Class 3 - Digging into data

Digging into Data

You'll all be working with your own data, so we'll review a concept with the `iris` data set, and then you'll have a few mintues to try it on your own. We'll cover the basic ways I explore data before diving in more depth.

Reading in data: two ways

```
##---- for CSV files ----##
#data <- read.csv("NAME.csv", as.is=T)

##---- for XLSX files ----##
#install.packages("xslx")
library(xlsx)
#data <- read.xlsx("NAME.xlsx", sheetIndex = 1) #you want the first (or only) sheet

## for the demo
iris_data <- iris
```

Some mild data cleaning

Open up your data from the Evnironment pane - how does it look? Do you need to fix anything?

```
#################################################################################
##--- Common issue #1: Column 1 is just row numbers, so let's remove it ---##
#################################################################################
iris_data <- iris
iris_data <- iris_data[, -1] # the - sign just removes that column number



#################################################################################
##--- Common issue #2: The column names get messed up (usually the first one), so let's rename --##
#################################################################################
##--- This is optional, but essential if you need to join data together on a key (advanced)
iris_data <- iris
names(iris_data)[1] <- "Sepal_Length_woohoo"



#################################################################################
##--- Common issue #3: Missing data is NA when it should be 0, or 0 when it should be NA ---##
#################################################################################
iris_data <- iris
## replacing all entries that are 0 to be NA instead
iris_data$Sepal.Length[iris_data$Sepal.Length == 0] <- NA
```

```
## replacing all entries that are NA to be 0 instead
iris_data$Sepal.Length[is.na(iris_data$Sepal.Length)] <- 0

################################################################################
##--- Common issue #4: (during analysis) oops, I deleted all my data! ----##
################################################################################
iris_data <- iris_data[iris_data$Species == "Yeehaw", ] #oops!

## It's ok! Just re-run the lines where you read in the data!
```

Data digging - let's go!

IMPORTANT: you need to keep track of interesting findings or weird results. Even something like a running page in a notebook. Lots of data analysis is exploration, so you need to ensure you're keeping track of what you've come upon.

Comparing means across groups

```
## loading up the right libraries
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(ggplot2)
```

```
## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang
```

```
## I got bored with the flowers - let's use titanic data
#install.packages("titanic")
library(titanic)
titanic_data <- titanic_train

################################################################################
##----Task 1: pick a group (strings or numbers) and summarize all other numeric columns - then dig into
################################################################################
summary(titanic_data)
```

```
##   PassengerId       Survived          Pclass          Name
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median :446.0   Median :0.0000   Median :3.000   Mode  :character
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000
##
##      Sex                 Age            SibSp           Parch
##  Length:891         Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  Class :character   1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000
##  Mode  :character   Median :28.00   Median :0.000   Median :0.0000
##                     Mean   :29.70   Mean   :0.523   Mean   :0.3816
##                     3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##                     Max.   :80.00   Max.   :8.000   Max.   :6.0000
##                     NA's   :177
##     Ticket               Fare           Cabin             Embarked
##  Length:891         Min.   :  0.00   Length:891         Length:891
##  Class :character   1st Qu.:  7.91   Class :character   Class :character
##  Mode  :character   Median : 14.45   Mode  :character   Mode  :character
##                     Mean   : 32.20
##                     3rd Qu.: 31.00
##                     Max.   :512.33
##
```

```
titanic_data %>%
  group_by(Sex) %>%
  summarise_if(is.numeric, mean, na.rm=T)
```

```
## # A tibble: 2 x 8
##   Sex    PassengerId Survived Pclass   Age SibSp Parch  Fare
##   <chr>        <dbl>    <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 female        431.    0.742   2.16  27.9 0.694 0.650  44.5
## 2 male          454.    0.189   2.39  30.7 0.430 0.236  25.5
```

```
## example of digging futher into a finding - men had fewer family members - what % were alone?
titanic_data %>%
  count(Sex, SibSp, Parch) # this isn't that useful :(
```

```
## # A tibble: 40 x 4
##    Sex    SibSp Parch     n
##    <chr>  <int> <int> <int>
##  1 female     0     0   126
##  2 female     0     1    24
##  3 female     0     2    20
##  4 female     0     3     1
##  5 female     0     4     1
##  6 female     0     5     2
##  7 female     1     0    63
##  8 female     1     1    26
##  9 female     1     2    11
## 10 female     1     3     2
## # ... with 30 more rows
```

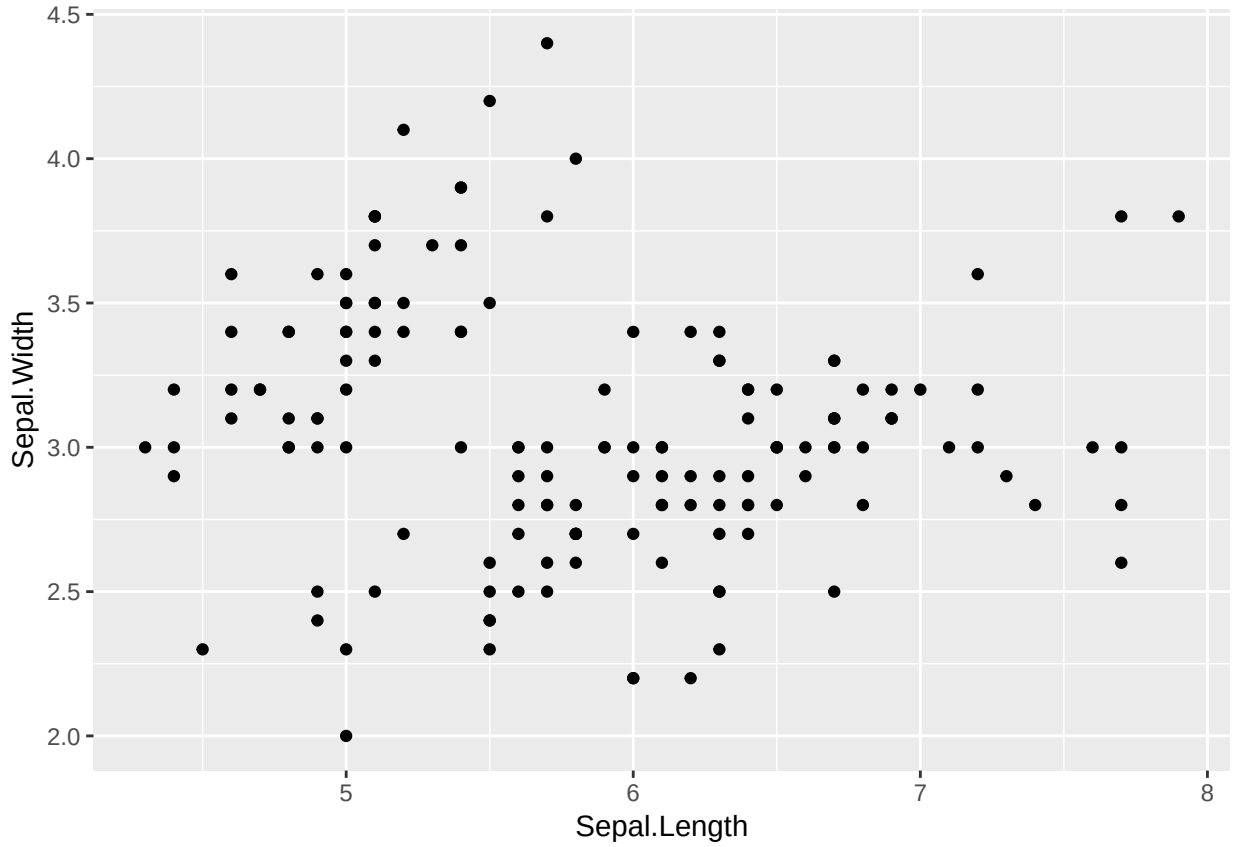
```r
titanic_data %>%
  group_by(Sex) %>%
  summarize(PercentAlone = mean(SibSp == 0 & Parch == 0), # grouping by sex, what % have both 0 sibs an
            NumAlone = sum(SibSp == 0 & Parch == 0), # grouping by sex, what # have both 0 sibs and 0 p
            total = n()) # how many of each sex are there?
```

```
## # A tibble: 2 x 4
##   Sex    PercentAlone NumAlone total
##   <chr>         <dbl>    <int> <int>
## 1 female        0.401      126   314
## 2 male          0.712      411   577
```

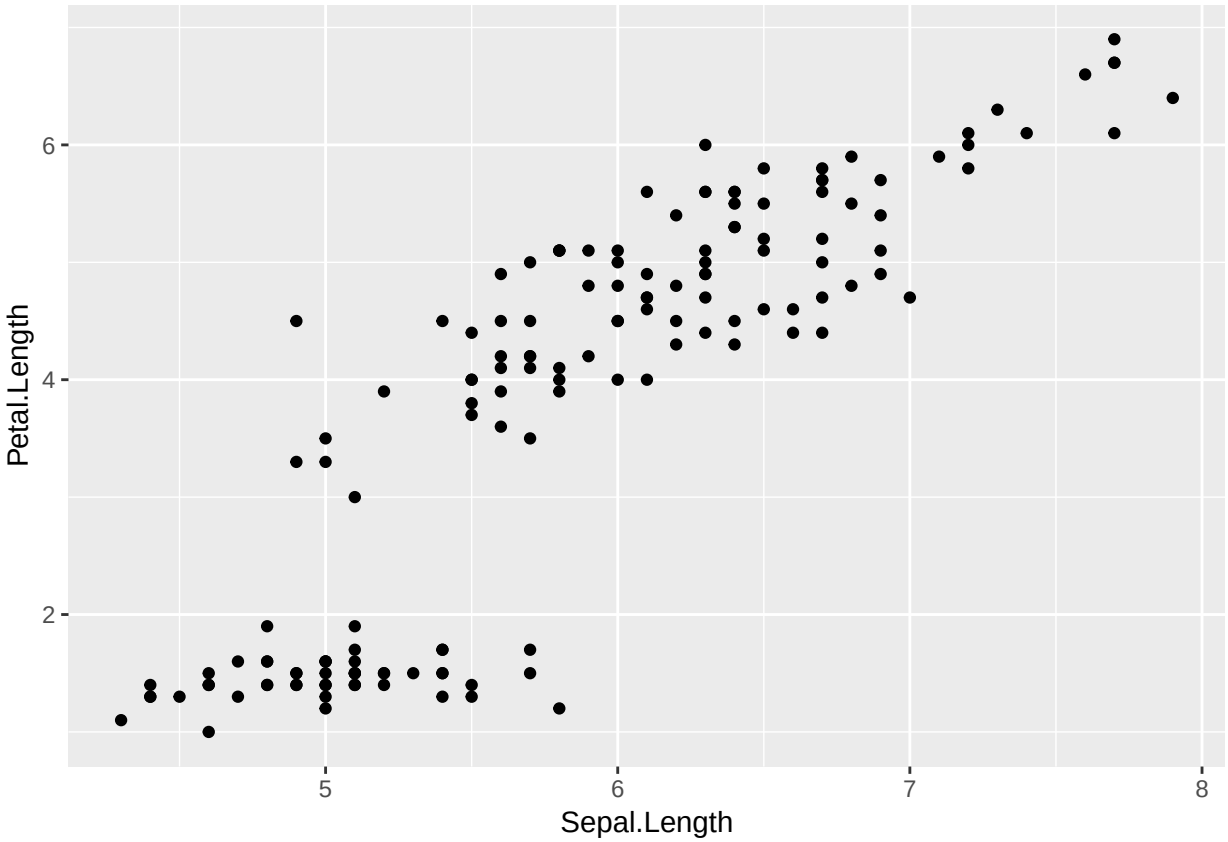
```r
##############################################################################################
### Copy & paste & try it on your own!
##############################################################################################
```

Plotting

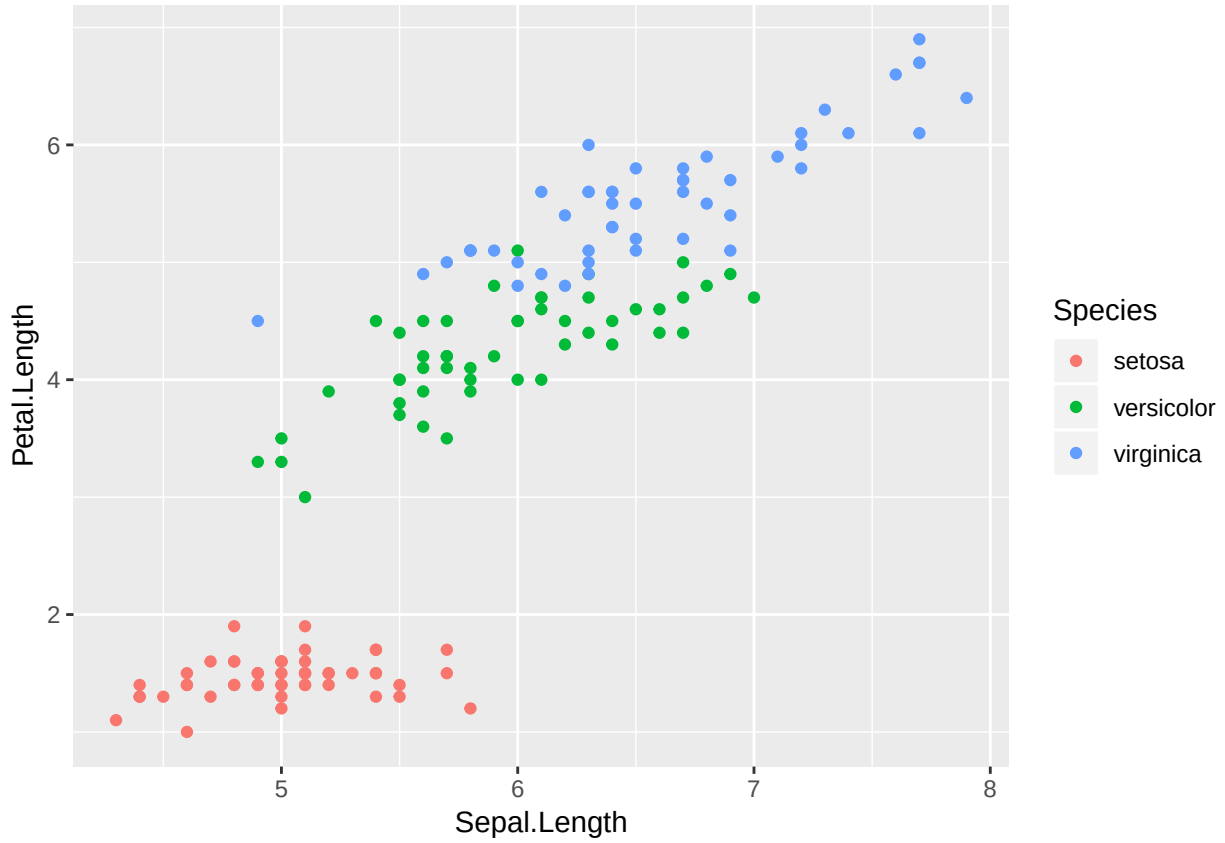
```r
library(dplyr)
library(ggplot2)
##############################################################################################
## Task 2: Scatter plots to understand relationship between variables
##############################################################################################
iris_data <- iris
ggplot(data = iris_data, aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point()
```




```
ggplot(data = iris_data, aes(x = Sepal.Length, y = Petal.Length)) +
  geom_point()
```

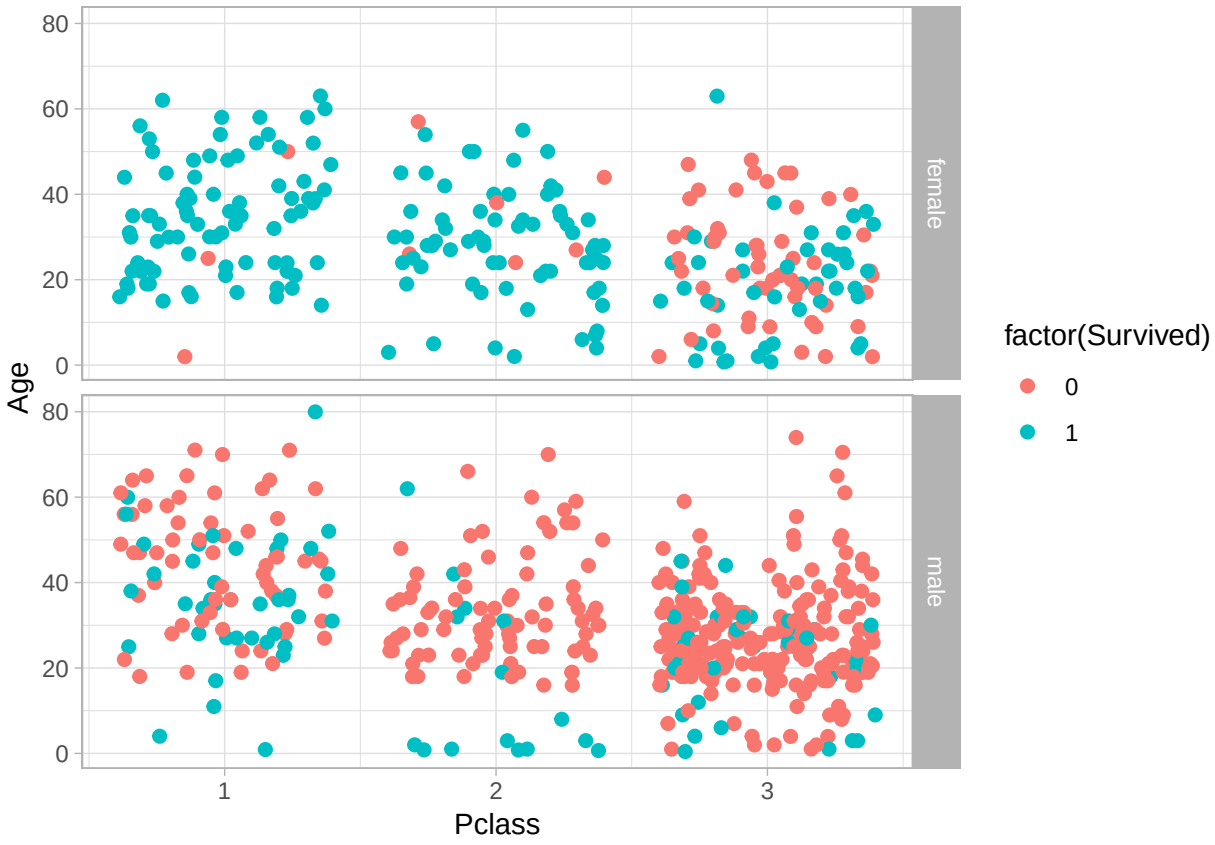



```
ggplot(data = iris_data, aes(x = Sepal.Length, y = Petal.Length, color = Species)) +
  geom_point()
```



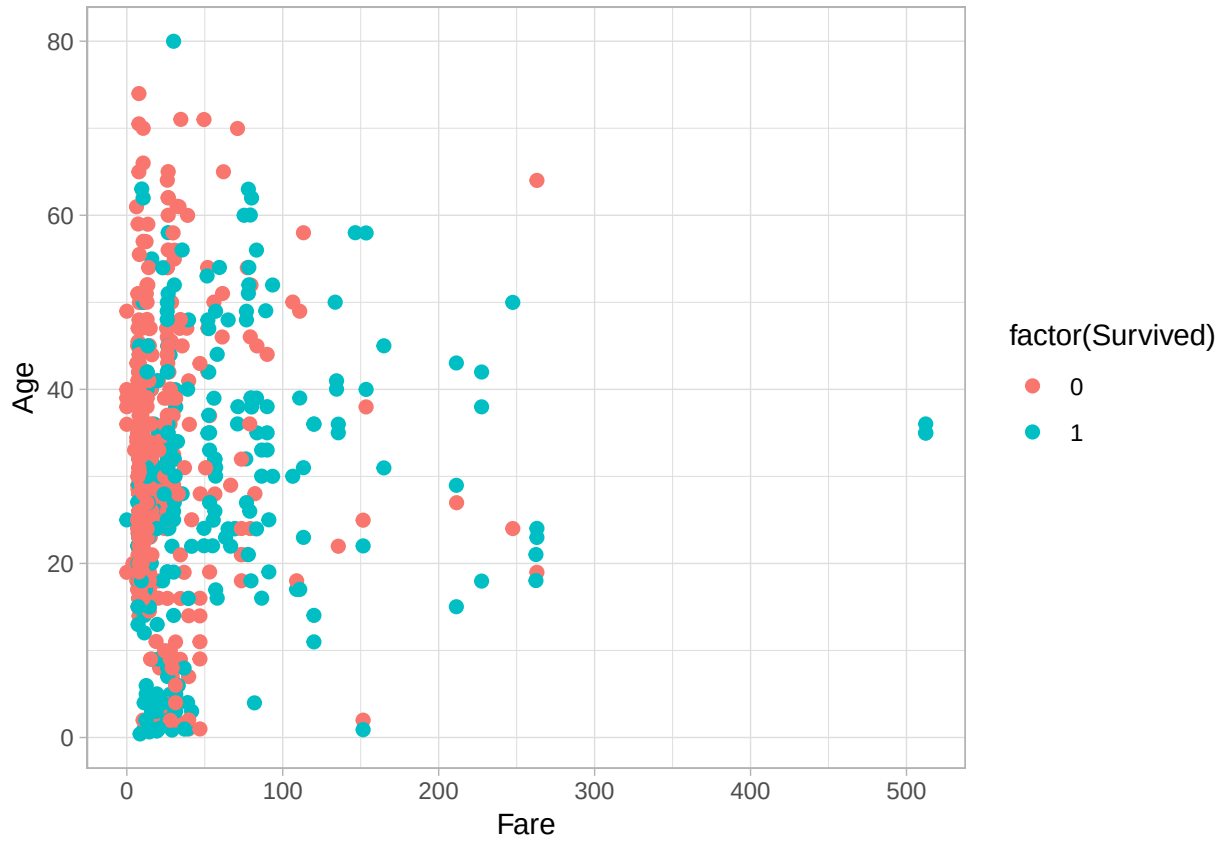

```
## same thing but exploring the titanic data set
## adapted from here - https://www.kaggle.com/josepandreu/titanic-visualization-with-ggplot2
ggplot(data = titanic_data, aes(Pclass, Age, colour = factor(Survived))) +
    geom_jitter(size = 2) +
    facet_grid(Sex ~ .) +
    theme_light()
```

```
## Warning: Removed 177 rows containing missing values (geom_point).
```




```
ggplot(data = titanic_data, aes(x= Fare, Age, colour = factor(Survived))) +
    geom_jitter(size = 2) +
    theme_light()
```

```
## Warning: Removed 177 rows containing missing values (geom_point).
```




```
##############################################################################################
## Copy & paste & try on your own!
##############################################################################################
```

Making a new column

```
##############################################################################################
## Task 3 - make a new column that you want to use in your analysis
##############################################################################################
## copying code from Class 2
iris_data <- iris
iris_data$SepalSize <- "Average"
iris_data$SepalSize[iris_data$Sepal.Length < 5.8 & iris_data$Sepal.Width < 3] <- "Small"
iris_data$SepalSize[iris_data$Sepal.Length > 5.8 & iris_data$Sepal.Width > 3] <- "Big"

## other example using titanic data
library(titanic)
titanic_data <- titanic_test

## how does age differ among those who were alone & those who weren't?
titanic_data$Alone <- 0
titanic_data$Alone[titanic_data$SibSp == 0 & titanic_data$Parch == 0] <- 1

titanic_data %>%
```

```
  group_by(Alone, Sex) %>%
  summarise(AvgAge = mean(Age, na.rm=T))
```

```
## # A tibble: 4 x 3
## # Groups:   Alone [2]
##   Alone Sex    AvgAge
##   <dbl> <chr>   <dbl>
## 1     0 female   31.0
## 2     0 male     30.3
## 3     1 female   29.2
## 4     1 male     30.3
```

```
##############################################################################################
## Make a new column & then either plot it or do some numeric summaries (recycle above code!)
##############################################################################################
```