CSCI E-106:Assignment 10

Due Date: December 7, 2020 at 7:20 pm EST

Instructions

Students should submit their reports on Canvas. The report needs to clearly state what question is being solved, step-by-step walk-through solutions, and final answers clearly indicated. Please solve by hand where appropriate.

Please submit two files: (1) a R Markdown file (.Rmd extension) and (2) a PDF document, word, or html generated using knitr for the .Rmd file submitted in (1) where appropriate. Please, use RStudio Cloud for your solutions.

Problem 1

Refer to the Cement Composition Data. The variables collected were the amount of tricalcium aluminate (X_1), the amount of tricalcium silicate (X_2), the amount of tetracalcium alumino ferrite (X_3), the amount of dicalcium silicate (X_4), and the heat evolved in calories per gram of cement (Y). (25 points, 5 points each)

a -) Fit regression model for four predictor variables to the data. State the estimated regression function. (5 pt)

```r
Cement <- read.csv("/cloud/project/Cement Composition.csv")
round(cor(Cement),2)
```

```
##        Y    X1    X2    X3    X4
## Y   1.00  0.73  0.82 -0.53 -0.82
## X1  0.73  1.00  0.23 -0.82 -0.25
## X2  0.82  0.23  1.00 -0.14 -0.97
## X3 -0.53 -0.82 -0.14  1.00  0.03
## X4 -0.82 -0.25 -0.97  0.03  1.00
```

```r
f<-lm(Y~.,data=Cement)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	62.41	70.07	0.8906	0.3991
X1	1.551	0.7448	2.083	0.07082
X2	0.5102	0.7238	0.7049	0.5009
X3	0.1019	0.7547	0.135	0.8959
X4	-0.1441	0.7091	-0.2032	0.8441

Table 2: Fitting linear model: Y ~ .

Observations	Residual Std. Error	R^2	Adjusted R^2
13	2.446	0.9824	0.9736

No variables are significant. R Square is 97%. X_1 is highly correlated with X_3.

b-) Fit a ridge regression model and find the best λ. Please see below.

```
library(glmnet)
```

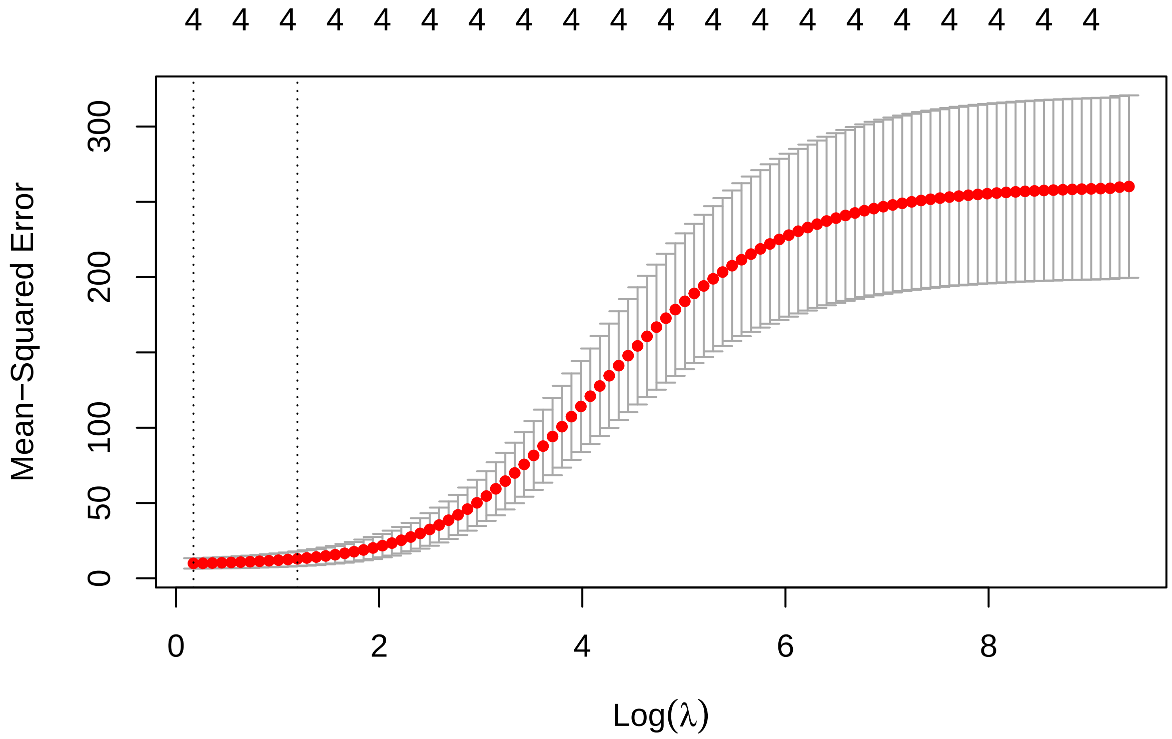
```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.0-2
```

```
x <- model.matrix(Y~., Cement)[,-c(1)]
y <- Cement$Y
RidgeMod <- glmnet(x, y, alpha=0, nlambda=100,lambda.min.ratio=0.0001)
#if you have a hold sample, repeat above to create x and y.
CvRidgeMod <- cv.glmnet(x, y, alpha=0, nlambda=100,lambda.min.ratio=0.0001)
```

```
## Warning: Option grouped=FALSE enforced in cv.glmnet, since < 3 observations per
## fold
```

```
par(mfrow=c(1,1))
plot(CvRidgeMod)
```



```
best.lambda.ridge <- CvRidgeMod$lambda.min
best.lambda.ridge
```

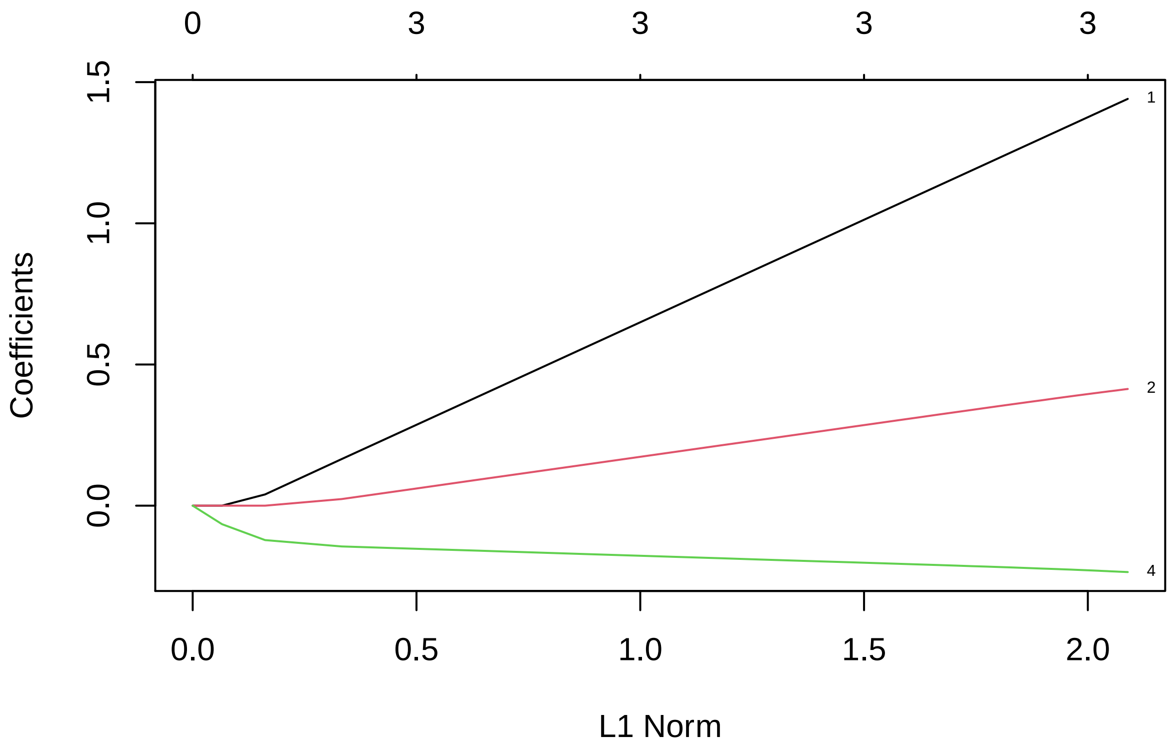
```
## [1] 1.187077
```

```
coef(RidgeMod,s=best.lambda.ridge)
```

```
## 5 x 1 sparse Matrix of class "dgCMatrix"
##                      1
## (Intercept) 86.4729570
## X1           1.1284667
## X2           0.2898568
## X3          -0.2558168
## X4          -0.3472310
```

c-) See Below

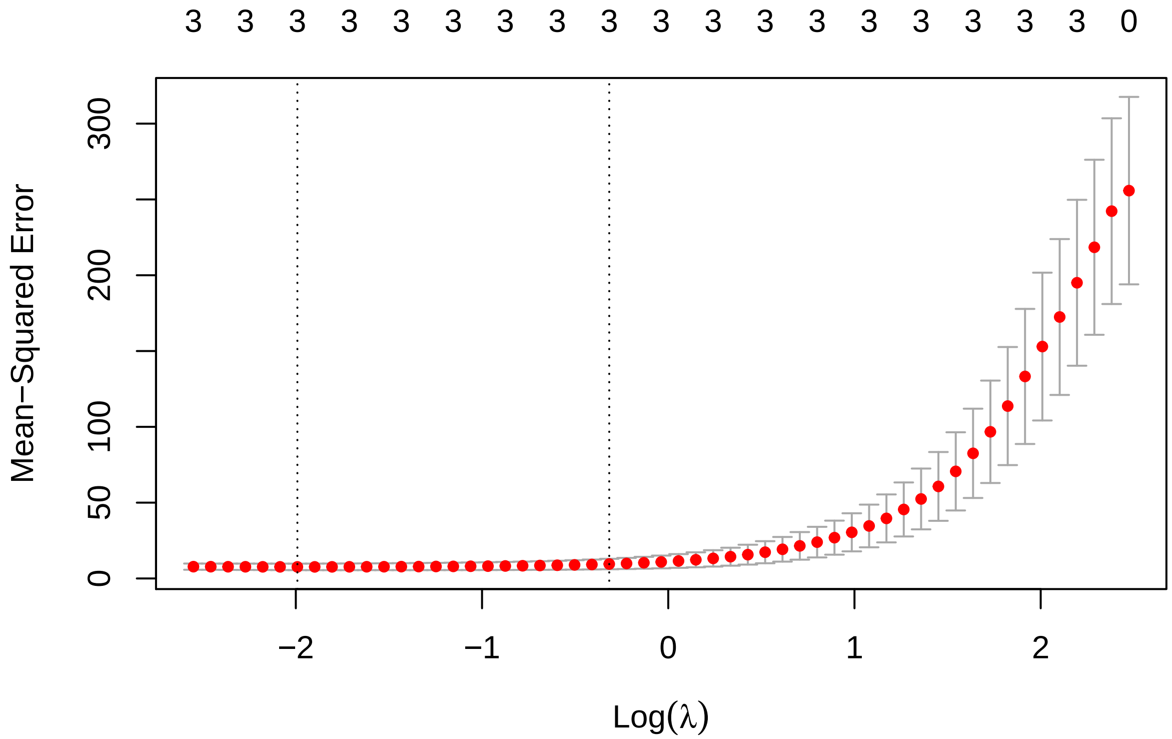
```
LassoMod <- glmnet(x, y, alpha=1, nlambda=100,lambda.min.ratio=0.0001)
plot(LassoMod,xvar="norm",label=TRUE)
```



```
CvLassoMod <- cv.glmnet(x, y, alpha=1, nlambda=100,lambda.min.ratio=0.0001)
```

```
## Warning: Option grouped=FALSE enforced in cv.glmnet, since < 3 observations per
## fold
```

```
plot(CvLassoMod)
```



```
best.lambda.lasso <- CvLassoMod$lambda.min
best.lambda.lasso
```

```
## [1] 0.136485
```

```
coef(CvLassoMod, s = "lambda.min")
```

```
## 5 x 1 sparse Matrix of class "dgCMatrix"
##                      1
## (Intercept) 71.9705570
## X1           1.4323974
## X2           0.4110560
## X3           .        
## X4          -0.2343099
```

d-) See Below

```
EnetMod <- glmnet(x, y, alpha=0.5, nlambda=100,lambda.min.ratio=0.0001)
CvElasticnetMod <- cv.glmnet(x, y,alpha=0.5,nlambda=100,lambda.min.ratio=0.0001)
```

```
## Warning: Option grouped=FALSE enforced in cv.glmnet, since < 3 observations per
## fold
```

```
best.lambda.enet <- CvElasticnetMod$lambda.min
best.lambda.enet
```

```
## [1] 0.2064917
```

```
coef(CvElasticnetMod, s = "lambda.min")
```

```
## 5 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept) 79.60918712
## X1           1.35030248
## X2           0.33261914
## X3          -0.07986444
## X4          -0.31127973
```

e-) Lasso and Elastic Net are very close to each other, almost similar SSE and R^2. I would chooice Lasso since it has a simpler form.

```
y_hat.ridge <- predict(RidgeMod, s = best.lambda.ridge, newx = x)
y_hat.lasso <- predict(LassoMod, s = best.lambda.lasso, newx = x)
y_hat.enet <- predict(CvElasticnetMod , s = best.lambda.enet, newx = x)
sst <- sum((y - mean(y))^2)
sse.ols<-sum(f$residuals^2)
sse.ridge <- sum((y-y_hat.ridge)^2)
sse.lasso <- sum((y-y_hat.lasso)^2)
sse.enet <- sum((y-y_hat.enet)^2)
cbind(sse.ols,sse.ridge,sse.lasso,sse.enet)
```

```
##       sse.ols sse.ridge sse.lasso sse.enet
## [1,] 47.86364  56.16554  48.36563 48.69478
```

```
# R squared
rsq.ols<-1 - sse.ols / sst
rsq.ridge <- 1 - sse.ridge / sst
rsq.lasso <- 1 - sse.lasso / sst
rsq.enet  <- 1 - sse.enet  / sst
cbind(rsq.ols,rsq.ridge,rsq.lasso,rsq.enet)
```

```
##        rsq.ols rsq.ridge rsq.lasso  rsq.enet
## [1,] 0.9823756 0.9793187 0.9821908 0.9820696
```

Problem 2

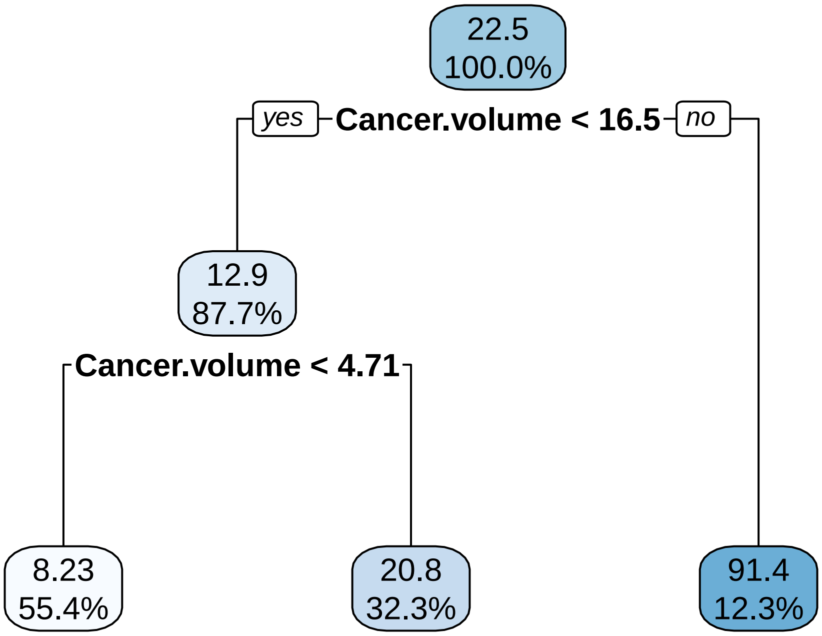
Refer to the Prostate cancer data set in the problem 3 in the Homework 9. Select a random sample of 65 observations to use as the model-building data set. (15 points, 5 each)

a-) Develop a regression tree for predicting PSA. Justify your choice of number of regions (tree size), and interpret your regression tree.

```
PC.Dat <- read.csv("/cloud/project/Prostate Cancer.csv")
set.seed(567)
IND=sample(1:nrow(PC.Dat), size = 65)
PC.Dev=PC.Dat[IND,]
PC.Hold=PC.Dat[-IND,]
library(rpart)
m.rpart <- rpart(PSA.level ~ ., data = PC.Dev)
m.rpart
```

```
## n= 65
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
## 1) root 65 102729.800 22.539650
##   2) Cancer.volume< 16.53225 57   5642.784 12.868260
##     4) Cancer.volume< 4.7117 36   1154.106  8.230694 *
##     5) Cancer.volume>=4.7117 21   2387.133 20.818380 *
##   3) Cancer.volume>=16.53225 8  53768.320 91.448250 *
```

```
library(rpart.plot)
rpart.plot(m.rpart, digits = 3)
```



b-) Assess your model's ability to predict and discuss its usefulness to the oncologists. See below

```
p.rpart <- predict(m.rpart, PC.Hold)
cor(p.rpart,PC.Hold$PSA.level)
```

```
## [1] 0.7116492
```

```
#Measuring performance with the SSE
SSE <- function(actual, predicted) {sum((actual - predicted)^2)}
SSE(PC.Hold$PSA.level,p.rpart)
```

```
## [1] 30330.39
```

```
#Measuring performance with the RSquare
R2 <- function(actual, predicted) {sum((actual - predicted)^2)/((length(actual)-1)*var(actual))}
1-R2(PC.Hold$PSA.level,p.rpart)
```

```
## [1] 0.46472
```

c-) Compare the performance of your regression tree model with that of the best regression model obtained in the problem 3 in the Homework 9. Which model is more easily interpreted and why?

Regression model has two variables. Tree is using one variable. R^2 for Tree is 43% and R^2 for regression model is 17%. Tree outperformed the regression model. In terms of interpretation, both models are equally transparent.

```
r.reg<-lm(PSA.level~Cancer.volume+Capsular.penetration,data=PC.Dev)
summary(r.reg)
```

```
##
## Call:
## lm(formula = PSA.level ~ Cancer.volume + Capsular.penetration,
##     data = PC.Dev)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -67.191  -4.595   1.055   5.135 141.423
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)              0.4517     4.7028   0.096  0.92379
## Cancer.volume            1.7197     0.6123   2.809  0.00664 **
## Capsular.penetration     3.7378     1.2663   2.952  0.00446 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28.65 on 62 degrees of freedom
## Multiple R-squared:  0.5046, Adjusted R-squared:  0.4886
## F-statistic: 31.58 on 2 and 62 DF,  p-value: 3.493e-10
```

```
p.reg <- predict(r.reg, PC.Hold)
cor(p.reg,PC.Hold$PSA.level)
```

```
## [1] 0.5123638
```

```
#Measuring performance with the SSE
SSE(PC.Hold$PSA.level,p.reg)
```

```
## [1] 44670.38
```

```
#Measuring performance with the RSquare
1-R2(PC.Hold$PSA.level,p.reg)
```

```
## [1] 0.2116434
```

Problem 3

Refer to the Prostate cancer data set in the problem 3 in the Homework 9. Select a random sample of 65 observations to use as the model-building data set. (15 points, 5 each)

a-) Develop a neural network model for predicting PSA. Justify your choice of number of hidden nodes and penalty function weight and interpret your model.

See below, R^2 is 97%.

```
library(neuralnet)
normalize <- function(x) {return((x - min(x)) / (max(x) - min(x)))}
scaled.PC.Dat <- as.data.frame(lapply(PC.Dat, normalize))
scaled.PC.Dev<- scaled.PC.Dat[IND,]
scaled.PC.Hold<- scaled.PC.Dat[-IND,]

PC.Dev=PC.Dat[IND,]
PC.Hold=PC.Dat[-IND,]

#we are trying 2 hidden layers with 5 notes
NN = neuralnet(PSA.level~.,hidden=c(5,5), scaled.PC.Dev,linear.output= T )

plot(NN)
#need to take out Y
predict_testNN= compute(NN, scaled.PC.Dev[,-c(1)])
#we need to transform it back to orginal scale
```

```
predict_testNN1 = (predict_testNN$net.result*(max(PC.Dat$PSA.level) -min(PC.Dat$PSA.level))) + min(PC.Da
1-R2(PC.Dev$PSA.level,predict_testNN1)
```

```
## [1] 0.9623268
```

```
plot(PC.Dev$PSA.level, predict_testNN1, col='blue', pch=16, ylab= "Predicted PSA Level", xlab= "Actual
```

b-) Assess your model's ability to predict and discuss its usefulness to the oncologists. See below, Out o

b-) Out of model performance is so bad. We overfitted the model.

```
nn.predict= compute(NN, scaled.PC.Hold[,-c(1)])
#we need to transform it back to orginal scale
nn.predict1 = (nn.predict$net.result*(max(PC.Dat$PSA.level) -min(PC.Dat$PSA.level))) + min(PC.Dat$PSA.le

#R Squares
1-R2(PC.Hold$PSA.level,nn.predict1)
```

```
## [1] 0.04073633
```

```
#SSE
SSE(PC.Hold$PSA.level,nn.predict1)
```

```
## [1] 54354.43
```

c-) Compare the performance of your neural network model with that of the best regression model obtained in the problem 3 in the Homework 9. Which model is more easily interpreted and why?

Regression model performs better than Neuron Network model.

Problem 4

Refer to the Advertising Agency Data. Monthly data on amount of billings (Y, in thousands of constant dollars) and on number of hours of staff time (X, in thousand hours) for the 20 most recent months follow. A simple linear regression model is believed to be appropriate. but positively autocorrelated error terms may be present. (20 points 5 each)

a-) Fit a simple linear regression model by ordinary least squares and obtain the residuals. Conduct a formal test for positive autocorrelation using $\alpha = .01$.

The model is significant, R^2 is almost 100%.

Ho:ρ=0 Ha:ρ>0

Reject null, there is an autocorrelation in the data.

```
AA.Dat <- read.csv("/cloud/project/Advertising Agency.csv")
m.q4<-lm(Y~X,data=AA.Dat)
summary(m.q4)
```

```
##
## Call:
## lm(formula = Y ~ X, data = AA.Dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.55515 -0.23700  0.05229  0.56250  0.80657
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  93.6865     0.8229   113.8   <2e-16 ***
## X            50.8801     0.2634   193.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.631 on 18 degrees of freedom
## Multiple R-squared:  0.9995, Adjusted R-squared:  0.9995
## F-statistic: 3.73e+04 on 1 and 18 DF,  p-value: < 2.2e-16
```

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
dwtest(m.q4)
```

```
##
##  Durbin-Watson test
##
## data:  m.q4
## DW = 0.97374, p-value = 0.002891
## alternative hypothesis: true autocorrelation is greater than 0
```

b-) Use a Cochrane-Orcutt procedure to estimate the model and test if the autocorrelation remains after the first iteration

After the first iteration, the autocorrelation is no longer present.

```
#manual solution
library(Hmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
et<-m.q4$residuals
et1<-Lag(et, shift = 1)

d1<-sum(na.omit(et1*et))
d2<-sum(na.omit(et1)^2)
rho<-d1/d2

Ytnew=AA.Dat$Y - rho*Lag(AA.Dat$Y , shift = 1)
```

```r
Xtnew=AA.Dat$X - rho*Lag(AA.Dat$X , shift = 1)

f1<-lm(Ytnew~Xtnew)
summary(f1)
```

```
##
## Call:
## lm(formula = Ytnew ~ Xtnew)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.95813 -0.29553 -0.02312  0.34451  0.60490
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  63.3840     0.5592   113.4   <2e-16 ***
## Xtnew        50.5470     0.2622   192.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4546 on 17 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.9995, Adjusted R-squared:  0.9995
## F-statistic: 3.715e+04 on 1 and 17 DF,  p-value: < 2.2e-16
```

```r
dwtest(Ytnew~Xtnew)
```

```
##
##  Durbin-Watson test
##
## data:  Ytnew ~ Xtnew
## DW = 1.7612, p-value = 0.2337
## alternative hypothesis: true autocorrelation is greater than 0
```

```r
#use the function
library(orcutt)
coch<- cochrane.orcutt(m.q4)
summary(coch)
```

```
## Call:
## lm(formula = Y ~ X, data = AA.Dat)
##
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept) 95.16377    0.91343  104.18 < 2.2e-16 ***
## X           50.46593    0.28415  177.61 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4514 on 17 degrees of freedom
## Multiple R-squared:  0.9995 ,  Adjusted R-squared:  0.9994
## F-statistic: 31543.9 on 1 and 17 DF,  p-value: < 3.137e-29
##
## Durbin-Watson statistic
## (original):    0.97374 , p-value: 2.891e-03
## (transformed): 1.96762 , p-value: 4.079e-01
```

c-) Restate the estimated regression function obtained in part (b) in terms of the original variables. Also obtain $s(b_0)$ and $s(b_1)$. Compare the estimated regression coefficients obtained.

Please see below for the coefficients and $s(b_0)$ and $s(b_1)$. The new model is $Y = 94.87257 + 75.65823X$

```
#transforming the coefficients back to original form
b0 <- summary(f1)[[4]][1,1]/(1-rho); print(b0)
```

```
## [1] 94.87257
```

```
s.b0 <- summary(f1)[[4]][1,2]/(1-rho)
b1 <- summary(f1)[[4]][2,1]; print(b1)
```

```
## [1] 50.54696
```

```
s.b1 <- summary(f1)[[4]][2,2]
correct.y.hats <- b0 + b1*AA.Dat$X
MSE<-summary(f1)$sigma^2
```

d-)Staff time in month 21 is expected to be 3.625 thousand hours. Predict the amount of billings in constant dollars for month 21, using a 99 percent prediction intervaL Interpret your interval.

```
X.prime<-Xtnew
X.bar.prime <- mean(X.prime[-1])

X.n.plus.1 <- 3.625
X.n <- rev(AA.Dat$X)[1]
X.n.plus.1.prime <- X.n.plus.1 - rho*X.n

# Point forecast:

Y.hat.n.plus.1 <- b0 + b1*X.n.plus.1
Y.n <- rev(AA.Dat$X)[1]
e.n <- Y.n - (b0 + b1*X.n)
Y.hat.FORECAST.n.plus.1 <- Y.hat.n.plus.1 + rho*e.n

print(paste("forecasted response at time n+1 is:", round(Y.hat.FORECAST.n.plus.1,4) ))
```

```
## [1] "forecasted response at time n+1 is: 187.1193"
```

```
# Prediction interval:

alpha <- 0.01
n<-length(AA.Dat$X)
s.pred <- sqrt(MSE*(1 + (1/n) + (X.n.plus.1.prime -X.bar.prime)^2/(sum((X.prime[-1]-X.bar.prime)^2))))
s.pred
```

```
## [1] 0.4737689
```

```
pred.L <- Y.hat.FORECAST.n.plus.1 - qt(1-alpha/2,df=n-3)*s.pred
pred.U <- Y.hat.FORECAST.n.plus.1 + qt(1-alpha/2,df=n-3)*s.pred

print(paste(100*(1-alpha) ,"percent PI for response at time n+1 is:", round(pred.L,4), ",", round(pred.U
```

```
## [1] "99 percent PI for response at time n+1 is: 185.7462 , 188.4924"
```

Problem 5

Refer to the Advertising Agency Data and Problem 4. (25 points, 5 points each)

a-) Use the Hildreth-Lu procedure to obtain a point estimate of the autocorrelation parameter. Do a search at the values $\rho = .1, .2, \ldots\ , 1.0$ and select from these the value of ρ that minimizes SSE. Based on your model, obtain an estimate of the transformed regression function.

ρ=0.4 gives the lowest SSE which is 3.485. the model is $Y = 95.0676 + 50.49249X$.

```
library(HoRM)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
prg1<-function(x,y,rh){
n<-length(rh)
out<-matrix(0,nrow=n,ncol=2)
out[,1]<-rh
for (i in 1:n){
d<-anova(hildreth.lu(y=y,x=x,rho=rh[i]))
out[i,2]<-d$"Sum Sq"[2]
}
out
}
rh<-seq(0.1,1,by=0.1)
hl<-prg1(AA.Dat$X,AA.Dat$Y,rh)
hl[which.min(hl[,2]),]
```

```
## [1] 0.400000 3.468497
```

```
rho=0.4
Ytnew=AA.Dat$Y - rho*Lag(AA.Dat$Y , shift = 1)
Xtnew=AA.Dat$X - rho*Lag(AA.Dat$X , shift = 1)

f2<-lm(Ytnew~Xtnew)
#transforming the coefficients back to original form
b0 <- summary(f2)[[4]][1,1]/(1-rho); print(b0)
```

```
## [1] 95.0676
```

```
b1 <- summary(f2)[[4]][2,1]; print(b1)
```

```
## [1] 50.49249
```

b-) Use the first difference procedure to obtain a point estimate of the autocorrelation parameter.Based on your model, obtain an estimate of the transformed regression function.

the model is $Y = 94.71167 + 50.16414X$.

```
rho=1
Ytnew=AA.Dat$Y - rho*Lag(AA.Dat$Y , shift = 1)
Xtnew=AA.Dat$X - rho*Lag(AA.Dat$X , shift = 1)

f3<-lm(Ytnew~Xtnew -1)
summary(f3)
```

```
##
## Call:
```

```
## lm(formula = Ytnew ~ Xtnew - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8016 -0.1744  0.1508  0.4578  1.0575
##
## Coefficients:
##       Estimate Std. Error t value Pr(>|t|)
## Xtnew   50.164      0.425     118   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5787 on 18 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.9987, Adjusted R-squared:  0.9986
## F-statistic: 1.393e+04 on 1 and 18 DF,  p-value: < 2.2e-16
```

```r
b0 <- mean(AA.Dat$Y)-mean(AA.Dat$X)*summary(f3)[[4]][1,1]; print(b0)
```

```
## [1] 95.88985
```

```r
b1 <- summary(f3)[[4]][1,1]; print(b1)
```

```
## [1] 50.16414
```

c-) Test whether any positive autocorrelation remains in the transformed regression model for both part a and b; use $\alpha = .01$. State the alternatives, decision rule, and conclusion.

No autocorrelation was detected for both models, please see below.

```r
#Hildreth-Lu procedure
rho=0.4
Ytnew=AA.Dat$Y - rho*Lag(AA.Dat$Y , shift = 1)
Xtnew=AA.Dat$X - rho*Lag(AA.Dat$X , shift = 1)

f2<-lm(Ytnew~Xtnew)
dwtest(f2)
```

```
##
##  Durbin-Watson test
##
## data:  f2
## DW = 1.9054, p-value = 0.3509
## alternative hypothesis: true autocorrelation is greater than 0
```

```r
#first difference

rho=1
Ytnew=AA.Dat$Y - rho*Lag(AA.Dat$Y , shift = 1)
Xtnew=AA.Dat$X - rho*Lag(AA.Dat$X , shift = 1)
f4<-lm(Ytnew~Xtnew)
dwtest(f4)
```

```
##
##  Durbin-Watson test
##
## data:  f4
## DW = 2.4246, p-value = 0.8374
## alternative hypothesis: true autocorrelation is greater than 0
```

d-) Which method would you choose? Explain your rationale.

Hildreth-Lu procedure has the smallest MSE, it is a better approach.

```
MSE.HL<-summary(f2)$sigma^2
MSE.FD<-summary(f3)$sigma^2
cbind(MSE.HL,MSE.FD)
```

```
##         MSE.HL    MSE.FD
## [1,] 0.2040292 0.3348636
```

e-) For the selected model in part d. Staff time in month 21 is expected to be 3.625 thousand hours. Predict the amount of billings in constant dollars for month 21, using a 99 percent prediction interval. Interpret your interval.

See below

```
rho=0.4
X.prime<-Xtnew
X.bar.prime <- mean(X.prime[-1])

X.n.plus.1 <- 3.625
X.n <- rev(AA.Dat$X)[1]
X.n.plus.1.prime <- X.n.plus.1 - rho*X.n

# Point forecast:

Y.hat.n.plus.1 <- b0 + b1*X.n.plus.1
Y.n <- rev(AA.Dat$X)[1]
e.n <- Y.n - (b0 + b1*X.n)
Y.hat.FORECAST.n.plus.1 <- Y.hat.n.plus.1 + rho*e.n

print(paste("forecasted response at time n+1 is:", round(Y.hat.FORECAST.n.plus.1,4) ))
```

```
## [1] "forecasted response at time n+1 is: 168.2286"
```

```
# Prediction interval:

alpha <- 0.01
n<-length(AA.Dat$X)
s.pred <- sqrt(MSE*(1 + (1/n) + (X.n.plus.1.prime -X.bar.prime)^2/(sum((X.prime[-1]-X.bar.prime)^2))))
s.pred
```

```
## [1] 0.8577154
```

```
pred.L <- Y.hat.FORECAST.n.plus.1 - qt(1-alpha/2,df=n-3)*s.pred
pred.U <- Y.hat.FORECAST.n.plus.1 + qt(1-alpha/2,df=n-3)*s.pred

print(paste(100*(1-alpha) ,"percent PI for response at time n+1 is:", round(pred.L,4), ",", round(pred.
```

```
## [1] "99 percent PI for response at time n+1 is: 165.7427 , 170.7144"
```