# YuchenPengHW6

Question 14.1 The breast cancer data set breast-cancer-wisconsin.data.txt from <http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/> (description at <http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29> ) has missing values.

1. Use the mean/mode imputation method to impute values for the missing data.
2. Use regression to impute values for the missing data.
3. Use regression with perturbation to impute values for the missing data.
4. (Optional) Compare the results and quality of classification models (e.g., SVM, KNN) build using
5. the data sets from questions 1,2,3;
6. the data that remains after data points with missing values are removed; and (3) the data set when a binary variable is introduced to indicate missing values.

Hide

Data = read.csv("breast-cancer-wisconsin.data.txt",header = FALSE,na.strings = "?")  
str(Data)

'data.frame': 699 obs. of 11 variables:  
 $ V1 : int 1000025 1002945 1015425 1016277 1017023 1017122 1018099 1018561 1033078 1033078 ...  
 $ V2 : int 5 5 3 6 4 8 1 2 2 4 ...  
 $ V3 : int 1 4 1 8 1 10 1 1 1 2 ...  
 $ V4 : int 1 4 1 8 1 10 1 2 1 1 ...  
 $ V5 : int 1 5 1 1 3 8 1 1 1 1 ...  
 $ V6 : int 2 7 2 3 2 7 2 2 2 2 ...  
 $ V7 : int 1 10 2 4 1 10 10 1 1 1 ...  
 $ V8 : int 3 3 3 3 3 9 3 3 1 2 ...  
 $ V9 : int 1 2 1 7 1 7 1 1 1 1 ...  
 $ V10: int 1 1 1 1 1 1 1 1 5 1 ...  
 $ V11: int 2 2 2 2 2 4 2 2 2 2 ...

Hide

print(summary(Data))

V1 V2 V3 V4   
 Min. : 61634 Min. : 1.000 Min. : 1.000 Min. : 1.000   
 1st Qu.: 870688 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 1.000   
 Median : 1171710 Median : 4.000 Median : 1.000 Median : 1.000   
 Mean : 1071704 Mean : 4.418 Mean : 3.134 Mean : 3.207   
 3rd Qu.: 1238298 3rd Qu.: 6.000 3rd Qu.: 5.000 3rd Qu.: 5.000   
 Max. :13454352 Max. :10.000 Max. :10.000 Max. :10.000   
   
 V5 V6 V7 V8   
 Min. : 1.000 Min. : 1.000 Min. : 1.000 Min. : 1.000   
 1st Qu.: 1.000 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 2.000   
 Median : 1.000 Median : 2.000 Median : 1.000 Median : 3.000   
 Mean : 2.807 Mean : 3.216 Mean : 3.545 Mean : 3.438   
 3rd Qu.: 4.000 3rd Qu.: 4.000 3rd Qu.: 6.000 3rd Qu.: 5.000   
 Max. :10.000 Max. :10.000 Max. :10.000 Max. :10.000   
 NA's :16   
 V9 V10 V11   
 Min. : 1.000 Min. : 1.000 Min. :2.00   
 1st Qu.: 1.000 1st Qu.: 1.000 1st Qu.:2.00   
 Median : 1.000 Median : 1.000 Median :2.00   
 Mean : 2.867 Mean : 1.589 Mean :2.69   
 3rd Qu.: 4.000 3rd Qu.: 1.000 3rd Qu.:4.00   
 Max. :10.000 Max. :10.000 Max. :4.00

Hide

# 1. Sample code number: id number  
# 2. Clump Thickness: 1 - 10  
# 3. Uniformity of Cell Size: 1 - 10  
# 4. Uniformity of Cell Shape: 1 - 10  
# 5. Marginal Adhesion: 1 - 10  
# 6. Single Epithelial Cell Size: 1 - 10  
# 7. Bare Nuclei: 1 - 10  
# 8. Bland Chromatin: 1 - 10  
# 9. Normal Nucleoli: 1 - 10  
# 10. Mitoses: 1 - 10  
# 11. Class: (2 for benign, 4 for malignant)  
  
colnames(Data) <- c("ID", "Clump\_Thickness", "Cell\_Size", "Cell\_Shape",  
 "Marginal\_Adhesion", "Single\_Epith\_Cell\_Size", "Bare\_Nuclei",   
 "Bland\_Chromatin","Normal\_Nucleoli", "Mitoses", "Class")  
  
require(dplyr)  
Data <- Data %>%mutate(Class = ifelse(Class == 4,1,0))  
  
drops <- c("ID")  
Data <- Data[ , !(names(Data) %in% drops)]  
summary(Data)

Clump\_Thickness Cell\_Size Cell\_Shape Marginal\_Adhesion  
 Min. : 1.000 Min. : 1.000 Min. : 1.000 Min. : 1.000   
 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 1.000 1st Qu.: 1.000   
 Median : 4.000 Median : 1.000 Median : 1.000 Median : 1.000   
 Mean : 4.418 Mean : 3.134 Mean : 3.207 Mean : 2.807   
 3rd Qu.: 6.000 3rd Qu.: 5.000 3rd Qu.: 5.000 3rd Qu.: 4.000   
 Max. :10.000 Max. :10.000 Max. :10.000 Max. :10.000   
   
 Single\_Epith\_Cell\_Size Bare\_Nuclei Bland\_Chromatin Normal\_Nucleoli   
 Min. : 1.000 Min. : 1.000 Min. : 1.000 Min. : 1.000   
 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 2.000 1st Qu.: 1.000   
 Median : 2.000 Median : 1.000 Median : 3.000 Median : 1.000   
 Mean : 3.216 Mean : 3.545 Mean : 3.438 Mean : 2.867   
 3rd Qu.: 4.000 3rd Qu.: 6.000 3rd Qu.: 5.000 3rd Qu.: 4.000   
 Max. :10.000 Max. :10.000 Max. :10.000 Max. :10.000   
 NA's :16   
 Mitoses Class   
 Min. : 1.000 Min. :0.0000   
 1st Qu.: 1.000 1st Qu.:0.0000   
 Median : 1.000 Median :0.0000   
 Mean : 1.589 Mean :0.3448   
 3rd Qu.: 1.000 3rd Qu.:1.0000   
 Max. :10.000 Max. :1.0000

Hide

print(summary(Data))

Clump\_Thickness Cell\_Size Cell\_Shape Marginal\_Adhesion  
 Min. : 1.000 Min. : 1.000 Min. : 1.000 Min. : 1.000   
 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 1.000 1st Qu.: 1.000   
 Median : 4.000 Median : 1.000 Median : 1.000 Median : 1.000   
 Mean : 4.418 Mean : 3.134 Mean : 3.207 Mean : 2.807   
 3rd Qu.: 6.000 3rd Qu.: 5.000 3rd Qu.: 5.000 3rd Qu.: 4.000   
 Max. :10.000 Max. :10.000 Max. :10.000 Max. :10.000   
   
 Single\_Epith\_Cell\_Size Bare\_Nuclei Bland\_Chromatin Normal\_Nucleoli   
 Min. : 1.000 Min. : 1.000 Min. : 1.000 Min. : 1.000   
 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 2.000 1st Qu.: 1.000   
 Median : 2.000 Median : 1.000 Median : 3.000 Median : 1.000   
 Mean : 3.216 Mean : 3.545 Mean : 3.438 Mean : 2.867   
 3rd Qu.: 4.000 3rd Qu.: 6.000 3rd Qu.: 5.000 3rd Qu.: 4.000   
 Max. :10.000 Max. :10.000 Max. :10.000 Max. :10.000   
 NA's :16   
 Mitoses Class   
 Min. : 1.000 Min. :0.0000   
 1st Qu.: 1.000 1st Qu.:0.0000   
 Median : 1.000 Median :0.0000   
 Mean : 1.589 Mean :0.3448   
 3rd Qu.: 1.000 3rd Qu.:1.0000   
 Max. :10.000 Max. :1.0000

Hide

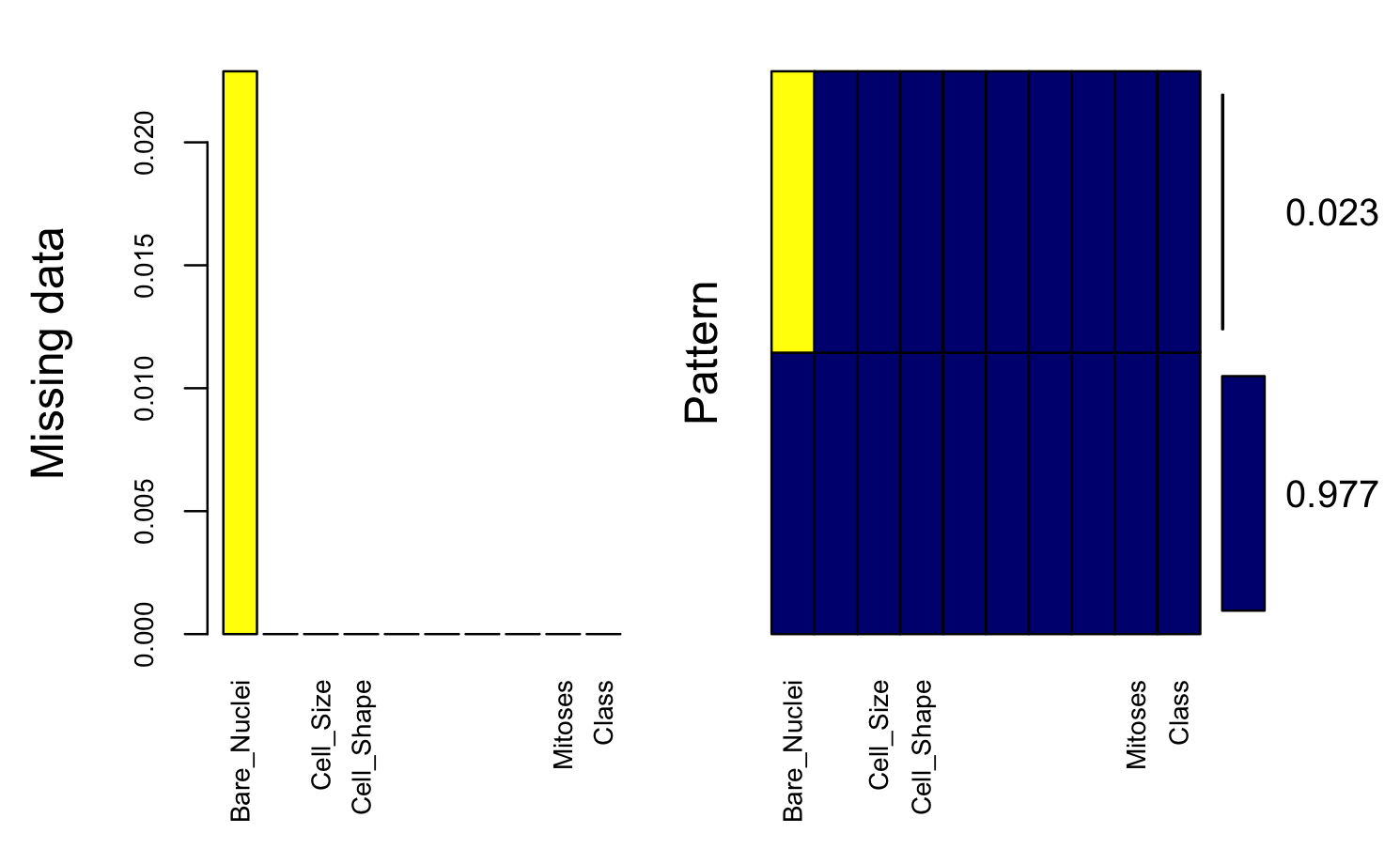
#Bare\_Nuclei has 16 missing values  
library(VIM)  
mice\_plot <- aggr(Data, col=c('navyblue','yellow'),numbers=TRUE, sortVars=TRUE,  
labels=names(Data), cex.axis=.7,gap=3, ylab=c("Missing data","Pattern"))

Variables sorted by number of missings:

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Variable**  <chr> | **Count**  <dbl> |
| Bare\_Nuclei | 0.02288984 |
| Clump\_Thickness | 0.00000000 |
| Cell\_Size | 0.00000000 |
| Cell\_Shape | 0.00000000 |
| Marginal\_Adhesion | 0.00000000 |
| Single\_Epith\_Cell\_Size | 0.00000000 |
| Bland\_Chromatin | 0.00000000 |
| Normal\_Nucleoli | 0.00000000 |
| Mitoses | 0.00000000 |
| Class | 0.00000000 |

1-10 of 10 rows



Hide

# From the chart I see:  
# There are 67% values in the data set with no missing value. There are 97.8% with  
# no missing values.  
# This chart can be very useful if multiple columns have missing values.  
  
Data\_imput\_w\_mean <- Data  
Data\_imput\_w\_mean$Bare\_Nuclei[is.na(Data\_imput\_w\_mean$Bare\_Nuclei)] <- mean(Data\_imput\_w\_mean$Bare\_Nuclei, na.rm=TRUE)

Hide

Data\_imput\_w\_mode <- Data  
Data\_imput\_w\_mode$Bare\_Nuclei[is.na(Data\_imput\_w\_mean$Bare\_Nuclei)] <- mode(Data\_imput\_w\_mean$Bare\_Nuclei)

Hide

# Use regression to impute missing values  
library(mice)  
Data\_imput\_w\_regression <- Data  
imp <- mice(Data, method="norm.predict",m=1)

iter imp variable  
 1 1 Bare\_Nuclei  
 2 1 Bare\_Nuclei  
 3 1 Bare\_Nuclei  
 4 1 Bare\_Nuclei  
 5 1 Bare\_Nuclei

Hide

#method="norm.predict"  
#Linear regression, predicted values (numeric)  
imp

Class: mids  
Number of multiple imputations: 1   
Imputation methods:  
 Clump\_Thickness Cell\_Size Cell\_Shape   
 "" "" ""   
 Marginal\_Adhesion Single\_Epith\_Cell\_Size Bare\_Nuclei   
 "" "" "norm.predict"   
 Bland\_Chromatin Normal\_Nucleoli Mitoses   
 "" "" ""   
 Class   
 ""   
PredictorMatrix:  
 Clump\_Thickness Cell\_Size Cell\_Shape Marginal\_Adhesion  
Clump\_Thickness 0 1 1 1  
Cell\_Size 1 0 1 1  
Cell\_Shape 1 1 0 1  
Marginal\_Adhesion 1 1 1 0  
Single\_Epith\_Cell\_Size 1 1 1 1  
Bare\_Nuclei 1 1 1 1  
 Single\_Epith\_Cell\_Size Bare\_Nuclei Bland\_Chromatin  
Clump\_Thickness 1 1 1  
Cell\_Size 1 1 1  
Cell\_Shape 1 1 1  
Marginal\_Adhesion 1 1 1  
Single\_Epith\_Cell\_Size 0 1 1  
Bare\_Nuclei 1 0 1  
 Normal\_Nucleoli Mitoses Class  
Clump\_Thickness 1 1 1  
Cell\_Size 1 1 1  
Cell\_Shape 1 1 1  
Marginal\_Adhesion 1 1 1  
Single\_Epith\_Cell\_Size 1 1 1  
Bare\_Nuclei 1 1 1

Hide

# list the actual imputations for Bare\_Nuclei  
  
# Class: mids  
# Number of multiple imputations: 1   
# Imputation methods:  
# Clump\_Thickness Cell\_Size Cell\_Shape   
# "" "" ""   
# Marginal\_Adhesion Single\_Epith\_Cell\_Size Bare\_Nuclei   
# "" "" "norm.predict"   
# Bland\_Chromatin Normal\_Nucleoli Mitoses   
# "" "" ""   
# Class   
# ""   
# PredictorMatrix:  
# Clump\_Thickness Cell\_Size Cell\_Shape Marginal\_Adhesion  
# Clump\_Thickness 0 1 1 1  
# Cell\_Size 1 0 1 1  
# Cell\_Shape 1 1 0 1  
# Marginal\_Adhesion 1 1 1 0  
# Single\_Epith\_Cell\_Size 1 1 1 1  
# Bare\_Nuclei 1 1 1 1  
# Single\_Epith\_Cell\_Size Bare\_Nuclei Bland\_Chromatin  
# Clump\_Thickness 1 1 1  
# Cell\_Size 1 1 1  
# Cell\_Shape 1 1 1  
# Marginal\_Adhesion 1 1 1  
# Single\_Epith\_Cell\_Size 0 1 1  
# Bare\_Nuclei 1 0 1  
# Normal\_Nucleoli Mitoses Class  
# Clump\_Thickness 1 1 1  
# Cell\_Size 1 1 1  
# Cell\_Shape 1 1 1  
# Marginal\_Adhesion 1 1 1  
# Single\_Epith\_Cell\_Size 1 1 1  
# Bare\_Nuclei 1 1 1  
Data\_imput\_w\_regression <- complete(imp)  
  
  
#Use regression with perturbation impute values  
imp\_perturbation <- mice(Data, method="norm.nob",m=1)

iter imp variable  
 1 1 Bare\_Nuclei  
 2 1 Bare\_Nuclei  
 3 1 Bare\_Nuclei  
 4 1 Bare\_Nuclei  
 5 1 Bare\_Nuclei

Hide

#method="norm.nob"  
#Linear regression, predicted values (numeric)  
imp\_perturbation

Class: mids  
Number of multiple imputations: 1   
Imputation methods:  
 Clump\_Thickness Cell\_Size Cell\_Shape   
 "" "" ""   
 Marginal\_Adhesion Single\_Epith\_Cell\_Size Bare\_Nuclei   
 "" "" "norm.nob"   
 Bland\_Chromatin Normal\_Nucleoli Mitoses   
 "" "" ""   
 Class   
 ""   
PredictorMatrix:  
 Clump\_Thickness Cell\_Size Cell\_Shape Marginal\_Adhesion  
Clump\_Thickness 0 1 1 1  
Cell\_Size 1 0 1 1  
Cell\_Shape 1 1 0 1  
Marginal\_Adhesion 1 1 1 0  
Single\_Epith\_Cell\_Size 1 1 1 1  
Bare\_Nuclei 1 1 1 1  
 Single\_Epith\_Cell\_Size Bare\_Nuclei Bland\_Chromatin  
Clump\_Thickness 1 1 1  
Cell\_Size 1 1 1  
Cell\_Shape 1 1 1  
Marginal\_Adhesion 1 1 1  
Single\_Epith\_Cell\_Size 0 1 1  
Bare\_Nuclei 1 0 1  
 Normal\_Nucleoli Mitoses Class  
Clump\_Thickness 1 1 1  
Cell\_Size 1 1 1  
Cell\_Shape 1 1 1  
Marginal\_Adhesion 1 1 1  
Single\_Epith\_Cell\_Size 1 1 1  
Bare\_Nuclei 1 1 1

Hide

# list the actual imputations for Bare\_Nuclei  
Data\_imput\_w\_regression\_pert <- complete(imp\_perturbation)

1. (Optional) Compare the results and quality of classification models (e.g., SVM, KNN) build using
2. the data sets from questions 1,2,3;
3. the data that remains after data points with missing values are removed; and (3) the data set when a binary variable is introduced to indicate missing values.

Hide

split\_test\_train <- function(df) {  
 set.seed(1)  
 train = sample(1:nrow(df), 0.75\*nrow(df))  
 train\_df <- df[train,]  
 test\_df <- df[-train,]  
 return (list("train" = train\_df, "test" = test\_df))  
}  
  
result\_w\_mean <- split\_test\_train(Data\_imput\_w\_mean)  
  
summary(result\_w\_mean$train)

Clump\_Thickness Cell\_Size Cell\_Shape Marginal\_Adhesion  
 Min. : 1.000 Min. : 1.000 Min. : 1.000 Min. : 1.000   
 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 1.000 1st Qu.: 1.000   
 Median : 4.000 Median : 1.000 Median : 2.000 Median : 1.000   
 Mean : 4.393 Mean : 3.145 Mean : 3.179 Mean : 2.847   
 3rd Qu.: 6.000 3rd Qu.: 5.000 3rd Qu.: 5.000 3rd Qu.: 4.000   
 Max. :10.000 Max. :10.000 Max. :10.000 Max. :10.000   
 Single\_Epith\_Cell\_Size Bare\_Nuclei Bland\_Chromatin Normal\_Nucleoli   
 Min. : 1.000 Min. : 1.000 Min. : 1.000 Min. : 1.000   
 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 2.000 1st Qu.: 1.000   
 Median : 2.000 Median : 1.000 Median : 3.000 Median : 1.000   
 Mean : 3.174 Mean : 3.498 Mean : 3.454 Mean : 2.855   
 3rd Qu.: 4.000 3rd Qu.: 5.000 3rd Qu.: 5.000 3rd Qu.: 3.250   
 Max. :10.000 Max. :10.000 Max. :10.000 Max. :10.000   
 Mitoses Class   
 Min. : 1.000 Min. :0.0000   
 1st Qu.: 1.000 1st Qu.:0.0000   
 Median : 1.000 Median :0.0000   
 Mean : 1.532 Mean :0.3359   
 3rd Qu.: 1.000 3rd Qu.:1.0000   
 Max. :10.000 Max. :1.0000

Hide

dim(result\_w\_mean$train)

[1] 524 10

Hide

dim(result\_w\_mean$test)

[1] 175 10

Hide

data\_train\_target\_category <- result\_w\_mean$train[,"Class"]  
data\_test\_target\_category <- result\_w\_mean$test[,"Class"]  
  
#run KNN models  
library(class)  
  
pr <- knn(result\_w\_mean$train,result\_w\_mean$test, cl=data\_train\_target\_category)  
pr

[1] 0 1 0 0 1 1 0 1 0 0 0 0 1 1 1 1 1 1 0 1 0 0 0 0 0 1 1 0 1 0 0 1 0 0 1 1 0  
 [38] 0 0 1 0 1 0 1 0 0 0 1 1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 0 1 0 0 1 1 0  
 [75] 1 1 1 0 1 0 1 0 1 0 1 0 1 1 0 1 0 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0  
[112] 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0  
[149] 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0  
Levels: 0 1

Hide

##create confusion matrix  
cf <- table(pr,data\_test\_target\_category)  
cf

data\_test\_target\_category  
pr 0 1  
 0 110 6  
 1 0 59

Hide

# data\_test\_target\_category  
#pr 0 1  
# 0 79 38  
# 1 31 27  
  
 ##this function divides the correct predictions by total number of predictions that tell us how accurate teh model is.  
   
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) \* 100}  
accuracy(cf)

[1] 96.57143

Hide

#60.5% accuracy of the data imputed with mean  
  
  
#Data\_imput\_w\_regression  
  
result\_w\_regression <- split\_test\_train(Data\_imput\_w\_regression)  
  
summary(result\_w\_regression$train)

Clump\_Thickness Cell\_Size Cell\_Shape Marginal\_Adhesion  
 Min. : 1.000 Min. : 1.000 Min. : 1.000 Min. : 1.000   
 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 1.000 1st Qu.: 1.000   
 Median : 4.000 Median : 1.000 Median : 2.000 Median : 1.000   
 Mean : 4.393 Mean : 3.145 Mean : 3.179 Mean : 2.847   
 3rd Qu.: 6.000 3rd Qu.: 5.000 3rd Qu.: 5.000 3rd Qu.: 4.000   
 Max. :10.000 Max. :10.000 Max. :10.000 Max. :10.000   
 Single\_Epith\_Cell\_Size Bare\_Nuclei Bland\_Chromatin Normal\_Nucleoli   
 Min. : 1.000 Min. : 1.000 Min. : 1.000 Min. : 1.000   
 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 2.000 1st Qu.: 1.000   
 Median : 2.000 Median : 1.000 Median : 3.000 Median : 1.000   
 Mean : 3.174 Mean : 3.462 Mean : 3.454 Mean : 2.855   
 3rd Qu.: 4.000 3rd Qu.: 6.000 3rd Qu.: 5.000 3rd Qu.: 3.250   
 Max. :10.000 Max. :10.000 Max. :10.000 Max. :10.000   
 Mitoses Class   
 Min. : 1.000 Min. :0.0000   
 1st Qu.: 1.000 1st Qu.:0.0000   
 Median : 1.000 Median :0.0000   
 Mean : 1.532 Mean :0.3359   
 3rd Qu.: 1.000 3rd Qu.:1.0000   
 Max. :10.000 Max. :1.0000

Hide

dim(result\_w\_regression$train)

[1] 524 10

Hide

dim(result\_w\_regression$test)

[1] 175 10

Hide

data\_train\_target\_category <- result\_w\_regression$train[,"Class"]  
data\_test\_target\_category <- result\_w\_regression$test[,"Class"]  
  
  
pr <- knn(result\_w\_regression$train,result\_w\_regression$test, cl=data\_train\_target\_category)  
  
##create confusion matrix  
cf <- table(pr,data\_test\_target\_category)  
cf

data\_test\_target\_category  
pr 0 1  
 0 110 6  
 1 0 59

Hide

# data\_test\_target\_category  
# pr 0 1  
# 0 110 6  
# 1 0 59  
  
accuracy(cf)

[1] 96.57143

Hide

#96% accuracy of the data imputed with regression  
  
  
#Data\_imput\_w\_regression\_w\_perturbation  
  
result\_w\_regression\_pert <- split\_test\_train(Data\_imput\_w\_regression\_pert)  
  
summary(result\_w\_regression\_pert$train)

Clump\_Thickness Cell\_Size Cell\_Shape Marginal\_Adhesion  
 Min. : 1.000 Min. : 1.000 Min. : 1.000 Min. : 1.000   
 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 1.000 1st Qu.: 1.000   
 Median : 4.000 Median : 1.000 Median : 2.000 Median : 1.000   
 Mean : 4.393 Mean : 3.145 Mean : 3.179 Mean : 2.847   
 3rd Qu.: 6.000 3rd Qu.: 5.000 3rd Qu.: 5.000 3rd Qu.: 4.000   
 Max. :10.000 Max. :10.000 Max. :10.000 Max. :10.000   
 Single\_Epith\_Cell\_Size Bare\_Nuclei Bland\_Chromatin Normal\_Nucleoli   
 Min. : 1.000 Min. :-0.5784 Min. : 1.000 Min. : 1.000   
 1st Qu.: 2.000 1st Qu.: 1.0000 1st Qu.: 2.000 1st Qu.: 1.000   
 Median : 2.000 Median : 1.0000 Median : 3.000 Median : 1.000   
 Mean : 3.174 Mean : 3.4775 Mean : 3.454 Mean : 2.855   
 3rd Qu.: 4.000 3rd Qu.: 6.0000 3rd Qu.: 5.000 3rd Qu.: 3.250   
 Max. :10.000 Max. :10.0000 Max. :10.000 Max. :10.000   
 Mitoses Class   
 Min. : 1.000 Min. :0.0000   
 1st Qu.: 1.000 1st Qu.:0.0000   
 Median : 1.000 Median :0.0000   
 Mean : 1.532 Mean :0.3359   
 3rd Qu.: 1.000 3rd Qu.:1.0000   
 Max. :10.000 Max. :1.0000

Hide

data\_train\_target\_category <- result\_w\_regression\_pert$train[,"Class"]  
data\_test\_target\_category <- result\_w\_regression\_pert$test[,"Class"]  
  
  
pr <- knn(result\_w\_regression\_pert$train,result\_w\_regression\_pert$test, cl=data\_train\_target\_category)  
pr

[1] 0 1 0 0 1 1 0 1 0 0 0 0 1 1 1 1 1 1 0 1 0 0 0 0 0 1 1 0 1 0 0 1 0 0 1 1 0  
 [38] 0 0 1 0 1 0 1 0 0 0 1 1 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 0 1 0 0 1 1 0  
 [75] 1 1 1 0 1 0 1 0 1 0 1 0 1 1 0 1 0 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0  
[112] 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0  
[149] 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0  
Levels: 0 1

Hide

##create confusion matrix  
cf <- table(pr,data\_test\_target\_category)  
cf

data\_test\_target\_category  
pr 0 1  
 0 110 6  
 1 0 59

Hide

# data\_test\_target\_category  
# pr 0 1  
# 0 110 6  
# 1 0 59  
  
accuracy(cf)

[1] 96.57143

Hide

#96% accuracy of the data imputed with regression  
  
#Data with rows w/ missing values removed  
  
data\_w\_no\_na = Data[complete.cases(Data), ]  
dim(data\_w\_no\_na)

[1] 683 10

Hide

result\_w\_no\_na <- split\_test\_train(data\_w\_no\_na)  
data\_train\_target\_category <- result\_w\_no\_na$train[,"Class"]  
data\_test\_target\_category <- result\_w\_no\_na$test[,"Class"]  
  
  
pr <- knn(result\_w\_no\_na$train,result\_w\_no\_na$test, cl=data\_train\_target\_category)  
pr

[1] 1 0 1 0 0 0 0 0 1 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 0 0 0 0  
 [38] 0 1 1 1 0 0 1 0 0 1 0 1 0 0 1 0 1 1 0 1 0 0 0 1 1 0 1 1 1 1 0 0 1 0 0 1 1  
 [75] 1 0 1 0 1 1 0 0 1 1 1 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 1 1 1 1 1 0 0 0 0  
[112] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 1 1 0 1 0 0 1 0 1 0 0  
[149] 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1  
Levels: 0 1

Hide

##create confusion matrix  
cf <- table(pr,data\_test\_target\_category)  
cf

data\_test\_target\_category  
pr 0 1  
 0 113 3  
 1 3 52

Hide

# data\_test\_target\_category  
# pr 0 1  
# 0 113 3  
# 1 3 52  
  
accuracy(cf)

[1] 96.49123

Hide

#96% accuracy of the data without missing values   
  
# Using KNN models with   
#(1) data without missing values rows  
#(2) data imputed with mean  
#(3) data imputed with regression  
#(4) data imputed with regression with perturbation  
#(1) performs the worst, while the other three perform similarly well   
#with accuracy of 96%   
#This seems suspiciously high. This might due to overfitting from imputation  
#based on multiple independent variables.

Question 15.1 Describe a situation or problem from your job, everyday life, current events, etc., for which optimization would be appropriate. What data would you need?

Optimization can be used for airline to decide the number of first-class tickets, premium tickets, and coach tickets they should sell to maximize their profits for their flights. The company might need to consider certain constraints, for example, only max. number of first class and coach seats are available on the flight. The number of staffs to serve first class might be limited, etc.