Final Project of Varibale Review text Sentimental Analysis

Yunbai Zhang

11/12/2018

```
library(carData)
library(cluster)
library(car)
library(tidyverse)
```

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

```
## ✔ ggplot2 3.0.0     ✔ purrr   0.2.5
## ✔ tibble  1.4.2     ✔ dplyr   0.7.7
## ✔ tidyr   0.8.1     ✔ stringr 1.3.1
## ✔ readr   1.1.1     ✔ forcats 0.3.0
```

```
## ── Conflicts ────────────────────────────────────────────────────────── tidyv
erse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ✖ dplyr::recode() masks car::recode()
## ✖ purrr::some()   masks car::some()
```

```
library(dplyr)
library(ggplot2)
library(forcats)
library(tidyverse)
library(AER)
```

```
## Loading required package: lmtest
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: sandwich
```

```
## Loading required package: survival
```

```
library(GGally)
```

```
##
## Attaching package: 'GGally'
```

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

```
library(corrgram)
library(stringr)
library(extracat)
library(sentimentr)
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:sentimentr':
##
##     highlight
```

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

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

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
setwd('/Users/yunbaizhang/Desktop')
data = read.csv('DatafinitiElectronicsProductData.csv', header = TRUE)
## Compare review text and review title of each product by using sentimental scores.
```

Compare review text and review title of each product by using sentimental scores.

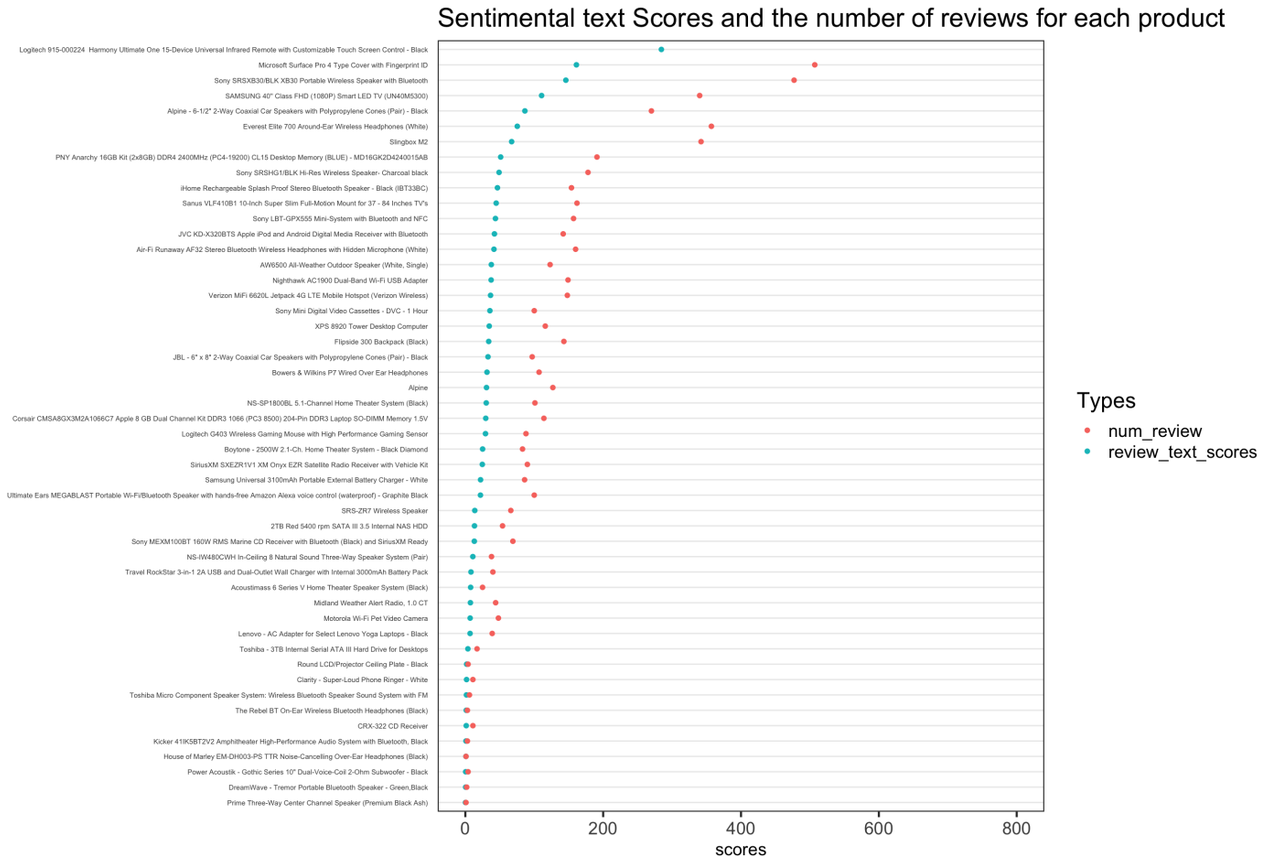
Compare the sentimental scores of review text and the sentimental scores of review title

```
## The sentimental scores of review text
my_text = get_sentences(as.character(data$reviews.text))
sen_text = sentiment_by(my_text)

## The sentimental scores of review title
my_title = get_sentences(as.character(data$reviews.title))
sen_title = sentiment_by(my_title)

sentiment_df = data.frame(data$name, sen_text$ave_sentiment, sen_title$ave_sentiment)
colnames(sentiment_df)[1]<-"name"
colnames(sentiment_df)[2]<-"text_scores"
colnames(sentiment_df)[3]<-"title_scores"

sentiment_table <- sentiment_df %>% select(name = name, review_text_scores = text_scores,   r
eview_title_scores = title_scores)%>% group_by(name) %>% summarise(review_text_scores = sum(r
eview_text_scores),review_title_scores = sum(review_title_scores))

sentiment_table<- data.frame(sentiment_table)

tidy_table = sentiment_table %>% gather(`review_text_scores`,`review_title_scores`, key = 'sc
ore_type', value =scores)

theme_dotplot <- theme_bw(18) +
  theme(axis.text.y = element_text(size = rel(.4)),
        axis.ticks.y = element_blank(),
        axis.title.x = element_text(size = rel(.8)),
        panel.grid.major.x = element_blank(),
        panel.grid.major.y = element_line(size = 0.5),
        panel.grid.minor.x = element_blank())


ggplot(tidy_table, aes(x = scores,
                   y = fct_reorder2(name, score_type, -scores),
                   color = score_type)) +
  geom_point() + ylab("Product Name") + theme_dotplot +
  ggtitle("Sentimental Scores for each product")
```

- 1. Almost coinside together implies high correlation
- 2. Logitech 915-000224 has the highest rank of review text scores and review title scores.
- 3. No-“super”-negative items

What if we compare the review text scores with reviews frequency?

```
my_text = get_sentences(as.character(data$reviews.text))
sen_text = sentiment_by(my_text)
sentiment_df = data.frame(data$name, sen_text$ave_sentiment)
colnames(sentiment_df)[1]<-"name"
colnames(sentiment_df)[2]<-"text_scores"


sentiment_table2 <- sentiment_df %>% select(name = name, review_text_scores = text_scores)%>%
 group_by(name) %>% summarise(review_text_scores = sum(review_text_scores), num_review = n())
sentiment_table2<- data.frame(sentiment_table2)

tidy_table2 = sentiment_table2 %>% gather(`review_text_scores`,`num_review`, key = 'Types', v
alue =scores)

ggplot(tidy_table2, aes(x = scores,
                        y = fct_reorder2(name, Types, -scores),
                        color = Types)) +
  geom_point() + ylab("") + theme_dotplot + xlim(0,800)+
  ggtitle("Sentimental text Scores and the number of reviews for each product")
```



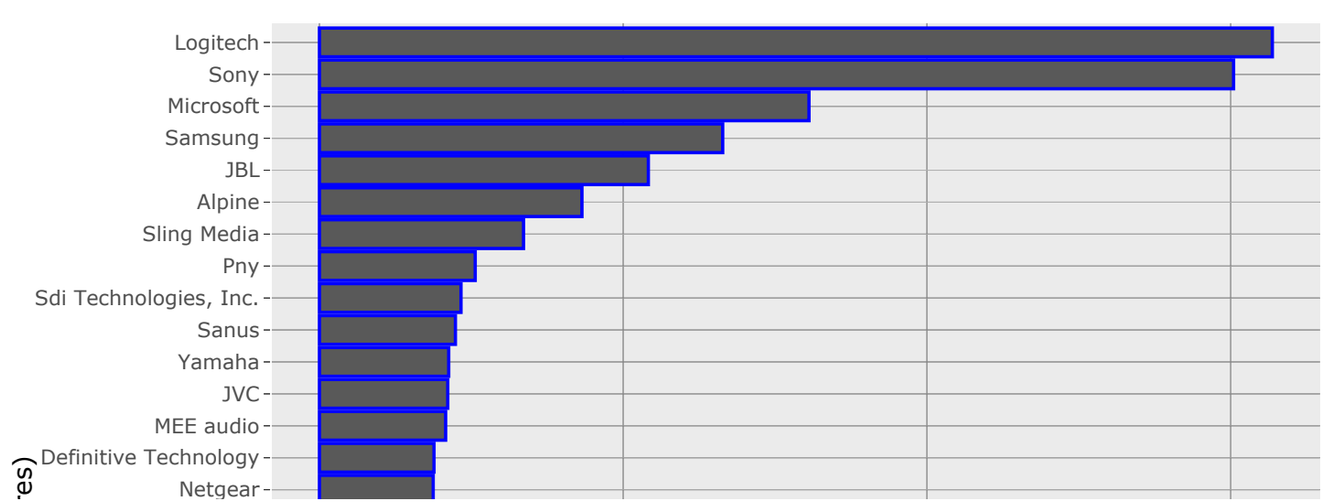

```
cor(sentiment_table2$review_text_scores, sentiment_table2$num_review,  method = "pearson", us
e = "complete.obs")
```

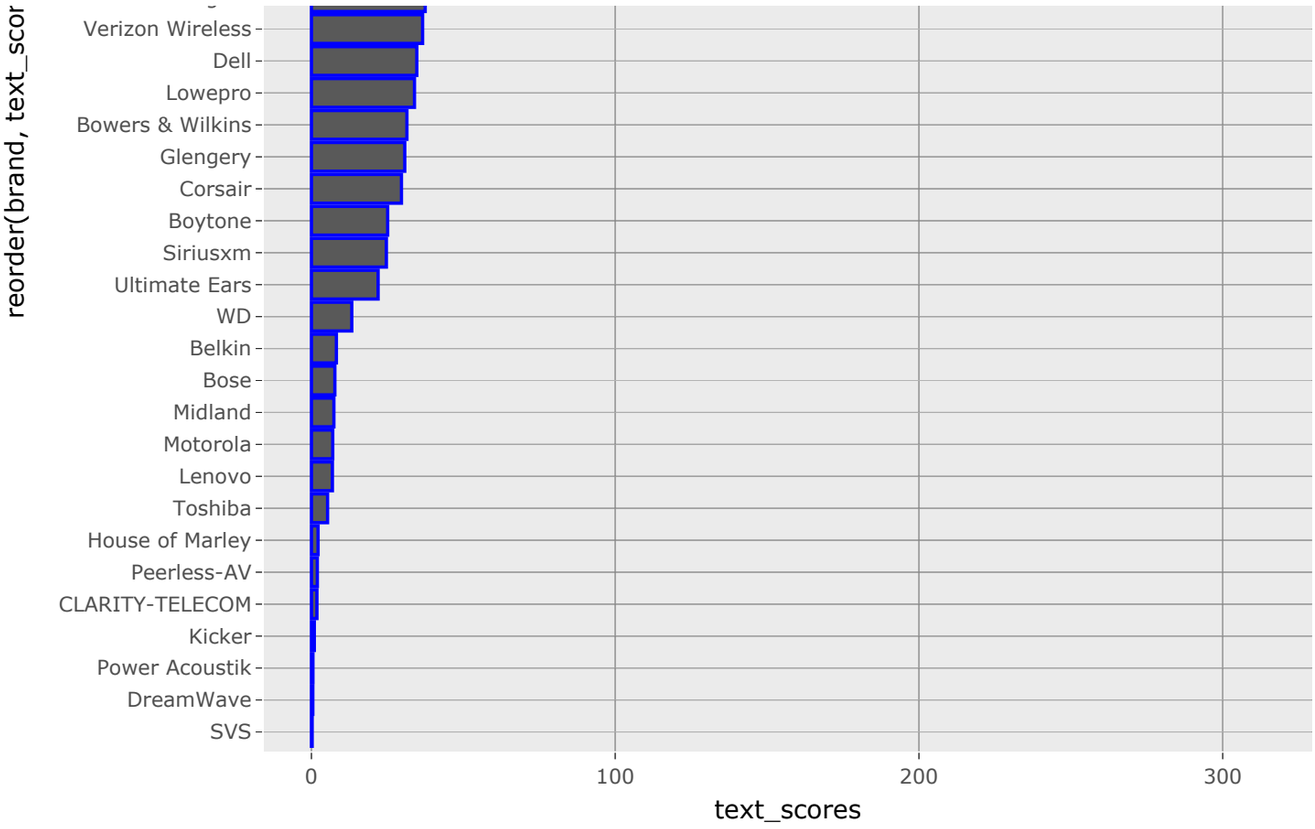
```
## [1] 0.9579697
```

- There is a pattern between number of reviews of each product and its sentimental scores.
- Very high correlation between the number of reviews and people's feedbacks of given products.
- High-frequency of people's reviews upon some items would bring good remarks of these items.

What about brand instead of product name?

```
senti_brand = data.frame(data$brand, sen_text$ave_sentiment)
colnames(senti_brand)[1]<-"brand"
colnames(senti_brand)[2]<-"text_scores"

sentiment_table3 <- senti_brand %>% select(brand = brand, text_scores = text_scores)%>% group
_by(brand) %>% summarise(text_scores = sum(text_scores))
sentiment_table3<- data.frame(sentiment_table3)

p <- ggplot(data=sentiment_table3, aes(x= reorder(brand, text_scores), y= text_scores)) +
  geom_bar(colour='blue', stat="identity") +
  guides(fill='grey')+coord_flip()

ggplotly(p)
```





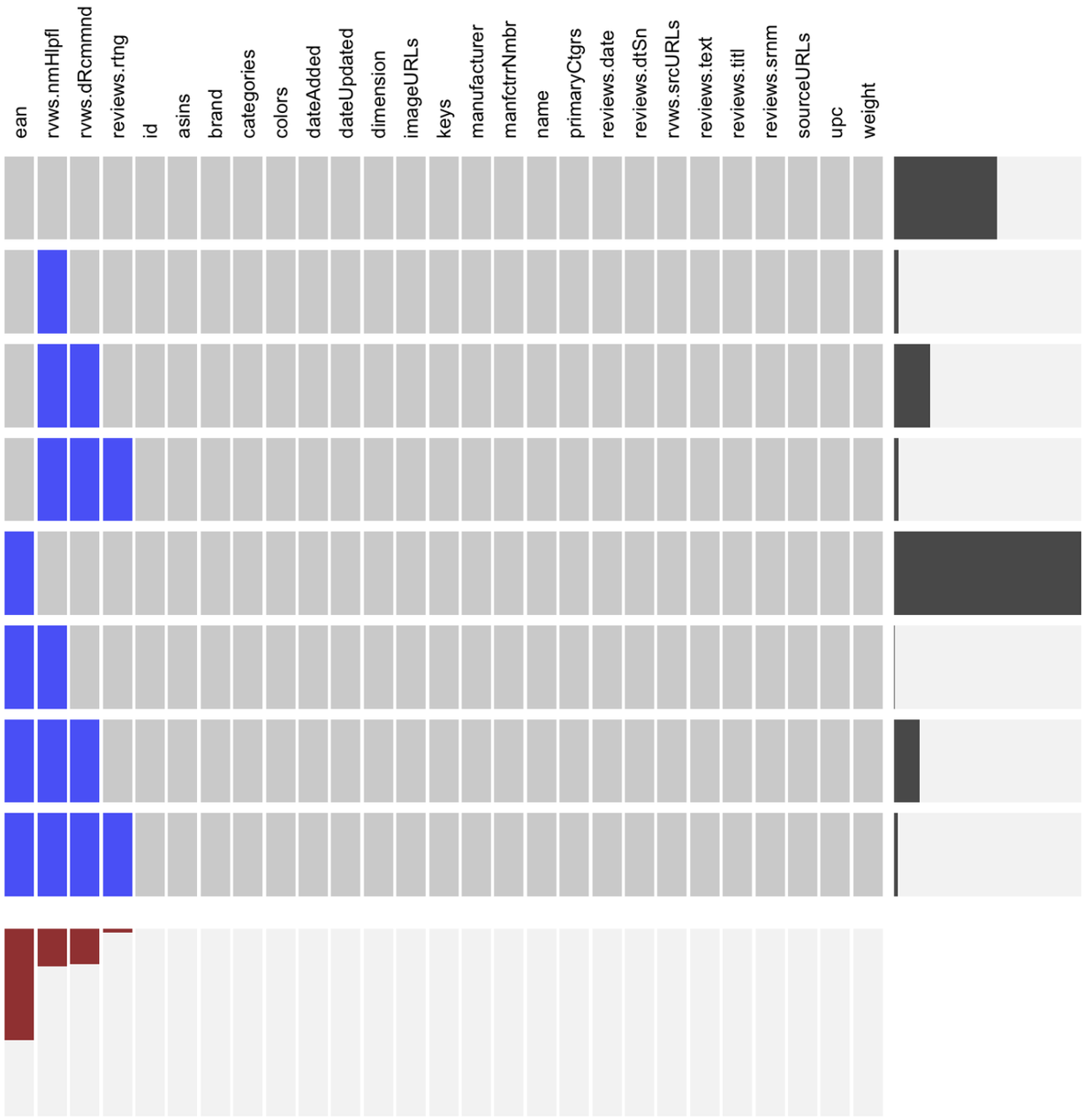
- We can see that logitech was the most popular tech. Sony is on the second rank.

Review's Missing value Analysis

```
miss_table = colSums(is.na(data)) %>%
  sort(decreasing = FALSE)
miss_table
```

```
##                   id                asins                brand
##                    0                    0                    0
##           categories               colors           dateAdded
##                    0                    0                    0
##          dateUpdated            dimension            imageURLs
##                    0                    0                    0
##                 keys         manufacturer  manufacturerNumber
##                    0                    0                    0
##                 name    primaryCategories        reviews.date
##                    0                    0                    0
##     reviews.dateSeen   reviews.sourceURLs        reviews.text
##                    0                    0                    0
##        reviews.title    reviews.username           sourceURLs
##                    0                    0                    0
##                  upc               weight      reviews.rating
##                    0                    0                  164
## reviews.doRecommend   reviews.numHelpful                 ean
##                 1391                 1486                 4348
```

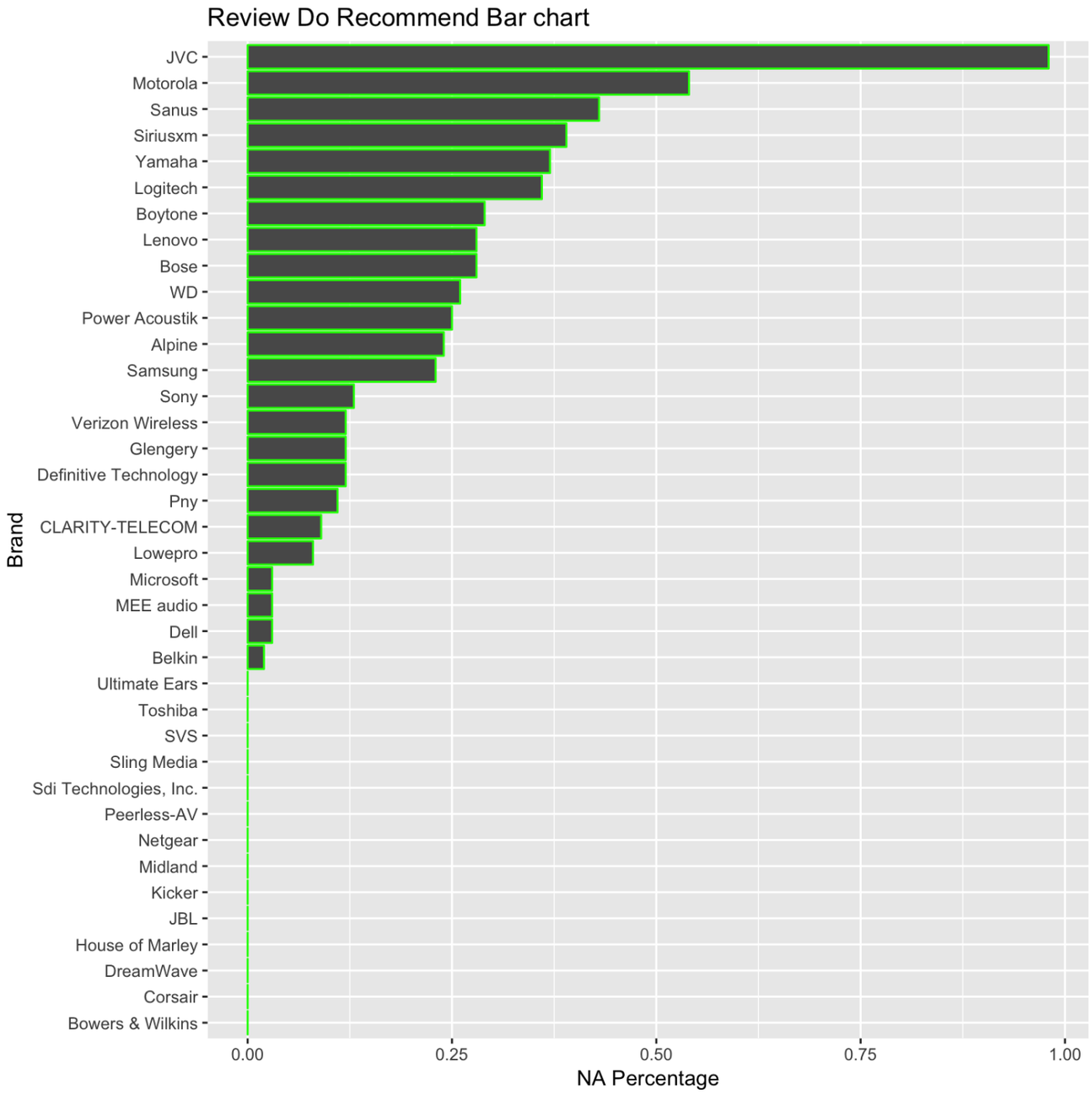
```
visna(data, sort = "c")
```



Which kinds of brands are more likely to have missing reviews.doRecommend or reviews.numHelpful?

Which kinds of product are more likely to have missing data?

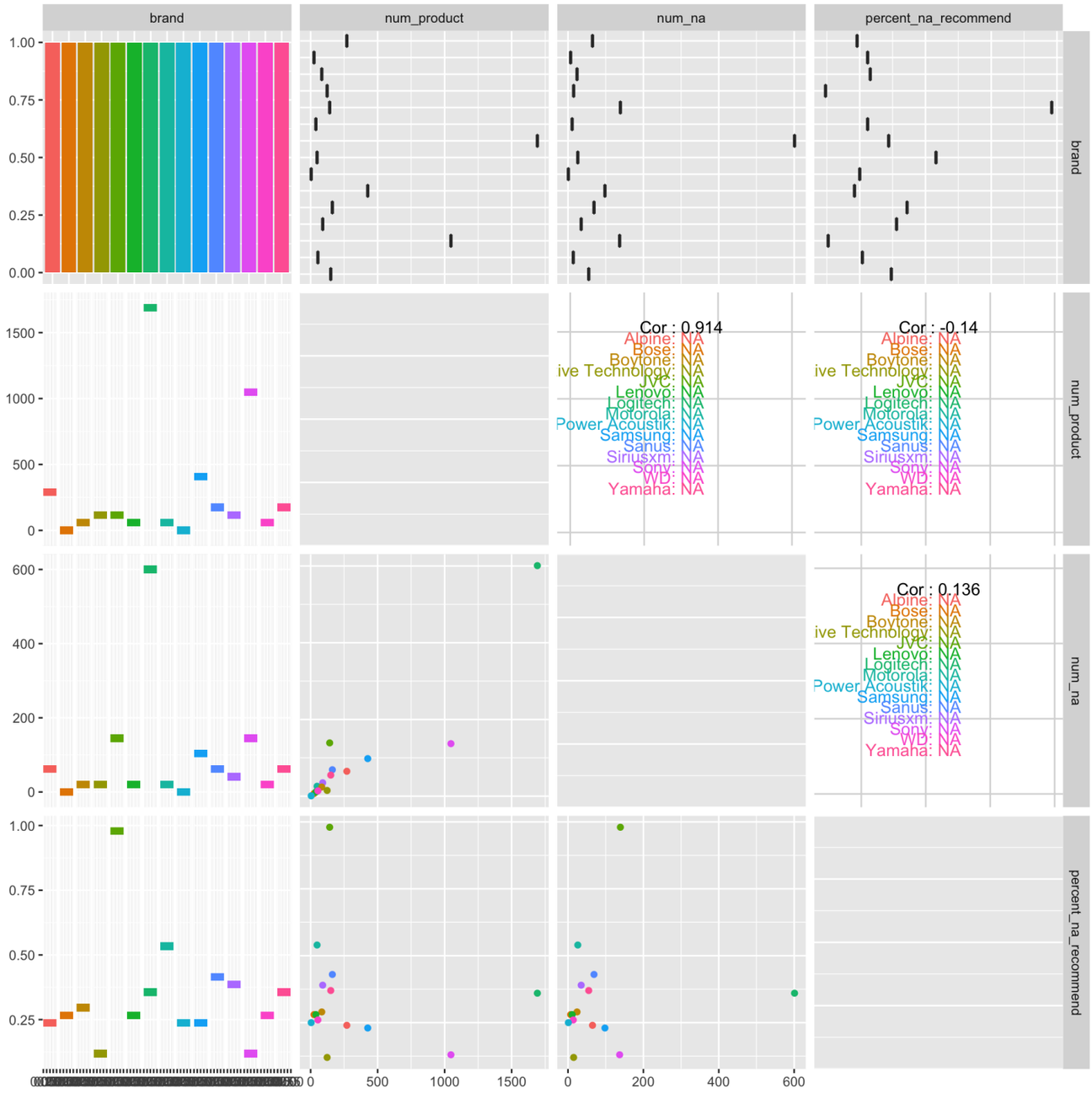
```
percent_missing_doRecomm <- data %>% group_by(brand) %>%
  summarise(num_product = n(), num_na = sum(is.na(reviews.doRecommend))) %>%
  mutate(percent_na_recommend = round(num_na/num_product, 2)) %>%
  arrange(-percent_na_recommend)

percent_missing_doRecomm = data.frame(percent_missing_doRecomm)

p1 <- ggplot(data=percent_missing_doRecomm, aes(x= reorder(brand, percent_na_recommend), y= p
ercent_na_recommend)) +
  geom_bar(colour='green', stat="identity") +
  guides(fill='grey')+coord_flip()+xlab('Brand')+ylab('NA Percentage')+ggtitle('Review Do Rec
ommend Bar chart')
p1
```




```
#ggplotly(p1)
```

```
percent_missing_doRecomm_sub<- percent_missing_doRecomm[1:15,]
percent_missing_doRecomm_sub$brand<- droplevels(percent_missing_doRecomm_sub$brand)
ggpairs(percent_missing_doRecomm_sub, aes(color = brand))
```

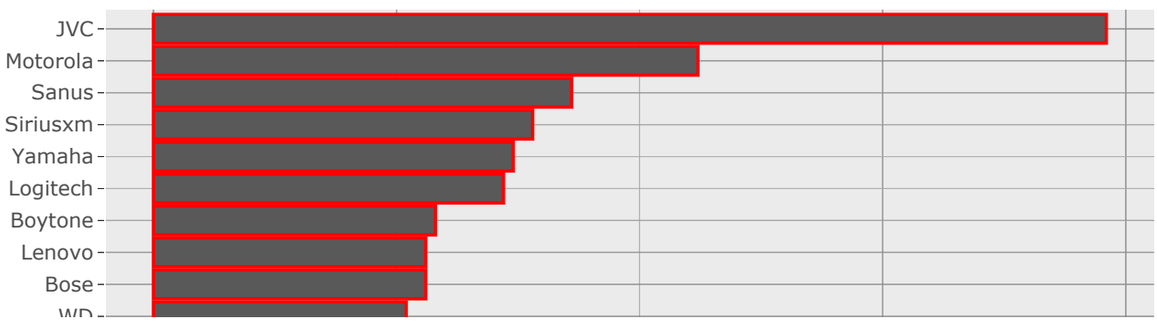
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

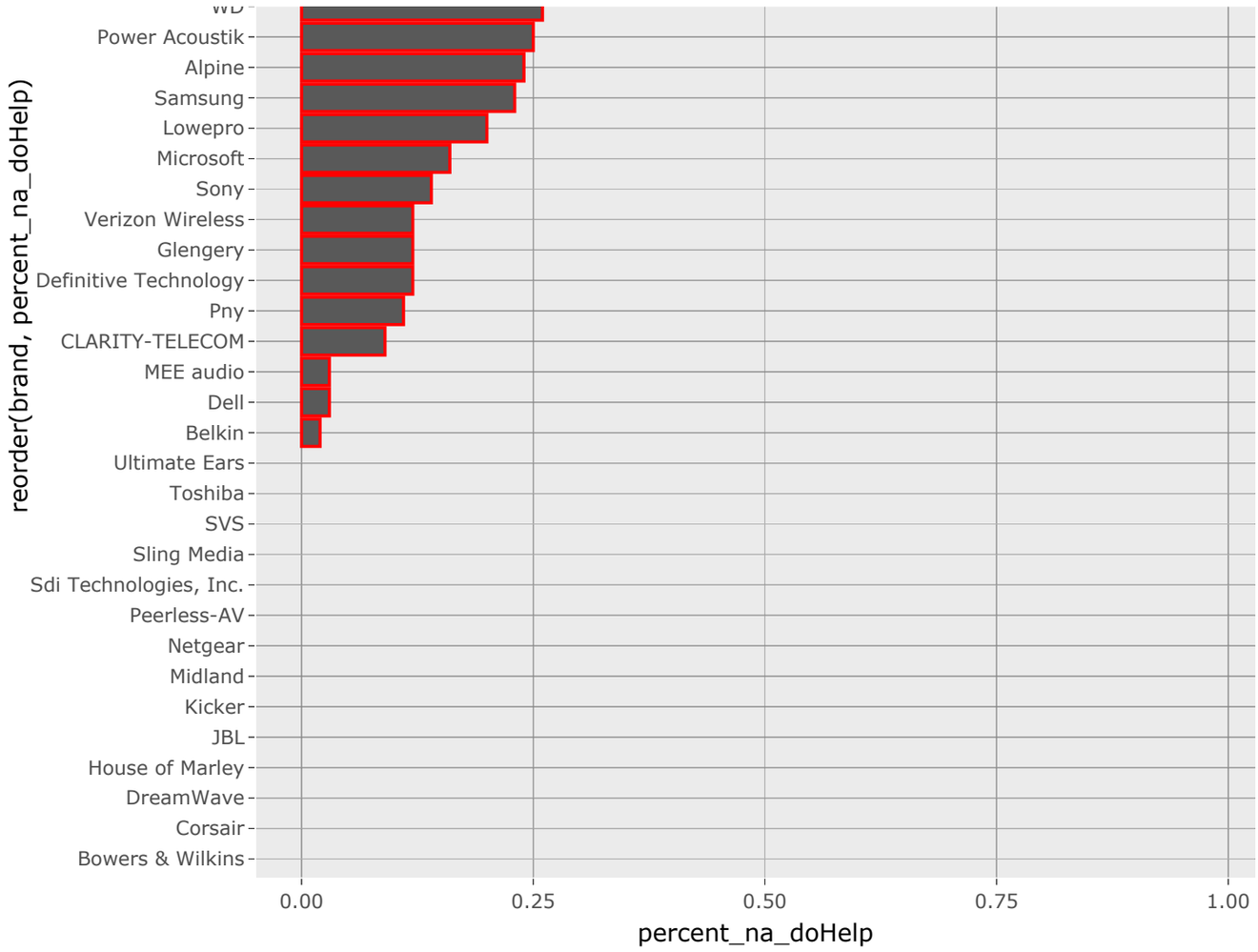


- JVC has the highest percent of missing recommendation, Motorola is on the second, and Sanus is on the third.
- Of the top 15 of the percentage of the missing recommendation, almost all the brand have similiar levels of na percentage and the percentages are between 20% and 35% .

Similar method use for reviews.numHelpful

```
percent_missing_doHelp <- data %>% group_by(brand) %>%
  summarise(num_product = n(), num_na = sum(is.na(reviews.numHelpful))) %>%
  mutate(percent_na_doHelp = round(num_na/num_product, 2)) %>%
  arrange(-percent_na_doHelp)


p2 <- ggplot(data=percent_missing_doHelp, aes(x= reorder(brand, percent_na_doHelp), y= percen
t_na_doHelp)) +
  geom_bar(colour='red', stat="identity") +
  guides(fill='grey')+coord_flip()

ggplotly(p2)
```



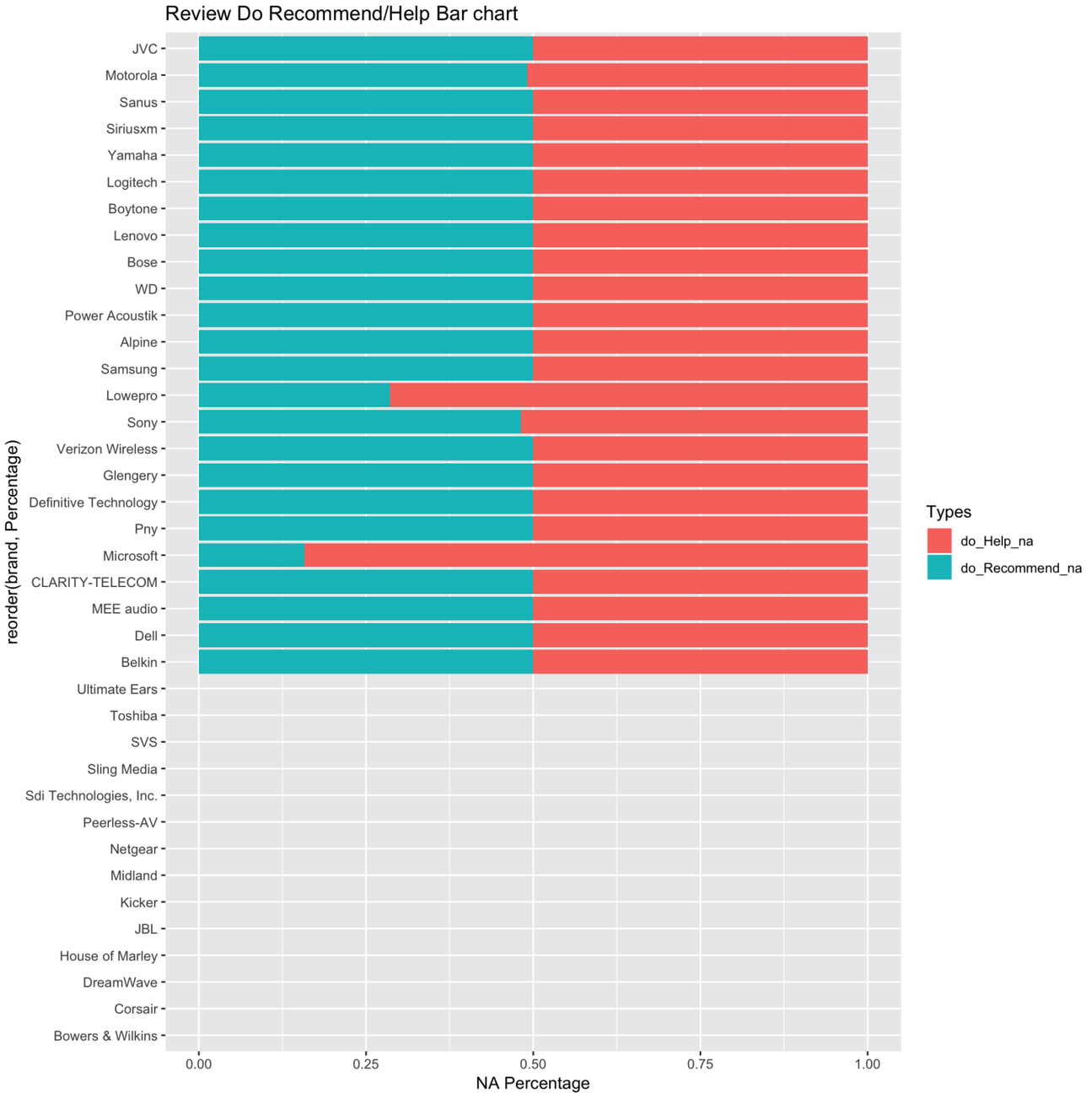


What if we compare Do-Recommend and Do-Help NA data

```
percent_missing_doRecomm2 <- data %>% group_by(brand) %>%
  summarise(num_product = n(), num_na = sum(is.na(reviews.doRecommend))) %>%
  mutate(percent_na_recommend = round(num_na/num_product, 2))
percent_missing_doRecomm = data.frame(percent_missing_doRecomm2)

percent_missing_doHelp2 <- data %>% group_by(brand) %>%
  summarise(num_product = n(), num_na = sum(is.na(reviews.numHelpful))) %>%
  mutate(percent_na_doHelp = round(num_na/num_product, 2))

percent_missing_doHelp2 = data.frame(percent_missing_doHelp2)


compare_na = data.frame(percent_missing_doHelp2$brand, percent_missing_doHelp2$percent_na_doH
elp, percent_missing_doRecomm$percent_na_recommend)


colnames(compare_na)[1]<-"brand"
colnames(compare_na)[2]<-"do_Help_na"
colnames(compare_na)[3]<-"do_Recommend_na"

cor(compare_na$do_Help_na, compare_na$do_Recommend_na)
```

```
## [1] 0.9903805
```

```
tidy_table3 = compare_na %>% gather(`do_Help_na`,`do_Recommend_na`, key = 'Types', value =Per
centage)

p3 <- ggplot(data=tidy_table3, aes(x=reorder(brand, Percentage), y=Percentage, fill=Types)) +
  geom_bar(stat="identity", position='fill')+coord_flip()+ylab('NA Percentage')+ggtitle('Revi
ew Do Recommend/Help Bar chart')
p3
```




```
#ggplotly(p3)
```

- Two variables of missing value is almost the same, except brands Lowepro and Microsoft.