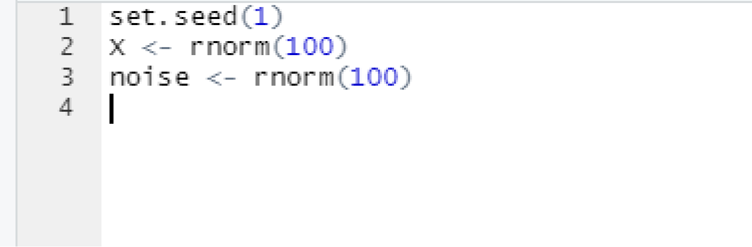
Alexa Summers
Santhoshini Sree Bolisetty
Gireesh Kumar Muppalla
CS 5565—Lab 4

CHAPTER-6

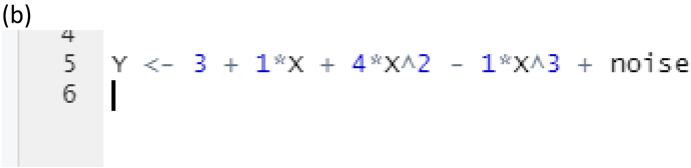
8.

(a)



```
set.seed(1)
X <- rnorm(100)
noise <- rnorm(100)
```

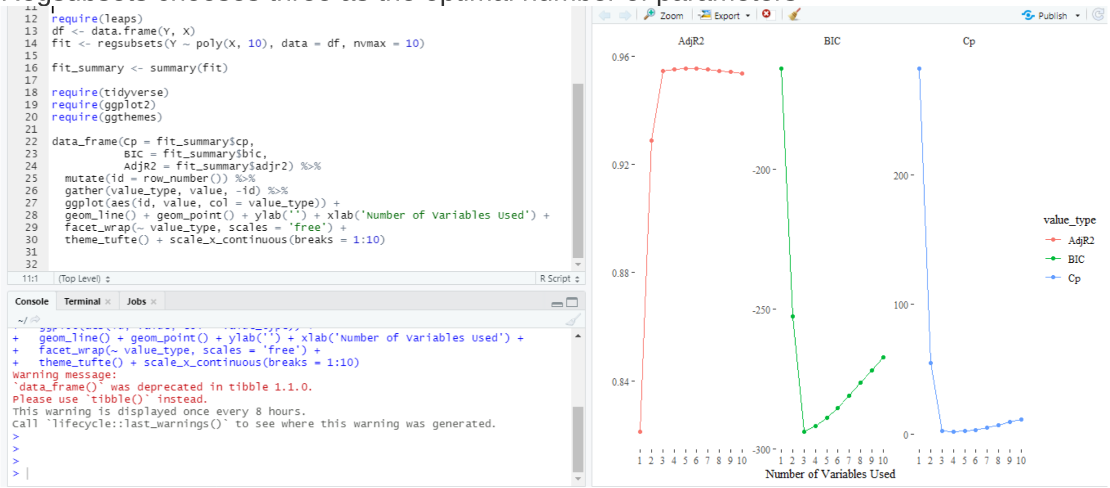
(b)



```
Y <- 3 + 1*X + 4*X^2 - 1*X^3 + noise
```

(c)

Regsubsets chooses three as the optimal number of parameters



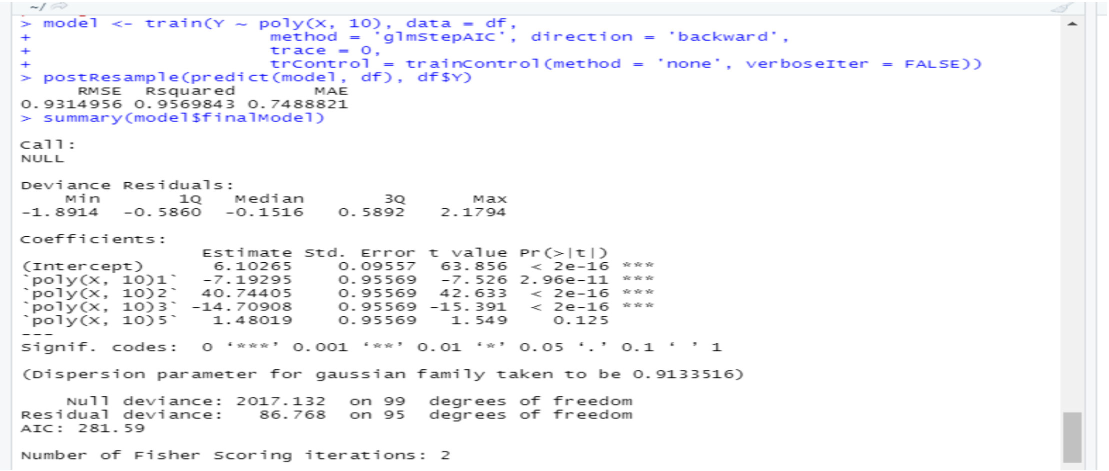
```
require(leaps)
df <- data.frame(Y, X)
fit <- regsubsets(Y ~ poly(X, 10), data = df, nvmax = 10)

fit_summary <- summary(fit)

require(tidyverse)
require(ggplot2)
require(ggthemes)

data_frame(Cp = fit_summary$cp,
           BIC = fit_summary$bic,
           AdjR2 = fit_summary$adjr2) %>%
  mutate(id = row_number()) %>%
  gather(value_type, value, -id) %>%
  ggplot(aes(id, value, col = value_type)) +
  geom_line() + geom_point() + ylab('') + xlab('Number of Variables Used') +
  facet_wrap(~ value_type, scales = 'free') +
  theme_tufte() + scale_x_continuous(breaks = 1:10)
```

```
+   geom_line() + geom_point() + ylab('') + xlab('Number of Variables Used') +
+   facet_wrap(~ value_type, scales = 'free') +
+   theme_tufte() + scale_x_continuous(breaks = 1:10)
Warning message:
`data_frame()` was deprecated in tibble 1.1.0.
Please use `tibble()` instead.
This warning is displayed once every 8 hours.
Call `lifecycle::last_warnings()` to see where this warning was generated.
>
>
>
> 
```

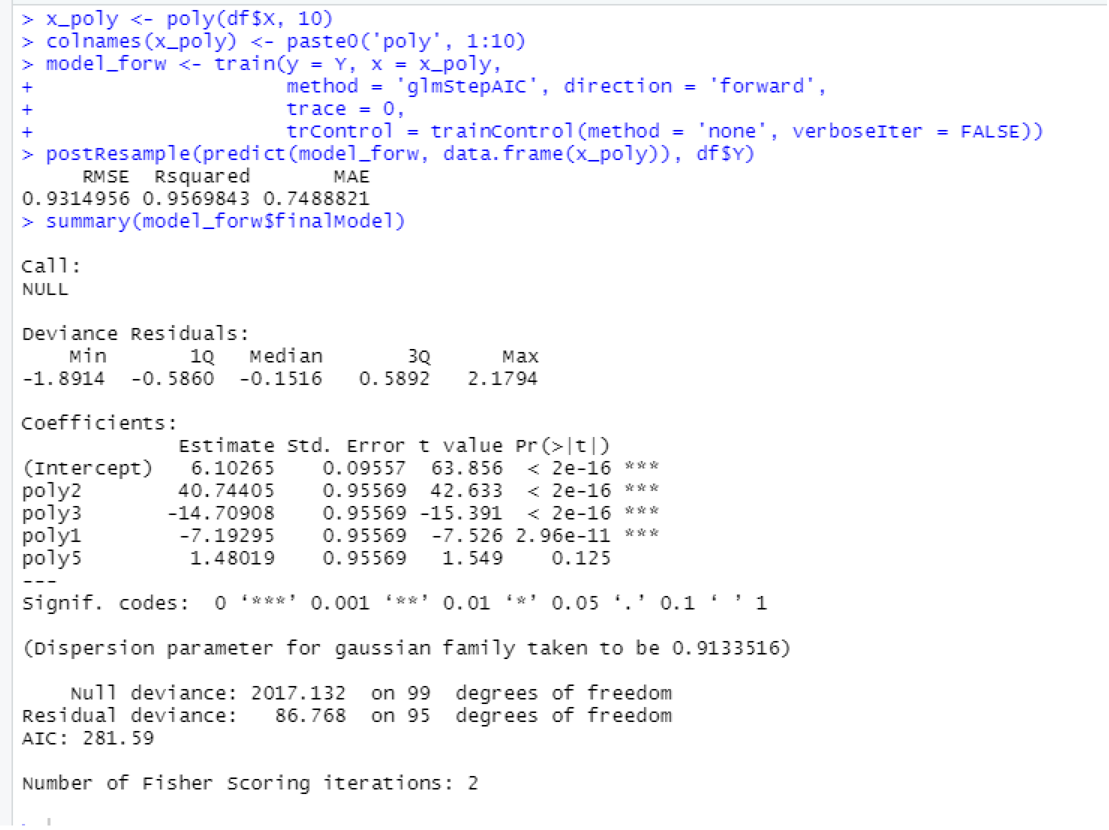
Alexa Summers
Santhoshini Sree Bolisetty
Gireesh Kumar Muppalla
CS 5565—Lab 4

(d)

The backward and forward stepwise model agrees with the best subsets model



```
> model <- train(Y ~ poly(X, 10), data = df,
+                       method = 'glmStepAIC', direction = 'backward',
+                       trace = 0,
+                       trControl = trainControl(method = 'none', verboseIter = FALSE))
> postResample(predict(model, df), df$Y)
     RMSE  Rsquared       MAE 
0.9314956 0.9569843 0.7488821 
> summary(model$finalModel)

Call:
NULL

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-1.8914  -0.5860  -0.1516   0.5892   2.1794  

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)      6.10265    0.09557  63.856  < 2e-16 ***
`poly(X, 10)1`  -7.19295    0.95569  -7.526 2.96e-11 ***
`poly(X, 10)2`  40.74405    0.95569  42.633  < 2e-16 ***
`poly(X, 10)3` -14.70908    0.95569 -15.391  < 2e-16 ***
`poly(X, 10)5`   1.48019    0.95569   1.549    0.125    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 0.9133516)

    Null deviance: 2017.132  on 99  degrees of freedom
Residual deviance:   86.768  on 95  degrees of freedom
AIC: 281.59

Number of Fisher Scoring iterations: 2
```

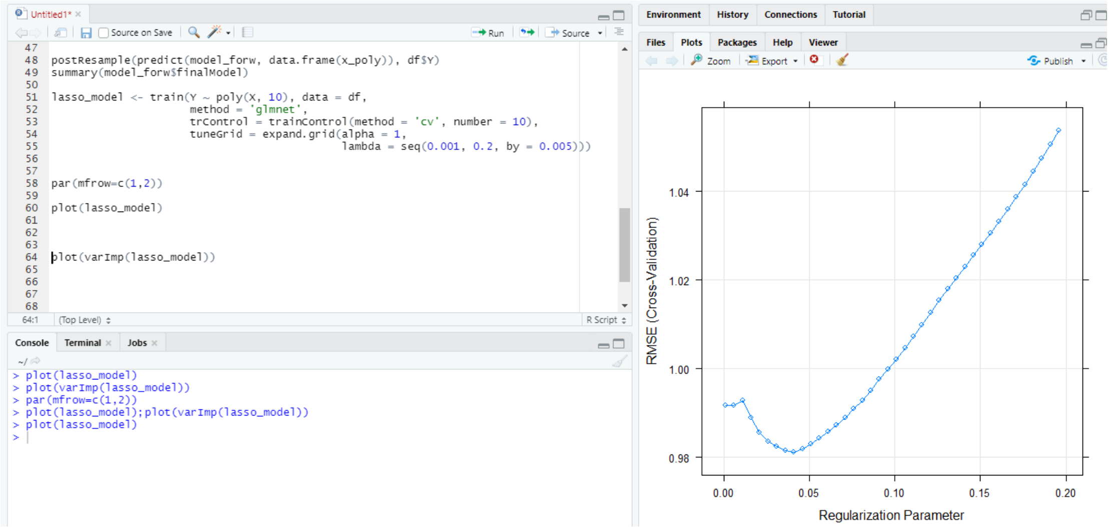


```
> x_poly <- poly(df$X, 10)
> colnames(x_poly) <- paste0('poly', 1:10)
> model_forw <- train(y = Y, x = x_poly,
+                       method = 'glmStepAIC', direction = 'forward',
+                       trace = 0,
+                       trControl = trainControl(method = 'none', verboseIter = FALSE))
> postResample(predict(model_forw, data.frame(x_poly)), df$Y)
     RMSE  Rsquared       MAE 
0.9314956 0.9569843 0.7488821 
> summary(model_forw$finalModel)

Call:
NULL

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-1.8914  -0.5860  -0.1516   0.5892   2.1794  

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)   6.10265    0.09557  63.856  < 2e-16 ***
poly2        40.74405    0.95569  42.633  < 2e-16 ***
poly3       -14.70908    0.95569 -15.391  < 2e-16 ***
poly1        -7.19295    0.95569  -7.526 2.96e-11 ***
poly5         1.48019    0.95569   1.549    0.125    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 0.9133516)

    Null deviance: 2017.132  on 99  degrees of freedom
Residual deviance:   86.768  on 95  degrees of freedom
AIC: 281.59

Number of Fisher Scoring iterations: 2
```

Alexa Summers
Santhoshini Sree Bolisetty
Gireesh Kumar Muppalla
CS 5565—Lab 4

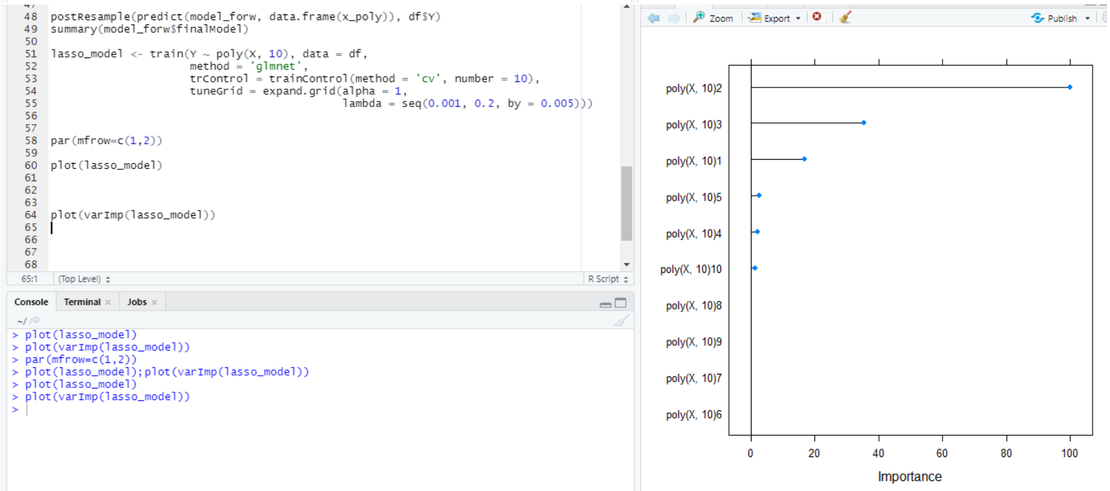
(e)

The Lasso model overestimates the number of predictors needed. This might be expected since we used only RSS to select the optimal model but not the Bayesian Inference Criterion and Adjusted R2R2 as regsubsets does or the Aikake Information Criterion as the stepwise selection does.



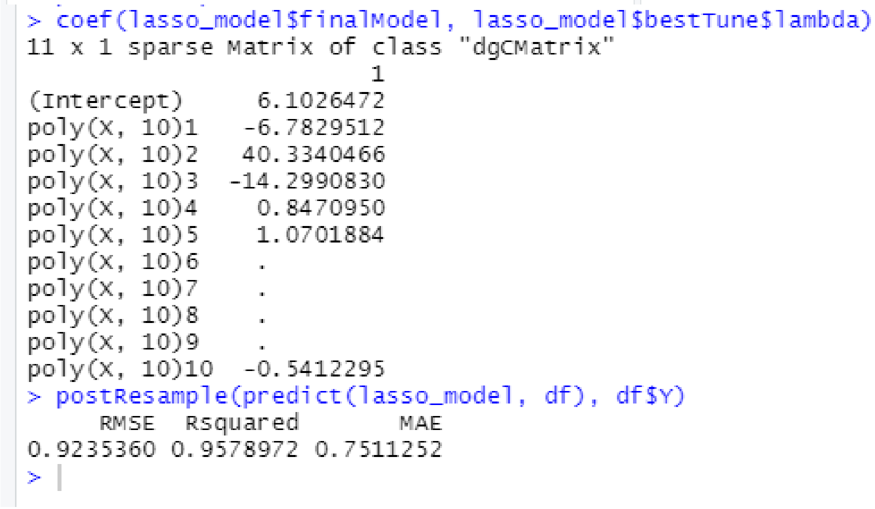
```
postResample(predict(model_forw, data.frame(x_poly)), df$Y)
summary(model_forw$finalModel)

lasso_model <- train(Y ~ poly(X, 10), data = df,
                     method = 'glmnet',
                     trControl = trainControl(method = 'cv', number = 10),
                     tuneGrid = expand.grid(alpha = 1,
                                            lambda = seq(0.001, 0.2, by = 0.005)))


par(mfrow=c(1,2))

plot(lasso_model)



plot(varImp(lasso_model))
```

```
> plot(lasso_model)
> plot(varImp(lasso_model))
> par(mfrow=c(1,2))
> plot(lasso_model);plot(varImp(lasso_model))
> plot(lasso_model)
> 
```



```
postResample(predict(model_forw, data.frame(x_poly)), df$Y)
summary(model_forw$finalModel)

lasso_model <- train(Y ~ poly(X, 10), data = df,
                     method = 'glmnet',
                     trControl = trainControl(method = 'cv', number = 10),
                     tuneGrid = expand.grid(alpha = 1,
                                            lambda = seq(0.001, 0.2, by = 0.005)))


par(mfrow=c(1,2))

plot(lasso_model)



plot(varImp(lasso_model))
```

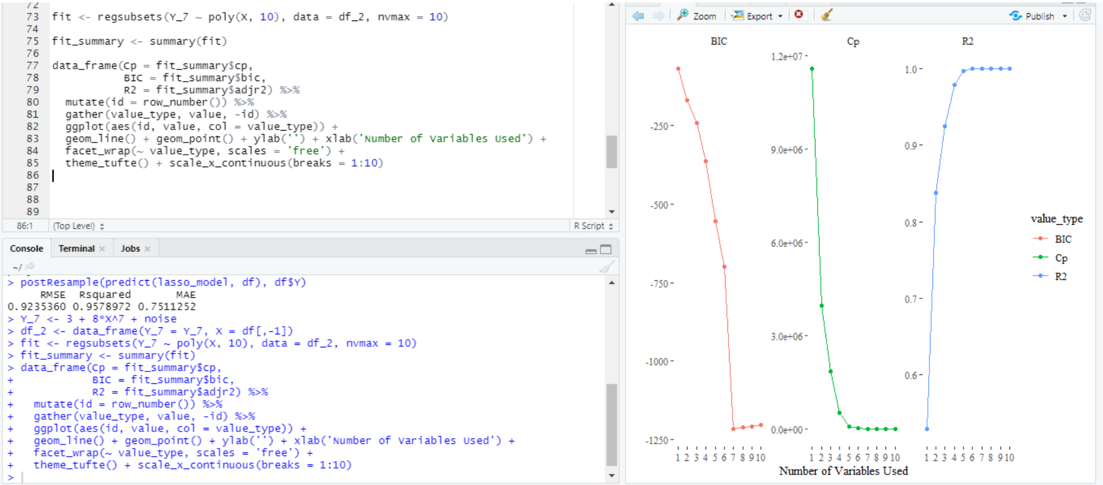
```
> plot(lasso_model)
> plot(varImp(lasso_model))
> par(mfrow=c(1,2))
> plot(lasso_model);plot(varImp(lasso_model))
> plot(lasso_model)
> plot(varImp(lasso_model))
> 
```



```
> coef(lasso_model$finalModel, lasso_model$bestTune$lambda)
11 x 1 sparse Matrix of class "dgCMatrix"
                      1
(Intercept)    6.1026472
poly(X, 10)1  -6.7829512
poly(X, 10)2  40.3340466
poly(X, 10)3 -14.2990830
poly(X, 10)4   0.8470950
poly(X, 10)5   1.0701884
poly(X, 10)6   .        
poly(X, 10)7   .        
poly(X, 10)8   .        
poly(X, 10)9   .        
poly(X, 10)10 -0.5412295
> postResample(predict(lasso_model, df), df$Y)
     RMSE  Rsquared       MAE 
0.9235360 0.9578972 0.7511252 
> 
```

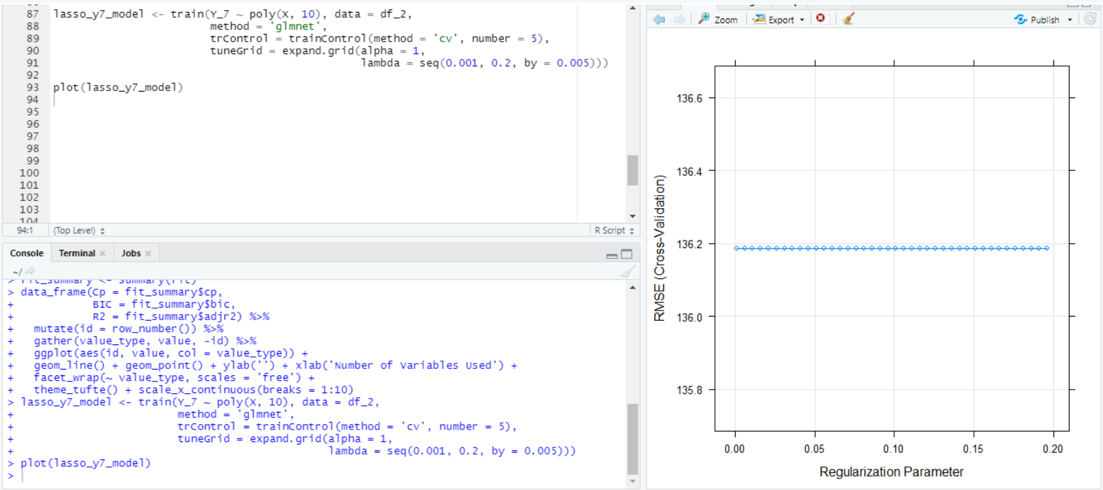
Alexa Summers
Santhoshini Sree Bolisetty
Gireesh Kumar Muppalla
CS 5565—Lab 4

(f)



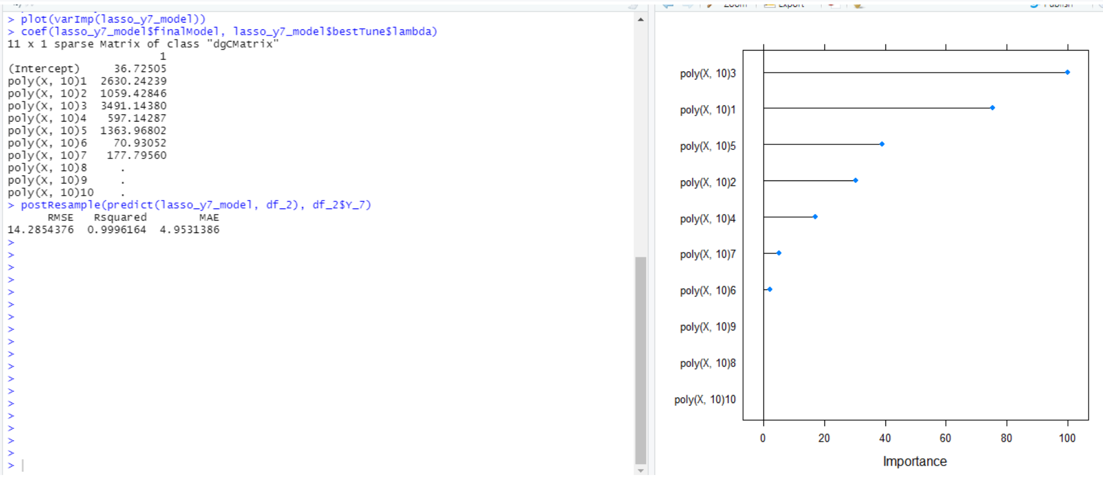
```
fit <- regsubsets(Y_7 ~ poly(X, 10), data = df_2, nvmax = 10)

fit_summary <- summary(fit)

data_frame(Cp = fit_summary$cp,
           BIC = fit_summary$bic,
           R2 = fit_summary$adjr2) %>%
  mutate(id = row_number()) %>%
  gather(value_type, value, -id) %>%
  ggplot(aes(id, value, col = value_type)) +
  geom_line() + geom_point() + ylab('') + xlab('Number of Variables Used') +
  facet_wrap(~ value_type, scales = 'free') +
  theme_tufte() + scale_x_continuous(breaks = 1:10)
```

```
> postResample(predict(lasso_model, df), df$Y)
     RMSE  Rsquared       MAE 
0.9235360 0.9578972 0.7511252 
> Y_7 <- 3 + 8*X^7 + noise
> df_2 <- data_frame(Y_7 = Y_7, X = df[,-1])
> fit <- regsubsets(Y_7 ~ poly(X, 10), data = df_2, nvmax = 10)
> fit_summary <- summary(fit)
> data_frame(Cp = fit_summary$cp,
+            BIC = fit_summary$bic,
+            R2 = fit_summary$adjr2) %>%
+   mutate(id = row_number()) %>%
+   gather(value_type, value, -id) %>%
+   ggplot(aes(id, value, col = value_type)) +
+   geom_line() + geom_point() + ylab('') + xlab('Number of Variables Used') +
+   facet_wrap(~ value_type, scales = 'free') +
+   theme_tufte() + scale_x_continuous(breaks = 1:10)
> 
```



```
lasso_y7_model <- train(Y_7 ~ poly(X, 10), data = df_2,
                        method = 'glmnet',
                        trControl = trainControl(method = 'cv', number = 5),
                        tuneGrid = expand.grid(alpha = 1,
                                               lambda = seq(0.001, 0.2, by = 0.005)))

plot(lasso_y7_model)
```

```
> data_frame(Cp = fit_summary$cp,
+            BIC = fit_summary$bic,
+            R2 = fit_summary$adjr2) %>%
+   mutate(id = row_number()) %>%
+   gather(value_type, value, -id) %>%
+   ggplot(aes(id, value, col = value_type)) +
+   geom_line() + geom_point() + ylab('') + xlab('Number of Variables Used') +
+   facet_wrap(~ value_type, scales = 'free') +
+   theme_tufte() + scale_x_continuous(breaks = 1:10)
> lasso_y7_model <- train(Y_7 ~ poly(X, 10), data = df_2,
+                         method = 'glmnet',
+                         trControl = trainControl(method = 'cv', number = 5),
+                         tuneGrid = expand.grid(alpha = 1,
+                                                lambda = seq(0.001, 0.2, by = 0.005)))
> plot(lasso_y7_model)
> 
```

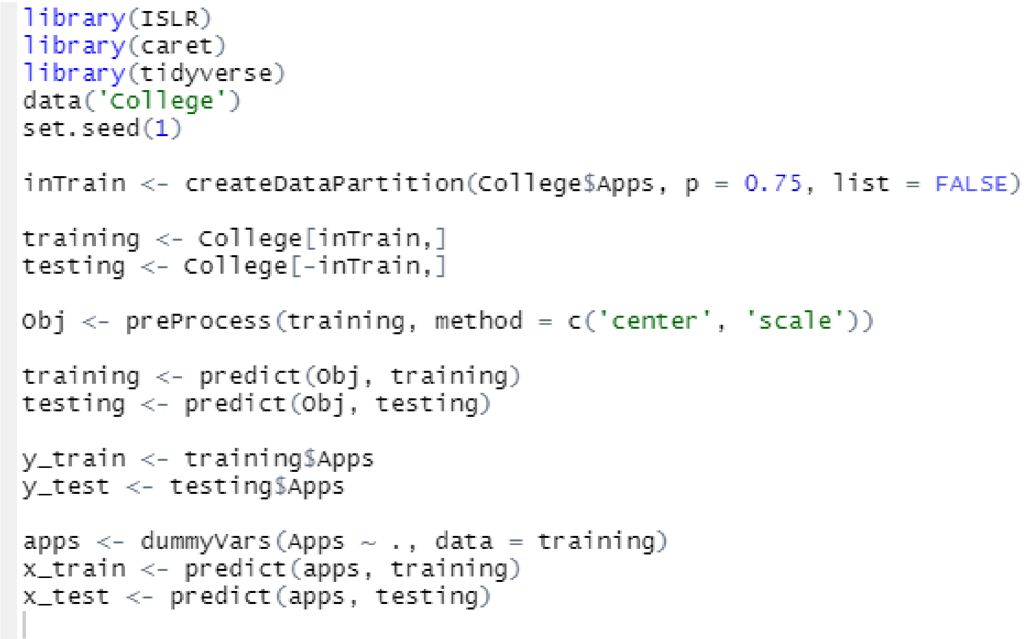


```
> plot(varImp(lasso_y7_model))
> coef(lasso_y7_model$finalModel, lasso_y7_model$bestTune$lambda)
11 x 1 sparse Matrix of class "dgCMatrix"
                        1
(Intercept)      36.72505
poly(X, 10)1   2630.24239
poly(X, 10)2   1059.42846
poly(X, 10)3   3491.14380
poly(X, 10)4    597.14287
poly(X, 10)5   1363.96802
poly(X, 10)6     70.93052
poly(X, 10)7    177.79560
poly(X, 10)8       .    
poly(X, 10)9       .    
poly(X, 10)10      .    
> postResample(predict(lasso_y7_model, df_2), df_2$Y_7)
      RMSE   Rsquared        MAE 
14.2854376  0.9996164  4.9531386 
> 
> 
> 
> 
> 
> 
> 
> 
> 
> 
> 
> 
> 
> 
> 
> 
> 
> 
> 
```

Alexa Summers
Santhoshini Sree Bolisetty
Gireesh Kumar Muppalla
CS 5565—Lab 4

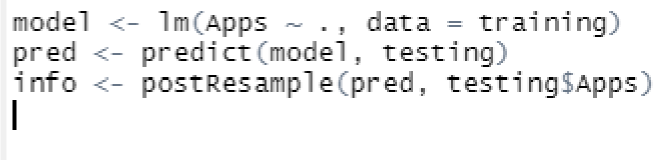
9.

(a)



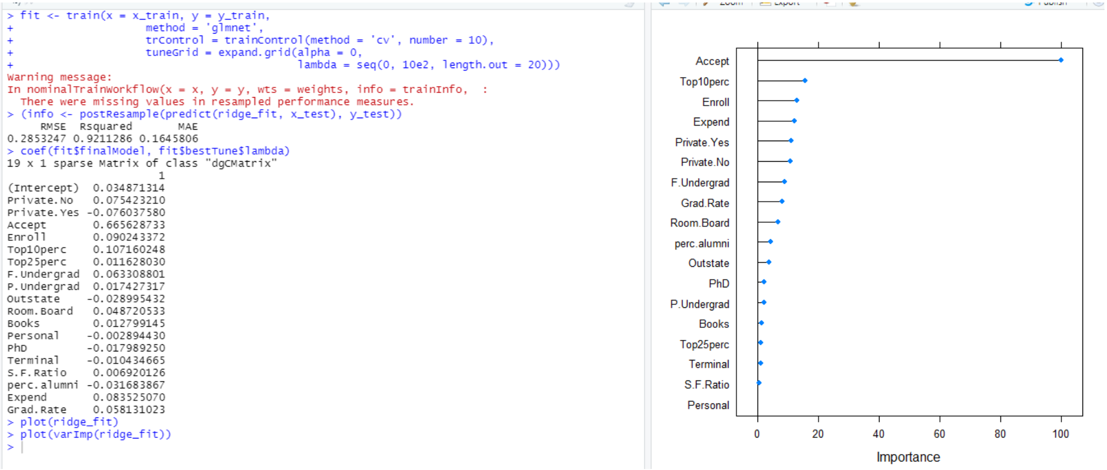
```
library(ISLR)
library(caret)
library(tidyverse)
data('College')
set.seed(1)

inTrain <- createDataPartition(College$Apps, p = 0.75, list = FALSE)

training <- College[inTrain,]
testing <- College[-inTrain,]

Obj <- preProcess(training, method = c('center', 'scale'))

training <- predict(Obj, training)
testing <- predict(Obj, testing)

y_train <- training$Apps
y_test <- testing$Apps

apps <- dummyVars(Apps ~ ., data = training)
x_train <- predict(apps, training)
x_test <- predict(apps, testing)
```

(b)



```
model <- lm(Apps ~ ., data = training)
pred <- predict(model, testing)
info <- postResample(pred, testing$Apps)
```

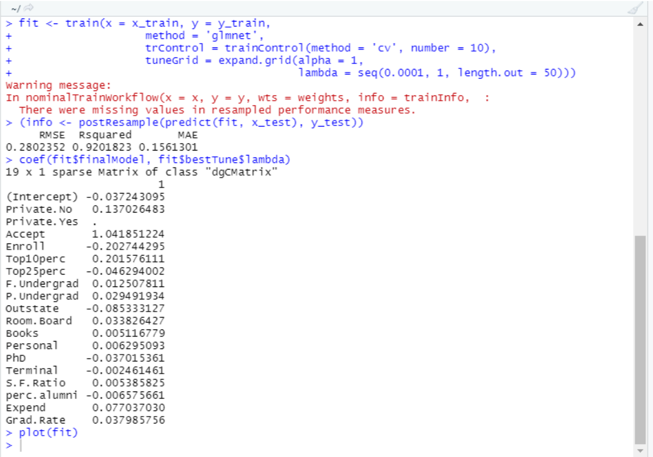
(c)



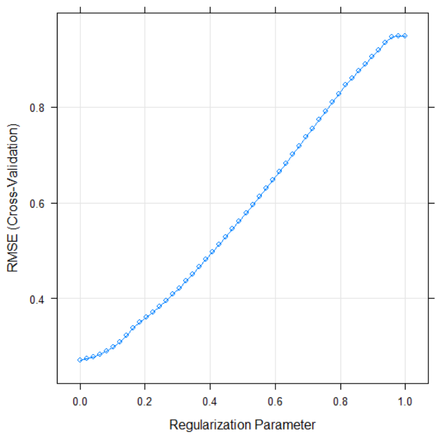
```
> fit <- train(x = x_train, y = y_train,
+                      method = 'glmnet',
+                      trControl = trainControl(method = 'cv', number = 10),
+                      tuneGrid = expand.grid(alpha = 0,
+                                             lambda = seq(0, 10e2, length.out = 20)))
Warning message:
In nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,  :
  There were missing values in resampled performance measures.
> (info <- postResample(predict(ridge_fit, x_test), y_test))
     RMSE  Rsquared       MAE 
0.2853247 0.9211286 0.1645806 
> coef(fit$finalModel, fit$bestTune$lambda)
19 x 1 sparse Matrix of class "dgCMatrix"
                       1
(Intercept)  0.034871314
Private.No   0.075423210
Private.Yes -0.076037580
Accept       0.665628733
Enroll       0.090243372
Top10perc    0.107160248
Top25perc    0.011628030
F.Undergrad  0.063308801
P.Undergrad  0.017427317
Outstate    -0.028995432
Room.Board   0.048720533
Books        0.012799145
Personal    -0.002894430
PhD         -0.017989250
Terminal    -0.010434665
S.F.Ratio    0.006920126
perc.alumni -0.031683867
Expend       0.083525070
Grad.Rate    0.058131023
> plot(ridge_fit)
> plot(varImp(ridge_fit))
> 
```

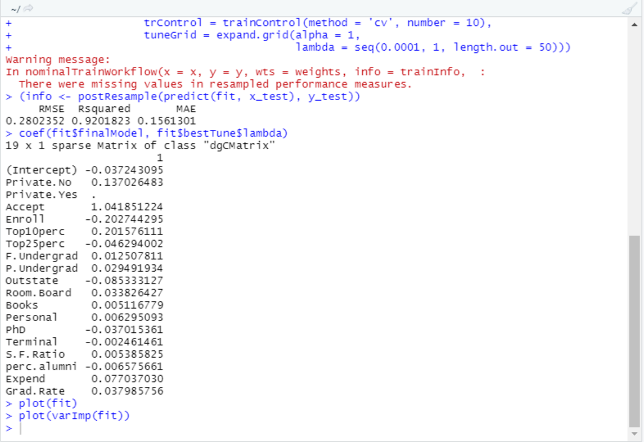
Alexa Summers
Santhoshini Sree Bolisetty
Gireesh Kumar Muppalla
CS 5565—Lab 4

(d)



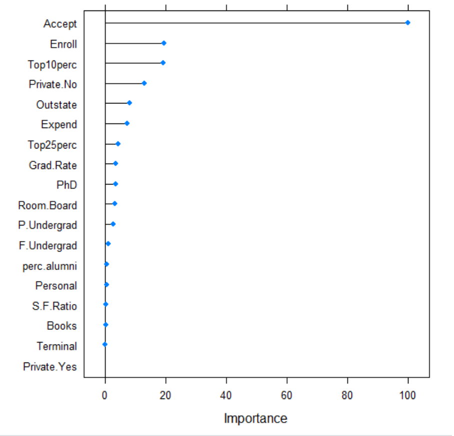
```
> fit <- train(x = x_train, y = y_train,
+                      method = 'glmnet',
+                      trControl = trainControl(method = 'cv', number = 10),
+                      tuneGrid = expand.grid(alpha = 1,
+                                             lambda = seq(0.0001, 1, length.out = 50)))
Warning message:
In nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,  :
  There were missing values in resampled performance measures.
> (info <- postResample(predict(fit, x_test), y_test))
     RMSE  Rsquared       MAE 
0.2802352 0.9201823 0.1561301 
> coef(fit$finalModel, fit$bestTune$lambda)
19 x 1 sparse Matrix of class "dgCMatrix"
                       1
(Intercept) -0.037243095
Private.No   0.137026483
Private.Yes  .          
Accept       1.041851224
Enroll      -0.202744295
Top10perc    0.201576111
Top25perc   -0.046294002
F.Undergrad  0.012507811
P.Undergrad  0.029491934
Outstate    -0.085333127
Room.Board   0.033826427
Books        0.005116779
Personal     0.006295093
PhD         -0.037015361
Terminal    -0.002461461
S.F.Ratio    0.005385825
perc.alumni -0.006575661
Expend       0.077037030
Grad.Rate    0.037985756
> plot(fit)
> 
```





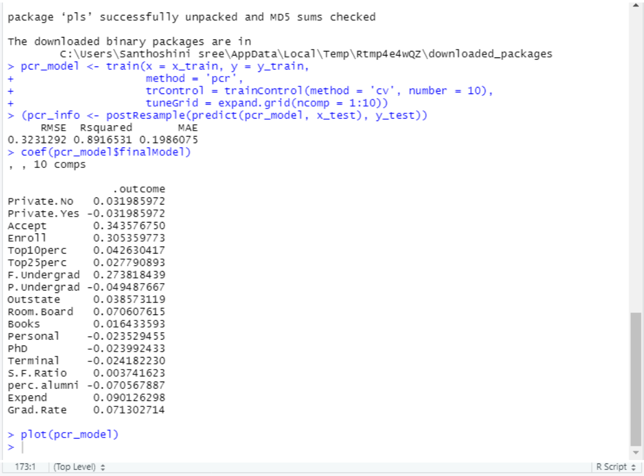
```
+                      trControl = trainControl(method = 'cv', number = 10),
+                      tuneGrid = expand.grid(alpha = 1,
+                                             lambda = seq(0.0001, 1, length.out = 50)))
Warning message:
In nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,  :
  There were missing values in resampled performance measures.
> (info <- postResample(predict(fit, x_test), y_test))
     RMSE  Rsquared       MAE 
0.2802352 0.9201823 0.1561301 
> coef(fit$finalModel, fit$bestTune$lambda)
19 x 1 sparse Matrix of class "dgCMatrix"
                       1
(Intercept) -0.037243095
Private.No   0.137026483
Private.Yes  .          
Accept       1.041851224
Enroll      -0.202744295
Top10perc    0.201576111
Top25perc   -0.046294002
F.Undergrad  0.012507811
P.Undergrad  0.029491934
Outstate    -0.085333127
Room.Board   0.033826427
Books        0.005116779
Personal     0.006295093
PhD         -0.037015361
Terminal    -0.002461461
S.F.Ratio    0.005385825
perc.alumni -0.006575661
Expend       0.077037030
Grad.Rate    0.037985756
> plot(fit)
```

```
> plot(varImp(fit))
> 
```

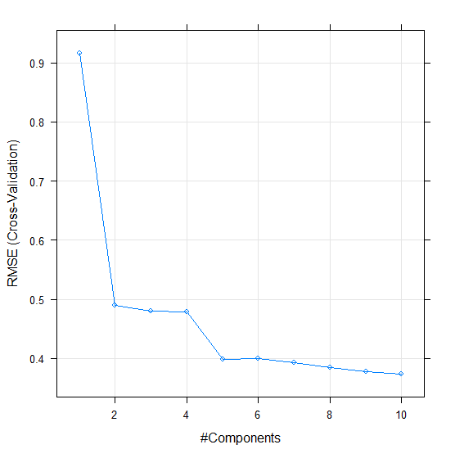


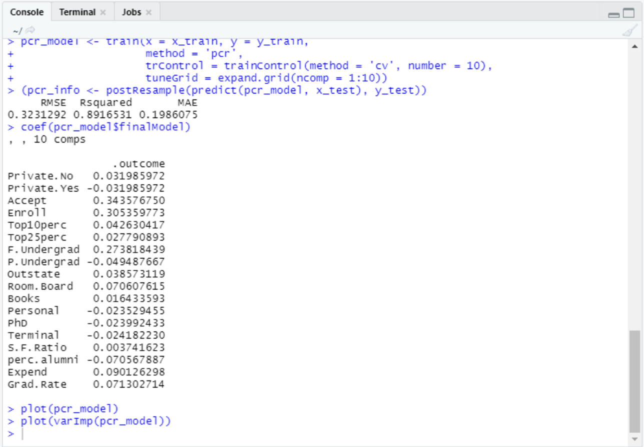
Alexa Summers
Santhoshini Sree Bolisetty
Gireesh Kumar Muppalla
CS 5565—Lab 4

(e)



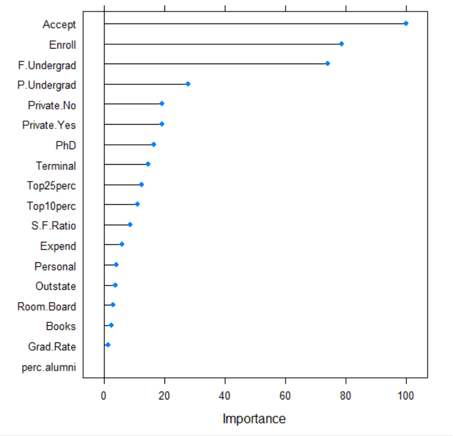
```
package ‘pls’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Santhoshini sree\AppData\Local\Temp\Rtmp4e4wQZ\downloaded_packages
> pcr_model <- train(x = x_train, y = y_train,
+                    method = 'pcr',
+                    trControl = trainControl(method = 'cv', number = 10),
+                    tuneGrid = expand.grid(ncomp = 1:10))
> (pcr_info <- postResample(predict(pcr_model, x_test), y_test))
     RMSE  Rsquared       MAE 
0.3231292 0.8916531 0.1986075 
> coef(pcr_model$finalModel)
, , 10 comps

                .outcome
Private.No   0.031985972
Private.Yes -0.031985972
Accept       0.343576750
Enroll       0.305359773
Top10perc    0.042630417
Top25perc    0.027790893
F.Undergrad  0.273818439
P.Undergrad -0.049487667
Outstate     0.038573119
Room.Board   0.070607615
Books        0.016433593
Personal    -0.023529455
PhD         -0.023992433
Terminal    -0.024182230
S.F.Ratio    0.003741623
perc.alumni -0.070567887
Expend       0.090126298
Grad.Rate    0.071302714

> plot(pcr_model)
> 
```



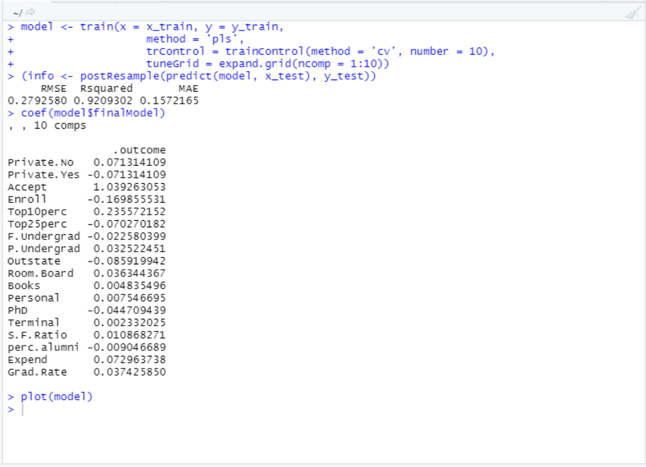


```
> pcr_model <- train(x = x_train, y = y_train,
+                    method = 'pcr',
+                    trControl = trainControl(method = 'cv', number = 10),
+                    tuneGrid = expand.grid(ncomp = 1:10))
> (pcr_info <- postResample(predict(pcr_model, x_test), y_test))
     RMSE  Rsquared       MAE 
0.3231292 0.8916531 0.1986075 
> coef(pcr_model$finalModel)
, , 10 comps

                .outcome
Private.No   0.031985972
Private.Yes -0.031985972
Accept       0.343576750
Enroll       0.305359773
Top10perc    0.042630417
Top25perc    0.027790893
F.Undergrad  0.273818439
P.Undergrad -0.049487667
Outstate     0.038573119
Room.Board   0.070607615
Books        0.016433593
Personal    -0.023529455
PhD         -0.023992433
Terminal    -0.024182230
S.F.Ratio    0.003741623
perc.alumni -0.070567887
Expend       0.090126298
Grad.Rate    0.071302714

> plot(pcr_model)
```

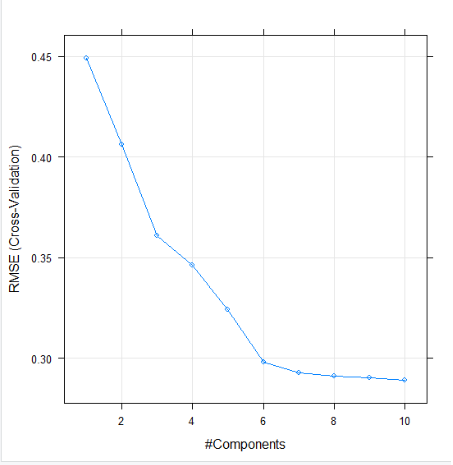
```
> plot(varImp(pcr_model))
> 
```



(f)

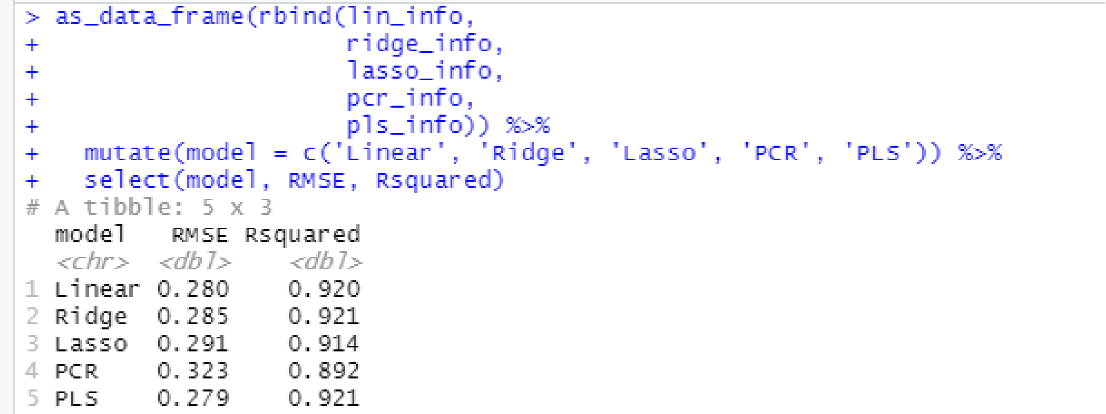


```
> model <- train(x = x_train, y = y_train,
+                    method = 'pls',
+                    trControl = trainControl(method = 'cv', number = 10),
+                    tuneGrid = expand.grid(ncomp = 1:10))
> (info <- postResample(predict(model, x_test), y_test))
     RMSE  Rsquared       MAE 
0.2792580 0.9209302 0.1572165 
> coef(model$finalModel)
, , 10 comps

                .outcome
Private.No   0.071314109
Private.Yes -0.071314109
Accept       1.039263053
Enroll      -0.169855531
Top10perc    0.235572152
Top25perc   -0.070270182
F.Undergrad -0.022580399
P.Undergrad  0.032522451
Outstate    -0.085919942
Room.Board   0.036344367
Books        0.004835496
Personal     0.007546695
PhD         -0.044709439
Terminal     0.002332025
S.F.Ratio    0.010868271
perc.alumni -0.009046689
Expend       0.072963738
Grad.Rate    0.037425850

> plot(model)
> 
```

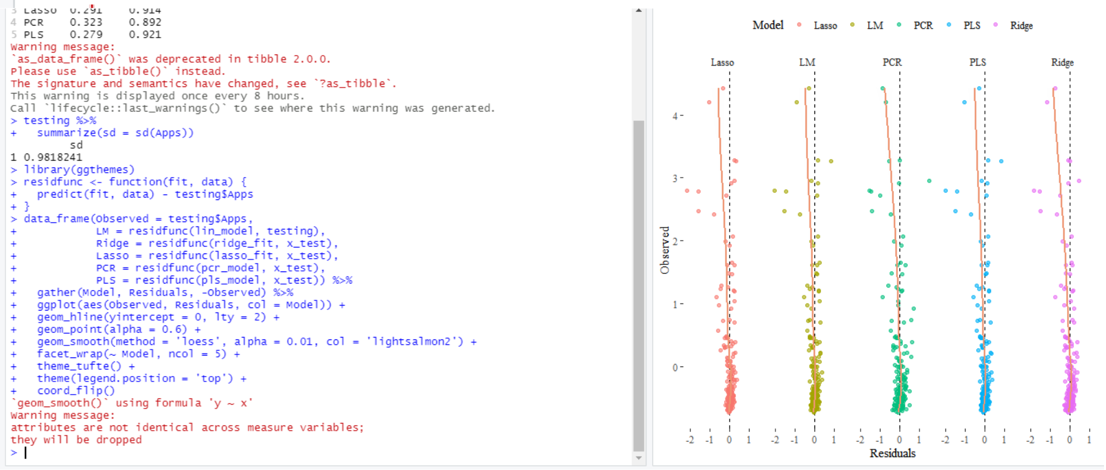


Alexa Summers
Santhoshini Sree Bolisetty
Gireesh Kumar Muppalla
CS 5565—Lab 4

(g)



```
> as_data_frame(rbind(lin_info,
+                     ridge_info,
+                     lasso_info,
+                     pcr_info,
+                     pls_info)) %>%
+   mutate(model = c('Linear', 'Ridge', 'Lasso', 'PCR', 'PLS')) %>%
+   select(model, RMSE, Rsquared)
# A tibble: 5 x 3
  model   RMSE Rsquared
  <chr>  <dbl>    <dbl>
1 Linear 0.280    0.920
2 Ridge  0.285    0.921
3 Lasso  0.291    0.914
4 PCR    0.323    0.892
5 PLS    0.279    0.921
```

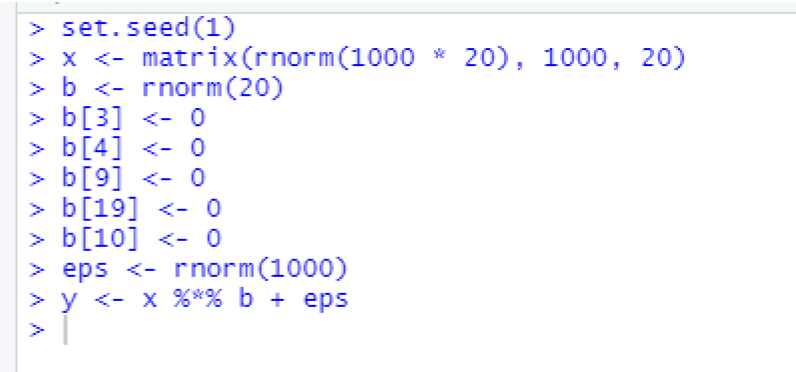


```
3 Lasso  0.291    0.914
4 PCR    0.323    0.892
5 PLS    0.279    0.921
Warning message:
`as_data_frame()` was deprecated in tibble 2.0.0.
Please use `as_tibble()` instead.
The signature and semantics have changed, see `?as_tibble`.
This warning is displayed once every 8 hours.
Call `lifecycle::last_warnings()` to see where this warning was generated.
> testing %>%
+   summarize(sd = sd(Apps))
         sd
1 0.9818241
> library(ggthemes)
> residfunc <- function(fit, data) {
+   predict(fit, data) - testing$Apps
+ }
> data_frame(Observed = testing$Apps,
+            LM = residfunc(lin_model, testing),
+            Ridge = residfunc(ridge_fit, x_test),
+            Lasso = residfunc(lasso_fit, x_test),
+            PCR = residfunc(pcr_model, x_test),
+            PLS = residfunc(pls_model, x_test)) %>%
+   gather(Model, Residuals, -Observed) %>%
+   ggplot(aes(Observed, Residuals, col = Model)) +
+   geom_hline(yintercept = 0, lty = 2) +
+   geom_point(alpha = 0.6) +
+   geom_smooth(method = 'loess', alpha = 0.01, col = 'lightsalmon2') +
+   facet_wrap(~ Model, ncol = 5) +
+   theme_tufte() +
+   theme(legend.position = 'top') +
+   coord_flip()
`geom_smooth()` using formula 'y ~ x'
Warning message:
attributes are not identical across measure variables;
they will be dropped
> |
```

Alexa Summers
Santhoshini Sree Bolisetty
Gireesh Kumar Muppalla
CS 5565—Lab 4

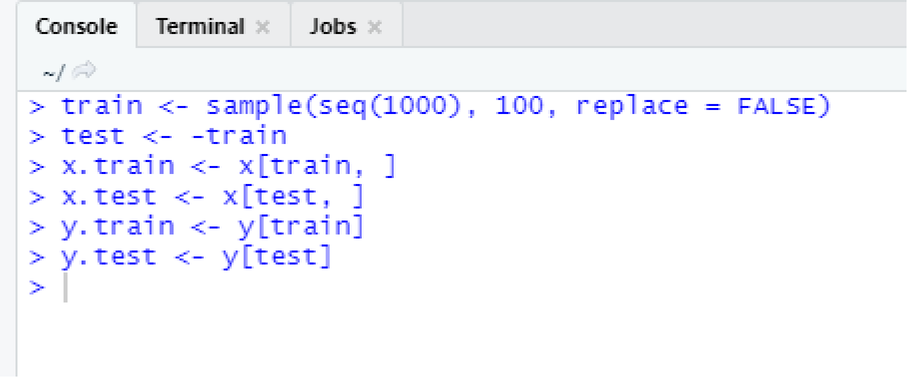
10.

(a)



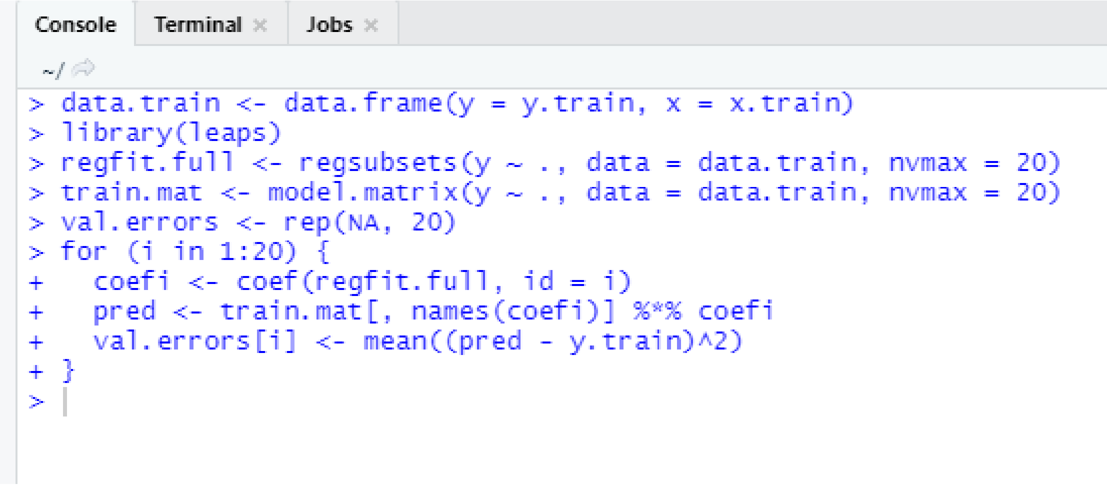
```
> set.seed(1)
> x <- matrix(rnorm(1000 * 20), 1000, 20)
> b <- rnorm(20)
> b[3] <- 0
> b[4] <- 0
> b[9] <- 0
> b[19] <- 0
> b[10] <- 0
> eps <- rnorm(1000)
> y <- x %*% b + eps
> 
```

(b)



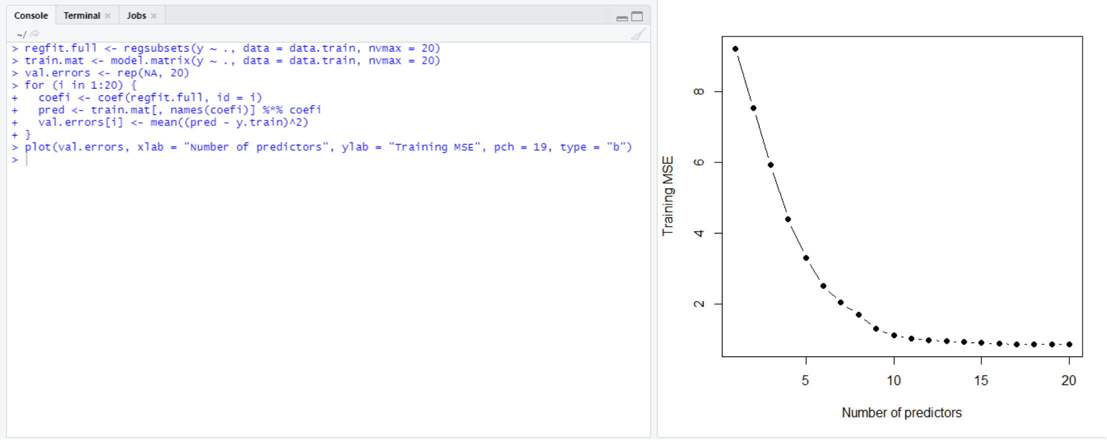
```
> train <- sample(seq(1000), 100, replace = FALSE)
> test <- -train
> x.train <- x[train, ]
> x.test <- x[test, ]
> y.train <- y[train]
> y.test <- y[test]
> 
```

(c)



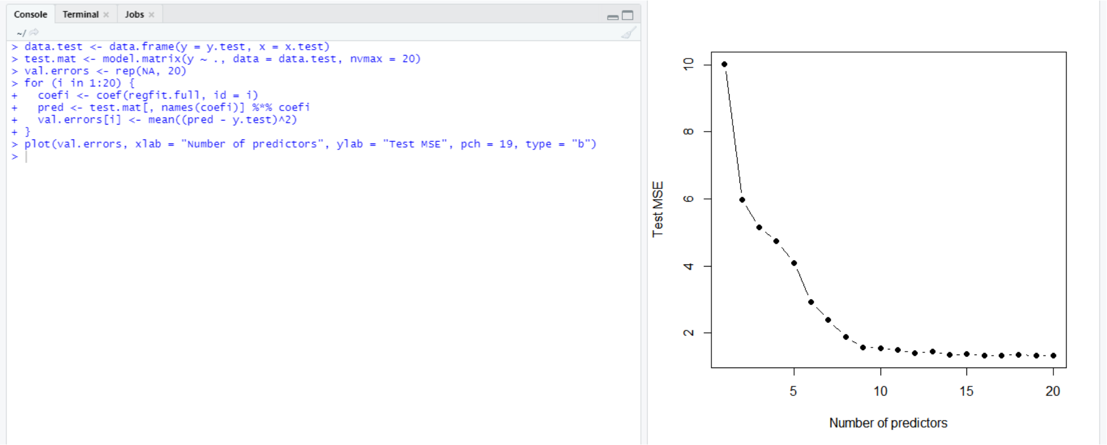
```
> data.train <- data.frame(y = y.train, x = x.train)
> library(leaps)
> regfit.full <- regsubsets(y ~ ., data = data.train, nvmax = 20)
> train.mat <- model.matrix(y ~ ., data = data.train, nvmax = 20)
> val.errors <- rep(NA, 20)
> for (i in 1:20) {
+   coefi <- coef(regfit.full, id = i)
+   pred <- train.mat[, names(coefi)] %*% coefi
+   val.errors[i] <- mean((pred - y.train)^2)
+ }
> 
```

Alexa Summers
Santhoshini Sree Bolisetty
Gireesh Kumar Muppalla
CS 5565—Lab 4



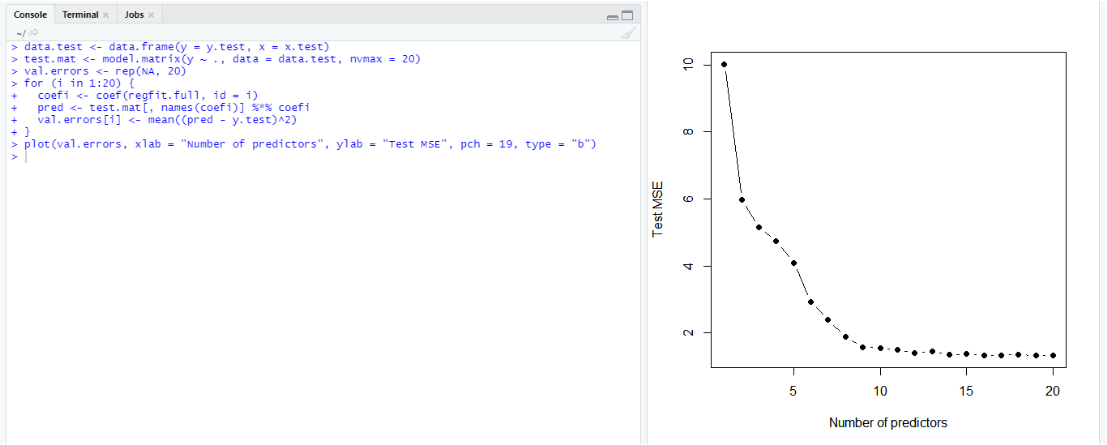
```
> regfit.full <- regsubsets(y ~ ., data = data.train, nvmax = 20)
> train.mat <- model.matrix(y ~ ., data = data.train, nvmax = 20)
> val.errors <- rep(NA, 20)
> for (i in 1:20) {
+   coefi <- coef(regfit.full, id = i)
+   pred <- train.mat[, names(coefi)] %*% coefi
+   val.errors[i] <- mean((pred - y.train)^2)
+ }
> plot(val.errors, xlab = "Number of predictors", ylab = "Training MSE", pch = 19, type = "b")
> 
```

(d)



```
> data.test <- data.frame(y = y.test, x = x.test)
> test.mat <- model.matrix(y ~ ., data = data.test, nvmax = 20)
> val.errors <- rep(NA, 20)
> for (i in 1:20) {
+   coefi <- coef(regfit.full, id = i)
+   pred <- test.mat[, names(coefi)] %*% coefi
+   val.errors[i] <- mean((pred - y.test)^2)
+ }
> plot(val.errors, xlab = "Number of predictors", ylab = "Test MSE", pch = 19, type = "b")
> 
```

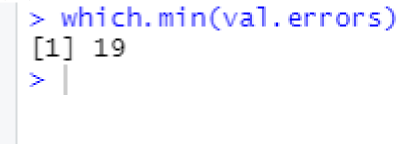
(d)



```
> data.test <- data.frame(y = y.test, x = x.test)
> test.mat <- model.matrix(y ~ ., data = data.test, nvmax = 20)
> val.errors <- rep(NA, 20)
> for (i in 1:20) {
+   coefi <- coef(regfit.full, id = i)
+   pred <- test.mat[, names(coefi)] %*% coefi
+   val.errors[i] <- mean((pred - y.test)^2)
+ }
> plot(val.errors, xlab = "Number of predictors", ylab = "Test MSE", pch = 19, type = "b")
> 
```

Alexa Summers
Santhoshini Sree Bolisetty
Gireesh Kumar Muppalla
CS 5565—Lab 4

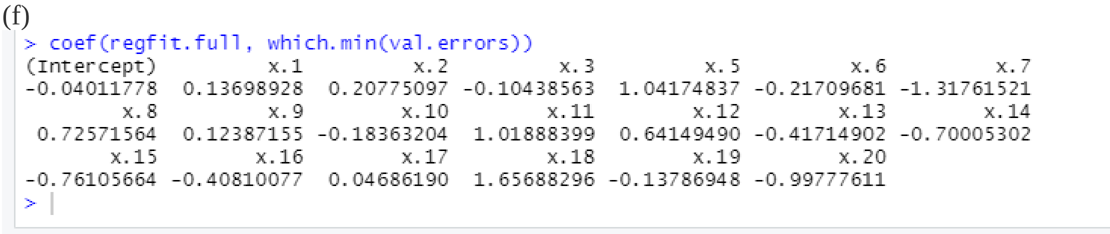
(e)



```
> which.min(val.errors)
[1] 19
> |
```

14-variables model has the smallest test MSE.

(f)

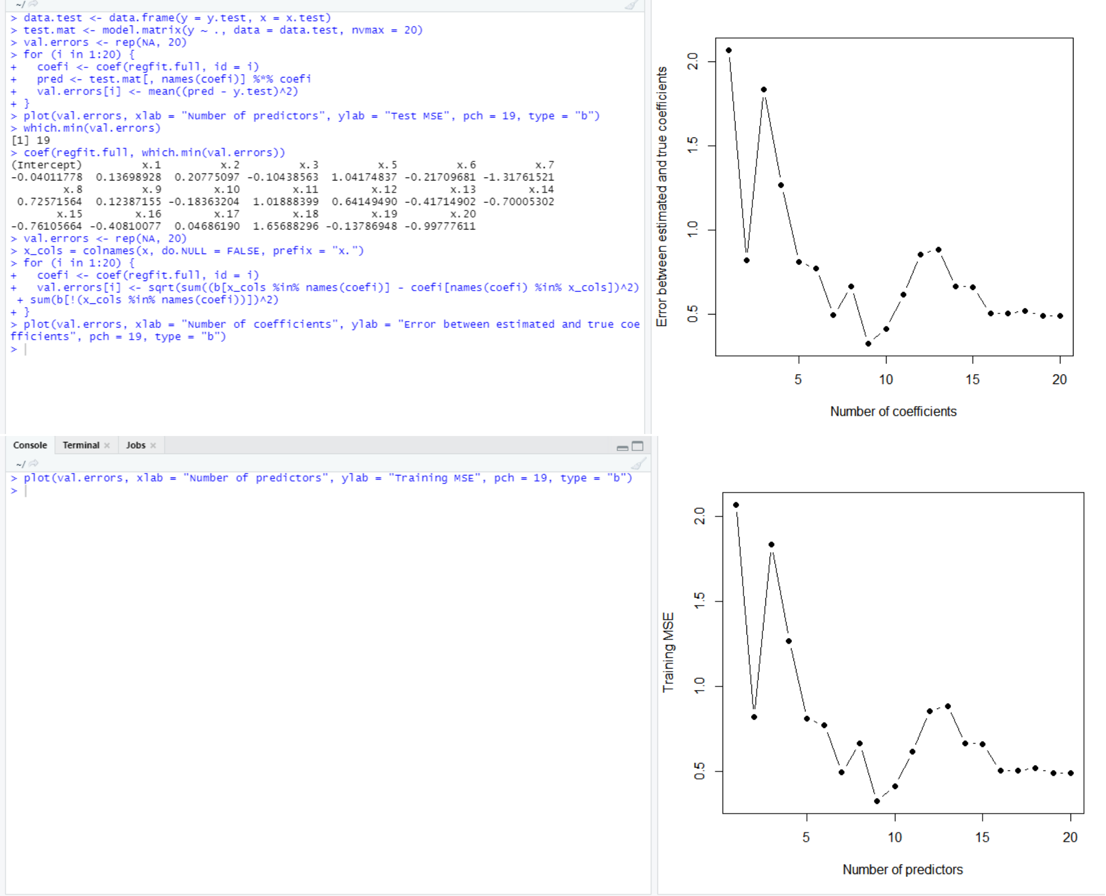


```
> coef(regfit.full, which.min(val.errors))
(Intercept)         x.1         x.2         x.3         x.5         x.6         x.7 
-0.04011778  0.13698928  0.20775097 -0.10438563  1.04174837 -0.21709681 -1.31761521 
        x.8         x.9        x.10        x.11        x.12        x.13        x.14 
 0.72571564  0.12387155 -0.18363204  1.01888399  0.64149490 -0.41714902 -0.70005302 
       x.15        x.16        x.17        x.18        x.19        x.20 
-0.76105664 -0.40810077  0.04686190  1.65688296 -0.13786948 -0.99777611 
> |
```

The best model caught all zeroed out coefficients.

(g)

We may see that the model with 3 variables minimizes the error between the estimated and true coefficients. However test error is minimized by the model with 14 variables. So, a better fit of true coefficients doesn't necessarily mean a lower test MSE.



```
> data.test <- data.frame(y = y.test, x = x.test)
> test.mat <- model.matrix(y ~ ., data = data.test, nvmax = 20)
> val.errors <- rep(NA, 20)
> for (i in 1:20) {
+   coefi <- coef(regfit.full, id = i)
+   pred <- test.mat[, names(coefi)] %*% coefi
+   val.errors[i] <- mean((pred - y.test)^2)
+ }
> plot(val.errors, xlab = "Number of predictors", ylab = "Test MSE", pch = 19, type = "b")
> which.min(val.errors)
[1] 19
> coef(regfit.full, which.min(val.errors))
(Intercept)         x.1         x.2         x.3         x.5         x.6         x.7 
-0.04011778  0.13698928  0.20775097 -0.10438563  1.04174837 -0.21709681 -1.31761521 
        x.8         x.9        x.10        x.11        x.12        x.13        x.14 
 0.72571564  0.12387155 -0.18363204  1.01888399  0.64149490 -0.41714902 -0.70005302 
       x.15        x.16        x.17        x.18        x.19        x.20 
-0.76105664 -0.40810077  0.04686190  1.65688296 -0.13786948 -0.99777611 
> val.errors <- rep(NA, 20)
> x_cols = colnames(x, do.NULL = FALSE, prefix = "x.")
> for (i in 1:20) {
+   coefi <- coef(regfit.full, id = i)
+   val.errors[i] <- sqrt(sum((b[x_cols %in% names(coefi)] - coefi[names(coefi) %in% x_cols])^2)
 + sum(b[!(x_cols %in% names(coefi))])^2)
+ }
> plot(val.errors, xlab = "Number of coefficients", ylab = "Error between estimated and true coe
fficients", pch = 19, type = "b")
> |
```

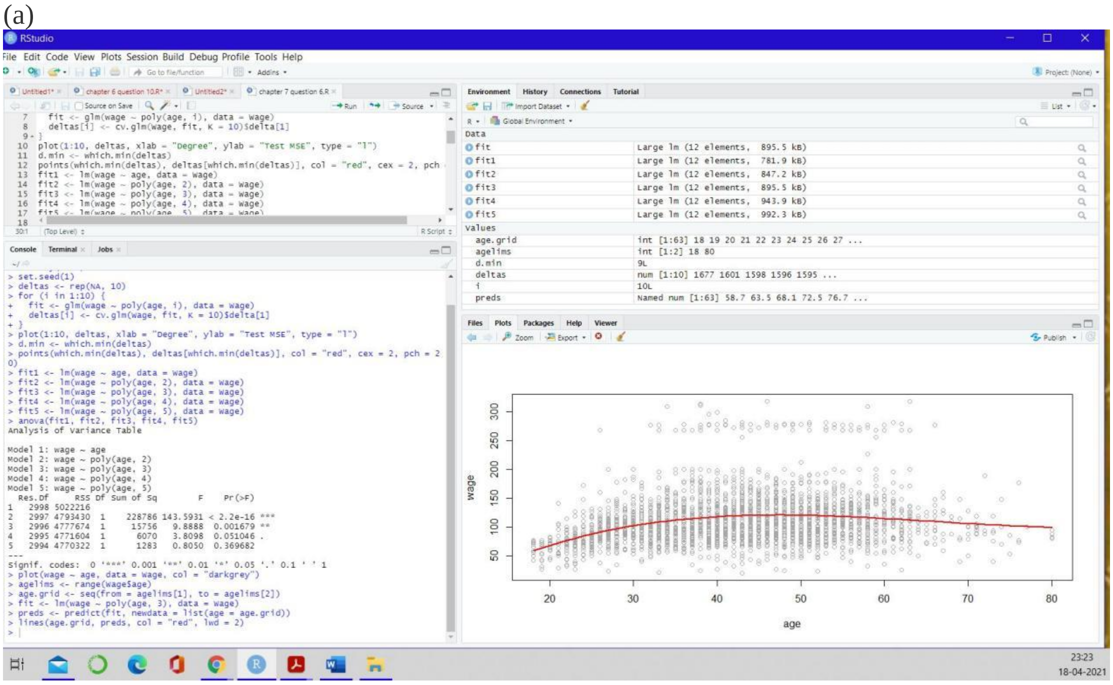
```
> plot(val.errors, xlab = "Number of predictors", ylab = "Training MSE", pch = 19, type = "b")
> |
```

Alexa Summers
Santhoshini Sree Bolisetty
Gireesh Kumar Muppalla
CS 5565—Lab 4

CHAPTER 7

6.

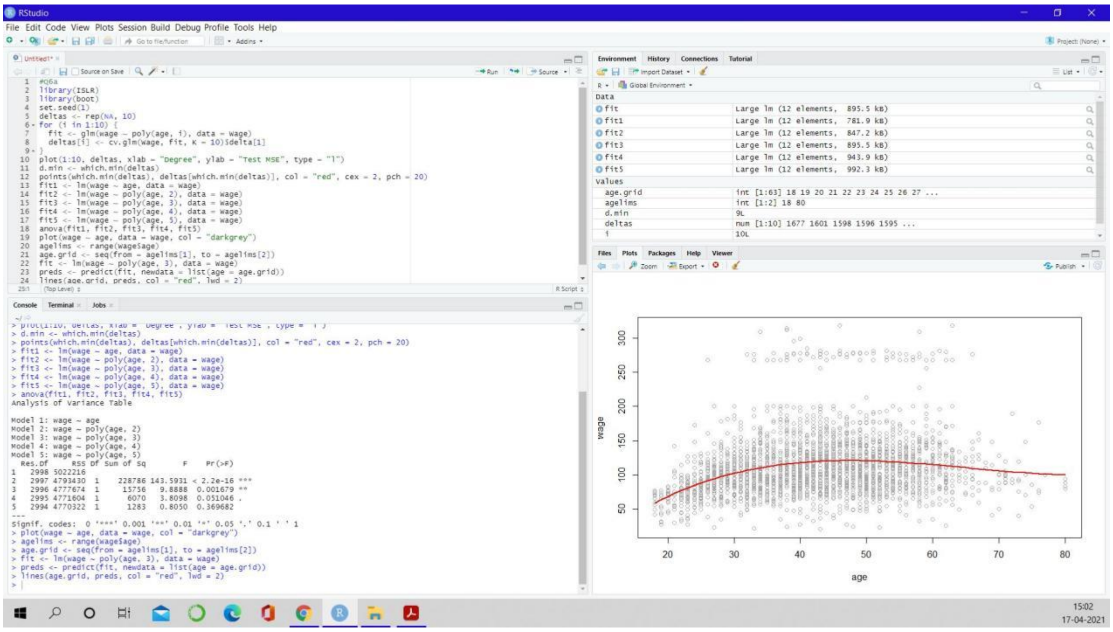
(a)



```
  fit <- glm(wage ~ poly(age, i), data = wage)
  deltas[i] <- cv.glm(wage, fit, K = 10)$delta[1]
}
plot(1:10, deltas, xlab = "Degree", ylab = "Test MSE", type = "l")
d.min <- which.min(deltas)
points(which.min(deltas), deltas[which.min(deltas)], col = "red", cex = 2, pch
fit1 <- lm(wage ~ age, data = wage)
fit2 <- lm(wage ~ poly(age, 2), data = wage)
fit3 <- lm(wage ~ poly(age, 3), data = wage)
fit4 <- lm(wage ~ poly(age, 4), data = wage)
fit5 <- lm(wage ~ poly(age, 5), data = wage)
```

```
> set.seed(1)
> deltas <- rep(NA, 10)
> for (i in 1:10) {
+   fit <- glm(wage ~ poly(age, i), data = wage)
+   deltas[i] <- cv.glm(wage, fit, K = 10)$delta[1]
+ }
> plot(1:10, deltas, xlab = "Degree", ylab = "Test MSE", type = "l")
> d.min <- which.min(deltas)
> points(which.min(deltas), deltas[which.min(deltas)], col = "red", cex = 2, pch = 2
0)
> fit1 <- lm(wage ~ age, data = wage)
> fit2 <- lm(wage ~ poly(age, 2), data = wage)
> fit3 <- lm(wage ~ poly(age, 3), data = wage)
> fit4 <- lm(wage ~ poly(age, 4), data = wage)
> fit5 <- lm(wage ~ poly(age, 5), data = wage)
> anova(fit1, fit2, fit3, fit4, fit5)
Analysis of Variance Table

Model 1: wage ~ age
Model 2: wage ~ poly(age, 2)
Model 3: wage ~ poly(age, 3)
Model 4: wage ~ poly(age, 4)
Model 5: wage ~ poly(age, 5)
  Res.Df     RSS Df Sum of Sq        F    Pr(>F)
1   2998 5022216
2   2997 4793430  1    228786 143.5931 < 2.2e-16 ***
3   2996 4777674  1     15756   9.8888  0.001679 **
4   2995 4771604  1      6070   3.8098  0.051046 .
5   2994 4770322  1      1283   0.8050  0.369682
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> plot(wage ~ age, data = wage, col = "darkgrey")
> agelims <- range(wage$age)
> age.grid <- seq(from = agelims[1], to = agelims[2])
> fit <- lm(wage ~ poly(age, 3), data = wage)
> preds <- predict(fit, newdata = list(age = age.grid))
> lines(age.grid, preds, col = "red", lwd = 2)
> |
```

```
Data
fit         Large lm (12 elements,  895.5 kB)
fit1        Large lm (12 elements,  781.9 kB)
fit2        Large lm (12 elements,  847.2 kB)
fit3        Large lm (12 elements,  895.5 kB)
fit4        Large lm (12 elements,  943.9 kB)
fit5        Large lm (12 elements,  992.3 kB)
Values
  age.grid  int [1:63] 18 19 20 21 22 23 24 25 26 27 ...
  agelims   int [1:2] 18 80
  d.min     9L
  deltas    num [1:10] 1677 1601 1598 1596 1595 ...
  i         10L
  preds     Named num [1:63] 58.7 63.5 68.1 72.5 76.7 ...
```



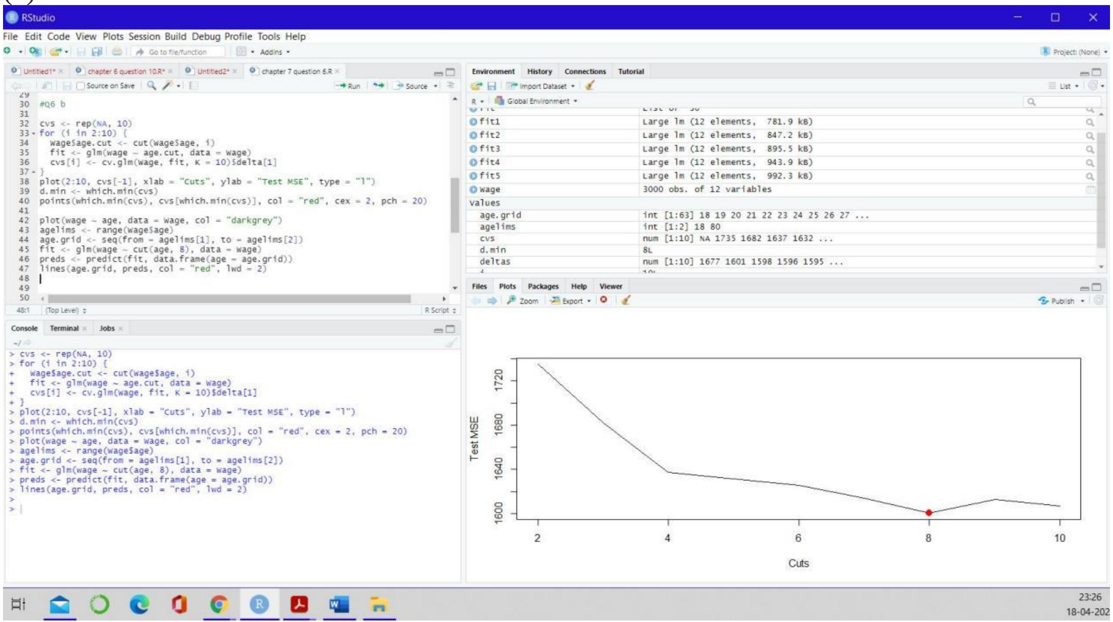
```
#Q6a
library(ISLR)
library(boot)
set.seed(1)
deltas <- rep(NA, 10)
for (i in 1:10) {
  fit <- glm(wage ~ poly(age, i), data = wage)
  deltas[i] <- cv.glm(wage, fit, K = 10)$delta[1]
}
plot(1:10, deltas, xlab = "Degree", ylab = "Test MSE", type = "l")
d.min <- which.min(deltas)
points(which.min(deltas), deltas[which.min(deltas)], col = "red", cex = 2, pch = 20)
fit1 <- lm(wage ~ age, data = wage)
fit2 <- lm(wage ~ poly(age, 2), data = wage)
fit3 <- lm(wage ~ poly(age, 3), data = wage)
fit4 <- lm(wage ~ poly(age, 4), data = wage)
fit5 <- lm(wage ~ poly(age, 5), data = wage)
anova(fit1, fit2, fit3, fit4, fit5)
plot(wage ~ age, data = wage, col = "darkgrey")
agelims <- range(wage$age)
age.grid <- seq(from = agelims[1], to = agelims[2])
fit <- lm(wage ~ poly(age, 3), data = wage)
preds <- predict(fit, newdata = list(age = age.grid))
lines(age.grid, preds, col = "red", lwd = 2)
```

```
> plot(1:10, deltas, xlab = "Degree", ylab = "Test MSE", type = "l")
> d.min <- which.min(deltas)
> points(which.min(deltas), deltas[which.min(deltas)], col = "red", cex = 2, pch = 20)
> fit1 <- lm(wage ~ age, data = wage)
> fit2 <- lm(wage ~ poly(age, 2), data = wage)
> fit3 <- lm(wage ~ poly(age, 3), data = wage)
> fit4 <- lm(wage ~ poly(age, 4), data = wage)
> fit5 <- lm(wage ~ poly(age, 5), data = wage)
> anova(fit1, fit2, fit3, fit4, fit5)
Analysis of Variance Table

Model 1: wage ~ age
Model 2: wage ~ poly(age, 2)
Model 3: wage ~ poly(age, 3)
Model 4: wage ~ poly(age, 4)
Model 5: wage ~ poly(age, 5)
  Res.Df     RSS Df Sum of Sq        F    Pr(>F)
1   2998 5022216
2   2997 4793430  1    228786 143.5931 < 2.2e-16 ***
3   2996 4777674  1     15756   9.8888  0.001679 **
4   2995 4771604  1      6070   3.8098  0.051046 .
5   2994 4770322  1      1283   0.8050  0.369682
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> plot(wage ~ age, data = wage, col = "darkgrey")
> agelims <- range(wage$age)
> age.grid <- seq(from = agelims[1], to = agelims[2])
> fit <- lm(wage ~ poly(age, 3), data = wage)
> preds <- predict(fit, newdata = list(age = age.grid))
> lines(age.grid, preds, col = "red", lwd = 2)
> |
```

```
Data
fit         Large lm (12 elements,  895.5 kB)
fit1        Large lm (12 elements,  781.9 kB)
fit2        Large lm (12 elements,  847.2 kB)
fit3        Large lm (12 elements,  895.5 kB)
fit4        Large lm (12 elements,  943.9 kB)
fit5        Large lm (12 elements,  992.3 kB)
Values
  age.grid  int [1:63] 18 19 20 21 22 23 24 25 26 27 ...
  agelims   int [1:2] 18 80
  d.min     9L
  deltas    num [1:10] 1677 1601 1598 1596 1595 ...
  i         10L
```

Alexa Summers
Santhoshini Sree Bolisetty
Gireesh Kumar Muppalla
CS 5565—Lab 4

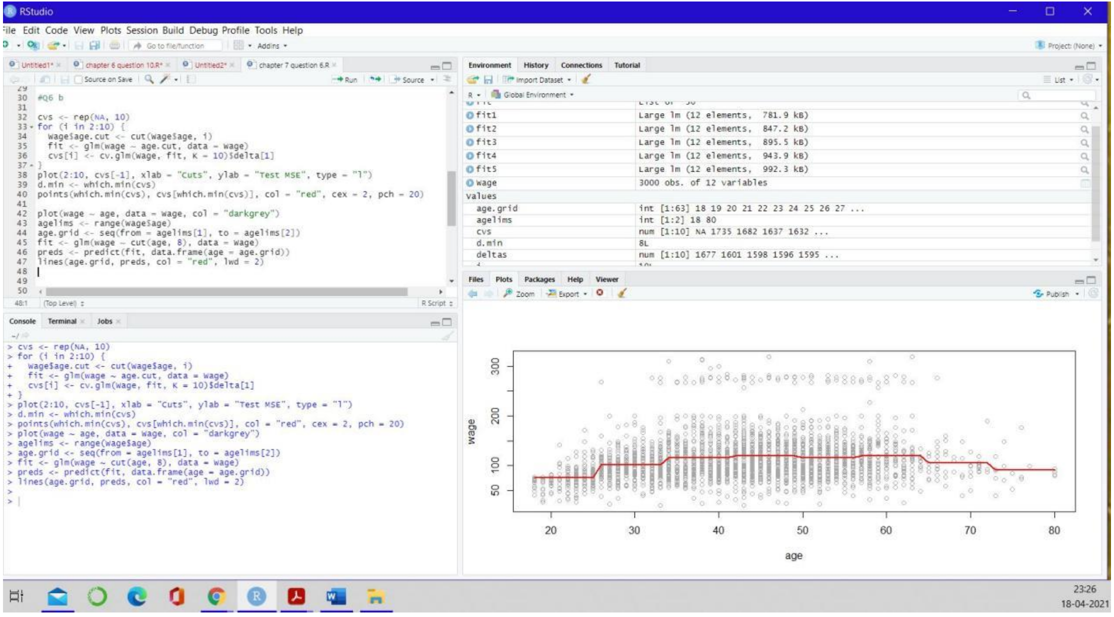
(b)


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Untitled1* chapter 6 question 10.R* Untitled2* chapter 7 question 6.R
Run Source
Environment History Connections Tutorial
Files Plots Packages Help Viewer
Test MSE
1600
1640
1680
1720
2
4
6
8
10
Cuts
23:26
18-04-202

```
#Q6 b

cvs <- rep(NA, 10)
for (i in 2:10) {
  wage$age.cut <- cut(wage$age, i)
  fit <- glm(wage ~ age.cut, data = wage)
  cvs[i] <- cv.glm(wage, fit, K = 10)$delta[1]
}
plot(2:10, cvs[-1], xlab = "Cuts", ylab = "Test MSE", type = "l")
d.min <- which.min(cvs)
points(which.min(cvs), cvs[which.min(cvs)], col = "red", cex = 2, pch = 20)

plot(wage ~ age, data = wage, col = "darkgrey")
agelims <- range(wage$age)
age.grid <- seq(from = agelims[1], to = agelims[2])
fit <- glm(wage ~ cut(age, 8), data = wage)
preds <- predict(fit, data.frame(age = age.grid))
lines(age.grid, preds, col = "red", lwd = 2)
```

```
> cvs <- rep(NA, 10)
> for (i in 2:10) {
+   wage$age.cut <- cut(wage$age, i)
+   fit <- glm(wage ~ age.cut, data = wage)
+   cvs[i] <- cv.glm(wage, fit, K = 10)$delta[1]
+ }
> plot(2:10, cvs[-1], xlab = "Cuts", ylab = "Test MSE", type = "l")
> d.min <- which.min(cvs)
> points(which.min(cvs), cvs[which.min(cvs)], col = "red", cex = 2, pch = 20)
> plot(wage ~ age, data = wage, col = "darkgrey")
> agelims <- range(wage$age)
> age.grid <- seq(from = agelims[1], to = agelims[2])
> fit <- glm(wage ~ cut(age, 8), data = wage)
> preds <- predict(fit, data.frame(age = age.grid))
> lines(age.grid, preds, col = "red", lwd = 2)
>
> |
```

```
fit1        Large lm (12 elements,  781.9 kB)
fit2        Large lm (12 elements,  847.2 kB)
fit3        Large lm (12 elements,  895.5 kB)
fit4        Large lm (12 elements,  943.9 kB)
fit5        Large lm (12 elements,  992.3 kB)
wage        3000 obs. of 12 variables
values
  age.grid  int [1:63] 18 19 20 21 22 23 24 25 26 27 ...
  agelims   int [1:2] 18 80
  cvs       num [1:10] NA 1735 1682 1637 1632 ...
  d.min     8L
  deltas    num [1:10] 1677 1601 1598 1596 1595 ...
```


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Untitled1* chapter 6 question 10.R* Untitled2* chapter 7 question 6.R
Run Source
Environment History Connections Tutorial
Files Plots Packages Help Viewer
wage
50
100
200
300
20
30
40
50
60
70
80
age
23:26
18-04-2021

```
#Q6 b

cvs <- rep(NA, 10)
for (i in 2:10) {
  wage$age.cut <- cut(wage$age, i)
  fit <- glm(wage ~ age.cut, data = wage)
  cvs[i] <- cv.glm(wage, fit, K = 10)$delta[1]
}
plot(2:10, cvs[-1], xlab = "Cuts", ylab = "Test MSE", type = "l")
d.min <- which.min(cvs)
points(which.min(cvs), cvs[which.min(cvs)], col = "red", cex = 2, pch = 20)

plot(wage ~ age, data = wage, col = "darkgrey")
agelims <- range(wage$age)
age.grid <- seq(from = agelims[1], to = agelims[2])
fit <- glm(wage ~ cut(age, 8), data = wage)
preds <- predict(fit, data.frame(age = age.grid))
lines(age.grid, preds, col = "red", lwd = 2)
```

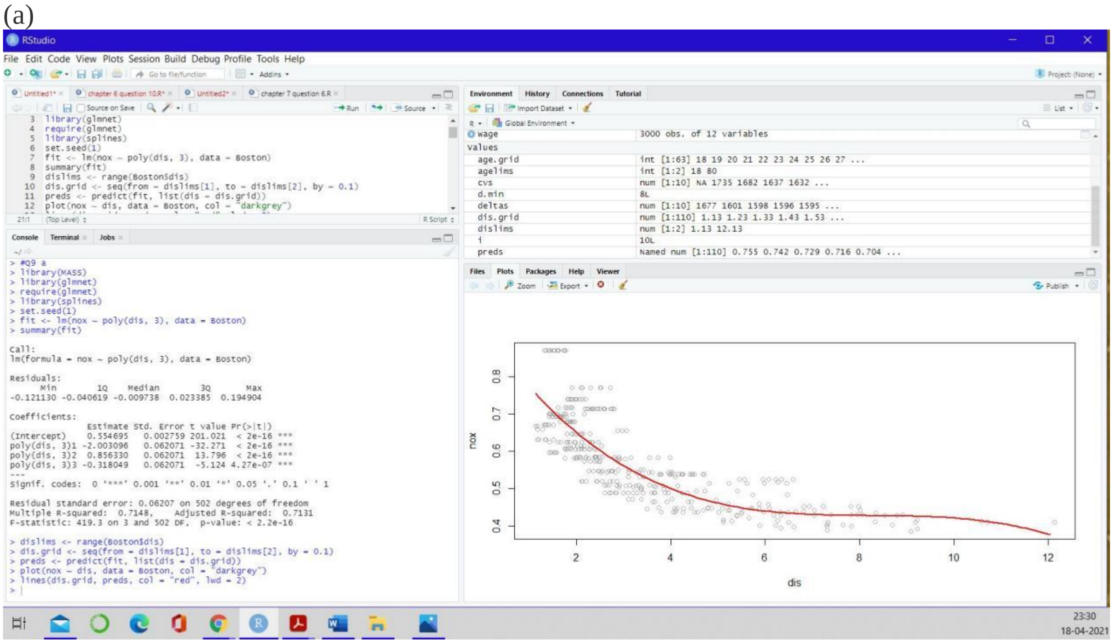
```
> cvs <- rep(NA, 10)
> for (i in 2:10) {
+   wage$age.cut <- cut(wage$age, i)
+   fit <- glm(wage ~ age.cut, data = wage)
+   cvs[i] <- cv.glm(wage, fit, K = 10)$delta[1]
+ }
> plot(2:10, cvs[-1], xlab = "Cuts", ylab = "Test MSE", type = "l")
> d.min <- which.min(cvs)
> points(which.min(cvs), cvs[which.min(cvs)], col = "red", cex = 2, pch = 20)
> plot(wage ~ age, data = wage, col = "darkgrey")
> agelims <- range(wage$age)
> age.grid <- seq(from = agelims[1], to = agelims[2])
> fit <- glm(wage ~ cut(age, 8), data = wage)
> preds <- predict(fit, data.frame(age = age.grid))
> lines(age.grid, preds, col = "red", lwd = 2)
>
> |
```

```
fit1        Large lm (12 elements,  781.9 kB)
fit2        Large lm (12 elements,  847.2 kB)
fit3        Large lm (12 elements,  895.5 kB)
fit4        Large lm (12 elements,  943.9 kB)
fit5        Large lm (12 elements,  992.3 kB)
wage        3000 obs. of 12 variables
values
  age.grid  int [1:63] 18 19 20 21 22 23 24 25 26 27 ...
  agelims   int [1:2] 18 80
  cvs       num [1:10] NA 1735 1682 1637 1632 ...
  d.min     8L
  deltas    num [1:10] 1677 1601 1598 1596 1595 ...
```

Alexa Summers
Santhoshini Sree Bolisetty
Gireesh Kumar Muppalla
CS 5565—Lab 4

9.

(a)

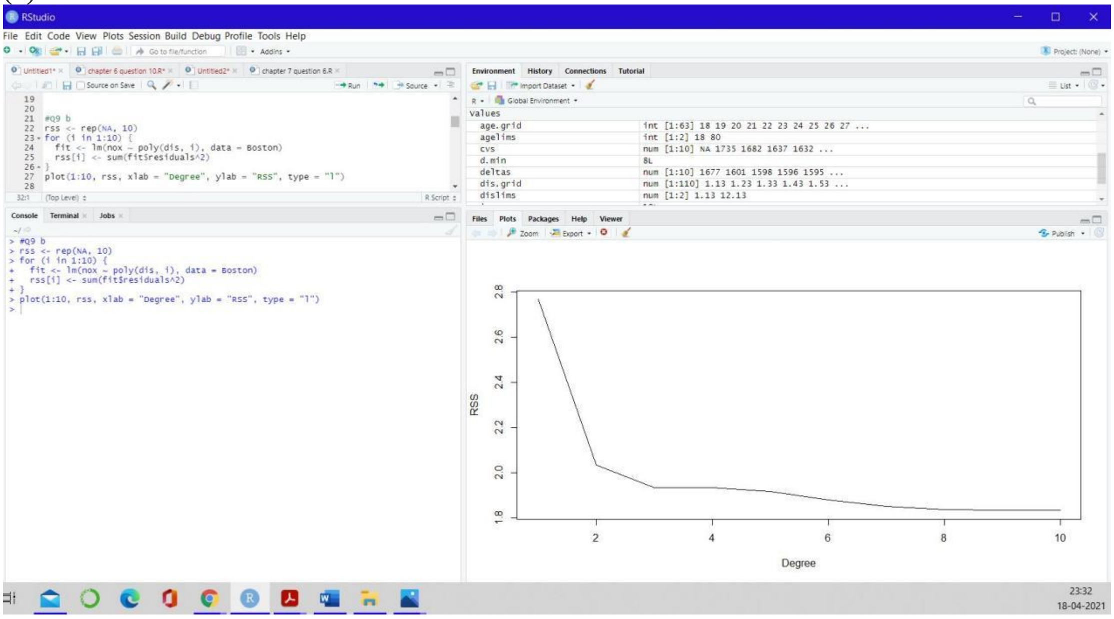


```
library(glmnet)
require(glmnet)
library(splines)
set.seed(1)
fit <- lm(nox ~ poly(dis, 3), data = Boston)
summary(fit)
dislims <- range(Boston$dis)
dis.grid <- seq(from = dislims[1], to = dislims[2], by = 0.1)
preds <- predict(fit, list(dis = dis.grid))
plot(nox ~ dis, data = Boston, col = "darkgrey")
```

```
> #Q9 a
> library(MASS)
> library(glmnet)
> require(glmnet)
> library(splines)
> set.seed(1)
> fit <- lm(nox ~ poly(dis, 3), data = Boston)
> summary(fit)

Call:
lm(formula = nox ~ poly(dis, 3), data = Boston)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.121130 -0.040619 -0.009738  0.023385  0.194904 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)    0.554695   0.002759 201.021  < 2e-16 ***
poly(dis, 3)1 -2.003096   0.062071 -32.271  < 2e-16 ***
poly(dis, 3)2  0.856330   0.062071  13.796  < 2e-16 ***
poly(dis, 3)3 -0.318049   0.062071  -5.124 4.27e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.06207 on 502 degrees of freedom
Multiple R-squared:  0.7148,	Adjusted R-squared:  0.7131 
F-statistic: 419.3 on 3 and 502 DF,  p-value: < 2.2e-16

> dislims <- range(Boston$dis)
> dis.grid <- seq(from = dislims[1], to = dislims[2], by = 0.1)
> preds <- predict(fit, list(dis = dis.grid))
> plot(nox ~ dis, data = Boston, col = "darkgrey")
> lines(dis.grid, preds, col = "red", lwd = 2)
> 
```

```
wage          3000 obs. of 12 variables
Values
  age.grid    int [1:63] 18 19 20 21 22 23 24 25 26 27 ...
  agelims     int [1:2] 18 80
  cvs         num [1:10] NA 1735 1682 1637 1632 ...
  d.min       8L
  deltas      num [1:10] 1677 1601 1598 1596 1595 ...
  dis.grid    num [1:110] 1.13 1.23 1.33 1.43 1.53 ...
  dislims     num [1:2] 1.13 12.13
  i           10L
  preds       Named num [1:110] 0.755 0.742 0.729 0.716 0.704 ...
```

(b)



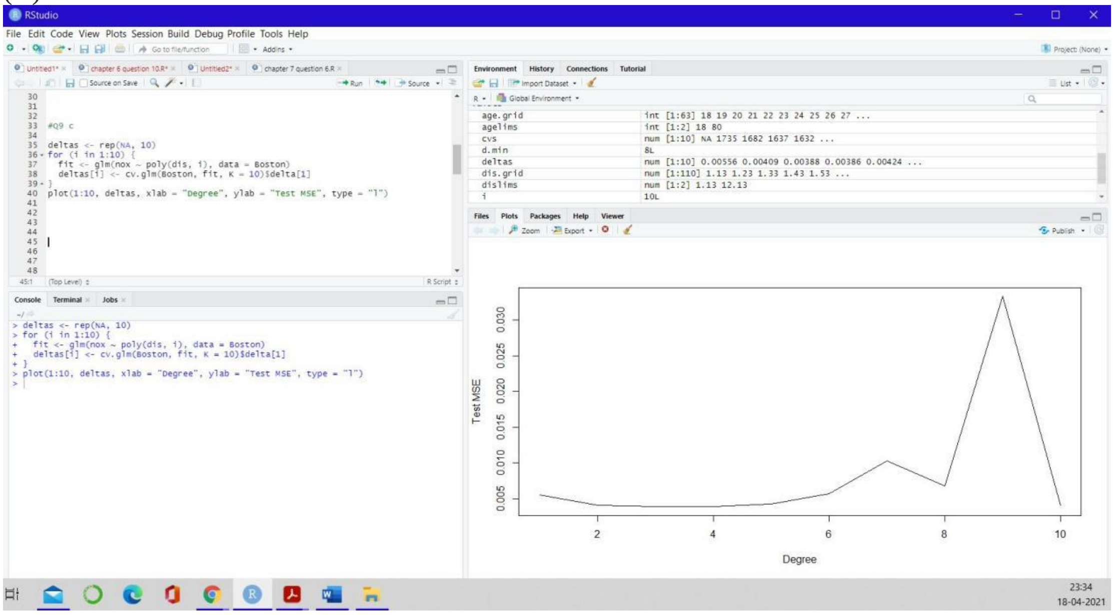
```
#Q9 b
rss <- rep(NA, 10)
for (i in 1:10) {
  fit <- lm(nox ~ poly(dis, i), data = Boston)
  rss[i] <- sum(fit$residuals^2)
}
plot(1:10, rss, xlab = "Degree", ylab = "RSS", type = "l")
```

```
> #Q9 b
> rss <- rep(NA, 10)
> for (i in 1:10) {
+   fit <- lm(nox ~ poly(dis, i), data = Boston)
+   rss[i] <- sum(fit$residuals^2)
+ }
> plot(1:10, rss, xlab = "Degree", ylab = "RSS", type = "l")
> 
```

```
Values
  age.grid    int [1:63] 18 19 20 21 22 23 24 25 26 27 ...
  agelims     int [1:2] 18 80
  cvs         num [1:10] NA 1735 1682 1637 1632 ...
  d.min       8L
  deltas      num [1:10] 1677 1601 1598 1596 1595 ...
  dis.grid    num [1:110] 1.13 1.23 1.33 1.43 1.53 ...
  dislims     num [1:2] 1.13 12.13
```

Alexa Summers
Santhoshini Sree Bolisetty
Gireesh Kumar Muppalla
CS 5565—Lab 4

(c)

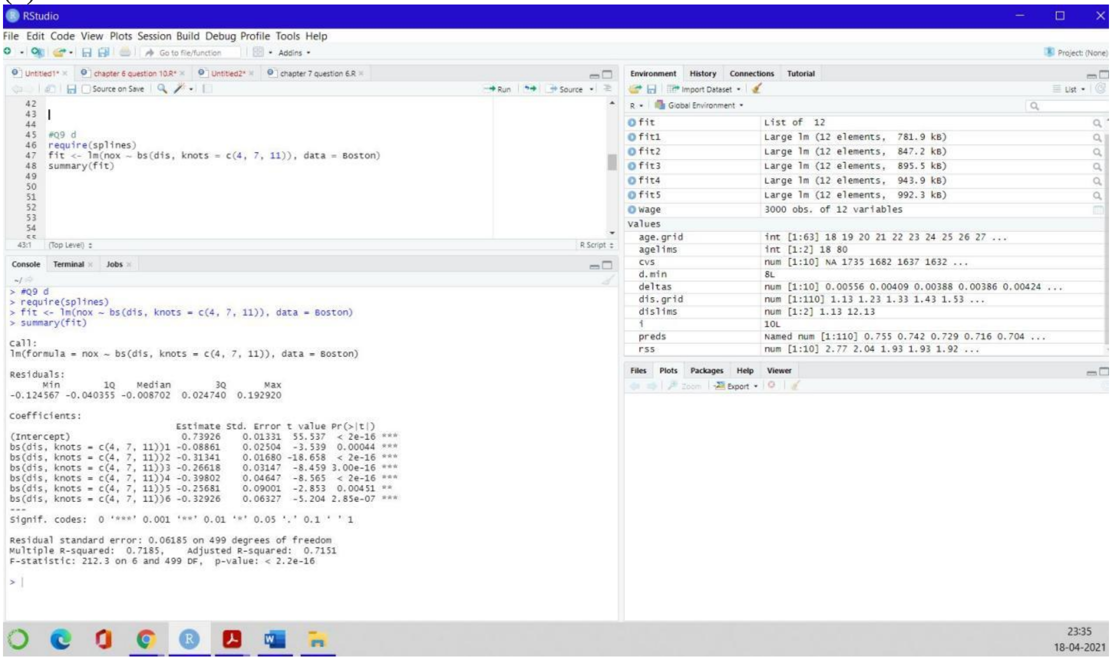


```
#Q9 c

deltas <- rep(NA, 10)
for (i in 1:10) {
  fit <- glm(nox ~ poly(dis, i), data = Boston)
  deltas[i] <- cv.glm(Boston, fit, K = 10)$delta[1]
}
plot(1:10, deltas, xlab = "Degree", ylab = "Test MSE", type = "l")
```

```
> deltas <- rep(NA, 10)
> for (i in 1:10) {
+   fit <- glm(nox ~ poly(dis, i), data = Boston)
+   deltas[i] <- cv.glm(Boston, fit, K = 10)$delta[1]
+ }
> plot(1:10, deltas, xlab = "Degree", ylab = "Test MSE", type = "l")
> 
```

```
age.grid      int [1:63] 18 19 20 21 22 23 24 25 26 27 ...
agelims       int [1:2] 18 80
cvs           num [1:10] NA 1735 1682 1637 1632 ...
d.min         8L
deltas        num [1:10] 0.00556 0.00409 0.00388 0.00386 0.00424 ...
dis.grid      num [1:110] 1.13 1.23 1.33 1.43 1.53 ...
dislims       num [1:2] 1.13 12.13
i             10L
```

(d)



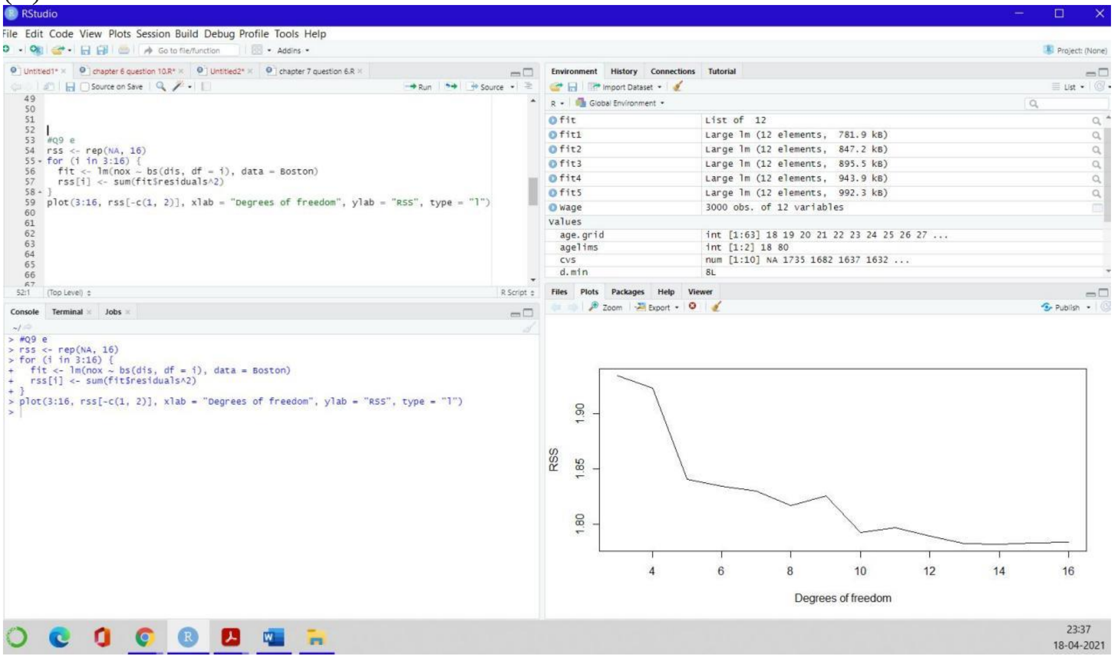
```
#Q9 d
require(splines)
fit <- lm(nox ~ bs(dis, knots = c(4, 7, 11)), data = Boston)
summary(fit)
```

```
> #Q9 d
> require(splines)
> fit <- lm(nox ~ bs(dis, knots = c(4, 7, 11)), data = Boston)
> summary(fit)

Call:
lm(formula = nox ~ bs(dis, knots = c(4, 7, 11)), data = Boston)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.124567 -0.040355 -0.008702  0.024740  0.192920 

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)    
(Intercept)                    0.73926    0.01331  55.537  < 2e-16 ***
bs(dis, knots = c(4, 7, 11))1 -0.08861    0.02504  -3.539  0.00044 ***
bs(dis, knots = c(4, 7, 11))2 -0.31341    0.01680 -18.658  < 2e-16 ***
bs(dis, knots = c(4, 7, 11))3 -0.26618    0.03147  -8.459 3.00e-16 ***
bs(dis, knots = c(4, 7, 11))4 -0.39802    0.04647  -8.565  < 2e-16 ***
bs(dis, knots = c(4, 7, 11))5 -0.25681    0.09001  -2.853  0.00451 ** 
bs(dis, knots = c(4, 7, 11))6 -0.32926    0.06327  -5.204 2.85e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.06185 on 499 degrees of freedom
Multiple R-squared:  0.7185,	Adjusted R-squared:  0.7151 
F-statistic: 212.3 on 6 and 499 DF,  p-value: < 2.2e-16

> 
```

```
fit           List of  12
fit1          Large lm (12 elements,  781.9 kB)
fit2          Large lm (12 elements,  847.2 kB)
fit3          Large lm (12 elements,  895.5 kB)
fit4          Large lm (12 elements,  943.9 kB)
fit5          Large lm (12 elements,  992.3 kB)
wage          3000 obs. of 12 variables
Values
  age.grid    int [1:63] 18 19 20 21 22 23 24 25 26 27 ...
  agelims     int [1:2] 18 80
  cvs         num [1:10] NA 1735 1682 1637 1632 ...
  d.min       8L
  deltas      num [1:10] 0.00556 0.00409 0.00388 0.00386 0.00424 ...
  dis.grid    num [1:110] 1.13 1.23 1.33 1.43 1.53 ...
  dislims     num [1:2] 1.13 12.13
  i           10L
  preds       Named num [1:110] 0.755 0.742 0.729 0.716 0.704 ...
  rss         num [1:10] 2.77 2.04 1.93 1.93 1.92 ...
```

Alexa Summers
Santhoshini Sree Bolisetty
Gireesh Kumar Muppalla
CS 5565—Lab 4

(e)

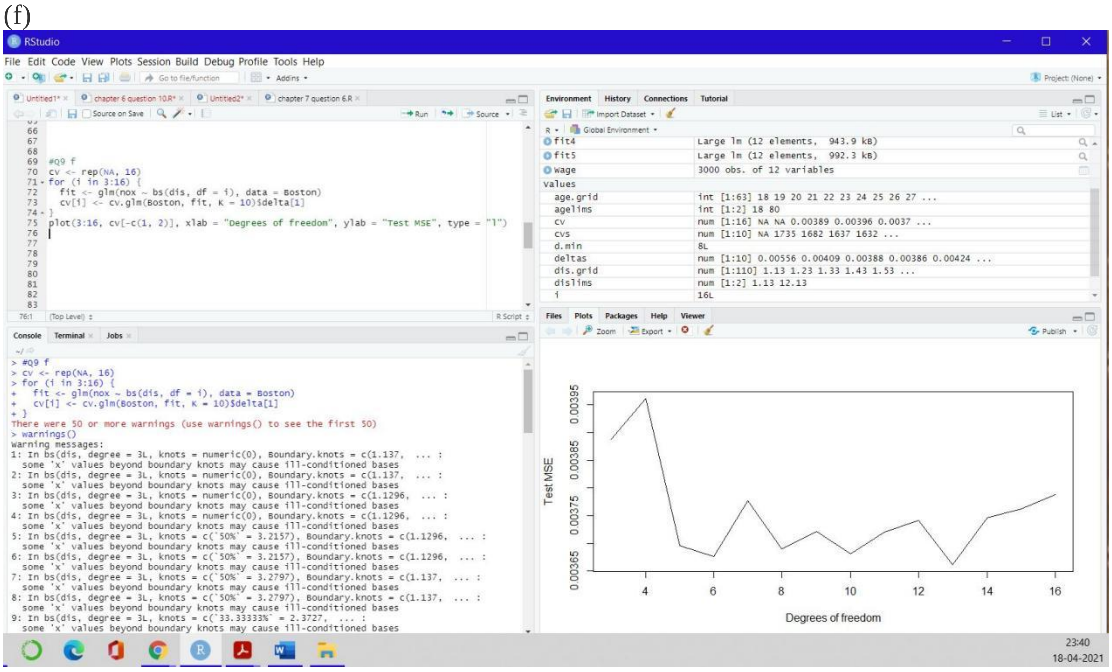


```
#Q9 e
rss <- rep(NA, 16)
for (i in 3:16) {
  fit <- lm(nox ~ bs(dis, df = i), data = Boston)
  rss[i] <- sum(fit$residuals^2)
}
plot(3:16, rss[-c(1, 2)], xlab = "Degrees of freedom", ylab = "RSS", type = "l")
```

```
> #Q9 e
> rss <- rep(NA, 16)
> for (i in 3:16) {
+   fit <- lm(nox ~ bs(dis, df = i), data = Boston)
+   rss[i] <- sum(fit$residuals^2)
+ }
> plot(3:16, rss[-c(1, 2)], xlab = "Degrees of freedom", ylab = "RSS", type = "l")
>
```

```
fit          List of  12
fit1         Large lm (12 elements,  781.9 kB)
fit2         Large lm (12 elements,  847.2 kB)
fit3         Large lm (12 elements,  895.5 kB)
fit4         Large lm (12 elements,  943.9 kB)
fit5         Large lm (12 elements,  992.3 kB)
wage         3000 obs. of 12 variables
Values
  age.grid   int [1:63] 18 19 20 21 22 23 24 25 26 27 ...
  agelims    int [1:2] 18 80
  cvs        num [1:10] NA 1735 1682 1637 1632 ...
  d.min      8L
```

(f)



```
#Q9 f
cv <- rep(NA, 16)
for (i in 3:16) {
  fit <- glm(nox ~ bs(dis, df = i), data = Boston)
  cv[i] <- cv.glm(Boston, fit, K = 10)$delta[1]
}
plot(3:16, cv[-c(1, 2)], xlab = "Degrees of freedom", ylab = "Test MSE", type = "l")
```

```
> #Q9 f
> cv <- rep(NA, 16)
> for (i in 3:16) {
+   fit <- glm(nox ~ bs(dis, df = i), data = Boston)
+   cv[i] <- cv.glm(Boston, fit, K = 10)$delta[1]
+ }
There were 50 or more warnings (use warnings() to see the first 50)
> warnings()
Warning messages:
1: In bs(dis, degree = 3L, knots = numeric(0), Boundary.knots = c(1.137,  ... :
  some 'x' values beyond boundary knots may cause ill-conditioned bases
2: In bs(dis, degree = 3L, knots = numeric(0), Boundary.knots = c(1.137,  ... :
  some 'x' values beyond boundary knots may cause ill-conditioned bases
3: In bs(dis, degree = 3L, knots = numeric(0), Boundary.knots = c(1.1296,  ... :
  some 'x' values beyond boundary knots may cause ill-conditioned bases
4: In bs(dis, degree = 3L, knots = numeric(0), Boundary.knots = c(1.1296,  ... :
  some 'x' values beyond boundary knots may cause ill-conditioned bases
5: In bs(dis, degree = 3L, knots = c(`50%` = 3.2157), Boundary.knots = c(1.1296,  ... :
  some 'x' values beyond boundary knots may cause ill-conditioned bases
6: In bs(dis, degree = 3L, knots = c(`50%` = 3.2157), Boundary.knots = c(1.1296,  ... :
  some 'x' values beyond boundary knots may cause ill-conditioned bases
7: In bs(dis, degree = 3L, knots = c(`50%` = 3.2797), Boundary.knots = c(1.137,  ... :
  some 'x' values beyond boundary knots may cause ill-conditioned bases
8: In bs(dis, degree = 3L, knots = c(`50%` = 3.2797), Boundary.knots = c(1.137,  ... :
  some 'x' values beyond boundary knots may cause ill-conditioned bases
9: In bs(dis, degree = 3L, knots = c(`33.33333%` = 2.3727,  ... :
  some 'x' values beyond boundary knots may cause ill-conditioned bases
```

```
fit4         Large lm (12 elements,  943.9 kB)
fit5         Large lm (12 elements,  992.3 kB)
wage         3000 obs. of 12 variables
Values
  age.grid   int [1:63] 18 19 20 21 22 23 24 25 26 27 ...
  agelims    int [1:2] 18 80
  cv         num [1:16] NA NA 0.00389 0.00396 0.0037 ...
  cvs        num [1:10] NA 1735 1682 1637 1632 ...
  d.min      8L
  deltas     num [1:10] 0.00556 0.00409 0.00388 0.00386 0.00424 ...
  dis.grid   num [1:110] 1.13 1.23 1.33 1.43 1.53 ...
  dislims    num [1:2] 1.13 12.13
  i          16L
```