Airbnb EDA Paris

Sandy Yu

Invalid Date

Setup

```
library(readr)
library(tidyverse)
library(arrow)
library(ggplot2)
library(naniar)
library(janitor)
library(modelsummary)
```

Download and Save data

```
#save a local copy for Paris Airbnb data
url <-
  paste0(
    "http://data.insideairbnb.com/france/ile-de-france/",
    "paris/2023-12-12/data/listings.csv.gz"
  )

airbnb_data <-
  read_csv(
    file = url,
    guess_max = 20000
  )

write_csv(airbnb_data, "airbnb_data.csv")
```

Created Selected Vairable Dataset

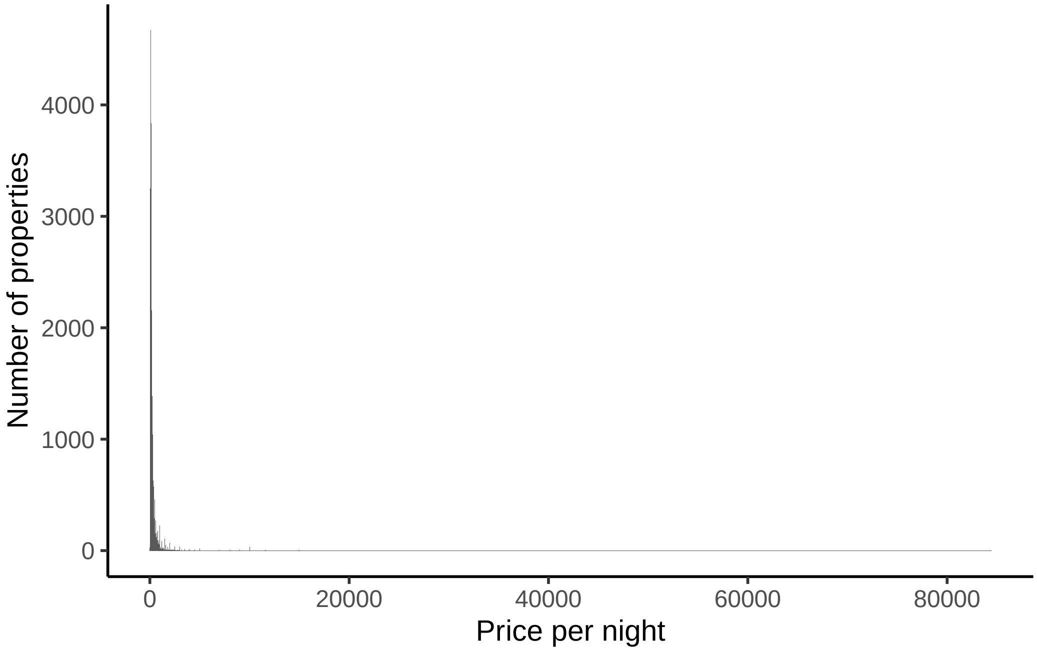
```r
#Created a parquet file for selected Airbnb variables
airbnb_data_selected <-
  airbnb_data |>
  select(
    host_id,
    host_response_time,
    host_is_superhost,
    host_total_listings_count,
    neighbourhood_cleansed,
    bathrooms,
    bedrooms,
    price,
    number_of_reviews,
    review_scores_rating,
    review_scores_accuracy,
    review_scores_value
  )

write_parquet(
  x = airbnb_data_selected,
  sink =
    "2023-12-12-paris-airbnblistings-select_variables.parquet"
  )

rm(airbnb_data)
```

Data Cleaning

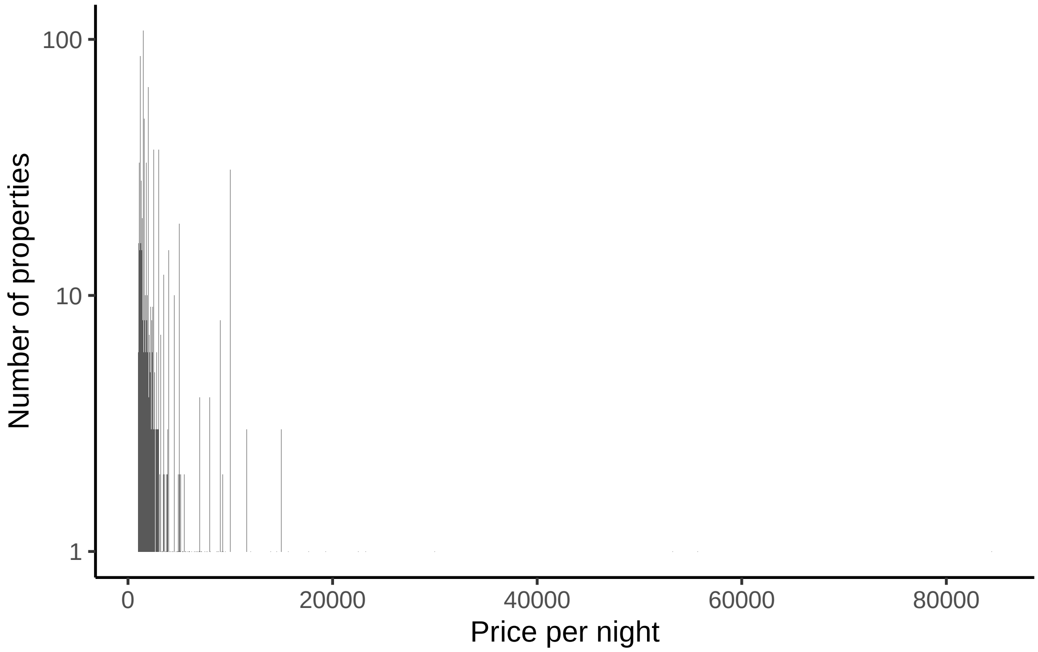
```r
# split and filter out $ signs, and save price values as integers
airbnb_data_selected <-
  airbnb_data_selected |>
  mutate(
    price = str_remove_all(price, "[\\$,]"),
    price = as.integer(price)
  )
```

Graphing

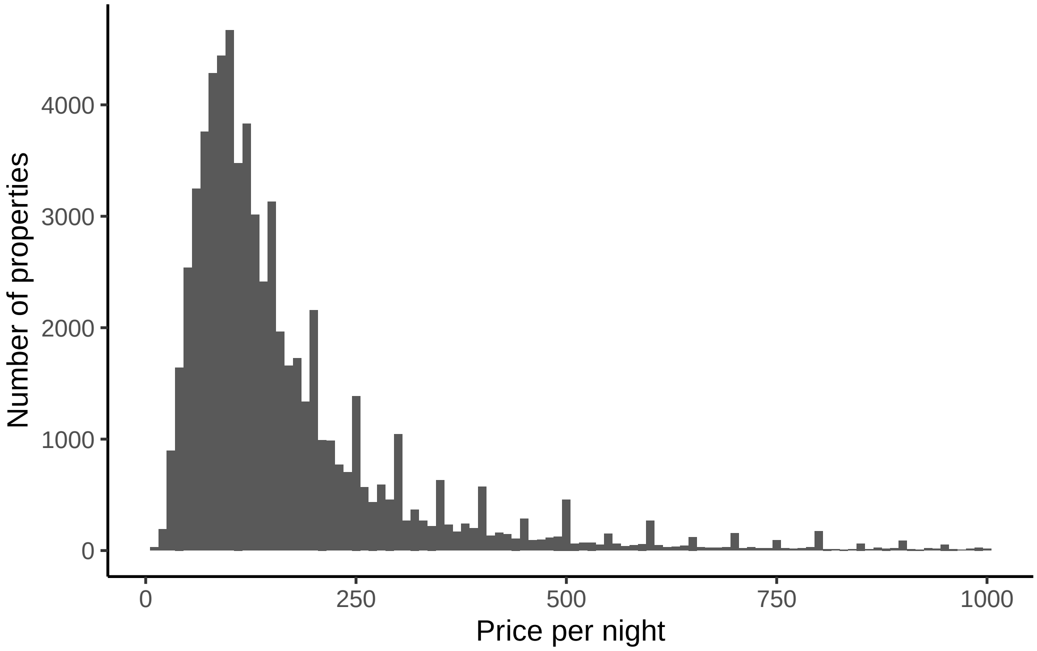
```
airbnb_data_selected |>
  ggplot(aes(x = price)) +
  geom_histogram(binwidth = 10) +
  theme_classic() +
  labs(
    x = "Price per night",
    y = "Number of properties"
  )
```



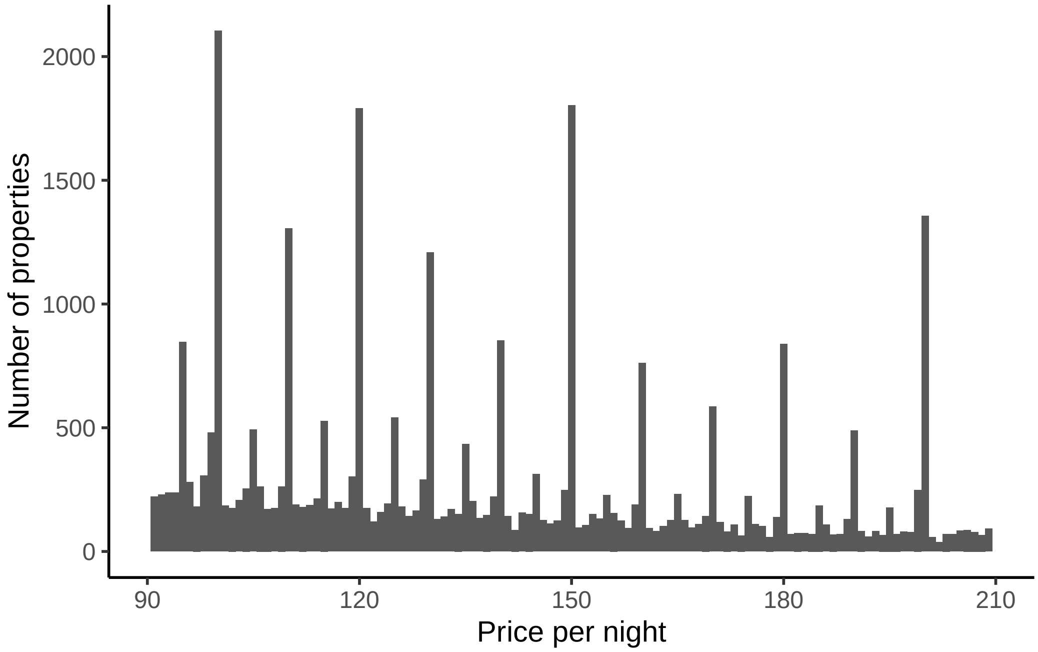
```
#take log to consider give appropraite weight to outliers
airbnb_data_selected |>
  filter(price > 1000) |>
  ggplot(aes(x = price)) +
  geom_histogram(binwidth = 10) +
  theme_classic() +
  labs(
    x = "Price per night",
    y = "Number of properties"
  ) +
  scale_y_log10()
```



```
#focus on price less than $1000
airbnb_data_selected |>
  filter(price < 1000) |>
  ggplot(aes(x = price)) +
  geom_histogram(binwidth = 10) +
  theme_classic() +
  labs(
    x = "Price per night",
    y = "Number of properties"
  )
```

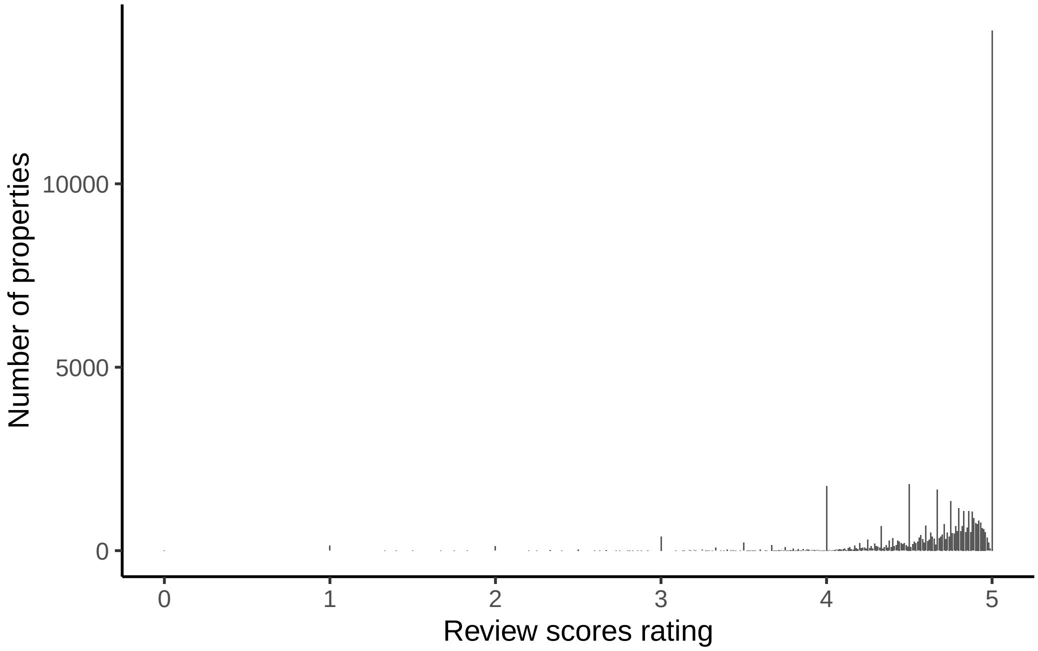


```
#reduce bin size and zoom in on prices between $90 - $120
airbnb_data_selected |>
  filter(price > 90) |>
  filter(price < 210) |>
  ggplot(aes(x = price)) +
  geom_histogram(binwidth = 1) +
  theme_classic() +
  labs(
    x = "Price per night",
    y = "Number of properties"
  )
```



Superhosts under $1000

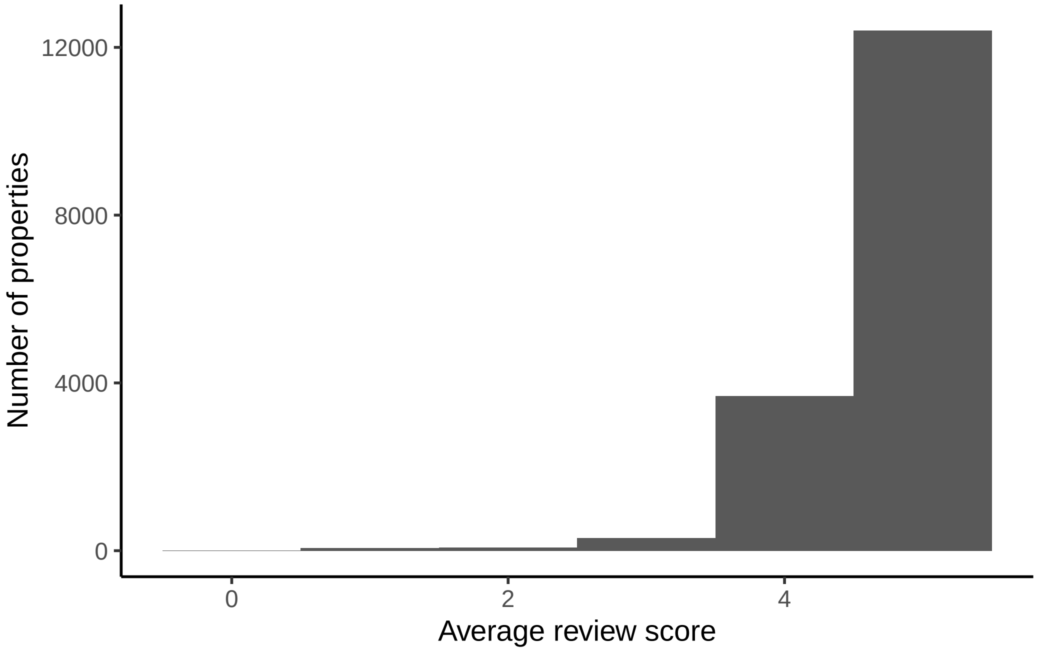
```
#remove all prices above 999
airbnb_data_less_1000 <-
  airbnb_data_selected |>
  filter(price < 1000)
#remove NAs for superhosts
airbnb_data_no_superhost_nas <-
  airbnb_data_less_1000 |>
  filter(!is.na(host_is_superhost)) |>
  mutate(
    host_is_superhost_binary =
      as.numeric(host_is_superhost)
  )

#graph
airbnb_data_no_superhost_nas |>
  ggplot(aes(x = review_scores_rating)) +
  geom_bar() +
  theme_classic() +
  labs(
    x = "Review scores rating",
    y = "Number of properties"
  )
```



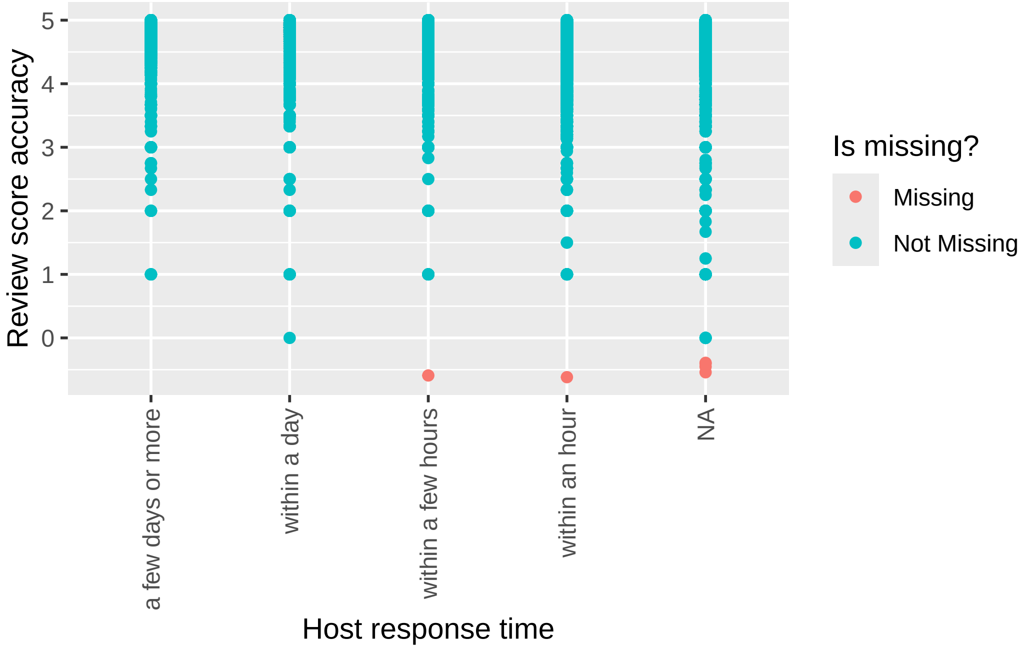
Reviews Distribution

```
#setup review dataset
airbnb_data_has_reviews <-
  airbnb_data_no_superhost_nas |>
  filter(!is.na(review_scores_rating))
airbnb_data_has_reviews <-
  airbnb_data_has_reviews |>
  mutate(
    host_response_time = if_else(
      host_response_time == "N/A",
      NA_character_,
      host_response_time
    ),
    host_response_time = factor(host_response_time)
  )
```

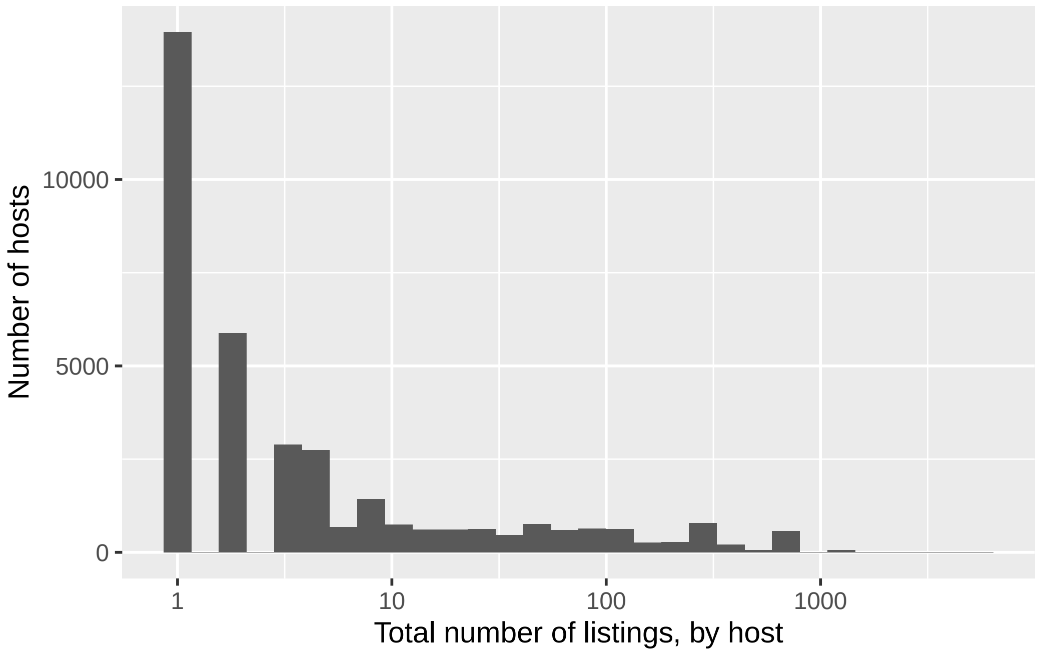
```r
#review socres distribution
airbnb_data_has_reviews <-
  airbnb_data_has_reviews |>
  mutate(
    host_response_time = if_else(
      host_response_time == "N/A",
      NA_character_,
      host_response_time
    ),
    host_response_time = factor(host_response_time)
  )
#graph
airbnb_data_has_reviews |>
  filter(is.na(host_response_time)) |>
  ggplot(aes(x = review_scores_rating)) +
  geom_histogram(binwidth = 1) +
  theme_classic() +
  labs(
    x = "Average review score",
    y = "Number of properties"
  )
```



```r
#including missing (NA) review values
airbnb_data_has_reviews |>
  ggplot(aes(
    x = host_response_time,
    y = review_scores_accuracy
  )) +
  geom_miss_point() +
  labs(
    x = "Host response time",
    y = "Review score accuracy",
    color = "Is missing?"
  ) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

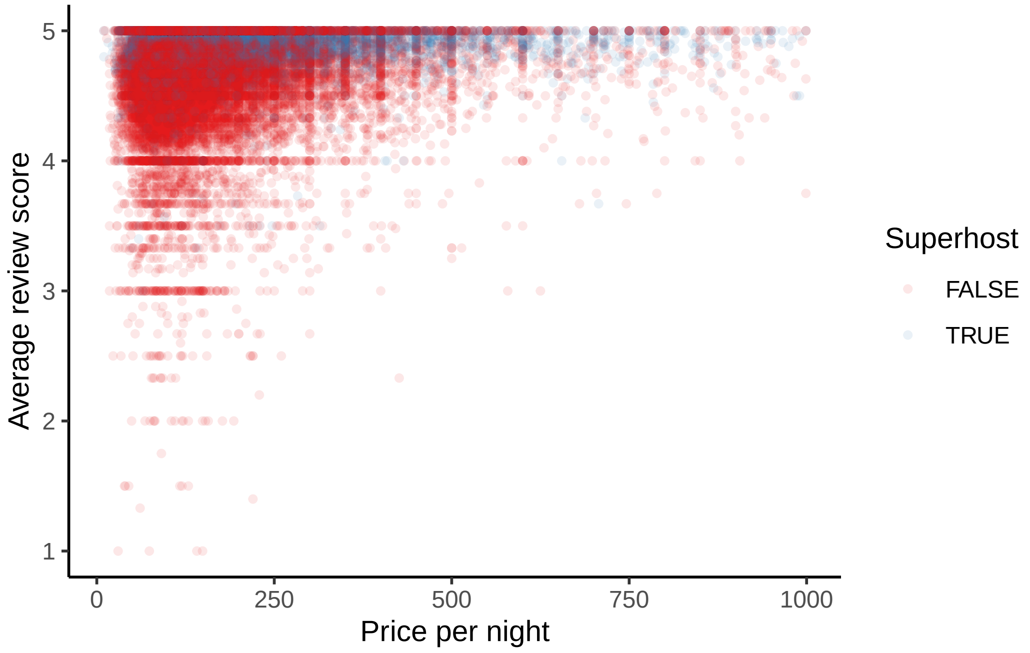


```r
#remove NA review rows
airbnb_data_selected <-
  airbnb_data_has_reviews |>
  filter(!is.na(host_response_time))

#graph
airbnb_data_selected |>
  ggplot(aes(x = host_total_listings_count)) +
  geom_histogram() +
  scale_x_log10() +
  labs(
    x = "Total number of listings, by host",
    y = "Number of hosts"
  )
```



Regressions

```
# Relationship between price and review and whether a host is a superhost
airbnb_data_selected |>
  filter(number_of_reviews > 1) |>
  ggplot(aes(x = price, y = review_scores_rating,
             color = host_is_superhost)) +
  geom_point(size = 1, alpha = 0.1) +
  theme_classic() +
  labs(
    x = "Price per night",
    y = "Average review score",
    color = "Superhost"
  ) +
  scale_color_brewer(palette = "Set1")
```



```
#relationship between Superhost status and Response time
airbnb_data_selected |>
  count(host_is_superhost) |>
  mutate(
    proportion = n / sum(n),
    proportion = round(proportion, digits = 2)
  )
```

```
# A tibble: 2 x 3
  host_is_superhost     n proportion
  <lgl>             <int>      <dbl>
1 FALSE             25557       0.72
2 TRUE               9888       0.28
```

```
airbnb_data_selected |>
  tabyl(host_response_time, host_is_superhost) |>
  adorn_percentages("col") |>
  adorn_pct_formatting(digits = 0) |>
  adorn_ns() |>
  adorn_title()
```

```
                    host_is_superhost
```

```
host_response_time             FALSE        TRUE
a few days or more       5%  (1,219)  0%    (24)
      within a day      17%  (4,326) 10%   (971)
within a few hours      18%  (4,660) 22% (2,151)
    within an hour      60% (15,352) 68% (6,742)
```

```
# take detail loo at demographic variable neighborhood
airbnb_data_selected |>
  tabyl(neighbourhood_cleansed) |>
  adorn_pct_formatting() |>
  arrange(-n) |>
  filter(n > 100) |>
  adorn_totals("row") |>
  head()
```

```
neighbourhood_cleansed    n percent
     Buttes-Montmartre 3737   10.5%
            Popincourt 3076    8.7%
             Vaugirard 2587    7.3%
              Entrepôt 2552    7.2%
    Batignolles-Monceau 2197    6.2%
        Buttes-Chaumont 1895    5.3%
```

```
#logistic regression to forecast probability of Superhost based on Response time and Review
logistic_reg_superhost_response_review <-
  glm(
    host_is_superhost ~
      host_response_time +
      review_scores_rating,
    data = airbnb_data_selected,
    family = binomial
  )
#summary of forecasting model
modelsummary(logistic_reg_superhost_response_review)
```

Save

	(1)
(Intercept)	−18.384
	(0.377)
host__response__timewithin a day	2.283
	(0.210)
host__response__timewithin a few hours	3.015
	(0.209)
host__response__timewithin an hour	3.190
	(0.208)
review__scores__rating	3.021
	(0.065)
Num.Obs.	35 445
AIC	37 601.0
BIC	37 643.4
Log.Lik.	−18 795.504
RMSE	0.43

```
write_parquet(
  x = airbnb_data_selected,
  sink = "2023-12-12-paris-airbnblistings-analysis_dataset.parquet"
  )
```