YK_Assignment9

Yinan Kang

4/22/2019

```r
# Packages
require(leaps)
require(car)
```

Problem 10.12 - Commercial Properties

```r
# Import Data
rm(list=ls())
colnames <- c("y","x1","x2","x3","x4")
df <- read.table(url("http://users.stat.ufl.edu/~rrandles/sta4210/Rclassnotes/data/textdatasets/KutnerDa
n <- nrow(df)
attach(df)

# Fitting model
model <- lm(y ~ x1+x2+x3+x4)
p=5

# Studentized deleted residuals
stud.del.resid <- rstudent(model)

# Manually doublechecking 'outlierTest()' results
max(abs(stud.del.resid))
```

```
## [1] 3.072105
```

```r
t.crit <- qt(1-0.01/(2*n),n-p-1) # p = 5
```

DECISION Rule: If the value of a studentized deleted residual is > t-critical value ('t.crit'), that observation can be said to be an outlier at the 0.01 significance level.

CONCLUSION: We see that the value of the largest absolute value studentized deleted residual IS LESS THAN t-critical value, thus NO outliers exist using this test at the 0.01 alpha level. ['outlierTest()' from 'car' package validates this *Conclusion*.]

```r
# Bonferroni outlier test
outlierTest(model, cutoff=0.01)
```

```
## No Studentized residuals with Bonferonni p < 0.01
## Largest |rstudent|:
##    rstudent unadjusted p-value Bonferonni p
## 6 -3.072105          0.0029614      0.23988
```

(a) Hat values and Outlier Detection

```r
hat.diag <- hatvalues(model)
sum(hat.diag) # Verify sum of diagonal hat values = p = 5
```

```
## [1] 5
```

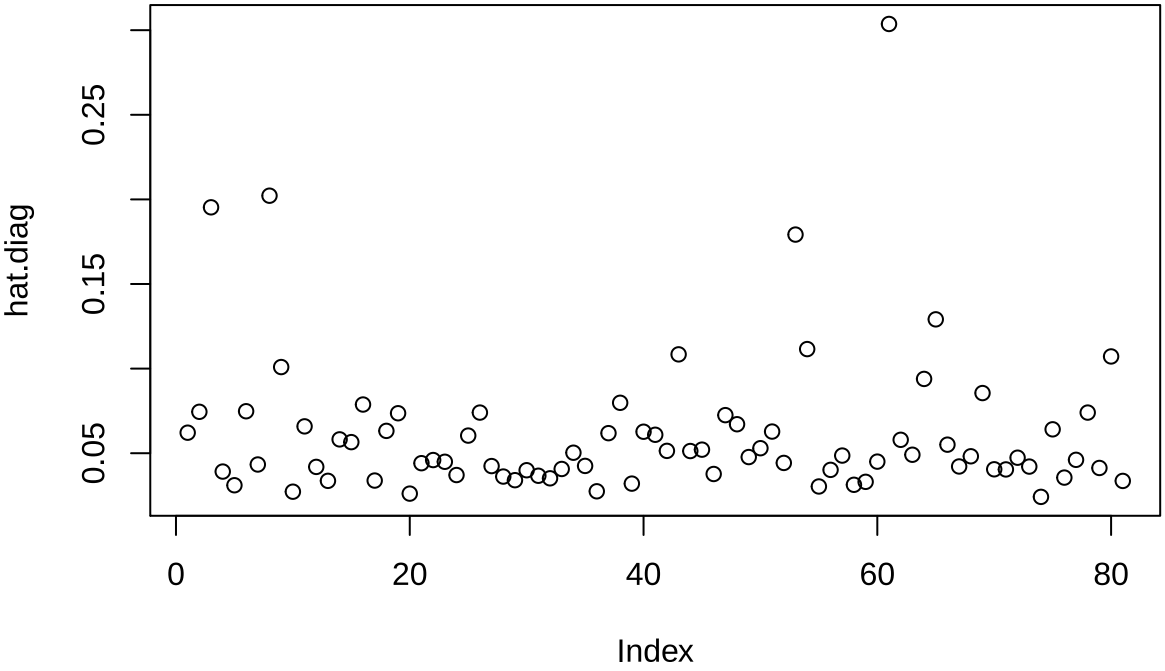
```
# If any diagonal hat value is > 2p/n, that is considered high leverage
print(2*p/n)
```

```
## [1] 0.1234568
```

```
print(max(hat.diag))
```

```
## [1] 0.3036714
```

```
# Here we see the maximum leverage value is greater than 2p/n,
# meaning it is considered large under one guideline discussed in the book.
# Another guideline is that a leverage is considered large if it exceeds 0.5.
# This value does not meet this second guideline.

plot(hat.diag)
```



```
# Plotting the leverage values, we see this 0.3067 leverage value to be
# significantly greater than the next closest leverage value.
# Based on all the above explanations, this point is probably an outlier.
# This point corresponds with observation #61.
```

ANALYSIS: Using the diagonal hat values, based on the guidelines discussed in the book and the visual gap between this value and the next closest, we point to observation #61 as an outlier.

(c) Hidden Extrapolation

```
# Create X matrix
x.matrix <- matrix(c(rep(1,81),df$x1,df$x2,df$x3,df$x4),nrow=81,ncol=5)

# Create X new
x.new <- c(1,10,12,0.05,350000)
```

```
# h new,new using eqn (10.29)
h.new.new <- t(x.matrix) %*% x.matrix # Calculated in a weird order but made syntax easier to handle
h.new.new <- h.new.new^(-1)
h.new.new <- t(x.new) %*% h.new.new * x.new

print(h.new.new)
```

```
##            [,1]      [,2]       [,3]       [,4]     [,5]
## [1,] 0.07785761 0.0896298 0.09309488 0.05034816 0.142693
```

Is extrapolation indicated?

ANALYSIS: As the values in ‘h.new.new’ are not vastly different from the leverage values in part (b), extrapolation IS NOT expected.

(d) DFFITS, DFBETAS, Cook’s Distance

```
# Cases: 61, 8, 3, 53, 6, 62
# Loop to print out required values for above Cases

for (i in c(61,8,3,53,6,62)) {
  print(paste("DFFITS for case ",i,dffits(model)[i]))
  print(paste("DFBETAS for case ",i,dfbetas(model)[i]))
  print(paste("Cook's Distance for case",i,cooks.distance(model)[i]))
}
```

```
## [1] "DFFITS for case  61 0.638720760055322"
## [1] "DFBETAS for case  61 -0.0554152847953117"
## [1] "Cook's Distance for case 61 0.0816621738178867"
## [1] "DFFITS for case  8 0.116413641980649"
## [1] "DFBETAS for case  8 -0.0142102315227387"
## [1] "Cook's Distance for case 8 0.0027446092958763"
## [1] "DFFITS for case  3 -0.28428045467762"
## [1] "DFBETAS for case  3 -0.231785724023509"
## [1] "Cook's Distance for case 3 0.0163061961277757"
## [1] "DFFITS for case  53 0.525226448955067"
## [1] "DFBETAS for case  53 -0.0196280259216763"
## [1] "Cook's Distance for case 53 0.0549818944445512"
## [1] "DFFITS for case  6 -0.873548812821503"
## [1] "DFBETAS for case  6 0.195115462782035"
## [1] "Cook's Distance for case 6 0.137366520897455"
## [1] "DFFITS for case  62 0.690331868565316"
## [1] "DFBETAS for case  62 0.275814688565408"
## [1] "Cook's Distance for case 62 0.0875358896757581"
```

```
# For small/medium data sets (like this one), compare DFFITS to 1
# For small/medium data sets (like this one), compare DFBETAS to 1
# For small/medium data sets (like this one), compare Cook's distance with respect F(p,n-p)

# Cook's distance for Case 6 (largest Cook's Distance), mentioned below
qf(0.137,p,n-p)
```

```
## [1] 0.3770254
```

ANALYSIS: Judging by the DFFITS and DFBETAS results, relative to 1 as is appropriate for small/medium datasets, those metrics DO NOT suggest any of these Cases are influential.

By the Cook's Distance, we compare the Cook's Distance for Case 6, the highest value calculated here of 0.137. On the F(p,n-p) distribution, this sits at the 37.7% percentile, indicating INFLUENCE, but NOT REMEDIAL measure.

(d) Avg Absolute Percent Difference

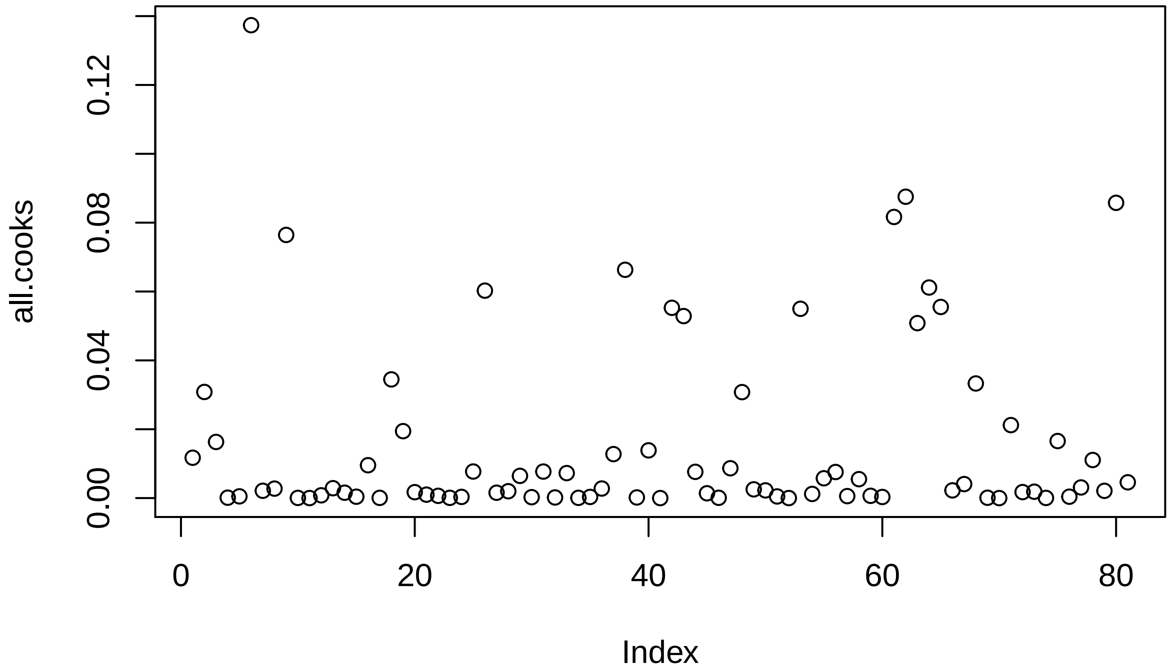
```
# Calculate avg abs percent difference WITH each case:
orig.fit <- model$fitted.values

# Create results dataframe
results.df <- data.frame(case = c(61,8,3,53,6,62), avg_abs_perc_diff = c(rep(NA,6)))
# Write loop to calculate avg abs percent difference without each case

j=1 # Counting index for automating results.df
for (i in c(61,8,3,53,6,62)){
  df.temp <- df
  # Set Y value of selected Case to be empty
  df.temp[i,1] <- NA
  # Fit model withouot selected Case
  model.temp <- lm(y~x1+x2+x3+x4, data=df.temp)
  new.fit <- model.temp$fitted.values
  orig.fit.temp <- orig.fit[-i]  # Making lengths the same
  result <- new.fit - orig.fit.temp
  result <- result*100
  result <- abs(result)
  result <- mean(result)

  results.df[j,2] <- result # Putting results for each case in results.df
  j = j+1

}
print(results.df)
```

```
##   case avg_abs_perc_diff
## 1   61         4.1192912
## 2    8         0.7561326
## 3    3         2.6377355
## 4   53         3.2886427
## 5    6         7.9291375
## 6   62         6.1860189
```

ANALYSIS: We see that the average absolute percent difference for each case is very small, thus indicating none of them are tremendously influential.

(e) Cook's distance, index plot

```
all.cooks <- cooks.distance(model)
plot(all.cooks)
```



ANALYSIS: We see Case 6, as we did before, appear to be inluential. However, as tested numerically with relation to the F-distribution in part (d), this influence is not high enough to dictate remedial action.

Problem 10.13 - Cosmetic Sales

```
# Import data
rm(list=ls())
colnames <- c("y","x1","x2","x3")
df <- read.table(url("http://users.stat.ufl.edu/~rrandles/sta4210/Rclassnotes/data/textdatasets/KutnerDa
n <- nrow(df)
attach(df)
```

(a)

Regression model to be deployed: Y = Beta0 + Beta1*x1 + Beta2*x2 + Beta3*x3 + e

```
# Fitting model
model <- lm(y ~ x1+x2+x3)
p=4
```

(b)

```
summary(model)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -5.4217 -0.9115  0.0703  1.1420  3.5479
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.0233     1.2029   0.851   0.4000
## x1            0.9657     0.7092   1.362   0.1809
## x2            0.6292     0.7783   0.808   0.4237
## x3            0.6760     0.3557   1.900   0.0646 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.825 on 40 degrees of freedom
## Multiple R-squared:  0.7417, Adjusted R-squared:  0.7223
## F-statistic: 38.28 on 3 and 40 DF,  p-value: 7.821e-12
```

Hypothesis:
H0: Bk=0 (k=1,2,3)
Ha: Bk != 0

DECISIONs:
If p-value of the F-statistic < 0.05, reject H0
If p-value of the F-statistic >= 0.05, fail to reject H0

CONCLUSION: As provided in 'summary(model)', the p-value of the F-statistic = 7.821e=12 < 0.05, thus we reject H0 and say THERE IS a regression relationship.

(c)

```
model1 <- lm(y~x1)
model2 <- lm(y~x2)
model3 <- lm(y~x3)

summary(model1)
```

```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.0060 -0.7919  0.1584  1.2961  3.4824
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.1628     0.6712   4.712 2.69e-05 ***
## x1            1.6581     0.1641  10.104 8.23e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.892 on 42 degrees of freedom
## Multiple R-squared:  0.7085, Adjusted R-squared:  0.7016
## F-statistic: 102.1 on 1 and 42 DF,  p-value: 8.231e-13
```

```r
summary(model2)
```

```
##
## Call:
## lm(formula = y ~ x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.4287 -1.2874  0.2027  1.0759  3.6742
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.8315     0.6990   4.051 0.000215 ***
## x2            1.7926     0.1769  10.135 7.51e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.888 on 42 degrees of freedom
## Multiple R-squared:  0.7098, Adjusted R-squared:  0.7029
## F-statistic: 102.7 on 1 and 42 DF,  p-value: 7.507e-13
```

```r
summary(model3)
```

```
##
## Call:
## lm(formula = y ~ x3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.3244 -1.1799 -0.1668  1.5554  8.1139
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.4605     2.0150   1.221  0.22887
## x3            1.9015     0.5449   3.489  0.00115 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.086 on 42 degrees of freedom
## Multiple R-squared:  0.2247, Adjusted R-squared:  0.2063
## F-statistic: 12.17 on 1 and 42 DF,  p-value: 0.001151
```

Conclusion: The p-value of the individual F-tests for each predictor, when fitted individually, all equaled <0.05. We conclude NONE of the Beta's = 0.

(d)

```r
# Correlation Matrix
cor(df)
```

```
##            y        x1        x2        x3
## y  1.0000000 0.8417342 0.8424849 0.4740581
## x1 0.8417342 1.0000000 0.9744313 0.3759509
## x2 0.8424849 0.9744313 1.0000000 0.4099208
## x3 0.4740581 0.3759509 0.4099208 1.0000000
```

(e)

RESPONSE: The research objective was to get a good grasp of Beta1, the change in price when x1 increases holding x2 and x3 constant.

To accomplish this, she needs a regression model containing x1, x3, and x3. According to parts (b) and (c), all three predictors are significant to the regression, thus pointing towards being suitable.

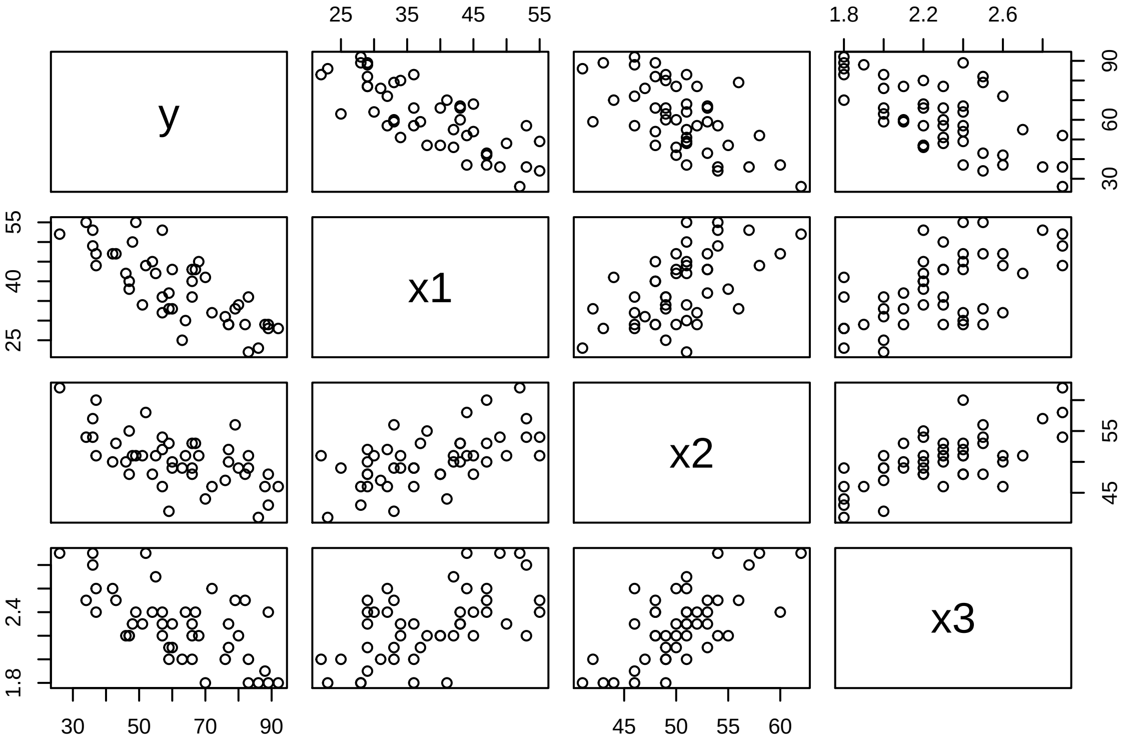
However, as seen in part (d), the correlation between x1 and x3 is rather low at 0.376. The assistant would have to bear this in mind, but overall the data is suitable for her research objective.

Problem 10.17

```r
rm(list=ls())
colnames <- c("y","x1","x2","x3")
df <- read.table(url("http://users.stat.ufl.edu/~rrandles/sta4210/Rclassnotes/data/textdatasets/KutnerDa
n <- nrow(df)
attach(df)
```

(a) Correlation and Scatterplot Matrix

```r
pairs(df)
```



```r
cor(df)
```

```
##             y         x1         x2         x3
## y   1.0000000 -0.7867555 -0.6029417 -0.6445910
## x1 -0.7867555  1.0000000  0.5679505  0.5696775
## x2 -0.6029417  0.5679505  1.0000000  0.6705287
## x3 -0.6445910  0.5696775  0.6705287  1.0000000
```

ANALYSIS: We see that all the predictors are ROUGHLY LINEARLY RELATED (via the scatterplot matrix), but NOT HIGHLY CORRELATED (via the correlation matrix).

(The highest correlation that exists is under <0.7.)

(b) Variance Inflation Factors

```
vif(lm(y~x1+x2+x3))
```

```
##       x1       x2       x3
## 1.632296 2.003235 2.009062
```

```
print(vif)
```

```
## function (mod, ...)
## {
##     UseMethod("vif")
## }
## <bytecode: 0x5fba948>
## <environment: namespace:car>
```

```
mean(vif(lm(y~x1+x2+x3)))
```

```
## [1] 1.881531
```

ANALYSIS: We see that VIF values are all relatively small, with the maximum VIF value being 2.00, much less than the VIF=10 guideline for multicollinearity.

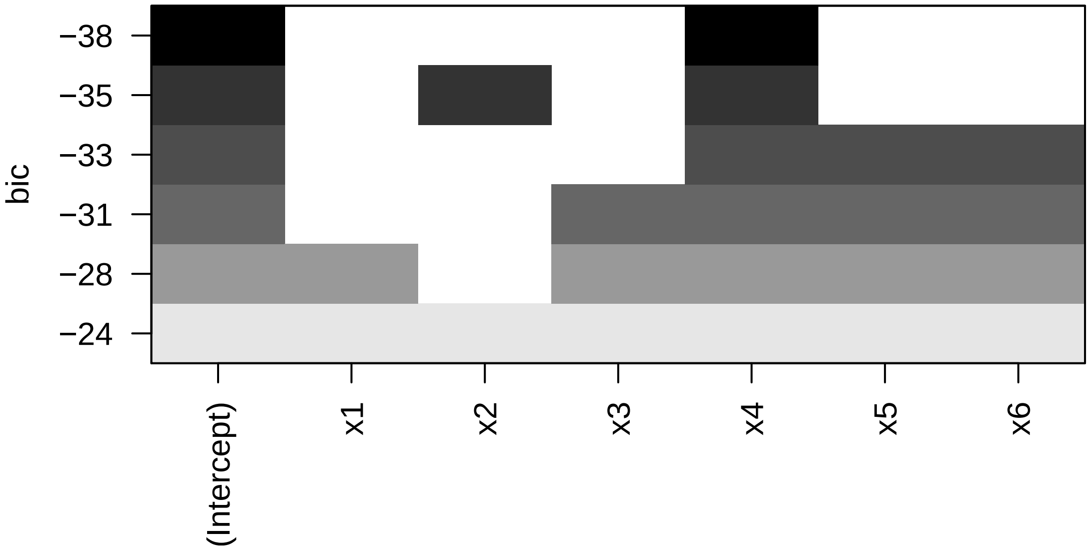
The mean VIF is 1.88, which is >1 (another guideline for multicollinearity), but not by much. Therefore, it may be the case the multicollinearity IS NOT SERIOUS here.

Problem 10.25 - ON SEPARATE HANDWRITTEN SHEET

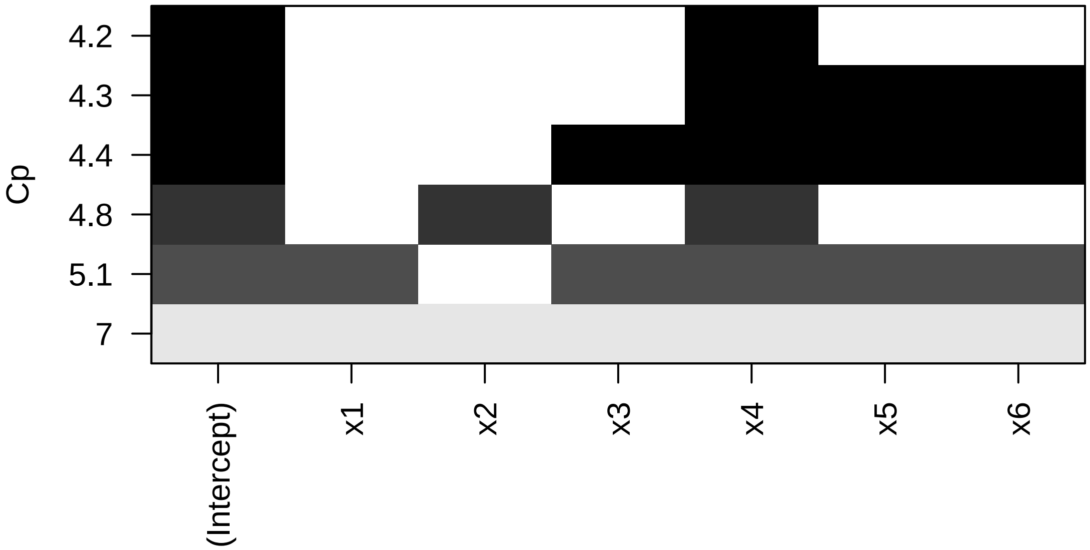
Problem 10.29 - Website Developer

```
# Import Data
rm(list=ls())
colnames <- c("id","y","x1","x2","x3","x4","x5","x6")
df <- read.table(url("http://users.stat.ufl.edu/~rrandles/sta4210/Rclassnotes/data/textdatasets/KutnerDa
n <- nrow(df)
attach(df)
```

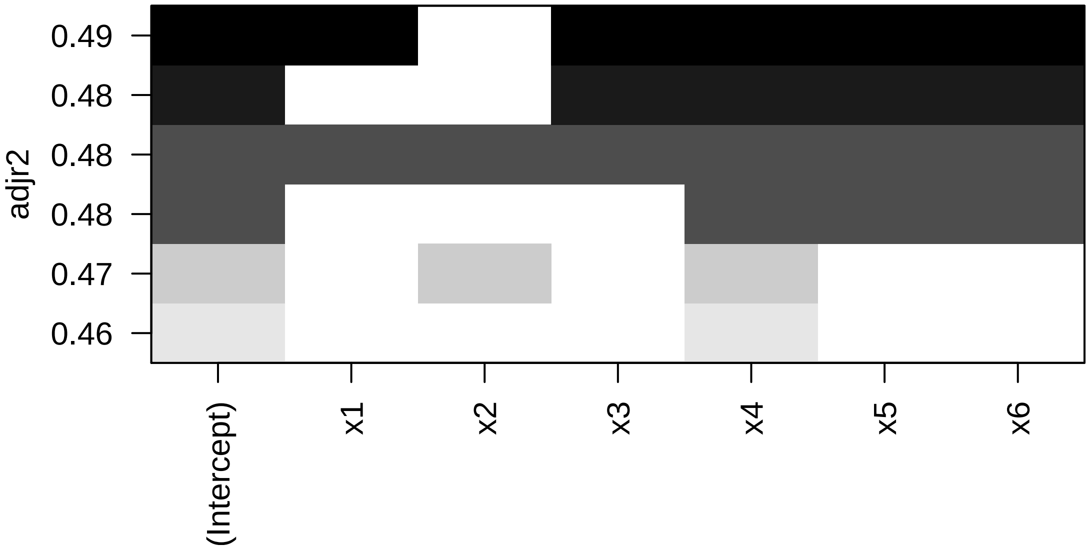
```
# Determine best subset model (with Ahmet's 'leaps' pkg)
allmods <- regsubsets(y~x1+x2+x3+x4+x5+x6, data=df)
plot(allmods)
```



```r
plot(allmods,scale="Cp")
```



```r
plot(allmods,scale="adjr2")
```



```
# Using 'allmods' plot as start point, will calculate AIC, R^2 values for further comparison

model1 <- lm(y~x4)
model2 <- lm(y~x2+x4)
model3 <- lm(y~x4+x5+x6)

summary(model1)$adj.r.squared
```

```
## [1] 0.4644287
```

```
summary(model2)$adj.r.squared
```

```
## [1] 0.4670552
```

```
summary(model3)$adj.r.squared
```

```
## [1] 0.4782849
```

```
AIC(model1)
```

```
## [1] 451.3927
```

```
AIC(model2)
```

```
## [1] 451.9983
```

```
AIC(model3)
```

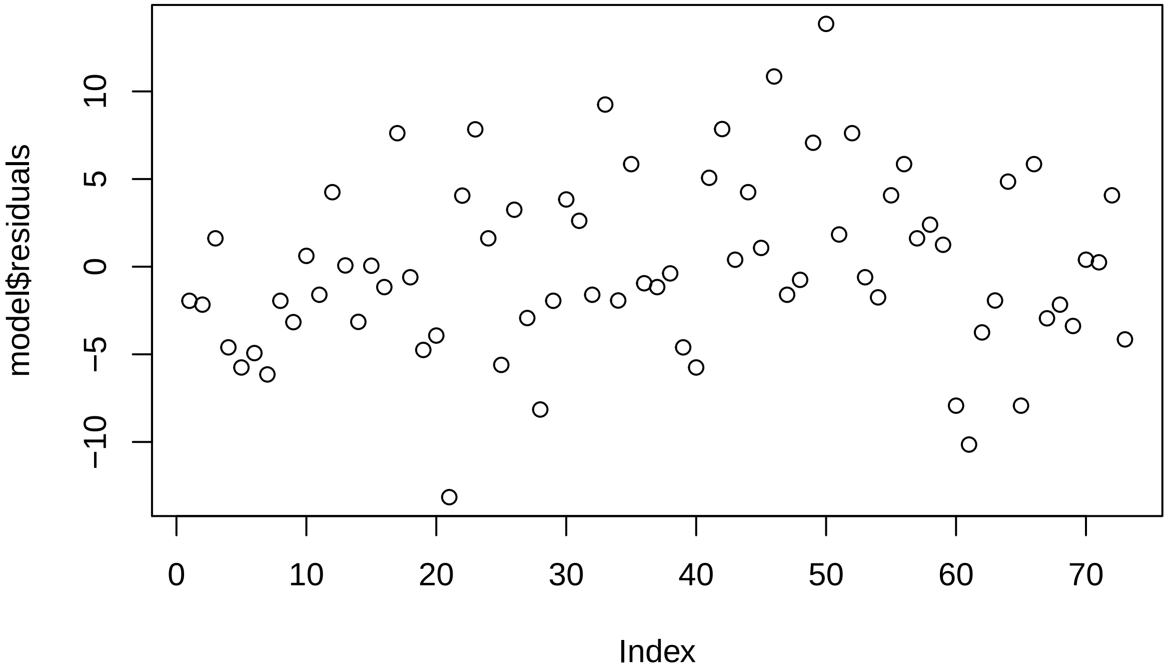
```
## [1] 451.3933
```

DECISION: Going with model that includes X4, X5, and X6:
X4 = Process change (1 or 0)
X5 = Year
X6 = Quarter

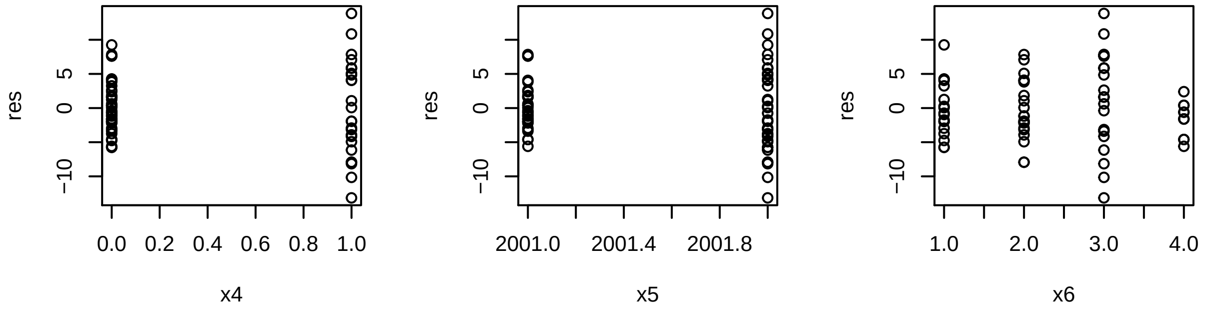
Rationale: This model came in 2nd using BIC as the criteria, and using Cp as the criteria, and first using adj-r.squared. As the only other candidate to show comparable criteria was the model with solely X4 involved, choosing the model with X4, X5, and X6 may capture more variance moving forward (as the plot for r-squared hints).

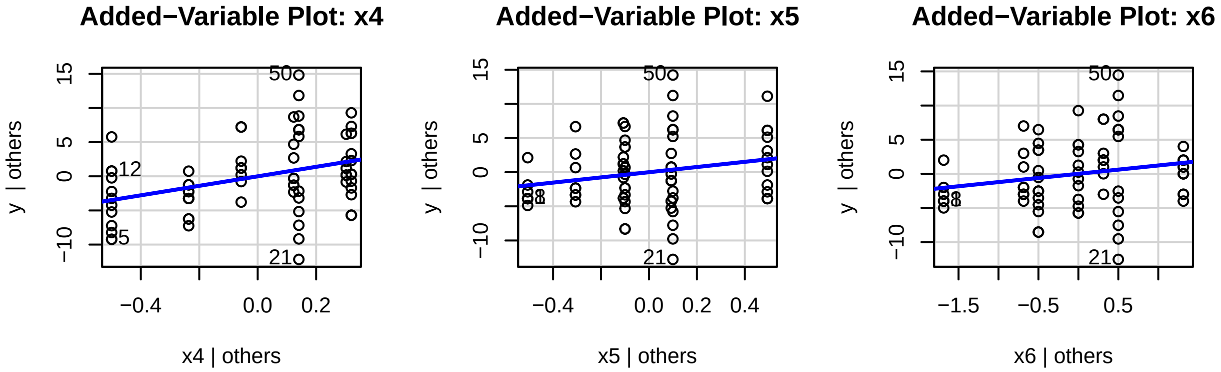
Diagnostics Checks

```
model <- model3
plot(model$residuals)
```



```
p = 4

res <- model$residuals

par(mfrow=c(2,3))
plot(x4,res)
plot(x5,res)
plot(x6,res)
avPlot(model, variable=x4)
avPlot(model, variable=x5)
avPlot(model, variable=x6)
```





Analysis: From the residual plots against each X, we see that the residuals center around 0 for each X. The residuals have a bit higher variability for X4 and X5, perhaps because those are binary predictors.

We see from the added variable plots that the variability around the lines are pretty consistent among the X's. The fits are not tight around the line, but due to this, glaring outliers don't seem to exist either.

Detecting Outliers and Influential Observations

```
rstud <- rstudent(model)
hat..model <- hatvalues(model)
dffits.model <- dffits(model)
cooks.model <- cooks.distance(model)

## Bonferroni outlier test with studentized deleted residuals (alpha = 0.05)

t.crit <- qt(1-0.05/(2*n),n-p-1)
print(t.crit)
```

```
## [1] 3.558685
```

```
print(max(abs(rstud)))
```

```
## [1] 2.914584
```

As the largest absolute studentized deleted residual is less than the relevant Bonferroni t-crit value, we reject that there are outliers present at the 0.05 significance level.

```
print(max(abs(dffits.model)))
```

```
## [1] 0.6212345
```

We see that the DFFITS values don't approach 1, the suggested value for suspecting influence. Therefore, according to DFFITS none of the observations are particularly influential.

```
print(max(abs(cooks.model)))
```

```
## [1] 0.08702985
```

```
qf(max(abs(cooks.model)),p,n-p)
```

```
## [1] 0.2429663
```

Here, we see that on the corresponding F-distribution for the model, the max absolute cooks distance lies around the 24th percentile, indicating there's no particularly influential observation here.

Problem 10.25

Yiran Kang

10.24(a): $t_i = \frac{e_i}{\sqrt{MSE_{(i)}(1-h_{ii})}}$

10.25: $(n-p)MSE = (n-p-1)MSE_{(i)} + \frac{e_i^2}{1-h_{ii}}$

$$(n-p)MSE - \frac{e_i^2}{1-h_{ii}} = (n-p-1)MSE_{(i)}$$

$$MSE_{(i)} = \frac{(n-p)MSE}{n-p-1} - \frac{e_i^2}{(1-h_{ii})(n-p-1)}$$

$$= \frac{(n-p)(1-h_{ii})MSE}{(1-h_{ii})(n-p-1)} - \frac{e_i^2}{\cdot\cdot \quad \cdot\cdot}$$

$$= \frac{(n-p)(1-h_{ii})MSE - e_i^2}{(1-h_{ii})(n-p-1)}$$

$$t_i = \frac{e_i}{\sqrt{\frac{(n-p)(1-h_{ii})MSE - e_i^2}{(1-h_{ii})(n-p-1)}(1-h_{ii})}}$$

$$= \frac{e_i}{\sqrt{\frac{(n-p)(1-h_{ii})^{\cancel{2}}MSE}{\cancel{(1-h_{ii})}(n-p-1)} - \frac{e_i^2\cancel{(1-h_{ii})}}{\cancel{(1-h_{ii})}(n-p-1)}}} = \frac{e_i}{\sqrt{\frac{(n-p)(1-h_{ii})MSE - e_i^2}{(n-p-1)}}}$$

$$= e_i\left(\frac{(n-p)(1-h_{ii})MSE - e_i^2}{n-p-1}\right)^{-1/2}$$

$$= e_i\left(\frac{(1-h_{ii})SSE - e_i^2}{n-p-1}\right)^{-1/2}$$

$*\quad t_i = e_i\left(\frac{n-p-1}{(1-h_{ii})SSE - e_i^2}\right)^{1/2}$ (10.26)

Scanned by CamScanner