

1)

الطالب : احمد توفيق امين ابراهيم

الرقم الجامعي :

اولا:عمل مقارنة بين

cash و credit

شهير جدا يستخدم فى table وهذا نوع (contingency table) اولاً:لعمل هذه المقارنة تم استخدام الأداة فى هيئة جدول colum أو vector عرض القيم بداخل

piechart و barblot ثانياً:مقارنة بواسطة

عمل مقارنة بين كل عمر ومجموع الرواتب لهذا العمر(٢)

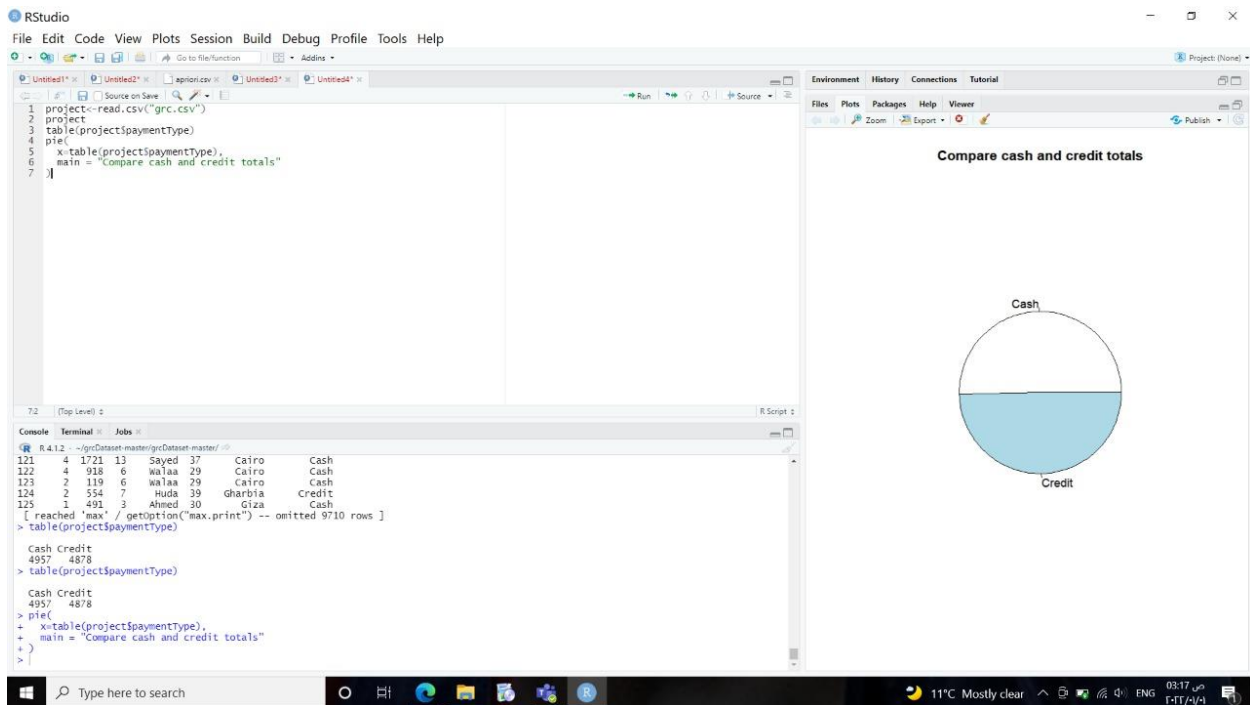
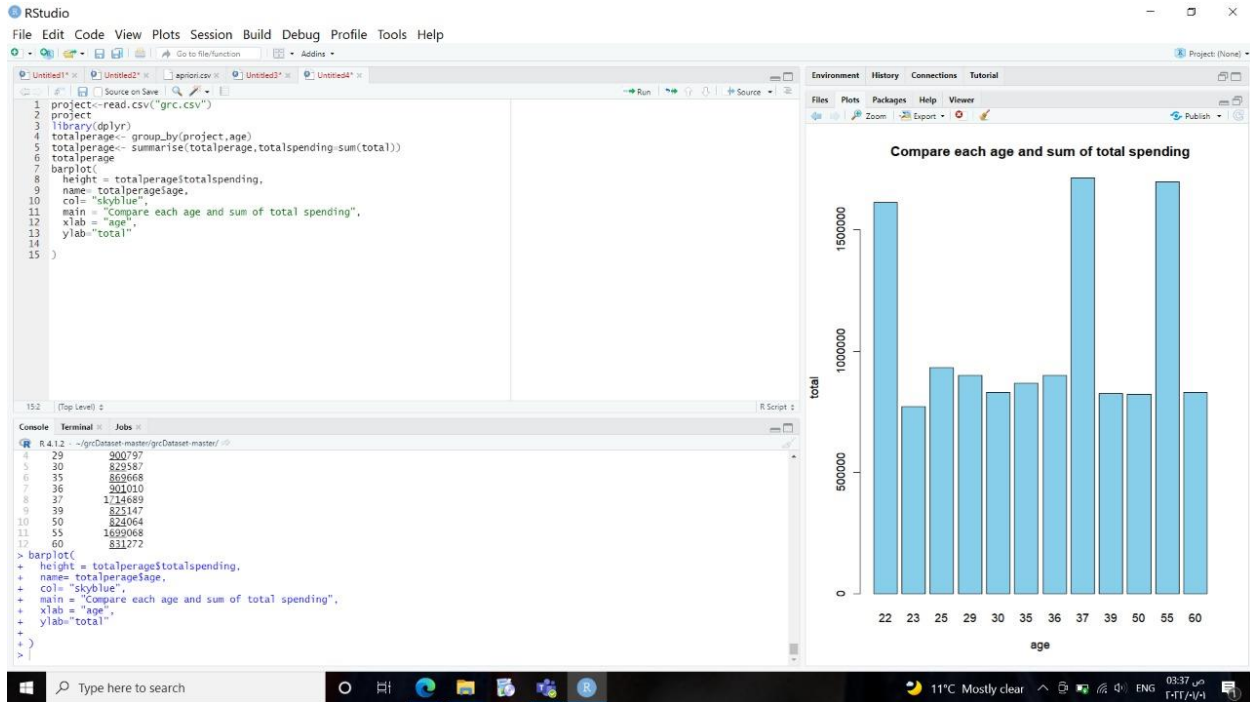
ضرورية فى هذه المقارنة بسبب احتوائها package وهذه dplyr تسمى package اولاً:يجب استخدام،

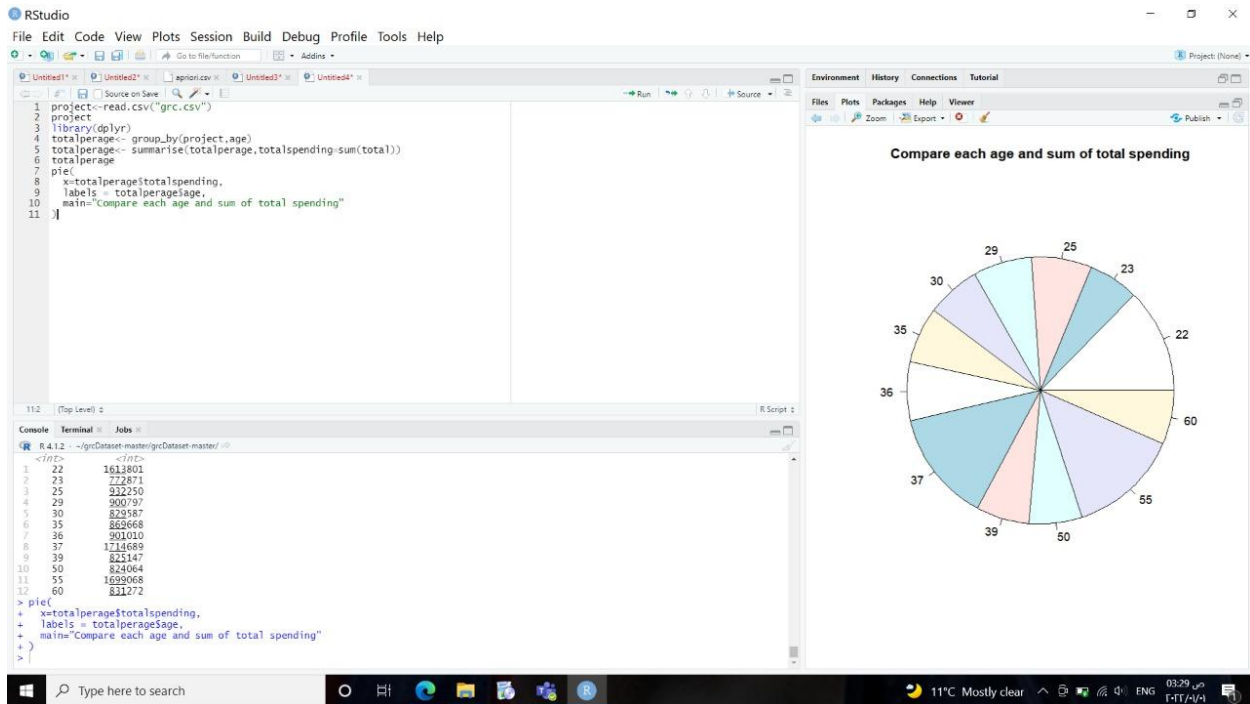
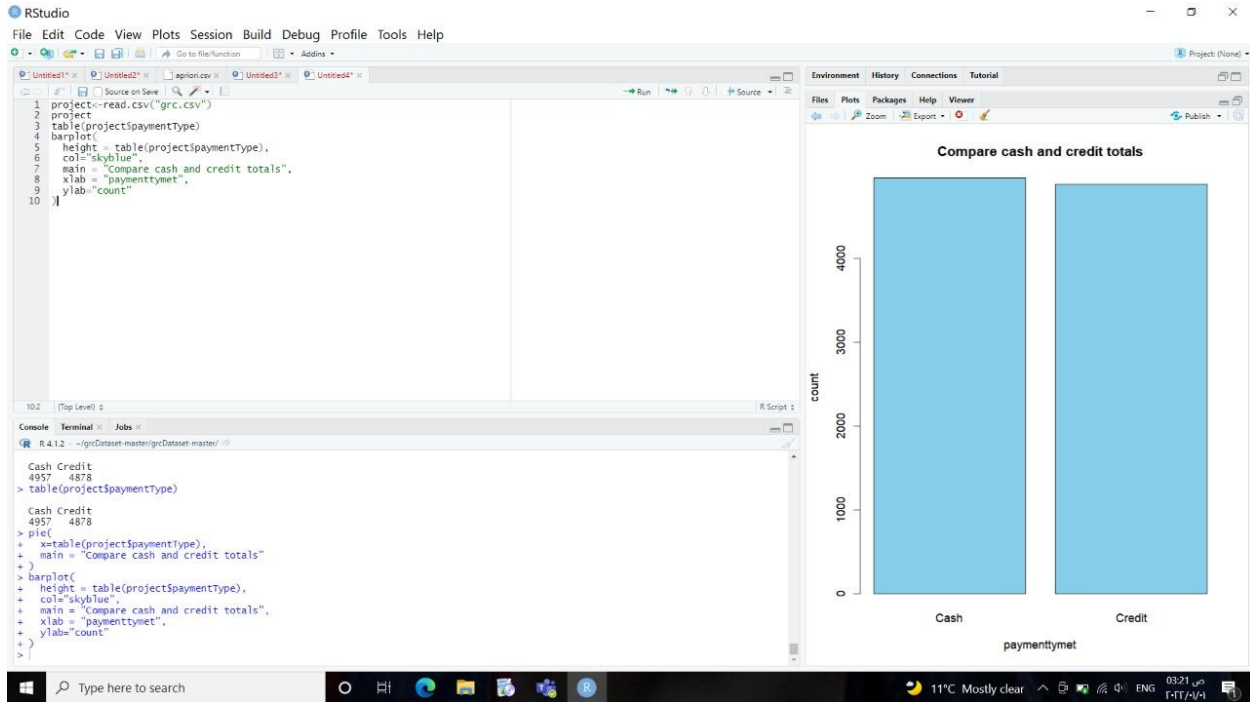
(group_by, summaries) على دوال

تقوم بتقسيم ال group_by وبداخل هذا المتغير يتم استخدام دالة totalperage ثم بعمل متغير يسمى إلى مجموعات حسب الاعمار data frame

من total spending يسمى totalpperage فى إضافة عمود جديد إلى summaries ثم استخدام دالة data frame فى total خلال جمع عمود ال

piechart,barblot ثم المقارنة بواسطة





2)

الطالب : عبدالرحمن محمد محمد جمعه

الرقم الجامعي : 20221374057

١). عرض إجمالي الإنفاق لكل مدينة ورتبها تنازلياً:

شهير جدا يستخدم table وهذا نوع (contingency table) اولا: لعرض هذا الاجمالي تم استخدام الأداة في هيئة جدول colum أو vector في عرض القيم بداخل piechart و barplot ثانيا: مقارنة بواسطة

عمل مقارنة بين كل مدينة ومجموع الانفاق لهذا المدينة(٢)

ضرورية في هذه المقارنة بسبب احتوائها package وهذه dplyr تسمى package اولا: يجب استخدم،

(group_by, summaries) على دوال

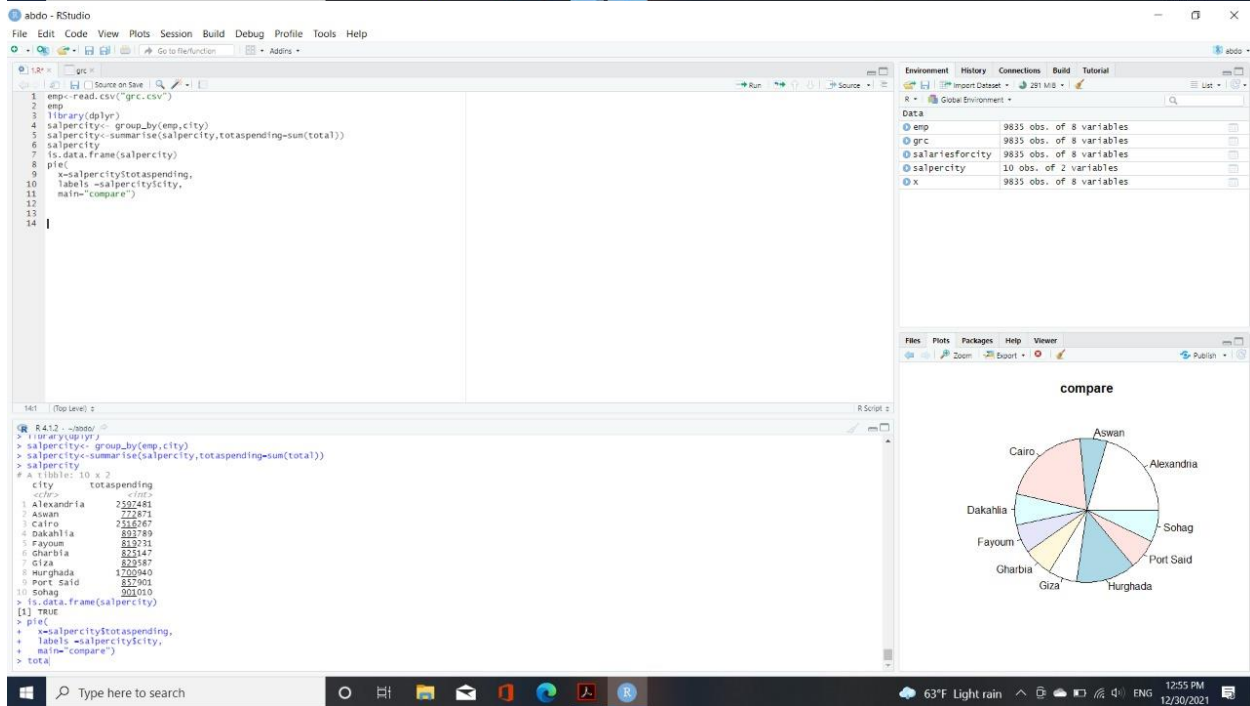
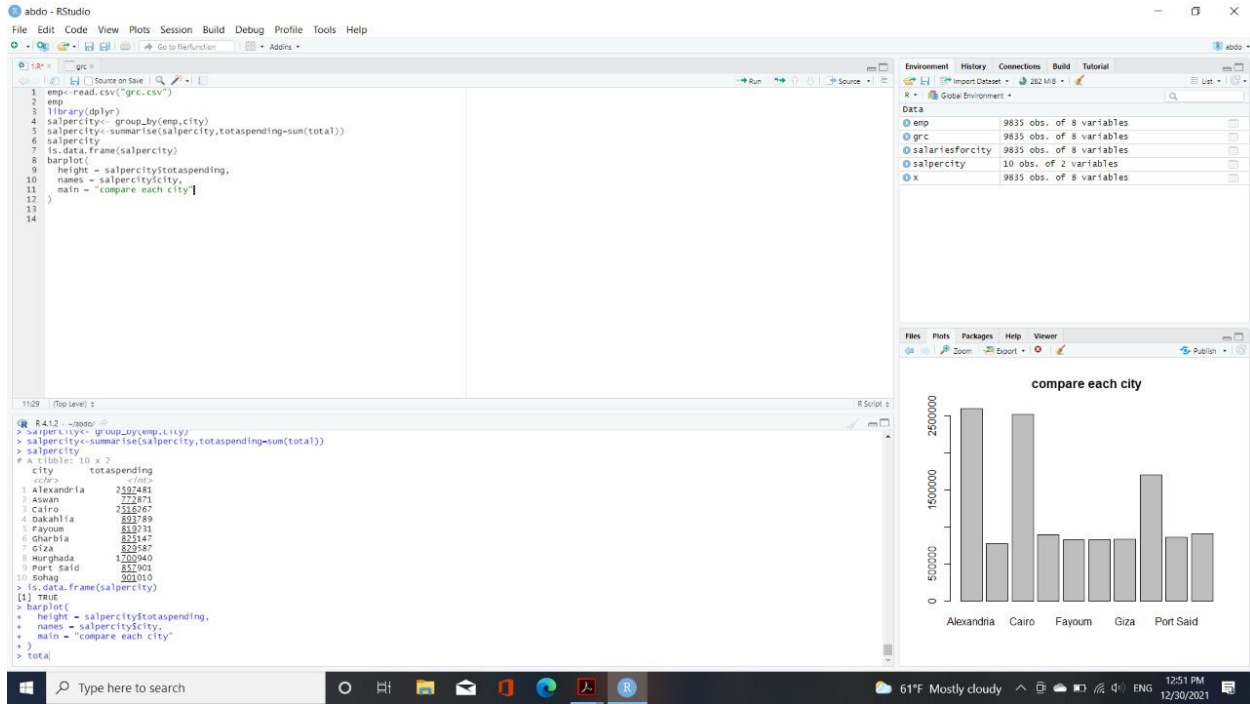
data تقوم بتقسيم ال group_by وبداخل هذا المتغير يتم استخدام دالة salerpercicity بعمل متغير يسمى إلى مجموعات حسب المدينة frame

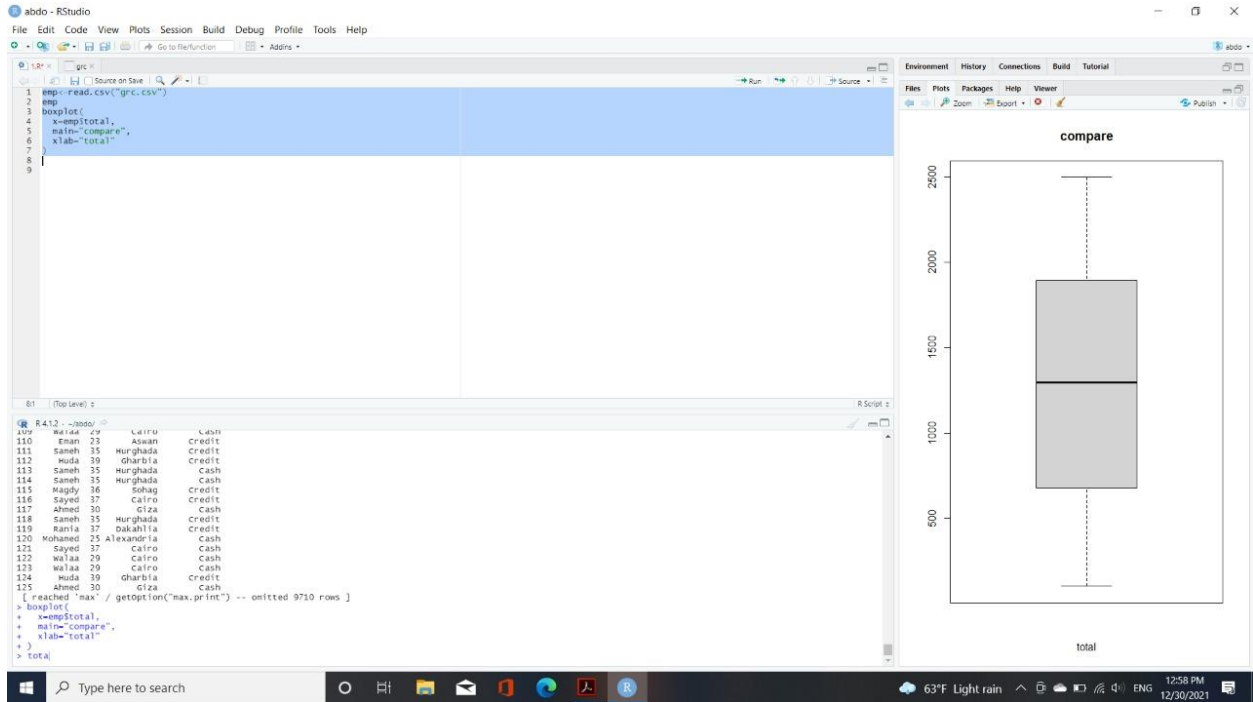
من total spending يسمى salerpercicity في إضافة عمود جديد إلى summaries ثم استخدام دالة data frame في total خلال جمع عمود ال

tpiechart,barblo ثم المقارنة بواسطة

٣) عرض توزيع اجمالي الانفاق

maximum و minimum وهو امر يستخدم لعرض boxplot لعمل هذا التوزيع تم اسخدام متغير يسمى main و





3)

الطالب : زياد محمد سليمان حسن

الرقم الجامعي : 20221041326

قمت بتحميل ملف Csv من خلال استدعاءها وبعدها قمت باستدعاء حزمه شهيره

تسمى Dplyr لتساعدنا فى حل المشكله بعدها قمت باخذ البيانات التى قام الطالبان عبدالرحمن

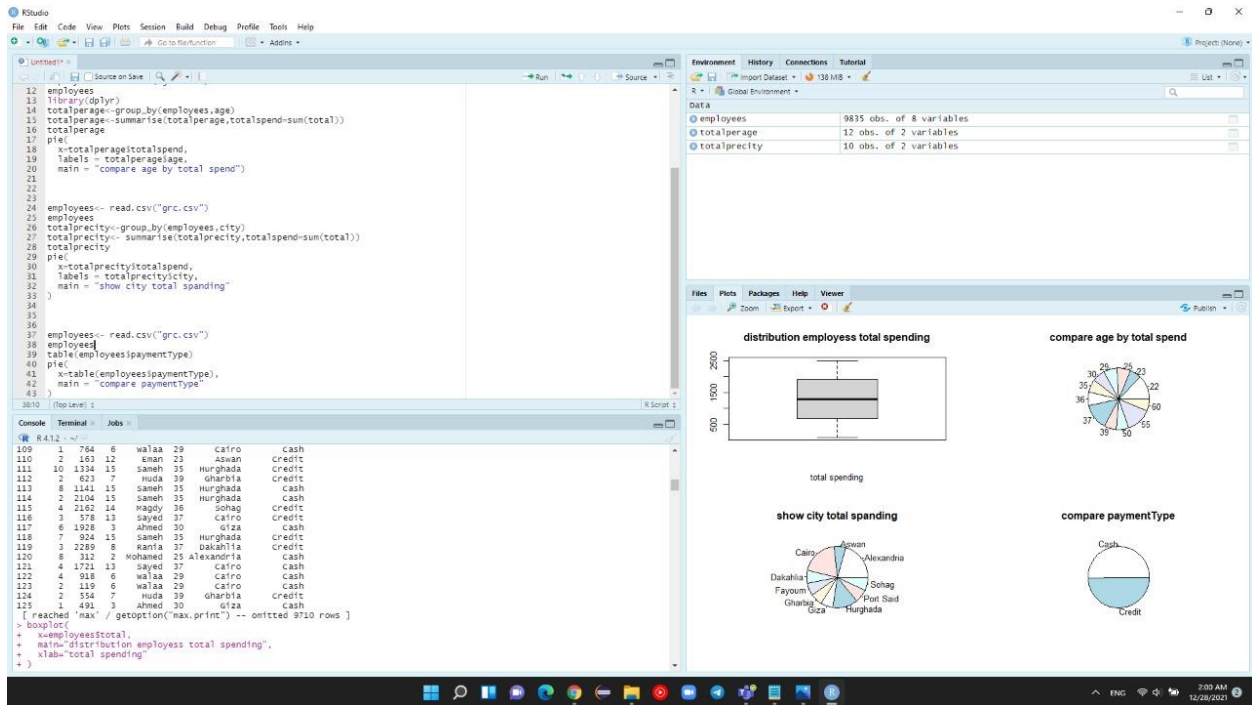
واحمد توفيق لعمل Dashboard للبيانات التى قامو بعمل لها Data visualization

وقمت باستخدام داله nfrow وهذه الداله تقوم بتقسيم الشاشة الى اقسام لوضع

البيانات التحليليه المختلفه مع بعضها البعض وقمت باعطائها (2,2) ليقوم بتقسيم الشاشة لاربع اجزاء

ووضعت بها البيانات كما قمت بمساعده زميلى عبدالرحمن عباس فى السؤال الرابع المختص

بالتحليل العنقودى للبيانات



4)

الطالب : عبدالرحمن عباس ابراهيم عباس

الرقم الجامعي : 20221375205

```
csv_file <- read.csv("D:/grc.csv")
```

```
library("dplyr")
```

اولا قمت باستدعاء ملف الذي يوجد به البيانات وبعدها قمت

باستدعاء مجموعه مساعده تسمى dplyr.

```
cluster_num <- as.integer
```

```
(readline(prompt = "enter clusters num(2:4): "))
```

بعدها قومت بعمل متغير يسمى cluster_num

لاقوم بحفظ به عدد العناقيد التي يريد المستخدم عملها

```
if(cluster_num >= 2 && cluster_num <= 4)
```

```
{
```

بعدها قومت بعمل داله شرطيه للتأكد ان

المستخدم سيدخل رقم مابين الواحد والخمسه

ووضعت داخل الداله الشرطيه البرنامج الخاص بي

```
new_data <- select(csv_file , age , total)
```

قمت بعمل متغير اسمه new_data

ليقوم بحفظ العمر والنتائج من داخل جدول البيانات من خلال

داله تسمى select تقوم بتحديد عض البيانات من داخل جدول البيانات

```
clusters <- kmeans(new_data , cluster_num)
```

بعدها قمت بعمل متغير اسمه clusters لادخل به البيانات الخارجه من

داله kmeans التى تقوم بالتحليل العنقودى للبيانات فقامت بتحليل new_data وقامت بوضعهم فى عدد من المجموعات قمت بتحديددها من خلال cluster_num

```
clusters <- clusters$cluster
```

بعدها قمت بفلتره البيانات من خلال اداه cluster التى تقوم بتحديد الجزى الذى تم تحليله فقط

```
clusters<-data.frame
```

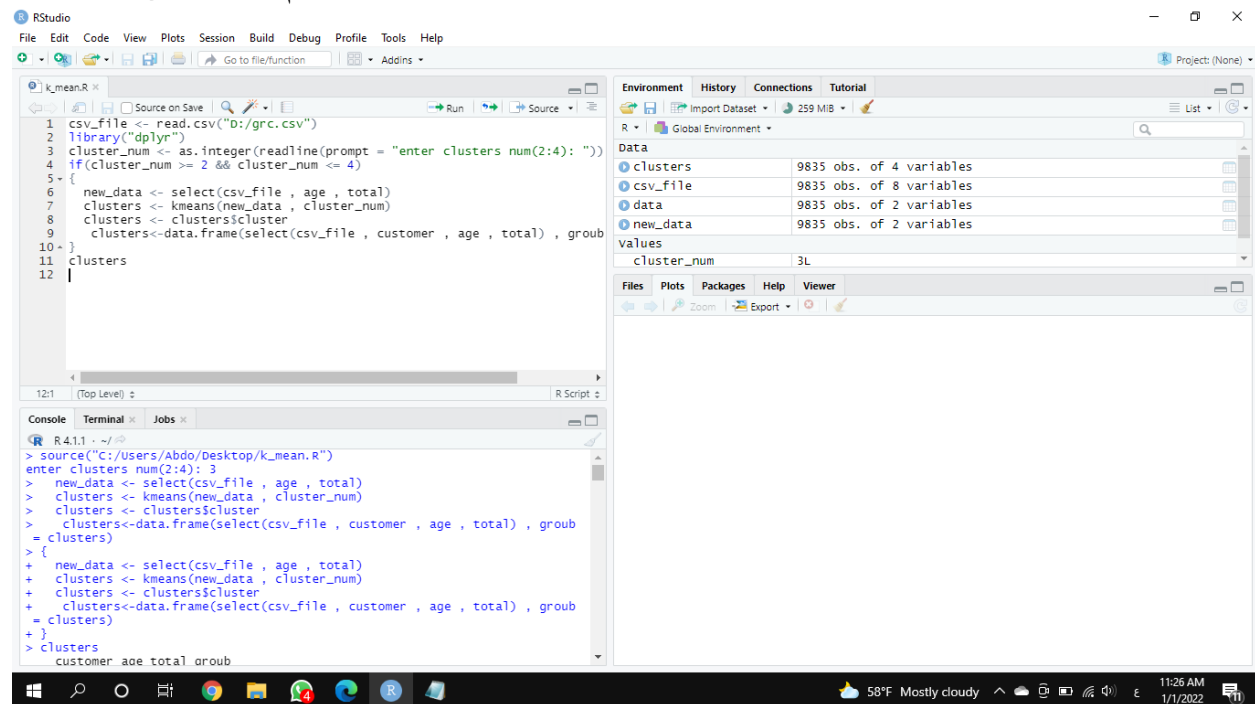
```
(select(csv_file , customer , age , total) , groub = clusters)
```

بعدها قمت باستخدام داله data.frame لتقوم بعمل ترتيب للبيانات من خلال اعطائها البيانات المتطلبه

```
(select(csv_file , customer , age , total) , groub = clusters)
```

ووضعها داخل متغير clusters ومن خلال طباعه

المتغير نجد ان البيانات الخاصه بنا تم تقسيمها الى مجموعات



The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains R code for reading a CSV file, performing k-means clustering, and creating a data frame.
- Environment:** Lists objects in the global environment: clusters (9835 obs. of 4 variables), csv_file (9835 obs. of 8 variables), data (9835 obs. of 2 variables), and new_data (9835 obs. of 2 variables).
- Console:** Shows the output of the R code execution, including the prompt for cluster number and the resulting data frame structure.

```
1 csv_file <- read.csv("D:/grc.csv")
2 library("dplyr")
3 cluster_num <- as.integer(readline(prompt = "enter clusters num(2:4): "))
4 if(cluster_num >= 2 && cluster_num <= 4)
5 {
6   new_data <- select(csv_file , age , total)
7   clusters <- kmeans(new_data , cluster_num)
8   clusters <- clusters$cluster
9   clusters<-data.frame(select(csv_file , customer , age , total) , groub
10 ~ }
11 clusters
12 |
```

Console Output:

```
> source("C:/Users/Abdo/Desktop/k_mean.R")
enter clusters num(2:4): 3
> new_data <- select(csv_file , age , total)
> clusters <- kmeans(new_data , cluster_num)
> clusters <- clusters$cluster
> clusters<-data.frame(select(csv_file , customer , age , total) , groub
= clusters)
> {
+ new_data <- select(csv_file , age , total)
+ clusters <- kmeans(new_data , cluster_num)
+ clusters <- clusters$cluster
+ clusters<-data.frame(select(csv_file , customer , age , total) , groub
= clusters)
+ }
> clusters
  customer age total groub
```

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins Project: (None)

k_mean.R

```
1 csv_file <- read.csv("D:/grc.csv")
2 library("dplyr")
3 cluster_num <- as.integer(readline(prompt = "enter clusters num(2:4): "))
4 if(cluster_num >= 2 && cluster_num <= 4)
5 {
6   new_data <- select(csv_file, age, total)
7   clusters <- kmeans(new_data, cluster_num)
8   clusters <- clusters$cluster
9   clusters<-data.frame(select(csv_file, customer , age , total) , group
10 }
11 clusters
12 |
```

Environment History Connections Tutorial

R Global Environment

Data

clusters	9835 obs. of 4 variables
csv_file	9835 obs. of 8 variables
data	9835 obs. of 2 variables
new_data	9835 obs. of 2 variables

Values

cluster_num 3L

Files Plots Packages Help Viewer

Console Terminal Jobs

R 4.1.1 ~ /

```
> clusters
  customer age total group
1    Maged  60 1612     1
2      Eman  23  509     3
3    Rania  37 2084     2
4    Rania  37  788     3
5    Magdy  36 1182     1
6    Ahmed  30 1771     2
7     Huda  39 2196     2
8    Walaa  29 1657     1
9  Mohamed  25 2373     2
10 Shimaa  55  343     3
11 Mohamed  25 1381     1
12 Farida  22 1965     2
13 Hanan  22  784     3
14 Hanan  22  784     3
```

12:11 (Top Level) R Script

58°F Mostly cloudy 11:26 AM 1/1/2022

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins Project: (None)

k_mean.R

```
1 csv_file <- read.csv("D:/grc.csv")
2 library("dplyr")
3 cluster_num <- as.integer(readline(prompt = "enter clusters num(2:4): "))
4 if(cluster_num >= 2 && cluster_num <= 4)
5 {
6   new_data <- select(csv_file, age, total)
7   clusters <- kmeans(new_data, cluster_num)
8   clusters <- clusters$cluster
9   clusters<-data.frame(select(csv_file, customer , age , total) , group
10 }
11 clusters
12 |
```

Environment History Connections Tutorial

R Global Environment

Data

clusters	9835 obs. of 4 variables
csv_file	9835 obs. of 8 variables
data	9835 obs. of 2 variables
new_data	9835 obs. of 2 variables

Values

cluster_num 3L

Files Plots Packages Help Viewer

Console Terminal Jobs

R 4.1.1 ~ /

```
12 Farida  22 1965     2
13 Hanan  22  784     3
14 Huda  39 1001     1
15 Sayed  37 1579     1
16 Rania  37  585     3
17 Shimaa  55  184     3
18 Eman  23 1737     2
19 Walaa  29  184     3
20 Huda  39  469     3
21 Shimaa  55  408     3
22 Huda  39 2252     2
23 Ahmed  30 1538     1
24 Maged  60 1215     1
25 Adel  50 1762     2
26 Sameh  35 2384     2
```

12:11 (Top Level) R Script

58°F Mostly cloudy 11:26 AM 1/1/2022

الطالب : محمد محمود محمد عبدالفتاح(5)

الرقم الجامعي : 20221374157

(1) library(arules)

transaction اسمها Data type بها package arules نستخدم

(apriori algorithm) مخصصه ل

datasetpath<- readline("Enter the path: ")

user من ال input عشان تسمح لي اخذ function readline استخدمنا ال

Data الملف الموجود عليه ال path حتي يستطيع يضع

X <- read.csv(datasetpath)

و X بإسم viriable عملت

Function read.csv

grc لكي تستطيع قراءة ملف

[10:27 AM, 1/1/2022] Mohamed Mahmoud: min_support <-

as.numeric(readline("Enter the min_support between [1:0.001]: "))

min_confidince <- as.numeric(readline("Enter the min_confidince between
[1:0.001] : "))

function و استخدمنا variables ك min support and min confidence هنا وضعنا

as.numeric(readline(""))

user رقميا من ال input حتي تسمح لي اخذ

ليضع القيم المناسبه

: if (min_support>=0.001 & min_support<=1 & min_confidince >=0.001 &
min_confidince<=1)

{

```
}else  
  print("[invaild] Check your numeric input")
```

حين ادخال قيمتيهما يشترط min support and min confidence لتحقيق شرط ان if استخدمنا حالة (1 : 0.001) ان يكون الرقم بين

حينما لا يتحقق الشرط يطبع له else و استخدمنا

```
[invaild] Check your numeric input
```

```
: write.table(X $items,"apriori.txt",row.names= FALSE ,col.names = FALSE , quote=  
FALSE )
```

grc و ياخذ من ملف txt يجعل الملف المدرج بامتداد

(" ") مع الغاء الزيادات بالصفوف و الاعمدة و علامه apriori و تسميته (items) العمود الاول

```
tdata <- read.transactions( "apriori.txt" , sep=",")
```

user الذي يضعه ال path الموجوده في ال transaction حتي يقرأ ال variable tdata وضعناها في (,) هي transaction التي بين ال sep و ال

```
apriori_rules <- apriori(tdata,
```

```
parameter = list(supp = min_support, conf = min_confidince , minlen=2 ))
```

```
inspect(apriori_rules)
```

apriori اسمها function استخدمت

algorithm بتاعت ال options بال list عباره عن parameter و بها (tdata) transaction لتأخذ

واحد rules اقل شيء ٢ متطلعش rules استخدمناها عشان تطلع لي minlen=2 و

للعرض inspect(apriori_rules) ومن ثم

RStudio interface showing a script editor with R code for data processing and association rule mining. The code includes reading a CSV file, validating input, and using the `arules` package to generate rules from transaction data.

```
1 library(arules)
2 datasetpath<- readline("Enter the path: ")
3 X <- read.csv(datasetpath)
4 min_support <- as.numeric(readline("Enter the min_support between [0.001:1] : "))
5 min_confidence <- as.numeric(readline("Enter the min_confidence between [0.001:1] : "))
6 if (min_support>=0.001 & min_support<=1 & min_confidence >=0.001 & min_confidence<=1 )
7 {
8   write.table(X $items,"apriori.txt",row.names= FALSE ,col.names = FALSE , quote= FALSE )
9   tdata <- read.transactions( "apriori.txt" , sep=",")
10  apriori_rules <- apriori(tdata,
11    parameter = list(supp = min_support, conf = min_confidence , minlen=2 ))
12  inspect(apriori_rules)
13 } else
14   print("[Invalid] check your numeric input")
```

The Environment pane shows loaded objects: `apriori` (15295 obs. of 4 variables), `apriori_rules` (Formal class rules), `data` (9835 obs. of 8 variables), `tdata` (Formal class transactions), and `X` (9835 obs. of 8 variables). The Values pane shows `datasetpath` as "E:/project 05/grc.csv", `min_confidence` as NA_real_, and `min_support` as 0.03.

The Files pane shows the project structure. The Packages pane lists installed and available user libraries, including `arules`, `arulesViz`, `askpass`, `base64enc`, `cellranger`, `class`, `cli`, `cluster`, `colorspace`, `cpp11`, `crayon`, `crossstalk`, `curl`, `data.table`, `digest`, `dplyr`, `DT`, and `ellipsis`.

RStudio interface showing the execution of the R script. The console displays the output of the script, including the source file path, library loading, and the results of the `inspect` function.

```
> source("E:/project 05/project 18.R", echo=TRUE)
> library(arules)
> datasetpath<- readline("Enter the path: ")
Enter the path: E:/project 05/grc.csv
> X <- read.csv(datasetpath)
> min_support <- as.numeric(readline("Enter the min_support between [0.001:1] : "))
Enter the min_support between [0.001:1] : 0.03
> min_confidence <- as.numeric(readline("Enter the min_confidence between [0.001:1] : "))
Enter the min_confidence between [0.001:1] : 0.23
```

The Environment pane shows the same loaded objects as the first screenshot. The Files pane shows the project structure. The Packages pane lists installed and available user libraries, including `arules`, `arulesViz`, `askpass`, `base64enc`, `cellranger`, `class`, `cli`, `cluster`, `colorspace`, `cpp11`, `crayon`, `crossstalk`, `curl`, `data.table`, `digest`, `dplyr`, `DT`, and `ellipsis`.

Windows Desktop with RStudio and Notepad++

Notepad++ Content:

```
apriori - Notepad
File Edit Format View Help
Citrus fruit,semi-finished bread,margarine,ready soups
tropical fruit,yogurt,coffee
whole milk
pip fruit,yogurt,cream cheese ,meat spreads
other vegetables,whole milk,condensed milk,long life bakery product
whole milk,butter,yogurt,rice,abrasive cleaner
rolls/buns
other vegetables,UMF-milk,rolls/buns,bottled beer,liquor (appetizer)
pot plants
whole milk,cereals
tropical fruit,other vegetables,white bread,bottled water,chocolate
citrus fruit,tropical fruit,whole milk,butter,curd,yogurt,flour,bottled water,dishes
beef
frankfurter,rolls/buns,soda
chicken,tropical fruit
butter,sugar,fruit/vegetable juice,newspapers
fruit/vegetable juice
packaged fruit/vegetables
chocolate
specialty bar
other vegetables
butter milk,pastry
whole milk
tropical fruit,cream cheese ,processed cheese,detergent,newspapers
tropical fruit,root vegetables,other vegetables,frozen dessert,rolls/buns,flour,sweet spreads,salty snack,waffles,candy,bathroom cleaner
bottled water,canned beer
yogurt
sausage,rolls/buns,soda,chocolate
other vegetables
brown bread,soda,fruit/vegetable juice,canned beer,newspapers,shopping bags
yogurt,beverages,bottled water,specialty bar
hamburger,meat,other vegetables,rolls/buns,spices,bottled water,hygiene articles,napkins
root vegetables,other vegetables,whole milk,beverages,sugar
pork,berries,other vegetables,whole milk,whipped/sour cream,artif. sweetener,soda,abrasive cleaner
beef,grapes,detergent
pastry,soda
fruit/vegetable juice
```

RStudio Console Output:

```
R> library(arules)
R> datapath<- readline("Enter the path: ")
Enter the path: E:/project 05/grc.csv
R> X <- read.csv(datapath)
R> min_support <- as.numeric(readline("Enter the min_support between [0.001:1] : "))
Enter the min_support between [0.001:1] : 0.03
R> min_confidence <- as.numeric(readline("Enter the min_confidence between [0.001:1] : "))
Enter the min_confidence between [0.001:1] : 0.25
R> if (min_support>=0.001 & min_support<=1 & min_confidence>=0.001 & min_confidence<=1)
+ {
+   write.table(X $items,"apriori.txt",row.names= FALSE .... [TRUNCATED]
+ }
R> apriori
Parameter specification:
confidence minval snax areal avar originalSupport maxtime support minlen maxlen target ext
0.25 0.1 1 none FALSE TRUE 5 0.03 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: 295
set item appearances ... [0 item(s)] done [0.00s].
set transactions ... [169 item(s), 9835 transaction(s)] done [0.01s].
sorting and recoding items ... [44 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [13 rule(s)] done [0.00s].
creating SA object ... done [0.00s].
[1] (whipped/sour cream) => (whole milk) support confidence coverage lift count
[2] (pip fruit) => (whole milk) 0.03223183 0.4496454 0.07168277 1.759754 317
[3] (pastry) => (whole milk) 0.0324860 0.3737143 0.08896797 1.462587 327
[4] (citrus fruit) => (whole milk) 0.03050330 0.3685504 0.08276963 1.442377 300
[5] (sausage) => (rolls/buns) 0.03060498 0.3257576 0.09395018 1.771048 301
[6] (bottled water) => (whole milk) 0.03426706 0.3209476 0.11052364 1.26940 338
[7] (tropical fruit) => (other vegetables) 0.03589222 0.3420343 0.10493137 1.767790 353
[8] (tropical fruit) => (whole milk) 0.04229792 0.4031008 0.10493137 1.577595 416
[9] (root vegetables) => (other vegetables) 0.04718180 0.4347015 0.10899847 2.246605 446
[10] (root vegetables) => (whole milk) 0.04890696 0.4486940 0.10899847 1.758031 481
[11] (yogurt) => (other vegetables) 0.04341637 0.3112245 0.13950178 1.608457 427
[12] (yogurt) => (whole milk) 0.05602440 0.4026035 0.13950178 1.571735 551
[13] (rolls/buns) => (whole milk) 0.05663447 0.3079049 0.18393493 1.203032 557
[14] (other vegetables) => (whole milk) 0.07483477 0.3867178 0.19349263 1.313634 718
[15] (whole milk) => (other vegetables) 0.07483477 0.2928770 0.25551601 1.313634 718
```

RStudio Environment:

name	class	attributes
apriori	arules	15295 obs. of 4 variables
apriori_rules	arules	Formal class rules
data	data.frame	9835 obs. of 8 variables
data	data.frame	9835 obs. of 8 variables
data	data.frame	Formal class transactions
X	data.frame	9835 obs. of 8 variables
values		
datapath	character	"E:/project 05/grc.csv"
min_confidence	numeric	0.25
min_support	numeric	0.03

RStudio User Library:

name	Description	Version
arules	Mining Association Rules and Frequent Itemsets	1.7-2
arulesViz	Visualizing Association Rules and Frequent Itemsets	1.5-1
arulesL	Safe Password Entry for R, Git, and SSH	1.1
base64enc	Tools for base64 encoding	0.1-3
cellranger	Translate Spreadsheet Cell Ranges to Rows and Columns	1.1.0
class	Functions for Classification	7.3-19
cli	Helpers for Developing Command Line Interfaces	3.1.0
cluster	"Finding Groups in Data": Cluster Analysis	2.1.2
colorspace	A Toolbox for Manipulating and Assessing Colors and Palettes	2.0-2
cpp11	A C++11 Interface for R's C Interface	0.4.2
crayon	Colored Terminal Output	1.4.2
crossstalk	Inter-Widget Interactivity for HTML Widgets	1.2.0
curl	A Modern and Flexible Web Client for R	4.3.2
data.table	Extension of data.frame	1.14.2
digest	Create Compact Hash Digests of R Objects	0.6.28
dplyr	A Grammar of Data Manipulation	1.0.7
DT	A Wrapper of the JavaScript Library	0.20
ellips	Tools for Working with ...	0.3.2