Assignment 11–R

Benjamin Maskell

November 19, 2018

Amazon Reviews

In this problem, you will use R to do further analysis on the Amazon reviews data. Where relevant, you are encouraged to use functions from `dplyr` and `ggformula`.

Load necessary libraries here.

```
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

library(ggformula)

## Loading required package: ggplot2

## Loading required package: ggstance

## 
## Attaching package: 'ggstance'

## The following objects are masked from 'package:ggplot2':
## 
##     geom_errorbarh, GeomErrorbarh

## 
## New to ggformula?  Try the tutorials: 
##  learnr::run_tutorial("introduction", package = "ggformula")
##  learnr::run_tutorial("refining", package = "ggformula")

install.packages("data.table")

## Installing package into 'C:/Users/benja/R/win-library/3.5'
## (as 'lib' is unspecified)
## package 'data.table' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\benja\AppData\Local\Temp\RtmpSsFFaW\downloaded_packages
```

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

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
install.packages("e1071")
```

```
## Installing package into 'C:/Users/benja/R/win-library/3.5'
## (as 'lib' is unspecified)
```

```
## package 'e1071' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\benja\AppData\Local\Temp\RtmpSsFFaW\downloaded_packages
```

```
library(e1071)
```

Reading and cleaning the data

a. In the Python assignment for homework 11, you created a `.csv` file with information about Amazon reviews. Using what you learned about reading files efficiently, read this data set into R.

```
col1<- fread("food_pandas.csv",sep = ",", select = c("Num_helpful") )
col2<- fread("food_pandas.csv",sep = ",", select = c("Num_voters") )
col3<- fread("food_pandas.csv",sep = ",", select = c("Product_ID") )
col4<- fread("food_pandas.csv",sep = ",", select = c("Review_score") )
col5<- fread("food_pandas.csv",sep = ",", select = c("Review_length") )
col6<- fread("food_pandas.csv",sep = ",", select = c("num_exclamations") )
col7<- fread("food_pandas.csv",sep = ",", select = c("fract_help") )

food_df = data.frame(cbind(col1, col2, col3, col4, col5, col6, col7))
```

b. Examine the helpful fraction column for unrealistic values. (There should be more than 0 but fewer than 100 unrealistic values. If this isn't what you got, double-check your Python code.)

- Set unrealistic values to missing.
- Also set to missing the corresponding value of the total votes column.
- (Because you computed the helpful fraction from the columns "helpful votes" and "total votes," an unrealistic value of "helpful fraction" means that at least one of the other two values must be unrealistic. Because we don't know which one, the safest course is to set them both to missing.)

```
summary(food_df)

##    Num_helpful        Num_voters       Product_ID         Review_score
##  Min.   :  0.000   Min.   :  0.000   Length:284225      Min.   :1.000
##  1st Qu.:  0.000   1st Qu.:  0.000   Class :character   1st Qu.:4.000
##  Median :  0.000   Median :  1.000   Mode  :character   Median :5.000
##  Mean   :  1.746   Mean   :  2.226                      Mean   :4.182
##  3rd Qu.:  2.000   3rd Qu.:  2.000                      3rd Qu.:5.000
##  Max.   :866.000   Max.   :878.000                      Max.   :5.000
##
##  Review_length     num_exclamations    fract_help
##  Min.   :   12.0   Min.   : 0.0000   Min.   :0.00
##  1st Qu.:  179.0   1st Qu.: 0.0000   1st Qu.:0.60
##  Median :  302.0   Median : 0.0000   Median :1.00
##  Mean   :  435.7   Mean   : 0.7555   Mean   :0.78
##  3rd Qu.:  528.0   3rd Qu.: 1.0000   3rd Qu.:1.00
##  Max.   :16952.0   Max.   :71.0000   Max.   :1.50
##                                      NA's   :135081

food_df$Num_voters[which(food_df$fract_help > 1)] <- NA
food_df$fract_help[food_df$fract_help > 1] <- NA
```

c. **Write 1-2 sentences** to document how many unrealistic values you found, what made them unrealistic, and the fact that you set those values to missing.

I only found one unrealistic value: in the case where the fraction of helpful voters exceeded 1. This is unrealistic because the number of helpful votes should not be more than total votes. Setting it to missing makes sense because this is most likely due to a data entry error.

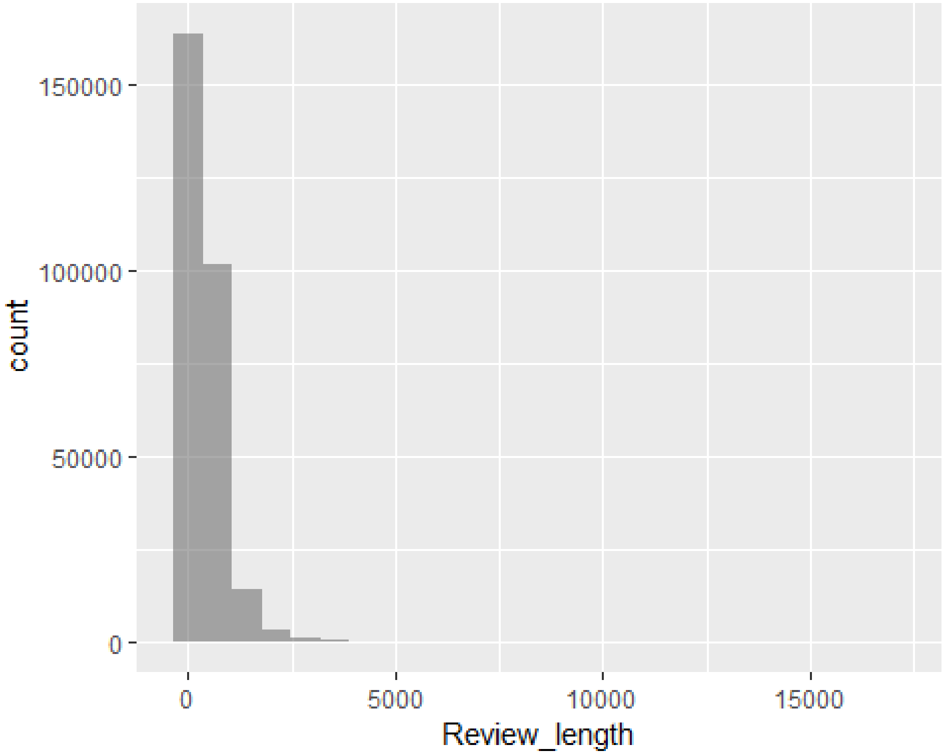
Investigating helpful reviews

d. Create a new variable that describes whether more than 50% of people who voted on a review considered it helpful. We will call these helpful reviews.

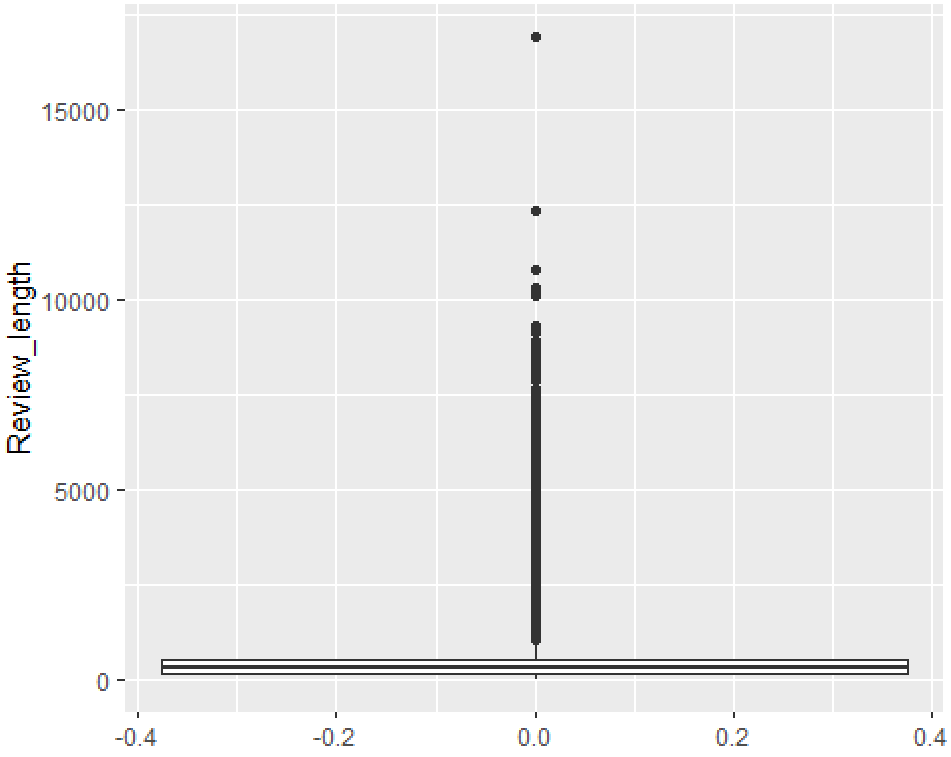
```
food_df <- food_df %>%
  mutate(helpful_reviews = case_when(
    fract_help >= .5 ~ TRUE,
    fract_help < .5 ~ FALSE
    ))
```

e. In this question, you will investigate the question, “Are helpful reviews longer than unhelpful ones?” Start by making appropriate graphical summaries to help you decide whether to transform the review length. Then do a hypothesis test of whether the typical length of helpful reviews is longer than the typical length of unhelpful reviews.

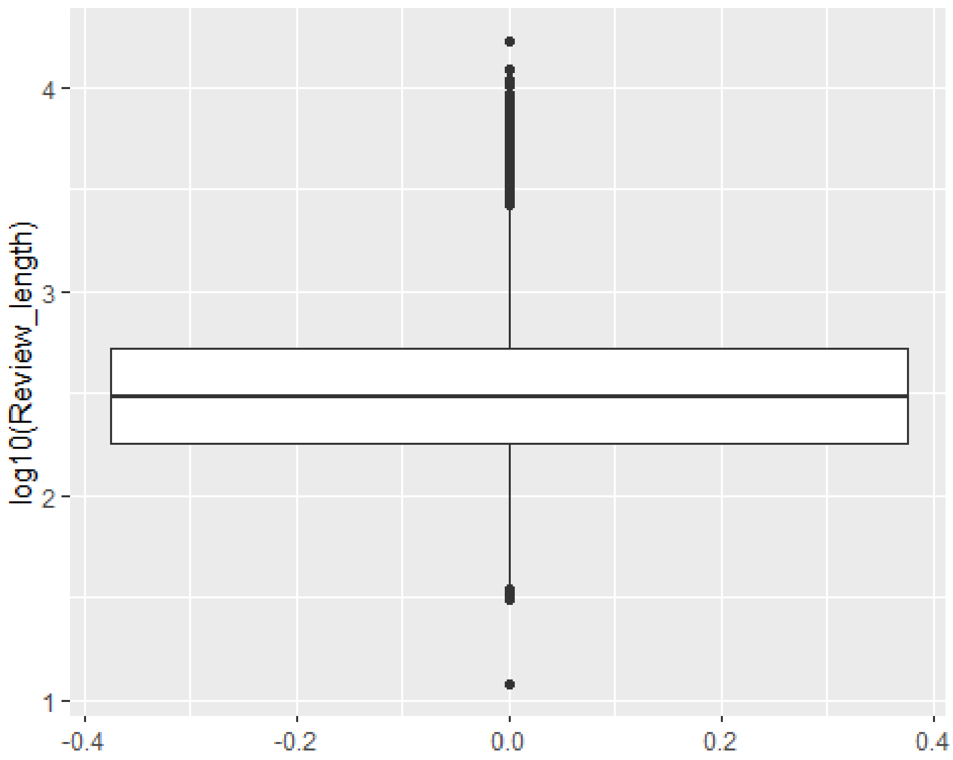
```
gf_histogram(~Review_length, data = food_df)
```



```
gf_boxplot(~Review_length, data = food_df)
```



```
gf_boxplot(~log10(Review_length), data = food_df)
```



```
skewness(food_df$Review_length)

## [1] 4.626502

z_scale <- scale(food_df$Review_length,center= TRUE, scale=TRUE)

z_scale <- cbind(z_scale, food_df$helpful_reviews)

z_scale[is.na(z_scale)] <- ""

num.rows = dim(z_scale)[1]

helpful_list <- numeric()
for (i in 1:num.rows){
  if (z_scale[i,2] == 1){
    helpful_list <- c(helpful_list, z_scale[i,1])
  }
}

not_list <- numeric()
for (i in 1:num.rows){
  if (z_scale[i,2] == 0){
    not_list <- c(not_list, z_scale[i,1])
  }
}
```

```
helpful_list <- as.numeric(helpful_list)
not_list <- as.numeric(not_list)

t.test(helpful_list, not_list, alternative = "two.sided")
```

```
##
##  Welch Two Sample t-test
##
## data:  helpful_list and not_list
## t = 18.956, df = 37277, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1297910 0.1597268
## sample estimates:
##    mean of x    mean of y
##  0.143171532 -0.001587358
```

State your conclusion in context.

The p-value is such that we can reject the null hypothesis, which states that the true diference in means IS equal to 0. Based on the findings from the above t-test, the mean review length for reviews rated as helpful are likely to be different than the length of reviews of those rated as unhelpful (unhelpful is defined as less than half of respondents rated the review as helpful).

Relationship between reviews and votes

In this part of the assignment, you will investigate whether there is a relationship between the number of reviews a product has and the number of times the reviews have been voted on (as helpful vs. unhelpful).

- Intuitively, it seems that products that have been on Amazon longer may have more reviews, and may also have accumulated more votes on their reviews. You will investigate whether this is supported by the data.

f. For each product ID, find the maximum number of votes received by any of the product's reviews. Also count the number of reviews for each product ID.

```
productID_DF <- food_df[,c(2,3)]

#To find the max number of votes by product ID:

max_votes <- aggregate(productID_DF$Num_voters, by =
list(productID_DF$Product_ID), max)

#To count the number of reviews by product ID:

num_productreviews <- productID_DF %>%
                        group_by(Product_ID) %>%
                        summarise(n_distinct(Num_voters))
```

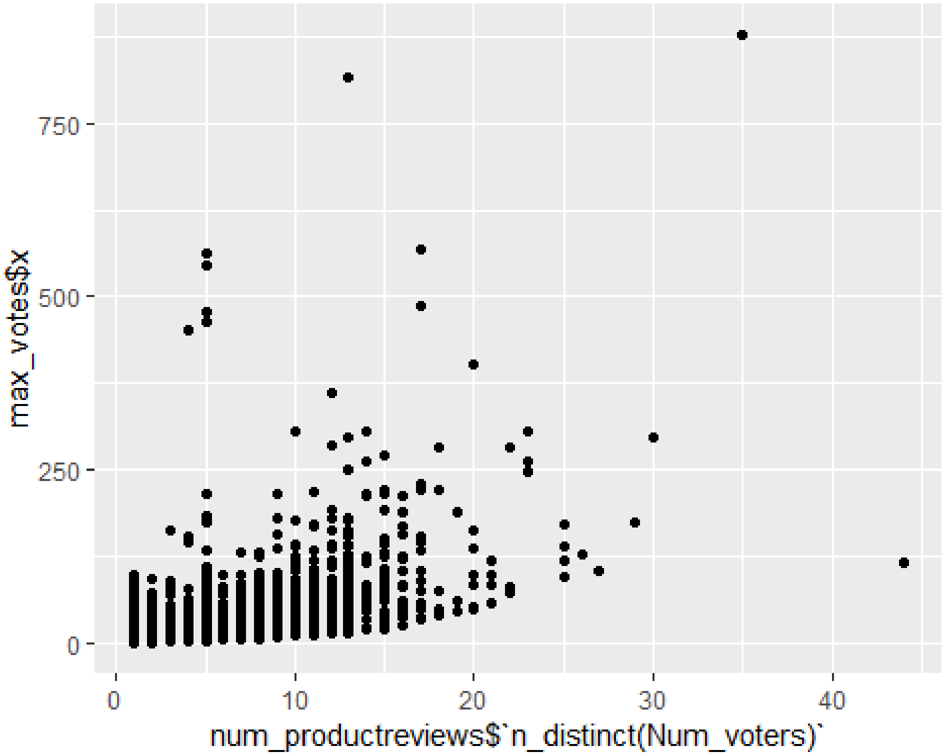
Hints to help you check your work:

- The number of elements in each of the resulting tbls or vectors should equal the total number of unique product IDs.
- The sum of the number of reviews for each product ID should equal the total number of reviews.

g. Make a scatterplot of max number of votes as a function of number of reviews.

```
gf_point(max_votes$x~num_productreviews$`n_distinct(Num_voters)`)

## Warning: Removed 1 rows containing missing values (geom_point).
```



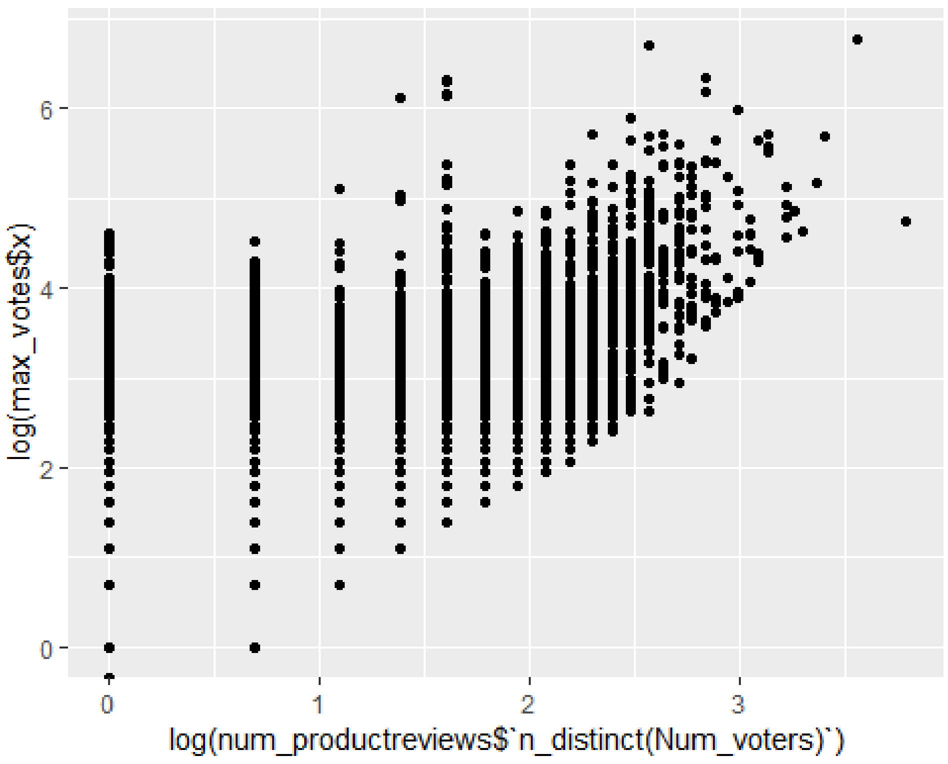
Is there a visible trend? If so, **describe it.**

There seems to be a positive relationship between the two variables, but there are a fair number of outliers that could be transformed to better assess the relationship. From this scatter plot, it seems that the number of product reviews positively correlates with the number of votes (reviews) a product receives.

h. Histograms of the review counts and number of votes indicate that both variables are right-skewed. (You can check this for yourself.) So, a log scale might be helpful in investigating the relationship between them. Modify your scatterplot to use log scales on both axes.

```
gf_point(log(max_votes$x)~log(num_productreviews$`n_distinct(Num_voters)`))

## Warning: Removed 1 rows containing missing values (geom_point).
```



Is there a visible trend? If so, **describe it.** Does this tell us anything about the relationship between max votes and number of reviews *without* the log scale?

With the log scale, the above scatter plot makes a relationship between the two variables seem more likely. I draw this conclusion because I can see that as the number of product reviews increases, the number of votes it receives also increases. the difference between the two scatter plots (one with a log scale and one without) tells us that without trasnforming the data, as we did with log, there are a number of outliers in the data that need to be studied.