```
# Define where you are running the code: colab or local
RunInColab          = True     # (False: no  | True: yes)

# If running in colab:
if RunInColab:
    # Mount your google drive in google colab
    from google.colab import drive
    drive.mount('/content/drive')

    # Find location
    #!pwd
    #!ls
    #!ls "/content/drive/My Drive/Colab Notebooks/MachineLearningWithPython/"

    # Define path del proyecto
    Ruta = '/content/drive/My Drive/NotebooksProfessorMio/'

else:
    # Define path del proyecto
    Ruta            = ""

# Import the packages that we will be using

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns




# Dataset url

url = "datasets/iris.csv"

# Load the dataset

df = pd.read_csv(Ruta + url)
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=Tr
```

```
#How many observations (rows) are in total?
num_rows = df.shape[0]
print(f"Total number of observations (rows): {num_rows}")


#How many variables (columns) are in total? What do they represent?
num_columns = df.shape[1]
print(f"Total number of variables (columns): {num_columns}")
print(f"The columns are: {df.columns.tolist()}")

#How many observations are for each type of flower?
flower_counts = df['variety'].value_counts()
print("Observations for each type of flower:")
print(flower_counts)

#What is the type of data for each variable?
data_types = df.dtypes
print("Data types for each variable:")
print(data_types)

#What are the units of each variable?
# The units must be cm for all variables except for the Species which is a string
```

```
Total number of observations (rows): 150
Total number of variables (columns): 5
The columns are: ['sepal.length', 'sepal.width', 'petal.length', 'petal.width', 'variety']
Observations for each type of flower:
variety
Setosa        50
Versicolor    50
Virginica     50
Name: count, dtype: int64
Data types for each variable:
```

```
sepal.length    float64
sepal.width     float64
petal.length    float64
petal.width     float64
variety          object
dtype: object
```

```
#Calculate the statistical summary for each quantitative variables. Explain the results
stat_summary = df.describe()
print("Statistical Summary for Quantitative Variables:")
print(stat_summary)
#Identify the name of each column
column_names = df.columns.tolist()
print(f"Column Names: {column_names}")
#Identify the type of each column
column_types = df.dtypes
print("Data Types of Each Column:")
print(column_types)
#Minimum, maximum, mean, average, median, standar deviation

numeric_columns = df.select_dtypes(include=[np.number])


min_values = numeric_columns.min()
max_values = numeric_columns.max()
mean_values = numeric_columns.mean()
median_values = numeric_columns.median()
std_dev_values = numeric_columns.std()

print("Min Values:")
print(min_values)
print("Max Values:")
print(max_values)
print("Mean Values:")
print(mean_values)
print("Median Value:")
print(median_values)
print("Standard Deviation:")
print(std_dev_values)
#Are there missing data? If so, create a new dataset containing only the rows with the non-missing data
missing_data = df.isnull().sum()
print("Missing Data:")
print(missing_data)
df_no_missing = df.dropna()
print(f"New dataset without missing data contains {df_no_missing.shape[0]} rows.")
#Create a new dataset containing only the petal width and length and the type of Flower
df_petal = df[['petal.width', 'petal.length', 'variety']]
print("Petal Width, Petal Length, and Type of Flower:")
print(df_petal.head())
#Create a new dataset containing only the setal width and length and the type of Flower
df_sepal = df[['sepal.width', 'sepal.length', 'variety']]
print("Dataset with Sepal Width, Sepal Length, and Type of Flower:")
print(df_sepal.head())
#Create a new dataset containing the setal width and length and the type of Flower encoded as a categorical numerical column
df_sepal_encoded = df_sepal.copy()
df_sepal_encoded['species_encoded'] = df_sepal_encoded['variety'].astype('category').cat.codes
print("Dataset with Sepal Width, Sepal Length, and Encoded Flower Type:")
print(df_sepal_encoded.head())
```

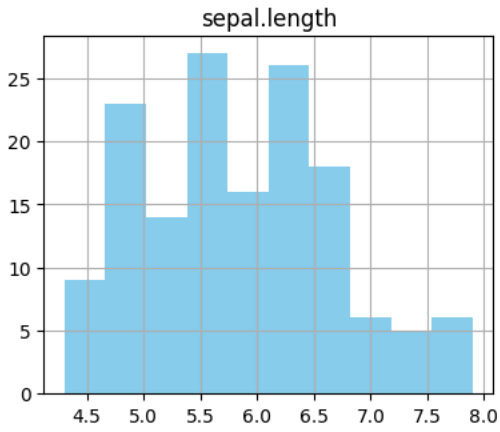
```
sepal.length    4.3
```

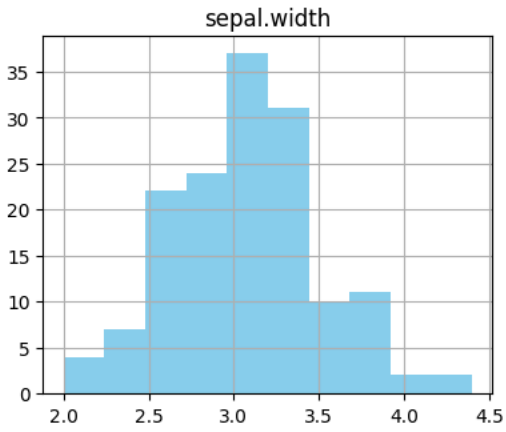
```
petal.length    4.35
petal.width     1.30
dtype: float64
Standard Deviation:
sepal.length    0.828066
sepal.width     0.435866
petal.length    1.765298
petal.width     0.762238
dtype: float64
Missing Data:
sepal.length    0
sepal.width     0
petal.length    0
petal.width     0
variety         0
dtype: int64
New dataset without missing data contains 150 rows.
Petal Width, Petal Length, and Type of Flower:
   petal.width  petal.length variety
0          0.2           1.4  Setosa
1          0.2           1.4  Setosa
2          0.2           1.3  Setosa
3          0.2           1.5  Setosa
4          0.2           1.4  Setosa
Dataset with Sepal Width, Sepal Length, and Type of Flower:
   sepal.width  sepal.length variety
0          3.5           5.1  Setosa
1          3.0           4.9  Setosa
2          3.2           4.7  Setosa
3          3.1           4.6  Setosa
4          3.6           5.0  Setosa
Dataset with Sepal Width, Sepal Length, and Encoded Flower Type:
   sepal.width  sepal.length variety  species_encoded
0          3.5           5.1  Setosa                0
1          3.0           4.9  Setosa                0
2          3.2           4.7  Setosa                0
3          3.1           4.6  Setosa                0
4          3.6           5.0  Setosa                0
```

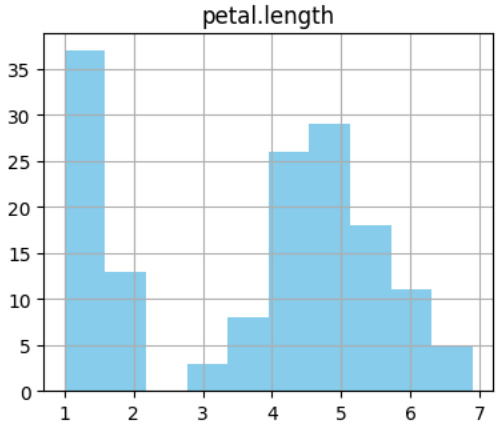
Haz doble clic (o ingresa) para editar

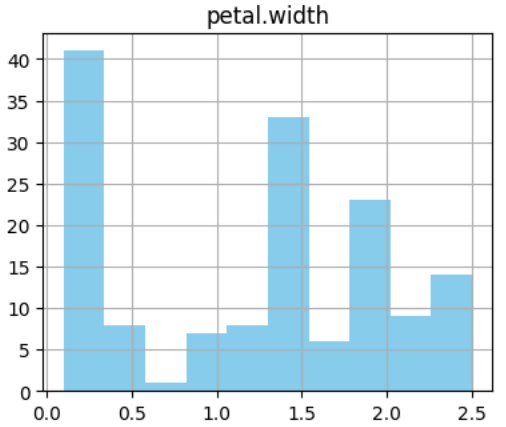
```
#Plot the histograms for each of the four quantitative variables
df.hist(column=['sepal.length', 'sepal.width', 'petal.length', 'petal.width'], figsize=(10, 8), color='skyblue')
plt.suptitle('Histograms:')
plt.show()

#Plot the histograms for each of the quantitative variables
df.hist(column=['sepal.length', 'sepal.width', 'petal.length', 'petal.width'], figsize=(10, 8), color='skyblue')
plt.suptitle('Histograms:')
plt.show()
#Plot the boxplots for each of the quantitative variables

plt.figure(figsize=(12, 8))
sns.boxplot(data=df[['sepal.length', 'sepal.width', 'petal.length', 'petal.width']])
plt.title('Boxplots:')
plt.show()

#Plot the boxplots of the petal width grouped by type of flower

plt.figure(figsize=(8, 6))
sns.boxplot(x='variety', y='petal.width', data=df)
plt.title('Boxplot:')
plt.show()

#Plot the boxplots of the setal length grouped by type of flower

plt.figure(figsize=(8, 6))
sns.boxplot(x='variety', y='sepal.length', data=df)
plt.title('Boxplot:')
plt.show()

#Provide a description (explaination from your observations) of each of the quantitative variables
#The sepal length had a value of around 5.5 and 6.5 for about 25 times.
#The sepal width appeared to be between 3 and 3.5 for about 35 times
#The petal length seemed to be around 1 35 times and 5 27 times
#The petal width was below o.5 for about 42 times and 1.5 for about 32.5 times
```

Histograms:

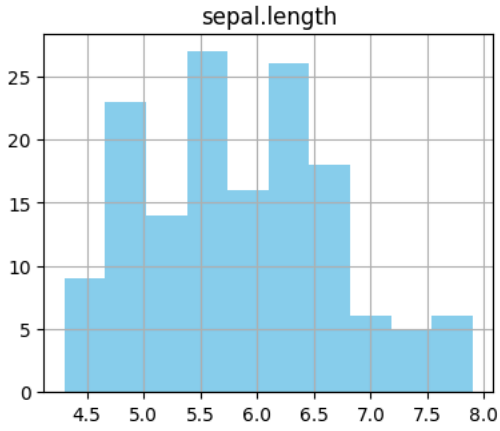


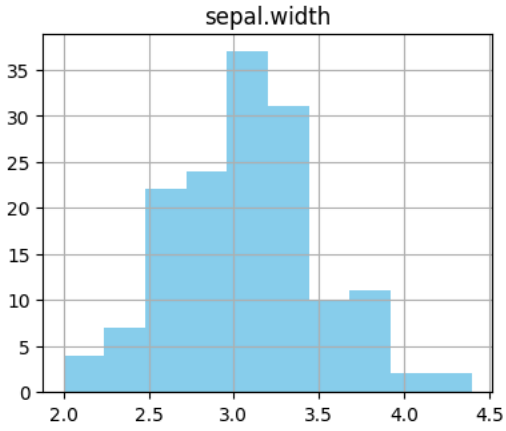


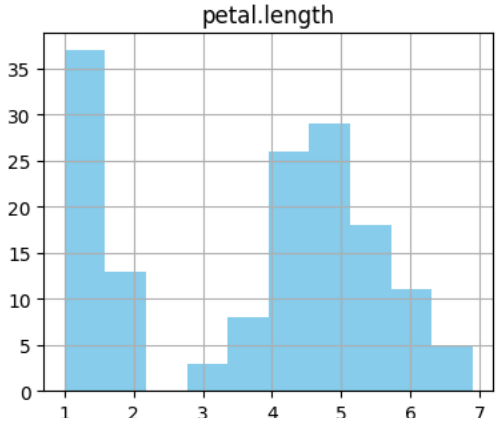


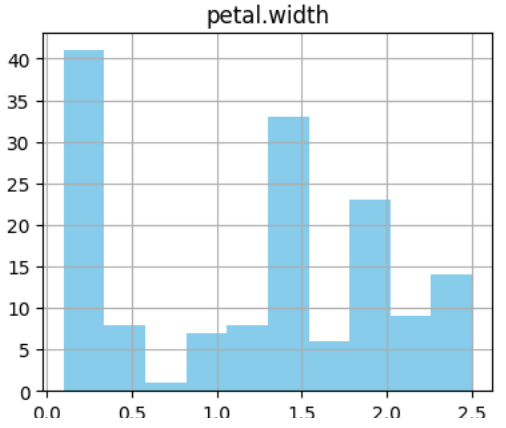


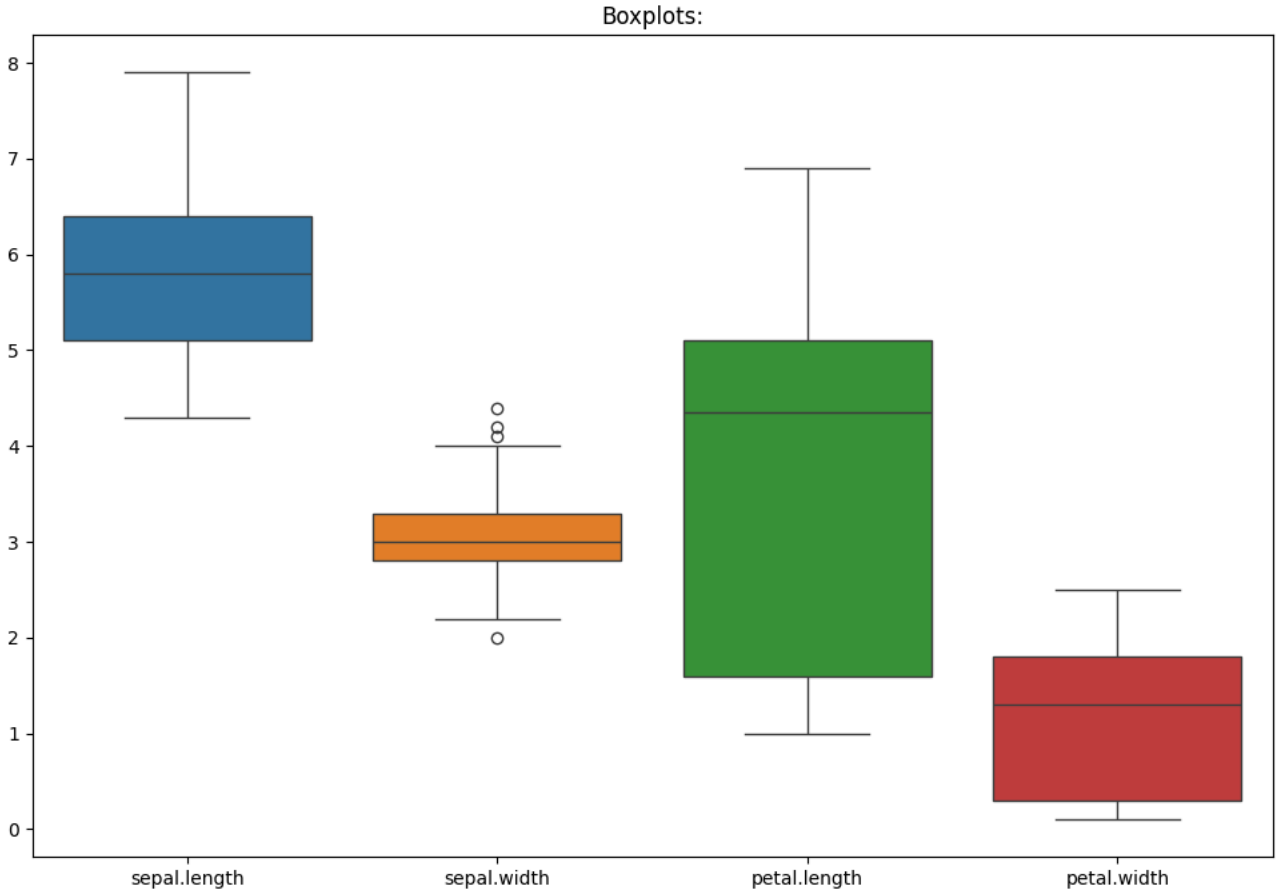
Histograms:

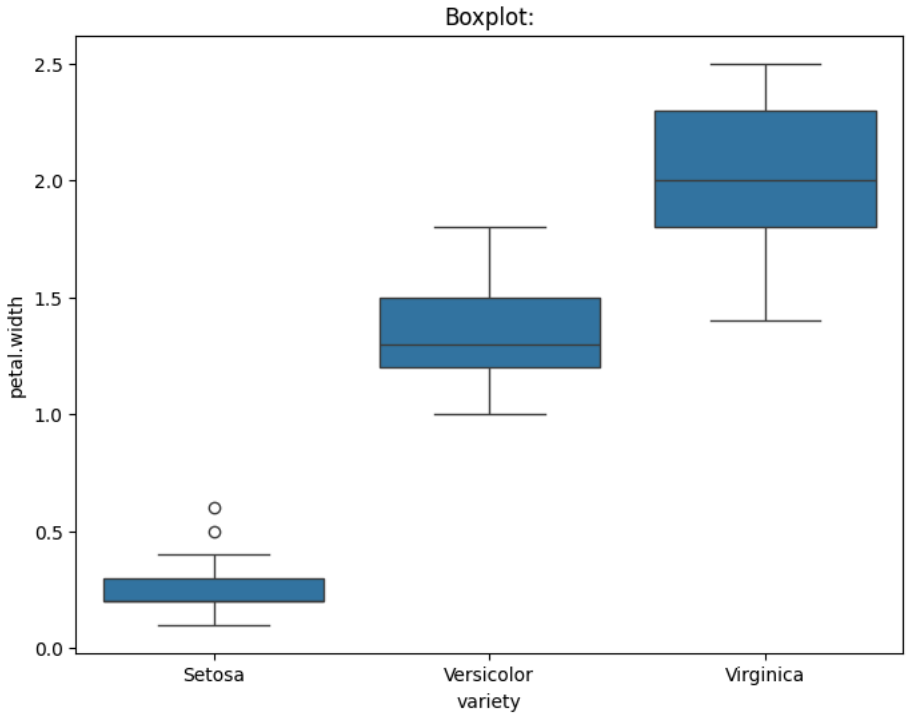


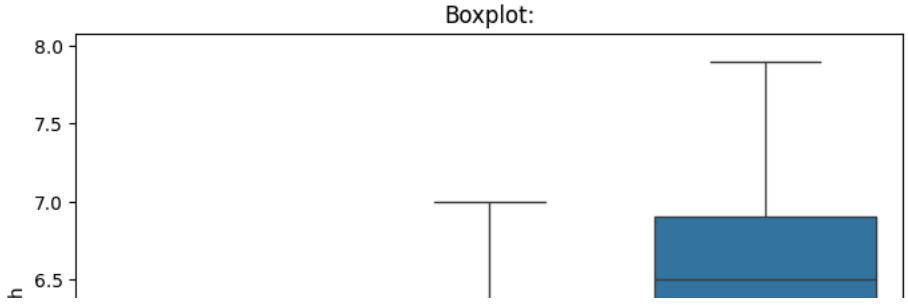


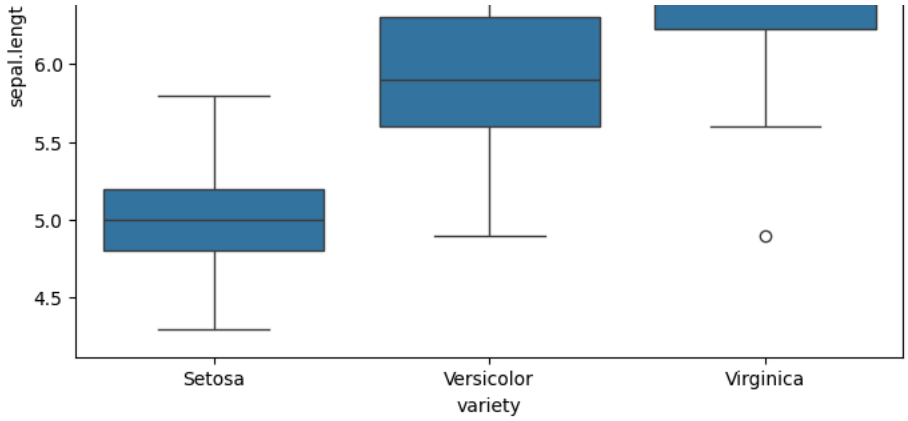





Boxplots:
8
7
6
5
4
3
2
1
0
sepal.length
sepal.width
petal.length
petal.width


Boxplot:
petal.width
2.5
2.0
1.5
1.0
0.5
0.0
Setosa
Versicolor
Virginica
variety


Boxplot:
8.0
7.5
7.0
6.5



Haz doble clic (o ingresa) para editar