# INFERÊNCIA ESTATÍSTICA E DATA MINING


## ANÁLISE DESCRITIVA

# 01

# TIPOS DE DADOS

QUALI-QUANTI

# ESTAMOS MANIPULANDO MUITOS DADOS

# MAS, QUAIS TIPOS DE DADOS PODEMOS ENCONTRAR?

# QUAIS TIPOS DE DADOS PODEMOS ENCONTRAR?

- Quantitativos
- Qualitativos

# QUAIS TIPOS DE DADOS PODEMOS ENCONTRAR?

- Quantitativos
- Qualitativos

Em geral são números:

Continuous

Discretos

# QUAIS TIPOS DE DADOS PODEMOS ENCONTRAR?

- Quantitativos
- Qualitativos

São os dados categóricos.

Texto, em geral.

# DADOS QUALITATIVOS PRECISAM SER TRATADOS?

# DADOS QUALITATIVOS PRECISAM SER TRATADOS?

Depende...

1. Para técnicas de mineração de texto, é importante trabalhar como eles são originalmente.

2. Para técnicas voltadas a dados quantitativos, é importante tratá-los.

# TRATAMENTO DE DADOS QUALITATIVOS

Em alguns contexto, as variáveis qualitativas podem ser transformadas em quantitativas.

# TRATAMENTO DE DADOS QUALITATIVOS

Em alguns contexto, as variáveis qualitativas podem ser transformadas em quantitativas.

## Variável dummy

| genero |
| --- |
| Fem |
| Masc |
| Fem |

# TRATAMENTO DE DADOS QUALITATIVOS

Em alguns contexto, as variáveis qualitativas podem ser transformadas em quantitativas.

## Variável dummy

| genero |
|---|
| Fem |
| Masc |
| Fem |

| genero_Fem | genero_Masc |
|---|---|
| 1 | 0 |
| 0 | 1 |
| 1 | 0 |

# TRATAMENTO DE DADOS QUALITATIVOS

Em alguns contexto, as variáveis qualitativas podem ser transformadas em quantitativas.

## Variável dummy

| genero |
|---|
| Fem |
| Masc |
| Fem |
| NA |

| genero_Fem | genero_Masc |
|---|---|
| 1 | 0 |
| 0 | 1 |
| 1 | 0 |
| 0 | 0 |

# TRATAMENTO DE DADOS QUALITATIVOS

## Variável dummy

```
import pandas as pd

# Tratamento de variável dummy
df = pd.get_dummies(df.genero)
```

# ATIVIDADE 08 - CLASSROOM - 30 MINUTOS

Avalie na base de dados disponibilizada pelo menos três colunas que precisam de tratamento.

Aplique a metodologia **one-hot-encoding** que acabamos de entender.

# VAMOS PARA OS QUANTITATIVOS...

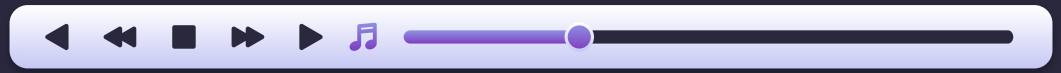
# MÉTODOS PARAMÉTRICOS E NÃO PARAMÉTRICOS

Os métodos paramétricos são uma categoria de técnicas estatísticas que fazem suposições específicas sobre a distribuição dos dados.

Os métodos não-paramétricos são técnicas estatísticas que não fazem suposições fortes sobre a forma da distribuição dos dados ou sobre a forma específica da relação entre variáveis

# MÉTODOS PARAMÉTRICOS PRECISAM DE DADOS NORMAIS

# O QUE SÃO DADOS NORMAIS?



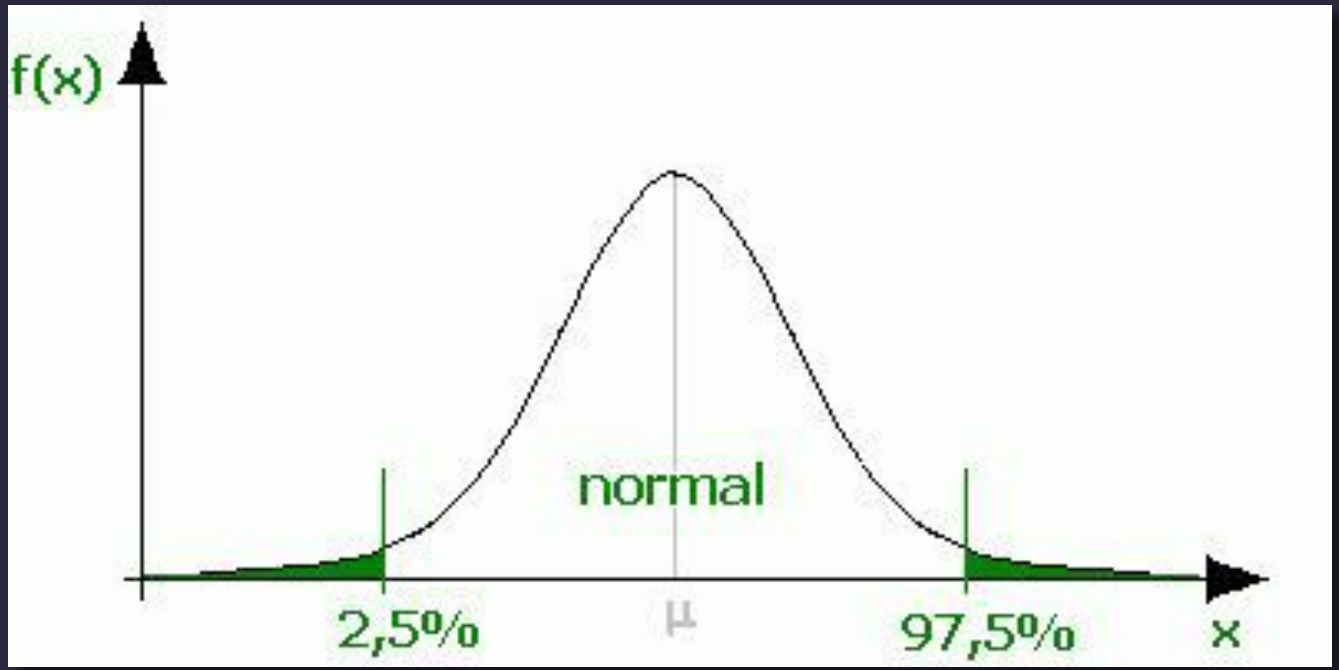
São dados provenientes de uma população que tem uma distribuição normal.

# DESCOMPLICANDO

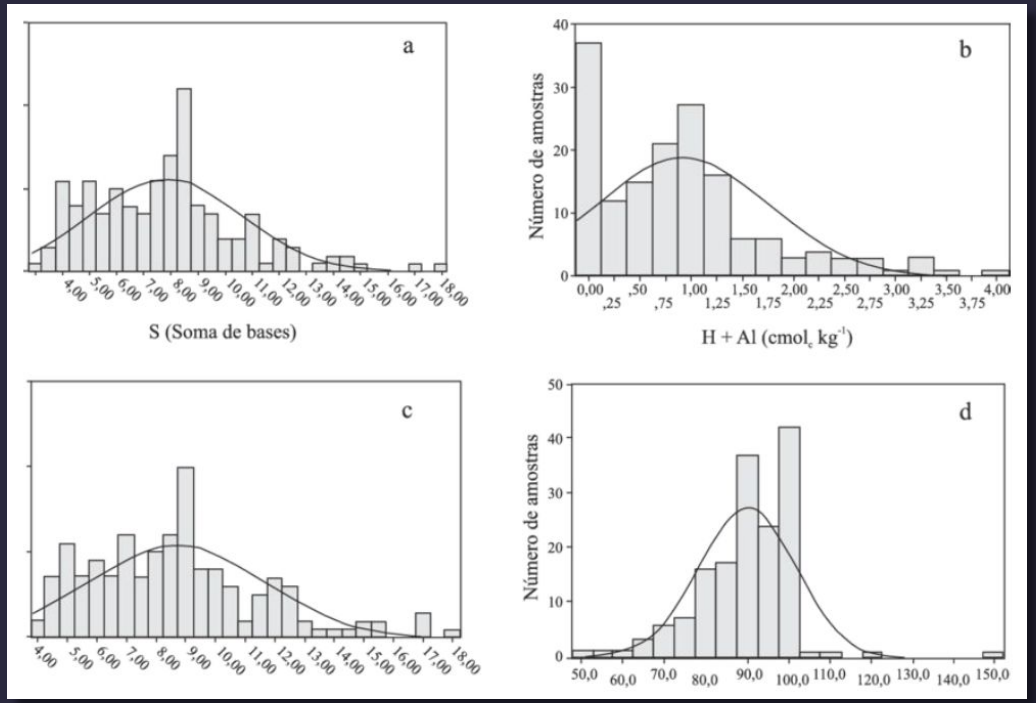
São dados provenientes de uma população que tem uma **distribuição** Normal

DISTRIBUIÇÃO É: Função que define uma curva (a área sob essa curva determina a probabilidade de ocorrer o evento por ela correlacionado).

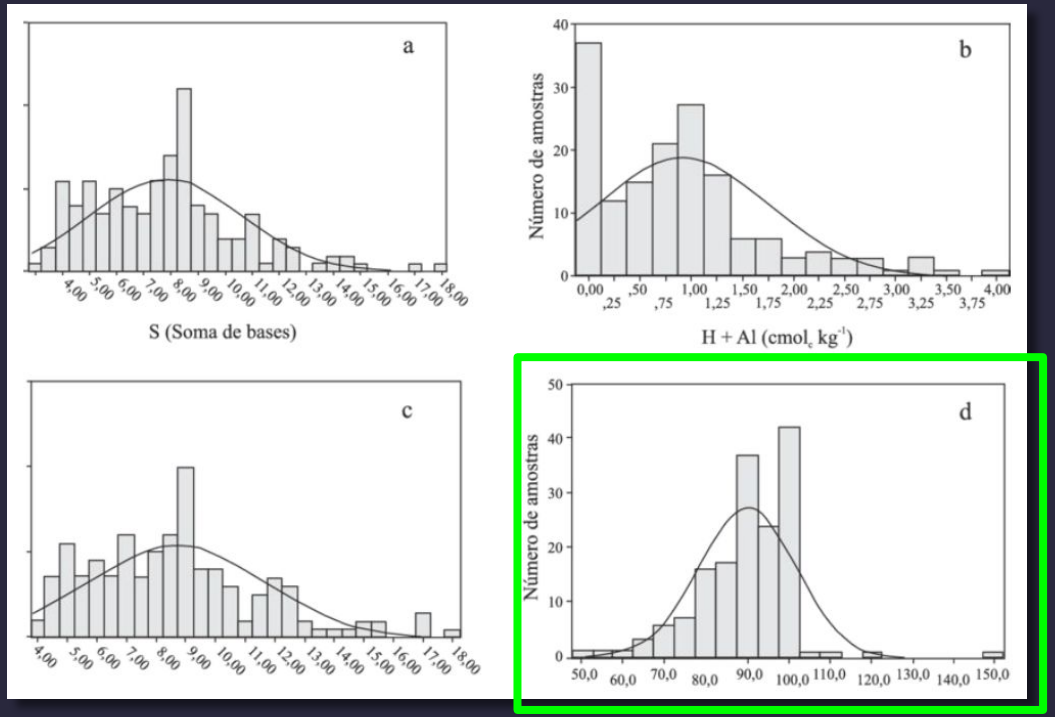
# DISTRIBUIÇÃO NORMAL OU GAUSSIANA



# QUAL DOS GRÁFICOS REPRESENTA UM DISTRIBUIÇÃO NORMAL?



# QUAL DOS GRÁFICOS REPRESENTA UM DISTRIBUIÇÃO NORMAL?



# EXISTEM OUTRAS DISTRIBUIÇÕES?

# SIM! MAS, POR ENQUANTO...

# COMO SABER SE O CONJUNTO DE DADOS SÃO NORMAIS?

# COMO SABER SE OS DADOS SÃO NORMAIS?

MÉTODO 1: observando visualmente o comportamento dos dados

# COMO SABER SE OS DADOS SÃO NORMAIS?

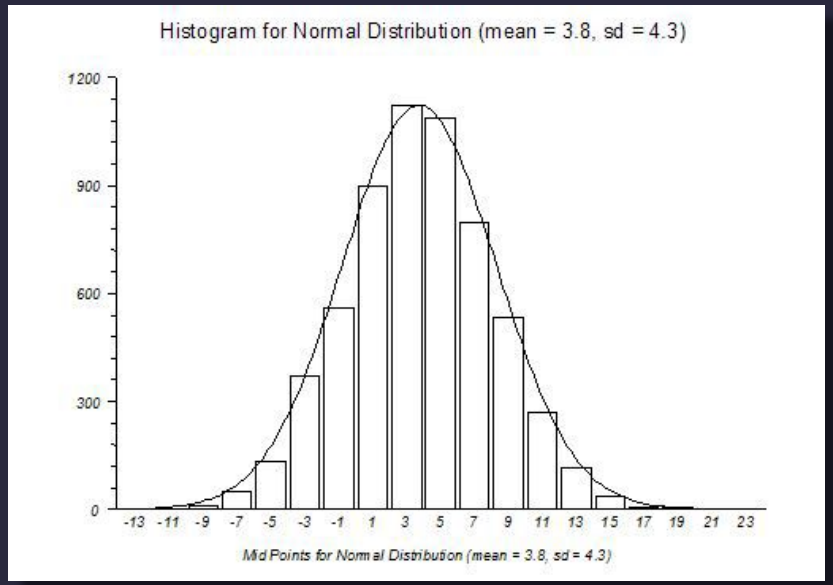
MÉTODO 1: observando visualmente o comportamento dos dados

**HISTOGRAMAS**

# COMO SABER SE OS DADOS SÃO NORMAIS?

MÉTODO 1: observando visualmente o comportamento dos dados

**HISTOGRAMAS**



# COMO SABER SE OS DADOS SÃO NORMAIS?

MÉTODO 1: observando visualmente o comportamento dos dados

```
import matplotlib.pyplot as plt

fig, ax = plt.subplots(figsize=(15, 20))
df['nome_da_coluna'].hist(bins=3, ax=ax)
plt.show()
```

# COMO SABER SE OS DADOS SÃO NORMAIS?

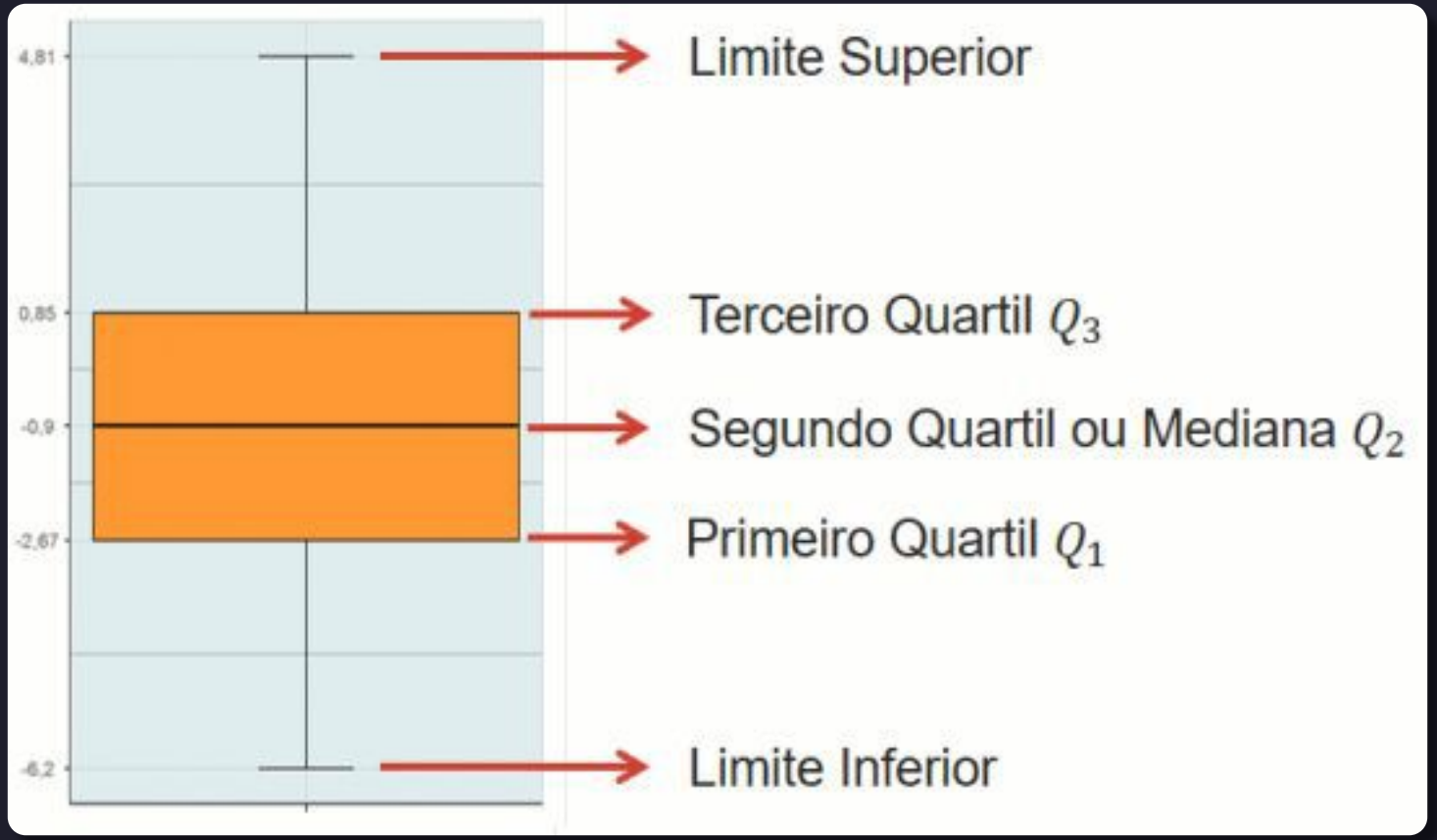
MÉTODO 1: observando visualmente o comportamento dos dados

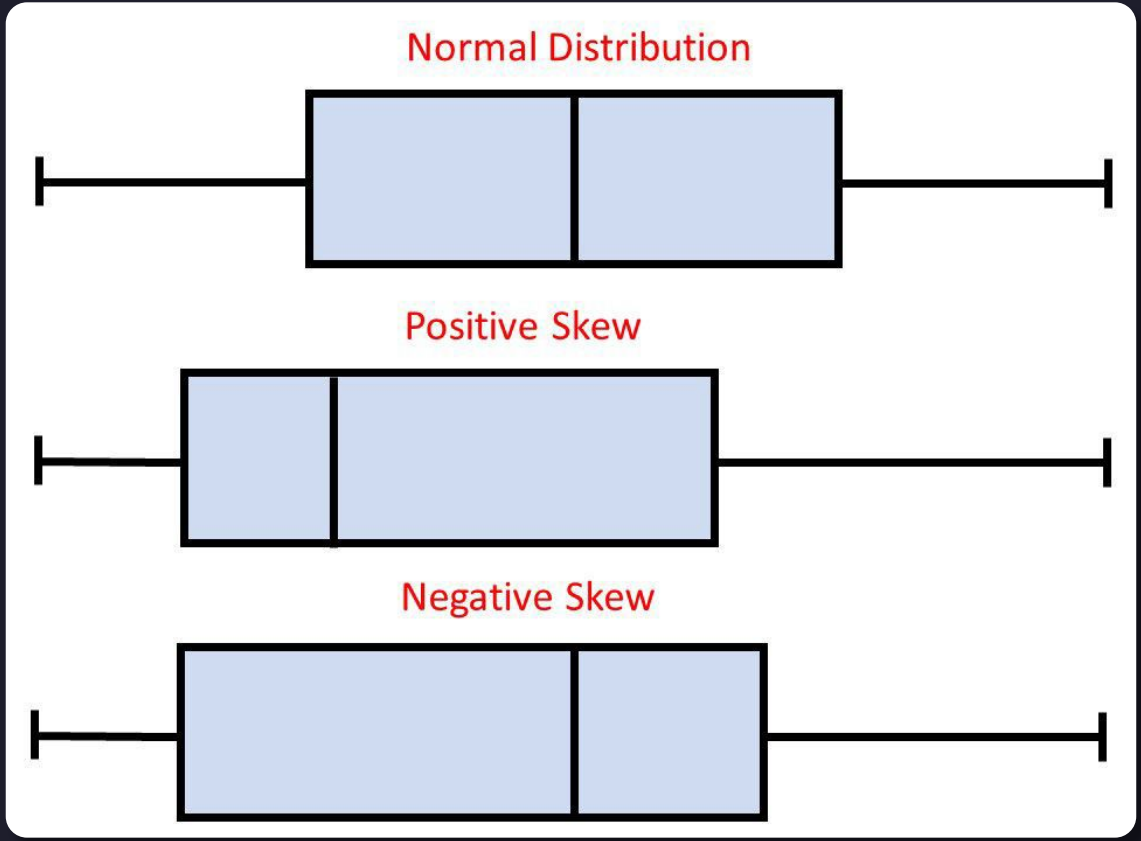
```
import matplotlib.pyplot as plt

fig = plt.figure(figsize = (15,20))
ax = fig.gca()
hist = df.hist(bins=3, ax=ax)
```

# COMO SABER SE OS DADOS SÃO NORMAIS?

MÉTODO 1: observando visualmente o comportamento dos dados

**BOXPLOT**


4,81
0,85
-0,9
-2,67
-6,2
Limite Superior
Terceiro Quartil $Q_3$
Segundo Quartil ou Mediana $Q_2$
Primeiro Quartil $Q_1$
Limite Inferior

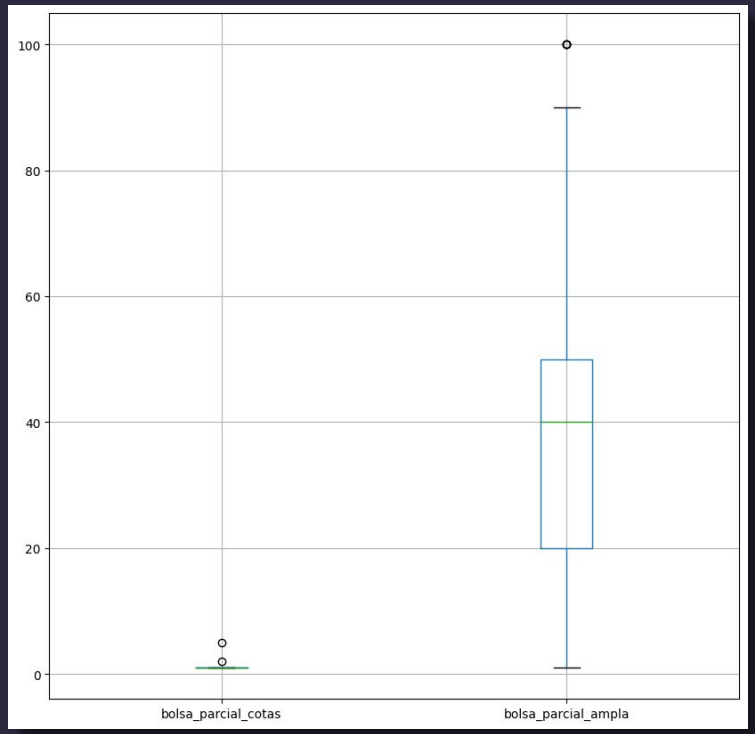

Normal Distribution
Positive Skew
Negative Skew

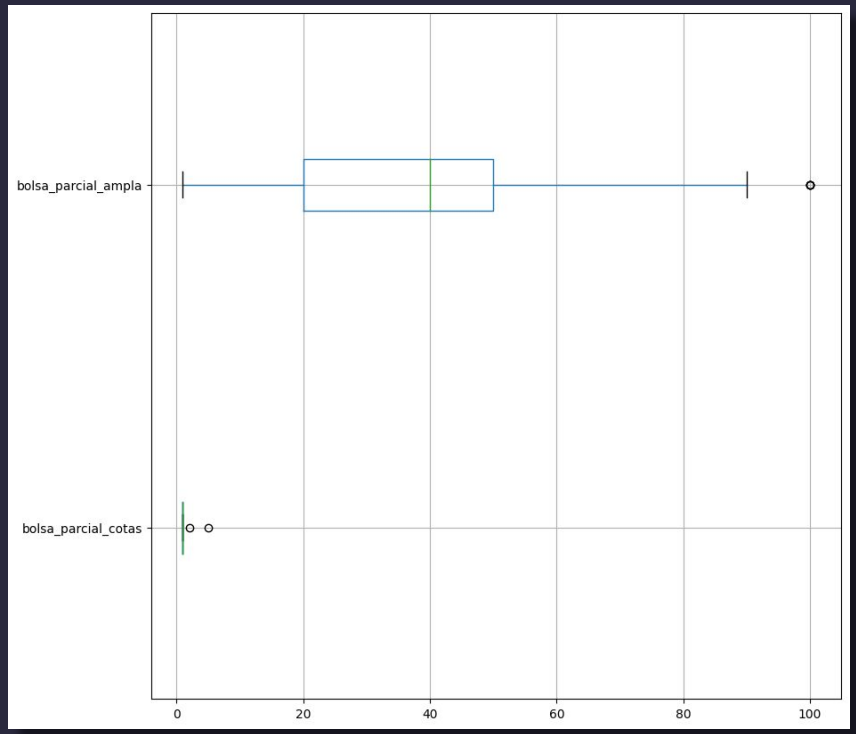
# COMO SABER SE OS DADOS SÃO NORMAIS?

## BOXPLOT

```
import matplotlib.pyplot as plt

fig = plt.figure(figsize = (10,10))

ax = fig.gca()

boxplot = df.boxplot(column=['bolsa_parcial_cotas',
'bolsa_parcial_ampla'],ax=ax, vert=False)
```

# COMO SABER SE OS DADOS SÃO NORMAIS?



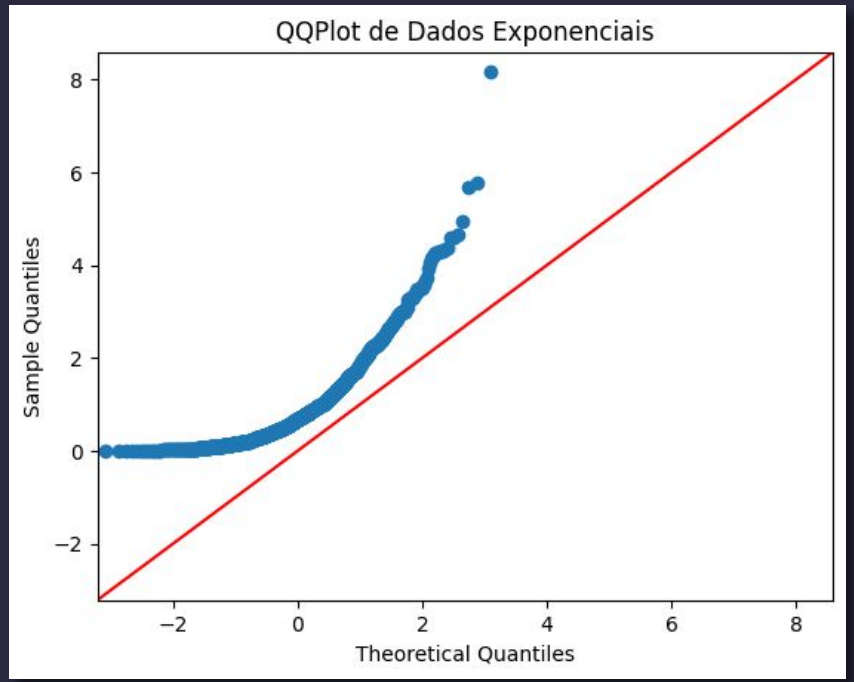
# COMO SABER SE OS DADOS SÃO NORMAIS?

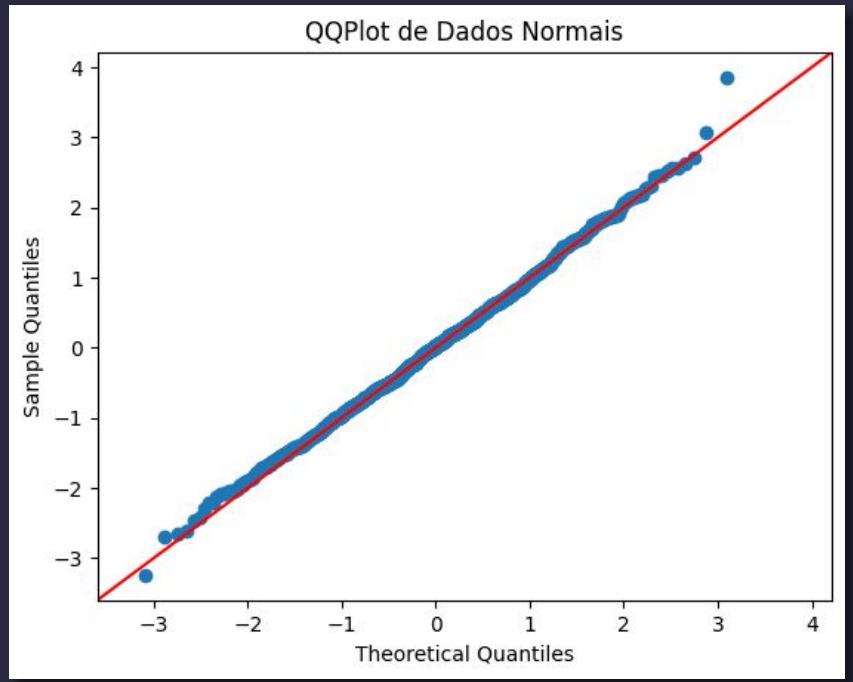
MÉTODO 1: observando visualmente o comportamento dos dados

**QQplot**

"**Quantile-Quantile Plot**", é uma ferramenta gráfica usada para ajudar a avaliar se um conjunto de dados segue uma certa distribuição.

# COMO SABER SE OS DADOS SÃO NORMAIS?



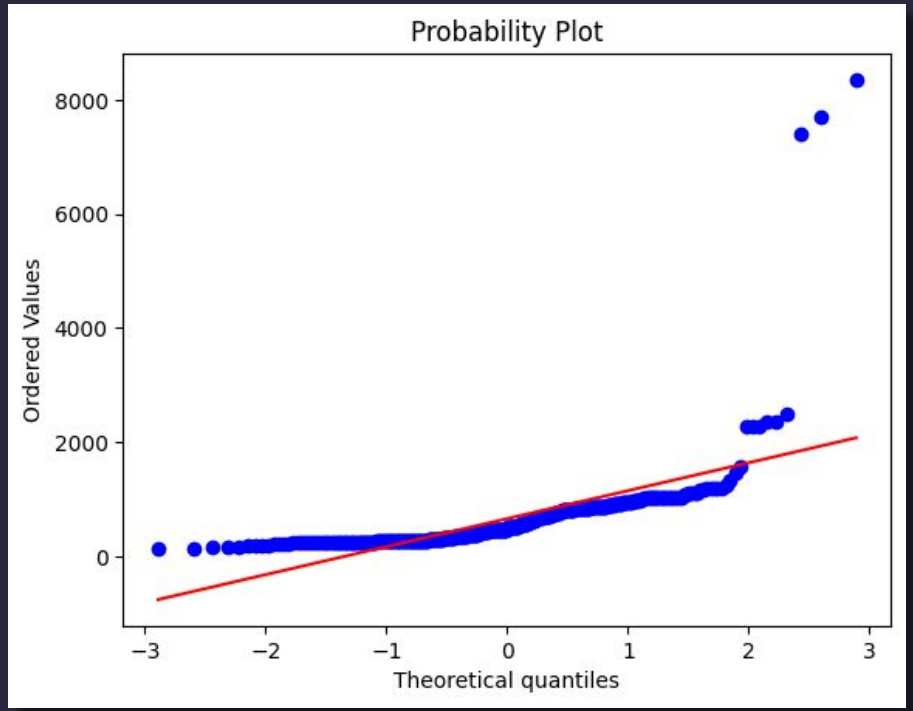
# COMO SABER SE OS DADOS SÃO NORMAIS?

## QQplot

```
import pylab
import scipy.stats as stats

stats.probplot(df.mensalidade, dist="norm",plot=pylab)
pylab.show()
```

# COMO SABER SE OS DADOS SÃO NORMAIS?

## QQplot



# COMO SABER SE OS DADOS SÃO NORMAIS?

MÉTODO 2: use o método do

**Kolmogorov Smirnov**

O teste de Kolmogorov-Smirnov (K-S) é um teste não-paramétrico usado para determinar se duas amostras são da mesma distribuição ou, em sua versão uniamostral, se uma amostra segue uma distribuição teórica específica.

# COMO SABER SE OS DADOS SÃO NORMAIS?

MÉTODO 2: use o método do

**Kolmogorov Smirnov**

```
from scipy.stats import kstest

valor_Ks, p_valor = kstest(df.mensalidade, 'norm')
print(valor_Ks, p_valor)
```

# COMO SABER SE OS DADOS SÃO NORMAIS?

MÉTODO 2: use o método do

**Kolmogorov Smirnov**

```
from scipy.stats import kstest

valor_Ks, p_valor = kstest(df.mensalidade, 'norm')
print(valor_Ks, p_valor)
```

```
1.0 0.0
```

# COMO SABER SE OS DADOS SÃO NORMAIS?

MÉTODO 2: use o método do

**Kolmogorov Smirnov**

```
from scipy.stats import kstest

valor_Ks, p_valor = kstest(df.mensalidade, 'norm')
print(valor_Ks, p_valor)
```

```
1.0 0.0
```

Se p_valor > 0.05: é normal

Se p_valor <= 0.05: não é normal

# COMO SABER SE OS DADOS SÃO NORMAIS?

MÉTODO 3: use o método do

**Lilliefors**

É uma extensão do teste de KS. O teste K-S, quando usado para testar a normalidade, requer que a média e a variância da população sejam conhecidas. No entanto, na prática, geralmente estimamos a média e a variância a partir dos dados. O teste de Lilliefors é uma modificação do K-S que leva em conta essa estimação.

# COMO SABER SE OS DADOS SÃO NORMAIS?

MÉTODO 3: use o método do

**Lilliefors**

```
from statsmodels.stats.diagnostic import lilliefors

lilliefors(df.mensalidade)
```

```
(0.2516123813493612, 0.0)
```

Índice de teste de Liliefors — p-valor

# COMO SABER SE OS DADOS SÃO NORMAIS?

MÉTODO 3: use o método do

**Lilliefors**

```
from statsmodels.stats.diagnostic import lilliefors

lilliefors(df.mensalidade)
```

```
(0.2516123813493612, 0.0)
```

Índice de teste de Liliefors — p-valor

Se p_valor > 0.05: é normal

Se p_valor <= 0.05: não é normal

# COMO SABER SE OS DADOS SÃO NORMAIS?

MÉTODO 4: use o método do

**Shapiro Wilk**

O teste de Shapiro-Wilk é um método estatístico utilizado para testar a hipótese nula de que um conjunto de dados foi extraído de uma população normalmente distribuída. Este teste é especialmente adequado para conjuntos de dados de pequeno a médio porte.

# COMO SABER SE OS DADOS SÃO NORMAIS?

MÉTODO 4: use o método do

**Shapiro Wilk**

```
from scipy.stats import shapiro

W, pvalue = shapiro(df.mensalidade)

print(f"Statistic (W): {W:.2f}")
print(f"P-value (notação): {pvalue:.2e}")
print(f"P-value (float): {pvalue:.20f}")
```

# COMO SABER SE OS DADOS SÃO NORMAIS?

## MÉTODO 4: use o método do

### Shapiro Wilk

```
from scipy.stats import shapiro

W, pvalue = shapiro(df.mensalidade)

print(f"Statistic (W): {W:.2f}")
print(f"P-value (notação): {pvalue:.2e}")
print(f"P-value (float): {pvalue:.20f}")
```

```
(0.42162, 0.0)
```

Índice de teste de Shapiro Wilk

p-valor

Se p_valor > 0.05: é normal

Se p_valor <= 0.05: não é normal

# COMO SABER SE OS DADOS SÃO NORMAIS?

MÉTODO 5: Avaliação de

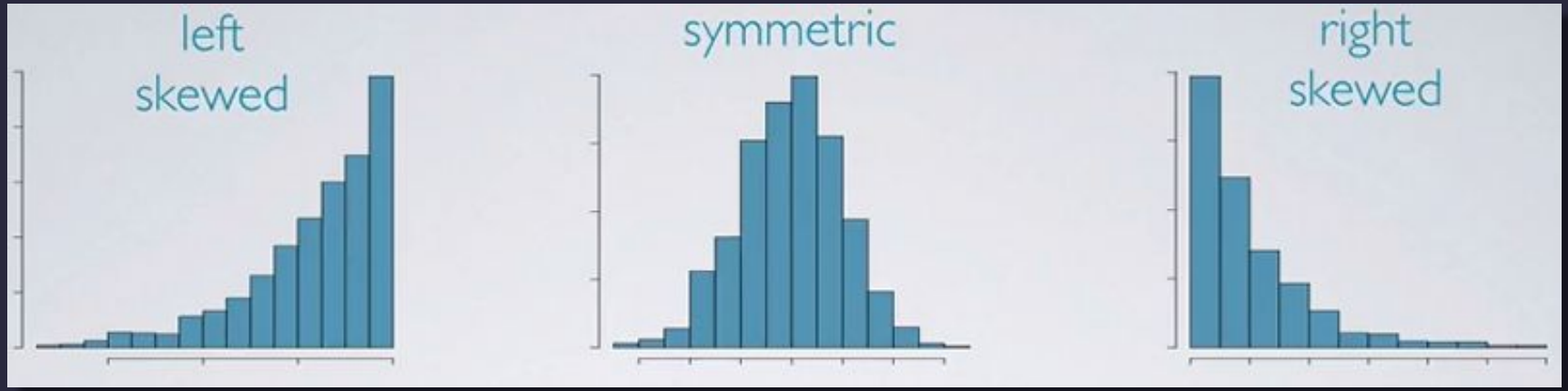
## Skewness e Kurtosis

Skewness (Assimetria) e Kurtosis (Curtose) são duas medidas estatísticas que fornecem informações sobre a forma e as características de uma distribuição.

# /COMO SABER SE OS DADOS SÃO NORMAIS?

MÉTODO 5: Avaliação de

## Skewness



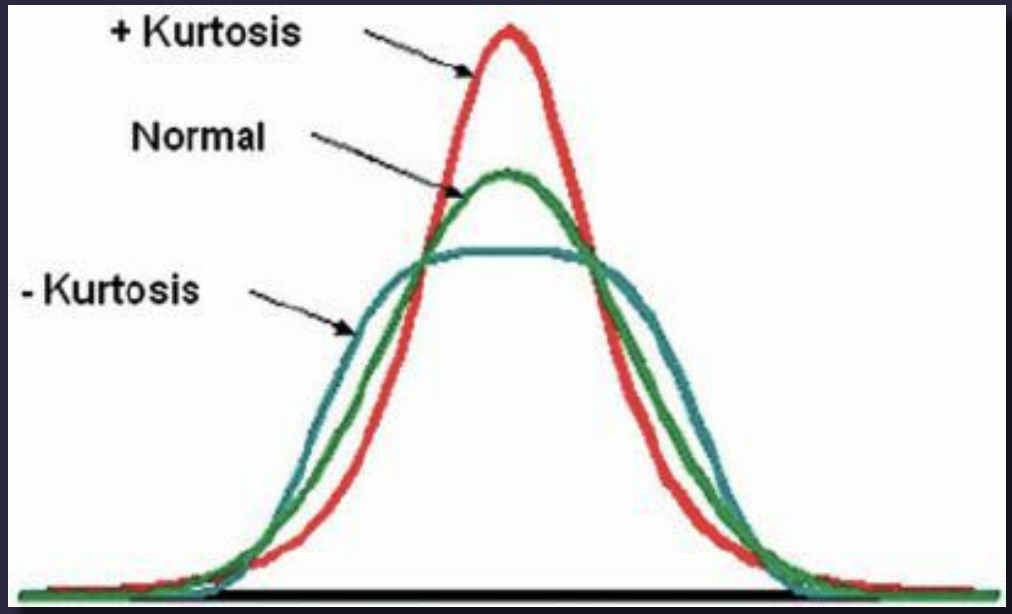
# COMO SABER SE OS DADOS SÃO NORMAIS?

## MÉTODO 5: Avaliação de **Skewness**

- **Assimetria negativa (ou à esquerda)**: A média é menor que a mediana.
- **Assimetria positiva (ou à direita)**: A média é maior que a mediana.
- **Simetria:** A distribuição é equilibrada em ambos os lados. A média é aproximadamente igual à mediana.

# /COMO SABER SE OS DADOS SÃO NORMAIS?

## MÉTODO 5: Avaliação de **Kurtosis**



# COMO SABER SE OS DADOS SÃO NORMAIS?

MÉTODO 5: Avaliação de

**Skewness e Kurtosis**

```
import scipy.stats as stats

S, pvalor = stats.normaltest(df.mensalidade)

print(f"Estatística de qui-quadrado: {S:.5f}")
print(f"P-value (notação): {pvalor:.2e}")
print(f"P-value (float): {pvalor:.20f}")
```

# COMO SABER SE OS DADOS SÃO NORMAIS?

## MÉTODO 5: Avaliação de

## Skewness e Kurtosis

```
import scipy.stats as stats

S, pvalor = stats.normaltest(df.mensalidade)

print(f"Estatística de qui-quadrado: {S:.5f}")
print(f"P-value (notação): {pvalor:.2e}")
print(f"P-value (float): {pvalor:.20f}")
```

```
(0.42162, 0.0)
```

Estatística de qui-quadrado

p-valor

Se p_valor > 0.05: é normal

Se p_valor <= 0.05: não é normal

# ATIVIDADE 09 - DESAFIO

Descubra se a base de dados do PROUNI tem comportamento normal ou não para cada uma das suas colunas.

Utilize pelo menos duas formas diferentes para cada coluna, uma visual e outra por estatística.

30 minutos

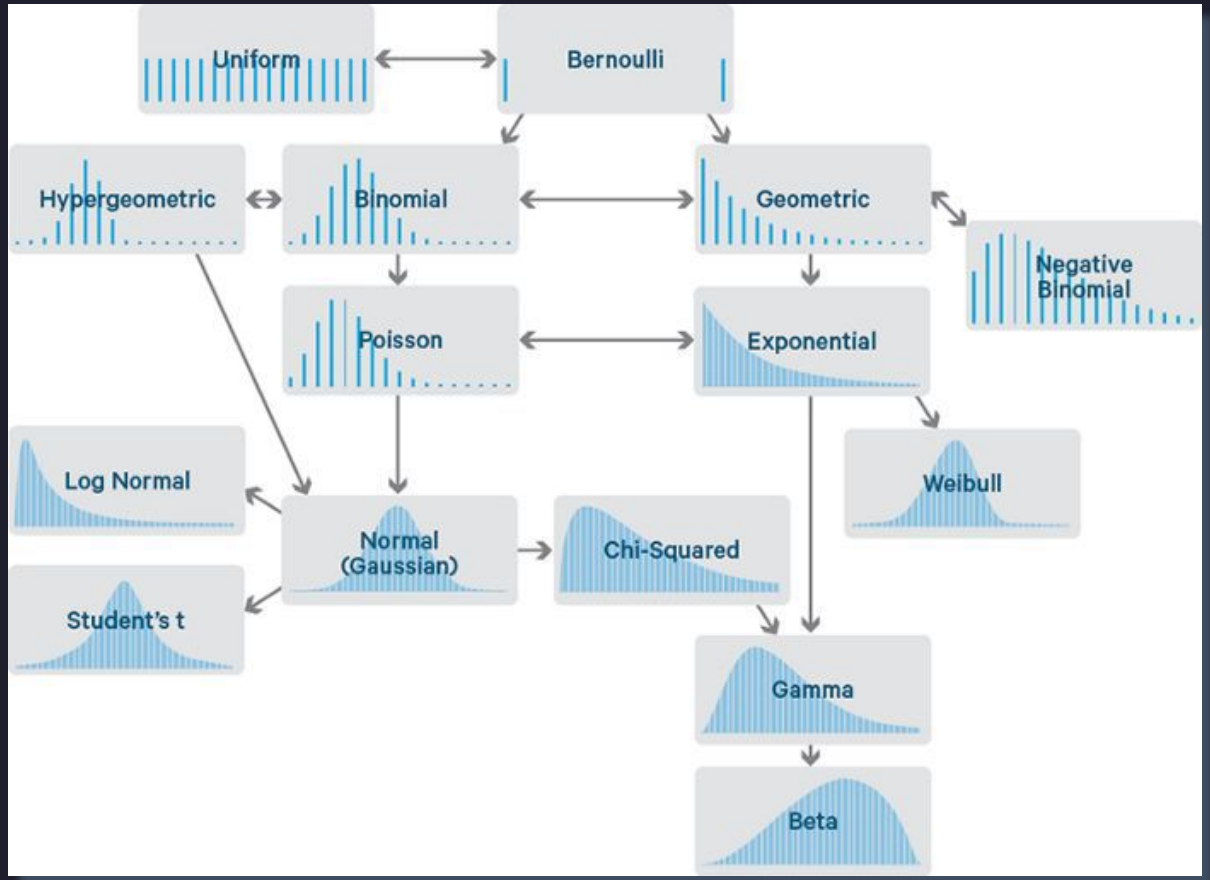
# COMO NORMALIZAR OS DADOS?

# COMO NORMALIZAR OS DADOS?

## Transformações

# COMO NORMALIZAR OS DADOS?

**Para realizarmos as transformações da forma correta, é IMPORTANTE saber tratar os diferentes tipos de distribuições.**


Uniform
Bernoulli
Hypergeometric
Binomial
Geometric
Negative Binomial
Poisson
Exponential
Log Normal
Weibull
Normal (Gaussian)
Chi-Squared
Student's t
Gamma
Beta

# TRANSFORMAÇÃO DE BOX COX

A transformação de Box-Cox é uma família de transformações potenciais que são usadas para estabilizar a variância e tornar os dados mais próximos a uma distribuição normal.

$$Y_i(\lambda) = \begin{cases} \ln(X_i), & \text{se } \lambda = 0, \\ \dfrac{X_i^{\lambda} - 1}{\lambda}, & \text{se } \lambda \neq 0, \end{cases}$$

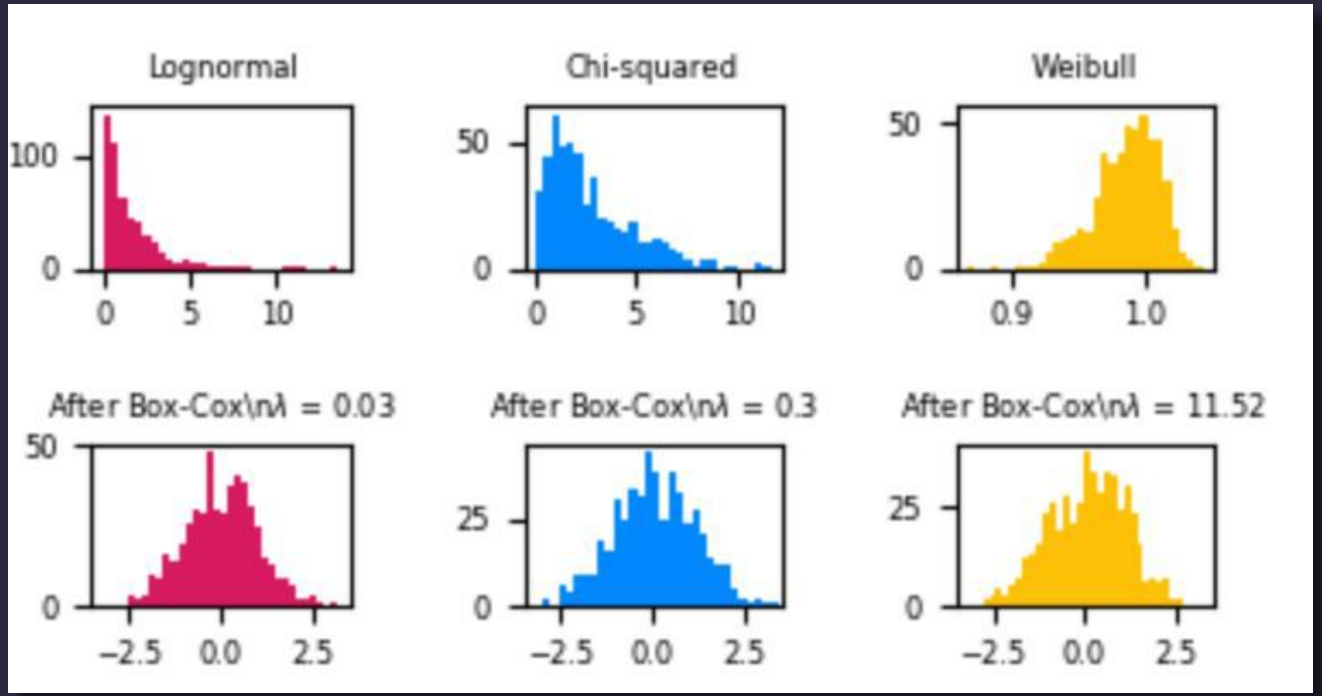
# TRANSFORMAÇÃO DE BOX COX

```
from sklearn import preprocessing as preprocessing

bc = preprocessing.PowerTransformer(method='box-cox')
bc = bc.fit(df.loc[:, ['mensalidade']])
df['mensalidade_norm'] = xt = bc.transform(df.loc[:, ['mensalidade']])
```

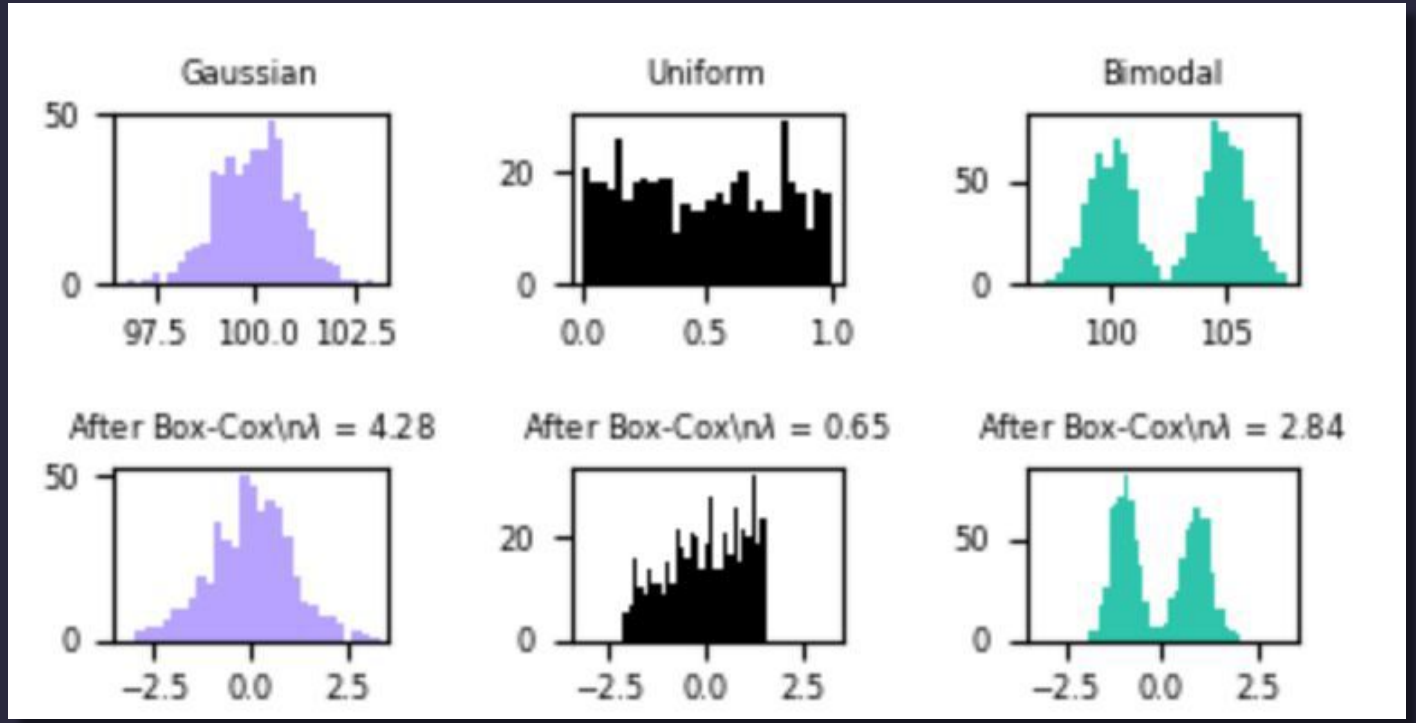
# TRANSFORMAÇÃO DE BOX COX

```
import matplotlib.pyplot as plt
import scipy.stats as stats


fig = plt.figure()
ax2 = fig.add_subplot(212)
xt, _ = stats.boxcox(df['mensalidade'])
prob = stats.probplot(xt, dist=stats.norm, plot=ax2)
```

# TRANSFORMAÇÃO DE BOX COX



# TRANSFORMAÇÃO DE BOX COX



# TRANSFORMAÇÃO DE YEO-JONHSON

A transformação de Yeo-Johnson é uma extensão da transformação de Box-Cox que pode ser aplicada a dados que incluem valores não positivos (ou seja, zero e negativos).

$$\psi(\lambda, y) = \begin{cases} ((y+1)^{\lambda} - 1)/\lambda & \text{if } \lambda \neq 0, y \geq 0 \\ \log(y+1) & \text{if } \lambda = 0, y \geq 0 \\ -[(-y+1)^{2-\lambda} - 1)]/(2-\lambda) & \text{if } \lambda \neq 2, y < 0 \\ -\log(-y+1) & \text{if } \lambda = 2, y < 0 \end{cases}$$

# TRANSFORMAÇÃO DE YEO-JONHSON

```
from sklearn import preprocessing as preprocessing

yt = preprocessing.PowerTransformer(method='yeo-johnson')
yt = yt.fit(df.loc[:, ['mensalidade']])
xt = yt.transform(df.loc[:, ['mensalidade']])
```

# TRANSFORMAÇÃO DE YEO-JONHSON

```
import matplotlib.pyplot as plt
import scipy.stats as stats


fig = plt.figure()
ax2 = fig.add_subplot(212)
xt, _ = stats.yeojohnson(df['mensalidade'])
prob = stats.probplot(xt, dist=stats.norm, plot=ax2)
```

# TRANSFORMAÇÃO POR QUANTILE

```
from sklearn import preprocessing as preprocessing

qt = preprocessing.QuantileTransformer(
    output_distribution='normal',
    random_state=0)
qt = qt.fit(df.loc[:, ['mensalidade']])
xt = qt.transform(df.loc[:, ['mensalidade']])
```

# OUTROS MÉTODOS DE TRANSFORMAÇÃO

- Standardize (sklearn)
- Normalize (sklearn)
- Análise de Componentes Principais (PCA)
- Análise dos Componentes Independentes (ICA).

# ATIVIDADE 10 << DESAFIO >>

Tente definir mecanismos de normalização sobre os dados do PROUNI.

Aplique em pelo menos 3 colunas que você identificou serem necessárias as normalizações.

30 minutos

# AGORA SIM! OS DADOS ESTÃO PRONTOS PARA SEREM UTILIZADOS!

# INFERÊNCIA ESTATÍSTICA E DATA MINING


## ANÁLISE DESCRITIVA