# Data Cleaning Example

jb

5/20/2020

## Intro

Lets start this exercise with loading some student admissions data. This is a simple example where we will explore our data – no other real goal.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.3
```

```
##
## DATA SET
##
##
##
Myfile="SummerStudentAdmissions3_.csv"
## USE YOUR OWN PATH AS NEEDED
MyData <- read.csv(Myfile)
```

## Data Acquisition and Data Cleaning

After loading the data, its a good idea to view it to confirm that the data loaded correctly. Try using commands "View", "str" or "head".

```
############################################
## Part 1: Cleaning the data
##         using data vis - ggplot
##
##         EDA is Exploratory Data ANalysis
##             Clean and explore...
################################################

## LOOK AT Each Variable.
str(MyData)
```

```
## 'data.frame':    88 obs. of  9 variables:
##  $ Decision      : Factor w/ 5 levels "","Admit","Banana",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Gender        : Factor w/ 3 levels "","Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ DateSub       : Factor w/ 73 levels "1/10/2020","1/11/2020",..: 2 2 3 40 32 37 41 23 15 5 ...
##  $ State         : Factor w/ 12 levels "Alabama","California",..: 4 4 3 3 3 2 2 2 3 4 ...
##  $ GPA           : num  3.54 3.55 3.59 3.6 3.6 3.66 3.7 3.7 3.75 3.77 ...
##  $ WorkExp       : num  0.7 0 1.7 0.9 1.2 0.9 1.2 2.7 1.1 1.4 ...
##  $ TestScore     : int  965 962 969 969 967 956 969 799 969 969 ...
##  $ WritingScore  : int  11 97 93 97 94 89 94 97 93 99 ...
##  $ VolunteerLevel: int  1 0 0 2 2 1 2 5 0 4 ...
```

```r
## Notice that there are 9 variables

## Variable (also called features, attributes, columns) Name
(MyVarNames<-names(MyData))
```

```
## [1] "Decision"       "Gender"         "DateSub"        "State"
## [5] "GPA"            "WorkExp"        "TestScore"      "WritingScore"
## [9] "VolunteerLevel"
```

```r
MyVarNames[1]
```

```
## [1] "Decision"
```

```r
MyData[MyVarNames[1]]
```

```
##    Decision
## 1     Admit
## 2     Admit
## 3     Admit
## 4     Admit
## 5     Admit
## 6     Admit
## 7     Admit
## 8     Admit
## 9     Admit
## 10    Admit
## 11    Admit
## 12    Admit
## 13    Admit
## 14    Admit
## 15    Admit
## 16    Admit
## 17    Admit
## 18    Admit
## 19   Banana
## 20  Decline
## 21  Decline
## 22  Decline
## 23  Decline
## 24  Decline
## 25  Decline
## 26  Decline
## 27  Decline
## 28  Decline
## 29  Decline
## 30  Decline
## 31  Decline
## 32  Decline
## 33  Decline
## 34  Decline
## 35  Decline
## 36  Decline
## 37 Waitlist
## 38 Waitlist
## 39 Waitlist
## 40 Waitlist
## 41 Waitlist
## 42 Waitlist
## 43 Waitlist
## 44 Waitlist
## 45 Waitlist
## 46 Waitlist
## 47 Waitlist
## 48 Waitlist
## 49 Waitlist
## 50 Waitlist
## 51 Waitlist
## 52 Waitlist
## 53 Waitlist
## 54 Waitlist
## 55 Waitlist
## 56 Waitlist
## 57
## 58    Admit
## 59    Admit
## 60    Admit
## 61    Admit
## 62    Admit
## 63    Admit
## 64    Admit
## 65    Admit
## 66    Admit
## 67    Admit
## 68    Admit
## 69    Admit
## 70    Admit
## 71    Admit
## 72    Admit
## 73  Decline
## 74  Decline
## 75  Decline
## 76  Decline
## 77  Decline
## 78  Decline
## 79  Decline
## 80  Decline
## 81  Decline
## 82  Decline
## 83 Waitlist
## 84 Waitlist
## 85 Waitlist
## 86 Waitlist
## 87 Waitlist
## 88  Decline
```

```
(NumColumns <-ncol(MyData))
```

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

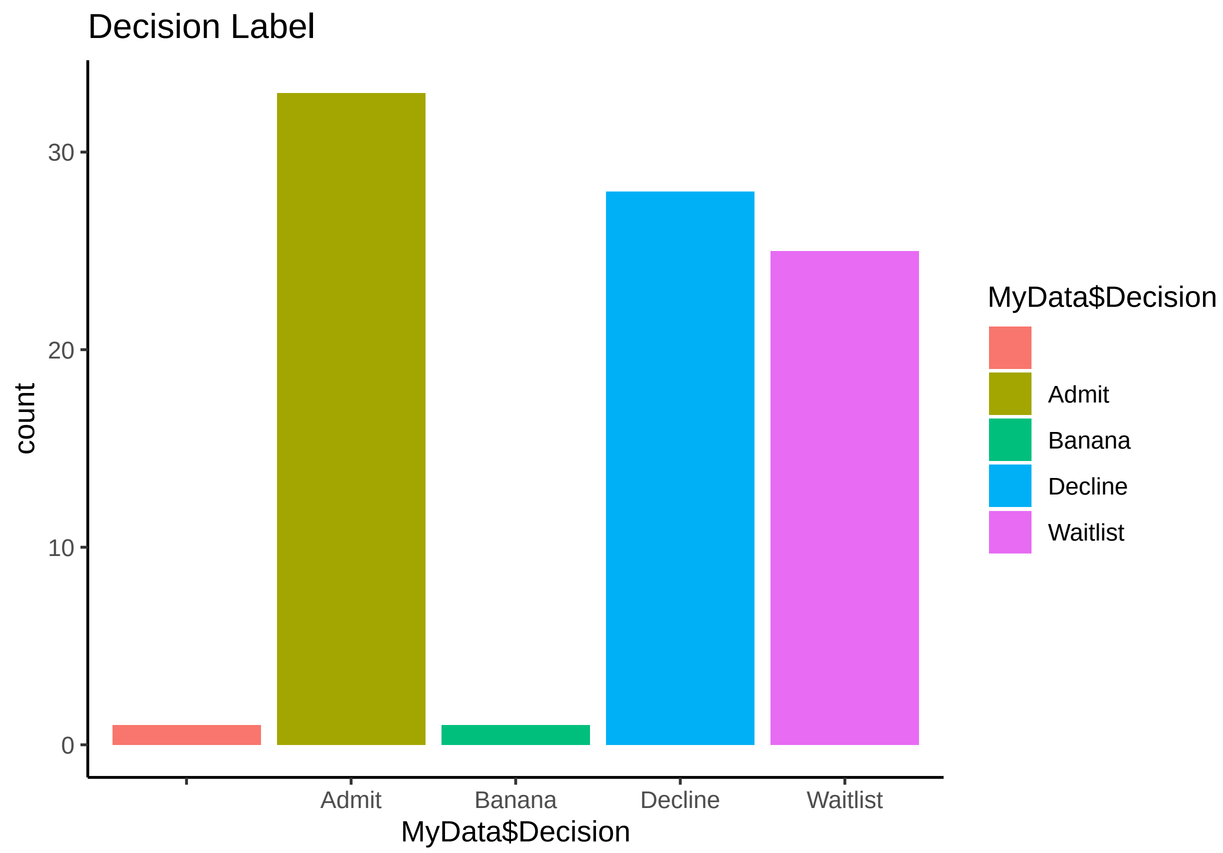
```
View(MyData)
```

Note that the "label" is the first column in the data frame. This is standard in R. The label is the class or classification of the data (often the dependent variable). Thus not considered part of the data, but rather the label. This variable should be of type factor, so lets confirm.

```
##############################
## Column 1: Decision
###################################

## THis is NOT part of the data!
## It is the LABEL of the data.

## Dataset labels should be of type factor
str(MyData$Decision)
```

```
##  Factor w/ 5 levels "","Admit","Banana",..: 2 2 2 2 2 2 2 2 2 2 ...
```

```
## VISUALIZE to SEE what/where the errors are
theme_set(theme_classic())
MyBasePlot1 <- ggplot(MyData)
(MyBasePlot1<-MyBasePlot1 +
    geom_bar(aes(MyData$Decision, fill = MyData$Decision)) +
    ggtitle("Decision Label"))
```



## Uncovering Issues

OK - We have problems. Upon inspection of this one column … - 1) We have a blank level - likely from a missing value. - 2) We have a label called banana - whichis wrong.??!?

## Fixing Issues

Let's fix these. To fix factor data, first convert it to char. Lets remove "invalid rows", and confirm via inspection.

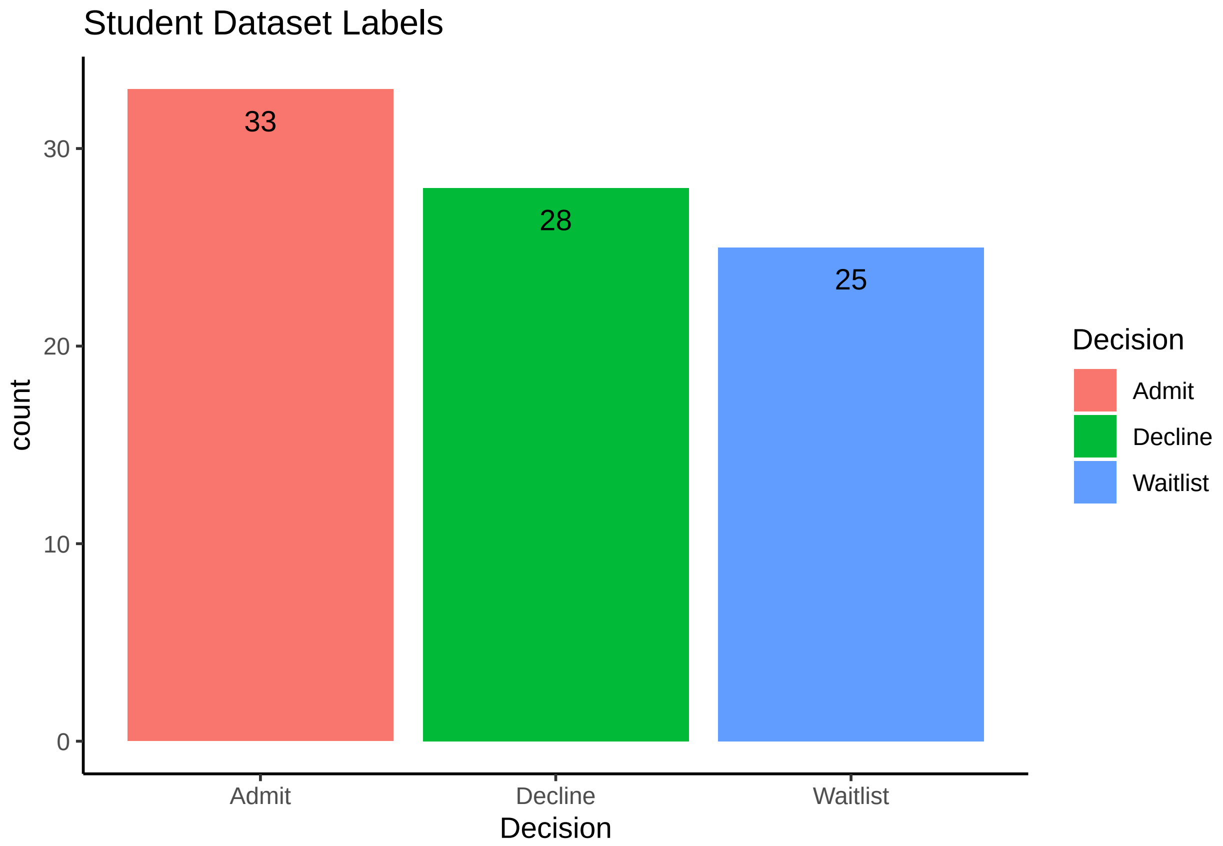
```r
nrow(MyData)
```

```
## [1] 88
```

```r
MyData$Decision <- as.character(MyData$Decision)

## Keep only rows that are  "Admit", "Decline", or "Waitlist"

MyData <- MyData[(MyData$Decision == "Admit" |
                    MyData$Decision == "Decline" |
                    MyData$Decision == "Waitlist"),]

nrow(MyData)
```

```
## [1] 86
```

```
## Check it again

(MyPlot1<-ggplot(MyData, aes(x=Decision, fill=Decision)) +
    geom_bar()+
    geom_text(stat='count',aes(label=..count..),vjust=2)+
    ggtitle("Student Dataset Labels"))
```



### More Cleaning ….

Success! Now we can see (and show others) that theLabel in the dataset it clean and balanced. NOTE that we have color, a title, an x-axis label and labeled bars. We also have a legend.

We are not done!! We need to change Decision back to a factor and inspect the other variables.

```
(str(MyData$Decision))
```

```
##  chr [1:86] "Admit" "Admit" "Admit" "Admit" "Admit" "Admit" "Admit" "Admit" ...
```

```
## NULL
## This needs to be changed to type: factor
MyData$Decision<-as.factor(MyData$Decision)
## Check it
table(MyData$Decision)
```

```
##
##    Admit  Decline Waitlist
##       33       28       25
```

```
str(MyData$Decision)
```

```
##  Factor w/ 3 levels "Admit","Decline",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
## Good! We now have factor data with 3 levels.
```

Lets look at Gender next! This is a qualitative variable, lets visualize using a pie chart.

```
#################################################
## THe  next variable to look at is Gender
## Like Decision, Gender is also qualitative.
## Let's use a pie to look at it...
#################################################

str(MyData$Gender)
```

```
##  Factor w/ 3 levels "","Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
```

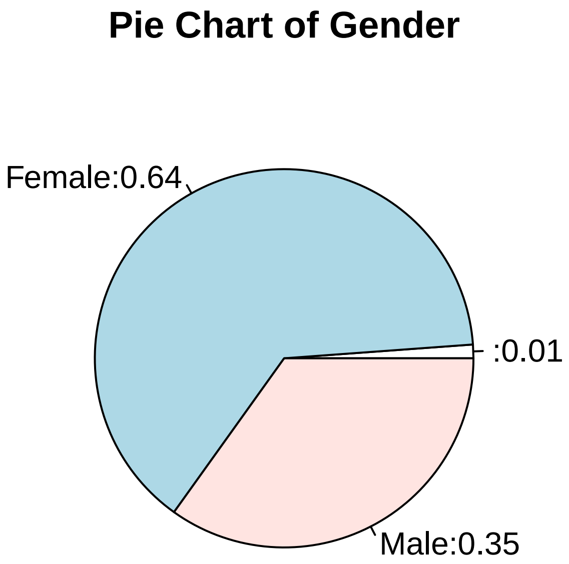
```
NumRows=nrow(MyData)
(TempTable <- table(MyData$Gender))
```

```
##
##        Female    Male
##      1     55     30
```

```
(MyLabels <- paste(names(TempTable), ":",
                   round(TempTable/NumRows,2) ,sep=""))
```

```
## [1] ":0.01"       "Female:0.64" "Male:0.35"
```

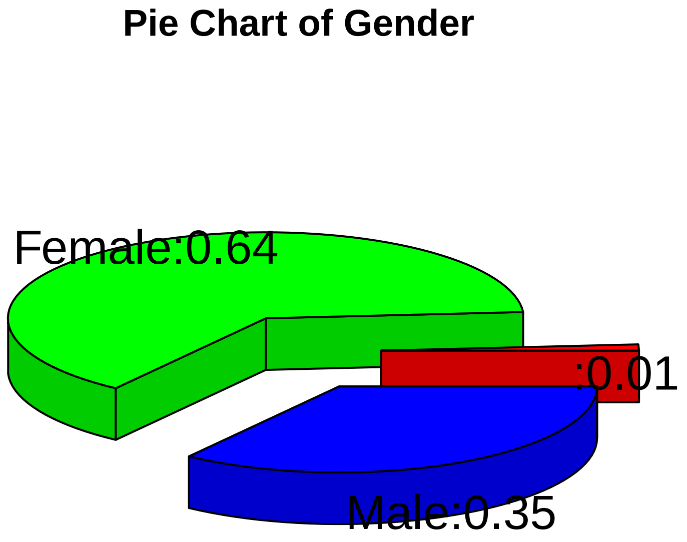
```
pie(TempTable, labels = MyLabels,
    main="Pie Chart of Gender")
```



```
#install.packages("plotrix")
library(plotrix) # Cool 3-d plot here!!
```

```
## Warning: package 'plotrix' was built under R version 3.5.3
```

```
pie3D(TempTable,labels=MyLabels,explode=0.3,
      main="Pie Chart of Gender ")
```



```
table(MyData$Gender)
```

```
##
##         Female   Male
##      1     55     30
```

Houston … We have one problem! We have a blank or NA in the data … but how to fix this? Lets use "is.na"

```
(sum(is.na(MyData$Gender)))  ## This confirms that it is not NA
```

```
## [1] 0
```

Interesting … our mystery value is not an "NA" … what is it??

```
## Let's look at str
str(MyData$Gender)
```

```
##  Factor w/ 3 levels "","Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
```

```
## This shows that we have blank and not NA....
## FIX - change to char, correct, change back to factor
## Keep track of what you are removing from the dataset
```

Its a “blank”. Lets get rid of this row.

```
nrow(MyData)
```

```
## [1] 86
```

```
MyData$Gender <- as.character(MyData$Gender)
## Keep only rows that are Male or Female

MyData <- MyData[(MyData$Gender == "Male" |
                    MyData$Gender == "Female")  ,]
nrow(MyData)
```

```
## [1] 85
```

```
## Turn back to factor
MyData$Gender<- as.factor(MyData$Gender)
str(MyData$Gender)
```

```
##  Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1 1 1 1 1 ...
```

```
table(MyData$Gender)
```

```
##
## Female   Male
##     55     30
```

Lets recreate our Data Viz to confirm!

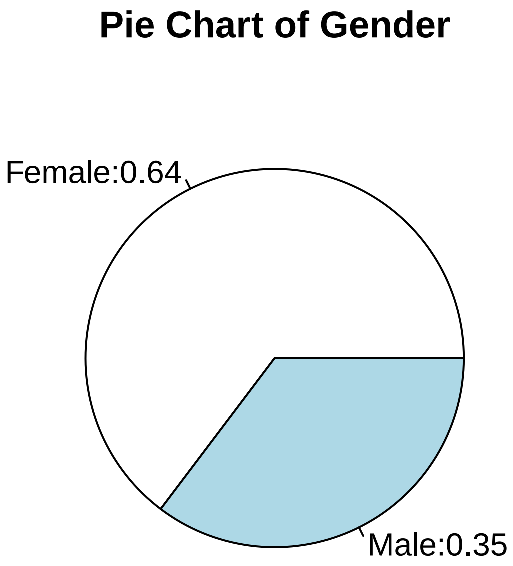
```
(TempTable <- table(MyData$Gender))
```

```
##
## Female   Male
##     55     30
```

```
(MyLabels <- paste(names(TempTable), ":",
                   round(TempTable/NumRows,2) ,sep=""))
```

```
## [1] "Female:0.64" "Male:0.35"
```

```
pie(TempTable, labels = MyLabels,
    main="Pie Chart of Gender")
```



Lets inspect and clean the remaining variables.

```
############################################
## Next variable is: DateSub
#############################################
#names(MyData)
## Check format
str(MyData$DateSub)  ## It is incorrect.
```

```
##  Factor w/ 73 levels "1/10/2020","1/11/2020",..: 2 2 3 40 32 37 41 23 15 5 ...
```

```
## Check for NAs
(sum(is.na(MyData$DateSub)))
```

```
## [1] 0
```

```
## Check the table
table(MyData$DateSub)
```

```
##
##  1/10/2020  1/11/2020  1/12/2020  1/14/2020  1/15/2020  1/17/2020  1/22/2020
##          2          2          3          1          2          1          1
##  1/23/2020  1/25/2020  1/28/2020  1/29/2020  1/30/2020  1/31/2020   1/5/2020
##          1          2          1          1          1          1          2
## 10/10/2019 10/14/2019 10/19/2019 10/25/2019  10/3/2019 10/30/2019 10/31/2019
##          1          1          1          1          1          1          1
##  10/4/2019  10/7/2019  11/1/2019 11/10/2019 11/15/2019 11/16/2019 11/17/2019
##          0          1          1          1          1          1          1
## 11/18/2019 11/19/2019  11/2/2019 11/21/2019 11/25/2019 11/26/2019 11/27/2019
##          0          1          1          1          1          1          1
## 11/28/2019  11/3/2019 11/30/2019  11/4/2019  11/7/2019  11/8/2019  11/9/2019
##          1          1          1          2          2          1          1
##  12/1/2019 12/10/2019 12/11/2019 12/20/2019 12/21/2019 12/23/2019 12/24/2019
##          1          1          1          2          1          1          1
## 12/25/2019 12/27/2019 12/28/2019 12/29/2019  12/3/2019 12/30/2019 12/31/2019
##          1          1          1          1          2          2          1
##  12/4/2019  12/6/2019  12/8/2019  12/9/2019   2/1/2020  2/10/2020  2/11/2020
##          2          1          1          1          2          1          1
##  2/15/2020  2/16/2020  2/19/2020   2/2/2020  2/27/2020   2/4/2020   2/7/2020
##          2          1          1          0          1          1          1
##  3/20/2020   3/6/2020  9/13/2019
##          1          1          1
```

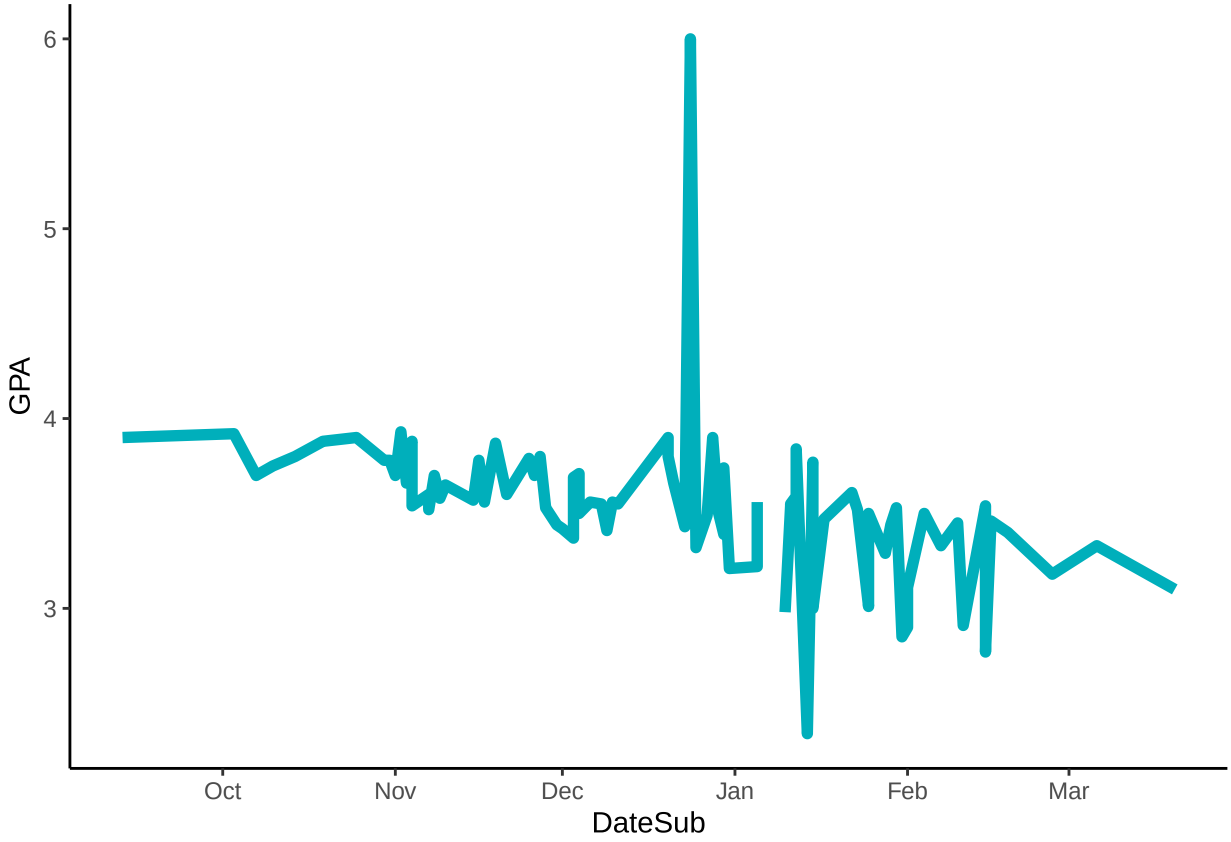
```
## The dates look ok - but the format is wrong and
## needs to be DATE
(MyData$DateSub <- as.Date(MyData$DateSub, "%m/%d/%Y") )
```

```
##  [1] "2020-01-11" "2020-01-11" "2020-01-12" "2019-11-07" "2019-11-21"
##  [6] "2019-11-03" "2019-11-08" "2019-10-07" "2019-10-10" "2020-01-15"
## [11] "2019-10-31" "2019-10-30" "2019-10-14" "2019-11-04" "2019-12-20"
## [16] "2019-10-25" "2019-12-28" "2020-01-10" "2020-01-14" "2020-01-31"
## [21] "2020-01-10" "2020-01-25" "2020-02-27" "2019-12-31" "2020-03-06"
## [26] "2020-02-07" "2019-12-03" "2019-11-30" "2020-01-12" "2020-02-15"
## [31] "2019-12-10" "2020-01-22" "2019-12-04" "2019-11-25" "2020-01-12"
## [36] "2019-12-30" "2020-02-19" "2019-12-09" "2019-12-23" "2020-01-29"
## [41] "2020-02-16" "2020-01-17" "2019-12-27" "2020-02-04" "2019-12-04"
## [46] "2020-01-23" "2020-01-30" "2019-11-28" "2019-11-04" "2019-12-08"
## [51] "2019-12-11" "2019-12-06" "2019-11-17" "2019-11-15" "2019-11-09"
## [56] "2020-01-25" "2019-11-10" "2019-12-21" "2019-12-03" "2019-11-26"
## [61] "2019-11-01" "2019-11-16" "2019-12-20" "2019-11-27" "2019-11-19"
## [66] "2019-10-19" "2019-09-13" "2019-10-03" "2019-11-02" "2019-12-24"
## [71] "2020-02-15" "2020-02-01" "2020-02-11" "2020-01-15" "2020-03-20"
## [76] "2020-02-01" "2020-01-05" "2019-12-25" "2020-01-05" "2019-12-30"
## [81] "2020-01-28" "2019-12-01" "2020-02-10" "2019-12-29" "2019-11-07"
```

```
str(MyData$DateSub)
```

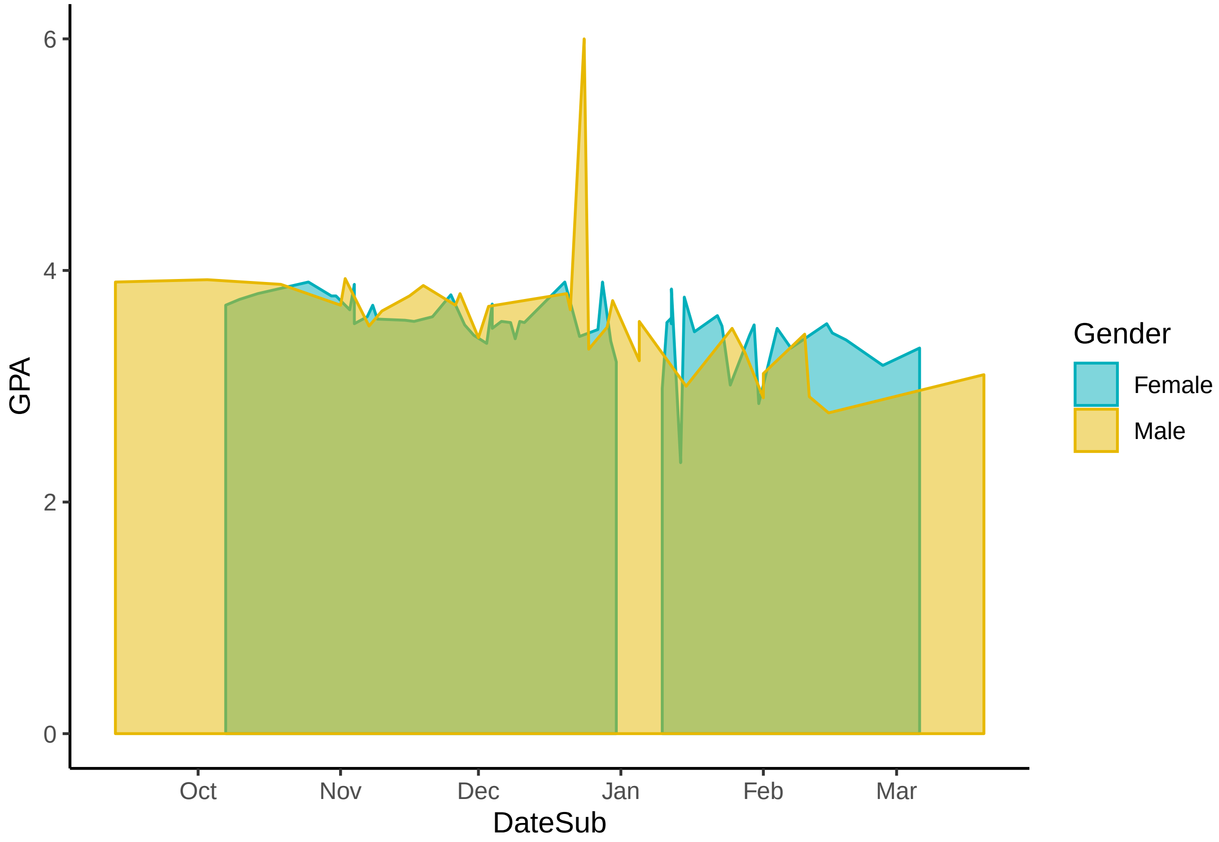
```
##  Date[1:85], format: "2020-01-11" "2020-01-11" "2020-01-12" "2019-11-07" "2019-11-21" ...
```

```
## NOw that we have dates, can visualize them with
## a time series vis option.

ggplot(data = MyData, aes(x = DateSub, y = GPA))+
  geom_line(color = "#00AFBB", size = 2)
```



GPA ... above 4.0 .... ?

```
## We have a problem!
## The GPA should never be above 4.0.

ggplot(MyData, aes(x = DateSub, y = GPA)) +
  geom_area(aes(color = Gender, fill = Gender),
            alpha = 0.5, position = position_dodge(0.8)) +
  scale_color_manual(values = c("#00AFBB", "#E7B800")) +
  scale_fill_manual(values = c("#00AFBB", "#E7B800"))
```



```
## We can already SEE many things.
## We can see that Males applied a bit early and a bit later.
## We can see that we have an error in at least one GPA
## value that we will need to fix.
## We can see that Female and Male application times and GPAs
## do not appear sig diff - but we can investigate this further.
```

### Let's look at GPA and then dates with it

```
str(MyData$GPA)
```

```
##  num [1:85] 3.54 3.55 3.59 3.6 3.6 3.66 3.7 3.7 3.75 3.77 ...
```

```
MyData$GPA<-as.numeric(MyData$GPA)
table(MyData$GPA)
```

```
##
## 2.34 2.77 2.85  2.9 2.91 2.98    3 3.01  3.1 3.11 3.18 3.21 3.22 3.29 3.32 3.33
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    2
## 3.37 3.39  3.4 3.41 3.42 3.43 3.44 3.45 3.46 3.47 3.49  3.5 3.51 3.52 3.53 3.54
##    1    1    1    1    1    1    2    1    1    1    1    3    1    2    2    4
## 3.55 3.56 3.57 3.58 3.59  3.6 3.61 3.65 3.66 3.69  3.7 3.71 3.74 3.75 3.77 3.78
##    3    4    1    1    1    2    1    1    2    1    4    1    1    1    1    3
## 3.79  3.8 3.84 3.87 3.88  3.9 3.92 3.93    6
##    1    3    1    1    2    4    1    1    1
```

```
## Are there NAs?
(sum(is.na(MyData$GPA)))
```

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

```
## Fix the missing GPA first
## Find it
(MissingGPA <- MyData[is.na(MyData$GPA),])
```

```
##    Decision Gender    DateSub      State GPA WorkExp TestScore WritingScore
## 18    Admit Female 2020-01-10 California  NA     2.8       967           95
##    VolunteerLevel
## 18              3
```

```
## OK - its a Female/Admit. We can replace the missing GPA
## with the median of all Female Admits.
(Temp<-MyData[MyData$Decision=="Admit" & MyData$Gender=="Female",])
```

```
##    Decision Gender    DateSub      State  GPA WorkExp TestScore WritingScore
## 1     Admit Female 2020-01-11    Florida 3.54     0.7       965           11
## 2     Admit Female 2020-01-11    Florida 3.55     0.0       962           97
## 3     Admit Female 2020-01-12   Colorado 3.59     1.7       969           93
## 4     Admit Female 2019-11-07   Colorado 3.60     0.9       969           97
## 5     Admit Female 2019-11-21   Colorado 3.60     1.2       967           94
## 6     Admit Female 2019-11-03 California 3.66     0.9       956           89
## 7     Admit Female 2019-11-08 California 3.70     1.2       969           94
## 8     Admit Female 2019-10-07 California 3.70     2.7       799           97
## 9     Admit Female 2019-10-10   Colorado 3.75     1.1       969           93
## 10    Admit Female 2020-01-15    Florida 3.77     1.4       969           99
## 11    Admit Female 2019-10-31 California 3.78     8.7       966           91
## 12    Admit Female 2019-10-30       Utah 3.78     1.2       968           87
## 13    Admit Female 2019-10-14    Florida 3.80     1.9       965           94
## 14    Admit Female 2019-11-04   Colorado 3.88     1.0       969           93
## 15    Admit Female 2019-12-20    Florida 3.90     4.7       961           93
## 16    Admit Female 2019-10-25   Colorado 3.90     3.8       967           98
## 17    Admit Female 2019-12-28    Florida 3.90     0.0       967           88
## 18    Admit Female 2020-01-10 California   NA     2.8       967           95
##    VolunteerLevel
## 1               1
## 2               0
## 3               0
## 4               2
## 5               2
## 6               1
## 7               2
## 8               5
## 9                0
## 10               4
## 11               2
## 12               2
## 13               5
## 14               4
## 15               1
## 16               3
## 17               0
## 18               3
```

```
## The median for Female Admits is:
(MyMed<-median(Temp$GPA, na.rm=TRUE))
```

```
## [1] 3.75
```

```
## NOW - replace the missing GPA with this Median
MyData$GPA[is.na(MyData$GPA)] <- MyMed
## Check to assure the missing value  was updated...
(sum(is.na(MyData$GPA)))
```

```
## [1] 0
```

```
table(MyData$GPA)
```

```
## 
## 2.34 2.77 2.85  2.9 2.91 2.98    3 3.01  3.1 3.11 3.18 3.21 3.22 3.29 3.32 3.33 
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    2 
## 3.37 3.39  3.4 3.41 3.42 3.43 3.44 3.45 3.46 3.47 3.49  3.5 3.51 3.52 3.53 3.54 
##    1    1    1    1    1    1    2    1    1    1    1    3    1    2    2    4 
## 3.55 3.56 3.57 3.58 3.59  3.6 3.61 3.65 3.66 3.69  3.7 3.71 3.74 3.75 3.77 3.78 
##    3    4    1    1    1    2    1    1    2    1    4    1    1    2    1    3 
## 3.79  3.8 3.84 3.87 3.88  3.9 3.92 3.93    6 
##    1    3    1    1    2    4    1    1    1
```

Well – the dilema faced by data scientists everywhere . . . what to do with missing data?!? Its common to either remove the row (as we have done previously); or we can try to replace the value with an estimate – like the mean or median estimate.

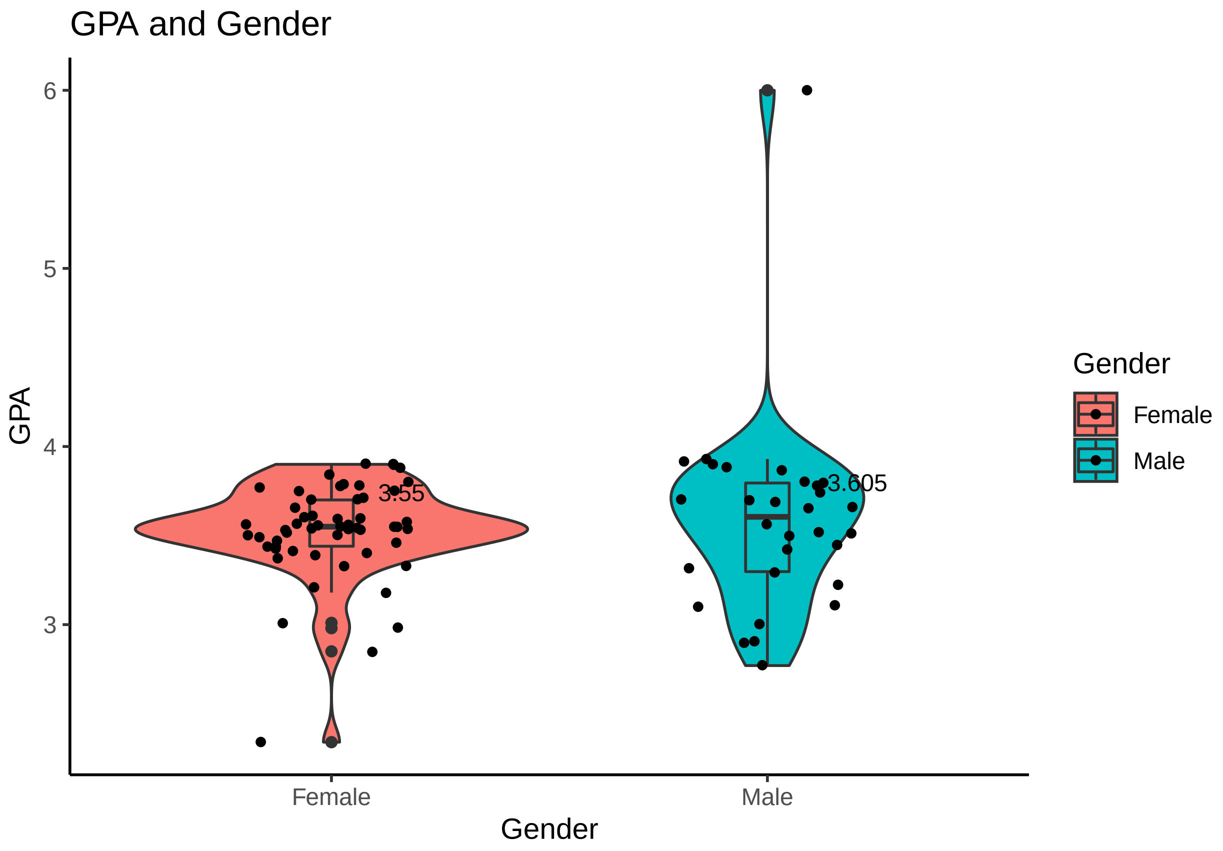
```
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 3.5.3
```

```
## Create a table using the dataset
## This table is BY Gender
## The method is summarize
## A new column is med and is the median for GPA
(TEMPmeds <- ddply(MyData, .(Gender), summarize,
                   med = median(GPA)))
```

```
##   Gender   med
## 1 Female 3.550
## 2   Male 3.605
```

```
## Next, we have an incorrect value....let's SEE IT

(MyV1 <- ggplot(MyData, aes(x=Gender, y=GPA, fill=Gender)) +
    geom_violin(trim=TRUE)+ geom_boxplot(width=0.1)+
    geom_text(data = TEMPmeds,
              aes(x = Gender, y = med, label = med),
              size = 3, vjust = -1.5,hjust=-1)+
    ggtitle("GPA and Gender")+
    geom_jitter(shape=16, position=position_jitter(0.2)))
```



```
## Now we can SEE the issue. There is at least one GPA
## that is out of range. Let's fix this.
## Let's replace the missing GPA by finding the median
## for the ADMITS in that Gender group

## FIND the row with GPA > 4
(WrongGPAs <- MyData[(MyData$GPA<0 | MyData$GPA >4),])
```

```
##    Decision Gender    DateSub    State GPA WorkExp TestScore WritingScore
## 72    Admit   Male 2019-12-24 Colorado   6     0.8       969           93
##    VolunteerLevel
## 72              1
```

```
##
## We have Male Admit with a GPA of 6.

## Fix it by using Male Admit GPA Median
(Temp<-MyData[MyData$Decision=="Admit" & MyData$Gender=="Male",])
```

```
##    Decision Gender    DateSub      State  GPA WorkExp TestScore WritingScore
## 58    Admit   Male 2020-01-25    Florida 3.50     0.7       965           91
## 59    Admit   Male 2019-11-10   Colorado 3.65     1.7       963           90
## 60    Admit   Male 2019-12-21    Florida 3.66     2.2       967           91
## 61    Admit   Male 2019-12-03 California 3.69     3.2       967           93
## 62    Admit   Male 2019-11-26 California 3.70     1.4       966           94
## 63    Admit   Male 2019-11-01    Florida 3.70     3.7       969           99
## 64    Admit   Male 2019-11-16   Colorado 3.78     1.2       966            1
## 65    Admit   Male 2019-12-20    Florida 3.80     1.4       969           97
## 66    Admit   Male 2019-11-27    Florida 3.80     1.7       968           91
## 67    Admit   Male 2019-11-19 California 3.87     1.7       966           97
## 68    Admit   Male 2019-10-19 California 3.88     1.5       967           95
## 69    Admit   Male 2019-09-13 California 3.90     6.7       962          100
## 70    Admit   Male 2019-10-03   Colorado 3.92     1.2       969           95
## 71    Admit   Male 2019-11-02    Florida 3.93     0.8       969           99
## 72    Admit   Male 2019-12-24   Colorado 6.00     0.8       969           93
##    VolunteerLevel
## 58              1
## 59              1
## 60              2
## 61              3
## 62              0
## 63              2
## 64              4
## 65              4
## 66              3
## 67              5
## 68              5
## 69              0
## 70              3
## 71              4
## 72              1
```

```
## The median for Male Admits is:
(MyMed<-median(Temp$GPA, na.rm=TRUE))
```

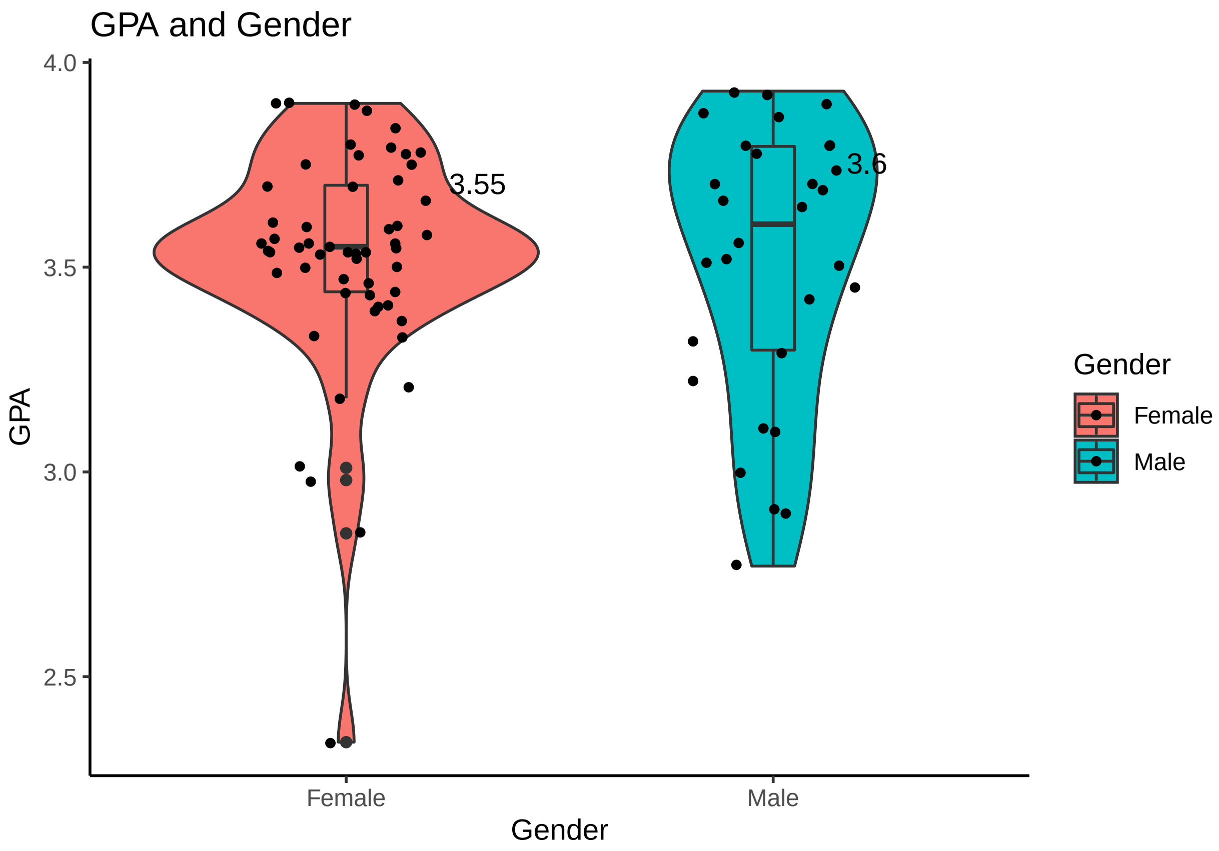
```
## [1] 3.8
```

```
## NOW - replace the missing GPA with this Median
MyData$GPA[MyData$GPA>4] <- MyMed

## NOW VISUALIZAE IT AGAIN:
(TEMPmeds <- ddply(MyData, .(Gender), summarize,
                   med = round(median(GPA),2)))
```

```
##   Gender  med
## 1 Female 3.55
## 2   Male 3.60
```

Fix it!!

```
(MyV1 <- ggplot(MyData, aes(x=Gender, y=GPA, fill=Gender)) +
    geom_violin(trim=TRUE)+ geom_boxplot(width=0.1)+
    geom_text(data = TEMPmeds,
              aes(x = Gender, y = med, label = med),
              size = 4, vjust = -2.5,hjust=-1.8)+
    ggtitle("GPA and Gender")+
    geom_jitter(shape=16, position=position_jitter(0.2)))
```



```
## That's better!

table(MyData$GPA)
```

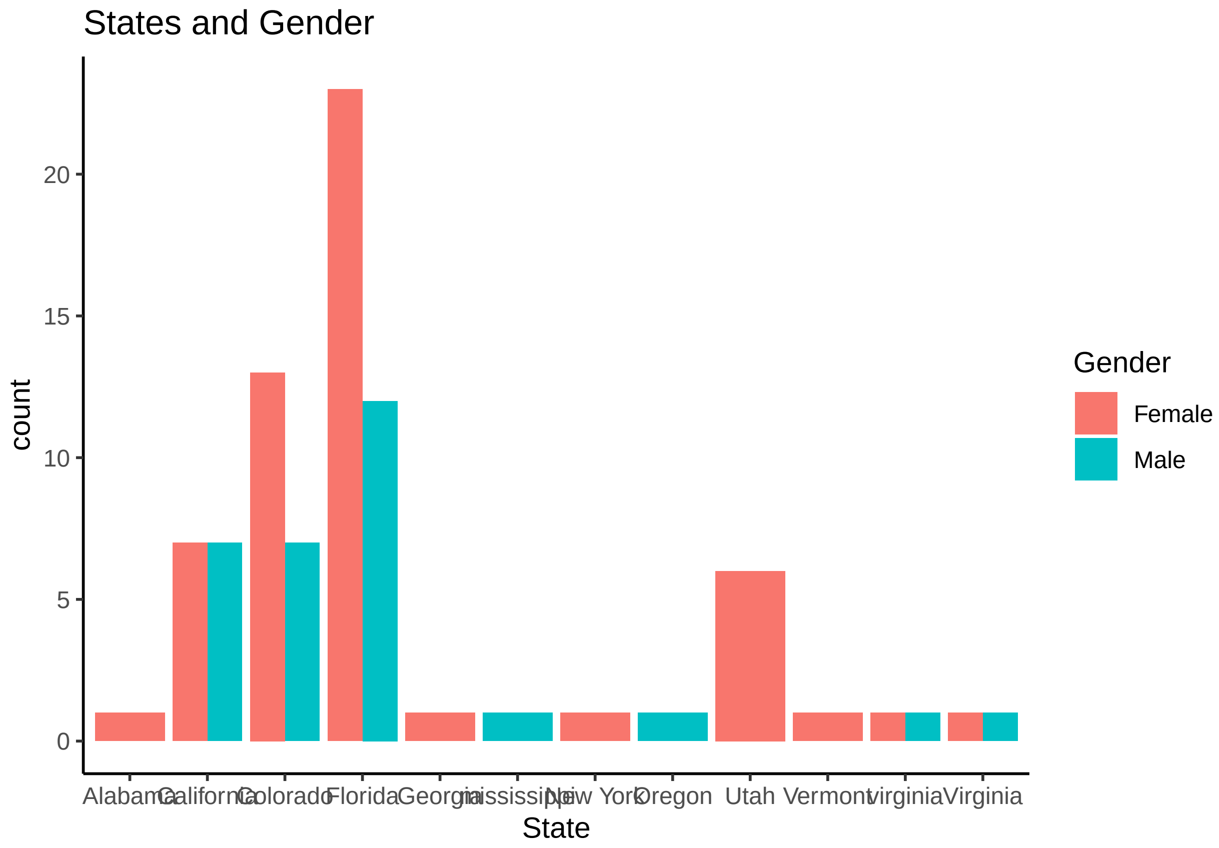
```
##
## 2.34 2.77 2.85  2.9 2.91 2.98    3 3.01  3.1 3.11 3.18 3.21 3.22 3.29 3.32 3.33
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    2
## 3.37 3.39  3.4 3.41 3.42 3.43 3.44 3.45 3.46 3.47 3.49  3.5 3.51 3.52 3.53 3.54
##    1    1    1    1    1    1    2    1    1    1    1    3    1    2    2    4
## 3.55 3.56 3.57 3.58 3.59  3.6 3.61 3.65 3.66 3.69  3.7 3.71 3.74 3.75 3.77 3.78
##    3    4    1    1    1    2    1    1    2    1    4    1    1    2    1    3
## 3.79  3.8 3.84 3.87 3.88  3.9 3.92 3.93
##    1    4    1    1    2    4    1    1
## LOOKS GOOD!
```

State is next

```
#############################################
##
##            Let's look at State next
############################################
#names(MyData)
str(MyData$State)
```

```
##  Factor w/ 12 levels "Alabama","California",..: 4 4 3 3 3 2 2 2 3 4 ...
```

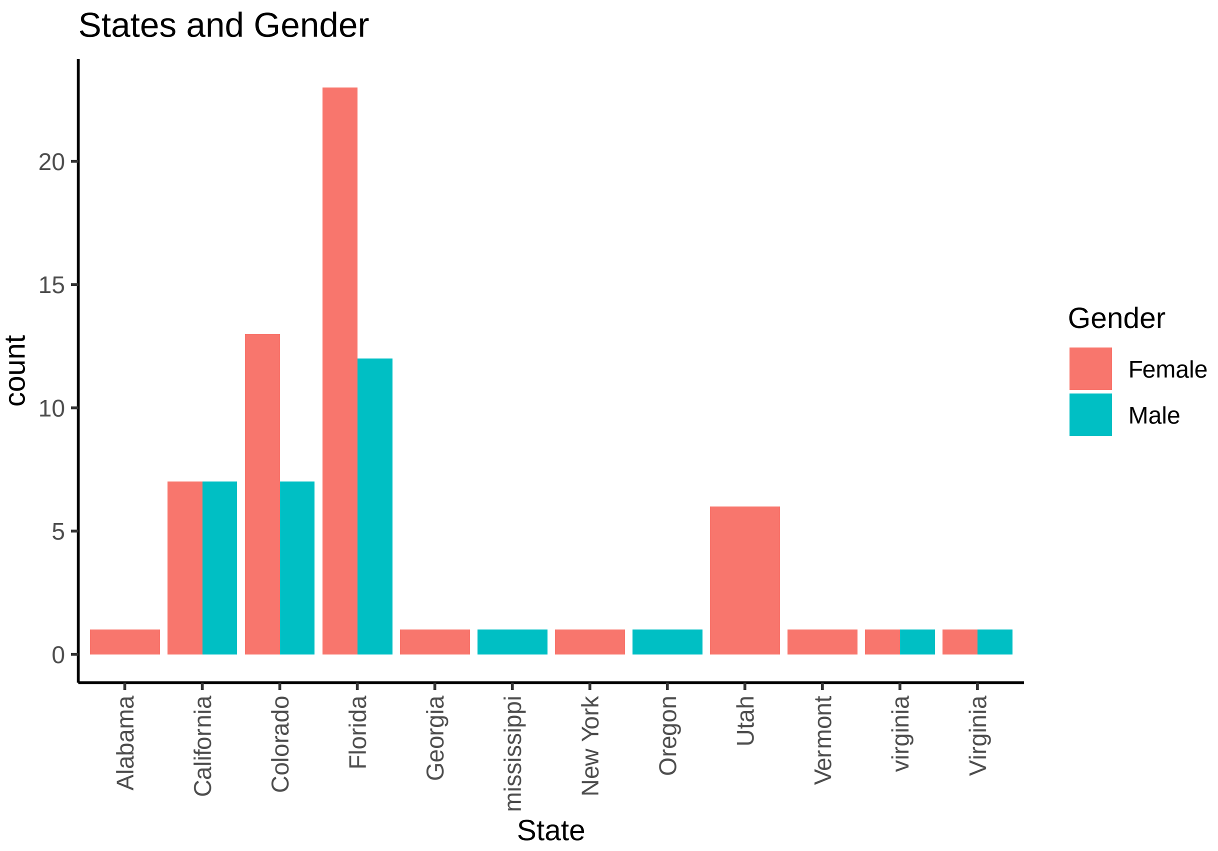
```
## Let's use a BAR to look
BaseGraph <- ggplot(MyData)
(MyG3<-BaseGraph +
    geom_bar(aes(State, fill = Gender), position="dodge")+
    ggtitle("States and Gender"))
```



```
## UGLY!!
```

This graph is not very aethestically pleasing . . . lets clean it up using “theme”s.

```
## Let's make this nicer so we can READ THE X AXIS
(MyG3<-BaseGraph +
    geom_bar(aes(State, fill = Gender), position="dodge")+
    ggtitle("States and Gender")+
    theme(axis.text.x=element_text(angle=90,hjust=1,vjust=0.5)))
```



```
## MUCH BETTER!
```

Now we can SEE that we have problems :( First, we have poor balance. It might be needed to collect all the lower count states, such as ALabama, Mississippi, etc. into a group called OTHER. However, we will not do this here. If you want to see how - look at this other tutorial http://drgates.georgetown.domains/SummerClassificationRMarkdown.html

Also - We have two Virginias (really!?!) - we need to combine them:

```
MyData$State[MyData$State == "virginia"] <- "Virginia"
table(MyData$State)
```

```
##
##     Alabama  California    Colorado     Florida     Georgia mississippi
##           1          14          20          35           1           1
##    New York      Oregon        Utah     Vermont    virginia    Virginia
##           1           1           6           1           0           4
## Now - we need to remove the level of virginia
MyData$State<-as.character(MyData$State)
table(MyData$State)
```

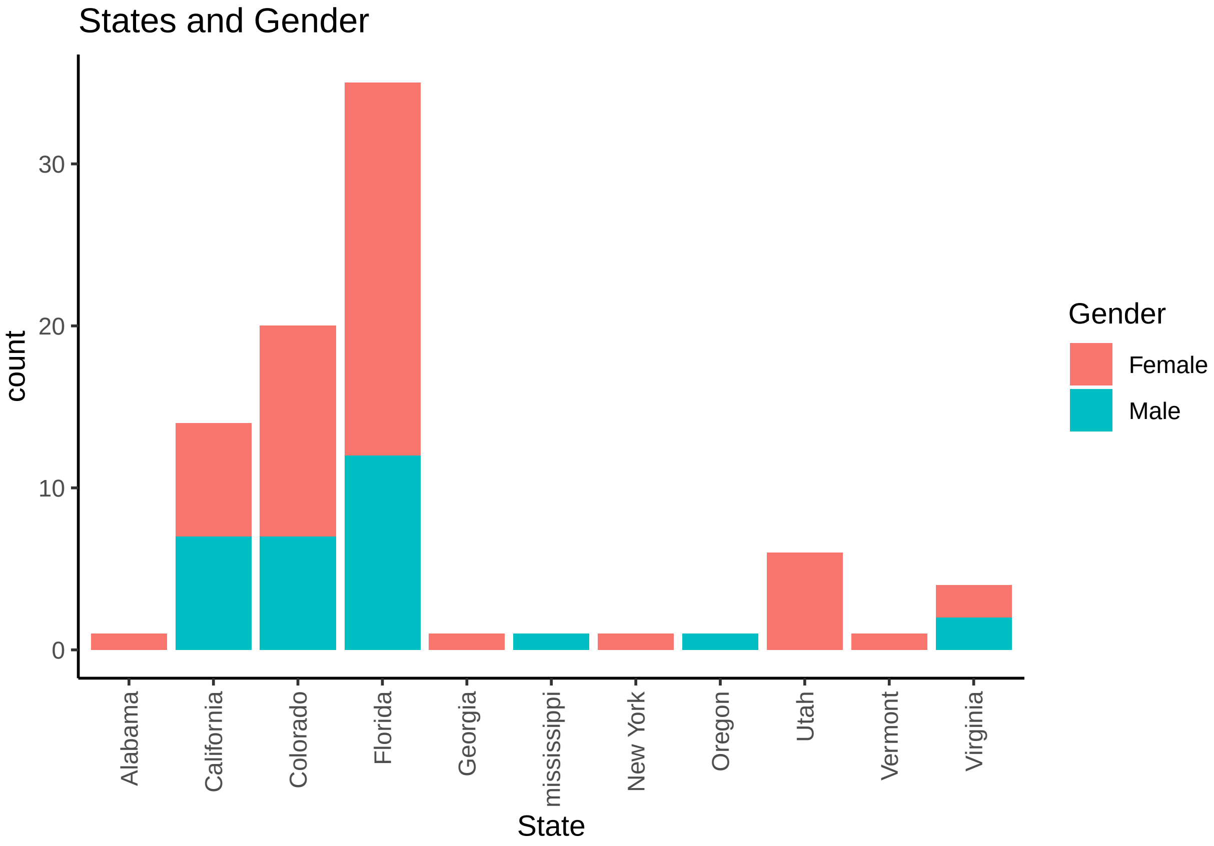
```
##
##      Alabama  California    Colorado     Florida     Georgia mississippi
##            1          14          20          35           1           1
##     New York      Oregon        Utah     Vermont    Virginia
##            1           1           6           1           4
```

```
MyData$State<-as.factor(MyData$State)
str(MyData$State)
```

```
##  Factor w/ 11 levels "Alabama","California",..: 4 4 3 3 3 2 2 2 3 4 ...
```

and confirm

```
## Check it
(MyG4<-ggplot(MyData) +
    geom_bar(aes(State, fill = Gender), position="stack")+
    ggtitle("States and Gender")+
    theme(axis.text.x=element_text(angle=90,hjust=1,vjust=0.5)))
```



Next: WorkExp

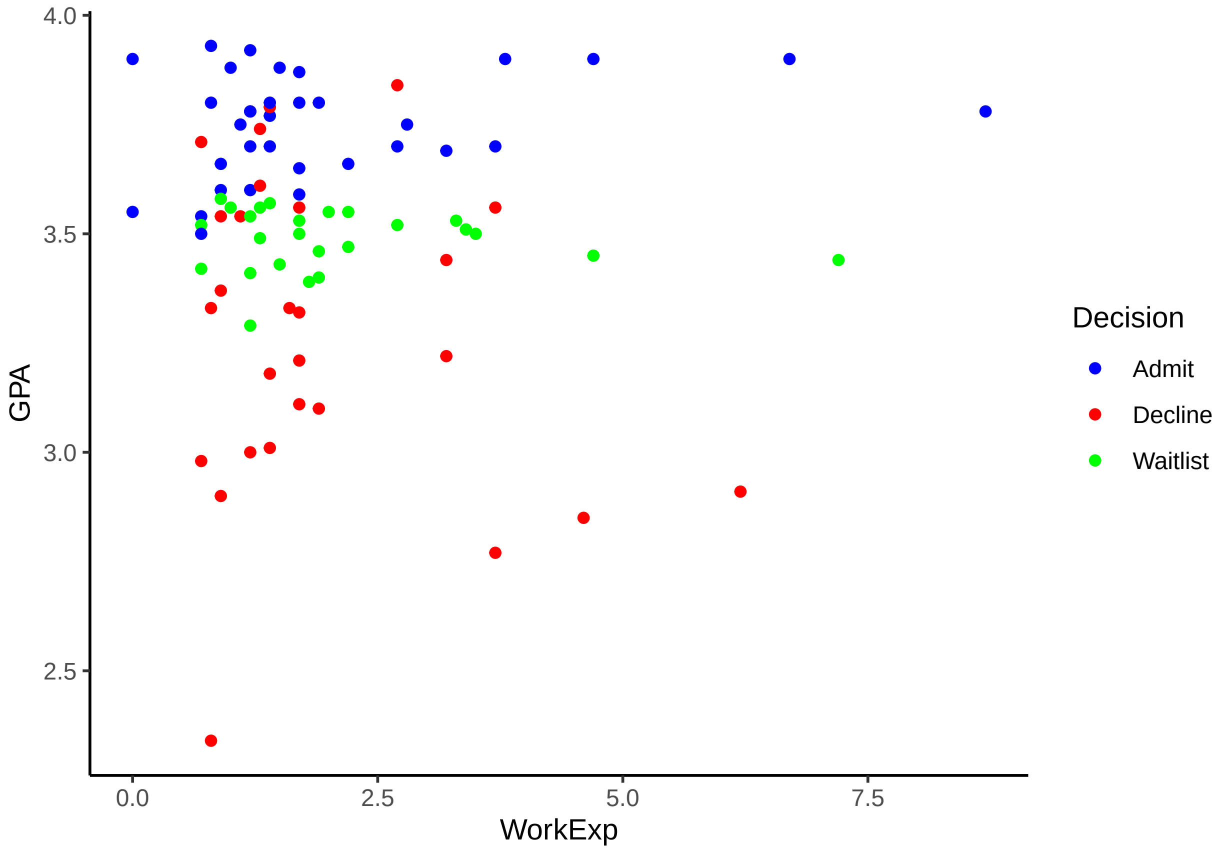
```
## Even better!

###########################################
##
## Now let's look at WorkExp
#########################################
#names(MyData)
(sum(is.na(MyData$WorkExp)))
```

```
## [1] 0
```

```
str(MyData$WorkExp)
```

```
##  num [1:85] 0.7 0 1.7 0.9 1.2 0.9 1.2 2.7 1.1 1.4 ...
```

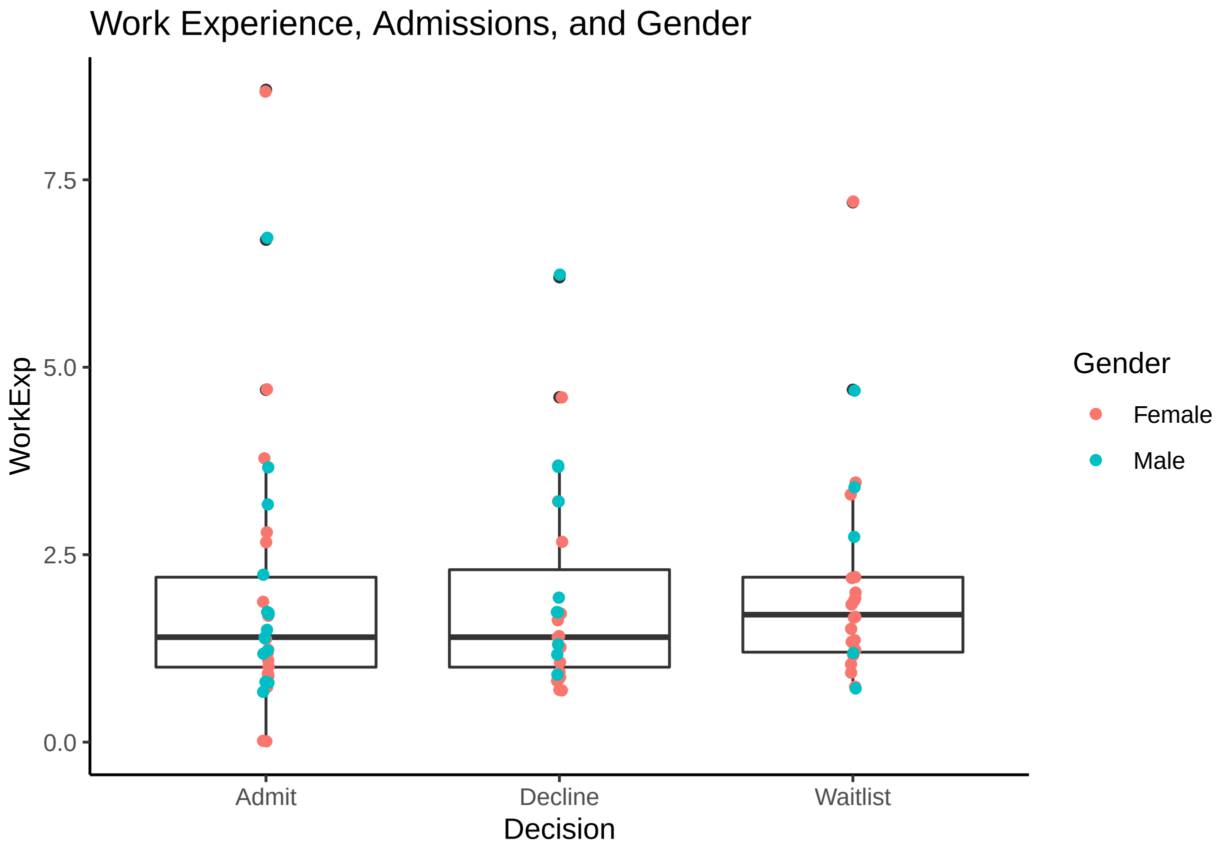
```
## Let's look
theme_set(theme_classic())

# Histogram on a Continuous (Numeric) Variable
(MyS3 <- ggplot(MyData,aes(x=WorkExp, y=GPA, color=Decision)) +
    geom_point() +
    scale_color_manual(values = c('blue',"red", "green")))
```



```
## This helps in many ways. We can see that we have no outliers
## or odd values.
```

However, let’s check it with a box plot as well.

```
(MyL1<-ggplot(MyData, aes(x=Decision, y=WorkExp))+
    geom_boxplot()+
    geom_jitter(position=position_jitter(.01), aes(color=Gender))+
    ggtitle("Work Experience, Admissions, and Gender"))
```



This looks good and it also starts to tell us that people were not penalized or prefered based on work experience.

Lets move on to TestScore and WritingScore.

```
######################################################
##
##           Let's look at TestScore and Writing Score
##
########################################################
(sum(is.na(MyData$TestScore)))
```

```
## [1] 0
```

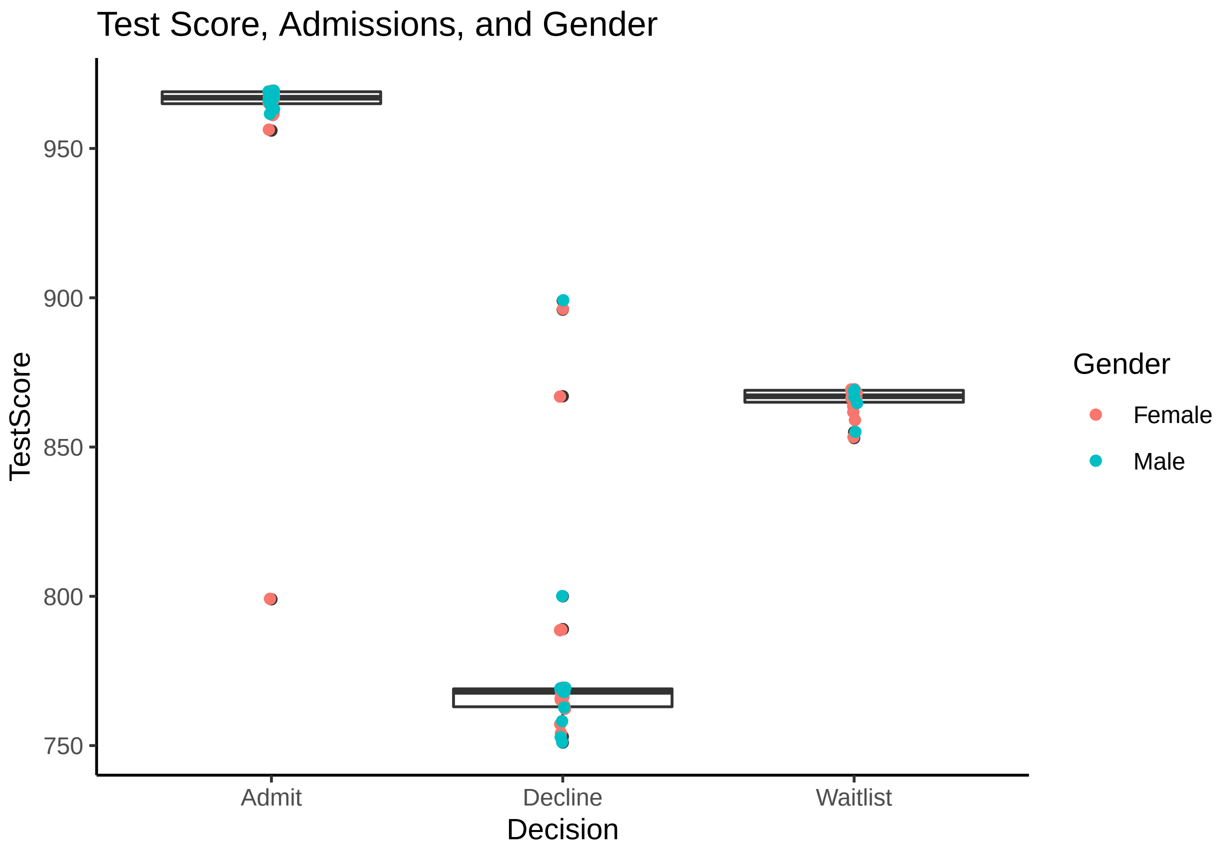
```r
(sum(is.na(MyData$WritingScore)))
```

```
## [1] 0
```

```r
str(MyData)
```

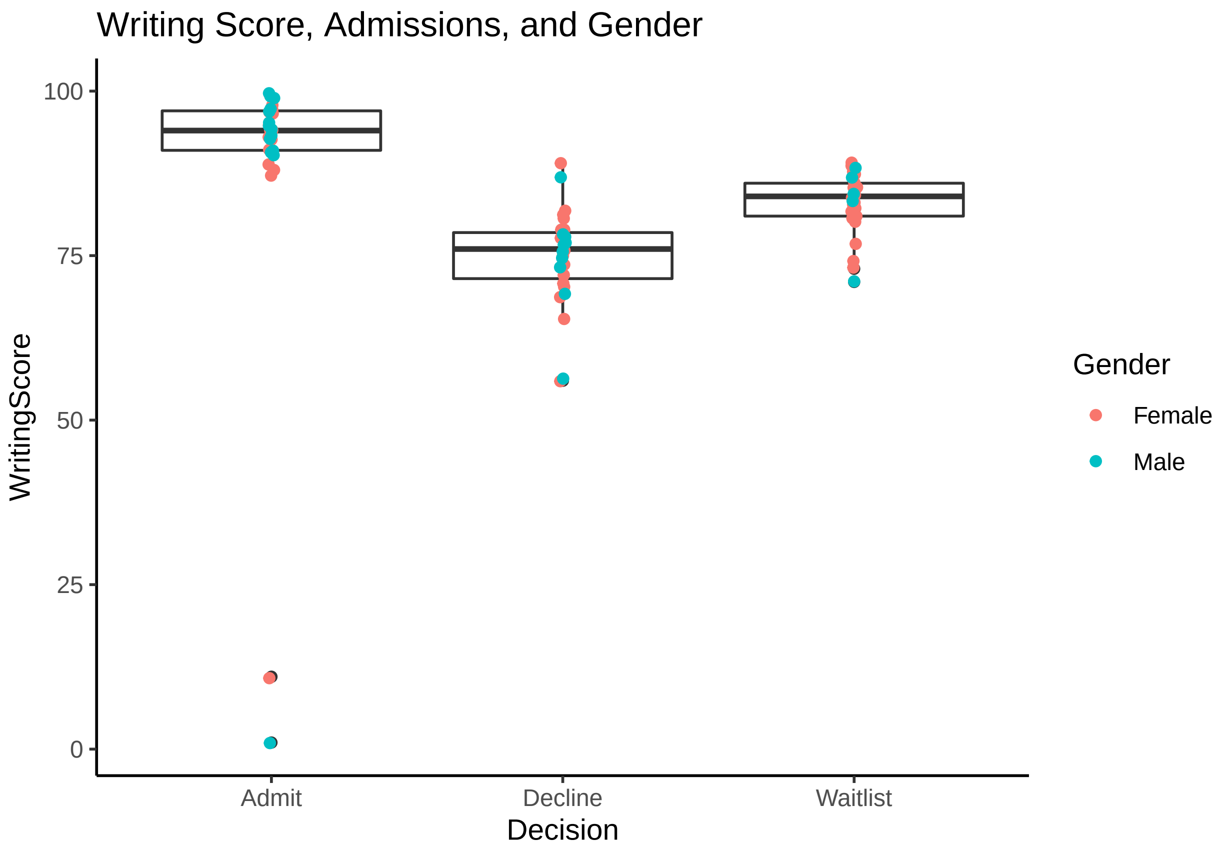
```
## 'data.frame':    85 obs. of  9 variables:
##  $ Decision      : Factor w/ 3 levels "Admit","Decline",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Gender        : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1 1 1 1 1 ...
##  $ DateSub       : Date, format: "2020-01-11" "2020-01-11" ...
##  $ State         : Factor w/ 11 levels "Alabama","California",..: 4 4 3 3 3 2 2 2 3 4 ...
##  $ GPA           : num  3.54 3.55 3.59 3.6 3.6 3.66 3.7 3.7 3.75 3.77 ...
##  $ WorkExp       : num  0.7 0 1.7 0.9 1.2 0.9 1.2 2.7 1.1 1.4 ...
##  $ TestScore     : int  965 962 969 969 967 956 969 799 969 969 ...
##  $ WritingScore  : int  11 97 93 97 94 89 94 97 93 99 ...
##  $ VolunteerLevel: int  1 0 0 2 2 1 2 5 0 4 ...
```

```r
## Box plots are great to look for odd values

(MyL2<-ggplot(MyData, aes(x=Decision, y=TestScore))+
    geom_boxplot()+
    geom_jitter(position=position_jitter(.01), aes(color=Gender))+
    ggtitle("Test Score, Admissions, and Gender"))
```



Interesting!! This mostly makes sense except for the 800 in the Admit group. However, it is not an outlier - it is just interesting.

```
(MyL3<-ggplot(MyData, aes(x=Decision, y=WritingScore))+
    geom_boxplot()+
    geom_jitter(position=position_jitter(.01), aes(color=Gender))+
    ggtitle("Writing Score, Admissions, and Gender"))
```



Hmmm - most of this looks OK, BUT, we have some very strangevalues for the Admit group. Let's look at these:

```
(Temp <- subset(MyData, Decision=="Admit",
                select=c(Decision,WritingScore)) )
```

```
##    Decision WritingScore
## 1     Admit           11
## 2     Admit           97
## 3     Admit           93
## 4     Admit           97
## 5     Admit           94
## 6     Admit           89
## 7     Admit           94
## 8     Admit           97
## 9     Admit           93
## 10    Admit           99
## 11    Admit           91
## 12    Admit           87
## 13    Admit           94
## 14    Admit           93
## 15    Admit           93
## 16    Admit           98
## 17    Admit           88
## 18    Admit           95
## 58    Admit           91
## 59    Admit           90
## 60    Admit           91
## 61    Admit           93
## 62    Admit           94
## 63    Admit           99
## 64    Admit            1
## 65    Admit           97
## 66    Admit           91
## 67    Admit           97
## 68    Admit           95
## 69    Admit          100
## 70    Admit           95
## 71    Admit           99
## 72    Admit           93
```

```
table(Temp$WritingScore)
```

```
## 
##   1  11  87  88  89  90  91  93  94  95  97  98  99 100 
##   1   1   1   1   1   1   4   6   4   3   5   1   3   1
```

OK - we can see that two score seem incorrect. The 1 and the 11, for an Admit, it not likely. Let's replace them with median

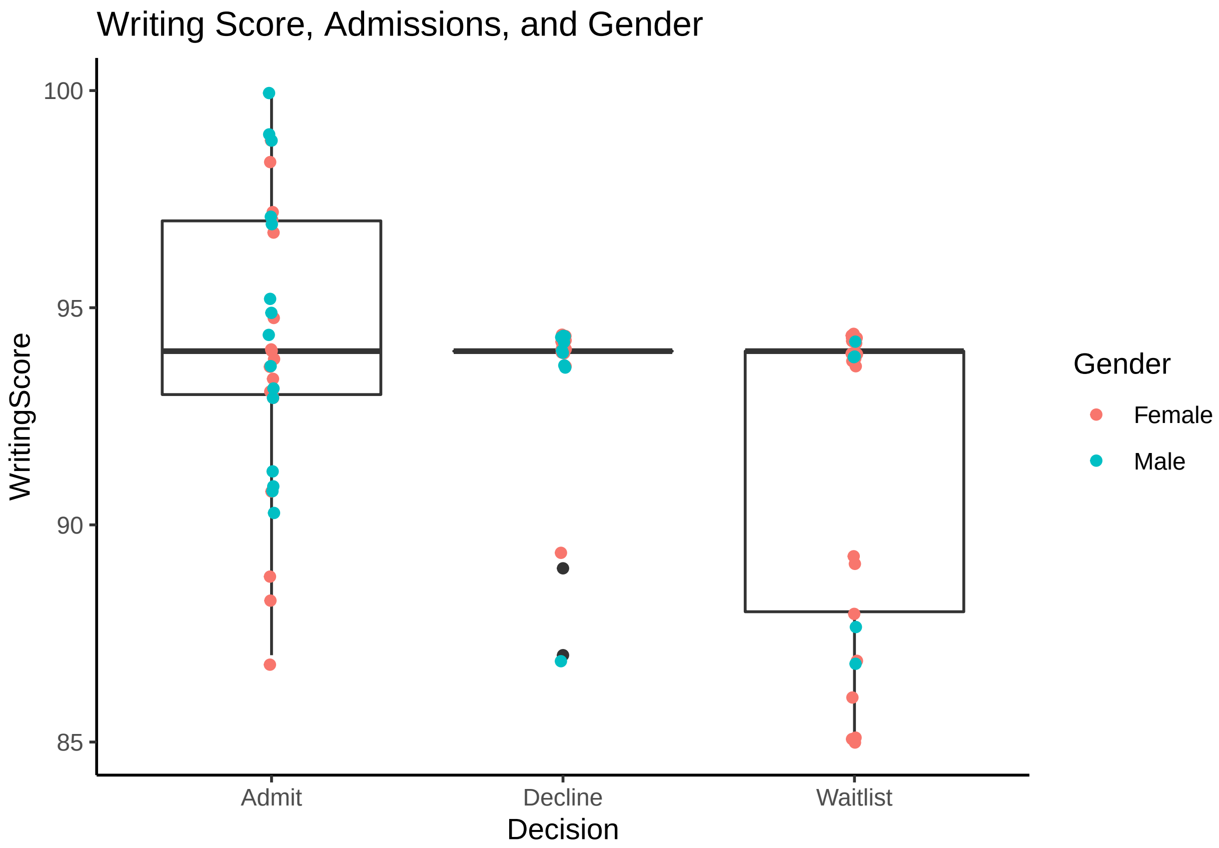
```
(Temp3<-MyData[MyData$Decision=="Admit",])
```

```
##    Decision Gender    DateSub      State  GPA WorkExp TestScore WritingScore
## 1     Admit Female 2020-01-11    Florida 3.54     0.7       965           11
## 2     Admit Female 2020-01-11    Florida 3.55     0.0       962           97
## 3     Admit Female 2020-01-12   Colorado 3.59     1.7       969           93
## 4     Admit Female 2019-11-07   Colorado 3.60     0.9       969           97
## 5     Admit Female 2019-11-21   Colorado 3.60     1.2       967           94
## 6     Admit Female 2019-11-03 California 3.66     0.9       956           89
## 7     Admit Female 2019-11-08 California 3.70     1.2       969           94
## 8     Admit Female 2019-10-07 California 3.70     2.7       799           97
## 9     Admit Female 2019-10-10   Colorado 3.75     1.1       969           93
## 10    Admit Female 2020-01-15    Florida 3.77     1.4       969           99
## 11    Admit Female 2019-10-31 California 3.78     8.7       966           91
## 12    Admit Female 2019-10-30       Utah 3.78     1.2       968           87
## 13    Admit Female 2019-10-14    Florida 3.80     1.9       965           94
## 14    Admit Female 2019-11-04   Colorado 3.88     1.0       969           93
## 15    Admit Female 2019-12-20    Florida 3.90     4.7       961           93
## 16    Admit Female 2019-10-25   Colorado 3.90     3.8       967           98
## 17    Admit Female 2019-12-28    Florida 3.90     0.0       967           88
## 18    Admit Female 2020-01-10 California 3.75     2.8       967            95
## 58    Admit   Male 2020-01-25    Florida 3.50     0.7       965            91
## 59    Admit   Male 2019-11-10   Colorado 3.65     1.7       963            90
## 60    Admit   Male 2019-12-21    Florida 3.66     2.2       967            91
## 61    Admit   Male 2019-12-03 California 3.69     3.2       967            93
## 62    Admit   Male 2019-11-26 California 3.70     1.4       966            94
## 63    Admit   Male 2019-11-01    Florida 3.70     3.7       969            99
## 64    Admit   Male 2019-11-16   Colorado 3.78     1.2       966             1
## 65    Admit   Male 2019-12-20    Florida 3.80     1.4       969            97
## 66    Admit   Male 2019-11-27    Florida 3.80     1.7       968            91
## 67    Admit   Male 2019-11-19 California 3.87     1.7       966            97
## 68    Admit   Male 2019-10-19 California 3.88     1.5       967            95
## 69    Admit   Male 2019-09-13 California 3.90     6.7       962           100
## 70    Admit   Male 2019-10-03   Colorado 3.92     1.2       969            95
## 71    Admit   Male 2019-11-02    Florida 3.93     0.8       969            99
## 72    Admit   Male 2019-12-24   Colorado 3.80     0.8       969            93
##    VolunteerLevel
## 1               1
## 2               0
## 3               0
## 4               2
## 5               2
## 6               1
## 7               2
## 8               5
## 9               0
## 10              4
## 11              2
## 12              2
## 13              5
## 14              4
## 15              1
## 16              3
## 17              0
## 18              3
## 58              1
## 59              1
## 60              2
## 61              3
## 62              0
## 63              2
## 64              4
## 65              4
## 66              3
## 67              5
## 68              5
## 69              0
## 70              3
## 71              4
## 72              1
```

```
## The median for Admits is:
(MyMed2<-median(Temp3$WritingScore, na.rm=TRUE))
```

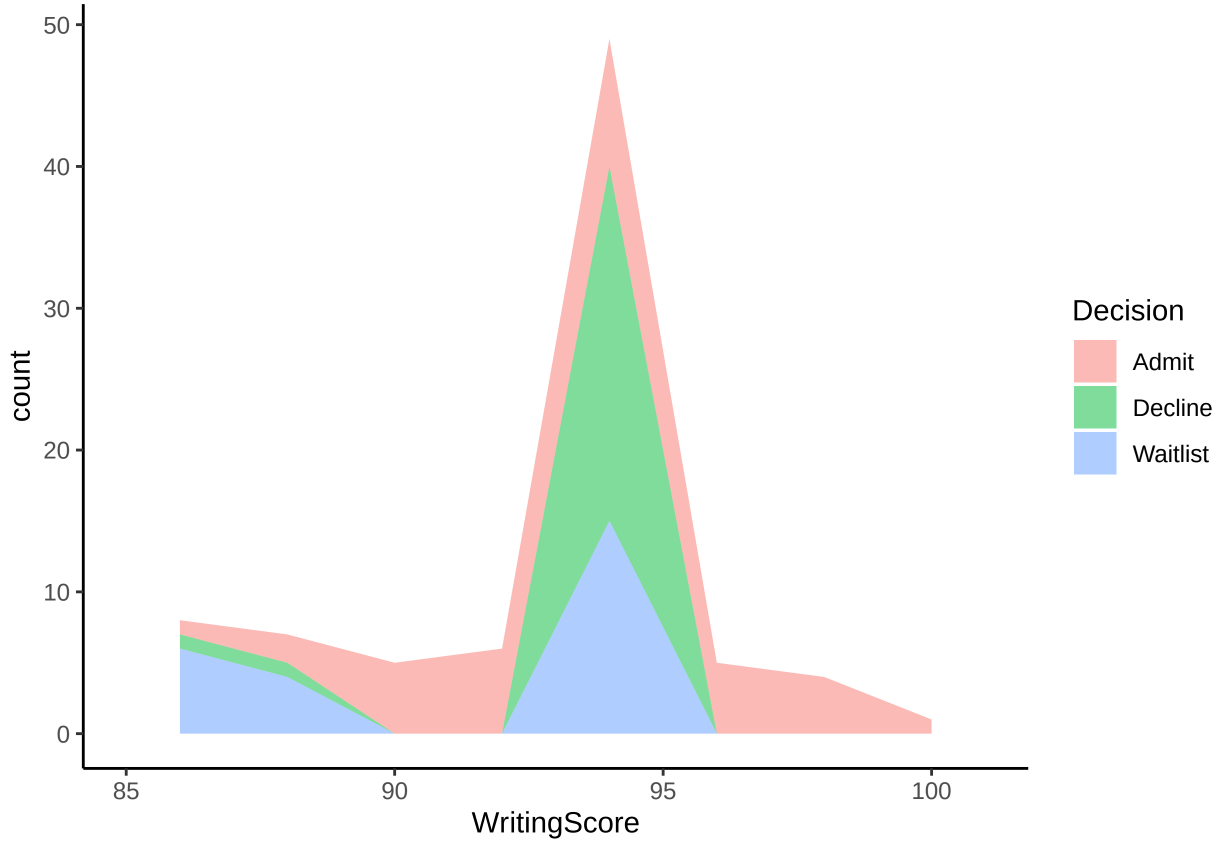
```
## [1] 94
```

```
## NOW - replace the incorrect  with this Median
MyData$WritingScore[MyData$WritingScore<85] <- MyMed2


## check again
(MyL4<-ggplot(MyData, aes(x=Decision, y=WritingScore))+
    geom_boxplot()+
    geom_jitter(position=position_jitter(.01), aes(color=Gender))+
    ggtitle("Writing Score, Admissions, and Gender"))
```

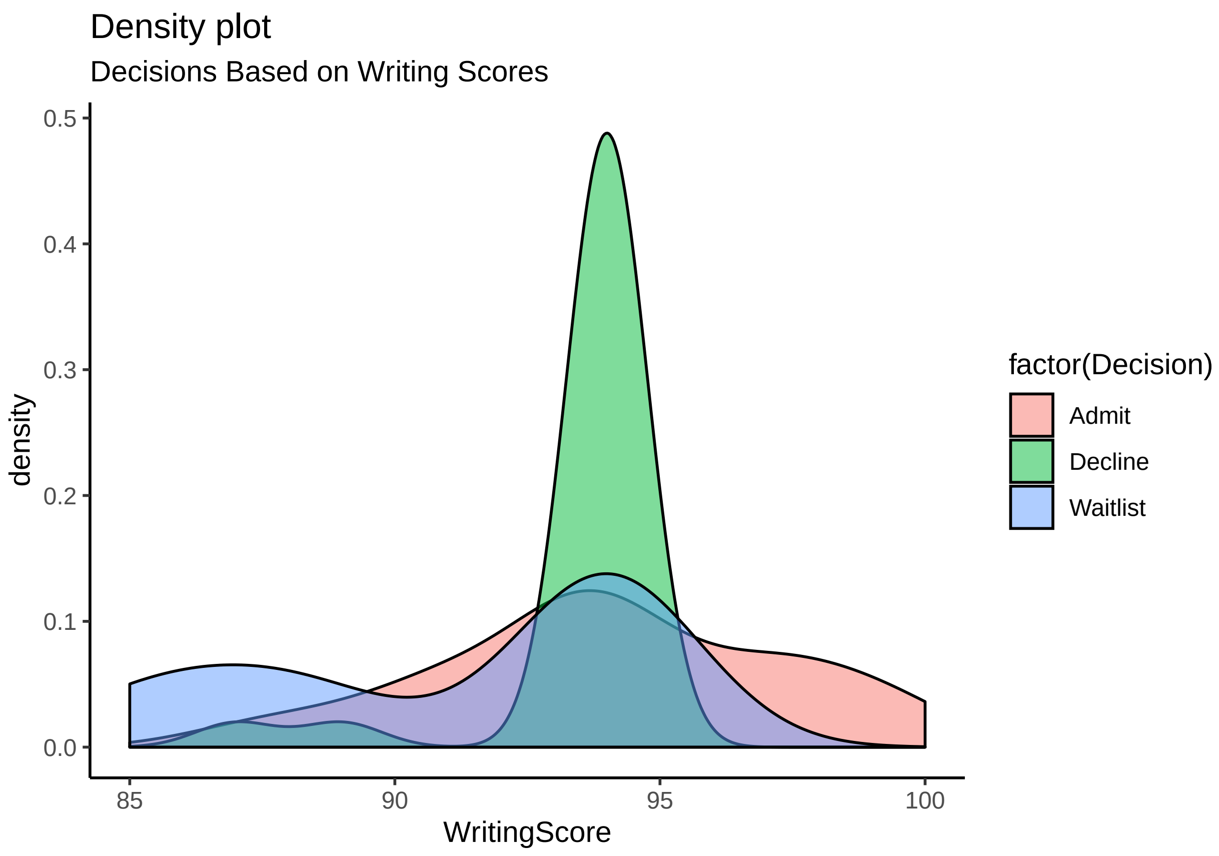


MUCH BETTER! We can also look using density area plots…

```
# Use semi-transparent fill
(MyPlot4<-ggplot(MyData, aes(x=WritingScore, fill=Decision)) +
    geom_area(stat ="bin", binwidth=2, alpha=0.5) +
    theme_classic())
```



```
## Here - using density - we can get a deeper look
MyPlot5 <- ggplot(MyData, aes(WritingScore))
MyPlot5 + geom_density(aes(fill=factor(Decision)), alpha=0.5) +
  labs(title="Density plot",
       subtitle="Decisions Based on Writing Scores")
```



## EDA

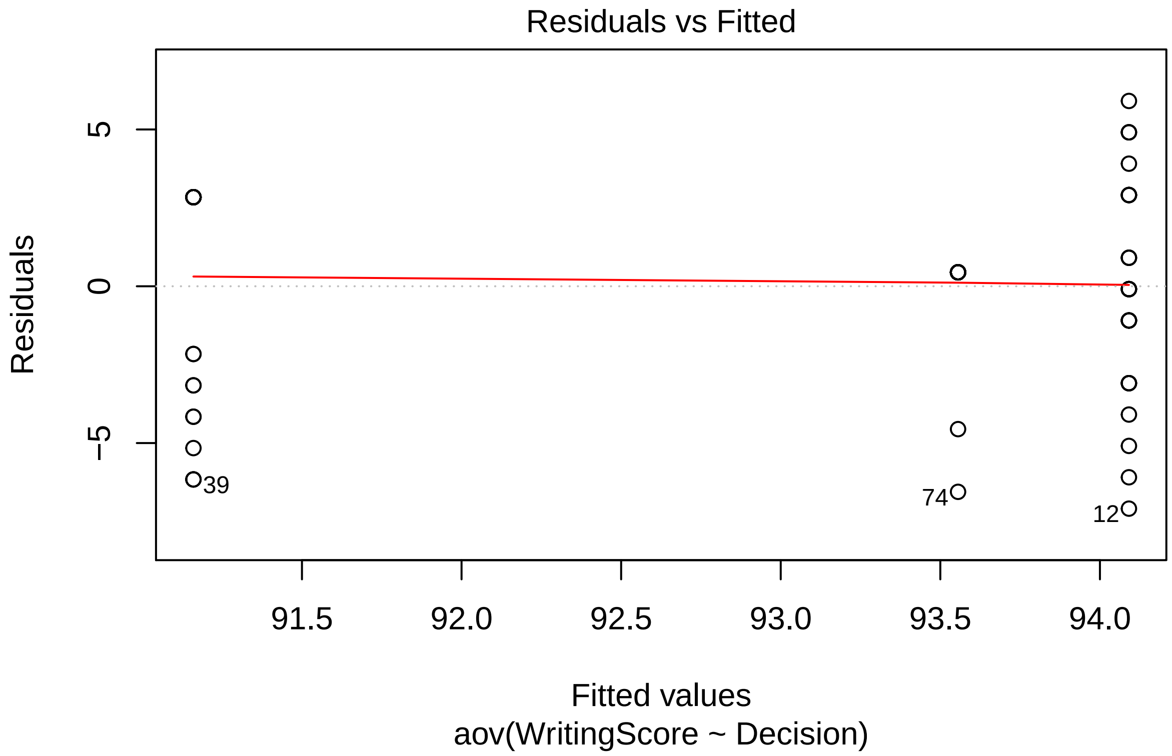
Let investigate some of these variables for associations with our dependent variable – EDA. Remember our goal is to leverage this data for prediction, decision-making, etc.

Does it seem like WritingScore is really related to Admissions?

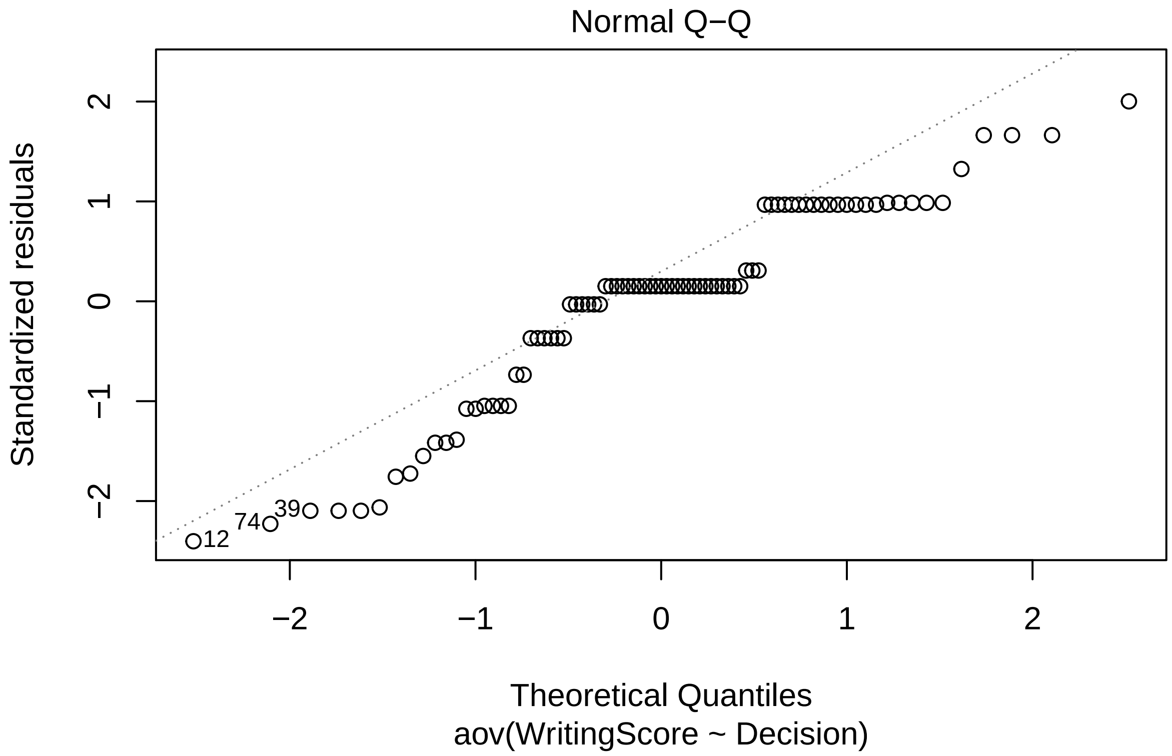
```
## Let's run an ANOVA test to see
MyANOVA_WS_Adm <- aov(WritingScore ~ Decision, data = MyData)
# Summary of the analysis
summary(MyANOVA_WS_Adm)  ## The test IS significant!
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## Decision     2  132.0   65.98   7.343 0.00117 **
## Residuals   82  736.8    8.98
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot(MyANOVA_WS_Adm, 1)
```



```
## The above shows we can assume the homogeneity of variances.
plot(MyANOVA_WS_Adm, 2) ## Close to normal
```



```
library("ggpubr")
```

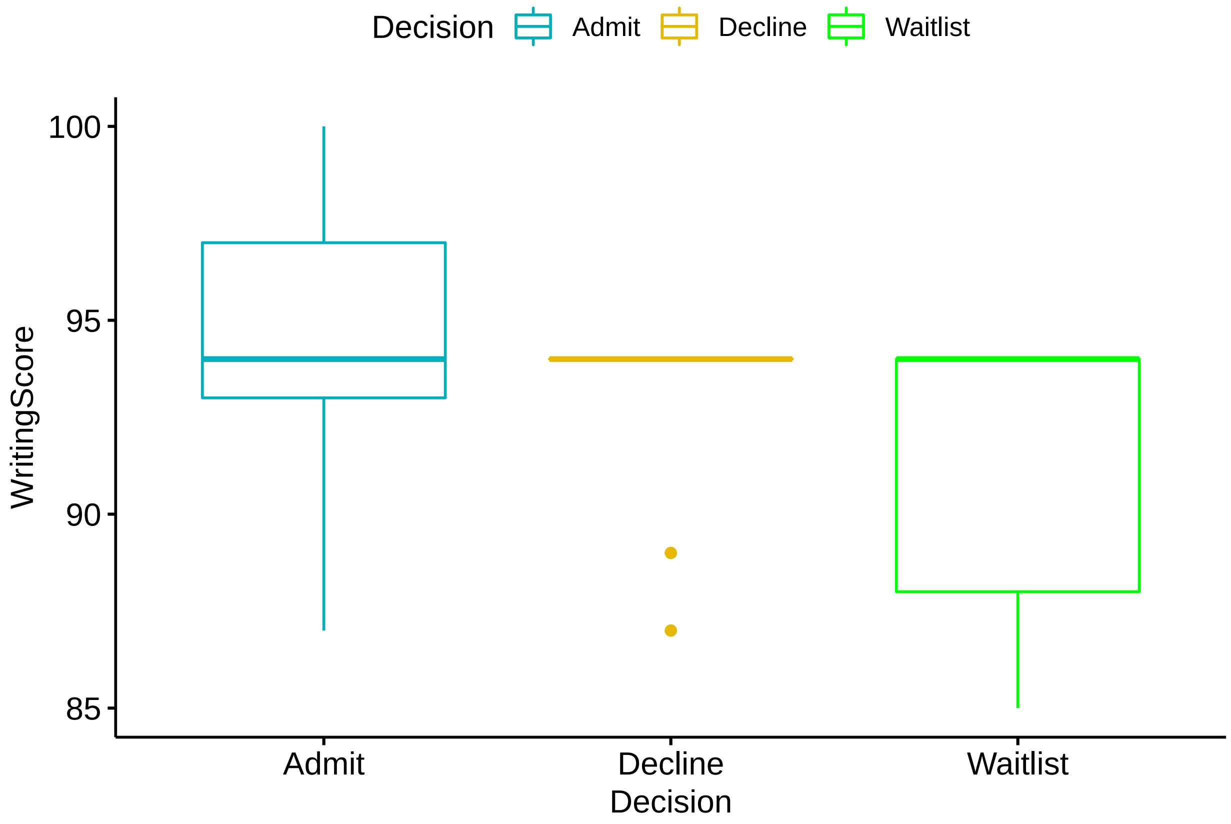
```
## Warning: package 'ggpubr' was built under R version 3.5.3
```

```
## Loading required package: magrittr
```

```
##
## Attaching package: 'ggpubr'
```

```
## The following object is masked from 'package:plyr':
##
##     mutate
```

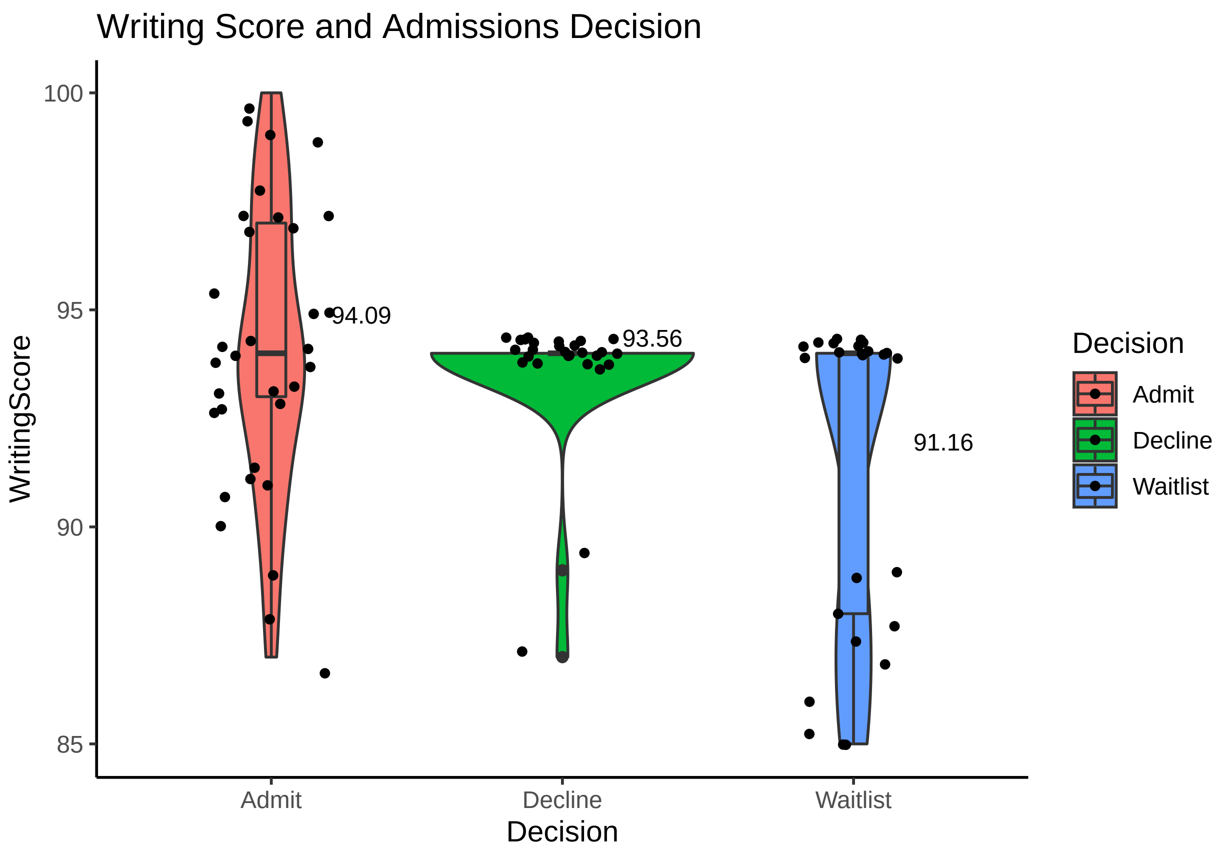
```
ggboxplot(MyData, x = "Decision", y = "WritingScore",
          color = "Decision", palette = c("#00AFBB", "#E7B800","green"),
          ylab = "WritingScore", xlab = "Decision")
```



```
## Let's add labels...

(TheMean <- ddply(MyData, .(Decision), summarize,
                  mean2 = round(  mean(WritingScore) ,2 )))
```

```
##   Decision mean2
## 1    Admit 94.09
## 2  Decline 93.56
## 3 Waitlist 91.16
```

```
## Another View...

(MyV2 <- ggplot(MyData, aes(x=Decision, y=WritingScore, fill=Decision)) +
    geom_violin(trim=TRUE)+ geom_boxplot(width=0.1)+
    geom_text(data = TheMean,
              aes(x = Decision, y = mean2, label = mean2),
              size = 3, vjust = -1.5,hjust=-1)+
    ggtitle("Writing Score and Admissions Decision")+
    geom_jitter(shape=16, position=position_jitter(0.2)))
```



And lastly … VolunteerLevel

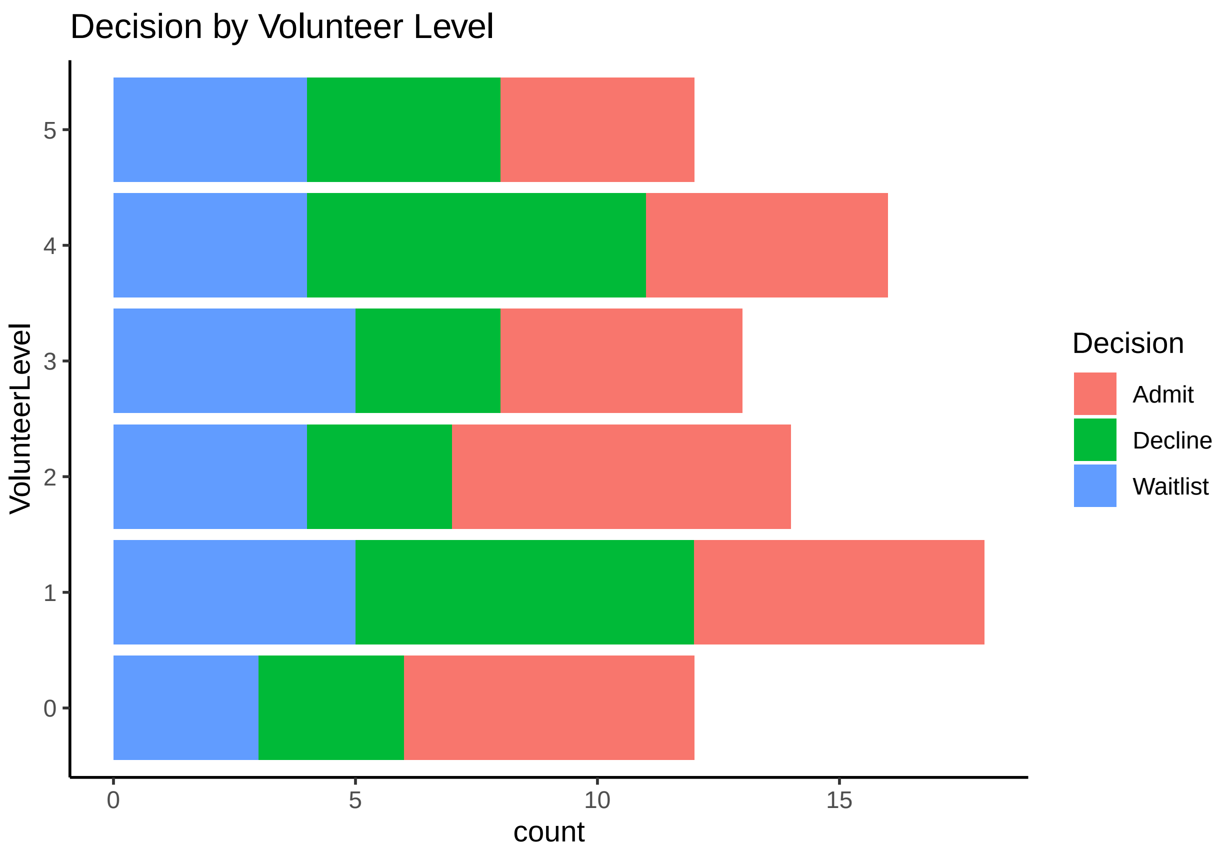
```
###########################################
##  The last variable is VolunteerLevel
##
##############################################
str(MyData$VolunteerLevel)
```

```
##  int [1:85] 1 0 0 2 2 1 2 5 0 4 ...
```

```
## This should NOT be an int
## COrrect it to factor
MyData$VolunteerLevel <- as.factor(MyData$VolunteerLevel)
table(MyData$VolunteerLevel)
```

```
##
##  0  1  2  3  4  5
## 12 18 14 13 16 12
```

```
(MyG1<-ggplot(MyData) +
    geom_bar(aes(VolunteerLevel, fill = Decision)) +
    ggtitle("Decision by Volunteer Level")+
    coord_flip())
```



This is a good starting point for some more extended EDA. Note that the first steps were to load and clean the data. We can then confirm the tidy-ness of the data visually. Next it is time to INVESTIGATE the data – EDA. We try to answer the question, how can we best leverage the data. If our research problem or goals was attempting to predict admissions based on these variables, we should assess the associations / correlations of these variables with our admissions variable (as we did in some instances above.)

This is a really good starting point for some more investigation, exploration and visualization that would be incorporated into a comprehensive EDA.