Assignment 1-R

Student Name and ID of the member submitting the assignment:

Dumpa Bharat Kumar, ID - 1001870815

Student Name and ID of the remaining members:

Likhita Muddana, ID – 1001949141

Dataset Details: health_stroke.csv

The dataset we used is related to the health_stroke

Required R packages:

```
library( data.table )

require(dplyr)
```

Read the file:

```
df <-read.csv("healthcare_stroke_dataset.csv", header=TRUE)
```

return the first 5 rows of the dataset

```
head(df, 5)
```

Task 1: Statistical EXPLORATORY DATA ANALYSIS.

a) print the details of data frame

Here we are giving the details of dimensions, summary, and columns names in the dataset

```
dim(df)
 str(df)
 summary(df)
 colnames(df)
```

b) Find the number of rows and columns in the dataset.

Here we are giving the number of rows and columns in the dataset.

```
ncol(df)
nrow(df)
```

c) Print the descriptive detail of a column in dataset.

Here we are giving the total summary in the dataset which includes everything.

```
summary(df)
```

d) Find all the count of unique values for 'avg_glucose_level' column in dataset

Here we are giving the unique values for the specified column in the dataset.

```
unique(df$avg_glucose_level)
```

e) Find all percentage of ‘Residency type ‘for all the values.

Here we are giving the percentage by using the SetDT(which is used to convert data.frames to data.tables) and then we group by Residence_type

```
library( data.table )

 setDT( df )[ , 100 *. N / nrow( df ), by = Residence_type ]
```

Task-2: Aggregation & Filtering & Rank

a) Find out the gender with the largest number of records.

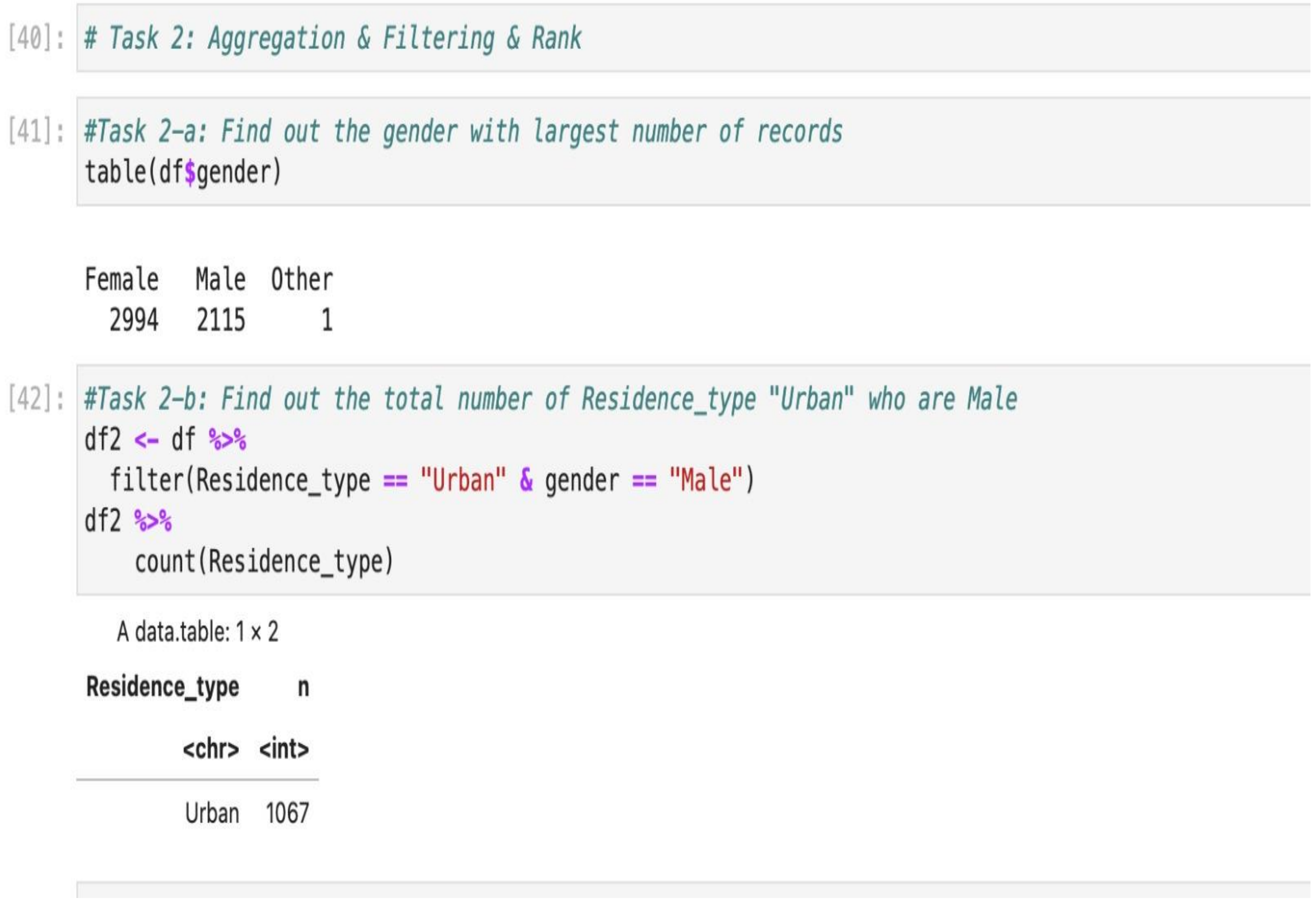
Here we are giving the gender records by using the ds$gender which helps us to get the gender column.

```
table(df$gender)
```

b) Find out the total number of Residency_type “Urban” who are Male

Here we are using the filter () so that we can select only male under the residency_type – Urban. So, we get the number of male urban residencies.

```
df2 <- df %>%

   filter(Residence_type == "Urban" & gender == "Male")

df2 %>%

   count(Residence_type)
```



```
[40]: # Task 2: Aggregation & Filtering & Rank
```

```
[41]: #Task 2-a: Find out the gender with largest number of records
      table(df$gender)
```

```
Female   Male  Other 
  2994   2115      1 
```

```
[42]: #Task 2-b: Find out the total number of Residence_type "Urban" who are Male
      df2 <- df %>%
        filter(Residence_type == "Urban" & gender == "Male")
      df2 %>%
          count(Residence_type)
```

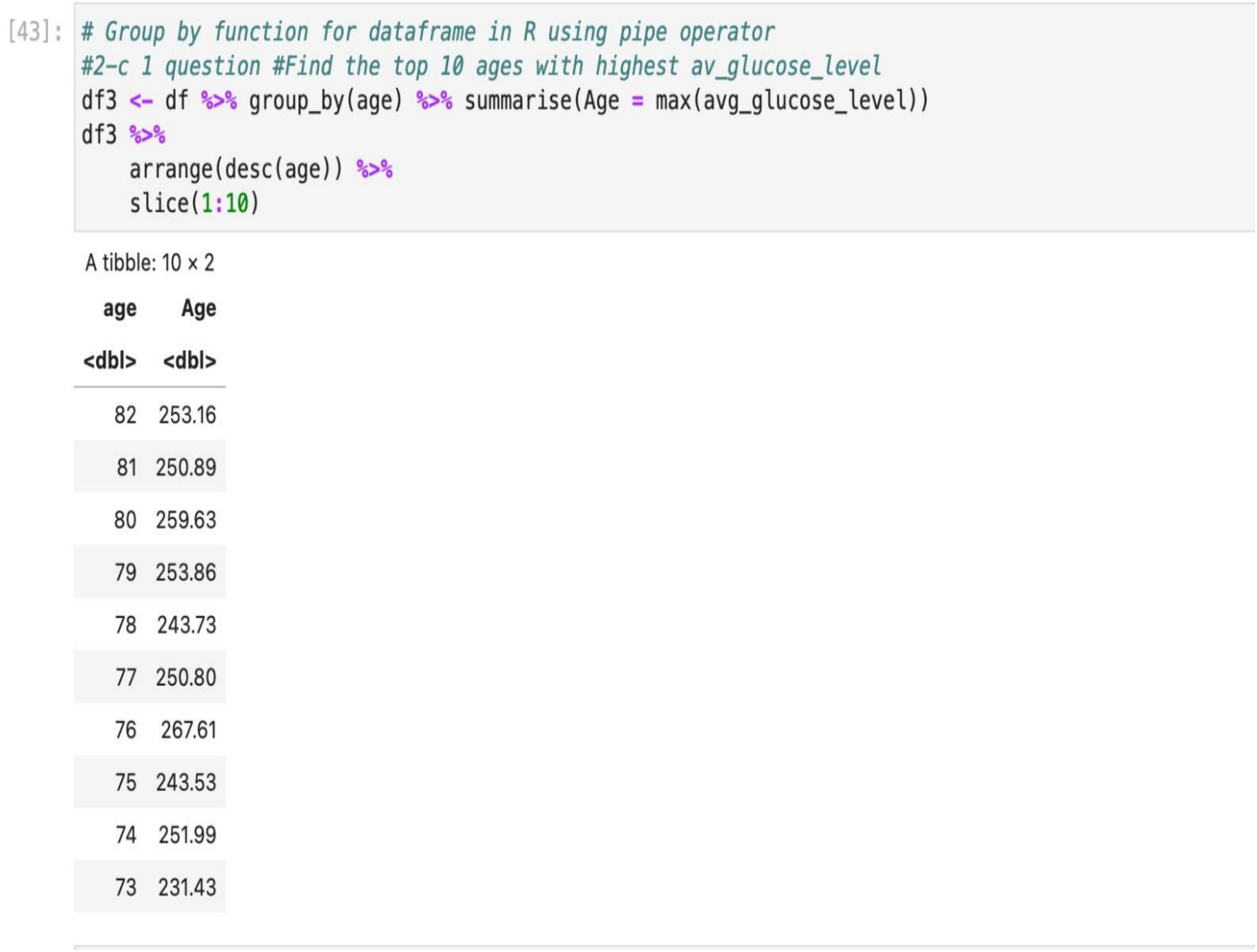
A data.table: 1 × 2

Residence_type	n
<chr>	<int>
Urban	1067

c) Find the top 10 ages with highest av_glucose_level

Here we are using max(avg_glucose_level) which gives us the maximum of glucose_level and using group_by(age) which gives the data frame record with age and arrange(desc(age)) which gives us the output of ages in the descending order and finally slice (1:10) is to get the top 10dataframe. rows of the

```
df3 <- df %>% group_by(age) %>% summarise(Age = max(avg_glucose_level))

df3 %>%

   arrange(desc(age)) %>%

   slice(1:10)
```



```
[43]: # Group by function for dataframe in R using pipe operator
      #2-c 1 question #Find the top 10 ages with highest av_glucose_level
      df3 <- df %>% group_by(age) %>% summarise(Age = max(avg_glucose_level))
      df3 %>%
          arrange(desc(age)) %>%
          slice(1:10)
```

A tibble: 10 × 2

age	Age
<dbl>	<dbl>
82	253.16
81	250.89
80	259.63
79	253.86
78	243.73
77	250.80
76	267.61
75	243.53
74	251.99
73	231.43

d) Top 10 ages with more number of strokes.

Here we are using max(stroke) which gives us the maximum strokes and we use group_by(age) in order to get the data frame containing ages and arrange(desc(age)) which gives the output age in descending order and [:10] gives us the output.

```
df4 <- df %>% group_by(age) %>% summarise(Age = max(stroke))

df4 %>%

   arrange(desc(age)) %>%

   slice(1:10)
```

[44]:

```r
#2-d 2nd question top 10 ages with more number of strokes
df4 <- df %>% group_by(age) %>% summarise(Age = max(stroke))
df4 %>%
    arrange(desc(age)) %>%
    slice(1:10)
```

A tibble: 10 × 2

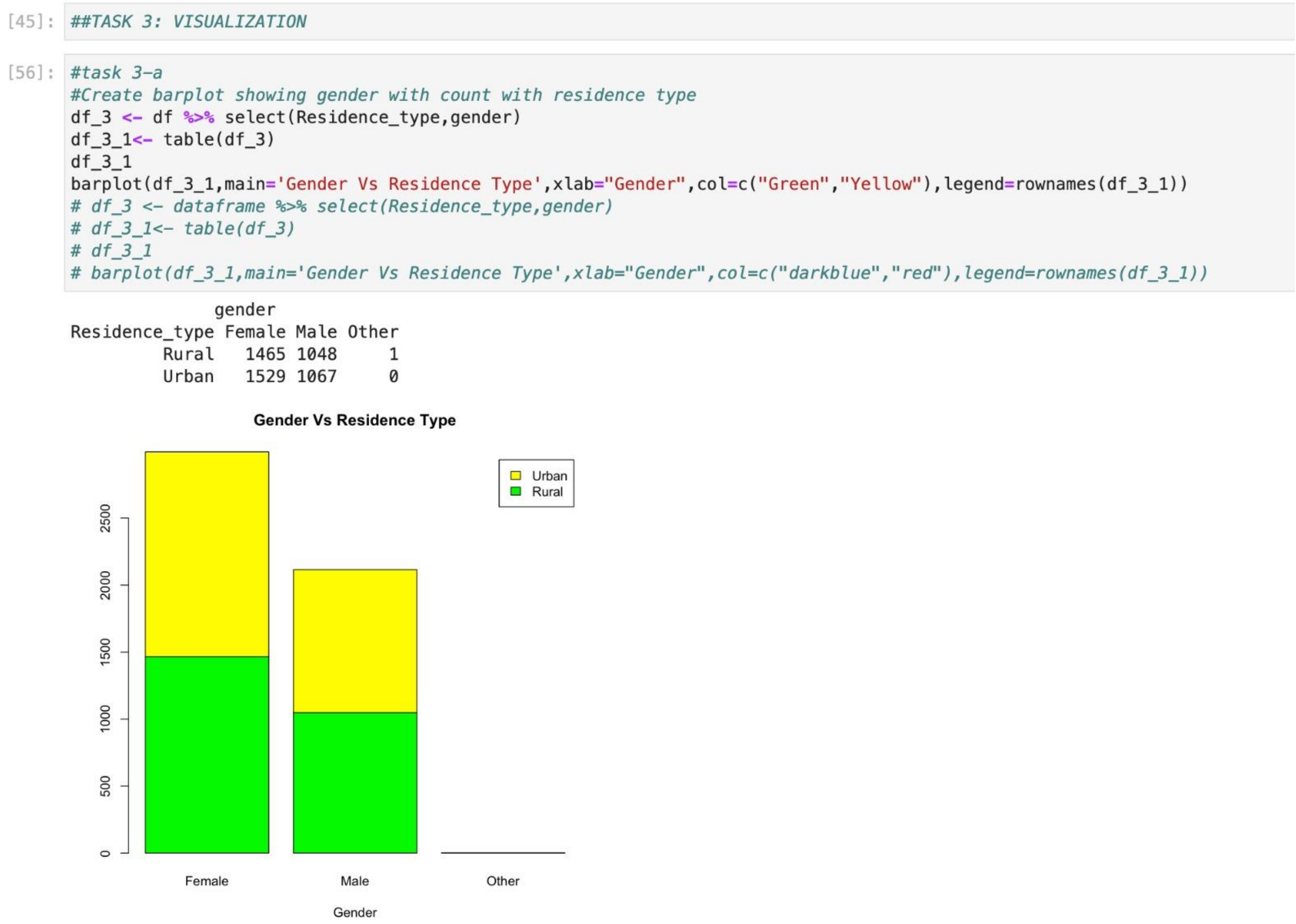
age	Age
<dbl>	<int>
82	1
81	1
80	1
79	1
78	1
77	1
76	1
75	1
74	1
73	1

TASK-3: Data Visualization

a) Create bar plot showing gender with count with residence type

Here we are using the table () which gives us the required data containing residence type and gender. Thus, finally barplot() is used to get the result using the bar plot.

```
Df_3 <- df %>% select(Residence_type,gender)

df_3_1<- table(df_3)

df_3_1

barplot(df_3_1,main='Gender Vs Residence
Type',xlab="Gender",col=c("green","red"),legend=rownames(df_3_1))
```



```
[45]: ##TASK 3: VISUALIZATION
```

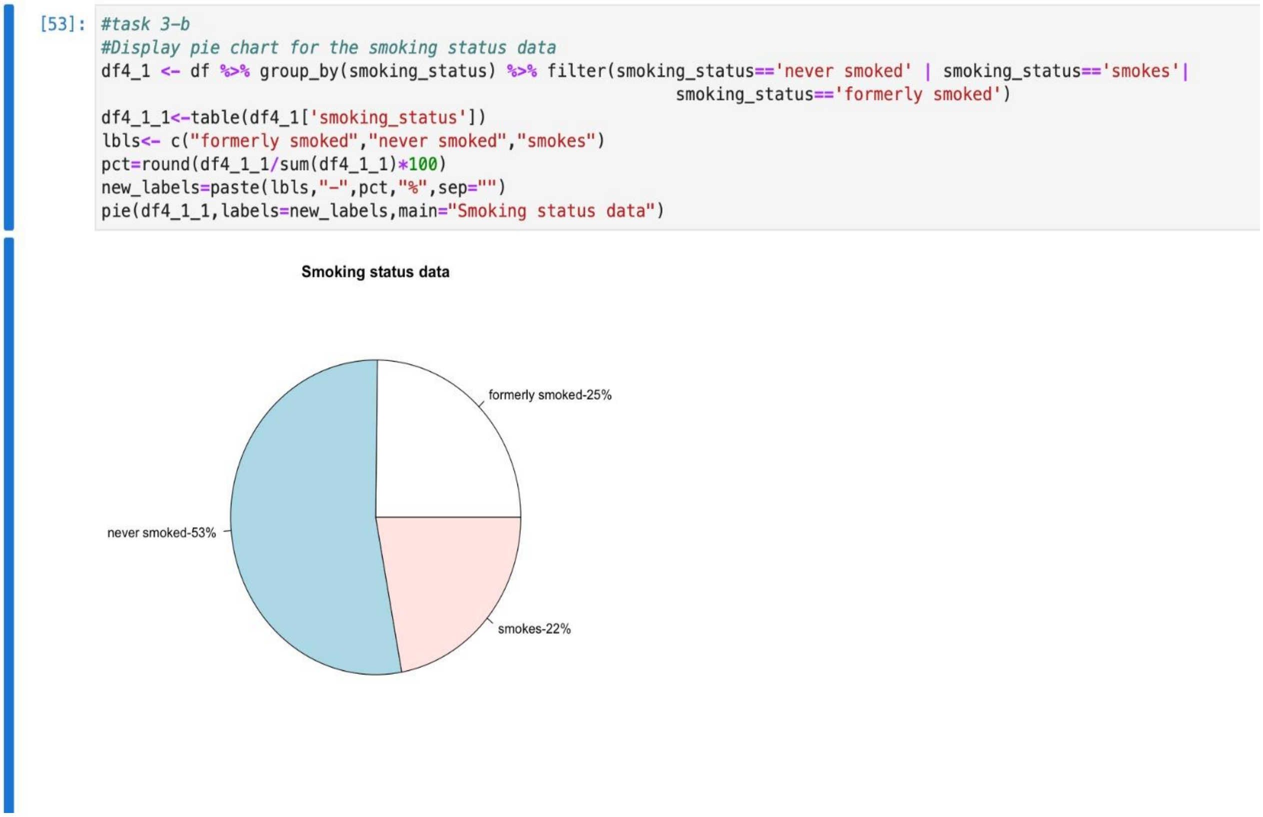
```
[56]: #task 3-a
      #Create barplot showing gender with count with residence type
      df_3 <- df %>% select(Residence_type,gender)
      df_3_1<- table(df_3)
      df_3_1
      barplot(df_3_1,main='Gender Vs Residence Type',xlab="Gender",col=c("Green","Yellow"),legend=rownames(df_3_1))
      # df_3 <- dataframe %>% select(Residence_type,gender)
      # df_3_1<- table(df_3)
      # df_3_1
      # barplot(df_3_1,main='Gender Vs Residence Type',xlab="Gender",col=c("darkblue","red"),legend=rownames(df_3_1))
```

```
              gender
Residence_type Female Male Other
         Rural   1465 1048     1
         Urban   1529 1067     0
```

b) Display pie chart for the smoking status data

Here for getting the pie chart we are using group_by and using filter for smoking status and finally we calculate the percentage and thus by using pie() for getting the pie chart.

```
df4_1 <- df %>% group_by(smoking_status) %>% filter(smoking_status=='never smoked' |
smoking_status=='smokes'| smoking_status=='formerly smoked')

df4_1_1<-table(df4_1['smoking_status'])

lbls<- c("formerly smoked","never smoked","smokes")

pct=round(df4_1_1/sum(df4_1_1)*100)

new_labels=paste(lbls,"-",pct,"%",sep="")

pie(df4_1_1,labels=new_labels,main="Smoking status data")
```



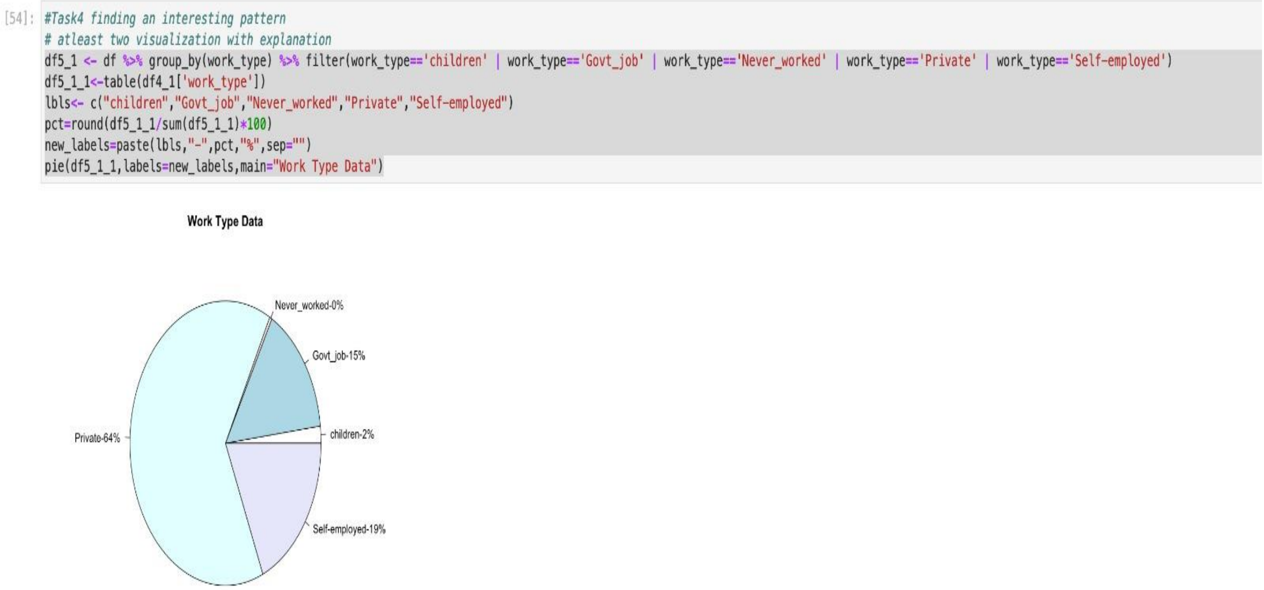
[53]:

```
#task 3-b
#Display pie chart for the smoking status data
df4_1 <- df %>% group_by(smoking_status) %>% filter(smoking_status=='never smoked' | smoking_status=='smokes'|
                                                        smoking_status=='formerly smoked')
df4_1_1<-table(df4_1['smoking_status'])
lbls<- c("formerly smoked","never smoked","smokes")
pct=round(df4_1_1/sum(df4_1_1)*100)
new_labels=paste(lbls,"-",pct,"%",sep="")
pie(df4_1_1,labels=new_labels,main="Smoking status data")
```

Task-4: finding an interesting pattern

At least two visualizations with explanation

Here we are giving the work type data by first group_by(work_type) And then by applying filter thus calculating the percentage work types and finally using pie() to get the pie chart.

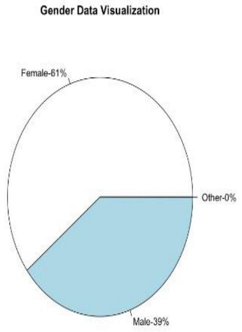


```
[54]: #Task4 finding an interesting pattern
      # atleast two visualization with explanation
      df5_1 <- df %>% group_by(work_type) %>% filter(work_type=='children' | work_type=='Govt_job' | work_type=='Never_worked' | work_type=='Private' | work_type=='Self-employed')
      df5_1_1<-table(df4_1['work_type'])
      lbls<- c("children","Govt_job","Never_worked","Private","Self-employed")
      pct=round(df5_1_1/sum(df5_1_1)*100)
      new_labels=paste(lbls,"-",pct,"%",sep="")
      pie(df5_1_1,labels=new_labels,main="Work Type Data")
```

Here we are giving the gender data visualization by first using group_by on genders and then applying filters for those required genders. Thus, we can get the required percentage of gender data and finally use pie() to get the required pie chart

[55]:

```
df5_1 <- df %>% group_by(gender) %>% filter(gender=='Female' | gender=='Male' | gender=='Other' )
df5_1_1<-table(df4_1['gender'])
lbls<- c("Female","Male","Other")
pct=round(df5_1_1/sum(df5_1_1)*100)
new_labels=paste(lbls,"-",pct,"%",sep="")
pie(df5_1_1,labels=new_labels,main="Gender Data Visualization")
```


Gender Data Visualization
Female-61%
Other-0%
Male-39%