CA_RMF

Philip Lin

2024-02-21

```
Tuango = read.csv('/Users/linpengyu/Desktop/Tuango_RFM.csv')
#Tuango
```

Q1

```
library(gmodels)
CrossTable(Tuango$buyer)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  13939
##
##
##              |         0 |         1 |
##              |-----------|-----------|
##              |     13507 |       432 |
##              |     0.969 |     0.031 |
##              |-----------|-----------|
##
##
```

```
##
##
```

#3.1% of customers responded after the push message. Q2

```
# Assuming you have already read the CSV into a data frame named Tuango

# Filter the data frame for buyers only
buyers <- subset(Tuango, buyer == 1)

# Calculate the average spending for buyers
average_spending_buyers <- mean(buyers$ordersize)

# Print the average spending
print(average_spending_buyers)
```

```
## [1] 202.3565
```

#Of those who bought, what was the average spending was 202.3565 RMB.

Q2_way2

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
Tuango %>%
  filter(buyer == 1) %>%
  summarise(average_spending = mean(ordersize))
```

```
##   average_spending
## 1         202.3565
```

Q3

```
library(dplyr)


# Assuming 'Tuango' is your dataframe
Tuango <- Tuango %>%
  mutate(
    rec_quin = ntile(recency, 5),
    freq_quin = ntile(frequency, 5),
    mon_quin = ntile(monetary, 5)
  )
head(Tuango %>% select(userid, buyer,recency,frequency,monetary,rec_quin,freq_quin,mon_quin))
```

```
##       userid buyer recency frequency monetary rec_quin freq_quin mon_quin
## 1 63775658     0     309         7     39.8        5         5         3
## 2 64880613     0     297         8     39.8        5         5         3
## 3 65051746     0     295         1     72.9        5         1         4
## 4 66689882     0     277         1     40.0        5         1         3
## 5 68839217     0     259         1     21.0        5         1         2
## 6 70630920     0     243         1     19.9        5         1         2
```

Q4avg_resp_rate_recency

```
#avg_rec
Tuango %>% group_by(rec_quin) %>% summarise(avg_rec = mean(recency), .groups = "drop")
```

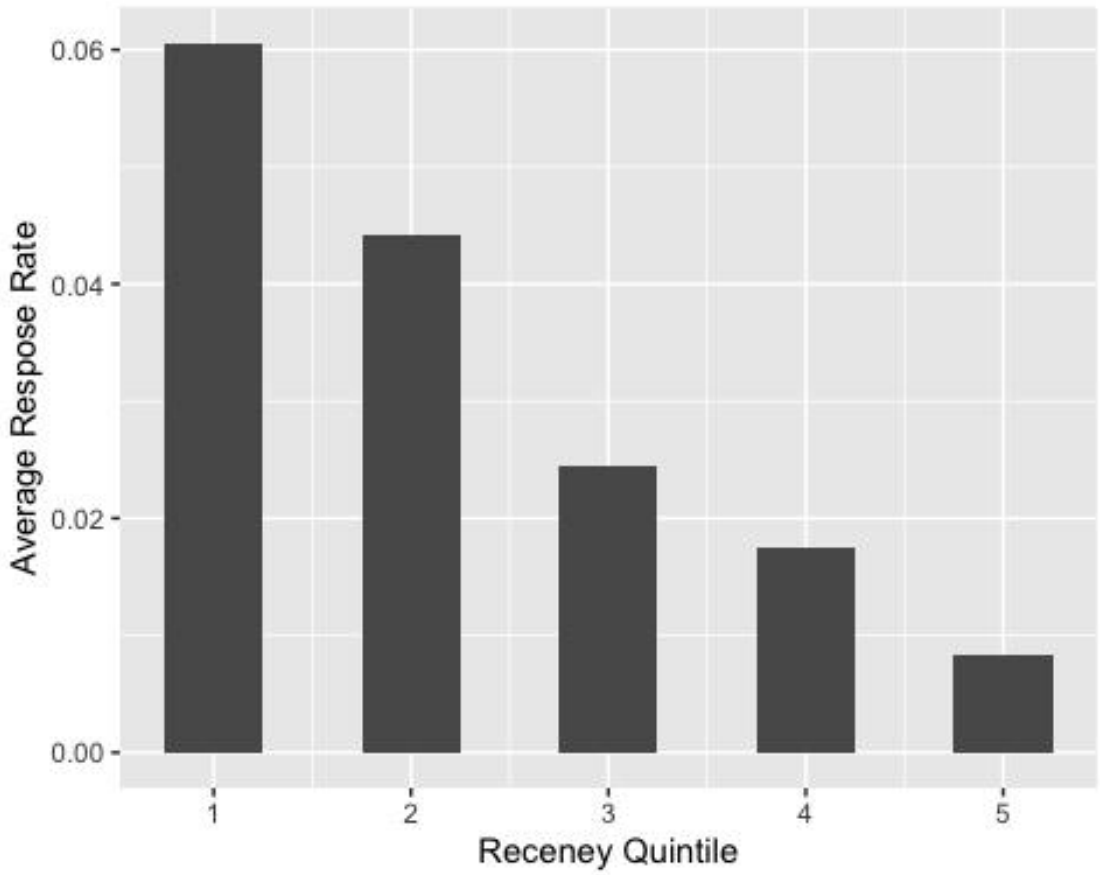
```
## # A tibble: 5 × 2
##   rec_quin avg_rec
##      <int>   <dbl>
## 1        1    9.12
## 2        2   11.5
## 3        3   21.6
## 4        4   50.5
## 5        5  183.
```

```
#avg_resp_rate_rec
avg_resp_rate_rec <- Tuango %>%
  group_by(rec_quin) %>%
  summarise(avg_resp_rate = mean(buyer), .groups = 'drop')
avg_resp_rate_rec
```

```
## # A tibble: 5 × 2
##   rec_quin avg_resp_rate
##      <int>         <dbl>
## 1        1        0.0606
## 2        2        0.0441
## 3        3        0.0244
## 4        4        0.0176
## 5        5        0.00825
```

```
#barplot
library(ggplot2)
bar_avg_resp_rate_rec <-
  ggplot(data = avg_resp_rate_rec,
         aes(x = rec_quin, y = avg_resp_rate)) +
  labs(x = "Receney Quintile", y = "Average Respose Rate") +
  geom_bar(stat = 'identity', width = 0.5)

bar_avg_resp_rate_rec
```



Q5avg_resp_rate_freq

```
#avg_freq
#Tuango %>% group_by(freq_quin) %>% summarise(avg_freq = mean(frequency),
 .groups = "drop")

#Flip indices
Tuango$freq_quin <- max(Tuango$freq_quin) + 1 - Tuango$freq_quin
Tuango %>% group_by(freq_quin) %>% summarise(avg_freq = mean(frequency),
 .groups = "drop")

## # A tibble: 5 × 2
##   freq_quin avg_freq
##       <dbl>    <dbl>
## 1         1     6.21
## 2         2     3.15
```

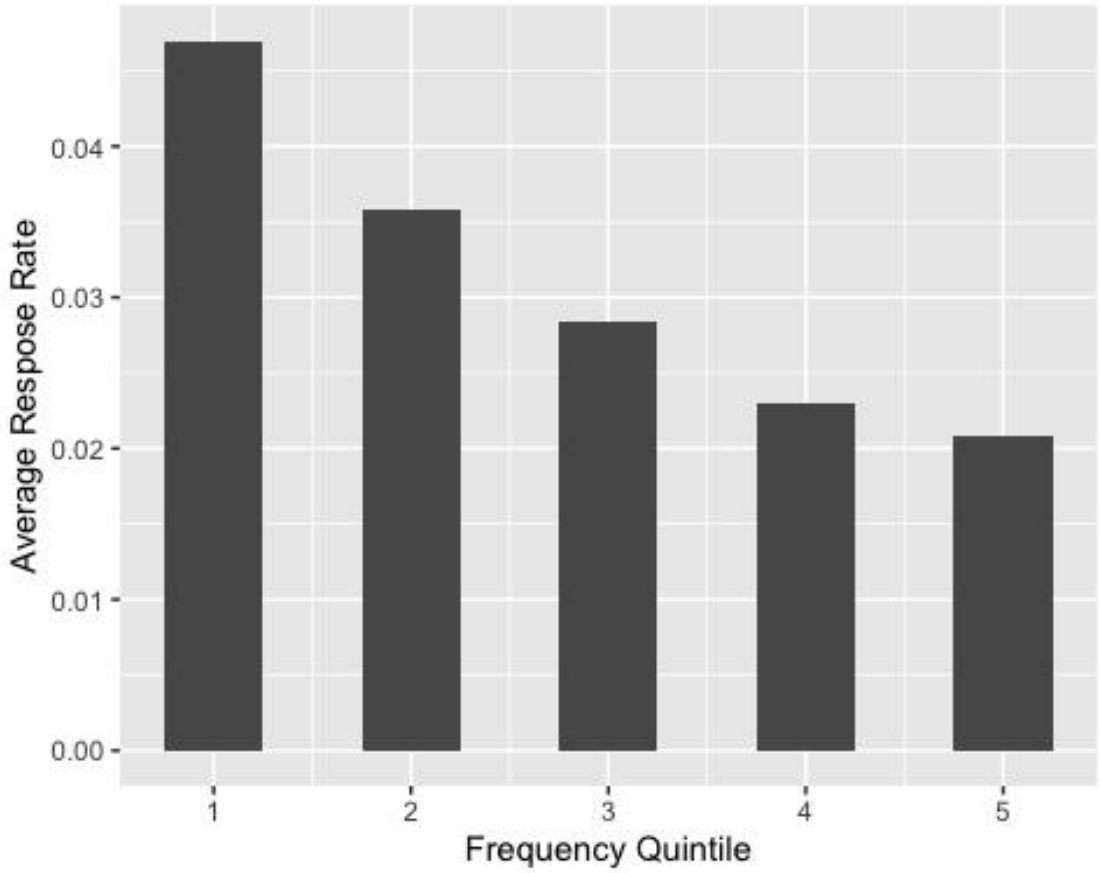
```
## 3           3       1.71
## 4           4       1
## 5           5       1

#avg_resp_rate_freq
avg_resp_rate_freq <- Tuango %>%
  group_by(freq_quin) %>%
  summarise(avg_resp_rate = mean(buyer), .groups = 'drop')
avg_resp_rate_freq

## # A tibble: 5 × 2
##   freq_quin avg_resp_rate
##       <dbl>         <dbl>
## 1         1        0.0470
## 2         2        0.0359
## 3         3        0.0283
## 4         4        0.0230
## 5         5        0.0208

#barplot
bar_avg_resp_rate_freq <-
  ggplot(data = avg_resp_rate_freq,
         aes(x = freq_quin, y = avg_resp_rate)) +
  labs(x = "Frequency Quintile", y = "Average Respose Rate") +
  geom_bar(stat = 'identity', width = 0.5)


bar_avg_resp_rate_freq
```



Q6avg_resp_rate_monetary

```
#Flip indices
Tuango$mon_quin <- max(Tuango$mon_quin) + 1 - Tuango$mon_quin
Tuango %>% group_by(mon_quin) %>% summarise(avg_mon = mean(monetary),
 .groups = "drop")#just checkout whether flip, does not have relationship wit fol
lowing calculation
```

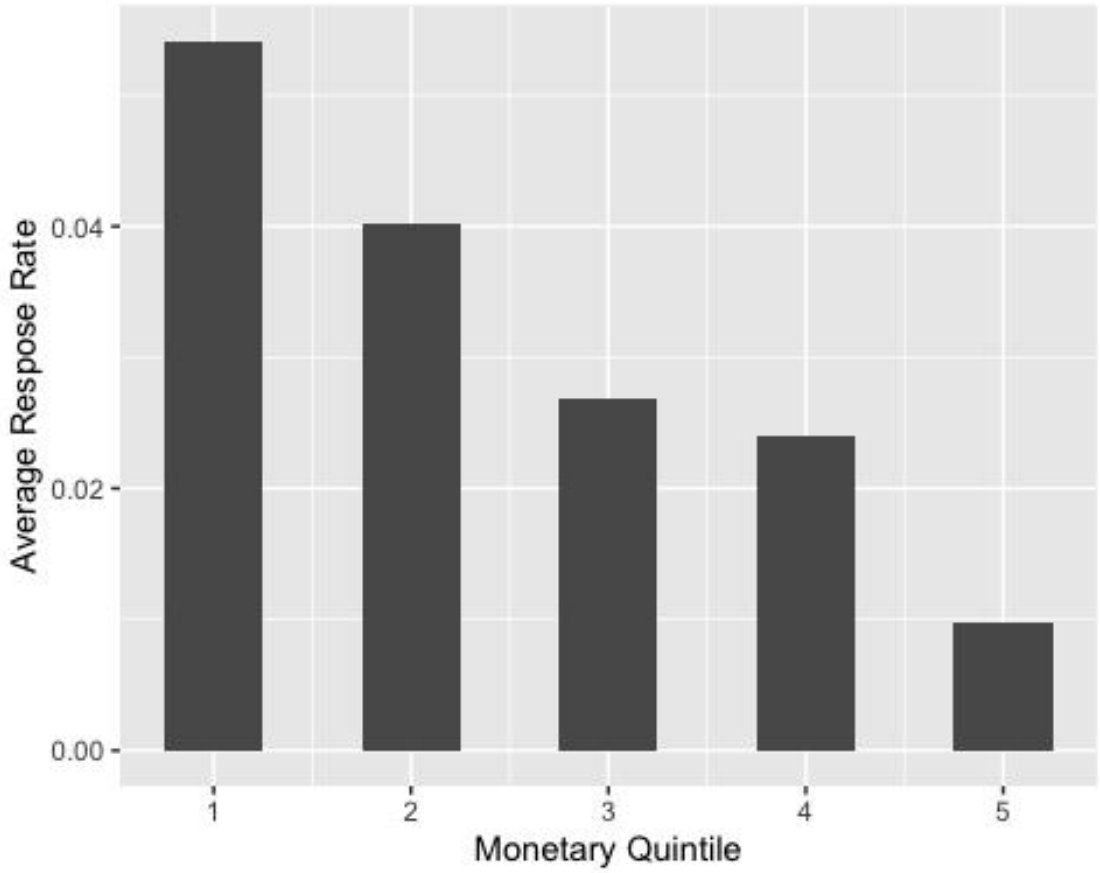
```
## # A tibble: 5 × 2
##   mon_quin avg_mon
##      <dbl>   <dbl>
## 1        1  219.
## 2        2   85.1
## 3        3   47.8
## 4        4   26.6
## 5        5    9.76
```

```r
#avg_resp_rate_mon
avg_resp_rate_mon <- Tuango %>%
  group_by(mon_quin) %>%
  summarise(avg_resp_rate = mean(buyer), .groups = 'drop')
avg_resp_rate_mon
```

```
## # A tibble: 5 × 2
##   mon_quin avg_resp_rate
##      <dbl>         <dbl>
## 1        1       0.0542
## 2        2       0.0402
## 3        3       0.0269
## 4        4       0.0240
## 5        5       0.00968
```

```r
#barplot
bar_avg_resp_rate_mon <-
  ggplot(data = avg_resp_rate_mon,
         aes(x = mon_quin, y = avg_resp_rate)) +
  labs(x = "Monetary Quintile", y = "Average Respose Rate") +
  geom_bar(stat = 'identity', width = 0.5)

bar_avg_resp_rate_mon
```



Q7avg_spd_rec

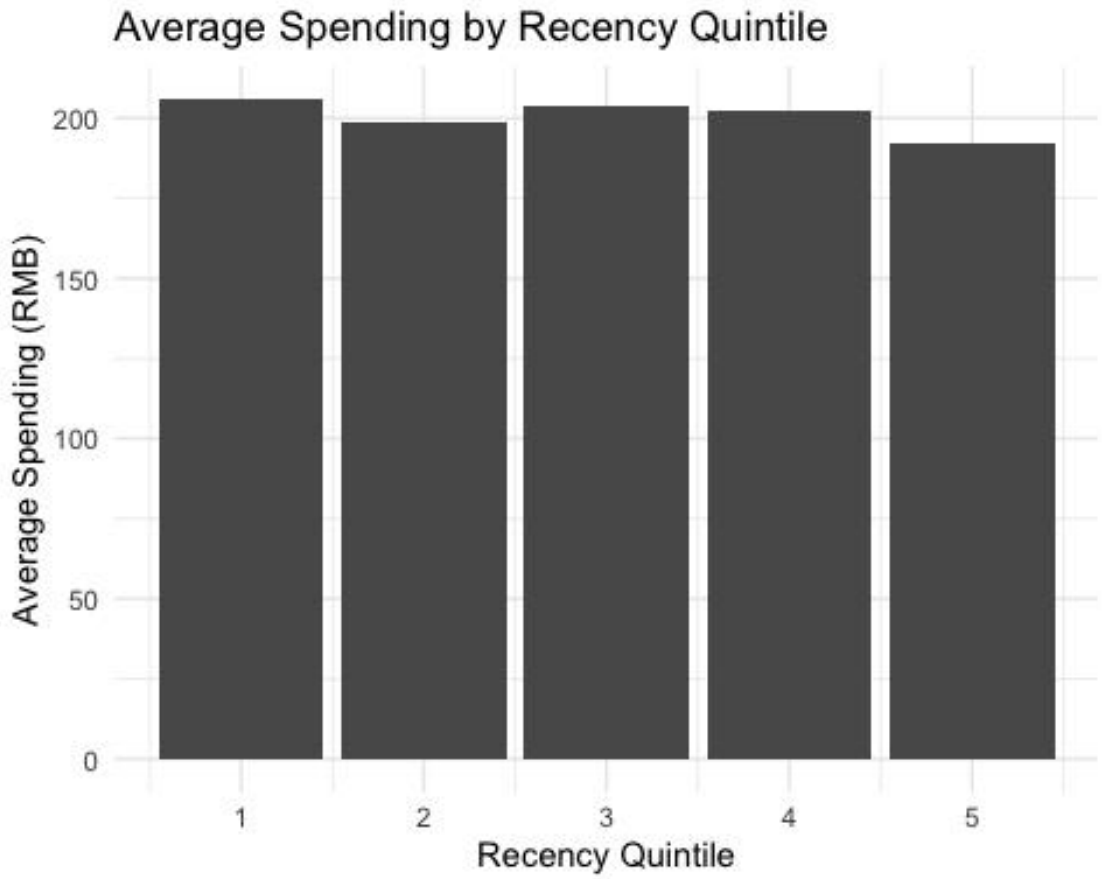
```
#avg_rec
Tuango %>% group_by(rec_quin) %>% summarise(avg_rec = mean(recency), .gr
oups = "drop")#just check whether flip

## # A tibble: 5 × 2
##   rec_quin avg_rec
##      <int>   <dbl>
## 1        1    9.12
## 2        2   11.5
## 3        3   21.6
## 4        4   50.5
## 5        5  183.
```

```r
# Calculate average order size for buyers by recency quintile
avg_spend_rec_quin <- Tuango %>%
  filter(buyer == 1) %>%
  group_by(rec_quin) %>%
  summarise(average_spending = mean(ordersize), .groups = 'drop')
avg_spend_rec_quin
```

```
## # A tibble: 5 × 2
##   rec_quin average_spending
##      <int>            <dbl>
## 1        1             206.
## 2        2             199.
## 3        3             204
## 4        4             202.
## 5        5             192.
```

```r
# Create bar chart for average spending by recency quintile
bar_avg_spend_rec <-
ggplot(avg_spend_rec_quin, aes(x = rec_quin, y = average_spending)) +
  geom_bar(stat = "identity") +
  labs(x = "Recency Quintile", y = "Average Spending (RMB)", title = "Average Spending by Recency Quintile") +
  theme_minimal()
bar_avg_spend_rec
```



Q7avg_spd_freq

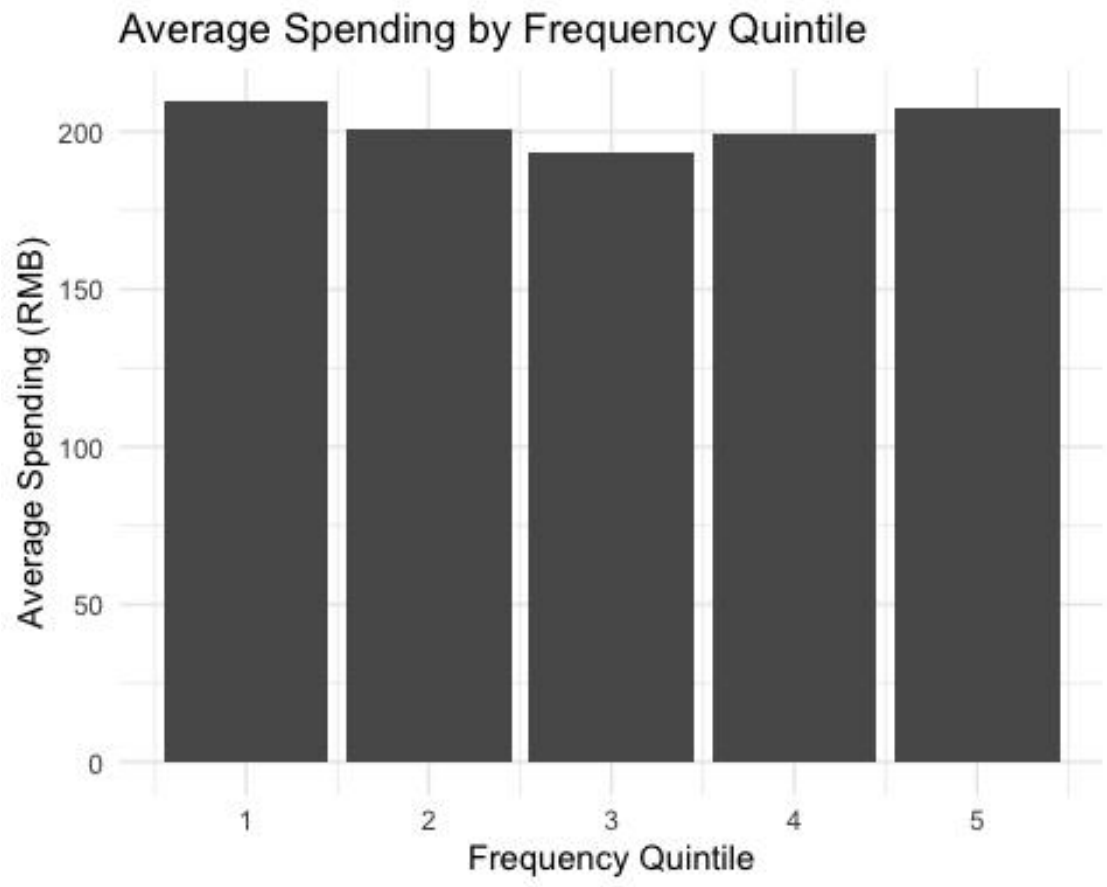
```
#Flip indices
Tuango$freq_quin <- max(Tuango$freq_quin) + 1 - Tuango$freq_quin

# Calculate average order size for buyers by frequency quintile
avg_spend_freq_quin <- Tuango %>%
  filter(buyer==1) %>%
  group_by(freq_quin) %>%
  summarise(average_spending = mean(ordersize), .groups = 'drop')
avg_spend_freq_quin

## # A tibble: 5 × 2
##   freq_quin average_spending
##       <dbl>            <dbl>
## 1         1             210.
```

```
## 2            2            201.
## 3            3            193.
## 4            4            199.
## 5            5            208.
```

```r
# Create bar chart for average spending by freq quintile
bar_avg_spend_freq <-
ggplot(avg_spend_freq_quin, aes(x = freq_quin, y = average_spending)) +
  geom_bar(stat = "identity") +
  labs(x = "Frequency Quintile", y = "Average Spending (RMB)", title = "Average Spending by Frequency Quintile") +
  theme_minimal()
bar_avg_spend_freq
```

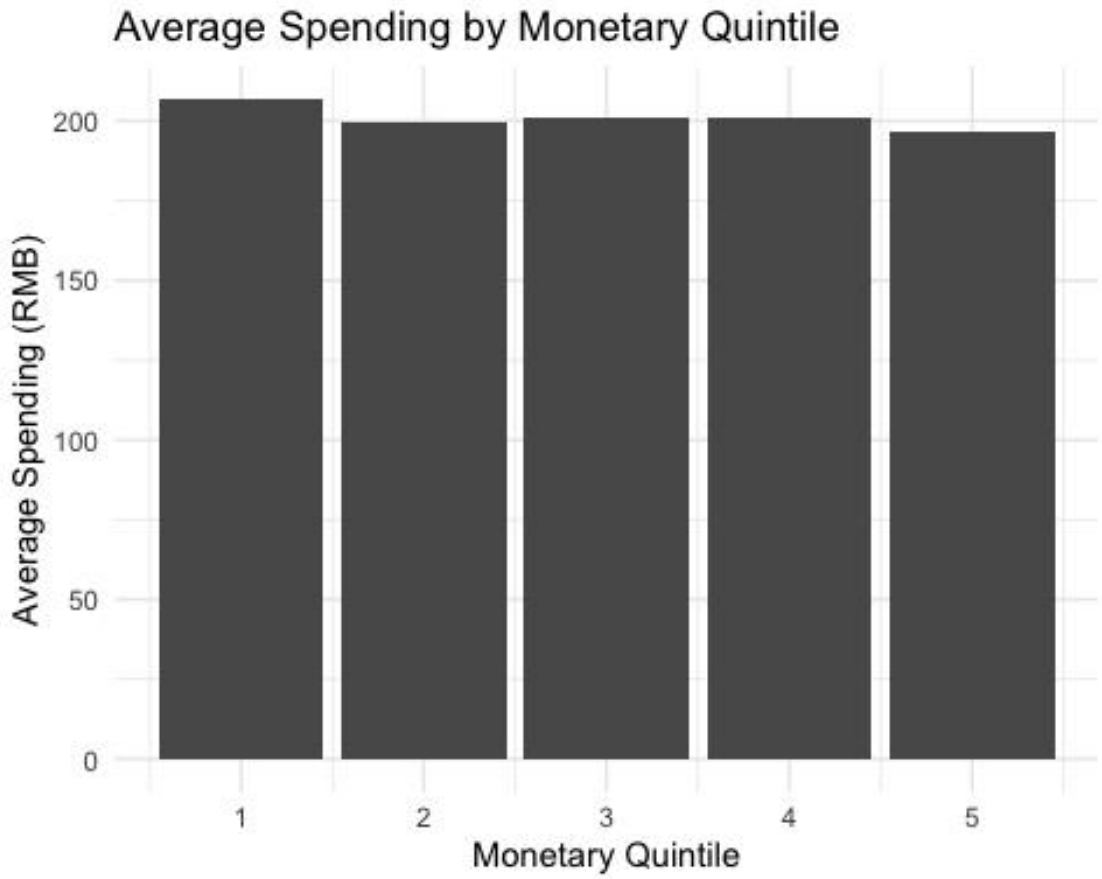



```r
#Flip indices
#Tuango$mon_quin <- max(Tuango$mon_quin) + 1 - Tuango$mon_quin
```

```r
# Calculate average order size for buyers by Monetary quintile
avg_spend_mon_quin <- Tuango %>%
  filter(buyer==1) %>%
  group_by(mon_quin) %>%
  summarise(average_spending = mean(ordersize), .groups = 'drop')
avg_spend_mon_quin
```

```
## # A tibble: 5 × 2
##   mon_quin average_spending
##      <dbl>            <dbl>
## 1        1             207.
## 2        2             199.
## 3        3             201
## 4        4             201.
## 5        5             197.
```

```r
# Create bar chart for average spending by Mon quintile
bar_avg_spend_mon <-
ggplot(avg_spend_mon_quin, aes(x = mon_quin, y = average_spending)) +
  geom_bar(stat = "identity") +
  labs(x = "Monetary Quintile", y = "Average Spending (RMB)", title = "Average Spending by Monetary Quintile") +
  theme_minimal()
bar_avg_spend_mon
```



Q8 #The outcomes indicate a slightly positive correlation between response and the recency, frequency, and average sizes of orders. Nevertheless, when analyzing solely customers who made purchases following push messages, the distribution of average spending remains relatively consistent across recency, frequency, and monetary quintiles. In contrast, considering all customers reveals significantly more variance. Q9

```
unit_cost <- 1.6

spend <- Tuango %>%
  filter(buyer==1) %>%
  summarize(avg_spend=mean(ordersize))

breakeven_response_rate = unit_cost/(0.5*spend)
```

```
breakeven_response_rate

##      avg_spend
## 1 0.01581368
```

#The breakeven response rate is 1.581368%.

Q11a

```
response <- Tuango %>%
  summarize(response = sum(buyer)/n())


number_of_buyers = 264841*response
number_of_buyers

##    response
## 1      8208

profit = number_of_buyers*(0.5*spend)-1.6*264841
profit

##    response
## 1 406725.4
```

#Profit is 406725.4 RMB. Q11b

```
market_expenditure = 1.6*264841
ROM = (profit/market_expenditure)*100
ROM

##    response
## 1 95.98339
```

#Return on marketing expenditures is 95.98339%.

Q12(use column'rfm1')

```
Tuango <- Tuango %>%
  group_by(rfm1) %>%
  mutate(avg_resp_rate_rfm=mean(buyer)) %>% ungroup()

Tuango %>%
  select(rfm1,avg_resp_rate_rfm,buyer)
```

```
## # A tibble: 13,939 × 3
##       rfm1 avg_resp_rate_rfm buyer
##      <int>             <dbl> <int>
##  1    514           0.0316        0
##  2    514           0.0316        0
##  3    553           0.00355       0
##  4    554           0             0
##  5    555           0.00380       0
##  6    555           0.00380       0
##  7    554           0             0
##  8    524           0             0
##  9    555           0.00380       0
## 10    553           0.00355       0
## #     13,929 more rows
```

```
Tuango <- Tuango %>%
  mutate(mailto = avg_resp_rate_rfm > breakeven_response_rate)

Tuango %>%
  select(rfm1,avg_resp_rate_rfm,buyer,mailto)
```

```
## # A tibble: 13,939 × 4
##       rfm1 avg_resp_rate_rfm buyer mailto["avg_spend"]
##      <int>             <dbl> <int> <lgl>
##  1    514           0.0316        0 TRUE
##  2    514           0.0316        0 TRUE
```

```
##  3    553              0.00355      0 TRUE
##  4    554              0            0 TRUE
##  5    555              0.00380      0 TRUE
##  6    555              0.00380      0 TRUE
##  7    554              0            0 TRUE
##  8    524              0            0 TRUE
##  9    555              0.00380      0 TRUE
## 10    553              0.00355      0 TRUE
## #     13,929 more rows
```

```
library(knitr)
Tuango %>%
  group_by(mailto) %>%
  summarise(n = n(), .groups = "drop") %>%
  mutate(Percentage = n/sum(n)*100) %>%
  kable(digits = 2)
```

mailto	n	Percentage
TRUE	13939	100

#Because if using RFM1, the effect does not change, therefore start using the previously calculated approach.

(use column calculated)

```
Tuango <- Tuango %>%
  mutate(rfmindex_iq = 100*rec_quin+ 10*freq_quin+ mon_quin)


head(Tuango %>% select(rec_quin, freq_quin, mon_quin,rfmindex_iq))
```

```
## # A tibble: 6 × 4
##    rec_quin freq_quin mon_quin rfmindex_iq
##       <int>     <dbl>    <dbl>       <dbl>
## 1         5         5        3         553
```

```
## 2          5          5          3          553
## 3          5          1          2          512
## 4          5          1          3          513
## 5          5          1          4          514
## 6          5          1          4          514
```

```
#Generate response rate specifically for each RFM cell.
avg_resp_rate_rfm  <- Tuango %>%
  group_by(rfmindex_iq) %>%
  summarise(resp_rate_rfm_iq = mean(buyer), .groups = "drop")


head(avg_resp_rate_rfm)
```

```
## # A tibble: 6 × 2
##    rfmindex_iq resp_rate_rfm_iq
##          <dbl>            <dbl>
## 1          111           0.0541
## 2          112           0.0769
## 3          113           0.0455
## 4          114           0.0404
## 5          115           0
## 6          121           0.0855
```

```
Tuango <- Tuango %>%
  group_by(rfmindex_iq) %>%
  mutate(resp_rate_rfm_iq = mean(buyer)) %>% ungroup()

Tuango <- Tuango %>%
  mutate(mailto_iq = resp_rate_rfm_iq > 0.01581368)


Tuango %>%
  select(rfmindex_iq, mailto_iq, buyer)
```

```
## # A tibble: 13,939 × 3
##    rfmindex_iq mailto_iq buyer
##          <dbl> <lgl>      <int>
##  1         553 FALSE          0
##  2         553 FALSE          0
##  3         512 FALSE          0
##  4         513 FALSE          0
##  5         514 FALSE          0
##  6         514 FALSE          0
##  7         513 FALSE          0
##  8         543 FALSE          0
##  9         514 FALSE          0
## 10         512 FALSE          0
## #    13,929 more rows
```

```
library(knitr)
Tuango %>%
  group_by(mailto_iq) %>%
  summarise(n = n(), .groups = "drop") %>%
  mutate(Percentage = n/sum(n)*100) %>%
  kable(digits = 2)
```

mailto_iq	n	Percentage
FALSE	5602	40.19
TRUE	8337	59.81

```
mailto_number = 264841*0.5981

marketing_expenditure = 1.6*mailto_number

mailto_number
```

```
## [1] 158401.4
```

```
marketing_expenditure
```

```
## [1] 253442.2
```

```
Tuango %>%
  filter(mailto_iq == TRUE) %>%
  group_by(buyer) %>%
  summarise(n = n(), .groups = "drop") %>%
  mutate(Percentage = n/sum(n)*100) %>%
  kable(digits = 2)
```

buyer	n	Percentage
0	7933	95.15
1	404	4.85

```
expected_buyer = mailto_number*0.0485
expected_buyer
```

```
## [1] 7682.468
```

```
profit = expected_buyer*0.5*spend$avg_spend - marketing_expenditure


ROM = profit/marketing_expenditure
ROM
```

```
## [1] 2.066965
```

```
#ROM = 206.6965%
```

Q12

```
Tuango <- Tuango %>% mutate(same_rfm = rfm1==rfm2)
CrossTable(Tuango$same_rfm)
```

```
##
##
##    Cell Contents
```

```
## |-------------------------|
## |                       N |
## |           N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  13939
##
##
##              |       FALSE |        TRUE |
##              |-------------|-------------|
##              |        1779 |       12160 |
##              |       0.128 |       0.872 |
##              |-------------|-------------|
##
##
##
##
```

#Approximately 87.2% of the dataset exhibits identical RFM (Recency, Frequency, Monetary) indices. #Utilizing a sequential RFM approach enhances the evenness of RFM group distribution; however, this method complicates the interpretability of the index, as the rankings for frequency and monetary values are contingent upon the recency rank. #Conversely, the independent RFM model boasts straightforward interpretability, though it risks yielding unpopulated categories due to the potential for uneven distribution.