Double-click (or enter) to edit

# Home assignement

## Time series data visualization

Double-click (or enter) to edit

**Task**: to depict the overall pricing behaviour of retailers (not grouped, see Requested results 2) in various categories and produce conditional descriptive statistics for meaningful interpretation of for instance the price dispersion and trajectories.

```
What can be said of the price dispersion in these categories, and are there other notable trends?
```

*Hint: visually present the descriptive statistics using the ggplot2 package in R*

```
install.packages("data.table")
install.packages("psych")
install.packages("dplyr")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘tmvnsim’, ‘mnormt’


Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

```
library(data.table)
```

```
Attaching package: ‘data.table’


The following objects are masked from ‘package:dplyr’:

    between, first, last
```

```
# LOAD DATA ############################################################
# creating a dataframe from the pricing dataset
```

```
# creating a dataframe from the pricing dataset
#data_pricing <- read.csv("home_assignment_data_pricing.csv")

data_pricing <- fread("home_assignment_data_pricing.csv")
```

## Explore the dataframe

```
# EXPLORE DATA ################################################
# Using head or tail to look at part of it, 6 first lines by default
# here the first 3 lines

tail(data_pricing,3)
```

| product_id | date | category | price | weekday | week | store_id | year | month | cpi_ |
|---|---|---|---|---|---|---|---|---|---|
| <int> | <date> | <chr> | <int> | <chr> | <int> | <int> | <int> | <fct> | |
| 969989 | 2016-05-12 | Headphones | 656 | Thursday | 19 | 4819 | 2016 | May | |
| 969989 | 2016-05-12 | Headphones | 547 | Thursday | 19 | 1260 | 2016 | May | |
| 969989 | 2016-05-12 | Headphones | 394 | Thursday | 19 | 2380 | 2016 | May | |

```
# Check the dimension, nrow x ncol
# 2612522 rows and 14 columns
dim(data_pricing)
```

2612522 · 14

```
# Let's look for missing values, if any
colSums(is.na(data_pricing))
```

```
product_id: 0 date: 0 category: 0 price: 0 weekday: 0 week: 0
store_id: 0 year: 0 month_numerical: 0 cpi_adjusted_price: 0
log_of_cpi_adjusted_price: 0 number_of_store_per_product_and_day: 0
total_days_for_product: 0 total_products_in_category: 0
```

```
# EXAMINE RESULTS #############################################
# Check datatype of dataframe columns
str(data_pricing)
```

```
Classes ‘data.table’ and 'data.frame':  1110820 obs. of  14 variables:
 $ product_id                       : int  1030146 1030146 1030146 1030146 1030146
 $ date                             : IDate, format: "2013-01-30" "2013-01-30" ...
 $ category                         : chr  "Cellphones" "Cellphones" "Cellphones"
 $ price                            : int  1195 1195 1195 1195 1195 1195 1195 1195
 $ weekday                          : chr  "Wednesday" "Wednesday" "Wednesday" "Th
 $ week                             : int  5 5 5 5 5 5 5 5 5 5 ...
 $ store_id                         : int  6852 11909 16724 6852 11909 16724 6852
```

```
$ year                              : int  2013 2013 2013 2013 2013 2013 2013 2013
$ month_numerical                   : int  1 1 1 1 1 1 2 2 2 2 ...
$ cpi_adjusted_price                : num  1225 1225 1225 1225 1225 ...
$ log_of_cpi_adjusted_price         : num  7.11 7.11 7.11 7.11 7.11 ...
$ number_of_store_per_product_and_day: int  3 3 3 3 3 3 3 3 3 3 ...
$ total_days_for_product            : int  4464 4464 4464 4464 4464 4464 4464 4464
$ total_products_in_category        : int  67 67 67 67 67 67 67 67 67 67 ...
- attr(*, ".internal.selfref")=<externalptr>
```

```
summary(data_pricing)
```

```
  product_id           date              category             price        
Min.   :  13189   Min.   :2012-01-01   Length:2612522     Min.   :   19  
1st Qu.: 164639   1st Qu.:2015-03-29   Class :character   1st Qu.:  252  
Median : 541262   Median :2016-04-01   Mode  :character   Median :  661  
Mean   : 929360   Mean   :2015-11-08                      Mean   : 1941  
3rd Qu.: 951367   3rd Qu.:2016-10-17                      3rd Qu.: 2585  
Max.   :3897064   Max.   :2017-02-25                      Max.   :21100  
  weekday               week          store_id          year         
Length:2612522     Min.   : 1.00   Min.   :    1   Min.   :2012  
Class :character   1st Qu.:11.00   1st Qu.:  658   1st Qu.:2015  
Mode  :character   Median :28.00   Median : 2714   Median :2016  
                   Mean   :26.88   Mean   : 5822   Mean   :2015  
                   3rd Qu.:42.00   3rd Qu.: 8744   3rd Qu.:2016  
                   Max.   :53.00   Max.   :29792   Max.   :2017  
month_numerical  cpi_adjusted_price log_of_cpi_adjusted_price
Min.   : 1.000   Min.   :   19.33   Min.   :2.962            
1st Qu.: 3.000   1st Qu.:  255.43   1st Qu.:5.543            
Median : 7.000   Median :  669.42   Median :6.506            
Mean   : 6.564   Mean   : 1962.08   Mean   :6.655            
3rd Qu.:10.000   3rd Qu.: 2603.66   3rd Qu.:7.865            
Max.   :12.000   Max.   :21544.74   Max.   :9.978            
number_of_store_per_product_and_day total_days_for_product
Min.   : 3.0                        Min.   : 3001         
1st Qu.: 6.0                        1st Qu.: 6322         
Median :10.0                        Median :10088         
Mean   :14.1                        Mean   :12779         
3rd Qu.:19.0                        3rd Qu.:18124         
Max.   :69.0                        Max.   :37577         
total_products_in_category
Min.   : 67.0             
1st Qu.:987.0             
Median :987.0             
Mean   :836.7             
3rd Qu.:987.0             
Max.   :987.0             
```

```
#library(psych)
```

```
describe(data_pricing)
```

```
Attaching package: ‘psych’


The following objects are masked from ‘package:ggplot2’:

    %+%, alpha


Warning message in FUN(newX[, i], ...):
“no non-missing arguments to min; returning Inf”
Warning message in FUN(newX[, i], ...):
“no non-missing arguments to min; returning Inf”
Warning message in FUN(newX[, i], ...):
“no non-missing arguments to min; returning Inf”
Warning message in FUN(newX[, i], ...):
“no non-missing arguments to max; returning -Inf”
Warning message in FUN(newX[, i], ...):
“no non-missing arguments to max; returning -Inf”
Warning message in FUN(newX[, i], ...):
“no non-missing arguments to max; returning -Inf”
```

A psych: 14 × 8

| | vars | n | mean | sd | |
|---|---|---|---|---|---|
| | <int> | <dbl> | <dbl> | <dbl> | |
| **product_id** | 1 | 2612522 | 9.293596e+05 | 1.107846e+06 | 1318 |
| **date** | 2 | 2612522 | NaN | NA | |
| **category** | 3 | 2612522 | NaN | NA | |
| **price** | 4 | 2612522 | 1.940538e+03 | 2.801254e+03 | |
| **weekday** | 5 | 2612522 | NaN | NA | |
| **week** | 6 | 2612522 | 2.687916e+01 | 1.625761e+01 | |
| **store_id** | 7 | 2612522 | 5.821576e+03 | 7.228734e+03 | |
| **year** | 8 | 2612522 | 2.015350e+03 | 1.235826e+00 | 20 |
| **month_numerical** | 9 | 2612522 | 6.564006e+00 | 3.719976e+00 | |
| **cpi_adjusted_price** | 10 | 2612522 | 1.962082e+03 | 2.830985e+03 | |
| **log_of_cpi_adjusted_price** | 11 | 2612522 | 6.655107e+00 | 1.441979e+00 | |
| **number_of_store_per_product_and_day** | 12 | 2612522 | 1.410284e+01 | 1.136254e+01 | |

```
# Check structure of data column
tail(data_pricing$date)
```

2017-02-20 · 2017-02-21 · 2017-02-22 · 2017-02-23 · 2017-02-24 · 2017-02-25

```
str(data_pricing$date)
```

```
 IDate[1:1110820], format: "2013-01-30" "2013-01-30" "2013-01-30" "2013-01-31" "2013
```

```
# Check dataframe row names, here the first 10 rows
row.names(data_pricing[1:10,])
```

'1' · '2' · '3' · '4' · '5' · '6' · '7' · '8' · '9' · '10'

```
# Check date class, convert into date if not already
class(data_pricing$date)
```

'IDate' · 'Date'

```
# Unify the date class all over the dataset
date <- as.Date(data_pricing$date)
tail(date,5)
class(date)
```

2016-05-12 · 2016-05-12 · 2016-05-12 · 2016-05-12 · 2016-05-12
'Date'

```
# Thus no need to convert the string of dates into R 'Date' object
#sub_headphones$date <- as.Date(sub_headphones$date, "%Y-%m-%d"); head(sub_headphones$date
```

```
# convert month numbers to names, using a built-in constant
data_pricing$month_numerical <- factor(data_pricing$month_numerical)
levels(data_pricing$month_numerical) <- month.abb
```

```
# rename month_numerical column into simply month

# out of safety, let's use another variable df for later us
# while we keep doing data manipulation with data_pricing
#df <- data.frame(data_pricing)

colnames(data_pricing)[9] <- 'month'
```

```
head(data_pricing, 5)
```

| product id | date | category | price | weekday | week | store id | year | month | cpi |
|---|---|---|---|---|---|---|---|---|---|
| 1030146 | 2013-... | Cellphones | 1195 | Wednesday | 5 | 6852 | 2013 | Jan | |

```
# load dplyr
library(dplyr)
```

```
# Identify unique category items in the dataframe
# using distinct function
distinct(data.frame(data_pricing$category))
```

A data.frame: 2 × 1

| data_pricing.category |
|---|
| <chr> |
| Cellphones |
| Headphones |

```
#Headphones <- subset(data_pricing, category == "Headphones")
```

```
dim(Headphones)
```

2185645 · 14

```
# Let's focus on one of these categories, Xbox 360 in my case
#xbox360 <- subset(data_pricing, category == "Xbox 360")

## Or load csv files you would have saved previously
xbox360 <- fread("xbox360.csv")
```

```
# remove the new index column created after saving as csv
xbox360 <- subset(xbox360[, 2:15])

tail(xbox360, 4)
```

```
# Explore this category's dataframe
# check the dataframe dimensions (nrows, ncols)
# 275068 rows and 14 columns here
dim(xbox360)
```

```
275068 · 14
```

```
3169815 2017- Xbox 360 249 Thursday 8 1595 2017 Feb
```

```
## Provide basic stats
summary(xbox360)
```

```
  product_id           date               category             price        
Min.   :  43931   Min.   :2012-08-23   Length:275068      Min.   :  29.0  
1st Qu.: 932875   1st Qu.:2015-10-29   Class :character   1st Qu.: 179.0  
Median :1619816   Median :2016-05-21   Mode  :character   Median : 257.0  
Mean   :1693781   Mean   :2016-03-20                      Mean   : 307.4  
3rd Qu.:2524808   3rd Qu.:2016-10-30                      3rd Qu.: 399.0  
Max.   :3730337   Max.   :2017-02-25                      Max.   :4062.0  
  weekday               week         store_id          year         
Length:275068      Min.   : 1.0   Min.   :    1   Min.   :2012  
Class :character   1st Qu.:10.0   1st Qu.:  127   1st Qu.:2015  
Mode  :character   Median :28.0   Median :  578   Median :2016  
                   Mean   :26.7   Mean   : 4164   Mean   :2016  
                   3rd Qu.:42.0   3rd Qu.: 7088   3rd Qu.:2016  
                   Max.   :53.0   Max.   :28603   Max.   :2017  
   month           cpi_adjusted_price log_of_cpi_adjusted_price
Length:275068      Min.   :  29.0     Min.   :3.367            
Class :character   1st Qu.: 182.8     1st Qu.:5.208            
Mode  :character   Median : 260.0     Median :5.561            
                   Mean   : 311.2     Mean   :5.594            
                   3rd Qu.: 407.3     3rd Qu.:6.010            
                   Max.   :4179.4     Max.   :8.338            
number_of_store_per_product_and_day total_days_for_product
Min.   : 3.000                      Min.   : 3004         
1st Qu.: 4.000                      1st Qu.: 3521         
Median : 6.000                      Median : 4092         
Mean   : 6.734                      Mean   : 4770         
3rd Qu.: 8.000                      3rd Qu.: 5400         
Max.   :24.000                      Max.   :10152         
total_products_in_category
Min.   :64                
1st Qu.:64                
Median :64                
Mean   :64                
3rd Qu.:64                
Max.   :64                
```

```
library(psych)
```

```
describe(xbox360)
```

```
Error in describe(xbox360): object 'xbox360' not found
Traceback:
```

```
# save it into a dataframe for later visualization
dsb <- describe(xbox360)
```

```
head(dsb,3)
```

```
Error in describe(xbox360): could not find function "describe"
Traceback:
```

SEARCH STACK OVERFLOW

```
str(xbox360)
```

```
Classes ‘data.table’ and 'data.frame':  275068 obs. of  14 variables:
 $ product_id                        : int  1208489 1208489 1208489 1208489 1208489
 $ date                              : IDate, format: "2015-12-04" "2015-12-05" ...
 $ category                          : chr  "Xbox 360" "Xbox 360" "Xbox 360" "Xbox
 $ price                             : int  443 443 443 443 443 443 443 443 443 443
 $ weekday                           : chr  "Friday" "Saturday" "Sunday" "Monday" .
 $ week                              : int  49 49 49 50 50 50 50 50 50 50 ...
 $ store_id                          : int  6843 6843 6843 6843 6843 6843 6843 6843
 $ year                              : int  2015 2015 2015 2015 2015 2015 2015 2015
 $ month                             : chr  "Dec" "Dec" "Dec" "Dec" ...
 $ cpi_adjusted_price                : num  451 451 451 451 451 ...
 $ log_of_cpi_adjusted_price         : num  6.11 6.11 6.11 6.11 6.11 ...
 $ number_of_store_per_product_and_day: int  6 6 6 6 6 6 6 6 6 6 ...
 $ total_days_for_product            : int  7216 7216 7216 7216 7216 7216 7216 7216
 $ total_products_in_category        : int  64 64 64 64 64 64 64 64 64 64 ...
 - attr(*, ".internal.selfref")=<externalptr>
```

```
#install.packages("qwarps2")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Warning message:
“package ‘qwarps2’ 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”
```

```
#library(qwraps2)
```

```
#summary_table(xbox360)
```

```
Error in summary_table(xbox360): could not find function "summary_table"
Traceback:
```

SEARCH STACK OVERFLOW

```
# select a fraction of a dataframe for quicker analysis of trends and patterns
# 30% of the whole dataframe here
sub_xbox360 <- sample_frac(xbox360, 0.3)
```

```
dim(sub_xbox360)
```

```
82520 · 14
```

```
str(xbox360)
```

```
Classes ‘data.table’ and 'data.frame':  0 obs. of  14 variables:
 $ product_id                          : int
 $ date                                : 'IDate' int(0)
 $ category                            : chr
 $ price                               : int
 $ weekday                             : chr
 $ week                                : int
 $ store_id                            : int
 $ year                                : int
 $ month_numerical                     : int
 $ cpi_adjusted_price                  : num
 $ log_of_cpi_adjusted_price           : num
 $ number_of_store_per_product_and_day: int
 $ total_days_for_product              : int
 $ total_products_in_category          : int
 - attr(*, ".internal.selfref")=<externalptr>
```

Double-click (or enter) to edit

```
# Quick visualization to have a feel of the data
#plot(xbox360) # takes too long to load for 275068 rows

# Thus let's split the dataset into training and tesing data
# Install needed packages
install.packages("caTools")
```

```
library(caTools)
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependency ‘bitops’
```

```
set.seed(123) # to shuffle the data in a way defined by the seed value
# so that we always have the same dataset everytime seeding data

sample xbox360 = sample.split(xbox360, SplitRatio = .70)
```

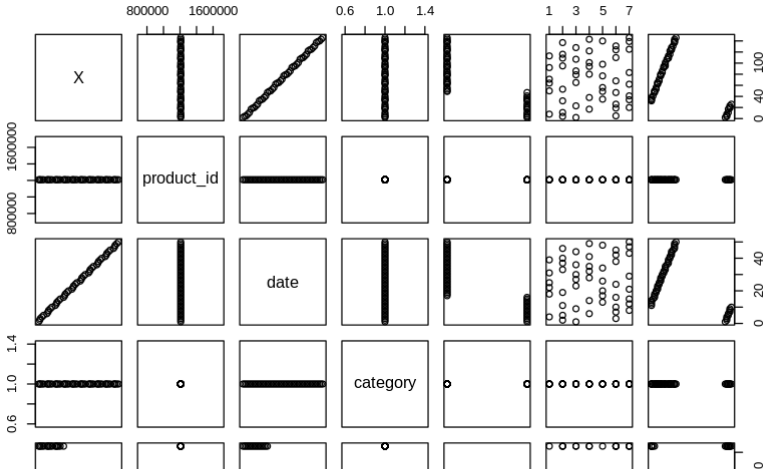
```
# Also build train and test datasets for later predictive analytics
train_xbox360 = subset(xbox360, sample_xbox360 == TRUE)
test_xbox360 = subset(xbox360, sample_xbox360 == FALSE)


#write.csv(train_xbox360, "train_xbox360.csv")
#write.csv(test_xbox360, "test_xbox360.csv")


head(train_xbox360,3)
```

| | X | product_id | date | category | price | weekday | week | store_id | year | mor |
|---|---|---|---|---|---|---|---|---|---|---|
| | <int> | <int> | <chr> | <chr> | <int> | <chr> | <int> | <int> | <int> | <ch |
| **1** | 1 | 1208489 | 2015-12-04 | Xbox 360 | 443 | Friday | 49 | 6843 | 2015 | D |
| **3** | 3 | 1208489 | 2015-12-06 | Xbox 360 | 443 | Sunday | 49 | 6843 | 2015 | D |
| **6** | 6 | 1208489 | 2015-12-09 | Xbox 360 | 443 | Wednesday | 50 | 6843 | 2015 | D |

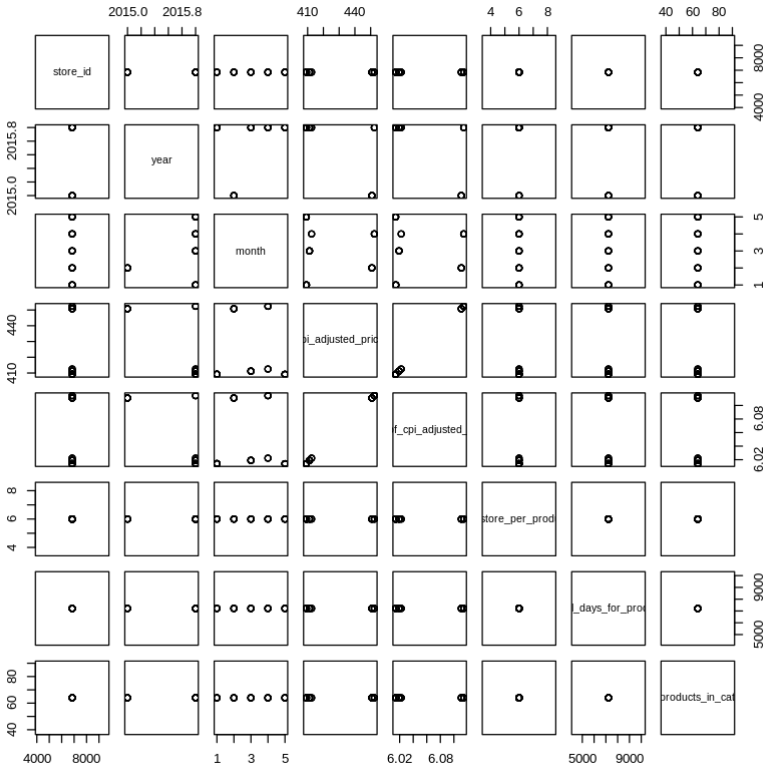
```
head(test_xbox360,3)
```

| | X | product_id | date | category | price | weekday | week | store_id | year | month |
|---|---|---|---|---|---|---|---|---|---|---|
| | <int> | <int> | <chr> | <chr> | <int> | <chr> | <int> | <int> | <int> | <chr> |
| **2** | 2 | 1208489 | 2015-12-05 | Xbox 360 | 443 | Saturday | 49 | 6843 | 2015 | Dec |
| **4** | 4 | 1208489 | 2015-12-07 | Xbox 360 | 443 | Monday | 50 | 6843 | 2015 | Dec |
| **5** | 5 | 1208489 | 2015-12-08 | Xbox 360 | 443 | Tuesday | 50 | 6843 | 2015 | Dec |

```
# Let's now plot this data subset, here the first 50 rows & 7 columns
plot(test_xbox360[1:50, 1:7])
```




```
# Now plot this for the following 7 columns
plot(test_xbox360[1:50, 8:15])
```




```
#?geom_point(mapping = )
```

```
# Save the xbox 360 dataframe as a csv file for possible later use
#write.csv(xbox360, "xbox360.csv")
```

```
# Now let's subset the dataframe on columns of interest
#sub_xbox360 <- subset(xbox360, select = c("date","price","week","store_id", "year", "mont
```

```
#tail(sub_xbox360, 4)
```

A data.frame: 4 × 6

| | date | price | week | store_id | year | month |
|---|---|---|---|---|---|---|
| | <chr> | <int> | <int> | <int> | <int> | <chr> |
| **275065** | 2017-02-22 | 249 | 8 | 1595 | 2017 | Feb |
| **275066** | 2017-02-23 | 249 | 8 | 1595 | 2017 | Feb |
| **275067** | 2017-02-24 | 249 | 8 | 1595 | 2017 | Feb |

```
# Save the subset dataframe as a csv file for possible later use
#write.csv(sub_xbox360, "sub_xbox360.csv")
```

## Visualizations

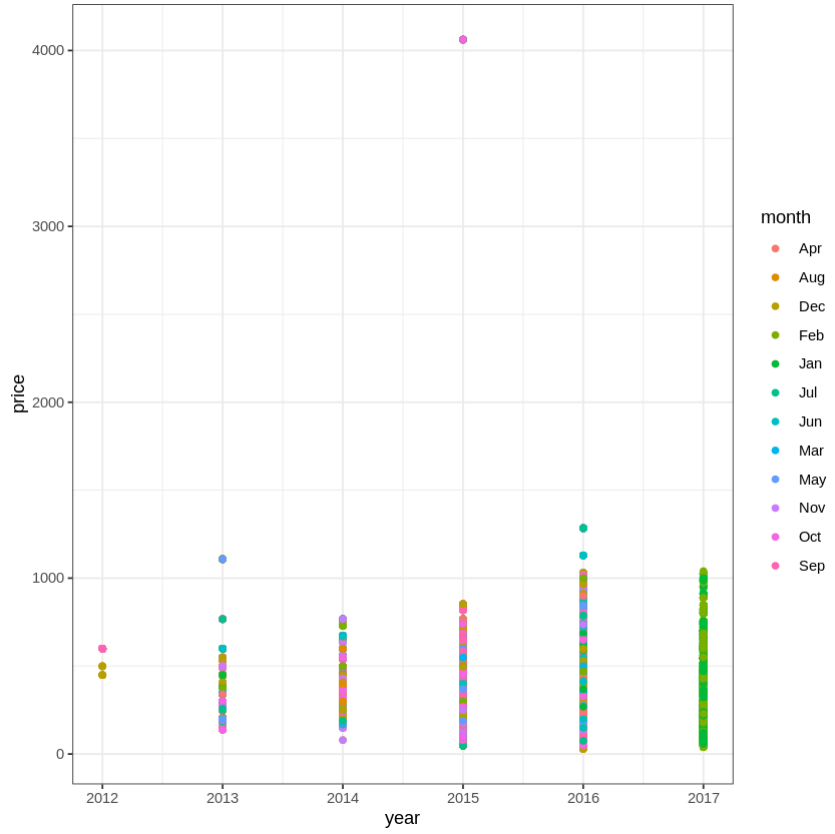
```
library(ggplot2)
```

```
Attaching package: ‘ggplot2’


The following objects are masked from ‘package:psych’:

    %+%, alpha
```

```
#plot(sub_xbox360$year, sub_headphones$price, type = "l", xlab = "year", ylab = "price")


# PLOTS RESULTS ##########################################
# Let's explore the data by creating a line plot of the time series dataframe
# coloured by week

plot1 <- ggplot(xbox360, aes(year, price, colour = month))

plot1 + geom_point() +
theme_bw()
#ggsave("plot1.png",width=6, height=4,dpi=300)
```

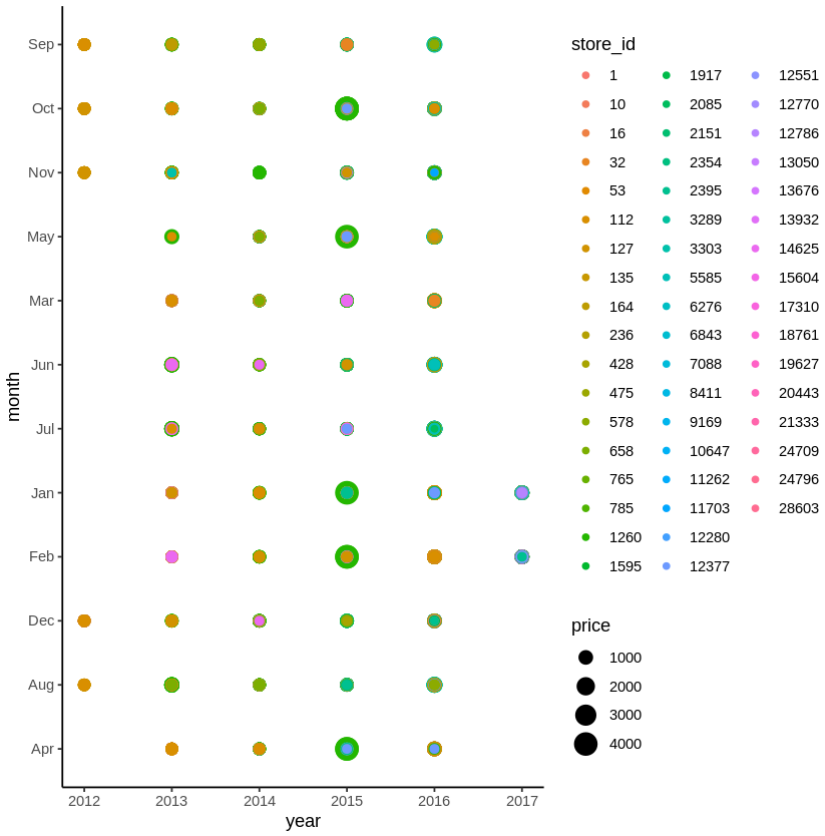
```
Warning message:
“Removed 1 rows containing missing values (geom_point).”
```


```
# Let's use a subset of xbox360 dataframe
# it will make the analysis and visualization faster
# while the df will be keeping the same behavious
plot01 <- ggplot(sub_xbox360, aes(year, price, colour = month))

plot01 + geom_point() +
theme_bw()
#ggsave("plot1.png",width=6, height=4,dpi=300)
```




```
# PLOTS RESULTS ##############################################
# Let's explore the data by creating a line plot of the time series dataframe
# coloured by week

# First, let's convert store_id
# from int (continuous variable) to factor (categorical)
sub_xbox360$store_id <- as.factor(sub_xbox360$store_id)
```

```
# plot results for sub_xbox360 instead of the whole dataset
plot2 <- ggplot(sub_xbox360, aes(year, month, size = price, colour = store_id))


plot2 +
geom_point() +
theme_classic() +
ggsave("plot2.png",width=6, height=4,dpi=300)
```



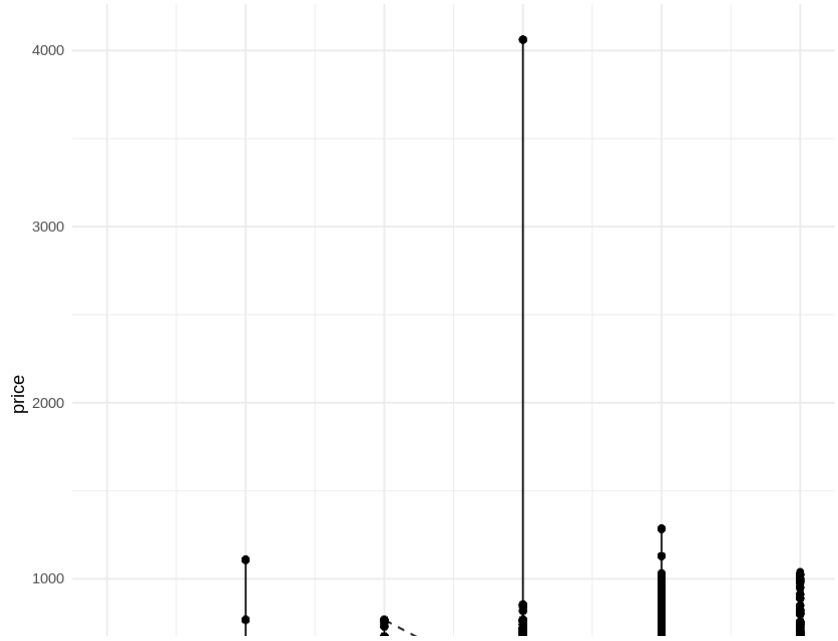

```
# Create scatter Plot
plot3 <- ggplot(sub_xbox360 , aes(x = year , y = price))

plot3 +
       geom_point(aes(color = store_id, size = price)) +
       facet_wrap(~month) +
       #geom_text( aes(label = row.names(sub_xbox360)) , hjust = 1 , vjust = -1.5 ,size =
       ylab("Price") +
       xlab("Year") +
       theme(
         panel.background = element_rect(fill = "transparent"), # bg of the panel
         plot.background = element_rect(fill = "transparent", color = NA), # bg of the plo
         panel.grid.major = element_blank(), # get rid of major grid
         panel.grid.minor = element_blank(), # get rid of minor grid
         #legend.background = element_rect(fill = "transparent"), # get rid of legend bg
         #legend.box.background = element_rect(fill = "transparent"), # get rid of legend
         axis.text.x = element_text(angle = 70)
         ) +
       ggsave("plot3.png",width=6, height=4,dpi=300)
```

```
ggsave("plot3.png",width=6, height=4,dpi=300)

# Run and display plot
```

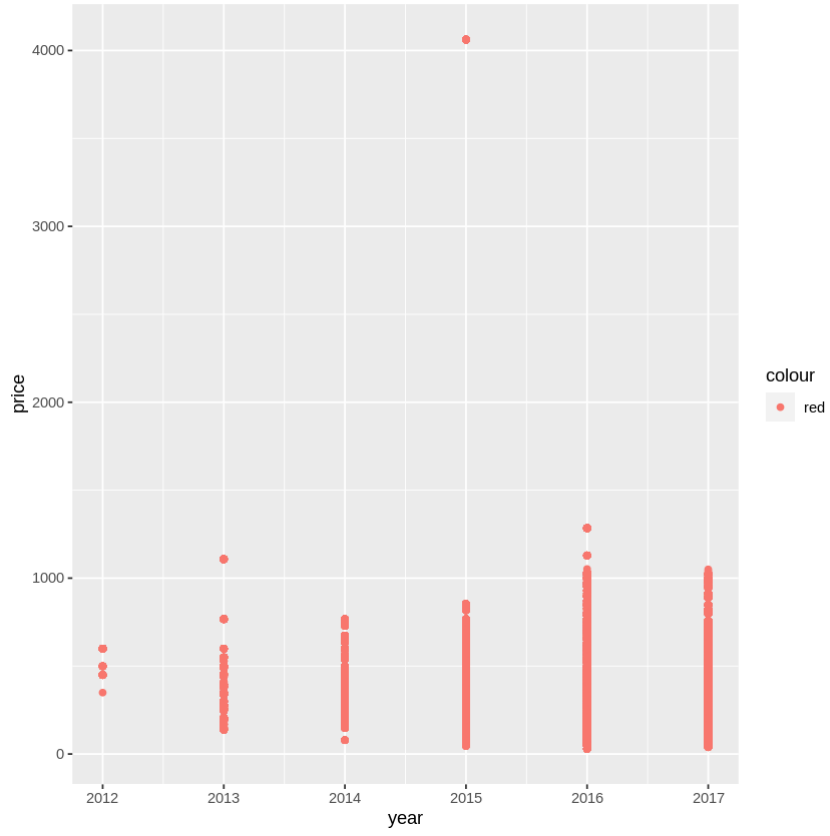

We identify already here a few data points outliers over a few months. Let's explore it a bit further later on.

```
plot02 <- ggplot(xbox360, aes(x = year, y = price, group = 1))

plot02 + geom_line(linetype = "dashed") +
geom_point() +
theme_minimal() +
ggsave("plot02.png",width=6, height=4,dpi=300)
```

```
plot002 <- ggplot(sub_xbox360, aes(x = year, y = price, group = 1))

plot002 + geom_line(linetype = "dashed") +
geom_point() +
theme_minimal() +
ggsave("plot02.png",width=6, height=4,dpi=300)
```



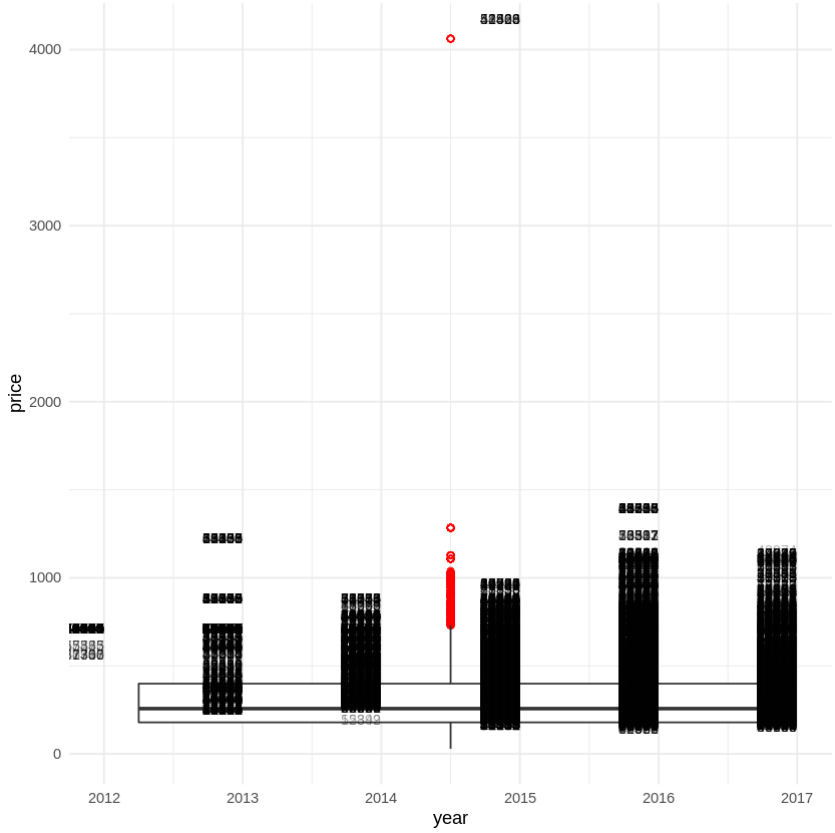

```
plot3 <- ggplot(sub_xbox360, aes(x = year, y = price, colour = "red"))

plot3 + geom_point()
```




There seeems to be outliers price wise above 4000, most of which being priced below 1500. Let's use box plot to illustrate it better.

```
# Let's also attempt to label data including outliers
# in order to identify relevant related information such as store_id
bxp1 <- ggplot(sub_xbox360, aes(x = year, y = price, group = 1))
```

```
bxp1 +
geom_boxplot(outlier.colour = "red", outlier.shape = 1) +
geom_text( aes(label = row.names(sub_xbox360)) , hjust = 1 , vjust = -1.5 ,size = 3, alpha
theme_minimal() +
ggsave("bxp1.png",width=6, height=4,dpi=300)
```




Let's visualize the distribution of the data by combining box and violin plots for better visualization of the outliers. To justify possibly to dismiss them if their effect is not significant.

```
install.packages("gridExtra")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

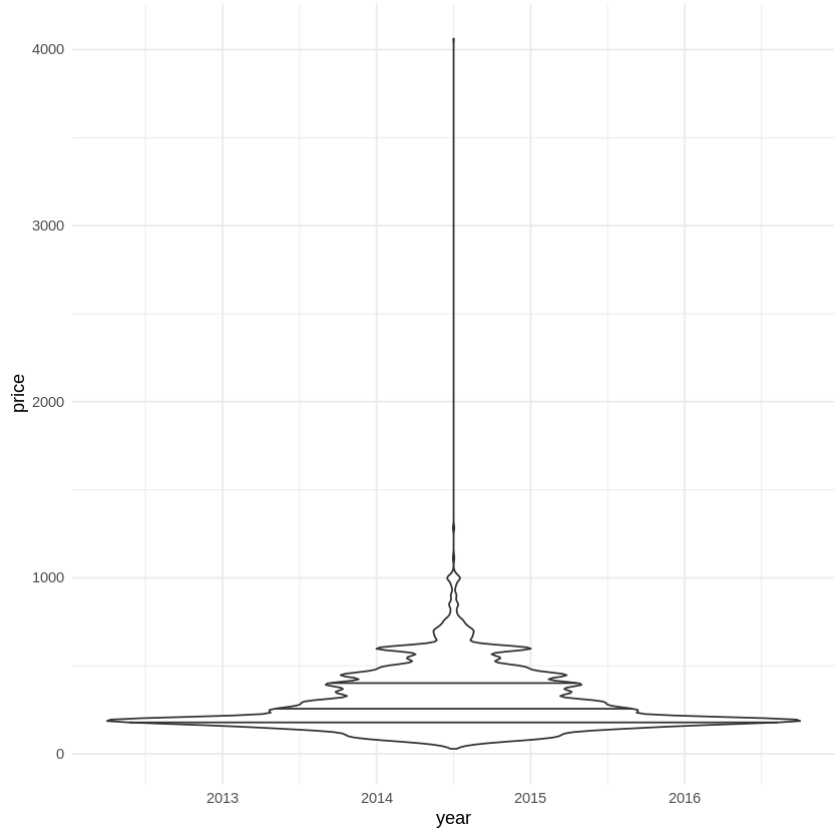
```
library(gridExtra)
```

```
Attaching package: ‘gridExtra’


The following object is masked from ‘package:dplyr’:

    combine
```

```
vp1 <- ggplot(sub_xbox360, aes(x = year, y = price))

# add horizontal lines at Q1, Q2, and Q3
```

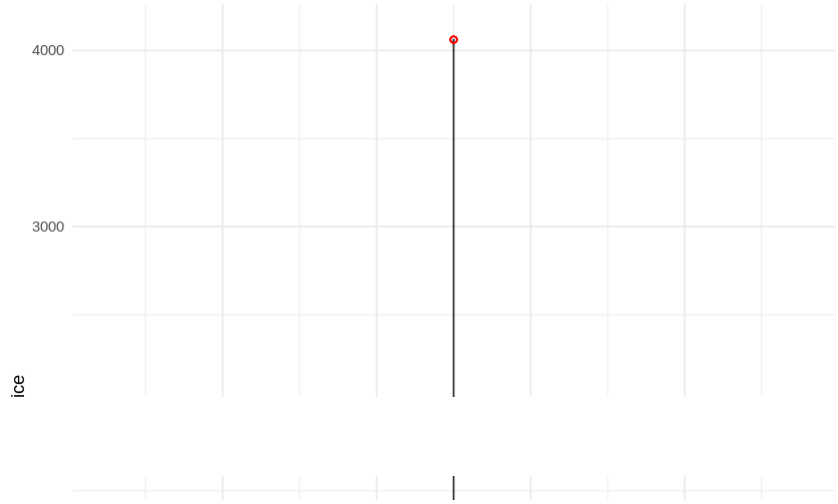
```
vp1 +
geom_violin(draw_quantiles = c(0.25, 0.5, 0.75)) +
#geom_boxplot(outlier.colour = "red", outlier.shape = 1) +
#ggtitle("Violin plot") +
theme_minimal() +
ggsave("vp1.png",width=6, height=4,dpi=300)
```

```
Warning message in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
“collapsing to unique 'x' values”
Warning message in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
“collapsing to unique 'x' values”
```




```
# combine box and violin plots with outliers highlighted
vp1 +
geom_violin(draw_quantiles = c(0.25, 0.5, 0.75)) +
geom_boxplot(outlier.colour = "red", outlier.shape = 1) +
#ggtitle("Violin plot") +
theme_minimal() +
ggsave("vp2.png",width=6, height=4,dpi=300)
```

```
Warning message:
“Continuous x aesthetic -- did you forget aes(group=...)?”
Warning message in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
“collapsing to unique 'x' values”
Warning message:
“Continuous x aesthetic -- did you forget aes(group=...)?”
Warning message in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
“collapsing to unique 'x' values”
```




```
# Let's plot them using boxplots
# with possible outliers in red
# and removes all data points outside the given range

bxp1 +
geom_boxplot(outlier.colour = "red", outlier.shape = 1) +
scale_y_continuous(limits = c(0, 1500)) +
theme_minimal() +
ggsave("bxp01.png",width=6, height=4,dpi=300)
```

```
Warning message:
“Removed 12 rows containing non-finite values (stat_boxplot).”
Warning message:
```

Interpretation of the boxplot:

```
The two 'hinges' or extremities of the box represent the first and third quartile, i.e., close to
Boxplot statistical values are represented mainly via the vector *stats* of are a vector of lengt
```

source

```
tail(sub_xbox360,3)
```

| V1 <int> | product_id <int> | date <date> | category <chr> | price <int> | weekday <chr> | week <int> | store_id <fct> | year <int> | month <chr> |
|---|---|---|---|---|---|---|---|---|---|
| 242908 | 2641145 | 2017-02-20 | Xbox 360 | 249 | Monday | 8 | 112 | 2017 | Feb |
| 153541 | 3152067 | 2015-11-14 | Xbox 360 | 479 | Saturday | 46 | 164 | 2015 | Nov |
| 65803 | 43931 | 2015-12-26 | Xbox 360 | 179 | Saturday | 52 | 127 | 2015 | Dec |

## Mahalanobis Distance

source

It is a measure of the distance between a point P and a distribution D; it is a multi-dimensional generalization of the measure of how many standard deviations is P away from the mean of D. That distance (MD) grows as P moves away from the mean of D along each principal component axis*.
Mahalanobis distance can also be defined as a dissimilarity measure between two random vectors {\displaystyle {\vec {x}}}{\vec {x}} and {\displaystyle {\vec {y}}}{\vec {y}} of the same distribution with the covariance matrix S. Then, here is a quick reminder of what covariance matrix is.

Also called dispersion matrix, variance matrix, or variance–covariance matrix, it is a square matrix giving the covariance between each pair of elements of a given random vector. Any covariance matrix is symmetric and positive semi-definite and its main diagonal contains

variances (i.e., the covariance of each element with itself). Intuitively, the covariance matrix generalizes the notion of variance to multiple dimensions.

source

- source A sequence of {\displaystyle p}p direction vectors, where the {\displaystyle i^{\text{th}}}i^{{\text{th}}} vector is the direction of a line that best fits the data while being orthogonal to the first {\displaystyle i-1}i-1 vectors. Here, a best-fitting line is defined as one that minimizes the average squared distance from the points to the line.

Note that if the covariance matrix is the identity matrix, the Mahalanobis distance reduces to the Euclidean distance.

```
# To demonstrate why we can remove this outliers
# Compute mahalonobis distance and flag outliers if any
# [source](https://www.youtube.com/watch?v=BdEOIQ2ozYM&t=223s)

# First, let's calculate mahalanobis distance with height and weight distribution

Sx <- cov(sub_xbox360[, c(4,8)])
MD <- mahalanobis(sub_xbox360[, c(4,8)], colMeans(sub_xbox360[, c(4,8)]), Sx)



# covariance matrix for price and year, with the variances in its diagonal
Sx
```

A matrix: 2 × 2 of type dbl

| | price | year |
|---|---|---|
| **price** | 30194.23874 | -20.2409830 |
| **year** | -20.24098 | 0.7107676 |

Let's explore the first 100 MD data while rounding it up to 2 decimals

```
#head(MD)
```

1.46343045772125 · 0.602163062952911 · 0.194938685534366 · 5.00209748932361 · 1.46343045772125 · 1.91238182651806

```
MD[1:300] %>% round(2)
```

1.46 · 0.6 · 0.19 · 5 · 1.46 · 1.91 · 0.3 · 0.86 · 0.29 · 2.74 · 0.15 · 1 · 2.63 · 0.2 · 1.19 · 1.64 · 0.75 · 0.6 · 4.26 · 0.99 · 0.96 · 2.44 · 0.84 · 0.91 · 1.46 · 1.41 · 11.62 · 0.55 · 0.6 · 2.57 · 0.13 · 7.21 · 2.37 · 0.14 · 8.55 · 1.28 · 4.49 · 0.44 · 1.47 · 1.75 · 0.46 · 3.07 · 0.15 · 20.01 · 0.6 · 0.45 · 0.13 · 0.19 · 0.2 · 1.47 · 4.86 · 10.96 · 1.75 · 0.13 · 1.81 · 0.35 · 1.47 · 0.91 · 1.76 · 1.47 · 0.19 · 4.49 · 1.07 · 2.36 · 4.06 · 1.57 · 1.33 · 2.85 · 0.45 · 0.11 · 0.45 · 1.28 · 0.87 · 0.94 · 0.66 · 4.3 · 4.9 · 0.78 · 0.19 · 3.21 · 1.21 · 0.44 ·

```
sub_xbox360$MD <- round(MD,3)
```

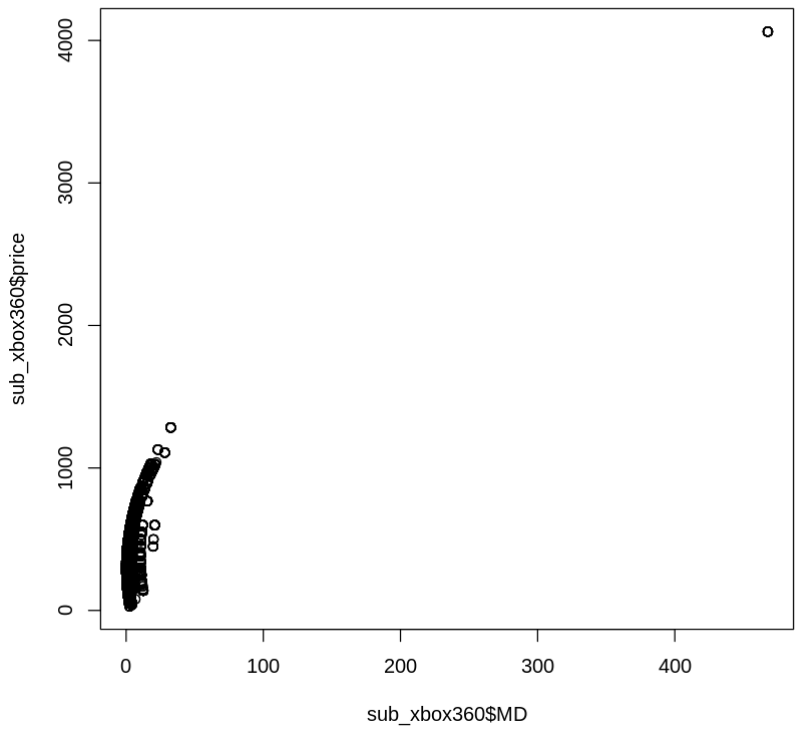
0.28 · 0.45 · 1.46 · 0.6 · 0.45 · 2.36 · 0.15 · 4.3 · 2.35 · 0.11 · 0.15 · 1.47 · 0.15 · 0.91 · 0.87 · 0.72 ·

```
head(sub_xbox360,3)
```

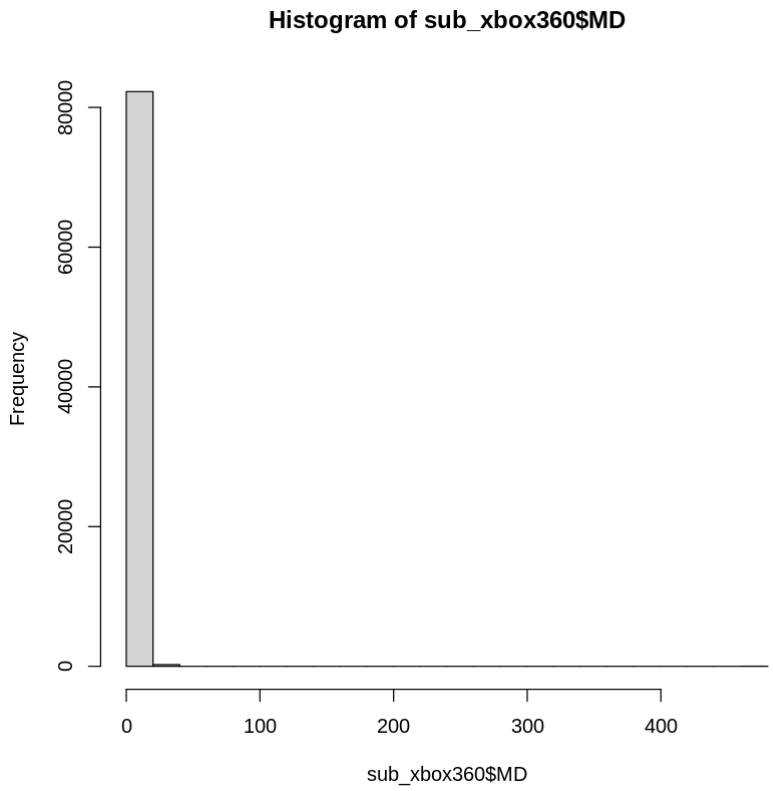
| product_id | date | category | price | weekday | week | store_id | year | month | cpi_a |
|---|---|---|---|---|---|---|---|---|---|
| <int> | <date> | <chr> | <int> | <chr> | <int> | <int> | <int> | <chr> | |
| 911420 | 2013-12-21 | Xbox 360 | 189 | Saturday | 51 | 112 | 2013 | Dec | |
| 1341635 | 2014-01-22 | Xbox 360 | 449 | Wednesday | 4 | 112 | 2014 | Jan | |
| 3186029 | 2015-10-23 | Xbox 360 | 445 | Friday | 43 | 112 | 2015 | Oct | |

We now want to flag outliers (price data points), and for that we need to get a sense of the data distribution of MDs in order to know what could be a reasonable threashold. Let's then do quick plots.

```
plot(sub_xbox360$MD, sub_xbox360$price)
```



```
hist(sub_xbox360$MD)
```




```
hist_md <- ggplot(sub_xbox360, aes(x = MD))

# add horizontal lines at Q1, Q2, and Q3
hist_md +
geom_histogram(binwidth = 5, aes(fill = price)) +
#ggtitle("MD histogram") +
theme_minimal()
ggsave("hist_md.png",width=6, height=4,dpi=300)
```


```
# Let's create a new column where outliers will be flagged
# called outlier_MD
sub_xbox360$outlier_MD <- FALSE
sub_xbox360$outlier_MD[sub_xbox360$MD > 50] <- TRUE
```

```
tail(sub_xbox360)
```

6 × 16

| rice | number_of_store_per_product_and_day | total_days_for_product | total_products_in_ |
|---|---|---|---|
| dbl> | <int> | <int> | |
| 9338 | 6 | 3521 | |
| 4103 | 10 | 6911 | |
| 5570 | 9 | 8106 | |
| 3974 | 9 | 6911 | |
| 5893 | 6 | 4227 | |
| 6333 | 4 | 3101 | |

```
#save as a csv file with the new MD column
# MD enables to identify unusual data in the df#
write.csv(sub_xbox360, "sub_xbox60.csv")
```

## Identify the products that are outliers

```
df_outliers <- sub_xbox360 %>% group_by(price) %>% filter(outlier_MD == "TRUE")
```

```
head(df_outliers)
```

| product_id | date | category | price | weekday | week | store_id | year | month | cpi_a |
|---|---|---|---|---|---|---|---|---|---|
| <int> | <date> | <chr> | <int> | <chr> | <int> | <fct> | <int> | <chr> | |
| 2687705 | 2015-10-06 | Xbox 360 | 4062 | Tuesday | 41 | 1260 | 2015 | Oct | |
| 2687705 | 2015-04-29 | Xbox 360 | 4062 | Wednesday | 18 | 1260 | 2015 | Apr | |
| 2687705 | 2015-10-04 | Xbox 360 | 4062 | Sunday | 40 | 1260 | 2015 | Oct | |
| 2687705 | 2015-02-17 | Xbox 360 | 4062 | Tuesday | 8 | 1260 | 2015 | Feb | |
| 2687705 | 2015-04-26 | Xbox 360 | 4062 | Sunday | 17 | 1260 | 2015 | Apr | |

```
dim(df_outliers)
```

12 · 16

There are twelve outliers data points identified. We can save them as a separate dataframe if interested to analyze them further later on.

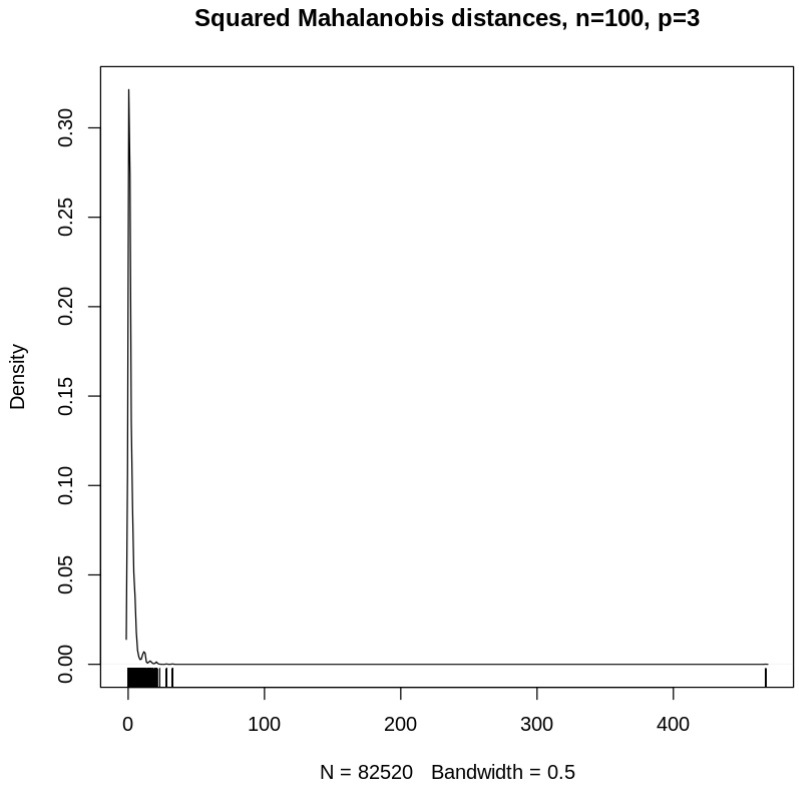
```
# save as a csv file
write.csv(df_outliers, "df_outliers.csv")
```

```
ggplot(MD, aes(price)) +
geom_density(kernel = "gaussian")
```

```
Error: `data` must be a data frame, or other object coercible by `fortify()`, not a
numeric vector
Traceback:

1. ggplot(MD, aes(price))
2. ggplot.default(MD, aes(price))
3. fortify(data, ...)
4. fortify.default(data, ...)
5. abort(msg)
6. signal_abort(cnd)
```

SEARCH STACK OVERFLOW

```
plot(density(MD, bw = 0.5),
     main="Squared Mahalanobis distances, n=100, p=3") ; rug(MD)
```

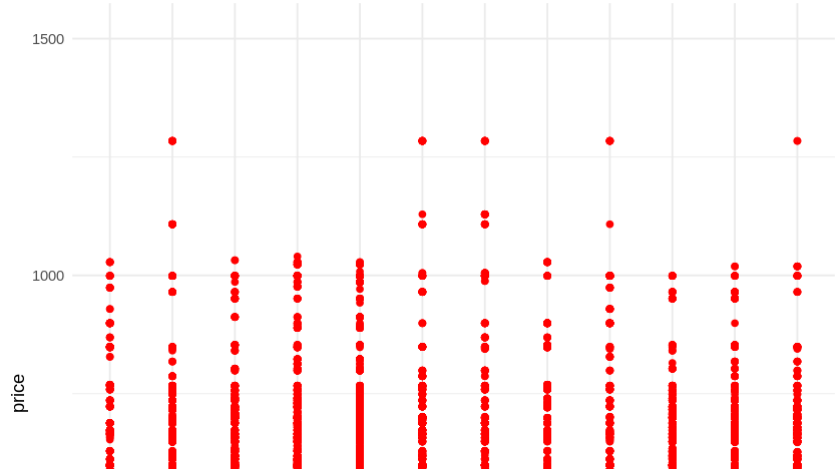



```
#qqplot(qchisq(ppoints(100), df = 3), D2,
#       main = expression("Q-Q plot of Mahalanobis" * ~D^2 *
#                         "vs. quantiles of" * ~ chi[3]^2))
#abline(0, 1, col = 'gray')
```

Now that we've decided not to consider outliers beyond a 1500 price in later visualizations, let's look at data in more granular way by zooming into months.

```
plot4 <- ggplot(sub_xbox360, aes(x = month, y = price))

plot4 + geom_point(colour = "red") +
scale_y_continuous(limits = c(0, 1500)) +
theme_minimal() +
ggsave("plot4.png",width=6, height=4,dpi=300)
```

```
Warning message:
“Removed 12 rows containing missing values (geom_point).”
Warning message:
“Removed 12 rows containing missing values (geom_point).”
```




```
# let's zoom in deeper
plot04 <- ggplot(subset(sub_xbox360[,], aes(x = year, y = price))

plot04 +
#geom_histogram(kernel = "gaussian") +
#scale_x_continuous(limits = c(2012, 2015)) +
#scale_y_continuous(limits = c(0, 1500)) +
theme_minimal()
```

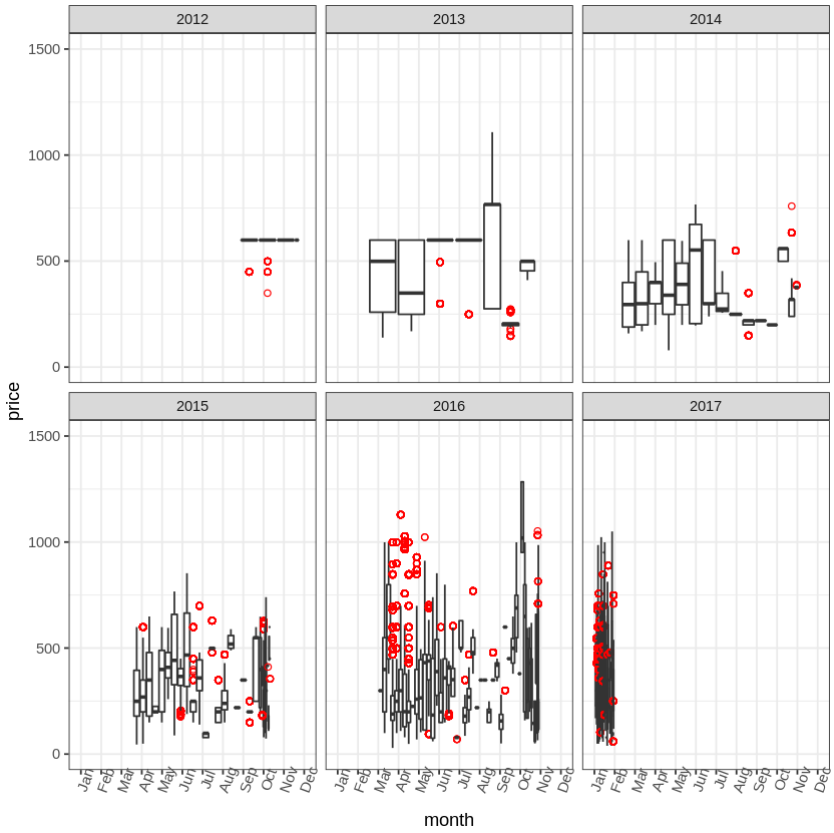
```
Error in parse(text = x, srcfile = src): <text>:4:1: unexpected symbol
3:
4: plot04
   ^
Traceback:
```

SEARCH STACK OVERFLOW

```
?boxplot.stats


# identify the outliers
#sub_xbox360$store_id[!sub_xbox360$store_id %in% boxplot.stats(sub_xbox360$store_id)$out]


# check Mahalanobis distance for determining outliners
# and explaining why
# source: https://towardsdatascience.com/mahalonobis-distance-and-outlier-detection-in-r-c
# https://www.datacamp.com/community/tutorials/pca-analysis-r


sub_xbox360$store_id <- as.factor(sub_xbox360$store_id)


# There seeems to be outliners.
# Let's plot by year over the months
```

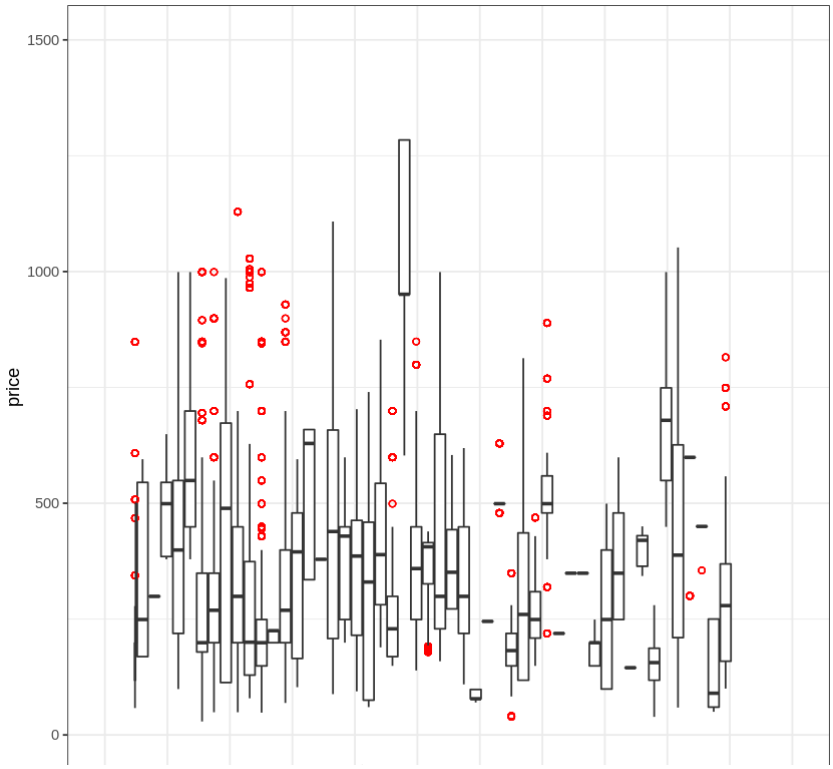
```
#sub_xbox360$store_id <- as.factor(sub_xbox360$store_id)

bxp2 <- ggplot(data = xbox360, mapping = aes(x = month, y = price, group = store_id))

bxp2 + geom_boxplot(outlier.colour = "red", outlier.shape = 1) +
scale_y_continuous(limits = c(0, 1500)) +
scale_x_discrete(limits = month.abb) +
facet_wrap(~year) +
theme_bw() +
theme(axis.text.x = element_text(angle = 70)) +
ggsave("bxp2.png",width=6, height=4,dpi=300)
```

```
Warning message:
“Removed 38 rows containing non-finite values (stat_boxplot).”
Warning message:
“Removed 38 rows containing non-finite values (stat_boxplot).”
```



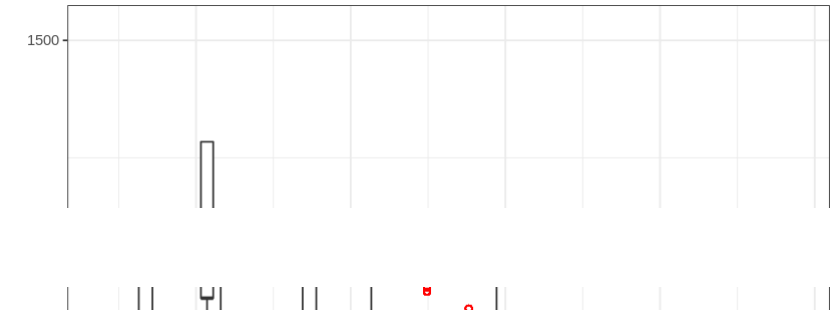

```
# Let's see if there is a pattern, cyclical behaviour
# Let's plot by month over cumulated years

#bxp2 <- ggplot(data = xbox360, mapping = aes(x = month, y = price, group = store_id))

bxp2 + geom_boxplot(outlier.colour = "red", outlier.shape = 1) +
scale_y_continuous(limits = c(0, 1500)) +
scale_x_discrete(limits = month.abb) +
#facet_wrap(~year) +
theme_bw() +
theme(axis.text.x = element_text(angle = 70)) +
ggsave("bxp02.png",width=6, height=4,dpi=300)
```

```
Warning message:
“Removed 38 rows containing non-finite values (stat_boxplot).”
Warning message:
“Removed 38 rows containing non-finite values (stat_boxplot).”
```




```
# Let's see if there is a pattern, cyclical behaviour
# Let's plot by month over cumulated years

bxp3 <- ggplot(data = xbox360, mapping = aes(x = week, y = price, group = store_id))

bxp3 + geom_boxplot(outlier.colour = "red", outlier.shape = 1) +
scale_y_continuous(limits = c(0, 1500)) +
#scale_x_discrete(limits = month.abb) +
#facet_wrap(~year) +
theme_bw() +
theme(axis.text.x = element_text(angle = 70)) +
ggsave("bxp3.png",width=6, height=4,dpi=300)
```

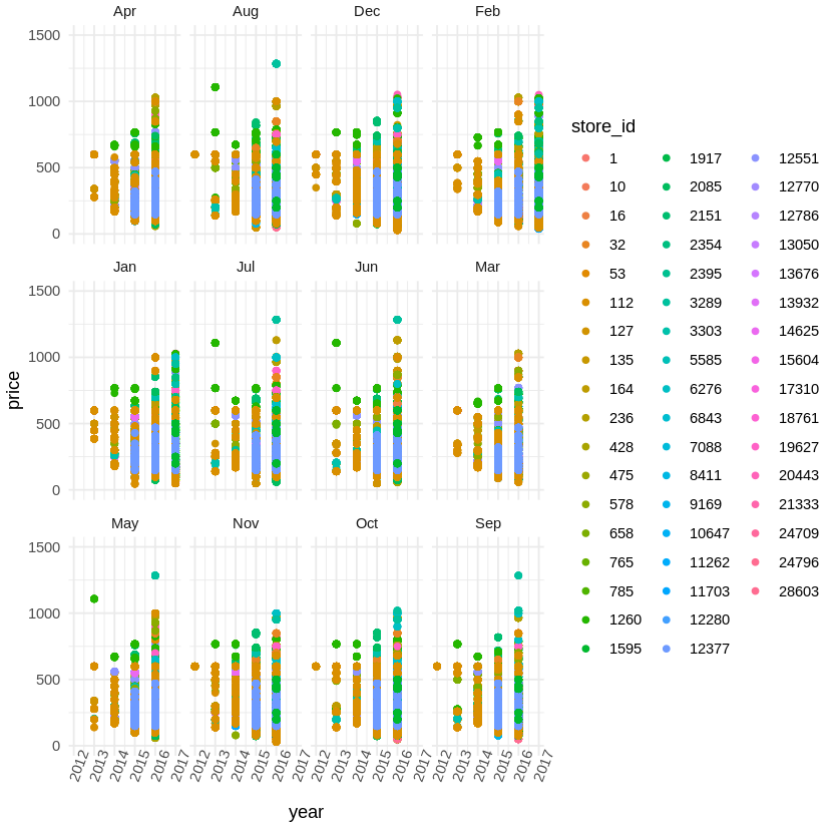
```
Warning message:
“Removed 38 rows containing non-finite values (stat_boxplot).”
Warning message:
“Removed 38 rows containing non-finite values (stat_boxplot).”
```




```
#?geom_boxplot
```

```
# Let's plot a
plot5 <- ggplot(data = xbox360)

plot5 +
  geom_point(mapping = aes(x = year, y = price, colour =store_id)) +
  scale_y_continuous(limits = c(0, 1500)) +
  facet_wrap(~month) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 70)) +
  ggsave("plot5.png",width=6, height=4,dpi=300)
```

```
Warning message:
“Removed 38 rows containing missing values (geom_point).”
Warning message:
“Removed 38 rows containing missing values (geom_point).”
```



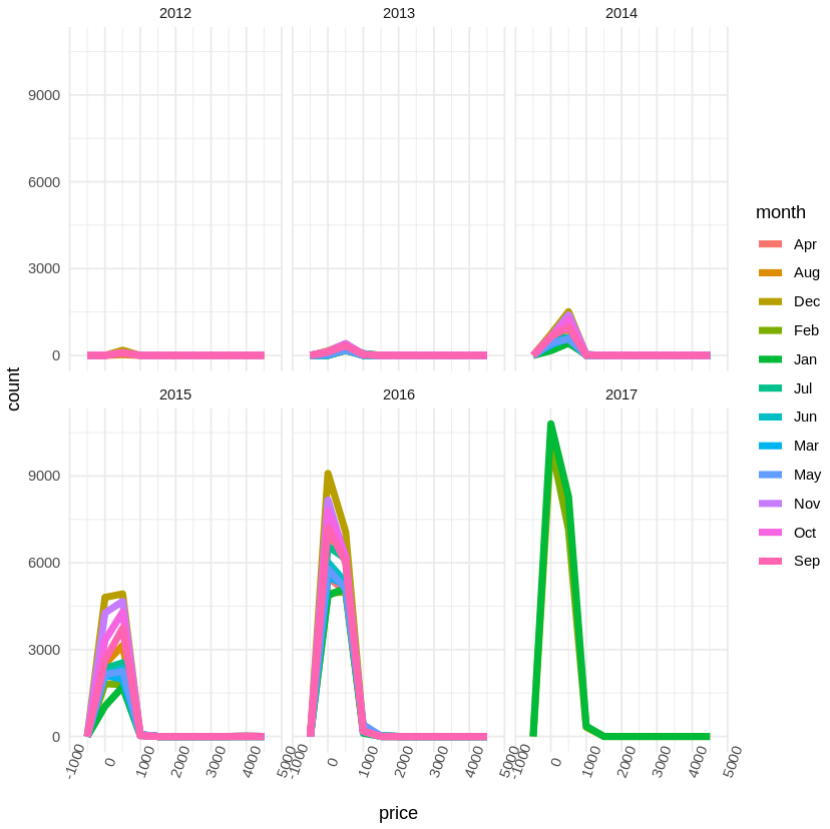

```
# Let's plot a
```

```
plot5 +
  #geom_col(mapping = aes(x = month, y = price)) +
  geom_bar(mapping = aes(x = month, fill = month)) +
  geom_line()+
  #scale_y_continuous(limits = c(0, 3000)) +
  scale_x_discrete(limits = month.abb) +
  facet_wrap(~year) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 70))
  #ggsave("plot05.png",width=6, height=4,dpi=300)
  #geom_smooth(method = lm)
```

```
Error in eval(expr, envir, enclos): object 'plot5' not found
Traceback:
```

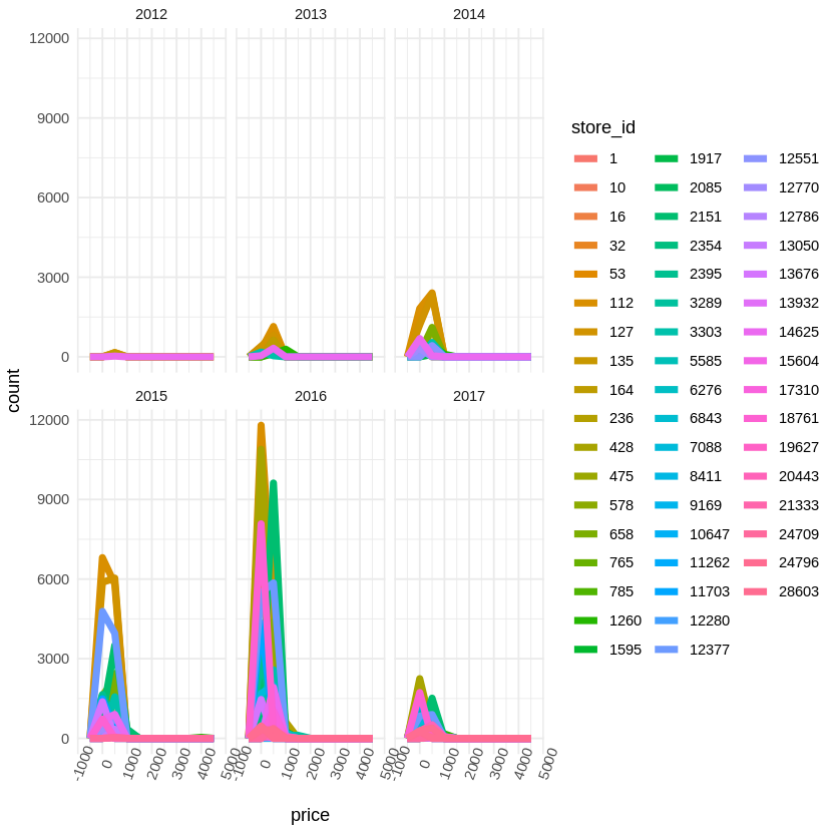
SEARCH STACK OVERFLOW

```
plot6 <- ggplot(xbox360, aes(price, colour = month))

plot6 + geom_freqpoly(binwidth = 500, size=2) +
facet_wrap(~year) +
theme_minimal() +
theme(axis.text.x = element_text(angle = 70)) +
ggsave("plot6.png",width=6, height=4,dpi=300)
```




```
plot7 <- ggplot(xbox360, aes(price, colour = store_id))

plot7 + geom_freqpoly(binwidth = 500, size=2) +
facet_wrap(~year) +
theme_minimal() +
theme(axis.text.x = element_text(angle = 70)) +
```

```
ggsave("plot7.png",width=6, height=4,dpi=300)
```



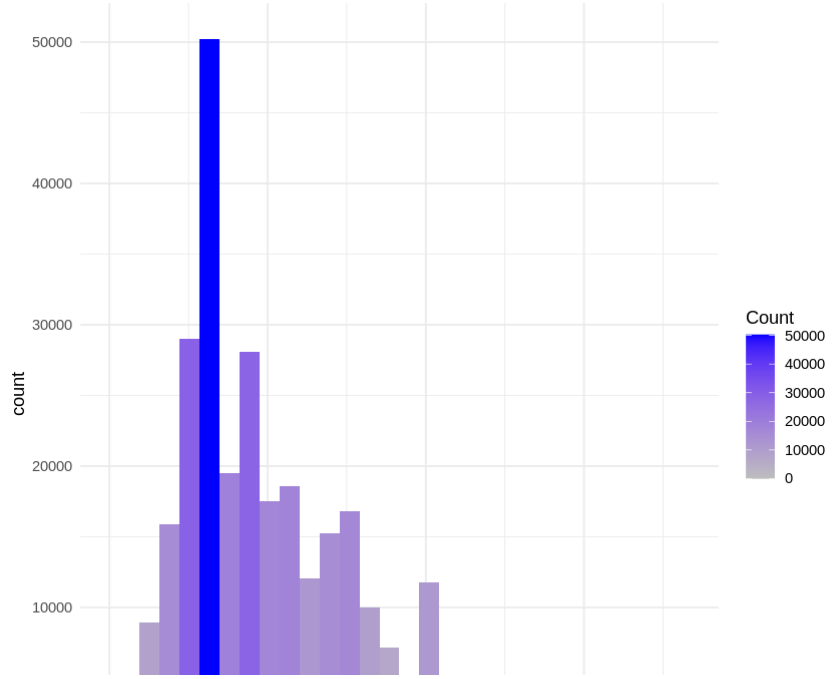

```
plot8 <- ggplot(xbox360, aes(price))

plot8 +
      geom_histogram(aes(fill=..count..), bins = 30) +
      scale_fill_gradient("Count", low = "grey", high = "blue")+
      geom_density(position = "stack") +
      theme_minimal() +
      ggsave("plot08.png",width=6, height=4,dpi=300)
```

```
Error in ggplot(xbox360, aes(price)): could not find function "ggplot"
Traceback:
```

SEARCH STACK OVERFLOW

```
# We observe that most data are located between price range of 0 and 1100
# Let's scale it dowm to it and observe the data distribution
plot8 +
      geom_histogram(aes(fill=..count..), bins = 30) +
      scale_fill_gradient("Count", low = "grey", high = "blue")+
      scale_x_continuous(limits = c(0, 1100)) +
      theme_minimal() +
      ggsave("plot008.png",width=6, height=4,dpi=300)
```

```
Warning message:
“Removed 281 rows containing non-finite values (stat_bin).”
Warning message:
“Removed 281 rows containing non-finite values (stat_density).”
Warning message:
“Removed 2 rows containing missing values (geom_bar).”
Warning message:
“Removed 281 rows containing non-finite values (stat_bin).”
Warning message:
“Removed 281 rows containing non-finite values (stat_density).”
Warning message:
“Removed 2 rows containing missing values (geom_bar).”
```




```
# Let's plot with autoplot function
install.packages("forecast")
install.packages("ffp2")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘xts’, ‘TTR’, ‘quadprog’, ‘quantmod’, ‘fracdiff’, ‘


Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Warning message:
“package ‘ffp2’ 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”
```

```
install.packages(c("quantmod", "xts", "tseries", "forecast", "timeseries"), dependencies =
```

```
Installing packages into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

```
Warning message:
“package ‘timeseries’ 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 message:
“Perhaps you meant ‘timeSeries’ ?”
also installing the dependencies ‘bit’, ‘bit64’, ‘plogr’, ‘tinytex’, ‘x13binary’, ‘R
```

```
library(forecast)
library(fpp2)
```

```
Error in library(fpp2): there is no package called ‘fpp2’
Traceback:

1. library(fpp2)
```

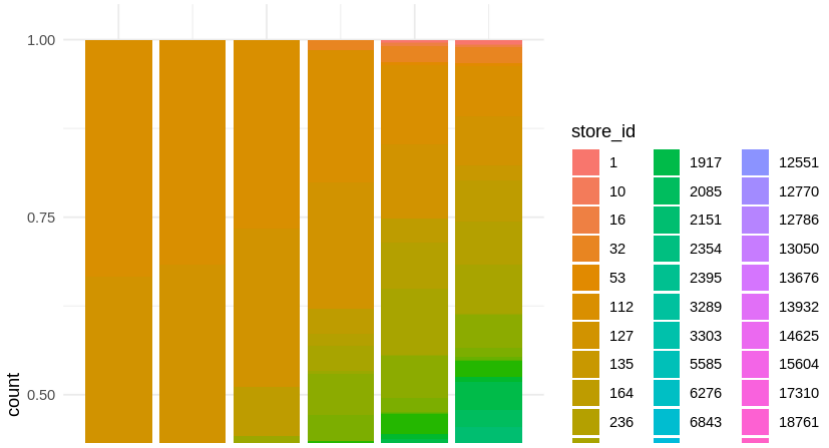
SEARCH STACK OVERFLOW

```
plot8 +
geom_smooth()
```

```
Error in eval(expr, envir, enclos): object 'plot8' not found
Traceback:
```

SEARCH STACK OVERFLOW

```
plot9 <- ggplot(xbox360, aes(year))

plot9 +
      geom_bar(mapping = aes(x = year, fill = store_id), position = "fill") +
      #scale_fill_gradient("Count", low = "grey", high = "blue") +
      theme_minimal() +
      ggsave("plot9.png",width=6, height=4,dpi=300)
```




```
plot8 +
      geom_bar(mapping = aes(x = year, fill = store_id), position = "fill") +
      #scale_fill_gradient("Count", low = "grey", high = "blue") +
      theme_minimal() +
      ggsave("plot008.png",width=6, height=4,dpi=300)
```

```
# Analyze possible correlation between features of the dataset
# install needed packages
install.packages("pps")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

```
predictive power scores (PPS)
```

## Clustering

```
tail(sub_xbox360,3)
```

| product_id | date | category | price | weekday | week | store_id | year | month | cpi_ad |
|---|---|---|---|---|---|---|---|---|---|
| <int> | <date> | <chr> | <int> | <chr> | <int> | <fct> | <int> | <chr> | |
| 471162 | 2016-06-13 | Xbox 360 | 149 | Monday | 24 | 11703 | 2016 | Jun | |
| 2715683 | 2016-06-28 | Xbox 360 | 414 | Tuesday | 26 | 5585 | 2016 | Jun | |
| 1180267 | 2015-11-19 | Xbox 360 | 249 | Thursday | 47 | 127 | 2015 | Nov | |

```
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
```

```
df10 <- sample_frac(sub_xbox360, 0.1)
```

```
head(df10,3)
```

| product_id | date | category | price | weekday | week | store_id | year | month | cpi_a |
|---|---|---|---|---|---|---|---|---|---|
| <int> | <date> | <chr> | <int> | <chr> | <int> | <int> | <int> | <chr> | |
| 2348284 | 2016-10-11 | Xbox 360 | 199 | Tuesday | 41 | 164 | 2016 | Oct | |
| 371134 | 2016-06-22 | Xbox 360 | 149 | Wednesday | 25 | 428 | 2016 | Jun | |
| 3152067 | 2016-09-16 | Xbox 360 | 542 | Friday | 37 | 2151 | 2016 | Sep | |

```
tail(sub_xbox360,3)
```

| V1 | product_id | date | category | price | weekday | week | store_id | year | month |
|---|---|---|---|---|---|---|---|---|---|
| <int> | <int> | <date> | <chr> | <int> | <chr> | <int> | <fct> | <int> | <chr> |
| 242908 | 2641145 | 2017-02-20 | Xbox 360 | 249 | Monday | 8 | 112 | 2017 | Feb |
| 153541 | 3152067 | 2015-11-14 | Xbox 360 | 479 | Saturday | 46 | 164 | 2015 | Nov |
| 65803 | 43931 | 2015-12-26 | Xbox 360 | 179 | Saturday | 52 | 127 | 2015 | Dec |

```
# Normalization
z <- df10[, c(4,8)]
```

```
z <- data[, c(1,3)]
m <- apply(z, 2, mean)
s <- apply(z, 2, sd)
z <- scale(z, m, s)



# Calculate the Euclidean distance
distance <- dist(z)
distance
print(distance, digits = 2)
```

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 2 | 0.273180048 | | | | |
| 3 | 1.874015130 | 2.147195178 | | | |
| 4 | 0.983448173 | 1.256628222 | 0.890566957 | | |
| 5 | 0.163908029 | 0.109272019 | 2.037923159 | 1.147356202 | |
| 6 | 1.304689087 | 1.440607617 | 1.779430288 | 1.262834212 | 1.381370721 |
| 7 | 0.622850510 | 0.349670462 | 2.496865640 | 1.606298683 | 0.458942481 |
| 8 | 1.226426845 | 1.185586234 | 2.490735742 | 1.759142021 | 1.194616002 |
| 9 | 1.244979455 | 1.354089262 | 1.904852915 | 1.328495607 | 1.304689087 |
| 10 | 1.982803905 | 2.207895126 | 1.218368248 | 1.330976196 | 2.116500532 |
| 11 | 1.120038197 | 1.393218246 | 0.753976933 | 0.136590024 | 1.283946226 |
| 12 | 1.215866604 | 1.304689087 | 1.991576566 | 1.381370721 | 1.262834212 |
| 13 | 0.109272019 | 0.163908029 | 1.983287150 | 1.092720193 | 0.054636010 |

| | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |
| 6 | | | | | |
| 7 | 1.664559311 | | | | |
| 8 | 2.521907173 | 1.223648966 | | | |
| 9 | 2.375222701 | 1.553813866 | 0.699340923 | | |
| 10 | 1.043547784 | 2.509980867 | 3.041494835 | 2.659467348 | |
| 11 | 1.316362725 | 1.742888707 | 1.862381168 | 1.395613685 | 1.274551599 |
| 12 | 0.273180048 | 1.485454483 | 2.441925131 | 2.372078711 | 1.316727832 |
| 13 | 1.354089262 | 0.513578491 | 1.202833742 | 1.282769206 | 2.071452773 |

| | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |
| 6 | | | | | |
| 7 | | | | | |
| 8 | | | | | |
| 9 | | | | | |
| 10 | | | | | |
| 11 | | | | | |
| 12 | 1.456321731 | | | | |
| 13 | 1.229310217 | 1.244979455 | | | |

| | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |
| 6 | | | | | |
| 7 | | | | | |
| 8 | | | | | |
| 9 | | | | | |
| 10 | | | | | |
| 11 | | | | | |
| 12 | | | | | |
| 13 | | | | | |

| | 21 | 22 | 23 | 24 | 25 |
|---|---|---|---|---|---|
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |
| 6 | | | | | |
| 7 | | | | | |
| 8 | | | | | |
| 9 | | | | | |

```
10
11
12
13
                26          27          28          29          30
2
3
4
5
6
7
8
9
10
11
12
13
                31          32          33          34          35
2
3
4
5
6
7
8
9
10
11
12
13
                36          37          38          39          40
2
3
4
5
6
7
8
9
10
11
12
13
                41          42          43          44          45
2
3
4
5
6
7
8
9
10
11
12
13
                46          47          48          49          50
2
3
4
5
6
```

```
6
7
8
9
10
11
12
13
```

51 52 53 54 55

```
2
3
4
5
6
7
8
9
10
11
12
13
```

56 57 58 59 60

```
2
3
4
5
6
7
8
9
10
11
12
13
```

61 62 63 64 65

```
2
3
4
5
6
7
8
9
10
11
12
13
```

66 67 68 69 70

```
2
3
4
5
6
7
8
9
10
11
12
13
```

71 72 73 74 75

```
2
```

3
4
5
6
7
8
9
10
11
12
13

| | | | | |
|---|---|---|---|---|
| 76 | 77 | 78 | 79 | 80 |

2
3
4
5
6
7
8
9
10
11
12
13

| | | | | |
|---|---|---|---|---|
| 81 | 82 | 83 | 84 | 85 |

2
3
4
5
6
7
8
9
10
11
12
13

| | | | | |
|---|---|---|---|---|
| 86 | 87 | 88 | 89 | 90 |

2
3
4
5
6
7
8
9
10
11
12
13

| | | | | |
|---|---|---|---|---|
| 91 | 92 | 93 | 94 | 95 |

2
3
4
5
6
7
8
9
10
11
12