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

Vamos analisar uma base de dados de venda de livros de programação que obtiveram reviews com objetivo de extrair insights desses dados. A ideia é demonstrar de forma simples e objetiva o passo a passo do processo de análise de dados, desde a coleta, limpeza, transformação até o resultado obtido após análises. No entanto, nessa parte 2 com algumas técnicas mais avançadas.

A base de dados pode ser obtida pela plataforma data.world (clique aqui).

Bibliotecas Utilizadas

```
library(readr)
library(dplyr)
library(purrr)
library(lubridate)
library(janitor)
library(glue)
library(stringr)
```

Coleta da Base

Com o código abaixo constatamos que a base possui 5 mil registros e 5 colunas, sendo uma delas do tipo double e as demais character. É possível já de cara notar que temos valores nulos e também uma coluna de data para ser manipulada.

```
base <- read_csv("https://query.data.world/s/7f4ah2qv53p4if4dass7cj4qyhkr3x")
```

```
## `curl` package not installed, falling back to using `url()`
## Rows: 5000 Columns: 5
## -- Column specification --------------------------------------------------------
```

```
## Delimiter: ","
## chr (4): date, user_submitted_review, title, customer_type
## dbl (1): total_purchased
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
glimpse(base)
```

```
## 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        <dbl> 7, 3, 1, 3, NA, 1, 5, NA, 7, 1, 7, NA, 3, 2, 0, ~
## $ customer_type          <chr> "Business", "Business", "Individual", "Individua~
```

Campo	Significado
date	Data da compra
user_submitted_review	Avaliação do comprador
title	Título do Livro
total_purchased	Quantidade de livros comprados
customer_type	Tipo de consumidor

Investigando dados

Entendendo valores únicos presentes na base apenas nas variáveis categóricas.

Para evitar o uso repetitivo da função table, imaginando situações onde o dataset tem muitas colunas, podemos utilizar a função map, assim num único comando replicamos o efeito para várias colunas além de ser uma função vetorizada e portanto mais eficiente.

Com o adorn_totals é possível calcular o total da tabela de maneira facilitada.

Com esse código é possível perceber que:

- os reviews tem uma distribuição bem parecida
- já sobre os livros, o R Made Easy foi um dos menos procurados, enquando os mais procurados foi Fundamentals of R for Beginners e R for Dummies
- boa parte das compras foram empresas, ou com foco em negócios

```r
tabelas <- map(base %>% select(user_submitted_review,
                    title,
                    customer_type),
    table,
    useNA = "ifany") # esse parâmetro a função map vai repassar a função table
                     # assim é possível calcular a qtde de nulos quando existir


indices <- 1:length(tabelas)

for(i in indices){
  result <- tabelas[[i]] %>%
                as.data.frame() %>%
                adorn_totals() # calcula total da tabela

  # trocando nome da coluna pelo nome da variável em análise
  names(result)[1] <- names(tabelas)[[i]]

  result %>%
    print()
}
```

```
##                 user_submitted_review Freq
##      A lot of material was not needed  428
##                                Awesome!  452
##                                Hated it  474
##                          I learned a lot  441
##                             it was okay  459
##               Never read a better book  449
##                                      OK  461
##  The author's other books were better  465
##                    Would not recommend  486
##                                    <NA>  885
##                                   Total 5000
##                                title Freq
##     Fundamentals of R For Beginners 1809
##                           R For Dummies 1630
##                             R Made Easy   12
##                 R vs Python: An Essay  771
##  Secrets Of R For Advanced Students  632
##    Top 10 Mistakes R Beginners Make  146
##                                Total 5000
##  customer_type Freq
```

```
##       Business 3445
##     Individual 1555
##          Total 5000
```

Para o campo de data, podemos fazer uso da biblioteca lubridate para facilitar nossa análise. Com ela foi possível facilmente conveter o formado mm/dd/yyyy para yyyy-mm-dd. Com o uso do substr, selecionamos apenas o ano e mês para agrupar e ter uma ideia da distribuição da base. Vemos que a base possui registros no período de 2019, de janeiro a dezembro.

```
base <- base %>% mutate(date = mdy(date))

base %>%
  mutate(safra = substr(date,1,7)) %>%
  group_by(safra) %>%
  summarise(Qtd = n()) %>%
  mutate(Perc = Qtd/sum(Qtd)) %>%
  adorn_totals()
```

```
##    safra  Qtd   Perc
##  2019-01  417 0.0834
##  2019-02  388 0.0776
##  2019-03  436 0.0872
##  2019-04  408 0.0816
##  2019-05  421 0.0842
##  2019-06  415 0.0830
##  2019-07  442 0.0884
##  2019-08  413 0.0826
##  2019-09  387 0.0774
##  2019-10  425 0.0850
##  2019-11  396 0.0792
##  2019-12  452 0.0904
##    Total 5000 1.0000
```

Uma alternativa para a coluna numérica (pensando que em geral a variação de valores pode ser bem maior que dados categóricos) é encontrar os pontos de cortes que dividem a base em partes iguais. Por exemplo aqui escolhemos dividir a base em 5 partes, cada parte vai acumular 20% dos registros. Sendo assim o resultado armazenado na variável quantil é o valor que se dividirmos a base vai permitir essa separação em 20%.

A função cut vai utilizar esses pontos de corte para de fato cortar a informação e criar intervalos, assim essa nova informação pode ser agrupada como uma variável categórica, visto que a informação foi reduzida de um valor bruto a intervalos.

```
quantil <- quantile(base %>% select(total_purchased) %>%
                            filter(!is.na(total_purchased)) %>%
                            pull(), probs = seq(0,1,0.2))
quantil
```

```
##   0%  20%  40%  60%  80% 100%
##    0    2    3    4    6   12
```

```
base %>%
  mutate(total_purchased_cut = cut(total_purchased,
                                   quantil)) %>%
  group_by(total_purchased_cut) %>%
  summarise(Qtd = n()) %>%
  mutate(Perc = Qtd/sum(Qtd)) %>%
  adorn_totals()
```

```
##  total_purchased_cut  Qtd   Perc
##                (0,2]  955 0.1910
##                (2,3]  820 0.1640
##                (3,4]  835 0.1670
##                (4,6] 1110 0.2220
##               (6,12]  491 0.0982
##                 <NA>  789 0.1578
##                Total 5000 1.0000
```

Limpeza de nulos

E para as avaliações temos a presença de dados nulos que precisamos decidir entre remover os dados ou imputar utilizando algum método estatísco como por exemplo a média.

```
for(column in colnames(base)){
  is_na <- base[[column]] %>% is.na() %>%
                              sum()
  if(is_na > 0){
    glue("base${column}: {is_na} nulos") %>% print()
  }
}
```

```
## base$user_submitted_review: 885 nulos
## base$total_purchased: 718 nulos
```

Com o resultado acima notamos que temos nulos nos reviews e no total de livros comprados.

Para a coluna de review com nulos, a abordagem de remoção aqui é o ideal visto que o review é um dos principais pontos para estudo se a campanha de venda foi efetiva.

```
base_clean <- base %>% filter(!is.na(user_submitted_review))

glimpse(base_clean)
```

```
## Rows: 4,115
## Columns: 5
## $ date                  <date> 2019-05-22, 2019-11-16, 2019-06-27, 2019-11-06,~
## $ user_submitted_review <chr> "it was okay", "Awesome!", "Awesome!", "Awesome!~
## $ title                 <chr> "Secrets Of R For Advanced Students", "R For Dum~
## $ total_purchased       <dbl> 7, 3, 1, 3, NA, 1, 5, NA, 7, 1, 7, NA, 3, 2, 6, ~
## $ customer_type         <chr> "Business", "Business", "Individual", "Individua~
```

Já para o total de vendas, uma melhor abordagem é utilizar a imputação de um valor estatístico, nesse caso a média, pois não vai distorcer tanto o resultado e não vamos desperdiçar os reviews.

```
#selecionando um exemplo para conferir o resultado
base_clean %>% filter(user_submitted_review == "Hated it",
                      title == "R For Dummies",
                      date == "2019-05-09") %>% head()
```

```
## # A tibble: 2 x 5
##   date       user_submitted_review title         total_purchased customer_type
##   <date>     <chr>                 <chr>                   <dbl> <chr>
## 1 2019-05-09 Hated it              R For Dummies              NA Business
## 2 2019-05-09 Hated it              R For Dummies               1 Business
```

```
#imputando media
base_clean <- base_clean %>%
  mutate(total_purchased = ifelse(is.na(total_purchased),
                                  mean(total_purchased, na.rm = TRUE) %>%
                                    round(),
                                  total_purchased))

#resultado
```

```
base_clean %>% filter(user_submitted_review == "Hated it",
                      title == "R For Dummies",
                      date == "2019-05-09") %>% head()
```

```
## # A tibble: 2 x 5
##   date       user_submitted_review title         total_purchased customer_type
##   <date>     <chr>                 <chr>                   <dbl> <chr>
## 1 2019-05-09 Hated it              R For Dummies               4 Business
## 2 2019-05-09 Hated it              R For Dummies               1 Business
```

Padronizando informações

Agora vamos criar um novo campo que tenha de forma simplificada se o review foi positivo ou negativo. Assim conseguiremos avaliar se houve evolução na campanha.

```
base_clean <- base_clean %>%
  mutate(
    is_positive =  case_when(str_detect(user_submitted_review, "lot")     ~ TRUE,
                             str_detect(user_submitted_review, "Awesome") ~ TRUE,
                             str_detect(user_submitted_review, "okay")    ~ TRUE,
                             str_detect(user_submitted_review, "Never")   ~ TRUE,
                             str_detect(user_submitted_review, "OK")      ~ TRUE,
                             TRUE ~ FALSE)

    )
base_clean %>% select(user_submitted_review,is_positive) %>% head()
```

```
## # A tibble: 6 x 2
##   user_submitted_review    is_positive
##   <chr>                    <lgl>
## 1 it was okay              TRUE
## 2 Awesome!                 TRUE
## 3 Awesome!                 TRUE
## 4 Awesome!                 TRUE
## 5 Hated it                 FALSE
## 6 Never read a better book TRUE
```

Após criado as colunas de código e descrição separadas, é interessante cruzar com a coluna original para conferir se o resultado está correto.

```
base_clean %>% select(user_submitted_review,is_positive) %>% table()
```

```
##                                       is_positive
## user_submitted_review                  FALSE TRUE
##   A lot of material was not needed         0  428
##   Awesome!                                 0  452
##   Hated it                               474    0
##   I learned a lot                          0  441
##   it was okay                              0  459
##   Never read a better book                 0  449
##   OK                                       0  461
##   The author's other books were better   465    0
##   Would not recommend                    486    0
```

Outro campo que podemos criar é uma identificação se o período da compra foi antes ou depois da campanha de vendas ter sido criada.

```
base_clean <- base_clean %>%
  mutate(
    sale_campaign_period =  ifelse(ymd(date) < "2019-07-01", "Antes", "Depois")
    )

base_clean %>% select(date, sale_campaign_period) %>% head()
```

```
## # A tibble: 6 x 2
##   date       sale_campaign_period
##   <date>     <chr>
## 1 2019-05-22 Antes
## 2 2019-11-16 Depois
## 3 2019-06-27 Antes
## 4 2019-11-06 Depois
## 5 2019-07-18 Depois
## 6 2019-01-28 Antes
```

Análise da base

Agora podemos agrupar as novas colunas e entender se a campanha de venda causou efeito nas compras e reviews. Pelo resultado conseguimos ver que não houve mudanças significativas tanto no volume quanto na percepção positiva dos clientes, apenas um pequeno aumento em vendas com reviews positivos. Então com essa

análise uma conclusão que podemos tirar é que a campanha não foi o suficiente para melhorar as vendas.

```
base_clean %>%
  group_by(sale_campaign_period) %>%
  summarise(Qtd = n(),
            Venda_Positiva = sum(is_positive),
            Taxa_aceitacao = Venda_Positiva/Qtd)
```

```
## # A tibble: 2 x 4
##   sale_campaign_period   Qtd Venda_Positiva Taxa_aceitacao
##   <chr>                <int>          <int>          <dbl>
## 1 Antes                 2050           1328          0.648
## 2 Depois                2065           1362          0.660
```

Por outro lado olhamos apenas o resultado geral das vendas e temos títulos diferentes misturados nessa análise. Vamos tentar olhar mais a fundo de outros pontos de vista.

Olhando pelo tipo de consumidor o aumento ainda não é muito significativo

```
base_clean %>%
  group_by(customer_type,sale_campaign_period) %>%
  summarise(Qtd = n(),
            Venda_Positiva = sum(is_positive),
            Taxa_aceitacao = Venda_Positiva/Qtd)
```

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

```
## # A tibble: 4 x 5
## # Groups:   customer_type [2]
##   customer_type sale_campaign_period   Qtd Venda_Positiva Taxa_aceitacao
##   <chr>         <chr>                <int>          <int>          <dbl>
## 1 Business      Antes                 1405            913          0.650
## 2 Business      Depois                1451            966          0.666
## 3 Individual    Antes                  645            415          0.643
## 4 Individual    Depois                 614            396          0.645
```

Por livro, há um aumento de vendas em alguns casos, melhores reviews, mas novamente não é um número significativo. O livro R Made Easy acaba tendo uma distorção nos percentuais devido ao volume muito baixo de livros vendidos.

```
base_clean %>%
  group_by(title,sale_campaign_period) %>%
  summarise(Qtd = n(),
            Venda_Positiva = sum(is_positive),
            Taxa_aceitacao = Venda_Positiva/Qtd)
```

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

```
## # A tibble: 12 x 5
## # Groups:   title [6]
##    title                    sale_campaign_p~   Qtd Venda_Positiva Taxa_aceitacao
##    <chr>                    <chr>            <int>          <int>          <dbl>
##  1 Fundamentals of R For B~ Antes              770            486          0.631
##  2 Fundamentals of R For B~ Depois             725            480          0.662
##  3 R For Dummies            Antes              659            427          0.648
##  4 R For Dummies            Depois             689            461          0.669
##  5 R Made Easy              Antes                4              2          0.5
##  6 R Made Easy              Depois               5              5          1
##  7 R vs Python: An Essay    Antes              304            211          0.694
##  8 R vs Python: An Essay    Depois             308            201          0.653
##  9 Secrets Of R For Advanc~ Antes              252            167          0.663
## 10 Secrets Of R For Advanc~ Depois             280            173          0.618
## 11 Top 10 Mistakes R Begin~ Antes               61             35          0.574
## 12 Top 10 Mistakes R Begin~ Depois              58             42          0.724
```

Uma última visualização é ver se houve alguma evolução ao longo do tempo que com as visualizções anteriores não foi possivel enxergar. Mas pelo resultado, não parece ter uma manifestação maior de vendas ou de reviews positivos evoluindo de acordo com os meses. Na verdade no mês anterior ao lançamento da campanha foi o mês com mais reviews positivos.

```
base_clean %>%
  mutate(safra = substr(date,1,7)) %>%
  group_by(safra) %>%
  summarise(Qtd = n(),
            Venda_Positiva = sum(is_positive),
            Taxa_aceitacao = Venda_Positiva/Qtd)
```

```
## # A tibble: 12 x 4
##    safra     Qtd Venda_Positiva Taxa_aceitacao
```

```
##    <chr>   <int>          <int>          <dbl>
##  1 2019-01   348            233          0.670
##  2 2019-02   312            201          0.644
##  3 2019-03   366            233          0.637
##  4 2019-04   341            215          0.630
##  5 2019-05   340            209          0.615
##  6 2019-06   343            237          0.691
##  7 2019-07   365            243          0.666
##  8 2019-08   339            217          0.640
##  9 2019-09   314            213          0.678
## 10 2019-10   333            210          0.631
## 11 2019-11   325            215          0.662
## 12 2019-12   389            264          0.679
```

Conclusão

Aqui as análises mostram que a campanha não demonstrou ter influenciado nas vendas. Claro que é um exemplo simples, na vida real, vários motivos podem influenciar, como sazonalidade nas vendas, falta de informações para chegar a uma conclusão, dados incorretos e etc.