

Descrição Geral: Este script tem como objetivo carregar um arquivo CSV contendo dados do ENEM, realizar a substituição de diversos valores nas colunas presente na base, renomear diversas colunas para melhorar a legibilidade e interpretação dos dados e liberar memória não utilizada. Após essas transformações, o script salva os dados tratados em um novo arquivo CSV.

Passos Realizados no Script:

1. Importação das Bibliotecas:

O script utiliza duas bibliotecas principais:

pandas: Para manipulação de dados em DataFrames.

gc: Para liberar memória não utilizada através da coleta de lixo.

2. Carregamento do Arquivo CSV:

O arquivo CSV contendo os dados do ENEM é carregado em um DataFrame chamado `df_enem`. O arquivo é localizado no caminho especificado, e são definidos o separador e a codificação do arquivo.

3. Filtro de Valores em branco ou nulos

Essas linhas de código preenchem valores em branco em diversas colunas do DataFrame `df_enem` com 'SEM RESPOSTA', '0' e afins, além de converter esses valores para o tipo apropriado de dados, garantindo consistência nos dados.

- `df_enem['TP_ENSINO'] = df_enem['TP_ENSINO'].fillna(0).astype(int)`: Preenche valores em branco na coluna 'TP_ENSINO' com zero e converte para o tipo inteiro.
- `df_enem['CO_MUNICIPIO_ESC'] = df_enem['CO_MUNICIPIO_ESC'].fillna(0).astype(int)`: Preenche valores em branco na coluna 'CO_MUNICIPIO_ESC' com zero e converte para o tipo inteiro.
- `df_enem['NO_MUNICIPIO_ESC'] = df_enem['NO_MUNICIPIO_ESC'].fillna('Não Informado').astype(str)`: Preenche valores em branco na coluna 'NO_MUNICIPIO_ESC' com 'Não Informado' e converte para o tipo string.
- `df_enem['CO_UF_ESC'] = df_enem['CO_UF_ESC'].fillna(0).astype(int)`: Preenche valores em branco na coluna 'CO_UF_ESC' com zero e converte para o tipo inteiro.
- `df_enem['SG_UF_ESC'] = df_enem['SG_UF_ESC'].fillna('Não Informado').astype(str)`: Preenche valores em branco na coluna 'SG_UF_ESC' com 'Não Informado' e converte para o tipo string.
- `df_enem['TP_DEPENDENCIA_ADM_ESC'] = df_enem['TP_DEPENDENCIA_ADM_ESC'].fillna(0).astype(int)`: Preenche valores

em branco na coluna 'TP_DEPENDENCIA_ADM_ESC' com zero e converte para o tipo inteiro.

- `df_enem['TP_LOCALIZACAO_ESC'] = df_enem['TP_LOCALIZACAO_ESC'].fillna(0).astype(int):` Preenche valores em branco na coluna 'TP_LOCALIZACAO_ESC' com zero e converte para o tipo inteiro.
- `df_enem['TP_SIT_FUNC_ESC'] = df_enem['TP_SIT_FUNC_ESC'].fillna(0).astype(int):` Preenche valores em branco na coluna 'TP_SIT_FUNC_ESC' com zero e converte para o tipo inteiro.
- `df_enem['CO_PROVA_CN'] = df_enem['CO_PROVA_CN'].fillna(0).astype(int):` Preenche valores em branco na coluna 'CO_PROVA_CN' com zero e converte para o tipo inteiro.
- `df_enem['CO_PROVA_CH'] = df_enem['CO_PROVA_CH'].fillna(0).astype(int):` Preenche valores em branco na coluna 'CO_PROVA_CH' com zero e converte para o tipo inteiro.
- `df_enem['CO_PROVA_LC'] = df_enem['CO_PROVA_LC'].fillna(0).astype(int):` Preenche valores em branco na coluna 'CO_PROVA_LC' com zero e converte para o tipo inteiro.
- `df_enem['CO_PROVA_MT'] = df_enem['CO_PROVA_MT'].fillna(0).astype(int):` Preenche valores em branco na coluna 'CO_PROVA_MT' com zero e converte para o tipo inteiro.
- `df_enem['NU_NOTA_CN'] = df_enem['NU_NOTA_CN'].fillna(0).astype(int):` Preenche valores em branco na coluna 'NU_NOTA_CN' com zero e converte para o tipo inteiro.
- `df_enem['NU_NOTA_CH'] = df_enem['NU_NOTA_CH'].fillna(0).astype(int):` Preenche valores em branco na coluna 'NU_NOTA_CH' com zero e converte para o tipo inteiro.
- `df_enem['NU_NOTA_LC'] = df_enem['NU_NOTA_LC'].fillna(0).astype(int):` Preenche valores em branco na coluna 'NU_NOTA_LC' com zero e converte para o tipo inteiro.
- `df_enem['NU_NOTA_MT'] = df_enem['NU_NOTA_MT'].fillna(0).astype(int):` Preenche valores em branco na coluna 'NU_NOTA_MT' com zero e converte para o tipo inteiro.
- `df_enem['NU_NOTA_COMP2'] = df_enem['NU_NOTA_COMP2'].fillna(0).astype(int):` Preenche valores em branco na coluna 'NU_NOTA_COMP2' com zero e converte para o tipo inteiro.
- `df_enem['NU_NOTA_COMP4'] = df_enem['NU_NOTA_COMP4'].fillna(0).astype(int):` Preenche valores em branco na coluna 'NU_NOTA_COMP4' com zero e converte para o tipo inteiro.
- `df_enem['NU_NOTA_COMP5'] = df_enem['NU_NOTA_COMP5'].fillna(0).astype(int):` Preenche valores em branco na coluna 'NU_NOTA_COMP5' com zero e converte para o tipo inteiro.

- `df_enem['NU_NOTA_REDACAO'] = df_enem['NU_NOTA_REDACAO'].fillna(0).astype(int):` Preenche valores em branco na coluna 'NU_NOTA_REDACAO' com zero e converte para o tipo inteiro.
- `df_enem['Q001'] = df_enem['Q001'].fillna('SEM RESPOSTA').astype(str):` Preenche valores em branco na coluna 'Q001' com 'SEM RESPOSTA' e converte para o tipo string.
- `df_enem['Q002'] = df_enem['Q002'].fillna('SEM RESPOSTA').astype(str):` Preenche valores em branco na coluna 'Q002' com 'SEM RESPOSTA' e converte para o tipo string.
- `df_enem['Q003'] = df_enem['Q003'].fillna('SEM RESPOSTA').astype(str):` Preenche valores em branco na coluna 'Q003' com 'SEM RESPOSTA' e converte para o tipo string.
- `df_enem['Q004'] = df_enem['Q004'].fillna('SEM RESPOSTA').astype(str):` Preenche valores em branco na coluna 'Q004' com 'SEM RESPOSTA' e converte para o tipo string.
- `df_enem['Q005'] = df_enem['Q005'].fillna('SEM RESPOSTA').astype(str):` Preenche valores em branco na coluna 'Q005' com 'SEM RESPOSTA' e converte para o tipo string.
- `df_enem['Q006'] = df_enem['Q006'].fillna('SEM RESPOSTA').astype(str):` Preenche valores em branco na coluna 'Q006' com 'SEM RESPOSTA' e converte para o tipo string.
- `df_enem['Q007'] = df_enem['Q007'].fillna('SEM RESPOSTA').astype(str):` Preenche valores em branco na coluna 'Q007' com 'SEM RESPOSTA' e converte para o tipo string.
- `df_enem['Q008'] = df_enem['Q008'].fillna('SEM RESPOSTA').astype(str):` Preenche valores em branco na coluna 'Q008' com 'SEM RESPOSTA' e converte para o tipo string.
- `df_enem['Q009'] = df_enem['Q009'].fillna('SEM RESPOSTA').astype(str):` Preenche valores em branco na coluna 'Q009' com 'SEM RESPOSTA' e converte para o tipo string.
- `df_enem['Q010'] = df_enem['Q010'].fillna('SEM RESPOSTA').astype(str):` Preenche valores em branco na coluna 'Q010' com 'SEM RESPOSTA' e converte para o tipo string.
- `df_enem['Q011'] = df_enem['Q011'].fillna('SEM RESPOSTA').astype(str):` Preenche valores em branco na coluna 'Q011' com 'SEM RESPOSTA' e converte para o tipo string.
- `df_enem['Q012'] = df_enem['Q012'].fillna('SEM RESPOSTA').astype(str):` Preenche valores em branco na coluna 'Q012' com 'SEM RESPOSTA' e converte para o tipo string.
- `df_enem['Q013'] = df_enem['Q013'].fillna('SEM RESPOSTA').astype(str):` Preenche valores em branco na coluna 'Q013' com 'SEM RESPOSTA' e converte para o tipo string.

- `df_enem['Q014'] = df_enem['Q014'].fillna('SEM RESPOSTA').astype(str)`: Preenche valores em branco na coluna 'Q014' com 'SEM RESPOSTA' e converte para o tipo string.
- `df_enem['Q015'] = df_enem['Q015'].fillna('SEM RESPOSTA').astype(str)`: Preenche valores em branco na coluna 'Q015' com 'SEM RESPOSTA' e converte para o tipo string.
- `df_enem['Q016'] = df_enem['Q016'].fillna('SEM RESPOSTA').astype(str)`: Preenche valores em branco na coluna 'Q016' com 'SEM RESPOSTA' e converte para o tipo string.
- `df_enem['Q017'] = df_enem['Q017'].fillna('SEM RESPOSTA').astype(str)`: Preenche valores em branco na coluna 'Q017' com 'SEM RESPOSTA' e converte para o tipo string.
- `df_enem['Q018'] = df_enem['Q018'].fillna('SEM RESPOSTA').astype(str)`: Preenche valores em branco na coluna 'Q018' com 'SEM RESPOSTA' e converte para o tipo string.
- `df_enem['Q019'] = df_enem['Q019'].fillna('SEM RESPOSTA').astype(str)`: Preenche valores em branco na coluna 'Q019' com 'SEM RESPOSTA' e converte para o tipo string.
- `df_enem['Q020'] = df_enem['Q020'].fillna('SEM RESPOSTA').astype(str)`: Preenche valores em branco na coluna 'Q020' com 'SEM RESPOSTA' e converte para o tipo string.
- `df_enem['Q021'] = df_enem['Q021'].fillna('SEM RESPOSTA').astype(str)`: Preenche valores em branco na coluna 'Q021' com 'SEM RESPOSTA' e converte para o tipo string.
- `df_enem['Q022'] = df_enem['Q022'].fillna('SEM RESPOSTA').astype(str)`: Preenche valores em branco na coluna 'Q022' com 'SEM RESPOSTA' e converte para o tipo string.
- `df_enem['Q023'] = df_enem['Q023'].fillna('SEM RESPOSTA').astype(str)`: Preenche valores em branco na coluna 'Q023' com 'SEM RESPOSTA' e converte para o tipo string.
- `df_enem['Q024'] = df_enem['Q024'].fillna('SEM RESPOSTA').astype(str)`: Preenche valores em branco na coluna 'Q024' com 'SEM RESPOSTA' e converte para o tipo string.
- `df_enem['Q025'] = df_enem['Q025'].fillna('SEM RESPOSTA').astype(str)`: Preenche valores em branco na coluna 'Q025' com 'SEM RESPOSTA' e converte para o tipo string.

4. Substituição de Valores nas colunas do DataFrame:

Coluna 'TP_PRESENCA_CN'

- `df_enem['TP_PRESENCA_CN'] = df_enem['TP_PRESENCA_CN'].replace(0, 'Ausente').astype(str)`: Substitui os valores 0 na coluna 'TP_PRESENCA_CN' por 'Ausente' e converte para o tipo string.
- `df_enem['TP_PRESENCA_CN'] = df_enem['TP_PRESENCA_CN'].replace('1', 'Presente').astype(str)`: Substitui os valores '1' na coluna 'TP_PRESENCA_CN' por 'Presente' e converte para o tipo string.
- `df_enem['TP_PRESENCA_CN'] = df_enem['TP_PRESENCA_CN'].replace('2', 'Eliminado').astype(str)`: Substitui os valores '2' na coluna 'TP_PRESENCA_CN' por 'Eliminado' e converte para o tipo string.

Coluna 'TP_PRESENCA_CH'

- `df_enem['TP_PRESENCA_CH'] = df_enem['TP_PRESENCA_CH'].replace(0, 'Ausente').astype(str)`: Substitui os valores 0 na coluna 'TP_PRESENCA_CH' por 'Ausente' e converte para o tipo string.
- `df_enem['TP_PRESENCA_CH'] = df_enem['TP_PRESENCA_CH'].replace('1', 'Presente').astype(str)`: Substitui os valores '1' na coluna 'TP_PRESENCA_CH' por 'Presente' e converte para o tipo string.
- `df_enem['TP_PRESENCA_CH'] = df_enem['TP_PRESENCA_CH'].replace('2', 'Eliminado').astype(str)`: Substitui os valores '2' na coluna 'TP_PRESENCA_CH' por 'Eliminado' e converte para o tipo string.

Coluna 'TP_PRESENCA_LC'

- `df_enem['TP_PRESENCA_LC'] = df_enem['TP_PRESENCA_LC'].replace(0, 'Ausente').astype(str)`: Substitui os valores 0 na coluna 'TP_PRESENCA_LC' por 'Ausente' e converte para o tipo string.
- `df_enem['TP_PRESENCA_LC'] = df_enem['TP_PRESENCA_LC'].replace('1', 'Presente').astype(str)`: Substitui os valores '1' na coluna 'TP_PRESENCA_LC' por 'Presente' e converte para o tipo string.
- `df_enem['TP_PRESENCA_LC'] = df_enem['TP_PRESENCA_LC'].replace('2', 'Eliminado').astype(str)`: Substitui os valores '2' na coluna 'TP_PRESENCA_LC' por 'Eliminado' e converte para o tipo string.

Coluna 'TP_PRESENCA_LC'

- `df_enem['TP_PRESENCA_MT'] = df_enem['TP_PRESENCA_MT'].replace(0, 'Ausente').astype(str)`: Substitui os valores 0 na coluna 'TP_PRESENCA_MT' por 'Ausente' e converte para o tipo string.
- `df_enem['TP_PRESENCA_MT'] = df_enem['TP_PRESENCA_MT'].replace('1', 'Presente').astype(str)`: Substitui os valores '1' na coluna 'TP_PRESENCA_MT' por 'Presente' e converte para o tipo string.
- `df_enem['TP_PRESENCA_MT'] = df_enem['TP_PRESENCA_MT'].replace('2', 'Eliminado').astype(str)`: Substitui os valores '2' na coluna 'TP_PRESENCA_MT' por 'Eliminado' e converte para o tipo string.

Coluna 'TP_COR_RACA'

- `df_enem['TP_COR_RACA'] = df_enem['TP_COR_RACA'].replace(0, 'Não Informado').astype(str)`: Substitui os valores 0 na coluna 'TP_COR_RACA' por 'Não Informado' e converte para o tipo string.
- `df_enem['TP_COR_RACA'] = df_enem['TP_COR_RACA'].replace('1', 'Branca').astype(str)`: Substitui os valores '1' na coluna 'TP_COR_RACA' por 'Branca' e converte para o tipo string.
- `df_enem['TP_COR_RACA'] = df_enem['TP_COR_RACA'].replace('2', 'Preta').astype(str)`: Substitui os valores '2' na coluna 'TP_COR_RACA' por 'Preta' e converte para o tipo string.
- `df_enem['TP_COR_RACA'] = df_enem['TP_COR_RACA'].replace('3', 'Parda').astype(str)`: Substitui os valores '3' na coluna 'TP_COR_RACA' por 'Parda' e converte para o tipo string.
- `df_enem['TP_COR_RACA'] = df_enem['TP_COR_RACA'].replace('4', 'Amarela').astype(str)`: Substitui os valores '4' na coluna 'TP_COR_RACA' por 'Amarela' e converte para o tipo string.
- `df_enem['TP_COR_RACA'] = df_enem['TP_COR_RACA'].replace('5', 'Indígena').astype(str)`: Substitui os valores '5' na coluna 'TP_COR_RACA' por 'Indígena' e converte para o tipo string.

Coluna 'TP_ESTADO_CIVIL'

- `df_enem['TP_ESTADO_CIVIL'] = df_enem['TP_ESTADO_CIVIL'].replace(0, 'Não Informado').astype(str)`: Substitui os valores 0 na coluna 'TP_ESTADO_CIVIL' por 'Não Informado' e converte para o tipo string.
- `df_enem['TP_ESTADO_CIVIL'] = df_enem['TP_ESTADO_CIVIL'].replace('1', 'Solteiro (a)').astype(str)`: Substitui os valores '1' na coluna 'TP_ESTADO_CIVIL' por 'Solteiro (a)' e converte para o tipo string.
- `df_enem['TP_ESTADO_CIVIL'] = df_enem['TP_ESTADO_CIVIL'].replace('2', 'Casado(a)/Mora com companheiro (a)').astype(str)`: Substitui os valores '2' na coluna 'TP_ESTADO_CIVIL' por 'Casado(a)/Mora com companheiro (a)' e converte para o tipo string.

- `df_enem['TP_ESTADO_CIVIL'] = df_enem['TP_ESTADO_CIVIL'].replace('3', 'Divorciado(a)/Desquitado(a)/Separado(a')).astype(str)`: Substitui os valores '3' na coluna 'TP_ESTADO_CIVIL' por 'Divorciado(a)/Desquitado(a)/Separado(a)' e converte para o tipo string.
- `df_enem['TP_ESTADO_CIVIL'] = df_enem['TP_ESTADO_CIVIL'].replace('4', 'Viúvo(a')).astype(str)`: Substitui os valores '4' na coluna 'TP_ESTADO_CIVIL' por 'Viúvo(a)' e converte para o tipo string.

Coluna 'TP_FAIXA_ETARIA'

- `df_enem['TP_FAIXA_ETARIA'] = df_enem['TP_FAIXA_ETARIA'].replace(1, 'Menor de 17 anos').astype(str)`: Substitui o valor 1 na coluna 'TP_FAIXA_ETARIA' por 'Menor de 17 anos' e converte para o tipo string.
- `df_enem['TP_FAIXA_ETARIA'] = df_enem['TP_FAIXA_ETARIA'].replace('2', '17 anos').astype(str)`: Substitui o valor '2' na coluna 'TP_FAIXA_ETARIA' por '17 anos' e converte para o tipo string.
- `df_enem['TP_FAIXA_ETARIA'] = df_enem['TP_FAIXA_ETARIA'].replace('3', '18 anos').astype(str)`: Substitui o valor '3' na coluna 'TP_FAIXA_ETARIA' por '18 anos' e converte para o tipo string.
- `df_enem['TP_FAIXA_ETARIA'] = df_enem['TP_FAIXA_ETARIA'].replace('4', '19 anos').astype(str)`: Substitui o valor '4' na coluna 'TP_FAIXA_ETARIA' por '19 anos' e converte para o tipo string.
- `df_enem['TP_FAIXA_ETARIA'] = df_enem['TP_FAIXA_ETARIA'].replace('5', '20 anos').astype(str)`: Substitui o valor '5' na coluna 'TP_FAIXA_ETARIA' por '20 anos' e converte para o tipo string.
- `df_enem['TP_FAIXA_ETARIA'] = df_enem['TP_FAIXA_ETARIA'].replace('6', '21 anos').astype(str)`: Substitui o valor '6' na coluna 'TP_FAIXA_ETARIA' por '21 anos' e converte para o tipo string.
- `df_enem['TP_FAIXA_ETARIA'] = df_enem['TP_FAIXA_ETARIA'].replace('7', '22 anos').astype(str)`: Substitui o valor '7' na coluna 'TP_FAIXA_ETARIA' por '22 anos' e converte para o tipo string.
- `df_enem['TP_FAIXA_ETARIA'] = df_enem['TP_FAIXA_ETARIA'].replace('8', '23 anos').astype(str)`: Substitui o valor '8' na coluna 'TP_FAIXA_ETARIA' por '23 anos' e converte para o tipo string.
- `df_enem['TP_FAIXA_ETARIA'] = df_enem['TP_FAIXA_ETARIA'].replace('9', '24 anos').astype(str)`: Substitui o valor '9' na coluna 'TP_FAIXA_ETARIA' por '24 anos' e converte para o tipo string.
- `df_enem['TP_FAIXA_ETARIA'] = df_enem['TP_FAIXA_ETARIA'].replace('10', '25 anos').astype(str)`: Substitui o valor '10' na coluna 'TP_FAIXA_ETARIA' por '25 anos' e converte para o tipo string.
- `df_enem['TP_FAIXA_ETARIA'] = df_enem['TP_FAIXA_ETARIA'].replace('11', 'Entre 26 e 30 anos').astype(str)`: Substitui o valor '11' na coluna 'TP_FAIXA_ETARIA' por 'Entre 26 e 30 anos' e converte para o tipo string.

- `df_enem['TP_FAIXA_ETARIA'] = df_enem['TP_FAIXA_ETARIA'].replace('12', 'Entre 31 e 35 anos').astype(str):` Substitui o valor '12' na coluna 'TP_FAIXA_ETARIA' por 'Entre 31 e 35 anos' e converte para o tipo string.
- `df_enem['TP_FAIXA_ETARIA'] = df_enem['TP_FAIXA_ETARIA'].replace('13', 'Entre 36 e 40 anos').astype(str):` Substitui o valor '13' na coluna 'TP_FAIXA_ETARIA' por 'Entre 36 e 40 anos' e converte para o tipo string.
- `df_enem['TP_FAIXA_ETARIA'] = df_enem['TP_FAIXA_ETARIA'].replace('14', 'Entre 41 e 45 anos').astype(str):` Substitui o valor '14' na coluna 'TP_FAIXA_ETARIA' por 'Entre 41 e 45 anos' e converte para o tipo string.
- `df_enem['TP_FAIXA_ETARIA'] = df_enem['TP_FAIXA_ETARIA'].replace('15', 'Entre 46 e 50 anos').astype(str):` Substitui o valor '15' na coluna 'TP_FAIXA_ETARIA' por 'Entre 46 e 50 anos' e converte para o tipo string.
- `df_enem['TP_FAIXA_ETARIA'] = df_enem['TP_FAIXA_ETARIA'].replace('16', 'Entre 51 e 55 anos').astype(str):` Substitui o valor '16' na coluna 'TP_FAIXA_ETARIA' por 'Entre 51 e 55 anos' e converte para o tipo string.
- `df_enem['TP_FAIXA_ETARIA'] = df_enem['TP_FAIXA_ETARIA'].replace('17', 'Entre 56 e 60 anos').astype(str):` Substitui o valor '17' na coluna 'TP_FAIXA_ETARIA' por 'Entre 56 e 60 anos' e converte para o tipo string.
- `df_enem['TP_FAIXA_ETARIA'] = df_enem['TP_FAIXA_ETARIA'].replace('18', 'Entre 61 e 65 anos').astype(str):` Substitui o valor '18' na coluna 'TP_FAIXA_ETARIA' por 'Entre 61 e 65 anos' e converte para o tipo string.
- `df_enem['TP_FAIXA_ETARIA'] = df_enem['TP_FAIXA_ETARIA'].replace('19', 'Entre 66 e 70 anos').astype(str):` Substitui o valor '19' na coluna 'TP_FAIXA_ETARIA' por 'Entre 66 e 70 anos' e converte para o tipo string.
- `df_enem['TP_FAIXA_ETARIA'] = df_enem['TP_FAIXA_ETARIA'].replace('20', 'Maior de 70 anos').astype(str):` Substitui o valor '20' na coluna 'TP_FAIXA_ETARIA' por 'Maior de 70 anos' e converte para o tipo string.

Coluna 'TP_ST_CONCLUSAO'

- `df_enem['TP_ST_CONCLUSAO'] = df_enem['TP_ST_CONCLUSAO'].replace(1, 'Concluído').astype(str):` Substitui o valor 1 na coluna 'TP_ST_CONCLUSAO' por 'Concluído' e converte para o tipo string.
- `df_enem['TP_ST_CONCLUSAO'] = df_enem['TP_ST_CONCLUSAO'].replace('2', 'Concluindo 2020').astype(str):` Substitui o valor '2' na coluna 'TP_ST_CONCLUSAO' por 'Concluindo 2020' e converte para o tipo string.
- `df_enem['TP_ST_CONCLUSAO'] = df_enem['TP_ST_CONCLUSAO'].replace('3', 'Conclusão após 2020').astype(str):` Substitui o valor '3' na coluna 'TP_ST_CONCLUSAO' por 'Conclusão após 2020' e converte para o tipo string.
- `df_enem['TP_ST_CONCLUSAO'] = df_enem['TP_ST_CONCLUSAO'].replace('4', 'S/Conclusão S/Curso').astype(str):` Substitui o valor '4' na coluna 'TP_ST_CONCLUSAO' por 'S/Conclusão S/Curso' e converte para o tipo string.

Coluna 'TP_ESCOLA'

- `df_enem['TP_ESCOLA'] = df_enem['TP_ESCOLA'].replace(1, 'Sem Informação').astype(str)`: Substitui o valor 1 na coluna 'TP_ESCOLA' por 'Sem Informação' e converte para o tipo string.
- `df_enem['TP_ESCOLA'] = df_enem['TP_ESCOLA'].replace(2, 'Pública').astype(str)`: Substitui o valor '2' na coluna 'TP_ESCOLA' por 'Pública' e converte para o tipo string.
- `df_enem['TP_ESCOLA'] = df_enem['TP_ESCOLA'].replace(3, 'Privada').astype(str)`: Substitui o valor '3' na coluna 'TP_ESCOLA' por 'Privada' e converte para o tipo string.
- `df_enem['TP_ESCOLA'] = df_enem['TP_ESCOLA'].replace(4, 'Exterior').astype(str)`: Substitui o valor '4' na coluna 'TP_ESCOLA' por 'Exterior' e converte para o tipo string.

Coluna 'TP_LOCALIZACAO_ESC'

- `df_enem['TP_LOCALIZACAO_ESC'] = df_enem['TP_LOCALIZACAO_ESC'].replace(1, 'Urbana').astype(str)`: Substitui o valor 1 na coluna 'TP_LOCALIZACAO_ESC' por 'Urbana' e converte para o tipo string.
- `df_enem['TP_LOCALIZACAO_ESC'] = df_enem['TP_LOCALIZACAO_ESC'].replace(2, 'Rural').astype(str)`: Substitui o valor '2' na coluna 'TP_LOCALIZACAO_ESC' por 'Rural' e converte para o tipo string.

Coluna 'TP_NACIONALIDADE'

- `df_enem['TP_NACIONALIDADE'] = df_enem['TP_NACIONALIDADE'].replace(0, 'Não Informado').astype(str)`: Substitui o valor 0 na coluna 'TP_NACIONALIDADE' por 'Não Informado' e converte para o tipo string.
- `df_enem['TP_NACIONALIDADE'] = df_enem['TP_NACIONALIDADE'].replace('1', 'Brasileiro(a)').astype(str)`: Substitui o valor '1' na coluna 'TP_NACIONALIDADE' por 'Brasileiro(a)' e converte para o tipo string.
- `df_enem['TP_NACIONALIDADE'] = df_enem['TP_NACIONALIDADE'].replace(2, 'Brasileiro(a) Naturalizado(a)').astype(str)`: Substitui o valor '2' na coluna 'TP_NACIONALIDADE' por 'Brasileiro(a) Naturalizado(a)' e converte para o tipo string.
- `df_enem['TP_NACIONALIDADE'] = df_enem['TP_NACIONALIDADE'].replace(3, 'Estrangeiro(a)').astype(str)`: Substitui o valor '3' na coluna 'TP_NACIONALIDADE' por 'Estrangeiro(a)' e converte para o tipo string.
- `df_enem['TP_NACIONALIDADE'] = df_enem['TP_NACIONALIDADE'].replace(4, 'Brasileiro(a) Nato(a), nascido(a) no`

exterior').astype(str): Substitui o valor '4' na coluna 'TP_NACIONALIDADE' por 'Brasileiro(a) Nato(a), nascido(a) no exterior' e converte para o tipo string.

Coluna 'TP_STATUS_REDACAO'

- `df_enem['TP_STATUS_REDACAO'] = df_enem['TP_STATUS_REDACAO'].replace(1, 'Sem problemas').astype(str):` Substitui o valor 1 na coluna 'TP_STATUS_REDACAO' por 'Sem problemas' e converte para o tipo string.
- `df_enem['TP_STATUS_REDACAO'] = df_enem['TP_STATUS_REDACAO'].replace(2, 'Anulada').astype(str):` Substitui o valor 2 na coluna 'TP_STATUS_REDACAO' por 'Anulada' e converte para o tipo string.
- `df_enem['TP_STATUS_REDACAO'] = df_enem['TP_STATUS_REDACAO'].replace(3, 'Cópia Texto Motivador').astype(str):` Substitui o valor 3 na coluna 'TP_STATUS_REDACAO' por 'Cópia Texto Motivador' e converte para o tipo string.
- `df_enem['TP_STATUS_REDACAO'] = df_enem['TP_STATUS_REDACAO'].replace(4, 'Em Branco').astype(str):` Substitui o valor 4 na coluna 'TP_STATUS_REDACAO' por 'Em Branco' e converte para o tipo string.
- `df_enem['TP_STATUS_REDACAO'] = df_enem['TP_STATUS_REDACAO'].replace(6, 'Fuga ao tema').astype(str):` Substitui o valor 6 na coluna 'TP_STATUS_REDACAO' por 'Fuga ao tema' e converte para o tipo string.
- `df_enem['TP_STATUS_REDACAO'] = df_enem['TP_STATUS_REDACAO'].replace(7, 'Não atendimento ao tipo textual').astype(str):` Substitui o valor 7 na coluna 'TP_STATUS_REDACAO' por 'Não atendimento ao tipo textual' e converte para o tipo string.
- `df_enem['TP_STATUS_REDACAO'] = df_enem['TP_STATUS_REDACAO'].replace(8, 'Texto insuficiente').astype(str):` Substitui o valor 8 na coluna 'TP_STATUS_REDACAO' por 'Texto insuficiente' e converte para o tipo string.
- `df_enem['TP_STATUS_REDACAO'] = df_enem['TP_STATUS_REDACAO'].replace(9, 'Parte desconectada').astype(str):` Substitui o valor 9 na coluna 'TP_STATUS_REDACAO' por 'Parte desconectada' e converte para o tipo string.
- `df_enem['TP_STATUS_REDACAO'] = df_enem['TP_STATUS_REDACAO'].fillna('N/A').astype(str):` Substitui os valores nulos na coluna 'TP_STATUS_REDACAO' por 'N/A' e converte para o tipo string.

Coluna 'TP_LINGUA'

- `df_enem['TP_LINGUA'] = df_enem['TP_LINGUA'].replace(0, 'Inglês').astype(str):` Substitui o valor 0 na coluna 'TP_LINGUA' por 'Inglês' e converte para o tipo string.
- `df_enem['TP_LINGUA'] = df_enem['TP_LINGUA'].replace('1', 'Espanhol').astype(str):` Substitui o valor '1' na coluna 'TP_LINGUA' por 'Espanhol' e converte para o tipo string.
- `df_enem['TREINEIRO'] = df_enem['TREINEIRO'].replace(0, 'NÃO').astype(str):` Substitui o valor 0 na coluna 'TREINEIRO' por 'NÃO' e converte para o tipo string.
- `df_enem['TREINEIRO'] = df_enem['TREINEIRO'].replace(1, 'SIM').astype(str):` Substitui o valor 1 na coluna 'TREINEIRO' por 'SIM' e converte para o tipo string.

Transformação no Power Query

- Além de fazer todas as transformações no Python, também foi feita uma alteração na coluna de sexo no Power Query, trocando 'F' por "Feminino" e "M" por Masculino.

5. Renomeação de Colunas:

Várias colunas no DataFrame são renomeadas para termos mais descritivos e intuitivos, além de se adequar ao padrão de um banco de dados. Foram selecionadas algumas colunas para o projeto, como por exemplo:

- TP_SEXO é renomeada para SEXO.
- TP_FAIXA é renomeada para FAIXA_ETARIA.
- TP_COR_RACA é renomeada para ETNIA.
- TP_ST_CONCLUSAO é renomeada para ST_CONCLUSAO.
- TP_ESCOLA é renomeada para TIPO_ESCOLA.
- IN_TREINEIRO é renomeada para TREINEIRO.
- CO_MUNICIPIO_PROVA é renomeada para COD_MUNICIPIO.
- NO_MUNICIPIO_PROVA é renomeada para NOME_MUNICIPIO.
- CO_UF_PROVA é renomeada para COD_UF.
- SG_UF_PROVA é renomeada para UF.
- TP_PRESENCA_CN é renomeada para PRESENCA_CN.
- TP_PRESENCA_CH é renomeada para PRESENCA_CH.
- TP_PRESENCA_LC é renomeada para PRESENCA_LC.
- TP_PRESENCA_MT é renomeada para PRESENCA_MT.
- CO_PROVA_CN é renomeada para COD_PCN.
- CO_PROVA_CH é renomeada para COD_PCH.
- CO_PROVA_LC é renomeada para COD_PLC.
- CO_PROVA_MT é renomeada para COD_PMT.
- NU_NOTA_CN é renomeada para NOTA_PCN.
- NU_NOTA_CH é renomeada para NOTA_PCH.
- NU_NOTA_LC é renomeada para NOTA_PLC.
- NU_NOTA_MT é renomeada para NOTA_PMT.
- TP_LINGUA é renomeada para LINGUA_ESCOLHIDA.

- TP_ESTADO_CIVIL é renomeada para ESTADO_CIVIL.
- TP_NACIONALIDADE é renomeada para NACIONALIDADE.
- TP_ANO_CONCLUIU é renomeada para ANO_CONCLUSAO.
- TP_LOCALIZACAO_ESC é renomeada para LOCALIZACAO_ESC.
- TP_SIT_FUNC_ESC é renomeada para FUNCIONAMENTO_ESC.
- NU_NOTA_COMP1 é renomeada para NOTA_COMP1.
- NU_NOTA_COMP2 é renomeada para NOTA_COMP2.
- NU_NOTA_COMP3 é renomeada para NOTA_COMP3.
- NU_NOTA_COMP4 é renomeada para NOTA_COMP4.
- NU_NOTA_COMP5 é renomeada para NOTA_COMP5.
- NU_NOTA_REDACAO é renomeada para NOTA_REDACAO.

6. Liberação de Memória:

A biblioteca gc é utilizada para liberar a memória que não está mais sendo usada, melhorando a eficiência do script.

-> gc.collect()

Este trecho de código gc.collect() é usado para forçar a coleta de lixo no Python. A coleta de lixo é o processo de liberação de memória que não está mais em uso pelo programa, o que pode ajudar a otimizar o uso de memória durante a execução do código. O método collect() do módulo gc (garbage collector) do Python é chamado para iniciar explicitamente o processo de coleta de lixo.

7. Criação do Arquivo CSV Tratado:

Após todas as transformações, os dados tratados são salvos em um novo arquivo CSV com um nome específico e no mesmo caminho do arquivo original.

```
df_enem.to_csv('C:/Users/peixe/Desktop/Desafio Técnico Mesha/Base Enem
Tratada2.csv', index=False, sep=';', encoding='UTF-8')
```

Onde:

- df_enem.to_csv(...): Este é um método do DataFrame df_enem que exporta os dados para um arquivo CSV.
- 'C:/Users/peixe/Desktop/Desafio Técnico Mesha/Base Enem Tratada2.csv': É o caminho onde o arquivo CSV será salvo. No caso, ele será salvo no diretório especificado.
- index=False: Indica que não queremos incluir o índice do DataFrame como uma coluna no arquivo CSV. Se index=True, o índice seria incluído.
- sep=';': Define o separador a ser usado no arquivo CSV. Neste caso, o separador é o ponto e vírgula (;).

- `encoding='UTF-8'`: Especifica o tipo de codificação a ser usado ao salvar o arquivo. UTF-8 é uma codificação que suporta uma ampla variedade de caracteres, útil para dados com diferentes idiomas e caracteres especiais.