Assignment 1

Statistics and Data Science 365/565

Due: Monday February 3rd 11:59pm

Problem 1: Two views of linear regression (10 points)

Problem 1 View 1

$$\hat{\beta} = argmin \mid\mid Y - X\beta \mid\mid^2$$

where $\beta \in \mathbf{R}^P$

To minimize this, we scan solve the partial derivative by $\beta = 0$

$$\frac{\partial \mid\mid Y - X\beta \mid\mid^2}{\partial \beta} = X^T(Y - X\beta) = 0$$

$$X^TY - X^TX\beta = 0$$

$$\hat{\beta} = (X^TX)^{-1}X^TY$$

Problem 1 View 2

we have the density of y as

$$f(y) = \frac{1}{\sqrt{|2\pi\Sigma|}} \exp\left(-\frac{1}{2}(y - X\beta)^T\Sigma^{-1}(y - X\beta)\right)$$

then we consider the log density of y, use mle, maximized the likehood of β

$$L(\beta) = log(f(x_1)f(x_2)\dots f(x_n)) = \Sigma_{i=1}^n\left(-\frac{1}{2}(y_i - x_i\beta)^T\Sigma^{-1}(y_i - x_i\beta)\right) - log(\sqrt{|2\pi\Sigma|})$$

$$= \Sigma_{i=1}^n\left(-\frac{1}{2}(y_i - x_i\beta)^T(y_i - x_i\beta)\right) - log(\sqrt{|2\pi\sigma|})$$

$$\frac{\partial L(\beta)}{\partial \beta} = -\frac{1}{2}(\Sigma_{i=1}^n(y_i - x_i\beta)x_i) = 0$$

$$\hat{\beta} = \frac{\Sigma_{i=1}^{n} x_i y_i}{\Sigma_{i=1}^{n} x_i^2}$$

the β solved in this way is exatly same as view 1 since $\Sigma_{i=1}^{n}\left(-\frac{1}{2}(y_i - x_i\beta)^T(y_i - x_i\beta)\right)$ can also be written as $\left(-\frac{1}{2}(Y - X\beta)^T(Y - X\beta)\right)$ and $\Sigma_{i=1}^{n} x_i y_i = X^T Y$ $\Sigma_{i=1}^{n} x_i^2 = X^T X$ # Problem 2: Linear regression and classification (30 points)

Problem 2 Part a.

The summary statistics, the scatter plot, box plot shows as follows, we can find some extreme values and outliers

- Strange stations number distributions

  the density function of the station numbers have two peak, with 75% of days the total number of the stations below 331, while the 20% days the total number of the stations within the range of $424 \sim 475$

citi_train
11 Variables 1001 Observations

trips

n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
1001	0	987	1	25022	13187	5401	8645	15545	26629	34208	38485	42170

`lowest :   876  1107  1144  1214  1459, highest: 50285 50705 51179 51368 52706`

n_stations

n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
1001	0	91	0.996	354.2	48.7	319	321	324	327	331	465	470

`lowest : 292 298 303 307 311, highest: 471 472 473 474 475`

PRCP

n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
1001	0	103	0.685	0.1193	0.2104	0.00	0.00	0.00	0.00	0.04	0.36	0.73

`lowest : 0.00 0.01 0.02 0.03 0.04, highest: 1.95 1.98 2.10 2.54 4.97`

SNWD

n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
1001	0	21	0.321	0.8932	1.647	0.0	0.0	0.0	0.0	0.0	2.0	7.9

`lowest :  0.0  1.0  1.2  2.0  3.1, highest: 15.0 16.1 16.9 18.1 18.9`

SNOW

n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
1001	0	30	0.129	0.1131	0.2216	0	0	0	0	0	0	0

`lowest :  0.0  0.1  0.2  0.3  0.4, highest:  5.5  7.5  8.0  9.5 11.0`

TMAX

n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
1001	0	83	1	62.62	21.9	31	36	46	64	80	86	89

`lowest : 15 17 18 19 20, highest: 94 95 96 97 98`

TMIN

n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
1001	0	82	1	48.12	20.65	17	23	35	50	64	71	73

`lowest : -1  2  3  4  5, highest: 78 79 81 82 83`

AWND

n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
1001	0	59	0.999	-24.61	62.37	2.2	2.7	3.8	4.9	6.7	8.5	9.8

Value	-10000	0	20
Frequency	3	948	50
Proportion	0.003	0.947	0.050

Outliers

- **PRCP SNWD SNOW**

Rainfall and snowfall belong to the small probability event, most days of our data there will be no snow or rain, thus the distribution of PRCP SNWD SNOW shoule be a skewed distribution naturally. However, we still need to discriminate the outliers. To acquire rigorious decision, we refered to the **nyuweather** and distribution of *non zero PRCP SNWD SNOW*,the precipitation should be smaller than 2, SNOW should be smaller than 8.0, outliers imputed with the average precipitation within the month

- **WIND**

from the descriptive statistics and the boxplot, we find there is an observation has a -10000 windspeed, which contradicts the definiation of the windspeed, wind doesn't have such strong seasonality, thus we remove this single outlier. After removing the unreasonable value, we recheck the outlier, eliminate the 24 outliers(with windspeed >10).

correlations within predictors

After manipulating the outlier, we final start to analyze the realtionship within predictors

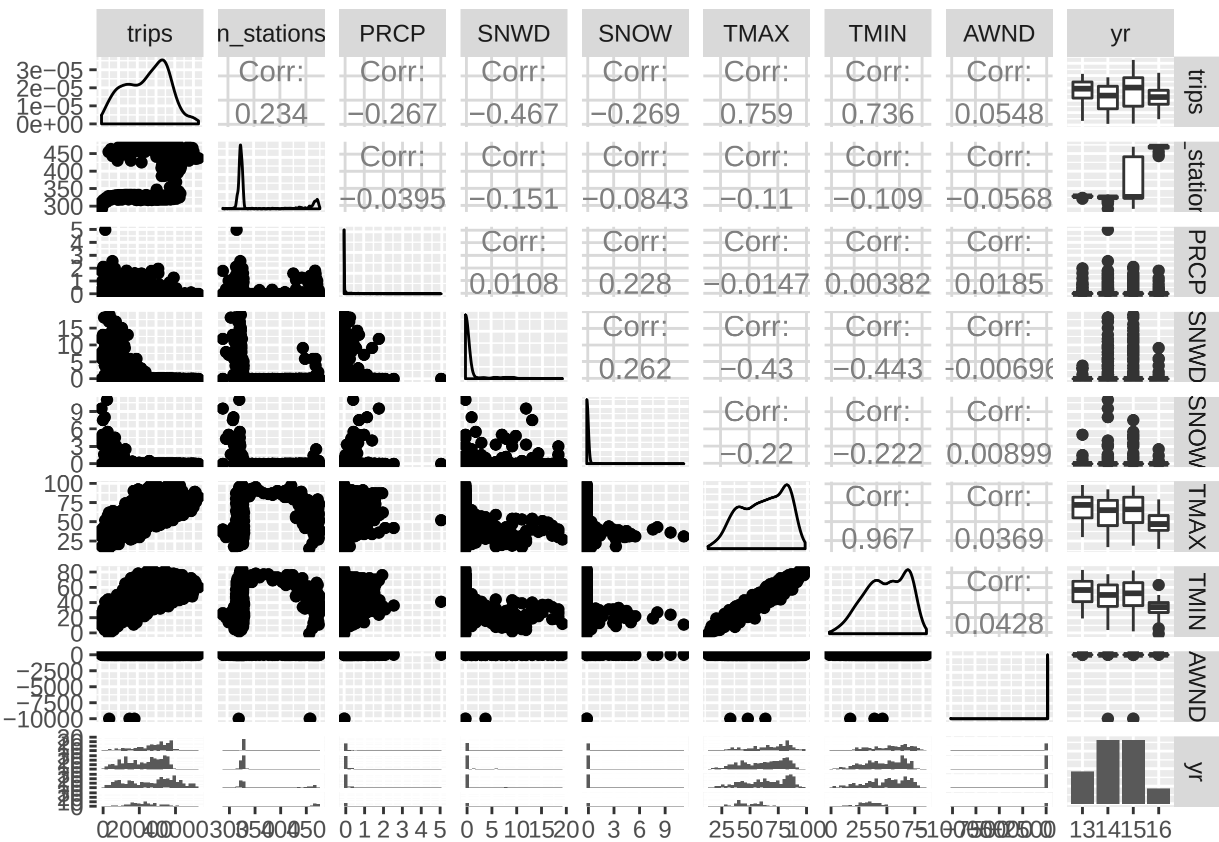
- **TMAX is closely related to the TMIN**

The maximum temperature for the day is postively correlated to the minimum temperature for the day, we can conclude both from common sense and the scatter plot($r = 0.967$), in the linear regression model we might need to consider removing multicollinearity of the two predictor, after reexaming the data we find out that during the sepetmber of 2015, there is a increase in the number of the stations, this might due to the citi bike is expanding its scale at that time. Thus, even though in boxplot, many observations are marked as the outliers, we cann't change its valuee or remove them.

- **Other strong relationships**

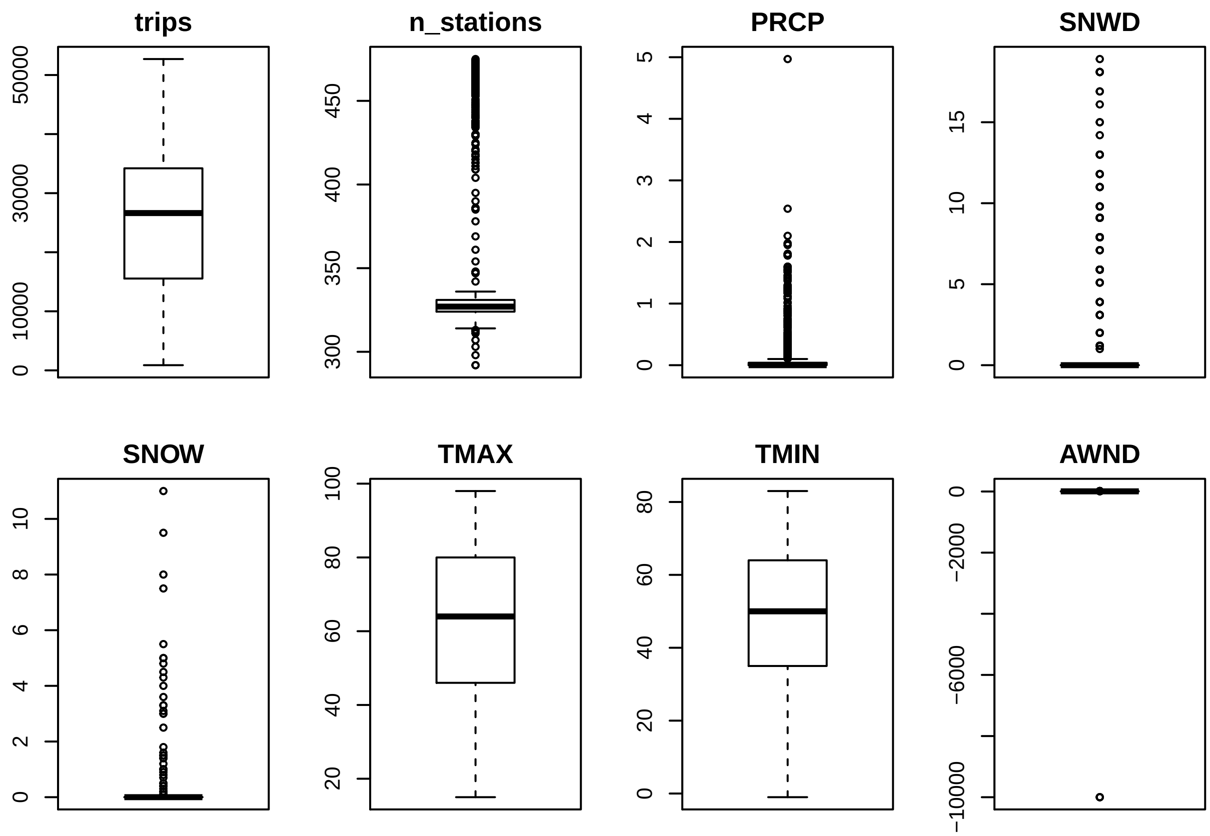
Windspeed $AWND$ has negative linear relationship with TMAX($r = -.502$) and TMIN($r = -.491$); Snowfall also has negative linear relationship with TMAX($r = -.43$) and TMIN($r = -.441$)

```r
library(pacman)
p_load(dplyr,Hmisc,GGally,FNN,stringr,MASS,car,knitr,polynom)
citi_train <- read.csv("citibike_train.csv")
citi_test <- read.csv("citibike_test.csv")
weather <- read.csv("weather.csv")


citi_train <- citi_train%>%left_join(weather,by = 'date')%>%
  mutate(yr =str_sub(date,-2,-1))%>%dplyr::select(-date)
citi_test <- citi_test%>%left_join(weather,by = 'date')%>%
  dplyr::select(-date)
#describe(citi_train)
ggpairs(citi_train%>%dplyr::select(-c(holiday,month,dayofweek)))
```




```r
par(mfrow = c(2,4))
par(mar = c(2, 2, 2, 2))
list_numeric = c(1:2,6:11)
for(i in list_numeric){
```

```
  boxplot(citi_train[,i],main=(names(citi_train)[i]))
}
```




```
#test whether the data is reasonable
#return 0
#dim(citi_train%>%filter(TMAX<=TMIN))[1]


#remove the unreasonable wind speed
citi_train <- citi_train%>%
  filter(AWND>=0 & AWND<=11.2 &SNWD<18.9)%>%group_by(month,yr)%>%
  mutate(PRCP = case_when(
    (PRCP<=2) ~ PRCP,
    (PRCP>2) ~ median(PRCP)
  ),
  SNOW = case_when(
    (SNOW<=8) ~ SNOW,
    (SNOW>8) ~ median(SNOW))
```

```r
  )%>%ungroup()%>%dplyr::select(-yr)


citi_test <-citi_test%>%mutate(AWND = case_when(
  AWND>=0 ~ AWND,
  TRUE ~ median(AWND)
))
```

Problem 2 Part b.

below is the model with all the possible variables

```r
#model with all the variables
fit <- lm(trips~.,data = citi_train)
summary(fit)
```

```
##
## Call:
## lm(formula = trips ~ ., data = citi_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18001.1  -2199.1    238.5   2476.9  15191.5
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -14232.299   1462.055  -9.734  < 2e-16 ***
## n_stations         71.668      2.804  25.557  < 2e-16 ***
## holidayTRUE    -10878.750    782.048 -13.911  < 2e-16 ***
## monthAug         4545.484    783.689   5.800 9.02e-09 ***
## monthDec        -2788.993    738.798  -3.775 0.000170 ***
## monthFeb        -4963.121    903.479  -5.493 5.07e-08 ***
## monthJan        -4976.984    815.816  -6.101 1.54e-09 ***
## monthJul         3057.235    811.453   3.768 0.000175 ***
## monthJun         4901.322    807.152   6.072 1.82e-09 ***
```

```
## monthMar        -2979.466    740.735  -4.022 6.22e-05 ***
## monthMay         3985.920    762.057   5.230 2.08e-07 ***
## monthNov         2337.717    711.269   3.287 0.001051 **
## monthOct         7113.183    694.104  10.248  < 2e-16 ***
## monthSep         7002.524    736.992   9.501  < 2e-16 ***
## dayofweekMon     -763.630    478.297  -1.597 0.110697
## dayofweekSat    -5096.812    477.994 -10.663  < 2e-16 ***
## dayofweekSun    -6574.046    481.302 -13.659  < 2e-16 ***
## dayofweekThurs    528.091    475.128   1.111 0.266648
## dayofweekTues    -421.366    477.103  -0.883 0.377366
## dayofweekWed      522.199    476.202   1.097 0.273098
## PRCP            -9753.592    462.889 -21.071  < 2e-16 ***
## SNWD             -241.946     63.355  -3.819 0.000143 ***
## SNOW              202.115    240.878   0.839 0.401637
## TMAX              335.410     29.090  11.530  < 2e-16 ***
## TMIN              -71.273     31.817  -2.240 0.025317 *
## AWND             -329.082     73.329  -4.488 8.08e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3967 on 949 degrees of freedom
## Multiple R-squared:  0.8817, Adjusted R-squared:  0.8786
## F-statistic:   283 on 25 and 949 DF,  p-value: < 2.2e-16
```

Model Selection: use AIC to decide the best model and make sure check the multicollinearity

Steps

1. for p =1,2,3,4,5 iterate all the possible combinations
2. when a model with lower AIC comes up, update the best model for the p
3. control multicollinearity by elimating the model with any parameters VIF>4
4. test whether the situations without intercept would have better performance

```
#model selection
best <- list()
#q = 1
fit0 = lm(trips~1,data = citi_train)
best[[1]]<-list(fit0,NULL)

#consider the conditions that has intercept
for (i in 1:4){
  AIC <- 10^10
  for (iter in 1:dim(combn(10, i))[2]){
    df <- data.frame(citi_train[,c(1,combn(10, i)[,iter]+1)])
    fit_temp <- lm(trips~.,data = df)
    #if(i>2){print(car::vif(fit_temp))}
    if(i >1&& any(car::vif(fit_temp)>4)){next}
    if (AIC >AIC(fit_temp)){
      best[[i+1]]<- list(fit_temp,names(citi_train)[combn(10, i)[,iter]+1])
    }
  }
}
#consider the conditions that no intercept
for (i in 1:5){
  AIC <- AIC(best[[i]][[1]])
  for (iter in 1:dim(combn(10, i))[2]){
    df <- data.frame(citi_train[,c(1,combn(10, i)[,iter]+1)])
    fit_temp <- lm(trips~.-1,data = df)
    #if(i>2){print(car::vif(fit_temp))}
    if(i >1&& any(car::vif(fit_temp)>4)){next}
    if (AIC >AIC(fit_temp)){
      best[[i]]<- list(fit_temp,names(citi_train)[combn(10, i)[,iter]+1])
    }
  }
}
```

```r
mse <- function(sm)
    mean(sm$residuals^2)
rsq <- function (preds, actual,y) {
  rss <- sum((preds -  (actual)) ^ 2) ## residual sum of squares
  ess <- sum((preds -  mean(actual)) ^ 2)
  tss <- sum((actual - mean(y)) ^ 2)  ## total sum of squares
rsq <-  1-rss/tss}
modelperf <- data.frame()
model <- NULL
for (i in 1:5){
  sm <- summary(best[[i]][[1]])
  y_test <- citi_test$trips
  new_data <- citi_test%>%dplyr::select(best[[i]][[2]])
  y_pred <- predict(best[[i]][[1]],new_data)
  mse_test <- mean(y_pred-y_test)^2
  model[i]<-paste(best[[i]][[2]], collapse = ',')
  modelperf <- rbind(modelperf,
                     c(i,mse(sm),sm$r.squared,mse_test,rsq(y_pred, y_test,citi_train$trips) ))
}
modelperf <- cbind(modelperf,model)
names(modelperf)<-c('q','mse','rsq','mse_test','rsq_test','model parameter')
kable(modelperf)
```

q	mse	rsq	mse_test	rsq_test	model parameter
1	62156100	0.9197630	179395619	0.4704363	TMIN
2	62142631	0.9197804	180258264	0.4684775	TMIN,AWND
3	54594133	0.9295247	175467386	0.4993307	SNOW,TMAX,AWND
4	52541373	0.9321746	172741726	0.5031978	SNWD,SNOW,TMAX,AWND
5	51190546	0.9339184	177835683	0.5012867	PRCP,SNWD,SNOW,TMIN,AWND

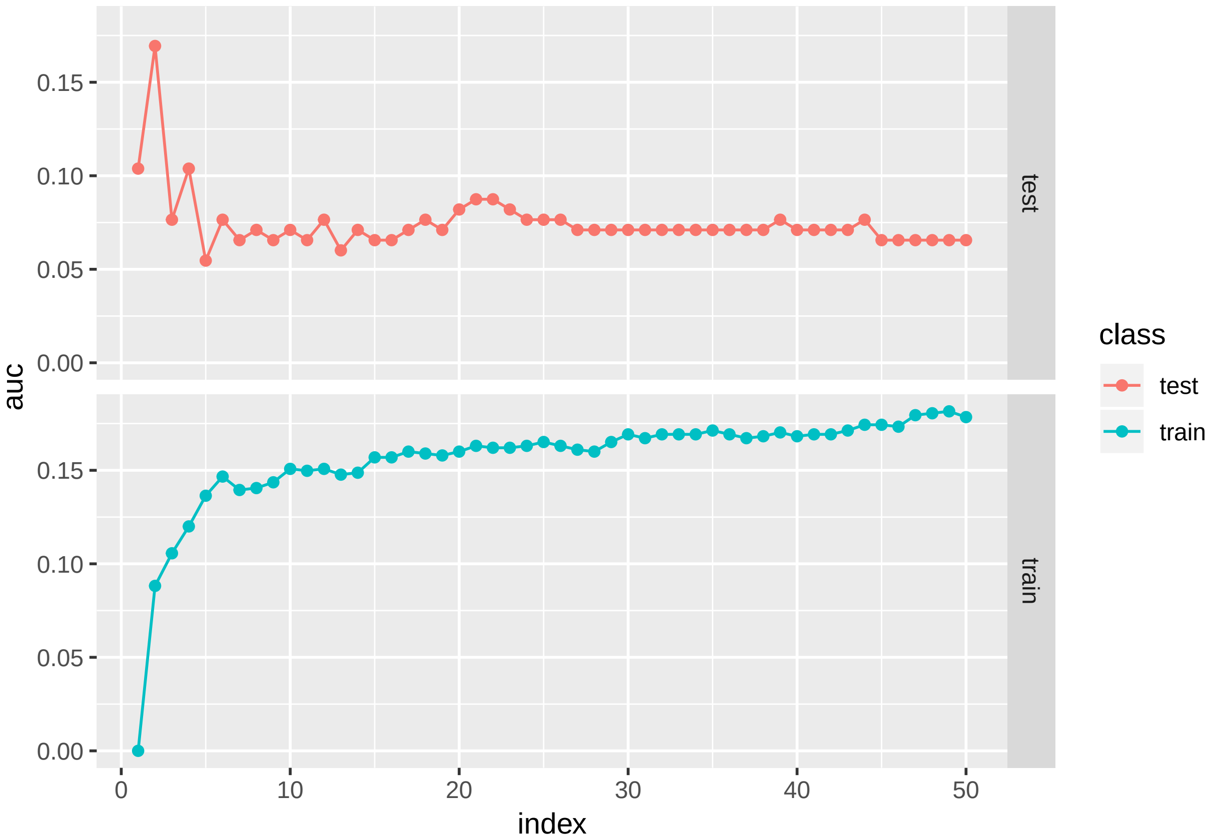
the second model gives the better result with least mse and largest rsq.

Problem 2 Part c.

The process of knn is showed as below, we can compare the test and train set, then pick the k = 5 since when k = 5 the test results are getting stable and reached the lowest error rate, wouldn't cause a overfitting or underfitting problem

```
citi_train$train <- 1
citi_test$train <-0
citi_train_new <- rbind(citi_train,citi_test)%>%
  mutate(class = ifelse(trips>median(trips),1,0))%>%
  dplyr::select(-c(trips,holiday,month,dayofweek))



y_train = (citi_train_new%>%filter(train ==1))$class
X_train = (citi_train_new%>%filter(train ==1))[,-c(8,9)]
y_test = (citi_train_new%>%filter(train ==0))$class
X_test = (citi_train_new%>%filter(train ==0))[,-c(8,9)]

X_train = scale(X_train)
X_test = scale(X_test, center=attr(X_train, "scaled:center"),
                              scale=attr(X_train, "scaled:scale"))

test_auc <- c()
train_auc <- c()
for (i in 1:50){
  train_auc <-c(train_auc,
                sum(as.numeric(knn(X_train, X_train, y_train, k = i))-1!=y_train)/length(y_train))
  test_auc <-c(test_auc,
               sum(as.numeric(knn(X_train, X_test, y_train, k = i))-1!=y_test)/length(y_test))
}
result <- data.frame(index = rep(1:50,2),
                     class = rep(c("train","test"),each=50),auc = c(train_auc,test_auc))
ggplot(data=result,
       aes(x = index, y = auc,group = class,color = class))+
```

```r
geom_line()+facet_grid(class~.)+geom_point()
```



Problem 3: Classification for a Gaussian Mixture (25 points)

Problem 3 Part a.

For loss function $\mathbf{1}\{f(X) \neq Y\}$,

$$E(\mathbf{1}\{f(X) \neq Y\}) = P(f(X) \neq Y)$$

is exactly the bayes risk, which can be minimized by the bayes rule where the decision boundary is $P(Y = 1 \mid X = x) > \frac{1}{2}$

$$P(Y = 1 \mid X = x) = \frac{P(X \mid Y = 1)P(Y = 1)}{P(X \mid Y = 0)P(Y = 0) + P(X \mid Y = 1)P(Y = 1)}$$

$$= \frac{P(X \mid Y = 1)}{P(X \mid Y = 0) + P(X \mid Y = 1)} = \frac{f_0(x)}{f_0(x) + f_1(x)} \geq \frac{1}{2}$$

we simplify it as

$$f(X) = \begin{cases} 1 & \text{if } f_1(X) > f_0(X) \\ 0 & \text{else} \end{cases}$$

$$f_0(x) = f_1(x) \text{ when } x^2 = (x-3)^2 \Rightarrow x = \frac{3}{2}$$

Thus, we have

$$f(X) = \begin{cases} 1 & \text{if } X > \frac{3}{2} \\ 0 & \text{else} \end{cases}$$

Bayes Error Rate $= P(Y=0)P(f(\hat{X}) = 1 \text{ while } f(X) = 0 \mid Y = 0) + P(Y=1)P(f(\hat{X}) = 0 \text{ while } f(X) = 1 \mid Y = 1)$

$$= \frac{1}{2}(0.0668072 + 0.0668072) = 0.0668072$$

Problem 3 Part b.

when the variances of two Guassians are different, the expected loss will still be minimized by the bayes classifers, but with the different decision boundary which is the root a of the following equation

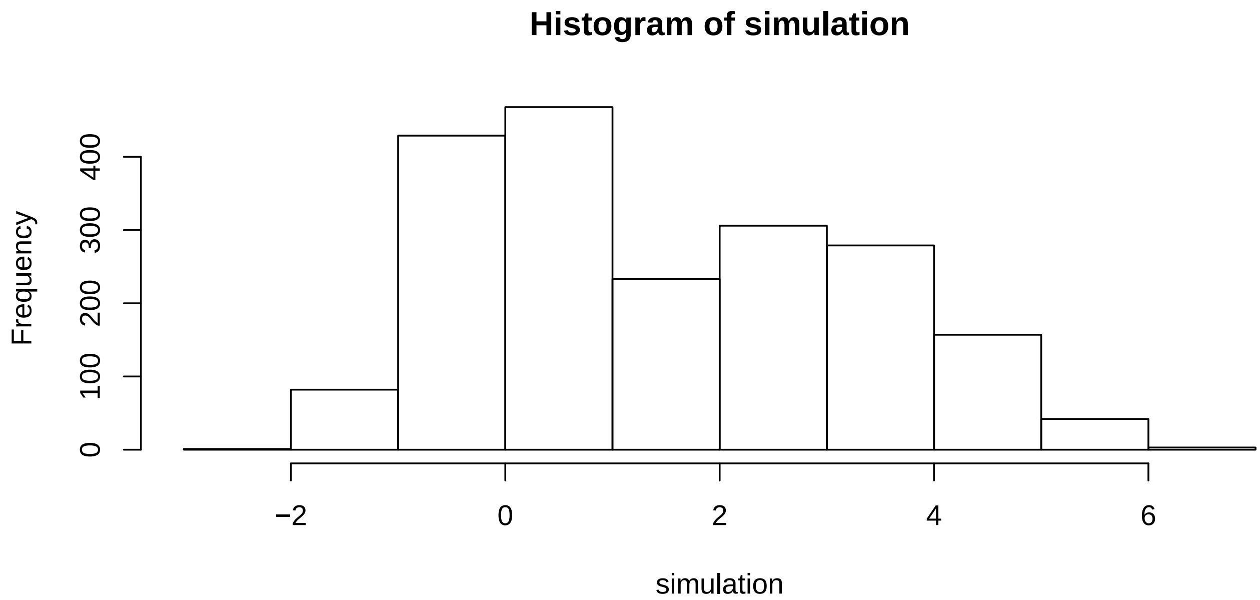
$$f_0(x) = f_1(x) \text{ when } \frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{x^2}{2\sigma_0^2}} = \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(x-3)^2}{2\sigma_1^2}}$$

Thus, we have

$$f(X) = \begin{cases} 1 & \text{if } X > a \\ 0 & \text{else} \end{cases}$$

Problem 3 Part c.

```
set.seed(22)
n = 2000
Y1 <- rbinom(n,1,1/2)
mixture <- function(x){if(x==1) return(rnorm(1,3,sqrt(1.5))) else return(rnorm(1,0,sqrt(0.5)))}
simulation<- sapply(Y1, mixture)
sim_df <- data.frame(label = Y1,simulation = simulation)
hist(simulation)
```




```
test<- sample(n,n/5)
sim_df_test <- sim_df[test,]
sim_df_train <- sim_df[-test,]
sim_df_train_mean <- sim_df_train%>%group_by(label)%>%summarise(mean = mean(simulation))
```

the decision boundary is decided by $\frac{\hat{\mu}_0+\hat{\mu}_1}{2} = 1.2520827$

$$f(X) = \begin{cases} 1 & \text{if } X > 1.252083 \text{ i.e. } f_1(x) > f_0(x) \\ 0 & \text{else} \end{cases}$$

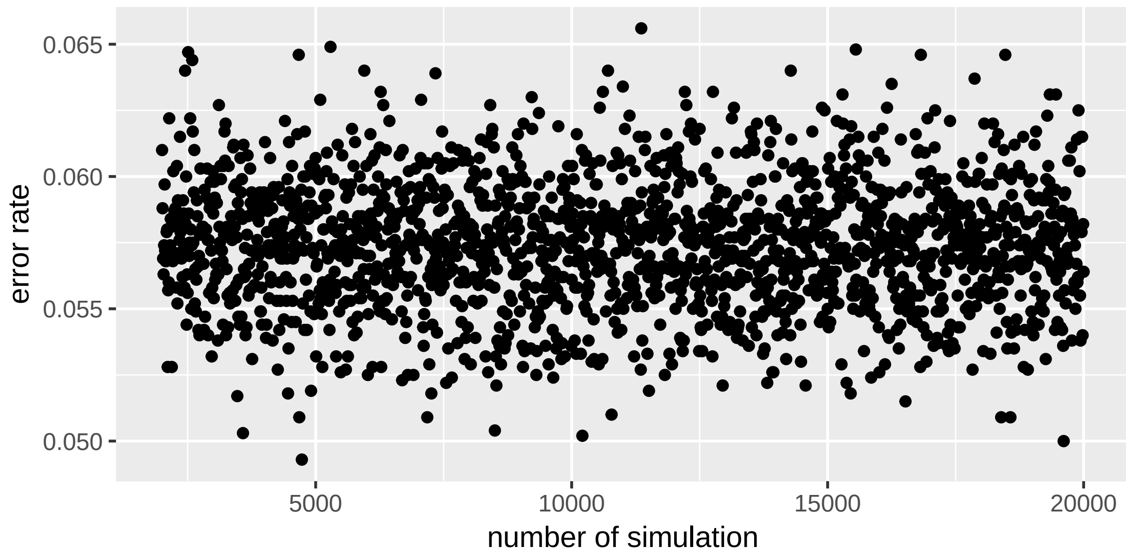
```
therhold <- solve(polynomial(c(-3-(1/2)*log(3),2,2/3)))[2]
sim_df_test <- sim_df_test%>%
  mutate(yhat =
           ifelse(simulation>therhold,1,0))%>%
  mutate(y_diff = (yhat ==label))
estimator<-1-sum(sim_df_test$y_diff)/length(sim_df_test$y_diff)# the error rate
```

The predict error rate is 0.0525

Problem 3 Part d.

The variance of the error rates are overall stable when n is increasing, we can see from the plot below, since when n is large enough the estimation values mu_1, μ_0 are

```r
seq.n <- seq(from = 2000, to = 20000, by = 10)
estimator <- c()
for(n in seq.n){
  Y1 <- rbinom(n,1,1/2)
  mixture <- function(x){if(x==1) return(rnorm(1,3,sqrt(1.5))) else return(rnorm(1,0,sqrt(0.5)))}
  simulation<- sapply(Y1, mixture)
  sim_df <- data.frame(label = Y1,simulation = simulation)
  sim_df_train_mean <- sim_df%>%group_by(label)%>%
    summarise(mean = mean(simulation))%>%arrange(label)
  mu0<- sim_df_train_mean[1,2]
  mu1<- sim_df_train_mean[2,2]
  #print(paste(mu0,mu1))
  Y1_test <- rbinom(10000,1,1/2)
  simulation_test<- sapply(Y1_test, mixture)
  sim_df <- data.frame(label = Y1_test,simulation = simulation_test)

  #print(solve(polynomial(c(mu0^2 -(1/3)*mu1^2 -(1/2)*log(3),(2/3)*mu1-2*mu0,2/3)))[2])
  sim_df <- sim_df%>%
  mutate(yhat = ifelse(simulation>
                       solve(polynomial(c(mu0^2 -(1/3)*mu1^2 -
                                          (1/2)*log(3),(2/3)*mu1-2*mu0,2/3)))[2],1,0))%>%
  mutate(y_diff = (yhat ==label))
  estimator<-c(estimator,1-sum(sim_df$y_diff)/length(sim_df$y_diff))
}
sim <- data.frame(n = seq.n, errorrate = estimator)

ggplot(data = sim, aes(x = n, y=errorrate))+geom_point()+ylab("error rate")+
  xlab("number of simulation")
```


0.065
0.060
0.055
0.050
error rate
5000
10000
15000
20000
number of simulation