Guided Project: Creating An Efficient Data Analysis Workflow, Part 2

Husen Wahyu

4/7/2021

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

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

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

```
sales2019 <- 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()
## )
```

Page 2, Review the Dataset

```r
# 1. How big is the dataset? What are the column names, and what do they seem to represent?
dim(sales2019)
```

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

```r
colnames(sales2019)
```

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

```r
# 2. What are the types of each of the columns?
for (col in colnames(sales2019)) {
  paste0(col, " : ", typeof(sales2019[[col]])) %>% print
}
```

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

```r
# 3. That being said, do any of the columns have missing data? If so, make a note of this.
for (col in colnames(sales2019)) {
  c(col, is.na(sales2019[[col]]) %>% sum) %>% print()
}
```

```
## [1] "date" "0"
## [1] "user_submitted_review" "885"
## [1] "title" "0"
## [1] "total_purchased" "718"
## [1] "customer_type" "0"
```

Page 3, Handling with Missing Values

```r
# 1. Remove all rows in the dataset that have an NA value for the user_submitted_review column.
sales2019 <- sales2019 %>%
  filter(is.na(user_submitted_review) != TRUE)
dim(sales2019)
```

```
## [1] 4115    5
```

```r
# 885 rows has been deleted

# 2.Using the remaining rows that have data, calculate the average number of books purchased on an orde
avg_total_purchased <- sales2019 %>%
  filter(!is.na(total_purchased)) %>%
    pull(total_purchased) %>%
```

```
      mean

sales2019 <- sales2019 %>%
  mutate(
    total_purchased_notnull = if_else(is.na(total_purchased),
                                      avg_total_purchased,
                                      total_purchased)
  )
```

Page 4, String Manipulation, Review Comprehension

```
unique(sales2019$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] "Would not recommend"
## [9] "I learned a lot"
```

```
#positive <- c("okay", "Awesome", "OK", "learn", "Never read a better book")
#negative <- c("Hate", "other", "not")

is_positive_rev <- function(review) {
  case_when(
    str_detect(review, "okay") ~ TRUE,
    str_detect(review, "Awesome") ~ TRUE,
    str_detect(review, "OK") ~ TRUE,
    str_detect(review, "learn") ~ TRUE,
    str_detect(review, "Never read a better book") ~ TRUE,
    TRUE ~ FALSE
  )
}

sales2019 <- sales2019 %>%
  mutate(
    positive_review =
      is_positive_rev(user_submitted_review)
  )
```

Page 5, performing date conversion and grouping the dataset

```
#creating new column of date_unix
sales2019 <- sales2019 %>%
```

```
  mutate(
    date_unix = unlist(map(date, mdy))
  )

#creating grouping column
sales2019 <- sales2019 %>%
  mutate(
    category_program = if_else(date_unix<mdy("07-01-2019"),"before", "after")
  )

#grouping and summarizing
program_summary <- sales2019 %>%
  group_by(category_program) %>%
    summarise(
      total_purchases = sum(total_purchased_notnull)
    )
program_summary
```

```
## # A tibble: 2 x 2
##   category_program total_purchases
##   <chr>                      <dbl>
## 1 after                      8190.
## 2 before                     8211.
```

Page 6, grouping and summarizing in subgroup

```
program_summary_subgroup <- sales2019 %>%
  group_by(customer_type, category_program) %>%
    summarise(
      total_purchases = sum(total_purchased_notnull)
    )
```

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

```
program_summary_subgroup
```

```
## # A tibble: 4 x 3
## # Groups:   customer_type [2]
##   customer_type category_program total_purchases
##   <chr>         <chr>                      <dbl>
## 1 Business      after                      5742.
## 2 Business      before                     5612.
## 3 Individual    after                      2448.
## 4 Individual    before                     2599.
```

Page 7, Grouping and summarizing based on review

```r
review_program_summary <- sales2019 %>%
  group_by(category_program) %>%
    summarise(
      percentage_of_positive_review = sum(positive_review) / (nrow(sales2019)-sum(positive_review))
    )
review_program_summary
```

```
## # A tibble: 2 x 2
##   category_program percentage_of_positive_review
##   <chr>                                   <dbl>
## 1 after                                   0.378
## 2 before                                  0.380
```

Program is not effective