# Estatistica_básica

April 2, 2025

## 1 Tipos de Variáveis

Uma variável é uma característica de interesse medida em cada elemento da amostra ou da população, cujos valores variam de elemento para elemento.

Na prática, cada variável será provavelmente representada por uma coluna da sua base de dados. As variáveis podem ter valores numéricos ou não numéricos e podem ser classificadas da seguinte forma:

### 1.1 Variáveis quantitativas (numéricas):

São as características que podem ser medidas em uma escala quantitativa, ou seja, apresentam valores numéricos que fazem sentido.

Podem ser:

**Discretas**: são mensuráveis e podem assumir apenas um número finito ou infinito contável de valores e, assim, somente fazem sentido valores inteiros. Geralmente são o resultado de contagens.

**Exemplos**: número de filhos, número de bactérias por litro de leite, número de cigarros fumados por dia.

**Contínuas**: características mensuráveis que assumem valores em uma escala contínua (na reta real), para as quais valores fracionais fazem sentido. Usualmente devem ser medidas através de algum instrumento.

**Exemplos**: peso (balança), altura (régua), tempo (relógio), pressão arterial, idade.

### 1.2 Variáveis qualitativas (ou categóricas):

Não possuem valores quantitativos mas são definidas por várias categorias, ou seja, representam uma classificação dos indivíduos.

Podem ser:

**Nominais**: não existe ordenação dentre as categorias.

**Exemplos**: sexo, cor dos olhos, fumante/não fumante, doente/sadio.

**Ordinais**: existe uma ordenação entre as categorias.

**Exemplos**: escolaridade (1o, 2o, 3o graus), estágio da doença (inicial, intermediário, terminal), mês de observação (janeiro, fevereiro, ..., dezembro).

# 2 Estatistica

A Estatística é o "ramo da matemática que trata da coleta, da análise, da interpretação e da apresentação de massas de dados numéricos".

Desta forma, a Estatística é também importante parte da Ciência de Dados, utilizada para analisar uma série de situações e problemas através de seus mais variados conceitos.

# 3 Estatística descritiva

Na prática geralmente trabalhamos com bases de dados muito grandes, sendo inviável analisar cada um dos dados individualmente.

Para nos ajudar neste problema, utilizamos a estatística descritiva, que tem o objetivo de descrever e sumarizar um conjunto de dados.

As estatísticas descritivas mais comuns, que são extraídas de variáveis contínuas ou discretas, são as que buscam responder às questões:

**Medidas de Tendência Central**:

em que região do domínio da variável a amostra obtida tende a se concentrar?

**Medidas de Dispersão**:

o quão espalhada a variável está em torno dessa região de concentração?

**Associação**:

qual o grau e sentido da relação entre essa variável e as demais disponíveis?

# 4 Medidas de Tendência Central

As medidas de tendência central definem valores significativos, representativos e adequados para um conjunto de dados, dependendo do que se deseja analisar.

São elas: * média, * mediana, * quantis e * moda.

## 4.1 Média

A média é uma medida de tendência central que indica o valor onde estão concentrados os dados de um conjunto de valores, representando um valor significativo para o mesmo.

$$\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{1}$$

No caso dos Dataframes do Pandas, a média é calculada através da função **mean()** dos Dataframes, que também está presente nas Series do Pandas. Vamos criar uma Series do Pandas e calcular sua média, e também vamos calcular a idade média dos passageiros do Titanic:

[ ]:
```
import pandas as pd
import numpy as np
```

```
import requests

train_df = pd.read_csv('train.csv')

example_series =  pd.Series([1,5,10,30,50,30,15,40,45])

print(train_df['age'].mean())

print(example_series.mean())
```

```
29.69911764705882
25.11111111111111
```

### 4.2 Mediana e Quantil

A mediana é o valor que separa a metade superior da metade inferior de uma distribuição de dados, ou o valor no centro da distribuição.

Na prática, se o número de observações na distribuição é ímpar, ele é o valor central, e se o número de observações é par, ele será a média das duas observações mais centrais.

Ela é calculada através da função **median()**, novamente, presente em dataframes e series.

Vejamos abaixo:

[ ]:
```
print(train_df['age'].median())
```

```
28.0
```

[ ]:
```
print(example_series.median())
```

```
30.0
```

A mediana é um conceito menos suscetível a grandes valores discrepantes do que a média.

Se o número de observações não é muito grande e você tem uma observação que é muito maior do que os outros, a sua média pode começar a ficar menos representativa com relação à maioria do seu grupo.

Por exemplo, se você está analisando o rendimento de uma classe da faculdade e um deles é um milionário, enquanto o restante é o trabalhador médio da empresa, a mediana será, provavelmente, uma melhor representação dos rendimentos do grupo como um todo, uma vez que a média estará "contaminada" pelo valor discrepante.

#### 4.2.1 Medidas separatrizes

Em um conjunto de dados ordenados, separa os dados em n partes de tamanho igual.

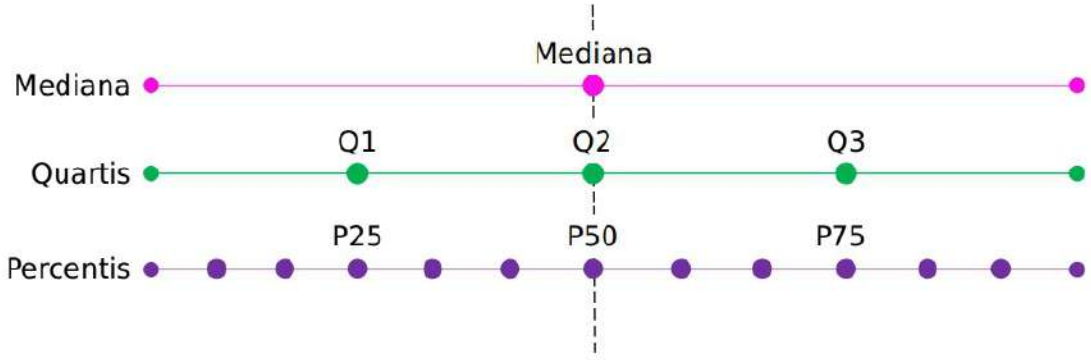
- **Percentis**: dividem o conjunto de dados em 100 partes iguais, ou seja, em pedaços de tamanhos iguais que contêm 1% dos dados.
- **Quartis**: dividem o conjunto de dados em 4 partes, ou seja, em pedaços de tamanhos iguais que contém 25% dos dados.

- **Mediana**: dividem o conjunto de dados em 2 partes. Acima da mediana estão 50% dos dados e abaixo dela também.

A mediana é o segundo quartil (Q2) e também o percentil 50 (P50):

[5]:
```
from IPython.display import Image
Image('download_4.jpg')
```

[5]:




### 4.3 Quantil

O quantil pode ser entendido como uma generalização da mediana.

O quantil é o valor abaixo do qual está um certo percentual dos dados.

No caso da mediana, esse percentual é de 50%.

Vejamos o código para o quantil, que pode ser calculado através da função **quantile()**.

Esta função, por padrão, adota o percentual (representado através do parâmetro q) de 50%, ou seja, é uma mediana por padrão.

Você pode configurar outros percentuais utilizando esse mesmo parâmetro:

[ ]:
```
print(train_df['age'].quantile())
```

```
28.0
```

[ ]:
```
print(example_series.quantile())
```

```
30.0
```

#### 4.3.1 Valor Minimo

maior do que 0% - valor mínimo

[ ]:
```
print(train_df['age'].quantile(q=0))
```

```
0.42
```

[ ]:
```
print(example_series.quantile(q=0))
```

```
1.0
```

#### 4.3.2 Primeiro Quartil

maior do que 25% - 1º quartil

```
[ ]: print(train_df['age'].quantile(q=0.25))
```

```
20.125
```

```
[ ]: print(example_series.quantile(q=0.25))
```

#### 4.3.3 Segundo Quartil

maior do que 50% - 2º quartil ou mediana

```
[ ]: print(train_df['age'].quantile(q=0.50))
```

```
28.0
```

```
[ ]: print(example_series.quantile(q=0.5))
```

```
30.0
```

#### 4.3.4 Terceiro Quartil

maior do que 75% - 3º quartil

```
[ ]: print(train_df['age'].quantile(q=0.75))
```

```
38.0
```

```
[ ]: print(example_series.quantile(q=0.75))
```

```
30.0
```

#### 4.3.5 Valor Máximo

maior do que 100% - valor máximo

```
[ ]: print(train_df['age'].quantile(q=1))
```

```
80.0
```

```
[ ]: print(example_series.quantile(q=1))
```

```
50.0
```

### 4.4 Moda

o valor que mais se repete dentro de um conjunto. No Pandas, a moda é calculada através da função **mode()**.

[ ]:
```
print(train_df['age'].mode())
```

```
0    24.0
dtype: float64
```

[ ]:
```
print(example_series.mode())
```

```
0    30
dtype: int64
```

Quando temos dois valores diferentes para a moda, a função retorna todos os valores, como no exemplo abaixo:

[ ]:
```
example_series_2 =  pd.Series([1,5,10,30,50,30,15,40,45,45])
print(example_series_2.mode())
```

```
0    30
1    45
dtype: int64
```

### 4.5 Medidas de Dispersão

São medidas que indicam o quão espalhados os dados estão, ou como eles variam. São: * amplitude, * variância, * desvio padrão e * desvio absoluto.

#### 4.5.1 Amplitude

A amplitude é a diferença entre o maior e o menor valor de um conjunto de dados.

Para fazer este cálculo usamos as funções **max()** e **min()**, que retornam o valor máximo e mínimo de um conjunto de dados, e depois subtrairemos um do outro:

[ ]:
```
print(train_df['age'].max() - train_df['age'].min())
```

```
79.58
```

[ ]:
```
print(example_series.max() - example_series.min())
```

```
49
```

#### 4.5.2 Variância

A variância mostra quanto os dados de um conjunto estão afastados de seu valor esperado.

$$var(x) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n} \tag{2}$$

A variância é calculada utilizando a função **var()** da biblioteca Pandas:

[ ]:
```
print(train_df['age'].var())
```

```
211.01912474630802
```

[ ]:
```
print(example_series.var())
```

```
325.1111111111111
```

### 4.5.3 Desvio Padrão

Indica quanto os dados estão afastados da média.

Um valor de desvio padrão alto indica que os valores estão mais espalhados, mais longe da média, e um desvio padrão baixo indica que os valores estão mais próximos da média.

$$\sigma = \sqrt{var(x)} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n}} \tag{3}$$

O Desvio Padrão é calculado utilizando a função **std()** da biblioteca Pandas: *texto em itálico*

[ ]:
```
print(train_df['age'].std())
```

```
14.526497332334042
```

[ ]:
```
print(example_series.std())
```

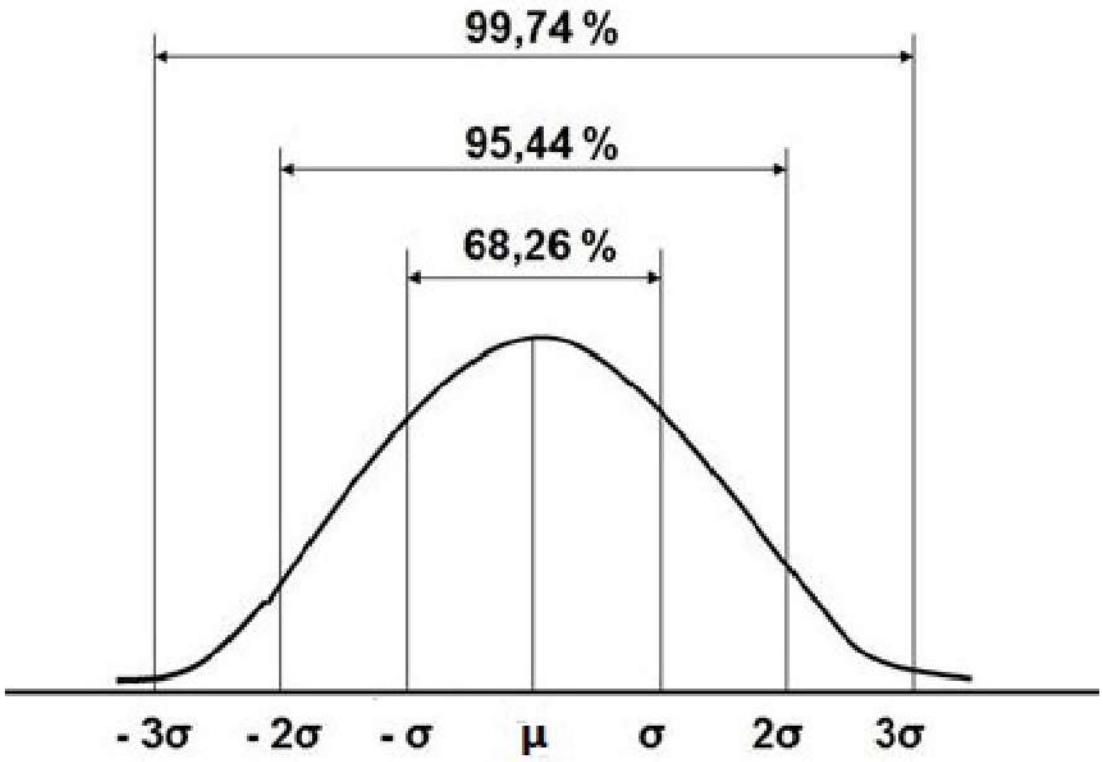
```
18.03083778173136
```

Os valores de desvio padrão também estão no conjunto dos números Reais Positivos, ou seja, se você encontrar um valor negativo, seus cálculos necessitam de revisão.

Considerando uma distribuição normal, 68% dos valores estão a 1 desvio padrão de distância da média:

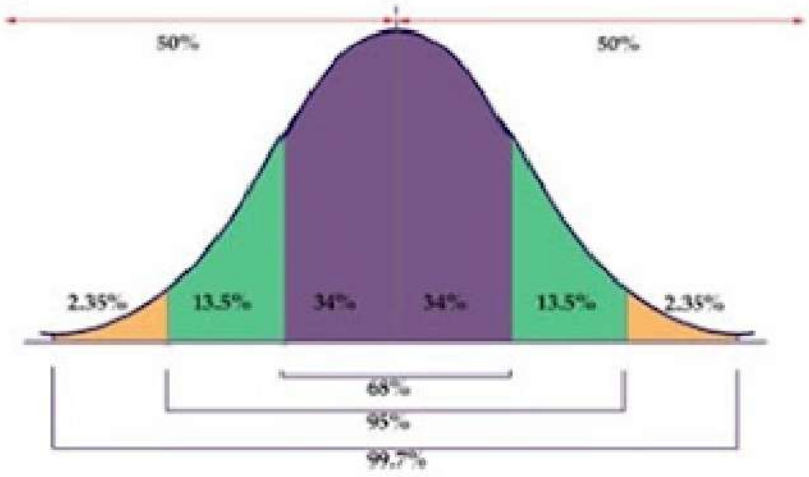
[3]:
```
Image('download_3.jpg')
```

[3]:




[7]:
```
Image('D11.jpg', width=500)
```

[7]:




Probabilidades na distribuição normal: regra empírica

### 4.5.4 Desvio absoluto

O Desvio Absoluto é calculado da seguinte forma: primeiro, encontramos a média dos valores; depois, calculamos a distância de cada ponto desta média; somamos as distâncias e dividimos o resultado pela média destas distâncias.

O Desvio Absoluto é calculado utilizando a função **mad()** da biblioteca Pandas:

[ ]:
```
print(train_df['age'].mad())
```

[ ]:
```
print(example_series.mad())
```

## 4.6 Medidas de associação

Existem duas medidas de associação comumentemente usadas, são elas:

- Covariância;
- Correlação.

### 4.6.1 Covariância

Em alguns momentos, queremos saber se duas variáveis possuem alguma relação entre si dentro de um conjunto de dados.

A covariância é uma medida numérica que indica a inter-dependência entre duas variáveis.

A covariância indica como duas variáveis se comportam conjuntamente em relação às suas médias.

Uma covariância igual a 0 indica que as duas variáveis são totalmente independentes, enquanto que uma covariância alta e positiva indica que uma variável é grande quando a outra é grande.

Analogamente, uma covariância negativa e com valor absoluto alto indica que uma variável é pequena quando a outra é grande.

A coverância entre duas variáveis x e y é expressa pela seguinte equação matemática

$$cov(x, y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{n} \tag{4}$$

A covariância pode ser calculada no Pandas através da função **cov()**. Ela retorna uma matriz indicando a covariância de cada coluna com outra:

[ ]:
```
print(train_df.cov())
```

```
          survived     pclass         age     sibsp     parch         fare
survived  0.236772  -0.137703   -0.551296 -0.018954  0.032017     6.221787
pclass   -0.137703   0.699015   -4.496004  0.076599  0.012429   -22.830196
age      -0.551296  -4.496004  211.019125 -4.163334 -2.344191    73.849030
sibsp    -0.018954   0.076599   -4.163334  1.216043  0.368739     8.748734
parch     0.032017   0.012429   -2.344191  0.368739  0.649728     8.661052
fare      6.221787 -22.830196   73.849030  8.748734  8.661052  2469.436846
```

### 4.6.2 Correlação

A covariância, entretanto, pode ser difícil de ser compreendida e comparada, pois ela nos dá valores em escalas que serão diferentes conforme as variáveis mudem.

Para uma melhor comparação, normalizamos a covariância para ter um valor que sempre estará entre 1 e -1, que é a correlação. Logo, a correlação também é outra medida que indica o quanto duas variáveis estão relacionadas.

Seu valor fica sempre entre -1, que indica uma anti-correlação perfeita, e 1, que indica uma correlação perfeita.

A correlação entre duas variáveis x e y é expressa pela seguinte equação:

$$\rho(x,y) = \frac{cov(x,y)}{\sigma(x)\cdot\sigma(y)} = \frac{\sum_{i=1}^{n}(x_i-\overline{x})(y_i-\overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\overline{x})^2}\cdot\sqrt{\sum_{i=1}^{n}(x_i-\overline{x})^2}} \tag{5}$$

Calculamos a correlação no Pandas com a função **corr()**, que assim como a função **cov()**, irá retornar uma matriz com a correlação de cada coluna com as outras colunas do Dataframe:

[ ]:
```
print(train_df.corr())
```

```
          survived    pclass       age     sibsp     parch      fare
survived  1.000000 -0.338481 -0.077221 -0.035322  0.081629  0.257307
pclass   -0.338481  1.000000 -0.369226  0.083081  0.018443 -0.549500
age      -0.077221 -0.369226  1.000000 -0.308247 -0.189119  0.096067
sibsp    -0.035322  0.083081 -0.308247  1.000000  0.414838  0.159651
parch     0.081629  0.018443 -0.189119  0.414838  1.000000  0.216225
fare      0.257307 -0.549500  0.096067  0.159651  0.216225  1.000000
```

## 4.7 Medidas de forma

As medidas de forma trazem informações sobre o formato da distribuição.

### 4.7.1 Assimetria

A assimetria (skewness), $\gamma$, é a medida de assimetria da função de distribuição de probabilidade de uma variável aleatória em torno de sua média.

Se a assimetria for negativa, a distribuição tende para a direita, mas, se for positiva, tende para a esquerda.

A assimetria é calculada da seguinte maneira:

$$\gamma = \frac{E(x-\overline{x})^3}{\sigma^3} \tag{6}$$

onde - $\sigma$ é o desvio padrão da variável x, - $\overline{x}$ é a média da variável x e - E, o valor esperado ou esperança da variável.

O valor esperado, também chamado esperança , de uma variável aleatória é a soma do produto de cada probabilidade de saída da experiência pelo seu respectivo valor.

Isto é, representa o valor médio "esperado" de uma experiência se ela for repetida muitas vezes.

Note-se que o valor em si pode não ser esperado no sentido geral; pode ser improvável ou impossível. Se todos os eventos tiverem igual probabilidade o valor esperado é a média aritmética.

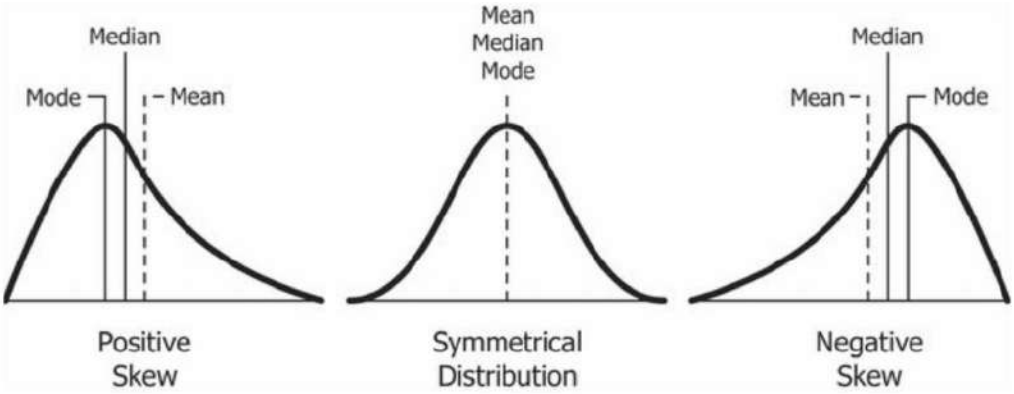
Para dados normalmente distribuídos, a assimetria é bem próxima de zero.

Uma assimetria positiva, com valor $> 0$, significa que a cauda da distribuição está mais para a direita. Ou seja, que os valores de moda $<$ mediana $<$ média.

Uma assimetria negativa, com valor $< 0$, significa que a cauda da distribuição está mais para a esquerda. Ou seja, que os valores de média $<$ mediana $<$ moda.

[4]:
```
Image('skewness.png')
```

[4]:




### 4.7.2 Curtose

A curtose (kurtosis), , é a medida de dispersão que caracteriza o pico ou achatamento da curva da função de distribuição de probabilidade de uma variável aleatória.

Se a curtose for igual a zero, então o pico da curva possui achatamento similar à distribuição normal; se a curtose for positiva, então o pico da função é mais afunilado e a função é mais concentrada; e se a curtose for negativa, então o pico da função é mais achatado e a função, mais dispersa.

A curtose é dada pela equação:

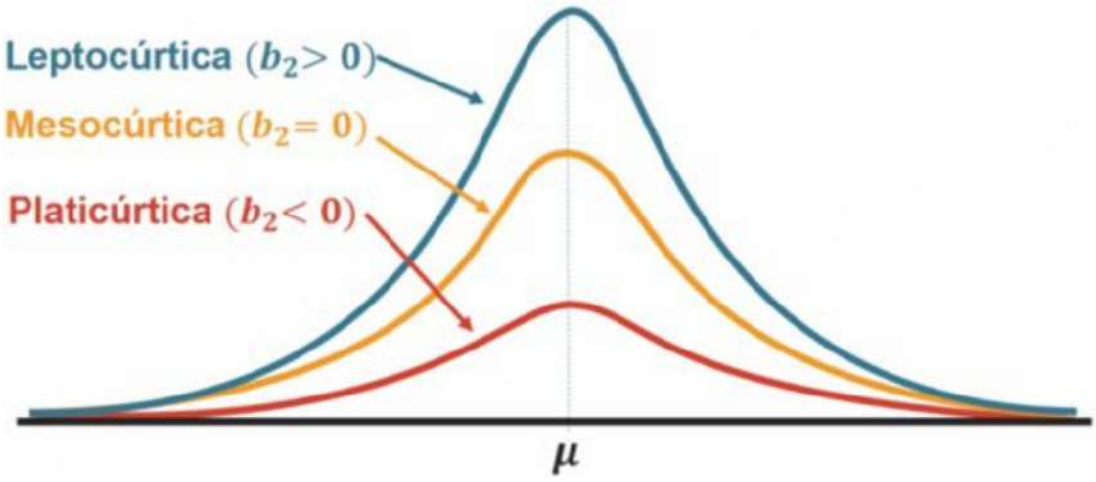
$$\beta = \frac{E(x - \overline{x})^4}{\sigma^4} \tag{7}$$

onde $\sigma$ é o desvio padrão da variável x, $\overline{x}$ é a média da variável x e E, o valor esperado ou esperança da variável.

A Figura ilustra o exemplo de curvas com diferentes valores de curtose.

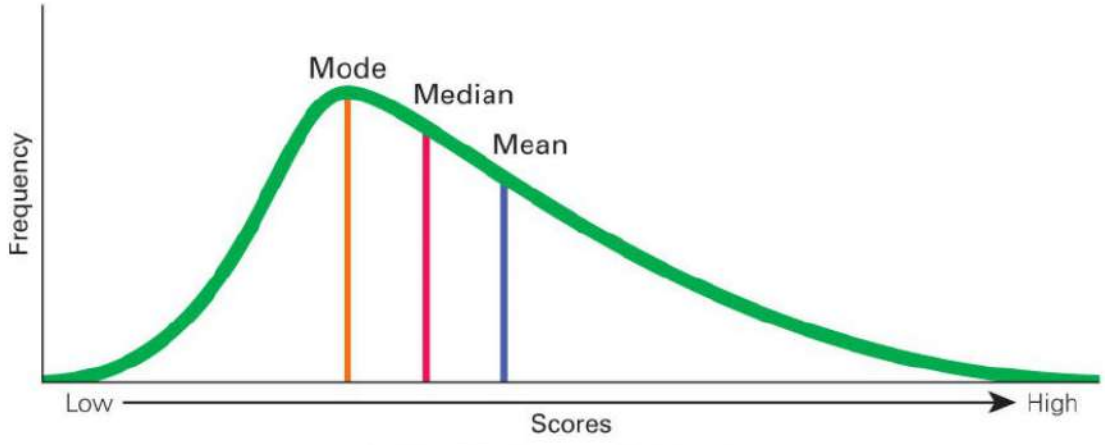
[2]:
```
Image('download_2.jpg')
```

[2]:




[8]:
```
Image('1_XU3Kdl521XnWHECHZ7XOaQ.jpg')
```

[8]:




**(a) Right-skewed distribution**


**(b) Left-skewed distribution**

# 5 Distribuições de probabilidade

As distribuições podem ser:

- Contínuas
  - Exemplos: medidas de comprimento, tensão elétrica, valores de uma ação, etc.
- Discretas:
  - Exemplos: número de vezes que uma máquina apresenta um problema em um periodo.

Quando criamos um número aleatório, precisamos definir o tipo de distribuição de probabilidade que originou aquele valor, assim temos:

- Distribuição Normal ou Gaussiana nelo o número aleatório pode variar entre $-\infty$ e $+\infty$, contudo é muito provável que o valor esteja entre -3 e +3.
- Distribuição Uniforme, nela o número randômico estará sempre entre 0 e 1.

[2]:
```
#gera e imprime números aleatórios
import numpy as np
#produz 5 números aleatórios
#dist. uniforme entre 0 e 1
x_unif=np.random.uniform(0,1,5)
print('uniforme: ',x_unif)

#dist. normal com média 0 e desvio padrão 1
x_norm=np.random.normal(0,1,5)
print('Normal: ',x_norm)
```

```
uniforme:  [0.40449335 0.21536541 0.36635171 0.38711118 0.96491955]
Normal:  [ 0.66437289 -2.49516575 -1.83580709  0.73667706 -0.63014416]
```

# 6 Histograma

É uma forma de representar um conjunto de dados. Em um histograma temos a frequencia em que valores se apresentam em um intervalo, chamado de bin.

Os gráficos de um dado podem ser representados em um gráfico de dispersão e em um histograma, mas no gráfico de dispersão não representa muito para nós.

Assim, vamos representar os dados com um histograma.

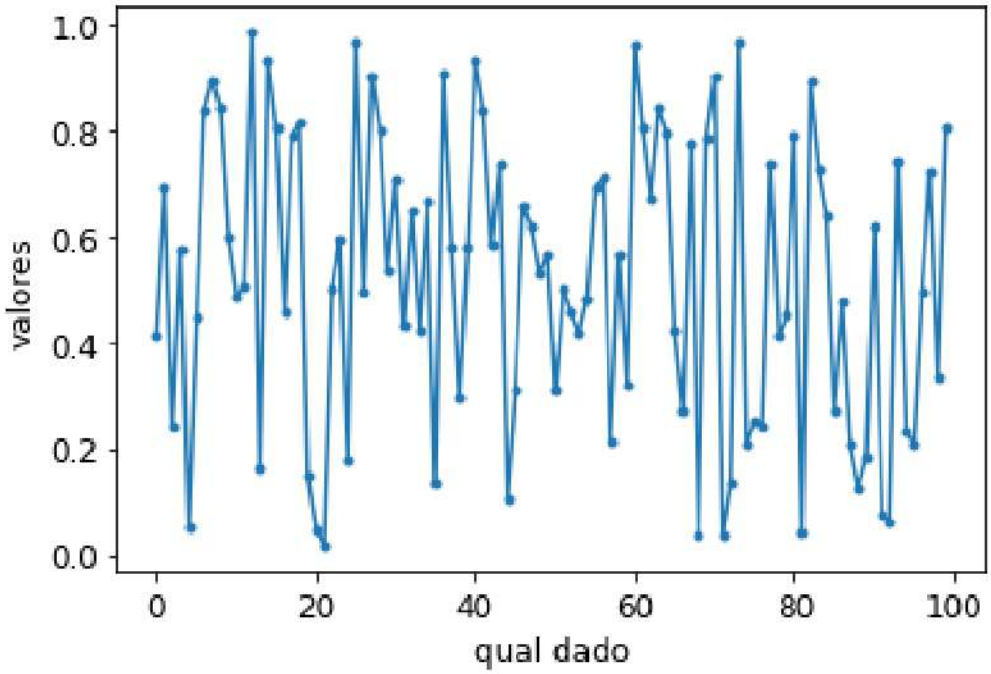
O comando "plt.hist()" gera um histograma de 10 bins como padrão.

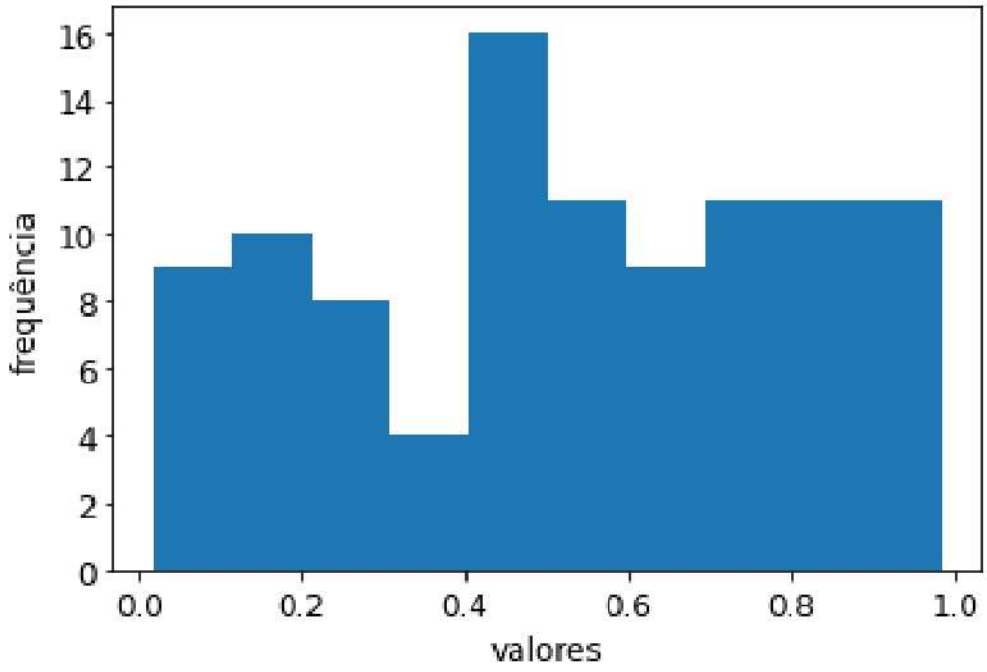
**Caixas (bins) são intervalos igualmente espaçados que são usados para classificar os dados em gráficos.**

[6]:
```
# Graficos e histogramas simples
```

```
#dist. uniforme
import numpy as np
import matplotlib.pyplot as plt
#produz 100 numeros aleatórios de 0 a 1
x=np.random.uniform(0,1,100)
# faz um gráfico dos valores
plt.rcParams.update({'font.size': 12})
plt.figure()
plt.plot(x,'.-')
plt.xlabel('qual dado')
plt.ylabel('valores')
# faz um histograma dos valores
plt.figure()
plt.hist(x)
plt.xlabel('valores')
plt.ylabel('frequência')
```

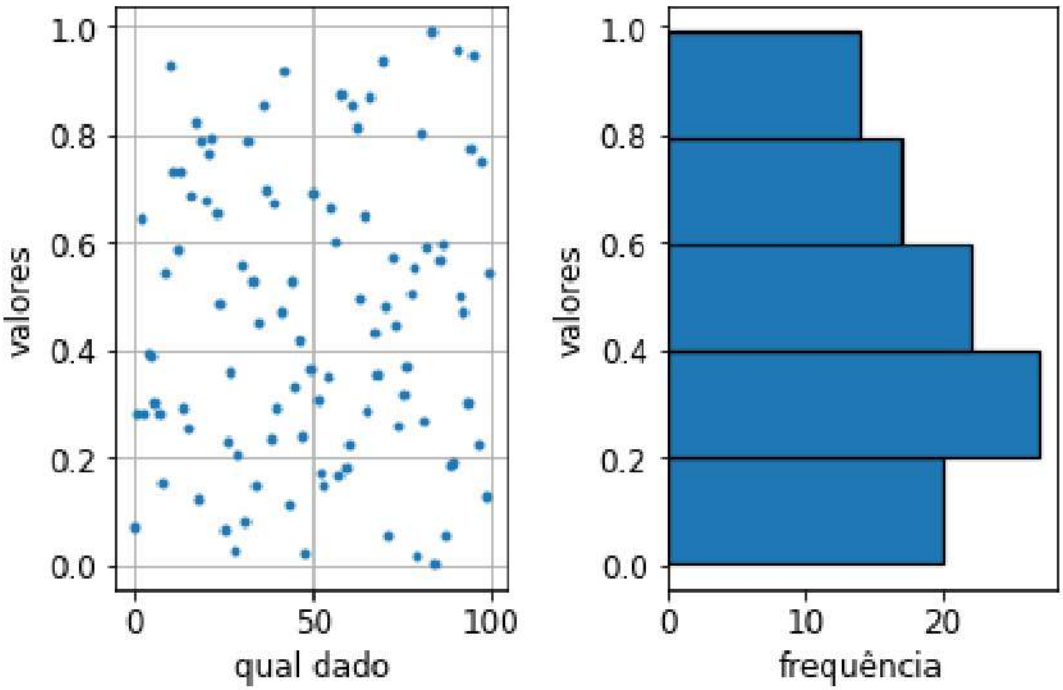
[6]: `Text(0, 0.5, 'frequência')`





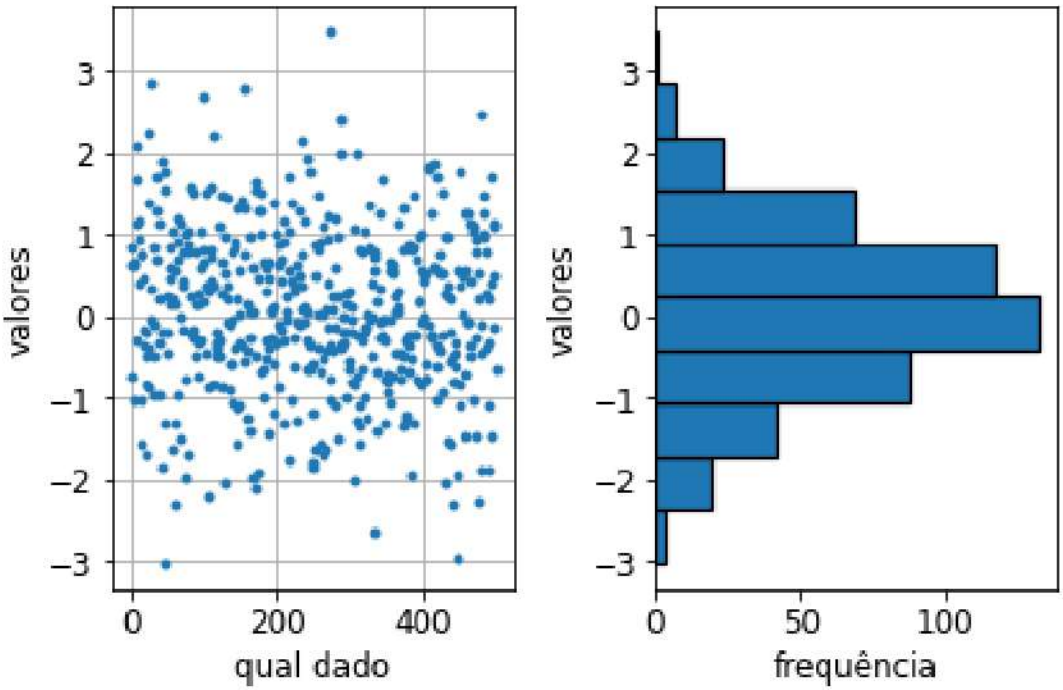

## 7 Relação entre dados e histograma

[7]:
```
#gráfico e histogramas para ajudar a interpretar
#dist. uniforme
import numpy as np
import matplotlib.pyplot as plt
#produz 100 numeros aleatórios de 0 a 1
x=np.random.uniform(0,1,100)
# gráfico e histograma em uma figura só
plt.rcParams.update({'font.size': 12})
plt.figure()
plt.subplot(1,2,1)
plt.plot(x,'.'),plt.grid()
plt.xlabel('qual dado'),plt.ylabel('valores')
plt.subplot(1,2,2)
plt.hist(x,bins=5, orientation="horizontal",edgecolor='black', linewidth=1.2)
plt.xlabel('frequência'),plt.ylabel('valores')
plt.tight_layout()
```




Utilizando uma distribuição gaussiana e mostrando os mesmos valores

[10]:
```
#gráfico e histogramas para ajudar a interpretar
#dist. normal
import numpy as np
import matplotlib.pyplot as plt
#produz 300 numeros aleatórios de média 0 e desvio padrão 1
x=np.random.normal(0,1,500)
# gráfico e histograma em uma figura só
plt.rcParams.update({'font.size': 12})
plt.figure()
plt.subplot(1,2,1)
plt.plot(x,'.'),plt.grid()
plt.xlabel('qual dado'),plt.ylabel('valores')
plt.subplot(1,2,2)
plt.hist(x,bins=10, orientation="horizontal",edgecolor='black', linewidth=1.2)
plt.xlabel('frequência'),plt.ylabel('valores')
plt.tight_layout()
```




## 8 Representação matemática de uma Distribuição Normal ou gaussiana

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\overline{x}}{\sigma})^2} \tag{8}$$

Ajustado pela média $\overline{x}$ e pelo desvio padrão $\sigma$, com uma normalização dada por $\frac{1}{\sigma\sqrt{2\pi}}$.

## 9 Média e Devio padrão

Fixando a semente para manter os mesmos números aleatórios para poder ter os mesmos valores.

**np.random.seed(2022)**

### 9.1 Média

é a soma de todos os números dividido pela quantidade de números

$$\overline{x} = \frac{x_1 + x_2 + x_3 + \cdots + x_n}{n} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{9}$$

Média, np.mean()

### 9.2 Desvio Padrão

Raiz quadrada da somatória da distância quadrática e divido por $n$ ou por $n-1$ (amostra)

$$\sigma = \sqrt{var(x)} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}} \tag{10}$$

[21]:
```
#medidas de posição e disperção
#dist. normal
import numpy as np
#produz 500 numeros aleatórios de média 0 e desvio padrão 1
np.random.seed(20221203)
x=np.random.normal(0,1,500) #dist. uniforme
media=np.mean(x)
d_pad=np.std(x,ddof=1)
d_pad
```

[21]:
```
0.985543373005187
```

[17]:
```
media
```

[17]:
```
-0.030737391942900177
```

A função *print* permite imprimir números e texto de fomra alternada e também fixar o número de casa depois da virgula que será impressa, neste caso colocamos 2 casas decimais depois da vírgula.

[18]:
```
print(f'média={media:.2f} e desvio padrão={d_pad:.2f}')
```

```
média=-0.03 e desvio padrão=0.99
```

## 10 Criando um Histograma Completo

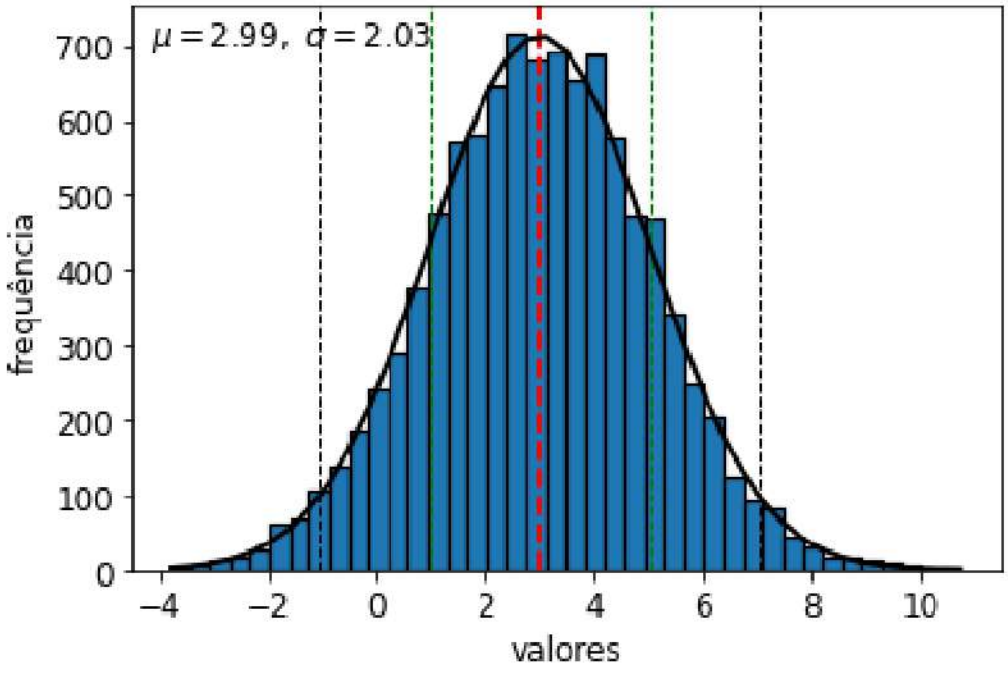
[29]:
```
# histograma
#dist. normal
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

#produz N números aleatórios de média 3 e desvio padrão 2
N=10000;
np.random.seed(20221203)
x=np.random.normal(3,2,N) #dist. uniforme com média 3 e desvio padrão 2
media=np.mean(x)
d_pad=np.std(x,ddof=1)

plt.rcParams.update({'font.size': 12})
n, bins,p  =plt.hist(x,bins=40,edgecolor='black', linewidth=1.2)
plt.xlabel('valores')
```

```
plt.ylabel('frequência')
plt.text(-4.2, 700, f'$\mu={media:.2f},\ \sigma={d_pad:.2f}$')
plt.axvline(media, color='r', linestyle='dashed', linewidth=2)
plt.axvline(media+d_pad, color='g', linestyle='dashed', linewidth=1)
plt.axvline(media-d_pad, color='g', linestyle='dashed', linewidth=1)
plt.axvline(media+2*d_pad, color='black', linestyle='dashed', linewidth=1)
plt.axvline(media-2*d_pad, color='black', linestyle='dashed', linewidth=1)
gauss = N*np.median(np.diff(bins))*norm.pdf(bins, media, d_pad)
plt.plot(bins, gauss, 'k-', linewidth=2)
```

[29]: [<matplotlib.lines.Line2D at 0x7f0858a90950>]




plt.axvline -> eixo vertical line

precisamos da biblioteca scipy.stats.norm para calcular a pdf da distribuição normal

## 11 Função Densidade de Probabilidade (PDF)

Em teoria das probabilidades e estatística, a função densidade de probabilidade (FDP), ou densidade de uma variável aleatória contínua, é uma função que descreve a verossimilhança de uma variável aleatória tomar um valor dado.

A probabilidade da variável aleatória cair em uma faixa particular é dada pela integral da densidade dessa variável sobre tal faixa - isto é, é dada pela área abaixo da função densidade mas acima do eixo horizontal e entre o menor e o maior valor dessa faixa.

A função densidade de probabilidade é não negativa sempre, e sua integral sobre todo o espaço é igual a um.

A função norm.pdf calcula a pdf, por definição tem área total 1, assim precisamos normaliza-lo multiplocando por N e pela largura dos bins, que calculamos usando "np.median(diff(bins))"

Preciso renormalizar a gaussiana, multiplicando pelo N e pelo tamanho do bin, calculo a diferença entre os bins e calculo a mediana da diferença entre os bins e armazeno em um vetor.

# 12 Calculando probabilidades conhecendo a PDF

O calculo de uma probabilidade em um intervalo utilizando a pdf é

## 12.1 *Probabilidade de um evento entre A e B* $= \int_A^B f(x)dx = \int_{-\infty}^B f(x)dx - \int_{-\infty}^A f(x)dx$

Calcular uma integral de A até B é calcular uma integral de menos infinito até um ponto e de menos infinito até o outro ponto.

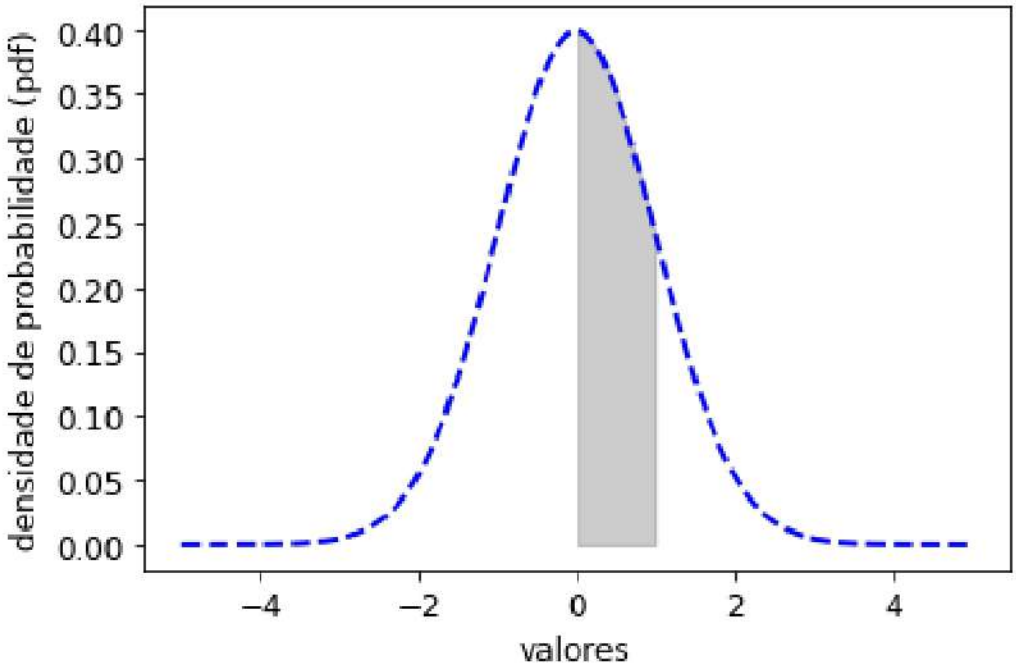
A função norm.cdf(B) vem de "cumulative" e calcula $\int_{-\infty}^B f(x)dx$ que corresponde a função cumulativa da função densidade.

[30]:
```
#calculando probabilidades
# calcula a probabilidade de encontrar um número entre A e B
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm
A=0;
B=1;
P_AB = norm.cdf(B)-norm.cdf(A)
print(f'prob entre {A:.1f} e {B:.1f} = {P_AB:.3f}')
```

```
prob entre 0.0 e 1.0 = 0.341
```

[31]:
```
bins=np.linspace(-5,5,1000)
gauss = norm.pdf(bins, 0, 1)
plt.rcParams.update({'font.size': 12})
plt.figure()
plt.plot(bins, gauss, 'b--', linewidth=2)
plt.xlabel('valores')
plt.ylabel('densidade de probabilidade (pdf)')
px=bins[(bins>A)&(bins<B)] # busco os bins maiores do que A e menores do que B␣
 ↪-> com isso consigo usar o fiil-between
plt.fill_between(px,gauss[(bins>A)&(bins<B)],color='k',alpha=0.2) #função␣
 ↪fill_between completar entre preciso saber os pontos de indice
```

[31]:
```
<matplotlib.collections.PolyCollection at 0x7f0858a209d0>
```

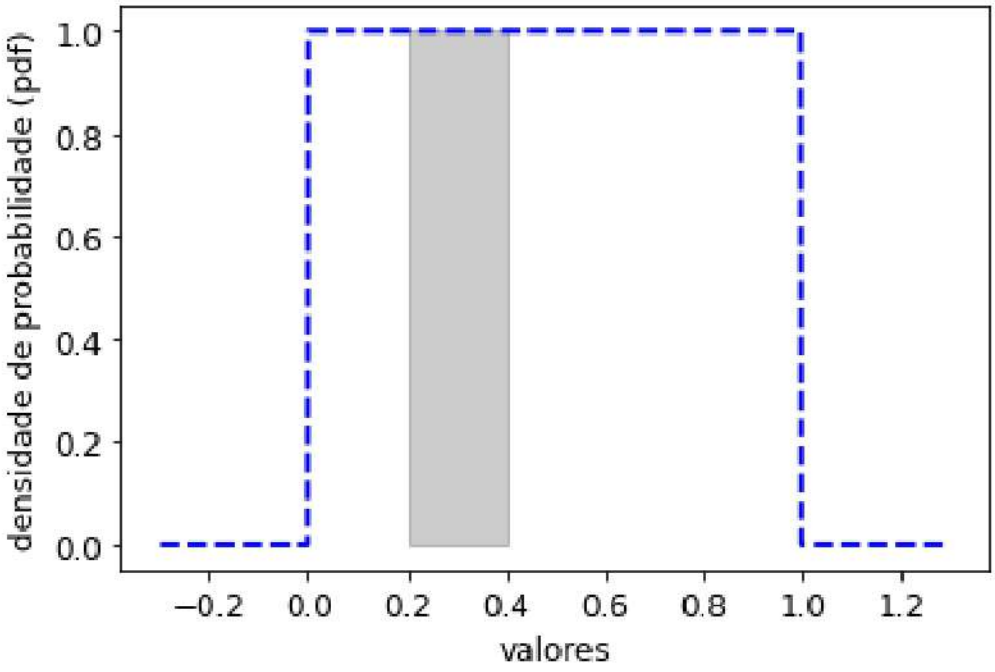



[32]:
```
#calculando probabilidades
# calcula a probabilidade de encontrar um número entre A e B
# dist uniform
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import uniform
A=0.2;
B=0.4;
P_AB = uniform.cdf(B)-uniform.cdf(A)
print(f'prob entre {A:.1f} e {B:.1f} = {P_AB:.3f}')
```

```
prob entre 0.2 e 0.4 = 0.200
```

[33]:
```
bins=np.linspace(-0.3,1.3,1000)
unif = uniform.pdf(bins, 0, 1)
plt.rcParams.update({'font.size': 12})
plt.figure()
plt.plot(bins, unif, 'b--', linewidth=2)
plt.xlabel('valores')
plt.ylabel('densidade de probabilidade (pdf)')
px=bins[(bins>A)&(bins<B)]
plt.fill_between(px,unif[(bins>A)&(bins<B)],color='k',alpha=0.2)
```

[33]:
```
<matplotlib.collections.PolyCollection at 0x7f0858930c50>
```




Confirmando, basta calcular a área em cinza, entre 0,2 e 0,4.

Corresponde a área de um retângulo:

base * altura $= 0,2 * 1 = 0,2$