Advanced Methods For Visualization And Interpretation Data With Python

Luis Eduardo Muñoz Guerrero

Facultad de Ingenierías, Universidad Tecnológica de Pereira Correo electrónico: lemunozg@utp.edu.co ORCID: https://orcid.org/0000-0002-9414-6187

Abstract

The article explores the essential steps and key techniques of exploratory data analysis, starting with the preparation and structuring of information, followed by the calculation of fundamental statistical metrics. Using the Iris dataset as an example, correlations are analyzed and various visualization techniques are employed, from basic graphs such as histograms and scatter plots, to advanced visualizations, including combined and 3D graphs. In addition, the concept of clustering is introduced as a crucial tool to detect hidden patterns in data. Throughout the process, practical recommendations are presented for selecting and applying the most appropriate methodologies, thus allowing for more accurate and useful interpretations. This approach seeks not only to facilitate the understanding of the data, but also to maximize its analytical value through the effective use of statistical and visual tools.

Keywords— Data analysis, Advanced graphics, Interpretation, Visualization techniques, Statistical analysis.

I. INTRODUCTION

The first step in any data science project is exploratory analysis of information. Data can come from many sources, such as sensors, videos, images, surveys, or forms. The Internet is one of the most important providers of data, which enters daily in enormous quantities. However, this does not mean that they are useful. A large part of this information is not structured, and requires interpretation for it to be useful. In fact, one of the great challenges of data science is to turn a torrent of data into useful information.

Although it may seem a new concept, data analysis or "Data Analytics" was first proposed in 1962 by John W. Tukey, who proposed a statistical reform that included inference as a fundamental pillar of statistics. Today, many concepts that were part of their research project, such as box or scatter plots, are still part of the daily work of many scientists. And because we have faster and more complex computational processes, it is even possible to perform analyses far beyond what had been their original scope.

Next, we will review the most important methods in an exploratory analysis and some useful tips for interpreting the results.

II. DATA USED FOR ANALYSIS

In this article, we will use the "Iris" dataset to perform the exploratory analysis of the data. This dataset was introduced by biologist and renowned statistician Ronald Fisher in 1936 in his paper "The use of multiple measurements in taxonomic problems".

The database considers four characteristics of three species of Iris flower: length and width of the petals and sepals. In addition, it has 50 samples from each category: Iris Setosa, Iris virginica and Iris versicolor. In Fig. 1, we can see the morphological differences between them.

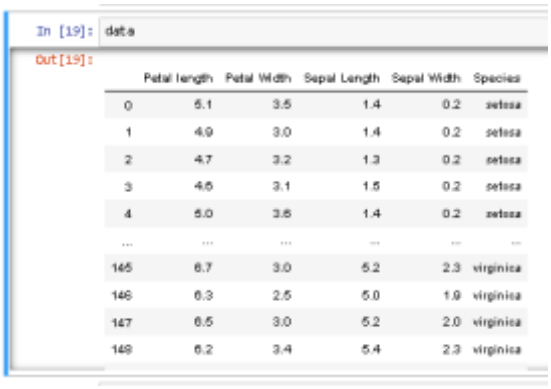
Fig. 1 Características de las flores de subespecies de Iris. Scikit-learn, the Iris Dataset, and Machine Learning: The Journey to a New Skill, Medium, 2021. [https://3tw.medium.com/scikit-learn-the-iris-dataset-and-machine-learning-the-journey-to-a-new-skill-c]

Before starting with the exploratory analysis, we will start with the import of the necessary libraries (numpy, pandas and sklearn), as well as the dataset. In addition, we will adjust the data to adapt it to the Data Frame format, which is a two-dimensional structure that facilitates the manipulation of its values.

```
import numpy as np
import pandas as pd
from sklearn.datasets import load_iris

dataset=load_iris()
data=pd.          DataFrame(dataset['data'],columns=['Petal
length','Petal Width','Sepal Length','Sepal Width'])
data['Species']=dataset['target']
data['Species']=data['Species'].apply(lambda x:
dataset['target_names'][x])
```

By executing the data variable, we can see (in Fig. 2) the structure of the data that we will analyze.



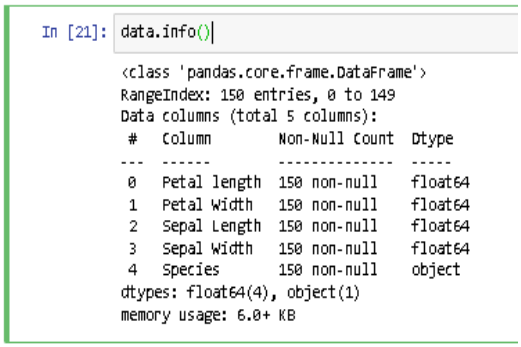
In [19]: data

Out[19]:

	Petal length	Petal Width	Sepal Length	Sepal Width	Species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
149	6.2	3.4	5.4	2.3	virginica

Fig. 2 Structure extracted from the Iris dataset. Own source, 2025.

As can be seen in Fig. 3, it is also convenient to use the info() function to determine the type of variable that corresponds to each piece of data.



```
In [21]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Petal length  150 non-null    float64
 1   Petal Width   150 non-null    float64
 2   Sepal Length  150 non-null    float64
 3   Sepal Width   150 non-null    float64
 4   Species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

Fig. 3 Determination of variable types in the Iris dataset. Own source, 2025.

Here we can see that all the values are numerical, except for the Species, which is categorical. It is recommended that, in any exploratory analysis, this type of control be carried out before starting to ensure that each data is correctly classified.

.

III. METRICS IN EXPLORATORY ANALYSIS

While it may seem like an obvious obvious fact, to know the data, we need to know what they are. When dealing with a considerable volume of records, we must have "typical values" that help us estimate how the database under study behaves. For example, what is its central tendency.

A. Stocking

It is an elementary estimate that estimates the average or average value of a data set. In a nutshell, it is the sum of all values divided by the number of values in a population (μ) or sample ($\bar{x}$).

Being a metric sensitive to extreme values, many analysts and scientists introduce the value of "truncated average". It is calculated using the same method, with the difference that outliers are previously eliminated.

For example, in international diving, the maximum and minimum scores of five judges are eliminated, and the final score is the average of the scores of the remaining three judges. In this way, it is more difficult for a judge to manipulate the score, perhaps to favor a contestant from his own country.

B. Median

It is the central value in a list of data ordered from lowest to highest. Unlike the average, it does not use all the data, but depends only on the values located in the center. While it may not seem like a representative metric, you can have applications where it's even more important than average.

Suppose we want to analyze typical household incomes in Colombia. When comparing Medellín with Bogotá, for example, the use of the average would give us very different results because the wealthiest families can be found in the capital. In the case of the median, these outliers will not matter, as the position of the intermediate observation remains the same.

C. Fashion

It is the value that appears most frequently in the data. In statistics, this value is a simple summary for categorical data and is generally not used for numerical values.

D. Variability

This is the second dimension in the localization of data. It is also known as dispersion, a fundamental characteristic to know how values deviate from the mean. One way to measure variability is to estimate the typical value of these deviations as absolute values.

The two ways to measure variability are variance and standard deviation, which are calculated from the square of the deviations. The variance, then, is an average of the square of the deviations, and the standard deviation is the square root of the variance.

E. Quartiles

These are those values that divide a sample into four equal parts. In many analyses, they can be used to assess the dispersion and trend of data. Quartiles are classified into three parts: Q1 (separating the bottom 25% from the top 75%), Q2 (separating the bottom 50% from the top 50%, which is why it's also called the Median), and Q3 (separating the bottom 75% from the top 25%)

Returning to the Iris dataset, we can obtain these metrics through the describe() function, the output of which can be seen in Fig. 4.

```
data.describe()
```

```
In [20]: data.describe()
Out[20]:
```

	Petal length	Petal Width	Sepal Length	Sepal Width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

Fig. 4 Observing metrics using the describe() function. Own source, 2025.

In the output of Fig. 4, we can see the count of all the variables, their mean, standard deviation, minimum and maximum value, as well as the values found in the quartiles 25%, 50% (median) and 75%. If we seek to observe this data graphically, it is possible to use a box graph. In this case, we apply the Species criterion to classify these metrics:

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.boxplot(x="Species", y="Petal length", data=data )
plt.show()
```

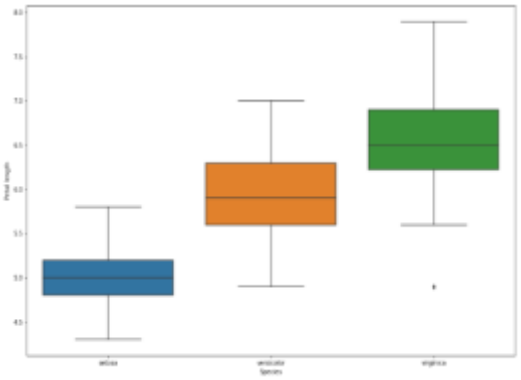


Fig. 5 Box charts. Own source, 2025.

The box graph helps us identify how the values of each variable are distributed. In this case, without carrying out an exhaustive analysis, we can observe that the length of the petal varies significantly between species. In Fig. 6, you can see in detail the structure of a diagram in its original conception.

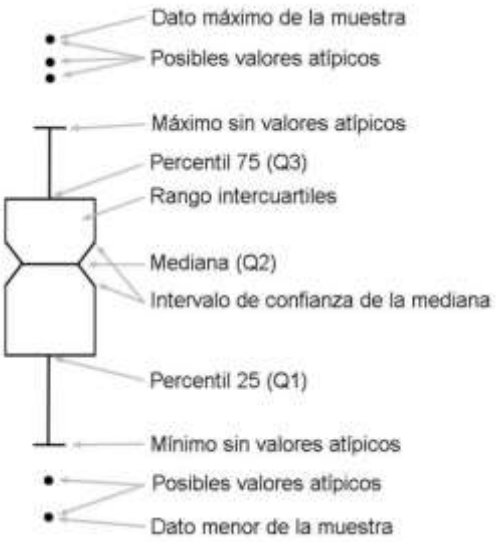


Fig. 6 Basic composition of a diagram of boxes and whiskers in their original conception, A. Rodríguez, E. Cuevas, D. Zaldivar, & L. Castañeda, 2019. [https://doi.org/10.13140/RG.2.2.34348.28806/1]. In the original Spanish language.

Returning to the Iris dataset, we can already determine the observations expressed in Table 1.

BOARD I INTERPRETATION OF THE BOX GRAPH OF THE IRIS DATASET.

Source: Own, 2025.

Feature	Setosa	Versicolor	Virginica
Median	5.0 cm	5.9 cm	6.5 cm
Interquartile Range	Narrow. It indicates low variability.	Moderate. Greater variability than setosa	Broad. High biodiversity
Total Range (Whiskers)	~4.5 cm to ~5.5 cm	~4.8 cm to ~7.0 cm	~4.8 cm to ~7.8 cm
Outliers	None	None	An outlier. This indicates a specimen with an exceptionally long petal compared to the rest.
Separation between subspecies	It is totally separate from the other two species	It partially overlaps with Virginia.	Partially overlaps with versicolo
Homogeneity	Loud. The petals are	Moderate	Casualty. The petals

	very uniform.		show high variation.

Only by using these metrics, we can draw interesting biological conclusions regarding the three subspecies of Iris. First, we can identify and classify specimens according to their characteristics, such as petal length. On the other hand, we can detect sources of variation that can be environmental or genetic within each population. Versicolor and virginica could probably be subspecies that share similar ecological niches or are adapted to common environments.

IV. INTERPRETATION OF DISTRIBUTIONS AND FREQUENCIES

In statistics, distributions are a fundamental pillar in inference. This is because they reflect how the measured values are dispersed in our database. In data analysis, we can make use of many graphs to study its appearance and behavior: is it symmetrical or asymmetrical? Do you have typical or widely dispersed values? How often do some observations occur?

A characteristic observed in most (well-obtained) samples and populations in general, is that they tend to continue. a Normal curve.

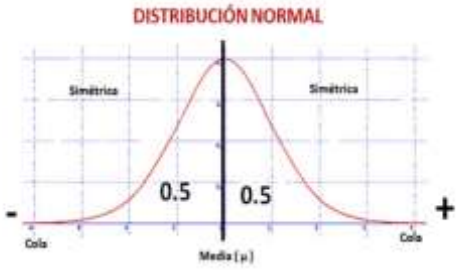


Fig. 7 Characteristics of a Normal Distribution, Matepedia, 2016. [https://matepedia-estadistica.blogspot.com/2016/09/caracteristicas-de-una-distribucion.htm]. In the original Spanish language.

The normal bell-shaped curve is iconic in statistics. As we can see, it is symmetrical on both sides of the center, where the largest number of observations are grouped. On the other hand, the most extreme values or those far from the center are less frequent.

Likewise, the Normal depends on two important values: the mean (in this case, it coincides with the mode) and the standard deviation. The former forms the center of the graph, while the latter configures the "queue length" and frequency around the central value. Approximately, it can be said that:

- 68% of the data is located within 1 standard deviation ($\mu \pm \sigma$),
- 95% of the data is within 2 standard deviations ($\mu \pm 2\sigma$).
- 99.7% of the data is within 3 standard deviations ($\mu \pm 3\sigma$).

Returning to the Iris dataset, we will perform the following exploratory analysis of all the variables:

```
# Set Chart Style
sns.set(style="whitegrid")

# Create subplots for each variable
fig, axes = plt.subplots(2, 2, figsize=(12, 8))
variables = ['Petal length', 'Petal Width', 'Sepal Length', 'Sepal Width']

for i, var in enumerate(variables):
ax = axes[i // 2, i % 2]
sns.histplot(data=data, x=var, hue='Species', kde=True, ax=ax, palette="Set2")
ax.set_title(f'Distribución de {var}')
ax.set_xlabel(var)
ax.set_ylabel('Frequency')

# Adjust Spacing
plt.tight_layout()
plt.show()
```

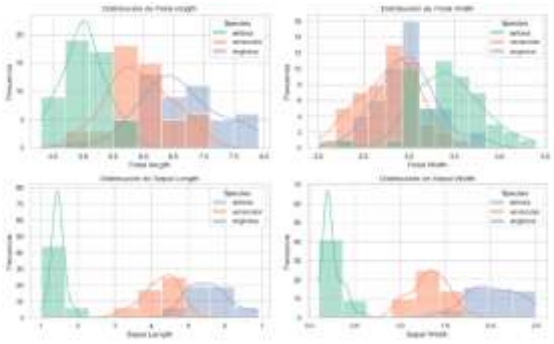


Fig. 8 Distributions and densities of the Iris dataset. Own source, 2025. In the original Spanish language.

In the output of the code expressed in Fig. 8 we will find four graphs, which demonstrate the distribution of width and length of the petal and sepal of the three subspecies. The frequencies will be shown as higher rectangles, which will be accompanied by density lines that represent a continuous approximation to the distribution of the data.

In the petal length, we can again detect a pattern similar to the one we had determined with the box chart. A similar behavior is found with the length of the sepal, where the differences in setose are even more noticeable. However, in the distribution of the petal's width, there is more overlap between Iris setosa and versicolor. This does not happen with the width of the sepal, where there are more differences.

However, it is not yet possible to ensure that these data continue with the same normal curve pattern. To determine if the data have a normal distribution, we can use a QQ or QQ-Plot diagram. It orders the Z-score (values transformed to a standard normal) from lowest to highest, recording its values on the Y-axis. The X-axis corresponds to the quantile corresponding to a normal distribution.

```
import scipy.stats as stats
```

```
# Choose the column for the probability graph (e.g. 'Petal length')
column_to_plot = 'Petal length'

# Create the probability plot (Q-Q plot)
fig, ax = plt.subplots(figsize=(10, 10))
stats.probplot(data[column_to_plot], dist="norm", plot=ax)

# Show the graph
plt.title(f'Q-Q Plot para {column_to_plot}')
plt.show()
```

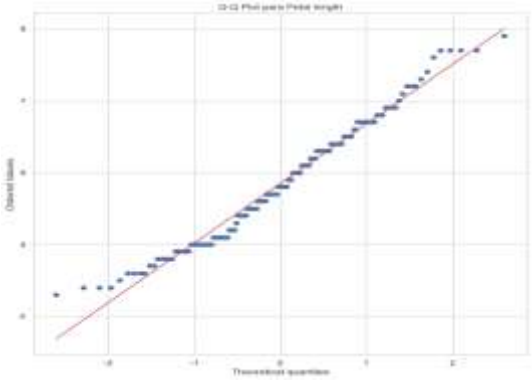


Fig. 9 Q-Q Plot for Petal Length. Own source, 2025.

In this case, we only use the length of the petal, although we can apply it to any variable of interest. As expected, the points follow closely to the center line. However, it is also possible to find values that do not follow the standard.

If the dots move away in the queues, as they do in the analyzed graph, it could suggest that the data has thicker tails than a normal distribution would expect. Logically, a database like Iris consists of a small sample, so the database is small and probably follows a Student T distribution.

The Student's T-distribution is actually a normal-shaped distribution, except that it is a bit thicker and longer in its tails. It is used to represent sample distributions. The larger the sample size, the distribution will tend to look more like a Normal and the Q-Q plot dots will be more faithful to the center line.

V. ANOVA OR ANALYSIS OF VARIANCE

The ANOVA test, or better known as analysis of variance, is a method widely used in statistics to know if the results of a test are significant. In research, they allow a hypothesis to be rejected or accepted. It is important to consider that, in order to perform an ANOVA, the data must be distributed normally and must have a homogeneous variance between groups.

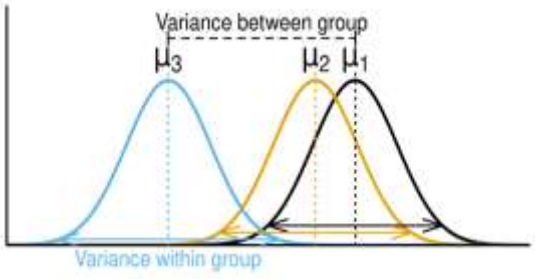


Fig. 10 Relationship between variances within and between groups, AnotherOrion, 2024. [https://anotherorion.com/pengertian-anova-dalam-penelitian/]

In the following code snippet, we can apply an ANOVA test between the three Iris subspecies, where an analysis of each variable will be performed:

```
import pandas as pd
from scipy import stats
from sklearn.datasets import load_iris

# Load the Iris dataset
iris = load_iris()

# Create a DataFrame with the data
df = pd. DataFrame(data=iris.data, columns=iris.feature_names)

# Add the species column (target)
df['species'] = iris.target

# Map the numerical values of the species to their names
df['species'] = df['species'].map({0: 'setosa', 1: 'versicolor', 2: 'virginica'})

# Display the first rows of the DataFrame
print(df.head())

# Perform the ANOVA for each numerical variable with respect to the species
for column in df.columns[:-1]:  # Excluir la columna 'species'
print(f"ANOVA para la variable '{column}':")
f_value, p_value = stats.f_oneway(df[df['species'] == 'setosa'][column],
df[df['species'] == 'versicolor'][column],
df[df['species'] == 'virginica'][column])

print(f"  - F-valor: {f_value}")
print(f"  - P-valor: {p_value}")

if p_value < 0.05:
print(f" There are significant differences between species in the variable '{column}'.\n")
else:
print(f" There are no significant differences between species in the variable '{column}'.\n")
```

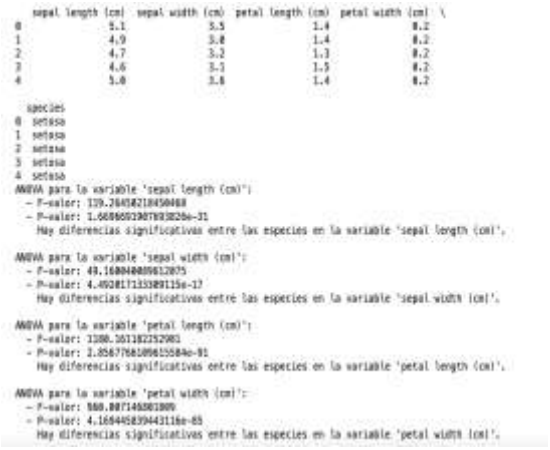


Fig. 11 ANOVA performed on Iris dataset. Own source, 2025.

As can be seen in Fig. 11, the p-value is small, which indicates that there are significant differences between variables. This suggests that Iris species can be clearly differentiated from each other based on measurable characteristics (length and width of sepals and petals).

VI. RELATIONSHIP BETWEEN VARIABLES

In many data analysis and data science projects, the need to explore the relationship between variables is involved. We have already seen that petal length and width have differences between species, but they have an unclear pattern between the two characteristics. If we must know more about these subspecies, we should carry out an analysis about them.

The variables X and Y (each with recorded data) are said to be positively correlated if high values of X accompany high values of Y, and low values of X accompany low values of Y. If the opposite were to occur, that is, that X accompanies low values and Y accompanies high values, this correlation would be negative.

For example, below we will observe two variables, which are perfectly correlated in the sense that each one goes from highest to lowest:

v1: { 1, 2, 3}
v2: { 4, 5, 6}

But how do we objectively measure exactly what the degree of correlation between the two variables is?

Fortunately, we have standardizations such as the correlation coefficient (r). It is an estimate of the correlation between two variables that always have the same scale.

This coefficient can be visualized as a number that can be negative or positive, with a range of -1 to 0 and 0 to 1. Zero indicates that there is no correlation between the variables, while the closer to 1, the greater the strength of the relationship between X and Y. If we look at different examples through a scatter plot, we can observe a behavior that can be graphed in Fig. 12.

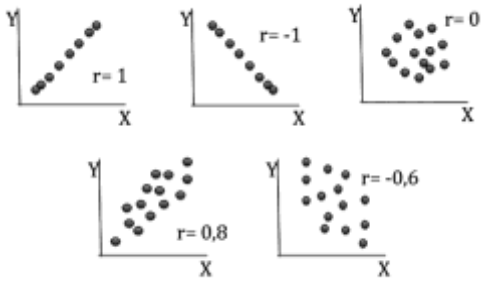


Fig. 12 Correlation, Chilean Journal of Anesthesia, 2014. [https://doi.org/10.25237/revchilanestv43n02.15]

We must also take into account that many variables can have a non-linear association, so this coefficient would not be the most appropriate metric.

In general, correlations are displayed in a table to visually show that relationship. Next, we'll look at how to apply it to our Iris dataset:

```
import matplotlib.pyplot as plt
import seaborn as sns

# Correlation Matrix
correlation_matrix = data.iloc[:, :-1].corr() # We exclude the "Species" column as it is not numeric
print(correlation_matrix)

# Visualization of the correlation matrix
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title('Correlation Matrix')
plt.show()
```

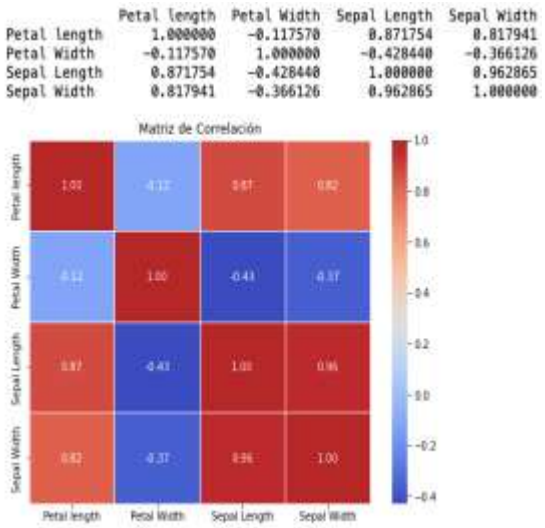


Fig. 13 Correlation analysis in the Iris dataset. Own source, 2025.

Now, we will analyze the output of the code graphed in Fig. 13. To begin with, we observed a matrix, which was

stored in the variable correlation_matrix. Quickly, we can see that all the variables have a correlation with each other, although it can take positive and negative values. This relationship can best be visualized in the graph, where it is represented with a color scale.

Emphasizing the data, we can observe a strong correlation between the length and width of the sepal (0.96). In other words, the sepal will be wider as its length increases and vice versa. On the contrary, this relationship is less evident in the petal, which has a correlation coefficient of -0.12. However, the width of the sepal and petal have a weak negative correlation (-0.37).

However, in this case we are analyzing the correlation between variables without considering the category of that source of information. Subspecies could have distinctive correlations between the characteristics of their flowers, something that we could not assess in this way. Next, a correlation matrix will be made using the subspecies as categorical.

```
# Filter data by species and calculate correlation matrices separately
species = data['Species'].unique()

for specie in species:
subset = data[data['Species'] == specie].iloc[:, :-1] # Excluir columna 'Species'
correlation_matrix = subset.corr()

# Visualization of the correlation matrix for the current species
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title(f'Correlation Matrix - {specie}')
plt.show()
```

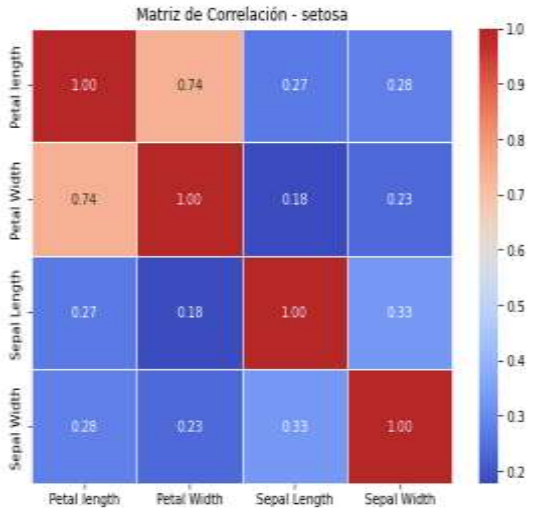


Fig. 14 Correlation analysis categorized by subspecies in the Iris dataset. Own source, 2025. In the original Spanish language.

In general, in Fig. 15 we can observe weaker correlations between the variables, although they are still significant. For example, Iris setosa appears to have a sparse data pattern.

In comparison, we can note that correlation without considering the subspecies tends to introduce additional variability and generates more apparent relationships that "amplify" trends. For example, if one species has longer petals and another has shorter petals, by combining that information, the length can be more strongly correlated with other variables, since there is a wider range of values available. In the second analysis, the characteristics may not be as closely related to each other due to differences in morphology or evolutionary adaptations of each variety.

At this point, many analysts would wonder what would be the most efficient way to study the correlations of the dataset. The answer is relative, since it will depend on the object under analysis.

A separate correlation by subspecies could give them a more accurate notion of the specific relationships. In a comparative study, it could be a convenient option. On the contrary, if the analyst seeks to obtain general information about the genus Iris, and is not focused on looking for differences between subspecies, the general correlation would provide more useful information for his research.

Although so far we have focused on identifying relevant patterns in data, they can also be useful for designing strategies or making decisions. In more complex analyses, these correlations could be relevant to make predictions using Machine Learning techniques, or to reduce the dimensionality of the data, eliminating variables that provide redundant information.

VII. DATA VISUALIZATION TECHNIQUES

Data analysts must not only turn raw information into useful information, but they also have a duty to turn them into representations that make it easier to understand and analyze. The data, on its own, can be overwhelming, and in many cases, it is difficult to interpret for people who are not involved in the research project.

Therefore, visualizations can help the analyst clarify the object of study, as well as demonstrate patterns and trends that they have discovered throughout their work. In the corporate sphere, visualizations are tools that can be used by different departments to make informed decisions or to communicate results that could change the course of the corporation.

Next, we will analyze some graphs and how to apply them, using the Iris dataset.

Basic Charts

So far, we've already explored some simple graphs like the boxplot and some histograms to analyze distributions. However, for a more advanced analysis, we will introduce ourselves to other visualizations.

A.1. Scatterplots

Scatter charts are useful visual tools if we are looking to determine the relationships between two numerical variables. Therefore, each point on the graph represents two values: one on the X-axis (first variable) and on the Y-axis (second variable). It has similarities with correlation coefficients and, in fact, they are usually used for the same analyses.

However, when the relationship between variables is not linear, it would be convenient to check it beforehand through scatter plots. It is also useful for determining if outliers exist and is so simple to understand that it is often the first step in making assumptions in research.

```
# Create the scatter chart
plt.figure(figsize=(8, 6))
for species in data['Species'].unique():
subset = data[data['Species'] == species]
plt.scatter(subset['Petal length'], subset['Petal Width'], label=species, s=60, alpha=0.7)
# Customize the chart
plt.title('Scatter Plot: Petal Length vs Petal Width')
plt.xlabel('Petal Length (cm)')
plt.ylabel('Petal Width (cm)')
plt.legend(title='Species')
plt.grid(True)
# Show the graph
plt.show()
```

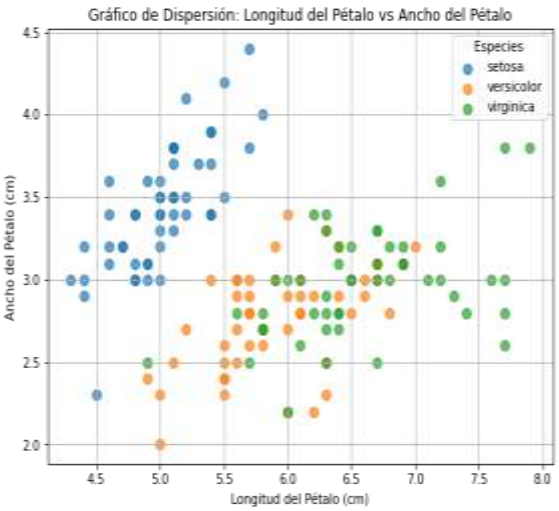


Fig. 15 Scatter Plot: Petal Length vs Petal Width. Own source, 2025. In the original Spanish language.

In Fig. 15 we can easily and intuitively observe how the data pairs are distributed according to the subspecies. Quickly, we notice that the length and width of the petal is differentiated in setosa, while in versicolor and virginica there is some overlap. Returning to the photographs in Fig. 1, we can easily notice these characteristics.
If we want to get all the relationships between variables, we can implement the following code for faster visualization:

```
# Create a DataFrame with the variables and species
iris_data = data[['Petal length', 'Petal Width', 'Sepal Length', 'Sepal Width', 'Species']]
# Create the scatter chart matrix
sns.pairplot(iris_data, hue='Species', markers=["o", "s", "D"], palette='Set2')
# Customize the chart
plt.suptitle('Scatter Plot Matrix - Iris', y=1.02)
# Show the graph
plt.show()
```

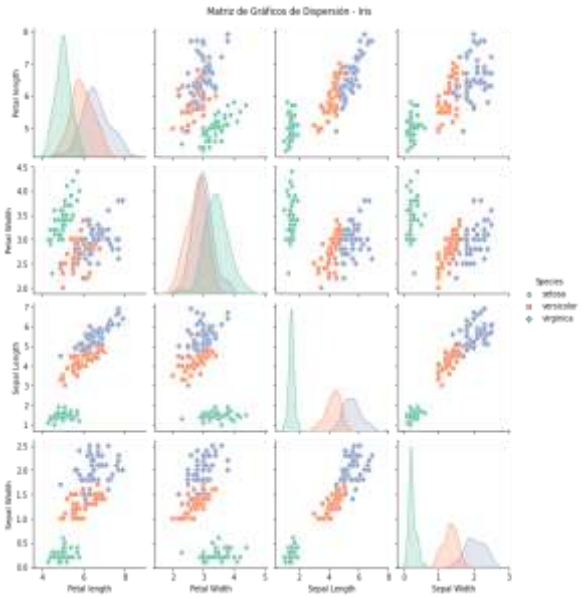


Fig. 16 Array of Scatter Plots in Iris dataset. Own source, 2025.

A.2. Bar diagams

Graphs or bar charts are widely used in data analysis, due to their ease of obtaining and interpreting them. If we are looking to compare quantities or frequencies between categories, it can be a convenient option. It is also recommended when the data is discrete and not continuous.

To obtain an easy-to-understand diagram, we will make some adjustments to the data, as well as to the design of the diagram:

```
# Create a DataFrame with the variables and species
iris_data = data[['Petal length', 'Petal Width', 'Sepal Length', 'Sepal Width', 'Species']]
# Convert the DataFrame to Long Form for use with Seaborn
iris_data_long = pd.melt(iris_data, id_vars='Species', var_name='Variable', value_name='Valor')
# Create the bar chart
plt.figure(figsize=(12, 8))
sns.barplot(x='Variable', y='Valor', hue='Species', data=iris_data_long, ci=None)
# Customize the chart
plt.title('Comparison of Variables by Species in the Iris Dataset', fontsize=16)
plt.xlabel('Variable', fontsize=14)
plt.ylabel('Valor', fontsize=14)
plt.xticks(rotation=45)
plt.grid(True, axis='y', linestyle='--', alpha=0.7)
# Show the graph
plt.show()
```

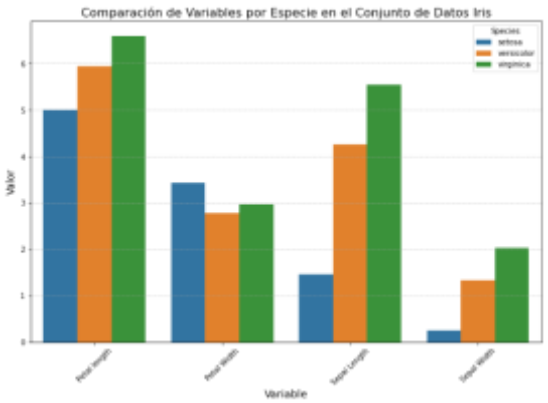


Fig. 17 Comparison of variables by species in Iris dataset using bar graphs. Own source, 2025. In the original Spanish language.

In the comparison shown in Fig. 17, it can be seen that the width and length of the petal do not show such marked differences between the subspecies, unlike the sepal, which does show more significant variations. In addition, we can notice that Iris setosa has the widest petal, at the same time that it has the narrowest sepal.

If, for example, a group of researchers wishes to present this analysis at a botany conference, it could be a useful representation to communicate their observations to the interested public.

A.3. Area Chart

If the analyst's interest is to determine the magnitude of the values, the area chart can be an optimal tool to achieve this objective. In some cases, it is useful when looking to demonstrate a cumulative relationship between variables or how they vary along an axis, such as time or the number of samples.

In this case, we will use the measurement of each variable using the subspecies as a category on the X-axis. To achieve this visualization, it will be necessary to obtain the average of the variables in each subspecies:

```
    # Calculate averages by species
    species_mean = data.groupby('Species').mean()
    # Create the area chart
    plt.figure(figsize=(10, 6))
    species_mean.plot.area(alpha=0.6, cmap='Blues',
figsize=(12, 8))
    # Customize the chart
    plt.title('Area Graph of Average Variables by Species
in the Iris Dataset', fontsize=16)
    plt.xlabel('Variables', fontsize=14)
    plt.ylabel('Promedio (cm)', fontsize=14)
    plt.legend(title='Species', loc='best')
    plt.grid(True)
    # Show the graph
    plt.show()
```

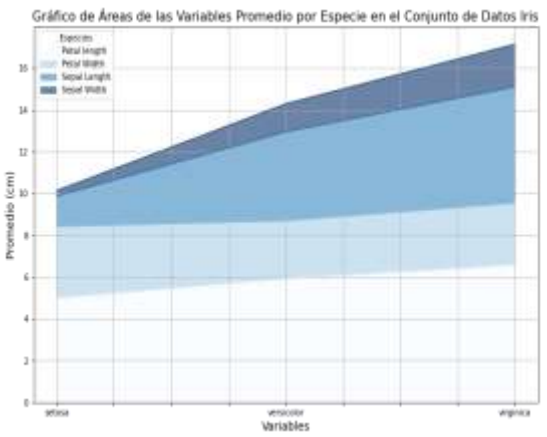


Fig. 18 Area graphs of the average variables by species in the Iris dataset. Own source, 2025. In the original Spanish language.

A. Advanced Graphics

In data analysis, a more thorough study of the information may be necessary, so we need to use more complex graphs to achieve useful visualizations. The main difference with simple charts is that advanced charts can contemplate multidimensionality and work with more complex data that could not be worked with histograms or boxplots.

B.1 Radial Charts

Also known as "spider charts", these are representations that are used to compare quantitative variables between different categories, which are represented on each radial axis. For the following analysis, the dataset was filtered to discriminate the average characteristics of Iris setosa.

```
    # We choose a species to display on the radial chart
(e.g. 'setosa')
    species_data = data[data['Species'] ==
'setosa'].drop('Species', axis=1).mean()

    # Variable Tags
    categories = species_data.index.tolist()

    # Values for the radial chart
    values = species_data.values.tolist()

    # Number of variables
    num_vars = len(categories)

    # We calculate the angle for each axis
    angles = np.linspace(0, 2 * np.pi, num_vars,
endpoint=False).tolist()

    # We close the graph in a circular way
    values += values[:1]
    angles += angles[:1]

    # Create the radial chart
    fig, ax = plt.subplots(figsize=(6, 6),
subplot_kw=dict(polar=True))

    # Create the chart on the radar
    ax.fill(angles, values, color='blue', alpha=0.25)
    ax.plot(angles, values, color='blue', linewidth=2)
```

```
    # Labels for each axis
    ax.set_yticklabels([])
    ax.set_xticks(angles[:-1])
    ax.set_xticklabels(categories, fontsize=12, ha='right')

    # Title
    ax.set_title('Average Characteristics of the Setosa
Species in the Iris Dataset', size=14, color='black',
weight='bold')

    # Show the graph
    plt.tight_layout()
    plt.show()
```

Fig. 19 Average characteristics of the setosa species in the Iris Dataset using radial graph. Own source, 2025.

B.2 Combined Charts

In many cases, it is necessary to use two graphs of different nature within the same visualization to see better results. Therefore, we can propose graphs that are "partitioned" and hide some limits such as xticks and yticks, which will allow us to add a simple histogram on the right side.

In the following code snippet, we will seek to incorporate three variables (sepal length and width, and petal length) along with the percentage distribution of subspecies.

```
    # Create sub-graphics
    fig, axes = plt.subplots(3, 2,
gridspec_kw={'width_ratios': [5, 1]}, figsize=(12, 10),
sharey='row')

    species = sorted(data['Species'].unique())
    features = ['Sepal Length', 'Sepal Width', 'Petal
Length']
    colors = ['blue', 'green', 'red'] # Consistent colors

    for i, feature in enumerate(features):
    #KDE per species with consistent colors
    for j, spec in enumerate(species):
    sns.kdeplot(
    data=data[data['Species'] == spec],
    x=feature,
    ax=axes[i, 0],
    fill=True,
    color=colors[j],
    alpha=0.5,
    label=spec
    )
    axes[i, 0].set_title(f'Distribution of {feature}',
loc='left', fontsize=12, fontweight='bold')
    axes[i, 0].set_xlim(data[feature].min() - 0.5,
data[feature].max() + 0.5)
    axes[i, 0].set_xlabel('')
    axes[i, 0].set_ylabel('Density' if i == 0 else '')
    axes[i, 0].legend([], [], frameon=False)  # Ocultar
leyenda

    # Percentage bar with consistent colors
    species_counts =
data['Species'].value_counts(normalize=True) * 100
    axes[i, 1].barh(range(len(species_counts)),
species_counts, color=colors, alpha=0.7)
    axes[i, 1].set_yticks(range(len(species_counts)))
    axes[i, 1].set_yticklabels(species)
    axes[i, 1].set_xlim(0, 100)
    if i != len(features) - 1:
    axes[i, 1].set_xticks([]) # Hide ticks on intermediate
charts

    # Adjust layout
    plt.tight_layout()
    plt.show()
```

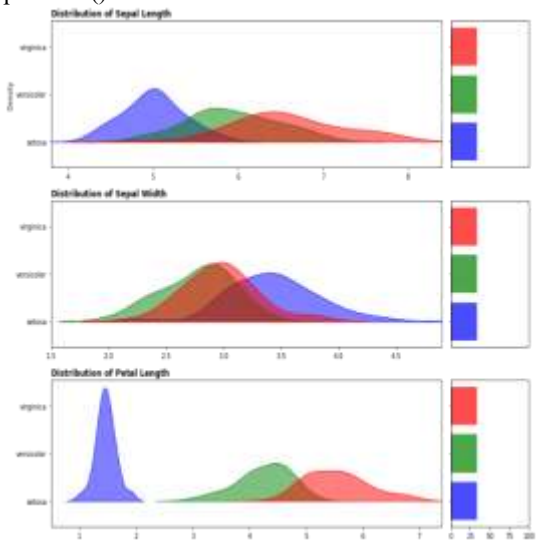


Fig. 20 Distribution of sepal width, length, and petal length of the iris dataset using combo plots. Own source, 2025.

As we observed, the plt.subplots(3, 2) statement allows the analyst to create the combined chart, using 3 rows and 2 columns. Then, the variables are declared and a nested loop is introduced that seeks to obtain the estimated density for each species.

As a result, we will obtain from the right side that the filled areas represent the relative probability of finding a specific value of the characteristic, separated by species. This allows us to compare the differences in the distributions of each characteristic between species. On the left, on the other hand, the percentage of each

subspecies in the global dataset, which coincides because the number of observations is the same per category.

B.3 3D graphics

If we are looking to compare three variables, we cannot use a two-dimensional graph. In these cases, 3D graphics can represent the information more intuitively, using coordinates based on the three selected variables. In the example below, the variables are colored by subspecies and will be represented: sepal width, sepal length, and petal length.

```
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')

# Define the three variables for the 3D chart
x = data['Sepal length']
y = data['Sepal width']
z = data['Petal length']

# Colors for each species
colors = data['Species'].map({'setosa': 'r', 'versicolor': 'g', 'virginica': 'b'})

# Create the 3D graphic
ax.scatter(x, y, z, c=colors, s=50, marker='o')

# Labels for the axes
ax.set_xlabel('Sepal Length (cm)')
ax.set_ylabel('Sepal Width (cm)')
ax.set_zlabel('Petal Length (cm)')

# Chart Title
ax.set_title('3D Graph: Sepal Length, Sepal Width and Petal Length')

# Show the graph
plt.show()
```

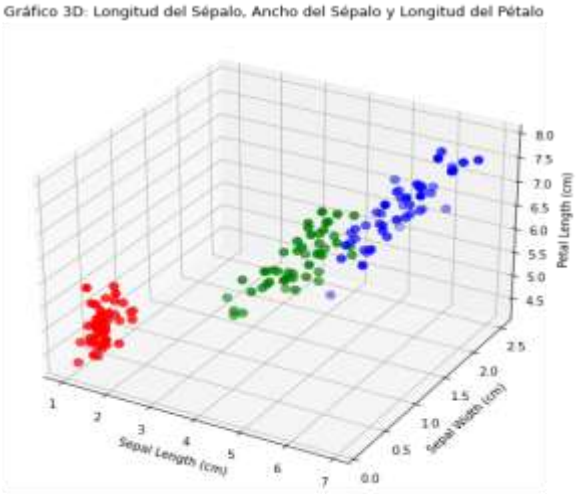


Fig. 21 3D Graph: Sepal's Length, Sepal's Width, and Petal Length. Own source, 2025.

B. Clustering visual

Clustering or also known as "Grouping" is a process that involves dividing objects into several groups, which are integrated as similar elements. In other words, it is a process very similar to what humans would perform when grouping visually similar elements together, something that is possible without any prior training. Although it seems an intuitive method from the point of view of our species, clustering represents a challenge for computational systems.

The method we will discuss is known as K-Means or K-Means. This is an algorithm that works by categorizing data points into groups using a mathematical distance measure, usually Euclidean. The Euclidean distance is calculated from the Pythagorean theorem and is represented as a straight line that draws the shortest path between two points.

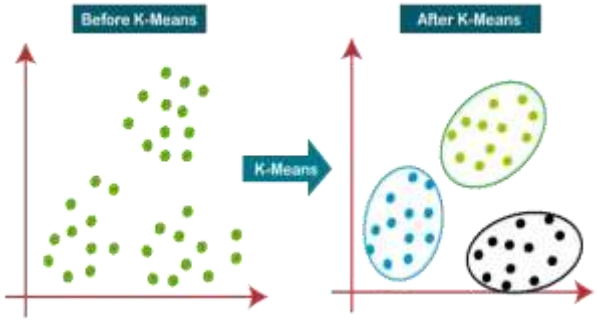


Fig. 22 Introduction to K-Means Clustering, J. Rinkal, 2024. [https://www.bombaysoftwares.com/blog/introduction-to-k-means-clustering]

As we can see, the objective is to minimize the sum of distances between points and assigned groups. However, the main drawback in these visualizations is that we must work in two-dimensional spaces. In the Iris dataset, we must analyze four variables, which is why it is necessary to reduce the dimensionality.

To solve the problem, a technique known as PCA or Principal Component Analysis is usually used. This process involves a new representation of the data, using a new coordinate system that takes into account the variability of the original data.

In the complete excerpt below, we will see how to implement it to achieve a graph using Clustering with grouping by K-Means

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.datasets import load_iris
from sklearn.decomposition import PCA

# Load the Iris dataset
iris = load_iris()
X = iris.data
y = iris.target
```

```
# Apply PCA to reduce dimensionality to 2D (for visualizing)
pca = pca(n_components=2)
X_2d = pca.fit_transform(X)

# Apply KMeans
kmeans = KMeans(n_clusters=3)
kmeans.fit(X_2d)
y_kmeans = kmeans.predict(X_2d)

# Visualize the clusters
plt.figure(figsize=(8, 6))
sns.scatterplot(x=X_2d[:, 0], y=X_2d[:, 1],
hue=y_kmeans, palette='viridis', s=100, alpha=0.7,
edgecolor='k')

# Centroids
centroids = kmeans.cluster_centers_
plt.scatter(centroids[:, 0], centroids[:, 1], c='red', s=200,
marker='X', label="Centroides")
plt.title('K-Means Clustering con Iris Dataset')
plt.xlabel('PCA 1')
plt.ylabel('PCA 2')
plt.legend()
plt.show()
```

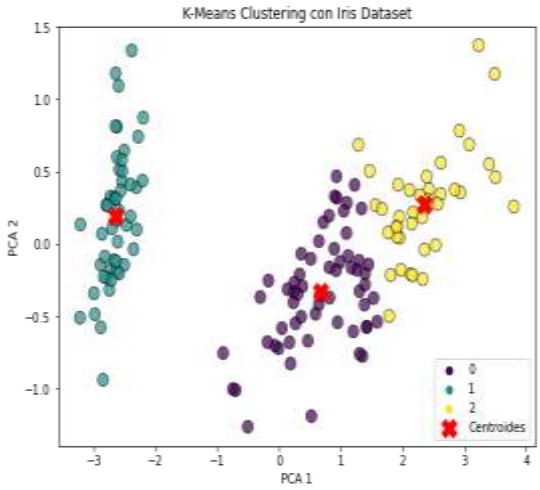


Fig. 23 Using K-Means Clustering in Iris dataset. Own source, 2025.

In the visualization of Fig 23 we can see that there are three centroids, around which the points are distributed. This indicates that these points share similar characteristics and therefore belong to the same cluster.

In the case of the Iris Dataset, we already have predefined categories, so in the example above, the original subspecies tags were not used. However, it can be a useful practice to uncover hidden patterns or to determine if the original categorization was done correctly. In case the analyst does not have the subspecies information, he can obtain an approximation to the labels through this method.

If we compare with the Scatter Graph Matrix in Fig. 16, it is possible to identify the subspecies labeled in the Clustering as 0, 1 and 2.

VIII. SYNTHESIZING INSIGHTS FROM DATA GRAPHS

While data visualization is a key tool in analysis, the true value of these resources lies not only in their appearance, but also in the researcher's ability to draw conclusions from them. Therefore, in addition to working on visual representations, it is important to propose useful interpretations for decision-making.

Definitely, the detection of patterns and trends are the most relevant objectives within a data analysis, since they allow predicting behaviors and reaching relevant conclusions about the object under study. To achieve this, there are some simple practices that are summarized in Table 2.

BOARD III IDENTIFYING PATTERNS AND TRENDS IN DATA ANALYTICS

Source: Own, 2025.

Concepts	Description	Examples
Patterns	Repetition or structure that emerges in the data, revealing constant relationships or behaviors	Observe whether the data is distributed evenly, biased, or normally. Correlation between variables, positive or negative. In time series, identify recurring cycles that suggest a seasonal or periodic pattern, such as monthly fluctuations in sales or temperatures.
Trending	General direction of data over time or between variables. Identification of upward, downward or stable movements.	Observe whether the values increase or decrease progressively over time or according to a variable. Analyze data variability

Once patterns and trends have been identified, it is important to include them in the context of the research. At this point, the analyst will be faced with the task of answering the following questions:

- What do the patterns indicate about the behavior of the variables?

- What explanation could they have within the context?
- What factors might be influencing trends and patterns?
- What implications do the patterns found have for the practical decisions that will be made from this research?

To answer these questions, it is essential to consider the nature of the research and its purpose. Depending on whether the research is biological, financial, mathematical, or otherwise, the patterns can have very different implications.

Throughout the article, the example of Iris was analyzed and phenomena associated with evolution, adaptation, and physiological responses of each subspecies were observed. However, financial research could be understood from the perspective of market behavior, regional policies, and asset dynamics. Each type of research requires an adaptation of the analysis and an interpretation of the patterns that is aligned with its specific context, its objectives and its limitations.

IX. Conclusions

In the era of Big Data, data analysis has established itself as one of the most crucial tasks. As highlighted at the beginning of the article, having large volumes of data alone has no real value. Their true potential lies in the ability to process, interpret and transform them into valuable information.

The role of the data analyst, therefore, is not only to understand this data, but also to extract patterns, identify trends, and present them in a way that others can understand and apply this knowledge effectively. Their ability to transform raw data into accessible and useful insights is what ultimately gives value and relevance to information in the context of decision-making.

Throughout the analysis, various techniques in Python and some tools that the analyst can use to extract meaningful information were highlighted. Surely, these tools alone would never achieve a deep understanding, since this task only involves knowing about their distributions and groupings. A data analysis professional must understand that data is immersed in a context and through these tools, they must be able to unravel its meaning.

References

[1] AnotherOrion, "Definition of ANOVA in Research," AnotherOrion. [Online]. Available: https://anotherorion.com/pengertian-anova-dalam-penelitian/. Accessed: Jan. 3, 2025.

[2] D. Ávila and V. Ramírez-Arrieta, "If a picture is worth 1000 words: how much can a box graph say?," ResearchGate, 2020. [Online]. Available: https://doi.org/10.13140/RG.2.2.34348.28806/1. Accessed: Jan. 3, 2025.

[3] P. Bruce, A. Bruce, and Y. P. Gedeck, Practical Statistics for Data Science with R and Python, O'Reilly Media, 2020.

[4] S. J. Dagnino, "Correlacion," Revista Chilena Anesthesia, vol. 43, no. 2, pp. 150–153, 2019. [Online]. Available: https://doi.org/10.25237/revchilanestv43n02.15.

[5] Matepedia Statistics, "Characteristics of a Distribution," Mathematics Statistics. [Online]. Available: https://matepedia-estadistica.blogspot.com/2016/09/caracteristicas-de-una-distribucion.html. Accessed: Jan. 3, 2025.

[6] J. Rinkal, "Introduction to K-Means Clustering," Bombay Softwares. [Online]. Available: https://www.bombaysoftwares.com/blog/introduction-to-k-means-clustering. Accessed: Jan. 3, 2025.

[7] A. Rodríguez, E. Cuevas, D. Zaldivar, and L. Castañeda, "Clustering with biological visual models," Physica A: Statistical Mechanics and its Applications, vol. 528, 2019. [Online]. Available: https://doi.org/10.1016/j.physa.2019.121505.

[8] A. Rodríguez, E. Cuevas, D. Zaldivar, and L. Castañeda, "If a picture is worth 1000 words: how much can a box graph say?," Physica A: Statistical Mechanics and its Applications, vol. 528, 2019. [Online]. Available: https://doi.org/10.13140/RG.2.2.34348.28806/1.

[9] 3TW, "Scikit-learn, the Iris Dataset, and Machine Learning: The Journey to a New Skill," Medium. [Online]. Available: https://3tw.medium.com/scikit-learn-the-iris-dataset-and-machine-learning-the-journey-to-a-new-skill-c8d2f537e087. Accessed: Jan. 3, 2025.