Project 1

Fabian Abrego

4/17/2020

##Part 1

For this part use the prostate dataset from the faraway package. Use ?prosate to learn about the dataset. The goal of this exercise is to find a model that is useful for explaining the response lpsa.

Fit a total of five models.

One must use all possible predictors. One must use only lcavol as a predictor. The remaining three you must choose. The models you choose must be picked in a way such that for any two of the five models, one is nested inside the other. Argue that one of the five models is the best among them for explaining the response. Use appropriate methods covered and justify your answer.

```r
prostate = faraway::prostate
model_all = lm(lpsa ~ ., data = prostate)
model_lcavol = lm(lpsa ~ lcavol, data = prostate)
model_5 = lm(lpsa ~ lcavol + lweight + age + lbph + lcp, data = prostate)
model_4 = lm(lpsa ~ lcavol + lweight + age + lbph, data = prostate)
model_2 = lm(lpsa ~ lcavol + lweight, data = prostate)
summary(model_all)
```

```
##
## Call:
## lm(formula = lpsa ~ ., data = prostate)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7331 -0.3713 -0.0170  0.4141  1.6381
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.669337   1.296387   0.516  0.60693
## lcavol       0.587022   0.087920   6.677 2.11e-09 ***
## lweight      0.454467   0.170012   2.673  0.00896 **
## age         -0.019637   0.011173  -1.758  0.08229 .
## lbph         0.107054   0.058449   1.832  0.07040 .
## svi          0.766157   0.244309   3.136  0.00233 **
## lcp         -0.105474   0.091013  -1.159  0.24964
## gleason      0.045142   0.157465   0.287  0.77503
## pgg45        0.004525   0.004421   1.024  0.30886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7084 on 88 degrees of freedom
## Multiple R-squared:  0.6548, Adjusted R-squared:  0.6234
## F-statistic: 20.86 on 8 and 88 DF,  p-value: < 2.2e-16
```

```r
summary(model_lcavol)
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol, data = prostate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.67625 -0.41648  0.09859  0.50709  1.89673
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.50730    0.12194   12.36   <2e-16 ***
## lcavol       0.71932    0.06819   10.55   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7875 on 95 degrees of freedom
## Multiple R-squared:  0.5394, Adjusted R-squared:  0.5346
## F-statistic: 111.3 on 1 and 95 DF,  p-value: < 2.2e-16
```

```r
summary(model_5)
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + age + lbph + lcp, data = prostate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.51922 -0.39020  0.00317  0.47268  1.75943
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.80750    0.87692   0.921  0.35957
## lcavol       0.62519    0.09017   6.933 5.78e-10 ***
## lweight      0.46363    0.17576   2.638  0.00981 **
## age         -0.01345    0.01134  -1.186  0.23870
## lbph         0.08493    0.06064   1.400  0.16477
## lcp          0.09044    0.07392   1.223  0.22432
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.746 on 91 degrees of freedom
## Multiple R-squared:  0.6041, Adjusted R-squared:  0.5823
## F-statistic: 27.77 on 5 and 91 DF,  p-value: < 2.2e-16
```

```r
summary(model_4)
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + age + lbph, data = prostate)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4885 -0.4241 -0.0001  0.4031  1.8073
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.73074    0.87703   0.833   0.4069
## lcavol       0.69854    0.06754  10.343   <2e-16 ***
## lweight      0.45770    0.17617   2.598   0.0109 *
## age         -0.01371    0.01137  -1.206   0.2309
## lbph         0.08404    0.06080   1.382   0.1702
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.748 on 92 degrees of freedom
## Multiple R-squared:  0.5976, Adjusted R-squared:  0.5801
## F-statistic: 34.15 on 4 and 92 DF,  p-value: < 2.2e-16
```

```r
summary(model_2)
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight, data = prostate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.61965 -0.50778 -0.02095  0.52291  1.89885
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.30262    0.56904  -0.532  0.59612
## lcavol       0.67753    0.06626  10.225  < 2e-16 ***
## lweight      0.51095    0.15726   3.249  0.00161 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7506 on 94 degrees of freedom
## Multiple R-squared:  0.5859, Adjusted R-squared:  0.5771
## F-statistic: 66.51 on 2 and 94 DF,  p-value: < 2.2e-16
```

Model All:
RSE = .7084 R^2 = .6548 Adjusted R^2 = .6234

Model lcavol:
RSE = .7875
R^2 = .5394
Adjusted R^2 = .5346

Model 5:
RSE = .746
R^2 = .6041
Adjusted R^2 = .5823

Model 4:
RSE = .748
R^2 = .5976
Adjusted R^2 = .5801

Model 2:
RSE = .7506

Rˆ2 = .5859
Adjusted Rˆ2 = .5771

Based on these 5 models, the model that best explains the response *lpsa* is the model that includes *all* predictors. The conclusion was formulated on the basis of each model's respective RSE, Rˆ2, and Adjusted Rˆ2. To explain further, in the case of RSE, the smaller the value the better as the RSE basically explains the amount of spread in error. Based on that, this model surpasses the others as it has the smallest RSE. In the case of Rˆ2 and Adjusted Rˆ2, their values are made to be interpreted as the percentage of varience in the model that is explained by the choosen predictors. In this case, the higher the value for these variables the better and for this model, such holds true as it has the highest valued variables.

##Part 2

```
boston = MASS::Boston
library(MASS)
set.seed(42)
train_index = sample(1:nrow(Boston), 400)
train_data = boston[train_index,]
test_data = boston[-train_index,]
Model_ALL = lm(medv ~ ., data = train_data)
Model_crim = lm(medv ~ crim, data = train_data)
Model_6 = lm(medv ~ crim + indus + nox + rm + age + dis, data = train_data)
Model_5 = lm(medv ~ crim + indus + nox + rm + age, data = train_data)
Model_3 = lm(medv ~ crim + nox + dis, data = train_data)
Y = test_data[,14]

#Model All

RSE_ALL = summary(Model_ALL)$sigma
Train_RMSE_ALL = sqrt((RSE_ALL^2)*Model_ALL$df.residual/length(Model_ALL$fitted.values))

beta_ALL = as.vector(Model_ALL$coefficients)
X_ALL = cbind(1,test_data[,-14])
Y_hat_ALL = as.matrix(X_ALL) %*% beta_ALL
SSE_ALL = sum((Y - Y_hat_ALL)^2)
Test_RMSE_ALL = sqrt(SSE_ALL/length(Y))

#Train RMSE = 4.675465 Test RMSE = 4.767746


#Model crim

RSE_crim = summary(Model_crim)$sigma
Train_RMSE_crim = sqrt((RSE_crim^2)*Model_crim$df.residual/length(Model_crim$fitted.values))

beta_crim = as.vector(Model_crim$coefficients)
X_crim = cbind(1,test_data[,1])
Y_hat_crim = as.matrix(X_crim) %*% beta_crim
SSE_crim = sum((Y - Y_hat_crim)^2)
Test_RMSE_crim = sqrt(SSE_crim/length(Y))

#Train RMSE = 8.238496  Test RMSE = 9.318085


#Model 6
```

```r
RSE_6 = summary(Model_6)$sigma
Train_RMSE_6 = sqrt((RSE_6^2)*Model_6$df.residual/length(Model_6$fitted.values))

beta_6 = as.vector(Model_6$coefficients)
X_6 = cbind(1,test_data[,c("crim","indus","nox","rm","age","dis")])
Y_hat_6 = as.matrix(X_6) %*% beta_6
SSE_6 = sum((Y - Y_hat_6)^2)
Test_RMSE_6 = sqrt(SSE_6/length(Y))

#Train RMSE = 5.758958  Test RMSE = 5.95507


#Model 5

RSE_5 = summary(Model_5)$sigma
Train_RMSE_5 = sqrt((RSE_5^2)*Model_5$df.residual/length(Model_5$fitted.values))

beta_5 = as.vector(Model_5$coefficients)
X_5 = cbind(1,test_data[,c("crim","indus","nox","rm","age")])
Y_hat_5 = as.matrix(X_5) %*% beta_5
SSE_5 = sum((Y - Y_hat_5)^2)
Test_RMSE_5 = sqrt(SSE_5/length(Y))

#Train RMSE = 5.995325  Test RMSE = 6.148281


#Model 3

RSE_3 = summary(Model_3)$sigma
Train_RMSE_3 = sqrt((RSE_3^2)*Model_3$df.residual/length(Model_3$fitted.values))

beta_3 = as.vector(Model_3$coefficients)
X_3 = cbind(1,test_data[,c("crim","nox","dis")])
Y_hat_3 = as.matrix(X_3) %*% beta_3
SSE_3 = sum((Y - Y_hat_3)^2)
Test_RMSE_3 = sqrt(SSE_3/length(Y))

#Train RMSE = 7.72839 Test RMSE = 8.643548

ALL = c(Test_RMSE_ALL, Train_RMSE_ALL) # Variance = .004258
crim = c(Test_RMSE_crim, Train_RMSE_crim) #Variance = .58276
Six = c(Test_RMSE_6, Train_RMSE_6) #Variance = .01923
Five = c(Test_RMSE_5, Train_RMSE_5) #Variance = .01170
Three = c(Test_RMSE_3, Train_RMSE_3) #Variance = .41876
summary(Model_ALL)
```

```
##
## Call:
## lm(formula = medv ~ ., data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.3126  -2.7134  -0.5522   1.5431  25.5431
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  40.211363   5.823305   6.905 2.07e-11 ***
## crim         -0.121911   0.034032  -3.582 0.000384 ***
## zn            0.037754   0.016166   2.335 0.020038 *
## indus         0.002787   0.069150   0.040 0.967867
## chas          1.918167   0.999327   1.919 0.055663 .
## nox         -17.987178   4.304668  -4.179 3.63e-05 ***
## rm            3.478935   0.457299   7.608 2.16e-13 ***
## age          -0.003087   0.014798  -0.209 0.834880
## dis          -1.456826   0.230828  -6.311 7.60e-10 ***
## rad           0.310637   0.074539   4.167 3.81e-05 ***
## tax          -0.011081   0.004234  -2.617 0.009212 **
## ptratio      -0.996107   0.148701  -6.699 7.45e-11 ***
## black         0.007692   0.003214   2.393 0.017194 *
## lstat        -0.533910   0.055318  -9.652  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.759 on 386 degrees of freedom
## Multiple R-squared:  0.7262, Adjusted R-squared:  0.7169
## F-statistic: 78.73 on 13 and 386 DF,  p-value: < 2.2e-16
```

```r
summary(Model_crim)
```

```
##
## Call:
## lm(formula = medv ~ crim, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.734  -5.147  -1.788   2.329  29.619
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 23.80157    0.44704  53.243  < 2e-16 ***
## crim        -0.37045    0.04425  -8.372 9.75e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.259 on 398 degrees of freedom
## Multiple R-squared:  0.1497, Adjusted R-squared:  0.1476
## F-statistic: 70.09 on 1 and 398 DF,  p-value: 9.753e-16
```

```r
summary(Model_6)
```

```
##
## Call:
## lm(formula = medv ~ crim + indus + nox + rm + age + dis, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.219  -3.166  -0.600   2.097  37.638
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.93105    4.52606  -0.206 0.837125
## crim         -0.19154    0.03456  -5.543 5.47e-08 ***
## indus        -0.23006    0.07173  -3.207 0.001451 **
## nox         -12.17993    4.66224  -2.612 0.009334 **
## rm            6.90354    0.45293  15.242  < 2e-16 ***
## age          -0.06287    0.01689  -3.723 0.000226 ***
## dis          -1.45760    0.25404  -5.738 1.92e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.81 on 393 degrees of freedom
## Multiple R-squared:  0.5845, Adjusted R-squared:  0.5782
## F-statistic: 92.15 on 6 and 393 DF,  p-value: < 2.2e-16
```

```
summary(Model_5)
```

```
##
## Call:
## lm(formula = medv ~ crim + indus + nox + rm + age, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.127  -3.120  -0.847   2.174  39.005
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.64406    3.60183  -4.899 1.41e-06 ***
## crim         -0.18183    0.03589  -5.067 6.23e-07 ***
## indus        -0.11816    0.07177  -1.646    0.101
## nox          -3.78538    4.60257  -0.822    0.411
## rm            7.28048    0.46594  15.625  < 2e-16 ***
## age          -0.02127    0.01586  -1.341    0.181
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.041 on 394 degrees of freedom
## Multiple R-squared:  0.5497, Adjusted R-squared:  0.544
## F-statistic:  96.2 on 5 and 394 DF,  p-value: < 2.2e-16
```

```
summary(Model_3)
```

```
##
## Call:
## lm(formula = medv ~ crim + nox + dis, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.488  -4.947  -2.063   2.625  29.138
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  48.09033    3.85312  12.481  < 2e-16 ***
## crim         -0.26589    0.04577  -5.809 1.29e-08 ***
## nox         -37.35806    5.35091  -6.982 1.24e-11 ***
## dis          -1.02654    0.29019  -3.537 0.000452 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.767 on 396 degrees of freedom
## Multiple R-squared:  0.2518, Adjusted R-squared:  0.2461
## F-statistic: 44.42 on 3 and 396 DF,  p-value: < 2.2e-16
```

Model All Train RMSE = 4.675465 Test RMSE = 4.767746
Variance = .004258
R^2= .7262 Adjusted R^2 = .7169

Model crim Train RMSE = 8.238496 Test RMSE = 9.318085
Variance = .58276
R^2 = .1497 Adjusted R^2 = .1476

Model 6 Train RMSE = 5.758958 Test RMSE = 5.95507
Variance = .01923
R^2 = .5845 Adjusted R^2 = .5782

Model 5 Train RMSE = 5.995325 Test RMSE = 6.148281
Variance = .01170
R^2 = .5497 Adjusted R^2 = .544

Model 3 Train RMSE = 7.72839 Test RMSE = 8.643548
Variance = .41876
R^2 = .2518 Adjusted R^2 = .2461

The model that is the best for predicting the response variable *medv* is the *Model ALL* which contains all predictors. This is based on the fact that it had the smallest RMSE for train and test data. This is important because the smaller the RMSE, the less error is attributed to the model. Secondly, this is due to this model having the smallest variance, or spread, between the RMSE calculated from the test data and the train data. Lastly, this choice is based on the fact that such model has the highest R^2 and Adjusted R^2 values. This is important because these values both essentially give the percentage of variation in the response variable that is described by the model.

##Part 3

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4} + \epsilon_i$$

$$\epsilon_i \sim N(0, \sigma^2).$$

$$\beta_0 = 2$$

$$\beta_1 = 3$$

$$\beta_2 = 4$$

$$\beta_3 = 0$$

$$\beta_4 = 1$$

$$\sigma^2 = 16$$

#Part A

```
set.seed(42)
n = 25
x0 = rep(1,n)
x1 = runif(n,min =  0, max =  10)
x2 = runif(n,min =  0, max =  10)
x3 = runif(n,min =  0, max =  10)
x4 = runif(n,min =  0, max =  10)
x = cbind(x0,x1,x2,x3,x4)
c = solve(t(x) %*% x)
y = rep(0, n)
ex_4_data = data.frame(y,x1,x2,x3,x4)


diag(c)
```

```
##          x0          x1          x2          x3          x4
## 0.744784994 0.004573055 0.005091328 0.005898213 0.005058979
```

```
ex_4_data[10,]
```

```
##    y       x1         x2       x3       x4
## 10 0 7.050648 0.03948339 5.144129 7.758234
```

#Part B

```
beta_hat_1 = numeric(1500)
beta_2_pval = numeric(1500)
beta_3_pval = numeric(1500)
```

#Part C

```
for (i in 1:1500) {
ex_4_data[,1] = 2 + 3*x1 + 4*x2 + 0*x3 + x4 + rnorm(n, 0, 4)
y = ex_4_data[,1]
model = lm(y ~ x1 + x2 + x3 + x4)
beta_hat_1[i] = summary(model)$coef[2,1]
beta_2_pval[i] = summary(model)$coef[3,4]
beta_3_pval[i] =summary(model)$coef[4,4]
}
```

#Part D

```
var_beta_1 = c * 16
var_beta_1[2,2] #0.07316889
```

```
## [1] 0.07316889
```

```
sd_beta_1 = sqrt(var_beta_1[2,2])
```

$$\hat{\beta}_1$$

is normally distributed with a mean of 3 (the same value used to construct the model) and a variance of 0.07316889.

#Part E

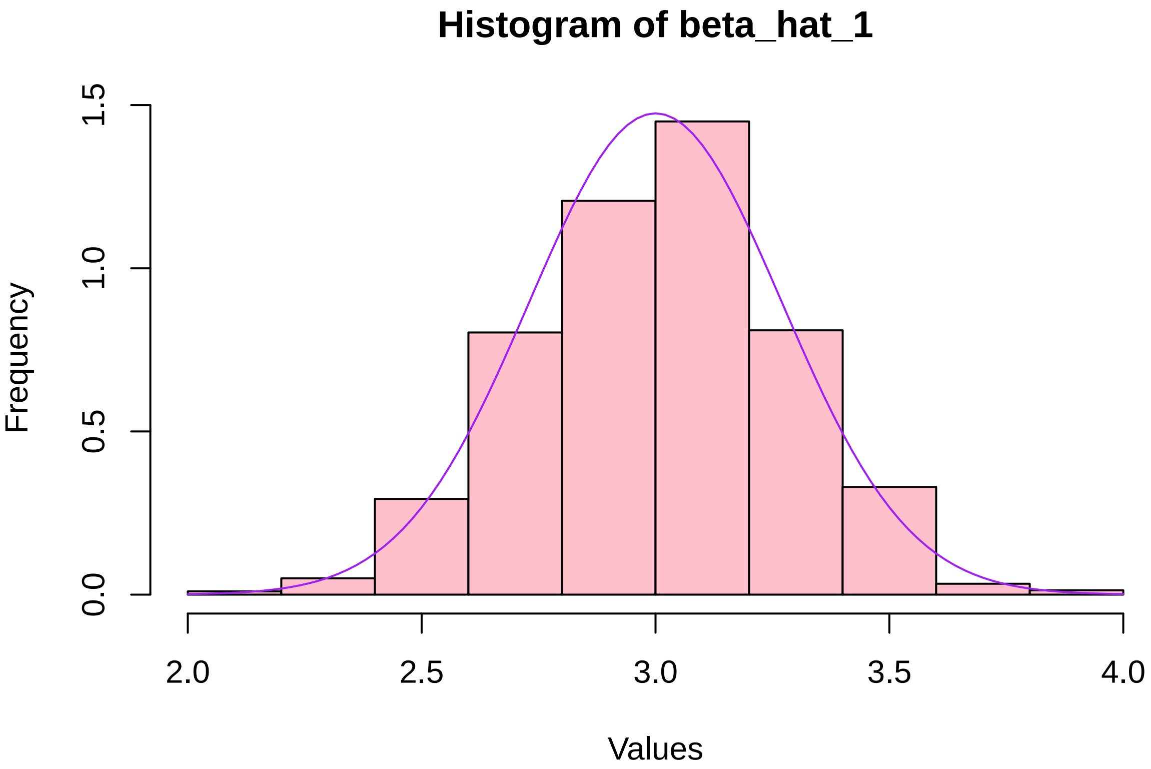
```
mean(beta_hat_1)
```

```
## [1] 3.006391
```

```r
var(beta_hat_1)
```

```
## [1] 0.07303341
```

```r
hist(beta_hat_1, xlab = "Values", ylab = "Frequency", probability = TRUE, col = "pink")
curve(dnorm(x, mean = 3, sd = sd_beta_1), add = TRUE, col = "purple")
```



The mean of `beta_hat_1` is equal to 3.006391 and the variance is equal to 0.07303341. This is close to the values of the true distribution of it and thus, close to what we would expect. The curve seems to strongly resemble the histogram.

#Part F

```r
length(which(beta_3_pval < 0.05))/1500
```

```
## [1] 0.04666667
```

The proportion of p values for beta hat 3 that are less than .05 is about .047. This is significant because when it comes to testing whether or not beta hat 3 is equal to zero, majority of tests would fail to reject such null hypothesis at a 5% significance level. This is important to consider because the actual value of beta 3 is indeed 0.

#Part G

```r
length(which(beta_2_pval < 0.05))/1500
```

```
## [1] 1
```

The proportion of beta hat 2 values that are smaller than .05 is all. This is important because on the basis of hypothesis testing, each test would reject the null hypothesis of beta hat 2 being equal to zero. This is important because the actual value of beta 2 is 4.