Introduction to Data Science Project

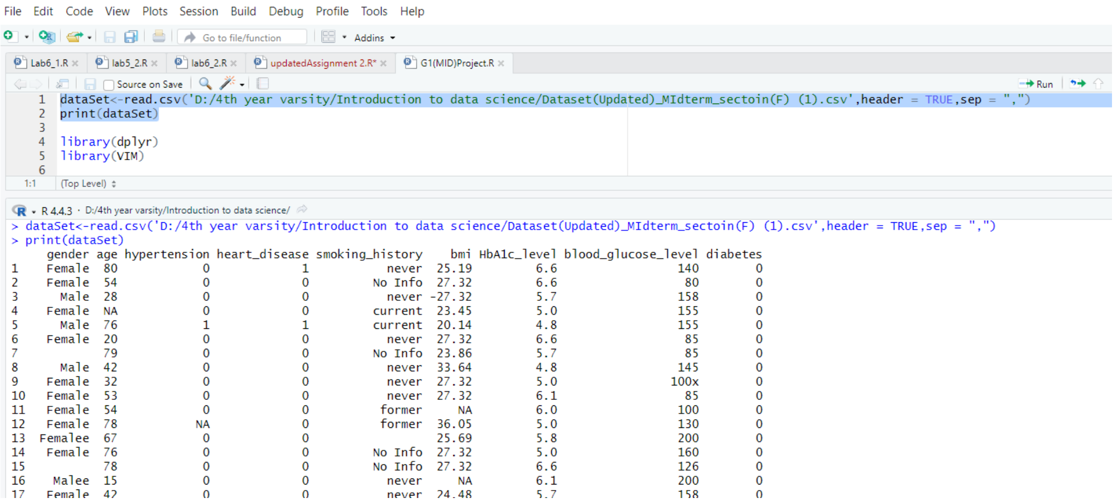
	ID	Name
Group 1	22-49401-3	Abdullah Al Meyad
	22-46066-1	Md. Zafrul Hasan
	22-47142-1	Md Sakib Hasan
	22-46281-1	Raihan Parvez

Description of Data Set:

This dataset is of predicting diabetes based on their age(continuous), gender(categorical), hypertension(categorical), heart disease(categorical), previous smoking history(categorical), BMI(continuous), HbA1c level(continuous), and blood glucose level(continuous). This is for binary classification to predict diabetes (0/1 or yes/no). It contains 59% female and 41% male. For this project we need to prepare the dataset and deal with missing value, outliers, invalid data. Balancing dataset and splitting it to train test.

Code explanation :

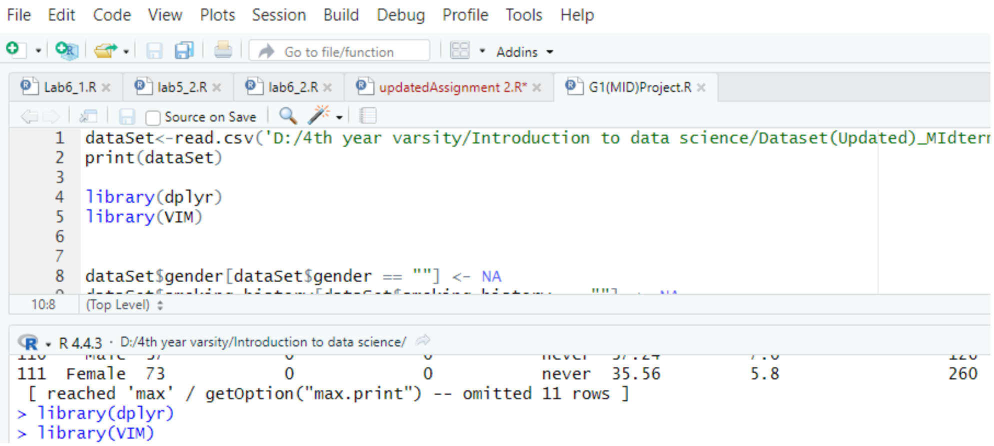
read.csv is used for reading any csv data set.


File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Lab6_1.R
lab5_2.R
lab6_2.R
updatedAssignment 2.R*
G1(MID)Project.R
Source on Save
Run
1:1
(Top Level)
R 4.4.3 · D:/4th year varsity/Introduction to data science/

```
dataSet<-read.csv('D:/4th year varsity/Introduction to data science/Dataset(Updated)_MIdterm_sectoin(F) (1).csv',header = TRUE,sep = ",")
print(dataSet)

library(dplyr)
library(VIM)

```

```
> dataSet<-read.csv('D:/4th year varsity/Introduction to data science/Dataset(Updated)_MIdterm_sectoin(F) (1).csv',header = TRUE,sep = ",")
> print(dataSet)
    gender age hypertension heart_disease smoking_history    bmi HbA1c_level blood_glucose_level diabetes
1   Female  80            0             1           never  25.19         6.6                 140        0
2   Female  54            0             0         No Info  27.32         6.6                  80        0
3     Male  28            0             0           never -27.32         5.7                 158        0
4   Female  NA            0             0         current  23.45         5.0                 155        0
5     Male  76            1             1         current  20.14         4.8                 155        0
6   Female  20            0             0           never  27.32         6.6                  85        0
7           79            0             0         No Info  23.86         5.7                  85        0
8     Male  42            0             0           never  33.64         4.8                 145        0
9   Female  32            0             0           never  27.32         5.0                100x        0
10  Female  53            0             0           never  27.32         6.1                  85        0
11  Female  54            0             0          former     NA         6.0                 100        0
12  Female  78           NA             0          former  36.05         5.0                 130        0
13 Femalee  67            0             0                  25.69         5.8                 200        0
14  Female  76            0             0         No Info  27.32         5.0                 160        0
15          78            0             0         No Info  27.32         6.6                 126        0
16   Malee  15            0             0           never     NA         6.1                 200        0
17  Female  42            0             0           never  24.48         5.7                 158        0
```

dplyr is a strong library for data manipulation and VIM is for data visualization.

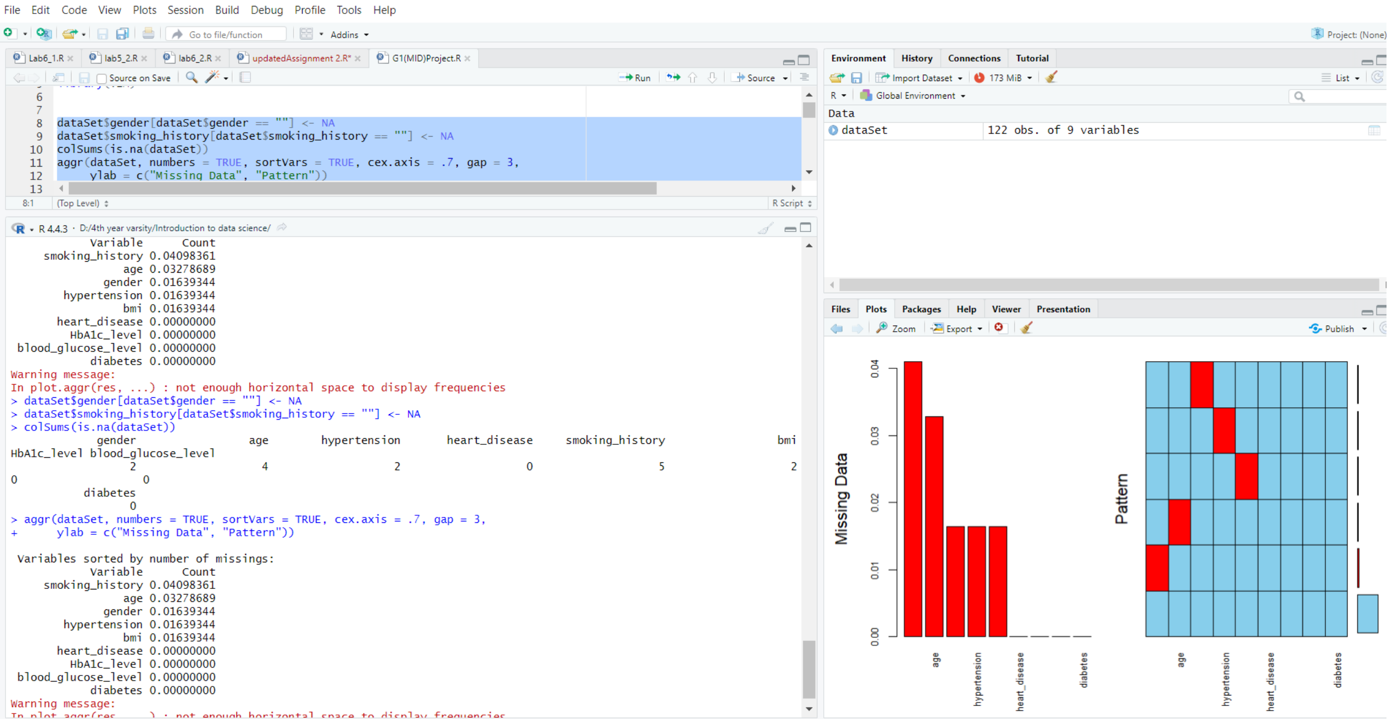


File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Lab6_1.R
lab5_2.R
lab6_2.R
updatedAssignment 2.R*
G1(MID)Project.R
Source on Save
10:8
(Top Level)
R 4.4.3 · D:/4th year varsity/Introduction to data science/

```r
dataSet<-read.csv('D:/4th year varsity/Introduction to data science/Dataset(Updated)_MIdterr
print(dataSet)

library(dplyr)
library(VIM)


dataSet$gender[dataSet$gender == ""] <- NA
```

```
111   Female  73           0           0     never   35.56         5.8             260
 [ reached 'max' / getOption("max.print") -- omitted 11 rows ]
> library(dplyr)
> library(VIM)
```

Using VIM library and aggr() function for plotting missing data. Replacing all empty value with NA so that helps agrr() to find empty value.

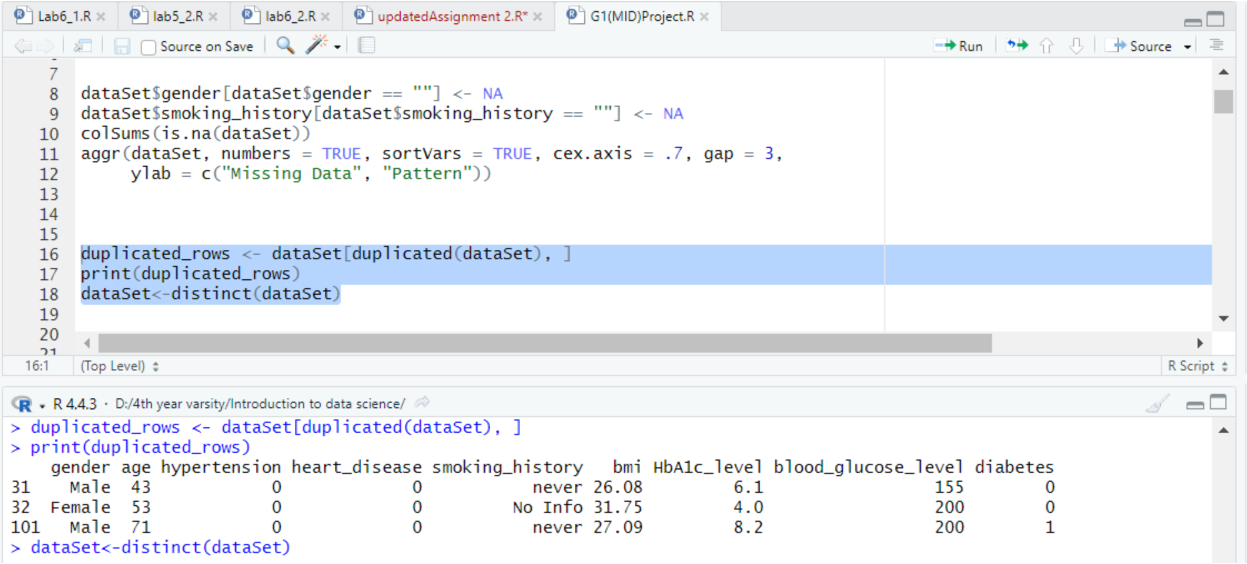


```r
dataSet$gender[dataSet$gender == ""] <- NA
dataSet$smoking_history[dataSet$smoking_history == ""] <- NA
colSums(is.na(dataSet))
aggr(dataSet, numbers = TRUE, sortVars = TRUE, cex.axis = .7, gap = 3,
     ylab = c("Missing Data", "Pattern"))
```

```
Data
dataSet          122 obs. of 9 variables
```

```
          Variable      Count
   smoking_history 0.04098361
               age 0.03278689
            gender 0.01639344
      hypertension 0.01639344
               bmi 0.01639344
     heart_disease 0.00000000
       HbA1c_level 0.00000000
 blood_glucose_level 0.00000000
          diabetes 0.00000000
Warning message:
In plot.aggr(res, ...) : not enough horizontal space to display frequencies
> dataSet$gender[dataSet$gender == ""] <- NA
> dataSet$smoking_history[dataSet$smoking_history == ""] <- NA
> colSums(is.na(dataSet))
             gender                 age        hypertension       heart_disease     smoking_history                 bmi
HbA1c_level blood_glucose_level
                  2                   4                   2                   0                   5                   2
0                   0
           diabetes
                  0
> aggr(dataSet, numbers = TRUE, sortVars = TRUE, cex.axis = .7, gap = 3,
+      ylab = c("Missing Data", "Pattern"))

 Variables sorted by number of missings:
          Variable      Count
   smoking_history 0.04098361
               age 0.03278689
            gender 0.01639344
      hypertension 0.01639344
               bmi 0.01639344
     heart_disease 0.00000000
       HbA1c_level 0.00000000
 blood_glucose_level 0.00000000
          diabetes 0.00000000
Warning message:
```

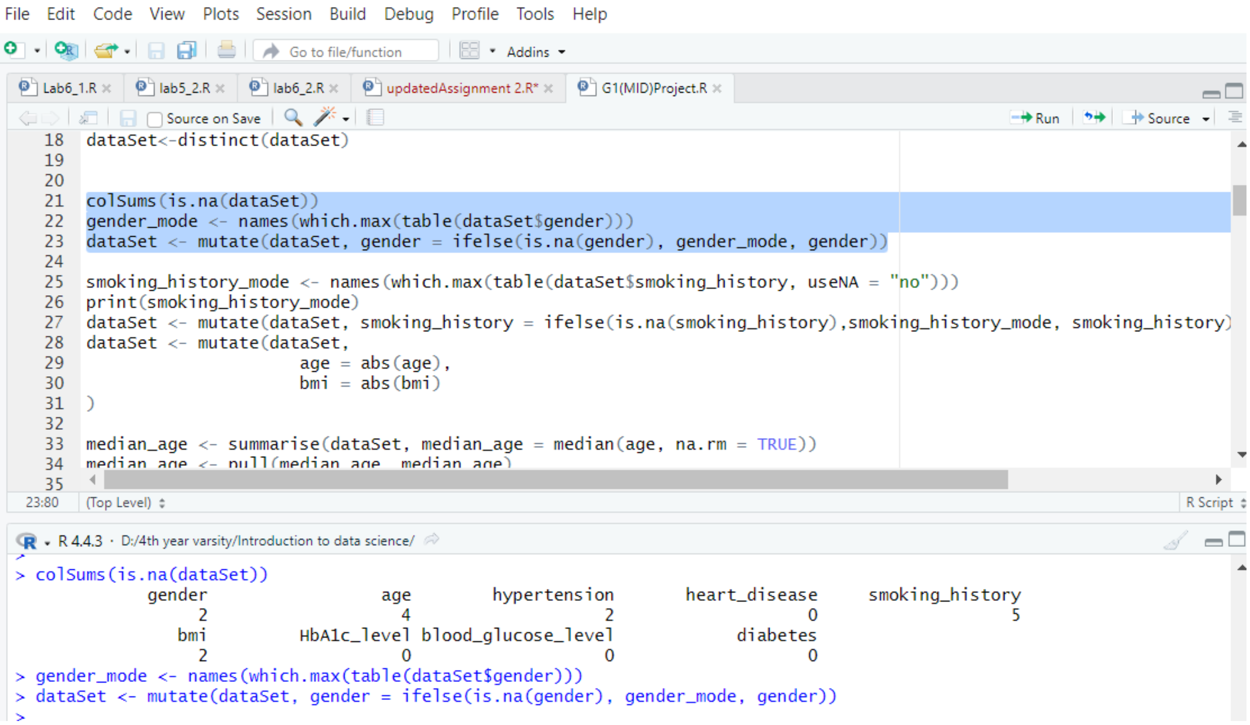
Finding duplicates and using distinct() a dplyr library function to make dataset duplicate value free.



```
dataSet$gender[dataSet$gender == ""] <- NA
dataSet$smoking_history[dataSet$smoking_history == ""] <- NA
colSums(is.na(dataSet))
aggr(dataSet, numbers = TRUE, sortVars = TRUE, cex.axis = .7, gap = 3,
     ylab = c("Missing Data", "Pattern"))



duplicated_rows <- dataSet[duplicated(dataSet), ]
print(duplicated_rows)
dataSet<-distinct(dataSet)
```

```
> duplicated_rows <- dataSet[duplicated(dataSet), ]
> print(duplicated_rows)
    gender age hypertension heart_disease smoking_history   bmi HbA1c_level blood_glucose_level diabetes
31    Male  43            0             0           never 26.08         6.1                 155        0
32  Female  53            0             0         No Info 31.75         4.0                 200        0
101   Male  71            0             0           never 27.09         8.2                 200        1
> dataSet<-distinct(dataSet)
```

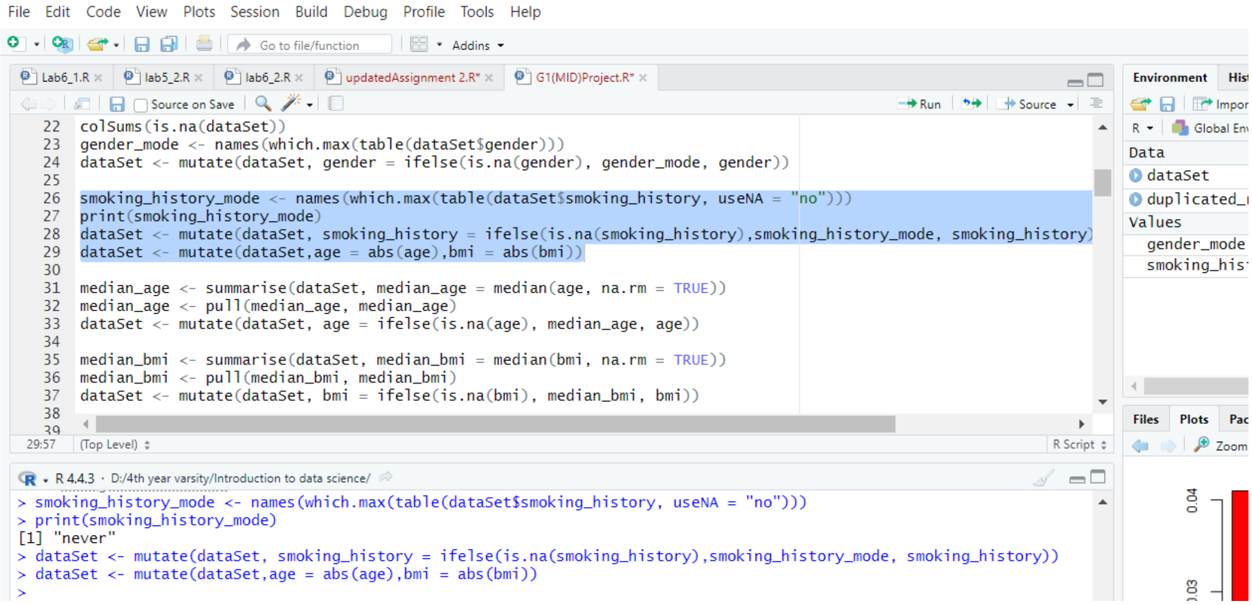
Checking all missing values (colSum(is.na(dataset)and replacing missing gender with most frequent value(Mode).



```
dataSet<-distinct(dataSet)


colSums(is.na(dataSet))
gender_mode <- names(which.max(table(dataSet$gender)))
dataSet <- mutate(dataSet, gender = ifelse(is.na(gender), gender_mode, gender))

smoking_history_mode <- names(which.max(table(dataSet$smoking_history, useNA = "no")))
print(smoking_history_mode)
dataSet <- mutate(dataSet, smoking_history = ifelse(is.na(smoking_history),smoking_history_mode, smoking_history)
dataSet <- mutate(dataSet,
                  age = abs(age),
                  bmi = abs(bmi)
)

median_age <- summarise(dataSet, median_age = median(age, na.rm = TRUE))
median_age <- pull(median_age, median_age)
```

```
> colSums(is.na(dataSet))
             gender                 age        hypertension       heart_disease     smoking_history 
                  2                   4                   2                   0                   5 
                bmi         HbA1c_level blood_glucose_level            diabetes 
                  2                   0                   0                   0 
> gender_mode <- names(which.max(table(dataSet$gender)))
> dataSet <- mutate(dataSet, gender = ifelse(is.na(gender), gender_mode, gender))
>
```

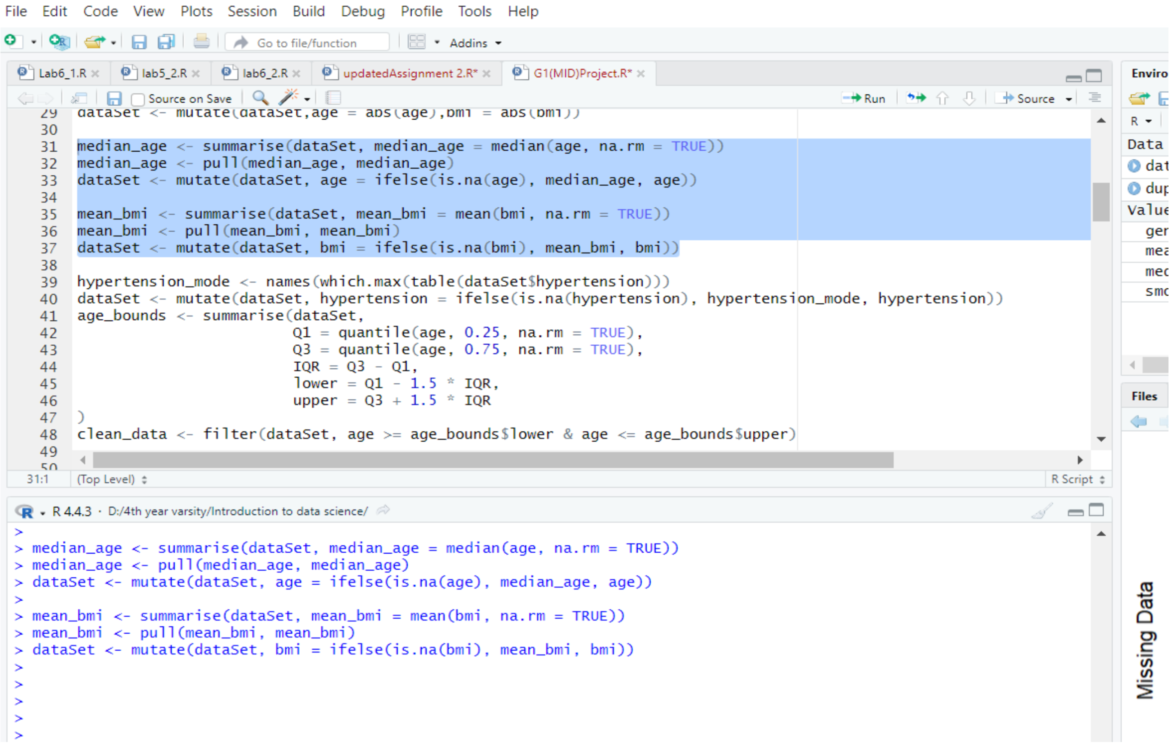
Replacing all Smoking history empty cell with frequent value(Mode) and mutating all negative value using abs() for age and bmi as those contains negative value.



```
colSums(is.na(dataSet))
gender_mode <- names(which.max(table(dataSet$gender)))
dataSet <- mutate(dataSet, gender = ifelse(is.na(gender), gender_mode, gender))

smoking_history_mode <- names(which.max(table(dataSet$smoking_history, useNA = "no")))
print(smoking_history_mode)
dataSet <- mutate(dataSet, smoking_history = ifelse(is.na(smoking_history),smoking_history_mode, smoking_history)
dataSet <- mutate(dataSet,age = abs(age),bmi = abs(bmi))

median_age <- summarise(dataSet, median_age = median(age, na.rm = TRUE))
median_age <- pull(median_age, median_age)
dataSet <- mutate(dataSet, age = ifelse(is.na(age), median_age, age))

median_bmi <- summarise(dataSet, median_bmi = median(bmi, na.rm = TRUE))
median_bmi <- pull(median_bmi, median_bmi)
dataSet <- mutate(dataSet, bmi = ifelse(is.na(bmi), median_bmi, bmi))
```

```
> smoking_history_mode <- names(which.max(table(dataSet$smoking_history, useNA = "no")))
> print(smoking_history_mode)
[1] "never"
> dataSet <- mutate(dataSet, smoking_history = ifelse(is.na(smoking_history),smoking_history_mode, smoking_history))
> dataSet <- mutate(dataSet,age = abs(age),bmi = abs(bmi))
>
```

Replacing missing value of age with median value as it has outliers and bmi with average or mean .

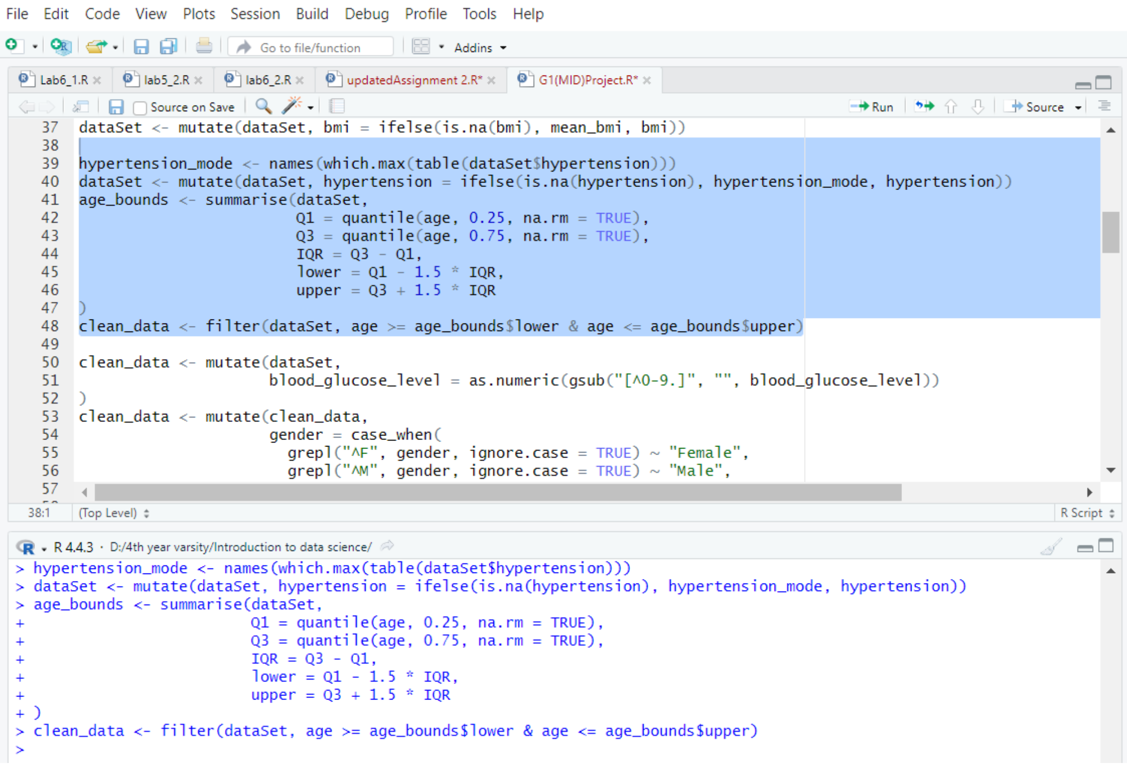


```
dataSet <- mutate(dataSet,age = abs(age),bmi = abs(bmi))

median_age <- summarise(dataSet, median_age = median(age, na.rm = TRUE))
median_age <- pull(median_age, median_age)
dataSet <- mutate(dataSet, age = ifelse(is.na(age), median_age, age))

mean_bmi <- summarise(dataSet, mean_bmi = mean(bmi, na.rm = TRUE))
mean_bmi <- pull(mean_bmi, mean_bmi)
dataSet <- mutate(dataSet, bmi = ifelse(is.na(bmi), mean_bmi, bmi))

hypertension_mode <- names(which.max(table(dataSet$hypertension)))
dataSet <- mutate(dataSet, hypertension = ifelse(is.na(hypertension), hypertension_mode, hypertension))
age_bounds <- summarise(dataSet,
                        Q1 = quantile(age, 0.25, na.rm = TRUE),
                        Q3 = quantile(age, 0.75, na.rm = TRUE),
                        IQR = Q3 - Q1,
                        lower = Q1 - 1.5 * IQR,
                        upper = Q3 + 1.5 * IQR
)
clean_data <- filter(dataSet, age >= age_bounds$lower & age <= age_bounds$upper)
```

```
>
> median_age <- summarise(dataSet, median_age = median(age, na.rm = TRUE))
> median_age <- pull(median_age, median_age)
> dataSet <- mutate(dataSet, age = ifelse(is.na(age), median_age, age))
>
> mean_bmi <- summarise(dataSet, mean_bmi = mean(bmi, na.rm = TRUE))
> mean_bmi <- pull(mean_bmi, mean_bmi)
> dataSet <- mutate(dataSet, bmi = ifelse(is.na(bmi), mean_bmi, bmi))
>
>
>
>
>
```

Replacing Hypertension with most frequent value as it is binary categorical.

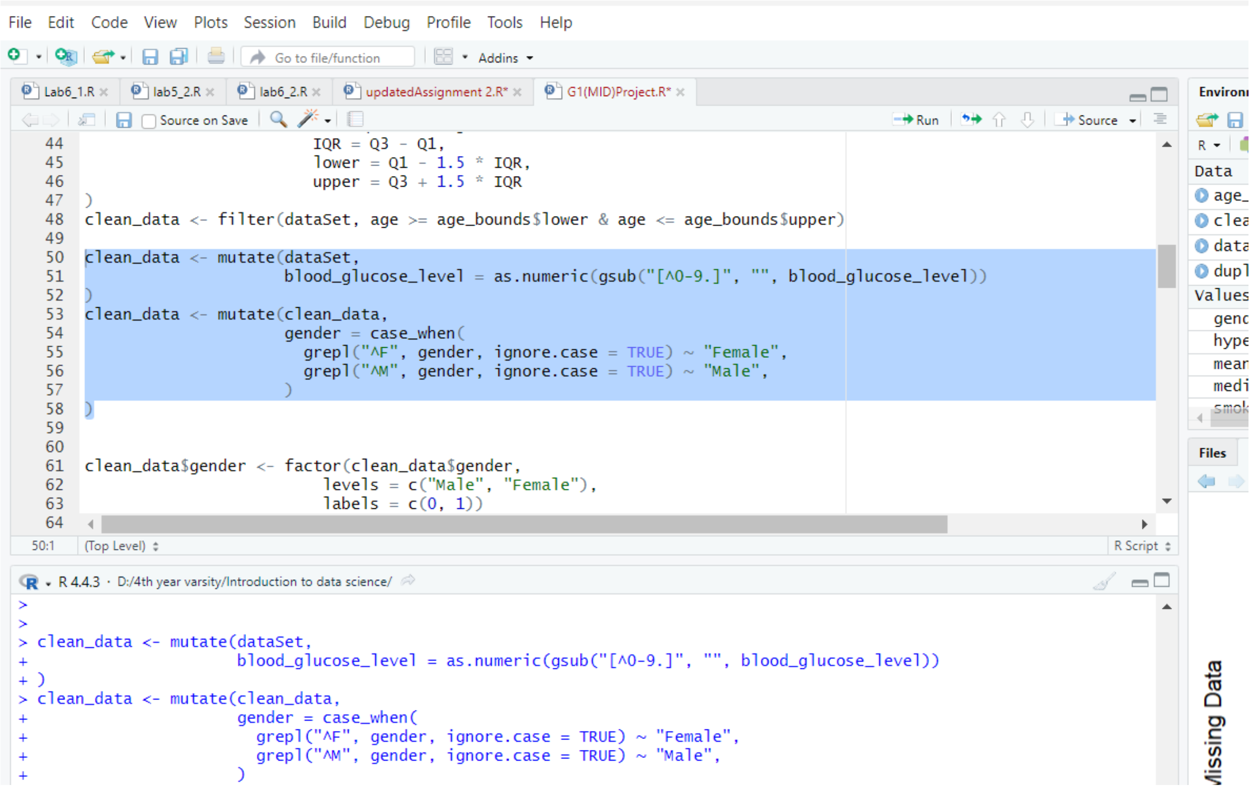
Calculating age value upper and lower bound using IQR to remove outliers.



```r
dataSet <- mutate(dataSet, bmi = ifelse(is.na(bmi), mean_bmi, bmi))

hypertension_mode <- names(which.max(table(dataSet$hypertension)))
dataSet <- mutate(dataSet, hypertension = ifelse(is.na(hypertension), hypertension_mode, hypertension))
age_bounds <- summarise(dataSet,
                        Q1 = quantile(age, 0.25, na.rm = TRUE),
                        Q3 = quantile(age, 0.75, na.rm = TRUE),
                        IQR = Q3 - Q1,
                        lower = Q1 - 1.5 * IQR,
                        upper = Q3 + 1.5 * IQR
)
clean_data <- filter(dataSet, age >= age_bounds$lower & age <= age_bounds$upper)

clean_data <- mutate(dataSet,
                     blood_glucose_level = as.numeric(gsub("[^0-9.]", "", blood_glucose_level))
)
clean_data <- mutate(clean_data,
                     gender = case_when(
                       grepl("^F", gender, ignore.case = TRUE) ~ "Female",
                       grepl("^M", gender, ignore.case = TRUE) ~ "Male",
```

```
> hypertension_mode <- names(which.max(table(dataSet$hypertension)))
> dataSet <- mutate(dataSet, hypertension = ifelse(is.na(hypertension), hypertension_mode, hypertension))
> age_bounds <- summarise(dataSet,
+                         Q1 = quantile(age, 0.25, na.rm = TRUE),
+                         Q3 = quantile(age, 0.75, na.rm = TRUE),
+                         IQR = Q3 - Q1,
+                         lower = Q1 - 1.5 * IQR,
+                         upper = Q3 + 1.5 * IQR
+ )
> clean_data <- filter(dataSet, age >= age_bounds$lower & age <= age_bounds$upper)
>
```

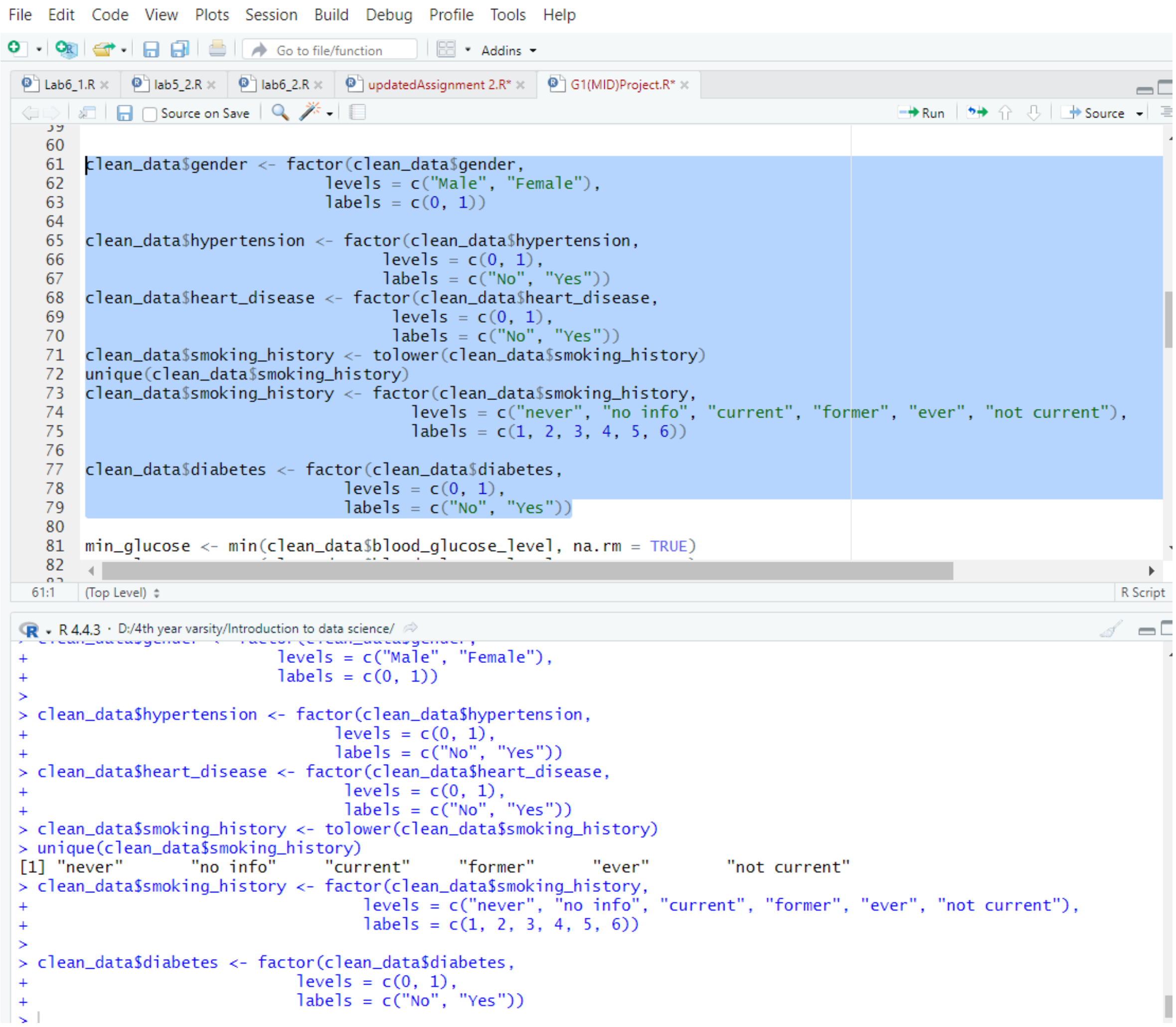
Removing invalid value from glucose level as it has 100x and in gender there ase Malee and Femalee type date . So this part will remove only remove x not entire row and gender to Male if it contains M as first letter and Female if F is first leter.



```
                        IQR = Q3 - Q1,
                        lower = Q1 - 1.5 * IQR,
                        upper = Q3 + 1.5 * IQR
)
clean_data <- filter(dataSet, age >= age_bounds$lower & age <= age_bounds$upper)

clean_data <- mutate(dataSet,
                     blood_glucose_level = as.numeric(gsub("[^0-9.]", "", blood_glucose_level))
)
clean_data <- mutate(clean_data,
                     gender = case_when(
                       grepl("^F", gender, ignore.case = TRUE) ~ "Female",
                       grepl("^M", gender, ignore.case = TRUE) ~ "Male",
                     )
)


clean_data$gender <- factor(clean_data$gender,
                         levels = c("Male", "Female"),
                         labels = c(0, 1))
```

```
>
>
> clean_data <- mutate(dataSet,
+                      blood_glucose_level = as.numeric(gsub("[^0-9.]", "", blood_glucose_level))
+ )
> clean_data <- mutate(clean_data,
+                      gender = case_when(
+                        grepl("^F", gender, ignore.case = TRUE) ~ "Female",
+                        grepl("^M", gender, ignore.case = TRUE) ~ "Male",
+                      )
```

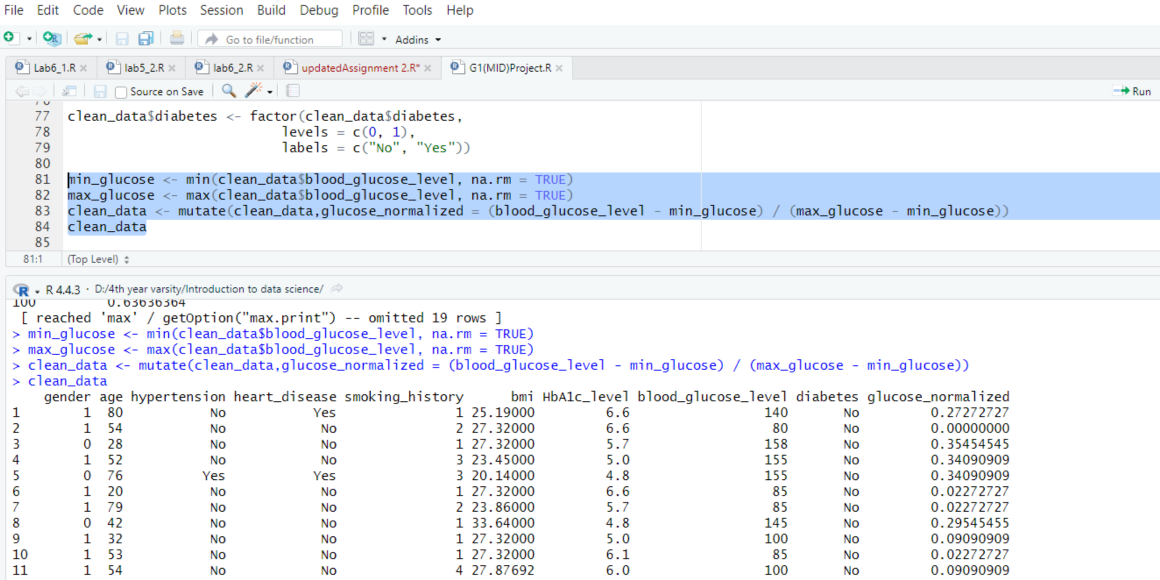
Converting attributes from numerical to categorical and categorical to numeric using factor().



```
clean_data$gender <- factor(clean_data$gender,
                         levels = c("Male", "Female"),
                         labels = c(0, 1))

clean_data$hypertension <- factor(clean_data$hypertension,
                               levels = c(0, 1),
                               labels = c("No", "Yes"))
clean_data$heart_disease <- factor(clean_data$heart_disease,
                                levels = c(0, 1),
                                labels = c("No", "Yes"))
clean_data$smoking_history <- tolower(clean_data$smoking_history)
unique(clean_data$smoking_history)
clean_data$smoking_history <- factor(clean_data$smoking_history,
                                  levels = c("never", "no info", "current", "former", "ever", "not current"),
                                  labels = c(1, 2, 3, 4, 5, 6))

clean_data$diabetes <- factor(clean_data$diabetes,
                           levels = c(0, 1),
                           labels = c("No", "Yes"))

min_glucose <- min(clean_data$blood_glucose_level, na.rm = TRUE)
```

```
> clean_data$gender <- factor(clean_data$gender,
+                          levels = c("Male", "Female"),
+                          labels = c(0, 1))
>
> clean_data$hypertension <- factor(clean_data$hypertension,
+                                levels = c(0, 1),
+                                labels = c("No", "Yes"))
> clean_data$heart_disease <- factor(clean_data$heart_disease,
+                                 levels = c(0, 1),
+                                 labels = c("No", "Yes"))
> clean_data$smoking_history <- tolower(clean_data$smoking_history)
> unique(clean_data$smoking_history)
[1] "never"       "no info"     "current"     "former"      "ever"        "not current"
> clean_data$smoking_history <- factor(clean_data$smoking_history,
+                                   levels = c("never", "no info", "current", "former", "ever", "not current"),
+                                   labels = c(1, 2, 3, 4, 5, 6))
>
> clean_data$diabetes <- factor(clean_data$diabetes,
+                            levels = c(0, 1),
+                            labels = c("No", "Yes"))
> 
```

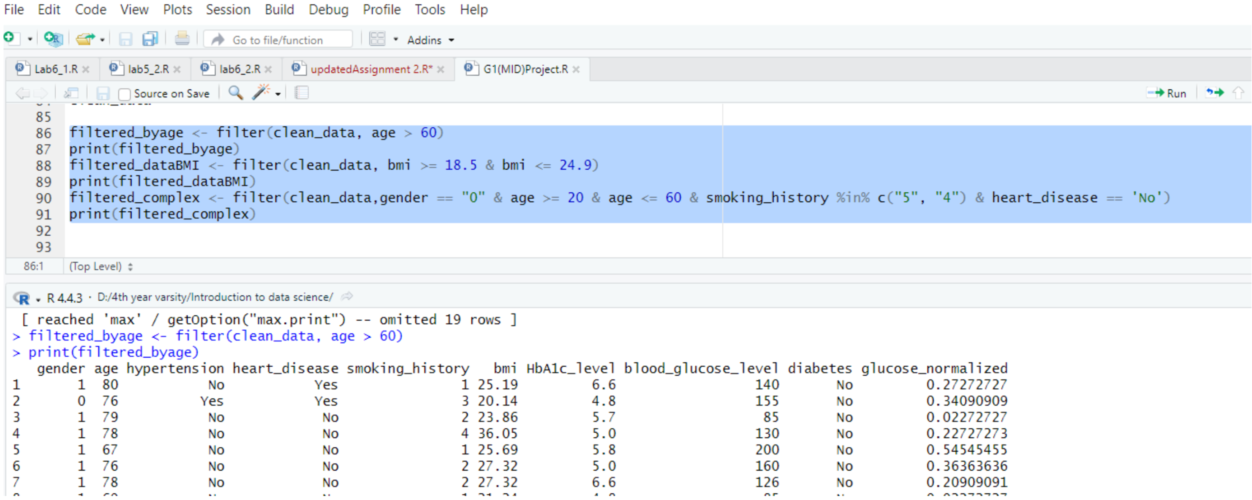
As blood glucose level is continuous so it’s normalized form is $\frac{value-min}{max-min}$. It will bring every value of this attribute to 0~1.


File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Lab6_1.R
lab5_2.R
lab6_2.R
updatedAssignment 2.R*
G1(MID)Project.R
Source on Save
Run
81:1
(Top Level)
R 4.4.3 · D:/4th year varsity/Introduction to data science/

```
clean_data$diabetes <- factor(clean_data$diabetes,
                              levels = c(0, 1),
                              labels = c("No", "Yes"))

min_glucose <- min(clean_data$blood_glucose_level, na.rm = TRUE)
max_glucose <- max(clean_data$blood_glucose_level, na.rm = TRUE)
clean_data <- mutate(clean_data,glucose_normalized = (blood_glucose_level - min_glucose) / (max_glucose - min_glucose))
clean_data
```

```
100         0.63636364
 [ reached 'max' / getOption("max.print") -- omitted 19 rows ]
> min_glucose <- min(clean_data$blood_glucose_level, na.rm = TRUE)
> max_glucose <- max(clean_data$blood_glucose_level, na.rm = TRUE)
> clean_data <- mutate(clean_data,glucose_normalized = (blood_glucose_level - min_glucose) / (max_glucose - min_glucose))
> clean_data
    gender age hypertension heart_disease smoking_history      bmi HbA1c_level blood_glucose_level diabetes glucose_normalized
1        1  80           No           Yes               1 25.19000         6.6                 140       No         0.27272727
2        1  54           No            No               2 27.32000         6.6                  80       No         0.00000000
3        0  28           No            No               1 27.32000         5.7                 158       No         0.35454545
4        1  52           No            No               3 23.45000         5.0                 155       No         0.34090909
5        0  76          Yes           Yes               3 20.14000         4.8                 155       No         0.34090909
6        1  20           No            No               1 27.32000         6.6                  85       No         0.02272727
7        1  79           No            No               2 23.86000         5.7                  85       No         0.02272727
8        0  42           No            No               1 33.64000         4.8                 145       No         0.29545455
9        1  32           No            No               1 27.32000         5.0                 100       No         0.09090909
10       1  53           No            No               1 27.32000         6.1                  85       No         0.02272727
11       1  54           No            No               4 27.87692         6.0                 100       No         0.09090909
```

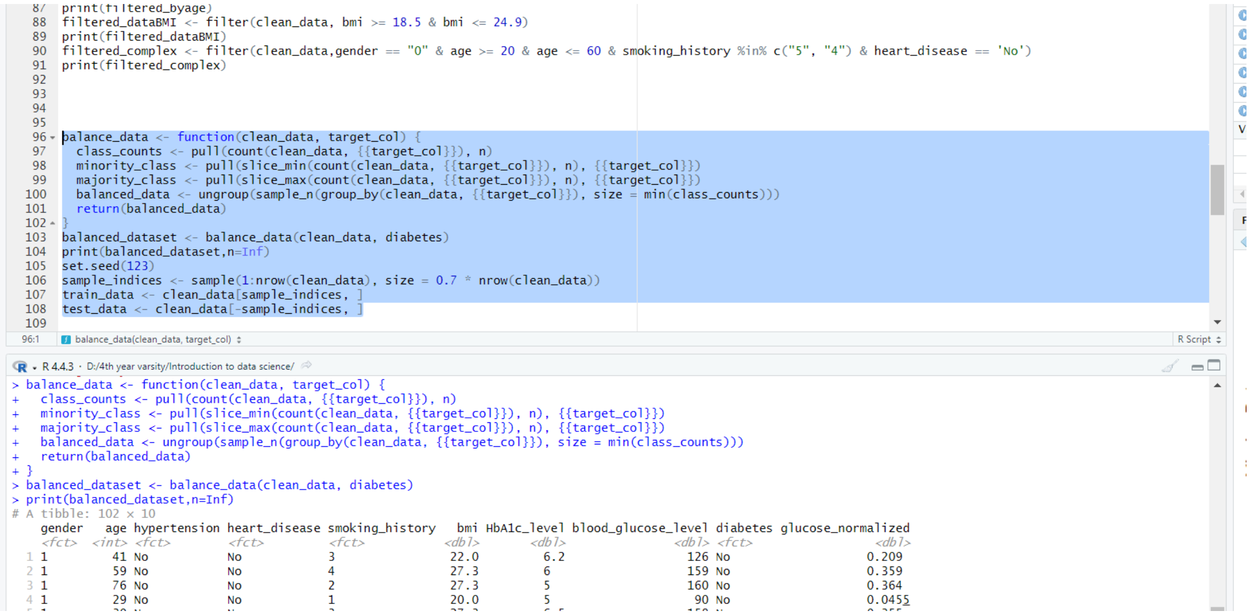
Here are some filters. First one selects people older than 60, 2nd one picks people with a healthy BMI between 18.5 and 24.9, 3rd one finds non-diabetic females aged 20–60 who smoke (either "ever" or "former") and don’t have heart disease.


File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Lab6_1.R
lab5_2.R
lab6_2.R
updatedAssignment 2.R*
G1(MID)Project.R
Source on Save
Run
86:1
(Top Level)
R 4.4.3 · D:/4th year varsity/Introduction to data science/

```

filtered_byage <- filter(clean_data, age > 60)
print(filtered_byage)
filtered_dataBMI <- filter(clean_data, bmi >= 18.5 & bmi <= 24.9)
print(filtered_dataBMI)
filtered_complex <- filter(clean_data,gender == "0" & age >= 20 & age <= 60 & smoking_history %in% c("5", "4") & heart_disease == 'No')
print(filtered_complex)
```

```
 [ reached 'max' / getOption("max.print") -- omitted 19 rows ]
> filtered_byage <- filter(clean_data, age > 60)
> print(filtered_byage)
  gender age hypertension heart_disease smoking_history   bmi HbA1c_level blood_glucose_level diabetes glucose_normalized
1      1  80           No           Yes               1 25.19         6.6                 140       No         0.27272727
2      0  76          Yes           Yes               3 20.14         4.8                 155       No         0.34090909
3      1  79           No            No               2 23.86         5.7                  85       No         0.02272727
4      1  78           No            No               4 36.05         5.0                 130       No         0.22727273
5      1  67           No            No               1 25.69         5.8                 200       No         0.54545455
6      1  76           No            No               2 27.32         5.0                 160       No         0.36363636
7      1  78           No            No               2 27.32         6.6                 126       No         0.20909091
```

For balancing down sampling being followed. As all class being calculated and majority call has been down sampled. The data set is devided into train and test in 70% and 30%.

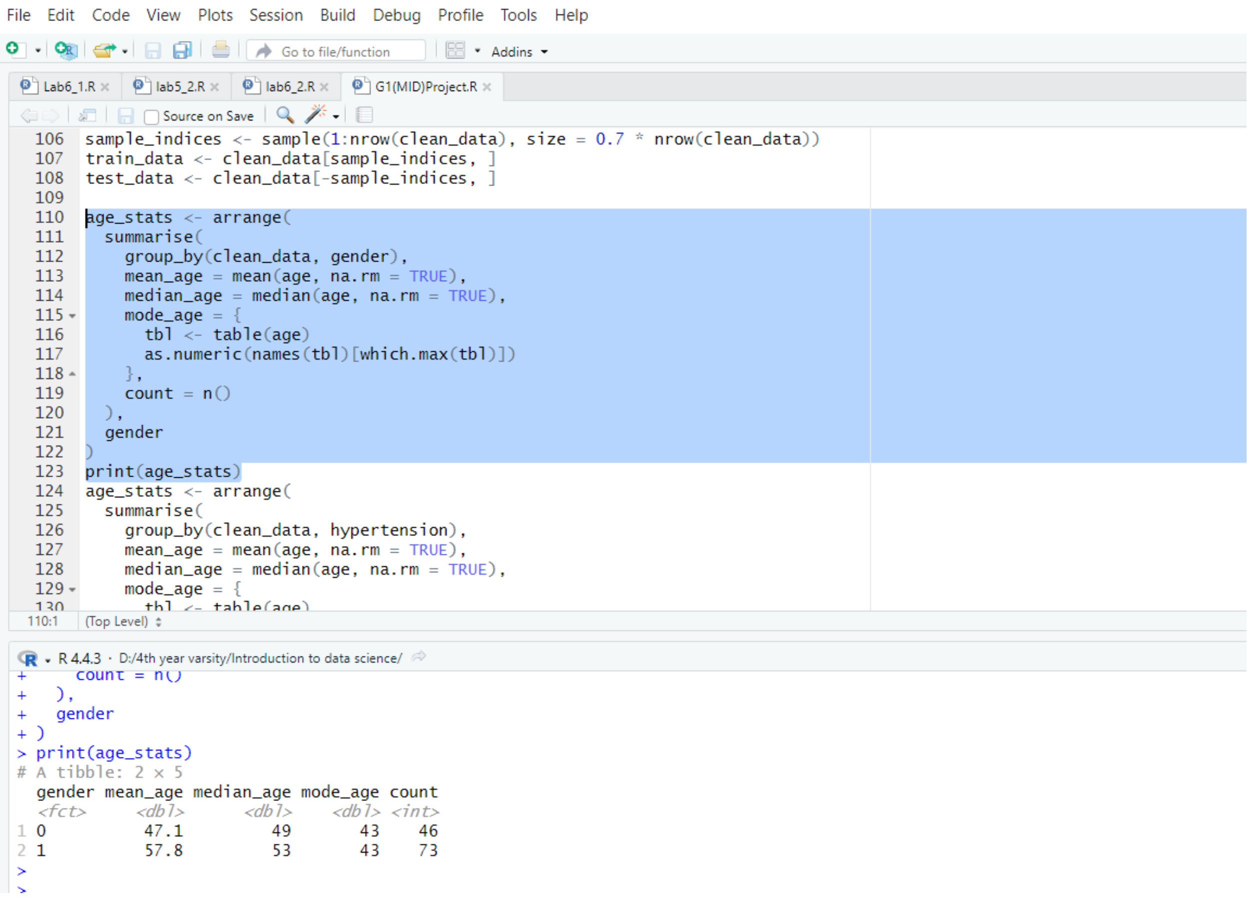


```
print(filtered_byage)
filtered_dataBMI <- filter(clean_data, bmi >= 18.5 & bmi <= 24.9)
print(filtered_dataBMI)
filtered_complex <- filter(clean_data,gender == "0" & age >= 20 & age <= 60 & smoking_history %in% c("5", "4") & heart_disease == 'No')
print(filtered_complex)




balance_data <- function(clean_data, target_col) {
  class_counts <- pull(count(clean_data, {{target_col}}), n)
  minority_class <- pull(slice_min(count(clean_data, {{target_col}}), n), {{target_col}})
  majority_class <- pull(slice_max(count(clean_data, {{target_col}}), n), {{target_col}})
  balanced_data <- ungroup(sample_n(group_by(clean_data, {{target_col}}), size = min(class_counts)))
  return(balanced_data)
}
balanced_dataset <- balance_data(clean_data, diabetes)
print(balanced_dataset,n=Inf)
set.seed(123)
sample_indices <- sample(1:nrow(clean_data), size = 0.7 * nrow(clean_data))
train_data <- clean_data[sample_indices, ]
test_data <- clean_data[-sample_indices, ]
```

96:1
balance_data(clean_data, target_col)
R Script
R 4.4.3 · D:/4th year varsity/Introduction to data science/

```
> balance_data <- function(clean_data, target_col) {
+   class_counts <- pull(count(clean_data, {{target_col}}), n)
+   minority_class <- pull(slice_min(count(clean_data, {{target_col}}), n), {{target_col}})
+   majority_class <- pull(slice_max(count(clean_data, {{target_col}}), n), {{target_col}})
+   balanced_data <- ungroup(sample_n(group_by(clean_data, {{target_col}}), size = min(class_counts)))
+   return(balanced_data)
+ }
> balanced_dataset <- balance_data(clean_data, diabetes)
> print(balanced_dataset,n=Inf)
# A tibble: 102 × 10
  gender   age hypertension heart_disease smoking_history   bmi HbA1c_level blood_glucose_level diabetes glucose_normalized
  <fct>  <int> <fct>        <fct>         <fct>           <dbl>       <dbl>               <dbl> <fct>                 <dbl>
1 1         41 No           No            3                22.0         6.2                 126 No                   0.209
2 1         59 No           No            4                27.3         6                   159 No                   0.359
3 1         76 No           No            2                27.3         5                   160 No                   0.364
4 1         29 No           No            1                20.0         5                    90 No                   0.0455
```

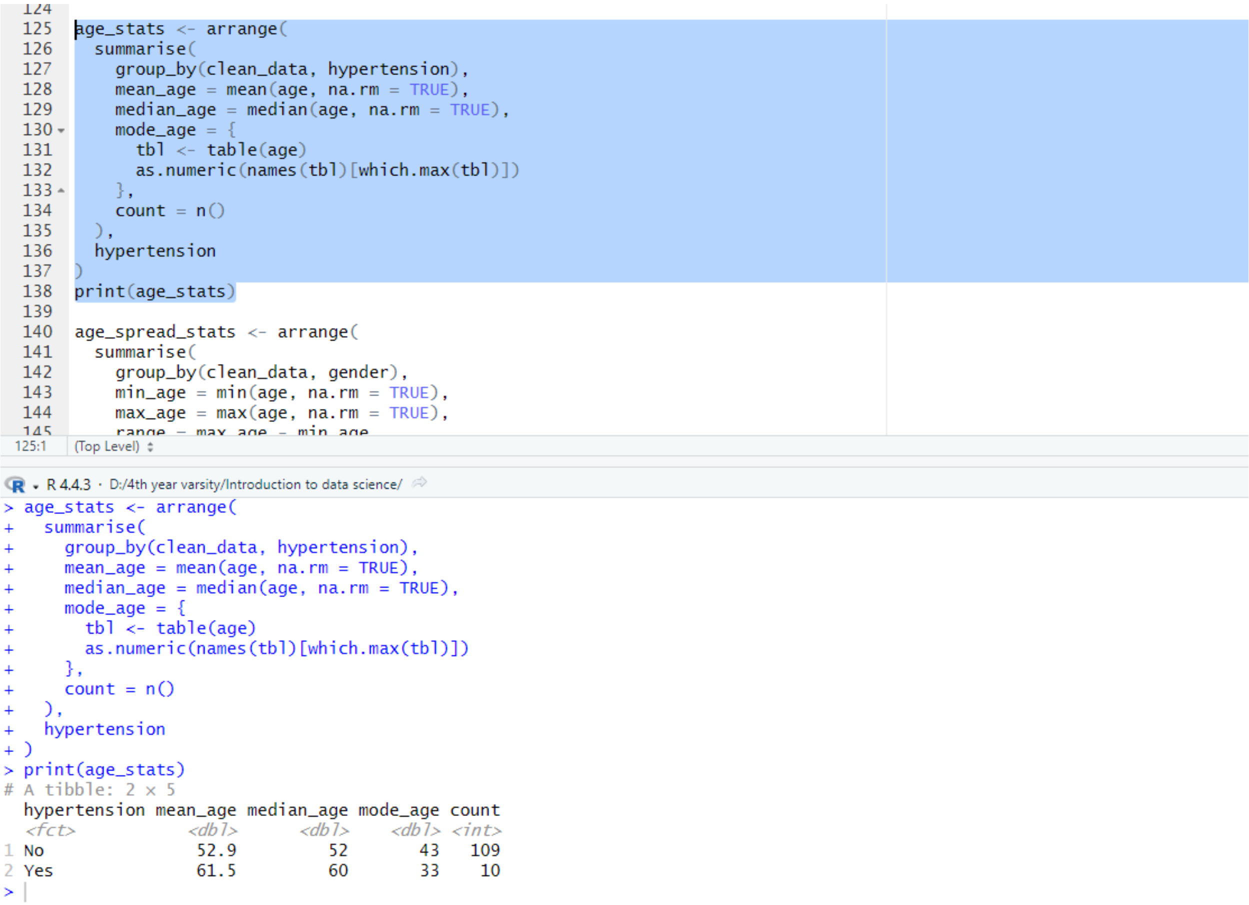
Here 0-> Male and 1->Female. As it shows Females are older in comparison to men in this dataset.



```
sample_indices <- sample(1:nrow(clean_data), size = 0.7 * nrow(clean_data))
train_data <- clean_data[sample_indices, ]
test_data <- clean_data[-sample_indices, ]

age_stats <- arrange(
  summarise(
    group_by(clean_data, gender),
    mean_age = mean(age, na.rm = TRUE),
    median_age = median(age, na.rm = TRUE),
    mode_age = {
      tbl <- table(age)
      as.numeric(names(tbl)[which.max(tbl)])
    },
    count = n()
  ),
  gender
)
print(age_stats)
age_stats <- arrange(
  summarise(
    group_by(clean_data, hypertension),
    mean_age = mean(age, na.rm = TRUE),
    median_age = median(age, na.rm = TRUE),
    mode_age = {
      tbl <- table(age)
```

```
+     count = n()
+   ),
+   gender
+ )
> print(age_stats)
# A tibble: 2 × 5
  gender mean_age median_age mode_age count
  <fct>     <dbl>      <dbl>    <dbl> <int>
1 0          47.1         49       43    46
2 1          57.8         53       43    73
>
>
```

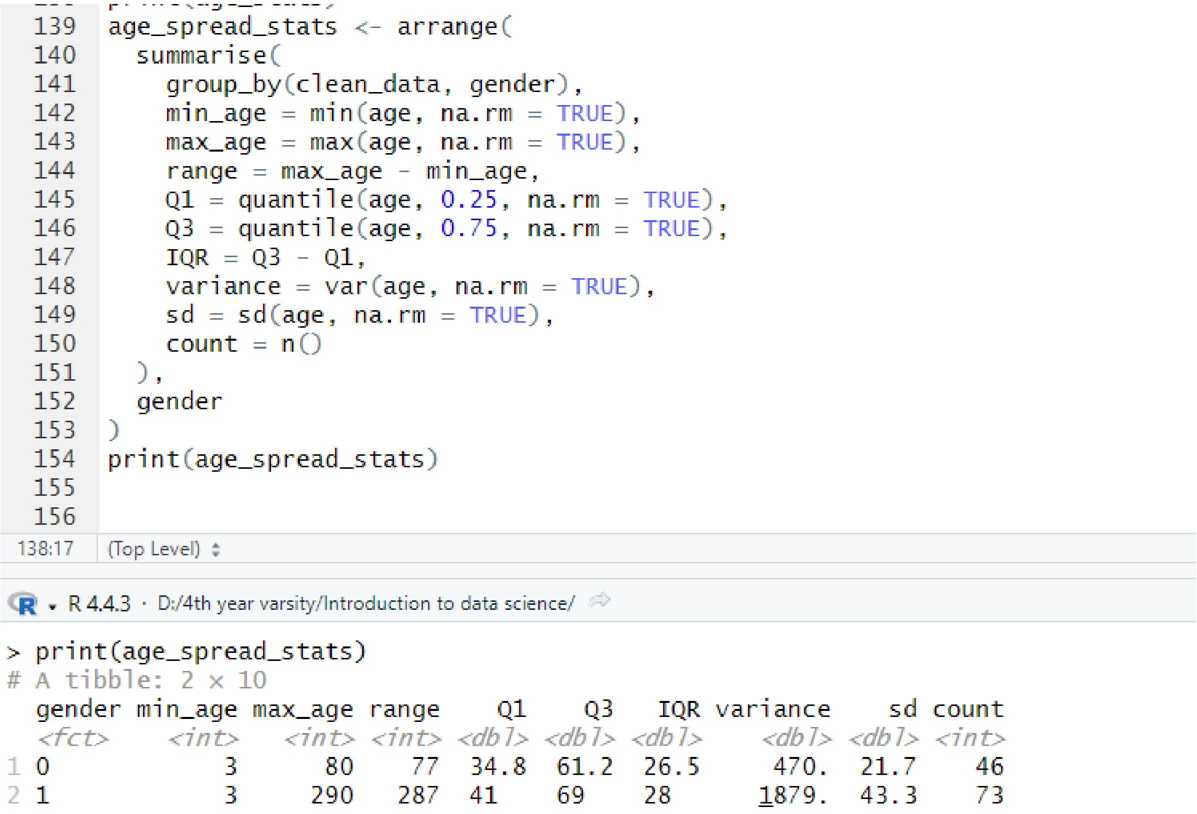
People with hypertension are generally older than those without it. Most people in the dataset don’t have hypertension.



```r
age_stats <- arrange(
  summarise(
    group_by(clean_data, hypertension),
    mean_age = mean(age, na.rm = TRUE),
    median_age = median(age, na.rm = TRUE),
    mode_age = {
      tbl <- table(age)
      as.numeric(names(tbl)[which.max(tbl)])
    },
    count = n()
  ),
  hypertension
)
print(age_stats)

age_spread_stats <- arrange(
  summarise(
    group_by(clean_data, gender),
    min_age = min(age, na.rm = TRUE),
    max_age = max(age, na.rm = TRUE),
    range = max_age - min_age
```

```
> age_stats <- arrange(
+   summarise(
+     group_by(clean_data, hypertension),
+     mean_age = mean(age, na.rm = TRUE),
+     median_age = median(age, na.rm = TRUE),
+     mode_age = {
+       tbl <- table(age)
+       as.numeric(names(tbl)[which.max(tbl)])
+     },
+     count = n()
+   ),
+   hypertension
+ )
> print(age_stats)
# A tibble: 2 × 5
  hypertension mean_age median_age mode_age count
  <fct>           <dbl>      <dbl>    <dbl> <int>
1 No               52.9         52       43   109
2 Yes              61.5         60       33    10
> 
```

Females show much wider age variability than males in this dataset.The standard deviation and variance are nearly double for females, showing their ages are more spread out. This could be due to the presence of extreme age values (like 290) in the female group.



```r
age_spread_stats <- arrange(
  summarise(
    group_by(clean_data, gender),
    min_age = min(age, na.rm = TRUE),
    max_age = max(age, na.rm = TRUE),
    range = max_age - min_age,
    Q1 = quantile(age, 0.25, na.rm = TRUE),
    Q3 = quantile(age, 0.75, na.rm = TRUE),
    IQR = Q3 - Q1,
    variance = var(age, na.rm = TRUE),
    sd = sd(age, na.rm = TRUE),
    count = n()
  ),
  gender
)
print(age_spread_stats)
```

```
> print(age_spread_stats)
# A tibble: 2 × 10
  gender min_age max_age range    Q1    Q3   IQR variance    sd count
  <fct>    <int>   <int> <int> <dbl> <dbl> <dbl>    <dbl> <dbl> <int>
1 0            3      80    77  34.8  61.2  26.5     470.  21.7    46
2 1            3     290   287  41    69    28      1879.  43.3    73
```