In [1]:
```
from IPython.display import Image
Image('work.png')

#Algumas ferramentas utilizadas nesse trabalho.
```

Out[1]:

In [2]:
```
#Importando bibliotecas que iremos utilizar.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

#Setando configurações de dataframe para mostrar 80 colunar e 80 linhas.
pd.set_option('display.max_columns', 80 )
pd.set_option('display.max_rows', 80)
```

In [3]:
```
#Atribuindo base de dados para variável df_enem_bahia
df_enem_bahia = pd.read_csv("C:\Arquivos Felipe\Ciencia de dados - Digital House\case enem sql\enem_bahia.cs
```

In [4]:
```
df_enem_bahia
```

Out[4]:

	NU_INSCRICAO	NU_ANO	TP_FAIXA_ETARIA	TP_SEXO	TP_ESTADO_CIVIL	TP_COR_RACA	TP_NACIONALIDADE	TP_ST_C
0	200001943954	2020	4	F	2	3	2	
1	200006138472	2020	7	M	1	1	1	
2	200001934470	2020	7	F	1	3	1	
3	200001994876	2020	10	F	1	2	2	
4	200006735848	2020	10	F	1	2	1	
...	...	...	...	...	...	...	...	
447686	200002283878	2020	6	M	1	1	1	
447687	200001340390	2020	14	F	2	3	1	
447688	200002773823	2020	12	M	1	2	1	
447689	200002266997	2020	13	M	1	3	2	
447690	200004829674	2020	13	F	1	3	1	

447691 rows × 76 columns

In [5]:
```
#Observando quais variáveis não são numéricas, para decidir se serão removidas ou transformadas em numéricas
df_enem_bahia.select_dtypes(include='object')
```

Out[5]:

	TP_SEXO	NO_MUNICIPIO_ESC	SG_UF_ESC	NO_MUNICIPIO_PROVA	SG_UF_PROVA	
0	F	Salvador	BA	Salvador	BA	
1	M	NaN	NaN	Eunápolis	BA	BAEACEDBACDDEDECEDCEDBCDECBAEC
2	F	NaN	NaN	Vitória da Conquista	BA	
3	F	Feira de Santana	BA	Feira de Santana	BA	
4	F	NaN	NaN	Salvador	BA	BABDAACDEEAEBDCAEBDCBBCEDDCC
...	...	...	...	...	...	
447686	M	NaN	NaN	Salvador	BA	CEDBDBDCAECBDAEEADBADCDCCAECE
447687	F	NaN	NaN	Salvador	BA	
447688	M	NaN	NaN	Vitória da Conquista	BA	
447689	M	NaN	NaN	Alagoinhas	BA	
447690	F	NaN	NaN	Entre Rios	BA	

447691 rows × 37 columns

In [6]:
```
#Para essa análise, vou desconsiderar as variáveis que se referem a localidade ou a gabarito. Portanto irei
#os dados das colunas Object de interesse em numéricos.

df_enem_bahia.TP_SEXO = df_enem_bahia.TP_SEXO.replace({'M':0, 'F':1})
valores = {'':99,'A':1, 'B':2, 'C':3, 'D':4, 'E':5, 'F':6, 'G':7, 'H':8, 'I':9, 'J':10, 'K':11, 'L':12, 'M':
for i in ['Q0'+'{:02}'.format(i) for i in range(1,26)]:
  df_enem_bahia[i] = df_enem_bahia[i].replace(valores)
```

In [7]:
```
df_enem_bahia
```

Out[7]:

	NU_INSCRICAO	NU_ANO	TP_FAIXA_ETARIA	TP_SEXO	TP_ESTADO_CIVIL	TP_COR_RACA	TP_NACIONALIDADE	TP_ST_C
0	200001943954	2020	4	1	2	3	2	
1	200006138472	2020	7	0	1	1	1	
2	200001934470	2020	7	1	1	3	1	
3	200001994876	2020	10	1	1	2	2	
4	200006735848	2020	10	1	1	2	1	
...	...	...	...	...	...	...	...	...
447686	200002283878	2020	6	0	1	1	1	
447687	200001340390	2020	14	1	2	3	1	
447688	200002773823	2020	12	0	1	2	1	
447689	200002266997	2020	13	0	1	3	2	
447690	200004829674	2020	13	1	1	3	1	

447691 rows × 76 columns

In [8]:
```
#Criando uma nova coluna chamada media_notas, que vai receber a media das 5 notas do enem por inscrito
df_enem_bahia['media_notas'] = (df_enem_bahia['NU_NOTA_CH'] + df_enem_bahia['NU_NOTA_CN'] + df_enem_bahia['N
```

In [9]:
```
df_enem_bahia
```

Out[9]:

	NU_INSCRICAO	NU_ANO	TP_FAIXA_ETARIA	TP_SEXO	TP_ESTADO_CIVIL	TP_COR_RACA	TP_NACIONALIDADE	TP_ST_C
0	200001943954	2020	4	1	2	3	2	
1	200006138472	2020	7	0	1	1	1	
2	200001934470	2020	7	1	1	3	1	
3	200001994876	2020	10	1	1	2	2	
4	200006735848	2020	10	1	1	2	1	
...	...	...	...	...	...	...	...	
447686	200002283878	2020	6	0	1	1	1	
447687	200001340390	2020	14	1	2	3	1	
447688	200002773823	2020	12	0	1	2	1	
447689	200002266997	2020	13	0	1	3	2	
447690	200004829674	2020	13	1	1	3	1	

447691 rows × 77 columns

In [10]:
```
#Avaliando a correlação entre as demais variáveis e a coluna media_notas, selecionando apenas aquelas com co
correlacao = df_enem_bahia.corr()
correlacao = correlacao [correlacao['media_notas'].abs() >= .20]
correlacao
```

Out[10]:

	NU_INSCRICAO	NU_ANO	TP_FAIXA_ETARIA	TP_SEXO	TP_ESTADO_CIVIL	TP_COR_RACA	TP_NACI
TP_DEPENDENCIA_ADM_ESC	-0.007397	NaN	-0.216803	-0.030187	-0.028941	-0.127700	
NU_NOTA_CN	-0.001103	NaN	-0.026522	-0.169459	-0.022202	-0.088789	
NU_NOTA_CH	0.000882	NaN	-0.038307	-0.124082	-0.023667	-0.089187	
NU_NOTA_LC	-0.001809	NaN	-0.087980	-0.059230	-0.038092	-0.090213	
NU_NOTA_MT	-0.003923	NaN	-0.097046	-0.228858	-0.040592	-0.095185	
TP_STATUS_REDACAO	0.001113	NaN	0.049850	-0.035068	0.017088	0.002694	
NU_NOTA_COMP1	-0.003184	NaN	-0.146971	0.092966	-0.048272	-0.058395	
NU_NOTA_COMP2	-0.001347	NaN	-0.196600	0.075644	-0.068772	-0.056840	
NU_NOTA_COMP3	-0.003045	NaN	-0.184297	0.082538	-0.067482	-0.059241	
NU_NOTA_COMP4	-0.002269	NaN	-0.181989	0.070056	-0.062688	-0.057911	
NU_NOTA_COMP5	-0.001895	NaN	-0.196294	0.045262	-0.067672	-0.061433	
NU_NOTA_REDACAO	-0.002553	NaN	-0.208889	0.080182	-0.072735	-0.066792	
Q001	-0.001184	NaN	-0.195634	-0.061600	-0.081308	-0.062935	
Q002	0.001336	NaN	-0.313485	-0.079625	-0.113564	-0.067267	
Q003	-0.001373	NaN	-0.082151	-0.049374	-0.039175	-0.064962	
Q004	0.001801	NaN	-0.118005	-0.074063	-0.039943	-0.070912	
Q006	-0.001142	NaN	-0.108266	-0.110481	-0.004753	-0.115033	
Q008	0.000397	NaN	-0.132289	-0.061737	-0.018509	-0.111085	
Q010	0.001554	NaN	-0.141134	-0.077994	-0.003400	-0.087708	
Q013	-0.001025	NaN	-0.172899	-0.048075	-0.049387	-0.071842	
Q014	0.000076	NaN	-0.032823	-0.096477	0.018594	-0.072129	
Q016	-0.000305	NaN	-0.047161	-0.072651	0.013102	-0.056452	
Q018	-0.003267	NaN	-0.057413	-0.055312	0.006172	-0.075103	
Q019	0.000290	NaN	-0.115816	-0.069883	-0.034370	-0.088788	
Q021	0.000177	NaN	-0.093870	-0.034777	-0.017407	-0.065767	
Q022	-0.000155	NaN	-0.193605	-0.065612	-0.058444	-0.057648	
Q024	0.000805	NaN	-0.021435	-0.121872	0.012265	-0.093874	
Q025	-0.001777	NaN	-0.021226	-0.062147	0.016900	-0.031581	
media_notas	-0.002681	NaN	-0.144006	-0.090480	-0.059107	-0.104880	

Conhecendo do que se trata cada variável com correlação superior a 20%. TP_DEPENDENCIA_ADM_ESC Dependência administrativa (Escola) 1 Federal 2 Estadual 3 Municipal 4 Privada NU_NOTA_CN Nota da prova de Ciências da Natureza NU_NOTA_CH Nota da prova de Ciências Humanas NU_NOTA_LC Nota da prova de Linguagens e Códigos NU_NOTA_MT Nota da prova de Matemática NU_NOTA_COMP1 Nota da competência 1 - Demonstrar domínio da modalidade escrita formal da Língua Portuguesa. NU_NOTA_COMP2 Nota da competência 2 - Compreender a proposta de redação e aplicar conceitos das várias áreas de conhecimento para desenvolver o tema, dentro dos limites estruturais do texto dissertativo-argumentativo em prosa. NU_NOTA_COMP3 Nota da competência 3 - Selecionar, relacionar, organizar e interpretar informações, fatos, opiniões e argumentos em defesa de um ponto de vista. NU_NOTA_COMP4 Nota da competência 4 - Demonstrar conhecimento dos mecanismos linguísticos necessários para a construção da argumentação. NU_NOTA_COMP5 Nota da competência 5 - Elaborar proposta de intervenção para o problema abordado, respeitando os direitos humanos. NU_NOTA_REDACAO Nota da prova de redação Q001 Até que série seu pai, ou o homem responsável por você, estudou? A Nunca estudou. B Não completou a 4ª série/5º ano do Ensino Fundamental. C Completou a 4ª série/5º ano, mas não completou a 8ª série/9º ano do Ensino Fundamental. D Completou a 8ª série/9º ano do Ensino Fundamental, mas não completou o Ensino Médio. 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. Q002 Até que série sua mãe, ou a mulher responsável por você, estudou? A Nunca estudou. B Não completou a 4ª série/5º ano do Ensino Fundamental. C Completou a 4ª série/5º ano, mas não completou a 8ª série/9º ano do Ensino Fundamental. D Completou a 8ª série/9º ano do Ensino Fundamental, mas não completou o Ensino Médio. 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. Q003 A partir da apresentação de algumas ocupações divididas em grupos ordenados, indique o grupo que contempla a ocupação mais próxima da ocupação do seu pai ou do homem responsável por você. (Se ele não estiver trabalhando, escolha uma ocupação pensando no último trabalho dele). A Grupo 1: Lavrador, agricultor sem empregados, bóia fria, criador de animais (gado, porcos, galinhas, ovelhas, cavalos etc.), apicultor, pescador, lenhador, seringueiro, extrativista. B Grupo 2: Diarista, empregado doméstico, cuidador de idosos, babá, cozinheiro (em casas particulares), motorista particular, jardineiro, faxineiro de empresas e prédios, vigilante, porteiro, carteiro, office-boy, vendedor, caixa, atendente de loja, auxiliar administrativo, recepcionista, servente de pedreiro, repositor de mercadoria. C Grupo 3: Padeiro, cozinheiro industrial ou em restaurantes, sapateiro, costureiro, joalheiro, torneiro mecânico, operador de máquinas, soldador,

operário de fábrica, trabalhador da mineração, pedreiro, pintor, eletricista, encanador, motorista, caminhoneiro, taxista. D Grupo 4: Professor (de ensino fundamental ou médio, idioma, música, artes etc.), técnico (de enfermagem, contabilidade, eletrônica etc.), policial, militar de baixa patente (soldado, cabo, sargento), corretor de imóveis, supervisor, gerente, mestre de obras, pastor, microempresário (proprietário de empresa com menos de 10 empregados), pequeno comerciante, pequeno proprietário de terras, trabalhador autônomo ou por conta própria. E Grupo 5: Médico, engenheiro, dentista, psicólogo, economista, advogado, juiz, promotor, defensor, delegado, tenente, capitão, coronel, professor universitário, diretor em empresas públicas ou privadas, político, proprietário de empresas com mais de 10 empregados. F Não sei. Q004 A partir da apresentação de algumas ocupações divididas em grupos ordenados, indique o grupo que contempla a ocupação mais próxima da ocupação da sua mãe ou da mulher responsável por você. (Se ela não estiver trabalhando, escolha uma ocupação pensando no último trabalho dela). A Grupo 1: Lavradora, agricultora sem empregados, bóia fria, criadora de animais (gado, porcos, galinhas, ovelhas, cavalos etc.), apicultora, pescadora, lenhadora, seringueira, extrativista. B Grupo 2: Diarista, empregada doméstica, cuidadora de idosos, babá, cozinheira (em casas particulares), motorista particular, jardineira, faxineira de empresas e prédios, vigilante, porteira, carteira, office-boy, vendedora, caixa, atendente de loja, auxiliar administrativa, recepcionista, servente de pedreiro, repositora de mercadoria. C Grupo 3: Padeira, cozinheira industrial ou em restaurantes, sapateira, costureira, joalheira, torneira mecânica, operadora de máquinas, soldadora, operária de fábrica, trabalhadora da mineração, pedreira, pintora, eletricista, encanadora, motorista, caminhoneira, taxista. D Grupo 4: Professora (de ensino fundamental ou médio, idioma, música, artes etc.), técnica (de enfermagem, contabilidade, eletrônica etc.), policial, militar de baixa patente (soldado, cabo, sargento), corretora de imóveis, supervisora, gerente, mestre de obras, pastora, microempresária (proprietária de empresa com menos de 10 empregados), pequena comerciante, pequena proprietária de terras, trabalhadora autônoma ou por conta própria. E Grupo 5: Médica, engenheira, dentista, psicóloga, economista, advogada, juíza, promotora, defensora, delegada, tenente, capitã, coronel, professora universitária, diretora em empresas públicas ou privadas, política, proprietária de empresas com mais de 10 empregados. F Não sei. Q006 Qual é a renda mensal de sua família? (Some a sua renda com a dos seus familiares.) A Nenhuma Renda B Até R$1.045,00 CDeR$ 1.045,01 até R$1.567,50 DDeR$ 1.567,51 até R$2.090,00 EDeR$ 2.090,01 até R$2.612,50 FDeR$ 2.612,51 até R$3.135,00 GDeR$ 3.135,01 até R$4.180,00 HDeR$ 4.180,01 até R$5.225,00 IDeR$ 5.225,01 até R$6.270,00 JDeR$ 6.270,01 até R$7.315,00 KDeR$ 7.315,01 até R$8.360,00 LDeR$ 8.360,01 até R$9.405,00 MDeR$ 9.405,01 até R$10.450,00 NDeR$ 10.450,01 até R$12.540,00 ODeR$ 12.540,01 até R$15.675,00 PDeR$ 15.675,01 até R$20.900,00 QAcimadeR$ 20.900,00 Q008 Na sua residência tem banheiro? A Não. B Sim, um. C Sim, dois. D Sim, três. E Sim, quatro ou mais. Q010 Na sua residência tem carro? A Não. B Sim, um. C Sim, dois. D Sim, três. E Sim, quatro ou mais. Q013 Na sua residência tem freezer (independente ou segunda porta da geladeira)? A Não. B Sim, um. C Sim, dois. D Sim, três. E Sim, quatro ou mais. Q014 Na sua residência tem máquina de lavar roupa? (o tanquinho NÃO deve ser considerado) A Não. B Sim, uma. C Sim, duas. D Sim, três. E Sim, quatro ou mais. Q016 Na sua residência tem forno micro-ondas? A Não. B Sim, um. C Sim, dois. D Sim, três. E Sim, quatro ou mais. Q018 Na sua residência tem aspirador de pó? A Não. B Sim. Q019 Na sua residência tem televisão em cores? A Não. B Sim, uma. C Sim, duas. D Sim, três. E Sim, quatro ou mais. Q021 Na sua residência tem TV por assinatura? A Não. B Sim. Q022 Na sua residência tem telefone celular? A Não. B Sim, um. C Sim, dois. D Sim, três. E Sim, quatro ou mais. Q024 Na sua residência tem computador? A Não. B Sim, um. C Sim, dois. D Sim, três. E Sim, quatro ou mais. Q025 Na sua residência tem acesso à Internet? A Não. B Sim. Para esta análise, não faz sentido utilizar as variáveis que compõe a nota, já que media_notas foi gerada a partir de algumas delas, portanto serão desconsideradas.

In [11]:
```
#Algumas variáveis possuem alta correlação com a renda familiar, e talvez fosse interessantes desconsiderá-l
#Não entrarei nesse tópico e manterei todas pois o Objetivo é meramente demonstrar a utilização de algumas f
print('Correlação das variáveis com a Renda Familiar')
print(correlacao['Q006'])
```

```
Correlação das variáveis com a Renda Familiar
TP_DEPENDENCIA_ADM_ESC    0.531284
NU_NOTA_CN                0.393776
NU_NOTA_CH                0.364456
NU_NOTA_LC                0.359148
NU_NOTA_MT                0.429602
TP_STATUS_REDACAO        -0.046023
NU_NOTA_COMP1             0.293362
NU_NOTA_COMP2             0.246575
NU_NOTA_COMP3             0.271267
NU_NOTA_COMP4             0.290720
NU_NOTA_COMP5             0.268304
NU_NOTA_REDACAO           0.307744
Q001                      0.258908
Q002                      0.340518
Q003                      0.268265
Q004                      0.348100
Q006                      1.000000
Q008                      0.617357
Q010                      0.575153
Q013                      0.352922
Q014                      0.425443
Q016                      0.343998
Q018                      0.425593
Q019                      0.520518
Q021                      0.415684
Q022                      0.419566
Q024                      0.548382
Q025                      0.229153
media_notas               0.453739
Name: Q006, dtype: float64
```

In [12]:
```
#TP_DEPENDENCIA_ADM_ESC será desconsiderada por ter muitas colunas NaN equivalente a 98% dos dados do DataFr
#Plotando soma da quantidade de valores únicos para cada variável.
df_enem_bahia.isna().sum()
```

Out[12]:
```
NU_INSCRICAO                  0
NU_ANO                        0
TP_FAIXA_ETARIA               0
TP_SEXO                       0
TP_ESTADO_CIVIL               0
TP_COR_RACA                   0
TP_NACIONALIDADE              0
TP_ST_CONCLUSAO               0
TP_ANO_CONCLUIU               0
TP_ESCOLA                     0
TP_ENSINO                362136
IN_TREINEIRO                  0
CO_MUNICIPIO_ESC         400530
NO_MUNICIPIO_ESC         400530
CO_UF_ESC                400530
SG_UF_ESC                400530
TP_DEPENDENCIA_ADM_ESC   400530
TP_LOCALIZACAO_ESC       400530
TP_SIT_FUNC_ESC          400530
CO_MUNICIPIO_PROVA            0
NO_MUNICIPIO_PROVA            0
CO_UF_PROVA                   0
SG_UF_PROVA                   0
TP_PRESENCA_CN                0
TP_PRESENCA_CH                0
TP_PRESENCA_LC                0
TP_PRESENCA_MT                0
CO_PROVA_CN              239740
CO_PROVA_CH              229356
CO_PROVA_LC              229356
CO_PROVA_MT              239740
NU_NOTA_CN               239740
NU_NOTA_CH               229356
NU_NOTA_LC               229356
NU_NOTA_MT               239740
TX_RESPOSTAS_CN          239740
TX_RESPOSTAS_CH          229356
TX_RESPOSTAS_LC          229356
TX_RESPOSTAS_MT          239740
TP_LINGUA                     0
TX_GABARITO_CN           239740
TX_GABARITO_CH           229356
TX_GABARITO_LC           229356
TX_GABARITO_MT           239740
TP_STATUS_REDACAO        229356
NU_NOTA_COMP1            229356
NU_NOTA_COMP2            229356
NU_NOTA_COMP3            229356
NU_NOTA_COMP4            229356
NU_NOTA_COMP5            229356
NU_NOTA_REDACAO          229356
Q001                       2955
Q002                       2955
Q003                       2955
Q004                       2955
Q005                       2955
Q006                       2955
Q007                       2955
Q008                       2955
Q009                       2955
Q010                       2955
Q011                       2955
Q012                       2955
Q013                       2955
Q014                       2955
Q015                       2955
Q016                       2955
Q017                       2955
Q018                       2955
Q019                       2955
Q020                       2955
Q021                       2955
Q022                       2955
Q023                       2955
Q024                       2955
Q025                       2955
media_notas              240330
dtype: int64
```

In [13]:
```
#Removendo valores NaN das colunas que serão usadas no modelo.
```

```
df_enem_bahia = df_enem_bahia.dropna(subset = ['media_notas','Q001', 'Q002', 'Q003', 'Q004',
       'Q006', 'Q008', 'Q013','Q014', 'Q016', 'Q018', 'Q019', 'Q021', 'Q022',
       'Q024', 'Q025'])
df_enem_bahia
```

Out[13]:

	NU_INSCRICAO	NU_ANO	TP_FAIXA_ETARIA	TP_SEXO	TP_ESTADO_CIVIL	TP_COR_RACA	TP_NACIONALIDADE	TP_ST_C
1	200006138472	2020	7	0	1	1	1	
4	200006735848	2020	10	1	1	2	1	
5	200001346535	2020	12	1	1	3	1	
8	200005885744	2020	6	1	1	3	1	
9	200001769014	2020	5	0	1	1	1	
...	...	...	...	...	...	...	...	
447669	200003001071	2020	5	1	1	1	1	
447670	200001270036	2020	13	1	2	1	1	
447675	200006396025	2020	4	1	1	2	1	
447679	200001959412	2020	7	1	2	2	1	
447686	200002283878	2020	6	0	1	1	1	

206328 rows × 77 columns

In [14]:
```
#Visualizando todas as colunas do DataFrame
df_enem_bahia.columns
```

Out[14]:
```
Index(['NU_INSCRICAO', 'NU_ANO', 'TP_FAIXA_ETARIA', 'TP_SEXO',
       'TP_ESTADO_CIVIL', 'TP_COR_RACA', 'TP_NACIONALIDADE', 'TP_ST_CONCLUSAO',
       'TP_ANO_CONCLUIU', 'TP_ESCOLA', 'TP_ENSINO', 'IN_TREINEIRO',
       'CO_MUNICIPIO_ESC', 'NO_MUNICIPIO_ESC', 'CO_UF_ESC', 'SG_UF_ESC',
       'TP_DEPENDENCIA_ADM_ESC', 'TP_LOCALIZACAO_ESC', 'TP_SIT_FUNC_ESC',
       'CO_MUNICIPIO_PROVA', 'NO_MUNICIPIO_PROVA', 'CO_UF_PROVA',
       'SG_UF_PROVA', 'TP_PRESENCA_CN', 'TP_PRESENCA_CH', 'TP_PRESENCA_LC',
       'TP_PRESENCA_MT', 'CO_PROVA_CN', 'CO_PROVA_CH', 'CO_PROVA_LC',
       'CO_PROVA_MT', 'NU_NOTA_CN', 'NU_NOTA_CH', 'NU_NOTA_LC', 'NU_NOTA_MT',
       'TX_RESPOSTAS_CN', 'TX_RESPOSTAS_CH', 'TX_RESPOSTAS_LC',
       'TX_RESPOSTAS_MT', 'TP_LINGUA', 'TX_GABARITO_CN', 'TX_GABARITO_CH',
       'TX_GABARITO_LC', 'TX_GABARITO_MT', 'TP_STATUS_REDACAO',
       'NU_NOTA_COMP1', 'NU_NOTA_COMP2', 'NU_NOTA_COMP3', 'NU_NOTA_COMP4',
       'NU_NOTA_COMP5', 'NU_NOTA_REDACAO', 'Q001', 'Q002', 'Q003', 'Q004',
       'Q005', 'Q006', 'Q007', 'Q008', 'Q009', 'Q010', 'Q011', 'Q012', 'Q013',
       'Q014', 'Q015', 'Q016', 'Q017', 'Q018', 'Q019', 'Q020', 'Q021', 'Q022',
       'Q023', 'Q024', 'Q025', 'media_notas'],
      dtype='object')
```

In [15]:
```
#Atribuinudo as colunas que serão "features" à variável "X" e a coluna "target" à variável "y"

X = df_enem_bahia[['Q001', 'Q002', 'Q003', 'Q004',
       'Q006', 'Q008', 'Q010', 'Q013','Q014', 'Q016', 'Q018', 'Q019', 'Q021', 'Q022',
       'Q024', 'Q025']]

y = df_enem_bahia['media_notas']
```

In [16]:
```
#Importando ferramenta de separação de dados de treino e teste do Scikit Learn.
from sklearn.model_selection import train_test_split
```

In [17]:
```
#Separando variáveis de treino e teste para X e y
X_train, X_test, y_train, y_test = train_test_split(X, y)
```

In [18]:
```
#Importando stats model

import statsmodels.api as sm
ols_regressor = sm.OLS(y_train, X_train)
model = ols_regressor.fit()
```

In [19]:
```
#Verificando aspectos estatísticos do modelo.
#Teste de Significância individuais ou p-values dos coeficientes: diz o quanto das variáveis preditoras expl
#Coeficiente R²: diz o quanto o meu modelo explica seus resultados. É um valor entre 0 e 1. Quanto mais próx

model.summary()
```

Out[19]:

OLS Regression Results

Dep. Variable:	media_notas	**R-squared (uncentered):**	0.970
Model:	OLS	**Adj. R-squared (uncentered):**	0.970
Method:	Least Squares	**F-statistic:**	3.099e+05
Date:	Tue, 30 Aug 2022	**Prob (F-statistic):**	0.00
Time:	21:47:38	**Log-Likelihood:**	-9.1538e+05
No. Observations:	154746	**AIC:**	1.831e+06
Df Residuals:	154730	**BIC:**	1.831e+06
Df Model:	16		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Q001	3.0042	0.145	20.694	0.000	2.720	3.289
Q002	10.3829	0.167	62.243	0.000	10.056	10.710
Q003	1.2914	0.178	7.238	0.000	0.942	1.641
Q004	-0.0655	0.194	-0.337	0.736	-0.446	0.316
Q006	-5.5812	0.114	-48.977	0.000	-5.805	-5.358
Q008	27.2221	0.498	54.674	0.000	26.246	28.198
Q010	2.3173	0.525	4.415	0.000	1.288	3.346
Q013	19.9149	0.535	37.205	0.000	18.866	20.964
Q014	6.3075	0.583	10.823	0.000	5.165	7.450
Q016	10.0272	0.542	18.515	0.000	8.966	11.089
Q018	79.6124	0.901	88.397	0.000	77.847	81.378
Q019	13.7341	0.476	28.871	0.000	12.802	14.666
Q021	37.9571	0.767	49.471	0.000	36.453	39.461
Q022	8.6568	0.262	33.053	0.000	8.144	9.170
Q024	11.1985	0.423	26.504	0.000	10.370	12.027
Q025	78.2950	0.578	135.526	0.000	77.163	79.427

Omnibus:	588.760	**Durbin-Watson:**	1.992
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	763.242
Skew:	-0.055	**Prob(JB):**	1.84e-166
Kurtosis:	3.326	**Cond. No.**	42.5

Notes:

[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.

[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [20]:
```
#Criando variável de previsão utilizando modelo anterior.
pred_X_test = model.predict(X_test)
```

In [21]:
```
#Criando novo DataFrame para receber resultados previstos pelo modelo e resultados reais separados para test
novo_df_enem_bahia = pd.DataFrame([pred_X_test, y_test])
novo_df_enem_bahia = novo_df_enem_bahia.transpose()
novo_df_enem_bahia.columns = ['pred_X_test', 'y_test']
novo_df_enem_bahia = novo_df_enem_bahia.reset_index().drop(columns = ['index'])
print(novo_df_enem_bahia)
```

```
       pred_X_test  y_test
0       457.677258  433.08
1       543.314082  645.94
2       555.690073  527.62
3       505.501491  517.84
4       510.963280  408.90
...            ...     ...
51577   481.220471  403.30
51578   579.200109  568.60
51579   730.699705  733.62
51580   487.330221  523.56
51581   563.623879  665.70

[51582 rows x 2 columns]
```

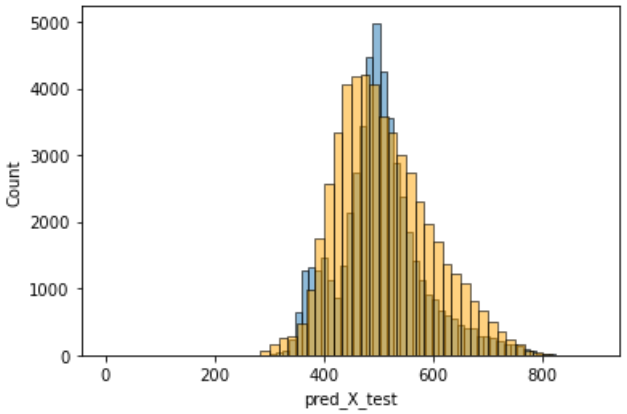
In [22]:
```
#Importando ferramentas de avalição de erro em modelo de regressão por métodos estatísticos
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
```

In [23]:
```
#Definindo uma função para tornar mais fácil a plotagem dos métodos estatísticos quando necessário utilizar
def desvio_3 (previsao, teste):
    print('root_mean_squared_error é igual a ' + str(mean_squared_error (previsao, teste)**0.5))
    print('mean_squared_error é igual a ' + str(mean_squared_error (previsao, teste)))
    print('mean_absolute_error é igual a ' + str(mean_absolute_error (previsao, teste)))
```

In [24]:
```
#Para todos os métodos, quanto mais próximo de "Zero", melhor.
desvio_3 (pred_X_test, y_test)
```

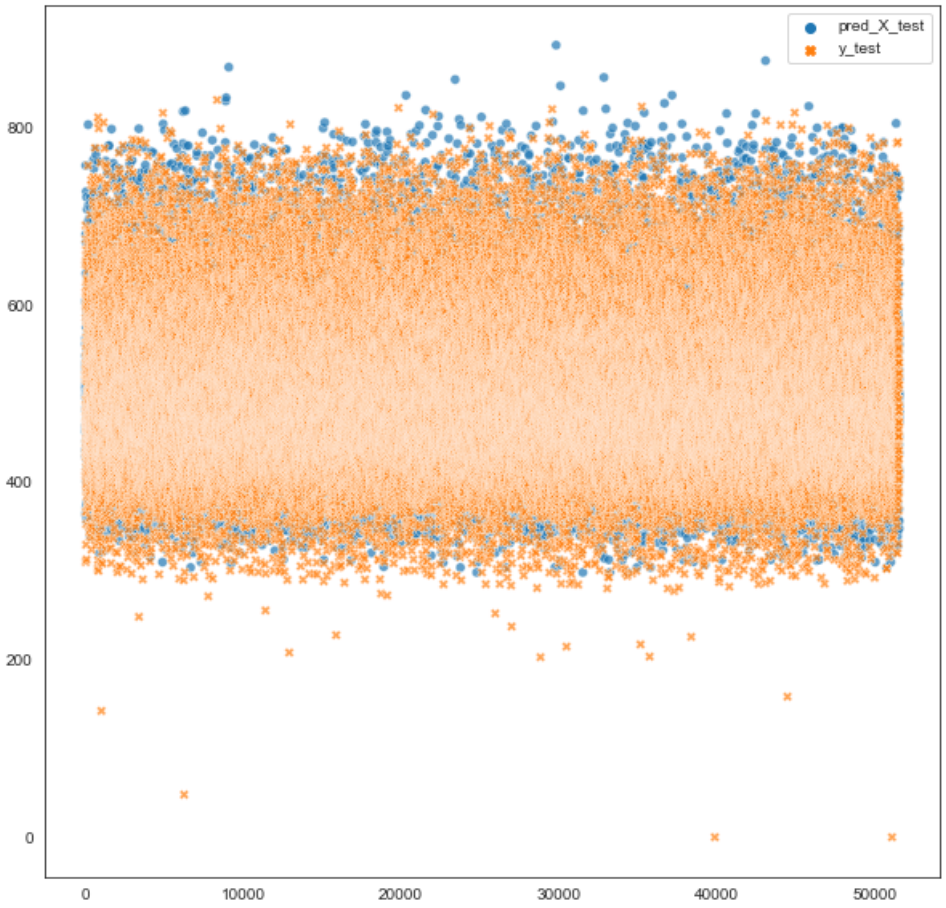
```
root_mean_squared_error é igual a 90.0303088977871
mean_squared_error é igual a 8105.456520230961
mean_absolute_error é igual a 70.9688196793623
```

In [25]:
```
#Plotando gráfico para visualizar melhor o comportamento do modelo - Laranja valor real, Azul valor previsto
sns.histplot(novo_df_enem_bahia['pred_X_test'], bins = 50, alpha = .5)
sns.histplot(novo_df_enem_bahia['y_test'], color = "orange", bins = 50, alpha = 0.50);
```

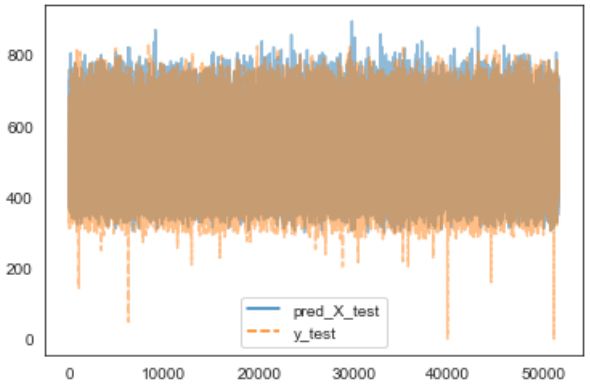


In [26]:
```
# Criando o ambiente do gráfico
sns.set_style("white")
plt.figure(figsize=(10, 10))

# Gráfico de Dispersão
g = sns.scatterplot(data=novo_df_enem_bahia, alpha = .7)
plt.show()
```



In [27]:
```
sns.lineplot(data=novo_df_enem_bahia, alpha = .5)
plt.show()
```



In [28]:
```
#Criando variável para mensurar erro absoluto do modelo para cada previsão.

novo_df_enem_bahia['erro_pred'] = abs(novo_df_enem_bahia['pred_X_test'] - novo_df_enem_bahia['y_test'])
```

In [29]:
```
novo_df_enem_bahia
```

Out[29]:

	pred_X_test	y_test	erro_pred
0	457.677258	433.08	24.597258
1	543.314082	645.94	102.625918
2	555.690073	527.62	28.070073
3	505.501491	517.84	12.338509
4	510.963280	408.90	102.063280
...	...	...	...
51577	481.220471	403.30	77.920471
51578	579.200109	568.60	10.600109
51579	730.699705	733.62	2.920295
51580	487.330221	523.56	36.229779
51581	563.623879	665.70	102.076121

51582 rows × 3 columns

In [30]:
```
#Somando todos os erros do modelo erro_pred
novo_df_enem_bahia.sum(axis = 0)
```

Out[30]:
```
pred_X_test    2.589991e+07
y_test         2.622295e+07
erro_pred      3.660714e+06
dtype: float64
```

In [31]:
```
#Criando um modelo de regressão utilizando os todos os dados originais, sem repartição em treino e teste.
ols_regressor2 = sm.OLS(y, X)
model_final = ols_regressor2.fit()
```

In [32]:
```
#Verificando aspectos estatísticos do modelo.
#Teste de Significância individuais ou p-values dos coeficientes: diz o quanto das variáveis preditoras expl
#Coeficiente R²: diz o quanto o meu modelo explica seus resultados. É um valor entre 0 e 1. Quanto mais próx

model_final.summary()
```

Out[32]:

OLS Regression Results

Dep. Variable:	media_notas	**R-squared (uncentered):**	0.970
Model:	OLS	**Adj. R-squared (uncentered):**	0.970
Method:	Least Squares	**F-statistic:**	4.126e+05
Date:	Tue, 30 Aug 2022	**Prob (F-statistic):**	0.00
Time:	21:47:46	**Log-Likelihood:**	-1.2207e+06
No. Observations:	206328	**AIC:**	2.441e+06
Df Residuals:	206312	**BIC:**	2.442e+06
Df Model:	16		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Q001	2.8351	0.126	22.501	0.000	2.588	3.082
Q002	10.3664	0.145	71.673	0.000	10.083	10.650
Q003	1.2804	0.155	8.265	0.000	0.977	1.584
Q004	0.0366	0.168	0.217	0.828	-0.293	0.366
Q006	-5.5683	0.098	-56.552	0.000	-5.761	-5.375
Q008	27.0078	0.431	62.597	0.000	26.162	27.853
Q010	2.4214	0.455	5.324	0.000	1.530	3.313
Q013	19.6174	0.464	42.309	0.000	18.709	20.526
Q014	6.6960	0.505	13.252	0.000	5.706	7.686
Q016	10.0168	0.469	21.353	0.000	9.097	10.936
Q018	80.5766	0.781	103.169	0.000	79.046	82.107
Q019	13.6430	0.412	33.121	0.000	12.836	14.450
Q021	37.4442	0.663	56.459	0.000	36.144	38.744
Q022	8.4744	0.227	37.365	0.000	8.030	8.919
Q024	11.2559	0.366	30.735	0.000	10.538	11.974
Q025	78.7450	0.501	157.192	0.000	77.763	79.727

Omnibus:	805.782	**Durbin-Watson:**	1.966
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	1053.208
Skew:	-0.054	**Prob(JB):**	1.99e-229
Kurtosis:	3.333	**Cond. No.**	42.5

Notes:

[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.

[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [33]:
```
#Criando um personagem Felipe para tentar prever a nota a partir das respostas do questionário sócio-economi
felipe = (5,6,4,4,8,2,1,2,2,2,1,2,1,2,2,2)
```

In [34]:
```
#Gerando a previsão a partir do modelo de teste.
previsao_felipe_test = model.predict(felipe)
print(previsao_felipe_test)
```

```
[508.17147603]
```

In [35]:
```
#Gerando a previsão a partir do modelo final, gerado utilizando todos os dados.
previsao_felipe = model_final.predict(felipe)
print(previsao_felipe)
```

```
[508.45109606]
```

A nota de corte de CIÊNCIAS DA COMPUTAÇÃO pelo SISU média em salvador-ba é 646,53 pontos, já a menor nota foi de 494,11 pontos para a instituição UFBA - Universidade Federal da Bahia (). Ou seja, segundo esse modelo, talvez eu entrasse raspando no curso de Ciencias da Computação na UFBA ('--). Espero que o próximo me dê uma moralzinha maior kkkkk.