ST517-HW8

Nora Quick

1. For each of the following parts, indicate whether we would generally expect the performance of a flexible statistical learning method to be better or worse than an inflexible method. Justify your answers.

   (a) The sample size n is extremely large, and the number of predictors p is small.

For this I believe that a flexible model would perform better here because we are less likely to overfit even with it being flexible based on the large smaple size.

   (b) The number of predictors p is extremely large, and the number of observations n is small.

For this I believe that an inflexible model would perform better here because in this situation a flexible model would cause overfitting based on the small sample size.

   (c) The relationship between the predictors and response is highly non-linear.

For this I believe that I would use a flexible model because it would help find a non-linear effect.

   (d) The variance of the error terms, i.e.,

$$sigma2$$

   = Var(), is extremely high.

For this I believe that I would use an inflexible model because a flexible model would cause too much noise because of the extremely high variance.

2. We will now perform cross-validation on a simulated dataset.

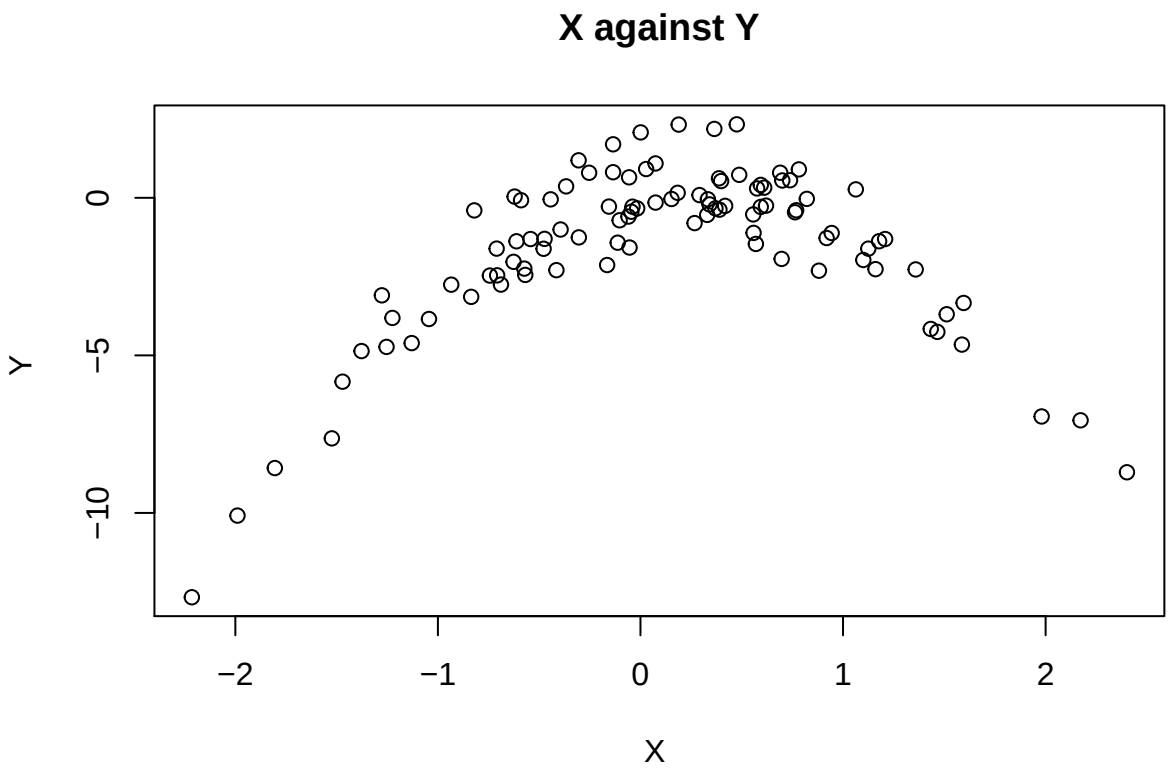
   (a) Generate a simulated data set as follows: In this data set, what is n and what is p? Write out the model used to generate the data in equation form.

```
set.seed(1)
x <- rnorm(100)
y <- x - 2*x^2 + rnorm(100)
```

Based on the above code I would say that n is 100 ("rnorm(100)") and p is 3 (1 = x, 2 = 2*x^2, and 3 = rnorm(100)).

   (b) Create a scatter plot of X against Y using the data you generated above. Comment on what you see.

```
plot(x, y, main="X against Y", xlab="X ", ylab="Y ")
```



I see an inverse parabolic shape that indicates that when Y reaches closer to 0 so does X.

(c) Set a random seed, and then compute the leave-one-out cross-validation (LOOCV) errors that result from fitting the following four models using least squares:

(i): Y = $$\beta0$$ + $$\beta1$$X + e

(ii): Y = $$\beta0$$ + $$\beta1$$X + $$\beta$$2X2 + e

(iii): Y = $$\beta0$$ + $$\beta1$$X + $$\beta$$2X2 + $$\beta$$3X3 + e

(iv): Y = $$\beta0$$ + $$\beta1$$X + $$\beta$$2X2 + $$\beta$$3X3 + $$\beta$$4X4 + e

```
set.seed(5)

# set df for err
df = data.frame(y,x)

# calculate
glm.fit1 <- glm(y ~ x)
cv.err1 <- cv.glm(df, glm.fit1)

glm.fit2 <- glm(y ~ poly(x,2))
cv.err2 <- cv.glm(df, glm.fit2)

glm.fit3 <- glm(y ~ poly(x,3))
```

```r
cv.err3 <- cv.glm(df, glm.fit3)

glm.fit4 <- glm(y ~ poly(x,4))
cv.err4 <- cv.glm(df, glm.fit4)

# print out
cv.err1$delta
```

```
## [1] 7.288162 7.284744
```

```r
cv.err2$delta
```

```
## [1] 0.9374236 0.9371789
```

```r
cv.err3$delta
```

```
## [1] 0.9566218 0.9562538
```

```r
cv.err4$delta
```

```
## [1] 0.9539049 0.9534453
```

(d) Repeat (c) using another random seed, and report your results. Are your results the same as what you got in (c)? Why or why not?

```r
set.seed(23)

# set df for err
df = data.frame(y,x)

# calculate
glm.fit1 <- glm(y ~ x)
cv.err1 <- cv.glm(df, glm.fit1)

glm.fit2 <- glm(y ~ poly(x,2))
cv.err2 <- cv.glm(df, glm.fit2)

glm.fit3 <- glm(y ~ poly(x,3))
cv.err3 <- cv.glm(df, glm.fit3)

glm.fit4 <- glm(y ~ poly(x,4))
cv.err4 <- cv.glm(df, glm.fit4)

# print out
cv.err1$delta
```

```
## [1] 7.288162 7.284744
```

```
cv.err2$delta
```

```
## [1] 0.9374236 0.9371789
```

```
cv.err3$delta
```

```
## [1] 0.9566218 0.9562538
```

```
cv.err4$delta
```

```
## [1] 0.9539049 0.9534453
```

The results of this part (part (d)) were the same as part (c). This is because it's the same data run in the same way so even when the seed is different the same actions are being taken.

(e) Which of the models in (c) had the smallest LOOCV error? Is this what you expected? Explain your answer.

Model 2 has the smallest error. Yes, I expected this because it's not just X but it isn't being help to a large power so it is reasonable that it has the smallest error.

(f) Comment on the statistical significance of the coefficient estimates that results from fitting each of the models in (c) using least squares. Do these results agree with the conclusions drawn based on the cross-validation results?

```
# calculate
lm1 <- lm(y~x, data=df)
lm2 <- lm(y~poly(x,2), data=df)
lm3 <- lm(y~poly(x,3), data=df)
lm4 <- lm(y~poly(x,4), data=df)

# print
summary(lm1)
```

```
##
## Call:
## lm(formula = y ~ x, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5161 -0.6800  0.6812  1.5491  3.8183
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.6254     0.2619  -6.205 1.31e-08 ***
## x             0.6925     0.2909   2.380   0.0192 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.6 on 98 degrees of freedom
## Multiple R-squared:  0.05465,    Adjusted R-squared:  0.045
## F-statistic: 5.665 on 1 and 98 DF,  p-value: 0.01924
```

```r
summary(lm2)
```

```
##
## Call:
## lm(formula = y ~ poly(x, 2), data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9650 -0.6254 -0.1288  0.5803  2.2700
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.5500     0.0958  -16.18  < 2e-16 ***
## poly(x, 2)1   6.1888     0.9580    6.46 4.18e-09 ***
## poly(x, 2)2 -23.9483     0.9580  -25.00  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.958 on 97 degrees of freedom
## Multiple R-squared:  0.873,  Adjusted R-squared:  0.8704
## F-statistic: 333.3 on 2 and 97 DF,  p-value: < 2.2e-16
```

```r
summary(lm3)
```

```
##
## Call:
## lm(formula = y ~ poly(x, 3), data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9765 -0.6302 -0.1227  0.5545  2.2843
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.55002    0.09626 -16.102  < 2e-16 ***
## poly(x, 3)1   6.18883    0.96263   6.429 4.97e-09 ***
## poly(x, 3)2 -23.94830    0.96263 -24.878  < 2e-16 ***
## poly(x, 3)3   0.26411    0.96263   0.274    0.784
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9626 on 96 degrees of freedom
## Multiple R-squared:  0.8731, Adjusted R-squared:  0.8691
## F-statistic: 220.1 on 3 and 96 DF,  p-value: < 2.2e-16
```

```r
summary(lm4)
```

```
##
## Call:
## lm(formula = y ~ poly(x, 4), data = df)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -2.0550 -0.6212 -0.1567  0.5952  2.2267
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.55002    0.09591 -16.162  < 2e-16 ***
## poly(x, 4)1   6.18883    0.95905   6.453 4.59e-09 ***
## poly(x, 4)2 -23.94830    0.95905 -24.971  < 2e-16 ***
## poly(x, 4)3   0.26411    0.95905   0.275    0.784
## poly(x, 4)4   1.25710    0.95905   1.311    0.193
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9591 on 95 degrees of freedom
## Multiple R-squared:  0.8753, Adjusted R-squared:  0.8701
## F-statistic: 166.7 on 4 and 95 DF,  p-value: < 2.2e-16
```

Model 1 having only the intercept be significant, model 2 having everything be significant, model 3 having everything but the last be significant, and model 4 where the first three are significant and the last two aren't.

Yes, this seems to match part (c) and (d) where the second model is the most significant.

3. The Boston data set is in the MASS package, you'll need to load that first.

```
library(MASS)
?Boston
head(Boston)
```

```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio  black lstat
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90  4.98
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90  9.14
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83  4.03
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63  2.94
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90  5.33
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12  5.21
##   medv
## 1 24.0
## 2 21.6
## 3 34.7
## 4 33.4
## 5 36.2
## 6 28.7
```

```
crim <- Boston$crim
zn <- Boston$zn
indus <- Boston$indus
chas <- Boston$chas
nox <- Boston$nox
rm <- Boston$rm
age <- Boston$age
dis <- Boston$dis
rad <- Boston$rad
```

```
tax <- Boston$tax
ptratio <- Boston$ptratio
black <- Boston$black
lstat <- Boston$lstat
medv <- Boston$mdev
```

(a) Your job is to build a regression model to predict the crime rate (crim) in Boston suburbs based on the other provided variables.

(i) A brief exploratory analysis (some summary statistics, and a few plots of any obvious relationshi

```
# calculate
regfit.best <- regsubsets(crim ~ ., data = Boston)

# summary
summary(regfit.best)
```

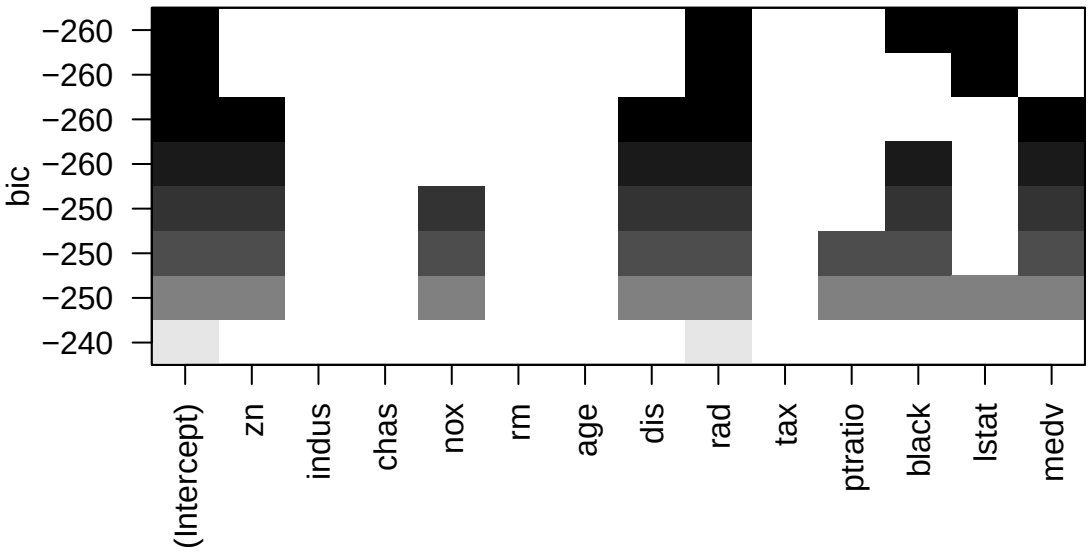
```
## Subset selection object
## Call: regsubsets.formula(crim ~ ., data = Boston)
## 13 Variables  (and intercept)
##         Forced in Forced out
## zn          FALSE      FALSE
## indus       FALSE      FALSE
## chas        FALSE      FALSE
## nox         FALSE      FALSE
## rm          FALSE      FALSE
## age         FALSE      FALSE
## dis         FALSE      FALSE
## rad         FALSE      FALSE
## tax         FALSE      FALSE
## ptratio     FALSE      FALSE
## black       FALSE      FALSE
## lstat       FALSE      FALSE
## medv        FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          zn  indus chas nox rm  age dis rad tax ptratio black lstat medv
## 1  ( 1 ) " " " "   " "  " " " " " " " " "*" " " " "     " "   " "   " "
## 2  ( 1 ) " " " "   " "  " " " " " " " " "*" " " " "     " "   "*"   " "
## 3  ( 1 ) " " " "   " "  " " " " " " " " "*" " " " "     "*"   "*"   " "
## 4  ( 1 ) "*" " "   " "  " " " " " " " " "*" "*" " " " "     " "   " "   "*"
## 5  ( 1 ) "*" " "   " "  " " " " " " " " "*" "*" " " " "     "*"   " "   "*"
## 6  ( 1 ) "*" " "   " "  "*" " " " " " " "*" "*" " " " "     "*"   " "   "*"
## 7  ( 1 ) "*" " "   " "  "*" " " " " " " "*" "*" " " "*"     "*"   " "   "*"
## 8  ( 1 ) "*" " "   " "  "*" " " " " " " "*" "*" " " "*"     "*"   "*"   "*"
```

```
# best
r <- lm(crim ~ dis + rad + medv + black, data = Boston)

# print
r
```

```
##
## Call:
## lm(formula = crim ~ dis + rad + medv + black, data = Boston)
##
## Coefficients:
## (Intercept)          dis          rad         medv        black
##    6.529163    -0.292778     0.483580    -0.141722    -0.009053
```

```
# plots
plot(regfit.best, scale="bic")
```



(ii) A description of the set of regression models you considered.

I chose those regression models (dis, rad, medv, black) because of their shown significance to crime.

(iii) A description of how the models were evaluated.

They were evaluated by fitting all the variables with resubsets to see which variables were significant. Once I saw which best predicted I fit the best ones with a regression model.

(iv) A summary of one (or a few) models that based on your analysis are the best among those you consi

I determined that dis, rad, medv, and black are the four best models to base the analysis on.

In summary there is a large significance for dis because it determines the distances between five Bostom employment centers from the population. This means how far away are citicens from a place where they can get help getting a job. The further the way they are the less easy it is for them to get help and, therefore, without a job and no help getting one crime is higher.