

8 Day 3: The general linear model and its special cases

8.1 Correlation

One of the most simple descriptive statistics is the Pearson Product Moment Correlation Coefficient (PPMCC, Pearson (1920)). To find the correlations between a set of variables, the `cor` may be used. The primary decision to make is what to do with missing values. Options to `cor` include using complete cases and pair-wise deletion of missing variables. If not specified, any missing value will cause an error. In the example below, pairwise deletion is requested and the output is rounded to two decimal places.

```
> data(sat.act)
> round(cor(sat.act, use = "pairwise"), 2)

      gender education   age   ACT  SATV  SATQ
gender    1.00     0.09 -0.02 -0.04 -0.02 -0.17
education 0.09     1.00  0.55  0.15  0.05  0.03
age       -0.02     0.55  1.00  0.11 -0.04 -0.03
ACT        -0.04     0.15  0.11  1.00  0.56  0.59
SATV       -0.02     0.05 -0.04  0.56  1.00  0.64
SATQ      -0.17     0.03 -0.03  0.59  0.64  1.00
```

To test the statistical significance of single correlation, the `cor.test` function can be used, although it is probably more convenient to use the `corr.test` function in the *psych* package. This will report the correlations between all the variables, the sample sizes of pairwise observations and the probability values associated with that correlation. Note that these probabilities are not corrected for multiple comparisons.

```
> with(sat.act, cor.test(age, education))

      Pearson's product-moment correlation

data:  age and education
t = 17.3204, df = 698, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.4942471 0.5980736
sample estimates:
      cor
0.5482695

> corr.test(sat.act)
```

```

Call:corr.test(x = sat.act)
Correlation matrix
      gender education  age  ACT  SATV  SATQ
gender  1.00      0.09 -0.02 -0.04 -0.02 -0.17
education 0.09      1.00  0.55  0.15  0.05  0.03
age      -0.02      0.55  1.00  0.11 -0.04 -0.03
ACT      -0.04      0.15  0.11  1.00  0.56  0.59
SATV     -0.02      0.05 -0.04  0.56  1.00  0.64
SATQ     -0.17      0.03 -0.03  0.59  0.64  1.00
Sample Size
      gender education age ACT SATV SATQ
gender  700      700 700 700  700  687
education 700      700 700 700  700  687
age      700      700 700 700  700  687
ACT      700      700 700 700  700  687
SATV     700      700 700 700  700  687
SATQ     687      687 687 687  687  687
Probability value
      gender education  age  ACT SATV SATQ
gender  0.00      0.02 0.58 0.33 0.62 0.00
education 0.02      0.00 0.00 0.00 0.22 0.36
age      0.58      0.00 0.00 0.00 0.26 0.37
ACT      0.33      0.00 0.00 0.00 0.00 0.00
SATV     0.62      0.22 0.26 0.00 0.00 0.00
SATQ     0.00      0.36 0.37 0.00 0.00 0.00

```

To test the difference between two correlations is a bit more complicated, because it depends upon whether the correlations are independent or dependent, and how many other variables are involved (Steiger, 1980). That is, do the correlations come from two different samples (independent), are they correlations with the same third variable, or are they two correlations from the same sample but between different variables? All of these cases are tested in the `r.test` function.

```

> r.test(50, 0.3)

Correlation tests
Call:r.test(n = 50, r12 = 0.3)
Test of significance of a correlation
t value 2.18 with probability < 0.034
and confidence interval 0.02 0.53

> r.test(30, 0.4, 0.6)

```

```

Correlation tests
Call:r.test(n = 30, r12 = 0.4, r34 = 0.6)
Test of difference between two independent correlations
z value 0.99 with probability 0.32

> r.test(103, 0.4, 0.5, 0.1)

Correlation tests
Call:r.test(n = 103, r12 = 0.4, r34 = 0.5, r23 = 0.1)
Test of difference between two correlated correlations
t value -0.89 with probability < 0.37

> r.test(103, 0.5, 0.6, 0.7, 0.5, 0.5, 0.8)

Correlation tests
Call:r.test(n = 103, r12 = 0.5, r34 = 0.6, r23 = 0.7, r13 = 0.5, r14 = 0.5,
r24 = 0.8)
Test of difference between two dependent correlations
z value -1.2 with probability 0.23

```

8.2 Regression

The idea that there is linear relationship between two variables is the basis of the correlation coefficient as well as the linear regression model. Important generalizations of the linear regression model include multiple regression, as well as the non-linear forms of regression such as logistic or Poisson regression. Before discussing the variants, consider first just normal linear regression.

To predict a single variable based upon one other variable was considered in the graphics section above. Consider three cases: a) what is the slope of the relationship between two variables, b) what is the direct effect of multiple variables in predicting a variable, and c) does the relationship between two variables depend upon a third variable.

The first case is the easiest

```

> mod1 <- lm(SATQ ~ SATV, data = sat.act)
> summary(mod1)

```

```

Call:
lm(formula = SATQ ~ SATV, data = sat.act)

```

```

Residuals:
    Min       1Q   Median       3Q      Max
-302.105  -46.477    2.403   51.319  282.845

```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	207.52528	18.57250	11.17	<2e-16 ***
SATV	0.65763	0.02983	22.05	<2e-16 ***

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

Residual standard error: 88.5 on 685 degrees of freedom

(13 observations deleted due to missingness)

Multiple R-squared: 0.4151, Adjusted R-squared: 0.4143

F-statistic: 486.2 on 1 and 685 DF, p-value: < 2.2e-16

The next asks for the effect of two (or more) variables upon another

```
> mod2 <- lm(SATQ ~ SATV + gender, data = sat.act)
> summary(mod2)
```

Call:

```
lm(formula = SATQ ~ SATV + gender, data = sat.act)
```

Residuals:

Min	1Q	Median	3Q	Max
-291.274	-50.457	5.635	51.891	295.343

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	269.89975	21.65705	12.462	< 2e-16 ***
SATV	0.65454	0.02925	22.375	< 2e-16 ***
gender	-36.80114	6.91400	-5.323	1.39e-07 ***

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

Residual standard error: 86.79 on 684 degrees of freedom

(13 observations deleted due to missingness)

Multiple R-squared: 0.4384, Adjusted R-squared: 0.4367

F-statistic: 267 on 2 and 684 DF, p-value: < 2.2e-16

The third asks does the relationship between two variables (e.g., SATV and SATQ) depend upon a third variable (e.g., gender). That is, is there an interaction between the two (or more) independent variables. Because an interaction can be thought of as the product of two variables, the simple way of doing this is find the product between gender and SATV. Unfortunately this confounds the interaction term with the main effects.

```
> mod3 <- lm(SATQ ~ SATV * gender, data = sat.act)
> summary(mod3)
```

Call:

```
lm(formula = SATQ ~ SATV * gender, data = sat.act)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-294.423	-49.876	5.577	53.210	291.100

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	211.19986	64.94501	3.252	0.00120 **
SATV	0.75009	0.10387	7.221	1.38e-12 ***
gender	-0.99528	37.98214	-0.026	0.97910
SATV:gender	-0.05835	0.06086	-0.959	0.33804

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

Residual standard error: 86.79 on 683 degrees of freedom

(13 observations deleted due to missingness)

Multiple R-squared: 0.4391, Adjusted R-squared: 0.4367

F-statistic: 178.3 on 3 and 683 DF, p-value: < 2.2e-16

Although the interaction effect is correct, the main effects of SATV and gender are not. To correct this we need to first zero center the data, and then do the analysis. To zero center just means to subtract the mean from each variable. It is done by the `scale` function. There are two problems with `scale`. Its default is to standardize (i.e., to subtract the mean and the divide by the standard deviation, and it returns a matrix when `lm` requires a data frame. Both of these problems are easy to rectify:

```
> cent.data <- data.frame(scale(sat.act, scale = FALSE))
> mod4 <- lm(SATQ ~ SATV * gender, data = cent.data)
> summary(mod4)
```

Call:

```
lm(formula = SATQ ~ SATV * gender, data = cent.data)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-294.423	-49.876	5.577	53.210	291.100

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.26696	3.31211	-0.081	0.936
SATV	0.65398	0.02926	22.350	< 2e-16 ***
gender	-36.71820	6.91495	-5.310	1.48e-07 ***
SATV:gender	-0.05835	0.06086	-0.959	0.338

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

Residual standard error: 86.79 on 683 degrees of freedom

(13 observations deleted due to missingness)

Multiple R-squared: 0.4391, Adjusted R-squared: 0.4367

F-statistic: 178.3 on 3 and 683 DF, p-value: < 2.2e-16

Compare mod4 (the correct model) with mod3 (the incorrect model). Note that the interaction effects are exactly equal, but that the main effects are very different.

In the previous example, by entering the produce of the two independent variables, the interaction and main effects were all implied. More complicated regression models can be considered, with just some interaction terms specified.

```
> mod5 <- lm(SATQ ~ SATV + ACT + gender * education, data = cent.data)
```

```
> summary(mod5)
```

Call:

```
lm(formula = SATQ ~ SATV + ACT + gender * education, data = cent.data)
```

Residuals:

Min	1Q	Median	3Q	Max
-305.78	-46.07	5.67	51.82	261.21

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.14552	3.10578	0.047	0.963
SATV	0.46905	0.03306	14.187	< 2e-16 ***
ACT	7.86001	0.78567	10.004	< 2e-16 ***
gender	-34.07509	6.49943	-5.243	2.11e-07 ***
education	-2.56801	2.23493	-1.149	0.251
gender:education	-5.45345	4.42642	-1.232	0.218

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

Residual standard error: 81.1 on 681 degrees of freedom

(13 observations deleted due to missingness)

Multiple R-squared: 0.5117, Adjusted R-squared: 0.5081
F-statistic: 142.7 on 5 and 681 DF, p-value: < 2.2e-16

Using the `mat.regress` it is possible to do regressions based upon the correlation matrix rather than the raw data, and to do more than one analysis at a time. This is discussed in detail in the vignette for the *psych* package. If the number of observations is specified, the function will return the traditional statistical tests, otherwise, it just reports the β weights and R values.

```
> r <- cor(sat.act, use = "pairwise")  
> mat.regress(r, c(1:3), c(4:6))
```

```
$beta  
      ACT SATV SATQ  
gender -0.05 -0.03 -0.18  
education 0.14 0.10 0.10  
age      0.03 -0.10 -0.09
```

```
$R  
  ACT SATV SATQ  
0.16 0.10 0.19
```

```
$R2  
  ACT SATV SATQ  
0.03 0.01 0.04
```

8.3 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.)⁴

8.4 The t-test

The t-test Student (1908) is the most simple comparison of means for small samples. As the sample size increases, the t-test tends towards a z test. Consider a data set from the web with three dosage levels of a drug. Compare Dosage levels “a” and “c”.

⁴The first five examples are taken from an online tutorial written at Northwestern by Teaching Assistants for a Research Methods in Psychology course taught by Roger Ratcliff.

```
> datafilename = "http://personality-project.org/r/datasets/R.appendix1.data"
> data.ex1 = read.table(datafilename, header = T)
> data.ex1
```

	Dosage	Alertness
1	a	30
2	a	38
3	a	35
4	a	41
5	a	27
6	a	24
7	b	32
8	b	26
9	b	31
10	b	29
11	b	27
12	b	35
13	b	21
14	b	25
15	c	17
16	c	21
17	c	20
18	c	19

```
> dose.2 <- subset(data.ex1, Dosage != "b")
> t.test(Alertness ~ Dosage, data = dose.2)
```

Welch Two Sample t-test

```
data: Alertness by Dosage
t = 4.6907, df = 5.956, p-value = 0.003424
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 6.325685 20.174315
sample estimates:
mean in group a mean in group c
      32.50      19.25
```

The generalization of the t-test was the Analysis of Variance, designed in particular for the case of equal sized samples for agricultural field stations.

8.4.1 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`.

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:

```
> datafilename = "http://personality-project.org/R/datasets/R.appendix1.data"
> data.ex1 = read.table(datafilename, header = T)
> aov.ex1 = aov(Alertness ~ Dosage, data = data.ex1)
> summary(aov.ex1)
```

	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)
```

Tables of means

Grand mean

27.66667

Dosage

	a	b	c
32.5	28.2	19.2	
rep	6.0	8.0	4.0

Graphical output is useful to summarize these results.

8.4.2 Two way ANOVA

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

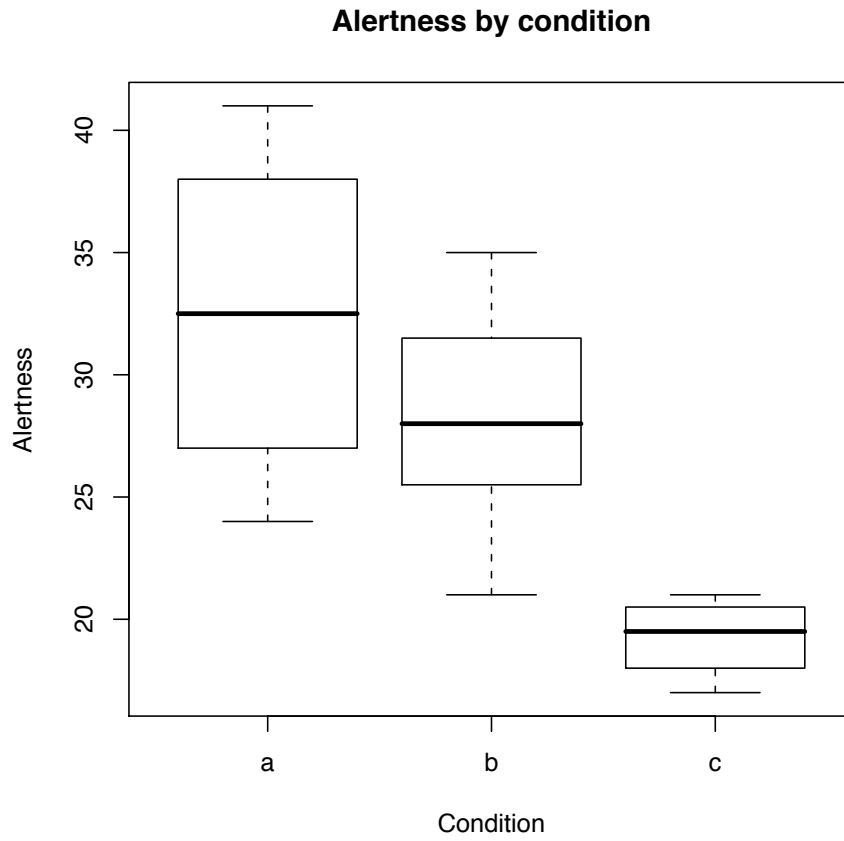


Figure 17: Summarizing the results of a one way ANOVA

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)
> data.ex2
```

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.ex2)
> summary(aov.ex2)
```

	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)
```

Tables of means

Grand mean

14.0625

Gender

```
Gender
  f    m
16.25 11.88
```

```
Dosage
Dosage
  a    b
13.50 14.62
```

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

8.4.3 1 way ANOVA- Within Subjects

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. 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 term `Error(Subject/Valence)` to the factor Valence, as shown below. 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).

```
> datafilename = "http://personality-project.org/r/datasets/R.appendix3.data"
> data.ex3 = read.table(datafilename, header = T)
> data.ex3
```

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)
```

Tables of means

Grand mean

26.46667

Valence

Valence

Neg Neu Pos

27.8 11.6 40.0

8.4.4 Two-way Within Subjects ANOVA

The online appendix4 contains the data from an experiment where five subjects were tested on their recall of words of differing valences. There were two different memory tasks: free

or cued recall. Thus, there were 2 independent factors: Valence (3 levels) and Task (2 levels). Again, Subject serves as a random factor. Enter the data into a file entitled ex4 and run the following analysis:

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)  
> data.ex4
```

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.ex4)
> 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)

Tables of means
Grand mean

11.8

Task
Task
Cued Free
12.8 10.8

Valence
Valence
Neg Neu Pos
11.0 12.1 12.3

```

```

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

```

Now it's time to get serious. The online 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))
```

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)
> head(data.ex5)

```

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

```
> tail(data.ex5)
```

Obs	Subject	Gender	Dosage	Task	Valence	Recall	
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(Recall ~ (Task * Valence * Gender * Dosage) + Error(Subject/(Task * Valence
+ (Gender * Dosage), data.ex5)

```

> 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		

8.4.5 Reorganizing the data for repeated measures ANOVA

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/recall11.data"
> data = read.table(datafilename, header = TRUE)
> data
```

	ss1	ss2	s11	s12	ls1	ls2	ll1	LL2	ss	s1	ls	ll
1	8	5	4	6	10	6	10	7	13	10	16	17
2	11	14	0	13	12	14	13	14	25	13	26	27
3	12	5	9	7	8	10	8	10	17	16	18	18
4	8	8	9	9	9	8	8	11	16	18	17	19
5	11	10	10	8	11	10	8	11	21	18	21	19
6	9	10	0	7	9	10	10	13	19	7	19	23
7	12	12	12	12	13	10	10	15	24	24	23	25
8	9	8	5	10	8	7	6	9	17	15	15	15
9	10	11	8	10	9	8	14	13	21	18	17	27
10	9	8	11	9	8	10	5	10	17	20	18	15
11	12	12	3	12	10	15	14	15	24	15	25	29
12	9	8	6	5	13	6	10	10	17	11	19	20
13	5	4	0	5	3	3	4	5	9	5	6	9
14	5	10	6	4	12	10	11	13	15	10	22	24
15	9	9	13	10	13	8	11	11	18	23	21	22
16	9	11	5	12	6	10	11	11	20	17	16	22
17	9	7	8	10	11	6	11	12	16	18	17	23
18	9	9	12	12	15	10	9	13	18	24	25	22
19	11	10	8	12	11	10	12	9	21	20	21	21
20	11	8	10	10	11	10	9	10	19	20	21	19
21	9	8	7	11	9	6	11	10	17	18	15	21
22	10	9	10	8	7	8	9	10	19	18	15	19
23	9	10	5	10	10	10	12	7	19	15	20	19
24	8	7	0	8	12	6	8	11	15	8	18	19
25	10	7	8	7	13	8	11	9	17	15	21	20
26	10	9	10	11	12	10	10	9	19	21	22	19

27 9 10 12 13 12 9 9 11 19 25 21 20

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]
> stackeds = stack(sums)
> numcases = 27
> numvariables = 4
> 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[c(1:4, 26:30, 53:58, 105:108), ]
```

	recall.values	recall.ind	subj	time	study
1	13	ss	subj1	short	d45
2	25	ss	subj2	short	d45
3	17	ss	subj3	short	d45
4	16	ss	subj4	short	d45
26	19	ss	subj26	short	d45
27	19	ss	subj27	short	d45
28	10	s1	subj1	long	d45
29	13	s1	subj2	long	d45
30	16	s1	subj3	long	d45
53	21	s1	subj26	long	d45
54	25	s1	subj27	long	d45
55	16	ls	subj1	short	d90
56	26	ls	subj2	short	d90
57	18	ls	subj3	short	d90
58	17	ls	subj4	short	d90
105	19	l1	subj24	long	d90
106	20	l1	subj25	long	d90
107	19	l1	subj26	long	d90

```

108          20          11 subj27 long d90
> recall.aov = aov(recall.values ~ time * study + Error(subj/(time * study)), data = recall)
> 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
time
long short
18.43 18.65

study
study
d45 d90

```

17.30 19.78

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

8.5 Multi-level models as an alternative to repeated measures ANOVA

Coming soon.

9 Day 4: Multivariate analysis

9.1 Factor analysis and Principal Components Analysis

9.2 Cluster Analysis, Multidimensional Scaling

9.3 Structural Equation Modeling

10 Day 5: R as a programming language

10.1 R in the lab

10.2 R in the classroom

10.3 Using R and Latex or OpenOffice to prepare documents

\LaTeX is a text processing and formatting language that can be combined with the `Sweave` function in R to integrate statistics within a manuscript. This is also possible to do with OpenOffice.

11 Various web resources

<http://www.rseek.org/> R seek: a search engine for R