

Analizando Dados com Python



Como nasceu esse workshop?

- GEDS - Grupo de Estudos em Data Science, integrantes do Pyladies São Paulo
- [Trilha e materiais](#)
- A missão foi estudar Ciência de Dados, disponibilizar o material para a comunidade e incentivar outras mulheres a formarem grupos de estudos de temas de interesse
- Material adaptado [dessa](#) 1a versão do Workshop feito pelas integrantes do GEDS



Agenda

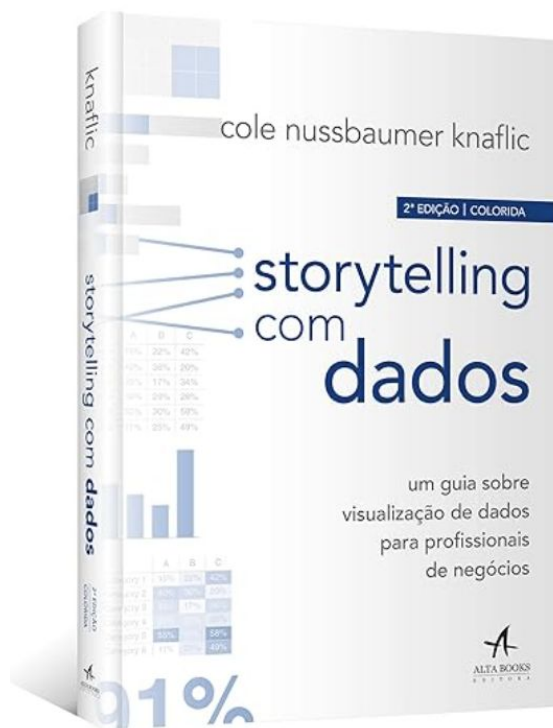
- Contexto
- Tipos de Dados
- Introdução ao pacote Pandas
- Estatística Descritiva
- Variabilidade dos dados
- Introdução à Visualização

Elementos importantes em uma análise de dados:

- Definição de Objetivos
- Limpeza e Preparação dos Dados - qualidade
- Exploração de Dados
- Análise Estatística
- Visualização de Dados
- Interpretação de Resultados
- Comunicação dos Insights - apoio para Tomada de Decisão
- Ética e Privacidade
- Atualização e Monitoramento

Contexto

A análise de dados não é apenas sobre a **coleta e manipulação de números**; é sobre **como usar esses dados para comunicar** informações de maneira significativa e para tomada de decisões.



O Colab segue o mesmo padrão do Jupyter Notebook. Nele é possível adicionar células de código, texto, importar arquivos etc. Para ter acesso ao Google Colab, basta logar na sua conta Google e acessar o link <https://colab.research.google.com>;



Tipos de Dados

Tipos de dados

Quantitativos - Quantifica ou mede

Discretos:

Assumem valores em um conjunto especificado de números.

Contínuos:

Assumem valores em um intervalo contínuo de números.

Qualitativos - Característica ou qualidade

Nominal:

Característica que não possui ordem.

Ordinal:

Característica que possui uma ordem de grandeza.

Tipos de Dados

Quantitativos



Discretos:

- Quantidade de pessoas na sala?
- Quantidade de dias no mês?



Contínuos:

- Qual sua altura?
- Qual a distância daqui até sua casa?

Qualitativos



Nominal:

- Qual seu Estado? (SP, MG, RJ, Outros)
- Qual gênero você se identifica?



Ordinal:

- Avaliação curso (Ruim a Ótimo).
- Qual seu nível de escolaridade?

Tipos de Dados - Exemplo PyLadies

ID	Estado Origem	Idade	Escolaridade	Trabalha como Programadora	Renda Mensal
1	SP	36	4	S	3737,52
2	SP	25	2	N	400,00
3	MG	34	3	S	2366,14
4	RJ	23	3	S	2841,29
5	SP	31	4	N	800
6	SP	34	5	S	3433,02
7	SP	39	5	S	2752,74
8	PE	24	3	S	3682,33
9	RJ	29	3	S	2359,28
10	SP	27	3	S	2119,15
11	SP	30	3	S	3326,79
12	SP	25	4	S	2684,05
13	SP	23	2	S	3507,84
14	SP	16	1	N	0
15	SP	36	4	N	800

Legenda Escolaridade

- | | |
|---|---------------------------|
| 1 | Ensino Medio
Completo |
| 2 | Graduanda |
| 3 | Graduação
Completa |
| 4 | Pós graduanda |
| 5 | Pós graduação
completa |

Dados meramente ilustrativos.



Tipos de Dados - Dados Qualitativos

ID	Estado Origem	Idade	Escolaridade	Trabalha como Programadora	Renda Mensal
1	SP	36	4	S	3737,52
2	SP	25	2	N	400,00
3	MG	34	3	S	2366,14
4	RJ	Dados Qualitativos			2841,29
5	SP				800
6	SP	34	5	S	3433,02
7	SP	39	5	S	2752,74
8	PE	24	3	S	3682,33
9	RJ	29	3	S	2359,28
10	SP	27	3	S	2119,15
11	SP	30	3	S	3326,79
12	SP	25	4	S	2684,05
13	SP	23	2	S	3507,84
14	SP	16	1	N	0
15	SP	36	4	N	800

Legenda Escolaridade

- | | |
|---|------------------------|
| 1 | Ensino Medio Completo |
| 2 | Graduanda |
| 3 | Graduação Completa |
| 4 | Pós graduanda |
| 5 | Pós graduação completa |

Tipos de Dados - Categóricos

ID	Estado Origem	Idade	Escolaridade	Trabalha como Programadora	Renda Mensal
1	SP	36	4	S	3737,52
2	SP	25	2	N	400,00
3	MG	34	3	S	2366,14
4	Nominal	23	Ordinal	Nominal	2841,29
5	SP	31	4	N	800
6	SP	34	5	S	3433,02
7	SP	39	5	S	2752,74
8	PE	24	3	S	3682,33
9	RJ	29	3	S	2359,28
10	SP	27	3	S	2119,15
11	SP	30	3	S	3326,79
12	SP	25	4	S	2684,05
13	SP	23	2	S	3507,84
14	SP	16	1	N	0
15	SP	36	4	N	800

Legenda Escolaridade

- | | |
|---|------------------------|
| 1 | Ensino Medio Completo |
| 2 | Graduanda |
| 3 | Graduação Completa |
| 4 | Pós graduanda |
| 5 | Pós graduação completa |

Tipos de Dados - Dados Quantitativos

ID	Estado Origem	Idade	Escolaridade	Trabalha como Programadora	Renda Mensal	
1	SP	36	4	S	3737,52	
2	SP	25	2	N	400,00	
3	MG	34	3	S	2366,14	
4	RJ	23	Dados Quantitativos			1,29
5	SP	31				00
6	SP	34	5	S	3433,02	
7	SP	39	5	S	2752,74	
8	PE	24	3	S	3682,33	
9	RJ	29	3	S	2359,28	
10	SP	27	3	S	2119,15	
11	SP	30	3	S	3326,79	
12	SP	25	4	S	2684,05	
13	SP	23	2	S	3507,84	
14	SP	16	1	N	0	
15	SP	36	4	N	800	

Legenda Escolaridade

- | | |
|---|------------------------|
| 1 | Ensino Medio Completo |
| 2 | Graduanda |
| 3 | Graduação Completa |
| 4 | Pós graduanda |
| 5 | Pós graduação completa |

Tipos de Dados - Numéricos

ID	Estado Origem	Idade	Escolaridade	Trabalha como Programadora	Renda Mensal
1	SP	36	4	S	3737,52
2	SP	25	2	N	400,00
3	MG	34	3	S	2366,14
4	RJ	Discreto	3	S	Contínuo
5	SP	31	4	N	800
6	SP	34	5	S	3433,02
7	SP	39	5	S	2752,74
8	PE	24	3	S	3682,33
9	RJ	29	3	S	2359,28
10	SP	27	3	S	2119,15
11	SP	30	3	S	3326,79
12	SP	25	4	S	2684,05
13	SP	23	2	S	3507,84
14	SP	16	1	N	0
15	SP	36	4	N	800

Legenda Escolaridade

- | | |
|---|------------------------|
| 1 | Ensino Medio Completo |
| 2 | Graduanda |
| 3 | Graduação Completa |
| 4 | Pós graduanda |
| 5 | Pós graduação completa |

Trabalhando com Pandas

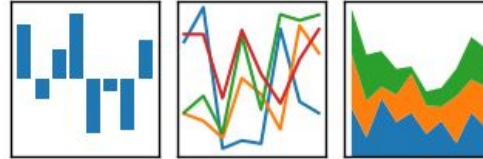
O acrônimo Pandas vem da combinação de *Panel Data* e *Python Data Analysis**.



- Dados de Paineis - Python para Análise de Dados

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



Pandas é uma biblioteca de código aberto que fornece estruturas de dados de alto desempenho e fáceis de usar e ferramentas de análise de dados para a linguagem de programação Python.

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



Primeiro Passo:

1. Abrir as bibliotecas que você utilizará
2. Subir o arquivo que possui seus dados

```
[ ] #abrindo as bibliotecas que serão utilizadas
import pandas as pd
import matplotlib.pyplot as plt
from google.colab import files
uploaded = files.upload()
```



Browse... dados.txt

dados.txt(text/plain) - 379 bytes, last modified: n/a - 100% done
Saving dados.txt to dados.txt

Abrindo o arquivo:

Importando um CSV para o Colab:

CSV - **C**omma-**S**eparated **V**alues

vírgula separando valor

`pd.read_csv('nome_arquivo', sep = ';', decimal = ',')`

```
# Transformando o arquivo importando em um dataframe
dados_pyladies = pd.read_csv('dados.txt', sep=';', decimal = ',')
```

Argumentos

separador

;

decimal

,

Visualizando o arquivo:

`nome_dataframe.head()`

```
#para ver as cinco primeiras linhas  
dados_pyladies.head()
```

	Estado	Origem	Idade	Escolaridade	Trabalha_como_Programadora	Renda_Mensal
0		SP	36	4	S	3737.52
1		SP	25	2	N	400.00
2		MG	34	3	S	2366.14
3		RJ	23	3	S	2841.29
4		SP	31	4	N	800.00

Visualizando o arquivo:

`nome_dataframe.tail()`

```
[22] # para ver as cinco últimas linhas  
dados_pyladies.tail()
```



	Estado	Origem	Idade	Escolaridade	Trabalha_como_Programadora	Renda_Mensal
--	--------	--------	-------	--------------	----------------------------	--------------

10		SP	30	3	S	3326.79
11		SP	25	4	S	2684.05
12		SP	23	2	S	3507.84
13		SP	16	1	N	0.00
14		SP	36	4	N	800.00

O que é um dataframe??



DataFrame é uma estrutura de dados bidimensional - parecida com uma tabela de excel ou um banco de dados.

As Estruturas dos Dados:

Estrutura de dados bidimensional (colunas e linhas) cujo índice começa no **zero**.

O dataframe contém colunas que armazenam diferentes tipos de informações (string, float, integer e etc)

Ele é uma classe de objeto da biblioteca Pandas.

dataframe



The diagram illustrates a DataFrame structure. It features a table with three columns and three rows. The columns are labeled 'VARIÁVEL 1', 'VARIÁVEL 2', and 'VARIÁVEL 3'. The rows are labeled '0', '1', and '2'. A purple arrow points to the 'COLUNA' label above the columns, and another purple arrow points to the 'INDEX' label below the rows.

	COLUNA			
		VARIÁVEL 1	VARIÁVEL 2	VARIÁVEL 3
0				
1				
2				
INDEX				

E o series ??



DataSerie é estrutura unidimensional - como uma coluna do excel

Series

INDEX

A	3
B	-5
C	7

Um array unidimensional e rotulado capaz de armazenar qualquer tipo de dado.



```
s = pd.Series([3,-5,7,4], index = ['a','b','c','d'])  
print(s)
```



```
a    3  
b   -5  
c    7  
d    4  
dtype: int64
```


As Estrutura dos Dados:

Linhas e Colunas

nome_dataframe.shape

```
dados_pyladies.shape
```

```
(15, 5)
```

Variáveis (colunas)

nome_dataframe.columns

```
dados_pyladies.columns
```

```
Index(['Estado Origem ', 'Idade', 'Escolaridade', 'Trabalha_como_Programadora',  
      'Renda_Mensal'],  
      dtype='object')
```

Conhecendo os Dados:

Informações Gerais

nome_dataframe.info()

```
dados_pyladies.info()  
  
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 15 entries, 0 to 14  
Data columns (total 5 columns):  
Estado Origem      15 non-null object  
Idade              15 non-null int64  
Escolaridade       15 non-null int64  
Trabalha_como_Programadora  15 non-null object  
Renda_Mensal       15 non-null float64  
dtypes: float64(1), int64(2), object(2)  
memory usage: 680.0+ bytes
```

Selecionando uma Variável (coluna):

`nome_dataframe['coluna']`

```
dados_pyladies['Estado Origem ']  
  
0      SP  
1      SP  
2      MG  
3      RJ  
4      SP  
5      SP  
6      SP  
7      PE  
8      RJ  
9      SP  
10     SP  
11     SP  
12     SP  
13     SP  
14     SP  
Name: Estado Origem , dtype: object
```



Lembrando que uma coluna de dataframe é uma series.

Selecionando duas ou mais variáveis (coluna):

`nome_dataframe[['coluna', 'coluna2', 'colunaX']]`

```
[19] dados_pyladies[['Estado Origem ', 'Escolaridade', 'Idade']]
```

	Estado Origem	Escolaridade	Idade
0	SP	4	36
1	SP	2	25
2	MG	3	34
3	RJ	3	23
4	SP	4	31
5	SP	5	34
6	SP	5	39
7	PE	3	24
8	RJ	3	29
9	SP	3	27
10	SP	3	30
11	SP	4	25
12	SP	2	23
13	SP	1	16
14	SP	4	36

Filtrando um dataframe:

`nome_dataframe[nome_dataframe['coluna'] == condição]`

```
dados_pyladies[dados_pyladies['Trabalha_como_Programadora'] == 'S']
```

	ID	Estado	Origem	Idade	Escolaridade	Trabalha_como_Programadora	Renda_Mensal
0	1		SP	36	4	S	3737,52
2	3		MG	34	3	S	2366,14
3	4		RJ	23	3	S	2841,29
5	6		SP	34	5	S	3433,02
6	7		SP	39	5	S	2752,74
7	8		PE	24	3	S	3682,33
8	9		RJ	29	3	S	2359,28
9	10		SP	27	3	S	2119,15
10	11		SP	30	3	S	3326,79
11	12		SP	25	4	S	2684,05
12	13		SP	23	2	S	3507,84

Aqui você insere a condição para o filtro que você quer. Se a condição for um texto, não se esqueça das aspas!

Aqui você coloca o operador lógico que atende o filtro que você precisa.

Estatística



População e amostra

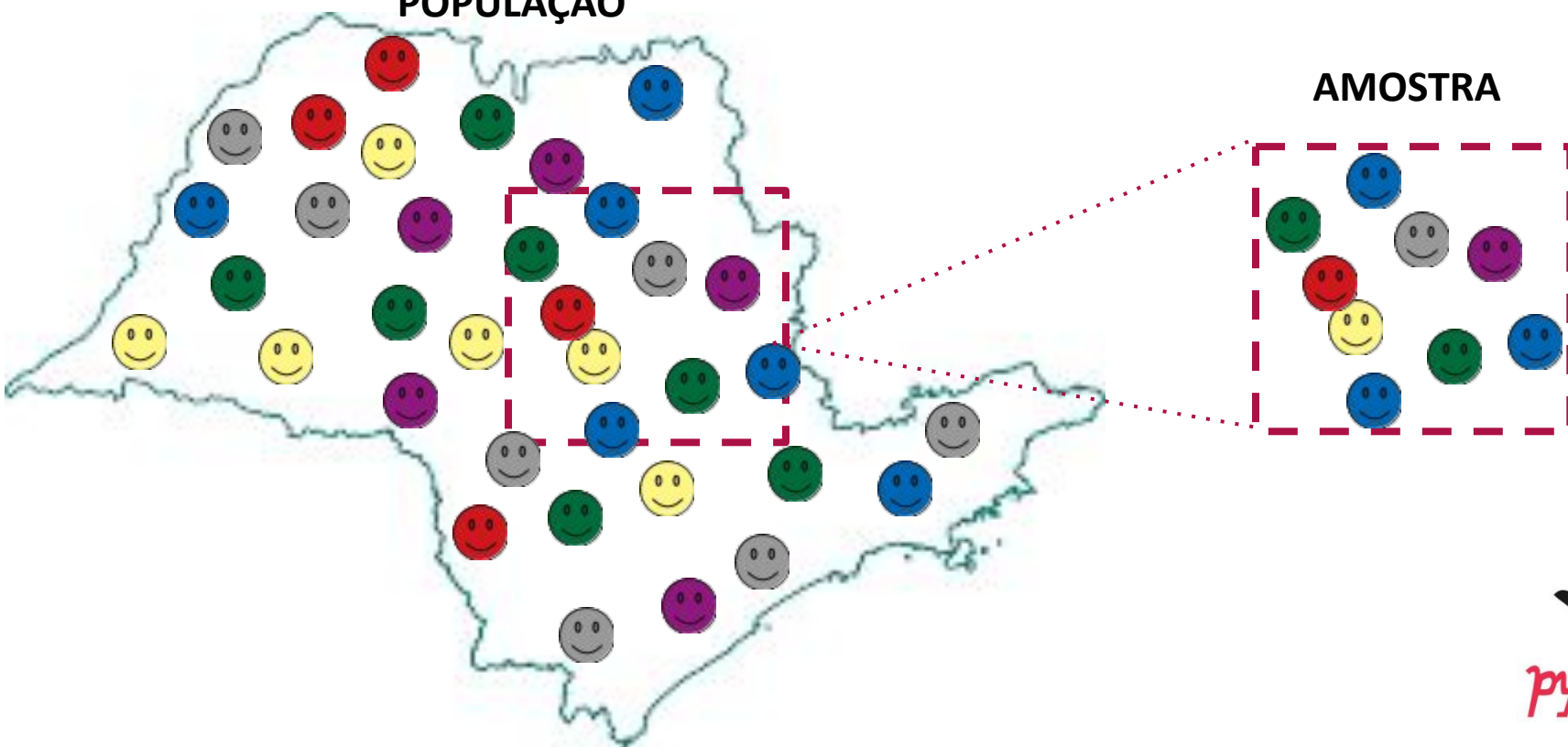
População é um conjunto de objetos, pessoas, itens sobre os quais deseja-se fazer inferências.

Amostra é um subconjunto de objetos, pessoas ou itens que representam a população.

Com uma amostra representativa, é possível inferir sobre essa população.

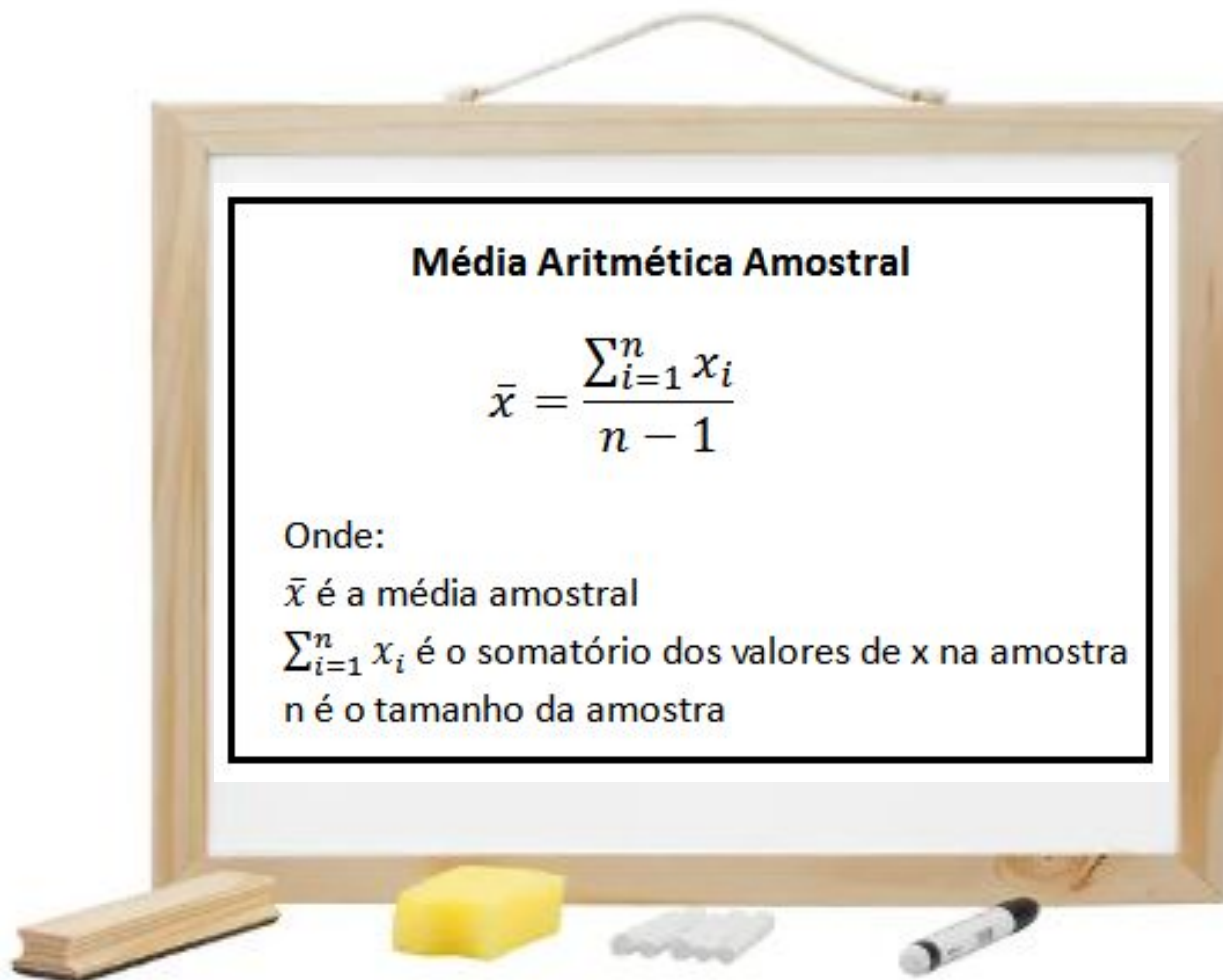
POPULAÇÃO

AMOSTRA



Média

A **média** é a soma de todos os elementos dividido pelo número de elementos.



Qual a renda média das meninas no dataframe dados_pyladies?

Para responder essa pergunta precisamos:

1º) Somar a renda mensal de todas as meninas;

```
valor = (3737.52 + 400.00 + 2366.14 + 2841.29 + 800.00 + 3433.02 + 2752.74 + 3682.33 + 2359.28 +  
        2119.15 + 3326.79 + 2684.05 + 3507.84 + 0.00 + 800.00)
```

2º) Dividir o valor obtido pelo total de meninas.

```
media = valor / 15  
print(media)
```

```
2320.6766666666667
```

Ou seja, em média a Renda Mensal é de R\$2.320,68.

Codando fica:

nome_dataframe['coluna'].mean()

Média Renda Mensal:



```
dados_pyladies['Renda_Mensal'].mean()
```



```
2320.6766666666667
```

Média Idade:



```
dados_pyladies['Idade'].mean()
```



```
28.8
```

A **Moda** é aquele elemento que mais se repete na distribuição dos dados.

Estado Origem	Frequência
SP	11

Qual será a UF que mais se repete?

nome_dataframe['coluna'].mode()

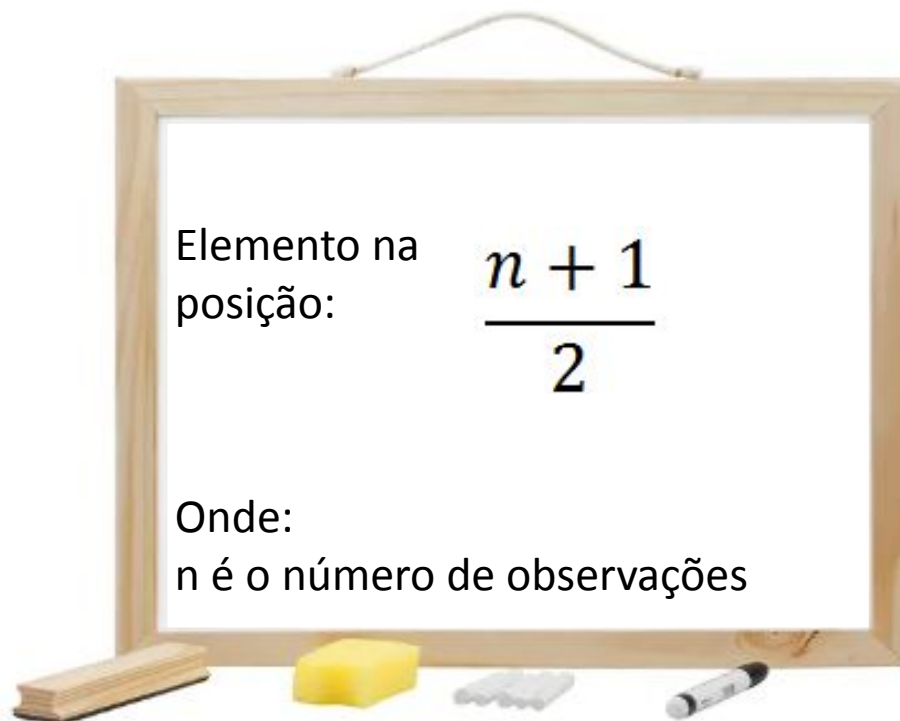
```
[48] dados_pyladies['Estado Origem'].mode()
```

```
0    SP  
dtype: object
```

Mediana

Mediana é o valor do meio de um conjunto de dados ordenados.

- Para um conjunto com número ímpar de observações: é o valor que divide exatamente na metade esse conjunto.
- Para um conjunto com número par de observações: é a média dos dois valores do meio.



Mediana

Qual a Mediana quando observamos a idade das meninas?

Para responder essa pergunta precisamos:

1º) Ordenar os dados do menor para o maior valor;

2º) Selecionar o valor mediano dos dados.

Idade														
16	23	23	24	25	25	27	29	30	31	34	34	36	36	39

Observamos que a Mediana não é influenciada pelo valor baixo de idade.

Mediana

Codando fica:

nome_dataframe['coluna'].median()

Idade Mediana:

```
[50] dados_pyladies['Idade'].median()
```

↳ 29.0

Salário Mediano:

```
dados_pyladies['Renda_Mensal'].median()
```

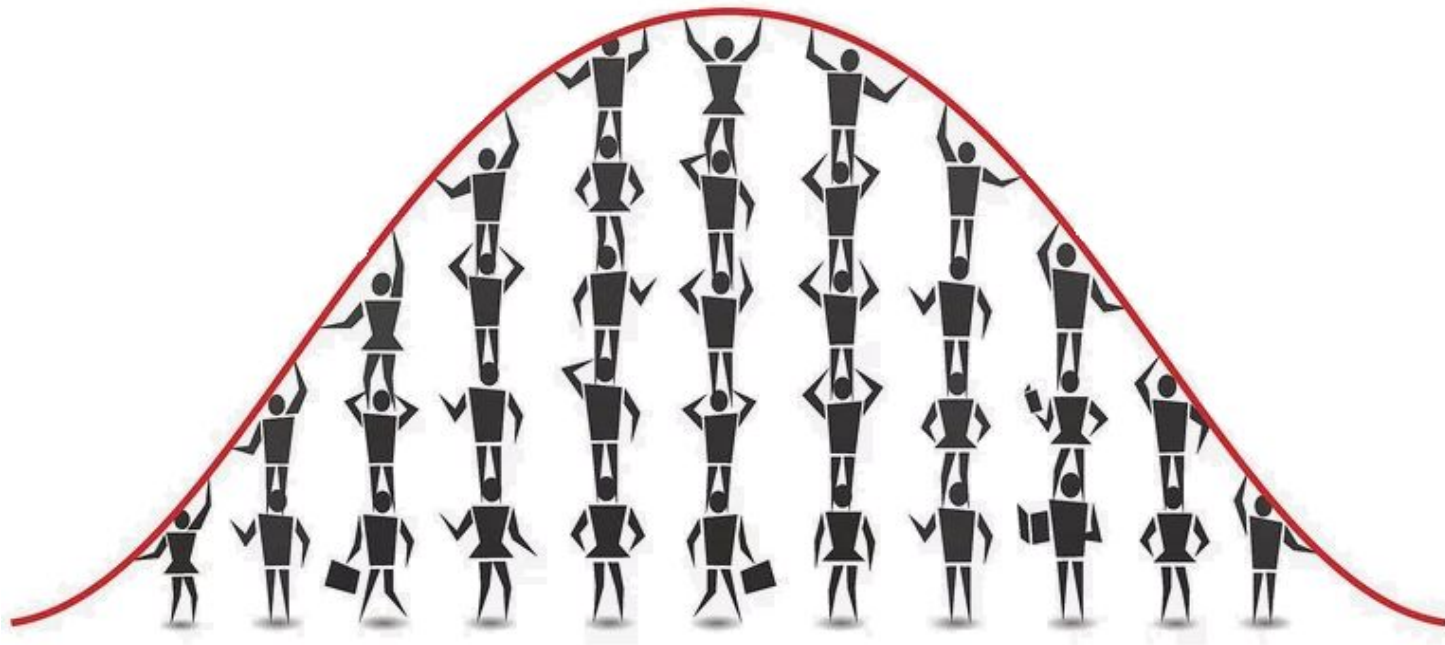
↳ 2684.05

**Mas média e mediana são as
mesmas coisas???**



Como Média, Moda e Mediana se relacionam?

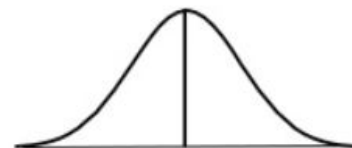
Falando brevemente sobre distribuições, há vários tipos de comportamento natural das medidas que observamos, um deles é a distribuição Normal.



Como Média, Moda e Mediana se relacionam?

Em amostras normalmente distribuídas a Média, a Mediana e a Moda possuem valores próximos!

Distribuição Simétrica
Média = Mediana = Moda



Se observarmos a renda mensal das meninas que trabalham com programação temos que a média e a mediana são muito próximas mesmo.

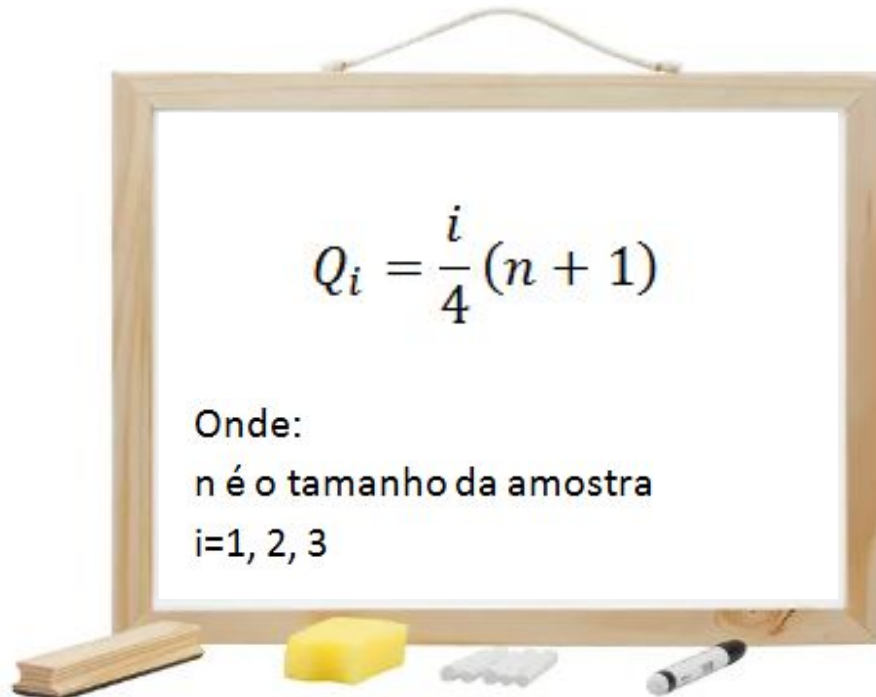
Média	R\$ 2982,74
Mediana	R\$ 2841,29

Se observarmos a idade das meninas, também obtemos valores de média e mediana próximos!

Média	Aprox. 29 anos
Mediana	29 anos

Quartis são valores que dividem uma amostra de dados ordenados em quatro partes iguais.

Com eles você pode rapidamente avaliar a dispersão e a tendência central de um conjunto de dados, que são etapas importantes na compreensão dos seus dados.



Quartis

Importante: para encontrar os quartis, os dados devem estar ordenados!

- 1º Quartil (Q1) - é onde estão 25% dos valores do conjunto de dados.
- 2º Quartil (Q2) - é onde estão até 50% dos valores, ou seja, a mediana!
- 3º Quartil (Q3) - é onde estão até 75% dos valores.

Isso quer dizer que:

- ✓ 25% dos valores do conjunto de dados são menores ou iguais ao Q1 e 75% dos valores são superiores ao Q1.
- ✓ 25% dos valores são superiores ou iguais ao Q3 e 75% dos valores são menores que Q3.
- ✓ 50% dos valores estão entre o 1º e o 3º Quartil.



Codando fica:

nome_dataframe['coluna'].quantile()

25%



```
dados_pyladies.quantile(.25)
```



```
Idade          24.500
Escolaridade    3.000
Renda_Mensal   1459.575
Name: 0.25, dtype: float64
```

25% do conjunto de dados tem:

- idade até 24,5 anos;
- até a escolaridade 3;
- a renda mensal até R\$ 1459, 58

75%

```
[21] dados_pyladies.quantile(.75)
```



```
Idade          34.000
Escolaridade    4.000
Renda_Mensal   3379.905
Name: 0.75, dtype: float64
```

75% do conjunto de dados tem:

- idade até 34 anos;
- até a escolaridade 4;
- a renda mensal até R\$ 3379,91

Tabela de Frequência

Mas será que a renda mensal média varia com relação às demais características?

Para respondermos isso podemos criar uma tabela de frequência que nos mostrará a variação dos dados um pelo outro

Trabalha como Programadora	Soma Renda Mensal	Quantidade Meninas	Renda Mensal Média
S	32810,15	11	2982,74
N	2000	4	500,00

Característica: Trabalhar ou não com programação!

Tabela de Frequência - Usando o Groupby

O Pandas possui a função groupby que nos permite agrupar dados, como o exemplo anterior.

Ele nos permite visualizar rapidamente uma tabela de frequência.

nome_dataframe.groupby('coluna').método()

```
dados_pyladies.groupby('Trabalha_como_Programadora').count()
```

	Estado	Origem	Idade	Escolaridade	Renda_Mensal
Trabalha_como_Programadora					
N	4	4	4	4	4
S	11	11	11	11	11

- alguns métodos não funcionam com o groupby, para saber mais consulte a documentação da [biblioteca](#) Pandas

Tabela de Frequência - Groupby

Há vários métodos que podem ser utilizados com o groupby.

Para contar os valores

```
nome_dataframe.groupby('coluna')['coluna'].value_counts()
```

Para somar valores

```
nome_dataframe.groupby('coluna')['coluna'].sum()
```

qual outro?

```
nome_dataframe.groupby('coluna')['coluna'].método()
```

Tabela de Frequência - Groupby + Agg

Podemos utilizar um Groupby com uma função Para isso utilizamos a função aggregation.

```
nome_dataframe.groupby('coluna')['coluna'].agg(['método', 'método'])
```

```
dados_pyladies.groupby('Trabalha_como_Programadora')['Renda_Mensal'].agg(['count', 'mean', 'median'])
```

	count	mean	median
Trabalha_como_Programadora			
N	4	500.000000	600.00
S	11	2982.740909	2841.29

Dispersão dos Dados

Dispersão dos Dados



Quando comparamos a média com o restante dos valores de uma variável, nós queremos entender o quanto aquele valor está distante da média.

ID	Estado Origem	Idade	Escolaridade	Trabalha como Programadora	Renda Mensal
1	SP	36	4	S	3737,52
2	SP	25	2	N	400,00

A média da Renda Mensal é de: **R\$ 2320,68**

Se compararmos os valores da tabela acima percebemos o quanto eles variam em relação a média

Variância

Uma medida muito interessante para avaliarmos a dispersão dos dados é a **variância**!

Vimos que a média nos informa sobre a tendência central, mas a variância que indica como esses dados variam.



Variância

Renda Mensal (x)
3737,52
400
2366,14
2841,29
800
3433,02
2752,74
3682,33
2359,28
2119,15
3326,79
2684,05
3507,84
0
800

2684,05

Será que a amostra tem renda parecida?

$$s = \frac{(x_1 - \text{média})^2 + (x_2 - \text{média})^2 + (x_3 - \text{média})^2 + \dots + (x_n - \text{média})^2}{n-1}$$

$$s = \frac{(3737,52 - 2684,05)^2 + (400 - 2684,05)^2 + \dots + (0 - 2684,05)^2}{15-1}$$

$$s = 1560989.622$$

Variância

Codando fica:

`nome_dataframe['coluna'].var()`

Variância Renda:

```
dados_pyladies['Renda_Mensal'].var()
```

```
1560989.6225666667
```

Variância Idade:

```
dados_pyladies['Idade'].var_()
```

```
39.600000000000001
```

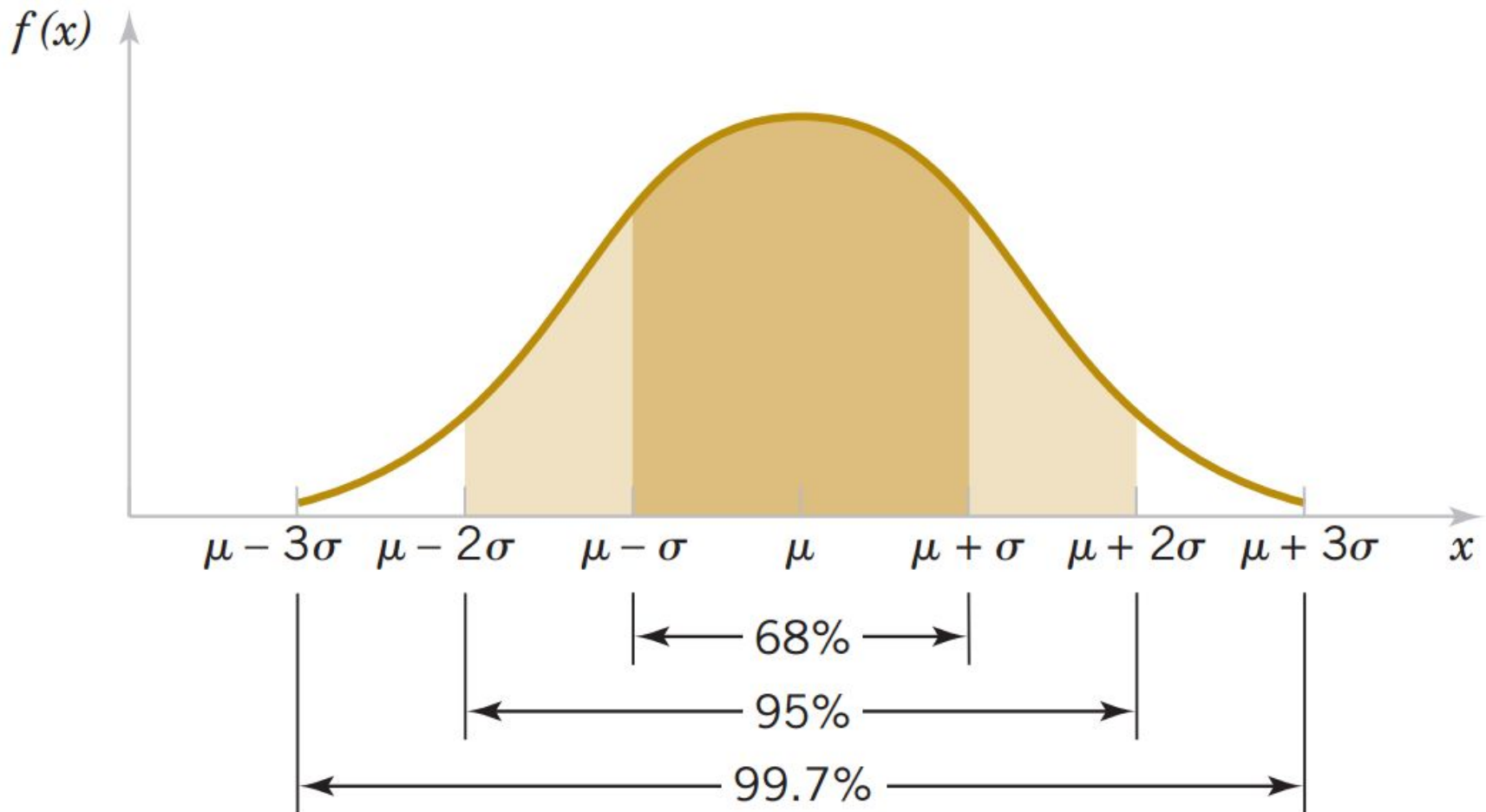
Desvio Padrão

O **desvio padrão** é uma medida que expressa o grau de dispersão de um conjunto de dados

Ele é a raiz quadrada da variância e a vantagem de utilizarmos esta medida é que o desvio padrão é expresso na mesma unidade dos dados, o que facilita a comparação.



Desvio Padrão



Fonte: <https://www.inf.ufsc.br/~andre.zibetti/probabilidade/normal.html>

Desvio Padrão

Codando fica:

nome_dataframe['coluna'].std()

Desvio Padrão Renda:

```
dados_pyladies['Renda_Mensal'].std()
```

```
1249.3957029567
```

Desvio Padrão Idade:

```
dados_pyladies['Idade'].std()
```

```
6.29285308902091
```


Por fim! Describe

Para conseguirmos visualizar as medidas centrais e de dispersão de um conjunto de dados, nós podemos utilizar o método describe.

nome_dataframe.describe()

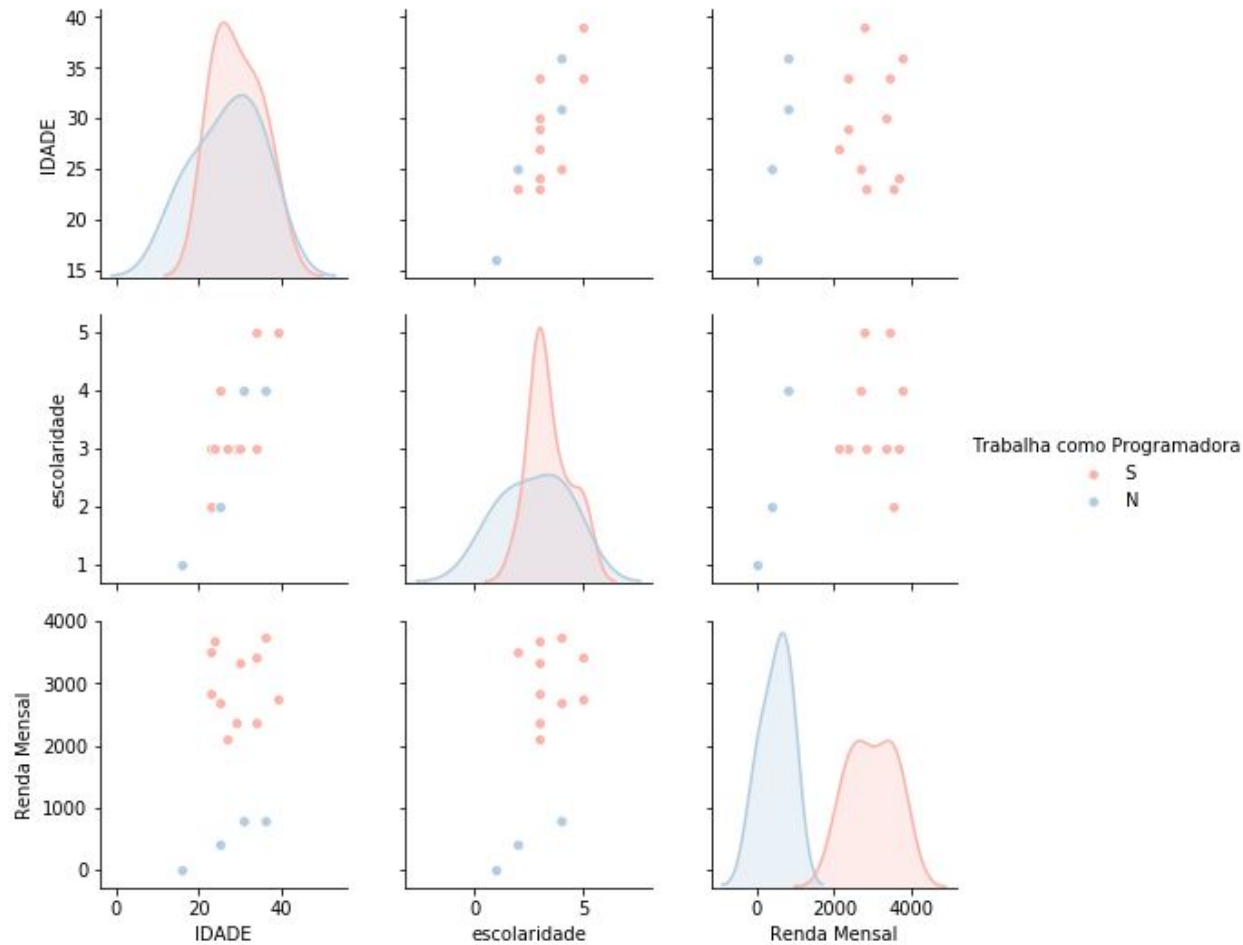
```
[14] dados_pyladies.describe()
```



	Idade	Escolaridade	Renda_Mensal
count	15.000000	15.000000	15.000000
mean	28.800000	3.266667	2320.676667
std	6.292853	1.099784	1249.395703
min	16.000000	1.000000	0.000000
25%	24.500000	3.000000	1459.575000
50%	29.000000	3.000000	2684.050000
75%	34.000000	4.000000	3379.905000
max	39.000000	5.000000	3737.520000

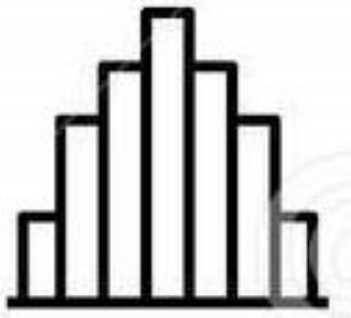
Introdução à Visualização dos Dados

Visualização de Dados - Gráficos



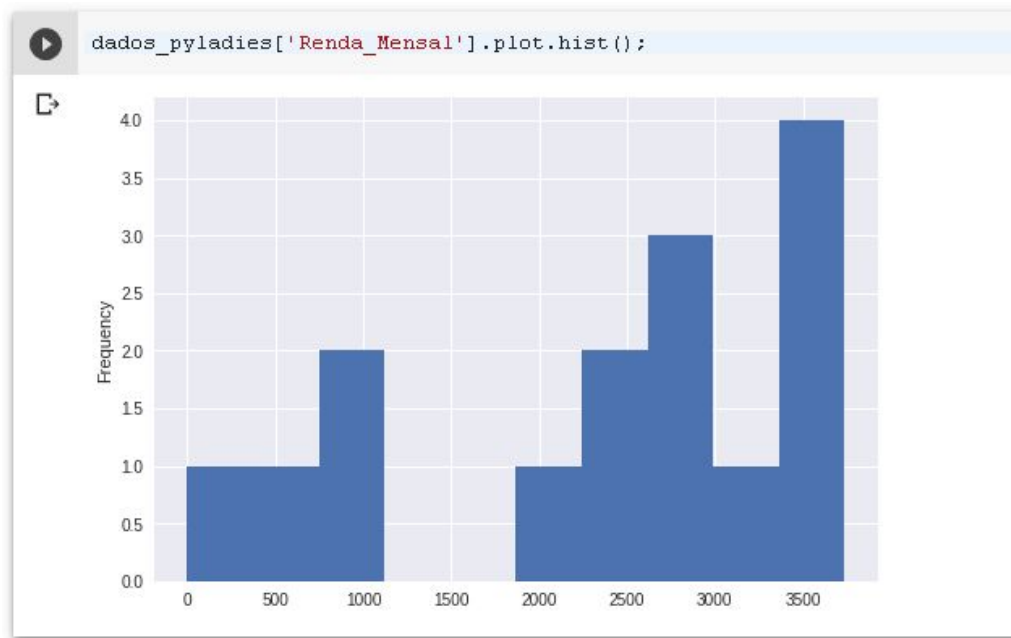
Visualização de Dados - Gráficos

Histograma



O Histograma é um gráfico que representa a distribuição de frequências de uma variável numérica contínua.

`nome_dataframe['coluna'].plot.hist()`



Histograma

pandas.DataFrame.hist

`DataFrame.hist(data, column=None, by=None, grid=True, xlabelsize=None, xrot=None, ylabelsize=None, yrot=None, ax=None, sharex=False, sharey=False, figsize=None, layout=None, bins=10, **kwargs)` [\[source\]](#)

Make a histogram of the DataFrame's.

A [histogram](#) is a representation of the distribution of data. This function calls `matplotlib.pyplot.hist()`, on each series in the DataFrame, resulting in one histogram per column.

data : DataFrame

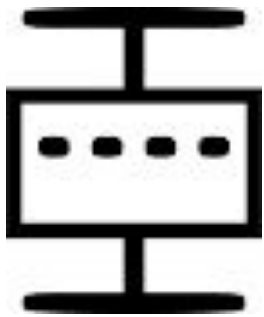
The pandas object holding the data.

column : string or sequence

If passed, will be used to limit data to a subset of columns.

A função hist possui vários argumentos possíveis. Sempre que houver dúvidas sobre quais são os argumentos possíveis, consulte a documentação da função.

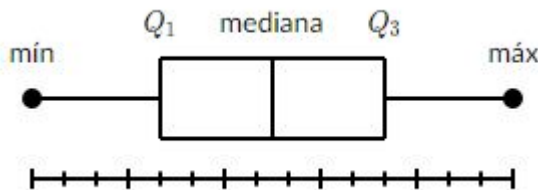
BoxPlot



O boxplot nos permite avaliar a distribuição do conjunto de dados, utilizando como referência os quartis.

A “caixa principal” é formada pelo primeiro quartil, a mediana e terceiro quartil.

As hastes inferior e superior são os limites e podem ser calculadas da seguinte forma:

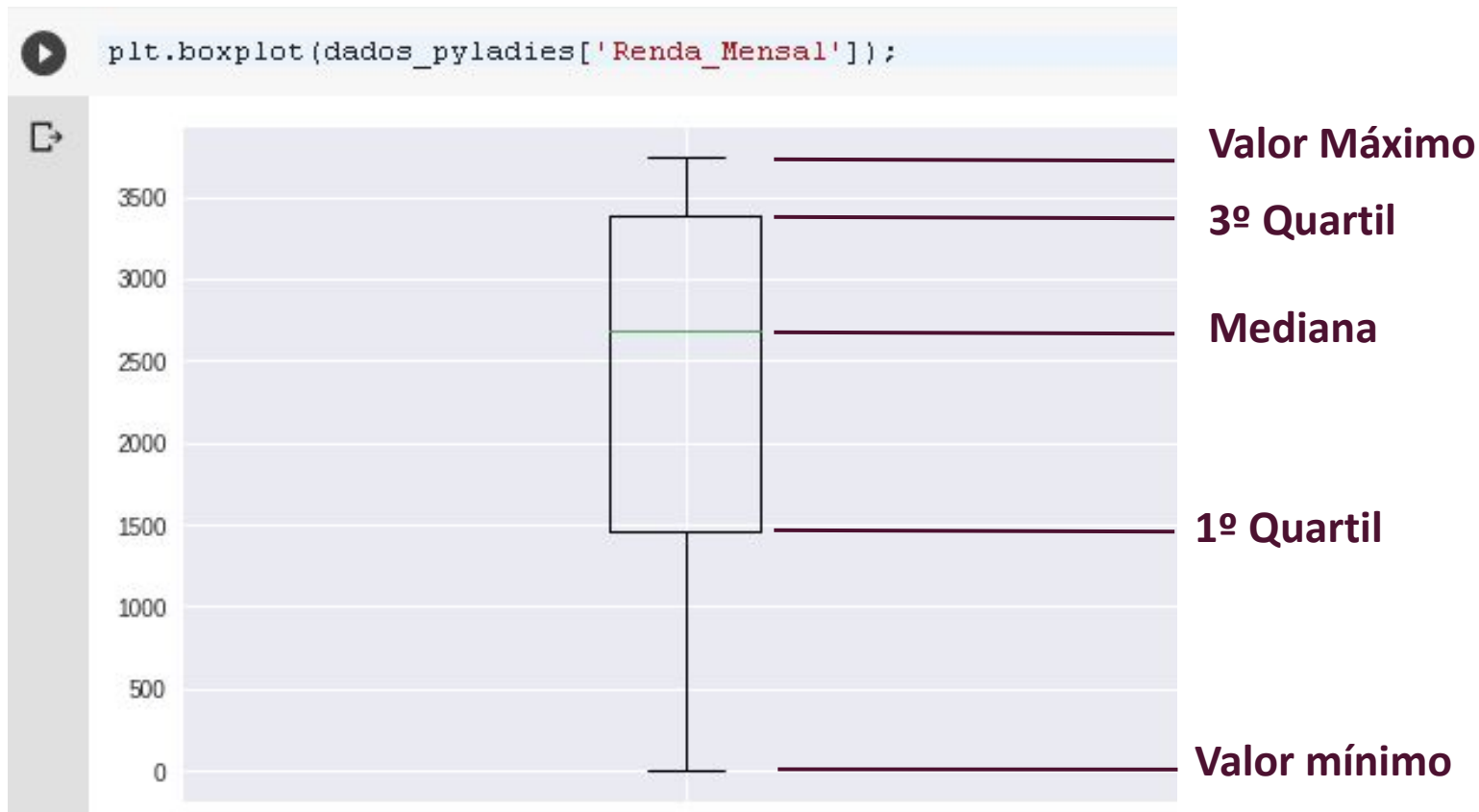


$$\text{Limite inferior: } Q_1 - 1,5(Q_3 - Q_1)$$

$$\text{Limite superior: } Q_3 + 1,5(Q_3 - Q_1)$$

Visualização de Dados - Gráficos

`plt.boxplot(nome_dataframe['coluna'])`



Tipos de gráficos - Data to Viz



<https://www.data-to-viz.com>

Para saber mais

- Livro Guia Mangá de Estatística - Shin Takahashi
- Plataforma Kaggle - <https://www.kaggle.com/>
- Podcast Pizza de Dados - <https://pizzadedados.com/>
- Documentação Pandas -
<https://pandas.pydata.org/pandas-docs/stable/index.html>
- Udacity - <https://www.udacity.com/>
- Canal EstaThiFisco
<https://www.youtube.com/channel/UC4jROkPjTvnXRkuo2GAwKXw>
- Minerando Dados - <http://minerandodados.com.br/>
- Cientista de Dados com GIFs - <https://paulovasconcellos.com.br/>
- Data Hackers <https://datahackers.com.br/>
- Estatística Básica - P. A. Bussab, W. de O. Moretin -
<https://edisciplinas.usp.br/mod/resource/view.php?id=2425203>

Obrigada!



Linkedin
[angelicacustodio](#)



[PyLadiesSP](#)



[PyLadies São Paulo](#)



Mulheres que
amam programar
e ensinar Python