University of Science and Technology of Hanoi

Data Analysis

Final Report

Exploring Data Analysis Code: Insights from USTH Course

Authors:
Nguyen Thi Yen Binh

Instructor:
Dr. Nguyen Le Dung

March 31, 2024

Contents

List of Figures

1 Introduction

This report serves as the final exam for the Data Analysis Course instructed by Dr. Nguyen Le Dung. All of the code is written in Python, and as you may guess, many packages are used in this code to make it very clear and compact. However, it may be hard for people to follow my work and it is important that we learn the mathematical model and the insights of the work we do. Therefore, this report provides explanations to make up for the complexity of the code and also explain about the purpose of the data and how can we use them to make insightful studies and powerful prediction about the future.

The contents are divided into seven main parts, covering the basics of data handling in Python, some insights on fundamental probability and statistics, regression and trend analysis, hypothesis testing, and also some advanced techniques like dimensionality reduction and clustering.

Additionally, this document is dedicated to explaining the concepts as well as giving interpretations to the results obtained from the program. All of my Python code is attached to this report in the 7 Jupyter Notebooks.

Python is friendly language providing huge collection of package which make it easy for us to working data. In this course, we mainly use `NumPy`, a core library for scientific computing, especially, for working with array, `Pandas`, a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, `Matplotlib`, a comprehensive library for creating static, animated, and interactive visualizations in Python. Also, some addition packages are `scienceplots` and `seaborn` for making the plot look more beautiful.

2 Basics of Python

Data handling one of the most basic and important step in Data Analysis, you should know how to handle the file before using them.

- The data we use are commonly stored in `.txt` or `.csv` file.
- There are several way to read those file. I will present 3 ways including the file handling of python, numpy package and pandas packages.
- I will compare the running time and to find out which is the best way.

2.1 File reading using built-in library of Python, NumPy and Pandas

2.1.1 Read file using Normal way

```
%%timeit
rain_DB_2006 = []
with open('data/Dien_bien_Aug_2006_luongmua_mm.txt') as f:
    for line in f:
        x1, y1 = line.strip().split(",")
        rain_DB_2006.append([int(x1), float(y1)])
rain_DB_2006 = np.array(rain_DB_2006).T
```

```
39.6 µs ± 654 ns per loop (mean ± std. dev. of 7 runs, 10000 loops each)
```

2.1.2 Read file using Numpy

```
%%timeit
rain_DB_2006 = np.loadtxt('data/Dien_bien_Aug_2006_luongmua_mm.txt',
   delimiter=',',unpack=True)
```

```
185 µs ± 1.19 µs per loop (mean ± std. dev. of 7 runs, 10000 loops each)
```

2.1.3 Read file using Pandas

```
%%timeit
rain_DB_2006 = pd.read_csv('data/Dien_bien_Aug_2006_luongmua_mm.txt', header
    = None)
```

```
698 µs ± 2.43 µs per loop (mean ± std. dev. of 7 runs, 1000 loops each)
```

2.1.4 Compare the time taken by three different ways of file reading

From the result, we see that the time using to read an array of small size (array(2,31) in our example) is fastest in normal method and a little bit slower in NumPy and Pandas. However,

the syntax of the normal way is too complicated and time-consuming while NumPy and Pandas provide a more friendly syntex. Also, with a larger data set, NumPy and Pandas will be faster than the built-in file reading functions in Python.

Moreover, Pandas are designed to give the best performance for data analysis and visulization. Therefore, you will see I mostly use Pandas for data retrieval through out this report.

2.2 Data modification and File writing

```
rain_DB_2006 = pd.read_csv('data/Dien_bien_Aug_2006_luongmua_mm.txt', header
    = None)
rain_DB_2006[1] = (rain_DB_2006[1]*1.5)
with open('output/Dien_bien_Jul_2006_luongmua_mm.txt', 'w') as f:
    rain_DB_2006 = rain_DB_2006.to_string(header=False, index=False)
    f.write(rain_DB_2006)
```

To read the data from *.csv* file, we use *read_csv* function in `Pandas` which will convert the data into a DataFrame, a two-dimensional, tabular, mutable data structure which is useful for data handling.

With this array, we can do basic mathematical operation and algebra for the whole columns pretty simple (similar to what I do on the second line) just like we operate with matrices.

Finally, we convert the data to string and write the and '.txt' file. However, we can do better by storing the numeric data in '.csv' file using 'to_csv' of Pandas.

2.3 Datetime parsing, file concatenation and mean value

```
rain_DB_jul = pd.read_csv('output/Dien_bien_Jul_2006_luongmua_mm.txt',
    header=None, delim_whitespace=True, names = ['Date', 'Rainfall'])
rain_DB_aug = pd.read_csv('data/Dien_bien_Aug_2006_luongmua_mm.txt', header=
    None, names = ['Date', 'Rainfall'])

rain_DB_jul['Date'] = pd.to_datetime(rain_DB_jul['Date'], unit='D', origin=
    pd.Timestamp('2006-06-30'))
rain_DB_aug['Date'] = pd.to_datetime(rain_DB_aug['Date'], unit='D', origin=
    pd.Timestamp('2006-07-31'))
```

Datetime is very important in a data, it is suggested that we should convert the datetime to the 'Datetime' type, so that it is more convenient for analysis and visulization.

```
rain_jul_aug = pd.concat([rain_DB_jul, rain_DB_aug])
```

Concatenation is a way to combine two different DataFrame forming a single data of rainfall amount from July to August. From this data, we can make a Time Series plot. (The code is provided in Jupyter Notebook.)

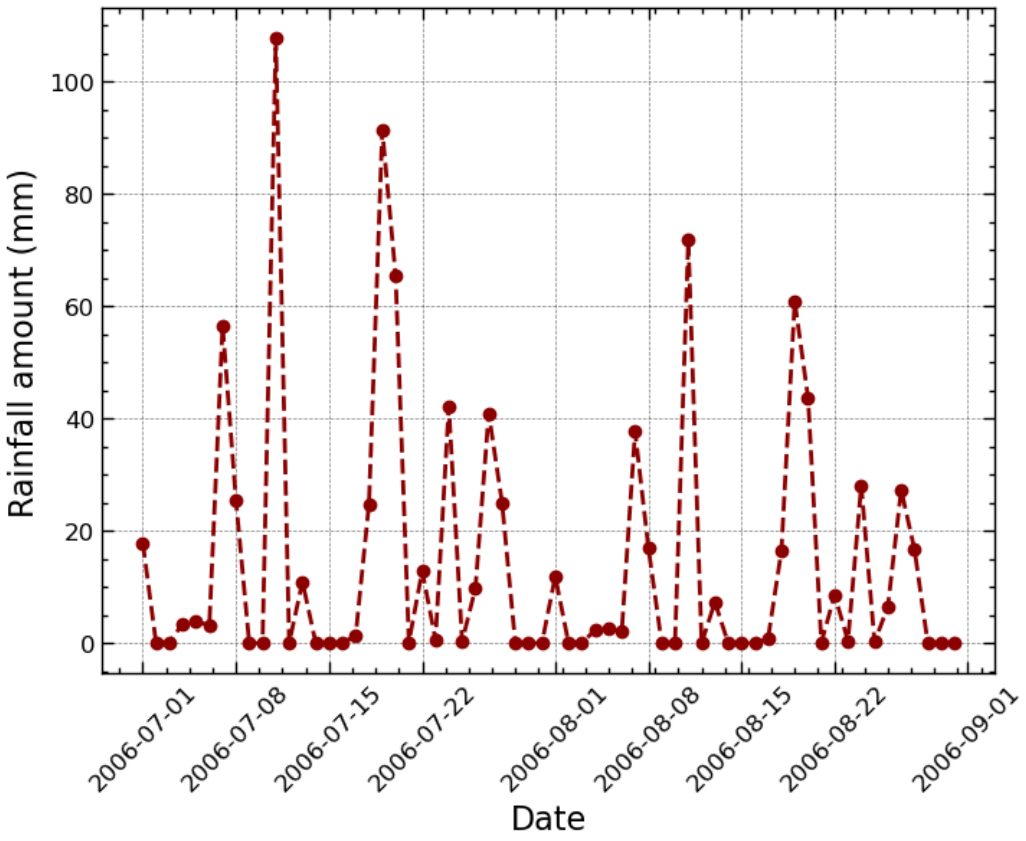

Figure 1: Time Series of Rainfall amount

To calculate the mean value of the rainfall amount in July and August, `NumPy` provide us mean function which operates very simple on an array, as I should on the code.

```
mean_jul = np.mean(rain_DB_jul['Rainfall'])
mean_aug = np.mean(rain_DB_aug['Rainfall'])
print("The monthly mean of rainfall in July and August of Dien Bien in 2006
    are %f mm and %f mm"%(mean_jul, mean_aug))
```

2.4 File reading of CO2 data

After the basic, we will try to read an example file of CO_2, `Pandas` provides us various arguments to treat the files, like seperate the data into the different columns (`sep`, `delimwhitespace`,...), mask the unwanted data (`skiprows`, `usecols`, `navalues`), parse datetime, etc.

```
df_co = pd.read_csv('data/co2_mm_mlo.txt',
                    delim_whitespace=True,
                    skiprows=72,
                    names = ['year','month','datetime_decimal','average','
    interpolated CO','trend', 'days'],
                    usecols = ['year', 'month', 'interpolated CO'], #year
    and month are used for timestamps, interpolated data are only considered
    in this homework.
                    parse_dates={ 'date': ['year', 'month'] },
                    na_values=[-99.99, -1],
                    index_col = ['date'],)
```

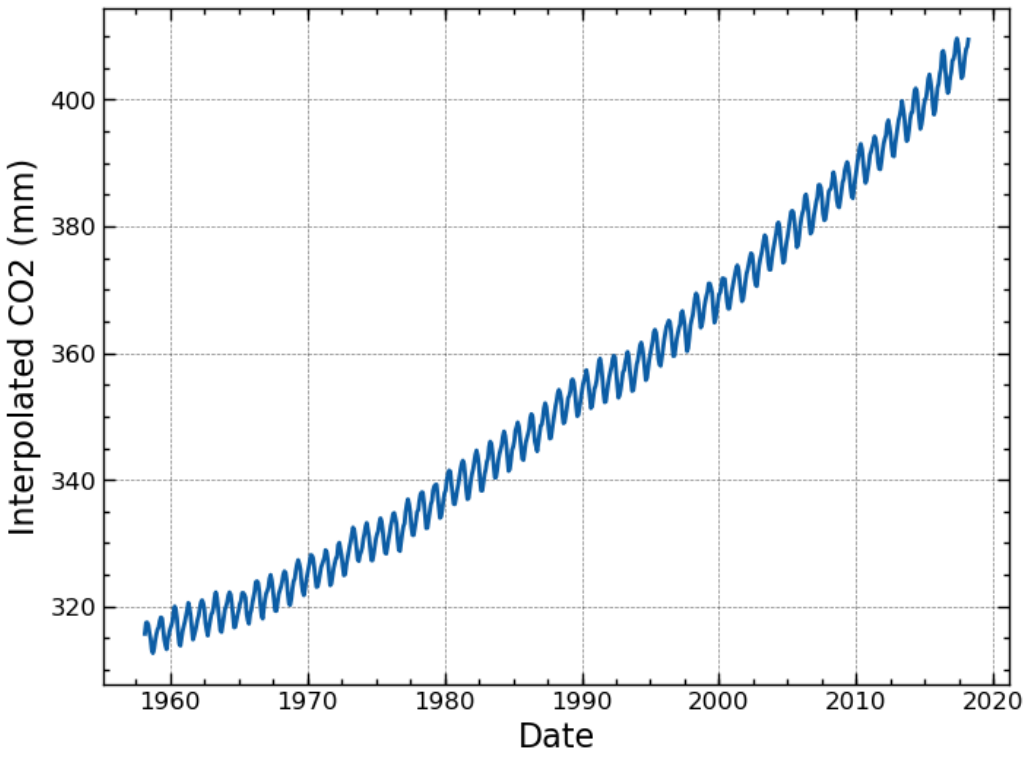

Figure 2: Interpolated CO_2 in mm

2.5 File reading of GTA data

```
df_gta = pd.read_csv('data/gta_1958_2018.csv',
                     skiprows=5,
                     header=None,
                     names=['Year', 'GTA'])
df_gta['date'] = pd.to_datetime(df_gta['Year'], format="%Y%m", errors='
    ignore')
df_gta['Year'] = df_gta['date'].dt.year
```

Means value of GTA from 1958 to 1977 and from 1999 to 2018 can be calculated by using a mask. A mask is a boolean argument on the desired range and DataFrame can return the value defined by our mask.

```
def means_func(start_date, end_date):
    mask = (df_gta['date'] >= start_date) & (df_gta['date'] <= end_date) #
    define the interval in the timeseries to take the mean value
    return df_gta[mask]['GTA'].mean()
```

```
means_1958_1977 = means_func('1958-01-01', '1977-12-01')
means_1999_2018 = means_func('1999-01-01', '2018-12-01')

print("Mean value of GTA from 1958 to 1977 is %.2f mm and from 1999 to 2018
    is %.2f mm"%(means_1958_1977, means_1999_2018))
```

```
Mean value of GTA from 1958 to 1977 is 0.02 mm and from 1999 to 2018 is 0.65 mm
```

Combine two data of GTA and CO

```
df_gta.set_index('date',inplace=True)
df_gtaco = df_co.join(df_gta['GTA'])
```

2.6 Some conclusions and comment

The work on the section 2 is fairly easy, so I did go a little into the technical detail with the hope that you can understand how to make use of some universal packages like `NumPy` and `Pandas`. However, from the next section, I will focus more on the theoretical background and the result interpretation rather than the technique.

2.6.1 What is GTA?

Working with data required you to understand the meaning of data, the rainfall amount and the CO_2 amount is comprehensive, with GTA, I do a little research.

GTA stands for Global Temperature Anomaly, which is the amount of temperature difference from the reference value or the long-term average. A positive anomaly indicates that the temperature is higher than normal and a negative anomaly indicates that temperature is cooler than the normal value.

3 Fundamental Statistics

3.1 Theory on Statistics

Statistics is the branch of mathematics that deals with the collection, analysis, interpretation, presentation, and organization of data. It is an essential tool for making decisions based on data-driven insights.

3.1.1 Median

The median of a dataset is the value separating the higher half from the lower half when the data is arranged in order. It is a measure of central tendency that is more robust to outliers than the mean. The formula for finding the median depends on whether the dataset has an odd or even number of values.

If the dataset has an odd number of values, the median is the middle value. That is, if we have a dataset $x_1, x_2, \ldots, x_n$ where n is odd, the median is:

$$\text{median} = x_{\frac{n+1}{2}} \tag{1}$$

If the dataset has an even number of values, the median is the average of the two middle values. That is, if we have a dataset $x_1, x_2, \ldots, x_n$ where n is even, the median is:

$$\text{median} = \frac{x_{\frac{n}{2}} + x_{\frac{n}{2}+1}}{2} \tag{2}$$

3.1.2 Mean

The mean of a dataset is the arithmetic average of its values. It is a measure of central tendency that is sensitive to outliers. The formula for the mean of a dataset $x_1, x_2, \ldots, x_n$ is:

$$\text{mean} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{x_1 + x_2 + \cdots + x_n}{n} = E[X] = \mu \tag{3}$$

where n is the number of values in the dataset.

3.1.3 Variance and Standard Deviation

The variance of a dataset measures how much the values in the dataset vary from the mean. The formula for the variance of a dataset $x_1, x_2, \ldots, x_n$ is:

$$\text{variance} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2 \tag{4}$$

where mean is the mean of the dataset.

The standard deviation of a dataset is the square root of its variance. It measures the amount of dispersion or spread of the values in the dataset. The formula for the standard deviation of a dataset $x_1, x_2, \ldots, x_n$ is:

$$\text{standard deviation} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2} \tag{5}$$

3.1.4 Percentile

A percentile is a measure used in statistics to indicate the value below which a given percentage of observations in a group of observations falls. The pth percentile is the value below which p percent of the observations fall. The formula for finding the pth percentile of a dataset $x_1, x_2, \ldots, x_n$ is:

$$p\text{th percentile} = x_{\lceil p/100 \cdot n \rceil} \tag{6}$$

3.2 Code for statistics

To perform statistical calculations in Python, we can use the `NumPy` library for mathematical operations and the `stats` module in the `scipy` library for statistical functions. Alternatively, we can use the describe function in *Pandas* to quickly obtain a summary of our data. The describe function provides statistics such as count, mean, standard deviation, minimum value, and maximum value for each column in a `DataFrame`.

```
gta_info['Mean'] = np.mean(df_gta['GTA'])
gta_info['Min'] = np.min(df_gta['GTA'])
gta_info['Max'] = np.max(df_gta['GTA'])
gta_info['Stadard Deviation'] = np.std(df_gta['GTA'])
gta_info['Variance'] = np.var(df_gta['GTA'])
gta_info['25%'] = np.percentile(df_gta['GTA'], 25)
gta_info['75%'] = np.percentile(df_gta['GTA'], 75)
gta_info['Median'] = np.median(df_gta['GTA'])
gta_info['Mode'] = stats.mode(df_gta['GTA']).mode[0]
```

3.3 Plots

Plots are a great way to visualize data and gain insights into patterns and relationships. In Python, we can use various libraries such as matplotlib and seaborn to create different types of plots. Here are some common types of plots that are often used in data analysis:

3.3.1 Time series plot

A time series plot is used to visualize data that changes over time. This type of plot is useful for exploring trends, seasonality, and patterns in data over a period of time. In matplotlib, we can create a time series plot using the plot function.

3.3.2 Histogram plot

A histogram plot is used to visualize the distribution of a dataset. It is a bar graph-like representation of data that is divided into intervals, or bins. The height of each bar represents the number of data points that fall within that bin. In matplotlib, we can create a histogram plot using the hist function.

3.3.3 Box plot

A box plot is used to visualize the distribution of a dataset, particularly to show the median, quartiles, and outliers. In matplotlib, we can create a box plot using the boxplot function.

3.3.4 Error bar plot

An error bar plot is used to visualize the variability of data. It typically includes a central mark indicating the mean or median, and bars indicating the amount of variability in the data. In matplotlib, we can create an error bar plot using the errorbar function.

3.3.5 Result

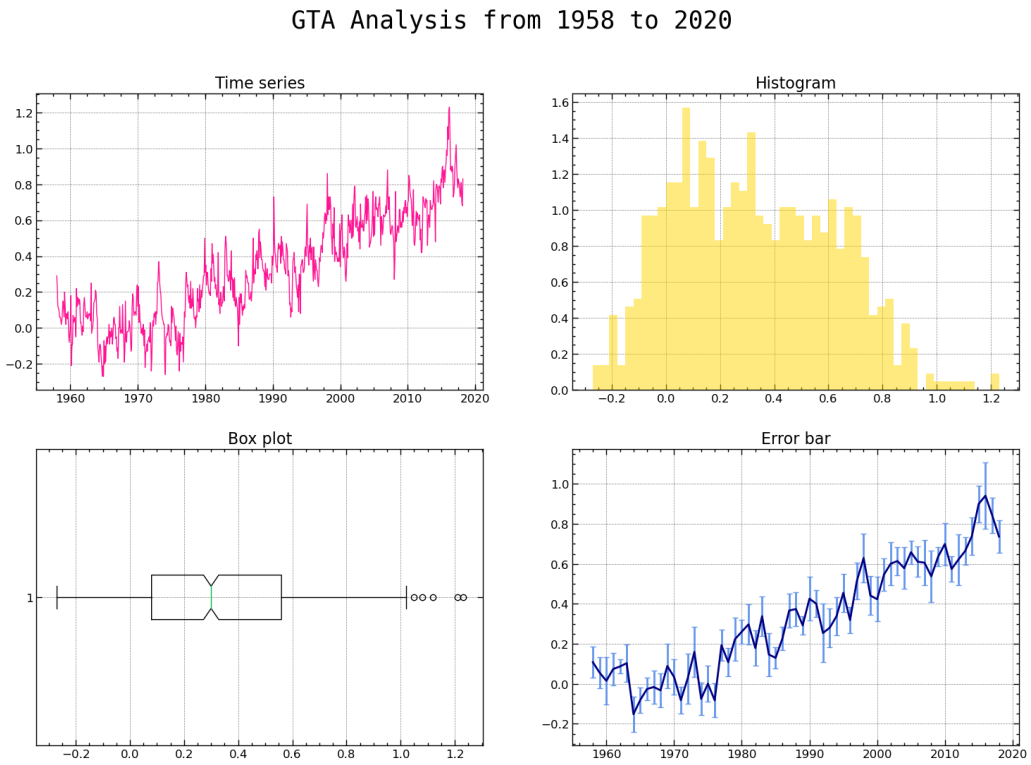

Figure 3: 4 types of plots of GTA data

From the Time Series, we can observe that the Temperature Anomaly in the world have an overall increasing trend from 1960 to 2020. The histogram suggest that the variation of temperature are distributed around 0 to 0.8 degree. The box plot shows the median at approximately 0.3 with the first quantile and third quantile at around 0.1 and 0.6 respectively. The error bar should us more clearly the trend and also the variation of each year.

3.4 Correlation

Correlation measures the strength and direction of the linear relationship between two variables. In Python, we can use the NumPy and scipy libraries to calculate various types of correlations. Here are some common types of correlations

3.4.1 Pearson Correlation

The Pearson correlation coefficient is a measure of the linear correlation between two variables. It ranges from -1 to 1, where -1 indicates a perfect negative correlation, 0 indicates no corre-

lation, and 1 indicates a perfect positive correlation. The formula for the Pearson correlation coefficient is:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{7}$$

where x and y are the two variables, n is the number of data points, $\bar{x}$ and $\bar{y}$ are the means of x and y, respectively.

In Python, we can calculate the Pearson correlation coefficient using the pearsonr function from the scipy.stats library

3.4.2 Spearman Correlation

The Spearman correlation coefficient is a nonparametric measure of the rank correlation between two variables. It ranges from -1 to 1, where -1 indicates a perfect negative monotonic correlation, 0 indicates no monotonic correlation, and 1 indicates a perfect positive monotonic correlation. The formula for the Spearman correlation coefficient is:

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \tag{8}$$

where d is the difference between the ranks of the corresponding data points.

In Python, we can calculate the Spearman correlation coefficient using the spearmanr function from the scipy.stats library

3.4.3 Kendall rank correlation

The Kendall rank correlation coefficient is another nonparametric measure of the rank correlation between two variables. It ranges from -1 to 1, where -1 indicates a perfect negative correlation, 0 indicates no correlation, and 1 indicates a perfect positive correlation. The formula for the Kendall rank correlation coefficient is:

$$\tau = \frac{2}{n(n-1)} \sum_{i<j} sgn(x_i - x_j) sgn(y_i - y_j) \tag{9}$$

where sgn is the sign function, which returns -1 if its argument is negative, 0 if its argument is zero, and 1 if its argument is positive.

In Python, we can calculate the Kendall rank correlation coefficient using the kendalltau function from the scipy.stats library.

3.4.4 Correlation between CO_2 and GTA

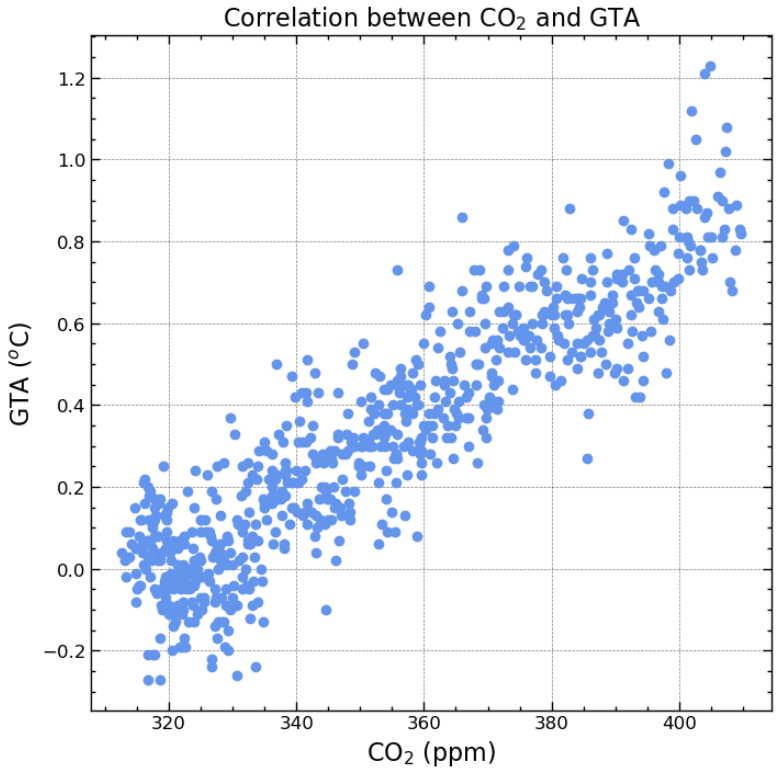

Figure 4: Correlation between CO_2 and GTA

From the figure 5, we can see that Pearson's correlation and Spearman's correlation have a high agreement with each other (0.91 and 0.9) which suggest that there is a strong linear relationship between CO_2 and GTA; however, Kendall's correlation has a lower value, which indicate that the relationship is not strictly linear as we can see on the figure. The reason why kendall can point out this is because Kendall only considers the concordant and discordant pairs in data, whereas Spearman and Pearsons correlation coefficients takes into account of the variation between values.

From the scatter plot, we can observe a strong linear correlation of CO_2 and the Global Temperature Anomaly in the period from 1960 to 2019 which is compatible with the calculation.

3.4.5 Why there is a correlation between CO_2 and GTA

Since the Industrial Revolution around 1750, a large amount of fossil fuels have been burned for generating power. This results to the increasing of CO_2, a heat-trapping gas and one of the reason for the global warming and climate change. On the other hand, the heating or cooling of

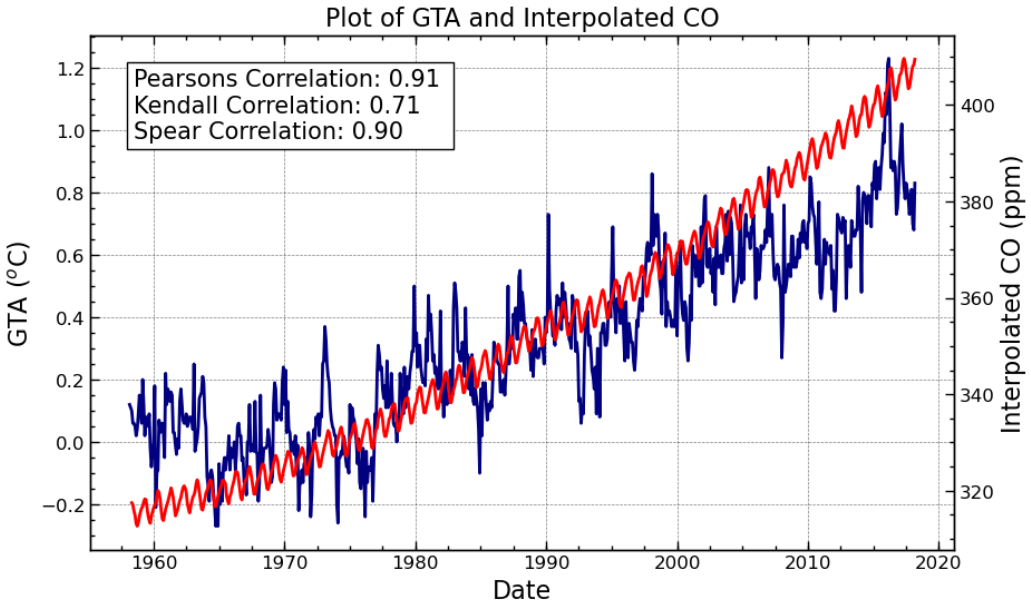

Figure 5: Time Series and correlation coefficient of CO_2 and GTA

Earth's surface can lead to changes in the concentrations of greenhouse gases in the atmosphere, which can cause additional warming or cooling (GTA).

From the data, we can confirm that, the concentration of CO_2 which used to stay betweeen 200 and 300 parts per million (ppm), today, go up to 400 ppm and still rising. Also, we can observe that the GTA has also been increasing over the years, which is consistent with the fact that the Earth's surface temperature has been rising due to the increased concentration of greenhouse gases.

The correlation between CO_2 and GTA can be explained by the fact that the concentration of greenhouse gases in the atmosphere, including CO_2, has a direct impact on the Earth's temperature and climate. The rising concentration of CO_2 due to human activities has led to an increase in the overall temperature of the Earth's surface, which in turn affects the concentration of greenhouse gases in the atmosphere.

Therefore, it is important to monitor and understand the relationship between CO_2 and GTA to better predict and mitigate the potential impacts of climate change. The correlation analysis provides a useful tool to measure the strength and direction of the relationship between these two variables, and can help inform policy decisions and actions to reduce greenhouse gas emissions and mitigate the effects of climate change.

3.5 Extra: Theil Correlation

The Theil correlation is a measure of association between two variables, similar to other correlation measures such as Pearson, Spearman, and Kendall. It was first introduced by Henri Theil in 1950 and has since been used in various fields, including economics and finance.

The Theil correlation coefficient is based on the ratio of the explained variance to the total variance in a linear regression model. It can be calculated using the following formula:

$$T = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n} Y_i^2}} \tag{10}$$

where $\hat{y}_i$ is the predicted value of the dependent variable based on the linear regression model, y_i is the observed value of the dependent variable, $\bar{y}$ is the mean of the observed values, and n is the number of observations.

The Theil correlation coefficient ranges from 0 to 1, where a value of 1 indicates a perfect relationship between the two variables, while a value of 0 indicates no relationship. It is worth noting that the Theil correlation does not assume linearity between the variables, making it a more flexible measure in certain cases.

However, the Theil correlation is less commonly used than other correlation measures, and its interpretation and application may be more complex. It may be more appropriate for certain types of data or analyses, depending on the specific research question and context.

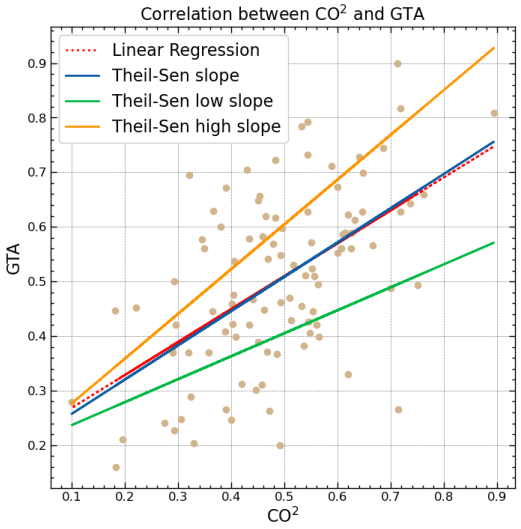

Figure 6: Theil Correlation using the `correlationexample.csv` file.

4 Probability distributions

Probability distributions are a fundamental concept in statistics that describe the probability of different outcomes in a random process. There are many types of probability distributions, including discrete and continuous distributions.

4.1 Center Limit Theorem

The Central Limit Theorem (CLT) is a fundamental result in probability theory that states that the distribution of the sum (or mean) of a large number of independent, identically distributed random variables approaches a normal distribution, regardless of the distribution of the original variables.

More formally, let $X_1, X_2, ..., X_n$ be a sequence of independent, identically distributed random variables with mean μ and standard deviation σ. Let $S_n = X_1 + X_2 + ... + X_n$ be the sum of the n random variables, and let $\bar{X}_n = S_n/n$ be the sample mean. Then, as n approaches infinity, the distribution of $\bar{X}_n$ approaches a normal distribution with mean μ and standard deviation $\sigma/\sqrt{n}$.

The CLT is a powerful tool in statistics because it allows us to approximate the distribution of a sum or mean of a large number of independent, identically distributed random variables using a normal distribution, even if the original distribution is not normal.

4.2 Law of Larger Numbers

The Law of Large Numbers (LLN) is another important result in probability theory that states that the sample mean of a large number of independent, identically distributed random variables approaches the true mean of the population as the sample size approaches infinity.

More formally, let $X_1, X_2, ..., X_n$ be a sequence of independent, identically distributed random variables with mean μ. Let $\bar{X}_n = (X_1 + X_2 + ... + X_n)/n$ be the sample mean. Then, as n approaches infinity, $\bar{X}_n$ approaches μ in probability, meaning that for any small positive number ϵ, the probability that $|\bar{X}_n - \mu| > \epsilon$ approaches zero as n approaches infinity.

The LLN is important because it provides a theoretical justification for using the sample mean as an estimator of the population mean.

4.3 The Sigma rule

The Sigma Rule, also known as the 68-95-99.7 rule, is a rule of thumb for the normal distribution that provides a quick way to estimate the proportion of observations that fall within a certain number of standard deviations from the mean.

According to the Sigma Rule:

- Approximately 68% of the observations fall within one standard deviation of the mean.
- Approximately 95% of the observations fall within two standard deviations of the mean.
- Approximately 99.7% of the observations fall within three standard deviations of the mean.

The Sigma Rule is useful for quickly estimating the proportion of observations that fall within a certain range, and can be applied to many real-world situations where the normal distribution is a reasonable approximation.

4.4 Monte Carlo simulation

Monte Carlo simulation was first introduced in the 1940s by Stanislaw Ulam and John von Neumann, who were working on the Manhattan Project, a research effort to develop the first atomic bomb. They used the simulation technique to estimate the probability of different outcomes for a complex system involving neutron diffusion, which was a critical component of the bomb.

The name "Monte Carlo" was later coined by Nicholas Metropolis, who was also working on the Manhattan Project and named the technique after the Monte Carlo Casino in Monaco, which is known for its games of chance.

Since then, Monte Carlo simulation has become a widely used method in many fields, including finance, engineering, physics, and computer science, to simulate and estimate the probability and variability of various outcomes under uncertain conditions. The technique has been further developed and refined over the years, with advances in computing power and algorithms.

Today, Monte Carlo simulation is used in a wide range of applications, such as risk analysis, optimization, design and planning, and decision-making. It has become an essential tool for modeling and analyzing complex systems or processes, and continues to be an active area of research and development.

4.4.1 Monte Carlo simulation on calculation of Pi

The Monte Carlo simulation can be used to estimate the value of Pi. The basic idea is to randomly generate a large number of points within a square, and then count the number of points that fall inside a circle inscribed within the square. The ratio of the number of points inside the circle to the total number of points generated is an estimate of the ratio of the areas of the circle to the square, which is approximately equal to Pi/4. Therefore, multiplying the estimated ratio by 4 gives an estimate of Pi.

Let n be the number of points generated, k be the number of points that fall inside the circle, and r be the radius of the circle. Then, the estimated value of Pi can be calculated as:

$$\pi \approx \frac{4k}{n} \tag{11}$$

As the number of points generated n increases, the estimated value of Pi becomes more accurate.

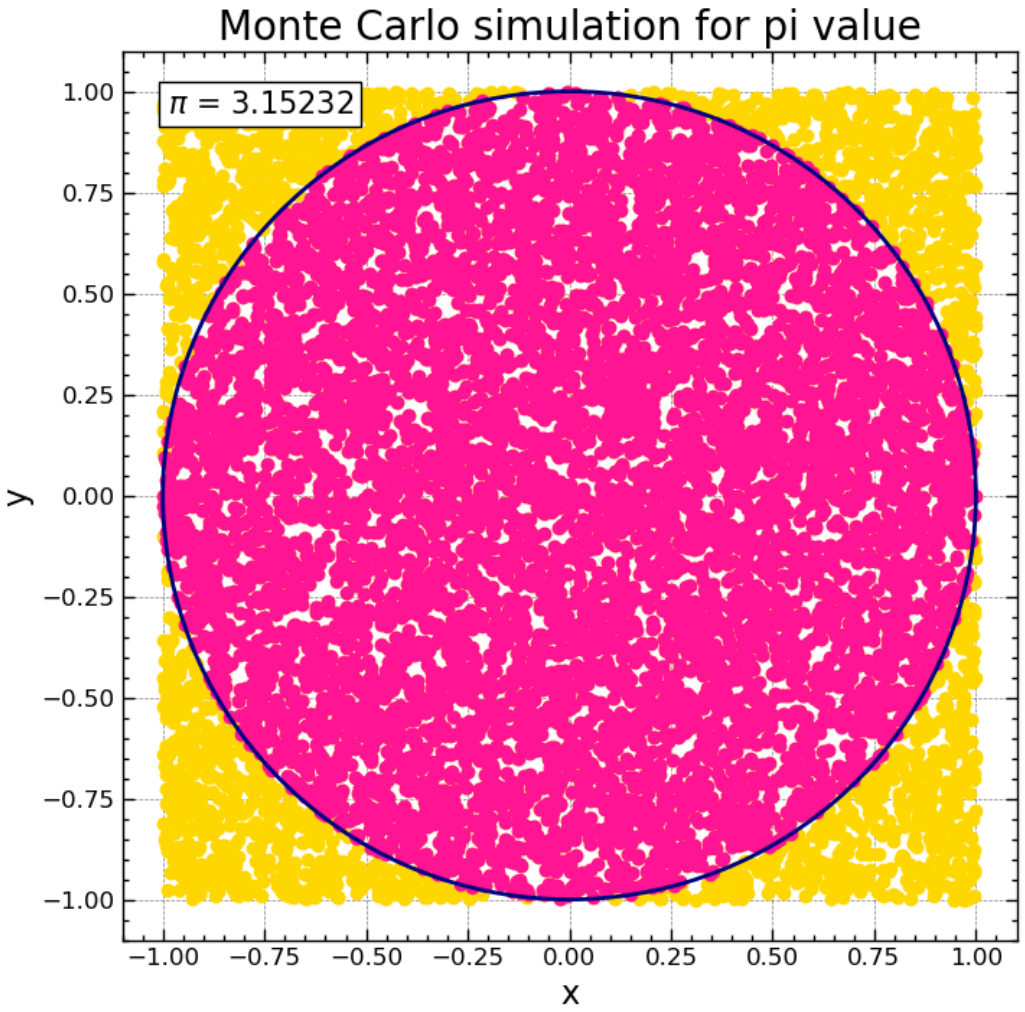

Figure 7: Caption

4.4.2 Buffon needle

Talking about monte carlo simulation, we cannot but mention about Buffon's needle. Buffon's needle problem is one of the earliest problems in which the Monte Carlo method was used to estimate a probability. The problem was first proposed by Georges-Louis Leclerc, Comte de Buffon, in the 18th century as a way to estimate the value of pi.

The Buffon's needle problem involves dropping a needle of a certain length onto a lined surface and calculating the probability that the needle will intersect one of the lines.

The probability of the needle intersecting a line can be calculated as:

$$P = \frac{k}{n} \tag{12}$$

where l is the length of the needle and d is the distance between the lines.

The value of π can then be estimated by using the Monte Carlo method to generate a large number of random placements of the needle and counting the number of times it intersects a line. The estimated value of π is:

$$\pi \approx \frac{2l}{dP} \tag{13}$$

where n is the number of trials and m is the number of times the needle intersects a line.

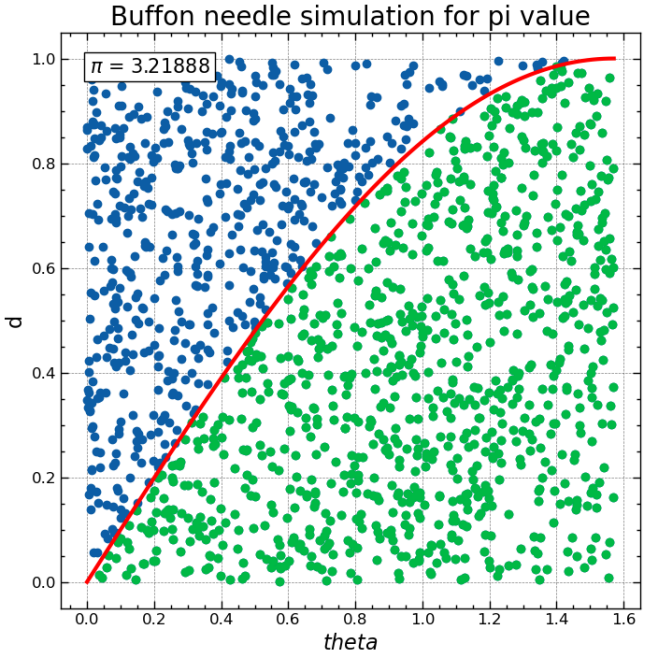

Figure 8: Buffon needle

4.4.3 Monte Carlo in finding the best parameters among the best fits

Monte Carlo simulations are particularly useful in finding the best parameters among the best fits in scientific models, including astrophysical models.

By generating random samples of the model parameters and running simulations, scientists can estimate the goodness of fit for each set of parameters and identify the set that produces the best fit to the data. This approach is particularly useful when dealing with complex or nonlinear models, where traditional analytical methods may not work.

Monte Carlo simulations can not only identify the most likely values for the model parameters, but also estimate the uncertainty or confidence interval associated with each parameter, helping scientists to better understand the underlying physical processes.

I will provide an example of fitting an exponential function using Monte Carlo to find the best fit among the best fit. We will try to find the best parameters of a and b in this equation:

$$a \times x + b \tag{14}$$

I will use the random packages to create an uniform data set for parameter x, and also generate some of noise for the data, then I use the `curvefit` function in the `scipy` package to find the best fit parameter of a and b. The figure 9 show the result. We can see that the fitting curve is actually pretty good, but as I said, it can be better.

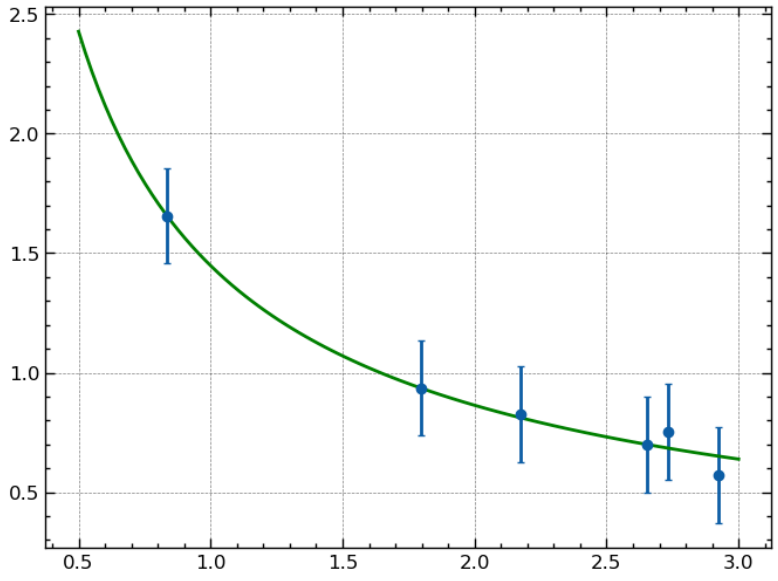

Figure 9: Curve fitting of exponential function

How can we determine which is the best fit among the best fits, we use the Monte Carlo method. To do that, we also create an uniform data set for parameter x, in this case, this data

set will remains the same in the whole process. Now, instead of fitting once, we will fit for 500 times, each times, we use a different noise with the same scale. After the fitting, we can plot the distribution of the best fit parameters of a and b, the result shows in figure 10. We can see that, according to the central limit theorem, the distribution form a gaussian distribution, which will provide us which is the best parameter among the best fits of a and b, in my case I simply pick the median.

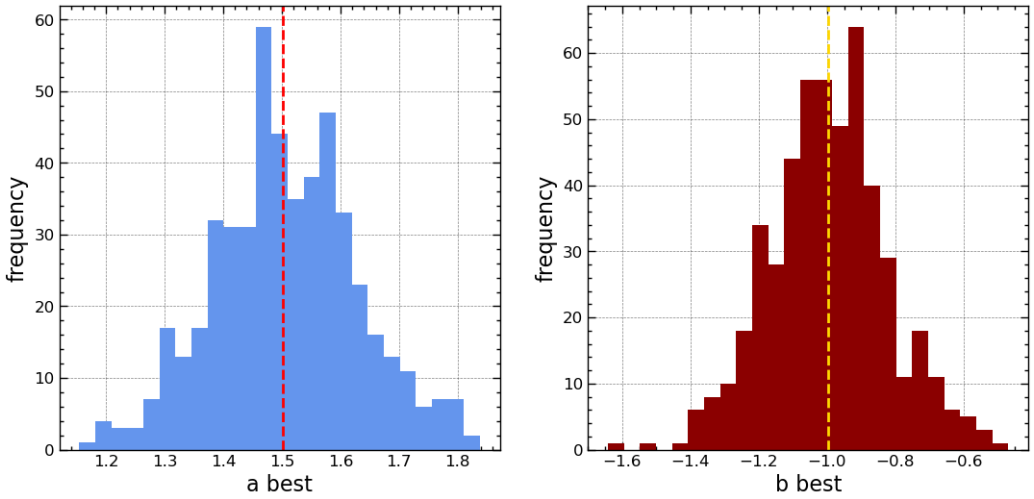

Figure 10: Distribution of best fit parameters of a and b after 500 iterations

We can also examine the correlation between the values of a best and b best.

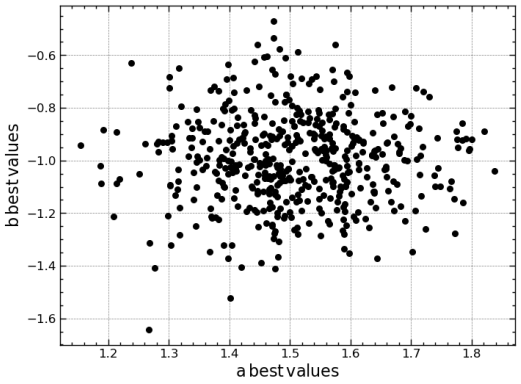

Figure 11: Correlation of a best and b best values

5 Regression and trend analysis

5.1 Regression model

Using the data provided on CO_2 and GTA provided on the previous section, as we know that the data have a strong linear relationship, we can build a model for prediction using Linear Regression in the `sklearn` packages, this function uses the least squares method to find the coefficients of the linear equation that best fits the data, this is similar to the method of `curve_fit` in `scipy` that we use before in the section 4, which is more compact, but only work for linear relationship. The linear regression coefficients can be used for prediction and inteference. The goodness-of-fit can be assessed by R-squared and mean squared error (MSE). In general, the higher value of R-squared and a lower value of MSE indicate a better fit of the model of data.

5.1.1 R-squared

R-squared, also known as the coefficient of determination, is a measure of how well the linear regression model fits the data. It measures the proportion of variation in the dependent variable (y) that is explained by the independent variable (x) in the model.

The R-squared value ranges from 0 to 1, where 0 indicates that the model does not explain any variation in the data, and 1 indicates that the model perfectly fits the data.

The formula for calculating R-squared is:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{15}$$

where n is the number of data points, y_i is the observed value, $\hat{y}_i$ is the predicted value, and $\bar{y}$ is the mean of the observed values.

5.1.2 Mean squared error

Mean squared error (MSE) is a measure of the average squared difference between the predicted and actual values in the regression model. It is a commonly used measure of the accuracy of the model's predictions.

The formula for calculating MSE is:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{16}$$

where y_i is the actual value of the dependent variable, $\hat{y}_i$ is the predicted value of the dependent variable, and n is the number of data points.

5.2 Linear Regression and Mann-Kendall regression

5.2.1 Linear Regression

In statistical modeling, regression analysis is a commonly used technique to establish the relationship between a dependent variable and one or more independent variables. We are provided the data of the temperature of Thai Binh province from 1960 and 2019, the data cleaning and mean calculation is provided in Jupyter Notebook. With this type of Time Series data, Linear Regression and Mann-Kendall Regression are the most common technique for trend analysis, in our case the independent variable is time and the dependant variable is the annual mean temperature. To perform linear regression, we can use the `sklearn` package in Python. This package provides the `LinearRegression` function which can be used to fit a linear model to the data.

The linear regression model can be represented as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \epsilon \tag{17}$$

where y is the dependent variable, $x_1, x_2, ..., x_n$ are the independent variables, $\beta_0, \beta_1, \beta_2, ..., \beta_n$ are the regression coefficients, and ϵ is the error term.

The coefficients $\beta_0, \beta_1, \beta_2, ..., \beta_n$ can be estimated using the least squares method, which minimizes the sum of the squared residuals between the predicted values and the actual values.

5.2.2 Mann-Kendall regression

In addition to linear regression, we can also use Mann-Kendall regression to analyze the trend in the time series data. Mann-Kendall regression is a non-parametric method that does not assume any particular distribution of the data. It tests for monotonic trends in the data, which can be either increasing or decreasing.

The Mann-Kendall regression model can be represented as:

$$S = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \text{sgn}(x_j - x_i) \tag{18}$$

where S is the Mann-Kendall statistic, n is the sample size, $x_1, x_2, ..., x_n$ are the observations, and sgn() is the sign function. The test statistic follows a standard normal distribution under the null hypothesis of no trend.

Mann-Kendall regression is useful for detecting trends in time series data that do not follow a linear relationship. However it may not suitable for data with seasonal trends or data that have many outliers.

5.2.3 Trend analysis of Thai Binh annual mean temperature from 1960 to 2019

From the figure 12, there is a high agreement in the slope of Linear Regression and Mann-Kendall Regression. This indicates a strong evidence of a significant increasing trend in the temperature data over the past 60 years in Thai Binh which is reasonable with the what we have worked on so far.

It is also important to note that two methods have different assumptions and limitations. Linear Regression assumes a linear relationship between variables, while Mann-Kendall Regression is a non-parametric method which is used for testing the monotonic without assuming any particular distribution of the data. Therefore, we use both methods to confirm the trend and obtain a comprehensive understanding of the data.

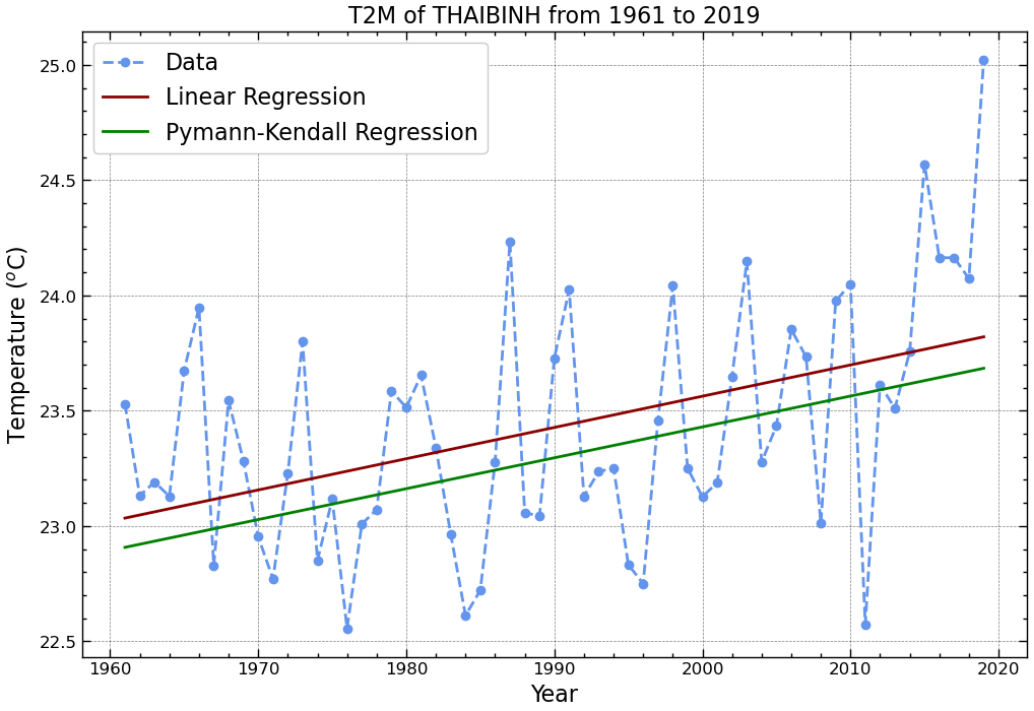

Figure 12: Linear and MK Regression of T2M of Thai Binh

5.3 Trend analysis and geographical plot

Following the method of section 5.2, we will do the same trend analysis using Linear Regression and Mann-Kendall Regression on the data of 26 provinces, the result is shown in the figure 13. It is important to notes that some missing data, so we have to remove them before conducting the model fitting. We store all the data of the slope for trend analysis and visualization, we also read the data on the coordinate of each province. To make the geographical graph, I use `cartopy` for creating the map of Vietnam and `matplotlib` to create the symbol indicating the trend for each province, the map of Linear trend and MK trend are shown in figure 14 and 15. I also plotted a mean temperature in whole period of 60 years which is presented in figure 16.

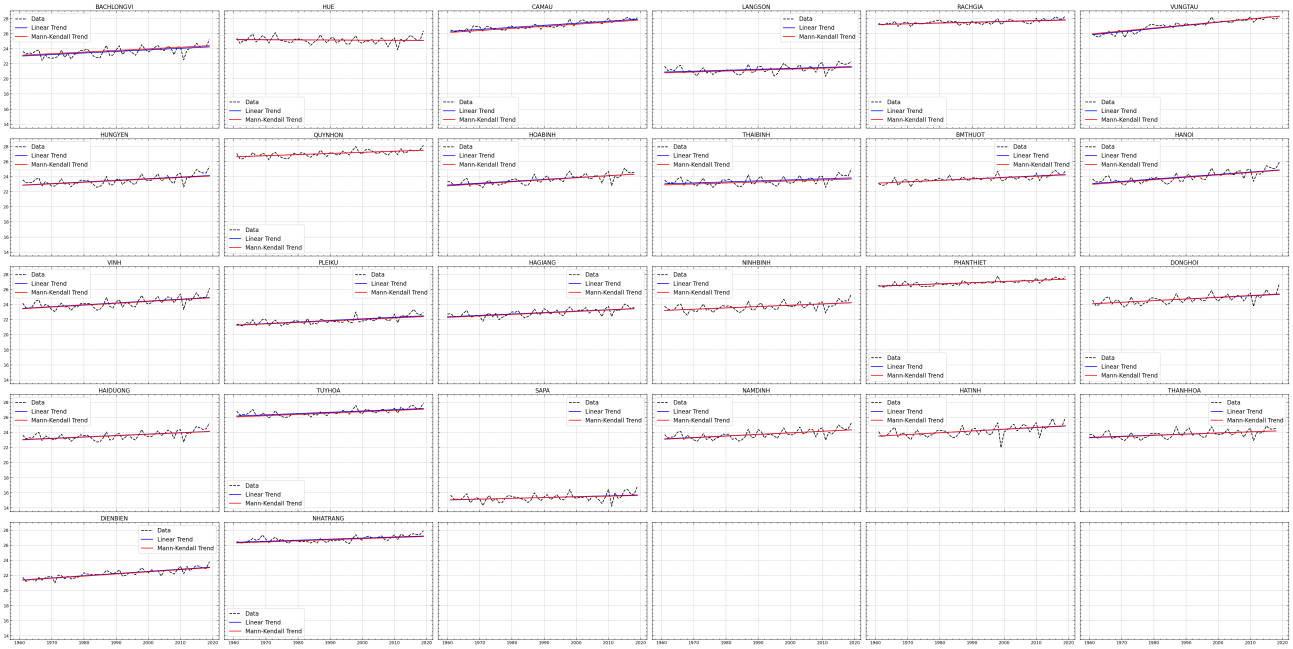

Figure 13: Trend analysis of 26 provinces

5.3.1 Temperature change of 26 provinces of Vietnam from 1960 to 2019

Based on the figures in 14 and 15, we can observe an overall increasing trend in temperature across 26 provinces of Vietnam, with the exception of Hue which shows a slight decrease. Among all the provinces, Ha Noi and Vung Tau stand out with the highest increasing temperature trends over the 60-year period. This could be attributed to the fact that Ha Noi is the capital of Vietnam and Vung Tau is located near Ho Chi Minh City, the country's largest metropolitan area, as well as Binh Duong, one of the biggest industrial regions. Industrial activities in these regions have resulted in high levels of fossil fuel consumption and CO_2 emissions, leading to increased temperatures. Other large and industrialized cities such as Vinh, Ha Tinh, Hoa Binh, and Ca Mau also show higher temperature increases compared to non-industrial cities like Ninh

Binh, Dong Hoi, Pleiku, and Buon Me Thuot. The smallest temperature increases are seen in mountainous areas with fewer populations, such as Ha Giang, Quy Nhon, and Tuy Hoa.

Furthermore, the trend analysis using both Linear Regression and Mann-Kendall Regression techniques has shown high precision, with only small discrepancies observed in Buon Me Thuot or Ha Giang. This indicates that both methods can be used to accurately predict the temperature trends in Vietnam.

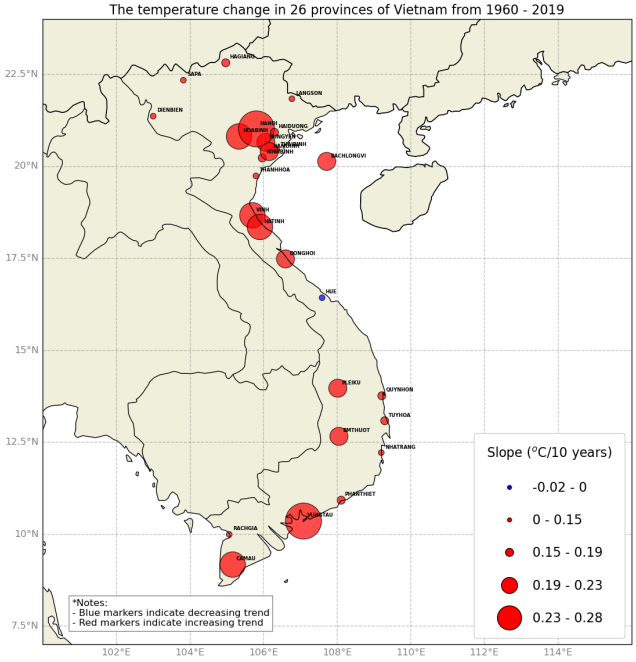

Figure 14: Linear Trend

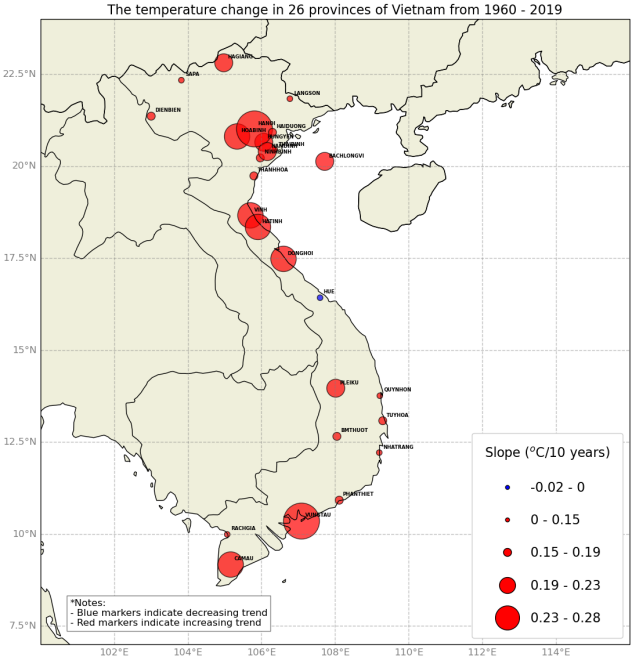

Figure 15: MK trend

In addition, I also analyzed the provided data to calculate the average temperature over the entire 60-year period, from 1960 to 2019. As seen in Figure 16, it is evident that the southern regions of Vietnam generally experience higher temperatures than the northern regions. This finding is consistent with the principle that temperatures tend to decrease as we move towards higher latitudes, due to the angle of solar radiation.

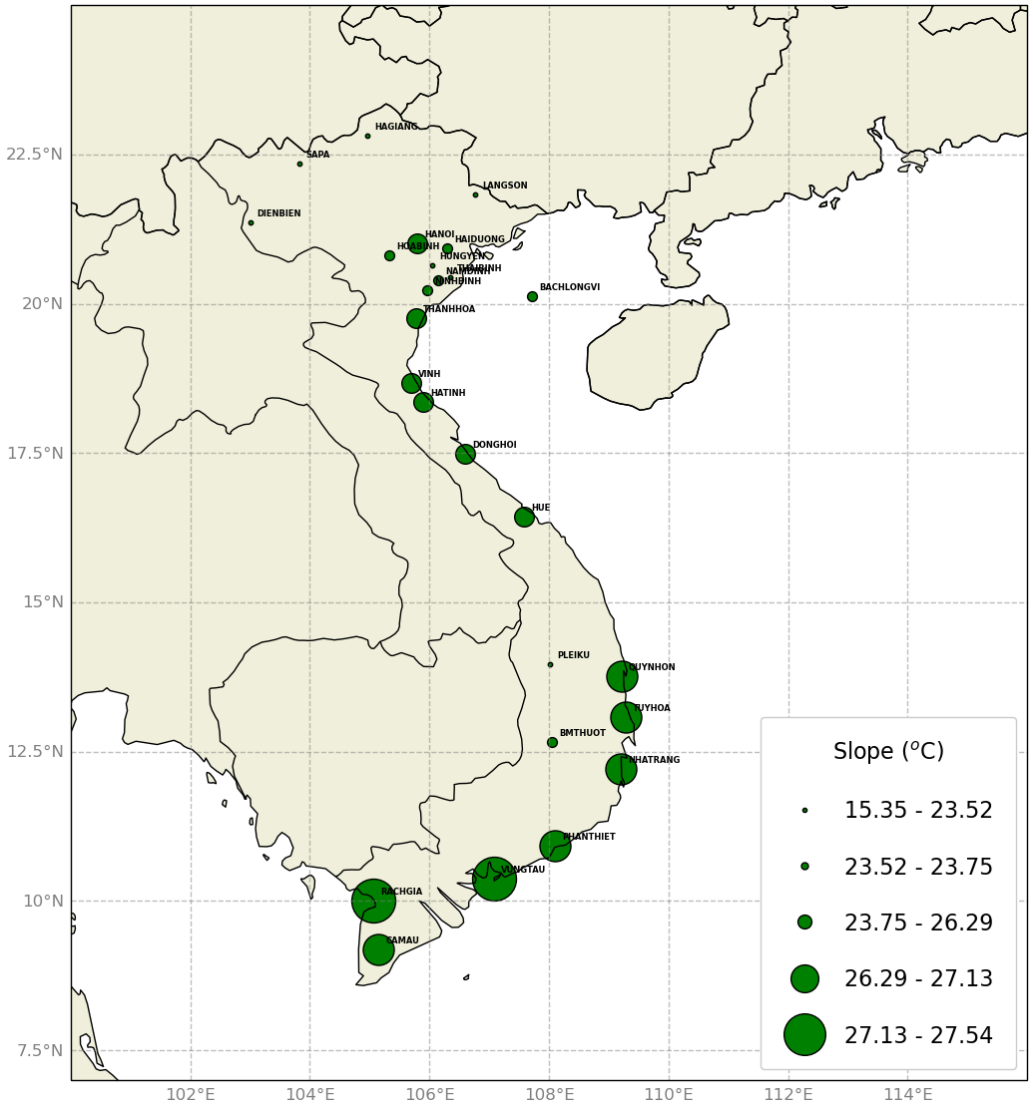

Figure 16: Mean Temperature of the period 1960-2019

6 Hypothesis testing and parameter estimation (model fitting)

Hypothesis testing and parameter estimation are important statistical techniques used to analyze data and draw conclusions. In this section, we will discuss two commonly used methods for hypothesis testing on mean temperature data of Thai Binh, namely T-test and Wilcoxon test.

6.1 Hypothesis testing on mean temperature of Thai Binh

Hypothesis testing is a statistical method used to make decisions about a population based on sample data. The process involves formulating a null hypothesis and an alternative hypothesis, collecting sample data, and using statistical tests to determine the likelihood of the null hypothesis being true. If the results of the tests show that the null hypothesis is unlikely to be true, the alternative hypothesis is accepted. On the other hand, if the results fail to reject the null hypothesis, it is concluded that there is not enough evidence to support the alternative hypothesis.

6.1.1 Null Hypothesis

In hypothesis testing, the null hypothesis (H_0) is a statement that there is no significant difference between two groups, or that there is no effect of a particular treatment or intervention. The alternative hypothesis (H_a) is a statement that there is a significant difference between two groups, or that there is an effect of a particular treatment or intervention. In hypothesis testing, the null hypothesis is usually denoted by H_0.

6.1.2 T-test

T-test is a statistical test used to compare the means of two groups. In the case of mean temperature of Thai Binh, we can use a one-sample T-test to test whether the mean temperature is significantly different from a certain value. The T-test is based on the assumption that the sample data follows a normal distribution.

The formula for T-test is:

$$t = \frac{\bar{x} - \mu}{s/\sqrt{n}} \tag{19}$$

where $\bar{x}$ is the sample mean, μ is the hypothesized population mean, s is the sample standard deviation, and n is the sample size.

The null hypothesis H_0 is that the mean temperature of Thai Binh is equal to the hypothesized population mean μ, while the alternative hypothesis H_a is that the mean temperature is different from μ. We can reject H_0 if the calculated T-statistic is greater than the critical value of T at a certain level of significance, usually 0.05.

The `scipy` package in Python provides functions to perform T-test for one sample and two samples. To perform T-test for one sample, the function `ttest_1samp` in the `stats` module of the `scipy` package can be used. To perform T-test for two samples, the function `ttest_ind` in the same module can be used.

6.1.3 Wilcoxon test

Wilcoxon test, also known as Mann-Whitney U test, is a non-parametric statistical test used to compare the medians of two groups. Unlike T-test, Wilcoxon test does not assume that the sample data follows a normal distribution. In the case of mean temperature of Thai Binh, we can use a one-sample Wilcoxon test to test whether the median temperature is significantly different from a certain value.

The formula for Wilcoxon test is:

$$W = \sum_{i=1}^{n} R_i \text{sgn}(X_i - Y_i) \tag{20}$$

where n_1 is the sample size of Thai Binh, n_2 is the hypothesized sample size, R_1 is the sum of the ranks of the Thai Binh sample.

The null hypothesis H_0 is that the median temperature of Thai Binh is equal to the hypothesized median, while the alternative hypothesis H_a is that the median temperature is different from the hypothesized median. We can reject H_0 if the calculated U-statistic is smaller than the critical value of U at a certain level of significance, usually 0.05.

In Python, wilcoxon test can be performed using the function wilcoxon in the stats module of the scipy package.

6.1.4 Mean temperature of Thai Binh with a selected temperature

In this problem, t-test and wilcoxon test is conducted to test whether T2mean of Thai Binh during 1961-2019 is significantly different with the temperature values of 23^{o}C, 23.5^{o}C and 24^{o}C. The code for this is pretty simple, after import the function into the Jupyter Notebook, with the data in `df_TB['T2m TB']` is the annual mean temperature of Thai Binh from 1961 to 2019, `temp` is the temperature value we need to test, `p_t` and `p_w` is the p-value of T-test and Wilcoxon test.

```
##T_test
t, p_t = ttest_1samp(df_TB['T2m TB'], temp)

##Wilcoxon test
w, p_w = wilcoxon(df_TB['T2m TB'] - temp)
```

```
T-test with 23°C with p-value = 3.99e-08: Reject null hypothesis
Wilcoxon with 23°C with p-value = 3.10e-07: Reject null hypothesis

T-test with 23.5°C with p-value = 2.88e-01: Fail to reject null hypothesis
Wilcoxon with 23.5°C with p-value = 1.74e-01: Fail to reject null hypothesis
```

```
T-test with 24°C with p-value = 1.01e-11: Reject null hypothesis
Wilcoxon with 24°C with p-value = 1.15e-08: Reject null hypothesis
```

After conducting hypothesis testing on the three selected temperatures with the null hypothesis is that the chosen temperature is equal to the mean value, only 23.5^{o}C failed to reject the null hypothesis, indicating that there is insufficient evidence to suggest a significant difference between 23.5^{o}C and the mean temperature of Thai Binh over the 60-year period. We can confirm that by calculation of the mean temperature of Thai Binh, the result is shown in figure 17. As we can see from the figure, 23.5^{o}C is pretty close to the real mean value (23.43^{o}C), which is the reason why the t-test on 23.5^{o}C is failed to reject the null hypothesis.

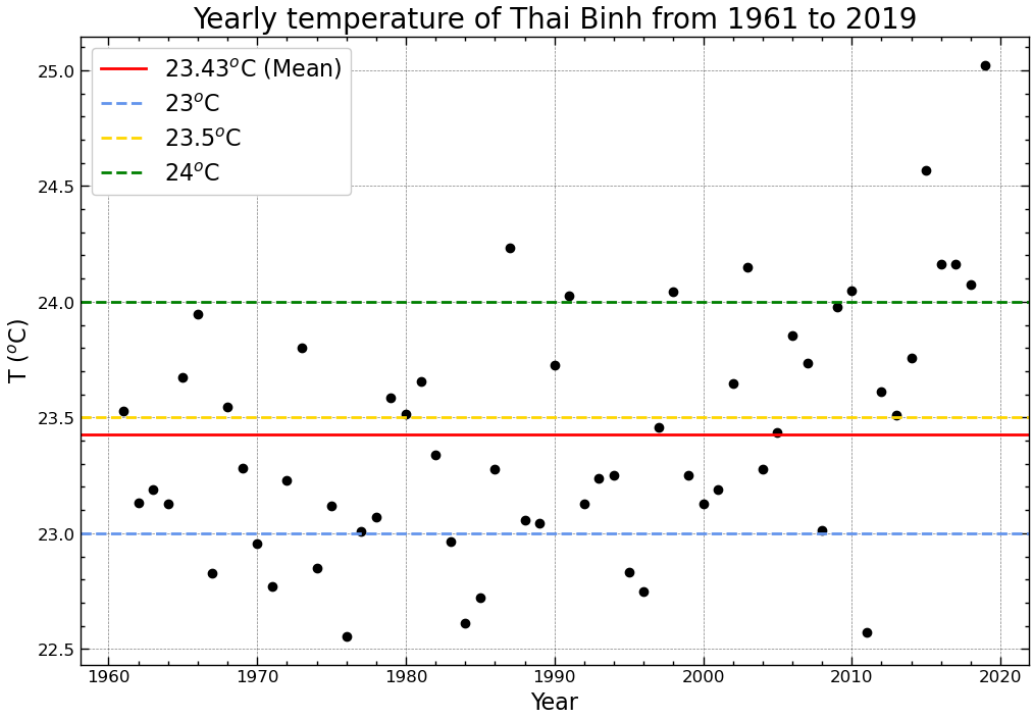

Figure 17: Mean temperature and three selected temperature values of Thai Binh in 1960-2019

6.1.5 Temperature of Thai Binh on two periods

In this section, we will explore how we can use t-test answer the question that "Does the T2mean during the period of 1961-1980 significantly differ from that during the period of 2000-2019". The code for t-test of two sample is provided with period1 and period2 is the T2M in the interval of 1961-1980 and 1980-2019, `equal_var = False` indicates that the samples have unequal variances:

```
t, p = ttest_ind(period1, period2, equal_var=False)
```

```
T-test for T2mean in 1961 - 1980 and 2000 - 2019 with p-value:
p1 = 2.13e-22 and p2 = 1.96e-03
Reject the null hypothesis.
```

Why there are two values of p-value in our analysis? Because we just did two-direction (two-tailed) t-test. The two-direction t-test, also known as two-tailed t-test, is a statistical test used to determine if there is a significant difference between two groups of data. Unlike one-tailed t-test, which only tests for a significant difference in one direction, the two-tailed t-test tests for a significant difference in both directions. The null hypothesis is that the mean temperature of Thai Binh is equal to that of Hanoi.

In my case, I have performed a two-tailed t-test on the temperature data of Thai Binh for two different time periods: 1961-1980 and 1980-2019. The p-value for the first period (1961-1980) is 2.13e-22, which is very small. This indicates that there is a significant difference between the temperature in this period and the overall mean temperature over the 60-year period. The p-value for the second period (1980-2019) is 1.96e-03, which is also small. This indicates that there is a significant difference between the temperature in this period and the overall mean temperature over the 60-year period.

Since the p-values are less than the significance level (usually 0.05), I can reject the null hypothesis and conclude that there is a significant difference in temperature between these two time periods. I can confirm this result using the figure 18, the mean value, the distribution and also time series of mean temperature in two period are very different.

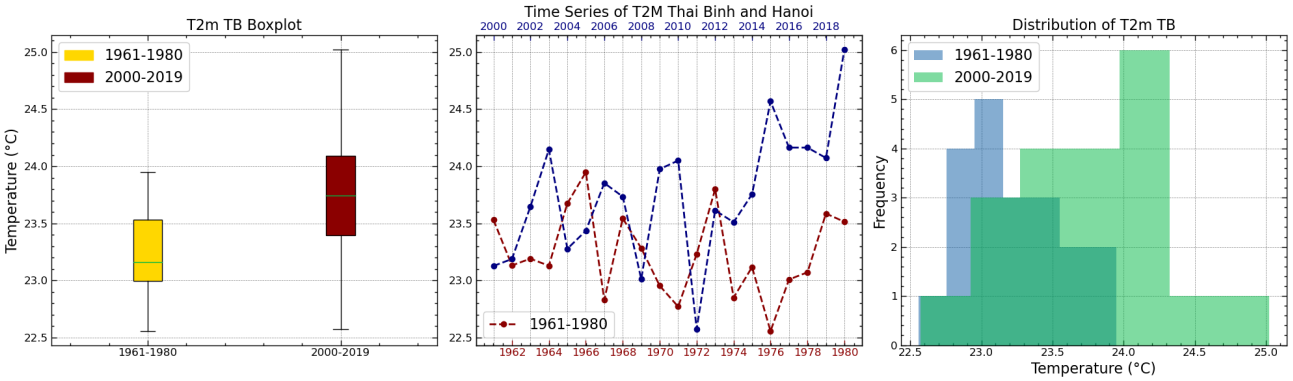

Figure 18: T2M of Thai Binh in 2 period for confirming T-test

6.2 Hypothesis testing on mean temperature of Thai Binh and Ha Noi

Similar to what we do with T2M of Thai Binh in 2 period, now we try to examine whether the mean temperature of Thai Binh stations significantly differ from that of Ha Noi stations during three selected period of 1961-1980, 1980-2019 and 1961-2019. To solve this problem, we use T-test for two samples of Thai Binh and Ha Noi on each selected periods.

6.2.1 Period 1961-1980

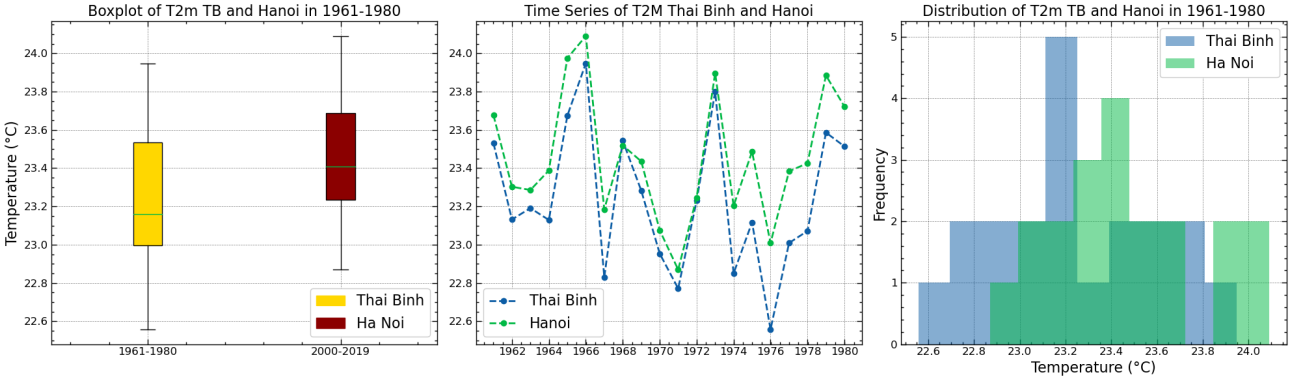

Figure 19: T2M of Thai Binh and Hanoi from 1961 to 1980

Based on the result from the T-test (The code is provided in Jupyter Notebook), the p-value for p1 is 1.00e+00, which is much greater than the significance level of 0.05. This indicates that there is not enough evidence to reject the null hypothesis that the means of T2 for Thai Binh and Hanoi stations are equal during the period of 1961-1980.

The p-value for p2 is 5.83e-02, which is greater than the significance level of 0.05, but still relatively small. This suggests that there may be some evidence to suggest that the means of T2 for Thai Binh and Hanoi stations are different during the period of 1961-1980, but this evidence is not strong enough to reject the null hypothesis.

Therefore, based on these results, we cannot conclude that the mean T2 for Thai Binh and Hanoi stations are significantly different for the period of 1961-1980. Compared with the figure 19, we can observe the similarity in the temperature of two provinces each year, even though the temperature of Ha Noi is slightly higher.

6.2.2 Period 1981-2019

Similarly, In the T-test for T2M of Thai Binh and Hanoi during 1981-2019, The p-value for p1 is 1.00e+00, which is much greater than the significance level of 0.05. This indicates that there

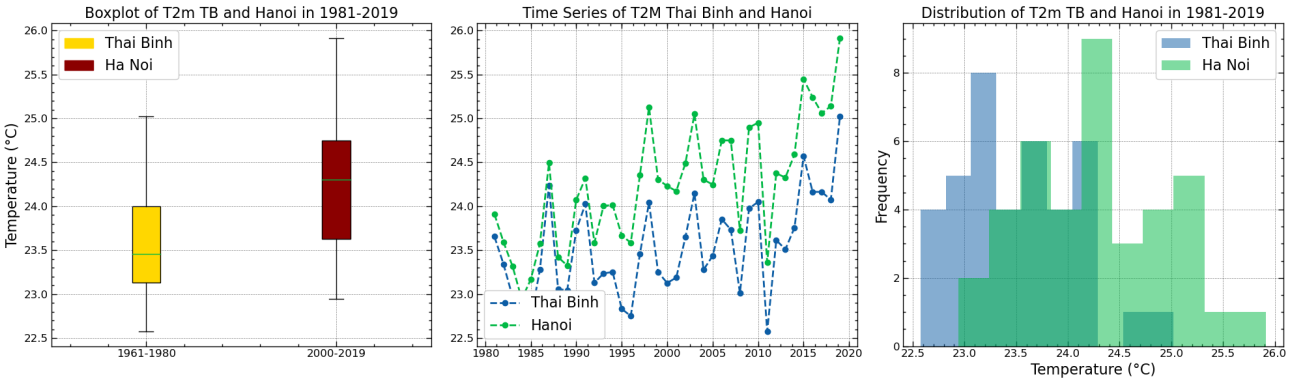

Figure 20: T2M of Thai Binh and Hanoi from 1981 to 2019

is not enough evidence to reject the null hypothesis that the means of T2 for Thai Binh and Hanoi stations are equal during the period of 1981-2019.

The p-value for p2 is 2.98e-06, which is much smaller than the significance level of 0.05. This suggests that there is strong evidence to suggest that the means of T2 for Thai Binh and Hanoi stations are different during the period of 1981-2019, and we can reject the null hypothesis at the 5

Therefore, based on these results, we can conclude that the mean T2 for Thai Binh and Hanoi stations are significantly different for the period of 1981-2019. Based on figure 20, there is a discrepancy between the temperature of Ha Noi and Thai Binh, which suggest that the mean temperature of those two provinces is significantly different.

6.2.3 Period 1961-2019

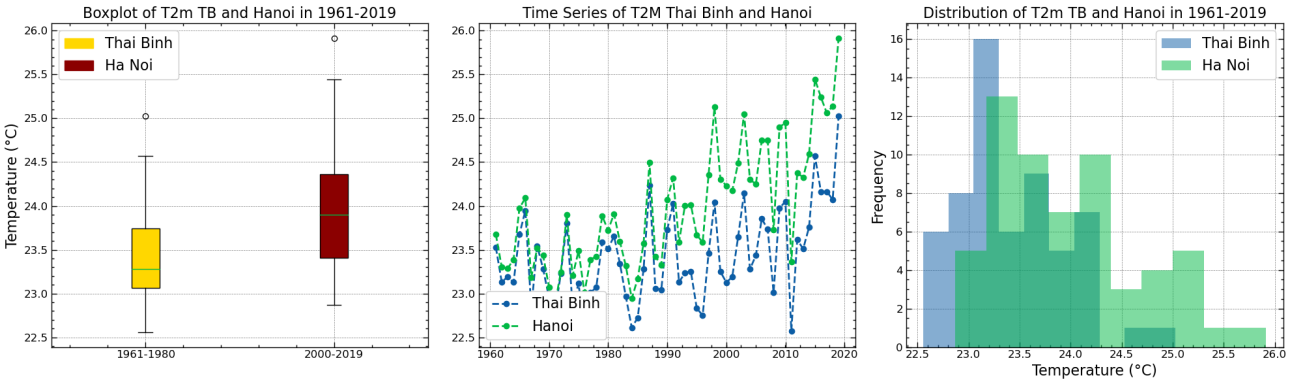

Figure 21: T2M of Thai Binh and Hanoi from 1961 to 2019

In the T-test for the whole 60-year period, the p-value for p1 is 1.00e+00, which is much greater

than the significance level of 0.05. This indicates that there is not enough evidence to reject the null hypothesis that the means of T2 for Thai Binh and Hanoi stations are equal during the entire period of 1961-2019.

The p-value for p2 is 4.34e-06, which is much smaller than the significance level of 0.05. This suggests that there is strong evidence to suggest that the means of T2 for Thai Binh and Hanoi stations are different during the entire period of 1961-2019, and we can reject the null hypothesis at the 5

Therefore, based on these results, we can conclude that the mean T2 for Thai Binh and Hanoi stations are significantly different for the entire period of 1961-2019. We can compare the result with figure 21.

7 Dimensionality reduction (PCA)

7.1 PCA

7.1.1 What is PCA?

PCA (Principal Component Analysis) is a statistical method used for **dimensionality reduction** of a large set of variables while retaining the most important features or patterns in the data. It is often used for data visualization, exploratory data analysis, and feature extraction.

The language of PCA is linear algebra. Basically, PCA is a linear transformation technique which transform the high-dimensional dataset into a low-dimensional dataset. This technique can be done by indentifying the **principal components**, the directions in that high-dimensional space which capture the most variation of the data, and projecting the data onto those principal components.

The number of principal components is equal to the number of variables in the original data set. The first principal component (PC1) is the direction capturing the most variation of data, the second principal component (PC2) is the direction capturing the second most variation of data, and so on. Based on the structure of the data, The number of principal components k is typically chosen based on the amount of variation that they capture. For example, we might choose to retain the first k principal components that capture 90% of the variation in the data.

7.1.2 How can we do the transformation?

By finding the eigenvector and eigenvalue of the covariance matrix of the data. The eigenvectors represent the directions of the principal components, and the corresponding eigenvalues represent the amount of variation captured by each principal component.

Let's say we have a high-dimensional dataset with n observations and p variables. We can

represent the dataset as an $n \times p$ matrix X. The first step in PCA is to center the data by subtracting the mean of each variable from each observation:

$$\widetilde{X} = X - \bar{X} \tag{21}$$

where $\bar{X}$ is the $1 \times p$ mean vector of X.

Next, we calculate the covariance matrix of $\widetilde{X}$:

$$S = \frac{1}{n-1}\widetilde{X}^T\widetilde{X} \tag{22}$$

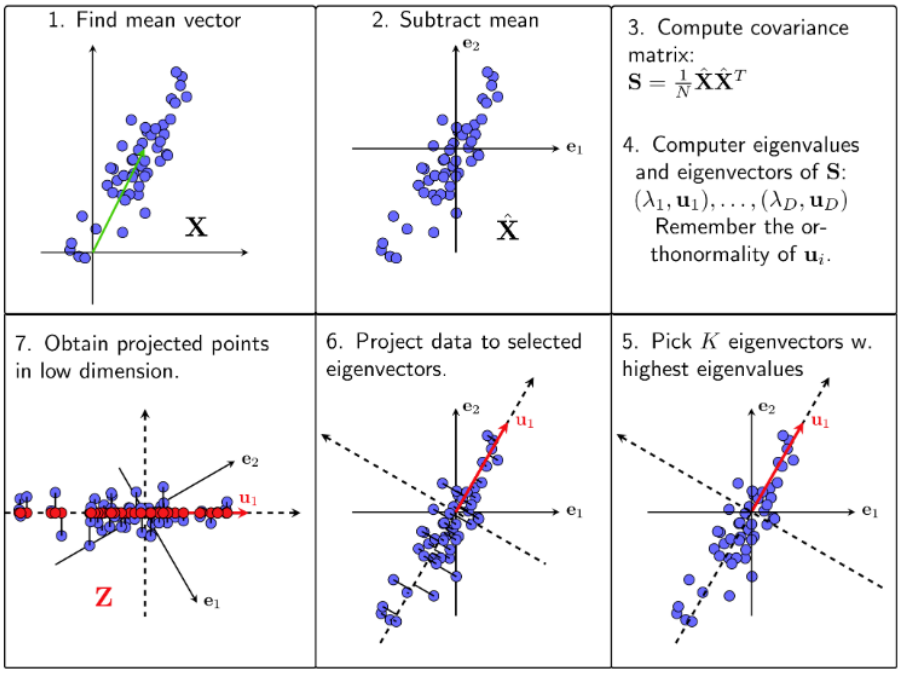

Figure 22: The procedure of PCA

The eigenvectors and eigenvalues of S can be found using an eigenvalue decomposition:

$$Sv = \lambda v \tag{23}$$

where v is a $p \times 1$ eigenvector and λ is the corresponding eigenvalue.

The eigenvectors v are the directions of the principal components, and the eigenvalues λ represent the amount of variation captured by each principal component.

We can order the eigenvectors and eigenvalues by the size of the eigenvalues, so that the first eigenvector corresponds to the direction that captures the most variation, the second eigenvector corresponds to the direction that captures the second most variation, and so on.

Finally, we can transform the data into the low-dimensional space by projecting it onto the first k principal components:

$$Z = \widetilde{X} V_k \tag{24}$$

where V_k is a $p \times k$ matrix containing the first k eigenvectors of S.

The resulting matrix Z has n rows (one for each observation) and k columns (one for each principal component).

7.2 PCA on the correlation example

The correlation example contains two columns of data, by conducting the PCA, we can find out the centroid, the eigenvalues and eigenvectors of the first principal components (PC1) shown in pink and second principal components (PC2) shown in blue, in the figure 23. The data then will be projected on those two axises, the axis rotate and the data is recovered again in a new coordinate shown in the second plot of figure 23

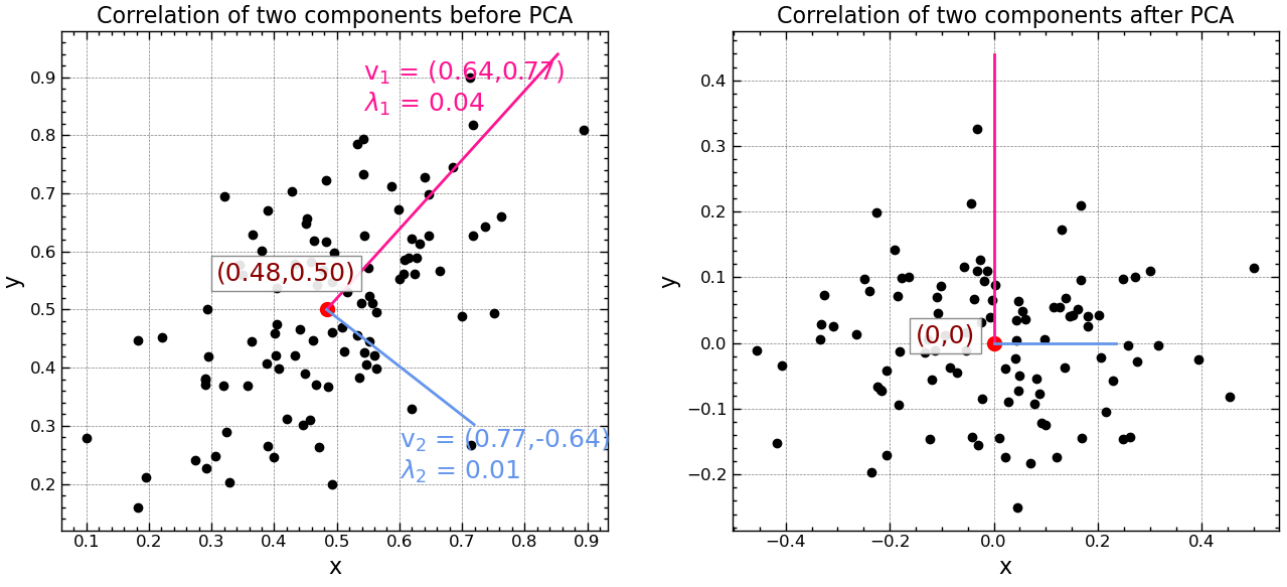

Figure 23: How the PCA transforms the data

7.3 PCA on False color image of Pluto

Similar to the previous problem, in this problem, the image of false pluto with 3 band colors with the size 1080000 can be reduced to a smaller one with the size of 360000 using PCA method, without sacrifice the resolution.

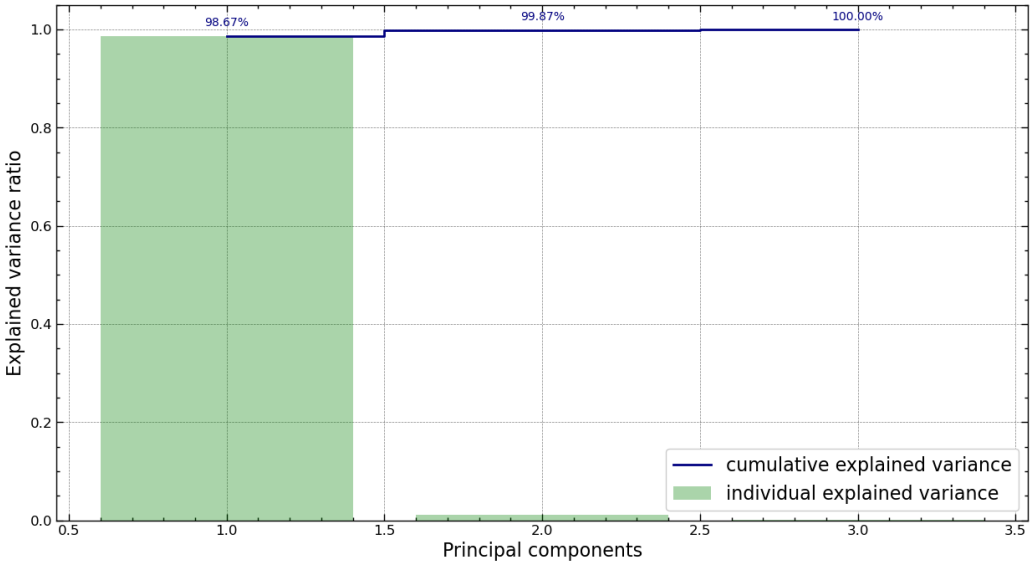

Figure 24: Scree plot and cumulative variances

From the figure 24, it is suggested that 99.87% of the data are captured by PC1 and PC2, so PC3 can be ignored. Therefore, we can do the transformation with 2 components which can efficiently reduce the complexity of the data. Also, the PC1 contains most of the information (98.67%) compare with PC2, the result is shown in figure 25, PC1 image has a really high resolution, while PC2 image is quite blur.

With this insights, we can simply take the PC1 components with the size of 360000 and compare to the original image with the size of 108000 to see how good of our method. The original false color image of Pluto has 3 band, in order to convert it to the grayscale, we use the method used here is called grayscale conversion or luminosity method. It converts a color image to a grayscale image by taking a weighted sum of the red, green, and blue color channels, where the weights are based on the perceived intensity of each color channel. The formula used here is a commonly used formula in digital image processing, which assigns a weight of 0.2989 to the red channel, 0.5870 to the green channel, and 0.1140 to the blue channel. The result is presented in the figure 26.

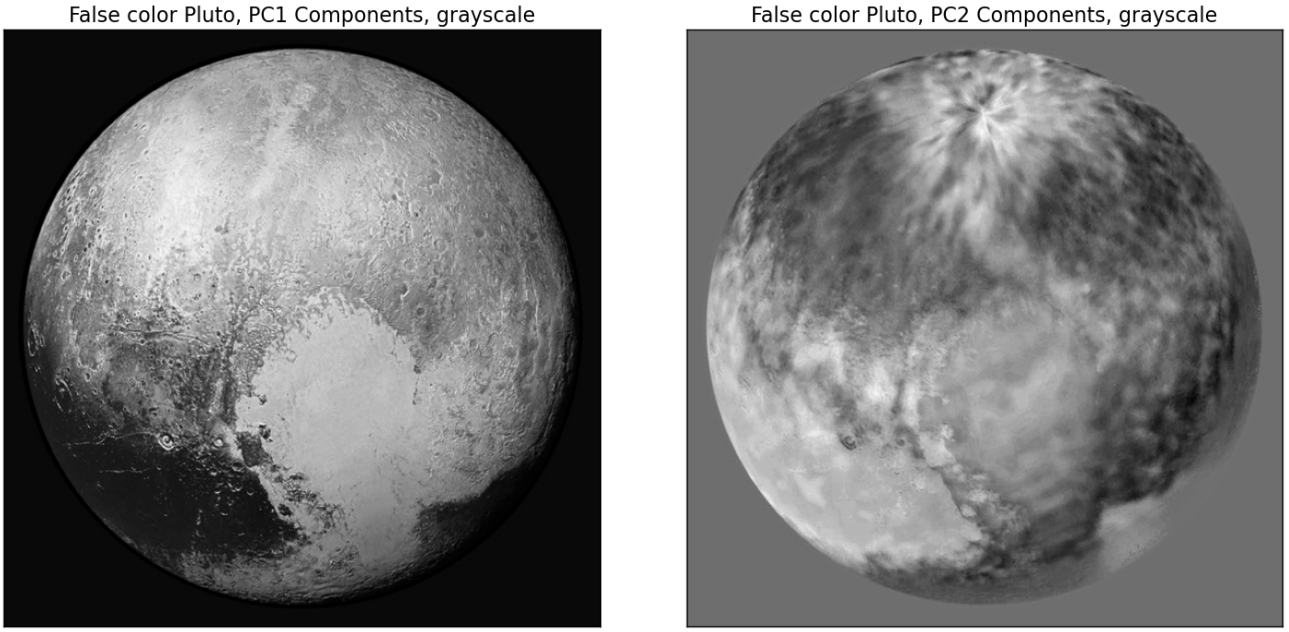

Figure 25: First and Second components of false image of Pluto

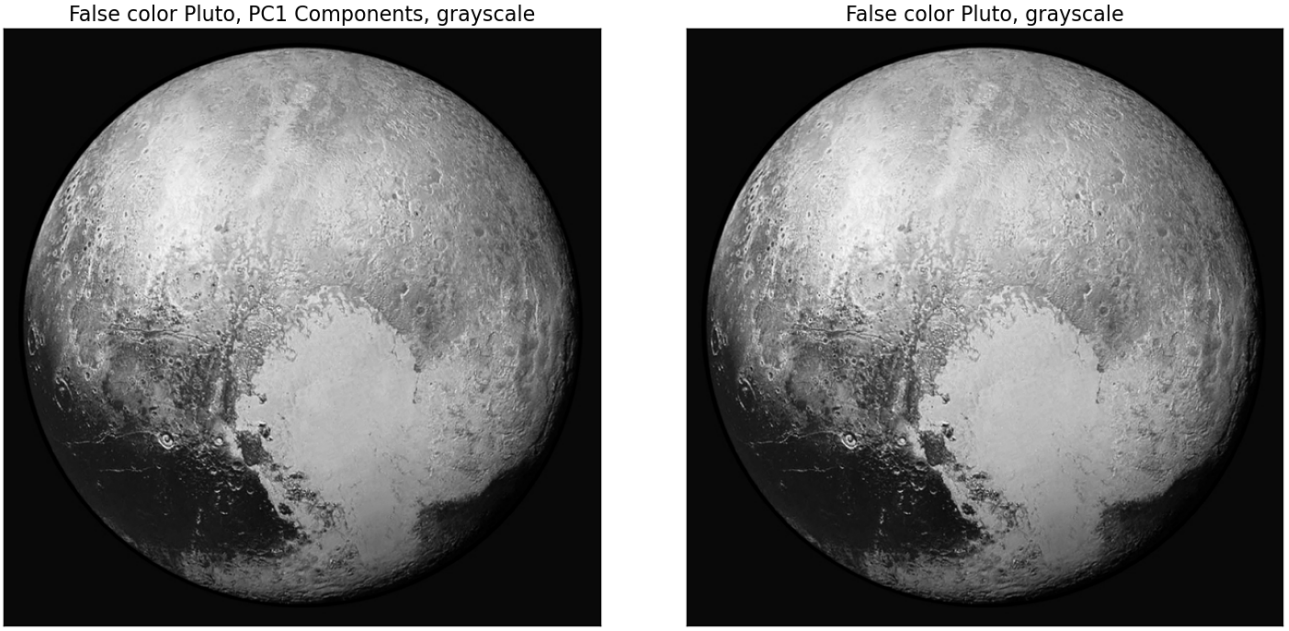

Figure 26: Comparison of the first components of false color pluto and the original data

7.4 Landsat

Landsat is a series of Earth observation satellites that provide valuable data for environmental monitoring, land use planning, and natural resource management. The Landsat program is managed jointly by NASA and the United States Geological Survey (USGS), and has been in operation since 1972. The Landsat satellites capture images of the Earth's surface in different spectral bands, allowing scientists and researchers to study changes over time.

7.4.1 Landsat images

Landsat images are composed of multiple spectral bands, each corresponding to a different range of wavelengths of the electromagnetic spectrum. The Landsat 8 satellite, for example, captures images in eleven spectral bands ranging from the visible to the thermal infrared. The spectral bands of Landsat images are often combined to create false-color composites, which can highlight different features of the Earth's surface such as vegetation, water bodies, and urban areas.

In this problem, we make use of the data of a part of Shenzhen, a major sub-provincial city and one of the special economic zones of China with 6 bands, The data is shown in figure 27.

- Band 1: blue (0.45-0.52 μm)
- Band 2: green (0.52-0.60 μm)
- Band 3: red (0.63-0.69 μm)
- Band 4: near-infrared (0.76-0.90 μm)
- Band 5: shortwave-infrared 1 (1.55-1.75 μm)
- Band 7: shortwave-infrared 2 (2.08-2.35 μm)

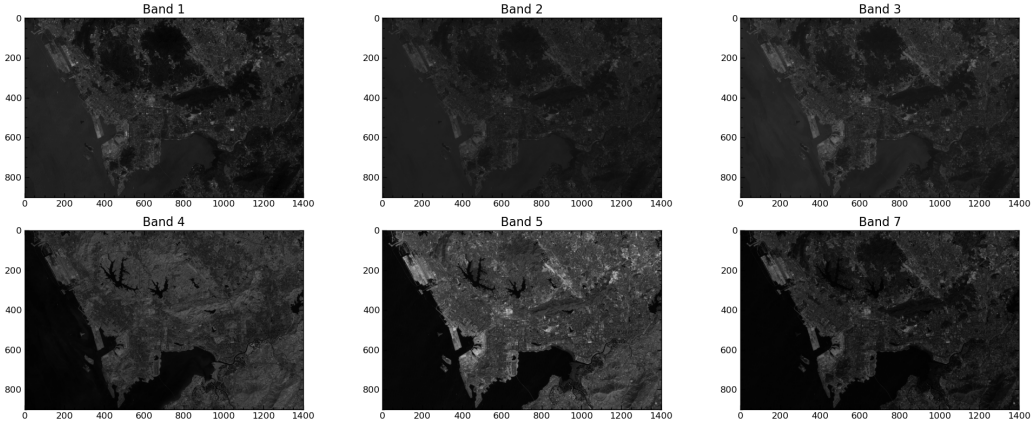

Figure 27: Landsat images in 6 bands

We can examine the scatter plot (histogram plot) to visualize the correlation of each bands with these others. The result is shown in figure 29. Also, the correlation can be represented by the correlation matrix or the heat map shown in figure 28 where we can see that band 1, 2 and 3 have really high correlation with each others because those 3 bands are in the same visible range, while band 4, 5 and 7 have high correlation with each others because they are in the same infrared range.

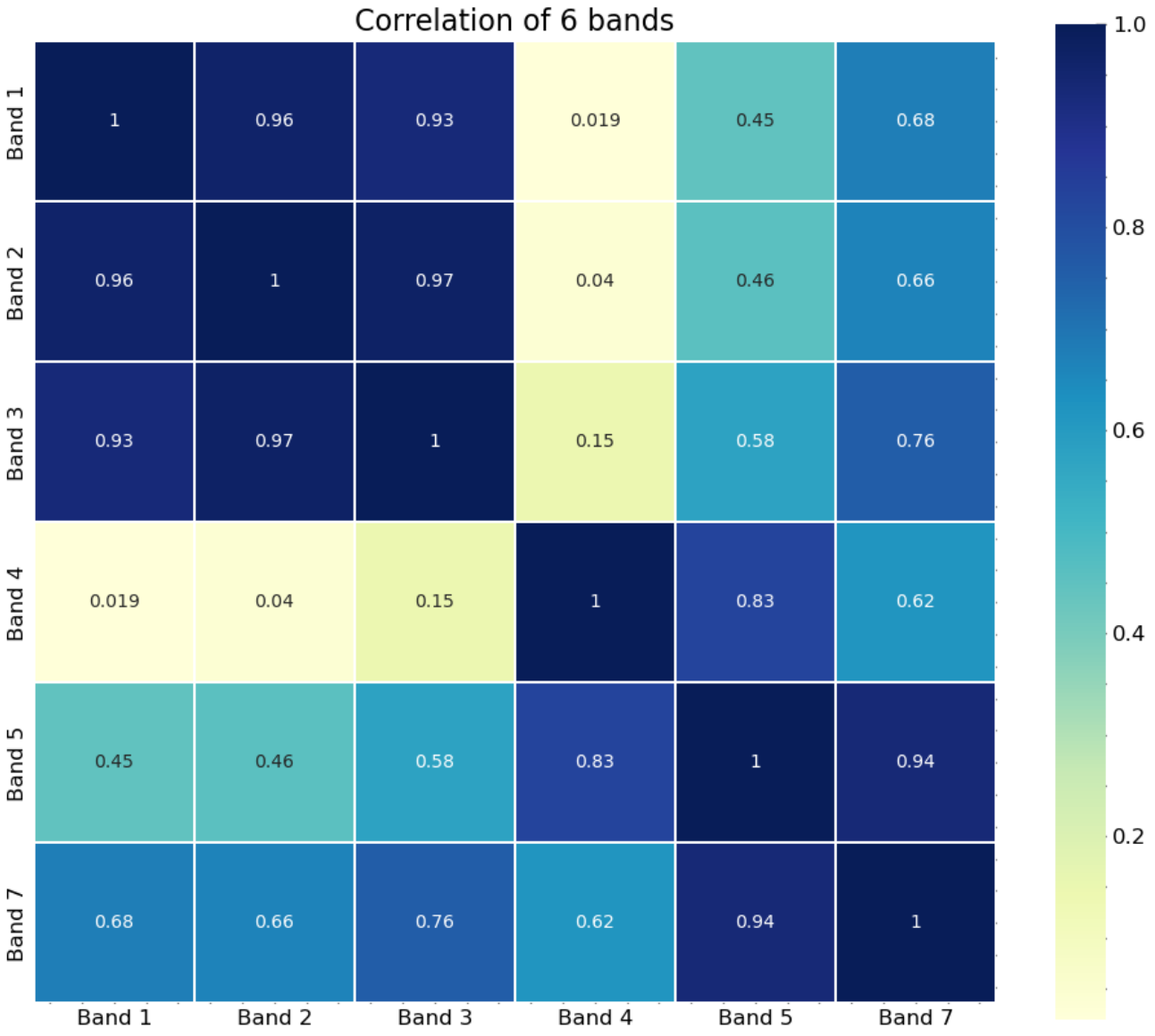

Figure 28: Caption

Similar to previous problem of Pluto, we can also apply the PCA method to find the principal

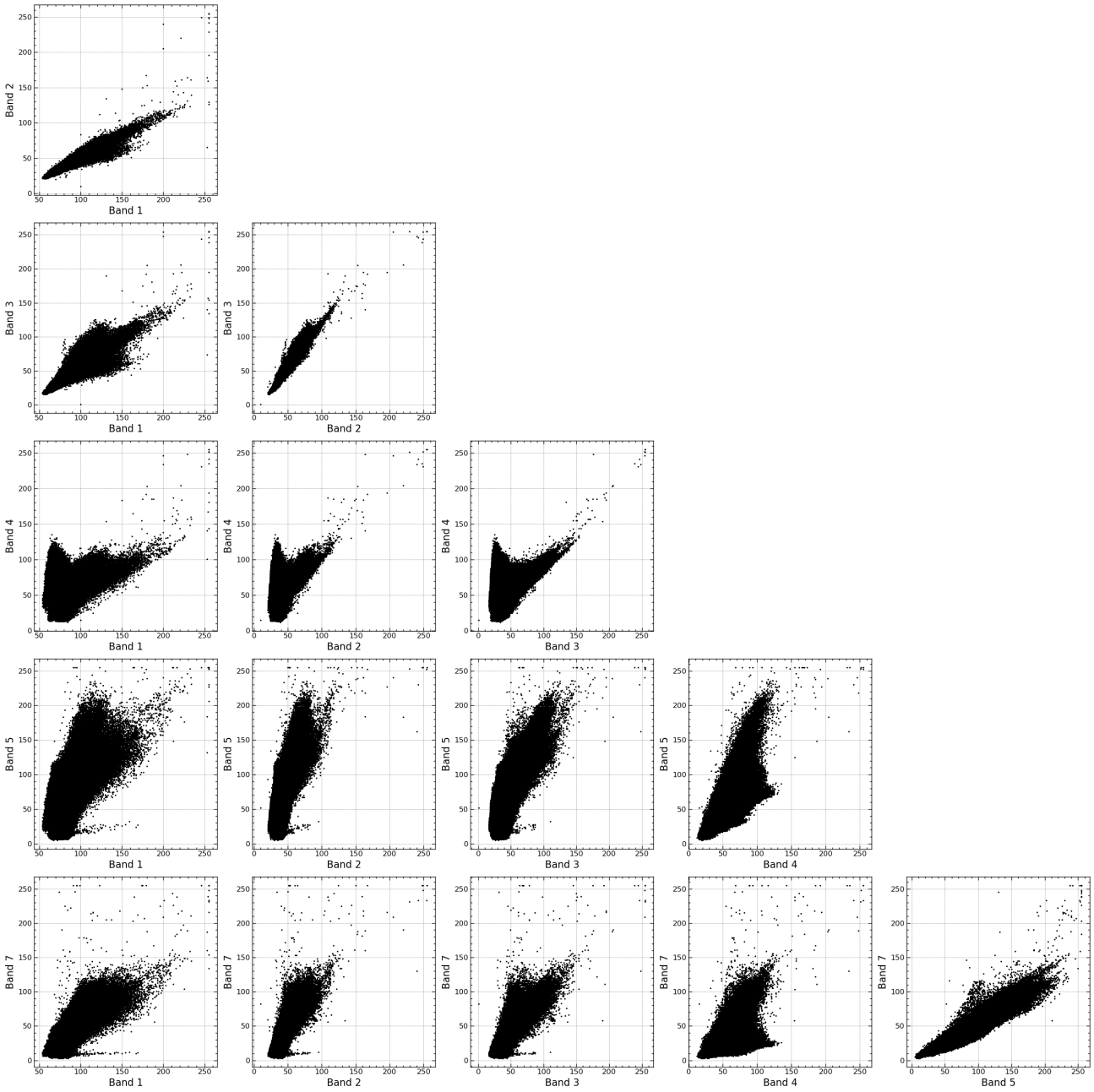

Figure 29: Histogram plots

components and the percentage of variance. From the figure 30, we can see that PC1 and PC2 are dominated over 90% of the variance of the data, so if we are about to do the dimension reduction, we can use only 2 components (PC1 and PC2) without sacrifice the resolution of the image.

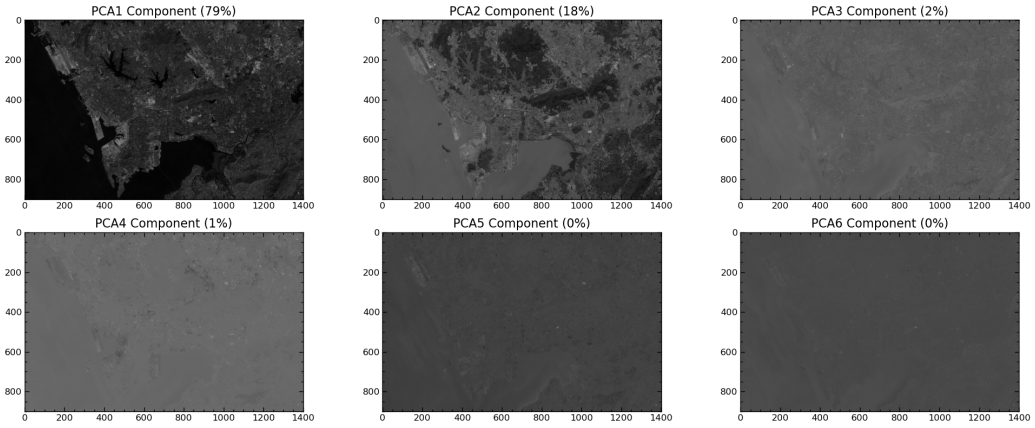

Figure 30: Percentage of Variance of Landsat

8 Clustering (K-means clustering)

Clustering is a technique in unsupervised learning that involves grouping similar data points together. One popular clustering algorithm is K-means clustering, which partitions a dataset into K clusters by minimizing the sum of distances between each point and its assigned cluster centroid.

Let $X = x_1, x_2, \ldots, x_n$ be a set of n observations, and K be the number of clusters we want to create. K-means clustering works as follows:

1. Initialize K cluster centroids randomly.
2. Assign each observation x_i to the cluster whose centroid is closest to it.
3. Recalculate the centroid of each cluster by taking the mean of all the observations assigned to it.
4. Repeat steps 2 and 3 until the centroids converge or a maximum number of iterations is reached.

To determine the optimal number of clusters K, we can use the elbow method. This involves computing the sum of squared distances between each point and its assigned centroid (also known as the Within-Cluster Sum of Squares, or WCSS) for different values of K, and plotting the WCSS against K. We then choose the value of K where the change in WCSS starts to level off, creating an elbow shape in the plot.

The objective function for K-means clustering can be written as:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{k} ||x_i - \mu_j||^2 \tag{25}$$

where n is the total number of data points, k is the number of clusters, x_i is the i-th data point, μ_j is the centroid of the j-th cluster, and $||.||^2$ is the squared Euclidean distance between two points.

One limitation of K-means clustering is that it assumes spherical clusters with equal variances. If our data has non-spherical or unevenly sized clusters, or if the data is not well-separated, K-means may not perform well.

8.1 K-means clustering for IMDB 500

In this study, we applied k-means clustering to a dataset of 5000 movies from IMDB, using six features including 'budget', 'popularity', 'revenue', 'runtime', 'vote_average', and 'vote_count'. The goal of this analysis is to explore patterns and trends within the movie dataset, and to identify any natural groupings or clusters based on similarities in these movie features. The results of this study can provide insights into the characteristics of successful movies, as well as guide future decision-making for movie production and marketing.

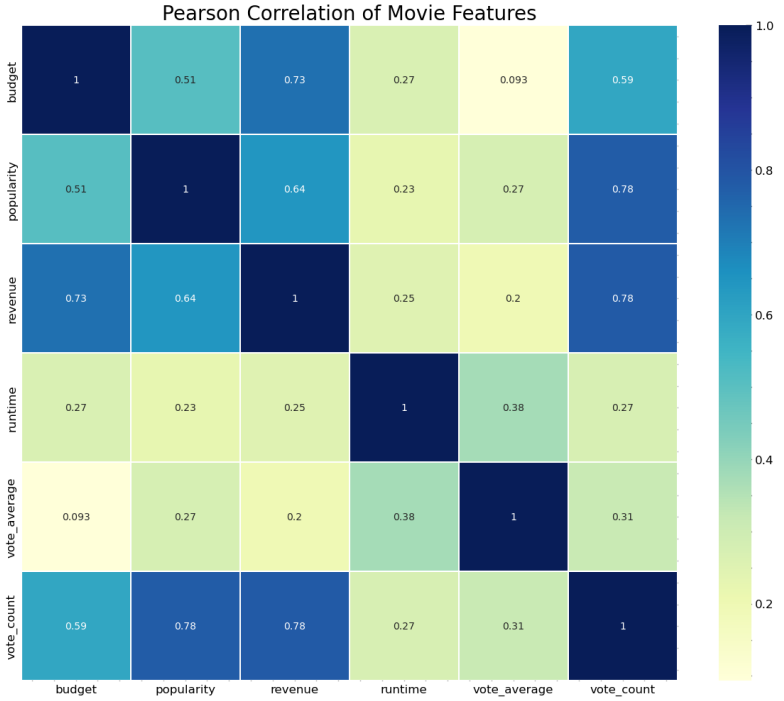

Figure 31: The correlation of 6 features of IMDB 5000 dataset.

In the figure 31, we can take a closer look of how different features relate to each other. For instance, it is obvious that features such as budget and revenue are highly positively correlated, suggesting that movies with higher budgets tend to generate higher revenues. On the other hand, features such as runtime and popularity do not have a strong correlation, suggesting that a movie's runtime may not have a significant impact on its popularity.

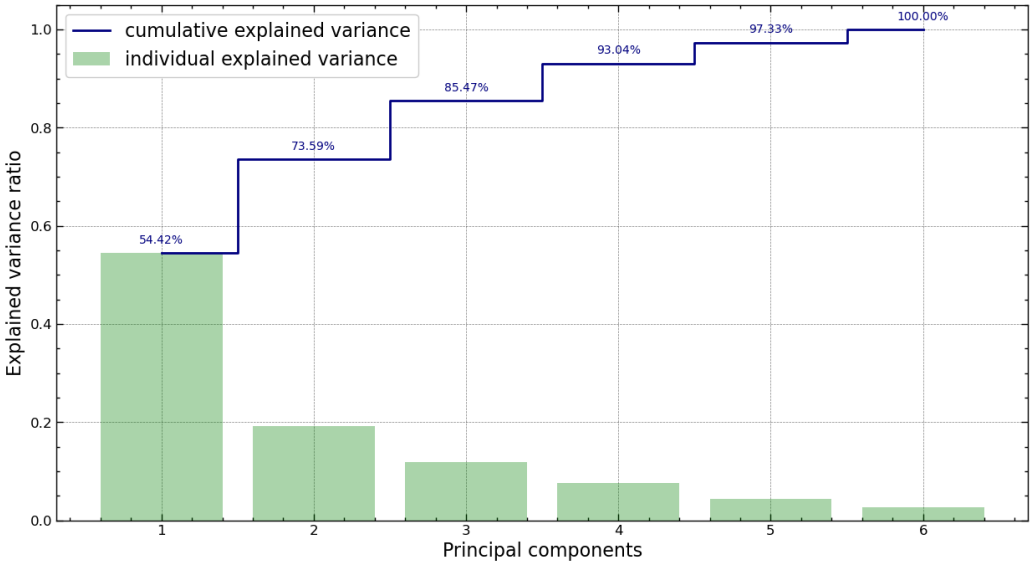

Figure 32: Scree Plot and Cumulative variance plot of IMDB 5000

After performing PCA on the movie features, based on figure 32, we found that the top 4 principal components contained 90% of the cumulative sum of variance explained. We then used the elbow method, shown in figure 33 to determine the optimal number of clusters for k-means clustering. From the resulting plot, we see that the elbow curve does not have any sharp turn or "elbow", where the addition of more clusters did not significantly improve the within-cluster sum of squares, that we need, so it is possible that the data can not be well clustered. However, based on the intuition, let's chose k = 4 and try to plot the k-means clustering for PC1 and PC2.

The results is shown in figure 34, the clustering is pretty arbitrary. Therefore, to assess how goodness of clustering, I use the silhouette score which is a metric used to evaluate the quality of clusters in a clustering analysis, such as k-means clustering. It measures how well-separated clusters are from each other, and how well the data points within each cluster belong to that cluster, compared to the neighboring clusters.

The silhouette score ranges from -1 to 1, with a higher score indicating better clustering. A score of 1 indicates that the clusters are well-separated, while a score of -1 indicates that the clustering is incorrect.

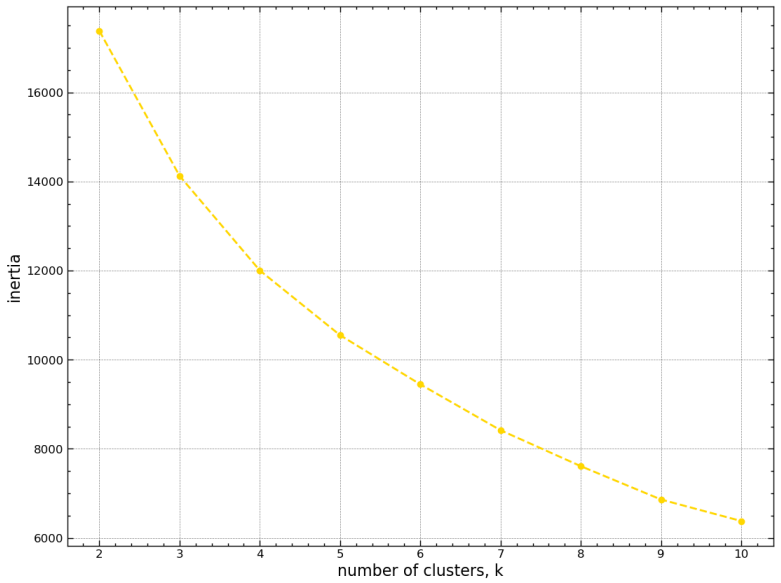

Figure 33: Elbow method on IMDB 5000

In Python, the sihoullute score can be calculated using the function `silhouette_score` in module `metrics` of the `sklearn` package.

The average silhouette score for 10 clusters is -0.002187, a negative value. It indicates that the clustering is not very good. A score of 0 means that the clusters are overlapping. It suggests that the clusters are not well-separated, because there is no clear structure in the data of IMDB 5000 that can be captured by clustering.

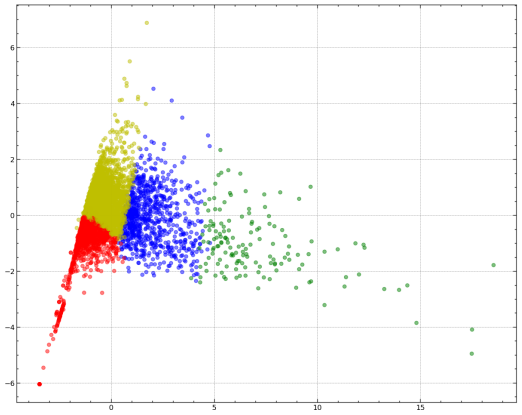

Figure 34: K-means clustering of PC1 and PC2 with k = 4

8.2 K-means clustering for Student's GPA

Similar to the previous study, in this section, we will use the K-means clustering to divide the students into groups based on their GPA on the subjects of 'Statistic', 'Algebra', 'Geometry', 'Geography', 'Biology', 'Physics', 'Chemistry', 'Earth Sciences', 'Literature', 'Writing ', 'Speech', 'Economics', 'Programming'.

From the correlation map, we can expect the student who is good at science subjects may struggle with subject like writing or literature, while literature and writing may also correlate with each other, as they both involve language and communication skills.

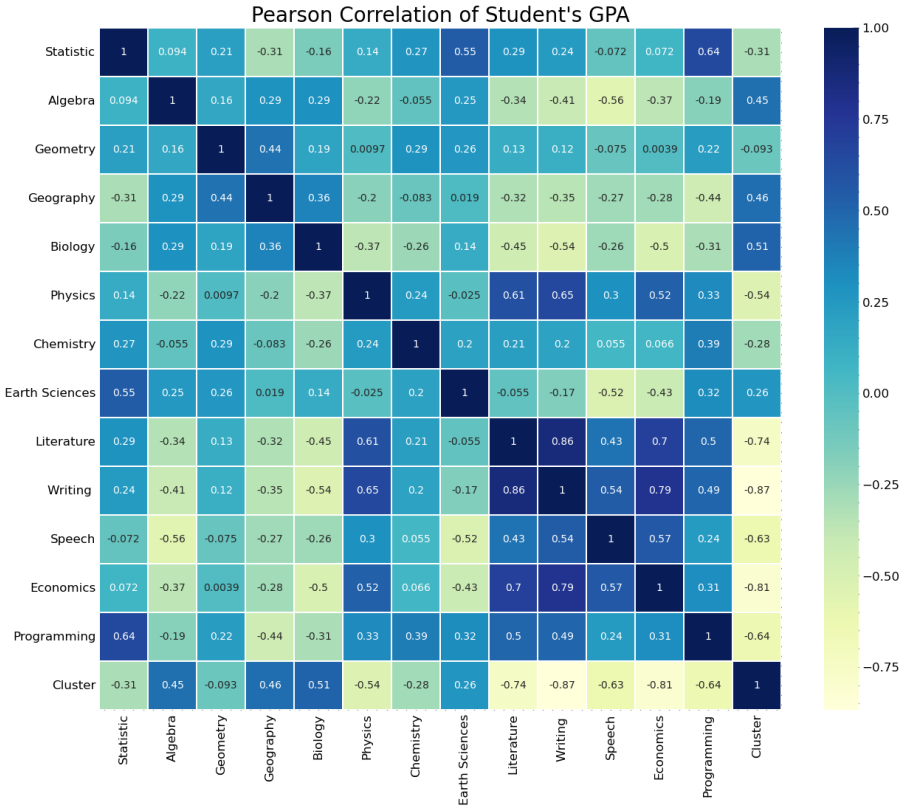

Figure 35: Correlation of subjects from student's GPA dataset

Going the same procedure with the previous section, we apply the PCA method and plot the scree plot and find out that 8 principal components make up over 90% of the percentage of variance. The result show in the figure 36.

We then used the elbow method to determine the optimal number of clusters for k-means clustering. From the resulting plot, we chose k = 3 as it appeared to be the "elbow" point, where the addition of more clusters did not significantly improve the within-cluster sum of squares. The result is shown in figure 37

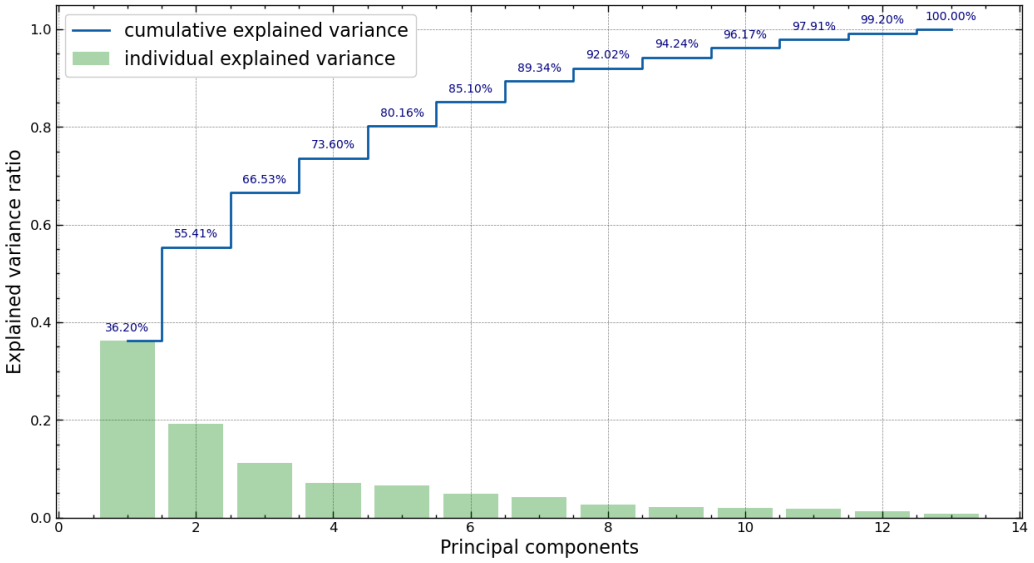

Figure 36: Scree Plot of Student's GPA.

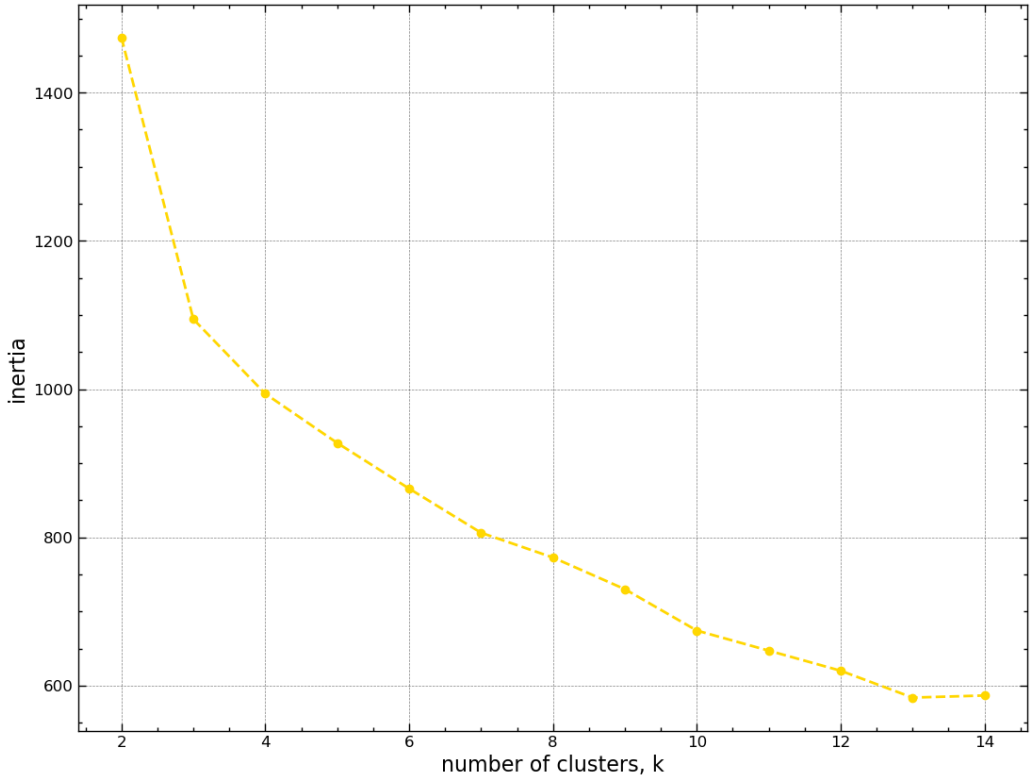

Figure 37: Elbow method for GPA

Now, we can visualize the K-means clustering for k = 3 on the PC1 and PC2 to intuitively assess the goodness of our clustering. From the figure 38, it suggests that the result of clustering for 2 first principal components are pretty good.

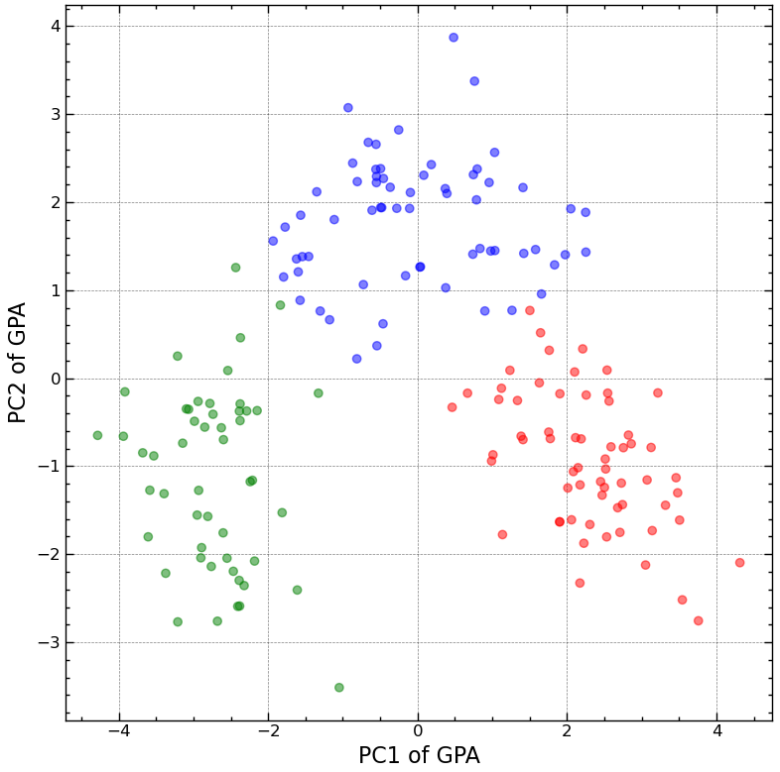

Figure 38: K-means clustering for PC1 and PC2 with k = 3

8.2.1 Grouping students

The most important thing is how can we divide the student into groups based on this results, to do that we will try to group the students in each clusters and calculate the mean values of GPA of each subjects, and group the students based on their performances on certain subjects.

```
         Statistic    Algebra  Geometry  Geography   Biology   Physics  \
Cluster
0        13.676774   7.991613  9.865161  17.462903  2.920968  1.922097
1        12.250923   7.589538  8.924923  20.063077  3.576923  1.624154
2        13.134118  13.229020  9.670588  21.241176  4.519608  1.145882

          Chemistry  Earth Sciences  Literature  Writing     Speech  \
Cluster
0        107.967742        5.453548   14.237903  3.003226  1.065484
1         92.738462        2.973077   11.238462  2.050000  1.062708
2         98.666667        7.234706    8.419608  0.818824  0.691961

         Economics  Programming
Cluster
0         3.163387  1100.225806
1         2.803385   510.169231
2         1.696667   619.058824
```

From this table, we can draw the conclusion that:

- Cluster 0: This cluster has the highest values for Statistics, Geography, and Literature, suggesting that the students in this group may have stronger skills in these subjects compared to the other clusters. They also have a higher average GPA in Programming.
- Cluster 1: This cluster has the lowest values for most subjects, including Statistics, Algebra, Geometry, Biology, and Physics, suggesting that the students in this group may struggle in these areas. They have the lowest average GPA in Programming.
- Cluster 2: This cluster has the highest values for most subjects, including Algebra, Biology, Physics, Chemistry, Earth Sciences, and Economics, suggesting that the students in this group may excel in these areas. However, they have the lowest average GPA in Writing and Speech.

Therefore, I suggest that we can divide the student in three groups:

- Group 1: Students with a strong background in Statistics, Literature and skill of Programming. Those students may consider go to Economics with their performance.
- Group 2: Low-achieving students, whose GPA are very low, especially on Science subjects.

Those students need more support to improve their performance.

- Group 3: Well-performance students with a strong background in Mathematics and Science, but have the lowest average GPA in Writing and Speech. Those students may consider go to Science and Technology career in the future.

9 Conclusions

After going this far with a lot of mathematics and knowledge, I hope that this report can make clear on my study on the course of Data Analysis. Personally, I am grateful that from this course I learn a lot about Monte Carlo method which also used a lot in astrophysics and also a little bit on Machine Learning by PCA and K-means clustering method. In addition, I have to note that not all the code are written completely by me, some of the code is provided by Dr. Nguyen Le Dung, particularly, the code on PCA and K-means clustering, those I did some improvement to make the plot of those method more intuitive. Finally, I think it is crucial to not just learn how to code but also learn how to think, to apply the knowledge that we learn to shed the light on the mystery and solve the practical problems of this world.

$$NDWI(soil) = \frac{NIR - SWIR}{NIR + SWIR} \tag{26}$$

where b = 2.89777×10^{-3} m.K