Análise dos Dados do Airbnb - Rio de Janeiro

O Airbnb já é considerado como sendo a **maior empresa hoteleira da atualidade**. Ah, o detalhe é que ele **não possui nenhum hotel**!

Conectando pessoas que querem viajar (e se hospedar) com anfitriões que querem alugar seus imóveis de maneira prática, o Airbnb fornece uma plataforma inovadora para tornar essa hospedagem alternativa.

No final de 2018, a Startup fundada 10 anos atrás, já havia **hospedado mais de 300 milhões** de pessoas ao redor de todo o mundo, desafiando as redes hoteleiras tradicionais.

Uma das iniciativas do Airbnb é disponibilizar dados do site, para algumas das principais cidades do mundo. Por meio do portal Inside Airbnb, é possível baixar uma grande quantidade de dados para desenvolver projetos e soluções de *Data Science*.

Neste *notebook*, iremos analisar os dados referentes à cidade do Rio de Janeiro, e ver quais insights podem ser extraídos a partir de dados brutos.

Clique duas vezes (ou pressione "Enter") para editar

Clique duas vezes (ou pressione "Enter") para editar

Clique duas vezes (ou pressione "Enter") para editar

Obtenção dos Dados

Todos os dados usados aqui foram obtidos a partir do site Inside Airbnb.

Para esta análise exploratória inicial, será baixado apenas o seguinte arquivo:

- `listings.csv` - *Summary information and metrics for listings in Rio de Janeiro (good for visualisations).*

```
# importar os pacotes necessarios
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Análise dos Dados

Esta etapa tem por objetivo criar uma consciência situacional inicial e permitir um entendimento de como os dados estão estruturados.

Dicionário das variáveis

- `id` - número de id gerado para identificar o imóvel
- `name` - nome da propriedade anunciada
- `host_id` - número de id do proprietário (anfitrião) da propriedade
- `host_name` - Nome do anfitrião
- `neighbourhood_group` - esta coluna não contém nenhum valor válido
- `neighbourhood` - nome do bairro
- `latitude` - coordenada da latitude da propriedade
- `longitude` - coordenada da longitude da propriedade
- `room_type` - informa o tipo de quarto que é oferecido
- `price` - preço para alugar o imóvel
- `minimum_nights` - quantidade mínima de noites para reservar
- `number_of_reviews` - número de reviews que a propriedade possui
- `last_review` - data do último review
- `reviews_per_month` - quantidade de reviews por mês
- `calculated_host_listings_count` - quantidade de imóveis do mesmo anfitrião
- `availability_365` - número de dias de disponibilidade dentro de 365 dias

Antes de iniciar qualquer análise, vamos verificar a cara do nosso *dataset*, analisando as 5 primeiras entradas.

```
# mostrar as 5 primeiras entradas
df.head()
```

	id	name	host_id	host_name	neighbourhood_group	neighbou
35865	43222409	Suíte fenomenal em mansão de luxo	27531233	Tiago	NaN	Itan

		estadia.					
35869	43227835	Quarto confortável ao lado da Jeunesse Arena	125684729	Gustavo		NaN	Barra da

Q1. Quantos atributos (variáveis) e quantas entradas o nosso conjunto de dados possui? Quais os tipos das variáveis?

Vamos prosseguir e identificar a quantidade de entradas que nosso conjunto de dados possui e ver os tipos de cada coluna.

Este *dataset* que baixamos é a versão "resumida" do Airbnb. Na mesma página que baixamos o arquivo `listings.csv`. Há uma versão mais completa com 35847 entradas e 106 variáveis (`listings.csv.gz`).

```
# identificar o volume de dados do DataFrame
print("Entradas:\t {}".format(df.shape[0]))
print("Variáveis:\t {}\n".format(df.shape[1]))

# verificar as 5 primeiras entradas do dataset
display(df.dtypes)
```

```
Entradas:        35870
Variáveis:       16

id                                int64
name                             object
host_id                           int64
host_name                        object
neighbourhood_group             float64
neighbourhood                    object
```

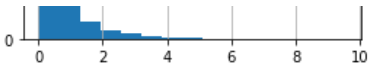
- É possível que a coluna `neighbourhood_group` possua 100% dos seus valores faltantes.
- As variáveis `reviews_per_month` e `last_review` possuem valores nulos em quase metade das linhas.
- As variáveis `name` e `host_name` têm aproximadamente 0,1% dos valores nulos.

```
# ordenar em ordem decrescente as variáveis por seus valores ausentes
(df.isnull().sum() / df.shape[0]).sort_values(ascending=False)
```

```
neighbourhood_group               1.000000
reviews_per_month                 0.411653
last_review                       0.411653
name                              0.001617
host_name                         0.000139
availability_365                  0.000000
calculated_host_listings_count    0.000000
number_of_reviews                 0.000000
minimum_nights                    0.000000
price                             0.000000
room_type                         0.000000
longitude                         0.000000
latitude                          0.000000
neighbourhood                     0.000000
host_id                           0.000000
id                                0.000000
dtype: float64
```

Q3. Qual o tipo de distribuição das variáveis?

Para identificar a distribuição das variáveis, irei plotar o histograma.



Q4. Há *outliers* presentes?

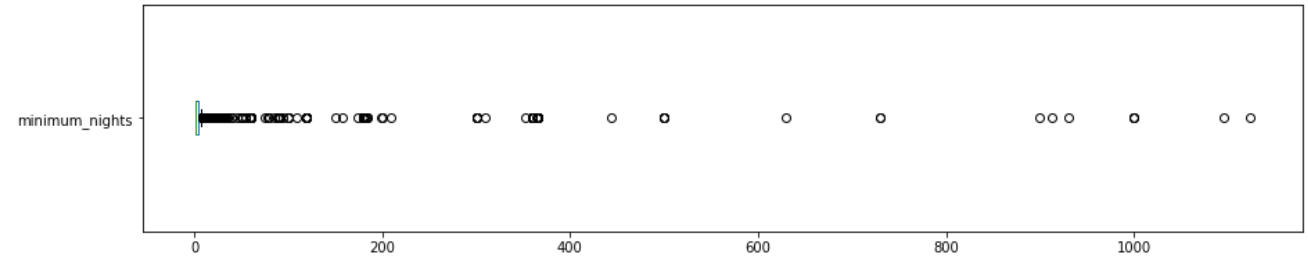
Pela distribuição do histograma, é possível verificar indícios da presença de *outliers*. Olhe por exemplo as variáveis `price`, `minimum_nights` e `calculated_host_listings_count`.

Os valores não seguem uma distruição, e distorcem toda a representação gráfica. Para confirmar, há duas maneiras rápidas que auxiliam a detecção de *outliers*. São elas:

- Resumo estatístico por meio do método `describe()`
- Plotar `boxplots` para a variável.

```
plt.show()

# ver quantidade de valores acima de 30 dias para minimum_nights
print("minimum_nights: valores acima de 30:")
print("{} entradas".format(len(df[df.minimum_nights > 30])))
print("{:.4f}%".format((len(df[df.minimum_nights > 30]) / df.shape[0])*100))
```

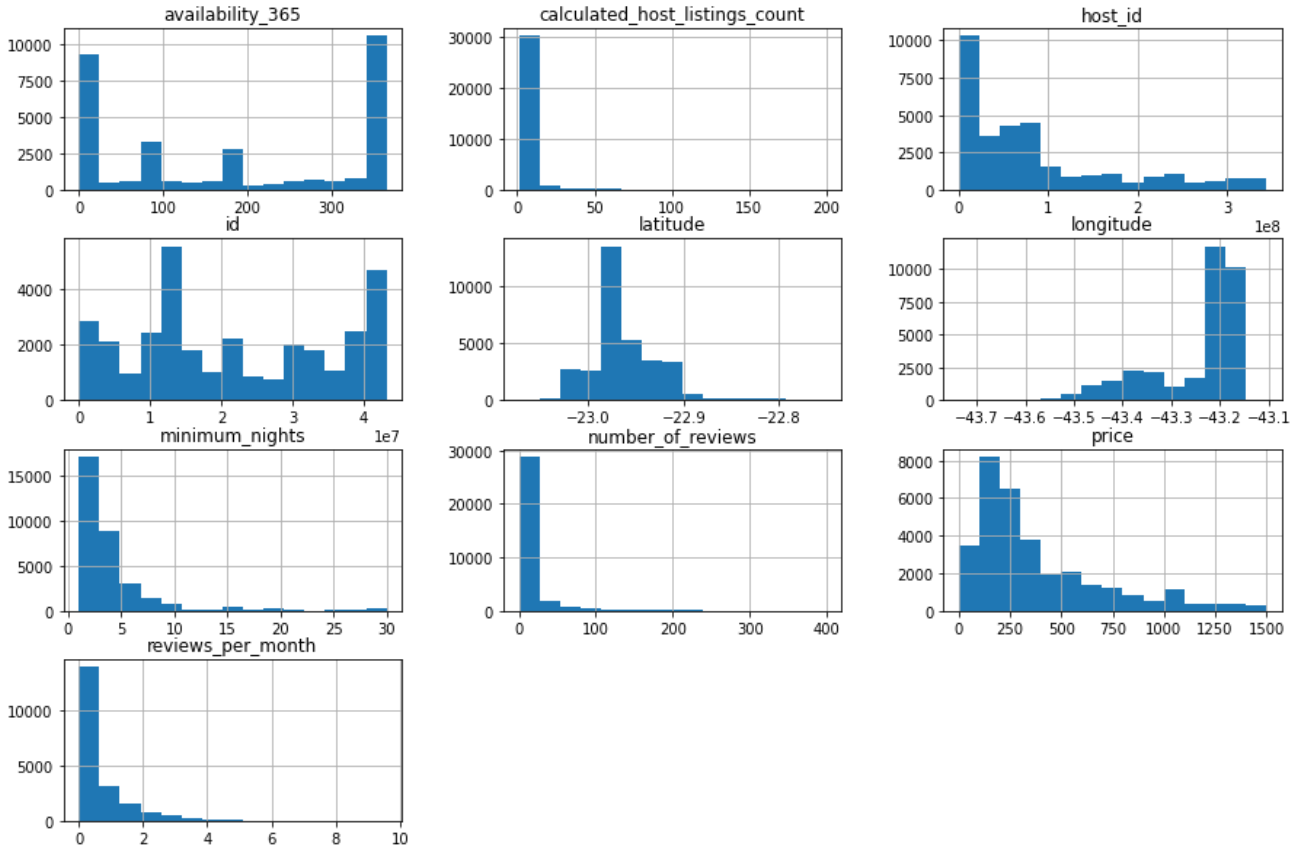



```
minimum_nights: valores acima de 30:
224 entradas
0.6245%
```

```
# remover os *outliers* em um novo DataFrame
df_clean = df.copy()
df_clean.drop(df_clean[df_clean.price > 1500].index, axis=0, inplace=True)
df_clean.drop(df_clean[df_clean.minimum_nights > 30].index, axis=0, inplace=True)

# remover `neighbourhood_group`, pois está vazio
df_clean.drop('neighbourhood_group', axis=1, inplace=True)

# plotar o histograma para as variáveis numéricas
df_clean.hist(bins=15, figsize=(15,10));
```



```
# criar uma matriz de correlação
corr = df_clean[['price', 'minimum_nights', 'number_of_reviews', 'reviews_per_month
    'calculated_host_listings_count', 'availability_365']].corr()

display(corr)
```

```
df_clean.room_type.value_counts()
```