

Lista2

June 26, 2021

0.1 Lista 02 de Exercicios - FECD

O objetivo dessa lista é apresentar gráficos e histogramas que foram sugeridos de acordo com o slide da aula e entregar um notebook com o código/resultado obtido.

Leonardo Santos Miranda

0.2 1º Questão da lista:

- Usando pandas, ler os dados do arquivo BloodPressure01.txt
- reproduzir os comandos do slide 10 de T01-RDados.pdf
- histograma básico do slide 13
- graficos do slide 15

```
[3]: #Lendo dados com pandas
import os
import pandas as pd
os.chdir("D:/Lista02")
bp_data = pd.read_csv("BloodPressure01.txt")
print(data)
```

	sbp	gender	married	smoke	exercise	age	weight	height	overwt	race	\
0	133	F	N	N	3	60	159	56	3	1	
1	115	M	N	Y	1	55	107	65	1	1	
2	140	M	N	Y	1	18	130	59	2	1	
3	132	M	Y	N	2	19	230	57	3	2	
4	133	M	N	N	2	58	201	74	2	1	
..	
495	146	F	N	N	3	49	130	67	1	1	
496	181	F	N	Y	2	22	200	69	2	1	
497	145	M	Y	Y	3	42	249	75	3	1	
498	180	M	Y	Y	3	60	193	67	3	2	
499	174	F	N	Y	1	24	217	58	3	1	

	...	g474	g475	g476	g477	g478	g479	g480	g481	g482	g483
0	...	0_0	0_0	0_1	0_1	0_1	1_1	0_0	0_0	0_1	0_1
1	...	0_1	0_1	0_0	1_1	0_0	1_1	0_0	0_0	1_1	0_1
2	...	0_0	0_1	0_1	0_1	0_1	0_1	0_0	0_0	1_1	0_1
3	...	0_0	0_0	1_1	0_0	1_1	0_1	1_1	0_1	1_1	0_1
4	...	0_0	0_0	0_0	0_0	1_1	1_1	0_0	0_1	0_1	0_0

```

..    ...    ...    ...    ...    ...    ...    ...    ...    ...    ...
495  ...    0_0    0_1    0_0    0_1    0_1    0_1    0_1    0_1    0_1    0_1
496  ...    0_0    0_0    0_1    0_1    0_1    0_1    0_1    0_0    1_1    1_1
497  ...    0_0    0_1    0_1    0_1    1_1    1_1    0_1    0_0    0_0    0_0
498  ...    0_0    1_1    1_1    0_0    1_1    1_1    0_1    0_1    0_1    0_0
499  ...    0_0    0_1    0_1    1_1    0_1    0_1    1_1    1_1    0_0    0_0

```

[500 rows x 501 columns]

```

[4]: #Reproduzindo comandos do slides 10
import numpy as np
np.shape(bp_data) #Dimensão - equivalente a dim em R

```

[4]: (500, 501)

```

[6]: #Mostrando as 18 primeiras colunas
print(bp_data.iloc[:,0:18].columns)

```

```

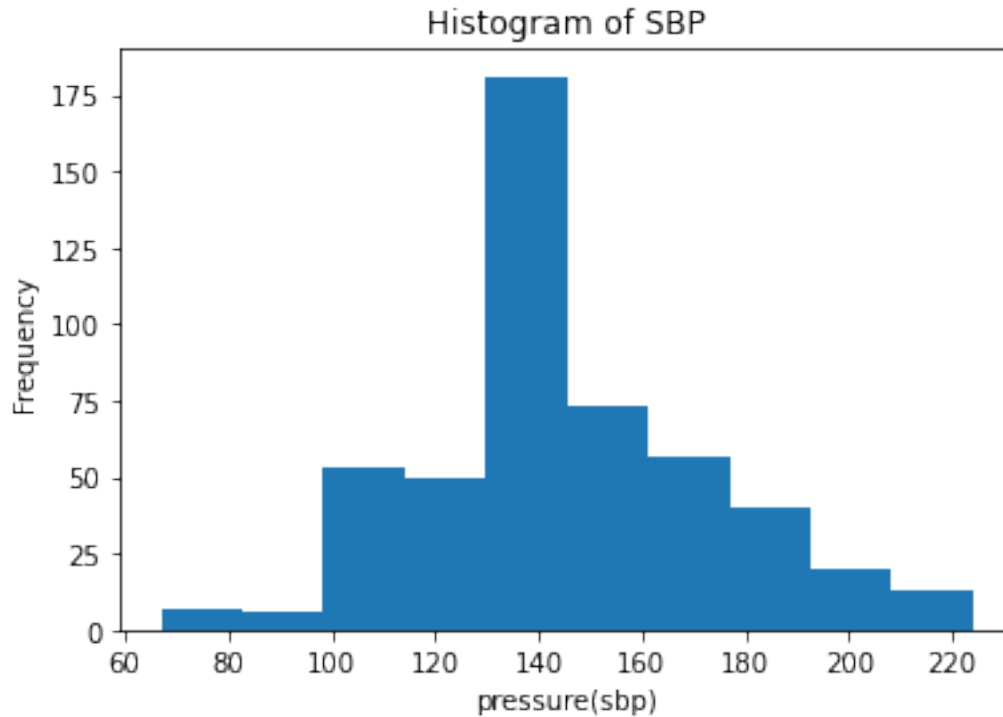
Index(['sbp', 'gender', 'married', 'smoke', 'exercise', 'age', 'weight',
      'height', 'overwt', 'race', 'alcohol', 'trt', 'bmi', 'stress', 'salt',
      'chldbear', 'income', 'educatn'],
      dtype='object')

```

```

[10]: #histograma básico do slide 13
import matplotlib.pyplot as plt
plt.hist(bp_data['sbp'])
plt.ylabel('Frequency')
plt.xlabel('pressure(sbp)')
plt.title('Histogram of SBP')
plt.show()

```



```
[17]: import seaborn as sea
fig, axes = plt.subplots(2,2,figsize= (20,10))
#esquerda superior
sea.histplot(ax = axes[0,0], data = bp_data,x="sbp",bins = 8)
axes[0,0].set_title("Histogram of SBP")
axes[0,0].set_xlabel("SBP")
axes[0,0].set_ylabel("Frequency")

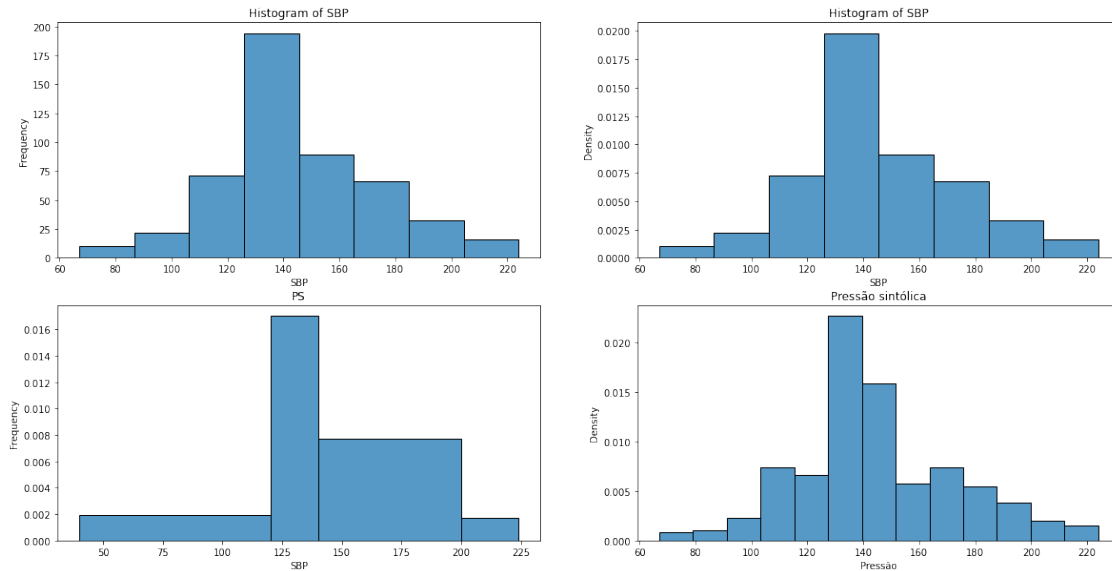
#direita superior
sea.histplot(ax = axes[0,1], data = bp_data, x="sbp", stat="density",bins = 8)
axes[0,1].set_title("Histogram of SBP")
axes[0,1].set_xlabel("SBP")
axes[0,1].set_ylabel("Density")

#esquerda inferior
bins = [40,120,140,200, max (bp_data["sbp"])]
sea.histplot(ax = axes[1,0], data = bp_data,x="sbp", stat="density",bins = bins)
axes[1,0].set_title("PS")
axes[1,0].set_xlabel("SBP")
axes[1,0].set_ylabel("Frequency")

#direita inferior
sea.histplot(ax = axes[1,1], data = bp_data,x="sbp", stat="density",bins = 13)
```

```
axes[1,1].set_title("Pressão sintólica")
axes[1,1].set_xlabel("Pressão")
axes[1,1].set_ylabel("Density")
```

```
[17]: Text(0, 0.5, 'Density')
```



0.3 2º Questão da lista:

- Usando pandas, ler os dados de população por município: arquivo POP2006.csv
- Refazer os histogramas dos slides 17 e 18

```
[23]: #lendo dados do pop2006 com pandas
pop2006_data = pd.read_csv("POP2006.csv", encoding = "ISO-8859-1")
print (pop2006_data)
```

	ESTADO	MUNICIPIO	POP2006
0	RO	Alta Floresta D'Oeste	29005
1	RO	Alto Alegre dos Parecis	15454
2	RO	Alto Paraíso	16510
3	RO	Alvorada D'Oeste	19542
4	RO	Ariquemes	86924
...
5559	GO	Vianópolis	12950
5560	GO	Vicentinópolis	6591
5561	GO	Vila Boa	3617
5562	GO	Vila Propício	5001
5563	DF	Brasília	2383784

```
[5564 rows x 3 columns]
```

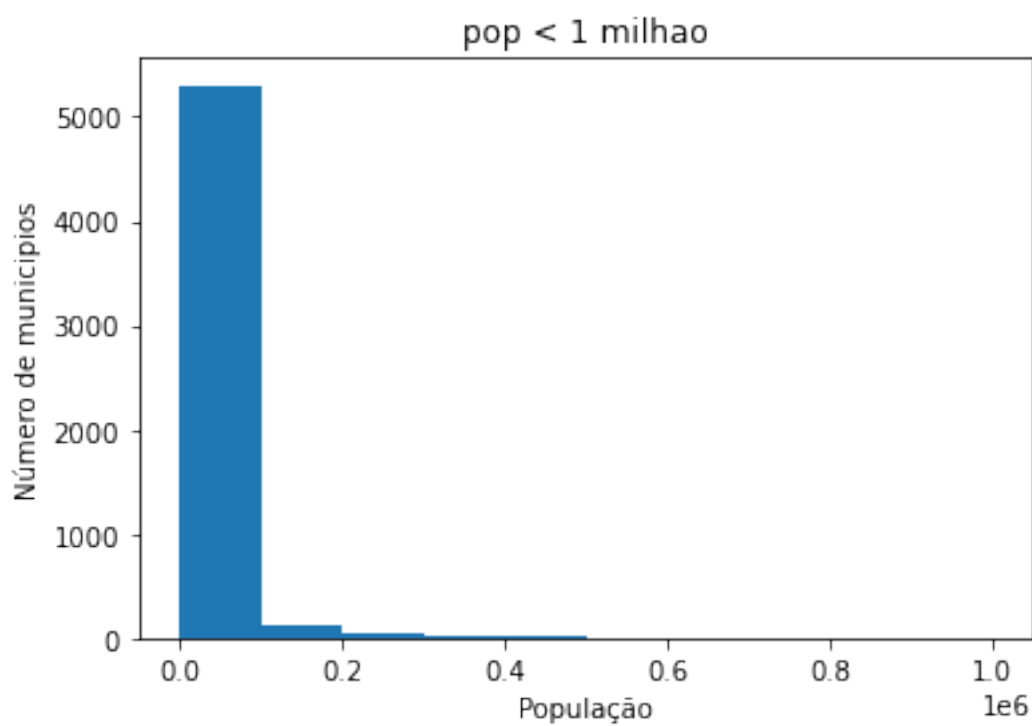
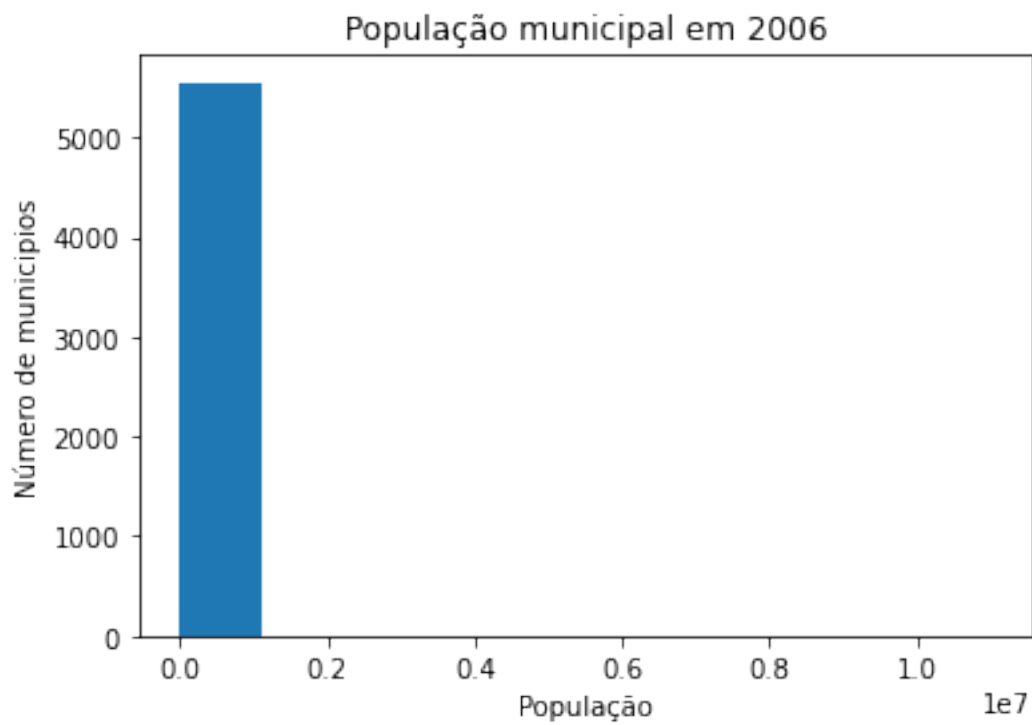
```
[32]: #Refazendo histogramas dos slides 17 e 18

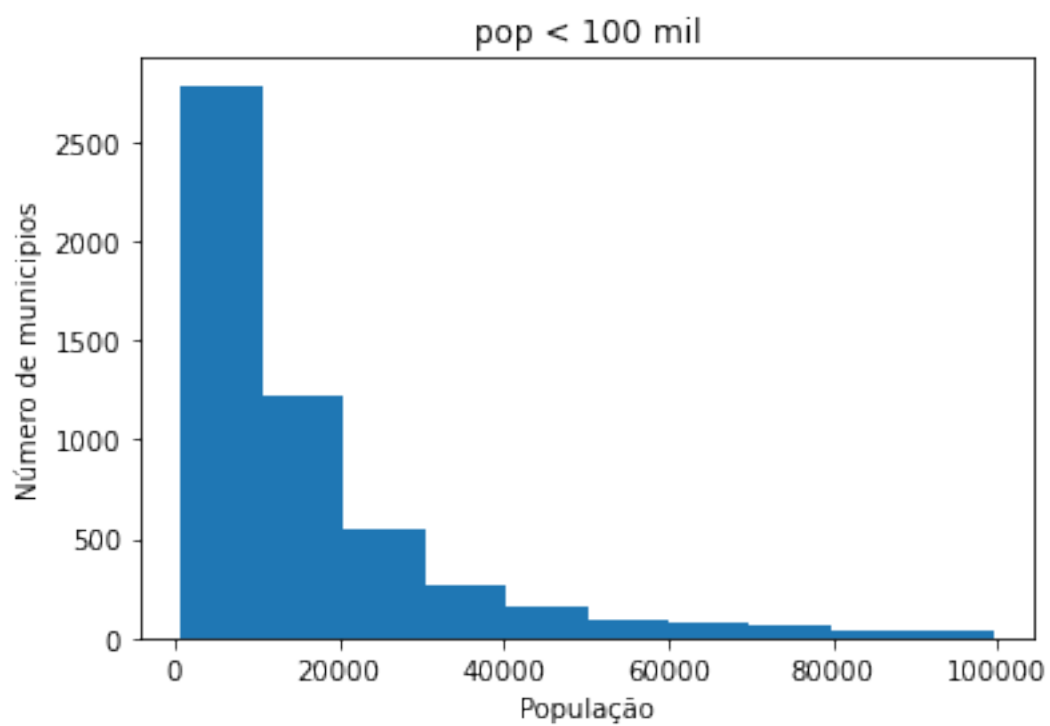
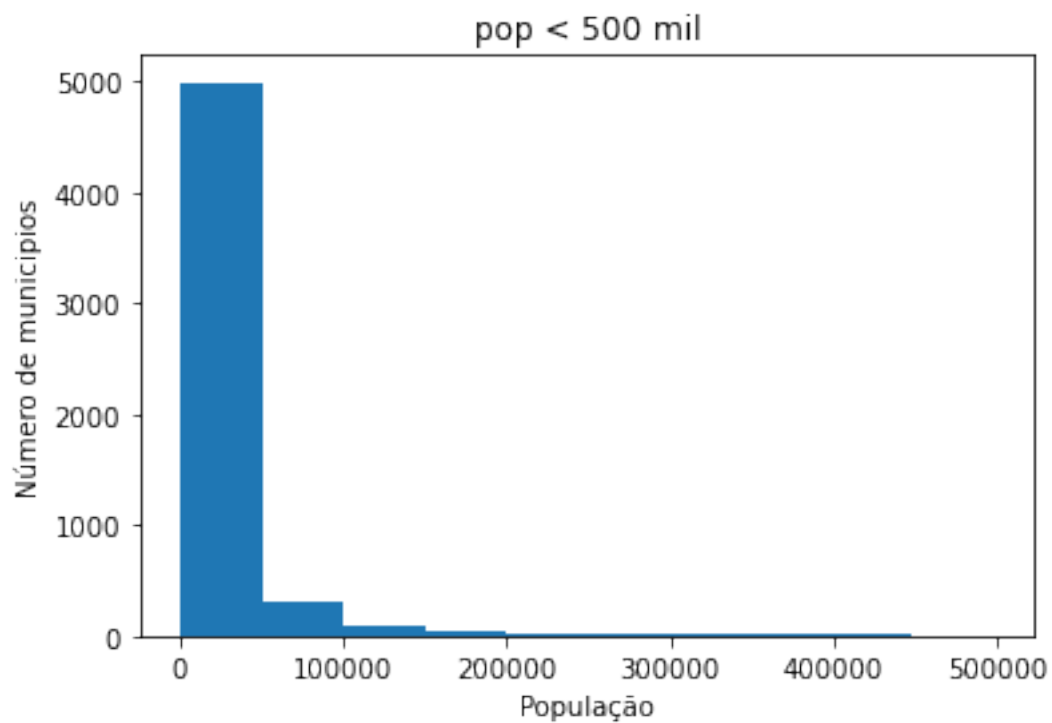
#Slide 17
#esquerda superior
plt.hist(pop2006_data['POP2006'])
plt.ylabel('Número de municípios')
plt.xlabel('População')
plt.title('População municipal em 2006')
plt.show()

#direita superior
plt.hist(pop2006_data['POP2006'].where(pop2006_data["POP2006"] < 1000000))
plt.ylabel('Número de municípios')
plt.xlabel('População')
plt.title('pop < 1 milhao')
plt.show()

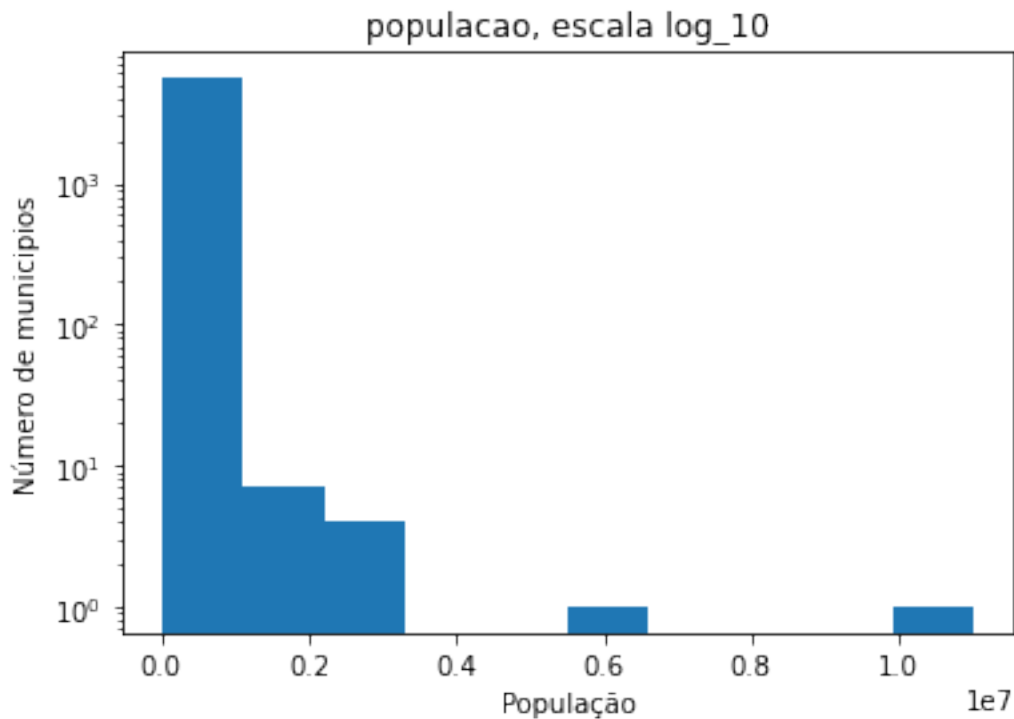
#esquerda inferior
plt.hist(pop2006_data['POP2006'].where(pop2006_data["POP2006"] < 500000))
plt.ylabel('Número de municípios')
plt.xlabel('População')
plt.title('pop < 500 mil')
plt.show()

#direita inferior
plt.hist(pop2006_data['POP2006'].where(pop2006_data["POP2006"] < 100000))
plt.ylabel('Número de municípios')
plt.xlabel('População')
plt.title('pop < 100 mil')
plt.show()
```





```
[33]: #Slide 18
plt.hist(pop2006_data['POP2006'])
plt.yscale("log")
plt.ylabel('Número de municípios')
plt.xlabel('População')
plt.title('populacao, escala log_10')
plt.show()
```



0.4 3º Questão da lista:

- Usando pandas, ler os dados de futebol do arquivo CampeonatoBrasileiro2014.txt
- Refazer os stem-and-leaf (ramo-e-folhas) dos slides 21 e 22
<https://www.geeksforgeeks.org/stem-and-leaf-plots-in-python/>

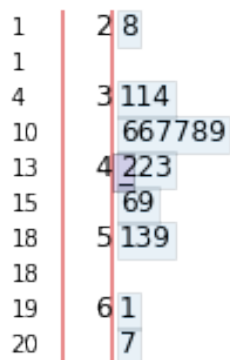
```
[35]: cb2014 = pd.read_csv("CampeonatoBrasileiro2014.txt")
print (cb2014.head())
```

	Time	\tPts	\tJogos	\tVit	\tEmp	\tDer	\tGols	\tGolsSofr	\
0	Cruzeiro	80	38	24	8	6	67	38	
1	Sao Paulo	70	38	20	10	8	59	40	
2	Internacional	69	38	21	6	11	53	41	
3	Corinthians	69	38	19	12	7	49	31	
4	Athletico Mineiro	62	38	17	11	10	51	38	

	\tSaldoGols	\tAprov
0	29	70
1	19	61
2	12	60
3	18	60
4	13	54

```
[44]: import stemgraphic as stem
      stem.stem_graphic (cb2014["\tGols"], asc=False)
```

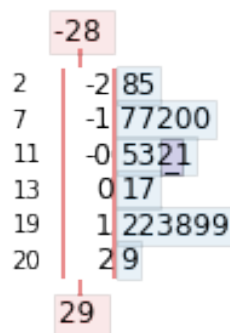
```
[44]: (<Figure size 540x234 with 1 Axes>,
      <matplotlib.axes._axes.Axes at 0x20758ecf6d0>)
```



20 | 6 | 1 = 6.1x10 = 61.0
Key: aggr|stem|leaf

```
[42]: stem.stem_graphic (cb2014["\tSaldoGols"], asc=False)
```

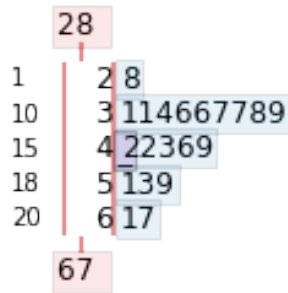
```
[42]: (<Figure size 540x162 with 1 Axes>,
      <matplotlib.axes._axes.Axes at 0x207590cbf40>)
```



20 | 2 | 9 = 2.9x10 = 29.0
Key: aggr|stem|leaf

```
[50]: stem.stem_graphic (cb2014["\tGoals"], scale = 10,asc =False)
```

```
[50]: (<Figure size 540x144 with 1 Axes>,
      <matplotlib.axes._axes.Axes at 0x207597dbac0>)
```

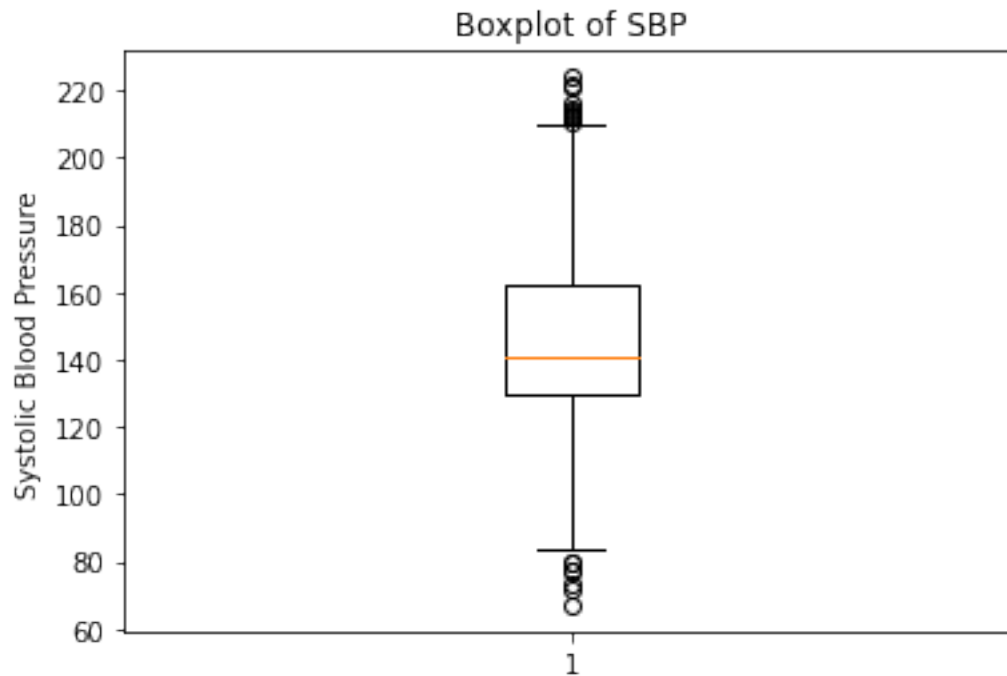


$x | 6 | 1 = 6.1 \times 10 = 61.0$
Key: aggr|stem|leaf

0.5 4° Questao da lista:

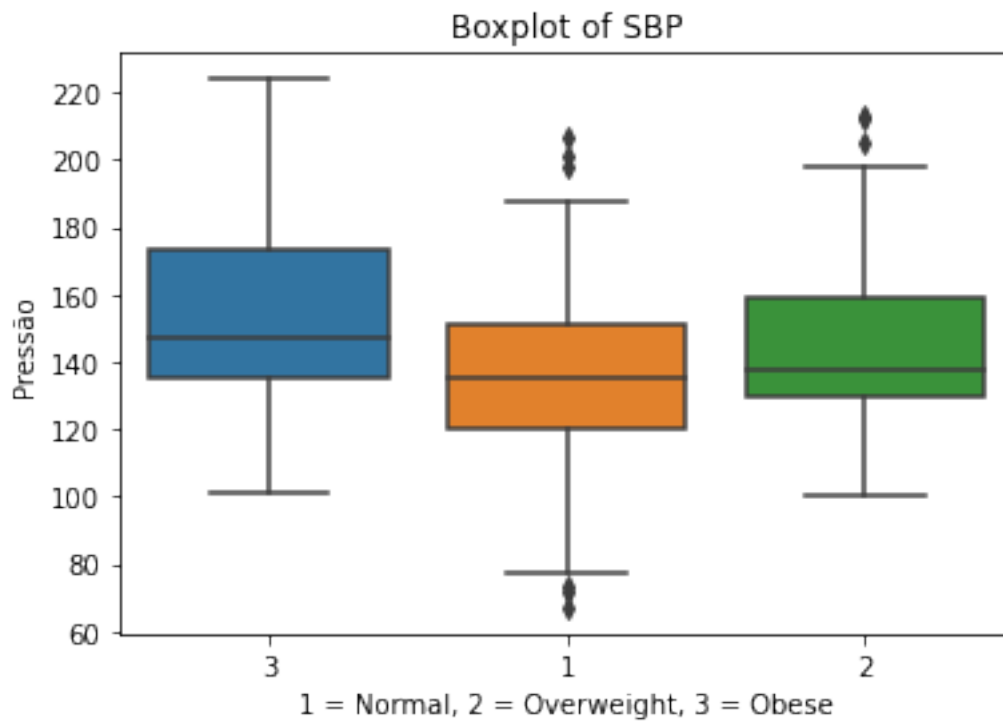
- Com os dados do arquivo de pressão sistalica, refazer o box-plot simples do slide 23
- Refazer slides 28, 29, 30, 33

```
[51]: #slide 23
plt.boxplot(bp_data["sbp"])
plt.ylabel('Pressão')
plt.title('Boxplot of SBP')
plt.show()
```



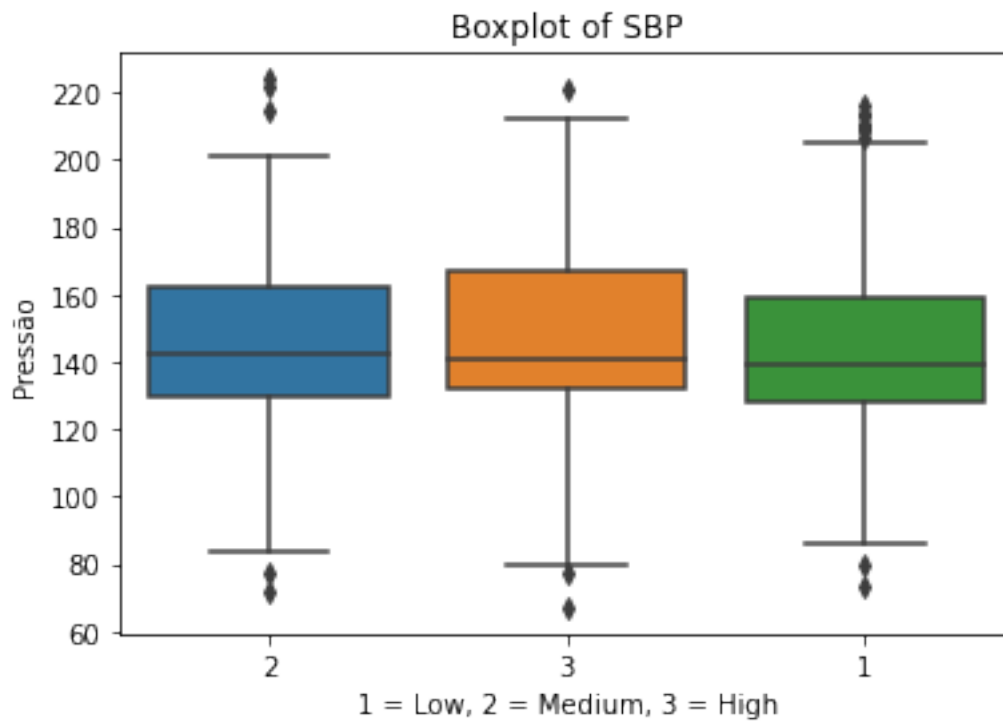
```
[57]: #slide 28
bp_data['overwt_mod'] = bp_data['overwt'].map({1: '1', 2: '2', 3: '3'})
box = sea.boxplot(x='overwt_mod', y='sbp', data=bp_data)
box.set_title('Boxplot of SBP')
box.set_ylabel('Pressão')
box.set_xlabel('1 = Normal, 2 = Overweight, 3 = Obese')
```

```
[57]: Text(0.5, 0, '1 = Normal, 2 = Overweight, 3 = Obese')
```



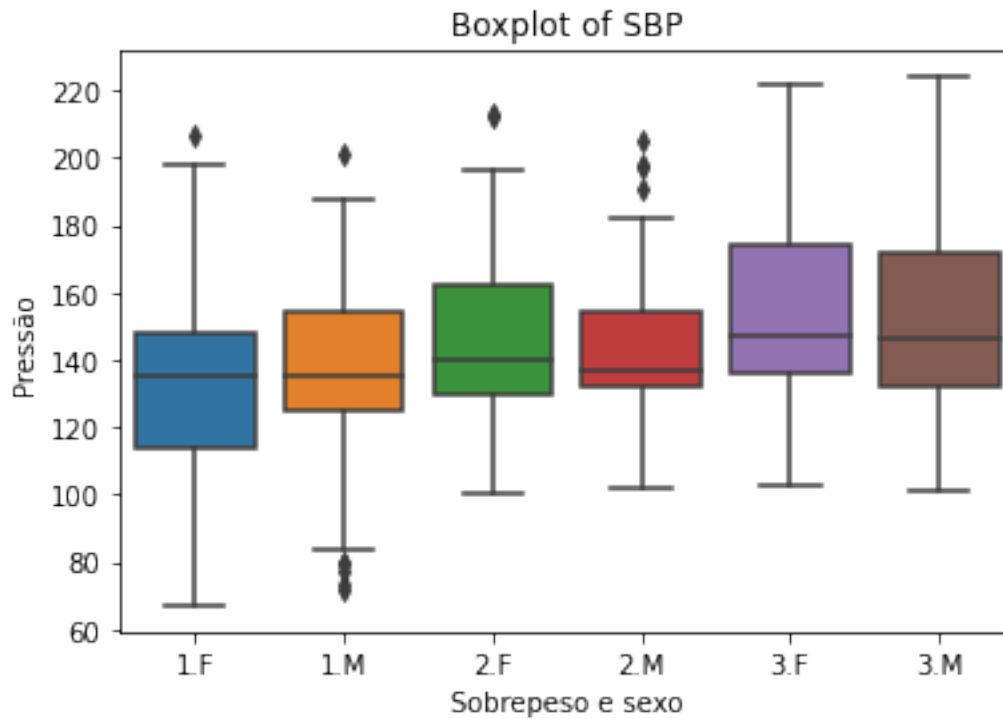
```
[59]: #slide 29
bp_data ['income_mod'] = bp_data['income'].map({1: '1', 2: '2', 3: '3'})
box = sea.boxplot(x='income_mod',y='sbp', data=bp_data)
box.set_title('Boxplot of SBP')
box.set_ylabel('Pressão')
box.set_xlabel('1 = Low, 2 = Medium, 3 = High')
```

```
[59]: Text(0.5, 0, '1 = Low, 2 = Medium, 3 = High')
```



```
[63]: #slide 30
bp_data ['overwt_gender'] = bp_data[['overwt', 'gender']].astype(str).agg('.'.
    ↪join, axis =1)
bp_data = bp_data.sort_values(by=['overwt', 'gender'])
box = sea.boxplot(x='overwt_gender',y ='sbp', data =bp_data)
box.set_title('Boxplot of SBP')
box.set_ylabel('Pressão')
box.set_xlabel('Sobrepeso e sexo')
```

[63]: Text(0.5, 0, 'Sobrepeso e sexo')

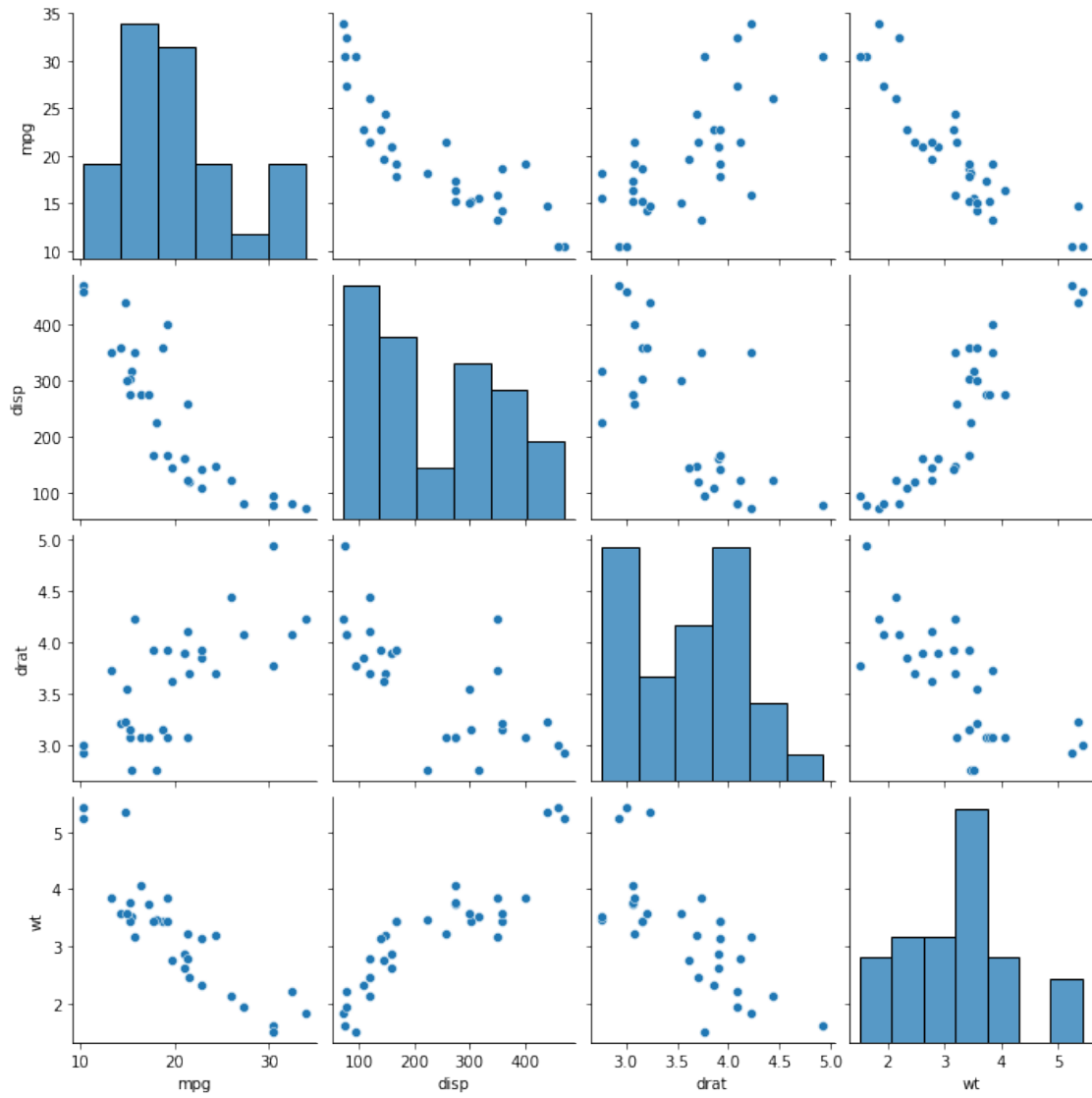


0.6 Questão 5 da lista:

- Usando os dados do arquivo mtcars.csv, fazer uma matriz de scatterplots como no slide 38. Ver a descrição das variáveis (que eu já não me lembrava em sala de aula) aqui: <https://rpubs.com/neros/61800>

```
[64]: #Lendo csv com pandas
mtcars = pd.read_csv("mtcars.csv", encoding = "ISO-8859-1")
#Plotando
sea.pairplot(mtcars[['mpg', 'disp', 'drat', 'wt']])
```

```
[64]: <seaborn.axisgrid.PairGrid at 0x2075acb0dc0>
```



[]: