Tutorial 5

Hotel bookings - data wrangling

Mine Çetinkaya-Rundel - Renata Oliveira (tradução)

```
library(tidyverse)
```

```
# From TidyTuesday: https://github.com/rfordatascience/tidytuesday/blob/master/data/2020/2020-02-11/rea
hotels <- read_csv("https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2020/2020
```

Exercícios

Exercício 1.

As pessoas estão viajando por um capricho? Vamos ver...

Preencha os espaços em branco para filtragem de reservas de hotel onde o hóspede é **não** dos EUA (código do país `USA`) e o `lead_time` é menos de 1 dia.

Nota: Você precisará definir `eval=TRUE` quando tiver uma resposta que queira experimentar.

```
# on the fly
hotels %>%
  filter(
    country != "USA",
    lead_time < 1
    )
```

```
## # A tibble: 6,174 x 32
##    hotel        is_canceled lead_time arrival_date_year arrival_date_month
##    <chr>              <dbl>     <dbl>             <dbl> <chr>
##  1 Resort Hotel           0         0              2015 July
##  2 Resort Hotel           0         0              2015 July
##  3 Resort Hotel           0         0              2015 July
##  4 Resort Hotel           0         0              2015 July
##  5 Resort Hotel           0         0              2015 July
##  6 Resort Hotel           0         0              2015 July
##  7 Resort Hotel           0         0              2015 July
##  8 Resort Hotel           0         0              2015 July
##  9 Resort Hotel           0         0              2015 July
## 10 Resort Hotel           0         0              2015 July
## # ... with 6,164 more rows, and 27 more variables:
## #   arrival_date_week_number <dbl>, arrival_date_day_of_month <dbl>,
## #   stays_in_weekend_nights <dbl>, stays_in_week_nights <dbl>, adults <dbl>,
## #   children <dbl>, babies <dbl>, meal <chr>, country <chr>,
## #   market_segment <chr>, distribution_channel <chr>, is_repeated_guest <dbl>,
## #   previous_cancellations <dbl>, previous_bookings_not_canceled <dbl>,
## #   reserved_room_type <chr>, assigned_room_type <chr>, booking_changes <dbl>,
## #   deposit_type <chr>, agent <chr>, company <chr>, days_in_waiting_list <dbl>,
## #   customer_type <chr>, adr <dbl>, required_car_parking_spaces <dbl>,
```

```
## #   total_of_special_requests <dbl>, reservation_status <chr>,
## #   reservation_status_date <date>
```

```r
# Com registro de objeto

hotel <- hotels %>%
  filter(country != "USA", lead_time < 1) %>%
  select(country, lead_time)

# Sem pipe

hotels_sem_pipe <- filter(hotels, country != "USA", lead_time < 1)
hotels_sem_pipe <- select(hotels_sem_pipe, country, lead_time)
```

Exercício 2.

Quantas marcações envolvem pelo menos 1 criança **ou** bebê?

No seguinte `chunk`, substitua

- `[AT LEAST]` com o operador lógico para "pelo menos" (em dois lugares)
- `[OR]` com o operador lógico para "ou"

Nota: Você precisará definir `eval=TRUE` quando tiver uma resposta que queira experimentar.

```r
hotels %>%
  filter(children >= 1 | babies >= 1)

# Com registro de objeto
hotel_bebe_chil <- hotels %>%
  filter(children >= 1 | babies >= 1)

# Sem pipe

hotel_bebe_chil_sem_pipe <- filter(hotels, children >= 1 | babies >= 1)
```

Exercício 3.

Você acha que é mais provável encontrar reservas com crianças ou bebês em hotéis urbanos ou resorts hoteleiros? Teste sua intuição.

Usando `filter()` determine o número de reservas em hotéis resort que têm mais de 1 criança **ou** bebê no quarto?

Então, faça o mesmo para hotéis urbanos, e compare o número de linhas no dataframe filtrado resultantes.

```r
# Com registro de objeto
hotel_bebe_chil_resort <- hotels %>%
  filter(children >= 1 | babies >= 1) %>%
  filter(hotel == "Resort Hotel")

# Sem pipe

hotel_bebe_chil_sem_pipe <- filter(hotels, children >= 1 | babies >= 1)
hotel_bebe_chil_resort_sem_pipe <- filter(hotel_bebe_chil_resort, hotel == "Resort Hotel")
```

```r
# Com registro de objeto
hotel_bebe_chil_city <- hotels %>%
```

```
  filter(children >= 1 | babies >= 1) %>%
  mutate(hotel = tolower(hotel)) %>%
  filter(hotel == "city hotel")

# Sem pipe

hotel_bebe_chil_sem_pipe <- filter(hotels, children >= 1 | babies >= 1)
hotel_bebe_chil_city_sem_pipe <- filter(hotel_bebe_chil_sem_pipe, hotel == "City Hotel")
```

```
hotel_bebe_chil_class <- hotels %>%
  filter(children >= 1 | babies >= 1) %>%
  group_by(hotel) %>%
  summarise(n = max(stays_in_weekend_nights))
```

Exercício 4

Criar uma tabela de freqüência do número de `adults` em uma reserva.

Mostre os resultados em ordem decrescente para que a observação mais comum esteja no topo.

Qual é o número mais comum de adultos em reservas neste conjunto de dados?

Há algum resultado surpreendente?

Nota: Não esqueça de rotular também seu `chunk` R (onde diz `lable-me-1`). Seu rótulo deve ser curto, informativo, e não deve incluir espaços. Também não deve repetir uma etiqueta anterior, caso contrário o R Markdown lhe dará um erro sobre a repetição de etiquetas R em pedaços.

Exercício 5

Repita o exercício 4, uma vez para reservas canceladas (`is_canceled` codificado como 1) e uma vez para reservas não canceladas (`is_canceled` codificado como 0).

O que isto revela sobre os resultados surpreendentes que você observou no exercício anterior?

Note: Não se esqueça de rotular também seu `chunk` de R (onde diz `label-me-2`).

```
# add code here
# pay attention to correctness and code style
```

Exercício 6

Calcular a tarifa mínima, média, mediana e máxima média diária (`adr`) agrupados por tipo de `hotel` para que você possa obter estas estatísticas separadamente para hotéis de resorts e cidades.

Que tipo de hotel é mais caro, em média?

```
# add code here
# pay attention to correctness and code style
```

Exercício 7

Observamos dois valores incomuns nas estatísticas resumidas acima – um mínimo negativo, e um máximo muito alto). Que tipos de hotéis são estes?

Localize estas observações no conjunto de dados e descubra a data de chegada (ano e mês), assim como quantas pessoas (adultos, crianças e bebês) permaneceram no quarto.

Você pode investigar os dados no espectador para localizar estes valores, mas de preferência você deve identificá-los de forma reprodutível com algum código.

Dica: Por exemplo, você pode `filter` para o dado quantidade `adr` e `select` as colunas relevantes.

```
# add code here
# pay attention to correctness and code style
```

Dicionário de dados

Abaixo está o dicionário de dados completo. Note que é longo (há muitas variáveis nos dados), mas utilizamos um conjunto limitado de variáveis para nossa análise.

variable	class	description
hotel	character	Hotel (H1 = Resort Hotel or H2 = City Hotel)
is_canceled	double	Value indicating if the booking was canceled (1) or not (0)
lead_time	double	Number of days that elapsed between the entering date of the booking into the PMS and the arrival date
arrival_date_year	double	Year of arrival date
arrival_date_month	character	Month of arrival date
arrival_date_week_number	double	Week number of year for arrival date
arrival_date_day_of_month	double	Day of arrival date
stays_in_weekend_nights	double	Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel
stays_in_week_nights	double	Number of week nights (Monday to Friday) the guest stayed or booked to stay at the hotel
adults	double	Number of adults
children	double	Number of children
babies	double	Number of babies
meal	character	Type of meal booked. Categories are presented in standard hospitality meal packages: Undefined/SC – no meal package;BB – Bed & Breakfast; HB – Half board (breakfast and one other meal – usually dinner); FB – Full board (breakfast, lunch and dinner)
country	character	Country of origin. Categories are represented in the ISO 3155–3:2013 format
market_segment	character	Market segment designation. In categories, the term “TA” means “Travel Agents” and “TO” means “Tour Operators”
distribution_channel	character	Booking distribution channel. The term “TA” means “Travel Agents” and “TO” means “Tour Operators”
is_repeated_guest	double	Value indicating if the booking name was from a repeated guest (1) or not (0)
previous_cancellations	double	Number of previous bookings that were cancelled by the customer prior to the current booking
previous_bookings_not_canceled	double	Number of previous bookings not cancelled by the customer prior to the current booking
reserved_room_type	character	Code of room type reserved. Code is presented instead of designation for anonymity reasons
assigned_room_type	character	Code for the type of room assigned to the booking. Sometimes the assigned room type differs from the reserved room type due to hotel operation reasons (e.g. overbooking) or by customer request. Code is presented instead of designation for anonymity reasons
booking_changes	double	Number of changes/amendments made to the booking from the moment the booking was entered on the PMS until the moment of check-in or cancellation
deposit_type	character	Indication on if the customer made a deposit to guarantee the booking. This variable can assume three categories:No Deposit – no deposit was made;Non Refund – a deposit was made in the value of the total stay cost;Refundable – a deposit was made with a value under the total cost of stay.
agent	character	ID of the travel agency that made the booking
company	character	ID of the company/entity that made the booking or responsible for paying the booking. ID is presented instead of designation for anonymity reasons
days_in_waiting_list	double	Number of days the booking was in the waiting list before it was confirmed to the customer

variable	class	description
customer_type	char	Type of booking, assuming one of four categories:Contract - when the booking has an allotment or other type of contract associated to it;Group – when the booking is associated to a group;Transient – when the booking is not part of a group or contract, and is not associated to other transient booking;Transient-party – when the booking is transient, but is associated to at least other transient booking
adr	double	Average Daily Rate as defined by dividing the sum of all lodging transactions by the total number of staying nights
required_car_parking_spaces	double	Number of car parking spaces required by the customer
total_of_special_requests	double	Number of special requests made by the customer (e.g. twin bed or high floor)
reservation_status	char	Reservation last status, assuming one of three categories:Canceled – booking was canceled by the customer;Check-Out – customer has checked in but already departed;No-Show – customer did not check-in and did inform the hotel of the reason why
reservation_status_date	date	Date at which the last status was set. This variable can be used in conjunction with the ReservationStatus to understand when was the booking canceled or when did the customer checked-out of the hotel