

# Clear Memory & Set Data File Path

Put the path of the customer\_churn file folder as argument for setwd("customer\_churn file folder path").

```
rm(list = ls(all.names = TRUE)); gc();
```

```
##          used (Mb) gc trigger (Mb) max used (Mb)
## Ncells 514050 27.5   1139724 60.9   657980 35.2
## Vcells 934988  7.2     8388608 64.0   1768057 13.5
```

```
setwd("D:/OneDrive - University of Ottawa/Desktop/Assignment 2/")
```

## Import Required Libraries

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.1
```

```
library(cluster)
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.3.1
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3Wba
```

## Load & Prepare the Data for Clustering

### Load framingham.csv file Data in DataFrame

```
Load framingham.csv file Data in DataFrame.
```

```
# Specify the name to your CSV file
csv_file_name <- "framingham.csv"
# Load the data from the csv file into a data frame
framingham_data <- read.csv(csv_file_name)
```

### Explore framingham data

```
head(framingham_data)
```

```
##   male age education currentSmoker cigsPerDay BPmeds prevalentStroke
## 1    1  39          4              0          0      0          0
## 2    0  46          2              0          0      0          0
## 3    1  48          1              1          20      0          0
## 4    0  61          3              1          23      0          0
## 5    0  46          3              1          30      0          0
## 6    0  43          2              0          0      0          0
##   prevalentHyp diabetes totChol sysBP diaBP BMI heartRate glucose TenYearCHD
## 1      0          0    195 106.0    70  26.97      80      77          0
## 2      0          0    250 121.0    81  28.73      95      76          0
## 3      0          0    245 127.5    80  25.34      75      70          0
## 4      1          0    225 150.0    95  28.58      65     103          1
## 5      0          0    285 130.0    84  23.10      85      85          0
## 6      1          0    228 180.0   110  30.30      77      99          0
```

```
str(framingham_data)
```

```
## 'data.frame':    4240 obs. of  16 variables:
## $ male          : int  1 0 1 0 0 0 0 1 1 ...
## $ age           : int  39 46 48 61 46 43 63 45 52 43 ...
## $ education     : int  4 2 1 3 3 2 1 2 1 1 ...
## $ currentSmoker : int  0 0 1 1 1 0 0 1 0 1 ...
## $ cigsPerDay    : int  0 0 20 30 23 0 0 20 0 30 ...
## $ BPmeds       : int  0 0 0 0 0 0 0 0 0 ...
## $ prevalentStroke: int  0 0 0 0 0 0 0 0 0 ...
## $ prevalentHyp  : int  0 0 0 1 0 1 0 0 1 ...
## $ diabetes     : int  0 0 0 0 0 0 0 0 0 ...
## $ sysBP       : int  195 250 245 225 285 228 205 313 260 225 ...
## $ totChol     : num  106 121 128 150 130 ...
## $ diaBP       : num  70 81 80 95 84 110 71 71 89 107 ...
## $ BMI         : num  27 28.7 25.3 28.6 23.1 ...
## $ heartRate   : int  80 95 75 65 85 77 60 79 76 93 ...
## $ glucose     : int  77 76 70 163 85 99 85 78 79 88 ...
## $ TenYearCHD  : int  0 0 0 1 0 0 1 0 0 ...
```

### Extract the Sex and Age Columns from Framingham Data

& Save them in a separate DataFrame.

```
clustering_data <- data.frame(sex = framingham_data$male, age = framingham_data$age)
```

### Explore Resulted sex & age Dataframe

```
head(clustering_data)
```

```
##   sex age
## 1    1  39
## 2    0  46
## 3    1  48
## 4    0  61
## 5    0  46
## 6    0  43
```

```
str(clustering_data)
```

```
## 'data.frame':    4240 obs. of  2 variables:
## $ sex: int  1 0 1 0 0 0 0 1 1 ...
## $ age: int  39 46 48 61 46 43 63 45 52 43 ...
```

Show Summary of sex & age values

```
summary(clustering_data)
```

```
##           sex           age
## Min.      :0.0000   Min.   :32.00
## 1st Qu.:0.0000   1st Qu.:42.00
## Median :0.0000   Median :49.00
## Mean     :0.4292   Mean   :49.58
## 3rd Qu.:1.0000   3rd Qu.:56.00
## Max.     :1.0000   Max.    :70.00
```

```
print("sex values counts:")
```

```
## [1] "sex values counts:"
```

```
table(clustering_data$sex)
```

```
##           0           1
## 2420 1820
```

```
print("age values counts:")
```

```
## [1] "age values counts:"
```

```
table(clustering_data$age)
```

```
##
##  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51
##  1   5  18  42  84  92 144 170 192 174 180 159 166 162 182 141 173 132 140 146
## 52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70
## 149 139 132 145 123 123 117 119 111 110  99 110  93  57  38  45 18  7  2
```

### Standardize the age column in clustering\_data

```
clustering_data$age <- scale(clustering_data$age)
```

```
head(clustering_data)
```

```
##   sex      age
## 1    1 -0.12341761
## 2    0 -0.4176149
## 3    1 -0.1843228
## 4    0 -0.1332078
## 5    0 -0.4176149
## 6    0 -0.7675531
```

```
summary(clustering_data)
```

```
##           sex           age.V1
## Min.      :0.0000   Min.   :-2.0506599
## 1st Qu.:0.0000   1st Qu.: -0.8841992
## Median :0.0000   Median : -0.0676767
## Mean     :0.4292   Mean   :  0.0000000
## 3rd Qu.:1.0000   3rd Qu.:  0.7488458
## Max.     :1.0000   Max.    :  2.3818907
```

```
print("age values counts:")
```

```
## [1] "age values counts:"
```

```
table(clustering_data$age)
```

```
##
## -2.05065989241606   -1.93401382394485   -1.81736775547364   -1.70072168700243
## [111] 1 3 3 3 1 1 3 3 1 1 3 3 4 4 3 3 1 1 3 3 2 4 3 1 1 1 3 3 2 4 1 3 2 4
## [185] 1 58407561853122   -1.46742955006   -1.35078348158879   -1.234137411311758
## [244] 4 3 4 1 4 1 4 1 4 3 2 3 2 4 2 1 3 4 2 3 1 1 4 1 3 3 4 2 1 2 2 1 3 2 3
## [316] 1 11749134464637   -1.00084527617516   -0.884199207703947   -0.767553139232736
## [390] 192      174      180      144
## [464] 0.6509070706761524   -0.534261002298313   -0.417614933819161   -0.30096886534789
## [538] 166      162      182      141
## [612] 0.184322796876678   -0.0676767284054669   0.0489693409657446   0.165615408536956
## [686] 0.282261477080168   0.308907545479379   0.51555361395959   0.632199682421802
## [760] 149      139      125      132
## [834] 0.748845750893013   0.865491819364259   0.982137887835436   1.09878395630665
## [908] 123      123      117      119
## [982] 1.21543002477786   1.33207609324907   1.44872216172028   1.56536823019149
## [1056] 111      110      99      110
## [1130] 1.68201429866271   1.79860636713392   1.91530643506513   2.03195250407634
## [1204] 93      57      38      45
## [1278] 2.14859857254755   2.26524464101876   2.38189070948997   2
## [1352] 18      7      2
```

## A)

### 1- K-means clustering of Data

at K = 4 ##1 Clustering nstart is set to 20 to try running kmeans with k different initial clusters' centroids & choosing the 1 with best Performance

```
k <- 4
set.seed(777)
clustering.K.4 <- kmeans(clustering_data, centers = k, nstart = 20)
clustering.K.4
```

```
## K-means clustering with 4 clusters of sizes 1176, 974, 926, 1164
```

```
## Cluster means:
## sex      age
## 1 0.3979592 0.4094216
## 2 1.0000000 -0.8264750
## 3 0.4082073 1.4310867
## 4 0.0000000 -0.8665493
##
## Clustering vector:
## [1] 2 4 2 3 4 4 1 3 1 1 3 2 3 2 3 1 4 2 4 4 2 4 4 2 4 4 1 1 2 2 2 3 2 3 2 2 3 3 1 2 1
## [38] 1 4 2 4 4 1 1 1 1 3 2 3 2 3 1 4 3 4 4 1 2 2 4 1 3 1 1 1 3 3 2 4 1 3 2 4
## [75] 4 3 4 4 4 2 3 2 1 1 3 4 3 3 2 2 1 2 1 3 3 3 3 4 1 1 4 3 4 4 1 2 2 1 3 2 3
## [112] 1 2 3 4 1 3 4 2 1 4 4 3 1 2 2 2 4 1 2 3 1 1 2 4 3 4 2 1 2 2 1 3 2 4
## [149] 3 1 3 3 3 3 4 2 3 1 1 4 4 2 4 1 1 3 2 4 2 1 2 1 4 1 3 3 3 4 1 2 2 1 3
## [186] 1 4 1 2 1 2 1 3 2 4 1 2 3 4 1 1 3 2 4 3 1 2 2 2 3 4 1 2 2 3 4 1 2 4 2 1 4
## [223] 3 4 4 1 1 1 1 2 1 3 2 4 1 1 3 4 1 1 3 4 4 4 3 2 3 4 1 3 2 3 4 1 3 2 2 4
## [260] 3 1 2 4 3 1 2 2 1 2 1 1 3 4 4 1 1 3 4 4 1 3 4 4 3 1 4 2 4 2 4 2 1 4 2 4
## [297] 1 4 2 3 2 4 1 1 4 2 1 2 4 1 4 3 2 2 4 4 3 2 2 4 2 2 1 1 2 3 2 1 3 1 4 3
## [334] 1 2 4 1 1 1 3 3 4 4 2 3 4 1 2 1 3 4 2 2 2 1 3 1 4 1 2 4 4 4 3 2 2 3 3
## [371] 4 1 4 1 3 4 3 3 3 1 4 1 2 2 2 3 1 4 4 1 1 4 1 3 3 2 4 3 3 1 2 4 2 2 4
## [408] 4 2 2 3 4 4 1 3 2 4 1 2 4 2 3 4 1 3 3 4 3 1 4 4 2 2 1 2 2 3 4 1 4 3
## [445] 3 2 3 2 2 4 1 1 4 4 4 4 4 1 4 4 4 1 1 2 1 2 1 1 4 4 2 2 1 4 4 1 1 4 3
## [482] 3 2 2 3 1 4 4 1 4 3 4 2 1 2 1 2 1 3 4 1 3 2 1 4 1 1 1 4 1 1 3 3 1 2 3
## [519] 2 2 3 1 3 1 1 1 4 2 2 3 4 4 1 2 2 3 4 1 4 2 2 4 2 4 1 2 2 1 2 1 3 1
## [556] 4 1 1 4 4 1 1 4 2 1 3 3 4 4 1 1 1 4 3 4 2 3 4 3 1 4 2 3 1 4 2 3 1 4 3
## [593] 2 4 4 2 2 3 4 2 4 1 2 3 3 2 1 2 4 1 4 4 2 1 3 2 1 3 3 4 3 1 3 4 2 1 3
## [630] 1 3 2 4 1 4 3 4 4 1 3 2 2 4 1 2 2 4 2 4 3 3 1 1 4 2 1 2 4 3 1 1 3 1
## [667] 1 2 4 1 3 2 2 3 2 2 1 4 3 4 1 1 2 3 1 4 2 4 1 3 3 1 3 2 4 1 4 4 3 3 2
## [704] 4 3 4 1 2 4 1 1 2 2 4 2 4 2 4 4 4 3 3 1 1 4 4 1 4 4 1 4 3 1 2 2 2 2
## [741] 3 3 1 4 4 3 2 3 2 3 3 1 3 4 4 1 1 4 4 2 3 1 1 4 4 2 3 1 1 4 2 3 4 1 3
## [778] 4 2 4 4 2 2 4 1 2 2 1 3 1 2 2 4 2 2 3 3 3 1 1 4 4 2 3 1 2 1 4 3 4 3
## [815] 2 2 2 2 2 4 1 3 4 4 1 3 2 1 4 1 2 1 3 1 4 2 3 3 1 4 2 4 2 4 2 4 3 4 2
## [852] 4 2 3 2 4 2 4 1 3 4 2 3 1 3 1 3 1 4 2 2 3 4 3 1 2 1 3 1 2 1 4 1 3 3 2
## [889] 1 3 3 2 4 4 2 3 3 3 1 3 4 3 1 2 1 4 3 2 4 3 2 1 4 3 2 4 3 1 2 4 3 2
## [926] 4 3 1 3 1 4 2 3 1 3 2 2 4 4 1 2 2 4 4 1 4 1 2 4 4 1 4 1 3 4 3 1 2 1 4
## [963] 4 4 4 1 3 3 1 3 1 4 1 1 3 2 1 3 1 4 4 4 1 1 2 2 1 2 2 1 3 4 1 2 2 1 2
## [1000] 4 2 2 3 4 1 4 1 1 1 2 3 1 1 3 3 3 1 4 2 1 2 1 2 2 3 4 2 3 2 4 2 4 4
## [1037] 4 2 3 1 3 1 1 3 1 1 2 1 2 2 1 2 2 1 3 3 2 2 1 4 1 2 4 2 2 1 1 1 4 3
## [1074] 2 4 1 3 1 1 3 1 3 1 1 2 1 4 4 2 4 2 2 1 1 1 1 1 4 1 4 2 3 1 1 4 3 4
## [1111] 3 1 3 1 2 4 3 1 3 1 3 1 3 3 4 4 2 4 3 4 3 1 1 3 3 2 4 4 3 1 1 3 3 1
## [1148] 1 1 2 4 4 1 3 4 3 1 4 3 4 3 4 3 4 1 1 1 1 2 3 2 1 4 3 3 2 1 2 2 4 1 2
## [1185] 3 2 2 1 3 4 3 3 4 3 3 4 3 3 4 2 1 1 4 3 2 3 4 4 1 2 1 1 3 2 4 2 4 3
## [1222] 2 4 4 2 2 4 1 4 4 2 3 4 3 4 1 3 3 4 2 4 1 2 4 1 1 2 4 1 2 4 3 2 1 4 3
## [1259] 3 4 1 4 4 1 2 3 3 4 3 4 3 4 1 1 2 1 3 2 1 3 2 4 2 1 2 4 3 1 4 3 1 2 1
## [1296] 4 2 1 2 1 1 4 3 1 2 2 2 3 4 1 3 2 4 3 3 1 1 1 4 1 3 1 2 1 4 2 3 2 1 2
## [1333] 4 3 2 4 4 4 1 4 1 1 4 1 1 4 1 3 2 3 1 3 2 2 1 2 2 3 4 2 2 4 1 3 1 1
## [1370] 4 2 1 1 3 1 3 4 4 2 3 2 4 4 2 2 1 1 4 1 3 1 4 2 2 4 2 3 4 3 4 3 2
## [1407] 4 4 2 3 2 4 4 2 4 1 2 1 2 2 4 4 1 1 4 4 2 2 1 2 1 2 1 2 3 1 4 2 4 3
## [1444] 2 1 4 3 3 4 2 4 1 2 2 3 3 3 1 4 1 1 3 1 4 4 3 2 1 1 1 3 2 4 4 3 1 4
## [1481] 4 3 4 1 3 1 1 1 2 1 3 1 3 1 4 4 1 1 3 4 1 1 3 4 2 1 1 2 1 4 3 4 1 2
## [1518] 1 4 2 4 1 1 4 4 4 3 1 4 4 4 2 1 1 3 4 4 4 1 1 2 3 1 2 4 2 4 3 2 4 3
## [1555] 1 4 4 3 3 1 1 3 4 1 1 3 3 1 4 4 2 4 4 2 4 4 4 1 3 4 1 1 3 1 1 2 3 1 4
## [1592] 3 1 4 4 3 2 4 3 4 3 3 3 2 4 2 4 4 4 4 3 4 3 1 4 1 1 3 1 1 2 3 1 2
## [1629] 1 4 1 3 2 4 2 1 2 4 2 1 2 4 2 1 2 4 2 2 1 2 3 4 1 1 2 1 4 1 4 1 3
## [1666] 1 4 1 2 4 2 1 1 2 1 2 1 4 3 4 3 1 4 1 1 2 3 4 4 3 1 4 4 3 1 1 3 1 4
## [1703] 4 2 3 4 1 4 1 1 1 1 2 2 4 4 3 4 3 2 1 2 1 4 1 1 1 4 3 3 1 1 4 1 2 4
## [1740] 3 2 1 4 1 2 1 4 1 1 2 4 3 4 4 2 4 4 1 1 2 1 4 1 1 3 4 1 2 4 4 1 2 3
## [1777] 2 4 4 1 2 1 3 3 2 1 1 4 3 1 3 2 1 2 3 4 3 1 2 2 2 1 3 4 2 1 2 2 1
## [1814] 2 4 1 1 3 3 4 1 3 2 1 1 1 4 2 3 4 4 1 3 4 3 4 3 2 2 1 2 1 2 3 4 2
## [1851] 4 4 1 3 3 3 1 3 3 4 3 2 1 1 4 4 1 1 2 4 1 3 4 3 4 3 4 1 1 3 1 4 4 2 1
## [1888] 4 3 3 1 1 4 2 1 2 1 3 3 1 1 3 1 4 4 3 3 1 1 4 2 1 4 4 1 3 2 3 1 4 1 3
## [1925] 4 3 3 4 4 2 3 3 2 1 3 1 3 4 3 1 4 3 2 4 2 1 3 2 4 3 1 3 2 1 4 3 2 4
## [1962] 4 3 2 1 4 1 2 1 3 4 4 2 2 2 3 1 3 1 2 4 4 3 3 3 3 4 4 1 4 1 2 1 3
## [1999] 1 4 4 2 3 1 4 2 2 3 1 4 1 1 2 2 2 4 2 4 4 4 3 3 1 2 4 2 4 3 1 4 3
## [2036] 2 1 4 2 2 1 3 1 1 4 3 2 4 4 1 1 1 2 2 4 1 1 3 2 2 1 4 1 2 3 4 2 1
## [2073] 3 4 2 4 3 3 4 2 1 2 4 3 4 4 1 1 1 4 3 4 3 1 4 2 2 4 1 1 4 4 4 1 3 1
## [2110] 4 2 1 1 2 3 1 4 1 1 1 3 4 4 3 3 1 2 3 1 3 2 3 1 4 4 2 4 4 1 4 2 1 3
## [2147] 2 3 1 4 2 4 1 4 2 1 2 4 2 4 3 3 1 1 2 4 4 2 2 3 4 3 3 2 1 1 4 3 1 2
## [2184] 3 1 4 1 1 2 2 1 2 2 2 2 2 3 2 1 4 4 1 1 2 4 4 1 2 4 4 1 2 4 3 2 2 2
## [2221] 2 3 4 4 2 2 1 3 4 2 1 4 2 1 4 2 3 1 4 4 1 2 2 4 4 1 4 1 3 4 1 2 2 1 2
## [2258] 1 2 2 4 1 3 3 2 2 2 3 2 2 1 4 4 2 1 2 1 1 2 1 1 2 2 2 4 1 1 3 4 2
## [2295] 4 3 4 1 3 2 4 4 2 4 3 4 3 4 3 4 2 4 1 3 3 4 4 2 4 4 4 1 1 1 4 4 4
## [2332] 4 2 1 4 1 1 4 4 2 1 3 4 2 2 3 4 2 3 1 3 3 4 2 3 1 4 2 4 2 1 3 2 2 4
## [2369] 1 1 3 1 4 1 2 3 3 1 3 4 3 4 1 2 1 1 2 4 4 1 1 2 3 3 4 4 1 1 3 2 4 1 1
## [2406] 1 3 1 4 2 2 1 2 2 3 3 1 3 2 3 1 3 2 3 1 2 4 2 3 2 4 1 4 2 1 3 2 1 1
## [2443] 2 3 1 4 3 3 1 1 2 3 2 3 4 4 1 3 1 3 3 4 3 4 2 3 3 2 2 1 1 3 1 1 2 3
## [2480] 1 2 4 4 1 4 1 1 4 3 2 3 4 4 1 2 1 4 3 1 4 2 3 1 1 4 1 4 4 1 4 2 4 1
## [2517] 4 3 3 2 1 1 4 2 2 4 2 3 4 1 4 4 2 1 3 2 2 4 1 1 4 1 4 4 4 1 4 3 1 2
## [2554] 4 3 1 1 2 1 2 4 4 4 1 1 2 3 4 1 2 4 3 4 2 1 2 1 3 2 1 4 4 4 1 1 1 3
## [2591] 4 4 4 1 2 3 4 3 1 1 3 1 3 1 4 4 3 1 1 3 1 2 3 1 2 4 2 4 3 2 1 4 3 4
## [2628] 4 1 4 1 3 2 4 3 4 3 4 3 4 1 4 1 2 2 3 3 4 3 2 2 1 4 1 2 1 4 2 1 3 1
## [2665] 2 3 4 2 1 1 3 3 1 2 3 1 1 4 4 3 3 1 2 1 3 3 1 3 4 3 1 4 3 1 4 3 3
## [2702] 1 2 4 1 2 3 4 2 2 2 2 1 2 1 4 3 3 3 2 4 3 2 4 4 4 2 4 1 4 1 3 2 1 2
## [2739] 3 4 4 2 4 2 3 4 3 4 3 2 1 4 4 2 1 1 4 4 2 2 2 4 3 4 3 4 3 4 3 4 4
## [2776] 1 1 4 2 2 1 2 1 3 1 4 1 2 1 1 1 4 2 1 3 2 1 3 4 3 4 1 4 2 1 4 4 4
## [2813] 2 4 2 4 2 2 3 1 2 2 1 3 4 1 1 2 4 4 3 1 4 3 2 3 2 1 1 1 2 2 1 1 1
## [2850] 2 1 4 1 2 3 1 1 4 3 1 1 4 3 1 1 4 2 1 2 3 1 3 4 2 4 2 2 1 2 3 2 2
## [2887] 2 4 1 1 1 2 3 3 4 4 1 2 3 4 1 4 1 2 4 2 3 3 1 4 4 4 1 4 1 3 1 2 1 2
## [2924] 2 3 1 2 1 2 1 1 4 1 2 3 4 3 4 2 1 1 2 1 3 1 4 4 1 1 3 2 4 3 4 1 1
## [2961] 1 1 1 1 3 2 1 4 2 1 4 1 3 1 3 2 2 1 2 4 4 1 1 1 3 1 4 1 2 4 3 4 3 1
## [2998] 2 2 2 1 2 1 3 1 4 1 2 2 2 1 2 1 4 1 1 1 2 1 3 1 2 1 3 4 4 4 3 1 1
## [3035] 1 1 4 2 4 2 4 1 3 4 1 4 2 1 3 3 3 1 1 1 1 4 3 2 1 1 1 1 2 1 2 4 3
## [3072] 4 1 2 3 2 4 3 4 1 3 1 2 1 2 4 2 1 2 3 1 3 1 2 1 1 4 4 1 1 1 1 4 3
## [3109] 2 4 2 3 2 2 2 2 4 1 4 1 4 1 3 1 2 2 4 2 1 3 1 3 1 3 2 4 4 1 3 3
## [3146] 4 3 1 1 4 1 2 4 1 2 2 2 2 3 2 1 2 4 2 2 3 1 4 1 2 3 4 1 2 2 1 2 2
## [3183] 1 4 4 2 4 3 3 1 4 1 4 4 4 4 2 2 3 2 2 4 2 1 2 1 2 4 1 4 2 2 1 4 4
## [3220] 3 3 4 4 1 4 1 4 4 1 3 4 4 3 4 4 4 4 2 3 4 1 1 3 3 1 4 4 2 4 2 4 3
## [3257] 4 3 1 4 1 3 3 2 1 4 3 2 2 4 2 4 2 4 2 4 3 4 1 3 2 2 3 4 1 4 3 3 2
## [3294] 1 4 4 2 4 4 1 4 3 2 4 1 4 3 2 4 1 4 3 2 4 4 3 4 4 3 1 4 3 3 2 2 3
## [3331] 1 4 4 4 3 4 1 2 4 3 3 3
```