Pacticum 1

Jiaming Xu

2/3/2022

```
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Q1:

part 0, 1:

```
df <- read_csv("diabetes.csv")
```

```
## Rows: 768 Columns: 9

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (9): Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, D...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

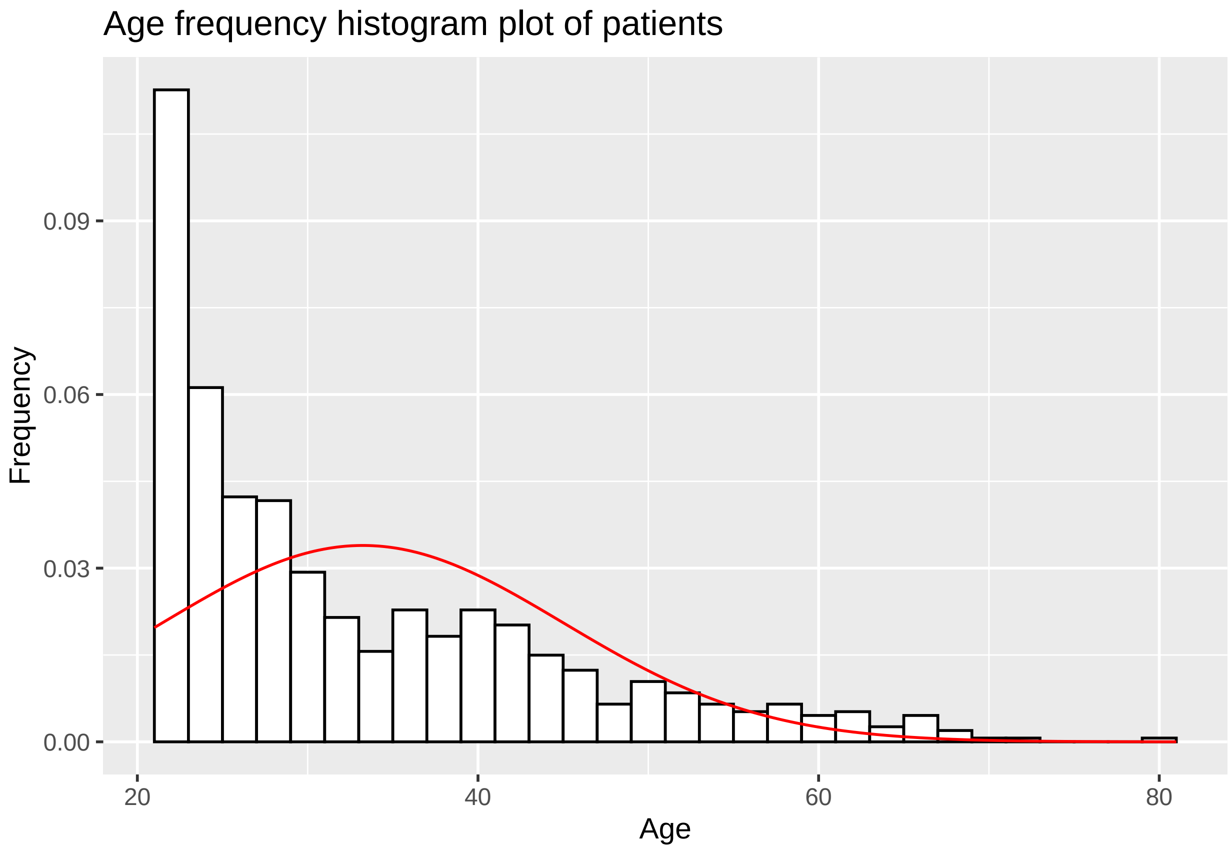
```
str(df)
```

```
## spec_tbl_df [768 x 9] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Pregnancies             : num [1:768] 6 1 8 1 0 5 3 10 2 8 ...
##  $ Glucose                 : num [1:768] 148 85 183 89 137 116 78 115 197 125 ...
##  $ BloodPressure           : num [1:768] 72 66 64 66 40 74 50 0 70 96 ...
##  $ SkinThickness           : num [1:768] 35 29 0 23 35 0 32 0 45 0 ...
##  $ Insulin                 : num [1:768] 0 0 0 94 168 0 88 0 543 0 ...
##  $ BMI                     : num [1:768] 33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
##  $ DiabetesPedigreeFunction: num [1:768] 0.627 0.351 0.672 0.167 2.288 ...
##  $ Age                     : num [1:768] 50 31 32 21 33 30 26 29 53 54 ...
##  $ Outcome                 : num [1:768] 1 0 1 0 1 0 1 0 1 1 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Pregnancies = col_double(),
##   ..   Glucose = col_double(),
##   ..   BloodPressure = col_double(),
##   ..   SkinThickness = col_double(),
##   ..   Insulin = col_double(),
##   ..   BMI = col_double(),
##   ..   DiabetesPedigreeFunction = col_double(),
##   ..   Age = col_double(),
##   ..   Outcome = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Part 3:

Use mutate function add a column for x, ranging from minimun to maximum value of ages, and another column for normal distribution with same mean and standard deviation as Age column as a function of x. Then use ggplot, geom_histogram to plot frequency distribution of Age column and use geom_line to plot normal distribution.

```
df %>%
    mutate( x = seq(min(Age), max(Age), length.out = nrow(df)),
            norm_age = dnorm(x, mean = mean(Age), sd = sd(Age))) %>%
    ggplot() +
        geom_histogram(mapping = aes(x = Age, y = ..density..), binwidth = 2, fill = 'white', color = 'l
        geom_line( mapping = aes(x = x, y = norm_age), color = 'red') +
        labs(title = "Age frequency histogram plot of patients",
             x = 'Age', y = 'Frequency')
```



Part 4:

“Normality Test in R” on STHDA:

“From the output, the p-value > 0.05 implying that the distribution of the data are not significantly different from normal distribution. In other words, we can assume the normality.”

The p value of age = 2.2e-16 which is really small almost zero, means the distribution of age is not a normal distribution.

```r
if(!require(devtools)) install.packages("devtools")
```

```
## Loading required package: devtools
```

```
## Loading required package: usethis
```

```r
devtools::install_github("kassambara/ggpubr")
```

```
## Skipping install of 'ggpubr' from a github remote, the SHA1 (ac5a01f5) has not changed since last ins
##   Use `force = TRUE` to force installation
```

```r
library(ggpubr)
shapiro.test(df$Age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$Age
## W = 0.87477, p-value < 2.2e-16
```

Q5:

1. make a function to calculate z score.
2. create a vector called “z_names” containing column names with “_z” as suffix.
3. use sapply to apply “zscore” function on each column of df, then bind them together, save as a tibble to “df_z”
4. rename column names of “df_z” by “z_names”.
5. use apply function to apply “dplyr::filter” function to each column to find the absolute value of z_score is larger than 2, then use the returned logical vector to find then in the initial tibble “df”. For better visulization, cbind the zscore of current column at the end.
6. use sapply to calculate how many outliters in each column.

For these outliers, there are several common methods:

1. Deleting observations
2. Transforming values, like log, cubic root, scaling, etc.
3. Imputation by mean, median, or mode.
4. Separately treating.

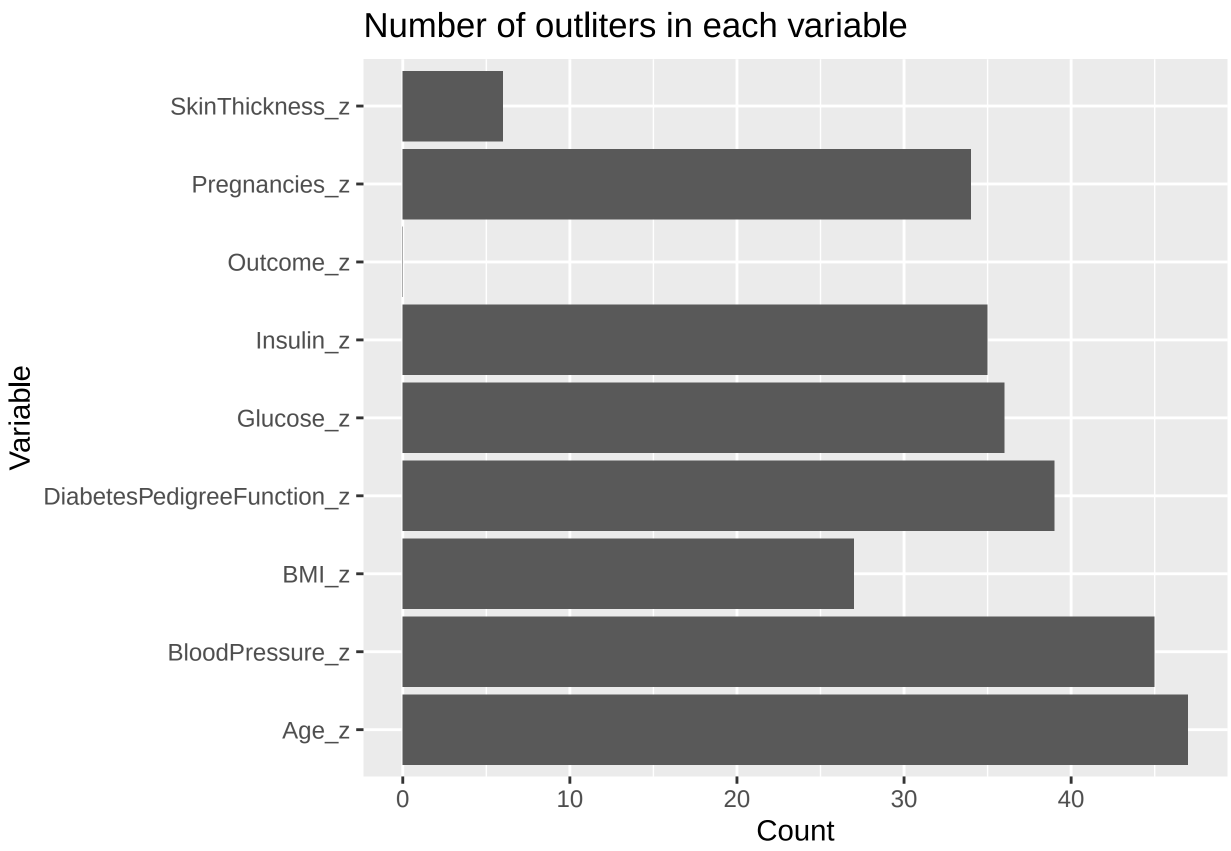
In our case, we should treate outliers of different varialbes separately. For example, out liers of pregnancies are more than 10, we may say this is reasonalbe and we can keep these data, or inputate them. However, outliers of blood pressure are zeros, which doesn’t make any sense. We have to replace these zeros by some values, median is preferred. However for outliers of age, DiabetesPedigreeFunction, BMI, etc. that’s not zero, they shouldn’t be rescaled because these values are medical significant. Therefore in my opinion, we should create another model for these valueable outliers to predict outcome of patitents like these outliers.

lapply is used to find outliers in “df” data frame since the length of each results are not the same. outliers is a list. Also, we can directly find the outliers in z standardized “df_z” data frame. It depends on what we want to see.

```r
zscore <- function(x){
  return ((x - mean(x))/sd(x))
}
z_names <- str_c(names(df), '_z')
df_z <- sapply(df, function(x){cbind( zscore(x))}) %>% as_tibble()
names(df_z) <- z_names

df_outliers <- lapply(df_z,function(x) {
    df[abs(x)>2,] %>%
        cbind(z = x[abs(x)>2])
    })
df_outliers
```

```
## $Pregnancies_z
##    Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI
## 1           11     143            94            33     146 36.6
## 2           13     145            82            19     110 22.2
## 3           11     138            76             0       0 33.2
## 4           13     126            90             0       0 43.4
## 5           13     106            72            54       0 36.6
## 6           15     136            70            32     110 37.1
## 7           17     163            72            41     114 40.9
## 8           11     135             0             0       0 52.3
## 9           12     151            70            40     271 41.8
## 10          12      92            62             7     258 27.6
## 11          11     155            76            28     150 33.3
## 12          13     106            70             0       0 34.2
## 13          14     100            78            25     184 36.6
## 14          13     152            90            33      29 26.8
## 15          12     106            80             0       0 23.6
## 16          13     129             0            30       0 39.9
## 17          12      88            74            40      54 35.3
## 18          12     140            82            43     325 39.2
## 19          12     140            85            33       0 37.4
## 20          14     175            62            30       0 33.6
## 21          12      84            72            31       0 29.7
## 22          13      76            60             0       0 32.8
## 23          11     103            68            40       0 46.2
## 24          11      85            74             0       0 30.1
## 25          12     121            78            17       0 26.5
## 26          11     111            84            40       0 46.8
## 27          11     138            74            26     144 36.1
## 28          13     104            72             0       0 31.2
## 29          11     136            84            35     130 28.3
## 30          11     127           106             0       0 39.0
## 31          13     158           114             0       0 42.3
## 32          11     120            80            37     150 42.3
## 33          13     153            88            37     140 40.6
## 34          12     100            84            33     105 30.0
##    DiabetesPedigreeFunction Age Outcome        z
## 1                     0.254  51       1 2.123396
## 2                     0.245  57       0 2.716942
## 3                     0.420  35       0 2.123396
## 4                     0.583  42       1 2.716942
## 5                     0.178  45       0 2.716942
## 6                     0.153  43       1 3.310488
## 7                     0.817  47       1 3.904034
## 8                     0.578  40       1 2.123396
## 9                     0.742  38       1 2.420169
## 10                    0.926  44       1 2.420169
## 11                    1.353  51       1 2.123396
## 12                    0.251  52       0 2.716942
## 13                    0.412  46       1 3.013715
## 14                    0.731  43       1 2.716942
## 15                    0.137  44       0 2.420169
## 16                    0.569  44       1 2.716942
## 17                    0.378  48       0 2.420169
## 18                    0.528  58       1 2.420169
## 19                    0.244  41       0 2.420169
## 20                    0.212  38       1 3.013715
## 21                    0.297  46       1 2.420169
## 22                    0.180  41       0 2.716942
## 23                    0.126  42       0 2.123396
## 24                    0.300  35       0 2.123396
## 25                    0.259  62       0 2.420169
## 26                    0.925  45       1 2.123396
## 27                    0.557  50       1 2.123396
## 28                    0.465  38       1 2.716942
## 29                    0.260  42       1 2.123396
## 30                    0.190  51       0 2.123396
## 31                    0.257  44       1 2.716942
## 32                    0.785  48       1 2.123396
## 33                    1.174  39       0 2.716942
## 34                    0.488  46       0 2.420169
##
## $Glucose_z
##    Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI
## 1            2     197            70            45     543 30.5
## 2            1     189            60            23     846 30.1
## 3            7     196            90             0       0 39.8
## 4            7     187            68            39     304 37.7
## 5            5      44            62             0       0 25.0
## 6            1       0            48            20       0 24.7
## 7            8     188            78             0       0 47.9
## 8            1       0            74            20      23 27.7
## 9            7     194            68            28       0 35.9
## 10           8     196            76            29     280 37.5
## 11           4     197            70            39     744 36.7
## 12           1     193            50            16     375 25.9
## 13           3     191            68            15     130 30.9
## 14           6     194            78             0       0 23.5
## 15           1       0            68            35       0 32.0
## 16           5       0            80            32       0 41.0
## 17           1     196            76            36     249 36.5
## 18           5     189            64            33     325 31.2
## 19           3     193            70            31       0 34.9
## 20           8     197            74             0       0 25.9
## 21           0     189           104            25       0 34.3
## 22           8     194            80             0       0 26.1
## 23           7     195            70            33     145 25.1
## 24           6       0            68            41       0 39.0
## 25           8     186            90            35     225 34.5
## 26           5     187            76            27     207 43.6
## 27           4     189           110            31       0 28.5
## 28           0     198            66            32     274 41.3
## 29           2     197            70            99       0 34.7
## 30           0     188            82            14     185 32.0
## 31           1     199            76            43       0 42.9
## 32           6     195            70             0       0 30.9
## 33           2      56            56            28      45 24.2
## 34           7     187            50            33     392 33.9
## 35           3     187            70            22     200 36.4
## 36           6     190            92             0       0 35.5
##    DiabetesPedigreeFunction Age Outcome         z
## 1                     0.158  53       1  2.380333
## 2                     0.398  59       1  2.130119
## 3                     0.451  41       1  2.349056
## 4                     0.254  41       1  2.067565
## 5                     0.587  36       0 -2.405012
## 6                     0.140  22       0 -3.781190
## 7                     0.137  43       1  2.098842
## 8                     0.299  21       0 -3.781190
## 9                     0.745  41       1  2.286502
## 10                    0.605  57       1  2.349056
## 11                    2.329  31       0  2.380333
## 12                    0.655  24       0  2.255226
## 13                    0.299  34       0  2.192672
## 14                    0.129  59       1  2.286502
## 15                    0.389  22       0 -3.781190
## 16                    0.346  37       1 -3.781190
## 17                    0.875  29       1  2.349056
## 18                    0.583  29       1  2.130119
## 19                    0.241  25       1  2.255226
## 20                    1.191  39       1  2.380333
## 21                    0.435  41       1  2.130119
## 22                    0.551  67       0  2.286502
## 23                    0.163  55       1  2.317779
## 24                    0.727  41       1 -3.781190
## 25                    0.423  37       1  2.036288
## 26                    1.034  53       1  2.067565
## 27                    0.680  37       0  2.130119
## 28                    0.502  28       1  2.411609
## 29                    0.575  62       1  2.380333
## 30                    0.682  22       1  2.098842
## 31                    1.394  22       1  2.442886
## 32                    0.328  31       1  2.317779
## 33                    0.332  22       0 -2.029691
## 34                    0.826  34       1  2.067565
## 35                    0.408  36       1  2.067565
## 36                    0.278  66       1  2.161395
## 
## $BloodPressure_z
##    Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI
## 1           10     115             0             0       0 35.3
## 2            7     100             0             0       0 30.0
## 3            1     103            30            38      83 43.3
## 4            9     171           110            24     240 45.4
## 5            7     105             0             0       0  0.0
## 6            2      84             0             0       0  0.0
## 7            0     131             0             0       0 43.2
## 8            2      74             0             0       0  0.0
## 9            5     137           108             0       0 48.8
## 10           1      96           122             0       0 22.4
## 11           1      88            30            42      99 55.0
## 12           2      87             0            23       0 28.9
## 13           0     129           110            46     130 67.1
## 14          11     135             0             0       0 52.3
## 15           7     119             0             0       0 25.2
## 16           3     141             0             0       0 30.0
## 17           0     138             0             0       0 36.3
## 18           2     146             0             0       0 27.5
## 19           0     167             0             0       0 32.3
## 20           1     180             0             0       0 43.3
## 21           0     117             0             0       0 33.8
## 22           3     116             0             0       0 23.5
## 23          13     129             0            30       0 39.9
## 24           5     103           108            37       0 39.2
## 25           0      94             0             0       0  0.0
## 26           2      99             0             0       0 22.2
## 27           0     141             0             0       0 42.4
## 28           2     119             0             0       0 19.6
## 29           8     120             0             0       0 30.0
## 30           0     145             0             0       0 44.2
## 31           3      80             0             0       0  0.0
## 32           6     114             0             0       0  0.0
## 33           6      91             0             0       0 29.8
## 34           4     132             0             0       0 32.9
## 35           4     189           110            31       0 28.5
## 36           0      73             0             0       0 21.1
## 37           1      89            24            19      25 27.8
## 38           6      96             0             0       0 23.7
## 39           4     183             0             0       0 28.4
## 40           0     119             0             0       0 32.4
## 41           4      90             0             0       0 28.0
## 42          13     158           114             0       0 42.3
## 43           0      99             0             0       0 25.0
## 44           2     129             0             0       0 38.5
## 45          10     115             0             0       0  0.0
##    DiabetesPedigreeFunction Age Outcome         z
## 1                     0.134  29       0 -3.570271
## 2                     0.484  32       1 -3.570271
## 3                     0.183  33       0 -2.020348
## 4                     0.721  54       1  2.112778
## 5                      0.305  24       0 -3.570271
## 6                      0.304  21       0 -3.570271
## 7                      0.270  26       1 -3.570271
## 8                      0.102  22       0 -3.570271
## 9                      0.227  37       1  2.009450
## 10                     0.207  27       0  2.732747
## 11                     0.496  26       1 -2.020348
## 12                     0.773  25       0 -3.570271
## 13                     0.319  26       1  2.112778
## 14                     0.578  40       1 -3.570271
## 15                     0.209  37       0 -3.570271
## 16                     0.761  27       1 -3.570271
## 17                     0.933  25       1 -3.570271
## 18                     0.240  28       1 -3.570271
## 19                     0.839  30       1 -3.570271
## 20                     0.282  41       1 -3.570271
## 21                     0.932  44       0 -3.570271
## 22                     0.187  23       0 -3.570271
## 23                     0.569  44       1 -3.570271
## 24                     0.305  65       0  2.009450
## 25                     0.256  25       0 -3.570271
## 26                     0.108  23       0 -3.570271
## 27                     0.205  29       1 -3.570271
## 28                     0.832  72       0 -3.570271
## 29                     0.183  38       1 -3.570271
## 30                     0.630  31       1 -3.570271
## 31                     0.174  22       0 -3.570271
## 32                     0.189  26       0 -3.570271
## 33                     0.501  31       0 -3.570271
## 34                     0.302  23       1 -3.570271
## 35                     0.680  37       0  2.112778
## 36                     0.342  25       0 -3.570271
## 37                     0.559  21       0 -2.330333
## 38                     0.190  28       0 -3.570271
## 39                     0.212  36       1 -3.570271
## 40                     0.141  24       1 -3.570271
## 41                     0.610  31       0 -3.570271
## 42                     0.257  44       1  2.319435
## 43                     0.253  22       0 -3.570271
## 44                     0.304  41       0 -3.570271
## 45                     0.261  30       1 -3.570271
## 
## $SkinThickness_z
##   Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI
## 1           0     100            88            60     110 46.8
## 2          13     106            72            54       0 36.6
## 3           0     162            76            56     100 53.2
## 4           0     147            85            54       0 42.8
## 5           0     180            78            63      14 59.4
## 6           2     197            70            99       0 34.7
##   DiabetesPedigreeFunction Age Outcome        z
## 1                    0.962  31       0 2.473859
## 2                    0.178  45       0 2.097736
## 3                    0.759  25       1 2.223110
## 4                   0.375  24       0 2.097736
## 5                   2.420  25       1 2.661921
## 6                   0.575  62       1 4.918660
## 
## $Insulin_z
##    Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI
## 1            2     197            70            45     543 30.5
## 2            1     189            60            23     846 30.1
## 3            7     150            66            42     342 34.7
## 4            8     155            62            26     495 34.0
## 5            5     105            72            29     325 36.9
## 6            1     153            82            42     485 40.6
## 7            8     181            68            36     495 30.1
## 8            4     148            60            27     318 30.9
## 9            0     177            60            29     478 34.6
## 10           4     197            70            39     744 36.7
## 11           6     134            80            37     370 46.2
## 12           0     165            90            33     680 52.3
## 13           9     124            70            33     402 35.4
## 14           1     193            50            16     375 25.9
## 15           5     155            84            44     545 38.7
## 16           2     146            70            38     360 28.0
## 17           5     189            64            33     325 31.2
## 18           3     173            82            48     465 38.4
## 19          12     140            82            43     325 39.2
## 20           1     131            64            14     415 23.7
## 21           1     172            68            49     579 42.4
## 22           3     173            84            33     474 35.7
## 23           3     158            70            30     328 35.5
## 24           1     139            62            41     480 40.7
## 25           6     129            90             7     326 19.6
## 26           1     143            86            30     330 30.1
## 27           8     124            76            24     600 28.7
## 28           7     168            88            42     321 38.2
## 29           2     157            74            35     440 39.4
## 30           2     155            52            27     540 38.7
## 31           7     142            90            24     480 30.4
## 32           2     127            46            21     335 34.4
## 33           3     158            64            13     387 31.2
## 34           7     187            50            33     392 33.9
## 35           0     181            88            44     510 43.3
##    DiabetesPedigreeFunction Age Outcome        z
## 1                     0.158  53       1 4.019303
## 2                     0.398  59       1 6.648507
## 3                     0.718  42       0 2.275177
## 4                     0.543  46       1 3.602795
## 5                     0.159  28       0 2.127664
## 6                     0.687  23       0 3.516023
## 7                     0.615  60       1 3.602795
## 8                     0.150  29       1 2.066923
## 9                     1.072  21       1 3.455282
## 10                    2.329  31       0 5.763428
## 11                    0.238  46       1 2.518140
## 12                    0.427  23       0 5.208085
## 13                    0.282  34       0 2.795812
## 14                    0.655  24       0 2.561526
## 15                    0.619  34       0 4.036657
## 16                    0.337  29       1 2.431367
## 17                    0.583  29       1 2.127664
## 18                    2.137  25       1 3.342478
## 19                    0.528  58       1 2.127664
## 20                    0.389  21       0 2.908616
## 21                    0.702  28       1 4.331683
## 22                    0.258  22       1 3.420573
## 23                    0.344  35       1 2.153696
## 24                    0.536  21       0 3.472636
## 25                    0.582  60       0 2.136341
## 26                    0.892  23       0 2.171050
## 27                    0.687  52       1 4.513905
## 28                    0.787  40       1 2.092955
## 29                    0.134  30       0 3.125547
## 30                    0.240  25       1 3.993271
## 31                    0.128  43       1 3.472636
## 32                    0.176  22       0 2.214436
## 33                    0.295  24       0 2.665653
## 34                    0.826  34       1 2.709039
## 35                    0.222  26       1 3.732954
##
## $BMI_z
##    Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI
## 1            8     125            96             0       0  0.0
## 2            7     105             0             0       0  0.0
## 3            2      84             0             0       0  0.0
## 4            2      74             0             0       0  0.0
## 5            5     137           108             0       0 48.8
## 6            1     122            90            51     220 49.7
## 7            0     162            76            56     100 53.2
## 8            1      88            30            42      99 55.0
## 9            0     102            75            23       0  0.0
## 10           8     188            78             0       0 47.9
## 11           7     152            88            44       0 50.0
## 12           0     129           110            46     130 67.1
## 13          11     135             0             0       0 52.3
## 14           0     165            90            33     680 52.3
## 15           5     115            98             0       0 52.9
## 16           0     165            76            43     255 47.9
## 17           0     118            64            23      89  0.0
## 18           4     156            75             0       0 48.3
## 19           0      94             0             0       0  0.0
## 20           0     180            78            63      14 59.4
## 21           3      80             0             0       0  0.0
## 22           6     114             0             0       0  0.0
## 23           3     123           100            35     240 57.3
## 24           0     162            76            36       0 49.6
## 25           5     136            82             0       0  0.0
## 26          10     115             0             0       0  0.0
## 27           1     147            94            41       0 49.3
##    DiabetesPedigreeFunction Age Outcome         z
## 1                     0.232  54       1 -4.057829
## 2                     0.305  24       0 -4.057829
## 3                     0.304  21       0 -4.057829
## 4                     0.102  22       0 -4.057829
## 5                     0.227  37       1  2.131796
## 6                     0.325  31       1  2.245949
## 7                     0.759  25       1  2.689877
## 8                     0.496  26       1  2.918183
## 9                     0.572  21       0 -4.057829
## 10                    0.137  43       1  2.017643
## 11                    0.337  36       1  2.284000
## 12                    0.319  26       1  4.452906
## 13                    0.578  40       1  2.575724
## 14                    0.427  23       0  2.575724
## 15                    0.209  28       1  2.651826
## 16                    0.259  26       0  2.017643
## 17                    1.731  21       0 -4.057829
## 18                    0.238  32       1  2.068378
## 19                    0.256  25       0 -4.057829
## 20                    2.420  25       1  3.476264
## 21                    0.174  22       0 -4.057829
## 22                    0.189  26       0 -4.057829
## 23                    0.880  22       0  3.209907
## 24                    0.364  26       1  2.233265
## 25                    0.640  69       0 -4.057829
## 26                    0.261  30       1 -4.057829
## 27                    0.358  27       1  2.195214
##
## $DiabetesPedigreeFunction_z
##    Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI
## 1            0     137            40            35     168 43.1
## 2           10     139            80             0       0 27.1
## 3            4     111            72            47     207 37.1
## 4            0     180            66            39       0 42.0
## 5            0     146            82             0       0 40.5
## 6            1     163            72             0       0 39.0
## 7            2     106            64            35     119 30.5
## 8            9     156            86            28     155 34.3
## 9            1     128            98            41      58 32.0
## 10           5      85            74            22       0 29.0
## 11           4     197            70            39     744 36.7
## 12           6     119            50            22     176 27.1
## 13           9     184            85            15       0 30.0
## 14          11     155            76            28     150 33.3
## 15          10     101            86            37       0 45.6
## 16           2     128            78            37     182 43.3
## 17           0     128            68            19     180 30.5
## 18           8     118            72            19       0 23.1
## 19           3     173            82            48     465 38.4
## 20           0     118            64            23      89  0.0
## 21           1      90            62            18      59 25.1
## 22           2     127            58            24     275 27.7
## 23           8     197            74             0       0 25.9
## 24           1      90            68             8       0 24.5
## 25           0     180            78            63      14 59.4
## 26           0     173            78            32     265 46.5
## 27           4     125            70            18     122 28.9
## 28           1      77            56            30      56 33.3
## 29           3     176            86            27     156 33.3
## 30           2      82            52            22     115 28.5
## 31           1     181            78            42     293 40.0
## 32           9     112            82            24       0 28.2
## 33           2      92            76            20       0 24.2
## 34           6     183            94             0       0 40.8
## 35           1     120            80            48     200 38.9
## 36           3      80            82            31      70 34.2
## 37           1     199            76            43       0 42.9
## 38          13     153            88            37     140 40.6
## 39           4     136            70             0       0 31.2
##    DiabetesPedigreeFunction Age Outcome        z
## 1                     2.288  33       1 5.481337
## 2                     1.441  57       0 2.924962
## 3                     1.390  56       1 2.771037
## 4                     1.893  25       1 4.289167
## 5                     1.781  44       0 3.951134
## 6                     1.222  33       1 2.263987
## 7                     1.400  34       0 2.801218
## 8                     1.189  42       1 2.164388
## 9                     1.321  33       1 2.562784
## 10                    1.224  32       1 2.270024
## 11                    2.329  31       0 5.605081
## 12                    1.318  33       1 2.553730
## 13                    1.213  49       1 2.236824
## 14                    1.353  51       1 2.659365
## 15                    1.136  38       1 2.004426
## 16                    1.224  31       1 2.270024
## 17                    1.391  25       1 2.774055
## 18                    1.476  46       0 3.030598
## 19                    2.137  25       1 5.025596
## 20                    1.731  21       0 3.800226
## 21                    1.268  25       0 2.402822
## 22                    1.600  25       0 3.404849
## 23                    1.191  39       1 2.170424
## 24                    1.138  36       0 2.010462
## 25                    2.420  25       1 5.879733
## 26                    1.159  58       0 2.073844
## 27                    1.144  45       1 2.028571
## 28                    1.251  24       0 2.351514
## 29                    1.154  52       1 2.058753
## 30                    1.699  25       0 3.703646
## 31                    1.258  22       1 2.372641
## 32                    1.282  50       1 2.445076
## 33                    1.698  28       0 3.700627
## 34                    1.461  45       0 2.985325
## 35                    1.162  41       0 2.082898
## 36                    1.292  27       1 2.475258
## 37                    1.394  22       1 2.783109
## 38                    1.174  39       0 2.119116
## 39                    1.182  22       1 2.143261
## 
## $Age_z
##    Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI
## 1           10     139            80             0       0 27.1
## 2            1     189            60            23     846 30.1
## 3           13     145            82            19     110 22.2
## 4            5     109            75            26       0 36.0
## 5            8     176            90            34     300 33.7
## 6            4     134            72             0       0 23.8
## 7            4     146            92             0       0 31.2
## 8            5     132            80             0       0 26.8
## 9            0     105            84             0       0 27.9
## 10           5     147            78             0       0 33.7
## 11           8     181            68            36     495 30.1
## 12           8     196            76            29     280 37.5
## 13           7     179            95            31       0 34.2
## 14           2     158            90             0       0 31.6
## 15           7     142            60            33     190 28.8
## 16           3     142            80            15       0 32.4
## 17           5     114            74             0       0 24.9
## 18           0     161            50             0       0 21.9
## 19           8     112            72             0       0 23.6
## 20           6     194            78             0       0 23.5
## 21           8      95            72             0       0 36.8
## 22           5     158            70             0       0 29.8
## 23           5     103           108            37       0 39.2
## 24           4     146            78             0       0 38.5
## 25          12     140            82            43     325 39.2
## 26           5     144            82            26     285 32.0
## 27           2     119             0             0       0 19.6
## 28           1     135            54             0       0 26.7
## 29           9     134            74            33      60 25.9
## 30           0     137            84            27       0 27.3
## 31           4     132            86            31       0 28.0
## 32           0     173            78            32     265 46.5
## 33           8     194            80             0       0 26.1
## 34           6     166            74             0       0 26.6
## 35           8     120            78             0       0 25.0
## 36           9      91            68             0       0 24.2
## 37           6     129            90             7     326 19.6
## 38           0      57            60             0       0 21.7
## 39           6     114            88             0       0 27.8
## 40           8     110            76             0       0 27.8
## 41           2     197            70            99       0 34.7
## 42          12     121            78            17       0 26.5
## 43           4     145            82            18       0 32.5
## 44           8      91            82             0       0 35.6
## 45           5     136            82             0       0  0.0
## 46           6     190            92             0       0 35.5
## 47          10     101            76            48     180 32.9
##    DiabetesPedigreeFunction Age Outcome        z
## 1                     1.441  57       0 2.020293
## 2                     0.398  59       1 2.190358
## 3                     0.245  57       0 2.020293
## 4                     0.546  60       0 2.275390
## 5                     0.467  58       1 2.105325
## 6                     0.277  60       1 2.275390
## 7                     0.539  61       1 2.360422
## 8                     0.186  69       0 3.040681
## 9                     0.741  62       1 2.445455
## 10                    0.218  65       0 2.700552
## 11                    0.615  60       1 2.275390
## 12                    0.605  57       1 2.020293
## 13                    0.164  60       0 2.275390
## 14                    0.805  66       1 2.785584
## 15                    0.687  61       0 2.360422
## 16                    0.200  63       0 2.530487
## 17                    0.744  57       0 2.020293
## 18                    0.254  65       0 2.700552
## 19                    0.840  58       0 2.105325
## 20                    0.129  59       1 2.190358
## 21                    0.485  57       0 2.020293
## 22                    0.207  63       0 2.530487
## 23                    0.305  65       0 2.700552
## 24                    0.520  67       1 2.870616
## 25                    0.528  58       1 2.105325
## 26                    0.452  58       1 2.105325
## 27                    0.832  72       0 3.295778
## 28                    0.687  62       0 2.445455
## 29                    0.460  81       0 4.061069
## 30                    0.231  59       0 2.190358
## 31                    0.419  63       0 2.530487
## 32                    1.159  58       0 2.105325
## 33                    0.551  67       0 2.870616
## 34                    0.304  66       0 2.785584
## 35                    0.409  64       0 2.615519
## 36                    0.200  58       0 2.105325
## 37                    0.582  60       0 2.275390
## 38                    0.735  67       0 2.870616
## 39                    0.247  66       0 2.785584
## 40                    0.237  58       0 2.105325
## 41                    0.575  62       1 2.445455
## 42                    0.259  62       0 2.445455
## 43                    0.235  70       1 3.125714
## 44                    0.587  68       0 2.955649
## 45                    0.640  69       0 3.040681
## 46                    0.278  66       1 2.785584
## 47                    0.171  63       0 2.530487
##
## $Outcome_z
##  [1] Pregnancies              Glucose                  BloodPressure
##  [4] SkinThickness            Insulin                  BMI
##  [7] DiabetesPedigreeFunction Age                      Outcome
## [10] z
## <0 rows> (or 0-length row.names)
```

```
df_outliers_distribution <- sapply(df_z, function(x) sum(abs(x) > 2))
ggplot()+
    geom_bar( mapping=aes(x = names(df_outliers_distribution), y = df_outliers_distribution), stat="iden
    labs( title = "Number of outliters in each variable", x = "Variable", y = "Count") +
    coord_flip()
```



Q6:

"zscore" function is defined by taking a vector as input, return z standardized vector. Using "sapply" to perform "zscore" function on each column then cbind results together, format it as a tibble, save it to a data frame "df_z".

```
zscore <- function(x){
  return ((x - mean(x))/sd(x))
}

df_z <- sapply(df, function(x){cbind( zscore(x))}) %>% as_tibble()
df_z$Outcome <- df$Outcome
str(df_z)
```

```
## tibble [768 x 9] (S3: tbl_df/tbl/data.frame)
##  $ Pregnancies             : num [1:768] 0.64 -0.844 1.233 -0.844 -1.141 ...
##  $ Glucose                 : num [1:768] 0.848 -1.123 1.942 -0.998 0.504 ...
##  $ BloodPressure           : num [1:768] 0.15 -0.16 -0.264 -0.16 -1.504 ...
##  $ SkinThickness           : num [1:768] 0.907 0.531 -1.287 0.154 0.907 ...
##  $ Insulin                 : num [1:768] -0.692 -0.692 -0.692 0.123 0.765 ...
##  $ BMI                     : num [1:768] 0.204 -0.684 -1.103 -0.494 1.409 ...
##  $ DiabetesPedigreeFunction: num [1:768] 0.468 -0.365 0.604 -0.92 5.481 ...
##  $ Age                     : num [1:768] 1.4251 -0.1905 -0.1055 -1.0409 -0.0205 ...
##  $ Outcome                 : num [1:768] 1 0 1 0 1 0 1 0 1 1 ...
```

Q7:

stratified sample:

Although it's unclear what's our prediction in the problem description, but we can infer that the goal is to predict if a given patient can discharge with 0 or 1 outcome code.

For stratified sample, there are two: 0 and 1. so sample n = floor(0.15 * nrow(df_z) / 2) from each layer from outcome.

```r
set.seed(1)

# Filter out two outcomes then randomly select 15% of observations in each subset.
# Then bind rows of train and test samples together, data frame reordered.

oc0 <- df_z %>% filter(Outcome == 0)
sample0 <- sample.int( nrow(oc0), floor(0.15 * nrow(oc0)), replace = FALSE)
oc1 <- df_z %>% filter(Outcome == 1)
sample1 <- sample.int( nrow(oc1), floor(0.15 * nrow(oc1)), replace = FALSE)

sample_test <- rbind(oc0[sample0, ], oc1[sample1, ])
sample_train <- rbind(oc0[-sample0, ], oc1[-sample1, ])

sample_test_label <- sample_test$Outcome    # argument cl takes only factor as an input of knn()
sample_train_label <- sample_train$Outcome

sample_test <- sample_test[ ,-9]    # Drop outcome columns
sample_train <- sample_train[ ,-9]
```

Test if stats of test set meet our needs.

~65% data are 0 in our sample_test, nearly the same as original data set. number of observations in our test sample is 14.97%, nearly 15%.

```r
sum(df$Outcome == 0)/nrow(df) # Number of 0 in original df.
```

```
## [1] 0.6510417
```

```r
sum(sample_test_label == 0)/(length(sample0) + length(sample1)) # Number of 0 in test data set.
```

```
## [1] 0.6521739
```

```r
nrow(sample_test)/ nrow(df)
```

```
## [1] 0.1497396
```

Q8:

getMode function: take the most common element in a vector.

My kNN:

1. create an empty modes_output vector for output.
2. loop through each row in the test data set.

3. inside loop: use sweep function to substract train data set by a row in the test data sett save to data set a. Then square the values in a, sort it, then take the indeces of the first k rows, to top_index. then use getMode function to find the most common value in of these indeces in train_labels. Append result into modes_output.

```
library(gmodels)
getMode <- function(x) {
    ## Took and modified from https://www.delftstack.com/howto/r/mode-in-r/.
    u <- unique(x)
    return (u[which.max(tabulate(match(x, u)))])
}

my_knn <- function( train, test, cl, k){
    modes_output <- vector()
    for (i in 1:nrow(test)) {
        a <- sweep( as.matrix(train), 2, as.matrix(test[i, ])) %>% as_tibble()
        top_index <- sort(rowSums(a^2), index.return=T)$ix[1:k]
        mode <- sample_train_label[top_index] %>% getMode()
        modes_output <- append(modes_output, mode)
    }
    return (modes_output)
}

goal <- tibble( Pregnancies = 4,
                Glucose = 118,
                BloodPressure = 50,
                SkinThickness = 30,
                Insulin = 78,
                BMI = 35,
                DiabetesPedigreeFunction = 0.279,
                Age = 29)

goal_z <- (goal - colMeans(df[, -9])) / sapply(df[, -9], function(x) sd(x))
goal_z
```

```
##   Pregnancies     Glucose BloodPressure SkinThickness     Insulin       BMI
## 1  0.04598437 -0.09053157    -0.9870665      0.593243 -0.01561451 0.3814511
##   DiabetesPedigreeFunction         Age
## 1                 -0.58213 -0.3606124
```

```
my_pred <- my_knn(sample_train, sample_test, sample_train_label, 5)
CrossTable(my_pred, sample_test_label)
```

```
## 
## 
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
## 
## 
```

```
## Total Observations in Table:  115
##
##
##               | sample_test_label
##       my_pred |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##             0 |        56 |        20 |        76 |
##               |     0.835 |     1.566 |           |
##               |     0.737 |     0.263 |     0.661 |
##               |     0.747 |     0.500 |           |
##               |     0.487 |     0.174 |           |
## -------------|-----------|-----------|-----------|
##             1 |        19 |        20 |        39 |
##               |     1.628 |     3.052 |           |
##               |     0.487 |     0.513 |     0.339 |
##               |     0.253 |     0.500 |           |
##               |     0.165 |     0.174 |           |
## -------------|-----------|-----------|-----------|
## Column Total |        75 |        40 |       115 |
##               |     0.652 |     0.348 |           |
## -------------|-----------|-----------|-----------|
##
##
```

```
my_knn(sample_train, goal_z, sample_train_table, 5)
```

```
## [1] 0
```

Q9:

Load class package, use knn function to compare accuracy of my_knn function. Results are identical, hence my_knn is reliable. The prediction of the new case is 1, same as my_knn function.

```
library(class)
set.seed(1)
df_pred <- knn( train = sample_train,
                test = sample_test,
                cl = sample_train_label, k=5)
CrossTable(df_pred, sample_test_label)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  115
##
##
##               | sample_test_label
##      df_pred |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |        56 |        20 |        76 |
##              |     0.835 |     1.566 |           |
##              |     0.737 |     0.263 |     0.661 |
##              |     0.747 |     0.500 |           |
##              |     0.487 |     0.174 |           |
## -------------|-----------|-----------|-----------|
##            1 |        19 |        20 |        39 |
##              |     1.628 |     3.052 |           |
##              |     0.487 |     0.513 |     0.339 |
##              |     0.253 |     0.500 |           |
##              |     0.165 |     0.174 |           |
## -------------|-----------|-----------|-----------|
## Column Total |        75 |        40 |       115 |
##              |     0.652 |     0.348 |           |
## -------------|-----------|-----------|-----------|
##
##
```

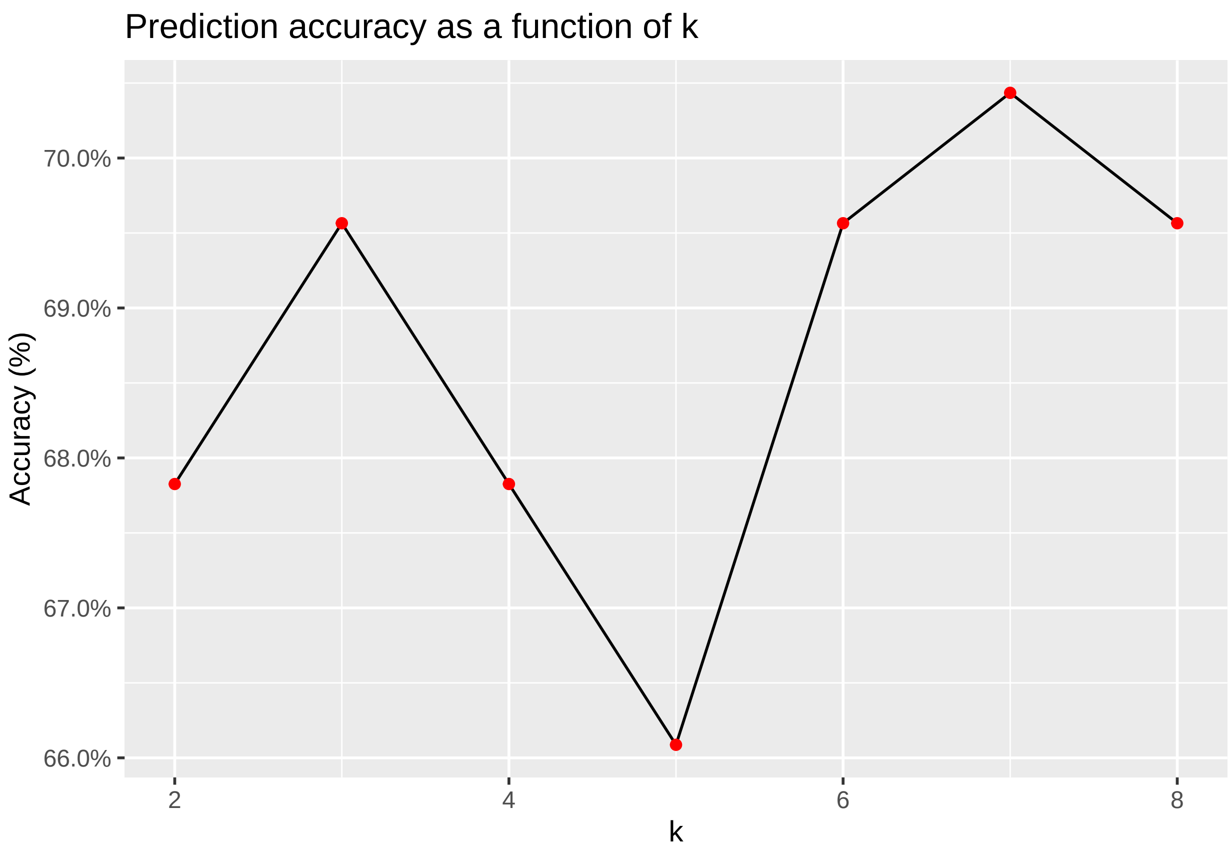
```r
knn( train = sample_train,
                test = goal_z,
                cl = sample_train_label, k=5)
```

```
## [1] 0
## Levels: 0 1
```

Q10:

Use for look calculate accurate rate with k = 2 to 8, save results in the vector "acc_rate". Plot it as line, define title, x and y labels.

```r
set.seed(1) # Make sure knn gives the same results.
acc_rate <- vector()
for (k in 2:8) {
  df_pred <-  knn(sample_train,
                sample_test,
                sample_train_label, k= k)
  acc_rate <- append(acc_rate, sum(df_pred == sample_test_label)/ length(sample_test_label))
}

ggplot( ) +
  geom_line( mapping = aes(x = seq(2,8), y = acc_rate)) +
  geom_point( mapping = aes( x = seq(2, 8), y = acc_rate), color = 'red') +
  labs( title = "Prediction accuracy as a function of k", x = 'k', y = 'Accuracy (%)') +
  scale_y_continuous(labels = scales::percent)
```



Problem 2:

Part 1, 2:

```
library(dplyr)
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
df <- as_tibble(Boston)

target_data <- df$medv
train_data <- df[, -ncol(df)]
```

Part 3:

use sapply to min-max standardize each column in the train data set.

```
train_data_norm <- sapply(train_data, function(x) ((x - min(x))/ (max(x) - min(x)))) %>% as_tibble()

# Test if normalization performed.
```

```
sapply(train_data_norm, function(x) range(x))
```

```
##      crim zn indus chas nox rm age dis rad tax ptratio black lstat
## [1,]    0  0     0    0   0  0   0   0   0   0       0     0     0
## [2,]    1  1     1    1   1  1   1   1   1   1       1     1     1
```

Part 5:

create knn.reg function that can basically use the same methodology as my_knn function described above, but not use getMode function for predicting kind of continuous variable in this “house price prediction” question, but weight the first k nearest items, and return a vector containing predicted prices.

```
knn.reg <- function(new_data, target_data, train_data, k ){
    if (k < 4) {stop("K must greater than 3.")}
    prices <- vector()
    for (i in 1:nrow(new_data)) {
        a <- sweep( as.matrix(train_data), 2, as.matrix(new_data[i, ])) %>% as_tibble()
        top_index <- sort(rowSums(a^2), index.return=T)$ix[1:k]
        weight_factor <- c(3,2,rep(1, k-2))
        price <- sum(target_data[top_index] * weight_factor / sum(weight_factor))
        prices <- append(prices, price)
    }
    return (prices)
}
```

Part 5:

create new data, normalize it use coefficients of min-max normalization of train data set, then apply it knn.reg function.

```
new_data <- tibble( crim = 0.15560, zn = 12.5,
                    indus = 7.87, chas = 0,
                    nox = 0.524, Rm = 6.173, age = 96.1,
                    dis = 5.9505, rad = 5, tax = 311,
                    pratio = 15.2, black = 396.9, lstat = 19.5)
new_data_norm <- (new_data - sapply(train_data, min)) / (sapply(train_data, max) - sapply(train_data, mi
knn.reg( new_data_norm, target_data, train_data_norm, 5)
```

```
## [1] 20.7
```

Part 6:

take out 10% randomly from train data set as test data set, keep the rest as train data set, apply same for labels. measure the MSE between results from knn.reg function and test label. MSE of knn.reg result is ~14.45 thousand dollars.

```
set.seed(1)
test_data_index <- sample.int( nrow(train_data_norm), floor(0.1 * nrow(train_data_norm)),
                              replace = FALSE)

sample_test <- train_data_norm[test_data_index, ]
sample_train <- train_data_norm[-test_data_index, ]
sample_train_label <- target_data[-test_data_index]
sample_test_label <- target_data[test_data_index]

knnreg_pred <- knn.reg( sample_test, sample_train_label, sample_train, 5)
```

```
my_mse <- mean(( knnreg_pred - sample_test_label)^2)
my_mse
```

```
## [1] 14.45019
```

Problem 3:

Part 1:

Load data.

```
library(tidyverse)
df <- read_csv('kc_house_data.csv')
```

```
## Rows: 21613 Columns: 21
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (1): id
## dbl  (19): price, bedrooms, bathrooms, sqft_living, sqft_lot, floors, waterf...
## dttm  (1): date
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Part 2:

Create year_month column contains Year-Month, then calculate average price per sqft of living room by the Yaer-Month column, save it to avg_price_sq_ft column. then separate Year-Month column into a Year and a Month column.

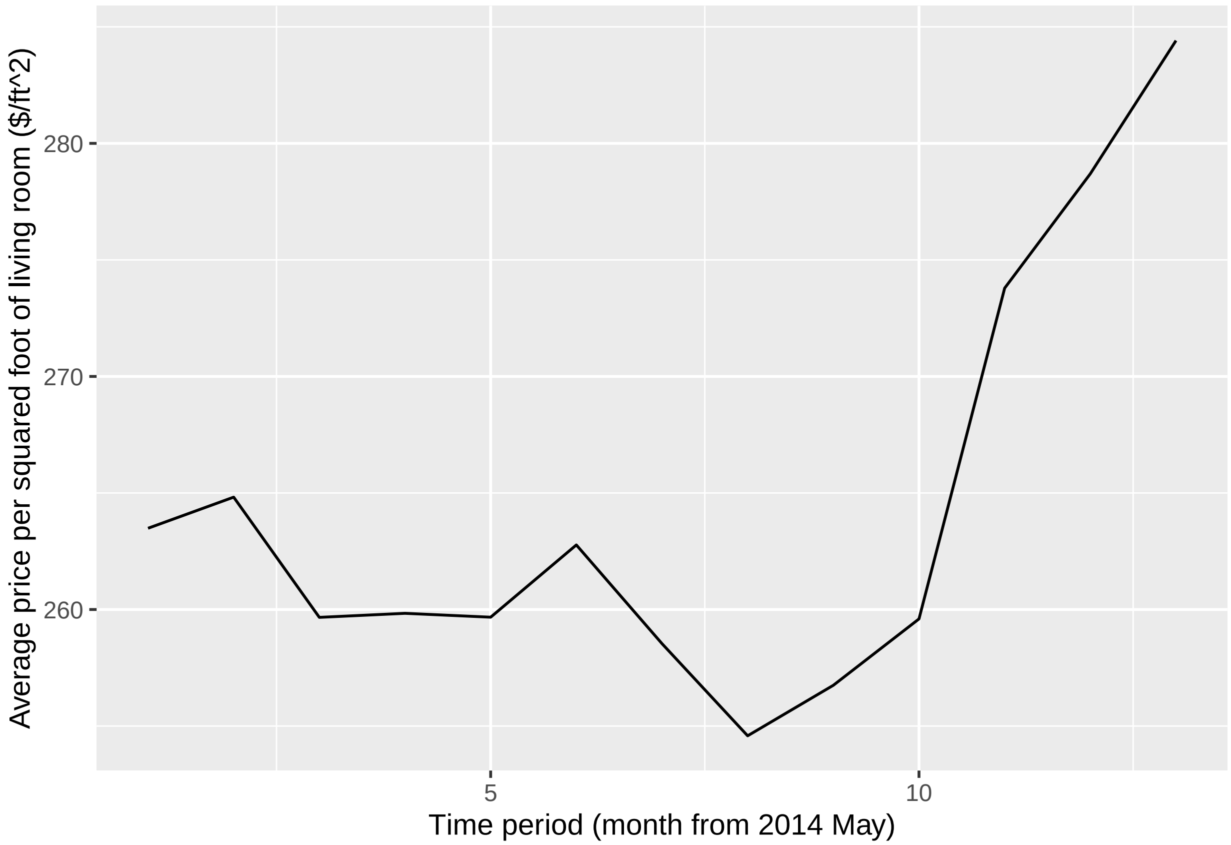
```
df_st <- df %>%
  mutate( year_month = format(date, "%Y-%m")) %>%
  group_by( year_month) %>%
  summarise( avg_price_sq_ft = mean(price/sqft_living)) %>%
  separate(year_month, c("year", "month"), sep = '-') %>%
  arrange( year,month) %>%
  mutate( tper = seq(1:length(year))) %>%
  relocate( tper, .before = year)

df_st
```

```
## # A tibble: 13 x 4
##     tper year  month avg_price_sq_ft
##    <int> <chr> <chr>           <dbl>
##  1     1 2014  05               263.
##  2     2 2014  06               265.
##  3     3 2014  07               260.
##  4     4 2014  08               260.
##  5     5 2014  09               260.
##  6     6 2014  10               263.
##  7     7 2014  11               259.
##  8     8 2014  12               255.
##  9     9 2015  01               257.
## 10    10 2015  02               260.
## 11    11 2015  03                274.
## 12    12 2015  04                279.
## 13    13 2015  05                284.
```

Part 3:

Plot average price per sqft of living room versus time period.

```
ggplot(df_st) +
    geom_line( mapping = aes( x = tper, y = avg_price_sq_ft)) +
    ylab("Average price per squared foot of living room ($/ft^2)") +
    xlab("Time period (month from 2014 May)")
```



Part 4:

Create weight_factor vector contains factors for last three months. Calculate weighted prediction for next month. (2015-Jun)

```
weight_factor = c(1, 3, 4) # ascending indexing.
pred <- sum(df_st$avg_price_sq_ft[ (nrow(df_st)-2): nrow(df_st)] *weight_factor ) / sum (weight_factor)
pred
```

```
## [1] 280.9403
```