

# Tratamento e Análise de Dados - Vendas

Gleimon Cordeiro

## ▼ Importando a biblioteca pandas

<https://pandas.pydata.org/>

```
import pandas as pd
```

## O Dataset e o Projeto

---

### Descrição:

Para este estudo é utilizado o dataset "base\_teste" com dados referentes as operações de vendas fictícias.

É utilizado no estudo Linguagem Python, Pandas, Seaborn para gerar os gráficos.

Neste contexto o objetivo principal do nosso projeto é realizar uma análise exploratória e uma apresentação com os insights encontrados.

Esse dataset é uma amostra de dados referentes as operações de vendas fictícias. A base de dados tem alguns valores faltantes e o tamanho da base influencia para algumas análises estatísticas

No Microsoft Excel foram realizados alguns tratamentos nos dados do dataset:

- **anal\_data\_venda** - Criada essa coluna para análise de vendas onde a data de concretização da venda é inferior a data de solicitação do cliente
- **data\_pagto\_trat** - Criada essa coluna, pois algumas datas de pagamento estavam com data configuradas no sistema inglês.
- **valor\_venda** - Realizado tratamento nos valores que estavam formatados com (.) ao invés de (,).
- **pol\_comissao** - Realizado tratamento nos valores que estavam formatados com (.) ao invés de (,).
- **valor\_comissao\_trat** - Criada essa coluna, pois alguns valores de comissão não estavam calculados conforme valor\_venda e pol\_comissao.

## ▼ Leitura dos dados

```
dados = pd.read_excel('base_teste_ad_1 (tratado).xlsx')
```

## ▼ Visualizar os dados

```
dados.head(10)
```

	data_solicitacao	data_venda	anal_data_venda	data_pagto	data_pagto_trat	valor.
0	2021-01-13	2021-01-12	data_venda dúvidosa	2021-02-01 00:00:00	2021-02-01	
1	2021-02-14	2021-02-16	NaN	2021-03-01 00:00:00	2021-03-01	
2	2021-02-17	2021-02-18	NaN	2021-03-01 00:00:00	2021-03-01	
3	2021-03-21	2021-03-25	NaN	2021-04-01 00:00:00	2021-04-01	
4	2021-01-29	2021-01-31	NaN	2021-03-01 00:00:00	2021-03-01	
5	2021-03-02	2021-03-07	NaN	2021-04-01 00:00:00	2021-04-01	
6	2021-01-15	2021-01-14	data_venda dúvidosa	2021-02-01 00:00:00	2021-02-01	
7	2021-02-16	2021-02-18	NaN	2021-03-01 00:00:00	2021-03-01	
8	2021-02-02	2021-01-31	data_venda dúvidosa	2021-03-01 00:00:00	2021-03-01	
9	2021-03-06	2021-03-07	NaN	2021-04-01 00:00:00	2021-04-01	

## ▼ Refinar a visualização dos dados do dataset

```
colunas_filtradas = dados[['data_solicitacao', 'data_venda', 'anal_data_venda', 'data_pagto',  
                           'valor_comissao_trat', 'estado_cliente', 'estado_vendedor', 'pr
```

```
colunas_filtradas
```

	data_solicitacao	data_venda	anal_data_venda	data_pagto_trat	valor_venda	pol
0	2021-01-13	2021-01-12	data_venda dúvidosa	2021-02-01	NaN	
1	2021-02-14	2021-02-16	NaN	2021-03-01	NaN	
2	2021-02-17	2021-02-18	NaN	2021-03-01	NaN	
3	2021-03-21	2021-03-25	NaN	2021-04-01	NaN	
4	2021-01-29	2021-01-31	NaN	2021-03-01	NaN	
...	...	...	...	...	...	
93	2021-06-13	2021-06-23	NaN	NaT	150000.0	
94	2021-05-09	2021-05-20	NaN	2021-06-02	180000.0	
95	2021-02-12	2021-02-18	NaN	2021-03-02	180000.0	

## ▼ Verificando o tamanho do dataset



```
colunas_filtradas.shape
```

```
(98, 11)
```

## ▼ Análises Preliminares

### ▼ Análise datas vendas dúvidosas por Vendedor

```
cruzamento_datas_vendas_duvidosas = pd.crosstab(colunas_filtradas.vendedor_id, colunas_fil  
cruzamento_datas_vendas_duvidosas
```

anal_data_venda	data_venda	dúvidosa	
vendedor_id			
V_1		1	
V_2		1	
V_20		1	
V_22		1	
V_4		1	

### ▼ Quantidade de vendas por produto

```
colunas_filtradas['produto'].value_counts()
```

```
A    20
F    20
D    20
C    18
B    10
E    10
Name: produto, dtype: int64
```

## ▼ Visualização das vendas sem registro de pagamento por Vendedor

```
quant_vendas_sem_pagamento = colunas_filtradas['data_pagto_trat'].isnull().sum()
filtro_vendas_sem_pagamento = colunas_filtradas[colunas_filtradas['data_pagto_trat'].isnull()]
filtro_vendas_sem_pagamento
```

	data_solicitacao	data_venda	anal_data_venda	data_pagto_trat	valor_venda	pol
<b>20</b>	2021-04-25	2021-07-12	NaN	NaT	280000.0	
<b>21</b>	2021-03-24	2021-06-07	NaN	NaT	336000.0	
<b>22</b>	2021-03-23	2021-06-30	NaN	NaT	360000.0	
<b>46</b>	2021-06-02	2021-06-20	NaN	NaT	14500.0	
<b>73</b>	2021-05-19	2021-05-22	NaN	NaT	NaN	
<b>79</b>	2021-05-11	2021-05-25	NaN	NaT	NaN	
<b>93</b>	2021-06-13	2021-06-23	NaN	NaT	150000.0	

```
total = filtro_vendas_sem_pagamento.valor_venda.sum()
total
```

```
1140500.0
```

## ▼ Valor médio de (R\$) venda por Produto

```
valor_medio_produto = pd.crosstab(colunas_filtradas.produto, colunas_filtradas.produto, ag
valor_medio_produto
```

produto	B	C	E
produto			
<b>B</b>	649000.0	NaN	NaN
<b>C</b>	NaN	8457.777778	NaN
<b>E</b>	NaN	NaN	149600.0

## ▼ Análise Vendedor por Produto

```
cruzamento_produtos_vendedor = pd.crosstab(colunas_filtradas.vendedor_id, colunas_filtrada
```

```
cruzamento_produtos_vendedor['Valor Total'] = cruzamento_produtos_vendedor['A'] + cruzamen  
cruzamento_produtos_vendedor['D'] + cruzamento_produtos_vendedor['E'] + cruzamento_produto
```

```
cruzamento_produtos_vendedor
```

19/05/2022 13:16Tratamento e Análise Dados - Vendas.ipynb - Colaboratory

produto	A	B	C	D	E	F	Valor Total
vendedor_id							
V_1	2	0	0	0	0	0	2
V_10	0	0	0	2	0	0	2
V_11	0	0	0	2	0	0	2
V_12	0	0	0	2	0	0	2
V_13	0	0	0	2	0	0	2
V_14	0	0	0	0	2	0	2
V_15	0	0	0	0	2	0	2
V_16	0	0	0	0	0	2	2
V_17	0	0	0	0	0	2	2
V_18	0	0	0	0	0	2	2
V_19	0	0	4	2	0	0	6
V_2	2	0	0	0	0	0	2
V_20	2	0	0	0	0	3	5

cruzamento\_produtos\_vendedor.describe()

produto	A	B	C	D	E	F	Valor Total
count	38.000000	38.000000	38.000000	38.000000	38.000000	38.000000	38.000000
mean	0.526316	0.263158	0.473684	0.526316	0.263158	0.526316	2.578947
std	0.892517	0.601090	1.083958	0.892517	0.723513	1.224454	1.535729
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	2.000000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	2.000000
75%	1.500000	0.000000	0.000000	1.500000	0.000000	0.000000	2.750000
max	2.000000	2.000000	4.000000	2.000000	3.000000	6.000000	8.000000
V_34	0	1	0	0	0	2	3

➤ Quantidade de Produtos Vendidos por Estado

```
cruzamento_produtos_valor = pd.crosstab(colunas_filtradas.estado_vendedor, colunas_filtrad
V_34
cruzamento_produtos_valor['Valor Total'] = cruzamento_produtos_valor['A'] + cruzamento_pro
cruzamento_produtos_valor['D'] + cruzamento_produtos_valor['E'] + cruzamento_produtos_valo
```

cruzamento\_produtos\_valor

produto	A	B	C	D	E	F	Valor Total
estado_vendedor							
AC	0	0	0	2	2	0	4
AL	0	0	0	0	2	1	3
MG	0	2	2	2	0	4	10
PE	0	1	0	0	0	2	3
PR	6	0	0	0	0	0	6
RJ	4	1	2	4	0	9	20
RS	4	2	4	6	3	0	19
SC	4	2	4	2	3	0	15
SP	2	2	6	4	0	4	18

## ▼ Valor médio de venda de cada Produto por Estado

```
cruzamento_produtos_valor_media = pd.crosstab(colunas_filtradas.produto, colunas_filtradas.estado_vendedor)
cruzamento_produtos_valor_media.describe()
```

estado_vendedor	AC	AL	MG	PE	RJ	RS
count	1.0	1.0	2.000000	1.0	2.000000	3.000000e+00
mean	88000.0	132000.0	158812.500000	432000.0	182475.000000	3.970192e+05
std	NaN	NaN	210982.985837	NaN	251058.26266	5.302439e+05
min	88000.0	132000.0	9625.000000	432000.0	4950.000000	8.057500e+04
25%	88000.0	132000.0	84218.750000	432000.0	93712.500000	9.502875e+04
50%	88000.0	132000.0	158812.500000	432000.0	182475.000000	1.820000e+05
75%	88000.0	132000.0	233406.250000	432000.0	271237.500000	5.915000e+05
max	88000.0	132000.0	308000.000000	432000.0	360000.000000	1.001000e+06

## ▼ Quantidade de Produtos Vendidos por Estado do Cliente

```
cruzamento_produtos_cliente = pd.crosstab(colunas_filtradas.estado_cliente, colunas_filtradas.produto)
```

```
cruzamento_produtos_cliente['Valor Total'] = cruzamento_produtos_cliente['A'] + cruzamento_produtos_cliente['D'] + cruzamento_produtos_cliente['E'] + cruzamento_produtos_cliente['F']
```

cruzamento\_produtos\_cliente

produto	A	B	C	D	E	F	Valor	Total
estado_cliente								
BA	0	0	4	0	0	2		6
MG	0	2	0	0	2	0		4
PR	0	2	0	0	2	0		4
RS	2	0	0	2	0	0		4
SC	2	0	0	2	0	0		4
SP	4	0	2	4	0	2		12

## ▼ Análise dos Valores de Venda e Valores de Comissão

colunas\_filtradas.describe().round(2)

	valor_venda	valor_comissao_trat
count	38.00	98.00
mean	214164.21	1578.55
std	305305.62	2590.27
min	2000.00	52.00
25%	8782.50	122.85
50%	88000.00	450.00
75%	264000.00	750.00
max	1092000.00	10920.00

## ▼ Análise por meio de gráficos e verificação do comportamento das variáveis.

### ► Importando biblioteca seaborn

<https://seaborn.pydata.org/>

O Seaborn é uma biblioteca Python de visualização de dados baseada no matplotlib.

[ ] ↳ 1 célula oculta



## ► Configurações de formatação dos gráficos

[ ] ↴ 2 células ocultas

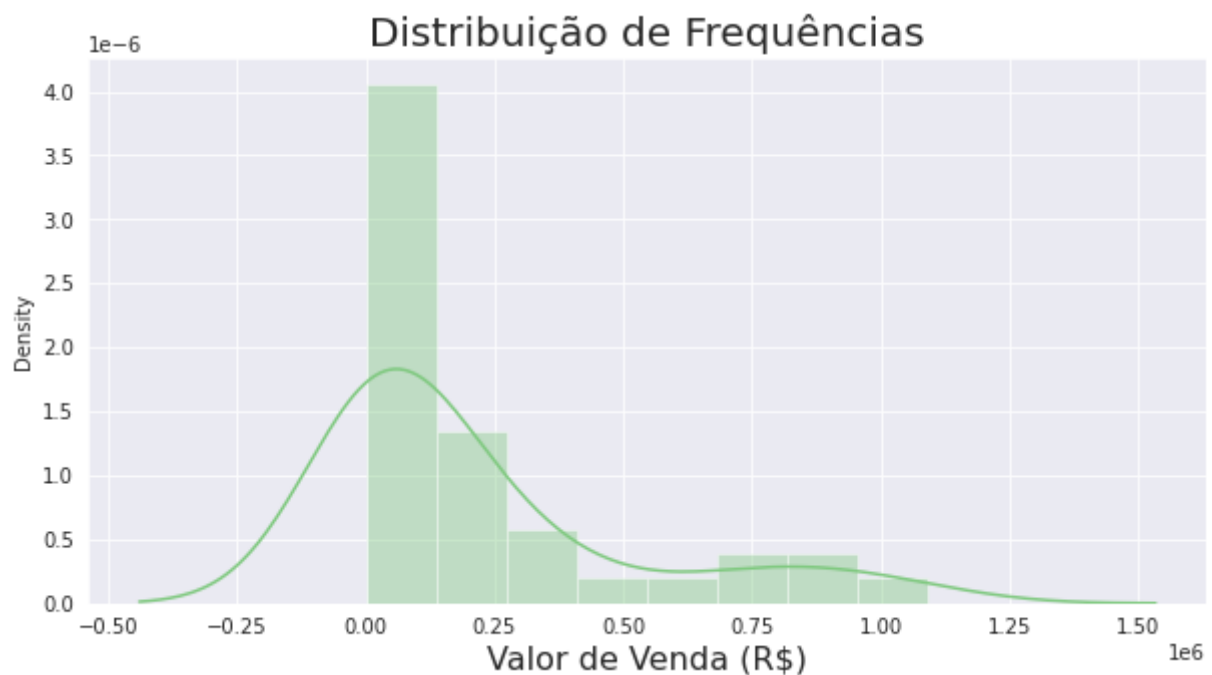
## ▼ Distribuição de Frequências

Distribuição de frequências da variável valor\_venda e valor\_comissao\_trat

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

```
ax = sns.distplot(colunas_filtradas['valor_venda'])
ax.figure.set_size_inches(10, 5)
ax.set_title('Distribuição de Frequências', fontsize=20)
ax.set_xlabel('Valor de Venda (R$)', fontsize=16)
ax
```

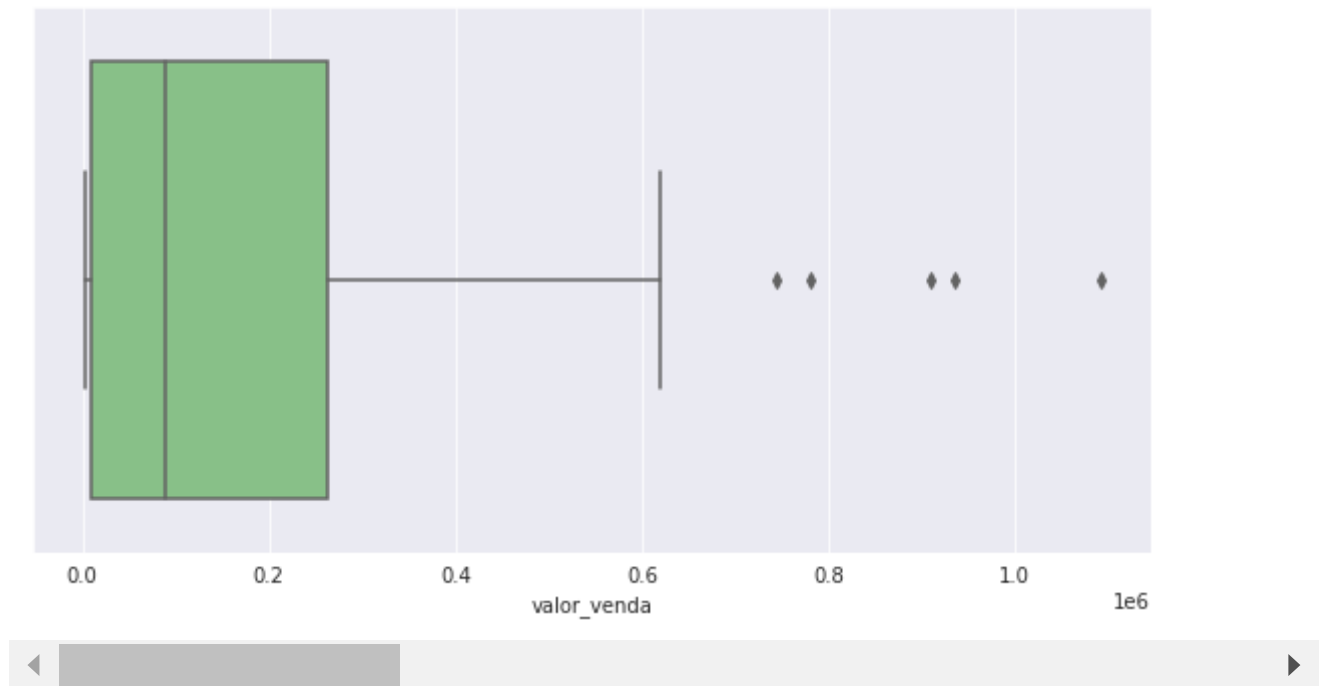
```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f0fbfcd2050>
```



```
axx = sns.boxplot(colunas_filtradas.valor_venda)
```

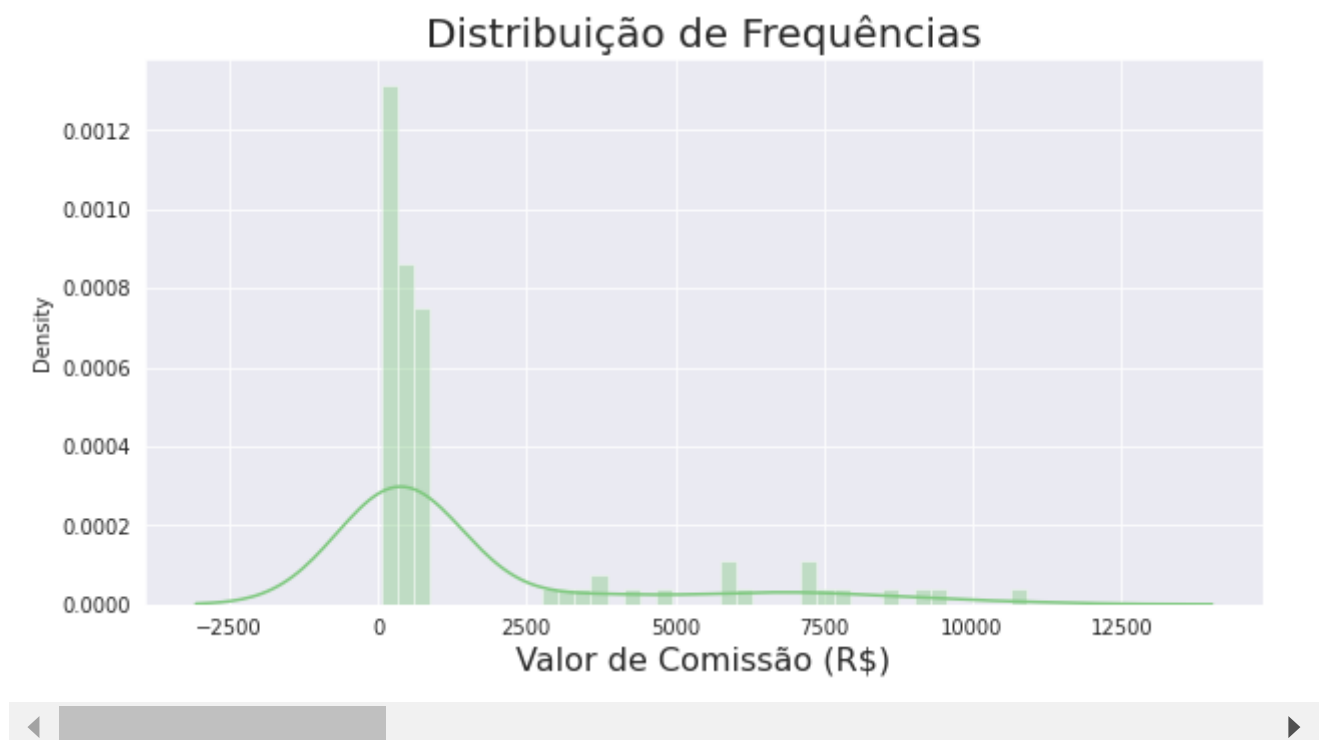
```
axx.figure.set_size_inches(10, 5)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass
FutureWarning
```



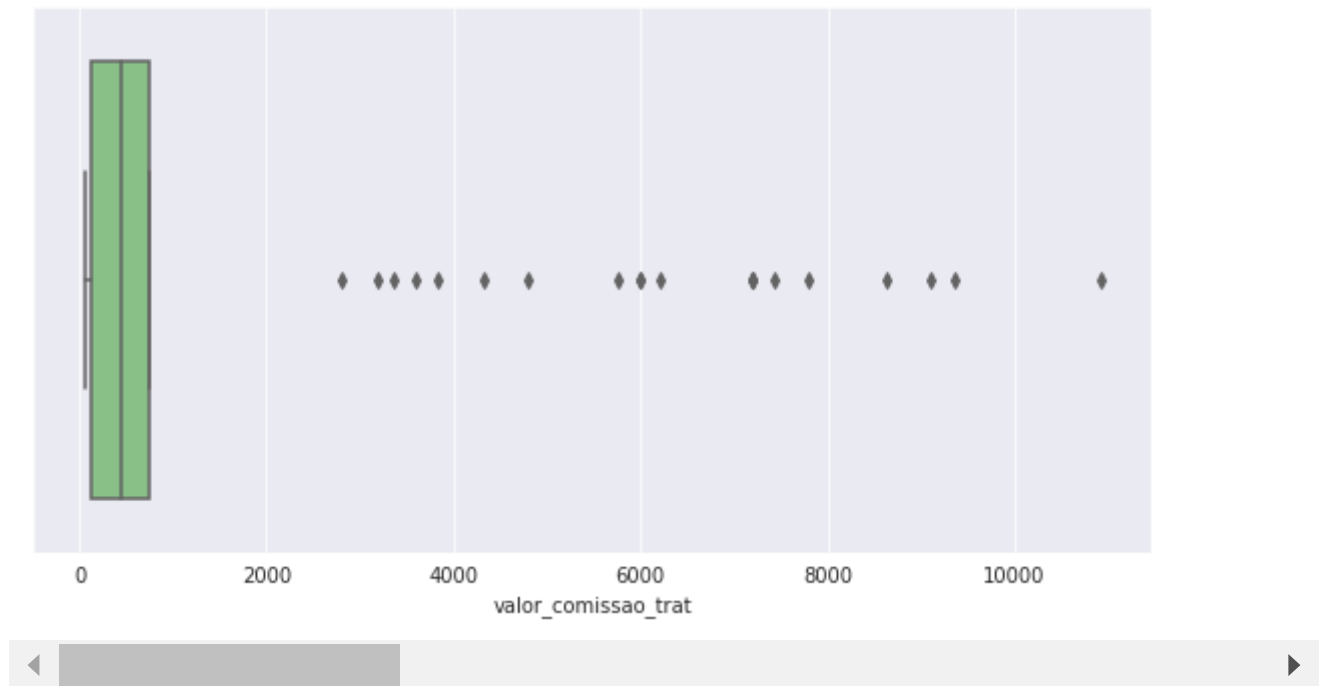
```
ax = sns.distplot(colunas_filtradas['valor_comissao_trat'])
ax.figure.set_size_inches(10, 5)
ax.set_title('Distribuição de Frequências', fontsize=20)
ax.set_xlabel('Valor de Comissão (R$)', fontsize=16)
ax
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:
warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f0fbfccd9d0>
```



```
axx = sns.boxplot(colunas_filtradas.valor_comissao_trat)
axx.figure.set_size_inches(10, 5)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass
FutureWarning
```



## ▼ Dispersão Entre as Variáveis

---

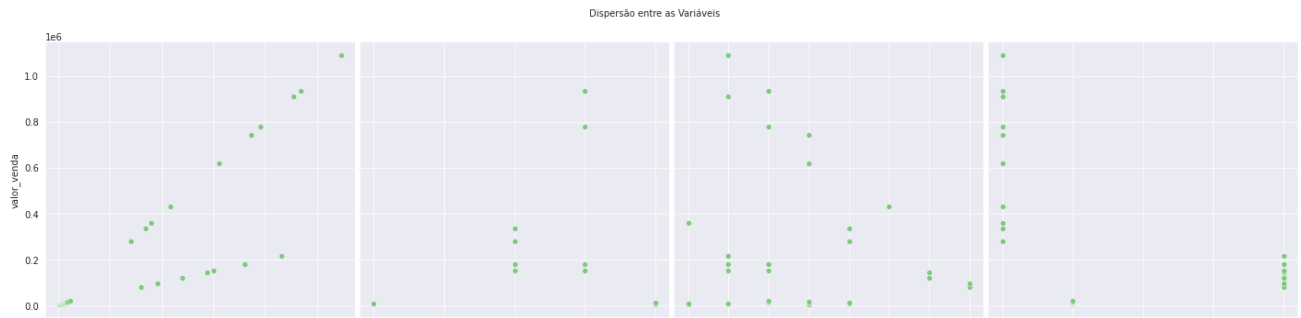
Gráficos de dispersão entre as variáveis do dataset

### ▼ seaborn.pairplot

Plota o relacionamento entre pares de variáveis em um dataset.

```
ax = sns.pairplot(colunas_filtradas, y_vars='valor_venda', x_vars=['valor_comissao_trat'],
ax.fig.suptitle('Dispersão entre as Variáveis', fontsize=10, y=1.05)
ax
```

```
<seaborn.axisgrid.PairGrid at 0x7f0fbc50f9d0>
```



## Conclusão

É possível fazer as seguintes observações:

- No item “Análise datas vendas dúbidas por Vendedor”: é possível verificar que tem 5 registros de vendas onde a data de concretização da venda é inferior a data de solicitação do cliente. Deve ser verificado com os respectivos Vendedores o que ocorreu.
- No item “Quantidade de Vendas por Produto”: é possível verificar que os produtos A, F e D totalizaram 60 registros de venda, ou seja, esses 03 produtos representam mais de 60% das vendas.
- No item “Visualização das vendas sem registro de pagamento por Vendedor”: é possível verificar que tem 07 vendas sem registro de pagamento que resultam em um valor total de R\$1.140.500,00.
- No item “Valor médio de (R\$) venda por Produto”: é possível verificar que o Produto B tem um valor médio de vendas de R\$649.000, o Produto C de R\$8.457,77 e o Produto E de R\$149.600.
- No item “Análise Vendedor por Produto”: é possível verificar que temos 38 vendedores cadastrados na nossa base de vendas. Na média cada vendedor vendeu 2,57 produtos.
- No item “Quantidade de Produtos Vendidos por Estado”: é possível verificar que os Estados RJ, RS e SP totalizaram 57 registros de venda, ou seja, esses 03 Estados representam aproximadamente 60% das vendas.
- No item “Quantidade de Produtos Vendidos por Estado do Cliente”: é possível verificar que os clientes dos Estados SP e BA totalizaram 18 registros de compra, ou seja, esses 02 Estados representam aproximadamente 50% dos registros de compra. Porém, o estado de RS é o que tem maior média de valor por venda R\$397.019,20 conforme pode ser visto no item “Valor médio de venda de cada Produto por Estado”.

- No item “Análise dos Valores de Venda e Valores de Comissão”: é possível verificar a média de valor por venda é de R\$214.164,21 e valor de comissão é R\$1578,55. O maior valor de venda foi R\$1.092.000,00 e o maior valor de comissão foi R\$10.920,00.
- No item “Análise por meio de gráficos e verificação do comportamento das variáveis”: é possível verificar que os valores de vendas e o valores de comissões não são uma distribuição normal. É possível visualizar alguns outliers que influenciam no valor da média quando comparada com a mediana. Por meio do pairplot é possível verificar que a variável comissão tem relação com o valor da venda, estado do vendedor e produto também, porém precisa de mais dados para prosseguir com as análises.

---

✓ 0s conclusão: 12:32

