

# aula01\_resolucao\_exercicio

February 27, 2021

## 1 Aula 01 - Resolução dos Exercícios

### 1.1 Novas perguntas do CEO para vocês

1. Quantas casas estão disponíveis para compra?
2. Quantos atributos as casas possuem?
3. Quais são os atributos das casas?
4. Qual a casa mais cara ( casa com o maior valor de venda )?
5. Qual a casa com o maior número de quartos?
6. Qual a soma total de quartos do conjunto de dados?
7. Quantas casas possuem 2 banheiros?
8. Qual o preço médio de todas as casas no conjunto de dados?
9. Qual o preço médio de casas com 2 banheiros?
10. Qual o preço mínimo entre as casas com 3 quartos?
11. Quantas casas possuem mais de 300 metros quadrados na sala de estar?
12. Quantas casas tem mais de 2 andares?
13. Quantas casas tem vista para o mar?
14. Das casas com vista para o mar, quantas tem 3 quartos?
15. Das casas com mais de 300 metros quadrados de sala de estar, quantas tem mais de 2 banheiros?

## 2 Resolução

### 2.1 Import Libraries

```
[4]: import numpy as np
import pandas as pd
import seaborn as sns

from matplotlib import pyplot as plt
import plotly.express as px
```

```
[5]: # Suppress Scientific Notation
np.set_printoptions(suppress=True)
pd.set_option('display.float_format', '{:.2f}'.format)
```

## 2.2 Loading Data

```
[21]: # loading data into memory
data = pd.read_csv( '../kc_house_data.csv' )
```

### 2.3 1. Quantas casas estão disponíveis para compra?

```
[22]: # Eu vou considerar que a coluna "id" representa a identificação única do imóvel
num_houses_unique = data['id'].nunique()

# Resultado
print( 'Estão disponíveis {} imóveis'.format( num_houses_unique ) )
```

Estão disponíveis 21436 imóveis

### 2.4 2. Quantos atributos as casas possuem?

```
[27]: # O numero de colunas representam os atributos do apartamento.
# id e date - não são atributos do apartamento
num_attributes = len( data.columns ) - 2

# Resultado
print( 'Os imóveis possuem {} atributos'.format( num_attributes ) )
```

Os imóveis possuem 19 atributos

### 2.5 3. Quais são os atributos das casas?

```
[29]: # Estratégia: Excluir as colunas "id", "date" e mostrar os atributos restantes
df = data.drop( ['id', 'date'], axis=1 )
print( df.columns.tolist() )
```

```
['price', 'bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'floors',
'waterfront', 'view', 'condition', 'grade', 'sqft_above', 'sqft_basement',
'yr_built', 'yr_renovated', 'zipcode', 'lat', 'long', 'sqft_living15',
'sqft_lot15']
```

### 2.6 4. Qual a casa mais cara ( casa com o maior valor de venda )?

```
[56]: # Estratégia: Selecionar a coluna "id", "price", ordenar as casas pela coluna
↳ "price" em ordem decrescente e
# escolher o imóvel do primeiro id.
house_expensive = data[['id', 'price']].sort_values( 'price', ascending=False ).
↳ loc[0, 'id']
```

```
print( 'Most expensive house: {}'.format( house_expensive ) )
```

Most expensive house: 7129300520

## 2.7 5. Qual a casa com o maior número de quartos?

```
[58]: # Estratégia:
# 1. Selecionar a coluna "id", "bedroom"
# 2. Ordenar os imóveis pelo numero de quartos de ordem decrescente
# 3. Selecionar a primeira coluna "id"

df = data[['id', 'bedrooms']].sort_values( 'bedrooms', ascending=False )

print( 'House with biggest number of bedrooms: {}'.format( df.iloc[0, 0] ) )
```

House with biggest number of bedrooms: 2402100895

## 2.8 6. Qual a soma total de quartos do conjunto de dados?

```
[61]: # Estratégia:
# 1. Somar a coluna "bedrooms"
print( 'Total sum of bedrooms: {}'.format( data['bedrooms'].sum() ) )
```

Total sum of bedrooms: 72854

## 2.9 7. Quantas casas possuem 2 banheiros?

```
[68]: # Estratégia:
# 1. Filtrar linhas (imóveis) que possuem 2 banheiros.
# 2. Contar o número de linhas do dataset

df = data.loc[data['bathrooms'] == 2, :]
num_houses = len( df )

print( 'Total number of houses with 2 bathrooms: {}'.format( num_houses ) )
```

Total number of houses with 2 bathrooms: 1930

## 2.10 8. Qual o preço médio de todas as casas no conjunto de dados?

```
[72]: # Estratégia:
# 1. Calcular o preço médio da coluna "price"
avg_price = np.round( data['price'].mean(), 2 )

print( 'Average Price: ${}'.format( avg_price ) )
```

Average Price: \$540088.14

### 2.11 9. Qual o preço médio de casas com 2 banheiros?

```
[94]: # Estratégia:
# 1. Selecionar imóveis com 2 banheiros.
# 2. Calcular o preço médio da coluna "price" do novo conjunto de dados

avg_price = np.round( data.loc[data['bathrooms'] == 2, 'price'].mean(), 2 )

print( 'Average Price: ${}'.format( avg_price ) )
```

Average Price: \$457889.72

### 2.12 10. Qual o preço mínimo entre as casas com 3 quartos?

```
[93]: # Estratégia:
# 1. Selecionar imóveis com 3 bedrooms.
# 2. Calcular o menor preço da coluna "price" do novo conjunto de dados

min_price = np.round( data.loc[data['bedrooms'] == 3, 'price'].min(), 2 )

print( 'Min Price: ${}'.format( min_price ) )
```

Min Price: \$82000.0

### 2.13 11. Quantas casas possuem mais de 300 metros quadrados na sala de estar?

```
[92]: # Estratégia:
# 1. Selecionar imóveis com mais de 300 sqft_living.
# 2. Contar o número de imóveis nesse novo conjunto de dados.

houses = data.loc[data['sqft_living'] > 300, 'id'].shape[0]

print( 'There are {} Houses with living room grater than 300 square foot'.
      ↪format( houses ) )
```

There are 21612 Houses with living room grater than 300 square foot

### 2.14 12. Quantas casas tem mais de 2 andares?

```
[91]: # Estratégia:
# 1. Selecionar imóveis com a coluna 'floors' maior que 2.
# 2. Contar o número de imóveis nesse novo conjunto de dados.

houses = data.loc[data['floors'] > 2, 'id'].shape[0]

print( 'There are {} Houses with the number of floors grater than 2'.format(
      ↪houses ) )
```

There are 782 Houses with the number of floors grater than 2

## 2.15 13. Quantas casas tem vista para o mar?

```
[95]: # Estratégia:
# 1. Selecionar imóveis com a coluna 'waterfront' igual a 1.
# 2. Contar o número de imóveis nesse novo conjunto de dados.

houses = data.loc[data['floors'] == 1, 'id'].shape[0]

print( 'There are {} Houses with the number of floors grater than 2'.format(
    ↪houses ) )
```

There are 10680 Houses with the number of floors grater than 2

## 2.16 14. Das casas com vista para o mar, quantas tem 3 quartos?

```
[97]: # Estratégia:
# 1. Selecionar imóveis com a coluna 'waterfront' igual a 1 e a coluna
    ↪'bedrooms' maior que 3.
# 2. Contar o número de imóveis nesse novo conjunto de dados.

houses = data.loc[(data['waterfront'] == 1) & (data['bedrooms'] > 2), 'id'].
    ↪shape[0]

print( 'There are {} Houses with waterfront and bedrooms greater then 3'.
    ↪format( houses ) )
```

There are 127 Houses with waterfront and bedrooms greater then 3

## 2.17 15. Das casas com mais de 300 metros quadrados de sala de estar, quantas tem mais de 2 banheiros?

```
[99]: # Estratégia:
# 1. Selecionar imóveis com a coluna 'sqft_living' maior que 300 e a coluna
    ↪'bathrooms' maior que 2.
# 2. Contar o número de imóveis nesse novo conjunto de dados.

houses = data.loc[(data['sqft_living'] > 300) & (data['bathrooms'] > 2), 'id'].
    ↪shape[0]

print( 'There are {} Houses with living room greater than 300 and bathrooms
    ↪bigger than 2'.format( houses ) )
```

There are 11242 Houses with living room greater than 300 and bathrooms bigger than 2

[ ]: