Airbnb Analysis of the Least and Most Expensive Cities to Live In.

Frederick Jones

2023-10-08

Libraries

```
library(tidyverse)
library(here)
library(rstatix)
library(gt)
```

Data Import

Listing Prices Dataset

```
listing_price= read_csv(here('airbnb_listing_price.csv'))
listing_price |> head()
```

```
## # A tibble: 6 x 4
##   listing_id price minimum_nights maximum_nights
##        <dbl> <dbl>          <dbl>          <dbl>
## 1     281420    53              2           1125
## 2    3705183   120              2           1125
## 3    4082273    89              2           1125
## 4    4797344    58              2           1125
## 5    4823489    60              2           1125
## 6    4898654    95              2           1125
```

```
dim(listing_price)
```

```
## [1] 279712      4
```

Location Information Dataset

```
location_info=read_csv(here('airbnb_location_info.csv'))
location_info |> head()
```

```
## # A tibble: 6 x 7
##   listing_id host_location       neighbourhood district city  latitude longitude
##        <dbl> <chr>               <chr>         <chr>    <chr>    <dbl>     <dbl>
## 1     281420 Paris, Ile-de-Fran~ Buttes-Montm~ <NA>     Paris     48.9      2.33
## 2    3705183 Paris, Ile-de-Fran~ Buttes-Montm~ <NA>     Paris     48.9      2.35
## 3    4082273 Paris, Ile-de-Fran~ Elysee        <NA>     Paris     48.9      2.32
## 4    4797344 Paris, Ile-de-Fran~ Vaugirard     <NA>     Paris     48.8      2.31
## 5    4823489 Paris, Ile-de-Fran~ Passy         <NA>     Paris     48.9      2.27
## 6    4898654 Paris, Ile-de-Fran~ Temple        <NA>     Paris     48.9      2.35
```

```
dim(location_info)
```

```
## [1] 279712      7
```

Property Information Dataset

```
property_info=read_csv(here('airbnb_property_info.csv'),locale = locale(encoding = 'utf8'))
property_info |> head()
```

```
## # A tibble: 6 x 8
##   listing_id name        property_type room_type accommodates bedrooms amenities
##        <dbl> <chr>       <chr>         <chr>            <dbl>    <dbl> <chr>
## 1     281420 Beautiful ~ Entire apart~ Entire p~            2        1 "[\"Heat~
## 2    3705183 39 mÃ‚Â² P~ Entire apart~ Entire p~            2        1 "[\"Sham~
## 3    4082273 Lovely apa~ Entire apart~ Entire p~            2        1 "[\"Heat~
## 4    4797344 Cosy studi~ Entire apart~ Entire p~            2        1 "[\"Heat~
## 5    4823489 Close to E~ Entire apart~ Entire p~            2        1 "[\"Heat~
## 6    4898654 NEW - Char~ Entire apart~ Entire p~            2        1 "[\"Heat~
## # i 1 more variable: instant_bookable <lgl>
```

```
property_info |> dim()
```

```
## [1] 279712      8
```

Data Cleaning

In which location is the host situated?

```
location_info$country= location_info$host_location |> str_split_i(',',-1)  |>  str_trim()
```

Propritye Info

I will exclude the names and amenities columns from the dataset. The column `names` doesn't provide significant information regarding pricing, and the `amenities` column may have a variable number of different values, making it challenging to use in the analysis.

```
property_info|> head()
```

```
## # A tibble: 6 x 8
##   listing_id name        property_type room_type accommodates bedrooms amenities
##        <dbl> <chr>       <chr>         <chr>            <dbl>    <dbl> <chr>
## 1     281420 Beautiful ~ Entire apart~ Entire p~            2        1 "[\"Heat~
## 2    3705183 39 mÃ‚Â² P~ Entire apart~ Entire p~            2        1 "[\"Sham~
## 3    4082273 Lovely apa~ Entire apart~ Entire p~            2        1 "[\"Heat~
## 4    4797344 Cosy studi~ Entire apart~ Entire p~            2        1 "[\"Heat~
## 5    4823489 Close to E~ Entire apart~ Entire p~            2        1 "[\"Heat~
## 6    4898654 NEW - Char~ Entire apart~ Entire p~            2        1 "[\"Heat~
## # i 1 more variable: instant_bookable <lgl>
```

```
property_info =property_info|> select(-c('name','amenities'))
property_info |> head()
```

```
## # A tibble: 6 x 6
##   listing_id property_type    room_type   accommodates bedrooms instant_bookable
##        <dbl> <chr>            <chr>              <dbl>    <dbl> <lgl>
## 1     281420 Entire apartment Entire pla~            2        1 FALSE
## 2    3705183 Entire apartment Entire pla~            2        1 FALSE
## 3    4082273 Entire apartment Entire pla~            2        1 FALSE
## 4    4797344 Entire apartment Entire pla~            2        1 FALSE
## 5    4823489 Entire apartment Entire pla~            2        1 FALSE
## 6    4898654 Entire apartment Entire pla~            2        1 FALSE
```

Join datasets

All datasets share the same number of rows and a common column, listing_id. Therefore, we only need to merge the columns. To avoid errors, I will use a left join approach with listing_price as the main dataset because it contains the primary information. If a listing is not present in listing_price, it will not be included in the final dataset.

```
property_listing_price = left_join(listing_price,property_info)
property_listing_price |> head()
```

```
## # A tibble: 6 x 9
##   listing_id price minimum_nights maximum_nights property_type    room_type
##        <dbl> <dbl>          <dbl>          <dbl> <chr>            <chr>
## 1     281420    53              2           1125 Entire apartment Entire place
## 2    3705183   120              2           1125 Entire apartment Entire place
## 3    4082273    89              2           1125 Entire apartment Entire place
## 4    4797344    58              2           1125 Entire apartment Entire place
## 5    4823489    60              2           1125 Entire apartment Entire place
## 6    4898654    95              2           1125 Entire apartment Entire place
## # i 3 more variables: accommodates <dbl>, bedrooms <dbl>,
## #   instant_bookable <lgl>
```

```r
df=left_join(property_listing_price,location_info)
df |> dim()
```

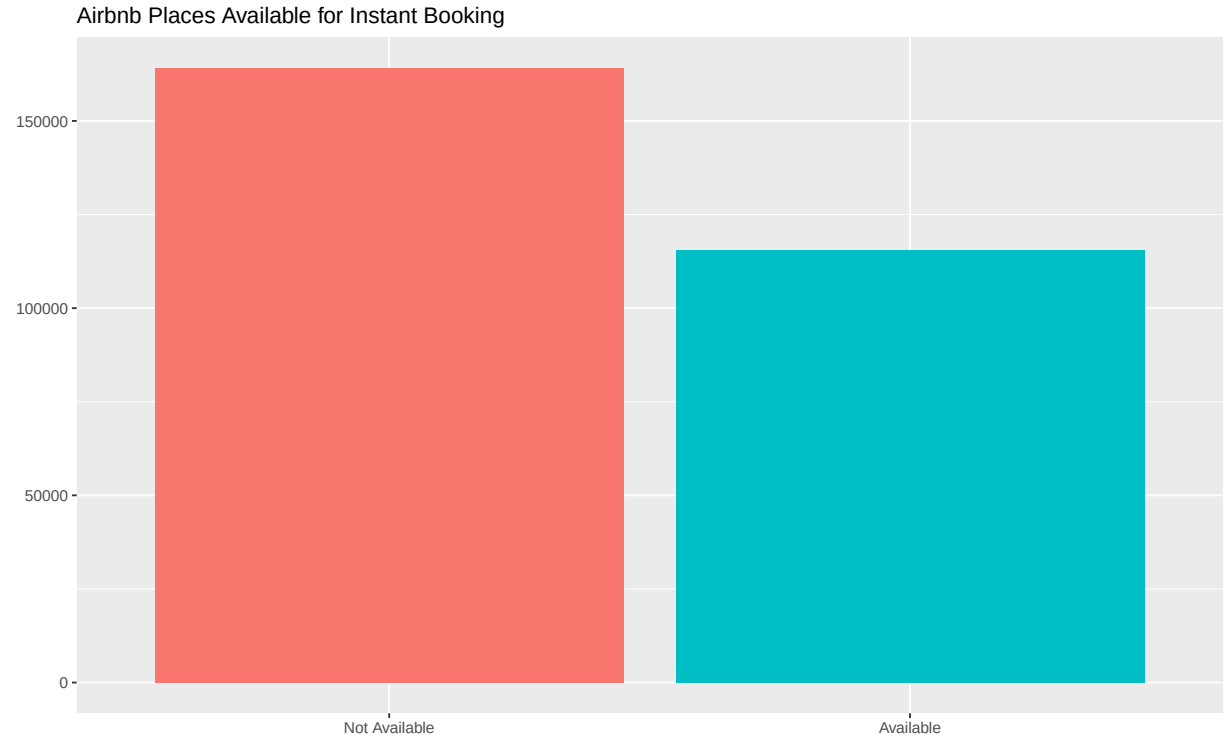
```
## [1] 279712     16
```

```r
#write.csv(df,here('ainb_full.csv'),row.names = FALSE)
```

Data Analysis

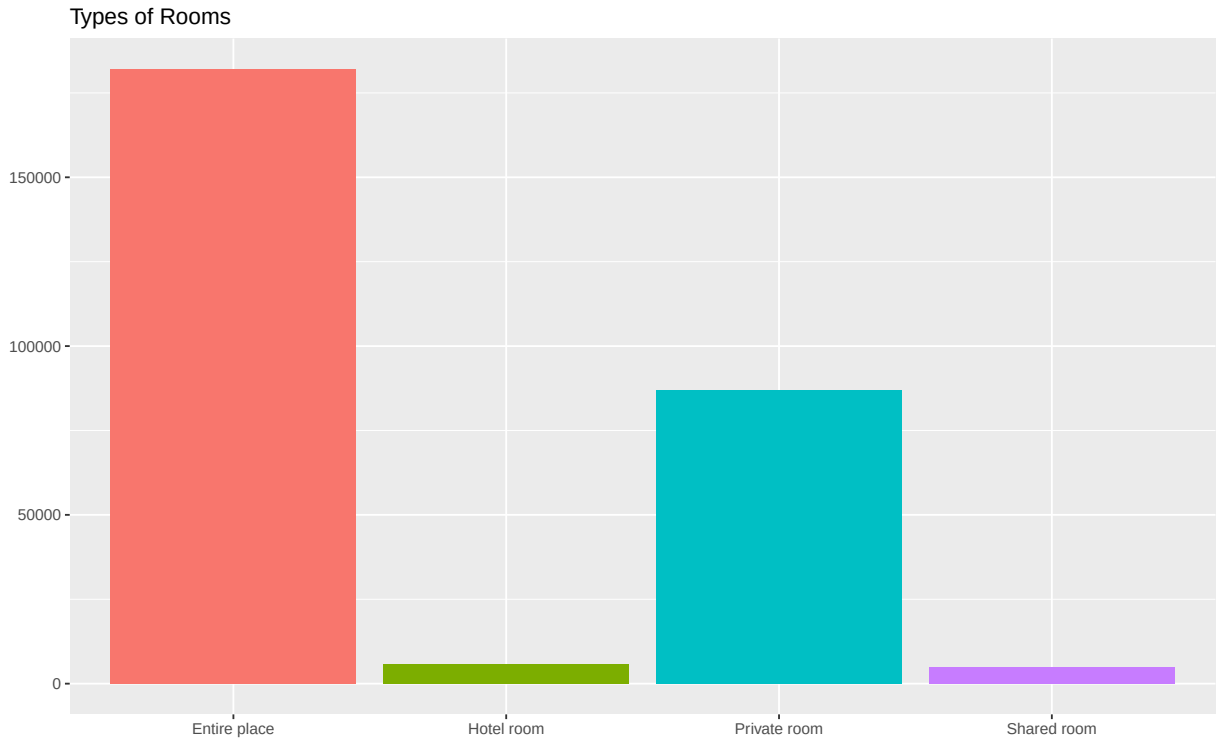
How many places are available for Instant Bookable?

```r
df$instant_bookable =as.factor(df$instant_bookable)
levels(df$instant_bookable) <- c('Not Available', 'Available')
df |> group_by(instant_bookable=df$instant_bookable) |>
  summarise(n=n()) |>
  ggplot() +
  geom_col(aes(x=instant_bookable,y=n, fill=instant_bookable))+
  labs(title = 'Airbnb Places Available for Instant Booking',
       x="",
       y="")  +
  theme(
    legend.position = 'none')
```



How many room types?

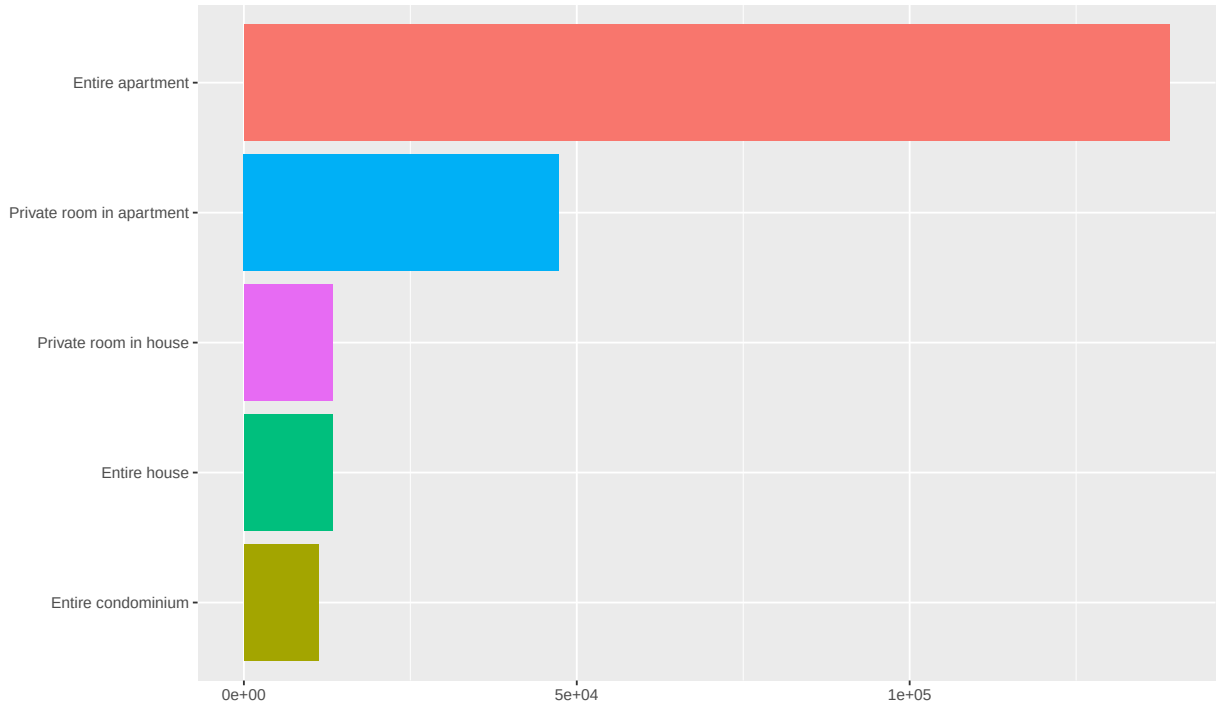
```
df |> group_by(room_type) |>
  summarise(n=n()) |>
  ggplot() +
  geom_col(aes(x=room_type,y=n, fill=room_type))+
  labs(title = 'Types of Rooms',
       x="",
       y="")  +
  theme(
    legend.position = 'none')
```



The five most common property types on Airbnb.

```
df |> group_by(property_type) |>
  summarise(n=n()) |> arrange(desc(n)) |>
  top_n(5) |>
  ggplot() +
  geom_col(aes(x=reorder(property_type,n),y=n, fill=property_type))+
  labs(title = 'The five most common property types on Airbnb.',
       x="",
       y="")  +
  theme(
    legend.position = 'none') +
  coord_flip()
```

The five most common property types on Airbnb.

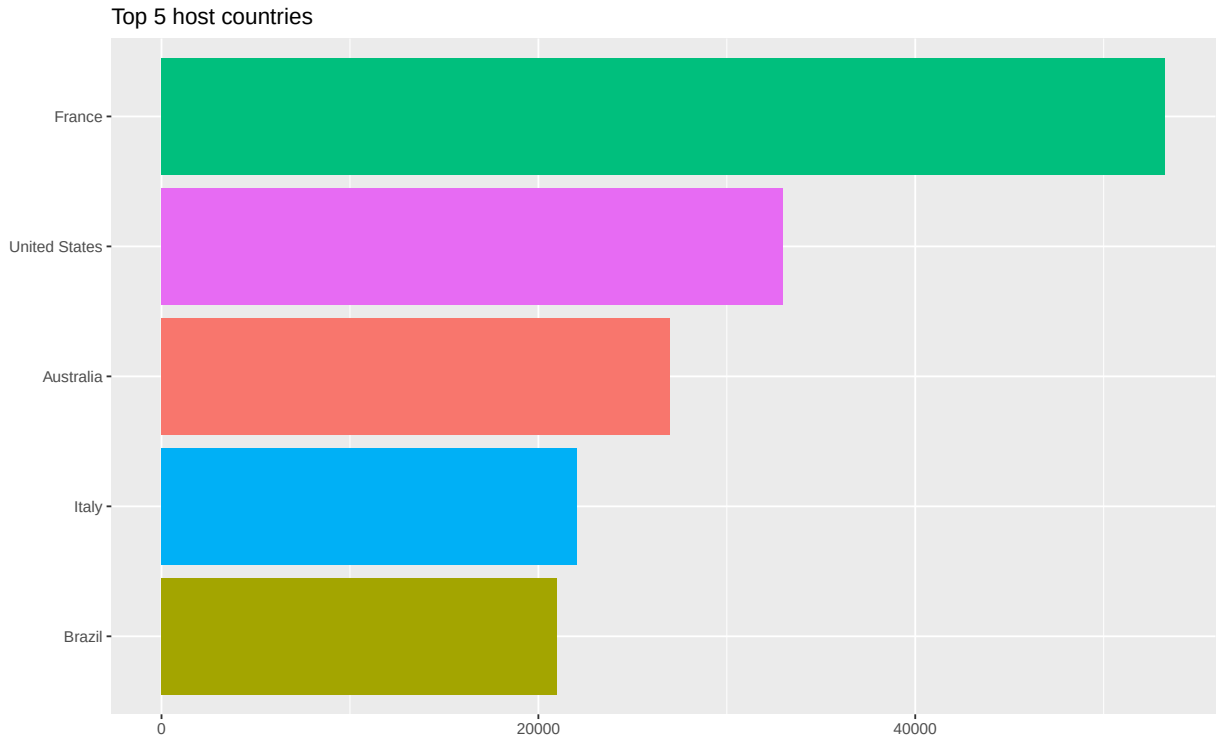


Top 5 host countries.

The host is the person who makes a place available for rent, but does not necessarily live there. Which country has the highest number of hosts?

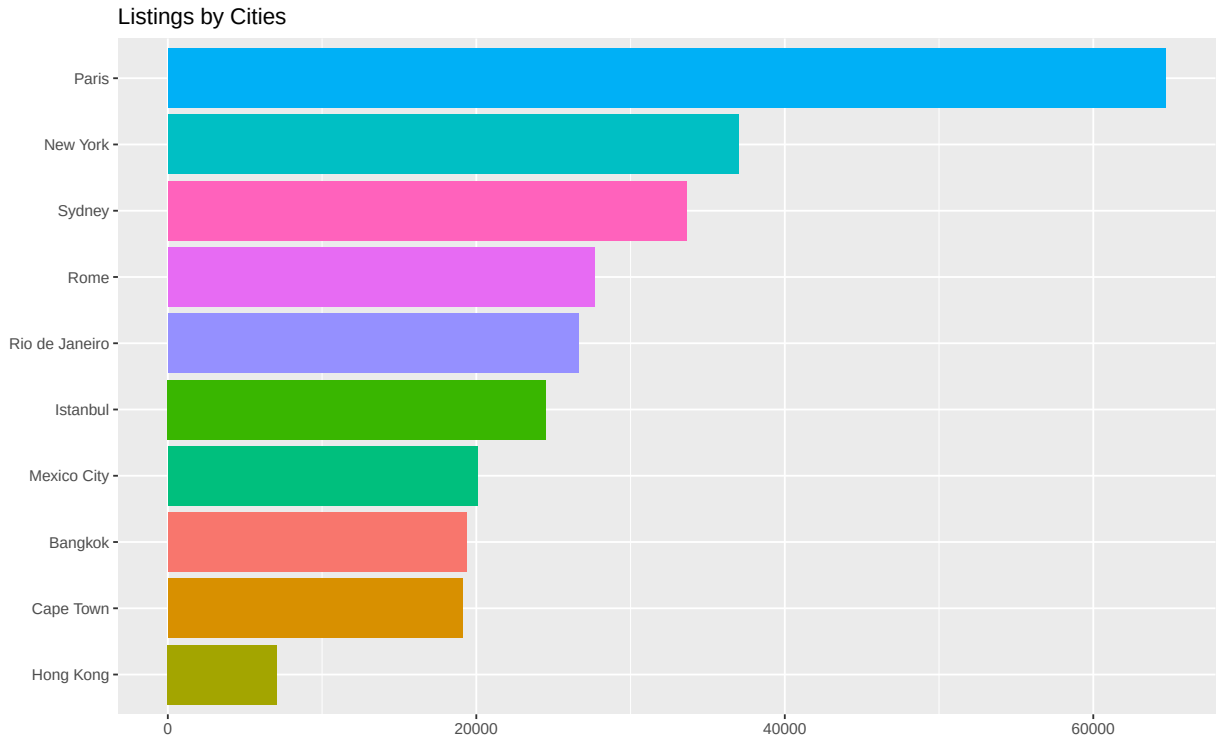
```
top_countries= df |> group_by(country) |>
  summarise(n=n()) |> arrange(desc(n)) |>top_n(5)
```

```
top_countries |>
  ggplot() +
  geom_col(aes(x=reorder(country,n),y=n, fill=country))+
  labs(title = 'Top 5 host countries',
       x="",
       y="")  +
  theme(
    legend.position = 'none') +
  coord_flip()
```



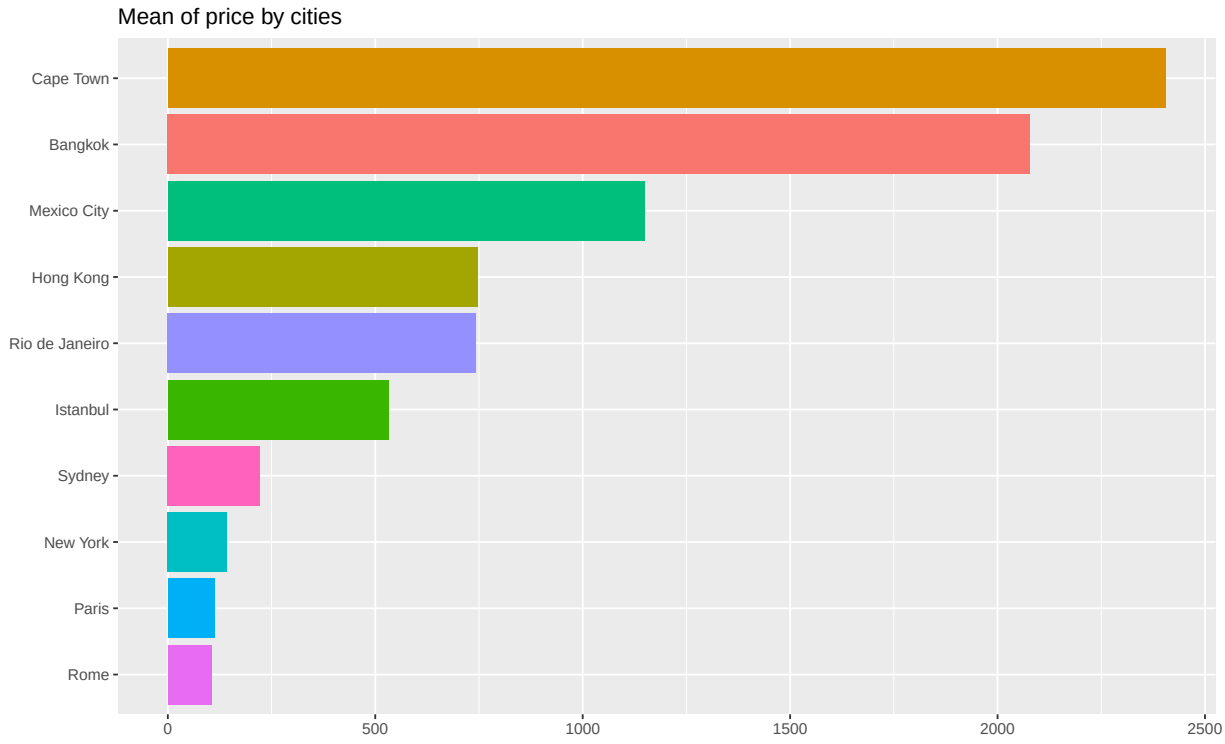
Listings by Cities

```
df |> group_by(city) |>
  summarise(n=n()) |> arrange(desc(n)) |>
  ggplot() +
  geom_col(aes(x=reorder(city,n),y=n, fill=city))+
  labs(title = 'Listings by Cities',
       x="",
       y="")  +
  theme(
    legend.position = 'none') +
  coord_flip()
```

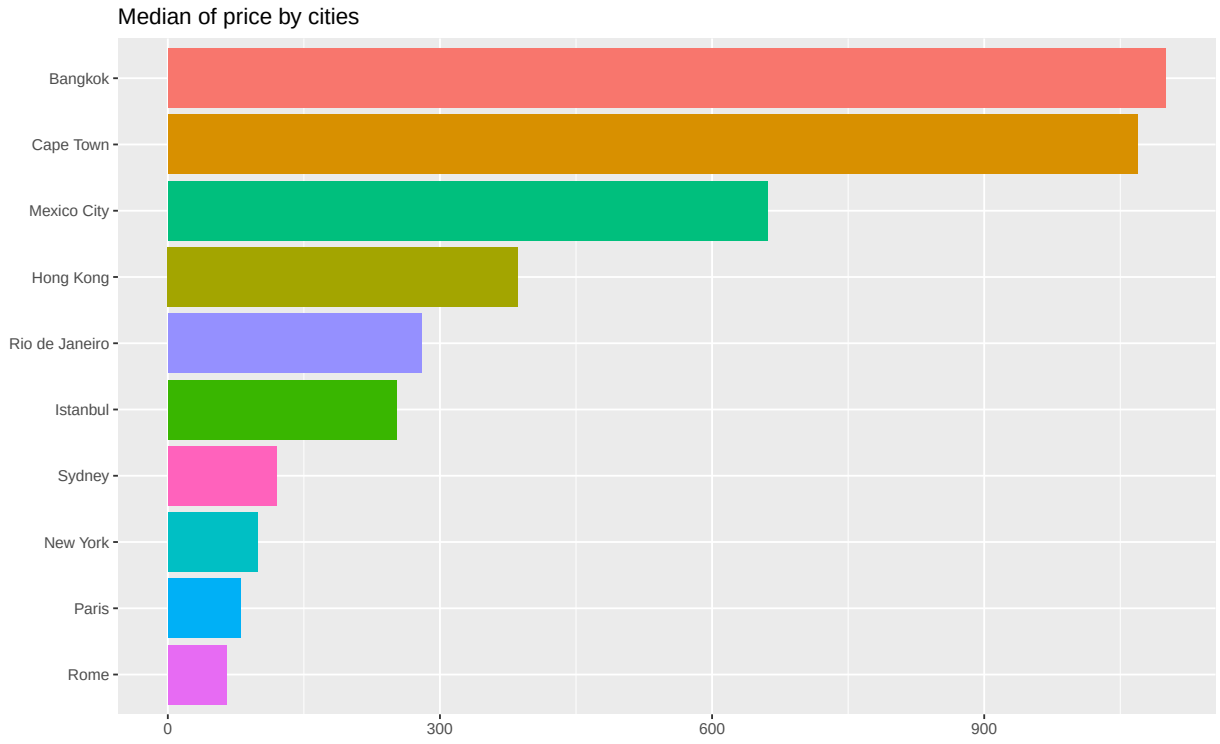


Prices rent by Cities

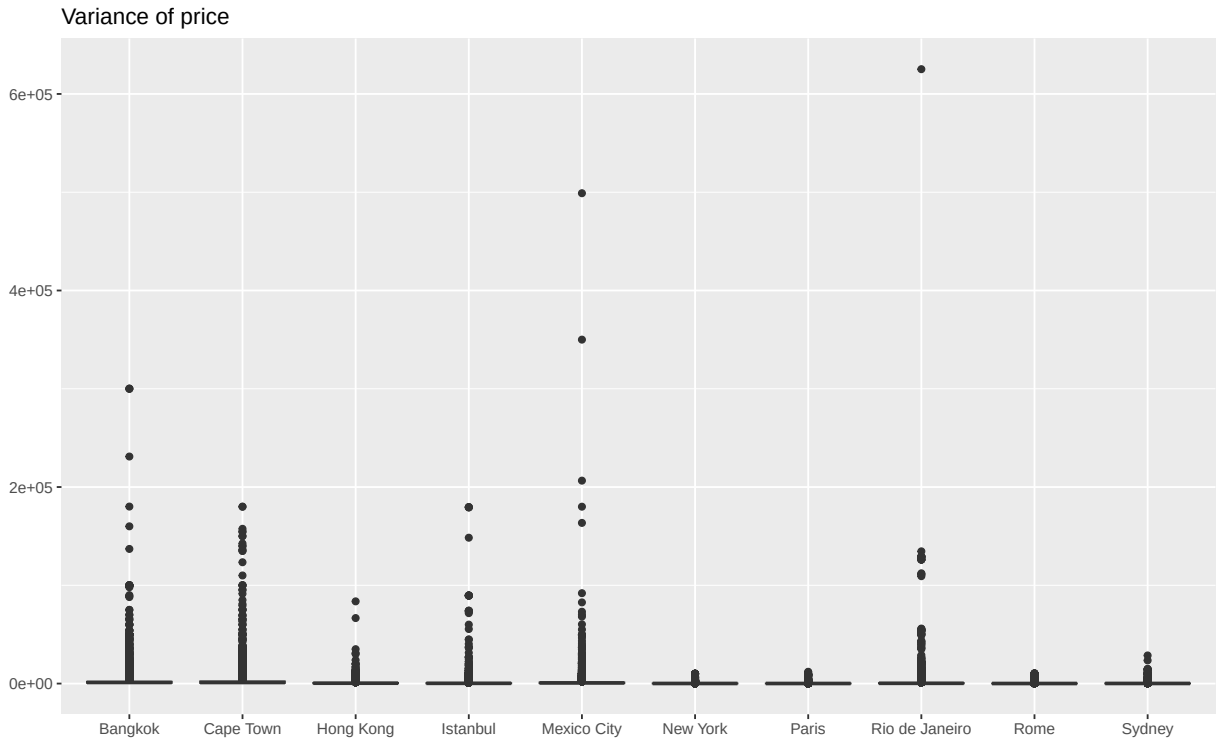
```
df|>  group_by(city) |>
  summarise(mean_price=mean(price)) |> arrange(desc(mean_price)) |>
  ggplot() +
  geom_col(aes(x=reorder(city,mean_price),y=mean_price, fill=city))+
  labs(title = 'Mean of price by cities',
       x="",
       y="")  +
  theme(
    legend.position = 'none') +coord_flip()
```



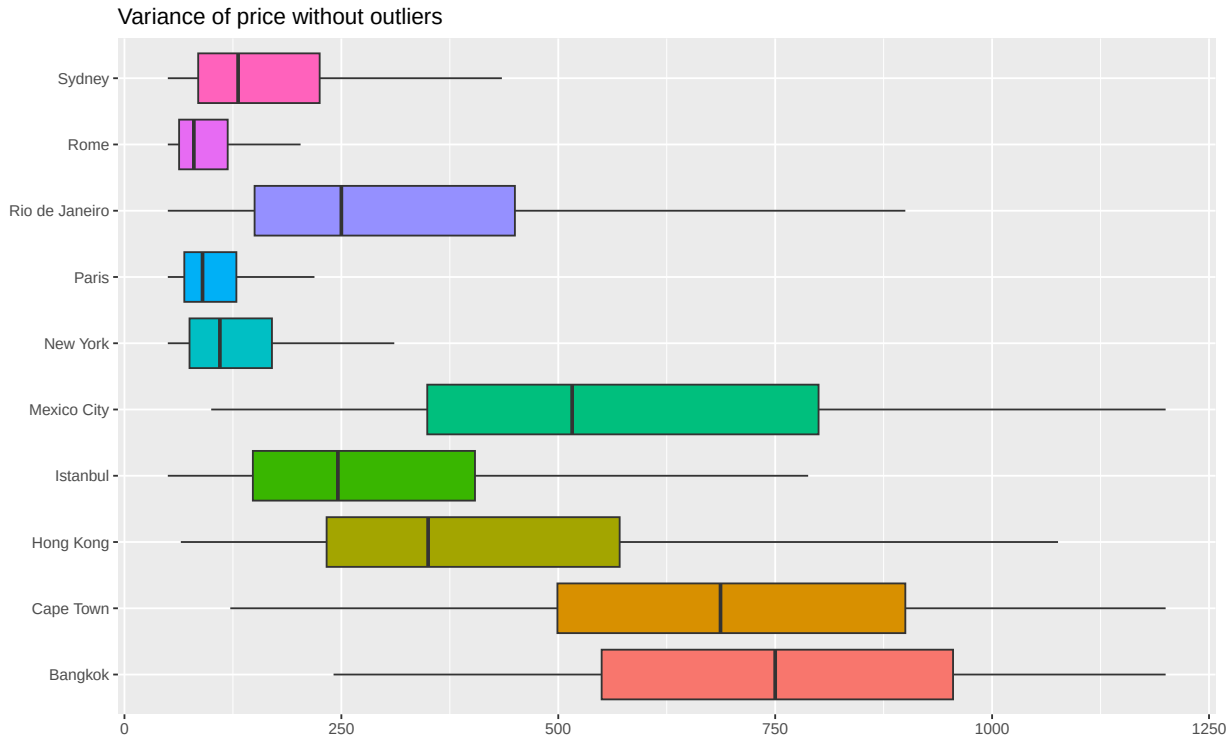

```
df|>  group_by(city) |>
  summarise(mean_price=median(price)) |> arrange(desc(mean_price)) |>
  ggplot() +
  geom_col(aes(x=reorder(city,mean_price),y=mean_price, fill=city))+
  labs(title = 'Median of price by cities',
       x="",
       y="")  +
  theme(
    legend.position = 'none') + coord_flip()
```




```
# With outlier
df |> ggplot(aes(city,price)) +
  geom_boxplot() +
  labs(x='',y='',title='Variance of price')
```




```
# Without Outlier
df |> ggplot(aes(city,price,fill=city)) +
  geom_boxplot(outlier.shape = NA) +  scale_y_continuous(limits = quantile(df$price,
  c(0.1, 0.9))) + theme(legend.position = "none") + labs(x='',y='',title='Variance of price without outl
```



Which cities have significant differences in prices?

```
kruskal.test(price ~ city, df)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  price by city
## Kruskal-Wallis chi-squared = 166361, df = 9, p-value < 2.2e-16
```

At least one group has a median different from the others. The Dunn test allows us to compare the medians between groups and identify where this difference occurs.

```
dunn_test(price ~ city, data = df, p.adjust.method = "bonferroni")
```

```
## # A tibble: 45 x 9
##    .y.   group1    group2    n1    n2 statistic         p     p.adj p.adj.signif
##  * <chr> <chr>     <chr>  <int> <int>     <dbl>     <dbl>     <dbl> <chr>
##  1 price Bangkok   Cape ~ 19361 19086     -2.20 2.78e-  2 1   e+  0 ns
##  2 price Bangkok   Hong ~ 19361  7087    -40.2  0          0         ****
##  3 price Bangkok   Istan~ 19361 24519    -95.8  0          0         ****
##  4 price Bangkok   Mexic~ 19361 20065    -26.2  1.23e-151 5.53e-150 ****
##  5 price Bangkok   New Y~ 19361 37012   -206.   0          0         ****
##  6 price Bangkok   Paris  19361 64690   -245.   0          0         ****
##  7 price Bangkok   Rio d~ 19361 26615    -88.1  0          0         ****
```

```
##  8 price Bangkok   Rome   19361 27647   -236.   0         0         ****
##  9 price Bangkok   Sydney 19361 33630   -173.   0         0         ****
## 10 price Cape Town Hong ~ 19086  7087    -38.6  0         0         ****
## # i 35 more rows
```

Relationship between price and number of bedrooms

```
df |>
  group_by (City=city) |> filter(bedrooms==1) |>
  summarise(`Median Price` = median(price)) |> arrange(desc(`Median Price`))
```

```
## # A tibble: 10 x 2
##    City           'Median Price'
##    <chr>                   <dbl>
##  1 Bangkok                   989
##  2 Cape Town                 732
##  3 Mexico City               500
##  4 Hong Kong                 314
##  5 Istanbul                  220
##  6 Rio de Janeiro            199
##  7 Sydney                     86
##  8 New York                   80
##  9 Paris                      79
## 10 Rome                       56
```

```
df |>
  group_by (City=city) |> filter(bedrooms==2) |>
  summarise(`Median Price` = median(price)) |> arrange(desc(`Median Price`))
```

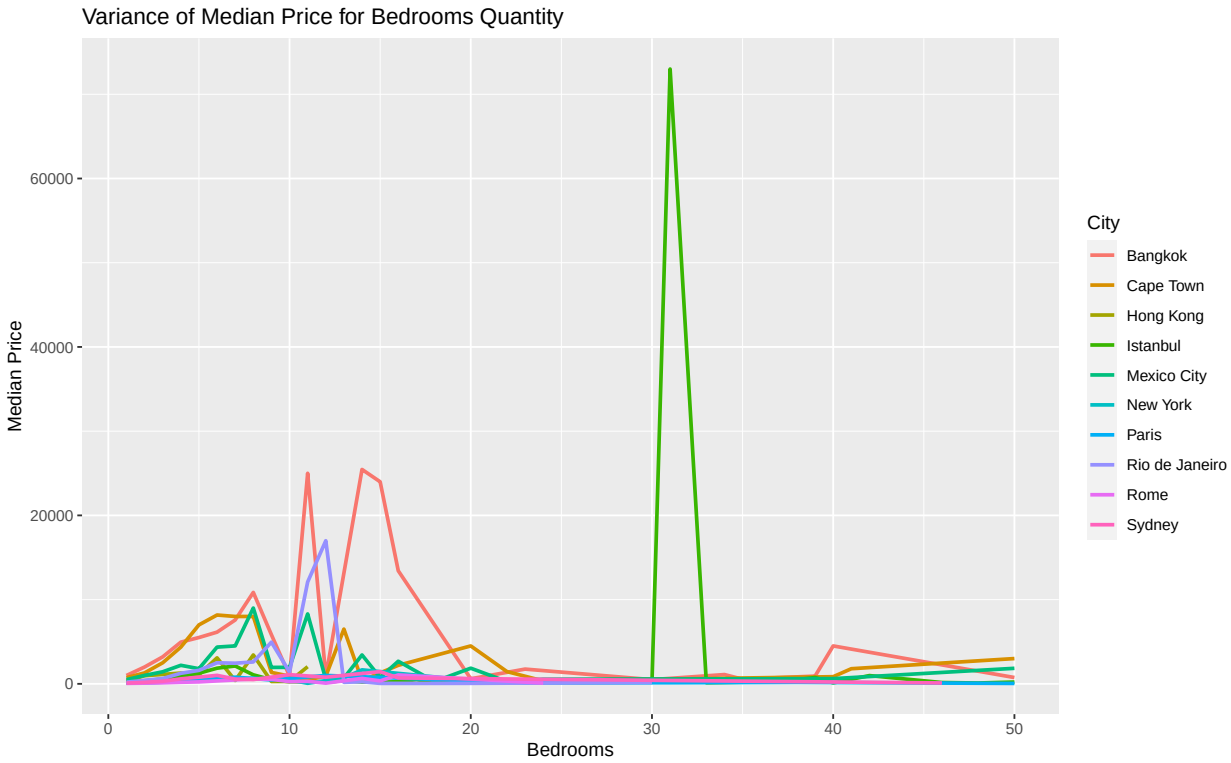
```
## # A tibble: 10 x 2
##    City           'Median Price'
##    <chr>                   <dbl>
##  1 Bangkok                  1999
##  2 Cape Town                1300
##  3 Mexico City              1000
##  4 Hong Kong                 965
##  5 Rio de Janeiro            414
##  6 Istanbul                  371
##  7 Sydney                    180
##  8 New York                  152
##  9 Paris                     125
## 10 Rome                       84
```

```
df |>
  group_by (City=city) |> filter(bedrooms>=3) |>
 summarise(`Median Price` = median(price)) |> arrange(desc(`Median Price`))
```

```
## # A tibble: 10 x 2
##    City           'Median Price'
##    <chr>                   <dbl>
```

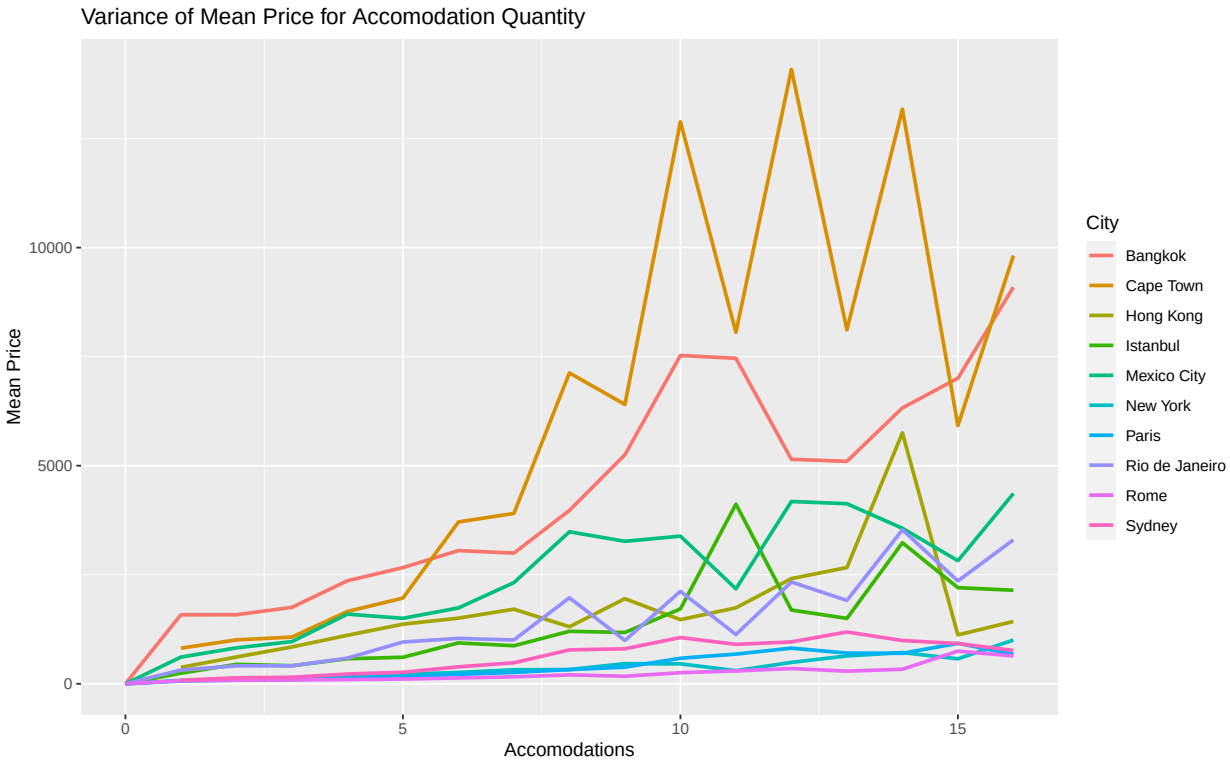
```
##  1 Bangkok                3800
##  2 Cape Town              3300
##  3 Mexico City            1549
##  4 Hong Kong              1100
##  5 Rio de Janeiro          793
##  6 Istanbul                600
##  7 Sydney                  410
##  8 New York                230.
##  9 Paris                   217
## 10 Rome                    131
```

```
df |>
  group_by(bedrooms, city) |>
  summarise(median_price = median(price)) |>
  ggplot(aes(x=bedrooms, y=median_price, color=city)) +
  geom_line(linewidth = 1) +
  labs(x='Bedrooms',y='Median Price',title='Variance of Median Price for Bedrooms Quantity',color='City
```



Relationship between price and number of accomodations

```
df |>group_by(city,accommodates) |>
  summarise(mean_price=mean(price)) |>
  ggplot(aes(x=accommodates,y=mean_price,color=city)) +
  geom_line(linewidth = 1) +
  labs(x='Accomodations',y='Mean Price',title='Variance of Mean Price for Accomodation Quantity',color=
```

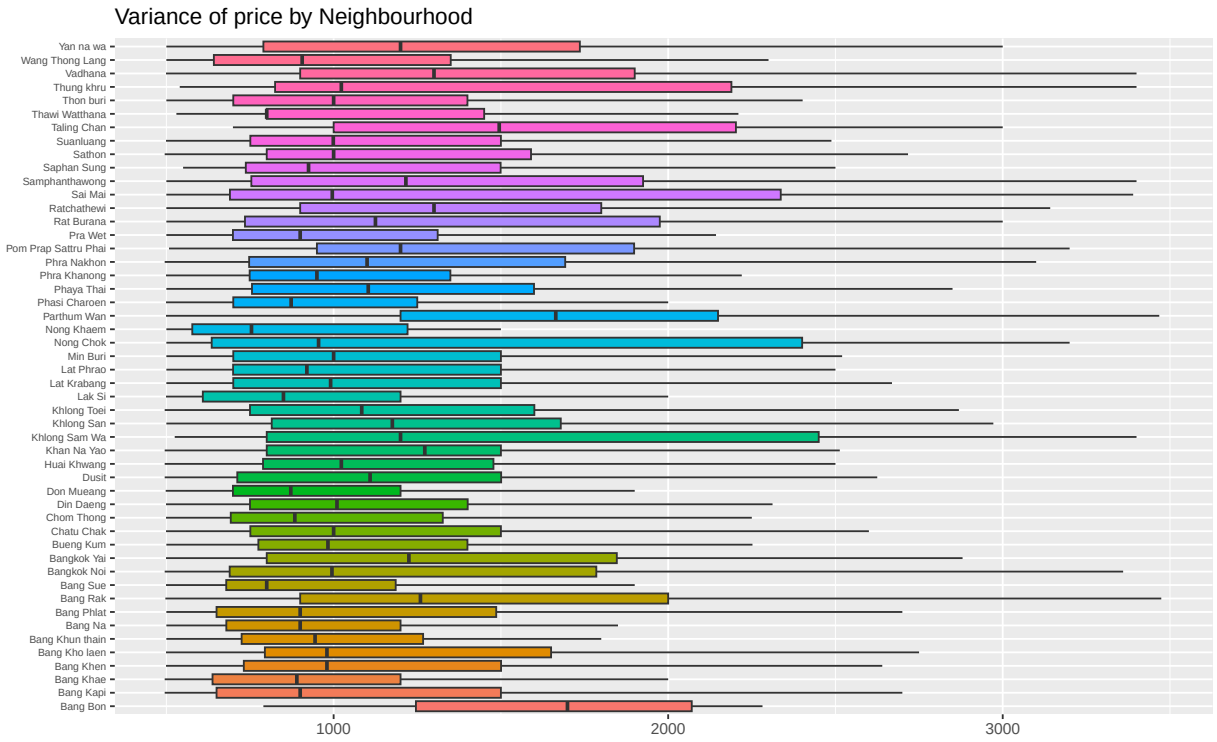


Bangkok Analysis

Bangkok appears to be the city with the most expensive places to rent. Let's explore this further

```
bang = df |> filter(city=='Bangkok')
```

```
bang |>
 ggplot(aes(neighbourhood,price,fill=neighbourhood)) +
  geom_boxplot(outlier.shape = NA) +  scale_y_continuous(limits = quantile(bang$price,
  c(0.1, 0.9))) +
  labs(x='',y='',title='Variance of price by Neighbourhood') +
  coord_flip() +
  theme(axis.text.y = element_text(size = 6, hjust = 1),
        legend.position = "none")
```



There is difference between prince by Neighourhood

```
kruskal.test(price ~ neighbourhood, bang)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  price by neighbourhood
## Kruskal-Wallis chi-squared = 1225.6, df = 49, p-value < 2.2e-16
```

The Dunn test allows us to compare the medians between groups and identify where this difference occurs.

```
dunn_test(price ~ neighbourhood, data=bang, p.adjust.method = "bonferroni")
```

```
## # A tibble: 1,225 x 9
##    .y.   group1   group2            n1    n2 statistic     p p.adj p.adj.signif
##  * <chr> <chr>    <chr>          <int> <int>     <dbl> <dbl> <dbl> <chr>
##  1 price Bang Bon Bang Kapi          7   332   -1.22   0.223     1 ns
##  2 price Bang Bon Bang Khae          7   103   -0.242  0.809     1 ns
##  3 price Bang Bon Bang Khen          7   148   -0.894  0.371     1 ns
##  4 price Bang Bon Bang Kho laen      7   154    0.124  0.901     1 ns
##  5 price Bang Bon Bang Khun thain    7    28    0.0608 0.952     1 ns
##  6 price Bang Bon Bang Na            7   575   -0.954  0.340     1 ns
##  7 price Bang Bon Bang Phlat         7   255   -1.11   0.266     1 ns
##  8 price Bang Bon Bang Rak           7  1079    0.635  0.526     1 ns
```

```
##  9 price Bang Bon Bang Sue            7   285   -1.33   0.183     1 ns
## 10 price Bang Bon Bangkok Noi         7   177   -0.115  0.909     1 ns
## # i 1,215 more rows
```

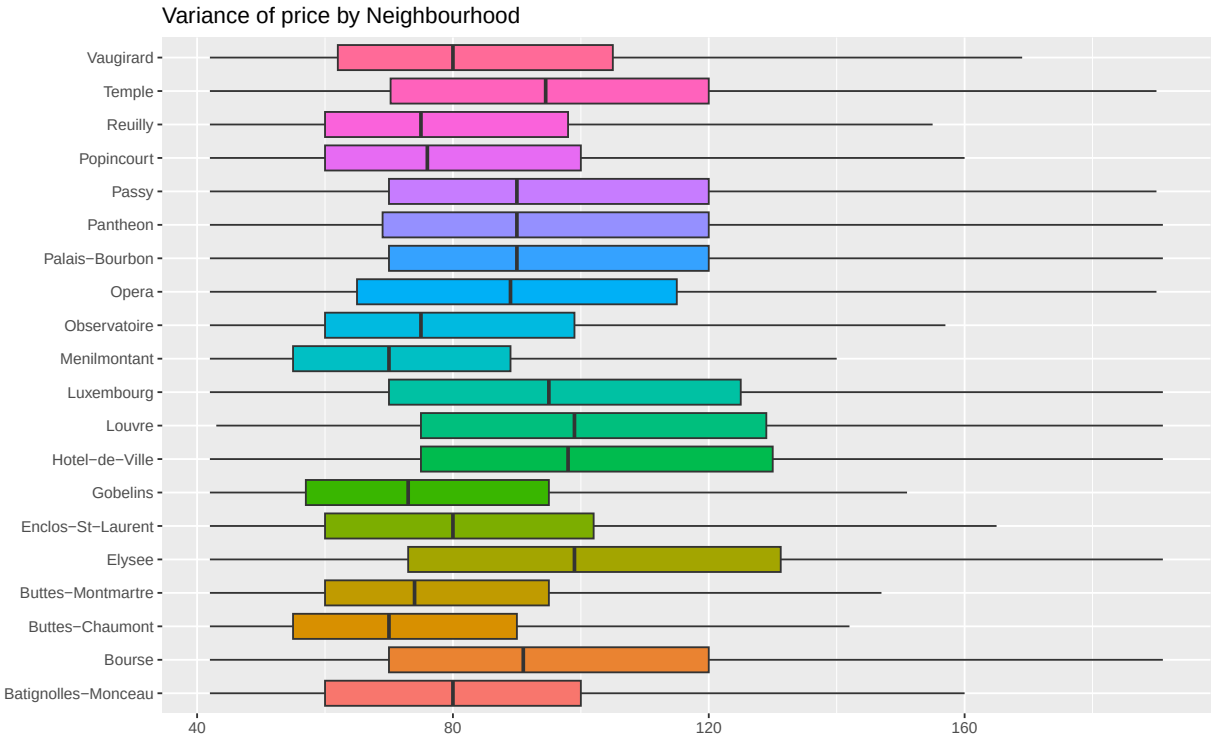
Few places in Bangkok actually have different rental prices considering the neighborhood

Paris analysis

Paris has the highest number of places available for hosting, so let's explore more about it.

```
paris = df |> filter(city=='Paris')
```

```
paris |>
 ggplot(aes(neighbourhood,price,fill=neighbourhood)) +
  geom_boxplot(outlier.shape = NA) +  scale_y_continuous(limits = quantile(paris$price,
  c(0.1, 0.9))) + theme(legend.position = "none") + labs(x='',y='',title='Variance of price by Neighbou
```



There is difference between price by Neighourhood

```
kruskal.test(price ~ neighbourhood, paris)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  price by neighbourhood
## Kruskal-Wallis chi-squared = 7281.2, df = 19, p-value < 2.2e-16
```

Again, at least one group has a median different from the others. Let's use the Dunn test again.

```
dunn_test(price ~ neighbourhood, data=paris, p.adjust.method = "bonferroni")
```

```
## # A tibble: 190 x 9
##    .y.   group1    group2    n1    n2 statistic         p     p.adj p.adj.signif
##  * <chr> <chr>     <chr>  <int> <int>     <dbl>     <dbl>     <dbl> <chr>
##  1 price Batignol~ Bourse  4330  2188    18.0   8.37e- 73 1.59e- 70 ****
##  2 price Batignol~ Butte~  4330  3728   -18.6   3.25e- 77 6.17e- 75 ****
##  3 price Batignol~ Butte~  4330  7237   -12.4   3.29e- 35 6.24e- 33 ****
##  4 price Batignol~ Elysee  4330  1768    25.6   1.02e-144 1.94e-142 ****
##  5 price Batignol~ Enclo~  4330  4628     0.301 7.64e-  1 1   e+  0 ns
##  6 price Batignol~ Gobel~  4330  2278    -9.62  6.39e- 22 1.21e- 19 ****
##  7 price Batignol~ Hotel~  4330  1972    21.6   2.16e-103 4.10e-101 ****
##  8 price Batignol~ Louvre  4330  1408    21.5   7.50e-103 1.42e-100 ****
##  9 price Batignol~ Luxem~  4330  1998    19.9   1.85e- 88 3.51e- 86 ****
## 10 price Batignol~ Menil~  4330  3758   -19.8   1.29e- 87 2.46e- 85 ****
## # i 180 more rows
```

Some places in Paris have price differences, while others do not.

Conclusion: What are the most expensive and least cities to book an Airbnb.

After my analysis, Bangkok was the city that presented the highest costs for booking an Airbnb, both in terms of the number of bedrooms and accommodations. However, Paris and Rome were the cheapest, using this same criteria.