Python

Introduction

Edgar Rios Linares

Python

Features

Powerful

Easy to learn

Efficient high-level data structures

Object-oriented programming

Elegant syntax

Interpreted nature

Extensive collection of free libraries

Rapid application development

Most platforms.

Python

Business with Python

- Google
- Facebook
- Instagram
- Spotify
- Quora
- Netflix
- Dropbox
- Reddit

Python

Platfforms

https://www.zepl.com/product/

https://docs.aws.amazon.com/sagemaker/latest/dg/notebooks.html

https://observablehq.com

https://deepnote.com

https://www.ibm.com/cloud/watson-studio

https://codeocean.com

https://www.kaggle.com/code

https://visualstudio.microsoft.com/es/

https://cloud.google.com/datalab/docs/how-to/working-with-notebooks/

https://mybinder.org

https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-managed-notebooks.html

https://colab.research.google.com

https://gradient.paperspace.com

https://databricks.com/product/collaborative-notebooks

Python

Learning

Kaggle Learn Courses

Cloud certification programs (AWS, Azure, GCP)

Fast.ai

Datacamp

Udacity

LinkedIn

edX

Coursera

University Courses

Udemy

Python

IDE

- PyScripter
- PyCharm
- Spyder
- Pydev
- Idle
- Wing
- Sublime Text
- Visual Studio Code
- Anaconda

Python

Python vs Jupyter Notebook

Python is contained in a .pyt file

it only contains code to be executed.

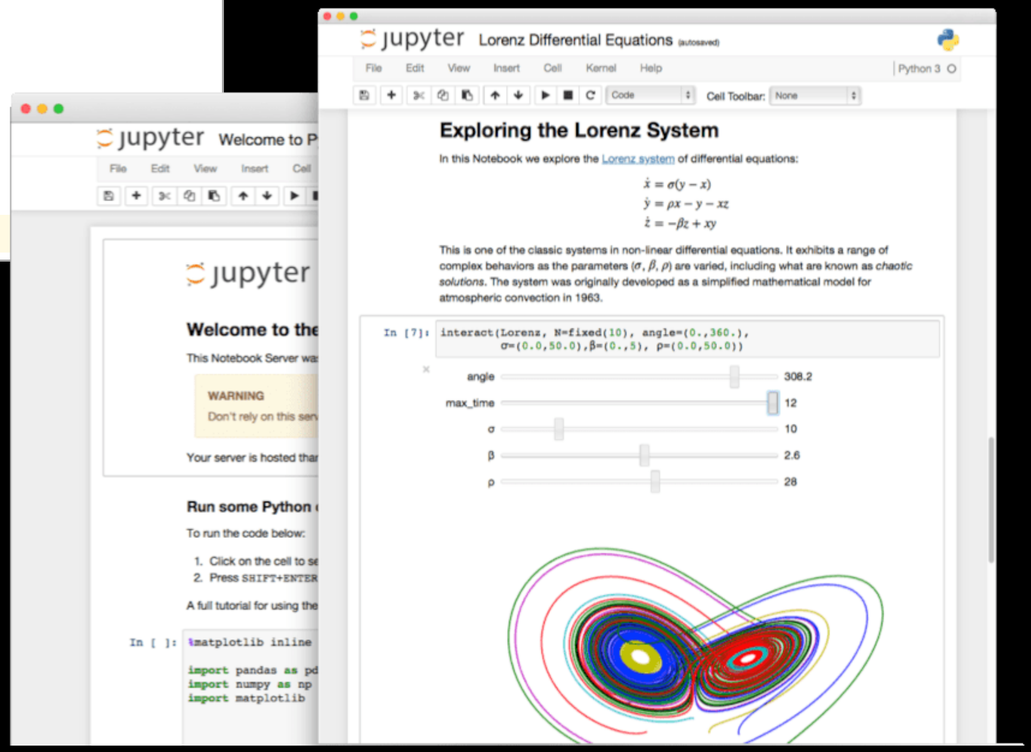
Jupyter Notebook is an open source web application that you can use to create and share documents containing live code, equations, visualizations, and text.

Python

Python vs Jupyter Notebook

test.py

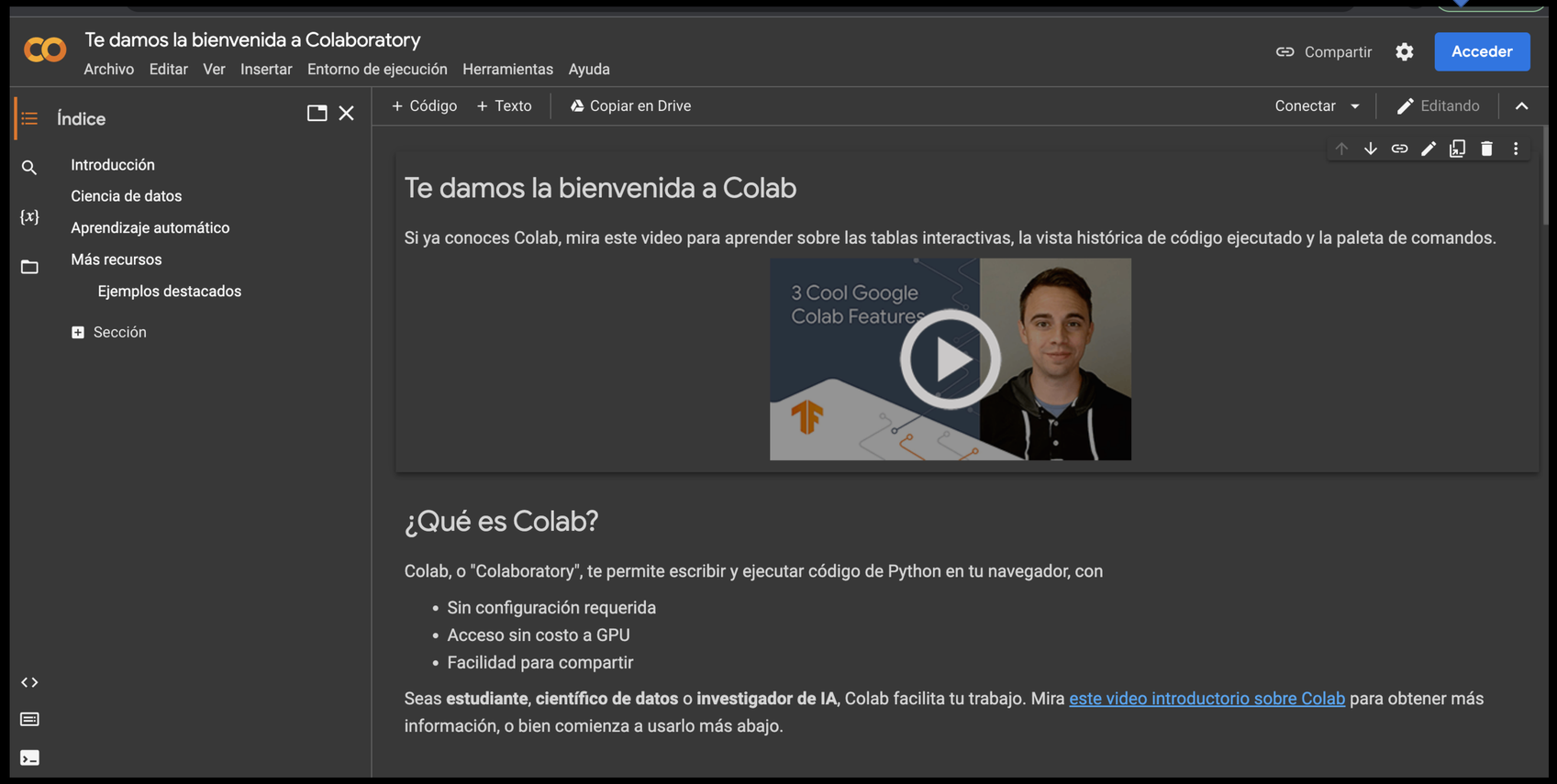
```
1  my_file = open("C:/Documents/Python/binaryf.dll", "wb+")
2  message = "Hello Python"
3  file_encode = message.encode("ASCII")
4  my_file.write(file_encode)
5  my_file.seek(0)
6  bdata = my_file.read()
7  print("Binary Data:", bdata)
8  ntext = bdata.decode("ASCII")
9  print("Normal data:", ntext)
10
```



Python

Create aacount in Colab

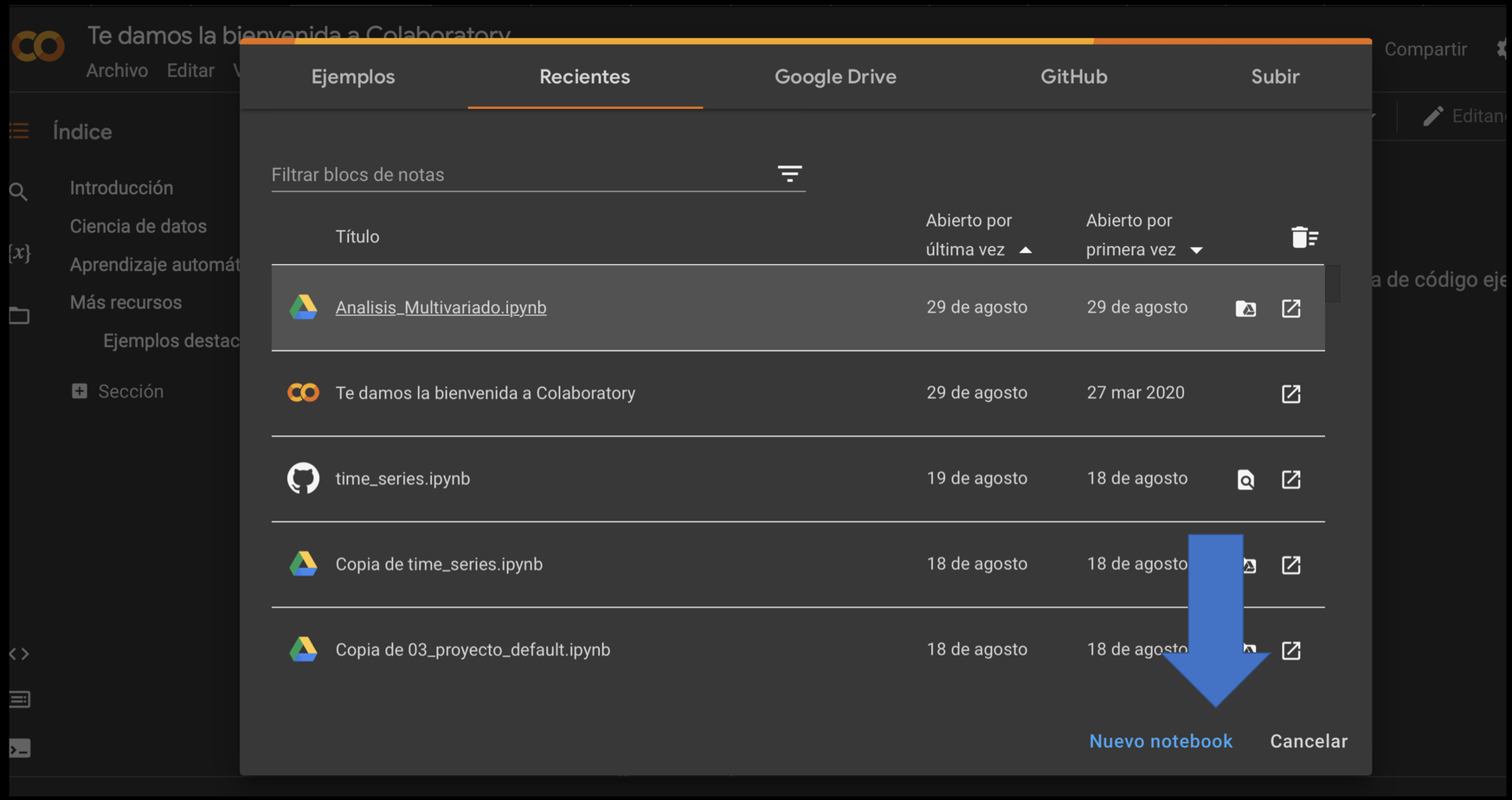
Go https://colab.research.google.com/
Use Gmail account



Python

Creación de cuenta en Colab

New **notebook**



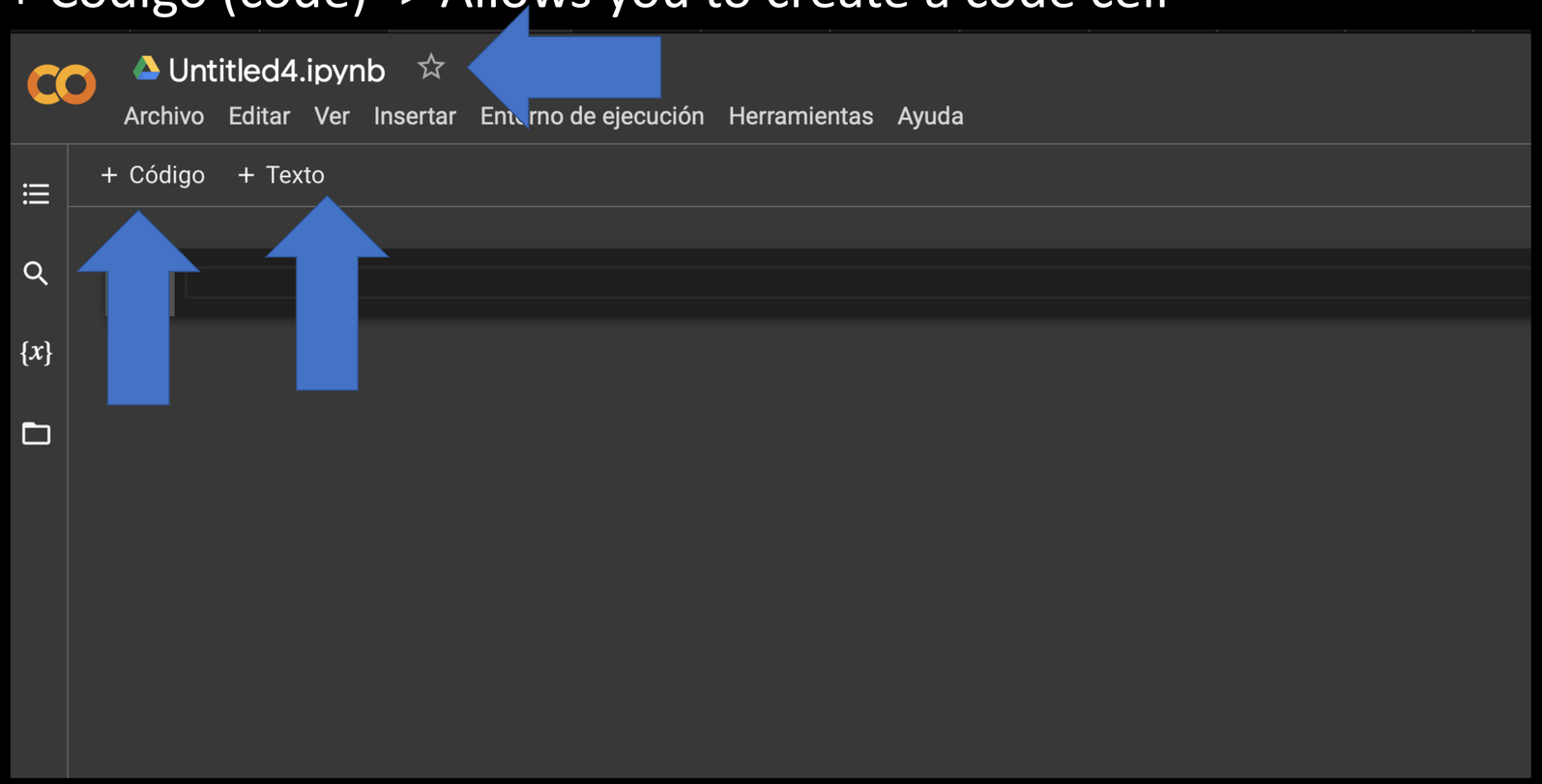
Python

Entorno

Name of notebook

+ Texto (text) -> Allows you to create a text cell

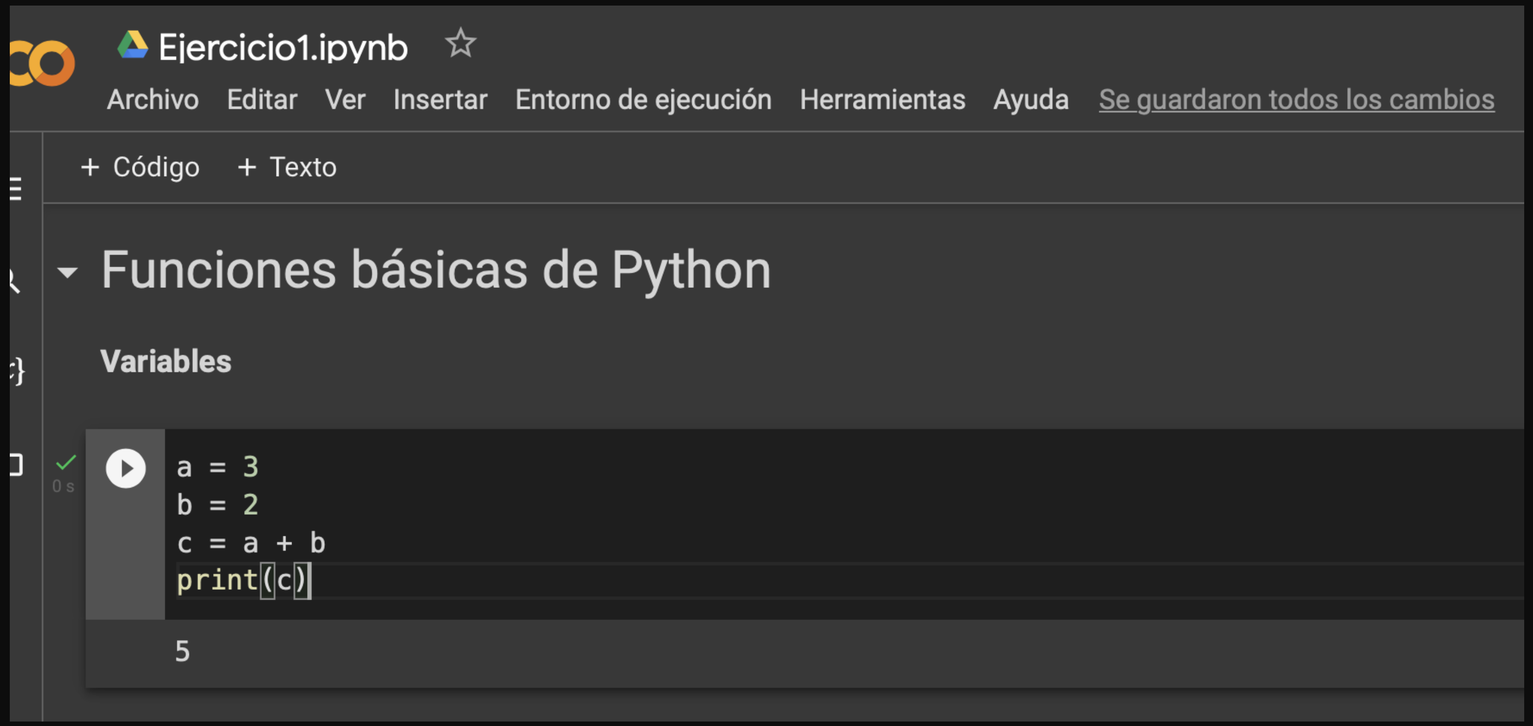
+ Código (code) -> Allows you to create a code cell



Python

Variables

Create a code cell and create variables a and b, assign them a numeric value and add both variables assigning the result to variable c, use the **print()** function to display the result



Python

Libraries

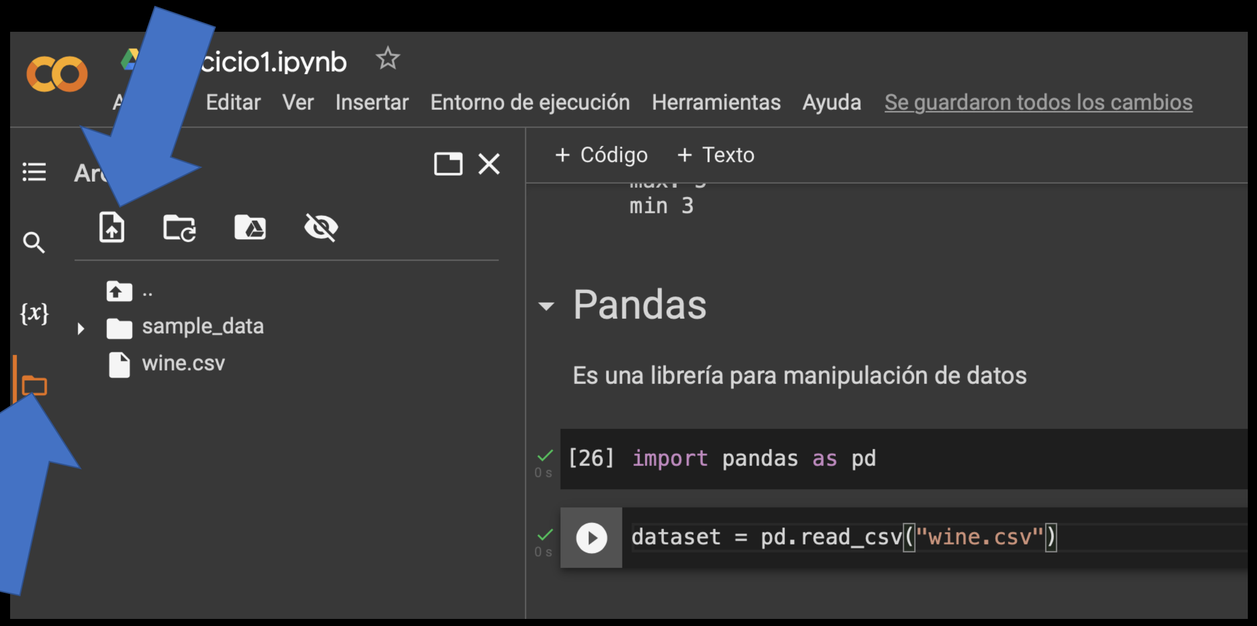
Create a code cell, import the numpy library, create two arrays b and c and multiply them and display the result.

```
[5]  import numpy as np

     a = np.arange(15) #Return evenly spaced values within a given interval
     print(a)

     [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14]

[7]  c = a.argmax()
     print(c)

     14

     b = np.array([3, 4, 5])
     c = np.array([2, 2, 2])
     print(b*c)

     [ 6  8 10]
```

Python

Pandas

Load the wine.csv file into your Colab environment.
Import the pandas library using the alias pd, and using the read_csv method, read the wine.csv file



Python

Pandas

Display the first 5 records of the dataset and the last 5. Use the head() and tail() methods

Ejercicio1.ipynb

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

```
dataset.head(5)
```

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735

```
dataset.tail(5)
```

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
173	3	13.71	5.65	2.45	20.5	95	1.68	0.61	0.52	1.06	7.7	0.64	1.74	740
174	3	13.40	3.91	2.48	23.0	102	1.80	0.75	0.43	1.41	7.3	0.70	1.56	750
175	3	13.27	4.28	2.26	20.0	120	1.59	0.69	0.43	1.35	10.2	0.59	1.56	835
176	3	13.17	2.59	2.37	20.0	120	1.65	0.68	0.53	1.46	9.3	0.60	1.62	840

Python

Pandas

Use the info() method to display the number of records, number of variables, and data type of each variable

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178 entries, 0 to 177
Data columns (total 14 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   V1      178 non-null    int64  
 1   V2      178 non-null    float64
 2   V3      178 non-null    float64
 3   V4      178 non-null    float64
 4   V5      178 non-null    float64
 5   V6      178 non-null    int64  
 6   V7      178 non-null    float64
 7   V8      178 non-null    float64
 8   V9      178 non-null    float64
 9   V10     178 non-null    float64
 10  V11     178 non-null    float64
 11  V12     178 non-null    float64
 12  V13     178 non-null    float64
 13  V14     178 non-null    int64  
dtypes: float64(11), int64(3)
memory usage: 19.6 KB
```

Python

Pandas

Use the describe() method to display the statistical description of each variable (column)

```
dataset.describe()
```

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12
count	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000
mean	1.938202	13.000618	2.336348	2.366517	19.494944	99.741573	2.295112	2.029270	0.361854	1.590899	5.058090	0.957449
std	0.775035	0.811827	1.117146	0.274344	3.339564	14.282484	0.625851	0.998859	0.124453	0.572359	2.318286	0.228572
min	1.000000	11.030000	0.740000	1.360000	10.600000	70.000000	0.980000	0.340000	0.130000	0.410000	1.280000	0.480000
25%	1.000000	12.362500	1.602500	2.210000	17.200000	88.000000	1.742500	1.205000	0.270000	1.250000	3.220000	0.782500
50%	2.000000	13.050000	1.865000	2.360000	19.500000	98.000000	2.355000	2.135000	0.340000	1.555000	4.690000	0.965000
75%	3.000000	13.677500	3.082500	2.557500	21.500000	107.000000	2.800000	2.875000	0.437500	1.950000	6.200000	1.120000
max	3.000000	14.830000	5.800000	3.230000	30.000000	162.000000	3.880000	5.080000	0.660000	3.580000	13.000000	1.710000

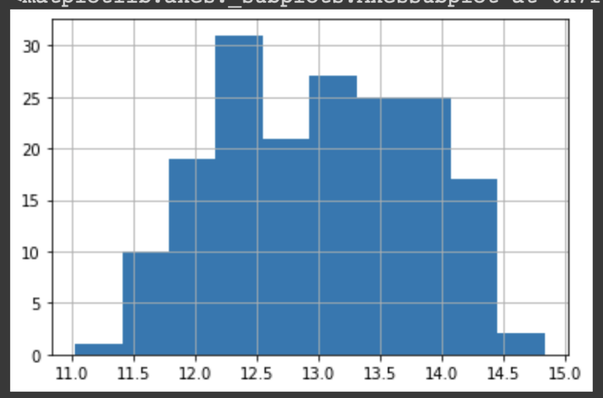
Python

Pandas

Use the hist() method to display the distribution of each variable.

```
dataset.V2.hist()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ed840db50>
```



Resources

- https://www.python.org/doc/
- https://realpython.com/world-class-companies-using-python/
- https://www.anaconda.com
  - https://jupyter.org
  - https://numpy.org
  - https://pandas.pydata.org

¿Was it helpful to you??

¡ Save it!

¡Share it!

¡Follow me for more!

Edgar Rios Linares

📍Linkedin —> https://www.linkedin.com/in/erlinares/

👋 Follow us on 𝕏: https://x.com/erlinares

💻 GitHub: https://github.com/erlinares/365_AI_Journey/

💬 Discord: https://discord.gg/5fFM2zh8