# Guided Project: Creating an Efficient Data Analysis Workflow, part 2

Mariby Cruz

2025-03-25

The book company I'm working for launched a new program encouraging customers to buy more books on July 1st, 2019, and it wants to know if this new program was successful at increasing sales and improving review quality. This guided project will focus on using the packages `purrr` and `stringr`.

## Data Exploration

There are 5,000 purchases in the sales2019 dataset with information on the date the purchase was made, whether the user submitted a review, the title of the book purchased, how many books purchased, and the type of customer that made the purchase (ie. Business or Individual).

All the columns, except for total amount purchased, are in a character or string format. This is a hint to the type of data manipulation that will take place in this project: string manipulation.

```
library(readr)
library(tibble)
library(stringr)
library(purrr)
library(dplyr)
library(lubridate)
```

```
sales_df <- read.csv("~/git/DataQuest_guided_projects/sales2019.csv")
glimpse(sales_df)
```

```
## Rows: 5,000
## Columns: 5
## $ date                   <chr> "5/22/19", "11/16/19", "6/27/19", "11/6/19", "7/~
## $ user_submitted_review <chr> "it was okay", "Awesome!", "Awesome!", "Awesome!~
## $ title                  <chr> "Secrets Of R For Advanced Students", "R For Dum~
## $ total_purchased        <int> 7, 3, 1, 3, NA, 1, 5, NA, 7, 1, 7, NA, 3, 2, 0, ~
## $ customer_type          <chr> "Business", "Business", "Individual", "Individua~
```

## Handling Missing Data

```
#glimpse of the data showed missing values for a column. Created a for loop to find out how many column

col_missing_counts <- c()
for (col_name in names(sales_df)) {
```

```
  #sum up the counts of NA for each column
  col_missing_counts[col_name] <- sum(is.na(sales_df[col_name]))
}
col_missing_counts
```

```
##                  date user_submitted_review                 title
##                     0                   456                     0
##       total_purchased         customer_type
##                   718                     0
```

This guided project will handle missing data in the sense that we want to keep as much sales data as possible. So rows with missing customer reviews will be removed. For missing values in total books purchased, they will be replaced with the dataset's average (imputation).

```
#removing NA rows from customer review column
sales_no_na_df <- sales_df %>%
  filter(!is.na(user_submitted_review))
dim(sales_no_na_df) #a total of 456 rows were removed
```

```
## [1] 4544    5
```

```
#calculating average books purhcased to replaced NA values with
avg_books_purchased <- mean(sales_no_na_df$total_purchased, na.rm = TRUE)

#complete dataset created
sales_complete <- sales_no_na_df %>%
  mutate(
    total_purchased = if_else(is.na(total_purchased), avg_books_purchased , total_purchased)
  )
```

## Processing Review Data

Customer reviews are entered as sentences which translates to one whole string. This part of the project focuses on identifying whether a review was positive or negative. Because the review is one whole string, the column will need to be parse out the string into multiple string words. From there, the reviews can be classified as positive or negative.

```
is_positive <- function(review) {
review_positive = case_when(
str_detect(review, "Awesome") ~ TRUE,
str_detect(review, 'OK') ~ TRUE,
str_detect(review, 'learned a lot') ~ TRUE,
str_detect(review, 'okay') ~ TRUE,
str_detect(review, 'Never') ~ TRUE,
TRUE ~ FALSE # review did not contain the above phrases
)
}
sales_complete<- sales_complete %>% mutate(review_is_positive = unlist(map(user_submitted_review, is_pos
```

## Comparing Book Sales Between Pre- and Post-Program Sales

A few more steps are needed to prepare the data and answer the question: Was the new book program effective in increasing book sales?

Background: The program started on July 1, 2019 and the data contains all of the sales for 2019.
1) Convert dates from strings to dates
2) Sort the data that's pre and post program sales
3) The analysis should be in a neat form so it's easy to understand

```
sales_complete$date_conv <- mdy(sales_complete$date)

sales_complete <- sales_complete %>%
  mutate(
    program_group = if_else(date_conv < "2019-07-01", "Pre", "Post")
  )

# Using group_by() and summarize() to create a summary tibble
program_sales_summary_table <- sales_complete %>%
  group_by(program_group) %>%
  summarize(
    sum_of_purchases = round(sum(total_purchased), 0)
  )

print(program_sales_summary_table)
```

```
## # A tibble: 2 x 2
##   program_group sum_of_purchases
##   <chr>                    <dbl>
## 1 Post                      9073
## 2 Pre                       9115
```

Based on the summary table, the sales before the program were higher than after the program was implemented.

## Comparing Book Sales by Customer Type

There is a sub-analysis that can be done to check if a certain customer type responded better to the program.

```
customer_sales_summary_table <- sales_complete %>%
  group_by(program_group, customer_type) %>%
  summarize(
    sum_of_purchases = round(sum(total_purchased), 0)
  )

print(customer_sales_summary_table)
```

```
## # A tibble: 4 x 3
## # Groups:   program_group [2]
##   program_group customer_type sum_of_purchases
##   <chr>         <chr>                    <dbl>
## 1 Post          Business                  6310
```

```
## 2 Post          Individual              2763
## 3 Pre           Business                6223
## 4 Pre           Individual              2892
```

It looks like businesses bought more books after the program's implementation by ~100.

## Comparing Review Sentiment Between Pre- and Post-Program Sales

Final question to answer is: did review scores improve as a result of the program?

```
#Summary table that compares the number of positive reviews before and after July 1, 2019

review_sentiment_summary_table <- sales_complete %>%
  group_by(program_group, review_is_positive) %>%
  summarize(
    sum_of_purchases = round(sum(total_purchased), 0)
  )
print(review_sentiment_summary_table)
```

```
## # A tibble: 4 x 3
## # Groups:   program_group [2]
##   program_group review_is_positive sum_of_purchases
##   <chr>         <lgl>                         <dbl>
## 1 Post          FALSE                          4589
## 2 Post          TRUE                           4483
## 3 Pre           FALSE                          4586
## 4 Pre           TRUE                           4528
```

Based on the summary table, there wasn't much of a difference on review sentiment before and after the program was implemented.

## Further Steps

- We filled all of the missing purchase quantity values using just the average purchase quantity in the entire dataset. This worked out for us, but it totally eliminates any information about the books themselves. It might be better to compute the average purchase quantity for each book instead, and then, impute these values for the books instead.
- Is there any relationship between month and the amount of books that were sold? We focused our attention on a new program, but we can also perform a similar analysis based on the month and try to look for any trends that are associated with a smaller unit of time.