Teoria Dados Absolutos e Relativos

A coleta de amostras de uma população pode ser:

Dados absolutos: não sofreram manipulação, no máximo contagem e ordenação.
Dados relativos: fácil entendimento, ajuda na comparação entre quantidades
Percentil Índices Coeficientes Taxas

In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
dados={'emprego':['Administrador','Programador','Executivo'],
      'São Paulo':[700, 300, 45],
      'Minas Gerais':[1200,200,9]
      }
type(dados)
```

Out[2]:
```
dict
```

In [3]:
```
dataset=pd.DataFrame(dados)
dataset
```

Out[3]:

	emprego	São Paulo	Minas Gerais
0	Administrador	700	1200
1	Programador	300	200
2	Executivo	45	9

In [4]:
```
dataset['São Paulo'].sum()
```

Out[4]:
```
1045
```

In [5]:
```
dataset['Minas Gerais'].sum()
```

Out[5]:
```
1409
```

In [6]:
```
dataset['%_SP']=(dataset['São Paulo'])/dataset['São Paulo'].sum()
```

In [7]:
```
dataset
```

Out[7]:

	emprego	São Paulo	Minas Gerais	%_SP
0	Administrador	700	1200	0.669856
1	Programador	300	200	0.287081
2	Executivo	45	9	0.043062

In [8]:
```
dataset['%_SP']*=100
dataset
```

Out[8]:

	emprego	São Paulo	Minas Gerais	%_SP
0	Administrador	700	1200	66.985646
1	Programador	300	200	28.708134
2	Executivo	45	9	4.306220

In [9]:

```
dataset['%_MG']=(dataset['São Paulo'])/dataset['São Paulo'].sum()*100
```

In [10]:

```
#dataset.drop(columns=['% MG'])
```

In [11]:

```
dataset
```

Out[11]:

	emprego	São Paulo	Minas Gerais	%_SP	%_MG
0	Administrador	700	1200	66.985646	66.985646
1	Programador	300	200	28.708134	28.708134
2	Executivo	45	9	4.306220	4.306220

Base census dados relativos entre educacao e renda

Método convencional e direto

In [12]:

```
dataset_censo=pd.read_csv('data_base/census.csv')
dataset_censo
```

Out[12]:

	age	workclass	final-weight	education	education-num	marital-status	occupation	relationsl
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-fam
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husba
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-fam
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husba
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	W
...	...	...	...	...	...	...	...	
32556	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	W
32557	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husba
32558	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarr
32559	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-ch
32560	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	W

32561 rows × 15 columns

In [13]:

```
dataset_ed_inc=dataset_censo[['education','income']]
dataset_ed_inc
```

Out[13]:

	education	income
0	Bachelors	<=50K
1	Bachelors	<=50K
2	HS-grad	<=50K
3	11th	<=50K
4	Bachelors	<=50K
...	...	...
32556	Assoc-acdm	<=50K
32557	HS-grad	>50K
32558	HS-grad	<=50K
32559	HS-grad	<=50K
32560	HS-grad	>50K

32561 rows × 2 columns

In [14]:

```
dataset_group=dataset_ed_inc.groupby(['education','income'])['education'].count(
dataset_group
```

Out[14]:

```
education      income
 10th           <=50K     871
                >50K       62
 11th           <=50K    1115
                >50K       60
 12th           <=50K     400
                >50K       33
 1st-4th        <=50K     162
                >50K        6
 5th-6th        <=50K     317
                >50K       16
 7th-8th        <=50K     606
                >50K       40
 9th            <=50K     487
                >50K       27
 Assoc-acdm     <=50K     802
                >50K      265
 Assoc-voc      <=50K    1021
                >50K      361
 Bachelors      <=50K    3134
                >50K     2221
 Doctorate      <=50K     107
                >50K      306
 HS-grad        <=50K    8826
                >50K     1675
 Masters        <=50K     764
                >50K      959
 Preschool      <=50K      51
 Prof-school    <=50K     153
                >50K      423
 Some-college   <=50K    5904
                >50K     1387
Name: education, dtype: int64
```

In [15]:

```
dataset_group.index
```

Out[15]:

```
MultiIndex([(        ' 10th', ' <=50K'),
            (        ' 10th',  ' >50K'),
            (        ' 11th', ' <=50K'),
            (        ' 11th',  ' >50K'),
            (        ' 12th', ' <=50K'),
            (        ' 12th',  ' >50K'),
            (     ' 1st-4th', ' <=50K'),
            (     ' 1st-4th',  ' >50K'),
            (     ' 5th-6th', ' <=50K'),
            (     ' 5th-6th',  ' >50K'),
            (     ' 7th-8th', ' <=50K'),
            (     ' 7th-8th',  ' >50K'),
            (         ' 9th', ' <=50K'),
            (         ' 9th',  ' >50K'),
            (  ' Assoc-acdm', ' <=50K'),
            (  ' Assoc-acdm',  ' >50K'),
            (   ' Assoc-voc', ' <=50K'),
            (   ' Assoc-voc',  ' >50K'),
            (   ' Bachelors', ' <=50K'),
            (   ' Bachelors',  ' >50K'),
            (   ' Doctorate', ' <=50K'),
            (   ' Doctorate',  ' >50K'),
            (     ' HS-grad', ' <=50K'),
            (     ' HS-grad',  ' >50K'),
            (     ' Masters', ' <=50K'),
            (     ' Masters',  ' >50K'),
            (   ' Preschool', ' <=50K'),
            ( ' Prof-school', ' <=50K'),
            ( ' Prof-school',  ' >50K'),
            (' Some-college', ' <=50K'),
            (' Some-college',  ' >50K')],
           names=['education', 'income'])
```

In [16]:

```
dataset_group[' Bachelors', ' <=50K'], dataset_group[' Bachelors', ' >50K']
```

Out[16]:

```
(3134, 2221)
```

In [17]:

```
testdataset=pd.DataFrame(dataset_group)
```

In [18]:

```
for x,y in dataset_group.index:
    print(x,y)
```

```
10th  <=50K
10th  >50K
11th  <=50K
11th  >50K
12th  <=50K
12th  >50K
1st-4th  <=50K
1st-4th  >50K
5th-6th  <=50K
5th-6th  >50K
7th-8th  <=50K
7th-8th  >50K
9th  <=50K
9th  >50K
Assoc-acdm  <=50K
Assoc-acdm  >50K
Assoc-voc  <=50K
Assoc-voc  >50K
Bachelors  <=50K
Bachelors  >50K
Doctorate  <=50K
Doctorate  >50K
HS-grad  <=50K
HS-grad  >50K
Masters  <=50K
Masters  >50K
Preschool  <=50K
Prof-school  <=50K
Prof-school  >50K
Some-college  <=50K
Some-college  >50K
```

resolucao torta

In [19]:
```
censo_segmentado=pd.DataFrame(dataset_censo['education']).join(pd.DataFrame(data
```

In [20]:
```
censo_segmentado
```

Out[20]:

	education	income
0	Bachelors	<=50K
1	Bachelors	<=50K
2	HS-grad	<=50K
3	11th	<=50K
4	Bachelors	<=50K
...	...	...
32556	Assoc-acdm	<=50K
32557	HS-grad	>50K
32558	HS-grad	<=50K
32559	HS-grad	<=50K
32560	HS-grad	>50K

32561 rows × 2 columns

In [21]:

```
np.unique(censo_segmentado['education'].values, return_counts=True)[1]
```

Out[21]:

```
array([  933,  1175,   433,   168,   333,   646,   514,  1067,  1382,
        5355,   413, 10501,  1723,    51,   576,  7291], dtype=int64)
```

In [22]:

```
censo_segmentado['education'].values.astype(str)
```

Out[22]:

```
array([' Bachelors', ' Bachelors', ' HS-grad', ..., ' HS-grad',
       ' HS-grad', ' HS-grad'], dtype='<U13')
```

In [23]:

```
censo_segmentado.groupby('education',sort=True).count()
```

Out[23]:

	income
education	
10th	933
11th	1175
12th	433
1st-4th	168
5th-6th	333
7th-8th	646
9th	514
Assoc-acdm	1067
Assoc-voc	1382
Bachelors	5355
Doctorate	413
HS-grad	10501
Masters	1723
Preschool	51
Prof-school	576
Some-college	7291

In [24]:

```
censo_segmentado.groupby('income').count()
```

Out[24]:

	education
income	
<=50K	24720
>50K	7841

In [25]:

```
censo_segmentado[censo_segmentado['income']==' <=50K']
```

Out[25]:

	education	income
0	Bachelors	<=50K
1	Bachelors	<=50K
2	HS-grad	<=50K
3	11th	<=50K
4	Bachelors	<=50K
...	...	...
32553	Masters	<=50K
32555	Some-college	<=50K
32556	Assoc-acdm	<=50K
32558	HS-grad	<=50K
32559	HS-grad	<=50K

24720 rows × 2 columns

In [26]:

```
censo_segmentado[censo_segmentado['income']==' >50K'][censo_segmentado['educatio
```

```
C:\Users\WILLIAM\AppData\Local\Temp\ipykernel_30628\1836430540.py:1: UserWarning:
Boolean Series key will be reindexed to match DataFrame index.
  censo_segmentado[censo_segmentado['income']==' >50K'][censo_segmentado['educati
on']==' Bachelors'].groupby('education').count()
```

Out[26]:

	income
education	
Bachelors	2221

In [27]:

```
censo_segmentado[censo_segmentado['education']==' Bachelors']
```

Out[27]:

	education	income
0	Bachelors	<=50K
1	Bachelors	<=50K
4	Bachelors	<=50K
9	Bachelors	>50K
11	Bachelors	>50K
...	...	...
32530	Bachelors	>50K
32531	Bachelors	<=50K
32533	Bachelors	>50K
32536	Bachelors	>50K
32538	Bachelors	>50K

5355 rows × 2 columns

Coeficiente e Taxa de Variação

In [28]:
```
tabela={'Ano Escolar':['1°','2°','3°','4°'],'matriculas março':[70,50,47,23],'ma
tabela
```

Out[28]:
```
{'Ano Escolar': ['1°', '2°', '3°', '4°'],
 'matriculas março': [70, 50, 47, 23],
 'matriculas abril': [65, 48, 40, 22]}
```

In [29]:
```
dataset_tabela=pd.DataFrame(tabela)
dataset_tabela
```

Out[29]:

	Ano Escolar	matriculas março	matriculas abril
0	1°	70	65
1	2°	50	48
2	3°	47	40
3	4°	23	22

A ideia é que o coeficiente de variação pelo total, a taxa de variação pode ser estudada em cima de um montante, nesse caso abaixo temos o 1° ano escolar com uma taxa de desistência de 7.14 a cada 100 pessoas ou 70 a cada 1000, ou seja é uma portagem do quando diminuiu em relação a março

taxa desistência= variação março - abril/matricula março * 100
coeficiente= variação março - abril/matricula março

In [30]:
```
dataset_tabela['taxa desistencia']=(dataset_tabela['matriculas março']-dataset_t
```

In [31]:

```
dataset_tabela
```

Out[31]:

	Ano Escolar	matriculas março	matriculas abril	taxa desistencia
0	1°	70	65	7.142857
1	2°	50	48	4.000000
2	3°	47	40	14.893617
3	4°	23	22	4.347826

In [ ]: