# Session 2: Basics of R continued

# Overview

Statistics and plots for categorical data

Statistics and plots for quantitative data

Generating random data and probability functions

For loops

Writing functions

# Categorical data

# Categorical data

```
# Get information about drinking behavior
> drinking_vec <- profiles$drinks

# Create a table showing how often people drink
> drinks_table <- table(drinking_vec)
> drinks_table
```

# Relative frequency table

We can create a relative frequency table using the function:

```
> prop.table(my_table)
```

Can you create a relative frequency table for the drinking behavior of the people in the okcupid data set?

```
> drinks_table <- table(profiles$drinks)
> prop.table(drinks_table)
```

What is the proper statistical notation for these values: $\hat{p}$ or $\pi$ ?

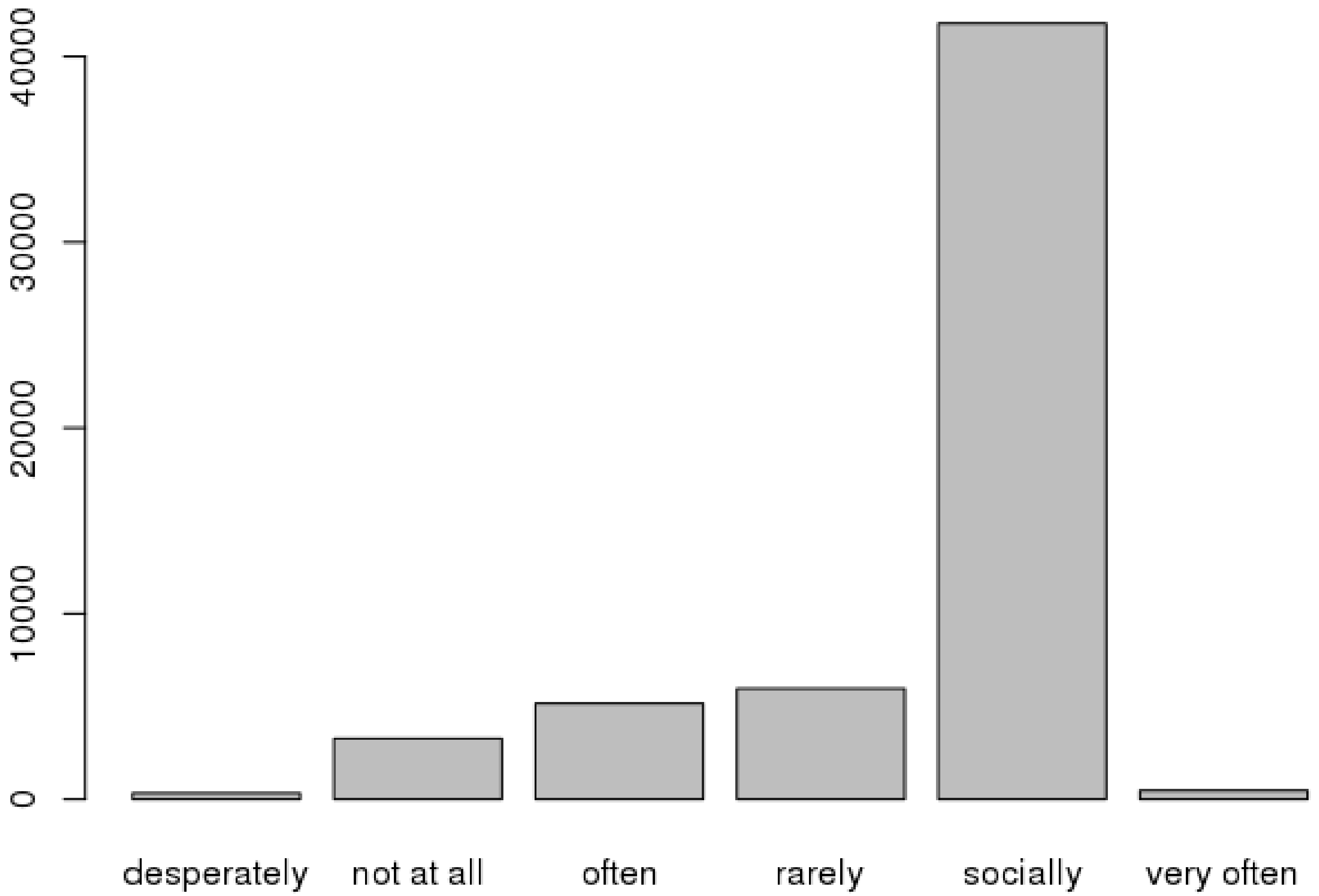
# Bar plots (pun intended?)

We can plot the number of items in each category using a bar plot

```
> barplot(my_table)
```

Can you create a bar plot for the drinking behavior of the people in the okcupid data set?

```
> drinks_table <- table(profiles$drinks)
> barplot(drinks_table)
```



What is wrong with this plot?

# Labeling plots

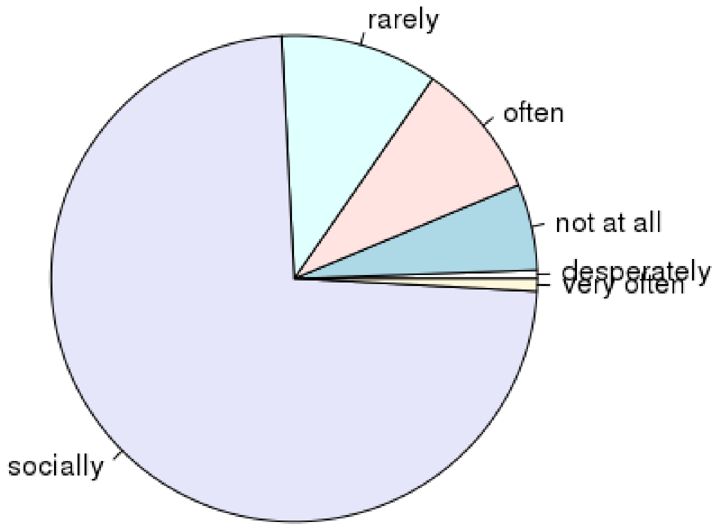
Can you figure out how to label the axes?

```
> barplot(drinks_table,
        ylab = "Count",
        xlab = "Type of drinker",
        main = "Counts of different types of drinkers")
```

# Pie charts

We can also use the pie() function to create pie charts

```
> pie(drinks_table)
```



# Questions?

# Quantitative data

# Quantitative data: statistics

There are several statistics that describe the central tendency of quantitative data…

- The mean: mean()
- The median: median()

Can you calculate the mean and median of OkCupid user’s heights?

# Quantitative data: Visualizing heights

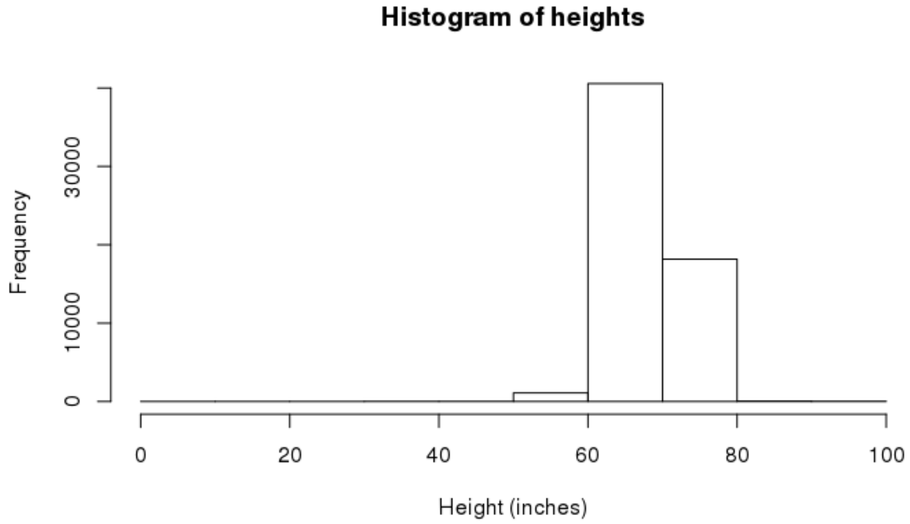
Q: How can we visualize the heights in the profiles data frame?

# Visualizing heights

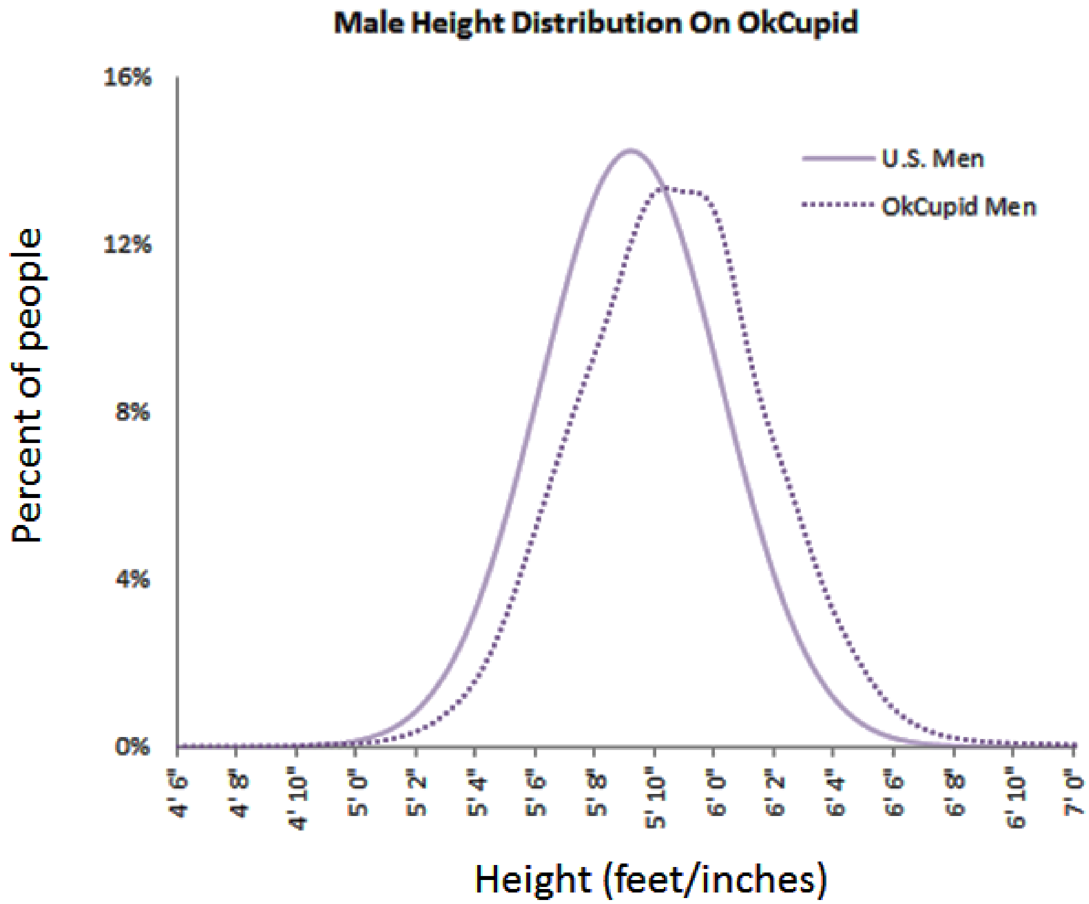
We can create histograms in R using the hist() function

Can you create a histogram of heights?

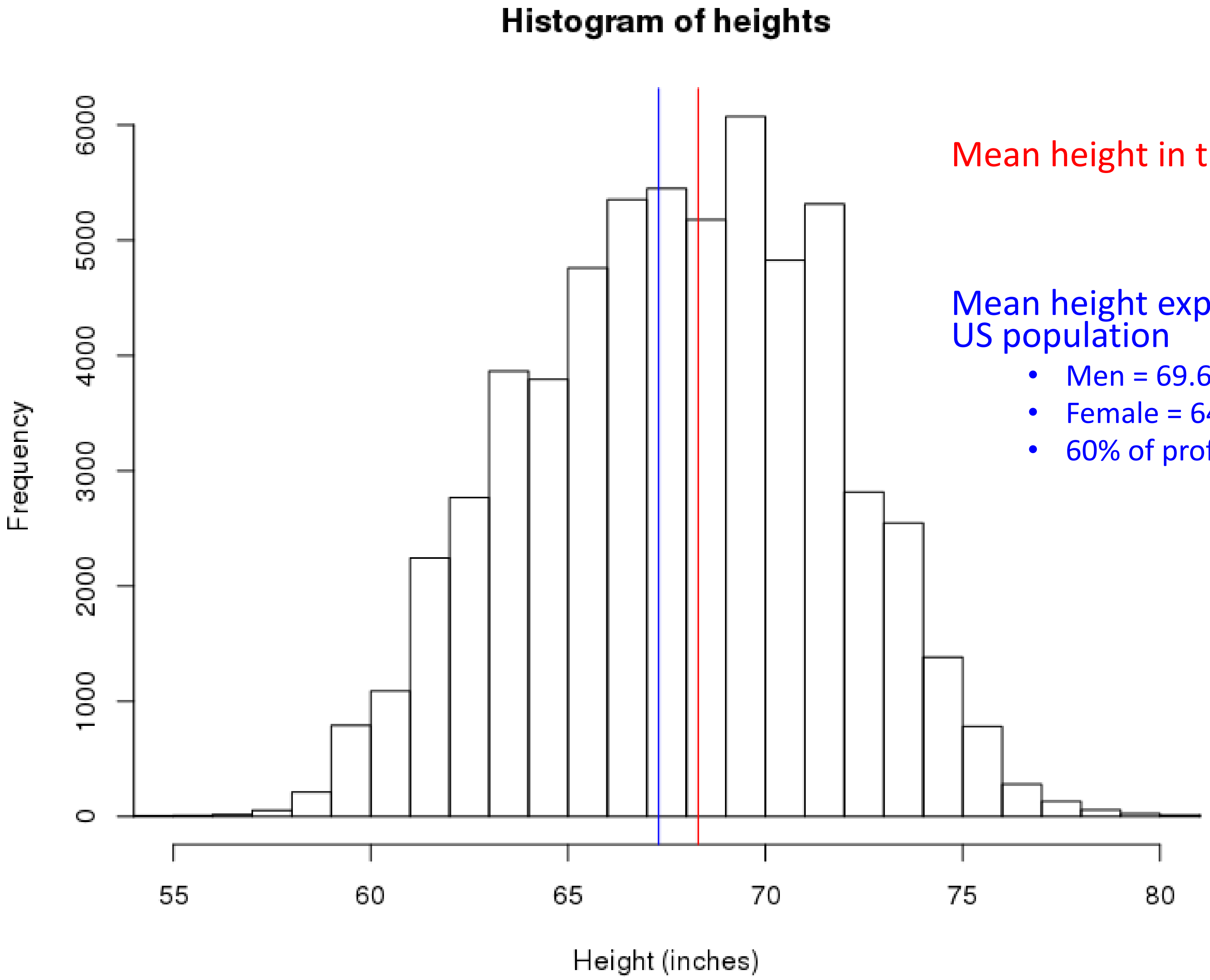
```
> hist(profiles$height)


> hist(profiles$height, breaks = 50)
```



# OkCupid users are taller than the average person



Can we see this in the profiles data?



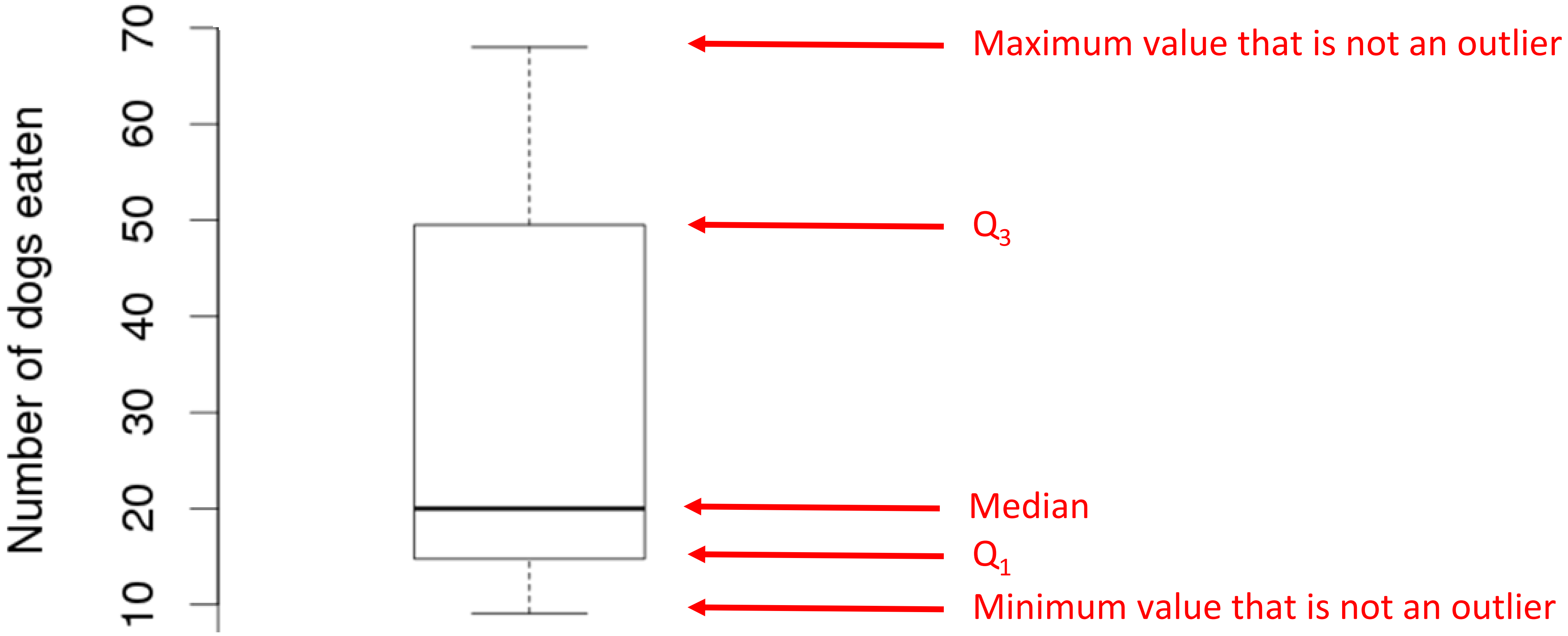
Mean height in the profiles data

Mean height expected based on US population

- Men = 69.6”
- Female = 64”
- 60% of profiles are men

abline() adds lines to plots

# Box plots can also visualize quantitative data



Maximum value that is not an outlier

$Q_3$

Median

$Q_1$

Minimum value that is not an outlier

```
R: boxplot(v)
```

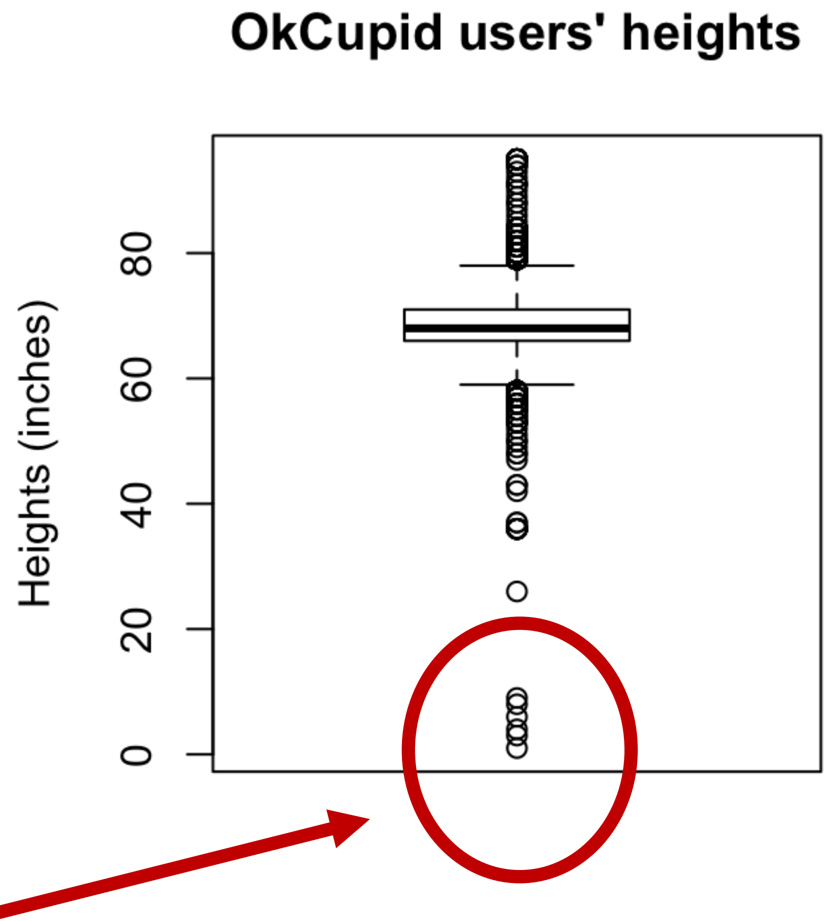
# Outliers

Outliers on boxplots are values that are more than 1.5 * IQR

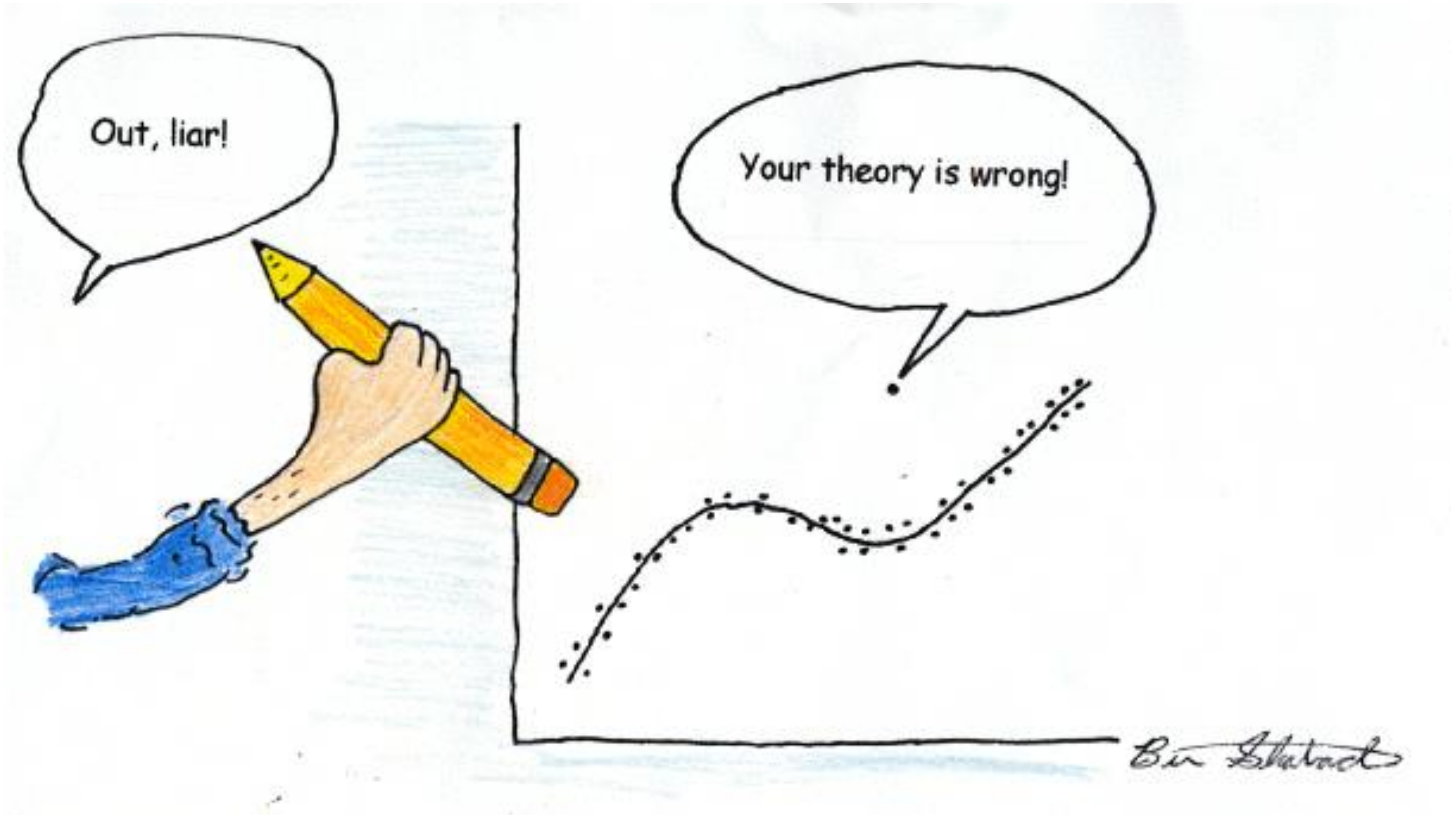
What should we do if we have outliers?

Investigate!

- If there are due to an error, remove them
- **If not, need to account for them**



People under 20” tall?


Out, liar!
Your theory is wrong!

# CitiBike data

Let’s look at the bike share data from NYC

```
> load('daily_bike_totals.rda’)
```

CitiBike analysis

What does each case correspond to?

We can use the dim() function to get how many cases and variables there are

- How many are there?

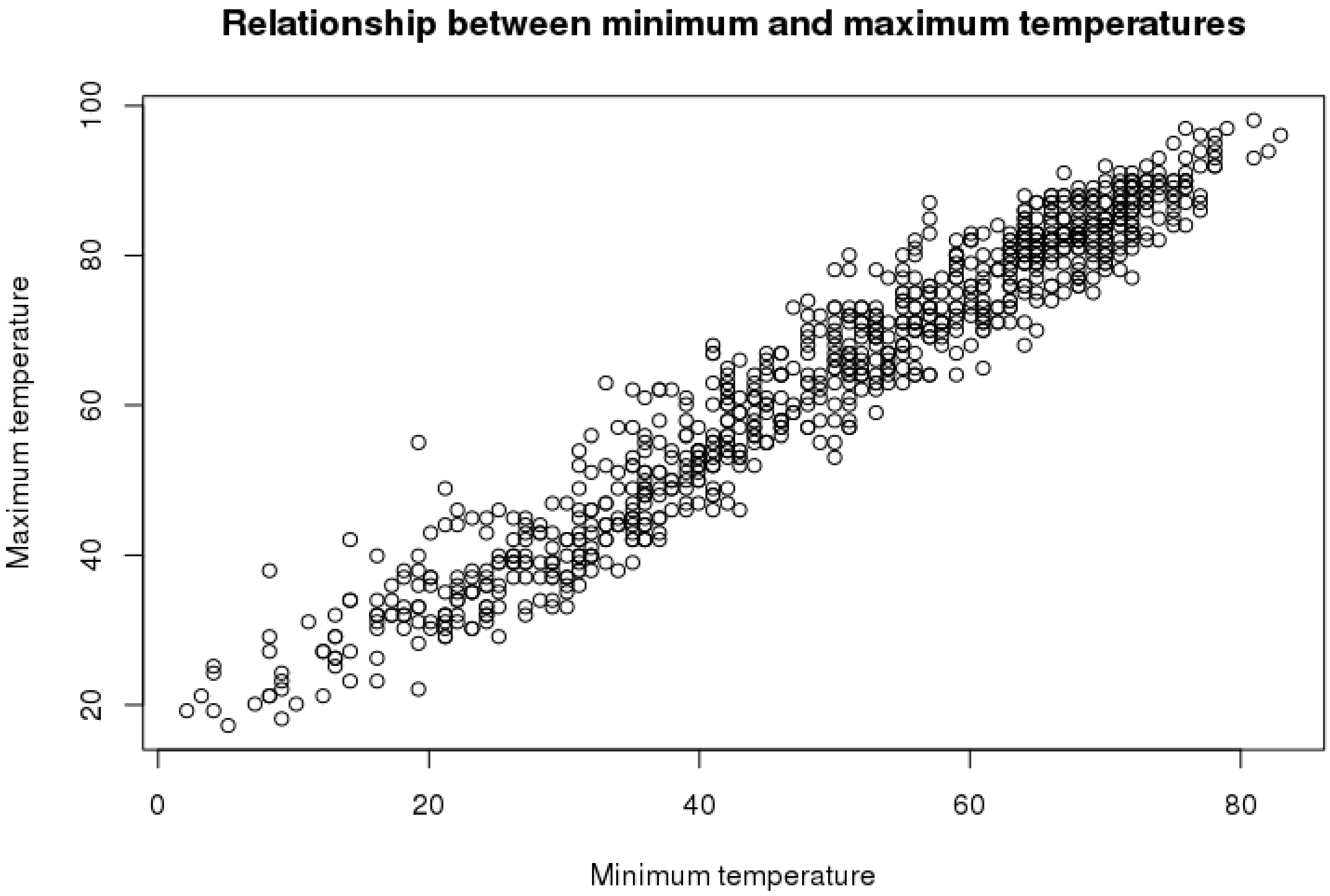
# Scatter plots

We can use the plot(x, y) function to create scatter plots

Can you create a scatter plot of the relationship between the minimum and maximum temperatures?

```
> plot(bike_daily_data$min_temperature,
       bike_daily_data$max_temperature,
       xlab = "Minimum temperature",
       ylab = "Maximum temperature",
       main = "Relationship between min and temp")
```

# Scatter plots

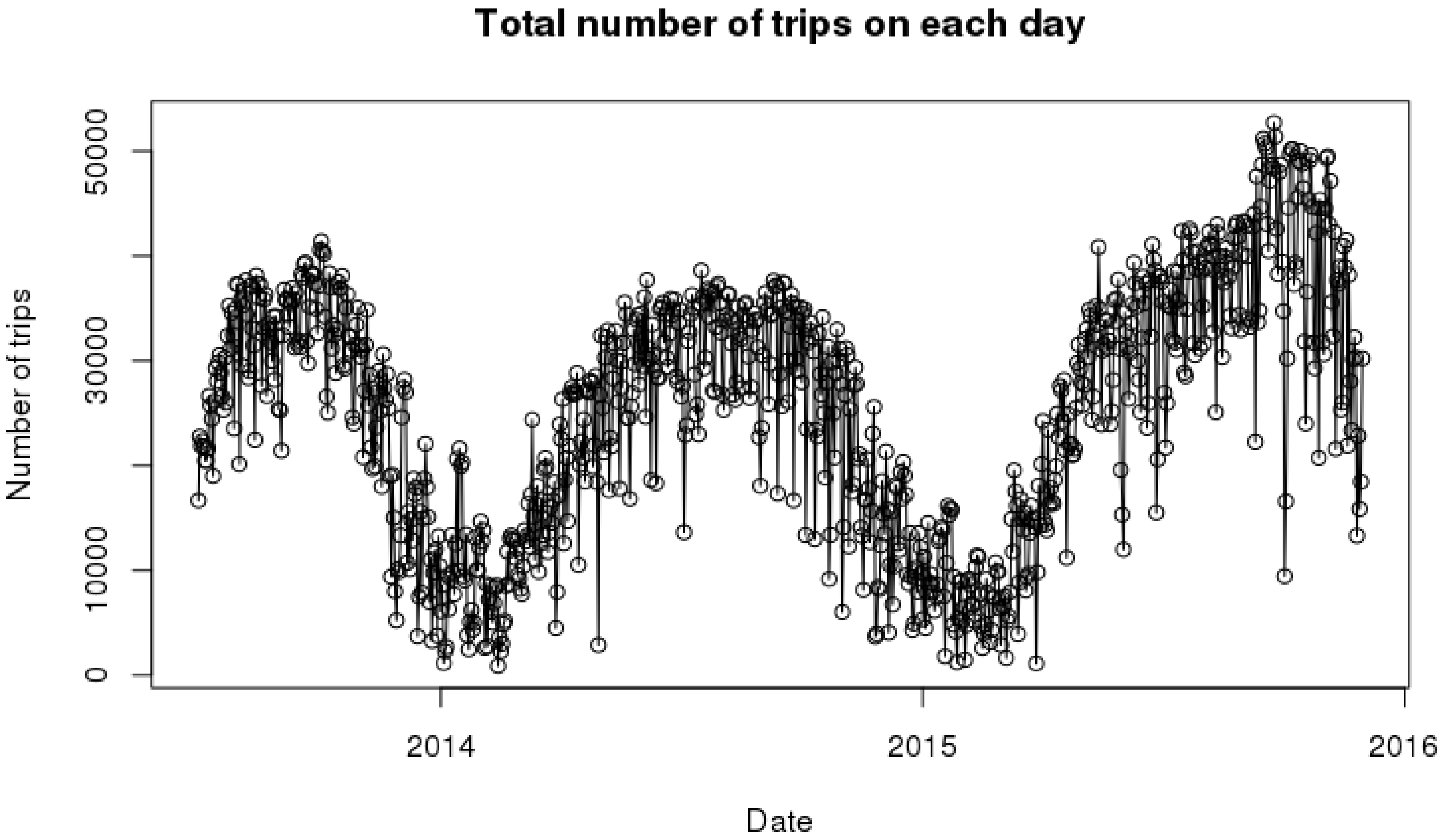


# Plotting time series

We can use the plot(x, y) function to plot time series

```
# we can connect the points in a plot using
> plot(x, y, type = 'l')    # connected points
> plot(x, y, type = 'o')   # both points and dots

> plot(bike_daily_data$date,  bike_daily_data$trips,
            type = 'o',
            xlab = "Date",
            ylab = "Number of trips",
            main = "Total number of trips on each day")
```

# Plotting time series



# Questions?

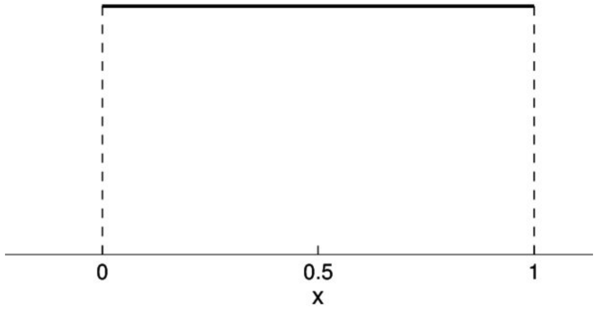
# Generating random data

R has built in functions to generate data from different distributions

- All these functions start with the letter ***r***

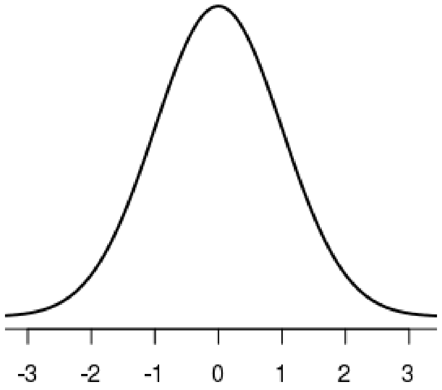
**The uniform distribution**

```
# generate n = 100 points from U(0, 1)
    > rand_data <- runif(100)
    > hist(rand_data)
```



**The normal distribution**

```
 # generate n = 100 points from N(0, 1)
    > rand_data  <- rnorm(100)
    > hist(rand_data)
```



# Generating random data

If we want the same sequence of random numbers we can set the random number generating seed

```
> set.seed(123)
> runif(100)
```

**Q: Why would we want the same sequence of random number?**

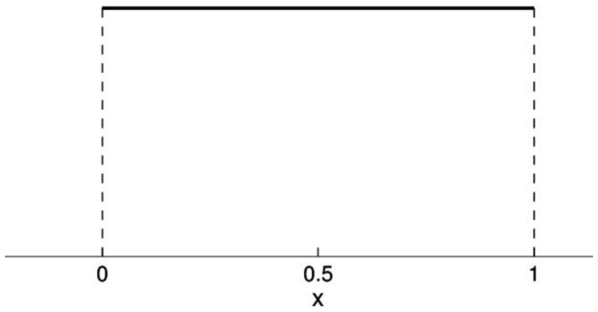
# Cumulative distributions

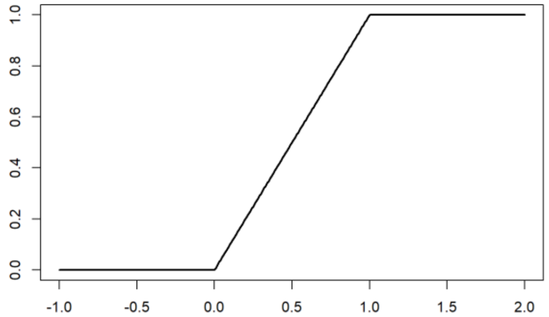
R has built in functions to get probabilities from different distributions

- All these functions start with the letter ***p***

**The uniform distribution**

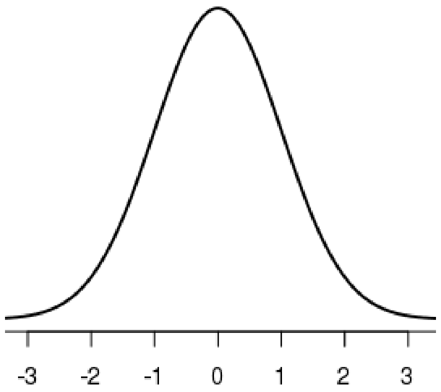
```
# P(X ≤ .25)
      punif(.25)
```

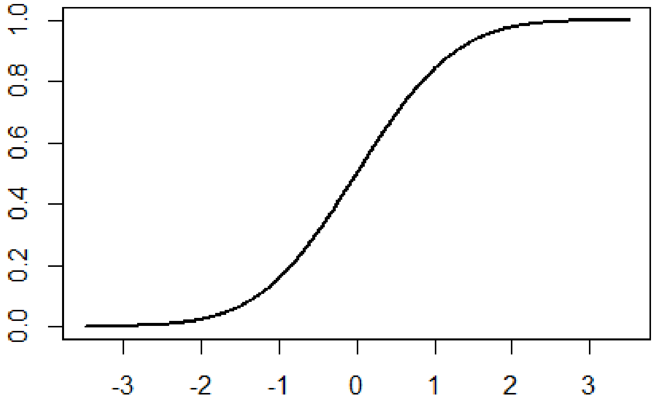




**The normal distribution**

```
# P(X ≤  2)
      pnorm(2)
```





# For loops

For loops are useful when you want to repeat a piece of code many times under similar conditions

The syntax for a for loop is:

```
for (i in 1:100) {
        # do something
}
```

This is repeated 100 times
i is incremented by 1 each time

# For loops

For loops are particularly useful in conjunction with vectors…

```
my_results <- NULL      # create an empty vector to store the results
for (i in 1:100) {
        my_results[i] <- i^2
}
```

# Using a for loop to Simulate a sampling distribution

```
sampling_dist <- NULL
for (i in 1:1000) {
        rand_data <- runif(100)   # generate n = 100 points from U(0, 1)
        sampling_dist[i] <- mean(rand_data)   # save the mean
}

hist(sampling_dist)        # visualize the sampling distribution
SE <- sd(sampling_dist)    # get the standard error
```

Let's try it in R!

# Writing functions

We've used many R functions in this class

Let's explore writing our own functions!