Capstone_project_Olist

Carmen Marquez

8/12/2019

The goal of my capstone project is understand and find out a way to predict Olist´s customers satisfaction. Olist(https://olist.com/) is a brazilian company that basically is a great sales channel it is present on the main marketplaces of Brazil and it is formed by thousands of retailers. It´s something similar to the "brazilian Amazon". I got the data from kaggle (https://www.kaggle.com/olistbr/brazilian-ecommerce and https://www.kaggle.com/olistbr/marketing-funnel-olist/home). After having a look to all the files available I decided to include in my analysis only the most relevant ones for the customer satisfaction, since the memory of my pc could not handle the total number of observation of all the files.

So, the steps I followed to achieve my goal are the following: -First, I did a Exploratory Data Analysis to have a better understanding of the data I was dealing with -Second, I did a little of feature enginering, creating new variables and I clean the data dropping the variables that were not relevant of were missing to many values. Also I perform a sentiment analysis. -Third, I did some unsupervised analysis (Cluster analysis in particular) too see how homogeneous was the information and how it could be split into different categories. -Fourth, I did some supervised analysis. -1st. A multivariable lineal regression model using the review score as the dependent variable as a numeric variable. -2n. An ordinal logistic regression model using the review score as the dependent variable as a factor -3rd. I run different machine learning models and chose the random forest as the best.

-Fith, I decided to choose the ordinal logistic regression model and I interpret the results.

1.Imported and merged the data files

```
library(tidyverse)
```

```
## ── Attaching packages ────────────────────────────────────────────────────────────────
─────────────────────── tidyverse 1.2.1 ──
```

```
## ✔ ggplot2 3.2.1     ✔ purrr   0.3.2
## ✔ tibble  2.1.3     ✔ dplyr   0.8.3
## ✔ tidyr   0.8.3     ✔ stringr 1.4.0
## ✔ readr   1.3.1     ✔ forcats 0.4.0
```

```
## ── Conflicts ─────────────────────────────────────────────────────────────────────────
──────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(dplyr)

#We need to import the data
#As there are some blank values in our data, will replace it by NA

setwd("/home/achaparro/personal/Carmen")
MQL <- read.csv("olist_marketing_qualified_leads_dataset.csv", na.strings = c("", "NA"))
closed_deals <- read.csv("olist_closed_deals_dataset.csv", na.strings = c("", "NA"))
total <- merge(closed_deals,MQL, by ="mql_id", all= TRUE)


order_items <- read.csv("olist_order_items_dataset.csv", na.strings = c("", "NA"))
order_reviews <- read.csv("olist_order_reviews_dataset _EN.csv", na.strings = c("", "NA"))
orders <- read.csv("olist_orders_dataset.csv", na.strings = c("", "NA"))

#The dataset "total" contains the Marketing funnel key variables, now let´s merge that information with the
Brazilian e-commerce public dataset

total1 <- merge(total,order_items, by ="seller_id", all= TRUE)

total2 <- merge(total1,orders, by ="order_id", all= TRUE)

total3 <- merge(total2,order_reviews, by ="order_id", all= TRUE)
```

2.Some Exploratory Data Analysis to understand better and clean our data

```r
#1.We drop the identifier variables as they are not usefull for our purpose


remove02 = c("order_id", "seller_id" , "mql_id" , "sdr_id" , "sr_id","landing_page_id" , "product_id" , "cus
tomer_id" , "review_id")

total4 = total3 %>% dplyr::select(-remove02)

#2. We calculate son new variables from the date variables

total4$won_date_new <- as.character(total4$won_date, format = "%Y-%m-%d")
total4$won_date_new <- as.Date(total4$won_date_new, format = "%Y-%m-%d")
total4$first_contact_date <- as.Date(total4$first_contact_date, format = "%Y-%m-%d")
total4$conversion_time <- (total4$won_date_new- total4$first_contact_date)
total4$conversion_time <- as.numeric(total4$conversion_time)

total4$order_delivered_customer_date <- as.Date(total4$order_delivered_customer_date, format = "%Y-%m-%d")
total4$order_purchase_timestamp <- as.Date(total4$order_purchase_timestamp, format = "%Y-%m-%d")
total4$delivery_time <- total4$order_delivered_customer_date - total4$order_purchase_timestamp
total4$delivery_time <- as.character(total4$delivery_time)
total4$delivery_time <- as.numeric(total4$delivery_time)
class(total4$delivery_time)
```

```
## [1] "numeric"
```

```r
total4$review_creation_date <- as.Date(total4$review_creation_date, format = "%Y-%m-%d")
total4$feedback_time <- total4$review_creation_date - total4$order_purchase_timestamp
total4$feedback_time <- as.numeric(total4$feedback_time)

total4$order_estimated_delivery_date <- as.Date(total4$order_estimated_delivery_date, format = "%Y-%m-%d")
total4$delay_time <- total4$order_delivered_customer_date -
total4$order_estimated_delivery_date
total4$delay_time <- as.numeric(total4$delay_time)

total4$order_approved_at <- as.Date(total4$order_approved_at, format = "%Y-%m-%d")
total4$approval_time <- total4$order_approved_at - total4$order_purchase_timestamp
total4$approval_time <- as.numeric(total4$approval_time)

#3. We delete the date variables since we got the information we need from them in our new variables

remove03 = c("won_date", "shipping_limit_date", "order_purchase_timestamp", "order_approved_at", "order_deli
vered_carrier_date", "order_delivered_customer_date", "order_estimated_delivery_date", "review_creation_date
" , "review_answer_timestamp", "won_date_new", "first_contact_date")

total4 = total4 %>% dplyr::select(-remove03)

str(total4)
```

```
## 'data.frame':    121720 obs. of  23 variables:
##  $ business_segment             : Factor w/ 33 levels "air_conditioning",..: NA NA NA NA NA NA NA NA NA 6
...
##  $ lead_type                    : Factor w/ 8 levels "industry","offline",..: NA NA NA NA NA NA NA NA NA
4 ...
##  $ lead_behaviour_profile       : Factor w/ 9 levels "cat","cat, wolf",..: NA NA NA NA NA NA NA NA NA 1 .
..
##  $ has_company                  : Factor w/ 2 levels "False","True": NA NA NA NA NA NA NA NA NA NA ...
##  $ has_gtin                     : Factor w/ 2 levels "False","True": NA NA NA NA NA NA NA NA NA NA ...
##  $ average_stock                : Factor w/ 6 levels "1-5","20-50",..: NA NA NA NA NA NA NA NA NA NA ...
##  $ business_type                : Factor w/ 3 levels "manufacturer",..: NA NA NA NA NA NA NA NA NA 3 ...
##  $ declared_product_catalog_size: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ declared_monthly_revenue     : num  NA NA NA NA NA NA NA NA NA 0 ...
##  $ origin                       : Factor w/ 10 levels "direct_traffic",..: NA NA NA NA NA NA NA NA NA 7 .
..
##  $ order_item_id                : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ price                        : num  58.9 239.9 199 13 199.9 ...
##  $ freight_value                : num  13.3 19.9 17.9 12.8 18.1 ...
##  $ order_status                 : Factor w/ 8 levels "approved","canceled",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ review_score                 : int  5 4 5 4 5 4 4 5 1 4 ...
##  $ review_comment_title         : Factor w/ 4244 levels " BOM , MENOS LOGISTICA",..: NA NA NA NA NA NA NA
NA NA NA ...
##  $ review_comment_message       : Factor w/ 36508 levels "\n","\n\n","\n\n\n\n\n\n\n\n\n\n",..: 26004 NA
5280 NA 14858 NA NA NA 18476 NA ...
##  $ EN_Review_comment_message    : Factor w/ 35745 levels "\n\n\nOf any request made on the site, this was
the one that took to deliver !!!",..: 21991 NA 34468 NA 12758 NA NA NA 17455 NA ...
##  $ conversion_time              : num  NA NA NA NA NA NA NA NA NA 0 ...
##  $ delivery_time                : num  7 16 8 6 25 7 8 5 10 2 ...
##  $ feedback_time                : num  8 17 9 7 26 8 9 6 11 3 ...
##  $ delay_time                   : num  -9 -3 -14 -6 -16 -15 -17 -16 0 -19 ...
##  $ approval_time                : num  0 0 0 0 0 2 0 1 1 0 ...
```

```
#Next, we´re going to obtain the sentiment score from the variable "review comment in english" to keep this
information as numeric

library(sentimentr)
library(stringr)
library(tidyverse)
library(tidytext)
library(tm)
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

```
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
library(gmodels)

total4$EN_Review_comment_message <- as.character(total4$EN_Review_comment_message)
En_review = get_sentences(total4$EN_Review_comment_message)
df = sentiment_by(En_review)
total4$sentiment = df$ave_sentiment

#Now we got the sentiment score. Let´s look at the most popular words before dropping the text variables.

#install.packages("RColorBrewer")
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```r
library(RColorBrewer)
library(tidyverse)
library(tm)
library(SnowballC)

corpus = Corpus(VectorSource(total4$EN_Review_comment_message))
corpus[[1]][1]
```

```
## $content
## [1] "Perfect product delivered before combined."
```

```r
#Conversion to Lowercase
corpus = tm_map(corpus, PlainTextDocument)
```

```
## Warning in tm_map.SimpleCorpus(corpus, PlainTextDocument): transformation
## drops documents
```

```r
corpus = tm_map(corpus, tolower)
```

```
## Warning in tm_map.SimpleCorpus(corpus, tolower): transformation drops
## documents
```

```r
#Removing Punctuation
corpus = tm_map(corpus, removePunctuation)
```

```
## Warning in tm_map.SimpleCorpus(corpus, removePunctuation): transformation
## drops documents
```

```r
#Remove stopwords
corpus = tm_map(corpus, removeWords, c("cloth", stopwords("english")))
```

```
## Warning in tm_map.SimpleCorpus(corpus, removeWords, c("cloth",
## stopwords("english"))): transformation drops documents
```

```r
# Stemming
corpus = tm_map(corpus, stemDocument)
```

```
## Warning in tm_map.SimpleCorpus(corpus, stemDocument): transformation drops
## documents
```

```r
# Eliminate white spaces
corpus = tm_map(corpus, stripWhitespace)
```

```
## Warning in tm_map.SimpleCorpus(corpus, stripWhitespace): transformation
## drops documents
```

```r
corpus[[1]][1]
```

```
## $content
## [1] "perfect product deliv combin"
```

```r
#Next step is extracting the word frequencies, to be used as tags, for building the word cloud:
DTM <- TermDocumentMatrix(corpus)
mat <- as.matrix(DTM)
f <- sort(rowSums(mat),decreasing=TRUE)
dat <- data.frame(word = names(f),freq=f)
head(dat, 5)
```

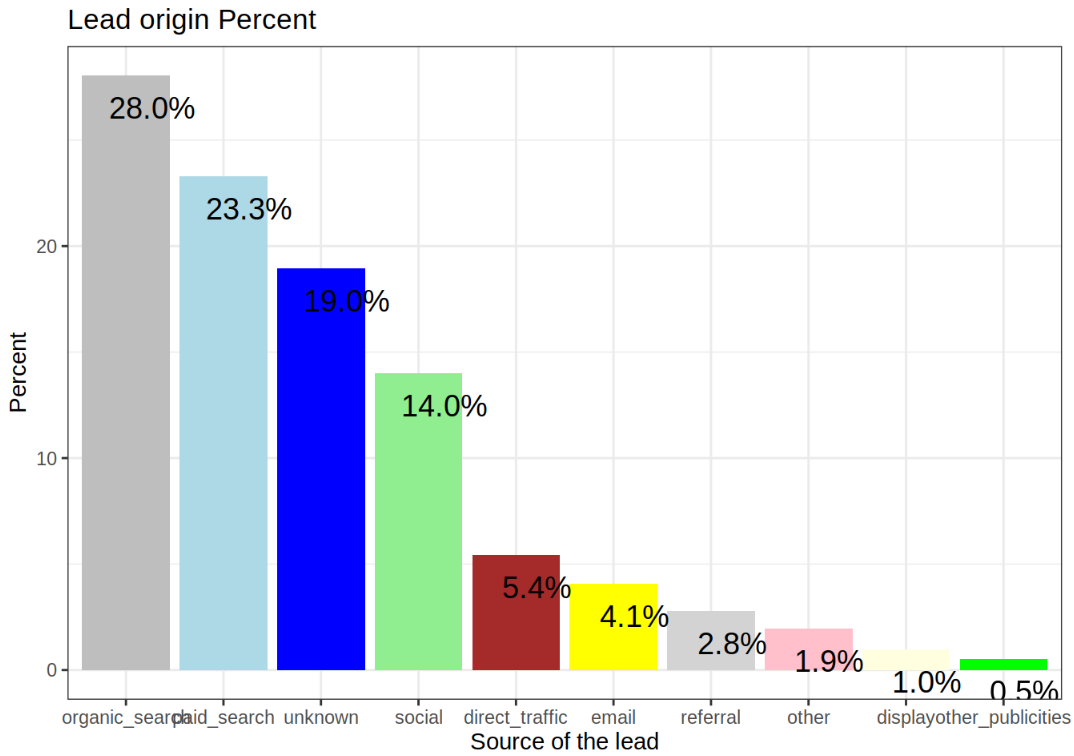
```
##              word  freq
## product   product 22968
## receiv     receiv  8130
## good         good  7532
## deliveri deliveri  7253
## deliv       deliv  6491
```

```
set.seed(100)
wordcloud(words = dat$word, freq = dat$freq, min.freq = 3, max.words=250, random.order=FALSE, rot.per=0.30, colors=brewer.pal(8, "Dark2"))
```

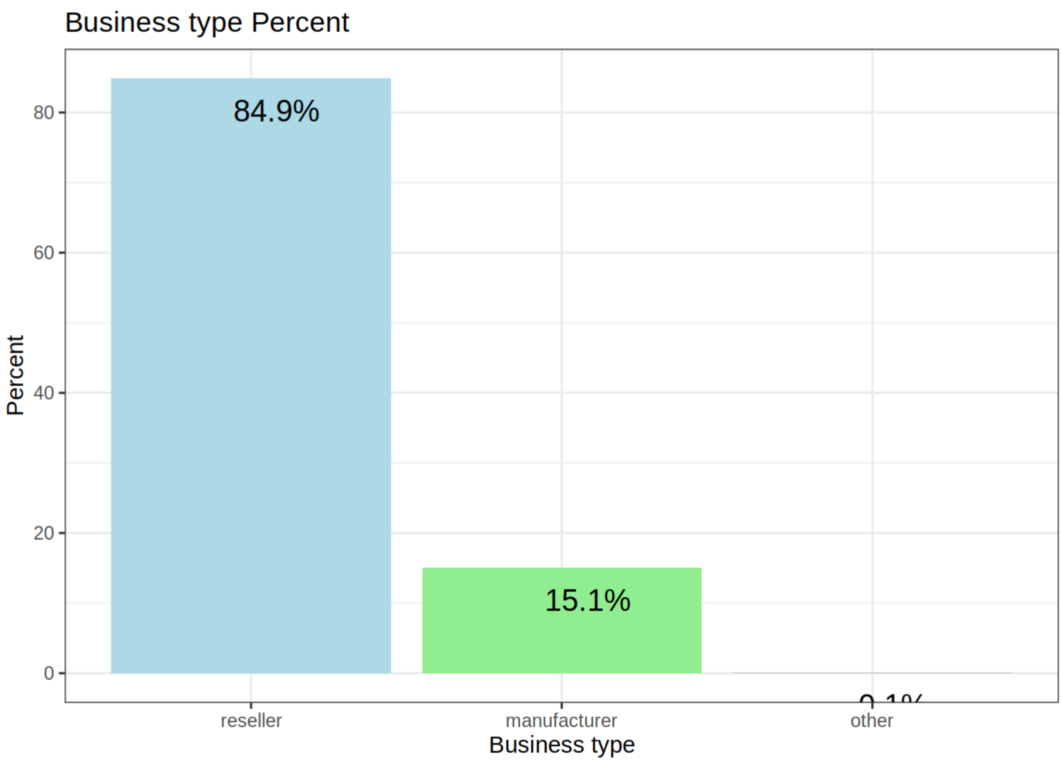

```
#4.Looking at the structure of our data there are some variables classified as a factor with too many levels, that´s because they should be classified as text. We are going to remove them since we don´t need them for our analysis. We´ll also remove other variables like the lead behaviour profile or the has_gtin, since we don´t know the meaning of them.

remove04 = c("review_comment_message", "review_comment_title", "EN_Review_comment_message", "has_gtin", "lead_behaviour_profile")

total5 = total4 %>% dplyr::select(-remove04)
str(total5)
```

```
## 'data.frame':    121720 obs. of  19 variables:
##  $ business_segment             : Factor w/ 33 levels "air_conditioning",..: NA NA NA NA NA NA NA NA NA 6
...
##  $ lead_type                    : Factor w/ 8 levels "industry","offline",..: NA NA NA NA NA NA NA NA NA
4 ...
##  $ has_company                  : Factor w/ 2 levels "False","True": NA NA NA NA NA NA NA NA NA NA ...
##  $ average_stock                : Factor w/ 6 levels "1-5","20-50",..: NA NA NA NA NA NA NA NA NA NA ...
##  $ business_type                : Factor w/ 3 levels "manufacturer",..: NA NA NA NA NA NA NA NA NA 3 ...
##  $ declared_product_catalog_size: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ declared_monthly_revenue     : num  NA NA NA NA NA NA NA NA NA 0 ...
##  $ origin                       : Factor w/ 10 levels "direct_traffic",..: NA NA NA NA NA NA NA NA NA 7 .
..
##  $ order_item_id                : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ price                        : num  58.9 239.9 199 13 199.9 ...
##  $ freight_value                : num  13.3 19.9 17.9 12.8 18.1 ...
##  $ order_status                 : Factor w/ 8 levels "approved","canceled",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ review_score                 : int  5 4 5 4 5 4 4 5 1 4 ...
##  $ conversion_time              : num  NA NA NA NA NA NA NA NA NA 0 ...
##  $ delivery_time                : num  7 16 8 6 25 7 8 5 10 2 ...
##  $ feedback_time                : num  8 17 9 7 26 8 9 6 11 3 ...
##  $ delay_time                   : num  -9 -3 -14 -6 -16 -15 -17 -16 0 -19 ...
##  $ approval_time                : num  0 0 0 0 0 2 0 1 1 0 ...
##  $ sentiment                    : num  0.335 0 0.769 0 0.158 ...
```

```
#5.Now let´s undertand better the distribution and relationship between our variables


#5.1 How many leads come from each origin source?

#In this variable there are too many missing values, but before we drop it, let´s see what we can find out f
rom the observations we have omitting those missing values using drop_na

total5 %>%
  drop_na(origin) %>%
  group_by(origin) %>%
  summarise(Count = n())%>%
  mutate(percent = prop.table(Count)*100)%>%
  ggplot(aes(reorder(origin, -percent), percent), fill = origin, na.rm = TRUE)+
  geom_col(fill = c("grey", "light blue", "blue", "light green", "brown", "yellow", "light grey", "pink", "
light yellow", "green"))+
  geom_text(aes(label = sprintf("%.1f%%", percent)), hjust = 0.2, vjust = 2, size = 5)+
  theme_bw()+
  xlab("Source of the lead") + ylab("Percent") + ggtitle("Lead origin Percent")
```



```
#Results: Paid search and Organic search seemst to be the best way to gain more leads, followed by social

#5.2 Which business type is the most common?

total5 %>%
  drop_na(business_type) %>%
  group_by(business_type) %>%
  summarise(Count = n())%>%
  mutate(percent = prop.table(Count)*100)%>%
  ggplot(aes(reorder(business_type, -percent), percent), fill = business_type, na.rm = TRUE)+
  geom_col(fill = c("light blue", "light green", "grey"))+
  geom_text(aes(label = sprintf("%.1f%%", percent)), hjust = 0.2, vjust = 2, size = 5)+
  theme_bw()+
  xlab("Business type") + ylab("Percent") + ggtitle("Business type Percent")
```




```
#Results: 84% are resellers, so this is the type of business that should be targeted by Olist

#5.3 Can we group the different categories from the business segment into less categories?

#We can reduce the number of levels by grouping those with less frecuency, but in this case it´s better to g
roup them using human judgement into similar categories

install.packages("ggplot2")
```

```
## Installing package into '/home/achaparro/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)
```

```
library(ggplot2)
library(ggpubr)
```

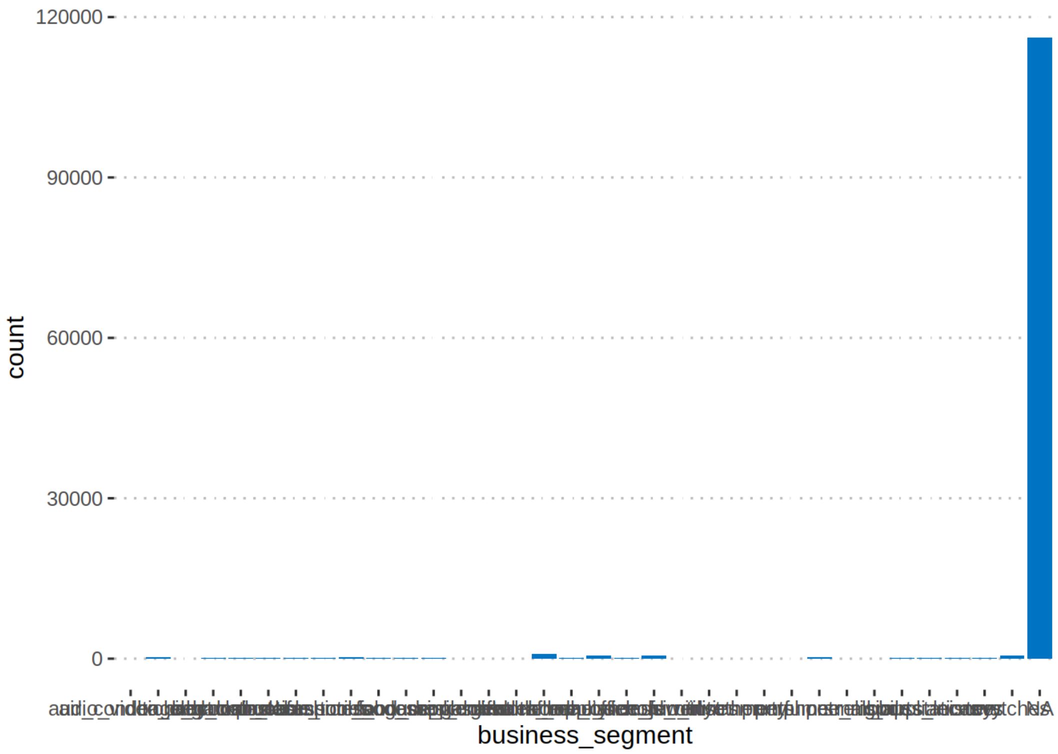
```
## Loading required package: magrittr
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
##
##     set_names
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

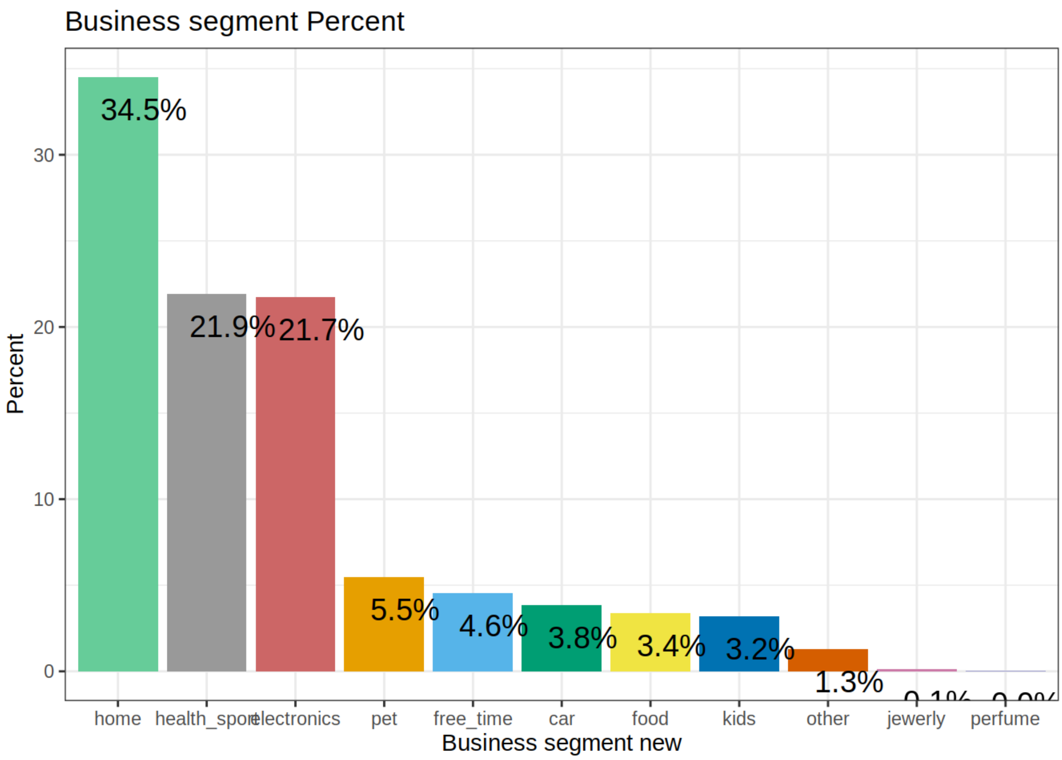
```
ggplot(total5, aes(x = business_segment,  na.rm = TRUE)) +
  geom_bar(fill = "#0073C2FF") +
  theme_pubclean()
```




```
library(dplyr)
df <- total5 %>%
  drop_na(business_segment) %>%
  group_by(business_segment) %>%
  summarise(counts = n())
df
```

```
## # A tibble: 33 x 2
##    business_segment                counts
##    <fct>                            <int>
##  1 air_conditioning                     6
##  2 audio_video_electronics            308
##  3 baby                                46
##  4 bags_backpacks                     148
##  5 bed_bath_table                     200
##  6 books                              111
##  7 car_accessories                    211
##  8 computers                          147
##  9 construction_tools_house_garden    356
## 10 fashion_accessories                 73
## # … with 23 more rows
```

```
total5$business_segment_new <- fct_collapse(total5$business_segment,
  home = c("home_appliances", "home_decor", "home_office_furniture", "household_utilities", "bed_bath_table"
, "construction_tools_house_garden", "air_conditioning"),
  food = c("food_supplement", "food_drink"),
  health_sport = c("sports_leisure", "health_beauty", "bags_backpacks"),
  electronics = c("audio_video_electronics", "computers", "games_consoles", "phone_mobile", "small_appliance
s", "watches"),
  free_time = c("gifts", "party", "fashion_accessories", "books", "handcrafted", "music_instruments"),
  kids = c("baby","toys"),
  car = "car_accessories",
  pet = "pet",
  other = c("stationery", "religious"))

#ggplot(total5, aes(x = business_segment_new, na.rm = TRUE)) +
 # geom_bar(fill = "#0073C2FF", na.rm = TRUE) +
 # theme_pubclean()

?ggplot
total5 %>%
  drop_na(business_segment_new) %>%
  group_by(business_segment_new) %>%
  summarise(Count = n())%>%
  mutate(percent = prop.table(Count)*100)%>%
  ggplot(aes(reorder(business_segment_new, -percent), percent), fill = business_type, na.rm = TRUE)+
  geom_col(fill = c("#66CC99", "#999999", "#CC6666", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2",
"#D55E00", "#CC79A7", "#9999CC"))+
  geom_text(aes(label = sprintf("%.1f%%", percent)), hjust = 0.2, vjust = 2, size = 5)+
  theme_bw()+
  xlab("Business segment new") + ylab("Percent") + ggtitle("Business segment Percent")
```




```
#Results: The segment more popular is Home, health&sport and electronics.

#Now we can drop the old "Business_segment" variable that contained too many levels

total5$business_segment <- NULL

str(total5)
```

```
## 'data.frame':    121720 obs. of  19 variables:
##  $ lead_type                    : Factor w/ 8 levels "industry","offline",..: NA NA NA NA NA NA NA NA NA 
4 ...
##  $ has_company                  : Factor w/ 2 levels "False","True": NA NA NA NA NA NA NA NA NA NA ...
##  $ average_stock                : Factor w/ 6 levels "1-5","20-50",..: NA NA NA NA NA NA NA NA NA NA ...
##  $ business_type                : Factor w/ 3 levels "manufacturer",..: NA NA NA NA NA NA NA NA NA 3 ...
##  $ declared_product_catalog_size: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ declared_monthly_revenue     : num  NA NA NA NA NA NA NA NA NA 0 ...
##  $ origin                       : Factor w/ 10 levels "direct_traffic",..: NA NA NA NA NA NA NA NA NA 7 .
..
##  $ order_item_id                : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ price                        : num  58.9 239.9 199 13 199.9 ...
##  $ freight_value                : num  13.3 19.9 17.9 12.8 18.1 ...
##  $ order_status                 : Factor w/ 8 levels "approved","canceled",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ review_score                 : int  5 4 5 4 5 4 4 5 1 4 ...
##  $ conversion_time              : num  NA NA NA NA NA NA NA NA NA 0 ...
##  $ delivery_time                : num  7 16 8 6 25 7 8 5 10 2 ...
##  $ feedback_time                : num  8 17 9 7 26 8 9 6 11 3 ...
##  $ delay_time                   : num  -9 -3 -14 -6 -16 -15 -17 -16 0 -19 ...
##  $ approval_time                : num  0 0 0 0 0 2 0 1 1 0 ...
##  $ sentiment                    : num  0.335 0 0.769 0 0.158 ...
##  $ business_segment_new         : Factor w/ 11 levels "home","electronics",..: NA NA NA NA NA NA NA NA NA 
5 ...
```

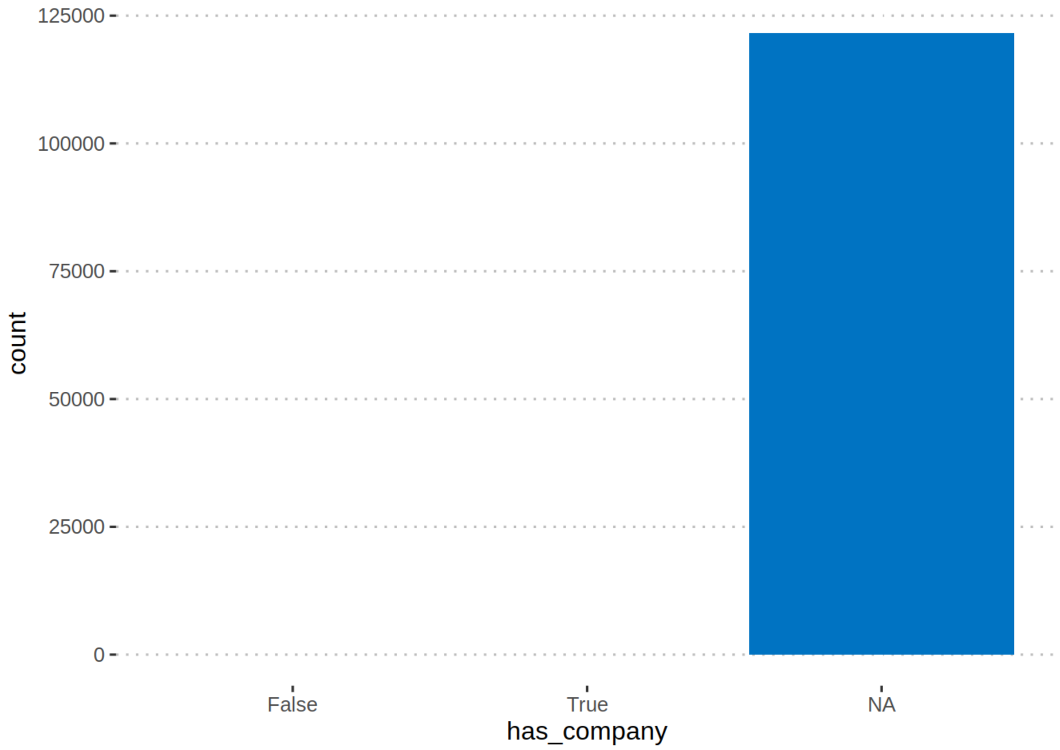
```
summary(total5)
```

```
##         lead_type      has_company    average_stock
##  online_medium:  2151   False:    29   1-5     :    10
##  online_big   :  1924   True :    59   20-50   :     8
##  online_small :   501   NA's :121632   200+    :     8
##  industry     :   361                  5-20    :    22
##  offline      :   219                  50-200  :    39
##  (Other)      :   305                  unknown:     4
##  NA's         :116259                  NA's    :121629
##       business_type    declared_product_catalog_size
##  manufacturer:   830   Min.   :   1
##  other       :     3   1st Qu.:  30
##  reseller    :  4667   Median : 100
##  NA's        :116220   Mean   : 233
##                        3rd Qu.: 300
##                        Max.   :2000
##                        NA's   :121651
##  declared_monthly_revenue            origin        order_item_id
##  Min.   :        0         organic_search:  3534   Min.   : 1.000
##  1st Qu.:        0         paid_search   :  2934   1st Qu.: 1.000
##  Median :        0         unknown       :  2390   Median : 1.000
##  Mean   :    11209         social        :  1763   Mean   : 1.199
##  3rd Qu.:        0         direct_traffic:   686   3rd Qu.: 1.000
##  Max.   :50000000          (Other)       :  1296   Max.   :21.000
##  NA's   :116208            NA's          :109117   NA's   :8398
##      price           freight_value          order_status      review_score
##  Min.   :   0.85   Min.   :  0.00   delivered  :110848   Min.   :1
##  1st Qu.:  39.90   1st Qu.: 13.08   shipped    :  1197   1st Qu.:3
##  Median :  74.90   Median : 16.26   canceled   :   711   Median :5
##  Mean   : 120.48   Mean   : 19.98   unavailable:   612   Mean   :4
##  3rd Qu.: 134.90   3rd Qu.: 21.15   invoiced   :   366   3rd Qu.:5
##  Max.   :6735.00   Max.   :409.68   (Other)    :   366   Max.   :5
##  NA's   :8398      NA's   :8398     NA's       :  7620   NA's   :7620
##  conversion_time  delivery_time    feedback_time       delay_time
##  Min.   : -2.0    Min.   :  0.00   Min.   :-111.00   Min.   :-147.00
##  1st Qu.:  4.0    1st Qu.:  7.00   1st Qu.:   8.00   1st Qu.: -17.00
##  Median : 11.0    Median : 10.00   Median :  11.00   Median : -13.00
##  Mean   : 25.1    Mean   : 12.42   Mean   :  13.28   Mean   : -12.04
##  3rd Qu.: 23.0    3rd Qu.: 16.00   3rd Qu.:  17.00   3rd Qu.:  -7.00
##  Max.   :427.0    Max.   :210.00   Max.   : 148.00   Max.   : 188.00
##  NA's   :116208   NA's   :10873    NA's   :7620      NA's   :10873
##  approval_time      sentiment           business_segment_new
##  Min.   :  0.00   Min.   :-1.62380   home          :  1901
##  1st Qu.:  0.00   1st Qu.: 0.00000   health_sport:  1207
##  Median :  0.00   Median : 0.00000   electronics :  1197
##  Mean   :  0.53   Mean   : 0.12190   pet          :   301
##  3rd Qu.:  1.00   3rd Qu.: 0.08866   free_time    :   251
##  Max.   :188.00   Max.   : 2.64545   (Other)      :   654
##  NA's   :7782                        NA's         :116209
```

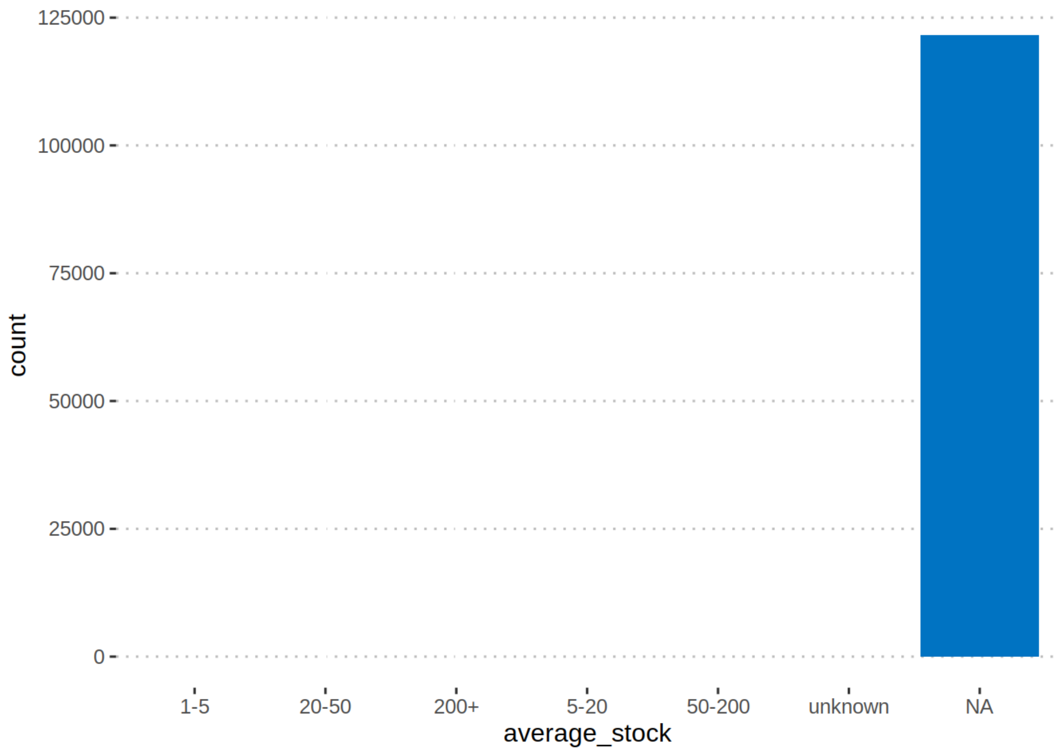
```r
#5.4 We also may want to see the variances of our variables to find out if all of them will be adding usefull information to our analysis. (first let´s look at our numeric variables)

total_num <- total5[-c(1:4,7,11, 19)]
sapply(total_num, var, na.rm=TRUE)
```

```
## declared_product_catalog_size      declared_monthly_revenue
##                  1.241721e+05                  4.652727e+11
##                 order_item_id                         price
##                  4.998388e-01                  3.359069e+04
##                 freight_value                  review_score
##                  2.491008e+02                  1.994452e+00
##               conversion_time                 delivery_time
##                  2.176785e+03                  8.928986e+01
##                 feedback_time                    delay_time
##                  6.247500e+01                  1.032310e+02
##                 approval_time                     sentiment
##                  1.387445e+00                  9.515489e-02
```

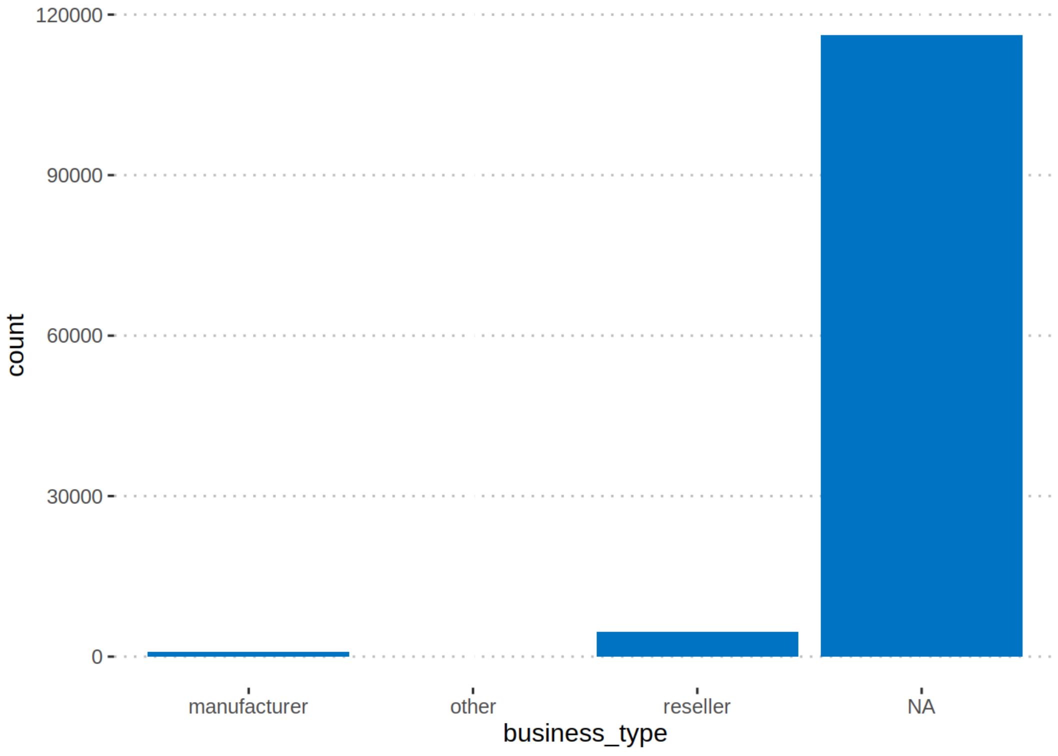
```r
#The variable order_item_id revenue have a very low variance0, so we decide to drop it
total5$order_item_id <- NULL

#And for the categorical variables, we may want to have a look to the observations of each level

ggplot(total5, aes(has_company)) +
  geom_bar(fill = "#0073C2FF") +
  theme_pubclean()
```



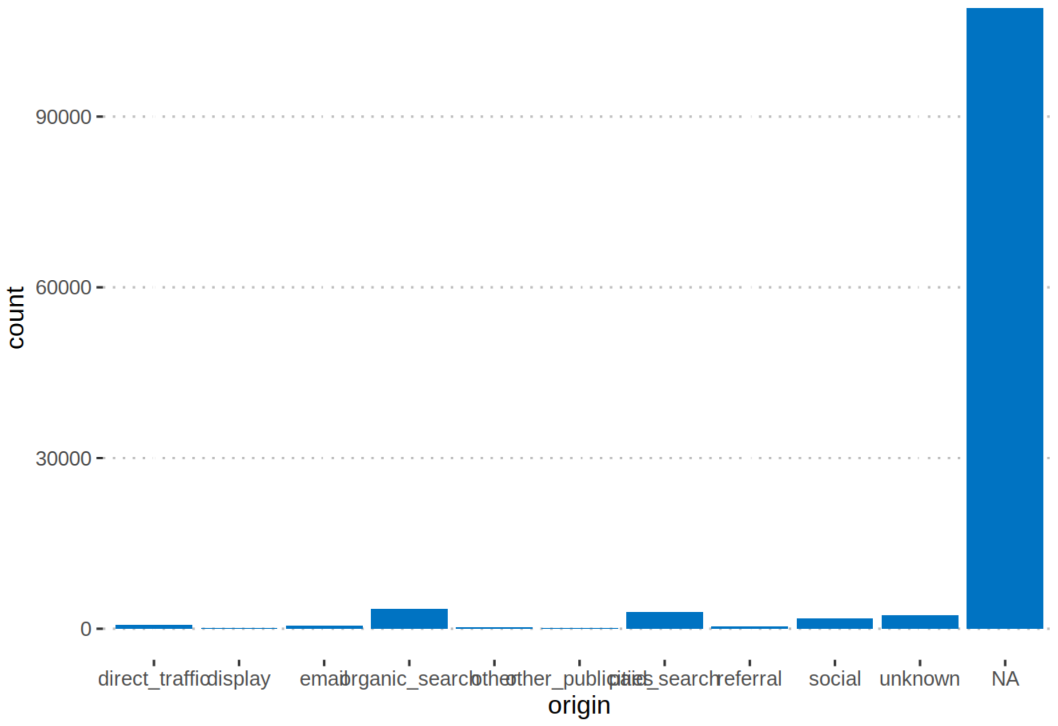

```r
ggplot(total5, aes(average_stock)) +
  geom_bar(fill = "#0073C2FF") +
  theme_pubclean()
```



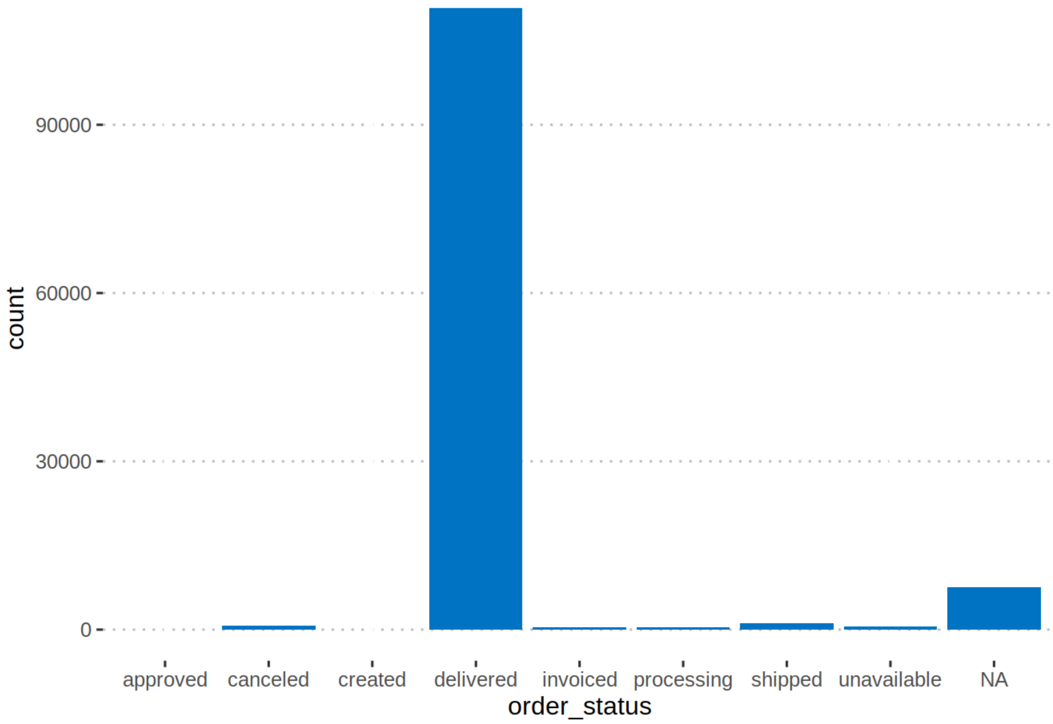
```r
ggplot(total5, aes(business_type)) +
  geom_bar(fill = "#0073C2FF") +
  theme_pubclean()
```




```r
ggplot(total5, aes(origin)) +
  geom_bar(fill = "#0073C2FF") +
  theme_pubclean()
```



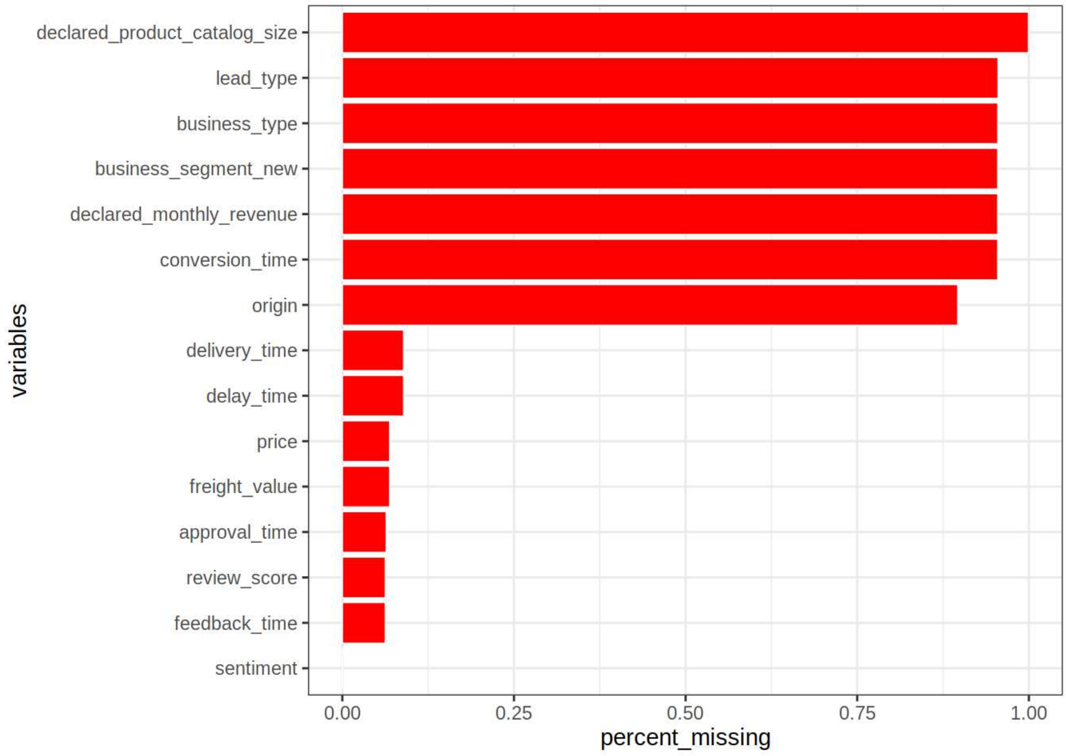

```r
ggplot(total5, aes(order_status)) +
  geom_bar(fill = "#0073C2FF") +
  theme_pubclean()
```




```
#Looking at the results, we decide to drop the variables: has_company, average_stock and order_status, since most of their observations are clasiffied in only one level, so they won´t bring usefull information to our analysis

total5$has_company <- NULL
total5$average_stock <- NULL
total5$order_status <- NULL


#6.Handling the remainding missing values

missing_data <- total5 %>% summarise_all(funs(sum(is.na(.))/n()))
```

```
## Warning: funs() is soft deprecated as of dplyr 0.8.0
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once per session.
```

```
missing_data <- gather(missing_data, key = "variables", value = "percent_missing")
ggplot(missing_data, aes(x = reorder(variables, percent_missing), y = percent_missing)) +
  geom_bar(stat = "identity", fill = "red", aes(color = I('white')), size = 0.3)+
  xlab('variables')+
  coord_flip()+
  theme_bw()
```




```
summary(total5)
```

```
##          lead_type          business_type
##  online_medium:  2151   manufacturer:   830
##  online_big   :  1924   other       :     3
##  online_small :   501   reseller    :  4667
##  industry     :   361   NA's        :116220
##  offline      :   219
##  (Other)      :   305
##  NA's         :116259
##  declared_product_catalog_size declared_monthly_revenue
##  Min.   :   1                  Min.   :       0
##  1st Qu.:  30                  1st Qu.:       0
##  Median : 100                  Median :       0
##  Mean   : 233                  Mean   :   11209
##  3rd Qu.: 300                  3rd Qu.:       0
##  Max.   :2000                  Max.   :50000000
##  NA's   :121651                NA's   :116208
##             origin           price          freight_value     review_score
##  organic_search:  3534   Min.   :   0.85   Min.   :  0.00   Min.   :1
##  paid_search   :  2934   1st Qu.:  39.90   1st Qu.: 13.08   1st Qu.:3
##  unknown       :  2390   Median :  74.90   Median : 16.26   Median :5
##  social        :  1763   Mean   : 120.48   Mean   : 19.98   Mean   :4
##  direct_traffic:   686   3rd Qu.: 134.90   3rd Qu.: 21.15   3rd Qu.:5
##  (Other)       :  1296   Max.   :6735.00   Max.   :409.68   Max.   :5
##  NA's          :109117   NA's   :8398      NA's   :8398     NA's   :7620
##  conversion_time  delivery_time    feedback_time       delay_time
##  Min.   : -2.0    Min.   :  0.00   Min.   :-111.00   Min.   :-147.00
##  1st Qu.:  4.0    1st Qu.:  7.00   1st Qu.:   8.00   1st Qu.: -17.00
##  Median : 11.0    Median : 10.00   Median :  11.00   Median : -13.00
##  Mean   : 25.1    Mean   : 12.42   Mean   :  13.28   Mean   : -12.04
##  3rd Qu.: 23.0    3rd Qu.: 16.00   3rd Qu.:  17.00   3rd Qu.:  -7.00
##  Max.   :427.0    Max.   :210.00   Max.   : 148.00   Max.   : 188.00
##  NA's   :116208   NA's   :10873    NA's   :7620      NA's   :10873
##  approval_time      sentiment           business_segment_new
##  Min.   :  0.00   Min.   :-1.62380   home          :  1901
##  1st Qu.:  0.00   1st Qu.: 0.00000   health_sport:   1207
##  Median :  0.00   Median : 0.00000   electronics :   1197
##  Mean   :  0.53   Mean   : 0.12190   pet          :   301
##  3rd Qu.:  1.00   3rd Qu.: 0.08866   free_time    :   251
##  Max.   :188.00   Max.   : 2.64545   (Other)      :   654
##  NA's   :7782                        NA's         :116209
```

```r
#We can see that still there are 7 variables with a high% of missing values (>100,000 obs). My decision is d
ropping those variables.

total5$lead_type <- NULL
total5$business_type <- NULL
total5$declared_product_catalog_size <- NULL
total5$declared_monthly_revenue <- NULL
total5$origin <- NULL
total5$conversion_time <- NULL
total5$business_segment_new <- NULL

summary(total5)
```

```
##      price          freight_value      review_score   delivery_time
##  Min.   :   0.85   Min.   :  0.00   Min.   :1      Min.   :  0.00
##  1st Qu.:  39.90   1st Qu.: 13.08   1st Qu.:3      1st Qu.:  7.00
##  Median :  74.90   Median : 16.26   Median :5      Median : 10.00
##  Mean   : 120.48   Mean   : 19.98   Mean   :4      Mean   : 12.42
##  3rd Qu.: 134.90   3rd Qu.: 21.15   3rd Qu.:5      3rd Qu.: 16.00
##  Max.   :6735.00   Max.   :409.68   Max.   :5      Max.   :210.00
##  NA's   :8398      NA's   :8398     NA's   :7620   NA's   :10873
##  feedback_time       delay_time       approval_time      sentiment
##  Min.   :-111.00   Min.   :-147.00   Min.   :  0.00   Min.   :-1.62380
##  1st Qu.:   8.00   1st Qu.: -17.00   1st Qu.:  0.00   1st Qu.: 0.00000
##  Median :  11.00   Median : -13.00   Median :  0.00   Median : 0.00000
##  Mean   :  13.28   Mean   : -12.04   Mean   :  0.53   Mean   : 0.12190
##  3rd Qu.:  17.00   3rd Qu.:  -7.00   3rd Qu.:  1.00   3rd Qu.: 0.08866
##  Max.   : 148.00   Max.   : 188.00   Max.   :188.00   Max.   : 2.64545
##  NA's   :7620      NA's   :10873     NA's   :7782
```

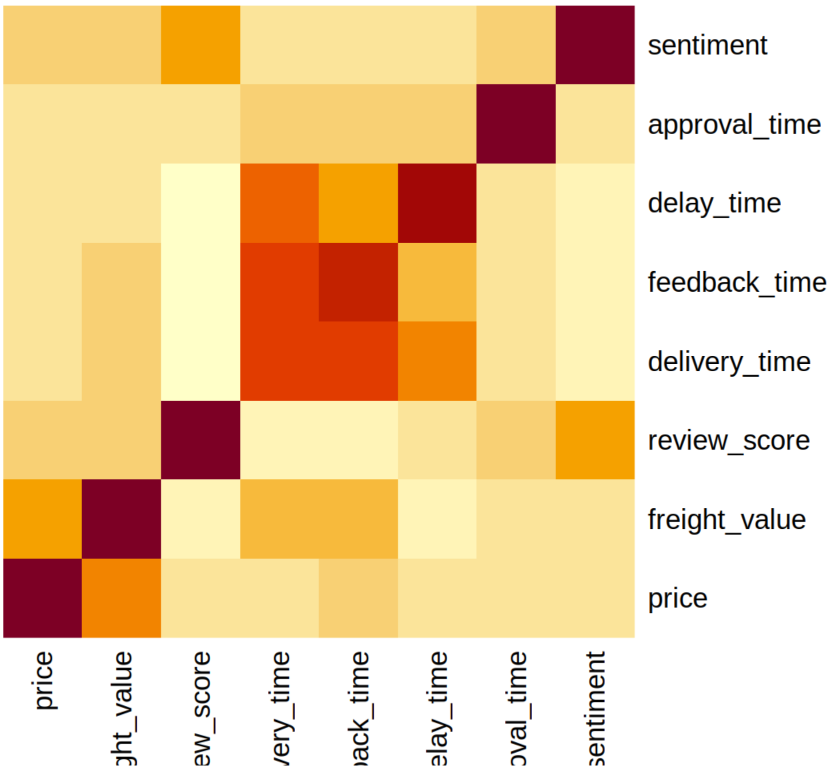
```r
#A better solution is remove the rows that cointains more than 50% of missing values
#dat[-which(rowMeans(is.na(dat)) > 0.5), ] is not working, why?

#Now we´ll remove the missing values of the remaining variables

total6 <- na.omit(total5)
summary(total6)
```

```
##      price          freight_value      review_score    delivery_time
##  Min.   :   0.85   Min.   :  0.00   Min.   :1.000   Min.   :  0.00
##  1st Qu.:  39.90   1st Qu.: 13.08   1st Qu.:4.000   1st Qu.:  7.00
##  Median :  74.90   Median : 16.25   Median :5.000   Median : 10.00
##  Mean   : 119.81   Mean   : 19.94   Mean   :4.066   Mean   : 12.42
##  3rd Qu.: 133.90   3rd Qu.: 21.15   3rd Qu.:5.000   3rd Qu.: 16.00
##  Max.   :6735.00   Max.   :409.68   Max.   :5.000   Max.   :210.00
##  feedback_time      delay_time       approval_time       sentiment
##  Min.   :-77.00   Min.   :-147.00   Min.   : 0.0000   Min.   :-1.6238
##  1st Qu.:  8.00   1st Qu.: -17.00   1st Qu.: 0.0000   1st Qu.: 0.0000
##  Median : 11.00   Median : -13.00   Median : 0.0000   Median : 0.0000
##  Mean   : 12.89   Mean   : -12.03   Mean   : 0.5244   Mean   : 0.1343
##  3rd Qu.: 17.00   3rd Qu.:  -7.00   3rd Qu.: 1.0000   3rd Qu.: 0.1595
##  Max.   :112.00   Max.   : 188.00   Max.   :31.0000   Max.   : 2.6454
```

```r
#7. Correlation of remaining variables (looking for multicolinearity)

cor(total6)
```

```
##                     price freight_value review_score delivery_time
## price          1.000000000  0.4128919858  0.002442175    0.06262833
## freight_value  0.412891986  1.0000000000 -0.033009004    0.21480068
## review_score   0.002442175 -0.0330090041  1.000000000   -0.30462955
## delivery_time  0.062628330  0.2148006822 -0.304629550    1.00000000
## feedback_time  0.067919606  0.2524741414 -0.263903562    0.87454660
## delay_time    -0.003572271 -0.0399560099 -0.230318097    0.59704373
## approval_time  0.006680589  0.0262119652 -0.019522745    0.08547120
## sentiment      0.007755309 -0.0008628659  0.256464702   -0.09825000
##               feedback_time   delay_time approval_time     sentiment
## price            0.06791961 -0.003572271   0.006680589  0.0077553088
## freight_value    0.25247414 -0.039956010   0.026211965 -0.0008628659
## review_score    -0.26390356 -0.230318097  -0.019522745  0.2564647018
## delivery_time    0.87454660  0.597043726   0.085471197 -0.0982499969
## feedback_time    1.00000000  0.387446372   0.113606449 -0.0912748999
## delay_time       0.38744637  1.000000000   0.046670743 -0.0877782852
## approval_time    0.11360645  0.046670743   1.000000000 -0.0017662492
## sentiment       -0.09127490 -0.087778285  -0.001766249  1.0000000000
```

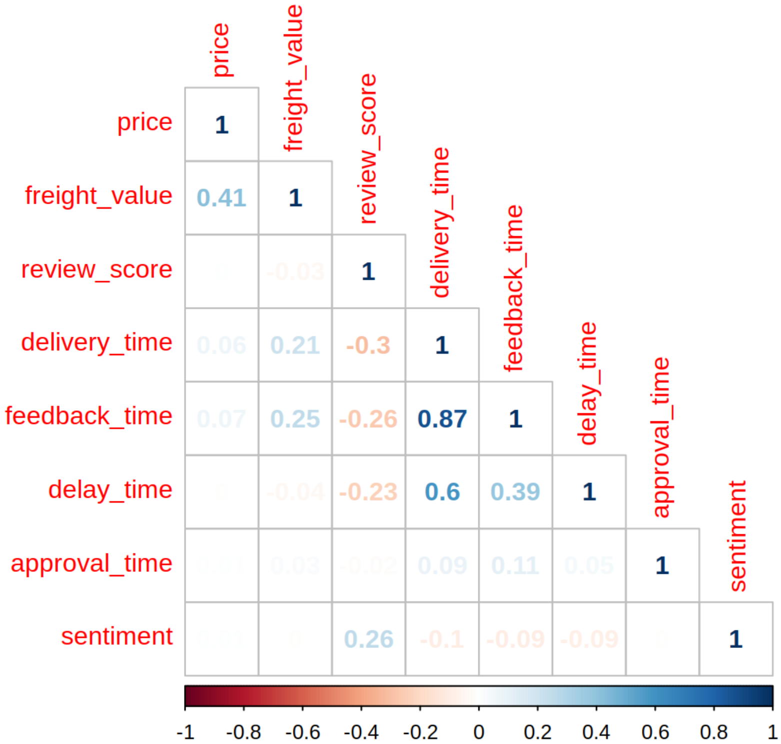
```
heatmap(cor(total6), Rowv= NA, Colv = NA)
```



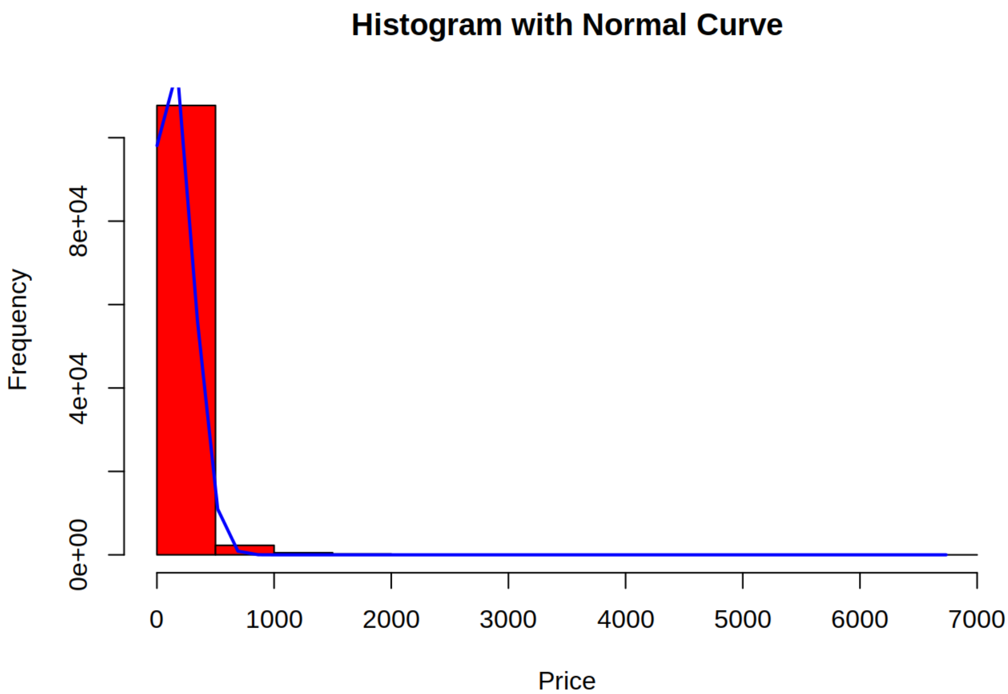

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

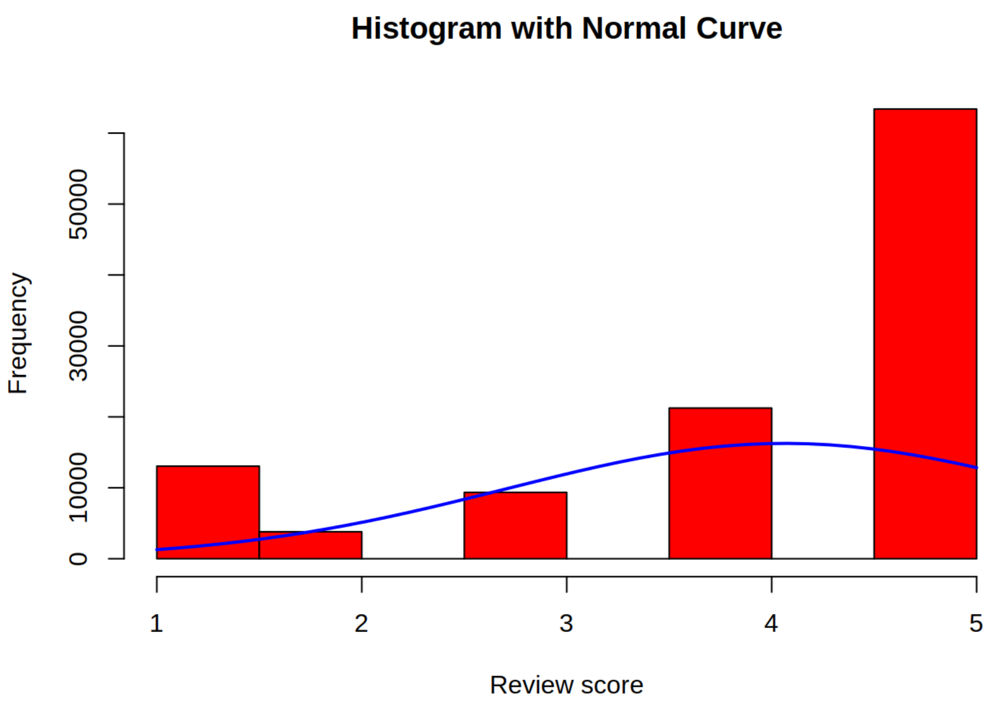
```
corrplot(cor(total6), method="number",type="lower")
```



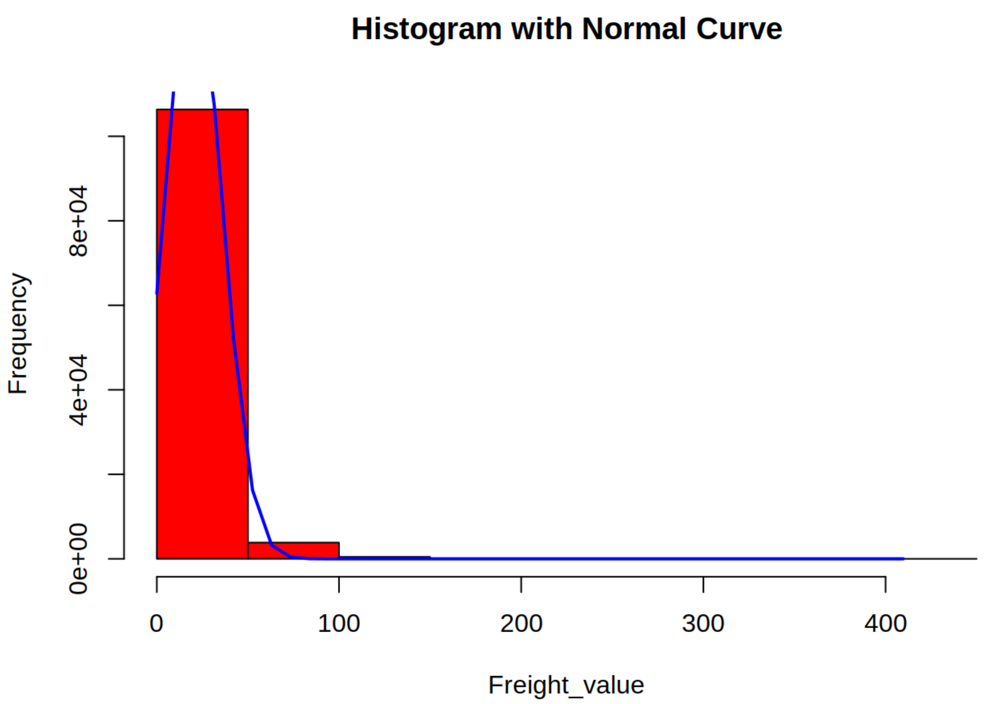

```
#Results: Feedback time (the number of days that pass since the customer buy an item until he/she writes a r
eview) and Delivery time are high correlated (0.87), but we´ll keep both since their autocorrelation is not
over 0.95.


#8. Inspecting Distribution of the remaining data

x0 <- total6$price
h0<-hist(x0, breaks=10, col="red", xlab="Price",
   main="Histogram with Normal Curve")
x0fit<-seq(min(x0),max(x0),length=40)
y0fit<-dnorm(x0fit,mean=mean(x0),sd=sd(x0))
y0fit <- y0fit*diff(h0$mids[1:2])*length(x0)
lines(x0fit, y0fit, col="blue", lwd=2)
```



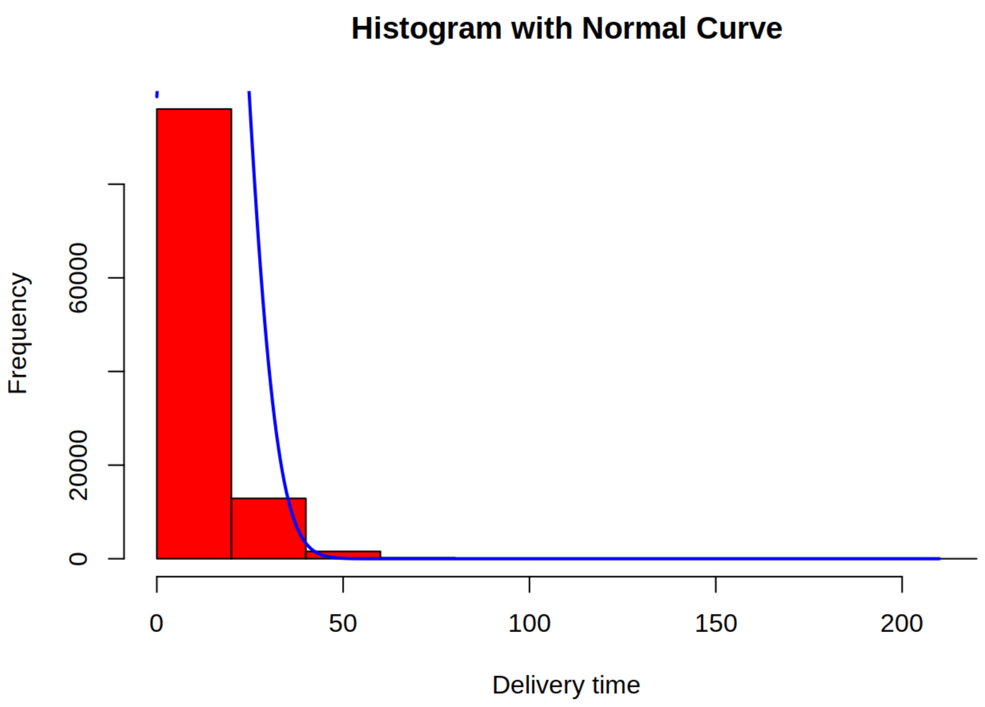
```
x1 <- total6$review_score
h1<-hist(x1, breaks=10, col="red", xlab="Review score",
   main="Histogram with Normal Curve")
x1fit<-seq(min(x1),max(x1),length=40)
yfit<-dnorm(x1fit,mean=mean(x1),sd=sd(x1))
yfit <- yfit*diff(h1$mids[1:2])*length(x1)
lines(x1fit, yfit, col="blue", lwd=2)
```



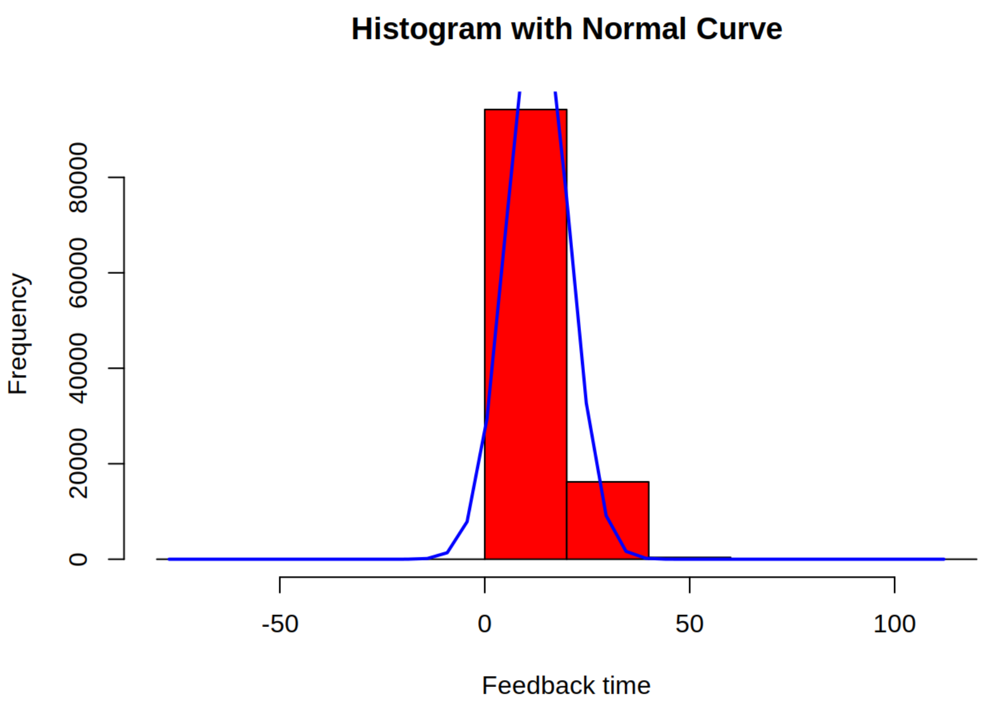

```
x <- total6$freight_value
h <-hist(x, breaks=10, col="red", xlab="Freight_value",
   main="Histogram with Normal Curve")
xfit<-seq(min(x),max(x),length=40)
yfit<-dnorm(xfit,mean=mean(x),sd=sd(x))
yfit <- yfit*diff(h$mids[1:2])*length(x)
lines(xfit, yfit, col="blue", lwd=2)
```



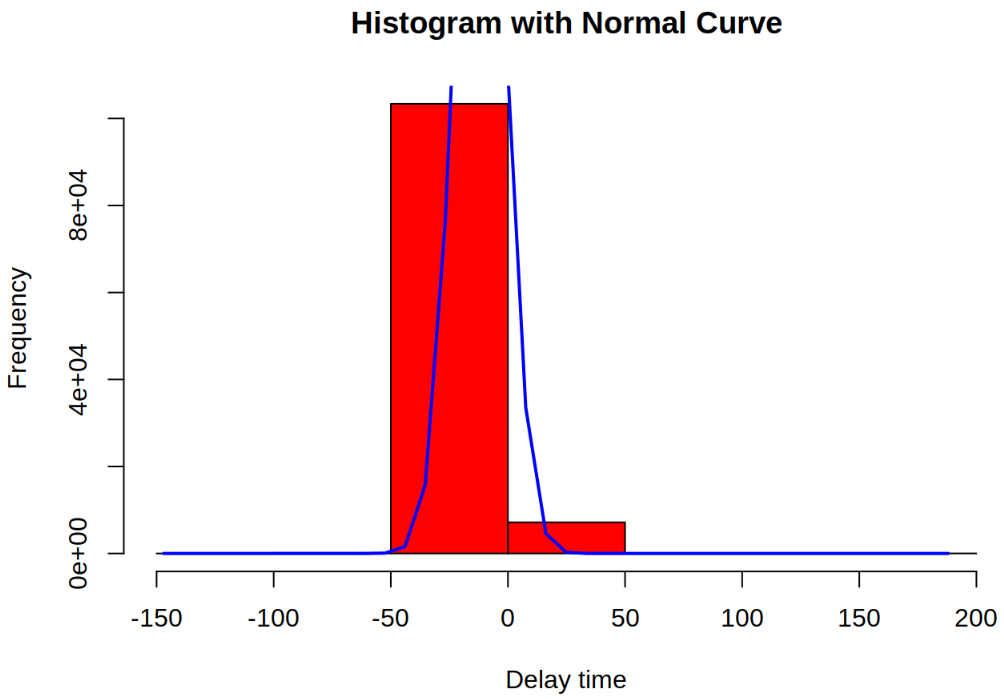
```
x1 <- total6$delivery_time
h1<-hist(x1, breaks=10, col="red", xlab="Delivery time",
   main="Histogram with Normal Curve")
x1fit<-seq(min(x1),max(x1),length=400)
y1fit<-dnorm(x1fit,mean=mean(x1),sd=sd(x1))
y1fit <- y1fit*diff(h$mids[1:2])*length(x1)
lines(x1fit, y1fit, col="blue", lwd=2)
```



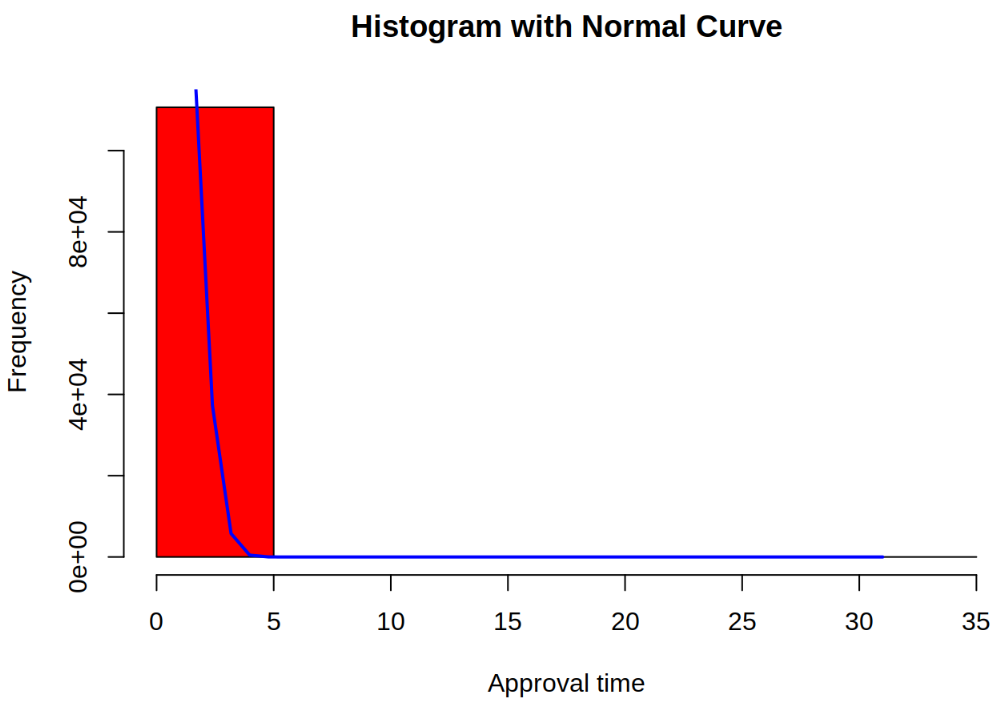

```
x <- total6$feedback_time
h<-hist(x, breaks=10, col="red", xlab="Feedback time",
   main="Histogram with Normal Curve")
xfit<-seq(min(x),max(x),length=40)
yfit<-dnorm(xfit,mean=mean(x),sd=sd(x))
yfit <- yfit*diff(h$mids[1:2])*length(x)
lines(xfit, yfit, col="blue", lwd=2)
```



```
x <- total6$delay_time
h<-hist(x, breaks=10, col="red", xlab="Delay time",
   main="Histogram with Normal Curve")
xfit<-seq(min(x),max(x),length=40)
yfit<-dnorm(xfit,mean=mean(x),sd=sd(x))
yfit <- yfit*diff(h$mids[1:2])*length(x)
lines(xfit, yfit, col="blue", lwd=2)
```




```
x <- total6$approval
h<-hist(x, breaks=10, col="red", xlab="Approval time",
   main="Histogram with Normal Curve")
xfit<-seq(min(x),max(x),length=40)
yfit<-dnorm(xfit,mean=mean(x),sd=sd(x))
yfit <- yfit*diff(h$mids[1:2])*length(x)
lines(xfit, yfit, col="blue", lwd=2)
```



#3.Unsupervised analysis

```
#CLUSTER Analysis

library(cluster)
library(factoextra)
```

```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

```
#install.packages("fastcluster")
library(fastcluster)
```

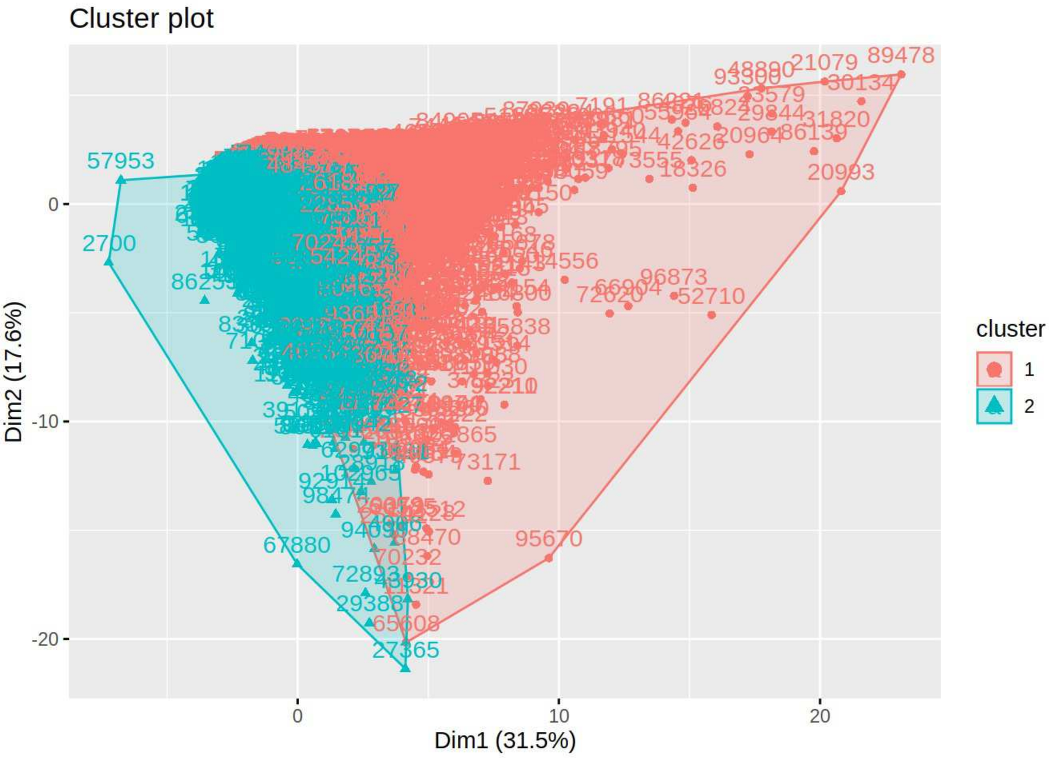
```
##
## Attaching package: 'fastcluster'
```

```
## The following object is masked from 'package:stats':
##
##     hclust
```

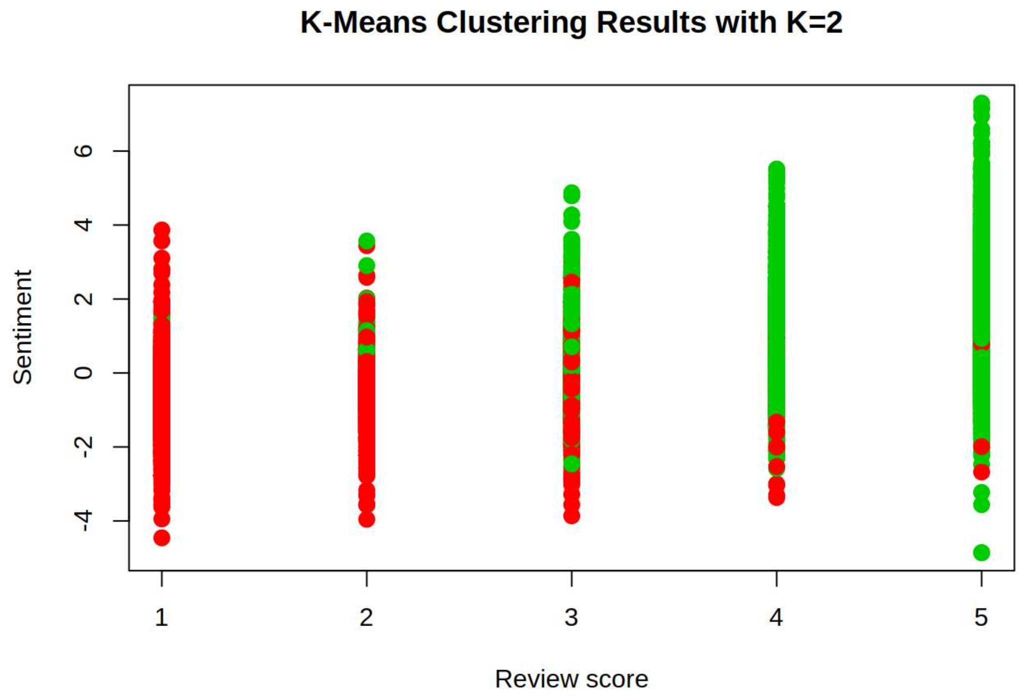
```
#km.out=kmeans(total6,2,nstart=5)

#When I try to run the cluster analysis I get this message: "Error: cannot allocate vector of size 45.8 Gb",
so I´m going to use Spark to solve this memory issue

#library(sparklyr)
library(dplyr)
#sc = spark_connect(master = "local")
# We need to copy the data frame "total6" into the database "sc" as a table.
#total6_tbl = copy_to(sc, total6)
#src_tbls(sc)

#spark_kmeans <-  ml_kmeans(total6_tbl, formula= NULL, k=3, max_iter = 10,
#features = c("price", "freight_value", "review_score", "delivery_time", "feedback_time", "delay_time","appr
oval_time", "sentiment"))

#summary(spark_kmeans)

#Time to compare the centers
# creating data frame from kmeans centers
#spark_kmeans_centers <- data.frame(spark_kmeans$centers)
# Printing centers of base and spark
#arrange(spark_kmeans_centers, review_score)

#Since I can´t find a way to visualize the clusters using Spark, I´m going to drop randomly some observation
s so it will be possible to perform the cluster analysis from my laptop.


total6_reduced <- total6[sample(nrow(total6), 50000), ]
str(total6_reduced)
```

```
## 'data.frame':    50000 obs. of  8 variables:
##  $ price        : num  120 59.9 140 53.9 89.9 ...
##  $ freight_value: num  20.3 17.7 15.7 19.4 16.4 ...
##  $ review_score : int  1 5 5 5 4 5 5 5 5 5 ...
##  $ delivery_time: num  8 15 17 13 12 18 10 7 4 5 ...
##  $ feedback_time: num  9 16 18 14 13 19 11 8 5 6 ...
##  $ delay_time   : num  -17 -5 -7 -31 -7 -16 -12 -17 -10 -10 ...
##  $ approval_time: num  0 0 0 2 0 0 0 0 0 0 ...
##  $ sentiment    : num  -0.158 0 0.694 0.411 0 ...
##  - attr(*, "na.action")= 'omit' Named int  80 85 262 272 424 546 556 561 562 563 ...
##   ..- attr(*, "names")= chr  "80" "85" "262" "272" ...
```

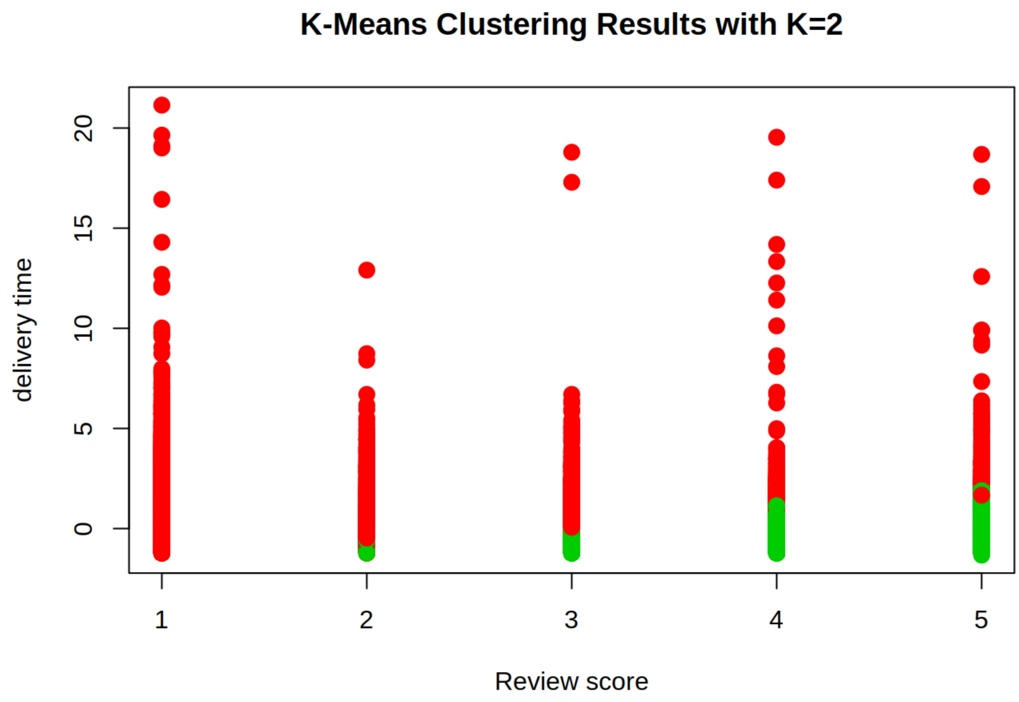
```
total6_reduced$price <- scale(total6_reduced$price)
total6_reduced$freight_value <- scale(total6_reduced$freight_value)
total6_reduced$delivery_time <- scale(total6_reduced$delivery_time)
total6_reduced$feedback_time <- scale(total6_reduced$feedback_time)
total6_reduced$delay_time <- scale(total6_reduced$delay_time)
total6_reduced$approval_time <- scale(total6_reduced$approval_time)
total6_reduced$sentiment <- scale(total6_reduced$sentiment)


km.out=kmeans(total6_reduced,2,nstart=25)
#km.out$cluster
#?fviz_cluster
fviz_cluster(km.out, data= total6_reduced)
```



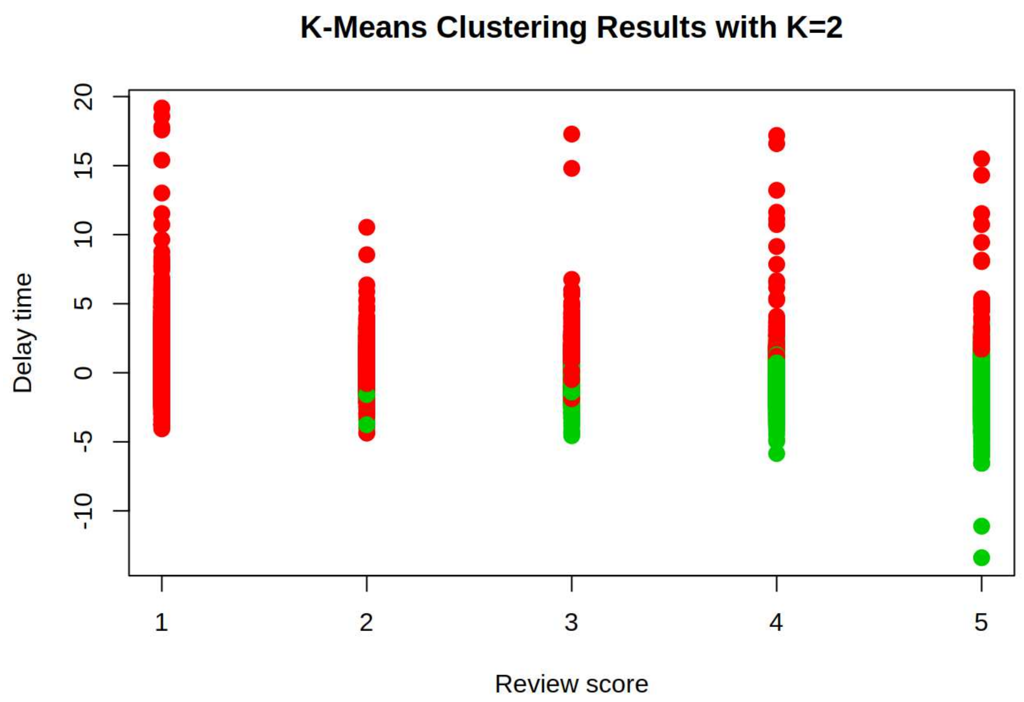

```
plot(total6_reduced[,c("review_score","sentiment")], col=(km.out$cluster+1),
     main="K-Means Clustering Results with K=2",
     xlab="Review score", ylab="Sentiment", pch=20, cex=2)
```




```r
plot(total6_reduced[,c("review_score","delivery_time")], col=(km.out$cluster+1),
     main="K-Means Clustering Results with K=2",
     xlab="Review score", ylab="delivery time", pch=20, cex=2)
```




```r
plot(total6_reduced[,c("review_score","delay_time")], col=(km.out$cluster+1),
     main="K-Means Clustering Results with K=2",
     xlab="Review score", ylab="Delay time", pch=20, cex=2)
```

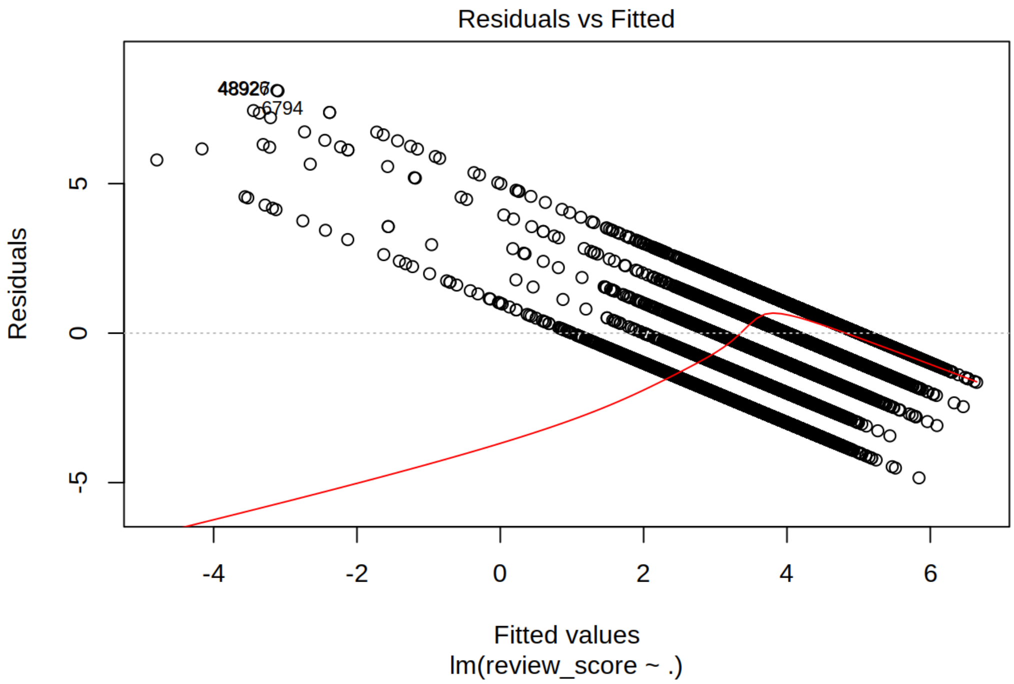


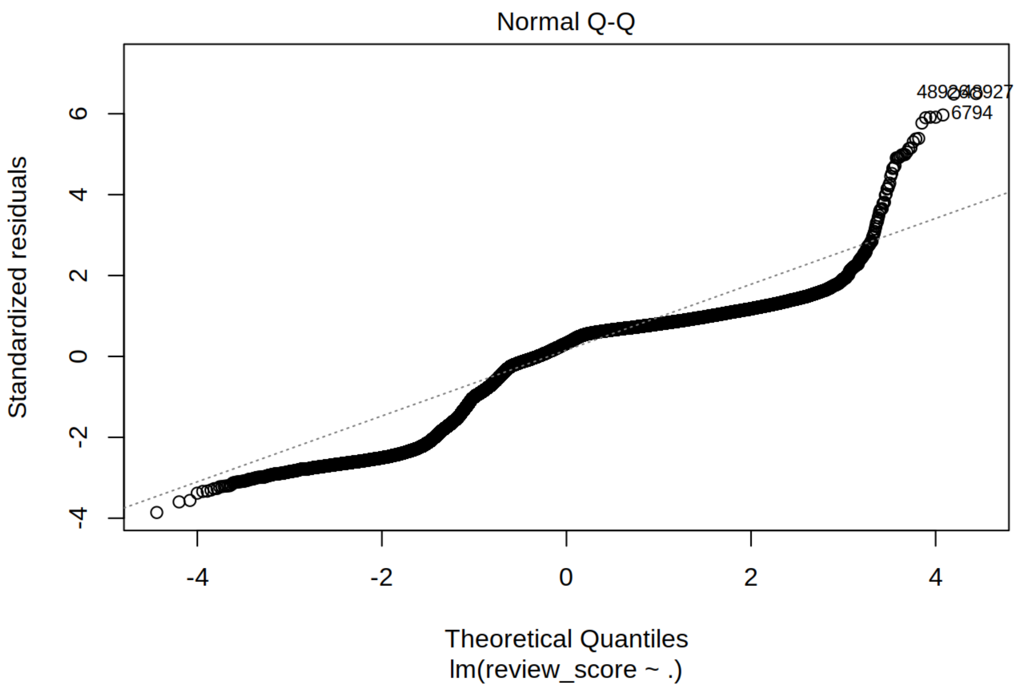
#SUPERVISED ANALYSIS (Regression predictive modeling) Treating the dependent variable as a numeric variable

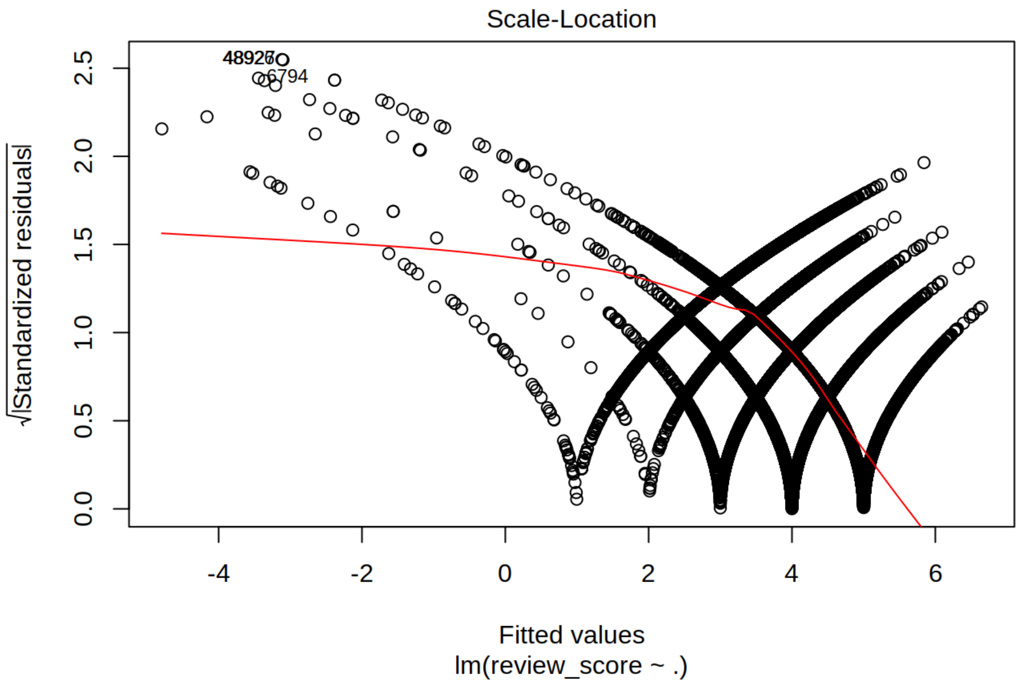
```
#Regression model

#Let´s see which variables explain the variability of the review score and see if we find the best model to predict it

fit01=lm(review_score ~., data=total6)
summary(fit01)
```

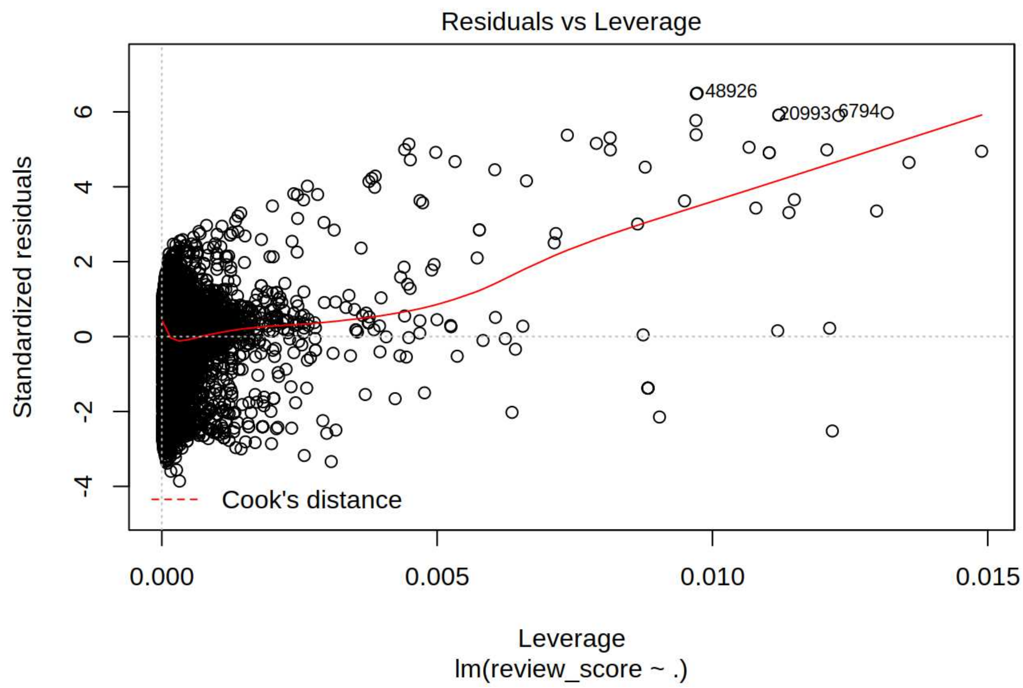
```
##
## Call:
## lm(formula = review_score ~ ., data = total6)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8422 -0.4930  0.4101  0.8853  8.1173
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.255e+00  1.167e-02 364.625  < 2e-16 ***
## price          7.474e-05  2.278e-05   3.280  0.00104 **
## freight_value  1.315e-03  2.770e-04   4.749 2.05e-06 ***
## delivery_time -3.208e-02  1.013e-03 -31.664  < 2e-16 ***
## feedback_time -5.127e-03  1.132e-03  -4.529 5.92e-06 ***
## delay_time    -8.861e-03  5.021e-04 -17.649  < 2e-16 ***
## approval_time  7.923e-03  3.872e-03   2.046  0.04074 *
## sentiment      9.654e-01  1.191e-02  81.073  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.255 on 110824 degrees of freedom
## Multiple R-squared:  0.1479, Adjusted R-squared:  0.1479
## F-statistic:  2748 on 7 and 110824 DF,  p-value: < 2.2e-16
```

```
plot(fit01)
```


Residuals vs Fitted
Residuals
5
0
-5
-4
-2
0
2
4
6
48927
6794
Fitted values
lm(review_score ~ .)


Normal Q-Q
Standardized residuals
6
4
2
0
-2
-4
-4
-2
0
2
4
6794
Theoretical Quantiles
lm(review_score ~ .)






```
#We found a model that explain the 14,79% of the review score variability.
#The most significant variables are the sentiment score(+), the delivery time(-), the delay time(-), the fee
dback time(-), the freight value (+). Followed by price, which is less significant and approval time, which
is not significant.

#We can run a 2nd regression model dropping that slightly significant variable and the R2 will not change.

fit02=lm(review_score ~ sentiment+delivery_time+delay_time+feedback_time+freight_value+price, data=total6)
summary(fit02)
```

```
##
## Call:
## lm(formula = review_score ~ sentiment + delivery_time + delay_time +
##     feedback_time + freight_value + price, data = total6)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8390 -0.4937  0.4095  0.8863  8.1264
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.257e+00  1.158e-02 367.539  < 2e-16 ***
## sentiment      9.656e-01  1.191e-02  81.092  < 2e-16 ***
## delivery_time -3.216e-02  1.012e-03 -31.762  < 2e-16 ***
## delay_time    -8.836e-03  5.019e-04 -17.605  < 2e-16 ***
## feedback_time -4.934e-03  1.128e-03  -4.373 1.22e-05 ***
## freight_value  1.316e-03  2.770e-04   4.752 2.02e-06 ***
## price          7.471e-05  2.278e-05   3.279  0.00104 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.255 on 110825 degrees of freedom
## Multiple R-squared:  0.1479, Adjusted R-squared:  0.1478
## F-statistic:  3206 on 6 and 110825 DF,  p-value: < 2.2e-16
```

#Continue in part 2 because of memomy issues

Carmen Marquez_Final project_part2

```r
#Getting Total6 again:
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
```

```r
library(tidyverse)
```

```
## ── Attaching packages ───────────────────────────────────────────────────────────────────────────────────────
───────────────────── tidyverse 1.2.1 ──
```

```
## ✔ ggplot2 3.2.1     ✔ readr   1.3.1
## ✔ tibble  2.1.3     ✔ purrr   0.3.2
## ✔ tidyr   0.8.3     ✔ stringr 1.4.0
## ✔ ggplot2 3.2.1     ✔ forcats 0.4.0
```

```
## ── Conflicts ───────────────────────────────────────────────────────────────────────────────────────────────
──────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```r
library(sentimentr)
library(stringr)
library(tidyverse)
library(tidytext)
library(tm)
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

```
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```r
library(gmodels)

setwd("/home/achaparro/personal/Carmen")
MQL <- read.csv("olist_marketing_qualified_leads_dataset.csv", na.strings = c("", "NA"))
closed_deals <- read.csv("olist_closed_deals_dataset.csv", na.strings = c("", "NA"))
total <- merge(closed_deals,MQL, by ="mql_id", all= TRUE)
order_items <- read.csv("olist_order_items_dataset.csv", na.strings = c("", "NA"))
order_reviews <- read.csv("olist_order_reviews_dataset _EN.csv", na.strings = c("", "NA"))
orders <- read.csv("olist_orders_dataset.csv", na.strings = c("", "NA"))
total1 <- merge(total,order_items, by ="seller_id", all= TRUE)
total2 <- merge(total1,orders, by ="order_id", all= TRUE)
total3 <- merge(total2,order_reviews, by ="order_id", all= TRUE)
remove02 = c("order_id", "seller_id" , "mql_id" , "sdr_id" , "sr_id","landing_page_id" , "product_id" , "cus
tomer_id" , "review_id")

total4 = total3 %>% dplyr::select(-remove02)
total4$won_date_new <- as.character(total4$won_date, format = "%Y-%m-%d")
total4$won_date_new <- as.Date(total4$won_date_new, format = "%Y-%m-%d")
total4$first_contact_date <- as.Date(total4$first_contact_date, format = "%Y-%m-%d")
total4$conversion_time <- (total4$won_date_new- total4$first_contact_date)
total4$conversion_time <- as.numeric(total4$conversion_time)

total4$order_delivered_customer_date <- as.Date(total4$order_delivered_customer_date, format = "%Y-%m-%d")
total4$order_purchase_timestamp <- as.Date(total4$order_purchase_timestamp, format = "%Y-%m-%d")
total4$delivery_time <- total4$order_delivered_customer_date - total4$order_purchase_timestamp
total4$delivery_time <- as.character(total4$delivery_time)
total4$delivery_time <- as.numeric(total4$delivery_time)
class(total4$delivery_time)
```

```
## [1] "numeric"
```

```r
total4$review_creation_date <- as.Date(total4$review_creation_date, format = "%Y-%m-%d")
total4$feedback_time <- total4$review_creation_date - total4$order_purchase_timestamp
total4$feedback_time <- as.numeric(total4$feedback_time)

total4$order_estimated_delivery_date <- as.Date(total4$order_estimated_delivery_date, format = "%Y-%m-%d")
total4$delay_time <- total4$order_delivered_customer_date -
total4$order_estimated_delivery_date
total4$delay_time <- as.numeric(total4$delay_time)

total4$order_approved_at <- as.Date(total4$order_approved_at, format = "%Y-%m-%d")
total4$approval_time <- total4$order_approved_at - total4$order_purchase_timestamp
total4$approval_time <- as.numeric(total4$approval_time)

#3. We delete the date variables since we got the information we need from them in our new variables

remove03 = c("won_date", "shipping_limit_date", "order_purchase_timestamp", "order_approved_at", "order_deli
vered_carrier_date", "order_delivered_customer_date", "order_estimated_delivery_date", "review_creation_date
" , "review_answer_timestamp", "won_date_new", "first_contact_date")

total4 = total4 %>% dplyr::select(-remove03)
total4$EN_Review_comment_message <- as.character(total4$EN_Review_comment_message)
En_review = get_sentences(total4$EN_Review_comment_message)
df = sentiment_by(En_review)
total4$sentiment = df$ave_sentiment
remove04 = c("review_comment_message", "review_comment_title", "EN_Review_comment_message", "has_gtin", "lea
d_behaviour_profile")

total5 = total4 %>% dplyr::select(-remove04)

total5$business_segment_new <- fct_collapse(total5$business_segment,
  home = c("home_appliances", "home_decor", "home_office_furniture", "household_utilities", "bed_bath_table"
, "construction_tools_house_garden", "air_conditioning"),
  food = c("food_supplement", "food_drink"),
  health_sport = c("sports_leisure", "health_beauty", "bags_backpacks"),
  electronics = c("audio_video_electronics", "computers", "games_consoles", "phone_mobile", "small_appliance
s", "watches"),
  free_time = c("gifts", "party", "fashion_accessories", "books", "handcrafted", "music_instruments"),
  kids = c("baby","toys"),
  car = "car_accessories",
  pet = "pet",
  other = c("stationery", "religious"))
total5$business_segment <- NULL
total_num <- total5[-c(1:4,7,11, 19)]
sapply(total_num, var, na.rm=TRUE)
```

```
## declared_product_catalog_size      declared_monthly_revenue
##                  1.241721e+05                  4.652727e+11
##                 order_item_id                         price
##                  4.998388e-01                  3.359069e+04
##                 freight_value                  review_score
##                  2.491008e+02                  1.994452e+00
##               conversion_time                 delivery_time
##                  2.176785e+03                  8.928986e+01
##                 feedback_time                    delay_time
##                  6.247500e+01                  1.032310e+02
##                 approval_time                     sentiment
##                  1.387445e+00                  9.515489e-02
```

```r
#The variable order_item_id revenue have a very low variance0, so we decide to drop it
total5$order_item_id <- NULL
total5$has_company <- NULL
total5$average_stock <- NULL
total5$order_status <- NULL

total5$lead_type <- NULL
total5$business_type <- NULL
total5$declared_product_catalog_size <- NULL
total5$declared_monthly_revenue <- NULL
total5$origin <- NULL
total5$conversion_time <- NULL
total5$business_segment_new <- NULL

total6 <- na.omit(total5)
```

#Supervised analysis (Ordered logistic regression model) Treating the dependent variable as an ordinal categorical variable

```r
#We can´t use the regular logistic model since our dependent variable is not binary, it is a categorical var
iable with several levels that are ordered, like a rank. So we are going to run an ordered logistic models a
nd improve it using the Stepwise AIC method.

#install.packages("stargazer")
library(stargazer)
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.
```

```
##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```r
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
str(total6)
```

```
## 'data.frame':    110832 obs. of  8 variables:
##  $ price         : num  58.9 239.9 199 13 199.9 ...
##  $ freight_value: num  13.3 19.9 17.9 12.8 18.1 ...
##  $ review_score : int  5 4 5 4 5 4 4 5 1 4 ...
##  $ delivery_time: num  7 16 8 6 25 7 8 5 10 2 ...
##  $ feedback_time: num  8 17 9 7 26 8 9 6 11 3 ...
##  $ delay_time   : num  -9 -3 -14 -6 -16 -15 -17 -16 0 -19 ...
##  $ approval_time: num  0 0 0 0 0 2 0 1 1 0 ...
##  $ sentiment    : num  0.335 0 0.769 0 0.158 ...
##  - attr(*, "na.action")= 'omit' Named int  80 85 262 272 424 546 556 561 562 563 ...
##   ..- attr(*, "names")= chr  "80" "85" "262" "272" ...
```

```r
total6$review_score=as.factor(total6$review_score)

olm <- polr(review_score ~., data=total6, Hess=TRUE, method="logistic")
summary(olm)
```

```
## Call:
## polr(formula = review_score ~ ., data = total6, Hess = TRUE,
##     method = "logistic")
##
## Coefficients:
##                    Value Std. Error t value
## price           0.000175  3.988e-05   4.389
## freight_value   0.001750  4.511e-04   3.880
## delivery_time  -0.069864  2.378e-03 -29.378
## feedback_time   0.016854  2.544e-03   6.624
## delay_time     -0.011458  8.187e-04 -13.995
## approval_time   0.007750  6.067e-03   1.277
## sentiment       1.806175  2.489e-02  72.571
##
## Intercepts:
##     Value     Std. Error t value
## 1|2   -2.4580    0.0206  -119.3753
## 2|3   -2.1199    0.0201  -105.6911
## 3|4   -1.4947    0.0194   -77.2070
## 4|5   -0.5045    0.0188   -26.8441
##
## Residual Deviance: 252049.53
## AIC: 252071.53
```

```
print(olm)
```

```
## Call:
## polr(formula = review_score ~ ., data = total6, Hess = TRUE,
##     method = "logistic")
##
## Coefficients:
##         price freight_value delivery_time feedback_time    delay_time
##  0.0001750439  0.0017503629 -0.0698643319  0.0168543491 -0.0114577769
## approval_time     sentiment
##  0.0077497715  1.8061750692
##
## Intercepts:
##        1|2        2|3        3|4        4|5
## -2.4580144 -2.1199184 -1.4946896 -0.5045025
##
## Residual Deviance: 252049.53
## AIC: 252071.53
```

```
#To get the pvalues we store the coefficient table, then calculate the p-values and combine back with the ta
ble:
(ctable <- coef(summary(olm)))
```

```
##                        Value   Std. Error     t value
## price           0.0001750439 3.988336e-05    4.388894
## freight_value   0.0017503629 4.510701e-04    3.880468
## delivery_time  -0.0698643319 2.378093e-03  -29.378297
## feedback_time   0.0168543491 2.544338e-03    6.624256
## delay_time     -0.0114577769 8.187273e-04  -13.994619
## approval_time   0.0077497715 6.067146e-03    1.277334
## sentiment       1.8061750692 2.488846e-02   72.570771
## 1|2            -2.4580144125 2.059065e-02 -119.375299
## 2|3            -2.1199184064 2.005768e-02 -105.691087
## 3|4            -1.4946896116 1.935950e-02  -77.207033
## 4|5            -0.5045024555 1.879382e-02  -26.844063
```

```
p <- pnorm(abs(ctable[, "t value"]), lower.tail = FALSE) * 2
(ctable <- cbind(ctable, "p value" = p))
```

```
##                        Value   Std. Error     t value       p value
## price          0.0001750439 3.988336e-05    4.388894  1.139285e-05
## freight_value  0.0017503629 4.510701e-04    3.880468  1.042557e-04
## delivery_time -0.0698643319 2.378093e-03  -29.378297 1.039998e-189
## feedback_time  0.0168543491 2.544338e-03    6.624256  3.490006e-11
## delay_time    -0.0114577769 8.187273e-04  -13.994619  1.681287e-44
## approval_time  0.0077497715 6.067146e-03    1.277334  2.014844e-01
## sentiment      1.8061750692 2.488846e-02   72.570771  0.000000e+00
## 1|2           -2.4580144125 2.059065e-02 -119.375299  0.000000e+00
## 2|3           -2.1199184064 2.005768e-02 -105.691087  0.000000e+00
## 3|4           -1.4946896116 1.935950e-02  -77.207033  0.000000e+00
## 4|5           -0.5045024555 1.879382e-02  -26.844063 9.894265e-159
```

```
#Interpretation of the 1st ordinal logistic model:
#Since the p-value for all the variables <0.05, hence they are statistically significant at 95% CI. The vari
able with the biggest pvalue is the approval time.

#As our predictive variables are continuous they can be interpreted as: E.g. With 1 unit increase in the del
ivery time the log of odds of a customer giving a better review score decreases by 0.069

#The intercepts can be interpreted in the following way: E.g. 1|2 means the log of odds of giving a review o
f 1, versus giving a review of 2,3,4 or 5.

#USing stepAIC to improve the model
step <- stepAIC(olm, direction="both")
```

```
## Start:  AIC=252071.5
## review_score ~ price + freight_value + delivery_time + feedback_time +
##     delay_time + approval_time + sentiment
##
##                 Df    AIC
## - approval_time  1 252071
## <none>             252072
## - freight_value  1 252085
## - price          1 252091
## - feedback_time  1 252115
## - delay_time     1 252268
## - delivery_time  1 253080
## - sentiment      1 258662
##
## Step:  AIC=252071.2
## review_score ~ price + freight_value + delivery_time + feedback_time +
##     delay_time + sentiment
##
##                 Df    AIC
## <none>             252071
## + approval_time  1 252072
## - freight_value  1 252084
## - price          1 252091
## - feedback_time  1 252116
## - delay_time     1 252267
## - delivery_time  1 253085
## - sentiment      1 258665
```

```
step$anova
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## review_score ~ price + freight_value + delivery_time + feedback_time +
##     delay_time + approval_time + sentiment
##
## Final Model:
## review_score ~ price + freight_value + delivery_time + feedback_time +
##     delay_time + sentiment
##
##
##              Step Df Deviance Resid. Df Resid. Dev      AIC
## 1                               110821   252049.5 252071.5
## 2 - approval_time  1 1.633586    110822   252051.2 252071.2
```

```
print(step)
```

```
## Call:
## polr(formula = review_score ~ price + freight_value + delivery_time +
##     feedback_time + delay_time + sentiment, data = total6, Hess = TRUE,
##     method = "logistic")
##
## Coefficients:
##         price freight_value delivery_time feedback_time    delay_time
##  0.0001748223  0.0017503217 -0.0699853660  0.0170874299 -0.0114288096
##     sentiment
##  1.8065426012
##
## Intercepts:
##        1|2        2|3        3|4        4|5
## -2.4608642 -2.1227753 -1.4975629 -0.5073952
##
## Residual Deviance: 252051.16
## AIC: 252071.16
```

```
#As we can see the variable approval time has been removed, and although the AIC has not improved too much,
now the model is now more simple.

(ctable <- coef(summary(step)))
```

```
##                       Value   Std. Error     t value
## price          0.0001748223 3.987916e-05    4.383802
## freight_value  0.0017503217 4.509722e-04    3.881219
## delivery_time -0.0699853660 2.376739e-03  -29.445967
## feedback_time  0.0170874299 2.538474e-03    6.731378
## delay_time    -0.0114288096 8.183619e-04  -13.965471
## sentiment      1.8065426012 2.488729e-02   72.588957
## 1|2           -2.4608641891 2.046992e-02 -120.218582
## 2|3           -2.1227752636 1.993296e-02 -106.495731
## 3|4           -1.4975629234 1.922876e-02  -77.881401
## 4|5           -0.5073951975 1.865700e-02  -27.195963
```

```
p1 <- pnorm(abs(ctable[, "t value"]), lower.tail = FALSE) * 2
(ctable <- cbind(ctable, "p value" = p1))
```

```
##                       Value   Std. Error     t value       p value
## price          0.0001748223 3.987916e-05    4.383802  1.166258e-05
## freight_value  0.0017503217 4.509722e-04    3.881219  1.039343e-04
## delivery_time -0.0699853660 2.376739e-03  -29.445967 1.417896e-190
## feedback_time  0.0170874299 2.538474e-03    6.731378  1.680638e-11
## delay_time    -0.0114288096 8.183619e-04  -13.965471  2.532245e-44
## sentiment      1.8065426012 2.488729e-02   72.588957  0.000000e+00
## 1|2           -2.4608641891 2.046992e-02 -120.218582  0.000000e+00
## 2|3           -2.1227752636 1.993296e-02 -106.495731  0.000000e+00
## 3|4           -1.4975629234 1.922876e-02  -77.881401  0.000000e+00
## 4|5           -0.5073951975 1.865700e-02  -27.195963 7.249713e-163
```

```
#Interpretation of the improved ordinal logistic model:
#All the variables are statistically significant at 95% CI. The variables with the biggest pvalue are now th
e freight value and the price.

#Our predictive variables are continuous so they can be interpreted as: E.g. With 1 unit increase in the del
ivery time the log of odds of a customer giving a better review score decreases by 0.069

#The intercepts can be interpreted in the following way: E.g. 1|2 means the log of odds of giving a review o
f 1, versus giving a review of 2,3,4 or 5.

#Test and train our logisitc model
#Set Testing Criteria -70/30
numberofobs = round(length(total6$review_score) * .7)

#Split Test and Train data
train <- total6[1:numberofobs,]
test <- total6[-(1:numberofobs),]

#Make predictions(Step)
setup2 <- test
setup2[, c("pred.prob")] <- predict(step, newdata=setup2, type="probs")
setup2[, c("pred.prob")] <- predict(step, newdata=setup2, type="class")
setup2$residuals <- residuals(step, type="response")

#Step AIC model confusion matrix
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
confusionMatrix(setup2$pred.prob, test$review_score, positive="TRUE")
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction     1     2     3     4     5
##          1  1010   170   186   142   191
##          2     0     0     0     0     0
##          3     0     0     0     0     0
##          4     0     0     0     0     0
##          5  2760   983  2603  6176 19029
## 
## Overall Statistics
##                                           
##                Accuracy : 0.6027          
##                  95% CI : (0.5974, 0.6079)
##     No Information Rate : 0.578           
##     P-Value [Acc > NIR] : < 2.2e-16       
##                                           
##                   Kappa : 0.1085          
##                                           
##  Mcnemar's Test P-Value : NA              
## 
## Statistics by Class:
## 
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity           0.26790  0.00000  0.00000     0.00   0.9901
## Specificity           0.97663  1.00000  1.00000     1.00   0.1075
## Pos Pred Value        0.59447      NaN      NaN      NaN   0.6031
## Neg Pred Value        0.91252  0.96532  0.91612     0.81   0.8876
## Prevalence            0.11338  0.03468  0.08388     0.19   0.5780
## Detection Rate        0.03038  0.00000  0.00000     0.00   0.5723
## Detection Prevalence  0.05110  0.00000  0.00000     0.00   0.9489
## Balanced Accuracy     0.62227  0.50000  0.50000     0.50   0.5488
```

#Supervised analysis (Machine learning predictive modeling)

```
#Let´s see which is the best model to predict the review score.

library(rattle)
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.2.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(DMwR)
```

```
## Loading required package: grid
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(caret)
library(lattice)
library(e1071)
library(tidyverse)

# 10-fold Cross-Validation
control <- trainControl(method="cv", number=10)
metric <- "Accuracy"

# Linear Discriminant Analysis (LDA)
set.seed(99)
fit.lda <- train(review_score ~., data=total6, method="lda", metric=metric, trControl=control)

# Classfication and Regression Trees (CART)
set.seed(99)
fit.cart <- train(review_score~., data=total6, method="rpart", metric=metric, trControl=control)

# k-Nearest Neighbors (KNN)
set.seed(99)
fit.knn <- train(review_score~., data=total6, method="knn", metric=metric, trControl=control)

# Bayesian Generalized Linear Model - Logistic Regression
set.seed(99)
fit.logi <- train(review_score~., data=total6, method="bayesglm", metric=metric, trControl=control)


# Random Forest
set.seed(99)
fit.rf <- train(review_score~., data=total6, method="rf", metric=metric, trControl=control)

# Gradient Boosting Machines/XGBoost-Linear Model
set.seed(99)
fit.xgb <- train(review_score~., data=total6, method="xgbLinear", metric=metric, trControl=control)

# Gradient Boosting Machines/XGBoost-Tree Model
#set.seed(99)
#fit.xgb.t <- train(review_score~., data=total6, method="xgbTree", metric=metric, trControl=control)

# Select Best Model
# summarize accuracy of models
results <- resamples(list(lda=fit.lda, cart=fit.cart, knn=fit.knn, logi=fit.logi, rf=fit.rf, xgb.l=fit.xgb))
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: lda, cart, knn, logi, rf, xgb.l
## Number of resamples: 10
##
## Accuracy
##             Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## lda   0.59379229 0.59524508 0.59683310 0.59685836 0.59763140 0.60010827
## cart  0.62501128 0.62677912 0.62866552 0.63012486 0.63280177 0.63758571
## knn   0.55724984 0.56071096 0.56317949 0.56274362 0.56502909 0.56677495
## logi  0.04736557 0.04806478 0.04876619 0.04887583 0.04933566 0.05143476
## rf    0.67421508 0.68026891 0.68319422 0.68168943 0.68323710 0.68528377
## xgb.l 0.64395922 0.64459081 0.64779392 0.64699730 0.64841089 0.65114600
##       NA's
## lda      0
## cart     0
## knn      0
## logi     0
## rf       0
## xgb.l    0
##
## Kappa
##             Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## lda   0.10346243 0.10736651 0.10920773 0.11100794 0.11303876 0.12144479
## cart  0.21658049 0.22326337 0.22858232 0.24102989 0.26525950 0.27279618
## knn   0.13439868 0.14378372 0.14639809 0.14592000 0.15109249 0.15257394
## logi  0.01147314 0.01223018 0.01293248 0.01302151 0.01350773 0.01552172
## rf    0.39918976 0.40638249 0.41238886 0.41040655 0.41559723 0.41663931
## xgb.l 0.28490837 0.28977149 0.29356106 0.29305427 0.29650678 0.30221779
##       NA's
## lda      0
## cart     0
## knn      0
## logi     0
## rf       0
## xgb.l    0
```

```
#The best model is Random Forest with a kappa of 0.40

# Summarize the Best Model
print(fit.rf)
```

```
## Random Forest
##
## 110832 samples
##      7 predictor
##      5 classes: '1', '2', '3', '4', '5'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 99749, 99748, 99748, 99749, 99750, 99748, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   2     0.6802187  0.3766649
##   4     0.6816894  0.4104066
##   7     0.6775841  0.4070959
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 4.
```

```
summary(fit.rf)
```

```
##                   Length Class      Mode
## call                   4 -none-     call
## type                   1 -none-     character
## predicted         110832 factor     numeric
## err.rate            3000 -none-     numeric
## confusion             30 -none-     numeric
## votes             554160 matrix     numeric
## oob.times         110832 -none-     numeric
## classes                5 -none-     character
## importance             7 -none-     numeric
## importanceSD           0 -none-     NULL
## localImportance        0 -none-     NULL
## proximity              0 -none-     NULL
## ntree                  1 -none-     numeric
## mtry                   1 -none-     numeric
## forest                14 -none-     list
## y                 110832 factor     numeric
## test                   0 -none-     NULL
## inbag                  0 -none-     NULL
## xNames                 7 -none-     character
## problemType            1 -none-     character
## tuneValue              1 data.frame list
## obsLevels              5 -none-     character
## param                  0 -none-     list
```

Random Forest model appears to be the best choice machine learning model when we treat our dependant variable as a categorical one. We can see it´s kappa it´s the highest in comparison with the rest of the machine learning models.

However, I will choose as the best model for my goal the Ordinal Logistic Regression. The reason for making this choice is that although its accuracy level is not has high as the one we find in the rf model(0.60 Accuracy and Kappa 0.10), it is a good alternative model for interpreting which factors influence in the review score. Random forest model is more complex and less easy for interpretation.