Chapter 1: open data and perform descriptive analysis

We start by loading the data. We create `listings_clean` and `listings_filtered`. The former selects observations with finite prices and review scores, more than 10 received reviews, and a price below 300 USD; the latter selects listings that accommodate 2 people.

```
# Import the listings.csv file
listings <- read_csv("../data/listings.csv", show_col_types = FALSE)

# Define the columns we want to clean
columns_to_clean <- c("price", "weekly_price", "monthly_price",
                      "security_deposit", "cleaning_fee", "extra_people")

# Identify which of these columns exist in the dataset
existing_columns <- intersect(columns_to_clean, names(listings))

# Clean and convert relevant columns to numeric and filter out non-finite values
listings_clean <- listings %>%
  mutate(across(all_of(existing_columns), ~ as.numeric(gsub("[\\$,]", "", .x)))) %>%
  filter(is.finite(price), is.finite(review_scores_rating), number_of_reviews > 10, price < 300)

# Filter the cleaned data based on accommodates == 2
listings_filtered <- listings_clean %>%
  filter(accommodates == 2)

# Display the first few rows of the cleaned dataset
head(listings_clean)
```

```
## # A tibble: 6 x 75
##        id listing_url            scrape_id last_scraped source name  description
##     <dbl> <chr>                      <dbl> <date>       <chr>  <chr> <chr>
## 1   73155 https://www.airbnb.co~   2.02e13 2024-06-25   city ~ Apar~ This brigh~
## 2   77592 https://www.airbnb.co~   2.02e13 2024-06-25   city ~ Char~ <NA>
## 3  101526 https://www.airbnb.co~   2.02e13 2024-06-25   city ~ Apar~ <NA>
## 4  539905 https://www.airbnb.co~   2.02e13 2024-06-25   city ~ Sunn~ <NA>
## 5  763422 https://www.airbnb.co~   2.02e13 2024-06-25   city ~ City~ A lovely, ~
## 6 1322782 https://www.airbnb.co~   2.02e13 2024-06-25   city ~ Spac~ <NA>
## # i 68 more variables: neighborhood_overview <chr>, picture_url <chr>,
## #   host_id <dbl>, host_url <chr>, host_name <chr>, host_since <date>,
## #   host_location <chr>, host_about <chr>, host_response_time <chr>,
## #   host_response_rate <chr>, host_acceptance_rate <chr>,
## #   host_is_superhost <lgl>, host_thumbnail_url <chr>, host_picture_url <chr>,
## #   host_neighbourhood <chr>, host_listings_count <dbl>,
## #   host_total_listings_count <dbl>, host_verifications <chr>, ...
```

Create some summary statistics.

```
# Create a summary statistics table with one row per variable
summary_stats <- tibble::tibble(
  Variable = c("Price", "Review Scores Rating", "Accommodates"),
```

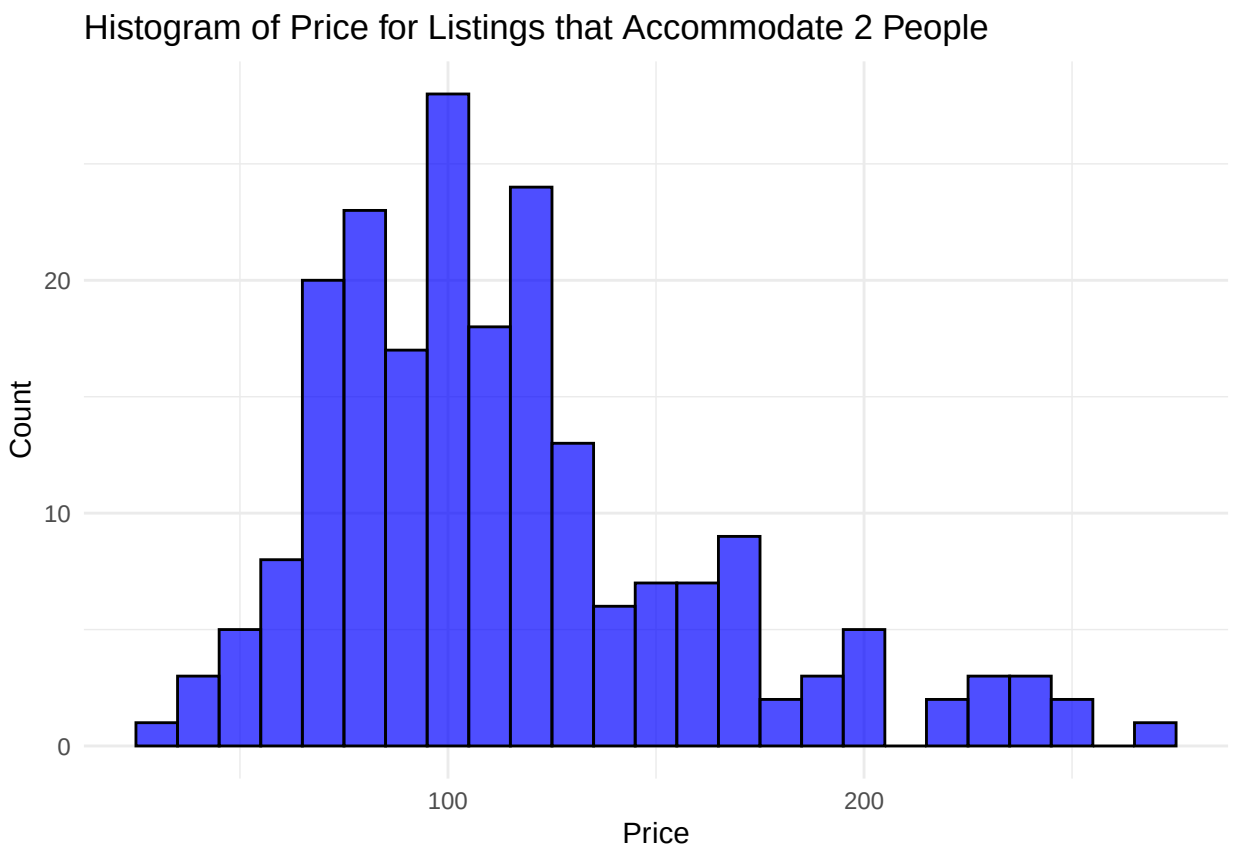
```
  Mean = c(round(mean(listings_clean$price, na.rm = TRUE), 2),
           round(mean(listings_clean$review_scores_rating, na.rm = TRUE), 2),
           round(mean(listings_clean$accommodates, na.rm = TRUE), 2)),
  Median = c(round(median(listings_clean$price, na.rm = TRUE), 2),
             round(median(listings_clean$review_scores_rating, na.rm = TRUE), 2),
             round(median(listings_clean$accommodates, na.rm = TRUE), 2)),
  Standard_Deviation = c(round(sd(listings_clean$price, na.rm = TRUE), 2),
                         round(sd(listings_clean$review_scores_rating, na.rm = TRUE), 2),
                         round(sd(listings_clean$accommodates, na.rm = TRUE), 2)),
  Min = c(round(min(listings_clean$price, na.rm = TRUE), 2),
          round(min(listings_clean$review_scores_rating, na.rm = TRUE), 2),
          round(min(listings_clean$accommodates, na.rm = TRUE), 2)),
  Max = c(round(max(listings_clean$price, na.rm = TRUE), 2),
          round(max(listings_clean$review_scores_rating, na.rm = TRUE), 2),
          round(max(listings_clean$accommodates, na.rm = TRUE), 2))
)

# Display the summary statistics table
summary_stats
```

```
## # A tibble: 3 x 6
##   Variable               Mean Median Standard_Deviation   Min   Max
##   <chr>                 <dbl>  <dbl>              <dbl> <dbl> <dbl>
## 1 Price                130.    123                59.0  28      293
## 2 Review Scores Rating   4.75    4.8                0.21  3.99     5
## 3 Accommodates           3.03    2                  1.87  1       16
```

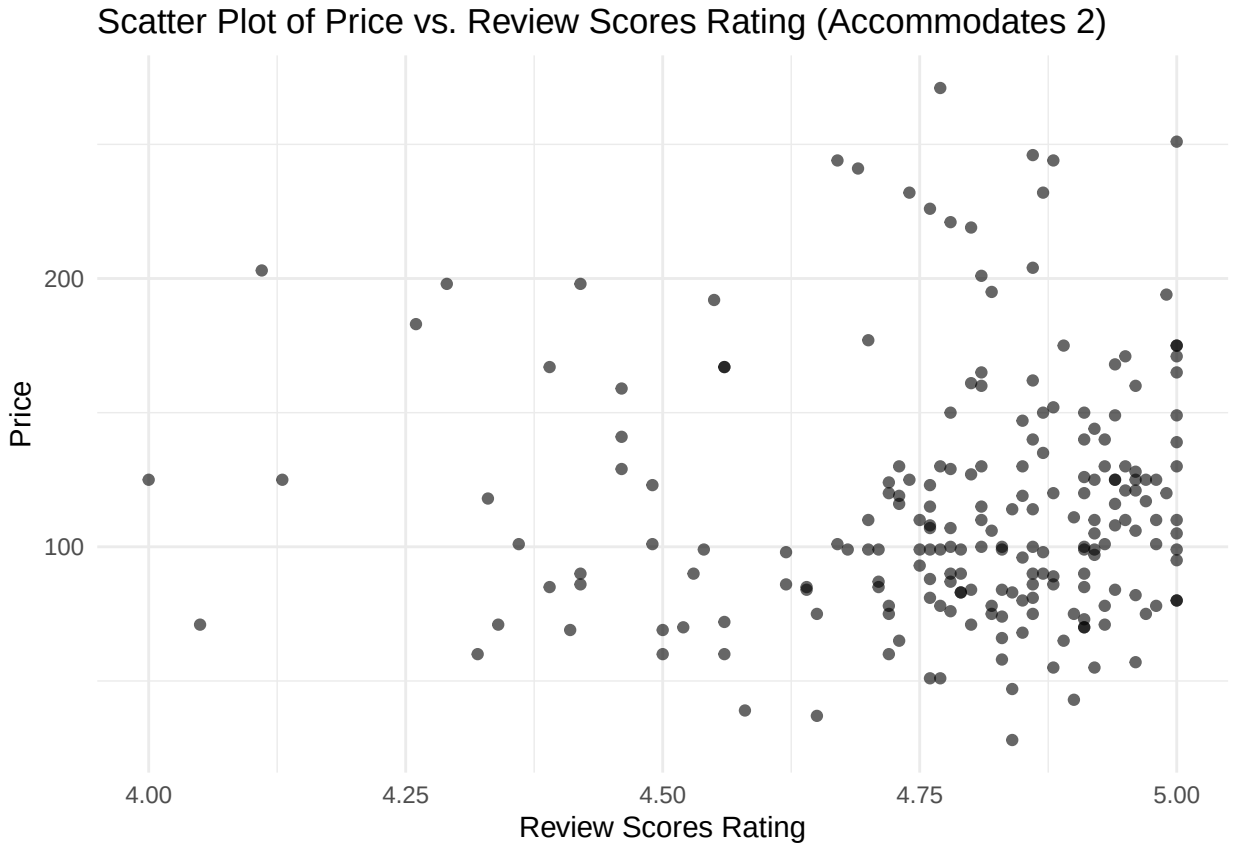
Histogram of `price` for those apartments accommodating 2 people.

```
# Create the histogram
ggplot(listings_filtered, aes(x = price)) +
  geom_histogram(binwidth = 10, fill = "blue", color = "black", alpha = 0.7) +
  labs(
    title = "Histogram of Price for Listings that Accommodate 2 People",
    x = "Price",
    y = "Count"
  ) +
  theme_minimal()
```



What is the empirical relationship between the price and the review score for an apartment that accommodates 2 people? One can look at a scatter plot.

```
# Create the scatter plot
ggplot(listings_filtered, aes(x = review_scores_rating, y = price)) +
  geom_point(alpha = 0.6) +
  labs(
    title = "Scatter Plot of Price vs. Review Scores Rating (Accommodates 2)",
    x = "Review Scores Rating",
    y = "Price"
  ) +
  theme_minimal()
```



Or at a correlation.

```
# Calculate the correlation between price and review_scores_rating
correlation <- cor(listings_filtered$price, listings_filtered$review_scores_rating, use = "complete.obs

# Display the correlation
correlation
```

```
## [1] 0.009672447
```

How does the price for an apartment that accommodates 2 vary by neighborhood?

```
# Create a summary table with the average price for each neighborhood
neighborhood_price_summary <- listings_filtered %>%
  group_by(neighbourhood) %>%
  summarise(average_price = round(mean(price, na.rm = TRUE), 2)) %>%
  arrange(neighbourhood)

# Display the summary table
neighborhood_price_summary
```

```
## # A tibble: 7 x 2
##   neighbourhood                          average_price
##   <chr>                                         <dbl>
## 1 Rotterdam, Netherlands                          121
## 2 Rotterdam, So, Netherlands                      110
## 3 Rotterdam, South Holland, Netherlands            96.4
## 4 Rotterdam, ZH, Netherlands                       94.2
## 5 Rotterdam, Zu, Netherlands                      115
```

```
## 6 Rotterdam, Zuid-Holland, Netherlands          117.
## 7 <NA>                                          118.
```

How does the price vary with the number of people that can stay in an apartment. We wille explore this using a set of so-called box plots.

A box plot, or box-and-whisker plot, visually displays the distribution of data using five key metrics: the minimum, first quartile (Q1), median (Q2), third quartile (Q3), and maximum. The box shows the interquartile range (IQR), representing the middle 50% of the data, with the line inside the box indicating the median. Whiskers extend from the box to the smallest and largest values within 1.5 times the IQR. Outliers, or points outside this range, are shown as individual dots. Box plots are useful for quickly assessing the spread, center, and potential outliers in the data.

We make a figure with one box plot for each value of `accommodates`.

```
# Create box plots of price by accommodates
ggplot(listings_clean, aes(x = factor(accommodates), y = price)) +
  geom_boxplot() +
  labs(
    title = "Box Plot of Price by Accommodates",
    x = "Number of People Accommodated",
    y = "Price"
  ) +
  theme_minimal()
```