# R project

## Healthcare Dataset Stroke Data Analysis and Visualizations

Akshay Kapre, Dharmit Dalvi - Term project

# Dataset details

The dataset is taken from Kaggle datasets, the link for which is as follows:
https://www.kaggle.com/asaumya/healthcare-dataset-stroke-data#train_2v.csv
(https://www.kaggle.com/asaumya/healthcare-dataset-stroke-data#train_2v.csv)

The dataset contains approximately 44,000 rows, and 12 columns. Each row contains data for a single patient. The columns include attributes for patients such as their age, gender, BMI, smoking status, etc, with which we can predict the final attribute: “stroke”, which predicts if the patient might suffer from a stroke or no.

# Objective

The objective of this project is to gain insights on patients’ health through analysis and visualization.

# Preprocessing

We read the dataset, our health care stroke data set and replace all the mising values in one of our columns- smoking status.

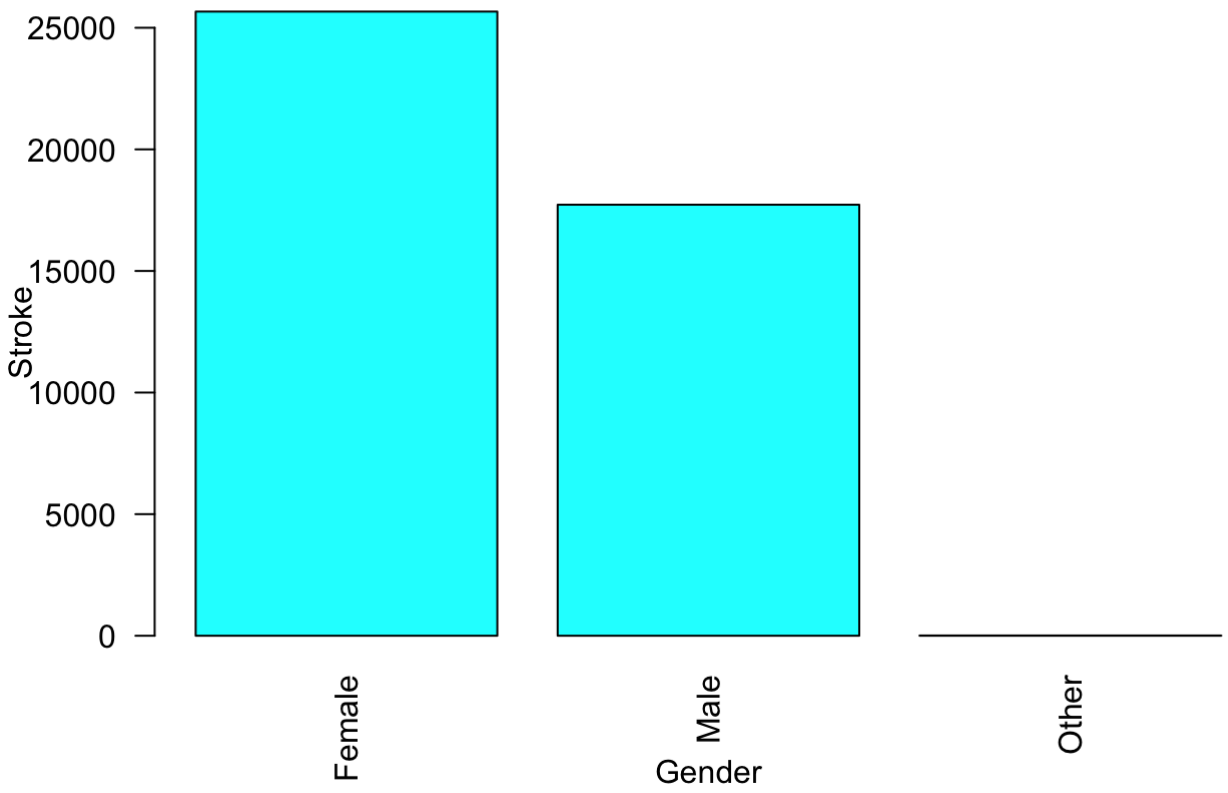
```
data <- as.data.frame(read.csv("stroke_data.csv", header=T, sep=",", na.strings = c("",
"NA")))
attach(data)
data$smoking_status <- as.character(data$smoking_status)
data$smoking_status <- ifelse(is.na(data$smoking_status),
                              'No information available', data$smoking_status)
```

The frequency for genders and a barplot for the same is calculated as follows: We have further calculated the same for only those patients who have stroke. (stroke = 1)

```
#categorical 1
table(data$gender)
```

```
##
## Female   Male  Other
##  25665  17724     11
```

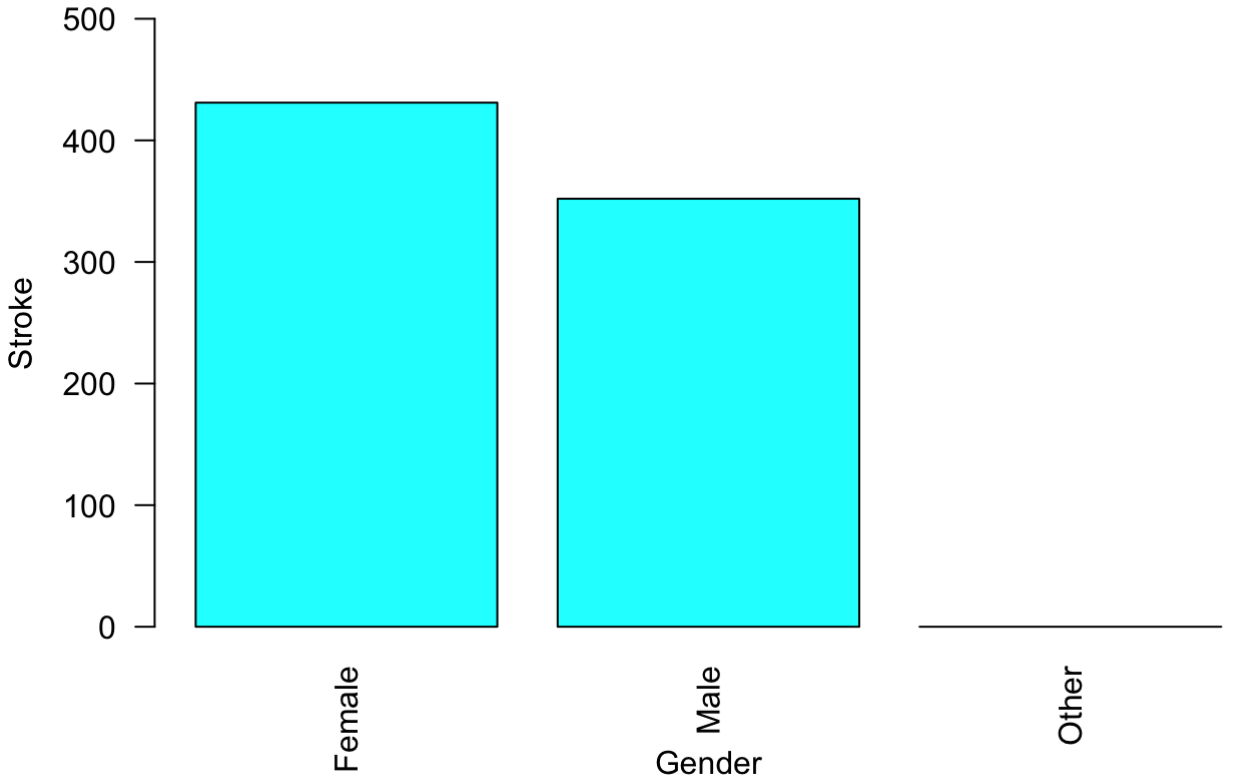
```
barplot(table(data$gender), col = "cyan", ylim = c(0,25000), las = 2, xlab = "Gender", y
lab = "Stroke")
```




```
yes_stroke <- data[data$stroke == 1, ]
table(yes_stroke$gender)
```

```
##
## Female   Male  Other
##    431    352      0
```

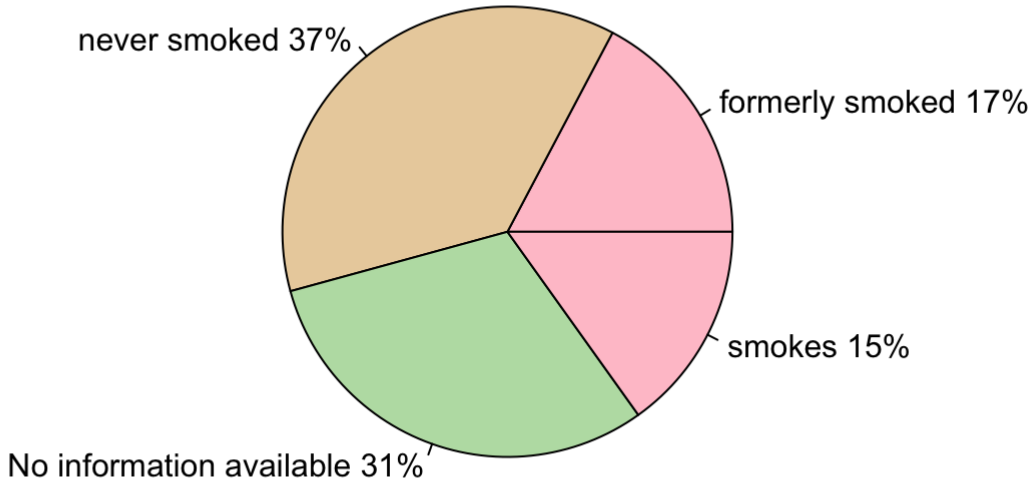
```
barplot(table(yes_stroke$gender), col = "cyan", ylim = c(0,500), las = 2, xlab = "Gende
r", ylab = "Stroke")
```




We can infer that the data has maximum number of females, followed by males and others.

We have further analysed the smoking status attribute, which is a categorical variable and visualized using a pie chart:

```
#categorical 2
smoking_status <- table(data$smoking_status)
slice.labels <- names(smoking_status)
slice.percents <- round(smoking_status/sum(smoking_status)*100)
slice.labels <- paste(slice.labels, slice.percents)
slice.labels <- paste(slice.labels, "%", sep = "")
pie(smoking_status, labels = slice.labels, col = hcl(c(0, 60, 120)))
```



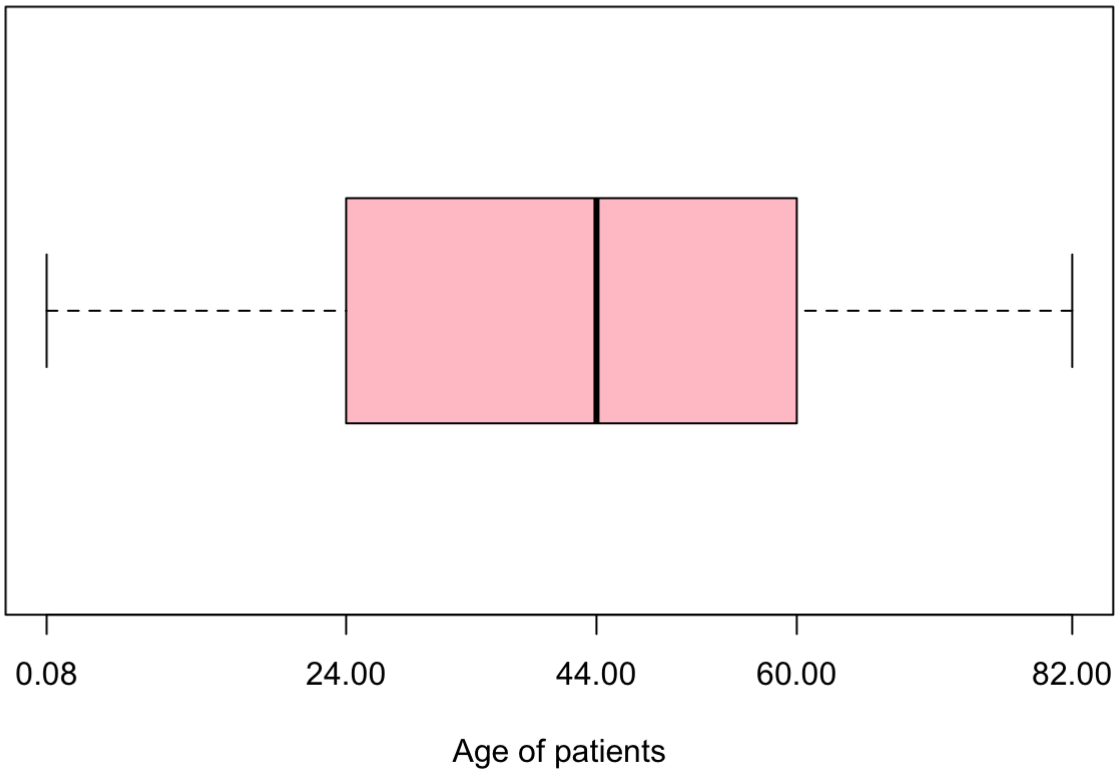

Percentage of people who have never smoked is the highest (37 %).

We further plot a boxplot for the age attribute and perform a five number summary:

```
fivenum(data$age)
```

```
## [1]  0.08 24.00 44.00 60.00 82.00
```

```
boxplot(data$age, horizontal = TRUE,  xaxt = "n", xlab = "Age of patients", col=hcl(1))
axis(side = 1, at=fivenum(data$age), labels = TRUE)
```



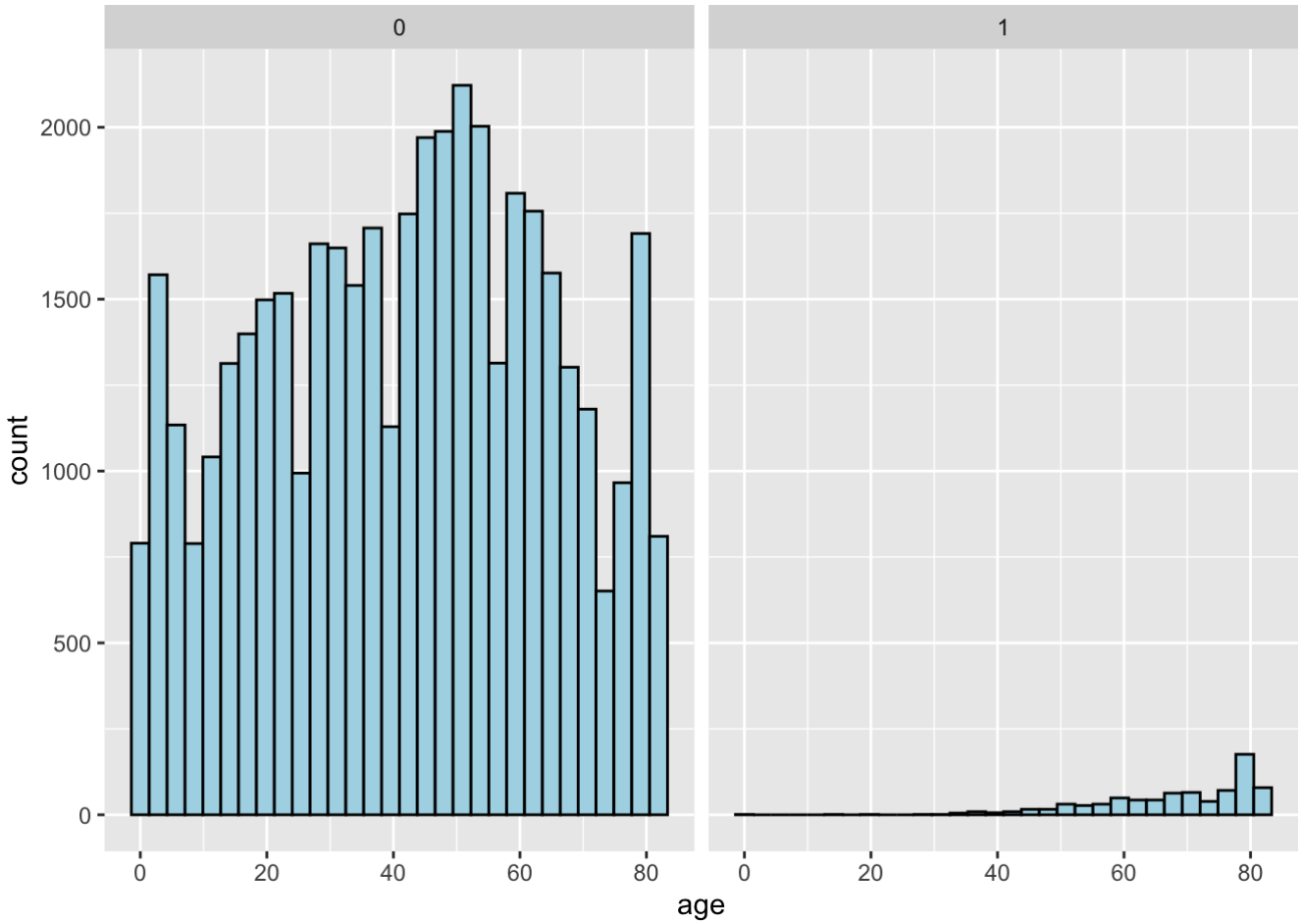

```
f <- fivenum(data$age)
f
```

```
## [1]  0.08 24.00 44.00 60.00 82.00
```

Next, we plot a ggplot for age, with respect to stroke values (1 or 0): We can see that for age = 80, the count for patients having stroke is maximum.

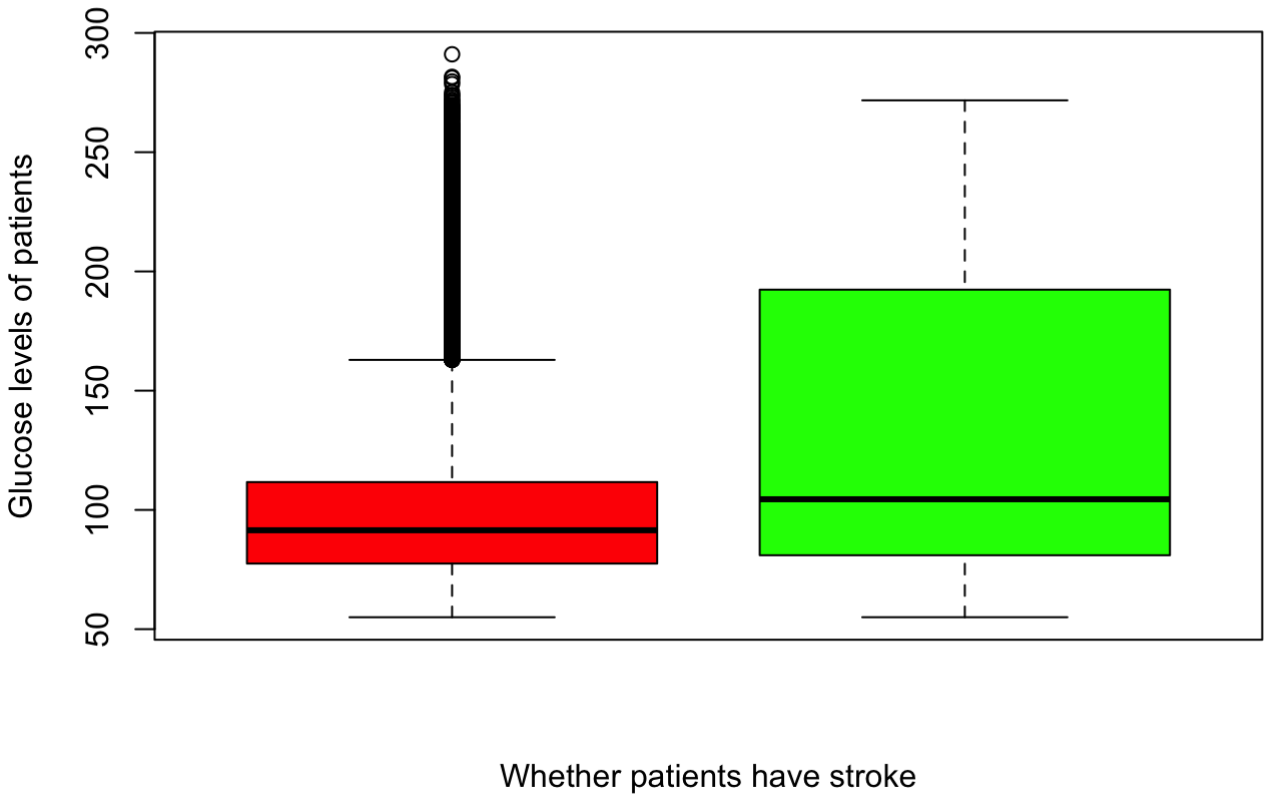
```
library(ggplot2)
ggplot(data, aes(x=age)) +
  geom_histogram(color="black", fill="lightblue") +   facet_grid(~stroke)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




We further plot a boxplot of average glucose levels in patients that have a stroke and average glucose levels in patients that don't have stroke.

```
no_stroke <- data[data$stroke == 0, ]
A <- no_stroke$avg_glucose_level
B <- yes_stroke$avg_glucose_level
boxplot(A,B, xaxt = "n", xlab = "Whether patients have stroke", ylab = "Glucose levels o
f patients", col=c("red", "green"))
```



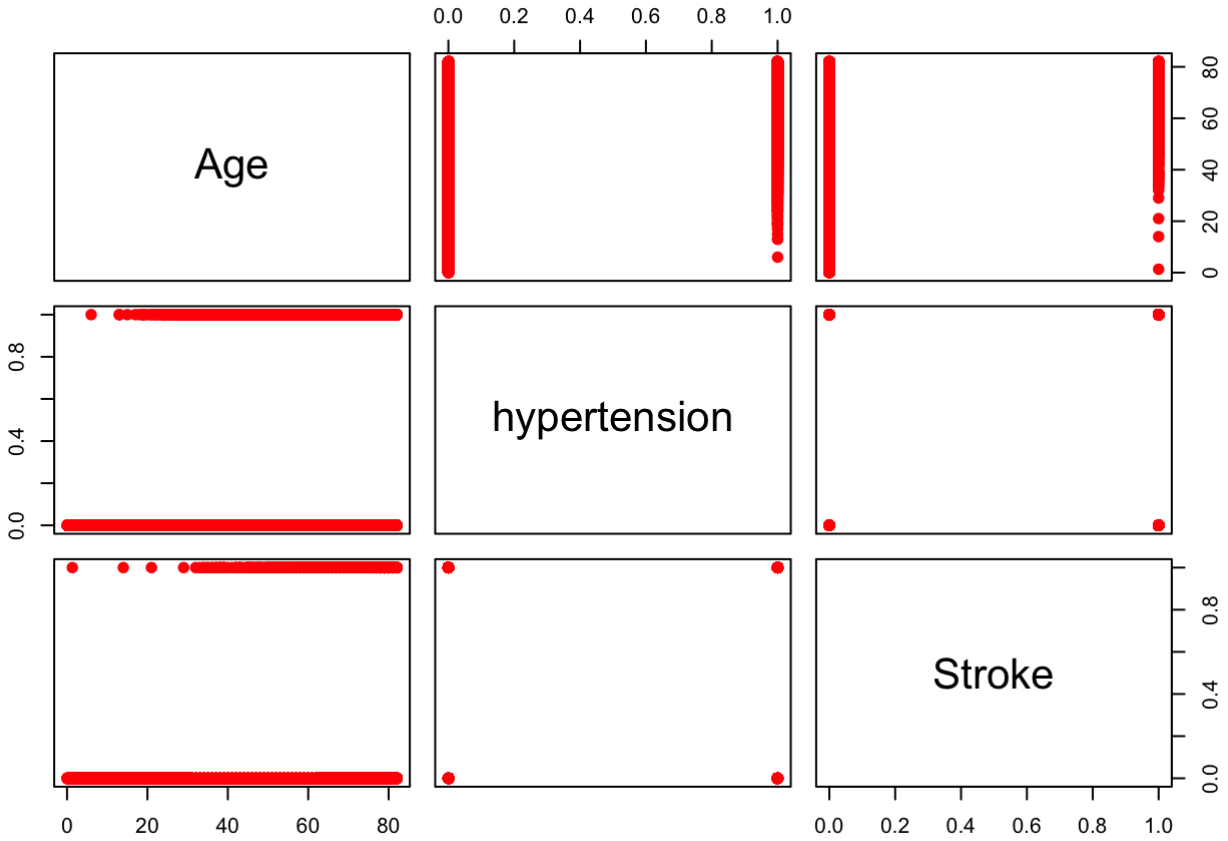

The patients that don't have stroke have lower average of glucose levels, the ones that have stroke have a greater average. The range of glucose values is between ~80 to 200 for patients having stroke, whereas there are a lot of outliers in glucose levels' data for patients not having stroke.

Following is the scatterplot between three variables: age, hypertension and stroke.

```
#2
dt=data.frame(Age=data$age,hypertension=data$hypertension,Stroke=data$stroke)
head(dt,10)
```

```
##    Age hypertension Stroke
## 1    3            0      0
## 2   58            1      0
## 3    8            0      0
## 4   70            0      0
## 5   14            0      0
## 6   47            0      0
## 7   52            0      0
## 8   75            0      0
## 9   32            0      0
## 10  74            1      0
```

```
plot(dt , pch=20 , cex=1.5 , col="red")
```

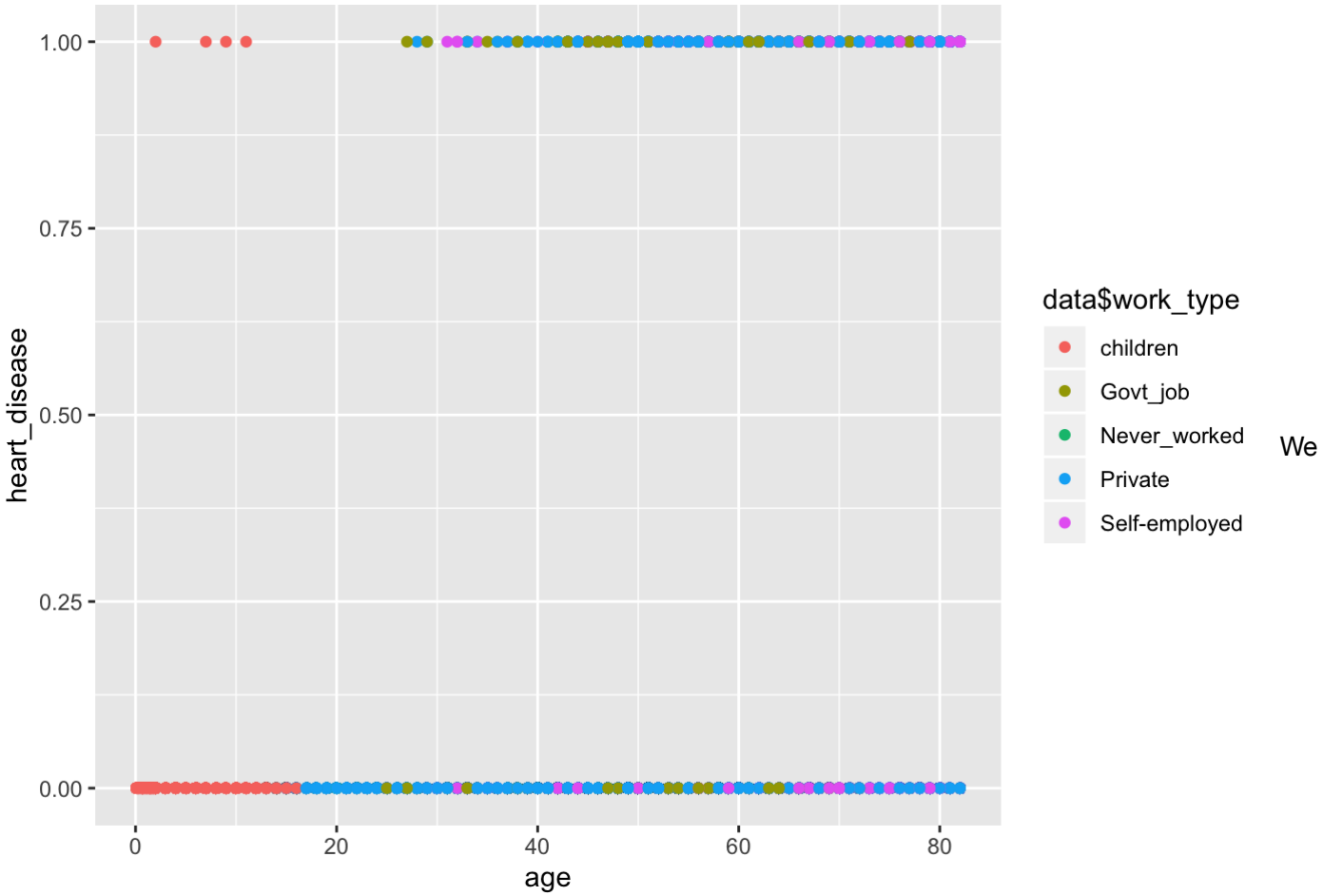



Major

inferences that we can make from the above plot are ages for patients with stroke start from 35, whereas for hypertension, majority of patients having hypertension start from the age of 15 itself.

We have further plotted a ggplot for age vs heart disease, and we have the points colored based on work_type.

```
#3
ggplot(data = data) +
  geom_point(mapping = aes(x = age, y = heart_disease, colour = data$work_type ))
```




We can infer that most patients with heart disease work in private sector.

Next we have boxplots of age, bmi and average glucose levels combined in a single plotly plot:

```
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

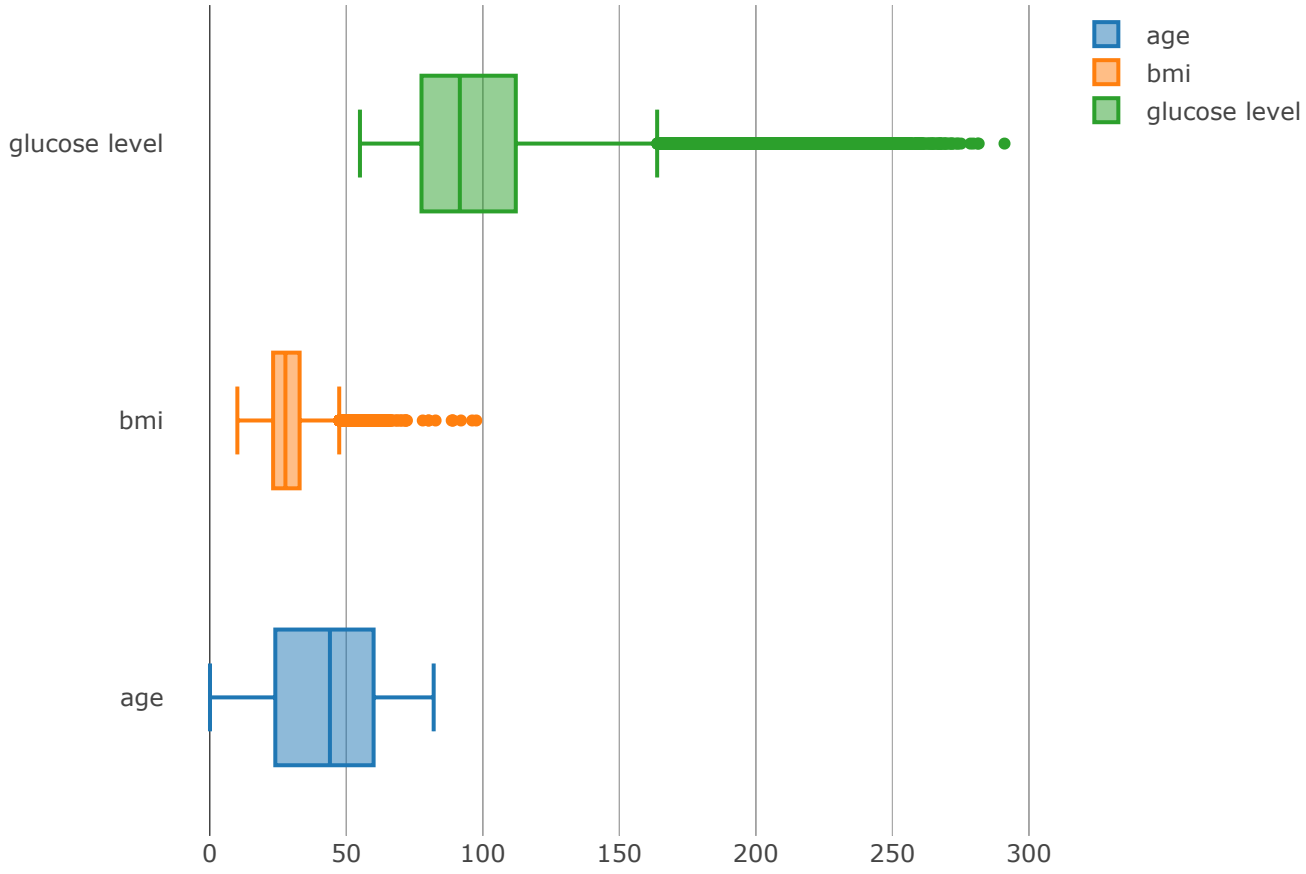
```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
p <- plot_ly(data, x = data$age, type="box", name = 'age')


q <-add_trace(p, x = data$bmi, type="box", name = 'bmi')


w <-add_trace(q , x = data$avg_glucose_level,type = "box" , name ="glucose level" )
w
```

```
## Warning: Ignoring 1462 observations
```




The pmf plot and cdf plots for age are as follows:

```
#age
values <- data$age
tab <- table(values)

dframe <- as.data.frame(tab)
#dframe

x <- as.numeric(as.character(dframe$values))

# probability distribution is
f <- dframe$Freq / (sum(dframe$Freq))

# calculate  the  mean
mu  <- sum(x * f)
mu
```

```
## [1] 42.21789
```

```
# variance of the  distribution is
sigmaSquare <-  sum((x -  mu)^2 * f)
sigmaSquare
```

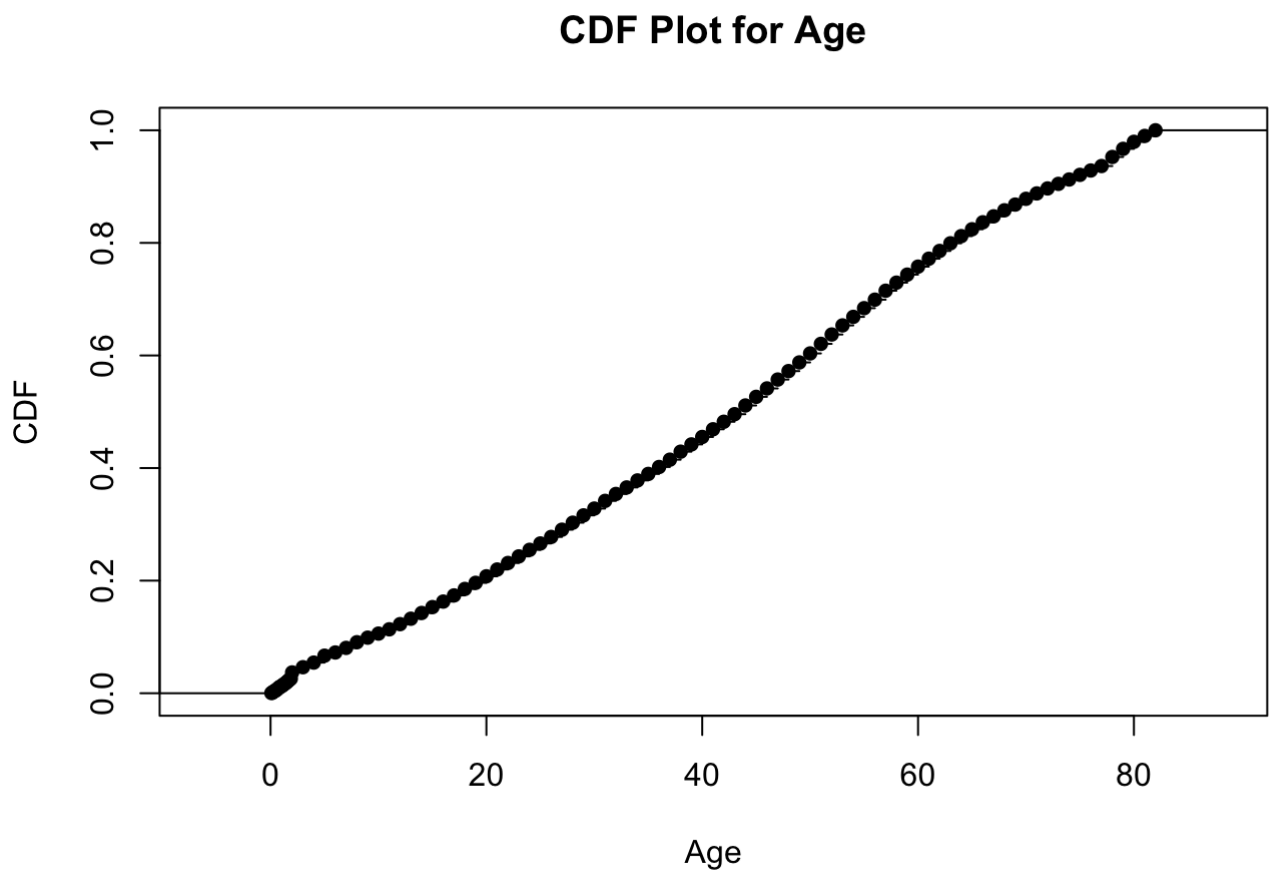
```
## [1] 507.1229
```

```
sigma <- sqrt(sigmaSquare)
sigma
```

```
## [1] 22.51939
```

```
plot(x, f, type = 'h',  xlab = "Age of patients", ylab = "PMF", ylim = c(0, 0.04), main
 = "Spike plot for Age")
abline(h = 0 )
```


```
cdf <- c(0, cumsum(f))
cdfplot <- stepfun(x, cdf)
plot(cdfplot,  verticals=FALSE, pch=16, main="CDF Plot for Age", xlab = "Age", ylab = "C
DF")
```




Similar analysis of distribution for bmi is as follows:

```
#bmi

values <- data$bmi
tab <- table(values)

dframe <- as.data.frame(tab)
#dframe

x <- as.numeric(as.character(dframe$values))

# probability distribution is
f <- dframe$Freq / (sum(dframe$Freq))

# calculate  the  mean
mu  <- sum(x * f)
mu
```

```
## [1] 28.60504
```

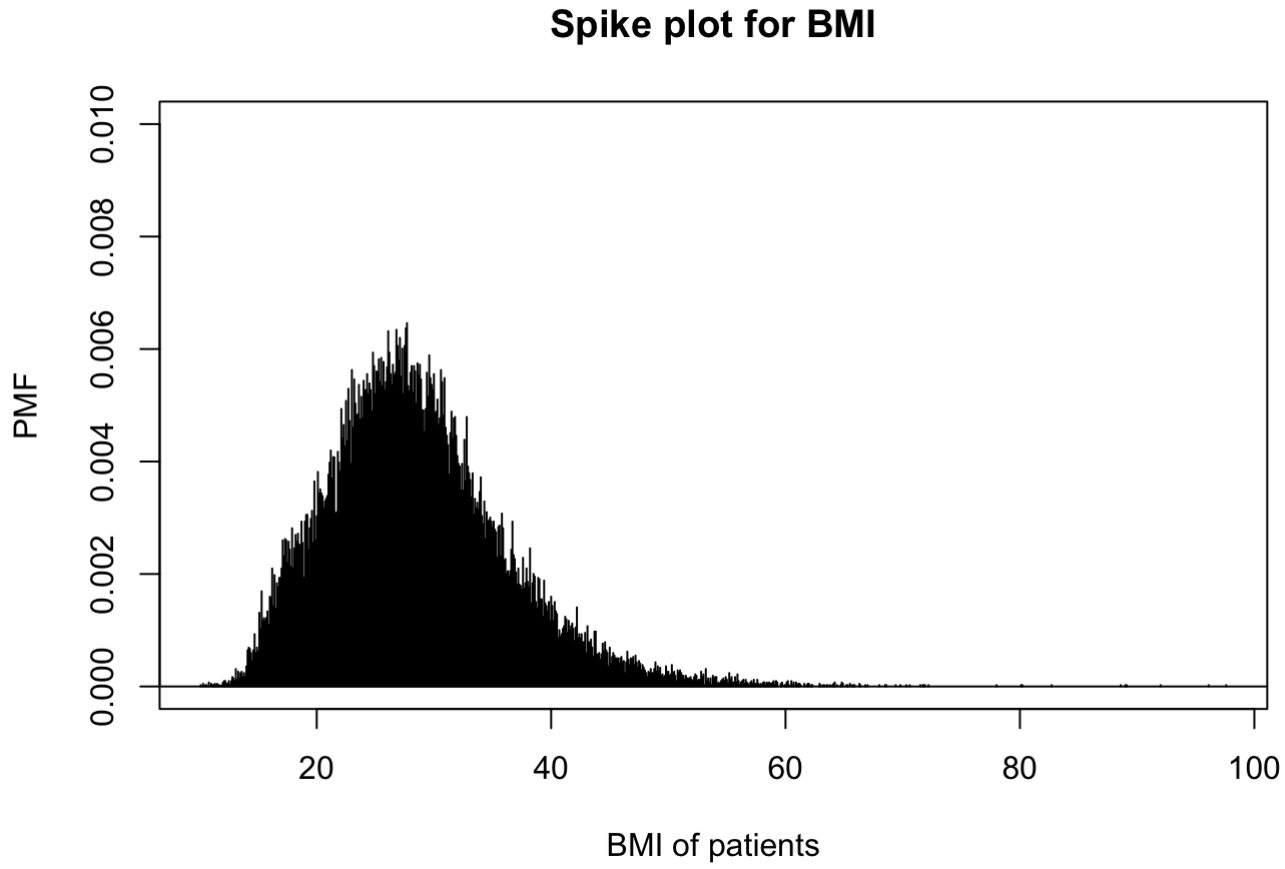
```
# variance of the  distribution is
sigmaSquare <-  sum((x -  mu)^2 * f)
sigmaSquare
```

```
## [1] 60.37178
```

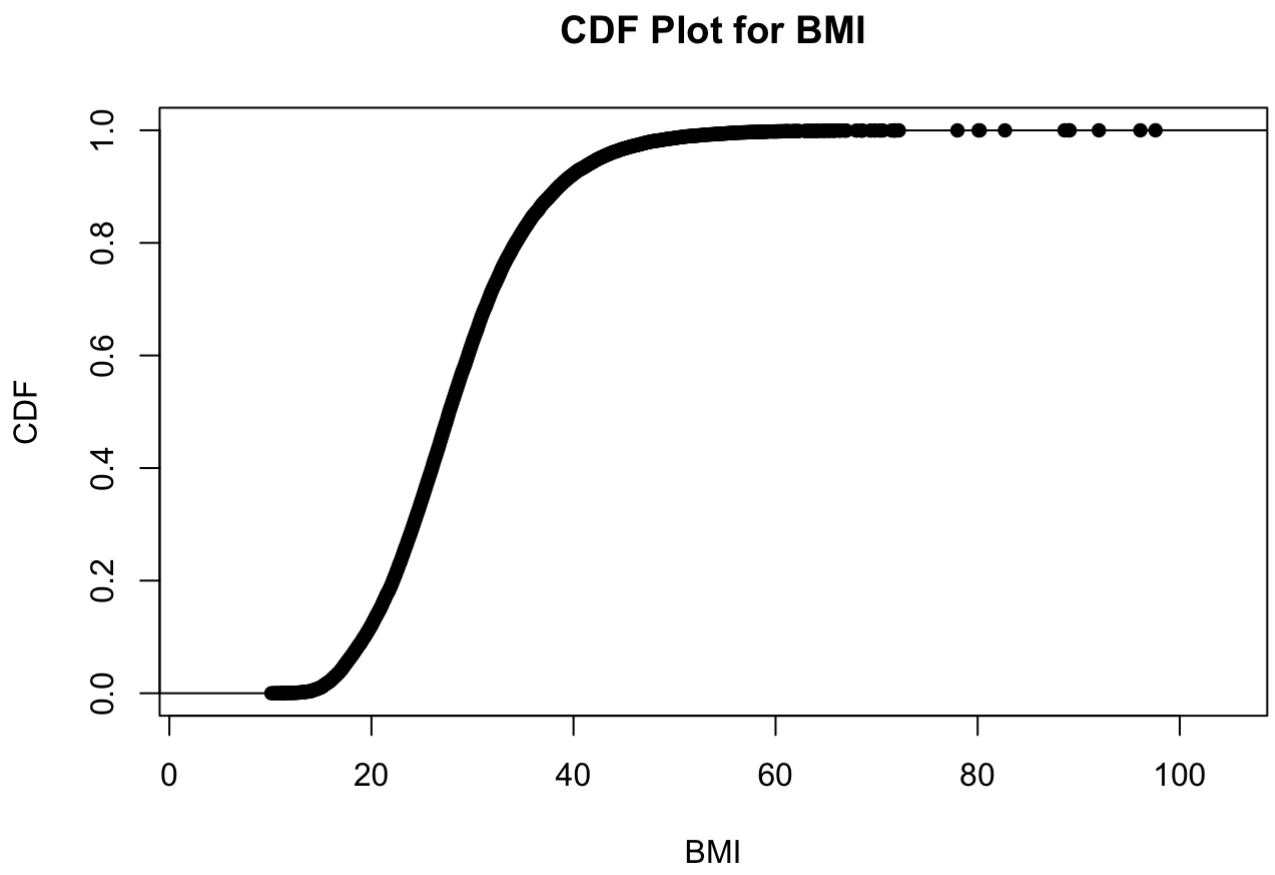
```
sigma <- sqrt(sigmaSquare)
sigma
```

```
## [1] 7.769928
```

```
plot(x, f, type = 'h',  xlab = "BMI of patients", ylab = "PMF",ylim = c(0, 0.01), main =
"Spike plot for BMI")
abline(h = 0 )
```




```
cdf <- c(0, cumsum(f))
cdfplot <- stepfun(x, cdf)
plot(cdfplot,  verticals=FALSE, pch=16, main="CDF Plot for BMI", xlab = "BMI", ylab = "C
DF")
```




We have applied Cental Limit Theorem on age attibute as follows:

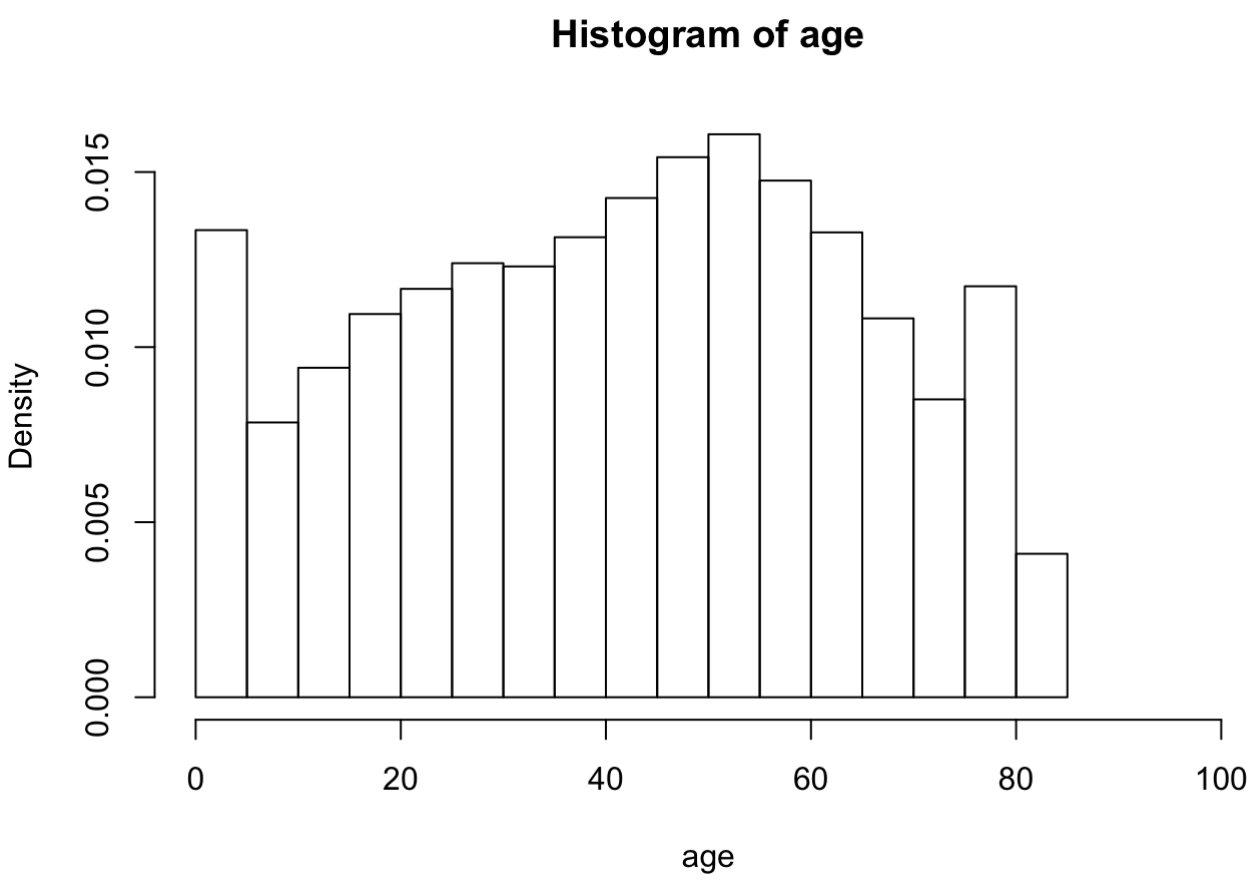
```
age <-data$age

ctable <- table(age)
#ctable
mu <- mean(age)
mu
```

```
## [1] 42.21789
```

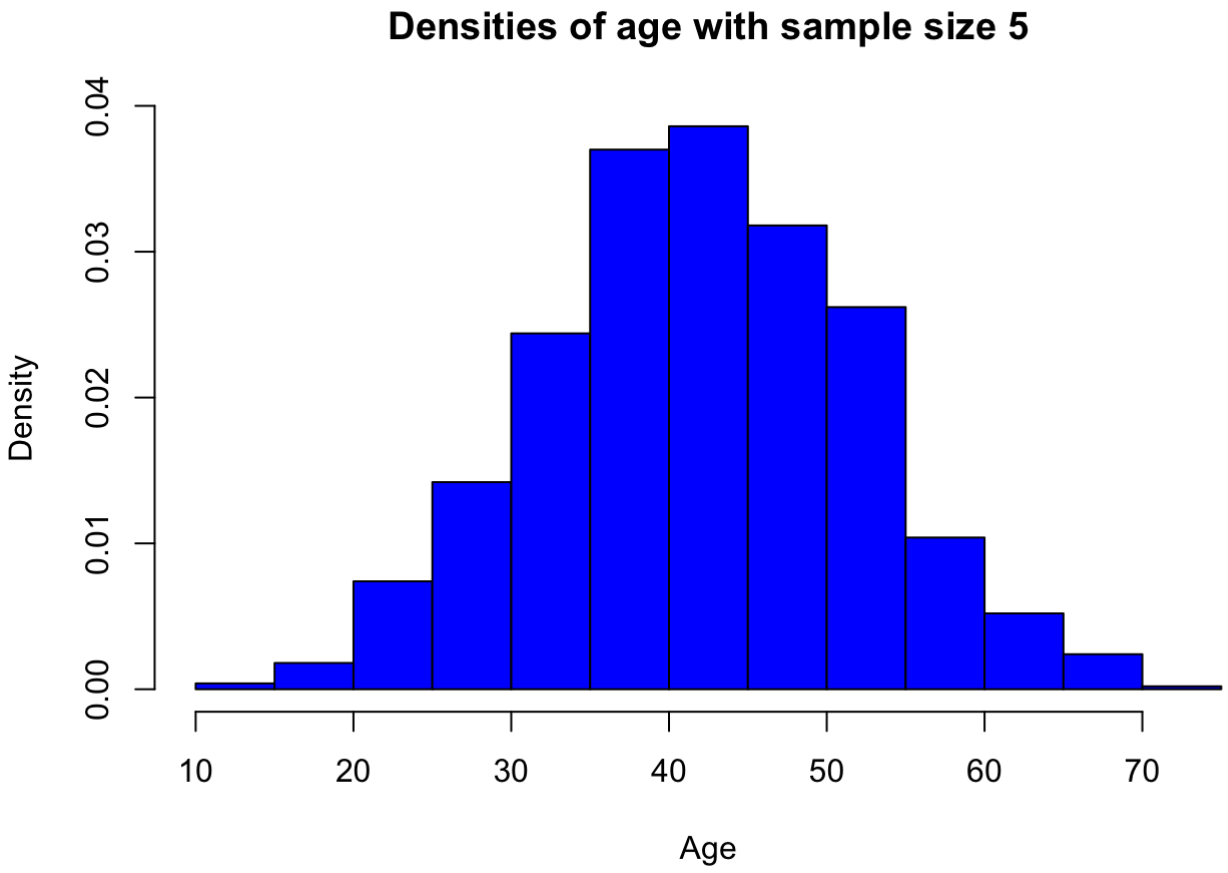
```
sigma <- sd(age)
sigma
```

```
## [1] 22.51965
```

```
dframe <- as.data.frame(ctable)
#dframe

x <- as.numeric(as.character(data$age))
#x
hist(x, probability = TRUE, xlim = c(0, 100), xlab = "age", ylab = "Density", main = "Hi
stogram of age")
```




```
# sample size 5
samples <- 1000
sample_size <- 5
xbar <- numeric(samples)
for(i in 1:samples){
  xbar[i] = mean(sample(x, size = sample_size, replace = T))
}
hist(xbar, prob = T, xlab = "Age",
     main = "Densities of age with sample size 5", col = "blue")
```



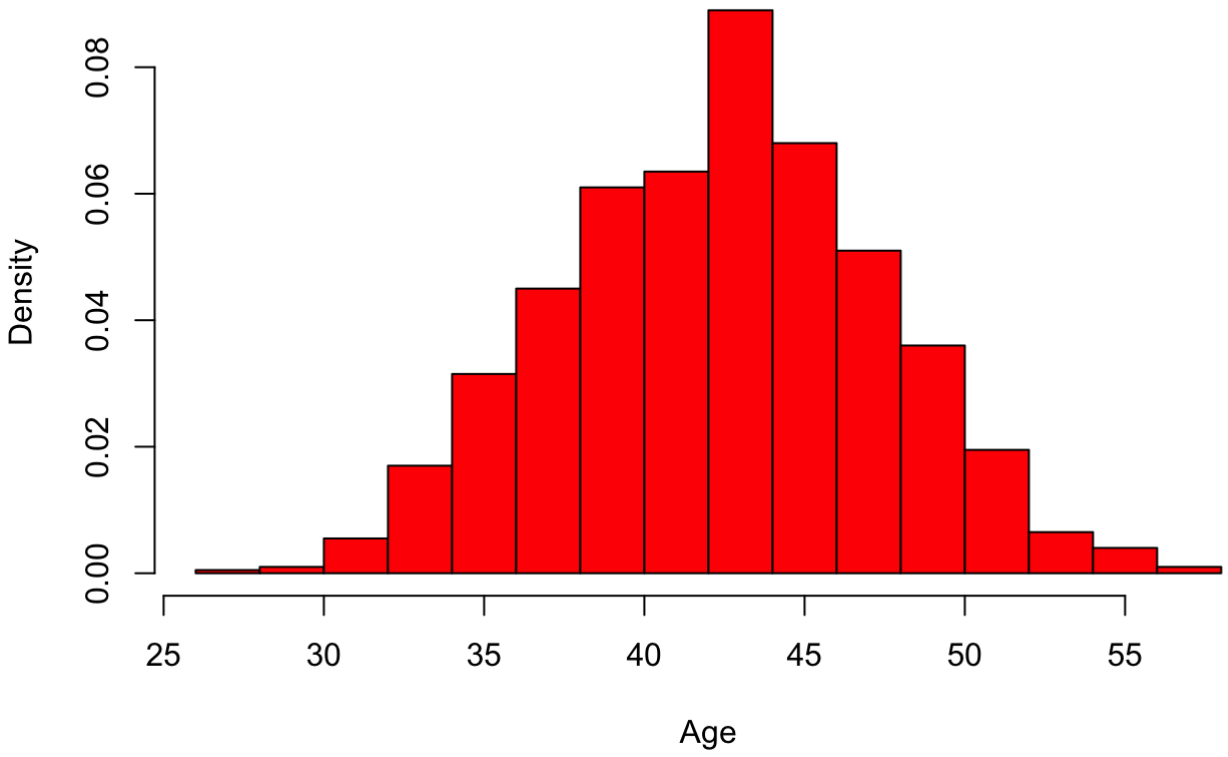

```
mean1 <- mean(xbar)
sd1 <- sd(xbar)
mean1
```

```
## [1] 42.10943
```

```
sd1
```

```
## [1] 10.02085
```

```
#sample size 20
samples <- 1000
sample_size <- 20
xbar <- numeric(samples)
for(i in 1:samples){
  xbar[i] = mean(sample(x, size = sample_size, replace = T))
}
hist(xbar, prob = T, xlab = "Age",
     main = "Densities of age with sample size 20", col = "red")
```



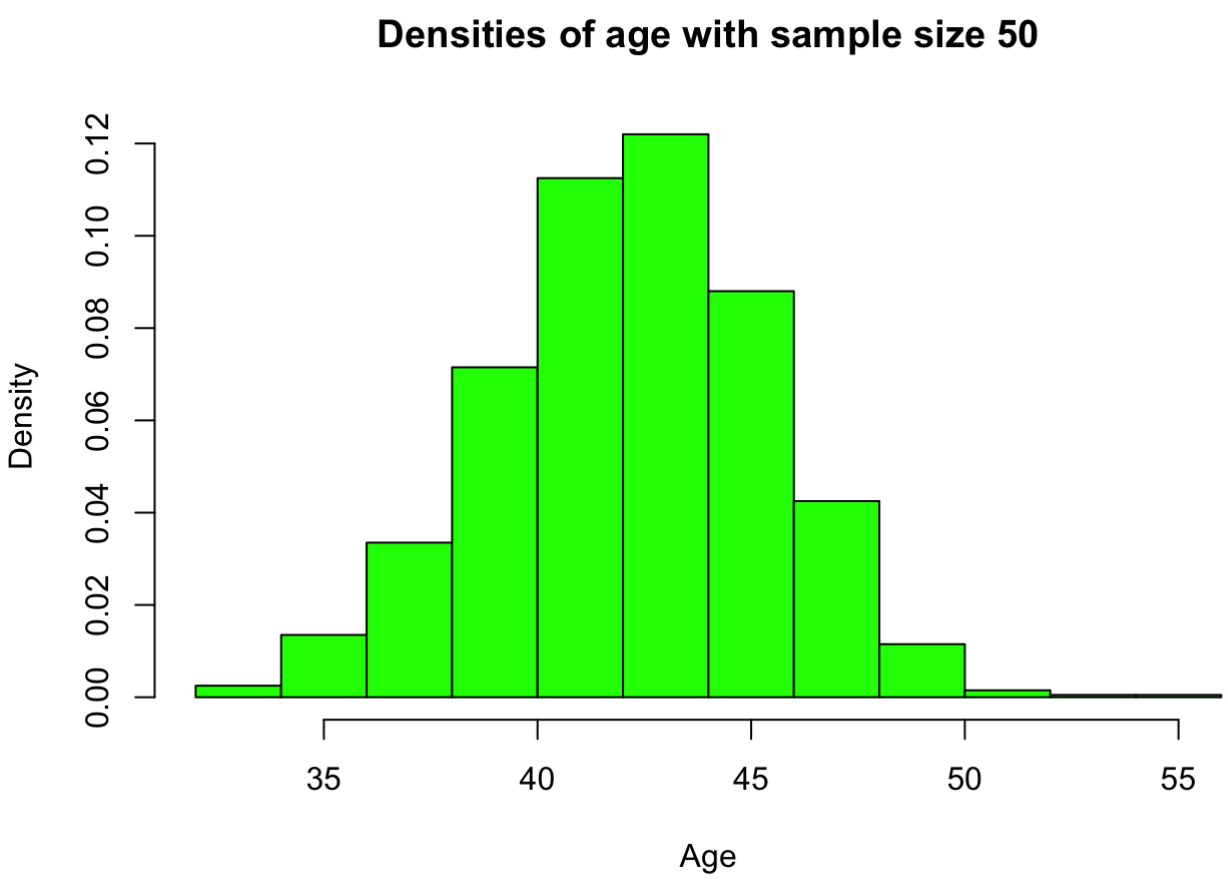

```
mean2 <- mean(xbar)
sd2 <- sd(xbar)
mean2
```

```
## [1] 42.40183
```

```
sd2
```

```
## [1] 4.982658
```

```
#sample size 50
samples <- 1000
sample_size <- 50
xbar <- numeric(samples)
for(i in 1:samples){
  xbar[i] = mean(sample(x, size = sample_size, replace = T))
}
hist(xbar, prob = T, xlab = "Age",
     main = "Densities of age with sample size 50", col = "green")
```




```
mean3 <- mean(xbar)
sd3 <- sd(xbar)
mean3
```

```
## [1] 42.19804
```

```
sd3
```

```
## [1] 3.223502
```

```
cat("1st distribution:\nMean =",mean1,"\nSD =",sd1)
```

```
## 1st distribution:
## Mean = 42.10943
## SD = 10.02085
```

```
cat("2nd distribution:\nMean =",mean2,"\nSD =",sd2)
```

```
## 2nd distribution:
## Mean = 42.40183
## SD = 4.982658
```

```
cat("3rd distribution:\nMean =",mean3,"\nSD =",sd3)
```

```
## 3rd distribution:
## Mean = 42.19804
## SD = 3.223502
```

Similarly, We have applied Cental Limit Theorem on average glucose level attibute as follows:

```
glucose <-data$avg_glucose_level

ctable <- table(glucose)
#ctable
mu <- mean(glucose)
mu
```

```
## [1] 104.4827
```

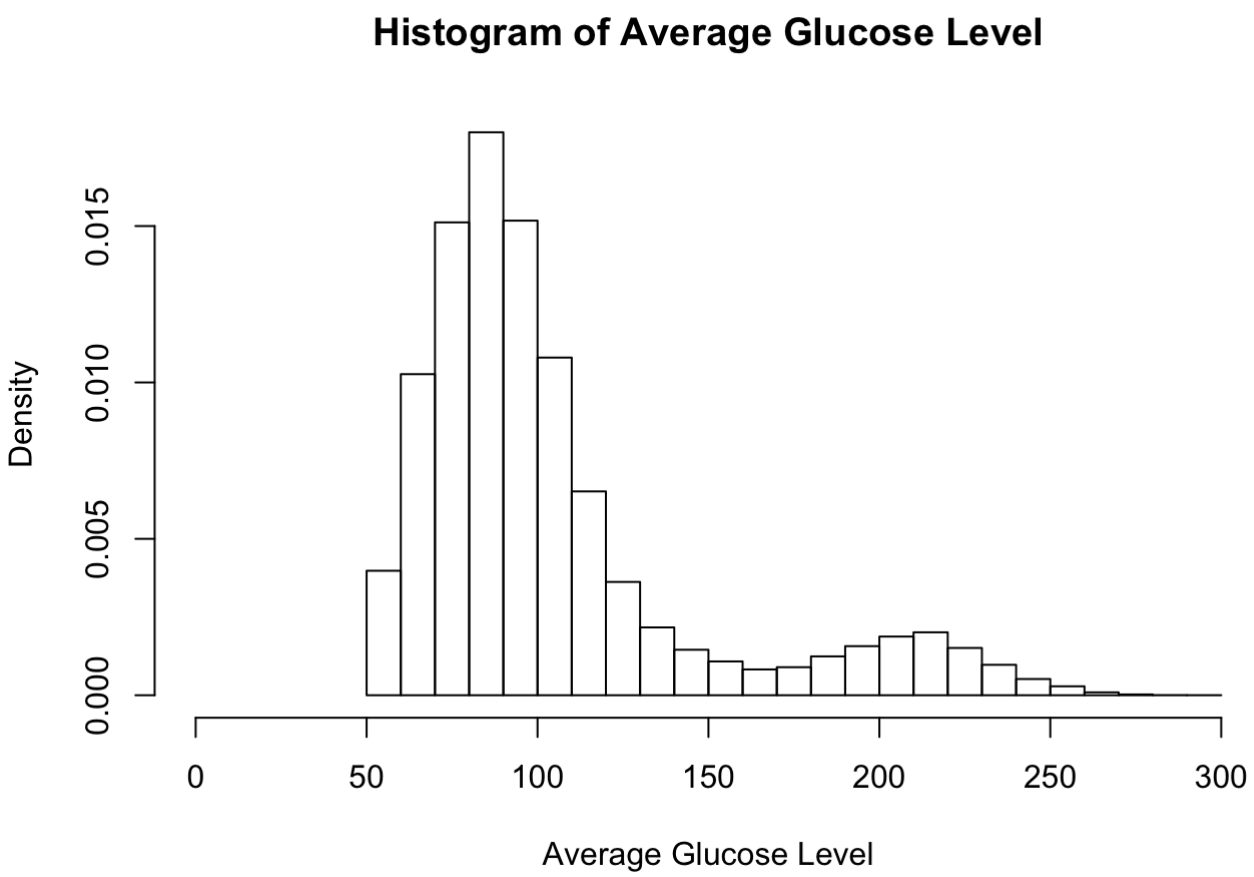
```
sigma <- sd(glucose)
sigma
```

```
## [1] 43.11175
```

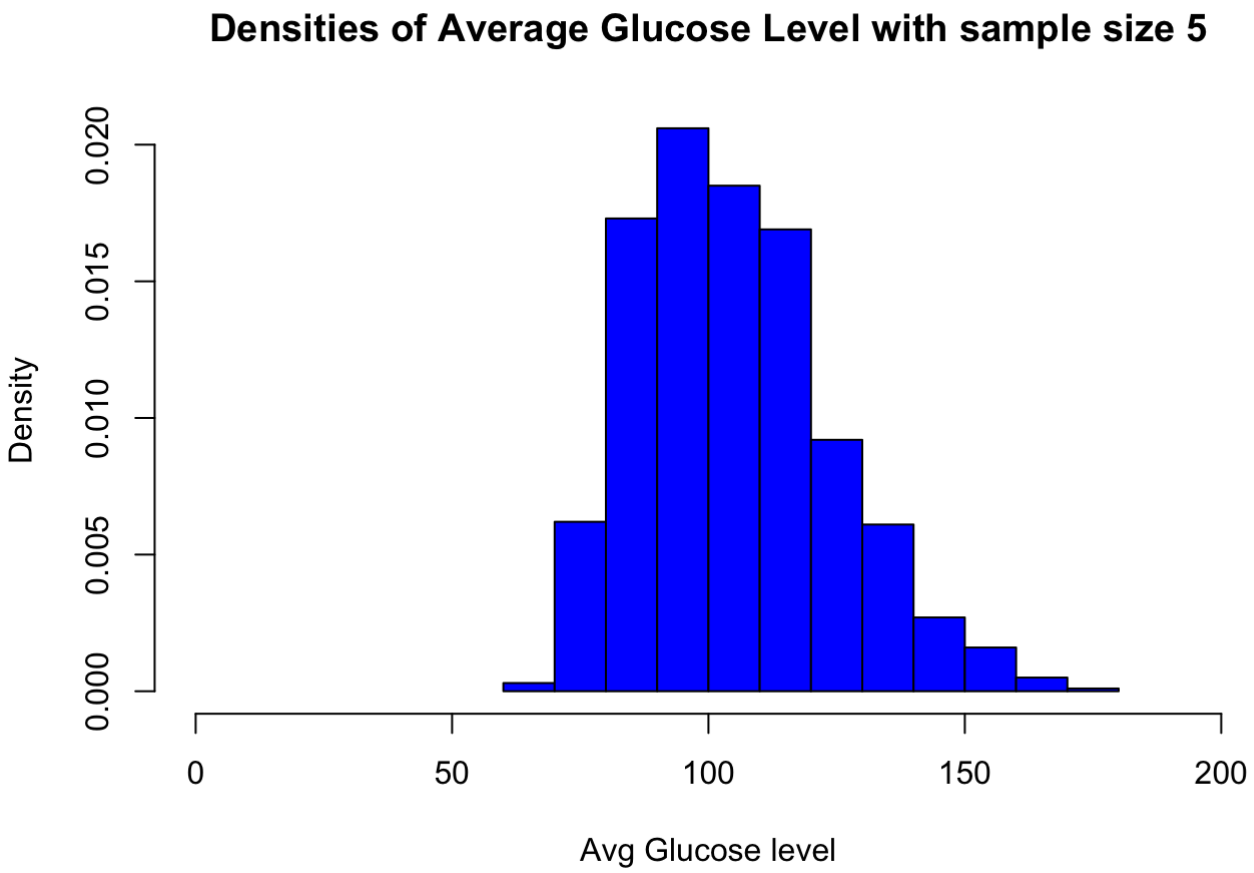
```
dframe <- as.data.frame(ctable)
#dframe

x <- as.numeric(as.character(data$avg_glucose_level))
#x
max(data$avg_glucose_level)
```

```
## [1] 291.05
```

```
hist(x, probability = TRUE, xlim = c(0,300 ), xlab = "Average Glucose Level", ylab = "De
nsity", main = "Histogram of Average Glucose Level")
```




```
#sample size 5
samples <- 1000
sample_size <- 5
xbar <- numeric(samples)
for(i in 1:samples){
  xbar[i] = mean(sample(x, size = sample_size, replace = T))
}
hist(xbar, prob = T,xlim = c(0,200), xlab = "Avg Glucose level",
     main = "Densities of Average Glucose Level with sample size 5", col = "blue")
```



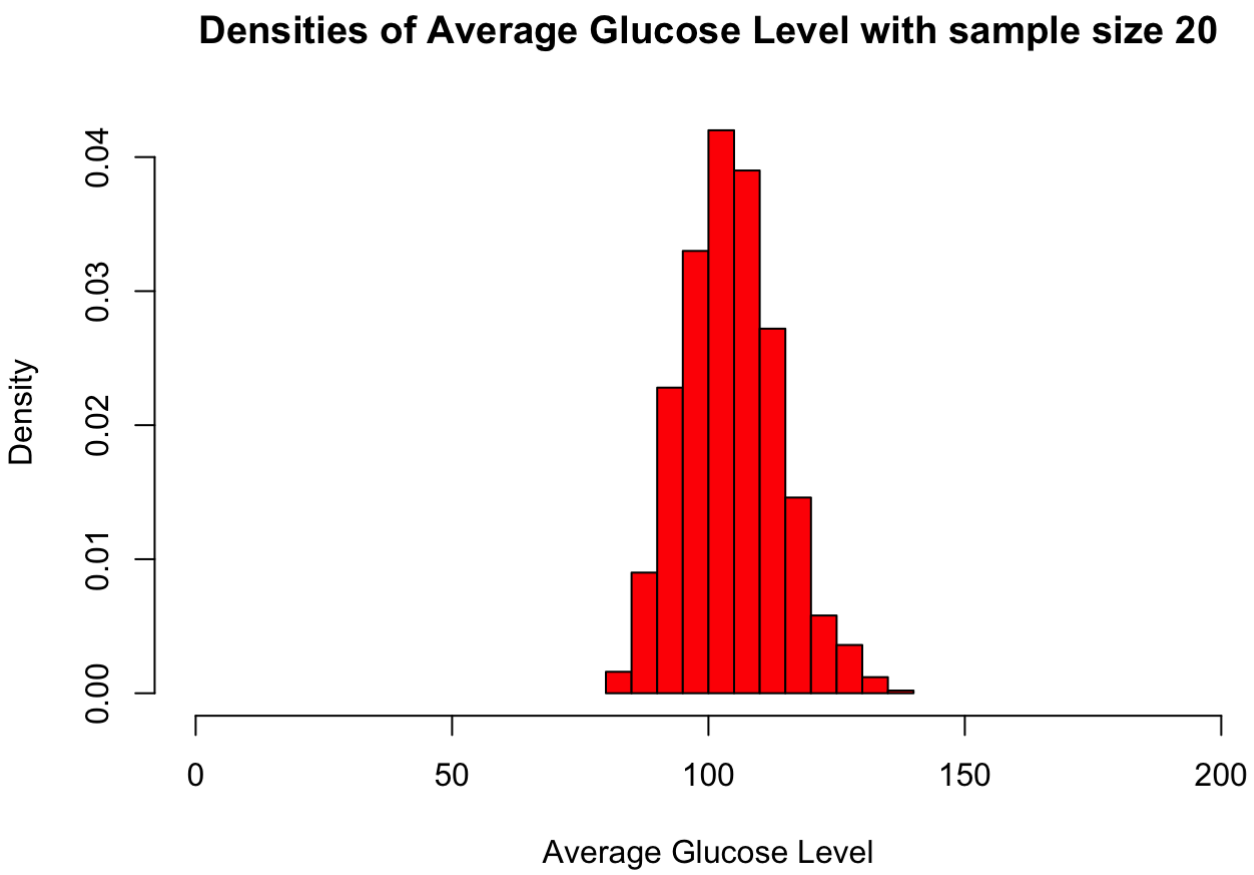

```
mean1 <- mean(xbar)
sd1 <- sd(xbar)
mean1
```

```
## [1] 105.1466
```

```
sd1
```

```
## [1] 19.01525
```

```
#sample size 20
samples <- 1000
sample_size <- 20
xbar <- numeric(samples)
for(i in 1:samples){
  xbar[i] = mean(sample(x, size = sample_size, replace = T))
}
hist(xbar, prob = T,xlim = c(0,200), xlab = "Average Glucose Level",
     main = "Densities of Average Glucose Level with sample size 20", col = "red")
```



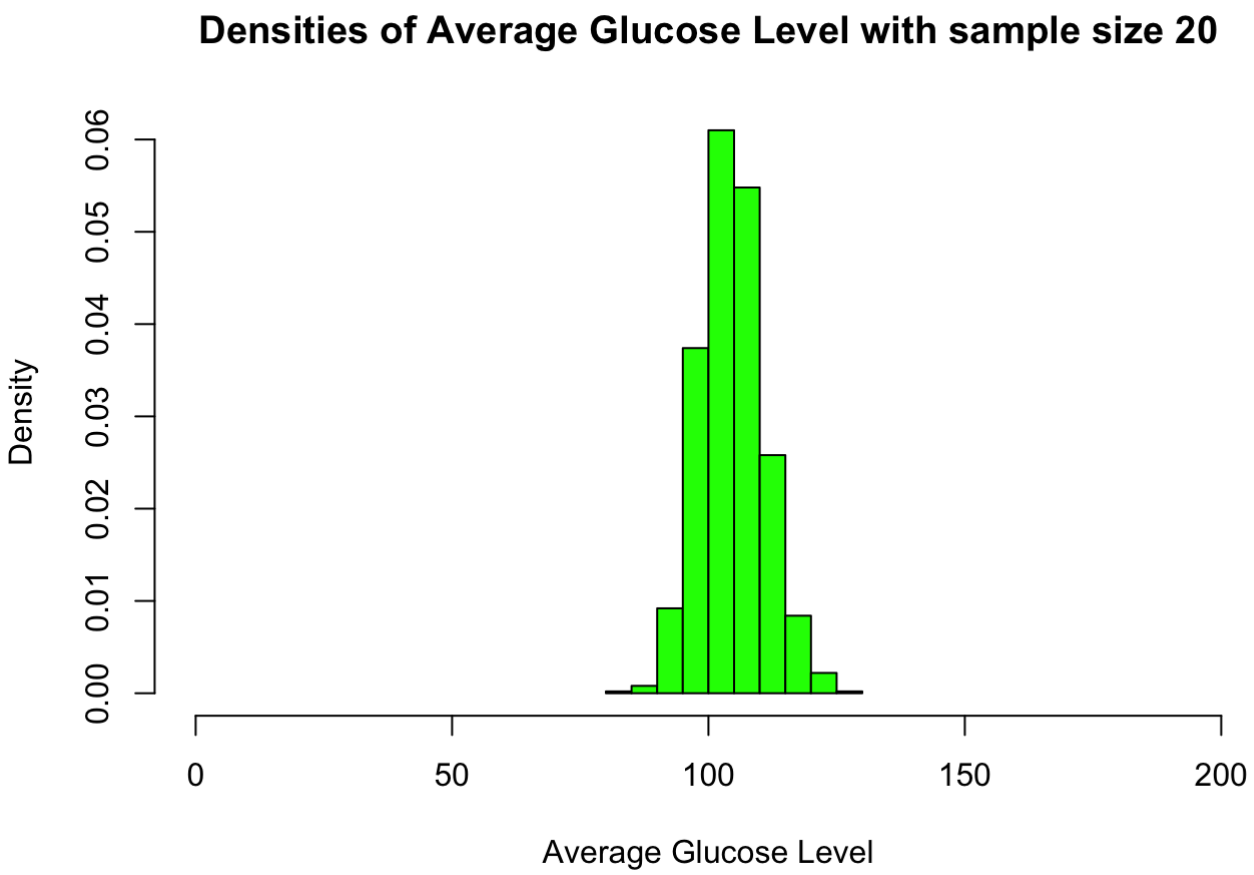

```
mean2 <- mean(xbar)
sd2 <- sd(xbar)
mean2
```

```
## [1] 104.3902
```

```
sd2
```

```
## [1] 9.338029
```

```
#sample size 50
samples <- 1000
sample_size <- 50
xbar <- numeric(samples)
for(i in 1:samples){
  xbar[i] = mean(sample(x, size = sample_size, replace = T))
}
hist(xbar, prob = T,xlim = c(0,200), xlab = "Average Glucose Level",
     main = "Densities of Average Glucose Level with sample size 20", col = "green")
```




```
mean3 <- mean(xbar)
sd3 <- sd(xbar)
mean3
```

```
## [1] 104.4944
```

```
sd3
```

```
## [1] 6.242851
```

```
cat("1st distribution:\nMean =",mean1,"\nSD =",sd1)
```

```
## 1st distribution:
## Mean = 105.1466
## SD = 19.01525
```

```
cat("2nd distribution:\nMean =",mean2,"\nSD =",sd2)
```

```
## 2nd distribution:
## Mean = 104.3902
## SD = 9.338029
```

```
cat("3rd distribution:\nMean =",mean3,"\nSD =",sd3)
```

```
## 3rd distribution:
## Mean = 104.4944
## SD = 6.242851
```

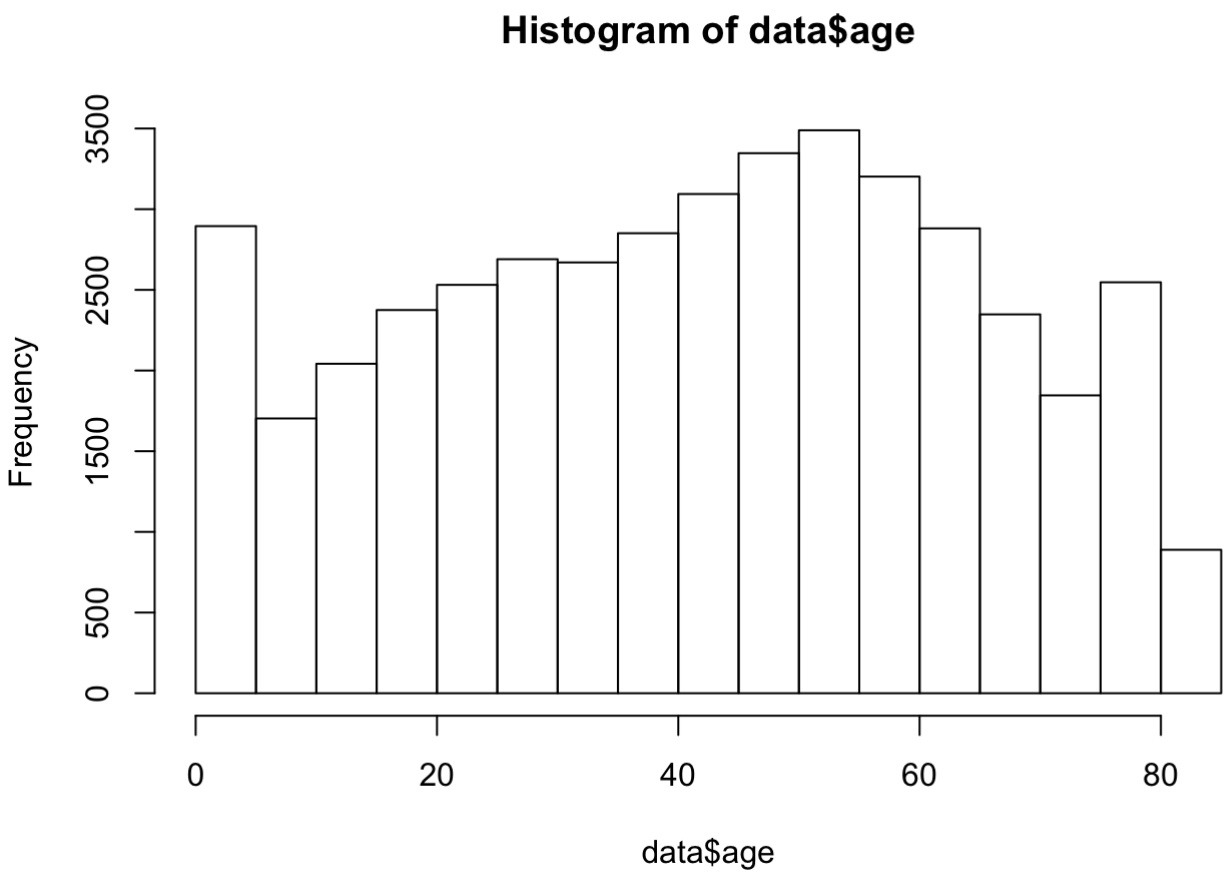
In both cases, our mean remains almost the same, but as sample size increases, standard deviation decreases, proving the applicability of Central Limit Theorem on both of our attributes.

We further performed sampling on age attributes of our data:

```
#sampling
library(sampling)
table(data$age)
```

```
##
## 0.08 0.16 0.24 0.32  0.4 0.48 0.56 0.64 0.72  0.8 0.88    1 1.08 1.16 1.24
##   17   26   50   53   35   37   47   58   66   61   46   34   62   48   44
## 1.32  1.4 1.48 1.56 1.64 1.72  1.8 1.88    2    3    4    5    6    7    8
##   52   55   45   63   60   57   62   47  479  402  356  533  246  355  436
##    9   10   11   12   13   14   15   16   17   18   19   20   21   22   23
##  353  313  330  398  419  443  452  426  475  498  472  504  523  503  508
##   24   25   26   27   28   29   30   31   32   33   34   35   36   37   38
##  506  491  503  558  540  564  525  592  533  504  540  501  535  564  617
##   39   40   41   42   43   44   45   46   47   48   49   50   51   52   53
##  561  574  592  580  585  671  666  649  684  652  668  694  738  721  701
##   54   55   56   57   58   59   60   61   62   63   64   65   66   67   68
##  658  671  655  690  623  618  616  612  598  589  553  529  537  459  465
##   69   70   71   72   73   74   75   76   77   78   79   80   81   82
##  441  446  415  384  352  338  357  336  344  698  626  543  454  435
```

```
hist(data$age)
```




```
mean_without_sampling <- mean(data$age)
sd(data$age)
```

```
## [1] 22.51965
```

Following are various sampling methods for sample size 500:

```
#sample size = 500
# srswor
sample.size <- 500
s <- srswor(sample.size,nrow(data))
sample.1 <- data[s != 0, ]

mean_srswor <- mean(sample.1$age)
#srswr
set.seed(153)
s <- srswr(sample.size, nrow(data))
sample.2 <- data[s != 0, ]
mean_srswr <- mean(sample.2$age)
```

```
#Systematic Sampling
N <- nrow(data)
n <- 1000

k <- ceiling(N / n)
k
```

```
## [1] 44
```

```
r <- sample(k, 1)
r
```

```
## [1] 22
```

```
s <- seq(r, by = k, length = n)
#s

sample.3 <- data[s, ]
table(sample.3$age)
```

```
##
## 0.08 0.24 0.32  0.4 0.72  0.8 0.88    1 1.08 1.32  1.4 1.56 1.64 1.72  1.8
##    1    3    2    1    1    1    1    1    2    3    1    1    1    1    3
##    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
##    9    9    9   16    7    8   11    3    4    6   10    7    8    6    5
##   17   18   19   20   21   22   23   24   25   26   27   28   29   30   31
##   10   10   11   10   14   17    9   11    9   15    9   11   12   13   15
##   32   33   34   35   36   37   38   39   40   41   42   43   44   45   46
##   12    8   10    9   12   14   15   15   16   17   14   13   19   15    8
##   47   48   49   50   51   52   53   54   55   56   57   58   59   60   61
##   19   12   19   14   18   17   16   14   13   12   18   15   14   13   16
##   62   63   64   65   66   67   68   69   70   71   72   73   74   75   76
##   17   16    8    8   14   10    7   11   13    9    5    8    9    9   10
##   77   78   79   80   81   82
##   10   13   19   24   12   10
```

```
mean_systematic <- mean(sample.3$age)
```

```
#Systematic Sampling with unequal probabilities
pik <- inclusionprobabilities(data$age, sample.size)
s <- UPsystematic(pik)
sample.4 <- data[s != 0, ]
table(sample.4$age)
```

```
##
##  6  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
##  1  3  1  1  1  1  2  1  4  2  2  2  4  3  4  3  5  4  6  2  1  7  3  8  4
## 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56
##  4  5  8  4  5 10  4  2  9  3  4  6  5 10 12  5  4 11  7 11 11  7 12 12  4
## 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81
## 12 11 15  5  8 14  6  9 12  5  8 11  5  6 10  9  8  6  6  6  6 18 15 16 10
## 82
## 13
```

```
data["age_range"] = NA

data$age_range <- cut(data$age, breaks = c(0, 25, 50, 75, Inf), labels = c('A', 'B', 'C'
, 'D'))

data_age <- data.frame(
  age = data$age,
  age_range = data$age_range
)

freq <- table(data_age$age_range)
freq
```

```
##
##     A     B     C     D
## 11546 14652 13766  3436
```

```
set.seed(123)
head(data_age)
```

```
##   age age_range
## 1   3         A
## 2  58         C
## 3   8         A
## 4  70         C
## 5  14         A
## 6  47         B
```

```
st.sizes <- sample.size * freq / sum(freq)

st.1 <- strata(data_age, stratanames = c("age_range"),
               size = st.sizes, method = c("srswor"),
               description = TRUE)
```

```
## Stratum 1
##
## Population total and number of selected units: 11546 133.0184
## Stratum 2
##
## Population total and number of selected units: 13766 168.8018
## Stratum 3
##
## Population total and number of selected units: 14652 158.5945
## Stratum 4
##
## Population total and number of selected units: 3436 39.58525
## Number of strata  4
## Total number of selected units 500
```

```
#st.1

st.sample1 <- getdata(data, st.1)

#st.sample1
```

```
# cluster sampling
cl <- cluster(data, c("age"), size = 4, method = "srswor")
sample.6 <- getdata(data, cl)

table(sample.6$age)
```

```
##
##  15  16  50  60
## 452 426 694 616
```

```
mean_cluster <- mean(sample.6$age)
```

For sample size 500, the means through all sampling methods' samples are as follows:

```
#sample size 500
mean_without_sampling
```

```
## [1] 42.21789
```

```
mean_srswor
```

```
## [1] 42.27424
```

```
mean_srswr
```

```
## [1] 42.98635
```

```
mean_systematic
```

```
## [1] NA
```

```
mean_cluster
```

```
## [1] 38.96527
```

Following are various sampling methods for sample size 1000:

```
#sample size = 1000
# srswor
sample.size <- 1000
s <- srswor(sample.size,nrow(data))
sample.1 <- data[s != 0, ]

mean_srswor <- mean(sample.1$age)
#srswr
set.seed(153)
s <- srswr(sample.size, nrow(data))
sample.2 <- data[s != 0, ]
mean_srswr <- mean(sample.2$age)
```

```
#Systematic Sampling
N <- nrow(data)
n <- 1000

k <- ceiling(N / n)
k
```

```
## [1] 44
```

```
r <- sample(k, 1)
r
```

```
## [1] 19
```

```
s <- seq(r, by = k, length = n)
s
```

```
##    [1]    19    63   107   151   195   239   283   327   371   415   459
##   [12]   503   547   591   635   679   723   767   811   855   899   943
##   [23]   987  1031  1075  1119  1163  1207  1251  1295  1339  1383  1427
##   [34]  1471  1515  1559  1603  1647  1691  1735  1779  1823  1867  1911
##   [45]  1955  1999  2043  2087  2131  2175  2219  2263  2307  2351  2395
##   [56]  2439  2483  2527  2571  2615  2659  2703  2747  2791  2835  2879
##   [67]  2923  2967  3011  3055  3099  3143  3187  3231  3275  3319  3363
##   [78]  3407  3451  3495  3539  3583  3627  3671  3715  3759  3803  3847
##   [89]  3891  3935  3979  4023  4067  4111  4155  4199  4243  4287  4331
##  [100]  4375  4419  4463  4507  4551  4595  4639  4683  4727  4771  4815
##  [111]  4859  4903  4947  4991  5035  5079  5123  5167  5211  5255  5299
##  [122]  5343  5387  5431  5475  5519  5563  5607  5651  5695  5739  5783
##  [133]  5827  5871  5915  5959  6003  6047  6091  6135  6179  6223  6267
##  [144]  6311  6355  6399  6443  6487  6531  6575  6619  6663  6707  6751
##  [155]  6795  6839  6883  6927  6971  7015  7059  7103  7147  7191  7235
##  [166]  7279  7323  7367  7411  7455  7499  7543  7587  7631  7675  7719
##  [177]  7763  7807  7851  7895  7939  7983  8027  8071  8115  8159  8203
##  [188]  8247  8291  8335  8379  8423  8467  8511  8555  8599  8643  8687
##  [199]  8731  8775  8819  8863  8907  8951  8995  9039  9083  9127  9171
##  [210]  9215  9259  9303  9347  9391  9435  9479  9523  9567  9611  9655
##  [221]  9699  9743  9787  9831  9875  9919  9963 10007 10051 10095 10139
##  [232] 10183 10227 10271 10315 10359 10403 10447 10491 10535 10579 10623
##  [243] 10667 10711 10755 10799 10843 10887 10931 10975 11019 11063 11107
##  [254] 11151 11195 11239 11283 11327 11371 11415 11459 11503 11547 11591
##  [265] 11635 11679 11723 11767 11811 11855 11899 11943 11987 12031 12075
##  [276] 12119 12163 12207 12251 12295 12339 12383 12427 12471 12515 12559
##  [287] 12603 12647 12691 12735 12779 12823 12867 12911 12955 12999 13043
##  [298] 13087 13131 13175 13219 13263 13307 13351 13395 13439 13483 13527
##  [309] 13571 13615 13659 13703 13747 13791 13835 13879 13923 13967 14011
##  [320] 14055 14099 14143 14187 14231 14275 14319 14363 14407 14451 14495
##  [331] 14539 14583 14627 14671 14715 14759 14803 14847 14891 14935 14979
##  [342] 15023 15067 15111 15155 15199 15243 15287 15331 15375 15419 15463
##  [353] 15507 15551 15595 15639 15683 15727 15771 15815 15859 15903 15947
##  [364] 15991 16035 16079 16123 16167 16211 16255 16299 16343 16387 16431
##  [375] 16475 16519 16563 16607 16651 16695 16739 16783 16827 16871 16915
##  [386] 16959 17003 17047 17091 17135 17179 17223 17267 17311 17355 17399
##  [397] 17443 17487 17531 17575 17619 17663 17707 17751 17795 17839 17883
##  [408] 17927 17971 18015 18059 18103 18147 18191 18235 18279 18323 18367
##  [419] 18411 18455 18499 18543 18587 18631 18675 18719 18763 18807 18851
##  [430] 18895 18939 18983 19027 19071 19115 19159 19203 19247 19291 19335
##  [441] 19379 19423 19467 19511 19555 19599 19643 19687 19731 19775 19819
##  [452] 19863 19907 19951 19995 20039 20083 20127 20171 20215 20259 20303
##  [463] 20347 20391 20435 20479 20523 20567 20611 20655 20699 20743 20787
##  [474] 20831 20875 20919 20963 21007 21051 21095 21139 21183 21227 21271
##  [485] 21315 21359 21403 21447 21491 21535 21579 21623 21667 21711 21755
##  [496] 21799 21843 21887 21931 21975 22019 22063 22107 22151 22195 22239
##  [507] 22283 22327 22371 22415 22459 22503 22547 22591 22635 22679 22723
##  [518] 22767 22811 22855 22899 22943 22987 23031 23075 23119 23163 23207
##  [529] 23251 23295 23339 23383 23427 23471 23515 23559 23603 23647 23691
##  [540] 23735 23779 23823 23867 23911 23955 23999 24043 24087 24131 24175
##  [551] 24219 24263 24307 24351 24395 24439 24483 24527 24571 24615 24659
##  [562] 24703 24747 24791 24835 24879 24923 24967 25011 25055 25099 25143
##  [573] 25187 25231 25275 25319 25363 25407 25451 25495 25539 25583 25627
```

```
##  [584] 25671 25715 25759 25803 25847 25891 25935 25979 26023 26067 26111
##  [595] 26155 26199 26243 26287 26331 26375 26419 26463 26507 26551 26595
##  [606] 26639 26683 26727 26771 26815 26859 26903 26947 26991 27035 27079
##  [617] 27123 27167 27211 27255 27299 27343 27387 27431 27475 27519 27563
##  [628] 27607 27651 27695 27739 27783 27827 27871 27915 27959 28003 28047
##  [639] 28091 28135 28179 28223 28267 28311 28355 28399 28443 28487 28531
##  [650] 28575 28619 28663 28707 28751 28795 28839 28883 28927 28971 29015
##  [661] 29059 29103 29147 29191 29235 29279 29323 29367 29411 29455 29499
##  [672] 29543 29587 29631 29675 29719 29763 29807 29851 29895 29939 29983
##  [683] 30027 30071 30115 30159 30203 30247 30291 30335 30379 30423 30467
##  [694] 30511 30555 30599 30643 30687 30731 30775 30819 30863 30907 30951
##  [705] 30995 31039 31083 31127 31171 31215 31259 31303 31347 31391 31435
##  [716] 31479 31523 31567 31611 31655 31699 31743 31787 31831 31875 31919
##  [727] 31963 32007 32051 32095 32139 32183 32227 32271 32315 32359 32403
##  [738] 32447 32491 32535 32579 32623 32667 32711 32755 32799 32843 32887
##  [749] 32931 32975 33019 33063 33107 33151 33195 33239 33283 33327 33371
##  [760] 33415 33459 33503 33547 33591 33635 33679 33723 33767 33811 33855
##  [771] 33899 33943 33987 34031 34075 34119 34163 34207 34251 34295 34339
##  [782] 34383 34427 34471 34515 34559 34603 34647 34691 34735 34779 34823
##  [793] 34867 34911 34955 34999 35043 35087 35131 35175 35219 35263 35307
##  [804] 35351 35395 35439 35483 35527 35571 35615 35659 35703 35747 35791
##  [815] 35835 35879 35923 35967 36011 36055 36099 36143 36187 36231 36275
##  [826] 36319 36363 36407 36451 36495 36539 36583 36627 36671 36715 36759
##  [837] 36803 36847 36891 36935 36979 37023 37067 37111 37155 37199 37243
##  [848] 37287 37331 37375 37419 37463 37507 37551 37595 37639 37683 37727
##  [859] 37771 37815 37859 37903 37947 37991 38035 38079 38123 38167 38211
##  [870] 38255 38299 38343 38387 38431 38475 38519 38563 38607 38651 38695
##  [881] 38739 38783 38827 38871 38915 38959 39003 39047 39091 39135 39179
##  [892] 39223 39267 39311 39355 39399 39443 39487 39531 39575 39619 39663
##  [903] 39707 39751 39795 39839 39883 39927 39971 40015 40059 40103 40147
##  [914] 40191 40235 40279 40323 40367 40411 40455 40499 40543 40587 40631
##  [925] 40675 40719 40763 40807 40851 40895 40939 40983 41027 41071 41115
##  [936] 41159 41203 41247 41291 41335 41379 41423 41467 41511 41555 41599
##  [947] 41643 41687 41731 41775 41819 41863 41907 41951 41995 42039 42083
##  [958] 42127 42171 42215 42259 42303 42347 42391 42435 42479 42523 42567
##  [969] 42611 42655 42699 42743 42787 42831 42875 42919 42963 43007 43051
##  [980] 43095 43139 43183 43227 43271 43315 43359 43403 43447 43491 43535
##  [991] 43579 43623 43667 43711 43755 43799 43843 43887 43931 43975
```

```
sample.3 <- data[s, ]
table(sample.3$age)
```

```
##
## 0.08 0.32  0.4 0.48 0.56 0.64  0.8 0.88    1 1.24 1.32  1.4 1.48 1.64 1.72
##    1    1    1    2    1    1    1    1    1    1    1    3    1    1    2
##  1.8 1.88    2    3    4    5    6    7    8    9   10   11   12   13   14
##    2    2   17    9    8    7    6   13   11   10   11    7   11    5   12
##   15   16   17   18   19   20   21   22   23   24   25   26   27   28   29
##    7    5    9   16   12   14   14   13   13   12   10   15   14   10   18
##   30   31   32   33   34   35   36   37   38   39   40   41   42   43   44
##   20   13   13    9   13   15    9   10   17    8    9   19   14   14   16
##   45   46   47   48   49   50   51   52   53   54   55   56   57   58   59
##   14   14    8   16    6   14    9   18   18   19   21   12   14   16   16
##   60   61   62   63   64   65   66   67   68   69   70   71   72   73   74
##   23    9   10   15    7   18    9    6   18    9   12    7    7    9    6
##   75   76   77   78   79   80   81   82
##    9    6    9   15   12    7    8    9
```

```
mean_systematic <- mean(sample.3$age)
```

```
#Systematic Sampling with unequal probabilities
pik <- inclusionprobabilities(data$age, sample.size)
s <- UPsystematic(pik)
sample.4 <- data[s != 0, ]
table(sample.4$age)
```

```
##
##  4  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
##  2  3  1  2  3  1  6  3  1  4  4  3  5  4  3  4  8  5  2  3  5  8  9  4  4
## 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54
## 11 10 10  8  8 17 15 15 12 15 16  9 14  9 22 15 15 19 18 19 15 28 22 25 18
## 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
## 20 20 20 21 24 18 18 17 27 15 18 17  8 14 11 22 23 16 18 15 14 12 17 27 35
## 80 81 82
## 27 11 13
```

```
# Stratified sampling
freq <- table(data_age$age_range)
freq
```

```
##
##     A     B     C     D
## 11546 14652 13766  3436
```

```
set.seed(123)
head(data_age)
```

```
##   age age_range
## 1   3         A
## 2  58         C
## 3   8         A
## 4  70         C
## 5  14         A
## 6  47         B
```

```
st.sizes <- sample.size * freq / sum(freq)

st.1 <- strata(data_age, stratanames = c("age_range"),
               size = st.sizes, method = c("srswor"),
               description = TRUE)
```

```
## Stratum 1
##
## Population total and number of selected units: 11546 266.0369
## Stratum 2
##
## Population total and number of selected units: 13766 337.6037
## Stratum 3
##
## Population total and number of selected units: 14652 317.1889
## Stratum 4
##
## Population total and number of selected units: 3436 79.17051
## Number of strata  4
## Total number of selected units 1000
```

```
#st.1

st.sample1 <- getdata(data, st.1)

#st.sample1
```

```
# cluster sampling
cl <- cluster(data, c("age"), size = 4, method = "srswor")
sample.6 <- getdata(data, cl)

table(sample.6$age)
```

```
##
##   1 1.4   7  39
##  34  55 355 561
```

```
mean_cluster <- mean(sample.6$age)
```

Following are the mean values for all of our sampling methods:

```
#sample size 1000
mean_without_sampling
```

```
## [1] 42.21789
```

```
mean_srswor
```

```
## [1] 42.91236
```

```
mean_srswr
```

```
## [1] 42.90006
```

```
mean_systematic
```

```
## [1] NA
```

```
mean_cluster
```

```
## [1] 24.35323
```

Following is an R code for finding if a patient with an input ID has a stroke or no. (Please refer to the .R file for code to dynamically insert any id as input)

```
id <- 35327
id_stroke <- data[data$id == id, ]
if(id_stroke$stroke == 0){
  print(paste("The patient with ID", id, "does not have a stroke"))
} else{
  print(paste("The patient with ID", id, "has a stroke"))
}
```

```
## [1] "The patient with ID 35327 does not have a stroke"
```

Following is the word cloud for work type:

```
# Word cloud for work type
# Install
#install.packages("tm")  # for text mining
#install.packages("wordcloud") # word-cloud generator
#install.packages("RColorBrewer") # color palettes
# Load
library("tm")
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

```
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
library("SnowballC")
library("wordcloud")
```

```
## Loading required package: RColorBrewer
```

```
library("RColorBrewer")
yes_stroke <- data[data$stroke == 1, ]

table(yes_stroke$work_type)
```

```
##
##      children      Govt_job  Never_worked       Private Self-employed
##             2            89             0           441           251
```

```
workCorpus <- Corpus(VectorSource(yes_stroke$work_type))
workCorpus <- tm_map(workCorpus, content_transformer(tolower))
```

```
## Warning in tm_map.SimpleCorpus(workCorpus, content_transformer(tolower)):
## transformation drops documents
```

```
wordcloud(workCorpus, max.words = 100, random.order = FALSE)
```


govt_job
private
self-employed