Lab 8 - ANOVA, Correlation, and Regression

Contents

Lab Objectives:

- Compare the means of multiple groups using Analysis of Variance (ANOVA)
- Make post-hoc comparisons with the Tukey-Kramer test
- Analyze the association between two numerical variables using correlation and regression

---

Exercise 1: ANOVA

A **One Way ANOVA** is an analysis of variance in which there is only one independent variable. It is used to compare mean differences in two or more groups. An ANOVA with two groups is mathematically equivalent to a two-sample two-tailed t-test. A significant ANOVA result implies that at least one group has a different mean from one other group, but does not directly tell you what the significant difference is. Post-hoc comparisons, such as the **Tukey-Kramer** test, can be used to determine which group or groups are significantly different from the others.

ANOVA assumes equal variance of each group, that each group has a normal distribution of the variable, and that each population is randomly sampled. The **Kruskal-Wallis** test is a nonparametric test that compares multiple groups. It is often used in place of ANOVA when the assumptions of ANOVA are violated.

Maternal Role Adaptation

Various factors can influence how well new mothers adapt to a maternal role. Low birth weight of a baby due to premature labour is believed to increase psychological stress and risk of depression in new mothers, reducing maternal role adaptation. In this study, maternal role adaptation was compared in a group of mothers of low birth-weight (LBW) infants, mothers of LBW infants who had an experimental intervention, and mothers of full-term infants. The hypothesis was that mothers of LBW infants in the experimental intervention would adapt to their maternal role as well as mothers of healthy full-term infants, and that each of these groups would adapt better than mothers of LBW infants in the control group. A lower maternal role adaptation score indicates better adaptation.

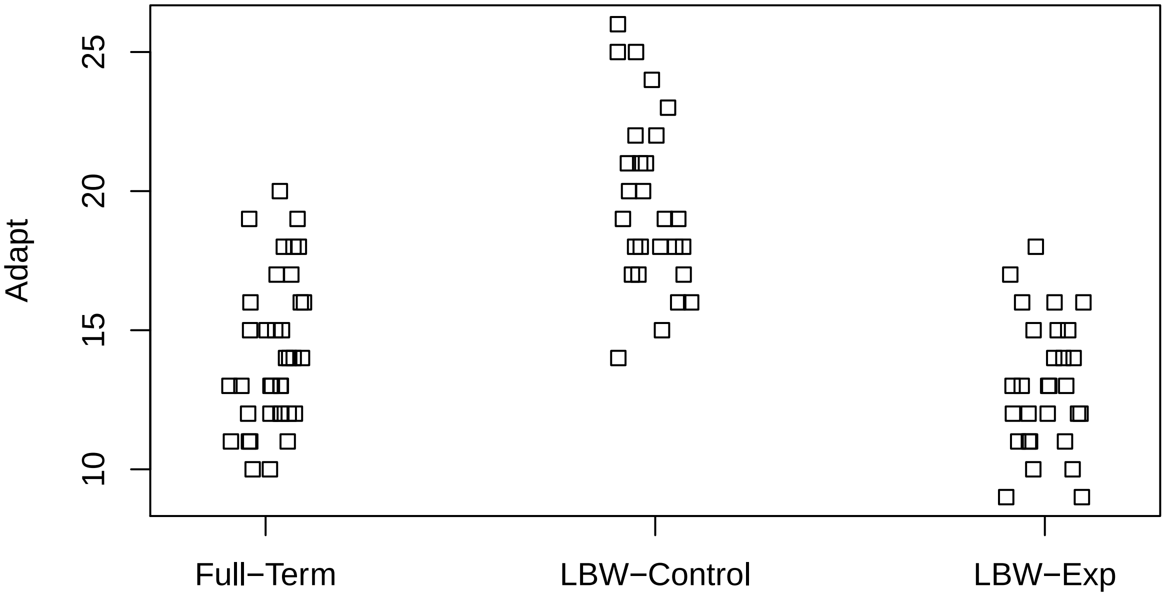
Open the file maternal role adaptation.csv

```
d <- read.csv("maternal role adaptation.csv")
str(d)
```

```
## 'data.frame':    93 obs. of  2 variables:
##  $ Group: Factor w/ 3 levels "Full-Term","LBW-Control",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ Adapt: int  9 9 10 10 11 11 11 11 12 12 ...
```

Visualise and describe the data requires some handiwork in R:

```
# Strip chart of adaptation by group.
stripchart(Adapt ~ Group, data = d, method = "jitter", vertical = TRUE)
```

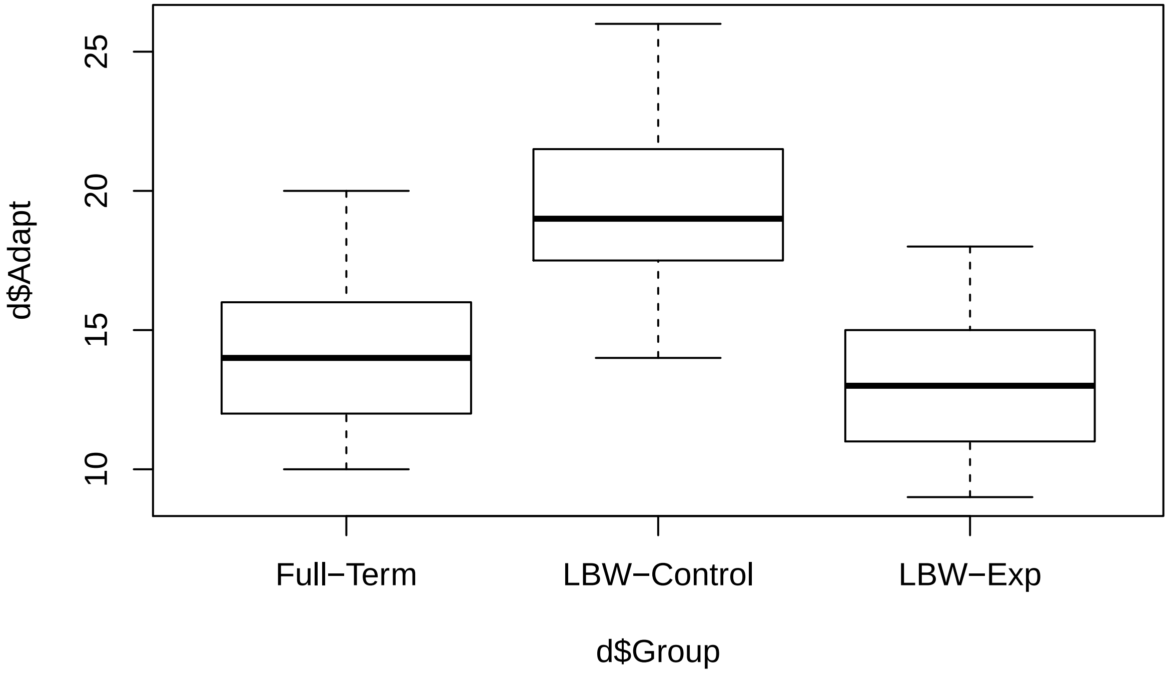


```
# calculate mean, sd and n using the tapply function
meanAdapt <- tapply(d$Adapt, d$Group, mean)
sdevAdapt <- tapply(d$Adapt, d$Group, sd)
n <- tapply(d$Adapt, d$Group, length)
data.frame(mean = meanAdapt, std.dev = sdevAdapt, n = n)
```

```
##                 mean  std.dev  n
## Full-Term   14.16216 2.713275 37
## LBW-Control 19.59259 3.165429 27
## LBW-Exp     13.00000 2.345208 29
```

ANOVA is robust to deviations from normality, particularly when sample sizes are large. It is also robust to departures from the assumption of equal variance, but only if the samples are all large, about the same size, and there is no more than about a tenfold difference among variances. However, it is still a good idea to check whether your data meet these assumptions. Homeogeneity of variance is simple enough to do with the **leveneTest()**:

```
plot(d$Adapt ~ d$Group)
```



```
library(car)
```

```
## Loading required package: carData
```

```
leveneTest(Adapt ~ Group, data = d)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  2  1.1527 0.3204
##       90
```

Visual inspection of the boxplot above suggests similar variance within each group, and the Levene test confirms this.

Variance tests like the Levene Test assume the underlying data do not depart from normality. It does not make sense to examine normality on the raw data, especially if there are obvious groupings within the data, but the **shapiro.test** only works on simple datasets. To examine normality of each group in R requires we create a simple function, **foo()** that will return only the p value from a **shapiro.test()** and then we aggregate the output on our data using the formula command. In R terminology, this function is a “wrapper” for another function, since it wraps up an output.

Note: **foo()** is commonly used nomenclature for a simple function in coding in R where the coder does not want to create a permanent function with a descriptive name!

```
foo <- function(x) shapiro.test(x)$p.value
aggregate(Adapt ~ Group, data = d, FUN = foo)
```

```
##         Group      Adapt
## 1   Full-Term 0.07868732
## 2 LBW-Control 0.45830380
## 3     LBW-Exp 0.59777791
```

```r
setNames(aggregate(Adapt ~ Group, data = d, FUN = foo), c("Group", "p-val"))
```

```
##         Group      p-val
## 1   Full-Term 0.07868732
## 2 LBW-Control 0.45830380
## 3     LBW-Exp 0.59777791
```

This returns a table of p values for each Shapiro test run on each group (we use setNames to make the output easier to follow).

Technically, it is the normality of the residuals that really needs to be assessed in an ANOVA, although that may come later. Generally, tests of normality on the raw data will be underpowered, especially if the sample sizes within each group are small. In the case here, all p values are > 0.05 above, so the data appear to be consistent with a normal distribution.

Now, run the ANOVA, using the **aov()** function, and run a **summary()** on the result:

```r
result <- aov(Adapt ~ Group, data = d)
summary(result)
```

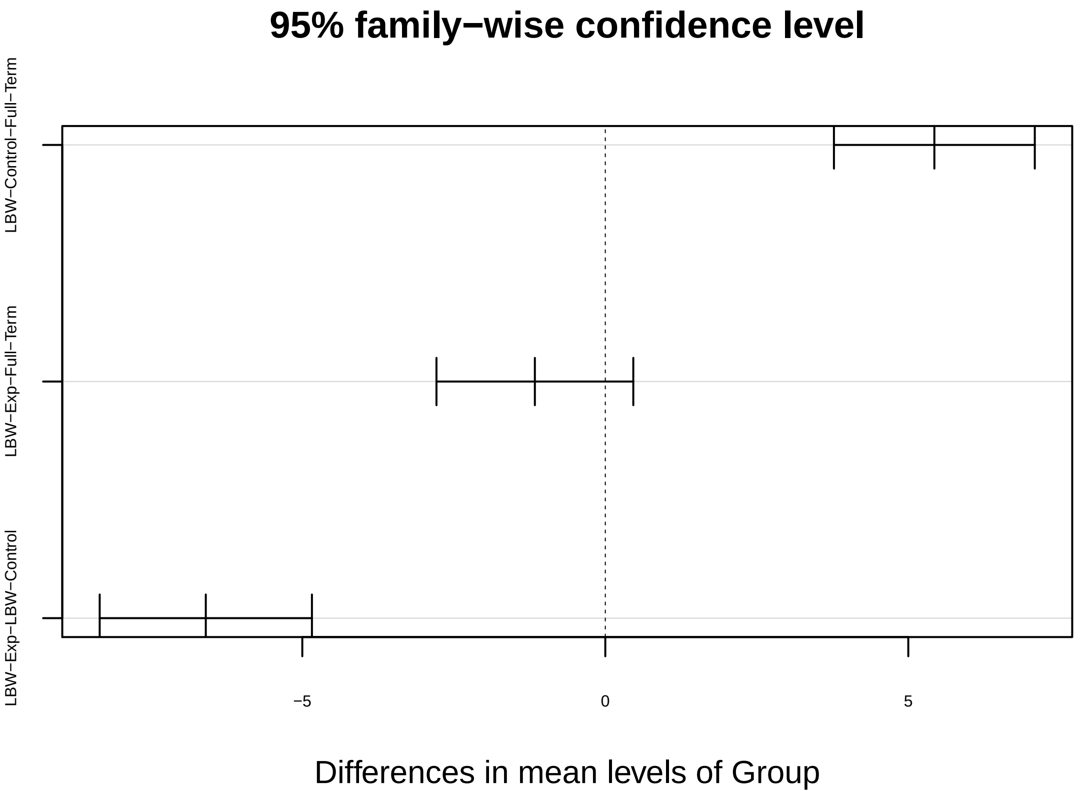
```
##             Df Sum Sq Mean Sq F value   Pr(>F)    
## Group        2  698.3   349.1   46.24 1.53e-14 ***
## Residuals   90  679.5     7.6                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We will learn how to calculate the important values in an ANOVA table in class. The table shows you the calculated variance ratio (F) and the corresponding P-value under “Pr(>F)”. A P-value less than 0.05 indicates that mean adaptation score (in this case) differs among treatments. However, this alone does not tell us which group or groups are significantly different.

```r
tukeyres <- TukeyHSD(result, conf.level = 0.95)
tukeyres
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
## 
## Fit: aov(formula = Adapt ~ Group, data = d)
## 
## $Group
##                            diff       lwr       upr     p adj
## LBW-Control-Full-Term  5.430430  3.772991  7.087870 0.0000000
## LBW-Exp-Full-Term     -1.162162 -2.786224  0.461900 0.2087986
## LBW-Exp-LBW-Control   -6.592593 -8.343825 -4.841361 0.0000000
```

```r
plot(tukeyres, cex.axis = 0.5)
```



To determine how the groups differ, view the output above which contains the results of the Tukey-Kramer post hoc test, comparing all pairs of means. It shows the mean difference for each combination of groups, the lower and upper confidence limits of the difference, and the significance value (P-value). For p values < 0.0001, the value will be listed as 0.0000.

The results of the Tukey-Kramer test are exact when the sample size in every group is the same. If sample sizes are different, then the test is conservative, which means that the probability of making at least one Type I error is lower than the stated alpha. The p adj refers to adjusted p values for cases for unbalanced designs.

So in this case, the means for the Full term and LBW experimental groups are different from the Control.

The box and whisker plot we made earlier allows you to visualize the difference between group means using a graphical method. If you examine this plot you can quickly see that the LBW Control group had a higher adaptability score (indicating poorer adaption to a maternal role) than the LBW experimental and Full term groups.

Answer questions 1 and 2 on Sakai

Circadian clock

In a 1998 study, Campbell and Murphy reported that the human circadian clock can be reset by exposing the back of the knee to light, which was met with much skepticism. A later experiment re- examined the phenomenon using 22 people who were awakened from sleep to receive a 3-hour light treatment of the eyes, knees, or neither (control). Effect on circadian rhythm was determined by measuring melatonin production two days later. Melatonin secretion is tightly connected to circadian rhythm, with secretion beginning to rise about two hours before a person's regular bedtime. A negative measurement indicates a delay in melatonin production, while a positive number indicates an advance. The data is found in the file Knees circadian clock.csv. Use a one-way ANOVA to assess whether light-treatment affected circadian rhythm.

```r
d <- read.csv("Knees circadian clock.csv")
str(d)
```

```
## 'data.frame':    22 obs. of  2 variables:
##  $ treatment: Factor w/ 3 levels "Control","Eyes",..: 1 1 1 1 1 1 1 1 3 3 ...
##  $ shift    : num  0.53 0.36 0.2 -0.37 -0.6 -0.64 -0.68 -1.27 0.73 0.31 ...
```

Answer questions 3 and 4 on Sakai

Memory recall

In a study of memory recall, a group of subjects were asked to read through a list of 27 words. One group was assigned to count the number of letters in each word. Another group was assigned to think of a word that rhymed with each word. The third group was instructed to think of an adjective that could be used to modify each word. An imagery group was asked to form vivid images of each word. These first four groups were not told that they would later have to recall the word list. Finally, the last group ("Intentional") was asked to memorize the words as best as they could for later recall. Each group reviewed the list of 27 words three times, and was then asked to write down as many words as they could remember. The data is found in the file Recall.csv. Examine the data to determine if there is any difference between the groups.

```r
d <- read.csv("Recall.csv")
str(d)
```

```
## 'data.frame':    50 obs. of  2 variables:
##  $ Group : Factor w/ 5 levels "Adjective","Counting",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Recall: int  9 8 6 8 10 4 6 5 7 7 ...
```

Answer questions 5 – 8 on Sakai.

Cuckoos Eggs

The European cuckoo does not look after its own eggs, but instead lays them in the nests of birds of other species. Do cuckoos lay eggs of different sizes in nests of different hosts? The data file cuckoo eggs.csv contains data on the lengths of cuckoo eggs laid in a variety of other species' nests. Examine the data to determine if there is any difference between the groups. Use a Tukey-Kramer test to determine which pairs of host species are significantly different from each other.

```r
d <- read.csv("Cuckoo eggs.csv")
str(d)
```

```
## 'data.frame':    105 obs. of  2 variables:
##  $ HostSpecies: Factor w/ 6 levels "Hedge Sparrow",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ EggLength  : num  20.9 21.6 22.1 22.9 23.1 ...
```

Answer question 9 on Sakai.

Exercise 2: Kruskal-Wallis test

If the data do not meet the assumptions of the ANOVA test (normal distribution and equal variances in all populations), and the data are not improved by data transformation, a Kruskal-Wallis test can be used instead. The Kruskal-Wallis test is a nonparametric method to compare more than two groups.

Malaria & Maize

The pollen of the corn (maize) plant is known to be a source of food for larval mosquitoes of the species Anopheles arabiensis, the main vector of malaria in Ethiopia. The production of maize has increased substantially in certain areas of Ethiopia recently, and over the same time malaria has entered in to new areas where it was previously rare. This raises the question: Is the increase of maize cultivation partly responsible for the increase in malaria?

Data in the file malaria maize.csv contains information on the level of cultivation of maize (low, medium or high) and the rate of malaria per 10,000 people for several sites in Ethiopia. Assume that you know from prior analysis that the incidence rate of malaria is not normally distributed.

Since we know the data are not normally distributed, and our samples are quite small, we will conduct a non-parametric test.

```
d <- read.csv("Malaria maize.csv")
str(d)
```

```
## 'data.frame':    19 obs. of  3 variables:
##  $ Villages          : Factor w/ 19 levels "Adel Agata                                   ",..: 9 6 13
##  $ Maizeyield        : Factor w/ 3 levels "High","Low","Medium": 1 1 1 1 1 1 1 1 3 3 ...
##  $ Incidencerate10000: int  291 43 133 219 176 301 466 246 59 83 ...
```

```
ktmaize <- kruskal.test(Incidencerate10000 ~ Maizeyield, data = d)
print(ktmaize)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Incidencerate10000 by Maizeyield
## Kruskal-Wallis chi-squared = 13.712, df = 2, p-value = 0.001053
```

```
ktmaize$p.value
```

```
## [1] 0.00105334
```

If the Kruskal–Wallis test is significant, a post-hoc analysis can be performed to determine which groups differ from each other group.

Probably the most popular host-hoc test for the Kruskal–Wallis test is the Dunn test. The Dunn test can be conducted with the **dunnTest()** function in the FSA package.

Because the post-hoc test will produce multiple p-values, adjustments to the p-values can be made to avoid inflating the possibility of making a type-I error. There are a variety of methods for controlling the familywise error rate or for controlling the false discovery rate. See ?p.adjust for details on these methods.

The resulting table shows us the significance value for each pairwise comparison, and an adjusted significance value according to the number of pairwise tests made. This adjusted significance helps keep the probability of type I error (α) at 0.05 total across all of the comparisons.

```
# install.packages('FSA')
library(FSA)
DT <- dunnTest(Incidencerate10000 ~ Maizeyield, data = d, method = "bh")
DT
```

```
##      Comparison         Z      P.unadj       P.adj
## 1    High - Low  3.701757 0.0002141113 0.000642334
## 2 High - Medium  1.418301 0.1561027793 0.156102779
## 3  Low - Medium -1.966246 0.0492701698 0.073905255
```

When there are many p-values to evaluate, it is useful to condense a table of p-values to a compact letter display format. In the output, groups are separated by letters. Groups sharing the same letter are not significantly different. Compact letter displays are a clear and succinct way to present results of multiple comparisons.

```
### Compact letter display
PT <- DT$res
PT
```

```
##      Comparison         Z      P.unadj       P.adj
## 1    High - Low  3.701757 0.0002141113 0.000642334
## 2 High - Medium  1.418301 0.1561027793 0.156102779
## 3  Low - Medium -1.966246 0.0492701698 0.073905255
```

```
# install.packages('rcompanion')
library(rcompanion)
cldList(P.adj ~ Comparison, data = PT, threshold = 0.05)
```

```
##    Group Letter MonoLetter
## 1   High      a         a
## 2    Low      b          b
## 3 Medium     ab         ab
```

Answer question 10 on Sakai

Nematode lifespan

An experiment in 2005 examined the effect of the anticonvulsant medication trimethadione on the lifespan of the nematode worm *C. elegans*. The study compared the effect of treatment provided at the larval stage, the adult stage, and both stages, to control worms treated with water only. Examine the data found in the file nematode.csv for deviations from the assumptions of a one-way ANOVA test. If the data violate these assumptions, perform a Kruskal-Wallis test to determine whether any of the treatments had a significantly different lifespan compared with the water control.

```
d <- read.csv("nematode.csv")
str(d)
```

```
## 'data.frame':    200 obs. of  2 variables:
##  $ treatment: Factor w/ 4 levels "adult treatment",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ lifespan : int  6 10 12 12 13 13 14 14 14 14 ...
```

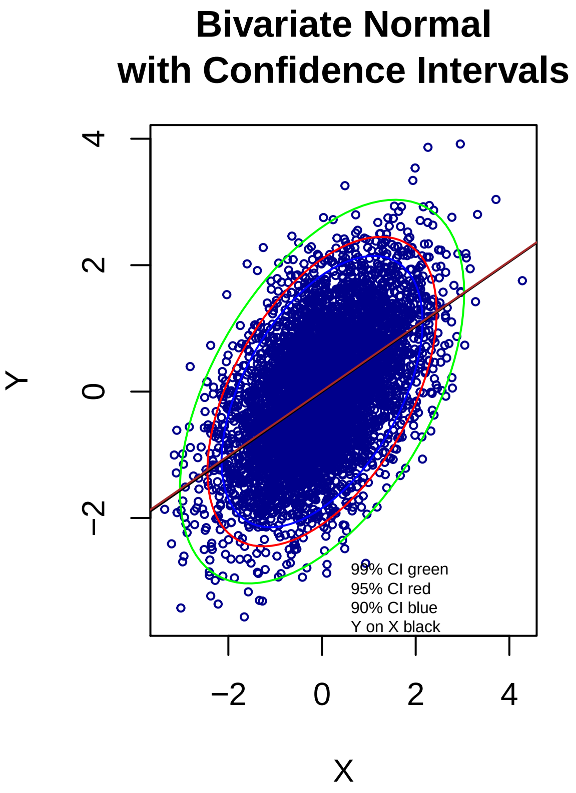
Answer question 11 on Sakai

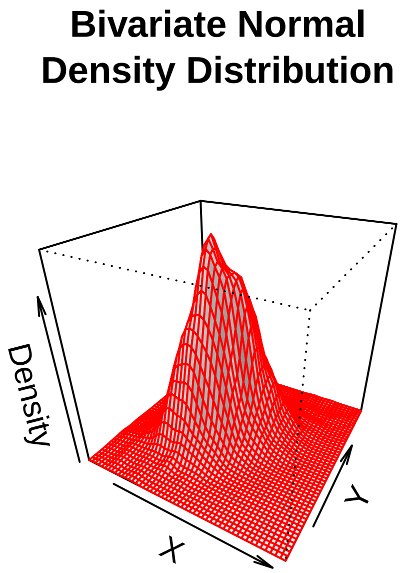
Exercise 3: Correlation

When two numerical variables are associated, we say they are correlated. The correlation coefficient is a quantity that describes the strength and direction of an association. It reflects the amount of “scatter” in a scatter plot of the two variables, but unlike linear regression does not measure how steeply one variable changes when the other changes. The maximum correlation possible is 1.0 or - 1.0. A correlation of 1.0 indicates that the measurements lie along a straight linear line and show a positive relationship (as one variable increases, the other increases as well). In comparison, a correlation of 0.5 will have a generally positive relationship, but a lot more scatter among the points. A correlation of -1.0 indicates that the measurements lie along a straight linear line and show a negative relationship (as one variable increases, the other decreases).

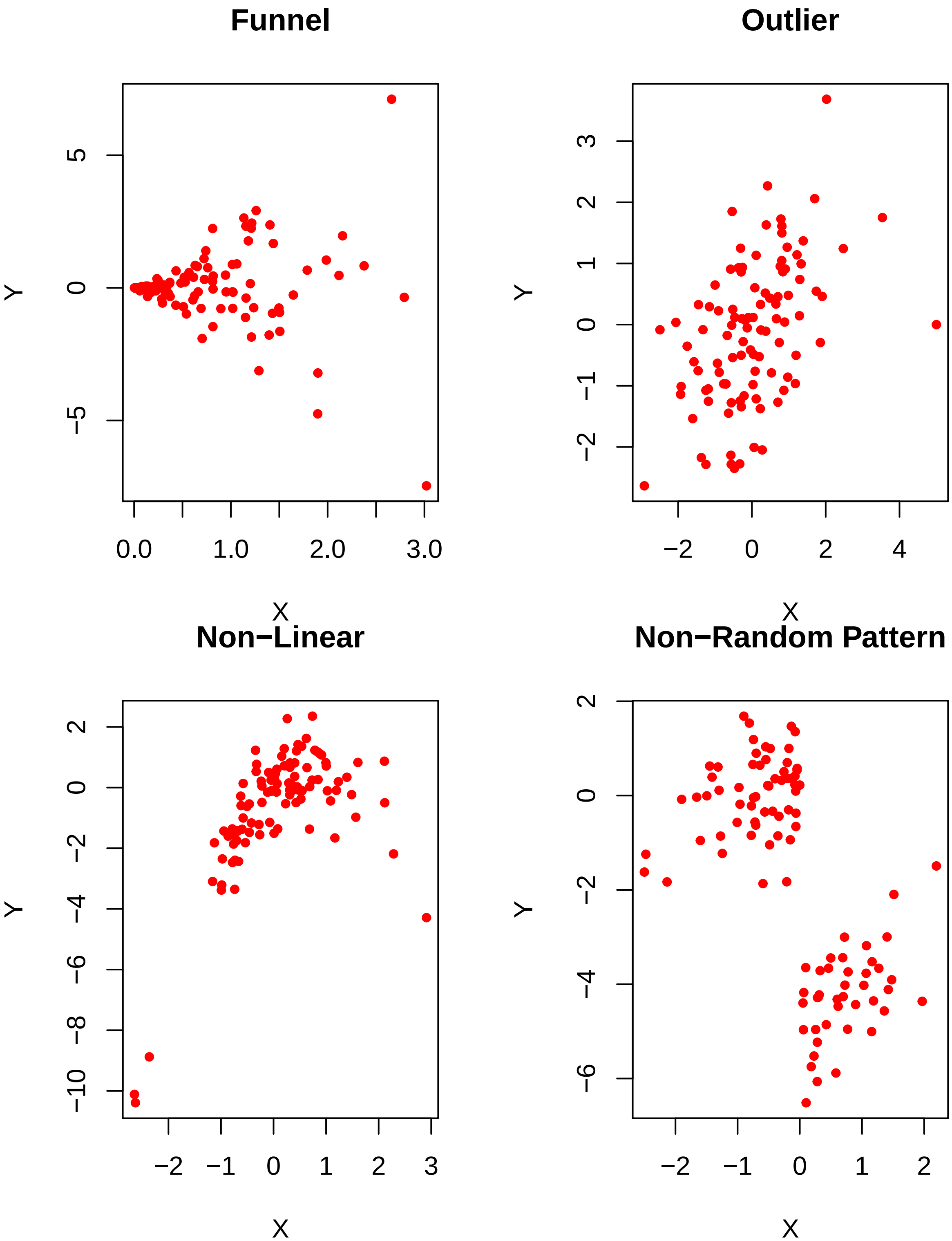
Correlation analysis assumes that the measurements have a bivariate normal distribution, which is a normal distribution in two dimensions rather than one (see Figures below). A bivariate normal distribution occurs when the relationship between X and Y is linear, the cloud of points on a scatter plot has a circular or elliptical shape, and the frequency distributions of X and Y separately are normal. Visualizing a scatter plot of the data is an important step to detect deviations from these assumptions. Histograms depicting the frequency distributions of X and Y separately are also helpful.

A bivariate normal distribution





Examples of data that deviate from a bivariate normal distribution.



If the assumptions of correlation analysis are violated, then data transformation may be tried. If data transformation does not improve the fit of the data to bivariate normality, then Spearman's rank correlation is used instead.

Height & weight by gender

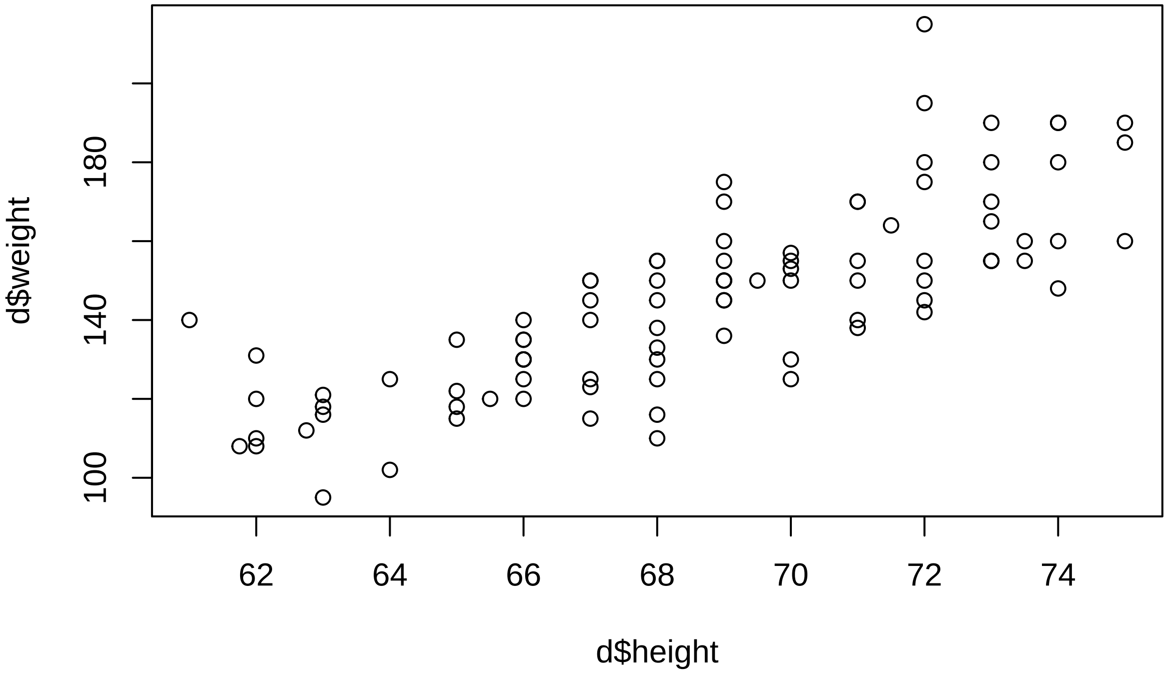
Data comparing the height and weight of a sample of males and females is found in the file Heightweight.csv. Open this file now.

```
d <- read.csv("Heightweight.csv")
str(d)
```

```
## 'data.frame':    92 obs. of  3 variables:
##  $ sex   : Factor w/ 2 levels "female","male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ height: num  66 72 73.5 73 69 73 72 74 72 71 ...
##  $ weight: int  140 145 160 190 155 165 150 190 195 138 ...
```

In this case, we know from prior knowledge that the variables height and weight are normally distributed. Construct a scatter plot to determine whether the relationship between the two variables appears to fit a bivariate normal distribution.

```
plot(d$weight ~ d$height)
```



Now, use the **cor.test()** function, defining x as height and y as weight. Specify alternative="two.sided" for the significance test. Note that if you are working with data that violate the assumptions of correlation analysis, you can conduct a Spearman's rank correlation by setting method="spearman". By default, method="pearson", and does not need to be indicated except to be explicit.

```
cor.test(x = d$height, y = d$weight, alternative = "two.sided", method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  d$height and d$weight
## t = 12.016, df = 90, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6911552 0.8526213
## sample estimates:
##       cor
## 0.7848664
```

In the output above, you will first see a summary of the descriptive statistics, followed by the confidence interval for the pearson correlation coefficient. The Pearson Correlation coefficient is a measure of the direction and strength of the relationship between the two variables. The significance value indicates whether or not this association is statistically significant.

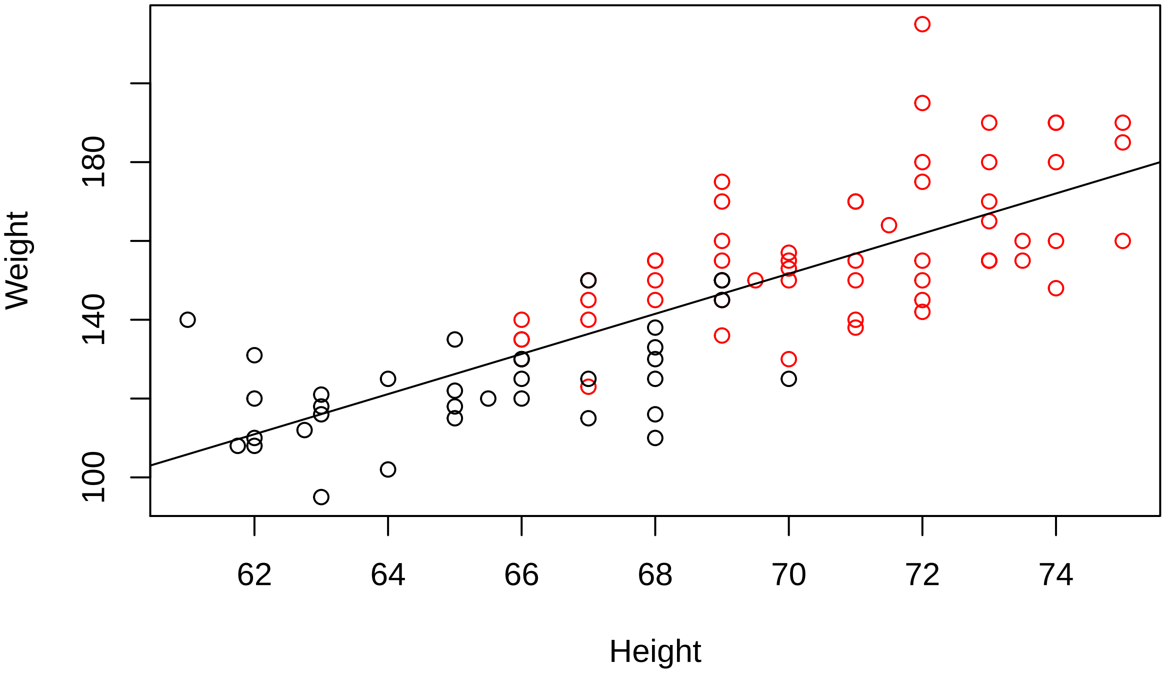
The data contains data for both males and females. If we want to conduct subgroup correlations for males and females separately, the easiest way to do this is to subset our data file, into two new dataframes, dm and df, and re-run the analyses above on the separate data frames:

```
dm <- subset(d, sex == "male")
df <- subset(d, sex == "female")

# Will leave these commented out, so you have to run the code yourself:

# cor.test(x=dm$height, y=dm$weight, alternative='two.sided')
# cor.test(x=df$height, y=df$weight, alternative='two.sided')
```

The descriptive statistics and correlations should now show up separately for males and females.

Let's go back to the data and create a scatterplot:

```
plot(d$weight ~ d$height, col = d$sex, xlab = "Height", ylab = "Weight")
abline(lm(weight ~ height, d))
```



```
corresult2 <- cor.test(d$height, d$weight)$estimate^2
corresult2
```

```
##       cor
## 0.6160153
```

```
rsquared <- summary(lm(weight ~ height, data = d))$r.squared
rsquared
```

```
## [1] 0.6160153
```

```
corresult2 == rsquared  # Are they the same?
```

```
##  cor
## TRUE
```

The **abline()** function allows you to add a line that is a linear regression through the data. We will come to linear models later.

The correlation test provides an R statistic.

A summary of a regression model also provides an r^2 statistic. Due to formatting issues, it is often written simply as R2 or R-squared. Above, we demonstrate that the result from the **cor.test()** squared is equivalent to the R2 statistic obtained from the **lm()** command. The two values are closely related – the correlation coefficient is R, and R-Squared is simply this value squared.

This r^2 statistic is used in regression analysis. It measures the fraction of the variation in Y that is explained by X.

Answer question 12 on Sakai

Chocolate & Nobel Prizes

There is evidence that higher consumption of foods containing chemicals called flavonols (including cocoa, red wine, green tea, and some fruits) increases brain function. Messerli (2012) examined whether chocolate consumption in a country is correlated with the number of Nobel prizes. The data are found in chocolate.sav. Begin by constructing a scatter plot to view the data. You should see that the cloud of points are funnel shaped (wider at one end then the other), which indicates deviation from bivariate normality. Imagine that you try many types of data transformation that does not improve the fit of the data to bivariate normality. Conduct an appropriate correlation test to analyze the association between the two variables.

```
d <- read.csv("chocolate.csv")
str(d)
```

```
## 'data.frame':    23 obs. of  3 variables:
##  $ country                 : Factor w/ 23 levels "Australia                ",..: 1
##  $ chocolateConsumption    : num  4.5 8.5 4.4 2.9 3.9 0.7 8.5 7.3 6.3 11.6 ...
##  $ nobelPrizesper100million: num  5.5 24.4 8.6 0 6 0 25.3 7.6 9 12.7 ...
```

Answer question 13 on Sakai.

You may receive a warning message like the following, which is safe to ignore:

```
## Warning in cor.test.default(x, y, method = "spearman"): Cannot compute
## exact p-value with ties
```

ADD, IQ, and GPA

Open the file appendixD.csv, which contains data that you have previously analyzed this term. Calculate the correlation coefficients for ADD score, IQ, GPA and English grade. Note that you can add all variables at the same time, and R will compute the correlation coefficient and significance for each pair of variables. Assume that there are no deviations from bivariate normality in this case.

We will demonstrate this using slightly different sets:

```
d <- read.csv("appendixd.csv")
str(d)
```

```
## 'data.frame':    88 obs. of  10 variables:
##  $ ID     : int  27 25 75 74 11 63 32 17 67 34 ...
##  $ addsc  : int  26 29 30 33 34 34 35 36 36 37 ...
##  $ sex    : Factor w/ 2 levels "female","male": 2 1 1 1 2 1 2 1 1 2 ...
##  $ repeat.: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ iq     : int  137 127 109 106 131 107 111 121 114 118 ...
##  $ engl   : int  2 1 1 1 2 1 2 1 2 2 ...
##  $ engg   : int  3 4 4 4 4 4 2 4 3 4 ...
##  $ gpa    : num  3 3.75 3.5 3.75 3.75 3.5 2.25 3.55 3.5 3.25 ...
##  $ socprob: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ dropout: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
```

How to create a correlation matrix with significance levels (p-value)?

The function **rcorr()** (in Hmisc package) can be used to compute the significance levels for pearson and spearman correlations. It returns both the correlation coefficients and the p-value of the correlation for all possible pairs of columns in the data table.

Simplified format:

```
library(Hmisc)
# rcorr(x, type = c('pearson','spearman'))
```

x should be a matrix, where each column corresponds to a variable of interest, and each row corresponds to the value. The correlation type can be either pearson or spearman.

For large datasets, you probably want to only run this on specific columns of interest. For example, let's select 4 columns of interest put these into a new dataframe, dsub:

```
dsub <- d[c("gpa", "engg", "iq")]

library(Hmisc)
cor1 <- rcorr(as.matrix(dsub))
cor1
```

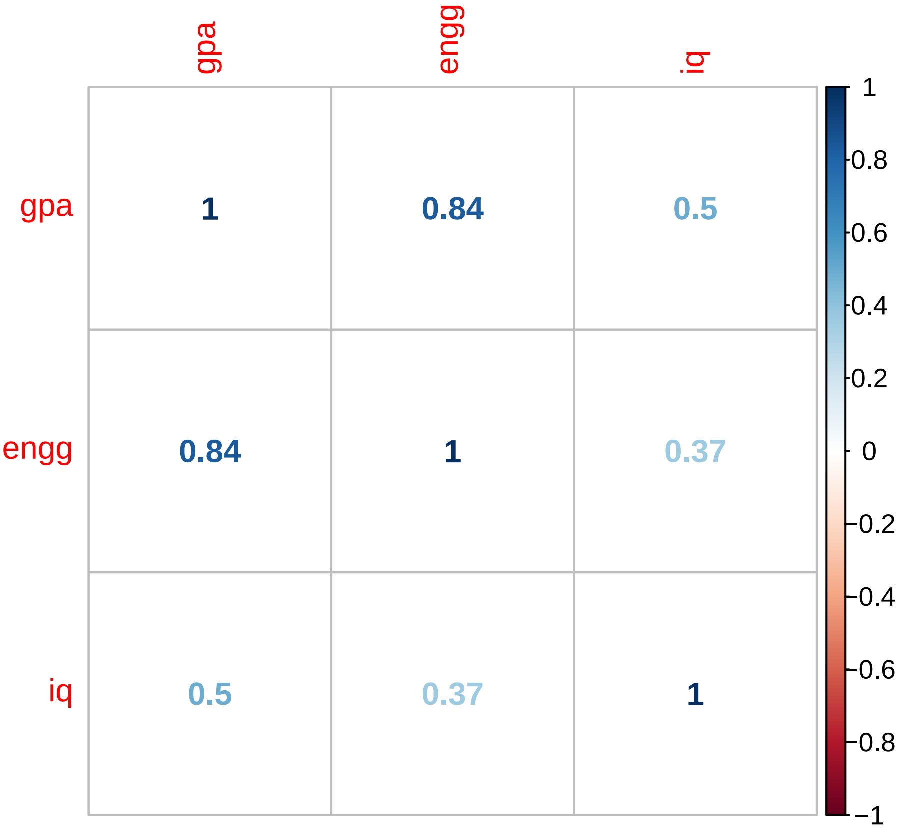
```
##       gpa engg   iq
## gpa  1.00 0.84 0.50
## engg 0.84 1.00 0.37
## iq   0.50 0.37 1.00
##
## n= 88
##
##
## P
##      gpa   engg  iq
## gpa        0e+00 0e+00
## engg 0e+00       4e-04
## iq   0e+00 4e-04
```

rcorr() returns two tables, the first is a table of correlation coefficients, where you read row name and column name to ascertain the correlation. This is often referred to as a correlation matrix.

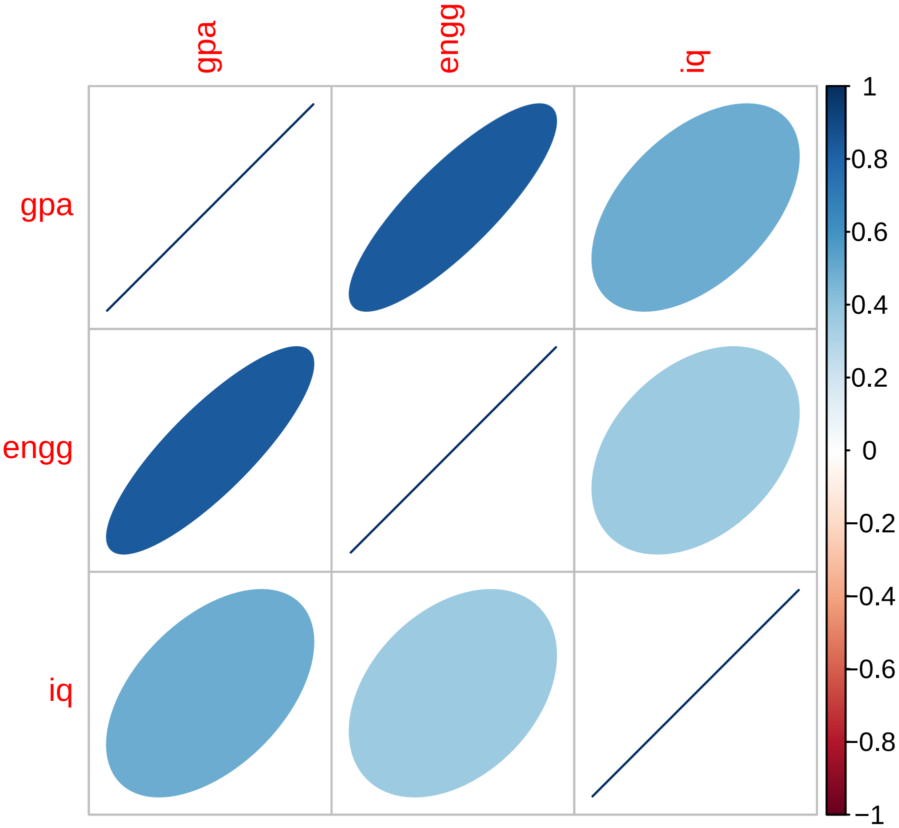
The second table is a table of p values corresponding to the correlation coefficients above. Along the diagonal there are no p values since this corresponds to autocorrelation and we do not need to compare the unique variable's correlation with itself.

For the graphically minded, see https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html:

```r
library(corrplot)
corrplot(cor(dsub), method = "number")
```



```r
corrplot(cor(dsub), method = "ellipse", p.mat = rcorr(as.matrix(dsub))$P,
    sig.level = 0.01)
```



Answer question 14 on Sakai

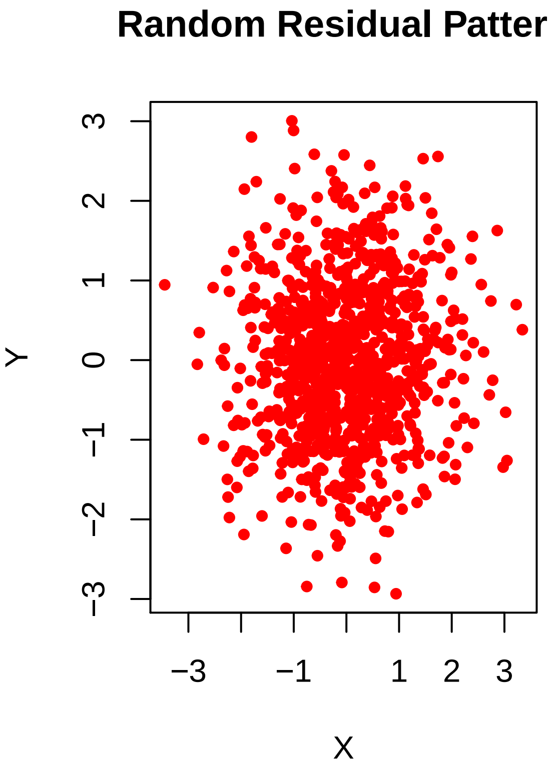
Exercise 4: Regression

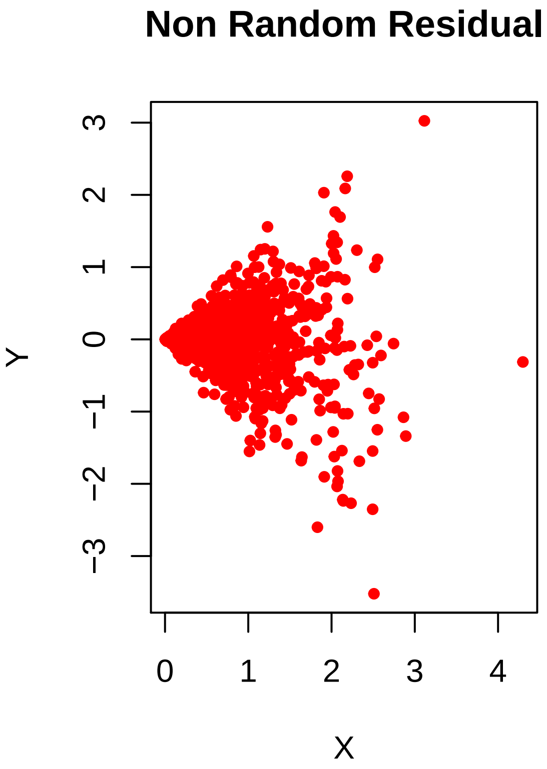
Linear regression uses a line to predict a response numerical variable (Y) from a numerical explanatory variable (X). The regression line will take the form Y = a + b X. In this equation, a is the intercept (where the line crosses the axis at X = 0), and b is the slope of this regression line.

A common statistic used in regression analysis is R2. R2 measures the fraction of variance in Y that is predicted by X. If R2 is close to one (the maximum possible value), then X predicts most of the variation in the values of Y, and the Y observations are tightly clustered around the regression line with little scatter.

Hypothesis tests about regression slopes can be made using either a t-test or ANOVA approach. A P-value of less than 0.05 indicates that the slope between the two variables is significantly different from zero.

Linear regression assumes that the true relationship between X and Y is linear, that for every value of X the corresponding values of Y are normally distributed, and that the variance of Y values is the same for all values of X. Non-linearity is best detected by visualizing a scatter plot of the data. Non-linear relationships between X and Y can often be examined by the use of transformations. Non-normality and unequal variance is best examined with a residuals plot, which calculates the difference between every Y data point and the predicted Y according to the regression equation, and plots this against X. A residuals plot should have a roughly symmetric cloud of points above and below zero, a higher density of points close to zero, and no noticeable change from left to right. The plot on the left below fits these requirements well, while the plot on the right does not.





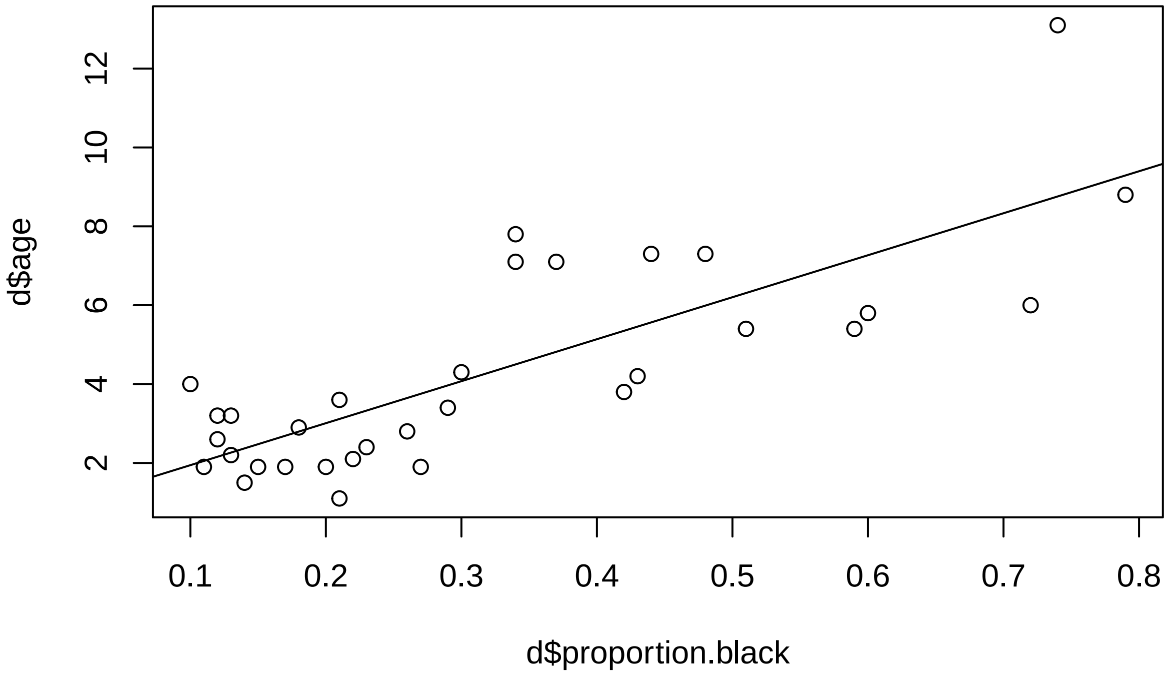
The Lion Nose

Whitman et al (2004) examined the relationship between age and the proportion of black pigmentation on the noses of 32 male lions of known age in Tanzania. The goal was to be able to predict the age of a lion from the amount of black on its nosepad. Open the file Lions ages.csv to view the data.

```r
d <- read.csv("Lions ages.csv")
str(d)
```

```
## 'data.frame':    32 obs. of  2 variables:
##  $ age             : num  1.1 1.5 1.9 2.2 2.6 3.2 3.2 2.9 2.4 2.1 ...
##  $ proportion.black: num  0.21 0.14 0.11 0.13 0.12 0.13 0.12 0.18 0.23 0.22 ...
```

It is always good practice to examine your data graphically before conducting statistical analysis. Construct a scatter plot to view the relationship between proportion black (X axis) and lion age (Y axis). Using the **abline()** function, add a linear fit line. This approach quickly allows you to view the linear equation describing the relationship between two variables, as well as the R2 value.



To perform a regression, we will use the **lm()** function to fit a linear model. This function is a work-horse in R for a family of statistical models. It is usually easiest to create an object to hold your regression analysis. We will use **lm1** for starters, setting proportion.black as the independent variable and age as the dependent varible. The variables are added in this order because we are interested in predicting a lion's age from the proportion of black pigmentation on its nose, which can be measured.

```
lm1 <- lm(age ~ proportion.black, d)
```

The object, lm1, now contains much useful information. Type summary(lm1) for most of the useful output, coefficients(lm1) for the regression coefficients, confint(lm1) for confidence limits associated with the regression coefficients, fitted(lm1) for the regression fits (or predictions), and residuals(lm1) for the regression residuals:

```
summary(lm1)
```

```
##
## Call:
## lm(formula = age ~ proportion.black, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5449 -1.1117 -0.5285  0.9635  4.3421
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.8790     0.5688   1.545    0.133
## proportion.black  10.6471     1.5095   7.053 7.68e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.669 on 30 degrees of freedom
## Multiple R-squared:  0.6238, Adjusted R-squared:  0.6113
## F-statistic: 49.75 on 1 and 30 DF,  p-value: 7.677e-08
```

```r
coefficients(lm1)
```

```
##      (Intercept) proportion.black 
##        0.8790062       10.6471194
```

```r
confint(lm1)
```

```
##                       2.5 %    97.5 %
## (Intercept)      -0.2826733  2.040686
## proportion.black  7.5643082 13.729931
```

```r
anova(lm1)
```

```
## Analysis of Variance Table
## 
## Response: age
##                  Df  Sum Sq Mean Sq F value    Pr(>F)    
## proportion.black  1 138.544 138.544  49.751 7.677e-08 ***
## Residuals        30  83.543   2.785                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In the summary output, you should see first, the formula "call" to remind you what model you fit. Below this is a quantile list of your residuals. Typically you expect your residuals to have a mean or median close to zero and to be roughly symmetrical, thus absolute values of min and max should not be extremely different.

Then you see a coefficients table, showing the regression coefficients, standard error, t-value statistics (testing whether the estimate is <> 0; i.e. this is a two-tailed t-test for if x = 0). We should be most interested in this table.

In the first row labelled (Intercept), the value under "Estimate" is the estimated constant value in the equation Y = a + b X (the intercept). The standard error for this value are given. Do not worry about the t-statistic and significance in this row, as we are not interested in the intercept, but rather the slope.

The next row labelled "proportion.black" contains the estimated value for the slope in the "Estimate". The standard error of the slope is also given.

The "proportion.black" row also contains the results of a hypothesis test using the t-test approach.

The t-statistic for the slope is calculated from the formula:

$$t = \frac{b - \beta_0}{SE_b}$$

where b is the estimate of the slope in the sample, β_0 is the null hypothesized value of the slope (usually zero) and SE_b is the standard error of the slope in the sample. Since we are only working with two variables, the P-values from both the ANOVA approach and t-test approach should be identical, and the square of the t-statistic should equal the F statistic reported (or very close to).

Beneath the coefficients table are the significance codes, to help categorise the level of significance.

The last 3 lines depict the model residual standard error, the R-squared values, and the F statistics on the overall regression. The R-squared value should be the same as that obtained from the cor.test(), except squared. Note that both positive and negative correlations will produce a positive R2 value, which allows them to be compared on the same scale. We will not use adjusted R2values, so you can ignore this information for now.

The ANOVA table contains the results of a hypothesis test using the ANOVA approach. The null hypothesis being tested is that the slope between the two variables is zero (no relationship). The P-value of this test is found under "Pr(>F)" (i.e. Probability the F ratio is greater than a critical F ratio given the degrees of freedom 1,30). If this value is less than 0.05 then there is evidence for a linear relationship between the two numerical variables.

Now let's examine the predictions and residuals:

```
fits <- fitted(lm1)
resids <- residuals(lm1)
```

We have created new values called fits and resids that are derived from the fitted model. These are also available from the lm1 object:

```
head(lm1$fitted.values)
```

```
##        1        2        3        4        5        6 
## 3.114901 2.369603 2.050189 2.263132 2.156661 2.263132 
```

```
head(lm1$residuals)
```

```
##           1           2           3           4           5           6 
## -2.01490129 -0.86960293 -0.15018934 -0.06313173  0.44333946  0.93686827 
```

```
d$age[1]
```

```
## [1] 1.1
```

```
d$proportion.black[1]
```

```
## [1] 0.21
```

```
fits[1]
```

```
##        1 
## 3.114901 
```

```
resids[1]
```

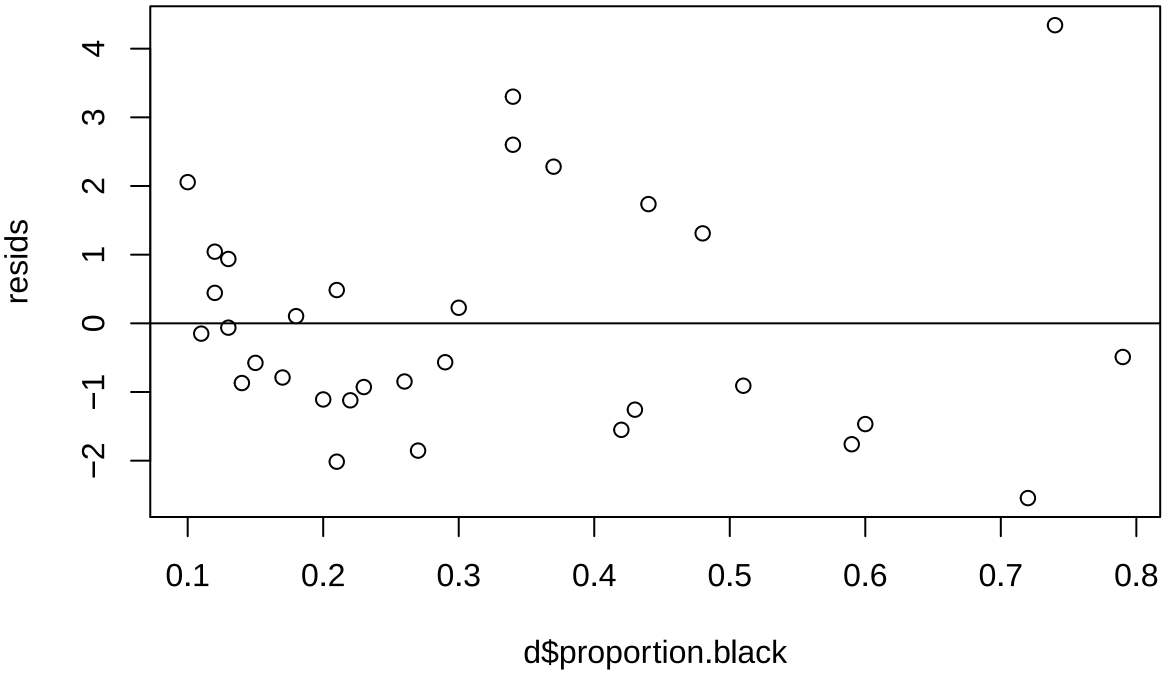
```
##         1 
## -2.014901 
```

For example, the first lion has an age of 1.1, and a proportion black of 0.21. According to the regression formula, a lion with 0.21 black on his nose would be predicted to be 3.11 years old.

The residuals are calculated by subtracting the predicted value of Y from the actual value of Y, for each data point. For example, for the first lion, the predicted age is 3.11 and the actual age is 1.1. Therefore the residual is -2.015.

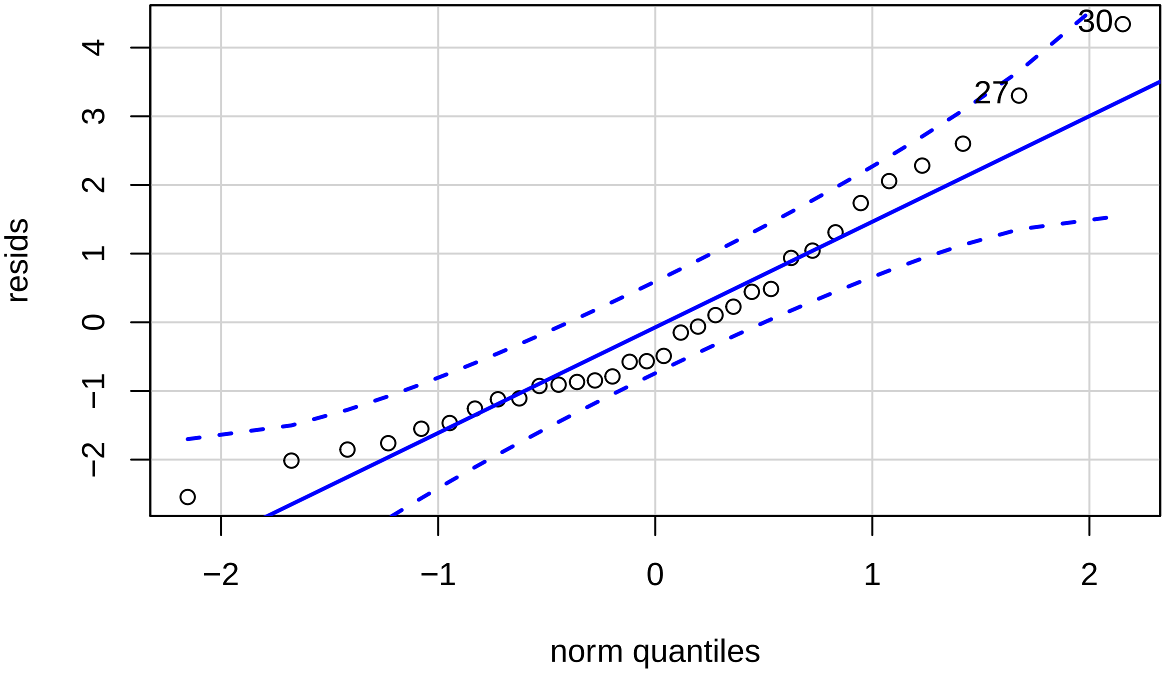
The residuals data can be used to construct a residuals plot, which lets us detect non-normality and unequal variance. Construct this graph now by plotting proportion black (X axis) against residuals (Y axis). You can add a horizontal line at zero, using the abline(h=0) function. In this case there are fewer points to the right of the graph (high proportion black), which may make the spread of points look different from left to right, but it is not significant enough for us to conclude that the distribution of Y values at each value of X is not normal or has unequal variance.

```
plot(resids ~ d$proportion.black) + abline(h = 0)
```

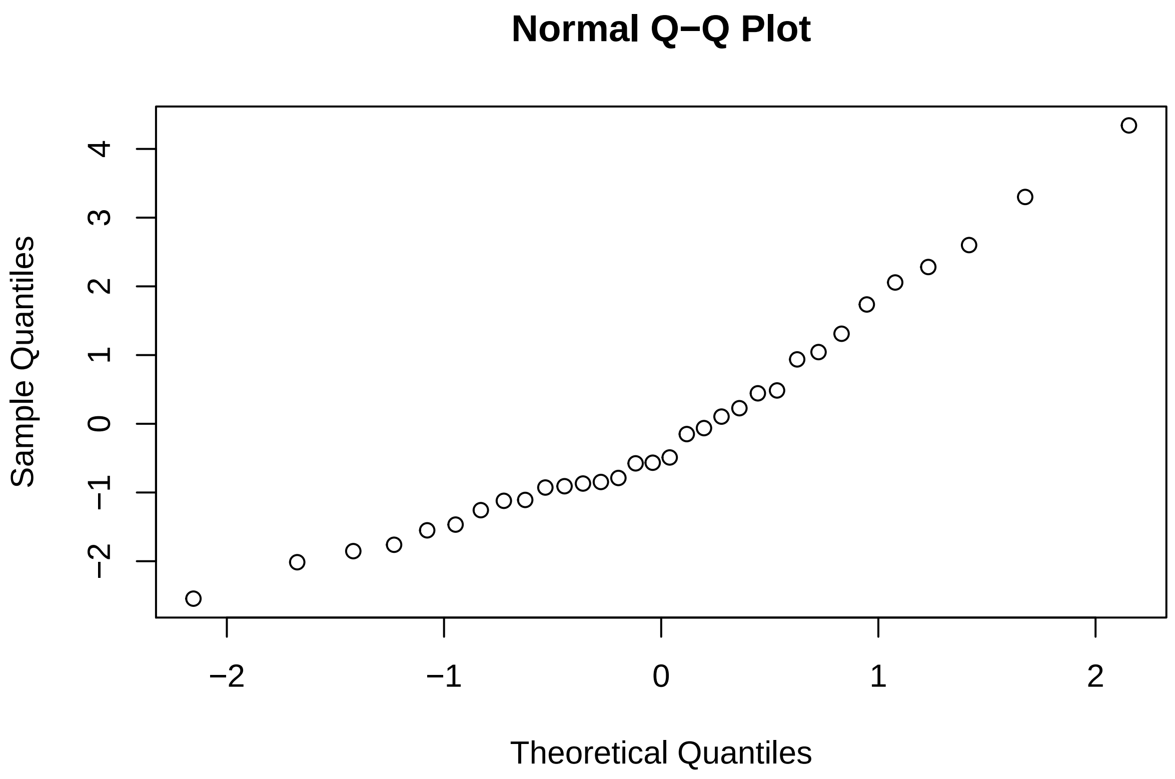


```
## integer(0)
```

Another way to evaluate the assumption that the residuals are normally distributed is to analyze the residuals data with a normality test and histogram. Try this now, using the qqp or qqnorm function:

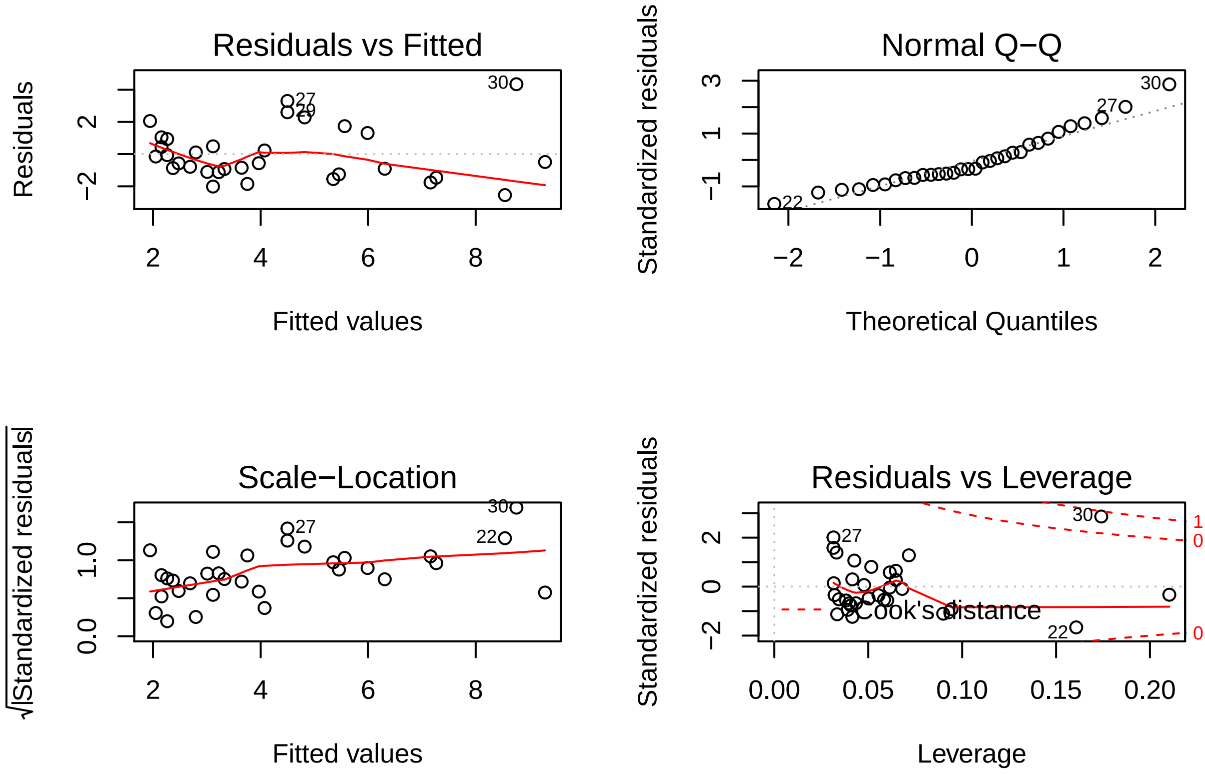


```
## [1] 30 27
```



One final, powerful shortcut when working with linear models is that you can simply plot the object, which will by default display typical diagnostic residual plots:

```
oldpar <- par()
par(mfrow = c(2, 2))
plot(lm1)
```



```r
par <- oldpar
```

With experience, you will learn to rely on visualisations to diagnose healthy and unhealthy residual plots.

Answer question 15 and 16 on Sakai

Brain size

Some nonlinear relationships can be made linear with a suitable transformation. One of the most common transformations is the log transformation. The data in the file mammals.csv contains species name, body mass (in kg) and brain size (in g) of 62 different mammal species. Open this data and construct a scatter plot to visualize brain size (on the Y-axis) compared to body mass. You should recognize immediately that this scatter of points does not look like a normal linear relationship. Transform each variable using the natural log transformation, as you learned last week. Re-plot the data using the log transformed variables. If this improved the linear relationship between the two variables conduct a linear regression analysis on the log-transformed variables.

```r
d <- read.csv("mammals.csv")
str(d)
```

```
## 'data.frame':    62 obs. of  3 variables:
##  $ name      : Factor w/ 62 levels "African elephant
##  $ bodymasskg: num  3.38 0.48 1.35 465 36.33 ...
##  $ brainmassg: num  44.5 15.5 8.1 423 119.5 ...
```

Answer question 17 on Sakai

Telomeres

The ends of chromosomes are called telomeres. As individuals age, their telomeres shorten, and there is evidence that shortened telomeres may play a role in aging. Telomeres can be lengthened in germ cells and stem cells by an enzyme called telomerase, but this is not expressed in most healthy somatic cells. (Cancer cells, on the other hand, usually express telomerase.) A set of data collected by Nordfjäll et al. (2005) examined whether there is a relationship between telomere length of fathers and their children. Examine the data in telomeres.csv by regression to determine whether offspring telomere length can be predicted from father telomere length.

```
d <- read.csv("telomeres.csv")
str(d)
```

```
## 'data.frame':    40 obs. of  2 variables:
##  $ father.telomere.length   : num  0.281 0.282 0.301 0.425 0.435 0.463 0.482 0.487 0.49 0.504 ...
##  $ offspring.telomere.length: num  0.41 0.582 0.311 0.574 0.811 0.592 0.846 0.424 0.756 0.924 ...
```

Answer question 18 and 19 on Sakai

Stress and mental health

Open the file symptoms and stress.csv. This file contains information on stress levels and severity of mental health symptoms in 106 patients. The authors are interested in whether the severity of mental health symptoms can be predicted from stress levels.

```
d <- read.csv("symptoms and stress.csv")
str(d)
```

```
## 'data.frame':    104 obs. of  2 variables:
##  $ stress  : int  1 1 2 3 3 3 4 5 5 6 ...
##  $ symptoms: int  65 68 61 61 100 100 64 58 62 78 ...
```

Answer question 20 on Sakai.