Personality Project

Big Five Test PMC Lab Psychometric Theory psych package

R and Analysis of Variance

A special case of the linear model is the situation where the predictor variables are categorical. In psychological research this usually reflects experimental design where the independent variables are multiple levels of some experimental manipulation (e.g., drug administration, recall instructions, etc.)

The first 5 examples are adapted from the guide to S+ developed by TAs for Roger Ratcliff. For more detail on data entry consult that guide. The last three examples discuss how to reorganize the data from a standard data frame into one appropriate for within subject analyses. For this discussion, I assume that appropriate data files have been created in a text editor and saved in a subjects x variables table.

One Way Analysis of Variance

Example 1: Three levels of drug were administered to 18 subjects. Do descriptive statistics on the groups, and then do a one way analysis of variance. The ANOVA command is aov:

```
aov.ex1= aov(Alertness~Dosage,data=ex1)
```

It is important to note the order of the arguments. The first argument is always the dependent variable (Alertness). It is followed by the tilde symbol (~) and the independent variable(s). The final argument for *aov* is the name of the data structure that is being analyzed. aov.ex1 is the name of the structure you want the analysis to store. This general format will hold true for all ANOVAs you will conduct.

The results of the ANOVA can be seen with the summary command:

```
#tell where the data come from
datafilename="http://personality-project.org/R/datasets/R.appendix1.data"
data.ex1=read.table(datafilename,header=T)   #read the data into a table

aov.ex1 = aov(Alertness~Dosage,data=data.ex1)  #do the analysis of variance
summary(aov.ex1)                                    #show the summary table
print(model.tables(aov.ex1,"means"),digits=3)       #report the means and the number of subjects/cell
boxplot(Alertness~Dosage,data=data.ex1)        #graphical summary
```

produces this output

```
> datafilename="http://personality-project.org/r/datasets/R.appendix1.data"
> data.ex1=read.table(datafilename,header=T)   #read the data into a table
>
> aov.ex1 = aov(Alertness~Dosage,data=data.ex1)  #do the analysis of variance
> summary(aov.ex1)                                    #show the summary table
            Df Sum Sq Mean Sq F value   Pr(>F)
Dosage       2 426.25  213.12  8.7887 0.002977 **
Residuals   15 363.75   24.25
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
> print(model.tables(aov.ex1,"means"),digits=3)       #report the means and the number of subjects/cell
Tables of means
Grand mean

27.66667

 Dosage
      a    b    c
    32.5 28.2 19.2
rep  6.0  8.0  4.0
```

Two way - between subject analysis of variance

Data are from an experiment in which alertness level of male and female subjects was measured after they had been given one of two possible dosages of a drug. Thus, this is a 2X2 design with the factors being Gender and Dosage. Read the data file containing this data. Notice that there are two independent variables in this example, separated by an asterisk *. The asterisk indicates to R that the interaction between the two factors is interesting and should be analyzed. If interactions are not important, replace the asterisk with a plus sign (+).

Run the analysis:

```
datafilename="http://personality-project.org/r/datasets/R.appendix2.data"
data.ex2=read.table(datafilename,header=T)   #read the data into a table
data.ex2                                      #show the data
aov.ex2 = aov(Alertness~Gender*Dosage,data=data.ex2)         #do the analysis of variance
summary(aov.ex2)                                    #show the summary table
print(model.tables(aov.ex2,"means"),digits=3)       #report the means and the number of subjects/cell
boxplot(Alertness~Dosage*Gender,data=data.ex2) #graphical summary of means of the 4 cells
```

The output should look like the following:

```
> datafilename="http://personality-project.org/r/datasets/R.appendix2.data"
> data.example2=read.table(datafilename,header=T)   #read the data into a table
> data.example2                                     #show the data

   Observation Gender Dosage Alertness
1            1      m      a         8
2            2      m      a        12
3            3      m      a        13
4            4      m      a        12
5            5      m      b         6
6            6      m      b         7
7            7      m      b        23
8            8      m      b        14
9            9      f      a        15
10          10      f      a        12
11          11      f      a        22
12          12      f      a        14
13          13      f      b        15
14          14      f      b        12
15          15      f      b        18
16          16      f      b        22

> aov.ex2 = aov(Alertness~Gender*Dosage,data=data.example2)         #do the analysis of variance
> summary(aov.ex2)                                    #show the summary table
              Df  Sum Sq Mean Sq F value Pr(>F)
Gender         1  76.562  76.562  2.9518 0.1115
Dosage         1   5.062   5.062  0.1952 0.6665
Gender:Dosage  1   0.063   0.063  0.0024 0.9617
Residuals     12 311.250  25.938
> print(model.tables(aov.ex2,"means"),digits=3)       #report the means and the number of subjects/cell
Tables of means
Grand mean

14.0625

 Gender
      f    m
   16.2 11.9
rep  8.0  8.0

 Dosage
      a    b
   13.5 14.6
rep  8.0  8.0

 Gender:Dosage
      Dosage
Gender a     b
   f   15.75 16.75
   rep  4.00  4.00
   m   11.25 12.50
   rep  4.00  4.00
```

The generalization to n way ANOVA is straightforward.

1 way ANOVA - within subjects

Example 3. One-Way Within-Subjects ANOVA

Five subjects are asked to memorize a list of words. The words on this list are of three types: positive words, negative words and neutral words. Their recall data by word type is displayed in Appendix III. Note that there is a single factor (Valence) with three levels (negative, neutral and positive). In addition, there is also a random factor Subject . Create a data file ex3 that contains this data. Again it is important that each observation appears on an individual line! Note that this is not the standard way of thinking about data. Example 6 will show how to transform data from the standard data table into this form.

```
                      #Run the analysis:
datafilename="http://personality-project.org/r/datasets/R.appendix3.data"
data.ex3=read.table(datafilename,header=T)   #read the data into a table
data.ex3                                     #show the data
aov.ex3 = aov(Recall~Valence+Error(Subject/Valence),data.ex3)
summary(aov.ex3)
print(model.tables(aov.ex3,"means"),digits=3)       #report the means and the number of subjects/cell
boxplot(Recall~Valence,data=data.ex3)          #graphical output
```

Because Valence is crossed with the random factor Subject (i.e., every subject sees all three types of words), you must specify the error term for Valence , which in this case is Subject by Valence . Do this by adding the termError(Subject/Valence) to the factor Valence , as shown above. The output will look like:

```
> datafilename="http://personality-project.org/r/datasets/R.appendix3.data"
> data.ex3=read.table(datafilename,header=T)   #read the data into a table
> data.ex3                                     #show the data
   Observation Subject Valence Recall
1            1     Jim     Neg     32
2            2     Jim     Neu     15
3            3     Jim     Pos     45
4            4  Victor     Neg     30
5            5  Victor     Neu     13
6            6  Victor     Pos     40
7            7    Faye     Neg     26
8            8    Faye     Neu     12
9            9    Faye     Pos     42
10          10     Ron     Neg     22
```

```
11          11     Ron     Neu     10
12          12     Ron     Pos     38
13          13   Jason     Neg     29
14          14   Jason     Neu      8
15          15   Jason     Pos     35
> aov.ex3 = aov(Recall~Valence+Error(Subject/Valence),data.ex3)
> summary(aov.ex3)

Error: Subject
          Df  Sum Sq Mean Sq F value Pr(>F)
Residuals  4 105.067  26.267

Error: Subject:Valence
          Df  Sum Sq Mean Sq F value    Pr(>F)
Valence    2 2029.73 1014.87  189.11 1.841e-07 ***
Residuals  8   42.93    5.37
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
> print(model.tables(aov.ex3,"means"),digits=3)       #report the means and the number of subjects/cell
Tables of means
Grand mean

26.46667

 Valence
Valence
 Neg  Neu  Pos
27.8 11.6 40.0
```

The analysis of between-subjects factors will appear first (there are none in this case), followed by the within-subjects factors. Note that the p value for Valence is displayed in exponential notation; this occurs when the p value is extremely low, as it is in this case (approximately .00000018).

Two-way Within Subjects ANOVA

Example 4. Two-Way Within-Subjects ANOVA

In this example, Subject is crossed with both Task and Valence, so you must specify three different error terms: one forTask , one for Valence and one for the interaction between the two. Fortunately, R is smart enough to divide up the within-subjects error term properly as long as you specify it in your command. The commands are:

```
datafilename="http://personality-project.org/r/datasets/R.appendix4.data"
data.ex4=read.table(datafilename,header=T)   #read the data into a table
 data.ex4                                      #show the data
 aov.ex4=aov(Recall~(Task*Valence)+Error(Subject/(Task*Valence)),data.ex4 )

summary(aov.ex4)
print(model.tables(aov.ex4,"means"),digits=3)       #report the means and the number of subjects/cell
boxplot(Recall~Task*Valence,data=data.ex4) #graphical summary of means of the 6 cells
```

results in the following output:

```
> datafilename="http://personality-project.org/r/datasets/R.appendix4.data"
> data.example4=read.table(datafilename,header=T)   #read the data into a table
>  data.example4                                      #show the data
   Observation Subject Task Valence Recall
1            1     Jim Free     Neg      8
2            2     Jim Free     Neu      9
3            3     Jim Free     Pos      5
4            4     Jim Cued     Neg      7
5            5     Jim Cued     Neu      9
6            6     Jim Cued     Pos     10
7            7  Victor Free     Neg     12
8            8  Victor Free     Neu     13
9            9  Victor Free     Pos     14
10          10  Victor Cued     Neg     16
11          11  Victor Cued     Neu     13
12          12  Victor Cued     Pos     14
13          13    Faye Free     Neg     13
14          14    Faye Free     Neu     13
15          15    Faye Free     Pos     12
16          16    Faye Cued     Neg     15
17          17    Faye Cued     Neu     16
18          18    Faye Cued     Pos     14
19          19     Ron Free     Neg     12
20          20     Ron Free     Neu     14
21          21     Ron Free     Pos     15
22          22     Ron Cued     Neg     17
23          23     Ron Cued     Neu     18
24          24     Ron Cued     Pos     20
25          25   Jason Free     Neg      6
26          26   Jason Free     Neu      7
27          27   Jason Free     Pos      9
28          28   Jason Cued     Neg      4
29          29   Jason Cued     Neu      9
30          30   Jason Cued     Pos     10
>  aov.ex4=aov(Recall~(Task*Valence)+Error(Subject/(Task*Valence)),data.example4 )
>
> summary(aov.ex4)

Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals  4 349.13   87.28
```

```
Error: Subject:Task
          Df  Sum Sq Mean Sq F value  Pr(>F)
Task       1 30.0000 30.0000  7.3469 0.05351 .
Residuals  4 16.3333  4.0833
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Error: Subject:Valence
          Df  Sum Sq Mean Sq F value Pr(>F)
Valence    2  9.8000  4.9000  1.4591 0.2883
Residuals  8 26.8667  3.3583

Error: Subject:Task:Valence
             Df  Sum Sq Mean Sq F value Pr(>F)
Task:Valence  2  1.4000  0.7000  0.2907 0.7553
Residuals     8 19.2667  2.4083
> print(model.tables(aov.ex4,"means"),digits=3)       #report the means and the number of subjects/cell
Tables of means
Grand mean

11.8

 Task
    Cued Free
    12.8 10.8
rep 15.0 15.0

 Valence
    Neg  Neu  Pos
     11 12.1 12.3
rep  10 10.0 10.0

 Task:Valence
      Valence
Task   Neg  Neu  Pos
  Cued 11.8 13.0 13.6
  rep   5.0  5.0  5.0
  Free 10.2 11.2 11.0
  rep   5.0  5.0  5.0
```

Mixed (between and Within) designs

Now it's time to get serious. Appendix V contains the data of an experiment with 18 subjects, 9 males and 9 females. Each subject is given one of three possible dosages of a drug. All subjects are then tested on recall of three types of words (positive, negative and neutral) using two types of memory tasks (cued and free recall). There are thus 2 between-subjects variables: Gender (2 levels) and Dosage (3 levels); and 2 within-subjects variables: Task (2 levels) and Valence (3 levels). Get the data from the file and run the following analysis:

```
aov.ex5 _ aov(Recall~(Task*Valence*Gender*Dosage)+Error(Subject/(Task*Valence))+(Gender*Dosage),ex5)
```

Notice that you must segregate between- and within-subjects variables in your command. In the above example, I have put the within-subjects factors first with the within-subjects error term, followed by the between-subjects factors.

```
datafilename="http://personality-project.org/r/datasets/R.appendix5.data"
data.ex5=read.table(datafilename,header=T)   #read the data into a table
data.ex5                                      #show the data
aov.ex5 = aov(Recall~(Task*Valence*Gender*Dosage)+Error(Subject/(Task*Valence))+(Gender*Dosage),data.ex5   )

summary(aov.ex5)
print(model.tables(aov.ex5,"means"),digits=3)       #report the means and the number of subjects/cell
boxplot(Recall~Task*Valence*Gender*Dosage,data=data.ex5) #graphical summary of means of the 36 cells
boxplot(Recall~Task*Valence*Dosage,data=data.ex5) #graphical summary of means of  18 cells
```

Should result in the following (extensive) output:

```
> datafilename="http://personality-project.org/r/datasets/R.appendix5.data"
> data.example5=read.table(datafilename,header=T)   #read the data into a table
> data.example5                                      #show the data
    Obs Subject Gender Dosage Task Valence Recall
1     1       A      M      A    F     Neg      8
2     2       A      M      A    F     Neu      9
3     3       A      M      A    F     Pos      5
4     4       A      M      A    C     Neg      7
5     5       A      M      A    C     Neu      9
6     6       A      M      A    C     Pos     10
7     7       B      M      A    F     Neg     12
8     8       B      M      A    F     Neu     13
9     9       B      M      A    F     Pos     14
10   10       B      M      A    C     Neg     16
...  SNIP    ....

100 100       Q      F      C    C     Neg     17
101 101       Q      F      C    C     Neu     19
102 102       Q      F      C    C     Pos     19
103 103       R      F      C    F     Neg     19
104 104       R      F      C    F     Neu     17
105 105       R      F      C    F     Pos     19
106 106       R      F      C    C     Neg     22
```

```
107 107       R      F      C    C     Neu     21
108 108       R      F      C    C     Pos     20
> aov.ex5=aov.ex5 = aov(Recall~(Task*Valence*Gender*Dosage)+Error(Subject/(Task*Valence))+(Gender*Dosage),data.example5   )
>
> summary(aov.ex5)

Error: Subject
              Df  Sum Sq Mean Sq F value  Pr(>F)
Gender         1  542.26  542.26  5.6853 0.03449 *
Dosage         2  694.91  347.45  3.6429 0.05803 .
Gender:Dosage  2   70.80   35.40  0.3711 0.69760
Residuals     12 1144.56   95.38
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Error: Subject:Task
                   Df Sum Sq Mean Sq F value    Pr(>F)
Task                1 96.333  96.333 39.8621 3.868e-05 ***
Task:Gender         1  1.333   1.333  0.5517    0.4719
Task:Dosage         2  8.167   4.083  1.6897    0.2257
Task:Gender:Dosage  2  3.167   1.583  0.6552    0.5370
Residuals          12 29.000   2.417
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Error: Subject:Valence
                      Df Sum Sq Mean Sq F value  Pr(>F)
Valence                2 14.685   7.343  2.9981 0.06882 .
Valence:Gender         2  3.907   1.954  0.7977 0.46193
Valence:Dosage         4 20.259   5.065  2.0681 0.11663
Valence:Gender:Dosage  4  1.037   0.259  0.1059 0.97935
Residuals             24 58.778   2.449
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Error: Subject:Task:Valence
                           Df Sum Sq Mean Sq F value Pr(>F)
Task:Valence                2  5.389   2.694  1.3197 0.2859
Task:Valence:Gender         2  2.167   1.083  0.5306 0.5950
Task:Valence:Dosage         4  2.778   0.694  0.3401 0.8482
Task:Valence:Gender:Dosage  4  2.667   0.667  0.3265 0.8574
Residuals                  24 49.000   2.042
> print(model.tables(aov.ex5,"means"),digits=3)       #report the means and the number of subjects/cell


Tables of means
Grand mean

15.62963

 Task
       C    F
    16.6 14.7
rep 54.0 54.0

 Valence
     Neg  Neu  Pos
    15.3 15.5 16.1
rep 36.0 36.0 36.0

 Gender
       F    M
    17.9 13.4
rep 54.0 54.0

 Dosage
       A    B    C
    14.2 13.5 19.2
rep 36.0 36.0 36.0

 Task:Valence
     Valence
Task  Neg   Neu   Pos
  C   16.00 16.72 17.00
  rep 18.00 18.00 18.00
  F   14.56 14.22 15.28
  rep 18.00 18.00 18.00

 Task:Gender
     Gender
Task  F     M
  C   18.93 14.22
  rep 27.00 27.00
  F   16.81 12.56
  rep 27.00 27.00

 Valence:Gender
       Gender
Valence F     M
    Neg 17.67 12.89
    rep 18.00 18.00
    Neu 17.44 13.50
    rep 18.00 18.00
    Pos 18.50 13.78
    rep 18.00 18.00

 ... snip ....
```

```
, , Gender = M, Dosage = B

     Valence
Task  Neg   Neu   Pos
  C   10.00 11.67 12.33
  rep  3.00  3.00  3.00
  F    8.33  8.67 11.00
  rep  3.00  3.00  3.00

, , Gender = F, Dosage = C

     Valence
Task  Neg   Neu   Pos
  C   20.67 21.67 21.33
  rep  3.00  3.00  3.00
  F   19.67 18.67 20.33
  rep  3.00  3.00  3.00

, , Gender = M, Dosage = C

     Valence
Task  Neg   Neu   Pos
  C   18.00 19.00 19.00
  rep  3.00  3.00  3.00
  F   17.33 17.33 17.33
  rep  3.00  3.00  3.00
```

Reorganizing the data for within subject analyses

The prior examples have assumed one line per unique subject/variable combination. This is not a typical way to enter data. A more typical way (found e.g., in Systat) is to have one row/subject. We need to "stack" the data to go from the standard input to the form preferred by the analysis of variance. Consider the following analyses of 27 subjects doing a memory study of the effect on recall of two presentation rates and two recall intervals. Each subject has two replications per condition. The first 8 columns are the raw data, the last 4 columns collapse across replications. The data are found in a file on the personality project server.

```
datafilename="http://personality-project.org/r/datasets/recall1.data"
data=read.table(datafilename,header=TRUE)
data    #show the data
```

We can use the "stack() function to arrange the data in the correct manner. We then need to create a new data.frame (recall.df) to attach the correct labels to the correct conditions. This seems more complicated than it really is (although it is fact somewhat tricky). It is useful to list the data after the data frame operation to make sure that we did it correctly. (This and the next example are adapted from Baron and Li's page.) We make use of the rep(), c(), and factor() functions.

rep (operation,number) repeats an operation number times
c(x,y) forms a vector with x and y elements
factor (vector) converts a numeric vector into factors for an ANOVA

```
sums=data[,9:12]   #get the summary numbers
stackeds=stack(sums)   #convert to a column vector to do anova with repeated measures
#stackeds            #show the data as they are now reorganized

numcases=27                  #How many subjects are there?
numvariables=4               #How many repeated measures are there?

recall.df=data.frame(recall=stackeds,
    subj=factor(rep(paste("subj", 1:numcases, sep=""), numvariables)),
    time=factor(rep(rep(c("short", "long"), c(numcases, numcases)), 2)),
    study=factor(rep(c("d45", "d90"), c(numcases*2, numcases*2))))

recall.df     #show the results of stacking and forming the data.frame
              #now, do the within subjects ANOVA and show the results
recall.aov= aov(recall.values ~ time * study + Error(subj/(time * study)), data=recall.df)
summary(recall.aov)
print(model.tables(recall.aov,"means"),digits=3)
```

results in the following output:

```
sums=data[,9:12]   #get the summary numbers
> stackeds=stack(sums)   #convert to a column vector to do anova with repeated measures
> #stackeds            #show the data as they are now reorganized
>
> numcases=27                  #How many subjects are there?
> numvariables=4               #How many repeated measures are there?
>
> recall.df=data.frame(recall=stackeds,
+     subj=factor(rep(paste("subj", 1:numcases, sep=""), numvariables)),
+     time=factor(rep(rep(c("short", "long"), c(numcases, numcases)), 2)),
+     study=factor(rep(c("d45", "d90"), c(numcases*2, numcases*2))))
> recall.df
    recall.values recall.ind   subj  time study
1              13         ss  subj1 short   d45
2              25         ss  subj2 short   d45
.snip ..
25             17         ss subj25 short   d45
26             19         ss subj26 short   d45
27             19         ss subj27 short   d45
28             10         sl  subj1  long   d45
```

```
29             13         sl  subj2  long   d45
30             16         sl  subj3  long   d45
.snip..
79             21         ls subj25 short   d90
80             22         ls subj26 short   d90
81             21         ls subj27 short   d90
82             17         ll  subj1  long   d90
.snip  ...
108            20         ll subj27  long   d90
> recall.aov= aov(recall.values ~ time * study + Error(subj/(time * study)), data=recall.df)
> summary(recall.aov)

Error: subj
          Df  Sum Sq Mean Sq F value Pr(>F)
Residuals 26 1175.35   45.21

Error: subj:time
          Df  Sum Sq Mean Sq F value Pr(>F)
time       1   1.333   1.333  0.2249 0.6393
Residuals 26 154.167   5.929

Error: subj:study
          Df  Sum Sq Mean Sq F value    Pr(>F)
study      1 166.259 166.259  14.997 0.0006512 ***
Residuals 26 288.241  11.086
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Error: subj:time:study
           Df  Sum Sq Mean Sq F value  Pr(>F)
time:study  1  71.704  71.704  6.8592 0.01452 *
Residuals  26 271.796  10.454
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
> print(model.tables(recall.aov,"means"),digits=3)
Tables of means
Grand mean

18.53704

 time
    long short
    18.4  18.6
rep 54.0  54.0

 study
     d45  d90
    17.3 19.8
rep 54.0 54.0

 time:study
       study
time    d45   d90
  long  16.37 20.48
  rep   27.00 27.00
  short 18.22 19.07
  rep   27.00 27.00
```

We can use this same procedure of stacking and forming a data frame on the raw data and consider replications as part of the design. I have written this code in a generic form so that it is (somewhat) easier to use for other data sets. The nex three analyses compare the effects of including the subject replication as part of the design.

```
raw=data[,1:8]              #just trial data
#First set some specific paremeters for the analysis -- this allows
numcases=27                  #How many subjects are there?
numvariables=8               #How many repeated measures are there?
numreplications=2            #How many replications/subject?
numlevels1=2                 #specify the number of levels for within subject variable 1
numlevels2=2                 #specify the number of levels for within subject variable 2

stackedraw=stack(raw)        #convert the data array into a vector
                             #add the various coding variables for the conditions
                             #make sure to check that this coding is correct
recall.raw.df=data.frame(recall=stackedraw,
     subj=factor(rep(paste("subj", 1:numcases, sep=""), numvariables)),
     replication=factor(rep(rep(c("1","2"), c(numcases, numcases)), numvariables/numreplications)),
     time=factor(rep(rep(c("short", "long"), c(numcases*numreplications, numcases*numreplications)),numlevels1)),
     study=rep(c("d45", "d90") ,c(numcases*numlevels1*numreplications,numcases*numlevels1*numreplications)))


recall.aov= aov(recall.values ~ time * study + Error(subj/(time * study)), data=recall.raw.df)   #do the ANOVA
summary(recall.aov)                                #show the output
print(model.tables(recall.aov,"means"),digits=3)   #show the cell means for the anova table

#compare with the complete analysis
recall.aov= aov(recall.values ~ time * study*replication + Error(subj/(time * study * replication)), data=recall.raw.df)   #do the ANOVA
summary(recall.aov)                                #show the output
print(model.tables(recall.aov,"means"),digits=3)   #show the cell means for the anova table


> recall.aov= aov(recall.values ~ time * study + Error(subj/(time * study)), data=recall.raw.df)   #do the ANOVA
> summary(recall.aov)                                #show the output

Error: subj
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 26 587.68   22.60
```

```
Error: subj:time
          Df Sum Sq Mean Sq F value Pr(>F)
time       1  0.667   0.667  0.2249 0.6393
Residuals 26 77.083   2.965

Error: subj:study
          Df  Sum Sq Mean Sq F value    Pr(>F)
study      1  83.130  83.130  14.997 0.0006512 ***
Residuals 26 144.120   5.543
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Error: subj:time:study
           Df  Sum Sq Mean Sq F value  Pr(>F)
time:study  1  35.852  35.852  6.8592 0.01452 *
Residuals  26 135.898   5.227
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Error: Within
           Df Sum Sq Mean Sq F value Pr(>F)
Residuals 108 586.00    5.43
> print(model.tables(recall.aov,"means"),digits=3)   #show the cell means for the anova table
Tables of means
Grand mean

9.268519

 time
      long  short
      9.21   9.32
rep 108.00 108.00

 study
       d45    d90
      8.65   9.89
rep 108.00 108.00

 time:study
       study
time    d45  d90
  long   8.2 10.2
  rep   54.0 54.0
  short  9.1  9.5
  rep   54.0 54.0
>
> #compare with the complete analysis
> recall.aov= aov(recall.values ~ time * study*replication + Error(subj/(time * study * replication)), data=recall.raw.df)   #do the ANOVA
> summary(recall.aov)                                  #show the output

Error: subj
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 26 587.68   22.60

Error: subj:time
          Df Sum Sq Mean Sq F value Pr(>F)
time       1  0.667   0.667  0.2249 0.6393
Residuals 26 77.083   2.965

Error: subj:study
          Df  Sum Sq Mean Sq F value    Pr(>F)
study      1  83.130  83.130  14.997 0.0006512 ***
Residuals 26 144.120   5.543
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Error: subj:replication
            Df  Sum Sq Mean Sq F value Pr(>F)
replication  1   4.741   4.741  0.7208 0.4036
Residuals   26 171.009   6.577

Error: subj:time:study
           Df  Sum Sq Mean Sq F value  Pr(>F)
time:study  1  35.852  35.852  6.8592 0.01452 *
Residuals  26 135.898   5.227
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Error: subj:time:replication
                 Df Sum Sq Mean Sq F value    Pr(>F)
time:replication  1 88.167  88.167  38.153 1.563e-06 ***
Residuals        26 60.083   2.311
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Error: subj:study:replication
                  Df  Sum Sq Mean Sq F value  Pr(>F)
study:replication  1  16.667  16.667  3.8662 0.06003 .
Residuals         26 112.083   4.311
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Error: subj:time:study:replication
                       Df  Sum Sq Mean Sq F value Pr(>F)
time:study:replication  1   0.463   0.463  0.0906 0.7657
Residuals              26 132.787   5.107
> print(model.tables(recall.aov,"means"),digits=3)   #show the cell means for the anova table
Tables of means
```

```
Grand mean

9.268519

 time
      long  short
      9.21   9.32
rep 108.00 108.00

 study
       d45    d90
      8.65   9.89
rep 108.00 108.00

 replication
         1      2
      9.12   9.42
rep 108.00 108.00

 time:study
       study
time    d45  d90
  long   8.2 10.2
  rep   54.0 54.0
  short  9.1  9.5
  rep   54.0 54.0

 time:replication
       replication
time    1    2
  long   8.4 10.0
  rep   54.0 54.0
  short  9.8  8.8
  rep   54.0 54.0

 study:replication
     replication
study 1    2
  d45  8.2  9.1
  rep 54.0 54.0
  d90 10.0  9.8
  rep 54.0 54.0

 time:study:replication
, , replication = 1

       study
time    d45   d90
  long   7.07  9.78
  rep   27.00 27.00
  short  9.37 10.26
  rep   27.00 27.00

, , replication = 2

       study
time    d45   d90
  long   9.30 10.70
  rep   27.00 27.00
  short  8.85  8.81
  rep   27.00 27.00
```

Useful R links

- Readings and software:
- Comprehensive R Archive Network (CRAN)
- An introduction to R
- R Studio

- Structural Equation modelling:
- sem
- lavaan
- psych for sem
- EFA and factor extension (fa)

- Multilevel modeling:
- Multilevel
- Linear and Non Linear Mixed Effects nlme
- statsBy

- Item Response Models:
- Latent Trait Model (ltm)
- mirt
- mokken
- irt by factor analysis (irt.fa)

More on the psych package

The psych package is a work in progress. The current released version is 1.3.2. Updates are added sporadically, but usually at least once a quarter. The development version is always available at the pmc repository.

If you want to help us develop our understanding of personality, please take our test at SAPA Project.

As is true of all webpages, this is a work in progress.
Design: HTML5 Up! | Modified by Jason A. French and William Revelle
Version of May 20, 2013