Ta Feng 資料集銷貨數據分析 / Ta Feng Grocery Dataset Analysis

利用R針對Ta Feng Grocery Dataset進行資料處理、探索性資料分析、資料視覺化、RFM模型、顧客集群分析、客戶終身價值計算等等，並且以分析結果進一步規劃行銷計畫與商業策略。

- 本專案獲得國立中山大學112-1學期商業分析實務個案競賽冠軍

Using R to conduct data processing, exploratory data analysis, data visualization, RFM model, customer segmentation, calculation of customer lifetime value, etc., on the Ta Feng Grocery Dataset. The analysis results are further utilized for developing business insights and marketing strategies.

- Achieved 1st place in the NSYSU 112-1 semester(2023 Fall) Practical Business Analytics Competition.

Video Link (Part1): https://www.youtube.com/watch?v=UisNREVOqfE (https://www.youtube.com/watch?v=UisNREVOqfE)
Video Link (Part2): https://www.youtube.com/watch?v=yVOBvzZqVEY (https://www.youtube.com/watch?v=yVOBvzZqVEY)
Ta Feng Grocery Dataset: https://www.kaggle.com/datasets/chiranjivdas09/ta-feng-grocery-dataset (https://www.kaggle.com/datasets/chiranjivdas09/ta-feng-grocery-dataset)

目錄 Table of Contents

TaFeng_Data_Analysis

G11, NSYSU

2024-01-30 16:29:54.673366

資料彙整流程

Fig-1:交易資料彙整

1. 載入套件與資料 Importing Libraries and Loading Data

```
#載入套件
rm(list=ls(all=T))
pacman::p_load(magrittr, readr, caTools, ggplot2, dplyr, vcd, plotly,tidyr, gridExtra, reshape2, heatmaply,morpheu
         s)
```

```
#讀進資料
df = read_csv("data/ta_feng_all_months_merged.csv") %>%
     data.frame %>%
     setNames(c("date","cust","age","area","cat","prod","qty","cost","price"))
```

```
## Rows: 817741 Columns: 9
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (5): TRANSACTION_DT, CUSTOMER_ID, AGE_GROUP, PIN_CODE, PRODUCT_ID
## dbl (4): PRODUCT_SUBCLASS, AMOUNT, ASSET, SALES_PRICE
##
## 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.
```

```
head(df)
```

```
##        date     cust   age   area    cat          prod qty cost price
## 1 11/1/2000 01104905 45-49    115 110411 4710199010372   2   24    30
## 2 11/1/2000 00418683 45-49    115 120107 4710857472535   1   48    46
## 3 11/1/2000 01057331 35-39    115 100407 4710043654103   2  142   166
## 4 11/1/2000 01849332 45-49 Others 120108 4710126092129   1   32    38
## 5 11/1/2000 01981995 50-54    115 100205 4710176021445   1   14    18
## 6 11/1/2000 01741797 35-39    115 110122 0078895770025   1   54    75
```

2.資料預處理 Data Preprocessing

```
#資料結構
str(df)
```

```
## 'data.frame':    817741 obs. of  9 variables:
##  $ date : chr  "11/1/2000" "11/1/2000" "11/1/2000" "11/1/2000" ...
##  $ cust : chr  "01104905" "00418683" "01057331" "01849332" ...
##  $ age  : chr  "45-49" "45-49" "35-39" "45-49" ...
##  $ area : chr  "115" "115" "115" "Others" ...
##  $ cat  : num  110411 120107 100407 120108 100205 ...
##  $ prod : chr  "4710199010372" "4710857472535" "4710043654103" "4710126092129" ...
##  $ qty  : num  2 1 2 1 1 1 1 1 2 1 ...
##  $ cost : num  24 48 142 32 14 54 85 45 70 43 ...
##  $ price: num  30 46 166 38 18 75 105 68 78 53 ...
```

```
#資料唯一值數量 / Checking Unique Values
col_list = colnames(df)
unique_counts = list()

for (col in col_list) {
  unique_count = length(unique(df[[col]]))
  unique_counts[[col]] = unique_count
}

for (col in col_list) {
  cat("Column", col, "has", unique_counts[[col]], "unique values.\n")
}
```

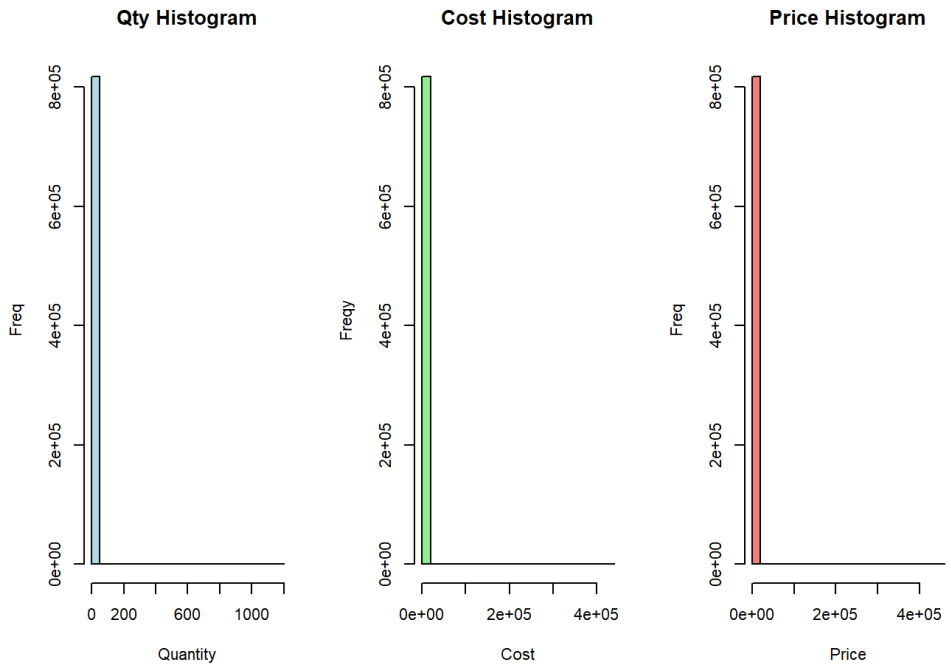
```
## Column date has 120 unique values.
## Column cust has 32266 unique values.
## Column age has 11 unique values.
## Column area has 8 unique values.
## Column cat has 2012 unique values.
## Column prod has 23812 unique values.
## Column qty has 90 unique values.
## Column cost has 1728 unique values.
## Column price has 2191 unique values.
```

```
#日期格式轉換、年齡層級與郵遞區號整理
df$date = as.Date(df$date, format="%m/%d/%Y")
age.group = c("<25","25-29","30-34","35-39","40-44",
              "45-49","50-54","55-59","60-64",">65")
df$age = c(paste0("u",seq(24,69,5)),"none")[match(df$age,age.group,11)]
df$area = paste0("z",df$area)
head(df)
```

```
##         date     cust age    area    cat          prod qty cost price
## 1 2000-11-01 01104905 u49    z115 110411 4710199010372   2   24    30
## 2 2000-11-01 00418683 u49    z115 120107 4710857472535   1   48    46
## 3 2000-11-01 01057331 u39    z115 100407 4710043654103   2  142   166
## 4 2000-11-01 01849332 u49 zOthers 120108 4710126092129   1   32    38
## 5 2000-11-01 01981995 u54    z115 100205 4710176021445   1   14    18
## 6 2000-11-01 01741797 u39    z115 110122 0078895770025   1   54    75
```

Area Code: 105=松山區、106=大安區、110=信義區、114=內湖區、115=南港區、221=汐止市、Others=其他、Unknown=未知

```
#檢視數量、成本、售價分布狀況
par(mfrow=c(1,3),cex=0.7)
hist(df$qty, main = "Qty Histogram", xlab = "Quantity", ylab = "Freq", col = "lightblue")
hist(df$cost, main = "Cost Histogram", xlab = "Cost", ylab = "Freqy", col = "lightgreen")
hist(df$price, main = "Price Histogram", xlab = "Price", ylab = "Freq", col = "lightcoral")
```




```
#可見三者資料都是極端右偏分布，且無負值
```

```
#查看數據極大值
apply(df[,7:9], 2, max)
```

```
##    qty   cost  price
##   1200 432000 444000
```

```
#定義離群值
sapply(df[,7:9], quantile, prob=c(.99, .999, .9995))
```

```
##        qty   cost   price
## 99%      6  858.0 1014.00
## 99.9%   14 2722.0 3135.82
## 99.95%  24 3799.3 3999.00
```

```
#移除離群值 Remove Outliers
df = subset(df, qty<=24 & cost<=3800 & price<=4000)
nrow(df)
```

```
## [1] 817182
```

```
#根據日期和客人彙總訂單 / Agrregate by Date & Customer
df$tid = group_indices(df, date, cust)
```

```
## Warning: The `...` argument of `group_indices()` is deprecated as of dplyr 1.0.0.
## ℹ Please `group_by()` first
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
#檢視彙總後的No. cust, cat, prod, tid唯一值 / Unique Values
sapply(df[c("cust","cat","prod","tid")], n_distinct)
```

```
##   cust    cat   prod    tid
##  32256   2007  23789 119422
```

```
#根據前述分好的tid，彙整出需要的資料
X = df %>% group_by(tid) %>%
  summarise(
    date = min(date),          # 交易日期
    cust = min(cust),          # 顧客 ID
    age = min(age),            # 顧客 年齡級別
    area = min(area),          # 顧客 居住區別
    items = n(),               # 交易項目(總)數
    pieces = sum(qty),         # 產品(總)件數
    total = sum(price),        # 交易(總)金額
    gross = sum(price - cost)  # 毛利
) %>% data.frame
nrow(X)
```

```
## [1] 119422
```

```
#彙整後資料概覽
summary(X)
```

```
##       tid              date                cust               age
##  Min.   :     1   Min.   :2000-11-01   Length:119422      Length:119422
##  1st Qu.: 29856   1st Qu.:2000-11-29   Class :character   Class :character
##  Median : 59712   Median :2001-01-01   Mode  :character   Mode  :character
##  Mean   : 59712   Mean   :2000-12-31
##  3rd Qu.: 89567   3rd Qu.:2001-02-02
##  Max.   :119422   Max.   :2001-02-28
##      area               items             pieces            total
##  Length:119422      Min.   :  1.000   Min.   :  1.000   Min.   :    5
##  Class :character   1st Qu.:  2.000   1st Qu.:  3.000   1st Qu.:  227
##  Mode  :character   Median :  5.000   Median :  6.000   Median :  510
##                     Mean   :  6.843   Mean   :  9.294   Mean   :  859
##                     3rd Qu.:  9.000   3rd Qu.: 12.000   3rd Qu.: 1082
##                     Max.   :112.000   Max.   :339.000   Max.   :30171
##      gross
##  Min.   :-1645.0
##  1st Qu.:   21.0
##  Median :   68.0
##  Mean   :  132.3
##  3rd Qu.:  169.0
##  Max.   : 8069.0
```

```
#毛利有負，代表並非都是賺錢
```

```
# 定義離群值
sapply(X[,6:9], quantile, prob=c(.999, .9995, .9999))
```

```
##        items   pieces     total    gross
## 99.9%     54  81.0000  9009.579 1824.737
## 99.95%    62  94.2895 10611.579 2179.817
## 99.99%    82 133.0000 16044.401 3226.548
```

```
# 移除離群值 / Remove Outliers
X = subset(X, items<=62 & pieces<95 & total<16000)
nrow(X)
```

```
## [1] 119328
```

```
#將X按照Cust進行分組，然後對每個客戶的數據進行摘要統計。=
#r（最近購買的天數）
#s（購買歷史最早的天數）
#f（購買次數）
#m（平均每次購買金額）
#rev（總收入貢獻）
#raw（總毛利貢獻）
#age（年齡組別）
#area（地區代號）
d0 = max(X$date) + 1
A = X %>% mutate(
  days = as.integer(difftime(d0, date, units="days"))) %>%
    group_by(cust) %>%
      summarize(
        r = min(days),      # recency
        s = max(days),      # seniority
        f = n(),            # frquency
        m = round(mean(total)),    # monetary
        rev = sum(total),   # total revenue contribution
        raw = sum(gross),   # total gross profit contribution
        age = min(age),     # age group
        area = min(area),   # area code
  ) %>%
    data.frame
nrow(A) # 共有32241個客戶的紀錄
```

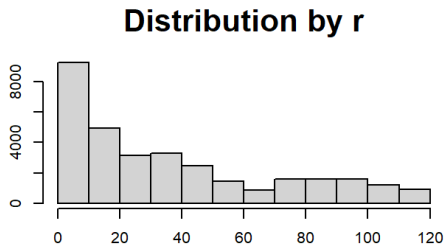
```
## [1] 32241
```

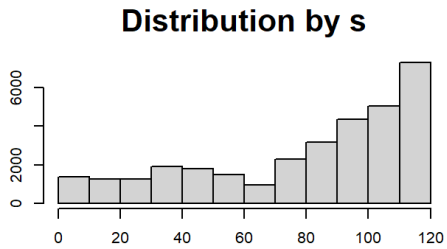
到目前，df為引入的原始資料，X則為根據日期、客人整理過的資料，A則為從X中再計算得出的數據指標資料

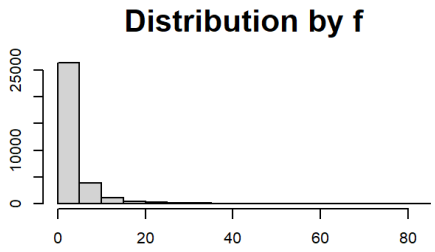
3.探索性資料分析與視覺化 Exploratory Data Analysis

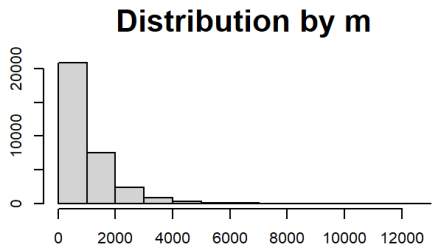
資料分布 Data Distribution

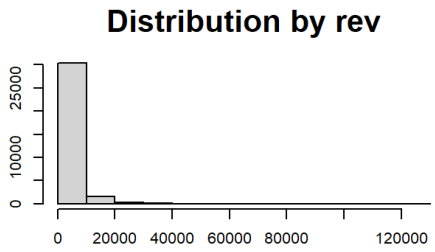
```
#對A中每欄資料做資料分布視覺化
par(mfrow=c(4,2), mar=c(3,3,4,2))
col_list1 = c("r","s","f","m","rev","raw")
col_list2 = c("age","area")
for (col in col_list1) {
  hist(A[,col],freq=T,main=paste("Distribution by", col),xlab=paste(col),ylab="",cex.main=2)
}
for (col in col_list2) {
  table(A[col],useNA="ifany") %>%
    barplot(main=paste("Distribution by", col), las=2, xlab = paste(col))
}
```

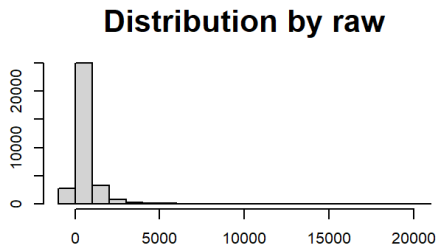


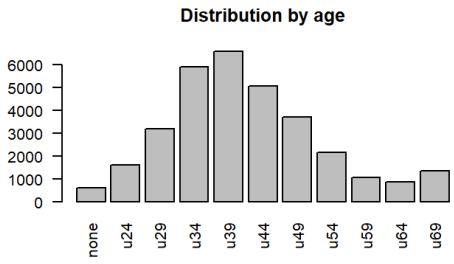


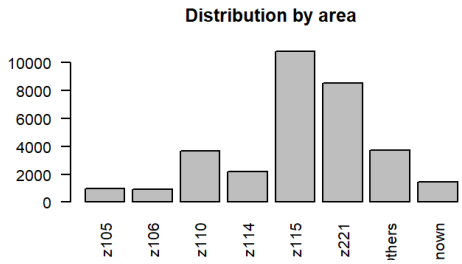












可以看出購買頻率、最近購買的天數、平均每次購買金額、總收入貢獻、總毛利都呈現右偏分布
而購買歷史最早的天數則為左偏分布
年齡的分佈則以30-39為多數，兩側逐漸遞減；地區則是z115和z221明顯較多

年齡和地區分群分析 Age/Area Segmentation

```
#依據年齡和地區，對其他欄做比較
#資料結構處理
linear_data <- A[, c('r', 'f', 'm', 'rev', 'raw')]
A$age_1 = as.character(A$age)
A$area_1 = as.character(A$area)

age = as.numeric(sub("u", "", A$age_1))
```

```
## Warning: NAs introduced by coercion
```

```
area = as.numeric(sub("z","", A$area_1))
```

```
## Warning: NAs introduced by coercion
```

```
combined_data = cbind(linear_data, age,area)
unique(combined_data$age)
```

```
##  [1] NA 39 69 54 49 44 34 29 59 24 64
```

```
unique(combined_data$area)
```

```
## [1] 115 221 114  NA 106 110 105
```

```
combined_data$age = as.factor(combined_data$age)
combined_data$area = as.factor(combined_data$area)

dim(combined_data)
```

```
## [1] 32241     7
```

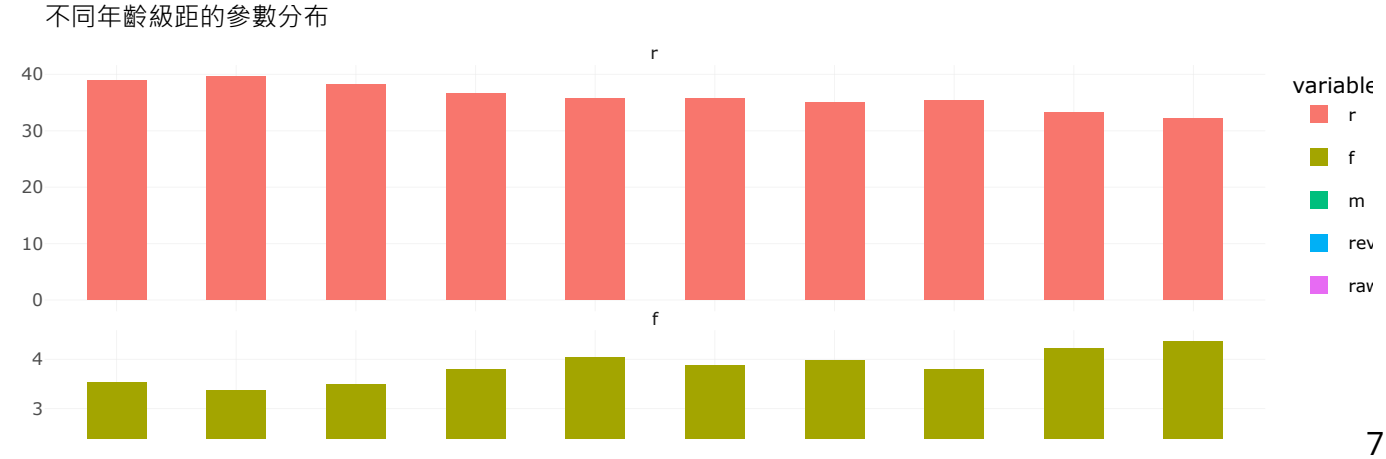
```
#依據年齡整合所需資料
grouped_data_age = aggregate(. ~ age, data = combined_data, FUN = mean)
grouped_data_age = grouped_data_age[,-7]
print(grouped_data_age)
```

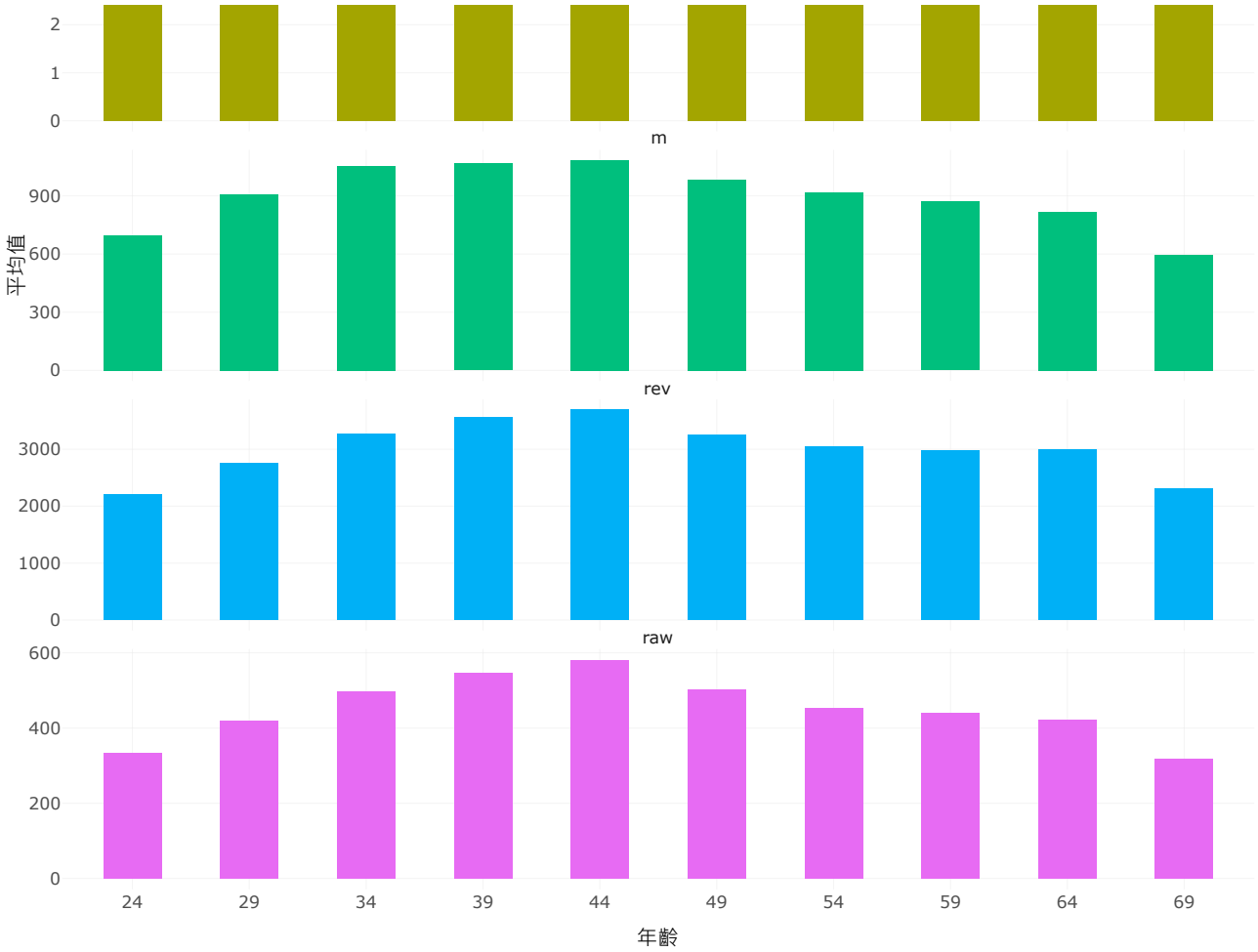
```
##    age        r        f         m      rev      raw
## 1   24 38.83986 3.532384  695.0776 2197.483 333.6683
## 2   29 39.67246 3.371947  909.8805 2747.777 418.9208
## 3   34 38.23150 3.488076 1053.7654 3273.639 497.9307
## 4   39 36.55418 3.797734 1066.7879 3553.817 545.2810
## 5   44 35.78454 4.029160 1083.1125 3690.766 580.8910
## 6   49 35.69766 3.885179  983.3476 3256.897 501.3466
## 7   54 35.08801 3.979413  919.3886 3037.196 453.7422
## 8   59 35.42312 3.800636  870.3499 2980.036 438.9682
## 9   64 33.16941 4.217446  816.0341 2993.977 422.7775
## 10  69 32.11343 4.374479  594.4921 2301.593 316.7456
```

```
#依據地區整合所需資料
grouped_data_area = aggregate(. ~ area, data = combined_data, FUN = mean)
grouped_data_area = grouped_data_area[,-7]
print(grouped_data_area)
```

```
##   area        r        f         m      rev      raw
## 1  105 43.68438 2.158333 1149.8031 2351.713 354.9375
## 2  106 41.02397 2.500000 1333.1264 2916.910 438.3388
## 3  110 44.38900 2.316178 1181.8333 2580.282 397.0348
## 4  114 39.55758 2.354779 1065.3161 2418.269 364.5828
## 5  115 32.93766 4.688758  831.7289 3451.352 543.4887
## 6  221 35.89808 3.960764  987.5049 3498.445 512.7810
```

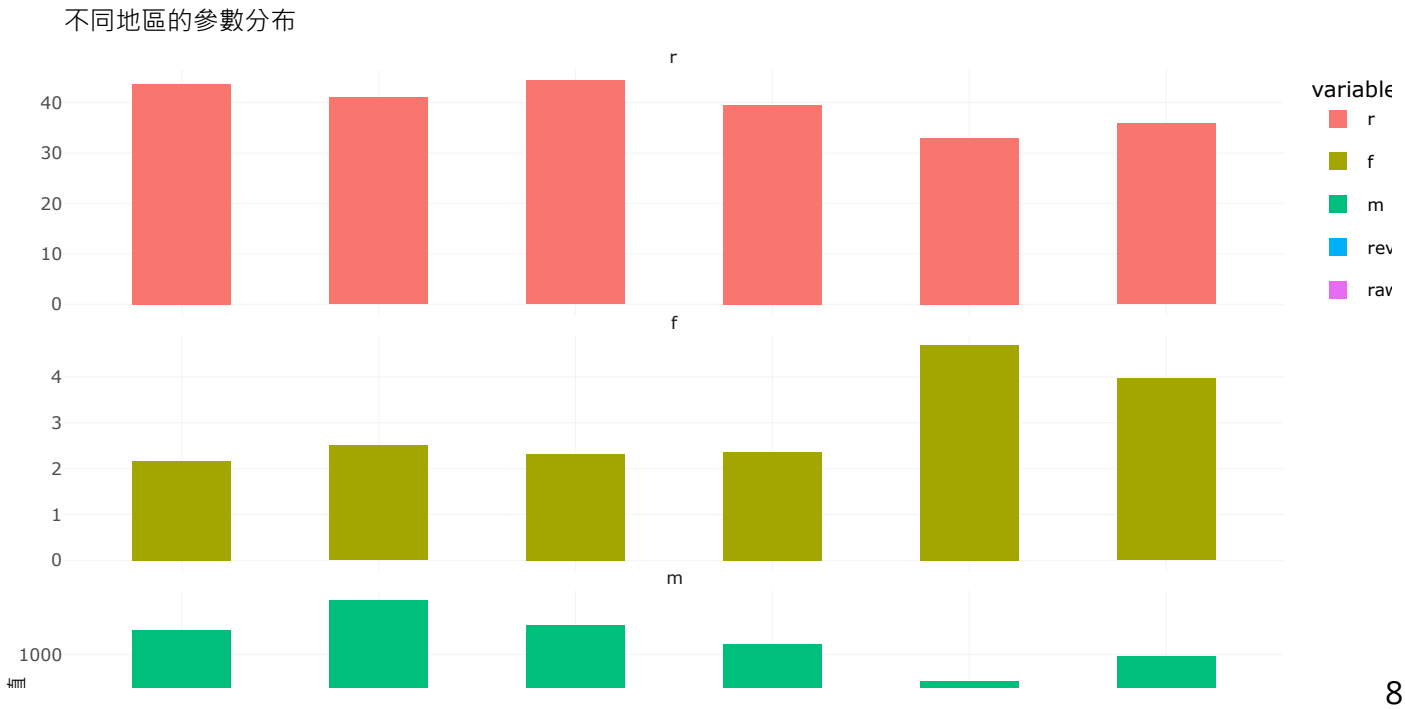
```
#將整理好的資料by age 做視覺化
melted_data_age = melt(grouped_data_age, id.vars = "age")

v = ggplot(melted_data_age, aes(x = age, y = value, fill = variable)) +
  geom_bar(stat = "identity", position = "dodge", width = 0.5) +
  labs(title = "不同年齡級距的參數分布", y = "平均值", x="年齡") +
  theme_minimal() +
  facet_wrap(~variable, scales = "free_y", ncol = 1)
ggplotly(v)
```

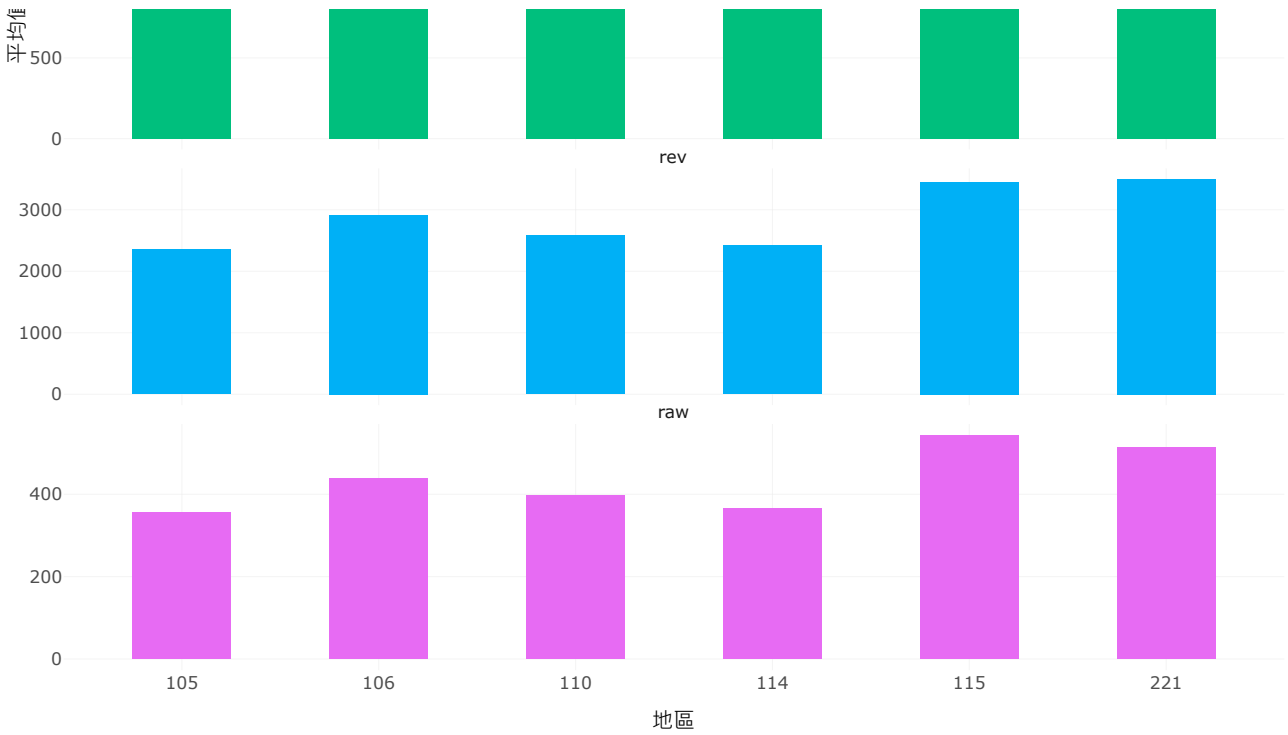




可以看出30-49歲為淨利與獲利貢獻最明顯的區間
而平均消費金額最多同樣為30-49歲，但整體峰度較raw和rev低，值得注意的是儘管m隨著年齡漸高而下降，但年齡未知族群卻很高
總到訪次數一樣以30-49歲較多，但平均到訪次數則以未知族群遠高於其他年齡區間

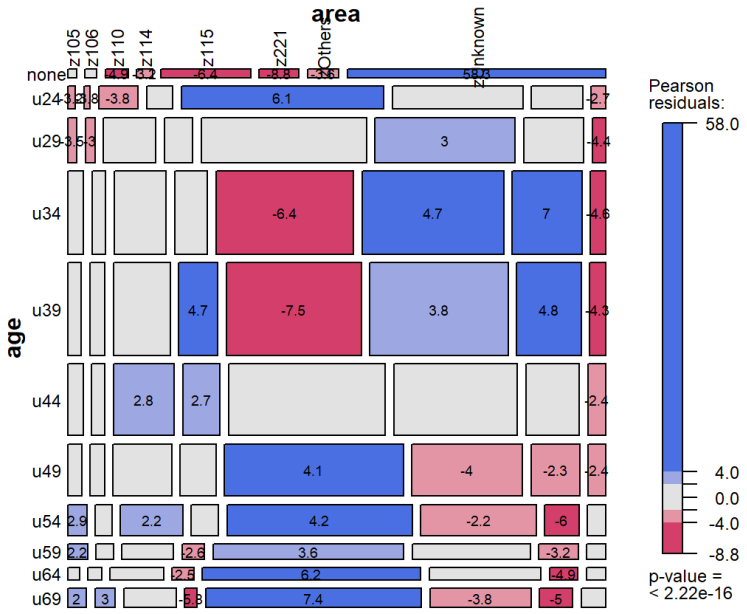
```
#將整理好的資料by area 做視覺化
melted_data_area = melt(grouped_data_area, id.vars = "area")
j = ggplot(melted_data_area, aes(x = area, y = value, fill = variable)) +
  geom_bar(stat = "identity", position = "dodge", width = 0.5) +
  labs(title = "不同地區的參數分布", y = "平均值",x= "地區") +
  theme_minimal() +
  facet_wrap(~variable, scales = "free_y", ncol = 1)
ggplotly(j)
```





可以看出南港和汐止為淨利與獲利貢獻最明顯的地區，遠高於其他區
但平均消費金額而言的話兩者排名居末，顯示此區可能是靠著高客戶基數或者高購買頻率撐起金流
平均到訪次數則以未知地區最多，南港次之，汐止第三，其餘無明顯差別

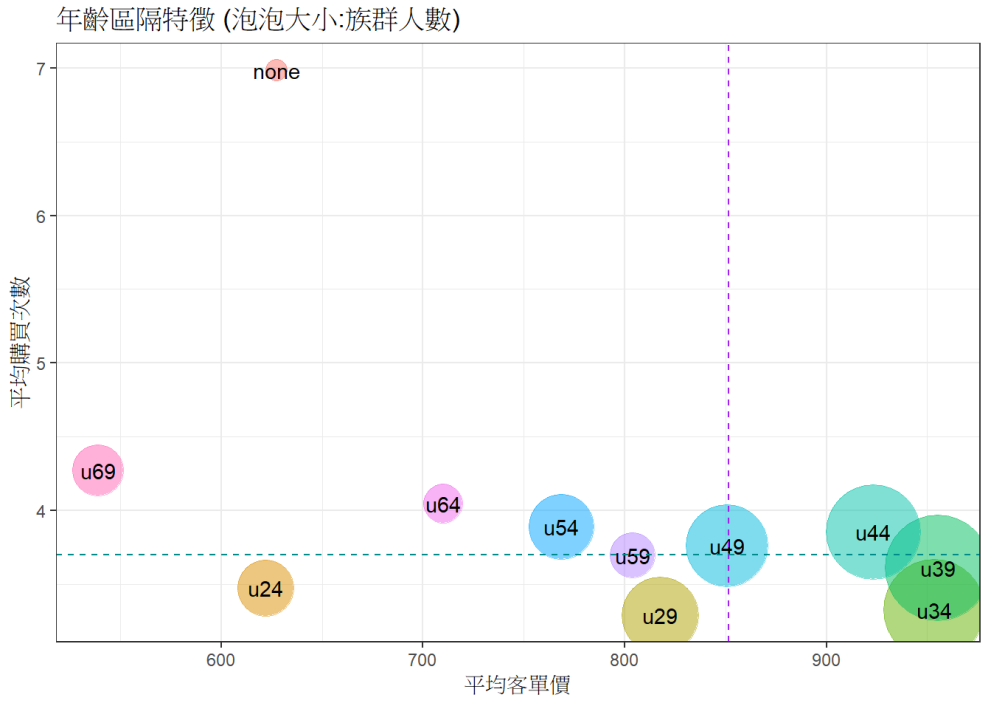
```
#馬賽克圖，以下為分析年齡與地區之間的關係
MOSA = function(formula, data) #定義一個叫MOSA的函數，他接受兩個參數
mosaic(formula, data, shade=T,
  margins=c(0,1,0,0),  #調整圖片邊距
  labeling_args = list(rot_labels=c(90,0,0,0)), #指定label的顯示模式，90=垂直顯示
  gp_labels=gpar(fontsize=9), #label字體大小
  legend_args=list(fontsize=9), #legend字體大小
  gp_text=gpar(fontsize=7), #裡面文本的字體大小(下圖方框中顯示的數字)
  labeling=labeling_residuals) #使用殘差labeling

MOSA(~age+area, A)
```



南港區的消費者多為45歲以上的族群，且 u69 最顯著，代表南港區的高齡族群普遍會前往購買，但30-39歲的族群則否
汐止內湖的消費者多為30-39歲族群，但u49以上(45歲以上)比較不會前往購買

年齡區隔特徵 Age Segmentation

```
#以年齡分群，針對平均購買次數、平均客單價作分析，並且以年齡族群人數作為泡泡大小
A %>%
  group_by(age) %>%
    summarize(
      Group.Size = n(),              # 族群人數
      avg.Freq = mean(f),            # 平均購買次數
      avg.Revenue = sum(f*m)/sum(f)  # 平均客單價
  ) %>%
    ggplot(aes(y=avg.Freq, x=avg.Revenue)) +
    geom_point(aes(col=age, size=Group.Size), alpha=0.5) +
    geom_text(aes(label=age)) +
    scale_size(range=c(5,25)) +
    theme_bw() + theme(legend.position="none") +
    ggtitle("年齡區隔特徵 (泡泡大小:族群人數)") +
    ylab("平均購買次數") + xlab("平均客單價") +
    geom_vline(xintercept = sum(A$f*A$m)/sum(A$f), col="purple", linetype = "dashed")+
    geom_hline(yintercept = mean(A$f), col="darkcyan", linetype="dashed")
```



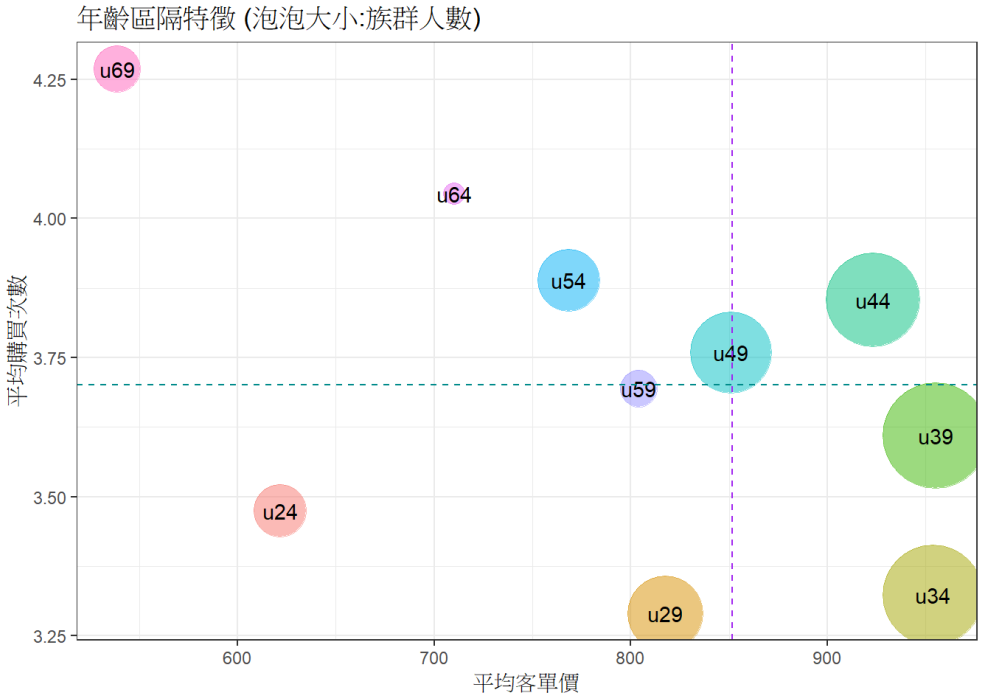
30-44歲的人平均客單價明顯較高
大於65歲的族群平均客單價最低，但平均購買次數較高

```
mean(A$age == "none")
```

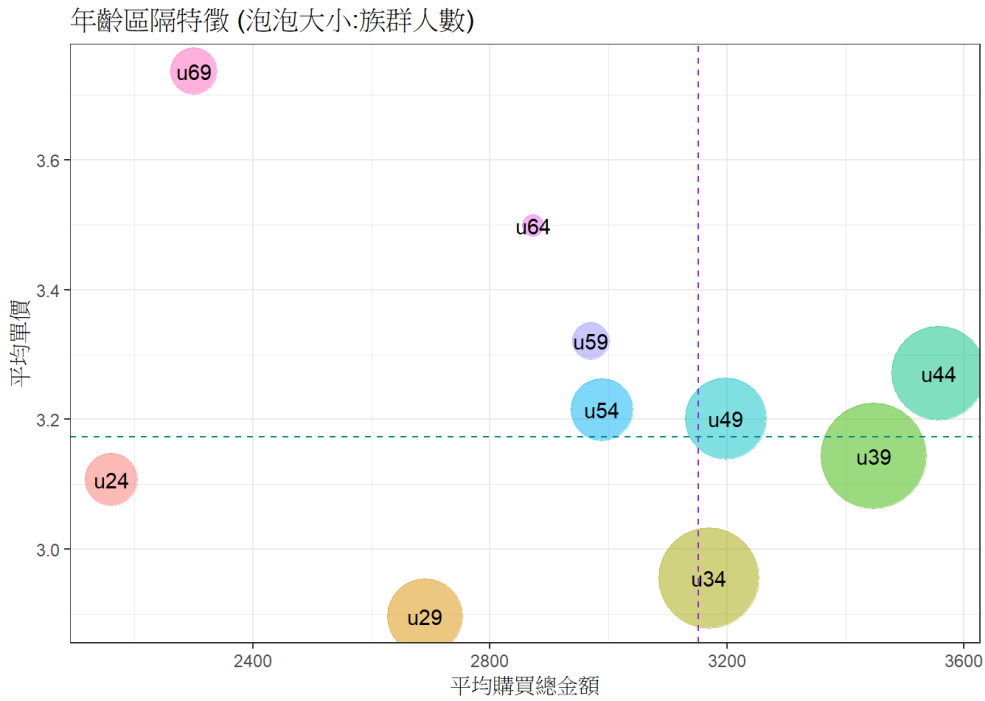
```
## [1] 0.01941627
```

由於 None (沒有年齡資料的顧客)人數不多，而且特徵很獨特，探索時我們可以考慮濾掉這群顧客

```
#濾掉沒有年齡資料的顧客，以年齡分群，針對平均購買次數、平均客單價作分析，並且以年齡族群人數作為泡泡大小
A %>%
  filter(age!="none") %>%    # 濾掉沒有年齡資料的顧客('a99')
  group_by(age) %>%
    summarize(
      Group.Size = n(),              # 族群人數
      avg.Freq = mean(f),            # 平均購買次數
      avg.Revenue = sum(f*m)/sum(f)  # 平均客單價
  ) %>%
      ggplot(aes(y=avg.Freq, x=avg.Revenue)) +
      geom_point(aes(col=age, size=Group.Size), alpha=0.5)+#geom_point=添加散點圖，根據age設置顏色、根據Group.Size設定
          大小
      geom_text(aes(label=age)) +
      scale_size(range=c(5,25)) +#設置點的大小Range from 5 to 25
      theme_bw() + theme(legend.position="none") +#設置主題為全白，並且不顯示圖例
      ggtitle("年齡區隔特徵 (泡泡大小:族群人數)") +
      ylab("平均購買次數") + xlab("平均客單價")  +
      geom_vline(xintercept = sum(A$f*A$m)/sum(A$f), col="purple", linetype = "dashed")+
      geom_hline(yintercept = mean(A$f), col="darkcyan", linetype="dashed")
```

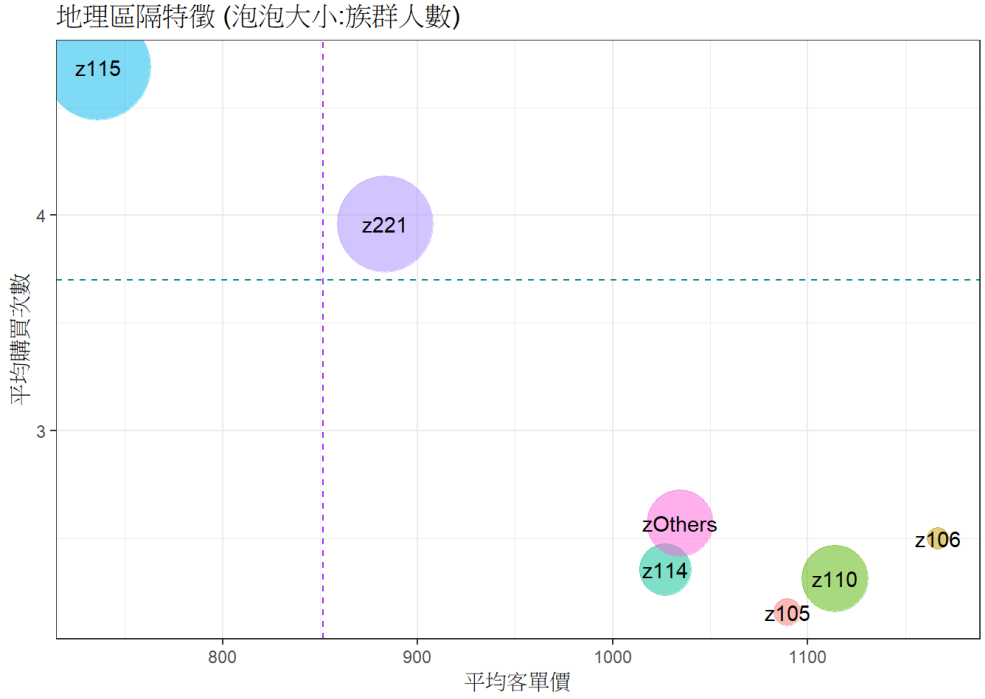



```
#濾掉沒有年齡資料的顧客，以年齡分群，針對平均購買單價、平均購買總金額作分析，並且以年齡族群人數作為泡泡大小
A%>%
  filter(age!="none")  %>%  #濾掉沒有年齡資料和地區未知的顧客
  group_by(age) %>%
  summarize(
  Group.Size = n(),              # 族群大小
  avg.total = mean(rev),          # 平均購買總金額
  avg.price = sum(f*m)/sum(m)  # 平均客單價
  ) %>%
  ggplot(aes(y=avg.price, x=avg.total)) +
  geom_point(aes(col=age, size=Group.Size), alpha=0.5) +
  geom_text(aes(label=age)) +
  scale_size(range=c(5,25)) +
  theme_bw() + theme(legend.position="none") +
  ggtitle("年齡區隔特徵 (泡泡大小:族群人數)") +
  geom_vline(xintercept = mean(A$rev),col="purple",linetype = "dashed")+
  geom_hline(yintercept = sum(A$f*A$m)/sum(A$m), col="darkcyan", linetype="dashed")+
  ylab("平均單價") + xlab("平均購買總金額")
```



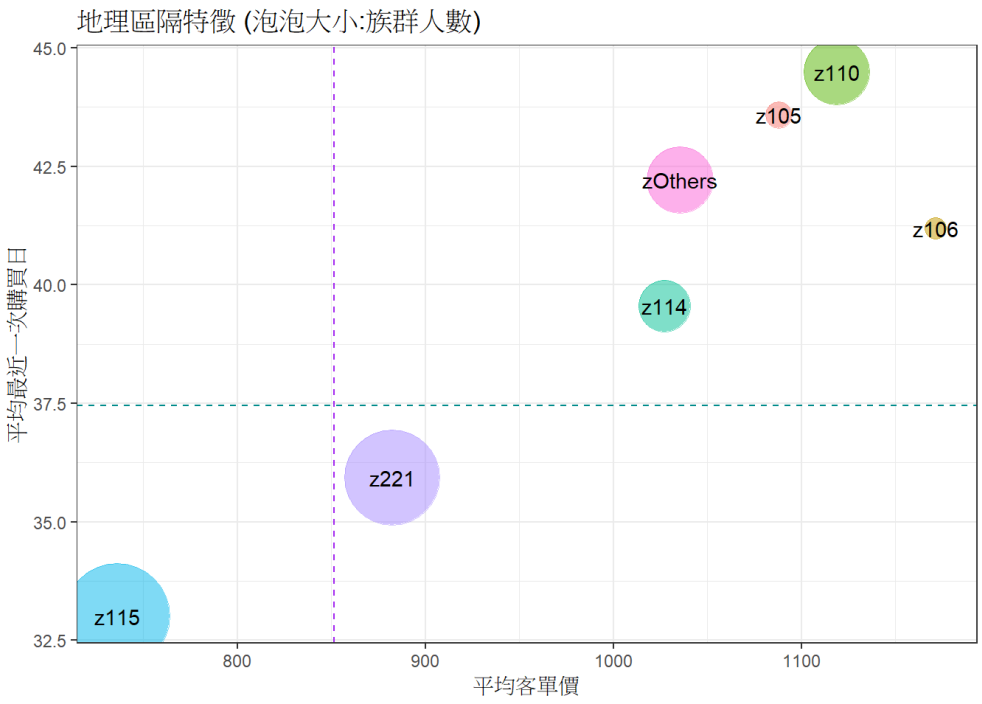
地理區隔特徵 Area Segmentation

```
#濾掉沒有年齡資料和未知地區的顧客，以地區分群，針對平均購買次數、平均客單價作分析，並且以地區族群人數作為泡泡大小
A %>%
  filter(age!="none"& area!="zUnknown") %>%    # 濾掉沒有年齡資料的顧客
  group_by(area) %>%
    summarize(
      Group.Size = n(),              # 族群人數
      avg.Freq = mean(f),            # 平均購買次數
      avg.Revenue = sum(f*m)/sum(f)  # 平均客單價
  ) %>%
    ggplot(aes(y=avg.Freq, x=avg.Revenue)) +
    geom_point(aes(col=area, size=Group.Size), alpha=0.5) + #geom_point=添加散點圖，根據area設置顏色、根據Group.Size設
        定大小
    geom_text(aes(label=area)) +
    scale_size(range=c(5,25)) + #設置點的大小Range from 5 to 25
    theme_bw() + theme(legend.position="none") + #設置主題為全白，並且不顯示圖例
    ggtitle("地理區隔特徵 (泡泡大小:族群人數)") +
    ylab("平均購買次數") + xlab("平均客單價")+
    geom_vline(xintercept = sum(A$f*A$m)/sum(A$f), col="purple", linetype = "dashed")+
    geom_hline(yintercept = mean(A$f), col="darkcyan", linetype= "dashed")
```

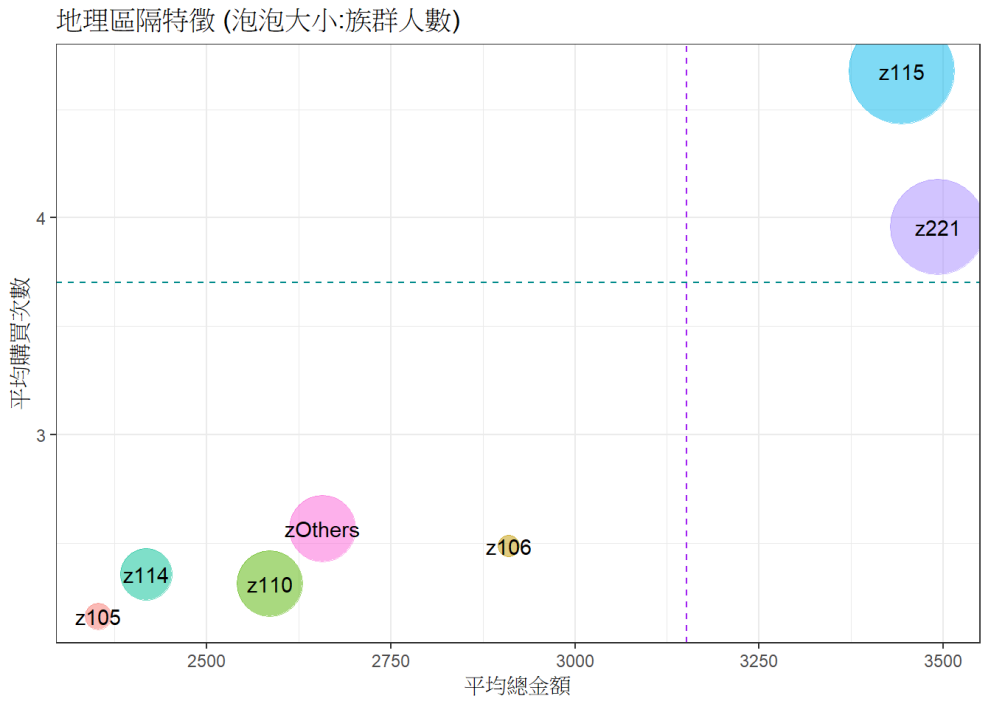


南港區的人平均購買次數高，但是平均客單價最低。大安區反之

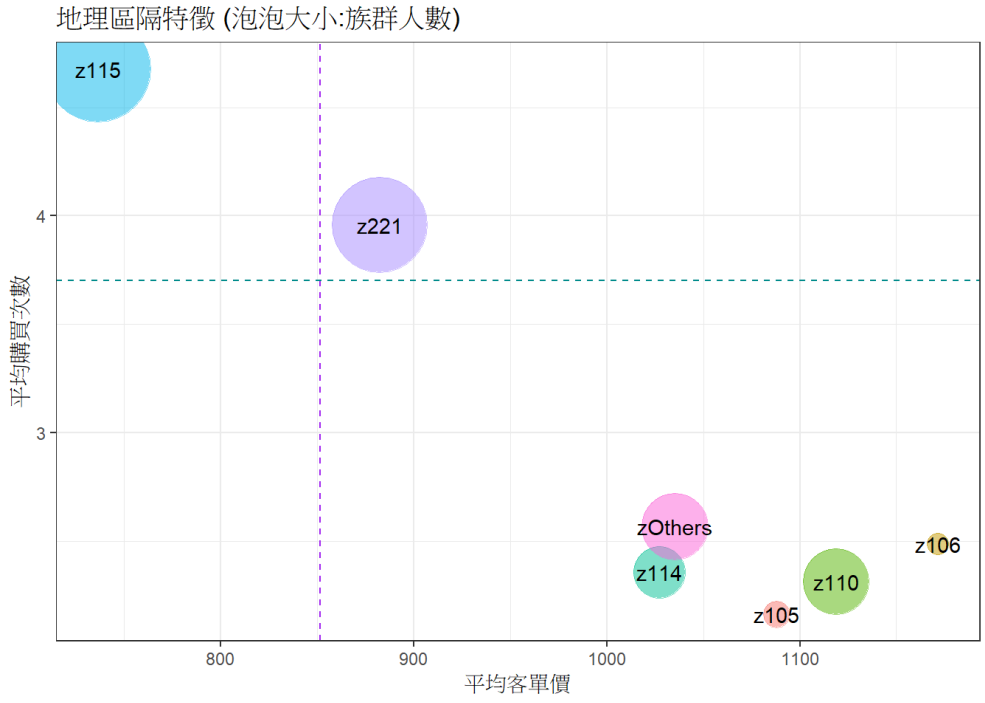
```
#濾掉沒有年齡資料和未知地區的顧客，以地區分群，針對最近一次購買日數、平均客單價作分析，並且以地區族群人數作為泡泡大小
A %>% filter(age!="None"& area!="zUnknown") %>%    # 濾掉沒有年齡資料和地區未知的顧客
  group_by(area) %>% summarize(
  Group.Size = n(),               # 族群人數
  avg.lastday = mean(r),            # 平均最近一次購買日
  avg.Revenue = sum(f*m)/sum(f)  # 平均客單價
  ) %>%
  ggplot(aes(y=avg.lastday, x=avg.Revenue)) +
  geom_point(aes(col=area, size=Group.Size), alpha=0.5) +
  geom_text(aes(label=area)) +
  scale_size(range=c(5,25)) +
  theme_bw() + theme(legend.position="none") +
  ggtitle("地理區隔特徵 (泡泡大小:族群人數)") +
  geom_vline(xintercept = sum(A$f*A$m)/sum(A$f),col="purple",linetype = "dashed")+
  geom_hline(yintercept = mean(A$r), col="darkcyan", linetype="dashed")+
  ylab("平均最近一次購買日") + xlab("平均客單價")
```




```
#濾掉沒有年齡資料和未知地區的顧客，以地區分群，針對平均購買次數率、平均總金額作分析，並且以地區族群人數作為泡泡大小

A %>% filter(age!="None"&area!="zUnknown") %>%    # 濾掉沒有年齡資料和地區未知的顧客
  group_by(area) %>% summarize(
  Group.Size = n(),               # 族群人數
  avg.freq = mean(f),           # 平均購買頻率
  avg.total = mean(rev)  # 平均總金額
  ) %>%
  ggplot(aes(y=avg.freq, x=avg.total)) +
  geom_point(aes(col=area, size=Group.Size), alpha=0.5) +
  geom_text(aes(label=area)) +
  scale_size(range=c(5,25)) +
  theme_bw() + theme(legend.position="none") +
  ggtitle("地理區隔特徵 (泡泡大小:族群人數)") +
  geom_vline(xintercept = mean(A$rev),col="purple",linetype = "dashed")+
  geom_hline(yintercept = mean(A$f), col="darkcyan", linetype="dashed")+
  ylab("平均購買次數") + xlab("平均總金額")
```

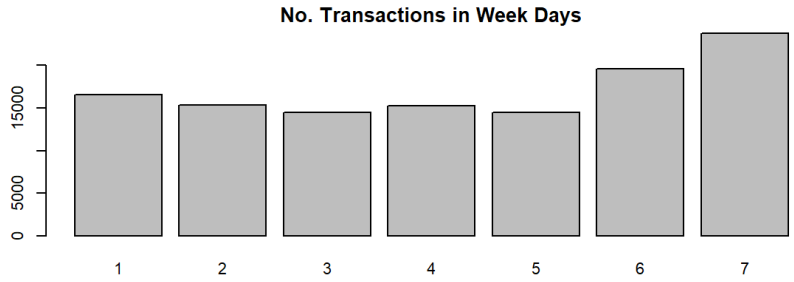



```
#濾掉沒有年齡資料和未知地區的顧客，以地區分群，針對平均購買次數、平均客單價作分析，並且以地區族群人數作為泡泡大小
A %>% filter(age!="None"&area!="zUnknown") %>%    # 濾掉沒有年齡資料和地區未知的顧客
  group_by(area) %>% summarize(
  Group.Size = n(),              # 族群人數
  avg.Freq = mean(f),            # 平均購買次數
  avg.Revenue = sum(f*m)/sum(f)  # 平均客單價
  ) %>%
  ggplot(aes(y=avg.Freq, x=avg.Revenue)) +
  geom_point(aes(col=area, size=Group.Size), alpha=0.5) +
  geom_text(aes(label=area)) +
  scale_size(range=c(5,25)) +
  theme_bw() + theme(legend.position="none") +
  ggtitle("地理區隔特徵 (泡泡大小:族群人數)") +
  geom_vline(xintercept = sum(A$f*A$m)/sum(A$f),col="purple",linetype = "dashed")+
  geom_hline(yintercept = mean(A$f), col="darkcyan", linetype="dashed")+
  ylab("平均購買次數") + xlab("平均客單價")
```

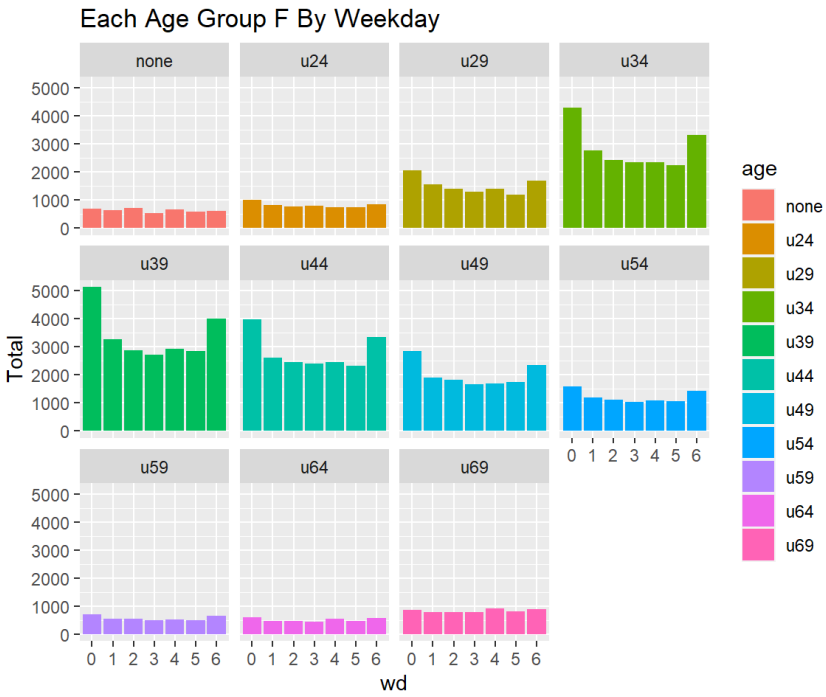


週內交易量分析 Transactions in Week Days

```
#週一到周日交易量分布
X$wday = format(X$date, "%u")
par(cex=0.7, mar=c(2,3,2,1))
table(X$wday) %>%
    barplot(main="No. Transactions in Week Days")
```



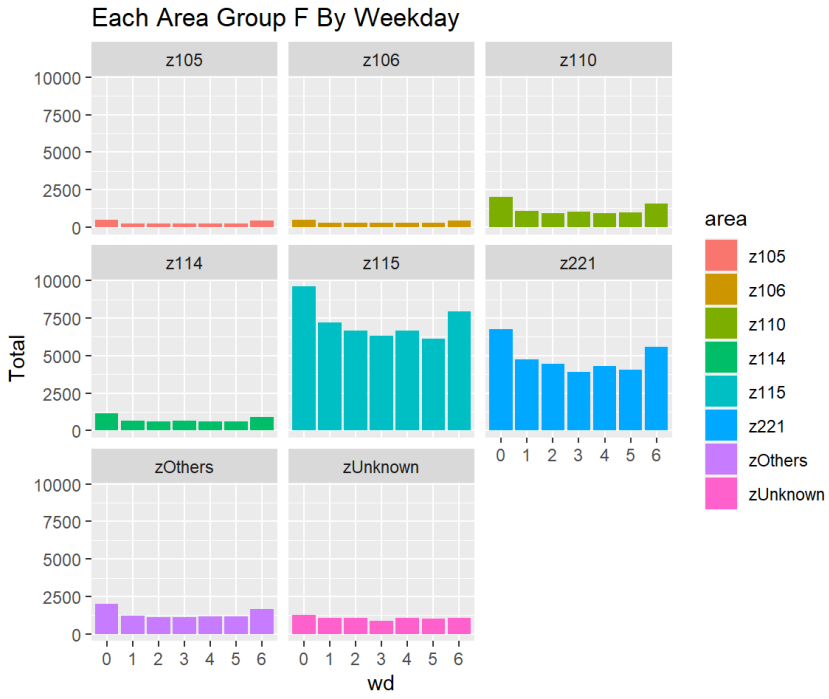

```
#以年齡分群，將週一到周日交易量分布視覺化輸出
X %>%
  mutate(wd=factor(format(date, "%w"))) %>%
    count(age, wd) %>%
    ggplot(aes(x=wd, y=n,fill=age)) +
    geom_bar(stat="Identity") +
    labs(title="Each Age Group F By Weekday", y="Total")+
    facet_wrap(~age) #根據age將圖表分成多個子圖
```




```
#下圖橫軸順序皆為日一二三...六
```

除了年紀較大和最小的族群以外，其他傾向於假日(可能因為要上班)

```
#以地區分群，將週一到周日交易量分布視覺化輸出
X %>%
  mutate(wd=factor(format(date, "%w"))) %>%
    count(area, wd) %>%
    ggplot(aes(x=wd, y=n,fill=area)) +
    geom_bar(stat="Identity") +
    labs(title="Each Area Group F By Weekday", y="Total")+
    facet_wrap(~area) #根據area將圖表分成多個子圖
```




```
#下圖橫軸順序皆為日一二三...六
```

南港汐止數量明顯高太多了，基本上可以確定這家店就在南港靠近汐止的地方。
至於週內分布整體而言也以六日居多

商品種類分群分析 Product Analysis

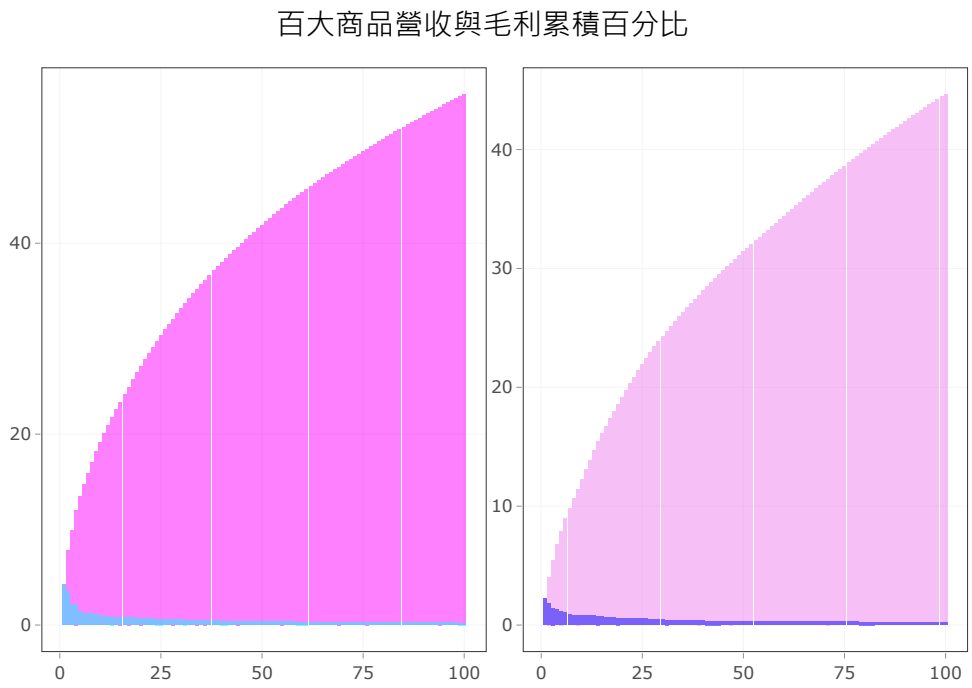
```
#根據商品種類分群並計算Total qunatity, total revenue, total profit, etc.
dfbycat = df %>% group_by(cat) %>% summarize(
  noProd = n_distinct(prod),
  Tqty = sum(qty),
  Trev = sum(price),
  Traw = sum(price) - sum(cost),
  GPM = Traw/Trev,  #Gross Profit Margin
  Weightedprice = round(Trev/Tqty,2)
  )
str(dfbycat)
```

```
## tibble [2,007 × 7] (S3: tbl_df/tbl/data.frame)
##  $ cat          : num [1:2007] 1e+05 1e+05 1e+05 1e+05 1e+05 ...
##  $ noProd       : int [1:2007] 29 136 52 35 9 40 7 31 28 35 ...
##  $ Tqty         : num [1:2007] 1696 11175 2470 1599 639 ...
##  $ Trev         : num [1:2007] 156913 820440 145361 73549 22609 ...
##  $ Traw         : num [1:2007] 24098 131083 29752 16434 4826 ...
##  $ GPM          : num [1:2007] 0.154 0.16 0.205 0.223 0.213 ...
##  $ Weightedprice: num [1:2007] 92.5 73.4 58.9 46 35.4 ...
```

```
head(dfbycat)
```

```
## # A tibble: 6 × 7
##      cat noProd  Tqty   Trev   Traw   GPM Weightedprice
##    <dbl>  <int> <dbl>  <dbl>  <dbl> <dbl>         <dbl>
## 1 100101     29  1696 156913  24098 0.154          92.5
## 2 100102    136 11175 820440 131083 0.160          73.4
## 3 100103     52  2470 145361  29752 0.205          58.8
## 4 100104     35  1599  73549  16434 0.223          46
## 5 100105      9   639  22609   4826 0.213          35.4
## 6 100106     40  3770 244784  42457 0.173          64.9
```

```
#百大商品營收與毛利之貢獻視覺化
Trev100 = arrange(dfbycat, desc(Trev)) %>%  #根據Trev排序
  mutate(Trev_ind=Trev/sum(Trev) *100, Trev_cum=cumsum(Trev_ind)) %>%  #
  head(100) %>%
    ggplot(aes(x=1:100)) +
    geom_col(aes(y=Trev_cum),fill="magenta",alpha=0.5) +
    geom_col(aes(y=Trev_ind), fill="cyan",alpha=0.5) +
    labs(title="百大商品貢獻營收", x="第n大商品", y="貢獻營收累計百分比") +
    theme_bw()

Traw100 = arrange(dfbycat, desc(Traw)) %>%  #根據Traw排序
  mutate(Traw_ind=Traw/sum(Traw) *100, Traw_cum=cumsum(Traw_ind)) %>%  #
  head(100) %>%
    ggplot(aes(x=1:100)) +
    geom_col(aes(y=Traw_cum),fill="violet",alpha=0.5) +
    geom_col(aes(y=Traw_ind), fill="blue",alpha=0.5) +
    labs(title="百大商品毛利營收", x="第n大商品", y="貢獻毛利累計百分比") +
    theme_bw()

subplot(Trev100, Traw100, nrows = 1) %>% layout(title = "百大商品營收與毛利累積百分比")
```

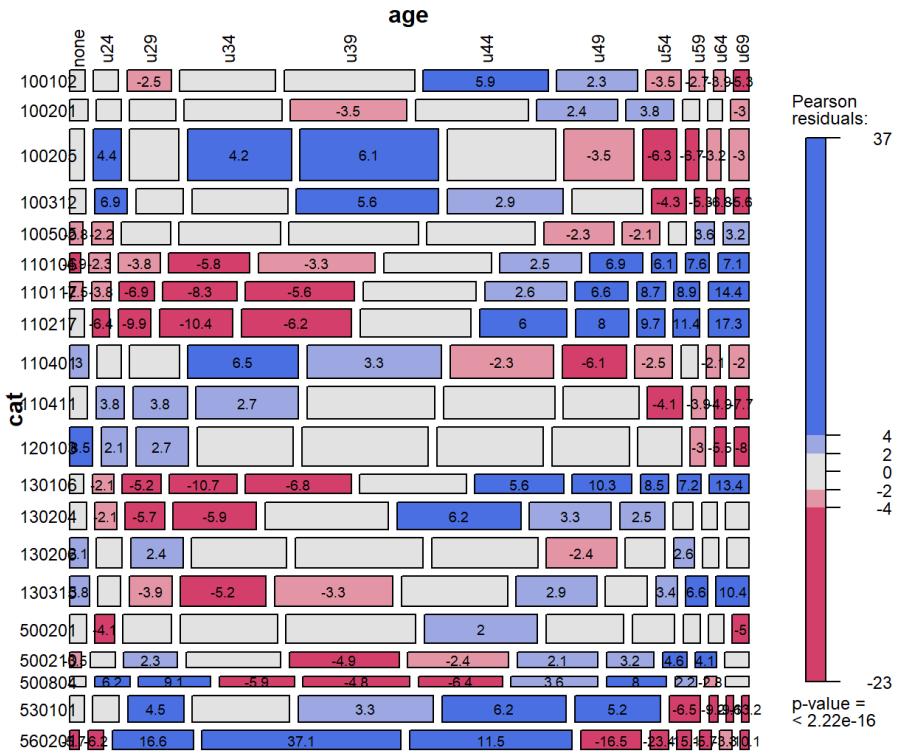


可以看見2007個商品中，前百大商品佔了將近一半的營收貢獻
前10大商品就佔了約19%的營收與12%的毛利
前20大商品佔了約27%的營收與19%的毛利
前50大商品佔了約42%的營收與31%的毛利
->顯現主力商品以薄利多銷為主

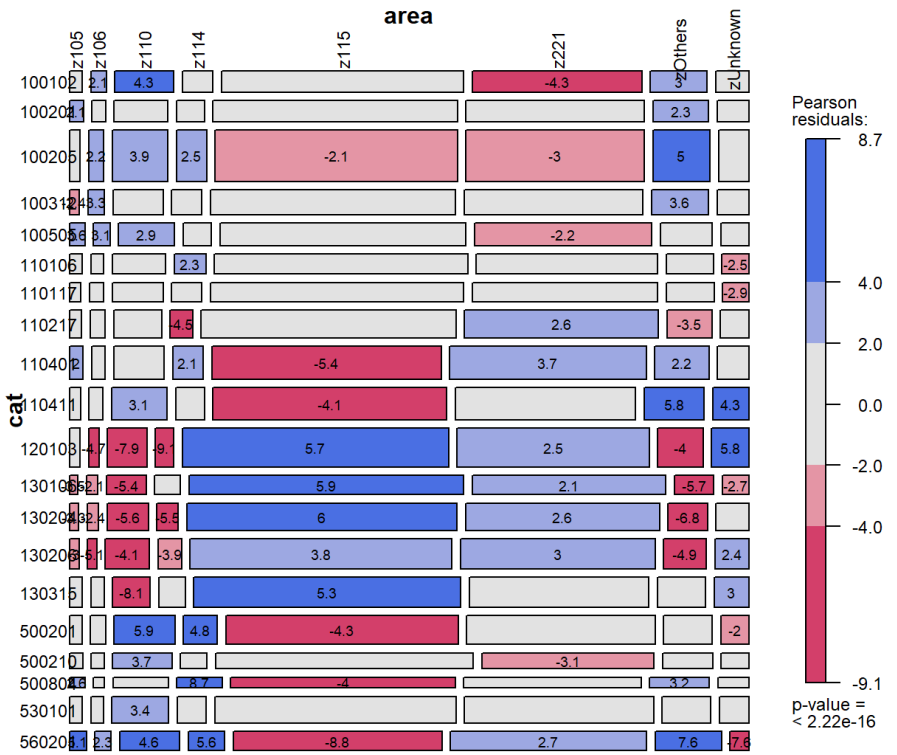
```
#前20大熱門的商品
top20 = tapply(df$qty,df$cat,sum) %>% sort %>% tail(20) %>% names
top20
```

```
##  [1] "500210" "100201" "110117" "130106" "100102" "500804" "130204" "100312"
##  [9] "110106" "110217" "530101" "130206" "100505" "560201" "110401" "500201"
## [17] "130315" "120103" "110411" "100205"
```

```
#前20大品項與年齡的關係
MOSA(~cat+age, df[df$cat %in% top20,])
```




```
#前20大品項與地區的關係
MOSA(~cat+area, df[df$cat %in% top20,])
```




```
# ggplot(df, aes(x = df[df$cat %in% top20,], y = area, color = age)) +
#   geom_point(size = 3) +
#   labs(title = "訂單分布三點圖", x = "商品類型 (cat)", y = "客戶地址代碼 (area)") +
#   theme_minimal()
```

4. RFM矩陣 規則分群 RFM Model

RFM模型：r（recency）最近一次消費：若顧客上次消費紀錄的時間愈近則價值愈大。f（frequency）消費頻率：顧客在所選期間中有幾次消費紀錄？頻率愈高則價值愈大。 m（monetary消費金額）：顧客的總消費額，即為公司貢獻了多少利潤？金額愈高則價值愈大。

```
#定義函數
hmap1 = function(x, ...) { heatmaply(
  as.data.frame.matrix(x), cexRow=0.7, cexCol=0.7,
  grid_color='gray70', ...)
  }
```

```
#在顧客資料框加入規則分群欄位
bm = c(0, quantile(A$m,c(.25,0.5,.75)), max(A$m)+100)
bf = c(0, quantile(A$f,c(.25,0.5,.75)), max(A$f)+100)
A = A %>% mutate(
  mx = cut(A$m, bm, labels=paste0('M',1:4)),
  fx = cut(A$f, bf, labels=paste0('F',1:4)),
  MF = paste0(mx, fx)
  )
table(A$mx, A$fx)
```

```
##
##        F1   F2   F3   F4
##   M1 3465 1477 1382 1750
##   M2 2477 1479 1571 2541
##   M3 2562 1553 1804 2128
##   M4 3388 1790 1695 1179
```

```
#找出營收最大的品類
cat100 = count(df, cat, wt=price, sort=T) %>% mutate(
  pc=n/sum(n), cum.pc=cumsum(pc)) %>% head(100)
cat100[c(1:5,96:100), ]
```

```
##        cat       n          pc     cum.pc
## 1   560201 4329366 0.042202618 0.04220262
## 2   560402 3634174 0.035425894 0.07762851
## 3   500201 2204325 0.021487739 0.09911625
## 4   110217 2201258 0.021457842 0.12057409
## 5   320402 1481172 0.014438451 0.13501254
## 96  100504  229815 0.002240234 0.54720217
## 97  110106  227899 0.002221557 0.54942373
## 98  100418  226905 0.002211868 0.55163559
## 99  100407  224486 0.002188287 0.55382388
## 100 110402  221145 0.002155719 0.55597960
```

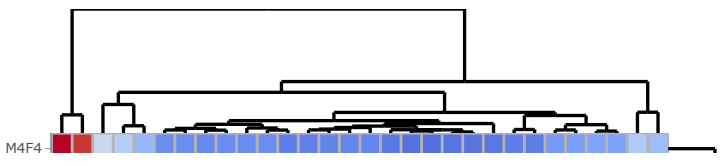
```
#做出顧客族群x品類 購買金額矩陣
df = inner_join(df, A[,c('cust','MF')])
```

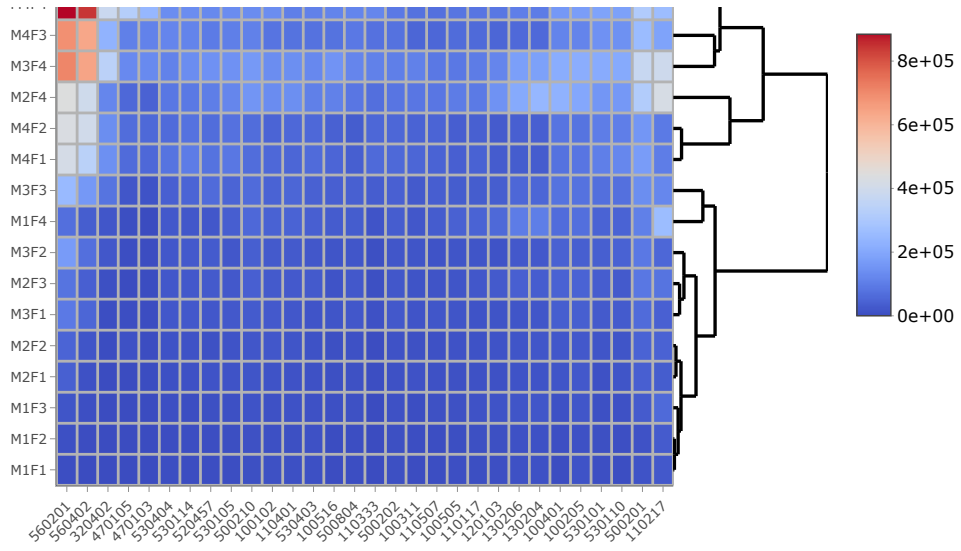
```
## Joining with `by = join_by(cust)`
```

```
mx0 = xtabs(price~MF+cat, filter(df, cat %in% cat100$cat[1:30]))
dim(mx0)
```

```
## [1] 16 30
```

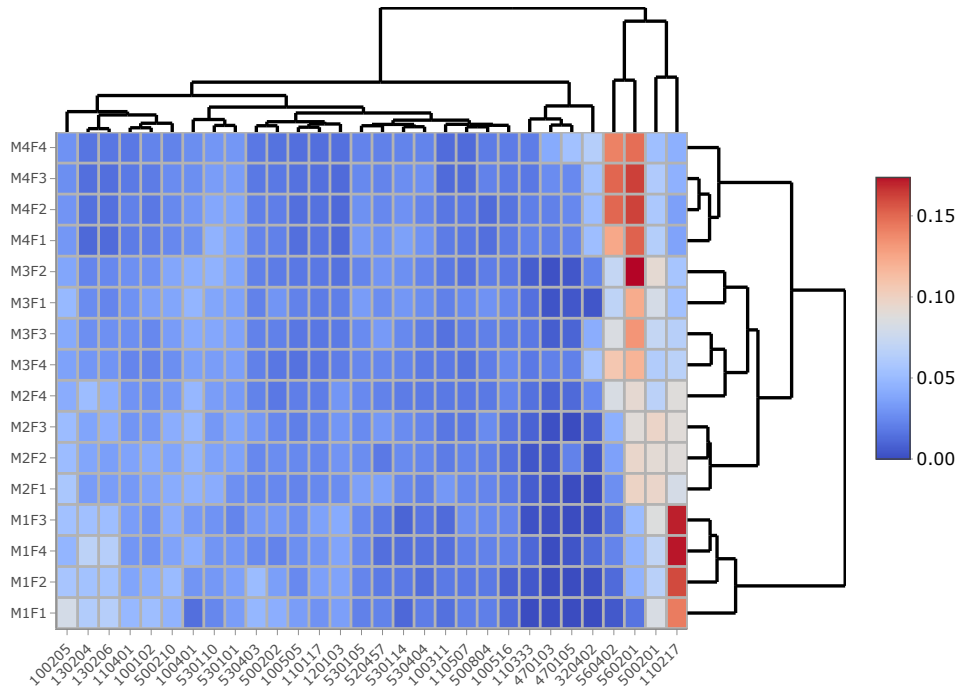
```
#依購買金額矩陣製作熱圖
hmap1(mx0, col=cool_warm)
```



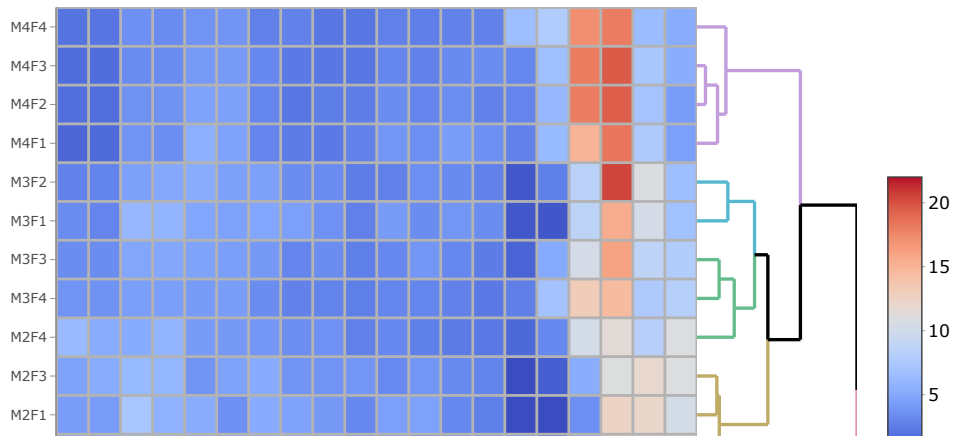


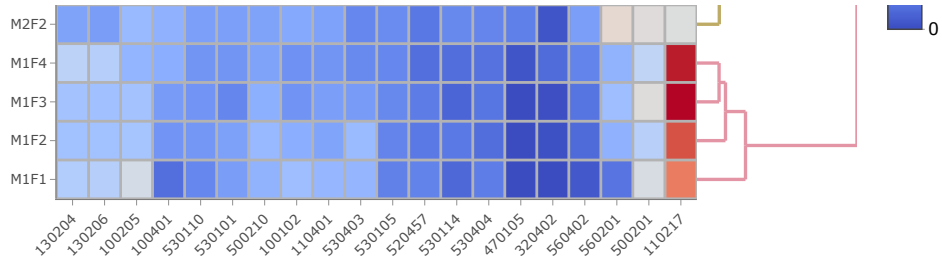
正規化 -購買比例矩陣

```
mx1 = mx0/rowSums(mx0)
hmap1(mx1, col=cool_warm)
```




```
#熱圖的分群功能
mx2 = xtabs(price~MF+cat, filter(df, cat %in% cat100$cat[1:20]))
mx3 = 100*mx2/rowSums(mx2)
hmap1(mx3, col=cool_warm, show_dendrogram=c(T,F),k_row=5)
```






```
#依據顧客族群與商品品類的購買金額矩陣
df0 = inner_join(df, A[,c('cust','MF')])
```

```
## Joining with `by = join_by(cust, MF)`
```

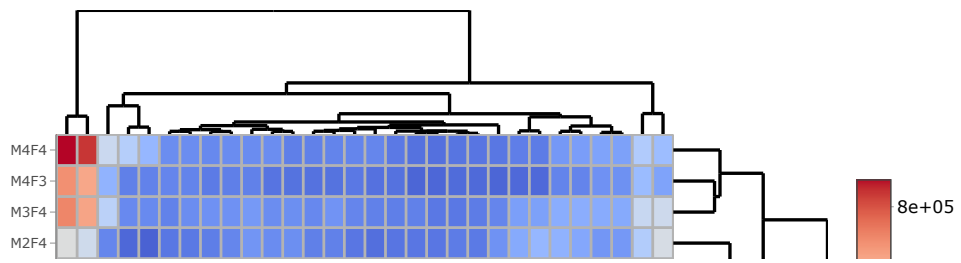
```
mx0 = xtabs(price~MF+cat, filter(df, cat %in% cat100$cat[1:30]))
dim(mx0)
```

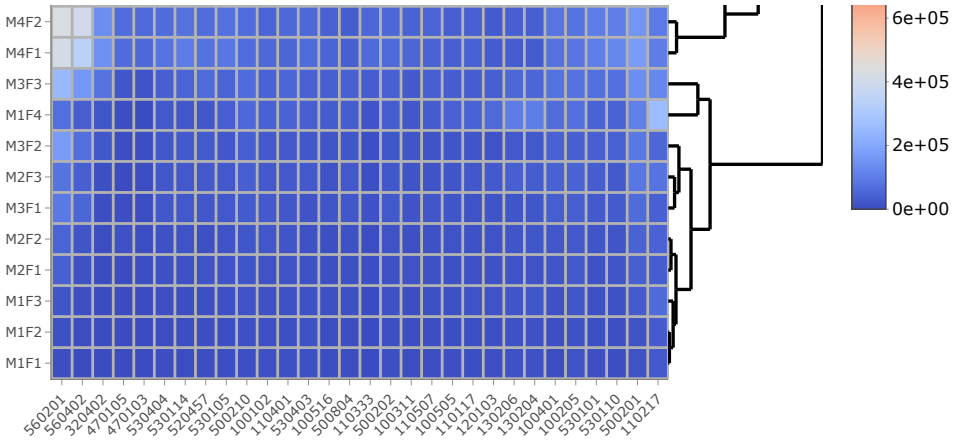
```
## [1] 16 30
```

```
print(mx0)
```

```
##       cat
## MF     100102 100205 100311 100401 100505 100516 110117 110217 110333 110401
##   M1F1   8838  13761   2510   2365   5843   2106   4934  24421      0   8243
##   M1F2   8249  10736   3474   5751   4747   1517   6750  30304    856   7187
##   M1F3   9925  18373   4151  11131   9248   7907  12421  58318    696  11567
##   M1F4  42906  69849  21622  64689  41193  32028  45471 258828  15920  45730
##   M2F1  14625  23052  12414  17914   9020   7293   9875  32233   2638  12829
##   M2F2  20112  24897   9449  22476  12191   6944  11641  43623   1767  18202
##   M2F3  27733  45010  15326  43022  18038  14081  20862  79021   7730  26779
##   M2F4 131924 199155  85682 227739  96406 108992  92948 420173  69539 138787
##   M3F1  26768  36766  15916  35089  16487  17952  14499  40675  10819  21518
##   M3F2  27501  36914  17417  41426  17250  17314  16969  54455   6568  27037
##   M3F3  46334  76497  28164  77411  32825  37980  32280 122668  33798  52384
##   M3F4 135604 215115 108246 213700  89340 148272  97373 393548 108446 138221
##   M4F1  54862  83818  51619  76446  37804  50495  43142 100050  59826  52023
##   M4F2  47215  79230  45432  80947  36393  41276  39646  93859  54262  56189
##   M4F3  81093 115364  52319 115243  63516  77863  60426 186007  74982  77688
##   M4F4 135460 171807  76147 161189  82224 115087  82721 260531 118940 104585
##       cat
## MF     110507 120103 130204 130206 320402 470103 470105 500201 500202 500210
##   M1F1   3510   5987  10629  10976      0    135      0  14252   7307   7766
##   M1F2   3356   6993  10390  10445    342      0      0  12493   6619   9440
##   M1F3   9455  13934  18251  17797    385    384      0  29572  10959  14528
##   M1F4  31484  56557 101840  96604  18754    270   4995 105291  33596  55780
##   M2F1   9861  10550  13401  13494      0    945      0  38117   9734  16481
##   M2F2  11088  14599  18727  17115   1719   1800  10557  44647  12899  18531
##   M2F3  15602  27084  33576  37827   6214   1485      0  85936  21671  37399
##   M2F4  85178 145943 244133 204810 119167  43369  55060 316641  83360 152347
##   M3F1  19127  15377  19608  17573   2722   2315   2988  61329  22789  28947
##   M3F2  14303  14573  22567  23624  21806   1836   4270  89243  18810  37276
##   M3F3  36331  35254  52205  52976  80263  13231  18648 136401  40276  63095
##   M3F4  90862 118165 181832 177072 338318 127726 127430 372090 103346 163176
##   M4F1  46069  30773  30485  32561 140688  55994  61718 171202  56916  66087
##   M4F2  47468  29987  38965  37840 136722  56926  65393 156320  53731  66021
##   M4F3  54241  52047  57347  57750 231429 109330 105486 256533  75745 109051
##   M4F4  73631  86769  96617 102516 380339 243370 323317 313300  91937 132191
##       cat
## MF     500804 520457 530101 530105 530110 530114 530403 530404 560201 560402
##   M1F1   3499   3794   5831   3601   4089   1880   8128   3396   2733    866
##   M1F2   3428   3493   6820   4390   5989   3459   9585   2549   8614   2278
##   M1F3   8676   6360   7850   8292   9980   3329  11067   5566  17305   5384
##   M1F4  32497  20495  45863  36374  45407  20401  38208  22586  68722  34346
##   M2F1   9197  14412  11154  14058  16600   9680   9881   8234  39070  10915
##   M2F2  10044   8925  17748  13069  17750  12513  12004  11789  46891  17623
##   M2F3  22636  27883  33763  22810  27688  23224  27854  20486  79065  38561
##   M2F4  85346 102276 155335 119036 161155  91100 105096  87862 440977 395070
##   M3F1  21848  20142  27232  25136  28952  23357  16290  20254  92433  51565
##   M3F2  19120  28767  36856  22117  43238  26789  20559  22418 168309  69592
##   M3F3  39345  61519  69062  48566  73606  47143  41382  37926 251225 160887
##   M3F4 118198 144650 209501 144397 204379 135260 125581 110895 705226 639541
##   M4F1  36991  77717 103668  86930 121605  96961  61648  79013 411030 336043
##   M4F2  32136  65624 100320  74907 103055  75536  55394  61803 430239 398902
##   M4F3  90790  96023 144090 104225 143773 113944  75480 117300 683142 632498
##   M4F4 116443 125865 185739 133446 181903 135660 107134 135702 884385 840103
```

```
hmap1 = function(x, ...) { heatmaply(
  as.data.frame.matrix(x), cexRow=0.7, cexCol=0.7,
  grid_color='gray70', ...)
  }
hmap1(mx0, col=cool_warm)
```





##顏色越深，表示價格在對應的產品與級距中占比较高，顏色越淺，表示價格低占比较高(?)

```
#依據顧客族群與商品品類的購買金額矩陣
df = inner_join(df, A[,c('cust','MF')])
```

```
## Joining with `by = join_by(cust, MF)`
```

```
A$age_1 = as.character(A$age)
# 刪除age的'u'
A$age_1 = as.numeric(sub("u", "", A$age_1))
```

```
## Warning: NAs introduced by coercion
```

```
df = inner_join(df, A[, c('cust', 'MF', 'age_1')])
```

```
## Joining with `by = join_by(cust, MF)`
```

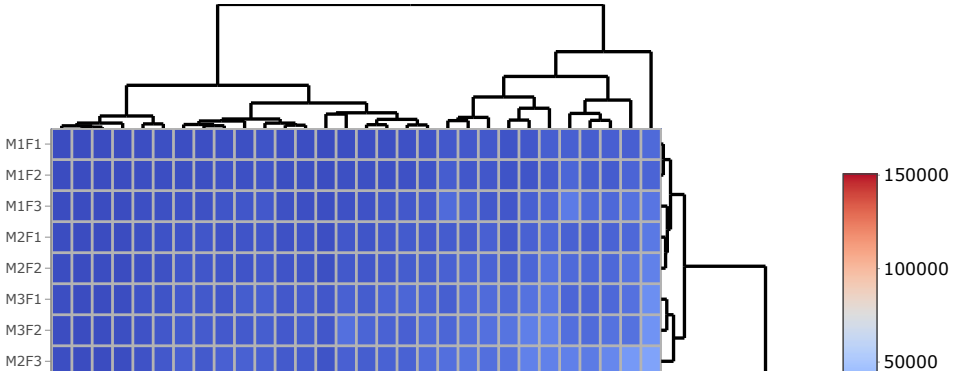
```
mx1 = xtabs(age_1~MF+cat, filter(df, cat %in% cat100$cat[1:30]))
dim(mx1)
```

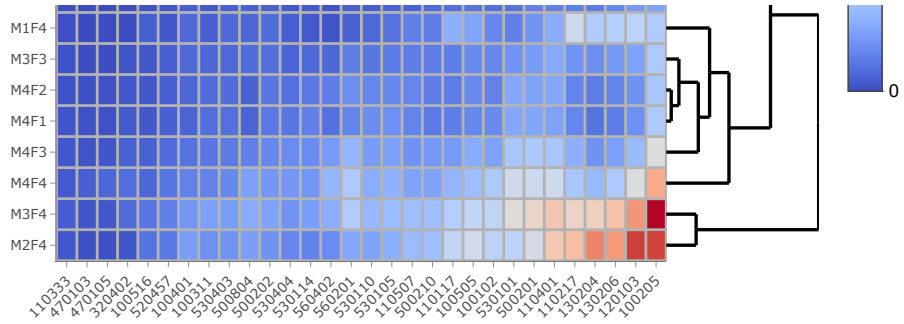
```
## [1] 16 30
```

```
print(mx1)
```

```
##       cat
## MF     100102 100205 100311 100401 100505 100516 110117 110217 110333 110401
##   M1F1   4335  10327   1009    488   4057    641   4094   6647      0   6309
##   M1F2   3609   8244   1070    766   3519    510   5706   8926     68   5070
##   M1F3   4386  14201   1482   1796   6992    987  10433  16352     49   8612
##   M1F4  21078  53246   6496   9446  32270   6250  37932  66215   1018  35225
##   M2F1   6872  15549   3326   2216   5105   1131   5237   7092    244   9226
##   M2F2   7971  18351   3120   3243   8348    947   7185  10214    201  13028
##   M2F3  11831  31912   4600   6390  13234   2074  12085  18103    530  18297
##   M2F4  58773 139878  24123  29628  68075  13700  61461  97704   4242  91613
##   M3F1  10709  23675   4318   4457   9817   2363   6307   8658    718  14700
##   M3F2  12276  25145   5032   5017  11363   2193   8834  11923    405  18399
##   M3F3  21172  52098   9198   9452  22030   3789  17703  25638   1767  34431
##   M3F4  60257 150644  30942  26889  61707  14222  55268  84196   5723  93150
##   M4F1  22093  51719  11727   8670  21353   4246  17449  20434   2868  31207
##   M4F2  18615  49925  11114   9301  21752   3614  16672  19602   2291  33470
##   M4F3  31320  75280  14509  13597  35303   7299  28285  37274   3647  49458
##   M4F4  51519 105895  19208  18815  44763   9147  40372  49733   4859  66710
##       cat
## MF     110507 120103 130204 130206 320402 470103 470105 500201 500202 500210
##   M1F1   1767   6512   6013   6483      0     44      0   3472   2541   2269
##   M1F2   1821   8251   5836   5358     34      0      0   2809   2383   2512
##   M1F3   4095  12947  10281   9532     44     44      0   6731   3984   4525
##   M1F4  18432  58703  52934  55014    345     58    220  26206  11101  17290
##   M2F1   4978   9568   6535   7754      0    147      0   7154   3175   4462
##   M2F2   5533  13293   9217   9524     93    103    170   8879   4176   5607
##   M2F3   7277  27471  16105  20740    264    132      0  18358   6777  10476
##   M2F4  43953 140734 120718 113507   3277   1458   1507  70152  25025  45252
##   M3F1   8812  12801   8711   9347    201    146    117  13102   6270   7733
##   M3F2   6889  12951  10645  12858    883     83    181  17824   6044   9717
##   M3F3  17178  31503  23724  27891   2878    541    633  28872  12043  16636
##   M3F4  45757 114611  87413  95777  10561   3150   3660  83335  31644  45594
##   M4F1  19573  24596  13471  16972   4717   1725   1945  33363  15423  15922
##   M4F2  20407  24849  16588  19357   4221   1444   1855  32087  14668  16413
##   M4F3  24665  43370  24910  30534   7807   2688   3322  51143  21107  27162
##   M4F4  32107  73848  42552  51514  11584   6243   9098  65724  26886  32183
##       cat
## MF     500804 520457 530101 530105 530110 530114 530403 530404 560201 560402
##   M1F1   1044   1091   2512   1268    935    611   1793    895    448     97
##   M1F2   1412    686   2571   1465   1159    792   2198    783   1074    278
##   M1F3   3138   1194   3466   3255   2112    889   2951   1581   1790    473
##   M1F4   9260   4277  18501  13109   9430   4855  10001   6022   7623   2571
##   M2F1   2733   2013   4251   4160   2980   2259   2125   1863   3734   1027
##   M2F2   2683   1546   6319   4434   3745   2610   2914   2786   4337   1119
##   M2F3   7217   4194  12933   7380   5929   4995   6810   4663   6819   2604
##   M2F4  30241  15458  59430  37463  32082  19181  25652  20448  33769  22523
##   M3F1   5483   2590   9669   7618   5590   5033   3372   4279   7171   3573
##   M3F2   4634   3715  13543   7525   8557   5546   4504   4540  12335   3740
##   M3F3  11081   6471  25280  15968  14623   9881   9458   8296  17621   9134
##   M3F4  35349  18467  76923  43624  42089  29861  29672  25225  52950  37032
##   M4F1   8223   6798  33675  24346  22983  18495  12240  15015  19584  12818
##   M4F2   7691   6819  34236  20902  20373  15421  11344  12344  23493  17231
##   M4F3  18576  10457  49630  30979  29163  22446  16176  22350  39933  28510
##   M4F4  30968  14686  67384  38477  36435  26385  21459  26867  52128  40744
```

```
hmap1(mx1, col=cool_warm)
```






```
#顏色越深，表示年齡大的客戶在對應的產品與級距中占比較高，顏色越淺，表示年齡小的客戶占比較高(?)
```

5. R(S)FM與集群分析： RSFM Model with Clustering Analysis

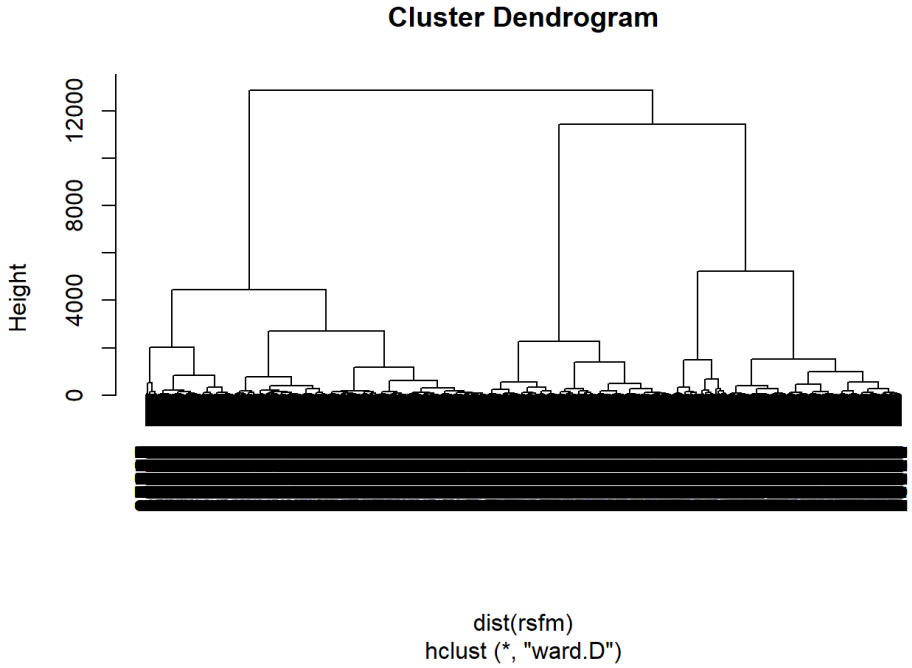
Ref:https://rpubs.com/skydome20/R-Note9-Clustering (https://rpubs.com/skydome20/R-Note9-Clustering)

在傳統的RFM模型加上新的變數，再將數據進行標準化後，利用K-Means針對 r(最近一次消費時間)、s(第一次購買日期)、f(消費頻率)、m(消費金額)四項特徵做階層式集群分析，將數萬名消費者依據四維特性分群。

```
rsfm = scale(A[,2:5]) %>% data.frame #Standardizing rsfm
head(rsfm)
```

```
##           r          s           f          m
## 1 -0.5478720  0.7985333  0.06190149 -0.5285849
## 2  0.4913842  0.8278659  0.06190149 -0.4535415
## 3 -0.5478720 -1.6360749 -0.35232969 -0.2096502
## 4  1.4712543  0.1825481 -0.55944528 -0.6557419
## 5 -0.0430904  1.1211922 -0.14521410 -0.1293954
## 6  0.5804633  0.2412134 -0.35232969  6.2899472
```

```
#Hierarchical method
rsfmcluster = hclust(dist(rsfm), method='ward.D')# Euclidean method to get distance matrix, Ward Method (ANOVA) to
        get cluster
plot(rsfmcluster)
```



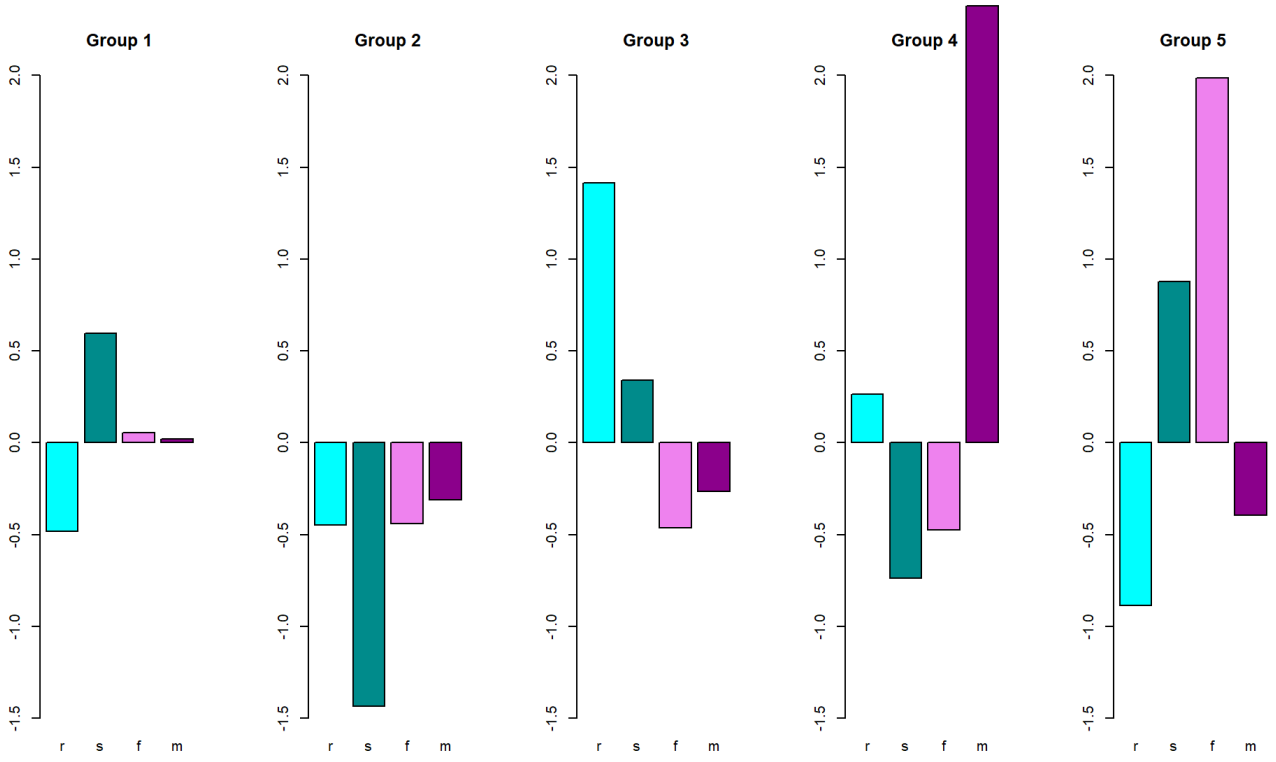

```
#Make 5 cluster(製作分群向量)
set.seed(11)
rsfm5 = cutree(rsfmcluster, k=5)
```

```
#r,s,f,m 的平均
rsfmtable = split(rsfm,rsfm5) %>% sapply(colMeans) %>% round(4)  %>% data.frame()
rsfmtable
```

```
##        X1      X2      X3      X4      X5
## r -0.4812 -0.4482  1.4167  0.2642 -0.8854
## s  0.5954 -1.4329  0.3425 -0.7374  0.8767
## f  0.0547 -0.4392 -0.4612 -0.4735  1.9873
## m  0.0194 -0.3115 -0.2659  2.3792 -0.3932
```

各客群的r,s,f,m視覺化

```
# par(cex=0.8)
#  split(rfm,rfm5) %>%
#    sapply(colMeans) %>% barplot(beside=T,col = c("cyan", "darkcyan", "violet","darkmagenta"))
#    legend('topright',legend=colnames(rfm),fill =c("cyan", "darkcyan", "violet","darkmagenta"))
color_list = c("cyan", "darkcyan", "violet", "darkmagenta")
par(mfrow = c(1, 5))
rsfm_means <- split(rsfm, rsfm5) %>% sapply(colMeans)

for (i in 1:5) {
  barplot(rsfm_means[, i], beside = TRUE, col = color_list,
          main = paste("Group", i), names.arg = colnames(rsfm),
          ylim = c(-1.5, 2))
}
```



我們可以將顧客分出五群：

Group1：一般顧客 (normal) r低、s高、f居中、m居中 –>最近購買的天數低於平均，可能表示他們是比較忠誠的顧客。同時，他們的購買歷史最早的天數高於平均，這可能意味著他們已經是長期的忠誠客戶。他們的購買次數和平均購買金額約為平均水平，可能是穩定且有中等消費水平的顧客。

Group2：新進顧客 (new) r低、s非常低、f低、m略低 –>最近購買的天數低於平均，可能也是相對忠誠的顧客。然而，他們的購買歷史最早的天數非常低於平均，這意味著他們可能是相對新的顧客。他們的購買次數低於平均，平均購買金額略低於平均，可能是平均貢獻銷售額相對較小的忠誠顧客。

Group3：沉默顧客 (silence) r非常高、s略高、f低、m略低 –>最近購買的天數遠高於平均，可能是不太活躍的顧客。然而，他們的購買歷史最早的天數略高於平均，這可能表示他們在較早的時間點曾經是活躍的顧客。他們的購買次數低於平均，平均購買金額也略低於平均，可能是不太活躍且有中等消費水平的顧客。

Group4：高購買力顧客 (highp) –> #重要發展客戶 r略高、s低、f低、m非常高 –>顧客最近購買的天數略高於平均，但他們的購買歷史最早的天數低於平均，這可能表示他們在較早的時間點曾經是活躍的顧客，但現在變得不太活躍。他們的購買次數低於平均，但平均購買金額非常高於平均，這可能是高價值但不太活躍的顧客。

Group5：長期忠誠顧客 (longterm) –>#一般保持客戶 r低、、s高、f非常高、m低 –>顧客最近購買的天數低於平均，可能是比較忠誠的顧客。他們的購買歷史最早的天數高於平均，這可能意味著他們已經是長期的忠誠客戶。然而，他們的購買次數遠遠高於平均，但平均購買金額低於平均，這可能是頻繁購買但購買金額不高的顧客。

```
#Non-hierarchical method: K-means
set.seed(11) #確保結果可復現
A$rsfm_group = kmeans(rsfm, centers= 5)$cluster #幫每個客人貼r,s,f,m的標籤!
head(A)
```

```
##       cust  r   s f    m   rev  raw  age    area age_1  area_1 mx fx   MF
## 1 00001069 19 108 4  486  1944   15 none    z115    NA    z115 M2 F3 M2F3
## 2 00001113 54 109 4  558  2230  241 none    z221    NA    z221 M2 F3 M2F3
## 3 00001250 19  25 2  792  1583  354  u39    z114    39    z114 M3 F2 M3F2
## 4 00001359 87  87 1  364   364  104 none zOthers    NA zOthers M1 F1 M1F1
## 5 00001823 36 119 3  869  2607  498 none    z114    NA    z114 M3 F3 M3F3
## 6 00002189 57  89 2 7028 14056 3299 none    z106    NA    z106 M4 F2 M4F2
##   rsfm_group
## 1          4
## 2          4
## 3          3
## 4          1
## 5          4
## 6          2
```

```
# # of clients in 各客群
table(A$rsfm_group)
```

```
##
##     1     2     3     4     5
##  7682  2713  8299 12375  1172
```

```
#將資料分進各自客群
normal = split(A,A$rsfm_group)[[1]]
new = split(A,A$rsfm_group)[[2]]
silence = split(A,A$rsfm_group)[[3]]
highp = split(A,A$rsfm_group)[[4]]
longterm = split(A,A$rsfm_group)[[5]]
```

6.製作預測變數 Preparing The Predictors (X)

Importing Data

```
load("data/tf0.rdata")
```

Preprocessing

Remove the data in the last period (After Feb 01).

```
feb01 = as.Date("2001-02-01")   # 資料分割日期
Zs = subset(Z0, date < feb01)    # 618212 項目
```

Aggregate for the Transaction Records

重新匯整交易紀錄，如前述的內容

```
Xs = group_by(Zs, tid) %>% summarise(
  date = first(date),  # date of transaction
  cust = first(cust),  # customer id
  age = first(age),    # age group
  area = first(area),  # area group
  items = n(),               # number of items
  pieces = sum(qty),         # number of pieces
  total = sum(price),        # total amount
  gross = sum(price - cost)  # raw profit
  ) %>% data.frame  # 88387 交易筆數
```

```
summary(Xs)
```

```
##       tid             date               cust              age
##  Min.   :    1   Min.   :2000-11-01   Length:88387       Length:88387
##  1st Qu.:22098   1st Qu.:2000-11-23   Class :character   Class :character
##  Median :44194   Median :2000-12-12   Mode  :character   Mode  :character
##  Mean   :44194   Mean   :2000-12-15
##  3rd Qu.:66291   3rd Qu.:2001-01-12
##  Max.   :88387   Max.   :2001-01-31
##      area               items             pieces            total
##  Length:88387       Min.   :  1.000   Min.   :  1.000   Min.   :    5.0
##  Class :character   1st Qu.:  2.000   1st Qu.:  3.000   1st Qu.:  230.0
##  Mode  :character   Median :  5.000   Median :  6.000   Median :  522.0
##                     Mean   :  6.994   Mean   :  9.453   Mean   :  888.7
##                     3rd Qu.:  9.000   3rd Qu.: 12.000   3rd Qu.: 1120.0
##                     Max.   :112.000   Max.   :339.000   Max.   :30171.0
##      gross
##  Min.   :-1645.0
##  1st Qu.:   23.0
##  Median :   72.0
##  Mean   :  138.3
##  3rd Qu.:  174.0
##  Max.   : 8069.0
```

Check Quantile and Remove Outlier

移除離群值

```
sapply(Xs[,6:9], quantile, prob=c(.999, .9995, .9999))
```

```
##          items   pieces     total    gross
## 99.9%  56.0000  84.0000  9378.684 1883.228
## 99.95% 64.0000  98.0000 11261.751 2317.087
## 99.99% 85.6456 137.6456 17699.325 3389.646
```

```
Xs = subset(Xs, items<=64 & pieces<=98 & total<=11260) # 88387 -> 88295
```

Aggregate for Customer Records

重新匯整顧客資料

```
d0 = max(Xs$date) + 1
As = Xs %>% mutate(
  days = as.integer(difftime(d0, date, units="days"))
  ) %>%
  group_by(cust) %>% summarise(
    r = min(days),      # recency
    s = max(days),      # seniority
    f = n(),            # frequency
    m = mean(total),    # monetary
    rev = sum(total),   # total revenue contribution
    raw = sum(gross),   # total gross profit contribution
    age = age[1],       # age group
    area = area[1],     # area code
  ) %>% data.frame      # 28584 顧客
nrow(As)
```

```
## [1] 28584
```

---

7. 製作預測變數 Preparing the Target Variables (Y)

Aggregate Feb's Transaction by Customer

彙整最後一期(二月後)資料

```
feb = filter(X, date>= feb01) %>% group_by(cust) %>%
  summarise(amount = sum(total))
summary(feb)
```

```
##      cust               amount
##  Length:16900       Min.   :    8
##  Class :character   1st Qu.:  423
##  Mode  :character   Median :  934
##                     Mean   : 1416
##                     3rd Qu.: 1863
##                     Max.   :28089
```

`feb$amount` 之中有最後一期來買過的 `16,900` 位顧客的營收貢獻

The Target for Regression - `A$amount`

將 `feb$amount` 匯入 `A`

```
As = merge(As, feb, by="cust", all.x=T) #A和feb兩個dataframe會根據cust合併 (這是Left Join)
```

The Target for Classification - `A$buy`

`A$amount` 是 `NA` 代表這位顧客最後一期沒來買
`A$amount` 不是 `NA` 代表這位顧客最後一期有來買過

```
As$buy = !is.na(As$amount)
table(As$buy, !is.na(As$amount))
```

```
##
##         FALSE  TRUE
##   FALSE 15342     0
##   TRUE      0 13242
```

Summary of the Dataset

```
summary(As)
```

```
##      cust                r               s              f
##  Length:28584       Min.   : 1.00   Min.   : 1.00   Min.   : 1.000
##  Class :character   1st Qu.:11.00   1st Qu.:47.00   1st Qu.: 1.000
##  Mode  :character   Median :21.00   Median :68.00   Median : 2.000
##                     Mean   :32.12   Mean   :61.27   Mean   : 3.089
##                     3rd Qu.:53.00   3rd Qu.:83.00   3rd Qu.: 4.000
##                     Max.   :92.00   Max.   :92.00   Max.   :60.000
##
##        m               rev             raw              age
##  Min.   :    8.0   Min.   :    8   Min.   : -742.0   Length:28584
##  1st Qu.:  359.4   1st Qu.:  638   1st Qu.:   70.0   Class :character
##  Median :  709.5   Median : 1566   Median :  218.0   Mode  :character
##  Mean   : 1012.4   Mean   : 2711   Mean   :  420.8
##  3rd Qu.: 1315.0   3rd Qu.: 3426   3rd Qu.:  535.0
##  Max.   :10634.0   Max.   :99597   Max.   :15565.0
##
##      area               amount         buy
##  Length:28584       Min.   :    8   Mode :logical
##  Class :character   1st Qu.:  454   FALSE:15342
##  Mode  :character   Median :  993   TRUE :13242
##                     Mean   : 1499
##                     3rd Qu.: 1955
##                     Max.   :28089
##                     NA's   :15342
```

```
normal_df = As[As$cust %in% normal$cust,]
new_df = As[As$cust %in% new$cust,]
silence_df = As[As$cust %in% silence$cust,]
highp_df = As[As$cust %in% highp$cust,]
longterm_df = As[As$cust %in% longterm$cust,]
```

讓 X 與 Z 之中的資料範圍與 A 一樣

```
Xs = subset(Xs, cust %in% As$cust & date < as.Date("2001-02-01"))
Zs = subset(Zs, cust %in% As$cust & date < as.Date("2001-02-01"))
```

8. 購買機率模型 Buying Probabilities Model

Normal

```
set.seed(11)
normal_spl = sample.split(normal_df$buy, SplitRatio=0.7)
c(nrow(normal_df), sum(normal_spl), sum(!normal_spl))
```

```
## [1] 7682 5377 2305
```

```
cbind(normal_df, normal_spl) %>% filter(buy) %>%
  ggplot(aes(x=log(amount))) + geom_density(aes(fill=normal_spl), alpha=0.5)
```


```
normal_A0 = subset(normal_df, buy) %>% mutate_at(c("m","rev","amount"), log10)
n = nrow(normal_A0)
set.seed(11)
normal_spl0 = 1:n %in% sample(1:n, round(0.7*n))
c(nrow(normal_A0), sum(normal_spl0), sum(!normal_spl0))
```

```
## [1] 0 0 2
```

```
normal_train = subset(normal_df, normal_spl)
normal_test = subset(normal_df, !normal_spl)
```

```
glm_normal = glm(buy ~ . ,normal_train[,-c(1,10)],family = binomial() )
```

```
## Warning: glm.fit: algorithm did not converge
```

```
summary(glm_normal)
```

```
##
## Call:
## glm(formula = buy ~ ., family = binomial(), data = normal_train[,
##     -c(1, 10)])
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.657e+01  4.027e+04  -0.001    0.999
## r             1.129e-15  4.791e+02   0.000    1.000
## s            -3.141e-15  5.221e+02   0.000    1.000
## f            -3.530e-14  1.048e+04   0.000    1.000
## m            -1.368e-16  1.774e+01   0.000    1.000
## rev           1.412e-18  1.350e+01   0.000    1.000
## raw           2.938e-16  4.754e+01   0.000    1.000
## agea29       -7.407e-15  2.464e+04   0.000    1.000
## agea34       -1.262e-14  2.290e+04   0.000    1.000
## agea39        4.281e-15  2.287e+04   0.000    1.000
## agea44        3.543e-14  2.355e+04   0.000    1.000
## agea49        1.313e-14  2.479e+04   0.000    1.000
## agea54        3.572e-14  2.794e+04   0.000    1.000
## agea59        4.810e-14  3.426e+04   0.000    1.000
## agea64        6.805e-14  3.659e+04   0.000    1.000
## agea69        4.833e-14  3.214e+04   0.000    1.000
## agea99        3.755e-12  4.083e+04   0.000    1.000
## areaz106     -1.575e-14  3.859e+04   0.000    1.000
## areaz110     -1.601e-14  2.854e+04   0.000    1.000
## areaz114     -1.913e-14  3.112e+04   0.000    1.000
## areaz115     -3.513e-14  2.693e+04   0.000    1.000
## areaz221      8.712e-15  2.716e+04   0.000    1.000
## areazOthers   5.117e-13  2.860e+04   0.000    1.000
## areazUnknown -8.877e-13  3.435e+04   0.000    1.000
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 0.0000e+00  on 5376  degrees of freedom
## Residual deviance: 3.1195e-08  on 5353  degrees of freedom
## AIC: 48
##
## Number of Fisher Scoring iterations: 25
```

```
normal_pred = predict(glm_normal, normal_test, type = "response")
normal_confmt = table(actual = normal_test$buy, predict = normal_pred > 0.5)
normal_confmt
```

```
##        predict
## actual  FALSE
##   FALSE  2305
```

```
normal_acc.ts = normal_confmt %>% {sum(diag(.))/sum(.)}
c(1-mean(normal_test$buy) , normal_acc.ts)
```

```
## [1] 1 1
```

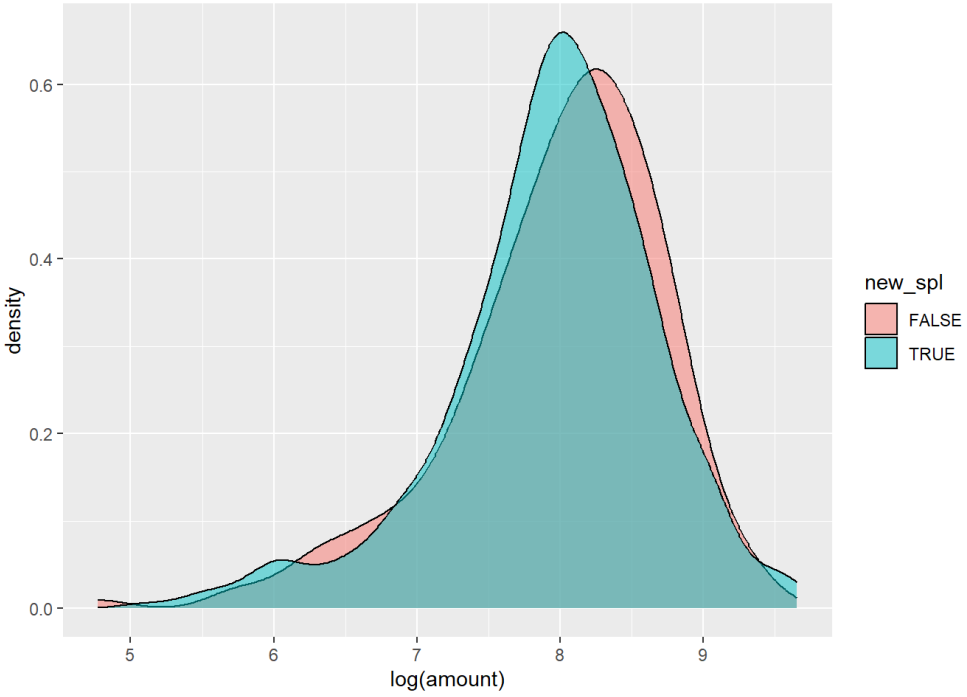
```
# colAUC(normal_pred, normal_test$buy)
```

New

```
set.seed(11)
new_spl = sample.split(new_df$buy, SplitRatio=0.7)
c(nrow(new_df), sum(new_spl), sum(!new_spl))
```

```
## [1] 2495 1746  749
```

```
cbind(new_df, new_spl) %>% filter(buy) %>%
  ggplot(aes(x=log(amount))) + geom_density(aes(fill=new_spl), alpha=0.5)
```




```
new_A0 = subset(new_df, buy) %>% mutate_at(c("m","rev","amount"), log10)
n = nrow(new_A0)
set.seed(11)
new_spl0 = 1:n %in% sample(1:n, round(0.7*n))
c(nrow(new_A0), sum(new_spl0), sum(!new_spl0))
```

```
## [1] 680 476 204
```

```
new_train = subset(new_df, new_spl)
new_test = subset(new_df, !new_spl)
```

```
glm_new = glm(buy ~ . ,new_train[,-c(1,10)],family = binomial() )
summary(glm_new)
```

```
##
## Call:
## glm(formula = buy ~ ., family = binomial(), data = new_train[,
##     -c(1, 10)])
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.748e+00  6.400e-01  -2.732 0.006297 **
## r             1.115e-03  3.518e-03   0.317 0.751210
## s             6.836e-03  3.479e-03   1.965 0.049465 *
## f            -4.012e-01  1.717e-01  -2.336 0.019466 *
## m            -3.398e-04  9.471e-05  -3.588 0.000333 ***
## rev           1.901e-04  5.957e-05   3.191 0.001419 **
## raw           2.527e-04  1.602e-04   1.577 0.114687
## agea29        7.547e-01  5.358e-01   1.409 0.158968
## agea34        1.162e+00  5.080e-01   2.287 0.022219 *
## agea39        1.206e+00  5.065e-01   2.381 0.017252 *
## agea44        1.143e+00  5.104e-01   2.239 0.025176 *
## agea49        8.761e-01  5.231e-01   1.675 0.093926 .
## agea54        2.879e-01  5.670e-01   0.508 0.611593
## agea59        1.386e+00  5.821e-01   2.381 0.017260 *
## agea64        1.053e+00  6.368e-01   1.653 0.098385 .
## agea69        7.184e-01  6.781e-01   1.059 0.289428
## agea99        8.242e-01  6.718e-01   1.227 0.219863
## areaz106      5.369e-02  3.356e-01   0.160 0.872903
## areaz110     -4.430e-01  2.883e-01  -1.536 0.124434
## areaz114     -8.100e-02  3.174e-01  -0.255 0.798551
## areaz115     -3.993e-02  2.830e-01  -0.141 0.887803
## areaz221     -2.368e-01  2.787e-01  -0.850 0.395518
## areazOthers  -1.541e-01  2.901e-01  -0.531 0.595166
## areazUnknown -7.794e-01  4.279e-01  -1.822 0.068511 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2045.8  on 1745  degrees of freedom
## Residual deviance: 1904.4  on 1722  degrees of freedom
## AIC: 1952.4
##
## Number of Fisher Scoring iterations: 4
```

```
new_pred = predict(glm_new, new_test, type = "response")
new_confmt = table(actual = new_test$buy, predict =new_pred > 0.5)
new_confmt
```

```
##        predict
## actual  FALSE TRUE
##   FALSE   530   15
##   TRUE    177   27
```

```
new_acc.ts = new_confmt %>% {sum(diag(.))/sum(.)}
c(1-mean(new_test$buy) , new_acc.ts) #accuracy approximately 0.73
```

```
## [1] 0.7276368 0.7436582
```

```
colAUC(new_pred, new_test$buy)
```

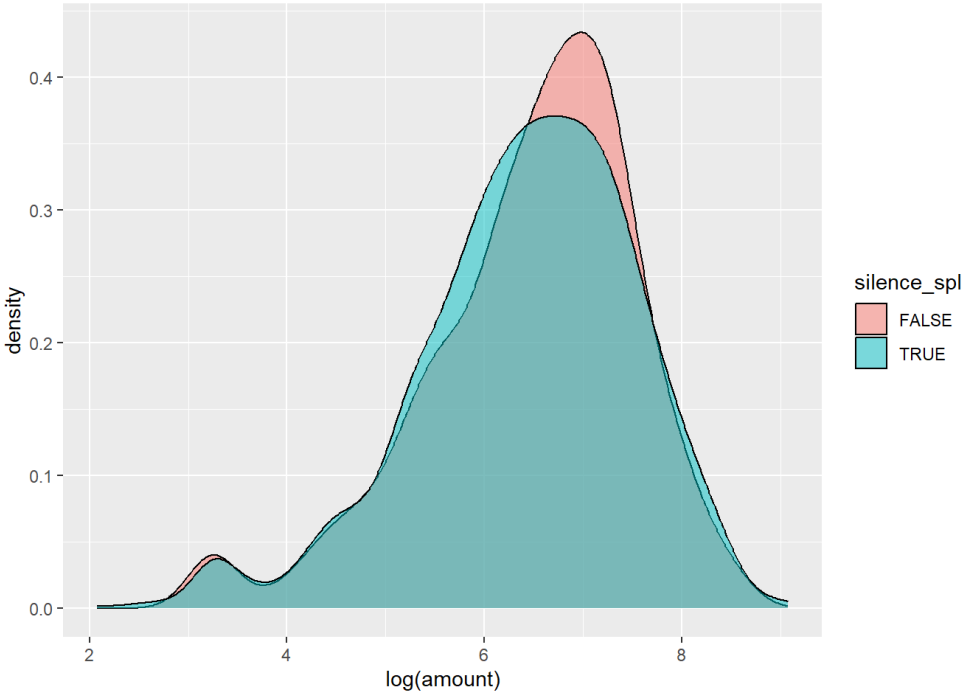
```
##                     [,1]
## FALSE vs. TRUE 0.6423907
```

Silence

```
set.seed(11)
silence_spl = sample.split(silence_df$buy, SplitRatio=0.7)
c(nrow(silence_df), sum(silence_spl), sum(!silence_spl))
```

```
## [1] 4857 3400 1457
```

```
cbind(silence_df, silence_spl) %>% filter(buy) %>%
  ggplot(aes(x=log(amount))) + geom_density(aes(fill=silence_spl), alpha=0.5)
```




```
silence_A0 = subset(silence_df, buy) %>% mutate_at(c("m","rev","amount"), log10)
n = nrow(silence_A0)
set.seed(11)
silence_spl0 = 1:n %in% sample(1:n, round(0.7*n))
c(nrow(silence_A0), sum(silence_spl0), sum(!silence_spl0))
```

```
## [1] 1727 1209  518
```

```
silence_train = subset(silence_df, silence_spl)
silence_test = subset(silence_df, !silence_spl)
```

```
glm_silence = glm(buy ~ . ,silence_train[,-c(1,10)],family = binomial() )
summary(glm_silence)
```

```
##
## Call:
## glm(formula = buy ~ ., family = binomial(), data = silence_train[,
##     -c(1, 10)])
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.287e+00  3.283e-01  -6.966 3.26e-12 ***
## r             2.208e-02  8.459e-03   2.611 0.009037 **
## s             1.331e-02  7.836e-03   1.699 0.089329 .
## f             4.082e-01  1.213e-01   3.366 0.000762 ***
## m             3.053e-04  1.456e-04   2.097 0.035967 *
## rev           6.849e-05  1.279e-04   0.536 0.592262
## raw          -3.482e-04  3.518e-04  -0.990 0.322315
## agea29       -1.311e-01  2.138e-01  -0.613 0.539625
## agea34       -6.265e-02  1.937e-01  -0.324 0.746301
## agea39       -1.140e-01  1.944e-01  -0.587 0.557534
## agea44       -6.597e-02  1.992e-01  -0.331 0.740436
## agea49       -1.330e-01  2.064e-01  -0.645 0.519138
## agea54        4.388e-03  2.226e-01   0.020 0.984272
## agea59        2.721e-01  2.648e-01   1.028 0.304182
## agea64        4.860e-01  2.593e-01   1.875 0.060852 .
## agea69        8.554e-03  2.411e-01   0.035 0.971694
## agea99       -7.085e-01  3.734e-01  -1.898 0.057746 .
## areaz106     -1.091e-02  3.206e-01  -0.034 0.972852
## areaz110     -5.551e-04  2.432e-01  -0.002 0.998179
## areaz114      3.683e-01  2.521e-01   1.461 0.144091
## areaz115      5.953e-01  2.274e-01   2.618 0.008843 **
## areaz221      3.780e-01  2.310e-01   1.636 0.101855
## areazOthers   2.052e-01  2.477e-01   0.828 0.407427
## areazUnknown  3.738e-01  2.772e-01   1.348 0.177559
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4425.7  on 3399  degrees of freedom
## Residual deviance: 4236.3  on 3376  degrees of freedom
## AIC: 4284.3
##
## Number of Fisher Scoring iterations: 4
```

```
silence_pred = predict(glm_silence, silence_test, type = "response")
silence_confmt = table(actual = silence_test$buy, predict = silence_pred > 0.5)
silence_confmt
```

```
##        predict
## actual  FALSE TRUE
##   FALSE   874   65
##   TRUE    421   97
```

```
silence_acc.ts = silence_confmt %>% {sum(diag(.))/sum(.)}
c(1-mean(silence_test$buy) , silence_acc.ts)
```

```
## [1] 0.6444749 0.6664379
```

```
colAUC(silence_pred, silence_test$buy)
```

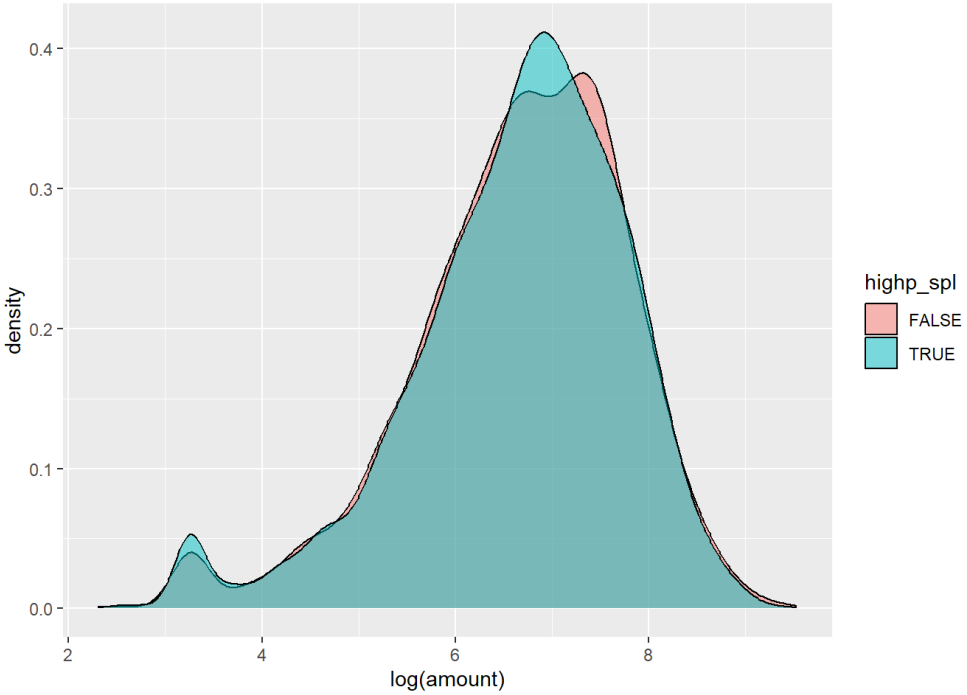
```
##                   [,1]
## FALSE vs. TRUE 0.609206
```

Highp

```
set.seed(11)
highp_spl = sample.split(highp_df$buy, SplitRatio=0.7)
c(nrow(highp_df), sum(highp_spl), sum(!highp_spl))
```

```
## [1] 12375  8663  3712
```

```
cbind(highp_df, highp_spl) %>% filter(buy) %>%
  ggplot(aes(x=log(amount))) + geom_density(aes(fill=highp_spl), alpha=0.5)
```




```
highp_A0 = subset(highp_df, buy) %>% mutate_at(c("m","rev","amount"), log10)
n = nrow(highp_A0)
set.seed(11)
highp_spl0 = 1:n %in% sample(1:n, round(0.7*n))
c(nrow(highp_A0), sum(highp_spl0), sum(!highp_spl0))
```

```
## [1] 9685 6780 2905
```

```
highp_train = subset(highp_df, highp_spl)
highp_test = subset(highp_df, !highp_spl)
```

```
glm_highp = glm(buy ~ . ,highp_train[,-c(1,10)],family = binomial() )
summary(glm_highp)
```

```
##
## Call:
## glm(formula = buy ~ ., family = binomial(), data = highp_train[,
##     -c(1, 10)])
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.230e+00  2.955e-01  -4.163 3.14e-05 ***
## r             7.860e-02  2.795e-03  28.120  < 2e-16 ***
## s            -5.315e-03  2.257e-03  -2.355   0.0185 *
## f             2.734e-01  2.875e-02   9.511  < 2e-16 ***
## m            -3.351e-05  1.169e-04  -0.287   0.7744
## rev           8.611e-05  3.903e-05   2.206   0.0274 *
## raw          -2.209e-04  1.379e-04  -1.602   0.1091
## agea29       -1.226e-01  1.642e-01  -0.746   0.4556
## agea34        7.403e-03  1.540e-01   0.048   0.9617
## agea39        8.534e-02  1.526e-01   0.559   0.5760
## agea44       -2.732e-02  1.557e-01  -0.175   0.8607
## agea49       -5.442e-02  1.606e-01  -0.339   0.7347
## agea54        7.891e-03  1.727e-01   0.046   0.9636
## agea59        1.549e-01  2.079e-01   0.745   0.4563
## agea64        1.273e-01  2.157e-01   0.590   0.5549
## agea69        1.525e-01  1.879e-01   0.812   0.4169
## agea99       -3.263e-01  2.786e-01  -1.171   0.2416
## areaz106      2.752e-01  2.823e-01   0.975   0.3296
## areaz110      4.986e-03  2.144e-01   0.023   0.9814
## areaz114      1.875e-01  2.327e-01   0.806   0.4204
## areaz115      4.524e-01  1.978e-01   2.287   0.0222 *
## areaz221      3.881e-01  1.990e-01   1.950   0.0512 .
## areazOthers   2.896e-01  2.158e-01   1.342   0.1797
## areazUnknown  5.377e-02  2.390e-01   0.225   0.8220
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9071  on 8662  degrees of freedom
## Residual deviance: 7505  on 8639  degrees of freedom
## AIC: 7553
##
## Number of Fisher Scoring iterations: 6
```

```
highp_pred = predict(glm_highp, highp_test, type = "response")
highp_confmt = table(actual = highp_test$buy, predict = highp_pred > 0.5)
highp_confmt
```

```
##        predict
## actual  FALSE TRUE
##   FALSE   128  679
##   TRUE    118 2787
```

```
highp_acc.ts = highp_confmt %>% {sum(diag(.))/sum(.)}
c(1-mean(highp_test$buy) , highp_acc.ts)
```

```
## [1] 0.2174030 0.7852909
```

```
colAUC(highp_pred, highp_test$buy)
```

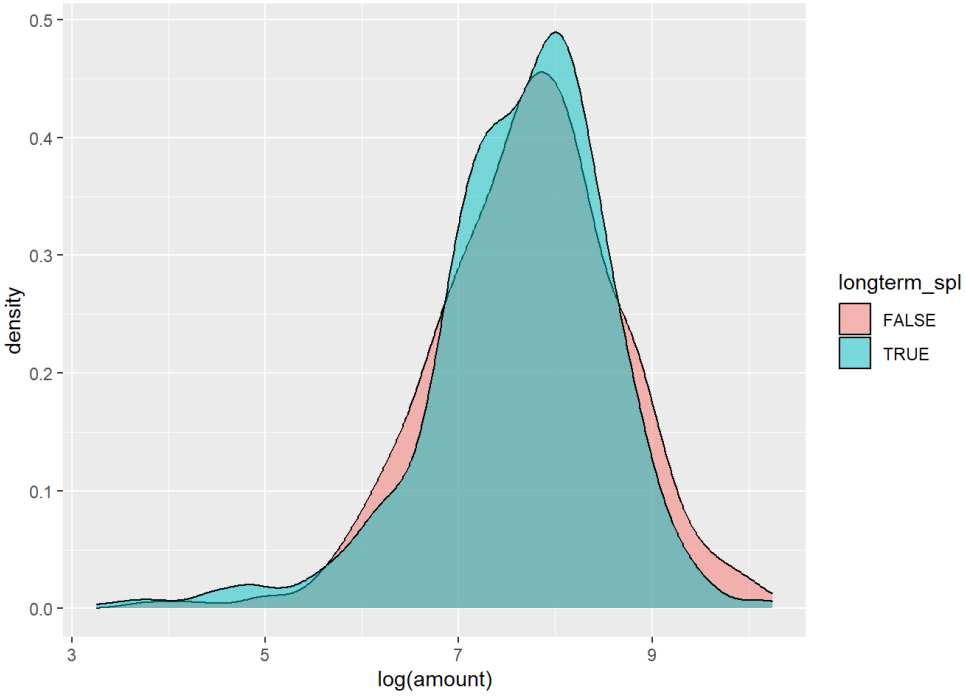
```
##                     [,1]
## FALSE vs. TRUE 0.7766049
```

Longterm

```
set.seed(11)
longterm_spl = sample.split(longterm_df$buy, SplitRatio=0.7)
c(nrow(longterm_df), sum(longterm_spl), sum(!longterm_spl))
```

```
## [1] 1172  820  352
```

```
cbind(longterm_df, longterm_spl) %>% filter(buy) %>%
  ggplot(aes(x=log(amount))) + geom_density(aes(fill=longterm_spl), alpha=0.5)
```




```
longterm_A0 = subset(longterm_df, buy) %>% mutate_at(c("m","rev","amount"), log10)
n = nrow(longterm_A0)
set.seed(11)
longterm_spl0 = 1:n %in% sample(1:n, round(0.7*n))
c(nrow(longterm_A0), sum(longterm_spl0), sum(!longterm_spl0))
```

```
## [1] 1150  805  345
```

```
longterm_train = subset(longterm_df, longterm_spl)
longterm_test = subset(longterm_df, !longterm_spl)
```

```
glm_longterm = glm(buy ~ . ,longterm_train[,-c(1,10)],family = binomial() )
summary(glm_longterm)
```

```
##
## Call:
## glm(formula = buy ~ ., family = binomial(), data = longterm_train[,
##     -c(1, 10)])
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.907e+01  7.191e+03   0.005    0.996
## r            -1.576e-01  3.022e-02  -5.215 1.84e-07 ***
## s            -4.025e-02  7.523e-02  -0.535    0.593
## f            -3.564e-02  1.019e-01  -0.350    0.727
## m            -2.375e-04  2.928e-03  -0.081    0.935
## rev          -1.159e-04  1.958e-04  -0.592    0.554
## raw           1.315e-03  9.573e-04   1.374    0.169
## agea29       -1.450e+01  1.783e+03  -0.008    0.994
## agea34       -1.433e+01  1.783e+03  -0.008    0.994
## agea39       -1.451e+01  1.783e+03  -0.008    0.994
## agea44       -1.437e+01  1.783e+03  -0.008    0.994
## agea49       -1.492e+01  1.783e+03  -0.008    0.993
## agea54       -1.498e+01  1.783e+03  -0.008    0.993
## agea59       -1.622e+01  1.783e+03  -0.009    0.993
## agea64       -1.718e+01  1.783e+03  -0.010    0.992
## agea69       -1.595e+01  1.783e+03  -0.009    0.993
## agea99       -1.690e+01  1.783e+03  -0.009    0.992
## areaz106     -7.490e-01  9.209e+03   0.000    1.000
## areaz110      1.032e+00  7.526e+03   0.000    1.000
## areaz114     -1.516e+01  6.966e+03  -0.002    0.998
## areaz115     -1.434e+01  6.966e+03  -0.002    0.998
## areaz221     -1.439e+01  6.966e+03  -0.002    0.998
## areazOthers  -1.567e+01  6.966e+03  -0.002    0.998
## areazUnknown -1.360e+01  6.966e+03  -0.002    0.998
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 149.762  on 819  degrees of freedom
## Residual deviance:  96.091  on 796  degrees of freedom
## AIC: 144.09
##
## Number of Fisher Scoring iterations: 18
```

```
longterm_pred = predict(glm_longterm, longterm_test, type = "response")
longterm_confmt = table(actual = longterm_test$buy, predict = longterm_pred > 0.5)
longterm_confmt
```

```
##        predict
## actual  FALSE TRUE
##   FALSE     0    7
##   TRUE      2  343
```

```
longterm_acc.ts = longterm_confmt %>% {sum(diag(.))/sum(.)}
c(1-mean(longterm_test$buy) , longterm_acc.ts)
```

```
## [1] 0.01988636 0.97443182
```

```
colAUC(longterm_pred, longterm_test$buy)
```

```
##                     [,1]
## FALSE vs. TRUE 0.6041408
```

9. 購買金額模型 Buying Amount Model

Normal

```
# normal_train1 = subset(normal_A0, normal_spl0)
# normal_test1 = subset(normal_A0, !normal_spl0)
```

```
# normal_lm = lm(amount ~ ., normal_train1[,2:10])
# summary(normal_lm)
```

```
# r2.tr_normal = summary(normal_lm)$r.sq
# SST_normal = sum((normal_test1$amount - mean(normail_train1$amount))^ 2)
# SSE_normal = sum((predict(normal_lm, normal_test1) -  normal_test1$amount)^2)
# r2.ts_normal = 1 - (SSE_normal/SST_normal)
# c(R0train_normal=r2.tr_normal, R0test_normal=r2.ts_normal)
```

New

```
new_train1 = subset(new_A0, new_spl0)
new_test1 = subset(new_A0, !new_spl0)
```

```
new_lm = lm(amount ~ ., new_train1[,2:10])
summary(new_lm)
```

```
##
## Call:
## lm(formula = amount ~ ., data = new_train1[, 2:10])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.15817 -0.16563  0.00933  0.18684  0.75198
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.056e+00  3.518e-01  14.372  < 2e-16 ***
## r             7.179e-04  1.101e-03   0.652  0.51458
## s             3.410e-04  1.097e-03   0.311  0.75612
## f             2.538e-03  3.417e-02   0.074  0.94081
## m            -4.322e-01  2.403e-01  -1.799  0.07274 .
## rev          -7.597e-02  2.186e-01  -0.348  0.72830
## raw           7.852e-05  2.962e-05   2.651  0.00831 **
## agea29       -8.699e-02  1.244e-01  -0.699  0.48482
## agea34       -1.568e-02  1.149e-01  -0.136  0.89153
## agea39        6.754e-02  1.142e-01   0.591  0.55458
## agea44        7.674e-02  1.164e-01   0.659  0.51016
## agea49        3.603e-02  1.185e-01   0.304  0.76120
## agea54       -8.172e-04  1.328e-01  -0.006  0.99509
## agea59        6.736e-02  1.391e-01   0.484  0.62847
## agea64       -1.023e-01  1.464e-01  -0.698  0.48524
## agea69        1.446e-01  1.494e-01   0.968  0.33375
## agea99        2.113e-01  1.787e-01   1.182  0.23785
## areaz106      5.923e-03  7.814e-02   0.076  0.93961
## areaz110      9.694e-02  6.687e-02   1.450  0.14785
## areaz114      1.403e-02  7.329e-02   0.191  0.84823
## areaz115     -2.916e-02  6.418e-02  -0.454  0.64980
## areaz221      3.851e-02  6.388e-02   0.603  0.54695
## areazOthers  -1.113e-02  6.877e-02  -0.162  0.87154
## areazUnknown -2.976e-01  1.207e-01  -2.466  0.01405 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2922 on 452 degrees of freedom
## Multiple R-squared:  0.1453, Adjusted R-squared:  0.1018
## F-statistic:  3.34 on 23 and 452 DF,  p-value: 4.858e-07
```

```
r2.tr_new = summary(new_lm)$r.sq
SST_new = sum((new_test1$amount - mean(new_train1$amount))^ 2)
SSE_new = sum((predict(new_lm, new_test1) -  new_test1$amount)^2)
r2.ts_new = 1 - (SSE_new/SST_new)
c(R0train_new=r2.tr_new, R0test_new=r2.ts_new)
```

```
## R0train_new  R0test_new
##  0.14525903  0.09027095
```

Silence

```
silence_train1 = subset(silence_A0, silence_spl0)
silence_test1 = subset(silence_A0, !silence_spl0)
```

```
silence_lm = lm(amount ~ ., silence_train1[,2:10])
summary(silence_lm)
```

```
##
## Call:
## lm(formula = amount ~ ., data = silence_train1[, 2:10])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.80858 -0.25881  0.05561  0.29774  1.15919
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.817e+00  1.748e-01  10.394  < 2e-16 ***
## r             9.549e-03  2.996e-03   3.187  0.00147 **
## s            -9.278e-03  2.829e-03  -3.280  0.00107 **
## f            -7.367e-02  8.971e-02  -0.821  0.41169
## m            -4.007e-01  3.989e-01  -1.004  0.31536
## rev           7.721e-01  3.965e-01   1.947  0.05173 .
## raw          -7.733e-05  7.934e-05  -0.975  0.32992
## agea29        5.173e-02  7.294e-02   0.709  0.47835
## agea34        1.042e-01  6.676e-02   1.561  0.11883
## agea39        5.518e-02  6.743e-02   0.818  0.41331
## agea44        2.523e-02  6.955e-02   0.363  0.71684
## agea49        4.341e-02  7.268e-02   0.597  0.55042
## agea54        4.289e-02  7.603e-02   0.564  0.57284
## agea59        4.794e-02  8.486e-02   0.565  0.57227
## agea64       -4.905e-03  8.872e-02  -0.055  0.95592
## agea69       -1.173e-01  8.514e-02  -1.378  0.16853
## agea99        1.056e-01  1.181e-01   0.894  0.37169
## areaz106      1.035e-02  1.141e-01   0.091  0.92771
## areaz110     -1.044e-02  8.770e-02  -0.119  0.90525
## areaz114     -3.549e-02  9.090e-02  -0.390  0.69631
## areaz115     -6.254e-02  8.021e-02  -0.780  0.43569
## areaz221      1.546e-02  8.163e-02   0.189  0.84978
## areazOthers  -8.789e-02  8.898e-02  -0.988  0.32351
## areazUnknown -1.311e-01  9.980e-02  -1.314  0.18926
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4414 on 1185 degrees of freedom
## Multiple R-squared:  0.1369, Adjusted R-squared:  0.1201
## F-statistic: 8.172 on 23 and 1185 DF,  p-value: < 2.2e-16
```

```
r2.tr_silence = summary(silence_lm)$r.sq
SST_silence = sum((silence_test1$amount - mean(silence_train1$amount))^ 2)
SSE_silence = sum((predict(silence_lm, silence_test1) -  silence_test1$amount)^2)
r2.ts_silence = 1 - (SSE_silence/SST_silence)
c(R0train_silence=r2.tr_silence, R0test_silence=r2.ts_silence)
```

```
## R0train_silence  R0test_silence
##      0.13689435      0.06346612
```

Highp

```
highp_train1 = subset(highp_A0, highp_spl0)
highp_test1 = subset(highp_A0, !highp_spl0)
```

```
highp_lm = lm(amount ~ ., highp_train1[,2:10])
summary(highp_lm)
```

```
##
## Call:
## lm(formula = amount ~ ., data = highp_train1[, 2:10])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.02529 -0.23099  0.04425  0.28488  1.52532
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.468e+00  7.018e-02  20.915  < 2e-16 ***
## r             2.385e-04  3.434e-04   0.695 0.487288
## s            -8.351e-04  4.512e-04  -1.851 0.064251 .
## f             1.252e-02  7.223e-03   1.733 0.083069 .
## m             4.155e-01  7.052e-02   5.891 4.02e-09 ***
## rev           3.632e-02  6.768e-02   0.537 0.591483
## raw           1.028e-04  1.467e-05   7.008 2.66e-12 ***
## agea29        3.231e-02  2.858e-02   1.130 0.258307
## agea34        9.627e-02  2.657e-02   3.623 0.000293 ***
## agea39        9.836e-02  2.626e-02   3.745 0.000182 ***
## agea44        8.614e-02  2.684e-02   3.210 0.001335 **
## agea49        7.268e-02  2.790e-02   2.605 0.009216 **
## agea54        7.462e-02  3.072e-02   2.429 0.015171 *
## agea59        2.417e-02  3.584e-02   0.675 0.500000
## agea64        2.937e-02  3.761e-02   0.781 0.434863
## agea69       -6.476e-02  3.228e-02  -2.006 0.044894 *
## agea99        4.869e-02  5.161e-02   0.943 0.345519
## areaz106      8.440e-02  4.990e-02   1.692 0.090785 .
## areaz110      5.911e-02  4.062e-02   1.455 0.145687
## areaz114      2.875e-02  4.312e-02   0.667 0.504945
## areaz115      3.351e-02  3.712e-02   0.903 0.366725
## areaz221      6.403e-02  3.743e-02   1.711 0.087187 .
## areazOthers   4.502e-02  4.019e-02   1.120 0.262661
## areazUnknown  3.123e-02  4.555e-02   0.686 0.493017
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4215 on 6756 degrees of freedom
## Multiple R-squared:  0.2323, Adjusted R-squared:  0.2297
## F-statistic: 88.87 on 23 and 6756 DF,  p-value: < 2.2e-16
```

```
r2.tr_highp = summary(highp_lm)$r.sq
SST_highp = sum((highp_test1$amount - mean(highp_train1$amount))^ 2)
SSE_highp = sum((predict(highp_lm, highp_test1) - highp_test1$amount)^2)
r2.ts_highp = 1 - (SSE_highp/SST_highp)
c(R0train_highp=r2.tr_highp, R0test_highp=r2.ts_highp)
```

```
## R0train_highp  R0test_highp
##     0.2322689     0.2169058
```

Longterm

```
longterm_train1 = subset(longterm_A0, longterm_spl0)
longterm_test1 = subset(longterm_A0, !longterm_spl0)
```

```
longterm_lm = lm(amount ~ ., longterm_train1[,2:10])
summary(longterm_lm)
```

```
##
## Call:
## lm(formula = amount ~ ., data = longterm_train1[, 2:10])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.78518 -0.15986  0.05144  0.20184  0.73818
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.497e+00  3.815e-01   6.546 1.07e-10 ***
## r            -6.673e-03  1.968e-03  -3.391 0.000731 ***
## s             6.299e-04  2.090e-03   0.301 0.763179
## f             3.564e-02  5.893e-03   6.047 2.28e-09 ***
## m             1.797e+00  2.777e-01   6.469 1.74e-10 ***
## rev          -1.161e+00  2.714e-01  -4.278 2.12e-05 ***
## raw           1.703e-05  1.338e-05   1.273 0.203539
## agea29        8.143e-03  7.257e-02   0.112 0.910683
## agea34        7.437e-02  6.651e-02   1.118 0.263867
## agea39        1.242e-01  6.420e-02   1.935 0.053391 .
## agea44        6.795e-02  6.384e-02   1.064 0.287502
## agea49        8.322e-02  6.713e-02   1.240 0.215471
## agea54        7.213e-03  7.236e-02   0.100 0.920631
## agea59        4.467e-02  9.076e-02   0.492 0.622695
## agea64        1.338e-02  9.630e-02   0.139 0.889530
## agea69       -7.950e-03  7.723e-02  -0.103 0.918042
## agea99        9.405e-02  9.281e-02   1.013 0.311182
## areaz106     -5.378e-02  2.375e-01  -0.226 0.820903
## areaz110     -1.203e-01  2.141e-01  -0.562 0.574235
## areaz114     -2.278e-01  2.255e-01  -1.010 0.312766
## areaz115     -2.280e-01  1.950e-01  -1.169 0.242701
## areaz221     -2.508e-01  1.958e-01  -1.281 0.200579
## areazOthers  -1.815e-01  2.021e-01  -0.898 0.369483
## areazUnknown -2.501e-01  2.036e-01  -1.228 0.219678
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3322 on 781 degrees of freedom
## Multiple R-squared:  0.3559, Adjusted R-squared:  0.337
## F-statistic: 18.77 on 23 and 781 DF,  p-value: < 2.2e-16
```

```
r2.tr_longterm = summary(longterm_lm)$r.sq
SST_longterm = sum((longterm_test1$amount - mean(longterm_train1$amount))^ 2)
SSE_longterm = sum((predict(longterm_lm, longterm_test1) - longterm_test1$amount)^2)
r2.ts_longterm = 1 - (SSE_longterm/SST_longterm)
c(R0train_longterm=r2.tr_longterm, R0test_longterm=r2.ts_longterm)
```

```
## R0train_longterm  R0test_longterm
##        0.3559308        0.3717247
```

10. 更多的預測變數 More Predictors for Better Prediction

實務上我們可以製作更多的預測變數來提高模型的預測能力

匯整 2000-12-01 ~ 2001~02-28 這三個月的資料來做預測變數

```
load("data/tf0.rdata")
d0 = max(X$date) + 1
B = X0 %>%
  filter(date >= as.Date("2000-12-01")) %>%
  mutate(days = as.integer(difftime(d0, date, units="days"))) %>%
  group_by(cust) %>% summarise(
    r = min(days),      # recency
    s = max(days),      # seniority
    f = n(),            # frquency
    m = mean(total),    # monetary
    rev = sum(total),   # total revenue contribution
    raw = sum(gross),   # total gross profit contribution
    age = age[1],       # age group
    area = area[1],     # area code
  ) %>% data.frame      # 28531
nrow(B)
```

```
## [1] 28531
```

```
normal_B = B[B$cust %in% normal$cust,]
new_B = B[B$cust %in% new$cust,]
silence_B = B[B$cust %in% silence$cust,]
highp_B = B[B$cust %in% highp$cust,]
longterm_B = B[B$cust %in% longterm$cust,]
```

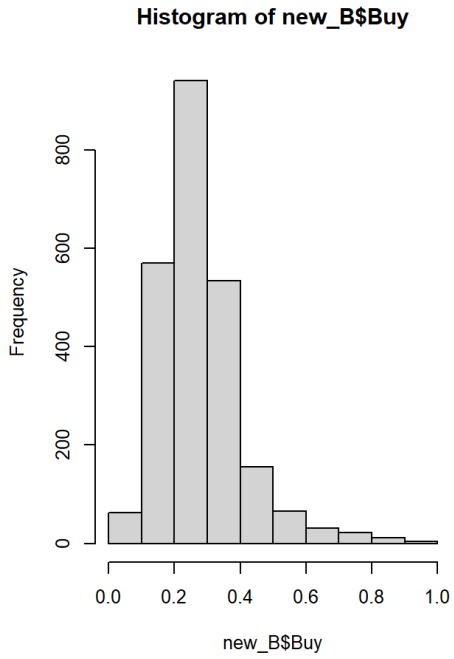
```
#B$Buy = predict(glm1, B, type="response")
normal_B$Buy = predict(glm_normal, normal_B, type="response")
new_B$Buy = predict(glm_new, new_B, type="response")
silence_B$Buy = predict(glm_silence, silence_B, type="response")
highp_B$Buy = predict(glm_highp, highp_B, type="response")
longterm_B$Buy = predict(glm_longterm, longterm_B, type="response")
```

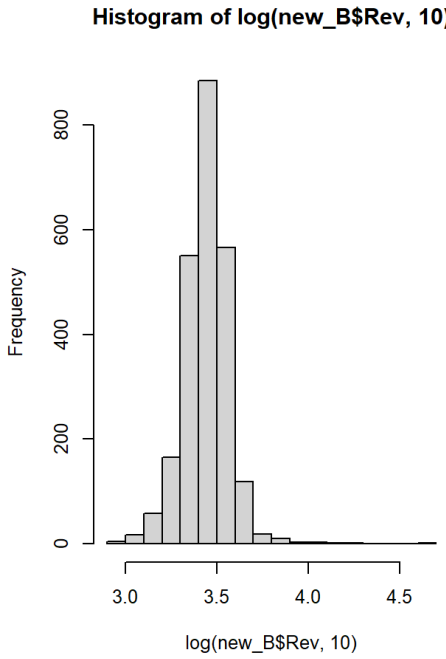
Normal

```
# normal_Bs = normal_B %>% mutate_at(c("m","rev"), log10)
# normal_B$Rev = 10^predict(normal_lm, normal_Bs)
# par(mfrow=c(1,2), cex=0.8)
# hist(normal_B$Buy)
# hist(log(normal_B$Rev,10))
```

New

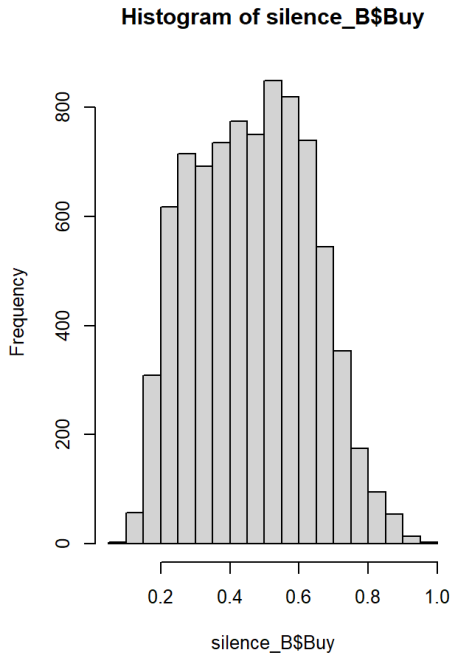
```
new_Bs = new_B %>% mutate_at(c("m","rev"), log10)
new_B$Rev = 10^predict(new_lm, new_Bs)
par(mfrow=c(1,2), cex=0.8)
hist(new_B$Buy)
hist(log(new_B$Rev,10))
```

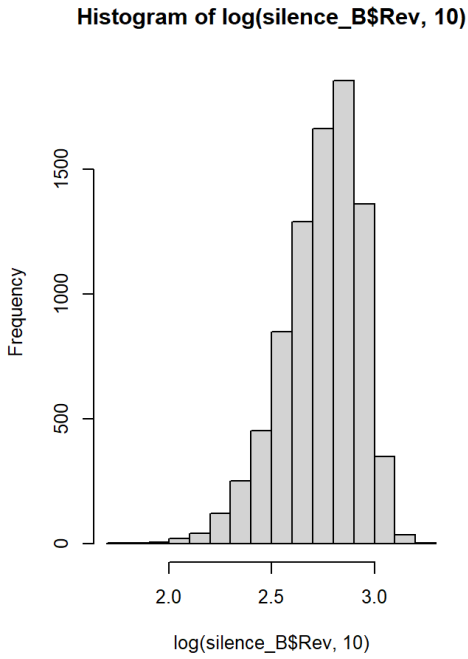




Silence

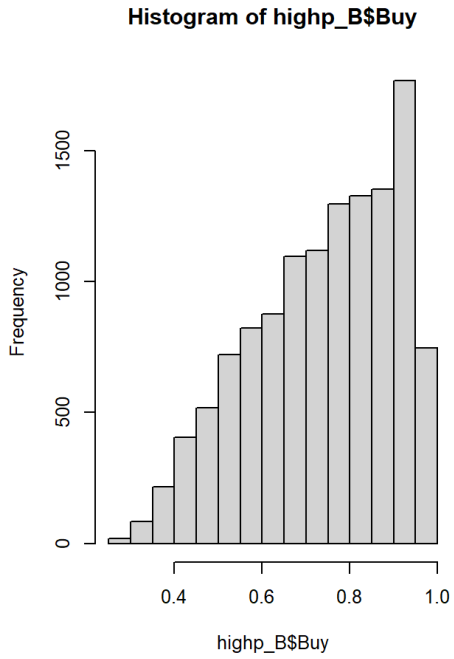
```
silence_Bs = silence_B %>% mutate_at(c("m","rev"), log10)
silence_B$Rev = 10^predict(silence_lm, silence_Bs)
par(mfrow=c(1,2), cex=0.8)
hist(silence_B$Buy)
hist(log(silence_B$Rev,10))
```

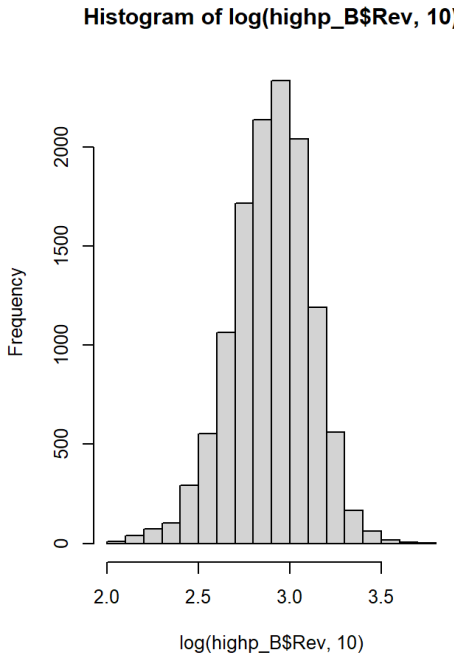




Highp

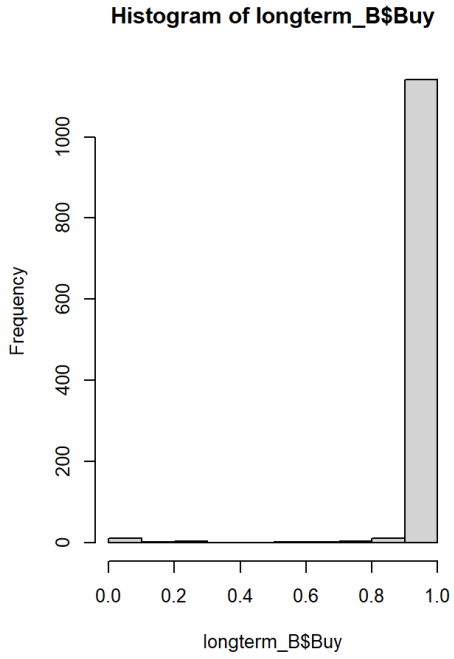
```
highp_Bs = highp_B %>% mutate_at(c("m","rev"), log10)
highp_B$Rev = 10^predict(highp_lm, highp_Bs)
par(mfrow=c(1,2), cex=0.8)
hist(highp_B$Buy)
hist(log(highp_B$Rev,10))
```

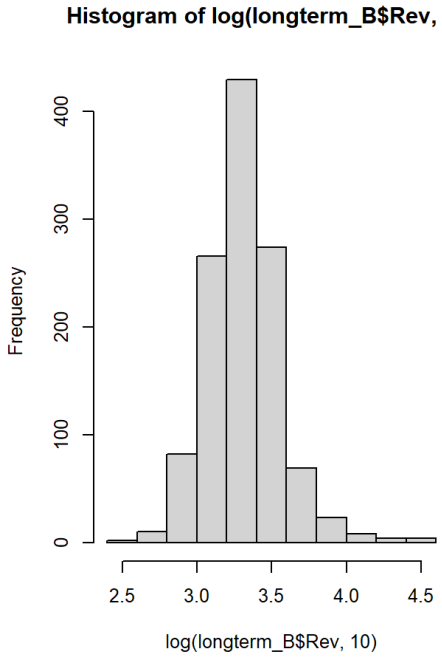




Longterm

```
longterm_Bs = longterm_B %>% mutate_at(c("m","rev"), log10)
longterm_B$Rev = 10^predict(longterm_lm, longterm_Bs)
par(mfrow=c(1,2), cex=0.8)
hist(longterm_B$Buy)
hist(log(longterm_B$Rev,10))
```





11. 顧客終生價值(CLV - Customer Live Time Value)

接著我們透過計算顧客終生價值讓我們了解每一個顧客的潛在價值有多大。 Given one's retention probability and expected buying amount, CLV can be estimated via discounted cash flow.

顧客i的終生價值 (customer i's CLV)

$$V_i = \sum_{t=0}^{N} g \times m_i \frac{r_i^t}{(1+d)^t} = g \times m_i \sum_{t=0}^{N} (\frac{r_i}{1+d})^t$$

m_i、r_i：顧客i的預期(每期)營收貢獻、保留機率
g、d：公司的(稅前)營業利潤利率、資金成本

m_i : i's expected buying amount
r_i : i's expected retention probability
g、d : company's operational margin rate
g、d : company's cost of capital

Basic Assumption

```
g = 0.3   # 平均毛利率
N = 36    # 估計CLV期間(三年)
d = 0.01  # 資本利率
```

Normal

```
# normal_B$CLV = g * normal_B$Rev * rowSums(sapply(
#   0:N, function(i) (normal_B$Buy/(1+d))^i ) )
# summary(normal_B$CLV)
```

```
# ggplot(normal_B, aes(CLV)) +
#   geom_histogram(bins=30, fill="green",alpha=0.6) +
#   scale_x_log10()
```

```
# ggplot(normal_B, aes(CLV,color= age)) +
#   geom_density(alpha=0.6) +
#   scale_x_log10()
```

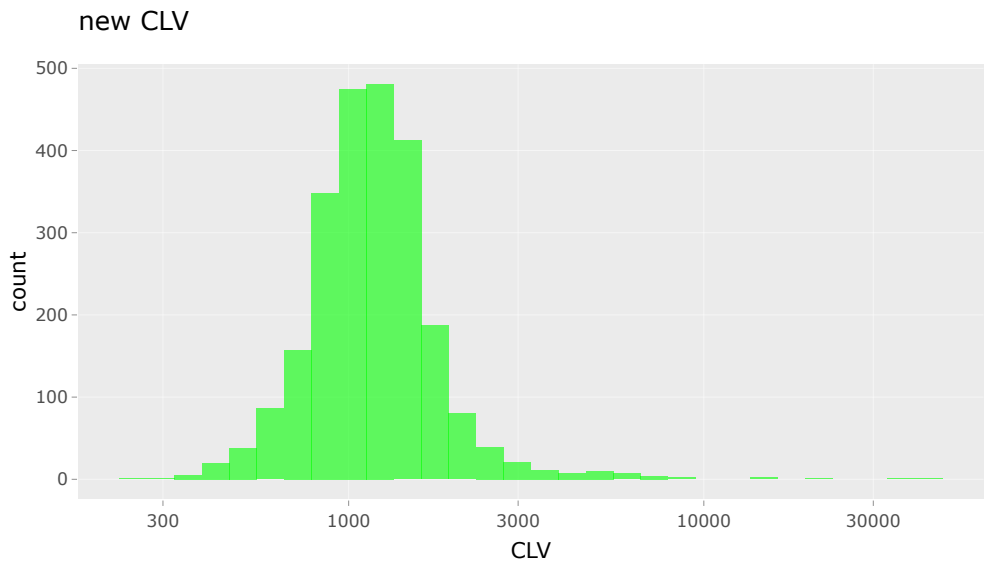
New

There're 2398 cust.

```
new_B$CLV = g * new_B$Rev * rowSums(sapply(
  0:N, function(i) (new_B$Buy/(1+d))^i ) )
summary(new_B$CLV)
```

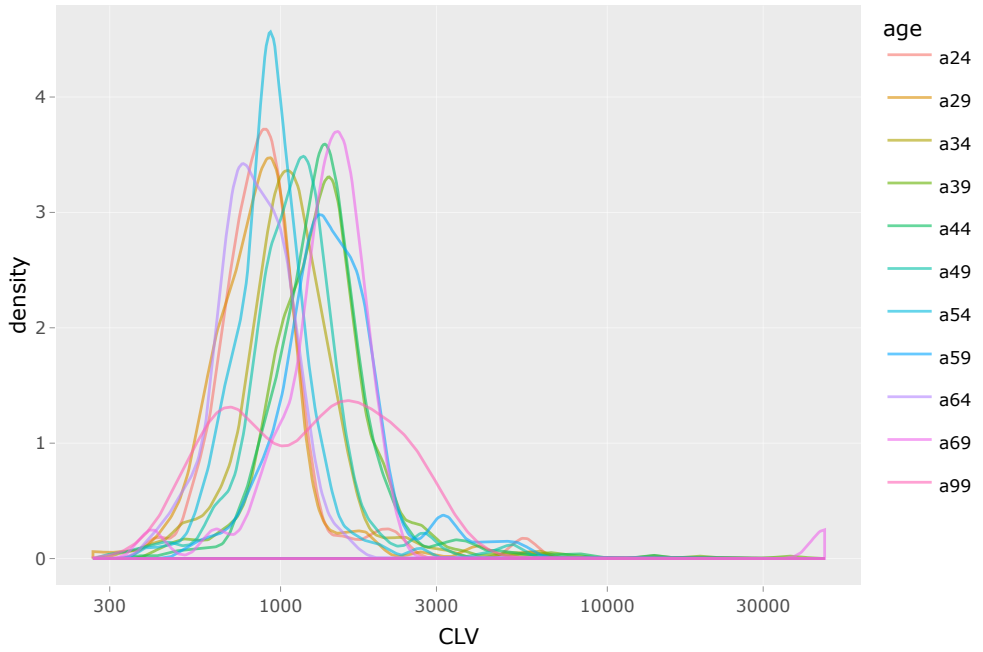
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   266.6   922.5  1148.0  1321.5  1437.4 46259.0
```

```
new1 =ggplot(new_B, aes(CLV)) +
  geom_histogram(bins=30, fill="green",alpha=0.6) +
  scale_x_log10()+
  ggtitle("new CLV")
ggplotly(new1)
```

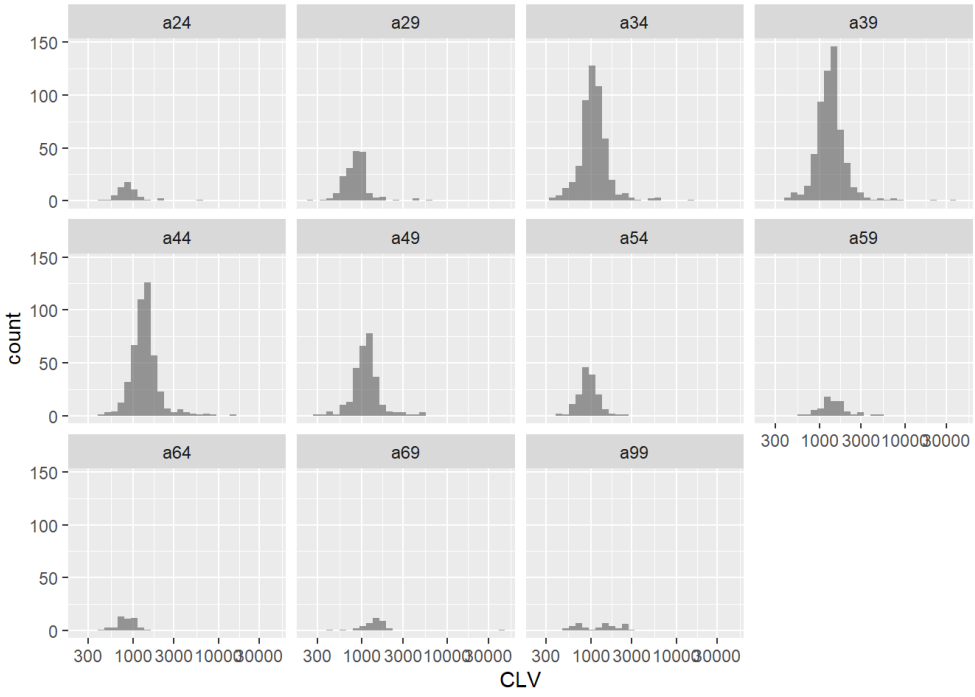


```
new2 = ggplot(new_B, aes(CLV,color= age)) +
  geom_density(alpha=0.6) +
  scale_x_log10()+
  ggtitle("new CLV by age")
ggplotly(new2)
```

new CLV by age

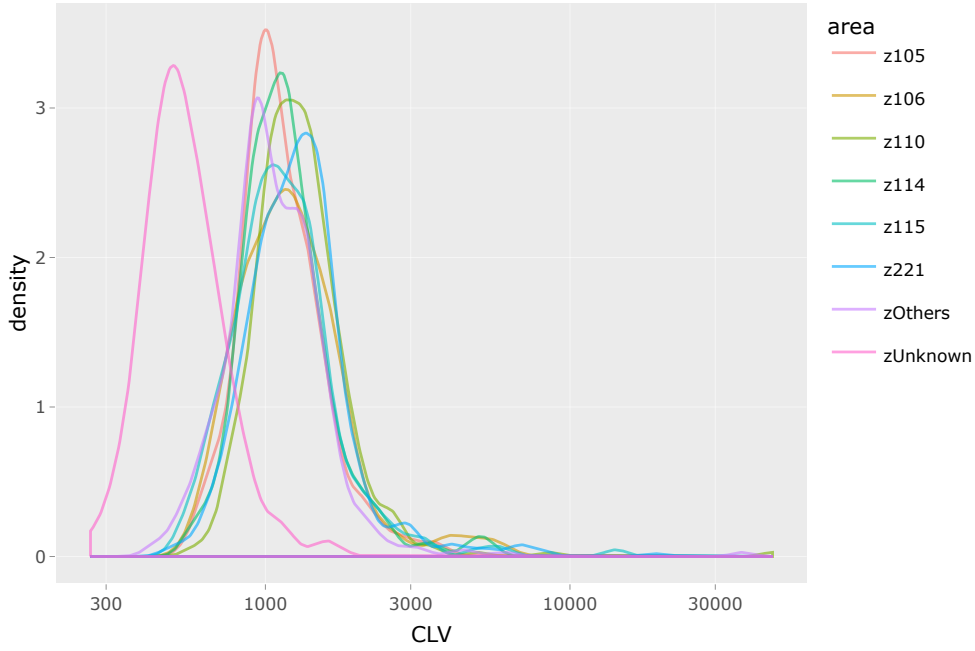



```
ggplot(new_B, aes(CLV)) +
  geom_histogram(bins=30,alpha=0.6) +
  scale_x_log10() +
  facet_wrap(~age)
```

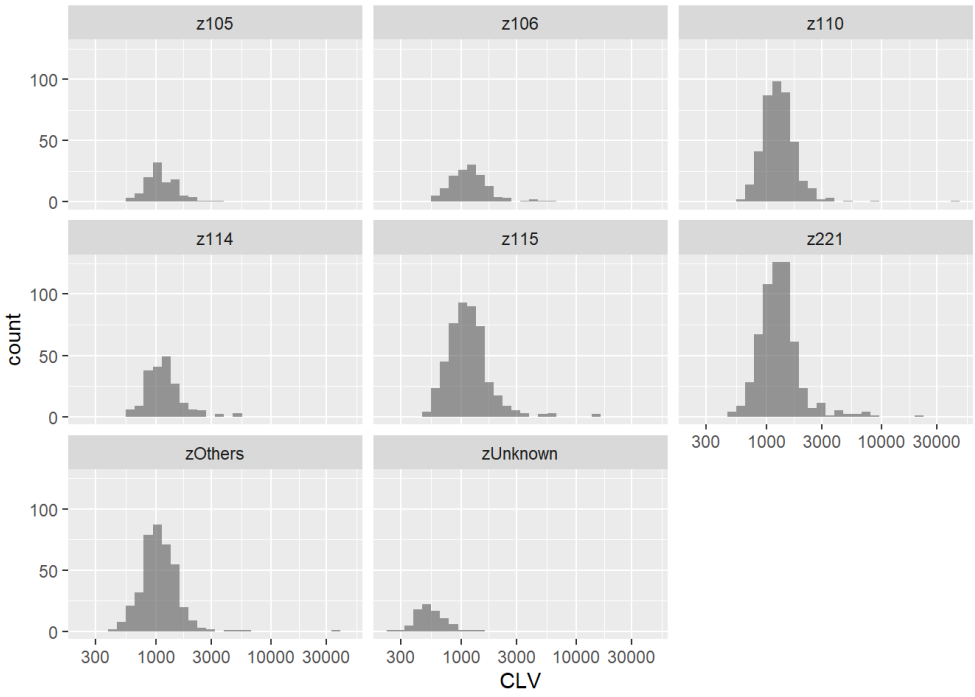



```
new_3 = ggplot(new_B, aes(CLV,color= area)) +
  geom_density(alpha=0.6) +
  scale_x_log10()+
  ggtitle("new CLV by area")
ggplotly(new_3)
```

new CLV by area




```
ggplot(new_B, aes(CLV)) +
  geom_histogram(bins=30,alpha=0.6) +
  scale_x_log10() +
  facet_wrap(~area)
```



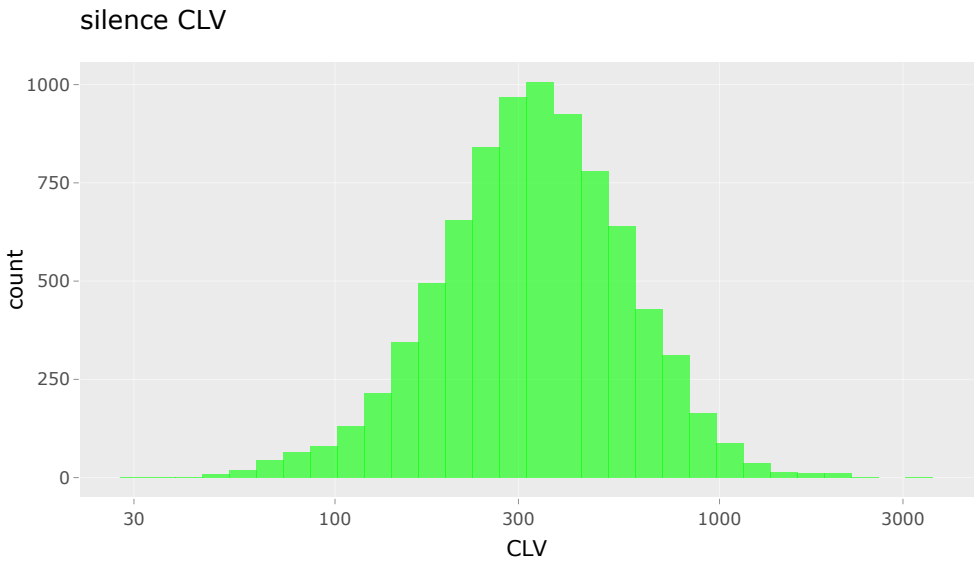
Silence

There're 8299 cust.

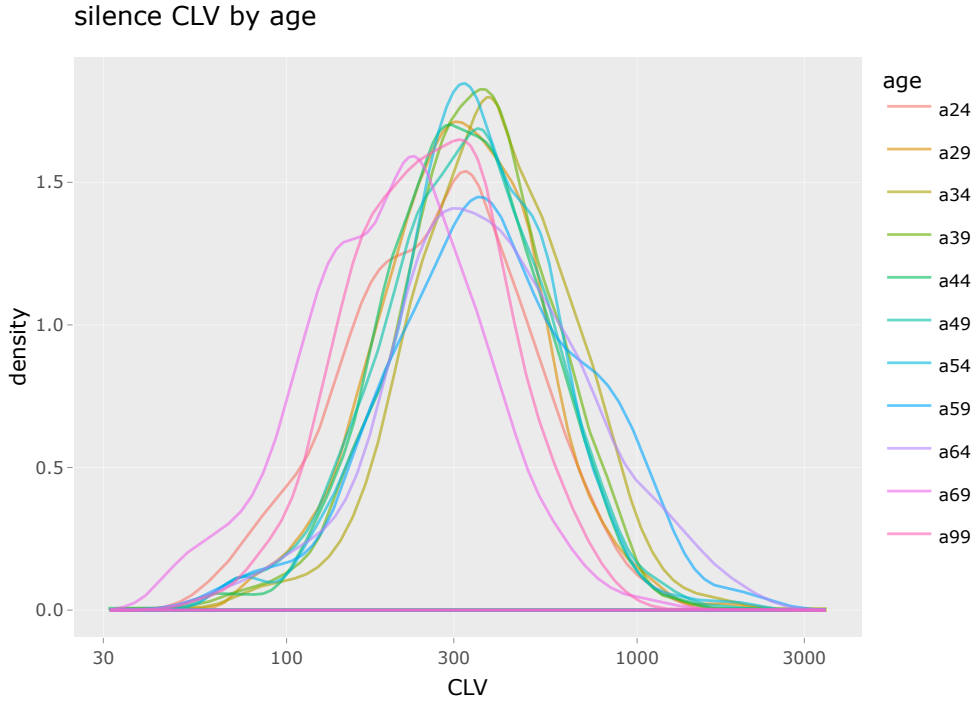
```
silence_B$CLV = g * silence_B$Rev * rowSums(sapply(
  0:N, function(i) (silence_B$Buy/(1+d))^i ) )
summary(silence_B$CLV)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   31.32  228.54  331.19  379.30  473.93 3446.23
```

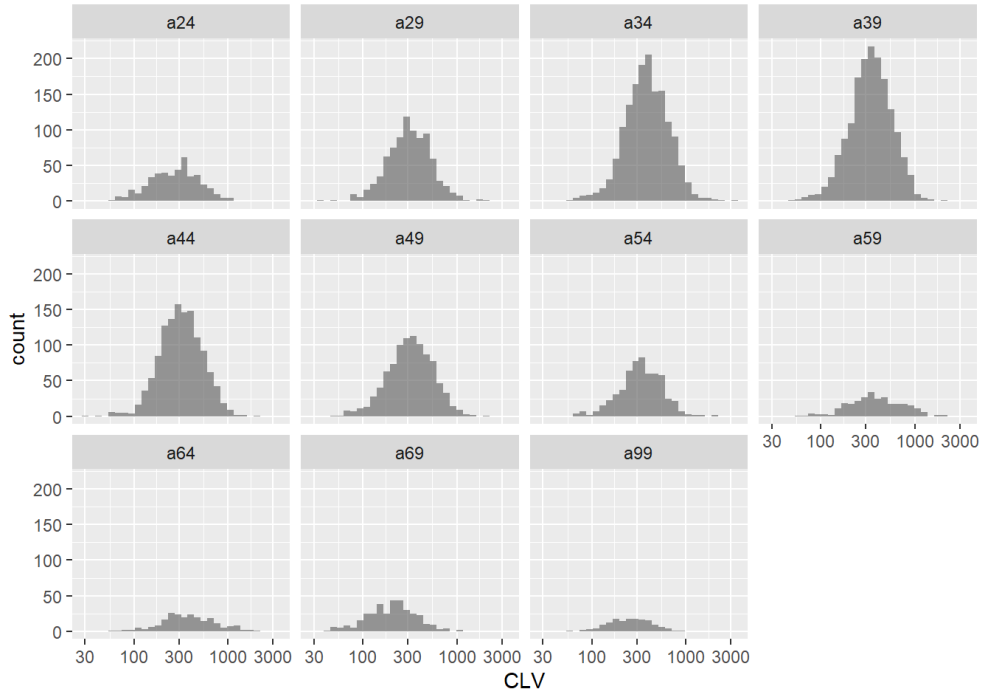
```
silence1 = ggplot(silence_B, aes(CLV)) +
  geom_histogram(bins=30, fill="green",alpha=0.6) +
  scale_x_log10()+
  ggtitle("silence CLV")
ggplotly(silence1)
```



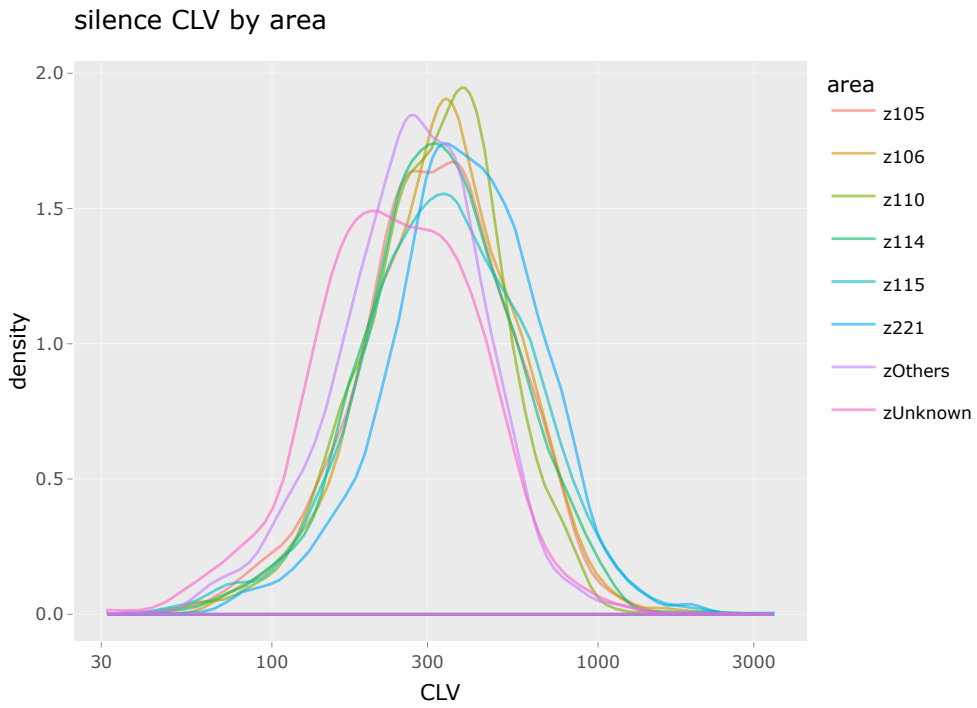

```
silence2 = ggplot(silence_B, aes(CLV,color= age)) +
  geom_density(alpha=0.6) +
  scale_x_log10()+
  ggtitle("silence CLV by age")
ggplotly(silence2)
```



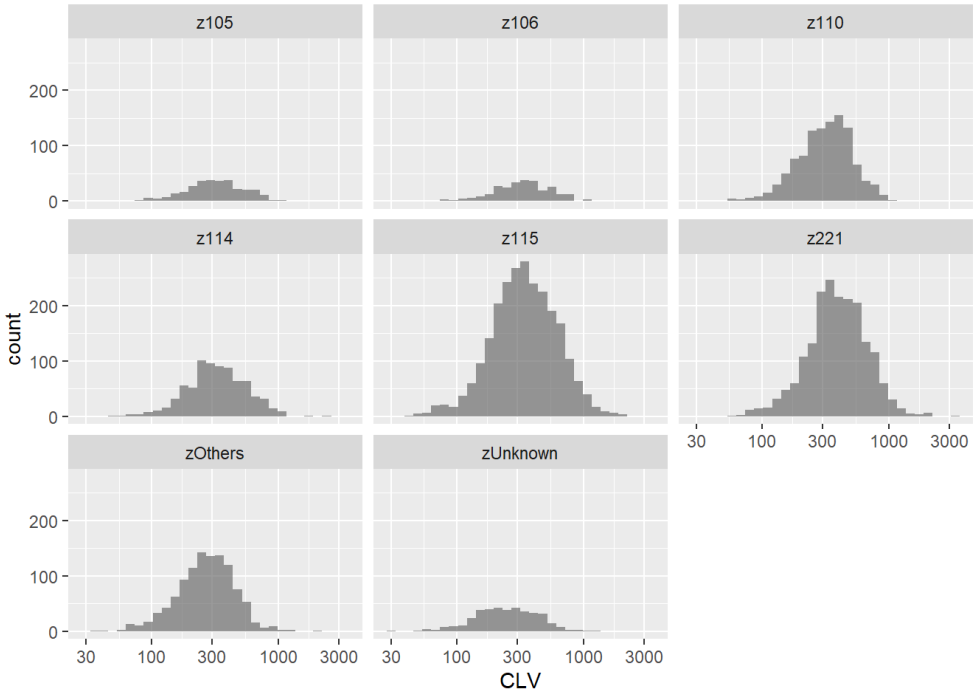

```
ggplot(silence_B, aes(CLV)) +
  geom_histogram(bins=30,alpha=0.6) +
  scale_x_log10() +
  facet_wrap(~age)
```




```
silence3 = ggplot(silence_B, aes(CLV,color= area)) +
  geom_density(alpha=0.6) +
  scale_x_log10()+
  ggtitle("silence CLV by area")
ggplotly(silence3)
```




```
ggplot(silence_B, aes(CLV)) +
  geom_histogram(bins=30,alpha=0.6) +
  scale_x_log10() +
  facet_wrap(~area)
```



Highp

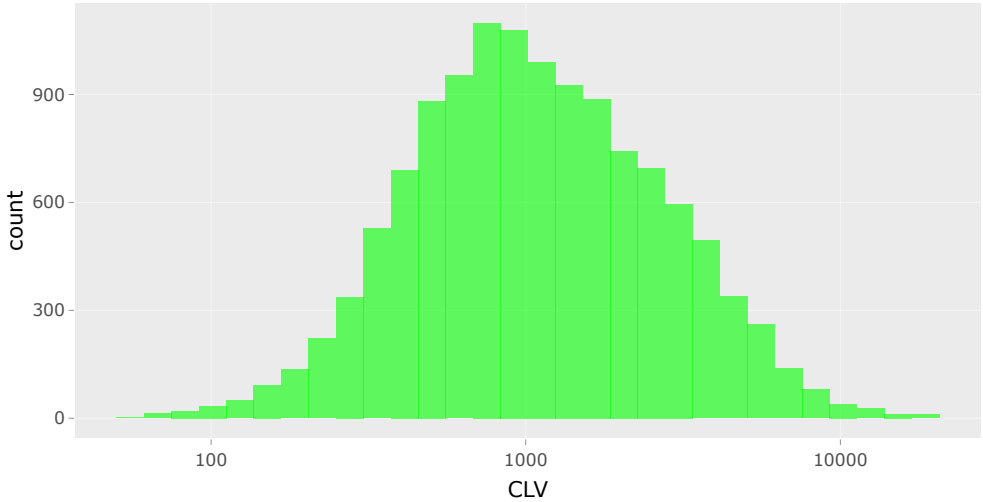
There're 12375 cust.

```
highp_B$CLV = g * highp_B$Rev * rowSums(sapply(
  0:N, function(i) (highp_B$Buy/(1+d))^i ) )
summary(highp_B$CLV)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    60.64   571.29  1027.98  1621.61  2028.68 20449.69
```

```
highp1 =ggplot(highp_B, aes(CLV)) +
  geom_histogram(bins=30, fill="green",alpha=0.6) +
  scale_x_log10()+
  ggtitle("highp CLV")
ggplotly(highp1)
```

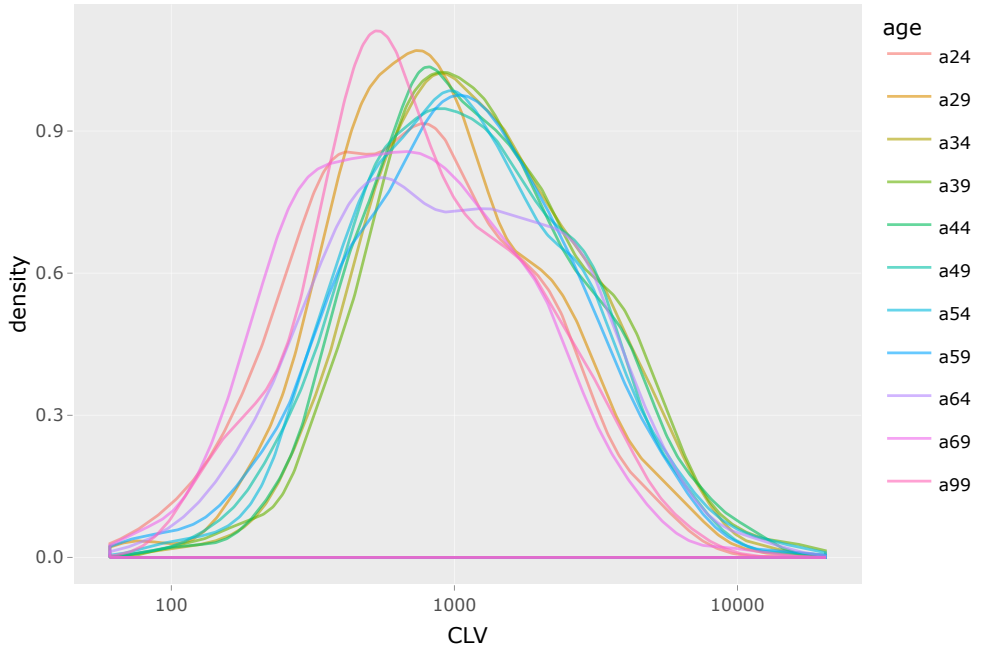
highp CLV



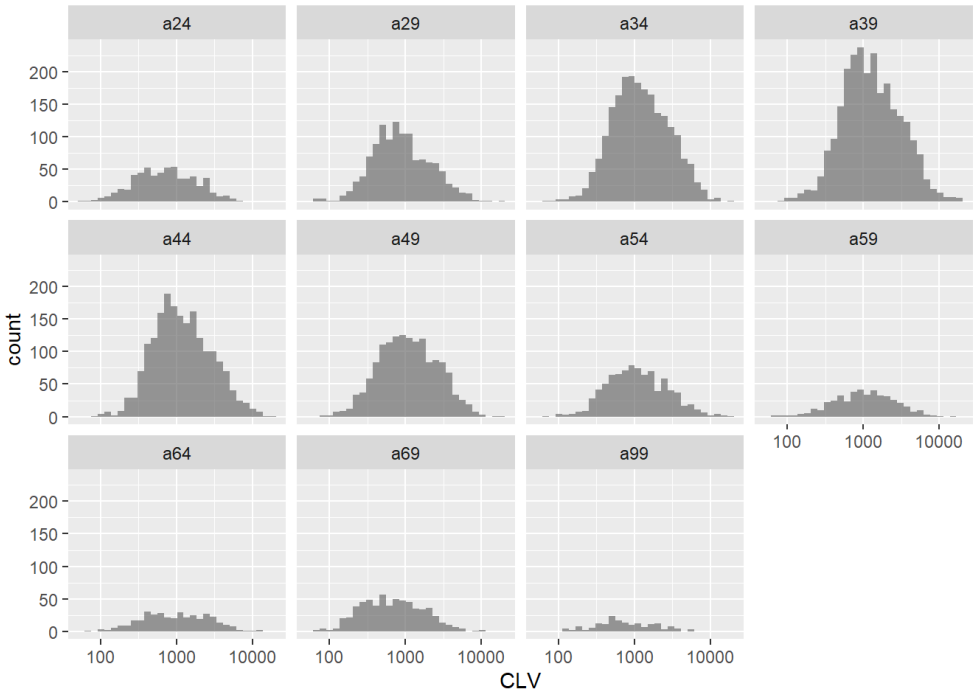

```
highp2 = ggplot(highp_B, aes(CLV,color= age)) +
  geom_density(alpha=0.6) +
  scale_x_log10()+
  ggtitle("highp CLV by age")
ggplotly(highp2)
```

highp CLV by age

highp CLV by age

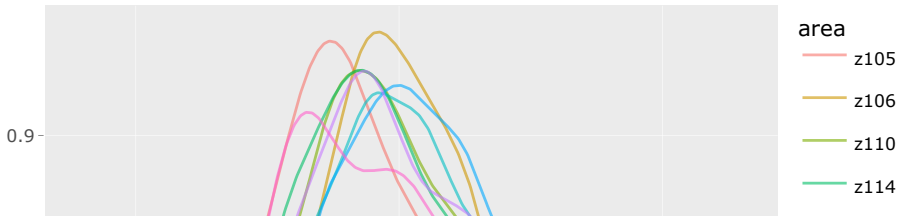


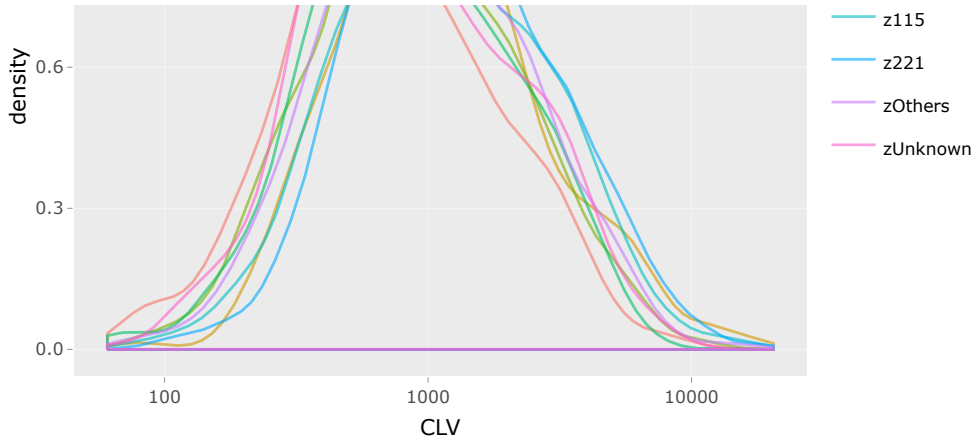

```
ggplot(highp_B, aes(CLV)) +
  geom_histogram(bins=30,alpha=0.6) +
  scale_x_log10() +
  facet_wrap(~age)
```



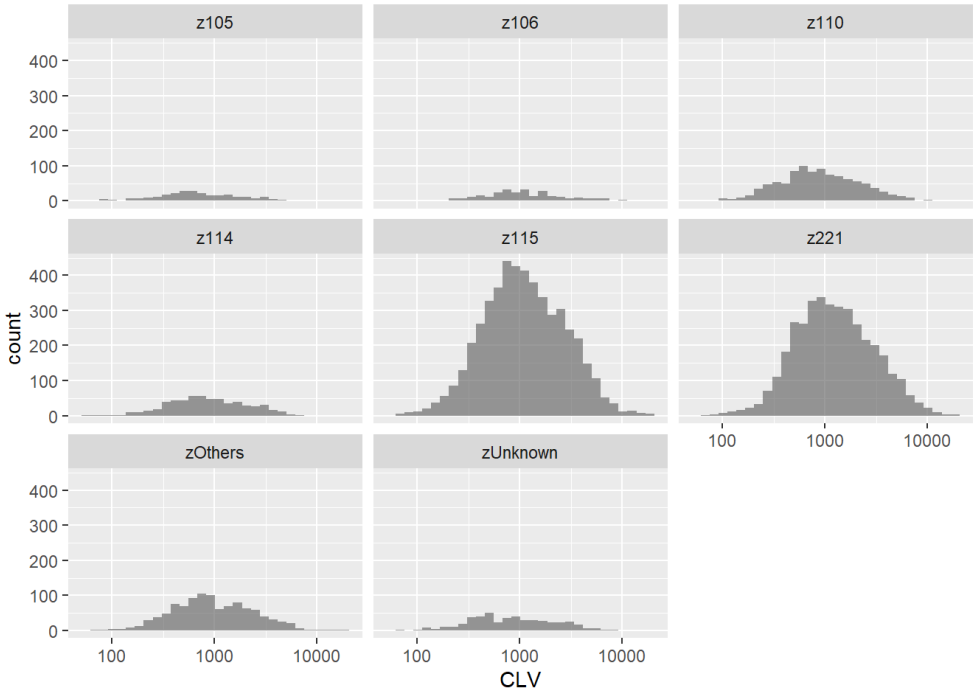

```
highp_3 = ggplot(highp_B, aes(CLV,color= area)) +
  geom_density(alpha=0.6) +
  scale_x_log10()+
  ggtitle("highp CLV by area")
ggplotly(highp_3)
```

highp CLV by area






```
ggplot(highp_B, aes(CLV)) +
  geom_histogram(bins=30,alpha=0.6) +
  scale_x_log10() +
  facet_wrap(~area)
```



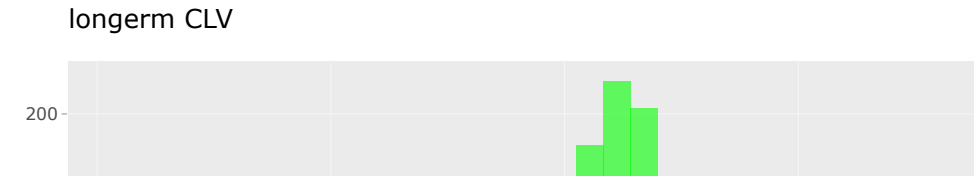
Longterm

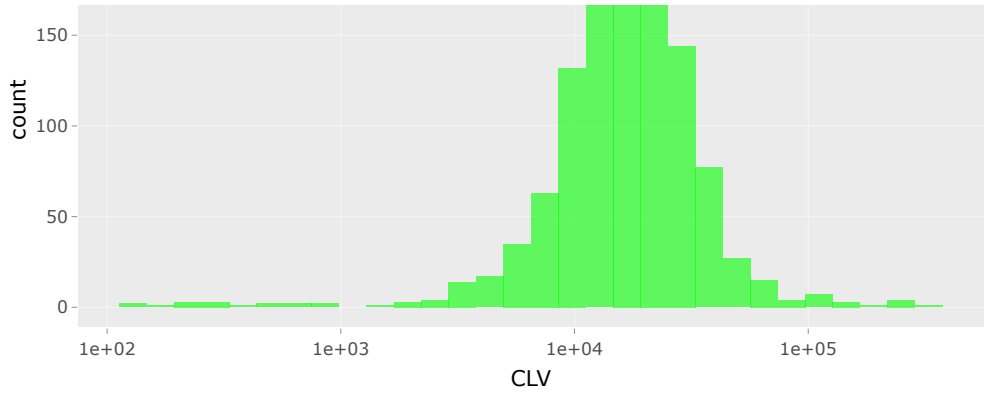
There're 1172 cust.

```
longterm_B$CLV = g * longterm_B$Rev * rowSums(sapply(
  0:N, function(i) (longterm_B$Buy/(1+d))^i ) )
summary(longterm_B$CLV)
```

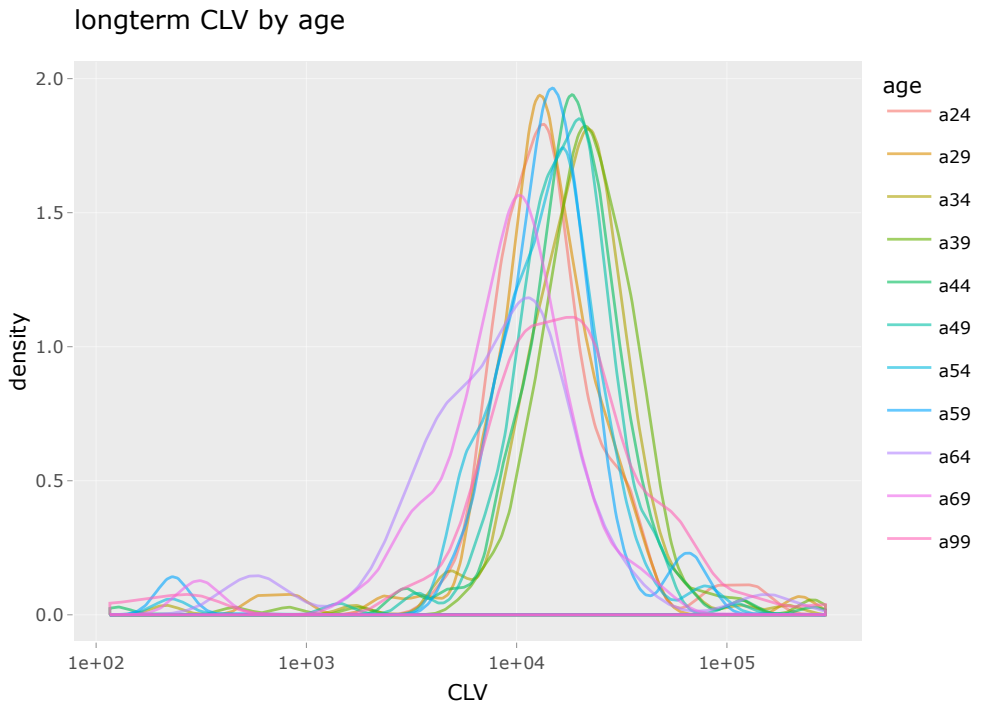
```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    116.2  11304.5  17116.0  21343.4  24603.9 294130.1
```

```
longterm1 = ggplot(longterm_B, aes(CLV)) +
  geom_histogram(bins=30, fill="green",alpha=0.6) +
  scale_x_log10()+
  ggtitle("longerm CLV")
ggplotly(longterm1)
```

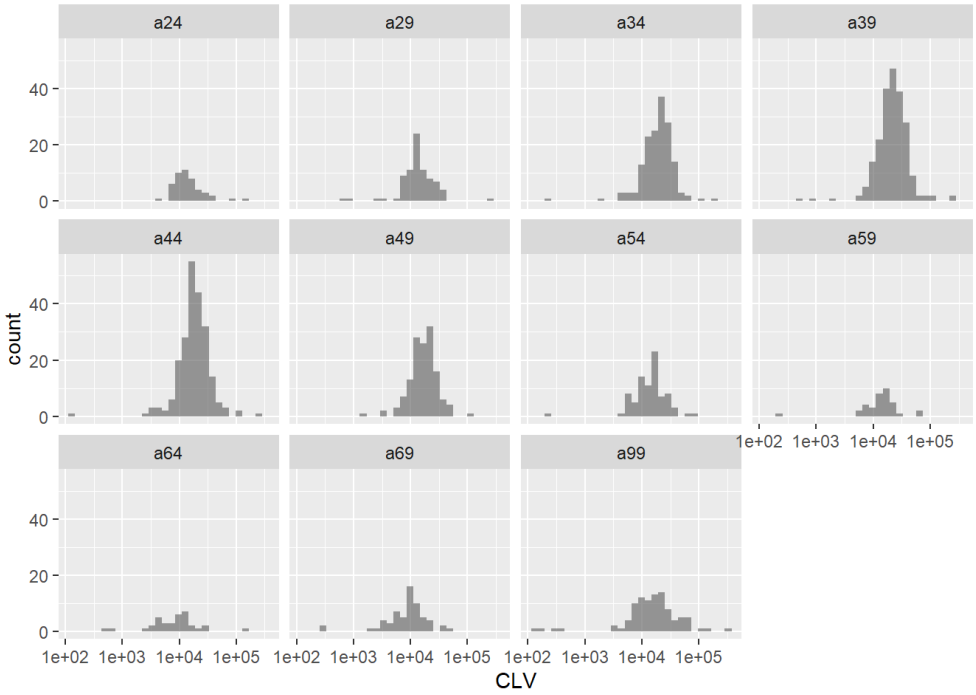




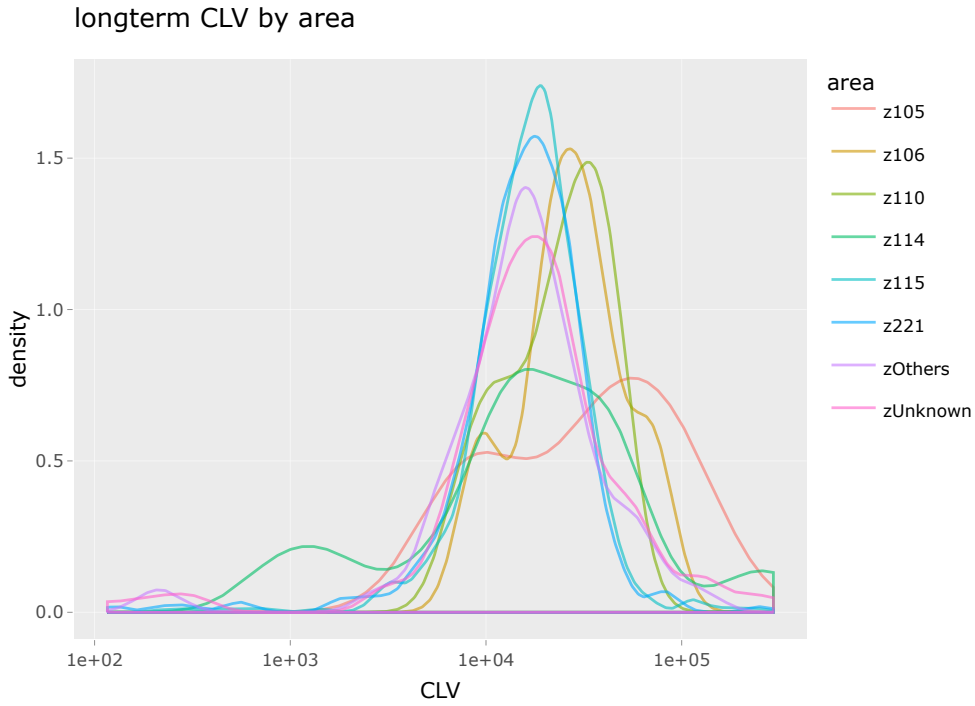

```
longterm2 = ggplot(longterm_B, aes(CLV,color= age)) +
  geom_density(alpha=0.6) +
  scale_x_log10()+
  ggtitle("longterm CLV by age")
ggplotly(longterm2)
```



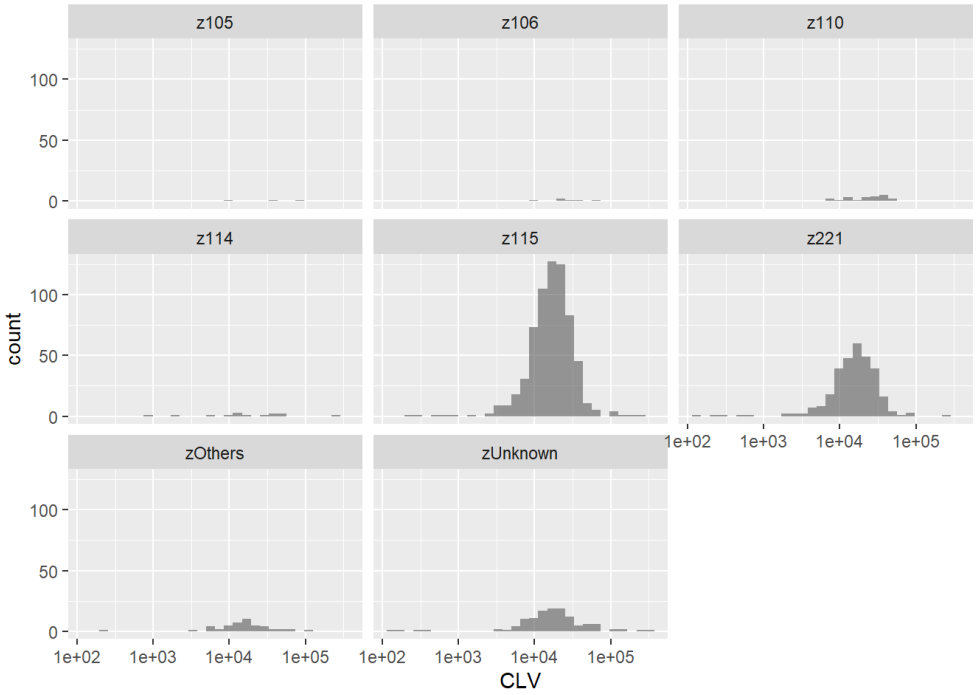

```
ggplot(longterm_B, aes(CLV)) +
  geom_histogram(bins=30,alpha=0.6) +
  scale_x_log10() +
  facet_wrap(~age)
```




```
longterm_3 = ggplot(longterm_B, aes(CLV,color= area)) +
  geom_density(alpha=0.6) +
  scale_x_log10()+
  ggtitle("longterm CLV by area")
ggplotly(longterm_3)
```




```
ggplot(longterm_B, aes(CLV)) +
  geom_histogram(bins=30,alpha=0.6) +
  scale_x_log10() +
  facet_wrap(~area)
```

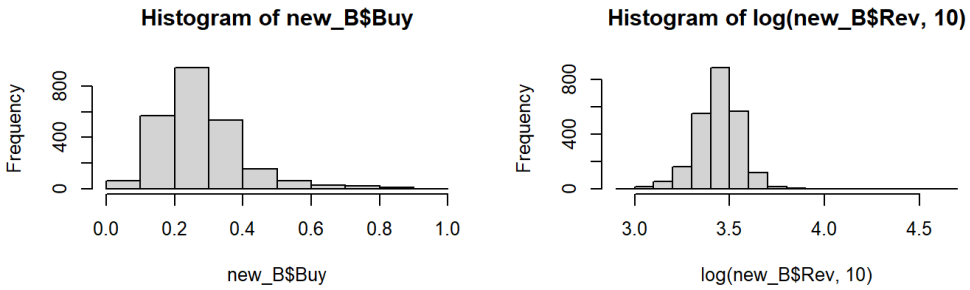


12. 使用模型做預測 Utilizing Model for Prediction

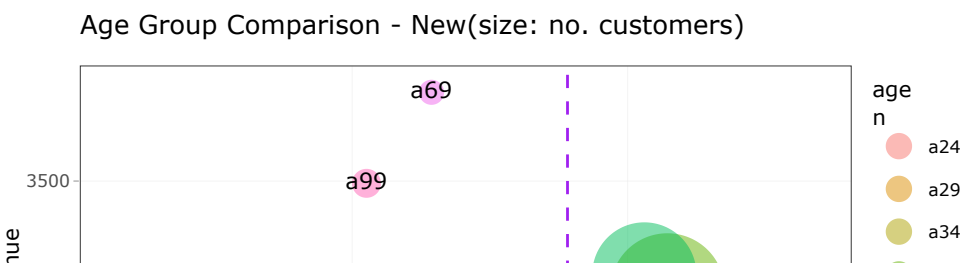
- `Buy` ：預期再購機率 Re-Purchase Probability
- `Rev` ：預期購買金額 Expected Revenue Contribution

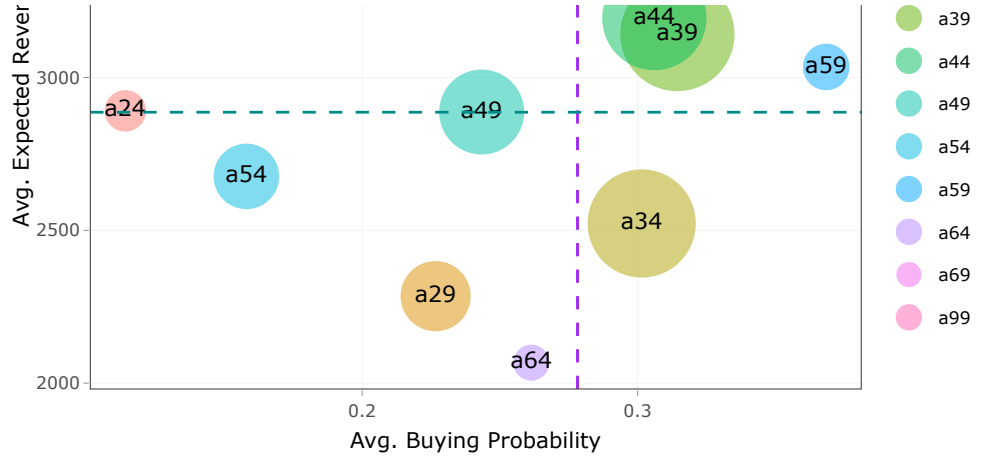
New

```
par(mfrow=c(1,2), cex=0.8)
hist(new_B$Buy)
hist(log(new_B$Rev,10))
```



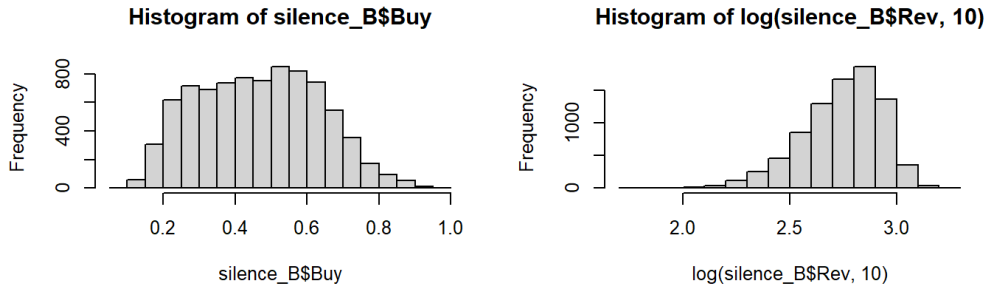

```
group_by(new_B,age) %>%
  summarise(n=n(), Buy=mean(Buy), Rev=mean(Rev)) %>%
  ggplot(aes(Buy,Rev,size=n,label=age)) +
  geom_point(alpha=0.5,aes(col=age)) +
  geom_text(size=4) +
  labs(title="Age Group Comparison - New(size: no. customers)") +
  xlab("Avg. Buying Probability") + ylab("Avg. Expected Revenue") +
  scale_size(range=c(4,20)) +
  theme_bw()  +
  geom_vline(xintercept = sum(new_B$Buy)/length(new_B$Buy), col="purple", linetype = "dashed")+
  geom_hline(yintercept = sum(new_B$Rev)/length(new_B$Rev), col="darkcyan", linetype="dashed")   -> p
ggplotly(p)
```





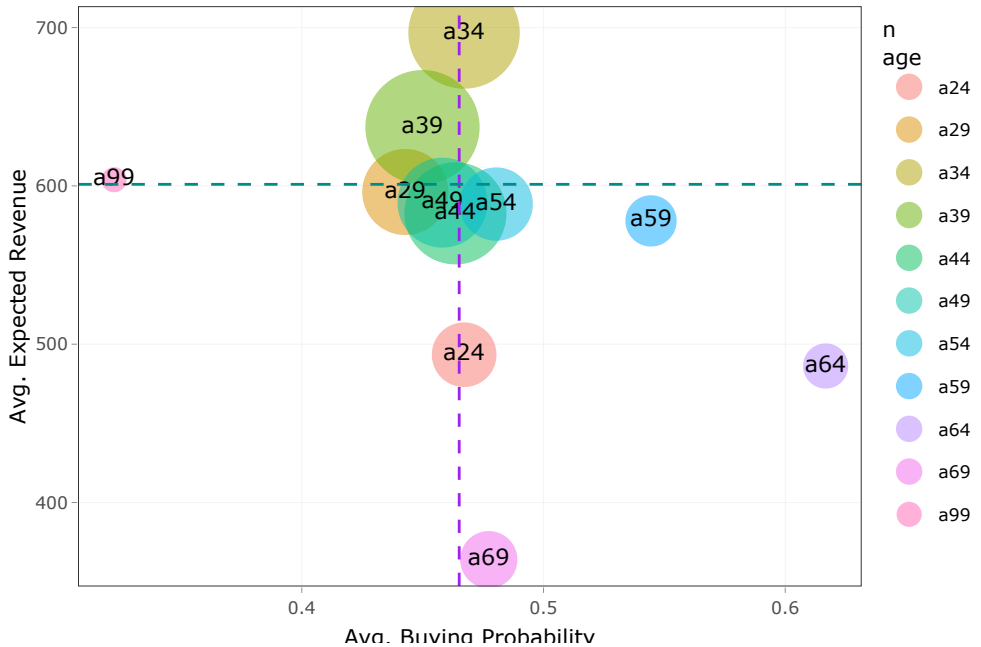
Silence

```
par(mfrow=c(1,2), cex=0.8)
hist(silence_B$Buy)
hist(log(silence_B$Rev,10))
```



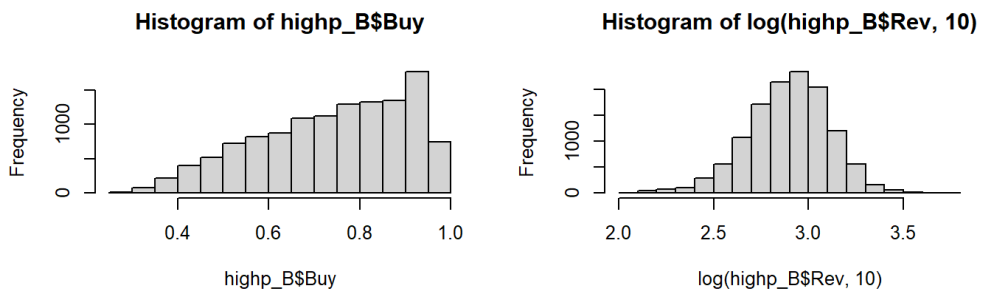

```
group_by(silence_B,age) %>%
  summarise(n=n(), Buy=mean(Buy), Rev=mean(Rev)) %>%
  ggplot(aes(Buy,Rev,size=n,label=age)) +
  geom_point(alpha=0.5,aes(col=age)) +
  geom_text(size=4) +
  labs(title="Age Group Comparison - Silence(size: no. customers)") +
  xlab("Avg. Buying Probability") + ylab("Avg. Expected Revenue") +
  scale_size(range=c(4,20)) + theme_bw() +
  geom_vline(xintercept = sum(silence_B$Buy)/length(silence_B$Buy), col="purple", linetype = "dashed")+
  geom_hline(yintercept = sum(silence_B$Rev)/length(silence_B$Rev), col="darkcyan", linetype="dashed")   -> p
ggplotly(p)
```

Age Group Comparison - Silence(size: no. customers)

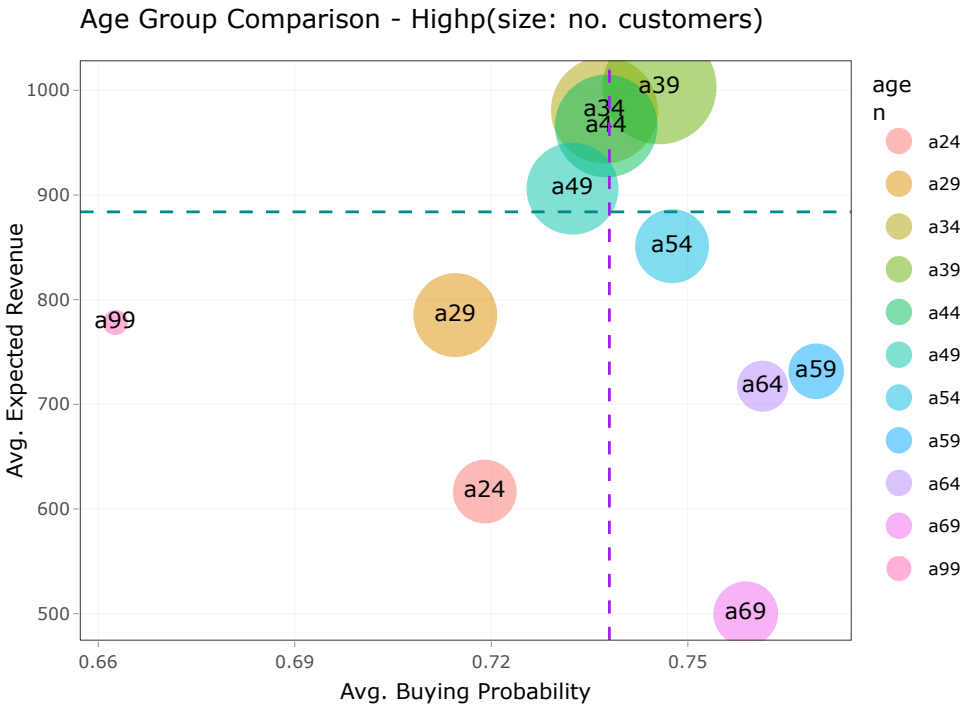


Highp

```
par(mfrow=c(1,2), cex=0.8)
hist(highp_B$Buy)
hist(log(highp_B$Rev,10))
```

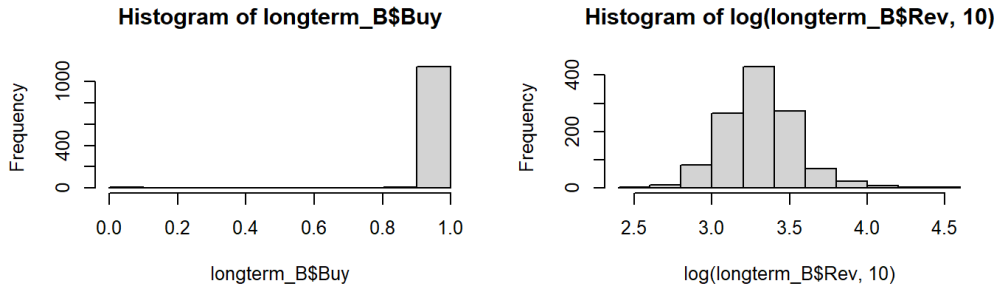



```
group_by(highp_B,age) %>%
  summarise(n=n(), Buy=mean(Buy), Rev=mean(Rev)) %>%
  ggplot(aes(Buy,Rev,size=n,label=age)) +
  geom_point(alpha=0.5,aes(col=age)) +
  geom_text(size=4) +
  labs(title="Age Group Comparison - Highp(size: no. customers)") +
  xlab("Avg. Buying Probability") + ylab("Avg. Expected Revenue") +
  scale_size(range=c(4,20)) + theme_bw() +
  geom_vline(xintercept = sum(highp_B$Buy)/length(highp_B$Buy), col="purple", linetype = "dashed")+
  geom_hline(yintercept = sum(highp_B$Rev)/length(highp_B$Rev), col="darkcyan", linetype="dashed")   -> p
ggplotly(p)
```

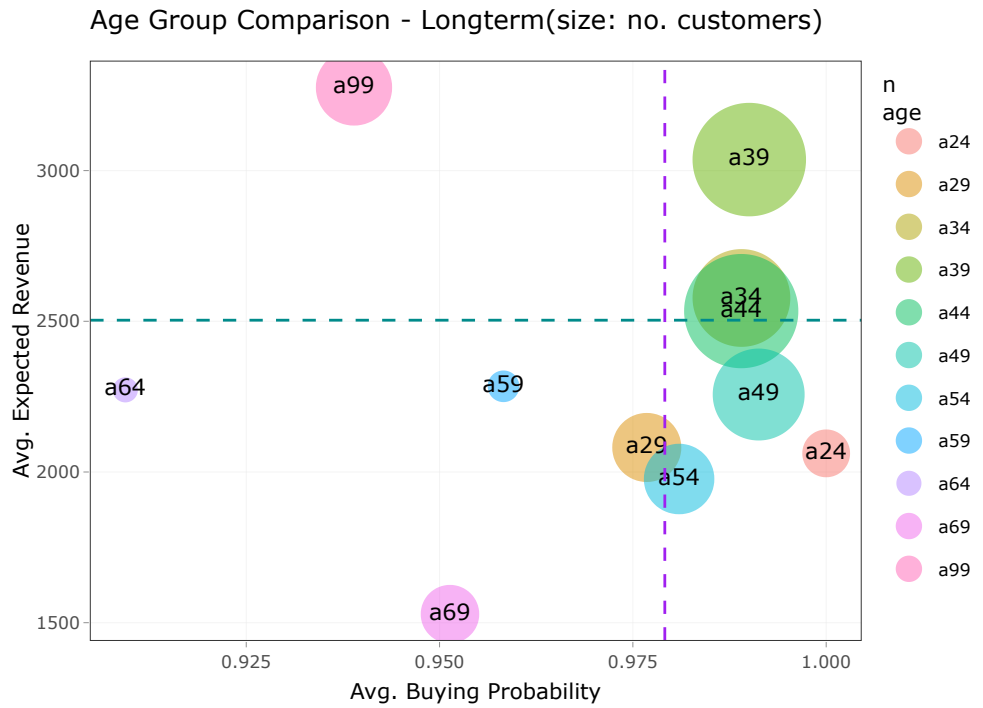


Longterm

```
par(mfrow=c(1,2), cex=0.8)
hist(longterm_B$Buy)
hist(log(longterm_B$Rev,10))
```




```
group_by(longterm_B,age) %>%
  summarise(n=n(), Buy=mean(Buy), Rev=mean(Rev)) %>%
  ggplot(aes(Buy,Rev,size=n,label=age)) +
  geom_point(alpha=0.5,aes(col=age)) +
  geom_text(size=4) +
  labs(title="Age Group Comparison - Longterm(size: no. customers)") +
  xlab("Avg. Buying Probability") + ylab("Avg. Expected Revenue") +
  scale_size(range=c(4,20)) +
  theme_bw()+
  geom_vline(xintercept = sum(longterm_B$Buy)/length(longterm_B$Buy), col="purple", linetype = "dashed")+
  geom_hline(yintercept = sum(longterm_B$Rev)/length(longterm_B$Rev), col="darkcyan", linetype="dashed")   -> p
ggplotly(p)
```



13. 成本效益函數 - 帶參數的假設 Cost Benefit Analysis

§ S曲線 (S-Curve)

🌻 S-Curve : 許多管理工具都呈現S型的成本效益函數

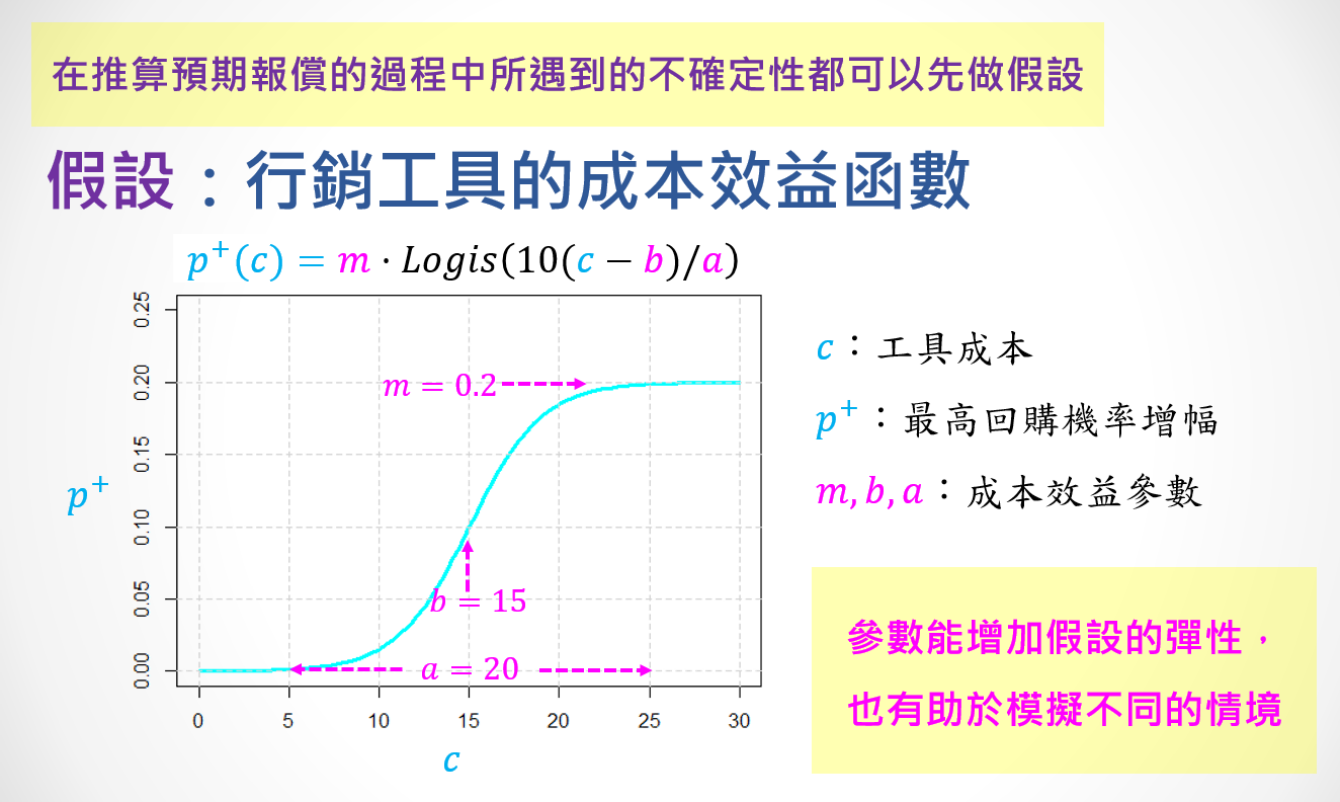
🌻 我們可以用R內建的邏輯式函數(`plogis()`)來模擬S曲線

$$\Delta P(x|m,b,a) = m \cdot Logis(\frac{10(x-b)}{a})$$

```
DP = function(x,m0,b0,a0) {m0*plogis((10/a0)*(x-b0))}
# X = 成本/ delta P = 購買效益
# DP = 機率增幅
# S曲線:用意單一函數，來衡量個行銷工具的成本效益
#調整m,b,a，即是調整不同的工具，或同一(不同)工具對不同族群的成本效益關係
```

§ **帶參數**的成本效益函數 (S曲線)

🌻 parameters(參數)可以帶入彈性，放寬模擬的範圍



🌻 透過這3個parameters(參數):

- `m` : 最大效果
- `b` : 效果的位置(上升波段的中點)*b越大成本越高、效益越低
- `a` : 效果的範圍(上升波段的寬度)

我們可以寫『一支程式』來模擬『所有可能』的成本效益函數(S曲線)
藉以描述策略變數(x,折價卷面額)和策略效果(ΔP,購買機率增幅)之間的關係

§ 估計預期獲利

有了行銷工具的成本效益函數之後，我們就可以估計將這個工具用在每一位顧客上的時候的預期效益:

$$\hat{R}(x) = \begin{cases} \Delta P \cdot M \cdot margin - x & , \quad P + \Delta P \leq 1 \\ (1-P) \cdot M \cdot margin - x & , \quad else \end{cases}$$

🌻 結合 ...

- 預測 (P, M) : 每位顧客的預期購買機率和購買金額，與
- 假設 $(\Delta P(x|m,b,a))$: 行銷工具帶來的再購機率增額

我們就可以估計這個工具用在每位顧客上的預期效益 $\hat{R}(x)$。

🌻 Note that both ΔP and $\hat{R}$ are functions of x given m, b, a

- P, M 預期購買機率和金額，是預測
- $Margin, X$ 利潤率和成本
- m, b, a 行銷工具的屬性，是假設
- x 行銷強度，是我們可以操作的、想要優化的策略變數

New

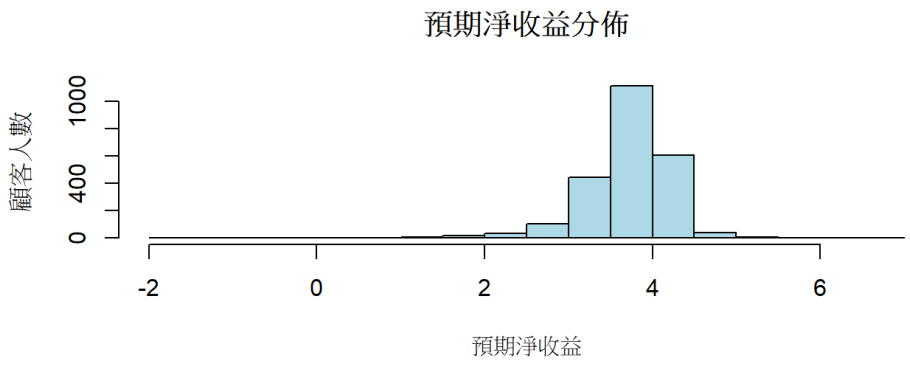
估計毛利率 m

```
# load(data/tf0.rdata)
# group_by(Z0, age) %>% summarise(sum(price)/sum(cost) - 1)
margin = 0.17  # assume margin = 0.17
```

估計每位顧客的淨收益 $\hat{R}(x)$

```
m=0.2; b=25; a=40; x=30
new_dp = pmin(1-new_B$Buy, DP(x,m,b,a)) # 機率增幅不可以超過1，有些人本來的購買機率就已經95%，如果增加0.15會超過1，所以該
         購買者最多只能增加0.05，因此才是1-B$Buy, DP(x,m,b,a)，取最小
new_eR = new_dp*new_B$Rev*margin - x
# eR = Expected return = 機率增幅*預期營收(Rev)*利潤率 - 成本
hist(log(new_eR),main="預期淨收益分佈",xlab="預期淨收益",ylab="顧客人數",col = "lightblue")
```

```
## Warning in log(new_eR): NaNs produced
```



Silence

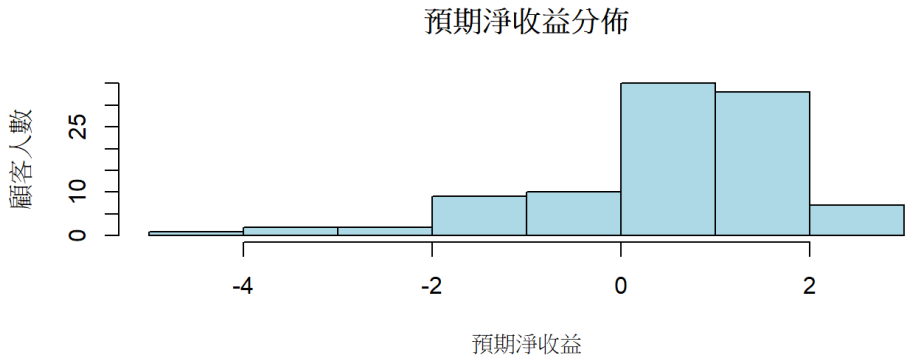
估計毛利率 m

```
# load(data/tf0.rdata)
# group_by(Z0, age) %>% summarise(sum(price)/sum(cost) - 1)
margin = 0.17  # assume margin = 0.17
```

估計每位顧客的淨收益 $\hat{R}(x)$

```
m=0.2; b=25; a=40; x=30
silence_dp = pmin(1-silence_B$Buy, DP(x,m,b,a)) # 機率增幅不可以超過1，有些人本來的購買機率就已經95%，如果增加0.15會超過
        1，所以該購買者最多只能增加0.05，因此才是1-B$Buy, DP(x,m,b,a)，取最小
silence_eR = silence_dp*silence_B$Rev*margin - x
# eR = Expected return = 機率增幅*預期營收(Rev)*利潤率 - 成本
hist(log(silence_eR),main="預期淨收益分佈",xlab="預期淨收益",ylab="顧客人數",col = "lightblue")
```

```
## Warning in log(silence_eR): NaNs produced
```



Highp

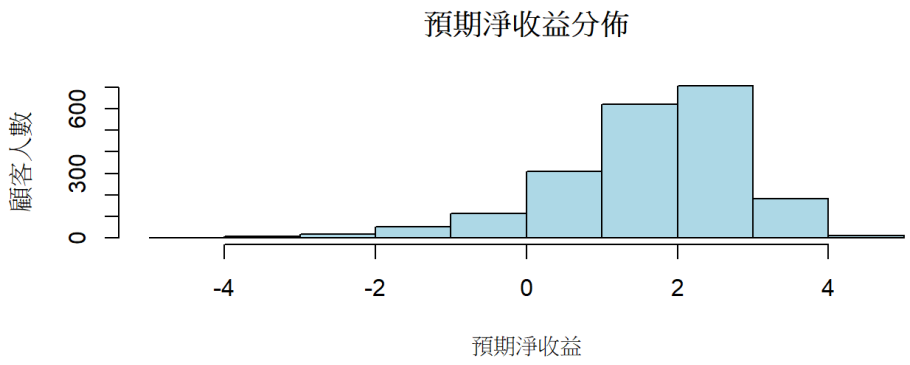
估計毛利率 m

```
# load(data/tf0.rdata)
# group_by(Z0, age) %>% summarise(sum(price)/sum(cost) - 1)
margin = 0.17  # assume margin = 0.17
```

估計每位顧客的淨收益 $\hat{R}(x)$

```
m=0.2; b=25; a=40; x=30
highp_dp = pmin(1-highp_B$Buy, DP(x,m,b,a)) # 機率增幅不可以超過1，有些人本來的購買機率就已經95%，如果增加0.15會超過1，所以該購買者最多只能增加0.05，因此才是1-B$Buy, DP(x,m,b,a)，取最小
highp_eR = highp_dp*highp_B$Rev*margin - x
# eR = Expected return = 機率增幅*預期營收(Rev)*利潤率 - 成本
hist(log(highp_eR),main="預期淨收益分佈",xlab="預期淨收益",ylab="顧客人數",col = "lightblue")
```

```
## Warning in log(highp_eR): NaNs produced
```



Longterm

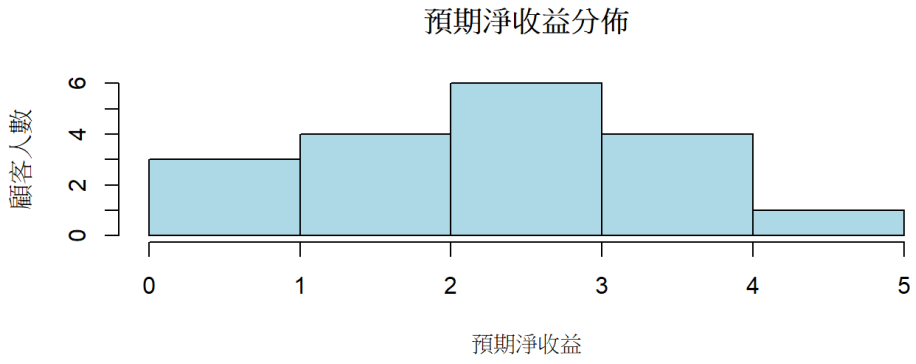
估計毛利率 m

```
# load(data/tf0.rdata)
# group_by(Z0, age) %>% summarise(sum(price)/sum(cost) - 1)
margin = 0.17  # assume margin = 0.17
```

估計每位顧客的淨收益 $\hat{R}(x)$

```
m=0.2; b=25; a=40; x=30
longterm_dp = pmin(1-longterm_B$Buy, DP(x,m,b,a)) # 機率增幅不可以超過1，有些人本來的購買機率就已經95%，如果增加0.15會超過1，所以該購買者最多只能增加0.05，因此才是1-B$Buy, DP(x,m,b,a)，取最小
longterm_eR = longterm_dp*longterm_B$Rev*margin - x
# eR = Expected return = 機率增幅*預期營收(Rev)*利潤率 - 成本
hist(log(longterm_eR),main="預期淨收益分佈",xlab="預期淨收益",ylab="顧客人數",col = "lightblue")
```

```
## Warning in log(longterm_eR): NaNs produced
```



根據以上的分析結果 ...

New

🚴 有多少顧客的淨預期報償大於零？(`eR > 0`)?

```
sum(new_eR>0)
```

```
## [1] 2387
```

🚴 如果我們針對所有顧客做促銷，總預期報償將是？

```
sum(new_eR)
```

```
## [1] 110532.2
```

🚴 如果我們針對預期報償大於零的顧客做促銷，預期報償將是？

```
sum(new_eR[ new_eR>0 ])
```

```
## [1] 110564.6
```

```
# 選擇性行銷賺七萬多
```

🚴 如果我們只針對預期報償大於10的顧客做促銷，預期報償將是？

```
sum(new_eR[ new_eR>10 ])
```

```
## [1] 110250.8
```

🚴 如果我們只針對預期報償大於10的南港(z115)顧客做促銷，預期報償將是？

```
sum(new_eR[ new_eR>10 & new_B$area=="z115"])
```

```
## [1] 18926.21
```

🚴 如果我們只針對預期報償大於10的汐止(z221)顧客做促銷，預期報償將是？

```
sum(new_eR[ new_eR>10 & new_B$area=="z221"])
```

```
## [1] 31516.54
```

Silence

🚴 有多少顧客的淨預期報償大於零？(eR > 0)?

```
sum(silence_eR>0)
```

```
## [1] 99
```

🚴 如果我們針對所有顧客做促銷，總預期報償將是？

```
sum(silence_eR)
```

```
## [1] -117356.4
```

🚴 如果我們針對預期報償大於零的顧客做促銷，預期報償將是？

```
sum(silence_eR[ silence_eR>0 ])
```

```
## [1] 287.9052
```

```
# 選擇性行銷賺七萬多
```

🚴 如果我們只針對預期報償大於10的顧客做促銷，預期報償將是？

```
sum(silence_eR[ silence_eR>10 ])
```

```
## [1] 22.66438
```

🚴 如果我們只針對預期報償大於10的南港(z115)顧客做促銷，預期報償將是？

```
sum(silence_eR[ silence_eR>10 & silence_B$area=="z115"   ])
```

```
## [1] 0
```

🚴 如果我們只針對預期報償大於10的汐止(z221)顧客做促銷，預期報償將是？

```
sum(silence_eR[ silence_eR>10 & silence_B$area=="z221"])
```

```
## [1] 11.92013
```

Highp

🚴 有多少顧客的淨預期報償大於零？(eR > 0)?

```
sum(highp_eR>0)
```

```
## [1] 2025
```

🚴 如果我們針對所有顧客做促銷，總預期報償將是？

```
sum(highp_eR)
```

```
## [1] -126168.9
```

🚴 如果我們針對預期報償大於零的顧客做促銷，預期報償將是？

```
sum(highp_eR[ highp_eR>0 ])
```

```
## [1] 18247.01
```

```
# 選擇性行銷賺七萬多
```

🚴 如果我們只針對預期報償大於10的顧客做促銷，預期報償將是？

```
sum(highp_eR[ highp_eR>10 ])
```

```
## [1] 12769.66
```

🚴 如果我們只針對預期報償大於10的南港(z115)顧客做促銷，預期報償將是？

```
sum(highp_eR[ highp_eR>10 & highp_B$area=="z115"   ])
```

```
## [1] 3629.628
```

🚴 如果我們只針對預期報償大於10的汐止(z221)顧客做促銷，預期報償將是？

```
sum(highp_eR[highp_eR>10 & highp_B$area=="z221"])
```

```
## [1] 5380.305
```

Longterm

🚴 有多少顧客的淨預期報償大於零？(eR > 0)?

```
sum(longterm_eR>0)
```

```
## [1] 18
```

🚴 如果我們針對所有顧客做促銷，總預期報償將是？

```
sum(longterm_eR)
```

```
## [1] -31406.71
```

🚴 如果我們針對預期報償大於零的顧客做促銷，預期報償將是？

```
sum(longterm_eR[ longterm_eR>0 ])
```

```
## [1] 297.3149
```

```
# 選擇性行銷賺七萬多
```

🚴 如果我們只針對預期報償大於10的顧客做促銷，預期報償將是？

```
sum(longterm_eR[ longterm_eR>10 ])
```

```
## [1] 269.2919
```

🚴 如果我們只針對預期報償大於10的南港(z115)顧客做促銷，預期報償將是？

```
sum(longterm_eR[ longterm_eR>10 & longterm_B$area=="z115"   ])
```

```
## [1] 100.3411
```

🚴 如果我們只針對預期報償大於10的汐止(z221)顧客做促銷，預期報償將是？

```
sum(longterm_eR[ longterm_eR>10 & longterm_B$area=="z221"])
```

```
## [1] 28.25648
```

14.策略市場模擬(重新設參數!!) Simulate Marketing Strategies

給定工具參數(m, b, a)，我們可在其有效成本範圍($x \in [b - \frac{a}{2}, b + \frac{a}{2}]$)之內，估計工具的效果：

- `eReturn`：對所有的人行銷的總預期收益
- `N`：預期收益大於零的人數
- `eReturn2`：只對期收益大於零的人做行銷的總預期收益

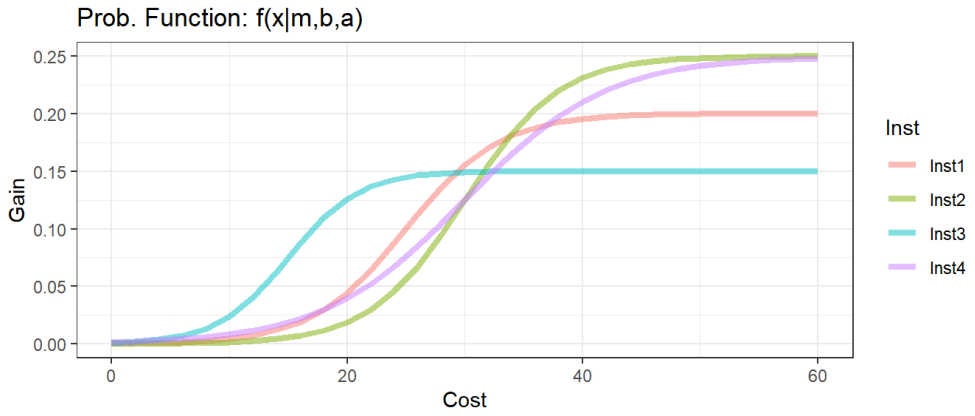
如何隨成本變化。

§ 多個行銷工具

稍微改一下程式，我們可以同時模擬多(4)個行銷工具，並比較他們的成本效益 With some modification of the code, we can define multiple (4) instruments

```
mm=c(0.20, 0.25, 0.15, 0.25)
bb=c(  25,   30,   15,   30)
aa=c(  40,   40,   30,   60)
X = seq(0,60,2)
do.call(rbind, lapply(1:length(mm), function(i) data.frame(
  Inst=paste0('Inst',i), Cost=X,
  Gain=DP(X,mm[i],bb[i],aa[i])
  ))) %>% data.frame %>%
  ggplot(aes(x=Cost, y=Gain, col=Inst)) +
  geom_line(size=1.5,alpha=0.5) + theme_bw() +
  ggtitle("Prob. Function: f(x|m,b,a)")
```

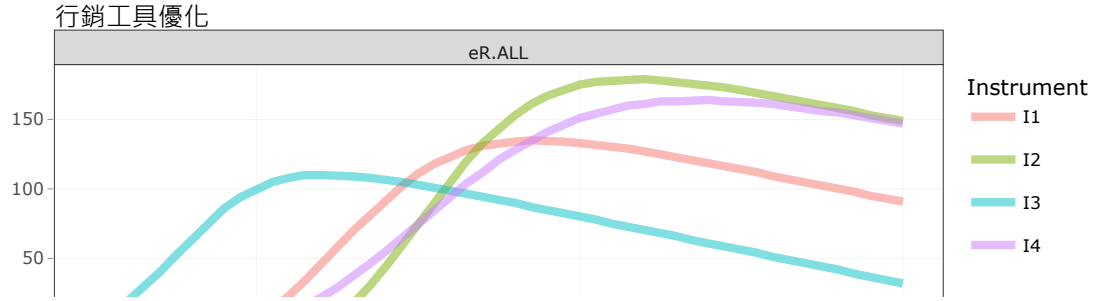
```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

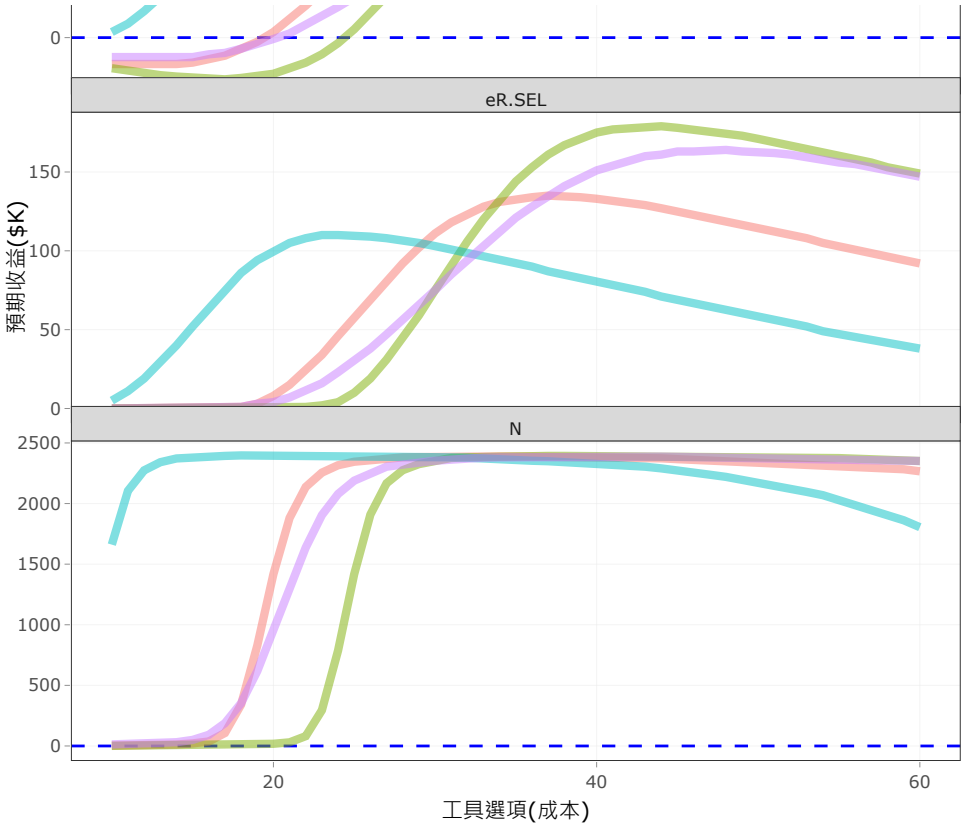


and run simulation on multiple instrument to compare their cost effectiveness.

New

```
X = seq(10, 60, 1)
new_df = do.call(rbind, lapply(1:length(mm), function(i) {
  sapply(X, function(x) {
    new_dp = pmin(1-new_B$Buy, DP(x,mm[i],bb[i],aa[i]))
    new_eR = new_dp*new_B$Rev*margin - x
    c(i=i, x=x, eR.ALL=sum(new_eR), N=sum(new_eR>0), eR.SEL=sum(new_eR[new_eR > 0]) )
    }) %>% t %>% data.frame
  }))
# i : 工具編號
new_df %>%
  mutate_at(vars(eR.ALL, eR.SEL), function(y) round(y/1000)) %>%
  gather('key','value',-i,-x) %>%
  mutate(Instrument = paste0('I',i)) %>%
  ggplot(aes(x=x, y=value, col=Instrument)) +
  geom_hline(yintercept=0, linetype='dashed', col='blue') +
  geom_line(size=1.5,alpha=0.5) +
  xlab('工具選項(成本)') + ylab('預期收益($K)') +
  ggtitle('行銷工具優化','假設行銷工具的效果是其成本的函數') +
    facet_wrap(~key,ncol=1,scales='free_y') + theme_bw() -> new_p

plotly::ggplotly(new_p)
```



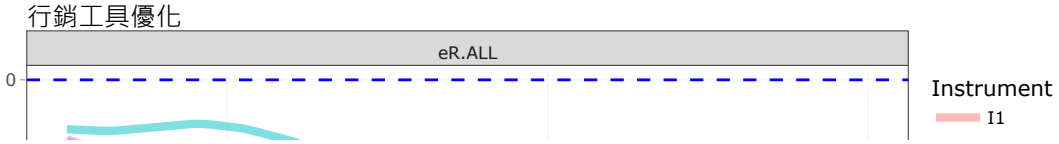


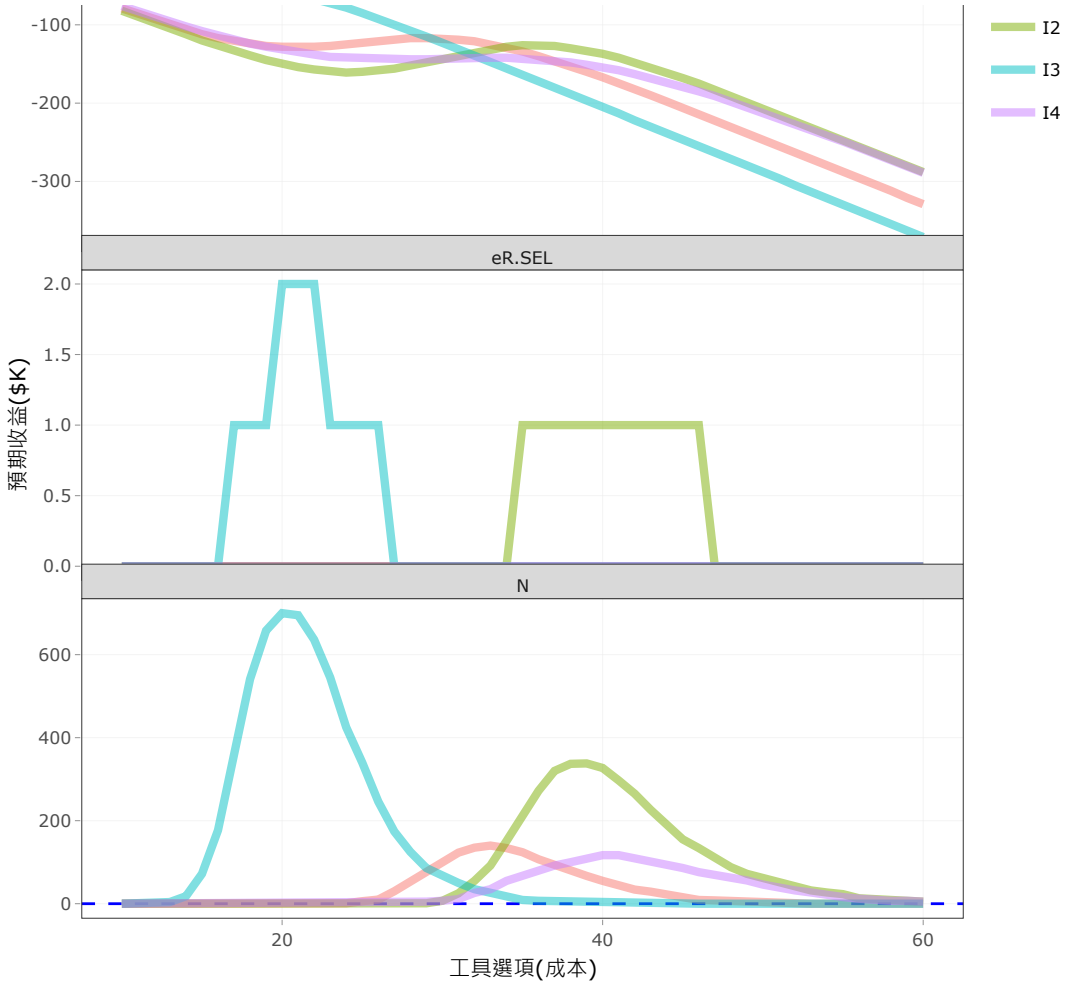

```
group_by(new_df, i) %>% top_n(1,eR.SEL)
```

```
## # A tibble: 4 × 5
## # Groups:   i [4]
##       i     x  eR.ALL     N  eR.SEL
##   <dbl> <dbl>   <dbl> <dbl>   <dbl>
## 1     1    37 134680.  2386 134723.
## 2     2    43 178626.  2392 178647.
## 3     3    24 110194.  2394 110200.
## 4     4    47 163753.  2382 163824.
```

Silence

```
X = seq(10, 60, 1)
silence_df = do.call(rbind, lapply(1:length(mm), function(i) {
  sapply(X, function(x) {
    silence_dp = pmin(1-silence_B$Buy, DP(x,mm[i],bb[i],aa[i]))
    silence_eR = silence_dp*silence_B$Rev*margin - x
    c(i=i, x=x, eR.ALL=sum(silence_eR), N=sum(silence_eR>0), eR.SEL=sum(silence_eR[silence_eR > 0]) )
    }) %>% t %>% data.frame
  }))
# i : 工具編號
silence_df %>%
  mutate_at(vars(eR.ALL, eR.SEL), function(y) round(y/1000)) %>%
  gather('key','value',-i,-x) %>%
  mutate(Instrument = paste0('I',i)) %>%
  ggplot(aes(x=x, y=value, col=Instrument)) +
  geom_hline(yintercept=0, linetype='dashed', col='blue') +
  geom_line(size=1.5,alpha=0.5) +
  xlab('工具選項(成本)') + ylab('預期收益($K)') +
  ggtitle('行銷工具優化','假設行銷工具的效果是其成本的函數') +
    facet_wrap(~key,ncol=1,scales='free_y') + theme_bw() -> silence_p

plotly::ggplotly(silence_p)
```



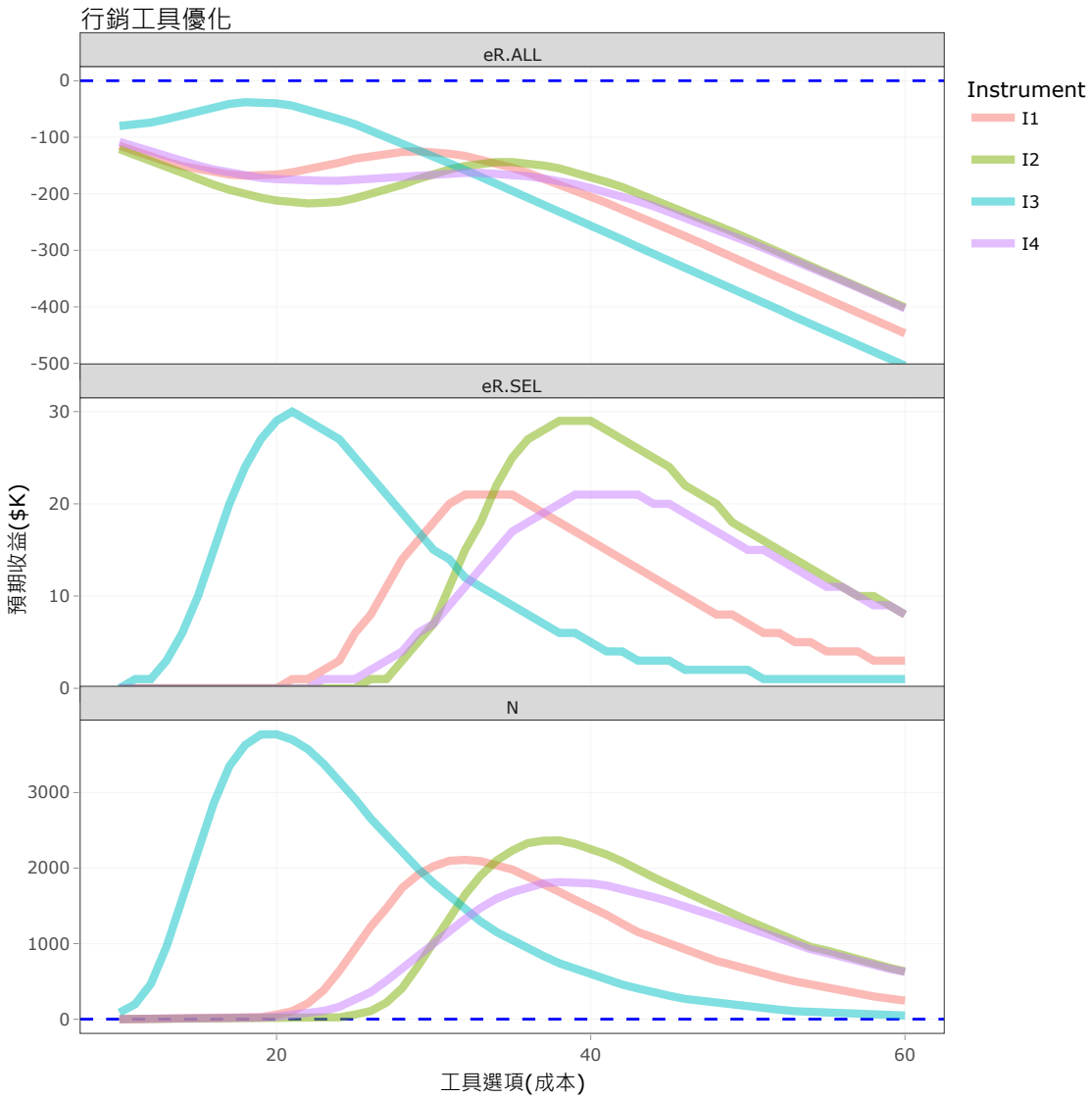



```
group_by(silence_df, i) %>% top_n(1,eR.SEL)
```

```
## # A tibble: 4 × 5
## # Groups:   i [4]
##       i     x   eR.ALL     N eR.SEL
##   <dbl> <dbl>    <dbl> <dbl>  <dbl>
## 1     1    33 -124863.   140   437.
## 2     2    39 -132998.   338  1271.
## 3     3    21  -62342.   695  1668.
## 4     4    41 -158395.   117   435.
```

Highp

```
X = seq(10, 60, 1)
highp_df = do.call(rbind, lapply(1:length(mm), function(i) {
  sapply(X, function(x) {
    highp_dp = pmin(1-highp_B$Buy, DP(x,mm[i],bb[i],aa[i]))
    highp_eR = highp_dp*highp_B$Rev*margin - x
    c(i=i, x=x, eR.ALL=sum(highp_eR), N=sum(highp_eR>0), eR.SEL=sum(highp_eR[highp_eR > 0]) )
    }) %>% t %>% data.frame
  }))
# i : 工具編號
highp_df %>%
  mutate_at(vars(eR.ALL, eR.SEL), function(y) round(y/1000)) %>%
  gather('key','value',-i,-x) %>%
  mutate(Instrument = paste0('I',i)) %>%
  ggplot(aes(x=x, y=value, col=Instrument)) +
  geom_hline(yintercept=0, linetype='dashed', col='blue') +
  geom_line(size=1.5,alpha=0.5) +
  xlab('工具選項(成本)') + ylab('預期收益($K)') +
  ggtitle('行銷工具優化','假設行銷工具的效果是其成本的函數') +
    facet_wrap(~key,ncol=1,scales='free_y') + theme_bw() -> highp_p

plotly::ggplotly(highp_p)
```

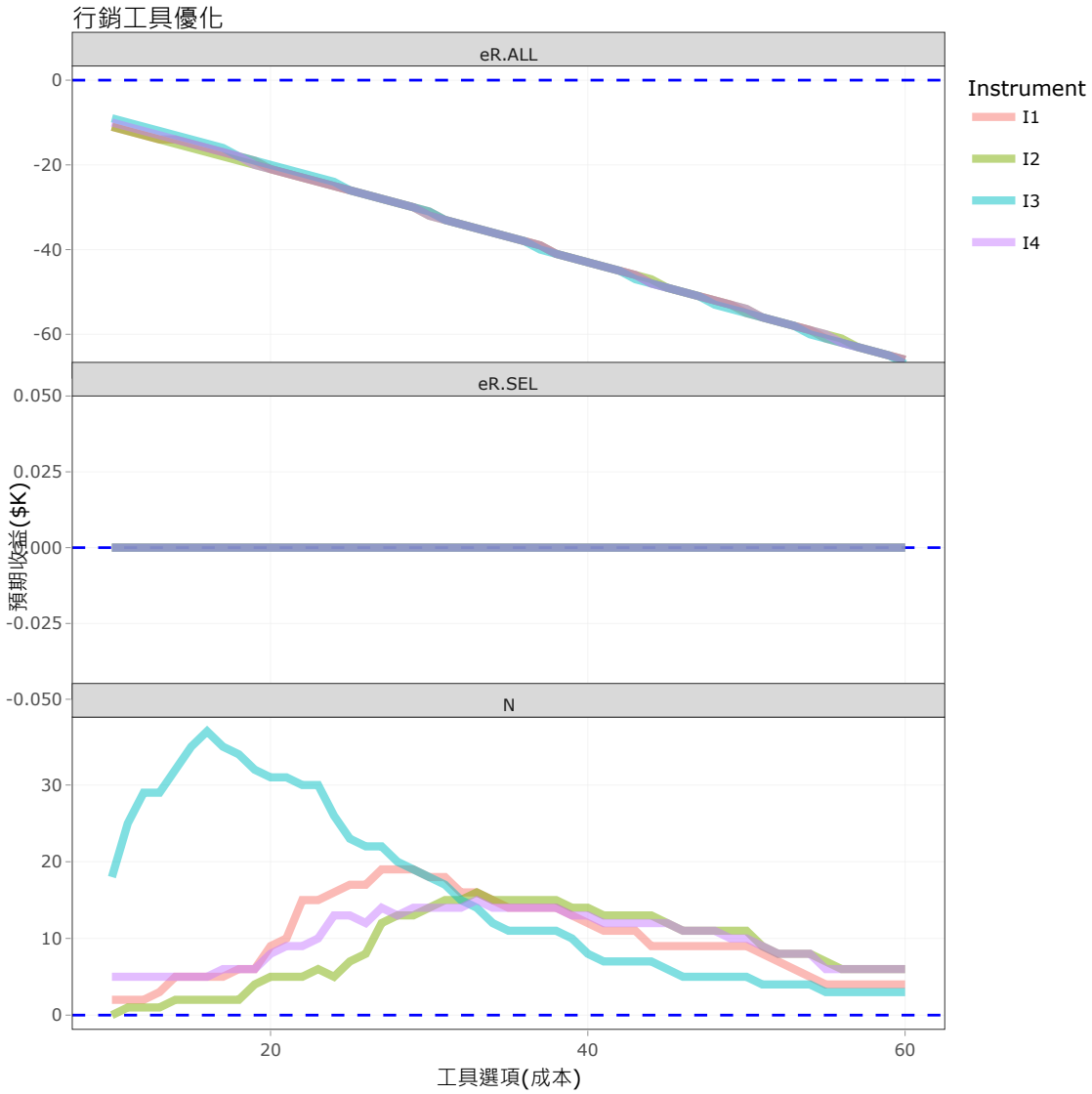



```
group_by(highp_df, i) %>% top_n(1,eR.SEL)
```

```
## # A tibble: 4 × 5
## # Groups:   i [4]
##       i     x   eR.ALL     N eR.SEL
##   <dbl> <dbl>    <dbl> <dbl>  <dbl>
## 1     1    33 -138991.  2092 21178.
## 2     2    39 -162055.  2321 29268.
## 3     3    21  -44321.  3700 29852.
## 4     4    41 -196792.  1770 21177.
```

Longterm

```
X = seq(10, 60, 1)
longterm_df = do.call(rbind, lapply(1:length(mm), function(i) {
  sapply(X, function(x) {
    longterm_dp = pmin(1-longterm_B$Buy, DP(x,mm[i],bb[i],aa[i]))
    longterm_eR = longterm_dp*longterm_B$Rev*margin - x
    c(i=i, x=x, eR.ALL=sum(longterm_eR), N=sum(longterm_eR>0), eR.SEL=sum(longterm_eR[longterm_eR > 0]) )
    }) %>% t %>% data.frame
  }))
# i : 工具編號
longterm_df %>%
  mutate_at(vars(eR.ALL, eR.SEL), function(y) round(y/1000)) %>%
  gather('key','value',-i,-x) %>%
  mutate(Instrument = paste0('I',i)) %>%
  ggplot(aes(x=x, y=value, col=Instrument)) +
  geom_hline(yintercept=0, linetype='dashed', col='blue') +
  geom_line(size=1.5,alpha=0.5) +
  xlab('工具選項(成本)') + ylab('預期收益($K)') +
  ggtitle('行銷工具優化','假設行銷工具的效果是其成本的函數') +
    facet_wrap(~key,ncol=1,scales='free_y') + theme_bw() -> longterm_p

plotly::ggplotly(longterm_p)
```




```
group_by(longterm_df, i) %>% top_n(1,eR.SEL)
```

```
## # A tibble: 4 × 5
## # Groups:   i [4]
##       i     x  eR.ALL     N eR.SEL
##   <dbl> <dbl>   <dbl> <dbl>  <dbl>
## 1     1    33 -34823.    16   315.
## 2     2    38 -40503.    15   361.
## 3     3    20 -19837.    31   435.
## 4     4    41 -44036.    12   309.
```

```
#在各組中找預期淨報酬最大的
```

🚴 從模擬的結果我們可以很容易看出每一個工具的：

- 最佳行銷強度
- 最佳行銷對象人數
- 最佳預期獲利

```
options(scipen=10)
pacman::p_load(latex2exp,Matrix,dplyr,tidyr,ggplot2,caTools,plotly,ggplot)
```

```
## Installing package into 'D:/R_Libs'
## (as 'lib' is unspecified)
```

```
## Warning: package 'ggplot' is not available for this version of R
##
## A version of this package for your version of R might be available elsewhere,
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages
```

```
## Warning: unable to access index for repository http://www.stats.ox.ac.uk/pub/RWin/bin/windows/contrib/4.3:
##   cannot open URL 'http://www.stats.ox.ac.uk/pub/RWin/bin/windows/contrib/4.3/PACKAGES'
```

```
## Warning in p_install(package, character.only = TRUE, ...):
```

```
## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'ggplot'
```

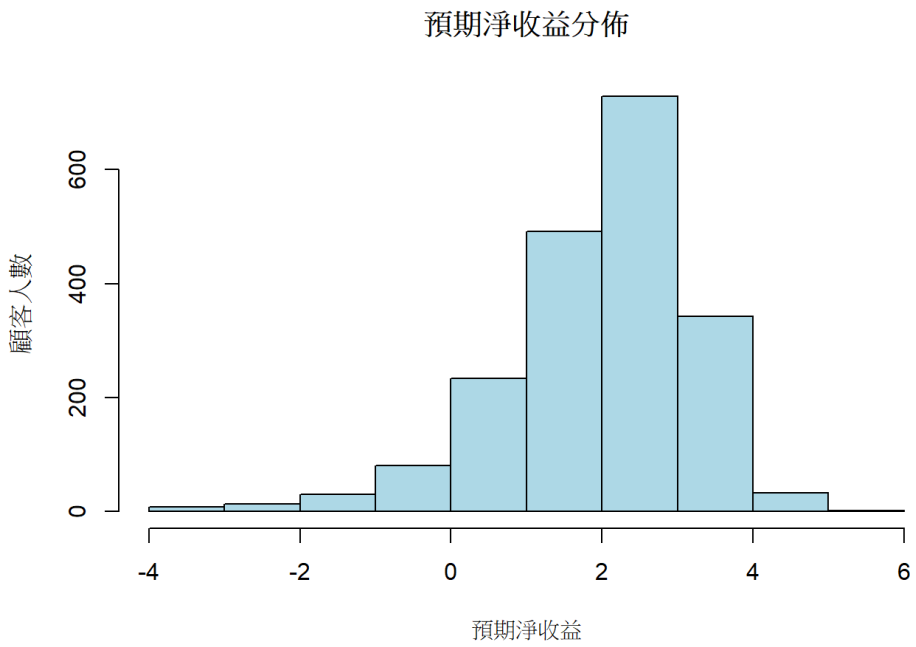
```
## Warning in pacman::p_load(latex2exp, Matrix, dplyr, tidyr, ggplot2, caTools, : Failed to install/load:
## ggplot
```

```
rm(list=ls(all=TRUE))
load("data/tf4.rdata")
```

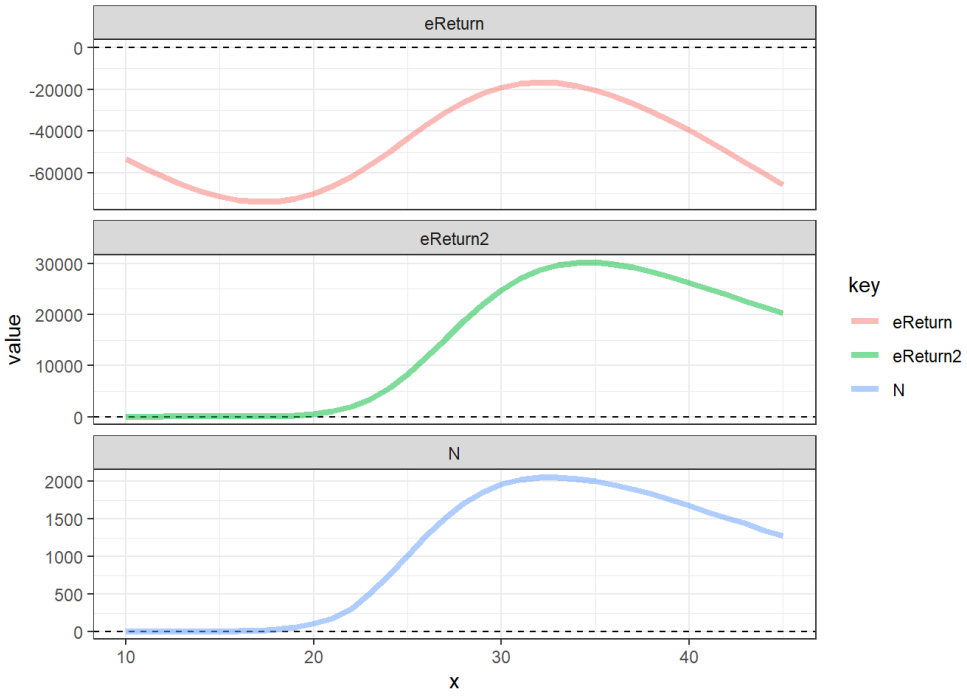
```
#A39 A44年齡兩群
a39df<-B[B$age=="a39",c("cust","r","s","f","m","rev","raw","age","area","Buy","Rev")]
a44df<-B[B$age=="a44",c("cust","r","s","f","m","rev","raw","age","area","Buy","Rev")]
```

```
DP = function(x,m0,b0,a0) {m0*plogis((10/a0)*(x-b0))}
margin = 0.17
m=0.2; b=25; a=40; x=30
a39df_dp = pmin(1-a39df$Buy, DP(x,m,b,a)) # 機率增幅不可以超過1，有些人本來的購買機率就已經95%，如果增加0.15會超過1，所以
         該購買者最多只能增加0.05，因此才是1-B$Buy, DP(x,m,b,a)，取最小
a39df_er = a39df_dp*a39df$Rev*margin - x
hist(log(a39df_er),main="預期淨收益分佈",xlab="預期淨收益",ylab="顧客人數",col = "lightblue")
```

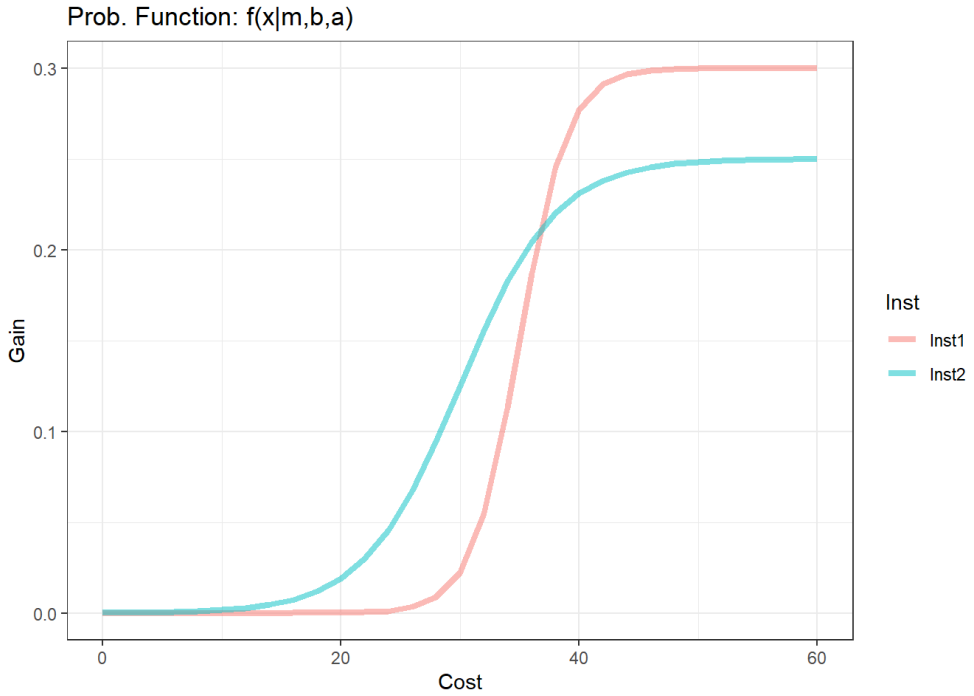
```
## Warning in log(a39df_er): NaNs produced
```




```
m=0.2; b=25; a=40; X = seq(10,45,1)
df = sapply(X, function(x) {
  dp = pmin(DP(x,m,b,a),1-a39df$Buy)
  eR = dp*a39df$Rev*margin - x
  c(x=x, eReturn=sum(eR),N=sum(eR > 0), eReturn2=sum(eR[eR > 0]))
  }) %>% t %>% data.frame

df %>% gather('key','value',-x)%>% ggplot(aes(x=x, y=value, col=key)) +
  geom_hline(yintercept=0,linetype='dashed') +
  geom_line(size=1.5,alpha=0.5) +
  facet_wrap(~key,ncol=1,scales='free_y') + theme_bw()
```

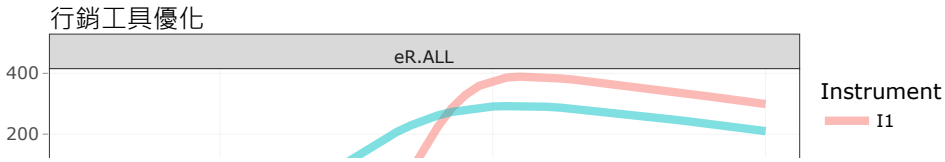


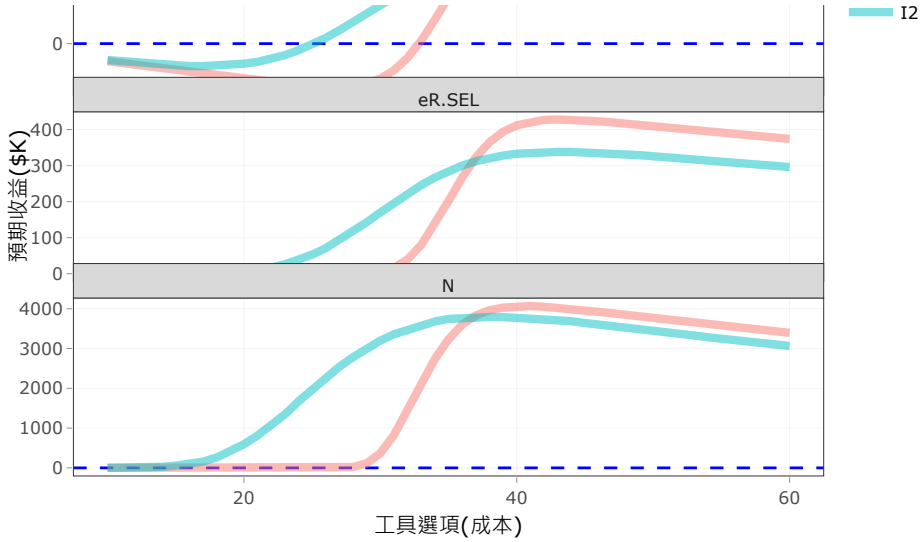
```
mm=c(0.3, 0.25)
bb=c(  35,   30)
aa=c(  20,   40)
X = seq(0,60,2)
do.call(rbind, lapply(1:length(mm), function(i) data.frame(
  Inst=paste0('Inst',i), Cost=X,
  Gain=DP(X,mm[i],bb[i],aa[i])
  ))) %>% data.frame %>%
  ggplot(aes(x=Cost, y=Gain, col=Inst)) +
  geom_line(size=1.5,alpha=0.5) + theme_bw() +
  ggtitle("Prob. Function: f(x|m,b,a)")
```



A39

```
X = seq(10, 60, 1)
a39df = do.call(rbind, lapply(1:length(mm), function(i) {
  sapply(X, function(x) {
    a39dp = pmin(1-a39df$Buy, DP(x,mm[i],bb[i],aa[i]))
    a39eR = a39dp*a39df$rev*margin - x
    c(i=i, x=x, eR.ALL=sum(a39eR), N=sum(a39eR>0), eR.SEL=sum(a39eR[a39eR > 0]) )
    }) %>% t %>% data.frame
  }))



a39df %>%
  mutate_at(vars(eR.ALL, eR.SEL), function(y) round(y/1000)) %>%
  gather('key','value',-i,-x) %>%
  mutate(Instrument = paste0('I',i)) %>%
  ggplot(aes(x=x, y=value, col=Instrument)) +
  geom_hline(yintercept=0, linetype='dashed', col='blue') +
  geom_line(size=1.5,alpha=0.5) +
  xlab('工具選項(成本)') + ylab('預期收益($K)') +
  ggtitle('行銷工具優化','假設行銷工具的效果是其成本的函數') +
    facet_wrap(~key,ncol=1,scales='free_y') + theme_bw() -> a39_p

plotly::ggplotly(a39_p)
```





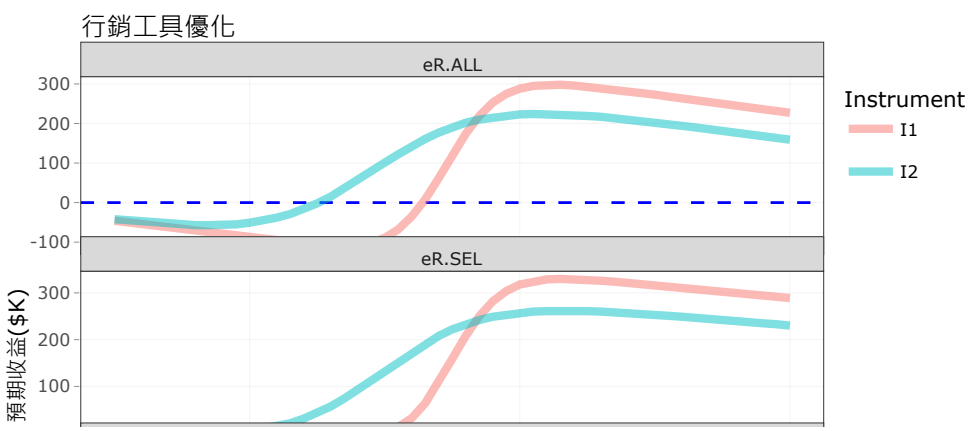

```
group_by(a39df, i) %>% top_n(1,eR.SEL)
```

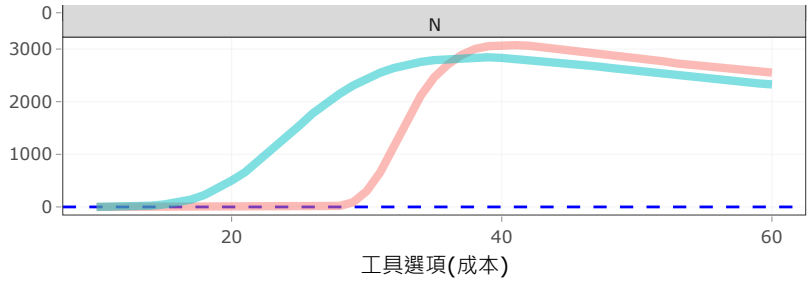
```
## # A tibble: 2 × 5
## # Groups:   i [2]
##       i     x  eR.ALL     N  eR.SEL
##   <dbl> <dbl>   <dbl> <dbl>   <dbl>
## 1     1    43 389212.  4028 428315.
## 2     2    43 291600.  3714 338234.
```

當成本為43時,預期的收益是42萬左右,人數約4028，m設不同時有不同的效果

A44

```
X = seq(10, 60, 1)
a44df = do.call(rbind, lapply(1:length(mm), function(i) {
  sapply(X, function(x) {
    a44dp = pmin(1-a44df$Buy, DP(x,mm[i],bb[i],aa[i]))
    a44eR = a44dp*a44df$rev*margin - x
    c(i=i, x=x, eR.ALL=sum(a44eR), N=sum(a44eR>0), eR.SEL=sum(a44eR[a44eR > 0]) )
    }) %>% t %>% data.frame
  }))


a44df %>%
  mutate_at(vars(eR.ALL, eR.SEL), function(y) round(y/1000)) %>%
  gather('key','value',-i,-x) %>%
  mutate(Instrument = paste0('I',i)) %>%
  ggplot(aes(x=x, y=value, col=Instrument)) +
  geom_hline(yintercept=0, linetype='dashed', col='blue') +
  geom_line(size=1.5,alpha=0.5) +
  xlab('工具選項(成本)') + ylab('預期收益($K)') +
  ggtitle('行銷工具優化','假設行銷工具的效果是其成本的函數') +
    facet_wrap(~key,ncol=1,scales='free_y') + theme_bw() -> a44_p

plotly::ggplotly(a44_p)
```






```
group_by(a44df, i) %>% top_n(1,eR.SEL)
```

```
## # A tibble: 2 × 5
## # Groups:   i [2]
##       i     x  eR.ALL     N  eR.SEL
##   <dbl> <dbl>   <dbl> <dbl>   <dbl>
## 1     1    43 297795.  3038 330126.
## 2     2    43 223365.  2762 261935.
```