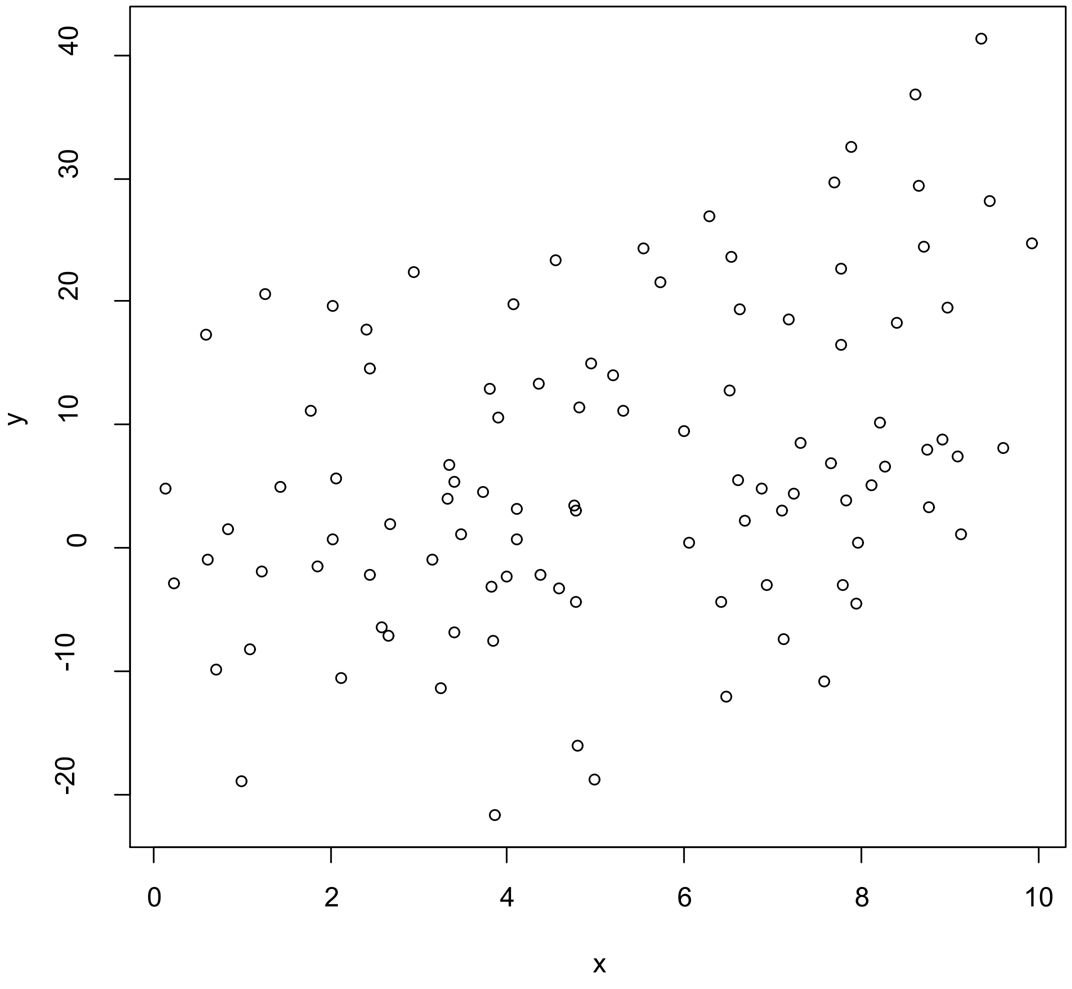
20153284 송재준 2020 Spring Data Mining Assignment

#1

'MID2020Sdata' consists of 2 continuous variables 'x' and 'y' with 100 samples. I made a scatter plots to briefly see the relationship between X and Y.



By looking at the plot, x and y doesn't seem to have a strong relationship.

Before getting into solving problems, I set seed to 20153284(which is my student ID)

#1-(a)

First, I made a function of correlation coefficent between two variables. Then I used this in 'boot()' function with 1000 bootstrap replicates to get correlation coefficient between x and y with bootstrap bias estimate and bootstrap standard error.

The result was

-correlation coefficient between x and y : **0.3731081**

-bootstrap bisa estimate : **-0.005297024**

-boostrap standard error : **0,0821492.**

As expected, correlation coefficient isn't quite large.

#1-(b)

I solved this problem by using boo.ci() function(which gives a confidence interval of bootstrap). I created 4 types of confidence Interval: normal, basic, perc for comparing. This is the result:

95% bootstrap C.I			
type	Normal	Basic	Percentile
95% C.I	(0.2174, 0.5394)	(0.2315, 0.5406)	(0.2057, 0.5148)

As seen, confidence intervals don't have much difference between types in this case.

#2

‘Hitters’ dataset consists of 20 variable with 322 observations.
By deleting NA values, 263 observations remained.

Players' salary is the response variable. I turned it into a binary response 'y' that counts values lower than 750 = 0, else 1.

#2-(a)

In this one, I used 'klaR' and 'caret' packages.

I set x(=every variables except y) as a predictor and y as a response variable and nb(Naive-Bayes) as a classification method, 10-fold CV as a resampling method. This is the accuracy table :

Resample	Accuracy	Classification Error
Fold1	0.6923077	0.3176923
Fold2	0.8461538	0.1538462
Fold3	0.8461538	0.1538462
Fold4	0.7692308	0.1307692
Fold5	0.9230769	0.0869231
Fold6	0.9259259	0.0840741
Fold7	0.7407407	0.2592593
Fold8	0.7307692	0.2692308
Fold9	0.8076923	0.1923077
Fold10	0.8518519	0.141481

(Classification Error = 1-Accuracy)

The accuracy without kernel is 0.7945869,
so the classification error is **0.2054131** (=1-accuracy).

This is the Naive-Bayes classifier table :

	0 (true, <750)	**1 (true, ≥750)**
0 (predict, <750)	164	10
1 (predict, ≥750)	29	60

#2-(b)

Same packages and functions are used.

I set x(=every variables except y) as a predictor and y as a response variable and glm(generalized linear model) as a classification method,
10-fold CV as a resampling method. This is the accuracy table:

Resample	Accuracy	Classification Error
Fold1	0.6923077	0.3076923
Fold2	0.8461538	0.1538462
Fold3	0.8148148	0.1851852
Fold4	0.7777778	0.2222222
Fold5	0.7692308	0.2307692
Fold6	0.6538462	0.3461538
Fold7	0.8888889	0.1111111
Fold8	0.9615385	0.0384615
Fold9	0.8076923	0.1923077
Fold10	0.6538462	0.3461538

This is the Logit Model Classifier table:

	0 (true, <750)	**1 (true, ≥750)**
0 (predict, <750)	177	30
1 (predict, ≥750)	16	40

Generally, Naive-Bayes method's Accuracy seems better than Logit Model Classifier.

#3

I made a data, 'data3'.

'data3' consists of 3 categorical variables 'X1', 'X2', 'Y' with 50 samples.

#3-(a)

I used multinom() from 'nnet' package.
Bayes Classifier table for $\hat{Y}$:

$\hat{Y}$(predict) \ Y(true)	1	2	3
1	8	3	4
2	1	15	3
3	1	2	13

#3-(b)

Error rate = (# of misclassification / n).

Error rate is **0.28**.

Appendix : R codes

```
##1

#loading data
data1 <- read.csv('MID2020Sdata.txt', header=TRUE, sep=' ')

str(data1)

#reordering data
data <- data.frame(x=data1[ , 1], y=data1[ , 2])
x <- data1[ , 1];x
y <- data1[ , 2];y

str(data1)

#visualizing data
plot(data1$x, data1$y,
     xlab='x', ylab='y')

set.seed(20153284)

##1-(a)

install.packages('ISLR')

library(ISLR)
library(boot)

#making correlation coefficient function
cor.fn <- function(data, number){
numbermatch <- data[number, ]
return(cor(numbermatch$x, numbermatch$y))
}

#bootstrap with 1000 replicates
data1.boot <- boot(data1, cor.fn, R=1000);data1.boot


##1-(b)
```

```r
boot.ci(data1.boot, type=c('norm', 'basic', 'perc'),
          conf=0.95)

######################################################################

##2

Hitters

str(Hitters)

#deleting NA
Hitters2 <- na.omit(Hitters);Hitters2

str(Hitters2)


#creating binary response vector with salary
Hitters2$y[Hitters2[ , 19]>=750]=1

Hitters2$y[Hitters2[ , 19]<750]=0

Hitters2$y=as.factor(Hitters2$y)

Hitters2 <- subset(Hitters2, select = -c(Salary));Hitters2

str(Hitters2)

set.seed(20153284)


##2-(a)

install.packages('klaR')

library(klaR)

library(caret)

library(e1071)
```

```
x <- Hitters2[ , -20];x
y <- Hitters2[, 20];y
nbmodel <- train(x,y, 'nb',
                  trControl=trainControl
                 (method='cv', number=10));nbmodel

nbmodel$resample

predict(nbmodel$finalModel,x)

table(predict(nbmodel$finalModel,x)$class, y)


##2-(b)

lmmodel <- train(x,y, 'glm',
                trControl=trainControl(method='cv', number=10));lmmodel

lmmodel$resample

predict(lmmodel$finalModel,x)

table(predict(lmmodel,x), y)

#################################################################

##3



data3 <- read.csv('data3.csv', header=TRUE);data3

str(data3)

#making Y a factor variable
data3$Y <- as.factor(data3$Y)

str(data3)
```

```r
##3-(a)

install.packages('nnet')

library(nnet)

fitm <- multinom(Y ~ X1 * X2, data3);fitm

predm <- predict(fitm, newdata=data3);predm

table(predm, data3$Y)


##3-(b)

mean(predm != data3$Y)
```