Demo RStudio and EDA

Cathy Lee

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

At the very beginning is an (optional) YAML header surrounded by `---`s, the header specifies useful meta data.

Below the header is the first code chunk, this chunk sets global options that apply to every chunk in your file. This is done by calling `knitr::opts_chunk$set` in this code chunk. Note that these global defaults can be overwritten in individual chunk headers. `include = FALSE` prevents both codes and results from appearing.

To understand `echo = TRUE`, `message = FALSE` and `warning = FALSE`, read RMarkdown tips and tricks. *Energetic Students: read the "Shortening knit time" section and try it in this file*

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

Exercise 8 from ISLR Chapter 2

The following illustrate commands for exploring this exercise using R and various packages for the `College` data.

Libraries

Try to load the `ISLR` library

```
library(ISLR)
```

If it is not available you will need to install the library from CRAN. Click on *Packages* in the lower right pane, and then *Install*. Enter the package name then click on the Install button.

You can also install from the console/command line using `install.packages("ISLR")`.

Ready?

Getting the College data

Next we will need to load the dataset. This is part of the library so we will not need to read it in using `read.csv` but rather we will use the `data` function to load it from the library.

```
data(College)
```

This loads the dataframe `College`. Note you can always see the content of any `R` object by simply typing its name.

```
College
```

For information about the variables, read the text or enter

```r
help(College)
```

The info will appear in the `help` tab.

To see explore the data, you can use the command `View(College)`.

This will open a new tab, where you may scroll left and right to look at the rows and columns. In the `View` you should see that the first column is the College/University name. These can be extracted using `rownames(College)`. Let's print out the first 5

```r
rownames(College)[1:5]
```

```
## [1] "Abilene Christian University" "Adelphi University"
## [3] "Adrian College"               "Agnes Scott College"
## [5] "Alaska Pacific University"
```

Summary

```r
summary(College)
```

```
##  Private        Apps           Accept          Enroll       Top10perc
##  No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00
##  Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00
##            Median : 1558   Median : 1110   Median : 434   Median :23.00
##            Mean   : 3002   Mean   : 2019   Mean   : 780   Mean   :27.56
##            3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902   3rd Qu.:35.00
##            Max.   :48094   Max.   :26330   Max.   :6392   Max.   :96.00
##    Top25perc      F.Undergrad     P.Undergrad         Outstate
##  Min.   :  9.0   Min.   :  139   Min.   :    1.0   Min.   : 2340
##  1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320
##  Median : 54.0   Median : 1707   Median :  353.0   Median : 9990
##  Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441
##  3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925
##  Max.   :100.0   Max.   :31643   Max.   :21836.0   Max.   :21700
##    Room.Board       Books           Personal         PhD
##  Min.   :1780   Min.   :  96.0   Min.   : 250   Min.   :  8.00
##  1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00
##  Median :4200   Median : 500.0   Median :1200   Median : 75.00
##  Mean   :4358   Mean   : 549.4   Mean   :1341   Mean   : 72.66
##  3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00
##  Max.   :8124   Max.   :2340.0   Max.   :6800   Max.   :103.00
##     Terminal       S.F.Ratio      perc.alumni        Expend
##  Min.   : 24.0   Min.   : 2.50   Min.   : 0.00   Min.   : 3186
##  1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00   1st Qu.: 6751
##  Median : 82.0   Median :13.60   Median :21.00   Median : 8377
##  Mean   : 79.7   Mean   :14.09   Mean   :22.74   Mean   : 9660
##  3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00   3rd Qu.:10830
##  Max.   :100.0   Max.   :39.80   Max.   :64.00   Max.   :56233
##    Grad.Rate
##  Min.   : 10.00
##  1st Qu.: 53.00
##  Median : 65.00
##  Mean   : 65.46
##  3rd Qu.: 78.00
##   Max.   :118.00
```

Dimensions of Data

How many observations and variables are in the dataframe?

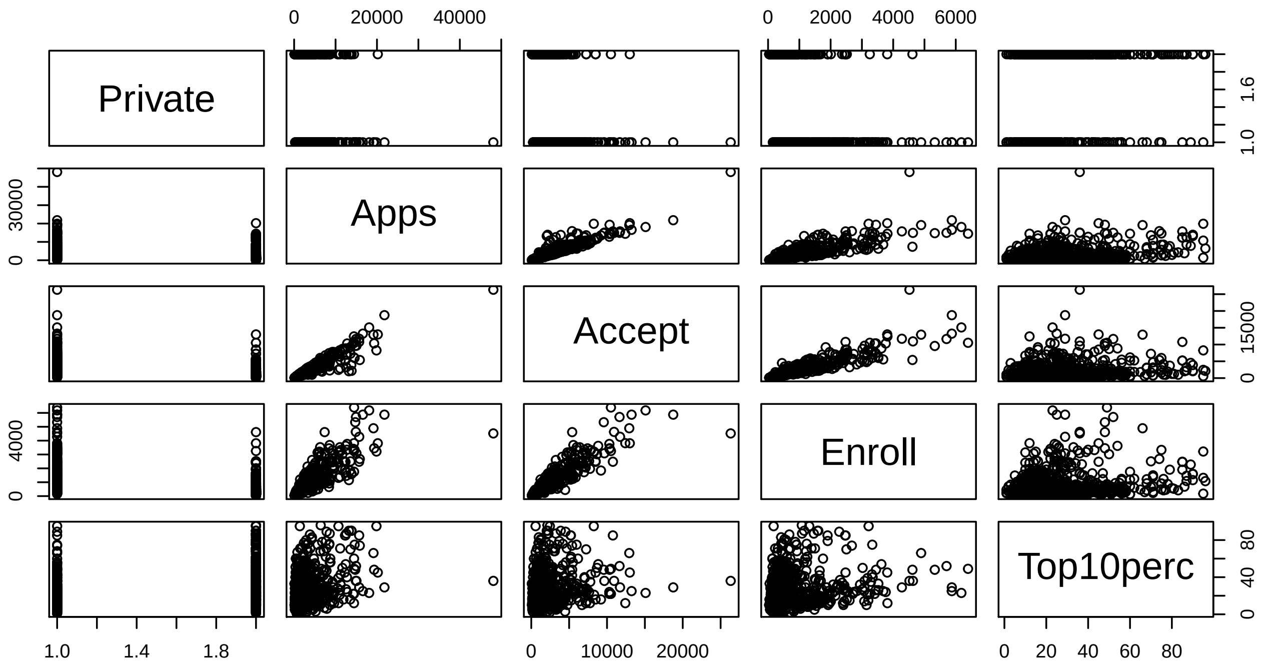
```
## [1] 777  18
```

Suppose we want to refer to those numbers in the text. We can extract them using n = 777 and d = 18. Look at the code to see how we extracted them.

Scatter plot matrices

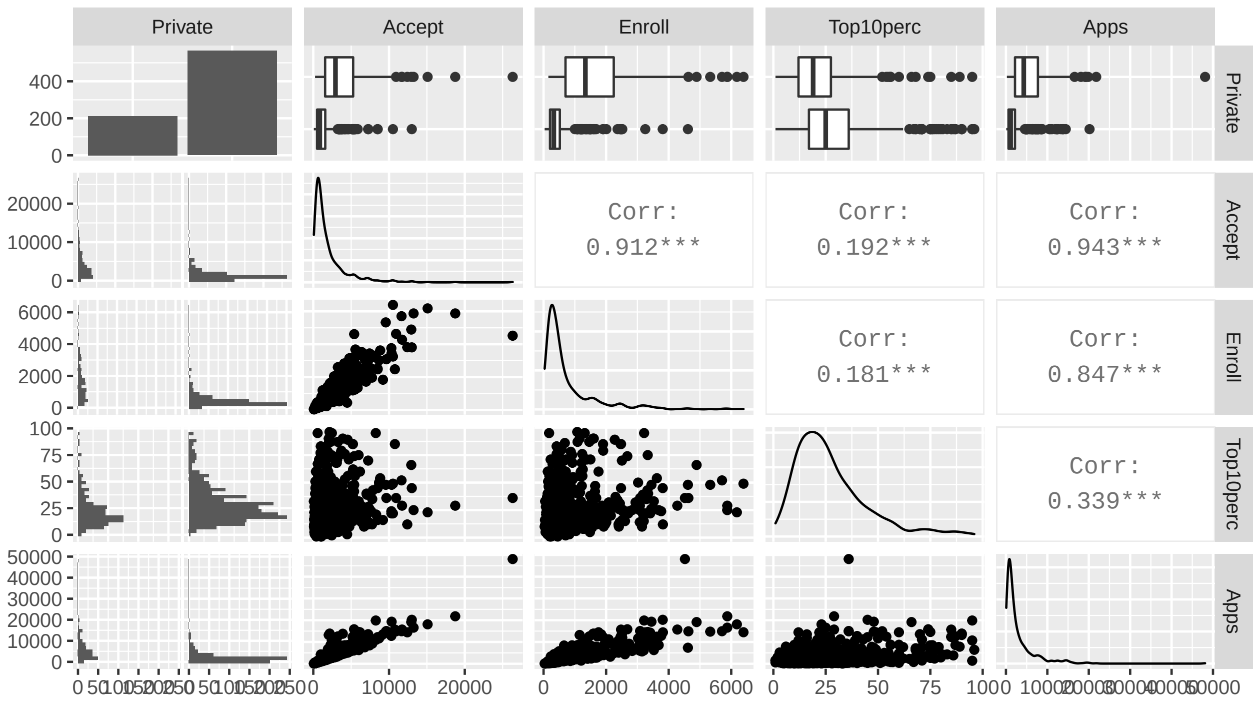
There `base R` version of scatter plot matrices is obtained using the `pairs` function to plot all variables versus each other. We can use subsetting of columns of the dataframe to look at the first 5 columns.

```
pairs(College[, 1:5])
```



We can also look at this using the `ggpairs` function. Install the library `GGally` if it is not available (and any dependent libraries) and load it.

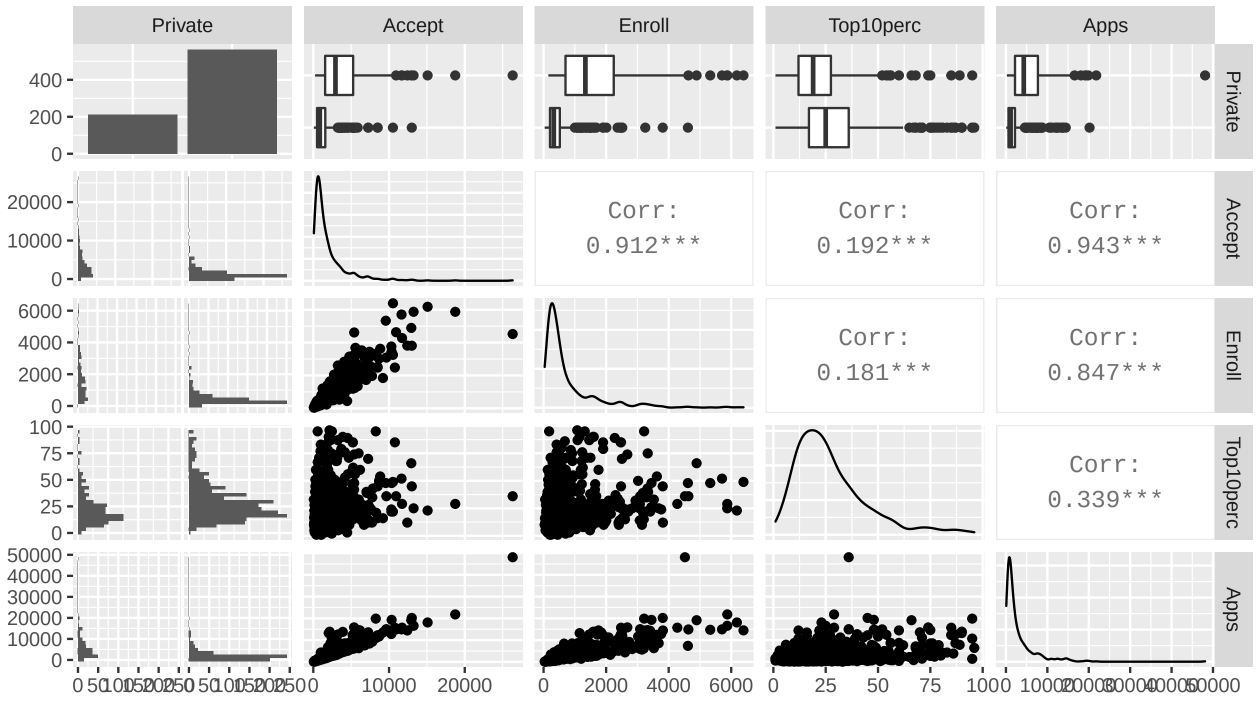
```
library(GGally)
ggpairs(College, columns= c(1,3:5, 2))
```



The last variable, 2 in this case, refers to our response variable.

Note `ggpairs` will show its progress in your output and using warning=FALSE or message=FALSE does not suppress it. This answer on this Stackexchange post: provided the trick or `progress=F` to suppress the progress bar.

```
gp = ggpairs(College, columns= c(1,3:5, 2))
print(gp, progress=F)
```



The `ggpairs` function realizes that the variable `Private` is categorial and plots side by side histograms. The density plots are also useful for seeing the skewness in the marginal distributions.

What other features do these plots indicate? *(Think about assumptions for linear regression)*

New variables

Let's create a new variable `Elite` by binning the `Top10perc` variable. We are going to divide universities into two groups based on whether or not the proportion of students coming from the top 10% of their high school classes exceeds 50 %. We will use the library `dplyr` to illustrate some of the possible transformations and the idea of pipes (`%>%`), which are quite powerful once you get the hang of them!

For students who are unfamiliar with `dplyr`, read this intro first.

```r
library(dplyr)
College = College %>%
  mutate(Elite = factor(Top10perc > 50)) %>%
  mutate(Elite =
           recode(Elite, 'TRUE' = "Yes", 'FALSE'="No"))
```

What is the above doing? Document the code here.

Compare to the base `R` code:

```r
Elite=rep("No",nrow(College))
Elite[College$Top10perc >50]="Yes"
Elite=as.factor(Elite)
college=data.frame(College ,Elite)
```

How many Elite universities are there?

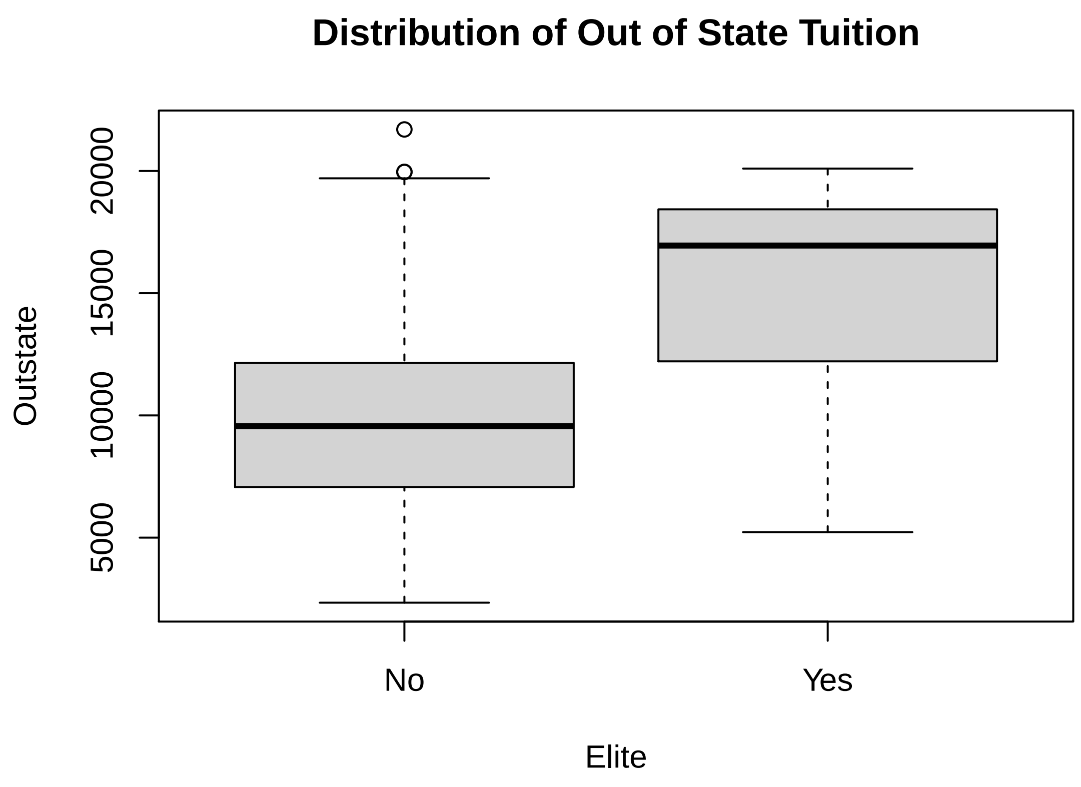
```r
summary(College$Elite)
```

```
##  No Yes
## 699  78
```

Side by Side Boxplots

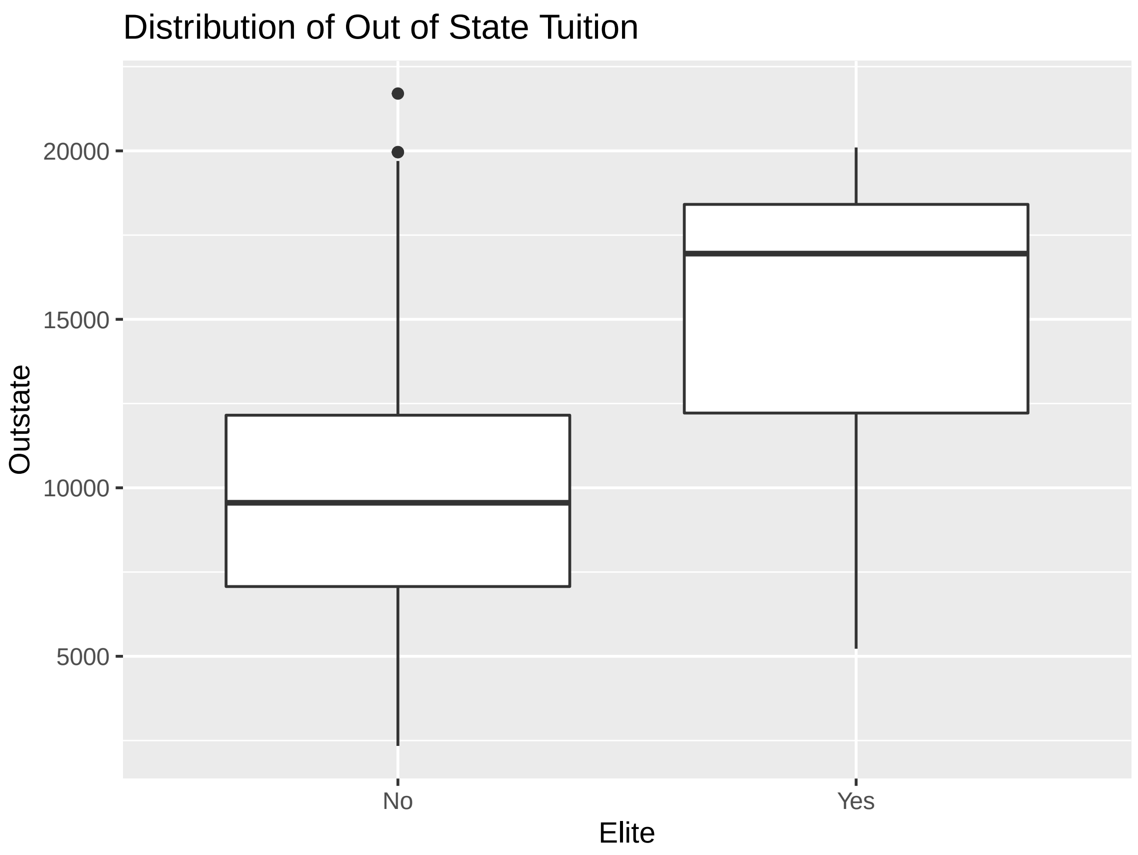
Let's plot the variable `Outstate` versus `Elite` using side-by-side boxplots. Using `base R` we would enter

```r
boxplot(Outstate ~ Elite, data=College,
        ylab="Outstate", xlab="Elite")
title("Distribution of Out of State Tuition")
```



Now for the `ggplot` version:

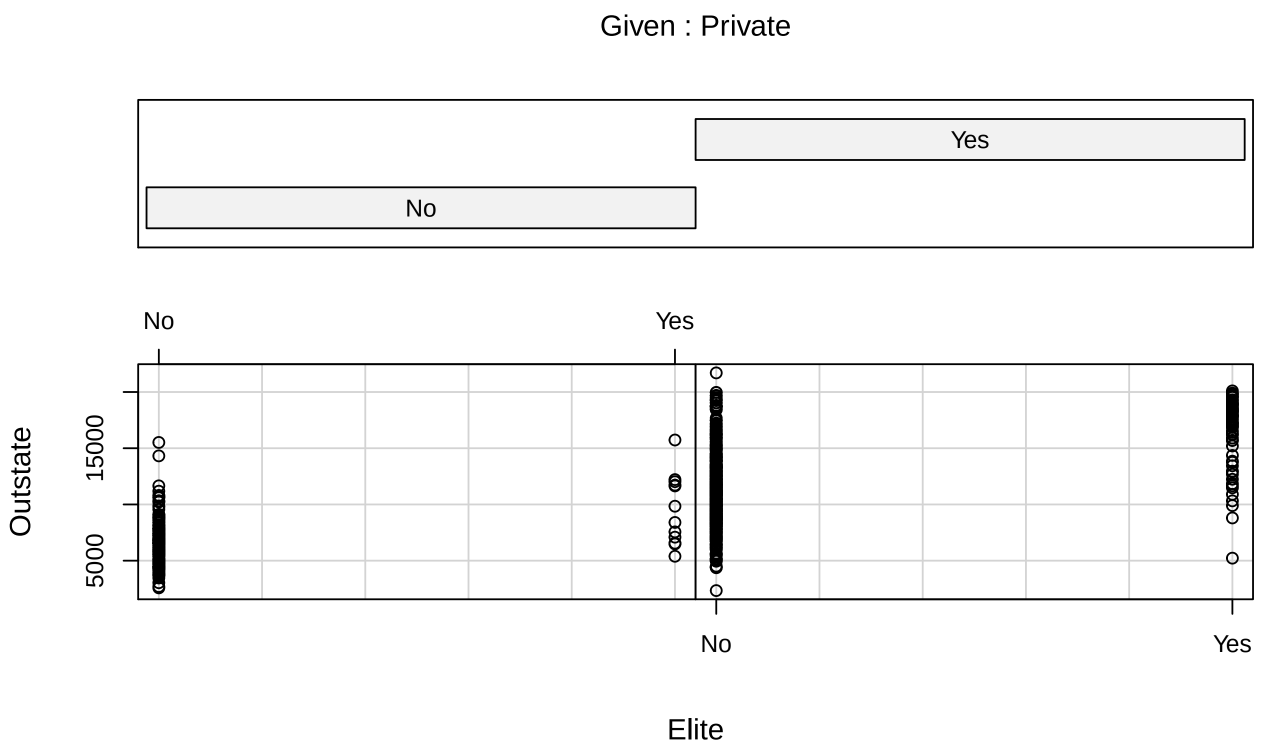
```
library(ggplot2)
my.bp <<-ggplot(data=College, aes(y= Outstate, x=Elite)) # Creates boxplots
my.bp <- my.bp + geom_boxplot() # Adds color
my.bp <- my.bp + ggtitle("Distribution of Out of State Tuition") # Adds a title
my.bp <- my.bp +  ylab("Outstate") + xlab("Elite") # Adds lables for axes
my.bp # displays the boxplots
```



Conditional Plots

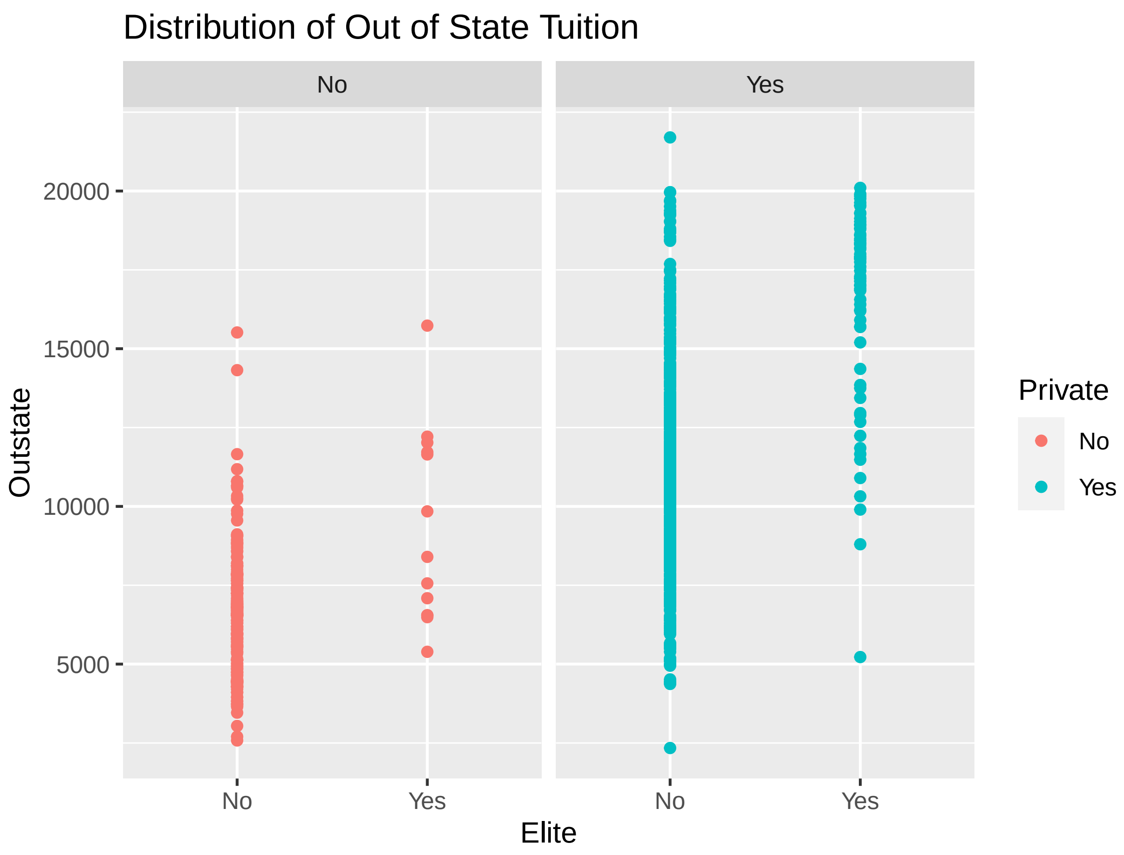
Let's look at the distribution of Out of state tuition versus Elite status for Private versus Public universities using *conditional plots*

```
coplot(Outstate ~ Elite | Private, data=College)
```



ggplot conditional plot

```
library(ggplot2)
ggplot(College, aes(x = Elite, y = Outstate,
                    group = Private,
                    color = Private)) +
  geom_point() + facet_grid(.~Private) + ggtitle("Distribution of Out of State Tuition")
```



Next Steps

Update this document and explore the other variables thinking about the objective of predicting `Apps`. Document what you discover thinking about models to predict `Apps`.

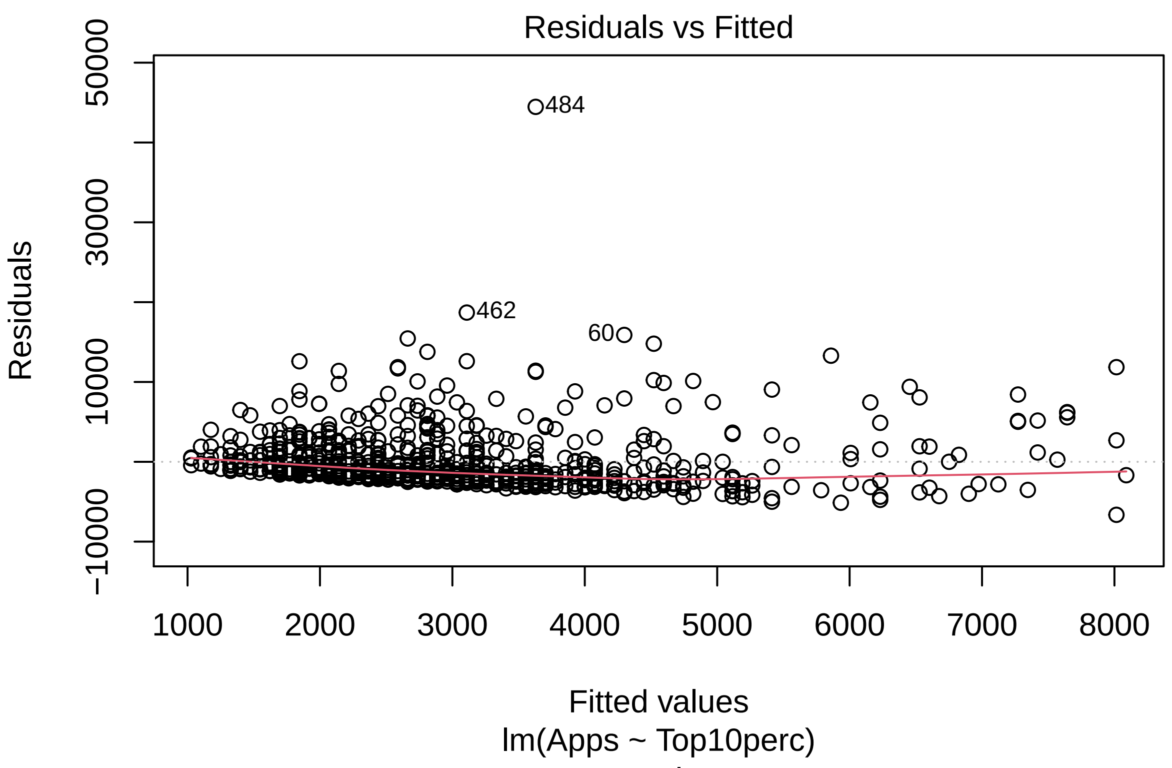
Fit a linear model

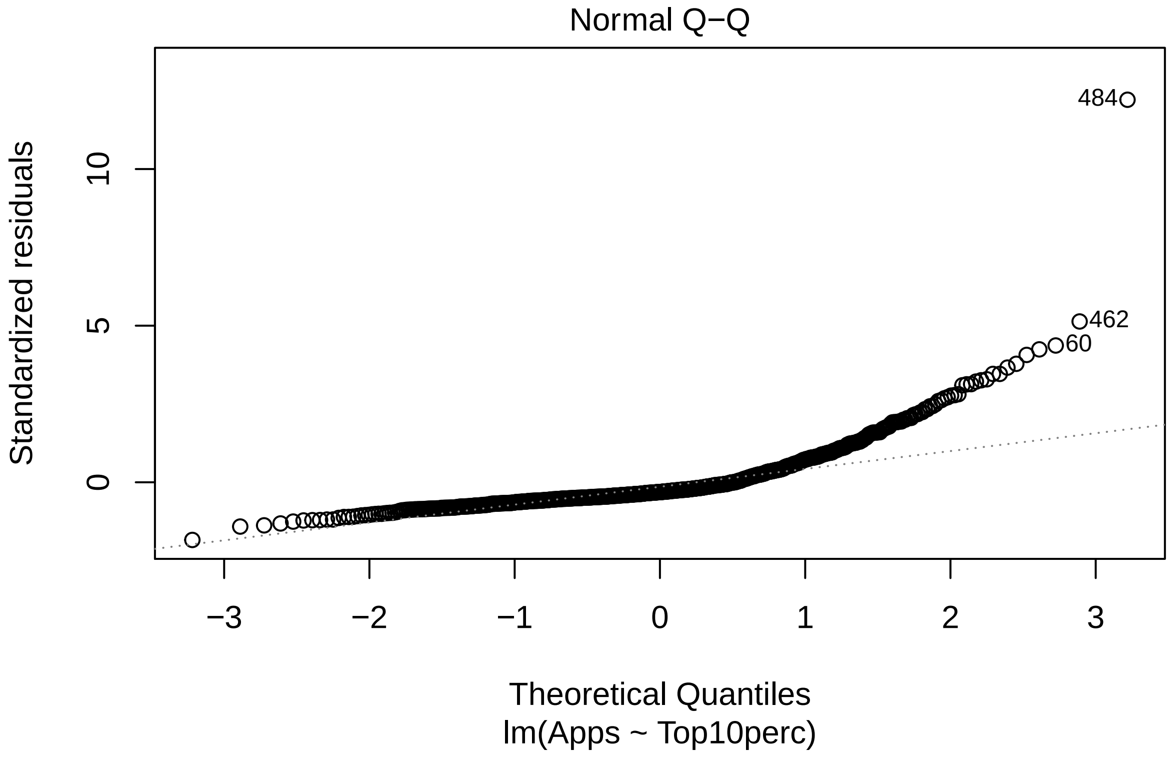
Use the `lm` function as described in class to fit a linear model with `Apps` as the response variable and add one of the predictors in place of `XXX` below.

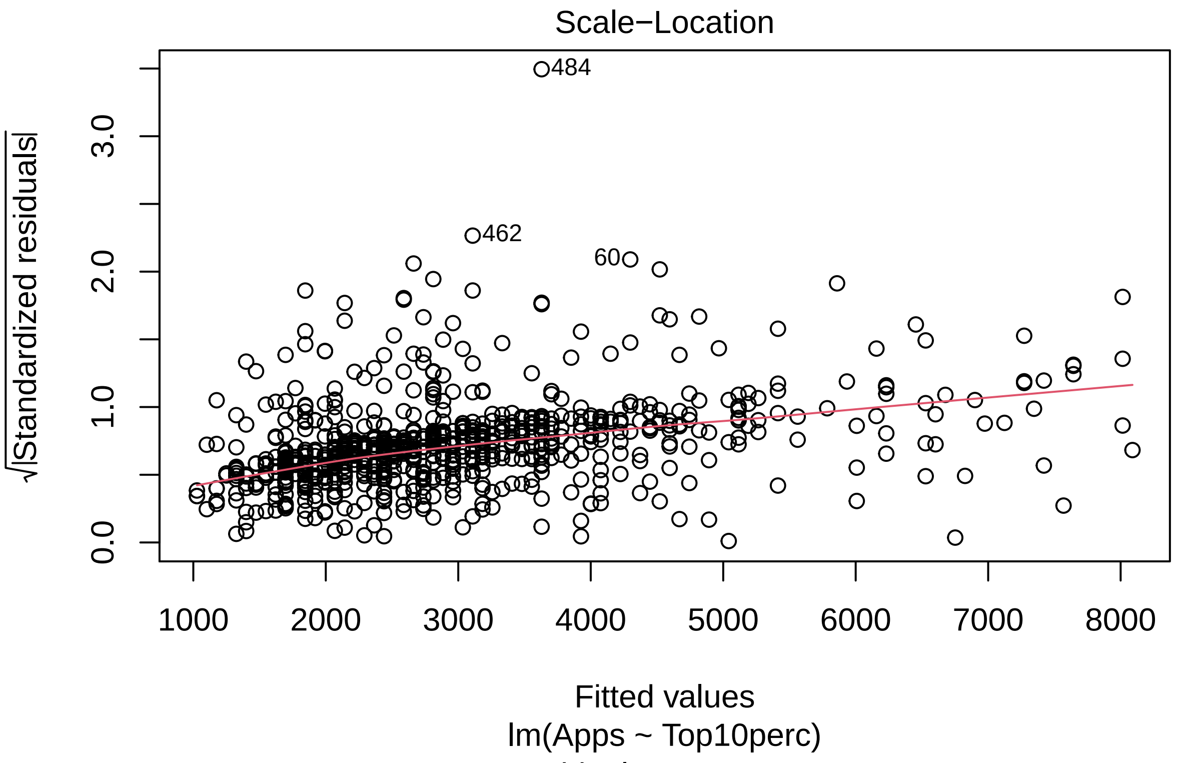
```
# add lm code here

mymodel = lm(Apps ~ Top10perc, data=College)
```

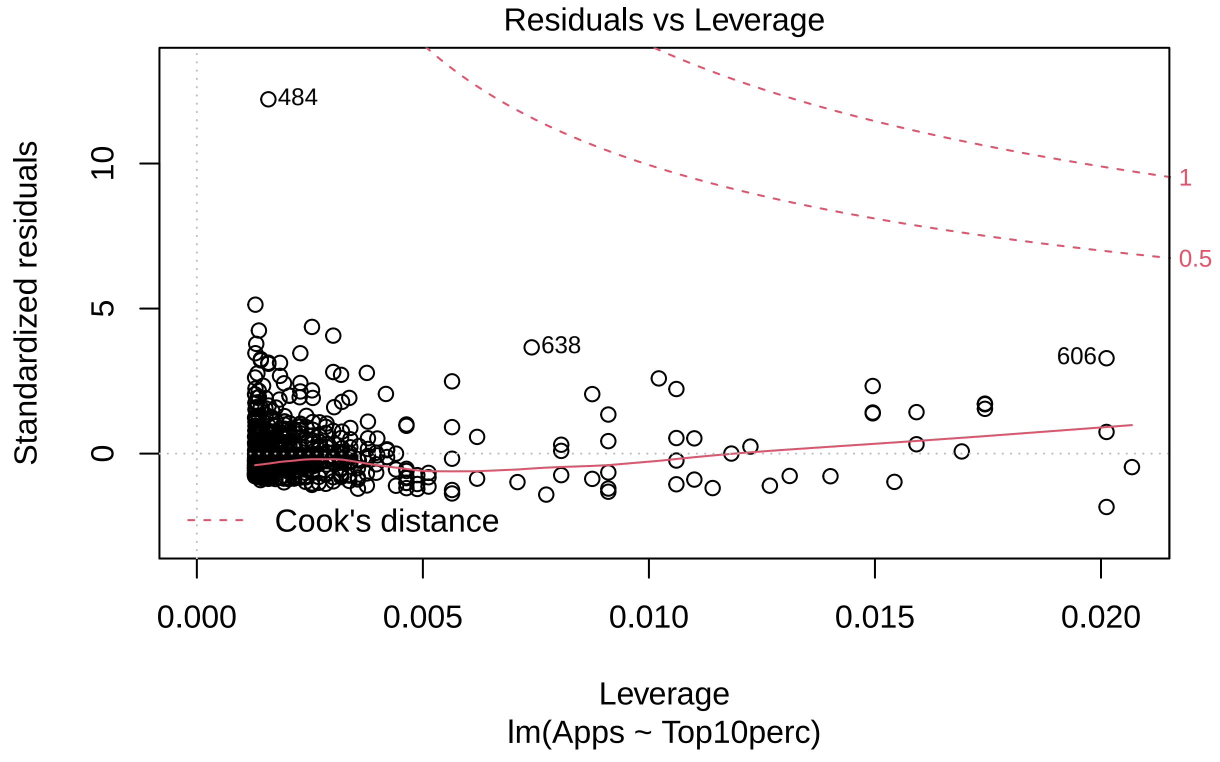
For the fitted model, produce diagnostic plots

```
# add diagnostic plots here
plot(mymodel)
```


Residuals vs Fitted
Residuals
Fitted values
lm(Apps ~ Top10perc)
484
462
60
−10000
10000
30000
50000
1000
2000
3000
4000
5000
6000
7000
8000


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Apps ~ Top10perc)
484
462
60
0
5
10
−3
−2
−1
0
1
2
3





Discuss whether the model seems appropriate in terms of assumptions.

Updating your work on GitHub

In the upper right you should see a tab labeled `Git`. Click on that to see a list of files that have been added/changed.

To save changes to github, check the box of any file to add/update. Then click on the `Commit` link (just above path). In the pop-up window add a Commit message (something meaningful that identifies the changes made). When done, click on Commit to save the changes. Click on the Green Up arrow to `push` your changes to Github.

Note: Update 8/29/2019 we are still waiting on the github classroom permissions, so you will not be able to push your changes to the class organization site.