Nombre: León Emiliano García Pérez

Matrícula: A00573074

Repeating the Tutorial with de Iris Dataset

In [128…
```
# Define where you are running the code: colab or local
RunInColab          = False     # (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            = ""

else:
    # Define path del proyecto
    Ruta            = 'datasets'
```

In [129…
```
# Import the packages that we will be using

import pandas as pd

# Dataset url

url = '/iris/iris.csv'

# Load the dataset

df = pd.read_csv(Ruta+url)
df.columns = ["PetalWidth", "PetalLength", "SepalWidth", "SepalLength", "Type"]
```

In [130…
```
type(df)
```

Out[130]:
```
pandas.core.frame.DataFrame
```

In [131…
```
df.shape
```

Out[131]:
```
(149, 5)
```

In [132…
```
df.shape[0]
```

Out[132]:
```
149
```

In [133…
```
df.shape[1]
```

Out[133]:
```
5
```

In [134…
```
df
```

Out[134]:

	PetalWidth	PetalLength	SepalWidth	SepalLength	Type
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa
...	...	...	...	...	...
144	6.7	3.0	5.2	2.3	Iris-virginica
145	6.3	2.5	5.0	1.9	Iris-virginica
146	6.5	3.0	5.2	2.0	Iris-virginica
147	6.2	3.4	5.4	2.3	Iris-virginica
148	5.9	3.0	5.1	1.8	Iris-virginica

149 rows × 5 columns

In [135…
```
df.head()
```

Out[135]:

	PetalWidth	PetalLength	SepalWidth	SepalLength	Type
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa

In [136…
```
df.tail()
```

Out[136]:

	PetalWidth	PetalLength	SepalWidth	SepalLength	Type
144	6.7	3.0	5.2	2.3	Iris-virginica
145	6.3	2.5	5.0	1.9	Iris-virginica
146	6.5	3.0	5.2	2.0	Iris-virginica
147	6.2	3.4	5.4	2.3	Iris-virginica
148	5.9	3.0	5.1	1.8	Iris-virginica

In [137…
```
df.columns
```

Out[137]:
```
Index(['PetalWidth', 'PetalLength', 'SepalWidth', 'SepalLength', 'Type'], dtype='o
bject')
```

In [138…
```
df.dtypes
```

Out[138]:
```
PetalWidth     float64
PetalLength    float64
SepalWidth     float64
SepalLength    float64
Type            object
dtype: object
```

In [139…
```
# Summary statistics for the quantitative variables

df.info()

df.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 149 entries, 0 to 148
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PetalWidth   149 non-null    float64
 1   PetalLength  149 non-null    float64
 2   SepalWidth   149 non-null    float64
 3   SepalLength  149 non-null    float64
 4   Type         149 non-null    object 
dtypes: float64(4), object(1)
memory usage: 5.9+ KB
```

Out[139]:

	PetalWidth	PetalLength	SepalWidth	SepalLength
count	149.000000	149.000000	149.000000	149.000000
mean	5.848322	3.054362	3.773826	1.206040
std	0.828594	0.435810	1.760543	0.760354
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.400000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [140…
```
# Drop observations with NaN values

df.PetalWidth.dropna().describe()
df.PetalLength.dropna().describe()
df.SepalWidth.dropna().describe()
df.SepalLength.dropna().describe()
```

Out[140]:
```
count    149.000000
mean       1.206040
std        0.760354
min        0.100000
25%        0.300000
50%        1.300000
75%        1.800000
max        2.500000
Name: SepalLength, dtype: float64
```

In [141…
```
#print("mean")
#df.mean()
#print("corr")
#df.corr()
#print("count")
#df.count()
#print("max")
#df.max()
#print("min")
#df.min()
#print("median")
#df.median()
#print("std")
df.std() ################### El único que, al menos en este tutorial, imprimió datos
```

```
C:\Users\LeonE\AppData\Local\Temp\ipykernel_7384\2295696299.py:14: FutureWarning:
The default value of numeric_only in DataFrame.std is deprecated. In a future vers
ion, it will default to False. In addition, specifying 'numeric_only=None' is depr
ecated. Select only valid columns or specify the value of numeric_only to silence
this warning.
  df.std() ################### El único que, al menos en este tutorial, imprimió da
tos. CON WARNING.
```

Out[141]:
```
PetalWidth     0.828594
PetalLength    0.435810
SepalWidth     1.760543
SepalLength    0.760354
dtype: float64
```

In [142…
```
df.to_csv('myNewDataFrame.csv')
df.to_csv('myNewDataFrame.csv', sep='\t')
```

In [143…
```
df = df.rename(columns={"PetalWidth": "AnchoPetal"})
df.head()
```

Out[143]:

	AnchoPetal	PetalLength	SepalWidth	SepalLength	Type
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa

In [144…
```
# Back to the original name

df = df.rename(columns={"AnchoPetal": "PetalWidth"})

df.head()
```

Out[144]:

	PetalWidth	PetalLength	SepalWidth	SepalLength	Type
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa

In [145…

```
a = df.PetalWidth
b = df["PetalLength"]
c = df.loc[:, "PetalLength"]
d = df.iloc[:, 1]

print(d)

e = df[["SepalWidth", "SepalWidth"]]

print(e)
```

```
0      3.0
1      3.2
2      3.1
3      3.6
4      3.9
      ...
144    3.0
145    2.5
146    3.0
147    3.4
148    3.0
Name: PetalLength, Length: 149, dtype: float64
     SepalWidth  SepalWidth
0           1.4         1.4
1           1.3         1.3
2           1.5         1.5
3           1.4         1.4
4           1.7         1.7
..          ...         ...
144         5.2         5.2
145         5.0         5.0
146         5.2         5.2
147         5.4         5.4
148         5.1         5.1

[149 rows x 2 columns]
```

In [146…

```
df.loc[:,"PetalLength"]


df.loc[:9, "PetalLength"]


df.loc[:,["SepalWidth", "SepalLength"]]


keep = ['PetalWidth', 'PetalLength']
df_gender = df[keep]
```

```
df.loc[4:9, ["SepalWidth", "PetalWidth", "Type"]]


df.loc[10:15,:]
```

Out[146]:

	PetalWidth	PetalLength	SepalWidth	SepalLength	Type
10	4.8	3.4	1.6	0.2	Iris-setosa
11	4.8	3.0	1.4	0.1	Iris-setosa
12	4.3	3.0	1.1	0.1	Iris-setosa
13	5.8	4.0	1.2	0.2	Iris-setosa
14	5.7	4.4	1.5	0.4	Iris-setosa
15	5.4	3.9	1.3	0.4	Iris-setosa

In [147…

```
df.iloc[:, :4]


df.iloc[:4, :]


df.iloc[:, 3:7]


df.iloc[4:8, 2:4]
```

Out[147]:

	SepalWidth	SepalLength
4	1.7	0.4
5	1.4	0.3
6	1.5	0.2
7	1.4	0.2

In [148…

```
df.Type.unique()
```

Out[148]:

```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

In [149…

```
df[df["SepalLength"] >= 1.0]
```

Out[149]:

	PetalWidth	PetalLength	SepalWidth	SepalLength	Type
49	7.0	3.2	4.7	1.4	Iris-versicolor
50	6.4	3.2	4.5	1.5	Iris-versicolor
51	6.9	3.1	4.9	1.5	Iris-versicolor
52	5.5	2.3	4.0	1.3	Iris-versicolor
53	6.5	2.8	4.6	1.5	Iris-versicolor
...	...	...	...	...	...
144	6.7	3.0	5.2	2.3	Iris-virginica
145	6.3	2.5	5.0	1.9	Iris-virginica
146	6.5	3.0	5.2	2.0	Iris-virginica
147	6.2	3.4	5.4	2.3	Iris-virginica
148	5.9	3.0	5.1	1.8	Iris-virginica

100 rows × 5 columns

In [150…
```
df.sort_values("PetalWidth")
```

Out[150]:

	PetalWidth	PetalLength	SepalWidth	SepalLength	Type
12	4.3	3.0	1.1	0.1	Iris-setosa
41	4.4	3.2	1.3	0.2	Iris-setosa
37	4.4	3.0	1.3	0.2	Iris-setosa
7	4.4	2.9	1.4	0.2	Iris-setosa
40	4.5	2.3	1.3	0.3	Iris-setosa
...	...	...	...	...	...
117	7.7	2.6	6.9	2.3	Iris-virginica
116	7.7	3.8	6.7	2.2	Iris-virginica
121	7.7	2.8	6.7	2.0	Iris-virginica
134	7.7	3.0	6.1	2.3	Iris-virginica
130	7.9	3.8	6.4	2.0	Iris-virginica

149 rows × 5 columns

In [151…
```
df.groupby(['Type'])
```

Out[151]:
```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x000001DC24B43550>
```

In [152…
```
df.isnull()
df.notnull()
```

Out[152]:

	PetalWidth	PetalLength	SepalWidth	SepalLength	Type
0	True	True	True	True	True
1	True	True	True	True	True
2	True	True	True	True	True
3	True	True	True	True	True
4	True	True	True	True	True
...	...	...	...	...	...
144	True	True	True	True	True
145	True	True	True	True	True
146	True	True	True	True	True
147	True	True	True	True	True
148	True	True	True	True	True

149 rows × 5 columns

In [153...]
```
df.groupby(['Type']).size()
```

Out[153]:
```
Type
Iris-setosa        49
Iris-versicolor    50
Iris-virginica     50
dtype: int64
```

In [154...]
```
df.isnull().sum()
```

Out[154]:
```
PetalWidth     0
PetalLength    0
SepalWidth     0
SepalLength    0
Type           0
dtype: int64
```

In [155...]
```
df.notnull().sum()
```

Out[155]:
```
PetalWidth     149
PetalLength    149
SepalWidth     149
SepalLength    149
Type           149
dtype: int64
```

In [156...]
```
print( df.PetalWidth.notnull().sum() )
print( pd.isnull(df.PetalWidth).sum() )
```

```
149
0
```

In [157...]
```
x = df.Type.dropna()
x.describe()
```

Out[157]:
```
count                 149
unique                  3
top       Iris-versicolor
freq                   50
Name: Type, dtype: object
```

In [158…
```
df.head()
```

Out[158]:

	PetalWidth	PetalLength	SepalWidth	SepalLength	Type
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa

In [159…
```
# Add a new column with new data

# Create a column data
NewColumnData = df.PetalWidth/df.PetalLength

# Insert that column in the data frame
df.insert(5, "PetalWidth/PetalLength", NewColumnData, True)

df.head()
```

Out[159]:

	PetalWidth	PetalLength	SepalWidth	SepalLength	Type	PetalWidth/PetalLength
0	4.9	3.0	1.4	0.2	Iris-setosa	1.633333
1	4.7	3.2	1.3	0.2	Iris-setosa	1.468750
2	4.6	3.1	1.5	0.2	Iris-setosa	1.483871
3	5.0	3.6	1.4	0.2	Iris-setosa	1.388889
4	5.4	3.9	1.7	0.4	Iris-setosa	1.384615

In [160…
```
# # Eliminate inserted column
#df.drop("PetalWidth/PetalLength", axis=1, inplace = True) #SOLO PUEDE SER USADO UN
#df.drop(columns=['PetalWidth/PetalLength'], inplace = True) #SOLO PUEDE SER USADO

# # Remove three columns as index base
df.drop(df.columns[[5]], axis = 1, inplace = True) #SOLO PUEDE SER USADO UNA A LA \

df.head()
```

Out[160]:

	PetalWidth	PetalLength	SepalWidth	SepalLength	Type
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa

In [161…
```
# # Add new column derived from existing columns
#
# # The new column is a function of another column
df["SepalWidth/SepalLength"] = df["SepalWidth"] / df["SepalLength"]
#
df.head()
```

Out[161]:

	PetalWidth	PetalLength	SepalWidth	SepalLength	Type	SepalWidth/SepalLength
0	4.9	3.0	1.4	0.2	Iris-setosa	7.00
1	4.7	3.2	1.3	0.2	Iris-setosa	6.50
2	4.6	3.1	1.5	0.2	Iris-setosa	7.50
3	5.0	3.6	1.4	0.2	Iris-setosa	7.00
4	5.4	3.9	1.7	0.4	Iris-setosa	4.25

In [162…
```
# # Eliminate inserted column
df.drop("SepalWidth/SepalLength", axis=1, inplace = True)
#
df.head()
```

Out[162]:

	PetalWidth	PetalLength	SepalWidth	SepalLength	Type
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa

In [163…
```
df["NumType"] = df.Type.replace({"Iris-setosa":1 , "Iris-virginica": 2, "Iris-virg

# Show the first 5 rows of the created data frame

df.head(5)
```

Out[163]:

	PetalWidth	PetalLength	SepalWidth	SepalLength	Type	NumType
0	4.9	3.0	1.4	0.2	Iris-setosa	1
1	4.7	3.2	1.3	0.2	Iris-setosa	1
2	4.6	3.1	1.5	0.2	Iris-setosa	1
3	5.0	3.6	1.4	0.2	Iris-setosa	1
4	5.4	3.9	1.7	0.4	Iris-setosa	1

In [164…
```
## Eliminate inserted column
df.drop("NumType", axis=1, inplace = True) #SOLO PUEDE SER USADO UNA A LA VEZ, NO S
#df.drop(['NumType'],vaxis='columns',vinplace=True) #SOLO PUEDE SER USADO UNA A LA
```

In [165…
```
## Add a new column with strata based on these cut points
#
## Create a column data
NewColumnData = df.PetalWidth/df.PetalWidth
```

```
#
## Insert that column in the data frame
df.insert(1, "ColumnStrata", NewColumnData, True)
#
df["ColumnStrata"] = pd.cut(df.PetalWidth, [0., 1., 2., 3., 4., 5., 6.])
#
## Show the first 5 rows of the created data frame
df.head()
```

Out[165]:

	PetalWidth	ColumnStrata	PetalLength	SepalWidth	SepalLength	Type
0	4.9	(4.0, 5.0]	3.0	1.4	0.2	Iris-setosa
1	4.7	(4.0, 5.0]	3.2	1.3	0.2	Iris-setosa
2	4.6	(4.0, 5.0]	3.1	1.5	0.2	Iris-setosa
3	5.0	(4.0, 5.0]	3.6	1.4	0.2	Iris-setosa
4	5.4	(5.0, 6.0]	3.9	1.7	0.4	Iris-setosa

In [166…

```
## Eliminate inserted column
df.drop("ColumnStrata", axis=1, inplace = True)
#
df.head()
```

Out[166]:

	PetalWidth	PetalLength	SepalWidth	SepalLength	Type
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa

In [167…

```
# Print tail

df.tail()
```

Out[167]:

	PetalWidth	PetalLength	SepalWidth	SepalLength	Type
144	6.7	3.0	5.2	2.3	Iris-virginica
145	6.3	2.5	5.0	1.9	Iris-virginica
146	6.5	3.0	5.2	2.0	Iris-virginica
147	6.2	3.4	5.4	2.3	Iris-virginica
148	5.9	3.0	5.1	1.8	Iris-virginica

In [168…

```
df.loc[len(df.index)] = [ 5.9, 3.0, 5.1, 1.8, 'Iris-virginica']

df.tail()
```

Out[168]:

	PetalWidth	PetalLength	SepalWidth	SepalLength	Type
145	6.3	2.5	5.0	1.9	Iris-virginica
146	6.5	3.0	5.2	2.0	Iris-virginica
147	6.2	3.4	5.4	2.3	Iris-virginica
148	5.9	3.0	5.1	1.8	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

In [169…

```
## Eliminate inserted row
df.drop([149], inplace = True )
#
df.tail()
```

Out[169]:

	PetalWidth	PetalLength	SepalWidth	SepalLength	Type
144	6.7	3.0	5.2	2.3	Iris-virginica
145	6.3	2.5	5.0	1.9	Iris-virginica
146	6.5	3.0	5.2	2.0	Iris-virginica
147	6.2	3.4	5.4	2.3	Iris-virginica
148	5.9	3.0	5.1	1.8	Iris-virginica

In [170…

```
vars = ["PetalWidth", "PetalLength", "SepalWidth", "Type"]
df = df[vars].dropna() #SOLO PUEDE SER USADO UNA A LA VEZ, NO SE PUEDE DROPEAR ALGO

df
```

Out[170]:

	PetalWidth	PetalLength	SepalWidth	Type
0	4.9	3.0	1.4	Iris-setosa
1	4.7	3.2	1.3	Iris-setosa
2	4.6	3.1	1.5	Iris-setosa
3	5.0	3.6	1.4	Iris-setosa
4	5.4	3.9	1.7	Iris-setosa
...	...	...	...	...
144	6.7	3.0	5.2	Iris-virginica
145	6.3	2.5	5.0	Iris-virginica
146	6.5	3.0	5.2	Iris-virginica
147	6.2	3.4	5.4	Iris-virginica
148	5.9	3.0	5.1	Iris-virginica

149 rows × 4 columns

In [ ]:

Activity: work with the iris dataset

1. 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
2. Are there missing data? If so, create a new dataset containing only the rows with the non-missing data
3. Create a new dataset containing only the petal width and length and the type of Flower
4. Create a new dataset containing only the setal width and length and the type of Flower
5. Create a new dataset containing the setal width and length and the type of Flower encoded as a categorical numerical column

In [171…
```
import pandas as pd

Ruta            = 'datasets'

newUrl = '/iris/iris.csv'

newDatos = pd.read_csv(Ruta+newUrl)
```

In [172…
```
newDatos.columns = ["PetalWidth", "PetalLength", "SepalWidth", "SepalLength", "Type

#Name and type of each column
### newDatos.info()
print(newDatos.columns)
print(newDatos.dtypes)

#Statistical Summary: - Minimun, Maximum, Mean - Average, Standar Deviation.
print(newDatos.describe())
```

```
Index(['PetalWidth', 'PetalLength', 'SepalWidth', 'SepalLength', 'Type'], dtype='o
bject')
PetalWidth     float64
PetalLength    float64
SepalWidth     float64
SepalLength    float64
Type            object
dtype: object
       PetalWidth  PetalLength  SepalWidth  SepalLength
count  149.000000   149.000000  149.000000   149.000000
mean     5.848322     3.054362    3.773826     1.206040
std      0.828594     0.435810    1.760543     0.760354
min      4.300000     2.000000    1.000000     0.100000
25%      5.100000     2.800000    1.600000     0.300000
50%      5.800000     3.000000    4.400000     1.300000
75%      6.400000     3.300000    5.100000     1.800000
max      7.900000     4.400000    6.900000     2.500000
```

In [173…
```
#Median
print("PetalWidth Median: ", newDatos['PetalWidth'].median())
print("PetalLength Median: ", newDatos['PetalLength'].median())
print("SepalWidth Median: ", newDatos['SepalWidth'].median())
print("SepalLength Median: ", newDatos['SepalLength'].median())
```

```
PetalWidth Median:  5.8
PetalLength Median:  3.0
SepalWidth Median:  4.4
SepalLength Median:  1.3
```

In [174…
```
newDatos
```

Out[174]:

	PetalWidth	PetalLength	SepalWidth	SepalLength	Type
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa
...	...	...	...	...	...
144	6.7	3.0	5.2	2.3	Iris-virginica
145	6.3	2.5	5.0	1.9	Iris-virginica
146	6.5	3.0	5.2	2.0	Iris-virginica
147	6.2	3.4	5.4	2.3	Iris-virginica
148	5.9	3.0	5.1	1.8	Iris-virginica

149 rows × 5 columns

In [175…
```
newDF0 = newDatos.dropna() #Dropping missing data
```

In [176…
```
newDF0
```

Out[176]:

	PetalWidth	PetalLength	SepalWidth	SepalLength	Type
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa
...	...	...	...	...	...
144	6.7	3.0	5.2	2.3	Iris-virginica
145	6.3	2.5	5.0	1.9	Iris-virginica
146	6.5	3.0	5.2	2.0	Iris-virginica
147	6.2	3.4	5.4	2.3	Iris-virginica
148	5.9	3.0	5.1	1.8	Iris-virginica

149 rows × 5 columns

In [177…
```
#Create a new dataset containing only the petal width and length and the type of Fl
#Create a new dataset containing only the sepal width and length and the type of Fl

newDF1 = newDF0[["PetalWidth","PetalLength","Type"]]
newDF2 = newDF0[["SepalWidth","SepalLength","Type"]]
```

In [178…
```
newDF1
```

Out[178]:

	PetalWidth	PetalLength	Type
0	4.9	3.0	Iris-setosa
1	4.7	3.2	Iris-setosa
2	4.6	3.1	Iris-setosa
3	5.0	3.6	Iris-setosa
4	5.4	3.9	Iris-setosa
...	...	...	...
144	6.7	3.0	Iris-virginica
145	6.3	2.5	Iris-virginica
146	6.5	3.0	Iris-virginica
147	6.2	3.4	Iris-virginica
148	5.9	3.0	Iris-virginica

149 rows × 3 columns

In [179...

```
newDF2
```

Out[179]:

	SepalWidth	SepalLength	Type
0	1.4	0.2	Iris-setosa
1	1.3	0.2	Iris-setosa
2	1.5	0.2	Iris-setosa
3	1.4	0.2	Iris-setosa
4	1.7	0.4	Iris-setosa
...	...	...	...
144	5.2	2.3	Iris-virginica
145	5.0	1.9	Iris-virginica
146	5.2	2.0	Iris-virginica
147	5.4	2.3	Iris-virginica
148	5.1	1.8	Iris-virginica

149 rows × 3 columns

In [180...

```
#Create a new dataset containing the sepal width and length and the type of Flower

newDF0["NewType"] = newDF0.Type.replace({"Iris-setosa":1 , "Iris-virginica": 2, "Ir

newDF3 = newDF0[["SepalWidth","SepalLength","NewType"]]
```

In [181...

```
newDF3
```

Out[181]:

	SepalWidth	SepalLength	NewType
0	1.4	0.2	1
1	1.3	0.2	1
2	1.5	0.2	1
3	1.4	0.2	1
4	1.7	0.4	1
...	...	...	...
144	5.2	2.3	3
145	5.0	1.9	3
146	5.2	2.0	3
147	5.4	2.3	3
148	5.1	1.8	3

149 rows × 3 columns

In [ ]: