main

November 16, 2022

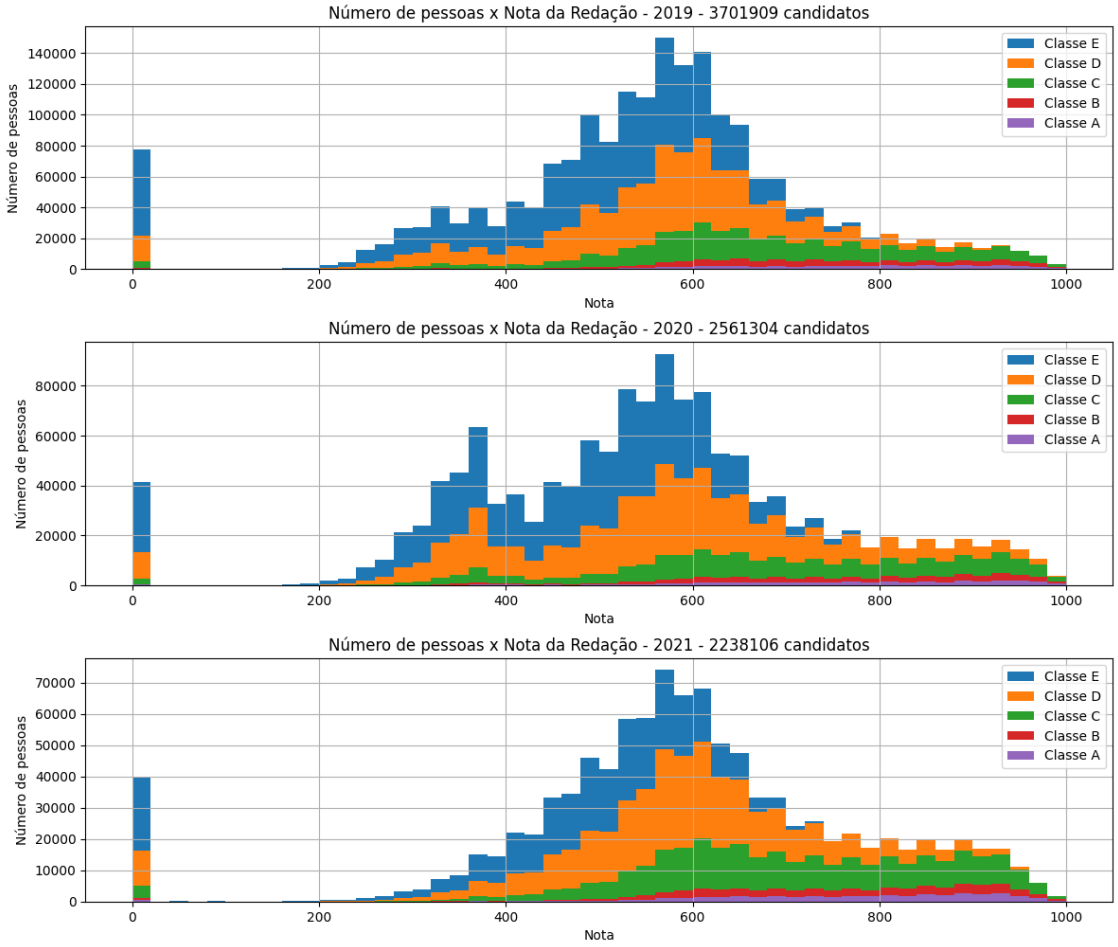
[1]:
```
import matplotlib.pyplot as plt
from dask import dataframe as dd
import pandas as pd
import numpy as np
import os

YEARS = ['2019', '2020', '2021']
PROVAS = ["NU_NOTA_CN","NU_NOTA_CH","NU_NOTA_MT","NU_NOTA_LC","NU_NOTA_REDACAO"]
PATH_2019 = os.path.join(os.getcwd(), 'dados', 'microdados_enem_2019', 'DADOS',␣
 ↪'MICRODADOS_ENEM_2019.csv')
PATH_2020 = os.path.join(os.getcwd(), 'dados', 'microdados_enem_2020', 'DADOS',␣
 ↪'MICRODADOS_ENEM_2020.csv')
PATH_2021 = os.path.join(os.getcwd(), 'dados', 'microdados_enem_2021', 'DADOS',␣
 ↪'MICRODADOS_ENEM_2021.csv')
OUTPUT = os.path.join(os.getcwd(), 'output')
ENCODING= "latin1"
SEP= ";"
COLS= [
        'NU_INSCRICAO',
        'Q006',
        'NU_ANO',
#         'TP_COR_RACA',
#         'SG_UF_ESC',
#         'TP_SEXO',
#         'TP_FAIXA_ETARIA',
#         'TP_ESCOLA',
#         'IN_TREINEIRO',
        'NU_NOTA_CN',
        'NU_NOTA_MT',
        'NU_NOTA_LC',
        'NU_NOTA_CH',
        'NU_NOTA_REDACAO',
]

if not os.path.isdir(OUTPUT):
    os.makedirs(OUTPUT)
```

[2]:
```
for i in YEARS:
    globals()[f'df_{i}'] = dd.read_csv(globals()[f'PATH_{i}'],␣
↪encoding=ENCODING, sep=SEP, dtype={'SG_UF_ESC': 'object'}, usecols=COLS)
    globals()[f'df_{i}'] = globals()[f'df_{i}'].compute()
    globals()[f'df_{i}'].dropna(inplace = True)

    for j in 'DEFG':
        globals()[f'df_{i}'].loc[globals()[f'df_{i}']['Q006'] == j, 'Q006'] =␣
↪'D'

    for j in 'ABC':
        globals()[f'df_{i}'].loc[globals()[f'df_{i}']['Q006'] == j, 'Q006'] =␣
↪'E'

    for j in 'HIJKLM':
        globals()[f'df_{i}'].loc[globals()[f'df_{i}']['Q006'] == j, 'Q006'] =␣
↪'C'

    for j in 'NOP':
        globals()[f'df_{i}'].loc[globals()[f'df_{i}']['Q006'] == j, 'Q006'] =␣
↪'B'

    globals()[f'df_{i}'].loc[globals()[f'df_{i}']['Q006'] == 'Q', 'Q006'] = 'A'
```

[3]:
```
for i in YEARS:
    with open(os.path.join(OUTPUT, f'describe{i}.txt'), 'w') as file:
        describe_provas = globals()[f'df_{i}'][PROVAS].describe(
                                        percentiles=[.15, .30, .45, .60, .75, .
↪90],
                                        include=['object', 'float64', 'int64']
                                    ).apply(lambda s: np.round(s, 2))
        output = f"""
    --------------------------------------------------------------------------
    DESCRIÇÃO DA AMOSTRA

    {describe_provas}
    --------------------------------------------------------------------------
    """
        file.write(output)
        for j in 'ABCDE':
            describe_provas_classe = globals()[f'df_{i}'].query(f"Q006 ==␣
↪'{j}'")[PROVAS].describe(
                                        percentiles=[.15, .25, .30, .45, .60, .
↪75, .90],
                                        include=['object', 'float64', 'int64']
                                    ).apply(lambda s: np.round(s, 2))
```

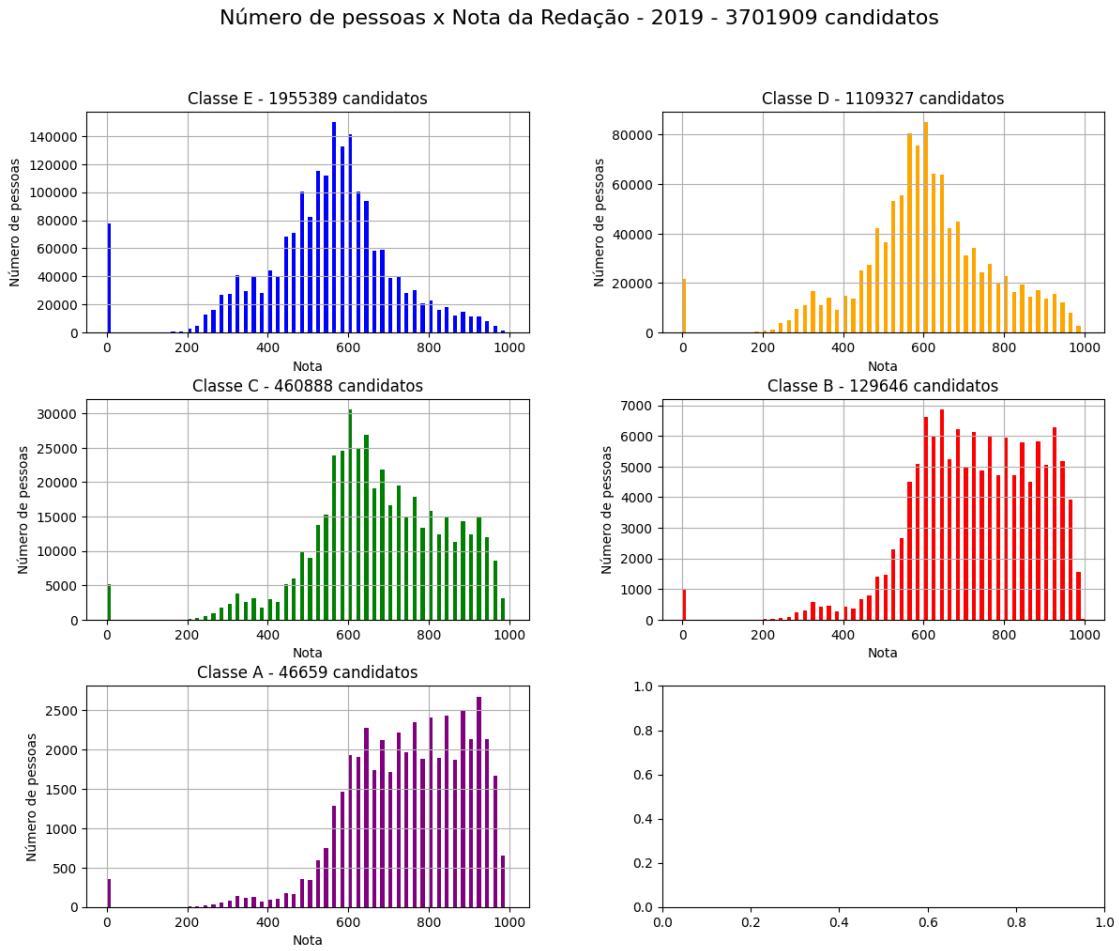
```
            output = f"""
    ------------------------------------------------------------------------
    DESCRIÇÃO DA AMOSTRA SOBRE A CLASSE {j}

    {describe_provas_classe}
    ------------------------------------------------------------------------
    """
            file.write(output)
        file.close()
```

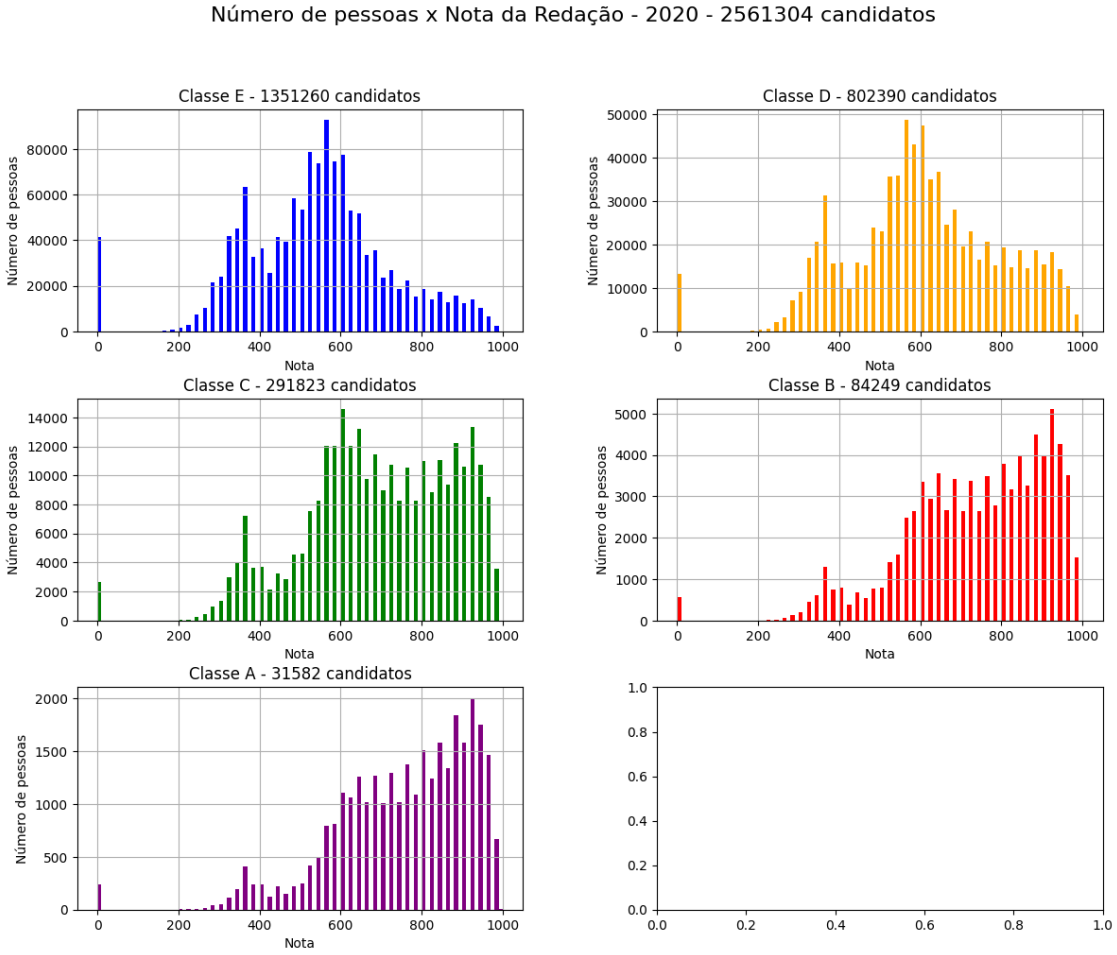
[4]:
```
fig, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(14, 12))

for index, value in enumerate(YEARS):
    globals()[f'df_{value}'].query("Q006 == 'E'")[['NU_NOTA_REDACAO']].hist(
        bins=50, ax=globals()[f'ax{index + 1}'], label="Classe E")
    globals()[f'df_{value}'].query("Q006 == 'D'")[['NU_NOTA_REDACAO']].hist(
        bins=50, ax=globals()[f'ax{index + 1}'], label="Classe D")
    globals()[f'df_{value}'].query("Q006 == 'C'")[['NU_NOTA_REDACAO']].hist(
        bins=50, ax=globals()[f'ax{index + 1}'], label="Classe C")
    globals()[f'df_{value}'].query("Q006 == 'B'")[['NU_NOTA_REDACAO']].hist(
        bins=50, ax=globals()[f'ax{index + 1}'], label="Classe B")
    globals()[f'df_{value}'].query("Q006 == 'A'")[['NU_NOTA_REDACAO']].hist(
        bins=50, ax=globals()[f'ax{index + 1}'], label="Classe A")

    globals()[f'ax{index + 1}'].set_title(
        f'Número de pessoas x Nota da Redação - {value} - {n_candidatos}␣
 ↪candidatos')
    globals()[f'ax{index + 1}'].set_ylabel('Número de pessoas')
    globals()[f'ax{index + 1}'].set_xlabel('Nota')
    globals()[f'ax{index + 1}'].legend()

fig.savefig(os.path.join(OUTPUT, 'all_classes_by_year.jpg'))
```

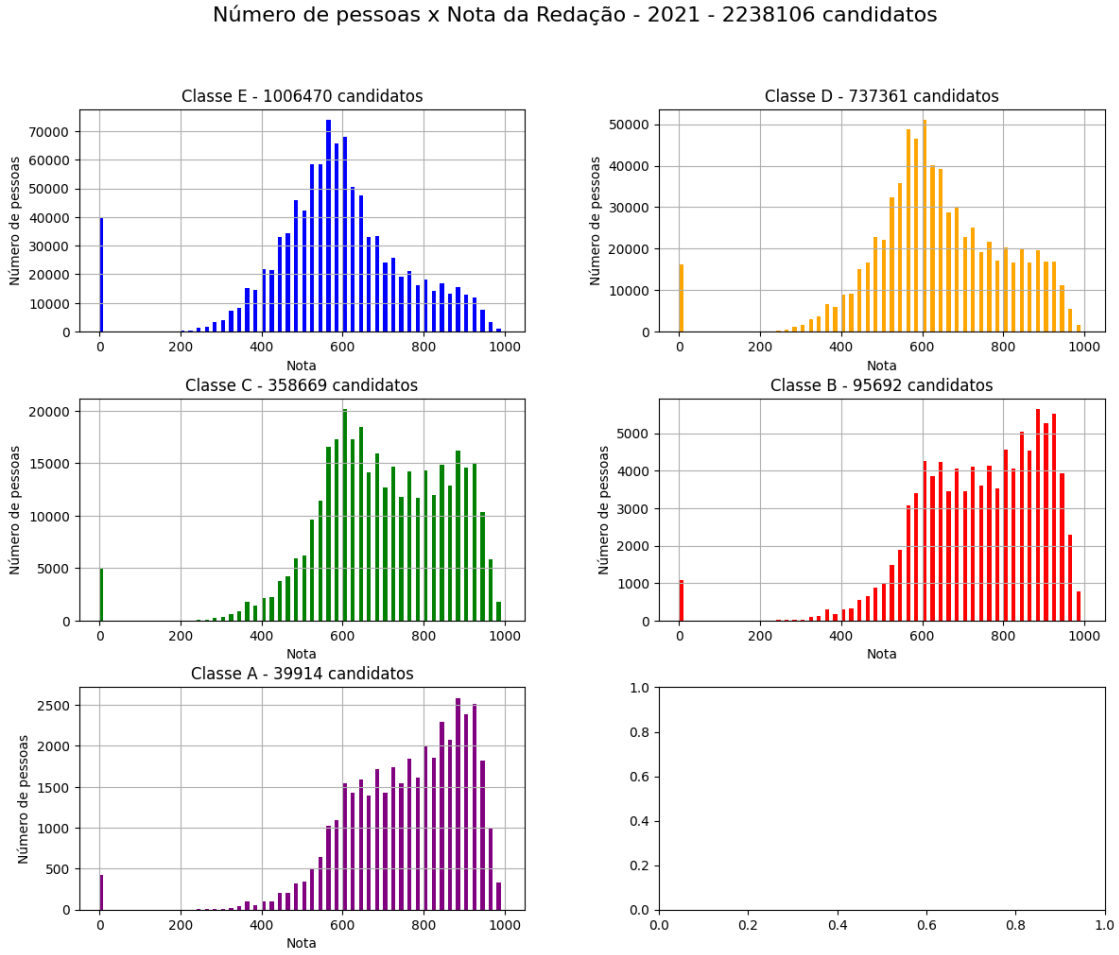


[5]:
```
for i in YEARS:
    fig, [[ax1, ax2], [ax3, ax4], [ax5, _]] = plt.subplots(3, 2,
                                                           figsize=(14, 11),
 ↪#constrained_layout=True
                                                           )
    for index, value in enumerate(['E', 'D', 'C', 'B', 'A']):

        color = ""
        if value == 'E':
            color = "blue"
        elif value == 'D':
            color = 'orange'
        elif value == 'C':
            color  ='green'
        elif value == 'B':
            color = 'red'
```

```
        elif value == 'A':
            color = 'purple'

        fig.suptitle(f"Número de pessoas x Nota da Redação - {i} -␣
↪{globals()[f'df_{i}'].shape[0]} candidatos", fontsize=16)
        globals()[f'df_{i}'].query(f"Q006 == '{value}'")[['NU_NOTA_REDACAO']].
↪hist(bins=100, ax=globals()[f'ax{index + 1}'], label=f"Classe {value}",␣
↪color=color)
        globals()[f'ax{index + 1}'].set_title("Classe {0} - {1} candidatos".
↪format(value, globals()[f'df_{i}'].query(f"Q006 == '{value}'").shape[0]))
        globals()[f'ax{index + 1}'].set_ylabel('Número de pessoas')
        globals()[f'ax{index + 1}'].set_xlabel('Nota')

    fig.savefig(os.path.join(OUTPUT, f'histogram_classes_{i}.jpg'))
```

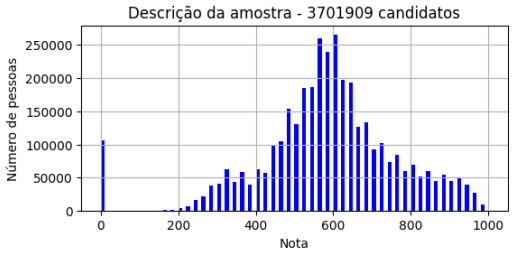


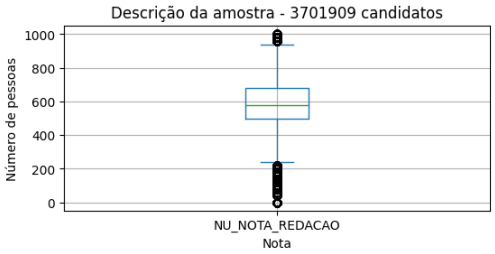

Número de pessoas x Nota da Redação - 2020 - 2561304 candidatos
Classe E - 1351260 candidatos
Classe D - 802390 candidatos
Classe C - 291823 candidatos
Classe B - 84249 candidatos
Classe A - 31582 candidatos
Número de pessoas
Nota

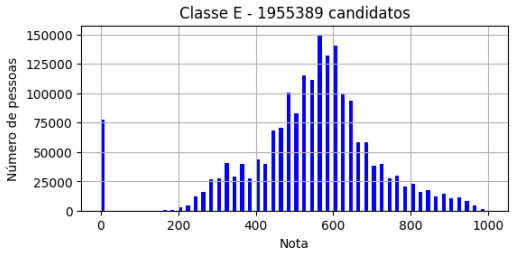


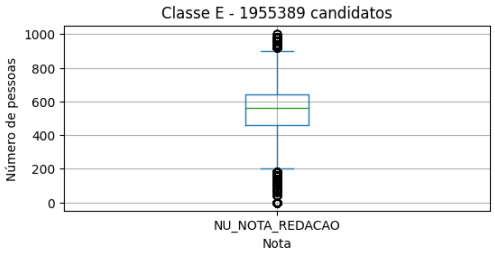
[6]:
```
from utils import utils

utils.make_analisys_image(df_2019, OUTPUT)
utils.make_analisys_image(df_2020, OUTPUT)
utils.make_analisys_image(df_2021, OUTPUT)
```

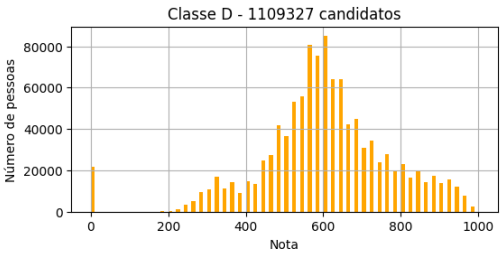
Número de pessoas x Nota da Redação - 2019 - 3701909 candidatos

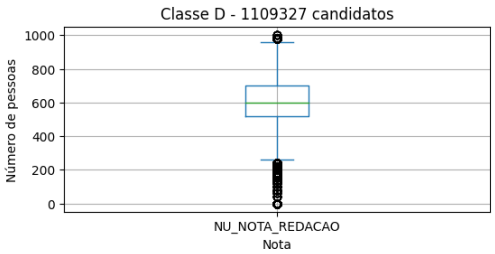


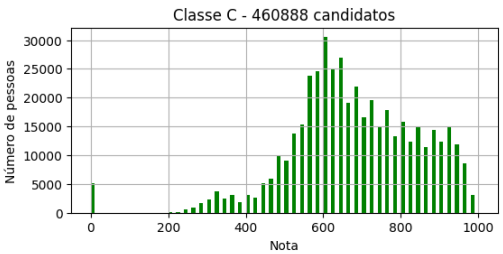


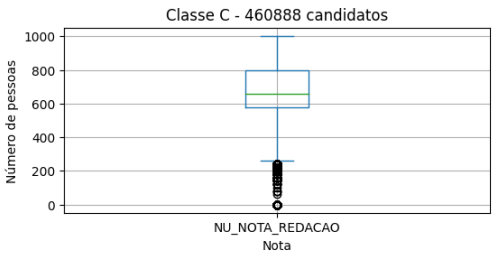


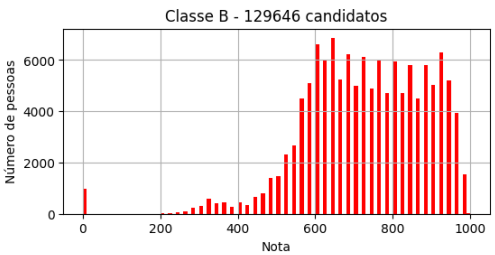


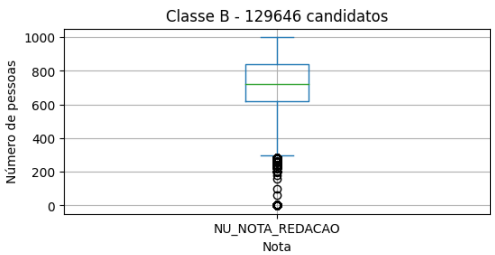


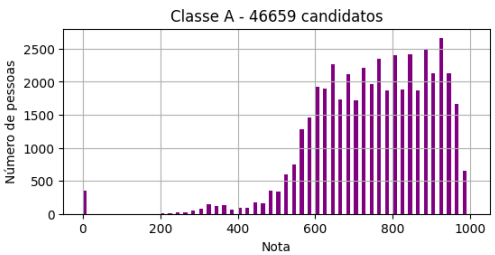


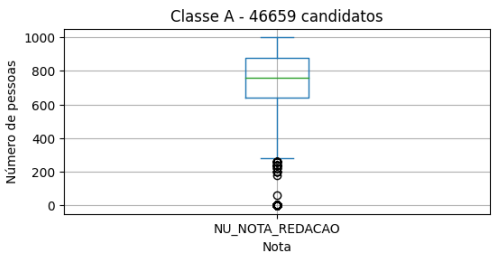




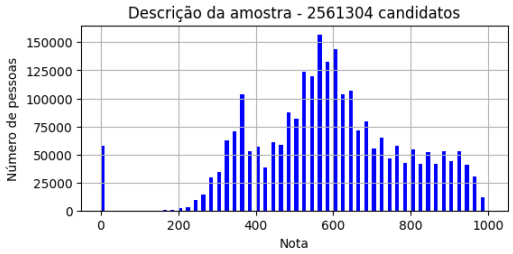


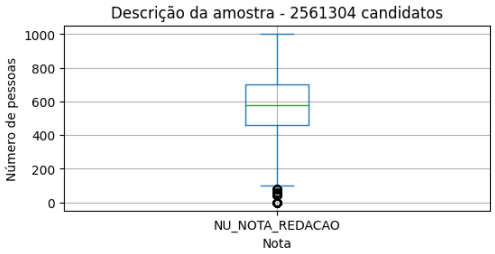


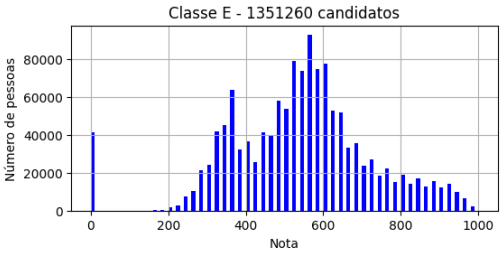


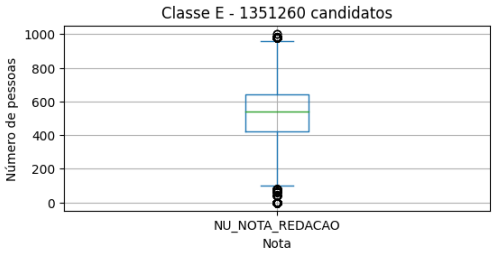


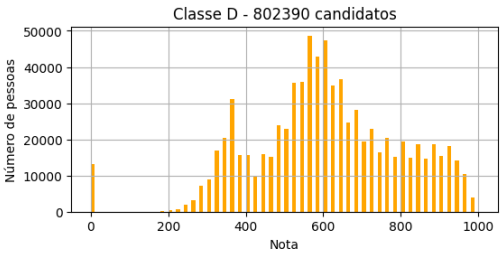
Número de pessoas x Nota da Redação - 2020 - 2561304 candidatos

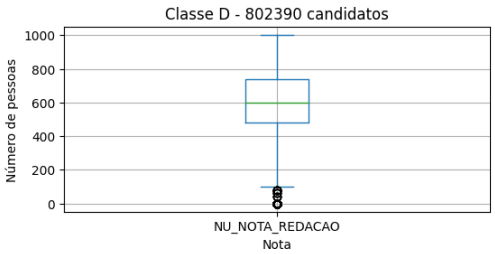


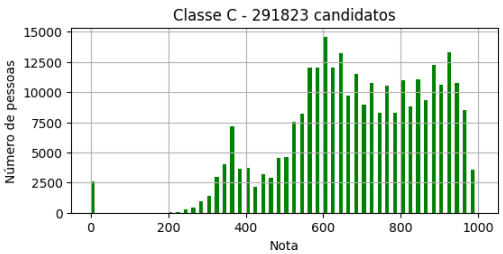


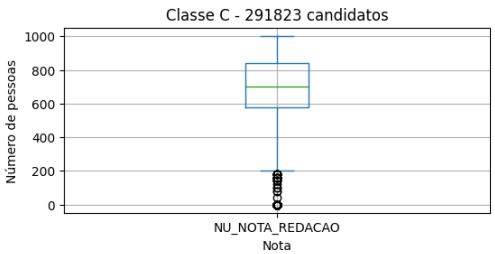


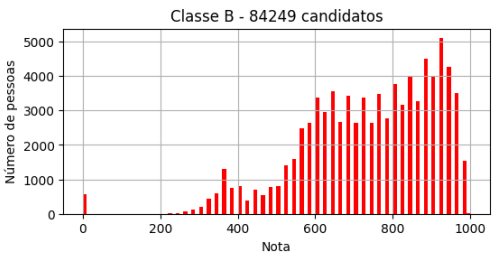


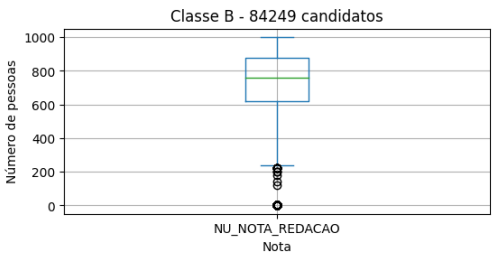


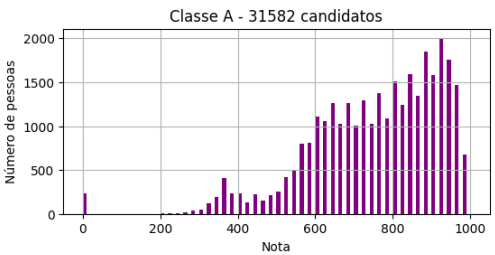


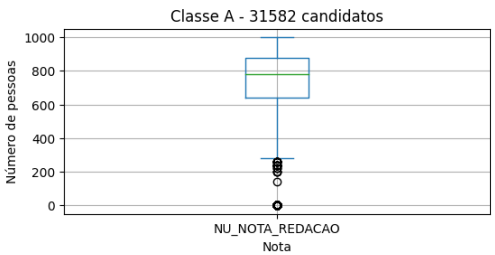




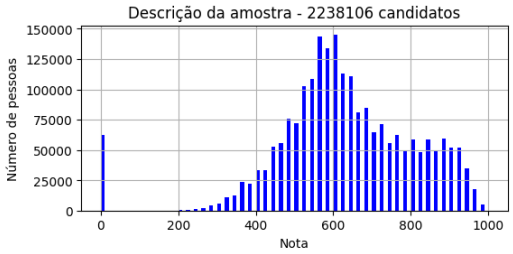


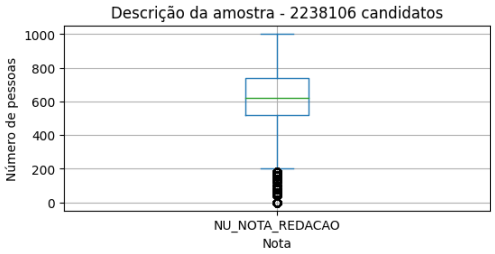


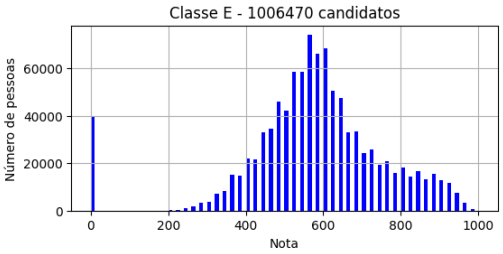


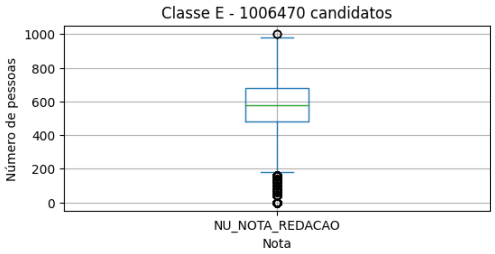


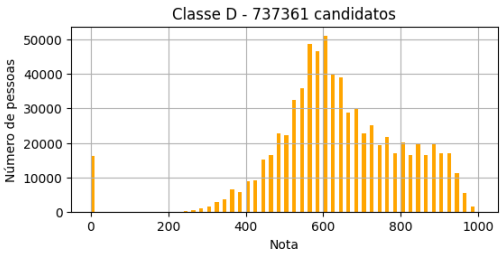
Número de pessoas x Nota da Redação - 2021 - 2238106 candidatos

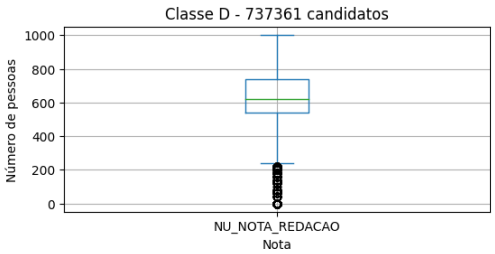


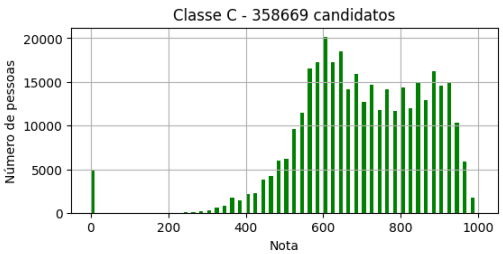


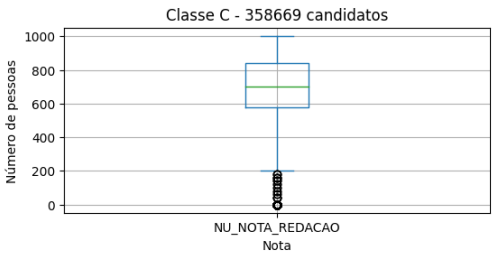


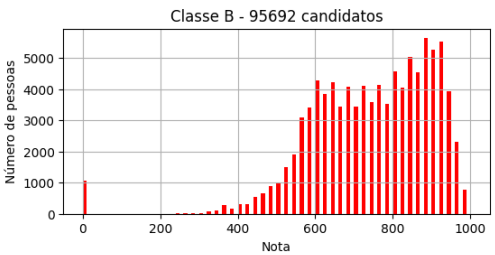


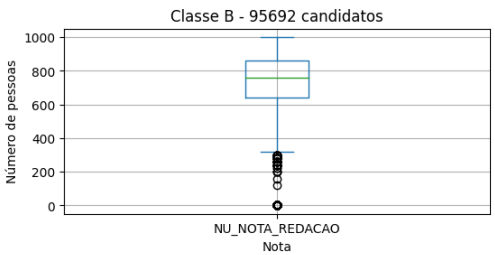


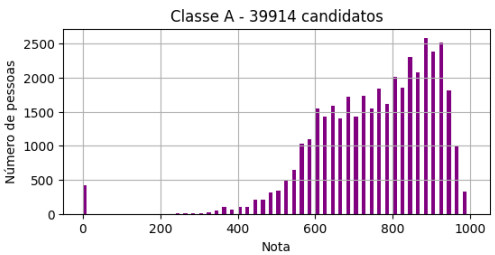


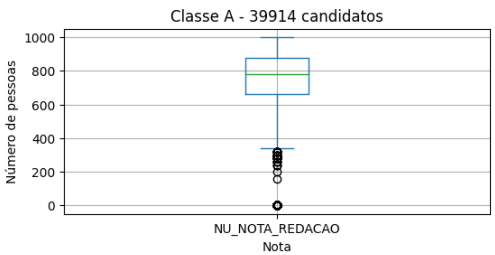




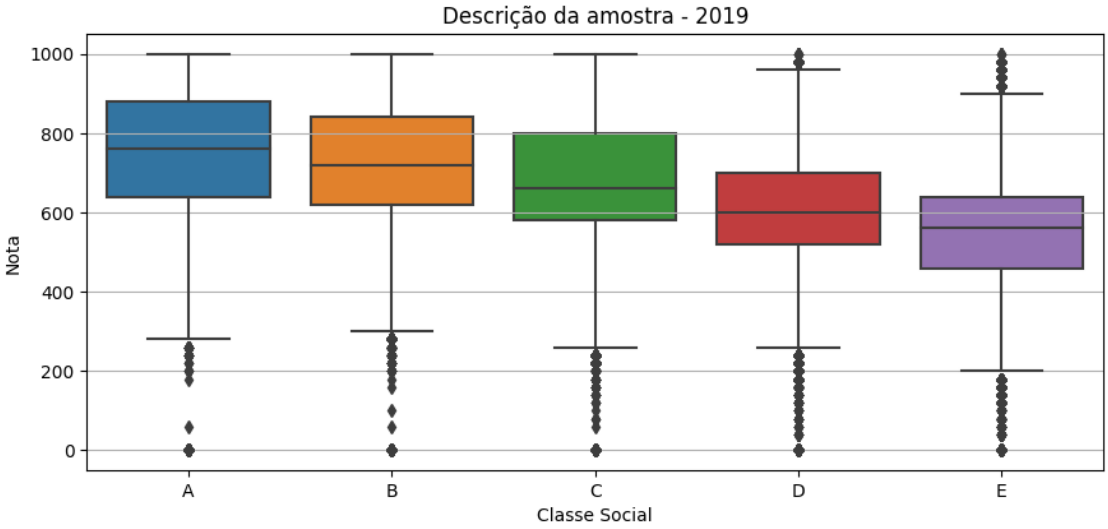


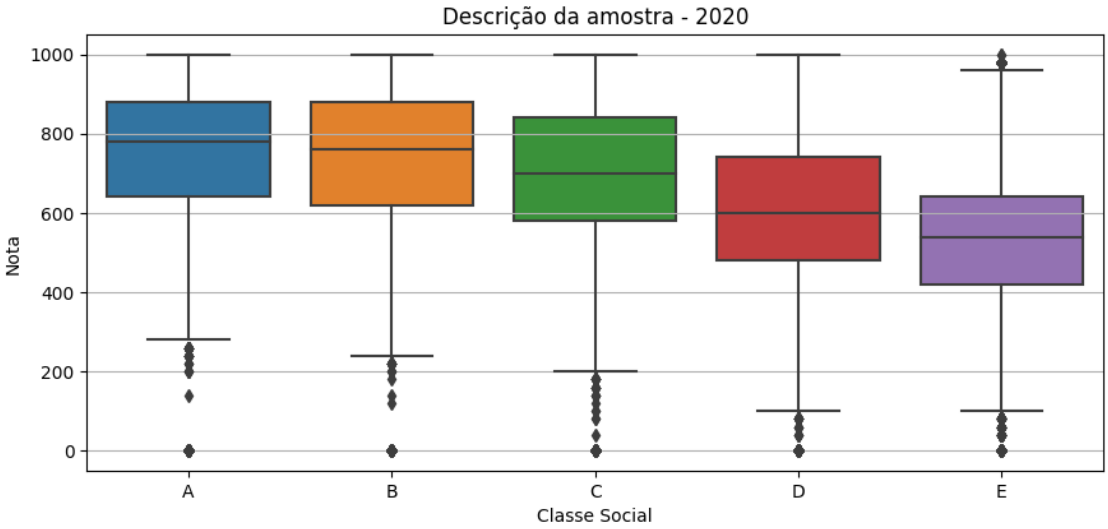


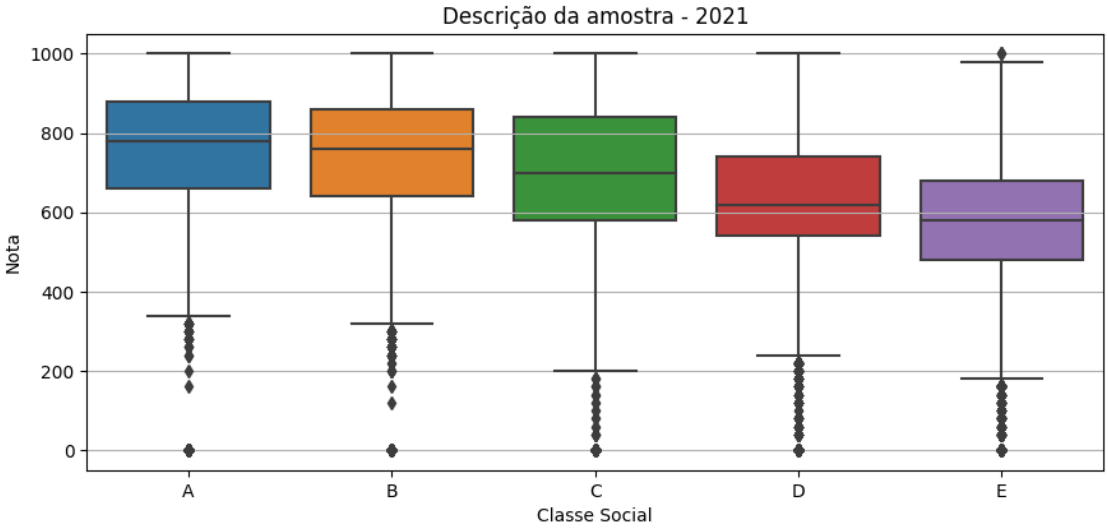




[76]:
```
import seaborn as sns

fig, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(10, 15))

sns.boxplot(x='Q006', y='NU_NOTA_REDACAO', data=df_2019,
            ax=ax1, order=['A', 'B', 'C', 'D', 'E']).set(xlabel='Classe␣
 ↪Social', ylabel='Nota')
ax1.set_title("Descrição da amostra - {0}".format(df_2019[['NU_ANO']].
 ↪iloc[0][0]))
ax1.set_ylabel('Nota')
ax1.set_xlabel('Classe Social')
ax1.yaxis.grid(True)

sns.boxplot(x='Q006', y='NU_NOTA_REDACAO', data=df_2020,
            ax=ax2, order=['A', 'B', 'C', 'D', 'E']).set(xlabel='Classe␣
 ↪Social', ylabel='Nota')
ax2.set_title("Descrição da amostra - {0}".format(df_2020[['NU_ANO']].
 ↪iloc[0][0]))
ax2.set_ylabel('Nota')
ax2.set_xlabel('Classe Social')
ax2.yaxis.grid(True)

sns.boxplot(x='Q006', y='NU_NOTA_REDACAO', data=df_2021,
            ax=ax3, order=['A', 'B', 'C', 'D', 'E']).set(xlabel='Classe␣
 ↪Social', ylabel='Nota')
ax3.set_title("Descrição da amostra - {0}".format(df_2021[['NU_ANO']].
 ↪iloc[0][0]))
ax3.set_ylabel('Nota')
ax3.set_xlabel('Classe Social')
ax3.yaxis.grid(True)

fig.tight_layout(pad=5.0)
fig.savefig(os.path.join(OUTPUT, 'all_classes_by_year_boxplot.jpg'))
```


Descrição da amostra - 2019
Nota
1000
800
600
400
200
0
A
B
C
D
E
Classe Social


Descrição da amostra - 2020
Nota
1000
800
600
400
200
0
A
B
C
D
E
Classe Social


Descrição da amostra - 2021
Nota
1000
800
600
400
200
0
A
B
C
D
E
Classe Social

[95]:
```
df_2019['NU_MEDIA_GERAL'] = (df_2019['NU_NOTA_MT'] + df_2019['NU_NOTA_LC'] +␣
↪df_2019['NU_NOTA_CH']
                             + df_2019['NU_NOTA_CN'] +␣
↪df_2019['NU_NOTA_REDACAO'])/5

df_2019.head()
```

[95]:

	NU_INSCRICAO	NU_ANO	NU_NOTA_CN	NU_NOTA_CH	NU_NOTA_LC	NU_NOTA_MT	NU_NOTA_REDACAO	Q006	NU_MEDIA_GERAL
3	190001199383	2019	483.8	503.6	537.3	392.0	460.0	D	475.34
4	190001237802	2019	513.6	575.5	570.7	677.0	860.0	D	639.36
5	190001782198	2019	563.7	644.9	564.2	675.3	800.0	D	649.62
6	190001421548	2019	484.6	488.4	507.2	594.7	600.0	E	534.98
9	190001592266	2019	543.9	548.1	502.5	480.7	400.0	D	495.04