Data Science - Regressão Linear II

1.2 Conhecendo o Dataset

Importando a biblioteca pandas

https://pandas.pydata.org/

In [1]:
```
import pandas as pd
```

O Dataset e o Projeto

Descrição:

O mercado imobiliário vem sendo objeto de diversos estudos e pesquisas nos últimos tempos. A crise financeira que afeta a economia tem afetado significativamente os investimentos e ganhos advindos deste setor. Este cenário incentiva o aumento do interesse por estudos de previsão de demanda baseados em características deste mercado, dos imóveis e do entorno destes imóveis.

Neste contexto o objetivo principal do nosso projeto é desenvolver um sistema de avaliação imobiliária utilizando a metodologia de regressões lineares que é uma das técnicas de machine learning.

Nosso *dataset* é uma amostra aleatória de tamanho 5000 de imóveis disponíveis para venda no município do Rio de Janeiro.

Dados:

- **Valor** - Valor (R$) de oferta do imóvel
- **Area** - Área do imóvel em m^2
- **Dist_Praia** - Distância do imóvel até a praia (km) (em linha reta)
- **Dist_Farmacia** - Distância do imóvel até a farmácia mais próxima (km) (em linha reta)

Leitura dos dados

In [2]:
```
dados = pd.read_csv('../Dados/dataset.csv', sep=';')
```

Visualizar os dados

In [3]:
```
dados.head()
```

Out[3]:

	Valor	Area	Dist_Praia	Dist_Farmacia
0	4600000	280	0.240925	0.793637
1	900000	208	0.904136	0.134494
2	2550000	170	0.059525	0.423318
3	550000	100	2.883181	0.525064
4	2200000	164	0.239758	0.192374

Verificando o tamanho do dataset

In [4]:
```
dados.shape
```

Out[4]:
```
(5000, 4)
```

1.3 Análises Preliminares

Estatísticas descritivas

In [5]:
```
dados.describe().round(2)
```

Out[5]:

	Valor	Area	Dist_Praia	Dist_Farmacia
count	5000.00	5000.00	5000.00	5000.00
mean	1402926.39	121.94	3.02	0.50
std	1883268.85	90.54	3.17	0.29
min	75000.00	16.00	0.00	0.00
25%	460000.00	70.00	0.44	0.24
50%	820000.00	93.00	1.48	0.50
75%	1590000.00	146.00	5.61	0.75
max	25000000.00	2000.00	17.96	1.00

Matriz de correlação

O **coeficiente de correlação** é uma medida de associação linear entre duas variáveis e situa-se entre **-1** e **+1** sendo que **-1** indica associação

negativa perfeita e **+1** indica associação positiva perfeita.

In [6]:
```
dados.corr().round(4)
```

Out[6]:

	Valor	Area	Dist_Praia	Dist_Farmacia
Valor	1.0000	0.7110	-0.3665	-0.0244
Area	0.7110	1.0000	-0.2834	-0.0310
Dist_Praia	-0.3665	-0.2834	1.0000	0.0256
Dist_Farmacia	-0.0244	-0.0310	0.0256	1.0000

2.1 Comportamento da Variável Dependente (Y)

Importando biblioteca seaborn

https://seaborn.pydata.org/

> O Seaborn é uma biblioteca Python de visualização de dados baseada no matplotlib. Ela fornece uma interface de alto nível para desenhar gráficos estatísticos.

In [7]:
```
import seaborn as sns
```

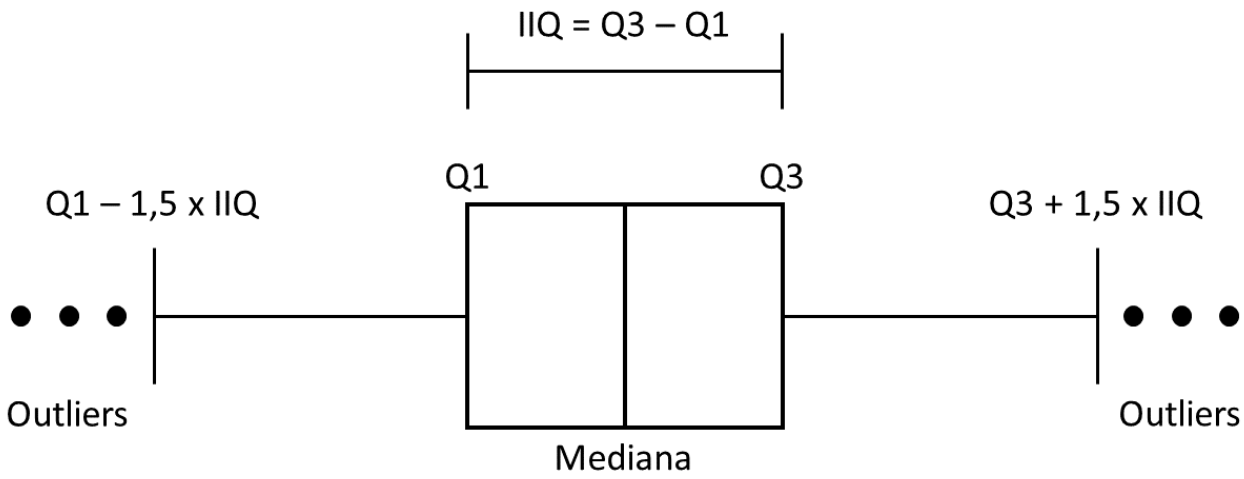
Configurações de formatação dos gráficos

In [8]:
```
# palette -> Accent, Accent_r, Blues, Blues_r, BrBG, BrBG_r, BuGn, BuGn_r, BuPu, BuPu_r, CMRmap

# style -> white, dark, whitegrid, darkgrid, ticks

sns.set_palette("Accent")
sns.set_style("darkgrid")
```

Box plot da variável *dependente* (y)



Box-plot

https://seaborn.pydata.org/generated/seaborn.boxplot.html?highlight=boxplot#seaborn.boxplot

In [9]:
```
ax = sns.boxplot(data = dados.Valor, orient = 'h', width = 0.3)
ax.figure.set_size_inches(20, 6)
ax.set_title('Preço dos Imóveis', fontsize=20)
ax.set_xlabel('Reais', fontsize=16)
ax
```

Out[9]:
```
<AxesSubplot:title={'center':'Preço dos Imóveis'}, xlabel='Reais'>
```

2.2 Distribuição de Frequências

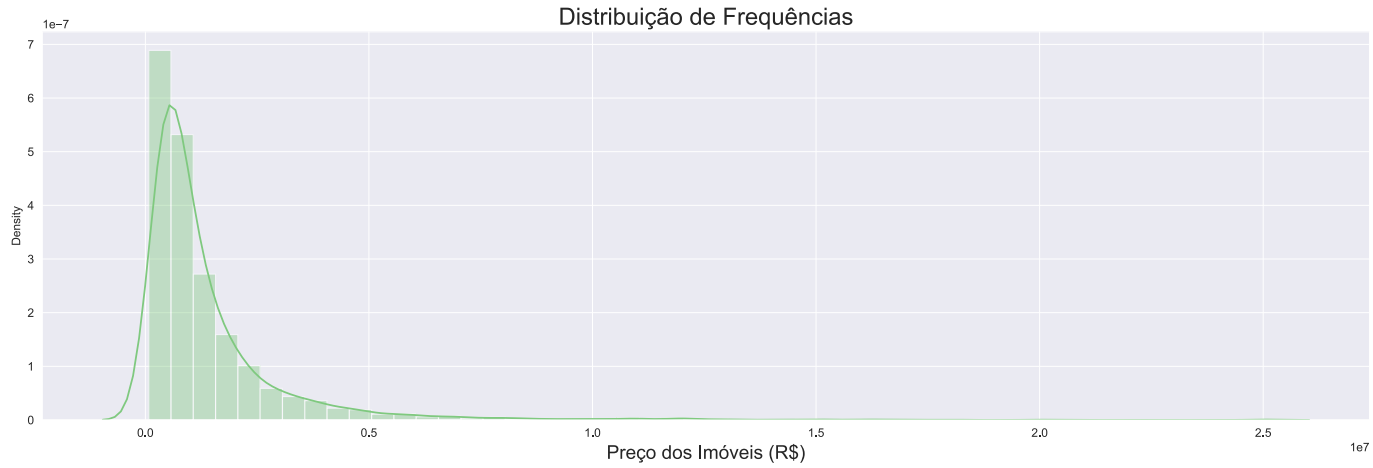
Distribuição de frequências da variável *dependente* (y)

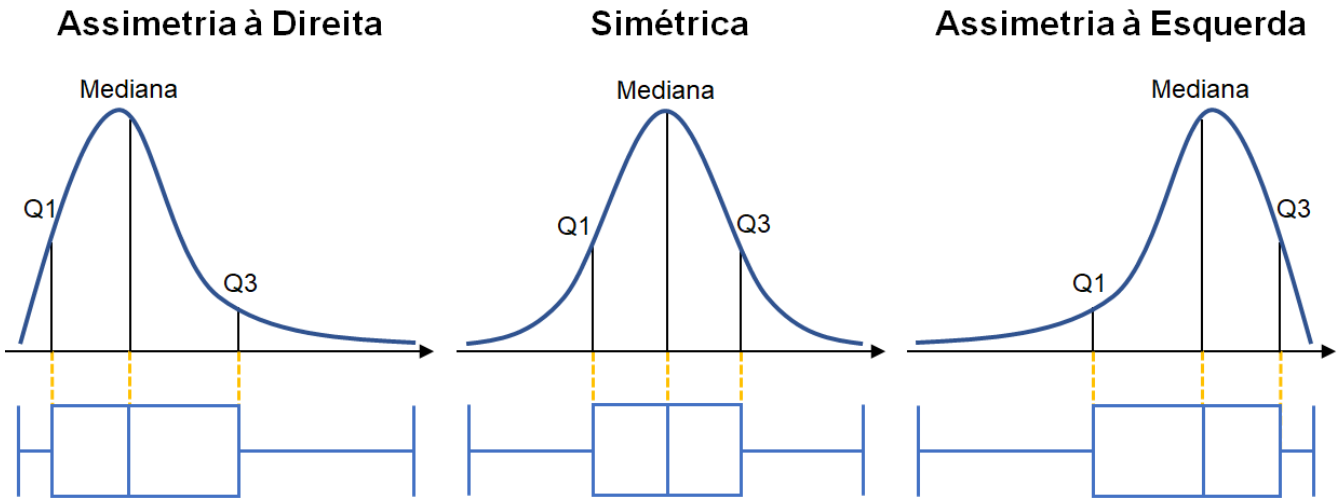
https://seaborn.pydata.org/generated/seaborn.distplot.html?highlight=distplot#seaborn.distplot

In [10]:
```
ax = sns.distplot(dados.Valor)
ax.figure.set_size_inches(20, 6)
ax.set_title('Distribuição de Frequências', fontsize=20)
ax.set_xlabel('Preço dos Imóveis (R$)', fontsize=16)
ax
```

Out[10]:
```
<AxesSubplot:title={'center':'Distribuição de Frequências'}, xlabel='Preço dos Imóveis (R$)', y
```

```
label='Density'>
```





2.3 Dispersão Entre as Variáveis

Gráficos de dispersão entre as variáveis do dataset

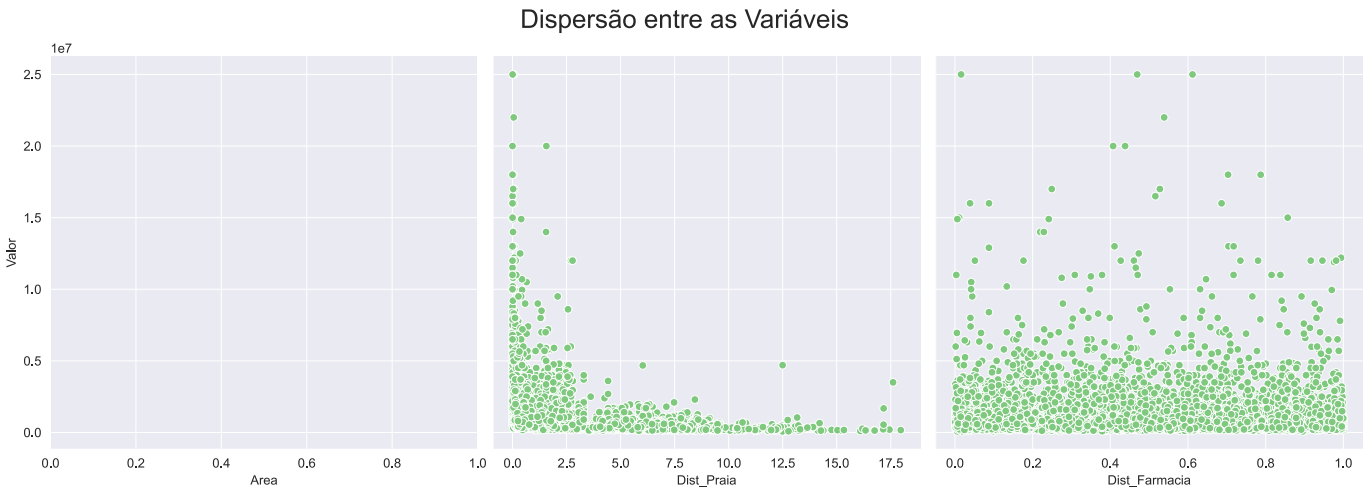
seaborn.pairplot

https://seaborn.pydata.org/generated/seaborn.pairplot.html?highlight=pairplot#seaborn.pairplot

Plota o relacionamento entre pares de variáveis em um dataset.

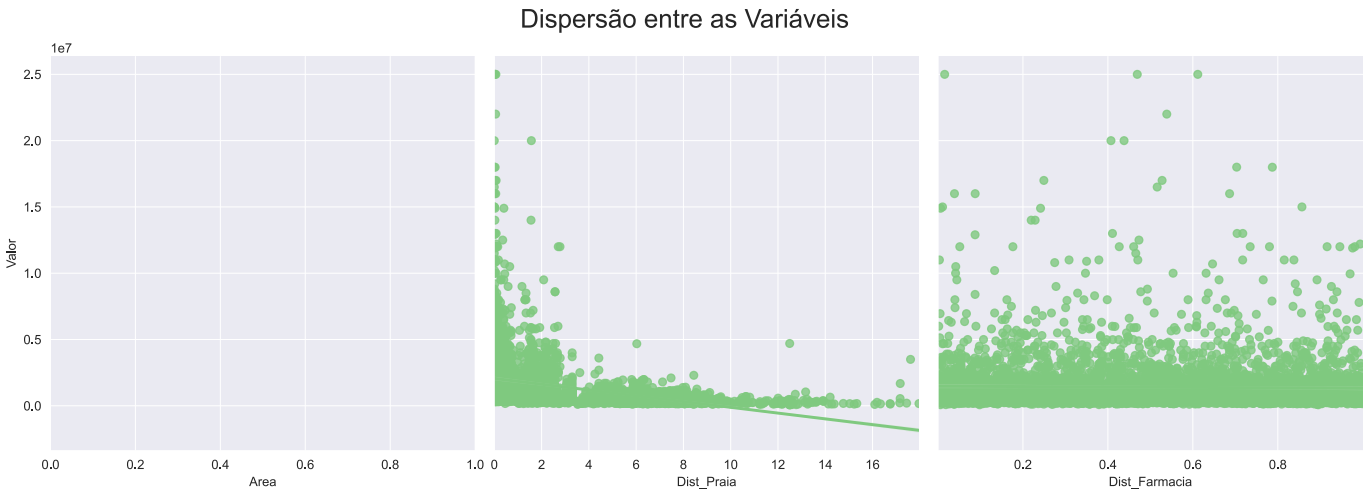
In [11]:
```
ax = sns.pairplot(data = dados, y_vars = 'Valor', x_vars = ['Area', 'Dist_Praia', 'Dist_Farmaci
ax.fig.suptitle('Dispersão entre as Variáveis', fontsize=20, y=1.05)
ax
```

Out[11]:
```
<seaborn.axisgrid.PairGrid at 0xb5c7d60>
```



In [12]:
```
ax = sns.pairplot(data = dados, y_vars = 'Valor', x_vars = ['Area', 'Dist_Praia', 'Dist_Farmaci
ax.fig.suptitle('Dispersão entre as Variáveis', fontsize=20, y=1.05)
ax
```

Out[12]: <seaborn.axisgrid.PairGrid at 0xb1e25b0>



3.1 Transformando os Dados

Distribuição Normal

Por quê?

Testes paramétricos assumem que os dados amostrais foram coletados de uma população com distribuição de probabilidade conhecida. Boa parte dos testes estatísticos assumem que os dados seguem uma distribuição normal (t de Student, intervalos de confiança etc.).

Importando biblioteca numpy

In [13]:
```
import numpy as np
```

Aplicando a transformação logarítmica aos dados do *dataset*

https://docs.scipy.org/doc/numpy-1.15.0/reference/generated/numpy.log.html

In [14]:
```
dados['log_Valor'] = np.log(dados.Valor)
dados['log_Area'] = np.log(dados.Area)
dados['log_Dist_Praia'] = np.log(dados.Dist_Praia + 1)
dados['log_Dist_Farmacia'] = np.log(dados.Dist_Farmacia + 1)
```

In [15]:
```
dados.head(5)
```

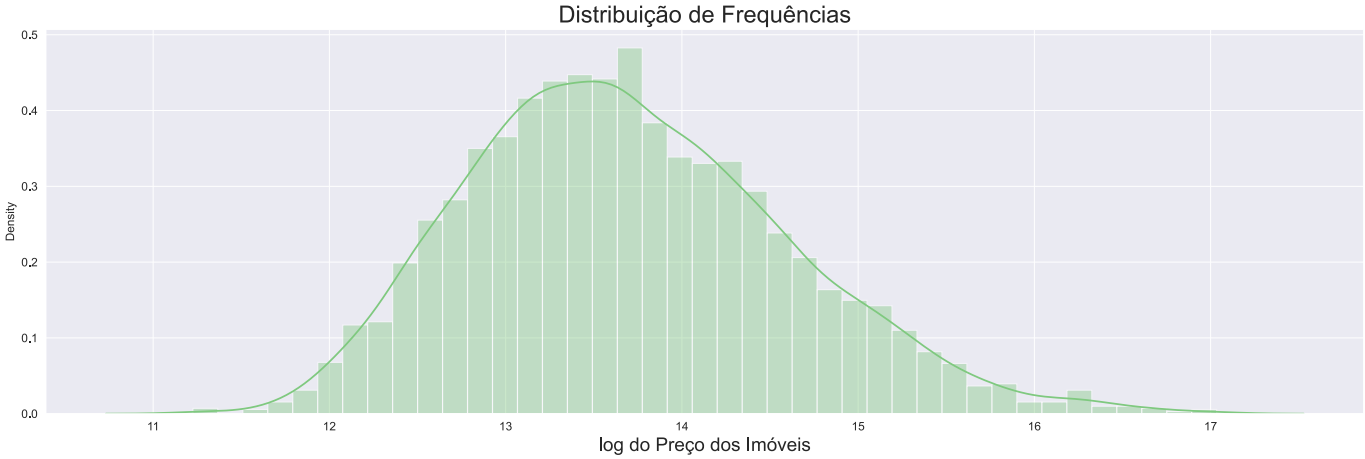
Out[15]:

	Valor	Area	Dist_Praia	Dist_Farmacia	log_Valor	log_Area	log_Dist_Praia	log_Dist_Farmacia
0	4600000	280	0.240925	0.793637	15.341567	5.634790	0.215857	0.584245
1	900000	208	0.904136	0.134494	13.710150	5.337538	0.644028	0.126187
2	2550000	170	0.059525	0.423318	14.751604	5.135798	0.057821	0.352991
3	550000	100	2.883181	0.525064	13.217674	4.605170	1.356655	0.422036
4	2200000	164	0.239758	0.192374	14.603968	5.099866	0.214916	0.175946

Distribuição de frequências da variável *dependente transformada* (y)

In [16]:
```
ax = sns.distplot(dados.log_Valor)
ax.figure.set_size_inches(20, 6)
ax.set_title('Distribuição de Frequências', fontsize=20)
ax.set_xlabel('log do Preço dos Imóveis', fontsize=16)
ax
```

Out[16]:
```
<AxesSubplot:title={'center':'Distribuição de Frequências'}, xlabel='log do Preço dos Imóveis',
ylabel='Density'>
```



3.2 Verificando Relação Linear

Gráficos de dispersão entre as variáveis transformadas do dataset

In [17]:
```
ax = sns.pairplot(dados, y_vars = 'log_Valor', x_vars = ['log_Area', 'log_Dist_Praia', 'log_Dis
ax.fig.suptitle('Dispersão entre as Variáveis Transformadas', fontsize=20, y=1.05)
ax
```

Out[17]: <seaborn.axisgrid.PairGrid at 0xc691a00>