CENTRO UNIVERSITÁRIO UNISATC

Estatística Aplicada

Amostragem por grupos

Prof. Me. Max Gabriel Steiner

AMOSTRAGEM POR GRUPOS

0 

1 

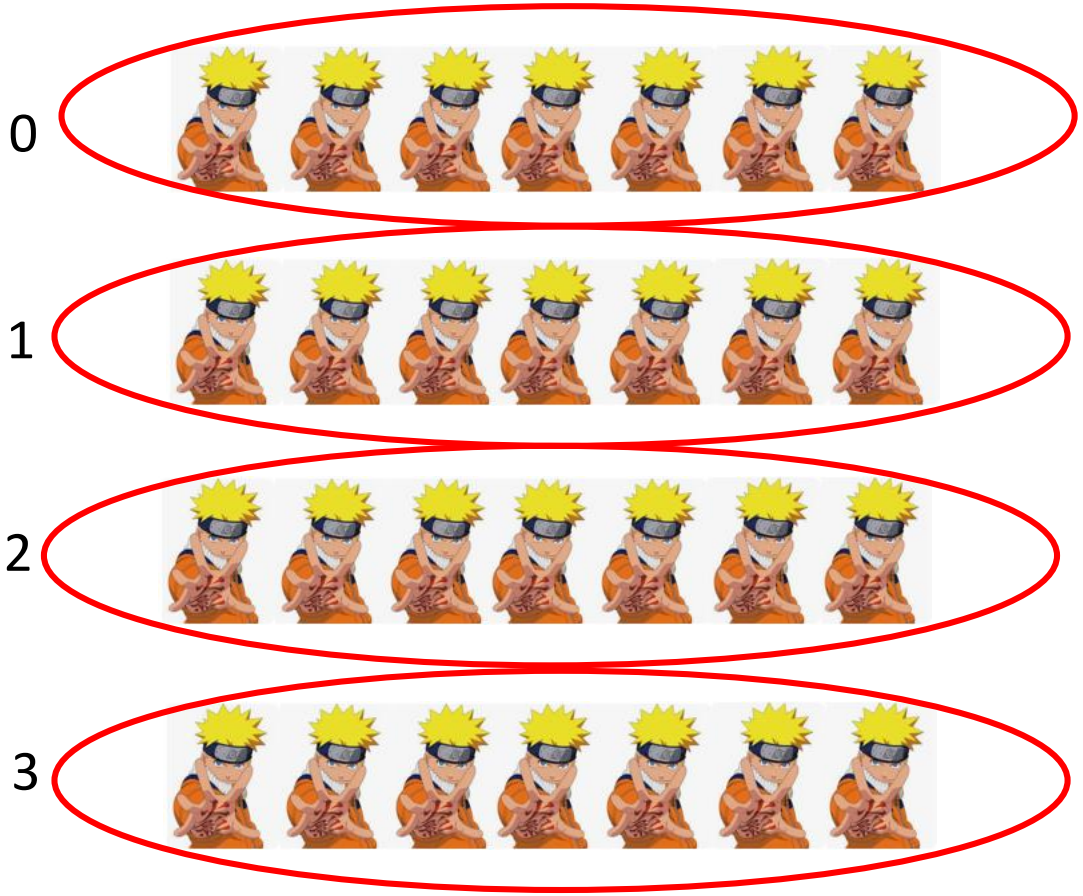
2 

3 

População: 28 narutos
4 grupos

Selecionar randomicamente um dos grupos

AMOSTRAGEM POR GRUPOS



População: 28 narutos
4 grupos

Selecionar randomicamente um dos grupos

Amostragem por grupos

+ Código + Texto

```
1 len(dataset)
```

0s

```
32562
```

Amostragem por grupos

```
1 len(dataset)//10
```

```
3256
```

```
grupos = []
id_grupo = 0
contagem = 0
for _ in dataset.iterrows():
  grupos.append(id_grupo)
  contagem += 1
  if contagem > 3256:
    contagem = 0
```

+ Code + Text

```
1 len(dataset) / 10
```

```
3256.2
```

+ Código + Texto

1s

```
1 grupos = []
2 id_grupo = 0
3 contagem = 0
4 for _ in dataset.iterrows():
5   grupos.append(id_grupo)
6   contagem += 1
7   if contagem > 3256:
8     contagem = 0
9     id_grupo += 1
```

+ Código + Texto

```
1 (grupos)
```

```
0,
0,
0,
0,
0,
0,
```

+ Código + Texto

```
1 print (grupos)
```

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
1 print (grupos)
```

```
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
```

+ Código + Texto

```
1 np.unique(grupos)
```

0s

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
[15] 1 np.unique(grupos, return_counts=True)
```

0s

```
(array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
 array([3257, 3257, 3257, 3257, 3257, 3257, 3257, 3257, 3257, 3249]))
```

```
1 np.shape(grupos), dataset.shape
```

```
((32562,), (32562, 15))
```

```
1 dataset.head()
```

	age	workclass	final-weight	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hour-per-week	native-country	income
0	age	workclass	final-weight	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loos	hour-per-week	native-country	income
1	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
2	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
3	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
4	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K

```
[22]  1 dataset['grupo'] = grupos
```

```
1 dataset.head()
```

	age	workclass	final-weight	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hour-per-week	native-country	income	grupo
0	age	workclass	final-weight	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loos	hour-per-week	native-country	income	0
1	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K	0
2	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K	0
3	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K	0
4	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K	0

```
1 dataset.tail()
```

	age	workclass	final-weight	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hour-per-week	native-country	income	grupo
32557	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	38	United-States	<=50K	9
32558	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	40	United-States	>50K	9
32559	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40	United-States	<=50K	9
32560	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	20	United-States	<=50K	9
32561	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024	0	40	United-States	>50K	9

+ Código

```
1 random.randint(0, 9)
```

```
1
```

```
1 random.randint(0, 9)
```

```
7
```

```
1 random.randint(0, 9)
```

```
9
```

```
1 df_agrupamento = dataset[dataset['grupo'] == 7]
2 df_agrupamento.shape
```

```
(3257, 16)
```

```
1 random.randint(0, 9)
```

0s

```
9
```

```
1 df_agrupamento['grupo'].value_counts()
```

0s

```
7    3257
Name: grupo, dtype: int64
```

```
def amostragem_agrupamento(dataset, numero_grupos):
  intervalo = len(dataset) / numero_grupos
```

```
[19]  len(dataset)//10
```

```
3256
```

```
grupos = []
id_grupo = 0
contagem = 0
for _ in dataset.iterrows():
  grupos.append(id_grupo)
  contagem += 1
  if contagem > 3256:
    contagem = 0
    id_grupo += 1
```

```
def amostragem_agrupamento(dataset, numero_grupos):
  intervalo = len(dataset) / numero_grupos

  grupos = []
  id_grupo = 0
  contagem = 0
  for _ in dataset.iterrows():
    grupos.append(id_grupo)
    contagem += 1
    if contagem > intervalo:
      contagem = 0
      id_grupo += 1
```

[47] 0s

```python
def amostragem_agrupamento(dataset, numero_grupos):
  intervalo = len(dataset) / numero_grupos

  grupos = []
  id_grupo = 0
  contagem = 0
  for _ in dataset.iterrows():
    grupos.append(id_grupo)
    contagem += 1
    if contagem > intervalo:
      contagem = 0
      id_grupo += 1

  dataset['grupo'] = grupos
  #grupo_selecionado = random.randint(0, numero_grupos)
  grupo_selecionado = random.randint(0, numero_grupos - 1) #Atualizado 16/10/2023
  return dataset[dataset['grupo'] == grupo_selecionado]
```

```
1 df_amostra_agrupamento = amostragem_agrupamento(dataset, 100)
2 df_amostra_agrupamento.shape, df_amostra_agrupamento['grupo'].value_counts()
```

```
((326, 16),
17    326
Name: grupo, dtype: int64)
```

```
1 df_amostra_agrupamento = amostragem_agrupamento(dataset, 100)
2 df_amostra_agrupamento.shape, df_amostra_agrupamento['grupo'].value_counts
```

```
((326, 16),
72    326
Name: grupo, dtype: int64)
```

```python
def amostragem_agrupamento(dataset, numero_grupos):
  intervalo = len(dataset) / numero_grupos

  grupos = []
  id_grupo = 0
  contagem = 0
  for _ in dataset.iterrows():
    grupos.append(id_grupo)
    contagem += 1
    if contagem > intervalo:
      contagem = 0
      id_grupo += 1

  dataset['grupo'] = grupos
  #grupo_selecionado = random.randint(0, numero_grupos)
  random.seed(1)
  grupo_selecionado = random.randint(0, numero_grupos - 1) #Atualizado 16/10/2023
  return dataset[dataset['grupo'] == grupo_selecionado]
```

1s

```python
df_amostra_agrupamento = amostragem_agrupamento(dataset, 100)
df_amostra_agrupamento.shape, df_amostra_agrupamento['grupo'].value_cc
```

```
((326, 16),
 17    326
 Name: grupo, dtype: int64)
```

```
1 len(dataset)/325
```

```
100.19076923076923
```

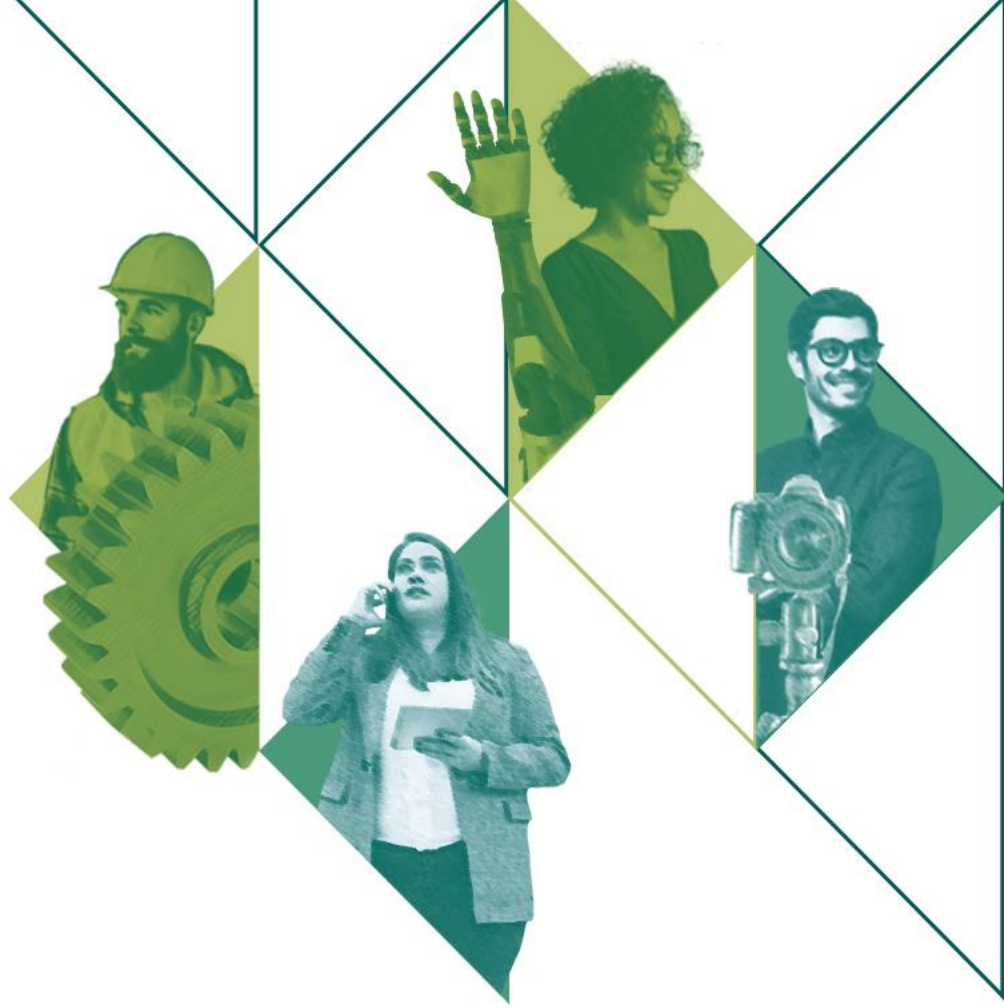
```
1 325*100
```

```
32500
```

```
1 df_amostra_agrupamento.head()
```

	age	workclass	final-weight	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hour-per-week	native-country	income
5542	33	Private	158416	HS-grad	9	Never-married	Machine-op-inspct	Not-in-family	White	Male	0	0	40	United-States	<=50K
5543	40	Self-emp-inc	169878	Assoc-acdm	12	Married-civ-spouse	Exec-managerial	Wife	White	Female	0	0	40	United-States	>50K
5544	44	Private	296728	Masters	14	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	40	United-States	>50K
5545	33	Local-gov	342458	Assoc-acdm	12	Divorced	Protective-serv	Not-in-family	White	Male	0	0	56	United-States	<=50K
5546	21	Local-gov	38771	Some-college	10	Never-married	Adm-clerical	Own-child	White	Male	0	0	40	United-States	<=50K

CENTRO UNIVERSITÁRIO
UNISATC



/UNISATC