CLEAN DATA IN PANDAS

Cleaning Data

[Clean Data](https://www.w3schools.com/python/pandas/pandas_cleaning.asp)

[Clean Empty Cells](https://www.w3schools.com/python/pandas/pandas_cleaning_empty_cells.asp)

[Clean Wrong Format](https://www.w3schools.com/python/pandas/pandas_cleaning_wrong_format.asp)

[Clean Wrong Data](https://www.w3schools.com/python/pandas/pandas_cleaning_wrong_data.asp)

[Remove Duplicates](https://www.w3schools.com/python/pandas/pandas_cleaning_duplicates.asp)

### LOAD FILES IN DATA FRAME

* import pandas as pd  
    
  df = pd.read\_csv('data.csv')  
    
  print(df.to\_string())

READ CSV FILES

* import pandas as pd  
    
  df = pd.read\_csv('data.csv')  
    
  print(df)

print(pd.\_\_version\_\_)

# Analyzing DataFrames

mport pandas as pd  
  
df = pd.read\_csv('data.csv')  
  
print(df.head())

print(df.tail())

print(df.info())

## Data Cleaning

Data cleaning means fixing bad data in your data set.

Bad data could be:

* Empty cells
* Data in wrong format
* Wrong data
* Duplicates

## Remove Rows

import pandas as pd  
  
df = pd.read\_csv('data.csv')  
  
new\_df = df.dropna()  
  
print(new\_df.to\_string())

DROP NULL VALUES BY REPLACING IT TRUE

import pandas as pd  
  
df = pd.read\_csv('data.csv')  
  
df.dropna(inplace = True)  
  
print(df.to\_string())

# Cleaning Data of Wrong Format

import pandas as pd  
  
df = pd.read\_csv('data.csv')  
  
df['Date'] = pd.to\_datetime(df['Date'])  
  
print(df.to\_string())

## Removing Rows

df.dropna(subset=['Date'], inplace = True)

# Fixing Wrong Data

## Replacing Values

df.loc[7, 'Duration'] = 45

# Removing Duplicates

Returns True for every row that is a duplicate, othwerwise False:

print(df.duplicated())

## Removing Duplicates

To remove duplicates, use the drop\_duplicates() method.

df.drop\_duplicates(inplace = True)

# Data Correlations

## Finding Relationships

A great aspect of the Pandas module is the corr() method.

The corr() method calculates the relationship between each column in your data set.

The examples in this page uses a CSV file called: 'data.csv'.

df.corr()

## Plotting

Pandas uses the plot() method to create diagrams.

We can use Pyplot, a submodule of the Matplotlib library to visualize the diagram on the screen.

Read more about Matplotlib in our [Matplotlib Tutorial](https://www.w3schools.com/python/matplotlib_intro.asp).

import pandas as pd  
import matplotlib.pyplot as plt  
  
df = pd.read\_csv('data.csv')  
  
df.plot()  
  
plt.show()

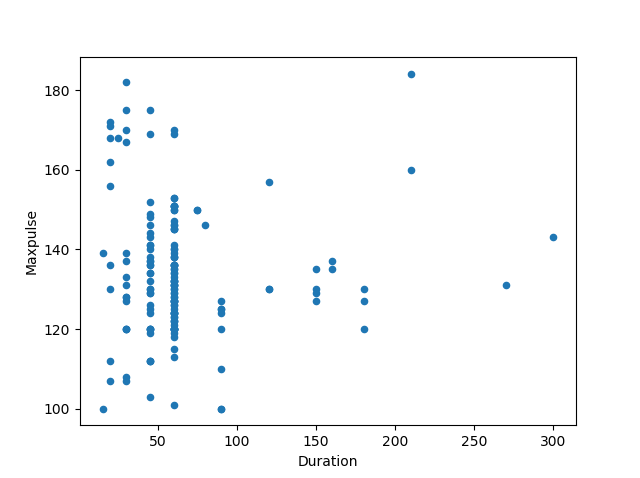
## Scatter Plot

mport pandas as pd  
import matplotlib.pyplot as plt  
  
df = pd.read\_csv('data.csv')  
  
df.plot(kind = 'scatter', x = 'Duration', y = 'Calories')  
  
plt.show()

### Example

import pandas as pd  
import matplotlib.pyplot as plt  
  
df = pd.read\_csv('data.csv')  
  
df.plot(kind = 'scatter', x = 'Duration', y = 'Maxpulse')  
  
plt.show()

### Result



**CONVERT DATA INTO SPECIFIED DTYPE**

|  |  |
| --- | --- |
| [astype()](https://www.w3schools.com/python/pandas/ref_df_astype.asp) | Convert the DataFrame into a specified dtype |

f.month=f.month.astype("int64")

# **replace space with underscore in column names pandas**

f.columns = f.columns.str.replace(' ','\_')

**DROP UNNAMED COLUMNS**

f.drop(columns="Unnamed:\_0",inplace=True)

**REMOVING DOLLAR SIGNS**

f.dtypes

f['Sales'] = f['Sales'].replace({'\$': '', ',': ''}, regex=**True**).astype(float)

* f['Income\_Commission\_Fees'] = f['Income\_Commission\_Fees'].replace({'\$': '', ',': ''}, regex=True).astype(float)

**SAVING FILE INTO CSV**

f.to\_csv('file2.csv')