Study3

Zhe GUAN

2024-07-11

Prepare, Process

```r
options(repos = "https://cran.rstudio.com/")
install.packages("data.table")
```

```
##
## The downloaded binary packages are in
##  /var/folders/pt/0nf2m3pj1f3b373wsyfc6glh0000gn/T//RtmppClujN/downloaded_packages
```

```r
install.packages("tidyverse")
```

```
## Warning in download.file(url, destfile, method, mode = "wb", ...): URL
## 'https://cran.rstudio.com/bin/macosx/big-sur-arm64/contrib/4.4/tidyverse_2.0.0.tgz':
## status was 'SSL connect error'

## Error in download.file(url, destfile, method, mode = "wb", ...) :
##   cannot open URL 'https://cran.rstudio.com/bin/macosx/big-sur-arm64/contrib/4.4/tidyverse_2.0.0.tgz

## Warning in download.packages(pkgs, destdir = tmpd, available = available, :
## download of package 'tidyverse' failed
```

```r
install.packages("dplyr")
```

```
## Warning in download.file(url, destfile, method, mode = "wb", ...): URL
## 'https://cran.rstudio.com/bin/macosx/big-sur-arm64/contrib/4.4/dplyr_1.1.4.tgz':
## status was 'SSL connect error'

## Error in download.file(url, destfile, method, mode = "wb", ...) :
##   cannot open URL 'https://cran.rstudio.com/bin/macosx/big-sur-arm64/contrib/4.4/dplyr_1.1.4.tgz'

## Warning in download.packages(pkgs, destdir = tmpd, available = available, :
## download of package 'dplyr' failed
```

```r
install.packages("ggmosaic")
```

```
##
## The downloaded binary packages are in
##  /var/folders/pt/0nf2m3pj1f3b373wsyfc6glh0000gn/T//RtmppClujN/downloaded_packages
```

```r
library(data.table)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::between()     masks data.table::between()
## x dplyr::filter()      masks stats::filter()
## x dplyr::first()       masks data.table::first()
## x lubridate::hour()    masks data.table::hour()
## x lubridate::isoweek() masks data.table::isoweek()
## x dplyr::lag()         masks stats::lag()
## x dplyr::last()        masks data.table::last()
## x lubridate::mday()    masks data.table::mday()
## x lubridate::minute()  masks data.table::minute()
## x lubridate::month()   masks data.table::month()
## x lubridate::quarter() masks data.table::quarter()
## x lubridate::second()  masks data.table::second()
## x purrr::transpose()   masks data.table::transpose()
## x lubridate::wday()    masks data.table::wday()
## x lubridate::week()    masks data.table::week()
## x lubridate::yday()    masks data.table::yday()
## x lubridate::year()    masks data.table::year()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(dplyr)
library(ggmosaic)
```

read data from link:

```r
filepath <- "/Users/zheguan/CWR_fig/QVI_case/"
transaction_data1 <- fread(paste(filepath,"QVI_transaction_data.csv",sep = ""))
customer_data1 <- fread(paste0(filepath,"QVI_purchase_behaviour.csv"))
str(customer_data1)
```

```
## Classes 'data.table' and 'data.frame':   72637 obs. of  3 variables:
##  $ LYLTY_CARD_NBR  : int  1000 1002 1003 1004 1005 1007 1009 1010 1011 1012 ...
##  $ LIFESTAGE       : chr  "YOUNG SINGLES/COUPLES" "YOUNG SINGLES/COUPLES" "YOUNG FAMILIES" "OLDER SI
##  $ PREMIUM_CUSTOMER: chr  "Premium" "Mainstream" "Budget" "Mainstream" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Primary key is LYLTY_CARD_NBR, then we can check transaction data:

```r
transaction_data1[!is.na(LYLTY_CARD_NBR),]
```

```
##          DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR
##         <int>     <int>          <int>  <int>    <int>
```

```
##      1: 43390         1           1000      1        5
##      2: 43599         1           1307    348       66
##      3: 43605         1           1343    383       61
##      4: 43329         2           2373    974       69
##      5: 43330         2           2426   1038      108
##     ---
## 264832: 43533       272         272319 270088       89
## 264833: 43325       272         272358 270154       74
## 264834: 43410       272         272379 270187       51
## 264835: 43461       272         272379 270188       42
## 264836: 43365       272         272380 270189       74
##                                        PROD_NAME PROD_QTY TOT_SALES
##                                           <char>    <int>     <num>
##      1:   Natural Chip        Compny SeaSalt175g        2       6.0
##      2:                 CCs Nacho Cheese    175g        3       6.3
##      3:   Smiths Crinkle Cut  Chips Chicken 170g        2       2.9
##      4:   Smiths Chip Thinly  S/Cream&Onion 175g        5      15.0
##      5: Kettle Tortilla ChpsHny&Jlpno Chili 150g        3      13.8
##     ---
## 264832:  Kettle Sweet Chilli And Sour Cream 175g        2      10.8
## 264833:            Tostitos Splash Of  Lime 175g        1       4.4
## 264834:                 Doritos Mexicana    170g        2       8.8
## 264835:  Doritos Corn Chip Mexican Jalapeno 150g        2       7.8
## 264836:            Tostitos Splash Of  Lime 175g        2       8.8
```

Select the shops from stores 77,86,88 in which we are intrested:

```r
selected_shops_transaction <- transaction_data1[STORE_NBR %in% c(77,86,88), ]
tibble(selected_shops_transaction)
```

```
## # A tibble: 3,974 x 8
##     DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR PROD_NAME   PROD_QTY TOT_SALES
##    <int>     <int>          <int>  <int>    <int> <chr>          <int>     <dbl>
##  1 43605        88          88320  87811      113 Twisties C~        2       9.2
##  2 43633        77          77069  74987       70 Tyrrells C~        2       8.4
##  3 43552        77          77000  74911       18 Cheetos Ch~        1       3.3
##  4 43568        77          77000  74912       69 Smiths Chi~        1       3
##  5 43298        77        2330211 236744       94 Burger Rin~        1       2.3
##  6 43527        77          77063  74977      112 Tyrrells C~        2       8.4
##  7 43516        77          77069  74985       98 NCC Sour C~        2       6
##  8 43532        77          77069  74986        8 Smiths Cri~        2       5.8
##  9 43524        77          77310  75254        9 Kettle Tor~        1       4.6
## 10 43375        77          77502  75463       94 Burger Rin~        2       4.6
## # i 3,964 more rows
```

left-link transaction data with the customer details:

```r
merged_data_selectedshops <- merge(selected_shops_transaction,customer_data1,by = "LYLTY_CARD_NBR",all.:
tibble(merged_data_selectedshops)
```

```
## # A tibble: 3,974 x 10
##    LYLTY_CARD_NBR  DATE STORE_NBR TXN_ID PROD_NBR PROD_NAME   PROD_QTY TOT_SALES
```

```
##              <int> <int>     <int>  <int>    <int> <chr>          <int>     <dbl>
##  1           77000 43552        77  74911       18 Cheetos Ch~        1       3.3
##  2           77000 43568        77  74912       69 Smiths Chi~        1       3
##  3           77000 43369        77  74910       36 Kettle Chi~        2      10.8
##  4           77001 43523        77  74913        7 Smiths Cri~        2      11.4
##  5           77001 43486        77  74914        9 Kettle Tor~        2       9.2
##  6           77002 43602        77  74915       63 Kettle 135~        1       4.2
##  7           77002 43634        77  74916      107 Smiths Cri~        2       5.2
##  8           77003 43542        77  74917       80 Natural Ch~        1       3
##  9           77004 43299        77  74918       30 Doritos Co~        1       4.4
## 10           77004 43307        77  74919       46 Kettle Ori~        2      10.8
## # i 3,964 more rows
## # i 2 more variables: LIFESTAGE <chr>, PREMIUM_CUSTOMER <chr>
```

the stakeholder asks us to focus on these measures:

consider the monthly sales experience of each store.

This can be broken down by:

1.total sales revenue 2.total number of customers 3.average number of transactions per customer

by checking, the date should be transformed with proper kind.

```
merged_data_selectedshops[,DATE := as.Date(DATE,origin = "1899/12/30")]
tibble(merged_data_selectedshops)
```

```
## # A tibble: 3,974 x 10
##    LYLTY_CARD_NBR DATE       STORE_NBR TXN_ID PROD_NBR PROD_NAME        PROD_QTY
##             <int> <date>         <int>  <int>    <int> <chr>               <int>
##  1          77000 2019-03-28        77  74911       18 Cheetos Chs & B~        1
##  2          77000 2019-04-13        77  74912       69 Smiths Chip Thi~        1
##  3          77000 2018-09-26        77  74910       36 Kettle Chilli 1~        2
##  4          77001 2019-02-27        77  74913        7 Smiths Crinkle ~        2
##  5          77001 2019-01-21        77  74914        9 Kettle Tortilla~        2
##  6          77002 2019-05-17        77  74915       63 Kettle 135g Swt~        1
##  7          77002 2019-06-18        77  74916      107 Smiths Crinkle ~        2
##  8          77003 2019-03-18        77  74917       80 Natural ChipCo ~        1
##  9          77004 2018-07-18        77  74918       30 Doritos Corn Ch~        1
## 10          77004 2018-07-26        77  74919       46 Kettle Original~        2
## # i 3,964 more rows
## # i 3 more variables: TOT_SALES <dbl>, LIFESTAGE <chr>, PREMIUM_CUSTOMER <chr>
```

1. check the total sales revenue per month among these stores:

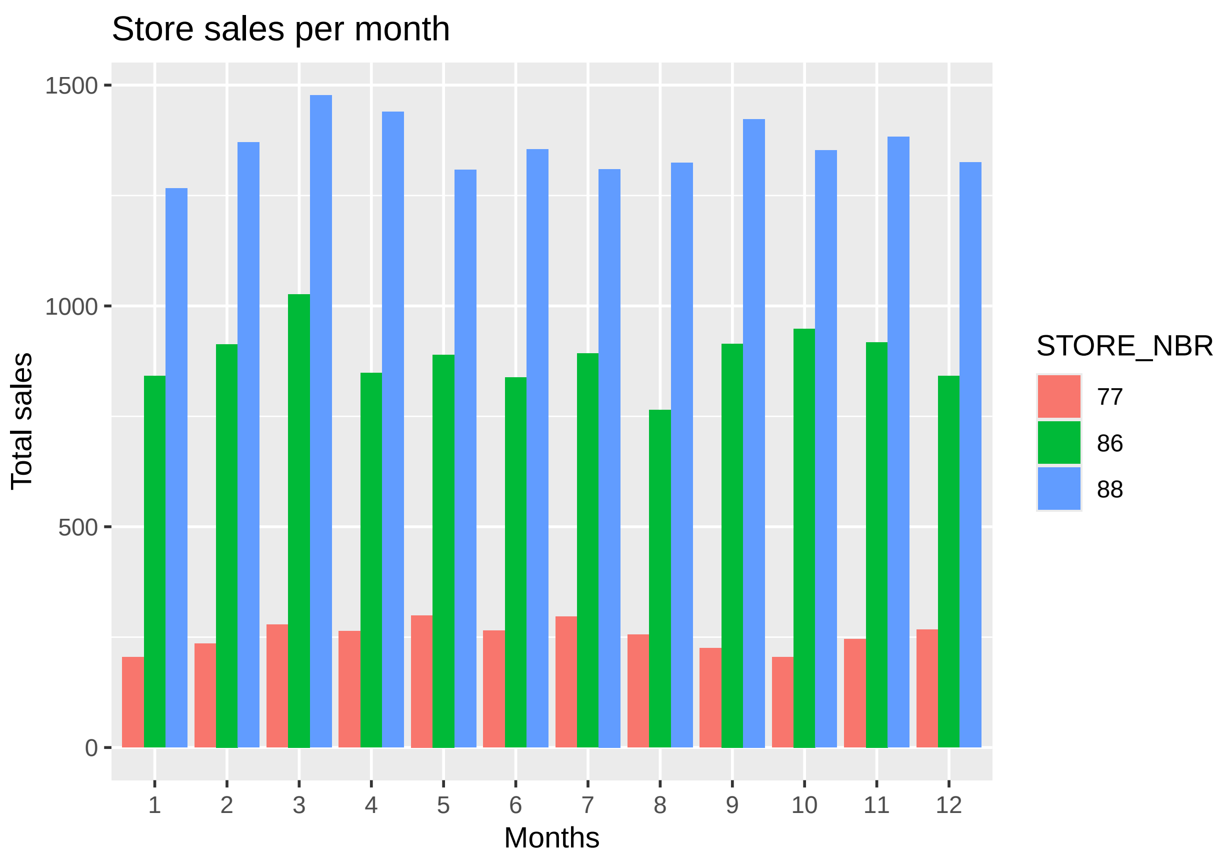
```
summary_table_1 <- merged_data_selectedshops[,.(month_income = sum(TOT_SALES)),.(month(DATE),STORE_NBR)]
tibble(summary_table_1)
```

```
## # A tibble: 36 x 3
##    month STORE_NBR month_income
##    <fct> <fct>            <dbl>
##  1 3     77                278.
##  2 4     77                263.
##  3 9     77                225.
```

```
##   4 2      77                  235
##   5 1      77                  204.
##   6 5      77                  299.
##   7 6      77                  265.
##   8 7      77                  297.
##   9 8      77                  256.
## 10 11     77                  245.
## # i 26 more rows
```

plot the contributions

```
ggplot(data = summary_table_1)+ geom_bar(position = position_dodge(), aes(x=month,fill= STORE_NBR,weigh
```



2. We count the total number of customers in separate stores, and only account the customers with unique loyality number.

```
summary_table_2 <- merged_data_selectedshops[,.(num_of_customer=uniqueN(LYLTY_CARD_NBR)),.(month(DATE),S
tibble(summary_table_2)
```

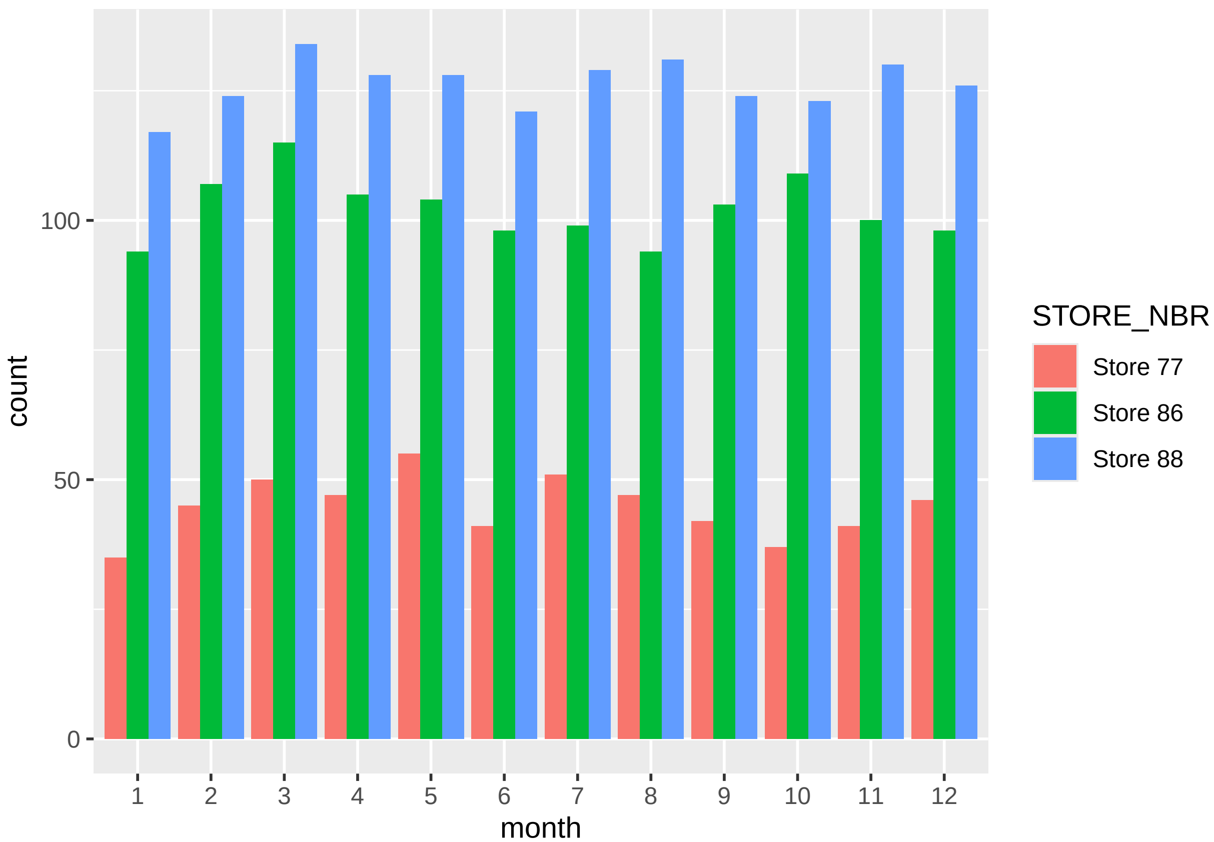
```
## # A tibble: 36 x 3
##    month STORE_NBR num_of_customer
##    <dbl>     <int>           <int>
##  1     3        77              50
##  2     4        77              47
```

```
## 3      9          77                 42
## 4      2          77                 45
## 5      1          77                 35
## 6      5          77                 55
## 7      6          77                 41
## 8      7          77                 51
## 9      8          77                 47
## 10    11          77                 41
## # i 26 more rows
```

```
summary_table_2[,STORE_NBR := as.factor(paste0("Store ",STORE_NBR))][,month := as.factor(month)]
```

plot1: bar plot

```
ggplot(data = summary_table_2) +geom_bar(position = position_dodge(), aes(x= month,weight = num_of_custo
```



plot2 mosaic:

```
p <- ggplot( data = summary_table_2) +geom_mosaic(aes(weight = num_of_customer, x = product(month,STORE_
tibble(ggplot_build(p)$data[[1]])
```

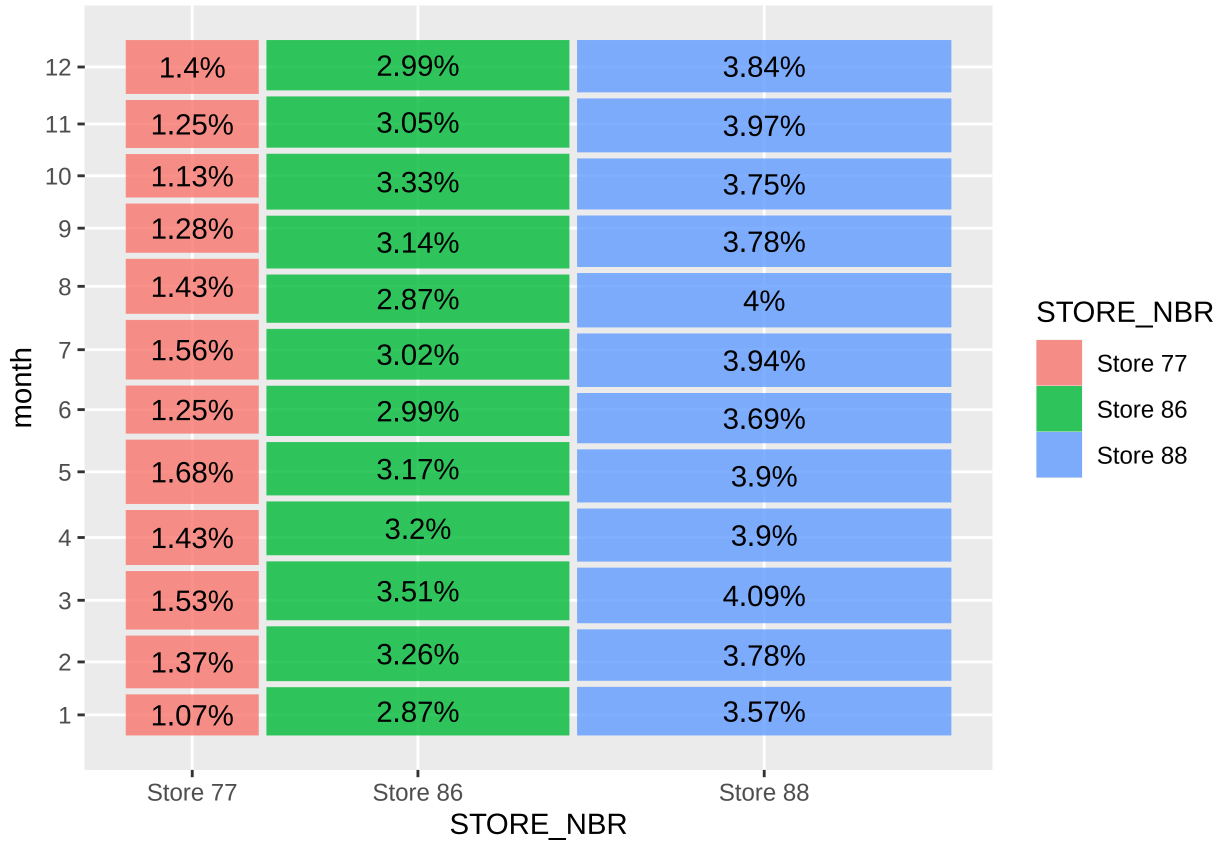
```
## Warning: The `scale_name` argument of `continuous_scale()` is deprecated as of ggplot2
## 3.5.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: The `trans` argument of `continuous_scale()` is deprecated as of ggplot2 3.5.0.
## i Please use the `transform` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: `unite_()` was deprecated in tidyr 1.2.0.
## i Please use `unite()` instead.
## i The deprecated feature was likely used in the ggmosaic package.
##   Please report the issue at <https://github.com/haleyjeppson/ggmosaic>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## # A tibble: 36 x 25
##    fill  x__fill__STORE_NBR   .wt  xmin  xmax   ymin   ymax level x__month    .n
##    <chr> <fct>              <dbl> <dbl> <dbl>  <dbl>  <dbl> <dbl> <fct>    <dbl>
##  1 #F87~ Store 77              35     0 0.161 0      0.0583     2 1           35
##  2 #F87~ Store 77              45     0 0.161 0.0679 0.143      2 2           45
##  3 #F87~ Store 77              50     0 0.161 0.152  0.236      2 3           50
##  4 #F87~ Store 77              47     0 0.161 0.245  0.324      2 4           47
##  5 #F87~ Store 77              55     0 0.161 0.333  0.425      2 5           55
##  6 #F87~ Store 77              41     0 0.161 0.434  0.503      2 6           41
##  7 #F87~ Store 77              51     0 0.161 0.512  0.597      2 7           51
##  8 #F87~ Store 77              47     0 0.161 0.607  0.685      2 8           47
##  9 #F87~ Store 77              42     0 0.161 0.695  0.765      2 9           42
## 10 #F87~ Store 77              37     0 0.161 0.774  0.836      2 10          37
## # i 26 more rows
## # i 15 more variables: label <chr>, x <dbl>, y <dbl>, group <dbl>, PANEL <fct>,
## #   width <dbl>, linetype <chr>, fontsize <dbl>, shape <dbl>, colour <lgl>,
## #   size <dbl>, alpha <dbl>, stroke <dbl>, linewidth <dbl>, weight <dbl>
```

```
p + geom_text(data = ggplot_build(p)$data[[1]],aes(x=xmin/2+xmax/2,y=ymin/2+ymax/2,label = paste0(round
```



plot3:

```
install.packages("viridis")
```

```
##
## The downloaded binary packages are in
##  /var/folders/pt/0nf2m3pj1f3b373wsyfc6glh0000gn/T//RtmppClujN/downloaded_packages
```

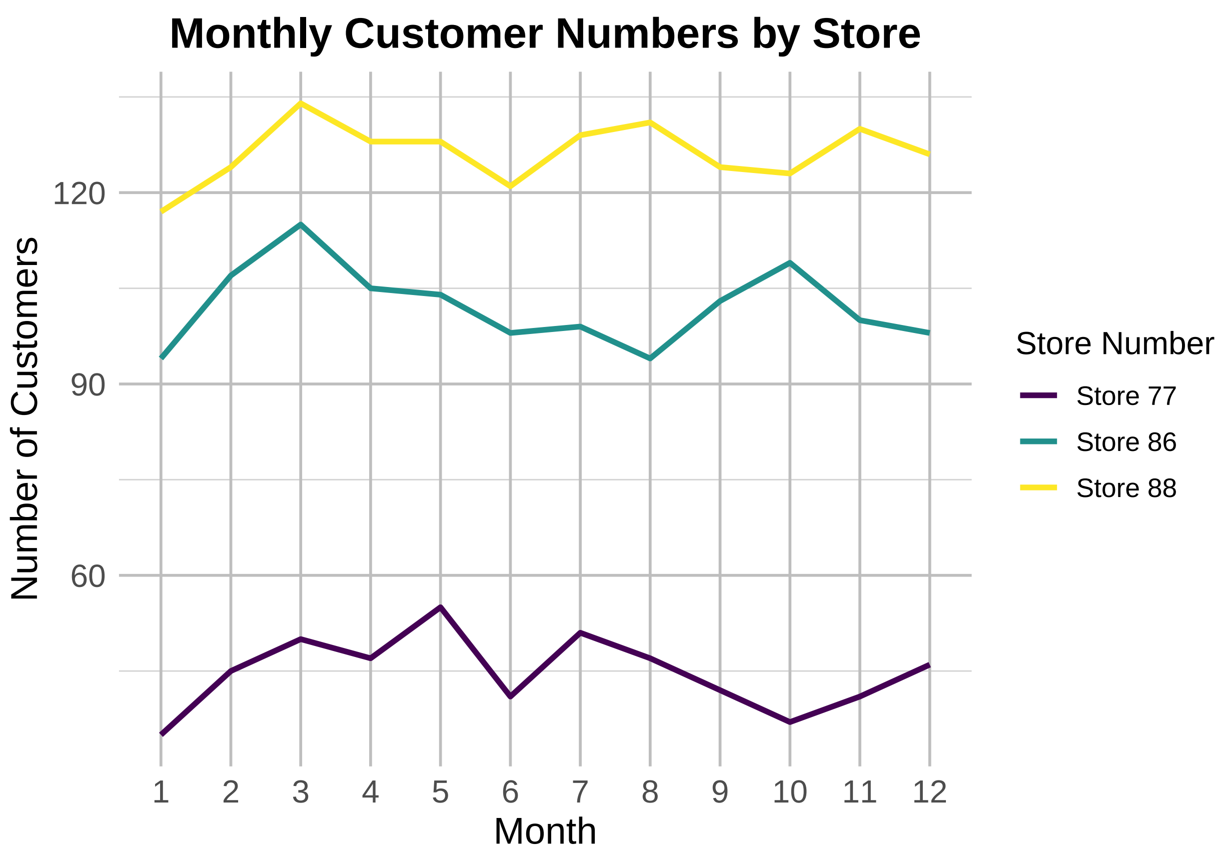
```
library(viridis)
```

```
## Loading required package: viridisLite
```

```
ggplot(data = summary_table_2) +
  geom_line(aes(x = month, y = num_of_customer, color = factor(STORE_NBR), group = STORE_NBR), size = 1
  scale_color_viridis(discrete = TRUE) + #
  labs(x = "Month", y = "Number of Customers", title = "Monthly Customer Numbers by Store", color = "Sto
  theme_minimal() + #
  theme(
    plot.title = element_text(hjust = 0.5, size = 16, face = "bold"), #
    axis.title.x = element_text(size = 14),
    axis.title.y = element_text(size = 14),
    axis.text = element_text(size = 12),
    legend.title = element_text(size = 12),
    legend.text = element_text(size = 10),
    panel.grid.major = element_line(color = "gray", size = 0.5),
```

```
    panel.grid.minor = element_line(color = "lightgray", size = 0.25)
  )
```

```
## 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.

## Warning: The `size` argument of `element_line()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



3. calculate the average number of transactions of customer:

```
summary_table_3 <- merged_data_selectedshops[,.(total_num = sum(PROD_QTY)),.(month(DATE),STORE_NBR)]
tibble(summary_table_3)
```

```
## # A tibble: 36 x 3
##    month STORE_NBR total_num
##    <dbl>     <int>     <int>
```

```
## 1      3         77         82
## 2      4         77         78
## 3      9         77         70
## 4      2         77         74
## 5      1         77         65
## 6      5         77         84
## 7      6         77         70
## 8      7         77         84
## 9      8         77         74
## 10    11         77         67
## # i 26 more rows
```

```r
summary_table_3[,STORE_NBR := as.factor(paste0("Store ",STORE_NBR))][,month := as.factor(month)]
```

merged table:

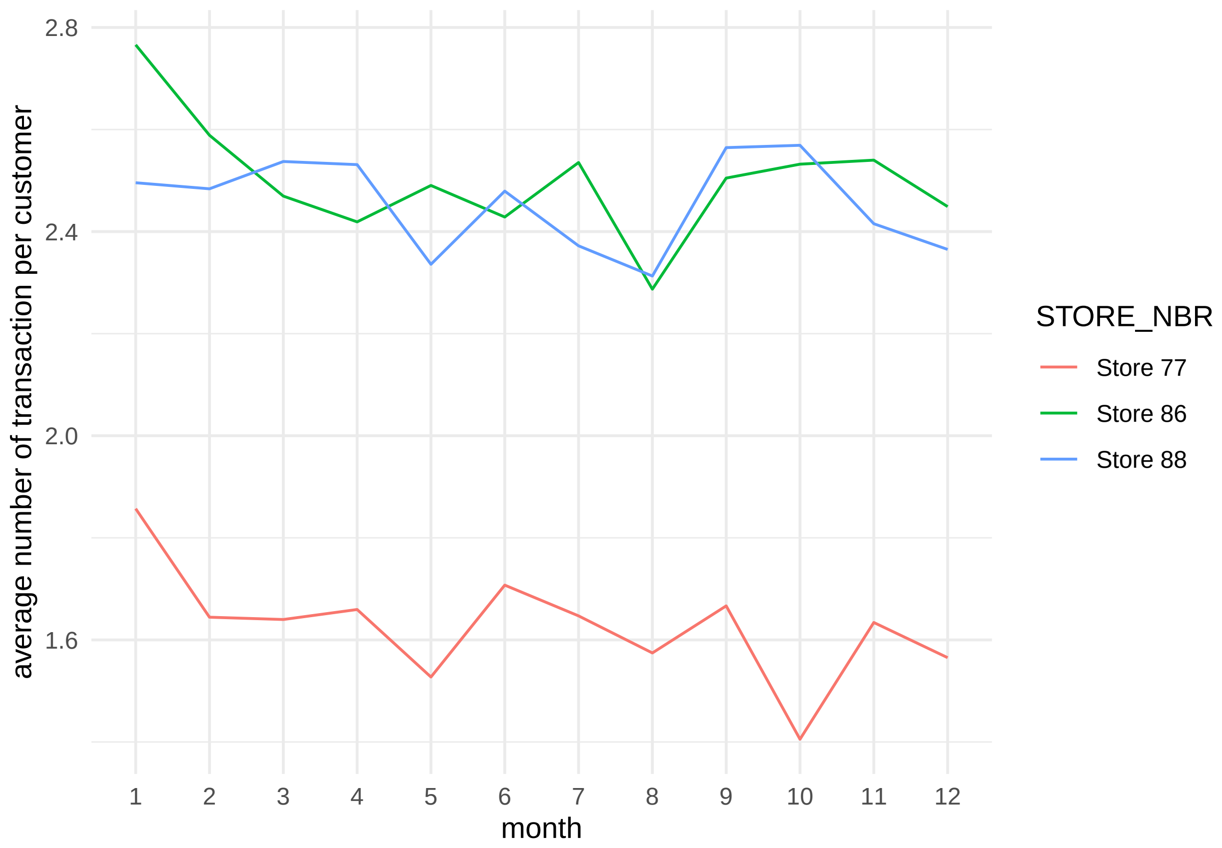
```r
summary_table_3_link_2 <- merge(summary_table_3,summary_table_2,by = c("STORE_NBR","month"),all.x = TRUE
tibble(summary_table_3_link_2)
```

```
## # A tibble: 36 x 4
##    STORE_NBR month total_num num_of_customer
##    <fct>     <fct>     <int>           <int>
##  1 Store 77  1            65              35
##  2 Store 77  2            74              45
##  3 Store 77  3            82              50
##  4 Store 77  4            78              47
##  5 Store 77  5            84              55
##  6 Store 77  6            70              41
##  7 Store 77  7            84              51
##  8 Store 77  8            74              47
##  9 Store 77  9            70              42
## 10 Store 77  10           52              37
## # i 26 more rows
```

```r
summary_table_3_link_2[,avg_num := total_num/num_of_customer]
```

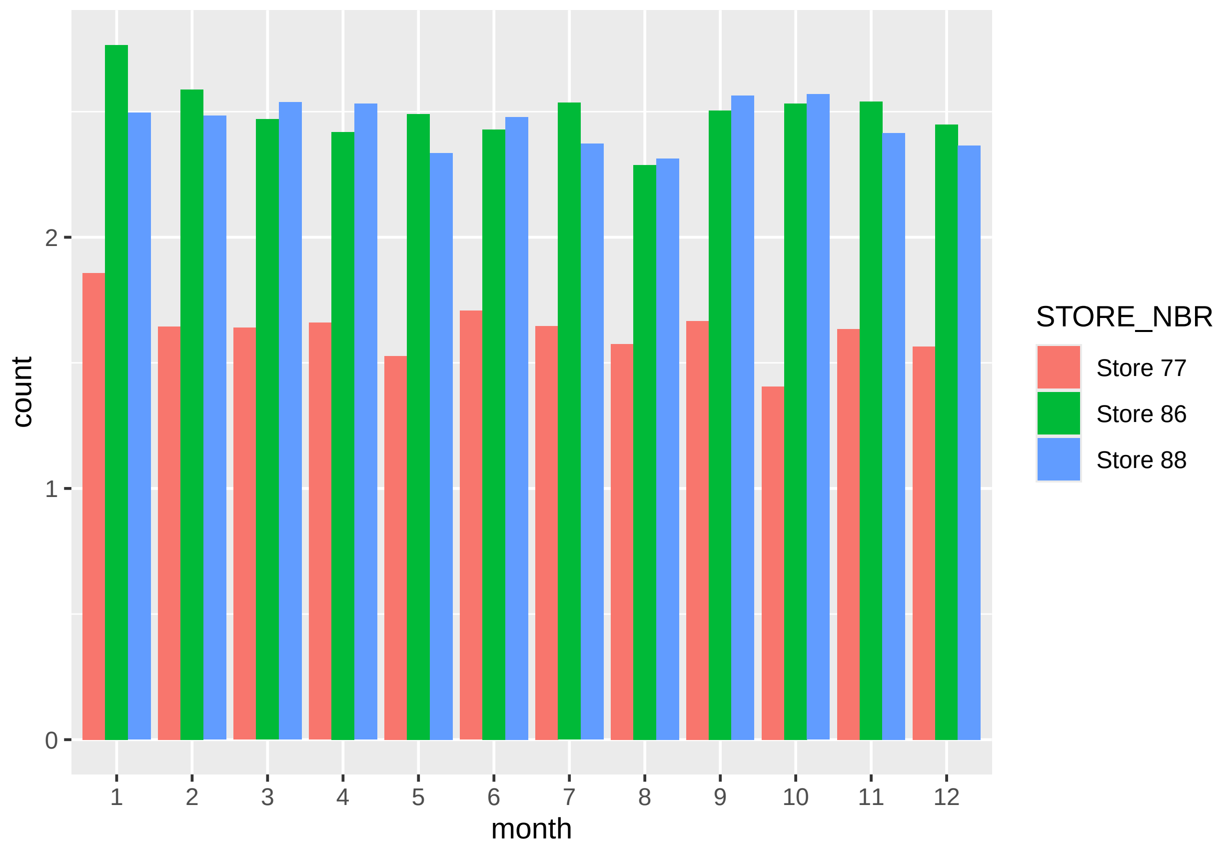
plot:

```r
ggplot(data = summary_table_3_link_2) +geom_line(aes(x= month, y = avg_num, group = STORE_NBR, colour =
```



bar plot:

```
ggplot(data = summary_table_3_link_2) + geom_bar(position = position_dodge(), aes(x=month,weight = avg_n
```



Next important step is to choose the Control Stores, here we define a function to calculate the correlations between trial stores and other stores.

firstly, we need to similary calculate a summary table:

```
transaction_data1[,DATE := as.Date(DATE,origin = "1899/12/30")]
measureOverTime <- transaction_data1[,.(totalSales = sum(TOT_SALES),nCustomers= uniqueN(LYLTY_CARD_NBR)
tibble(measureOverTime)
```

```
## # A tibble: 3,169 x 6
##    month STORE_NBR totalSales nCustomers nChipsPerTxn avgPricePerUnit
##    <dbl>     <int>      <dbl>      <int>        <int>           <dbl>
##  1     1         1       155.         35           42            3.69
##  2     1         2       163.         43           49            3.32
##  3     1         3      1052.        102          236            4.46
##  4     1         4      1525         134          335            4.55
##  5     1         5       838          92          236            3.55
##  6     1         7       975.        102          221            4.41
##  7     1        10       879.        102          252            3.49
##  8     1        15       874.         99          248            3.52
##  9     1        17       365.         39           83            4.39
## 10     1        19       992         111          224            4.43
## # i 3,159 more rows
```

Analyze

define a function which can calculate the correlation between two stores by using the information in separate 12 months.

```
testN <- measureOverTime[,N := .N ,.(STORE_NBR)][order(N)]

filtered_stores <- measureOverTime[N==12,]
filtered_stores
```

```
##       month STORE_NBR totalSales nCustomers nChipsPerTxn avgPricePerUnit     N
##       <num>     <int>      <num>      <int>        <int>           <num> <int>
##    1:     1         1      154.8         35           42        3.685714    12
##    2:     1         2      162.8         43           49        3.322449    12
##    3:     1         3     1051.7        102          236        4.456356    12
##    4:     1         4     1525.0        134          335        4.552239    12
##    5:     1         5      838.0         92          236        3.550847    12
##   ---
## 3116:    12       224       20.5          5            8        2.562500    12
## 3117:    12       177        5.3          2            2        2.650000    12
## 3118:    12       204       33.6          5            9        3.733333    12
## 3119:    12       159       11.0          3            4        2.750000    12
## 3120:    12       244        3.0          1            1        3.000000    12
```

```
tibble(testN)
```

```
## # A tibble: 3,169 x 7
##    month STORE_NBR totalSales nCustomers nChipsPerTxn avgPricePerUnit     N
##    <dbl>     <int>      <dbl>      <int>        <int>           <dbl> <int>
##  1     3        92        9.2          1            2            4.6      1
##  2     7        85       13.9          3            5            2.78     1
##  3    10        76        6            1            2            3        1
##  4     3       252        3.7          1            1            3.7      2
##  5     4       211        2.6          1            1            2.6      2
##  6     4       206        4.6          1            1            4.6      2
##  7     7       211        2.6          1            1            2.6      2
##  8     7       206        3            1            1            3        2
##  9     8       252        3.7          1            1            3.7      2
## 10     9        31        6            1            2            3        2
## # i 3,159 more rows
```

```
comparison_vector1 <- measureOverTime[STORE_NBR == 1]$totalSales
comparison_vector2 <- measureOverTime[STORE_NBR == 77]$totalSales
unique(measureOverTime$STORE_NBR)
```

```
##   [1]   1   2   3   4   5   7  10  15  17  19  20  22  23  24  25  26  27  28
##  [19]  29  33  34  36  37  38  39  40  41  43  45  46  47  48  49  52  53  54
##  [37]  55  56  57  58  59  61  65  69  71  72  75  78  80  81  84  86  90  91
##  [55]  94  95  96  97 100 101 102 105 106 107 108 109 110 111 112 113 114 116
##  [73] 118 119 120 121 122 123 124 125 126 127 128 129 130 131 133 136 138 142
##  [91] 144 145 148 150 151 152 153 154 155 156 157 160 162 164 165 166 168 169
## [109] 173 175 178 181 182 183 184 185 186 187 188 189 190 191 194 196 199 200
```

```
## [127] 201 203 205 207 208 210 212 213 214 216 217 218 219 223 225 226 227 229
## [145] 230 231 232 233 234 235 236 237 238 241 243 247 249 250 253 255 256 257
## [163] 259 260 261 262 264 265 266 267 269 270 272   6   8   9  12  13  16  18
## [181]  21  30  32  35  50  60  62  63  64  66  67  68  73  74  79  82  83  87
## [199]  88  89  93  99 103 104 115 134 135 137 141 143 146 147 149 158 163 170
## [217] 172 174 179 180 195 197 202 209 215 220 221 222 228 239 240 242 245 246
## [235] 248 251 254 271  14  51  77  98 132 159 171 244 268  70 176 177 192  42
## [253] 167 204 224  44 140 258 139 263 161 198 193 117  92 252 211 206  85  31
## [271]  76  11
```

```r
cor(comparison_vector1, comparison_vector2, method = "kendall")
```

```
## [1] 0.2424242
```

```r
tibble(filtered_stores)
```

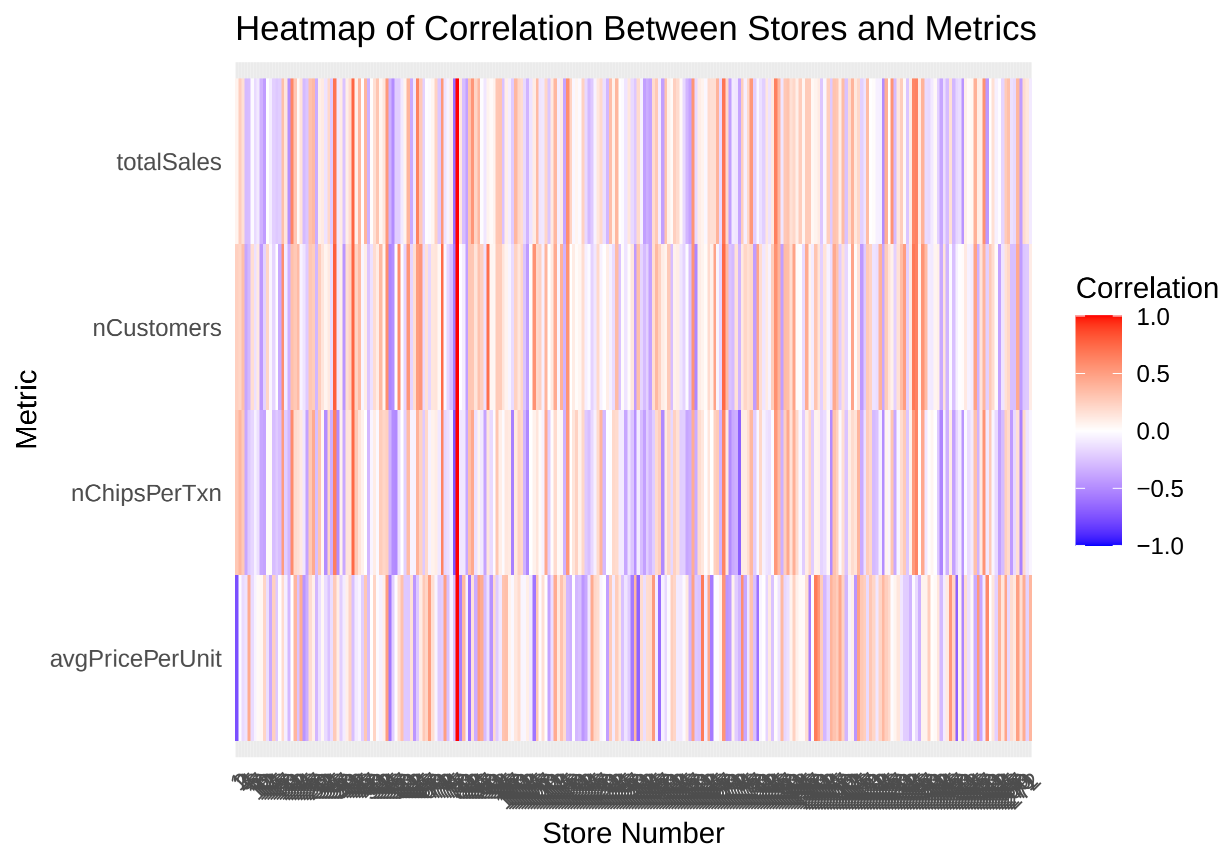
```
## # A tibble: 3,120 x 7
##    month STORE_NBR totalSales nCustomers nChipsPerTxn avgPricePerUnit     N
##    <dbl>     <int>      <dbl>      <int>        <int>           <dbl> <int>
##  1     1         1       155.         35           42            3.69    12
##  2     1         2       163.         43           49            3.32    12
##  3     1         3      1052.        102          236            4.46    12
##  4     1         4      1525         134          335            4.55    12
##  5     1         5       838          92          236            3.55    12
##  6     1         7       975.        102          221            4.41    12
##  7     1        10       879.        102          252            3.49    12
##  8     1        15       874.         99          248            3.52    12
##  9     1        17       365.         39           83            4.39    12
## 10     1        19       992         111          224            4.43    12
## # i 3,110 more rows
```

```r
calculate_cor1 <- function(store1,metrics,method){
  #calculate the total number of stores
  unique_num <- unique(filtered_stores$STORE_NBR)
  result <- numeric(length = length(unique_num))

  for (i in seq_along(unique_num)){
    trail_vector <- filtered_stores[STORE_NBR == store1,..metrics]
    comparison_vector <- filtered_stores[STORE_NBR == unique_num[i],..metrics]
    result[i] <- cor(trail_vector,comparison_vector,method = method)
  }
  result_table <- data.table(STORE_NBR = unique_num,Correlation = result)
  return(result_table)
}

calculate_cor <- function(store1, metrics, method) {
  if (!exists("filtered_stores") || !all(c("STORE_NBR", metrics) %in% colnames(filtered_stores))) {
    stop("filtered_stores don't exist")
  }

  unique_num <- unique(filtered_stores$STORE_NBR)
  result <- numeric(length = length(unique_num))
```

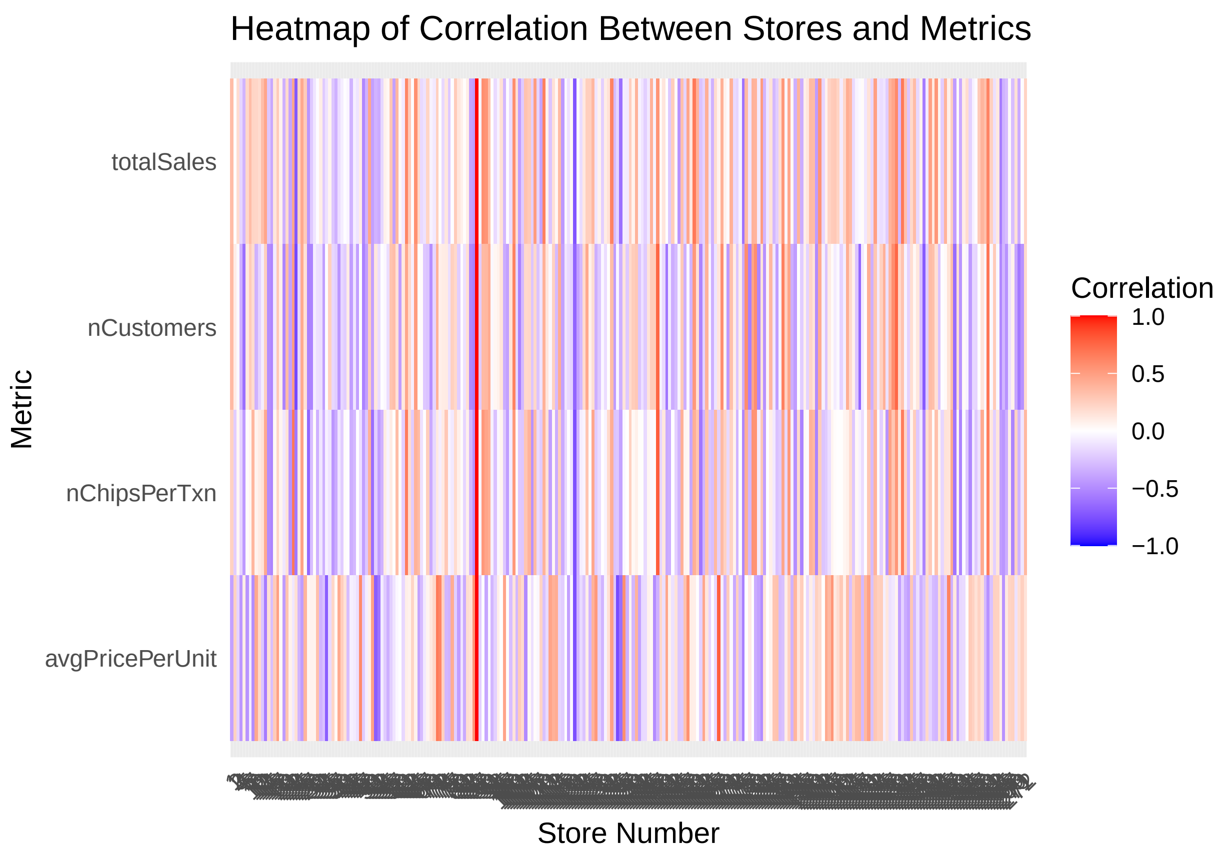
```r
  for (i in seq_along(unique_num)) {
    trail_vector <- filtered_stores[STORE_NBR == store1, ..metrics, with = FALSE]
    comparison_vector <- filtered_stores[STORE_NBR == unique_num[i], ..metrics, with = FALSE]

    if (all(sapply(trail_vector, function(x) all(is.na(x)))) || all(sapply(comparison_vector, function(x
      result[i] <- NA
    } else {
      result[i] <- cor(trail_vector, comparison_vector, method = method, use = "complete.obs")
    }
  }

  result_table <- data.table(STORE_NBR = unique_num, Correlation = result)
  return(result_table)
}

metrics <- c("totalSales","nCustomers","nChipsPerTxn","avgPricePerUnit")
result_cor_77_pearson <- sapply(metrics,function(m) calculate_cor1(77,m,"pearson"), simplify = FALSE)
result_cor_77_pearson <- rbindlist(result_cor_77_pearson, idcol = "Metric")
result_cor_86_pearson <- sapply(metrics,function(m) calculate_cor1(86,m,"pearson"), simplify = FALSE)
result_cor_86_pearson <- rbindlist(result_cor_86_pearson, idcol = "Metric")
result_cor_88_pearson <- sapply(metrics,function(m) calculate_cor1(88,m,"pearson"), simplify = FALSE)
result_cor_88_pearson <- rbindlist(result_cor_88_pearson, idcol = "Metric")
tibble(result_cor_77_pearson[order(-Correlation)])
```

```
## # A tibble: 1,040 x 3
##    Metric          STORE_NBR Correlation
##    <chr>               <int>       <dbl>
##  1 totalSales             77       1
##  2 nCustomers             77       1
##  3 nChipsPerTxn           77       1
##  4 avgPricePerUnit        77       1
##  5 nCustomers             35       0.788
##  6 totalSales             41       0.762
##  7 nCustomers             41       0.761
##  8 nCustomers            167       0.749
##  9 nChipsPerTxn           41       0.746
## 10 nCustomers             71       0.737
## # i 1,030 more rows
```

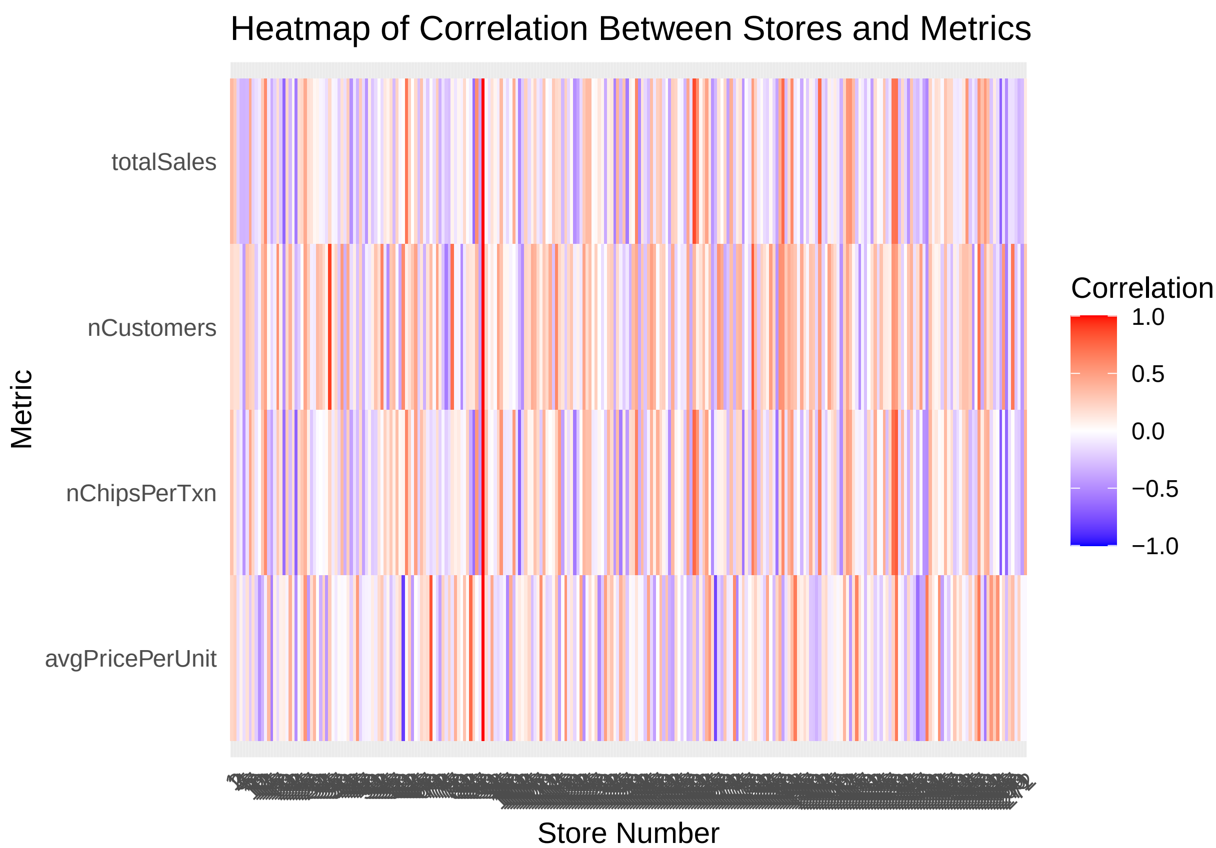
```r
ggplot(data = result_cor_77_pearson, aes(x = as.factor(STORE_NBR), y = Metric, fill = Correlation)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0,
                       limit = c(-1, 1), space = "Lab", name="Correlation") +
  theme_minimal() +
  labs(title = "Heatmap of Correlation Between Stores and Metrics",
       x = "Store Number",
       y = "Metric") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```r
ggplot(data = result_cor_86_pearson, aes(x = as.factor(STORE_NBR), y = Metric, fill = Correlation)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0,
                       limit = c(-1, 1), space = "Lab", name="Correlation") +
  theme_minimal() +
  labs(title = "Heatmap of Correlation Between Stores and Metrics",
       x = "Store Number",
       y = "Metric") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```r
ggplot(data = result_cor_88_pearson, aes(x = as.factor(STORE_NBR), y = Metric, fill = Correlation)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0,
                       limit = c(-1, 1), space = "Lab", name="Correlation") +
  theme_minimal() +
  labs(title = "Heatmap of Correlation Between Stores and Metrics",
       x = "Store Number",
       y = "Metric") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```r
#we can define a distance which means the difference of metrics between trial and other stores.
tibble(result_cor_77_pearson)
```

```
## # A tibble: 1,040 x 3
##    Metric      STORE_NBR Correlation
##    <chr>           <int>       <dbl>
##  1 totalSales          1      0.0644
##  2 totalSales          2      0.262
##  3 totalSales          3      0.163
##  4 totalSales          4     -0.295
##  5 totalSales          5     -0.285
##  6 totalSales          7     -0.159
##  7 totalSales         10     -0.424
##  8 totalSales         15     -0.236
##  9 totalSales         17      0.338
## 10 totalSales         19     -0.485
## # i 1,030 more rows
```

```r
sum_cor_central <- result_cor_77_pearson[STORE_NBR == 77,]
tibble(sum_cor_central)
```

```
## # A tibble: 4 x 3
##   Metric          STORE_NBR Correlation
##   <chr>               <int>       <dbl>
## 1 totalSales             77           1
```

```
## 2 nCustomers              77           1
## 3 nChipsPerTxn             77           1
## 4 avgPricePerUnit          77           1
```

```r
distance_avg <- result_cor_77_pearson[,.(distance = sum((Correlation - sum_cor_central$Correlation)^2))
tibble(distance_avg[order(distance)])
```

```
## # A tibble: 260 x 2
##    STORE_NBR distance
##        <int>    <dbl>
##  1        77    0
##  2       167    0.505
##  3        35    0.787
##  4       157    1.01
##  5        53    1.22
##  6       184    1.49
##  7        17    1.51
##  8       233    1.54
##  9       268    1.67
## 10       111    1.68
## # i 250 more rows
```

```r
#we can define a distance which means the difference of metrics between trial and other stores.
tibble(result_cor_86_pearson)
```

```
## # A tibble: 1,040 x 3
##    Metric     STORE_NBR Correlation
##    <chr>          <int>       <dbl>
##  1 totalSales         1     0.344
##  2 totalSales         2     0.00110
##  3 totalSales         3     0.178
##  4 totalSales         4    -0.179
##  5 totalSales         5    -0.322
##  6 totalSales         7     0.347
##  7 totalSales        10     0.184
##  8 totalSales        15    -0.375
##  9 totalSales        17     0.270
## 10 totalSales        19    -0.337
## # i 1,030 more rows
```

```r
sum_cor_central <- result_cor_86_pearson[STORE_NBR == 86,]
tibble(sum_cor_central)
```

```
## # A tibble: 4 x 3
##   Metric          STORE_NBR Correlation
##   <chr>               <int>       <dbl>
## 1 totalSales             86           1
## 2 nCustomers             86           1
## 3 nChipsPerTxn           86           1
## 4 avgPricePerUnit        86           1
```

```r
distance_avg <- result_cor_86_pearson[,.(distance = sum((Correlation - sum_cor_central$Correlation)^2))
tibble(distance_avg[order(distance)])
```

```
## # A tibble: 260 x 2
##    STORE_NBR distance
##        <int>    <dbl>
##  1        86     0
##  2       132     1.17
##  3        99     1.27
##  4       229     1.32
##  5       190     1.38
##  6        61     1.40
##  7       159     1.50
##  8       222     1.62
##  9        64     1.66
## 10       201     1.71
## # i 250 more rows
```

```r
#we can define a distance which means the difference of metrics between trial and other stores.
tibble(result_cor_88_pearson)
```

```
## # A tibble: 1,040 x 3
##    Metric     STORE_NBR Correlation
##    <chr>          <int>       <dbl>
##  1 totalSales         1      0.371
##  2 totalSales         2      0.284
##  3 totalSales         3     -0.208
##  4 totalSales         4     -0.320
##  5 totalSales         5     -0.325
##  6 totalSales         7      0.444
##  7 totalSales        10     -0.0938
##  8 totalSales        15     -0.357
##  9 totalSales        17      0.170
## 10 totalSales        19     -0.668
## # i 1,030 more rows
```

```r
sum_cor_central <- result_cor_88_pearson[STORE_NBR == 88,]
tibble(sum_cor_central)
```

```
## # A tibble: 4 x 3
##   Metric         STORE_NBR Correlation
##   <chr>              <int>       <dbl>
## 1 totalSales            88           1
## 2 nCustomers            88           1
## 3 nChipsPerTxn          88           1
## 4 avgPricePerUnit       88           1
```

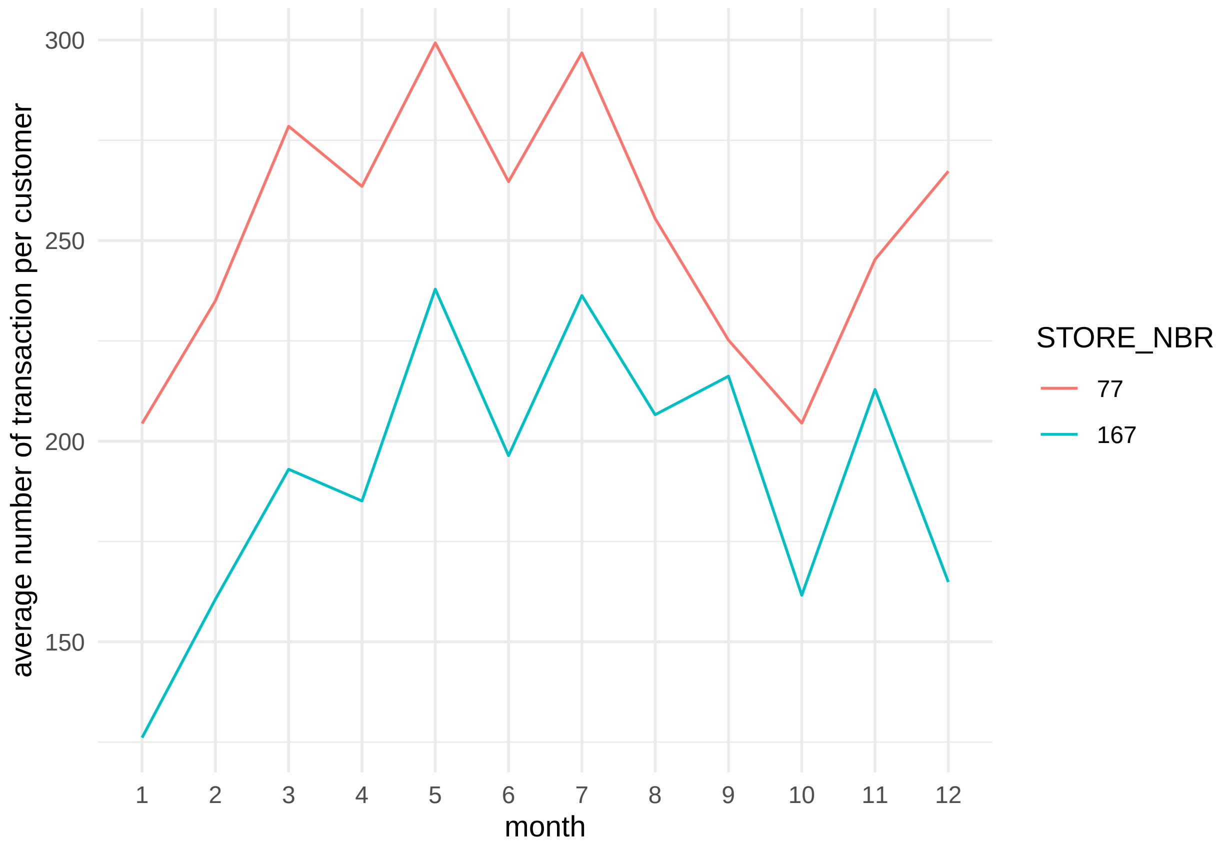
```r
distance_avg <- result_cor_88_pearson[,.(distance = sum((Correlation - sum_cor_central$Correlation)^2))
tibble(distance_avg[order(distance)])
```

```
## # A tibble: 260 x 2
```

```
##    STORE_NBR distance
##        <int>    <dbl>
##  1        88    0
##  2       229    0.486
##  3       257    0.694
##  4       228    0.980
##  5       159    1.04
##  6        26    1.17
##  7       178    1.18
##  8       191    1.25
##  9       140    1.36
## 10       187    1.57
## # i 250 more rows
```

We can get the store 167 has the minmum distance to store 77. Then we can compare their metrics:

```
ggplot(data = measureOverTime[STORE_NBR %in% c(77,167),]) +geom_line(aes(x= as.factor(month), y = totalS
```



we can find the store 77's total sales are more than store 167, then we can compare more metrics:

```
tibble(measureOverTime[STORE_NBR %in% c(77,167),])
```

```
## # A tibble: 24 x 7
##    month STORE_NBR totalSales nCustomers nChipsPerTxn avgPricePerUnit     N
##    <dbl>     <int>      <dbl>      <int>        <int>           <dbl> <int>
##  1     1        77       204.         35           65            3.14    12
```

```
## 2      1       167       126.          30          40            3.15    12
## 3      2        77       235           45          74            3.18    12
## 4      2       167       161.          35          50            3.21    12
## 5      3        77       278.          50          82            3.40    12
## 6      3       167       193           44          53            3.64    12
## 7      4        77       263.          47          78            3.38    12
## 8      4       167       185.          42          55            3.37    12
## 9      5       167       238.          55          72            3.30    12
## 10     5        77       299.          55          84            3.56    12
## # i 14 more rows
```

```r
measureOverTime[STORE_NBR %in% c(77,167),][order(month)]
```

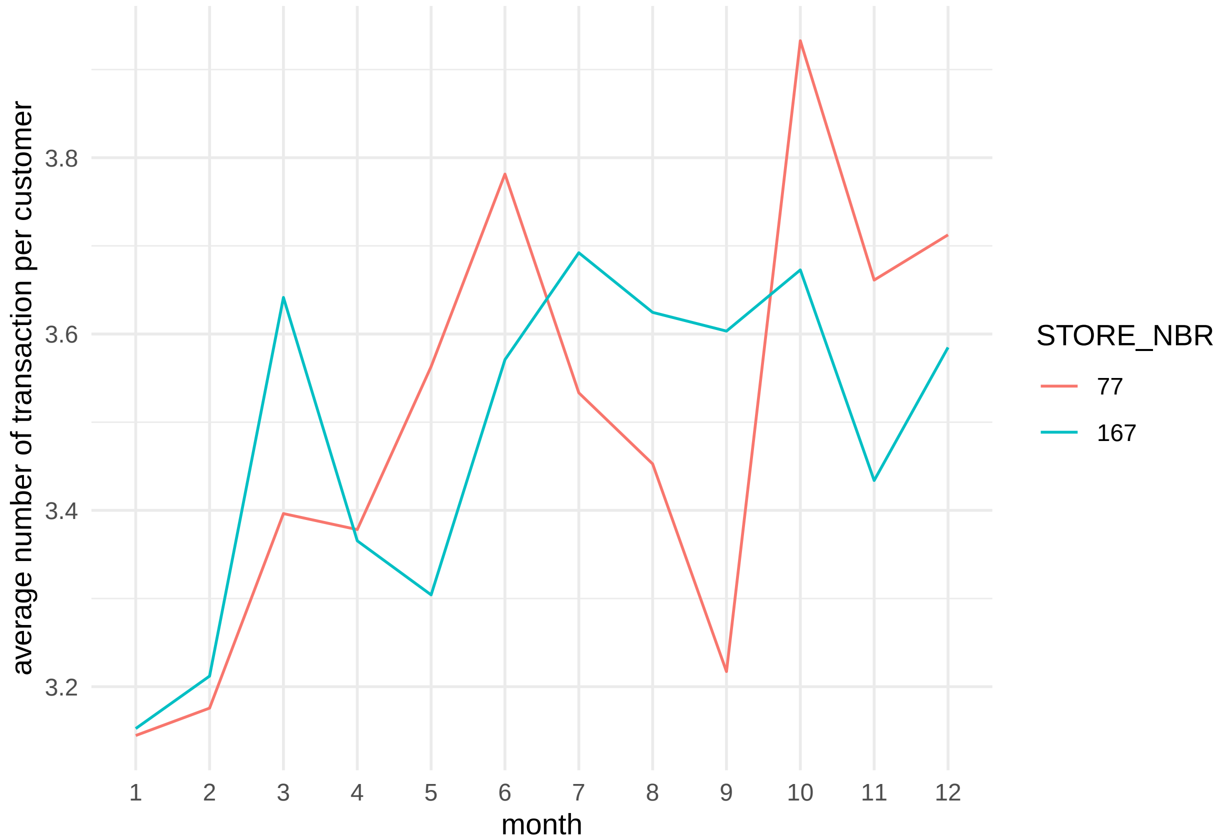
```
##      month STORE_NBR totalSales nCustomers nChipsPerTxn avgPricePerUnit     N
##      <num>     <int>      <num>      <int>        <int>           <num> <int>
##  1:      1        77      204.4         35           65        3.144615    12
##  2:      1       167      126.1         30           40        3.152500    12
##  3:      2        77      235.0         45           74        3.175676    12
##  4:      2       167      160.6         35           50        3.212000    12
##  5:      3        77      278.5         50           82        3.396341    12
##  6:      3       167      193.0         44           53        3.641509    12
##  7:      4        77      263.5         47           78        3.378205    12
##  8:      4       167      185.1         42           55        3.365455    12
##  9:      5       167      237.9         55           72        3.304167    12
## 10:      5        77      299.3         55           84        3.563095    12
## 11:      6        77      264.7         41           70        3.781429    12
## 12:      6       167      196.4         41           55        3.570909    12
## 13:      7        77      296.8         51           84        3.533333    12
## 14:      7       167      236.3         53           64        3.692188    12
## 15:      8        77      255.5         47           74        3.452703    12
## 16:      8       167      206.6         37           57        3.624561    12
## 17:      9        77      225.2         42           70        3.217143    12
## 18:      9       167      216.2         45           60        3.603333    12
## 19:     10        77      204.5         37           52        3.932692    12
## 20:     10       167      161.6         34           44        3.672727    12
## 21:     11       167      212.9         46           62        3.433871    12
## 22:     11        77      245.3         41           67        3.661194    12
## 23:     12       167      164.9         40           46        3.584783    12
## 24:     12        77      267.3         46           72        3.712500    12
##      month STORE_NBR totalSales nCustomers nChipsPerTxn avgPricePerUnit     N
```

```r
tibble(measureOverTime)
```

```
## # A tibble: 3,169 x 7
##     month STORE_NBR totalSales nCustomers nChipsPerTxn avgPricePerUnit     N
##     <dbl>     <int>      <dbl>      <int>        <int>           <dbl> <int>
##   1     1         1       155.         35           42            3.69    12
##   2     1         2       163.         43           49            3.32    12
##   3     1         3      1052.        102          236            4.46    12
##   4     1         4      1525         134          335            4.55    12
##   5     1         5       838          92          236            3.55    12
##   6     1         7       975.        102          221            4.41    12
##   7     1        10       879.        102          252            3.49    12
```

```
## 8       1          15        874.          99           248              3.52     12
## 9       1          17        365.          39            83              4.39     12
## 10      1          19        992          111           224              4.43     12
## # i 3,159 more rows
```

```
ggplot(data = measureOverTime[STORE_NBR %in% c(77,167),]) +geom_line(aes(x= as.factor(month), y = avgPri
```



the average sales are similar

```
tibble(measureOverTime[STORE_NBR %in% c(77,167),])
```

```
## # A tibble: 24 x 7
##    month STORE_NBR totalSales nCustomers nChipsPerTxn avgPricePerUnit     N
##    <dbl>     <int>      <dbl>      <int>        <int>           <dbl> <int>
##  1     1        77       204.         35           65            3.14    12
##  2     1       167       126.         30           40            3.15    12
##  3     2        77       235          45           74            3.18    12
##  4     2       167       161.         35           50            3.21    12
##  5     3        77       278.         50           82            3.40    12
##  6     3       167       193          44           53            3.64    12
##  7     4        77       263.         47           78            3.38    12
##  8     4       167       185.         42           55            3.37    12
##  9     5       167       238.         55           72            3.30    12
## 10     5        77       299.         55           84            3.56    12
## # i 14 more rows
```

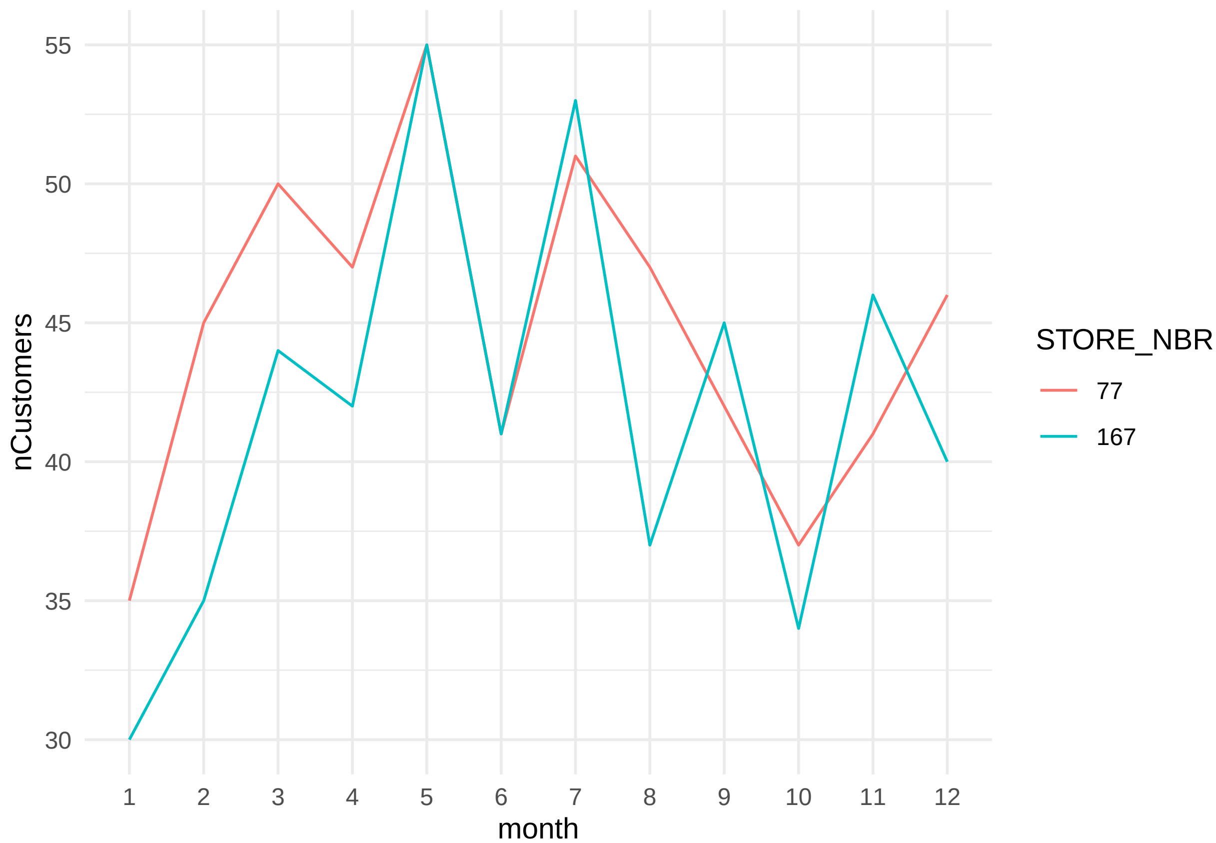
```r
measureOverTime[STORE_NBR %in% c(77,167),][order(month)]
```

```
##     month STORE_NBR totalSales nCustomers nChipsPerTxn avgPricePerUnit     N
##     <num>     <int>      <num>      <int>        <int>           <num> <int>
##  1:     1        77      204.4         35           65        3.144615    12
##  2:     1       167      126.1         30           40        3.152500    12
##  3:     2        77      235.0         45           74        3.175676    12
##  4:     2       167      160.6         35           50        3.212000    12
##  5:     3        77      278.5         50           82        3.396341    12
##  6:     3       167      193.0         44           53        3.641509    12
##  7:     4        77      263.5         47           78        3.378205    12
##  8:     4       167      185.1         42           55        3.365455    12
##  9:     5       167      237.9         55           72        3.304167    12
## 10:     5        77      299.3         55           84        3.563095    12
## 11:     6        77      264.7         41           70        3.781429    12
## 12:     6       167      196.4         41           55        3.570909    12
## 13:     7        77      296.8         51           84        3.533333    12
## 14:     7       167      236.3         53           64        3.692188    12
## 15:     8        77      255.5         47           74        3.452703    12
## 16:     8       167      206.6         37           57        3.624561    12
## 17:     9        77      225.2         42           70        3.217143    12
## 18:     9       167      216.2         45           60        3.603333    12
## 19:    10        77      204.5         37           52        3.932692    12
## 20:    10       167      161.6         34           44        3.672727    12
## 21:    11       167      212.9         46           62        3.433871    12
## 22:    11        77      245.3         41           67        3.661194    12
## 23:    12       167      164.9         40           46        3.584783    12
## 24:    12        77      267.3         46           72        3.712500    12
##     month STORE_NBR totalSales nCustomers nChipsPerTxn avgPricePerUnit     N
```

```r
tibble(measureOverTime)
```

```
## # A tibble: 3,169 x 7
##    month STORE_NBR totalSales nCustomers nChipsPerTxn avgPricePerUnit     N
##    <dbl>     <int>      <dbl>      <int>        <int>           <dbl> <int>
##  1     1         1       155.         35           42            3.69    12
##  2     1         2       163.         43           49            3.32    12
##  3     1         3      1052.        102          236            4.46    12
##  4     1         4      1525         134          335            4.55    12
##  5     1         5       838          92          236            3.55    12
##  6     1         7       975.        102          221            4.41    12
##  7     1        10       879.        102          252            3.49    12
##  8     1        15       874.         99          248            3.52    12
##  9     1        17       365.         39           83            4.39    12
## 10     1        19       992         111          224            4.43    12
## # i 3,159 more rows
```

```r
ggplot(data = measureOverTime[STORE_NBR %in% c(77,167),]) +geom_line(aes(x= as.factor(month), y = nCust
```




```
tibble(measureOverTime[STORE_NBR %in% c(77,167),])
```

```
## # A tibble: 24 x 7
##    month STORE_NBR totalSales nCustomers nChipsPerTxn avgPricePerUnit     N
##    <dbl>     <int>      <dbl>      <int>        <int>           <dbl> <int>
##  1     1        77       204.         35           65            3.14    12
##  2     1       167       126.         30           40            3.15    12
##  3     2        77       235          45           74            3.18    12
##  4     2       167       161.         35           50            3.21    12
##  5     3        77       278.         50           82            3.40    12
##  6     3       167       193          44           53            3.64    12
##  7     4        77       263.         47           78            3.38    12
##  8     4       167       185.         42           55            3.37    12
##  9     5       167       238.         55           72            3.30    12
## 10     5        77       299.         55           84            3.56    12
## # i 14 more rows
```

```
measureOverTime[STORE_NBR %in% c(77,167),][order(month)]
```

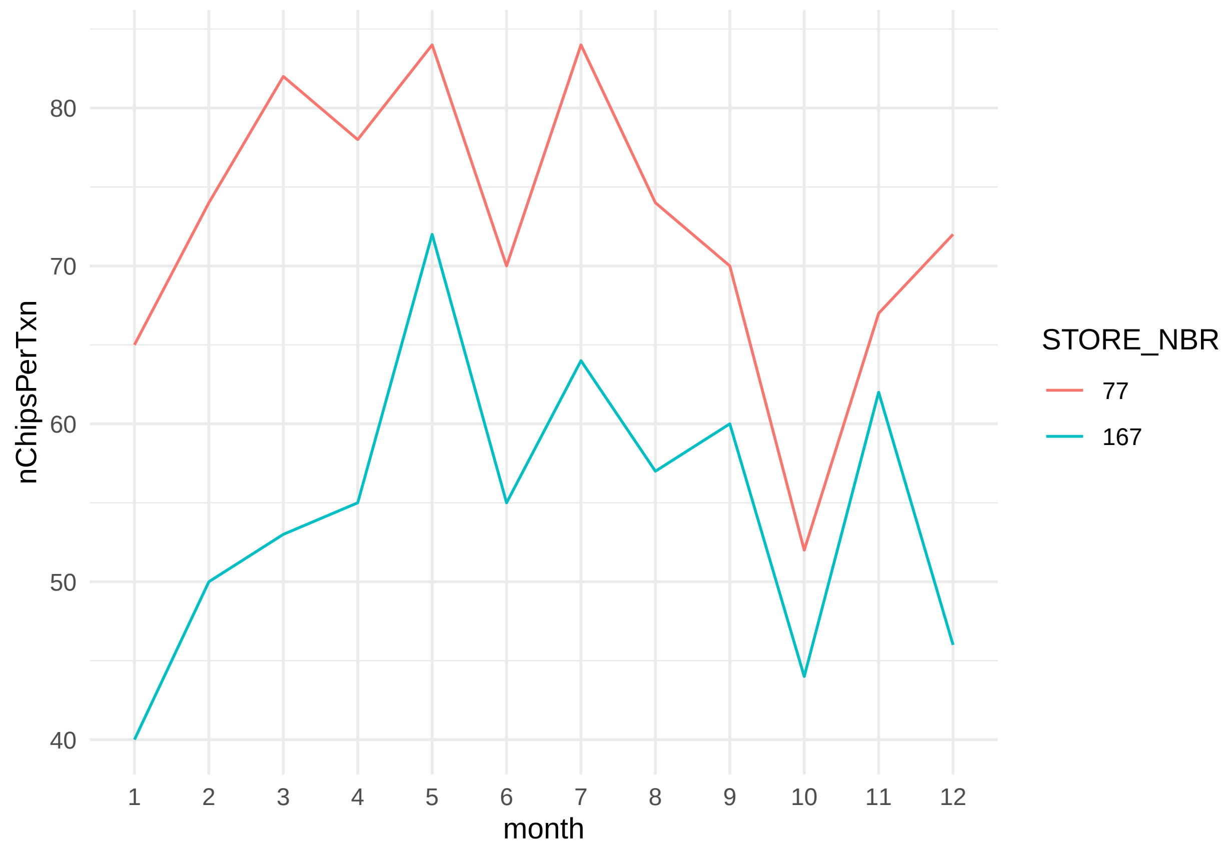
```
##     month STORE_NBR totalSales nCustomers nChipsPerTxn avgPricePerUnit     N
##     <num>     <int>      <num>      <int>        <int>           <num> <int>
##  1:     1        77      204.4         35           65        3.144615    12
##  2:     1       167      126.1         30           40        3.152500    12
##  3:     2        77      235.0         45           74        3.175676    12
##  4:     2       167      160.6         35           50        3.212000    12
```

```
##  5:     3       77      278.5          50            82         3.396341    12
##  6:     3      167      193.0          44            53         3.641509    12
##  7:     4       77      263.5          47            78         3.378205    12
##  8:     4      167      185.1          42            55         3.365455    12
##  9:     5      167      237.9          55            72         3.304167    12
## 10:     5       77      299.3          55            84         3.563095    12
## 11:     6       77      264.7          41            70         3.781429    12
## 12:     6      167      196.4          41            55         3.570909    12
## 13:     7       77      296.8          51            84         3.533333    12
## 14:     7      167      236.3          53            64         3.692188    12
## 15:     8       77      255.5          47            74         3.452703    12
## 16:     8      167      206.6          37            57         3.624561    12
## 17:     9       77      225.2          42            70         3.217143    12
## 18:     9      167      216.2          45            60         3.603333    12
## 19:    10       77      204.5          37            52         3.932692    12
## 20:    10      167      161.6          34            44         3.672727    12
## 21:    11      167      212.9          46            62         3.433871    12
## 22:    11       77      245.3          41            67         3.661194    12
## 23:    12      167      164.9          40            46         3.584783    12
## 24:    12       77      267.3          46            72         3.712500    12
##      month STORE_NBR totalSales nCustomers nChipsPerTxn avgPricePerUnit     N
```

```r
tibble(measureOverTime)
```

```
## # A tibble: 3,169 x 7
##     month STORE_NBR totalSales nCustomers nChipsPerTxn avgPricePerUnit     N
##     <dbl>     <int>      <dbl>      <int>        <int>           <dbl> <int>
##  1      1         1       155.         35           42            3.69    12
##  2      1         2       163.         43           49            3.32    12
##  3      1         3      1052.        102          236            4.46    12
##  4      1         4      1525         134          335            4.55    12
##  5      1         5       838          92          236            3.55    12
##  6      1         7       975.        102          221            4.41    12
##  7      1        10       879.        102          252            3.49    12
##  8      1        15       874.         99          248            3.52    12
##  9      1        17       365.         39           83            4.39    12
## 10      1        19       992         111          224            4.43    12
## # i 3,159 more rows
```

```r
ggplot(data = measureOverTime[STORE_NBR %in% c(77,167),]) +geom_line(aes(x= as.factor(month), y = nChips
```



we found the answer is that the customer from store 77 tend to buy more chips every times which result in the total sales much larger than store 167 (although the total number of customer is similar).

By correlations, we can also get that store 132 is similar with store 88 comprehensively. but we can compare the difference:

```
tibble(measureOverTime[STORE_NBR %in% c(86,132),])
```

```
## # A tibble: 24 x 7
##    month STORE_NBR totalSales nCustomers nChipsPerTxn avgPricePerUnit     N
##    <dbl>     <int>      <dbl>      <int>        <int>           <dbl> <int>
##  1     1        86      841.          94          260            3.24    12
##  2     1       132       29.6          5            9            3.29    12
##  3     2        86      913.         107          277            3.30    12
##  4     2       132       15.4          3            6            2.57    12
##  5     3        86     1027.         115          284            3.62    12
##  6     3       132       72.8          9           18            4.04    12
##  7     4        86      848.         105          254            3.34    12
##  8     4       132       22            3            6            3.67    12
##  9     5        86      889.         104          259            3.43    12
## 10     5       132       75.1         11           23            3.27    12
## # i 14 more rows
```

```
measureOverTime[STORE_NBR %in% c(86,132),][order(month)]
```

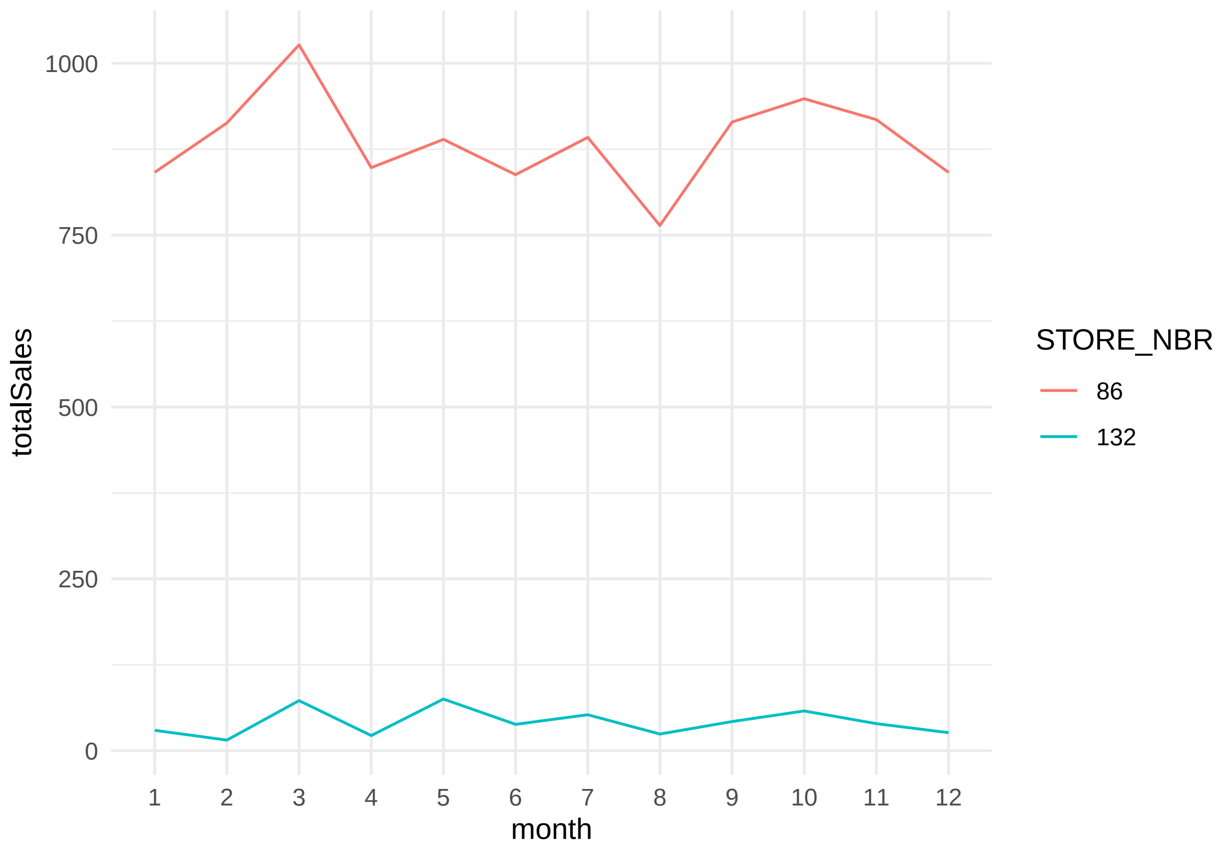
```
##     month STORE_NBR totalSales nCustomers nChipsPerTxn avgPricePerUnit     N
```

```
##      <num>     <int>      <num>      <int>        <int>           <num> <int>
##   1:     1        86     841.40         94          260        3.236154    12
##   2:     1       132      29.60          5            9        3.288889    12
##   3:     2        86     913.20        107          277        3.296751    12
##   4:     2       132      15.40          3            6        2.566667    12
##   5:     3        86    1026.80        115          284        3.615493    12
##   6:     3       132      72.80          9           18        4.044444    12
##   7:     4        86     848.20        105          254        3.339370    12
##   8:     4       132      22.00          3            6        3.666667    12
##   9:     5        86     889.30        104          259        3.433591    12
##  10:     5       132      75.10         11           23        3.265217    12
##  11:     6        86     838.00         98          238        3.521008    12
##  12:     6       132      38.30          5            9        4.255556    12
##  13:     7        86     892.20         99          251        3.554582    12
##  14:     7       132      52.30         11           17        3.076471    12
##  15:     8       132      24.20          4            7        3.457143    12
##  16:     8        86     764.05         94          215        3.553721    12
##  17:     9        86     914.60        103          258        3.544961    12
##  18:     9       132      42.40          5           12        3.533333    12
##  19:    10        86     948.40        109          276        3.436232    12
##  20:    10       132      57.80         10           18        3.211111    12
##  21:    11        86     918.00        100          254        3.614173    12
##  22:    11       132      39.30          6           11        3.572727    12
##  23:    12        86     841.20         98          240        3.505000    12
##  24:    12       132      26.20          4            8        3.275000    12
##      month STORE_NBR totalSales nCustomers nChipsPerTxn avgPricePerUnit     N
```

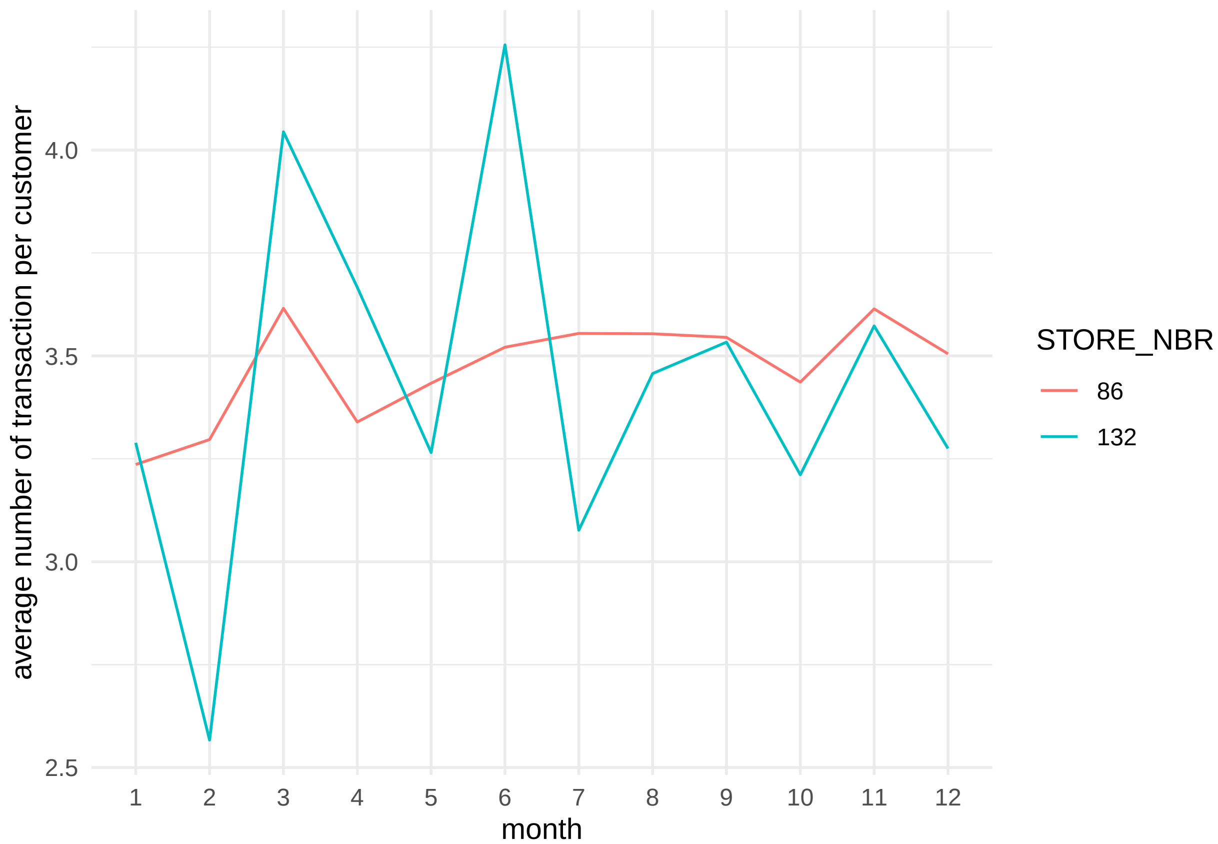
```
tibble(measureOverTime)
```

```
## # A tibble: 3,169 x 7
##    month STORE_NBR totalSales nCustomers nChipsPerTxn avgPricePerUnit     N
##    <dbl>     <int>      <dbl>      <int>        <int>           <dbl> <int>
##  1     1         1       155.         35           42            3.69    12
##  2     1         2       163.         43           49            3.32    12
##  3     1         3      1052.        102          236            4.46    12
##  4     1         4      1525         134          335            4.55    12
##  5     1         5       838          92          236            3.55    12
##  6     1         7       975.        102          221            4.41    12
##  7     1        10       879.        102          252            3.49    12
##  8     1        15       874.         99          248            3.52    12
##  9     1        17       365.         39           83            4.39    12
## 10     1        19       992         111          224            4.43    12
## # i 3,159 more rows
```

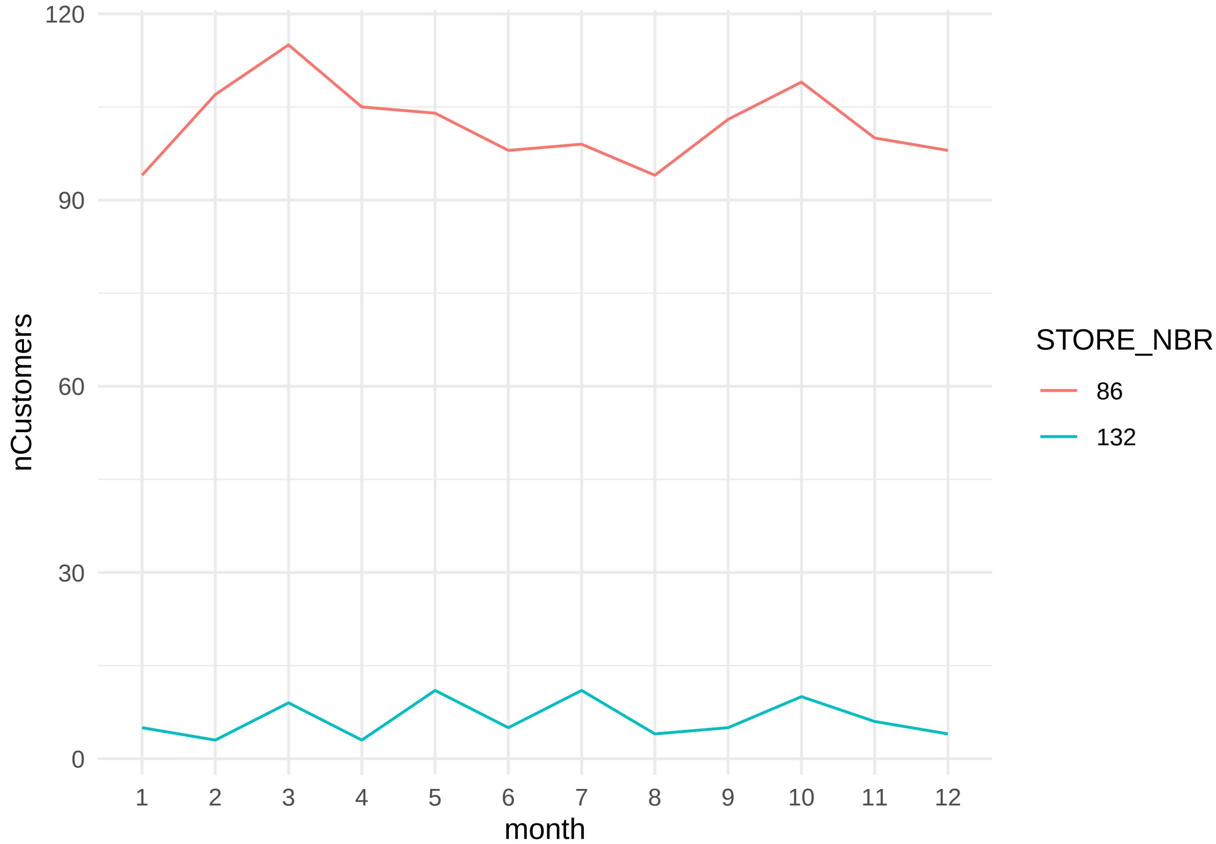
```
ggplot(data = measureOverTime[STORE_NBR %in% c(86,132),]) +geom_line(aes(x= as.factor(month), y = totalS
```



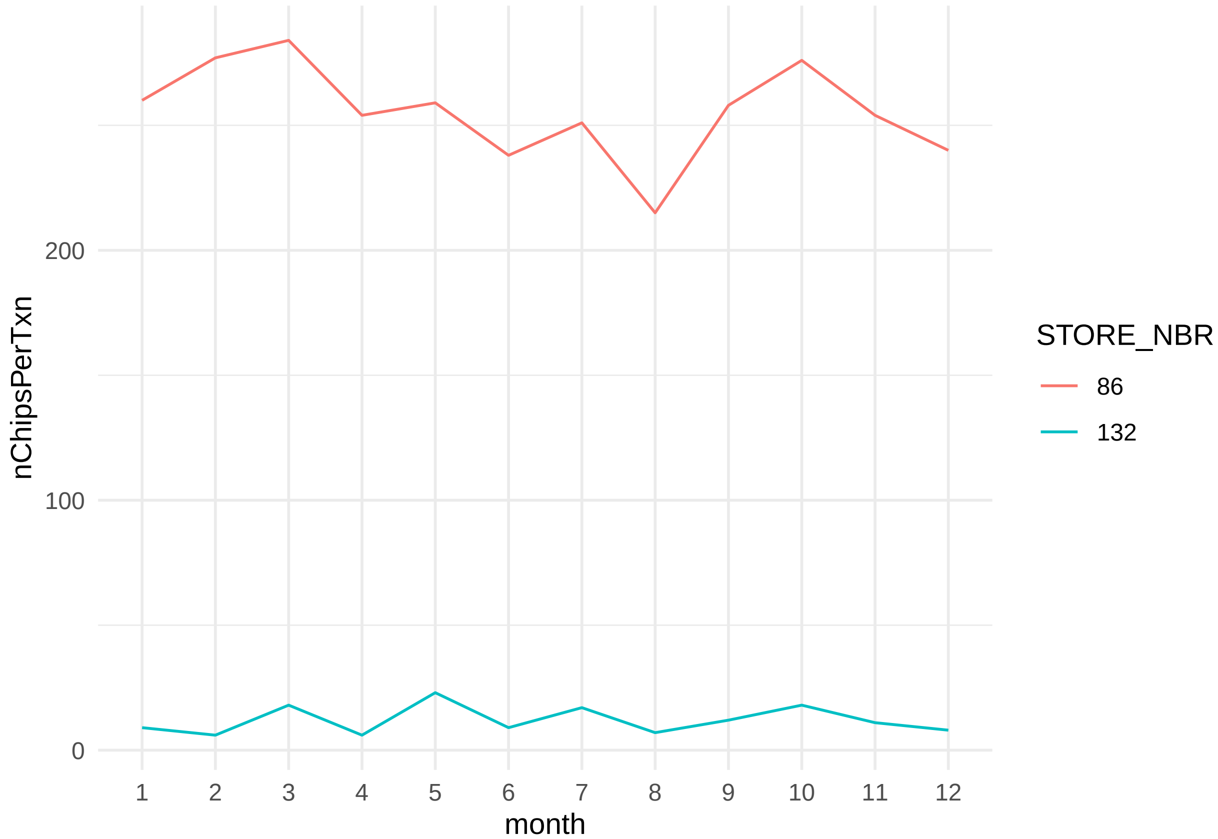

```
ggplot(data = measureOverTime[STORE_NBR %in% c(86,132),]) +geom_line(aes(x= as.factor(month), y = avgPri
```




```
ggplot(data = measureOverTime[STORE_NBR %in% c(86,132),]) +geom_line(aes(x= as.factor(month), y = nCust
```



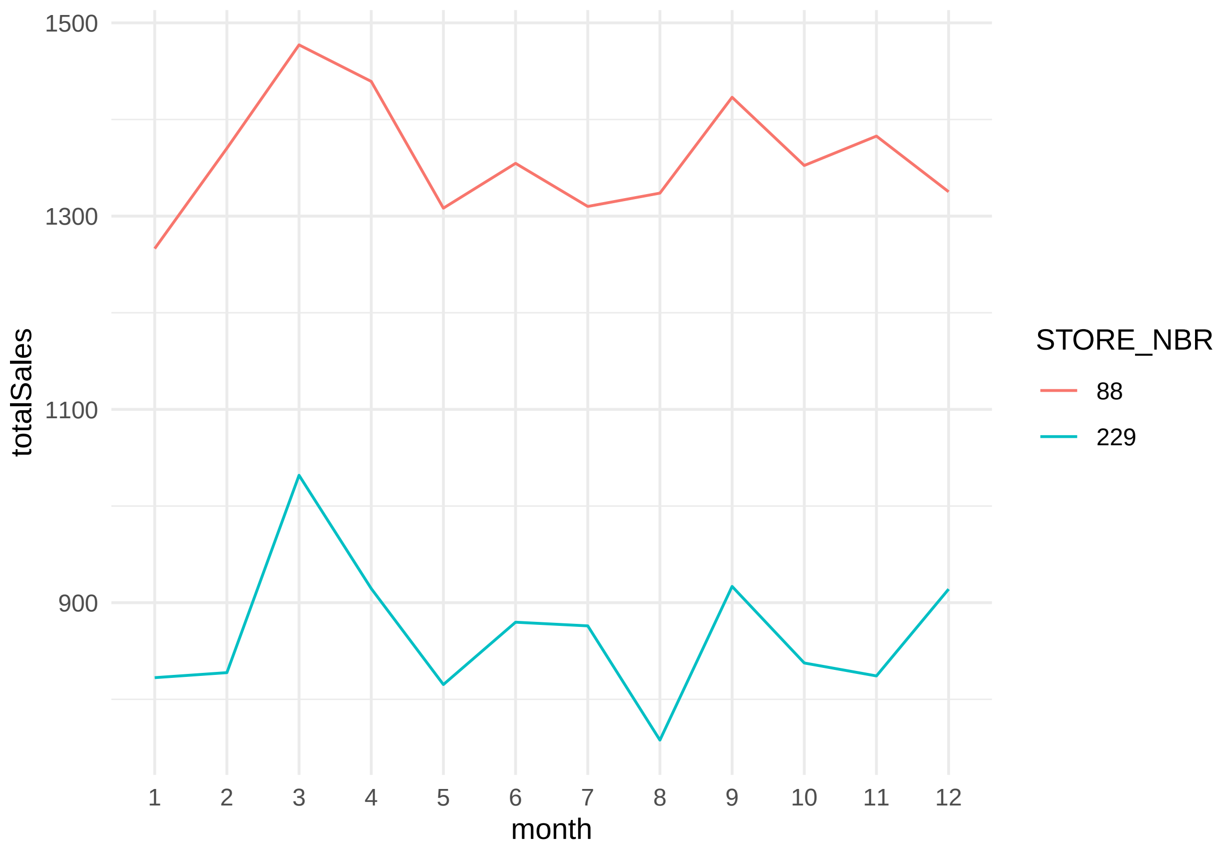

```
ggplot(data = measureOverTime[STORE_NBR %in% c(86,132),]) +geom_line(aes(x= as.factor(month), y = nChips
```



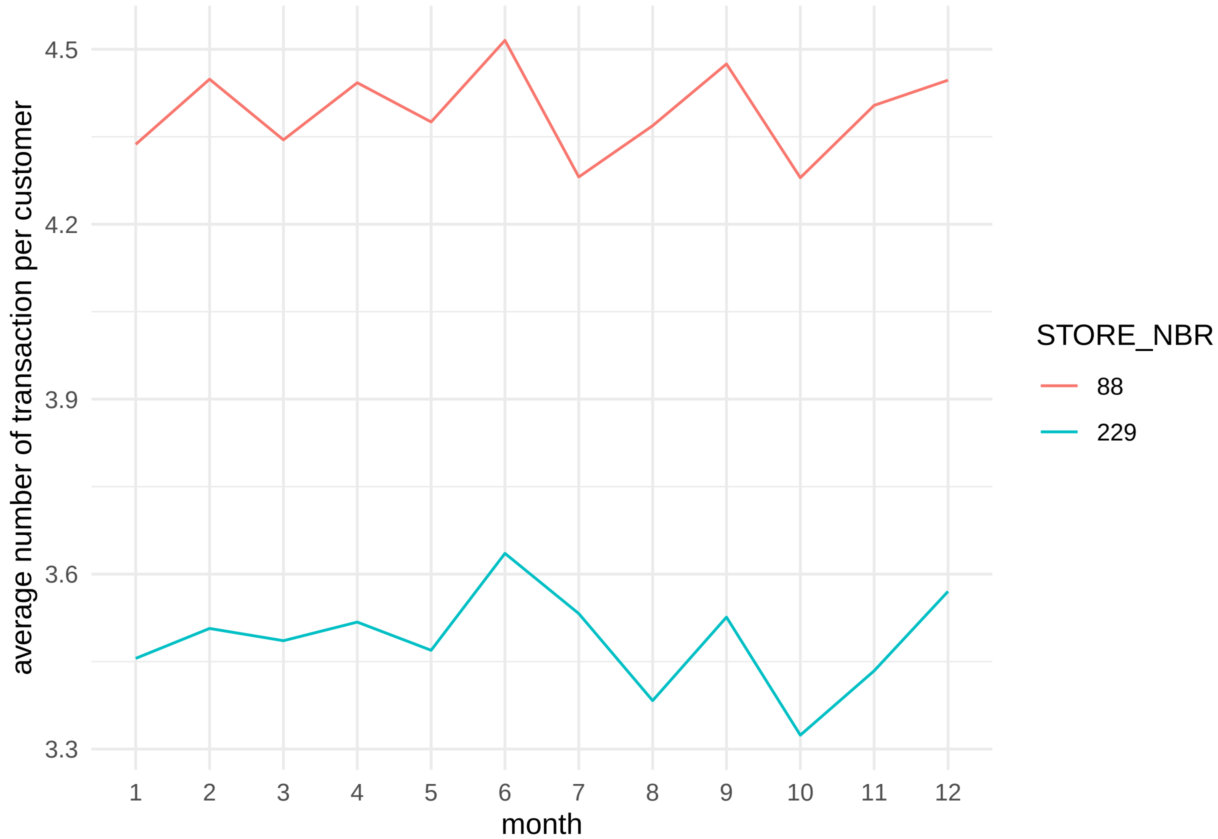
the differences between two stores are from the total number of customer and the number of chips purchased every time.

Finally for store 88 and 229

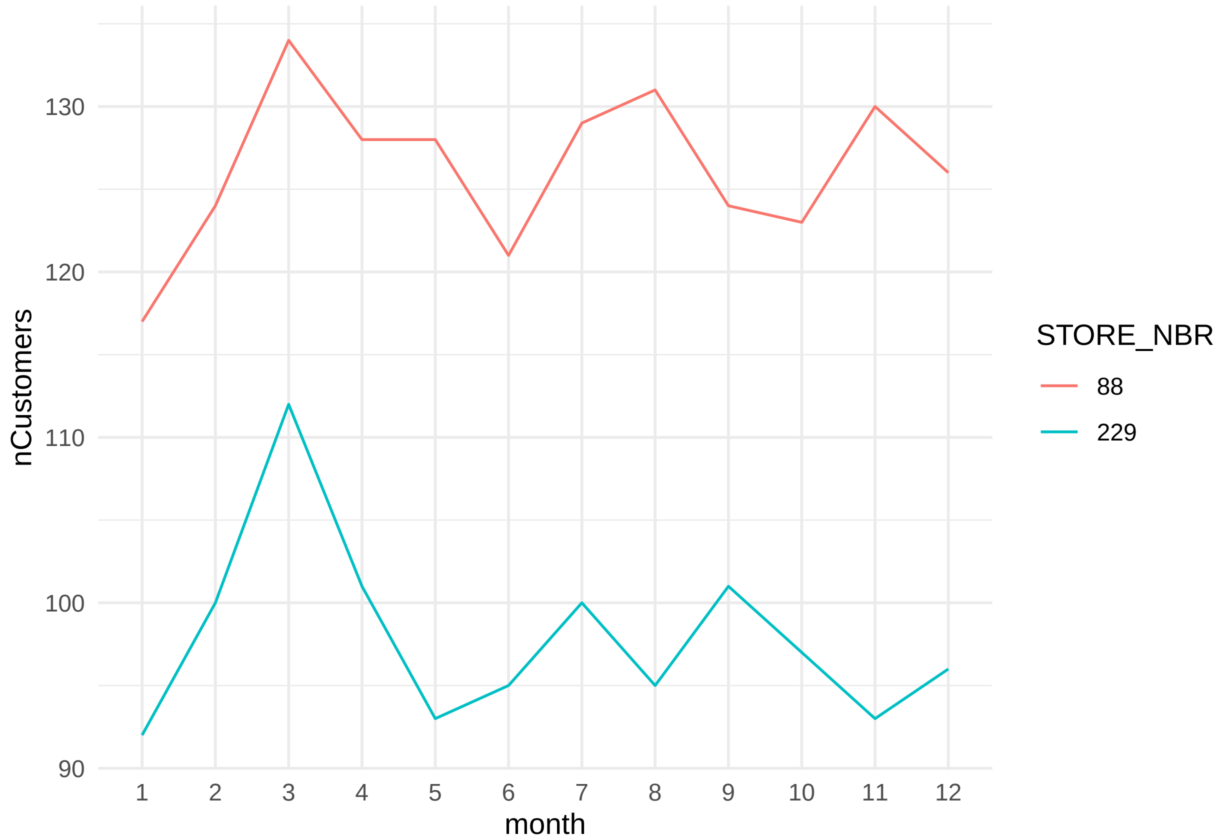
```
ggplot(data = measureOverTime[STORE_NBR %in% c(88,229),]) +geom_line(aes(x= as.factor(month), y = totalS
```



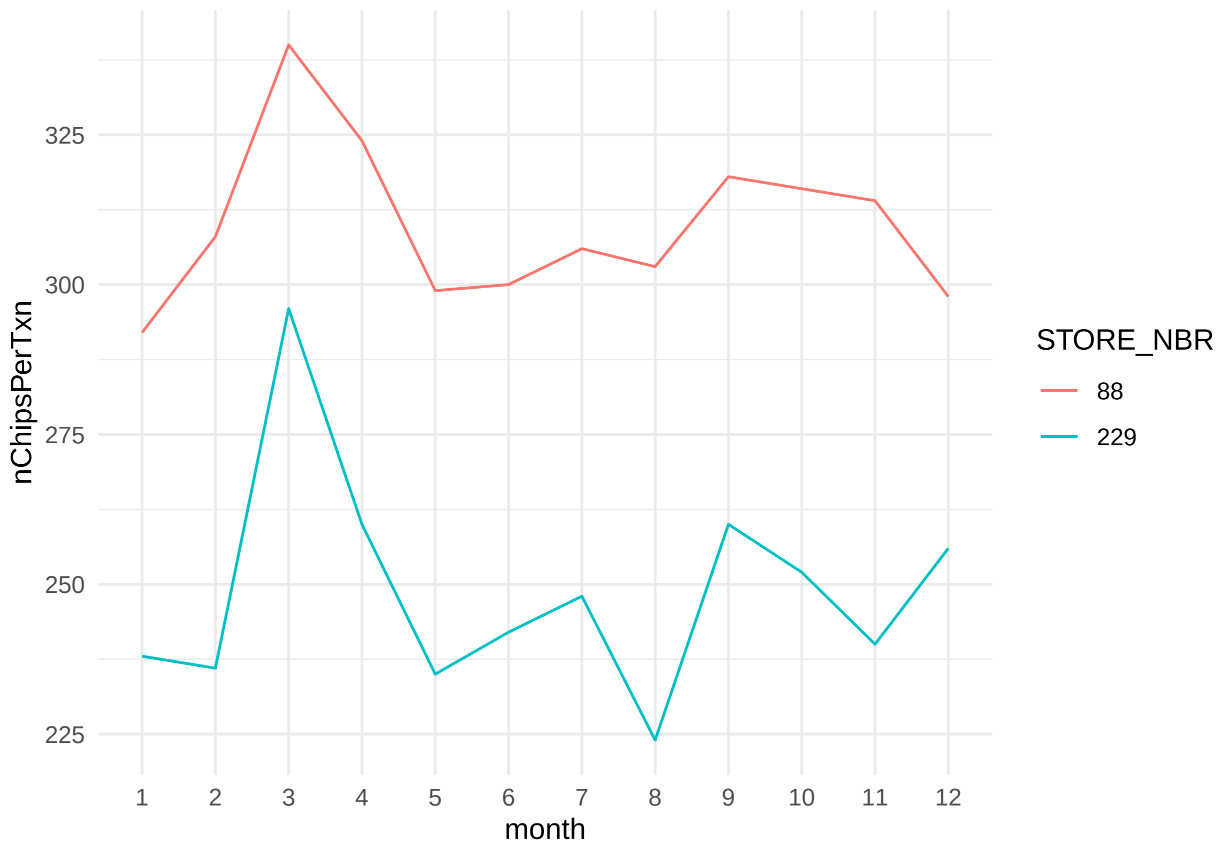

```
ggplot(data = measureOverTime[STORE_NBR %in% c(88,229),]) +geom_line(aes(x= as.factor(month), y = avgPri
```




```
ggplot(data = measureOverTime[STORE_NBR %in% c(88,229),]) +geom_line(aes(x= as.factor(month), y = nCust
```




```
ggplot(data = measureOverTime[STORE_NBR %in% c(88,229),]) +geom_line(aes(x= as.factor(month), y = nChips
```



the store metrics are much better than store 229.

Conclusion

- three stores have their own pros and cons compared to the corrlated stores in the whole year.
- the most effective one is store 86, it shows this store has the highest improvement compared to similar stores.
- the store 77 performance is little better than store 167 and it is from the improvement of number of chips per transaction.
- the store 88 have a comprehensive improvement compared with store 229 in all metrics.