Problem 1

Orthogonal PLS components

WLOG, assume $h > l$, at the first stage, prove that $z_h^T z_{h-1} = 0$

$$z_h^T = w_h^T X_{h-1}^T = w_h^T (X_{h-2}^T - p_{h-1} z_{h-1}^T) \tag{1}$$

$$\begin{aligned} z_h^T z_{h-1} &= w_h^T (X_{h-2}^T z_{h-1} - p_{h-1} z_{h-1}^T z_{h-1}) \\ &= w_h^T (p_{h-1} z_{h-1}^T z_{h-1} - p_{h-1} z_{h-1}^T z_{h-1}) \\ &= 0 \end{aligned} \tag{2}$$

The following is illustrated by mathematical induction.
If z_h and z_j are orthogonal for all $j \in [i+1, h-1]$, then when $j = i+1$ the inner product of both PLS components is that

$$\begin{aligned} z_h &= w_h^T (X_{i-1} - z_i p_i - z_{i+1} p_{i+1} - \ldots - z_{h-1} p_{h-1}^T) \\ z_h^T z_i &= w_h^T (X_{i-1}^T z_i - p_i z_i^T z_i - \sum_{j=i+1}^{h-1} p_j z_j z_i) \\ &= 0 \end{aligned} \tag{3}$$

Note that the first two terms are the inner product of $z_{i-1}^T z_i$, and the following terms satisfying our assumptions equals 0
Q.E.D.

Weights and loadings are collinear

$$w_h^T p_h = w_h^T X_{h-1}^T z_h / z_h^T z_h = z_h^T z_h / z_h^T z_h = 1 \tag{4}$$

Weights and predictor residuals are orthogonal

$$X_i w_i = (X_{i-1} - z_i p_i^T) w_i = z_i w_i^T w_i - z_i p_i^T w_i \tag{5}$$

Due to the fact that the weightings are standardized

$$w_i^T w_i = 1 \tag{6}$$

and as is proved in the last subsection

$$p_i^T X_i = 1 \tag{7}$$

We can easily find out that $X_i w_i = 0$

Problem 2 Bias of Regression Coefficients in PCR

Bias of $\hat{\beta}^{(k)}$

$$\begin{aligned}\mathbb{E}[\hat{\beta}^{(k)} - \beta] &= \mathbb{E}[V_k(Z - k^T Z_k)^{-1} Z_k^T Y - \beta] \\ &= \mathbb{E}[V_k(V_k^T X^T X V_k)^{-1} V_k^T X^T Y - \beta] \\ &= \mathbb{E}[(V_k^T V_k \Lambda V_k^T V_k)^{-1} V_k^T X^T Y - \beta] \\ &= \mathbb{E}[V_k \Lambda^{-1} V_k^T X^T Y - \beta] \\ &= (V_k \Lambda^{-1} V_k^T X^T X - I)\beta\end{aligned} \tag{8}$$

Note that $V_k^T V_k = 1$ and $\mathbb{E}(Y) = X\beta$

Bias of $\hat{\beta}^{(k)}$ when $k = p$

When k = p,

$$(X^T X)^{-1} = (V_k^T)^{-1} \Lambda^{-1} V_k^{-1} = V_k \Lambda^{-1} V_k^T \tag{9}$$

As is proved in the last subsection,

$$\mathbb{E}[\hat{\beta}^{(k)} - \beta] = ((X^T X)^{-1} X^T X - I)\beta = 0 \tag{10}$$

which is unbiased.

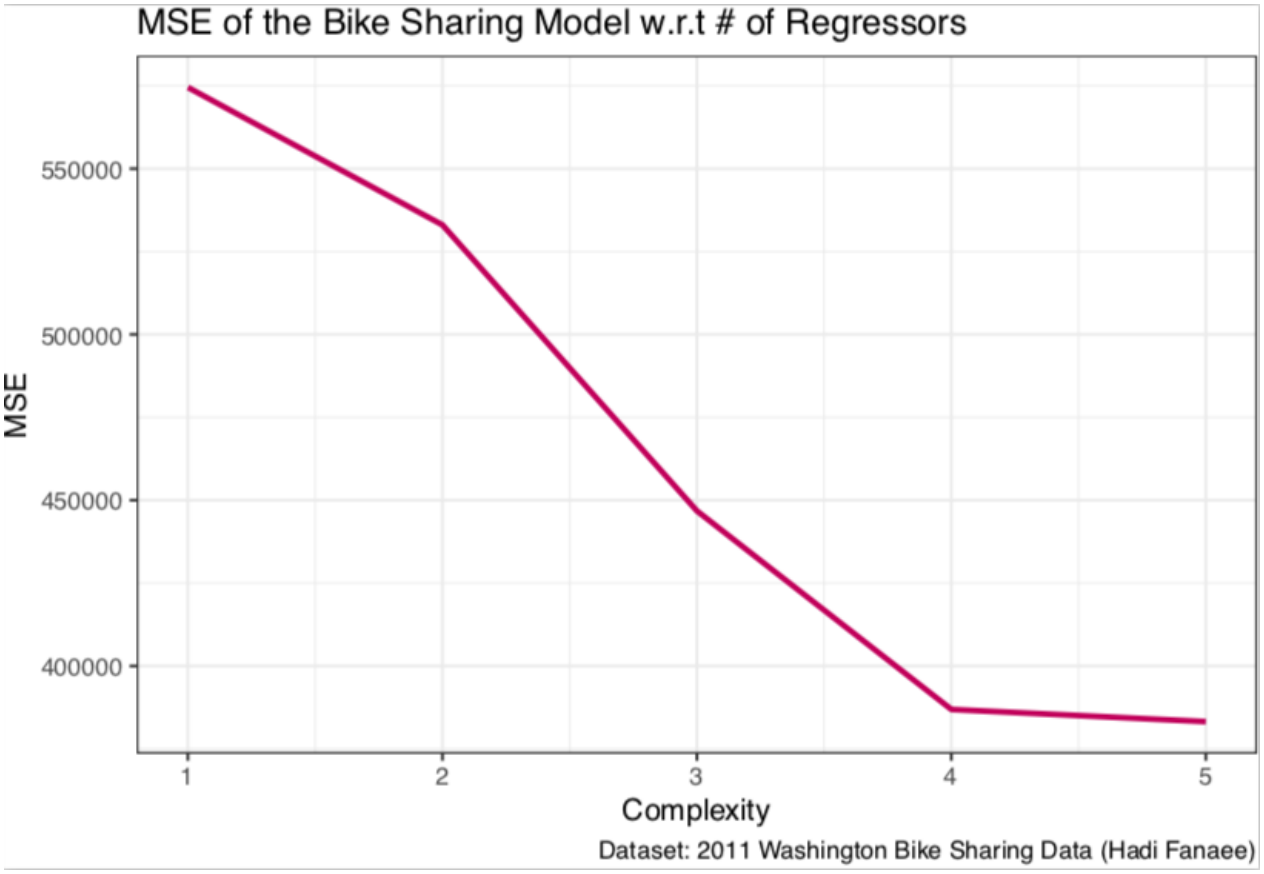
Problem 3 Fitting Models

```
set.seed(10000)
# Data Processing
day <- read.csv("day.csv")
hour <- read.csv("hour.csv")
data_day <- subset(day, yr == 0)
data_day <- data_day %>%
  mutate(clearday = ifelse(weathersit == 1, 1, 0))
data_day <- data_day %>%
  mutate(temp = (temp-min(temp))/(max(temp)-min(temp)))
```

```
# Fitting Models
model_1 <- lm(registered ~ temp, data = data_day)
model_2 <- lm(registered ~ temp + I(temp^2), data = data_day)
model_3 <- lm(registered ~ temp + I(temp^2) + workingday, data = data_day)
```

```
model_4 <- lm(registered ~ temp + I(temp^2) + workingday + clearday, data = data_day)
model_5 <- lm(registered ~ temp + I(temp^2) + workingday + clearday + temp*workingday, d
# stargazer(model_1, model_2, model_3, model_4, model_5, type = 'latex')
```

```
# Basic Model Comparison
MSE <- rep(0,5)
MSE[1] = mean(model_1$residuals^2)
MSE[2] = mean(model_2$residuals^2)
MSE[3] = mean(model_3$residuals^2)
MSE[4] = mean(model_4$residuals^2)
MSE[5] = mean(model_5$residuals^2)
MSE %>% data.frame("Complexity" = 1:5, MSE = MSE) %>%
  ggplot() + theme_bw() +
  geom_line(aes(x = Complexity, y = MSE), color = 'maroon', size = 1) +
  labs(
    title = "MSE of the Bike Sharing Model w.r.t # of Regressors",
    caption = "Dataset: 2011 Washington Bike Sharing Data (Hadi Fanaee)"
  )
```



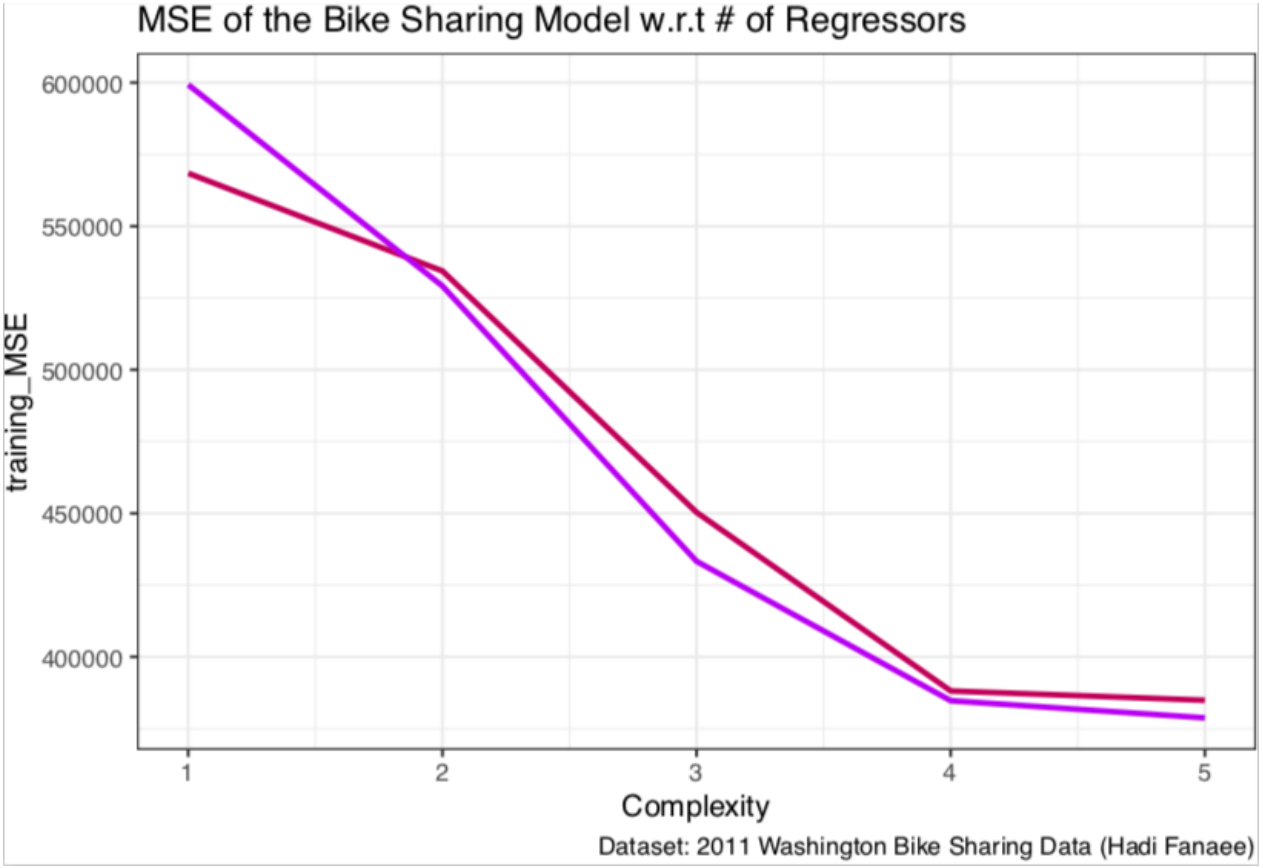
There's a downward trend in this graph, indicating that in-sample error is generally going down w.r.t. the complexity of models. As we're minimizing the in-sample error in our algorithm, this shall be invariably true.

Problem 4 Hold Out Method

```
# Hold-out Method
subs <- sample(365, 292)
# Subtracting Training and Test Set
trainingSet <- data_day[subs, ]
testSet <- data_day[-subs, ]
# Train the Model
model_1 <- lm(registered ~ temp, data = trainingSet)
model_2 <- lm(registered ~ temp + I(temp^2), data = trainingSet)
model_3 <- lm(registered ~ temp + I(temp^2) + workingday, data = trainingSet)
model_4 <- lm(registered ~ temp + I(temp^2) + workingday + clearday, data = trainingSet)
model_5 <- lm(registered ~ temp + I(temp^2) + workingday + clearday + temp*workingday, d
# Compute the training MSE
MSE <- rep(0,5)
MSE[1] = mean(model_1$residuals^2)
MSE[2] = mean(model_2$residuals^2)
MSE[3] = mean(model_3$residuals^2)
MSE[4] = mean(model_4$residuals^2)
MSE[5] = mean(model_5$residuals^2)
# Compute the test MSE
test_MSE = rep(0,5)
testX = cbind(1, testSet$temp)
test_MSE[1] = mean((testSet[, "registered"] - testX %*% model_1$coefficients)^2)
testX = cbind(1, testSet$temp, testSet$temp^2)
test_MSE[2] = mean((testSet[, "registered"] - testX %*% model_2$coefficients)^2)
testX = cbind(1, testSet$temp, testSet$temp^2, testSet$workingday)
test_MSE[3] = mean((testSet[, "registered"] - testX %*% model_3$coefficients)^2)
testX = cbind(1, testSet$temp, testSet$temp^2, testSet$workingday, testSet$clearday)
test_MSE[4] = mean((testSet[, "registered"] - testX %*% model_4$coefficients)^2)
testX = cbind(1, testSet$temp, testSet$temp^2, testSet$workingday, testSet$clearday, tes
test_MSE[5] = mean((testSet[, "registered"] - testX %*% model_5$coefficients)^2)
```

```
data.frame("Complexity" = 1:5, training_MSE = MSE, test_MSE = test_MSE) %>%
  ggplot() + theme_bw() +
  geom_line(aes(x = Complexity, y = training_MSE), color = 'maroon', size = 1) +
```

```
  geom_line(aes(x = Complexity, y = test_MSE), color = 'purple', size = 1) +
  labs(
    title = "MSE of the Bike Sharing Model w.r.t # of Regressors",
    caption = "Dataset: 2011 Washington Bike Sharing Data (Hadi Fanaee)"
  )
```



Still, the most complex model gives us the lowest holdout test MSE.

Problem 5 Bike Sharing: Cross-validation

```
# Create 10 folds
fold <- list()
tempdata <- data_day
i = 1
data_day <- data_day %>%
  mutate(fold = 0)
while (nrow(tempdata)>5){
```

```
  fold[[i]] <-   sample_n(tempdata, 36)
  tempdata <- tempdata %>%
    filter(!(instant %in% fold[[i]]$instant))
  i = i+1
}
for (i in 1:nrow(data_day)){
  for (j in 1:10){
    if (data_day$instant[i] %in% fold[[j]]$instant){
      data_day$fold[i] = j
    }
  }
}
```

```
MSE <- matrix(0, nrow = 5, ncol = 10)
for (i in 1:10){
  trainingSet = data_day %>% filter(fold != i)
  testSet = data_day %>% filter(fold == i)
  model_1 <- lm(registered ~ temp, data = trainingSet)
  model_2 <- lm(registered ~ temp + I(temp^2), data = trainingSet)
  model_3 <- lm(registered ~ temp + I(temp^2) + workingday, data = trainingSet)
  model_4 <- lm(registered ~ temp + I(temp^2) + workingday + clearday, data = trainingSe
  model_5 <- lm(registered ~ temp + I(temp^2) + workingday + clearday + temp*workingday,
  testX = cbind(1, testSet$temp)
  MSE[1, i] = mean((testSet[, "registered"] - testX %*% model_1$coefficients)^2)
  testX = cbind(1, testSet$temp, testSet$temp^2)
  MSE[2, i] = mean((testSet[, "registered"] - testX %*% model_2$coefficients)^2)
  testX = cbind(1, testSet$temp, testSet$temp^2, testSet$workingday)
  MSE[3, i] = mean((testSet[, "registered"] - testX %*% model_3$coefficients)^2)
  testX = cbind(1, testSet$temp, testSet$temp^2, testSet$workingday, testSet$clearday)
  MSE[4, i] = mean((testSet[, "registered"] - testX %*% model_4$coefficients)^2)
  testX = cbind(1, testSet$temp, testSet$temp^2, testSet$workingday, testSet$clearday, t
  MSE[5, i] = mean((testSet[, "registered"] - testX %*% model_5$coefficients)^2)
}
MSE
```

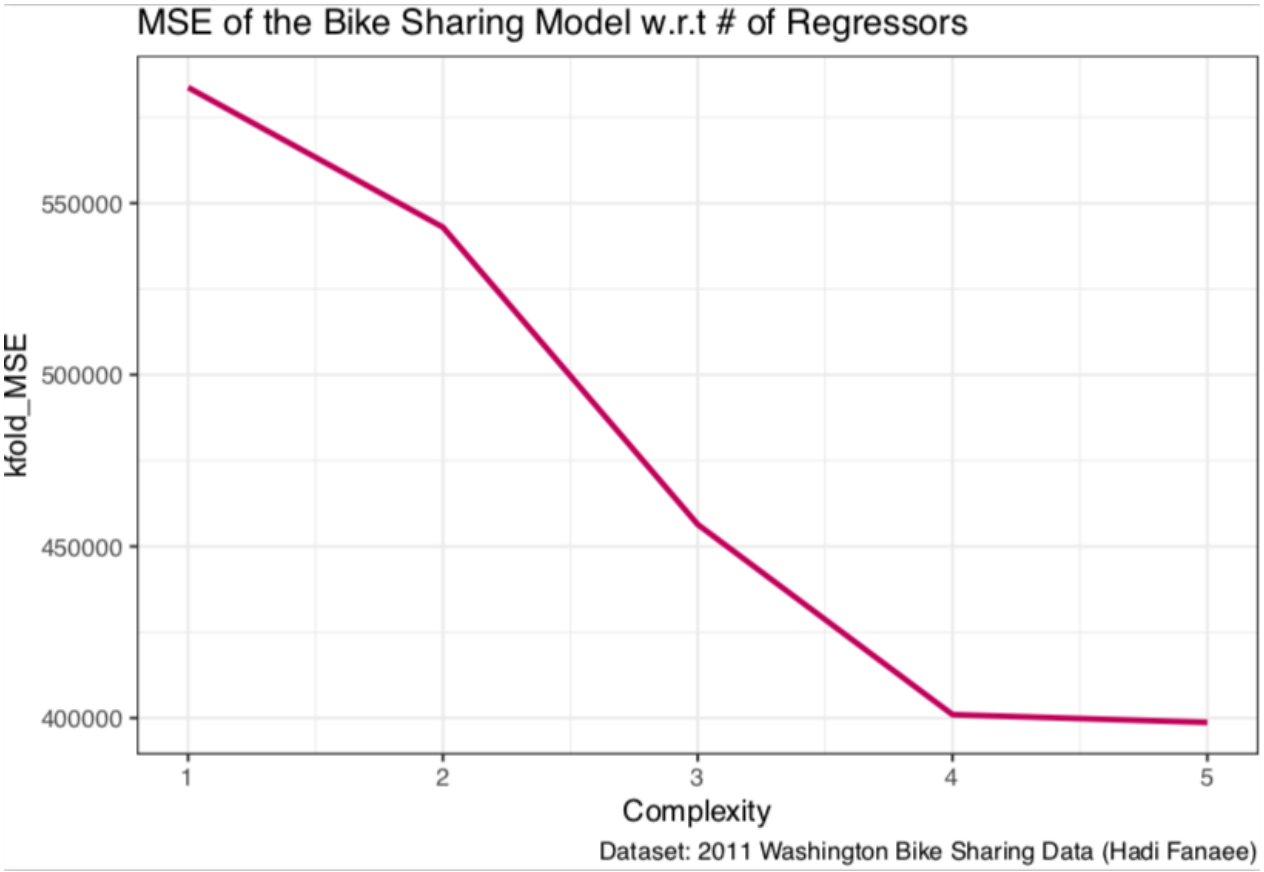
```
##          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]
## [1,] 758856.4 466304.8 509418.4 707323.6 677892.4 485899.7 521152.0
## [2,] 717297.5 469081.1 470483.5 638717.2 589642.4 438569.0 453312.3
## [3,] 648848.3 412820.9 305583.0 551156.8 436616.1 534552.3 349198.8
## [4,] 577146.7 359617.3 344021.8 391933.1 419637.6 460725.7 330256.5
## [5,] 555948.2 377214.5 344582.6 375084.5 409372.8 444682.4 349162.2
##          [,8]     [,9]    [,10]
```

```
## [1,] 443595.7 507893.4 758339.8
## [2,] 434179.4 461025.4 757344.9
## [3,] 314773.5 361993.6 648256.4
## [4,] 276733.8 342148.6 507739.0
## [5,] 277027.0 342049.1 512121.6
```

```
MSE_bar <- c()
for (i in 1:5){
  print(mean(MSE[i, ]))
  MSE_bar <- c(MSE_bar, mean(MSE[i, ]))
}
```

```
## [1] 583667.6
## [1] 542965.3
## [1] 456380
## [1] 400996
## [1] 398724.5
```

```
data.frame("Complexity" = 1:5, "kfold_MSE" = MSE_bar) %>%
  ggplot() + theme_bw() +
  geom_line(aes(x = Complexity, y = kfold_MSE), color = 'maroon', size = 1) +
  labs(
    title = "MSE of the Bike Sharing Model w.r.t # of Regressors",
    caption = "Dataset: 2011 Washington Bike Sharing Data (Hadi Fanaee)"
  )
```



Explanation

The most complex model gives us the best estimation (lowest k-fold MSE), but the improvement from model 4 to model 5 isn't so significant as the in-sample error. It shall be plausible because on one hand, the model captures more information within the sample as the complexity of it increases. On the other hand, it poses some risk of overfitting when the model is too complex as to capture the noise as patterns. In our case, the fisrt effect is more significant, leading to a downward-sloping k-fold MSE

Problem 6 BootStrap MSE

```
MSE <- matrix(0, nrow = 5, ncol = 200)
for (i in 1:200){
  trainingSet <- data_day[sample(365, 200, replace = TRUE), ]
  testSet <- data_day %>%
```

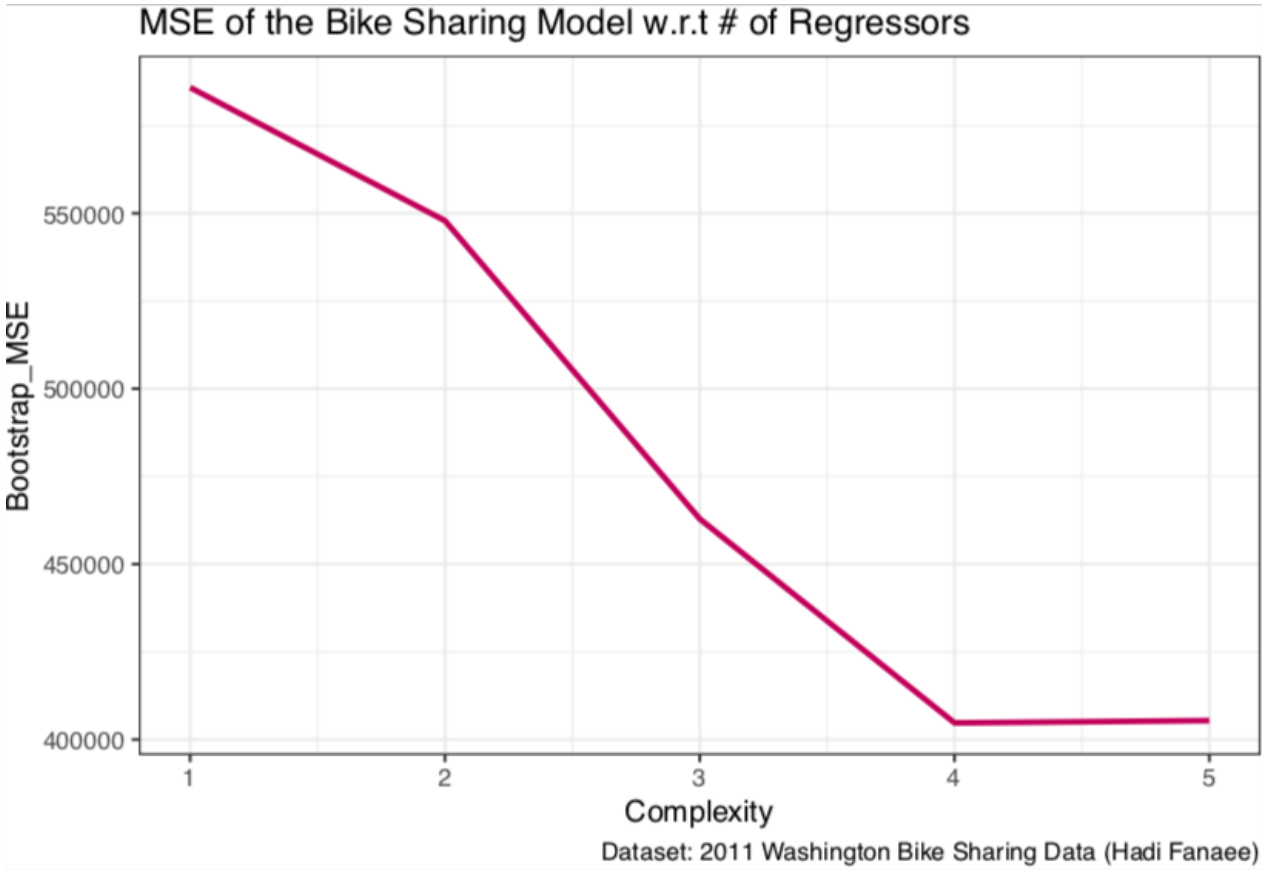
```
    filter(! instant %in% trainingSet$instant)
  model_1 <- lm(registered ~ temp, data = trainingSet)
  model_2 <- lm(registered ~ temp + I(temp^2), data = trainingSet)
  model_3 <- lm(registered ~ temp + I(temp^2) + workingday, data = trainingSet)
  model_4 <- lm(registered ~ temp + I(temp^2) + workingday + clearday, data = trainingSe
  model_5 <- lm(registered ~ temp + I(temp^2) + workingday + clearday + temp*workingday,
  testX = cbind(1, testSet$temp)
  MSE[1, i] = mean((testSet[, "registered"] - testX %*% model_1$coefficients)^2)
  testX = cbind(1, testSet$temp, testSet$temp^2)
  MSE[2, i] = mean((testSet[, "registered"] - testX %*% model_2$coefficients)^2)
  testX = cbind(1, testSet$temp, testSet$temp^2, testSet$workingday)
  MSE[3, i] = mean((testSet[, "registered"] - testX %*% model_3$coefficients)^2)
  testX = cbind(1, testSet$temp, testSet$temp^2, testSet$workingday, testSet$clearday)
  MSE[4, i] = mean((testSet[, "registered"] - testX %*% model_4$coefficients)^2)
  testX = cbind(1, testSet$temp, testSet$temp^2, testSet$workingday, testSet$clearday, t
  MSE[5, i] = mean((testSet[, "registered"] - testX %*% model_5$coefficients)^2)
}
MSE_bar <- c()
for (i in 1:5){
  print(mean(MSE[i, ]))
  MSE_bar <- c(MSE_bar, mean(MSE[i, ]))
}
```

```
## [1] 585855.7
## [1] 547896.3
## [1] 462879.2
## [1] 404732
## [1] 405384.7
```

```
MSE_bar
```

```
## [1] 585855.7 547896.3 462879.2 404732.0 405384.7
```

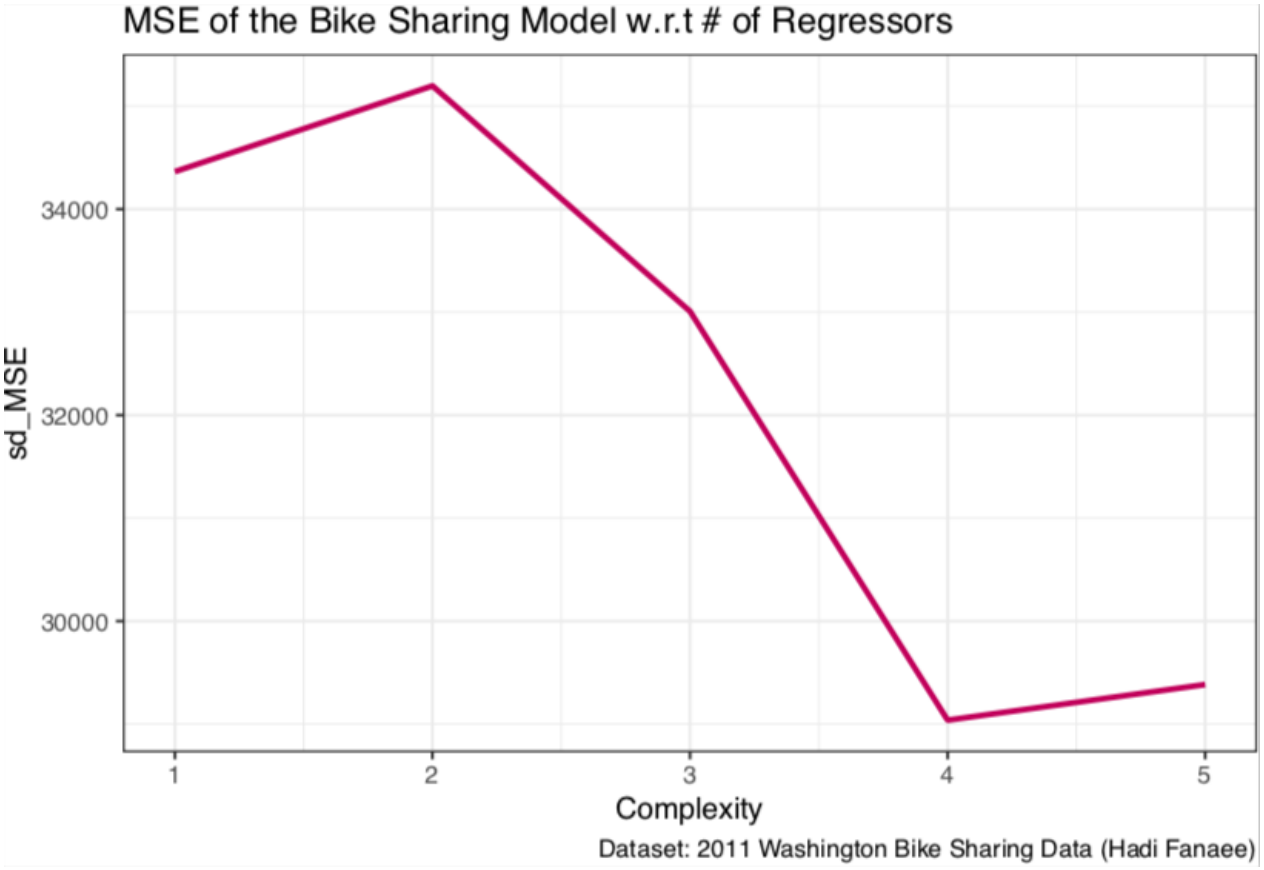
```
data.frame("Complexity" = 1:5, "Bootstrap_MSE" = MSE_bar) %>%
  ggplot() + theme_bw() +
  geom_line(aes(x = Complexity, y = Bootstrap_MSE), color = 'maroon', size = 1) +
  labs(
    title = "MSE of the Bike Sharing Model w.r.t # of Regressors",
    caption = "Dataset: 2011 Washington Bike Sharing Data (Hadi Fanaee)"
  )
```




```
MSE_var <- c()
for (i in 1:5){
  print(sd(MSE[i, ]))
  MSE_var <- c(MSE_var, sd(MSE[i, ]))
}
```

```
## [1] 34363.53
## [1] 35195.96
## [1] 33006.79
## [1] 29038.2
## [1] 29383.97
```

```
data.frame("Complexity" = 1:5, "sd_MSE" = MSE_var) %>%
  ggplot() + theme_bw() +
  geom_line(aes(x = Complexity, y = sd_MSE), color = 'maroon', size = 1) +
  labs(
    title = "MSE of the Bike Sharing Model w.r.t # of Regressors",
    caption = "Dataset: 2011 Washington Bike Sharing Data (Hadi Fanaee)"
  )
```




```
data.frame("Complexity" = 1:5, "MSE" = MSE[1,]) %>%
  ggplot() + theme_bw() +
  geom_histogram(aes(x = MSE),color = 'maroon', size = 1) +
  labs(
    title = "MSE of the Bike Sharing Model 1",
    caption = "Dataset: 2011 Washington Bike Sharing Data (Hadi Fanaee)"
  )
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
data.frame("Complexity" = 1:5, "MSE" = MSE[2,]) %>%
  ggplot() + theme_bw() +
  geom_histogram(aes(x = MSE),color = 'maroon', size = 1) +
  labs(
    title = "MSE of the Bike Sharing Model 2",
    caption = "Dataset: 2011 Washington Bike Sharing Data (Hadi Fanaee)"
  )
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

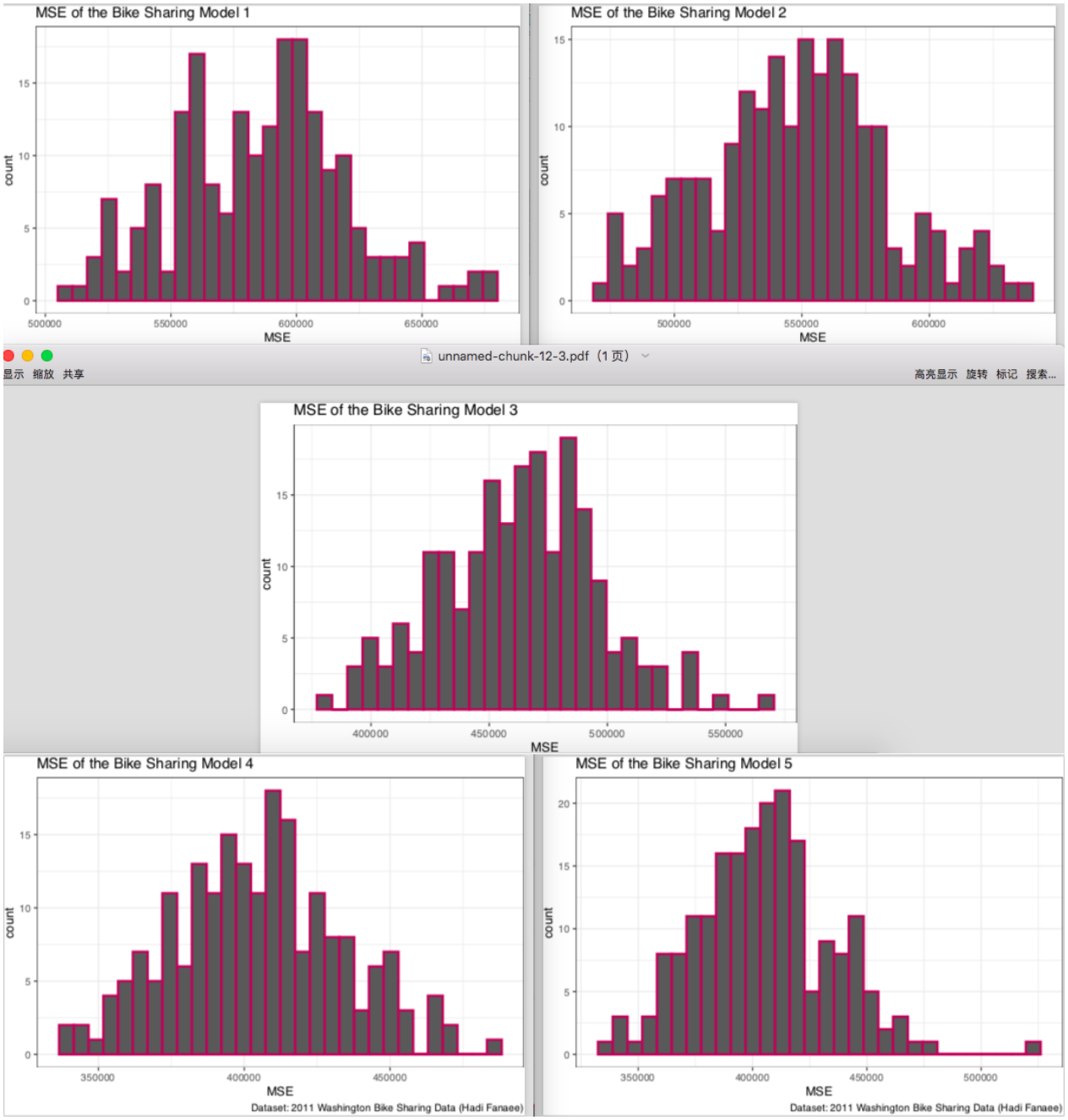
```
data.frame("Complexity" = 1:5, "MSE" = MSE[3,]) %>%
  ggplot() + theme_bw() +
  geom_histogram(aes(x = MSE),color = 'maroon', size = 1) +
  labs(
    title = "MSE of the Bike Sharing Model 3",
    caption = "Dataset: 2011 Washington Bike Sharing Data (Hadi Fanaee)"
  )
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
data.frame("Complexity" = 1:5, "MSE" = MSE[4,]) %>%
  ggplot() + theme_bw() +
  geom_histogram(aes(x = MSE),color = 'maroon', size = 1) +
  labs(
    title = "MSE of the Bike Sharing Model 4",
    caption = "Dataset: 2011 Washington Bike Sharing Data (Hadi Fanaee)"
  )
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
data.frame("Complexity" = 1:5, "MSE" = MSE[5,]) %>%
  ggplot() + theme_bw() +
  geom_histogram(aes(x = MSE),color = 'maroon', size = 1) +
  labs(
    title = "MSE of the Bike Sharing Model 5",
    caption = "Dataset: 2011 Washington Bike Sharing Data (Hadi Fanaee)"
  )
```



Explanation

The 4-th model gives us the best estimation (lowest mean bootstrap MSE), indicating that there's a problem of overfitting moving forward from model 4 to model 5. It shall be plausible because on one hand, the model captures more information within the sample as the complexity of it increases. On the other hand, it poses some risk of overfitting when the model is too

complex as to capture the noise as patterns. In our case, the second effect is more significant, leading to a U-shape bootstrap MSE.

The bootstrap estimate is not so reliable if you do not carry it out repeatedly, as the variance of MSE is highly related to the sample. In addition to comparing the mean of the bootstrap MSE, it'll be more credible if we statistically compare the whole distribution of bootstrap MSE in the different models. In that sense, we can still make it reliable.