Project Report

Contents

Team members

- Ahmed Ashraf Mohamed
  - ID: 2022446758
- Abdelrhman Mohamed Abdelhady Hodid
  - ID : 2022513643
- Antonuose Gerges Nageh
  - ID : 20221903971

Project Description

Getting data set

importing the library we are going to use

```
library(dplyr)
library(ggplot2)
library(forcats)
library(arules)
library(hrbrthemes)
```

Reading the data and checking the frist 10 rows of it

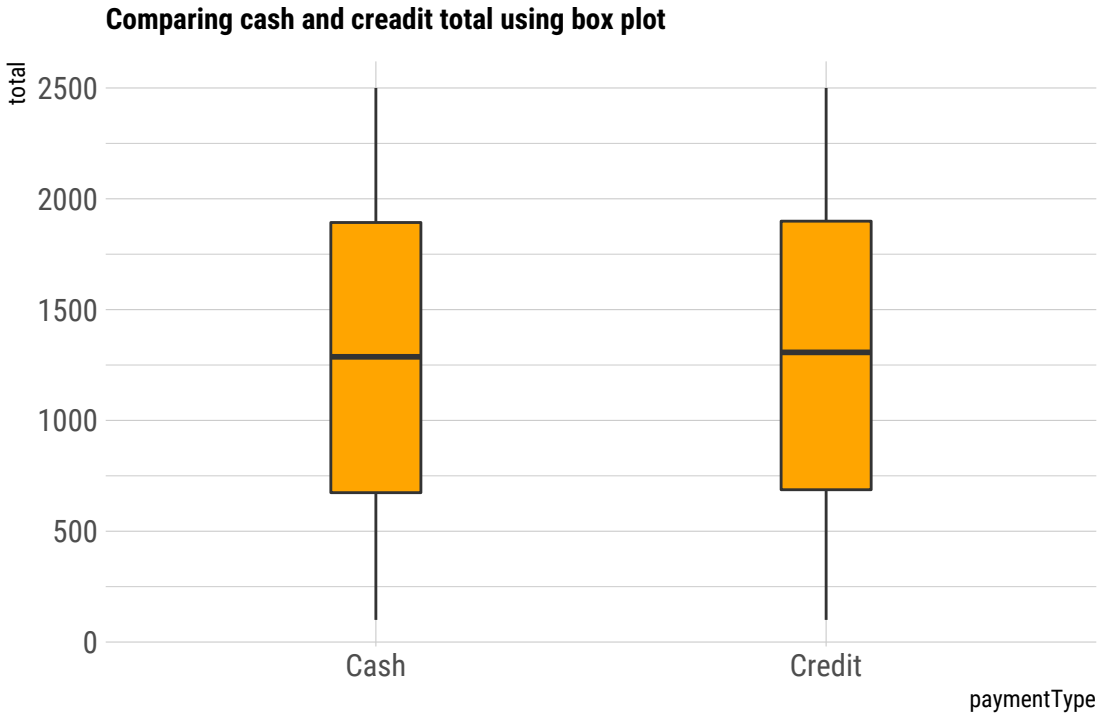
```
dataPath <- readline("Enter the path to the data set : ")
grc <- as_tibble(read.csv(dataPath,stringsAsFactors = FALSE))
# displaying first 10 rows of our data
print(grc,n = 10, width = 80)
```

```
## # A tibble: 9,835 x 8
##    items                     count total   rnd customer   age city   paymentType
##    <chr>                     <int> <int> <int> <chr>    <int> <chr>  <chr>
##  1 citrus fruit,semi-finish~     4  1612     9 Maged       60 Hurgh~ Cash
##  2 tropical fruit,yogurt,co~     3   509    12 Eman        23 Aswan  Cash
##  3 whole milk                    1  2084     8 Rania       37 Dakah~ Cash
##  4 pip fruit,yogurt,cream c~     4   788     8 Rania       37 Dakah~ Cash
##  5 other vegetables,whole m~     4  1182    14 Magdy       36 Sohag  Cash
##  6 whole milk,butter,yogurt~     5  1771     3 Ahmed       30 Giza   Credit
##  7 rolls/buns                    1  2196     7 Huda        39 Gharb~ Cash
##  8 other vegetables,UHT-mil~     5  1657     6 Walaa       29 Cairo  Cash
##  9 pot plants                    1  2373     2 Mohamed     25 Alexa~ Credit
## 10 whole milk,cereals            2   343     5 Shimaa      55 Port ~ Cash
## # ... with 9,825 more rows
```

Visualizing our Data

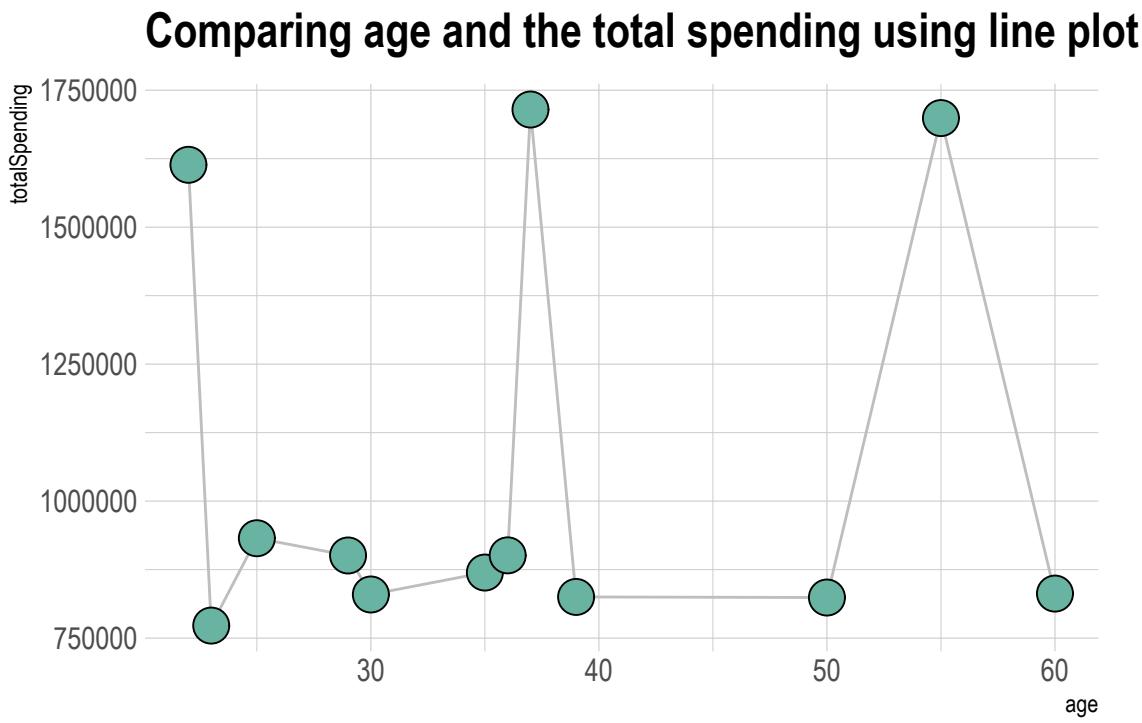
Comparison between cash and creadit total spending using box plot

```
ggplot(
  grc,
  aes(x = paymentType, y = total, fill = paymentType)
) +
  geom_boxplot(width = .2,
               fill = "orange",
               outlier.color = "orange",
               outlier.size = 2)+
  theme_ipsum_rc() +
  theme(
    legend.position="none",
    plot.title = element_text(size=11)
  ) +
  ggtitle("Comparing cash and creadit total using box plot")
```

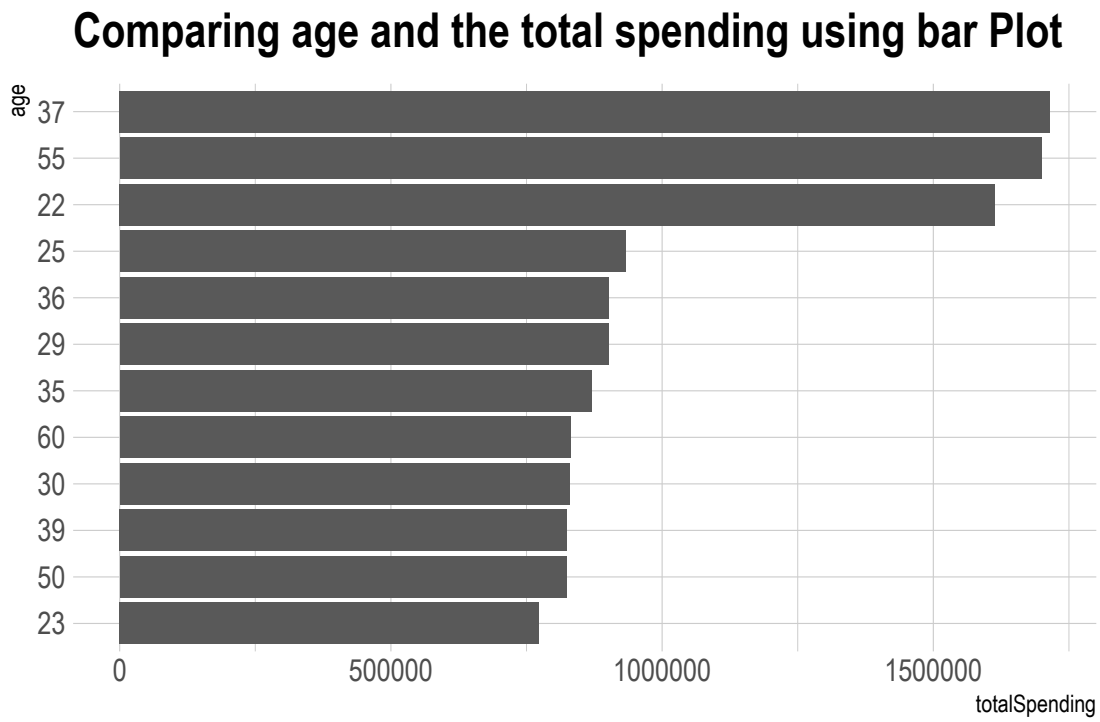


Compare each age and sum of total spending.

```
grc_age <- select(grc,age,total)
grc_age <- group_by(grc_age, age)
grc_age <- summarise(grc_age,totalSpending = sum(total))
ggplot(
  grc_age,
  aes(x = age, y = totalSpending)) +
  geom_line( color="gray") +
  geom_point(shape=21, color="black", fill="#69b3a2", size=6) +
  theme_ipsum() +
  ggtitle("Comparing age and the total spending using line plot")
```




```
grc_age <- mutate(grc_age,age = fct_reorder(as.factor(age),totalSpending))
ggplot(
  grc_age,
  aes(x = age, y = totalSpending)) +
  geom_col() +
  coord_flip() +
  theme_ipsum() +
  ggtitle("Comparing age and the total spending using bar Plot")
```



Generating of association rules

Brief explanition of Apriori algorithm for generating the rules

Apriori algorithm is an iterative approach for discovering the most frequent item sets. The frequent item sets generated by the algorithm can be used to determine association rules that highlight general trends in the data-set, it is especially useful in the analysis of super-market items in our data set

Implementing the alogrithm

Preparing the data for generating the association rules

```
tdata<- strsplit(as.vector(grc$items), ',')
tdata <- transactions(tdata)
```

Reading both minimum support and minimum confidance from the user

```
min_support <- as.numeric(readline("Enter the minimum Support : "))
min_conf <- as.numeric(readline("Enter the minimum Confidance : "))
```

implementing the algorithm using the built-in function

```
apriori_rules <- apriori(
  tdata,
  parameter = list(supp = min_support, conf = min_conf, minlen = 2))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
##         0.5    0.1    1 none FALSE            TRUE       5    0.01      2     10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 98
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [88 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [15 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(apriori_rules,linebreak = T)
```

```
##      lhs                     rhs                   support confidence   coverage     lift count
## [1]  {curd,
##       yogurt}             => {whole milk}       0.01006609  0.5823529 0.01728521 2.279125    99
## [2]  {butter,
##       other vegetables}   => {whole milk}       0.01148958  0.5736041 0.02003050 2.244885   113
## [3]  {domestic eggs,
##       other vegetables}   => {whole milk}       0.01230300  0.5525114 0.02226741 2.162336   121
## [4]  {whipped/sour cream,
##       yogurt}             => {whole milk}       0.01087951  0.5245098 0.02074225 2.052747   107
## [5]  {other vegetables,
##       whipped/sour cream} => {whole milk}       0.01464159  0.5070423 0.02887646 1.984385   144
## [6]  {other vegetables,
##       pip fruit}          => {whole milk}       0.01352313  0.5175097 0.02613116 2.025351   133
## [7]  {citrus fruit,
##       root vegetables}    => {other vegetables} 0.01037112  0.5862069 0.01769192 3.029608   102
## [8]  {root vegetables,
##       tropical fruit}     => {other vegetables} 0.01230300  0.5845411 0.02104728 3.020999   121
## [9]  {root vegetables,
##       tropical fruit}     => {whole milk}       0.01199797  0.5700483 0.02104728 2.230969   118
## [10] {tropical fruit,
##       yogurt}             => {whole milk}       0.01514997  0.5173611 0.02928317 2.024770   149
## [11] {root vegetables,
##       yogurt}             => {other vegetables} 0.01291307  0.5000000 0.02582613 2.584078   127
## [12] {root vegetables,
##       yogurt}             => {whole milk}       0.01453991  0.5629921 0.02582613 2.203354   143
## [13] {rolls/buns,
##       root vegetables}    => {other vegetables} 0.01220132  0.5020921 0.02430097 2.594890   120
## [14] {rolls/buns,
##       root vegetables}    => {whole milk}       0.01270971  0.5230126 0.02430097 2.046888   125
## [15] {other vegetables,
##       yogurt}             => {whole milk}       0.02226741  0.5128806 0.04341637 2.007235   219
```