

Amostragens

In [4]:

```
import pandas as pd
import random
import numpy as np
import random
import matplotlib.pyplot as plt
```

1 - Amostra aleatória simples

1.1 - Amostra aleatória simples sem repetição

Ex: um pesquisador precisa selecionar, aleatoriamente e sem repetição, alunos de uma sala de aula para uma entrevista.

In [5]:

```
# https://www.reformattext.com/sequential-number-generator.htm
c = [
1,2,3,4,5,6,7,8,9,10,
11,12,13,14,15,16,17,18,19,20,
21,22,23
]
a = random.sample(c, 3)
print("números escolhidos: ", a)
```

números escolhidos: [17, 15, 9]

1.2 tamanho da amostra

In [6]:

```
len(a)
```

Out[6]:

3

1.3 ordenar amostra

In [7]:

```
print("Amostra: ", a)
a.sort(reverse=True)
print("Amostra ordenada decrescente: ", a)
a.sort()
print("Amostra ordenada crescente: ", a)
```

Amostra: [17, 15, 9]

Amostra ordenada decrescente: [17, 15, 9]

Amostra ordenada crescente: [9, 15, 17]

1.4 Tamanho da amostra

In [8]:

```
max(a)
```

Out[8]:

17

In [9]:

```
min(a)
```

Out[9]:

9

1.5 Amplitude

In [10]:

```
max(a)-min(a)
```

Out[10]:

8

1.6 - Amostra aleatória simples com repetição

Um pesquisador está estudando a compatibilidade de aparelho de som portátil com um pendrive de capacidade acima de 512 GB. Uma loja deixou à sua disposição quatro exemplares de cada uma das seguintes marcas: JBL, LG e Multilaser. Ele precisa selecionar aleatoriamente quatro exemplares de uma dessas marcas, podendo haver repetição.

In [11]:

```
c = ["Multilaser", "LG", "Multilaser", "Multilaser", "LG", "JBL",  
     "LG", "JBL", "Multilaser", "LG", "JBL", "JBL",  
     "Multilaser"]  
a = random.choices(c, k=4)  
print(a)
```

```
['Multilaser', 'Multilaser', 'Multilaser', 'Multilaser']
```

1.7 - Amostra aleatória simples a partir de uma base de dados

1.7.1 Ler dados

Ex: trabalhar com a base de dados de Censo Americana.

In [12]:

```
dataset = pd.read_csv('004_census.csv', sep=',', encoding='iso-8859-1')  
dataset.shape
```

Out[12]:

```
(32561, 15)
```

1.7.2 Ver os primeiros cinco registros

In [13]:

```
dataset.head()
```

Out[13]:

	age	workclass	final-weight	education	education-num	marital-status	occupation	relationship	race
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black

1.7.3 Ver os últimos cinco registros

In [14]:

```
dataset.tail()
```

Out[14]:

	age	workclass	final-weight	education	education-num	marital-status	occupation	relationship	race
32556	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White
32557	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White
32558	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White
32559	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White
32560	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White

1.7.4 Função de amostragem aleatória simples

In [15]:

```
def amostragem_simples_aleatoria(dataset, amostras):
    return dataset.sample(n = amostras, random_state=1)
```

In [16]:

```
a = amostragem_simples_aleatoria(dataset, 100)
len(a)
```

Out[16]:

100

In [17]:

```
a.head()
```

Out[17]:

	age	workclass	final-weight	education	education-num	marital-status	occupation	relationship	race
9646	62	Self-emp-not-inc	26911	7th-8th	4	Widowed	Other-service	Not-in-family	White
709	18	Private	208103	11th	7	Never-married	Other-service	Other-relative	White
7385	25	Private	102476	Bachelors	13	Never-married	Farming-fishing	Own-child	White
16671	33	Private	511517	HS-grad	9	Married-civ-spouse	Prof-specialty	Husband	White
21932	36	Private	292570	11th	7	Never-married	Machine-op-inspct	Unmarried	White

1.8 Amostragem sistemática

In [18]:

```
def amostragem_sistemica(dataset, amostras):
    intervalo = len(dataset) // amostras
    random.seed(1)
    inicio = random.randint(0, intervalo)
    indices = np.arange(inicio, len(dataset), step = intervalo)
    amostra_sistemica = dataset.iloc[indices]
    return amostra_sistemica
```

In [19]:

```
dataset = pd.read_csv('004_census.csv', sep=',', encoding='iso-8859-1')
```

In [20]:

```
df_amostra_sistemica = amostragem_sistemica(dataset, 100)
```

In [21]:

```
df_amostra_sistemica.head()
```

Out[21]:

	age	workclass	final-weight	education	education-num	marital-status	occupation	relationship	race
68	49	Self-emp-inc	191681	Some-college	10	Married-civ-spouse	Exec-managerial	Husband	White
393	34	State-gov	98101	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White
718	22	Private	214399	Some-college	10	Never-married	Sales	Own-child	White
1043	44	Private	167005	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White
1368	52	Private	152234	HS-grad	9	Married-civ-spouse	Exec-managerial	Husband	Asian Pac Islande

In []: