Guided Project: Creating An Efficient Data Analysis Workflow

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.

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

Bibliotecas Utilizadas

```
library(readr)
library(dplyr)
```

Coleta da Base

Com o código abaixo constatamos que a base possui 2 mil registros e 4 colunas, sendo uma delas do tipo double e as demais character. É possível já de cara notar que temos valores nulos, e informações iguais descritas de maneiras distintas.

```
base <- read_csv("https://query.data.world/s/wmu3zbxkfwmq7wejwmwyj4qacawznu")
```

```
## Rows: 2000 Columns: 4
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): book, review, state
## dbl (1): price
```

```
##
## 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: 2,000
## Columns: 4
## $ book   <chr> "R Made Easy", "R For Dummies", "R Made Easy", "R Made Easy", "~
## $ review <chr> "Excellent", "Fair", "Excellent", "Poor", "Great", NA, "Great",~
## $ state  <chr> "TX", "NY", "NY", "FL", "Texas", "California", "Florida", "CA",~
## $ price  <dbl> 19.99, 15.99, 19.99, 19.99, 50.00, 19.99, 19.99, 19.99, 29.99, ~
```

Campo	Significado
book	Título do Livro
review	Avaliação do comprador
state	Estado onde o comprador mora
price	Preço que o comprador pagou pelo livro

Acima fizemos o uso de bibliotecas para facilitar a investigação da base. Mas existe uma forma alternativa de ver as mesmas informações.

Tamanho da base:

```
dim(base)
```

```
## [1] 2000    4
```

Colunas e seu tipo de dado

```
for (column in colnames(base)){
  print( paste(column, class(base[[column]]) ) )
}
```

```
## [1] "book character"
## [1] "review character"
## [1] "state character"
## [1] "price numeric"
```

ou

```
for (column in colnames(base)){
  print( paste(column, typeof(base[[column]]) ) )
}
```

```
## [1] "book character"
## [1] "review character"
## [1] "state character"
## [1] "price double"
```

Investigando dados

Entendendo valores únicos presentes na base

```
base$book %>% unique()
```

```
## [1] "R Made Easy"                          "R For Dummies"
## [3] "Secrets Of R For Advanced Students" "Top 10 Mistakes R Beginners Make"
## [5] "Fundamentals of R For Beginners"
```

```
base$review %>% unique()
```

```
## [1] "Excellent" "Fair"      "Poor"      "Great"     NA          "Good"
```

```
base$state %>% unique()
```

```
## [1] "TX"         "NY"         "FL"         "Texas"      "California"
## [6] "Florida"    "CA"         "New York"
```

```
base$price %>% unique()
```

```
## [1] 19.99 15.99 50.00 29.99 39.99
```

Uma alternativa para a coluna numérica (pensando que em geral a variação de valores pode ser bem maior que dados categóricos)

```
base$price %>% min()
```

```
## [1] 15.99
```

```
base$price %>% max()
```

```
## [1] 50
```

```
base$price %>% mean()
```

```
## [1] 31.28703
```

```
base$price %>% summary()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   15.99   19.99   29.99   31.29   39.99   50.00
```

Também é interessante entender a distribuição da base, quantos dos livros possuem qual valor em termos absolutos e relativos.

Com o resultado abaixo podemos ver que os valores são bem distribuídos, para os 5 preços existentes na base, cada um deles possuem cerca de 20% de livros, em outras palavras não existe uma grande concentração em algum valor específico.

```
base$price %>% table()
```

```
## .
## 15.99 19.99 29.99 39.99    50
##   410   389   385   410   406
```

```
base$price %>% table() %>% prop.table()
```

```
## .
##  15.99  19.99  29.99  39.99     50
## 0.2050 0.1945 0.1925 0.2050 0.2030
```

Os títulos presentes na base também têm uma distribuição uniforme.

```
base$book %>% table() %>% prop.table()
```

```
## .
##    Fundamentals of R For Beginners                     R For Dummies
##                              0.2050                             0.2050
##                        R Made Easy Secrets Of R For Advanced Students
##                              0.1945                             0.2030
##   Top 10 Mistakes R Beginners Make
##                              0.1925
```

Já para o estado não conseguimos analisar bem o resultado visto que temos valores distintos, mas que representam a mesma informação.

```
base$state %>% table() %>% prop.table()
```

```
## .
##         CA California         FL    Florida   New York         NY      Texas
##     0.1310     0.1280     0.1240     0.1005     0.1360     0.1295     0.1355
##         TX
##     0.1155
```

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.

```
base$review %>% is.na() %>% sum()
```

```
## [1] 206
```

Limpeza dos dados

Resolvendo nulos: Eliminação ou Imputação

Vamos começar avaliando quais colunas possuem dados nulos e qual o volume.

```
for(column in colnames(base)){
  nulos <- base[[column]] %>% is.na() %>% sum()
  print(paste(column,", nulos: ", nulos))

}
```

```
## [1] "book , nulos:  0"
## [1] "review , nulos:  206"
## [1] "state , nulos:  0"
## [1] "price , nulos:  0"
```

Com o resultado acima notamos que apenas a coluna de reviews possui dados nulos e o volume é de 206 nulos em torno de 10% da base. Sendo assim vamos optar pelo método de remoção dos nulos, ou melhor vamos criar um novo objeto sem nulos.

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

glimpse(base_clean)
```

```
## Rows: 1,794
## Columns: 4
## $ book   <chr> "R Made Easy", "R For Dummies", "R Made Easy", "R Made Easy", "~
## $ review <chr> "Excellent", "Fair", "Excellent", "Poor", "Great", "Great", "Po~
## $ state  <chr> "TX", "NY", "NY", "FL", "Texas", "Florida", "CA", "CA", "Texas"~
## $ price  <dbl> 19.99, 15.99, 19.99, 19.99, 50.00, 19.99, 19.99, 29.99, 50.00, ~
```

Padronizando informações

Agora vamos criar novos campos que tenha o estado do comprador de forma padronizada.

```
base_clean <- base_clean %>%
   mutate(state_code = case_when(state == "California" ~ "CA",
                                 state == "Florida"    ~ "FL",
                                 state == "New York"   ~ "NY",
                                 state == "Texas"      ~ "TX",
                                 TRUE                  ~ state),
          state_desc = case_when(state == "CA" ~ "California",
                                 state == "FL" ~ "Florida",
                                 state == "NY" ~ "New York",
                                 state == "TX" ~ "Texas",
                                 TRUE          ~ state))
```

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(state,state_code) %>% table()
```

```
##             state_code
## state         CA  FL  NY  TX
##   CA         234   0   0   0
##   California 230   0   0   0
##   FL           0 225   0   0
##   Florida      0 181   0   0
##   New York     0   0 250   0
##   NY           0   0 234   0
##   Texas        0   0   0 238
##   TX           0   0   0 202
```

```
base_clean %>% select(state,state_desc) %>% table()
```

```
##             state_desc
## state        California Florida New York Texas
##   CA                234       0        0     0
##   California        230       0        0     0
##   FL                  0     225        0     0
##   Florida             0     181        0     0
##   New York            0       0      250     0
##   NY                  0       0      234     0
##   Texas               0       0        0   238
##   TX                  0       0        0   202
```

Por fim podemos novamente olhar para a distribuição dos estados e notamos que a maior compra de livros ocorreu em New York enquanto que a Florida teve o menor índice.

```
base_clean$state_code %>% table() %>% prop.table()
```

```
## .
##        CA        FL        NY        TX
## 0.2586399 0.2263099 0.2697882 0.2452620
```

Convertendo dados

O review dos usuários está de forma descritiva e pode causar dúvidas e ser mais difícil de trabalhar, então vamos converter em uma nota numérica. Também vamos criar uma flag para determinar se o título está dentre aqueles com melhor avaliação.

```
base_clean <- base_clean %>%
  mutate(review_num = case_when(review == "Poor"      ~ 1,
                                review == "Fair"      ~ 2,
                                review == "Good"      ~ 3,
                                review == "Great"     ~ 4,
                                review == "Excellent" ~ 5))
base_clean <- base_clean %>%
  mutate(is_high_review = ifelse(review_num >= 4, TRUE, FALSE))

base_clean %>% select(review, review_num, is_high_review) %>% head()
```

```
## # A tibble: 6 x 3
##   review    review_num is_high_review
##   <chr>          <dbl> <lgl>
## 1 Excellent          5 TRUE
## 2 Fair               2 FALSE
## 3 Excellent          5 TRUE
## 4 Poor               1 FALSE
## 5 Great              4 TRUE
## 6 Great              4 TRUE
```

Analisando dados tranformados

O objetivo ao analisar os dados, é conseguir de alguma forma uma melhoria, ideias, informações que permitirão obter resultados positivos. Como estamos falando de review de livros, é interessante justamente saber quais são os livros melhor avaliados, quais são os mais lucráveis?

O exercício aqui nos permitiu escolher uma métrica, num projeto real pode ser que essa métrica seja determinada pelo cliente, ou é possível que seja necessário nós mesmos determinar.

Aqui escolhemos mais lucrável os livros que tem **melhor avaliação, menor custo e maior número de vendas**.

Foi decidido dessa forma, pois um livro que vende fácil, custa pouco e é bem avaliado é uma boa aposta para melhorar o lucro sem muito esforço.

Abaixo então:

- agrupamos o resultado por título,
- calculamos a quantidade de vendas de cada título,
- calculamos o percentual vendas com review com nota alta,
- calculamos o preço médio de cada título,
- calculamos o custo total das vendas de cada título,
- ordenamos o custo total de forma ascendente.

E descobrimos que:

- o total de vendas foi parecido em todos títulos, em torno de 360 vendas,
- praticamente todos os livros apresentaram em torno de 40% de ótimas avaliações,
- no entanto o custo total varia bastante devido ao preço dos títulos.

Por fim tanto o título *R For Dummies* quanto o *R Made Easy* parecem ser os mais lucrativos visto que um é o menos custoso, e o outro apesar de ser o segundo menos custoso é 4 pontos percentuais mais bem avaliado.

```
base_clean %>%
  group_by(book) %>%
  summarise(Total_Vendas      = n(),
            Perc_Bem_Avaliado = sum(is_high_review)/n(),
            Preco_Medio       = mean(price),
            Custo_Total       = mean(price)*n()) %>%
  arrange(Custo_Total)
```

```
## # A tibble: 5 x 5
##   book                     Total_Vendas Perc_Bem_Avalia~ Preco_Medio Custo_Total
##   <chr>                           <int>            <dbl>       <dbl>       <dbl>
## 1 R For Dummies                     361            0.355        16.0       5772.
## 2 R Made Easy                       352            0.395        20.0       7036.
## 3 Top 10 Mistakes R Begin~          355            0.397        30.0      10646.
## 4 Fundamentals of R For B~          366            0.413        40.0      14636.
## 5 Secrets Of R For Advanc~          360            0.375        50        18000
```