Exam 3 - Question 3

Adam McQuistan

Sunday, May 01, 2016

Question 3 - Problem 9.31

Part A

Select a model based off stepwise regression. To select the best model the full data set will be split up into two equal parts then the model will be built off the first set and validated against the second set

For forward stepwise regression it is important to identify an α cut off for determining which predictors to let into the model. For example, if your cut of is 0.05 then you would only include variables with pvalues below the variable.

```
library(MASS)
library(dplyr)
df <- read.csv(file="data/9.31.csv")
df$id <- NULL
idx1 <- seq(1,dim(df)[1], by=2)
idx2 <- seq(2,dim(df)[1], by=2)

dfValidate <- df[idx1,]
dfTrain <- df[idx2,]

nullModel <- lm(Sales ~ 1, data=dfTrain) # just the intercept
fullModel <- lm(Sales ~ ., data=dfTrain) # all parameters
addterm(nullModel, scope=fullModel, test="F")
```

```
## Single term additions
##
## Model:
## Sales ~ 1
##           Df  Sum of Sq        RSS    AIC F Value     Pr(F)
## <none>                  4.9118e+12 6176.8
## FinisSq    1 3.3251e+12 1.5868e+12 5883.9  542.74 < 2.2e-16 ***
## No_Bed     1 9.1410e+11 3.9977e+12 6125.0   59.22 2.964e-13 ***
## No_Bath    1 2.1367e+12 2.7752e+12 6029.8  199.41 < 2.2e-16 ***
## AirCon     1 4.8251e+11 4.4293e+12 6151.8   28.21 2.340e-07 ***
## Gara_Size  1 1.4498e+12 3.4620e+12 6087.5  108.47 < 2.2e-16 ***
## Pool       1 1.3411e+11 4.7777e+12 6171.6    7.27  0.007471 **
## YearBuilt  1 1.5784e+12 3.3334e+12 6077.6  122.64 < 2.2e-16 ***
## Quality    1 2.8273e+12 2.0845e+12 5955.1  351.29 < 2.2e-16 ***
## Style      1 6.8064e+11 4.2312e+12 6139.8   41.66 5.310e-10 ***
## LotSize    1 1.6358e+11 4.7482e+12 6169.9    8.92  0.003086 **
## AdjHw      1 2.3659e+10 4.8882e+12 6177.5    1.25  0.263913
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Begin by adding in the parameter with the highest F value (lowest p-value) which is Finished Square Feet.

```r
newModel <- lm(Sales ~ FinisSq, data=dfTrain)
addterm(newModel, scope=fullModel, test="F")
```

```
## Single term additions
##
## Model:
## Sales ~ FinisSq
##            Df  Sum of Sq        RSS    AIC F Value     Pr(F)
## <none>                   1.5868e+12 5883.9
## No_Bed      1 2.0559e+10 1.5662e+12 5882.5   3.387   0.06687 .
## No_Bath     1 1.8972e+10 1.5678e+12 5882.7   3.122   0.07842 .
## AirCon      1 3.5823e+10 1.5509e+12 5879.9   5.959   0.01531 *
## Gara_Size   1 1.2849e+11 1.4583e+12 5863.8  22.733 3.114e-06 ***
## Pool        1 3.2333e+09 1.5835e+12 5885.3   0.527   0.46862
## YearBuilt   1 2.1228e+11 1.3745e+12 5848.4  39.847 1.191e-09 ***
## Quality     1 4.0263e+11 1.1841e+12 5809.5  87.727 < 2.2e-16 ***
## Style       1 1.2406e+11 1.4627e+12 5864.6  21.883 4.680e-06 ***
## LotSize     1 4.0287e+10 1.5465e+12 5879.1   6.721   0.01007 *
## AdjHw       1 2.1509e+09 1.5846e+12 5885.5   0.350   0.55452
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Next add in Quality.

```r
newModel <- lm(Sales ~ FinisSq + Quality, data=dfTrain)
addterm(newModel, scope=fullModel, test="F")
```

```
## Single term additions
##
## Model:
## Sales ~ FinisSq + Quality
##            Df  Sum of Sq        RSS    AIC F Value     Pr(F)
## <none>                   1.1841e+12 5809.5
## No_Bed      1 1.2274e+10 1.1718e+12 5808.7  2.6919   0.10208
## No_Bath     1 6.2788e+09 1.1778e+12 5810.1  1.3700   0.24290
## AirCon      1 3.8340e+09 1.1803e+12 5810.6  0.8348   0.36173
## Gara_Size   1 2.5053e+10 1.1591e+12 5805.9  5.5549   0.01918 *
## Pool        1 9.4309e+08 1.1832e+12 5811.3  0.2049   0.65122
## YearBuilt   1 3.1373e+10 1.1527e+12 5804.5  6.9944   0.00868 **
## Style       1 7.4353e+10 1.1098e+12 5794.5 17.2186 4.533e-05 ***
## LotSize     1 3.0143e+10 1.1540e+12 5804.7  6.7131   0.01012 *
## AdjHw       1 6.6343e+09 1.1775e+12 5810.0  1.4480   0.22995
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Next add in Style

```r
newModel <- lm(Sales ~ FinisSq + Quality + Style, data=dfTrain)
addterm(newModel, scope=fullModel, test="F")
```

```
## Single term additions
```

```
##
## Model:
## Sales ~ FinisSq + Quality + Style
##           Df  Sum of Sq        RSS    AIC F Value    Pr(F)
## <none>                  1.1098e+12 5794.5
## No_Bed     1 8.8764e+09 1.1009e+12 5794.4  2.0641 0.152023
## No_Bath    1 8.1325e+08 1.1090e+12 5796.3  0.1877 0.665172
## AirCon     1 4.7387e+09 1.1050e+12 5795.4  1.0978 0.295735
## Gara_Size  1 1.7493e+10 1.0923e+12 5792.4  4.0999 0.043924 *
## Pool       1 2.7294e+09 1.1070e+12 5795.9  0.6312 0.427666
## YearBuilt  1 3.2841e+10 1.0769e+12 5788.7  7.8067 0.005599 **
## LotSize    1 2.0607e+10 1.0892e+12 5791.6  4.8435 0.028643 *
## AdjHw      1 1.2347e+10 1.0974e+12 5793.6  2.8802 0.090892 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now add in LotSize

```r
newModel <- lm(Sales ~ FinisSq + Quality + Style + LotSize, data=dfTrain)
addterm(newModel, scope=fullModel, test="F")
```

```
## Single term additions
##
## Model:
## Sales ~ FinisSq + Quality + Style + LotSize
##           Df  Sum of Sq        RSS    AIC F Value     Pr(F)
## <none>                  1.0892e+12 5791.6
## No_Bed     1 8.4880e+09 1.0807e+12 5791.6  2.0029 0.1582225
## No_Bath    1 7.5052e+08 1.0884e+12 5793.5  0.1758 0.6753278
## AirCon     1 2.1324e+09 1.0870e+12 5793.1  0.5002 0.4800475
## Gara_Size  1 1.8281e+10 1.0709e+12 5789.2  4.3531 0.0379347 *
## Pool       1 1.4514e+09 1.0877e+12 5793.3  0.3403 0.5601890
## YearBuilt  1 4.7391e+10 1.0418e+12 5782.0 11.6003 0.0007659 ***
## AdjHw      1 1.3187e+10 1.0760e+12 5790.5  3.1254 0.0782787 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

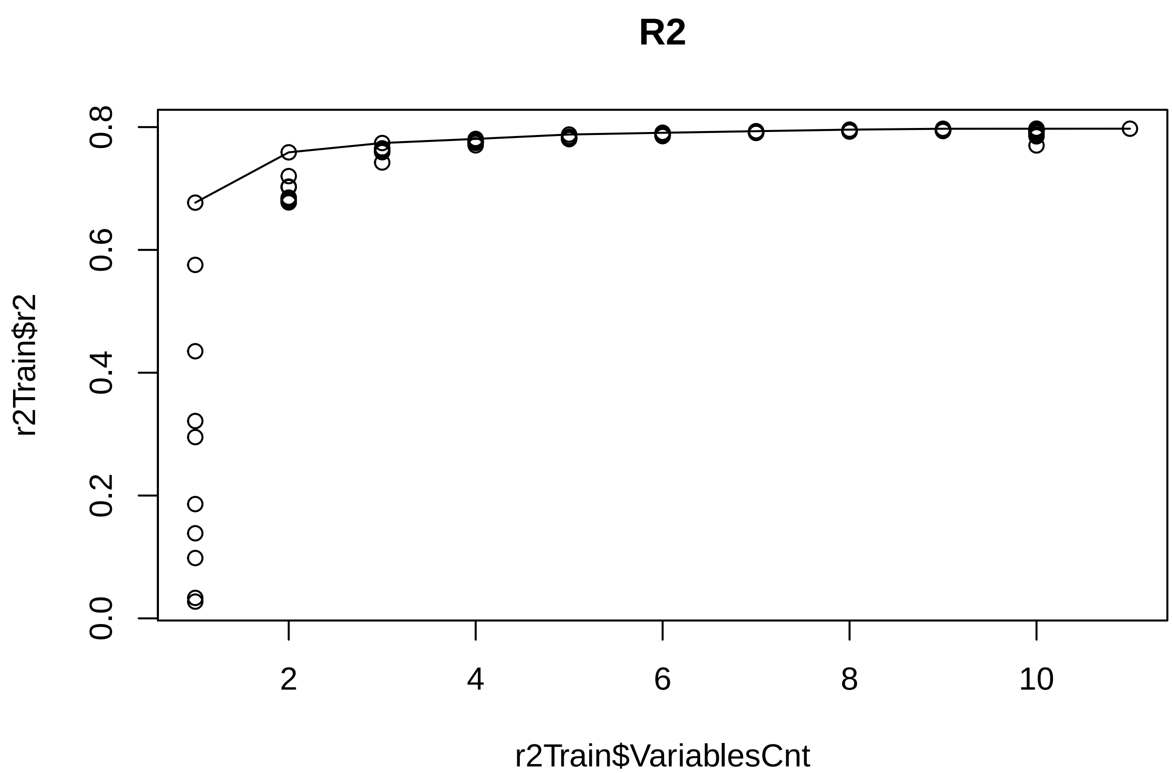
This time add in YearBuilt

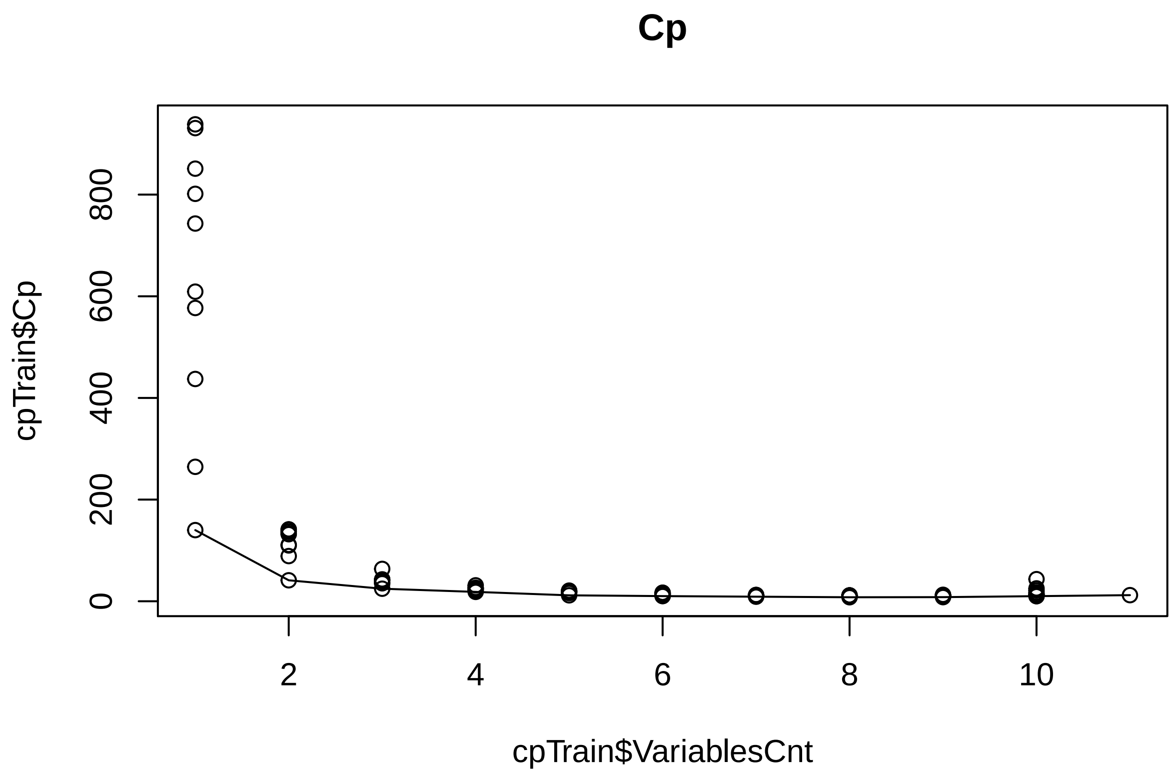
```r
newModel <- lm(Sales ~ FinisSq + Quality + Style + LotSize + YearBuilt, data=dfTrain)
addterm(newModel, scope=fullModel, test="F")
```

```
## Single term additions
##
## Model:
## Sales ~ FinisSq + Quality + Style + LotSize + YearBuilt
##           Df  Sum of Sq        RSS    AIC F Value   Pr(F)
## <none>                  1.0418e+12 5782.0
## No_Bed     1 9.6190e+09 1.0322e+12 5781.6  2.3671 0.12516
## No_Bath    1 1.9295e+09 1.0398e+12 5783.6  0.4713 0.49301
## AirCon     1 1.2587e+10 1.0292e+12 5780.9  3.1066 0.07918 .
## Gara_Size  1 8.7423e+09 1.0330e+12 5781.8  2.1495 0.14385
```

```
## Pool       1 6.9659e+08 1.0411e+12 5783.9  0.1700 0.68050
## AdjHw      1 1.3966e+10 1.0278e+12 5780.5  3.4513 0.06436 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

A quick plot of R^2 gains and Cp reductions per number of parameters will help verify model quality.





There does not appear to be any significant improvement after adding 5 parameters.

The Best Model is:

$Sales = \beta_0 + \beta_1 FinisSq + \beta_2 Quality + \beta_3 Style + \beta_4 LotSize + \beta_5 YearBuilt + \varepsilon$

Part B and C

The two models are listed below. For dividing the data set, this was done to build the model for part A.

Model1

$Sales = \beta_0 + \beta_1 FinisSq + \beta_2 Quality + \beta_3 Style + \beta_4 LotSize + \beta_5 YearBuilt + \varepsilon$

Model2

$Sales = \beta_0 + \beta_1 FinisSq + \beta_2 Quality + \beta_3 Style + \beta_4 LotSize + \varepsilon$

Part D

Test the above model(s) for validation

Model 1 Training

```
print(result1T)
```

```
##
## Call:
## lm(formula = Sales ~ FinisSq + Quality + Style + LotSize + YearBuilt,
##     data = dfTrain)
##
## Coefficients:
## (Intercept)      FinisSq      Quality        Style      LotSize
##  -1.877e+06    1.284e+02   -6.038e+04   -7.850e+03    1.056e+00
##   YearBuilt
##   1.016e+03
```

Model 1 Validation

```r
print(result1V)
```

```
##
## Call:
## lm(formula = Sales ~ FinisSq + Quality + Style + LotSize + YearBuilt,
##     data = dfValidate)
##
## Coefficients:
## (Intercept)      FinisSq      Quality        Style      LotSize
##  -2.913e+06    1.366e+02   -3.995e+04   -1.149e+04    1.367e+00
##   YearBuilt
##   1.511e+03
```

Model 2 Training

```r
print(result2T)
```

```
##
## Call:
## lm(formula = Sales ~ FinisSq + Quality + Style + LotSize, data = dfTrain)
##
## Coefficients:
## (Intercept)      FinisSq      Quality        Style      LotSize
##   1.572e+05    1.312e+02   -7.634e+04   -7.919e+03    7.888e-01
```

Model 2 Validation

```r
print(result2V)
```

```
##
## Call:
## lm(formula = Sales ~ FinisSq + Quality + Style + LotSize, data = dfValidate)
##
## Coefficients:
## (Intercept)      FinisSq      Quality        Style      LotSize
##   1.286e+05    1.367e+02   -6.667e+04   -1.166e+04    9.553e-01
```

Model Comparison Summary Table

```
library(knitr)
options(scipen=999)
kable(df_test)
```

Statistic	Model1Train	Model1Validate	Model2Train	Model2Validate
p	5.000	5.000	4.000	4.000
SSEp	1041769452275.179	1076728269630.670	1089160894174.772	1188701911390.270
PRESSp	1120878606971.817	1163489697231.443	1152112872881.770	1273591769971.398
Cp	11.621	11.412	18.417	23.589
MSEp	4085370401.079	4222463802.473	4254534742.870	4643366841.368
R2p	0.788	0.785	0.781	0.773

Since both the training and validation model for the 5 paramter model (model 1) has a PRESSp value closest to SSEp, lowest Cp values, and highest R2p values it is the better of the two models.