514 Lab 1

Joseph Fulkerson, Julio Pagan

Due Date 9/13

```
# If you don't already have the tidyverse library installed,
# you will need to type install.packages("tidyverse") into the Console
library(tidyverse)
```

1. Changing the author field and file name. (5 points)

(a) Change the `author:` field on the Rmd document

(b) Rename this file to "HW1_YourGroupNumberHere.Rmd", where YourGroupNumberHere is changed to your group number (e.g. Group1).

2. Hello World! (5 points)

Here's an R code chunk that prints the text 'Hello world!'.

```
print("Hello world!")
```

```
## [1] "Hello world!"
```

```


print("Julio Pagan, Joseph Fulkerson")
```

(a) Modify the code chunk below to print your name

```
## [1] "Julio Pagan, Joseph Fulkerson"
```

3. Creating a numeric vector (30 points)

We just learned about the `c()` operator, which forms a vector from its arguments. If we're trying to build a vector containing a sequence of numbers, there are several useful functions at our disposal. These are the colon operator `:` and the sequence function `seq()`.

```
1:10 # Numbers 1 to 10
```

: Colon operator:

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
127:132 # Numbers 127 to 132
```

```
## [1] 127 128 129 130 131 132
```

```
seq(1,10,1) # Numbers 1 to 10
```

seq function: `seq(from, to, by)`

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
seq(1,10,2) # Odd numbers from 1 to 10
```

```
## [1] 1 3 5 7 9
```

```
seq(2,10,2) # Even numbers from 2 to 10
```

```
## [1]  2  4  6  8 10
```

> To learn more about a function, type `?functionname` into your console. E.g., `?seq` pulls up a Help file with the R documentation for the `seq` function.

```
3:12
```

(a) Use : to output the sequence of numbers from 3 to 12

```
##  [1]  3  4  5  6  7  8  9 10 11 12
```

```
print(seq(3,30,3))
```

(b) Use `seq()` to output the sequence of numbers from 3 to 30 in increments of 3

```
##  [1]  3  6  9 12 15 18 21 24 27 30
```

```
x <- 3:12
y <- seq(3,30,3)
print(x*y)
```

(c) Save the sequence from (a) as a variable x, and the sequence from (b) as a variable y. Output their product x*y

```
##  [1]   9  24  45  72 105 144 189 240 297 360
```

4. Cars data (60 points)

We'll look at data frame and plotting in much more detail in later classes. For a previous of what's to come, here's a very basic example.

For this example we'll use a very simple dataset. The `cars` data comes with the default installation of R. To see the first few columns of the data, just type `head(cars)`.

```
head(cars)
```

```
##   speed dist
## 1     4    2
## 2     4   10
## 3     7    4
## 4     7   22
## 5     8   16
## 6     9   10
```

```
speed_average <- mean(cars$speed)
speed_standard_deviation <- sd(cars$speed)
speed_results <- print(paste("The average of speed is " , speed_average , " and the standard deviation
```

(a) Calculate the average and standard deviation of speed

```
## [1] "The average of speed is  15.4  and the standard deviation is  5.28764443523478"
```

> The average of speed is 15.4 and the standard deviation is 5.28764443523478

```
average_dist <- mean(cars$dist)
standard_dev_dist <- sd(cars$dist)
answer_dist <- print(paste("The average of dist is " , average_dist , " and the standard deviation is "
```

(b) Calculate the average and standard deviation of dist

```
## [1] "The average of dist is  42.98  and the standard deviation is  25.7693774920259"
```

> The average of dist is 42.98 and the standard deviation is 25.7693774920259

```
new_cars <- cars[cars$speed > speed_average,]
print(new_cars)
```

(c) Calulate the average and standard deviation of `dist` when `speed` is greater than the average. Compare the results with the answers in (b)

```
##    speed dist
## 27    16   32
## 28    16   40
## 29    17   32
## 30    17   40
## 31    17   50
## 32    18   42
## 33    18   56
## 34    18   76
## 35    18   84
## 36    19   36
## 37    19   46
## 38    19   68
## 39    20   32
## 40    20   48
## 41    20   52
## 42    20   56
## 43    20   64
## 44    22   66
## 45    23   54
## 46    24   70
## 47    24   92
## 48    24   93
## 49    24  120
## 50    25   85
```

```
new_dist_average <- mean(new_cars$dist)
new_standard_dev_dist <- sd(new_cars$dist)
answer_dist <- print(paste("The average of dist with a speed greater than ", speed_average, " is ", new_
```

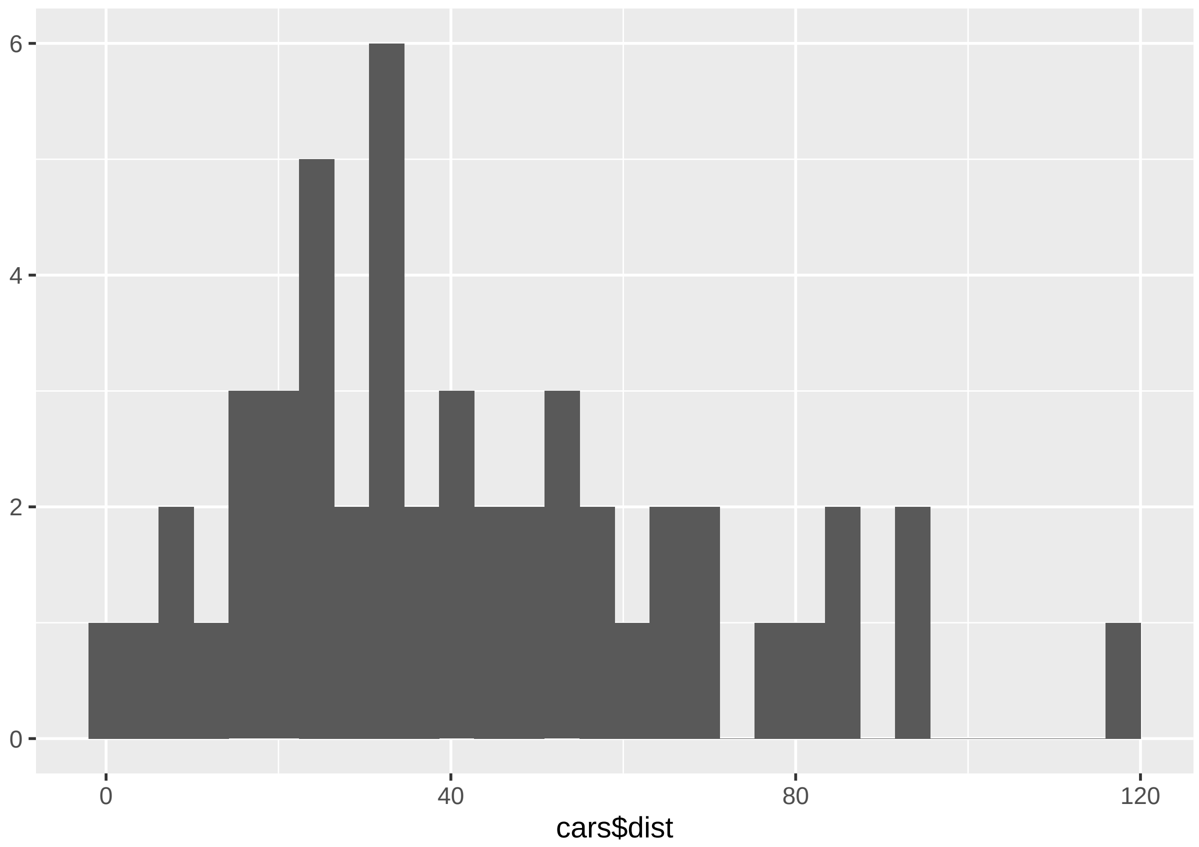
```
## [1] "The average of dist with a speed greater than  15.4  is  59.75  and the standard deviation is  2
```

> The average of dist with a speed greater than 15.4 is 59.75 and the standard deviation is 22.8496979334996

We can easily produce a histogram of stopping distance using the `qplot` function (built-in `tidyverse` package).

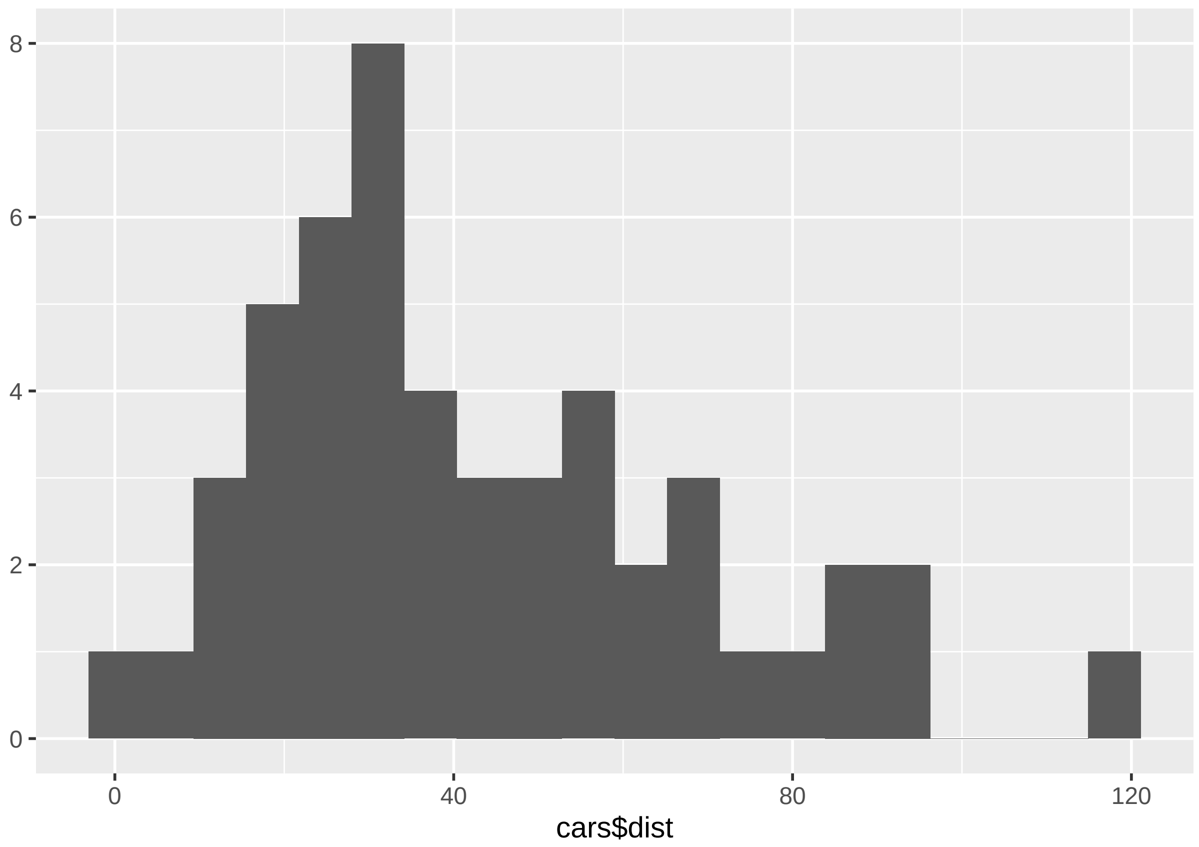
```
qplot(cars$dist) # Histogram of stopping distance
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```

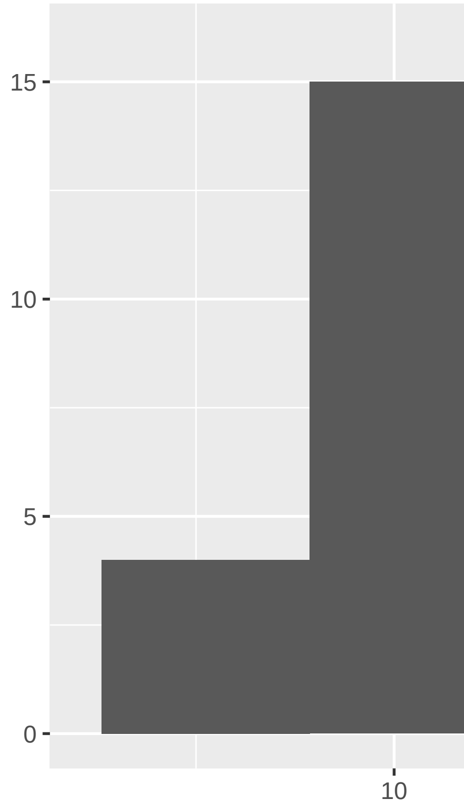


How to change the number of bins?

```
qplot(cars$dist,bins = 20)
```



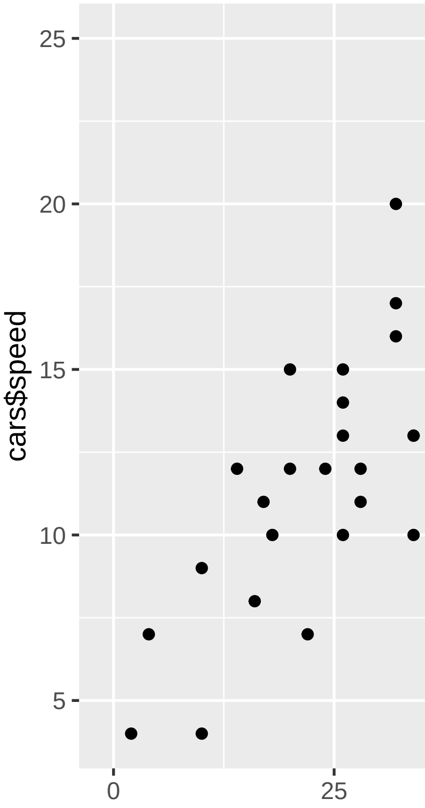
```
qplot(cars$speed,bins = 5)
```



(d) Produce a histogram of speed using the `qplot` function with 5 bins.

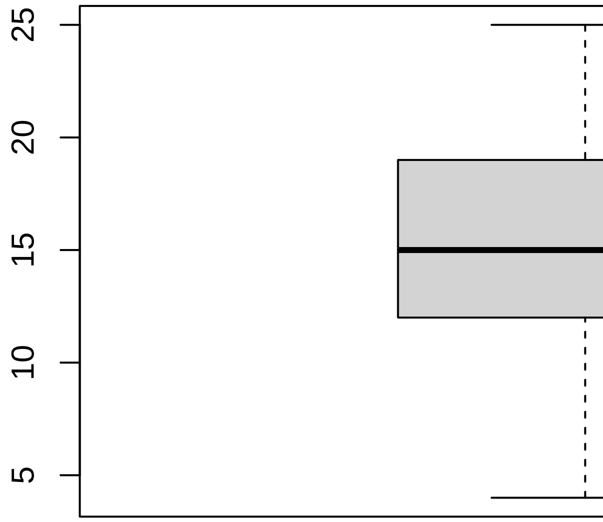
The `qplot(x,y,...)` function can also be used to plot a vector `y` against a vector `x`. You can type `?qplot` into the Console to learn more about the basic qplot function.

```
qplot(cars$dist,cars$speed)
```



(e) Use the `qplot(x,y)` function to create a scatterplot of dist against speed.

```
boxplot(cars$speed)
```



(f) Use the `boxplot` function to create a boxplot of speed.