

Python



```
199
200 alpha = 40
201 m = load_npz('sparse_user_item.npz')
202 n_users, n_movies = m.shape
203
204 ratings = [alpha for i in range(len(user_ratings))]
205
206 m.data = np.hstack((m.data, ratings))
207 m.indices = np.hstack((m.indices, user_ratings))
208 m.indptr = np.hstack((m.indptr, len(m.data)))
209 m._shape = (n_users+1, n_movies)
210
211 # recommend N items to new user
212 with open('model.sav', 'rb') as pickle_in:
213     model = pickle.load(pickle_in)
214 recommended, _ = zip(*model.recommend(n_users, m, recalculate_user=True))
```

← variable unpacking

← list comprehension



Python

Lenguaje de programación

Potente

Fácil de aprender

Estructuras de datos de alto nivel eficientes

Programación orientada a objetos

Sintaxis elegante

Naturaleza interpretada

Extensa colección de bibliotecas gratuitas

Desarrollo rápido de aplicaciones

Mayoría de las plataformas.





Python

Empresas que usan Python

Google

Facebook

Instagram

Spotify

Quora

Netflix

Dropbox

Reddit



NETFLIX





Python

Plataformas

<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

Plataformas Cursos

Kaggle Learn Courses

Cloud certification programs (AWS, Azure, Google CP)

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 está contenido en un archivo .pyt solo contiene código para ser ejecutado.

Jupyter Notebook es una aplicación web de código abierto que puede usar para crear y compartir documentos que contienen código en vivo, ecuaciones, visualizaciones y texto.

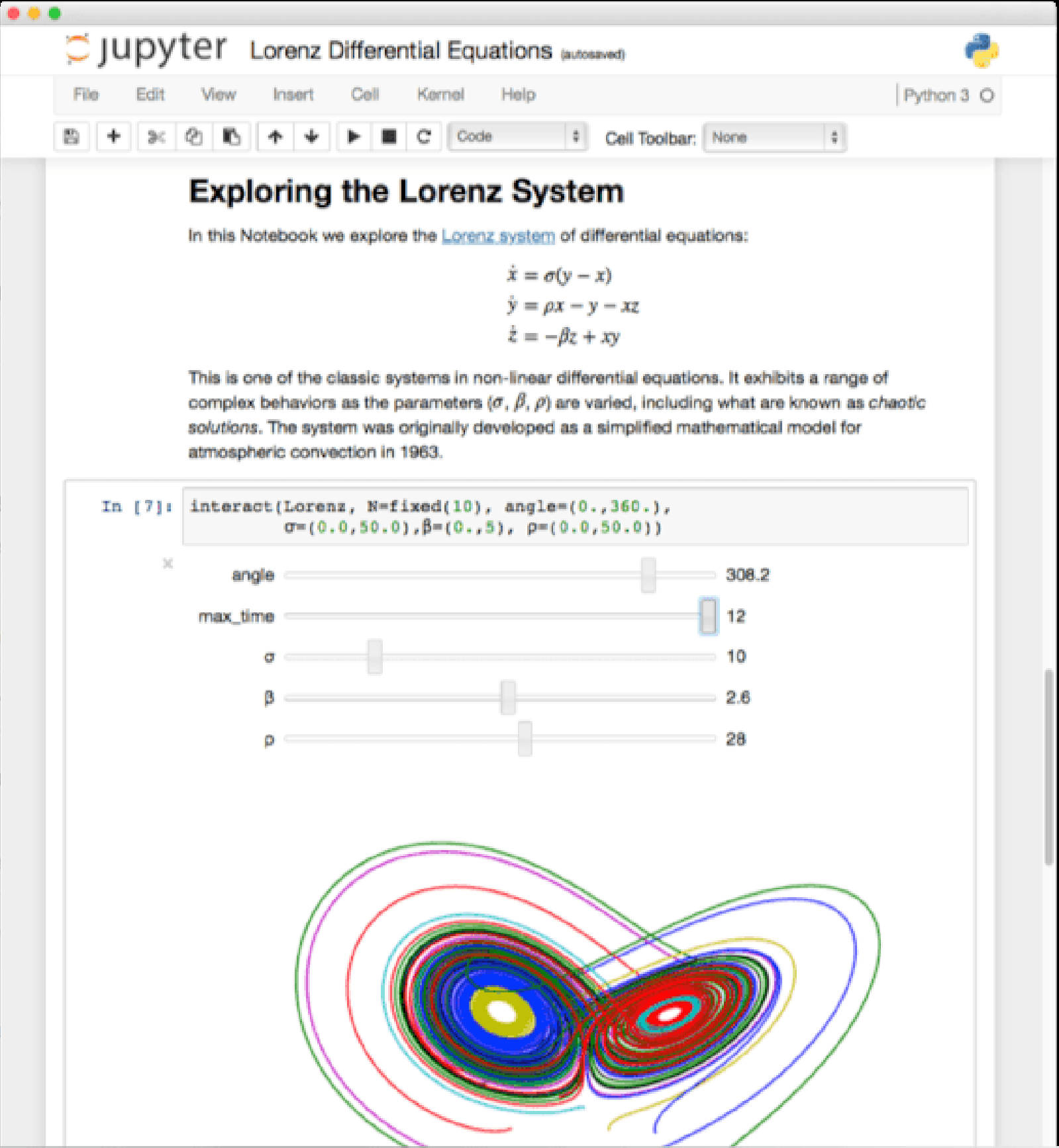
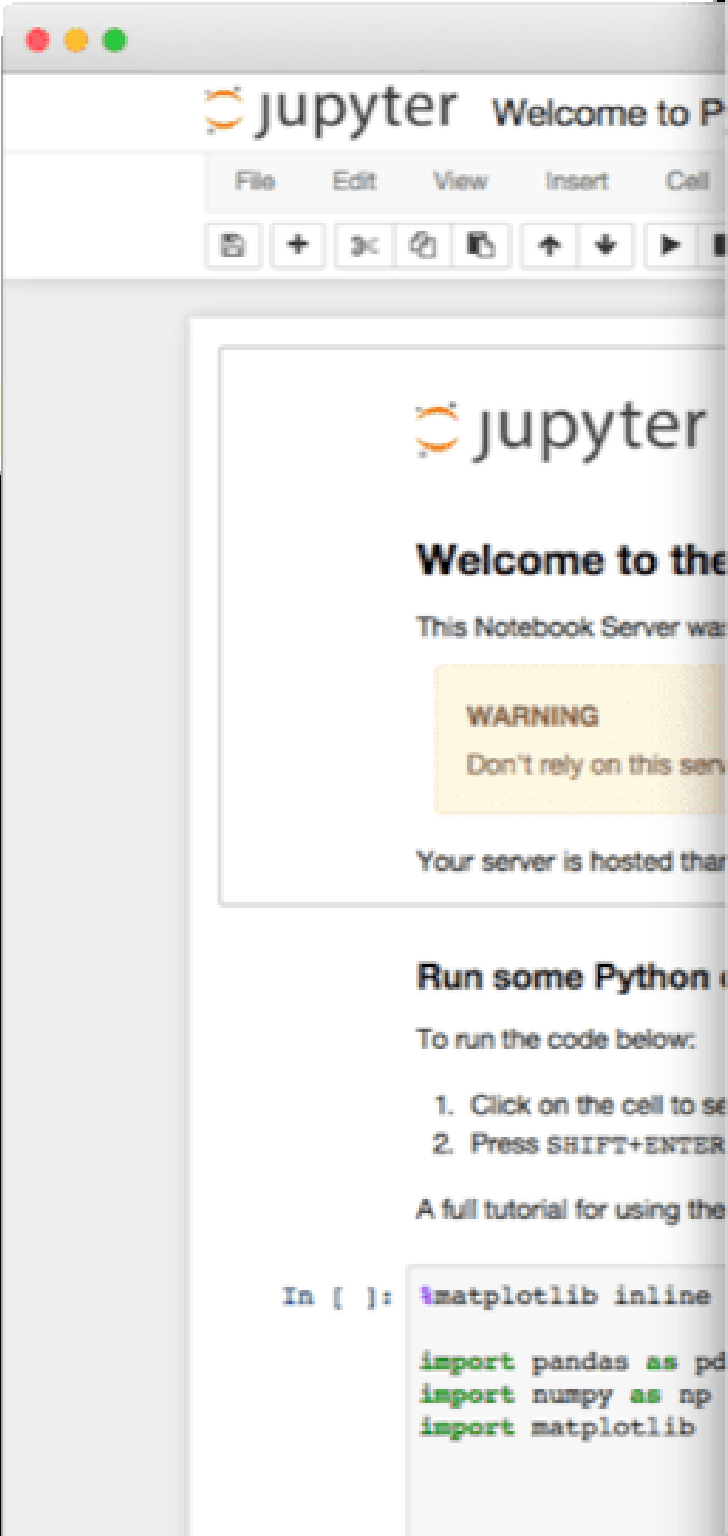




Python

Python vs Jupyter Notebook

```
test.py x
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
```

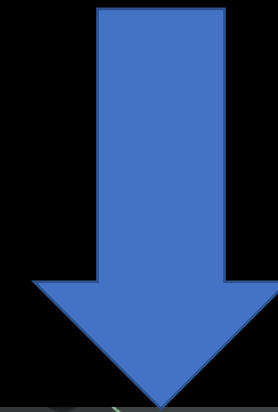




Python

Creación de cuenta en Colab

Ingresar a <https://colab.research.google.com/>
Autenticarse con su cuenta de Gmail.



Te damos la bienvenida a Colaboratory

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda

Compartir Acceder

Índice

- Introducción
- Ciencia de datos
- Aprendizaje automático
- Más recursos
 - Ejemplos destacados
- Sección

Te damos la bienvenida a Colab

Si ya conoces Colab, mira este video para aprender sobre las tablas interactivas, la vista histórica de código ejecutado y la paleta de comandos.

3 Cool Google Colab Features

¿Qué es Colab?

Colab, o "Colaboratory", te permite escribir y ejecutar código de Python en tu navegador, con

- Sin configuración requerida
- Acceso sin costo a GPU
- Facilidad para compartir

Seas **estudiante, científico de datos o investigador de IA**, Colab facilita tu trabajo. Mira [este video introductorio sobre Colab](#) para obtener más información, o bien comienza a usarlo más abajo.





Python

Creación de cuenta en Colab

Crear un notebook nuevo

Te damos la bienvenida a Colaboratory

Archivo Editar V

Ejemplos Recientes Google Drive GitHub Subir

Índice

Introducción

Ciencia de datos

Aprendizaje automático

Más recursos















Ejemplos destacados


Sección

Compartir

Editar

Filtrar blocs de notas

| Título | Abierto por última vez ▲ | Abierto por primera vez ▼ | |
|---|--------------------------|---------------------------|---|
|  Análisis_Multivariado.ipynb | 29 de agosto | 29 de agosto |   |
|  Te damos la bienvenida a Colaboratory | 29 de agosto | 27 mar 2020 |  |
|  time_series.ipynb | 19 de agosto | 18 de agosto |   |
|  Copia de time_series.ipynb | 18 de agosto | 18 de agosto |   |
|  Copia de 03_proyecto_default.ipynb | 18 de agosto | 18 de agosto |   |



[Nuevo notebook](#) Cancelar





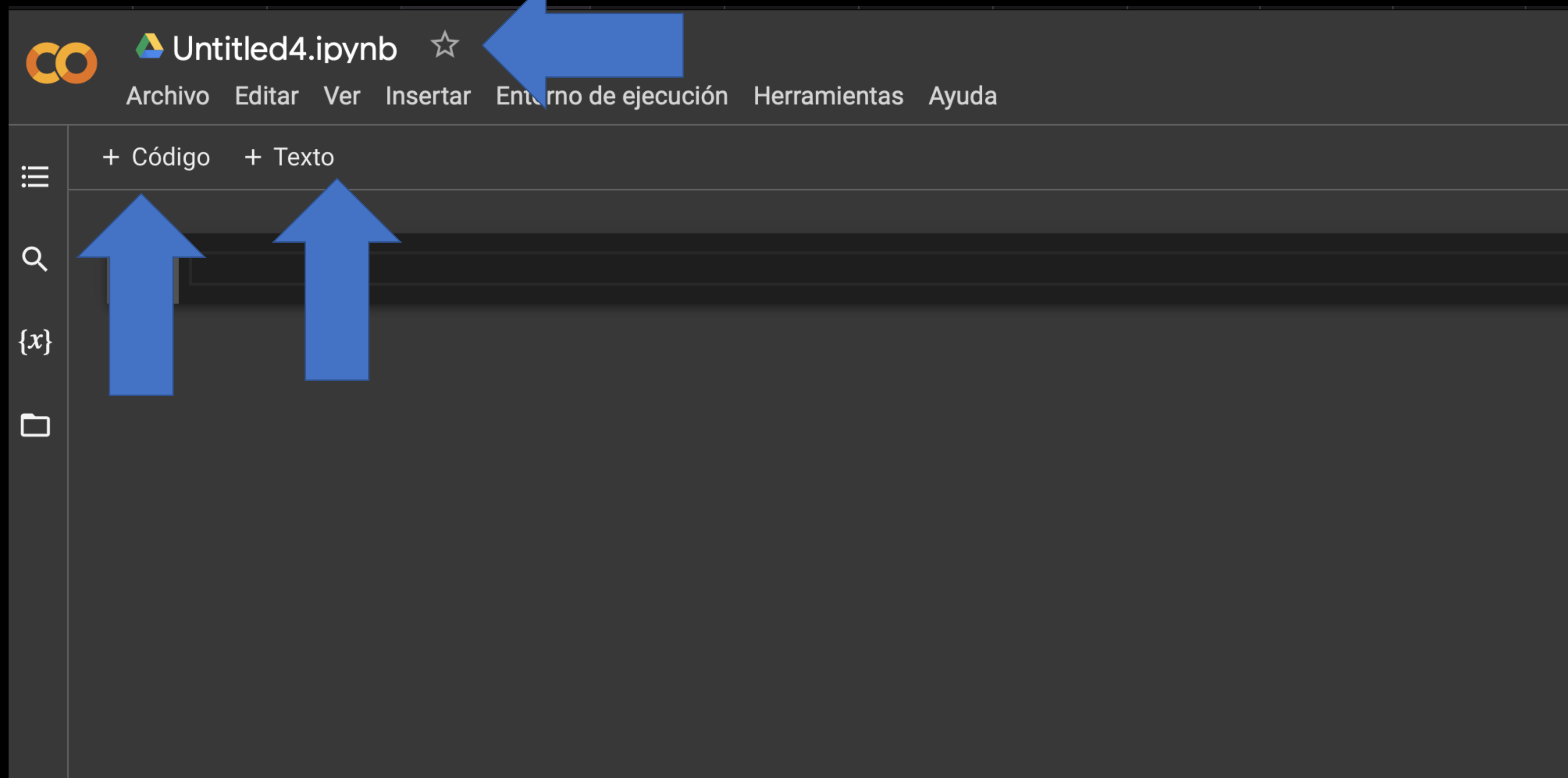
Python

Entorno

Asignar nombre al notebook

+ Texto -> Permite crear una celda de texto

+ Código -> Permite crear una celda de código





Python

Variables

Crear una celda de código y crear las variables a y b, asignarles un valor numérico y sumar ambas variables asignando el resultado en la variable c, usar la función **print()** para mostrar el resultado

The screenshot shows a Jupyter Notebook titled "Ejercicio1.ipynb". The interface includes a menu bar with options: Archivo, Editar, Ver, Insertar, Entorno de ejecución, Herramientas, Ayuda, and a status message "Se guardaron todos los cambios". Below the menu bar, there are tabs for "+ Código" and "+ Texto". The main content area is titled "Funciones básicas de Python" and contains a section labeled "Variables". A code cell is shown with the following code:

```
a = 3
b = 2
c = a + b
print(c)
```

The output of the code cell is the number 5.





Python

Librerías


Crear una celda de código importe la librería numpy, cree dos arreglos b y c y multiplíquelos y muestre el resultado.


```
✓ [5] import numpy as np  
0 s  
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()  
0 s  
print(c)
```

```
14
```

```
✓ [8]  b = np.array([3, 4, 5])  
0 s  
c = np.array([2, 2, 2])  
print(b*c)
```

```
 [ 6  8 10]
```





Python

Pandas

Cargue el archivo wine.csv dentro de su entorno de Colab.
Importe la librería pandas usando el alias pd, y usando el método `read_csv`, lea el archivo wine.csv

The screenshot shows the Google Colab interface. On the left, the file manager sidebar is open, displaying a folder named 'sample_data' and a file named 'wine.csv'. A blue arrow points to the 'upload' icon (a square with an upward arrow) in the sidebar. Another blue arrow points to the 'wine.csv' file. The main area shows the code editor with two lines of code: `import pandas as pd` and `dataset = pd.read_csv("wine.csv")`. The code is executed successfully, indicated by green checkmarks and '0 s' execution time. The Pandas library documentation is visible in the background.





Python

Pandas

Muestre los primeros 5 registros del dataset y los últimos 5. Use los métodos head() y tail()

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 el método `info()` para mostrar el número de registros, número de variables y tipo de dato de cada 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 el método describe() para mostrar la descripción estadística de cada variable (columna)

| | 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 |

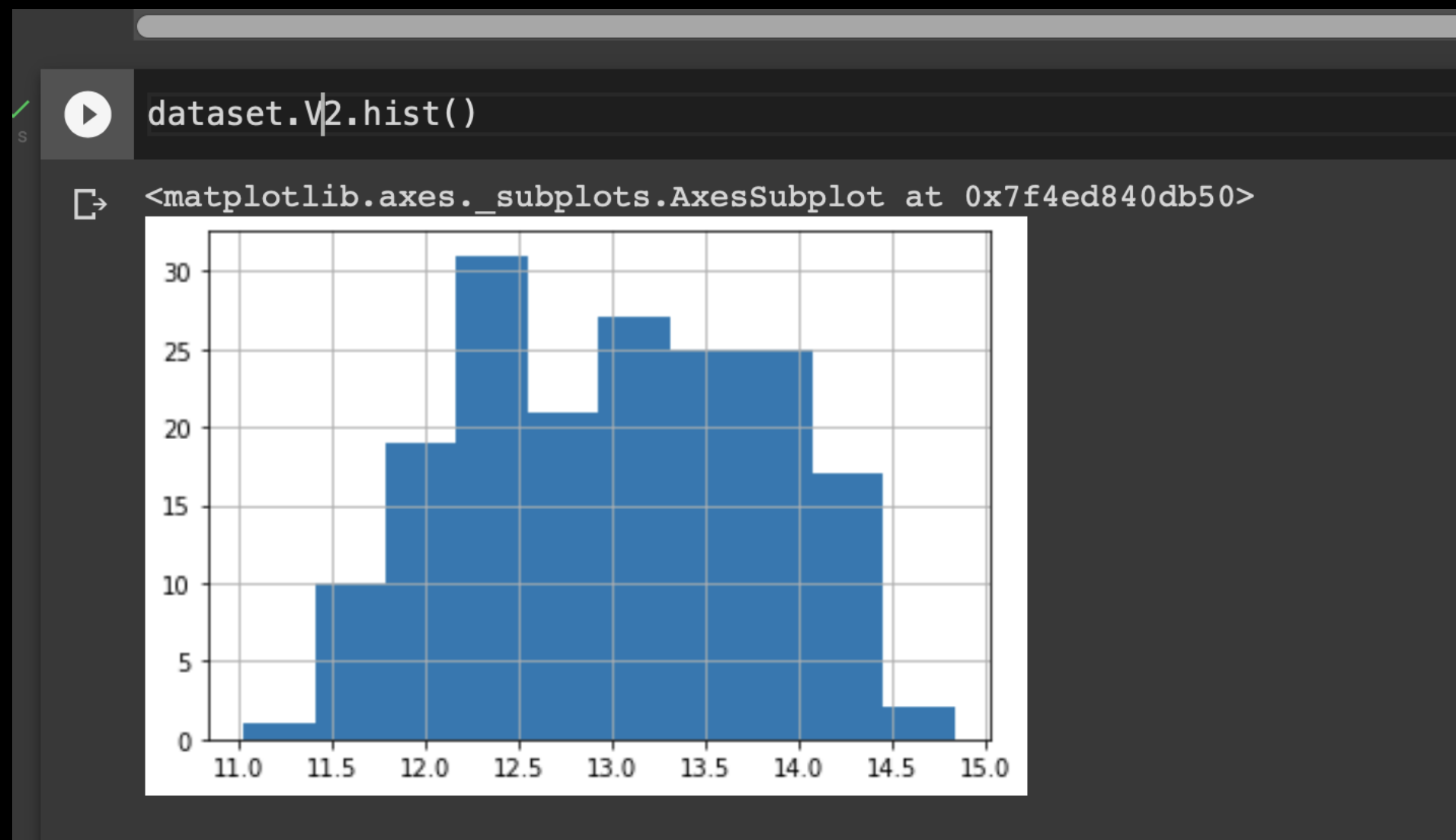




Python

Pandas

Use el método `hist()` para mostrar la distribución de cada variable.





- Dentro de Colab
- <https://colab.research.google.com>

```
3 require File.expand_path("../..", __FILE__)
4 # Prevent database truncation if the environment is production
5 abort("The Rails environment is running in production")
6 require 'spec_helper'
7 require 'rspec/rails'
8
9 require 'capybara/rspec'
10 require 'capybara/rails'
11
12 Capybara.javascript_driver = :webkit
13 Category.delete_all; Category.create(:name => "Category")
14 Shoulda::Matchers.configure do |config|
15   config.integrate do |integrate|
16     with.test_framework :rspec
17     with.library :rails
18   end
19 end
20
21 # Add additional requires below this line
22
23 # Requires supporting ruby files with spec/
24 # spec/support/ and its subdirectories
25 # run as spec files by default. The
26 # in _spec.rb will both be required
27 # run twice. It is recommended to
28 # end with _spec.rb. You can also
29 # add an option on the command line to
30 # run the spec files with the --no-require
31
32 No results found for 'mongoid'
```




Bibliográficos

- <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>