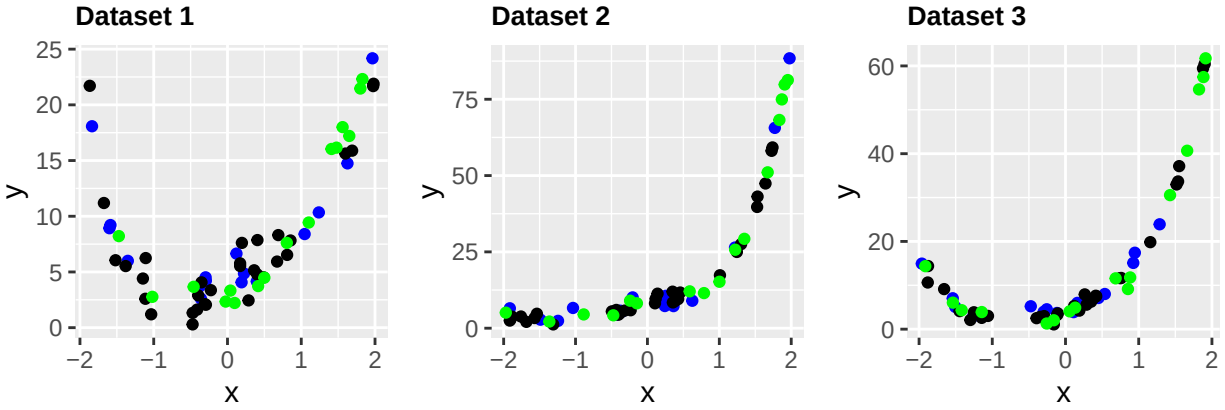
Assignment 1 Question 5 Report

2023-09-27

Finding K for Each Dataset

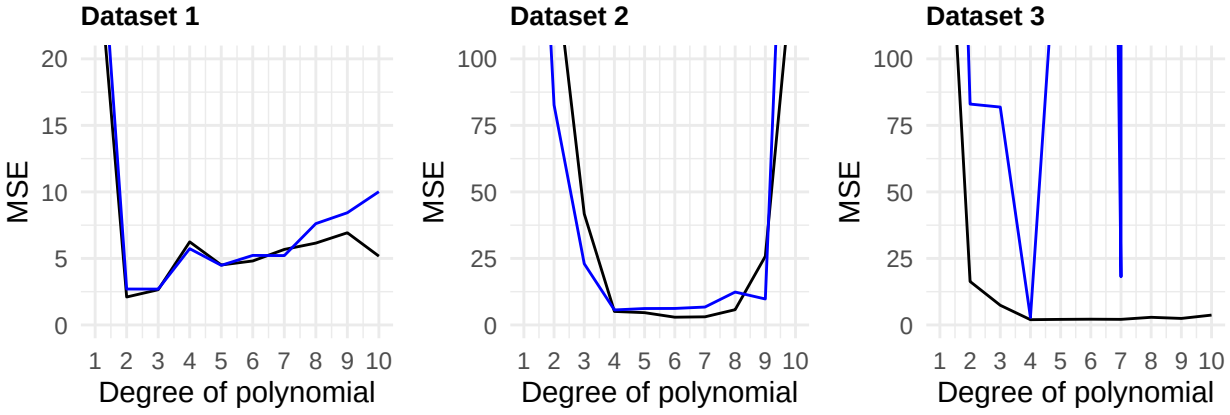
Visualizing the data

The blue data is from the small training set, the black data is from the large training set and the green data is from the test data set.



Testing for different degrees

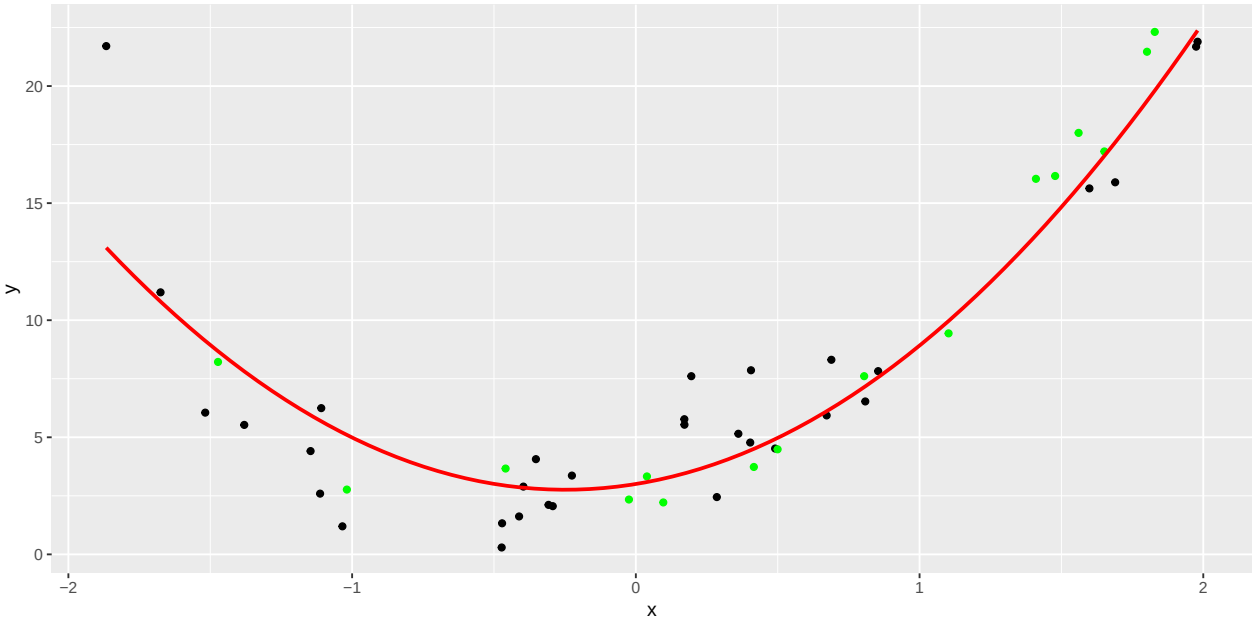
To find the optimal regression model degree K, I created models of varying degrees from 1 to 10. To test the fit of the data, I calculated the MSE to its predictor which was created using the respective small and large dataset, I have included the detailed MSE table for each datasheet in Appendix A, it is instead visualized here. Black lines represent the MSE from the large data and the blue lines represent the MSE from the small data.



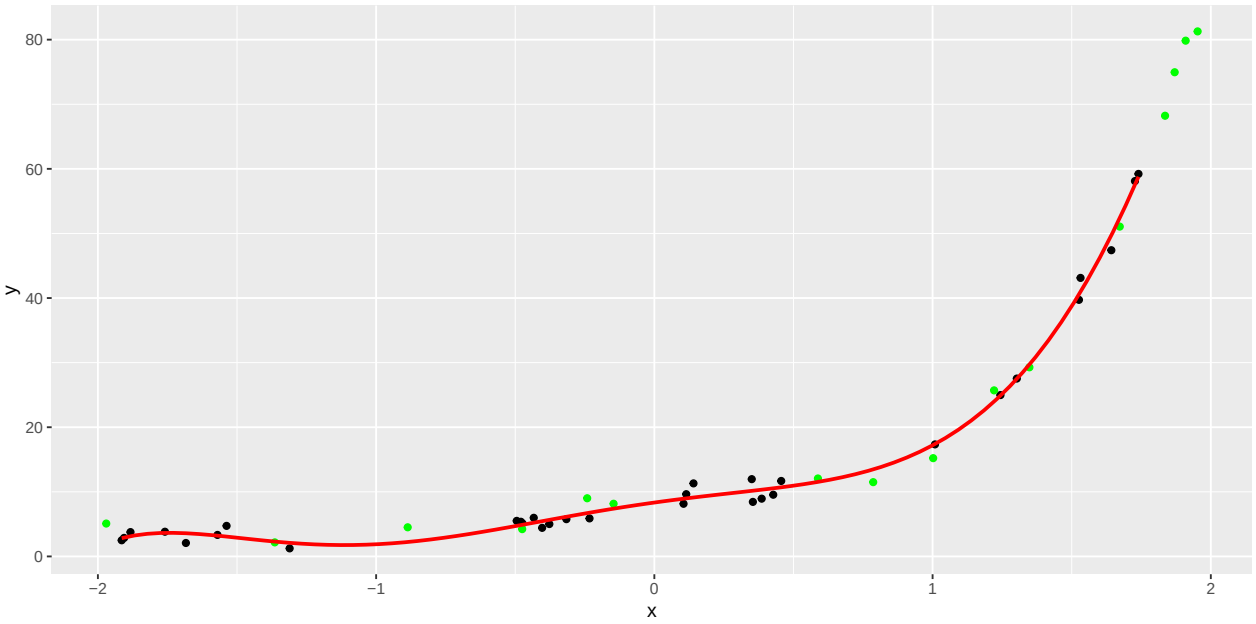
Visualizing the Final Model

Dataset 1 I noticed that MSE is minimized at degree 2 using the large data set in testing. Hence, the model should be implemented with K = 2. Therefore, the equation would look like $y_i = \beta_0 + x_i\beta_1 + x_i^2\beta_2 + \epsilon_i$

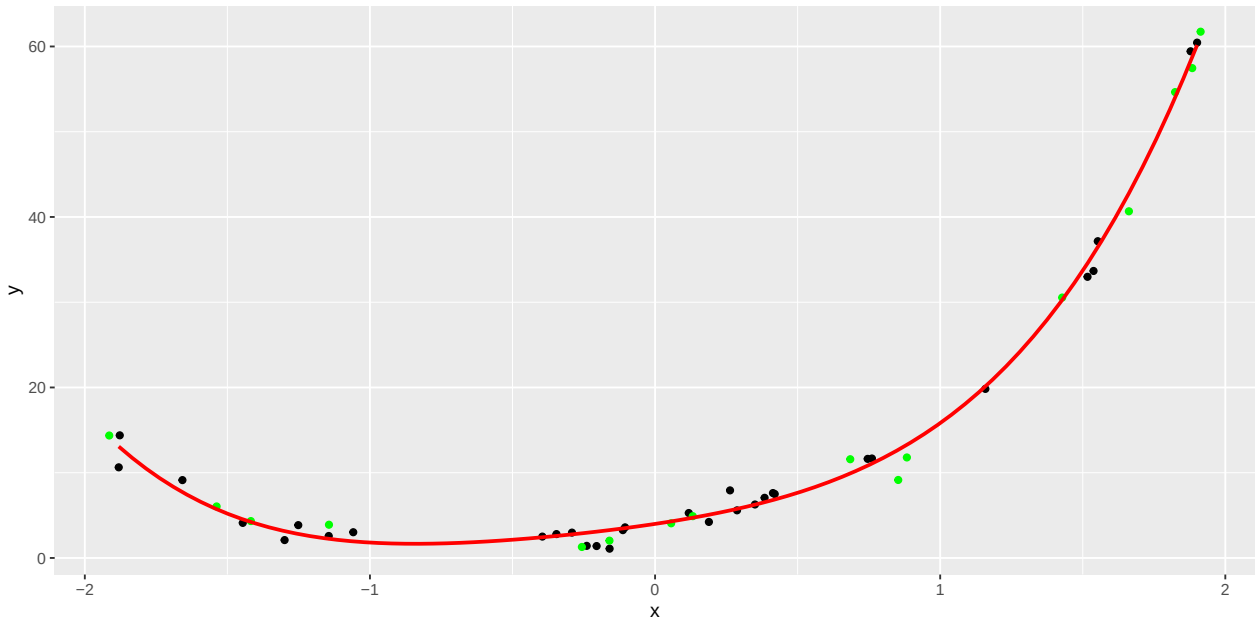
where $\epsilon_i \sim N(0,1)$.



Dataset 2 Similarily for dataset 2, the model should be implemented with K = 6. Hence the regression equation should be $y_i = \beta_0 + x_i\beta_1 + x_i^2\beta_2 + x_i^3\beta_3 + x_i^4\beta_4 + x_i^5\beta_5 + x_i^6\beta_6 + \epsilon_i$ where $\epsilon_i \sim N(0,1)$.

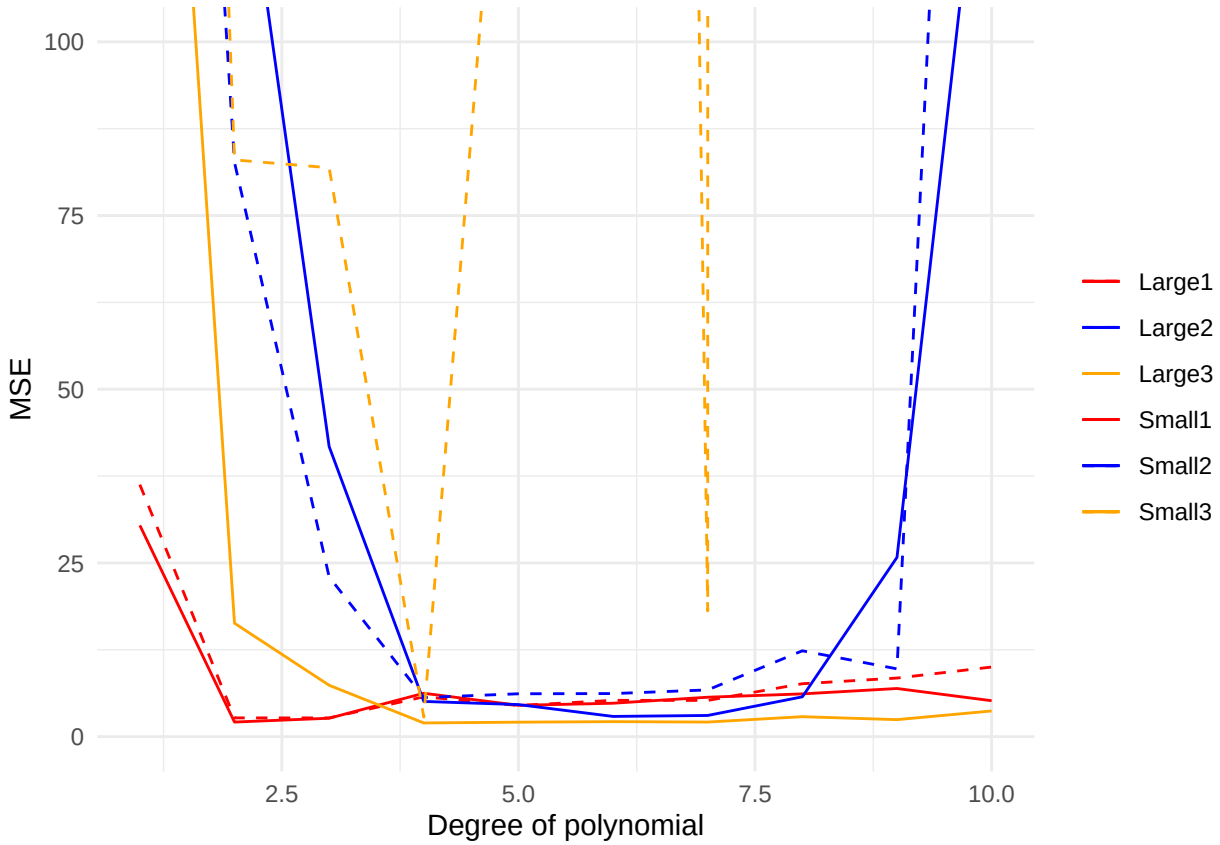


Dataset 3 Similarily for dataset 3, the model should be implemented with K = 4. Hence the regression equation should be $y_i = \beta_0 + x_i\beta_1 + x_i^2\beta_2 + x_i^3\beta_3 + x_i^4\beta_4 + \epsilon_i$ where $\epsilon_i \sim N(0,1)$.



Finding The Influence of Dataset Size

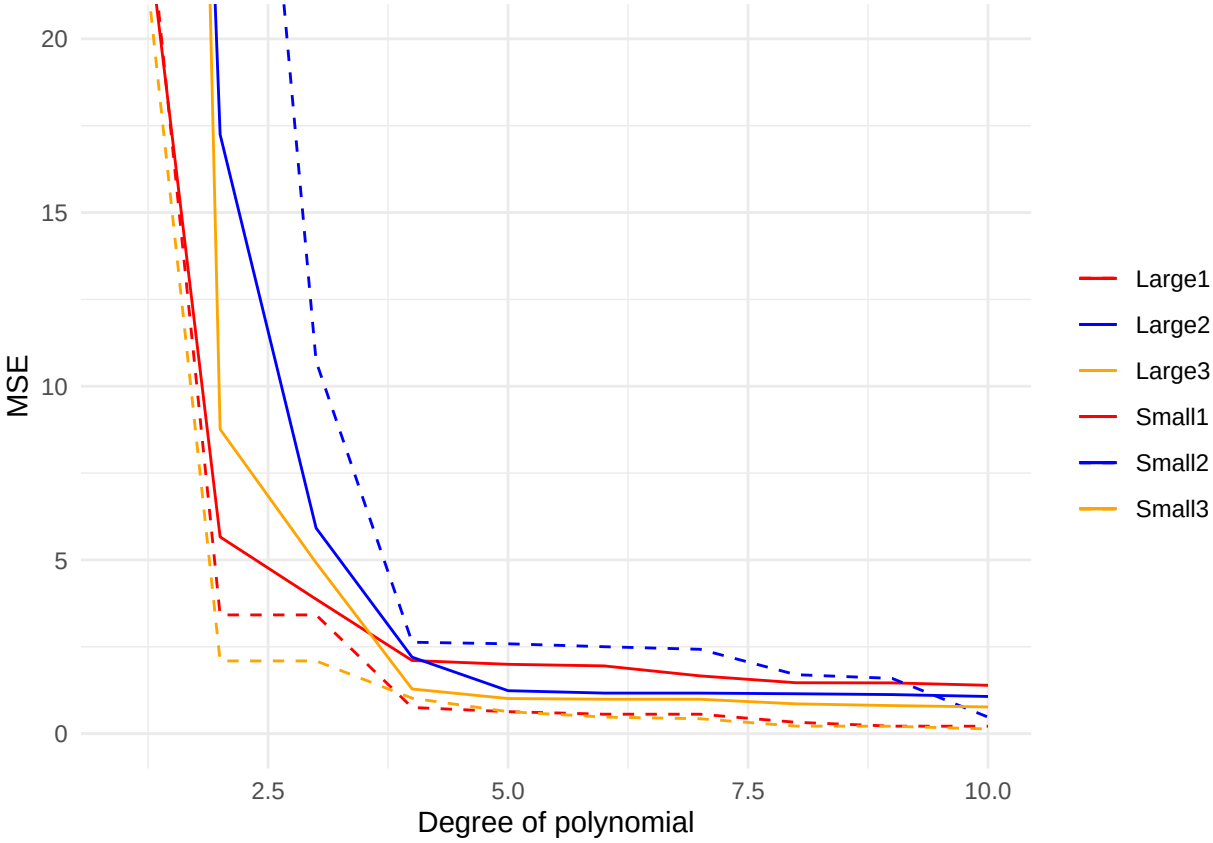
Influence on Predicting Test Data



The solid lines are generated using MSE of the large training data of each dataset predicting test data of their respective dataset, and the dotted uses the small dataset from each respective dataset. Visually, the sold lines of each color are generally lower than dotted lines of each color. In addition, looking at the MSE

datasets in Appendix A, I noticed that the larger dataset usually has less MSE. The large datasets will only have a larger MSE when the model is either underfitted or overfitted.

Influence on Predicting Training Data



Similarly, the solid lines are generated using the MSE of large model predicting the large training set, and the dashed lines small model predicting the small training set. In this case, the large model generally has more MSE than the small model. As the degree increases and the model overfits, the smaller model will reduce its MSE at a rate faster than the large model and generally has less MSE as the degree increases.

Conclusion

In conclusion, having more data will generally mean having less MSE and a better model fit for test data, and worse MSE and worse model fit for training data. Therefore, having more data will generally increase model accuracy.

Finding The Influence of Degrees

See Appendix A for a detailed description of the MSE tables and Appendix B for a visual representation of each small and large model from degrees 1-10.

When the model is underfitted (ie. linear), both the test and training MSE are generally high. As the model approaches the optimal degree, the training MSE is decreasing and the test MSE is also decreasing. When the model starts to overfit, the training MSE is still decreasing however the test MSE will be increasing.

Therefore, the further it deviates from the optimal degree, the more MSE it will have when predicting test data and the less MSE it will have when predicting training data. Hence finding the optimal degree is crucial in having a accurate model.

Appendix A: Detailed MSE Tables

Dataset 1

```
##    Degree Large_To_Test Small_To_Test
## 1       1     30.407982     36.263945
## 2       2      2.099428      2.697821
## 3       3      2.647117      2.694149
## 4       4      6.244531      5.730822
## 5       5      4.505936      4.480249
## 6       6      4.810027      5.225318
## 7       7      5.674785      5.212097
## 8       8      6.152186      7.615242
## 9       9      6.927631      8.433212
## 10     10      5.169417     10.016358
```

```
##    Degree Large_To_Train Small_To_Train
## 1       1      28.771679     30.5970661
## 2       2       5.663763      3.4179651
## 3       3       3.870950      3.4178977
## 4       4       2.105443      0.7532015
## 5       5       1.996164      0.6325159
## 6       6       1.950295      0.5597886
## 7       7       1.662841      0.5594646
## 8       8       1.465245      0.3269160
## 9       9       1.460570      0.2154632
## 10     10       1.391774      0.2139145
```

Dataset 2

```
##    Degree Large_To_Test Small_To_Test
## 1       1    438.073761    298.281854
## 2       2    138.863935     82.644572
## 3       3     41.748396     22.988521
## 4       4      5.068467      5.629123
## 5       5      4.612989      6.167429
## 6       6      2.909228      6.199278
## 7       7      3.046615      6.724248
## 8       8      5.728867     12.356811
## 9       9     25.799732      9.757250
## 10     10    145.482754    286.947548
```

```
##    Degree Large_To_Train Small_To_Train
## 1       1      90.240691    230.4641021
## 2       2      17.248005     41.3686368
## 3       3       5.918811     10.7219987
## 4       4       2.201385      2.6314235
## 5       5       1.236560      2.5868202
## 6       6       1.167758      2.5028055
## 7       7       1.166510      2.4282766
## 8       8       1.148167      1.6970697
## 9       9       1.124857      1.5913511
## 10     10       1.070496      0.4771781
```

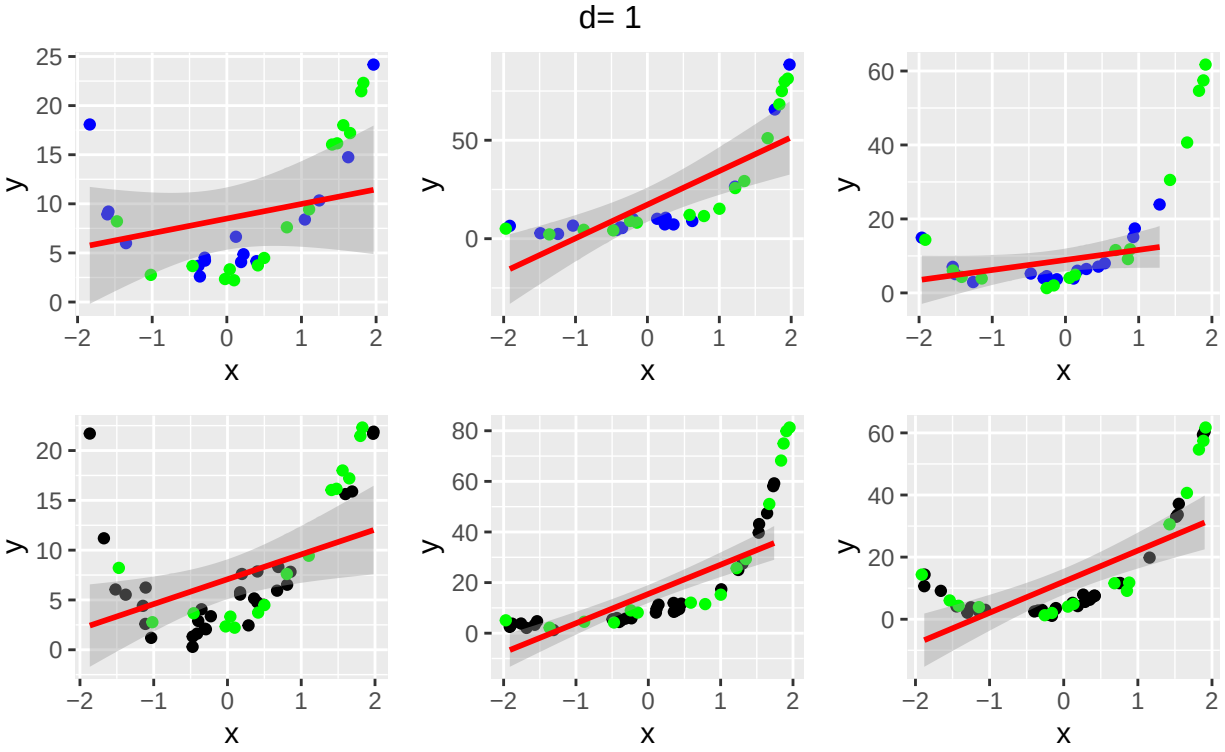
Dataset 3

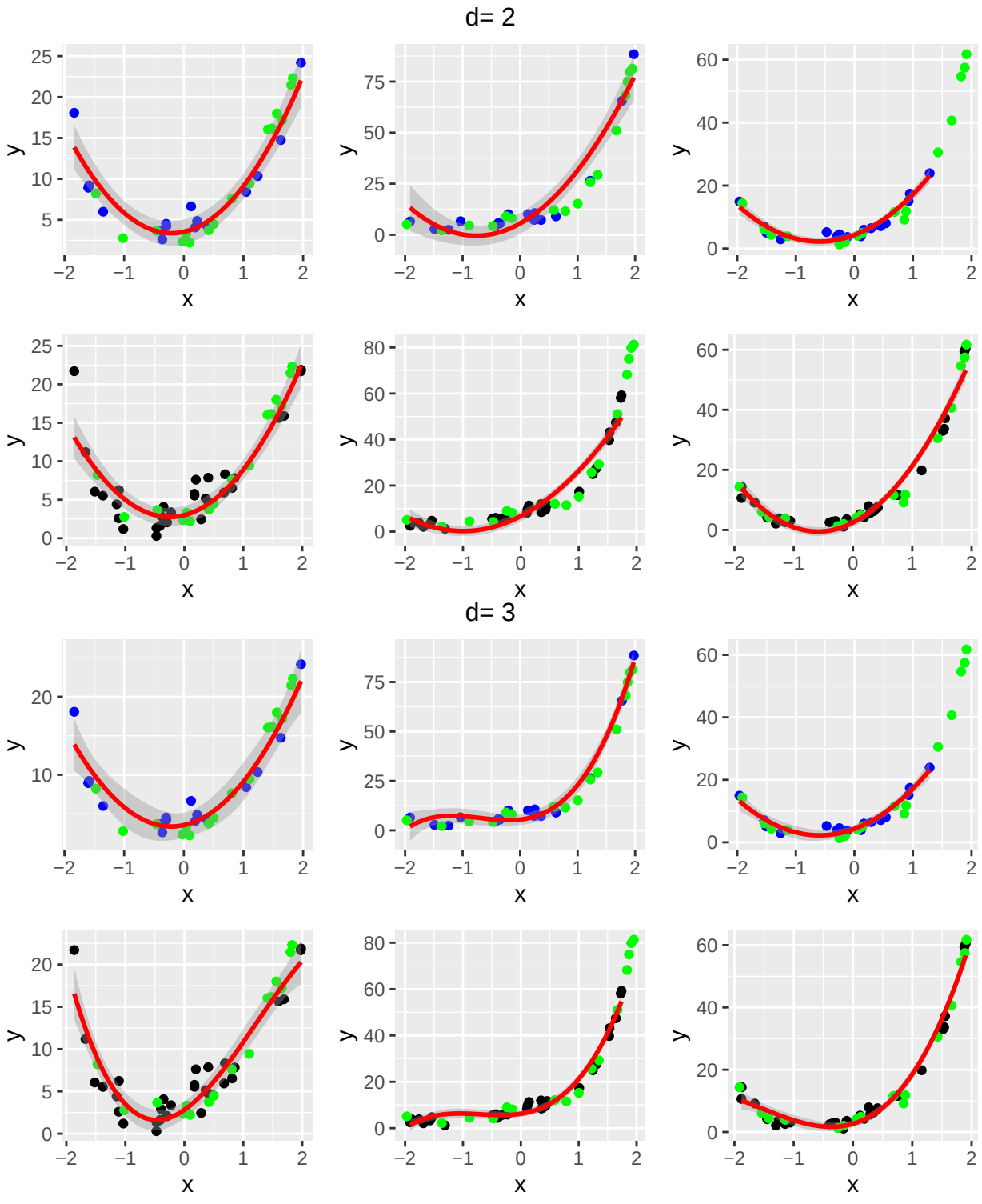
```
##    Degree Large_To_Test Small_To_Test
## 1       1    220.937512  4.457262e+02
## 2       2     16.324396  8.300451e+01
## 3       3      7.392283  8.188083e+01
## 4       4      1.970621  2.813325e+00
## 5       5      2.086343  1.700593e+02
## 6       6      2.158495  9.593558e+02
## 7       7      2.099509  1.810553e+01
## 8       8      2.871519  1.405368e+05
## 9       9      2.445929  3.418990e+05
## 10     10      3.696438  1.237566e+07
```

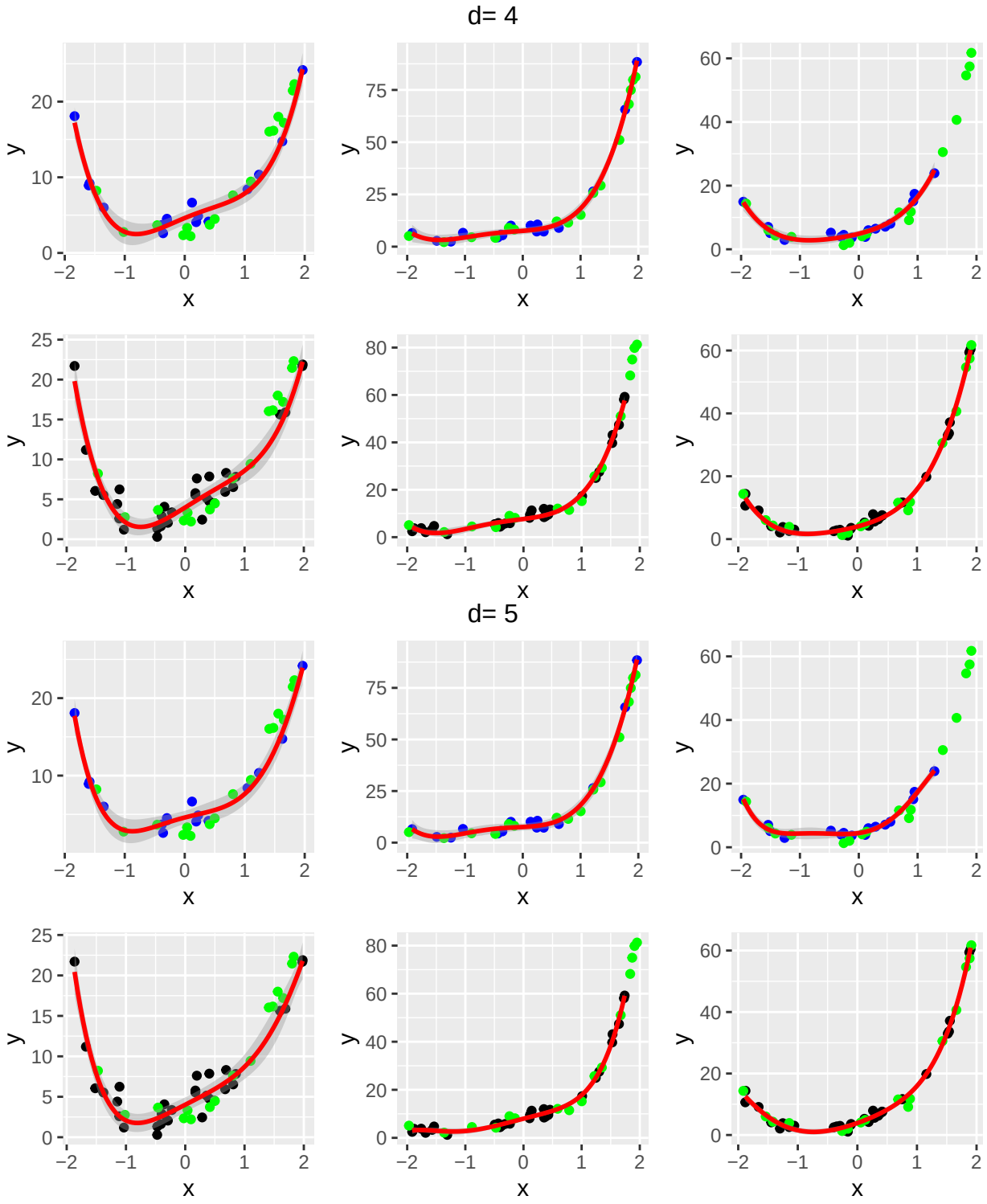
```
##    Degree Large_To_Train Small_To_Train
## 1       1    126.4979416     27.9602339
## 2       2      8.7625695      2.0936462
## 3       3      4.9184891      2.0934227
## 4       4      1.2860318      1.0210157
## 5       5      1.0110131      0.6336013
## 6       6      0.9916978      0.4772344
## 7       7      0.9904718      0.4295020
## 8       8      0.8572659      0.2164968
## 9       9      0.8078484      0.2135462
## 10     10      0.7686227      0.1342066
```

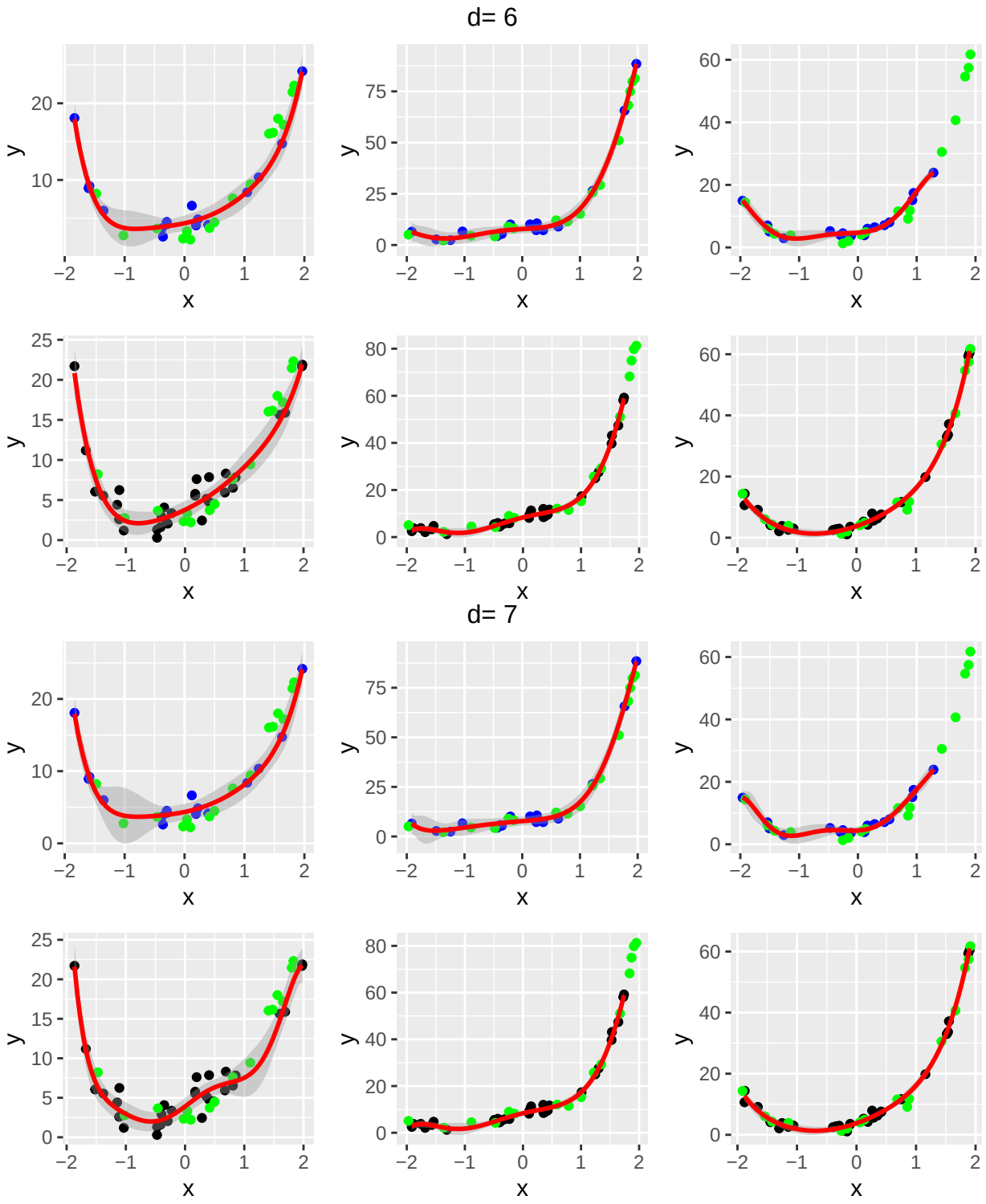
Appendix B: Different Degrees Graphs

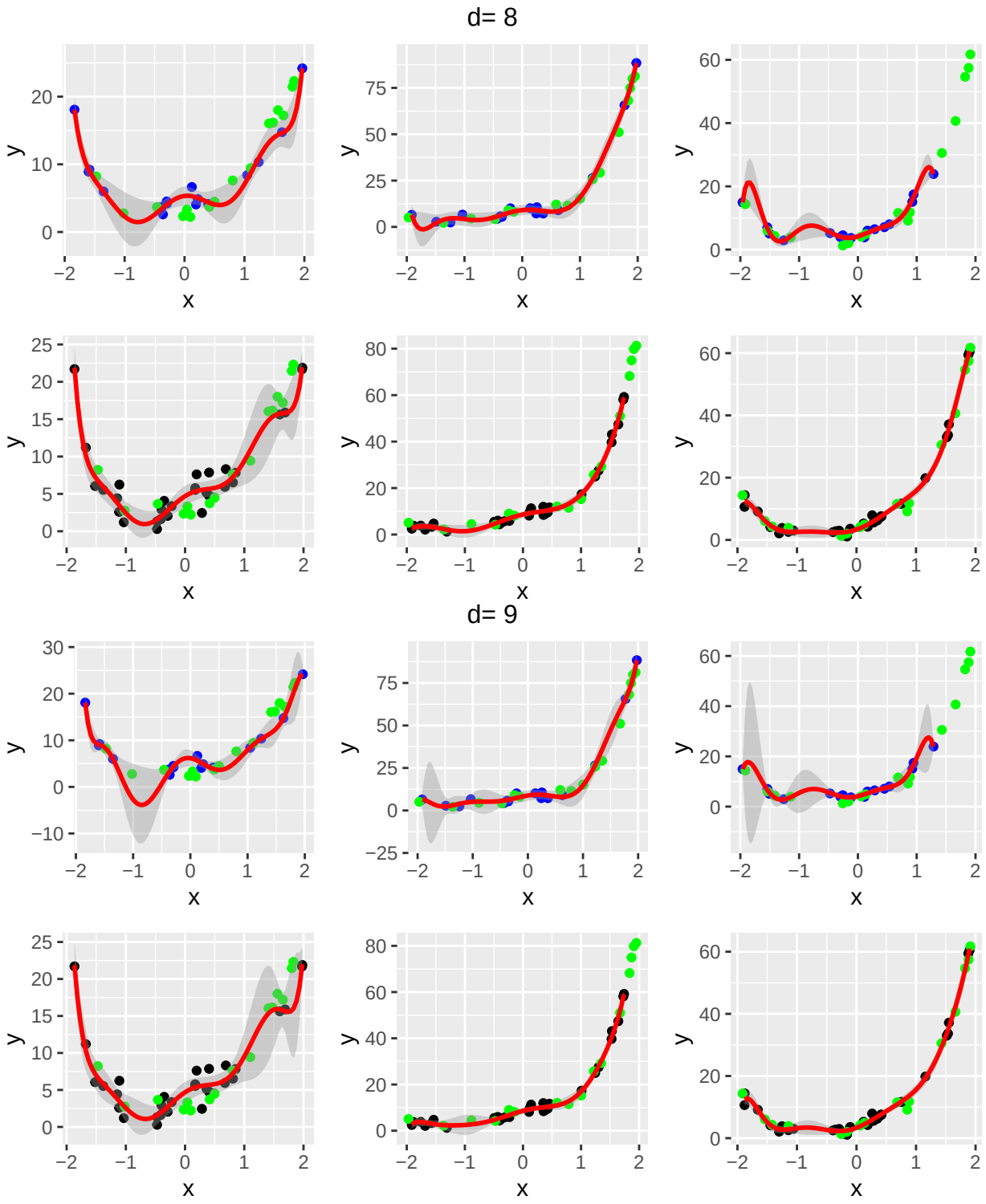
Left data is from dataset 1, middle model is from dataset 2, and the right model is from dataset 3.

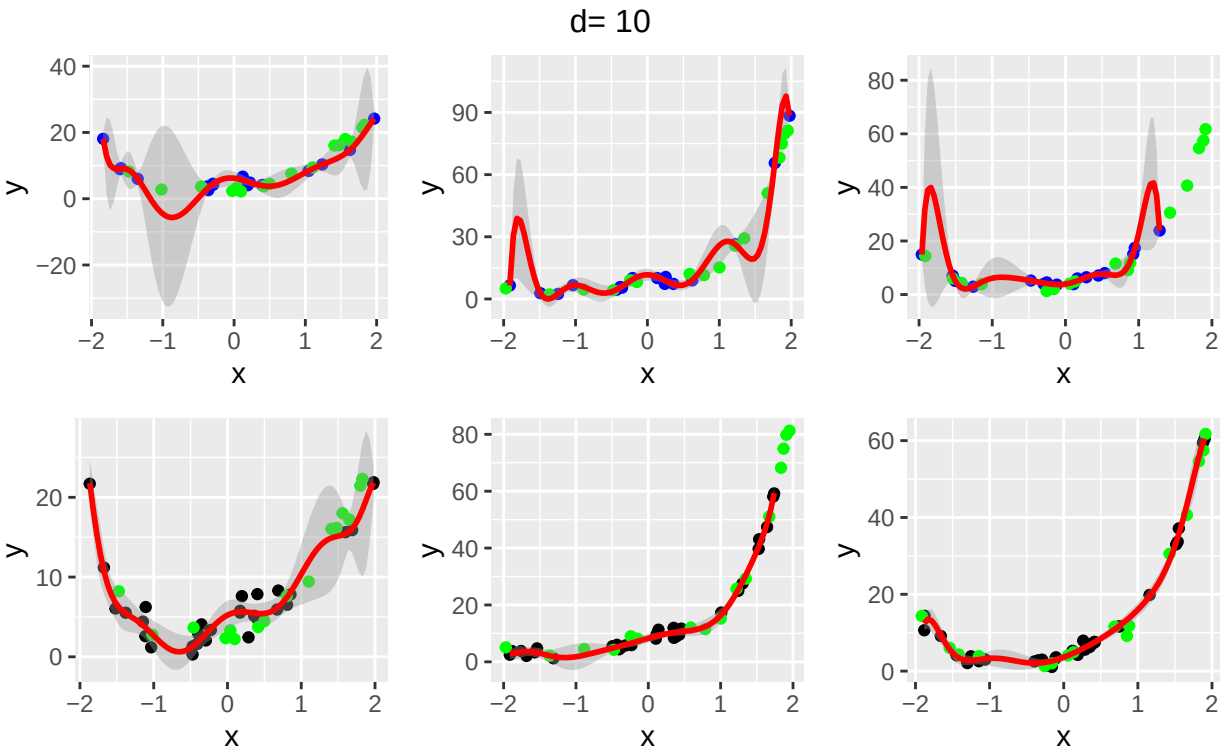



d= 2
d= 3
x
y


d= 4
d= 5
x
y


d= 6
d= 7
x
y


d= 8
d= 9
x
y



Appendix C: Implementation details

Data Import and Setup

```
#Setting up needed libraries
library(tidyverse)
library(ggplot2)
library(readxl)
library(gridExtra)
library(grid)
library(rio)
library(formatR)
#Importing data
data <- import_list("~/Desktop/ UofTears Code/STA314/A1/Dataset_1.xlsx")
```

Visualization Code

```
for (s in 1:3) {
  small = paste("data$small_train_",s, sep="")
  large = paste("data$large_train_",s, sep="")
  test = paste("data$test_data_",s, sep="")
  title = paste("Dataset",s)
  p = paste("plot", s, sep="")
  assign(p, ggplot(data=NULL, aes(x=x,y=y)) +
    geom_point(data = eval(parse(text = small)), color="blue") +
    geom_point(data = eval(parse(text = large)), color="black") +
    geom_point(data = eval(parse(text = test)), color="green") +
    ggtitle(title) + theme(plot.title = element_text(size = 10, face = "bold")))
```

```
}
grid.arrange(plot1,plot2,plot3, ncol = 3, nrow = 2)
```

Optimal K Visualization

```
# For Dataset 3
  ggplot(data=NULL, aes(x=x,y=y)) +
    geom_point(data = data$large_train_3, color="black") +
    geom_point(data = data$test_data_3, color="green") +
    stat_smooth(data = data$large_train_3, method = "lm",
                formula = y ~ poly(x,4,raw = TRUE), se =FALSE, color="red")
```

```
#For Dataset 2
  ggplot(data=NULL, aes(x=x,y=y)) +
    geom_point(data = data$large_train_2, color="black") +
    geom_point(data = data$test_data_2, color="green") +
    stat_smooth(data = data$large_train_2, method = "lm",
                formula = y ~ poly(x,6,raw = TRUE), se =FALSE, color="red")
```

```
#For Dataset 1
  ggplot(data=NULL, aes(x=x,y=y)) +
    geom_point(data = data$large_train_1, color="black") +
    geom_point(data = data$test_data_1, color="green") +
    stat_smooth(data = data$large_train_1, method = "lm",
                formula = y ~ poly(x,2,raw = TRUE), se =FALSE, color="red")
```

Single Dataset MSE graphs and calculations

Training MSE

```
for (s in 1:3) {
  #resetting the data tables
  large_mse_test <- c()
  small_mse_test <- c()

  #changing datasets
  small = paste("data$small_train_",s, sep="")
  large = paste("data$large_train_",s, sep="")
  test = paste("data$test_data_",s, sep="")
  mse = paste("mse",s, sep="")
  title = paste("Dataset",s)
  p = paste("plot", s, sep="")

  for (d in 1:10){
    # creating the models of degree d
    model_small <- lm(y ~ poly(x, d, raw = TRUE), data = eval(parse(text = small)))
    model_large <- lm(y ~ poly(x, d, raw = TRUE), data = eval(parse(text = large)))

    #create the prediction models
    large_test_fit <- model_large %>% predict(eval(parse(text = test)))
    small_test_fit <- model_small %>% predict(eval(parse(text = test)))

    #calculating MSE
```

```
    small_test_mse <- mean((eval(parse(text = test))$y-small_test_fit)^2)
    large_test_mse <- mean((eval(parse(text = test))$y-large_test_fit)^2)

    small_mse_test <- append(small_mse_test, small_test_mse)
    large_mse_test <- append(large_mse_test, large_test_mse)
  }

  #storing data table for Appendix
  assign(mse, data.frame(
  Degree = 1:10,
  Large_To_Test = large_mse_test ,
  Small_To_Test = small_mse_test
  ))

  #limit for each graph
  if (s == 1) {
    lim = 20
  } else {
    lim = 100
  }

  #Visualizing MSE
  assign(p, ggplot(eval(parse(text = mse)), aes(x = Degree)) +
    geom_line(aes(y = Large_To_Test, color = "Large"), linetype = "solid") +
    geom_line(aes(y = Small_To_Test, color = "Small"), linetype = "solid") +
    coord_cartesian(ylim=c(0, lim)) +
    labs(
    x = "Degree of polynomial",
    y = "MSE"
    ) +
    scale_color_manual(values = c("Small" = "blue", "Large" = "black")) +
    theme_minimal() + ggtitle(title) +
    theme(plot.title = element_text(size = 10, face = "bold"), legend.position = "none")
    + scale_x_continuous(breaks=c(1,2,3,4,5,6,7,8,9,10)))
}
grid.arrange(plot1,plot2,plot3, ncol = 3, nrow = 2)
```

Test

```
#Generating Training Data MSE
for (s in 1:3) {
  large_mse_train <- c()
  small_mse_train <- c()

  #changing datasets
  small = paste("data$small_train_",s, sep="")
  large = paste("data$large_train_",s, sep="")
  mse = paste("mse_train_",s, sep="")
  mseT = paste("mse",s,sep="")
  title = paste("Dataset",s)

  for (d in 1:10) {
    small_train_model <- lm(y ~ poly(x, d, raw = TRUE), data = eval(parse(text = small)))
```

```
    large_train_model <- lm(y ~ poly(x, d, raw = TRUE), data = eval(parse(text = large)))

    small_train_fit <- small_train_model %>% predict(eval(parse(text = small)))
    large_train_fit <- large_train_model %>% predict(eval(parse(text = large)))

    small_train_mse <- mean((eval(parse(text = small))$y-small_train_fit)^2)
    large_train_mse <- mean((eval(parse(text = large))$y-large_train_fit)^2)

    small_mse_train[d] <- small_train_mse
    large_mse_train[d] <- large_train_mse
  }

  #save for Appendix
  assign(mse, data.frame(
    Degree = 1:10,
    Large_To_Train = large_mse_train,
    Small_To_Train = small_mse_train
    ))
}
```

Multiple Dataset MSE graphs

Training

```
#visualizing MSE of training Data
ggplot(data =NULL, aes(x = mse1$Degree)) +
    geom_line(aes(y = mse_train_1$Large_To_Train, color = "Large1"), linetype = "solid") +
    geom_line(aes(y = mse_train_1$Small_To_Train, color = "Small1"), linetype = "dashed") +
    geom_line(aes(y = mse_train_2$Large_To_Train, color = "Large2"), linetype = "solid") +
    geom_line(aes(y = mse_train_2$Small_To_Train, color = "Small2"), linetype = "dashed") +
    geom_line(aes(y = mse_train_3$Large_To_Train, color = "Large3"), linetype = "solid") +
    geom_line(aes(y = mse_train_3$Small_To_Train, color = "Small3"), linetype = "dashed") +
    labs(
    x = "Degree of polynomial",
    y = "MSE"
    ) +
    coord_cartesian(ylim=c(0, 20)) +
    scale_color_manual(values = c("Large1" = "red", "Small1" = "red",
                                  "Large2" = "blue", "Small2" = "blue",
                                  "Large3" = "orange", "Small3" = "orange")) +
    theme_minimal() +
    theme(legend.title = element_blank())
```

Test

```
ggplot(data =NULL, aes(x = mse1$Degree)) +
    geom_line(aes(y = mse1$Large_To_Test, color = "Large1"), linetype = "solid") +
    geom_line(aes(y = mse1$Small_To_Test, color = "Small1"), linetype = "dashed") +
    geom_line(aes(y = mse2$Large_To_Test, color = "Large2"), linetype = "solid") +
    geom_line(aes(y = mse2$Small_To_Test, color = "Small2"), linetype = "dashed") +
    geom_line(aes(y = mse3$Large_To_Test, color = "Large3"), linetype = "solid") +
    geom_line(aes(y = mse3$Small_To_Test, color = "Small3"), linetype = "dashed") +
    labs(
```

```
    x = "Degree of polynomial",
    y = "MSE"
    ) +
    coord_cartesian(ylim=c(0, 100)) +
    scale_color_manual(values = c("Large1" = "red", "Small1" = "red",
                                  "Large2" = "blue", "Small2" = "blue",
                                  "Large3" = "orange", "Small3" = "orange")) +
    theme_minimal() +
    theme(legend.title = element_blank())
```

Implementation for Appendix B

```
for (d in 1:10) {
  a1 <- ggplot(data=NULL, aes(x=x,y=y)) +
  geom_point(data = data$small_train_1, color="blue") +
  geom_point(data = data$test_data_1, color="green") +
  stat_smooth(data = data$small_train_1, method = "lm",
              formula = y ~ poly(x,d,raw = TRUE), se =TRUE, color="red")

  a2 <- ggplot(data=NULL, aes(x=x,y=y)) +
  geom_point(data = data$small_train_2, color="blue") +
  geom_point(data = data$test_data_2, color="green") +
  stat_smooth(data = data$small_train_2, method = "lm",
              formula = y ~ poly(x,d,raw = TRUE), se =TRUE, color="red")

  a3 <- ggplot(data=NULL, aes(x=x,y=y)) +
  geom_point(data = data$small_train_3, color="blue") +
  geom_point(data = data$test_data_3, color="green") +
  stat_smooth(data = data$small_train_3, method = "lm",
              formula = y ~ poly(x,d,raw = TRUE), se =TRUE, color="red")

  b1 <- ggplot(data=NULL, aes(x=x,y=y)) +
  geom_point(data = data$large_train_1, color="black") +
  geom_point(data = data$test_data_1, color="green") +
  stat_smooth(data = data$large_train_1, method = "lm",
              formula = y ~ poly(x,d,raw = TRUE), se =TRUE, color="red")

  b2 <- ggplot(data=NULL, aes(x=x,y=y)) +
  geom_point(data = data$large_train_2, color="black") +
  geom_point(data = data$test_data_2, color="green") +
  stat_smooth(data = data$large_train_2, method = "lm",
              formula = y ~ poly(x,d,raw = TRUE), se =TRUE, color="red")

  b3 <- ggplot(data=NULL, aes(x=x,y=y)) +
  geom_point(data = data$large_train_3, color="black") +
  geom_point(data = data$test_data_3, color="green") +
  stat_smooth(data = data$large_train_3, method = "lm",
              formula = y ~ poly(x,d,raw = TRUE), se =TRUE, color="red")

  text <- paste("d=",d)
  grid.arrange(a1, a2,a3,b1,b2,b3, ncol=3, nrow=2, top=textGrob(text))
}
```