CCT College Dublin Continuous Assessment

Assessment Cover Page

Module Title:	*Statistical Techniques for Data Analysis*
Assessment Title:	Integrated CA
Lecturer Name:	*Aldana Louzan*
Student Full Name:	*Laercio Santos Lima*
Student Number:	*2022055*
Assessment Due Date:	*May 27th 2022*
Date of Submission:	*May 27th 2022*

Declaration

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

CCT College Dublin
HDip in Science in Data Analytics for Business
Continuous Assessment

LAERCIO SANTOS LIMA

ENEM 2019

A Statistical Perspective

DUBLIN
2022

ABSTRACT

This project aims to use a Jupyter Notebook to complete a list of tasks regarding statistics. In order to perform the statistical tests, this project also contains Exploratory Data Analysis (EDA) and Data Preparation. The dataset used for this project is about ENEM 2019. Some hypothesis tests are performed in order to draw some analysis and conclusions. In addition, correlation analysis is carried out in some variables. The concepts of correlation vs causation are addressed in this project, using some variables of the dataset to exemplify. Finally, linear regression is used to support the previous idea, along with different types of predictions. Different types of plots were used in every stage of this project.

Keywords: statistics, EDA, Data Preparation, hypothesis test, t-test, correlation, causation, linear regression

TABLE OF CONTENT

1. INTRODUCTION

According to Chappelow (2019), statistics can be described as a branch of applied mathematics that involves the collection, analysis, and inference of conclusions from quantitative data. In other words, it is possible to make use of statistics to do different things, such as, to test hypotheses and draw conclusions.

For this project, different techniques will be applied in order to better understand ENEM in Brazil. Some data preparation and machine learning models will also be applied so that it can be possible to have a better idea of the dataset.

1.1 What Is ENEM?

ENEM is a non-mandatory Brazilian National High School Exam. In general, this exam is taken for those who want to be admitted to a college program in Brazil. However, it is also accepted for some universities in Portugal. ENEM is also popular among people who want to test their own knowledge, since this exam is considered rather difficult for those that are not used to the recent school subjects.

Why did I choose ENEM?

Before moving to Dublin, I used to be a teacher in Brazil. I used to plan some of my classes to help and prepare my students for this exam. Thus, being able to have the opportunity to analyze this dataset is something that may help me to have a completely different picture of what happens with the students. Besides that, I am from Rio de Janeiro, the second richest state in Brazil. It means that the school's situation in Rio is not the same as in the rest of the country. With this in mind, I would like to know if in general Rio performs better than other states according to ENEM. I believe that by analyzing this dataset I may better understand the education system in Brazil.

2. DATA UNDERSTANDING

This dataset was found online; however, it was not available for download. So, in order to have the dataset, I had to individually copy each of the 196 pages of the website (EVOLUCIONAL), and paste them one by one in a google sheet. After doing that, I saved the file as a CSV document in order to read it in a Jupyter Notebook. Still about ENEM dataset, it reflects schools in Brazil that are registered at Evolucional. It contains over 19 thousand rows and 14 columns. The exam was taken in 2019.

Finally, all the calculations were done using Python (Jupyter Notebook). I decided to keep some of my Python code in the pictures, since it may help me to review them in the future.

2.1 Data Dictionary

Data dictionary

	Abbreviation / Name	Full name of Variable in Portuguese	Full name of Variable in English	Qualitative / Quantitative	type	Definition
0	rank	posicao	rank	Quantitative	integer	rank of the schools in Brazil
1	inep_code	codigo do inep	inep code	Quantitative	integer	school id number
2	school	escola	school	Qualitative	string	name of the school
3	state	estado	state name	Qualitative	string	self-explanatory
4	city	cidade	city name	Qualitative	string	self-explanatory
5	school_type	tipo	school type	Qualitative	string	self-explanatory
6	location	locatizacao	location	Qualitative	string	self-explanatory
7	students	numero de alunos	number of students	Quantitative	integer	number of students who took the test this year
8	ch	ciencias humanas e suas tecnologias	human sciences and their technologies	Quantitative	float	enem subject
9	cn	ciencias da natureza e suas tecnologias	natural sciences and their technologies	Quantitative	float	enem subject
10	lc	liguagens, codigos e suas tecnologias	languages, codes and their technologies	Quantitative	float	enem subject
11	mt	matematica e suas tecnologias	mathematics and its technologies	Quantitative	float	enem subject
12	rd	redacao	essay	Quantitative	float	enem subject
13	average_exam	media	average	Quantitative	float	average of the subjects

2.2 First look

Enem dataset

	rank	inep_code	school	state	city	school_type	location	students	ch	cn	lc	mt	rd	average_exam
0	1	23246847	FARIAS BRITO COLEGIO DE APLICACAO	Ceará	Fortaleza	Privada	Urbana	35	692.85	674.50	652.24	845.89	935.43	760.18
1	2	23246871	ARI DE SA CAVALCANTE SEDE MARIO MAMEDE COLEGIO	Ceará	Fortaleza	Privada	Urbana	33	695.67	676.34	652.91	836.65	915.15	755.34
2	3	31350664	COLEGIO BERNOULLI	Minas Gerais	Belo Horizonte	Privada	Urbana	280	681.58	668.20	634.47	823.80	906.64	742.94

2.3 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is an important step for this project since it is in this part where I am able to identify features patterns. According to Patil (2018), EDA can be defined as a critical process of performing initial investigations in a dataset. As previously mentioned in this report, ML models will be performed. Consequently, here in the EDA I intend to discover the necessary patterns and characteristics that will help me to build better models in the future. Besides that, any extra anomalies spotted in this stage may be treated in future steps.

2.3.1 Info and describe

In order to have a better idea of the dataset characteristics, I decided to use info() and describe(). As it is possible to see below, the dataset contains over 19 thousand observations and 14 features.

Enem - Info

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19598 entries, 0 to 19597
Data columns (total 14 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   rank          19598 non-null  int64
 1   inep_code     19598 non-null  int64
 2   school        19598 non-null  object
 3   state         19598 non-null  object
 4   city          19598 non-null  object
 5   school_type   19598 non-null  object
 6   location      19598 non-null  object
 7   students      19598 non-null  int64
 8   ch            19598 non-null  float64
 9   cn            19598 non-null  float64
 10  lc            19598 non-null  float64
 11  mt            19598 non-null  float64
 12  rd            19598 non-null  float64
 13  average_exam  19598 non-null  float64
dtypes: float64(6), int64(3), object(5)
memory usage: 2.1+ MB
```

Enem - describe

```
# Showing some info
enem.describe()
```

	rank	inep_code	students	ch	cn	lc	mt	rd	average_exam
count	19598.000000	1.959800e+04	19598.000000	19598.000000	19598.000000	19598.000000	19598.000000	19598.000000	19598.000000
mean	9799.499949	3.247308e+07	46.481988	504.924703	472.743004	517.846299	522.267425	576.807078	518.917680
std	5657.599574	9.324204e+06	46.220439	46.444994	46.433354	36.839751	68.626260	108.502122	58.754365
min	1.000000	1.100006e+07	10.000000	396.650000	380.660000	377.890000	401.860000	138.890000	360.460000
25%	4900.250000	2.611379e+07	17.000000	472.180000	440.000000	492.250000	474.522500	504.290000	478.460000
50%	9799.500000	3.304433e+07	30.000000	495.610000	458.795000	514.660000	501.550000	554.670000	503.760000
75%	14698.750000	3.590848e+07	58.000000	528.330000	492.517500	538.660000	548.925000	633.585000	547.030000
max	19598.000000	5.308200e+07	616.000000	695.670000	682.900000	652.910000	845.890000	938.950000	760.180000

```
# Looking for objects
enem.describe(include = object)
```

	school	state	city	school_type	location
count	19598	19598	19598	19598	19598
unique	18789	27	4742	4	2
top	EE - COLEGIO ESTADUAL LUIS EDUARDO MAGALHAES	São Paulo	São Paulo	Estadual	Urbana
freq	18	4330	905	14666	18752

2.3.2 Shape, size and number of zeros

Now I am going to show some characteristics regarding the dataset's size. In order to do it, I am going to check its size, shape and number of zeros.

Enem - size, shape and number of zeros

```
# Checking the shape
enem.shape
```

```
(19598, 14)
```

```
# Checking the size
enem.size
```

```
274372
```

```
# Checking the number of zeros
number_of_zeros = (enem.to_numpy() == 0).sum()
print (number_of_zeros)
```

```
0
```

2.3.3 Duplicated rows

According to Bhutani (2018), another important part of data analysis is analyzing duplicated values, and subsequently, deciding to remove them or not. Based on that, I am going to look for duplicated rows in the dataset. In order to do it, first I am going to create a new dataframe. After that, I am going to show the amount of rows that are duplicated.

Enem - duplicated rows

```
# Looking for duplicated rows
dup_enem = enem[enem.duplicated()]
```

```
# Showing the number of duplicated rows
print("Number of duplicated rows: ", dup_enem.shape)
```

```
Number of duplicated rows:  (0, 14)
```

2.3.4 Missing values

According to Chandradas (2021), NaN is a short form for Not A Number. In other words, it is a possible form to show a missing value in a dataset. With this information, I decided to count the amount of missing values in the dataset. In order to do this, I am going to use isna() and sum(), since my intention is to better understand the situation with missing values in the dataset.

Enem - missing values

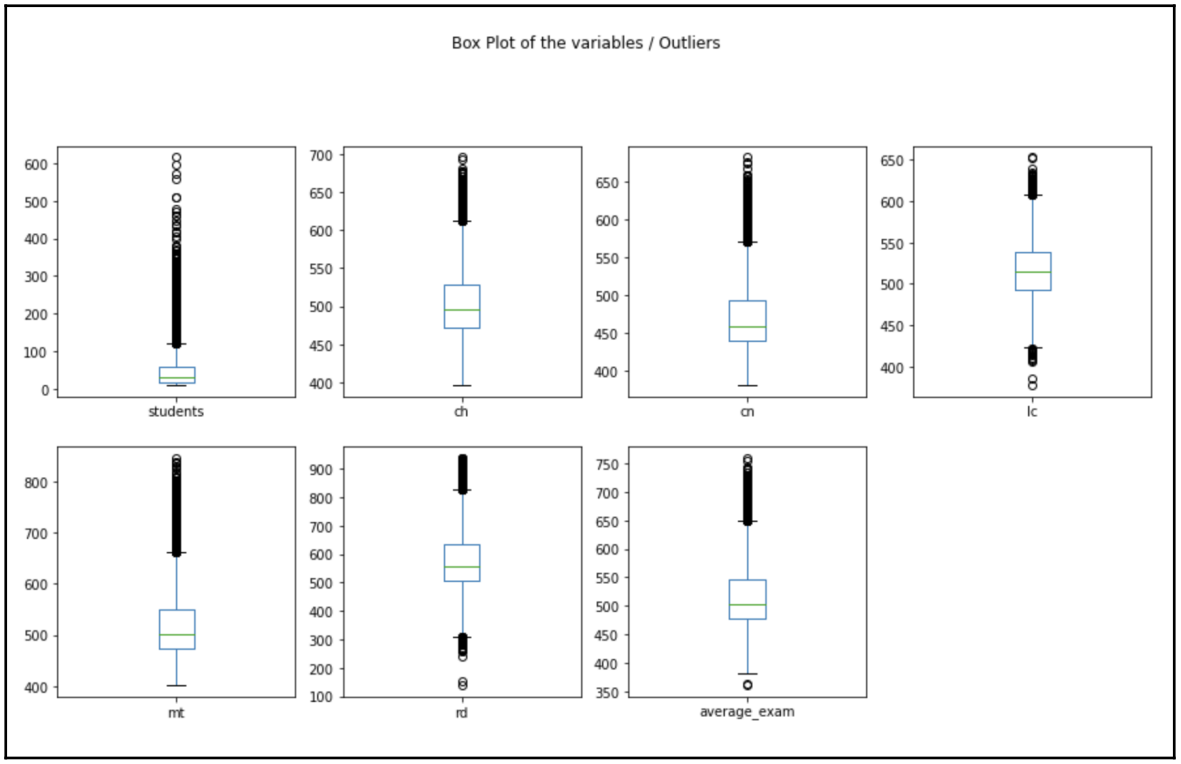
```
#Looking for missing values
enem.isna().sum()
```

```
rank            0
inep_code       0
school          0
state           0
city            0
school_type     0
location        0
students        0
ch              0
cn              0
lc              0
mt              0
rd              0
average_exam    0
dtype: int64
```

2.3.5 Outliers

It is always important to understand if the dataset has outliers. For this project, I decided to look for them using a box plot. According to Galarnyk (2018), everything that was plotted out of the whiskers as points is considered outliers.

Enem - Outliers



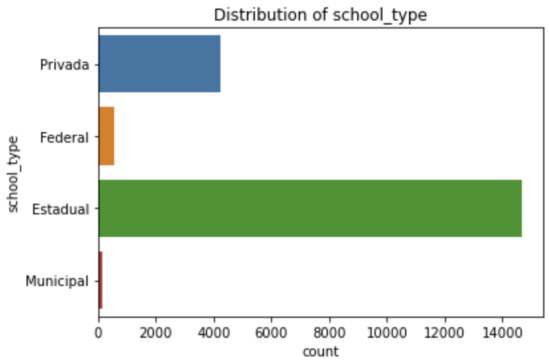
2.4 EDA Visualizations

2.4.1 Showing categorical variables

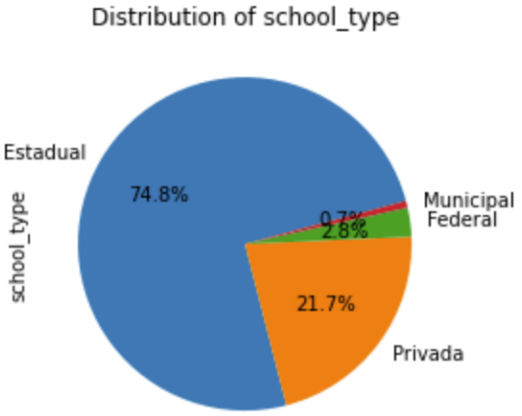
According to Yi (2021), a bar chart is one of the best ways to demonstrate a distribution of data points in a categorical feature. In other words, it will give a better idea of some characteristics of these features, such as their frequency and mode.

I am also using value_counts() since it is important to know the real number that the plots are representing.

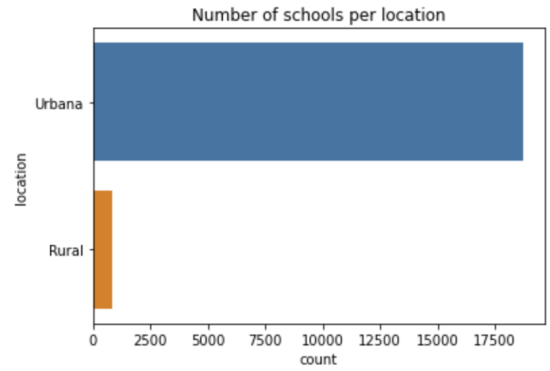
School_type



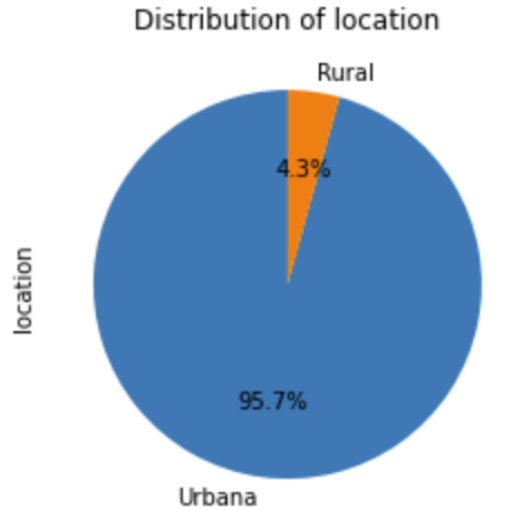

```
Estadual     14666
Privada       4252
Federal        547
Municipal      133
Name: school_type, dtype: int64
```



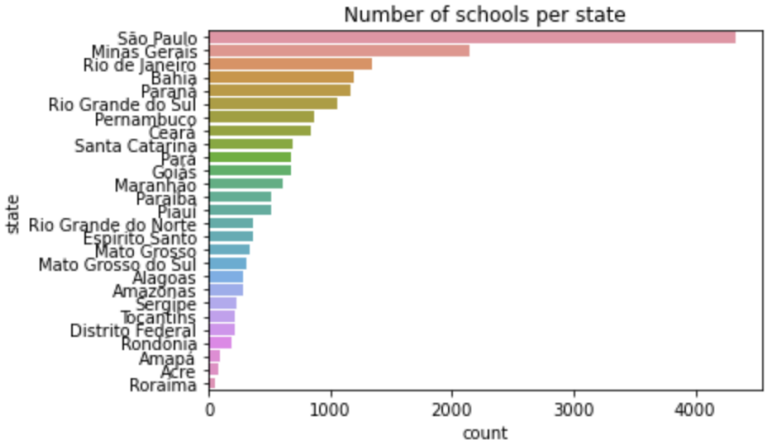
Location




```
Urbana    18752
Rural       846
Name: location, dtype: int64
```



State

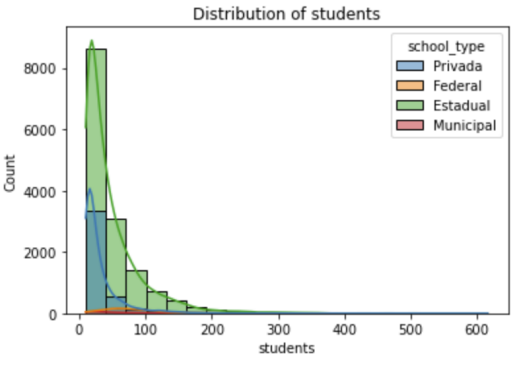



```
São Paulo                4330
Minas Gerais             2143
Rio de Janeiro           1346
Bahia                    1193
Paraná                   1167
Rio Grande do Sul        1059
Pernambuco                861
Ceará                     840
Santa Catarina            687
Pará                      676
Goiás                     671
Maranhão                  602
Paraíba                   512
Piauí                     508
Rio Grande do Norte       368
Espírito Santo            366
Mato Grosso               339
Mato Grosso do Sul        310
Alagoas                   284
Amazonas                  282
Sergipe                   226
Tocantins                 219
Distrito Federal          214
Rondônia                  181
Amapá                      86
Acre                       82
Roraima                    46
Name: state, dtype: int64
```

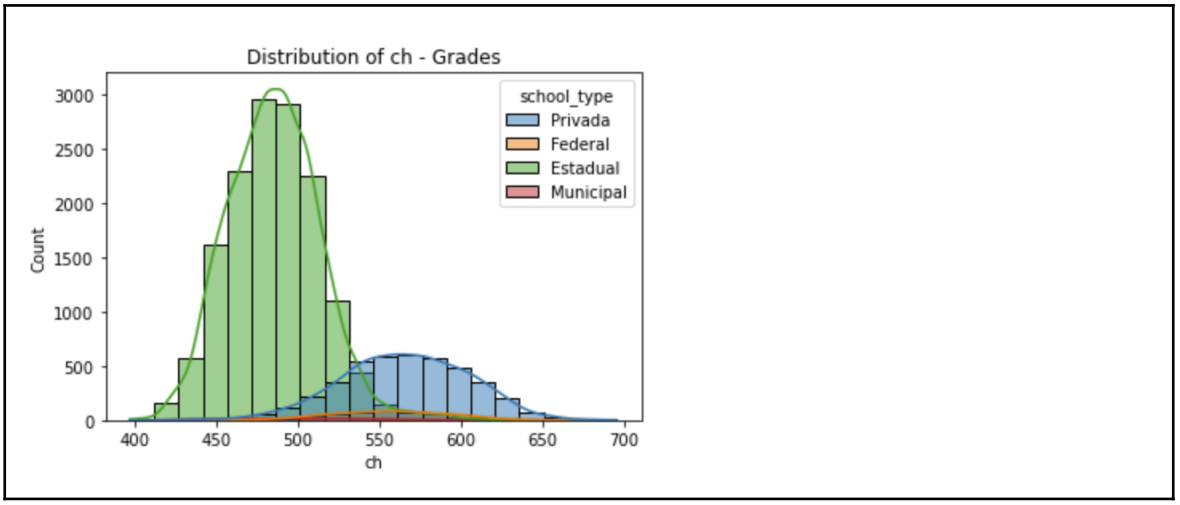
2.4.2 Showing Numerical Variables

I am going to plot the distribution of the numerical features. I also decided to use hue=”school_type” since it gives me a better idea of the situation and differences between public and private schools in Brazil.

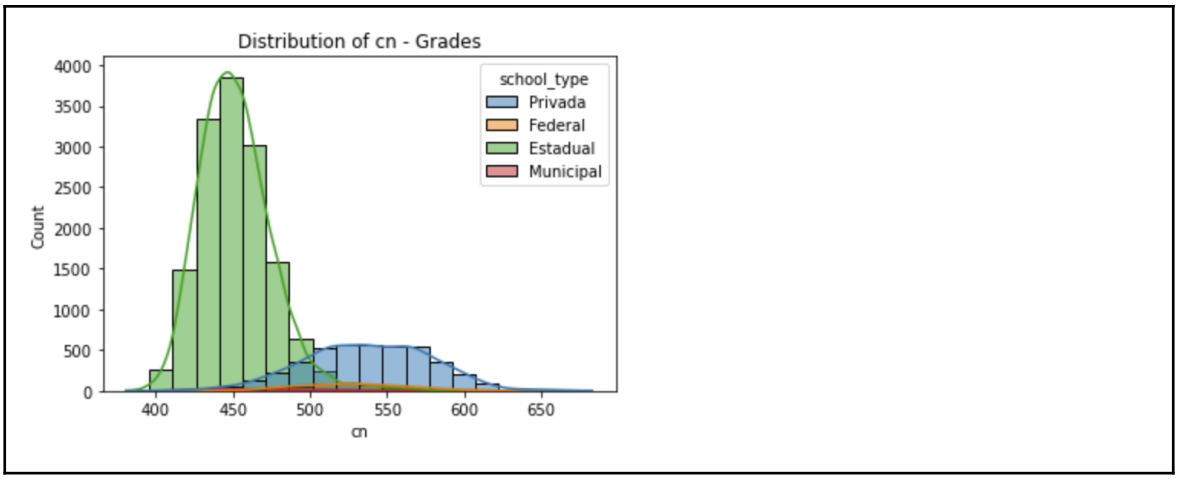
Students



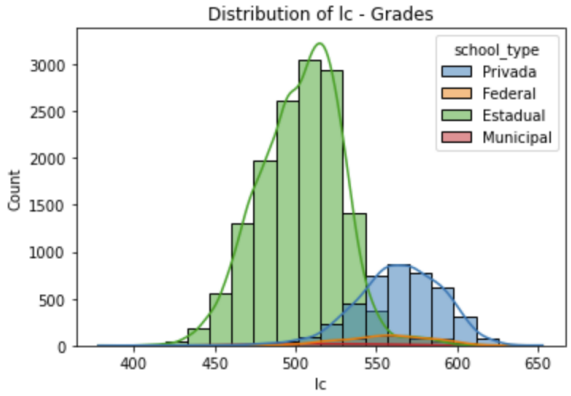
ch


Distribution of ch - Grades
school_type
Privada
Federal
Estadual
Municipal
Count
ch
0
500
1000
1500
2000
2500
3000
400
450
500
550
600
650
700

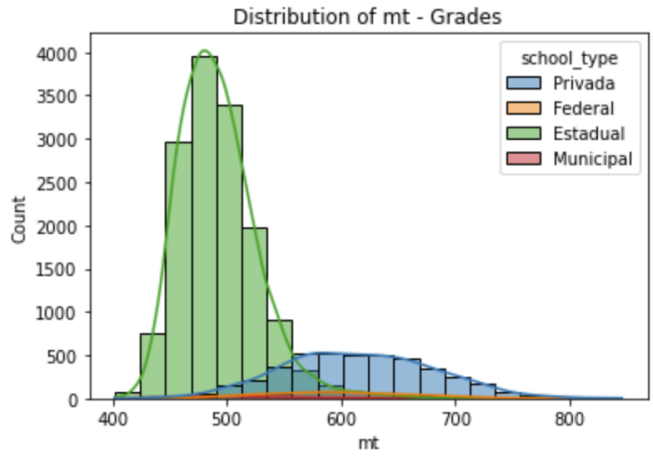
cn


Distribution of cn - Grades
school_type
Privada
Federal
Estadual
Municipal
Count
cn
0
500
1000
1500
2000
2500
3000
3500
4000
400
450
500
550
600
650

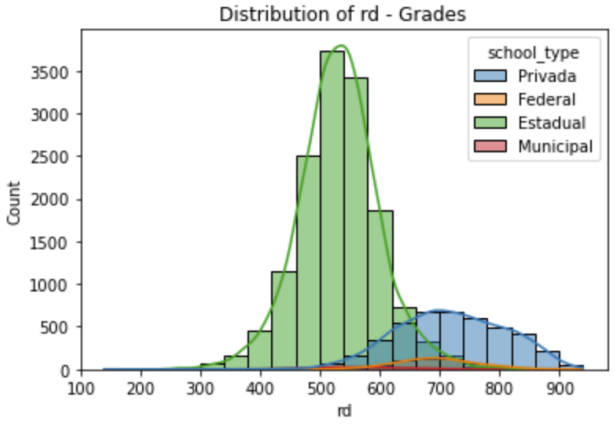
lc


Distribution of lc - Grades
school_type
Privada
Federal
Estadual
Municipal
Count
lc
0
500
1000
1500
2000
2500
3000
400
450
500
550
600
650

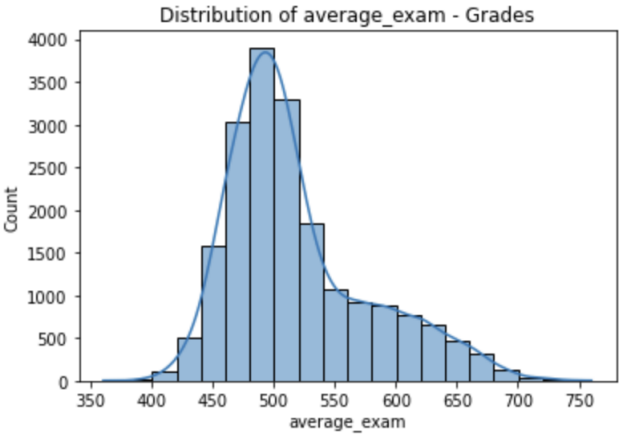
mt


Distribution of mt - Grades
school_type
Privada
Federal
Estadual
Municipal
Count
mt
0
500
1000
1500
2000
2500
3000
3500
4000
400
500
600
700
800

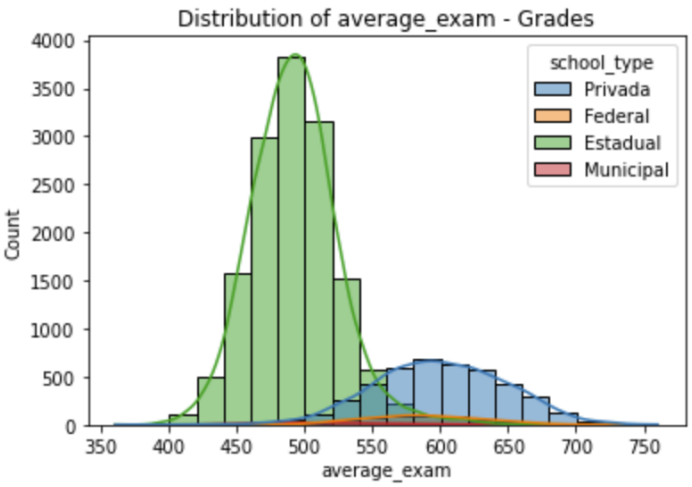
rd



average_exam



average_exam with school_type



2.4.3 General overview

Dataset statistics

Dataset statistics

Number of variables	14
Number of observations	19598
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	2.1 MiB
Average record size in memory	112.0 B

Variable types

Numeric	9
Categorical	5

Later in this project, the features average_exam, lc and mt will be used to illustrate different scenarios. Keeping this in mind, I decided to show some statistical details of these features.

Statistical details - average_exam

Distinct	12102
Distinct (%)	61.8%
Missing	0
Missing (%)	0.0%
Infinite	0
Infinite (%)	0.0%
Mean	518.9176799

Minimum	360.46
Maximum	760.18
Zeros	0
Zeros (%)	0.0%
Negative	0
Negative (%)	0.0%
Memory size	153.2 KiB

Quantile statistics

Minimum	360.46
5-th percentile	447.087
Q1	478.46
median	503.76
Q3	547.03
95-th percentile	640.689
Maximum	760.18
Range	399.72
Interquartile range (IQR)	68.57

Descriptive statistics

Standard deviation	58.75436549
Coefficient of variation (CV)	0.1132248289
Kurtosis	0.455880344
Mean	518.9176799
Median Absolute Deviation (MAD)	30.235
Skewness	0.9711445232
Sum	10169748.69
Variance	3452.075464
Monotonicity	Decreasing

Statistical details - mt

Distinct	12225
Distinct (%)	62.4%
Missing	0
Missing (%)	0.0%
Infinite	0
Infinite (%)	0.0%
Mean	522.2674252

Minimum	401.86
Maximum	845.89
Zeros	0
Zeros (%)	0.0%
Negative	0
Negative (%)	0.0%
Memory size	153.2 KiB

Quantile statistics

Minimum	401.86
5-th percentile	447.9
Q1	474.5225
median	501.55
Q3	548.925
95-th percentile	670.1735
Maximum	845.89
Range	444.03
Interquartile range (IQR)	74.4025

Descriptive statistics

Standard deviation	68.62626008
Coefficient of variation (CV)	0.1314006135
Kurtosis	1.289266287
Mean	522.2674252
Median Absolute Deviation (MAD)	32.65
Skewness	1.304488159
Sum	10235397
Variance	4709.563573
Monotonicity	Not monotonic

Statistical details - lc

Distinct	10413	**Minimum**	377.89
Distinct (%)	53.1%	**Maximum**	652.91
Missing	0	**Zeros**	0
Missing (%)	0.0%	**Zeros (%)**	0.0%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	517.8462991	**Memory size**	153.2 KiB

Quantile statistics

Minimum	377.89
5-th percentile	462.897
Q1	492.25
median	514.66
Q3	538.66
95-th percentile	587.136
Maximum	652.91
Range	275.02
Interquartile range (IQR)	46.41

Descriptive statistics

Standard deviation	36.83975094
Coefficient of variation (CV)	0.07114031906
Kurtosis	-0.166468946
Mean	517.8462991
Median Absolute Deviation (MAD)	23.07
Skewness	0.3845713594
Sum	10148751.77
Variance	1357.167249
Monotonicity	Not monotonic

3. DATA PREPARATION

Data preparation may be explained as the process of gathering, combining and organizing data in a way where it can be used in business intelligence and/or data visualization applications. According to Pearlman (2018), data preparation is mainly the process of cleaning and transforming raw data. Before starting to make changes in the dataset, I am going to create a copy.

Making a copy - ENEM

```
# copying df
df_preparing = enem.copy()
df_preparing.head(3)
```

3.1 Dropping unnecessary columns

It is important to know and understand what the dataset is about, in order to decide how to deal with the features that do not contribute to the dataset analysis. Keeping this in mind, I decided to drop the features “rank”, “inep_code”, “school” and “city”. I decided to drop these features because for this project, they either do not pass any relevant information, or they pass a lot of information that is not relevant for the analysis.

Dropping unnecessary columns

```
1 # Dropping unnecessary columns
2 df_preparing.drop(["rank", "inep_code", 'school', 'city'], axis=1, inplace=True)
3 df_preparing.head(3)
```

	state	school_type	location	students	ch	cn	lc	mt	rd	average_exam
0	Ceará	Privada	Urbana	35	692.85	674.50	652.24	845.89	935.43	760.18
1	Ceará	Privada	Urbana	33	695.67	676.34	652.91	836.65	915.15	755.34
2	Minas Gerais	Privada	Urbana	280	681.58	668.20	634.47	823.80	906.64	742.94

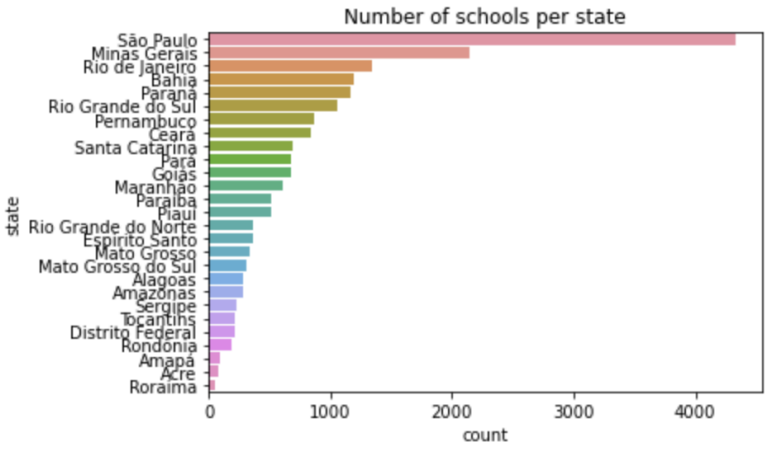
3.2 Dealing with Outliers

As it was possible to confirm during previous steps, there are some outliers in this dataset. However, I decided that they are important for the analysis, and consequently, I decided to keep them in the dataset. I took this decision because after the EDA stage I could realize that part of the data that are shown as outliers in the boxplot are related to private schools, since their performance is better when compared with the majority of the schools in the dataset.

3.3 Separating the 26 states and DF

Let me show in a graph and in numbers the number of schools per state:

ENEM - number of schools




```
São Paulo              4330
Minas Gerais           2143
Rio de Janeiro         1346
Bahia                  1193
Paraná                 1167
Rio Grande do Sul      1059
Pernambuco              861
Ceará                   840
Santa Catarina          687
Pará                    676
Goiás                   671
Maranhão                602
Paraíba                 512
Piauí                   508
Rio Grande do Norte     368
Espirito Santo          366
Mato Grosso             339
Mato Grosso do Sul      310
Alagoas                 284
Amazonas                282
Sergipe                 226
Tocantins               219
Distrito Federal        214
Rondônia                181
Amapá                    86
Acre                     82
Roraima                  46
Name: state, dtype: int64
```

In order to better understand the education system in general, let me show which states in Brazil performed better in ENEM 2019. To start off with, I decided to create variables containing only the information about a specific state. I called each variable by its acronym. For example, Rio de Janeiro is rj.

Creating the state variables

```
# Separating the 26 states and DF

for states in enem["state"]:

    if states == "São Paulo":
        sp = enem.loc[enem["state"]=="São Paulo"]

    elif states == "Minas Gerais":
        mg = enem.loc[enem["state"]=="Minas Gerais"]

    elif states == "Rio de Janeiro":
        rj = enem.loc[enem["state"]=="Rio de Janeiro"]

    elif states == "Bahia":
        ba = enem.loc[enem["state"]=="Bahia"]

    elif states == "Paraná":
        pr = enem.loc[enem["state"]=="Paraná"]


    elif states == "Rio Grande do Sul":
        rs = enem.loc[enem["state"]=="Rio Grande do Sul"]


    elif states == "Pernambuco":
        pe = enem.loc[enem["state"]=="Pernambuco"]


    elif states == "Ceará":
        ce = enem.loc[enem["state"]=="Ceará"]


    elif states == "Santa Catarina":
        sc = enem.loc[enem["state"]=="Santa Catarina"]


    elif states == "Pará":
        pa = enem.loc[enem["state"]=="Pará"]
```

```
elif states == "Goiás":
    go = enem.loc[enem["state"]=="Goiás"]


elif states == "Maranhão":
    ma = enem.loc[enem["state"]=="Maranhão"]


elif states == "Paraíba":
    pb = enem.loc[enem["state"]=="Paraíba"]


elif states == "Piauí":
    pi = enem.loc[enem["state"]=="Piauí"]


elif states == "Rio Grande do Norte":
    rn = enem.loc[enem["state"]=="Rio Grande do Norte"]


elif states == "Espírito Santo":
    es = enem.loc[enem["state"]=="Espírito Santo"]


elif states == "Mato Grosso":
    mt = enem.loc[enem["state"]=="Mato Grosso"]


elif states == "Mato Grosso do Sul":
    ms = enem.loc[enem["state"]=="Mato Grosso do Sul"]


elif states == "Alagoas":
    al = enem.loc[enem["state"]=="Alagoas"]


elif states == "Amazonas":
    am = enem.loc[enem["state"]=="Amazonas"]


elif states == "Sergipe":
    se = enem.loc[enem["state"]=="Sergipe"]
```

```
elif states == "Tocantins":
    to = enem.loc[enem["state"]=="Tocantins"]


elif states == "Distrito Federal":
    df = enem.loc[enem["state"]=="Distrito Federal"]


elif states == "Rondônia":
    ro = enem.loc[enem["state"]=="Rondônia"]


elif states == "Amapá":
    ap = enem.loc[enem["state"]=="Amapá"]


elif states == "Acre":
    ac = enem.loc[enem["state"]=="Acre"]


elif states == "Roraima":
    rr = enem.loc[enem["state"]=="Roraima"]
```

Enem - RJ

	rank	inep_code	school	state	city	school_type	location	students	ch	cn	lc	mt	rd	average_exam
12	13	33135827	COLEGIO E CURSO PENSI	Rio de Janeiro	Rio de Janeiro	Privada	Urbana	31	658.80	645.99	619.74	783.69	920.65	725.77
21	22	33040516	COLEGIO IPIRANGA	Rio de Janeiro	Petrópolis	Privada	Urbana	27	655.66	633.59	619.04	814.72	879.26	720.45
22	23	33178879	COLEGIO ALFA CEM BILINGUE	Rio de Janeiro	Rio de Janeiro	Privada	Urbana	13	646.41	621.19	617.39	798.65	918.46	720.42
23	24	33062633	COL DE SAO BENTO	Rio de Janeiro	Rio de Janeiro	Privada	Urbana	45	676.73	631.62	618.65	790.36	882.22	719.92
37	38	33145237	COLEGIO SAO JOAO BATISTA NOVA FRIBURGO	Rio de Janeiro	Nova Friburgo	Privada	Urbana	24	648.63	618.02	617.99	767.24	920.83	714.54

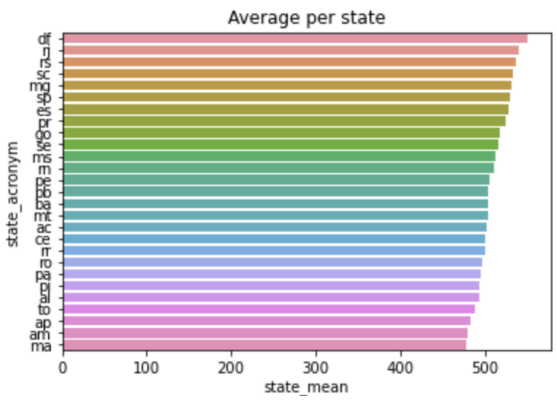
After having isolated the data by state, I simply calculated the mean of average_exam of each state. Let me show the code I used.

Python - Calculating the mean of every state

```
# Average of each Brazilian state
average_state = pd.DataFrame(
    {'state_acronym': ["se", "ms", "rn",
                       "pe", "pb", "ba",
                       "mt", "ac", "ce",

                       "rr", "ro", "pa",
                       "pi", "al", "to",
                       "ap", "am", "ma",

                       "df", "rj", "rs",
                       "sc", "mg", "sp",
                       "es", "pr", "go"],




                   'state_mean': [
                               se["average_exam"].mean(), ms["average_exam"].mean(), rn["average_exam"].mean(),
                               pe["average_exam"].mean(), pb["average_exam"].mean(), ba["average_exam"].mean(),
                               mt["average_exam"].mean(), ac["average_exam"].mean(), ce["average_exam"].mean(),

                               rr["average_exam"].mean(), ro["average_exam"].mean(), pa["average_exam"].mean(),
                               pi["average_exam"].mean(), al["average_exam"].mean(), to["average_exam"].mean(),
                               ap["average_exam"].mean(), am["average_exam"].mean(), ma["average_exam"].mean(),

                               df["average_exam"].mean(), rj["average_exam"].mean(), rs["average_exam"].mean(),
                               sc["average_exam"].mean(), mg["average_exam"].mean(), sp["average_exam"].mean(),
                               es["average_exam"].mean(), pr["average_exam"].mean(), go["average_exam"].mean()
                ]
    })
average_state
```

Finally, it is possible to compare the state's performance based on the average grade of each state. Let me show it first in a plot.

Enem - mean per state



And now using numbers.

Enem - mean per state - numbers

	state_acronym	state_mean
0	df	551.316121
1	rj	539.492771
2	rs	536.958546
3	sc	533.276114
4	mg	531.799356
5	sp	530.511649
6	es	528.293415
7	pr	525.112322
8	go	517.949747
9	se	515.955398
10	ms	512.733710
11	rn	510.499239
12	pe	505.481370
13	pb	504.398945
14	ba	503.632590
15	mt	503.308437
16	ac	501.972195
17	ce	501.190774
18	rr	500.825652
19	ro	496.570663
20	pa	495.894068
21	pi	493.808681
22	al	492.961796
23	to	487.997260
24	ap	482.650233
25	am	478.910745
26	ma	477.088007

Returning to the reason why I chose this dataset to work with, I can see now that maybe, being the second richest state in Brazil gives some advantage for the students. I could note that Rio is the third state with more schools taking the exam. Besides that, Rio had the second best general average in Brazil, according to EVOLUCIONAL. Or the best one if we do not count DF, since DF is just the capital of Brazil. Besides that I could see that having more schools does not necessarily make a state perform better in ENEM. São Paulo is an example, since it is the state with more schools but it was only the sixth considering the performance in this exam.

3.4 Encoding

According to Team (2020), one good practice for data science projects is to encode categorical variables in order to transform them into a numerical variable. For this project, I used get_dummies since I would like to encode just two features, school_type and location.

Encoding

```
# Encoding the categorical variables
df_encoded = pd.get_dummies(df_preparing, columns = ['school_type', 'location'])
df_encoded.head(3)
```

lc	mt	rd	average_exam	school_type_Estadual	school_type_Federal	school_type_Municipal	school_type_Privada	location_Rural	location_Urbana
652.24	845.89	935.43	760.18	0	0	0	1	0	1
652.91	836.65	915.15	755.34	0	0	0	1	0	1
634.47	823.80	906.64	742.94	0	0	0	1	0	1

4. FIRST SECTION

4.1 Parameters Of The Population - Research

According to G1 (2020a), in 2019 Minas Gerais, Ceará and Rio de Janeiro were the Brazilian States where, in general, it was possible to find the students with the highest average grade in ENEM.

Taking into consideration the schools' average and the dataset used for this project, Ceará and Minas Gerais have 2 and 5 schools, respectively, among the best 10 schools in Brazil.

As it is possible to see below, the best school average is 760.18 and the worst school average is 360.46. According to this dataset, the mean score of the average grade of Brazilian schools in 2019 was 518.91.

Best 10 schools - Brazil

	rank	inep_code	school	state	city	school_type	location	students	ch	cn	lc	mt	rd	average_exam
0	1	23246847	FARIAS BRITO COLEGIO DE APLICACAO	Ceará	Fortaleza	Privada	Urbana	35	692.85	674.50	652.24	845.89	935.43	760.18
1	2	23246871	ARI DE SA CAVALCANTE SEDE MARIO MAMEDE COLEGIO	Ceará	Fortaleza	Privada	Urbana	33	695.67	676.34	652.91	836.65	915.15	755.34
2	3	31350664	COLEGIO BERNOULLI	Minas Gerais	Belo Horizonte	Privada	Urbana	280	681.58	668.20	634.47	823.80	906.64	742.94
3	4	35399197	OBJETIVO COLEGIO INTEGRADO	São Paulo	São Paulo	Privada	Urbana	53	677.45	682.90	638.72	836.11	868.68	740.77
4	5	31349720	FIBONACCI COLEGIO	Minas Gerais	Ipatinga	Privada	Urbana	57	666.41	657.87	630.84	809.55	938.95	740.72
5	6	31351725	COLEGUIUM	Minas Gerais	Belo Horizonte	Privada	Urbana	33	676.18	646.74	631.78	817.78	926.67	739.83
6	7	31004812	COLEGIO SANTO ANTONIO	Minas Gerais	Belo Horizonte	Privada	Urbana	142	665.88	658.94	625.22	830.47	893.38	734.78
7	8	31128074	COL DE APLICACAO DA UFV - COLUNI	Minas Gerais	Viçosa	Federal	Urbana	153	674.53	647.35	619.84	807.24	916.47	733.09
8	9	35463279	VITAL BRAZIL COLEGIO	São Paulo	São Paulo	Privada	Urbana	54	665.11	656.84	623.58	787.05	917.04	729.92
9	10	35141240	VERTICE COLEGIO UNIDADE II	São Paulo	São Paulo	Privada	Urbana	43	674.52	657.56	629.40	785.75	894.42	728.33

Worst 3 schools - Brazil

	rank	inep_code	school	state	city	school_type	location	students	ch	cn	lc	mt	rd	average_exam
19595	19596	13252208	ESC EST INDIGENA PROFESSOR GILDO SAMPAIO MEGAT...	Amazonas	Benjamin Constant	Estadual	Rural	56	422.24	407.61	411.16	425.47	241.79	381.65
19596	19597	13008196	ESCOLA ESTADUAL ALMIRANTE TAMANDARE	Amazonas	Tabatinga	Estadual	Rural	21	432.40	405.50	377.89	441.75	154.29	362.36
19597	19598	23071265	INSTITUTO CEARENSE DE EDUCACAO DE SURDOS	Ceará	Fortaleza	Estadual	Urbana	18	421.57	414.88	406.04	420.93	138.89	360.46

5 number summary - ENEM - average_exam

```
# Using .describe() the show the 5 number summary
enem["average_exam"].describe()
```

```
count    19598.000000
mean       518.917680
std         58.754365
min        360.460000
25%        478.460000
50%        503.760000
75%        547.030000
max        760.180000
Name: average_exam, dtype: float64
```

However, as previously said, this dataset only contains the data of the schools registered at EVOLUCIONAL. In other words, there is no data in this dataset of the people that do not go to school anymore and the students that do not study at a registered school.

In order to have a real picture of ENEM 2019, I decided to research its results. According to G1 (2020b), 3,709,809 people did ENEM in 2019 and the average grade of the exam in 2019 was 524.54.

Considering Rio de Janeiro, there are 1346 schools in this dataset. When we compare schools in Brazil, the best school in RJ is seen in the 13th position. The school with the lowest average grade is seen in the 19516th position.

Best 3 schools - Rio de Janeiro

	rank	inep_code	school	state	city	school_type	location	students	ch	cn	lc	mt	rd	average_exam
12	13	33135827	COLEGIO E CURSO PENSI	Rio de Janeiro	Rio de Janeiro	Privada	Urbana	31	658.80	645.99	619.74	783.69	920.65	725.77
21	22	33040516	COLEGIO IPIRANGA	Rio de Janeiro	Petrópolis	Privada	Urbana	27	655.66	633.59	619.04	814.72	879.26	720.45
22	23	33178879	COLEGIO ALFA CEM BILINGUE	Rio de Janeiro	Rio de Janeiro	Privada	Urbana	13	646.41	621.19	617.39	798.65	918.46	720.42

Worst 3 schools - Rio de Janeiro

	rank	inep_code	school	state	city	school_type	location	students	ch	cn	lc	mt	rd	average_exam
19461	19462	33060185	CIEP 383 MAXIMO GORKI	Rio de Janeiro	Nova Iguaçu	Estadual	Urbana	13	451.18	448.76	452.81	442.51	309.23	420.90
19471	19472	33099898	CIEP 346 BELARMINO ALFREDO DOS SANTOS	Rio de Janeiro	Queimados	Estadual	Urbana	13	428.18	434.09	467.86	430.05	340.00	420.04
19515	19516	33005362	CE WALDEMIRO PITA	Rio de Janeiro	Cambuci	Estadual	Rural	11	447.06	400.59	455.21	406.40	363.64	414.58

Considering all this information, and my background as a teacher in Rio de Janeiro, I decided to implement a test comparing the average of the schools in RJ regarding ENEM.

4.2 Choosing the Variable

For this section, I decided to perform a Hypothesis Test in the variable “average_exam”. However, as my intention is to analyze the situation in RJ, I am going to use “average_exam” just taking into consideration the schools in RJ.

Considering all the information above regarding ENEM in Brazil, ENEM according to this dataset and my teaching experience, it is possible to formulate a hypothesis.

4.3 Hypothesis Test - Schools in RJ

Some schools in RJ claim that the mean score of the schools in RJ in ENEM 2019 is equal or greater than the national average, which is 524.54. A random sample of the average grade of 25 schools in RJ will be taken. Is there enough evidence to support the schools' claim at a 5% significance level?

Considering the situation above, it is possible to write the null hypothesis and the alternative hypothesis.

null hypothesis and alternative hypothesis

H_0: μ >= 524.54
H_1: μ < 524.54

4.3.1 Choosing The Test

According to Hayes (2021), t-test is a tool that can be used to test an assumption applicable to a population. It is mainly a statistical test used to compare the means of two groups.

In the hypothesis created above, the standard deviation of the population is unknown. Besides that, the sample size is 25 (that is, less than 30).

Considering all that, in this situation the t-test is more appropriate.

Moreover, I am going to perform a lower-tailed test. As I would like to know if one population mean is less than the other, this one-tailed test is more appropriate. It is important to highlight that in this case, the "reject region" is in the left tail.

4.3.2 Doing The Test

To start off with, let me organize all the information:

T- test formula	$t = \frac{\bar{x} - \mu}{s/\sqrt{n}}$

Mean of the sample	`# Finding the mean of the sample`
	`np.mean(rj_sample)`
	`537.2815999999999`

Mean of the population	524.54

Standard deviation of the sample	`# Finding the standard deviation of the sample`
	`np.std(rj_sample)`
	`58.24748438722482`

Data sample size	25
Degrees of freedom	24
Level of confidence	95%
Alpha	5%

After having everything set, it is now possible to find the t-test statistic.

T-test

Preparing the formula - RJ

```
# mean of the sample
x1_rj = np.mean(rj_sample)
```

```
# mean of the population
X2 = 524.54
```

```
# standard deviation of the sample
s_rj = np.std(rj_sample)
```

```
# squate root
n_sqrt_rj= math.sqrt(25)
n_sqrt_rj
```

```
5.0
```

t-test (RJ)

```
# t-test - RJ
t_rj = (x1_rj-X2)/(s_rj/n_sqrt_rj)
t_rj
```

```
1.093746805895922
```

4.3.3 Understanding the results

First it is necessary to check the table of critical values of t. As the degree of freedom is 24 and I am performing a lower-tailed test, the critical value found was -1.711.

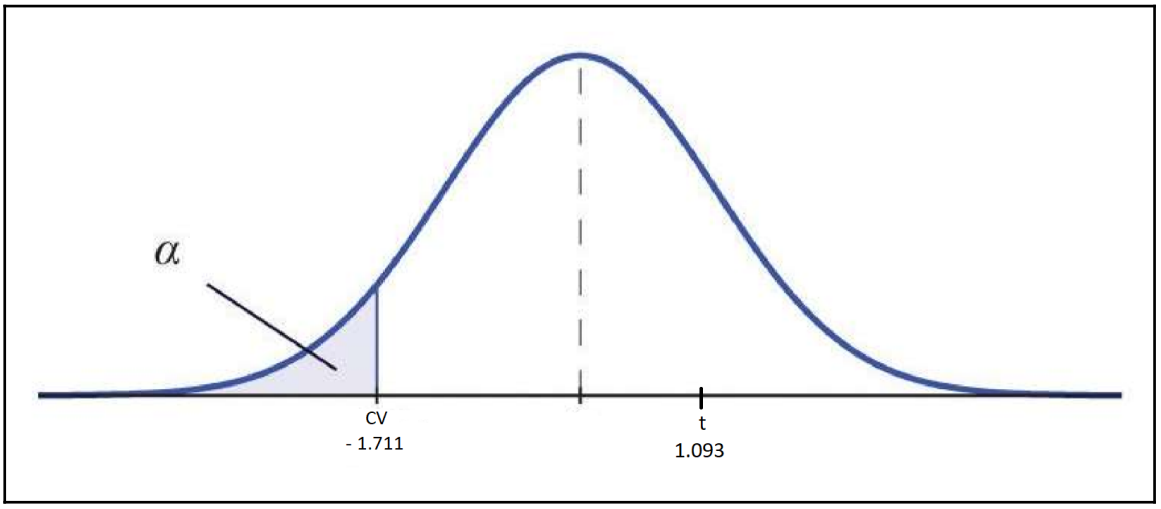
Critical values of t

Degrees of freedom	*Two-tailed test:* / *One-tailed test:* 10% / 5%
1	6.314
2	2.920
3	2.353
4	2.132
5	2.015
6	1.943
7	1.894
8	1.860
9	1.833
10	1.812
11	1.796
12	1.782
13	1.771
14	1.761
15	1.753
16	1.746
17	1.740
18	1.734
19	1.729
20	1.725
21	1.721
22	1.717
23	1.714
24	1.711
25	1.708

The t test statistic value is 1.093. As 1.093 > -1.711, it means that the value falls inside the range. In other words, we do not have enough evidence at the 95% level of confidence to reject the null hypothesis. That is, the average *mean score of the schools in RJ in ENEM 2019 is equal or greater than the national average.*

4.3.4 Understanding the results - Plots

T- test (lower-tailed test)

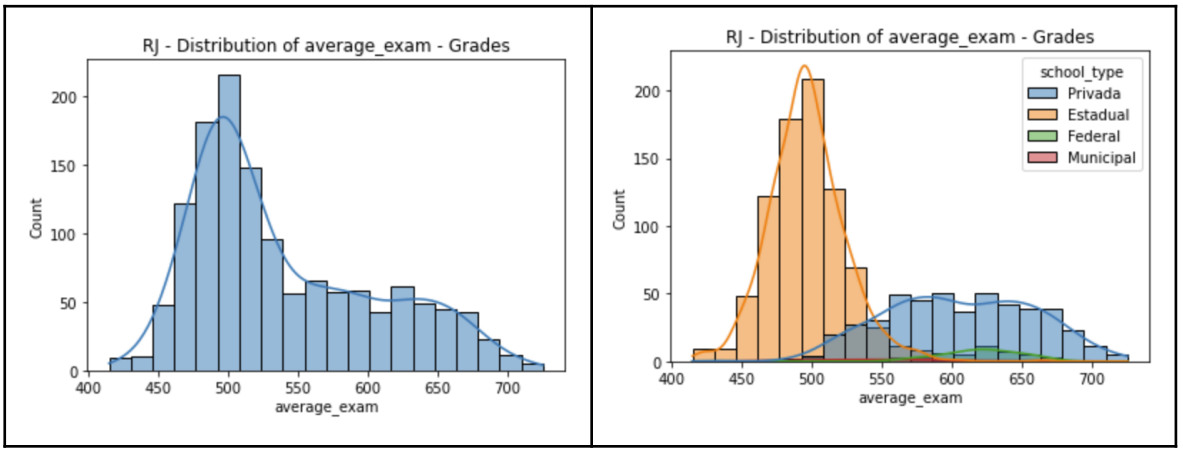


In the plot above it is possible to confirm the 4.3.3 affirmation.

4.4 The real situation of schools in RJ

Even though it is possible to say that, in general, RJ schools may have a good performance in ENEM, as a person and teacher who studied and taught in RJ, I know that it is impossible to compare the situation and investment between public and private schools. As it is possible to see below, basically the public schools are on the left and the private schools are on the right.

Schools in RJ

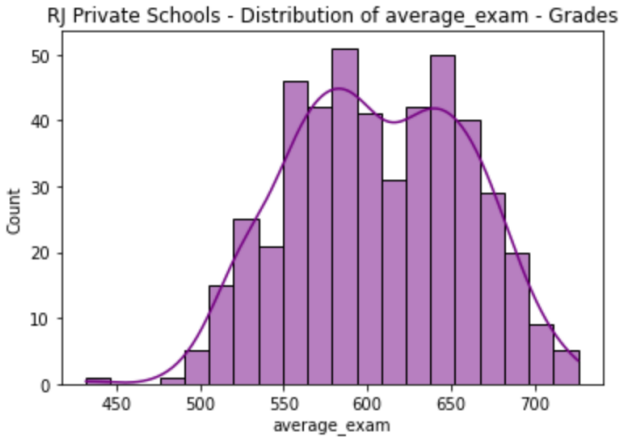


Keeping that in mind I decided to repeat the test 2 more times; however, in the first test I am going to consider only the private schools. In the second test, I am going to consider only the schools that are administered (run) by the State of RJ (government).

In order to have it done, I separated the data into “rj_private_schools” and “rj_state_schools”. It is possible to see below that the distribution of these new variables is closer to what can be called “symmetric distribution”, even though it is not a perfect symmetric distribution.

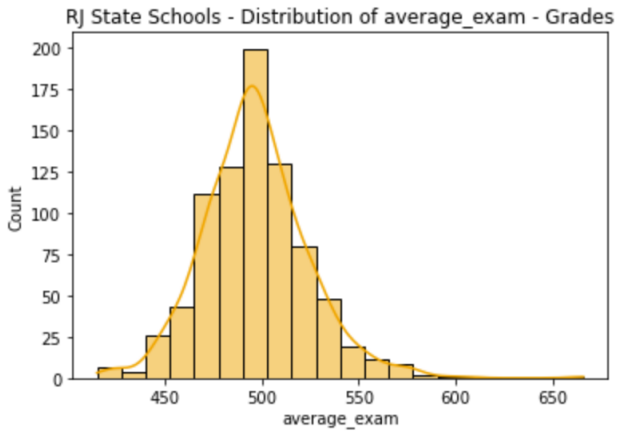
RJ - Private Schools

```
# Creating the variable
rj_private_schools = rj.loc[rj["school_type"]=="Privada"]
```



RJ - State Schools

```
# Creating the variable
rj_state_schools = rj.loc[rj["school_type"]=="Estadual"]
```



4.5 Hypothesis Test - Private Schools in RJ

Private *schools in RJ claim that the mean score of their schools in ENEM 2019 is equal or greater than the national average, which is 524.54. A random sample of the average grade of 25* ***private*** *schools in RJ will be taken. Is there enough evidence to support the* ***private*** *schools' claim at a 0.05 significance level?*

Considering the situation above, it is possible to write the null hypothesis and the alternative hypothesis.

null hypothesis and alternative hypothesis - Private Schools in RJ

H_0: μ >= 524.54
H_1: μ < 524.54

4.5.1 Choosing The Test - Private Schools in RJ

The scenario is similar to the previous case. However, I decided to change the level of confidence.

In the hypothesis created above, the standard deviation of the population is unknown. Besides that, the sample size is 25 (that is, less than 30).

Considering all that, in this situation the t-test is more appropriate.

Moreover, I am going to perform a lower-tailed test. As I would like to know if one population mean is less than the other, this one-tailed test is more appropriate. It is important to highlight that in this case, the "reject region" is in the left tail.

4.5.2 Doing The Test - Private Schools in RJ

To start off with, let me organize all the information:

T- test formula	$t = \frac{\bar{x} - \mu}{s/\sqrt{n}}$

Mean of the sample	`# Finding the mean of the sample`
	`np.mean(rj_private_sample)`
	`593.2352000000001`

Mean of the population	524.54

Standard deviation of the sample	`# Finding the standard deviation of the sample`
	`np.std(rj_private_sample)`
	`53.92640919772055`

Data sample size	25
Degrees of freedom	24
Level of confidence	99.95%
Alpha	0.05%

After having everything set, it is now possible to find the t-test statistic.

T-test

Preparing the formula - RJ Private Schools

```
# mean of the sample
x1_rj_private = np.mean(rj_private_sample)
```

```
# mean of the population
X2 = 524.54
```

```
# standard deviation of the sample
s_rj_private = np.std(rj_private_sample)
```

```
# squate root
n_sqrt_rj= math.sqrt(25)
n_sqrt_rj
```

```
5.0
```

t-test (RJ Private Schools)

```
# t-test - RJ Private
t_rj_private = (x1_rj_private-X2)/(s_rj_private/n_sqrt_rj)
t_rj_private
```

```
6.369346765527254
```

4.5.3 Understanding the results - Private Schools in RJ

First it is necessary to check the table of critical values of t. As the degree of freedom is 24 and I am performing a lower-tailed test, the critical value found was -3.745.

Critical values of t

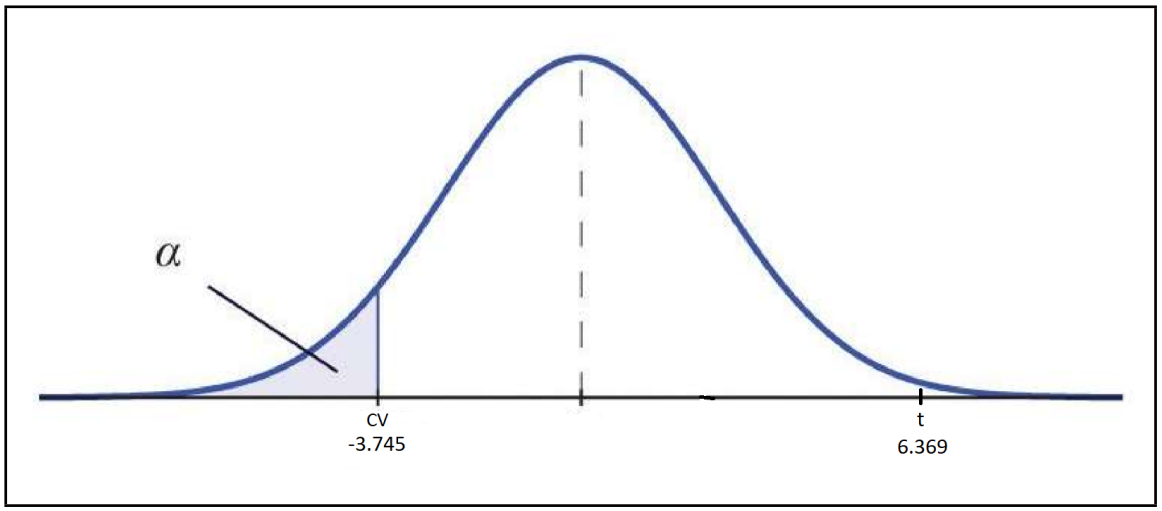
t Distribution: Critical Values of t

		Significance level					
Degrees of freedom	*Two-tailed test:* / *One-tailed test:*	10% / 5%	5% / 2.5%	2% / 1%	1% / 0.5%	0.2% / 0.1%	0.1% / 0.05%
1		6.314	12.706	31.821	63.657	318.309	636.619
2		2.920	4.303	6.965	9.925	22.327	31.599
3		2.353	3.182	4.541	5.841	10.215	12.924
4		2.132	2.776	3.747	4.604	7.173	8.610
5		2.015	2.571	3.365	4.032	5.893	6.869
6		1.943	2.447	3.143	3.707	5.208	5.959
7		1.894	2.365	2.998	3.499	4.785	5.408
8		1.860	2.306	2.896	3.355	4.501	5.041
9		1.833	2.262	2.821	3.250	4.297	4.781
10		1.812	2.228	2.764	3.169	4.144	4.587
11		1.796	2.201	2.718	3.106	4.025	4.437
12		1.782	2.179	2.681	3.055	3.930	4.318
13		1.771	2.160	2.650	3.012	3.852	4.221
14		1.761	2.145	2.624	2.977	3.787	4.140
15		1.753	2.131	2.602	2.947	3.733	4.073
16		1.746	2.120	2.583	2.921	3.686	4.015
17		1.740	2.110	2.567	2.898	3.646	3.965
18		1.734	2.101	2.552	2.878	3.610	3.922
19		1.729	2.093	2.539	2.861	3.579	3.883
20		1.725	2.086	2.528	2.845	3.552	3.850
21		1.721	2.080	2.518	2.831	3.527	3.819
22		1.717	2.074	2.508	2.819	3.505	3.792
23		1.714	2.069	2.500	2.807	3.485	3.768
24		1.711	2.064	2.492	2.797	3.467	3.745
25		1.708	2.060	2.485	2.787	3.450	3.725

The t test statistic value is 6.369. As 6.369 > -3.745, it means that the value falls inside the range. In other words, we do not have enough evidence at the 99.95% level of confidence to reject the null hypothesis. That is, the average *mean score of the **private** schools in RJ in ENEM 2019 is equal or greater than the national average.*

4.5.4 Understanding the results - Private Schools in RJ - Plots

T- test (lower-tailed test)



In the plot above it is possible to confirm the 4.5.3 affirmation.

4.6 Hypothesis Test - State Schools in RJ

State *schools in RJ claim that the mean score of their schools in ENEM 2019 is equal or greater than the national average, which is 524.54. A random sample of the average grade of 25* ***state*** *schools in RJ will be taken. Is there enough evidence to support the* ***state*** *schools' claim at a 0.05 significance level?*

Considering the situation above, it is possible to write the null hypothesis and the alternative hypothesis.

null hypothesis and alternative hypothesis - State Schools in RJ

H_0: μ >= 524.54
H_1: μ < 524.54

4.6.1 Choosing The Test - State Schools in RJ

The scenario is similar to the two previous cases. I decided to keep the level of confidence at 99.95%.

In the hypothesis created above, the standard deviation of the population is unknown. Besides that, the sample size is 25 (that is, less than 30).

Considering all that, in this situation the t-test is more appropriate.

Moreover, I am going to perform a lower-tailed test. As I would like to know if one population mean is less than the other, this one-tailed test is more appropriate. It is important to highlight that in this case, the "reject region" is in the left tail.

4.6.2 Doing The Test - State Schools in RJ

To start off with, let me organize all the information:

T- test formula	$t = \frac{\bar{x} - \mu}{s/\sqrt{n}}$

Mean of the sample	`# Finding the mean of the sample`
	`np.mean(rj_state_sample)`
	`490.2468`

Mean of the population	524.54

Standard deviation of the sample	`# Finding the standard deviation of the sample`
	`np.std(rj_state_sample)`
	`26.929324643592533`

Data sample size	25
Degrees of freedom	24
Level of confidence	99.95%
Alpha	0.05%

After having everything set, it is now possible to find the t-test statistic.

T-test

Preparing the formula - RJ State Schools

```
# mean of the sample
x1_rj_state = np.mean(rj_state_sample)
```

```
# mean of the population
X2 = 524.54
```

```
# standard deviation of the sample
s_rj_state = np.std(rj_state_sample)
```

```
# squate root
n_sqrt_rj= math.sqrt(25)
n_sqrt_rj
```

```
5.0
```

t-test (RJ State Schools) ¶

```
# t-test - RJ State
t_rj_state = (x1_rj_state-X2)/(s_rj_state/n_sqrt_rj)
t_rj_state
```

```
-6.367259568122805
```

4.6.3 Understanding the results - State Schools in RJ

First it is necessary to check the table of critical values of t. As the degree of freedom is 24 and I am performing a lower-tailed test, the critical value found was -3.745.

Critical values of t

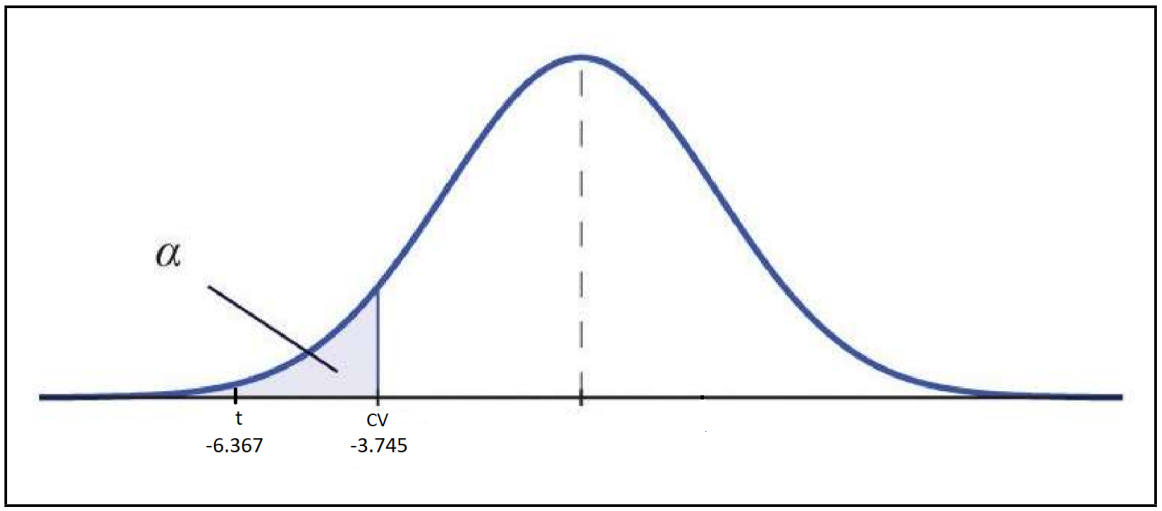
t Distribution: Critical Values of t

		Significance level					
Degrees of freedom	*Two-tailed test:* *One-tailed test:*	10% 5%	5% 2.5%	2% 1%	1% 0.5%	0.2% 0.1%	0.1% 0.05%
1		6.314	12.706	31.821	63.657	318.309	636.619
2		2.920	4.303	6.965	9.925	22.327	31.599
3		2.353	3.182	4.541	5.841	10.215	12.924
4		2.132	2.776	3.747	4.604	7.173	8.610
5		2.015	2.571	3.365	4.032	5.893	6.869
6		1.943	2.447	3.143	3.707	5.208	5.959
7		1.894	2.365	2.998	3.499	4.785	5.408
8		1.860	2.306	2.896	3.355	4.501	5.041
9		1.833	2.262	2.821	3.250	4.297	4.781
10		1.812	2.228	2.764	3.169	4.144	4.587
11		1.796	2.201	2.718	3.106	4.025	4.437
12		1.782	2.179	2.681	3.055	3.930	4.318
13		1.771	2.160	2.650	3.012	3.852	4.221
14		1.761	2.145	2.624	2.977	3.787	4.140
15		1.753	2.131	2.602	2.947	3.733	4.073
16		1.746	2.120	2.583	2.921	3.686	4.015
17		1.740	2.110	2.567	2.898	3.646	3.965
18		1.734	2.101	2.552	2.878	3.610	3.922
19		1.729	2.093	2.539	2.861	3.579	3.883
20		1.725	2.086	2.528	2.845	3.552	3.850
21		1.721	2.080	2.518	2.831	3.527	3.819
22		1.717	2.074	2.508	2.819	3.505	3.792
23		1.714	2.069	2.500	2.807	3.485	3.768
24		1.711	2.064	2.492	2.797	3.467	3.745
25		1.708	2.060	2.485	2.787	3.450	3.725

The t test statistic value is -6.367. As -6.367 < -3.745, it means that the value falls **outside** the range. In other words, we have enough evidence at the 99.95% level of confidence to reject the null hypothesis. That is, the alternative hypothesis is accepted. That means that the average *mean score of the **state** schools in RJ in ENEM 2019 is less than the national average.*

4.6.4 Understanding the results - State Schools in RJ - Plots

T- test (lower-tailed test)



In the plot above it is possible to confirm the 4.6.3 affirmation.

4.7 Conclusion - First Section

My intention in this part of the project was to have a better picture of the school's situation in RJ. With this dataset, I was able to filter the information I wanted in order to find the answers for the questions I was asking.

At first, when I did the research to better understand ENEM 2019 in Brazil, I had the opportunity to see the exam regarding the whole country. However, when I compared this information with the one found in the dataset, I was surprised.

As a person born and raised in RJ, I could realize a t-test to analyze a hypothesis that says that RJ schools have better performance than the national average in ENEM 2019. All the calculations (finding a random sample, standard deviation, mean e etc) were done using Python, and, with 95% level of confidence, it is possible to confirm that hypothesis.

Nevertheless, because of my life experience as a student and teacher in RJ, I was able to create 2 new hypotheses regarding private and public schools.

As expected, it is possible to affirm with 99.95% level of confidence that in ENEM 2019 private schools in RJ had better performance than the national average.

Surprisingly, when just state schools are considered, there is no enough evidence at a 99.95% level of confidence to make a similar affirmation.

5. SECOND SECTION

5.1 Choosing the Variables

When we analyze the dataset, it is possible to see that, in general, schools that have students with high grades in some subjects tend to also have students with high grades in the other subjects. In other words, for example, schools with good performance in “science” tend to have good performance in “composition”, and vice versa, even if these 2 subjects are not directly related.

It may happen because good schools tend to have high performance in all subjects, while bad schools tend to perform badly in all the topics.

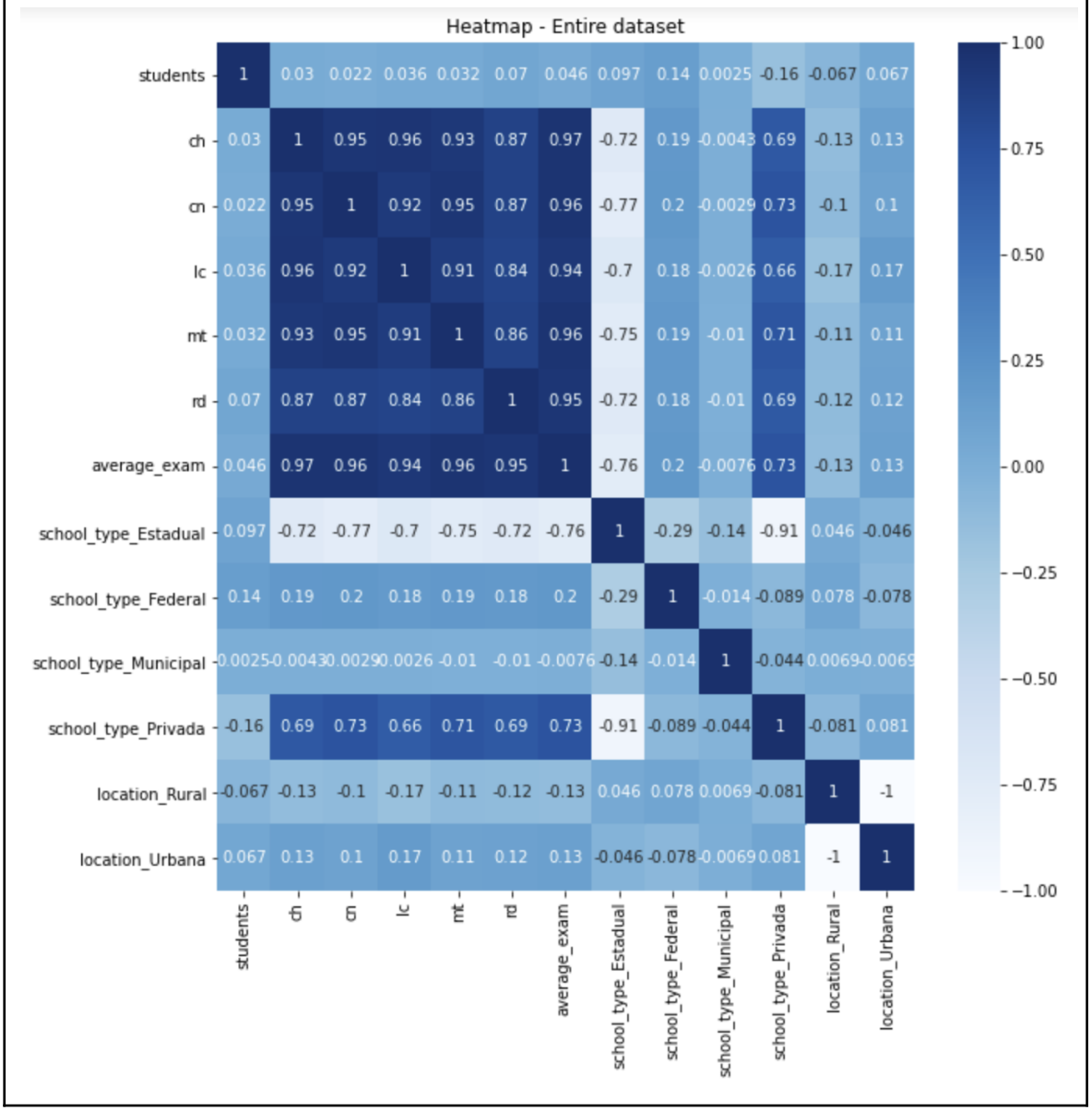
Keeping this idea in mind, I decided to understand the correlation among the different variables of the dataset. I also decided to highlight the variable “mt” (mathematics and its technologies), and see if there is any high correlation.

5.2 Correlation Analysis

According to Hayes (2022), correlation can be described as a statistical term that explains the degree to which 2 variables move in coordination with each other. Mainly, it is possible to say that if the 2 variables move in the same direction, they have a positive correlation. Oppositely, the variables would have a negative correlation if they move in opposite directions. Regarding numbers, the correlation coefficient must fall between -1.0 and +1.0.

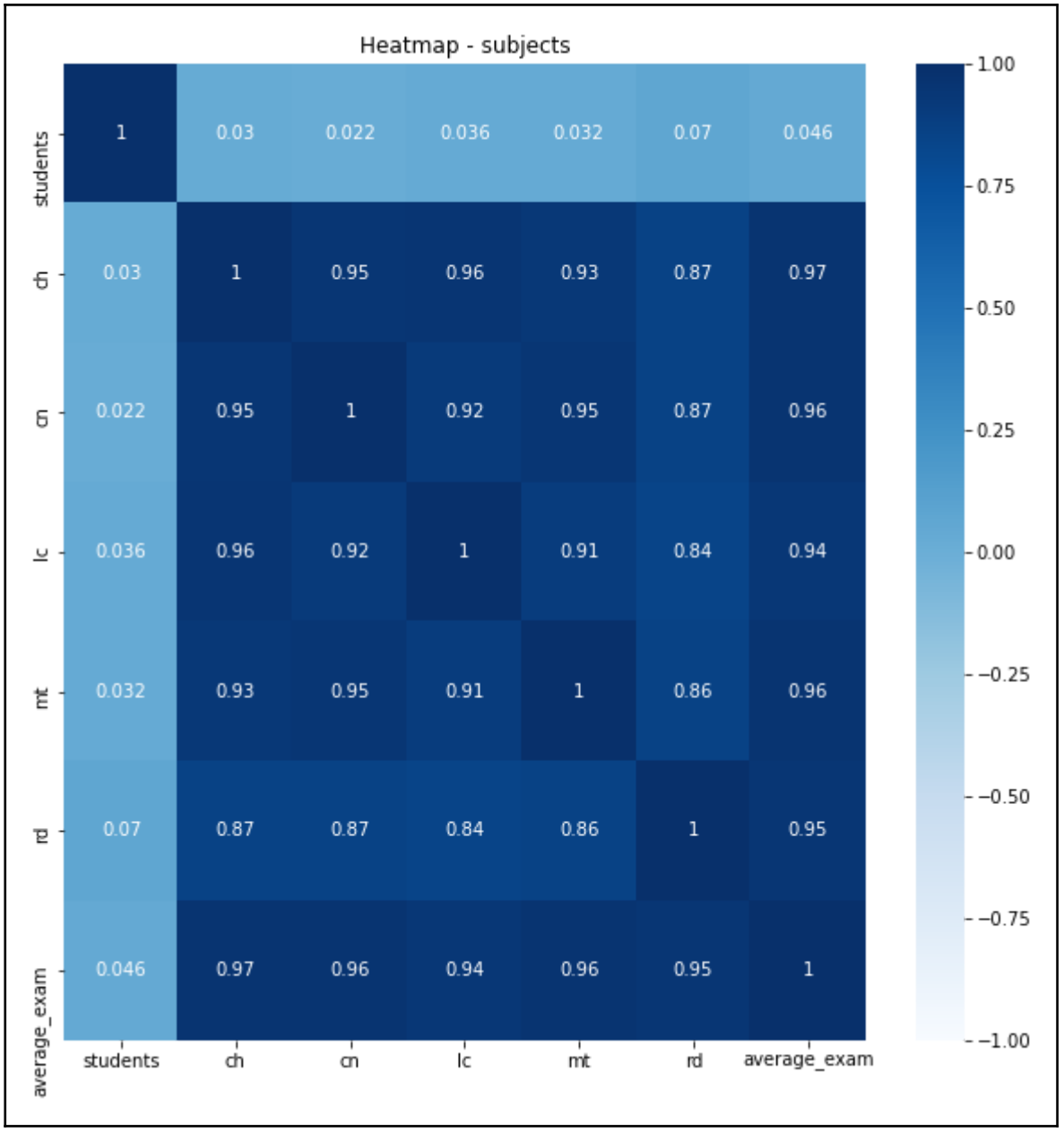
For this project, the function **corr()** is used to find correlation coefficient. For better visualization, this information will be displayed using a heatmap. I am using the method='pearson'.

Heatmap - entire dataset



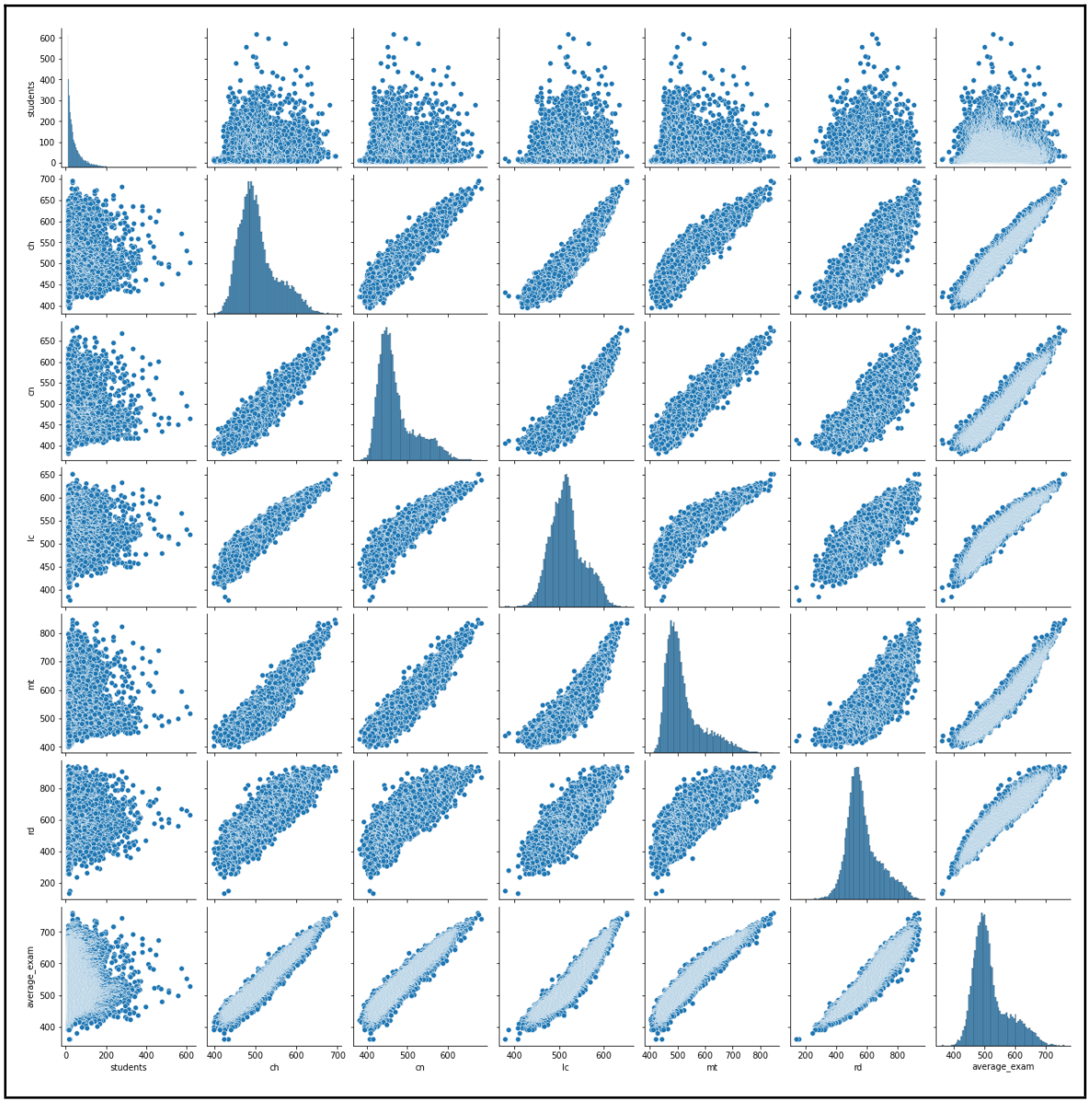
As it is possible to see above, the ENEM subjects have high correlation among themselves. For better visualization, I am going to show the correlation of just the school's subjects and the number of students.

Heatmap - subjects and number of students



I also decided to use a scatter plot to show the correlation among these variables.

Pairplot - subjects and number of students



This information can also be confirmed by using different methods.

Correlation - different methods

```
1 df_preparing.corr(method='spearman')
```

	students	ch	cn	lc	mt	rd	average_exam
students	1.000000	-0.001790	-0.000242	0.003342	0.012298	0.074318	0.035402
ch	-0.001790	1.000000	0.917496	0.949579	0.899279	0.822149	0.949637
cn	-0.000242	0.917496	1.000000	0.906532	0.902356	0.816644	0.938915
lc	0.003342	0.949579	0.906532	1.000000	0.894698	0.806192	0.939114
mt	0.012298	0.899279	0.902356	0.894698	1.000000	0.819859	0.941153
rd	0.074318	0.822149	0.816644	0.806192	0.819859	1.000000	0.935592
average_exam	0.035402	0.949637	0.938915	0.939114	0.941153	0.935592	1.000000

```
1 df_preparing.corr(method='kendall')
```

	students	ch	cn	lc	mt	rd	average_exam
students	1.000000	-0.000663	0.000816	0.002830	0.009128	0.052932	0.025581
ch	-0.000663	1.000000	0.761830	0.813518	0.735267	0.635051	0.813433
cn	0.000816	0.761830	1.000000	0.744495	0.740272	0.629745	0.796029
lc	0.002830	0.813518	0.744495	1.000000	0.727143	0.616960	0.793300
mt	0.009128	0.735267	0.740272	0.727143	1.000000	0.631815	0.798688
rd	0.052932	0.635051	0.629745	0.616960	0.631815	1.000000	0.785595
average_exam	0.025581	0.813433	0.796029	0.793300	0.798688	0.785595	1.000000

```
1 df_preparing.corr(method='pearson')
```

	students	ch	cn	lc	mt	rd	average_exam
students	1.000000	0.029988	0.021889	0.035938	0.031819	0.070456	0.046163
ch	0.029988	1.000000	0.946825	0.955913	0.934893	0.866870	0.966193
cn	0.021889	0.946825	1.000000	0.918505	0.949427	0.866444	0.964737
lc	0.035938	0.955913	0.918505	1.000000	0.905033	0.843832	0.944791
mt	0.031819	0.934893	0.949427	0.905033	1.000000	0.862999	0.963709
rd	0.070456	0.866870	0.866444	0.843832	0.862999	1.000000	0.950760
average_exam	0.046163	0.966193	0.964737	0.944791	0.963709	0.950760	1.000000

Finally, I decided to choose two different variables to work on in this project. They are:

Feature	**Description**
mt	mathematics and its technologies
lc	languages, codes and their technologies

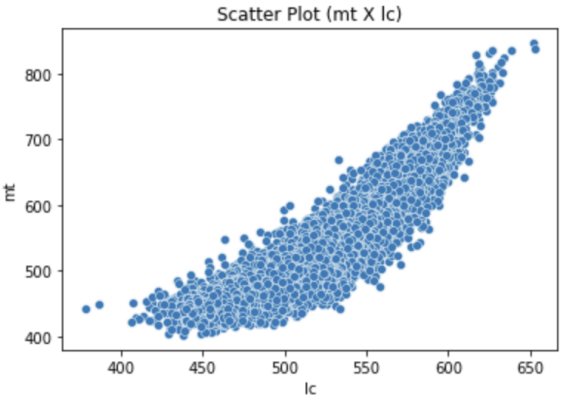
I have chosen these two variables because they are considered the two most important subjects in Brazil. LC englobes the language, it means, the students are tested in subjects like Portuguese, English, Spanish, grammar and literature. On the other hand, MT is pure math. It means, students are tested in statistics, probability, trigonometry, and etc.

Regarding the strength of the correlation between the 2 variables, as it was possible to see after using the **corr()** function above, the variables have a strong positive correlation. It can be affirmed because when we analyze the Pearson, Kendall and Spearman correlation coefficient, the values found are between 0.7 and 0.9.

Correlation Between MT and LC		
Pearson	Kendall	Spearman
0.905033	0.727143	0.894698

First, I am going to use a scatter plot to demonstrate the correlation between these two variables.

Correlation - scatter plot: mt X lc



It is also important to verify the line that best explains the correlation between these two variables. For this example, I am assuming that the grade in MT can be found by analyzing the grades in LC. First, let me find a and b.

Finding the values

```
# best fit polynomials
# polynomial
mt_grade = np.polyfit(df_preparing.lc, df_preparing.mt, 1)

mt_grade
```

```
array([   1.68592363, -350.78188921])
```

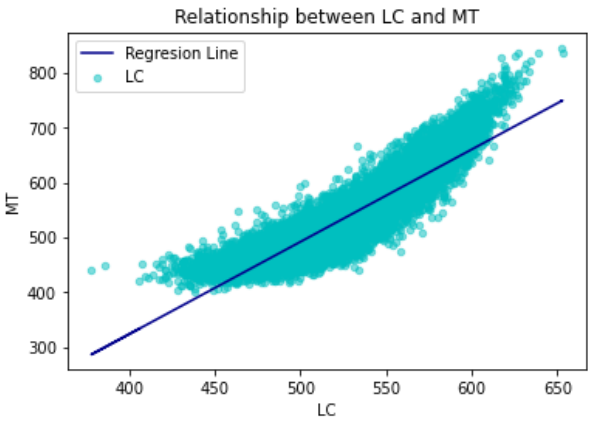
As it is possible to see after the calculations, a= 1.69 and b = -350.78.
It is now possible to write the Linear Regression Equation:

MT = M
LC = L

M = -350.78 + 1.69*L

Finally, it is possible now to use the information above to plot a scatter plot and the regression line model together:

Relationship between LC and MT (Regression line)



y=-350.78+1.69*x

5.3 Correlation vs Causation

First it is important to understand the difference between correlation and causation. According to Bhandari (2021), correlation is a statistical association between variables, that is, when one changes, the other does too. On the other hand, causation means that there is a cause-and-effect relationship between variables, or in other words, that changes in one variable brings changes in the other variable.

According to Ramzai (2020), Pearson correlation coefficient is recommended when there is a linear relationship between two variables. As it is possible to confirm in the scatter plots and tables above, LC and MT have a linear relationship. Besides that, it was also shown above that LC and MT have a strong positive correlation.

Keeping this in mind it is important to think if in this case correlation implies causation. That is the sentence that is important to analyze:

> The reason why a certain school has a certain grade in MT is this school grade in LC.

For obvious reasons, this sentence does not make sense. Of course, it is possible to affirm that the level of education in the schools are similar among the different subjects, and consequently, students may have similar performance in the national exam.

However, it is impossible to affirm that one grade is caused by the other. It means that, in this case, correlation does not imply causation.

5.4 Conclusion - Second Section

My intention was to carry out a correlation analysis between 2 variables. In order to choose these two variables, first I performed a correlation analysis using all variables and 3 different methods (Spearman, Kendall and Pearson). Besides that, heatmaps and scatter plots were used in order to have a better visualization of the correlations.

Finally, the two variables were chosen, and after analyzing the scatter plot, I found and plotted the regression line and the equation.

After all the information discovered, I could analyze the correlation vs causation of the two variables. It was interesting to think about this scenario, and in the end, conclude that in this case, correlation does not imply causation.

6. THIRD SECTION

6.1 Using the same 2 variables

After analyzing the correlation between LC and MT in the previous section, I am going to use these variables to build a linear regression model. As demonstrated previously, M = 350.78 + 1.69*L is the equation of the model. It is important to notice that these numbers can also be found using Scikit Learn, as demonstrated below:

Intercept_ and coef_ values

```
from sklearn.linear_model import LinearRegression

# create linear regression object
regressor = LinearRegression()

# fitting the model
regressor.fit(df_preparing[['lc']], df_preparing['mt'])

# get the slope and intercept of the line best fit
print('intercept_ = ', regressor.intercept_)

print('coef_      = ', regressor.coef_)

```

```
intercept_ =  -350.78188920624007
coef_      =  [1.68592363]
```

6.2 Machine Learning

To start off with, it is important to understand the concept of Machine Learning. According to Brown (2021), Machine Learning can be explained as the capability of a machine to imitate human behavior in order to solve problems by performing complex tasks and calculations. In other words, a Machine Learning model may be able to make predictions after using a variety of algorithms and statistical models. It means that computer systems may be able to learn, grow and/or adapt new data in order to imitate how humans acquire new knowledge, gradually improving its accuracy. For this project, linear regression will be used. The main idea is to better understand the correlation between “mt” and “lc”, and build a model that allows the prediction of the “mt” average grades.

In order to test the Linear Regression model in the future, I decided to split the dataset into train and test. For this project, the test size will be 20% of the dataset.

Train test split

```
# Spliting the dataset
from sklearn.model_selection import train_test_split
X = df_preparing['lc'].values
y = df_preparing['mt'].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=0)
```

6.3 Linear Regression (using 2 variables)

Linear regression models represent the proportional relationship between the dependent and the independent variable. Maily, it shows that both variables tend to increase or decrease together. Graphically speaking, linear regression models have a line trying to demonstrate the pattern among the points that represent the variables. It is important to notice that those points are regularly not on the line and that they do not need to be. The main idea is that this straight line gives an adequate representation of the variable's distribution.

In order to perform linear regression in this dataset, some reshaping is necessary. After doing this, as demonstrated below, linear regression was used on the training set.

Linear regression

6.2.2 Reshaping

```
X_train= X_train.reshape(-1, 1)
X_test = X_test.reshape(-1, 1)
```

6.2.4 Training the model

```
# Training the model on the Training set
regressor_LR = LinearRegression()
regressor_LR.fit(X_train, y_train)
```

```
LinearRegression()
```

6.4 Predicting

A natural step after performing the machine learning model is to check its performance. First, I am going to compare y_pred and y_test values to analyze how close the values are.

Predicting the test set result

```
# Predicting the Test set results
y_pred = regressor_LR.predict(X_test)
np.set_printoptions(precision=2) # only 2 decimals after the comma
print(np.concatenate((y_pred.reshape(len(y_pred),1),
                      y_test.reshape(len(y_test),1)),1))

```

```
[[477.87 473.97]
 [564.04 525.97]
 [527.41 515.77]
 ...
 [612.02 622.2 ]
 [650.58 673.74]
 [536.29 524.78]]
```

As it is possible to see above, the values found are similar. Below, I am going to compare the function polyval() used in the second section and the linear regression. Both will be used to predict the “mt” of a school with “lc” equal to 630.

It is important to keep in mind that the prediction made for the linear regression is made based on the X_train and y_train, while the prediction made for the polyval() is based on the entire dataset.

Predicting (numpy X scikit learn)

```
# predictions using numpy
print('Predictions using numpy (Second section): ' ,np.polyval(mt_grade, [630]))

# predictions using scikit learn
print('Predictions using Scikit Learn:           ', regressor_LR.predict([[630]]))

```

```
Predictions using numpy (Second section):  [711.35]
Predictions using Scikit Learn:            [711.71]
```

As expected, the values found were pretty similar.

6.5 Precision

As it can be seen below, the train and test set have good precision when using score() and r2_score(). According to Fernando (2021), R-squared (R^2) is a statistical measure that is used to show the proportion of the variance for a dependent variable that is explained by an independent variable in a regression model.

Precision

```
# Precision of the model - Train set
print('The precision of the model is ')
print(regressor_LR.score(X_train, y_train))
```

```
The precision of the model is
0.8204417957755553
```

```
# Precision of the model - Test set
R_square = r2_score(y_test,y_pred)
print('Coefficient of Determination', R_square)
```

```
Coefficient of Determination 0.8134992346756789
```

6.6 The equation

In order to compare the equations, I am showing below the coef_ and intercept_ of the regressor used in the linear regression. I am also showing the mean squared error.

equation

```
# the equation
print('coef =        ', regressor_LR.coef_)

print('intercept_ =  ', regressor_LR.intercept_)

# The mean squared error
print("Mean squared error: %.2f" % mean_squared_error(y_test, y_pred))
```

```
coef =         [1.69]
intercept_ =   -352.2671336594657
Mean squared error: 868.64
```

It is possible to notice the similarity of the two equations.

Linear Regression

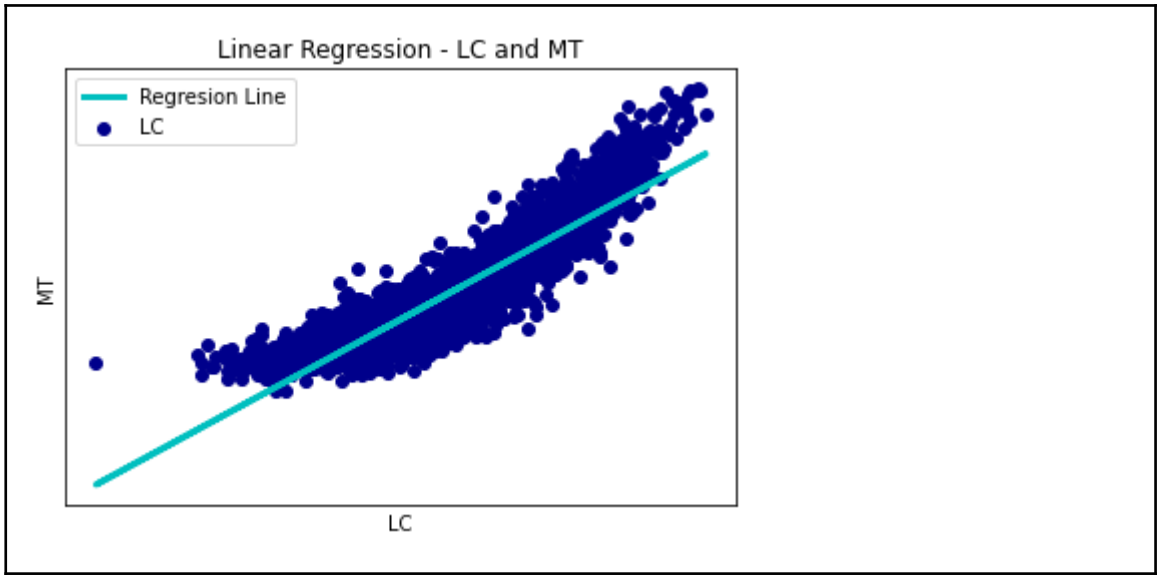
Linear Regression (lc and mt)	MT = M LC = L M = -350.78 + 1.69*L
Linear Regression (X_train and y_train)	MT = M LC = L M = -352.26 + 1.69*L

```
# Linear Regression (lc and mt)
print('Linear Regression (lc and mt)')
print('coef =         ', regressor.coef_)
print('intercept_ =   ', regressor.intercept_)

###################
print('')
# Linear Regression (X_train and y_train)
print('Linear Regression (X_train and y_train)')
print('coef =         ', regressor_LR.coef_)
print('intercept_ =   ', regressor_LR.intercept_)


```

```
Linear Regression (lc and mt)
coef =         [1.69]
intercept_ =   -350.78188920624007

Linear Regression (X_train and y_train)
coef =         [1.69]
intercept_ =   -352.2671336594657
```

Finally, I decided to plot the information found after using linear regression in the train and test set.

Linear Regression - Plot



6.7 Conclusion - Third Section

When I started this section, firstly I decided to confirm that I could find the same linear regression equation using the linear regression function. And, as expected, the values were equal to the ones found previously.

In order to be able to test the machine learning model in the end, I decided to split the dataset before performing the LinearRegression().

Regarding the predictions, it was possible to see that the values found by the regressor were pretty similar to the ones in the test set. This precision was also confirmed when I compared the predictions made with Numpy and with Scikit Learn.

Regarding scores, the coefficient of determination was found (81%). Besides that, the function score() was also used (82%). Additionally, the mean squared error was also found.

After that, I decided to compare the equations (lc and mt X train set). As expected, they were similar and with the same coef_.

Finally, I plotted the linear regression used in this third section, and could confirm the linear correlation. As the scatter plot was also in the same plot, it was possible to visualize the error (the difference between the best-fit line and the observed value).

It is important to state that what was being analyzed was how the variables are related and not any type of cause-and-effect relationship.

7. CONCLUSION

During the research period that was needed to conclude this CA, I had the opportunity to better understand some concepts of statistics. Besides that, after having analyzed a bit the ENEM dataset, I could have a better picture of schools in Brazil.

In this project, I have cleaned and prepared a dataset in order to use different statistical techniques. Regarding the hypothesis and the linear regression models, the results were satisfactory and may serve for future analysis.

Thinking about future steps, I would like to go deeper in the analysis of each state. For example, divide Brazil into its regions and compare their performance. I also would like to discover what the best schools have in common among themselves and what the main difference to the worst schools is.

Finally, all the acquired knowledge after studying Python to conclude this CA may accompany me during my career as a data analyst.

8. REFERENCE LIST

Bhandari, P. (2021). *Correlation vs causation*. [online] Scribbr. Available at: https://www.scribbr.com/methodology/correlation-vs-causation/ [Accessed 27 May 2022].

Bhutani, K. (2018). *Python | Pandas dataframe.drop_duplicates()*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/python-pandas-dataframe-drop_duplicates/ [Accessed 18 May 2022].

Chandradas, A. (2021). *5 Methods to Check for NaN Values in Python*. [online] Medium. Available at: https://towardsdatascience.com/5-methods-to-check-for-nan-values-in-in-python-3f21ddd17eed#:~:text=NaN%20stands%20for%20Not%20A [Accessed 21 May 2022].

Chappelow, J. (2019). *How Statistics Work*. [online] Investopedia. Available at: https://www.investopedia.com/terms/s/statistics.asp [Accessed 27 May 2022].

Fernando, J. (2021). *R-Squared Definition*. [online] Investopedia. Available at: https://www.investopedia.com/terms/r/r-squared.asp [Accessed 27 May 2022].

G1 (2020a). *Enem 2019: 60% das redações com nota mil foram escritas por mulheres; região Sudeste e Nordeste têm 77,3% das notas máximas*. [online] G1. Available at: https://g1.globo.com/educacao/noticia/2020/01/17/enem-2020-60percent-das-redacoes-com-nota-mil-foram-escritas-por-mulheres-regiao-sudeste-e-nordeste-tem-83percent-das-notas-maximas.ghtml [Accessed 23 May 2022].

G1 (2020b). *Notas médias do Enem 2019 caem em todas as provas objetivas*. [online] G1. Available at: https://g1.globo.com/educacao/noticia/2020/01/17/notas-medias-do-enem-2019-caem-em-todas-as-provas-objetivas.ghtml [Accessed 24 May 2022].

Galarnyk, M. (2018). *Understanding Boxplots*. [online] Medium. Available at: https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51 [Accessed 24 May 2022].

Hayes, A. (2021). *How t-tests work*. [online] Investopedia. Available at: https://www.investopedia.com/terms/t/t-test.asp [Accessed 27 May 2022].

Hayes, A. (2022). *What Is Correlation in Finance?* [online] Investopedia. Available at: https://www.investopedia.com/terms/c/correlation.asp#:~:text=Correlation%20is%20a%20statistical%20term [Accessed 23 May 2022].

Patil, P. (2018). *What is Exploratory Data Analysis?* [online] Towards Data Science. Available at: https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15 [Accessed 23 May 2022].

Pearlman, S. (2018). *What is Data Preparation? (+ How to Make It Easier) - Talend*. [online] Talend Real-Time Open Source Data Integration Software. Available at: https://www.talend.com/resources/what-is-data-preparation/ [Accessed 21 May 2022].

Ph. D., M.S. and E., B. A., C. and B. A., C.S. (2019). *What Is a Population Parameter?* [online] ThoughtCo. Available at: https://www.thoughtco.com/population-parameter-4588247#:~:text=In%20statistics%2C%20a%20population%20parameter [Accessed 23 May 2022].

Ramzai, J. (2020). Clearly explained: Pearson V/S Spearman Correlation Coefficient. *Medium*. [online] 25 Jun. Available at: https://towardsdatascience.com/clearly-explained-pearson-v-s-spearman-correlation-coefficient-ada2f473b8 [Accessed 23 May 2022].

Team, G.L. (2020). *What is Label Encoding in Python | Great Learning*. [online] GreatLearning Blog: Free Resources what Matters to shape your Career! Available at: https://www.mygreatlearning.com/blog/label-encoding-in-python/ [Accessed 27 May 2022].

Yi, M. (2021). *A Complete Guide to Bar Charts*. [online] Chartio. Available at: https://chartio.com/learn/charts/bar-chart-complete-guide/ [Accessed 26 May 2022].