

Probabilidade e Estatística - Atividade 3



PARTE A

Para realizar a amostra simples com 15% da população foi utilizado a função "sample" do Python, onde passei o parâmetro 0.15 para pegar 15% e deixei como randômico.

```
In [418]: # importar bibliotecas necessárias
import pandas as pd
import matplotlib.pyplot as plt
import statistics as std
import numpy as np
import random
import math
```

```
In [419]: df = pd.read_csv("Populacao360.csv", sep = ';')
df = df.drop(['Unnamed: 8'], axis = 1)
df.IMC = df.IMC.apply(lambda x: x.replace(',', '.'))
df.IMC = df.IMC.astype(float)
```

```
In [420]: # ver os valores iniciais
df.head()
```

Out[420]:

	nº	sexo	idade	peso	altura	IES	IMC	Clas IMC
0	1	F	19	52	167	UFAC	18.6	ad
1	2	F	20	78	177	UFAC	24.9	ad
2	3	F	22	56	172	UFAC	18.9	ad
3	4	F	19	45	165	UFAC	16.5	mg
4	5	F	18	60	160	UFAC	23.4	ad

```
In [421]: # amostra aleatória simples com 15% da população
df_amostra = df.sample(frac=0.15, random_state=20)
df_amostra.head()
df_amostra.to_excel("amostra_simples.xlsx")
```

```
In [422]: # 5 primeiro itens da amostra
df_amostra.head()
```

Out[422]:

	nº	sexo	idade	peso	altura	IES	IMC	Clas IMC
219	220	M	20	70	180	UFAC	21.6	ad
14	15	F	18	64	171	UFAC	21.9	ad
232	233	M	19	50	152	UFAC	21.6	ad
293	294	M	18	103	167	UFSCAR	36.9	ob
238	239	M	22	59	170	UFAC	20.4	ad

1 - Para as variáveis SEXO e IES, obter o número absoluto e a frequência relativa percentual.

```
In [423]: # número absoluto
df_amostra.sexo.value_counts()
```

Out[423]: M 34
F 20
Name: sexo, dtype: int64

```
In [424]: # número absoluto
df_amostra.IES.value_counts()
```

Out[424]: UFSCAR 30
UFAC 24
Name: IES, dtype: int64

```
In [425]: # frequência relativa em porcentagem
df_amostra.sexo.value_counts()/df_amostra.shape[0] * 100
```

Out[425]: M 62.962963
F 37.037037
Name: sexo, dtype: float64

```
In [426]: # frequência relativa em porcentagem
df_amostra.IES.value_counts()/df_amostra.shape[0] * 100
```

```
Out[426]: UFSCAR    55.555556
UFAC          44.444444
Name: IES, dtype: float64
```

Para os dados populacionais das variáveis IDADE, PESO, ALTURA e IMC as seguintes medidas e gráficos:

- i) Média aritmética.
- ii) Mediana.
- iii) Variância amostral.
- iv) Desvio Padrão amostral.
- v) Coeficiente de Variação.
- vi) Box-plot de cada uma das variáveis separadamente.

Média

```
In [427]: # média aritmética
print("Idade: ", round(df_amostra.idade.mean(),2))
print("Peso: ", round(df_amostra.peso.mean(),2))
print("Altura: ", round(df_amostra.altura.mean(),2))
print("IMC: ", round(df_amostra.IMC.mean(),2))
```

```
Idade:  20.93
Peso:   68.3
Altura: 169.46
IMC:    23.73
```

Erro relativo da média da amostra em relação à população

```
In [428]: # ErroRelativo = 100 x |ValorPopulacional - ValorAmostral|/ValorPopulacional
## Media

mediaA = []
mediaA.append(100 * abs(df.idade.mean() - df_amostra.idade.mean())/df.idade.m
mediaA.append(100 * abs(df.peso.mean() - df_amostra.peso.mean())/df.peso.mean
mediaA.append(100 * abs(df.altura.mean() - df_amostra.altura.mean())/df.altur
mediaA.append(100 * abs(df.IMC.mean() - df_amostra.IMC.mean())/df.IMC.mean())

print("Erro relativo Idade:")
print(mediaA[0], "\n")
print("Erro relativo Peso:")
print(mediaA[1], "\n")
print("Erro relativo Altura:")
print(mediaA[2], "\n")
print("Erro relativo IMC:")
print(mediaA[3])
```

Erro relativo Idade:
3.319644079397672

Erro relativo Peso:
3.374817804686621

Erro relativo Altura:
0.7808284049735095

Erro relativo IMC:
2.1007382264239878

Mediana

```
In [429]: # mediana
print("Idade: ", round(df_amostra.idade.median(),2))
print("Peso: ", round(df_amostra.peso.median(),2))
print("Altura: ", round(df_amostra.altura.median(),2))
print("IMC: ", round(df_amostra.IMC.median(),2))
```

```
Idade:  20.0
Peso:   65.5
Altura: 170.0
IMC:    23.4
```

Erro relativo da mediana da amostra em relação à população

```
In [430]: # ErroRelativo = 100 x |ValorPopulacional - ValorAmostral|/ValorPopulacional
#Mediana

medianaA = []
medianaA.append(100 * abs(df.idade.median() - df_amostra.idade.median())/df.i
medianaA.append(100 * abs(df.peso.median() - df_amostra.peso.median())/df.pes
medianaA.append(100 * abs(df.altura.median() - df_amostra.altura.median())/df
medianaA.append(100 * abs(df.IMC.median() - df_amostra.IMC.median())/df.IMC.m

print("Erro relativo Idade:")
print(medianaA[0], "\n")
print("Erro relativo Peso:")
print(medianaA[1], "\n")
print("Erro relativo Altura:")
print(medianaA[2], "\n")
print("Erro relativo IMC:")
print(medianaA[3])
```

```
Erro relativo Idade:
4.761904761904762
```

```
Erro relativo Peso:
2.34375
```

```
Erro relativo Altura:
1.1904761904761905
```

```
Erro relativo IMC:
2.4070021881837946
```

Variância

Variância de uma amostra (ou coleção) de dados de tipo quantitativo é a medida que se obtém somando os quadrados dos desvios dos dados relativamente à média, e dividindo pelo número de dados menos um.

Por exemplo, temos que a média da variável idade é:

```
In [431]: media_idade = df_amostra.idade.mean()
          print("Média da idade", media_idade)
```

Média da idade 20.925925925925927

Assim, a variância amostral será:

```
In [432]: acumulador = 0
          for idade in df_amostra.idade:
              acumulador = (idade - media_idade)**2 + acumulador
          vams = acumulador/(df_amostra.shape[0] - 1)

          print("A variância amostral da idade é: ", vams)
```

A variância amostral da idade é: 16.334032145352904

De modo semelhante, teremos:

```
In [433]: # variância amostral
          print("Idade: ", round(std.variance(df_amostra.idade),2))
          print("Peso: ", round(std.variance(df_amostra.peso),2))
          print("Altura: ", round(std.variance(df_amostra.altura),2))
          print("IMC: ", round(std.variance(df_amostra.IMC),2))
```

Idade: 16.33
Peso: 241.38
Altura: 104.18
IMC: 21.49

Erro relativo da variância da amostra em relação à população

```
In [495]: # ErroRelativo = 100 x |ValorPopulacional - ValorAmostral|/ValorPopulacional
# Variância

varianciaA = []
varianciaA.append(100 * abs(std.pvariance(df.idade) -std.variance(df_amostra.
varianciaA.append(100 * abs(std.pvariance(df.peso) -std.variance(df_amostra.p
varianciaA.append(100 * abs(std.pvariance(df.altura) -std.variance(df_amostra
varianciaA.append(100 * abs(std.pvariance(df.IMC) -std.variance(df_amostra.IM

print("Erro relativo Idade:")
print(varianciaA[0], "\n")
print("Erro relativo Peso:")
print(varianciaA[1], "\n")
print("Erro relativo Altura:")
print(varianciaA[2], "\n")
print("Erro relativo IMC:")
print(varianciaA[3], "\n")
```

Erro relativo Idade:
8.562053628310666

Erro relativo Peso:
6.9003819276712

Erro relativo Altura:
13.869744434591745

Erro relativo IMC:
8.475199460219937

Desvio Padrão

O desvio padrão amostral é a raiz quadrada da variância amostral, no caso exemplo anterior, seria:

```
In [435]: dams = math.sqrt(vams)
print("O desvio padrão amostral de idade é: ", dams)
```

O desvio padrão amostral de idade é: 4.0415383389685795

De modo semelhante, temos:

```
In [436]: # desvio padrão amostral
print("Idade: ", round(std.stdev(df_amostra.idade),2))
print("Peso: ", round(std.stdev(df_amostra.peso),2))
print("Altura: ", round(std.stdev(df_amostra.altura),2))
print("IMC: ", round(std.stdev(df_amostra.IMC),2))
```

```
Idade:  4.04
Peso:  15.54
Altura: 10.21
IMC:   4.64
```

Coeficiente de variação

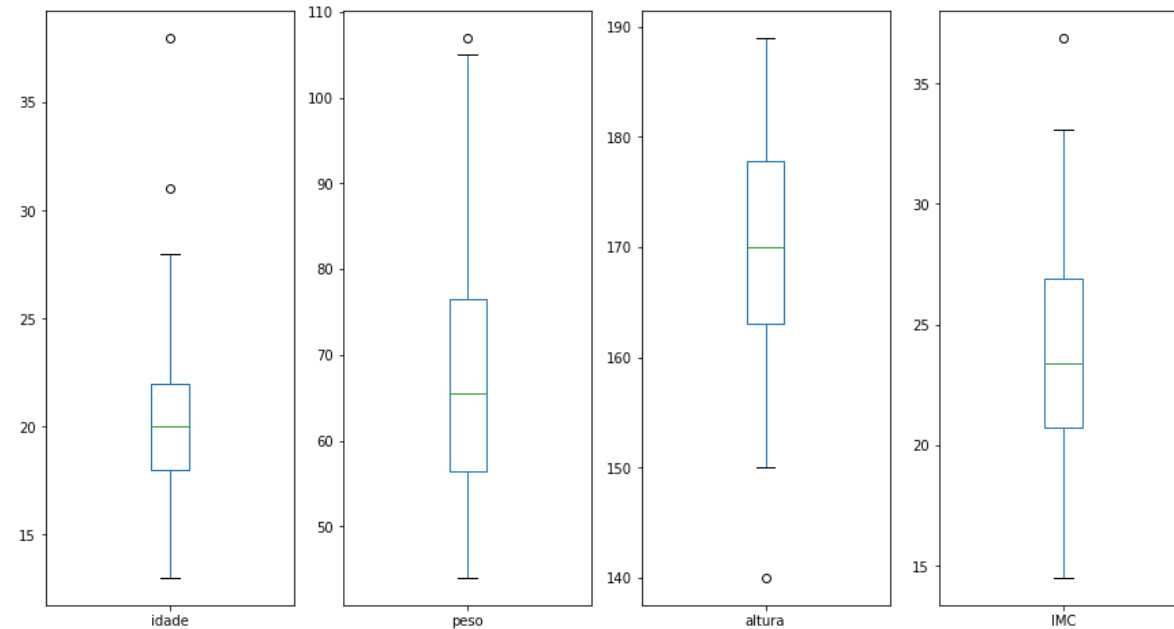
O coeficiente de variação é dado pelo desvio padrão amostral dividido pela média

```
In [437]: # coeficiente de variação
print("Idade: ", round(std.stdev(df_amostra.idade)/df_amostra.idade.mean() *
print("Peso: ", round(std.stdev(df_amostra.peso)/df_amostra.peso.mean() * 100
print("Altura: ", round(std.stdev(df_amostra.altura)/df_amostra.altura.mean()
print("IMC: ", round(std.stdev(df_amostra.IMC)/df_amostra.IMC.mean() * 100,2)
```

```
Idade:  19.31
Peso:   22.75
Altura:  6.02
IMC:   19.54
```

Boxplot

```
In [438]: # boxplot para as variáveis quantitativas
fig, (ax1, ax2, ax3, ax4) = plt.subplots(1, 4, figsize = (15, 8))
df_amostra.idade.plot(kind = 'box', ax = ax1);
df_amostra.peso.plot(kind = 'box', ax = ax2)
df_amostra.altura.plot(kind = 'box', ax = ax3)
df_amostra.IMC.plot(kind = 'box', ax = ax4)
plt.show()
```



PARTE B

Nesse caso, como era uma amostra estratificada por sexo, utilizei uma função que irá manter uma porcentagem semelhante à população.

```
In [439]: # amostra estratificada por sexo
df_sexo = df.groupby('sexo', group_keys=False).apply(lambda x: x.sample(frac=
df_sexo.head()
df_sexo.to_excel("amostra_sexo.xlsx")
```

1 - Para as variáveis SEXO e IES, obter o número absoluto e a frequência relativa percentual.

```
In [440]: # número absoluto
df_sexo.sexo.value_counts()
```

```
Out[440]: M    28
         F    26
         Name: sexo, dtype: int64
```

```
In [441]: # número absoluto
df_sexo.IES.value_counts()
```

```
Out[441]: UFSCAR    31
         UFAC       23
         Name: IES, dtype: int64
```

```
In [442]: # frequência relativa em porcentagem
df_sexo.sexo.value_counts()/df_sexo.shape[0] * 100
```

```
Out[442]: M    51.851852
         F    48.148148
         Name: sexo, dtype: float64
```

```
In [443]: # frequência relativa em porcentagem
df_sexo.IES.value_counts()/df_sexo.shape[0] * 100
```

```
Out[443]: UFSCAR    57.407407
         UFAC       42.592593
         Name: IES, dtype: float64
```

Para os dados populacionais das variáveis IDADE, PESO, ALTURA e IMC as seguintes medidas e gráficos:

- i) Média aritmética.

- ii) Mediana.
- iii) Variância amostral.
- iv) Desvio Padrão amostral.
- v) Coeficiente de Variação.
- vi) Box-plot de cada uma das variáveis separadamente.

Média

```
In [444]: # média aritmética
print("Idade: ", round(df_sexo.idade.mean(),2))
print("Peso: ", round(df_sexo.peso.mean(),2))
print("Altura: ", round(df_sexo.altura.mean(),2))
print("IMC: ", round(df_sexo.IMC.mean(),2))
```

```
Idade:  21.37
Peso:   67.26
Altura: 167.81
IMC:    23.75
```

Erro relativo da média da amostra em relação à população

```
In [445]: # ErroRelativo = 100 x |ValorPopulacional - ValorAmostral|/ValorPopulacional
## Media

mediaB = []
mediaB.append(100 * abs(df.idade.mean() - df_sexo.idade.mean())/df.idade.mean())
mediaB.append(100 * abs(df.peso.mean() - df_sexo.peso.mean())/df.peso.mean())
mediaB.append(100 * abs(df.altura.mean() - df_sexo.altura.mean())/df.altura.mean())
mediaB.append(100 * abs(df.IMC.mean() - df_sexo.IMC.mean())/df.IMC.mean())

print("Erro relativo Idade:")
print(mediaB[0], "\n")
print("Erro relativo Peso:")
print(mediaB[1], "\n")
print("Erro relativo Altura:")
print(mediaB[2], "\n")
print("Erro relativo IMC:")
print(mediaB[3])
```

Erro relativo Idade:
1.2662559890486031

Erro relativo Peso:
1.8051351048323758

Erro relativo Altura:
0.19933701170692603

Erro relativo IMC:
2.1804172791993497

Mediana

```
In [446]: # mediana
print("Idade: ", round(df_sexo.idade.median(),2))
print("Peso: ", round(df_sexo.peso.median(),2))
print("Altura: ", round(df_sexo.altura.median(),2))
print("IMC: ", round(df_sexo.IMC.median(),2))
```

Idade: 21.0
Peso: 63.0
Altura: 167.5
IMC: 23.55

Erro relativo da mediana da amostra em relação à população

```
In [447]: # ErroRelativo = 100 x |ValorPopulacional - ValorAmostral|/ValorPopulacional
#Mediana

medianaB = []
medianaB.append(100 * abs(df.idade.median() - df_sexo.idade.median())/df.idade.median())
medianaB.append(100 * abs(df.peso.median() - df_sexo.peso.median())/df.peso.median())
medianaB.append(100 * abs(df.altura.median() - df_sexo.altura.median())/df.altura.median())
medianaB.append(100 * abs(df.IMC.median() - df_sexo.IMC.median())/df.IMC.median())

print("Erro relativo Idade:")
print(medianaB[0], "\n")
print("Erro relativo Peso:")
print(medianaB[1], "\n")
print("Erro relativo Altura:")
print(medianaB[2], "\n")
print("Erro relativo IMC:")
print(medianaB[3])
```

Erro relativo Idade:
0.0

Erro relativo Peso:
1.5625

Erro relativo Altura:
0.2976190476190476

Erro relativo IMC:
3.0634573304157517

Variância

```
In [448]: # variância amostral
print("Idade: ", round(std.variance(df_sexo.idade),2))
print("Peso: ", round(std.variance(df_sexo.peso),2))
print("Altura: ", round(std.variance(df_sexo.altura),2))
print("IMC: ", round(std.variance(df_sexo.IMC),2))
```

```
Idade:  12.5
Peso:   235.03
Altura:  88.0
IMC:    21.28
```

Erro relativo da variância da amostra em relação à população

```
In [494]: # ErroRelativo = 100 x |ValorPopulacional - ValorAmostral|/ValorPopulacional
# Variância

varianciaB = []
varianciaB.append(100 * abs(std.pvariance(df.idade) -std.variance(df_sexo.idade))
varianciaB.append(100 * abs(std.pvariance(df.peso) -std.variance(df_sexo.peso))
varianciaB.append(100 * abs(std.pvariance(df.altura) -std.variance(df_sexo.altura))
varianciaB.append(100 * abs(std.pvariance(df.IMC) -std.variance(df_sexo.IMC))

print("Erro relativo Idade:")
print(varianciaB[0], "\n")
print("Erro relativo Peso:")
print(varianciaB[1], "\n")
print("Erro relativo Altura:")
print(varianciaB[2], "\n")
print("Erro relativo IMC:")
print(varianciaB[3], "\n")
```

Erro relativo Idade:
16.908738794794314

Erro relativo Peso:
4.085340806438105

Erro relativo Altura:
3.810109062426782

Erro relativo IMC:
7.408672330872284

Desvio Padrão


```
In [450]: # desvio padrão amostral
print("Idade: ", round(std.stdev(df_sexo.idade),2))
print("Peso: ", round(std.stdev(df_sexo.peso),2))
print("Altura: ", round(std.stdev(df_sexo.altura),2))
print("IMC: ", round(std.stdev(df_sexo.IMC),2))
```

```
Idade:  3.54
Peso:  15.33
Altura: 9.38
IMC:   4.61
```

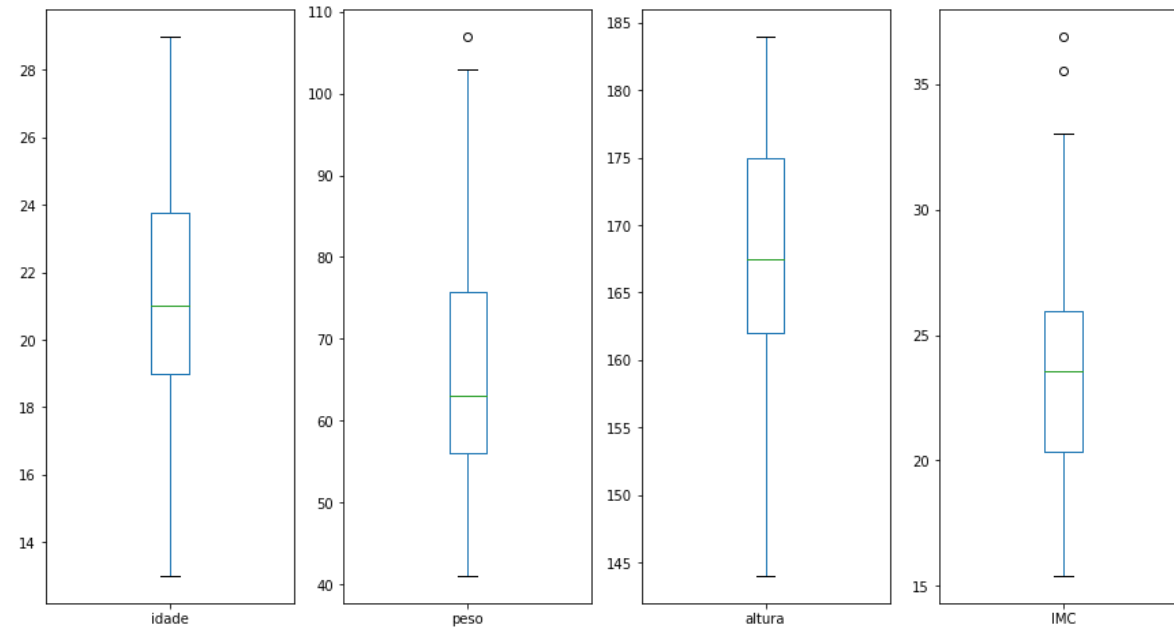
Coeficiente de variação

```
In [451]: # coeficiente de variação
print("Idade: ", round(std.stdev(df_sexo.idade)/df_sexo.idade.mean() * 100,2))
print("Peso: ", round(std.stdev(df_sexo.peso)/df_sexo.peso.mean() * 100,2))
print("Altura: ", round(std.stdev(df_sexo.altura)/df_sexo.altura.mean() * 100,2))
print("IMC: ", round(std.stdev(df_sexo.IMC)/df_sexo.IMC.mean() * 100,2))
```

```
Idade:  16.55
Peso:  22.79
Altura: 5.59
IMC:  19.42
```

Boxplot

```
In [452]: # boxplot para as variáveis quantitativas
fig, (ax1, ax2, ax3, ax4) = plt.subplots(1, 4, figsize = (15, 8))
df_sexo.idade.plot(kind = 'box', ax = ax1);
df_sexo.peso.plot(kind = 'box', ax = ax2);
df_sexo.altura.plot(kind = 'box', ax = ax3);
df_sexo.IMC.plot(kind = 'box', ax = ax4)
plt.show()
```



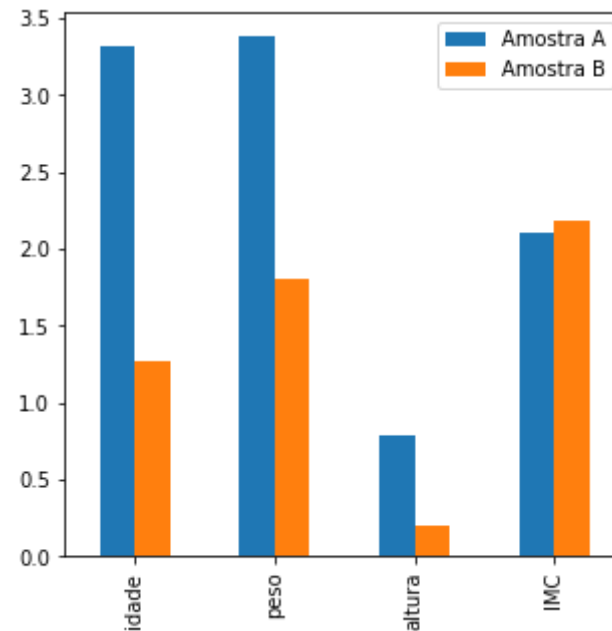
In [496]: mediaA

Out[496]: [3.319644079397672, 3.374817804686621, 0.7808284049735095, 2.10073822642398]

In [497]: mediaB

Out[497]: [1.2662559890486031,
1.8051351048323758,
0.19933701170692603,
2.1804172791993497]

```
In [498]: # Média
df_erro = pd.DataFrame(list(zip(mediaA, mediaB)),
                        columns=["Amostra A", "Amostra B"])
pd.concat([df_erro["Amostra A"],
            df_erro["Amostra B"]],
          axis=1).plot.bar(figsize=(5, 5));
plt.xticks((0, 1, 2, 3), ('idade', 'peso', 'altura', 'IMC'));
```



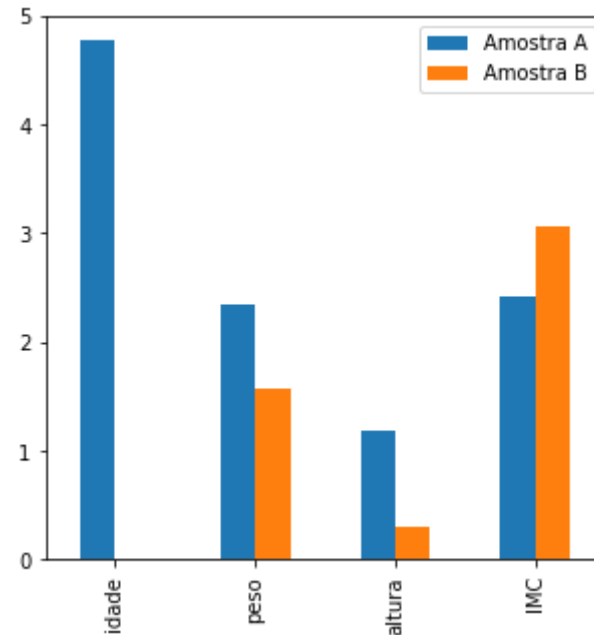
In [499]: medianaA

Out[499]: [4.761904761904762, 2.34375, 1.1904761904761905, 2.4070021881837946]

```
In [500]: medianaB
```

```
Out[500]: [0.0, 1.5625, 0.2976190476190476, 3.0634573304157517]
```

```
In [501]: # Mediana
df_erro = pd.DataFrame(list(zip(medianaA, medianaB)),
                        columns=["Amostra A", "Amostra B"])
pd.concat([df_erro["Amostra A"],
           df_erro["Amostra B"]],
          axis=1).plot.bar(figsize=(5, 5));
plt.xticks((0, 1, 2, 3), ('idade', 'peso', 'altura', 'IMC'));
```



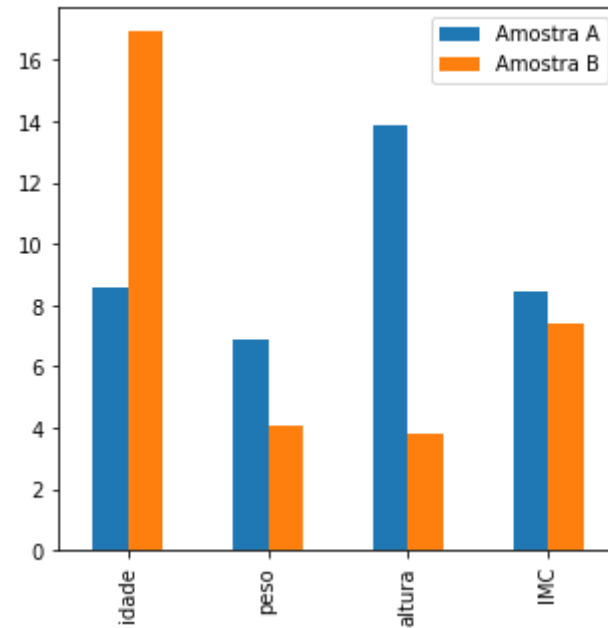
```
In [502]: varianciaA
```

```
Out[502]: [8.562053628310666, 6.9003819276712, 13.869744434591745, 8.475199460219937]
```

```
In [503]: varianciaB
```

```
Out[503]: [16.908738794794314, 4.085340806438105, 3.810109062426782, 7.408672330872284]
```

```
In [504]: # Variância
df_erro = pd.DataFrame(list(zip(varianciaA, varianciaB)),
                        columns=["Amostra A", "Amostra B"])
pd.concat([df_erro["Amostra A"],
            df_erro["Amostra B"]],
          axis=1).plot.bar(figsize=(5, 5));
plt.xticks((0, 1, 2, 3), ('idade', 'peso', 'altura', 'IMC'));
```



Observando os gráficos podemos concluir que na amostra B (laranja), houveram erros relativos menores, portanto nessa situação é a mais indicada para uma análise fidedigna que representa a realidade da população.

PARTE C

Nesse caso, como era uma amostra estratificada por ies, utilizei uma função que irá manter uma porcentagem semelhante à população.

```
In [462]: # amostra estratificada por IES
df_ies = df.groupby('IES', group_keys=False).apply(lambda x: x.sample(frac=0.
df_ies.to_excel("amostra_ies.xlsx")
```

```
In [505]: # ver os 5 valores iniciais da amostra
df_ies.head()
```

Out[505]:

	nº	sexo	idade	peso	altura	IES	IMC	Clas IMC
39	40	F	18	52	163	UFAC	19.6	ad
207	208	M	21	75	173	UFAC	25.1	ob
234	235	M	24	87	174	UFAC	28.7	ob
33	34	F	22	42	152	UFAC	18.2	mg
15	16	F	20	66	162	UFAC	25.1	ob

1 - Para as variáveis SEXO e IES, obter o número absoluto e a frequência relativa percentual.

```
In [464]: # número absoluto
df_ies.sexo.value_counts()
```

```
Out[464]: F    32
          M    22
          Name: sexo, dtype: int64
```

```
In [465]: # número absoluto
df_ies.IES.value_counts()
```

```
Out[465]: UFSCAR    33
          UFAC      21
          Name: IES, dtype: int64
```

```
In [466]: # frequência relativa em porcentagem
df_ies.sexo.value_counts()/df_ies.shape[0] * 100
```

```
Out[466]: F    59.259259
          M    40.740741
          Name: sexo, dtype: float64
```

```
In [467]: # frequência relativa em porcentagem
df_ies.IES.value_counts()/df_ies.shape[0] * 100
```

```
Out[467]: UFSCAR    61.111111
UFAC         38.888889
Name: IES, dtype: float64
```

Para os dados populacionais das variáveis IDADE, PESO, ALTURA e IMC as seguintes medidas e gráficos:

- i) Média aritmética.
- ii) Mediana.
- iii) Variância amostral.
- iv) Desvio Padrão amostral.
- v) Coeficiente de Variação.
- vi) Box-plot de cada uma das variáveis separadamente.

Média

```
In [468]: # média aritmética
print("Idade: ", round(df_ies.idade.mean(),2))
print("Peso: ", round(df_ies.peso.mean(),2))
print("Altura: ", round(df_ies.altura.mean(),2))
print("IMC: ", round(df_ies.IMC.mean(),2))
```

```
Idade:  21.59
Peso:   63.72
Altura: 166.11
IMC:    22.86
```

Erro relativo da média da amostra em relação à população

```
In [469]: # ErroRelativo = 100 x |ValorPopulacional - ValorAmostrual|/ValorPopulacional
# Média

mediaC= []
mediaC.append(100 * abs(df.idade.mean() - df_ies.idade.mean())/df.idade.mean()
mediaC.append(100 * abs(df.peso.mean() - df_ies.peso.mean())/df.peso.mean()
mediaC.append(100 * abs(df.altura.mean() - df_ies.altura.mean())/df.altura.me
mediaC.append(100 * abs(df.IMC.mean() - df_ies.IMC.mean())/df.IMC.mean())

print("Erro relativo Idade:")
print(mediaC[0], "\n")
print("Erro relativo Peso:")
print(mediaC[1], "\n")
print("Erro relativo Altura:")
print(mediaC[2], "\n")
print("Erro relativo IMC:")
print(mediaC[3])
```

Erro relativo Idade:
0.23956194387406893

Erro relativo Peso:
3.548604103599054

Erro relativo Altura:
1.2125417120956832

Erro relativo IMC:
1.620273538188226

Mediana


```
In [470]: # mediana
print("Idade: ", round(df_ies.idade.median(),2))
print("Peso: ", round(df_ies.peso.median(),2))
print("Altura: ", round(df_ies.altura.median(),2))
print("IMC: ", round(df_ies.IMC.median(),2))
```

```
Idade:  21.0
Peso:   62.0
Altura: 164.5
IMC:    22.8
```

Erro relativo da mediana da amostra em relação à população

```
In [471]: # ErroRelativo = 100 x |ValorPopulacional - ValorAmostral|/ValorPopulacional
#Mediana

medianaC = []
medianaC.append(100 * abs(df.idade.median() - df_ies.idade.median())/df.idade
medianaC.append(100 * abs(df.peso.median() - df_ies.peso.median())/df.peso.me
medianaC.append(100 * abs(df.altura.median() - df_ies.altura.median())/df.alt
medianaC.append(100 * abs(df.IMC.median() - df_ies.IMC.median())/df.IMC.media

print("Erro relativo Idade:")
print(medianaC[0], "\n")
print("Erro relativo Peso:")
print(medianaC[1], "\n")
print("Erro relativo Altura:")
print(medianaC[2], "\n")
print("Erro relativo IMC:")
print(medianaC[3])
```

```
Erro relativo Idade:
0.0
```

```
Erro relativo Peso:
3.125
```

```
Erro relativo Altura:
2.0833333333333335
```

```
Erro relativo IMC:
0.2188183807439856
```

Variância

```
In [472]: # variância amostral
print("Idade: ", round(std.variance(df_ies.idade),2))
print("Peso: ", round(std.variance(df_ies.peso),2))
print("Altura: ", round(std.variance(df_ies.altura),2))
print("IMC: ", round(std.variance(df_ies.IMC),2))
```

```
Idade:  13.57
Peso:  232.69
Altura:  72.7
IMC:  21.53
```

Erro relativo da variância da amostra em relação à população

```
In [506]: # ErroRelativo = 100 x |ValorPopulacional - ValorAmostral|/ValorPopulacional
# Variância

varianciaC = []
varianciaC.append(100 * abs(std.pvariance(df.idade) -std.variance(df_ies.idad
varianciaC.append(100 * abs(std.pvariance(df.peso) -std.variance(df_ies.peso)
varianciaC.append(100 * abs(std.pvariance(df.altura) -std.variance(df_ies.alt
varianciaC.append(100 * abs(std.pvariance(df.IMC) -std.variance(df_ies.IMC))/

print("Erro relativo Idade:")
print(varianciaC[0], "\n")
print("Erro relativo Peso:")
print(varianciaC[1], "\n")
print("Erro relativo Altura:")
print(varianciaC[2], "\n")
print("Erro relativo IMC:")
print(varianciaC[3], "\n")
```

Erro relativo Idade:
9.830422300846095

Erro relativo Peso:
3.0530662176218044

Erro relativo Altura:
20.531745282016345

Erro relativo IMC:
8.660979215684291

Desvio padrão

```
In [474]: # desvio padrão amostral
print("Idade: ", round(std.stdev(df_ies.idade),2))
print("Peso: ", round(std.stdev(df_ies.peso),2))
print("Altura: ", round(std.stdev(df_ies.altura),2))
print("IMC: ", round(std.stdev(df_ies.IMC),2))
```

```
Idade:  3.68
Peso:   15.25
Altura:  8.53
IMC:    4.64
```

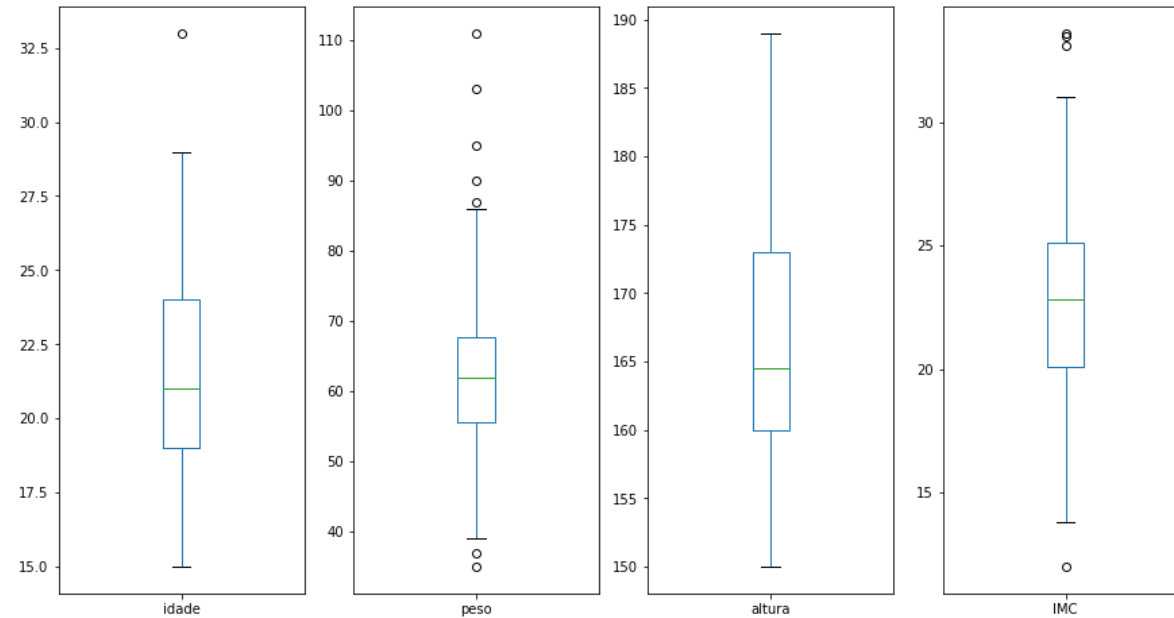
Coeficiente de variação

```
In [475]: # coeficiente de variação
print("Idade: ", round(std.stdev(df_ies.idade)/df_ies.idade.mean() * 100,2))
print("Peso: ", round(std.stdev(df_ies.peso)/df_ies.peso.mean() * 100,2))
print("Altura: ", round(std.stdev(df_ies.altura)/df_ies.altura.mean() * 100,2))
print("IMC: ", round(std.stdev(df_ies.IMC)/df_ies.IMC.mean() * 100,2))
```

```
Idade:  17.06
Peso:   23.94
Altura:  5.13
IMC:   20.29
```

Boxplot

```
In [476]: # boxplot para as variáveis quantitativas
fig, (ax1, ax2, ax3, ax4) = plt.subplots(1, 4, figsize = (15, 8))
df_ies.idade.plot(kind = 'box', ax = ax1);
df_ies.peso.plot(kind = 'box', ax = ax2);
df_ies.altura.plot(kind = 'box', ax = ax3);
df_ies.IMC.plot(kind = 'box', ax = ax4);
plt.show()
```



Comparando A, B e C

In [507]: mediaA

Out[507]: [3.319644079397672, 3.374817804686621, 0.7808284049735095, 2.10073822642398
◀ ▶ ▼

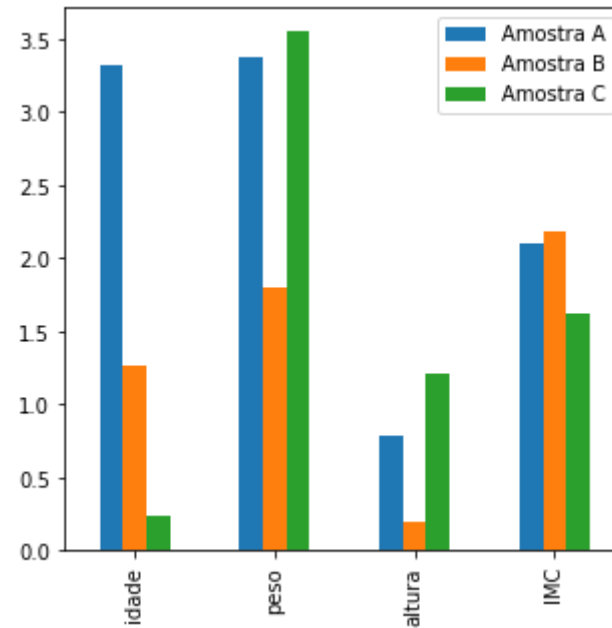
In [508]: mediaB

Out[508]: [1.2662559890486031,
1.8051351048323758,
0.19933701170692603,
2.1804172791993497]

In [509]: mediaC

Out[509]: [0.23956194387406893, 3.548604103599054, 1.2125417120956832, 1.620273538188
◀ ▶ ▼

```
In [510]: df_erro = pd.DataFrame(list(zip(mediaA, mediaB, mediaC)),
                                   columns=["Amostra A", "Amostra B", "Amostra C"])
# Média
pd.concat([df_erro["Amostra A"],
           df_erro["Amostra B"],
           df_erro["Amostra C"]],
          axis=1).plot.bar(figsize = (5, 5));
plt.xticks((0, 1, 2, 3), ('idade', 'peso', 'altura', 'IMC'));
```



```
In [511]: medianaA
```

```
Out[511]: [4.761904761904762, 2.34375, 1.1904761904761905, 2.4070021881837946]
```

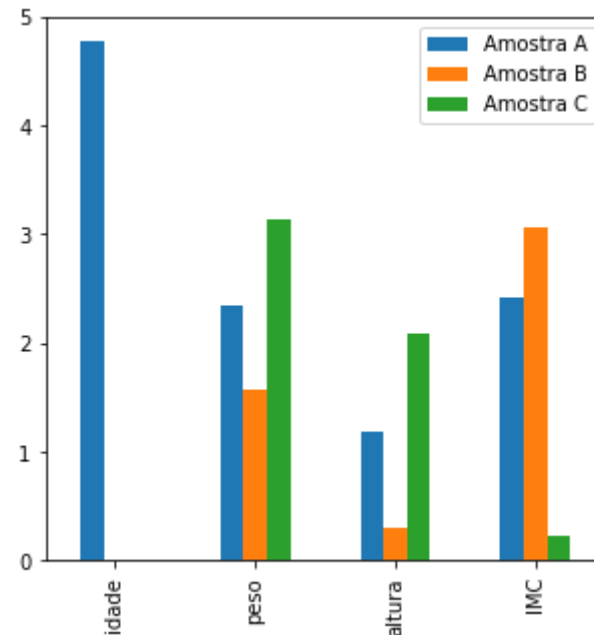
```
In [512]: medianaB
```

```
Out[512]: [0.0, 1.5625, 0.2976190476190476, 3.0634573304157517]
```

```
In [513]: medianaC
```

```
Out[513]: [0.0, 3.125, 2.0833333333333335, 0.2188183807439856]
```

```
In [514]: df_erro = pd.DataFrame(list(zip(medianaA, medianaB, medianaC)),
                                   columns=["Amostra A", "Amostra B", "Amostra C"])
# Mediana
pd.concat([df_erro["Amostra A"],
           df_erro["Amostra B"],
           df_erro["Amostra C"]],
          axis=1).plot.bar(figsize = (5, 5));
plt.xticks((0, 1, 2, 3), ('idade', 'peso', 'altura', 'IMC'));
```



```
In [515]: varianciaA
```

```
Out[515]: [8.562053628310666, 6.9003819276712, 13.869744434591745, 8.475199460219937]
```

```
In [516]: varianciaB
```

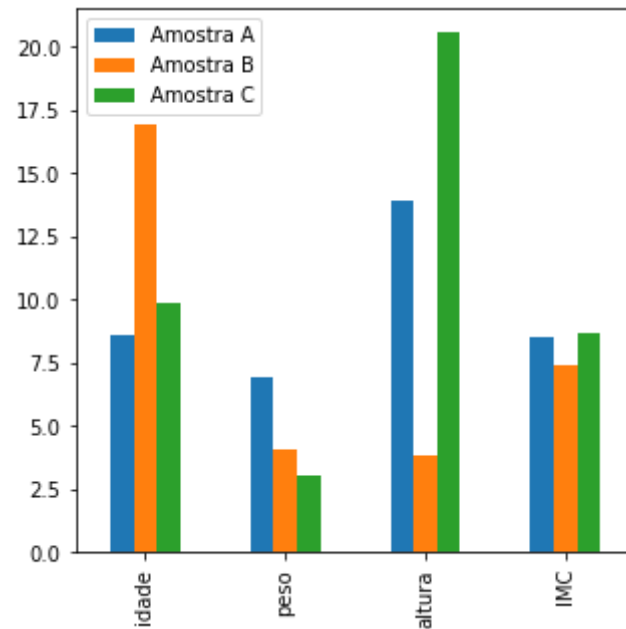
```
Out[516]: [16.908738794794314, 4.085340806438105, 3.810109062426782, 7.408672330872284]
```



```
In [517]: varianciaC
```

```
Out[517]: [9.830422300846095, 3.0530662176218044, 20.531745282016345, 8.660979215684291
```

```
In [518]: df_erro = pd.DataFrame(list(zip(varianciaA, varianciaB, varianciaC)),  
                                   columns=["Amostra A", "Amostra B", "Amostra C"])  
  
# Variância  
pd.concat([df_erro["Amostra A"],  
            df_erro["Amostra B"],  
            df_erro["Amostra C"]],  
           axis=1).plot.bar(figsize = (5, 5));  
plt.xticks((0, 1, 2, 3), ('idade', 'peso', 'altura', 'IMC'));
```



Observando os gráficos podemos concluir que ainda, na amostra B (laranja), houveram erros relativos menores, portanto nessa situação é a mais indicada para uma análise fidedigna que representa a realidade da população.