Dados Tabulares de Grande Volume

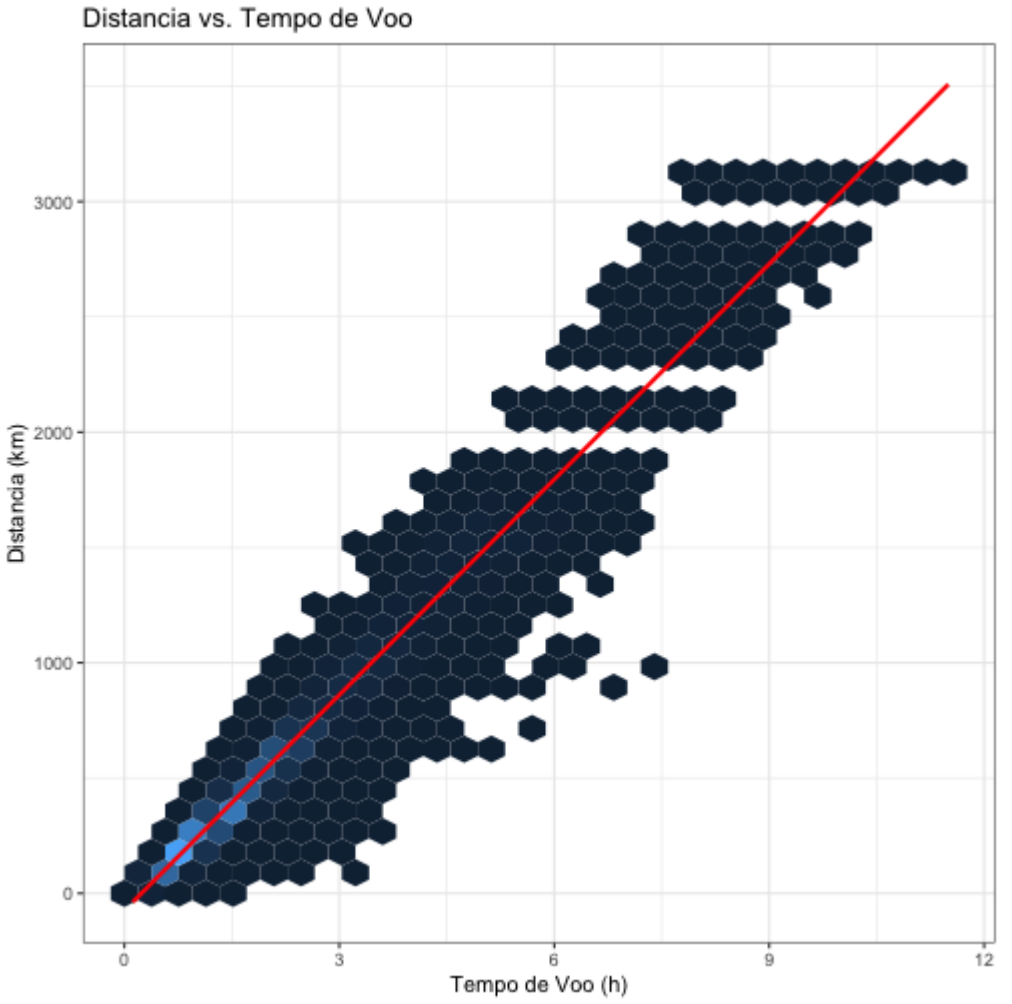
Inferência e Ciência de Dados

Benilton Carvalho, Guilherme Ludwig, Tatiana Benaglia

flights.csv.zip

- Possui 5714008 observações;
- Tem 31 variáveis;
- Ocupa 1GB de RAM;
- Entre as variáveis:
  - Dia, mês, ano, dia da semana;
  - Cia aérea, número do vôo, registro do avião;
  - Aeroportos de partida e de chegada;
  - Horários de partida e chegada (reais e programados);
  - Tempo de vôo e distância voada;
  - Atraso na chegada.

Distância vs. Tempo de Vôo



Reta de Regressão

$$Distancia_i = b_0 + b_1 \times Tempo_i + \epsilon_i$$

term	estimate
(Intercept)	-73.95665
AIR_TIME	311.46177

- Como estimar b_0 e b_1?

Estimadores via Mínimos Quadrados

$$
\begin{aligned}
y_i &= b_0 + b_1 x_i + \epsilon_i \\
\hat{b}_0 &= \bar{y} - \hat{b}_1 \bar{x} \\
\hat{b}_1 &= \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2} \\
&= \frac{\sum x_i y_i - \frac{\sum x_i \sum y_i}{n}}{\sum x_i^2 - \frac{(\sum x_i)^2}{n}}
\end{aligned}
$$

- Precisamos pensar em meios de calcular estas estatísticas usando apenas partes do conjunto de dados;
- Estas estatísticas "parciais" devem poder ser combinadas;
- Estatísticas suficientes:
  - $\sum x_i$;
  - $\sum y_i$;
  - $\sum x_i y_i$;
  - $\sum x_i^2$;
  - n;

Particionando operações

$$\sum x_i y_i = \sum_{i=1}^{k_1} x_i y_i + \sum_{i=k_1+1}^{k_2} x_i y_i + \cdots$$

$$\sum x_i = \sum_{i=1}^{k_1} x_i + \sum_{i=k_1+1}^{k_2} x_i + \cdots$$

$$\sum x_i^2 = \sum_{i=1}^{k_1} x_i^2 + \sum_{i=k_1+1}^{k_2} x_i^2 + \cdots$$

Para um bloco dos dados

$$\hat{b}_0 = \bar{y} - \hat{b}_1 \bar{x}$$

$$\hat{b}_1 = \frac{\sum x_i y_i - \frac{\sum x_i \sum y_i}{n}}{\sum x_i^2 - \frac{(\sum x_i)^2}{n}}$$

```
getStats = function(input, pos){
  input %>% filter(!is.na(AIR_TIME), !is.na(DISTANCE)) %>%
    mutate(AIR_TIME=AIR_TIME/60, DISTANCE=DISTANCE/1.6) %>%
    summarise(Sxy=sum(AIR_TIME*DISTANCE), Sx=sum(AIR_TIME),
              Sy=sum(DISTANCE), Sx2=sum(AIR_TIME^2), n=n())
}
computeStats = function(stats){
  stats %>%
    summarise(num = sum(Sxy)-(sum(Sx)*sum(Sy))/sum(n),
              den = sum(Sx2)-(sum(Sx)^2)/sum(n),
              b1 = num/den,
              b0 = sum(Sy)/sum(n)-b1*sum(Sx)/sum(n)) %>%
    select(b0, b1) %>% gather(key='coef', value='valor') %>%
    knitr::kable('html')
}
```

Para um bloco dos dados

```
in1 %>% getStats() %>% computeStats()
```

coef	valor
b0	-73.95665
b1	311.46177

Processando Dados em Lote

- O pacote `readr` possui funções de importação aprimoradas;
- São funções mais rápidas e inteligentes;
- Uma classe de funções é a de operação em porções de arquivos:
  - `read_csv_chunked`;
  - `read_csv2_chunked`;
  - `read_delim_chunked`;
  - `read_tsv_chunked`;
- As funções `read_***_chunked` aceitam argumentos especiais:
  - `chunk_size`: número de linhas a serem importadas por iteração;
  - `callback`: função que é executada em cada porção dos dados;
- O argumento `callback` deve instanciar:
  - `DataFrameCallback`: se se deseja combinar resultados tabulares;
  - `ListCallback`: se se deseja combinar resultados 'flexíveis';
  - `SideEffectChunkCallback`: se se deseja visualizar efeitos colaterais.

Importação de Dados com Leitura em Lotes

```
in2 = read_csv_chunked('../dados/flights.csv.zip',
                       callback=DataFrameCallback$new(getStats),
                       chunk_size = 1e6)
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   AIRLINE = col_character(),
##   TAIL_NUMBER = col_character(),
##   ORIGIN_AIRPORT = col_character(),
##   DESTINATION_AIRPORT = col_character(),
##   SCHEDULED_DEPARTURE = col_character(),
##   DEPARTURE_TIME = col_character(),
##   WHEELS_OFF = col_character(),
##   WHEELS_ON = col_character(),
##   SCHEDULED_ARRIVAL = col_character(),
##   ARRIVAL_TIME = col_character(),
##   CANCELLATION_REASON = col_character()
## )

## See spec(...) for full column specifications.
```

Importação de Dados com Leitura em Lotes

```
in2
```

```
## # A tibble: 6 x 5
##           Sxy       Sx          Sy      Sx2      n
##         <dbl>    <dbl>       <dbl>    <dbl>  <int>
## 1 1323121892. 1797787. 482227611. 4748642. 957394
## 2 1392718440. 1872273. 506332984. 4951091. 989242
## 3 1433684731. 1873591  512218232. 5025248. 981161
## 4 1441177514. 1871883. 517199389. 4996242. 987786
## 5 1387819679. 1849173  506928412. 4850165. 993158
## 6 1170987680. 1545399. 419439209. 4160835. 805267
```

```
in2 %>% computeStats()
```

coef	valor
b0	-73.95665
b1	311.46177

Importação de Dados - Colunas Específicas

- As funções de importação `read_***` possuem um argumento `col_types`;
- Opções válidas para `col_types`:
  - Especificação criada por `cols()`: todas as colunas;
  - Especificação criada por `cols_only()`: apenas um subconjunto;
- `cols()`: `cols(NOME=col_TIPO())`
  - `cols(a=col_integer());`
  - `cols(a='i')`
- `cols_only()`: `cols_only(NOME=col_TIPO())`
  - `cols_only(a=col_integer())`
  - `cols_only(a='i')`

```
mycols = cols_only(AIR_TIME='i', DISTANCE='i')
in3 = read_csv_chunked('../dados/flights.csv.zip',
                       callback=DataFrameCallback$new(getStats),
                       chunk_size = 1e6, col_types=mycols)
```

Importação de Dados - Colunas Específicas

```
in3
```

```
## # A tibble: 6 x 5
##           Sxy       Sx          Sy      Sx2      n
##         <dbl>    <dbl>       <dbl>    <dbl>  <int>
## 1 1323121892. 1797787. 482227611. 4748642. 957394
## 2 1392718440. 1872273. 506332984. 4951091. 989242
## 3 1433684731. 1873591  512218232. 5025248. 981161
## 4 1441177514. 1871883. 517199389. 4996242. 987786
## 5 1387819679. 1849173  506928412. 4850165. 993158
## 6 1170987680. 1545399. 419439209. 4160835. 805267
```

```
in3 %>% computeStats()
```

coef	valor
b0	-73.95665
b1	311.46177

Atividade para Laboratório

- Lendo 1 milhão de observações por vez, determine o percentual de vôos, por Cia. Aérea, que apresentou atraso na chegada (`ARRIVAL_DELAY`) superior a 10 minutos.
- As companhias a serem utilizadas são: AA, DL, UA e US.
- A estatística de interesse deve ser calculada para cada um dos dias de 2015.
- Para a determinação deste percentual de atrasos, apenas verbos do pacote `dplyr` e comandos de importação do pacote `readr` podem ser utilizados.
- Os resultados para cada Cia. Aérea devem ser apresentados em um formato de calendário.

Calendário Esperado

```
## `summarise()` regrouping output by 'MONTH', 'DAY' (override with `.groups`
```