Creating An Efficient Data Analysis Workflow : Book Sales Review

Eda AYDIN

28 03 2021

Contents

Upload necesessary libraries

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## Warning: package ’tibble’ was built under R version 4.0.4
```

```
## Warning: package ’tidyr’ was built under R version 4.0.4
```

```
## Warning: package ’dplyr’ was built under R version 4.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.0.4
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

Data Preparation

```r
sales <- read_csv("sales2019.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   date = col_character(),
##   user_submitted_review = col_character(),
##   title = col_character(),
##   total_purchased = col_double(),
##   customer_type = col_character()
## )
```

Data Exploration

```r
# How big is the dataset?
dim(sales)
```

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

```r
# What are the column names? What do they seem to represent?
colnames(sales)
```

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

The "date" column shows the data that the order of books was made.

```r
# What are the types of each of the columns?
for (col in colnames(sales)) {
  paste0(col, ":", typeof(sales[[col]])) %>% print
}
```

```
## [1] "date:character"
## [1] "user_submitted_review:character"
## [1] "title:character"
## [1] "total_purchased:double"
## [1] "customer_type:character"
```

```r
# Do any of the columns have missing data?
for (col in colnames(sales)) {
  paste0(col, ", numbers of missing dara rows:",
          is.na(sales[[col]]) %>% sum) %>% print
}
```

```
## [1] "date, numbers of missing dara rows:0"
## [1] "user_submitted_review, numbers of missing dara rows:885"
## [1] "title, numbers of missing dara rows:0"
## [1] "total_purchased, numbers of missing dara rows:718"
## [1] "customer_type, numbers of missing dara rows:0"
```

The `user_submitted_review` column has some missing data in it.

Handling Missing Data

```r
# Remove the rows with no user_submitted_review
complete_sales <- sales %>%
  filter(
    !is.na(user_submitted_review)
  )
complete_sales
```

```
## # A tibble: 4,115 x 5
##    date    user_submitted_review   title           total_purchased customer_type
##    <chr>   <chr>                   <chr>                     <dbl> <chr>
##  1 5/22/19 it was okay             Secrets Of R F~               7 Business
##  2 11/16/~ Awesome!                R For Dummies                 3 Business
##  3 6/27/19 Awesome!                R For Dummies                 1 Individual
##  4 11/6/19 Awesome!                Fundamentals o~               3 Individual
##  5 7/18/19 Hated it                Fundamentals o~              NA Business
##  6 1/28/19 Never read a better bo~ Secrets Of R F~               1 Business
##  7 2/20/19 Hated it                R For Dummies                 5 Business
##  8 12/17/~ Awesome!                R For Dummies                NA Business
##  9 7/13/19 OK                      R vs Python: A~               7 Business
## 10 6/22/19 The author's other boo~ R For Dummies                 1 Business
## # ... with 4,105 more rows
```

```r
# Calculate the mean of the total_purchased column, without the missing values

purchase_mean <- complete_sales %>%
  filter (!is.na(total_purchased)) %>%
  pull (total_purchased) %>%
  mean
purchase_mean
```

```
## [1] 3.985561
```

```r
# Assign this mean to all of the rows where total_purchased was NA
complete_sales <- complete_sales %>%
  mutate(
    imputed_purchased = if_else(is.na(total_purchased),
                                purchase_mean,
                                total_purchased)
  )
complete_sales
```

```
## # A tibble: 4,115 x 6
##    date   user_submitted_~ title  total_purchased customer_type imputed_purchas~
##    <chr>  <chr>            <chr>            <dbl> <chr>                    <dbl>
##  1 5/22/~ it was okay      Secre~               7 Business                  7
##  2 11/16~ Awesome!         R For~               3 Business                  3
##  3 6/27/~ Awesome!         R For~               1 Individual                1
##  4 11/6/~ Awesome!         Funda~               3 Individual                3
##  5 7/18/~ Hated it         Funda~              NA Business                  3.99
##  6 1/28/~ Never read a be~ Secre~               1 Business                  1
##  7 2/20/~ Hated it         R For~               5 Business                  5
##  8 12/17~ Awesome!         R For~              NA Business                  3.99
##  9 7/13/~ OK               R vs ~               7 Business                  7
## 10 6/22/~ The author’s ot~ R For~               1 Business                  1
## # ... with 4,105 more rows
```

Processing Review Data

```r
# Examine the unique sentences that are present in user_submitted_review

complete_sales %>% pull(user_submitted_review) %>% unique
```

```
## [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] "Would not recommend"
## [9] "I learned a lot"
```

```
is_positive <- function(review) {
  review_positive = case_when(
    str_detect(review, "Awesome!") ~TRUE,
    str_detect(review, "Ok") ~ TRUE,
    str_detect(review, "a lot") ~TRUE,
    str_detect(review, "okay") ~ TRUE,
    str_detect(review, "Never") ~ TRUE,
    TRUE ~ FALSE # The review did not contain any of the above phrases
  )
}
```

```
complete_sales <- complete_sales %>%
  mutate(
    is_positive = unlist(map(user_submitted_review, is_positive))
  )
```

Comparing Book Sales Between Pre- and Post- Program Sales

```
complete_sales <- complete_sales %>%
  mutate (
    date_status = if_else(mdy(date) < ymd("2019/07/01"), "Pre", "Post")
  )
```

```
complete_sales %>%
  group_by(date_status,title) %>%
  summarize(
    books_purchased = sum(imputed_purchased)
  ) %>%
  arrange(title, date_status)
```

```
## `summarise()` has grouped output by 'date_status'. You can override using the `.groups` argument.
```

```
## # A tibble: 12 x 3
## # Groups:   date_status [2]
##    date_status title                              books_purchased
##    <chr>       <chr>                                        <dbl>
##  1 Post        Fundamentals of R For Beginners              2832.
##  2 Pre         Fundamentals of R For Beginners              3093.
##  3 Post        R For Dummies                                2779.
##  4 Pre         R For Dummies                                2626.
##  5 Post        R Made Easy                                    24
##  6 Pre         R Made Easy                                    15
##  7 Post        R vs Python: An Essay                        1172.
##  8 Pre         R vs Python: An Essay                        1271.
##  9 Post        Secrets Of R For Advanced Students           1154.
## 10 Pre         Secrets Of R For Advanced Students            965.
## 11 Post        Top 10 Mistakes R Beginners Make              228.
## 12 Pre         Top 10 Mistakes R Beginners Make              241.
```

Comparing Book Sales Within Customer Type

```
complete_sales %>%
  group_by(date_status, customer_type) %>%
  summarize (
    books_purchased = sum(imputed_purchased)
  ) %>%
  arrange(customer_type, date_status)
```

```
## ‘summarise()‘ has grouped output by ’date_status’. You can override using the ‘.groups‘ argument.
```

```
## # A tibble: 4 x 3
## # Groups:   date_status [2]
##   date_status customer_type books_purchased
##   <chr>       <chr>                   <dbl>
## 1 Post        Business                5742.
## 2 Pre         Business                5612.
## 3 Post        Individual              2448.
## 4 Pre         Individual              2599.
```

Comparing Review Sentiment Between Pre- and Post-Program Sales

```
# Create another summary table that compares the number of positive
# reviews before and after July 1, 2019
complete_sales %>%
  group_by(date_status) %>%
  summarize(
    num_positive_reviews = sum(is_positive)
  )
```

```
## # A tibble: 2 x 2
##   date_status num_positive_reviews
##   <chr>                      <int>
## 1 Post                         909
## 2 Pre                          892
```