Dataquest Guided Project: Creating An Efficient Data Analysis Workflow (part 2)

Cindy Zhang

8/14/2020

Contents

Introduction

This is my solution to Dataquest's Guided Project from Course 4 (Specialized Data Processing in R: Strings and Dates), which evaluates new data related to the data from part 1 and answers the question of whether a book company's new program was successful at increasing sales and improving review quality.

More details, such as descriptions for variables, can be found in the "ReadMe" file of this project's repository in GitHub.

Findings

Step 1: Data Exploration

I loaded the data as a data frame and examined its dimensions, column titles, and column types:

```
booksales_df <- data.frame(read.csv("sales2019.csv"))
dim(booksales_df)
```

```
## [1] 5000    5
```

```
colnames(booksales_df)
```

```
## [1] "date"                  "user_submitted_review" "title"
## [4] "total_purchased"       "customer_type"
```

```
for (i in colnames(booksales_df)) {
  print(class(booksales_df[[i]]))
}
```

```
## [1] "character"
## [1] "character"
## [1] "character"
## [1] "integer"
## [1] "character"
```

There are 5000 rows and 5 columns in the dataframe. Each of the columns in `booksales_df` seems to represent one entry per unique customer that purchased books from the company. I ran a for loop to display which columns had NA values and how many:

```
for (i in colnames(booksales_df))
  print(sum(is.na(booksales_df[[i]])))
```

```
## [1] 0
## [1] 456
## [1] 0
## [1] 718
## [1] 0
```

The results show that the `user_submitted review` and `total_purchased` columns each contain 456 and 718 NA values, respectively.

Step 2: Handling Missing Data

I handled the NA values for each column differently. First, I created a new dataframe that filters out the NA values from the `review` column:

```
booksales_df_filter <- booksales_df %>%
  filter(!(is.na(user_submitted_review)))
```

Filtering out NA values removed 456 observations from the dataset, which is about 9.12 percent of the original dataset removed. This may have an impact on calculations performed later in this project.

Since the focus of this project is to determine the program's effect on sales, I created a new column within `booksales_df_filter` that filled in missing values from the `total_purchased` column with the average of total number of books purchased in the dataset, which will serve as an estimate stand-in for the missing values.

```
booksales_df_filter <- booksales_df_filter %>%
  mutate(total_purchased_fill = ifelse(is.na(total_purchased) == TRUE, mean(total_purchased, na.rm=TRUE)
booksales_df_filter$total_purchased=NULL
```

Step 3: Processing Review Data

To analyze the text data in the `user_submitted_review` column, I first examined what the unique values in the column are:

```
print(unique(booksales_df_filter$user_submitted_review))
```

```
##  [1] "it was okay"
##  [2] "Awesome!"
##  [3] "Hated it"
##  [4] "Never read a better book"
##  [5] "OK"
##  [6] "The author's other books were better"
##  [7] "A lot of material was not needed"
##  [8] ""
##  [9] "Would not recommend"
## [10] "I learned a lot"
```

The print results above show a range of positive to negative reviews. I wrote a binary function that detects whether reviews include positive language (e.g., Awesome or okay) and prints the result "Positive" or "Not positive."

```
positive_review <- function(x) {
 string_pos <- str_detect(x, "okay|OK|Awesome|I learned a lot|Never read a better book")
  is_positive <- case_when(
    string_pos == TRUE ~ "Positive",
    string_pos == FALSE ~ "Not positive"
 )
}
```

I then applied the `positive_review` function to the `user_submitted_review` column and created a new column that prints whether a review is positive or not positive:

```
booksales_df_filter <- booksales_df_filter %>%
  mutate(is_positive_review = positive_review(booksales_df_filter$user_submitted_review)
  )
head(booksales_df_filter)
```

```
##       date    user_submitted_review                              title
## 1  5/22/19              it was okay Secrets Of R For Advanced Students
## 2 11/16/19                 Awesome!                       R For Dummies
## 3  6/27/19                 Awesome!                       R For Dummies
## 4  11/6/19                 Awesome!    Fundamentals of R For Beginners
## 5  7/18/19                 Hated it    Fundamentals of R For Beginners
## 6  1/28/19 Never read a better book Secrets Of R For Advanced Students
##   customer_type total_purchased_fill is_positive_review
## 1      Business                    7           Positive
```

```
## 2      Business                    3           Positive
## 3    Individual                    1           Positive
## 4    Individual                    3           Positive
## 5      Business                    4       Not positive
## 6      Business                    1           Positive
```

Step 4: Comparing Book Sales Between Pre- and Post-Program Sales

Some last data cleaning steps are required before performing any kind of analysis on this data. First, the `date` column was listed as a character/string type variable back in Step 1. For this analysis, the dates need to be converted into quantitative values using the `lubridate` package. Second, to answer the question of whether the program worked, I created a new column that notes whether a purchase was made before or after July 1, 2019, the first day of the sales/review improvement program. Finally, I extracted a summary table from the data that lists the total sum of books purchased before and after July 1, 2019.

```
mdy(booksales_df_filter$date)
```

```
booksales_df_filter <- booksales_df_filter %>%
  mutate(before_after = if_else(booksales_df_filter$date<2019-07-01, "Before", "After"))
booksales_sum_table <- booksales_df_filter %>%
  group_by(before_after) %>%
  summarize(
    purchase_sum = sum(total_purchased_fill)
  )
booksales_sum_table
```

```
## # A tibble: 2 x 2
##   before_after purchase_sum
##   <chr>               <dbl>
## 1 After              10625.
## 2 Before              7563.
```

Based on the table above, total book purchases increased after July 1, 2019 by 3062.

Step 5: Comparing Book Sales Within Customer Type

I repeated the code creating a summary table in Step 4 but included `customer_type` as a grouping variable:

```
booksales_sum_table_sub <- booksales_df_filter %>%
  group_by(before_after, customer_type) %>%
  summarize(
    purchase_sum = sum(total_purchased_fill)
  )
booksales_sum_table_sub
```

```
## # A tibble: 4 x 3
## # Groups:   before_after [2]
##   before_after customer_type purchase_sum
##   <chr>        <chr>                <dbl>
## 1 After        Business             7399.
```

```
## 2 After        Individual           3226.
## 3 Before       Business             5134.
## 4 Before       Individual           2429.
```

Based on the table above, total book purchases by businesses increased by 2265 while total book purchases by individuals increased by 797.

Step 6: Comparing Review Sentiment Between Pre- and Post-Program Sales

I created another summary table, this time measuring the total of positive reviews before and after July 1, 2019:

```
bookreviews_sum_table <- booksales_df_filter %>%
  group_by(before_after) %>%
  summarize(
    pos_review_sum = sum(is_positive_review == "Positive")
  )
bookreviews_sum_table
```

```
## # A tibble: 2 x 2
##   before_after pos_review_sum
##   <chr>                <int>
## 1 After                 1321
## 2 Before                 941
```

Based on the table above, positive reviews increased by 380, suggesting that review sentiment improved after the program was created.

Conclusion

The improvement program appears to have worked as both total purchases (in total and at the customer level among businesses and individuals) and positive reviews increased after the program was implemented on July 1, 2019. However, anyone familiar with statistics knows that any number of confounding factors not represented in the data or even hidden in the data can introduce bias that renders the findings inaccurate. For a more accurate answer as to whether this program improved sales, I would conduct a regression analysis on a more comprehensive dataset.

Thanks to everyone who's been reading so far and stay tuned for more projects!