

```
!pip install pandas==1.5.3
!pip install plotly==5.13.0
!pip install sidetable

from scipy import stats
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
# import sidetable
```

Existem diversos fatores que os norte-americanos consideram importantes na hora de comprar um imóvel,elenco aqui alguns dos principais:

1. Localização: A localização é muitas vezes considerada a característica mais importante na escolha de um imóvel nos Estados Unidos. As pessoas geralmente procuram por imóveis em bairros seguros e próximos a escolas, comércio, transporte público e outros serviços.
2. Preço: O preço é sempre um fator importante na compra de um imóvel. Os norte-americanos geralmente têm um orçamento definido para sua compra e procuram encontrar a melhor relação entre preço e qualidade.
3. Tamanho: O tamanho da propriedade também é uma consideração importante. As necessidades variam dependendo do tamanho da família e da finalidade do imóvel (por exemplo, se será uma casa para morar ou uma propriedade de investimento).
4. Condição da propriedade: A condição da propriedade é um fator importante na hora de decidir comprar uma casa ou apartamento. As pessoas geralmente preferem imóveis em boas condições, com poucas ou nenhuma reforma necessária.
5. Comodidades: Os norte-americanos também valorizam as comodidades oferecidas pelo imóvel, como ar-condicionado, aquecedores, piscinas, áreas externas, entre outros.
6. História do imóvel: Para alguns compradores, a história do imóvel também pode ser um fator importante na decisão de compra. Isso pode incluir coisas como a idade da propriedade, sua história arquitetônica e sua história como um local importante na comunidade.
7. A localização costeira pode oferecer vistas deslumbrantes, acesso a praias e atividades ao ar livre, além de um estilo de vida tranquilo. No entanto, a proximidade ao mar pode influenciar no preço do imóvel, então os compradores devem estar preparados para pagar um valor mais elevado nesses casos. É importante lembrar que a localização é um fator pessoal e depende das preferências de cada indivíduo ou família que está procurando comprar um imóvel.

Sobre os Dados

Variável	Descrição
id	Identificador único do imóvel
date	Data da Venda
price	Preço de venda
bedrooms	Nº de Quartos
bathrooms	Nº de Banheiros
sqft_liv	Tamanho de área habitável em ft²
sqft_lot	Tamanho do terreno em ft²
floors	Número de andares
waterfront	Indicativo se o imóvel é a beira-mar

Variável	Descrição
view	Grau de quão belo é a vista do imóvel (0 a 4)
condition	Condição da casa (1 a 5)
grade	Classificação por qualidade de material utilizado na construção
sqft_above	Área em acima do solo em ft²
sqft_basmt	Área em abaixo do solo em ft²
yr_built	Ano de construção
yr_renov	Ano de restauração, caso o contrário, 0.
zipcode	Zip Code 5 - Similar ao CEP
lat	Latitude
long	Longitude
sqft_liv15	Média da área habitável dos 15 imóveis mais próximos, em ft²
sqft_lot15	Média da área do lote dos 15 imóveis mais próximos, em ft²

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21613 entries, 0 to 21612
Data columns (total 21 columns):
#   Column              Non-Null Count  Dtype
---  -
0   id                   21613 non-null  int64
1   date                 21613 non-null  object
2   price                21613 non-null  float64
3   bedrooms             21613 non-null  int64
4   bathrooms            21613 non-null  float64
5   sqft_living          21613 non-null  int64
6   sqft_lot             21613 non-null  int64
7   floors               21613 non-null  float64
8   waterfront           21613 non-null  int64
9   view                 21613 non-null  int64
10  condition            21613 non-null  int64
11  grade                21613 non-null  int64
12  sqft_above           21613 non-null  int64
13  sqft_basement        21613 non-null  int64
14  yr_built              21613 non-null  int64
15  yr_renovated          21613 non-null  int64
16  zipcode              21613 non-null  int64
17  lat                  21613 non-null  float64
18  long                 21613 non-null  float64
19  sqft_living15         21613 non-null  int64
20  sqft_lot15           21613 non-null  int64
dtypes: float64(5), int64(15), object(1)
memory usage: 3.5+ MB
```

```
# Verificar a consistência dos dados do DataSet:
# Ver se ha valores nulos
if df.isna().any().any():
    print("Existem valores nulos no dataset")
else:
    print("Nao Existem valores nulos no dataset")
```

Nao Existem valores nulos no dataset

```
# Primeira busca por outliers, observando o desvio padrão de cada atributo
# compara com média fornecida pelo describe()
df.describe()
```

	Unnamed: 0	id	preco_venda	num_quarto	num_banheiro	t_area_habitavel	tamanho_terreno	qtd_andar
count	21613.00000	2.161300e+04	2.161300e+04	21613.000000	21613.000000	21613.000000	2.161300e+04	21613.000000
mean	10806.00000	4.580302e+09	5.400881e+05	3.370842	2.114757	2079.899736	1.510697e+04	1.494300e+03
std	6239.28002	2.876566e+09	3.671272e+05	0.930062	0.770163	918.440897	4.142051e+04	0.539900e+03
min	0.00000	1.000102e+06	7.500000e+04	0.000000	0.000000	290.000000	5.200000e+02	1.000000e+02
25%	5403.00000	2.123049e+09	3.219500e+05	3.000000	1.750000	1427.000000	5.040000e+03	1.000000e+03
50%	10806.00000	3.904930e+09	4.500000e+05	3.000000	2.250000	1910.000000	7.618000e+03	1.500000e+03
75%	16209.00000	7.308900e+09	6.450000e+05	4.000000	2.500000	2550.000000	1.068800e+04	2.000000e+03
max	21612.00000	9.900000e+09	7.700000e+06	33.000000	8.000000	13540.000000	1.651359e+06	3.500000e+03

8 rows × 21 columns



```
# convertendo alguns atributos para facilitar minha interpretação dos dados
df = df.rename(columns={
    'bedrooms': 'num_quarto',
    'bathrooms': 'num_banheiro',
    'condition': 'condicao_casa_1_5',
    'date': 'data_venda',
    'price': 'preco_venda',
    'sqft_living': 't_area_habitavel',
    'sqft_lot': 'tamanho_terreno',
    'floors': 'qtd_andares',
    'waterfront': 'imovel_beira_mar',
    'view': 'grau_bela_vista',
    'grade': 'qualidade_material_const',
```

```
'sqft_above': 'area_acima_solo',
'sqft_basement': 'area_baixo_solo',
'sqft_living15': 'media_area_habitavel_15_prox',
'sqft_lot15': 'media_area_terreno_15_prox',
'yr_built': 'ano_construcao',
'yr_renovated': 'ano_reforma',
})
```

```
# Criei uma cópia do dataframe para manter a integridade do anterior
house_data = df
```

```
# Busca o nome da cidade pelo zipcode, para na sequencia, adicionar o campo cidade no dataframe
# Este método possibilita buscar uma cidade com maior vantagem para o cliente
zipcode = pd.read_html('https://www.zipcode.com.ng/2022/12/king-county-zip-codes-wa.html')
zipcode = zipcode[1] # tabela 1
zipcode.head()
```

```
# inserindo as cidades com base no zip
zipcodes = []
for row in house_data.index:
    indiceZip = np.where(house_data['zipcode'][row] == zipcode['ZIP Code'])[0][0]
    zipcodes.append(zipcode['City'][indiceZip])
```

```
house_data['cidade'] = zipcodes
del zipcodes
```

```
# Converter a data para ficar mais legível e poder operar com mais facilidade
house_data['data_venda'] = pd.to_datetime(house_data['data_venda']).dt.date
```

```
# Cria um dataset com os nomes dos atributos traduzidos para português
# par ajudar na compreensão de cada um
house_data.to_csv('/content/sample_data/house.csv')
```

```
# Busca o dataset direto do meu repositório git
house_data = df = pd.read_csv('https://raw.githubusercontent.com/valdirdpg/curso-dnc/master/MODULO-3/MINI-PROJETO/house_data.csv')
# Removendo a coluna 'Unnamed: 0', criada de forma arbitrária
house_data = house_data.drop('Unnamed: 0', axis=1)
house_data
```

	id	data_venda	preco_venda	num_quarto	num_banheiro	t_area_habitavel	tamanho_terreno	qtd_andare:
0	7129300520	2014-10-13	221900	3	1.00	1180	5650	1.0
1	6414100192	2014-12-09	538000	3	2.25	2570	7242	2.0
2	5631500400	2015-02-25	180000	2	1.00	770	10000	1.0
3	2487200875	2014-12-09	604000	4	3.00	1960	5000	1.0
4	1954400510	2015-02-18	510000	3	2.00	1680	8080	1.0
...
21608	263000018	2014-05-21	360000	3	2.50	1530	1131	3.0
21609	6600060120	2015-02-23	400000	4	2.50	2310	5813	2.0
21610	1523300141	2014-06-23	402101	2	0.75	1020	1350	2.0
21611	291310100	2015-01-16	400000	3	2.50	1600	2388	2.0
21612	1523300157	2014-10-15	325000	2	0.75	1020	1076	2.0

21613 rows × 22 columns



...

Com base na pesquisa que fiz sobre os interesses dos americanos na hora de comprar imóveis, decide procurar imóveis com o seguinte requisitos:

- 1 - com vista para mar
- 2 - com bela paisagem - 3 ou 4
- 3 - Menor preço - entre 250000 e 1500000
- 4 - Maior quantidade de quartos - entre 2 e 4

```
5 - maior quantidade de banheiro entr 1 e 3
6 - Maior área habitável
7 - Ano de Construção - preferência acima de 1980 se atender as outras condições
8 - Ano de Restauração* - Não é muito importante
9 - Quantidade de andares - entr 1 e 8
10- Estado de conservação - entre 3 e 5
11- Maior Avaliação de Material de construção - entre 10 e 13
12- Local com opções de lazer e segurança

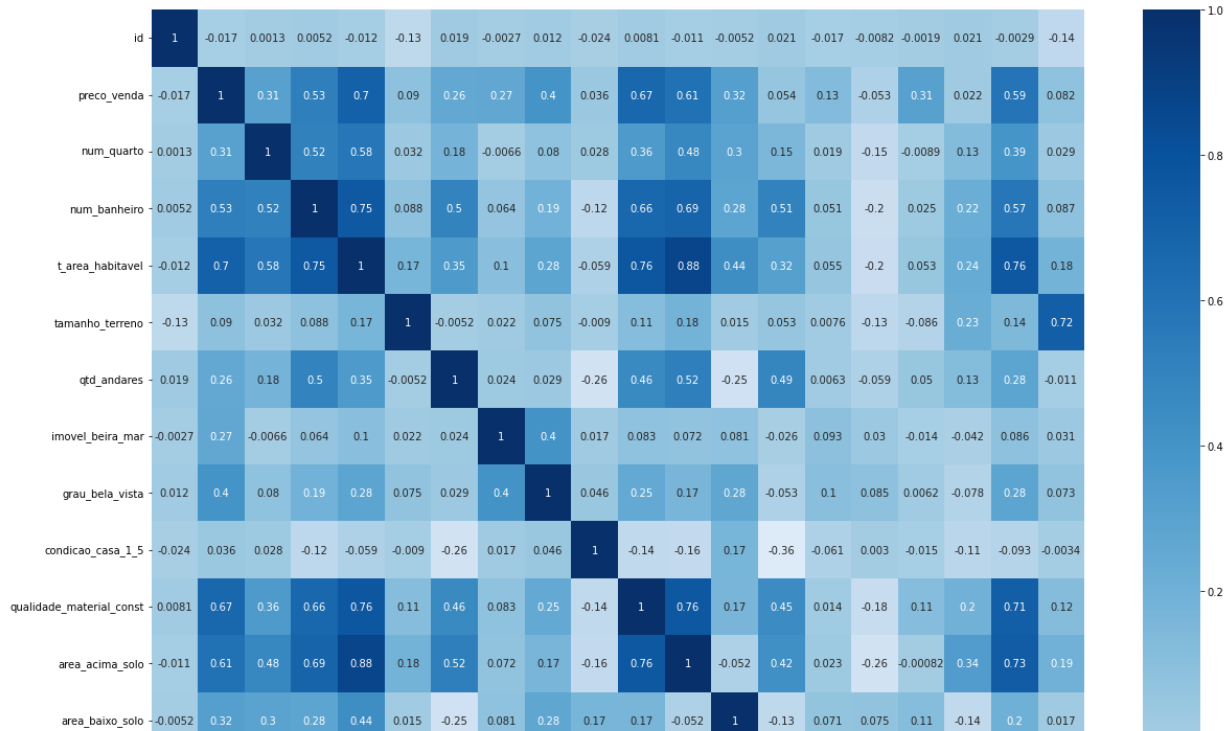
Desta forma devo selecionar as principais colunas que devo buscar respostas
para as minhas perguntas:
Quais os imoveis mas conservados, com quantidade comoda de quartos e basn
heiros
com bela vista e proximo ao mar que representa o melhor custo beneficio?

...

# Com o describe, estou buscando os primeiros outliers
# nesta primeira tomada não vi nada de muito significativo da diferença
# da média para o desvio padrão
house_data.describe()
```

	id	preco_venda	num_quarto	num_banheiro	t_area_habitavel	tamanho_terreno	qtd_andares	imovel_l
count	2.161300e+04	2.161300e+04	21613.000000	21613.000000	21613.000000	2.161300e+04	21613.000000	21
mean	4.580302e+09	5.400881e+05	3.370842	2.114757	2079.899736	1.510697e+04	1.494309	
std	2.876566e+09	3.671272e+05	0.930062	0.770163	918.440897	4.142051e+04	0.539989	
min	1.000102e+06	7.500000e+04	0.000000	0.000000	290.000000	5.200000e+02	1.000000	
25%	2.123049e+09	3.219500e+05	3.000000	1.750000	1427.000000	5.040000e+03	1.000000	
50%	3.904930e+09	4.500000e+05	3.000000	2.250000	1910.000000	7.618000e+03	1.500000	
75%	7.308900e+09	6.450000e+05	4.000000	2.500000	2550.000000	1.068800e+04	2.000000	
max	9.900000e+09	7.700000e+06	33.000000	8.000000	13540.000000	1.651359e+06	3.500000	

```
# Buscando as principais correlações de variáveis que afetam o preço
plt.figure(figsize=(20,20))
correlacao = sns.heatmap(house_data.corr(),annot=True, cmap='Blues');
```



Existe uma boa relação entre:

- preço de venda e a qualidade do material utilizado
- preço de venda e numero de banheiro
- preço de venda e tamanho de área habitável
- preço de venda e quantidade de quartos apresenta uma relação menor. - vou testar com o boxplot esta variável
- achei importante entender que não existe uma relação direta entre tamanho do terreno e o preço
- Existem outras realações, mas não interessa para uma análise imediata do problema

```
# Aqui faço agrupamento dos dados utilizando as principais váriáveis que afetam o preço de venda
agrupa_house = house_data.groupby([
    'preco_venda', 'qualidade_material_const',
    'num_banheiro', 't_area_habitavel', 'media_area_habitavel_15_prox'
], sort=False).apply(lambda x: x.sort_values(['preco_venda'], ascending=False))
```

```
# Busco compreender a relação entre a quantidade de quartos e preço tendo, observando
# quantos banheiros possuem uma casa com determinada quantidade de quartos.
# apesar da quantidade de quartos não dizer muito sobre o preço, este item é importante
# para quem buscar a comodidade de uma residência
gboxplot = px.box(agrupa_house, x='num_quarto', y='preco_venda', hover_name='num_banheiro')
gboxplot.show()
```





Neste Gráfico de caixas fica claro que realmente não há uma relação direta entre preço de venda e quantidade de quartos. Há muita discrepância nestas duas variáveis. Tem casas com grande numero de quartos mais baratas do que outras que possuem um número bem menor, aparentemente pode representar alguma vantagem, mas ainda precisa ser comparado com outra variáveis. Se passar o mouse no gráfico do perceber por exemplo que tem casas com 3 quartos com valor mínimo de 82k e máximo de 955k. Desta forma percebo que outras variáveis afetam este conjunto, e por isso estou construindo filtros com estas variáveis:

- Qualidade do Material usado na construção
- Número de Banheiros
- Quantidade de andar
- Proximidade com praias (Mar)
- Bo vista da casa
- Tamanho da área habitável
- Média de área habitável dos 15 imóveis mais próximos

```
# No gráfico acima percebe-se um claro outliers no número de quartos e não representa uma oportunidade
house_data[((house_data['num_quarto']>30) | (house_data['num_quarto']>10))]
```

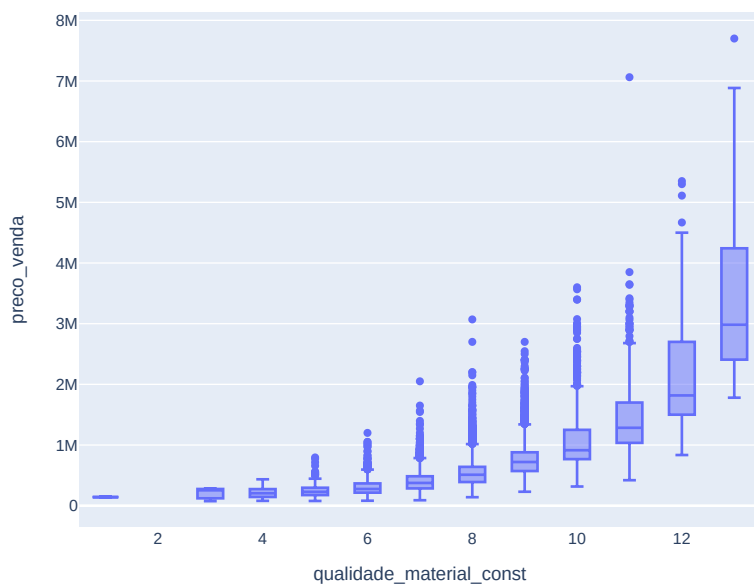
	id	data_venda	preco_venda	num_quarto	num_banheiro	t_area_habitavel	tamanho_terreno	qtd_andares
8757	1773100755	2014-08-21	520000	11	3.00	3000	4960	2.00
15870	2402100895	2014-06-25	640000	33	1.75	1620	6000	1.00

2 rows × 22 columns

Warning: Total number of columns (22) exceeds max_columns (20) limiting to first (20) columns.

```
# observando que uma casa com 33 quartos (descobri esta casa com auxilio do Box)
# com um tamanho habitável considerável, custando apenas 640000, chama atenção
# Resta ver o quanto a qualidade do material impacta no preço das casas, já que a qualidade
# do material é baixa (7 de 13), mas o estado da casa é 5, nos padrões americanos isso repretenta:
# Nível 4: A casa está em excelentes condições e pronta para morar, sem necessidade de reparos
# ou atualizações significativas.
# Nível 5: A casa está em condições excepcionais e tem recursos de alto nível,
# como acabamentos de luxo, equipamentos modernos e/ou design personalizado.
```

```
# Análise do preço em relação a qualidade do material e o estado de conservação do imóvel
gboxplot = px.box(agrupa_house,x='qualidade_material_const', y='preco_venda', hover_name='condicao_casa_1_5')
gboxplot.show()
```



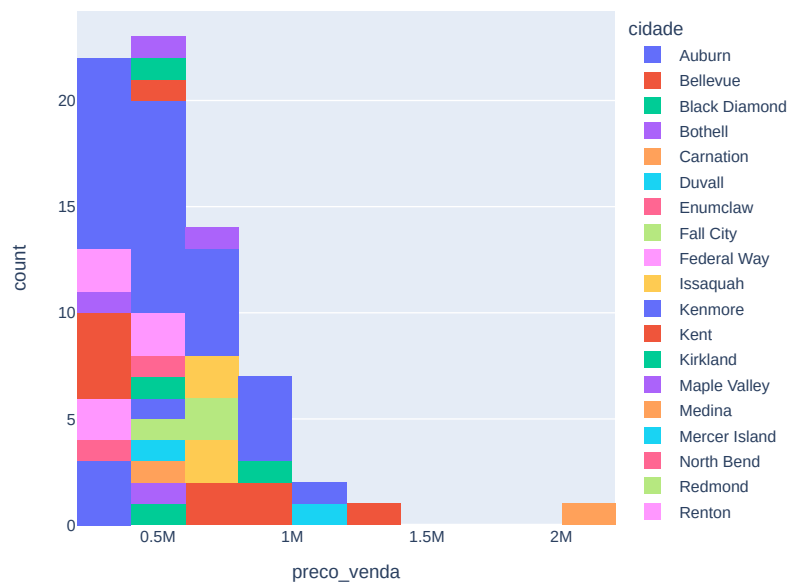
Apesar de alguns valores estarem distante da média geral, percebe-se que quanto maior a qualidade do material, maior é avaliação do imóvel e maior o preço

```
# Encontrando a media dos preços por região, para identificar
# a região com preços atrativos.
median_preco = house_data.groupby(['cidade', 'zipcode']).agg({'preco_venda': 'mean'}).reset_index()
median_preco
```

	cidade	zipcode	preco_venda
0	Auburn	98001	2.808047e+05
1	Auburn	98002	2.342840e+05
2	Auburn	98092	3.349211e+05
3	Bellevue	98004	1.355927e+06
4	Bellevue	98005	8.101649e+05
...
65	Seattle	98199	7.918208e+05
66	Snoqualmie	98065	5.279612e+05
67	Vashon	98070	4.874796e+05
68	Woodinville	98072	5.699585e+05
69	Woodinville	98077	6.827749e+05

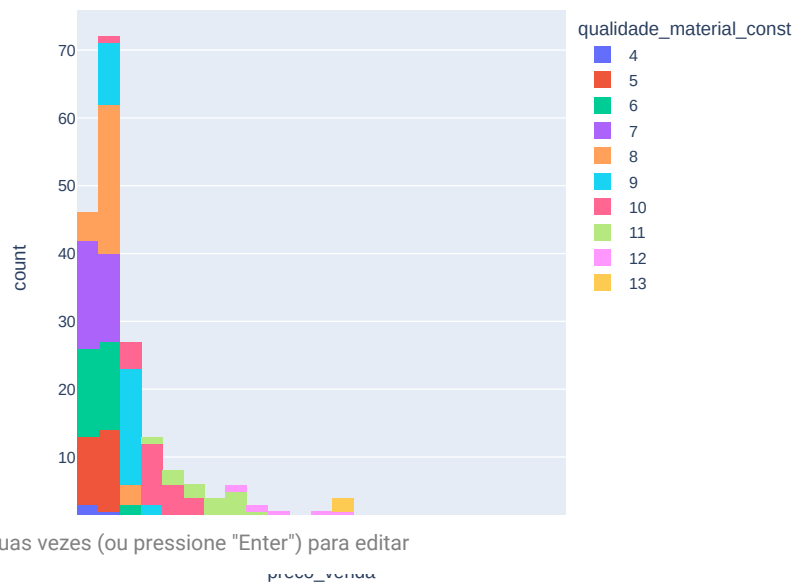
70 rows × 3 columns

```
# Com o histograma construido no plotly, facilitou a observação de concentração de imóveis pela média de preço
# em cada cidade do condado, optei pela média a princípio, pois me dá a possibilidade de
# comparar a média de preço por cidade
px.histogram(median_preco, x='preco_venda', color='cidade')
```



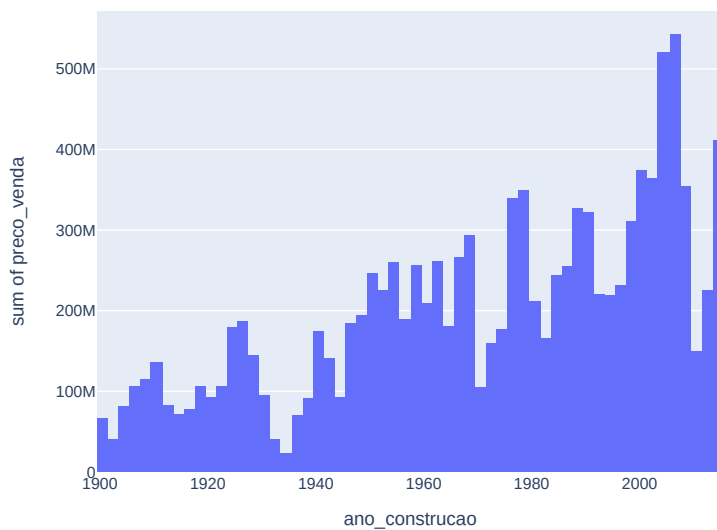
```
# Analisando primeiramente Seattle, por ser um grande centro urbano, oferecendo muitas possibilidades
# que atraem os estadunidenses
house_seattle = house_data.loc[house_data.cidade=='Seattle']
agrupa_seattle = house_seattle.groupby(['qualidade_material_const', 'zipcode']).agg({'preco_venda': 'min'}).reset_index()

# aqui busco onde tem a maior concentração de imóveis com preços atraentes em Seattle
px.histogram(agrupa_seattle, x='preco_venda', color='qualidade_material_const')
```



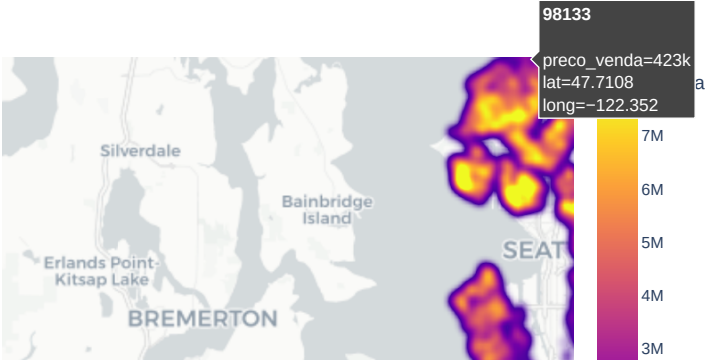
```
# Analisando preços com Histograma com base no ano de construção, que acaba influenciando nos preços
histograma = px.histogram(house_data, x='ano_construcao', y='preco_venda',
                           title='Histograma de preços de Imóveis no Condado de Kings')
histograma.show()
```

Histograma de preços de Imóveis no Condado de Kings



```
# O Mapa de Calor me dá uma noção da área de maior concentração de imóveis
# com custos dentro da faixa buscada, dando uma visão melhor para
# filtrar os dados desejados. O mapa exibe o zipcode, com este filtro a região.
mapa_calor = px.density_mapbox(house_data, lat='lat', lon='lon', z='preco_venda', radius=10,
                                center=dict(lat=47.5, lon=-122.3), zoom=9,
                                hover_name='zipcode', mapbox_style='carto-positron',
                                title='Mapa de calor de preços de imóveis no condado de King')
mapa_calor.show()
```


Mapa de calor de preços de imóveis no condado de King



```
novo_house = pd.DataFrame(agrupa_house)
```



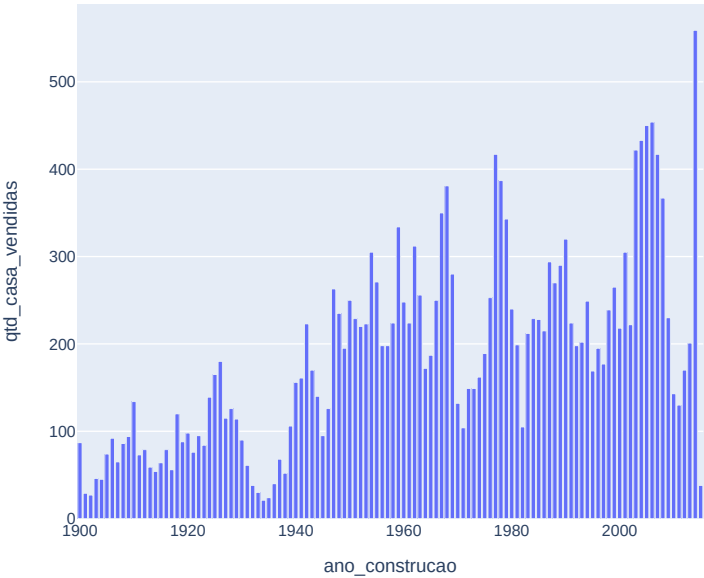
```
# Buscando oportunidade através do zipcode que aparece em regiões próximas do mar
imovel_praia = novo_house[((novo_house.zipcode==98199) | (novo_house.zipcode==98034)) & (novo_house.imovel_b
imovel_praia.groupby(['zipcode','qualidade_material_const', 'preco_venda','imovel_beira_mar','cidade']).size()
```

	zipcode	qualidade_material_const	preco_venda	imovel_beira_mar	cidade	Qtd_Imovel
0	98034	8	1595000	1	Kirkland	1
1	98034	8	1955000	1	Kirkland	1
2	98034	8	3070000	1	Kirkland	1
3	98034	10	3000000	1	Kirkland	1
4	98034	11	2150000	1	Kirkland	1
5	98034	12	3120000	1	Kirkland	1
6	98199	11	1875000	1	Seattle	1

```
# busco a quantidade de casas vendidas pelo ano de construção
df_ano_construcao = novo_house.groupby(['ano_construcao']).size().reset_index(name='qtd_casa_vendidas')
```

Quantidade de casas vendidas com ano de construção até 1989: 12334
Quantidade de casas vendidas com ano de construção a partir de 1990: 9279

```
# Gráfico de barras para ter uma visão mais clara do comportamento de compra de imóveis antigas e novos
px.bar(df_ano_construcao, y='qtd_casa_vendidas', x='ano_construcao')
```



```
# 0 número de casas vendidas com ano de construção inferior a 1980 é relevante
# desta forma não colocarei restrição de busca por ano
soma_casas_1900_1989 = df_ano_construcao[df_ano_construcao.ano_construcao < 1980]
soma_casas_1990_2015 = df_ano_construcao[df_ano_construcao.ano_construcao >= 1980]
print(f'Quantidade de casas vendidas com ano de construção até 1979: {sum(soma_casas_1900_1989.qtd_casa_vendi
print(f'Quantidade de casas vendidas com ano de construção a partir de 1990: {sum(soma_casas_1990_2015.qtd_ca

# Com base em dados do interesse de americanos em imóveis, desenvolvi um filtro
# para buscar oportunidades localizadas nas análises acima
house_filter = novo_house[
    (novo_house.media_area_habitavel_15_prox < 6000) &
    (novo_house.qualidade_material_const > 7) &
    (novo_house.t_area_habitavel < 6000) &
    (novo_house.num_banheiro > 1) &
    (novo_house.condicao_casa_1_5 > 3) &
    (novo_house.preco_venda < 2000000) &
    (novo_house.grau_bela_vista > 2) &
    ((novo_house.cidade == 'Seattle')|(novo_house.cidade == 'Kirkland') | (novo_house.cidade == 'Bellevue'))|

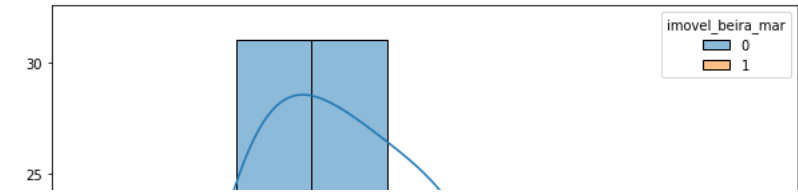
]
house_filter
```

	id	data_venda	preco_venda	num_quarto	num_banheiro	t_area_habitavel	tamanho_terreno	qtd_andare:
58	7922800400	2014-08-27	951000	5	3.25	3250	14342	2.0
60	1516000055	2014-12-10	650000	3	2.25	2150	21235	1.0
294	9297300045	2014-07-09	550000	3	2.00	1970	4166	2.0
551	1373800295	2014-10-13	1450000	3	3.00	4380	6320	2.0
673	1959701890	2014-07-29	865000	4	1.75	1800	4180	2.0
...
19053	9322800210	2014-05-20	879950	4	2.25	3500	13875	1.0
19068	6613001241	2014-08-11	1415000	4	3.00	3110	4408	2.0
19468	3271800870	2014-08-07	1225000	4	2.25	2020	5800	1.0
19562	3271800910	2014-07-01	1356920	4	3.50	4270	5800	2.0
19600	284000025	2015-04-20	1410000	2	2.00	2180	18525	1.0

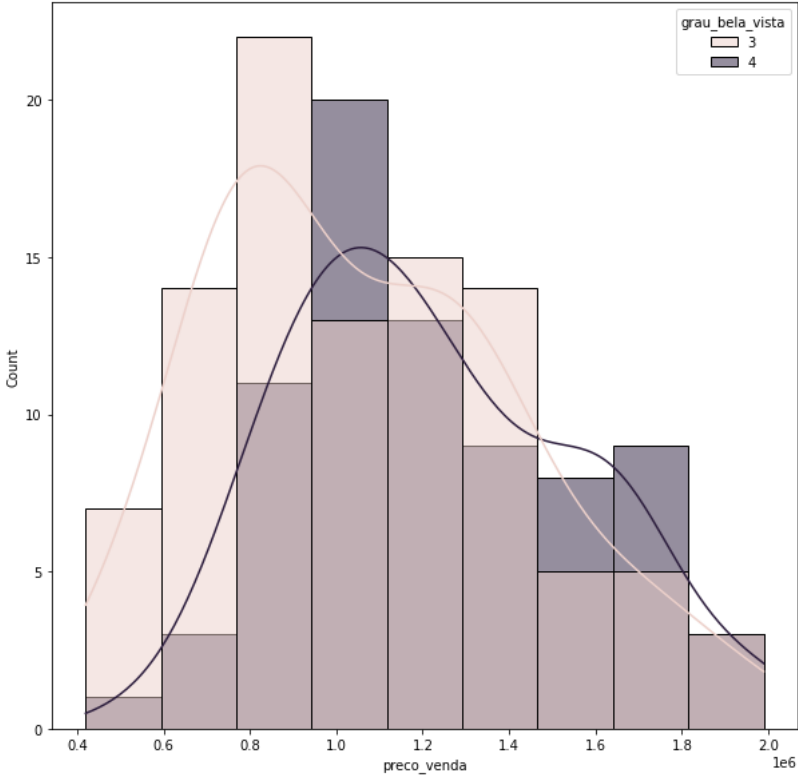
175 rows × 22 columns



```
fig = plt.figure(figsize=(10,10))
ghistograma = sns.histplot(data=house_filter,hue='imovel_beira_mar', x='preco_venda', kde=True)
fig.show()
```



```
fig = plt.figure(figsize=(10,10))
ghistograma = sns.histplot(data=house_filter,hue='grau_bela_vista', x='preco_venda', kde=True)
fig.show()
```



```
house_filter.describe()
```

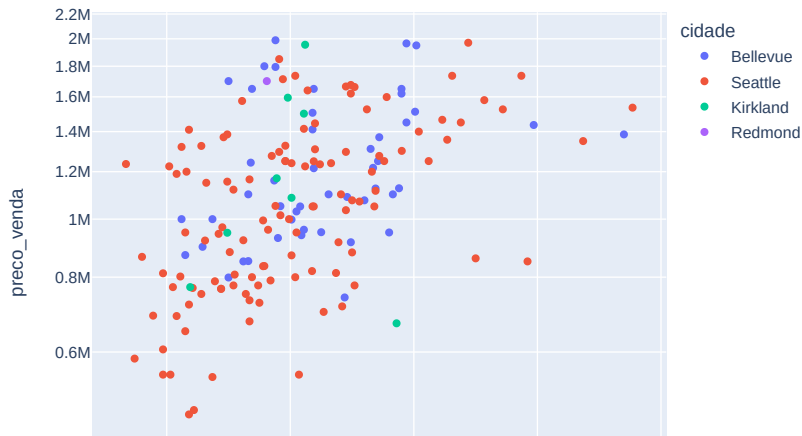
	id	preco_venda	num_quarto	num_banheiro	t_area_habtabavel	tamanho_terreno	qtd_andares	imovel_be
count	6.000000e+00	6.000000e+00	6.000000	6.000000	6.000000	6.000000	6.000000	
mean	3.897384e+09	1.277667e+06	3.500000	2.625000	2855.000000	11166.833333	1.666667	
std	2.865922e+09	2.159784e+05	0.547723	0.518411	105.971694	4690.934253	0.516398	
min	8.680020e+08	9.950000e+05	3.000000	2.000000	2690.000000	7861.000000	1.000000	
25%	2.845475e+09	1.190000e+06	3.000000	2.312500	2797.500000	8470.000000	1.250000	
50%	3.135451e+09	1.262500e+06	3.500000	2.625000	2870.000000	9220.000000	2.000000	
75%	3.914201e+09	1.313250e+06	4.000000	2.750000	2942.500000	11485.000000	2.000000	
max	9.322800e+09	1.650000e+06	4.000000	3.500000	2960.000000	20240.000000	2.000000	

```
house_filter.shape
```

(175, 22)

```
# Criar grafico de dispersao para verificar a relacao entre preco e tamanho do imove e suas tendencias
gdispersao = px.scatter(house_filter, x='t_area_habitavel',
                        y='preco_venda',color='cidade', hover_name='grau_bela_vista',
                        title='Preco por Tamanho do imovel', log_y=True)
gdispersao.show()
```

Preço por Tamanho do imóvel



```
# Devido a um ponto fora da curva, no gráfico de dispersão entendi que poderia
# se tratar de uma grande oportunidade, mas busco verificar se não teria sido erro de digitação
house_teste = house_data[(house_data.t_area_habitavel==13540) & (house_data.qualidade_material_const > 10)]
house_teste
```

	id	data_venda	preco_venda	num_quarto	num_banheiro	t_area_habitavel	tamanho_terreno	qtd_andare:
12777	1225069038	2014-05-05	2280000	7	8.0	13540	307752	3.0

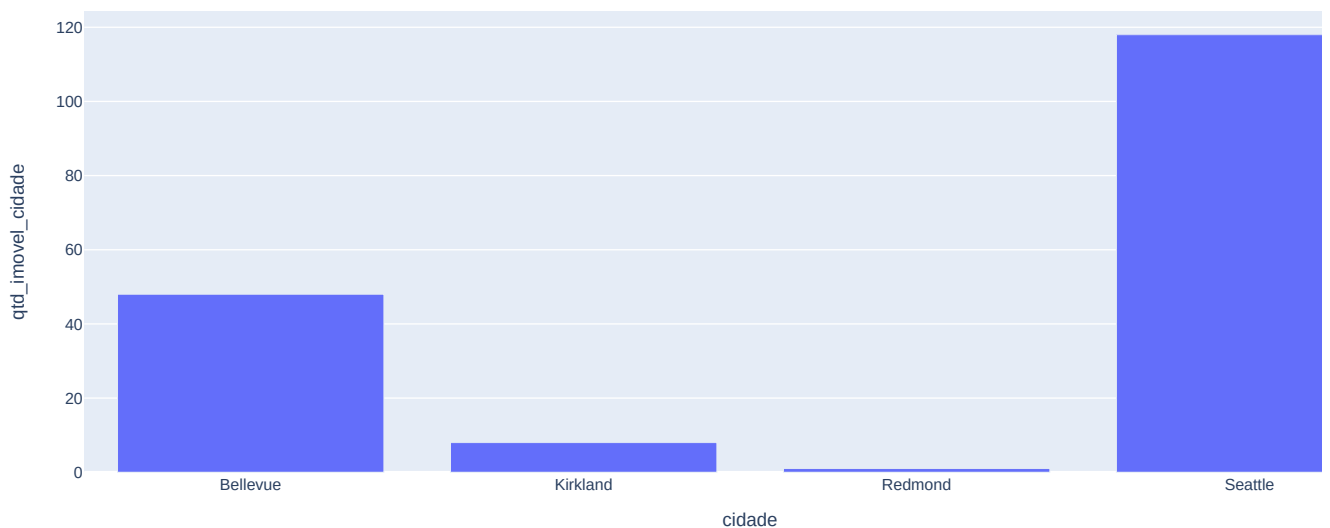
1 rows × 22 columns

Warning: Total number of columns (22) exceeds max_columns (20) limiting to first (20) columns.

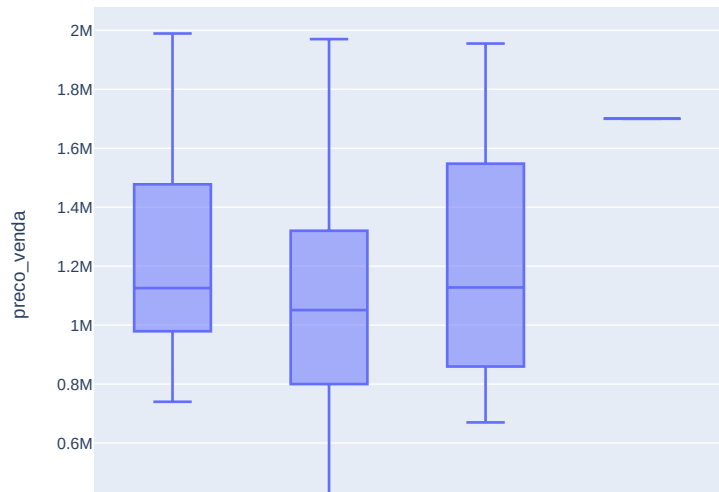
```
qtd_imovel_cidade = house_filter.groupby('cidade').size().reset_index(name='qtd_imovel_cidade')
```

```
gbarras = px.bar(qtd_imovel_cidade, x='cidade', y='qtd_imovel_cidade', title='Numero de imoveis a venda, no c
gbarras.show()
```

Numero de imoveis a venda, no condado de Kings por cidade



```
gboxplot = px.box(house_filter, x='cidade', y='preco_venda')
gboxplot.show()
```



```
top_5_indicacao_mar = house_filter[
    (house_filter.preco_venda < 800000) &
    (house_filter.imovel_beira_mar == 1)
]
```

	id	data_venda	preco_venda	num_quarto	num_banheiro	t_area_habitavel	tamanho_terreno	qtd_andares
11167	7567600030	2015-01-27	750000	5	1.75	2640	13290	1.0
11844	2623039082	2015-02-18	770000	3	3.50	2050	21744	2.0

2 rows × 22 columns



```
top_5_indicacao_vista = house_filter[
    (house_filter.preco_venda < 1500000) &
    (house_filter.ano_construcao >= 1980) &
    (house_filter.imovel_beira_mar == 0) &
    (house_filter.qualidade_material_const > 9)
]
```

```
top_5_indicacao_vista.sort_values(['preco_venda', 'qualidade_material_const'], ascending=[True, False])
```

1 to 8 of 8 entries Filter										
index	id	data_venda	preco_venda	num_quarto	num_banheiro	t_area_habitavel	tamanho_terreno	qtd_andares	imovel_beira_mar	grau_bel
16527	7856640460	2014-12-18	950000	4	2.75	3800	12200	2.0	0	
9861	7805450560	2014-08-20	960000	4	2.5	3110	11397	2.0	0	
13470	4104500191	2014-05-22	1170000	3	2.75	2890	12130	2.0	0	
8328	2600020330	2014-08-13	1218000	4	2.75	3670	15400	2.0	0	
13163	9322800230	2014-12-12	1250000	4	2.5	2960	20240	2.0	0	
13770	1795800040	2014-09-03	1350000	4	3.25	5370	20388	2.0	0	
18991	7856400300	2014-07-02	1411600	2	2.5	3180	9400	2.0	0	
6390	4139430250	2015-03-30	1436000	4	3.5	4970	16582	2.0	0	

Show 25 per page

Like what you see? Visit the [data table notebook](#) to learn more about interactive tables.

Warning: Total number of columns (22) exceeds max_columns (20) limiting to first (20) columns.

```
# Imóveis que podem ser um péssimo negócio para se investir
top_5_menos_indicado = novo_house[
    (novo_house.media_area_habitavel_15_prox <= 6000) &
    (novo_house.qualidade_material_const < 7) &
    (novo_house.t_area_habitavel <= 6000) &
    (novo_house.num_banheiro > 1) &
    (novo_house.condicao_casa_1_5 < 3) &
```

```
((novo_house.cidade == 'Seattle')|(novo_house.cidade == 'Kirkland') | (novo_house.cidade == 'Bellevue'))
]
top_5_menos_indicado.sort_values(['preco_venda'], ascending=False)
```

solo	area_baixo_solo	ano_construcao	ano_reforma	zipcode	lat	long	media_area_habitavel_15_prox	media_area
1850	0	1900	0	98107	47.6711	-122.386		1360
1440	0	1921	0	98118	47.5503	-122.285		1360
1730	450	1950	0	98005	47.5890	-122.177		2180
1580	0	1906	0	98144	47.5848	-122.302		1190
1550	0	1925	0	98166	47.4693	-122.349		1190
720	0	1924	0	98126	47.5840	-122.375		2600
1380	1000	1918	0	98118	47.5294	-122.279		1720
920	1000	1914	0	98136	47.5507	-122.381		1120
1430	0	1957	0	98118	47.5388	-122.275		1550
850	190	1950	0	98118	47.5293	-122.272		1350
590	0	1947	0	98118	47.5450	-122.278		1360