STATS216v Introduction to Statistical Learning
Stanford University, Summer 2017

Problem Set 1 (Solution)

Due: Friday, July 7

---

1. *Grading notes: 16 points total. 3 points each for parts a,b,c,e,f. 1 point for part d. For each part, assign one point for a correct answer to supervised/unsupervised, one point for a correct answer to classification/regression, and one point for the correct number of observations and predictors.*

   (a) It's supervised learning and it's regression. The idea is to do a regression using information of the students as predictors and the grades as observations and considering an indicator variable for being refugee and see whether the coefficient of that predictor is considerably greater than zero. There are 500 observations and 3 predictors.

   (b) It's a supervised learning problem and it's classification because they want to allocate people to different activities based on their information. There are 3000 training observations and 5 predictors.

   (c) It's supervised learning and regression. They want to model blood consumption with a regression model to predict the amount of blood they need for each day. There are 730 observations and 4 predictors.

   (d) This is unsupervised learning.

   (e) It is supervised learning because we have complete information about the sample points and it's regression because the idea is to estimate the amount of oil for any point in that area. There are 35 observations and 28 predictors.

   (f) It is supervised learning because we have the measurements for real earthquakes and also nuclear explosion. It is a classification problem because we want to label our data points with earthquake or nuclear explosion. There are 118 training data points and 600 predictors.

2. *Grading notes: 20 points total, 10 for each part. Assign full credit for reasonable answers.*

   (a) With supervised learning we usually have an outcome, and we want to build a model to predict the outcome. Our training data has both predictors and outcomes, and so can be used to supervise the learning of the model (to make sure it predicts well on the training data). With unsupervised learning we have no outcome to guide the model building, and so the problem is more vague: looking for patterns or structure in the data. For example in the last part of the previous problem set we wanted to label our data points as real earthquakes versus nuclear explosion and we had a set of labeled data that we could use. This is supervised. If we didnt have the labels, we might try to look for clusters in the time series — this would be unsupervised.

(b) In regression, we want to predict a continuously varying response, while in classification we simply want to assign a discrete class label. For a two-class problem, we could treat a 0-1 response as continuous, fit a linear regression, and then classify as 1 if the prediction is bigger than 0.5. The reverse would be unsatisfactory, since a binary classification would not capture all the variation in a continuous response.

3. You are a data science consultant! In each of the following cases decide whether you would suggest a flexible regression model or an inflexible one. Provide your reasons for your client as clear as possible.

   Grading notes: 20 points total, 5 for each part. Assign 5 points for correct answers with correct explanations, 3 points for incorrect answers with reasonable explanation, 3 points for correct answer with no or incorrect explanation.

   (a) The number of predictors is very large in camparison to the number of patients so I would suggest an inflexible regression model.

   (b) Since they are going to provide a huge data set with a small number of predictors a flexible regression model is good.

   (c) It seems that she will have predictors with high varicance so it's better to consider an inflexible model.

   (d) Now with precise measurements she can control the variance of her predictors and therefore a flexible regression model makes more sense.

4. Problem 4

   Grading notes: 21 points total: 3 points each for parts a,b,d,e,f,g,h

   (a) *grading notes: verify that the student used* `read.csv()`

   (b) We can use read.csv(header = FALSE)

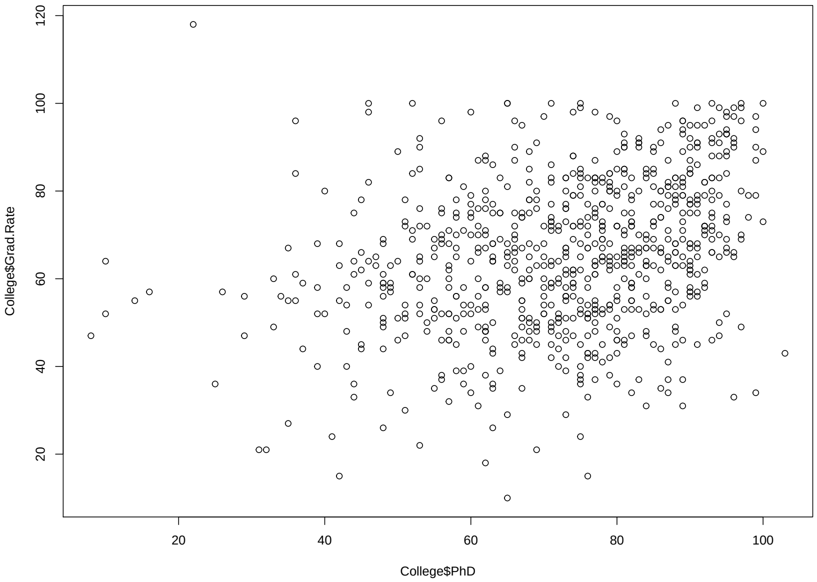
   (c) Nothing to grade

   (d)

```
 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
```

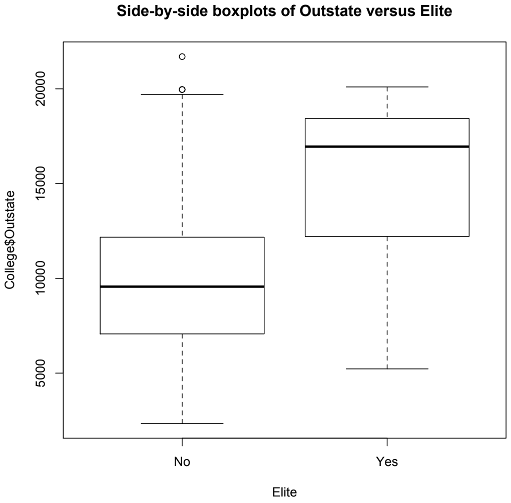
```
1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320
Median : 54.0   Median : 1707   Median :  353.0   Median : 9990
Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441
3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925
Max.   :100.0   Max.   :31643   Max.   :21836.0   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
```

(e) Scatterplot matrix:

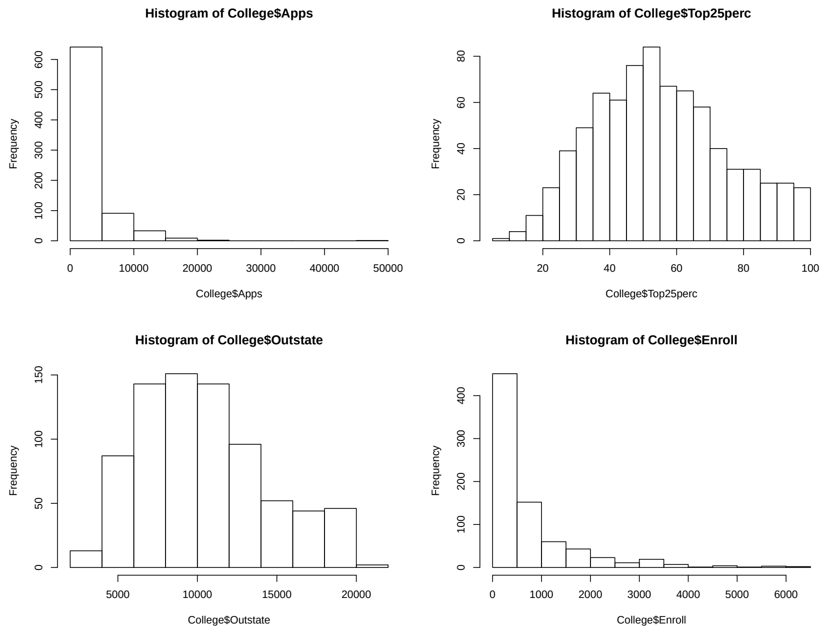


(f) There are 565 private schools.

(g) There are 78 elite schools. The boxplot should be the following:



(h) *grading notes: Students may choose different predictors, but should have a 2x2 grid* Sample output:



5. Problem 5

   Grading notes: 23 points total. 2 points each for parts a-d. 5 points each for parts e-f.

   (a) *Grading notes: Verify that the student used the load command*

   (b) The length of `train.y` is 1197 and the length of `test.y` is 625.

   (c) There are several answers to this question, but it is important to point out that the distribution is heavily skewed, or at least non-symmetric.

   (d) *Grading notes: Full credit for 2 or more pairwise plots*

   (f) The test error is 0.7527.

   (g) The main reason is overfitting: since we have a lot of predictors, it is possible that most of the patterns the model is finding is just noise. This can be seen through the bias-variance trade-off: because we are likely adding irrelevant predictors, our bias is not going down, but the variance is going up quite a bit, resulting in poor test error.

Appendix: Relevant R code

```
#                                 QUESTION 4


# d
summary(College)
```

```
# e
plot(College$PhD, College$Grad.Rate)

# f
length(College[which(College$Private == "Yes"), 1])
#alternative: sum(College$Private == "Yes")

# g
Elite = rep("No", nrow(College))
Elite[College$Top10perc > 50] = "Yes"
Elite = as.factor(Elite)

college = data.frame(College, Elite)

summary(Elite)

plot(College$Outstate ~ Elite,
        main = ’Side-by-side boxplots of Outstate versus Elite’)

# h
par(mfrow = c(2,2))
hist(College$Apps)
hist(College$Top25perc, breaks = 20)
hist(College$Outstate)
hist(College$Enroll)

#                               QUESTION 5

#(a)
load("als.RData")

#(b)
length(train.y)
length(test.y)

#(c)
summary(train.y)
hist(train.y, breaks = 40)

#(d)
colnames(train.X)[1:20]
pairs(train.X[, 21:25])

#(e)
lm.model = lm(train.y ~ ., data = train.X)
```

```
coef(summary(lm.model))[1:20, 1:4]
summary(lm.model)$r.squared

#(f)
lm.test.pred = predict(lm.model, test.X)
sqrt(mean((lm.test.pred - test.y)^2))
```