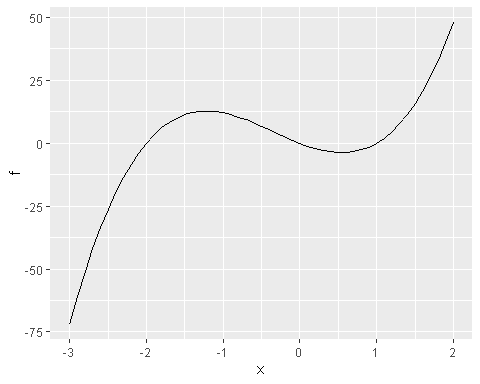
DSE6211 Module 06 Lab 06

Joseph Annand

2024-02-24

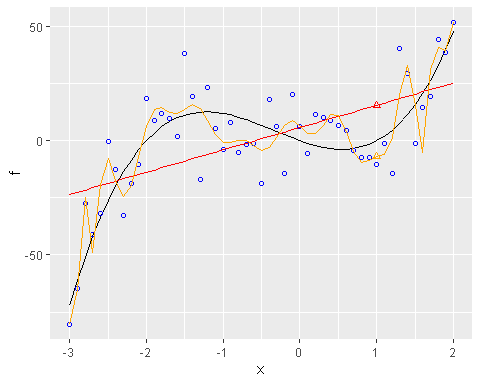
## Bias Variance Trade-off

set.seed(123)  
library(ggplot2)  
  
true\_relationship <- function(x) {   
 return(6\*x^3 + 6\*x^2 - 12\*x)   
 }  
x <- seq(-3, 2, by = 0.1)  
f <- true\_relationship(x)  
  
ggplot() + geom\_line(aes(x = x, y = f), color = "black")



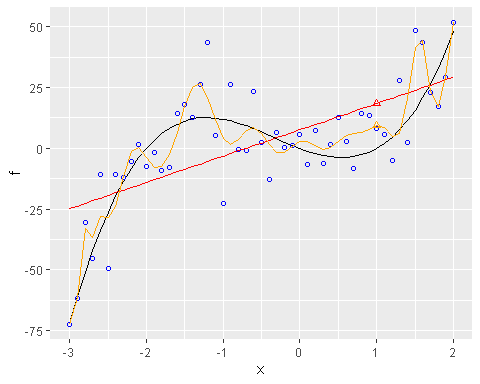
observations <- f + rnorm(length(x), mean = 0, sd = 15)  
  
model1 <- lm(observations ~ poly(x, 1))  
predictions1 <- predict(model1, newdata = data.frame(x = x))  
  
model25 <- lm(observations ~ poly(x, 25))  
predictions25 <- predict(model25, newdata = data.frame(x = x))

data <- data.frame(x = x,  
 f = f,  
 observations = observations,  
 lm = predictions1,  
 pm = predictions25)  
ggplot(data, aes(x = x)) +   
 geom\_line(aes(y = f), color = "black") +  
 geom\_point(aes(y = observations), color = "blue", shape = 1) +  
 geom\_line(aes(y = lm), color = "red", linetype = "solid") +  
 geom\_line(aes(y = pm), color = "orange", linetype = "solid") +  
 geom\_point(aes(x = 1, y = data[x == 1, "lm"]), color = "red", shape=2) +  
 geom\_point(aes(x = 1, y = data[x == 1, "pm"]), color = "orange", shape=2)

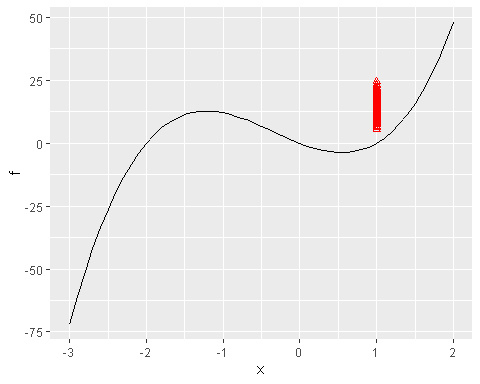


observations\_new <- f + rnorm(length(x), mean = 0, sd = 15)  
  
model1 <- lm(observations\_new ~ poly(x, 1))  
predictions1 <- predict(model1, newdata = data.frame(x = x))  
  
model25 <- lm(observations\_new ~ poly(x, 25))  
predictions25 <- predict(model25, newdata = data.frame(x = x))

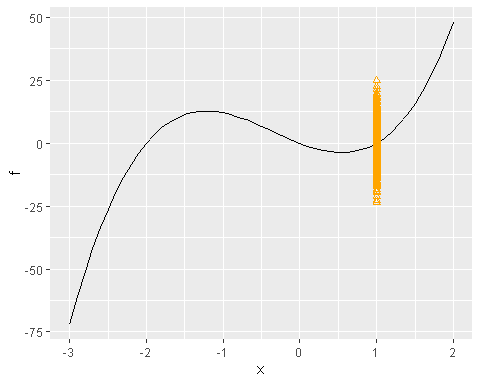
data <- data.frame(x = x,  
 f = f,  
 observations = observations\_new,  
 lm = predictions1,  
 pm = predictions25)  
  
ggplot(data, aes(x = x)) +   
 geom\_line(aes(y = f), color = "black") +  
 geom\_point(aes(y = observations\_new), color = "blue", shape = 1) +  
 geom\_line(aes(y = lm), color = "red", linetype = "solid") +  
 geom\_line(aes(y = pm), color = "orange", linetype = "solid") +  
 geom\_point(aes(x = 1, y = data[x == 1, "lm"]), color = "red", shape=2) +  
 geom\_point(aes(x = 1, y = data[x == 1, "pm"]), color = "orange", shape=2)



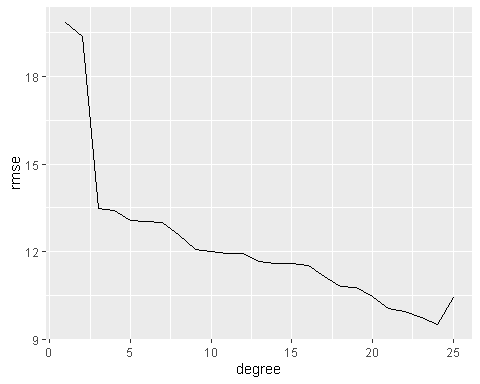
results1 <- data.frame(x = 1, f\_pred = 0)  
  
for (i in 1:500) {  
 x <- seq(-3, 2, by = 0.1)  
 f <- true\_relationship(x)  
 temp\_observations <- f + rnorm(length(x), mean=0, sd=15)  
 model1 <- lm(temp\_observations ~ poly(x, 1))  
 results1[i, 1] <- 1  
 results1[i, 2] <- predict(model1, newdata = data.frame(x=1))  
}  
  
ggplot() +  
 geom\_line(data = data, aes(x = x, y = f), color = "black") +  
 geom\_point(data = results1, aes(x = x, y = f\_pred), color="red", shape=2)



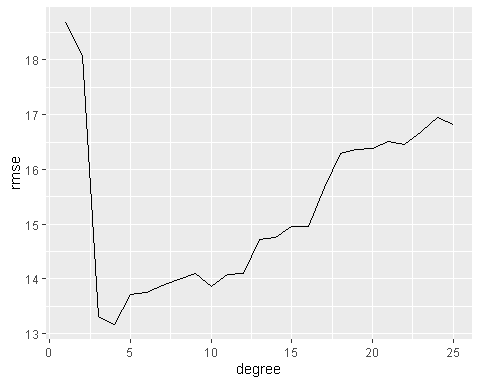
results20 <- data.frame(x = 1, f\_pred = 0)  
  
for (i in 1:500) {  
 x <- seq(-3, 2, by = 0.1)  
 f <- true\_relationship(x)  
 temp\_observations <- f + rnorm(length(x), mean=0, sd=15)  
 model20 <- lm(temp\_observations ~ poly(x, 20))  
 results20[i, 1] <- 1  
 results20[i, 2] <- predict(model20, newdata = data.frame(x=1))  
}  
  
ggplot() +  
 geom\_line(data = data, aes(x = x, y = f), color = "black") +  
 geom\_point(data = results20, aes(x = x, y = f\_pred), color="orange", shape=2)



models <- vector("list", 25)  
  
for (degree in 1:25) {  
 model <- lm(observations ~ poly(x, degree))  
 models[[degree]] <- model  
}  
  
results <- data.frame(degree = 1:25, rmse = 0)  
  
for (degree in 1:25) {  
 predictions <- predict(models[[degree]], newdata = data.frame(x=x))  
 results[results$degree==degree, "rmse"] <-  
 sqrt((1/length(predictions))\*sum((predictions-observations)^2))  
}  
  
ggplot() +  
 geom\_line(data = results, aes(x = degree, y = rmse), color = "black")



results <- data.frame(degree = 1:25, rmse = 0)  
  
for (degree in 1:25) {  
 predictions <- predict(models[[degree]], newdata = data.frame(x=x))  
 results[results$degree==degree, "rmse"] <-  
 sqrt((1/length(predictions))\*sum((predictions-observations\_new)^2))  
}  
  
ggplot() +  
 geom\_line(data = results, aes(x = degree, y = rmse), color = "black")



## Exercises

### Exercise 1

In the first plot, the model fits the original data better with each addition of a degree in its polynomial function. The RMSE continues to decrease when evaluated using the original data that was used to fit the model because the model is simply matching the original observations. The gradual decline in RMSE in the first plot is an example of the model overfitting the data by trying to model the observations rather than the true relationship.

In the second plot, the left side looks similar to the first plot: for the few degrees of the polynomial function, the model does not have enough flexibility to accurately predict values and is underfitting the data as seen by the high RMSE values. The plot reaches a minimum at degree = 4 and then RMSE begins to increase as the degree of the polynomial increases. The second plot uses RMSE values determined by the predictions by the model on a test data set, which the model has never seen before. Because the model tries to match the observations of the training data more than the true relationship as the degree of the polynomial increases, the variance of the model is too great to accurately predict the never-before-seen test data; therefore, the RMSE increases on the right side of the plot.

### Exercise 2

Based on the last plot above, the best degree of the polynomial function for the model is 4. The plot below shows that the fourth degree polynomial best models the true relationship of the data and performs well on test data.

model <- lm(observations ~ poly(x, 4))  
  
predictions=predict(model, newdata = data.frame(x=x))  
  
data = data.frame(x=x, f=f, predictions=predictions)  
  
ggplot(data, aes(x=x)) +  
 geom\_line(aes(y = f), color = "black") +  
 geom\_line(aes(y = predictions), color = "red", linetype="solid")

