PredictiveAnalytics1_HW5

Jieda Li, Kristiyan Dimitrov, Parth Patel, Kristian Nikolov

11/3/2019

Problem 5.4 (Gas mileages of cars: Ridge and lasso regression)

Perform ridge and lasso regression on the gas mileage data considered in exercise 4.11.

```
#Use library glmnet
library(glmnet)
```

```
## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-18
```

```
#First import the data
setwd("/Users/jiedali/Documents/courses/401_predictive_analysis/homeworks/hw5")
mpg_raw <- read.csv("./mpg.csv")
```

```
####################
#Ridge Regression
####################
#Get response vector y and model matrix X
y=mpg_raw$mpg
x=model.matrix(mpg~., mpg_raw)
#Use alpha=0, for ridge regression (alpha=1, for lasso regression)
ridgefit=glmnet(x,y,alpha=0, lambda=seq(0,5,0.0001))
#set the seed for n-fold cross validation
set.seed(12345)
ridgecv=cv.glmnet(x, y, alpha=0,lambda=seq(0,5,0.001),nfold=3)
lambdaridge=ridgecv$lambda.min
#Find beta coefficients corresponding to the optimum lambda found
small.lambda.index <- which(ridgecv$lambda == ridgecv$lambda.min)
small.lambda.betas <- coef(ridgecv$glmnet.fit)[,small.lambda.index]
#Print the optimum lambda
print("The optimum lambda is:")
```

```
## [1] "The optimum lambda is:"
```

```
print(lambdaridge)
```

```
## [1] 0.168
```

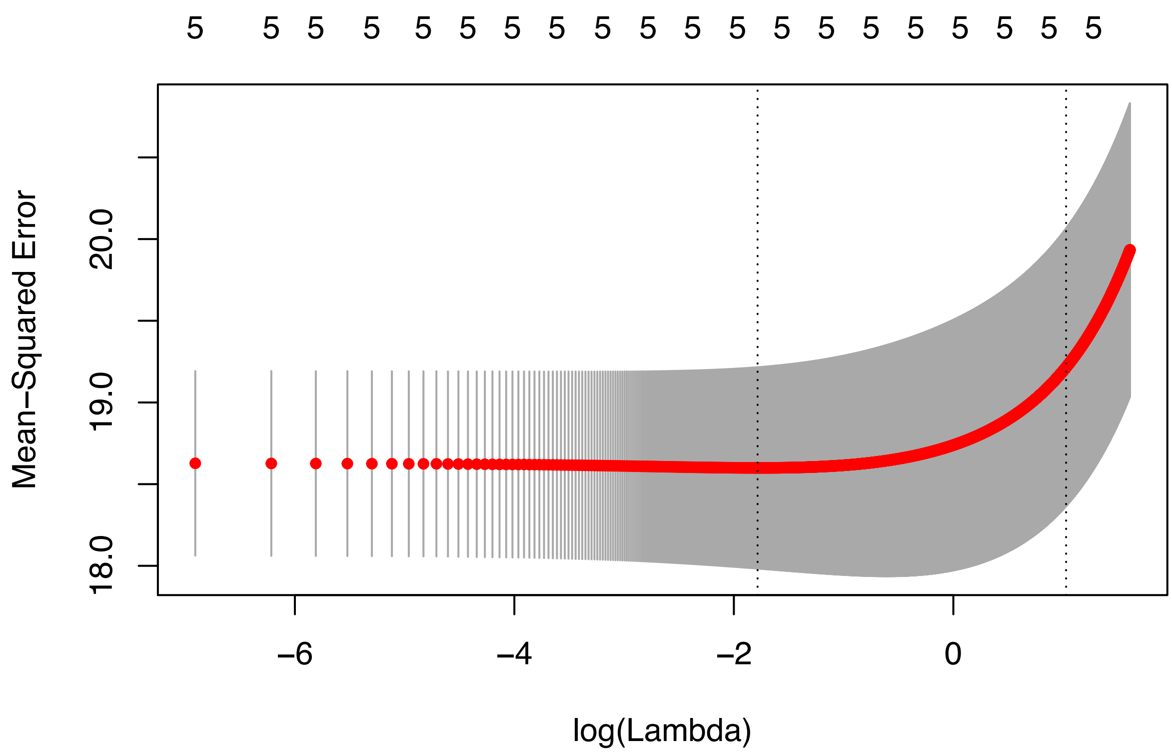
```
print("beta coefficients for the optimum lambda are:")
```

```
## [1] "beta coefficients for the optimum lambda are:"
```

```r
print(small.lambda.betas)
```

```
##  (Intercept)  (Intercept)    cylinders displacement   horsepower
## 45.680827139  0.000000000 -0.406607024 -0.005441543 -0.045804238
##       weight acceleration
## -0.004472311 -0.054709327
```

```r
plot(ridgecv)
```




```r
##############################
#Compute R^2 and adjusted R^2
################################
#Define the function for R^2
r_square <- function(true, predicted)
{
  sse <- sum((predicted - true)^2)
  sst <- sum((true - mean(true))^2)
  rsq <- 1 - sse / sst
}
#Define the function for adjusted R^2
r_square_adj <-function(true, predicted,n,p)
{
  sse <- sum((predicted - true)^2)
  sst <- sum((true - mean(true))^2)
```

```
  rsq <- 1 - (sse/(n-(p+1))) / (sst/(n-1))
}
#Compute R^2 and adjusted R^2 for ridge regression
fit_ridge<-ridgecv$glmnet.fit
y_true = y
y_predicted <- predict(fit_ridge, s = lambdaridge, newx = x)
r_square_ridge <-r_square(y_true,y_predicted)
n<-nrow(mpg_raw)
r_square_adj_ridge <-r_square_adj(y_true,y_predicted,n,5)
print("R^2 from the ridge regression is:")
```

```
## [1] "R^2 from the ridge regression is:"
```

```
r_square_ridge
```

```
## [1] 0.7069993
```

```
print("R^2 adjusted from the ridge regression is:")
```

```
## [1] "R^2 adjusted from the ridge regression is:"
```

```
r_square_adj_ridge
```

```
## [1] 0.703204
```

```
####################
#Lasso Regression
####################
set.seed(12345)
lassofit=glmnet(x, y, alpha=1,lambda=seq(0,5,0.001))
lassocv=cv.glmnet(x,y,alpha=1,lambda=seq(0,5,0.001),nfold=3)
lambdalasso=lassocv$lambda.min
print("the optimum lambda found for lasso regression is:")
```

```
## [1] "the optimum lambda found for lasso regression is:"
```

```
print(lambdalasso)
```

```
## [1] 0.076
```

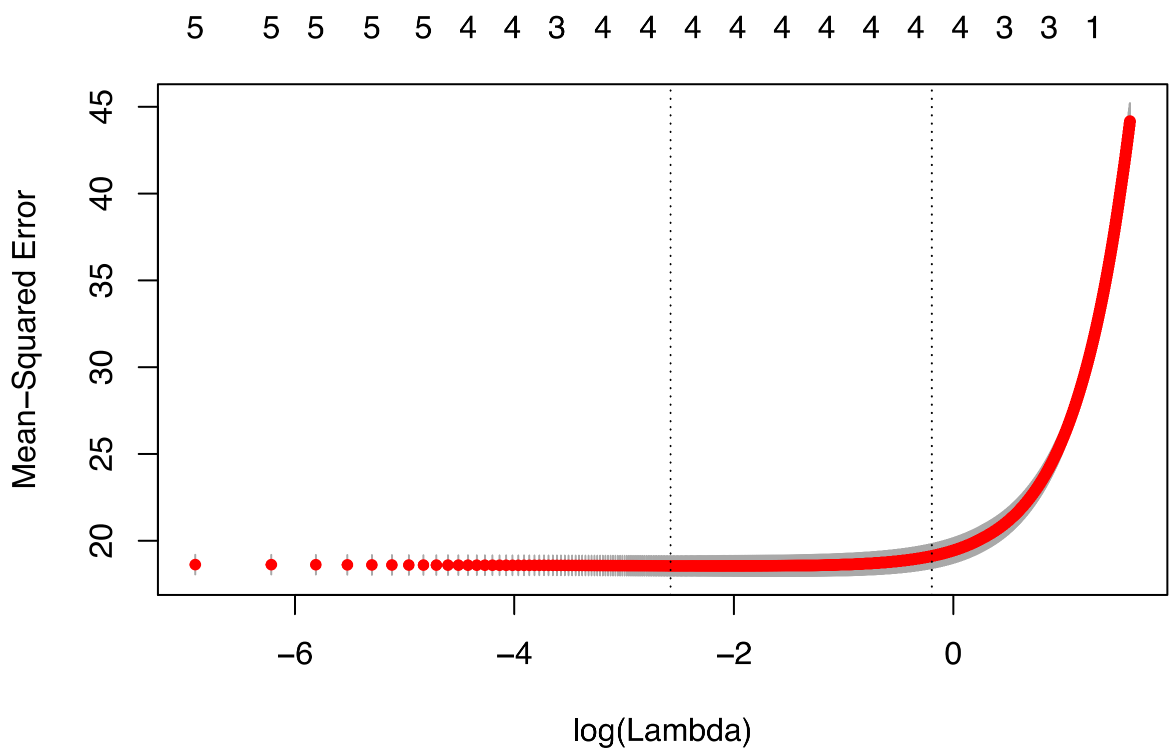
```
#
small.lambda.index <- which(lassocv$lambda == lassocv$lambda.min)
small.lambda.betas <- coef(lassocv$glmnet.fit)[,small.lambda.index]
print("the beta coefficients found for lasso regression is:")
```

```
## [1] "the beta coefficients found for lasso regression is:"
```

```
print(small.lambda.betas)
```

```
##   (Intercept)   (Intercept)     cylinders  displacement    horsepower
## 45.4611552072  0.0000000000 -0.3689097139 -0.0001239583 -0.0417812631
##        weight  acceleration
## -0.0052417069  0.0000000000
```

```
plot(lassocv)
```




```
##############################
#Compute R^2 and adjusted R^2
################################
fit_lasso<-lassocv$glmnet.fit
y_true = y
y_predicted <- predict(fit_lasso, s = lambdalasso, newx = x)
r_square_lasso <-r_square(y_true,y_predicted)
n<-nrow(mpg_raw)
r_square_adj_lasso <-r_square_adj(y_true,y_predicted,n,5)
print("R^2 from the lasso regression is:")
```

```
## [1] "R^2 from the lasso regression is:"
```

```r
r_square_lasso
```

```
## [1] 0.7075472
```

```r
print("R^2 adjusted from the lasso regression is:")
```

```
## [1] "R^2 adjusted from the lasso regression is:"
```

```r
r_square_adj_lasso
```

```
## [1] 0.703759
```

```r
######
#Comments: As can be seen from the beta coefficients, ridge regression drives all
#coefficients towards 0, but didn't drop any varaibles. But lasso regression drops
#certain variable by setting the coefficient to 0, here lasso regression dropped
#variable "accelaration". Yet the adjusted R square is the same for the two
#regression outcome, meaning they have same goodness of fit.
```

Problem 5.5 (Acetylene data: Ridge and lasso regression)

```r
#Use library glmnet
library(glmnet)
#First import the data
setwd("/Users/jiedali/Documents/courses/401_predictive_analysis/homeworks/hw5")
acety <- read.csv("./acetylene.csv")
####################
#Ridge Regression
####################
#Get response vector y and model matrix X
y=acety$y
x=model.matrix(y~., acety)
#Use alpha=0, for ridge regression (alpha=1, for lasso regression)
ridgefit=glmnet(x,y,alpha=0, lambda=seq(0,5,0.0001))
```

```
## Warning: from glmnet Fortran code (error code -49910); Number of nonzero
## coefficients along the path exceeds pmax=10 at 39910th lambda value;
## solutions for larger lambdas returned
```

```r
#set the seed for n-fold cross validation
set.seed(123456789)
ridgecv=cv.glmnet(x, y, alpha=0,lambda=seq(0,5,0.001),nfold=3)
lambdaridge=ridgecv$lambda.min
#Find beta coefficients corresponding to the optimum lambda found
small.lambda.index <- which(ridgecv$lambda == ridgecv$lambda.min)
small.lambda.betas <- coef(ridgecv$glmnet.fit)[,small.lambda.index]
#Print the optimum lambda
print("The optimum lambda is:")
```

```
## [1] "The optimum lambda is:"
```

```r
print(lambdaridge)
```

```
## [1] 0.118
```

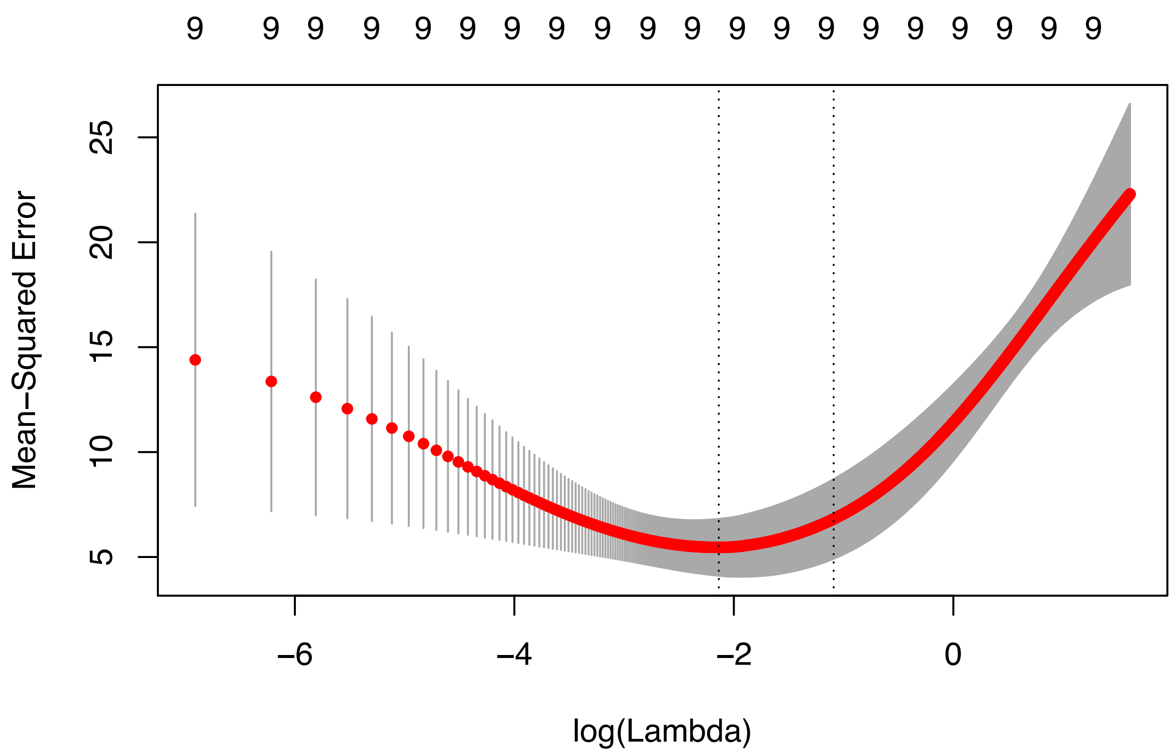
```r
print("beta coefficients for the optimum lambda are:")
```

```
## [1] "beta coefficients for the optimum lambda are:"
```

```r
print(small.lambda.betas)
```

```
##   (Intercept)   (Intercept)            x1            x2            x3
## -8.362989e+01  0.000000e+00  6.462935e-02  2.949670e-01 -9.808018e+01
##          x1x2          x1x3          x2x3          x1.2          x2.2
##  1.244846e-05 -1.160913e-01  1.738576e+01  2.856647e-05 -2.188772e-02
##          x3.2
##  2.854276e+02
```

```r
plot(ridgecv)
```



```r
###############################
#Compute R^2 and adjusted R^2
#################################
#Define the function for R^2
r_square <- function(true, predicted)
{
  sse <- sum((predicted - true)^2)
  sst <- sum((true - mean(true))^2)
  rsq <- 1 - sse / sst
}
#Define the function for adjusted R^2
r_square_adj <-function(true, predicted,n,p)
{
  sse <- sum((predicted - true)^2)
  sst <- sum((true - mean(true))^2)
  rsq <- 1 - (sse/(n-(p+1))) / (sst/(n-1))
}
#Compute R^2 and adjusted R^2 for ridge regression
fit_ridge<-ridgecv$glmnet.fit
y_true = y
y_predicted <- predict(fit_ridge, s = lambdaridge, newx = x)
r_square_ridge <-r_square(y_true,y_predicted)
n<-nrow(mpg_raw)
r_square_adj_ridge <-r_square_adj(y_true,y_predicted,n,5)
print("R^2 from the ridge regression is:")
```

```
## [1] "R^2 from the ridge regression is:"
```

```r
r_square_ridge
```

```
## [1] 0.9881811
```

```r
print("R^2 adjusted from the ridge regression is:")
```

```
## [1] "R^2 adjusted from the ridge regression is:"
```

```r
r_square_adj_ridge
```

```
## [1] 0.988028
```

```r
####################
#Lasso Regression
####################
set.seed(123456789)
lassofit=glmnet(x, y, alpha=1,lambda=seq(0,5,0.001))
lassocv=cv.glmnet(x,y,alpha=1,lambda=seq(0,5,0.001),nfold=3)
lambdalasso=lassocv$lambda.min
print("the optimum lambda found for lasso regression is:")
```

```
## [1] "the optimum lambda found for lasso regression is:"
```

```r
print(lambdalasso)
```

```
## [1] 0.065
```

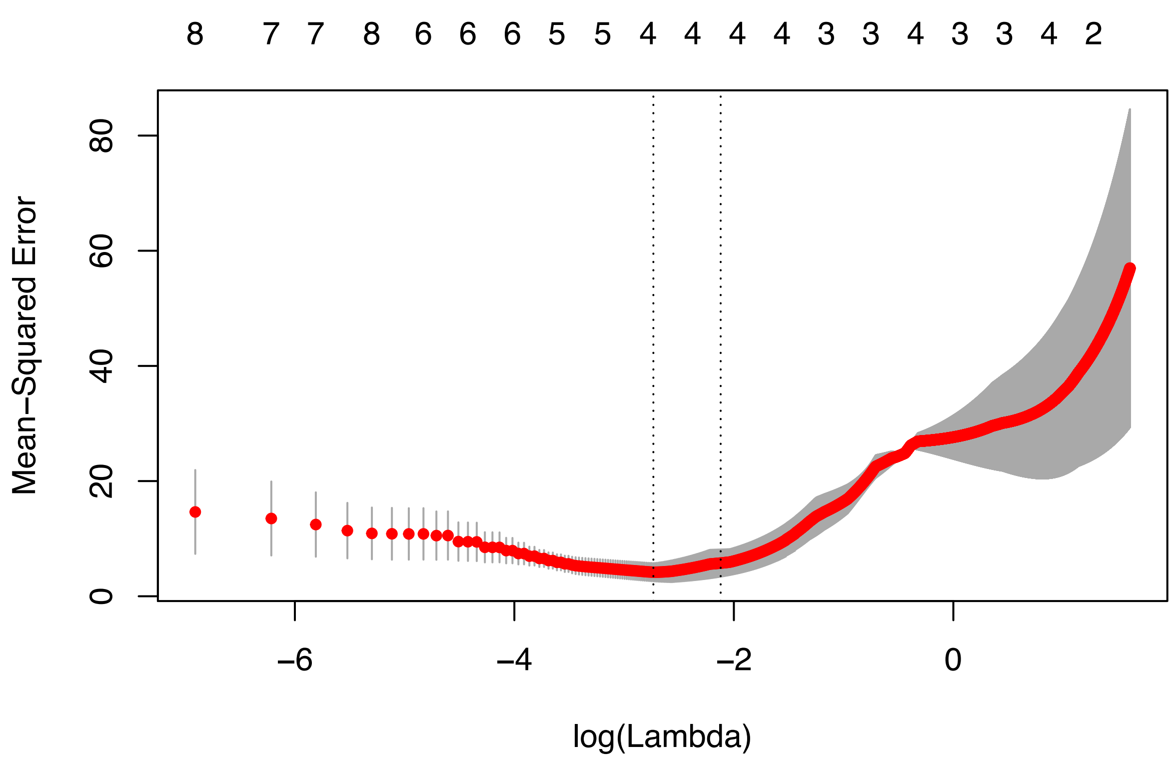
```r
#
small.lambda.index <- which(lassocv$lambda == lassocv$lambda.min)
small.lambda.betas <- coef(lassocv$glmnet.fit)[,small.lambda.index]
print("the beta coefficients found for lasso regression is:")
```

```
## [1] "the beta coefficients found for lasso regression is:"
```

```r
print(small.lambda.betas)
```

```
##   (Intercept)   (Intercept)            x1            x2            x3
## -4.472565e+01  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##          x1x2          x1x3          x2x3          x1.2          x2.2
##  0.000000e+00 -1.831454e-01  1.803677e+01  5.638535e-05 -1.187084e-02
##          x3.2
##  0.000000e+00
```

```r
plot(lassocv)
```



```r
##############################
#Compute R^2 and adjusted R^2
################################
fit_lasso<-lassocv$glmnet.fit
y_true = y
y_predicted <- predict(fit_lasso, s = lambdalasso, newx = x)
r_square_lasso <-r_square(y_true,y_predicted)
n<-nrow(mpg_raw)
r_square_adj_lasso <-r_square_adj(y_true,y_predicted,n,5)
print("R^2 from the lasso regression is:")
```

```
## [1] "R^2 from the lasso regression is:"
```

```r
r_square_lasso
```

```
## [1] 0.9869069
```

```r
print("R^2 adjusted from the lasso regression is:")
```

```
## [1] "R^2 adjusted from the lasso regression is:"
```

```r
r_square_adj_lasso
```

```
## [1] 0.9867373
```

```r
######################
#Commets:Comparing ridge and lasso regression results, we can see that ridge regression
#doesn't drop any predictor variables, but lasso regression drops some variables, yet
#they achieve similar adjusted r square.
######################
```

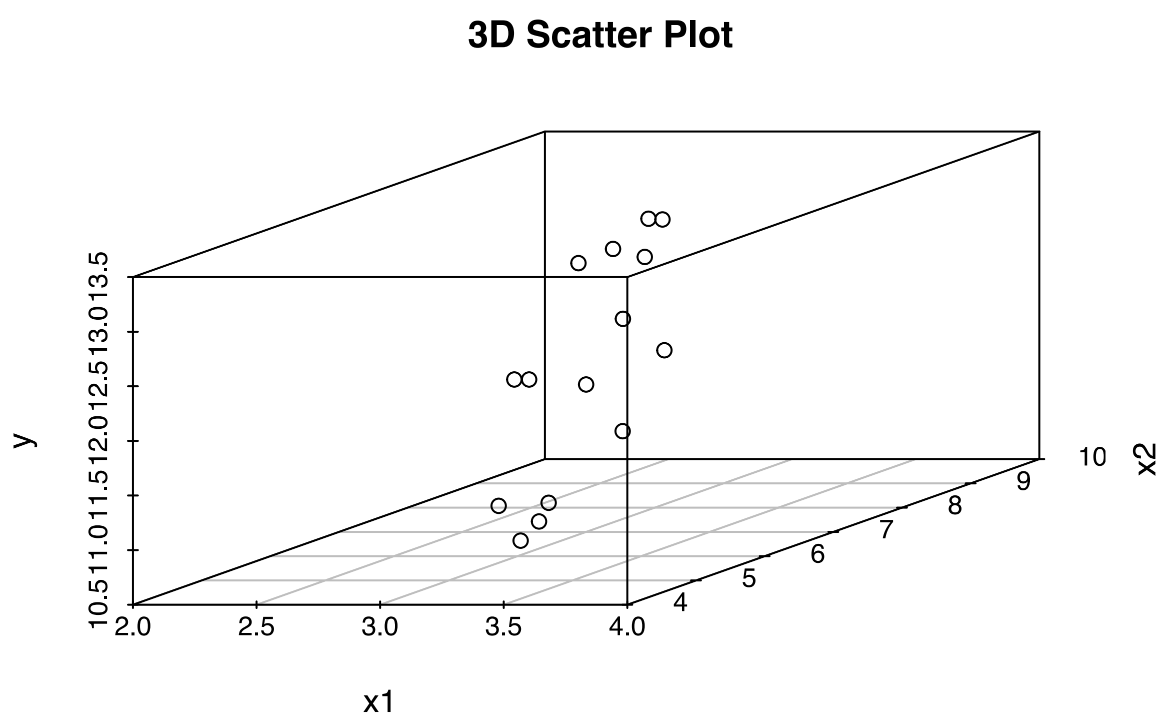
Problem 6.2(Hamilton data)

(a) calculate the correlation matrix between y, x1,and x2. Notice that the correlations between y and x1 and y and x2 are quite small. Make a three-dimensional scatter plot of y versus x1 and x2. What do you notice? Verify the joint relationship of y with x1 and x2 seen in this plot by running a regression.

```r
#First import the data
setwd("/Users/jiedali/Documents/courses/401_predictive_analysis/homeworks/hw5")
hamilton <- read.csv("./Hamilton.csv")
#get the correlation matrix
cor(hamilton)
```

```
##              x1         x2           y
## x1  1.000000000 -0.8997765 0.002497966
## x2 -0.899776481  1.0000000 0.434068758
## y   0.002497966  0.4340688 1.000000000
```

```
#it can be seen from the correlation matrix that the correlation between x1,x2 are very high(-0.899),
# yet the correlation between y and x1, y and x2 are quite small.
#############
#Make a 3D scatter plot of y versus x1 and x2
########
#use the library "scatterplot3d"
library("scatterplot3d")
scatterplot3d(hamilton$x1,hamilton$x2,hamilton$y,main="3D Scatter Plot",
              xlab = "x1",
              ylab = "x2",
              zlab = "y")
```




```
############
#From the 3D plot, we see that y,x1,x2 falls nicely onto the same 2D surface.
#This indicates that the regression plane has a very good fit, high R squre
##########
#Run regression of y on x1 and x2
fit1 <-lm(y~x1+x2, hamilton)
summary(fit1)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2, data = hamilton)
##
## Residuals:
```

```
##       Min        1Q    Median        3Q       Max
## -0.013632 -0.009451 -0.002279  0.008630  0.016325
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.515414   0.061142  -73.85   <2e-16 ***
## x1           3.097008   0.012274  252.31   <2e-16 ***
## x2           1.031859   0.003684  280.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01072 on 12 degrees of freedom
## Multiple R-squared:  0.9998, Adjusted R-squared:  0.9998
## F-statistic: 3.922e+04 on 2 and 12 DF,  p-value: < 2.2e-16
```

Commets: as can be seen from the regression results, the R square is 0.9998, meaning that this is a very good fit for Y on x1 and x2. This agrees with what we have observed from the 3D plot.However it seems to indicate that when x1 and x2 are highly linearly correlated (multicolinear), then y will have a very good fit over both x1 and x2. (This point I don't quite understand)

(b) Why forward stepwise regression would fail but backward stepwise regression would work for this data set?

```
##############################
#First run Forward regression
##############################
fit1 = lm(y ~ 1, hamilton)
summary(fit1)
```

```
##
## Call:
## lm(formula = y ~ 1, data = hamilton)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.170 -0.680  0.000  0.645  1.130
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12.0000     0.2071   57.94   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8022 on 14 degrees of freedom
```

```
step(fit1,direction="forward",scope=~x1+x2)
```

```
## Start:  AIC=-5.65
```

```
## y ~ 1
##
##        Df Sum of Sq    RSS     AIC
## + x2    1   1.69736 7.3112 -6.7796
## <none>              9.0086 -5.6481
## + x1    1   0.00006 9.0085 -3.6482
##
## Step:  AIC=-6.78
## y ~ x2
##
##        Df Sum of Sq    RSS     AIC
## + x1    1    7.3099 0.0014 -133.43
## <none>              7.3112   -6.78
##
## Step:  AIC=-133.43
## y ~ x2 + x1
```

```
##
## Call:
## lm(formula = y ~ x2 + x1, data = hamilton)
##
## Coefficients:
## (Intercept)           x2           x1
##      -4.515        1.032        3.097
```

```r
#############################
#Second run backward regression
#############################
fit2 = lm(y ~ ., hamilton)
summary(fit2)
```

```
##
## Call:
## lm(formula = y ~ ., data = hamilton)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.013632 -0.009451 -0.002279  0.008630  0.016325
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.515414   0.061142  -73.85   <2e-16 ***
## x1           3.097008   0.012274  252.31   <2e-16 ***
## x2           1.031859   0.003684  280.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01072 on 12 degrees of freedom
## Multiple R-squared:  0.9998, Adjusted R-squared:  0.9998
## F-statistic: 3.922e+04 on 2 and 12 DF,  p-value: < 2.2e-16
```

```r
##
step(fit2,direction="backward")
```

```
## Start:  AIC=-133.43
## y ~ x1 + x2
##
##        Df Sum of Sq    RSS      AIC
## <none>              0.0014 -133.429
## - x1    1    7.3099 7.3112   -6.780
## - x2    1    9.0072 9.0085   -3.648
```

```
##
## Call:
## lm(formula = y ~ x1 + x2, data = hamilton)
##
## Coefficients:
## (Intercept)           x1           x2
##      -4.515        3.097        1.032
```

```r
###########################
```

Answer: In theory, when there is multicollinearity, it is recommended to use "backward" stepwise regression, starting from a full model. Becasue when using "foward" stepwise regression, starting from a null model, each individual x1, or x2 won't be significant to be added to the model, but adding both x1 and x2 will be significant. So starting from a null model can end up with a null model. That's why forward stepwise will not work, but backward stepwise will word.

However, seen from the R code, but forward and backward stepwise regression has given the mode of y~x1+x2, my understanding is that when R runs the stepwise regression, it will only look at the AIC score, and will not take into consideration the significance, so a tiny drop in AIC will result the addtion of a varaible into the model.

9.3 (a)

Please disregard the Cp values here, see next page for Cp calculation

Variables in Model	SSE_p	p	Error d.f.	MSE_p	$R^2_{adj,p}$	C_p	AIC_p
None	950	0	19	50	0	1 ~~20~~	79.21 ~~32.081~~ ~~79.21~~
X_1	720	1	18	40	0.2	2	75.67
X_2	630	1	18	35	0.3	2	73
X_3	540	1	18	30	0.4	2	70
X_1, X_2	595	2	17	35	0.3	3	73.86
X_1, X_3	425	2	17	25	0.5	3	67.13
X_2, X_3	510	2	17	30	0.4	3	70.77
X_1, X_2, X_3	400	3	16	25	0.5	4	67.91

Formulas Used.

① $MSE_p = \frac{SSE_p}{n-(p+1)} = \frac{SSE_p}{\text{error d.f}}$

② $R^2_{adj.p} = 1 - \frac{SSE_p / [n-(p+1)]}{SST / [n-1]}$

③ $C_p = \frac{SSE_p}{\hat{\sigma}^2} + 2(p+1) - n = \frac{SSE_p}{MSE_m} + 2(p+1) - n$

④ $AIC_p = n \ln \frac{SSE_p}{n} + 2(p+1)$

⑤ For null model

$R^2_{adj,p} = 1 - \frac{50}{MST} = 0$

$\Rightarrow MST = 50$

* For C_p, $MSE_m = \frac{SSE_m}{n-(m+1)} = \frac{400}{20-4} = \frac{400}{16} = 25$

please Use Cp value here, disregard the Cp values in the previous page!!!

6.3 (a)

Variables in Model	SSE_p	p	Error d.f.	MSE_p	$R^2_{adj,p}$	C_p	AIC_p
None	950	0	19	50	$1-\frac{50}{MST}$	20	79.21
X_1	720	1	18	40	$1-\frac{40}{MST}$	12.8	75.67
X_2	630	1	18	35	$1-\frac{35}{MST}$	9.2	73.00
X_3	540	1	18	30	$1-\frac{30}{MST}$	5.6	69.92
X_1, X_2	595	2	17	35	$1-\frac{35}{MST}$	9.8	73.86
X_1, X_3	425	2	17	25	$1-\frac{25}{MST}$	3.0	67.13
X_2, X_3	510	2	17	30	$1-\frac{30}{MST}$	6.40	70.77
X_1, X_2, X_3	400	3	16	25	$1-\frac{25}{MST}$	4.0	67.91

Formulas used:

① $MSE_p = \frac{SSE_p}{n-(p+1)} = \frac{SSE_p}{\text{error d.f.}}$

② $R^2_{adj,p} = 1 - \frac{SSE_p/[n-(p+1)]}{SST/[n-1]}$

③ $C_p = \frac{SSE_p}{\hat{\sigma}^2} + 2(p+1) - n = \frac{SSE_p}{MSE_m} + 2(p+1) - n$

④ $AIC_p = n \ln \frac{SSE_p}{n} + 2(p+1)$

(b) Which models will be selected as the best using $R^2_{adj,p}$, C_p and AIC_p Criteria? Which model will you choose and why?

Answer: $R^2_{adj,p}$: (X_1, X_3)
C_p : (X_1, X_3)
AIC_p : (X_1, X_3)

I will choose (X_1, X_3) as the model, since all criteria pointed out this model to be the optimum.

(c) Suppose that stepwise regression is to be carried out with $f_{in} = f_{out} = 4.0$. Which variable would be the first to enter the model? What its F_{in} value?

Answer: In the beginning, since there is no variables in the model, the variable that correspond to minimum SSE will enter first, which is X_3.

$$SSE(X_3) = 540$$

$$F_{in} = \frac{SSE(\phi) - SSE(X_3)}{MSE(X_3)} = \frac{950 - 540}{30} = 13.67$$

(d) Which is the second variable to enter the model? F_{in} value? What is the partial correlation coefficient with respect to y controlling for the first variable that entered the model?

Answer: Controlling X_3 already in the model,
$SSE(X_1, X_3) = 425$ $SSE(X_2, X_3) = 510$,
X_1 should be the next that enters the model, because it gives smaller SSE.

$$F_{in} = \frac{SSE(X_3) - SSE(X_1, X_3)}{MSE(X_1, X_3)} = \frac{540 - 425}{25} = 4.6$$

(d) Partial correlation coefficient of y on X_1, controlling X_3, is:

$$r^2_{yx_1|x_3} = 1 - \frac{SSE(X_1, X_3)}{SSE(X_3)} = 1 - \frac{425}{540} = 0.213$$

$$r_{yx_1|x_3} = \pm 0.46$$

(e) Will the first variable that entered the model be removed upon the entry of the second variable? Check by doing the partial F-test.

Answer: the F_{out} statistic for variable X_3, which entered model in step 1:

$$F_{out} = \frac{SSE(X_1) - SSE(X_1, X_3)}{MSE(X_1, X_3)} = \frac{720 - 425}{25} = 11.8 > 4$$

Therefore, X_3 can't be removed from the model.

(f) Will stepwise regression enter the third variable in the model? (i.e. will it choose the full model? Check by doing the partial F-test.

Answer: Now with X_3, X_1 in the model, F_{in} for X_2 is:

$$F_{in} = \frac{SSE(X_1, X_3) - SSE(X_1, X_2, X_3)}{MSE(X_1, X_2, X_3)} = \frac{425 - 400}{25} = 1 < 4.0$$

Therefore, X_2 will NOT be entered into the model.

6.4 (Partial Correlation Coefficient)

(a) Stepwise regression chooses X_1 at step 1 b/c it has the highest correlation w/ y. Which variable will be chose next - X_2 or X_3? Determine by calculating $r_{yx_2|x_1}$ and $r_{yx_3|x_1}$.

$r_{yx_1} = 0.775$
$r_{yx_2} = 0.247$
$r_{yx_3} = 0.742$

$$r_{yx_2|x_1} = \frac{r_{yx_2} - r_{yx_1} r_{x_1x_2}}{\sqrt{(1 - r^2_{yx_1})(1 - r^2_{x_1x_2})}} = \frac{0.247 - 0.775 \cdot (-0.14)}{\sqrt{(1 - 0.775^2)[1 - (-0.14)^2]}}$$

$r_{x_1x_2} = -0.14$
$r_{x_1x_3} = 0.803$

$$= 0.568$$

$$r_{yx_3|x_1} = \frac{r_{yx_3} - r_{yx_1} r_{x_1x_3}}{\sqrt{(1-r_{yx_1}^2)(1-r_{x_1x_3}^2)}} = \frac{0.742 - 0.775 \cdot 0.803}{\sqrt{(1-0.775^2)(1-0.803^2)}} = 0.318$$

Therefore, X_2 will be entered next, since $r_{yx_2|x_1} > r_{yx_3|x_1}$.

(b) Explain why $r_{yx_2|x_1} > r_{yx_3|x_1}$ even though $r_{yx_2} < r_{yx_3}$?

Answer: Given $r_{x_1x_3} = 0.803$, we know X_1, X_3 are highly correlated. Once X_1 is already in the model, the marginal contribution from X_3 to y is very small, because much of the contribution has already been accounted for by X_1.

$r_{x_1x_2} = -0.14$, the correlation between X_1, X_2 are much smaller than $r_{x_1x_3}$. So controlling X_1 in the model, ~~an~~ marginal contribution from X_2 is bigger. So $r_{yx_2|x_1} > r_{yx_3|x_1}$.

(c) Do an F-test on $r_{yx_2|x_1}$ at $\alpha = 0.05$ to check if X_2 should be added to the model that already includes X_1.

Answer: $$F_p = \frac{r_{yx_p|x_1 \cdots x_{p-1}}^2 [n-(p+1)]}{1 - r_{yx_p|x_1 \cdots x_{p-1}}^2}$$

Here $$F_p = \frac{r_{yx_2|x_1}^2 [33-(2+1)]}{1 - r_{yx_2|x_1}^2} = 14.29 > f_{1,30,0.05} = 4.17$$

So X_2 should be added to the model.