ISyE 7406: Data Mining and Statistical Learning **(Due: 01/16/2020)**

Homework Assignment #1

Instructor: Yajun, Mei *Name:* Chen-Yang(Jim), Liu *GTID:* 903450732

Problem 1 (? points)

(a) There are 6 observations, and we can write the observed data in the matrix form $Y_{n\times 1} = X_{n\times p}\ \beta_{p\times 1}$ $+\epsilon_{n\times 1}$ with n = 6 and p = 2.

In this problem, simple linear regression model is chosen since there is only one variable - weight. There are two kinds of balls, A and B. We could see below dataframe to know Y which is an outcome after A and B are estimated. Furthermore, there are random errors every measurement and we could clearly see that A + B is not equal to A measured weight plus B measured weight. As a result, we assume that random errors are identically and independently distributed.

And, the model is $Y_{n\times 1} = X_{n\times p}\ \beta_{p\times 1}$ $+\epsilon_{n\times 1}$ where n = 6, p = 2, $\epsilon \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2)$

1. estimate the weights of balls A and B

   From the assumption of linear models, the estimate formula is $\hat{\beta} = (X^TX)^{-1}X^TY$ Therefore, from the below R code output, we could see that the estimate of A and B are 2.3333 and 1.8333, respectively.

```
data_7406 <- data.frame(Y = c(2, 1, 4, 2, 2, 5),
                        A = c(1, 0, 1, 1, 0, 1),
                        B = c(0, 1, 1, 0, 1, 1))
print(data_7406)

##   Y A B
## 1 2 1 0
## 2 1 0 1
## 3 4 1 1
## 4 2 1 0
## 5 2 0 1
## 6 5 1 1

lmod <- lm(Y ~ 0 + A + B, data = data_7406)
summary(lmod)
##
## Call:
## lm(formula = Y ~ 0 + A + B, data = data_7406)
##
## Residuals:
##       1       2       3       4       5       6
## -0.3333 -0.8333 -0.1667 -0.3333  0.1667  0.8333
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## A   2.3333     0.3727   6.261  0.00332 **
## B   1.8333     0.3727   4.919  0.00793 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6455 on 4 degrees of freedom
## Multiple R-squared:  0.9691, Adjusted R-squared:  0.9537
## F-statistic:  62.8 on 2 and 4 DF,  p-value: 0.0009526
```

2. find a 70% confidence interval on the weight of ball A

   The confidence interval is $\beta_A \pm t_{0.3/2,4}\hat{\sigma}\sqrt{d_1}$ where d_1 is diagonal of $(X^TX)^{-1}$ which is the same as $\beta_A \pm t_{0.3/2,4}$S.E($\hat{\beta_A}$) Thus, from the below R code, the 70% confidence interval of Ball A is $[1.889948, 2.776652]$

   ```
   qt(.85, 6 - 2)
   ## [1] 1.189567
   2.3333 + c(-1, 1) * 1.189567 * 0.3727
   ## [1] 1.889948 2.776652
   ```

3. Suppose the student plans to measure the weight of ball A one more time. Find a 70% prediction interval on the new observed weight of ball A.

   Prediction intervals is different from the confidence interval. The formula of prediction intervals is $\beta_A \pm t_{0.3/2,4}\hat{\sigma}\sqrt{1 + x_{new}^T(X^TX)^{-1}x_{new}}$. The reason that there is a 1 in square root is that we consider future noise. And as what we expect, when we predict one value without previous data, we naturally consider wider intervals at this point.

   ```
   x <- data.matrix(data_7406[2:3])
   qt(.85, 6 - 2)
   ## [1] 1.189567
   x <- data.matrix(data_7406[2:3])
   # transpose x * x to
   diagonal <- t(x) %*% x
   print(solve(diagonal))
   ##            A          B
   ## A  0.3333333 -0.1666667
   ## B -0.1666667  0.3333333
   ##sigma_square: RSS/4
   print(anova(lmod))
   ## Analysis of Variance Table
   ##
   ## Response: Y
   ##           Df Sum Sq Mean Sq F value    Pr(>F)
   ## A          1 42.250  42.250   101.4 0.0005471 ***
   ## B          1 10.083  10.083    24.2 0.0079331 **
   ## Residuals  4  1.667   0.417
   ## ---
   ## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
   RSS <- 1.667
   sigma_hat <- sqrt(RSS / 4)
   #new data
   x_new <- c(1, 0)
   x_new_square <- t(x_new) %*% x_new

   #prediction interval
   2.3333 + c(-1, 1) * 1.189567 * sigma_hat * sqrt(1 + 0.3333333 * 1)
   ## [1] 1.44656 3.22004
   ```

   From the above code, the 70% prediction interval of Ball A is $[1.44656, 3.22004]$

(b) Repeat part (a) by writing the "new observed average weights" in the matrix form $Y_{n\times 1} = X_{n\times p}\ \beta_{p\times 1}$ $+\epsilon_{n\times 1}$ with n = 3 and p = 2.

The second problem is different from the first problem. For now, we average the data to derive outcomes. But, the formula of estimate, confidence intervals and prediction confidence intervals are the same.

1. estimate the weights of balls A and B
   The estimate of A and B are 2.3333 and 1.83333, respectively.

   ```
   data_new <- data.frame(Y_new = c(2, 1.5, 4.5),
   ```

```
                    A_new = c(1, 0, 1),
                    B_new = c(0, 1, 1))
print(data_new)
##     Y_new A_new B_new
## 1    2.0      1     0
## 2    1.5      0     1
## 3    4.5      1     1

lmod_new <- lm(Y_new ~ 0 + A_new + B_new, data = data_new)
summary(lmod_new)
##
## Call:
## lm(formula = Y_new ~ 0 + A_new + B_new, data = data_new)
##
## Residuals:
##       1       2       3
## -0.3333 -0.3333  0.3333
##
## Coefficients:
##       Estimate Std. Error t value Pr(>|t|)
## A_new   2.3333     0.4714   4.950    0.127
## B_new   1.8333     0.4714   3.889    0.160
##
## Residual standard error: 0.5774 on 1 degrees of freedom
## Multiple R-squared:  0.9874, Adjusted R-squared:  0.9623
## F-statistic: 39.25 on 2 and 1 DF,  p-value: 0.1122
```

2. find a 70% confidence interval on the weight of ball A

   The confidence intervals of $\hat{\beta_A}$ is $[1.408125, 3.258475]$

```
qt(.85, 3 - 2)
## [1] 1.962611
2.3333 + c(-1, 1) * 1.962611 * 0.4714
## [1] 1.408125 3.258475
```

3. Suppose the student plans to measure the weight of ball A one more time. Find a 70% prediction interval on the new observed weight of ball A.

   The prediction interval of $\hat{\beta_A}$ is $[0.2645263, 4.4020737]$

```
qt(.85, 3 - 2)
## [1] 1.962611
x_b <- data.matrix(data_new[2:3])
# transpose x * x to
diagonal_b <- t(x_b) %*% x_b
print(solve(diagonal_b))
##            A_new      B_new
## A_new  0.6666667 -0.3333333
## B_new -0.3333333  0.6666667
#sigma_square: RSS/4
print(anova(lmod_new))
## Analysis of Variance Table
##
## Response: Y_new
##           Df  Sum Sq Mean Sq F value  Pr(>F)
## A_new      1 21.1250 21.1250  63.375 0.07955 .
## B_new      1  5.0417  5.0417  15.125 0.16022
## Residuals  1  0.3333  0.3333
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
RSS_b <- 0.6666667
sigma_hat_b <- sqrt(RSS_b / (3 - 2))
#new data
x_new_b <- c(1, 0)
x_new_square_b <- t(x_new_b) %*% x_new_b

#prediction interval
2.3333 + c(-1, 1) * 1.962611 * sigma_hat_b * sqrt(1 + 0.6666667 * 1)
## [1] 0.2645263 4.4020737
```

(c) Compare your results in (a) and (b)

Clearly, there are three dimensions that we could compare:

- Estimate: The estimate β_A of (a) and (b) is the same. β_A is 2.3333.
- Confidence Intervals: Since there are different numbers of observations that lead to different t-distribution. And, from (a) and (b), the confidence interval of (b)(≈ 1.84) is wider than one of (a)(≈ 0.89).
- Prediction Confidence Intervals: It is true that the prediction confidence interval of (b) (≈ 4.14) is wider than one of (a) (≈ 1.78)

Problem 2 (? points)

Consider a simple linear regression model Y = β_0 + β_1x + ϵ. Suppose that we choose m different values of the independent variables x_i's, and each choice of x_i is duplicated, yielding k independent observations $Y_{i1}, Y_{i2}, ..., Y_{ik}$ Is it true that the least squares estimates of the intercept and slope can be found by doing a regression of the mean responses, $Y_i = (Y_{i1} + Y_{i2} + ... + Y_{ik})/k$, on the x_i's? Why or why not? Explain.

From hints, we only consider $k = 2$ case, and then we will separately discuss estimate of β_0 and β_1 and their confidence intervals.

- Estimate of β_0 and β_1:

$$\frac{dSS_{err,1}}{d\beta_0} = \Sigma_{i=1}^m 2(y_{i1} - (\beta_0 + \beta_1 x_i))(-1) + \Sigma_{i=1}^m 2(y_{i2} - (\beta_0 + \beta_1 x_i))(-1) = 0 \tag{1}$$

$$\frac{dSS_{err,1}}{d\beta_1} = \Sigma_{i=1}^m 2(y_{i1} - (\beta_0 + \beta_1 x_i))(-x_i) + \Sigma_{i=1}^m 2(y_{i2} - (\beta_0 + \beta_1 x_i))(-x_i) = 0 \tag{2}$$

Then, we take derivative of $SS_{err,2}$ to get the same equation as (1) and (2)

$$\begin{aligned}\frac{dSS_{err,2}}{d\beta_0} &= \Sigma_{i=1}^m(\bar{y}_i - (\beta_0 + \beta_1 x_i))(-1) = 0\\ &= (\frac{Y_{11} + Y_{22}}{2} + ... + \frac{Y_{m1} + Y_{m2}}{2} - m\beta_0 + \Sigma_{i=1}^m \beta_1 x_i)(-1)\\ &= (\frac{Y_{11} + ... + Y_{m1}}{2} + \frac{Y_{12} + ... + Y_{m2}}{2} - m\beta_0 + \Sigma_{i=1}^m \beta_1 x_i)(-1)\\ &= (\frac{1}{2}\Sigma_{i=1}^m (y_{i1} + y_{i2} - (2\beta_0 + 2\beta_1 x_i)))(-1)\\ &= (\frac{1}{2}\Sigma_{i=1}^m (y_{i1} - (\beta_0 + \beta_1 x_i)))(-1) + (\frac{1}{2}\Sigma_{i=1}^m (y_{i2} - (\beta_0 + \beta_1 x_i)))(-1) = 0\\ &= (\Sigma_{i=1}^m (y_{i1} - (\beta_0 + \beta_1 x_i)))(-1) + (\Sigma_{i=1}^m (y_{i2} - (\beta_0 + \beta_1 x_i)))(-1) = 0\\ &= \Sigma_{i=1}^m 2(y_{i1} - (\beta_0 + \beta_1 x_i))(-1) + \Sigma_{i=1}^m 2(y_{i2} - (\beta_0 + \beta_1 x_i))(-1) = 0\end{aligned}$$

Since coefficients do not matter when solve (1) equation, we can multiply any number we want. Now, we take derivative of $SS_{err,2}$ with respect to β_1.

$$\frac{dSS_{err,2}}{d\beta_1} = \Sigma_{i=1}^m 2(\bar{y}_i - (\beta_0 + \beta_1 x_i))(-x_i) = 0$$

From the above equation, we transform the equation:

$$\begin{aligned}&= 2((-x_1)\frac{Y_{11} + Y_{12}}{2} + ... + (-x_m)\frac{Y_{m1} + Y_{m2}}{2} - \Sigma_{i=1}^m(-x_i(\beta_0 + \beta_1 x_i))\\ &= 2\Sigma_{i=1}^m(-x_i)(\frac{(Y_{11} + ... + Y_{m1})}{2} + \frac{(Y_{12} + ... + Y_{m2})}{2}) - \Sigma_{i=1}^m(-x_i(\beta_0 + \beta_1 x_i))\\ &= \Sigma_{i=1}^m(-x_i)((Y_{11} + ... + Y_{m1}) + (Y_{12} + ... + Y_{m2})) - 2\Sigma_{i=1}^m(-x_i(\beta_0 + \beta_1 x_i))\\ &= \Sigma_{i=1}^m(y_{i1} - (\beta_0 + \beta_1 x_i))(-x_i) + \Sigma_{i=1}^m(y_{i2} - (\beta_0 + \beta_1 x_i))(-x_i) = 0\end{aligned}$$

Similarly, after taking derivative, we could get the identical equation as (2). Therefore, $\hat{\beta}_0$ and $\hat{\beta}_1$ are the same as ones from taking derivative of $SS_{err,2}$.

- Confidence Intervals and Prediction Intervals: It is true that different sample size leads to different t-distribution. But, $\hat{\sigma}$ is RSS/2m-2 for 2m observations and RSS/m-2 for averaged observations. In this problem, with more and more observations observed, $\hat{\sigma}$ becomes smaller, so that's why $\hat{\sigma}$ of averaged observations has larger values than $\hat{\sigma}$ of 2m observations. Naturally, we have high uncertainty of the true mean from just m observations, so it is better to have more error there and less on the more confident estimate.

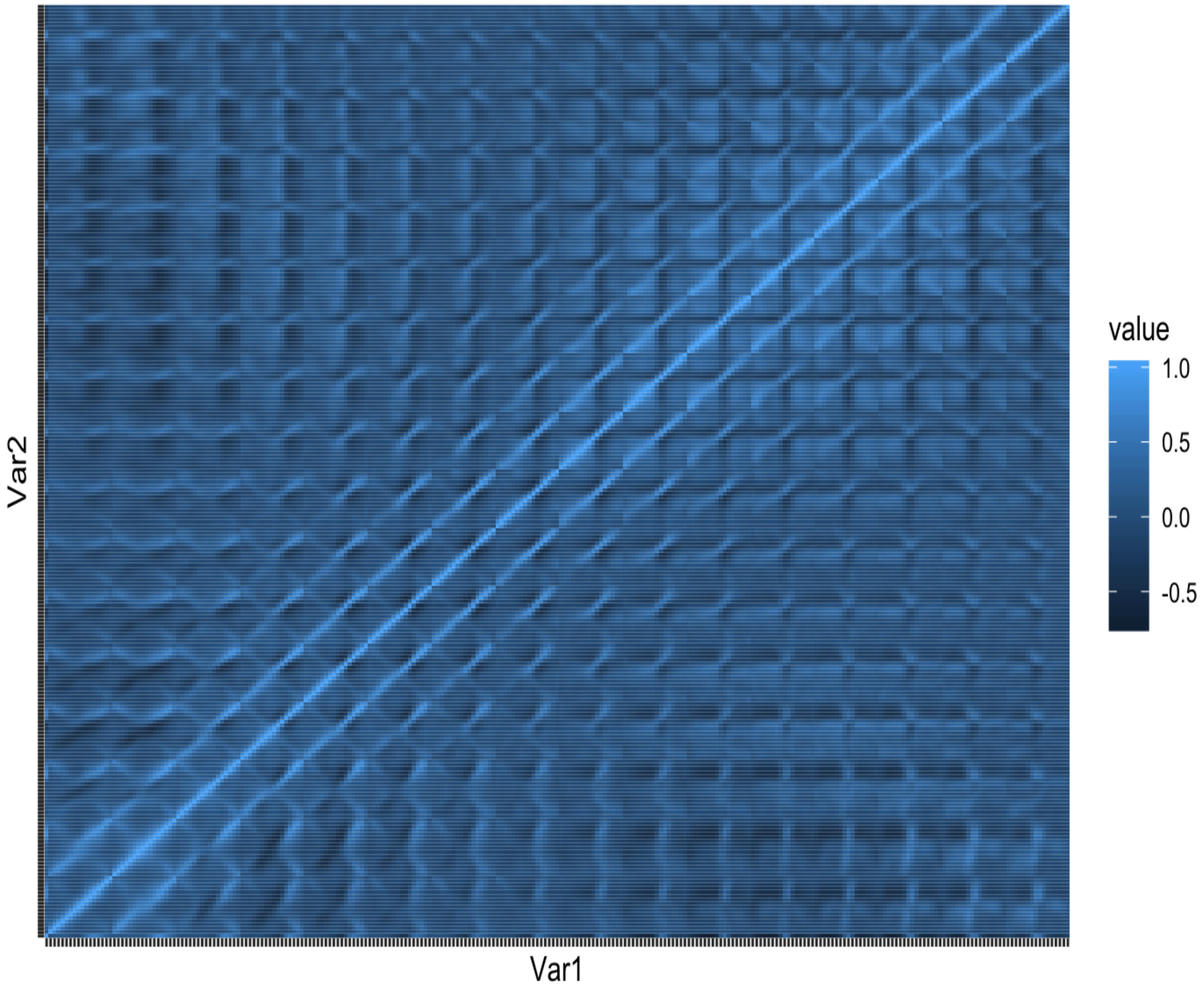
Problem 3 (? points)

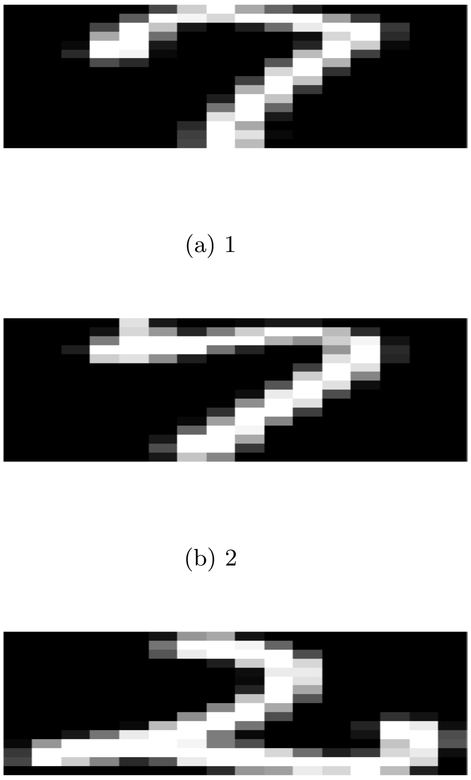
(a) Read the data

```
ziptrain <- read.table(file="http://www.isye.gatech.edu/~ymei/7406/Handouts
/zip.train.csv",sep = ",")
ziptrain27 <- subset(ziptrain, ziptrain[,1]==2 | ziptrain[,1]==7)
ziptrain27
```

(b) Exploratory Data Analysis

```
    ziptrain27Cor_ <- melt(ziptrain27Cor)
    head(ziptrain27Cor_)
    ggplot(data = ziptrain27Cor_, aes(x=Var1, y=Var2, fill=value))
   + geom_tile()
```



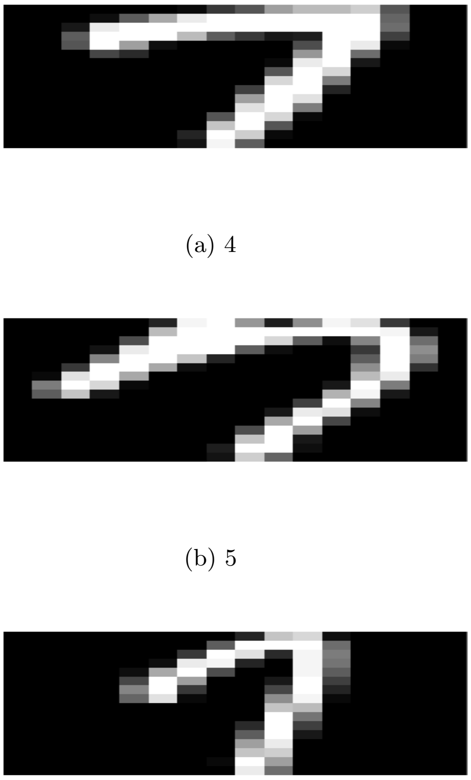


(a) 1

(b) 2

(c) 3

Figure 1: Top left

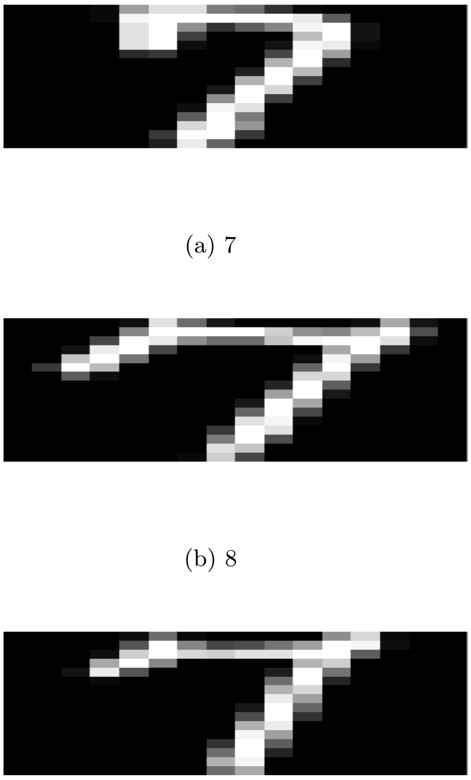


(a) 4

(b) 5

(c) 6

Figure 2: Top right

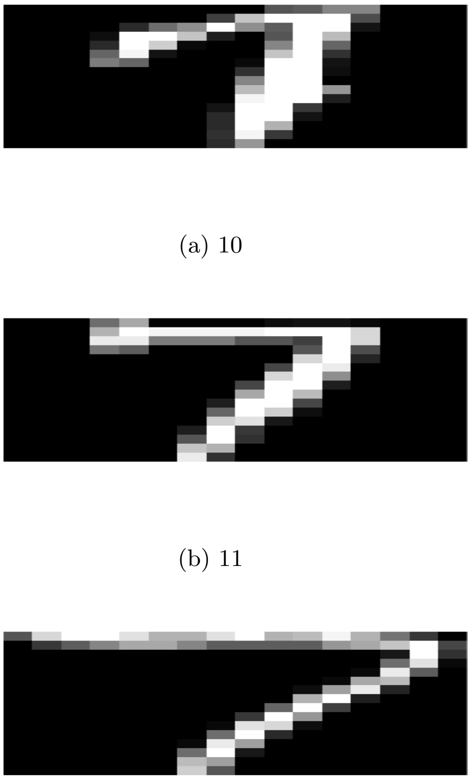


(a) 7

(b) 8

(c) 9

Figure 3: Bottom left



(a) 10

(b) 11

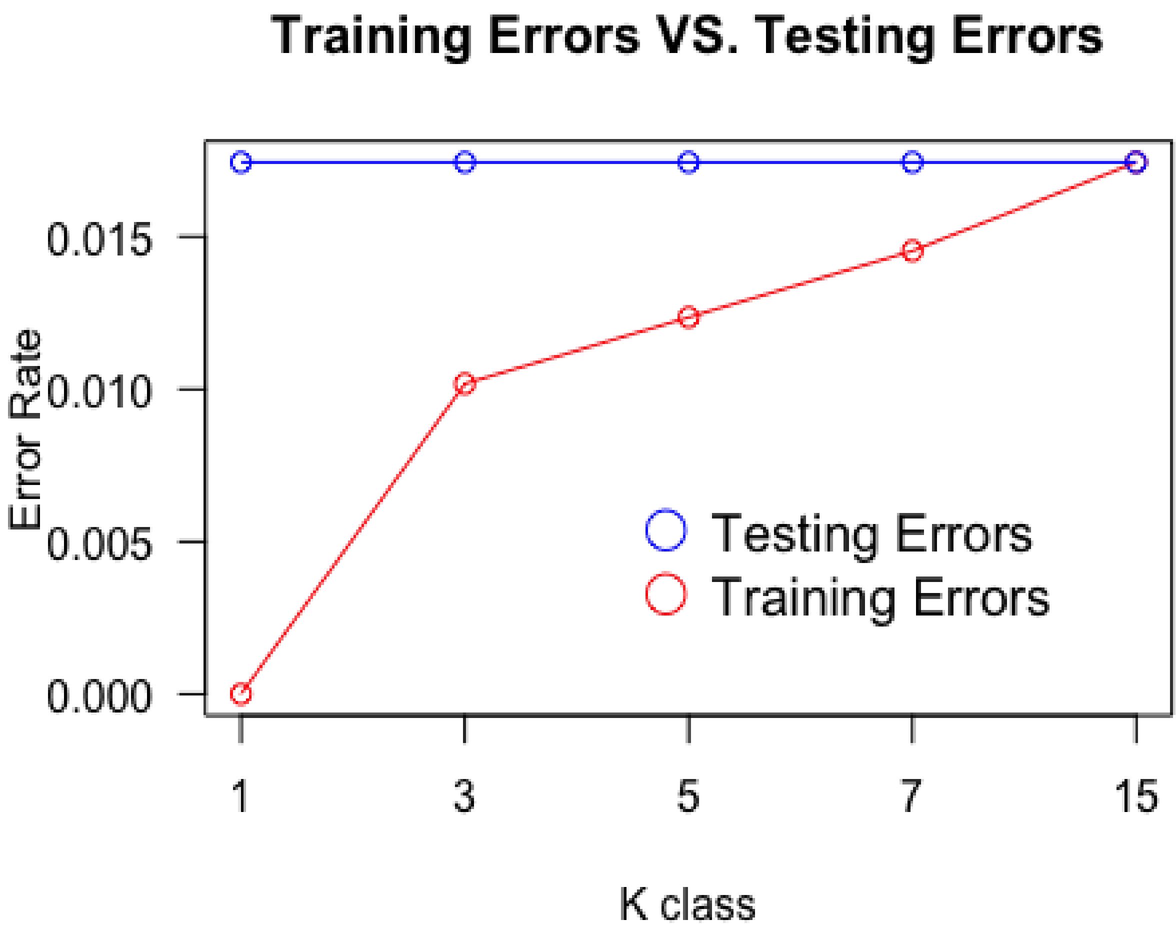
(c) 12

Figure 4: Bottom right

(c) Linear Regression VS. KNN

```
mod1 <- lm( V1 ~ . , data= ziptrain27);
pred1.train <- predict.lm(mod1, ziptrain27[,-1]);
y1pred.train <- 2 + 5*(pred1.train >= 4.5);
mean( y1pred.train != ziptrain27[,1])
## [1] 0.0007267442
library(class);
xnew <- ziptrain27[,-1];
z <- c()
for (i in c(1, 3, 5, 7, 15)) {
  ypred2.train <- knn(ziptrain27[,-1], xnew, cl = ziptrain27[,1], k = i);
  z <- c(z, mean(ypred2.train != ziptrain27[,1]))
}
z
## [1] 0.00000000 0.01017442 0.01235465 0.01453488 0.01744186
```

(d) (4) Let us consider the testing data set, and derive the testing errors of each classification rule in (3).



(e) Summary

The goal of this problem is to investigate whether handwritten 2 and handwritten 7 have special features. In order to build a machine model, we would like to figure out which one is suitable for the handwritten dataset? Linear Regression or KNN.

First, exploratory data analysis is performed. There are two graphs we try to figure out. Since 2 and 7 are really similar, we want to know the correlation between these two digits. Therefore, the heatmap is provided to see which vector has higer correlation and which one has lower one. Clearly, the two vector are close that there is higher correlation. On the other hand, the two vectore are too far that there is lower correlation. Furthermore, some of images are selected to see whether they are clearly seen as 2 and 7. Twelve images are shown that 2 and 7 sometimes are hard to tell difference. After exploring data, we switch our focus on methods to train this dataset. From what we learn in linear regression, the output of this dataset is 2 and 7. If we run linear regression, we might get 4.5 value, but it does not make sense at all. We cannot find any handwritten digit which is 4.5. As a result, classification problem should be addressed by the KNN method. We choose KNN method to know the error rate which gives us an overview how our model is good or not and which parameter we should select. From the Training Errors VS. Testing Errors picture, each number of K does not lead to large testing errors, but as for training errors, k = 3 could be a good choice.