R Notebook

1.a. Write a function that calculates the mean of any numeric vector you give it, without using the built-in mean() or sum() functions.

```
mean.fun <- function(x)
{
  count <-  0
  for (i in 1 : length(x))
  {
    count <-  count + x[i]
  }
  mean.val <- count / length(x)
  return(mean.val)
}

v <- 1:10
mean.fun(v)
```

```
## [1] 5.5
```

1.b. Write a function that takes as its input a vector with four elements. If the sum of the first two elements is greater than the sum of the second two, the function returns the vector; otherwise it returns 0.

```
func1 <- function(x)
{
  sum1 <- x[1] + x[2]
  sum2 <- x[3] + x[4]
  if (sum1 > sum2)
  {
    return(x)
  }
  else
  {
      return(0)
  }
}

vec <- 20:2
vec
```

```
##  [1] 20 19 18 17 16 15 14 13 12 11 10  9  8  7  6  5  4  3  2
```

```
func1(vec)
```

```
##  [1] 20 19 18 17 16 15 14 13 12 11 10  9  8  7  6  5  4  3  2
```

1.c. Write a function that calculates the Fibonacci sequence up to the nth element, where n is any number input into your function (its argument). The Fibonacci sequence is: 1, 1, 2, 3, 5, 8, 13, 21. . . , ie, each element is the sum of the previous two elements. One way to do this is to start off with the first two elements, c(1,1) and set an internal variable to this sequence. Then write a loop that counts up to n, where for each new element, you first calculate it by adding the last two elements of the growing sequence, and then stick that new number onto the growing sequence using c(). When the loop is finished, the function should return the final vector of Fibonacci numbers

```
fib.fun <- function(x)
{
  fib.val <- numeric(x)
  fib.val[1] <- 1
  fib.val[2] <- 1
  for (i in 3 : x)
  {
    fib.val[i] <- fib.val[i - 1] + fib.val[i - 2]
  }
  return(fib.val)
}

x <- 10
fib.fun(x)
```

```
##  [1]  1  1  2  3  5  8 13 21 34 55
```

1.d. Create a 4x4 matrix of the numbers 1 through 16. Use apply to apply you function from (a) to each of the rows in your matrix.

```
m1 <- matrix(data = 1 : 16, nrow = 4, ncol = 4)
m1
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    5    9   13
## [2,]    2    6   10   14
## [3,]    3    7   11   15
## [4,]    4    8   12   16
```

```
apply(m1, 1, mean.fun)
```

```
## [1]  7  8  9 10
```

2.a. Using the airquality dataset, constuct an aggregated dataset which shows the maximum wind and ozone by month.

```
data("airquality")
head(airquality)
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

```
aggregate(data = airquality, cbind(Wind,Ozone) ~ Month,  max)
```

```
##   Month Wind Ozone
## 1     5 20.1   115
## 2     6 20.7    71
## 3     7 14.9   135
## 4     8 15.5   168
## 5     9 16.6    96
```

2.b. Create the authors and books datasets following the example and data in the lecture, and then create a new data set by merging these two datasets by author, preserving all rows.

```
authors <- data.frame(
    surname = c("Tukey", "Venables", "Tierney", "Ripley", "McNeil"),
    nationality = c("US", "Australia", "US", "UK", "Australia"),
    stringsAsFactors=FALSE)
books <- data.frame(
    name = c("Tukey", "Venables", "Tierney",
             "Ripley", "Ripley", "McNeil", "R Core"),
    title = c("Exploratory Data Analysis",
              "Modern Applied Statistics ...",
              "LISP-STAT",
              "Spatial Statistics", "Stochastic Simulation",
              "Interactive Data Analysis",
              "An Introduction to R"),
    stringsAsFactors=FALSE)
merge(authors, books, by.x = "surname", by.y = "name", all = TRUE)
```

```
##    surname nationality                         title
## 1   McNeil   Australia     Interactive Data Analysis
## 2   R Core        <NA>           An Introduction to R
## 3   Ripley          UK             Spatial Statistics
## 4   Ripley          UK         Stochastic Simulation
## 5  Tierney          US                     LISP-STAT
## 6    Tukey          US     Exploratory Data Analysis
## 7 Venables   Australia Modern Applied Statistics ...
```

2.c. Take the following string and replace every instance of "to" or "To" with "2" To be, or not to be – that is the question: Whether 'tis nobler in the mind to suffer The slings and arrows of outrageous fortune, Or to take arms against a sea of troubles, And by opposing end them. To die – to sleep – No more...

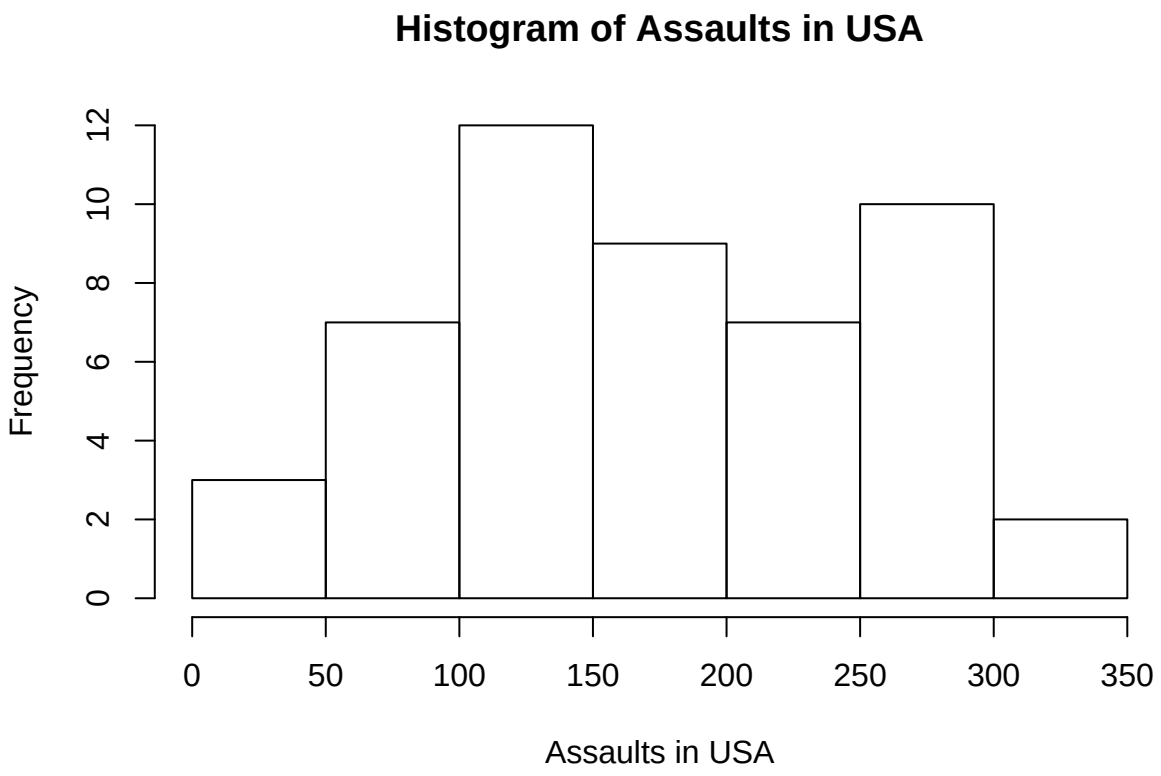
```
verse <- "To be, or not to be - that is the question: Whether 'tis nobler in the mind to suffer The sli

gsub("(T|t)o", "2", verse)
```

```
## [1] "2 be, or not 2 be - that is the question: Whether 'tis nobler in the mind 2 suffer The slings a
```

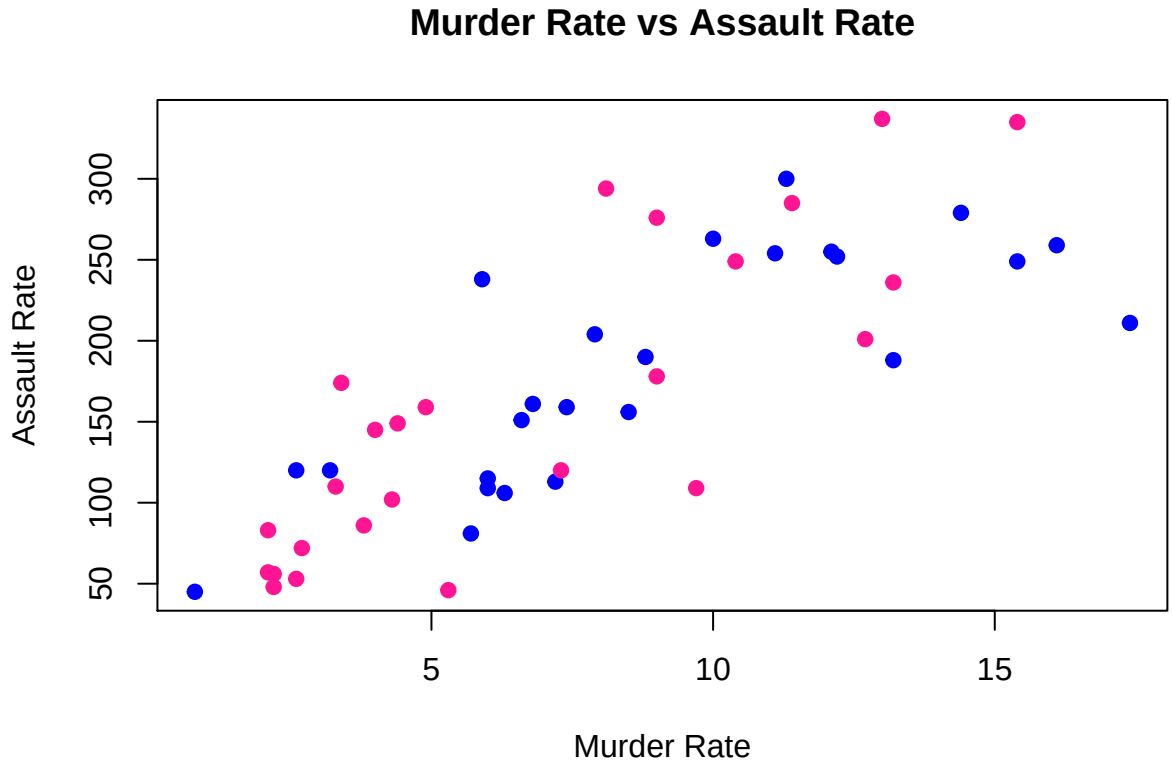
3.a. Create a histogram using the base R graphics using some dataset or variable other than the one in the lessons. Always make sure your graph has well-labeled x and y axes and an explanatory title.

```
data("USArrests")
hist(USArrests$Assault, main = "Histogram of Assaults in USA", xlab = "Assaults in USA")
```



3.b. Create a scatter plot using the base R graphics, again with some variable other than the one in the lessons.

```
plot(USArrests$Murder, USArrests$Assault, main = "Murder Rate vs Assault Rate", xlab = "Murder Rate", yl
```



3.c. Create a histogram using ggplot, using some new data. In this and the later plots, please tinker with the

settings using the examples in http://www.cookbook-r.com/Graphs/ to make it prettier.

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  2.1.3     v purrr   0.3.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

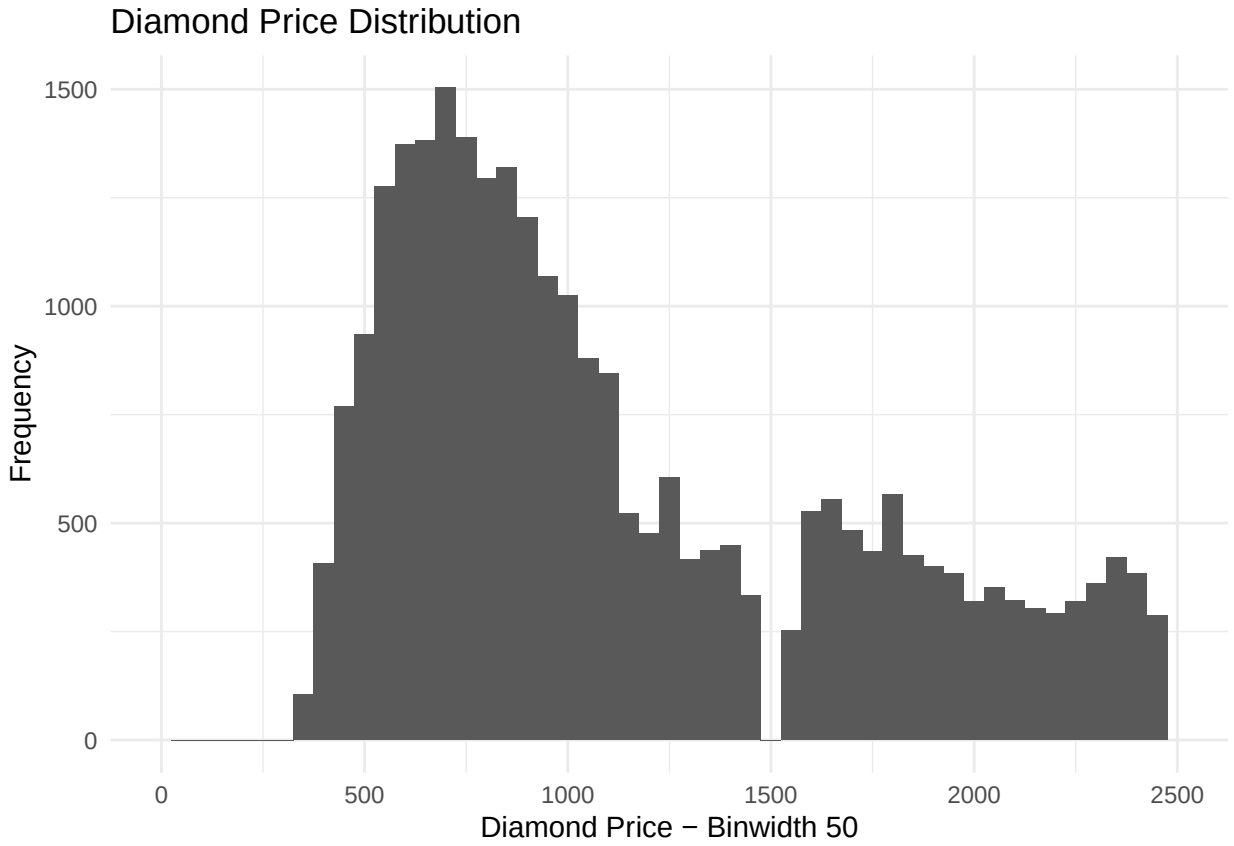
```
data("diamonds")
head(diamonds)
```

```
## # A tibble: 6 x 10
##   carat cut       color clarity depth table price     x     y     z
##   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1 0.23  Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
## 2 0.21  Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
## 3 0.23  Good      E     VS1      56.9    65   327  4.05  4.07  2.31
## 4 0.290 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
## 5 0.31  Good      J     SI2      63.3    58   335  4.34  4.35  2.75
## 6 0.24  Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
```

```
ggplot(data = diamonds, mapping = aes(x = price)) + geom_histogram(binwidth = 50) + ggtitle("Diamond Pr
```

```
## Warning: Removed 26398 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

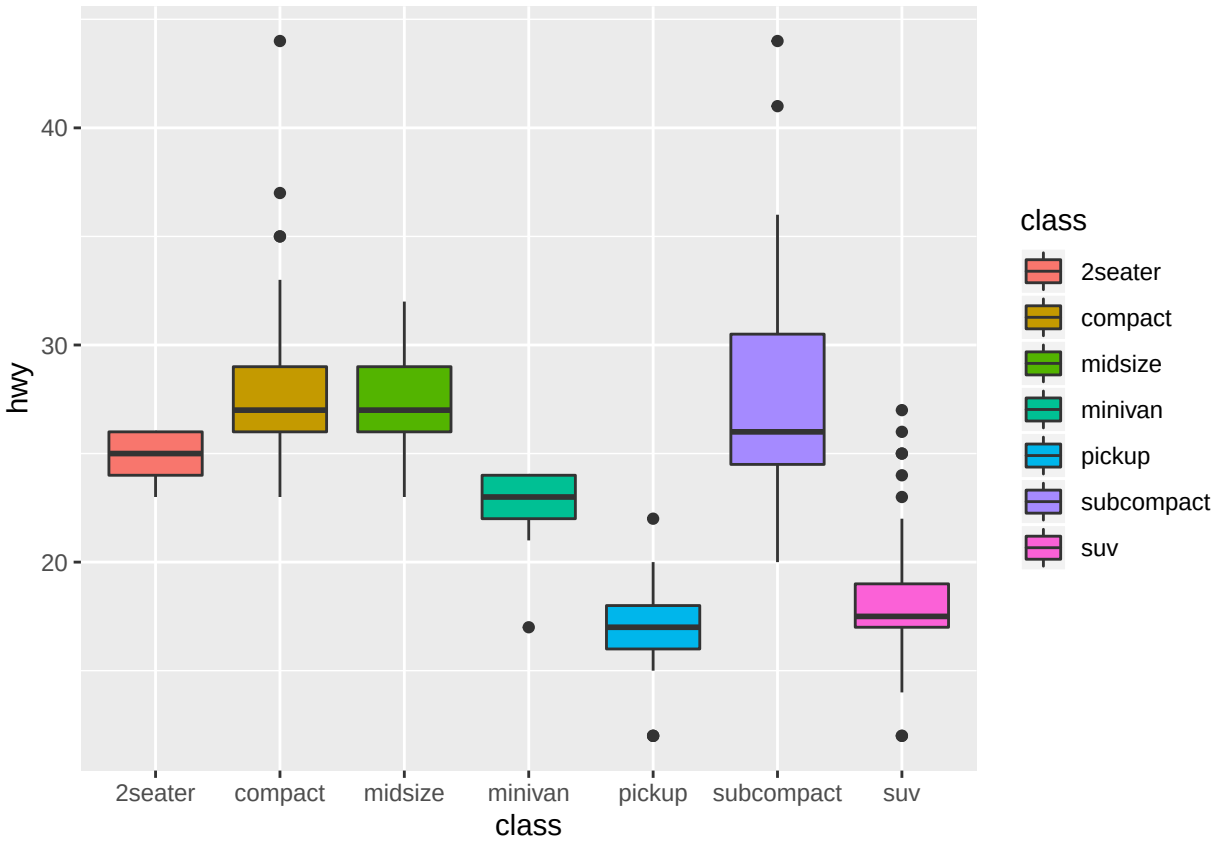


3.d. Create a box plot (with multiple categories) using ggplot, using some new data.

```
data(mpg)
head(mpg)
```

```
## # A tibble: 6 x 11
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa~
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
## 5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p     compa~
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa~
```

```
qplot(data=mpg, x = class, y = hwy, geom = 'boxplot', fill = class)
```

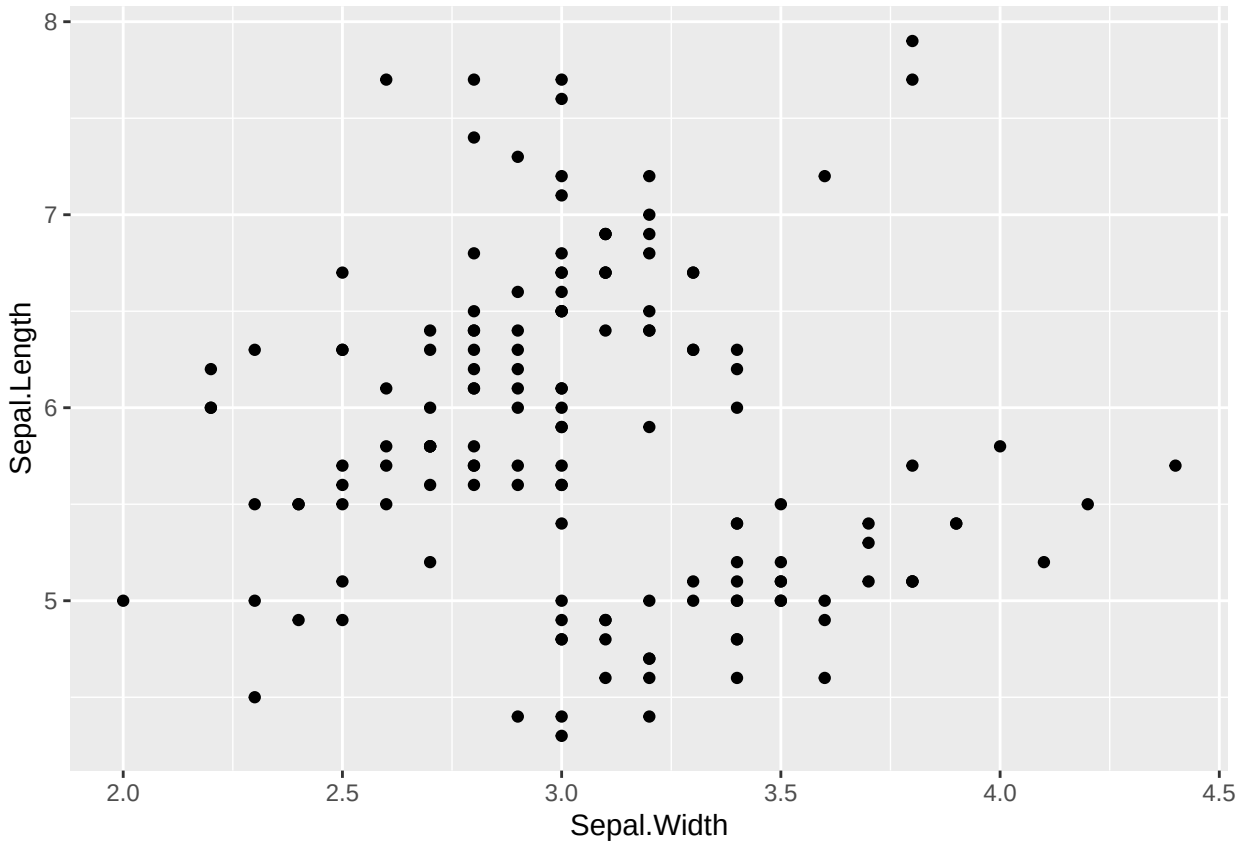


3.e. Create a scatter plot using ggplot, using some new data

```
data("iris")
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
ggplot(data = iris, mapping = aes(x = Sepal.Width, y = Sepal.Length)) + geom_point()
```


Sepal.Length
Sepal.Width
8
7
6
5
2.0
2.5
3.0
3.5
4.0
4.5