Exploratory Data Analysis

Rakibul Islam Prince

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

```
library(pacman)
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

```
p_load(dplyr, GGally, ggplot2, ggthemes, ggvis, httr, lubridate, plotly, rio, rmarkdown, shiny, stringr
```

```
stroke_data<-import("./../data/healthcare-dataset-stroke-data.csv")
```

```
str(stroke_data)
```

```
## 'data.frame':    5110 obs. of  12 variables:
##  $ id               : int  9046 51676 31112 60182 1665 56669 53882 10434 27419 60491 ...
##  $ gender           : chr  "Male" "Female" "Male" "Female" ...
##  $ age              : num  67 61 80 49 79 81 74 69 59 78 ...
##  $ hypertension     : int  0 0 0 0 1 0 1 0 0 0 ...
##  $ heart_disease    : int  1 0 1 0 0 0 1 0 0 0 ...
##  $ ever_married     : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ work_type        : chr  "Private" "Self-employed" "Private" "Private" ...
##  $ Residence_type   : chr  "Urban" "Rural" "Rural" "Urban" ...
##  $ avg_glucose_level: num  229 202 106 171 174 ...
##  $ bmi              : chr  "36.6" "N/A" "32.5" "34.4" ...
##  $ smoking_status   : chr  "formerly smoked" "never smoked" "never smoked" "smokes" ...
##  $ stroke           : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
install.packages("binom", repos = "https://cloud.r-project.org/")
```

```
## Installing package into '/geode2/home/u060/mdprin/Carbonate/R/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
library(binom)
```

```
head(stroke_data,15)
```

```
##       id gender age hypertension heart_disease ever_married     work_type
## 1   9046   Male  67            0             1          Yes       Private
## 2  51676 Female  61            0             0          Yes Self-employed
```

```
## 3  31112   Male  80            0             1          Yes       Private
## 4  60182 Female  49            0             0          Yes       Private
## 5   1665 Female  79            1             0          Yes Self-employed
## 6  56669   Male  81            0             0          Yes       Private
## 7  53882   Male  74            1             1          Yes       Private
## 8  10434 Female  69            0             0           No       Private
## 9  27419 Female  59            0             0          Yes       Private
## 10 60491 Female  78            0             0          Yes       Private
## 11 12109 Female  81            1             0          Yes       Private
## 12 12095 Female  61            0             1          Yes      Govt_job
## 13 12175 Female  54            0             0          Yes       Private
## 14  8213   Male  78            0             1          Yes       Private
## 15  5317 Female  79            0             1          Yes       Private
##    Residence_type avg_glucose_level  bmi  smoking_status stroke
## 1           Urban            228.69 36.6 formerly smoked      1
## 2           Rural            202.21  N/A    never smoked      1
## 3           Rural            105.92 32.5    never smoked      1
## 4           Urban            171.23 34.4          smokes      1
## 5           Rural            174.12   24    never smoked      1
## 6           Urban            186.21   29 formerly smoked      1
## 7           Rural             70.09 27.4    never smoked      1
## 8           Urban             94.39 22.8    never smoked      1
## 9           Rural             76.15  N/A         Unknown      1
## 10          Urban             58.57 24.2         Unknown      1
## 11          Rural             80.43 29.7    never smoked      1
## 12          Rural            120.46 36.8          smokes      1
## 13          Urban            104.51 27.3          smokes      1
## 14          Urban            219.84  N/A         Unknown      1
## 15          Urban            214.09 28.2    never smoked      1
```

data cleaning

```r
#removing all the rows having NA values

stroke_data<-na.omit(stroke_data)
```

```r
#removing all rows having "N/A" values
clean_stroke_data<-stroke_data[!apply(stroke_data=="N/A",1,any),]
```

```r
head(clean_stroke_data,n=15)
```

```
##       id gender age hypertension heart_disease ever_married     work_type
## 1   9046   Male  67            0             1          Yes       Private
## 3  31112   Male  80            0             1          Yes       Private
## 4  60182 Female  49            0             0          Yes       Private
## 5   1665 Female  79            1             0          Yes Self-employed
## 6  56669   Male  81            0             0          Yes       Private
## 7  53882   Male  74            1             1          Yes       Private
## 8  10434 Female  69            0             0           No       Private
## 10 60491 Female  78            0             0          Yes       Private
## 11 12109 Female  81            1             0          Yes       Private
## 12 12095 Female  61            0             1          Yes      Govt_job
## 13 12175 Female  54            0             0          Yes       Private
## 15  5317 Female  79            0             1          Yes       Private
## 16 58202 Female  50            1             0          Yes Self-employed
## 17 56112   Male  64            0             1          Yes       Private
```

```
## 18 34120   Male  75            1             0          Yes       Private
##    Residence_type avg_glucose_level  bmi  smoking_status stroke
## 1           Urban            228.69 36.6 formerly smoked      1
## 3           Rural            105.92 32.5    never smoked      1
## 4           Urban            171.23 34.4          smokes      1
## 5           Rural            174.12   24    never smoked      1
## 6           Urban            186.21   29 formerly smoked      1
## 7           Rural             70.09 27.4    never smoked      1
## 8           Urban             94.39 22.8    never smoked      1
## 10          Urban             58.57 24.2         Unknown      1
## 11          Rural             80.43 29.7    never smoked      1
## 12          Rural            120.46 36.8          smokes      1
## 13          Urban            104.51 27.3          smokes      1
## 15          Urban            214.09 28.2    never smoked      1
## 16          Rural            167.41 30.9    never smoked      1
## 17          Urban            191.61 37.5          smokes      1
## 18          Urban            221.29 25.8          smokes      1
```

```
#keeping only male and females
clean_g_stroke_data<-clean_stroke_data %>%
  filter(gender!="Other")
```

```
#converting bmi column to numeric
clean_g_stroke_data$bmi<-as.numeric(clean_g_stroke_data$bmi)
str(clean_g_stroke_data)
```

```
## 'data.frame':    4908 obs. of  12 variables:
##  $ id               : int  9046 31112 60182 1665 56669 53882 10434 60491 12109 12095 ...
##  $ gender           : chr  "Male" "Male" "Female" "Female" ...
##  $ age              : num  67 80 49 79 81 74 69 78 81 61 ...
##  $ hypertension     : int  0 0 0 1 0 1 0 0 1 0 ...
##  $ heart_disease    : int  1 1 0 0 0 1 0 0 0 1 ...
##  $ ever_married     : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ work_type        : chr  "Private" "Private" "Private" "Self-employed" ...
##  $ Residence_type   : chr  "Urban" "Rural" "Urban" "Rural" ...
##  $ avg_glucose_level: num  229 106 171 174 186 ...
##  $ bmi              : num  36.6 32.5 34.4 24 29 27.4 22.8 24.2 29.7 36.8 ...
##  $ smoking_status   : chr  "formerly smoked" "never smoked" "smokes" "never smoked" ...
##  $ stroke           : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
bmi_gen<-clean_g_stroke_data %>% select(gender, bmi)
str(bmi_gen)
```

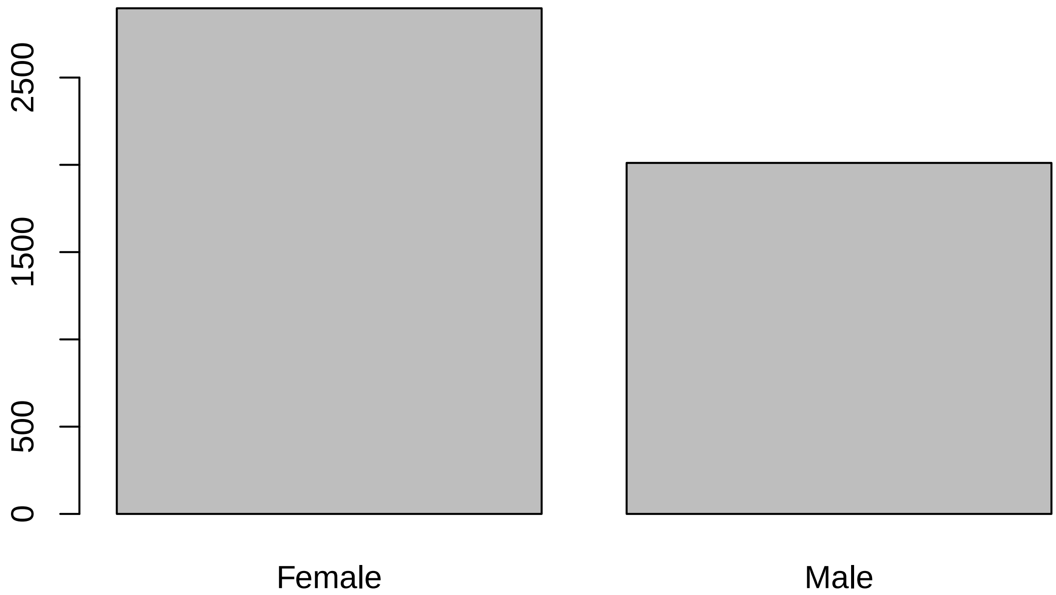
```
## 'data.frame':    4908 obs. of  2 variables:
##  $ gender: chr  "Male" "Male" "Female" "Female" ...
##  $ bmi   : num  36.6 32.5 34.4 24 29 27.4 22.8 24.2 29.7 36.8 ...
```

```
bmi_gen$gender<-as.factor(bmi_gen$gender)
```

```
str(bmi_gen)
```

```
## 'data.frame':    4908 obs. of  2 variables:
##  $ gender: Factor w/ 2 levels "Female","Male": 2 2 1 1 2 2 1 1 1 1 ...
##  $ bmi   : num  36.6 32.5 34.4 24 29 27.4 22.8 24.2 29.7 36.8 ...
```

```
plot(bmi_gen$gender)
```




```r
bmi_mar<- clean_g_stroke_data %>% select(ever_married, bmi)
```

```r
bmi_mar$ever_married<-as.factor(bmi_mar$ever_married)
bmi_mar$bmi<-as.numeric(bmi_mar$bmi)
```

```r
#calculating the mean and the median

# I have to ensure that the ever_married column is of factor or character type.

mar_mean<-mean(bmi_mar$bmi[bmi_mar$ever_married=="Yes"], na.rm=TRUE)
mar_median<-median(bmi_mar$bmi[bmi_mar$ever_married=="Yes"], na.rm=TRUE)
unmar_mean<-mean(bmi_mar$bmi[bmi_mar$ever_married!="Yes"], na.rm=TRUE)
unmar_median<-median(bmi_mar$bmi[bmi_mar$ever_married!="Yes"], na.rm=TRUE)



p <- ggplot(bmi_mar, aes(x=bmi, fill=ever_married)) +
  geom_histogram(alpha=0.5, position="identity", bins=40) +
  scale_fill_manual(values=c("red", "green4"))+
  geom_vline(aes(xintercept=mar_mean), color="green4", linetype="dashed", size=1, show.legend=TRUE, labe
  geom_vline(aes(xintercept=mar_median), color="green4", linetype="solid", size=1, show.legend=TRUE, la
  geom_vline(aes(xintercept=unmar_mean), color="red", linetype="dashed", size=1, show.legend=TRUE, label
  geom_vline(aes(xintercept=unmar_median), color="red", linetype="solid", size=1, show.legend=TRUE, labe
  theme_minimal() +
  labs(title = "Histograms by Marital Status with Mean and Median", x = "BMI Value", y = "Count")
```

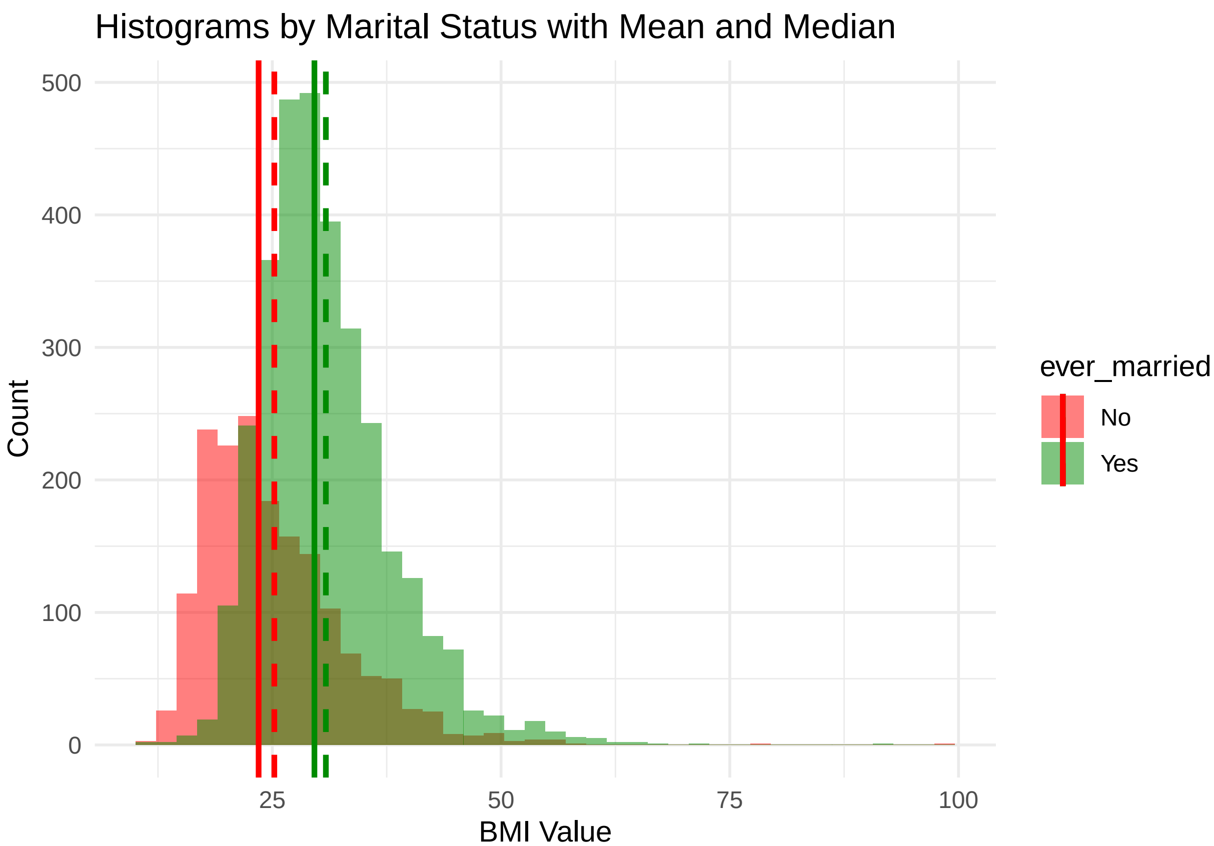
```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning in geom_vline(aes(xintercept = mar_mean), color = "green4", linetype =
## "dashed", : Ignoring unknown parameters: `label`
```

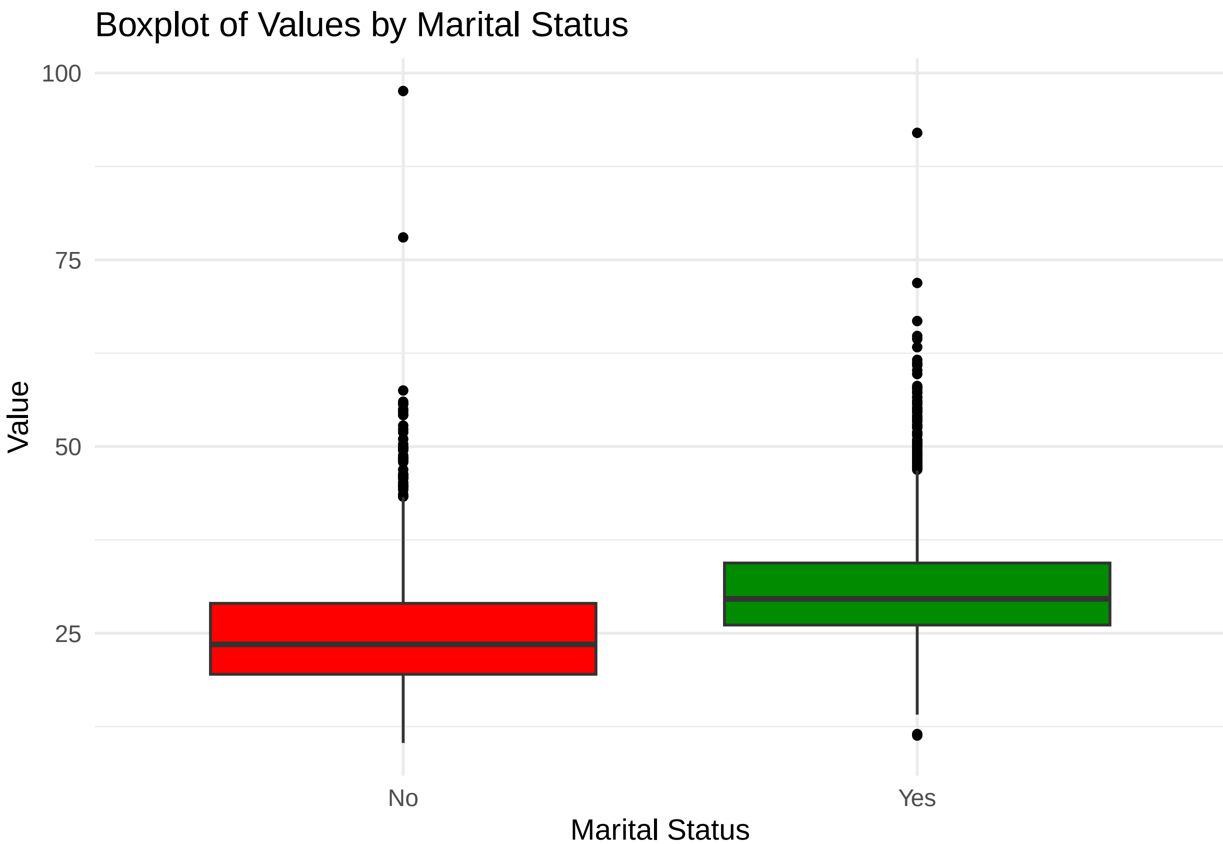
```
## Warning in geom_vline(aes(xintercept = mar_median), color = "green4", linetype
## = "solid", : Ignoring unknown parameters: `label`
```

```
## Warning in geom_vline(aes(xintercept = unmar_mean), color = "red", linetype =
## "dashed", : Ignoring unknown parameters: `label`

## Warning in geom_vline(aes(xintercept = unmar_median), color = "red", linetype =
## "solid", : Ignoring unknown parameters: `label`
```

```r
print(p)
```




```r
q <- ggplot(bmi_mar, aes(x=ever_married, y=bmi)) +
  geom_boxplot(aes(fill=ever_married), outlier.color="black", outlier.shape=16) +
  labs(title="Boxplot of Values by Marital Status", x="Marital Status", y="Value") +
  theme_minimal() +
  scale_fill_manual(values=c("Yes"="green4", "No"="red")) +  # Manual coloring
  theme(legend.position="none")  # Removing legend since fill color indicates category

print(q)
```




```
summary(filter(bmi_mar, bmi_mar$ever_married=="Yes"))
```

```
##  ever_married      bmi
##  No  :   0      Min.   :11.30
##  Yes:3204       1st Qu.:26.10
##                 Median :29.60
##                 Mean   :30.85
##                 3rd Qu.:34.40
##                 Max.   :92.00
```

```
summary(filter(bmi_mar, bmi_mar$ever_married=="No"))
```

```
##  ever_married      bmi
##  No :1704       Min.   :10.30
##  Yes:   0       1st Qu.:19.50
##                 Median :23.50
##                 Mean   :25.22
##                 3rd Qu.:29.00
##                 Max.   :97.60
```

Here, we can see that Mean and Median BMI of married people are 30.85 and 29.60 respectively whereas, Mean and Median BMI of unmarried people are 25.21 and 23.50. In histogram also, it is skewed to the right(greater BMI) which belongs to married population. In Boxplot also, the 3rd quantile of Unmarried people is in between 1st quartile and mean BMI of married people. That means, the proposition is true, people trends to get weight after marriage.

```
summary(clean_g_stroke_data)
```

```
##        id            gender               age          hypertension
```

```
##  Min.   :   77   Length:4908        Min.   : 0.08   Min.   :0.00000
##  1st Qu.:18602   Class :character   1st Qu.:25.00   1st Qu.:0.00000
##  Median :37580   Mode  :character   Median :44.00   Median :0.00000
##  Mean   :37060                      Mean   :42.87   Mean   :0.09189
##  3rd Qu.:55182                      3rd Qu.:60.00   3rd Qu.:0.00000
##  Max.   :72940                      Max.   :82.00   Max.   :1.00000
##  heart_disease     ever_married        work_type         Residence_type
##  Min.   :0.00000   Length:4908        Length:4908        Length:4908
##  1st Qu.:0.00000   Class :character   Class :character   Class :character
##  Median :0.00000   Mode  :character   Mode  :character   Mode  :character
##  Mean   :0.04951
##  3rd Qu.:0.00000
##  Max.   :1.00000
##  avg_glucose_level      bmi        smoking_status         stroke
##  Min.   : 55.12    Min.   :10.30   Length:4908        Min.   :0.00000
##  1st Qu.: 77.07    1st Qu.:23.50   Class :character   1st Qu.:0.00000
##  Median : 91.68    Median :28.10   Mode  :character   Median :0.00000
##  Mean   :105.30    Mean   :28.89                      Mean   :0.04258
##  3rd Qu.:113.50    3rd Qu.:33.10                      3rd Qu.:0.00000
##  Max.   :271.74    Max.   :97.60                      Max.   :1.00000
```

```r
clean_g_stroke_data$hypertension<-as.factor(clean_g_stroke_data$hypertension)
clean_g_stroke_data$heart_disease<-as.factor(clean_g_stroke_data$heart_disease)
clean_g_stroke_data$stroke<-as.factor(clean_g_stroke_data$stroke)
```

```r
install.packages("cowplot",repos="https://cloud.r-project.org/")
```

```
## Installing package into '/geode2/home/u060/mdprin/Carbonate/R/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
library(cowplot)
```

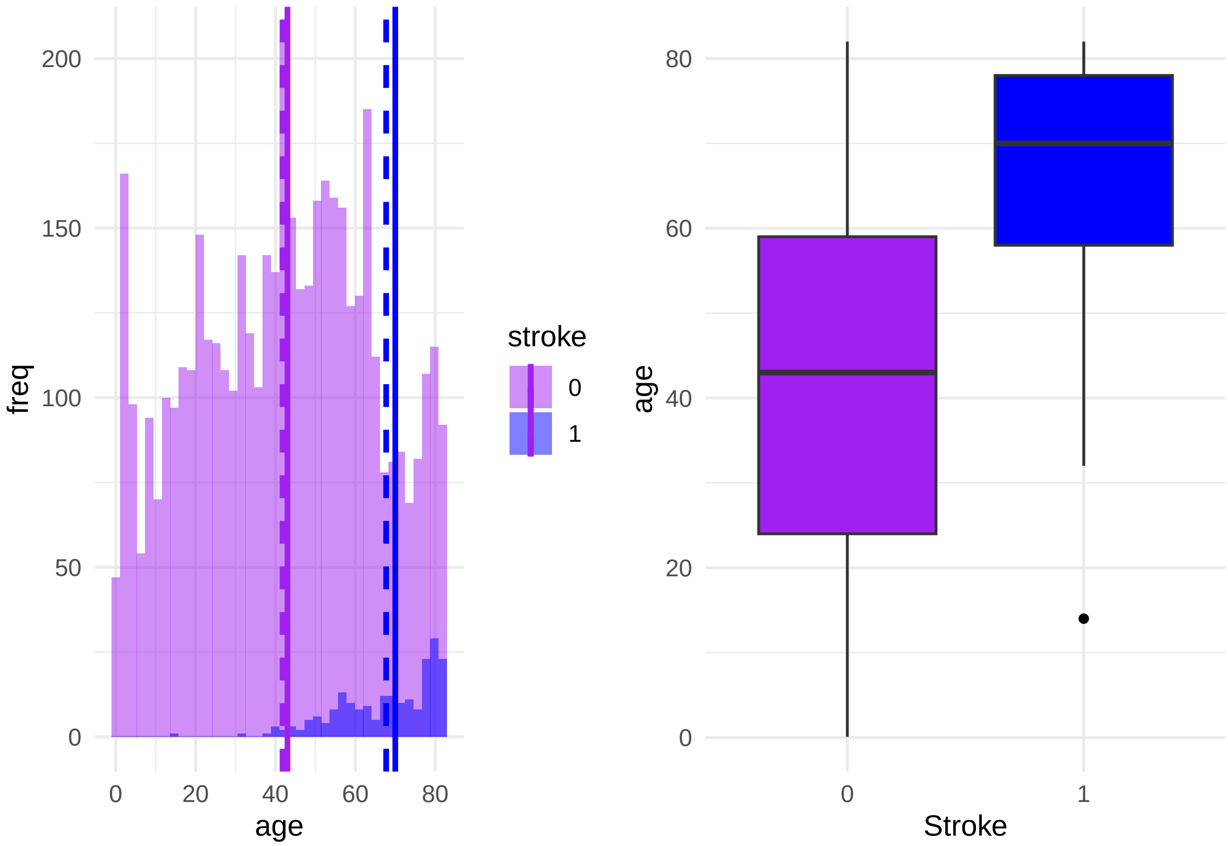
```
##
## Attaching package: 'cowplot'

## The following object is masked from 'package:lubridate':
##
##     stamp

## The following object is masked from 'package:ggthemes':
##
##     theme_map
```

```r
data<-clean_g_stroke_data

# a generic function to plot any column variable with respect of stroke
dist_plot<-function(param){
  #calculating the mean and the median
  #people who had a stroke
  mar_mean<-mean(data[[param]][data$stroke==1], na.rm=TRUE)
  mar_median<-median(data[[param]][data$stroke==1], na.rm=TRUE)
  #people who did not have a stroke
  unmar_mean<-mean(data[[param]][data$stroke==0], na.rm=TRUE)
  unmar_median<-median(data[[param]][data$stroke==0], na.rm=TRUE)


  #histogram by stroke status(e.g. age distribution by stroke)
```

```r
  hist <- ggplot(data, aes(x=eval(parse(text=param)), fill=stroke)) +
    geom_histogram(alpha=0.5, position="identity", bins=40) +
    scale_fill_manual(values=c("purple", "blue"))+
    geom_vline(aes(xintercept=mar_mean), color="blue", linetype="dashed", size=1, show.legend=TRUE, labe
    geom_vline(aes(xintercept=mar_median), color="blue", linetype="solid", size=1, show.legend=TRUE, lal
    geom_vline(aes(xintercept=unmar_mean), color="purple", linetype="dashed", size=1, show.legend=TRUE,
    geom_vline(aes(xintercept=unmar_median), color="purple", linetype="solid", size=1, show.legend=TRUE
    theme_minimal() +
    labs( x = param, y = "freq")


  box <- ggplot(data, aes(x=stroke, y=eval(parse(text=param)))) + #kind of opposite of histogram
  geom_boxplot(aes(fill=stroke), outlier.color="black", outlier.shape=16) +
  labs( x="Stroke", y=param) +
  theme_minimal() +
  scale_fill_manual(values=c("1"="blue", "0"="purple")) +  # Manual coloring
  theme(legend.position="none")  # Removing legend since fill color indicates category

  return(plot_grid(hist,box))
}
```

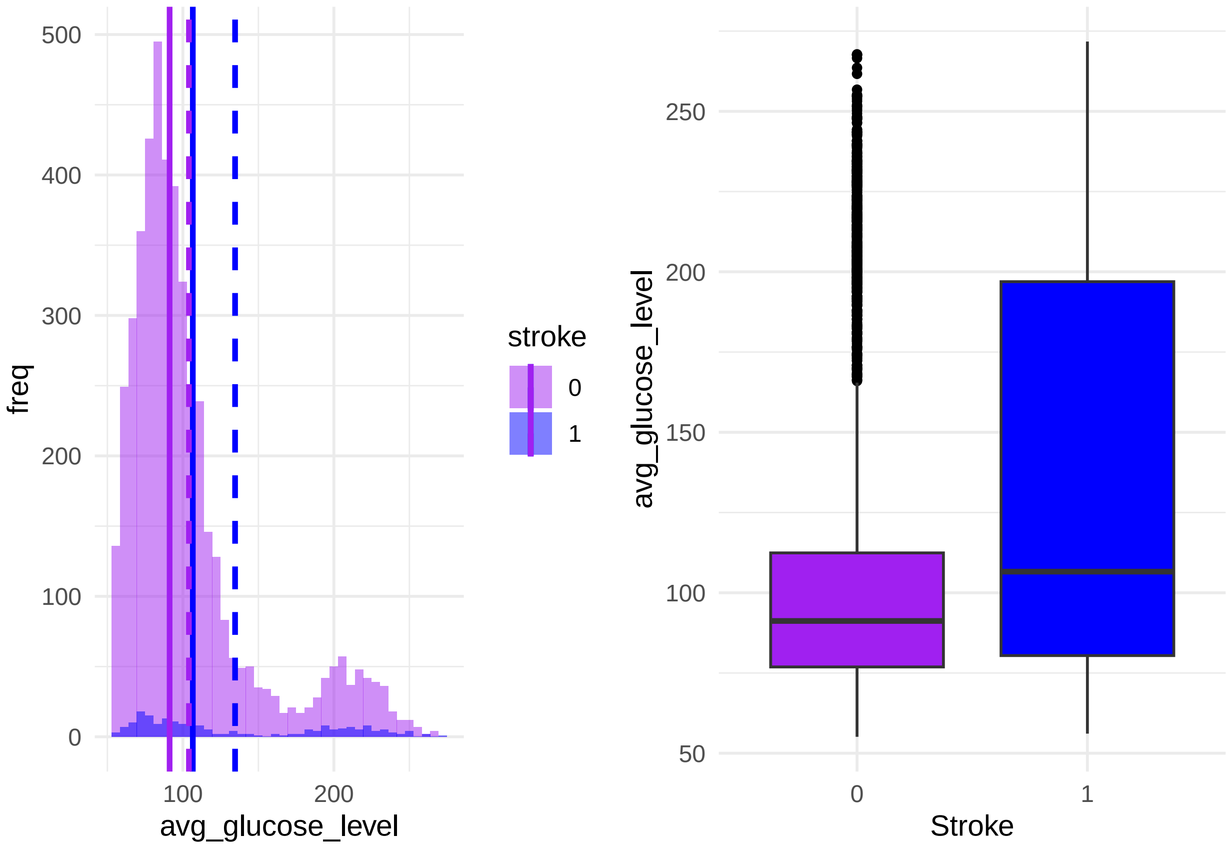
```r
dist_plot("age")
```

```
## Warning in geom_vline(aes(xintercept = mar_mean), color = "blue", linetype =
## "dashed", : Ignoring unknown parameters: `label`

## Warning in geom_vline(aes(xintercept = mar_median), color = "blue", linetype =
## "solid", : Ignoring unknown parameters: `label`

## Warning in geom_vline(aes(xintercept = unmar_mean), color = "purple", linetype
## = "dashed", : Ignoring unknown parameters: `label`

## Warning in geom_vline(aes(xintercept = unmar_median), color = "purple", :
## Ignoring unknown parameters: `label`
```




```r
dist_plot("avg_glucose_level")
```

```
## Warning in geom_vline(aes(xintercept = mar_mean), color = "blue", linetype =
## "dashed", : Ignoring unknown parameters: `label`

## Warning in geom_vline(aes(xintercept = mar_median), color = "blue", linetype =
## "solid", : Ignoring unknown parameters: `label`

## Warning in geom_vline(aes(xintercept = unmar_mean), color = "purple", linetype
## = "dashed", : Ignoring unknown parameters: `label`

## Warning in geom_vline(aes(xintercept = unmar_median), color = "purple", :
## Ignoring unknown parameters: `label`
```



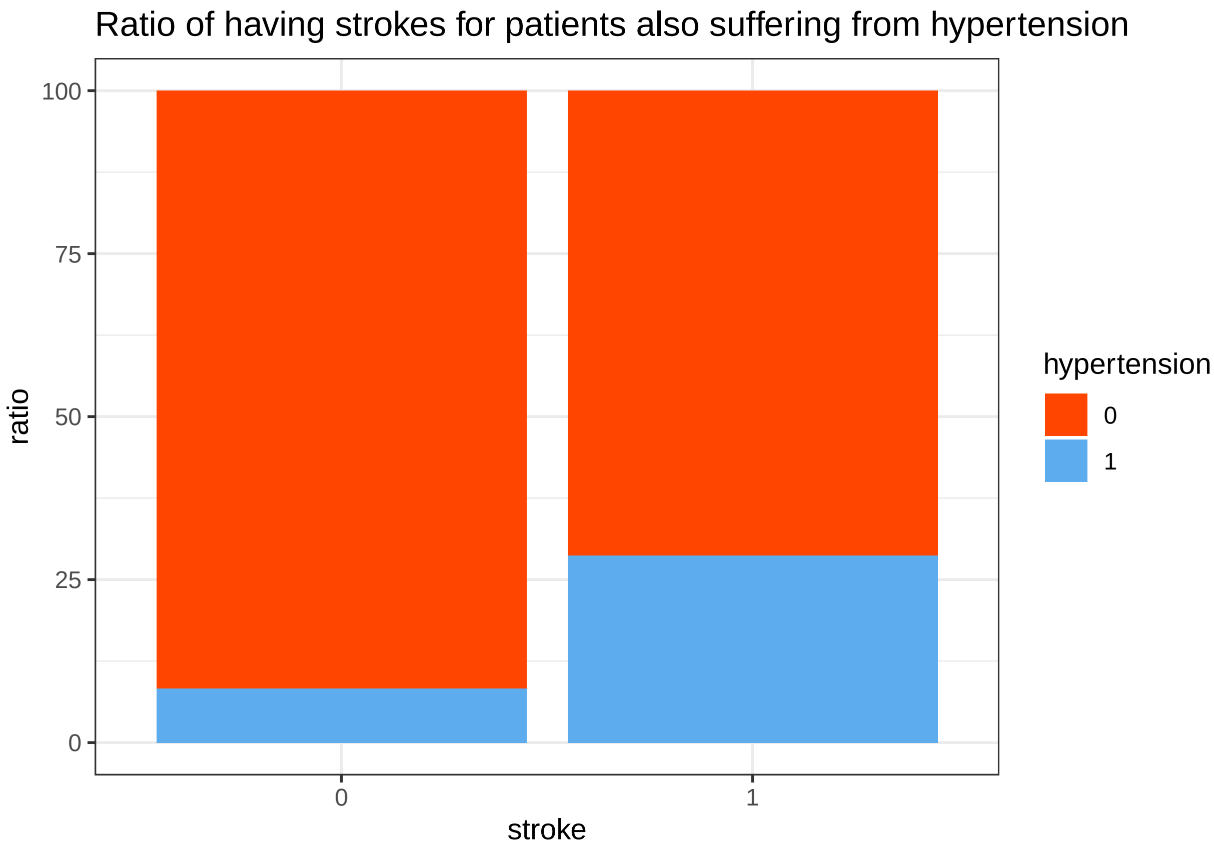

```
ratio_barpplot<-function(x){
  # here at first x will be in character format.
  barplt<-data %>% select(stroke, x, gender) %>%
            group_by(stroke,param=eval(parse(text=x))) %>% #parsing x from text to act as a column name
            summarise(count = length(gender)) %>%
            group_by(stroke) %>%
            mutate(ratio=round((count*100)/sum(count),1)) %>%
            ggplot(aes(x=stroke, y=ratio, fill=param)) +
            geom_bar(stat="identity") +
            scale_fill_manual(values=c("orangered", "steelblue2"))+

            labs(title = paste("Ratio of having strokes for patients also suffering from",x), fill=x)+
            theme_bw()
  return(barplt)
}
```

```
ratio_barpplot("hypertension")
```

```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(x)
##
##   # Now:
##   data %>% select(all_of(x))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
```

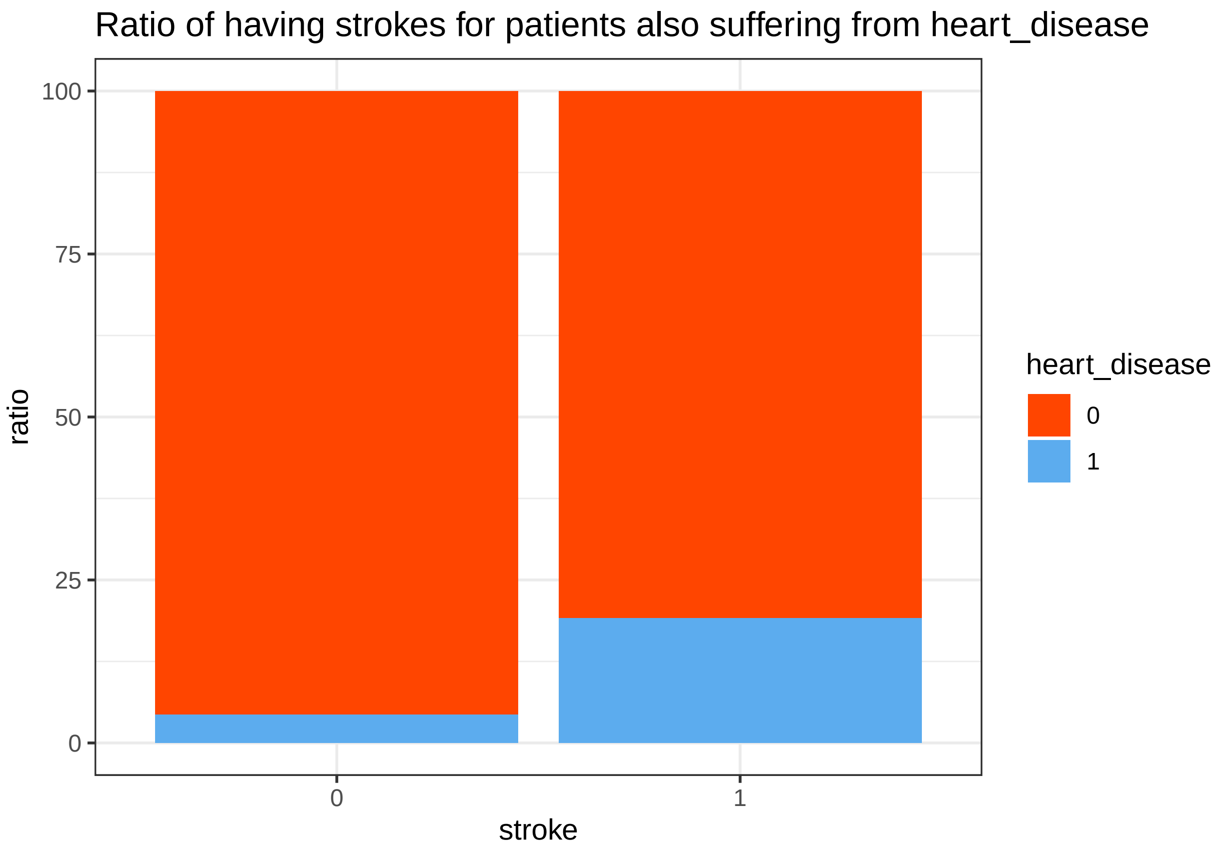
```
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `summarise()` has grouped output by 'stroke'. You can override using the
## `.groups` argument.
```



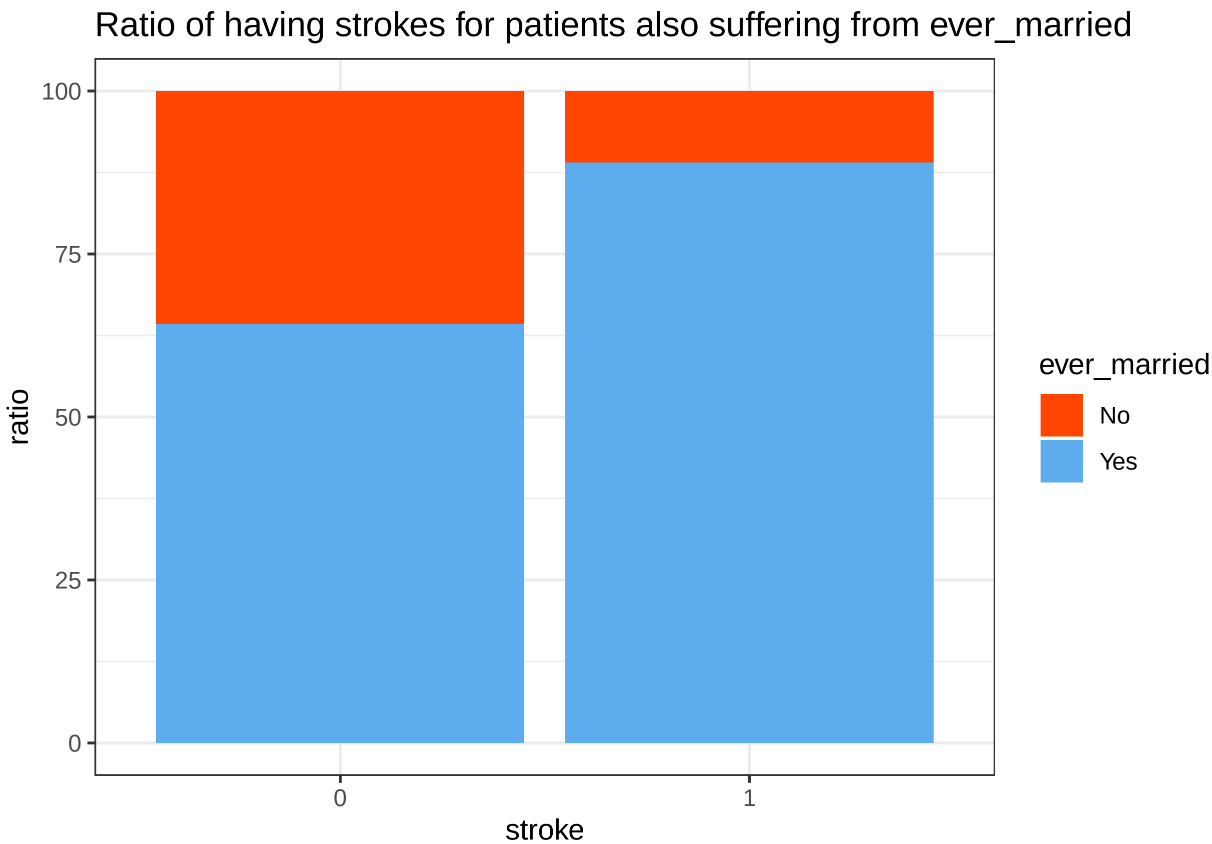

```r
ratio_barpplot("heart_disease")
```

```
## `summarise()` has grouped output by 'stroke'. You can override using the
## `.groups` argument.
```




```
ratio_barpplot("ever_married")
```

```
## `summarise()` has grouped output by 'stroke'. You can override using the
## `.groups` argument.
```



Predicting Stroke outcome

```r
install.packages("randomForest", repos = "https://cloud.r-project.org/")
```

```
## Installing package into '/geode2/home/u060/mdprin/Carbonate/R/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

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

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

```r
install.packages("caret", repos = "https://cloud.r-project.org/")
```

```
## Installing package into '/geode2/home/u060/mdprin/Carbonate/R/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'

## The following object is masked from 'package:httr':
##
##     progress
```

```
formula<-(stroke~gender+age+hypertension+heart_disease+ever_married+work_type+Residence_type+avg_glucose
rf_clf<-randomForest(formula=formula,data=data)
pred<-predict(object=rf_clf, newdata=select(data, -stroke), type="class")
confusionMatrix(data=pred, reference=data$stroke)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 4699   31
##          1    0  178
##
##                Accuracy : 0.9937
##                  95% CI : (0.991, 0.9957)
##     No Information Rate : 0.9574
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9166
##
##  Mcnemar's Test P-Value : 7.118e-08
##
##             Sensitivity : 1.0000
##             Specificity : 0.8517
##          Pos Pred Value : 0.9934
##          Neg Pred Value : 1.0000
##              Prevalence : 0.9574
##          Detection Rate : 0.9574
##    Detection Prevalence : 0.9637
##       Balanced Accuracy : 0.9258
##
##        'Positive' Class : 0
##
```

```
# Central Limit Theorem
bmi<-data$bmi[data$stroke==0]
#hist(bmi, col="lightblue",breaks=40)
print(paste("Mean BMI of the total population is", round(mean(bmi),1)))
```

```
## [1] "Mean BMI of the total population is 28.8"
```

```
s<-2000
bmi_set1<-rep(0,20)
bmi_set2<-rep(0,200)
bmi_set3<-rep(0,2000)
for (i in 1:s){
  bmi_sample<-sample(bmi,size=20,replace=F)
  if (i<=2000){
    bmi_set3[i]<-mean(bmi_sample)
  }
```

```r
  if(i<=200){
    bmi_set2[i]<-mean(bmi_sample)
  }
  if(i<=20){
    bmi_set1[i]<-mean(bmi_sample)
  }
}
```

```r
library(reshape2)
```

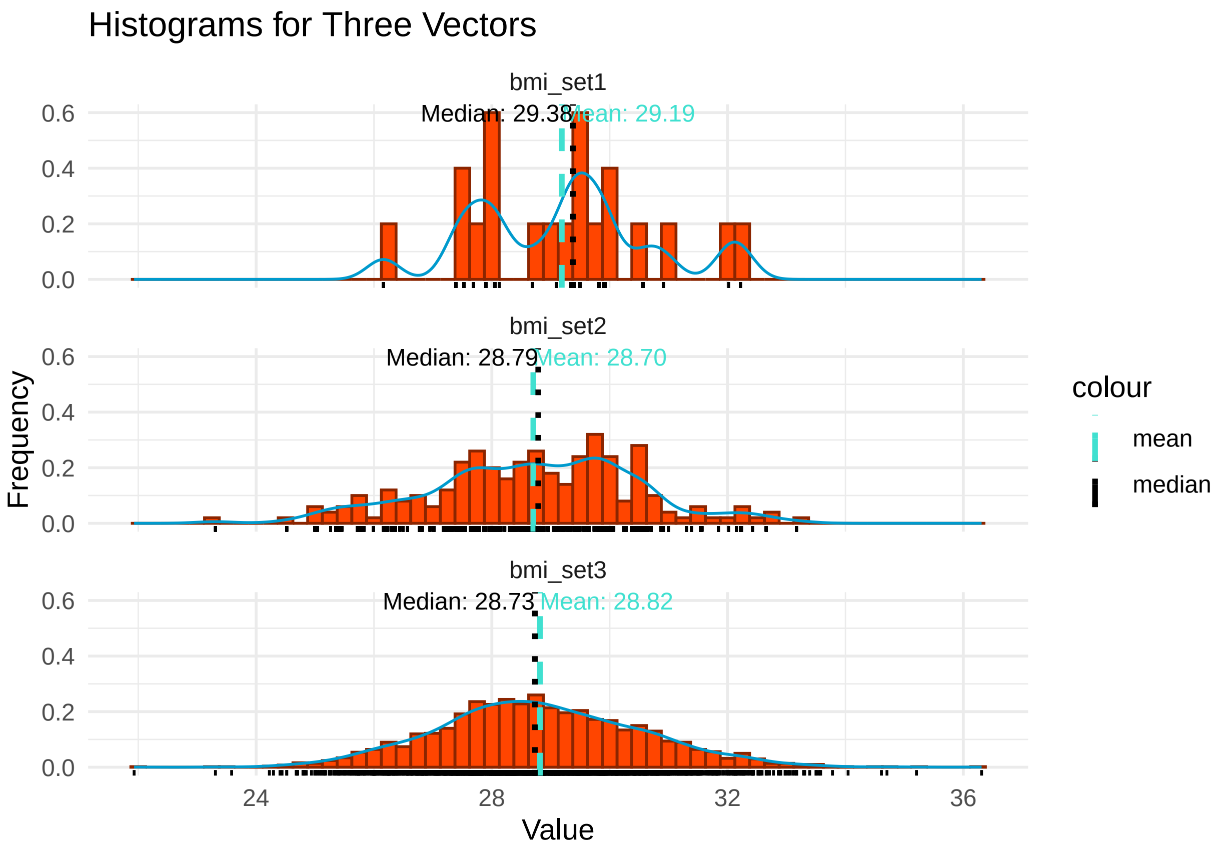
```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

```r
# Combine into a single data frame
df <- data.frame(bmi_set1, bmi_set2, bmi_set3)
mean_vals <- sapply(df, mean)
median_vals <- sapply(df, median)

# Create data frames for mean and median
mean_df <- data.frame(variable = names(mean_vals), value = mean_vals)
median_df <- data.frame(variable = names(median_vals), value = median_vals)
# Melt the data
melted_df <- melt(df)
```

```
## No id variables; using all as measure variables
```

```r
# Plot using ggplot2
ggplot(melted_df, aes(x=value)) +
  geom_histogram(aes(y=..density..), binwidth=0.25, bins=100, fill="orangered", color="orangered4", alph
  geom_density(color="deepskyblue3") +
  geom_rug() +
  geom_vline(data=mean_df, aes(xintercept=value, color="mean"), linetype="dashed", size=1) +
  geom_vline(data=median_df, aes(xintercept=value, color="median"), linetype="dotted", size=1) +
  geom_text(data=mean_df, aes(x=value, label=sprintf("Mean: %.2f", value), y=Inf, hjust=0), vjust=1, si
  geom_text(data=median_df, aes(x=value, label=sprintf("Median: %.2f", value), y=Inf, hjust=1), vjust=1
  facet_wrap(~variable, ncol=1) +
  scale_color_manual(values=c("mean"="turquoise", "median"="black")) +
  theme_minimal() +
  labs(title="Histograms for Three Vectors", x="Value", y="Frequency")
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



So, we can observe as the sample set is becoming larger and larger, the sampling distribution of the sample mean is having a normal or central tendency. And according to the Central Limit Theorem, mean of the sampling distribution of the sample mean is becoming equal to that of the actual population mean. 28.95 –> 28.97 –> 28.82 (whre, original population mean= 28.8)

```
#Law of Large Numbers (LLN)
#converting [1,2] to [0,1] using levels
hyper<-as.numeric(levels(data$hypertension))[data$hypertension]
str(hyper)
```

```
##  num [1:4908] 0 0 0 1 0 1 0 0 1 0 ...
```

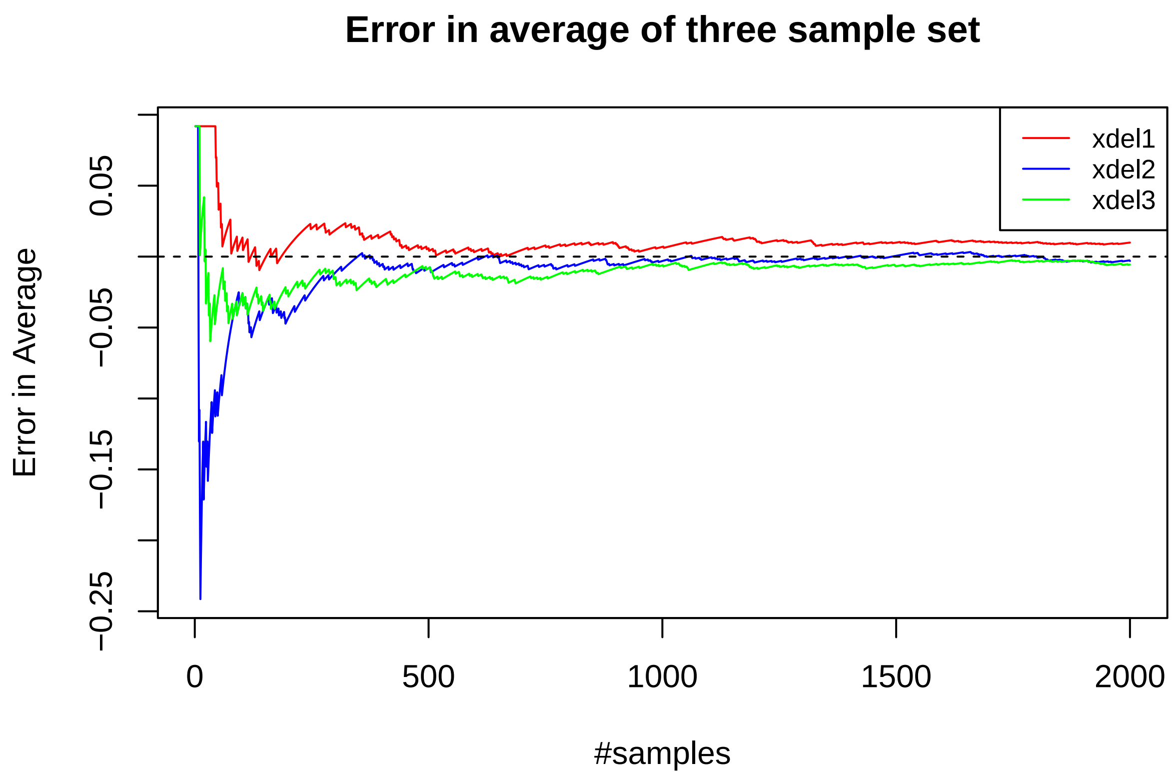
```
avg<-mean(hyper==1)
```

```
samples<- 2000
set1<-sample(hyper, samples, replace=FALSE)
set2<-sample(hyper, samples, replace=FALSE)
set3<-sample(hyper, samples, replace=FALSE)
```

```
xdel1<-rep(0,samples)
xdel2<-rep(0,samples)
xdel3<-rep(0,samples)
for (i in 1:samples){
  xdel1[i]<-mean(set1[1:i])
  xdel2[i]<-mean(set2[1:i])
  xdel3[i]<-mean(set3[1:i])
}
```

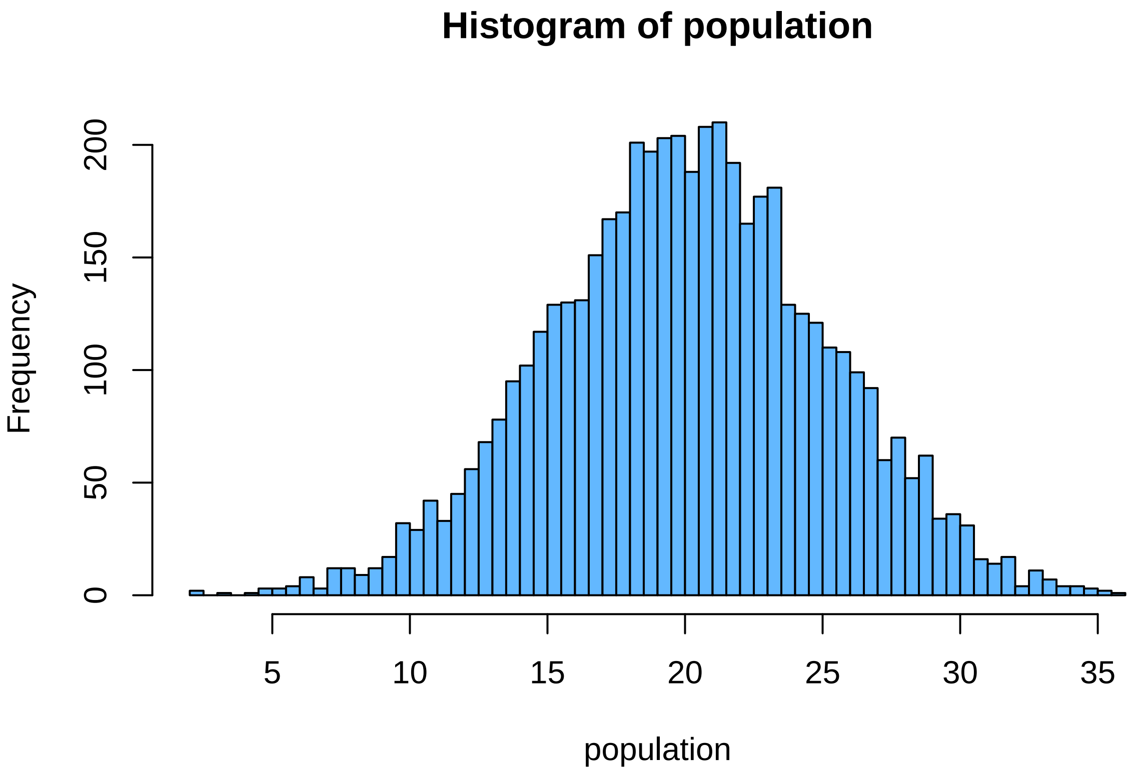
```
# Create a sequence of indices for x-axis
x <- seq(1, samples, by=1)
```

```r
# Plot the first vector
plot(x, avg-xdel1, type="l", col="red", ylim=c(min(avg-xdel1, avg-xdel2, avg-xdel3), max(avg-xdel1, avg-

# Add the second vector
lines(x, avg-xdel2, col="blue")

# Add the third vector
lines(x, avg-xdel3, col="green")

# Add the horizontal line at y=0
abline(h=0, col="black", lty=2)  # lty=2 makes the line dashed

# Add a legend
legend("topright", legend=c("xdel1", "xdel2", "xdel3"), col=c("red", "blue", "green"), lty=1, cex=0.8)
```



Hence, as the sample size become larger and larger the error in calculating ratio of people having hypertension trends to zero (a pure demonstration of Law of Large Numbers(LLN))

```r
#Confidence Intervals
# first using a random population
population<-rnorm(5000, 20, 5)
hist(population, breaks=50, col="steelblue1")
```




```
pop_mean=round(mean(population),3)
pop_sd=round(sd(population),3)
print(paste0("Population Mean ", pop_mean))
```

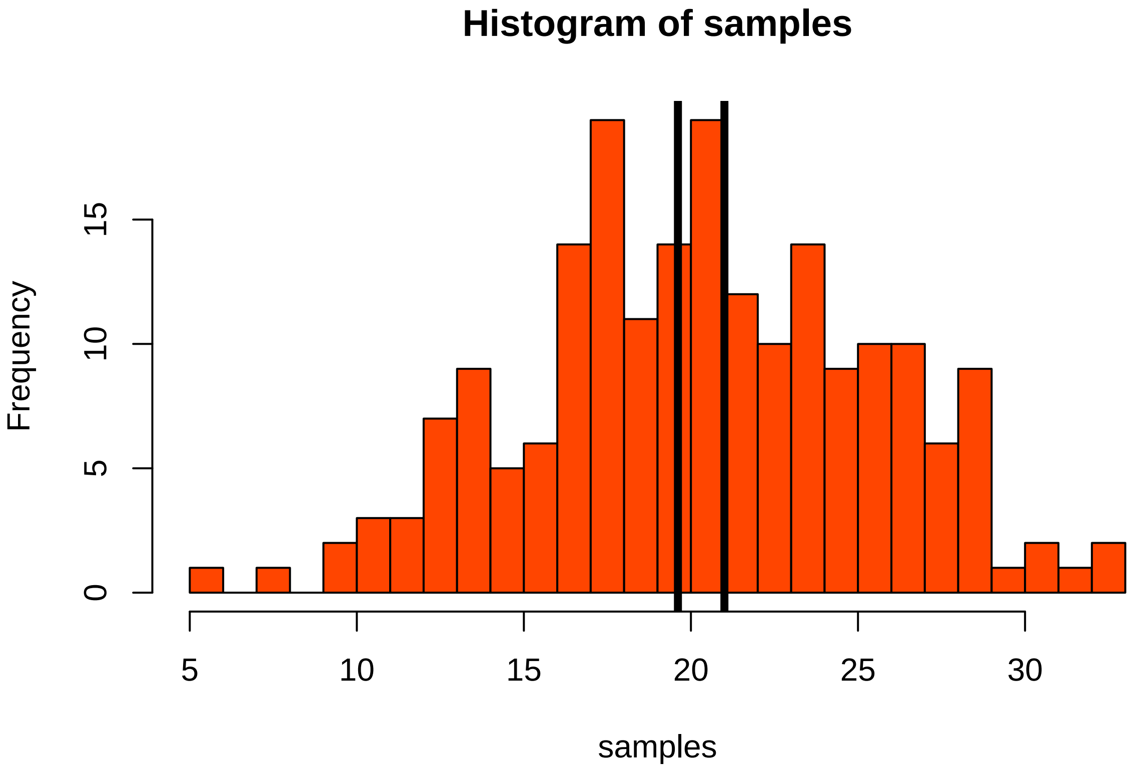
```
## [1] "Population Mean 20.085"
```

```
print(paste0("Population Std Err ", pop_sd))
```

```
## [1] "Population Std Err 5.019"
```

```
# constructing confidence interval:

# taking n=200 samples and trying to estimate the true population mean
n=200
samples<-sample(population, n, replace=FALSE)
samp_mean<-mean(samples)
# say, our claim is mean has changed.
# So, Null Hypothesis H_o=> mu = 19.977
# and, alternative Hypothesis H_a=> mu != 19.977

# for 95% CI:
alpha<- 0.05
quant<- 1-alpha/2
z<-qnorm(quant,lower.tail=TRUE) # calculating z-score from z-table
# using Z-statistic needs to know the true population std dev.
# assuming we know the true population std. dev (from previous step)
se<-pop_sd/sqrt(n)

upper_bound<-samp_mean + z*se
lower_bound<-samp_mean - z*se
```

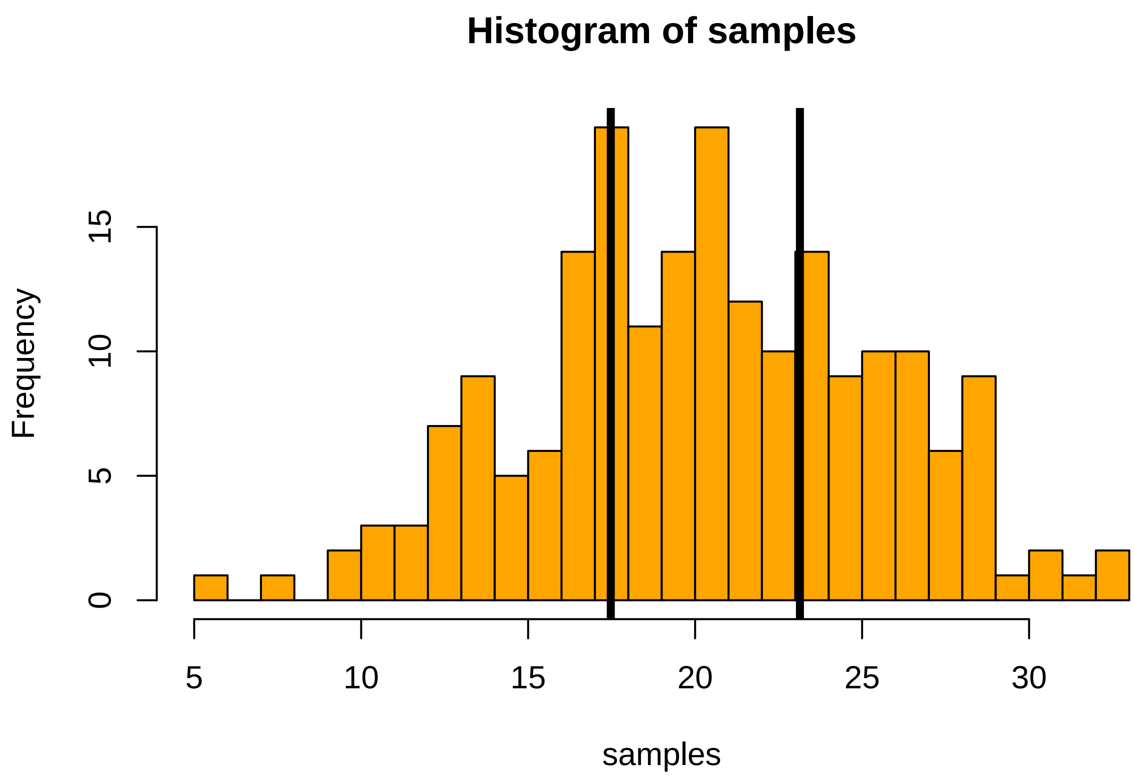
```r
hist(samples,breaks=20, col="orangered")
abline(v=upper_bound,lwd=4)
abline(v=lower_bound, lwd=4)
```




```r
print(paste0("With 95% confidence we can say that population mean is in interval [", round(lower_bound,
```

```
## [1] "With 95% confidence we can say that population mean is in interval [19.6; 21]"
```

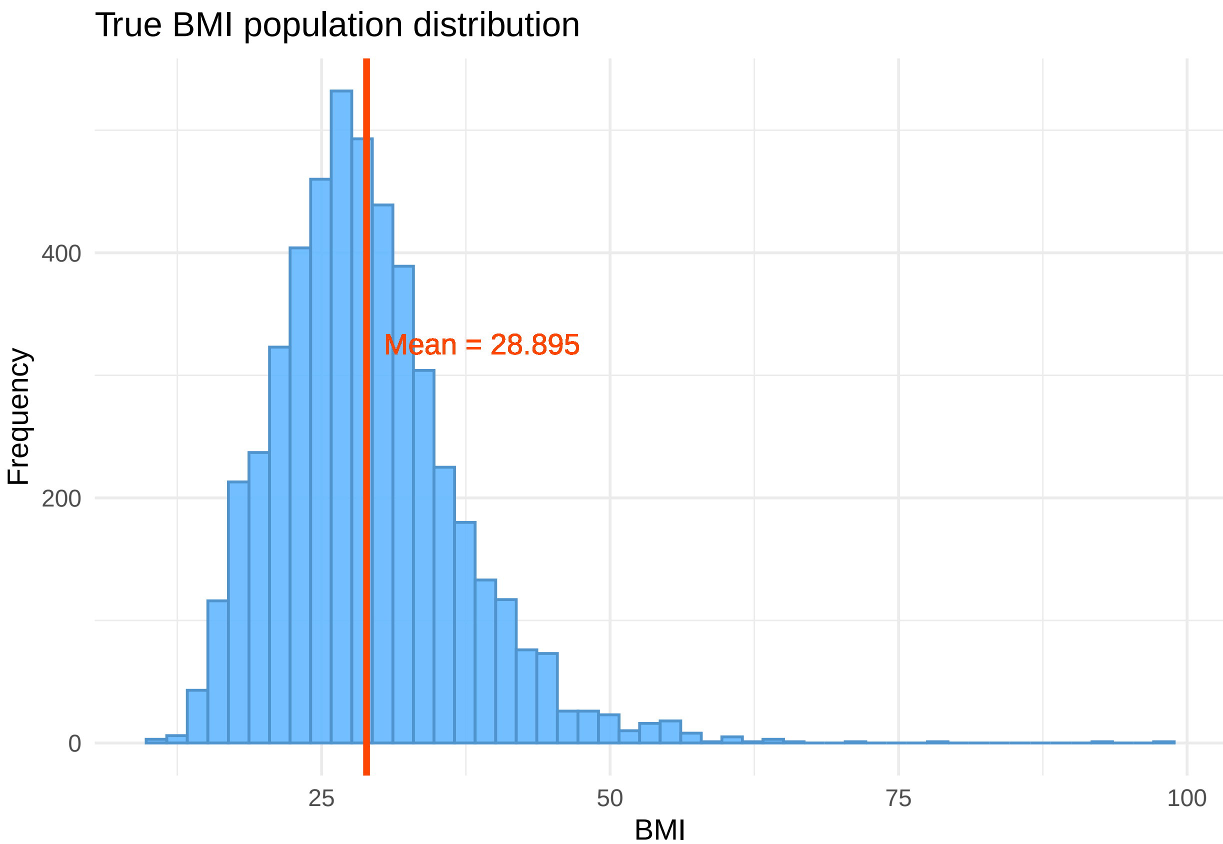
```r
# using t-statistic
# for t-statistic we do not have to have any prior knowledge(mean,sd) of the original population mean
dff<- n-1
t<-qt(quant,dff)
se<-mean(samples)/sqrt(n)
upper_bound<-samp_mean + t*se
lower_bound<-samp_mean - t*se

hist(samples,breaks=20, col="orange")
abline(v=upper_bound,lwd=4)
abline(v=lower_bound, lwd=4)
```




```
print(paste0("With 95% confidence we can say that population mean is in interval [", round(lower_bound,
```

```
## [1] "With 95% confidence we can say that population mean is in interval [17.5; 23.1]"
```

```
all_bmi<- data$bmi
bmi_pop_mean<-round(mean(all_bmi),3)
p <- ggplot(data.frame(all_bmi), aes(x = all_bmi)) +
  geom_histogram(bins = 50, fill = "steelblue1",col="steelblue3", alpha = 0.9) +
  geom_vline(aes(xintercept = bmi_pop_mean), color = "orangered1", linetype = "solid", size = 1.2) +
  geom_text(aes(x = bmi_pop_mean+10,y=300, label = paste("Mean =", bmi_pop_mean)), vjust = -1, hjust = 
  xlab("BMI") +
  ylab("Frequency") +
  theme_minimal()+
  labs(title="True BMI population distribution")

# Show the plot
print(p)
```

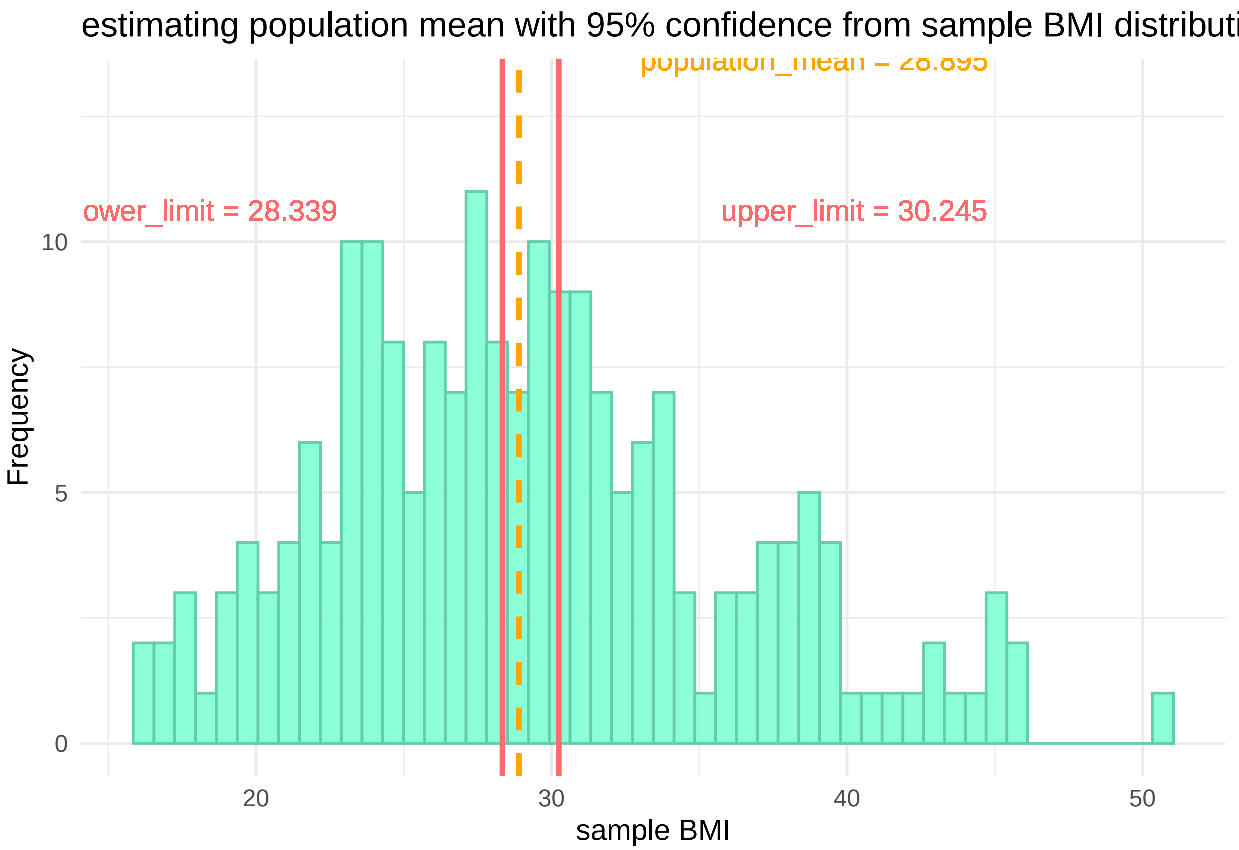



```
print(paste0("True population mean= ", bmi_pop_mean))
```

```
## [1] "True population mean= 28.895"
```

```
# 95% CI on bmi
# for t-statistic we do not have to have any prior knowledge(mean,sd) of the original population mean
sample_size<-200
samp_bmi<-sample(all_bmi, sample_size)
samp_bmi_mean<-round(mean(samp_bmi),3)
samp_bmi_sd<-sd(samp_bmi)
alpha<-0.05# as Confidence level= 95%
quant<-(1-alpha/2)
df_bmi<-sample_size-1
t_score<-qt(quant,df_bmi)
bmi_conf_upper<-samp_bmi_mean + t_score*(samp_bmi_sd/sqrt(sample_size))
bmi_conf_lower<-samp_bmi_mean - t_score*(samp_bmi_sd/sqrt(sample_size))


q <- ggplot(data.frame(samp_bmi), aes(x = samp_bmi)) +
  geom_histogram(bins = 50, fill = "aquamarine1",col="aquamarine3", alpha = 0.9) +
  geom_vline(aes(xintercept = bmi_conf_upper), color = "indianred1", linetype = "solid", size = 1) +
  geom_text(aes(x = bmi_conf_upper+10,y=10, label = paste("upper_limit =", round(bmi_conf_upper,3))), v
  geom_vline(aes(xintercept = bmi_conf_lower), color = "indianred1", linetype = "solid", size = 1) +
  geom_text(aes(x = bmi_conf_lower-10,y=10, label = paste("lower_limit =", round(bmi_conf_lower,3))), v
  geom_vline(aes(xintercept = bmi_pop_mean), color = "orange", linetype = "dashed", size = 1) +
  geom_text(aes(x = bmi_pop_mean+10,y=13, label = paste("population_mean =", bmi_pop_mean)), vjust = -1
  xlab("sample BMI") +
  ylab("Frequency") +
```

```r
  theme_minimal()+
  labs(title="estimating population mean with 95% confidence from sample BMI distribution ")

# Show the plot
print(q)
```




```r
print(paste0("sample population mean= ", samp_bmi_mean))
```

```
## [1] "sample population mean= 29.292"
```

```r
print(paste0("With 95% confidence we can say that population mean is in interval [", round(bmi_conf_lowe
```

```
## [1] "With 95% confidence we can say that population mean is in interval [28.3; 30.2]"
```

```r
sample_n(data,10)
```

```
##       id gender age hypertension heart_disease ever_married     work_type
## 1  28559   Male   2            0             0           No      children
## 2  47330   Male   9            0             0           No      children
## 3  58037   Male  21            0             0           No       Private
## 4  56976 Female  42            0             0          Yes       Private
## 5  49976 Female  54            0             1          Yes       Private
## 6  20364 Female   4            0             0           No      children
## 7  56233 Female  44            0             0           No       Private
## 8  54347   Male  61            0             0          Yes Self-employed
## 9  55680   Male  13            0             0           No      children
## 10 12807 Female  63            1             0          Yes       Private
##    Residence_type avg_glucose_level  bmi  smoking_status stroke
```

```
## 1           Urban             88.54 17.5         Unknown      0
## 2           Rural             60.39 16.4         Unknown      0
## 3           Rural             78.52 27.2    never smoked      0
## 4           Urban             96.01 38.7         Unknown      0
## 5           Urban            140.28 37.1 formerly smoked      0
## 6           Urban            107.25 12.0         Unknown      0
## 7           Rural            116.95 26.1    never smoked      0
## 8           Rural            155.32 26.6 formerly smoked      0
## 9           Urban            114.84 18.3         Unknown      0
## 10          Urban             81.54 24.2    never smoked      0
```

```r
# t- test
str<-data$age[data$stroke==1]
no_str<-data$age[data$stroke==0]
str_df <- data.frame(str)

# Create a data frame for individuals without stroke
no_str_df <- data.frame(no_str)

# Summarise the data frame for individuals with stroke
summary_str <- summarise(str_df,
                         mean_age = mean(str, na.rm = TRUE),
                         median_age = median(str, na.rm = TRUE),
                         sd_age = sd(str, na.rm = TRUE),
                         min_age = min(str, na.rm = TRUE),
                         max_age = max(str, na.rm = TRUE),
                         count = nrow(str_df))

# Summarise the data frame for individuals without stroke
summary_no_str <- summarise(no_str_df,
                            mean_age = mean(no_str, na.rm = TRUE),
                            median_age = median(no_str, na.rm = TRUE),
                            sd_age = sd(no_str, na.rm = TRUE),
                            min_age = min(no_str, na.rm = TRUE),
                            max_age = max(no_str, na.rm = TRUE),
                            count = nrow(no_str_df))
```

```r
# Print summaries
print(summary_str)
```

```
##   mean_age median_age   sd_age min_age max_age count
## 1 67.71292         70 12.40285      14      82   209
```

```r
print(summary_no_str)
```

```
##   mean_age median_age   sd_age min_age max_age count
## 1 41.76381         43 22.26931    0.08      82  4699
```

```r
#difference in average age for people having and not having a stroke:
mean(str)-mean(no_str)
```

```
## [1] 25.94911
```

So, we see here the average age difference = 25.949 our hypothesis: H_o : there is no age difference, mu= 0 H_a : there is a difference, mu != 0

```r
len_str<- length(str)
var_str<-var(str)


len_no_str<- length(no_str)
var_no_str<-var(no_str)

diff_mean<- mean(no_str)-mean(str)

se<-sqrt(var_str/len_str + var_no_str/len_no_str)


alpha<-0.05
quant<-(1-alpha/2)
df_age<-len_str+len_no_str-2

t_age<-qt(quant,df_age)

upper_limit<- diff_mean+t_age*se
lower_limit<- diff_mean-t_age*se

print(paste0("With 95% confidence we can say that population mean is in interval [", round(upper_limit,
```

```
## [1] "With 95% confidence we can say that population mean is in interval [-24.2; -27.7]"
```

As, the confidence interval doesn't include 0 rather is way below 0. So, with 95% confidence we can say that the age difference in between people having stroke and not having stroke is significant. So, we reject our Null Hypothesis (H_o) and accept alternative hypothesis (H_a)

```r
# alternative way:
t_age<- (diff_mean - 0)/se
print(paste("t score",t_age))
```

```
## [1] "t score -28.2863881984247"
```

```r
p_value <- pt(-abs(t_age), df_age, lower.tail = TRUE)
print(paste("Probability of having a t_score of that is:", p_value))
```

```
## [1] "Probability of having a t_score of that is: 1.70630768710462e-163"
```

Here, P_value « alpha : reject H_o accept H_a Hence, the age difference is significant.

```r
#using R library

t.test(age~stroke, data=data, var.equal=FALSE, paired=FALSE,conf.level=0.95)
```

```
##
##  Welch Two Sample t-test
##
## data:  age by stroke
## t = -28.286, df = 271.68, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -27.75517 -24.14305
## sample estimates:
## mean in group 0 mean in group 1
##        41.76381        67.71292
```

```r
# using F statistic
var.test(age~stroke, data= data)
```

```
##
##  F test to compare two variances
##
## data:  age by stroke
## F = 3.2238, num df = 4698, denom df = 208, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  2.624126 3.890156
## sample estimates:
## ratio of variances
##           3.223819
```

```r
install.packages("bootstrap", repos = "https://cloud.r-project.org/")
```

```
## Installing package into '/geode2/home/u060/mdprin/Carbonate/R/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
library(bootstrap)
```

```r
#Resampling
bmi_len<- length(bmi)
bias_var=function(x){
  n=length(x)
  (n-1)*var(x)/n
}
jackknife(bmi, bias_var)
```

```
## $jack.se
## [1] 2.120335
##
## $jack.bias
## [1] -0.01331039
##
## $jack.values
##    [1] 62.52059 62.52261 62.51871 62.53487 62.52540 62.44916 62.51919 62.54482
##    [9] 62.54296 62.40409 62.53741 62.53562 62.52172 62.54395 62.50630 62.54182
##   [17] 62.52781 62.53702 62.54173 62.54274 62.50969 62.32649 62.54401 62.54504
##   [25] 62.53994 62.54155 62.54549 62.54474 62.54457 62.54433 62.54242 62.54304
##   [33] 62.54516 62.51052 62.54482 62.54389 62.54258 62.54462 62.54209 62.53927
##   [41] 62.54546 62.52195 62.52895 62.52702 62.53457 62.54526 62.54516 62.54551
##   [49] 62.54466 62.54068 62.54552 62.54551 62.53767 62.54258 62.53661 62.54428
##   [57] 62.54401 62.51847 62.50215 62.53233 62.54549 62.53316 62.54433 62.53380
##   [65] 62.54535 62.53915 62.54453 62.54526 62.53457 62.52477 62.54521 62.54258
##   [73] 62.51320 62.53781 62.53633 62.54118 62.53915 62.51847 62.54345 62.53533
##   [81] 62.52601 62.54395 62.54494 62.54371 62.40299 62.54529 62.52328 62.54490
##   [89] 62.54225 62.54274 62.52662 62.54553 62.46561 62.53843 62.54173 62.54359
##   [97] 62.54395 62.54553 62.54497 62.53148 62.54524 62.53024 62.53961 62.53972
##  [105] 62.54274 62.52127 62.53950 62.53904 62.26999 62.54542 62.54137 62.54533
##  [113] 62.54225 62.53715 62.54296 62.49905 62.53250 62.47286 62.51372 62.52951
##  [121] 62.54098 62.54389 62.53442 62.54453 62.54443 62.50773 62.54504 62.53078
##  [129] 62.53576 62.51424 62.50969 62.50913 62.51267 62.51627 62.54550 62.54490
##  [137] 62.52800 62.49255 62.53781 62.54146 62.53647 62.54428 62.52081 62.54542
```

```
## [145] 62.52261 62.53457 62.52838 62.51750 62.51133 62.51919 62.53316 62.51078
## [153] 62.54016 62.53994 62.54209 62.52349 62.52800 62.54443 62.53880 62.52456
## [161] 62.53426 62.53364 62.51651 62.54304 62.47902 62.51475 62.47826 62.54428
## [169] 62.54407 62.53688 62.53216 62.53199 62.53891 62.53619 62.49747 62.51750
## [177] 62.54549 62.54377 62.53938 62.40299 62.54173 62.53165 62.54443 62.54352
## [185] 62.54365 62.52059 62.54551 62.54544 62.53562 62.54546 62.54553 62.54407
## [193] 62.54474 62.53880 62.54535 62.53199 62.54553 62.54453 62.54546 62.50659
## [201] 62.53095 62.54494 62.52682 62.53880 62.54513 62.48844 62.53503 62.54137
## [209] 62.51577 62.52702 62.45186 62.54531 62.54258 62.54466 62.33595 62.54516
## [217] 62.54542 62.54504 62.54510 62.53619 62.53806 62.54448 62.53831 62.54250
## [225] 62.53591 62.53702 62.54318 62.54412 62.54383 62.52895 62.54389 62.54433
## [233] 62.53473 62.54098 62.54242 62.53233 62.54553 62.54537 62.49454 62.44084
## [241] 62.52035 62.53818 62.53741 62.53348 62.54552 62.54037 62.53457 62.51293
## [249] 62.54490 62.54146 62.54304 62.54478 62.53793 62.52581 62.54377 62.53547
## [257] 62.52838 62.54537 62.53517 62.53972 62.52035 62.51651 62.53380 62.51500
## [265] 62.54191 62.53250 62.48983 62.54407 62.54547 62.51078 62.52498 62.54281
## [273] 62.53983 62.52172 62.54553 62.54365 62.54433 62.53216 62.54325 62.54453
## [281] 62.53856 62.54457 62.54535 62.54504 62.43606 62.53042 62.54541 62.54486
## [289] 62.54068 62.15049 62.54098 62.51919 62.53728 62.54542 62.54438 62.54553
## [297] 62.52127 62.53715 62.54535 62.52622 62.54549 62.54457 62.51449 62.51676
## [305] 62.54547 62.51106 62.54521 62.51320 62.53517 62.51160 62.54173 62.54339
## [313] 62.53661 62.52581 62.51701 62.54478 62.54118 62.54428 62.53818 62.54377
## [321] 62.53576 62.53781 62.54539 62.53576 62.51601 62.54549 62.53250 62.52895
## [329] 62.54457 62.52240 62.53868 62.54526 62.53283 62.54137 62.52662 62.50687
## [337] 62.54311 62.54466 62.54304 62.53950 62.44826 62.54395 62.51823 62.54098
## [345] 62.54470 62.53927 62.54462 62.53972 62.52371 62.54526 62.51424 62.54433
## [353] 62.51577 62.54533 62.52781 62.54552 62.54497 62.53647 62.54482 62.50801
## [361] 62.53938 62.51449 62.54546 62.54217 62.53533 62.52081 62.54547 62.46887
## [369] 62.52217 62.54539 62.54490 62.54433 62.53856 62.53182 62.53961 62.54519
## [377] 62.53517 62.53843 62.54533 62.54457 62.54513 62.54537 62.40409 62.53904
## [385] 62.53576 62.54482 62.53442 62.54546 62.53781 62.51726 62.54533 62.53332
## [393] 62.53818 62.54553 62.54457 62.53591 62.54526 62.51651 62.53831 62.54289
## [401] 62.53591 62.54539 62.48527 62.54549 62.54541 62.53395 62.53781 62.39176
## [409] 62.54118 62.54474 62.54289 62.53661 62.52195 62.54383 62.53250 62.39176
## [417] 62.54521 62.53078 62.54513 62.53533 62.54553 62.53395 62.54371 62.53380
## [425] 62.53042 62.54389 62.50124 62.53994 62.54513 62.52702 62.54250 62.53938
## [433] 62.50275 62.50744 62.52781 62.51024 62.53442 62.54098 62.52581 62.54359
## [441] 62.53880 62.53182 62.54549 62.54513 62.53856 62.51424 62.52127 62.54026
## [449] 62.53843 62.54242 62.49715 62.54490 62.54371 62.53755 62.54490 62.54332
## [457] 62.54549 62.37047 62.53517 62.52702 62.54433 62.49905 62.54433 62.53950
## [465] 62.52371 62.54137 62.54546 62.53130 62.54546 62.52560 62.53950 62.51214
## [473] 62.53487 62.53487 62.53006 62.54296 62.54535 62.53647 62.54365 62.54537
## [481] 62.54490 62.54155 62.54553 62.54173 62.54389 62.54296 62.51345 62.54524
## [489] 62.54428 62.54524 62.53182 62.49650 62.54486 62.51676 62.54552 62.54510
## [497] 62.53950 62.54365 62.53950 62.54551 62.54234 62.40844 62.54332 62.51293
## [505] 62.54482 62.53348 62.53915 62.54544 62.54389 62.54395 62.54516 62.54438
## [513] 62.54146 62.51798 62.54266 62.53927 62.54266 62.54037 62.54544 62.54058
## [521] 62.51919 62.54407 62.54127 62.51775 62.54553 62.53473 62.52838 62.50969
## [529] 62.54098 62.54443 62.53856 62.54258 62.51240 62.53755 62.54058 62.53078
## [537] 62.53042 62.54521 62.54482 62.54428 62.53767 62.53741 62.54551 62.54296
## [545] 62.54037 62.42820 62.54428 62.53216 62.53904 62.54234 62.54438 62.54535
## [553] 62.54507 62.54521 62.50246 62.51449 62.54549 62.53994 62.54535 62.54274
## [561] 62.53891 62.53233 62.53619 62.54497 62.54504 62.54504 62.50744 62.54428
## [569] 62.52172 62.53793 62.52895 62.53961 62.54412 62.51966 62.48199 62.54542
```

```
##  [577] 62.54191 62.53755 62.53843 62.53380 62.54510 62.53266 62.54200 62.54191
##  [585] 62.53380 62.54164 62.54457 62.54137 62.54359 62.53915 62.52105 62.49779
##  [593] 62.54474 62.53266 62.54225 62.54521 62.53216 62.54507 62.52988 62.53487
##  [601] 62.44734 62.54401 62.54137 62.52392 62.54552 62.54516 62.49842 62.54482
##  [609] 62.54549 62.53818 62.51942 62.54457 62.54209 62.54537 62.50997 62.54412
##  [617] 62.54531 62.53605 62.54377 62.54146 62.54526 62.54352 62.54359 62.54526
##  [625] 62.53927 62.54118 62.54507 62.54552 62.52933 62.53316 62.54389 62.51078
##  [633] 62.53332 62.54494 62.54217 62.54296 62.54519 62.53818 62.54537 62.54423
##  [641] 62.53576 62.53688 62.54521 62.53688 62.54542 62.54365 62.53904 62.54553
##  [649] 62.52702 62.54524 62.54318 62.54553 62.51798 62.54200 62.53675 62.53165
##  [657] 62.52819 62.48740 62.54281 62.54209 62.51267 62.53487 62.53831 62.03069
##  [665] 62.54494 62.54466 62.53661 62.54535 62.53591 62.52642 62.51397 62.53781
##  [673] 62.54519 62.53182 62.54377 62.54519 62.54470 62.50513 62.53095 62.53938
##  [681] 62.50030 62.52371 62.53983 62.53880 62.51024 62.50543 62.53113 62.52456
##  [689] 62.54448 62.54519 62.54325 62.50543 62.54088 62.53927 62.50154 62.52540
##  [697] 62.54395 62.51726 62.54325 62.54500 62.54529 62.54217 62.53755 62.54553
##  [705] 62.53688 62.53675 62.54296 62.52328 62.50801 62.52761 62.53647 62.54108
##  [713] 62.53880 62.52059 62.52933 62.52581 62.54553 62.54542 62.53728 62.52059
##  [721] 62.53380 62.54531 62.54521 62.54311 62.54225 62.54490 62.54173 62.54173
##  [729] 62.52456 62.51214 62.54365 62.53250 62.52933 62.54552 62.54296 62.54546
##  [737] 62.44826 62.49051 62.54551 62.52702 62.48983 62.53781 62.54318 62.54401
##  [745] 62.54389 62.54453 62.54544 62.53856 62.54478 62.49187 62.54088 62.54371
##  [753] 62.54547 62.54466 62.54531 62.53457 62.53591 62.52601 62.54519 62.54407
##  [761] 62.54448 62.54250 62.54482 62.54289 62.54352 62.52240 62.54500 62.54545
##  [769] 62.53547 62.53741 62.54497 62.53831 62.54242 62.54155 62.54462 62.54383
##  [777] 62.54497 62.54500 62.53473 62.54552 62.51871 62.54550 62.54542 62.54448
##  [785] 62.54474 62.53395 62.54371 62.51449 62.54401 62.53904 62.52914 62.51989
##  [793] 62.53715 62.53533 62.41695 62.53891 62.54058 62.54549 62.54383 62.54423
##  [801] 62.54453 62.54438 62.51052 62.53283 62.50744 62.54526 62.53972 62.52195
##  [809] 62.49454 62.54504 62.39742 62.49187 62.54164 62.54524 62.51847 62.53856
##  [817] 62.54507 62.54494 62.53781 62.50857 62.54529 62.54058 62.53233 62.53619
##  [825] 62.54318 62.53915 62.51052 62.54332 62.53904 62.51078 62.53576 62.53562
##  [833] 62.54466 62.54352 62.52781 62.54443 62.53891 62.54146 62.53442 62.54304
##  [841] 62.51078 62.54164 62.53503 62.54524 62.53688 62.53576 62.54531 62.54549
##  [849] 62.53060 62.54395 62.53442 62.53755 62.54513 62.52059 62.54531 62.53950
##  [857] 62.51293 62.53755 62.54550 62.50744 62.54058 62.48633 62.52662 62.54545
##  [865] 62.52933 62.54497 62.54438 62.51871 62.50395 62.54549 62.53348 62.54533
##  [873] 62.54078 62.53688 62.52414 62.51775 62.54401 62.54529 62.54504 62.54542
##  [881] 62.54553 62.54274 62.54118 62.52800 62.54545 62.52240 62.53856 62.52172
##  [889] 62.54544 62.52195 62.54553 62.54200 62.54457 62.52933 62.51871 62.49842
##  [897] 62.51345 62.54466 62.54433 62.54510 62.54258 62.54182 62.54478 62.50335
##  [905] 62.50062 62.54146 62.50275 62.53647 62.54217 62.54494 62.53457 62.54438
##  [913] 62.52933 62.54542 62.51160 62.54524 62.54423 62.54457 62.52876 62.53517
##  [921] 62.54482 62.53148 62.53619 62.51133 62.54088 62.53503 62.53591 62.54513
##  [929] 62.52477 62.54325 62.51240 62.52933 62.53591 62.54225 62.53831 62.54549
##  [937] 62.53781 62.54389 62.52662 62.53702 62.54389 62.54552 62.50030 62.54438
##  [945] 62.54537 62.53503 62.53927 62.53831 62.53411 62.50246 62.54551 62.54553
##  [953] 62.54474 62.54541 62.54551 62.54318 62.54494 62.53831 62.53781 62.54412
##  [961] 62.53688 62.53904 62.54433 62.54209 62.54359 62.46395 62.50215 62.54531
##  [969] 62.53380 62.53591 62.54365 62.52933 62.54552 62.53728 62.54258 62.54127
##  [977] 62.50997 62.53182 62.52702 62.52349 62.53856 62.52414 62.44458 62.53364
##  [985] 62.54217 62.53517 62.54457 62.54418 62.54438 62.49842 62.54191 62.48126
##  [993] 62.45452 62.50092 62.54462 62.52477 62.54552 62.54513 62.54494 62.54005
## [1001] 62.54552 62.54155 62.51676 62.54497 62.52622 62.52284 62.54339 62.49322
```

```
## [1009] 62.53130 62.54470 62.54191 62.52195 62.49650 62.54521 62.48704 62.54551
## [1017] 62.51871 62.54516 62.54217 62.54544 62.53332 62.54395 62.53661 62.52105
## [1025] 62.54490 62.53831 62.39966 62.54448 62.54078 62.54507 62.54311 62.54438
## [1033] 62.54486 62.54359 62.50913 62.54550 62.49454 62.54552 62.51823 62.54352
## [1041] 62.54311 62.54383 62.54304 62.40188 62.54535 62.54217 62.53983 62.54318
## [1049] 62.53576 62.54547 62.53165 62.53591 62.54401 62.54457 62.54225 62.54281
## [1057] 62.54311 62.54507 62.54553 62.54078 62.54289 62.54209 62.53793 62.52261
## [1065] 62.54191 62.54552 62.54526 62.54164 62.54504 62.53006 62.54005 62.54547
## [1073] 62.54281 62.52781 62.54047 62.53605 62.52895 62.51214 62.53332 62.54542
## [1081] 62.54474 62.54546 62.51676 62.54438 62.52702 62.54486 62.54352 62.54448
## [1089] 62.52858 62.52059 62.54457 62.54289 62.54412 62.53283 62.52012 62.54516
## [1097] 62.52914 62.54078 62.52105 62.54453 62.54448 62.52477 62.54497 62.54531
## [1105] 62.54547 62.54516 62.50092 62.52933 62.54371 62.54359 62.51267 62.54546
## [1113] 62.54553 62.54457 62.54531 62.52477 62.54466 62.53426 62.54457 62.54482
## [1121] 62.53316 62.53250 62.52988 62.54438 62.54553 62.54544 62.51919 62.53767
## [1129] 62.54542 62.53199 62.54127 62.54550 62.53983 62.54005 62.54553 62.53283
## [1137] 62.52682 62.54526 62.53517 62.54490 62.53078 62.53576 62.50830 62.50275
## [1145] 62.51966 62.54359 62.53078 62.52858 62.54137 62.52195 62.53715 62.53715
## [1153] 62.53095 62.54546 62.54377 62.54412 62.54164 62.54551 62.54058 62.52059
## [1161] 62.54521 62.51526 62.51267 62.53715 62.54553 62.51871 62.53348 62.52781
## [1169] 62.53024 62.53605 62.54318 62.54524 62.53533 62.48633 62.53767 62.52622
## [1177] 62.54325 62.54474 62.54266 62.52702 62.54377 62.53250 62.53006 62.53411
## [1185] 62.51701 62.54519 62.51798 62.53818 62.54371 62.53994 62.53915 62.53915
## [1193] 62.54418 62.53591 62.54127 62.53024 62.53078 62.52540 62.53818 62.51214
## [1201] 62.54539 62.51676 62.51345 62.54242 62.54516 62.53411 62.50543 62.54541
## [1209] 62.54513 62.53767 62.54478 62.53299 62.53411 62.54146 62.44826 62.53647
## [1217] 62.54504 62.54547 62.54352 62.54225 62.54234 62.54428 62.54209 62.54478
## [1225] 62.54304 62.54339 62.54137 62.53042 62.54544 62.54137 62.53781 62.52895
## [1233] 62.53364 62.54217 62.53891 62.54494 62.54537 62.53165 62.53938 62.53130
## [1241] 62.47128 62.54098 62.54486 62.42315 62.52519 62.54217 62.50335 62.54371
## [1249] 62.54191 62.53517 62.54433 62.54395 62.54470 62.51651 62.54274 62.54553
## [1257] 62.54500 62.50601 62.53266 62.53868 62.53994 62.53547 62.52392 62.51894
## [1265] 62.52240 62.54553 62.53113 62.54545 62.23850 62.54407 62.53793 62.53868
## [1273] 62.54486 62.39854 62.52414 62.54552 62.53591 62.54553 62.53818 62.53006
## [1281] 62.54258 62.52150 62.54318 62.53891 62.46311 62.54274 62.54266 62.50801
## [1289] 62.52702 62.54526 62.54258 62.54541 62.54200 62.38946 62.54258 62.54191
## [1297] 62.51823 62.53182 62.53904 62.53781 62.54448 62.54318 62.54359 62.54304
## [1305] 62.37290 62.52970 62.54526 62.53856 62.50062 62.53904 62.53927 62.53781
## [1313] 62.54462 62.54549 62.50913 62.53915 62.54258 62.47248 62.54365 62.54365
## [1321] 62.53831 62.54304 62.49937 62.53715 62.54474 62.54182 62.53591 62.53182
## [1329] 62.54524 62.49650 62.51372 62.54549 62.54418 62.52172 62.50886 62.53961
## [1337] 62.54533 62.53972 62.52195 62.54547 62.52284 62.52240 62.54078 62.54318
## [1345] 62.54289 62.51942 62.54553 62.53060 62.45186 62.53915 62.54553 62.53060
## [1353] 62.52662 62.54407 62.53283 62.54389 62.54549 62.54549 62.53661 62.51847
## [1361] 62.52349 62.54401 62.54551 62.54551 62.38830 62.54552 62.53688 62.54266
## [1369] 62.53487 62.54258 62.51475 62.54535 62.51214 62.53380 62.54234 62.53972
## [1377] 62.52819 62.54164 62.54542 62.52392 62.54016 62.54026 62.54457 62.54311
## [1385] 62.54542 62.53348 62.49650 62.53411 62.53024 62.49120 62.53380 62.54296
## [1393] 62.53633 62.52781 62.54423 62.53605 62.54359 62.54217 62.53487 62.50969
## [1401] 62.54504 62.54526 62.49488 62.51577 62.52895 62.51449 62.52560 62.54462
## [1409] 62.54108 62.51627 62.54552 62.54058 62.54339 62.54549 62.53006 62.54026
## [1417] 62.50154 62.53755 62.54529 62.54529 62.53755 62.51106 62.54026 62.53728
## [1425] 62.54318 62.53702 62.54535 62.53950 62.54311 62.54529 62.54553 62.53060
## [1433] 62.53619 62.54448 62.54173 62.51024 62.54407 62.54490 62.50969 62.53938
```

```
## [1441] 62.52581 62.54274 62.52371 62.50275 62.54545 62.53818 62.54318 62.51989
## [1449] 62.53182 62.54545 62.54453 62.53442 62.54407 62.54383 62.53702 62.54058
## [1457] 62.54401 62.53364 62.54304 62.54311 62.51989 62.52328 62.54478 62.51267
## [1465] 62.53078 62.54553 62.54553 62.54137 62.54529 62.54289 62.53950 62.53078
## [1473] 62.54088 62.53562 62.54553 62.54478 62.52988 62.43315 62.51372 62.54544
## [1481] 62.54504 62.54516 62.54005 62.54535 62.53806 62.54325 62.52150 62.54513
## [1489] 62.54423 62.53006 62.53880 62.54470 62.53806 62.50395 62.53619 62.54553
## [1497] 62.54296 62.54098 62.54037 62.53938 62.54553 62.54529 62.53831 62.51726
## [1505] 62.53250 62.53915 62.52819 62.54118 62.51267 62.54250 62.53994 62.54482
## [1513] 62.54332 62.53983 62.54225 62.54418 62.50215 62.52081 62.50601 62.50969
## [1521] 62.52933 62.52217 62.54547 62.54516 62.54005 62.52895 62.53793 62.53148
## [1529] 62.54173 62.52172 62.54546 62.54209 62.54296 62.54289 62.52349 62.54546
## [1537] 62.54352 62.54428 62.53113 62.54418 62.53972 62.52059 62.54428 62.53562
## [1545] 62.54494 62.54552 62.54225 62.53332 62.54433 62.54217 62.54389 62.53517
## [1553] 62.53688 62.54539 62.54225 62.53547 62.54478 62.54005 62.54531 62.53880
## [1561] 62.51627 62.54389 62.54016 62.54541 62.48983 62.54332 62.54526 62.54318
## [1569] 62.52195 62.53250 62.54552 62.53950 62.53426 62.54550 62.54521 62.53457
## [1577] 62.53042 62.52858 62.54550 62.54535 62.52722 62.54078 62.52838 62.54552
## [1585] 62.53547 62.53868 62.53961 62.53283 62.36802 62.53619 62.53078 62.54407
## [1593] 62.53950 62.54552 62.51989 62.53316 62.52970 62.54470 62.54026 62.51627
## [1601] 62.53348 62.53647 62.54553 62.54457 62.51397 62.51424 62.54552 62.53972
## [1609] 62.54531 62.54412 62.54545 62.54470 62.53426 62.54234 62.54428 62.53148
## [1617] 62.52284 62.54088 62.49937 62.53927 62.53880 62.53457 62.53576 62.53503
## [1625] 62.53767 62.53042 62.54242 62.53647 62.54504 62.54553 62.54500 62.53165
## [1633] 62.54526 62.49388 62.54545 62.53741 62.53113 62.54438 62.54258 62.53619
## [1641] 62.54504 62.54504 62.50913 62.52435 62.52195 62.52702 62.51676 62.53182
## [1649] 62.54016 62.54191 62.53856 62.53675 62.54529 62.52895 62.54553 62.54407
## [1657] 62.54541 62.54470 62.54234 62.53503 62.53547 62.54304 62.54217 62.53283
## [1665] 62.54377 62.52895 62.53818 62.54539 62.51267 62.54173 62.54173 62.54507
## [1673] 62.53994 62.54088 62.54225 62.45973 62.54412 62.52240 62.54098 62.53915
## [1681] 62.54258 62.54127 62.53728 62.54418 62.54537 62.52284 62.53380 62.48633
## [1689] 62.53938 62.54234 62.52722 62.54457 62.53728 62.54526 62.53148 62.54345
## [1697] 62.51726 62.53904 62.51160 62.53078 62.54068 62.52970 62.45627 62.54407
## [1705] 62.54519 62.54258 62.54234 62.54428 62.53250 62.54078 62.53042 62.54462
## [1713] 62.52305 62.54526 62.54544 62.54507 62.52838 62.54513 62.54098 62.54274
## [1721] 62.53619 62.51919 62.54541 62.54311 62.54553 62.52702 62.52560 62.53113
## [1729] 62.53411 62.49779 62.52059 62.54545 62.54448 62.53605 62.54478 62.52371
## [1737] 62.53675 62.54274 62.54098 62.53078 62.54155 62.54209 62.50513 62.54016
## [1745] 62.50687 62.50744 62.54395 62.53781 62.50687 62.54551 62.53831 62.54217
## [1753] 62.45186 62.54016 62.54377 62.53767 62.41059 62.54466 62.53348 62.52819
## [1761] 62.50630 62.51267 62.54545 62.54418 62.51577 62.53904 62.38714 62.54258
## [1769] 62.53661 62.53364 62.54500 62.49715 62.54428 62.54547 62.50687 62.52761
## [1777] 62.54234 62.54553 62.52560 62.54519 62.54423 62.54549 62.54448 62.53042
## [1785] 62.53818 62.54266 62.54478 62.54541 62.52172 62.54395 62.54497 62.51240
## [1793] 62.53216 62.51133 62.54443 62.54155 62.53831 62.54191 62.52622 62.53130
## [1801] 62.54453 62.54164 62.54448 62.54526 62.51750 62.52702 62.54457 62.54395
## [1809] 62.52682 62.53348 61.53849 62.51449 62.54504 62.53216 62.54550 62.52012
## [1817] 62.52761 62.54395 62.41166 62.53619 62.54173 62.54108 62.53891 62.51871
## [1825] 62.52581 62.54365 62.54542 62.54325 62.51775 62.52970 62.53927 62.52127
## [1833] 62.53411 62.54118 62.50659 62.54281 62.51106 62.53915 62.49779 62.52742
## [1841] 62.53233 62.54047 62.52392 62.54539 62.54037 62.51372 62.50335 62.51627
## [1849] 62.54497 62.54507 62.54005 62.52895 62.52662 62.53411 62.54551 62.53891
## [1857] 62.49842 62.49683 62.54047 62.52059 62.54535 62.53503 62.53517 62.53856
## [1865] 62.54507 62.54339 62.54242 62.54478 62.54118 62.48163 62.54478 62.54047
```

```
## [1873] 62.54438 62.51871 62.53702 62.52284 62.53675 62.54289 62.53024 62.52035
## [1881] 62.54539 62.51186 62.54289 62.54542 62.51293 62.54553 62.54453 62.54547
## [1889] 62.54037 62.49520 62.54521 62.53547 62.52988 62.54516 62.53927 62.54359
## [1897] 62.53781 62.52702 62.52761 62.53927 62.54544 62.54545 62.50000 62.54553
## [1905] 62.54412 62.54281 62.53562 62.54428 62.54395 62.52914 62.52414 62.54016
## [1913] 62.45540 62.54016 62.53042 62.51750 62.53972 62.53915 62.54234 62.54553
## [1921] 62.54155 62.54209 62.53880 62.53831 62.52217 62.52933 62.54304 62.54016
## [1929] 62.54549 62.50717 62.54545 62.54383 62.51133 62.54311 62.54345 62.54182
## [1937] 62.53741 62.54510 62.51267 62.54401 62.54550 62.54371 62.54296 62.54500
## [1945] 62.48914 62.54371 62.53266 62.52328 62.54516 62.54423 62.52951 62.53856
## [1953] 62.53741 62.54482 62.54533 62.53950 62.46311 62.54365 62.52895 62.54546
## [1961] 62.54118 62.48704 62.54544 62.45714 62.54546 62.54552 62.52435 62.53927
## [1969] 62.46143 62.52781 62.53006 62.54088 62.53233 62.54137 62.52435 62.49520
## [1977] 62.54108 62.54266 62.50543 62.53411 62.54500 62.54108 62.52150 62.52371
## [1985] 62.52601 62.53332 62.53113 62.45973 62.52150 62.51577 62.54365 62.54182
## [1993] 62.54191 62.54542 62.51750 62.53688 62.54433 62.52914 62.52477 62.54542
## [2001] 62.54550 62.54443 62.54078 62.52371 62.52081 62.50886 62.41273 62.54281
## [2009] 62.50454 62.52328 62.54412 62.54047 62.54155 62.54047 62.53633 62.53702
## [2017] 62.54068 62.54537 62.54345 62.52970 62.53755 62.51775 62.53619 62.54016
## [2025] 62.54553 62.54164 62.54539 62.51424 62.54490 62.54428 62.53994 62.54127
## [2033] 62.52456 62.54318 62.53487 62.53880 62.53619 62.53983 62.53042 62.53868
## [2041] 62.54352 62.52819 62.54553 62.53233 62.53348 62.53994 62.54345 62.53983
## [2049] 62.54478 62.49120 62.54526 62.53880 62.54519 62.54549 62.54544 62.54359
## [2057] 62.54407 62.54553 62.54533 62.54438 62.54531 62.54553 62.52540 62.51627
## [2065] 62.54173 62.52195 62.50395 62.47902 62.52601 62.54383 62.54466 62.53283
## [2073] 62.54553 62.50275 62.53266 62.54516 62.51798 62.53095 62.54339 62.54311
## [2081] 62.53113 62.54078 62.54242 62.54274 62.54521 62.46058 62.54521 62.52127
## [2089] 62.54497 62.54418 62.54553 62.54524 62.53994 62.54318 62.54173 62.53148
## [2097] 62.53332 62.53927 62.54296 62.52414 62.50773 62.53380 62.54377 62.54537
## [2105] 62.53915 62.53591 62.53411 62.54500 62.53442 62.54462 62.53533 62.52581
## [2113] 62.54507 62.53831 62.54078 62.42417 62.54521 62.54037 62.54474 62.54037
## [2121] 62.51651 62.54339 62.53078 62.49322 62.51160 62.53517 62.54058 62.54005
## [2129] 62.53457 62.53806 62.49322 62.51823 62.54296 62.54507 62.54088 62.54462
## [2137] 62.54318 62.54524 62.54311 62.53473 62.54191 62.54553 62.54482 62.54383
## [2145] 62.54490 62.52284 62.52414 62.54401 62.52105 62.54118 62.54539 62.51726
## [2153] 62.51676 62.54474 62.53675 62.54359 62.53348 62.50275 62.54137 62.54519
## [2161] 62.53348 62.51966 62.54549 62.53927 62.54118 62.54531 62.54494 62.54359
## [2169] 62.54457 62.54542 62.53619 62.54258 62.53216 62.54553 62.53487 62.54533
## [2177] 62.54457 62.54549 62.54457 62.54266 62.49779 62.54547 62.53266 62.54164
## [2185] 62.54482 62.53972 62.53503 62.53950 62.54553 62.54389 62.54352 62.54274
## [2193] 62.54371 62.54266 62.54542 62.54510 62.53316 62.54466 62.52284 62.54486
## [2201] 62.52858 62.54438 62.49779 62.54365 62.53266 62.53806 62.52371 62.54541
## [2209] 62.49905 62.53006 62.53348 62.52261 62.42315 62.54513 62.53661 62.53130
## [2217] 62.54325 62.54494 62.53781 62.53868 62.52988 62.52702 62.39176 62.54500
## [2225] 62.54289 62.53856 62.54478 62.52414 62.54552 62.54516 62.54438 62.51750
## [2233] 62.52800 62.51627 62.41590 62.54068 62.50030 62.54494 62.52722 62.54529
## [2241] 62.54234 62.54552 62.52560 62.53806 62.52761 62.49779 62.53182 62.54466
## [2249] 62.54318 62.54304 62.54457 62.44551 62.53348 62.53818 62.52012 62.53364
## [2257] 62.52240 62.53868 62.54108 62.52662 62.54332 62.54155 62.54524 62.54553
## [2265] 62.54542 62.53831 62.54088 62.53793 62.51871 62.53283 62.54553 62.50942
## [2273] 62.51293 62.54537 62.51500 62.54146 62.53130 62.54155 62.53868 62.54539
## [2281] 62.54466 62.53250 62.54016 62.54549 62.54137 62.53728 62.53781 62.54537
## [2289] 62.54281 62.53517 62.54443 62.54352 62.54047 62.54332 62.52858 62.48774
## [2297] 62.50773 62.54547 62.51798 62.54118 62.54173 62.54448 62.54462 62.53950
```

```
## [2305] 62.54037 62.53938 62.53767 62.54407 62.53395 62.53675 62.52622 62.51024
## [2313] 62.48273 62.54474 62.51847 62.54526 62.54544 62.54383 62.53755 62.51052
## [2321] 62.54526 62.54164 62.54164 62.54537 62.53983 62.54127 62.54482 62.50000
## [2329] 62.52662 62.50659 62.54553 62.53250 62.54553 62.53006 62.54539 62.54541
## [2337] 62.54513 62.53619 62.54549 62.54541 62.54552 62.54250 62.54553 62.54519
## [2345] 62.54433 62.53148 62.48914 62.50366 62.52988 62.52858 62.51052 62.54510
## [2353] 62.54108 62.52284 62.54108 62.53182 62.53562 62.54266 62.52581 62.54200
## [2361] 62.53950 62.53473 62.54552 62.54507 62.54546 62.54519 62.54200 62.53983
## [2369] 62.52519 62.54549 62.49905 62.53755 62.53266 62.53728 62.54332 62.54412
## [2377] 62.54383 62.54482 62.54127 62.51726 62.54474 62.54209 62.52895 62.53517
## [2385] 62.53165 62.54541 62.54443 62.54258 62.43217 62.53364 62.54545 62.53517
## [2393] 62.54462 62.53591 62.51871 62.54544 62.54482 62.54146 62.54137 62.53891
## [2401] 62.53233 62.54497 62.54516 62.54250 62.54549 62.54318 62.54553 62.54016
## [2409] 62.53891 62.54395 62.54551 62.54541 62.54546 62.54545 62.54553 62.52722
## [2417] 62.53619 62.54345 62.51919 62.54428 62.54217 62.53868 62.52127 62.53950
## [2425] 62.52435 62.53316 62.54339 62.29249 62.54250 62.54274 62.54510 62.52519
## [2433] 62.54200 62.54383 62.54550 62.54550 62.54428 62.53856 62.53316 62.54407
## [2441] 62.53927 62.52819 62.53042 62.54371 62.53961 62.52195 62.53741 62.54428
## [2449] 62.54521 62.54482 62.54423 62.52059 62.53591 62.53891 62.54146 62.53856
## [2457] 62.52035 62.51551 62.54462 62.54516 62.54547 62.54108 62.54164 62.51133
## [2465] 62.51627 62.54524 62.54225 62.54266 62.54395 62.52581 62.52477 62.54544
## [2473] 62.54542 62.49937 62.54005 62.54383 62.42315 62.54550 62.54418 62.53972
## [2481] 62.52328 62.54537 62.54462 62.53781 62.54318 62.54332 62.53831 62.53113
## [2489] 62.54549 62.54537 62.54383 62.52951 62.50687 62.53380 62.54549 62.54453
## [2497] 62.53562 62.53216 62.48346 62.32237 62.54552 62.54549 62.53728 62.54098
## [2505] 62.51214 62.53487 62.54533 62.53130 62.53517 62.53283 62.52914 62.54209
## [2513] 62.54352 62.51823 62.51871 62.52392 62.50601 62.54510 62.54535 62.51106
## [2521] 62.52349 62.53266 62.54047 62.54521 62.54453 62.54551 62.54412 62.54289
## [2529] 62.51676 62.54474 62.54339 62.53702 62.51726 62.54482 62.54164 62.54242
## [2537] 62.54443 62.52742 62.54191 62.46725 62.52761 62.54058 62.54524 62.54173
## [2545] 62.53856 62.51775 62.50215 62.54490 62.50062 62.47674 62.54383 62.54549
## [2553] 62.44916 62.52328 62.54482 62.54345 62.54318 62.54504 62.54078 62.50454
## [2561] 62.51293 62.52970 62.54462 62.54526 62.53647 62.52172 62.53938 62.52933
## [2569] 62.53199 62.54005 62.54526 62.53755 62.54311 62.53562 62.53880 62.54457
## [2577] 62.53831 62.53927 62.54078 62.54401 62.54395 62.53994 62.53182 62.52414
## [2585] 62.54266 62.54118 62.54078 62.54266 62.52217 62.54037 62.53562 62.53024
## [2593] 62.54182 62.54191 62.54217 62.54533 62.54078 62.53442 62.54490 62.54482
## [2601] 62.52371 62.53517 62.54490 62.54164 62.54371 62.54457 62.53619 62.54383
## [2609] 62.53950 62.53950 62.50997 62.54225 62.54438 62.53547 62.54325 62.54551
## [2617] 62.51847 62.54448 62.53473 62.54553 62.54339 62.54412 62.54325 62.52819
## [2625] 62.54289 62.54553 62.54383 62.54478 62.50092 62.53688 62.53216 62.53661
## [2633] 62.54482 62.54395 62.54552 62.54250 62.53503 62.54296 62.54542 62.54296
## [2641] 62.53473 62.54242 62.54553 62.53473 62.48774 62.51798 62.53702 62.53348
## [2649] 62.54395 62.54553 62.54281 62.53042 62.54371 62.52435 62.54470 62.54173
## [2657] 62.53113 62.52435 62.51701 62.53395 62.52414 62.54521 62.54547 62.52172
## [2665] 62.54088 62.51701 62.54371 62.54521 62.49187 62.54470 62.53647 62.54535
## [2673] 62.53806 62.54504 62.54242 62.46478 62.53755 62.52305 62.54531 62.53233
## [2681] 62.52560 62.54546 62.54494 62.53972 62.53113 62.54377 62.54225 62.54108
## [2689] 62.53619 62.54438 62.54490 62.54507 62.52328 62.54428 62.51966 62.52988
## [2697] 62.54325 62.52682 62.52305 62.54544 62.54552 62.51847 62.54365 62.53843
## [2705] 62.54155 62.54407 62.54182 62.53647 62.36306 62.53165 62.49715 62.52150
## [2713] 62.53856 62.54250 62.51293 62.54552 62.54383 62.53199 62.53517 62.54098
## [2721] 62.53806 62.53856 62.54553 62.54438 62.54549 62.54209 62.52933 62.52150
## [2729] 62.53818 62.54058 62.54553 62.54504 62.54521 62.53042 62.51919 62.54535
```

```
## [2737] 62.51133 62.54510 62.54531 62.51966 62.54339 62.53024 62.54542 62.54552
## [2745] 62.53806 62.52435 62.54553 62.54552 62.54533 62.54553 62.54552 62.51424
## [2753] 62.53961 62.54345 62.54470 62.54037 62.52414 62.50969 62.53332 62.54553
## [2761] 62.54547 62.50543 62.53755 62.54448 62.53182 62.53831 62.53517 62.54529
## [2769] 62.50030 62.53364 62.51847 62.53487 62.54281 62.53006 62.54146 62.53647
## [2777] 62.52105 62.54545 62.54537 62.54553 62.54304 62.54438 62.53755 62.54516
## [2785] 62.54016 62.53576 62.54497 62.54500 62.53411 62.54365 62.54544 62.54486
## [2793] 62.54521 62.53831 62.54209 62.54539 62.53619 62.54531 62.54513 62.54466
## [2801] 62.53562 62.53915 62.54550 62.54507 62.53380 62.54448 62.54524 62.54047
## [2809] 62.54462 62.54443 62.53831 62.52970 62.54389 62.54037 62.54395 62.50942
## [2817] 62.54553 62.54553 62.54258 62.52819 62.54470 62.53591 62.53042 62.54433
## [2825] 62.54225 62.54482 62.52933 62.54137 62.53728 62.53675 62.54407 62.53806
## [2833] 62.54250 62.54423 62.49715 62.50969 62.52876 62.53182 62.53442 62.53591
## [2841] 62.52261 62.53233 62.54339 62.54371 62.54553 62.54549 62.54296 62.54146
## [2849] 62.54389 62.53113 62.53904 62.54470 62.53994 62.45627 62.54507 62.54137
## [2857] 62.54545 62.54026 62.50513 62.54352 62.53983 62.52819 62.54494 62.53473
## [2865] 62.54173 62.52722 62.52581 62.54519 62.53904 62.53868 62.54470 62.49779
## [2873] 62.54529 62.44643 62.53728 62.52702 62.54108 62.54098 62.42417 62.54332
## [2881] 62.54474 62.53688 62.53591 62.53891 62.54383 62.54474 62.52933 62.54191
## [2889] 62.54325 62.54533 62.54516 62.53781 62.54521 62.54182 62.54047 62.54258
## [2897] 62.50659 62.53619 62.53380 62.54383 62.54529 62.54552 62.53316 62.51052
## [2905] 62.53755 62.54289 62.54389 62.52392 62.54490 62.54078 62.54401 62.54339
## [2913] 62.54365 62.53983 62.53927 62.54549 62.51676 62.54352 62.54389 62.53880
## [2921] 62.54524 62.54418 62.54365 62.54318 62.53348 62.54541 62.54553 62.54470
## [2929] 62.54325 62.53983 62.51526 62.54325 62.53728 62.54423 62.54553 62.54521
## [2937] 62.51919 62.53487 62.54026 62.51871 62.46478 62.51942 62.48015 62.54401
## [2945] 62.54541 62.52662 62.52105 62.53395 62.52012 62.54182 62.53042 62.54068
## [2953] 62.53868 62.54524 62.49086 62.54407 62.49554 62.54332 62.52414 62.50185
## [2961] 62.53927 62.52035 62.51500 62.54544 62.54182 62.54332 62.54164 62.54542
## [2969] 62.45627 62.51526 62.53487 62.54325 62.53961 62.50543 62.50857 62.51475
## [2977] 62.54553 62.54311 62.51160 62.49520 62.53927 62.54453 62.54553 62.53972
## [2985] 62.53547 62.52127 62.54504 62.54423 62.54108 62.54550 62.54383 62.52105
## [2993] 62.53675 62.54546 62.53442 62.54535 62.54541 62.54513 62.50030 62.54494
## [3001] 62.54146 62.54526 62.54088 62.54539 62.54401 62.54058 62.54510 62.52800
## [3009] 62.54407 62.53675 62.49842 62.54549 62.54533 62.54209 62.53793 62.54510
## [3017] 62.54088 62.53741 62.54137 62.54547 62.54552 62.54423 62.54217 62.54552
## [3025] 62.54352 62.50000 62.54418 62.53781 62.54209 62.53961 62.53411 62.54395
## [3033] 62.53380 62.54088 62.53831 62.52284 62.53927 62.54412 62.54443 62.52876
## [3041] 62.54289 62.53576 62.54209 62.54234 62.54225 62.54383 62.53781 62.54058
## [3049] 62.53006 62.54507 62.53994 62.54345 62.53533 62.50942 62.54494 62.54345
## [3057] 62.52498 62.54553 62.53675 62.54281 62.53266 62.54535 62.54526 62.53915
## [3065] 62.54457 62.52498 62.53113 62.54457 62.54550 62.53348 62.43217 62.49937
## [3073] 62.54553 62.54462 62.53426 62.54242 62.54552 62.53078 62.51424 62.53266
## [3081] 62.54428 62.54164 62.54516 62.54474 62.54037 62.54474 62.54531 62.54234
## [3089] 62.54553 62.53728 62.54438 62.54547 62.54118 62.54058 62.54281 62.54553
## [3097] 62.53233 62.54526 62.54209 62.54433 62.54395 62.54466 62.54457 62.52742
## [3105] 62.54359 62.54266 62.54547 62.54155 62.54274 62.54296 62.54519 62.53547
## [3113] 62.52895 62.53950 62.53927 62.53647 62.54553 62.53216 62.54546 62.54521
## [3121] 62.49715 62.54553 62.53818 62.53095 62.54098 62.53755 62.54549 62.52895
## [3129] 62.53755 62.54304 62.52914 62.49187 62.54418 62.54242 62.54296 62.54225
## [3137] 62.54047 62.54448 62.41695 62.51024 62.53562 62.53006 62.50454 62.54545
## [3145] 62.54266 62.53702 62.54438 62.51397 62.54482 62.54371 62.53994 62.54524
## [3153] 62.50484 62.53411 62.54371 62.54118 62.52261 62.50124 62.54448 62.54478
## [3161] 62.53633 62.54550 62.54457 62.53411 62.52540 62.54304 62.53983 62.54037
```

```
## [3169] 62.53216 62.52150 62.54521 62.44734 62.54155 62.51372 62.54058 62.52414
## [3177] 62.53983 62.54359 62.54507 62.54553 62.53348 62.54466 62.54550 62.54242
## [3185] 62.53411 62.51627 62.54266 62.54474 62.54500 62.54533 62.53006 62.51345
## [3193] 62.54005 62.52349 62.54433 62.52682 62.49187 62.54088 62.54482 62.54443
## [3201] 62.54550 62.54448 62.52012 62.54098 62.52702 62.54551 62.53348 62.51240
## [3209] 62.54423 62.51500 62.53266 62.54457 62.52371 62.50601 62.54466 62.54258
## [3217] 62.54423 62.34257 62.54533 62.54088 62.54155 62.54068 62.52035 62.47207
## [3225] 62.54428 62.54182 62.54412 62.54497 62.53547 62.54544 62.53250 62.54448
## [3233] 62.54118 62.49715 62.42619 62.54482 62.54234 62.54546 62.50543 62.54325
## [3241] 62.54200 62.54521 62.54438 62.52581 62.54371 62.53533 62.51775 62.49356
## [3249] 62.54274 62.54545 62.54401 62.54274 62.50185 62.54539 62.50543 62.54412
## [3257] 62.54418 62.52081 62.54553 62.54412 62.53915 62.54482 62.54497 62.54016
## [3265] 62.54289 62.54250 62.54365 62.53675 62.54539 62.54037 62.54289 62.51267
## [3273] 62.54474 62.54304 62.52371 62.52951 62.51186 62.52951 62.50942 62.52914
## [3281] 62.54494 62.53591 62.52581 62.52761 62.52371 62.54345 62.54519 62.53364
## [3289] 62.53042 62.53216 62.54549 62.54026 62.54058 62.50092 62.53332 62.52800
## [3297] 62.54250 62.54037 62.52105 62.53332 62.54550 62.54470 62.54016 62.52012
## [3305] 62.54365 62.52702 62.54401 62.54078 62.53487 62.54389 62.52498 62.52414
## [3313] 62.54088 62.54242 62.42213 62.54209 62.54155 62.51701 62.52970 62.54352
## [3321] 62.54304 62.50717 62.52519 62.54453 62.54428 62.52876 62.53688 62.40299
## [3329] 62.53442 62.54037 62.54395 62.52035 62.54529 62.54412 62.53868 62.54546
## [3337] 62.54474 62.53675 62.53078 62.52328 62.54412 62.52284 62.50969 62.53316
## [3345] 62.54552 62.51052 62.54486 62.54553 62.54552 62.53793 62.50857 62.54507
## [3353] 62.54462 62.54325 62.52742 62.54217 62.54401 62.53818 62.53411 62.54521
## [3361] 62.54552 62.53283 62.54016 62.54088 62.54108 62.53904 62.53250 62.53316
## [3369] 62.52240 62.54474 62.52970 62.53533 62.54098 62.54541 62.52722 62.53562
## [3377] 62.54533 62.54058 62.54474 62.52761 62.51526 62.53927 62.52560 62.53442
## [3385] 62.50997 62.54258 62.51345 62.53411 62.53250 62.54078 62.54486 62.52819
## [3393] 62.54550 62.54047 62.54453 62.53411 62.52540 62.53380 62.54209 62.53715
## [3401] 62.54552 62.53547 62.48704 62.53868 62.53316 62.53755 62.52305 62.53533
## [3409] 62.51320 62.53831 62.53702 62.53042 62.54553 62.53675 62.54418 62.53972
## [3417] 62.54371 62.54478 62.54155 62.54058 62.52933 62.52761 62.54448 62.54155
## [3425] 62.51240 62.53793 62.53605 62.54078 62.52498 62.51372 62.54552 62.53675
## [3433] 62.52519 62.54339 62.51847 62.54339 62.54037 62.54242 62.54551 62.52819
## [3441] 62.54345 62.54539 62.53060 62.54209 62.52261 62.31683 62.54497 62.52761
## [3449] 62.53395 62.51372 62.52761 62.54412 62.53661 62.53517 62.54552 62.53983
## [3457] 62.50773 62.54443 62.52951 62.52876 62.54098 62.54108 62.54182 62.54345
## [3465] 62.52662 62.54108 62.52059 62.51847 62.54474 62.53702 62.54521 62.53904
## [3473] 62.53547 62.54549 62.54546 62.54352 62.54258 62.53576 62.53283 62.51966
## [3481] 62.54401 62.54448 62.49120 62.54423 62.51676 62.54289 62.54553 62.45097
## [3489] 62.52059 62.53950 62.52105 62.51024 62.53148 62.54507 62.54531 62.52456
## [3497] 62.53042 62.54365 62.49120 62.53095 62.51942 62.54078 62.52105 62.53364
## [3505] 62.52540 62.53283 62.54553 62.52456 62.50092 62.53060 62.54311 62.54418
## [3513] 62.54304 62.52195 62.53380 62.53806 62.51160 62.54541 62.53411 62.54098
## [3521] 62.53024 62.54127 62.54164 62.53619 62.54137 62.54507 62.54545 62.51577
## [3529] 62.41273 62.54500 62.53647 62.54289 62.54533 62.49779 62.53868 62.54318
## [3537] 62.54401 62.52261 62.47286 62.51577 62.52682 62.54443 62.53675 62.54339
## [3545] 62.54191 62.53364 62.52970 62.54510 62.54541 62.40736 62.54552 62.49715
## [3553] 62.53619 62.54549 62.53994 62.52195 62.52195 62.54553 62.54490 62.49747
## [3561] 62.53533 62.54058 62.53182 62.51989 62.54407 62.49715 62.54504 62.53562
## [3569] 62.51894 62.52933 62.50744 62.54359 62.54497 62.54462 62.46806 62.54164
## [3577] 62.52838 62.54304 62.54549 62.51372 62.53715 62.54234 62.54533 62.49222
## [3585] 62.54516 62.53972 62.53165 62.51424 62.54438 62.53130 62.54526 62.53950
## [3593] 62.51701 62.54526 62.54547 62.39966 62.54553 62.54242 62.54209 62.53605
```

```
## [3601] 62.54242 62.51726 62.52838 62.54217 62.54553 62.54553 62.54549 62.54225
## [3609] 62.54541 62.54395 62.54544 62.54037 62.54359 62.53395 62.54423 62.54395
## [3617] 62.52540 62.51871 62.53380 62.54058 62.54318 62.52858 62.52858 62.53605
## [3625] 62.54474 62.54098 62.54524 62.54462 62.54438 62.54486 62.52540 62.53880
## [3633] 62.54494 62.50942 62.51526 62.52059 62.49811 62.53233 62.54470 62.53843
## [3641] 62.51214 62.52682 62.54504 62.54500 62.53868 62.51775 62.54339 62.54108
## [3649] 62.51798 62.54519 62.48126 62.53576 62.54546 62.54504 62.51775 62.54474
## [3657] 62.51919 62.53972 62.54526 62.52761 62.53688 62.54281 62.53426 62.54486
## [3665] 62.54553 62.53647 62.54504 62.51775 62.52349 62.54457 62.54098 62.53781
## [3673] 62.54332 62.53533 62.54371 62.54486 62.54345 62.54127 62.54016 62.51676
## [3681] 62.54453 62.54191 62.54490 62.52642 62.44178 62.54550 62.53972 62.54088
## [3689] 62.51214 62.54470 62.54443 62.54428 62.52328 62.53024 62.54448 62.51871
## [3697] 62.53647 62.52970 62.54462 62.54401 62.54537 62.54266 62.54217 62.54304
## [3705] 62.54497 62.54389 62.52081 62.52819 62.54541 62.54318 62.54359 62.54325
## [3713] 62.54332 62.54418 62.52172 62.54332 62.53983 62.53517 62.54225 62.54549
## [3721] 62.53927 62.54155 62.54533 62.54553 62.54504 62.53880 62.53938 62.53904
## [3729] 62.54474 62.53503 62.54474 62.54529 62.53675 62.54516 62.53741 62.54535
## [3737] 62.54318 62.54552 62.53793 62.44458 62.54462 62.50366 62.54412 62.53647
## [3745] 62.52540 62.54510 62.51823 62.53395 62.54304 62.54345 62.54552 62.54047
## [3753] 62.52800 62.54549 62.52895 62.54552 62.54521 62.54521 62.54098 62.54453
## [3761] 62.53868 62.54482 62.53148 62.54395 62.52581 62.54539 62.53972 62.37411
## [3769] 62.54182 62.53619 62.54209 62.54191 62.54535 62.53442 62.53793 62.54200
## [3777] 62.54539 62.54242 62.54191 62.54504 62.53938 62.52305 62.53755 62.54005
## [3785] 62.52328 62.51424 62.53113 62.52933 62.48844 62.54289 62.54281 62.53767
## [3793] 62.49842 62.53831 62.53411 62.54225 62.53806 62.54242 62.53348 62.48199
## [3801] 62.27608 62.53130 62.53348 62.54547 62.54443 62.54258 62.54521 62.54546
## [3809] 62.49937 62.53688 62.54037 62.53983 62.54490 62.54544 62.53148 62.52127
## [3817] 62.54047 62.53880 62.54519 62.54553 61.69581 62.52761 62.54526 62.50154
## [3825] 62.54078 62.54553 62.53793 62.53950 62.54551 62.53006 62.50275 62.52195
## [3833] 62.54497 62.53562 62.44272 62.53633 62.38946 62.54535 62.52819 62.54383
## [3841] 62.54466 62.54266 62.50484 62.50000 62.51577 62.54407 62.54225 62.54448
## [3849] 62.53755 62.54234 62.53233 62.54401 62.53165 62.54521 62.53148 62.52970
## [3857] 62.53927 62.54289 62.54545 62.54482 62.54209 62.53793 62.51726 62.54531
## [3865] 62.51160 62.53473 62.53380 62.54549 62.53395 62.54547 62.53950 62.54371
## [3873] 62.54490 62.53411 62.54494 62.53927 62.52914 62.54058 62.54258 62.54553
## [3881] 62.50969 62.54332 62.54497 62.54137 62.53148 62.52742 62.54209 62.54250
## [3889] 62.53078 62.53395 62.48704 62.54521 62.53818 62.54281 62.54553 62.48774
## [3897] 62.51750 62.52392 62.48273 62.54281 62.52662 62.54516 62.54407 62.54443
## [3905] 62.53843 62.54118 62.53843 62.54551 62.54325 62.54539 62.53216 62.54026
## [3913] 62.54359 62.54365 62.54466 62.54371 62.54550 62.54289 62.51024 62.52435
## [3921] 62.54552 62.54462 62.52217 62.50969 62.54407 62.54026 62.54395 62.54078
## [3929] 62.54478 62.53675 62.53831 62.54513 62.54507 62.54553 62.53364 62.51214
## [3937] 62.53868 62.53182 62.54453 62.51871 62.53755 62.52601 62.51500 62.54200
## [3945] 62.54423 62.54551 62.54521 62.51798 62.54191 62.51919 62.54304 62.54551
## [3953] 62.54407 62.54383 62.48774 62.54401 62.53113 62.54546 62.36555 62.52305
## [3961] 62.54016 62.52722 62.52702 62.53605 62.54428 62.53619 62.54552 62.50913
## [3969] 62.53728 62.54127 62.53348 62.53042 62.54553 62.54553 62.52477 62.54296
## [3977] 62.52800 62.54289 62.54526 62.54258 62.51475 62.54526 62.54552 62.52914
## [3985] 62.54234 62.53755 62.51052 62.53950 62.48418 62.53182 62.51424 62.53619
## [3993] 62.54164 62.54234 62.51186 62.54078 62.54539 62.54098 62.53950 62.53831
## [4001] 62.53006 62.54209 62.54395 62.54098 62.52622 62.52150 62.54098 62.51133
## [4009] 62.54539 62.53130 62.54311 62.53487 62.47048 62.54325 62.39176 62.53619
## [4017] 62.52933 62.50246 62.54549 62.53113 62.52819 62.53927 62.54258 62.54519
## [4025] 62.54127 62.54058 62.54531 62.53380 62.54318 62.53503 62.52456 62.53533
```

```
## [4033] 62.54486 62.54533 62.54553 62.53605 62.54552 62.54217 62.53781 62.54423
## [4041] 62.54553 62.52498 62.54078 62.53380 62.54539 62.54521 62.54553 62.54155
## [4049] 62.54242 62.54541 62.54529 62.53576 62.54535 62.50513 62.53316 62.54551
## [4057] 62.54332 62.54443 62.54500 62.53806 62.54352 62.46058 62.50275 62.53503
## [4065] 62.53605 62.54118 62.53856 62.52284 62.50630 62.54296 62.53728 62.51966
## [4073] 62.52240 62.54474 62.54339 62.53927 62.54412 62.53675 62.54553 62.54510
## [4081] 62.37411 62.54549 62.53950 62.54547 62.52682 62.54127 62.54127 62.54553
## [4089] 62.54325 62.54553 62.54318 62.54371 62.52702 62.54016 62.54225 62.52284
## [4097] 62.54486 62.51106 62.54266 62.54510 62.54513 62.53950 62.53457 62.54209
## [4105] 62.54516 62.51894 62.52105 62.52414 62.53972 62.54537 62.49842 62.54504
## [4113] 62.52970 62.54016 62.53380 62.54258 62.52012 62.53868 62.54490 62.54395
## [4121] 62.50030 62.54146 62.53060 62.51240 62.53633 62.54395 62.54155 62.54234
## [4129] 62.54550 62.53042 62.54088 62.49875 62.54088 62.54516 62.53661 62.54519
## [4137] 62.54250 62.47128 62.54389 62.53950 62.51750 62.53961 62.51106 62.54428
## [4145] 62.52498 62.53818 62.53904 62.54250 62.54383 62.53715 62.52328 62.54500
## [4153] 62.54217 62.53442 62.53904 62.54412 62.54418 62.52933 62.54547 62.51701
## [4161] 62.52328 62.49187 62.54289 62.54547 62.51942 62.53818 62.54352 62.53741
## [4169] 62.54234 62.54486 62.54535 62.53042 62.54541 62.53950 62.47977 62.54510
## [4177] 62.54274 62.54359 62.54127 62.54526 62.54553 62.54164 62.52722 62.54535
## [4185] 62.51775 62.54490 62.53755 62.54547 62.53411 62.54524 62.52328 62.48844
## [4193] 62.52035 62.54401 62.53880 62.54504 62.54438 62.52622 62.54395 62.52819
## [4201] 62.54258 62.53024 62.54507 62.51106 62.54258 62.54457 62.54549 62.52477
## [4209] 62.53517 62.53473 62.54412 62.54474 62.54182 62.54155 62.49051 62.54318
## [4217] 62.53591 62.54118 62.54281 62.54164 62.54457 62.52800 62.52081 62.52819
## [4225] 62.54524 62.53856 62.53915 62.52601 62.54217 62.53891 62.49255 62.54217
## [4233] 62.54428 62.54504 62.54281 62.50154 62.54068 62.54428 62.47597 62.53364
## [4241] 62.53767 62.54541 62.51320 62.53633 62.53972 62.54164 62.51798 62.50366
## [4249] 62.53793 62.54088 62.54352 62.51106 62.44826 62.53487 62.52081 62.54516
## [4257] 62.54296 62.50717 62.54549 62.52702 62.54242 62.54418 62.52081 62.52519
## [4265] 62.54005 62.54311 62.54258 62.52988 62.54519 62.54482 62.54281 62.54466
## [4273] 62.54497 62.53233 62.53457 62.53060 62.52540 62.53503 62.54438 62.53741
## [4281] 62.54250 62.54037 62.54047 62.54546 62.54553 62.52970 62.54510 62.54242
## [4289] 62.54478 62.53831 62.54448 62.52105 62.47442 62.54537 62.52858 62.49422
## [4297] 62.54234 62.54365 62.54412 62.53042 62.54146 62.54494 62.54507 62.53938
## [4305] 62.53891 62.46643 62.53781 62.54494 62.53487 62.54047 62.54078 62.54304
## [4313] 62.54318 62.54550 62.51775 62.54529 62.53411 62.54191 62.54258 62.53891
## [4321] 62.54535 62.54401 62.54365 62.54519 62.54428 62.54497 62.53702 62.54533
## [4329] 62.54395 62.54225 62.53950 62.54519 62.53024 62.54318 62.53661 62.54521
## [4337] 62.54553 62.54529 62.53938 62.54225 62.53199 62.54155 62.53843 62.54524
## [4345] 62.54553 62.54552 62.53972 62.52933 62.53283 62.53961 62.54544 62.52838
## [4353] 62.54433 62.54443 62.53938 62.51919 62.52261 62.53619 62.54544 62.54482
## [4361] 62.51475 62.53904 62.54088 62.50830 62.54545 62.53728 62.52970 62.53619
## [4369] 62.53728 62.54457 62.53806 62.53411 62.54507 62.50366 62.54118 62.51775
## [4377] 62.54553 62.54418 62.43412 62.54550 62.52195 62.53533 62.51214 62.54510
## [4385] 62.53042 62.53880 62.54507 62.51424 62.54453 62.54088 62.51577 62.52662
## [4393] 62.50246 62.54311 62.52876 62.54510 62.54242 62.52933 62.53473 62.52682
## [4401] 62.51577 62.54549 62.54524 62.53961 62.52622 62.53442 62.54200 62.54497
## [4409] 62.54531 62.54529 62.54549 62.54078 62.54058 62.53927 62.54250 62.54164
## [4417] 62.54026 62.54507 62.53165 62.54537 62.52435 62.54462 62.53165 62.52414
## [4425] 62.54318 62.54296 62.54359 62.54443 62.54535 62.52328 62.52933 62.54457
## [4433] 62.54507 62.54500 62.54524 62.54225 62.32649 62.54118 62.52477 62.52328
## [4441] 62.51627 62.54546 62.51526 62.53938 62.54521 62.46887 62.52456 62.54137
## [4449] 62.54274 62.54478 62.54395 62.54289 62.54217 62.48490 62.54173 62.54296
## [4457] 62.47048 62.53891 62.52150 62.49322 62.53266 62.52662 62.53806 62.52895
```

```
## [4465] 62.53426 62.54504 62.50744 62.54428 62.53972 62.52328 62.52059 62.54443
## [4473] 62.54524 62.49779 62.54016 62.53395 62.54250 62.52059 62.54318 62.54529
## [4481] 62.54325 62.54304 62.53927 62.53927 62.53503 62.54470 62.52800 62.50572
## [4489] 62.53927 62.53633 62.54296 62.54482 62.48051 62.54519 62.53818 62.52195
## [4497] 62.51160 62.54146 62.52328 62.54258 62.54078 62.54164 62.54209 62.53927
## [4505] 62.40952 62.54553 62.52722 62.53060 62.52742 62.54553 62.54457 62.53078
## [4513] 62.53961 62.50630 62.52876 62.52305 62.54553 62.54478 62.54026 62.54325
## [4521] 62.51651 62.54209 62.54365 62.54482 62.49683 62.53927 62.51267 62.50425
## [4529] 62.52435 62.52519 62.54311 62.54486 62.54542 62.53688 62.53503 62.54542
## [4537] 62.54545 62.54549 62.52988 62.53503 62.52819 62.51160 62.54068 62.53316
## [4545] 62.54549 62.51345 62.54225 62.38128 62.54553 62.54047 62.52519 62.54026
## [4553] 62.53675 62.53024 62.53927 62.52305 62.53316 62.54433 62.54539 62.53283
## [4561] 62.50886 62.54552 62.54395 62.51160 62.54490 62.54553 62.52800 62.53442
## [4569] 62.54519 62.54537 62.54428 62.48273 62.54395 62.53880 62.54549 62.53078
## [4577] 62.54173 62.54311 62.54494 62.54225 62.54438 62.53868 62.53411 62.50997
## [4585] 62.51919 62.54438 62.54137 62.54448 62.54037 62.54519 62.53972 62.52240
## [4593] 62.54325 62.54137 62.54504 62.53793 62.54549 62.51798 62.54443 62.54058
## [4601] 62.52722 62.54529 62.54155 62.45452 62.53605 62.54544 62.54553 62.53591
## [4609] 62.54546 62.48983 62.54490 62.54547 62.52933 62.54423 62.51871 62.54155
## [4617] 62.54078 62.54423 62.54401 62.54529 62.54513 62.54531 62.50830 62.54418
## [4625] 62.54209 62.51627 62.53006 62.52371 62.54553 62.54553 62.54173 62.54289
## [4633] 62.54537 62.53283 62.54474 62.54407 62.54418 62.54209 62.54242 62.53843
## [4641] 62.50454 62.54155 62.52622 62.54549 62.53591 62.54531 62.54191 62.54289
## [4649] 62.51293 62.47048 62.53927 62.53755 62.54537 62.54296 62.53165 62.54526
## [4657] 62.54118 62.52988 62.54258 62.54118 62.54118 62.54553 62.51475 62.53517
## [4665] 62.51894 62.54118 62.54332 62.52838 62.53831 62.53250 62.50773 62.51627
## [4673] 62.54539 62.51500 62.52951 62.54098 62.54474 62.54164 62.53503 62.53818
## [4681] 62.54513 62.51475 62.53250 62.54462 62.52392 62.54118 62.51775 62.54545
## [4689] 62.51500 62.51823 62.54547 62.54155 62.53473 62.47597 62.52328 62.51894
## [4697] 62.54486 62.54332 62.54407
##
## $call
## jackknife(x = bmi, theta = bias_var)
```

B O 0 T S T R A P P I N G

Statistical Rigor: Estimation of Sampling Distribution: It allows for the empirical estimation of the sampling distribution of almost any statistic, providing a way to estimate standard errors, confidence intervals, and other distributional characteristics. Bias and Variance Estimation: Bootstrapping can help in estimating the bias and variance of a statistic, providing more information than just a point estimate.

Flexibility: Model Validation: It can be used for model validation and for diagnosing the fit of a model. Hypothesis Testing: Bootstrapping can be adapted for hypothesis testing, especially for tests where the analytic form is complex or not available. Small Sample Performance Small Sample Size: Bootstrapping can be particularly useful when your sample size is small, making it difficult to make reliable parametric assumptions.

Limitations and Caveats: It's important to note that while bootstrapping is powerful, it is not without its limitations. For example:

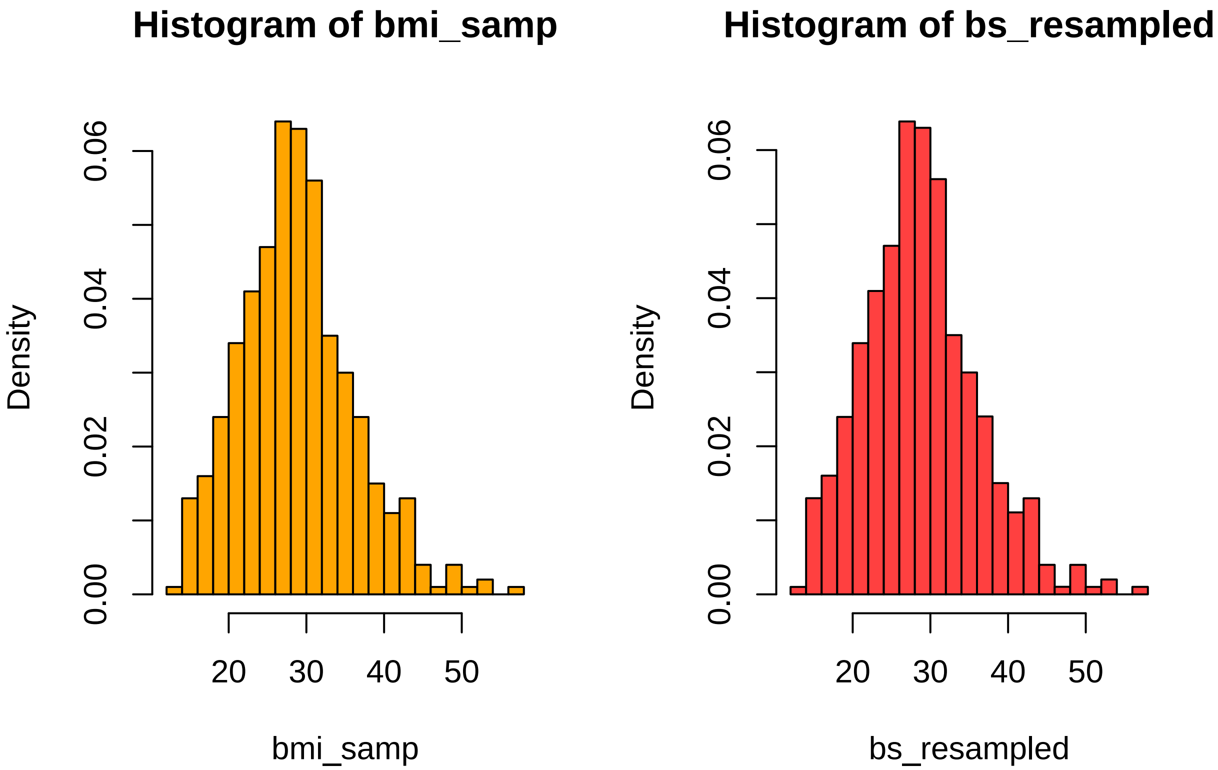
It may not perform well for data that has a complex dependence structure (e.g., time series data). Bootstrapping provides an approximation, and the quality of this approximation depends on the actual sample size and the underlying distribution.

```
#Bootstrapping
sample_size<- 500
samples<- 10000
```

```r
bmi_samp<- sample(bmi, sample_size, replace = FALSE)
print(paste("Mean of the sampled true population: ",mean(bmi_samp)))
```

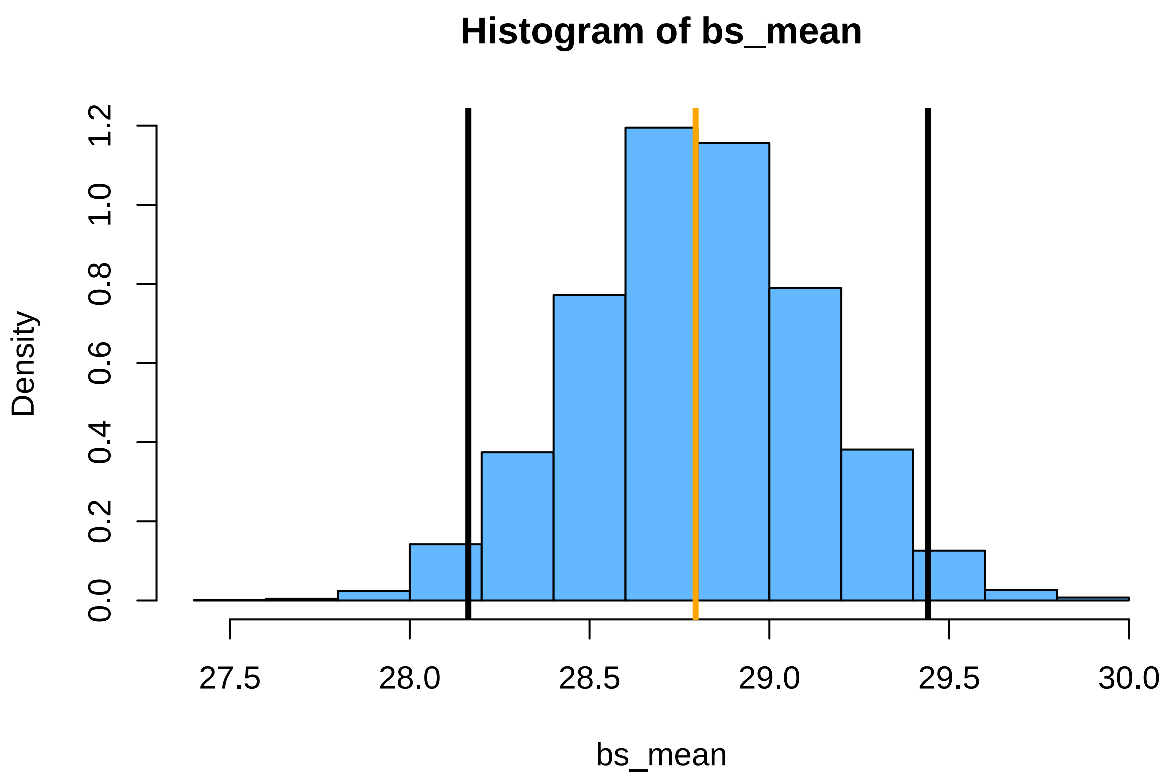
```
## [1] "Mean of the sampled true population:  28.7948"
```

```r
bs_resampled<-matrix(sample(bmi_samp, sample_size*samples,replace = TRUE),nrow=samples, ncol=sample_siz
par(mfcol=c(1,2))
hist(bmi_samp,probability=TRUE,breaks=20, col="orange")
hist(bs_resampled,probability=TRUE, breaks=20, col="brown1")
```



Here, both distribution is looking identical. And it was expected. Cause we were trying to emulate the left one ((assumed)true distribution(though it was sampled from total population, just assume that it is the true distribution)) by sampling(bootsrapping) many times and got that on the right histogram.

```r
#calculating mean for each row
bs_mean<-rowMeans(bs_resampled)

hist(bs_mean,probability = TRUE, breaks=10, col="steelblue1")
abline(v=mean(bmi_samp),col="orange",lwd=3)
abline(v=quantile(bs_mean,c(.025,0.975)), lwd=3)# 95% confidence interval
```




```
print(paste("Std. error: ", round(sd(bs_mean),3)))
```

```
## [1] "Std. error:  0.324"
```

The standard deviation of this bootstrapped distribution gives us an idea of how much mean(statistic) would vary if we were to draw many different samples from the population, thereby acting as an estimate of the standard error.

```
# getting confidence interval in every possible way:
# say, we want to see the prevalence of hypertension in our population
n=200
hyper<-sample(data$hypertension,n)
hyper_prop<-length(hyper[hyper==1])
print(binom.confint(hyper_prop, n=n, conf.level = 0.95, methods = "all"))
```

```
##           method  x   n       mean      lower      upper
## 1  agresti-coull 13 200 0.06500000 0.03744211 0.10895332
## 2     asymptotic 13 200 0.06500000 0.03083389 0.09916611
## 3          bayes 13 200 0.06716418 0.03457082 0.10225126
## 4        cloglog 13 200 0.06500000 0.03640889 0.10485405
## 5          exact 13 200 0.06500000 0.03506071 0.10858724
## 6          logit 13 200 0.06500000 0.03811331 0.10871006
## 7         probit 13 200 0.06500000 0.03724733 0.10663196
## 8        profile 13 200 0.06500000 0.03629941 0.10475423
## 9            lrt 13 200 0.06500000 0.03629393 0.10475306
## 10     prop.test 13 200 0.06500000 0.03650749 0.11104309
## 11        wilson 13 200 0.06500000 0.03837635 0.10801908
```

That means, with 95% confidence we can say, roughly [4% to 12%] of the population have hypertension.