

Métodos Computacionais para Estatística e Otimização

Lista 2 - Luiz Henrique Barretta Francisco

Considere o conjunto de dados sobre voos e operações aéreas disponibilizado pela ANAC (Agência Nacional de Aviação Comercial) no seguinte endereço clique aqui. O arquivo chamado de *Dados_Estatisticos.csv* apresenta uma série de informações sobre as operadoras aéreas que atuam no Brasil. Forneça o código R usando o framework {tidyverse} para responder cada uma das seguintes questões práticas.

1. Quantas empresas nacionais e internacionais operam no mercado brasileiro?

```
df %>% distinct(EMPRESA_SIGLA, EMPRESA_NACIONALIDADE) %>%
  count(EMPRESA_NACIONALIDADE, name = "Quantidade")
```

```
##   EMPRESA_NACIONALIDADE Quantidade
## 1           BRASILEIRA        58
## 2       ESTRANGEIRA       250
```

2. Considerando apenas empresas nacionais e o ano de 2020. Qual é a região de origem com maior número de decolagens? Apresente o resultado em uma tabela ordenada do maior para o menor.

```
df %>% filter(EMPRESA_NACIONALIDADE == "BRASILEIRA", ANO == 2020, AEROPORTO_DE_ORIGEM_REGIAO != "") %>%
  group_by(AEROPORTO_DE_ORIGEM_REGIAO) %>%
  summarise(Total_Decolagens = sum(DECOLAGENS, na.rm = TRUE)) %>%
  arrange(desc(Total_Decolagens))
```

```
## # A tibble: 5 x 2
##   AEROPORTO_DE_ORIGEM_REGIAO Total_Decolagens
##   <chr>                      <int>
## 1 SUDESTE                     208794
## 2 NORDESTE                     77217
## 3 CENTRO-OESTE                  51510
## 4 SUL                          48882
## 5 NORTE                        32090
```

3. Considerando apenas empresas nacionais. Faça uma tabela comparando o número de decolagens de acordo com a região de origem para cada ano. Organize a tabela resultante no formato wide com regiões lado a lado e ANO nas linhas.

```
g1 <- df %>% filter(EMPRESA_NACIONALIDADE == "BRASILEIRA", AEROPORTO_DE_ORIGEM_REGIAO != "") %>%
  group_by(ANO, AEROPORTO_DE_ORIGEM_REGIAO) %>%
  summarise(Total_Decolagens = sum(DECOLAGENS, na.rm = TRUE)) %>%
  pivot_wider(names_from = AEROPORTO_DE_ORIGEM_REGIAO,
              values_from = Total_Decolagens, values_fill = 0) %>%
  arrange(ANO)
```

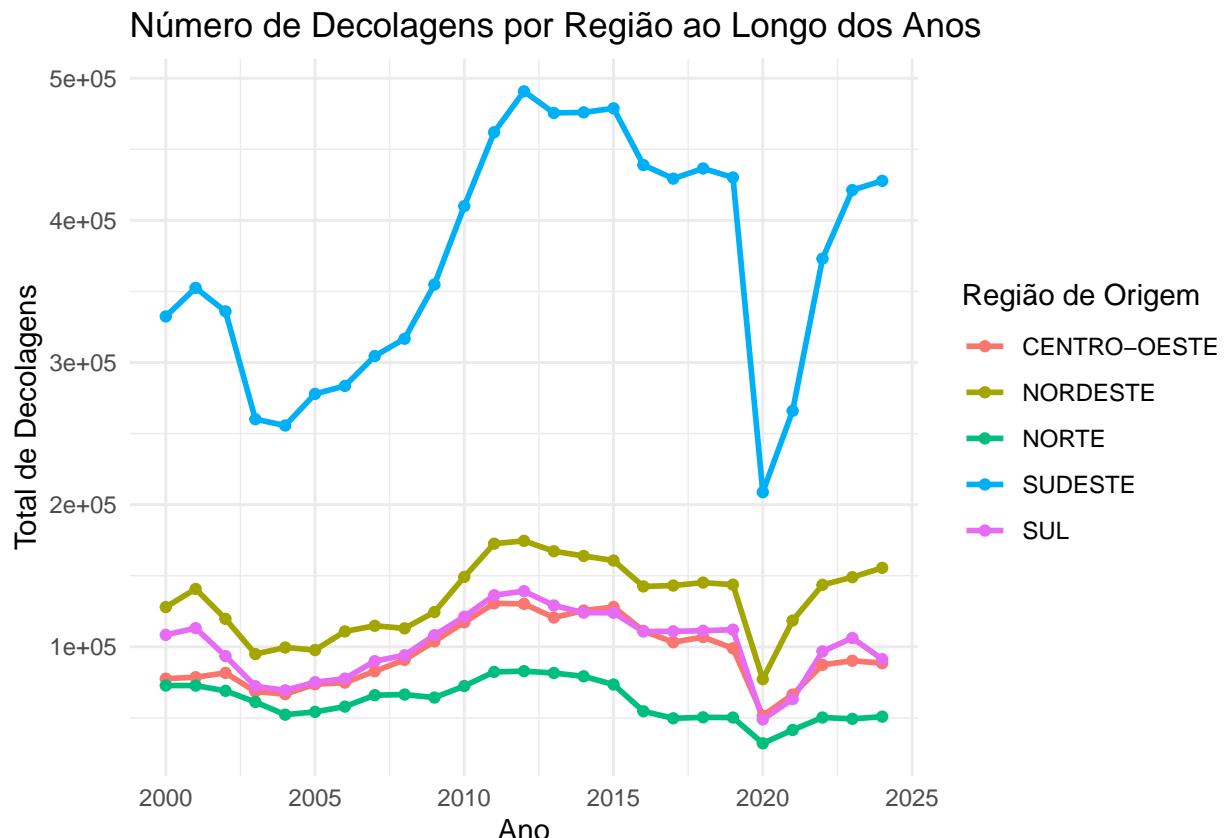
```
g1
```

```

## # A tibble: 26 x 6
## # Groups:   ANO [26]
##       ANO `CENTRO-OESTE` NORDESTE NORTE SUDESTE     SUL
##     <int>        <int>    <int> <int>    <int>    <int>
## 1   2000        77563  127952 72758  332420 108384
## 2   2001        78618  140684 72698  352550 113148
## 3   2002        81633  119631 69021  335966 93443
## 4   2003        68511  94912 61163  260122 72276
## 5   2004        66684  99503 52237  255666 69383
## 6   2005        73647  97691 54205  277931 75070
## 7   2006        74801  110823 57934  283542 77631
## 8   2007        82788  114789 65952  304544 89904
## 9   2008        90823  112979 66405  316750 94063
## 10  2009       103880  124451 64281  354896 108209
## # ... i 16 more rows

g1 %>% filter(ANO <= 2024) %>%
  pivot_longer(cols = -ANO, names_to = "Regiao", values_to = "Total_Decolagens") %>%
  ggplot(aes(x = ANO, y = Total_Decolagens, color = Regiao)) +
  geom_line(size = 1) + geom_point() +
  labs(title = "Número de Decolagens por Região ao Longo dos Anos",
       x = "Ano", y = "Total de Decolagens", color = "Região de Origem") +
  theme_minimal()

```



4. Considerando empresas nacionais e internacionais. Quais são as cinco empresas que mais voaram em termos de horas de voo no ano de 2020?

```

df$HORAS_VOADAS <- as.numeric(gsub(", ", ".", df$HORAS_VOADAS))

df %>% filter(ANO == 2020) %>%
  mutate(HORAS_VOADAS = ifelse(is.na(HORAS_VOADAS) | HORAS_VOADAS == "", 0, HORAS_VOADAS)) %>%
  group_by(EMPRESA_SIGLA) %>%
  summarise(total_horas = sum(HORAS_VOADAS, na.rm = TRUE)) %>%
  arrange(desc(total_horas)) %>%
  head(5)

```

```

## # A tibble: 5 x 2
##   EMPRESA_SIGLA total_horas
##   <chr>          <dbl>
## 1 AZU            255711.
## 2 TAM            254084.
## 3 GLO            249348.
## 4 TAP            26560.
## 5 UAL            17948.

```

5. Considerando apenas empresas internacionais. Para o ano de 2022, quais são as cinco empresas que mais carregaram passageiros de forma gratuita?

```

df %>% filter(ANO == 2022, EMPRESA_NACIONALIDADE == "ESTRANGEIRA") %>%
  group_by(EMPRESA_SIGLA) %>%
  summarise(total_passageiros_gratis = sum(as.numeric(PASSAGEIROS_GRATIS), na.rm = TRUE)) %>%
  arrange(desc(total_passageiros_gratis)) %>%
  head(5)

```

```

## # A tibble: 5 x 2
##   EMPRESA_SIGLA total_passageiros_gratis
##   <chr>          <dbl>
## 1 AAL            32180
## 2 LAN            26447
## 3 UAL            23259
## 4 TAP            16521
## 5 CMP            10879

```

6. Considerando o consumo de combustível por distância voada em km. Qual empresa nacional tem a menor taxa média de consumo, considerando o ano de 2022? Considere o consumo como a distância/combustível e ignore voos com NA.

```

df %>% filter(ANO == 2022, EMPRESA_NACIONALIDADE == "BRASILEIRA",
               !is.na(COMBUSTIVEL_LITROS), !is.na(DISTANCIA_VOADA_KM)) %>%
  mutate(taxa_consumo = as.numeric(DISTANCIA_VOADA_KM) / as.numeric(COMBUSTIVEL_LITROS)) %>%
  group_by(EMPRESA_SIGLA) %>%
  summarise(media_taxa_consumo = mean(taxa_consumo, na.rm = TRUE)) %>%
  arrange(media_taxa_consumo) %>%
  slice(1)

```

```

## # A tibble: 1 x 2
##   EMPRESA_SIGLA media_taxa_consumo
##   <chr>          <dbl>
## 1 LTG            0.119

```

7. Considere voos saindo do sudeste. Ordene as UFs de destino de acordo com sua frequência (linhas na base de dados).

```
df %>% filter(AEROPORTO_DE_ORIGEM_REGIAO == "SUDESTE",
                 AEROPORTO_DE_DESTINO_UF != "", AEROPORTO_DE_DESTINO_UF != " ") %>%
  count(AEROPORTO_DE_DESTINO_UF, sort = TRUE)
```

AEROPORTO_DE_DESTINO_UF	n
SP	68317
MG	33974
RJ	27783
BA	24429
PR	18619
DF	12364
SC	11854
RS	11016
PE	9587
CE	8377
ES	7167
GO	6989
MT	5376
RN	5263
AL	4962
AM	4949
PA	4915
MS	4583
PB	3111
MA	2509
SE	2325
PI	1640
TO	1588
RO	1528
AP	1060
RR	574
AC	481

8. Qual é o aeroporto de origem mais frequente de voos NÃO REGULARES (veja a coluna GRUPO DE VOO)?

```
df %>% filter(GRUPO_DE_VOO == "NÃO REGULAR") %>%
  count(AEROPORTO_DE_ORIGEM_NOME, sort = TRUE) %>%
  slice(1)
```

AEROPORTO_DE_ORIGEM_NOME	n
GUARULHOS	23181

9. Considere os voos DOMESTICOS e a proporção de voos NÃO REGULAR. Quais são os 10 aeroportos de destino mais frequente?

```
df %>% filter(NATUREZA == "DOMÉSTICA", GRUPO_DE_VOO == "NÃO REGULAR") %>%
  count(AEROPORTO_DE_DESTINO_SIGLA, sort = TRUE) %>%
  slice(1:10)
```

```

##      AEROPORTO_DE_DESTINO_SIGLA      n
## 1                      SBGR  17267
## 2                      SBSP  9910
## 3                      SBGL  9687
## 4                      SBKP  8791
## 5                      SBBR  8601
## 6                      SBCF  8212
## 7                      SBSV  7946
## 8                      SBEG  7332
## 9                      SBRF  6747
## 10                     SBFZ  6396

```

10. Considere todas as combinações de aeroportos de origem e destino. Quais são as dez combinações mais frequentes em termos de total de decolagens?

```

df %>% filter(DECOLAGENS != "" & !is.na(DECOLAGENS)) %>%
  mutate(DECOLAGENS = as.numeric(DECOLAGENS)) %>%
  group_by(AEROPORTO_DE_ORIGEM_NOME, AEROPORTO_DE_DESTINO_NOME) %>%
  summarise(total_decolagens = sum(DECOLAGENS, na.rm = TRUE), .groups = "drop") %>%
  arrange(desc(total_decolagens)) %>%
  slice(1:10)

```

```

## # A tibble: 10 x 3
##   AEROPORTO_DE_ORIGEM_NOME AEROPORTO_DE_DESTINO_NOME total_decolagens
##   <chr>                   <chr>                         <dbl>
## 1 SÃO PAULO                RIO DE JANEIRO            559712
## 2 RIO DE JANEIRO             SÃO PAULO                  555629
## 3 GUARULHOS                 RIO DE JANEIRO            225399
## 4 RIO DE JANEIRO              GUARULHOS                224581
## 5 SÃO PAULO                 BRASÍLIA                  198693
## 6 BRASÍLIA                  SÃO PAULO                  197155
## 7 BRASÍLIA                  RIO DE JANEIRO            183580
## 8 RIO DE JANEIRO              BRASÍLIA                  182111
## 9 SÃO PAULO                 SÃO JOSÉ DOS PINHAIS    162698
## 10 SÃO JOSÉ DOS PINHAIS       SÃO PAULO                 161435

```