# HDS Exercise set 3

*Shabbeer Hassan*

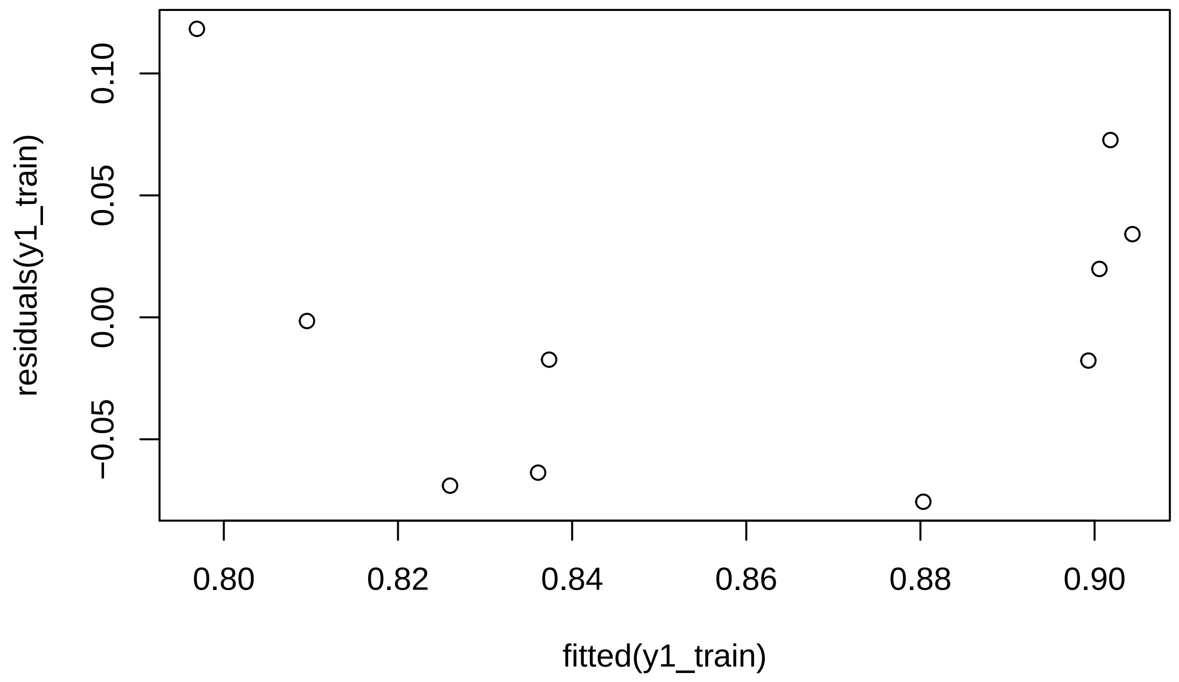
### Problem 1 – Solution

(a)

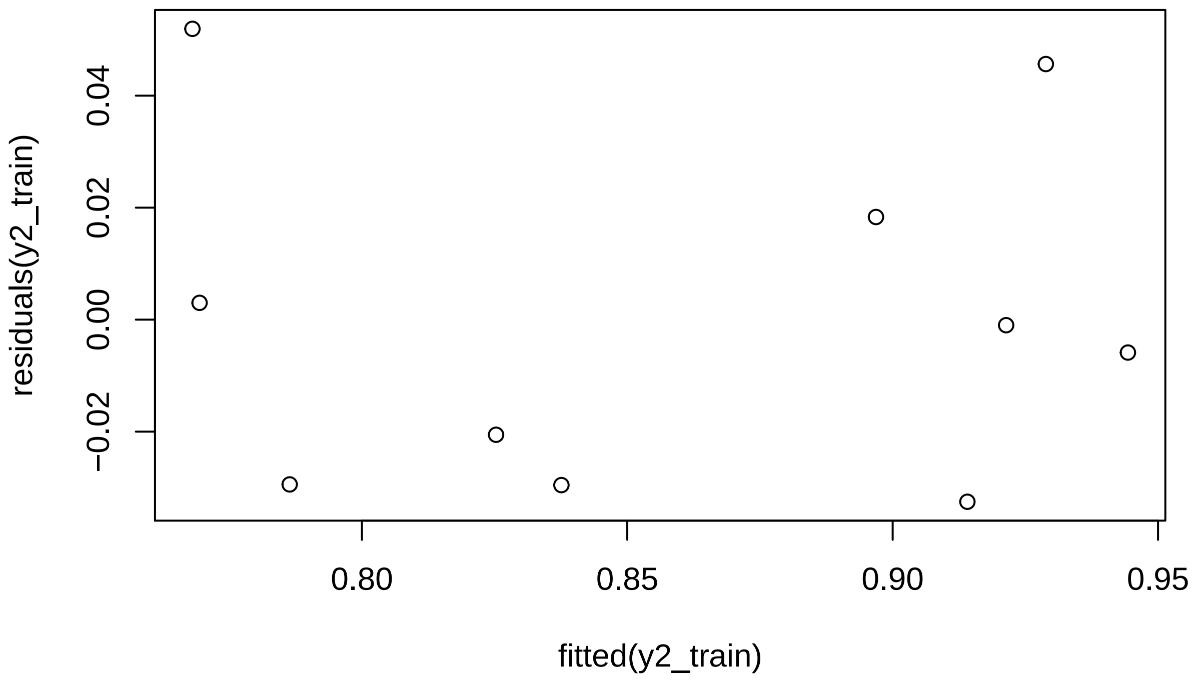
```
library(tidyverse)
```

```
## -- Attaching packages -----------------------------------------------------------------------------------

## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ---------------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

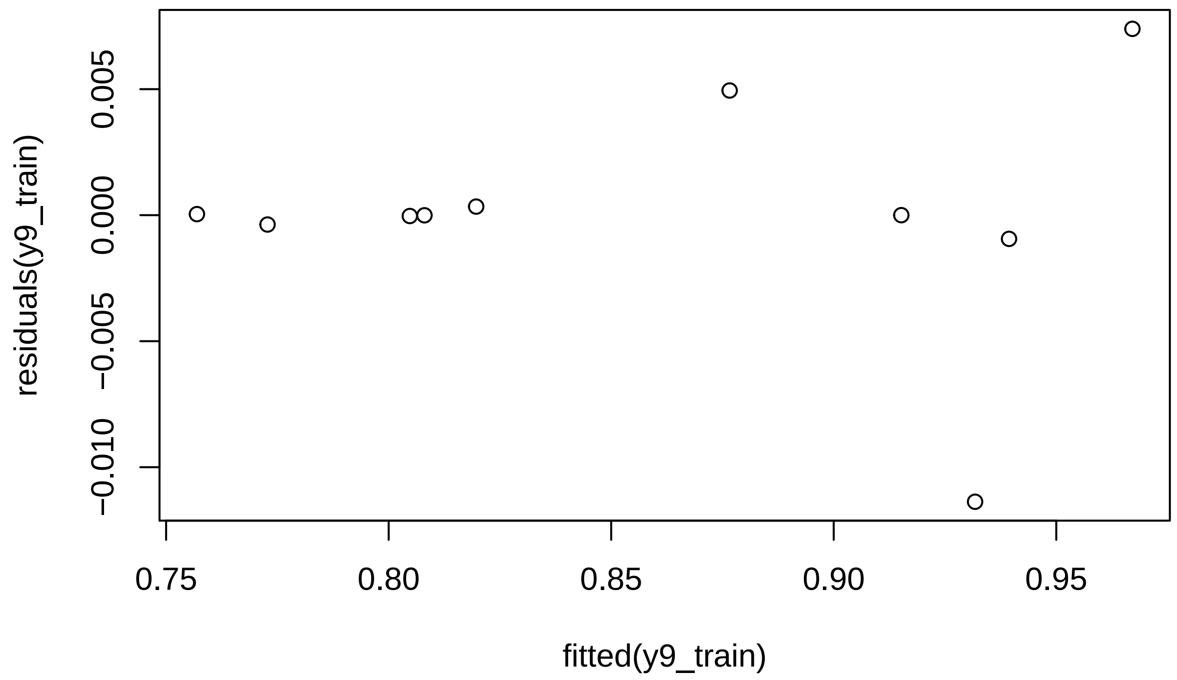
```
library(ggplot2)

# Dataset
HDS_ex3 <- read.csv("E:/Dropbox/Important_Documents/Doctoral_Work/Courses/High Dimensional Stats/2019/We

train <- HDS_ex3[ c(1:10), ]
test <- HDS_ex3[ -c(1:10), ]

### Models using TRAIN data

# 1st order LM
y1_train <- lm(y ~ x, data = train)
sm_y1_train <- summary( y1_train )
plot(fitted(y1_train), residuals(y1_train)) # Fitted vs Residuals
```



```r
# 2nd order LM
y2_train <- lm(y ~ x + poly(x, 2, raw = T), data = train)
sm_y2_train <- summary( y2_train )
plot(fitted(y2_train), residuals(y2_train)) # Fitted vs Residuals
```



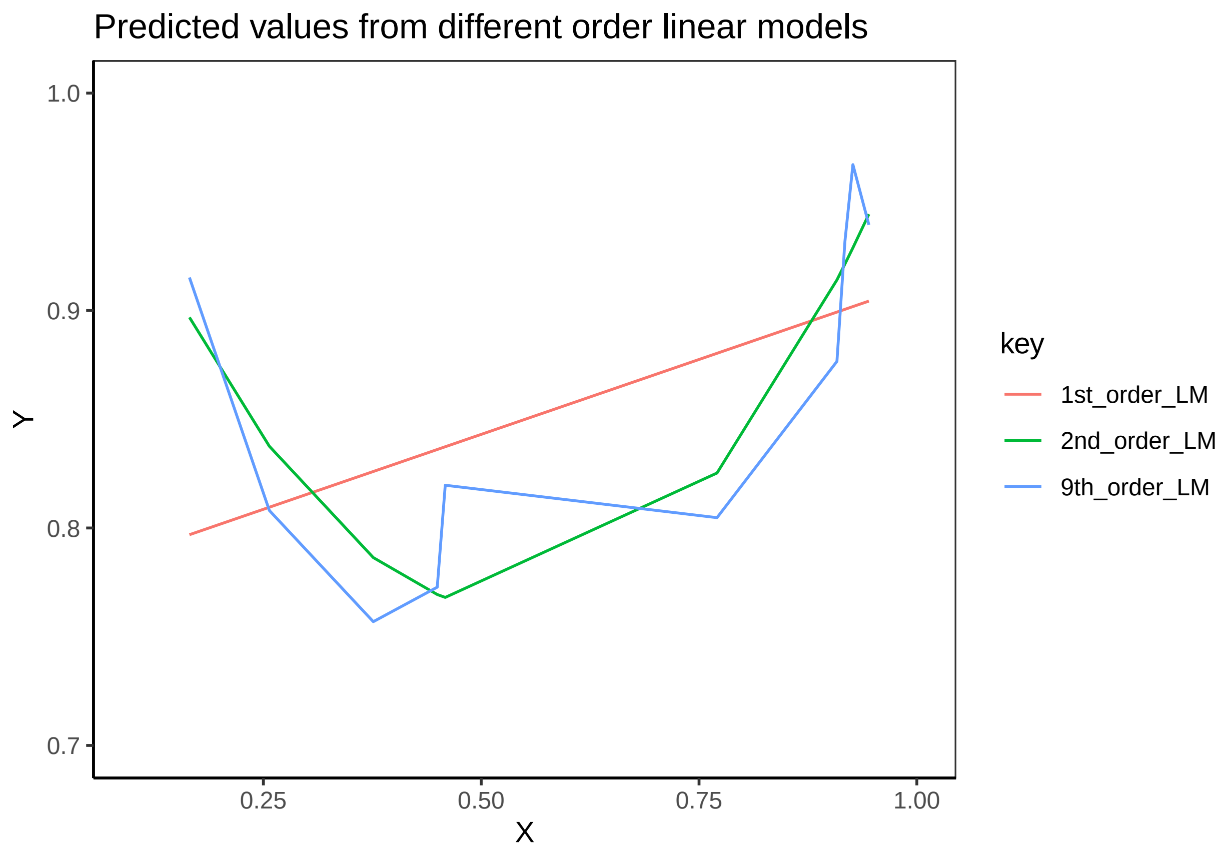
```r
# 9th order LM
y9_train <- lm(y ~ x + poly(x, 9, raw = T), data = train)
sm_y9_train <- summary( y9_train )
plot( fitted(y9_train), residuals(y9_train) ) # Fitted vs Residuals
```



(b)

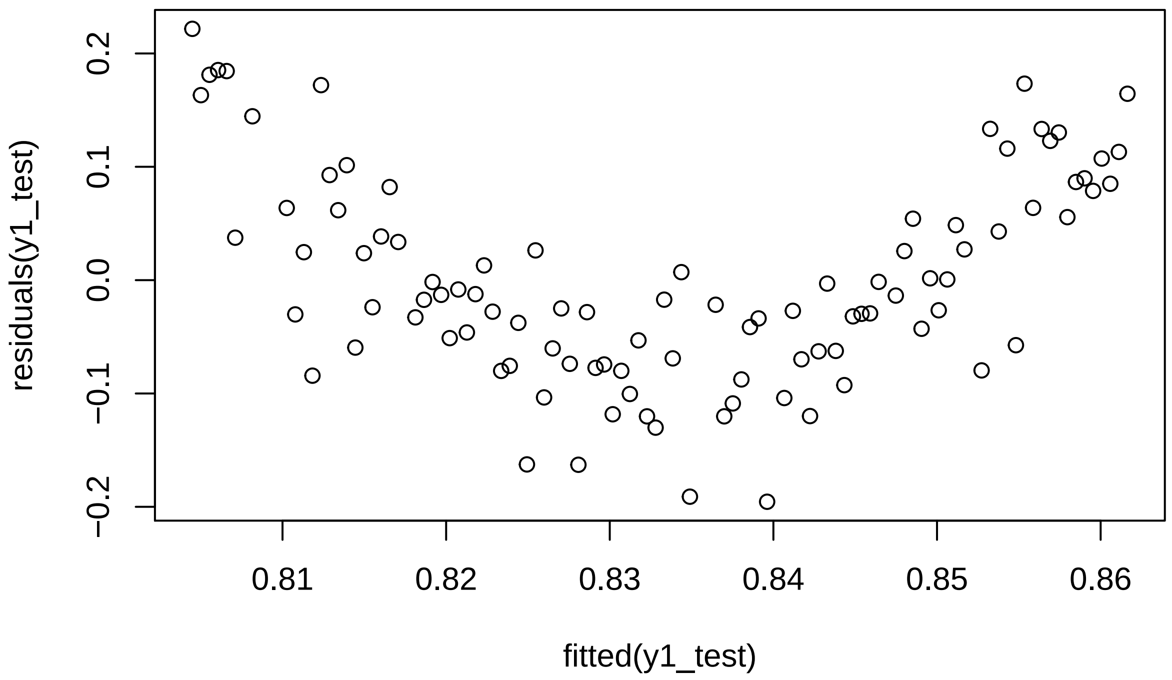
```r
##### Plot training data for all models

# Adding predicted values to a dataset
pred_train <- as.data.frame(cbind( train$x,
                                   predict( y1_train ),
                                   predict( y2_train ),
                                   predict( y9_train ) ))
colnames( pred_train ) <- c( "x", "1st_order_LM", "2nd_order_LM", "9th_order_LM")

# Convert from wide to long
pred_train.df <- reshape2::melt(pred_train,
                      id = "x")

# Plotting
pred_train %>%
    gather(key,value, "1st_order_LM", "2nd_order_LM", "9th_order_LM") %>%
    ggplot(aes(x=x, y=value, colour=key)) +
    geom_line() +
  #theme with white background
     theme_bw() +
  #eliminates background, gridlines, and chart border
     theme(
       plot.background = element_blank()
```

```
      ,panel.grid.major = element_blank()
      ,panel.grid.minor = element_blank()
      ) +
  #draws x and y axis line
     theme(axis.line = element_line(color = 'black')) +
  scale_x_continuous( "X", limits = c(0.1, 1) ) +
  scale_y_continuous( "Y", limits = c(0.7, 1)) +
  labs(title = "Predicted values from different order linear models")
```

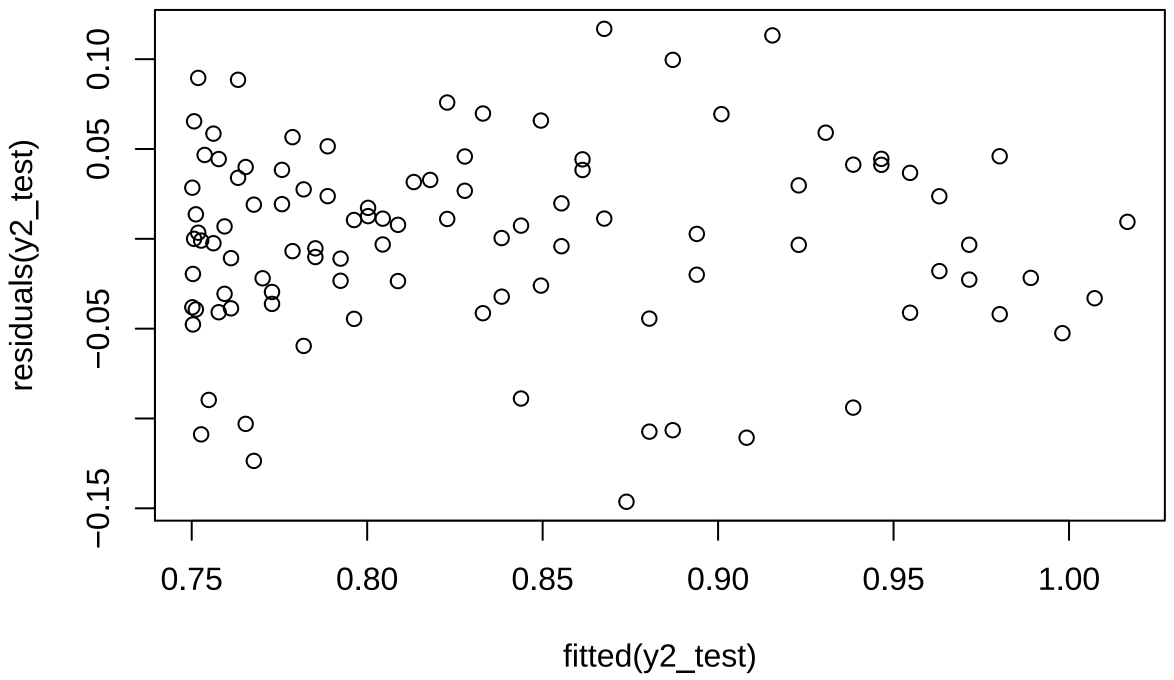


(c)

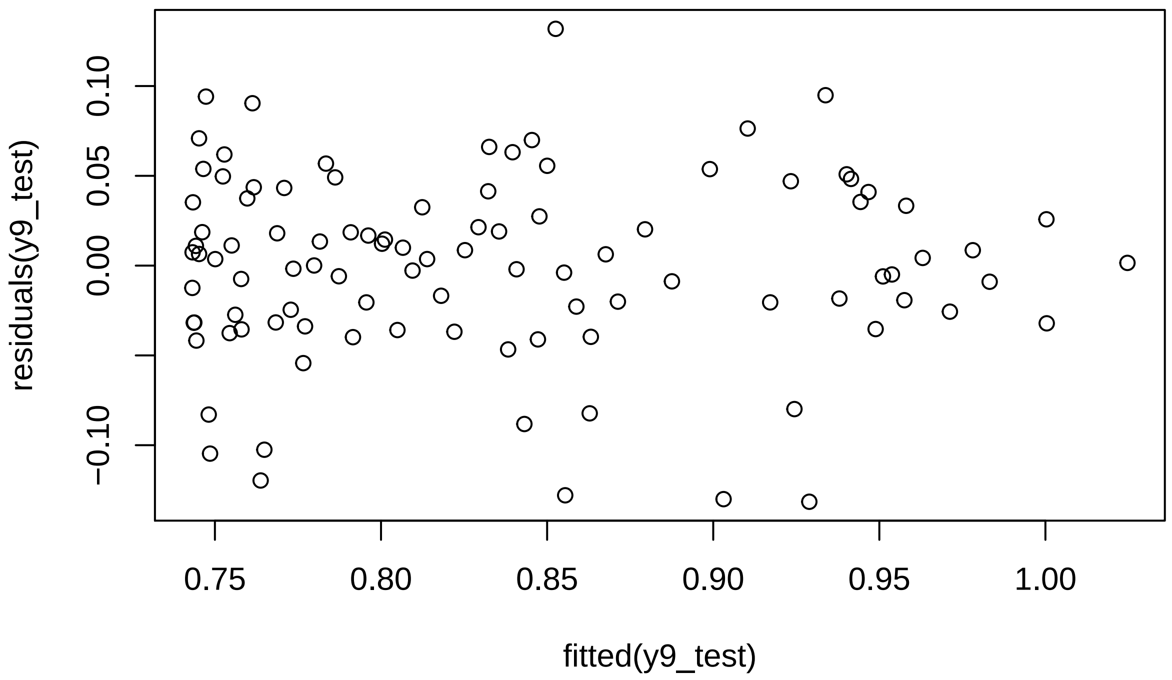
```
### Models using test

# 1st order LM
y1_test <- lm(y ~ x, data = test)
sm_y1_test <- summary( y1_test )
plot(fitted(y1_test), residuals(y1_test)) # Fitted vs Residuals
```



```r
# 2nd order LM
y2_test <- lm(y ~ x + poly(x, 2, raw = T), data = test)
sm_y2_test <- summary( y2_test )
plot(fitted(y2_test), residuals(y2_test)) # Fitted vs Residuals
```



```
# 9th order LM
y9_test <- lm(y ~ x + poly(x, 9, raw = T), data = test)
sm_y9_test <- summary( y9_test )
plot( fitted(y9_test), residuals(y9_test) ) # Fitted vs Residuals
```



```
# MSE function
mse <- function(sm)
    mean(sm$residuals^2)

#mse <- function(sm) {sum( sm$residuals^2 )/sm$df.residual}

# Obtain TEST MSE's
y1_test_mse <- mse(sm_y1_test)
y2_test_mse <- mse(sm_y2_test)
y9_test_mse <- mse(sm_y9_test)
test_mse.df <- as.data.frame(cbind( "Test_MSE", y1_test_mse, y2_test_mse, y9_test_mse ))
colnames(test_mse.df) <- c("Data", "1st_order", "2nd order", "9th_order")

# Obtain TRAIN MSE's
y1_train_mse <- mse(sm_y2_train)
y2_train_mse <- mse(sm_y2_train)
y9_train_mse <- mse(sm_y9_train)
train_mse.df <- as.data.frame(cbind( "Train_MSE", y1_train_mse, y2_train_mse, y9_train_mse ))
colnames(train_mse.df) <- c("Data", "1st_order", "2nd order", "9th_order")

# Get a df to showcase diff bet train and test MSE
diff_mse <- rbind(test_mse.df, train_mse.df)
diff_mse
```

```
##       Data           1st_order           2nd order           9th_order
## 1 Test_MSE 0.0083714171301726 0.00273715162189185 0.00256754067814762
## 2 Train_MSE 0.000838066404415258 0.000838066404415258 2.09569420646987e-05
```

**Training MSE's for the 1st, 2nd and 9th order models are higher than the test MSE's. In fact, as the order of model increases, the MSE of test dataset remains similar but the training set MSE's decrease.**

(d)

```r
### Plot predicted x-values against actual x from train data

## Prediction for each models
predict_y1 <- predict(y1_train, data.frame(x = test$x),
                      interval = 'confidence',
                      level = 0.99)

predict_y2 <- predict(y2_train, data.frame(x = test$x),
                      interval = 'confidence',
                      level = 0.99)
```

```
## Warning in predict.lm(y2_train, data.frame(x = test$x), interval =
## "confidence", : prediction from a rank-deficient fit may be misleading
```

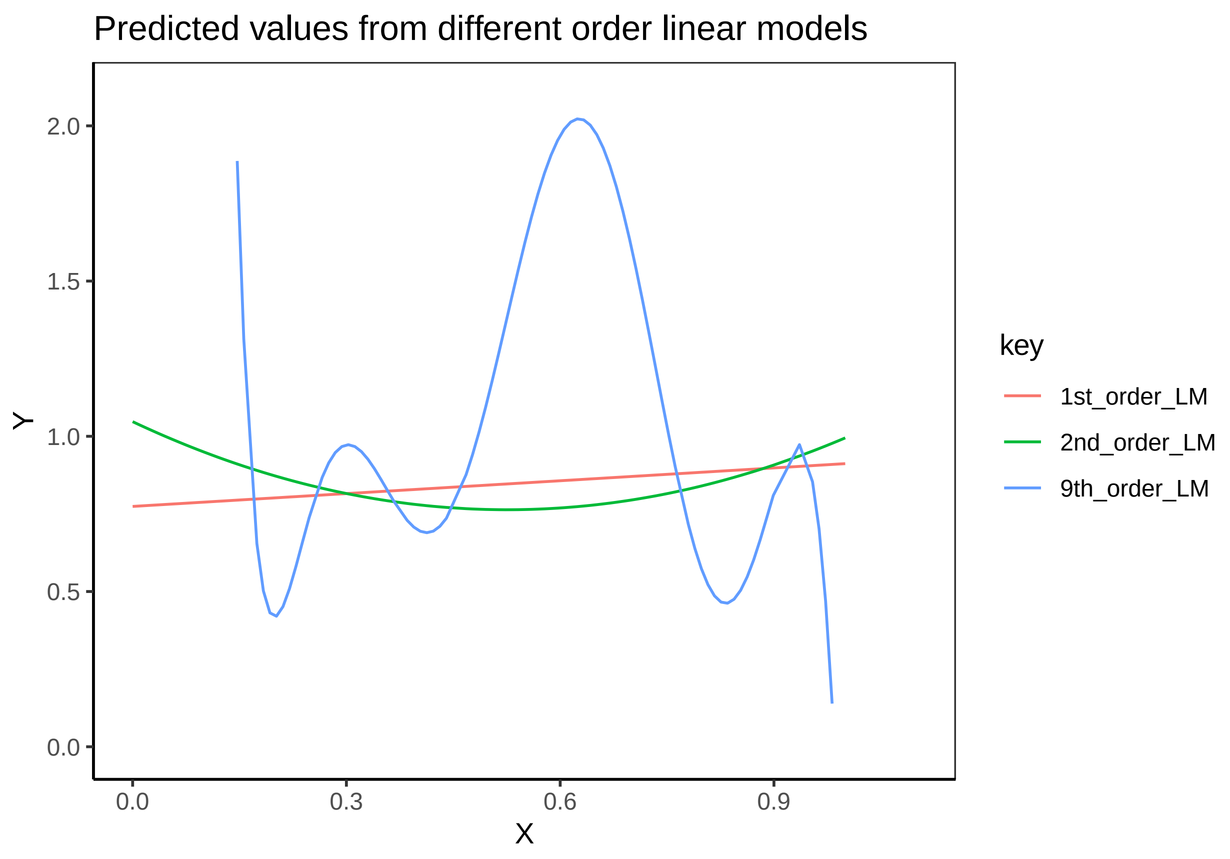
```r
predict_y9 <- predict(y9_train, data.frame(x = test$x),
                      interval = 'confidence',
                      level = 0.99)
```

```
## Warning in predict.lm(y9_train, data.frame(x = test$x), interval =
## "confidence", : prediction from a rank-deficient fit may be misleading
```

```r
# Adding all predicted values in a df
predicted_df <- as.data.frame(cbind( test$x, predict_y1[,1], predict_y2[,1], predict_y9[,1] ))
colnames(predicted_df) <- c( "x", "1st_order_LM", "2nd_order_LM", "9th_order_LM")


# Convert from wide to long
predicted_test.df <- reshape2::melt(predicted_df,
                      id = "x")

# Plotting
predicted_df %>%
     gather(key,value, "1st_order_LM", "2nd_order_LM", "9th_order_LM") %>%
    ggplot(aes(x=x, y=value, colour=key)) +
    geom_line() +
  #theme with white background
     theme_bw() +
  #eliminates background, gridlines, and chart border
     theme(
       plot.background = element_blank()
      ,panel.grid.major = element_blank()
      ,panel.grid.minor = element_blank()
      ) +
  #draws x and y axis line
```

```
    theme(axis.line = element_line(color = 'black')) +
  scale_x_continuous( "X", limits = c(0, 1.1) ) +
  scale_y_continuous( "Y", limits = c(0, 2.1)) +
  labs(title = "Predicted values from different order linear models")
```

```
## Warning: Removed 18 rows containing missing values (geom_path).
```



**The first degree model is reasonable, but we can see that the second degree model fits much better. The ninth degree model seem rather wild.**

(e).

```
# Get truth "y" into predicted dataframe
pred_truth.df <- as.data.frame( cbind(test$y, predicted_df) )
colnames(pred_truth.df) <- c("truth", "x", "1st_order_LM", "2nd_order_LM", "9th_order_LM")

## Bias-Variance tradeoff
get_bias = function(estimate, truth) {
  mean(estimate) - truth
}

get_var = function(estimate) {
  mean((estimate - mean(estimate)) ^ 2)
}
```

```r
get_mse = function(truth, estimate) {
  mean((estimate - truth) ^ 2)
}


# Bias from 3 models
bias_1st <- get_bias(pred_truth.df$`1st_order_LM`, pred_truth.df$truth)
bias_2nd <- get_bias(pred_truth.df$`2nd_order_LM`, pred_truth.df$truth)
bias_9th <- get_bias(pred_truth.df$`9th_order_LM`, pred_truth.df$truth)

bias_df <- as.data.frame( cbind(bias_1st, bias_2nd, bias_9th) )
bias <- c(mean(bias_df$bias_1st), mean(bias_df$bias_2nd), mean(bias_df$bias_9th))

# Variance from 3 models
var_1st <- get_var(pred_truth.df$`1st_order_LM`)
var_2nd <- get_var(pred_truth.df$`2nd_order_LM`)
var_9th <- get_var(pred_truth.df$`9th_order_LM`)

var <- as.data.frame( rbind(var_1st, var_2nd, var_9th) )

# MSE from 3 models
mse_1st <- get_mse(pred_truth.df$`1st_order_LM`, pred_truth.df$truth)
mse_2nd <- get_mse(pred_truth.df$`2nd_order_LM`, pred_truth.df$truth)
mse_9th <- get_mse(pred_truth.df$`9th_order_LM`, pred_truth.df$truth)

mse <- as.data.frame( rbind(mse_1st, mse_2nd, mse_9th) )

# Summarize these above results in the following table
results <- data.frame (
  cbind(poly_degree = c(1, 2, 9),
        round(bias^2, 5),
        round(mse, 5),
        round(var, 5))
)
colnames(results) = c("Degree", "Mean Squared Error", "Bias Squared", "Variance")
rownames(results) = NULL
knitr::kable(results, booktabs = TRUE, escape = TRUE, align = "c")
```

| Degree | Mean Squared Error | Bias Squared | Variance |
|:---:|:---:|:---:|:---:|
| 1 | 0.00006 | 0.01160 | 0.00158 |
| 2 | 0.00028 | 0.00305 | 0.00650 |
| 9 | 25.35862 | 270.84226 | 246.74820 |

```r
# Bias-Variance Tradeoff
# Defined as, bias ^ 2 + variance == mse
```

**We see that the Bias in general decreases upon considering the 2nd order model than 1st one**

**We see that the 2nd order model gets the best bias-variance tradeoff here**

**Problem 2 – Solution**

(a)

```r
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```r
data(Boston)

#Using all variables as predictors

lm.fit <- lm(medv ~ ., Boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ ., data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.595  -2.730  -0.518   1.777  26.199
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
## crim        -1.080e-01  3.286e-02  -3.287 0.001087 **
## zn           4.642e-02  1.373e-02   3.382 0.000778 ***
## indus        2.056e-02  6.150e-02   0.334 0.738288
## chas         2.687e+00  8.616e-01   3.118 0.001925 **
## nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
## rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
## age          6.922e-04  1.321e-02   0.052 0.958229
## dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
## rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
## tax         -1.233e-02  3.760e-03  -3.280 0.001112 **
## ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
## black        9.312e-03  2.686e-03   3.467 0.000573 ***
## lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.745 on 492 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7338
## F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

```r
AIC(lm.fit)
```

```
## [1] 3027.609
```

```r
BIC(lm.fit)
```

```
## [1] 3091.007
```

```r
# Using all but age
lm.fit1 <- lm(medv .-age ,data=Boston )
summary (lm.fit1)
```

```
##
## Call:
## lm(formula = medv ~ . - age, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.6054  -2.7313  -0.5188   1.7601  26.2243
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.436927   5.080119   7.172 2.72e-12 ***
## crim         -0.108006   0.032832  -3.290 0.001075 **
## zn            0.046334   0.013613   3.404 0.000719 ***
## indus         0.020562   0.061433   0.335 0.737989
## chas          2.689026   0.859598   3.128 0.001863 **
## nox         -17.713540   3.679308  -4.814 1.97e-06 ***
## rm            3.814394   0.408480   9.338  < 2e-16 ***
## dis          -1.478612   0.190611  -7.757 5.03e-14 ***
## rad           0.305786   0.066089   4.627 4.75e-06 ***
## tax          -0.012329   0.003755  -3.283 0.001099 **
## ptratio      -0.952211   0.130294  -7.308 1.10e-12 ***
## black         0.009321   0.002678   3.481 0.000544 ***
## lstat        -0.523852   0.047625 -10.999  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.74 on 493 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7343
## F-statistic: 117.3 on 12 and 493 DF,  p-value: < 2.2e-16
```

```r
AIC(lm.fit1)
```

```
## [1] 3025.611
```

```r
BIC(lm.fit1)
```

```
## [1] 3084.783
```

```r
# Using all but rm
lm.fit2 <- lm(medv .-rm ,data=Boston )
summary (lm.fit2)
```

```
##
## Call:
## lm(formula = medv ~ . - rm, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.8415  -2.9471  -0.5922   1.7921  23.1236
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  69.467163   3.884501  17.883  < 2e-16 ***
## crim         -0.115899   0.035484  -3.266 0.001166 **
## zn            0.065917   0.014645   4.501 8.45e-06 ***
## indus        -0.030978   0.066138  -0.468 0.639713
## chas          2.931577   0.930110   3.152 0.001721 **
## nox         -21.021201   4.107510  -5.118 4.44e-07 ***
## age           0.025592   0.013959   1.833 0.067351 .
## dis          -1.718414   0.213494  -8.049 6.29e-15 ***
## rad           0.401657   0.070757   5.677 2.35e-08 ***
## tax          -0.014881   0.004050  -3.674 0.000265 ***
## ptratio      -1.142943   0.139493  -8.194 2.20e-15 ***
## black         0.006747   0.002885   2.339 0.019752 *
## lstat        -0.772070   0.046280 -16.683  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.125 on 493 degrees of freedom
## Multiple R-squared:  0.6968, Adjusted R-squared:  0.6895
## F-statistic: 94.43 on 12 and 493 DF,  p-value: < 2.2e-16
```

```r
AIC(lm.fit2)
```

```
## [1] 3104.581
```

```r
BIC(lm.fit2)
```

```
## [1] 3163.753
```

```r
### Comparing AIC & BIC
AIC(lm.fit, lm.fit1, lm.fit2)
```

```
##         df      AIC
## lm.fit  15 3027.609
## lm.fit1 14 3025.611
## lm.fit2 14 3104.581
```

```r
BIC(lm.fit, lm.fit1, lm.fit2)
```

```
##         df      BIC
## lm.fit  15 3091.007
## lm.fit1 14 3084.783
## lm.fit2 14 3163.753
```

**Both AIC and BIC are minimized in the second model with just the age removed.**

(b)

```r
# Get log-lk values from AIC
loglk_aic_full <- -(AIC(lm.fit) - 2*15)
loglk_aic_reduced <- -(AIC(lm.fit1) - 2*14)

loglk_aic_full
```

```
## [1] -2997.609
```

```r
loglk_aic_reduced
```

```
## [1] -2997.611
```

```r
# The Likelihood ratio should be converted to -2*difference in log likelihoods: -2ln(Likelihood_reduced
# And the above can be written as:
# -2ln(Likelihood_reduced)- -2ln(Likelihood_full)
# LRT then should be approximately Chi-squared distributed with df equal to the number of fixed dimensi

# Likelihood-Ratio test (frequentist)
Deviance <- loglk_aic_reduced - loglk_aic_full
Deviance
```

```
## [1] -0.002824145
```

```r
Chisq.crit <- qchisq(0.95,1)
Chisq.crit
```

```
## [1] 3.841459
```

```r
# LRT
Deviance >= Chisq.crit   # perform the LRT
```

```
## [1] FALSE
```

```r
# p-value
1-pchisq(Deviance,1)
```

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

**The p-values for LET calculation based on AIC agrees with the one from full model for variable age (0.95 vs 1)**

(c)

```
# Get log-lk values from BIC
loglk_bic_full <- -(BIC(lm.fit) - 2*15)
loglk_bic_reduced <- -(BIC(lm.fit2) - 2*14)

loglk_bic_full
```

```
## [1] -3061.007
```

```
loglk_bic_reduced
```

```
## [1] -3135.753
```

```
# The Likelihood ratio should be converted to -2*difference in log likelihoods: -2ln(Likelihood_reduced
# And the above can be written as:
# -2ln(Likelihood_reduced)- -2ln(Likelihood_full)
# LRT then should be approximately Chi-squared distributed with df equal to the number of fixed dimensi

# Likelihood-Ratio test (frequentist)
Deviance <- loglk_bic_reduced - loglk_bic_full
Deviance
```

```
## [1] -74.746
```

```
Chisq.crit <- qchisq(0.95,1)
Chisq.crit
```

```
## [1] 3.841459
```

```
# LRT
Deviance >= Chisq.crit   # perform the LRT
```

```
## [1] FALSE
```

```
# p-value
1-pchisq(Deviance,1)
```

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

## Problem 3 – Solution

(a)

```
library(MASS)
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```
data(Boston)

tr.ind = 1:350

# Train & Test dataset
train <- Boston[tr.ind, ]
test <- Boston[-tr.ind, ]

# Using all variables as predictors
train_lm.fit <- lm(medv ~ ., train)
test_lm.fit <- lm(medv ~ ., test)

# MSE for train and test data based models
mse_train <- sum( train_lm.fit$residuals^2 )/train_lm.fit$df.residual
mse_train
```

```
## [1] 9.361612
```

```
mse_test <- sum( test_lm.fit$residuals^2 )/test_lm.fit$df.residual
mse_test
```

```
## [1] 23.09557
```

**The test data model has a higher RMSE than the train dataset. This could be an indication that your model is overfitting**

(b)

```
# 10-fold cross-validation (CV) within training data
modelcv <- train(
  medv ~ .,
  data = train,
  method = "lm",
  trControl = trainControl(
    method = "cv", number = 10
  )
)

RMSE_Modelcv <- modelcv$results$RMSE
RMSE_Modelcv
```

```
## [1] 3.132821
```

```r
pcv <- predict(modelcv, test)
errorcv <- (pcv- test$medv)
RMSE_NewDatacv <- sqrt(mean(errorcv^2))
RMSE_NewDatacv
```

```
## [1] 23.36543
```

**THe cross validation didnt work that well at all. It could still be the case of overfitting because in K-fold cross validation, the dataset is divided into k separate parts. The training process is repeated k times. Each time, one part is used as validation data, and the rest is used for training a model. ANd in this, we arent randomly doing that which increases overfitting error**

(c)

```r
tr.ind <- read.csv("E:/Dropbox/Important_Documents/Doctoral_Work/Courses/High Dimensional Stats/2019/We

# Train & Test dataset

id <- which(rownames(Boston) %in% tr.ind$V1 )
train <- Boston[id, ]
test <- Boston[-id, ]

# Using all variables as predictors
train_lm.fit <- lm(medv ~ ., train)
```

```
## Error in lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...): 0 (non-NA) cases
```

```r
test_lm.fit <- lm(medv ~ ., test)
```

```
## Error in lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...): 0 (non-NA) cases
```

```r
# MSE for train and test data based models
mse_train <- sum( train_lm.fit$residuals^2 )/train_lm.fit$df.residual
mse_train
```

```
## [1] 9.361612
```

```r
mse_test <- sum( test_lm.fit$residuals^2 )/test_lm.fit$df.residual
mse_test
```

```
## [1] 23.09557
```

```r
# 10-fold cross-validation (CV) within training data
modelcv <- train(
  medv ~ .,
  data = train,
  method = "lm",
  trControl = trainControl(
    method = "cv", number = 10
  )
)
```

```
## Error: Every row has at least one missing value were found
```

```
RMSE_Modelcv <- modelcv$results$RMSE
RMSE_Modelcv
```

```
## [1] 3.132821
```

```
pcv <- predict(modelcv, test)
errorcv <- (pcv- test$medv)
RMSE_NewDatacv <- sqrt(mean(errorcv^2))
RMSE_NewDatacv
```

```
## [1] NaN
```

## We see that randomly splitting did massively improve th CV based rmse because it remóved underlying bias associated in separating our datasets non-randomly, thus restring the assumption of iid for inference.

**Problem 4 – Solution**

(a)

```
tr.ind <- fread("E:/Dropbox/Important_Documents/Doctoral_Work/Courses/High Dimensional Stats/2019/Week 
```

```
## Error in fread("E:/Dropbox/Important_Documents/Doctoral_Work/Courses/High Dimensional Stats/2019/Wee
```

```
# Train & Test dataset
train <- Boston[tr.ind$V1, ]

# Create null model
glm.null <- glm(medv ~ 1, data = train)
```

```
## Warning in glm.fit(x = structure(numeric(0), .Dim = 0:1, .Dimnames = list(:
## no observations informative at iteration 1
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Error in glm.fit(x = structure(numeric(0), .Dim = 0:1, .Dimnames = list(: object 'fit' not found
```

```r
# Using all variables as predictors
train_glm.fit <- glm(medv ~ ., family = gaussian, train)
```

```
## Warning in glm.fit(x = structure(numeric(0), .Dim = c(0L, 14L), .Dimnames =
## list(: no observations informative at iteration 1

## Warning in glm.fit(x = structure(numeric(0), .Dim = c(0L, 14L), .Dimnames =
## list(: glm.fit: algorithm did not converge

## Error in glm.fit(x = structure(numeric(0), .Dim = c(0L, 14L), .Dimnames = list(: object 'fit' not fo
```

```r
# AIC-based forward selection
model.aic.forward <- step(glm.null, direction = "forward", k = 2, trace = FALSE,
                          scope = list(lower=glm.null, upper=train_glm.fit))
```

```
## Error in terms(object): object 'glm.null' not found
```

```r
summary(model.aic.forward)
```

```
## Error in summary(model.aic.forward): object 'model.aic.forward' not found
```

```r
# BIC-based forward selection
model.bic.forward <- step(glm.null, direction = "forward", k = log(nrow(train)), trace = FALSE,
                          scope = list(lower=glm.null, upper=train_glm.fit))
```

```
## Error in terms(object): object 'glm.null' not found
```

```r
summary(model.bic.forward)
```

```
## Error in summary(model.bic.forward): object 'model.bic.forward' not found
```

(b)

```r
# AIC-based backward selection
model.aic.backward <- step(train_glm.fit,  data = train, direction = "backward", k = 2, trace = FALSE)
```

```
## Error in terms(object): object 'train_glm.fit' not found
```

```r
summary(model.aic.backward)
```

```
## Error in summary(model.aic.backward): object 'model.aic.backward' not found
```

```r
# BIC-based backward selection
model.bic.backward <- step(train_glm.fit, direction = "backward", k = log(nrow(train)), trace = FALSE)
```

```
## Error in terms(object): object 'train_glm.fit' not found
```

```r
summary(model.bic.backward)
```

```
## Error in summary(model.bic.backward): object 'model.bic.backward' not found
```

(c)

```r
# Use AIC for interaction terms in the model
# Limiting to 2nd order interactions only

# AIC-based forward selection
model.aic.interaction.forward <- step(train_glm.fit, direction = "forward", k = 2, trace = FALSE,
                                      scope = . ~ .^2)
```

```
## Error in terms(object): object 'train_glm.fit' not found
```

```r
summary(model.aic.interaction.forward)
```

```
## Error in summary(model.aic.interaction.forward): object 'model.aic.interaction.forward' not found
```

```r
# BIC-based forward selection
model.bic.interaction.forward <- step(train_glm.fit, direction = "forward", k = log(nrow(train)),
                                      trace = FALSE, scope = . ~ .^2)
```

```
## Error in terms(object): object 'train_glm.fit' not found
```

```r
summary(model.bic.interaction.forward)
```

```
## Error in summary(model.bic.interaction.forward): object 'model.bic.interaction.forward' not found
```

(d)

```r
##################### Getting predictors + MSE from previous models

# Use glm on test data
test_glm.fit <- glm(medv ~ ., family = gaussian, test)
```

```
## Warning in glm.fit(x = structure(numeric(0), .Dim = c(0L, 14L), .Dimnames =
## list(: no observations informative at iteration 1

## Warning: glm.fit: algorithm did not converge

## Error in glm.fit(x = structure(numeric(0), .Dim = c(0L, 14L), .Dimnames = list(: object 'fit' not fo
```

```r
##### AIC_forward model
AIC_forward <- attr(model.aic.forward$terms , "term.labels")
```

```
## Error in eval(expr, envir, enclos): object 'model.aic.forward' not found
```

```r
AIC_forward
```

```
## Error in eval(expr, envir, enclos): object 'AIC_forward' not found
```

```r
# Train
model.aic.forward.train <- step(glm.null, data = train, direction = "forward", k = 2, trace = FALSE,
                                scope = list(lower=glm.null, upper=train_glm.fit))
```

```
## Error in terms(object): object 'glm.null' not found
```

```r
AIC_forward_mse_train <- sum( model.aic.forward.train$residuals^2 )/model.aic.forward.train$df.residual
```

```
## Error in eval(expr, envir, enclos): object 'model.aic.forward.train' not found
```

```r
# Test
model.aic.forward.test <- step(glm.null, data = test, direction = "forward", k = 2, trace = FALSE,
                               scope = list(lower=glm.null, upper=test_glm.fit))
```

```
## Error in terms(object): object 'glm.null' not found
```

```r
AIC_forward_mse_test <- sum( model.aic.forward.test$residuals^2 )/model.aic.forward.test$df.residual
```

```
## Error in eval(expr, envir, enclos): object 'model.aic.forward.test' not found
```

```r
##### BIC_forward model
BIC_forward <- attr(model.bic.forward$terms , "term.labels")
```

```
## Error in eval(expr, envir, enclos): object 'model.bic.forward' not found
```

```r
BIC_forward
```

```
## Error in eval(expr, envir, enclos): object 'BIC_forward' not found
```

```r
# Train
model.bic.forward.train <- step(glm.null, data = train, direction = "forward", k = log(nrow(train)), tra
                                scope = list(lower=glm.null, upper=train_glm.fit))
```

```
## Error in terms(object): object 'glm.null' not found
```

```r
BIC_forward_mse_train <- sum( model.bic.forward.train$residuals^2 )/model.bic.forward.train$df.residual
```

```
## Error in eval(expr, envir, enclos): object 'model.bic.forward.train' not found
```

```r
# Test
model.bic.forward.test <- step(glm.null, data = test, direction = "forward", k = log(nrow(test)), trace
                               scope = list(lower=glm.null, upper=test_glm.fit))
```

```
## Error in terms(object): object 'glm.null' not found
```

```
BIC_forward_mse_test <- sum( model.bic.forward.test$residuals^2 )/model.bic.forward.test$df.residual
```

```
## Error in eval(expr, envir, enclos): object 'model.bic.forward.test' not found
```

```
##### AIC_backward model
AIC_backward <- attr(model.aic.backward$terms , "term.labels")
```

```
## Error in eval(expr, envir, enclos): object 'model.aic.backward' not found
```

```
AIC_backward
```

```
## Error in eval(expr, envir, enclos): object 'AIC_backward' not found
```

```
# Train
model.aic.backward.train <- step(train_glm.fit,  data = train, direction = "backward", k = 2, trace = F
```

```
## Error in terms(object): object 'train_glm.fit' not found
```

```
AIC_backward_mse_train <- sum( model.aic.backward.train$residuals^2 )/model.aic.backward.train$df.resid
```

```
## Error in eval(expr, envir, enclos): object 'model.aic.backward.train' not found
```

```
# Test
model.aic.backward.test <- step(test_glm.fit,  data = test, direction = "backward", k = 2, trace = FALS
```

```
## Error in terms(object): object 'test_glm.fit' not found
```

```
AIC_backward_mse_test <- sum( model.aic.backward.test$residuals^2 )/model.aic.backward.test$df.residual
```

```
## Error in eval(expr, envir, enclos): object 'model.aic.backward.test' not found
```

```
##### BIC_backward model
BIC_backward <- attr(model.bic.backward$terms , "term.labels")
```

```
## Error in eval(expr, envir, enclos): object 'model.bic.backward' not found
```

```
BIC_backward
```

```
## Error in eval(expr, envir, enclos): object 'BIC_backward' not found
```

```
# Train
model.bic.backward.train <- step(train_glm.fit, data = train, direction = "backward", k = log(nrow(trai
```

```
## Error in terms(object): object 'train_glm.fit' not found
```

```
BIC_backward_mse_train <- sum( model.bic.backward.train$residuals^2 )/model.bic.backward.train$df.resid
```

```
## Error in eval(expr, envir, enclos): object 'model.bic.backward.train' not found
```

```
# Test
model.bic.backward.test <- step(test_glm.fit, data = test, direction = "backward", k = log(nrow(test)),
```

```
## Error in terms(object): object 'test_glm.fit' not found
```

```
BIC_backward_mse_test <- sum( model.bic.backward.test$residuals^2 )/model.bic.backward.test$df.residual
```

```
## Error in eval(expr, envir, enclos): object 'model.bic.backward.test' not found
```

```
##### AIC-based forward selection + interaction
AIC_forward_interaction <- attr(model.aic.interaction.forward$terms , "term.labels")
```

```
## Error in eval(expr, envir, enclos): object 'model.aic.interaction.forward' not found
```

```
AIC_forward_interaction
```

```
## Error in eval(expr, envir, enclos): object 'AIC_forward_interaction' not found
```

```
# Train
model.aic.interaction.forward.train <- step(train_glm.fit,  data = train, direction = "forward", k = 2,
                                     scope = . ~ .^2)
```

```
## Error in terms(object): object 'train_glm.fit' not found
```

```
AIC_forward_interaction_mse_train <- sum( model.aic.interaction.forward.train$residuals^2 )/model.aic.i
```

```
## Error in eval(expr, envir, enclos): object 'model.aic.interaction.forward.train' not found
```

```
# Test
model.aic.interaction.forward.test <- step(test_glm.fit, data = test, direction = "forward", k = 2, tra
                                     scope = . ~ .^2)
```

```
## Error in terms(object): object 'test_glm.fit' not found
```

```
AIC_forward_interaction_mse_test <- sum( model.aic.interaction.forward.test$residuals^2 )/model.aic.int
```

```
## Error in eval(expr, envir, enclos): object 'model.aic.interaction.forward.test' not found
```

```
##### BIC-based forward selection + interaction
BIC_forward_interaction <- attr(model.bic.interaction.forward$terms , "term.labels")
```

```
## Error in eval(expr, envir, enclos): object 'model.bic.interaction.forward' not found
```

```
BIC_forward_interaction
```

```
## Error in eval(expr, envir, enclos): object 'BIC_forward_interaction' not found
```

```
# Train
model.bic.interaction.forward.train <- step(train_glm.fit, direction = "forward", k = log(nrow(train)),
                                     trace = FALSE, scope = . ~ .^2)
```

```
## Error in terms(object): object 'train_glm.fit' not found
```

```
BIC_forward_interaction_mse_train <- sum( model.bic.interaction.forward$residuals^2 )/model.bic.interac
```

```
## Error in eval(expr, envir, enclos): object 'model.bic.interaction.forward' not found
```

```
# Test
model.bic.interaction.forward.test <- step(train_glm.fit, direction = "forward", k = log(nrow(test)), t
                                     scope = . ~ .^2)
```

```
## Error in terms(object): object 'train_glm.fit' not found
```

```
BIC_forward_interaction_mse_test <- sum( model.bic.interaction.forward.test$residuals^2 )/model.bic.int
```

```
## Error in eval(expr, envir, enclos): object 'model.bic.interaction.forward.test' not found
```

```
### Get results from above together

results <- data.frame(rbind(
                  c(AIC_forward_mse_train, AIC_forward_mse_test),
                  c(BIC_forward_mse_train, BIC_forward_mse_test),
                  c(AIC_backward_mse_train, AIC_backward_mse_test),
                  c(BIC_backward_mse_train, BIC_backward_mse_test),
                  c(AIC_forward_interaction_mse_train, AIC_forward_interaction_mse_test),
                  c(BIC_forward_interaction_mse_train, BIC_forward_interaction_mse_test)))
```

```
## Error in rbind(c(AIC_forward_mse_train, AIC_forward_mse_test), c(BIC_forward_mse_train, : object 'AI
```

```
colnames(results) <- c("Training_MSE", "Test_MSE")
results <- results %>% mutate(Model = c("AIC_forward", "BIC_forward", "AIC_backward",
                                        "BIC_backward", "AIC_forward_interaction", "BIC_forward_interac
```

```
## Error: Columns 3, 4 must be named.
## Use .name_repair to specify repair.
```

```
results
```

```
##   Training_MSE Test_MSE        NA        NA
## 1            1  0.00006   0.01160   0.00158
## 2            2  0.00028   0.00305   0.00650
## 3            9 25.35862 270.84226 246.74820
```

**The results suggest that stepwise AIC-forward regression model with 2nd order interaction terms has the lowest RMSE indictaing it to be the better model. The variance not explained is given by MSE/Variance(Y) and hence 1-Variance Unexplained will give us the explained variance**

```
## Variance explained
(1 - AIC_forward_interaction_mse_test/var(test$medv)) * 100
```

```
## Error in eval(expr, envir, enclos): object 'AIC_forward_interaction_mse_test' not found
```

### Problem 5 – Solution

(a)

```
# Dataset
HDS_ex3 <- fread("E:/Dropbox/Important_Documents/Doctoral_Work/Courses/High Dimensional Stats/2019/Week
```

```
## Error in fread("E:/Dropbox/Important_Documents/Doctoral_Work/Courses/High Dimensional Stats/2019/Weel
```

```
# Null Model
glm.null <- glm(z ~ 1, data = HDS_ex3)

# Log Reg
glm.fit <- glm(z ~ x + I(x^2), family = "binomial", data = HDS_ex3)

# Forward stepwise selection with AIC
model.aic.forward <- step(glm.null, data = HDS_ex3, direction = "forward", k = 2, trace = TRUE,
                          scope = list(lower=glm.null, upper=glm.fit))
```

```
## Start:  AIC=162.36
## z ~ 1
##
##          Df Deviance    AIC
## <none>        27.173 162.36
## + x       1   27.099 164.06
## + I(x^2)  1   27.118 164.13
```

```
summary(model.aic.forward)
```

```
##
## Call:
## glm(formula = z ~ 1, data = HDS_ex3)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.5545  -0.5545   0.4455   0.4455   0.4455
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.55455    0.04761   11.65   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.2492911)
##
##     Null deviance: 27.173  on 109  degrees of freedom
## Residual deviance: 27.173  on 109  degrees of freedom
## AIC: 162.36
##
## Number of Fisher Scoring iterations: 2
```

```r
### THe intercept model is the one chosen by the stepwise method
new.model <- glm(z ~ 1 + x + I(x^2), family = "binomial", data = HDS_ex3)
AIC(new.model)
```

```
## [1] 139.7259
```

**We may not necessarily always get the highest variance explained and lower AIC because we only compare a subset of possible models and might miss the one with the highest adjR2/lowest AIC which would include all the variables, like the above case. Given a set of predictors, there is no guarantee that stepwise will find the "best" combination of predictors (defined as, say, the highest adjusted R^2); it can get stuck in local optima´and never reach the so-called global optima which might be the desired solution**

(b)

```r
# Logistic reg - 1
log.reg1 <- glm(z ~ 1 + x + I(x^2), family = "binomial", data = HDS_ex3)
summary(log.reg1)
```

```
##
## Call:
## glm(formula = z ~ 1 + x + I(x^2), family = "binomial", data = HDS_ex3)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.130  -1.015   0.524   1.001   1.489
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.4880     0.7475   3.328 0.000873 ***
## x           -12.2659     3.3066  -3.709 0.000208 ***
## I(x^2)       11.7626     3.1477   3.737 0.000186 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 151.18  on 109  degrees of freedom
## Residual deviance: 133.73  on 107  degrees of freedom
## AIC: 139.73
##
## Number of Fisher Scoring iterations: 4
```

```r
# Logistic reg - 2
log.reg2 <- glm(z ~ 1+ x + I(x^2) + y, family = "binomial", data = HDS_ex3)
summary(log.reg2)
```

```
##
## Call:
## glm(formula = z ~ 1 + x + I(x^2) + y, family = "binomial", data = HDS_ex3)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0623  -0.9985   0.4939   0.9803   1.8278
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -6.267      4.396  -1.426   0.1540
## x             -3.692      5.246  -0.704   0.4815
## I(x^2)         3.540      5.026   0.704   0.4812
## y              8.648      4.326   1.999   0.0456 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 151.18  on 109  degrees of freedom
## Residual deviance: 129.49  on 106  degrees of freedom
## AIC: 137.49
##
## Number of Fisher Scoring iterations: 4
```

(c)

```r
# Backward Selection
model.aic.backward <- step(log.reg2,  data = HDS_ex3, direction = "backward", k = 2, trace = FALSE)
summary(model.aic.backward)
```

```
##
## Call:
## glm(formula = z ~ y, family = "binomial", data = HDS_ex3)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0331  -1.0155   0.5088   0.9903   1.9030
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -8.961      2.262  -3.962 7.42e-05 ***
## y             11.065      2.737   4.043 5.28e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 151.18  on 109  degrees of freedom
## Residual deviance: 129.99  on 108  degrees of freedom
## AIC: 133.99
##
## Number of Fisher Scoring iterations: 3
```

**Based on this the choice is the model with just "y" (formula = z ~ y) instead of "x" and its quadratic term.**