Manuel Alejandro Preciado Morán A01639643

This document contains all the activities related to the iris database

In [ ]:

```
# Import the packages that we will be using
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import csv

# Dataset url
path = "/home/alex/TC1002S/NotebooksStudents/A01639643/iris/iris.csv"
header = ["sepal_length", "sepal_width", "petal_length", "petal_width", "Clas
# Load the dataset
ds = pd.read_csv(path, names=header)

# Print the dataset
ds
```

Out[ ]:

	sepal_length	sepal_width	petal_length	petal_width	Class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

Activity 1:

How many observations (rows) are in total? 150 How many variables (columns) are in total? What do they represent? 5 How many observations are for each type of flower? Iris Setosa: 50 Iris Versicolour: 50 Iris Virginica: 50

What is the type of data for each variable? float object

What are the units of each variable? cm

In [ ]:

```
#num of rows
ds.shape[0]
```

Out[ ]:

```
150
```

In [ ]:

```
#num of col
ds.shape[1]
```

Out[ ]:

```
5
```

In [ ]:

```
#How many observations are for each type of flower
sC = ds["Class"]
counts = pd.Series(sC).value_counts()
print("Iris-setosa: ", counts.get("Iris-setosa"))
print("Iris-virginica: ", counts.get("Iris-virginica"))
print("Iris-versicolor: ", counts.get("Iris-versicolor"))
```

In [ ]:

```
#Type of data for each variable
ds.dtypes
```

Out[ ]:

```
sepal_length    float64
sepal_width     float64
petal_length    float64
petal_width     float64
Class            object
dtype: object
```

Activity2: work with the iris dataset

Repeat this tutorial with the iris data set and respond to the following inquiries

```
Calculate the statistical summary for each quantitative
variables. Explain the results
    Identify the name of each column
    Identify the type of each column
    Minimum, maximum, mean, average, median, standar deviation

Are there missing data? If so, create a new dataset containing
only the rows with the non-missing data
NO

Create a new dataset containing only the petal width and length
and the type of Flower

Create a new dataset containing only the setal width and length
and the type of Flower

Create a new dataset containing the setal width and length and
the type of Flower encoded as a categorical numerical column
```

In [ ]:

```
#Identifty the name of each column
ds.columns
```

Out[ ]:

```
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'Class
'], dtype='object')
```

In [ ]:

```
#Identifty the type of each column
ds.dtypes
```

Out[ ]:

```
sepal_length    float64
sepal_width     float64
petal_length    float64
petal_width     float64
Class            object
dtype: object
```

In [ ]:

```
#Minimum, maximum, mean, average, median, standar deviation
ds.describe()
```

Out[ ]:

	sepal_length	sepal_width	petal_length	petal_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

In [ ]:

```
# Create a new dataset containing only the petal width and length and the typ
ds1 = ds.iloc[:,2:4]
ds1
```

Out[ ]:

	petal_length	petal_width
0	1.4	0.2
1	1.4	0.2
2	1.3	0.2
3	1.5	0.2
4	1.4	0.2
...	...	...
145	5.2	2.3
146	5.0	1.9
147	5.2	2.0

	petal_length	petal_width
148	5.4	2.3
149	5.1	1.8

In [ ]:

```
# Create a new dataset containing only the setal width and length and the typ
ds2 = ds.iloc[:,0:2]
ds2
```

Out[ ]:

	sepal_length	sepal_width
0	5.1	3.5
1	4.9	3.0
2	4.7	3.2
3	4.6	3.1
4	5.0	3.6
...	...	...
145	6.7	3.0
146	6.3	2.5
147	6.5	3.0
148	6.2	3.4
149	5.9	3.0

150 rows × 2 columns

In [ ]:

```
# Create a new dataset containing the setal width and length and the type of
ds2 = ds.loc[:,["sepal_length", "sepal_width", "Class"]]
ds2["Class"] = ds2.Class.replace({"Iris-setosa":1, "Iris-versicolor":2, "Iris
ds2
```

Out[ ]:

	sepal_length	sepal_width	Class
0	5.1	3.5	1
1	4.9	3.0	1
2	4.7	3.2	1
3	4.6	3.1	1
4	5.0	3.6	1
...	...	...	...
145	6.7	3.0	3
146	6.3	2.5	3
147	6.5	3.0	3
148	6.2	3.4	3