



Aula 5- Estatística Aplicada

Técnicas de amostragem- parte 2

Amostragem aleatória simples

Profa. Me. Aline Cipriano

SEQUÊNCIA DIDÁTICA

- Censo e Amostragem
- Técnicas de amostragem probabilística: amostragem aleatória simples
- ✓ Números aleatórios (TNA, calculadora)
- ✓ Implementação no Python

CENSO x AMOSTRAGEM

Censo: coleta dados de toda a população, fornecendo informações completas e detalhadas sobre todos os seus membros.

Amostragem: coleta dados a partir de uma amostra, que é uma parte da população, para estimar características do grupo todo.



CENSO x AMOSTRAGEM

1. Precisão Necessária:

- **CENSO:** se precisar de informações completas e detalhadas sobre toda a população.
- **AMOSTRAGEM:** se precisar de uma visão geral

2. Recursos Disponíveis:

- **CENSO:** recursos abundantes.
- **AMOSTRAGEM:** recursos limitados (tempo, dinheiro, pessoal). A amostragem é mais econômica e rápida em comparação com um censo.

3. Tamanho da População:

- **CENSO:** populações pequenas onde é viável coletar dados de todos.
- **AMOSTRAGEM:** populações grandes, a amostragem é mais prática e ainda fornece resultados confiáveis.

EXERCÍCIOS

1. Analise as situações descritas abaixo e decida se a pesquisa deve ser feita por amostragem ou por censo, justificando sua resposta.
 - a) Numa linha de produção de empacotamento de café, observar o peso dos pacotes produzidos.
 - a) CENSO. Se houver possibilidade de agregar uma balança automática ao processo produtivo pode-se utilizar censo. Pois como não se trata de teste destrutivo, e peso dos pacotes é importante para a imagem da empresa (e para não haver desperdício) todos os pacotes podem ser medidos.

EXERCÍCIOS

b) Em uma sala de aula composta por 40 alunos, analisar suas idades.

CENSO, porque a população é pequena, apenas 40 elementos.

c) Observar se a água de uma lagoa está contaminada.

AMOSTRAGEM. É no mínimo contraditório retirar toda a água da lagoa para exame de sua contaminação.

EXERCÍCIOS

d) Verificar a carga horária diária de trabalho dos funcionários da cozinha de um restaurante.

CENSO, por razões políticas, para que ninguém se sinta prejudicado, além da população ser pequena e acessível.

e) Pesquisa de opinião eleitoral para governador do estado de São Paulo.

AMOSTRAGEM. A necessidade de rápido processamento exige amostragem, devido ao grande tamanho da população que tornaria muito lenta a obtenção dos resultados, além de muito cara.

TÉCNICAS DE AMOSTRAGEM PROBABILÍSTICA

- Amostragem Aleatória Simples (TNA, calculadora);
- Amostragem Sistemática;
- Amostragem por Conglomerados.
- Amostragem Estratificada.

AMOSTRAGEM ALEATÓRIA SIMPLES

A amostragem aleatória simples é uma técnica de seleção de amostra em que cada indivíduo de uma população tem a mesma probabilidade de ser selecionado.



AMOSTRAGEM ALEATÓRIA SIMPLES

Exemplo1

- Pesquisa sobre prática de atividade física de famílias de funcionários de uma certa empresa.
- **População: 32 funcionários**
- **Amostra: 5 funcionários**

01. Aristóteles	02. Anastácia	03. Arnaldo	04. Bartolomeu	05. Bernardino
06. Cardoso	07. Carlito	08. Cláudio	09. Ermílio	10. Hercílio
11. Ernestino	12. Endevaldo	13. Francisco	14. Felício	15. Fabrício
16. Geraldo	17. Gabriel	18. Getúlio	19. Hiraldo	20. João da Silva
21. Joana	22. Joaquim	23. Joaquina	24. José da Silva	25. José de Souza
26. Josefa	27. Josefina	28. Maria José	29. M ^a Cristina	30. Mauro
31. Paula	32. Paulo César			

AMOSTRAGEM ALEATÓRIA SIMPLES

Exemplo 2

- Pesquisa de estatura de crianças do 6º ano do ensino fundamental de uma escola
- População: 1000 alunos
- Amostra: 10% da população

AMOSTRAGEM ALEATÓRIA SIMPLES

Exemplo 2

- Pesquisa de estatura de crianças do 6º ano do ensino fundamental de uma escola
- População: 1000 alunos
- Amostra: 10% da população

- 1º passo - numeramos os alunos de 1 a 1000.
- 2º passo - escrevemos os números dos alunos, de 1 a 1000, em pedacinhos de papel
- 3º passo - retiramos 100 números.
- **Procedimento MUITO trabalhoso!!!!**
- **Alternativa:** tabela de números aleatórios no Excel ALEATORIO ENTRE (0 e 1000)

NÚMEROS ALEATÓRIOS- TNA

141592653589793238462643383279502
419716939937510582097494459230781
062862089986280348253421170679821
086513282306647093844609550582231
535940812848111745028410270193852
055596446229489549303819644288109
665933446128475648233786783165271
190914564856692346034861045432664
133936072602491412737245870066063
588174881520920962829254091715364
789259036001133053054882046652138
469519415116094330572703657595919
092186117381932611793105118548074
237996274956735188575272489122793

Situação-problema 1:

População: 80

Amostra: 5

Leitura: primeira linha- esquerda para a direita

R: 14, 15, 65, 35, 79

Situação-problema 2:

População: 400

Amostra: 5

Leitura: última linha- direita para a esquerda.

R: 122, 272, 188, 274, 237

NÚMEROS ALEATÓRIOS- TNA

98543	59525	21114	73109	69095
87060	95250	50277	17486	07962
82170	68014	07937	98003	40146
48673	26100	23776	66959	84477
08560	52600	66188	63746	05849
68708	28373	27635	52562	18148
80511	00208	61965	66983	70232
02253	27120	53172	99800	74603
37110	07752	38216	54843	22496
01548	06209	79410	99823	17603
81417	85771	25961	84381	88582
36602	77275	35226	53601	91939
79337	00250	64655	89710	19526
60564	55609	64304	10940	69422
87552	78655	14220	30037	07403
04951	65135	00626	99163	34098
01761	01488	35218	11762	11586
41451	57175	88050	23528	46360
03646	98017	51286	18545	02393
02863	33742	19979	10905	34863
.....

População: 20

Amostra: 2

11, 09

NÚMEROS ALEATÓRIOS- TNA

57720039848441796771402113975649865408932968746483
28805351590993988758702771771706320278621674696517
92591852873048869748352518887403629838586586424103
90381291743019758907506415597188137495305278301175
80911694675860820666904756184645111235324550411343
22017031329691927540165429727495009597610098243007
56241004302046299053531105844121647919762951626066
79449262029686643000945669302059878735442250977819
53996645088978507753372577412762380223576201416035
18928735885505213651392850146685793019797266643145
53085896630561257022504128966266436306630132798522
03588029287689511824888946474855192987031033996712
27078188656949980028047051300147189733218582454324
05210859010622249891811755446616077307661012317858
40361327843082333639694205586461123389278952667193
54602528858820001059610536613372010119016110512091
71516340767111737352373160458892734371280498090248
61020181739260667358533442682638340327449604466593
82559313463095265506961765917235799612495280632699
89985414217413576819862860894733152628774538480808
00998484146795137758901450794273633106604340125504
62415078204805884352980319939203049725849595036331
94279069246809921186076383193298511555710927026700
44892928843628251582877418972576106326760226745328
97307695332110542695666552049936584803089363581796
39165804448015595983909554668184396085388866333569
60781103266750340961313020769366308351093383647605
0319234762895779133884760593754394877674985384391
4128526756253959665513690322239330522990339979699
77549850392537425297100356049281668670014889558210
28634161916424838137344883279638716973067750256460
74244885401233596750149814264279791352896978804471
00240337964668750532421663332897263647277365383446
05414769694536167118955197220413239658600369487983
62698497974723665156130869115275592686818043009892

População: 600

Amostra: 3

577, 200, 398

População: 1000

Amostra: 1

0039,

NÚMEROS ALEATÓRIOS- TNA

Considere uma classe de 40 alunos que irão participar de um bingo.
Selecione 5 pessoas usando a tabela de números aleatórios.

•	60	80	85	44	44	74	41	28	11	05
•	80	94	04	48	93	10	40	83	62	22
•	85	27	48	68	93	11	30	32	92	70
•	84	13	38	96	40	44	03	55	21	66
•	64	42	52	81	08	16	55	41	60	16
•	90	04	58	54	97	51	98	15	06	54
•	19	51	69	01	20	46	75	97	16	43
•	49	38	65	44	80	23	60	42	35	54
•	06	31	28	89	40	15	99	56	93	21
•	60	94	20	03	07	11	89	79	26	74

População: 40

Amostra: 5

13, 38, 40, 03, 21.

População: 100

Amostra: 1

085

NÚMEROS ALEATÓRIOS- CALCULADORA



AMOSTRAGEM ALEATÓRIA SIMPLES- PYTHON

Amostragem.ipynb ☆

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda As alterações não serão salvas

- Localizar no Drive
- Novo notebook
- Abrir notebook Ctrl+O
- Fazer upload de notebook
- Rename
- Mover
- Mover para a lixeira
- Salvar uma cópia no Drive**
- Salvar uma cópia como Gist do GitHub
- Salvar uma cópia no GitHub
- Salvar Ctrl+S
- Salvar e fixar revisão Ctrl+M S
- Histórico de revisões
- Fazer download ▶
- Imprimir Ctrl+P

```
ple(n = 100, random_state = 1)
```

<https://colab.google/>





Amostragem.ipynb ☆

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda As alterações não serão salvas

Arquivos



{x} ..
sample_data



+ Código

+ Texto

Copiar para o Drive

✓ Carregamento da base de dados

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

Arquivos



{x}



..



sample_data



census.csv



Disco



Disponível: 81.45 GB

✓ Carregamento da base de dados

✓
0s

```
1 import pandas as pd  
2 import random  
3 import numpy as np
```

+ Código

+ Texto

```
1 dataset=pd.read_csv('census.csv')
```


✓ Amostragem

✓ Carregamento da base de dados

```
[4] 1 import pandas as pd  
    2 import random  
    3 import numpy as np
```

```
[15] 1 dataset=pd.read_csv('census.csv')
```

```
[16] 1 dataset.shape
```



Cópia de Amostragem.ipynb ☆

Arquivo Editar Ver Inserir Ambiente de execu

+ Código + Texto

✓ Carregamento da base de da

```
[ ] 1 import pandas as pd  
    2 import random  
    3 import numpy as np
```



```
1 dataset = pd.read_csv('census.csv')  
2  
3 print(dataset.shape)  
4
```

```
(32562, 15)
```

Amostragem

Carregamento da base de dados

```
[4] 1 import pandas as pd  
    2 import random  
    3 import numpy as np
```

```
[15] 1 dataset=pd.read_csv('census.csv')
```

```
[16] 1 dataset.shape
```



Cópia de Amostragem.ipynb ☆

Arquivo Editar Ver Inserir Ambiente de execuç

+ Código + Texto

Carregamento da base de da

```
[ ] 1 import pandas as pd  
    2 import random  
    3 import numpy as np
```

```
1 dataset = pd.read_csv('census.csv')  
2  
3 print(dataset.shape)  
4
```

```
(32562, 15)
```

Caso não apareça as 15 colunas, utilize um dos caminhos abaixo:



```
dataset=pd.read_csv('census.csv', delimiter=';')
```

```
dataset = pd.read_csv('census.csv', sep=';', header=None, names=['age',  
'workclass', 'final-weight', 'education', 'education-num', 'marital-  
status', 'occupation', 'relationship', 'race', 'sex', 'capital-gain', 'capital-  
loss', 'hour-per-week', 'native-country', 'income'])
```



+ Código + Texto

RAM
Disco

Colab AI



```
[20] 3 print(dataset.shape)
      4
```

(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



+ Código + Texto

RAM
Disco

Colab AI



[19] 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
32557	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	38	United-States	<=50K
32558	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	40	United-States	>50K
32559	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40	United-States	<=50K
32560	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	20	United-States	<=50K
32561	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024	0	40	United-States	>50K

✓ Amostragem aleatória simples

```
1 df_amostra_aleatoria_simples = dataset.sample(n = 100)
```

```
[ ] 1 df_amostra_aleatoria_simples.shape
```

```
(100, 15)
```

```
[ ] 1 df_amostra_aleatoria_simples.head()
```

	age	workclass	final-weight	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loos
9646	62	Self-emp-not-inc	26911	7th-8th	4	Widowed	Other-service	Not-in-family	White	Female	0	0
709	18	Private	208103	11th	7	Never-married	Other-service	Other-relative	White	Male	0	0
7385	25	Private	102476	Bachelors	13	Never-married	Farming-fishing	Own-child	White	Male	27828	0
16671	33	Private	511517	HS-grad	9	Married-civ-	Prof-specialty	Husband	White	Male	0	0

```
1 df_amostra_aleatoria_simples = dataset.sample(n=100, )
```

```
1 df_amostra_aleatoria_simples.shape
```

```
100, 15)
```

```
1 df_amostra_aleatoria_simples.head()
```

age	workclass	final-weight	education	education nu
-----	-----------	--------------	-----------	--------------

(n: int | None = ..., frac: float | None = ..., replace: _bool = ..., weights: _str | ListLike@sample | None = ..., random_state: RandomState | None = ..., axis: AxisIndex | None = ..., ignore_index: _bool = ...) -> DataFrame

Return a random sample of items from an axis of object.

You can use `random_state` for reproducibility.

Parameters

```
1 df_amostra_aleatoria_simples = dataset.sample(n=100, )
```

```
1 df_amostra_aleatoria_simples.sh
```

```
100, 15)
```

```
1 df_amostra_aleatoria_simples.he
```

age	workclass	final-weight
-----	-----------	--------------

3044	21	Private	194723
------	----	---------	--------

Parameters

n : int, optional

Number of items from axis to return. Cannot be used with `frac`.

Default = 1 if `frac` = None.

frac : float, optional

Fraction of axis items to return. Cannot be used with `n`.

replace : bool, default False

Allow or disallow sampling of the same row more than once.

weights : str or ndarray-like, optional

Default 'None' results in equal probability weighting.

If passed a Series, will align with target object on index. Index

values in weights not found in sampled object will be ignored and

```
1 df_amostra_aleatoria_simples = dataset.sample(n=100, random_state = 1)
```

```
1 df_amostra_aleatoria_simples.shape
```

```
(100, 15)
```

```
1 df_amostra_aleatoria_simples.head()
```

	age	workclass	final-weight	education	education-num	marital-status
9646	43	Private	262038	5th-6th	3	Married-spouse-absent

weights of zero.

If called on a DataFrame, will accept the name of a column when axis = 0.

Unless weights are a Series, weights must be same length as axis being sampled.

If weights do not sum to 1, they will be normalized to sum to 1.

Missing values in the weights column will be treated as zero.

Infinite values not allowed.

random_state : int, array-like, BitGenerator, np.random.RandomState, np.random.Generator, or None

If int, array-like, or BitGenerator, seed for random number generator

If np.random.RandomState or np.random.Generator, use as given.

axis : {0 or 'index', 1 or 'columns', None}, default None

Axis to sample. Accepts axis number or name. Default is stat axis.



```
1 df_amostra_aleatoria_simples.head()
```



	age	workclass	final-weight	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hour-per-week
9646	43	Private	262038	5th-6th	3	Married-spouse-absent	Farming-fishing	Unmarried	White	Male	0	0	35
709	60	Private	177665	HS-grad	9	Married-civ-spouse	Craft-repair	Husband	White	Male	0	0	35
7385	37	Private	172538	HS-grad	9	Never-married	Machine-op-inspct	Own-child	White	Male	0	0	40
32273	38	Private	447346	Some-college	10	Married-civ-spouse	Craft-repair	Husband	White	Male	0	0	36
17122	54	Private	284129	Some-college	10	Married-civ-spouse	Transport-moving	Husband	White	Male	0	0	45

[+ Código](#)[+ Texto](#)

+ Código + Texto

```
✓ [58] 1 def amostragem_aleatoria_simples(dataset, amostras):  
0s 2     return dataset.sample(n = amostras)
```

```
✓ [59] 1 df_amostra_aleatoria_simples = amostragem_aleatoria_simples(dataset, 100)  
0s 2 df_amostra_aleatoria_simples.shape
```

(100, 15)

```
✓ 1 df_amostra_aleatoria_simples.head()  
0s
```

	age	workclass	final-weight	education	education-num	marital-status	occupation	relationship	race	sex	ca
15436	47	Private	162034	Bachelors	13	Married-civ-spouse	Sales	Husband	White	Male	
5997	20	Private	293091	11th	7	Never-married	Transport-moving	Own-child	White	Male	
18075	34	Federal-gov	436341	Some-college	10	Married-AF-spouse	Adm-clerical	Wife	White	Female	
		Self-emp-				Married-civ-					


```
[77] 1 def amostragem_aleatoria_simples(dataset, amostras):  
2     return dataset.sample(n = amostras, random_state=1)
```

```
1 df_amostra_aleatoria_simples = amostragem_aleatoria_simples(dataset, 100)  
2 df_amostra_aleatoria_simples.shape
```

(100, 15)

```
1 df_amostra_aleatoria_simples.head()
```

	age	workclass	final-weight	education	education-num	marital-status	occupation	re:
9646	43	Private	262038	5th-6th	3	Married-spouse-absent	Farming-fishing	
709	60	Private	177665	HS-grad	9	Married-civ-spouse	Craft-repair	
7385	37	Private	172538	HS-grad	9	Never-married	Machine-op-inspct	

+ Código + Texto

✓ 0s [80] 1 def amostragem_aleatoria_simples(dataset, amostras):
2 return dataset.sample(n = amostras, random_state=2)

✓ 0s 1 df_amostra_aleatoria_simples = amostragem_aleatoria_simples(dataset, 100)
2 df_amostra_aleatoria_simples.shape

(100, 15)

✓ 0s 1 df_amostra_aleatoria_simples.head()

	age	workclass	final-weight	education	education-num	marital-status	occupation	relationship	race	sex	ca
16054	59	Private	159724	Masters	14	Married-civ-spouse	Sales	Husband	White	Male	
				Some-		Married-civ-	Protective-				



CENTRO UNIVERSITÁRIO
UNISATC



  /UNISATC