Introduction to Pandas in Python

Estimated time needed: **15** minutes

Objectives

After completing this lab you will be able to:

- Use Pandas to access and view data

The table has one row for each album and several columns.

You can see the dataset here:

Artist Album Released Length Genre Music recording sales (millions) Claimed sales (millions) Released Soundtrack Rating (friends) Michael Jackson Thriller 1982 00:42:19 Pop, rock, R&B 46 65 30-Nov-82 10.0 AC/DC Back in Black 1980 00:42:11 Hard rock 26.1 50 25-Jul-80 8.5 Pink Floyd The Dark Side of the Moon 1973 00:42:49 Progressive rock 24.2 45 01-Mar-73 9.5 Whitney Houston The Bodyguard 1992 00:57:44 Soundtrack/R&B, soul, pop 26.1 50 25-Jul-80 Y 7.0 Meat Loaf Bat Out of Hell 1977 00:46:33 Hard rock, progressive rock 20.6 43 21-Oct-77 7.0 Eagles Their Greatest Hits (1971-1975) 1976 00:43:08 Rock, soft rock, folk rock 32.2 42 17-Feb-76 9.5 Bee Gees Saturday Night Fever 1977 1:15:54 Disco 20.6 40 15-Nov-77 Y 9.0 Fleetwood Mac Rumours 1977 00:40:01 Soft rock 27.9 40 04-Feb-77 9.5

```
# Dependency needed to install file

# If running the notebook on your machine, else leave it commented
#!pip install xlrd

#!pip install openpyxl
import piplite
await piplite.install(['xlrd','openpyxl'])

# Import required library

import pandas as pd
```

After the import command, we now have access to a large number of pre-built classes and functions. This assumes the library is installed; in our lab environment all the necessary libraries are installed. One way pandas allows you to work with data is a dataframe. Let's go through the process to go from a comma separated values (.csv) file to a dataframe. This variable csv_path stores the path of the .csv, that is used as an argument to the read_csv function. The result is stored in the object df, this is a common short form used for a variable referring to a Pandas dataframe.

```
# Read data from CSV file

# csv_path = 'https://cf-courses-data.s3.us.cloud-object-
```

```
storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-
SkillsNetwork/labs/Module%204/data/TopSellingAlbums.csv'
# df = pd.read_csv(csv_path)

from pyodide.http import pyfetch
import pandas as pd

filename = "https://cf-courses-data.s3.us.cloud-object-
storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-
SkillsNetwork/labs/Module%204/data/TopSellingAlbums.csv"

async def download(url, filename):
    response = await pyfetch(url)
    if response.status == 200:
        with open(filename, "wb") as f:
            f.write(await response.bytes())


await download(filename, "TopSellingAlbums.csv")
df = pd.read_csv("TopSellingAlbums.csv")
```

We can use the method head() to examine the first five rows of a dataframe:

```
# Print first five rows of the dataframe

df.head()
```

We use the path of the excel file and the function read_excel. The result is a data frame as before:

```
# Read data from Excel File and print the first five rows

xlsx_path = 'https://cf-courses-data.s3.us.cloud-object-
storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-
SkillsNetwork/jupyterlite/files/Module%205/data/TopSellingAlbums.xlsx'

await download(xlsx_path, "TopSellingAlbums.xlsx")
df = pd.read_excel("TopSellingAlbums.xlsx")
df.head()
```

We can access the column Length and assign it a new dataframe x:

```
# Access to the column Length

x = df[['Length']]
x
```

The process is shown in the figure:

You can also get a column as a series. You can think of a Pandas series as a 1-D dataframe. Just use one bracket:

```
# Get the column as a series

x = df['Length']
x
```

You can also get a column as a dataframe. For example, we can assign the column Artist:

```
# Get the column as a dataframe

x = df[['Artist']]
type(x)
```

You can do the same thing for multiple columns; we just put the dataframe name, in this case, df, and the name of the multiple column headers enclosed in double brackets. The result is a new dataframe comprised of the specified columns:

```
# Access to multiple columns

y = df[['Artist','Length','Genre']]
y
```

The process is shown in the figure:

One way to access unique elements is the iloc method, where you can access the 1st row and the 1st column as follows:

```
# Access the value on the first row and the first column

df.iloc[0, 0]
```

You can access the 2nd row and the 1st column as follows:

```
# Access the value on the second row and the first column

df.iloc[1,0]
```

You can access the 1st row and the 3rd column as follows:

```
# Access the value on the first row and the third column

df.iloc[0,2]

# Access the value on the second row and the third column
df.iloc[1,2]
```

This is shown in the following image

You can access the column using the name as well, the following are the same as above:

```
# Access the column using the name
df.loc[1, 'Artist']
# Access the column using the name
df.loc[1, 'Artist']
# Access the column using the name
df.loc[0, 'Released']
# Access the column using the name
df.loc[1, 'Released']
```

You can perform slicing using both the index and the name of the column:

```
# Slicing the dataframe
df.iloc[0:2, 0:3]
```

```
# Slicing the dataframe using name
df.loc[0:2, 'Artist':'Released']
```

Use a variable q to store the column Rating as a dataframe

```
# Write your code below and press Shift+Enter to execute
```

Assign the variable q to the dataframe that is made up of the column Released and Artist:

```
# Write your code below and press Shift+Enter to execute
```

Access the 2nd row and the 3rd column of df:

```
# Write your code below and press Shift+Enter to execute
```

Use the following list to convert the dataframe index df to characters and assign it to df_new; find the element corresponding to the row index a and column 'Artist'. Then select the rows a through d for the column 'Artist'

```
new_index=['a','b','c','d','e','f','g','h']
```

Congratulations, you have completed your first lesson and hands-on lab in Python.

Authors:

Joseph Santarcangelo

Joseph Santarcangelo has a PhD in Electrical Engineering, his research focused on using machine learning, signal processing, and computer vision to determine how videos impact human cognition. Joseph has been working for IBM since he completed his PhD.

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2022-01-10	2.1	Malika	Removed the readme for GitShare
2020-08-26	2.0	Lavanya	Moved lab to course repo in GitLab
2020-11-24	3.0	Nayef	Added new images