An introduction to R: Part 2

William Revelle
Northwestern University
Evanston, Illinois USA

March, 2024

Outline

Part I: What is R, where did it come from, why use it

- Installing R and adding packages: the building blocks of R

Part II: A brief introduction – an overview

- R is just a fancy (very fancy) calculator
- Descriptive data analysis
- Some inferential analysis

Outline of Part II

- Basic R: A brief example
  - Basic R capabilities: Calculation, Statistical tables
  - Basic Graphics
- A brief example of exploratory and confirmatory data analysis
  - Data preparation, descriptive statistics, data cleaning, correlation plots: (Examples part ii)
  - Inferential statistics
- Multiple regression modeling and graphics
- Basic statistics and graphics
  - 4 steps: read, explore, test, graph
  - Foreign files
- Basic descriptive statistics and graphics
  - Graphic displays
  - Correlations
- Inferential statistics
  - The t-test
  - ANOVA

Basic R commands – remember don't enter the >

R is just a fancy calculator. Add, subtract, sum, products, group

```
> 2 + 2          #sum two numbers

[1] 4       #show the output

> 3^4       #3 raised to the 4th

[1] 81      #that was easy

> sum(1:10)   #find the sum of the first 10 numbers

[1] 55     #the answer

> prod(c(1, 2, 3, 5, 7)) #the product of the concatenated (c) numbers

[1] 210   #Note how we combined product with concatenate
```

It is also a statistics table (the normal distribution, the t, the F, the χ^2 distribution, the xyz distribution)

```
> pnorm(q = 1)   #the probability of a normal with value of 1 sd

[1] 0.8413447   #
```

```
> pt(q = 2, df = 20) #what about the probability of a t-test value of 2 with
```

R is a set of distributions. Don't buy a stats book with tables!

Table: To obtain the density, prefix with *d*, probability with *p*, quantiles with *q* and to generate random values with *r*. (e.g., the normal distribution may be chosen by using dnorm, pnorm, qnorm, or rnorm.) Each function can be modified with various parameters.

Distribution	base name	P 1	P 2	P 3	example application
Normal	norm	mean	sigma		Most data
Multivariate normal	mvnorm	mean	r	sigma	Most data
Log Normal	lnorm	log mean	log sigma		income or reaction time
Uniform	unif	min	max		rectangular distributions
Binomial	binom	size	prob		Bernuilli trials (e.g. coin flips)
Student's t	t	df		nc	Finding significance of a t-test
Multivariate t	mvt	df	corr	nc	Multivariate applications
Fisher's F	f	df1	df2	nc	Testing for significance of F test
χ^2	chisq	df		nc	Testing for significance of χ^2
Exponential	exp	rate			Exponential decay
Gamma	gamma	shape	rate	scale	distribution theoryh
Hypergeometric	hyper	m	n	k	
Logistic	logis	location	scale		Item Response Theory
Poisson	pois	lambda			Count data
Weibull	weibull	shape	scale		Reaction time distributions

An example of using r, p, and q for a distributions

R code

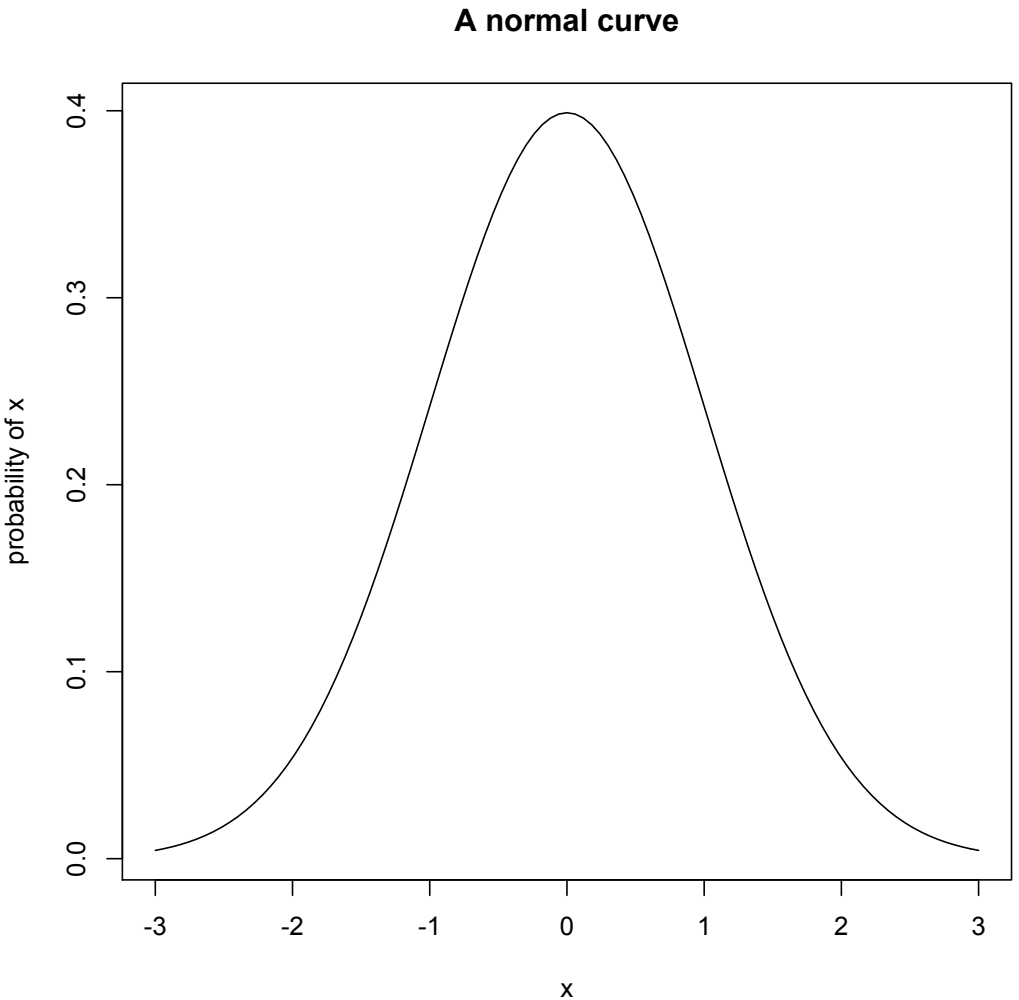
```
set.seed(42) #set the random seed to get the same sequence
x <- rnorm(5) #find 5 randomly distributed normals
round(x,2) #show them, rounded to 2 decimals
round(pnorm(x),2) #show their probabilities to 2 decimals
round(qnorm(pnorm(x)),2)  #find the quantiles of the normal
```

Produces this output

```
> set.seed(42) #set the random seed to get the same sequence
> x <- rnorm(5) #find 5 randomly distributed normals
> round(x,2) #show them, rounded to 2 decimals
[1]  1.37 -0.56  0.36  0.63  0.40
> round(pnorm(x),2) #show their probabilities to 2 decimals
[1] 0.91 0.29 0.64 0.74 0.66
> round(qnorm(pnorm(x)),2)  #find the quantiles of the normal
[1]  1.37 -0.56  0.36  0.63  0.40
```

See (Example 2)

R can draw distributions

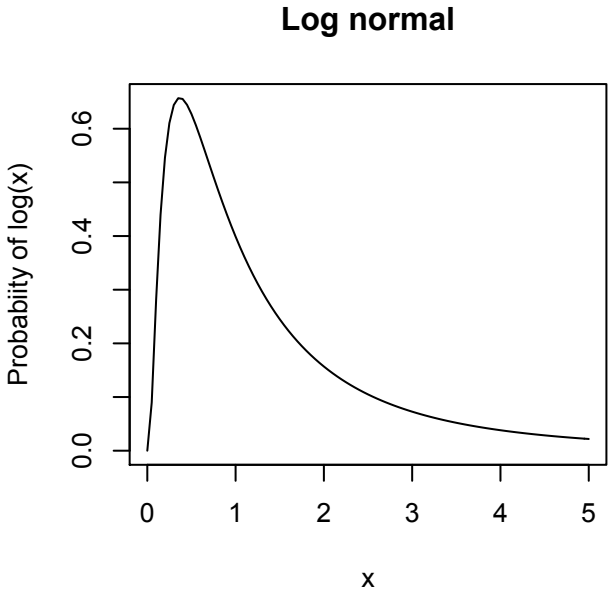


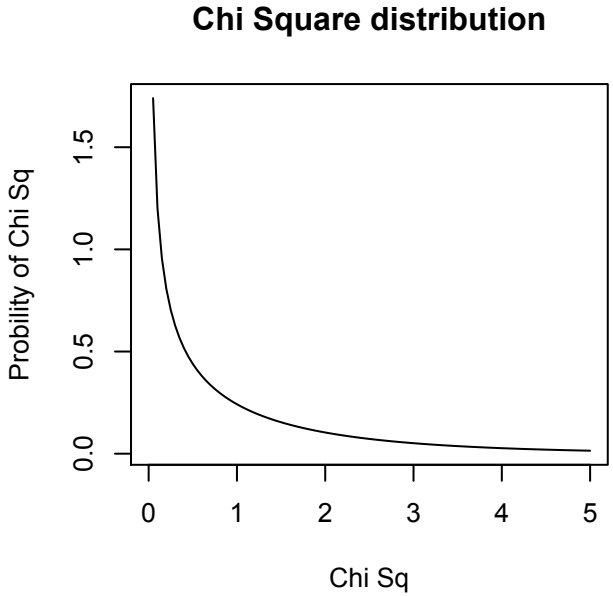
(Example 3)

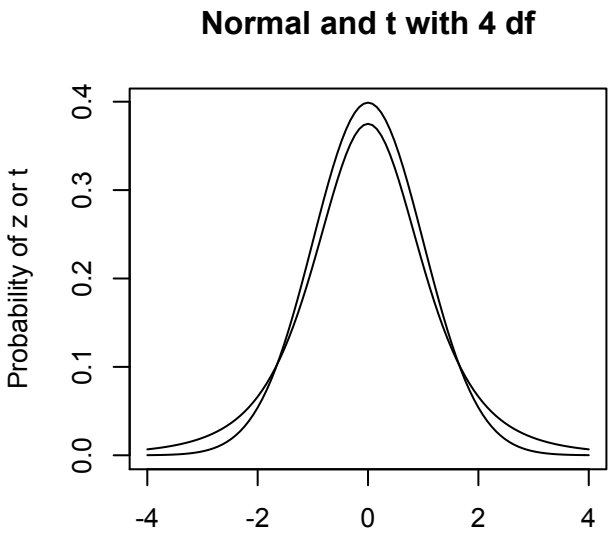
We do this by using the `curve` function to which we pass the values of the `dnorm` function.

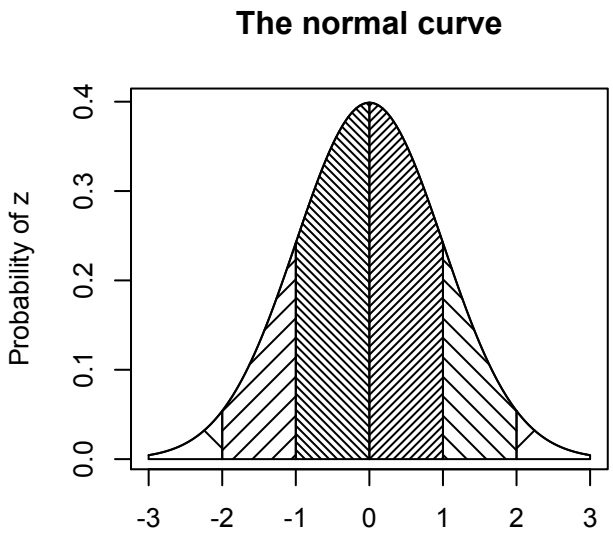
```
curve(dnormal(x),-3,3,
ylab="probability of
x",main="A normal
curve")
```

R can draw more interesting distributions









R is also a graphics calculator

R code

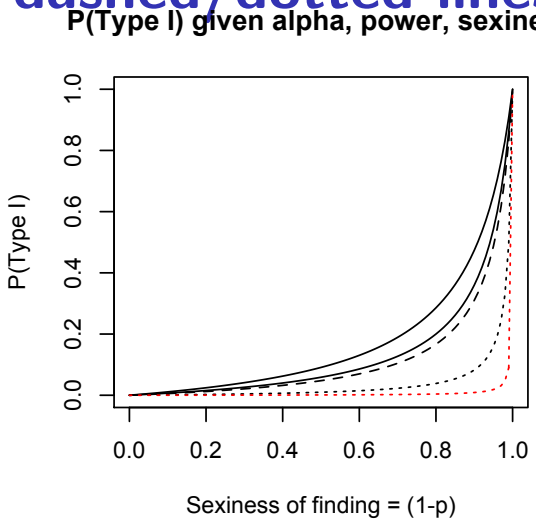
```
op <- par(mfrow=c(2,2))        #set up a 2 x 2 graph
curve(dlnorm(x),0,5,ylab='Probabiity of log(x)',main='Log normal')
curve(dchisq(x,1),0,5,ylab='Probility of Chi Sq',xlab='Chi Sq',main='Chi Square distribution
curve(dnorm(x),-4,4,ylab='Probability of z or t',xlab='z or t',main='Normal and t with 4 df')
curve(dt(x,4),add=TRUE)
#
#somewhat more complicated
#first draw the normal curve
curve(dnorm(x),-3,3,xlab="",ylab="Probability of z") #the range of x
title(main="The normal curve",outer=FALSE) #the title
#add the cross hatching by using polygons
xvals <-  seq(-3,-2,length=100)  #From -3 to 2 with 100 points
dvals <- dnorm(xvals)
polygon(c(xvals,rev(xvals)),c(rep(0,100),rev(dvals)),density=2,angle=-45)
xvals <-  seq(-2,-1,length=100)
dvals <- dnorm(xvals)
polygon(c(xvals,rev(xvals)),c(rep(0,100),rev(dvals)),density=14,angle=45)
xvals <-  seq(-1,-0,length=100)
dvals <- dnorm(xvals)
polygon(c(xvals,rev(xvals)),c(rep(0,100),rev(dvals)),density=34,angle=-45)
xvals <-  seq(2,3,length=100)
dvals <- dnorm(xvals)
polygon(c(xvals,rev(xvals)),c(rep(0,100),rev(dvals)),density=2,angle=45)
xvals <-  seq(1,2,length=100)
dvals <- dnorm(xvals)
polygon(c(xvals,rev(xvals)),c(rep(0,100),rev(dvals)),density=14,angle=-45)
xvals <-  seq(0,1,length=100)
dvals <- dnorm(xvals)
polygon(c(xvals,rev(xvals)),c(rep(0,100),rev(dvals)),density=34,angle=45)
op <- par(mfrow=c(1,1)) #back to a normal 1 x 1 graph
```

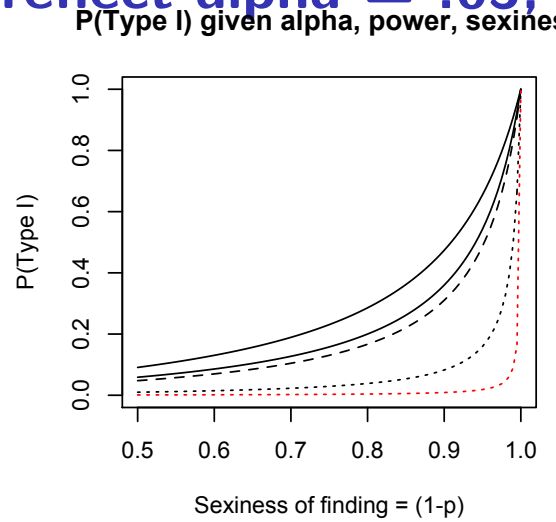
R can help teach with 100s of example data sets.

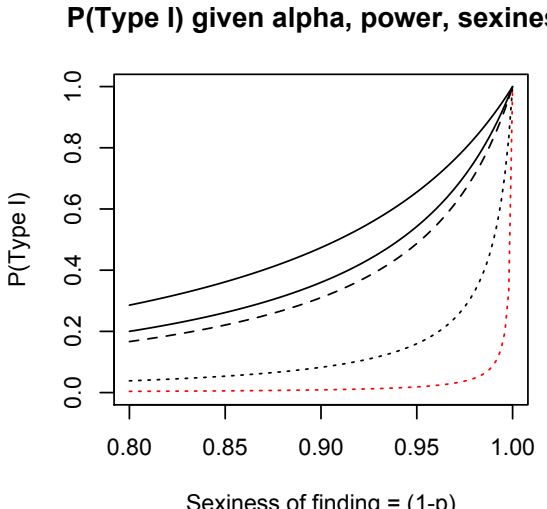
```
> data()

> data(package="psych")
#see the names of the 56
data sets


> data(Titanic)
> ? Titanic


> data(cushny)
> ? cushny

> data(UCBAdmissions)
> ? UCBAdmissions
```

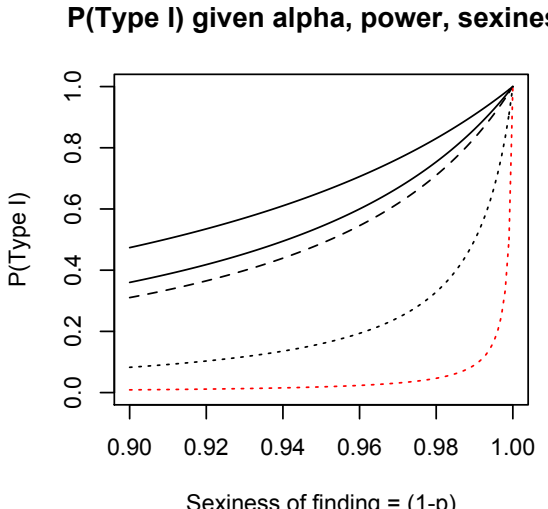
1. This opens up a separate text window and lists all of the data sets in the currently loaded packages.
2. Show the data sets available in a particular package (e.g., *psych*).
3. Gets the particular data set with its help file (e.g., the survival rates on the Titanic cross classified by age, gender and class).
4. Another original data set used by "student" (Gossett) for the t-test.
5. The UC Berkeley example of "sex discrimination" as a Simpson paradox

R can show current statistical concepts:
Type I Errors: It is not the power, it is the prior likelihood
dashed/dotted lines reflect alpha = .05, .01, .001 with power = 1



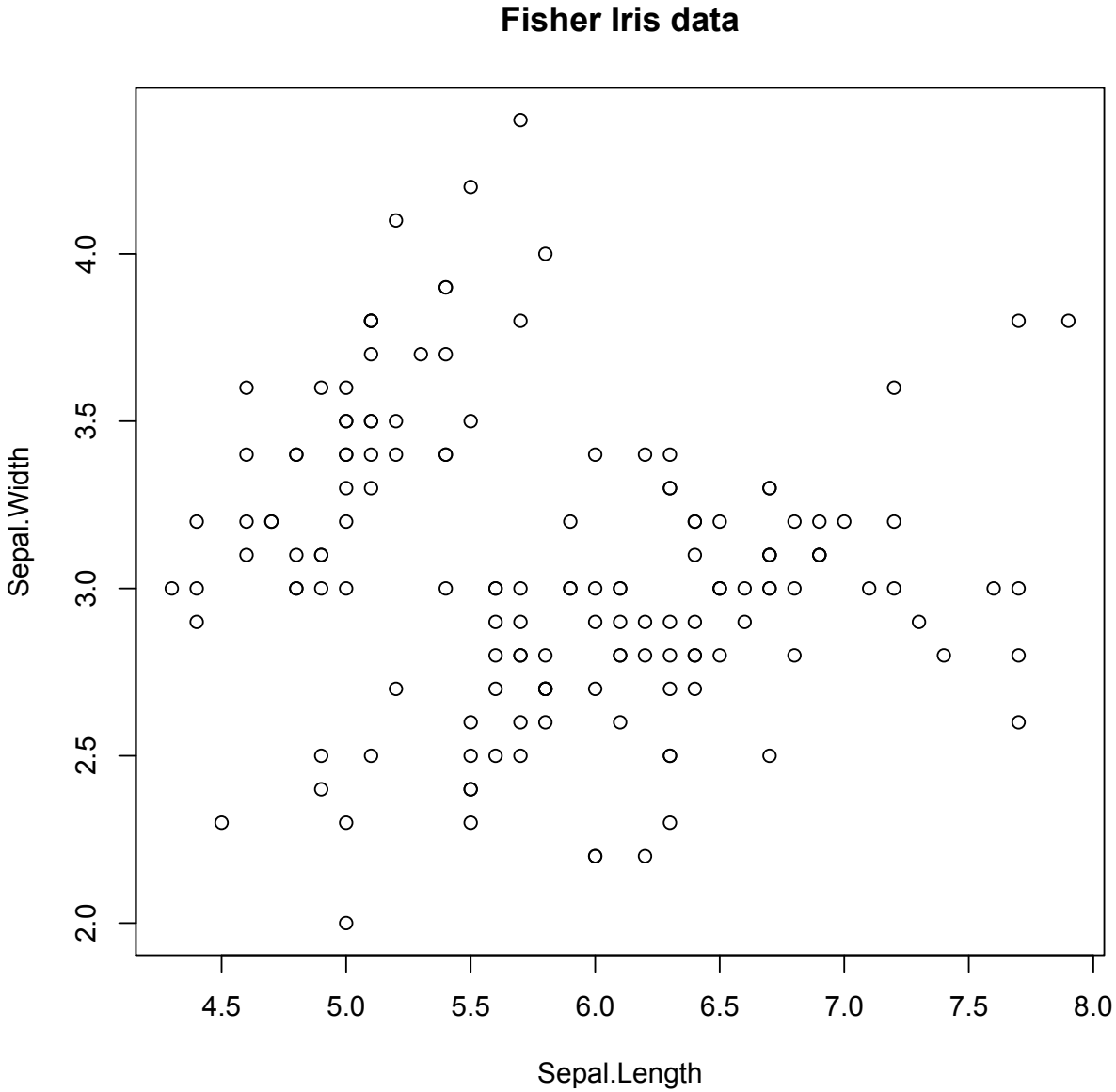






1. Extreme claims require extreme probabilities
2. Given that a finding is "significant", what is the likelihood that it is a Type I error?
3. Depends upon the prior likelihood (the 'sexiness') of the claim.

A simple scatter plot using `plot` with Fisher's Iris data set.



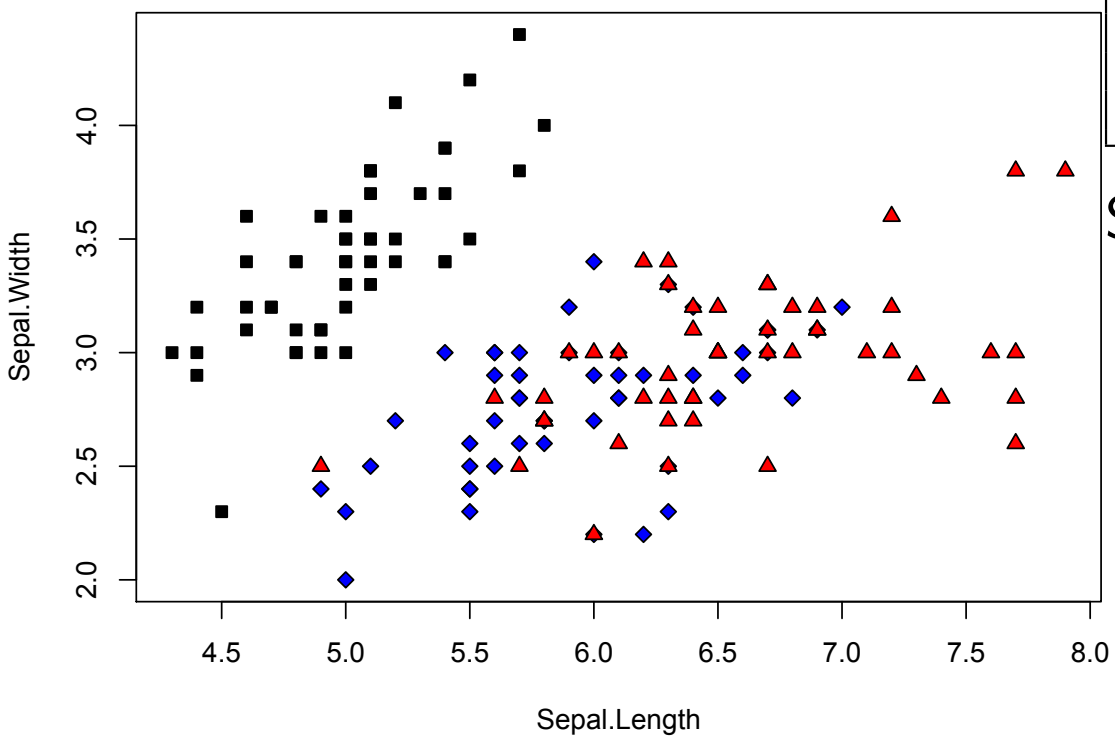
R code

```
plot(iris[1:2],xlab="Sepal.Length",
   ylab="Sepal.Width"
,main="Fisher Iris data")
```

Set parameters

1. xlab for x axis label
2. ylab for y axis label
3. main for title
4. (Example 4)

A simple scatter plot using `plot` with some colors and shapes



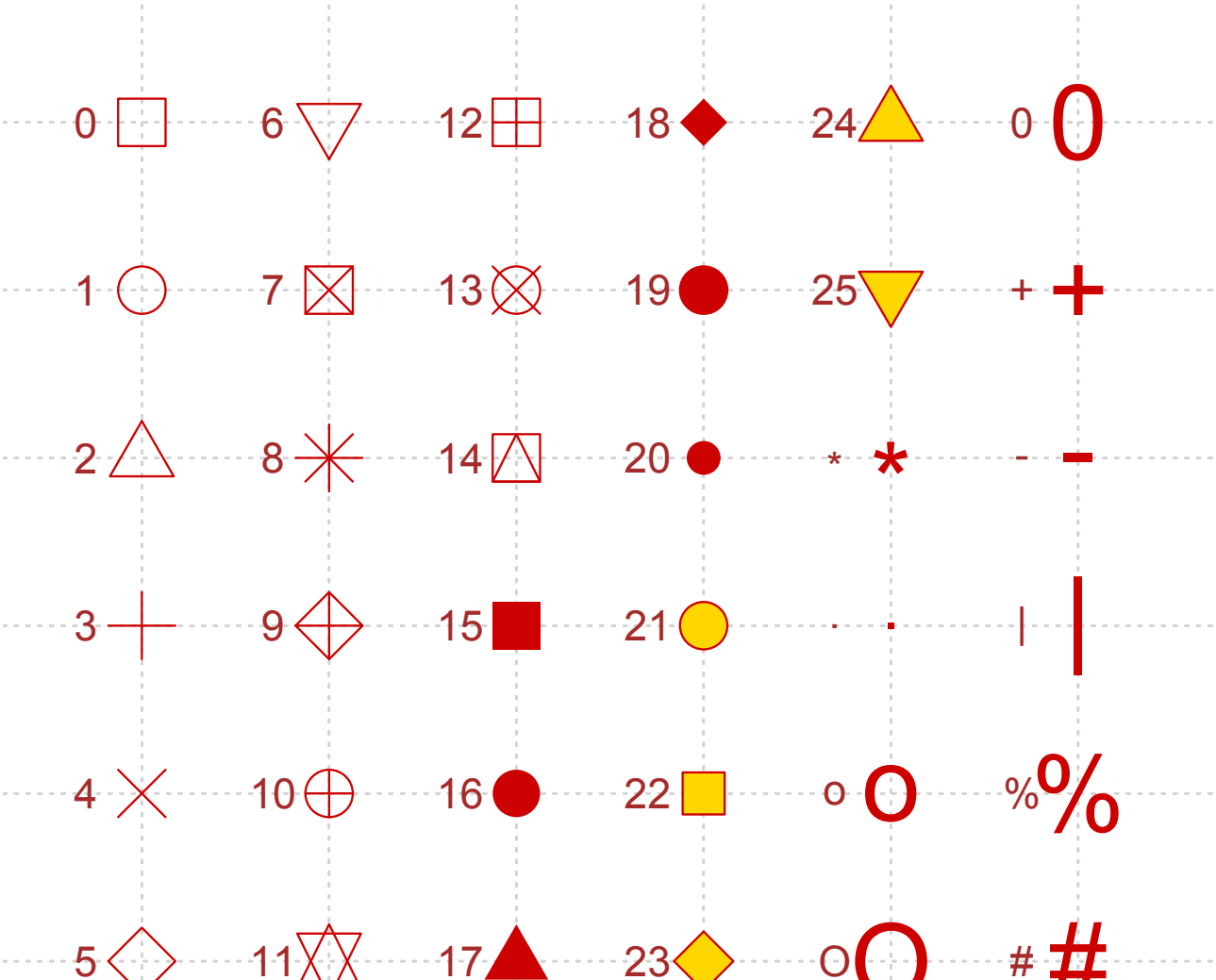
R code

```
plot(iris[1:2],xlab="Sepal.Length",
ylab="Sepal.Width" ,
main="Fisher Iris data with
colors and shapes",
 bg=c("black","blue",
"red")[iris[,"Species"]],
 pch=21+ as.numeric(iris[,5]))
```

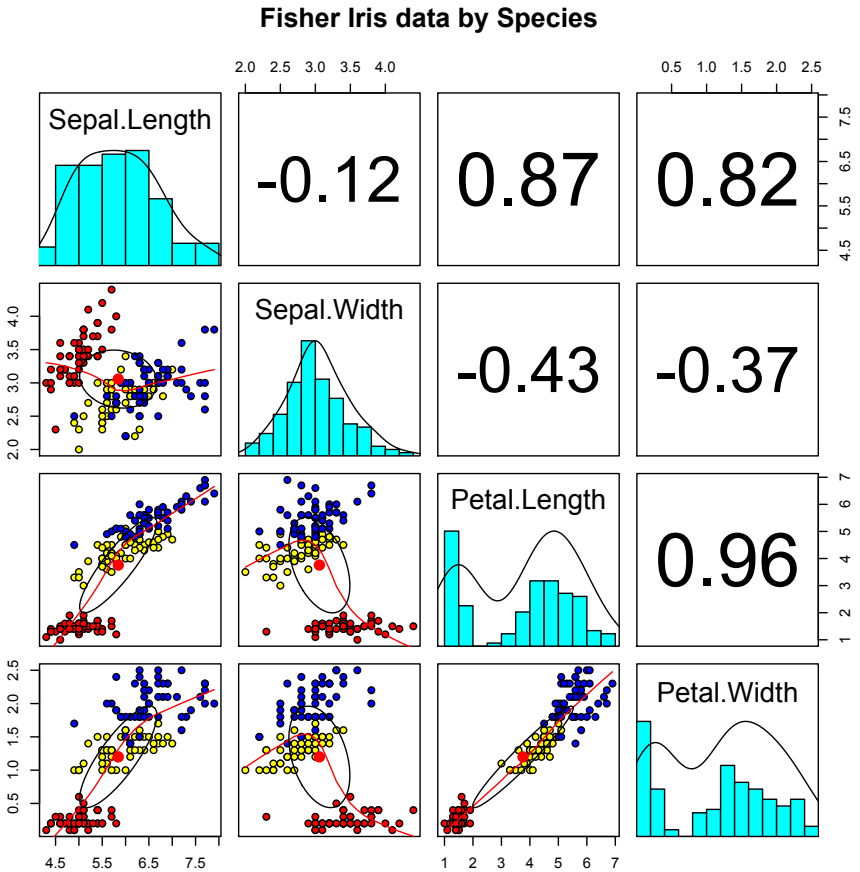
Set parameters

1. bg for background colors of symbols
2. pch chooses the plot character
3. Note how these depend upon iris[,5] which is the species

Show the various graphic options for plot character (pch)



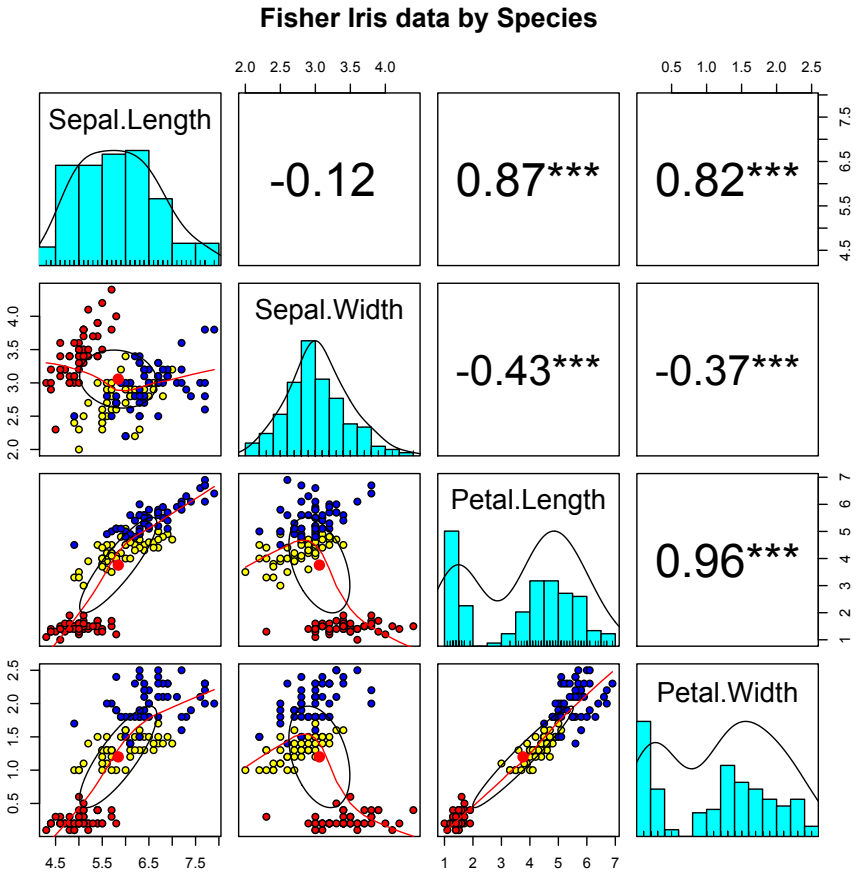
A scatter plot matrix plot with loess regressions using `pairs.panels`



1. Correlations above the diagonal
2. Diagonal shows histograms and densities
3. scatter plots below the diagonal with correlation ellipse
4. locally smoothed (loess) regressions for each pair
5. optional color coding of grouping variables.

```
pairs.panels(iris[1:4],bg=c("red","yellow","blue")
[iris$Species],pch=21,main="Fisher Iris data by
Species")
```

A scatter plot matrix plot with loess regressions using `pairs.panels`



Show "significance" using magic asterisks

```
pairs.panels(iris[1:4],bg=c("red","yellow","blue")
[iris$Species],pch=21,main="Fisher Iris data by
Species",stars=TRUE)
```

A brief example with real data - example 5

1. Get the data
2. Descriptive statistics
   - Graphic
   - Numerical
3. Inferential statistics using the linear model
   - regressions
4. More graphic displays

Get the data and describe it

1. First read the data, either from a built in data set, a local file, a remote file, or from the clipboard.
2. Describe the data using the describe function from *psych*

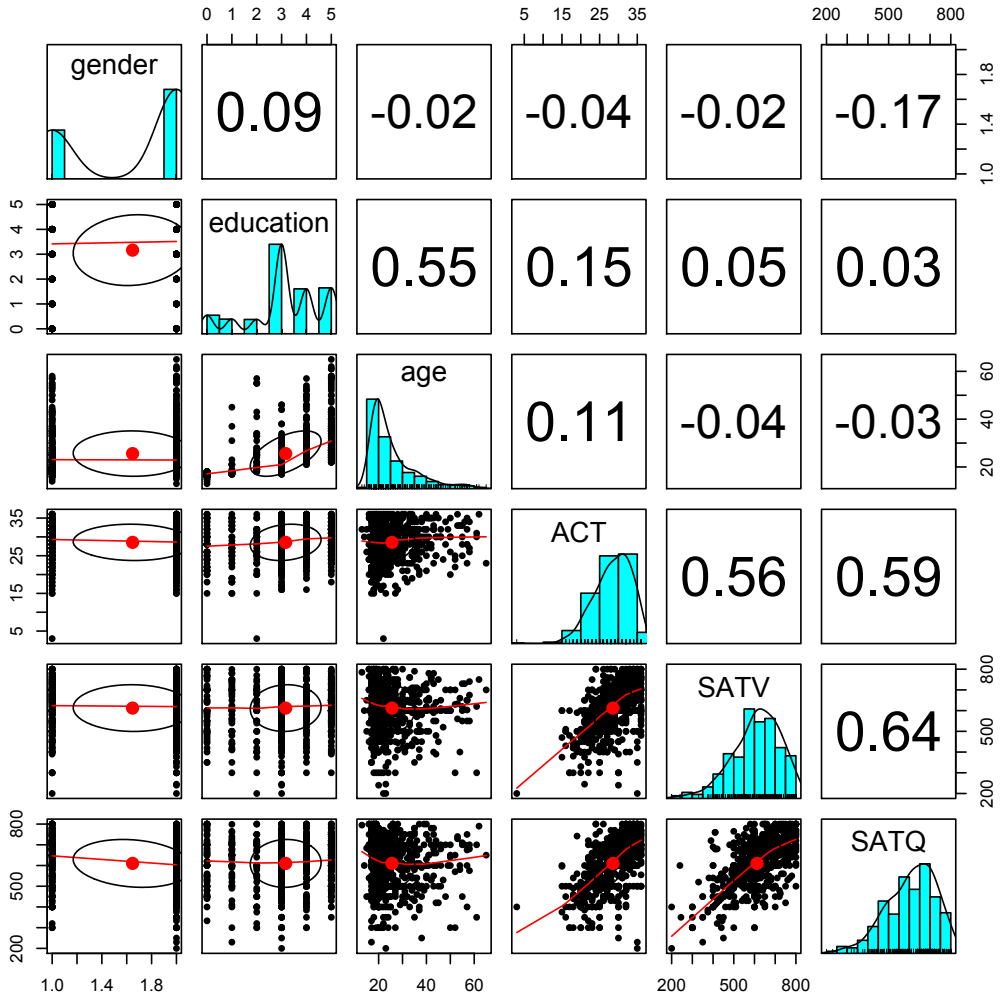
R code

```
my.data <- sat.act  #an example data file that is part of psych
#or
#my.data <-read.file()   #look for it on your hard drive
#or
file.name <-"http://personality-project.org/r/aps/sat.act.txt"
#now read it either locally or remotely
 my.data <- read.file(file.name)
#or if you have copied the data to the clipboard
# my.data <- read.clipboard()  #you can read it from there
describe(my.data) #report basic descriptive statistics
```

	var	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
gender	1	700	1.65	0.48	2	1.68	0.00	1	2	1	-0.61	-1.62	0.02
education	2	700	3.16	1.43	3	3.31	1.48	0	5	5	-0.68	-0.06	0.05
age	3	700	25.59	9.50	22	23.86	5.93	13	65	52	1.64	2.47	0.36
ACT	4	700	28.55	4.82	29	28.84	4.45	3	36	33	-0.66	0.56	0.18
SATV	5	700	612.23	112.90	620	619.45	118.61	200	800	600	-0.64	0.35	4.27
SATQ	6	687	610.22	115.64	620	617.25	118.61	200	800	600	-0.59	0.00	4.41

Graphic display of data using `pairs.panels`

pairs.panels(my.data) #Note the outlier for ACT



Clean up the data using `scrub`. Use ?scrub for help on the parameters.

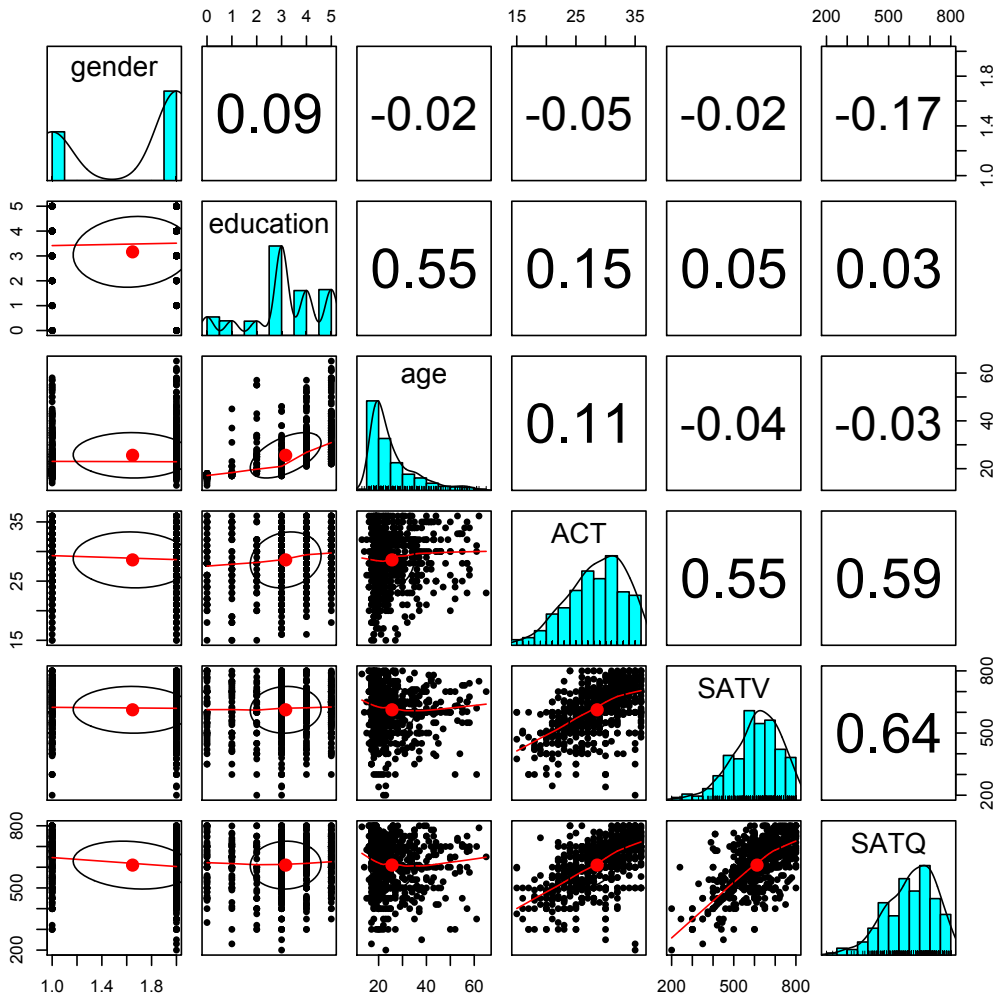
We noticed an outlier in the ACT data in the previous graph (you always graph your data, don't you).
We also noticed that the minimum value for ACT was unlikely (of course, you always describe your data).
So we change any case below 4 on the ACT to be missing (NA).

R code

```
cleaned <- scrub(my.data,"ACT",min=4)  #what data set,
                #which variable, what value to fix
describe(cleaned)   #look at the data again
pairs.panels(cleaned)
```

```
          var   n   mean     sd median trimmed    mad min max range  skew kurtosis   se
gender      1 700   1.65   0.48      2    1.68   0.00   1   2     1 -0.61    -1.62 0.02
education   2 700   3.16   1.43      3    3.31   1.48   0   5     5 -0.68    -0.06 0.05
age         3 700  25.59   9.50     22   23.86   5.93  13  65    52  1.64     2.47 0.36
ACT         4 699  28.58   4.73     29   28.85   4.45  15  36    21 -0.50    -0.36 0.18
SATV        5 700 612.23 112.90    620  619.45 118.61 200 800   600 -0.64     0.35 4.27
SATQ        6 687 610.22 115.64    620  617.25 118.61 200 800   600 -0.59     0.00 4.41
```

Graphic display of cleaned data using `pairs.panels`



Find the pairwise correlations, round to 2 decimals

This also shows how two functions can be nested. We are rounding the output of the cor function.

R code

```
#specify all the parameters being passed
round(cor(x=sat.act,use="pairwise"),digits=2)
#the short way to specify the rounding parameter
round(cor(cleaned,use="pairwise"),2)
```

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

Display it differently using the lowerCor function

Operations that are done a lot may be made into your own functions. Thus, `lowerCor` finds the pairwise correlations, rounds to 2 decimals, displays the lower half of the correlation matrix, and then abbreviates the column labels to make them line up nicely

R code

```
lowerCor(cleaned)
```

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

Testing the significance of one correlation using `cor.test`.

R code

```
cor.test(my.data$ACT,my.data$SATQ)
```

```
        Pearson's product-moment correlation

data:  my.data$ACT and my.data$SATQ
t = 18.9822, df = 685, p-value < 2.2e-16
alternative hypothesis: true correlation
                  is not equal to 0
95 percent confidence interval:
 0.5358435 0.6340672
sample estimates:
      cor
0.5871122
```

1. Specify the variables to correlate
2. Various statistics associated with the correlation.
3. But what if you want to do many tests? Use `corr.test`

Test many correlations for significance using `corr.test`

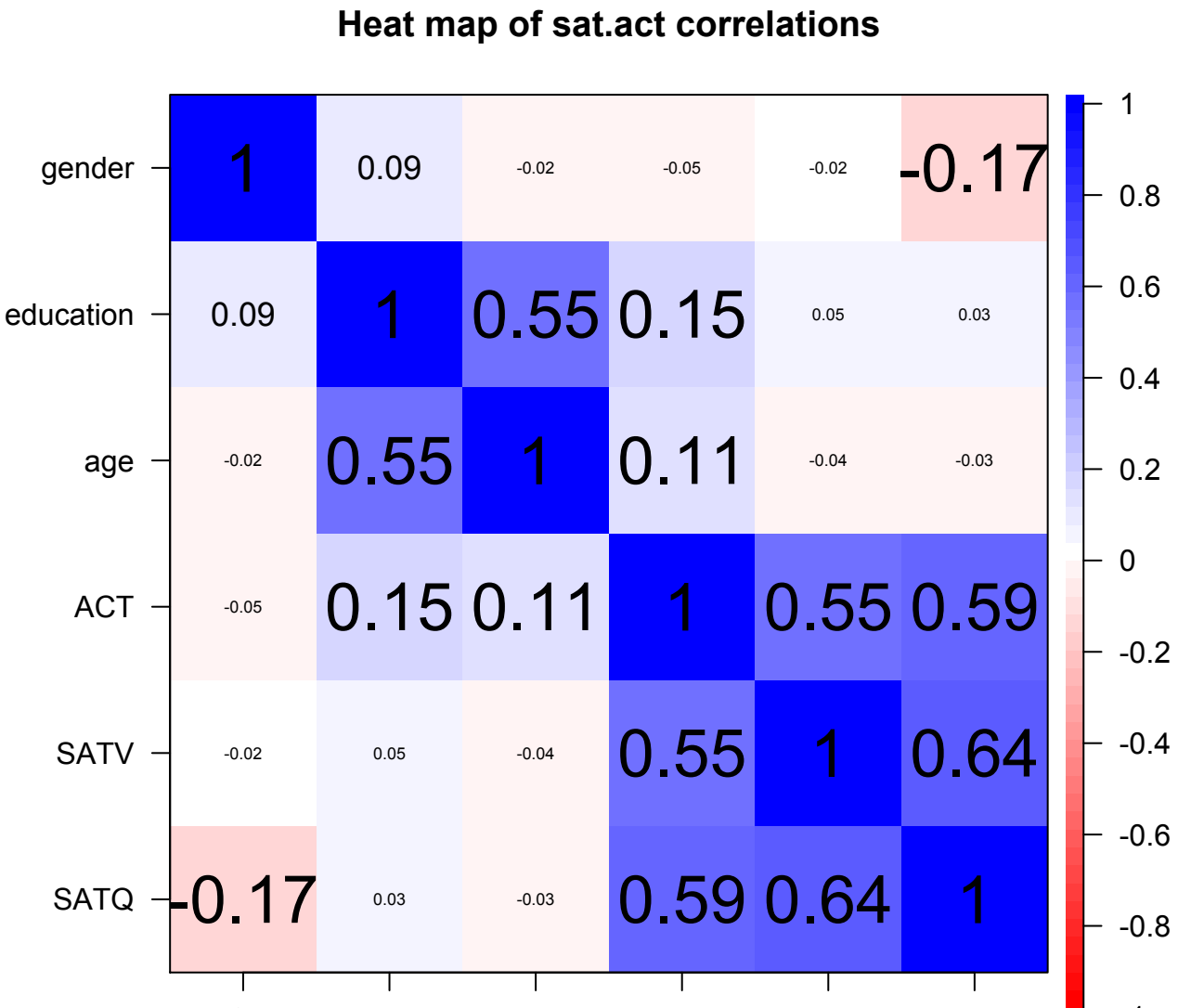
R code

```
corr.test(cleaned)
```

```
all:corr.test(x = cleaned)
Correlation matrix
          gender education   age   ACT  SATV  SATQ
gender      1.00      0.09 -0.02 -0.05 -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        -0.05      0.15  0.11  1.00  0.55  0.59
SATV       -0.02      0.05 -0.04  0.55  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       700 700 699  700  687
...
SATQ          687       687 687 686  687  687
Probability values (Entries above the diagonal are
                          adjusted for multiple tests.)
          gender education  age  ACT SATV SATQ
gender      0.00      0.17 1.00 1.00    1    0
education   0.02      0.00 0.00 0.00    1    1
age         0.58      0.00 0.00 0.03    1    1
ACT         0.21      0.00 0.00 0.00    0    0
SATV        0.62      0.22 0.26 0.00    0    0
```

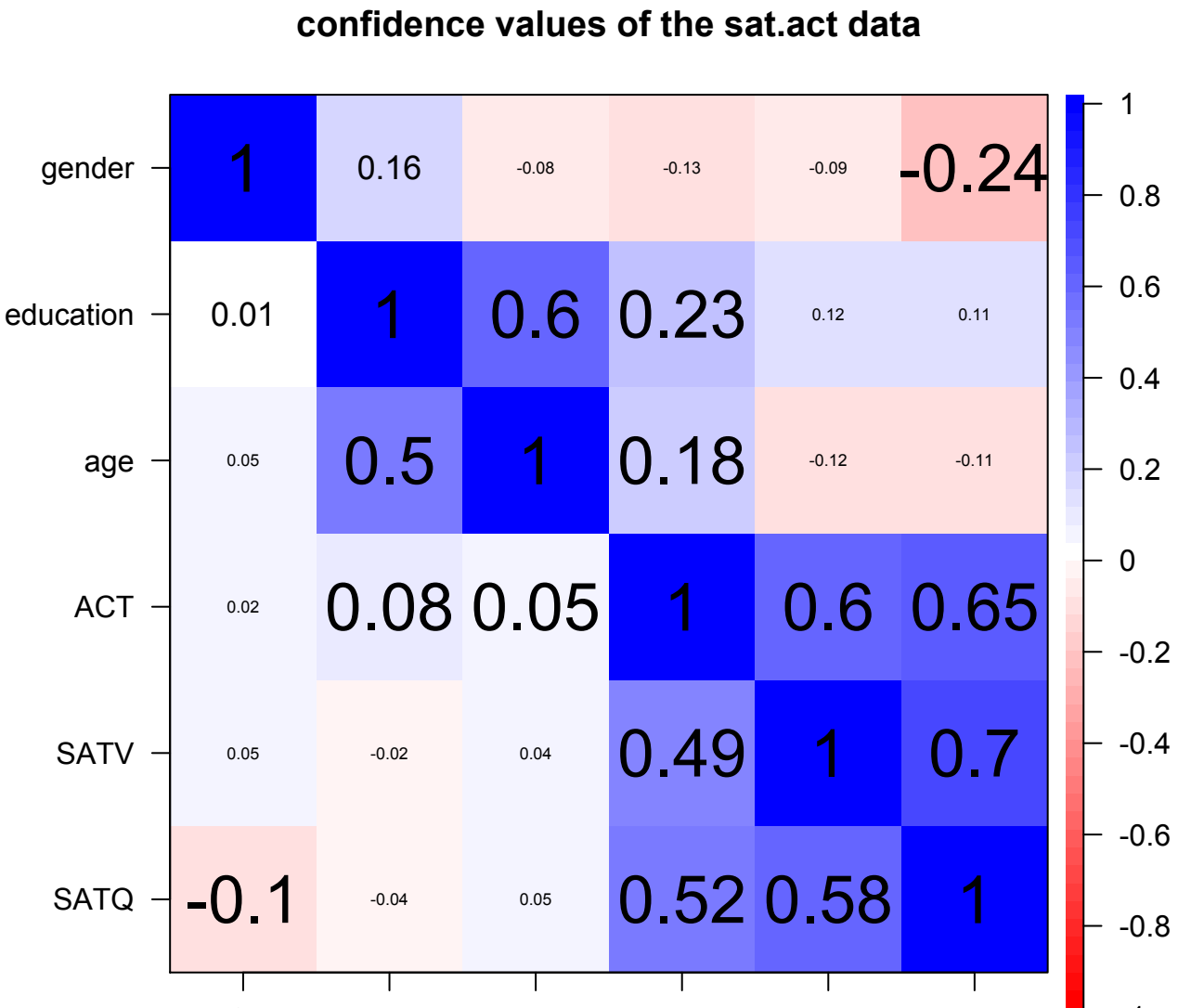
The SAT.ACT correlations. Confidence values from resampling

```
ci <- cor.ci(cleaned,main='Heat map of sat.act')
```



The SAT.ACT bootstrapped confidence intervals of correlation

cor.plot.upperLowerCi(ci,main="Heat map of sat.act")



Are education and gender independent? χ^2 Test of association

```
T <- with(my.data, table(gender, education))
```

```
> T
        education
gender    0    1    2    3
4    5
      1   27   20   23   80   51
46
      2   30   25   21  195   87
95
```

1. First create a table of associations
   - Do this on our data (my.data)
   - Use the "with" command to specify the data set
2. Show the table
3. Apply χ^2 test

```
> chisq.test(T)
        Pearson's Chi-squared test
```

Finding χ^2 from a table of data

- Consider the effect of a treatment on later arrest (From Ashley Kendall, 2016)

Condition	Arrested	Not Arrested
Control	14	21
Treatment	3	23

R code

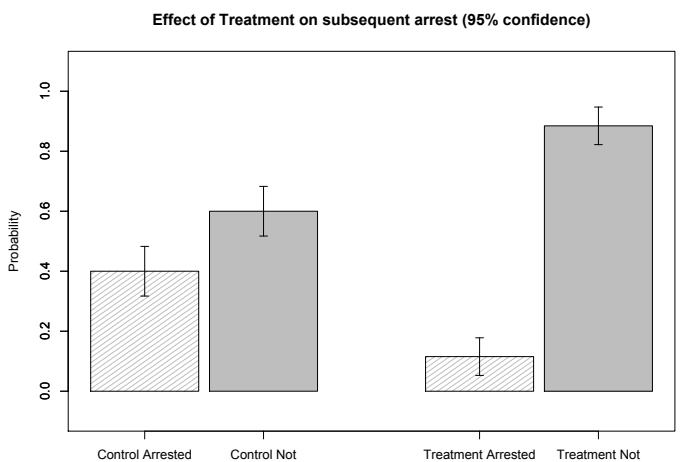
```
ak.df <- data.frame(Control=c(14,21),Treated =c(3,23))
rownames(ak.df) <- c("Arrested","Not Arrested")
ak.df #show the data frame
chisq.test(ak.df)  #Test it using the Yates continuity correction
```

```
> ak.df #show the data frame
              Control Treated
Arrested           14       3
Not Arrested       21      23
> chisq.test(ak.df)  #Test it using the Yates continuity correctio
        Pearson's Chi-squared test with Yates' continuity correcti
data:  ak.df
X-squared = 4.6791, df = 1, p-value = 0.03053
```

Graph the tabled data showing confidence intervals of proportions

R code

```
ak.df <- data.frame(Control=c(14,21),Treated =c(3,23))
ak.p <- t(t(ak.df)/colSums(ak.df)) #convert to probabilities
standard.error  <- sqrt(ak.p[1,] * ak.p[2,]/colSums(ak.df))
stats <- data.frame(mean=as.vector(ak.p),
                          se=rep(standard.error,each=2))
rownames(stats) <- c("Control Arrested","Control Not",
                        "Treatment Arrested","Treatment Not")
error.bars(stats=stats,bars=TRUE,space=c(.1,.1,1,.1),
 density=c(20,-10,20,-10),ylab="Probability",
 xlab="Control vs Treatment",
 main ="Effect of Treatment on subsequent arrest (95% confidence)")
```




```
round(stats,2)
                          mean   se
Control Arrested          0.40 0.08
Control Not               0.60 0.08
Treatment Arrested        0.12 0.06
Treatment Not             0.88 0.06
```

Multiple regression and the general linear model

1. Use the sat.act data example
2. Do the linear model
3. Summarize the results

R code

```
mod1 <- lm(SATV ~ education + gender + SATQ,data=my.data)
summary(mod1,digits=2)
```

```
Call:
lm(formula = SATV ~ education + gender + SATQ, data = my.data)
Residuals:
    Min      1Q  Median      3Q     Max
-372.91  -49.08    2.30   53.68  251.93
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 180.87348   23.41019   7.726 3.96e-14 ***
education     1.24043    2.32361   0.534  0.59363
gender       20.69271    6.99651   2.958  0.00321 **
SATQ          0.64489    0.02891  22.309  < 2e-16 ***
Signif. codes:  0 Ô***Õ 0.001 Ô**Õ 0.01 Ô*Õ 0.05 Ô.Õ 0.1 Ô Õ 1
Residual standard error: 86.24 on 683 degrees of freedom
  (13 observations deleted due to missingness)
Multiple R-squared: 0.4231,     Adjusted R-squared: 0.4205
F-statistic:   167 on 3 and 683 DF,  p-value: < 2.2e-16
```

Zero center the data before examining interactions

In order to examine interactions using multiple regression, we must first "zero center" the data. This may be done using the `scale` function. By default, `scale` will standardize the variables. So to keep the original metric, we make the scaling parameter FALSE.

R code

```
csat <- data.frame(scale(my.data,scale=FALSE))
describe(csat)  #centered not standardized data
```

```
          vars   n mean    sd median trimmed   mad    min    max
gender       1 700    0  0.48   0.35    0.04  0.00  -0.65   0.35
education    2 700    0  1.43  -0.16    0.14  1.48  -3.16   1.84
age          3 700    0  9.50  -3.59   -1.73  5.93 -12.59  39.41
ACT          4 700    0  4.82   0.45    0.30  4.45 -25.55   7.45
SATV         5 700    0 112.90   7.77    7.22 118.61 -412.23 187.77
SATQ         6 687    0 115.64   9.78    7.04 118.61 -410.22 189.78
```

Note that we need to take the output of scale (which comes back as a matrix) and make it into a data.frame if we want to use the linear model on it.

Zero center the data before examining interactions

R code

```
csat <- data.frame(scale(my.data,scale=FALSE))
mod2 <-  lm(SATV  ~ education * gender * SATQ,data=csat)
summary(mod2)
```

```
Call:
all:
lm(formula = SATV ~ education * gender * SATQ, data = csat)

Residuals:
    Min      1Q  Median      3Q     Max
-372.53  -48.76    3.33   51.24  238.50

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)            0.773576   3.304938   0.234  0.81500
education              2.517314   2.337889   1.077  0.28198
gender                18.485906   6.964694   2.654  0.00814 **
SATQ                   0.620527   0.028925  21.453  < 2e-16 ***
education:gender       1.249926   4.759374   0.263  0.79292
education:SATQ        -0.101444   0.020100  -5.047 5.77e-07 ***
gender:SATQ            0.007339   0.060850   0.121  0.90404
education:gender:SATQ  0.035822   0.041192   0.870  0.38481
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 84.69 on 679 degrees of freedom
  (13 observations deleted due to missingness)
Multiple R-squared:  0.4469,        Adjusted R-squared:  0.4412
F-statistic: 78.37 on 7 and 679 DF,  p-value: < 2.2e-16
```

Compare model 1 and model 2 using `anova`

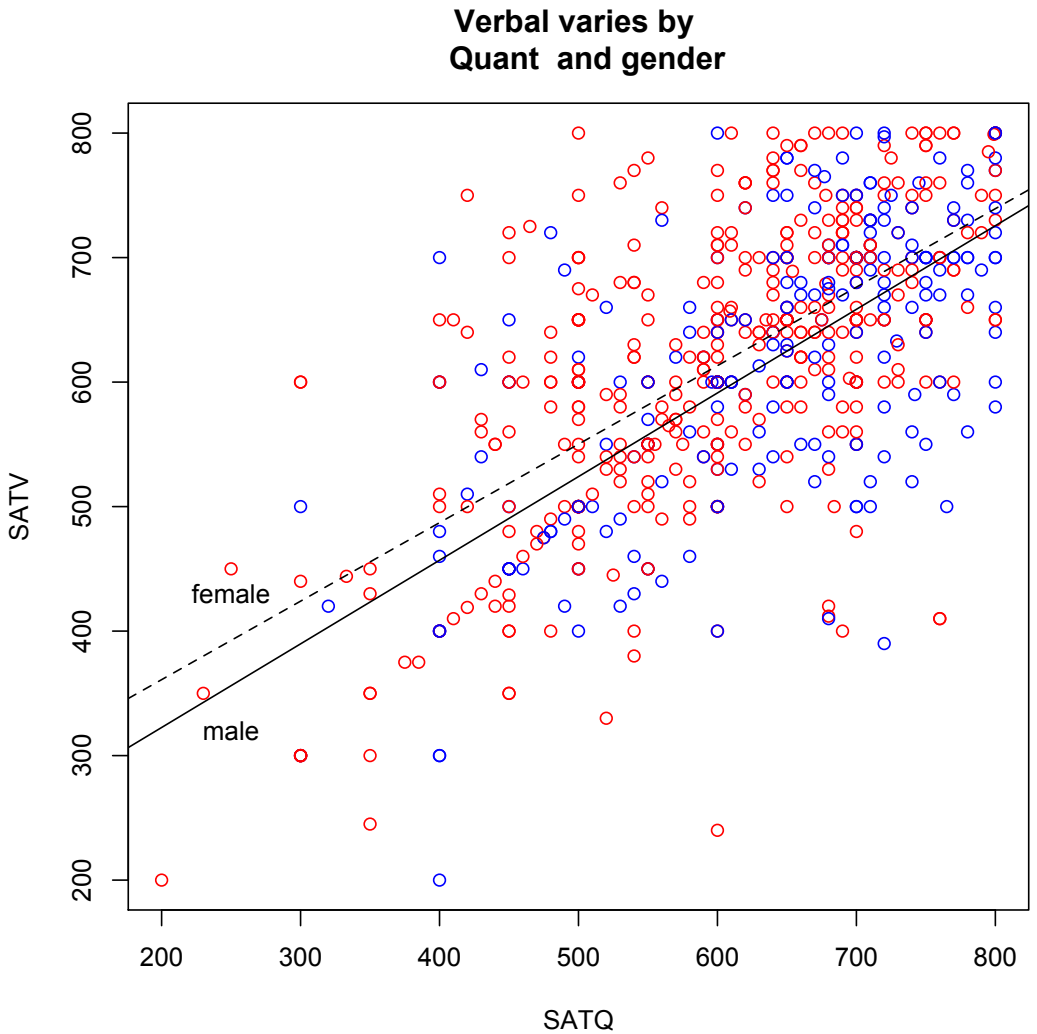
Test the difference between the two linear models

R code

```
anova(mod1,mod2)
```

```
Analysis of Variance Table
Analysis of Variance Table

Model 1: SATV ~ education + gender + SATQ
Model 2: SATV ~ education * gender * SATQ
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)
1    683 5079984
2    679 4870243  4    209742 7.3104 9.115e-06 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1
```

Show the regression lines by gender



First plot all the data.

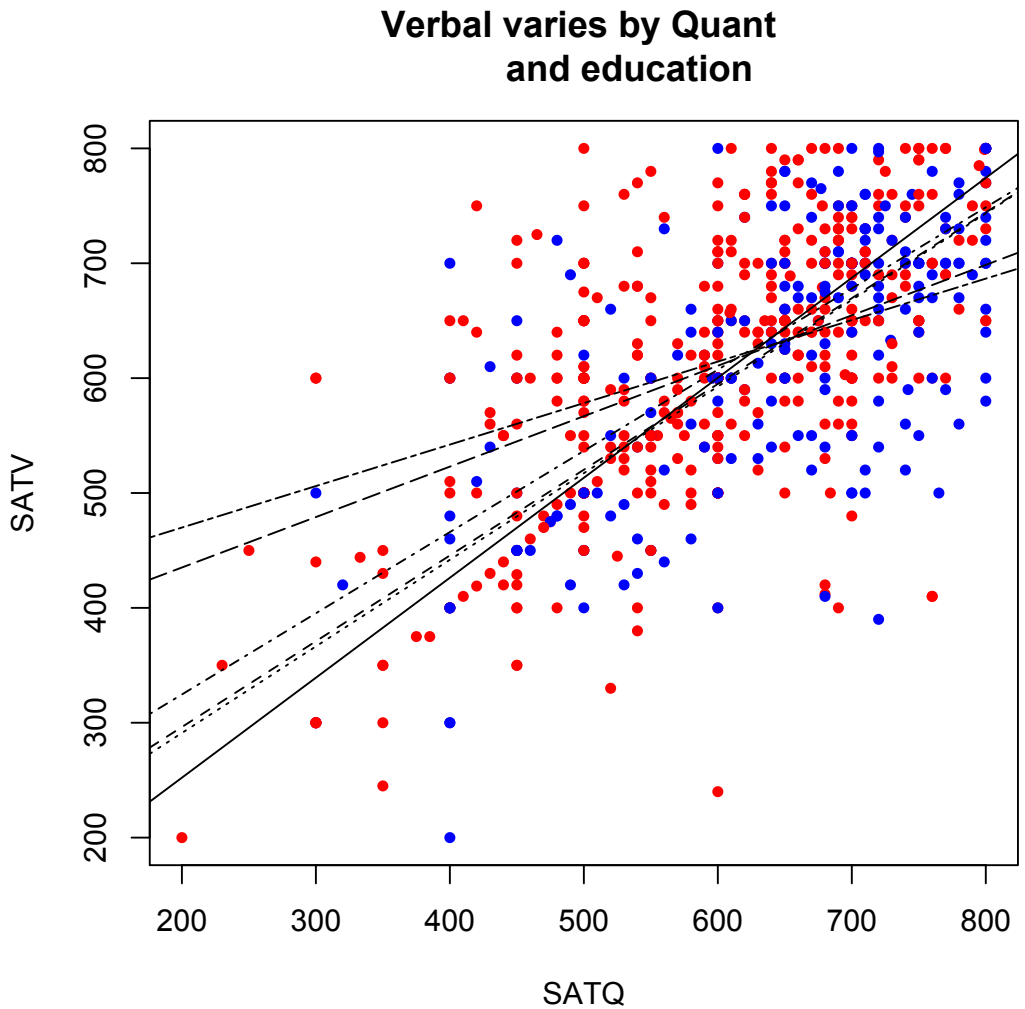
Then add the regression lines.

Then put a title on the whole thing.

R code

```
#first plot the data points
with(my.data,plot(SATV~SATQ,
 col=c("blue","red")[gender]))
#add the regression lines
by(my.data,my.data$gender,
   function(x) abline
  (lm(SATV~SATQ,data=x),
   lty=c("solid","dashed"
         )[x$gender]))
#add a title
title("Verbal varies by
   Quant  and gender")
 #label the lines
text(250,320,"male")
text(250,430,"female")
```

Show the regression lines by education



Do this again, but for levels of education as the moderator.

R code

```
with(my.data,plot(SATV~SATQ,
    col=c("blue","red")[gender],
    pch=20)) #plot character
by(my.data,my.data$education,
   function(x) abline
 (lm(SATV~SATQ,data=x),
 lty=c("solid", "dashed","dotted",
   "dotdash", "longdash",
  "twodash")[(x$education+1)]))

 title("Verbal varies by Quant
         and education")
```

Questions?

Using R for psychological statistics: Basic statistics

1. Writing syntax
   - For a single line, just type it
   - Mistakes can be redone by using the up arrow key
   - For longer code, use a text editor (built into some GUIs)
2. Data entry
   - Using built in data sets for examples
   - Copying from another program
   - Reading a text or csv file
   - Importing from SPSS or SAS
   - Simulate it (using various simulation routines)
3. Descriptives
   - Graphical displays
   - Descriptive statistics
   - Correlation
4. Inferential
   - the t test
   - the F test
   - the linear model

Data entry overview

1. Using built in data sets for examples
   - `data()` will list $>$ 100 data sets in the `datasets` package as well as all sets in loaded packages.
   - Most packages have associated data sets used as examples
   - *psych* has $>$ 50 example data sets
2. Copying from another program
   - use copy and paste into R using `read.clipboard` and its variations
3. Reading a text or csv file
   - read a local or remote file
4. Importing from SPSS or SAS
   - Use either the *foreign*, *haven* or *rio* packages
5. Simulate it (using various simulation routines)
6. Model it using simulations (e.g., `cta` (?))

Examples of built in data sets from the psych package

```
> data(package="psych")
```

ability	16 multiple choice IQ items from the ICAR project (?)
Bechtoldt	Seven data sets showing a bifactor solution (???).
Dwyer	8 cognitive variables used by ? for an example.
Reise	Seven data sets showing a bifactor solution (?).
affect	Data sets of affect and arousal scores as a function of personality and movie conditions (?)
income	US family income from US census 2008
bfi	25 Personality items representing 5 factors (N=2800)
blot	Bond's Logical Operations Test - BLOT (N=150) (?)
burt	11 emotional variables from ?
cities	Distances between 11 US cities
epi.bfi	13 scales from the Eysenck Personality Inventory and Big 5 inventory
income	US family income from US census 2008
msq	75 mood items from the Motivational State Questionnaire for N=3896
neo	NEO correlation matrix from the NEOPI-R manual (?)
sat.act	3 Measures of ability: SATV, SATQ, ACT (N=700)
Thurstone	Seven data sets showing a bifactor solution.
veg (vegetables)	Paired comparison of preferences for 9 vegetables (?)

Reading data from another program –using the clipboard

1. Read the data in your favorite spreadsheet or text editor
2. Copy to the clipboard
3. Execute the appropriate `read.clipboard` function with or without various options specified

```
my.data <- read.clipboard() #assumes headers and tab or space delimited
my.data <- read.clipboard.csv() #assumes headers and comma delimited
my.data <- read.clipboard.tab() #assumes headers and tab delimited
                                                    (e.g., from Excel)
my.data <- read.clipboard.lower() #read in a matrix given the lower
my.data <- read.clipboard.upper() # or upper off diagonal
my.data <- read.clipboard.fwf() #read in data using a fixed format width
                                                    (see read.fwf for instr
```

4. `read.clipboard()` has default values for the most common cases and these do not need to be specified. Consult ?read.clipboard for details. In particular, are headers provided for each column of input?

Reading from a local or remote file

1. Perhaps the standard way of reading in data is using the `read` command.
   - First must specify the location of the file
   - Can either type this in directly or use the `file.choose` function. This goes to your normal system file handler.
   - The file name/location can be a remote URL.
2. Two examples of reading data

R code

```
file.name <- file.choose() #this opens a window to allow you find the file
#or
file.name="http://personality-project.org/r/datasets/R.appendix1.data"
my.data <- read.file(file.name)
#or
my.data = read.table(file.name,header=TRUE)   #the conventional way
dim(my.data) #find the dimensionality of our data
describe(my.data) #describe it to check the means, ranges, etc.
```

```
> dim(my.data )  #what are the dimensions of what we read?
[1] 18  2
> describe(my.data ) #do the data look right?
          var  n  mean   sd median trimmed  mad min max range skew kurtosis   se
Dosage*     1 18  1.89 0.76      2    1.88 1.48   1   3     2 0.16    -1.12 0.18
Alertness   2 18 27.67 6.82     27   27.50 8.15  17  41    24 0.25    -0.68 1.61
```

Put it all together: read, show, describe

R code

```
datafilename="http://personality-project.org/r/datasets/R.appendix1.data"
data.ex1<- read.file(datafilename)
dim(data.ex1)  #what are the dimensions of what we read?
data.ex1  #show the data
headTail(data.ex1) #just the top and bottom lines
describe(data.ex1) #descriptive stats
```

```
   Dosage Alertness
1       a        30
2       a        38
...   (rows deleted by hand)
17      c        20
18      c        19

> headTail(data.ex1) #just the top and bottom lines
    Dosage Alertness
1        a        30
2        a        38   'head' rows
3        a        35
4        a        41
...   <NA>       ... (rows automatically deleted)
15       c        17
16       c        21
17       c        20  'tail' rows
18       c        19
> describe(data.ex1) #descriptive stats
          vars  n  mean   sd median trimmed  mad min max range skew kurtosis   se
Dosage*      1 18  1.89 0.76      2    1.88 1.48   1   3     2 0.16    -1.35 0.18
Alertness    2 18 27.67 6.82     27   27.50 8.15  17  41    24 0.25    -1.06 1.61
```

1. Read the data from a remote file
2. Show all the cases (problematic if there are are 100s – 1000s)
3. Just show the first and last (4) lines
4. Find descriptive statistics

However, some might want to Import SAS or SPSS files

The first thing to try is the `read.file` function. For more complicated data sets, there are several different packages that make importing SPSS, SAS, Systat, etc. files possible to do.

- read.file Function in psych to read .txt, .csv, .sav, ,xpt, .r, ,.rda, .text (etc.)
- foreign Read data stored by Minitab, S, SAS, SPSS, Stata, Systat, Weka, dBase. Comes installed with R. Somewhat complicated syntax.
- haven Reads/writes SPSS and Stata files. Handles SPSS labels nicely (keeps the item labels, but converts the data to factors).
- rio A general purpose package that requires installation of many of the other packages used for data import. Easiest to use, but overkill if just reading in one type of file. Basically a front end to many import/export packages. It determines which package to use based

Read a "foreign" file e.g., an SPSS sav file, using foreign package

`read.spss` Reads a file stored by the SPSS save or export commands. (The defaults lead to problems, make sure to specify that you want use.value.labels = FALSE, to.data.frame = TRUE)

```
read.spss(file, use.value.labels = FALSE, to.data.frame = TRUE,
          max.value.labels = Inf, trim.factor.names = FALSE,
          trim_values = TRUE, reencode = NA, use.missings = to.data.frame)
```

file	Character string: the name of the file or URL to read.
use.value.labels	Convert variables with value labels into R factors with those levels? Should be FALSE
to.data.frame	return a data frame? Defaults to FALSE, probably should be TRUE in most cases.
max.value.labels	Only variables with value labels and at most this many unique values will be converted to factors if use.value.labels = *TRUE*.
trim.factor.names	Logical: trim trailing spaces from factor levels?
trim_values	logical: should values and value labels have trailing spaces ignored when matching for use.value.labels = *TRUE*?
use.missings	logical: should information on user-defined missing values be used to set the corresponding values to NA?

An example of reading from an SPSS file using foreign

```
> library(foreign)

> datafilename <- "http://personality-project.org/r/datasets/finkel.sav"

>   eli <- read.spss(datafilename,to.data.frame=TRUE,
                     use.value.labels=FALSE)
> headTail(eli,2,2)
> describe(eli,skew=FALSE)
```

```
      USER HAPPY SOULMATE ENJOYDEX UPSET
1    "001"     4        7        7     1
2    "003"     6        5        7     0
...   <NA>   ...      ...      ...   ...
68   "076"     7        7        7     0
69   "078"     2        7        7     1
>
          var  n mean    sd median trimmed
mad min max range   se
USER*       1 69 35.00 20.06     35   35.00 25.20   1
69     68 2.42
HAPPY       2 69  5.71  1.04      6    5.82  0.00   2
7      5 0.13
SOULMATE    3 69  5.09  1.80      5    5.32  1.48   1
7      6 0.22
ENJOYDEX    4 68  6.47  1.01      7    6.70  0.00   2
```

1. Make the *foreign* package active
2. Specify the name (and location) of the file to read
3. Read from a SPSS file
4. Show the top and bottom 2 cases
5. Describe it to make sure it is right

An example of reading from an SPSS file using rio

```
> library(rio)

> datafilename <- "http://personality-project.org/r/datasets/finkel.sav"

>   eli <- import(datafilename)     #note that it figures out what to
do
> headTail(eli,2,2) #The first and last 2
> describe(eli,skew=FALSE)
```

```
      USER HAPPY SOULMATE ENJOYDEX UPSET
1    "001"     4        7        7     1
2    "003"     6        5        7     0
...   <NA>   ...      ...      ...   ...
68   "076"     7        7        7     0
69   "078"     2        7        7     1
>
          var  n  mean    sd median trimmed
mad min max range   se
USER*       1 69 35.00 20.06     35   35.00 25.20   1
69     68 2.42
HAPPY       2 69  5.71  1.04      6    5.82  0.00   2
7      5 0.13
SOULMATE    3 69  5.09  1.80      5    5.32  1.48   1
7      6 0.22
ENJOYDEX    4 68  6.47  1.01      7    6.70  0.00   2
```

1. Make the *rio* package active
2. Specify the name (and location) of the file to read
3. Import from a SPSS file
4. Show the top and bottom 2 cases
5. Describe it to make sure it is right

An example of reading from an SPSS file using haven

```
> library(haven)

> datafilename <- "http://personality-project.org/r/datasets/finkel.sav"

>   eli <- read_spss(datafilename)     #note that it figures out what
to do
> headTail(eli,3,2) The first 3 and last 2
> describe(eli,skew=FALSE)
```

```
     USER HAPPY SOULMATE ENJOYDEX UPSET
1    "001"     4        7        7      1
2    "003"     6        5        7      0
3    "004"     6        7        7      0
...    <NA>   ...      ...      ...    ...
68   "076"     7        7        7      0
69   "078"     2        7        7      1>
          var  n mean   sd median trimmed
mad min max range  se
USER*       1 69 35.00 20.06     35   35.00 25.20   1
69     68 2.42
HAPPY       2 69  5.71  1.04      6    5.82  0.00   2
7       5 0.13
SOULMATE    3 69  5.09  1.80      5    5.32  1.48   1
7       6 0.22
ENJOYDEX    4 68  6.47  1.01      7    6.70  0.00   2
```

1. Make the *haven* package active
2. Specify the name (and location) of the file to read
3. Import from a SPSS file
4. Show the top 3 and bottom 2 cases
5. Describe it to make sure it is right

`read.file` as a convenient solution to reading files

1. Combines `file.choose` and `read.table`
2. Also, based upon the suffix of the data, will choose the most likely way to read a SPSS, csv, text, rds or SAS export file.
3. Not as powerful as *foreign* or *rio* but easier.
4. Automatically reads SPSS .sav files as numeric values but can read them with the item information as well.

R code

```
eli <- read.file().  #goes off and searches for a local file
datafilename <- "http://personality-project.org/r/datasets/finkel.sav"
eli <- read.file(datafilename). #uses that remote address to get it
ashley <- read.file() #a file from Ashley Kendall on my computer
kendall <- read.file(read.file(use.value.labels=TRUE) #keep the labels
ashley[1:3,8:17]
kendall[1:3,8:17]
```

```
ashley[1:3,8:17]
  HighNA LowPA LowNA Active Alert Nervs Frust Worried Irrit Stress
1      8     3     0      1     0     1     2       0     3      2
2      6     1     0      0     2     0     3       0     2      1
3      1     7     0      3     3     1     0       0     0      0
> kendall[1:3,8:17]
  HighNA LowPA LowNA     Active Alert      Nervs      Frust Worried      Irrit     Stress
1      8     3     0   a little     0   a little   somewhat       0  very much   somewhat
2      6     1     0 not at all     2 not at all  very much       0   somewhat   a little
3      1     7     0  very much     3   a little not at all       0 not at all not at all
```

Simulate data (Remember to always call them simulated!)

For many demonstration purposes, it is convenient to generate simulated data with a certain defined structure. The *psych* package has a number of built in simulation functions. Here are a few of them.

1. Simulate various item structures
   - sim.congeneric A one factor congeneric measure model
   - sim.items A two factor structure with either simple structure or a circumplex structure.
   - sim.rasch Generate items for a one parameter IRT model.
   - sim.irt Generate items for a one-four parameter IRT Model
2. Simulate various factor structures
   - sim.simplex Default is a four factor structure with a three time point simplex structure.
   - sim.hierarchical Default is 9 variables with three correlated factors.

Get the data and look at them

Read in some data, look at the first and last few cases (using `headTail`), and then get basic descriptive statistics. For this example, we will use a built in data set.

R code

```
headTail(epi.bfi)
```

```
    epiE epiS epiImp epilie epiNeur bfagree bfcon bfext bfneur bfopen bdi traitanx stateanx
1     18   10      7      3       9     138    96   141     51   138   1       24       22
2     16    8      5      1      12     101    99   107    116   132   7       41       40
3      6    1      3      2       5     143   118    38     68    90   4       37       44
4     12    6      4      3      15     104   106    64    114   101   8       54       40
...  ...  ...    ...    ...     ...     ...   ...   ...    ...   ... ...      ...      ...
228   12    7      4      3      15     155   129   127     88   110   9       35       34
229   19   10      7      2      11     162   152   163    104   164   1       29       47
230    4    1      1      2      10      95   111    75    123   138   5       39       58
231    8    6      3      2      15      85    62    90    131    96  24       58       58
```

epi.bfi has 231 cases from two personality measures.

Now find the descriptive statistics for this data set

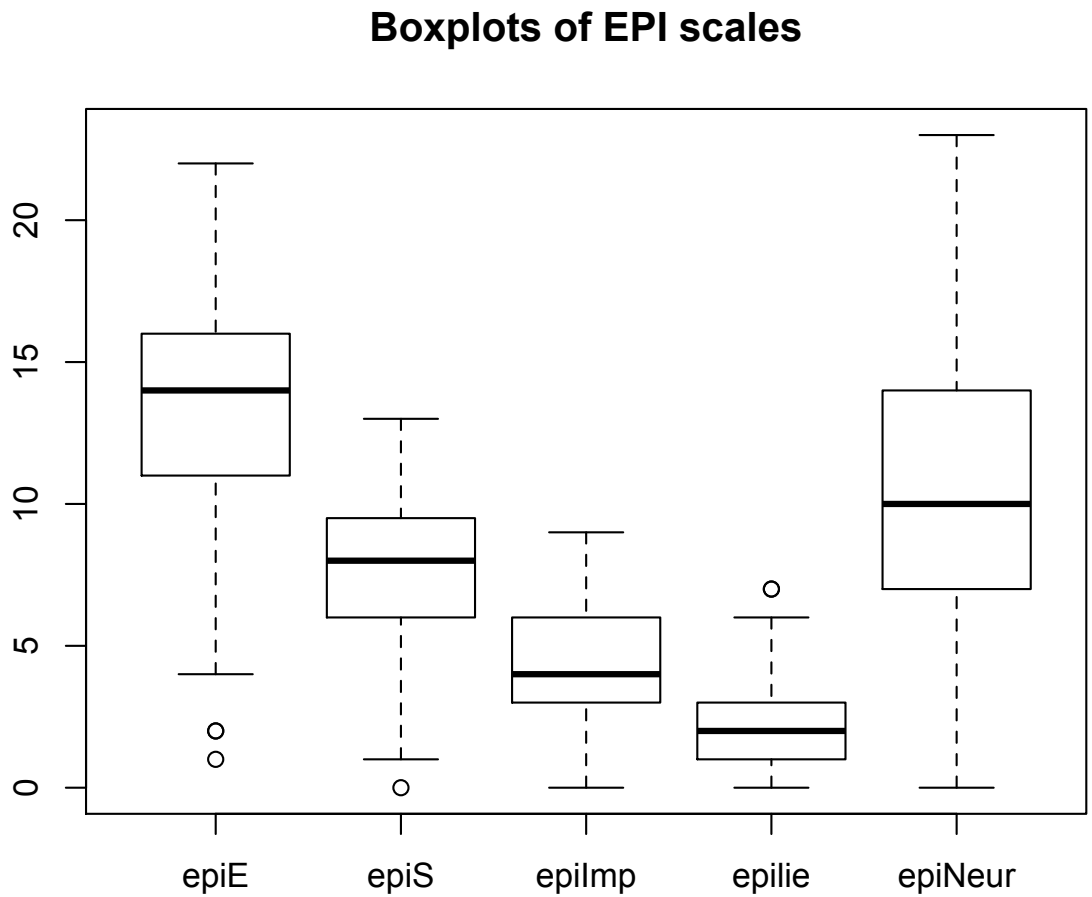
R code

```
describe(epi.bfi)
```

	var	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
epiE	1	231	13.33	4.14	14	13.49	4.45	1	22	21	-0.33	-0.01	0.27
epiS	2	231	7.58	2.69	8	7.77	2.97	0	13	13	-0.57	0.04	0.18
epiImp	3	231	4.37	1.88	4	4.36	1.48	0	9	9	0.06	-0.59	0.12
epilie	4	231	2.38	1.50	2	2.27	1.48	0	7	7	0.66	0.30	0.10
epiNeur	5	231	10.41	4.90	10	10.39	4.45	0	23	23	0.06	-0.46	0.32
bfagree	6	231	125.00	18.14	126	125.26	17.79	74	167	93	-0.21	-0.22	1.19
bfcon	7	231	113.25	21.88	114	113.42	22.24	53	178	125	-0.02	0.29	1.44
bfext	8	231	102.18	26.45	104	102.99	22.24	8	168	160	-0.41	0.58	1.74
bfneur	9	231	87.97	23.34	90	87.70	23.72	34	152	118	0.07	-0.51	1.54
bfopen	10	231	123.43	20.51	125	123.78	20.76	73	173	100	-0.16	-0.11	1.35
bdi	11	231	6.78	5.78	6	5.97	4.45	0	27	27	1.29	1.60	0.38
traitanx	12	231	39.01	9.52	38	38.36	8.90	22	71	49	0.67	0.54	0.63
stateanx	13	231	39.85	11.48	38	38.92	10.38	21	79	58	0.72	0.04	0.76

Boxplots are a convenient descriptive device

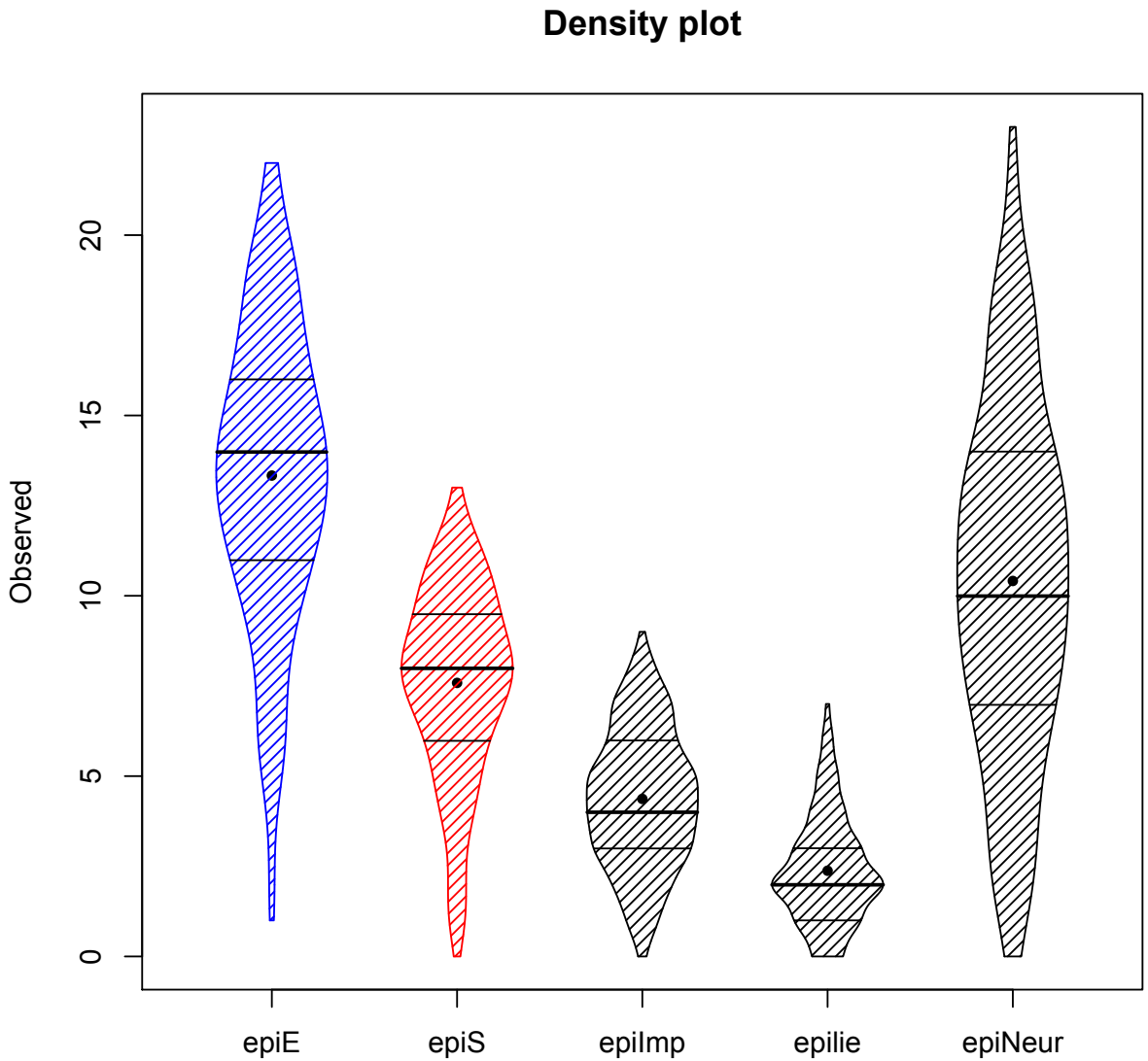
Show the Tukey "boxplot" for the Eysenck Personality Inventory



Use the `box plot` function and select the first five variables.

```
my.data <- epi.bfi
boxplot(my.data[1:5])
```

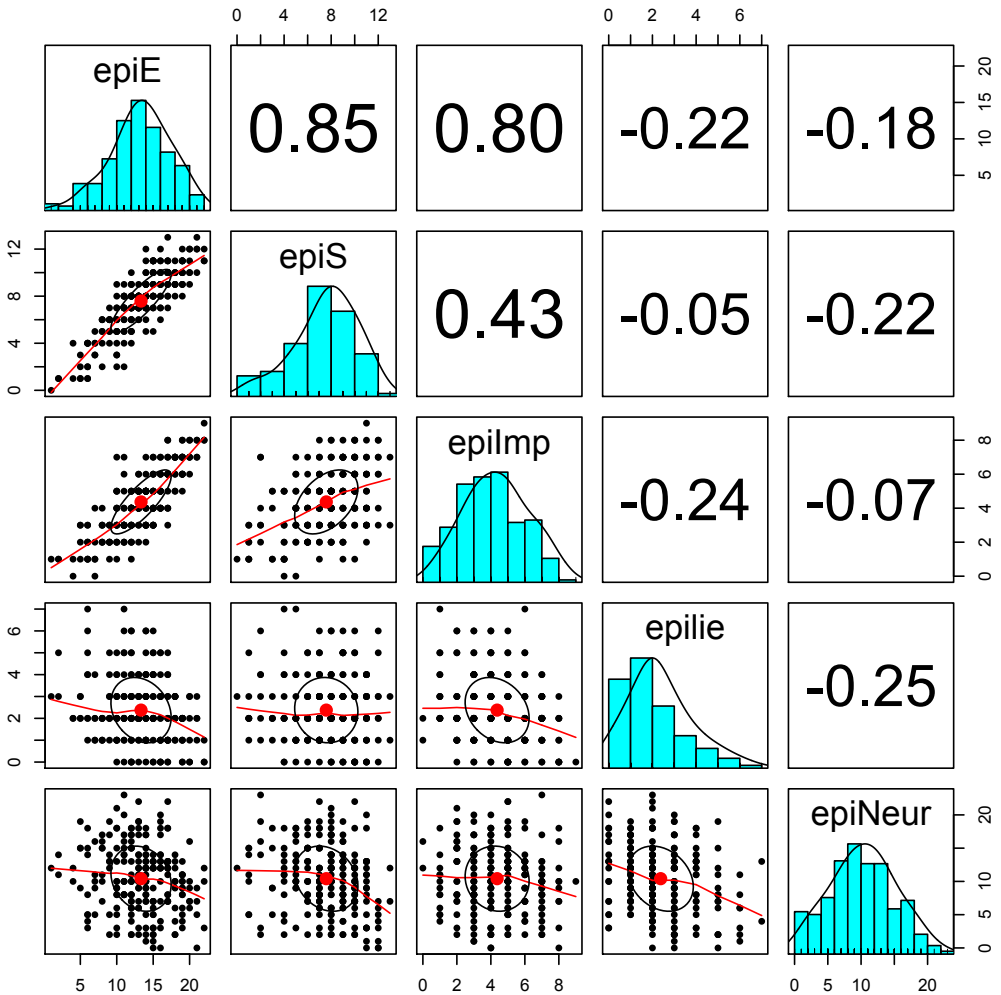
An alternative display is a 'violin' plot (available as `violinBy`)



Use the `violinBy` function from *psych*

```
violinBy(my.data[1:5])
```

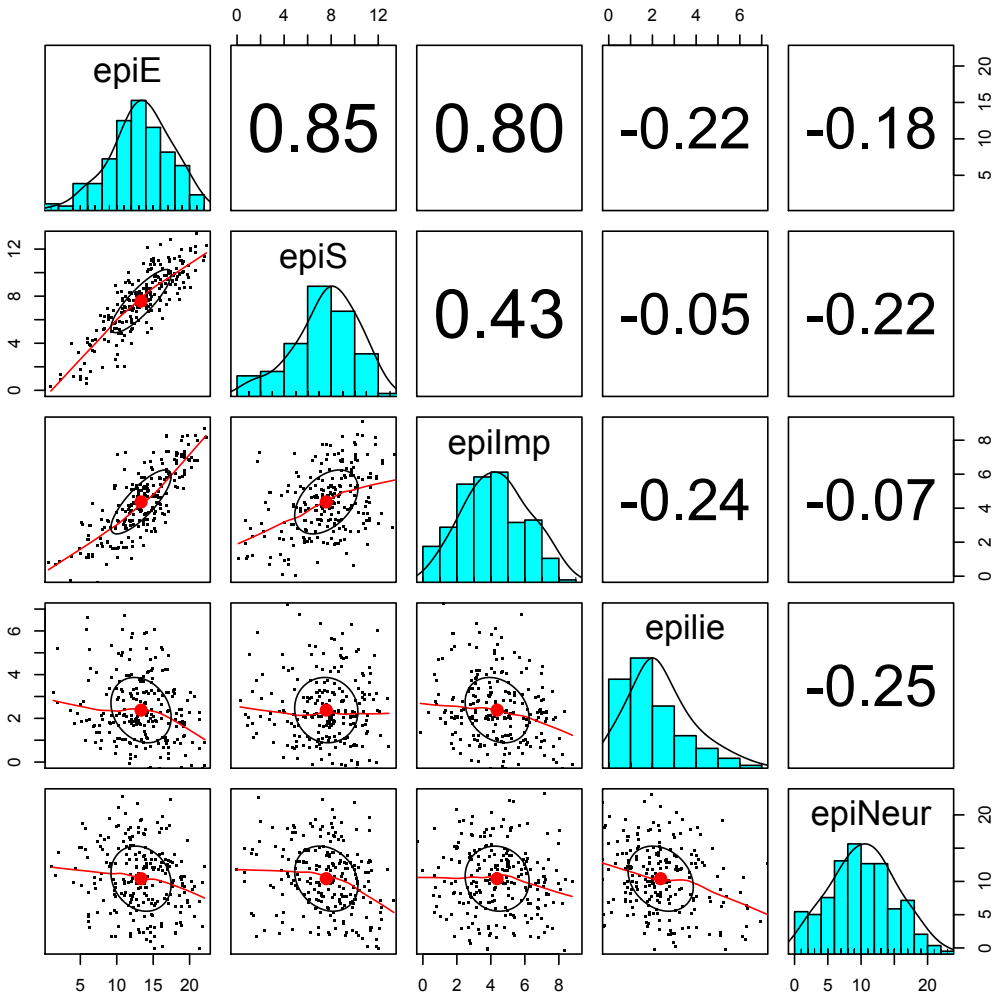
Plot the scatter plot matrix (SPLOM) of the first 5 variables using the `pairs.panels` function. Note that the plotting points overlap because of the polytomous nature of the data.



Use the `pairs.panels` function from *psych*

```
pairs.panels(my.data[1:5])
```

Plot the scatter plot matrix (SPLOM) of the first 5 variables using the `pairs.panels` function but with smaller plot charactet (pch) and jittering the points in order to better show the distributions.



Use the `pairs.panels` function from *psych*

```
pairs.panels(my.data[1:5],pch='.',
       jiggle=TRUE)
```

Find the correlations for this data set, round off to 2 decimal places.

Because we have some missing data, we use "pairwise complete" correlations. For the purists amongst us, it is irritating that the columns are not equally spaced.

R code

```
round(cor(my.data, use = "pairwise"), 2)
```

```
          epiE  epiS epiImp epilie epiNeur bfagree bfcon bfext bfneur bfopen   bdi traitanx st
epiE      1.00  0.85   0.80  -0.22   -0.18    0.18 -0.11  0.54  -0.09   0.14 -0.16    -0.23
epiS      0.85  1.00   0.43  -0.05   -0.22    0.20  0.05  0.58  -0.07   0.15 -0.13    -0.26
epiImp    0.80  0.43   1.00  -0.24   -0.07    0.08 -0.24  0.35  -0.09   0.07 -0.11    -0.12
epilie   -0.22 -0.05  -0.24   1.00   -0.25    0.17  0.23 -0.04  -0.22  -0.03 -0.20    -0.23
epiNeur  -0.18 -0.22  -0.07  -0.25    1.00   -0.08 -0.13 -0.17   0.63   0.09  0.58     0.73
bfagree   0.18  0.20   0.08   0.17   -0.08    1.00  0.45  0.48  -0.04   0.39 -0.14    -0.31
bfcon    -0.11  0.05  -0.24   0.23   -0.13    0.45  1.00  0.27   0.04   0.31 -0.18    -0.29
bfext     0.54  0.58   0.35  -0.04   -0.17    0.48  0.27  1.00   0.04   0.46 -0.14    -0.39
bfneur   -0.09 -0.07  -0.09  -0.22    0.63   -0.04  0.04  0.04   1.00   0.29  0.47     0.59
bfopen    0.14  0.15   0.07  -0.03    0.09    0.39  0.31  0.46   0.29   1.00 -0.08    -0.11
bdi      -0.16 -0.13  -0.11  -0.20    0.58   -0.14 -0.18 -0.14   0.47  -0.08  1.00     0.65
traitanx -0.23 -0.26  -0.12  -0.23    0.73   -0.31 -0.29 -0.39   0.59  -0.11  0.65     1.00
stateanx -0.13 -0.12  -0.09  -0.15    0.49   -0.19 -0.14 -0.15   0.49  -0.04  0.61     0.57
```

Find the correlations for this data set, round off to 2 decimal places using `lowerCor`

This is just a wrapper for round(cor(x,use='pairwise'),2) that has been prettied up with `lowerMat`.

R code

```
lowerCor(my.data)
```

```
         epiE  epiS  epImp epili epiNr bfagr bfcon bfext bfner bfopn bdi   trtnx sttnx
epiE      1.00
epiS      0.85  1.00
epiImp    0.80  0.43  1.00
epilie   -0.22 -0.05 -0.24  1.00
epiNeur  -0.18 -0.22 -0.07 -0.25  1.00
bfagree   0.18  0.20  0.08  0.17 -0.08  1.00
bfcon    -0.11  0.05 -0.24  0.23 -0.13  0.45  1.00
bfext     0.54  0.58  0.35 -0.04 -0.17  0.48  0.27  1.00
bfneur   -0.09 -0.07 -0.09 -0.22  0.63 -0.04  0.04  0.04  1.00
bfopen    0.14  0.15  0.07 -0.03  0.09  0.39  0.31  0.46  0.29  1.00
bdi      -0.16 -0.13 -0.11 -0.20  0.58 -0.14 -0.18 -0.14  0.47 -0.08  1.00
traitanx -0.23 -0.26 -0.12 -0.23  0.73 -0.31 -0.29 -0.39  0.59 -0.11  0.65  1.00
stateanx -0.13 -0.12 -0.09 -0.15  0.49 -0.19 -0.14 -0.15  0.49 -0.04  0.61  0.57  1.00
```

Test the significance and use Holm correction for multiple tests

R code

```
corr.test(my.data)
```

```
Call:corr.test(x = my.data)
Correlation matrix
          epiE  epiS epiImp epilie epiNeur bfagree bfcon bfext bfneur bfopen   bdi traitanx st
epiE      1.00  0.85   0.80  -0.22   -0.18    0.18 -0.11  0.54  -0.09   0.14 -0.16    -0.23
epiS      0.85  1.00   0.43  -0.05   -0.22    0.20  0.05  0.58  -0.07   0.15 -0.13    -0.26
epiImp    0.80  0.43   1.00  -0.24   -0.07    0.08 -0.24  0.35  -0.09   0.07 -0.11    -0.12
..
stateanx -0.13 -0.12  -0.09  -0.15    0.49   -0.19 -0.14 -0.15   0.49  -0.04  0.61     0.57
Sample Size
         epiE epiS epiImp epilie epiNeur bfagree bfcon bfext bfneur bfopen bdi traitanx statea
epiE      231  231    231    231     231     231   231   231    231    231 231      231      2
..
stateanx  231  231    231    231     231     231   231   231    231    231 231      231      2
Probability values (Entries above the diagonal are adjusted for multiple tests.)
         epiE epiS epiImp epilie epiNeur bfagree bfcon bfext bfneur bfopen  bdi traitanx stat
epiE     0.00 0.00   0.00   0.03    0.27    0.27  1.00  0.00   1.00   1.00 0.59     0.02     1
epiS     0.00 0.00   0.00   1.00    0.04    0.08  1.00  0.00   1.00   0.62 1.00     0.00     1
epiImp   0.00 0.00   0.00   0.01    1.00    1.00  0.01  0.00   1.00   1.00 1.00     1.00     1
epilie   0.00 0.43   0.00   0.00    0.01    0.32  0.03  1.00   0.03   1.00 0.08     0.02     0
epiNeur  0.01 0.00   0.26   0.00    0.00    1.00  1.00  0.33   0.00   1.00 0.00     0.00     0
bfagree  0.01 0.00   0.23   0.01    0.21    0.00  0.00  0.00   1.00   0.00 0.95     0.00     0
bfcon    0.08 0.48   0.00   0.00    0.04    0.00  0.00  0.00   1.00   0.00 0.25     0.00     1
bfext    0.00 0.00   0.00   0.50    0.01    0.00  0.00  0.00   1.00   0.00 0.99     0.00     0
bfneur   0.15 0.30   0.18   0.00    0.00    0.50  0.50  0.57   0.00   0.00 0.00     0.00     0
bfopen   0.04 0.02   0.30   0.70    0.19    0.00  0.00  0.00   0.00   0.00 1.00     1.00     1
bdi      0.02 0.04   0.11   0.00    0.00    0.03  0.01  0.03   0.00   0.25 0.00     0.00     0
traitanx 0.00 0.00   0.07   0.00    0.00    0.00  0.00  0.00   0.00   0.11 0.00     0.00     0
stateanx 0.05 0.07   0.18   0.02    0.00    0.00  0.04  0.02   0.00   0.52 0.00     0.00     0
>
```

t.test demonstration with Student's data using `cushny` dataset

William Gossett, publishing under the name Student reported a small sample approximation (t) to the large sample z test. His first example was a data set on the different effect of optical isomers of hyoscyamine hydrobromide reported by **?**. The sleep of 10 patients was measured without any drug and then following administration of D. and L isomers. The data from Cushny are available as the `cushny` data set.

Variable	Cntrl	drug1	drg2L	drg2R	delt1	dlt2L	dlt2R
1	0.60	1.3	2.50	2.10	0.70	1.90	1.50
2	3.00	1.4	3.80	4.40	-1.60	0.80	1.40
3	4.70	4.5	5.80	4.70	-0.20	1.10	0.00
4	5.50	4.3	5.60	4.80	-1.20	0.10	-0.70
5	6.20	6.1	6.10	6.70	-0.10	-0.10	0.50
6	3.20	6.6	7.60	8.30	3.40	4.40	5.10
7	2.50	6.2	8.00	8.20	3.70	5.50	5.70
8	2.80	3.6	4.40	4.30	0.80	1.60	1.50
9	1.10	1.1	5.70	5.80	0.00	4.60	4.70
10	2.90	4.9	6.30	6.40	2.00	3.40	3.50
Mean	3.25	4.0	5.58	5.57	0.75	2.33	2.32
Sd	1.78	2.1	1.66	1.91	1.79	2.00	2.27

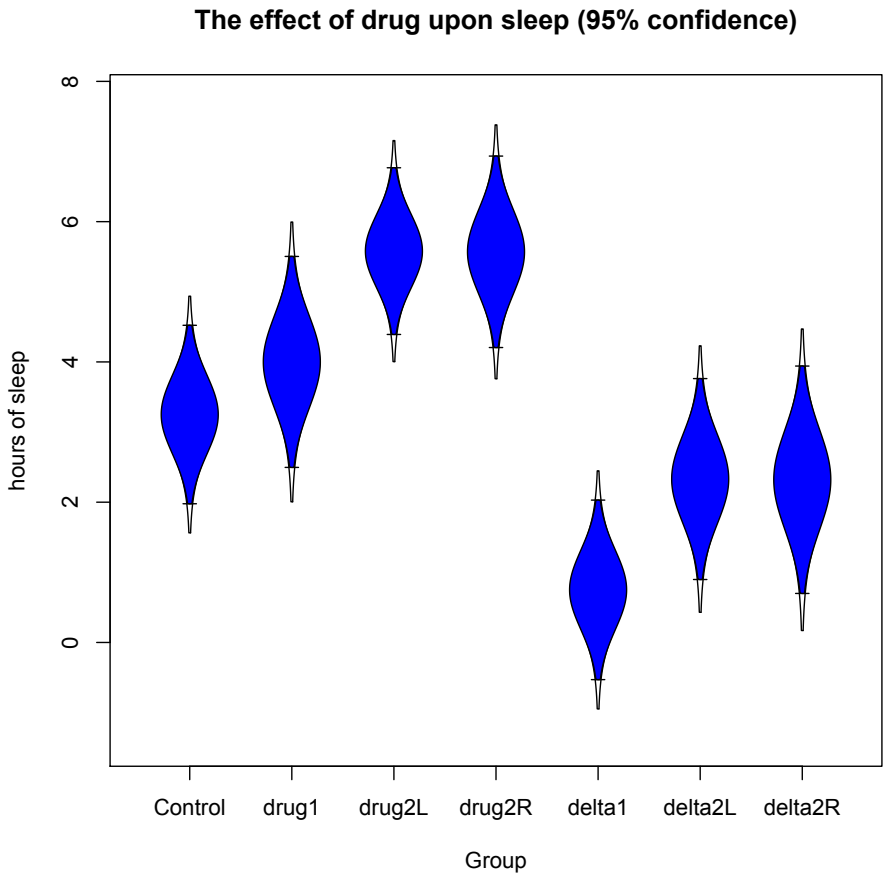
R code

```
error.bars(cushny,xlab="Group",ylab="hours of sleep",
     main="The effect of drug upon sleep (95% confidence)")
```

The cushny data set with error bars (?)

R code

```
error.bars(cushny,xlab="Group",ylab="hours of sleep",
      main="The effect of drug upon sleep (95% confidence)")
```



We can show these data graphically using the `error.bars` function. We pass labels to the x and y axis using the xlab and ylab parameters, and then supply an appropriate figure title.

We will use these data to show how to do t-tests as well as the generalization to Analysis of Variance.

Student's t.test: As done by Student

R code

```
with(cushny,t.test(delta1))  #control versus drug 1 difference scores
with(cushny,t.test(delta2L))   #control versus drug2L difference scores
with(cushny,t.test(delta1,delta2L,paired=TRUE)) #difference of differences
```

```
> with(cushny,t.test(delta1))  #control versus drug 1 difference scores
        One Sample t-test
data:  delta1
t = 1.3257, df = 9, p-value = 0.2176
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.5297804  2.0297804
sample estimates:
mean of x
     0.75
with(cushny,t.test(delta2L))   #control versus drug2L difference scores
        One Sample t-test
data:  delta2L
t = 3.6799, df = 9, p-value = 0.005076
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.8976775 3.7623225
sample estimates:
mean of x
     2.33
> with(cushny,t.test(delta1,delta2L,paired=TRUE)) #difference of differences
        Paired t-test
data:  delta1 and delta2L
t = -4.0621, df = 9, p-value = 0.002833
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.4598858 -0.7001142
sample estimates:
```

Two ways of organizing the data: Wide versus long

We can take the wide format of the cushny data set and make it long.

```
cushny[c("delta1","delta2L")]
   delta1 delta2L
1     0.7     1.9
2    -1.6     0.8
3    -0.2     1.1
4    -1.2     0.1
5    -0.1    -0.1
6     3.4     4.4
7     3.7     5.5
8     0.8     1.6
9     0.0     4.6
10    2.0     3.4
```

R code

```
long.sleep <-
  stack(cushny[c("delta1","delta2L")])
long.sleep
```

```
   values     ind
1     0.7  delta1
2    -1.6  delta1
3    -0.2  delta1
4    -1.2  delta1
5    -0.1  delta1
6     3.4  delta1
7     3.7  delta1
8     0.8  delta1
9     0.0  delta1
10    2.0  delta1
11    1.9 delta2L
12    0.8 delta2L
13    1.1 delta2L
14    0.1 delta2L
15   -0.1 delta2L
16    4.4 delta2L
17    5.5 delta2L
18    1.6 delta2L
19    4.6 delta2L
20    3.4 delta2L
```

R code

```
long.sleep <-
+   stack(cushny[c("delta1",
          "delta2L")])
```

```
> long.sleep
   values     ind
1     0.7  delta1
2    -1.6  delta1
3    -0.2  delta1
4    -1.2  delta1
5    -0.1  delta1
6     3.4  delta1
7     3.7  delta1
8     0.8  delta1
9     0.0  delta1
10    2.0  delta1
11    1.9 delta2L
12    0.8 delta2L
13    1.1 delta2L
14    0.1 delta2L
15   -0.1 delta2L
16    4.4 delta2L
17    5.5 delta2L
18    1.6 delta2L
19    4.6 delta2L
20    3.4 delta2L
```

R code

```
t.test(values ~ ind,data=long.sleep)
```

```
        Welch Two Sample t-test
data:  values by ind
t = -1.8608, df = 17.776, p-value = 0.07939
alternative hypothesis: true difference in means is not e
95 percent confidence interval:
 -3.3654832  0.2054832
sample estimates:
 mean in group delta1 mean in group delta2L
                 0.75                  2.33
```

But, the data were paired

R code

```
t.test(values ~ ind,data=long.sleep,
          paired=TRUE)
```

```
data:  values by ind
t = -4.0621, df = 9, p-value = 0.002833
alternative hypothesis: true difference in means is not e
95 percent confidence interval:
 -2.4598858 -0.7001142
sample estimates:
mean of the differences
                  -1.58
```

t.test demonstration with Student's data (from the sleep dataset)

Sleep data set is just 2 columns of cushny

R code

```
sleep
```

```
> sleep
   extra group ID
1    0.7     1  1
2   -1.6     1  2
3   -0.2     1  3
4   -1.2     1  4
5   -0.1     1  5
6    3.4     1  6
7    3.7     1  7
8    0.8     1  8
9    0.0     1  9
10   2.0     1 10
11   1.9     2  1
12   0.8     2  2
13   1.1     2  3
14   0.1     2  4
15  -0.1     2  5
16   4.4     2  6
17   5.5     2  7
18   1.6     2  8
19   4.6     2  9
20   3.4     2 10
```

R code

```
with(sleep,t.test(extra~group))
with(sleep,t.test(extra~group,var.equal=TRUE))
```

```
        Welch Two Sample t-test
data:  extra by group
t = -1.8608, df = 17.776, p-value = 0.07939  <-- default value
t = -1.8608, df = 18, p-value = 0.07919. <- equal variances
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.3654832  0.2054832
sample estimates:
mean in group 1 mean in group 2
           0.75            2.33
```

But the data were actually paired. Do it for a paired t-test

R code

```
with(sleep,t.test(extra~group,paired=TRUE))
```

```
        Paired t-test
data:  extra by group
t = -4.0621, df = 9, p-value = 0.002833
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.4598858 -0.7001142
sample estimates:
mean of the differences
                 -1.58
```

Analysis of Variance as special case of linear model

1. aov provides a wrapper to lm for fitting linear models to balanced or unbalanced experimental designs.
2. The main difference from lm is in the way print, summary and so on handle the fit: this is expressed in the traditional language of the analysis of variance rather than that of linear models.
3. If the formula contains a single Error term, this is used to specify error strata, and appropriate models are fitted within each error stratum.
4. The formula can specify multiple responses.
5. aov is designed for balanced designs, and the results can be hard to interpret without balance: beware that missing values in the response(s) will likely lose the balance.
6. If there are two or more error strata, the methods used are statistically inefficient without balance, and it may be better to use `lme` in package *nlme*.

aov of the sleep data set: compare with the t.test results

R code

```
> sleep
```

```
   extra group ID
1    0.7     1  1
2   -1.6     1  2
3   -0.2     1  3
4   -1.2     1  4
5   -0.1     1  5
6    3.4     1  6
7    3.7     1  7
8    0.8     1  8
9    0.0     1  9
10   2.0     1 10
11   1.9     2  1
12   0.8     2  2
13   1.1     2  3
14   0.1     2  4
15  -0.1     2  5
16   4.4     2  6
17   5.5     2  7
18   1.6     2  8
19   4.6     2  9
20   3.4     2 10
```

R code

```
#independent subjects
summary(aov(extra ~ group, data=sleep))
```

```
> summary(aov(extra ~ group, data=sleep))
            Df Sum Sq Mean Sq F value Pr(>F)
group        1  12.48  12.482   3.463 0.0792 .
Residuals   18  64.89   3.605
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1
t = -1.8608, df = 17.776, p-value = 0.07939. <--
t = -1.8608, df = 18, p-value = 0.07919. <- equal variances
```

R code

```
 #correlated subjects
summary(aov(extra~group + Error(ID),data=sleep))
```

```
> summary(aov(extra~group + Error(ID),data=sleep))

Error: ID
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals  9  58.08   6.453

Error: Within
          Df Sum Sq Mean Sq F value  Pr(>F)
group      1 12.482  12.482    16.5 0.00283 **
Residuals  9  6.808   0.756
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1
t = -4.0621, df = 9, p-value = 0.002833. <---
```

aov: an example of chemicals upon the growth of peas.

R code

```
npk #from Venables
```

	block	N	P	K	yield
1	1	0	1	1	49.5
2	1	1	1	0	62.8
3	1	0	0	0	46.8
4	1	1	0	1	57.0
5	2	1	0	0	59.8
6	2	1	1	1	58.5
7	2	0	0	1	55.5
8	2	0	1	0	56.0
9	3	0	1	0	62.8
10	3	1	1	1	55.8
11	3	1	0	0	69.5
12	3	0	0	1	55.0
13	4	1	0	0	62.0
14	4	1	1	1	48.8
15	4	0	0	1	45.5
16	4	0	1	0	44.2
17	5	1	1	0	52.0
18	5	0	0	0	51.5
19	5	1	0	1	49.8
20	5	0	1	1	48.8
21	6	1	0	1	57.2
22	6	1	1	0	59.0
23	6	0	1	1	53.2
24	6	0	0	0	56.0

Several models

R code

```
mod1 <- aov(yield ~ N,data=npk)
mod2 <- aov(yield ~ N+ P + N*P,data=npk)
mod2a <- aov(yield ~N*P,data=npk)
mod3 <- aov(yield ~ N*P*K,data=npk)
mod4 <- aov(yield ~ block + N*P*K,data=npk)
```

```
> summary(mod1)
            Df Sum Sq Mean Sq F value Pr(>F)
N            1  189.3  189.28   6.061 0.0221 *
Residuals   22  687.1   31.23
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

> summary(mod4)
            Df Sum Sq Mean Sq F value  Pr(>F)
block        5  343.3   68.66   4.447 0.01594 *
N            1  189.3  189.28  12.259 0.00437 **
P            1    8.4    8.40   0.544 0.47490
K            1   95.2   95.20   6.166 0.02880 *
N:P          1   21.3   21.28   1.378 0.26317
N:K          1   33.1   33.13   2.146 0.16865
P:K          1    0.5    0.48   0.031 0.86275
Residuals   12  185.3   15.44
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1
```

Analysis of Variance: Another example

aov is designed for balanced designs, and the results can be hard to interpret without balance: beware that missing values in the response(s) will likely lose the balance.

R code

```
datafilename="http://personality-project.org/r/datasets/R.appendix2.data"
data.ex2=read.file(datafilename)   #read the data into a data.frame
data.ex2                                    #show the data
```

```
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
```

R code

```
#do the analysis of variance
aov.ex2 = aov(Alertness~Gender*Dosage,data=data.ex2)
summary(aov.ex2)        #show the summary table
```

```
Call:

 summary(aov.ex2)        #show the summary table
              Df Sum Sq Mean Sq F value Pr(>F)
Gender         1  76.56   76.56   2.952  0.111
Dosage         1   5.06    5.06   0.195  0.666
Gender:Dosage  1   0.06    0.06   0.002  0.962
Residuals     12 311.25   25.94
```

Analysis of Variance

Do the analysis of variances and the show the table of results.

R code

```
#do the analysis of variance
aov.ex2 <- aov(Alertness ~ Gender * Dosage, data=data.ex2)

summary(aov.ex2)        #show the summary table
aov.ex2.  #This shows the coefficients
```

```
>aov.ex2 <- aov(Alertness ~ Gender * Dosage, data=data.ex2)
> summary(aov.ex2)        #show the summary table
             Df Sum Sq Mean Sq F value Pr(>F)
Gender        1  76.56   76.56   2.952  0.111
Dosage        1   5.06    5.06   0.195  0.666
Gender:Dosage 1   0.06    0.06   0.002  0.962
Residuals    12 311.25   25.94

  aov(formula = Alertness ~ Gender * Dosage, data = data.ex2)

Terms:
                   Gender   Dosage Gender:Dosage Residuals
Sum of Squares   76.5625   5.0625        0.0625  311.2500
Deg. of Freedom        1        1             1        12

Residual standard error: 5.092887
Estimated effects may be unbalanced
```

Show the results table

R code

```
print(model.tables(aov.ex2,"means"),digits=3)
```

```
> 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
```