saeb2017new

July 9, 2020

1 Explorando os dados SAEB 2017

Prof. Doherty Andrade

2 1. Introdução

A estatística divide-se em duas grandes áreas:

(a) estatística paramétrica

(b) estatística não-paramétrica.

Estatística paramétrica é o ramo da estatística que pressupõe que os dados são provenientes de algum tipo de distribuição de probabilidade e faz inferências sobre os parâmetros dessa distribuição.

Na verdade, os métodos paramétricos fazem mais suposições relativamente aos métodos da estatística não-paramétrica, com essas suposições adicionais os métodos paramétricos poderão produzir estimativas mais precisas. Mas se as hipóteses não forem satisfeitas os métodos paramétricos não poderão ser aplicados sob pena de retornar parâmetros errados. Por este motivo, são considerados menos robustos. Por outro lado, fórmulas paramétricas são comumente mais simples de se escrever e mais rápidas de computar.

A distribuição normal é uma das mais importantes distribuições. Antes de aplicar os testes de normalidade, precisamos saber como interpretar os resultados. Todo teste retorna pelo menos dois valores:

- estatística: um valor calculado pelo teste que pode ser usado para para comparar com o valor critico do teste.
- p-valor: usado para interpretar o teste.

Os testes assumem que a distriuição é normal, tecnicamente isto é chamado de hipótese nula H_0. Um bom limite é escolher $\alpha = 0.05$ que é usado pra interpretar o p-valor.

Se $p \leq \alpha = 0.05$, rejeitamos H_0 , a distribuição não é normal.

Se $p > \alpha$, falha ao rejeitar H_0 , a distribuição é normal.

2.1 2. Explorando os microdados do Saeb 2017

O SAEB é uma prova aplicada de modo censitário aos estudantes brasileiros do quinto e do nono ano do Ensino Fundamental e, também dos alunos do terceiro ano do Ensino Médio. A prova é aplicada a cada dois anos. Os dados mais recentes são da prova de 2017. Nesta prova os alunos respondem a questões de Língua Portugues e de Matemática, a correção é feita utilizando a Teoria de Resposta ao Item (TRI). Juntamente com as provas os estudantes respondem a um questionário socio-econômico. Também professores e diretores de escolas respondem a um longo questionário, ao todo são 8 arquivos de dados que compõem os microdados do SAEB.

Para baixar os dados veja o site do INEP http://inep.gov.br/microdados Os microdados do SAEB vem com os arquivos com dados separados por ano de matrícula do aluno. Vamos utilizar os dados dos alunos do nono ano de Ensino Fundamental: TS_ALUNO_9EF de 2017. O arquivo é do tipo csv e separado por ",".

O nosso objetivo é apresentar o Pandas em um exemplo concreto. Atualize o caminho para os dados no seu caso.

Primeiramente vamos importar os pacotes do Python que iremos utilizar.
pandas é o pacote para Estatística.
numpy é o pacote para Matemática.
matlibplot é o pacote para plotar gráficos.

```
In [1]: # Importando os pacotes
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt

        import colorsys
        plt.style.use('seaborn-talk')
        import warnings
        warnings.filterwarnings("ignore")
        %matplotlib inline
```

Para econimizar tempo de execução vamos tomar apenas parte do data set TS_ALUNO_9EF.csv com apenas 10000 linhas.

```
In [2]: # Carregando o dataset PARCIAL COM 10000 LINHAS
        df = pd.read_csv("C:/SAEB2017/microdados_saeb_2017/DADOS/TS_ALUNO_9EF.CSV", sep = ',',

In [3]: # Carregando o dataset COMPLETO. Habite este comando se assim desejar.
        #df = pd.read_csv("C:/SAEB2017/microdados_saeb_2017/DADOS/TS_ALUNO_9EF.CSV", sep = ','
```

Usamos head() para visualizar o cabeçalho do banco de dados.

```
In [4]: df.head()
```

Out[4]:

	ID_PROVA_BRASIL	ID_REGIAO	ID_UF	ID_MUNICIPIO	ID_AREA	ID_ESCOLA	\
0	2017	1	11	1100015	2	11024666	
1	2017	1	11	1100015	2	11024666	
2	2017	1	11	1100015	2	11024666	
3	2017	1	11	1100015	2	11024666	

```
4             2017          1     11       1100015        2   11024666

   ID_DEPENDENCIA_ADM  ID_LOCALIZACAO  ID_TURMA  ID_TURNO  ...  TX_RESP_Q048  \
0                   3               2    938098         2  ...           NaN
1                   3               2    938098         2  ...             A
2                   3               2    938098         2  ...             A
3                   3               2    938098         2  ...             A
4                   3               2    938098         2  ...             A

   TX_RESP_Q049  TX_RESP_Q050  TX_RESP_Q051  TX_RESP_Q052  TX_RESP_Q053  \
0           NaN           NaN           NaN           NaN           NaN
1             A             A             B             A             A
2             A             A             A             A             B
3             A             B             A             A             A
4             A             B             A             A             A

   TX_RESP_Q054  TX_RESP_Q055 TX_RESP_Q056 TX_RESP_Q057
0           NaN           NaN          NaN          NaN
1             A             A            B            C
2             A             A            B            A
3             A             A            B            A
4             A             A            B            C

[5 rows x 93 columns]
```

Com dropna() retiramos do dataset as observações com NaN. O novo dataset recebe o mesmo nome df.

```
In [5]: # drop NaN
        df = df.dropna()
```

Vamos listar os nomes de todas as colunas para termos conhecimento inicial das variáveis ou colunas.

```
In [6]: # Listar todas as colunas
        list(df)
```

```
Out[6]: ['ID_PROVA_BRASIL',
         'ID_REGIAO',
         'ID_UF',
         'ID_MUNICIPIO',
         'ID_AREA',
         'ID_ESCOLA',
         'ID_DEPENDENCIA_ADM',
         'ID_LOCALIZACAO',
         'ID_TURMA',
         'ID_TURNO',
         'ID_SERIE',
         'ID_ALUNO',
```

```
'IN_SITUACAO_CENSO',
'IN_PREENCHIMENTO_PROVA',
'IN_PRESENCA_PROVA',
'ID_CADERNO',
'ID_BLOCO_1',
'ID_BLOCO_2',
'TX_RESP_BLOCO_1_LP',
'TX_RESP_BLOCO_2_LP',
'TX_RESP_BLOCO_1_MT',
'TX_RESP_BLOCO_2_MT',
'IN_PROFICIENCIA',
'IN_PROVA_BRASIL',
'ESTRATO_ANEB',
'PESO_ALUNO_LP',
'PESO_ALUNO_MT',
'PROFICIENCIA_LP',
'ERRO_PADRAO_LP',
'PROFICIENCIA_LP_SAEB',
'ERRO_PADRAO_LP_SAEB',
'PROFICIENCIA_MT',
'ERRO_PADRAO_MT',
'PROFICIENCIA_MT_SAEB',
'ERRO_PADRAO_MT_SAEB',
'IN_PREENCHIMENTO_QUESTIONARIO',
'TX_RESP_Q001',
'TX_RESP_Q002',
'TX_RESP_Q003',
'TX_RESP_Q004',
'TX_RESP_Q005',
'TX_RESP_Q006',
'TX_RESP_Q007',
'TX_RESP_Q008',
'TX_RESP_Q009',
'TX_RESP_Q010',
'TX_RESP_Q011',
'TX_RESP_Q012',
'TX_RESP_Q013',
'TX_RESP_Q014',
'TX_RESP_Q015',
'TX_RESP_Q016',
'TX_RESP_Q017',
'TX_RESP_Q018',
'TX_RESP_Q019',
'TX_RESP_Q020',
'TX_RESP_Q021',
'TX_RESP_Q022',
'TX_RESP_Q023',
'TX_RESP_Q024',
```

```
'TX_RESP_Q025',
'TX_RESP_Q026',
'TX_RESP_Q027',
'TX_RESP_Q028',
'TX_RESP_Q029',
'TX_RESP_Q030',
'TX_RESP_Q031',
'TX_RESP_Q032',
'TX_RESP_Q033',
'TX_RESP_Q034',
'TX_RESP_Q035',
'TX_RESP_Q036',
'TX_RESP_Q037',
'TX_RESP_Q038',
'TX_RESP_Q039',
'TX_RESP_Q040',
'TX_RESP_Q041',
'TX_RESP_Q042',
'TX_RESP_Q043',
'TX_RESP_Q044',
'TX_RESP_Q045',
'TX_RESP_Q046',
'TX_RESP_Q047',
'TX_RESP_Q048',
'TX_RESP_Q049',
'TX_RESP_Q050',
'TX_RESP_Q051',
'TX_RESP_Q052',
'TX_RESP_Q053',
'TX_RESP_Q054',
'TX_RESP_Q055',
'TX_RESP_Q056',
'TX_RESP_Q057']
```

As variáveis numéricas 'PROFICIENCIA_LP_SAEB' e 'PROFICIENCIA_MT_SAEB' são as notas da prova em Língua Portuguesa e da prova de Matemática na escala do SAEB. As mesmas notas estão nas variáveis 'PROFICIENCIA_LP' e 'PROFICIENCIA_MT', mas em outra escala. Isso é uma quest"ao de historia para manter os dados antigos em concordância com os novos.

As variáveis nominais do tipo 'TX_RESP_Q001' até 'TX_RESP_Q057' são as respostas do questionário socio-econômico.

Vamos calcular as médias de'PROFICIENCIA_LP_SAEB' e 'PROFICIENCIA_MT_SAEB'.

```
In [7]: df.loc[:,"PROFICIENCIA_LP_SAEB"].mean()

Out[7]: 268.1024906595455

In [8]: df.loc[:,"PROFICIENCIA_MT_SAEB"].mean()

Out[8]: 267.75944060969476
```

Se o banco de dados contém apenas variáveis numéricas e que faz sentido calcular média, mediana, desvio padrão, as estatísticas básicas, podemos utilizar o comando describe(). Embora não seja o nosso caso, vamos fazer um teste.

```
In [9]: print(df.describe())
```

```
       ID_PROVA_BRASIL  ID_REGIAO   ID_UF  ID_MUNICIPIO      ID_AREA  \
count           4353.0     4353.0  4353.0  4.353000e+03  4353.000000
mean            2017.0        1.0    11.0  1.100107e+06     1.882380
std                0.0        0.0     0.0  5.811764e+01     0.322195
min             2017.0        1.0    11.0  1.100015e+06     1.000000
25%             2017.0        1.0    11.0  1.100049e+06     2.000000
50%             2017.0        1.0    11.0  1.100114e+06     2.000000
75%             2017.0        1.0    11.0  1.100148e+06     2.000000
max             2017.0        1.0    11.0  1.100205e+06     2.000000

          ID_ESCOLA  ID_DEPENDENCIA_ADM  ID_LOCALIZACAO       ID_TURMA  \
count  4.353000e+03         4353.000000     4353.000000    4353.000000
mean   1.446535e+07            2.136917        1.158511  847487.052607
std    1.269663e+07            0.343799        0.365262   71185.192998
min    1.100026e+07            2.000000        1.000000  726514.000000
25%    1.100752e+07            2.000000        1.000000  791695.000000
50%    1.101563e+07            2.000000        1.000000  849461.000000
75%    1.102775e+07            2.000000        1.000000  911028.000000
max    6.124941e+07            3.000000        2.000000  968843.000000

          ID_TURNO  ...  PESO_ALUNO_MT  PROFICIENCIA_LP  ERRO_PADRAO_LP  \
count  4353.000000  ...    4353.000000      4353.000000     4353.000000
mean      1.464737  ...       1.135492         0.328851        0.343311
std       0.507036  ...       0.155977         0.774656        0.054253
min       1.000000  ...       0.875000        -2.169548        0.257983
25%       1.000000  ...       1.037037        -0.176719        0.306256
50%       1.000000  ...       1.105263         0.375990        0.327084
75%       2.000000  ...       1.181818         0.877625        0.361901
max       3.000000  ...       2.438710         2.257527        0.628434

       PROFICIENCIA_LP_SAEB  ERRO_PADRAO_LP_SAEB  PROFICIENCIA_MT  \
count           4353.000000          4353.000000      4353.000000
mean             268.102491            18.914146         0.318381
std               42.678394             2.988955         0.766116
min              130.457295            14.213151        -2.036646
25%              240.248946            16.872673        -0.202121
50%              270.699543            18.020157         0.314965
75%              298.336303            19.938343         0.849288
max              374.359744            34.622542         2.862915

       ERRO_PADRAO_MT  PROFICIENCIA_MT_SAEB  ERRO_PADRAO_MT_SAEB  \
count     4353.000000           4353.000000          4353.000000
```

```
mean          0.398267          267.759441          22.260052
std           0.072644           42.819996           4.060218
min           0.259236          136.131495          14.489304
25%           0.343670          238.667368          19.208516
50%           0.386322          267.568508          21.592436
75%           0.439208          297.433064          24.548358
max           0.758988          409.979365          42.421606

       IN_PREENCHIMENTO_QUESTIONARIO
count                         4353.0
mean                             1.0
std                              0.0
min                              1.0
25%                              1.0
50%                              1.0
75%                              1.0
max                              1.0

[8 rows x 32 columns]
```

Podemos calcular apenas a média do dataframe usando o comando mean(). Note que para muitas de nossas variáveis não faz sentido calcular média.

```
In [10]: df.mean()

Out[10]: ID_PROVA_BRASIL                2.017000e+03
         ID_REGIAO                      1.000000e+00
         ID_UF                          1.100000e+01
         ID_MUNICIPIO                   1.100107e+06
         ID_AREA                        1.882380e+00
         ID_ESCOLA                      1.446535e+07
         ID_DEPENDENCIA_ADM             2.136917e+00
         ID_LOCALIZACAO                 1.158511e+00
         ID_TURMA                       8.474871e+05
         ID_TURNO                       1.464737e+00
         ID_SERIE                       9.000000e+00
         ID_ALUNO                       2.151826e+07
         IN_SITUACAO_CENSO              1.000000e+00
         IN_PREENCHIMENTO_PROVA         1.000000e+00
         IN_PRESENCA_PROVA              1.000000e+00
         ID_CADERNO                     1.110361e+01
         ID_BLOCO_1                     3.979784e+00
         ID_BLOCO_2                     4.048932e+00
         IN_PROFICIENCIA                1.000000e+00
         IN_PROVA_BRASIL                1.000000e+00
         ESTRATO_ANEB                   1.100107e+08
         PESO_ALUNO_LP                  1.135492e+00
```

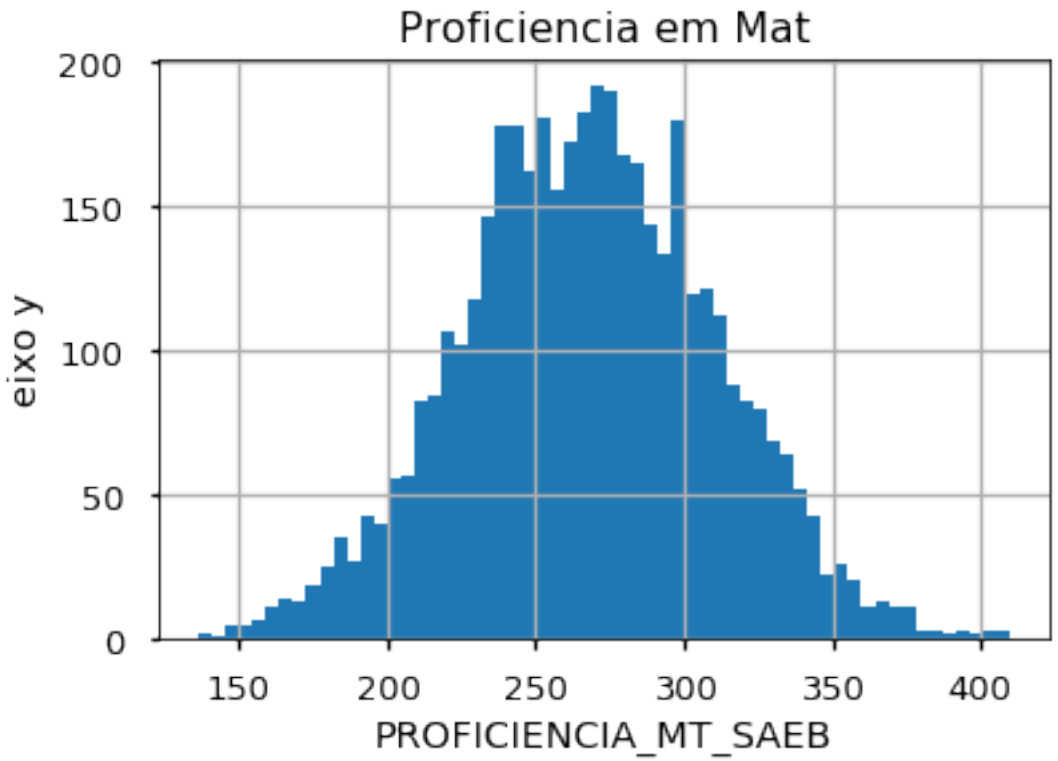
```
PESO_ALUNO_MT                    1.135492e+00
PROFICIENCIA_LP                  3.288509e-01
ERRO_PADRAO_LP                   3.433108e-01
PROFICIENCIA_LP_SAEB             2.681025e+02
ERRO_PADRAO_LP_SAEB              1.891415e+01
PROFICIENCIA_MT                  3.183811e-01
ERRO_PADRAO_MT                   3.982667e-01
PROFICIENCIA_MT_SAEB             2.677594e+02
ERRO_PADRAO_MT_SAEB              2.226005e+01
IN_PREENCHIMENTO_QUESTIONARIO    1.000000e+00
dtype: float64
```

3 3. Histograma

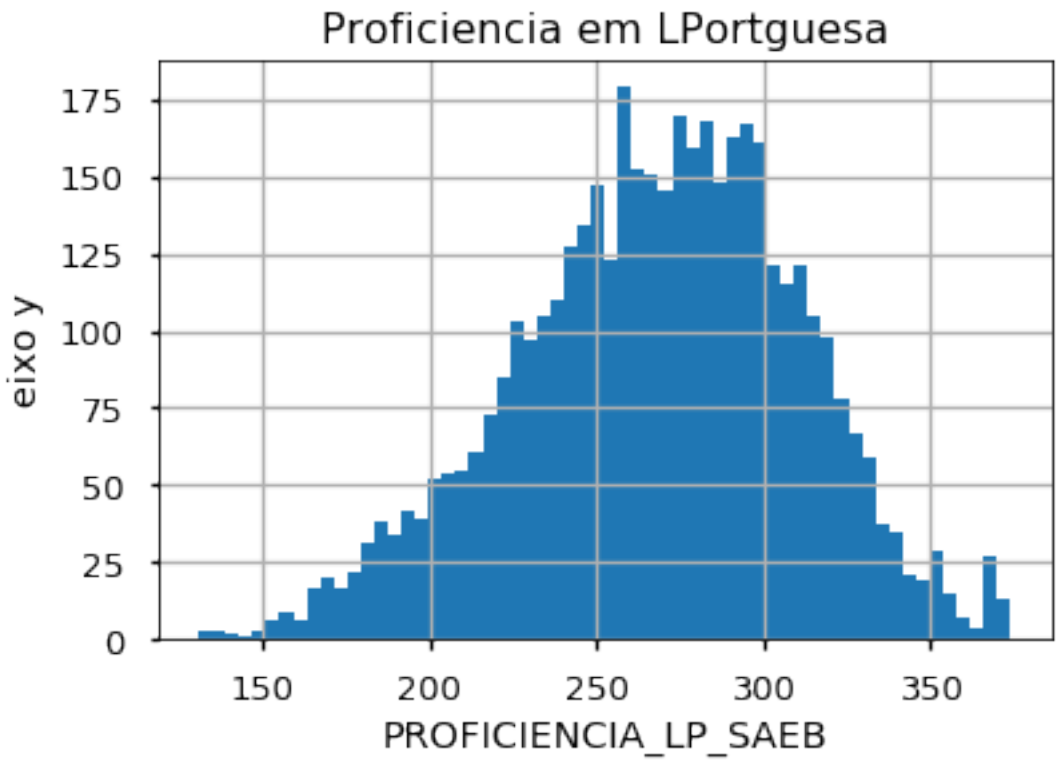
Com o histograma podemos ter uma ideia da distribuição das variáveis "PROFICIENCIA_MT_SAEB" e "PROFICIENCIA_LP_SAEB". Aqui usamos o pacote matlibplot.

Lembre-se que o nosso data frame foi denominado por df.

```
In [55]: df.PROFICIENCIA_MT_SAEB.hist(bins = 60)
         plt.xlabel("PROFICIENCIA_MT_SAEB")
         plt.ylabel("eixo y")
         plt.title("Proficiencia em Mat")
         plt.show()
```



```
In [12]: df.PROFICIENCIA_LP_SAEB.hist(bins = 60)
         plt.xlabel("PROFICIENCIA_LP_SAEB")
         plt.ylabel("eixo y")
         plt.title("Proficiencia em LPortguesa")
         plt.show()
```



Vamos trabalhar com uma coluna específica: PROFICIENCIA_MT_SAEB.
Depois vamos fazer o mesmo com a variável PROFICIENCIA_LP_SAEB.

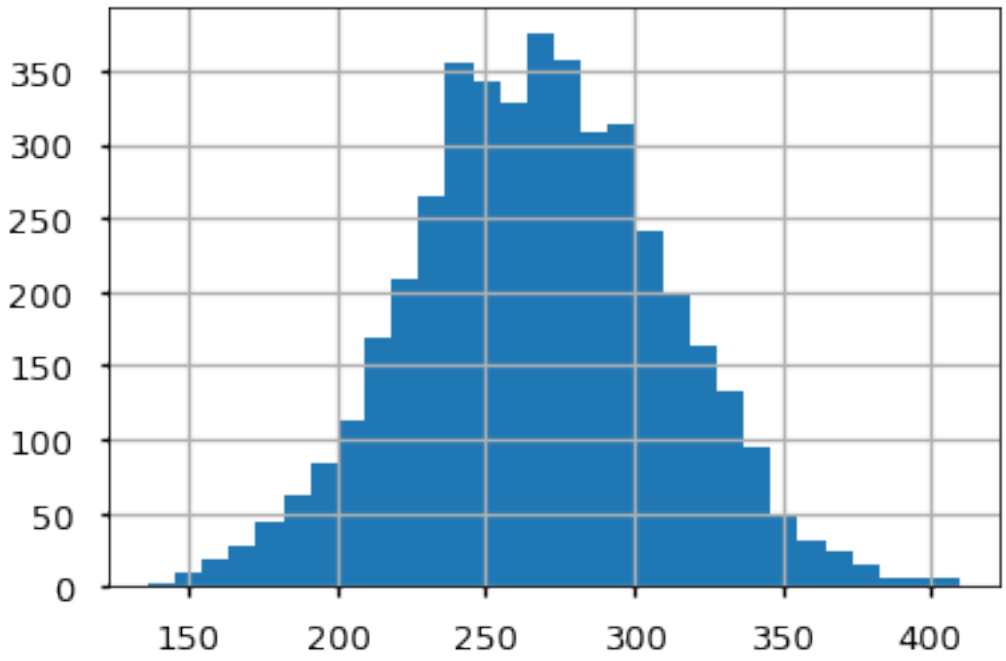
```
In [13]: coluna_mat =  df["PROFICIENCIA_MT_SAEB"]

In [14]: coluna_mat.value_counts().sort_index()

Out[14]: 136.131495    1
         137.703020    1
         142.503332    1
         146.054675    1
         146.932967    1
         147.810923    1
         148.634106    1
         148.881094    1
         150.494594    1
         151.031719    1
         151.993346    1
```

```
153.861659    1
153.956620    1
154.625875    1
156.178340    1
156.317959    1
156.965472    1
157.124094    1
157.317761    1
158.285537    1
158.958201    1
159.227267    1
159.230453    1
159.384268    1
160.205215    1
160.546885    1
161.037955    1
161.889977    1
161.962023    1
162.050947    1
              ..
372.643905    1
373.197127    1
373.576636    1
373.971627    1
374.202295    1
374.573531    1
375.026092    1
375.306112    1
375.366085    1
375.921095    1
376.776528    1
376.873165    1
377.153801    1
379.061741    1
380.476041    2
383.412176    1
383.906712    1
385.002201    1
388.010997    1
389.535125    1
392.731775    1
393.184838    1
395.257829    1
397.381290    1
397.946026    1
402.197364    2
403.100864    1
406.398958    1
```

```
409.137235    1
409.979365    1
Name: PROFICIENCIA_MT_SAEB, Length: 4350, dtype: int64
```

```
In [15]: #histograma
         coluna_mat.hist(bins=30)
```

```
Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x1b58f829550>
```




```
In [16]: coluna_port =  df["PROFICIENCIA_LP_SAEB"]
```

```
In [17]: coluna_port.value_counts().sort_index()
```

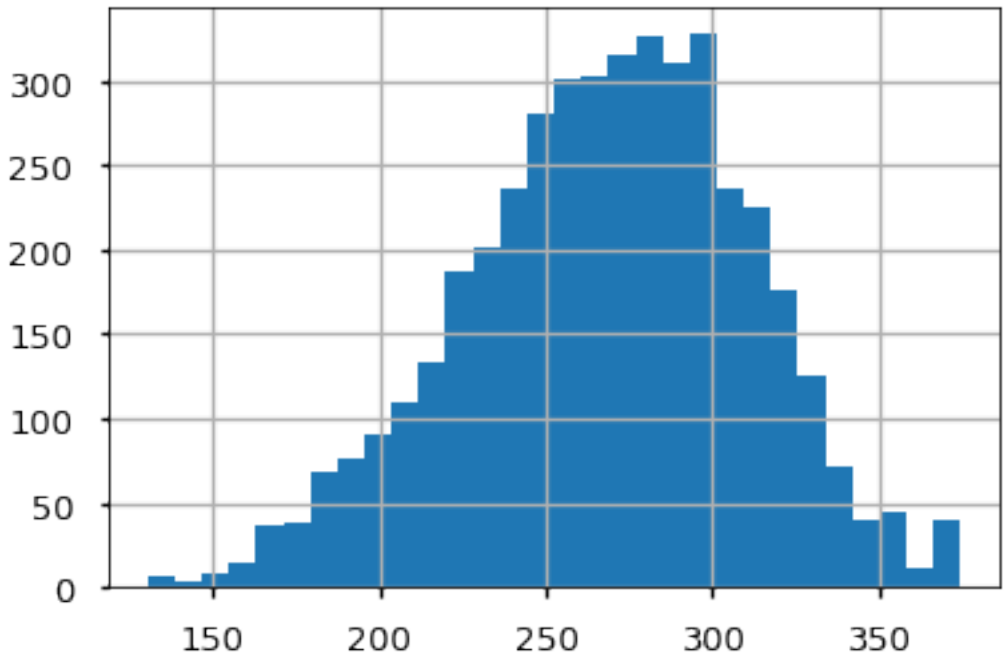
```
Out[17]: 130.457295    1
         131.771547    1
         131.912751    1
         137.790001    1
         138.503019    1
         138.578497    1
         140.174056    1
         141.445170    1
         145.183090    1
         148.085849    1
         148.942440    1
         150.535961    1
         150.996817    1
```

```
151.498938    1
151.821013    1
152.883654    1
153.107444    1
154.188155    1
155.361974    1
155.534031    1
156.030312    1
156.081549    1
156.121381    1
156.251677    1
156.256470    1
156.392606    1
157.930592    1
159.722779    1
160.427699    1
160.807898    1
              ..
356.493352    2
357.392421    1
357.616706    1
358.565193    2
358.821047    1
359.118496    1
360.440241    1
361.237607    1
361.335177    1
362.258652    1
363.246752    1
364.491917    1
365.302891    1
367.682318    3
368.254187    3
368.391976    2
368.489216    1
368.900763    2
368.972274    1
369.252314    2
369.314349    2
369.563426    4
369.614993    4
370.208184    3
372.714822    3
373.080366    3
373.399687    2
373.803191    1
374.129344    1
374.359744    3
```

```
        Name: PROFICIENCIA_LP_SAEB, Length: 4263, dtype: int64
```

```
In [18]: #histograma
         coluna_port.hist(bins=30)
```

```
Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x1b58f8c2630>
```



Agora vamos selecionar três variáveis 'PROFICIENCIA_LP_SAEB', 'PROFICIENCIA_MT_SAEB', 'TX_RESP_Q001' e fazer um estudo sobre suas relações. Sendo que 'TX_RESP_Q001' é a declaração de gênero do aluno.

```
In [19]: colunaS = ('PROFICIENCIA_LP_SAEB','PROFICIENCIA_MT_SAEB','TX_RESP_Q001')
```

Agora vamos criar um dataframe com estas colunas selecionadas.

```
In [20]: saebmtlpgen = df.filter(items = colunaS)
```

Vamos ver o cabeçalho deste dataframe (saebmtlpgen).

```
In [21]: saebmtlpgen.head()
```

```
Out[21]:    PROFICIENCIA_LP_SAEB  PROFICIENCIA_MT_SAEB TX_RESP_Q001
         1            370.208184            360.521641            A
         2            226.748923            220.594975            B
         3            268.226347            249.367227            B
         4            274.938923            273.207429            B
         6            263.443252            245.078721            B
```

Em seguida vamos agrupar as variáveis 'PROFICIENCIA_LP_SAEB' e 'PROFICIENCIA_MT_SAEB' em função da variável'TX_RESP_Q001' e suas frequências.

```
In [22]: saebmtlpgen.groupby('TX_RESP_Q001',).count()
```

```
Out[22]:               PROFICIENCIA_LP_SAEB  PROFICIENCIA_MT_SAEB
         TX_RESP_Q001
         A                             1949                  1949
         B                             2404                  2404
```

Em seguida vamos agrupar as variáveis 'PROFICIENCIA_LP_SAEB' e 'PROFICIENCIA_MT_SAEB'em função da variável'TX_RESP_Q001' e suas médias.

```
In [23]: saebmtlpgen.groupby('TX_RESP_Q001',).mean()
```

```
Out[23]:               PROFICIENCIA_LP_SAEB  PROFICIENCIA_MT_SAEB
         TX_RESP_Q001
         A                        261.935382            275.180076
         B                        273.102364            261.743293
```

Daqui vemos que os meninos possuem desempenho melhor em Matemática do que as meninas e meninas melhor desempenho em Língua Portuguesa. Mas cuidado! Não podemos tirar conclusões precipitadas, pois utilizamos um pequena amostra.

Podemos obter mais estatísticas das variáveis 'PROFICIENCIA_LP_SAEB' e 'PROFICIENCIA_MT_SAEB' agrupadas em função da variável'TX_RESP_Q001', usando o comando describe(). Obtemos frequência, média, desvio padrão, valor mínimo, quartis e valor máximo.

```
In [24]: saebmtlpgen.groupby('TX_RESP_Q001',).describe()
```

```
Out[24]:               PROFICIENCIA_LP_SAEB                                    \
                                      count        mean        std         min
         TX_RESP_Q001
         A                           1949.0  261.935382  45.603002  130.457295
         B                           2404.0  273.102364  39.459412  137.790001

                                                                  \
                                25%         50%         75%         max
         TX_RESP_Q001
         A                230.725452  264.234943  295.424618  374.359744
         B                247.574146  274.758657  299.827831  374.129344

                        PROFICIENCIA_MT_SAEB                                    \
                                      count        mean        std         min
         TX_RESP_Q001
         A                           1949.0  275.180076  44.331437  136.131495
         B                           2404.0  261.743293  40.579071  142.503332


                                25%         50%         75%         max
         TX_RESP_Q001
         A                245.336608  275.896688  305.532869  409.979365
         B                234.440496  260.967484  289.485874  406.398958
```

4 4. Utilizando dicionário

Dentre os vários indicadores socio-econômicos contidos no questionário do aluno encontramos informações sobre cor declarada da pele (TX_RESP_Q002). A escolaridade da mãe (TX_RESP_Q019) e escolaridade do pai(TX_RESP_Q023) com respostas de A até F são:

A: Nunca estudou.
B: Não completou a 4.ł série/5.ž ano.
C:Completou a 4.ł série/5.ž ano, mas não completou a 8.ł série/9.ž ano.
D: Completou a 8.ł série/9.ž ano, mas não completou o Ensino Médio.
E: Completou o Ensino Médio, mas não completou a Faculdade. Completou a Faculdade.
F: Não sei.

Para facilitar a interpretação é muito comum criar um dicionário. Vamos ver como se faz?

Primeiramente, vamos selecionar as variáveis mencionadas. Depois vamos filtar o data set utilizando-as.

```
In [25]: colunaS2 = ('PROFICIENCIA_LP_SAEB','PROFICIENCIA_MT_SAEB','TX_RESP_Q001','TX_RESP_Q00
```

```
In [26]: saebmtlpgen_cor_mae_pai = df.filter(items = colunaS2)
```

```
In [27]: saebmtlpgen_cor_mae_pai.head()
```

```
Out[27]:    PROFICIENCIA_LP_SAEB  PROFICIENCIA_MT_SAEB TX_RESP_Q001 TX_RESP_Q002  \
         1            370.208184            360.521641            A            C
         2            226.748923            220.594975            B            C
         3            268.226347            249.367227            B            C
         4            274.938923            273.207429            B            A
         6            263.443252            245.078721            B            B

           TX_RESP_Q019 TX_RESP_Q023
         1            D            F
         2            B            B
         3            G            G
         4            G            C
         6            B            G
```

```
In [28]: #criando um dicionario
         q0012024dicionario = {'A':'Nunca estudou',
         'B': 'Não completou a 4ł série/5ž ano do Ensino Fundamental',
         'C': 'Completou a 4ł série/5ž ano, mas não completou a 8ł série/9ž ano do Ensino Funda
         'D': 'Completou a 8ł série/9ž ano do Ensino Fundamental, mas não completou o Ensino Me
         'E': 'Completou o Ensino Médio, mas não completou a Faculdade',
         'F': 'Completou a Faculdade, mas não completou a Pós-graduação',
         'G': 'Completou a Pós-graduação',
         'H': 'Não sei'}
```

Agora as respostas A, B, C, D, E e F serão substituídas pelas seus respectivos valores dados no dicionário. Tente o comando abaixo.

```
In [29]: q0012024dicionario['C']
```

```
Out[29]: 'Completou a 4ł série/5ž ano, mas não completou a 8ł série/9ž ano do Ensino Fundamenta
```

Vamos criar uma coluna com as respostas relativas ao grau de escolaridade da mãe com a utilização do dicionário.

```
In [30]: #criando uma coluna
         saebmtlpgen_cor_mae_pai['TX_RESP_Q019']=[q0012024dicionario[resp] for resp in saebmtlp

In [31]: saebmtlpgen_cor_mae_pai.TX_RESP_Q019

Out[31]: 1       Completou a 8ł série/9ž ano do Ensino Fundamen...
         2       Não completou a 4ł série/5ž ano do Ensino Fund...
         3                                 Completou a Pós-graduação
         4                                 Completou a Pós-graduação
         6       Não completou a 4ł série/5ž ano do Ensino Fund...
         7       Completou a 4ł série/5ž ano, mas não completou...
         9       Não completou a 4ł série/5ž ano do Ensino Fund...
         12                                Completou a Pós-graduação
         14      Não completou a 4ł série/5ž ano do Ensino Fund...
         17                                Completou a Pós-graduação
         18                                Completou a Pós-graduação
         27      Completou a Faculdade, mas não completou a Pós...
         30      Completou a Faculdade, mas não completou a Pós...
         32      Não completou a 4ł série/5ž ano do Ensino Fund...
         33      Não completou a 4ł série/5ž ano do Ensino Fund...
         34                                           Nunca estudou
         38      Completou a 4ł série/5ž ano, mas não completou...
         39      Não completou a 4ł série/5ž ano do Ensino Fund...
         41      Completou a Faculdade, mas não completou a Pós...
         42      Completou a 4ł série/5ž ano, mas não completou...
         45      Não completou a 4ł série/5ž ano do Ensino Fund...
         47      Completou a Faculdade, mas não completou a Pós...
         49      Completou o Ensino Médio, mas não completou a ...
         53      Completou a 4ł série/5ž ano, mas não completou...
         54      Completou a Faculdade, mas não completou a Pós...
         55      Completou o Ensino Médio, mas não completou a ...
         56      Completou a Faculdade, mas não completou a Pós...
         58      Completou a 4ł série/5ž ano, mas não completou...
         60                                Completou a Pós-graduação
         62      Completou o Ensino Médio, mas não completou a ...
                                        ...
         9942    Completou a 4ł série/5ž ano, mas não completou...
         9943    Completou a Faculdade, mas não completou a Pós...
         9944    Completou o Ensino Médio, mas não completou a ...
         9947    Completou a Faculdade, mas não completou a Pós...
         9948                              Completou a Pós-graduação
         9949    Completou a 8ł série/9ž ano do Ensino Fundamen...
         9950                              Completou a Pós-graduação
         9951    Completou o Ensino Médio, mas não completou a ...
```

```
9952    Completou a 8ł série/9ž ano do Ensino Fundamen...
9954                            Completou a Pós-graduação
9955    Completou a 8ł série/9ž ano do Ensino Fundamen...
9956    Completou a Faculdade, mas não completou a Pós...
9958    Completou o Ensino Médio, mas não completou a ...
9959    Completou a Faculdade, mas não completou a Pós...
9961    Completou a Faculdade, mas não completou a Pós...
9962    Completou a 8ł série/9ž ano do Ensino Fundamen...
9965    Completou o Ensino Médio, mas não completou a ...
9968    Não completou a 4ł série/5ž ano do Ensino Fund...
9972                            Completou a Pós-graduação
9975    Completou a Faculdade, mas não completou a Pós...
9978    Completou a Faculdade, mas não completou a Pós...
9979                            Completou a Pós-graduação
9981    Completou a Faculdade, mas não completou a Pós...
9983    Completou a Faculdade, mas não completou a Pós...
9989    Completou a 8ł série/9ž ano do Ensino Fundamen...
9992    Completou o Ensino Médio, mas não completou a ...
9993    Completou o Ensino Médio, mas não completou a ...
9995    Completou o Ensino Médio, mas não completou a ...
9998    Completou a 4ł série/5ž ano, mas não completou...
9999    Completou a Faculdade, mas não completou a Pós...
Name: TX_RESP_Q019, Length: 4353, dtype: object
```

```
In [32]: coluna_mae = saebmtlpgen_cor_mae_pai['TX_RESP_Q019']
```

```
In [33]: coluna_mae
```

```
Out[33]: 1       Completou a 8ł série/9ž ano do Ensino Fundamen...
         2       Não completou a 4ł série/5ž ano do Ensino Fund...
         3                               Completou a Pós-graduação
         4                               Completou a Pós-graduação
         6       Não completou a 4ł série/5ž ano do Ensino Fund...
         7       Completou a 4ł série/5ž ano, mas não completou...
         9       Não completou a 4ł série/5ž ano do Ensino Fund...
         12                              Completou a Pós-graduação
         14      Não completou a 4ł série/5ž ano do Ensino Fund...
         17                              Completou a Pós-graduação
         18                              Completou a Pós-graduação
         27      Completou a Faculdade, mas não completou a Pós...
         30      Completou a Faculdade, mas não completou a Pós...
         32      Não completou a 4ł série/5ž ano do Ensino Fund...
         33      Não completou a 4ł série/5ž ano do Ensino Fund...
         34                                           Nunca estudou
         38      Completou a 4ł série/5ž ano, mas não completou...
         39      Não completou a 4ł série/5ž ano do Ensino Fund...
         41      Completou a Faculdade, mas não completou a Pós...
         42      Completou a 4ł série/5ž ano, mas não completou...
```

```
45      Não completou a 4ł série/5ž ano do Ensino Fund...
47      Completou a Faculdade, mas não completou a Pós...
49      Completou o Ensino Médio, mas não completou a ...
53      Completou a 4ł série/5ž ano, mas não completou...
54      Completou a Faculdade, mas não completou a Pós...
55      Completou o Ensino Médio, mas não completou a ...
56      Completou a Faculdade, mas não completou a Pós...
58      Completou a 4ł série/5ž ano, mas não completou...
60                              Completou a Pós-graduação
62      Completou o Ensino Médio, mas não completou a ...
                               ...
9942    Completou a 4ł série/5ž ano, mas não completou...
9943    Completou a Faculdade, mas não completou a Pós...
9944    Completou o Ensino Médio, mas não completou a ...
9947    Completou a Faculdade, mas não completou a Pós...
9948                            Completou a Pós-graduação
9949    Completou a 8ł série/9ž ano do Ensino Fundamen...
9950                            Completou a Pós-graduação
9951    Completou o Ensino Médio, mas não completou a ...
9952    Completou a 8ł série/9ž ano do Ensino Fundamen...
9954                            Completou a Pós-graduação
9955    Completou a 8ł série/9ž ano do Ensino Fundamen...
9956    Completou a Faculdade, mas não completou a Pós...
9958    Completou o Ensino Médio, mas não completou a ...
9959    Completou a Faculdade, mas não completou a Pós...
9961    Completou a Faculdade, mas não completou a Pós...
9962    Completou a 8ł série/9ž ano do Ensino Fundamen...
9965    Completou o Ensino Médio, mas não completou a ...
9968    Não completou a 4ł série/5ž ano do Ensino Fund...
9972                            Completou a Pós-graduação
9975    Completou a Faculdade, mas não completou a Pós...
9978    Completou a Faculdade, mas não completou a Pós...
9979                            Completou a Pós-graduação
9981    Completou a Faculdade, mas não completou a Pós...
9983    Completou a Faculdade, mas não completou a Pós...
9989    Completou a 8ł série/9ž ano do Ensino Fundamen...
9992    Completou o Ensino Médio, mas não completou a ...
9993    Completou o Ensino Médio, mas não completou a ...
9995    Completou o Ensino Médio, mas não completou a ...
9998    Completou a 4ł série/5ž ano, mas não completou...
9999    Completou a Faculdade, mas não completou a Pós...
Name: TX_RESP_Q019, Length: 4353, dtype: object
```

```
In [34]: coluna_mae.value_counts()
```

```
Out[34]: Completou o Ensino Médio, mas não completou a Faculdade
         Completou a Faculdade, mas não completou a Pós-graduação
         Completou a Pós-graduação
```

```
        Completou a 4ł série/5ž ano, mas não completou a 8ł série/9ž ano do Ensino Fundamental
        Não completou a 4ł série/5ž ano do Ensino Fundamental
        Completou a 8ł série/9ž ano do Ensino Fundamental, mas não completou o Ensino Médio
        Nunca estudou
        Name: TX_RESP_Q019, dtype: int64
```

```
In [35]: coluna_pai = saebmtlpgen_cor_mae_pai['TX_RESP_Q023']
```

```
In [36]: coluna_pai.value_counts()
```

```
Out[36]: G    1300
         E     865
         C     645
         B     578
         D     493
         F     370
         A     102
         Name: TX_RESP_Q023, dtype: int64
```

Agora faremos um dicionario de cor de pele declarada.

```
In [37]: coluna_cor = saebmtlpgen_cor_mae_pai['TX_RESP_Q002']
```

```
In [38]: #criando um dicionario
         cordicionario = {'A':'Branca',
         'B': 'Preta',
         'C': 'Parda',
         'D': 'Amarelo',
         'E': 'Indígena',
         'F': 'Não quero declarar'}
```

```
In [39]: #criando uma coluna para cor
         saebmtlpgen_cor_mae_pai['TX_RESP_Q002']=[cordicionario[resp] for resp in saebmtlpgen_
```

```
In [40]: coluna_cor.value_counts()
```

```
Out[40]: Parda                2374
         Branca               1056
         Preta                 378
         Não quero declarar    294
         Amarelo               170
         Indígena               81
         Name: TX_RESP_Q002, dtype: int64
```

Agora vamos calcular as médias da nota de matemática agrupadas pela cor de pele declarado pelo aluno e registrando os valores do maior para o menor.

```
In [41]: saebmtlpgen_cor_mae_pai.filter(items = ['PROFICIENCIA_MT_SAEB','TX_RESP_Q002']).groupl
```

```
Out[41]:                     PROFICIENCIA_MT_SAEB
         TX_RESP_Q002
         Branca                        271.835994
         Parda                         268.407447
         Amarelo                       263.625313
         Preta                         262.670047
         Indígena                      259.957820
         Não quero declarar            258.968000
```

Faremos o mesmo com o grau de escolaridade da mãe e do pai.

```
In [42]: saebmtlpgen_cor_mae_pai.filter(items = ['PROFICIENCIA_MT_SAEB','TX_RESP_Q019']).groupb
```

```
Out[42]:                                                    PROFICIENCIA_MT_SAEB
         TX_RESP_Q019
         Completou a Faculdade, mas não completou a Pós-...            279.423046
         Completou o Ensino Médio, mas não completou a F...            271.879170
         Completou a 4ł série/5ž ano, mas não completou ...            266.162406
         Completou a 8ł série/9ž ano do Ensino Fundament...            264.859554
         Não completou a 4ł série/5ž ano do Ensino Funda...            261.988626
         Completou a Pós-graduação                                     258.272838
         Nunca estudou                                                 244.584470
```

```
In [43]: saebmtlpgen_cor_mae_pai.filter(items = ['PROFICIENCIA_MT_SAEB','TX_RESP_Q023']).groupb
```

```
Out[43]:                PROFICIENCIA_MT_SAEB
         TX_RESP_Q023
         F                       280.227846
         E                       275.177644
         D                       270.130798
         C                       268.972096
         A                       265.480735
         B                       262.017585
         G                       260.505535
```

5 5. Teste de normalidade de Shapiro-Wilk

Vamos estudar se a variável numérica PROFICIENCIA_MT_SAEB tem distribuição normal. Para isso vamos criar um data set incluindo apenas esta variável.

Vamos importar o teste Shapiro-Wilk.

O teste está implementado no pacote SciPy, interpretamos o resultado por meio do p-valor como segue:

Se $p \leq \alpha = 0.05$, rejeitamos H_0 , a distribuição não é normal.

Se $p > \alpha$, falha ao rejeitar H_0 , a distribuição é normal.

```
In [44]: from scipy.stats import shapiro
```

```
In [45]: coluna_mat =  df["PROFICIENCIA_MT_SAEB"]
```

```
In [46]: stat, p = shapiro(coluna_mat)
         print('Estatistica=%.3f, p=%.3f' % (stat, p))
         # interpretação
         alpha = 0.05
         if p > alpha:
                 print('A amostra tem comportamento Gaussiano ou Normal (falha ao rejeitar H0)
         else:
                 print('A amostra não tem comportamento Gaussiano ou Normal (rejeitar H0)')
```

```
Estatistica=0.999, p=0.214
A amostra tem comportamento Gaussiano ou Normal (falha ao rejeitar H0)
```

O mesmo podemos fazer com a variável PROFICIENCIA_LP_SAEB.

```
In [47]: coluna_port =  df["PROFICIENCIA_LP_SAEB"]
```

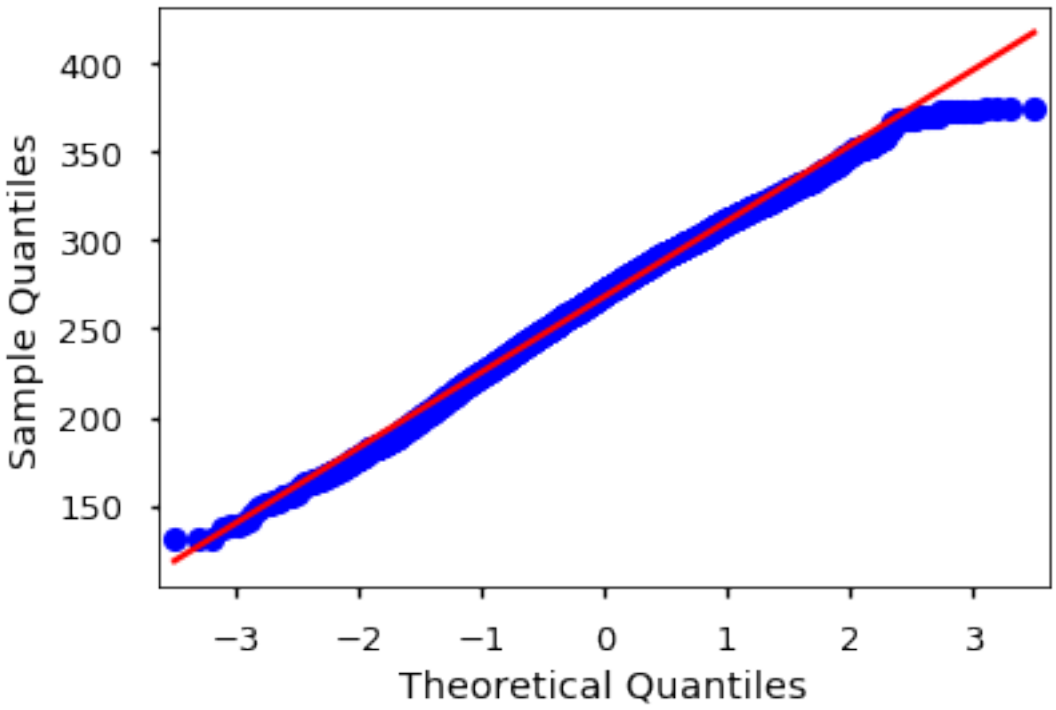
```
In [48]: stat, p = shapiro(coluna_port)
         print('Estatistica=%.3f, p=%.3f' % (stat, p))
         # interpretação
         alpha = 0.05
         if p > alpha:
                 print('A amostra tem comportamento Gaussiano ou Normal (falha ao rejeitar H0)
         else:
                 print('A amostra não tem comportamento Gaussiano ou Normal (rejeitar H0)')
```

```
Estatistica=0.994, p=0.000
A amostra não tem comportamento Gaussiano ou Normal (rejeitar H0)
```

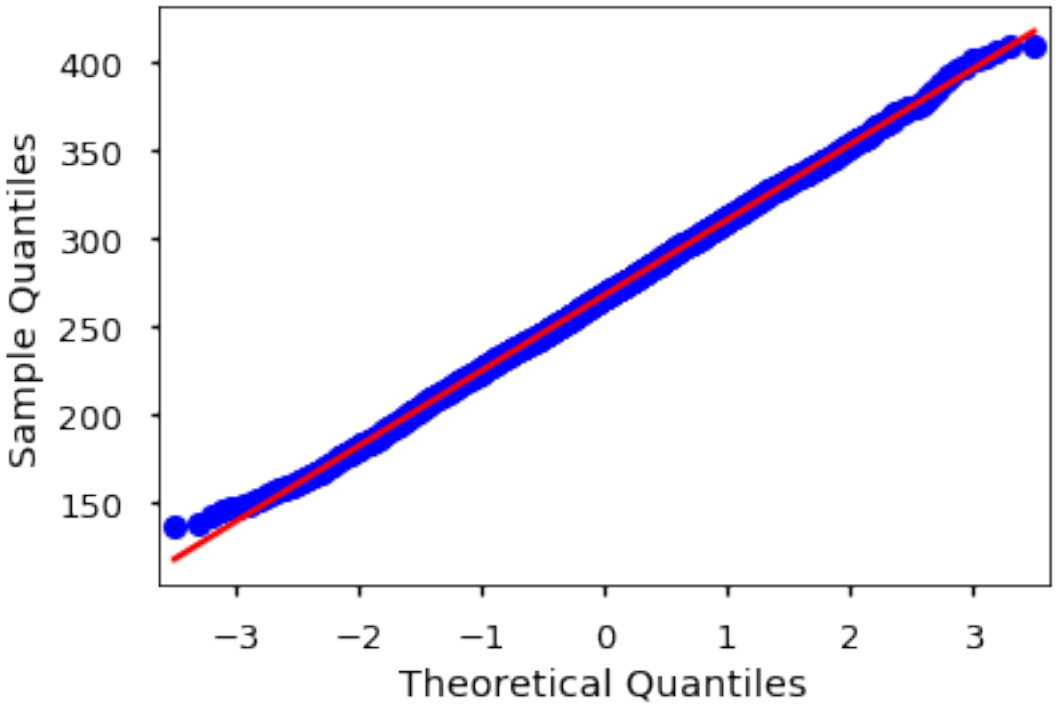
Vamos investigar visualmente a distribuição normal usando qqplot. Se os dados têm distribuição normal perfeita o gráfico mostrará por uma linha de pontos em um ângulo de 45 graus (bissetriz), do canto inferior esquerdo da plotagem para o canto superior direito. É traçada uma linha no gráfico para ajudar a tornar essa expectativa clara.

```
In [49]: from statsmodels.graphics.gofplots import qqplot
         from matplotlib import pyplot
```

```
In [50]: # q-q plot
         qqplot(coluna_port, line='s')
         pyplot.show()
```



In [51]:
```
# q-q plot
qqplot(coluna_mat, line='s')
pyplot.show()
```



6 Teste de Anderson-Darling

Teste que pode ser usado para avaliar se uma amostra é normal. É uma versão modificada de um teste não paramétrico chamado de Kolmogorov-Smirnov.

O teste está implementado no pacote scipy.

```
In [52]: # Teste Anderson-Darling
         from scipy.stats import anderson

In [53]: resultado = anderson(coluna_mat)
         print('Estatistica: %.3f' % resultado.statistic)
         p = 0
         for i in range(len(resultado.critical_values)):
                 sl, cv = resultado.significance_level[i], resultado.critical_values[i]
                 if resultado.statistic < resultado.critical_values[i]:
                         print('%.3f: %.3f, dados com distribuição normal (falha ao rejeitar  H
                 else:
                         print('%.3f: %.3f, dados sem distribuição normal (rejeitar H0)' % (sl
```

```
Estatistica: 0.411
15.000: 0.575, dados com distribuição normal (falha ao rejeitar  H0)
10.000: 0.655, dados com distribuição normal (falha ao rejeitar  H0)
5.000: 0.786, dados com distribuição normal (falha ao rejeitar  H0)
2.500: 0.917, dados com distribuição normal (falha ao rejeitar  H0)
1.000: 1.091, dados com distribuição normal (falha ao rejeitar  H0)
```

O mesmo teste pode ser aplicado ’a variável com as notas de língua portuguesa.

7 6. Tabela de dupla entrada

É inegável a importância da tabela de dupla entrada. Vamos ver alguns exemplos.

Vamos voltar ao data set saebmtlpgen_cor_mae_pai

```
In [57]: df2 = pd.DataFrame(df,columns=['PROFICIENCIA_LP_SAEB','PROFICIENCIA_MT_SAEB','TX_RESP_

In [58]: df2.head()
```

Out[58]:

	PROFICIENCIA_LP_SAEB	PROFICIENCIA_MT_SAEB	TX_RESP_Q001	TX_RESP_Q002	\
1	370.208184	360.521641	A	C	
2	226.748923	220.594975	B	C	
3	268.226347	249.367227	B	C	
4	274.938923	273.207429	B	A	
6	263.443252	245.078721	B	B	

```
  TX_RESP_Q019 TX_RESP_Q019
1            D            D
2            B            B
3            G            G
4            G            G
6            B            B
```

```
In [59]: pd.crosstab(df2.TX_RESP_Q001, df2.TX_RESP_Q002, margins=True)
```

```
Out[59]: TX_RESP_Q002     A    B     C    D   E    F   All
         TX_RESP_Q001
         A              467  201   995   74  35  177  1949
         B              589  177  1379   96  46  117  2404
         All           1056  378  2374  170  81  294  4353
```

Ainda no questionário socio-econômico tem as variáveis TX_RESP_Q048 (se já ficou reprovado) e TX_RESP_Q054 (se faz tarefa de matemática). Vamos fazer uma tabela de dupla entrada com estas variávies. As possíveis respostas para TX_RESP_Q048 são:

A: Não,
B: Sim, uma vez,e
C: Sim, duas vezes ou mais.
As possíveis respostas para TX_RESP_Q054 são:
A: Sempre ou quase sempre.
B: De vez em quando.
C: Nunca ou quase nunca.
D: O(A) professor(a) não passa dever de casa.

```
In [60]: #   cross table- reprovado e faz tarefa de mat
         import pandas as pd
         import numpy as np
```

```
In [61]: df3 = pd.DataFrame(df,columns=['TX_RESP_Q054','TX_RESP_Q048'])
```

```
In [62]: pd.crosstab(df3.TX_RESP_Q054, df3.TX_RESP_Q048,margins=True)
```

```
Out[62]: TX_RESP_Q048     A    B    C   All
         TX_RESP_Q054
         A             2151  490  151  2792
         B              914  291  116  1321
         C              107   36   36   179
         D               48   10    3    61
         All           3220  827  306  4353
```

Vemos que alunos que nunca ficaram reprovados e fazem sempre ou quase sempre suas tarefas de matemática são em maior número. Estas variáveis estão relacionadas? Esta é uma boa questão para pesquisa.

Vamos considerar apenas os alunos que estão nas respostas A e B. Isto é, no caso de TX_RESP_Q048 nunca ficaram reprovados (A) ou já ficaram reprovados uma vez (B). E no caso de TX_RESP_Q054 sempre fazem tarefas de matemática (A) ou fazem a tarefa de vez em quando (B). Vamos ver o ODDS nesta situação.

Antes vamos recordar este importante conceito.

8 ODDS Ratio = OR

Vamos apresentar este conceito por meio de um exemplo.

Suponha que dentre 10 homens, 7 entram em uma determinada universidade e 3 não. E que de 10 mulheres, 3 entram e 7 não.

Vamos ver o OR neste caso.

A probabilidade de um homem entrar na universidade é $p = \frac{7}{10} = 0.7$, enquanto que a $q = 1 - p = 1 - 0.7 = 0.3$ é a probabilidade de um homem não entrar na universidade.

Para as mulheres, a probabilidade de uma mulher entrar na universidade é $p = \frac{3}{10} = 0.3$, enquanto que a $q = 1 - p = 1 - 0.3 = 0.7$ é a probabilidade de uma mulher não entrar na universidade.

Agora vamos ver o odds (probabilidade ou chance) de cada um:

$$\text{odds(homem)} = \frac{0.7}{0.3} = 2.3333,$$

e

$$\text{odds(mulher)} = \frac{0.3}{0.7} = 0.4257.$$

Logo,

$$OR = \frac{2.3333}{0.4257} = 5.44$$

Assim, para um homem a probabilidade de ser admitido na universidade é 5.44 maior do que a probabilidade para uma mulher ser admitida.

Vamos considerar a situação em que os alunos fiquem apenas nas respostas A e B das questoões TX_RESP_Q054 e TX_RESP_Q048. Isto é, 2151 responderam que fazem as tarefas de matemática sempre e nunca ficaram reprovados, 490 fazem a tarefa sempre ficaramm reprovados uma vez, enquanto que 914 291 fazem tarefa de vez em quando e nunca ficaram reprovados, e 291 fazem tarefas de vez em quando e ja ficaram reprovados uma vez.

Vamos criar um array e calcular o ODDS Ratio.

```
In [63]: N = np.array([[2191,490], [914,291]])

In [64]: N

Out[64]: array([[2191,  490],
                [ 914,  291]])

In [65]: ODDS = (N[0,0]*N[1,1])/(N[0,1]*N[1,0])

In [66]: ODDS

Out[66]: 1.4236167552360113
```

Assim, um aluno que sempre faz suas tarefas sempre tem 1.42 vezes mais chance de não ter sido reprovado nunca comprado com um aluno que faz suas tarefas de vez em quando.

```
In [ ]:
```