

Cumbuca Case

Matheus Pasche

29/05/2022

DNA

O Rbase contém uma função relativamente pouco utilizada, mas poderosa para traduções simples: `chartr`

```
## nucleotides
## Esta função traduz DNA para RNA
## @param dna character com nucleotideos
##
## @return
## @export
##
## @examples
## nucleotides("CGAT")
nucleotides <- function(dna) {

  if(!is.character(dna)){stop("Please insert character nucleotides, such as ACGT")}

  old <- "GCTA"
  new <- "CGAU"

  translate <- chartr(
    old = old,
    new = new,
    x = dna
  )

  return(translate)
}

##
test_that("Translate dna", {
  expect_equal(nucleotides("GACATGG"), "CUGUACC")
  expect_equal(nucleotides("AAATTT"), "UUUAAA")
  expect_error(nucleotides(1234))
})

## Test passed
```

NycFlights

É possível identificar que o dataframe possui alguns voos sem informação sobre partida e chegada. Como representam menos de 3% dos registros, optarei por eliminar a fim de evitar problemas.

```

flights <- nycflights13::flights %>%
  mutate(total_delay = arr_delay + dep_delay, month_year = floor_date(time_hour, 'month'),
         time = as_date(time_hour),
         month_year = ym(paste0(year, month)),
         status = case_when(total_delay > 0 ~ 'delay', total_delay <0~ 'early', total_delay ==0 ~'in time'))

flights %>%
  summarise(across(everything(), ~ sum(is.na(.)))) %>%
  pivot_longer(cols = everything(), names_to = 'Variable', values_to = 'NA Count') %>%
  knitr::kable(caption = 'Contagem de NAs')

```

Table 1: Contagem de NAs

Variable	NA Count
year	0
month	0
day	0
dep_time	8255
sched_dep_time	0
dep_delay	8255
arr_time	8713
sched_arr_time	0
arr_delay	9430
carrier	0
flight	0
tailnum	2512
origin	0
dest	0
air_time	9430
distance	0
hour	0
minute	0
time_hour	0
total_delay	9430
month_year	0
time	0
status	9430

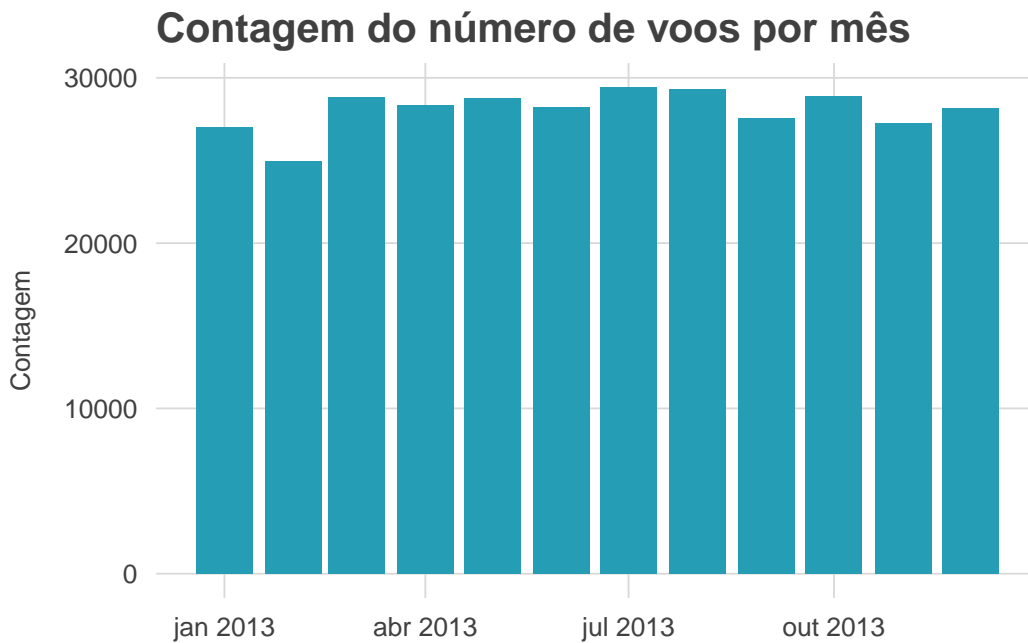
Questão 1:

A quantidade total de voos, que inclui atrasados e não atrasados, não variou consideravelmente ao longo do tempo

```

flights %>%
  count(month_year) %>%
  ggplot(aes(x = month_year, y = n, label = n, vjust = 1))+
  geom_bar(stat = 'identity', fill = pilot_color(2))+
  theme_pilot()+
  #geom_text_pilot()+
  labs(y = "Contagem", x = "", title = "Contagem do número de voos por mês")

```



Tendo em vista que estamos interessados em analisar o atraso, foram retirados os voos com saldo total (embarque + desembarque) negativo. Abaixo podemos identificar os primeiros registros do atraso médio, mediano e estatísticas percentuais para cada dia do ano. Evidentemente não é prático exibir todos os 365 dias.

```
stats <- function(.data, variable) {
  .data %>%
    summarise(
      count = n(),
      mean = mean({{variable}}),
      median = median({{variable}}),
      sd = sd({{variable}}),
      var = var({{variable}}),
      'p.01'=quantile({{variable}}, probs = .1),
      'p.025'=quantile({{variable}}, probs = .25),
      'p.09'=quantile({{variable}}, probs = .9),
      'p.99'=quantile({{variable}}, probs = .99)
    )
}

lateDay = flights %>%
  filter(!is.na(air_time), status == 'delay') %>%
  with_groups(time, ~stats(., variable = total_delay))

lateDay%>%
  head() %>%
  knitr::kable(caption = 'Registros de estatísticas descritivas por dia', digits=1)
```

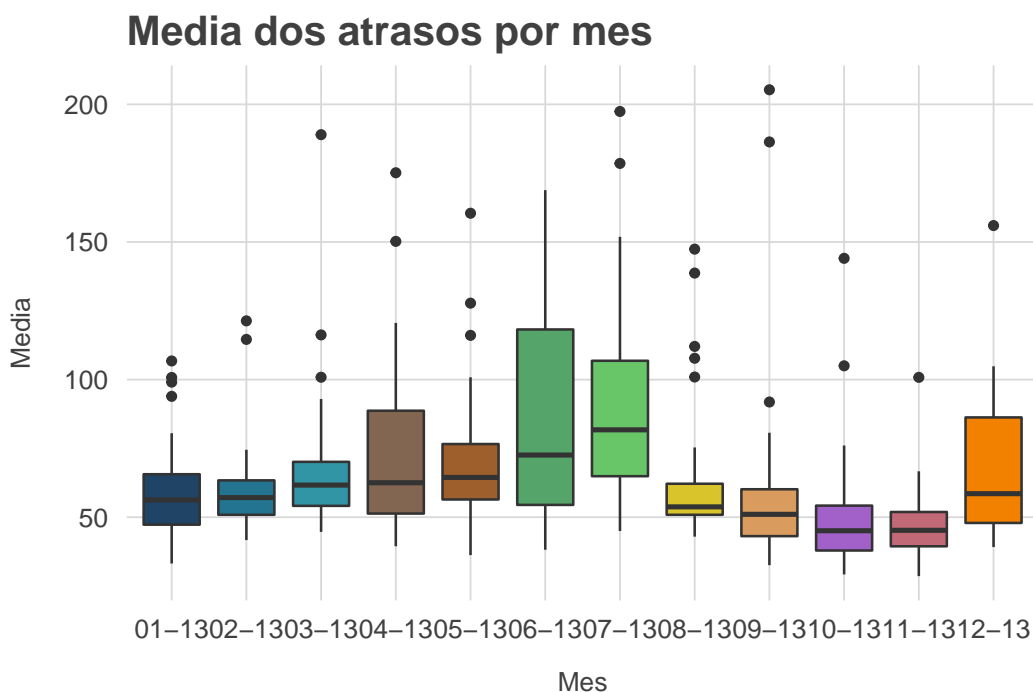
Table 2: Registros de estatísticas descritivas por dia

time	count	mean	median	sd	var	p.01	p.025	p.09	p.99
2013-01-01	447	57.1	23.0	117.3	13769.2	4	9.0	139.8	514.7
2013-01-02	540	56.8	24.5	88.6	7841.8	4	9.0	157.2	399.3
2013-01-03	455	51.4	24.0	74.5	5542.9	4	10.0	129.0	349.5
2013-01-04	318	56.0	30.5	71.7	5143.6	5	10.2	130.3	334.4
2013-01-05	255	41.8	17.0	72.2	5214.7	4	7.0	120.0	340.3
2013-01-06	371	44.1	23.0	56.3	3171.6	4	9.0	111.0	284.0

Para fins de análise anual, agregaremos os dados diários médios por mês. Inicialmente podemos verificar que os períodos de maior atraso mediano são junho-julho, períodos de férias escolares e verão, e dezembro, marcado pelas festas de natal e ano novo.

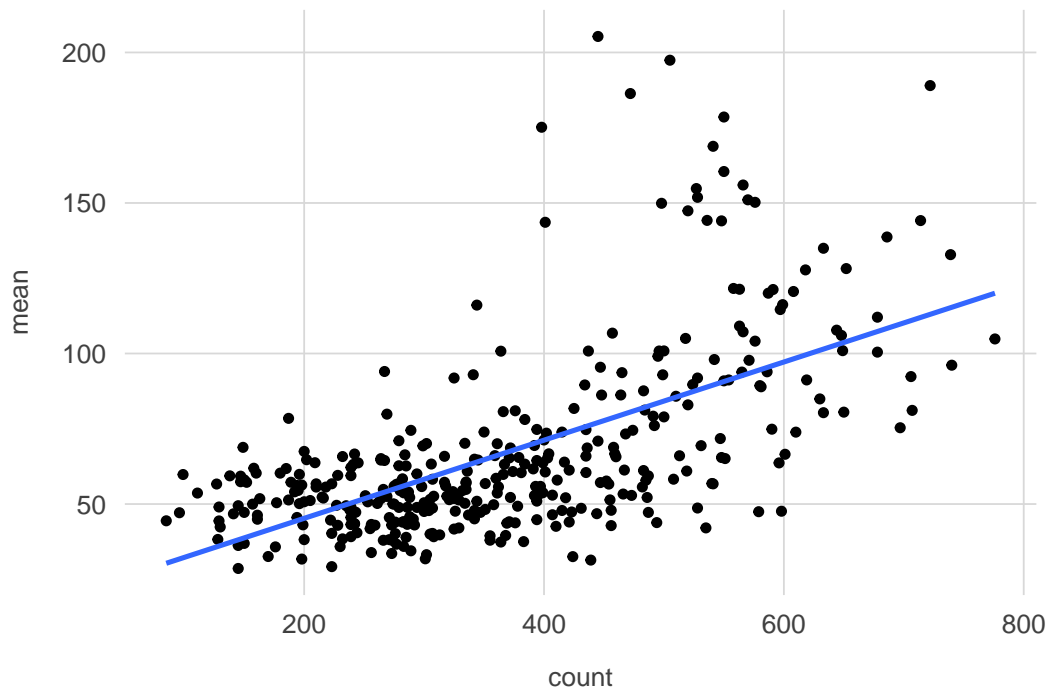
```
lateMonth = flights %>%
  filter(!is.na(air_time), status == 'delay') %>%
  mutate(month_year = floor_date(time, 'month')) %>%
  with_groups(month_year, ~stats(., variable = total_delay)) %>%
  knitr::kable(caption = 'Registros de estatísticas descritivas por mês', digits=1)
```

```
lateDay %>%
  mutate(month = format(time, '%m-%y')) %>%
  ggplot(aes(x = month, y = mean, fill = month))+
  geom_boxplot(show.legend = F)+
  pilot::scale_fill_pilot()+
  pilot::theme_pilot()+
  labs(title = 'Media dos atrasos por mes', y = 'Media', x = 'Mes')
```



O maior atraso nesses meses pode ser explicado em partes pela maior quantidade de voos. Uma vez que estamos observando apenas os voos com atraso, é evidente que um número maior de voos tende a causar um número maior de atrasos, seja porque estamos realizando o experimento o evento mais vezes, seja por um aumento das atribuições das equipes nos meses de pico.

```
lateDay %>%  
  ggplot(aes(x = count, y = mean))+  
  geom_point()+  
  geom_smooth(method="lm", formula= (y ~ x), se=FALSE)+  
  pilot::theme_pilot()
```



Questão 2:

Ao observar

```
airlines <- nycflights13::airlines  
flights <- flights %>% left_join(airlines)
```

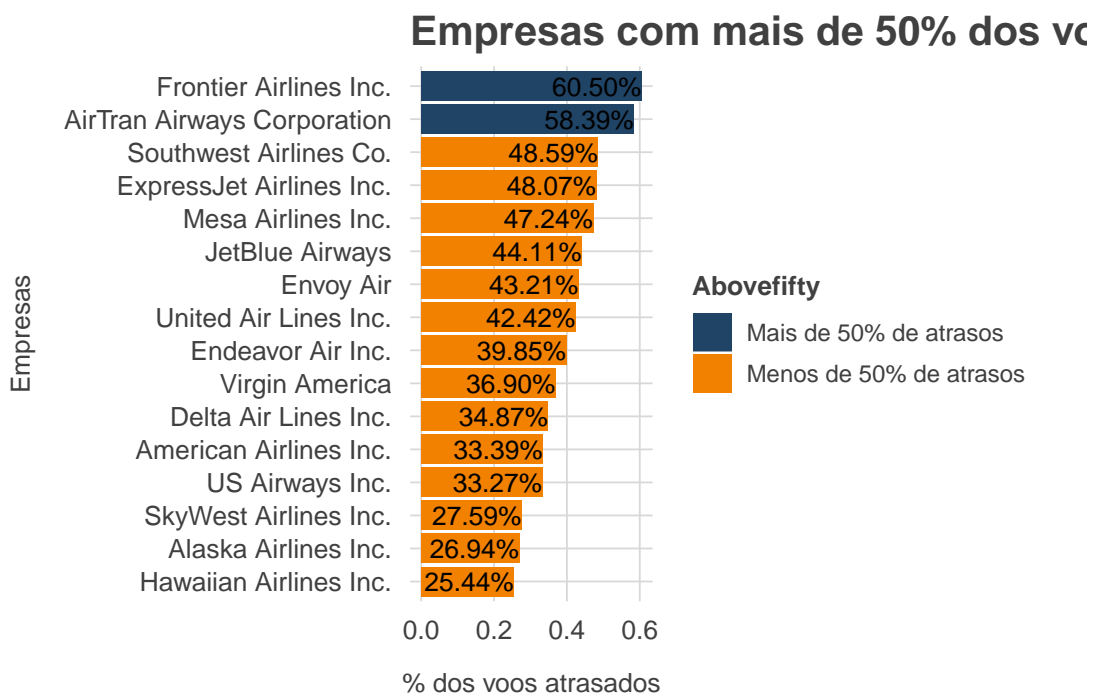
```
## Joining, by = "carrier"
```

```
Companies = flights %>%  
  filter(!is.na(air_time)) %>%  
  with_groups(c(name,status), ~summarise(.,count = n())) %>%  
  with_groups(name, ~mutate(., Razao = count/sum(count)))
```

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

```
Companies %>%
```

```
  mutate(Abovefifty = case_when (Razao >= .5 ~ 'Mais de 50% de atrasos', TRUE ~ 'Menos de 50% de atrasos')
  filter(status == 'delay') %>%
  ggplot(aes(x = Razao, y = reorder(name, Razao), fill = Abovefifty)) +
  geom_bar(stat = 'identity') +
  geom_text(aes(label = scales::percent(Razao)), hjust = 1) +
  pilot::scale_fill_pilot() +
  pilot::theme_pilot() +
  labs(title = 'Empresas com mais de 50% dos voos atrasados', x = '% dos voos atrasados', y = 'Empresas')
```



```
Companies %>%
```

```
  filter(status == 'delay') %>%
  mutate(Razao = count/sum(count),
         Abovefifteen = case_when (Razao >= .15 ~ 'Mais de 15% de atrasos', TRUE ~ 'Menos de 15% de atrasos')
  ggplot(aes(x = Razao, y = reorder(name, Razao), fill = Abovefifteen)) +
  geom_bar(stat = 'identity') +
  pilot::scale_fill_pilot()
```

