# AE 05: Data types and classes

APPLICATION EXERCISE ANSWERS

MODIFIED
September 20, 2024

## Packages

We will use the following packages in this application exercise.

- **tidyverse**: For data import, wrangling, and visualization.
- **skimr**: For summarizing the entire data frame at once.
- **scales**: For better axis labels.

```
library(tidyverse)
library(skimr)
library(scales)
```

## Hotel bookings

Antonio, Almeida, and Nunes (2019) collected detailed information on hotel bookings from two hotels (one resort hotel and one city hotel) in Portugal. The data set contains information such as when the booking was made, length of stay, number of adults, number of children, and number of available parking spaces.

### Load the data

The data is stored in `data/hotels-tt.csv`. Let's load the data file and examine it's contents. Since the dataset is substantially large (nearly 30 variables and over 100,000 observations), we'll use `skimr::skim()` to provide a compact summary of the data.

```
hotels <- read_csv("data/hotels-tt.csv")
skim(hotels) # much more useful to run interactively in the console
```

Data summary

| | |
|---|---|
| Name | hotels |
| Number of rows | 119390 |
| Number of columns | 29 |
| _____ | |
| Column type frequency: | |

| | |
|---|---|
| character | 13 |
| Date | 1 |
| numeric | 15 |
| _____ | |
| Group variables | None |

### Variable type: character

| skim_variable | n_missing | complete_rate | min | max | empty | n_unique | whitespace |
|---|---|---|---|---|---|---|---|
| hotel | 0 | 1 | 10 | 12 | 0 | 2 | 0 |
| arrival_date | 0 | 1 | 11 | 18 | 0 | 793 | 0 |
| meal | 0 | 1 | 2 | 9 | 0 | 5 | 0 |
| country | 0 | 1 | 2 | 4 | 0 | 178 | 0 |
| market_segment | 0 | 1 | 6 | 13 | 0 | 8 | 0 |
| distribution_channel | 0 | 1 | 3 | 9 | 0 | 5 | 0 |
| reserved_room_type | 0 | 1 | 1 | 1 | 0 | 10 | 0 |
| assigned_room_type | 0 | 1 | 1 | 1 | 0 | 12 | 0 |
| deposit_type | 0 | 1 | 10 | 10 | 0 | 3 | 0 |
| agent | 0 | 1 | 1 | 4 | 0 | 334 | 0 |
| company | 0 | 1 | 1 | 4 | 0 | 353 | 0 |
| customer_type | 0 | 1 | 5 | 15 | 0 | 4 | 0 |
| reservation_status | 0 | 1 | 7 | 9 | 0 | 3 | 0 |

### Variable type: Date

| skim_variable | n_missing | complete_rate | min | max | median | n_unique |
|---|---|---|---|---|---|---|
| reservation_status_date | 0 | 1 | 2014-10-17 | 2017-09-14 | 2016-08-07 | 926 |

### Variable type: numeric

| skim_variable | n_missing | complete_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
|---|---|---|---|---|---|---|---|---|---|---|
| is_canceled | 0 | 1 | 0.37 | 0.48 | 0.00 | 0.00 | 0.00 | 1 | 1 | ▇▁▁▁▅ |
| lead_time | 0 | 1 | 104.01 | 106.86 | 0.00 | 18.00 | 69.00 | 160 | 737 | ▇▂▁▁▁ |
| stays_in_weekend_nights | 0 | 1 | 0.93 | 1.00 | 0.00 | 0.00 | 1.00 | 2 | 19 | ▇▁▁▁▁ |
| stays_in_week_nights | 0 | 1 | 2.50 | 1.91 | 0.00 | 1.00 | 2.00 | 3 | 50 | ▇▁▁▁▁ |
| adults | 0 | 1 | 1.86 | 0.58 | 0.00 | 2.00 | 2.00 | 2 | 55 | ▇▁▁▁▁ |
| children | 4 | 1 | 0.10 | 0.40 | 0.00 | 0.00 | 0.00 | 0 | 10 | ▇▁▁▁▁ |
| babies | 0 | 1 | 0.01 | 0.10 | 0.00 | 0.00 | 0.00 | 0 | 10 | ▇▁▁▁▁ |
| is_repeated_guest | 0 | 1 | 0.03 | 0.18 | 0.00 | 0.00 | 0.00 | 0 | 1 | ▇▁▁▁▁ |

| skim_variable | n_missing | complete_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
|---|---|---|---|---|---|---|---|---|---|---|
| previous_cancellations | 0 | 1 | 0.09 | 0.84 | 0.00 | 0.00 | 0.00 | 0 | 26 | ▇▁▁▁▁ |
| previous_bookings_not_canceled | 0 | 1 | 0.14 | 1.50 | 0.00 | 0.00 | 0.00 | 0 | 72 | ▇▁▁▁▁ |
| booking_changes | 0 | 1 | 0.22 | 0.65 | 0.00 | 0.00 | 0.00 | 0 | 21 | ▇▁▁▁▁ |
| days_in_waiting_list | 0 | 1 | 2.32 | 17.59 | 0.00 | 0.00 | 0.00 | 0 | 391 | ▇▁▁▁▁ |
| adr | 0 | 1 | 101.83 | 50.54 | -6.38 | 69.29 | 94.58 | 126 | 5400 | ▇▁▁▁▁ |
| required_car_parking_spaces | 0 | 1 | 0.06 | 0.25 | 0.00 | 0.00 | 0.00 | 0 | 8 | ▇▁▁▁▁ |
| total_of_special_requests | 0 | 1 | 0.57 | 0.79 | 0.00 | 0.00 | 0.00 | 1 | 5 | ▇▁▁▁▁ |

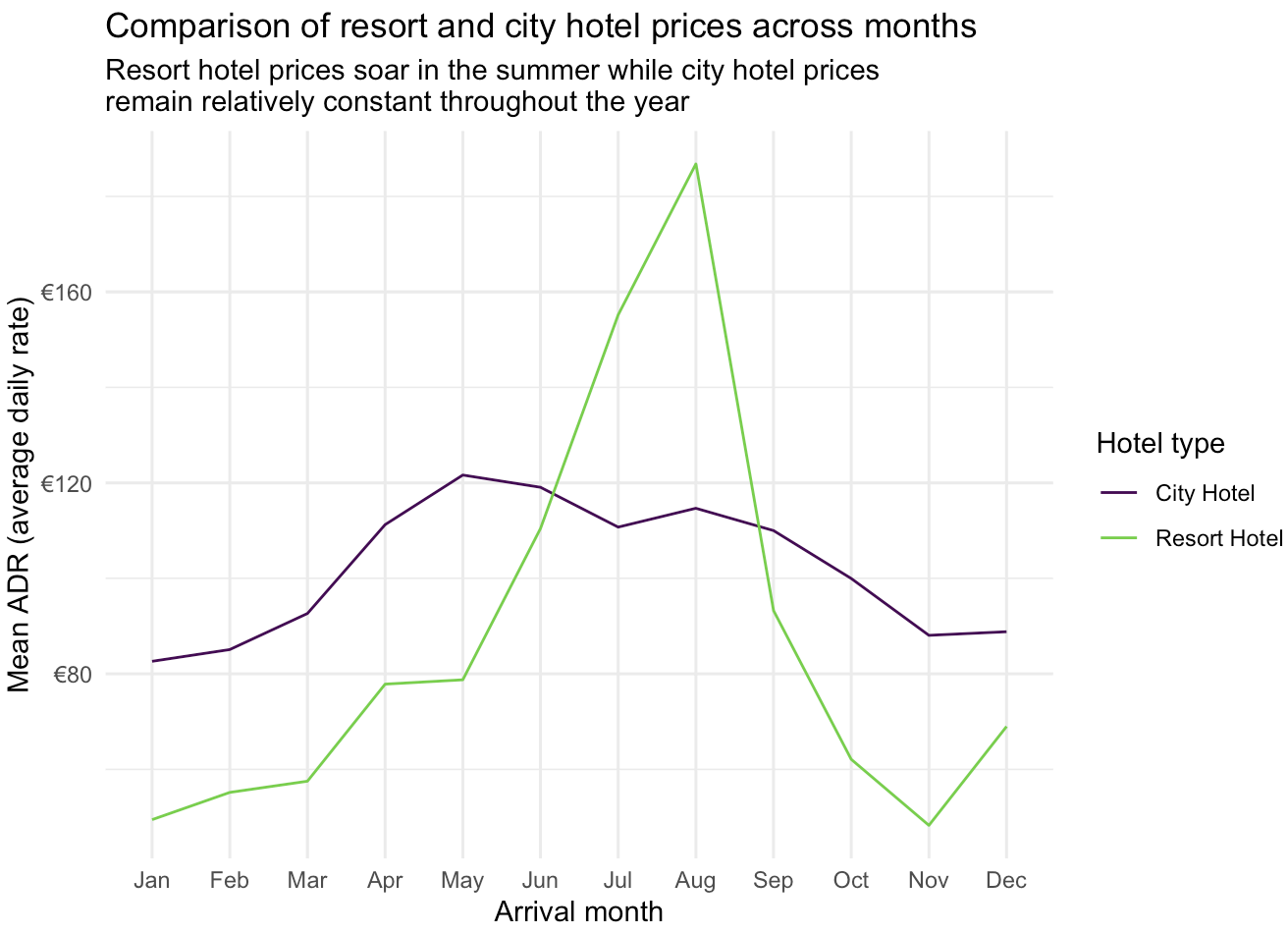
## How does the Average Daily Rate (ADR) change over time? Are there differences between the city and resort hotel?

**Your turn:** Create a visualization that shows the average daily rate (ADR) over time for the city and resort hotels. Calculate the average (mean) ADR for each hotel by month based on when the guest(s) are scheduled to arrive, then visualize using a line graph. Ensure the $x$-axis is ordered chronologically.

> **Note**
>
> Use the **lubridate** package to restructure the data and determine the month when each stay began.

```
hotels |>
  # generate month variable using lubridate
  mutate(
    arrival_date = mdy(arrival_date),
    arrival_date_month = month(arrival_date, label = TRUE),
    .after = arrival_date
  ) |>
  group_by(hotel, arrival_date_month) |>
  summarize(mean_adr = mean(adr), .groups = "drop") |>
  ggplot(mapping = aes(
    x = arrival_date_month,
    y = mean_adr,
    # explicitly use the group aesthetic to ensure correct points are connected
    group = hotel,
    color = hotel
  )) +
  geom_line() +
  scale_y_continuous(labels = label_currency(prefix = "€")) +
  scale_color_viridis_d(end = 0.8) +
  theme_minimal() +
  labs(
    x = "Arrival month",
    y = "Mean ADR (average daily rate)",
    title = "Comparison of resort and city hotel prices across months",
```

```
    subtitle = "Resort hotel prices soar in the summer while city hotel prices\nremain
        relatively constant throughout the year",
    color = "Hotel type"
  )
```




## How often is each meal package booked?

**Your turn:** `meal` reports the type of meal booked with the hotel stay. Categories are presented in standard hospitality meal packages:

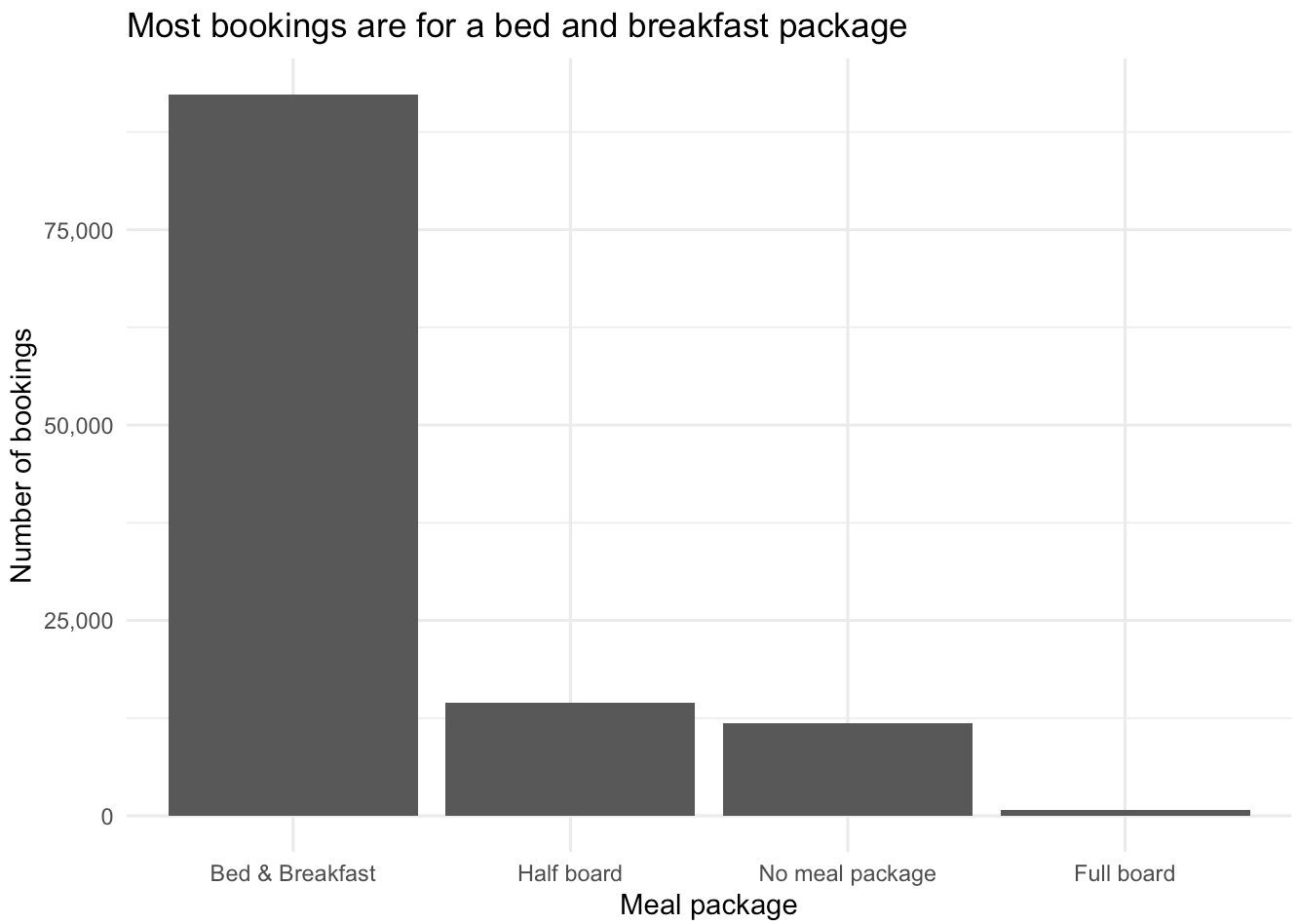
- `Undefined` / `SC` – no meal package
- `BB` – Bed & Breakfast
- `HB` – Half board (breakfast and one other meal – usually dinner)
- `FB` – Full board (breakfast, lunch and dinner)

Create a bar chart reporting the total number of bookings for each meal package. Order the bars by frequency (i.e. most frequent meal package on the left, least frequent meal package on the right).

> **Note**
>
> **forcats** will be your friend in preparing the data for the visualization.

## Create without summarizing the data frame first

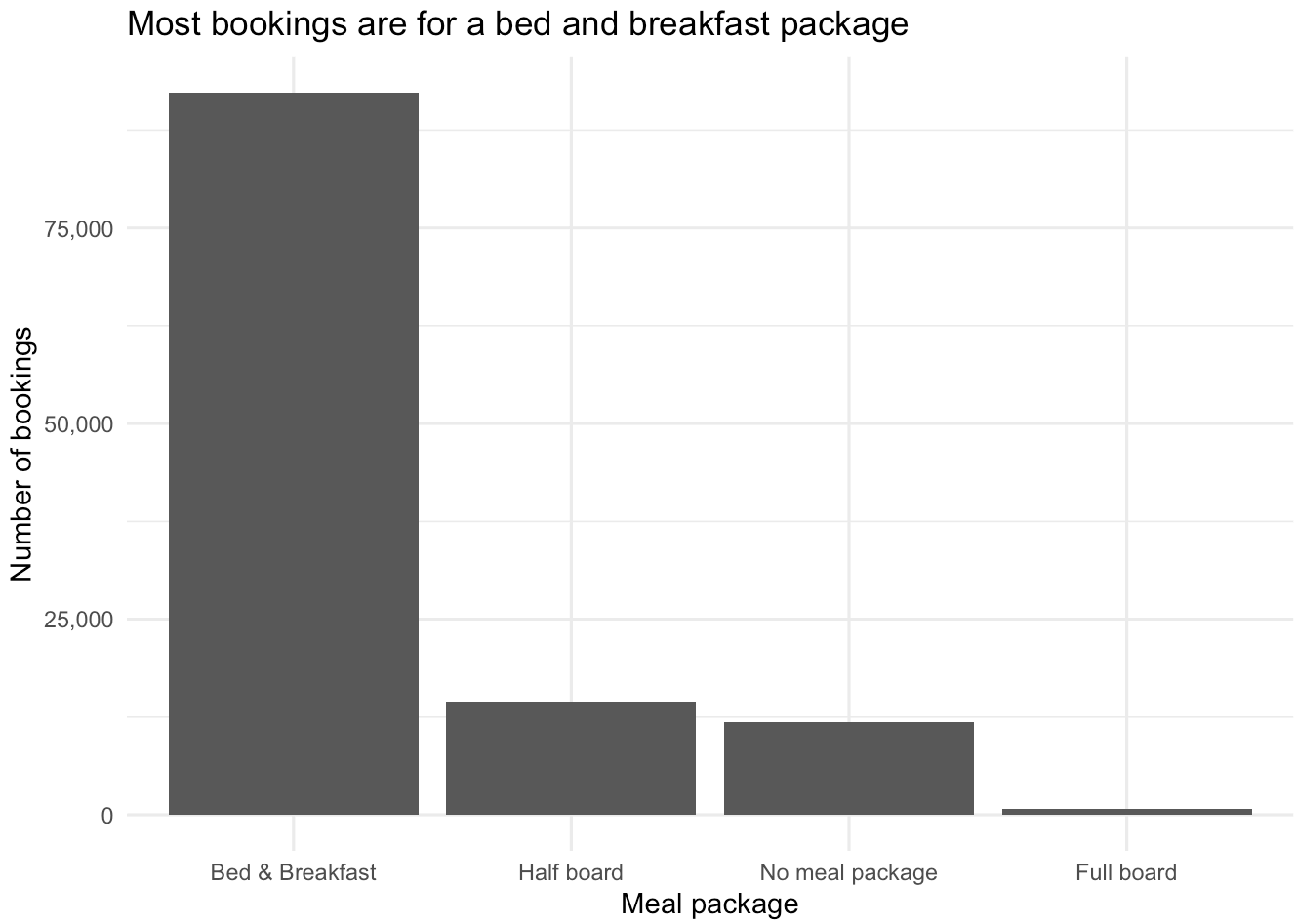
```
hotels |>
  mutate(
    # convert to factor column
    meal = factor(x = meal),
    # recode levels to human-readable, collapsing Undefined and SC simultaneously
    meal = fct_recode(
      .f = meal,
      `No meal package` = "Undefined",
      `No meal package` = "SC",
      `Bed & Breakfast` = "BB",
      `Half board` = "HB",
      `Full board` = "FB"
    ),
    # order by frequency
    meal = fct_infreq(f = meal)
  ) |>
  ggplot(mapping = aes(x = meal)) +
  geom_bar() +
  scale_y_continuous(labels = label_comma()) +
  labs(
    x = "Meal package",
    y = "Number of bookings",
    title = "Most bookings are for a bed and breakfast package"
  ) +
  theme_minimal()
```




## Summarize first, then graph

```
hotels |>
  mutate(
    # convert to factor column
    meal = factor(x = meal),
    # recode levels to human-readable, collapsing Undefined and SC simultaneously
    meal = fct_recode(
      .f = meal,
      `No meal package` = "Undefined",
      `No meal package` = "SC",
      `Bed & Breakfast` = "BB",
      `Half board` = "HB",
      `Full board` = "FB"
    )
  ) |>
  # generate frequency count table
  count(meal) |>
  # reorder meal based on the n column
  # need to reverse the order so it plots correctly
  mutate(meal = fct_reorder(.f = meal, .x = n, .desc = TRUE)) |>
  ggplot(mapping = aes(x = meal, y = n)) +
  geom_col() +
```

```
  scale_y_continuous(labels = label_comma()) +
  labs(
    x = "Meal package",
    y = "Number of bookings",
    title = "Most bookings are for a bed and breakfast package"
  ) +
  theme_minimal()
```




# Acknowledgments

- The first exercise is derived from Data Science in a Box and licensed under CC BY-SA 4.0.

Session information