Week4Quiz

Exploratory Data Analysis 1

Histogram Assessment

QUESTION 1.1

Given the above histogram: how many people are between the ages of 35 and 45?

Answer is 9

QQ-plot Assessment

Download this RData file to your working directory: link. Then load the data into R with the following command:

```
load("skew.RData")
```

You should have a 1000 x 9 dimensional matrix 'dat':

```
dim(dat)
```

```
## [1] 1000    9
```

Using QQ-plots, compare the distribution of each column of the matrix to a normal. That is, use qqnorm() on each column. To accomplish this quickly, you can use the following line of code to set up a grid for 3x3=9 plots. ("mfrow" means we want a multifigure grid filled in row-by-row. Another choice is mfcol.)

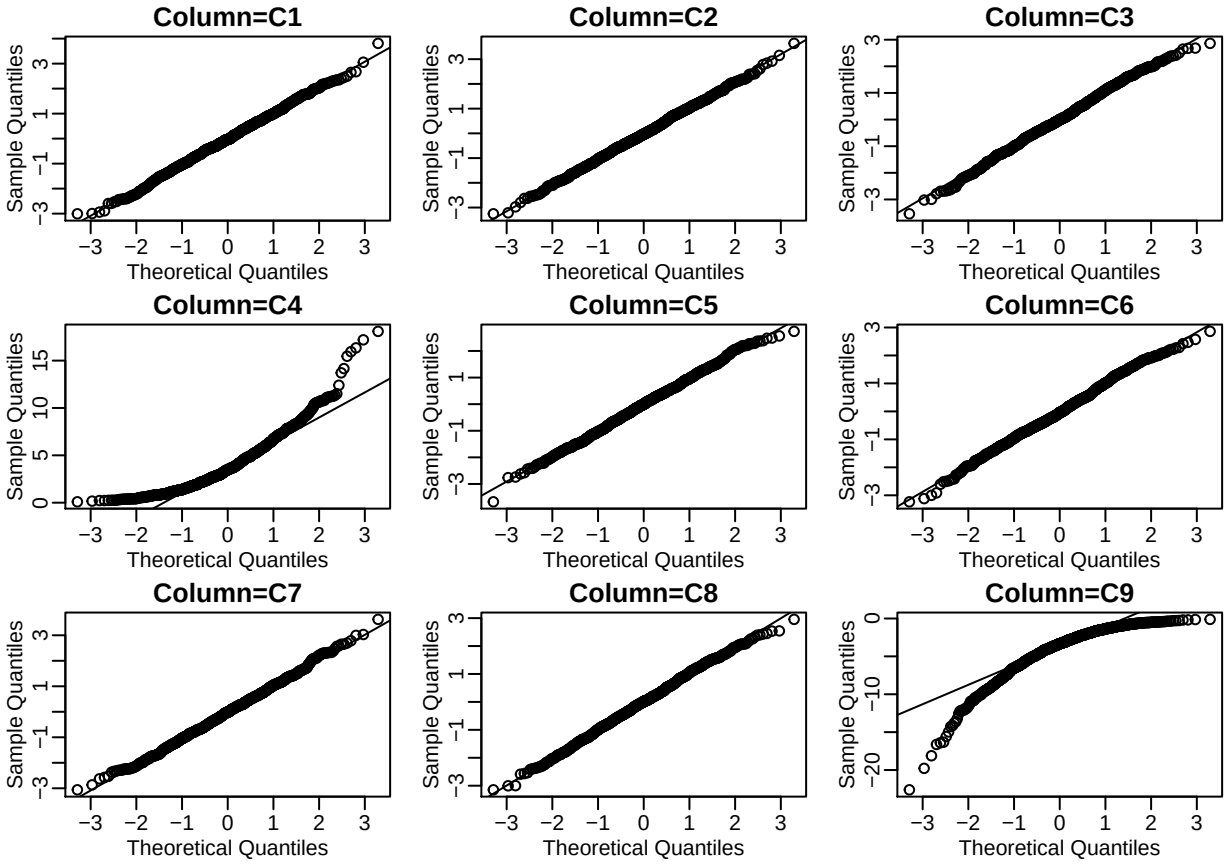
```
library(rafalib)
```

```
## Loading required package: RColorBrewer
```

```
mypar2(3,3)
```

Then you can use a for loop, to loop through the columns, and display one qqnorm() plot at a time. You should replace the text between ** with your own code.

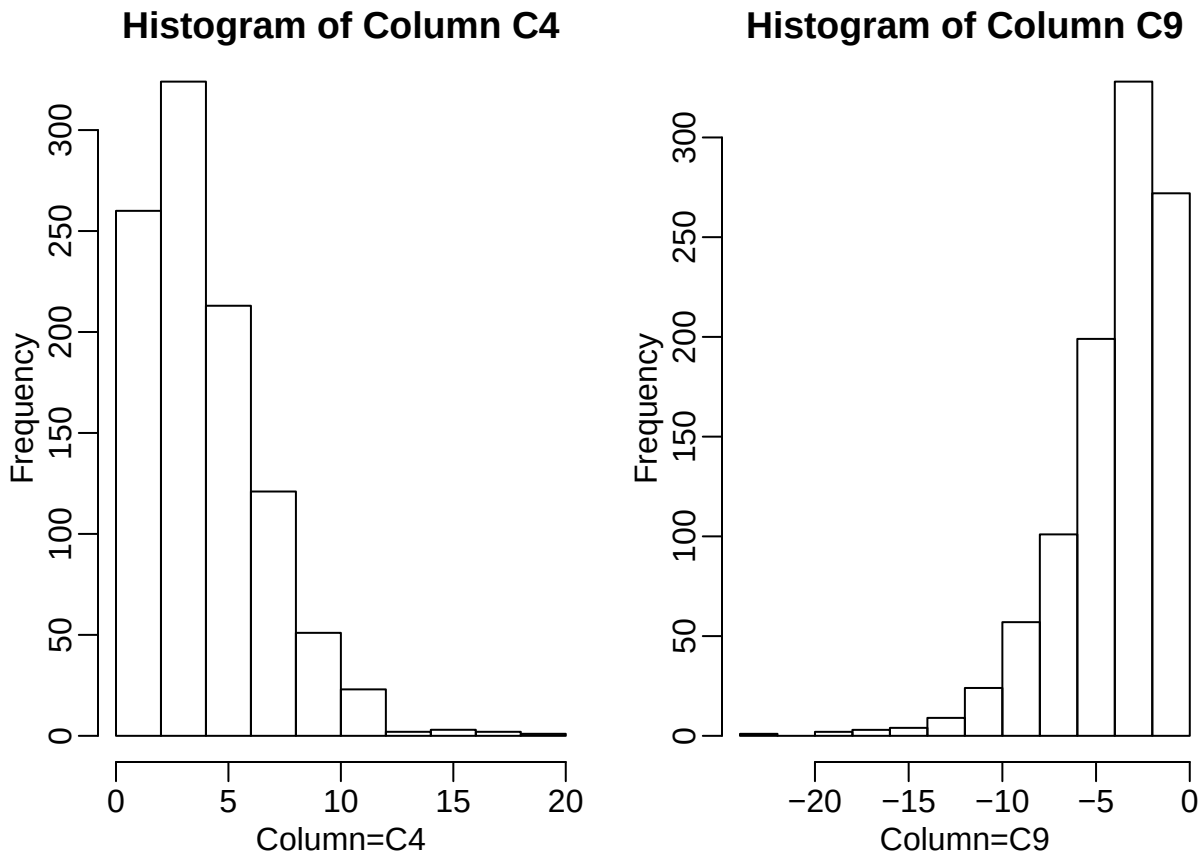
```
mypar2(3,3)
colnames(dat)<-c("C1","C2","C3","C4","C5","C6","C7","C8","C9")
for (i in 1:9) {
qqnorm(dat[,i],main=paste0("Column=",colnames(dat)[i]))
qqline(dat[,i])
}
```



Identify the two columns which are skewed.

Examine each of these two columns using a histogram. Note which column has "positive skew", in other words the histogram shows a long tail to the right (toward larger values). Note which column has "negative skew", that is, a long tail to the left (toward smaller values). Note that positive skew looks like an up-shaping curve in a qqnorm() plot, while negative skew looks like a down-shaping curve.

```
library(rafalib)
mypar2(1,2)
hist(dat[,4],xlab=paste0("Column=",colnames(dat)[4]),
     main=paste0("Histogram of Column ",colnames(dat)[4]))
hist(dat[,9],xlab=paste0("Column=",colnames(dat)[9]),
     main=paste0("Histogram of Column ",colnames(dat)[9]))
```



You can use the following line to reset your graph to just show one at a time:

```
library(rafalib)
mypar2(1,1)
```

QUESTION 2.1

Which column has positive skew (a long tail to the right)?

Answer is 4

QUESTION 2.2

Which column has negative skew (a long tail to the left)?

Answer is 9

Boxplot Assessment

The InsectSprays data set measures the counts of insects in agricultural experimental units treated with different insecticides. This dataset is included in R, and you can examine it by typing:

```
head(InsectSprays)
```

```
##   count spray
## 1    10     A
## 2     7     A
## 3    20     A
## 4    14     A
## 5    14     A
## 6    12     A
```

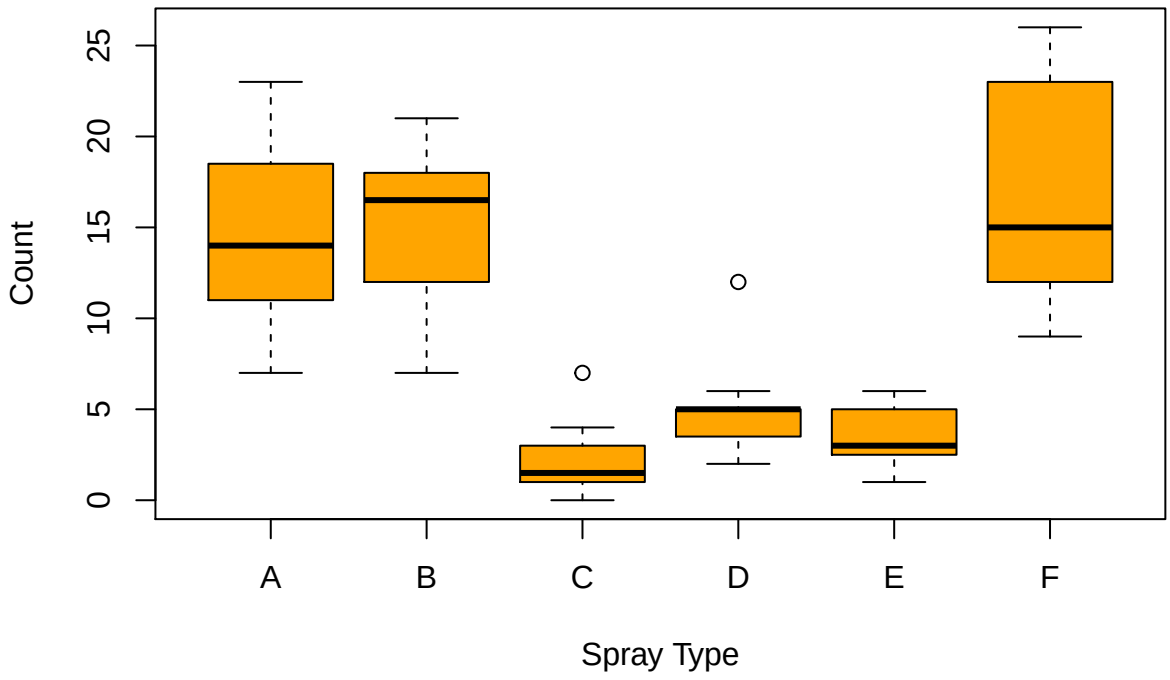
Try out two equivalent ways of drawing boxplots in R, using the InsectSprays dataset. Below is pseudocode, which you should modify to work with the InsectSprays dataset.

1) using split:

```
s<-split(InsectSprays, InsectSprays$spray)
```

2) using a formula:

```
boxplot(InsectSprays$count ~ InsectSprays$spray,
        xlab="Spray Type",
        ylab="Count",
        col = "orange")
```



QUESTION 3.1

Which spray seems the most effective (has the lowest median)?

```
median(s$C$count)
```

```
## [1] 1.5
```

Exploratory Data Analysis 2

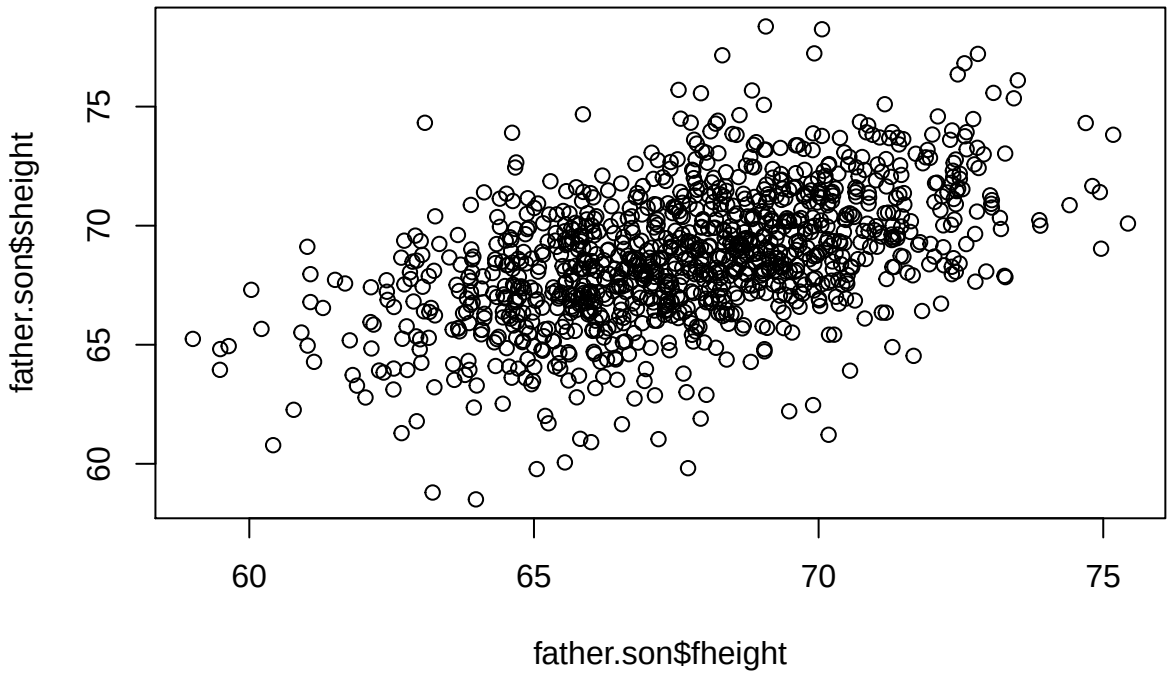
Scatter Plot

Load the father's and son's heights data by installing the UsingR package:

```
library(UsingR)
data(father.son)
```

We can make the scatterplot which was made in the previous video:

```
plot(father.son$fheight, father.son$sheight)
```



And calculate the correlation between these two vectors:

```
cor(father.son$fheight, father.son$sheight)
```

```
## [1] 0.5013383
```

A useful trick for scatterplots in R is the identify() function. This let's you click on points in the plot, and then once you hit the ESC key, the row number of the point(s) you clicked will be printed in the R console and on the figure:

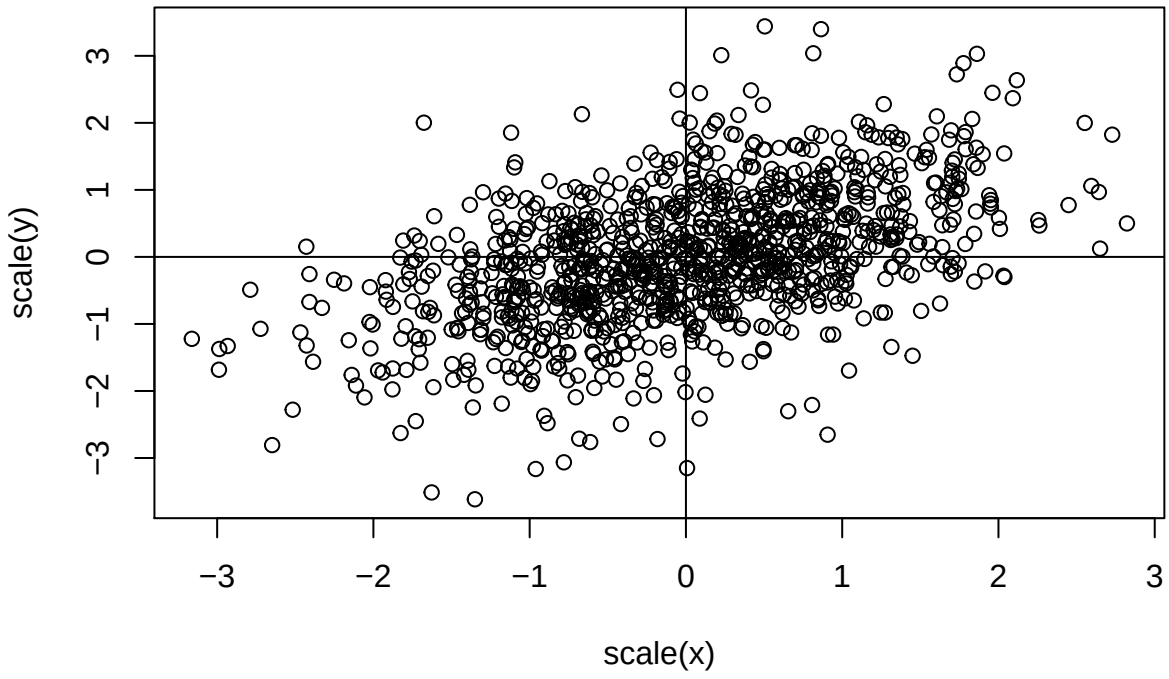
```
identify(father.son$fheight, father.son$sheight)
```

We saw that the correlation is related to the scaled vectors seen in the scatterplot. Let's try this with the father's and son's heights. Set the following variables (to save yourself keystrokes):

```
library(UsingR)
x = father.son$fheight
y = father.son$sheight
n = nrow(father.son)
```

The scale() function subtracts the mean and divides by the standard deviation. Make a scatterplot of scale(x) and scale(y) and add horizontal and vertical lines.

```
plot(scale(x), scale(y))
abline(h=0, v=0)
```



Now consider the number of points in the four quadrants of the figure. The correlation (or "Pearson correlation") is nearly the same as multiplying the scaled x values with the scaled y values, and taking the

average. If the points fall on a diagonal line pointing up and to the right, then the points are mostly in the +/+ quadrant and the -/- quadrant. So both of these quadrants contribute positive values to the average. If the points falls on a diagonal line to down and to the right, then we have mostly +/- and -/+, contributing negative values to the average.

QUESTION 1.1

Calculate the average of (scaled x values times scaled y values)

```
mean(scale(x) * scale(y))
```

```
## [1] 0.5008732
```

Note that this value above is not exactly the same as:

```
cor(x,y)
```

```
## [1] 0.5013383
```

This is because the standard deviation has in its formula (n-1) while the Pearson correlation instead uses (n). Check for yourself:

```
sum(scale(x) * scale(y)) / (n - 1)
```

```
## [1] 0.5013383
```

EDA Assessment

Here we will use the plots we've learned about to explore a dataset: some stats on a random sample of runners from the New York City Marthon in 2002. This data set can be found in the UsingR package (used in the previous assessment). Load the library and then load the nym.2002 data set with the following lines:

```
library(UsingR)
data(nym.2002)
```

Examine the head() of the data in nym.2002.

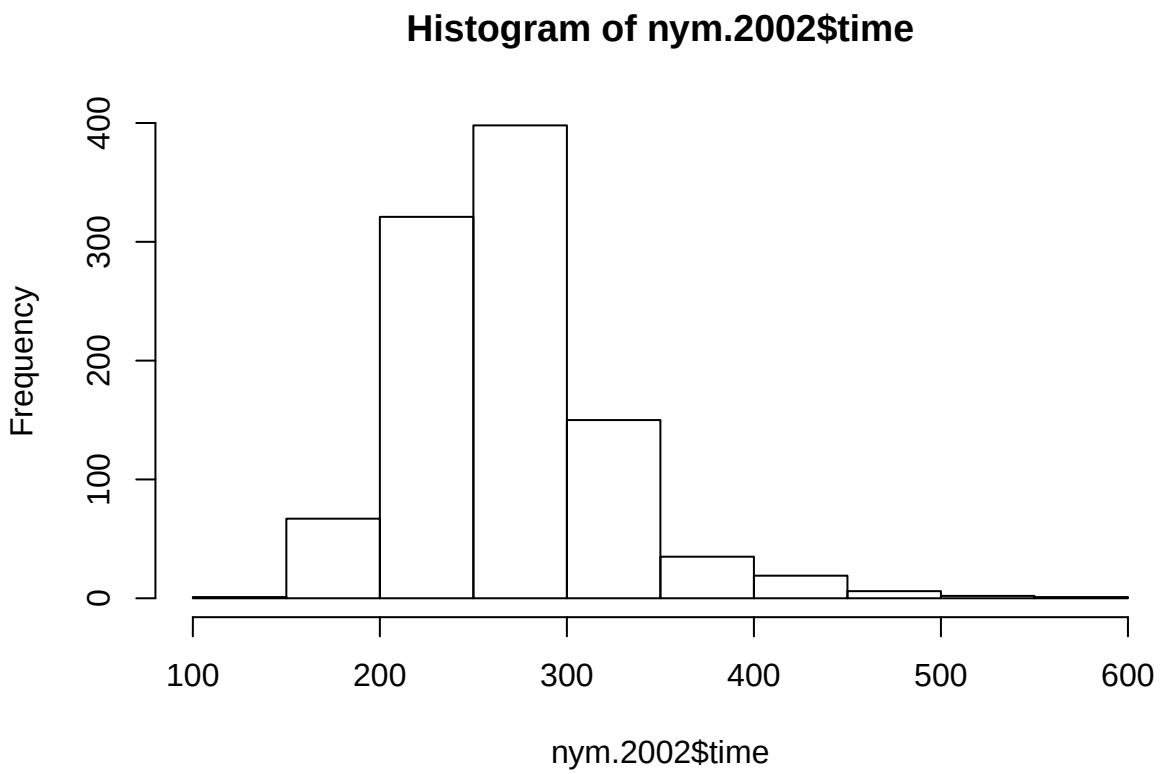
```
head(nym.2002)
```

```
##       place gender age home     time
## 3475   3592   Male  52  GBR 217.4833
## 13594 13853 Female  40   NY 272.5500
## 12012 12256   Male  31  FRA 265.2833
## 10236 10457 Female  33   MI 256.1500
## 9476   9686   Male  33   NY 252.2500
## 1720   1784   Male  40   NJ 201.9667
```

Try the following plots:

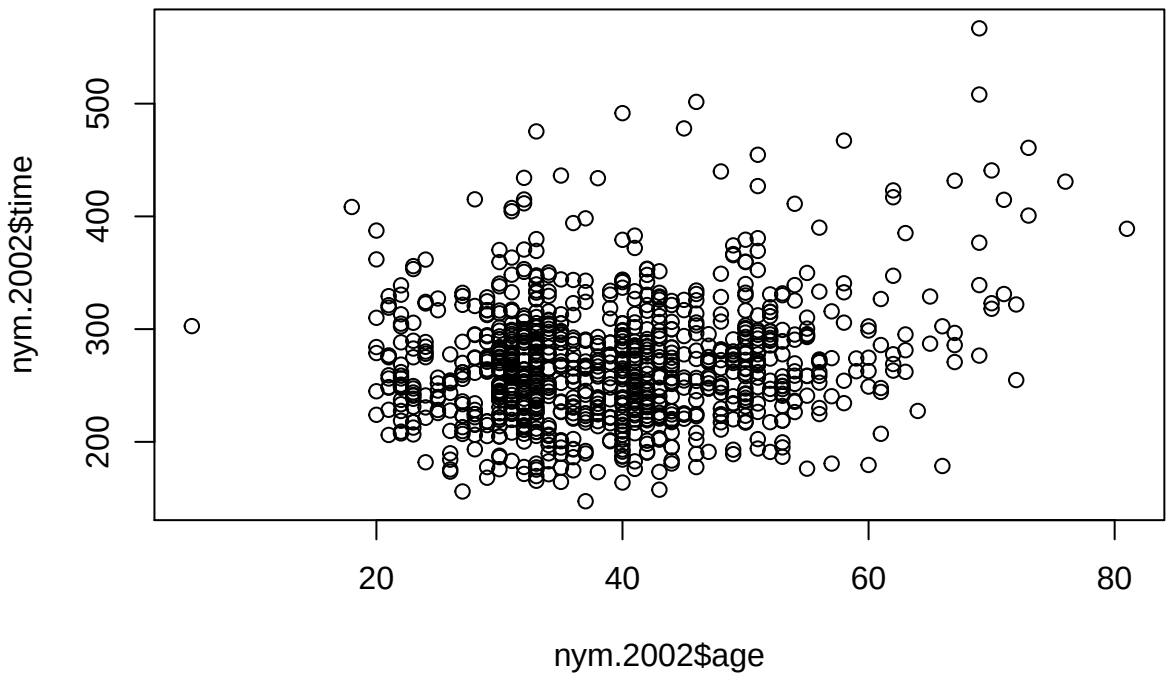
- histogram of times
- plot of runner's age vs their time
- plot of runner's time vs their place in the race
- qqnorm() of the runner's times. Are they normal? Positive or negative skew?
- a barplot of the most common location of origin. hint: tail(sort(table(nym.2002$home)),10)
- a boxplot of the runner's time over their gender
- histogram of times

```
hist(nym.2002$time)
```



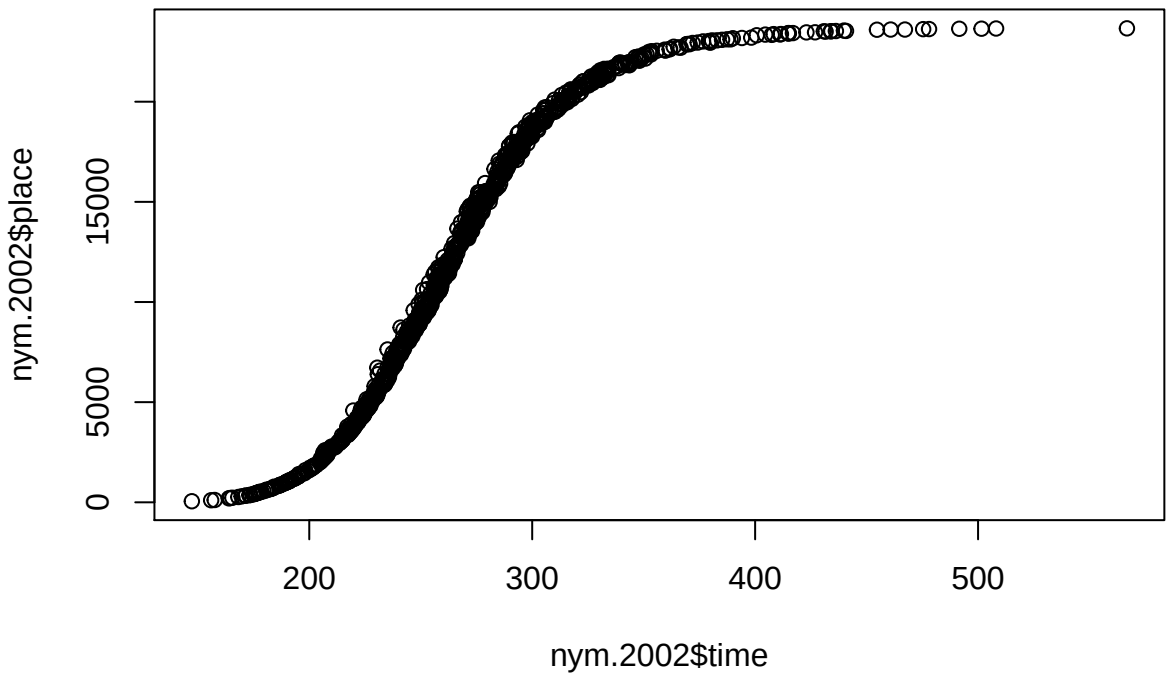
- plot of runner's age vs their time

```
plot(nym.2002$age,nym.2002$time)
```



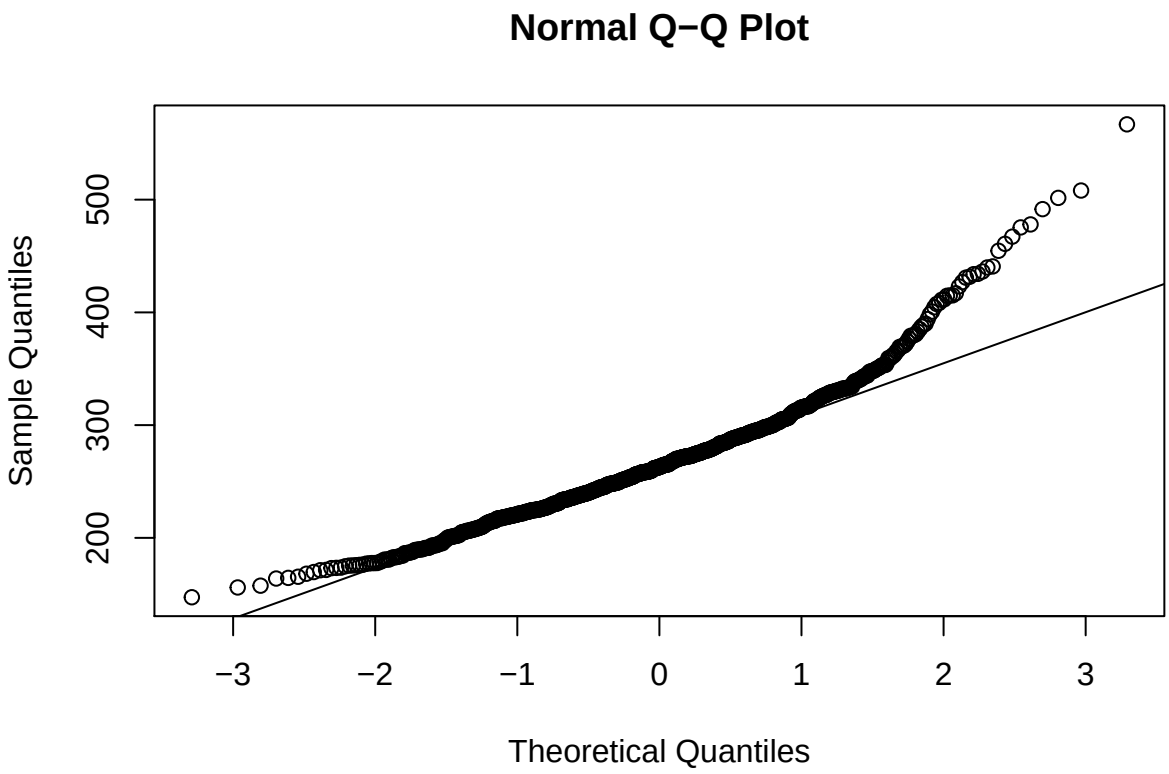
- plot of runner's time vs their place in the race

```
plot(nym.2002$time,nym.2002$place)
```



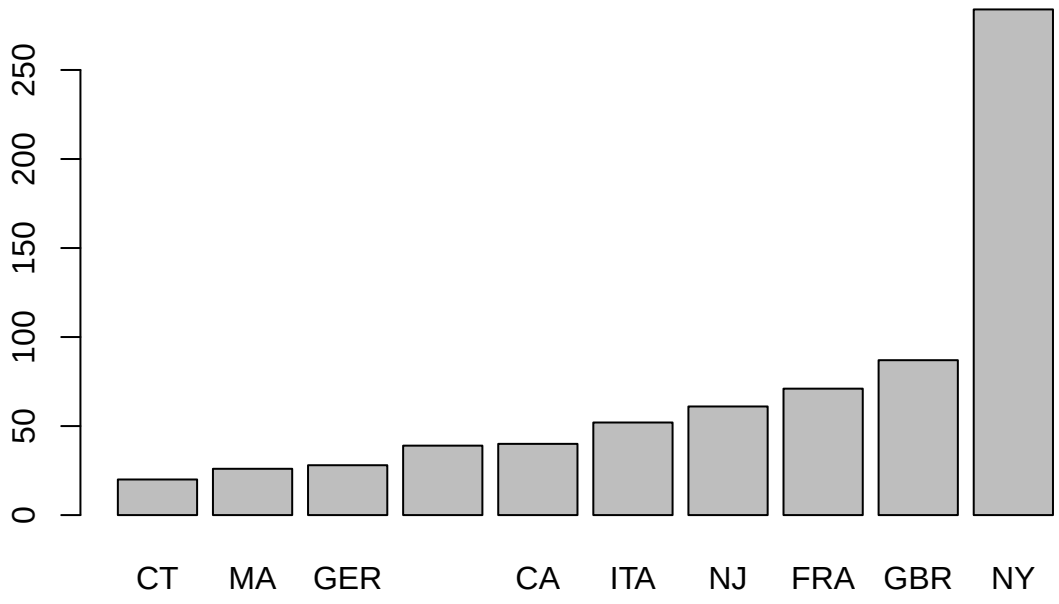
- qqnorm() of the runner's times. Are they normal? Positive or negative skew?

```
qqnorm(nym.2002$time)
qqline(nym.2002$time)
```



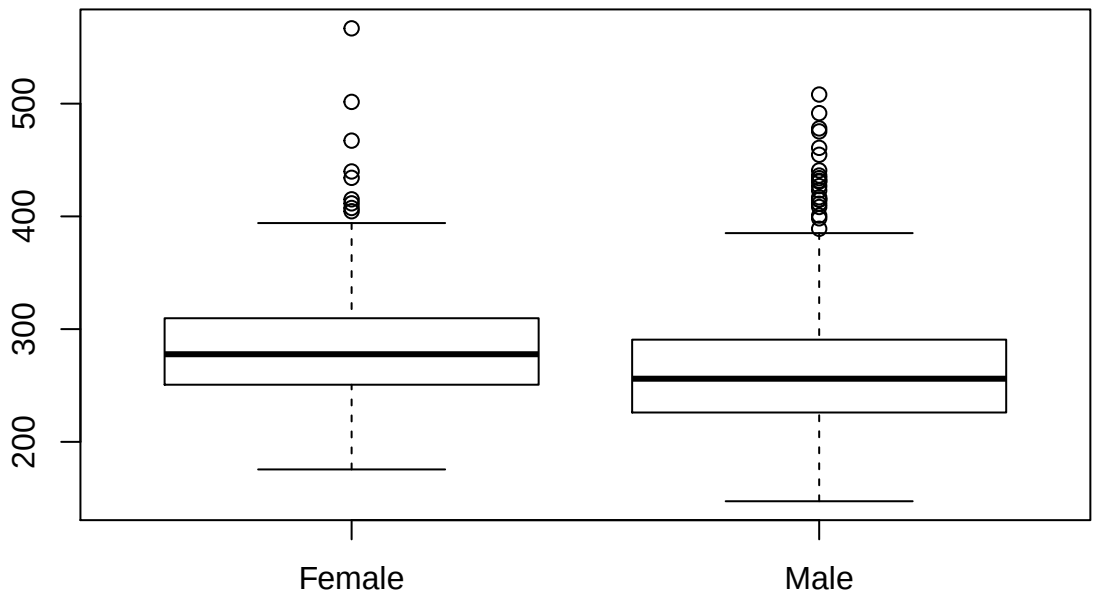
- a barplot of the most common location of origin. hint: tail(sort(table(nym.2002$home)),10)

```
barplot(tail(sort(table(nym.2002$home)),10))
```



- a boxplot of the runner's time over their gender

```
boxplot(nym.2002$time ~ nym.2002$gender)
```



In the previous video, we saw that multiplicative changes are symmetric around 0 when we are on the logarithmic scale. In other words, if we use the log scale, 1/2 times a number x, and 2 times a number x, are equally far away from x. We will explore this with the NYC marathon data.

Create a vector time of the sorted times:

```
time = sort(nym.2002$time)
```

QUESTION 2.1

What is the fastest time divided the median time?

```
min(time) / median(time)
```

```
## [1] 0.5605402
```

QUESTION 2.2

What is the slowest time divided the median time?

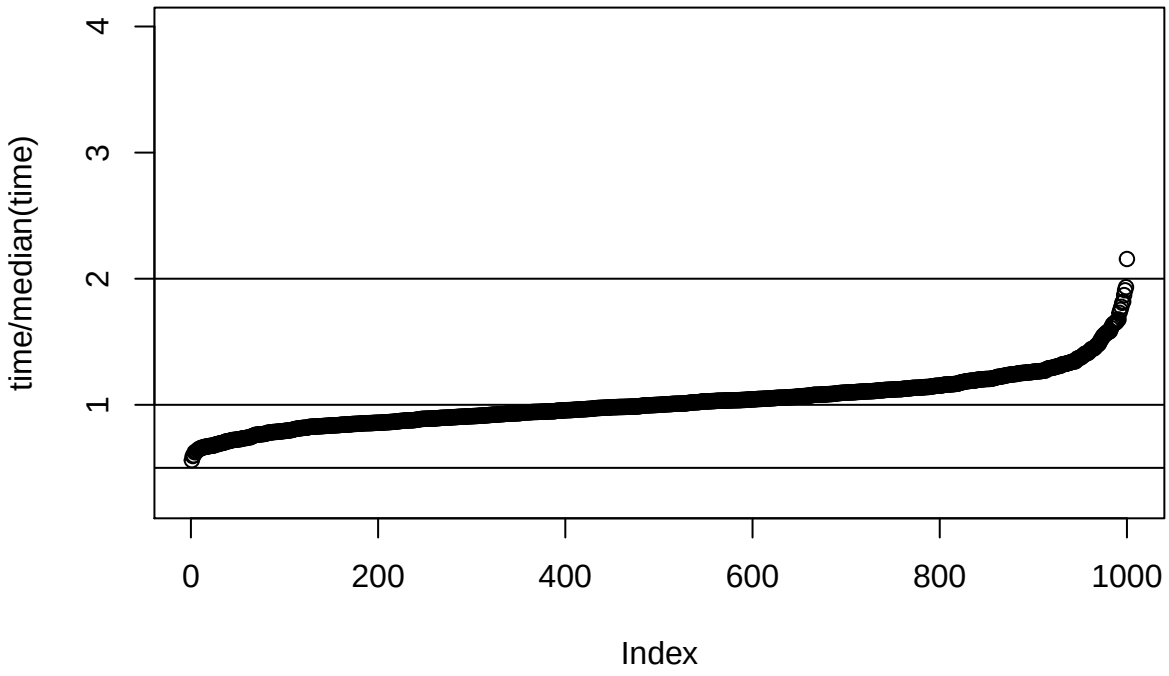
```
max(time) / median(time)
```

```
## [1] 2.156368
```

Compare the following two plots.

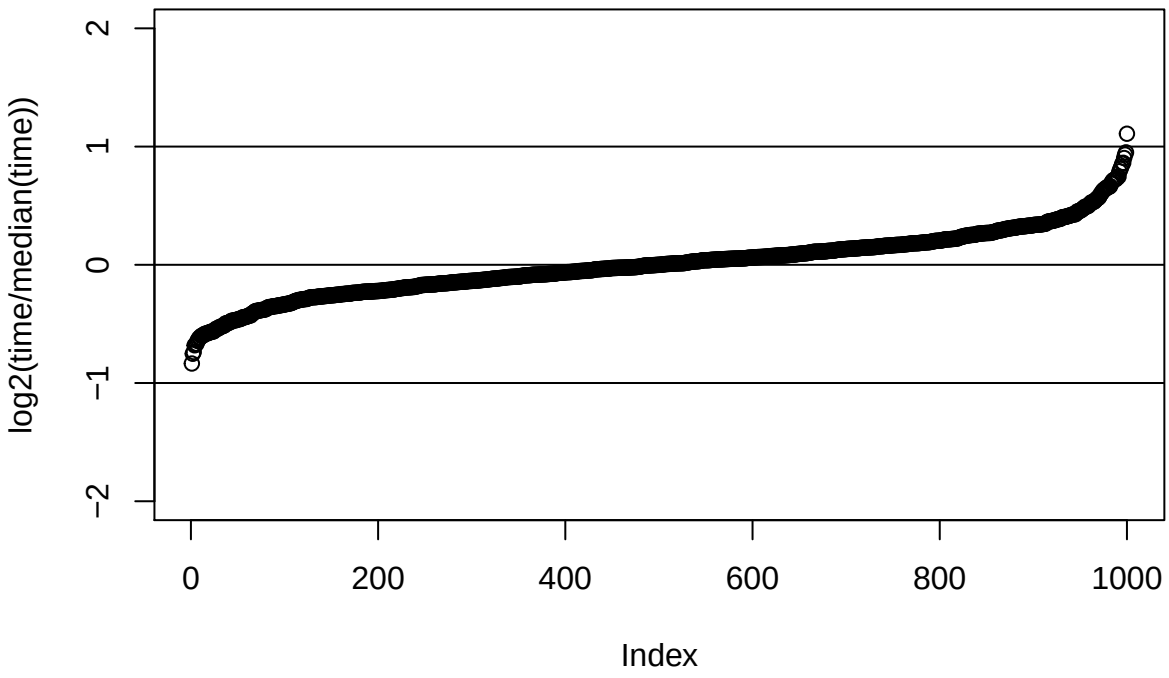
1) A plot of the ratio of times to the median time, with horizontal lines at twice as fast as the median time, and twice as slow as the median time.

```
plot(time/median(time), ylim=c(1/4,4))
abline(h=c(1/2,1,2))
```



2) A plot of the log2 ratio of times to the median time. The horizontal lines indicate the same as above: twice as fast and twice as slow.

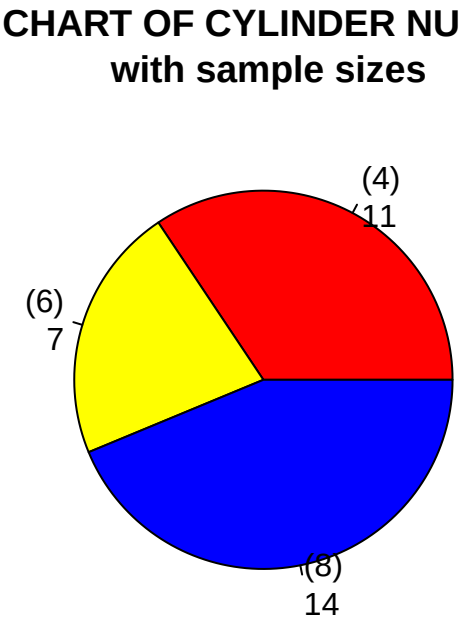
```
plot(log2(time/median(time)),ylim=c(-2,2))
abline(h=-1:1)
```



Pie Chart

A Pie chart is not recommended but can be created in R using pie(). An example is as follows.

```
cyltable<- table(mtcars$cyl)
labs<- paste("(",names(cyltable),")", "\n", cyltable, sep="")
pie(cyltable, labels = labs, col = c("red", "yellow", "blue"),
    main="PIE CHART OF CYLINDER NUMBERS\n with sample sizes")
```



QUESTION 3.1

When is it appropriate to use pie charts?

- When you are hungry
- To compare percentages
- To compare values that add up to 100%
- Never

Answer is Never

EXPLANATION Quoting from the R help page for the function pie: "Pie charts are a very bad way of displaying information. The eye is good at judging linear measures and bad at judging relative areas. A bar chart or dot chart is a preferable way of displaying this type of data."

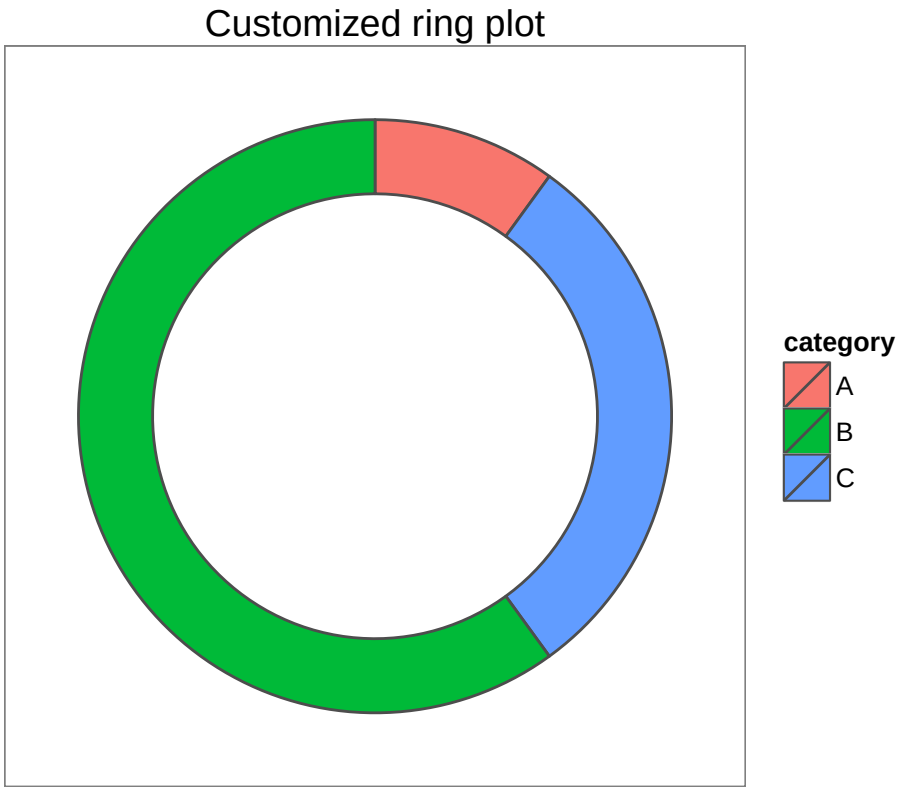
Donut plots

There is actually a plot that is less useful than a piechart. A donut plot is like a piechart but with the middle removed. An example

```
library(ggplot2)
# Create test data.
dat = data.frame(count=c(10, 60, 30), category=c("A", "B", "C"))
```

```
dat$fraction = dat$count / sum(dat$count)
dat = dat[order(dat$fraction), ]
dat$ymax = cumsum(dat$fraction)
dat$ymin = c(0, head(dat$ymax, n=-1))

p2 = ggplot(dat, aes(fill=category, ymax=ymax, ymin=ymin, xmax=4, xmin=3)) +
  geom_rect(colour="grey30") +
  coord_polar(theta="y") +
  xlim(c(0, 4)) +
  theme_bw() +
  theme(panel.grid=element_blank()) +
  theme(axis.text=element_blank()) +
  theme(axis.ticks=element_blank()) +
  labs(title="Customized ring plot")
p2
```



Donut plots are actually worse than pie charts. The reason is that by removing the center we remove one of the visual cues for determining the different areas: the anges. There is no reason to ever use a donut to display data.

QUESTION 3.2

The use of pseudo-3D plots in the literature mostly adds

- Pizzazz

- The ability to see three dimensional data
- Confusion

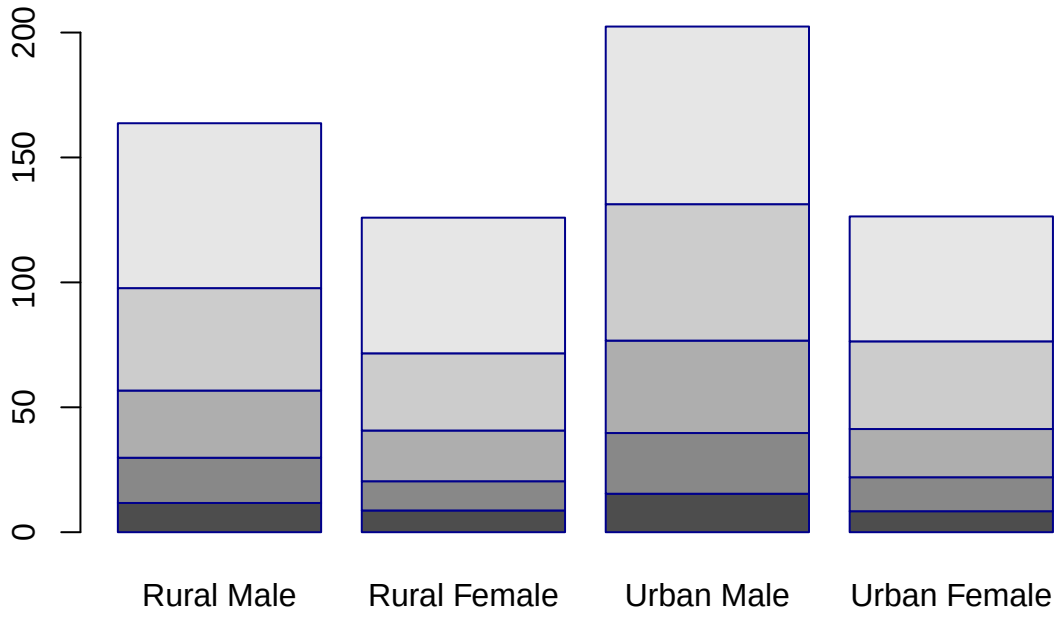
Answer is Confusion ####EXPLANATION

Humans have a hard time seeing 3 dimensions. It is even harder when it is a pseudo 3-D plot drawn on paper. Making two dimensional versions that display the same information is almost always possible.

Bar Plots

Bar Plots are used when we have to compare percentages that add up to 100%

```
barplot(VADeaths, border = "dark blue")
```



QUESTION 3.3

When is it appropirate to use a barplot

- To compare percentages that add up to 100%
- To display data from different groups: show the mean of each group
- To illustrate data that resulted in a p-value by adding an antenna at the top
- To summarize the relationship between two variables

Answer is To compare percentages that add up to 100%

EXPLANATION Barplots are much better than piecharts for displaying percentages. For the 2nd and 3rd answers we would use either boxplots or stripcharts while for the 4th we would use a scatter plot.

Introducing dplyr

dplyr Assessment

Read in the msleep dataset from CSV file

```
library(dplyr)
msleep <- read.csv("msleep_ggplot2.csv")
```

QUESTION 1.1

Using dplyr and the pipe command %>%, and perform the following steps:

Add a column of the proportion of REM sleep to total sleep

```
msleep %>%
  mutate(rem_proportion = sleep_rem / sleep_total) %>%
    head(3)
```

```
##              name      genus  vore     order conservation sleep_total
## 1         Cheetah   Acinonyx carni Carnivora           lc        12.1
## 2      Owl monkey      Aotus  omni  Primates         <NA>        17.0
## 3 Mountain beaver Aplodontia herbi  Rodentia           nt        14.4
##   sleep_rem sleep_cycle awake brainwt bodywt rem_proportion
## 1        NA          NA  11.9      NA  50.00             NA
## 2       1.8          NA   7.0  0.0155   0.48      0.1058824
## 3       2.4          NA   9.6      NA   1.35      0.1666667
```

Group the animals by their taxonomic order

```
msleep %>%
  group_by(order) %>%
    mutate(rem_proportion = sleep_rem / sleep_total) %>%
      head(3)
```

```
## Source: local data frame [3 x 12]
## Groups: order
##
##              name      genus  vore     order conservation sleep_total
## 1         Cheetah   Acinonyx carni Carnivora           lc        12.1
## 2      Owl monkey      Aotus  omni  Primates           NA        17.0
## 3 Mountain beaver Aplodontia herbi  Rodentia           nt        14.4
## Variables not shown: sleep_rem (dbl), sleep_cycle (dbl), awake (dbl),
##   brainwt (dbl), bodywt (dbl), rem_proportion (dbl)
```

Summarise by the median REM proportion

```
msleep %>%
  group_by(order) %>%
    mutate(rem_proportion = sleep_rem / sleep_total) %>%
      summarise(avg_sleep = median(rem_proportion) )  %>%
        head(3)
```

```
## Source: local data frame [3 x 2]
##
##          order avg_sleep
## 1 Afrosoricida 0.1474359
## 2 Artiodactyla        NA
## 3    Carnivora        NA
```

Arrange by the median REM proportion

```
msleep %>%
  group_by(order) %>%
    mutate(rem_proportion = sleep_rem / sleep_total) %>%
      summarise(med_sleep = median(rem_proportion) ) %>%
        arrange(med_sleep) %>%
          head(3)
```

```
## Source: local data frame [3 x 2]
##
##           order  med_sleep
## 1    Hyracoidea 0.09433962
## 2    Lagomorpha 0.10714286
## 3 Diprotodontia 0.13326100
```

Take the head() of this to see just the orders with smallest median REM proportion

What is the median REM proportion of the order with the smallest median REM proportion?

Answer

```
msleep %>%
  group_by(order) %>%
    mutate(rem_proportion = sleep_rem / sleep_total) %>%
      summarise(med_sleep = median(rem_proportion) ) %>%
        arrange(med_sleep) %>%
          head(1)
```

```
## Source: local data frame [1 x 2]
##
##        order  med_sleep
## 1 Hyracoidea 0.09433962
```

Robust Summaries

Median, MAD, And Spearman Assessment

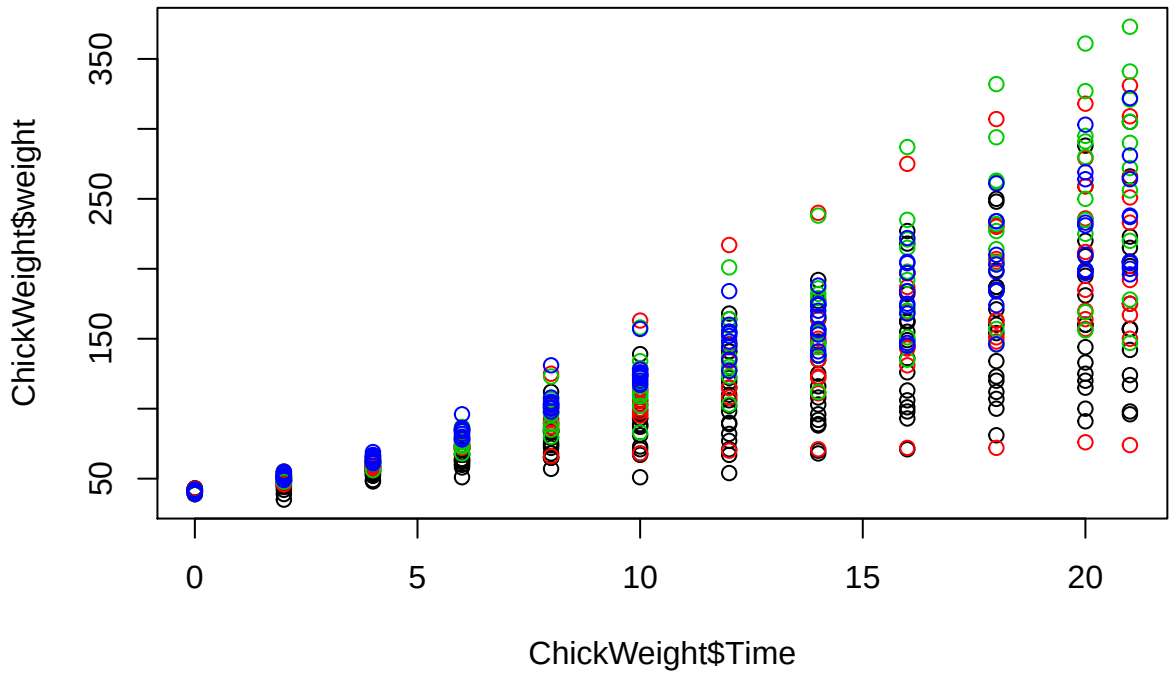
We're going to explore the properties of robust statistics, using a dataset of the weight of chicks in grams as they grow from day 0 to day 21. This dataset also splits up the chicks by different protein diets, which are coded from 1 to 4.

This dataset is built into R, and can be loaded with:

```
data(ChickWeight)
```

Just to start, take a look at the weights of all observations over time, and color this by the Diet:

```
plot(ChickWeight$Time, ChickWeight$weight, col=ChickWeight$Diet)
```



First, in order to easily compare weights at different time points across the different chicks, we will use the reshape() function in R to change the dataset from a "long" shape to a "wide" shape. Long data and wide data are useful for different purposes (for example, the plotting library ggplot2 and the manipulation library dplyr want to have data in the long format).

```
head(ChickWeight,3)
```

```
##   weight Time Chick Diet
## 1     42    0     1    1
## 2     51    2     1    1
## 3     59    4     1    1
```

```
chick = reshape(ChickWeight,idvar=c("Chick","Diet"),timevar="Time",direction="wide")
```

The meaning of this line is: reshape the data from long to wide, where the columns Chick and Diet are the ID's and the column Time indicates different observations for each ID. Now examine the head of this dataset:

```
head(chick,3)
```

```
##    Chick Diet weight.0 weight.2 weight.4 weight.6 weight.8 weight.10
## 1      1    1       42       51       59       64       76        93
## 13     2    1       40       49       58       72       84       103
## 25     3    1       43       39       55       67       84        99
##    weight.12 weight.14 weight.16 weight.18 weight.20 weight.21
## 1        106       125       149       171       199       205
## 13       122       138       162       187       209       215
## 25       115       138       163       187       198       202
```

The only remaining step is that we want to remove any chicks which have missing observations at any time points (NA for "not available") . The following line of code identifies these rows, and then removes them

```
chick = na.omit(chick)
```

QUESTION 2.1

We will focus on the chick weights on day 4 (check the column names of 'chick' and note the numbers). How much does the average of chick weights at day 4 increase if we add an outlier measurement of 3000 grams? Specifically what is the average weight of the day 4 chicks including the outlier chick divided by the average of the weight of the day 4 chicks without the outlier. Hint: use c() to add a number to a vector.

```
obserOutliner<-c(chick$weight.4,3000)
mean(obserOutliner)/mean(chick$weight.4)
```

```
## [1] 2.062407
```

QUESTION 2.2

In the problem above we saw how sensitive the mean is to outliers. Now let's see what happens when we use the median instead of the mean. Compute the same ratio but now using median instead of the mean. Specifically what is the median weight of the day 4 chick including the outlier chick divided by the median of the weight of the day 4 chicks without the outleir.

```
median(obserOutliner)/median(chick$weight.4)
```

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

QUESTION 2.3

Now try the same thing with the sample standard deviation, (the sd() function in R). Add a chick with weight 3000 grams to the chick weights from day 4. How much does the standard deviation change? What's the standard devition with the outlier chick divided by the standard deviation without the outlier chick?

```
sd(obserOutliner)/sd(chick$weight.4)
```

```
## [1] 101.2859
```

Compare this to the median absolute deviation in R, which is calculated with the mad() function. Note that the mad is unaffected by the addition of a single outlier. The mad() function in R includes the scaling factor 1.4826 which was mentioned in the video, such that mad(x) and sd(x) are very similar for a sample from a normal distribution.

```
mad(obserOutliner)/mad(chick$weight.4)
```

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

QUESTION 2.4

Our last question relates to how the Pearson correlation is affected by an outlier, and compare to the Spearman correlation. The Pearson correlation between x and y is given in R by cor(x,y). The Spearman correlation is given by

`cor(x,y,method="spearman").`

- Pearson correlation

```
cor(chick$weight.4,chick$weight.21)
```

```
## [1] 0.4159499
```

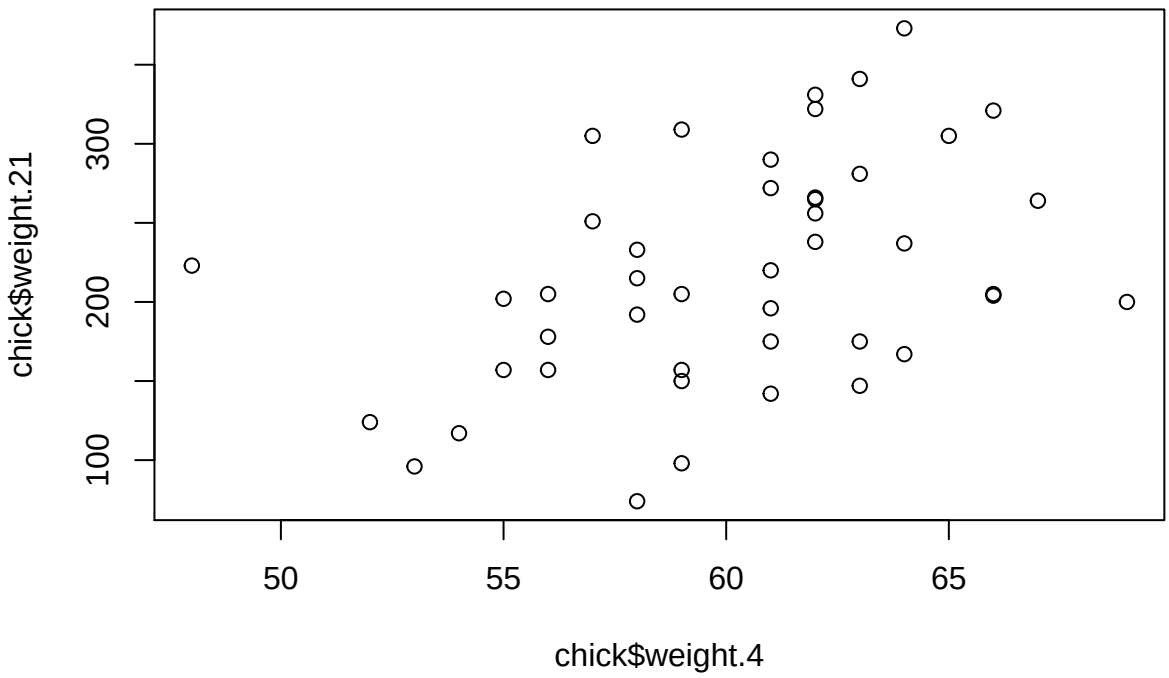
- Spearman correlation

```
cor(chick$weight.4,chick$weight.21,method="spearman")
```

```
## [1] 0.4303941
```

Plot the weights of chicks from day 4 and day 21.

```
plot(chick$weight.4,chick$weight.21)
```



We can see that there is some general trend, with the lower weight chicks on day 4 having low weight again on day 21, and likewise for the high weight chicks.

Calculate the Pearson correlation of the weights of chicks from day 4 and day 21. Now calculate how much the Pearson correlation changes if we add a chick that weighs 3000 on day4 and 3000 on day 21. Again, divide the Pearson correlation with the outlier chick over the Pearson correlation computed without the outliers.

```
# Pearson Correlation without outliers.
pWithout0<-cor(chick$weight.4,chick$weight.21)
# Pearson Correlation with outliers.
pWith0<-cor(c(chick$weight.4,3000),c(chick$weight.21,3000))
pWith0/pWithout0
```

```
## [1] 2.370719
```

Note that the Spearman correlation also changes with the addition of this outlier chick, but much less drastically: cor(x,y,method=“spearman”) compared to cor(c(x,3000),c(y,3000),method=“spearman”) with x and y the vectors of interest.

```
# Spearman Correlation without outliers.
sWithout0<-cor(chick$weight.4,chick$weight.21,method = "spearman")
# Spearman Correlation with outliers.
sWith0<-cor(c(chick$weight.4,3000),c(chick$weight.21,3000),method = "spearman")
sWith0/sWithout0
```

```
## [1] 1.084826
```

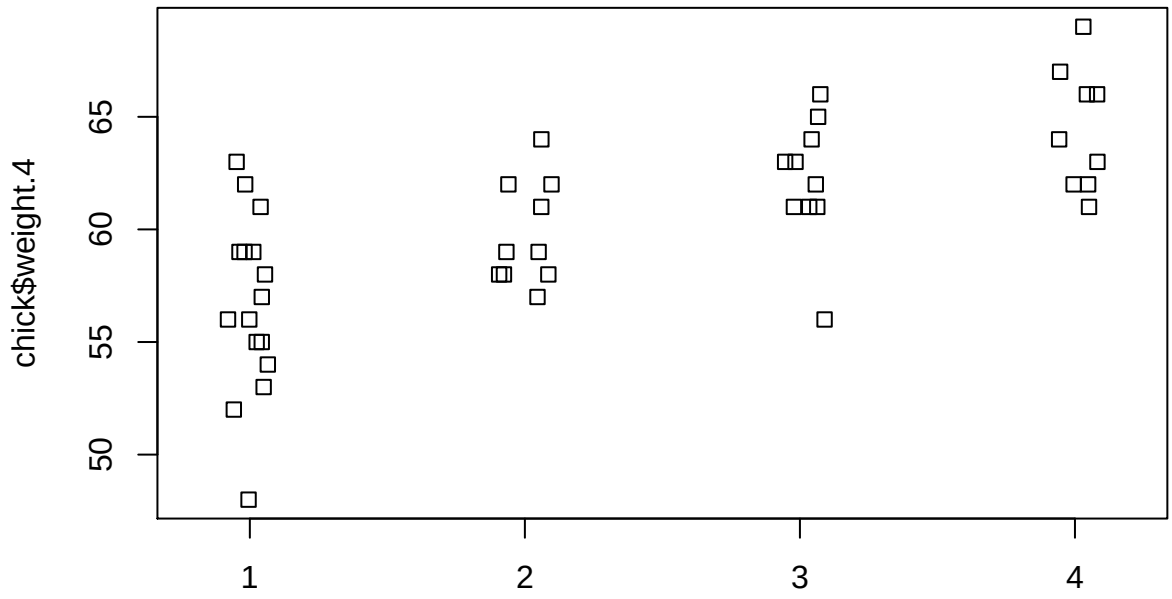
Mann-Whitney-Wilcoxon Test

We will continue using the chick weight dataset from the previous problem. As a reminder, run these lines of code in R to prepare the data:

```
data(ChickWeight)
chick <- reshape(ChickWeight,idvar=c("Chick","Diet"),timevar="Time",direction="wide")
chick <- na.omit(chick)
```

Make a strip chart with horizontal jitter of the chick weights from day 4 over the different diets:

```
stripchart(chick$weight.4 ~ chick$Diet, method="jitter", vertical=TRUE)
```



Suppose we want to know if diet 4 has a significant impact on chick weight over diet 1 by day 4. It certainly appears so, but we can use statistical tests to quantify the probability of seeing such a difference if the different diets had equal effect on chick weight.

QUESTION 2.1

Save the weights of the chicks on day 4 from diet 1 as a vector 'x'. Save the weights of the chicks on day 4 from diet 4 as a vector 'y'.

```
x<-subset(chick,select=weight.4,subset = Diet==1)
y<-subset(chick,select=weight.4,subset = Diet==4)
```

Now perform a t test comparing x and y (in R the function t.test(x,y) will perform the test).

```
t.test(x,y)
```

```
##
##  Welch Two Sample t-test
##
## data:  x and y
## t = -5.8393, df = 21.827, p-value = 7.32e-06
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -10.513134  -5.000755
## sample estimates:
## mean of x mean of y
##  56.68750  64.44444
```

Now, perform a Wilcoxon test of x and y (in R the function wilcox.test(x,y) will perform the test).

```
wilcox.test(x$weight.4,y$weight.4)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  x$weight.4 and y$weight.4
## W = 6, p-value = 0.0002012
## alternative hypothesis: true location shift is not equal to 0
```

Note that a warning will appear that an exact p-value cannot be calculated with ties (so an approximation is used, which is fine for our purposes).

Now, perform a t-test of x and y, after adding a single chick of weight 200 grams to 'x' (the diet 1 chicks). What is the p-value from this test? The p-value of a test is available with the following code: t.test(x,y)$p.value

```
t.test(c(x$weight.4,200),y$weight.4)$p.value
```

```
## [1] 0.9380347
```

QUESTION 2.2

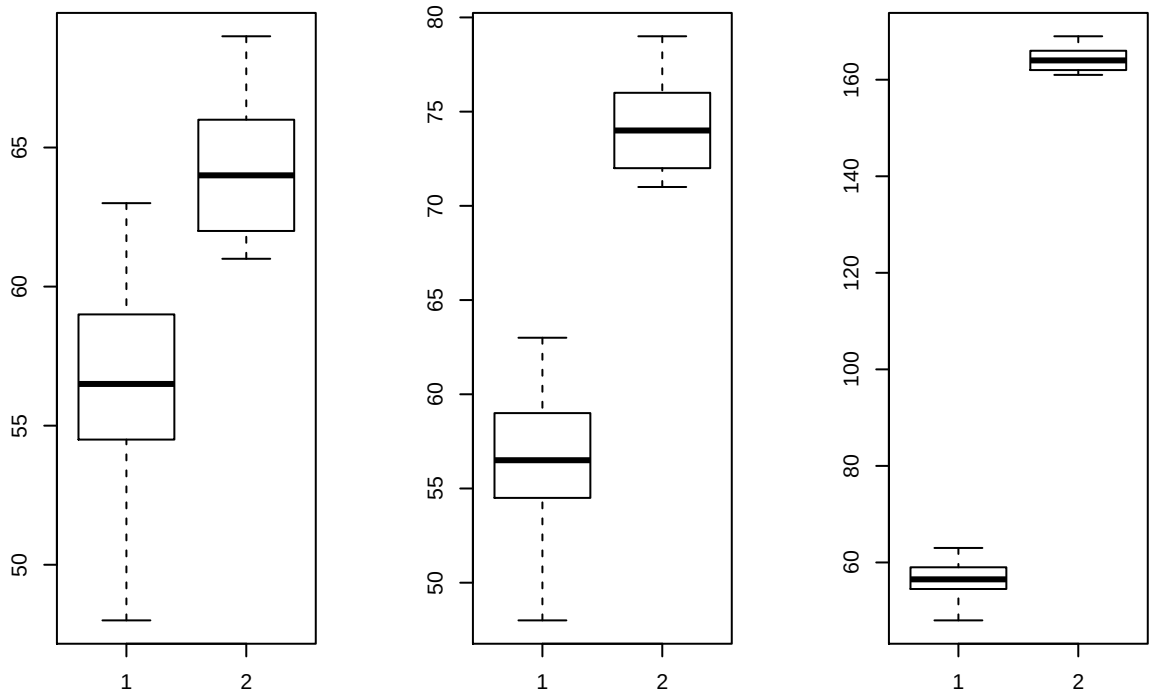
We will now investigate a possible downside to the Wilcoxon-Mann-Whitney test statistic. Using the following code to make three boxplots, showing the true Diet 1 vs 4 weights, and then two altered versions: one with an additional difference of 10 grams and one with an additional difference of 100 grams. (Use the x and y as defined above, NOT the ones with the added outlier.)

```
par(mfrow=c(1,3))

boxplot(x$weight.4,y$weight.4)

boxplot(x$weight.4,y$weight.4+10)

boxplot(x$weight.4,y$weight.4+100)
```



What is the difference in t-test statistic (the statistic is obtained by t.test(x,y)$statistic) between adding 10 and adding 100 to all the values in the group 'y'? So take the the t-test statistic with x and y+10 and substract away the t-test statistic with x and y+100. (The value should be positive).

```
t.test(x$weight.4, y$weight.4+10)$statistic - t.test(x$weight.4, y$weight.4+100)$statistic
```

```
##        t
## 67.75097
```

Now examine the Wilcoxon test statistic for x and y+10 and for x and y+100.

```
wilcox.test(x$weight.4, y$weight.4+10)$statistic - t.test(x$weight.4, y$weight.4+100)$statistic
```

```
##        W
## 81.11819
```

Because the Wilcoxon works on ranks, after the groups have complete separation (all points from group 'y' are above all points from group 'x'), the statistic will not change, regardless of how large the difference grows. Likewise, the p-value has a minimum value, regardless of how far apart the groups are. This means that the Wilcoxon test can be considered less powerful than the t-test in certain contexts, and with small significance levels (alpha). In fact for small sample sizes, the p-value can't be very small, even when the difference is very large. Compare:

```r
wilcox.test(c(1,2,3),c(4,5,6))
```

```
##
##  Wilcoxon rank sum test
##
## data:  c(1, 2, 3) and c(4, 5, 6)
## W = 0, p-value = 0.1
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(c(1,2,3),c(400,500,600))
```

```
##
##  Wilcoxon rank sum test
##
## data:  c(1, 2, 3) and c(400, 500, 600)
## W = 0, p-value = 0.1
## alternative hypothesis: true location shift is not equal to 0
```

This issue becomes important in Course 3 on Advanced Statistics, when we discuss correction for performing thousands of tests, as is done in high-throughput biological assays.
