512: Homework #1

Brody Vogel

Preparation

```r
set.seed(runif(1,1,1000))

library('ISLR')
```

```
## Warning: package 'ISLR' was built under R version 3.4.2
```

Problem 1 (#4)

a]

1) One could predict the way a person will vote. The response, for example, could be: Republican, Democrat, or Independent. And the predictors could be, for example: Income, Location, Education, Age, and Model of Car Driven (among many others). The main goal, here, would be *prediction*: we wouldn't necessarily care *why* a person votes the way they do, but we'd rather be interested in knowing that they *will* vote that way.

ii) One could use classification to predict whether a person will like or dislike a new TV show. The response could be: Fan or Not a Fan. And the predictors could be, for example: Age, Education, Gender, Marital Status, etc. The goal, here, would be *inference*: assuming we're the show's marketing group, we'd be more interested in knowing the types of people who like our show than making any predictions. Predictions wouldn't be all that useful; knowing who likes our show, though, could help us design a marketing/advertising campaign.

iii) One could predict whether a geographic region will be heavily affected by this year's flu virus. The response could be: Heavily Affected, Moderately Affected, or Lightly Affected. And the predictors could be, for example: Average Age, Gender Dispersion, Ethnicity, Climate, etc. The main goal, here, would be *prediction*: if we're in charge of allocating resources (like flu shots), we'd be more interested in knowing *if* the flu will heavily affect a region than *why* it will. Although, in this case even more so than the previous two, knowing the *why* would also be useful and interesting.

b]

1) One could try to predict the price of Bitcoin. The quantitative response would be the actual price of a Bitcoin in U.S. Dollars. And the predictors could be, for example: Price of the Dow, Price of the NASDAQ, Value of USD Relative to Euro, etc. The goal, here, would be *prediction*: if we're investors, we don't really care *why* the price of Bitcoin is what it is - we just want to know what it will be.

ii) One could predict the GPA of college students after their first year. The response would be a student's GPA, for example on a 0.0-4.0 scale. And the predictors could be, for example: High School GPA, SAT Score, Relationship with Roommate(s), Intended Major, etc. The goal, here, would be *inference*: if we're administration, we don't much care what any particular student's GPA will be after their first year. Instead, we'd want to know what most contributes to a student being academically successful in their first year of college.

iii) One could predict how many pizzas they should order for their Super Bowl party. The response would be the number of pizzas that should be ordered. And the predictors could be, for example: Number of Guests, Guests' Appetite, Type of Pizza, etc. The goal, here, would be *prediction*: if we're hosting, we don't care *why* we'll need X number of pizzas - we just want to have enough pizza to feed all of our guests.

c]

1) One could try to use clustering to group running backs in terms of their fantasy football value. If I could accurately separate the NFL's running backs into tiers based on fantasy production, I'd have an advantage in my fantasy draft next fall.

ii) If I ran an accounting firm, I could try to use clustering to group my clients. If I could do that accurately, I could better serve each client, as in: knowing how much time to allocate for every appointment, knowing when each client is likely to begin their tax return process, knowing whether it's likely that a client will be amenable to digital tax forms, etc.

iii) If I was responsible for advertising on a website, I could use clustering to group my pageviewers. If I could do that accurately, I could tailor the advertisements on a viewer's screen to be in-line with what they're most likely to click on, and this could make my website more profitable.

Problem 2 (#7)

a)

Euclidean Distances: O1 $= 3$; O2 $= 2$; O3 ≈ 3.16 ; O4 ≈ 2.24 ; O5 ≈ 1.41 ; O6 ≈ 1.73

```
o1 <- sqrt(0^2 + 3^2 + 0^2)
o2 <- sqrt(2^2 + 0^2 + 0^2)
o3 <- sqrt(0^2 + 1^2 + 3^2)
o4 <- sqrt(0^2 + 1^2 + 2^2)
o5 <- sqrt((-1)^2 + 0^2 + 1^2)
o6 <- sqrt(1^2 + 1^2 + 1^2)

c(o1, o2, o3, o4, o5, o6)
```

```
## [1] 3.000000 2.000000 3.162278 2.236068 1.414214 1.732051
```

b) When K = 1, our prediction for $X_1 = X_2 = X_3 = 0$ is Green. This is because Observation 5 is the closest neighbor - based on Euclidean distance - of the point, and Observation 5 is Green.

c) When K = 3, our prediction for $X_1 = X_2 = X_3 = 0$ is Red. This is because Observations 5, 6, and 2 are the three closest neighbors - based on Euclidean distance - of the point, and two of those three Observations are Red.

d) We'd expect the best value for K to be small, because a large K would take more points into consideration and so try to fit a more linear boundary. Because the boundary is non-linear, the high risk of overfitting with a large K, here, outweighs the risk of underfitting with a small K.

Problem 3 (#9)

a)

Qualitative: Name, Origin

Quantitative: MPG, Cylinders, Displacement, Horsepower, Weight, Acceleration, Year

```
data(Auto)
Auto <- na.omit(Auto)
#head(Auto)
```

b)

MPG: 9-46 ; Cylinders: 3-8 ; Displacement: 68-455 ; Horsepower: 46-230 ; Weight: 1613-5140 ; Acceleration: 8.0-24.8 ; Year: 70-82

```
range(Auto$mpg)
```

```
## [1]  9.0 46.6
```

```
range(Auto$cylinders)
```

```
## [1] 3 8
```

```
range(Auto$displacement)
```

```
## [1]  68 455
```

```
range(Auto$horsepower)
```

```
## [1]  46 230
```

```
range(Auto$weight)
```

```
## [1] 1613 5140
```

```
range(Auto$acceleration)
```

```
## [1]  8.0 24.8
```

```
range(Auto$year)
```

```
## [1] 70 82
```

c)

Mean

MPG=23.446 ; Cylinders=5.472 ; Displacement=194.412 ; Horsepower=104.469 ; Weight=2977.584 ; Acceleration=15.541 ; Year=75.98

Standard Deviation

MPG=7.805 ; Cylinders=1.706 ; Displacement=104.644 ; Horsepower=38.491 ; Weight=849.403 ; Acceleration=2.759 ; Year=3.684

```
sapply(Auto[, 1:7], mean)
```

```
##          mpg    cylinders displacement   horsepower       weight
##    23.445918     5.471939   194.411990   104.469388  2977.584184
## acceleration         year
##    15.541327    75.979592
```

```
sapply(Auto[, 1:7], sd)
```

```
##          mpg    cylinders displacement   horsepower       weight
##     7.805007     1.705783   104.644004    38.491160   849.402560
## acceleration         year
##     2.758864     3.683737
```

d)

Variable	Range	Mean	SD
MPG	11-46	24.404	7.867
Cylinders	3-8	5.373	1.654
Displacement	68-455	187.241	99.678
Horsepower	46-230	100.722	35.709
Weight	1649-4997	2935.972	811.3
Acceleration	8.5-24.8	15.727	2.694
Year	70-82	77.146	3.106

```
Auto1 <- Auto[-(10:85), ]

sapply(Auto1[, 1:7], range)
```

```
##       mpg cylinders displacement horsepower weight acceleration year
## [1,] 11.0         3           68         46   1649          8.5   70
## [2,] 46.6         8          455        230   4997         24.8   82
```

```
sapply(Auto1[, 1:7], mean)
```

```
##          mpg    cylinders displacement   horsepower       weight
##    24.404430     5.373418   187.240506   100.721519  2935.971519
## acceleration         year
##    15.726899    77.145570
```

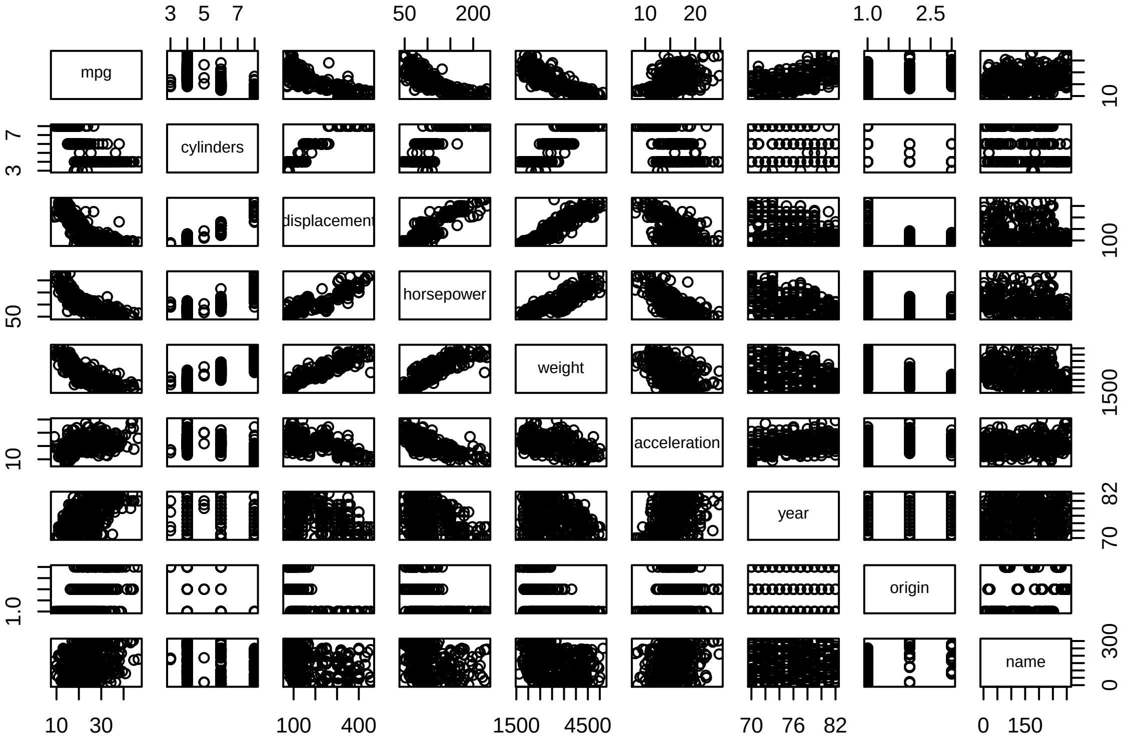
```
sapply(Auto1[, 1:7], sd)
```

```
##          mpg    cylinders displacement   horsepower       weight
##     7.867283     1.654179    99.678367    35.708853   811.300208
## acceleration         year
##     2.693721     3.106217
```

e) From the scatter matrix, it looked like MPG, Horsepower, Weight, and Cylinders have the strongest correlation. After investigating a few scatter plots between those predictors, it looks like the data suggest a few things:

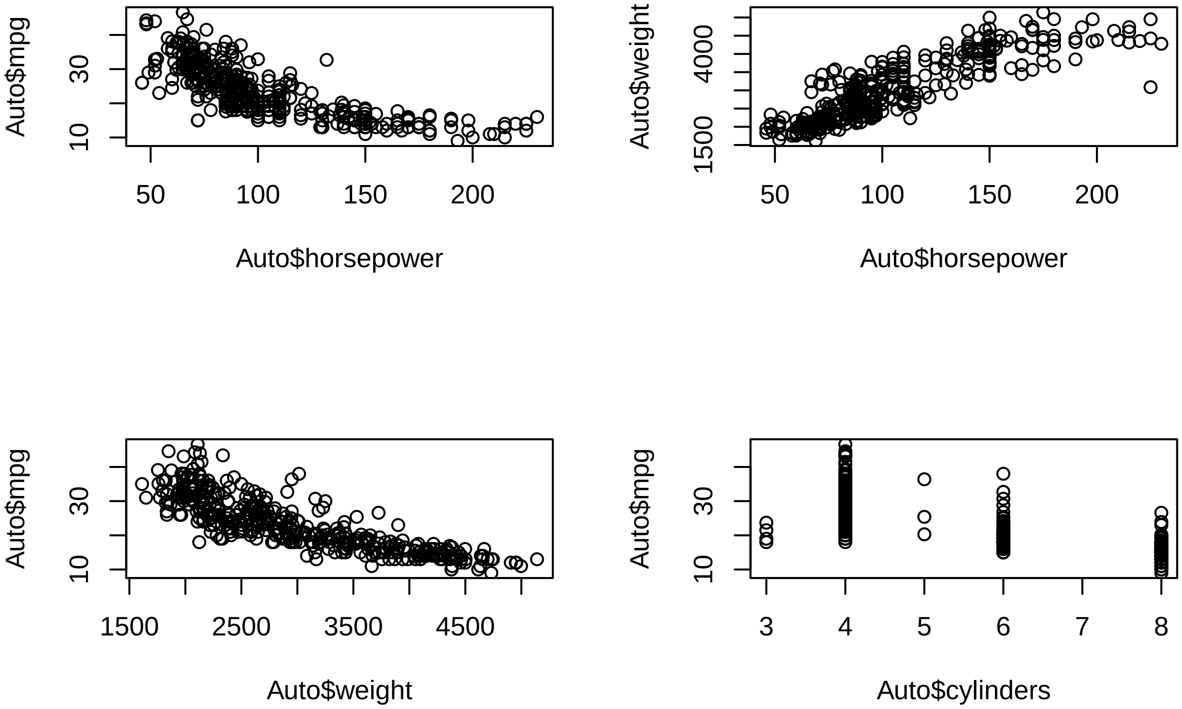
- As horsepower increases, the miles per gallon decrease.
- As the weight of a vehicle increases, so does its horsepower.
- As the weight of a vehicle increases, its mpg decrease.
- The more cylinders a vehicle has, the fewer miles per gallon it gets. (Although it looks like 4-cylinders do better than 3-cylinders. I'm not sure what a 3-cylinder vehicle even is, and there are very few in the dataset, so I think they can be ignored.)

```
pairs(Auto)
```



```r
par(mfrow = c(2,2))

plot(Auto$horsepower, Auto$mpg)
plot(Auto$horsepower, Auto$weight)
plot(Auto$weight, Auto$mpg)
plot(Auto$cylinders, Auto$mpg)
```



f) Looking at the pairs() plot, each of the predictors has some sort of relationship with miles per gallon.

Some look to be very-strongly correlated, like the Weight, Horsepower, and Number of Cylinders that were investigated in (e). Others - like the Name - seem to have weaker correlation, but even the Name seems to have some relationship with MPG, so I wouldn't immediately throw it out. So, yes, in a broad sense, my plots definitely suggest that the other variables in the Auto dataset could be useful in predicting MPG; and, in a narrow sense, the data suggest that Horsepower, Weight, and Number of Cylinders are most correlated with MPG.

Problem 4

a)

I got Residual SSE values of: 143.32, 163.2, 116.48, 83.04, 83.31, 97.3, 72.59, 108.55, 105.19, and 70.83, for an average of 104.381, from the simulations with model complexity = 1.

The range of the highest order coefficient was ≈ -5 to -3.

b)

I got Residual SSE values of: 23.48, 79.13, 31.09, 71.12, 58.05, 56.4, 67.85, 21.73, 65.7, and 28.55, for an average of 50.31, from the simulations with model complexity = 3.

The largest range of the coefficients was that of β_1, which was ≈ -1.5 to 10.

c)

I got Residual SSE values of: 11.28, 1.09, 7.33, 1.55, 1.79, 11.45, 2.97, 3.11, 6.47, and 4.53, for an average of 5.157, from the simulations with model complexity = 15.

The range of the coefficient β_6 was the largest. It was ≈ -1,200,000 to 1,750,000.

d)

My results illustrate the bias-variance trade-off in a few ways. First, for models of lower complexity, there's more error as evidenced by the Residual SSE; that is, the models' predictions aren't as accurate when run on the training data. As I added more parameters, the average Residual SSE of my simulations decreased. Another way of putting this is to say that, for these models of lower complexity, there's more *bias*: because of their simplicity, the models miss a lot of the data's intricacies. But, for the more complex models, the simulations showed an enormous amount of variance, as evidenced by the range of the coefficients. This range grew by magnitudes when more parameters were added. What this means is that, as we add more parameters, the models become more sensitive to the training data. And this - and the same is true of the models with lower complexity and their higher bias - will make models less accurate. So, in summation, my simulations showed that not having enough parameters leads to a high bias and underfitting, while having too many parameters leads to an extreme variance and overfitting, both of which subtract from a model's efficacy.

e)

For models of complexity 2, 3, and 4, I seem to get a curve fairly close to the true, overlaid curve. For models with complexity higher than that, the line seems to overfit to the training data; and, for models with complexity less than that, it's obviously always too linear. If I had to pick one, *I think I'd say the model with complexity = 3 gives the best approximation of the true curve.* For models with complexity = 2, the estimated line is often too linear and so misses some of the true curve's shape. And, for models of complexity = 4, there is a tendency for the model to sometimes overfit to the training data, as happened fairly often when I ran simulations of models with complexity > 4.