very particular population: deep-sea divers. The divers were asked to learn a list of 40 unrelated words either on the beach or under about 10 feet of water, The divers were then tested either on the beach or undersea. The divers needed to be tested in both environments in order to make sure that any effect observed could not be attributed to a global effect of one of the environments. The rationale behind using divers was twofold. The first reason was practical: is it worth designing training programs on dry land for divers if they are not able to recall undersea what they have learned? There is strong evidence, incidently, that the problem is real. The second reason was more akin to good principles of experimental design, The difference between contexts undersea and on the beach seems quite important, hence a context effect should be easier to demonstrate in this experiment.

Because it is not very easy to find deep-sea divers (willing in addition to participate in a memory experiment) it was decided to use the small number of divers in all possible conditions of the design. The list of words were randomly created and assigned to each subject. The order of testing was randomized in order to eliminate any possible carry-over effects by confounding them with the experimental error.

The first independent variable is the place of learning. It has 2 levels (on the beach and undersea), and it is denoted $\mathcal{A}$. The second independent variable is the place of testing. It has 2 levels (on the beach and undersea, like $\mathcal{A}$), and it is denoted $\mathcal{B}$. Crossing these 2 independent variables gives 4 experimental conditions:

- **1.** Learning on the beach and recalling on the beach.
- **2.** Learning on the beach and recalling undersea.

- **3.** Learning undersea and recalling on the beach.
- **4.** Learning undersea and recalling undersea.

Because each subject in this experiment participates in all four experimental conditions, the factor $\mathcal{S}$ is crossed with the 2 experimental factors. Hence, the design can be symbolized as a $\mathcal{S} \times \mathcal{A} \times \mathcal{B}$ design. For this (fictitious) replication of Godden and Baddeley's (1975) experiment we have been able to convince $S = 5$ (fictitious) subjects to take part in this experiment (the original experiment had 16 subjects).

The subjects to learn lists made of 40 short words each. Each list has been made by drawing randomly words from a dictionary. Each list is used just once (hence, because we have $S = 5$ subjects and $A \times B = 2 \times 2 = 4$ experimental conditions, we have $5 \times 4 = 20$ lists). The dependent variable is the number of words recalled 10 minutes after learning (in order to have enough time to plunge or to come back to the beach).

The results of the (fictitious) replication are given in Table 12.1. Please take a careful look at it and make sure you understand the way the results are laid out.

Recall that the prediction of the authors was that memory should be better when the context of encoding and testing is the same than when the context of encoding and testing are different. This means that they have a very specific shape of effects (a so-called X-shaped interaction) in

		$\mathcal{A}$ Learning Place: a_1 On Land	$\mathcal{A}$ Learning Place: a_2 Underwater	$\sum$ $Y_{.1s}$	Means $M_{.1s}$
b_1 Testing place On Land	s_1	34	14	48	24
	s_2	37	21	58	29
	s_3	27	31	58	29
	s_4	43	27	70	35
	s_5	44	32	76	38
		$Y_{11.} = 185$	$Y_{21.} = 125$	$Y_{.1.} = 310$	
		$M_{11.} = 37$	$M_{21.} = 25$	$M_{.1.} = 31$	
b_2 Testing place Underwater	s_1	18	22	40	20
	s_2	21	25	46	23
	s_3	25	33	58	29
	s_4	37	33	70	35
	s_5	34	42	76	38
		$Y_{12.} = 135$	$Y_{22.} = 155$	$Y_{.1.} = 290$	
		$M_{12.} = 27$	$M_{22.} = 31$	$M_{.1.} = 29$	

TABLE 12.1 Result of a (fictitious) replication of Godden and Baddeley's (1975) experiment with deep sea divers (see text for explanation).

mind. As a consequence, they predict that all of the experimental sums of squares should correspond to the sum of squares of interaction.

12.1.1 [R] code

```
# ANOVA Two-factor within subjects, SxA
# Plungin' - Deep Sea Diving Example

# We have 2 Factors, A (Learning), with 2 levels and Factor
# B(Testing) with 2 levels and 5 subjects.

# The 2 levels of Factor A and B are: On Land and Underwater.
# Therefore there are 4 groups with the same 5 observations
#(subjects) per group.

# We collect the data for each subject for all levels of Factor
# A and Factor B for each subject.
b1=c(34,37,27,43,44, 14,21,31,27,32)
b2=c(18,21,25,37,34, 22,25,33,33,42)

# We now combine the observations into one long column (score).
score=c(b1,b2)

# We now prepare the labels for the 4x5 scores according to the
# factor levels:
# a1 a2, a1 a2........etc for Factor A
Learning=gl(2,5*1,5*4*1, labels=c("a1","a2"))

# b1 b2, b1 b2....... etc for Factor B
Testing=gl(2,2*5*1,5*4*1,labels=c("b1","b2"))

# sub_1 sub_1......., sub_2 sub_2......,sub_3 sub_3 ....,sub_4
# sub_4 ....., sub_5 sub_5.....etc for Subjects

Subject=gl(5,1,5*4*1, labels=c("sub_1", "sub_2", "sub_3",
     "sub_4", "sub_5"))

# We now form a data frame with the dependent variable and the
# factors.
data = data.frame(score = score, Learning = factor(Learning), Testing = factor(Testing), Subject = f

# We now perform an anova when "Subject" is considered as a random factor.
aov1 = aov(score ~ (Learning * Testing) + Error(Subject /
    (Learning * Testing)), data = data)

# We now print the data and all the results
summary(aov(score~Learning*Testing*Subject))
summary(aov1)
print(model.tables(aov(score ~ Learning * Testing * Subject,
     data = data), "means"), digits = 3)
```

12.1.2 [R] output

```
> # ANOVA Two-factor within subjects, SxA
> # Plungin' - Deep Sea Diving Example

> # We have 2 Factors, A (Learning), with 2 levels and Factor
> # B(Testing) with 2 levels and 5 subjects.

> # The 2 levels of Factor A and B are: On Land and Underwater.
> # Therefore there are 4 groups with the same 5 observations
> #(subjects) per group.

> # We collect the data for each subject for all levels of Factor
> # A and Factor B for each subject.
> b1=c(34,37,27,43,44, 14,21,31,27,32)
> b2=c(18,21,25,37,34, 22,25,33,33,42)

> # We now combine the observations into one long column (score).
> score=c(b1,b2)

> # We now prepare the labels for the 4x5 scores according to the
> # factor levels:
> # a1 a2, a1 a2........etc for Factor A
> Learning=gl(2,5*1,5*4*1, labels=c("a1","a2"))

> # b1 b2, b1 b2....... etc for Factor B
> Testing=gl(2,2*5*1,5*4*1,labels=c("b1","b2"))

> # sub_1 sub_1......., sub_2 sub_2......,sub_3 sub_3 ....,sub_4
> # sub_4 ....., sub_5 sub_5.....etc for Subjects

> Subject=gl(5,1,5*4*1, labels=c("sub_1", "sub_2", "sub_3",
    "sub_4", "sub_5"))

> # We now form a data frame with the dependent variable and the
> # factors.
> data = data.frame(score = score, Learning = factor(Learning), Testing = factor(Testing), Su

> # We now perform an anova when "Subject" is considered as a random factor.
> aov1 = aov(score ~ (Learning * Testing) + Error(Subject /
   (Learning * Testing)), data = data)

> # We now print the data and all the results
> print(data)
```

	score	Learning	Testing	Subject
1	34	a1	b1	sub_1
2	37	a1	b1	sub_2
3	27	a1	b1	sub_3
4	43	a1	b1	sub_4
5	44	a1	b1	sub_5
6	14	a2	b1	sub_1

```
7     21       a2      b1   sub_2
8     31       a2      b1   sub_3
9     27       a2      b1   sub_4
10    32       a2      b1   sub_5
11    18       a1      b2   sub_1
12    21       a1      b2   sub_2
13    25       a1      b2   sub_3
14    37       a1      b2   sub_4
15    34       a1      b2   sub_5
16    22       a2      b2   sub_1
17    25       a2      b2   sub_2
18    33       a2      b2   sub_3
19    33       a2      b2   sub_4
20    42       a2      b2   sub_5
---------------------------------
```

```
> summary(aov(score~Learning*Testing*Subject))

       -----------------------------------------
                                Df Sum Sq Mean Sq
       -----------------------------------------
       Learning                  1     80      80
       Testing                   1     20      20
       Subject                   4    680     170
       Learning:Testing          1    320     320
       Learning:Subject          4    160      40
       Testing:Subject           4     32       8
       Learning:Testing:Subject  4     64      16
       -----------------------------------------

> summary(aov1)
              Error: Subject
       -----------------------------------------
                 Df Sum Sq Mean Sq F value Pr(>F)
       -----------------------------------------
       Residuals  4    680     170
       -----------------------------------------
              Error: Subject:Learning
       -----------------------------------------
                 Df Sum Sq Mean Sq F value Pr(>F)
       -----------------------------------------
       Learning   1     80      80       2 0.2302
       Residuals  4    160      40
       -----------------------------------------
              Error: Subject:Testing
       -----------------------------------------
                 Df Sum Sq Mean Sq F value Pr(>F)
       -----------------------------------------
       Testing    1     20      20     2.5  0.189
       Residuals  4     32       8
       -----------------------------------------
              Error: Subject:Learning:Testing
       -------------------------------------------------
                        Df Sum Sq Mean Sq F value  Pr(>F)
```

```
  --------------------------------------------------
  Learning:Testing  1    320     320      20 0.01106 *
  Residuals         4     64      16
  --------------------------------------------------
  ---
  Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’
  0.1 ’ ’ 1

> print(model.tables(aov(score ~ Learning * Testing * Subject,
    data = data), "means"), digits = 3)

  Tables of means
  Grand mean
          30

  Learning
  -----
  a1 a2
  -----
  32 28
  -----

  Testing
  -----
  b1 b2
  -----
  31 29
  -----
         Subject
  -----------------------------
  sub_1 sub_2 sub_3 sub_4 sub_5
  -----------------------------
     22    26    29    35    38
  -----------------------------

  Learning:Testing
  ---------------
          Testing
          -------
  Learning  b1 b2
  ----------------
        a1  37 27
        a2  25 31
  ---------------

  Learning:Subject
  --------------------------------------
                       Subject
           -----------------------------
  Learning sub_1 sub_2 sub_3 sub_4 sub_5
  --------------------------------------
        a1    26    29    26    40    39
        a2    18    23    32    30    37
  --------------------------------------
```

```
       Testing:Subject
       -------------------------------------
                         Subject
               -----------------------------
       Testing sub_1 sub_2 sub_3 sub_4 sub_5
       -------------------------------------
            b1    24    29    29    35    38
            b2    20    23    29    35    38
       -------------------------------------

Learning:Testing:Subject

, , Subject = sub_1
---------------
        Testing
         ------
Learning b1  b2
---------------
      a1 34  18
      a2 14  22
---------------


, , Subject = sub_2
---------------
        Testing
         ------
Learning b1  b2
---------------
      a1 37 21
      a2 21 25
---------------


, , Subject = sub_3
---------------
        Testing
         ------
Learning b1  b2
---------------
      a1 27 25
      a2 31 33
---------------


, , Subject = sub_4
---------------
        Testing
         ------
Learning b1  b2
---------------
      a1 43 37
      a2 27 33
```

```
---------------


, , Subject = sub_5
---------------
        Testing
         ------
Learning b1  b2
---------------
      a1 44 34
      a2 32 42
---------------
```

12.1.3 ANOVA table

Here are the final results of the Godden and Baddeley's experiment presented in an ANOVA table.

Source	R^2	*df*	*SS*	*MS*	*F*	P(*F*)
$\mathcal{A}$	0.05900	1	80.00	80.00	2.00	.22973
$\mathcal{B}$	0.01475	1	20.00	20.00	2.50	.18815
$\mathcal{S}$	0.50147	4	680.00	170.00	—	
$\mathcal{AB}$	0.23599	1	320.00	320.00	20.00	.01231
$\mathcal{AS}$	0.11799	4	160.00	40.00	—	
$\mathcal{BS}$	0.02360	4	32.00	8.00	—	
$\mathcal{ABS}$	0.04720	4	64.00	16.00	—	
Total	1.00	19	1,356.00			

13

Factorial Designs: Partially Repeated Measures, $\mathcal{S}(\mathcal{A}) \times \mathcal{B}$

13.1 Bat and Hat....

To illustrate a partially repeated measures or split-plot design, our example will be a (fictitious) replication of an experiment by Conrad (1971). The general idea was to explore the hypothesis that young children do not use phonological coding in short term memory. In order to do this, we select 10 children: 5 five year olds and 5 twelve year olds. This constitutes the first independent variable ($\mathcal{A}$ or *age* with 2 levels), which happens also to be what we have called a "tag" or "classificatory" variable. Because a subject is either five years old or twelve years old, the subject factor ($\mathcal{S}$) is nested in the ($\mathcal{A}$) age factor.

The second independent variable deals with phonological similarity, and we will use the letter $\mathcal{B}$ to symbolize it. But before describing it, we need to delve a bit more into the experiment. Each child was shown 100 pairs of pictures of objects. A pilot study had made sure that children will always use the same name for these pictures (*i.e.,* the cat picture was always called "a cat", never "a pet" or "an animal").

After the children had looked at the pictures, the pictures were turned over so that the children could only see their backs. Then the experimenter gives an identical pair of pictures to the children and asks them to position each new picture on top of the old ones (that are hidden by now) such that the new pictures match the hidden ones. For half of the pairs of pictures, the sound of the name of the objects was similar (*i.e.,* hat and cat), whereas for the other half of the pairs, the sound of the names of the objects in a pair was dissimilar (*i.e.,* horse and chair). This manipulation constitutes the second experimental factor $\mathcal{B}$ or *"phonological similarity."* It has two levels: b_1 phonologically similar and b_2 phonologically dissimilar. The dependent variable will be the number of pairs of pictures correctly positioned by the child.

Conrad reasoned that if the older children use a phonological code to rehearse information, then it would be more difficult for them to remember the phonologically similar pairs than the phonologically dissimilar pairs. This should happen because of an interference effect. If the young children do not use a phonological code to rehearse the material they want to learn, then their performance should be unaffected by phonological similarity, and they should perform at the same level for both conditions of phonological similarity. In addition, because of the usual age effect, one can expect the old children to perform on the whole better than the young ones. Could you draw the graph corresponding to the expected pattern of results? Could you express these predictions in terms of the analysis of variance model?

We expect a main effect of age (which is rather trivial), and also (and, this is the *crucial* point) we expect an interaction effect. This interaction will be the really important test of Conrad's theoretical prediction.

The results of this replication are given in Table 13.1.

13.1.1 [R] code

```
# ANOVA Two-factor Partially Repeated Measures, S(A)xB
# Bat and Hat Example

# We have 2 Factors, A (Age), with 2 levels and Factor B
# (Phonological Similarity) with 2 levels and 10 subjects.

# The 2 levels of Factor A are: Five Years and Twelve Years
```

		$\mathcal{B}$ Phonological Similarity b_1 Similar	$\mathcal{B}$ Phonological Similarity b_2 Dissimilar	$\sum$	Means
				$Y_{1.s}$	$M_{1.s}$
	s_1	15	13	28	14
a_1	s_2	23	19	42	21
Age:	s_3	12	10	22	11
Five Years	s_4	16	16	32	16
	s_5	14	12	26	13
		$Y_{11.} = 80$	$Y_{12.} = 70$	$Y_{1..} = 150$	
		$M_{11.} = 16$	$M_{12.} = 14$	$M_{1..} = 15$	
				$Y_{2.s}$	$M_{2.s}$
	s_6	39	29	68	34
a_2	s_7	31	15	46	23
Age:	s_8	40	30	70	35
Twelve Years	s_9	32	26	58	29
	s_{10}	38	30	68	34
		$Y_{21.} = 180$	$Y_{22.} = 130$	$Y_{2..} = 310$	
		$M_{21.} = 36$	$M_{22.} = 26$	$M_{2..} = 31$	

TABLE 13.1 Results of a replication of Conrad's (1971) experiment.

```
# The 2 levels of Factor B are: Similar and Dissimilar

# The Subjects are nested in Age and crossed with Phonological
# Similarity.
# Therefore there are 4 groups with 5 observations (subjects)
# per group.

# We collect the data for each subjects for all levels of
# Factor A and Factor B for each subject.
b1=c(15,23,12,16,14, 39,31,40,32,38)
b2=c(13,19,10,16,12, 29,15,30,26,30)

# We now combine the observations into one long column (score).
score=c(b1,b2)

# We now prepare the labels for the 4x5 scores according to the
# factor levels:
# a1 a2, a1 a2........etc for Factor A
Age=gl(2,5*1,5*4*1, labels=c("a1","a2"))

# b1 b2, b1 b2....... etc for Factor B
Phono_Sim=gl(2,2*5*1,5*4*1,labels=c("b1","b2"))

# sub_1 sub_1......., sub_2 sub_2......,sub_3 sub_3 ....,sub_4
# sub_4 ....., sub_5 sub_5.....etc for Subjects

Subject=gl(10,1,5*4*1, labels = c("sub_1", "sub_2", "sub_3",
     "sub_4", "sub_5", "sub_6", "sub_7", "sub_8", "sub_9",
     "sub_10"))

# We now form a data frame with the dependent variable and the
# factors.
data = data.frame(score = score, Age = factor(Age), Phono_Sim =
     factor(Phono_Sim), Subject=factor(Subject))

# Anova when "Subject" is considered as a random factor.
aov1 = aov(score ~ (Age * Phono_Sim) + Error(Subject / (Age *
     Phono_Sim) + Age), data=data)

#  We now print the data and all the results

print(data)
summary(aov1)
print(model.tables(aov(score ~ Age * Phono_Sim * Subject, data
     = data), "means"), digits = 3)
```

13.1.2 [R] output

```
> # ANOVA Two-factor Partially Repeated Measures, S(A)xB
> # Bat and Hat Example

> # We have 2 Factors, A (Age), with 2 levels and Factor B
> # (Phonological Similarity) with 2 levels and 10 subjects.
```

```
> # The 2 levels of Factor A are: Five Years and Twelve Years
> # The 2 levels of Factor B are: Similar and Dissimilar

> # The Subjects are nested in Age and crossed with Phonological
> # Similarity.
> # Therefore there are 4 groups with 5 observations (subjects)
> # per group.

> # We collect the data for each subjects for all levels of
> # Factor A and Factor B for each subject.
> b1=c(15,23,12,16,14, 39,31,40,32,38)
> b2=c(13,19,10,16,12, 29,15,30,26,30)

> # We now combine the observations into one long column (score).
> score=c(b1,b2)

> # We now prepare the labels for the 4x5 scores according to the
> # factor levels:
> # a1 a2, a1 a2........etc for Factor A
> Age=gl(2,5*1,5*4*1, labels=c("a1","a2"))

> # b1 b2, b1 b2....... etc for Factor B
> Phono_Sim=gl(2,2*5*1,5*4*1,labels=c("b1","b2"))

> # sub_1 sub_1......., sub_2 sub_2......,sub_3 sub_3 ....,sub_4
> # sub_4 ....., sub_5 sub_5.....etc for Subjects

> Subject=gl(10,1,5*4*1, labels = c("sub_1", "sub_2", "sub_3",
    "sub_4", "sub_5", "sub_6", "sub_7", "sub_8", "sub_9",
    "sub_10"))

> # We now form a data frame with the dependent variable and the
> # factors.
> data = data.frame(score = score, Age = factor(Age), Phono_Sim =
    factor(Phono_Sim), Subject=factor(Subject))

> # Anova when "Subject" is considered as a random factor.
> aov1 = aov(score ~ (Age * Phono_Sim) + Error(Subject / (Age *
    Phono_Sim) + Age), data=data)

> #  We now print the data and all the results

> print(data)
```

	score	Age	Phono_Sim	Subject
1	15	a1	b1	sub_1
2	23	a1	b1	sub_2
3	12	a1	b1	sub_3
4	16	a1	b1	sub_4
5	14	a1	b1	sub_5
6	39	a2	b1	sub_6

```
7      31  a2        b1   sub_7
8      40  a2        b1   sub_8
9      32  a2        b1   sub_9
10     38  a2        b1  sub_10
11     13  a1        b2   sub_1
12     19  a1        b2   sub_2
13     10  a1        b2   sub_3
14     16  a1        b2   sub_4
15     12  a1        b2   sub_5
16     29  a2        b2   sub_6
17     15  a2        b2   sub_7
18     30  a2        b2   sub_8
19     26  a2        b2   sub_9
20     30  a2        b2  sub_10
------------------------------
```

```
> summary(aov1)

       Error: Subject
       ---------------------------------------------
                 Df Sum Sq Mean Sq F value    Pr(>F)
       ---------------------------------------------
       Age        1   1280    1280      32 0.0004776 ***
       Residuals  8    320      40
       ---------------------------------------------
       ---
       Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'
       0.1 ' ' 1

       Error: Subject:Phono_Sim
       -------------------------------------------------
                     Df Sum Sq Mean Sq F value    Pr(>F)
       -------------------------------------------------
       Phono_Sim      1    180     180      45 0.0001514 ***
       Age:Phono_Sim  1     80      80      20 0.0020773 **
       Residuals      8     32       4
       -------------------------------------------------
       ---
       Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'
       0.1 ' ' 1

> print(model.tables(aov(score ~ Age * Phono_Sim * Subject, data
     = data), "means"), digits = 3)

       Tables of means
       Grand mean
               23

       Age
       -----
       a1 a2
       -----
       15 31
       -----
```

```
Phono_Sim
-----
b1 b2
-----
26 20
-----

Subject
-------------------------------------------------------
 sub_1  sub_2  sub_3  sub_4  sub_5  sub_6  sub_7  sub_8
-------------------------------------------------------
    22     29     19     24     21     26     15     27
-------------------------------------------------------


------------
sub_9 sub_10
------------
   21     26
------------

        Age:Phono_Sim
-------------
    Phono_Sim
     --------
Age    b1  b2
-------------
  a1   16  14
  a2   36  26
-------------

Phono_Sim:Subject
---------------------------------------------------------
                                 Subject
          -----------------------------------------------
Phono_Sim sub_1 sub_2 sub_3 sub_4 sub_5 sub_6 sub_7 sub_8
---------------------------------------------------------
       b1    25    33    22    26    24    29    21    30
       b2    19    25    16    22    18    23     9    24


----------------------
             Subject
          ------------
Phono_Sim sub_9 sub_10
----------------------
       b1    22    28
       b2    20    24
---------------------
```

13.1.3 ANOVA table

We can now fill in the ANOVA Table as shown in Table 13.2.

Source	*df*	*SS*	*MS*	*F*	$Pr(F)$
between subjects					
$\mathcal{A}$	1	1,280.00	1,280.00	32.00	.00056
$\mathcal{S}(\mathcal{A})$	8	320.00	40.00	-----	
within subjects					
$\mathcal{B}$	1	180.00	180.00	45.00	.00020
$\mathcal{AB}$	1	80.00	80.00	20.00	.00220
$\mathcal{BS}(\mathcal{A})$	8	32.00	4.00	-----	
Total...........	19	1,892.00			

TABLE 13.2 The analysis of variance Table for a replication of Conrad's (1971) experiment (data from Table 13.1).

As you can see from the results of the analysis of variance, the experimental predictions are supported by the experimental results. The results section of an APA style paper would indicate the following information:

> The results were treated as an Age $\times$ Phonological similarity analysis of variance design with Age (5 year olds *versus* 12 year olds) being a between-subject factor and phonological similarity (similar *versus* dissimilar) being a within-subject factor. There was a very clear effect of age, $F(1, 8) = 31.5$, $MS_e = 33.38$, $p < 01$. The expected interaction of age by phonological similarity was also very reliable $F(1, 8) = 35.2$, $MS_e = 2.86$, $p < 01$. A main effect of phonological similarity was also detected $F(1, 8) = 52.6$, $MS_e = 2.86$, $p < 01$, but its interpretation as a main effect is delicate because of the strong interaction between phonological similarity and age.

14 Nested Factorial Design: $\mathcal{S} \times \mathcal{A}(\mathcal{B})$

14.1 Faces in Space

Some faces give the impression of being original or bizarre. Some other faces, by contrast, give the impression of being average or common. We say that original faces are *atypical*; and that common faces are *typical*. In terms of design factors, we say that: Faces vary on the Typicality factor (which has 2 levels: typical *vs.* atypical).

In this example, we are interested by the effect of typicality on reaction time. Presumably, typical faces should be easier to process as faces than atypical faces. In this[1] example, we measured the reaction time of 4 subjects in a face identification task. Ten faces, mixed with ten "distractor faces," were presented on the screen of a computer. The distractor faces were jumbled faces (*e.g.,* with the nose at the place of the mouth). Five of the ten faces were typical faces, and the other five faces were atypical faces. Subjects were asked to respond as quickly as they can. Only the data recorded from the normal faces (*i.e.,* not jumbled) were kept for further analysis. All the subjects identified correctly the faces as faces. The data (made nice for the circumstances) are given in Table 14.1 on page 180. As usual, make sure that you understand its layout, and try to figure out whether there is some effect of Typicality.

Here, like in most $\mathcal{S} \times \mathcal{A}(\mathcal{B})$ designs, we are mainly interested in the nesting factor (*i.e.,* $\mathcal{B}$). The nested factor [*i.e.,* $\mathcal{A}(\mathcal{B})$] is not, however, without interest. If it is statistically significant, this may indicate that the pattern of effects, which we see in the results, depends upon the *specific* sample of items used in this experiment.

14.1.1 [R] code

```
# ANOVA Two-factor Nested factorial Design, SxA(B)
# Faces in Space

# Factor B is Typicality. Factor A(B) is Faces nested in
```

[1]Somewhat fictitious, but close to some standard experiments in face recognition.

```
# Typicality. There are 4 subjects in the experiment.

# Therefore there are 4 groups with 5 observations (subjects)
# per group.

# We collect the data for each subjects for all levels of
# Factor A and Factor B for each subject.
a_b1=c(20,22,25,24,19, 9,8,21,21,21, 18,20,18,21,33,
     5,14,16,22,23)
a_b2=c(37,37,43,48,45, 34,35,35,37,39, 35,39,39,37,40,
     38,49,51,50,52)

# We now combine the observations into one long column (score).
score=c(a_b1,a_b2)

# We now prepare the labels for the 4x5x2 scores according to
     the factor levels:
# a1 a2 a3 a4 a5, a1 a2 a3 a4 a5........etc for Factor A
Face=gl(5,1,5*4*2, labels=c("a1","a2","a3","a4","a5"))

# b1 b2, b1 b2....... etc for Factor B
Typicality=gl(2,4*5*1,5*4*2,labels=c("Atypical","Typical"))

# sub_1 sub_1......., sub_2 sub_2......,sub_3 sub_3 ....,sub_4
# sub_4 ....., sub_5 sub_5.....etc for Subjects
Subject=gl(4, 5*1, 5*4*2, labels = c("sub_1", "sub_2", "sub_3",
     "sub_4"))

# We now form a data frame with the dependent variable and the # factors.
data = data.frame(score = score, Face = factor(Face),
     Typicality = factor(Typicality))

# Anova when "Subject" is considered as a random factor.
aov1 = aov(score ~ (Subject + Face%in%Typicality + Typicality +
     Typicality:Subject))

Df = summary(aov(score ~ (Subject + Face%in%Typicality +
     Typicality + Typicality:Subject)))[[1]]$Df

Sum_Sq = summary(aov(score ~ (Subject + Face%in%Typicality +
     Typicality + Typicality:Subject)))[[1]]$Sum

MS = summary(aov(score ~ (Subject + Face%in%Typicality +
     Typicality + Typicality:Subject)))[[1]]$Mean

F = summary(aov(score ~ (Subject + Face%in%Typicality +
     Typicality + Typicality:Subject)))[[1]]$F
F[2]=NA
Pr = summary(aov(score ~ (Subject + Face%in%Typicality +
     Typicality + Typicality:Subject)))[[1]]$Pr
Pr[2]=NA

Source_names = c("Subject", "Typicality", "Face(Typicality)",
     "Subject * Typicality", "Error:Face * Subject(Typicality)")
```

```
Anova_table = data.frame("Df" = Df, "Sum Sq" = Sum_Sq, "Mean
     Sq" = MS, "F Value" = F, "Pr F" = Pr,row. Names =
     Source_names)

# We now print the data and all the results
print(data)
print(Anova_table)
print("The 'Typicality' factor has a Quasi F or F'. This F' has
      not been displayed in the Anova table and has to be
      calculated separately")
print(model.tables(aov1,"means"),digits=3)
```

14.1.2 [R] output

```
> # ANOVA Two-factor Nested factorial Design, SxA(B)
> # Faces in Space

> # Factor B is Typicality. Factor A(B) is Faces nested in
> # Typicality. There are 4 subjects in the experiment.

> # Therefore there are 4 groups with 5 observations (subjects)
> # per group.

> # We collect the data for each subjects for all levels of
> # Factor A and Factor B for each subject.
> a_b1=c(20,22,25,24,19, 9,8,21,21,21, 18,20,18,21,33,
     5,14,16,22,23)
> a_b2=c(37,37,43,48,45, 34,35,35,37,39, 35,39,39,37,40,
     38,49,51,50,52)

> # We now combine the observations into one long column (score).
> score=c(a_b1,a_b2)

> # We now prepare the labels for the 4x5x2 scores according to
     the factor levels:
> # a1 a2 a3 a4 a5, a1 a2 a3 a4 a5........etc for Factor A
> Face=gl(5,1,5*4*2, labels=c("a1","a2","a3","a4","a5"))

> # b1 b2, b1 b2....... etc for Factor B
> Typicality=gl(2,4*5*1,5*4*2,labels=c("Atypical","Typical"))

> # sub_1 sub_1......., sub_2 sub_2......,sub_3 sub_3 ....,sub_4
> # sub_4 ....., sub_5 sub_5.....etc for Subjects
> Subject=gl(4, 5*1, 5*4*2, labels = c("sub_1", "sub_2", "sub_3",
     "sub_4"))

> # We now form a data frame with the dependent variable and the > # factors.
> data = data.frame(score = score, Face = factor(Face),
     Typicality = factor(Typicality))

> # Anova when "Subject" is considered as a random factor.
> aov1 = aov(score ~ (Subject + Face%in%Typicality + Typicality +
```

```
   Typicality:Subject))

> Df = summary(aov(score ~ (Subject + Face%in%Typicality +
   Typicality + Typicality:Subject)))[[1]]$Df

> Sum_Sq = summary(aov(score ~ (Subject + Face%in%Typicality +
   Typicality + Typicality:Subject)))[[1]]$Sum

> MS = summary(aov(score ~ (Subject + Face%in%Typicality +
   Typicality + Typicality:Subject)))[[1]]$Mean

> F = summary(aov(score ~ (Subject + Face%in%Typicality +
   Typicality + Typicality:Subject)))[[1]]$F
> F[2]=NA
> Pr = summary(aov(score ~ (Subject + Face%in%Typicality +
   Typicality + Typicality:Subject)))[[1]]$Pr
> Pr[2]=NA

> Source_names = c("Subject", "Typicality", "Face(Typicality)",
   "Subject * Typicality", "Error:Face * Subject(Typicality)")

> Anova_table = data.frame("Df" = Df, "Sum Sq" = Sum_Sq, "Mean
   Sq" = MS, "F Value" = F, "Pr > F" = Pr,row. Names =
   Source_names)

> # We now print the data and all the results
> print(data)

     ------------------------
        score Face Typicality
     ------------------------
     1     20   a1   Atypical
     2     22   a2   Atypical
     3     25   a3   Atypical
     4     24   a4   Atypical
     5     19   a5   Atypical
     6      9   a1   Atypical
     7      8   a2   Atypical
     8     21   a3   Atypical
     9     21   a4   Atypical
     10    21   a5   Atypical
     11    18   a1   Atypical
     12    20   a2   Atypical
     13    18   a3   Atypical
     14    21   a4   Atypical
     15    33   a5   Atypical
     16     5   a1   Atypical
     17    14   a2   Atypical
     18    16   a3   Atypical
     19    22   a4   Atypical
     20    23   a5   Atypical
     21    37   a1    Typical
     22    37   a2    Typical
     23    43   a3    Typical
```

```
        24    48   a4    Typical
        25    45   a5    Typical
        26    34   a1    Typical
        27    35   a2    Typical
        28    35   a3    Typical
        29    37   a4    Typical
        30    39   a5    Typical
        31    35   a1    Typical
        32    39   a2    Typical
        33    39   a3    Typical
        34    37   a4    Typical
        35    40   a5    Typical
        36    38   a1    Typical
        37    49   a2    Typical
        38    51   a3    Typical
        39    50   a4    Typical
        40    52   a5    Typical
        ------------------------

> print(Anova_table)

        ---------------------------------------------------------
                                       Df Sum.Sq Mean.Sq  F.Value
        ---------------------------------------------------------
        Subject                         3    240      80 5.333333
        Typicality                      1   4840    4840       NA
        Face(Typicality)                8    480      60 4.000000
        Subject*Typicality              3    360     120 8.000000
        Error:Face*Subject(Typicality) 24    360      15       NA
        ---------------------------------------------------------


        -------------------------------------------
                                               Pr.F
        -------------------------------------------
        Subject                        0.0058525025
        Typicality                               NA
        Face(Typicality)               0.0038873328
        Subject*Typicality             0.0007215102
        Error:Face*Subject(Typicality)           NA
        -------------------------------------------

> print("The 'Typicality' factor has a Quasi F or F'. This F' has
      not been displayed in the Anova table and has to be
      calculated separately")

        [1] "The 'Typicality' factor has a Quasi F or F'. This F'
        has not been displayed in the Anova table and has to be
        calculated separately"

> print(model.tables(aov1,"means"),digits=3)

Tables of means
```

```
Grand mean
30


Subject
-----------------------
sub_1 sub_2 sub_3 sub_4
-----------------------
   32    26    30    32
-----------------------


Typicality
-----------------
Atypical  Typical
-----------------
      19       41
-----------------


Face:Typicality
---------------------
        Typicality
     ----------------
Face Atypical Typical
---------------------
  a1       13      36
  a2       16      40
  a3       20      42
  a4       22      43
  a5       24      44
---------------------


Subject:Typicality
------------------------
       Typicality
        ----------------
Subject Atypical Typical
------------------------
  sub_1       22      42
  sub_2       16      36
  sub_3       22      38
  sub_4       16      48
------------------------
```

14.1.3 *F* and Quasi-*F* ratios

Remember the standard procedure? In order to evaluate the reliability of a source of variation, we need to find the expected value of its mean square. Then we assume that the null hypothesis is true, and we try to find another source whose mean square has the same expected value. This mean square is the test mean square. Dividing the first mean square (the effect mean square) by the test mean square gives an F ratio. When there is no test mean square, there is no way to compute an F ratio, especially with SAS. However, as you know, combining several mean squares gives a test mean square called a test "quasi-mean square" or a "test mean square prime." The ratio of the effect mean square by its "quasi-mean square" give a "quasi-F ratio" (or F'). The expected values of the mean squares for a $\mathcal{S} \times \mathcal{A}(\mathcal{B})$ design with $\mathcal{A}(\mathcal{B})$ random and $\mathcal{B}$ fixed are given in Table 14.2 on page 181.

From Table 14.2 on page 181, we find that most sources of variation may be evaluated by computing F ratios using the $\mathcal{AS}(\mathcal{B})$ mean square. Unfortunately, the experimental factor of prime interest (*i.e.,* $\mathcal{B}$), cannot be tested this way, but requires the use of a quasi-F'. The test mean square for the main effect of $\mathcal{B}$ is obtained as

$$MS'_{\text{test},\mathcal{B}} = MS_{A(B)} + MS_{BS} - MS_{AS(B)} \,. \tag{14.1}$$

The number of degrees of freedom of the mean square of test is approximated by the following formula (Eeek!):

$$\nu'_2 = \frac{(MS_{A(B)} + MS_{BS} - MS_{AS(B)})^2}{\dfrac{MS^2_{A(B)}}{df_{A(B)}} + \dfrac{MS^2_{BS}}{df_{BS}} + \dfrac{MS^2_{AS(B)}}{df_{AS(B)}}} \,. \tag{14.2}$$

14.1.4 ANOVA table

We can now fill in the ANOVA Table as shown in Table 14.3 on page 181.

	Factor $\mathcal{B}$ (Typicality: Atypical *vs.* Typical)												
	b_1: (Atypical) $\mathcal{A}_{a(b_1)}$: (Atypical *Faces*)						b_2: (Typical) $\mathcal{A}_{a(b_2)}$: (Typical *Faces*)						
	a_1	a_2	a_3	a_4	a_5	$M_{.1s}$	a_1	a_2	a_3	a_4	a_5	$M_{.2s}$	$M_{..s}$
s_1	20	22	25	24	19	*22*	37	37	43	48	45	*42*	**32**
s_2	9	8	21	21	21	*16*	34	35	35	37	39	*36*	**26**
s_3	18	20	18	21	33	*22*	35	39	39	37	40	*38*	**30**
s_4	5	14	16	22	23	*16*	38	49	51	50	52	*48*	**32**
	$M_{11.}$	$M_{21.}$	$M_{31.}$	$M_{41.}$	$M_{51.}$		$M_{12.}$	$M_{22.}$	$M_{32.}$	$M_{42.}$	$M_{52.}$		
	13	16	20	22	24		36	40	42	43	44	$M_{...} = 30$	
	$M_{.1.} = 19$						$M_{.2.} = 41$						

TABLE 14.1

Data from a fictitious experiment with a $\mathcal{S} \times \mathcal{A}(\mathcal{B})$ design. Factor $\mathcal{B}$ is Typicality. Factor $\mathcal{A}(\mathcal{B})$ is Faces (nested in Typicality). There are 4 subjects in this experiment. The dependent variable is measured in centiseconds (in case you wonder: 1 centisecond equals 10 milliseconds); and it is the time taken by a subject to respond that a given face was a face.

Source	Expected Mean Squares	MS_{test}
$\mathcal{B}$	$\sigma_e^2 + \sigma_{as(b)}^2 + A\sigma_{bs}^2 + S\sigma_{a(b)}^2 + AS\vartheta_b^2$	$MS_{A(B)} + MS_{BS} - MS_{AS(B)}$
$\mathcal{S}$	$\sigma_e^2 + \sigma_{as(b)}^2 + AB\sigma_s^2$	$MS_{AS(B)}$
$\mathcal{A}(\mathcal{B})$	$\sigma_e^2 + \sigma_{as(b)}^2 + S\sigma_{a(b)}^2$	$MS_{AS(B)}$
$\mathcal{BS}$	$\sigma_e^2 + \sigma_{as(b)}^2 + A\sigma_{bs}^2$	$MS_{AS(B)}$
$\mathcal{AS}(\mathcal{B})$	$\sigma_e^2 + \sigma_{as(b)}^2$	

TABLE 14.2 The expected mean squares when $\mathcal{A}$ is random and $\mathcal{B}$ is fixed for an $\mathcal{S} \times \mathcal{A}(\mathcal{B})$ design.

Source	R^2	*df*	*SS*	*MS*	*F*	Pr(F)	ν_1	ν_2
Face $\mathcal{A}(\mathcal{B})$	0.08	8	480.00	60.00	4.00	.0040	8	24
Typicality $\mathcal{B}$	0.77	1	4,840.00	4,840.00	29.33†	.0031†	1	5†
Subject $\mathcal{S}$	0.04	3	240.00	80.00	5.33	.0060	3	24
Subject by Face $\mathcal{AS}(\mathcal{B})$	0.06	24	360.00	15.00				
Subject by Typicality $\mathcal{BS}$	0.06	3	360.00	120.00	8.00	.0008	3	24
Total	1.00	39	13,254.00					

† This value has been obtained using a Quasi-F approach. See text for explanation.

TABLE 14.3 The ANOVA Table for the data from Table 14.1.

Index