

111208094_hw1

1. 請讀入資料：stock = read_csv("stock.csv") 資料為2023/12/1~2024/3/7，5個股票的股價，變數為：
id_name：每檔股票的證券代碼與名稱，如：2330台積電，表示證券代碼2330，公司名稱為台積電。
type：open 代表開盤價，close 代表收盤價。2023/12/1：為該天交易價格（剩下日期變數依此類推）。
請用tidyr提到的gather,spread,sperate等函數指令，將資料整理成下方「tidy」格式

```
#By myself (我寫好爛好長)
install.packages("readr")
```

```
## 將程式套件安裝入 'C:/Users/Ava/AppData/Local/R/win-library/4.4'
## (因為 'lib' 沒有被指定)
```

```
## 程式套件 'readr' 開啟成功，MD5 和檢查也透過
##
## 下載的二進位程式套件在
## C:\Users\Ava\AppData\Local\Temp\RtmpsdWoq6\downloaded_packages 裡
```

```
library(readr)
library(tidyr)
stock_short <- read_csv("C:\\Users\\Ava\\Desktop\\R\\HW1\\stock.csv")
```

```
## Rows: 10 Columns: 63
```

```
## — Column specification —————
## Delimiter: ","
## chr (2): id_name, type
## dbl (61): 2024/3/7, 2024/3/6, 2024/3/5, 2024/3/4, 2024/3/1, 2024/2/29, 2024...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
stock_short1 <- separate(stock_short, id_name, into = c("id", "name"), sep = " ")
stock_short2 <- gather(stock_short1, key = "time", value = "value", -id, -name, -type)
stock_short3 <- separate(stock_short2, time, into = c("year", "month", "date"), sep = "/")
stock_long <- pivot_wider(stock_short3, names_from = type, values_from = value)
print(stock_long) #answer
```

```
## # A tibble: 305 x 7
##   id    name  year month date    open  close
##   <chr> <chr>  <chr> <chr> <chr>  <dbl> <dbl>
## 1 2330 台積電 2024 3      7      755   760
## 2 2002 中鋼    2024 3      7       24   24.0
## 3 2885 元大金 2024 3      7      27.3  28.1
## 4 9904 寶成    2024 3      7      30.3  30.4
## 5 2454 聯發科 2024 3      7     1220  1230
## 6 2330 台積電 2024 3      6      718   735
## 7 2002 中鋼    2024 3      6      24.0  24.0
## 8 2885 元大金 2024 3      6      27.4  27.5
## 9 9904 寶成    2024 3      6      30.5  30.3
## 10 2454 聯發科 2024 3      6     1145  1190
## # i 295 more rows
```

2. 電商公司，有三個資料集合：

sales.df：產品銷售狀況（“salesID”銷售紀錄編號，“Store”店家編號，“Product”產品編號，“Client”顧客編號，“UnitPrice”單價，“Quantity”購買數量，“Region”顧客國家）

client.df：顧客的個人資料（“Client”顧客編號，“Age”年紀，“Membership”會員等級，“Gender”性別）

prod.df：產品的相關資料（“Item”代號_產品）請用tidyverse套件裡學到的方法，分析

(1.) prod.df 裡將兩個變數，誤紀錄為在同一個column，其將其分為兩個變數Product（數字部分）及Item（商品部分），取代原prod.df。(2.) 將3個報表合併為full.table(3.) 在full.table. 新增一個變數「總消費」為spend = UnitPrice*Quantity(4.) 在full.table將會員等級分組，其中gold和diamond的顧客為一組，其他等級的為一組，針對兩組客戶進行比較介紹（例如平均年紀、性別、國家、消費情況差異等）。(5.) 在full.table針對女性客戶進行分析（例如平均年紀、國家、消費情況等），並對他們在不同產品的「總消費」畫圖分析。

#資料匯入

```
client_list <- read_csv("C:\\Users\\Ava\\Desktop\\R\\HW1\\client_list.csv")
```

```
## New names:
## Rows: 20 Columns: 5
## — Column specification
## _____ Delimiter: "," chr
## (2): Membership, Gender dbl (3): ...1, Client, Age
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this
## message.
## • `` -> `...1`
```

```
salesdata <- read_csv("C:\\Users\\Ava\\Desktop\\R\\HW1\\salesdata.csv")
```

```
## New names:
## Rows: 100 Columns: 7
## — Column specification
## _____ Delimiter: "," chr
## (2): Store, Region dbl (5): ...1, Product, Client, UnitPrice, Quantity
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this
## message.
## • `` -> `...1`
```

```
product_list <- read_csv("C:\\Users\\Ava\\Desktop\\R\\HW1\\product_list.csv")
```

```
## Rows: 6 Columns: 1
## — Column specification —————
## Delimiter: ","
## chr (1): Item
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
##(1.)
product_list1 <- separate(product_list,Item,into = c("Product","Item"),sep = "_",,convert=TRUE)
product_list1
```

```
## # A tibble: 6 × 2
##   Product Item
##   <int> <chr>
## 1    101 iPhone
## 2    102 iPad
## 3    103 MacBook
## 4    104 iMac
## 5    105 AirPods
## 6    106 AppleWatch
```

```
##
## 載入套件: 'dplyr'
```

```
## 下列物件被遮斷自 'package:stats':
##
##   filter, lag
```

```
## 下列物件被遮斷自 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
## # A tibble: 100 × 12
##   ...1.x Store Product Client UnitPrice Quantity Region   ...1.y   Age
##   <dbl> <chr>   <dbl> <dbl>   <dbl>   <dbl> <chr>   <dbl> <dbl>
## 1      1  1 A      103      1      10      72 Brazil      1    36
## 2      2 25 A      103      1      18      26 Korea      1    36
## 3      3 28 C      105      1      11       4 Japan      1    36
## 4      4 10 A      103      2      14      58 China      2    46
## 5      5 11 C      101      2       8      35 Brazil      2    46
## 6      6 41 B      105      2       4      76 Korea      2    46
## 7      7 59 B      101      2       7      58 Korea      2    46
## 8      8 67 B      105      2      14      98 Thailand    2    46
## 9      9 84 C      105      2      14      59 Germany     2    46
## 10     97 B      105      2       9      20 Brazil      2    46
## # i 90 more rows
## # i 3 more variables: Membership <chr>, Gender <chr>, Item <chr>
```

```
#(3.)
full_table%>%
  mutate( Spend = UnitPrice*Quantity ) -> full_table1
library(writexl)
write_xlsx(full_table1, path = "C:/Users/Ava/Downloads/full_table1.xlsx") #保存下來方便之後跑EDA!!
full_table1
```

```
## # A tibble: 100 × 13
##   ...1.x Store Product Client UnitPrice Quantity Region   ...1.y   Age
##   <dbl> <chr>   <dbl> <dbl>   <dbl>   <dbl> <chr>   <dbl> <dbl>
## 1      1  1 A      103      1      10      72 Brazil      1    36
## 2      2 25 A      103      1      18      26 Korea      1    36
## 3      3 28 C      105      1      11       4 Japan      1    36
## 4      4 10 A      103      2      14      58 China      2    46
## 5      5 11 C      101      2       8      35 Brazil      2    46
## 6      6 41 B      105      2       4      76 Korea      2    46
## 7      7 59 B      101      2       7      58 Korea      2    46
## 8      8 67 B      105      2      14      98 Thailand    2    46
## 9      9 84 C      105      2      14      59 Germany     2    46
## 10     97 B      105      2       9      20 Brazil      2    46
## # i 90 more rows
## # i 4 more variables: Membership <chr>, Gender <chr>, Item <chr>, Spend <dbl>
```

```
#(4.) EDA Start!!!
#分組
full_table2 <- full_table1 %>%
  mutate(Group = ifelse(Membership %in% c("gold", "diamond"), "Gold & Diamond", "Other"))
full_table2
```

```
## # A tibble: 100 × 14
##   ...1.x Store Product Client UnitPrice Quantity Region   ...1.y   Age
##   <dbl> <chr>   <dbl> <dbl>   <dbl>   <dbl> <chr>   <dbl> <dbl>
## 1      1      1 A      103      1      10      72 Brazil      1    36
## 2      2     25 A      103      1      18      26 Korea       1    36
## 3      3     28 C      105      1      11       4 Japan       1    36
## 4      4     10 A      103      2      14      58 China       2    46
## 5      5     11 C      101      2       8      35 Brazil      2    46
## 6      6     41 B      105      2       4      76 Korea       2    46
## 7      7     59 B      101      2       7      58 Korea       2    46
## 8      8     67 B      105      2      14      98 Thailand    2    46
## 9      9     84 C      105      2      14      59 Germany    2    46
## 10     10     97 B      105      2       9      20 Brazil      2    46
## # i 90 more rows
## # i 5 more variables: Membership <chr>, Gender <chr>, Item <chr>, Spend <dbl>,
## #   Group <chr>
```

```
#年紀
age_comparison <- full_table2 %>%
  group_by(Group) %>%
  summarise(Average_Age = mean(Age, na.rm = TRUE))
age_comparison
```

```
## # A tibble: 2 × 2
##   Group      Average_Age
##   <chr>         <dbl>
## 1 Gold & Diamond    41.8
## 2 Other            40.9
```

```
#性別
gender_comparison <- full_table2 %>%
  group_by(Group, Gender) %>%
  summarise(Count = n()) %>%
  mutate(Percentage = Count / sum(Count) * 100)
```

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

```
gender_comparison
```

```
## # A tibble: 4 × 4
## # Groups:   Group [2]
##   Group      Gender Count Percentage
##   <chr>         <chr> <int>     <dbl>
## 1 Gold & Diamond female    43     75.4
## 2 Gold & Diamond male     14     24.6
## 3 Other         female    21     48.8
## 4 Other         male     22     51.2
```

#國家

```
country_comparison <- full_table2 %>%
  group_by(Group, Region) %>%
  summarise(Count = n()) %>%
  arrange(Group, desc(Count))
```

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

country_comparison

```
## # A tibble: 20 × 3
## # Groups:   Group [2]
##   Group      Region    Count
##   <chr>      <chr>    <int>
## 1 Gold & Diamond China      8
## 2 Gold & Diamond Korea      8
## 3 Gold & Diamond Germany    6
## 4 Gold & Diamond Taiwan     6
## 5 Gold & Diamond Thailand    6
## 6 Gold & Diamond Brazil      5
## 7 Gold & Diamond France      5
## 8 Gold & Diamond Spain       5
## 9 Gold & Diamond USA         5
## 10 Gold & Diamond Japan       3
## 11 Other      Korea      11
## 12 Other      Taiwan      6
## 13 Other      Spain       5
## 14 Other      USA         5
## 15 Other      Brazil      4
## 16 Other      France      4
## 17 Other      China       3
## 18 Other      Germany     2
## 19 Other      Thailand    2
## 20 Other      Japan       1
```

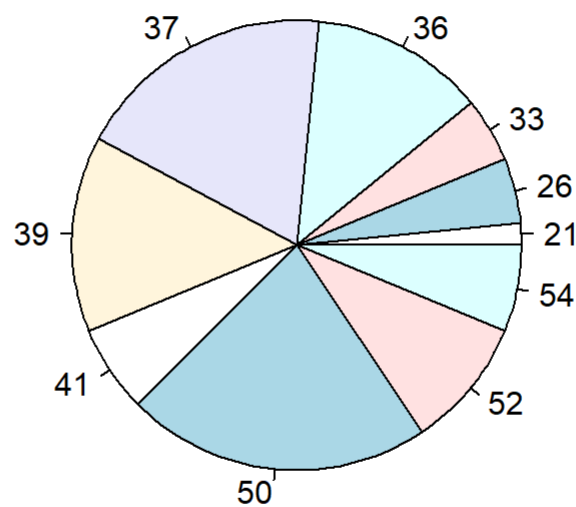
#消費差異

```
spend_comparison <- full_table2 %>%
  group_by(Group) %>%
  summarise(Average_Spend = mean(Spend, na.rm = TRUE),
            Total_Spend = sum(Spend, na.rm = TRUE),
            Std_Dev_Spend = sd(Spend, na.rm = TRUE))
```

```
#(5.)
full_table2_female <- full_table2[full_table2$Gender == "female",]
write_xlsx(full_table2_female, path = "C:/Users/Ava/Downloads/full_table2_female.xlsx")

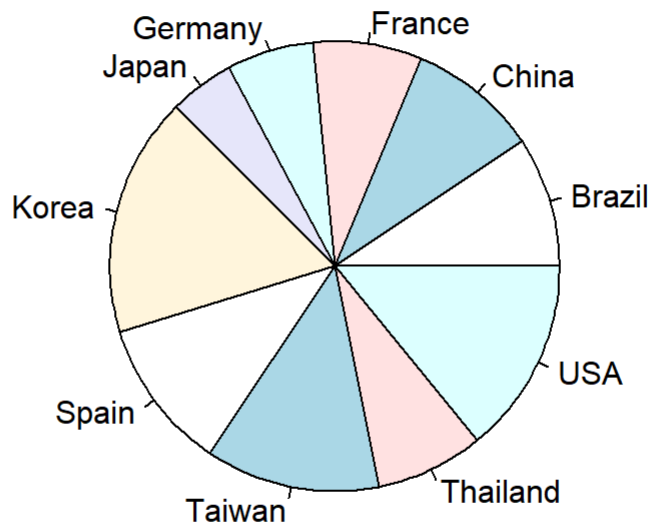
#年紀
age_comparison_female <- full_table2_female %>%
  summarise(Average_Age = mean(Age, na.rm = TRUE))

pie(table(full_table2_female$Age))
```



```
#國家
country_comparison_female <- full_table2_female %>%
  group_by(Region) %>%
  summarise(Count = n()) %>%
  arrange(desc(Count))

pie(table(full_table2_female$Region))
```

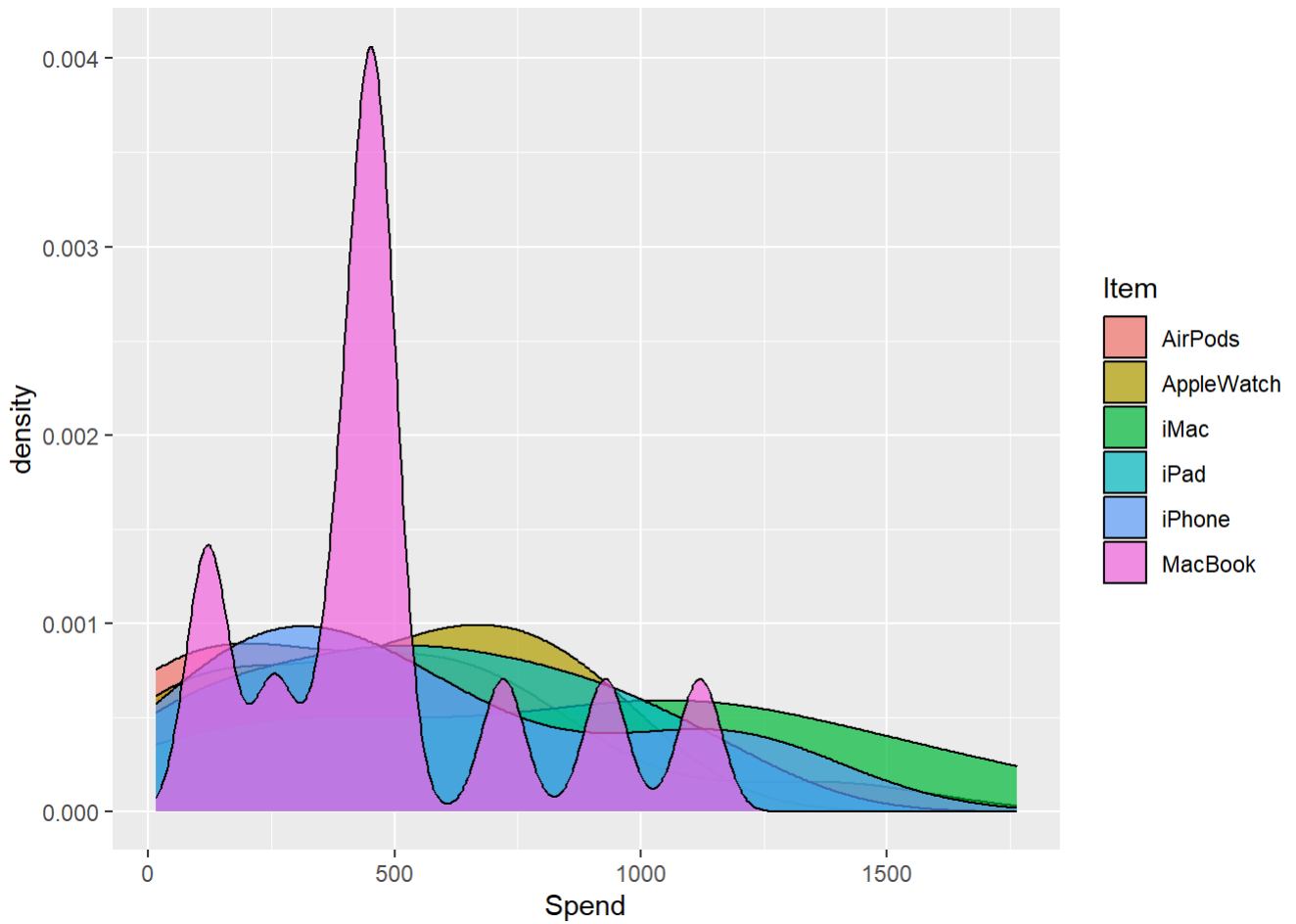


#消費差異

```
spend_comparison_female <- full_table2_female %>%  
  summarise(Average_Spend = mean(Spend, na.rm = TRUE),  
            Total_Spend = sum(Spend, na.rm = TRUE),  
            Std_Dev_Spend = sd(Spend, na.rm = TRUE))
```

#不同產品的「總消費」畫圖分析

```
library(ggplot2)  
p <- ggplot(full_table2_female, aes(x = Spend , fill = Item)) + geom_density(alpha = 0.7)  
plot(p, labels = "AUTO")
```

plot

```
## function (x, y, ...)\n## UseMethod("plot")\n## <bytecode: 0x000001feb30f0df0>\n## <environment: namespace:base>
```