Matthew Reynolds
Oren Mangoubi
DS 4635
22 March 2023

Assignment 1

Problem 1)

A) When our sample size is extremely large and the number of predictors is small, we would expect a flexible statistical learning method to have a worse performance than an inflexible method because due to the large sample size the inflexible method will be capable of making enough observations for reliable predictions.
B) When our number of predictors is extremely larger and the number of observations is small, we would expect a flexible statistical learning method to perform better than an inflexible method due to inflexible methods having a limited ability to model relationships among predictors.
C) When the relationship between the predictors and response is highly non-linear, we would generally expect a flexible statistical learning method to perform better than an inflexible method because inflexible methods struggle with non-linear relationships as they are designed to model linear relationships.
D) When the variance of the error terms is extremely high, we would expect a flexible statistical learning method to performance worse than an inflexible method since a high variance of error can mean that the data has unneeded noise or external factors unaccounted for.

Problem 2)

A) This is a regression problem since we are attempting to predict the CEO salary, a continuous number. There's not enough information to deduce n and p.
B) This is a classification problem since we are trying to predict whether it will be a success or a failure, a category. There's not enough information to deduce n and p.
C) This is a regression problem since we are attempting to predict the % change of the USD/Euro exchange rate. Our n is 52 and the p is 3.

Problem 5)

A very flexible approach is advantageous because it can capture complex relationships between predictor and response variables, get higher accuracy, and avoid underfitting. It can be bad because it can be overfit to the data and require large data sets to train. A more flexible approach would be better in situations when the data between predictors and response variables are nonlinear. A less flexible approach would be better in situations when the relationship between predictors and response variables are linear and when there's limited data to train on.

Problem 8)

I did this in Rmarkdown. Not entirely sure how you would like me submit this so I'll just take screenshots of the html file knit and attach them below for you all.

Assignment 1

Matthew Reynolds

22 March 2023

Introduction: Completing exercises from “Introduction to Statistical Learning First Edition” from www.statlearning.com. These exercises are from Chapter 2.

Exercise 8)

A. Use the read.csv() function to read the data into R. Call the loaded data college. Make sure that you have the directory set to the correct location for the data.

```
college <- read.csv("https://www.statlearning.com/s/College.csv")
```

B. Look at the data using the fix() function. You should notice that the first column is just the name of each university. We don’t really want R to treat this as data. However, it may be handy to have these names for later.

```
rownames(college)=college[,1]
head(college)
```

	X <chr>	Private <chr>	A... <int>	Accept <int>	Enroll <int>	Top10perc <int>
Abilene Christian University	Abilene Christian University	Yes	1660	1232	721	23
Adelphi University	Adelphi University	Yes	2186	1924	512	16
Adrian College	Adrian College	Yes	1428	1097	336	22
Agnes Scott College	Agnes Scott College	Yes	417	349	137	60
Alaska Pacific University	Alaska Pacific University	Yes	193	146	55	16
Albertson College	Albertson College	Yes	587	479	158	38

6 rows | 1-7 of 20 columns

You should see that there is now a row.names column with the name of each university recorded. This means that R has given each row a name corresponding to the appropriate university. R will not try to perform calculations on the row names. However, we still need to eliminate the first column in the data where the names are stored.

You should see that there is now a row.names column with the name of each university recorded. This means that R has given each row a name corresponding to the appropriate university. R will not try to perform calculations on the row names. However, we still need to eliminate the first column in the data where the names are stored.

```
college=college[,-1]
head(college)
```

	Private <chr>	Ap... <int>	Accept <int>	Enroll <int>	Top10perc <int>	Top25perc <int>	F.Undergrad <int>
Abilene Christian University	Yes	1660	1232	721	23	52	2885
Adelphi University	Yes	2186	1924	512	16	29	2683
Adrian College	Yes	1428	1097	336	22	50	1036
Agnes Scott College	Yes	417	349	137	60	89	510
Alaska Pacific University	Yes	193	146	55	16	44	249
Albertson College	Yes	587	479	158	38	62	678

6 rows | 1-8 of 19 columns

Now you should see that the first data column is Private. Note that another column labeled row.names now appears before the Private column. However, this is not a data column but rather the name that R is giving to each row.

C. I. Use the summary() function to produce a numerical summary of the variables in the data set.

```
summary(college)
```

```
##    Private               Apps           Accept          Enroll
##  Length:777         Min.   :   81   Min.   :   72   Min.   :  35
##  Class :character   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242
##  Mode  :character   Median : 1558   Median : 1110   Median : 434
##                     Mean   : 3002   Mean   : 2019   Mean   : 780
##                     3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902
##                     Max.   :48094   Max.   :26330   Max.   :6392
##    Top10perc       Top25perc      F.Undergrad     P.Undergrad
##  Min.   : 1.00   Min.   :  9.0   Min.   :  139   Min.   :    1.0
##  1st Qu.:15.00   1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0
##  Median :23.00   Median : 54.0   Median : 1707   Median :  353.0
##  Mean   :27.56   Mean   : 55.8   Mean   : 3700   Mean   :  855.3
```

C. I. Use the summary() function to produce a numerical summary of the variables in the data set.

```
summary(college)
```

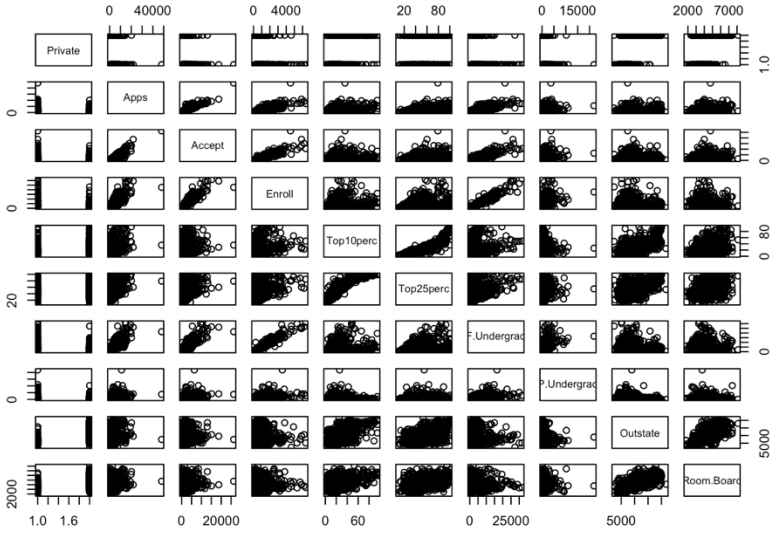
```
##    Private               Apps           Accept          Enroll
##  Length:777         Min.   :   81   Min.   :   72   Min.   :  35
##  Class :character   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242
##  Mode  :character   Median : 1558   Median : 1110   Median : 434
##                     Mean   : 3002   Mean   : 2019   Mean   : 780
##                     3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902
##                     Max.   :48094   Max.   :26330   Max.   :6392
##    Top10perc       Top25perc      F.Undergrad     P.Undergrad
##  Min.   : 1.00   Min.   :  9.0   Min.   :  139   Min.   :    1.0
##  1st Qu.:15.00   1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0
##  Median :23.00   Median : 54.0   Median : 1707   Median :  353.0
##  Mean   :27.56   Mean   : 55.8   Mean   : 3700   Mean   :  855.3
##  3rd Qu.:35.00   3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0
##  Max.   :96.00   Max.   :100.0   Max.   :31643   Max.   :21836.0
##     Outstate       Room.Board       Books           Personal
##  Min.   : 2340   Min.   :1780   Min.   :  96.0   Min.   : 250
##  1st Qu.: 7320   1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850
##  Median : 9990   Median :4200   Median : 500.0   Median :1200
##  Mean   :10441   Mean   :4358   Mean   : 549.4   Mean   :1341
##  3rd Qu.:12925   3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700
##  Max.   :21700   Max.   :8124   Max.   :2340.0   Max.   :6800
##       PhD            Terminal       S.F.Ratio      perc.alumni
##  Min.   :  8.00   Min.   : 24.0   Min.   : 2.50   Min.   : 0.00
##  1st Qu.: 62.00   1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00
##  Median : 75.00   Median : 82.0   Median :13.60   Median :21.00
##  Mean   : 72.66   Mean   : 79.7   Mean   :14.09   Mean   :22.74
##  3rd Qu.: 85.00   3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00
##  Max.   :103.00   Max.   :100.0   Max.   :39.80   Max.   :64.00
##      Expend        Grad.Rate
##  Min.   : 3186   Min.   : 10.00
##  1st Qu.: 6751   1st Qu.: 53.00
##  Median : 8377   Median : 65.00
##  Mean   : 9660   Mean   : 65.46
##  3rd Qu.:10830   3rd Qu.: 78.00
##  Max.   :56233   Max.   :118.00
```

II. Use the pairs() function to produce a scatterplot matrix of the first ten columns or variables of the data. Recall that you can reference the

II. Use the pairs() function to produce a scatterplot matrix of the first ten columns or variables of the data. Recall that you can reference the first ten columns of a matrix A using A[,1:10].

```
#Changing Private to a factor variable
college$Private <- as.factor(college$Private)

pairs(college[,1:10])
```

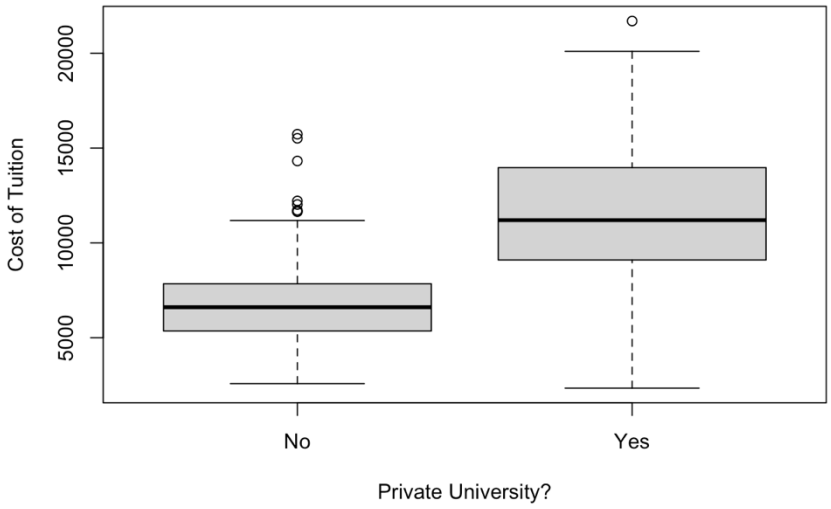


III. Use the plot() function to produce side-by-side boxplots of Outstate versus Private.

```
plot(college$Private, college$Outstate, xlab = "Private University?", ylab = "Cost of Tuition")
```

III. Use the plot() function to produce side-by-side boxplots of Outstate versus Private.

```
plot(college$Private, college$Outstate, xlab = "Private University?", ylab = "Cost of Tuition")
```



IV. Create a new qualitative variable, called 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 %.

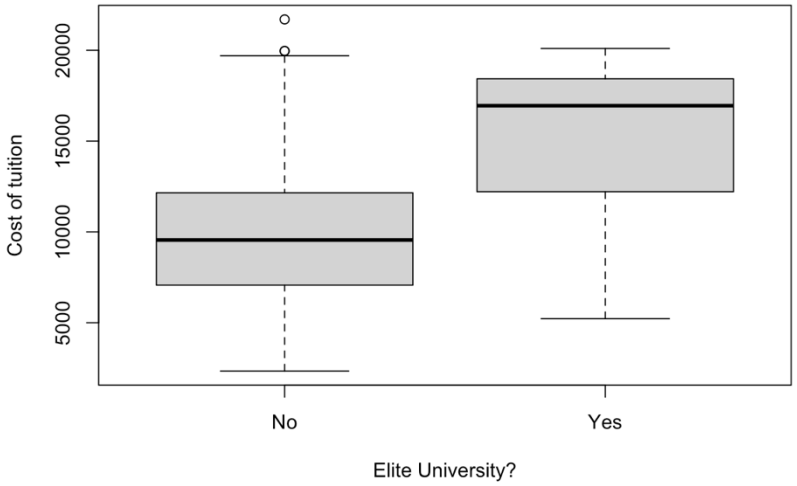
```
# Provided from textbook
Elite=rep("No",nrow(college))
Elite[college$Top10perc >50]="Yes"
Elite=as.factor(Elite)
college=data.frame(college ,Elite)
```

Use the summary() function to see how many elite univer- sities there are. Now use the plot() function to produce side-by-side boxplots of Outstate versus Elite.

```
summary(Elite)
```

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

```
plot(college$Elite, college$Outstate, xlab="Elite University?", ylab="Cost of tuition")
```

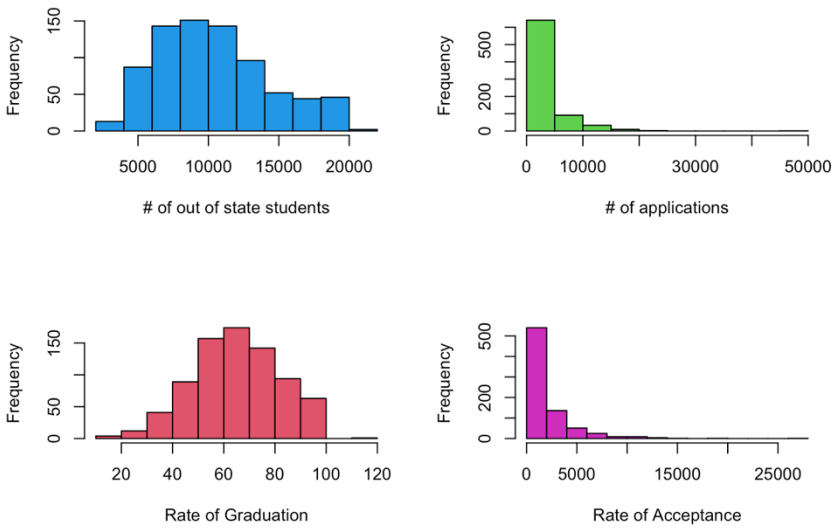


V. Use the hist() function to produce some histograms with differing numbers of bins for a few of the quantitative vari- ables. You may find the command par(mfrow=c(2,2)) useful: it will divide the print window into four regions so that four plots can be made simultaneously. Modifying the arguments to this function will divide the screen in other ways.

V. Use the hist() function to produce some histograms with differing numbers of bins for a few of the quantitative vari- ables. You may find the command par(mfrow=c(2,2)) useful: it will divide the print window into four regions so that four plots can be made simultaneously. Modifying the arguments to this function will divide the screen in other ways.

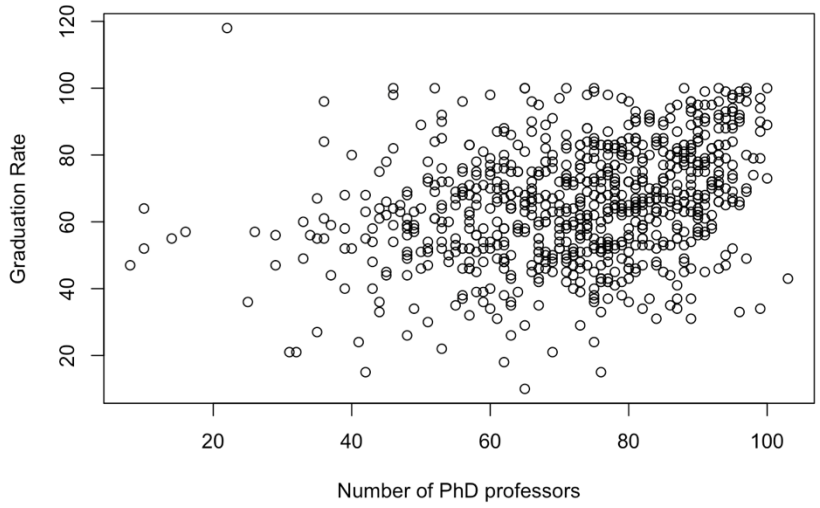
```
# Fragmenting the display
par(mfrow=c(2,2))

# Creating the histograms
hist(college$Outstate, col = 4, xlab = "# of out of state students", main = "")
hist(college$Apps, col = 3, xlab = "# of applications", main = "")
hist(college$Grad.Rate, col = 2, xlab = "Rate of Graduation", main = "")
hist(college$Accept, col = 6, xlab = "Rate of Acceptance", main = "")
```



VI. Continue exploring the data, and provide a brief summary of what you discover.

```
# Number of PhD professors correlation to graduation rate
plot(college$PhD, college$Grad.Rate, xlab = "Number of PhD professors", ylab = "Graduation Rate")
```




```
# School with highest amount from top 10 percent of high school class
row.names(college)[which.max(college$Top10perc)]
```

```
## [1] "Massachusetts Institute of Technology"
```