

# Curso de Verano: Complejidad Social y Modelos Computacionales

## Lección Py.1: Instalación de Python y uso de interfaces

Impartido por: Gonzalo Castañeda

Basado en: McKinney, Wes. 2018. "Python for Data Analysis. Data Wrangling with Pandas, NumPy, and IPython", 2a edición, California USA: O'Reilly Media, Inc. Cap. 1 – 2.2

### (1) Python: antecedentes y beneficios

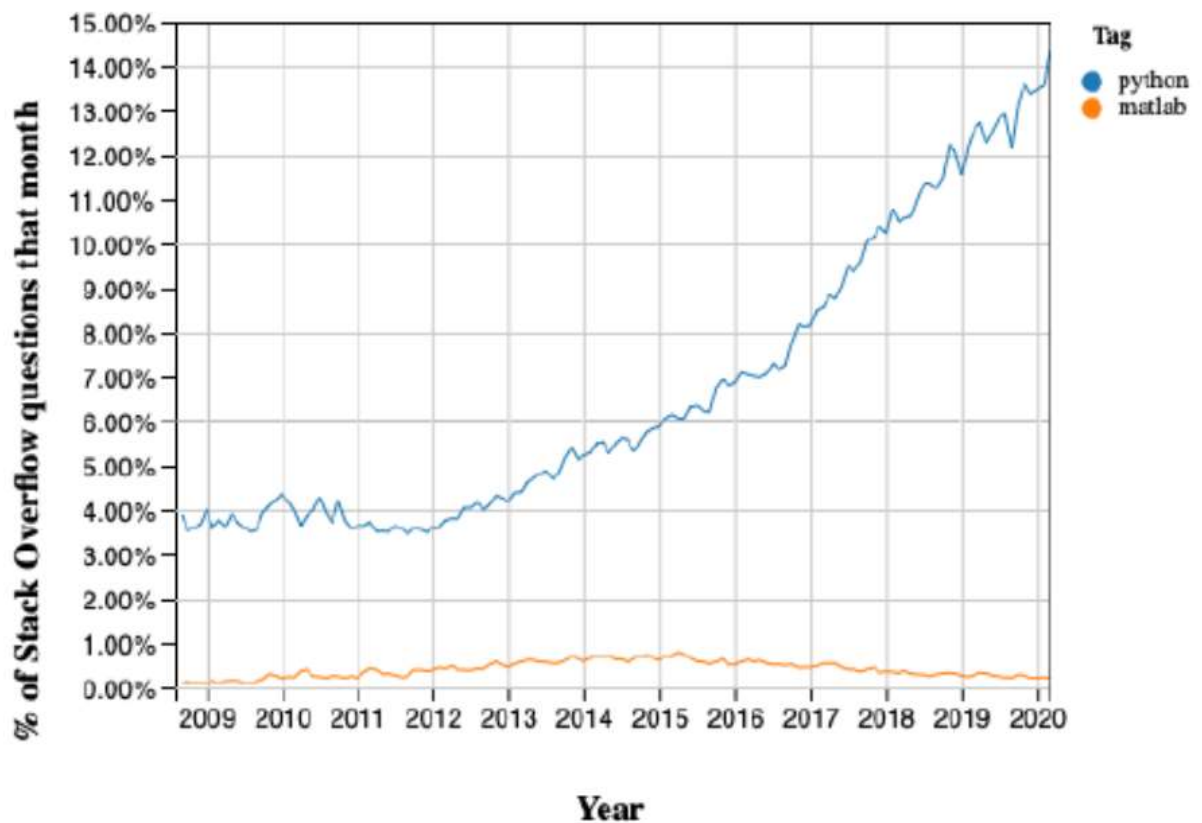
In [1]:

```
# Python es un lenguaje interpretador de programación concebido en 1989 por Guido Van R
# y que en la última década se ha vuelto muy popular
# Este tipo de programas pueden ejecutar las instrucciones conforme son leídas, por lo
# convenientes para la interacción con el usuario
# Al ser un lenguaje de 'alto nivel' es relativamente amigable y requiere menos tiempo
# programación, aunque el tiempo de ejecución sea más lento versus Java o C++
# Su popularidad también se debe a que cuenta ya con bastante ecosistema de librerías que
# un gran potencial analítico
# Aunque se trata de un software de propósito general (vs Stata, Matlab, R) es muy conv
# para el análisis de datos
# En este curso nos centraremos en el análisis de datos estructurados, es decir datos q
# encuentran organizados en un formato particular (e.g., esquema tabular)
# Esto contrasta con datos no-estructurados, como podría ser la información contenida e
# archivo de columnas periodísticas
# Los datos estructurados permiten su análisis, visualización y modelación
```

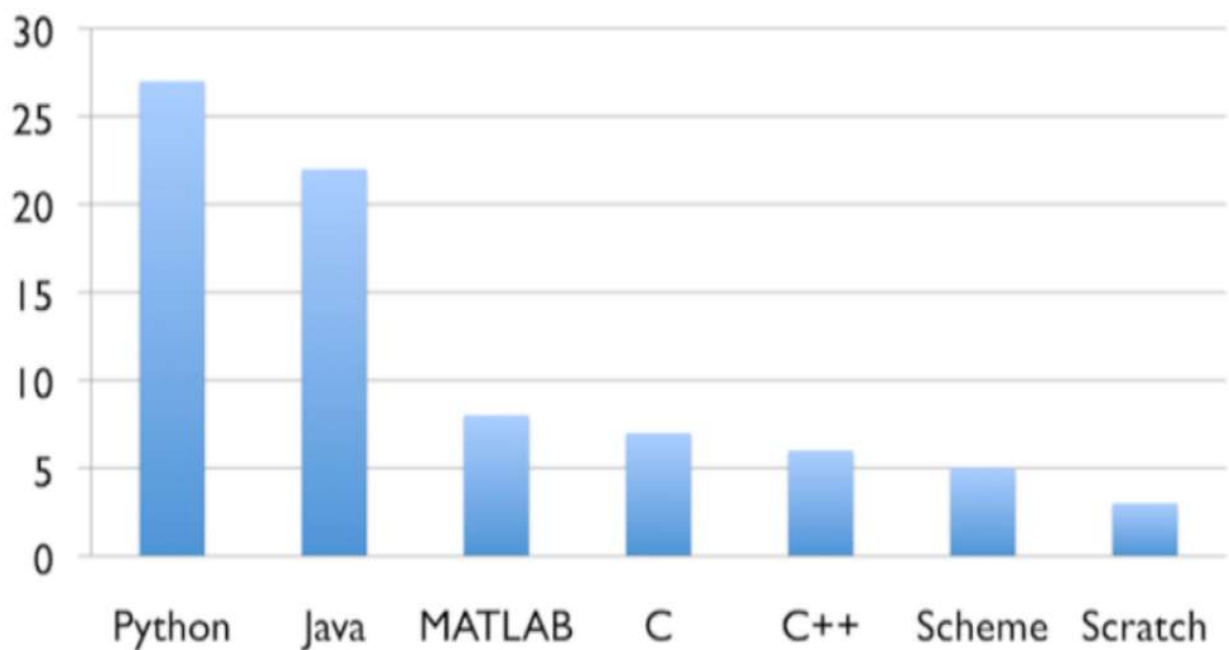
**Indicadores de la popularidad de Python:**

**(a) Su uso en la enseñanza de programación**

**(b) Tasa de adopción de Python (preguntas en Stack Overflow)**



Number of top 39 U.S. computer science departments that use each language to teach introductory courses



Analysis done by Philip Guo ([www.pgbovine.net](http://www.pgbovine.net)) in July 2014, last updated 2014-07-29

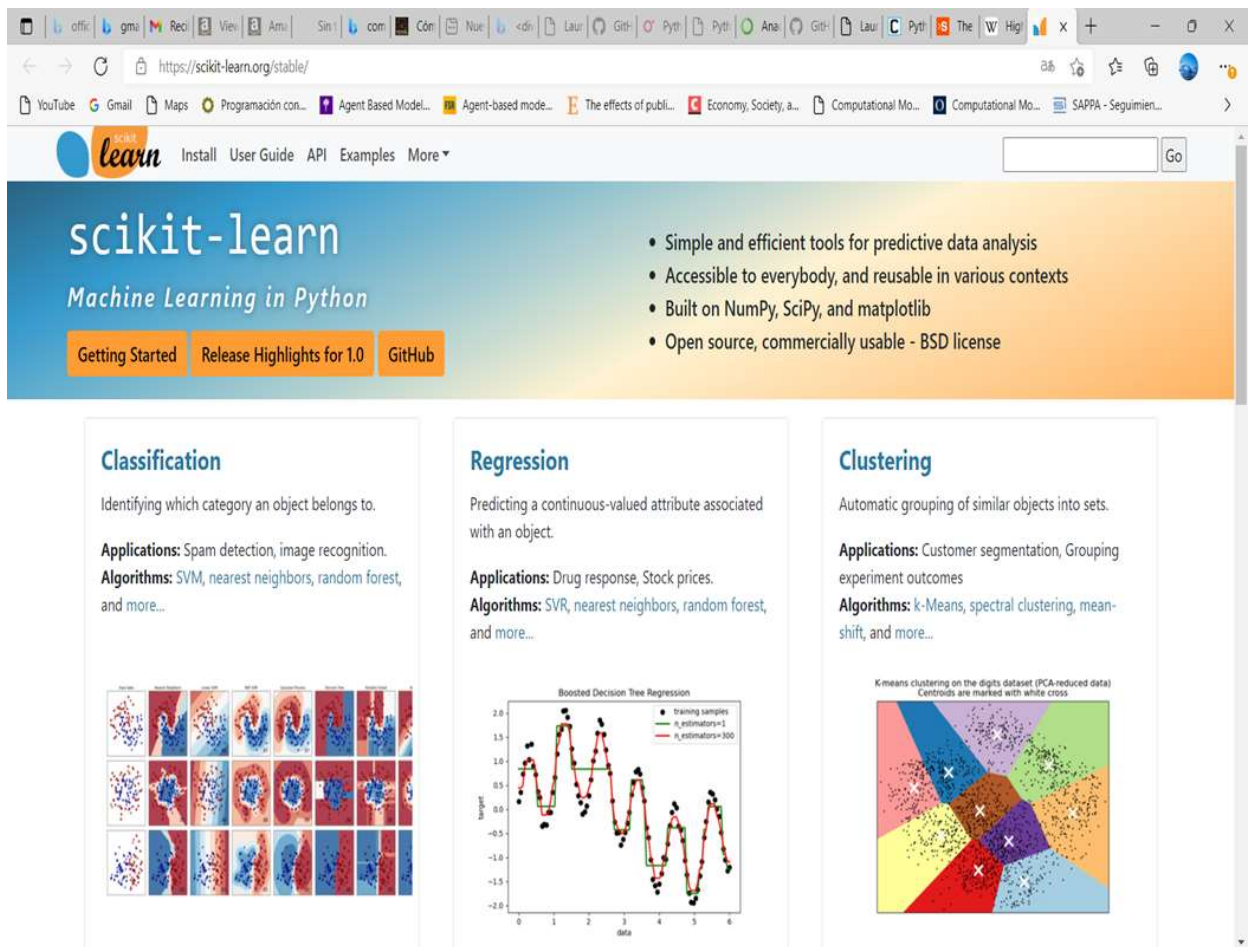
## (2) Ecosistema de librerías

In [2]: `# Las librerías del 'nivel bajo' son de uso general y son empleadas en diferentes aplic`

```
# manejo de datos, programación científica, machine Learning (ML)
# Las librerías de 'nivel alto' permiten realizar tareas muy especializadas con muy poc
# Muchas de estas librerías son de acceso libre y están disponibles a través de distrib
# de programas como Anaconda
```



Python se ha vuelto muy popular en el manejo de datos (Pandas) y en ML (scikit-learn)



In [3]:

```
# Numpy: permite trabajar con datos numéricos y realizar operaciones matemáticas media
# vectoriales y métodos de álgebra lineal
# Los objetos de Numpy (array) son muy eficientes para almacenar y manipular datos num
# comparación con otras estructuras de datos de Python
# Pandas: provee una estructura de 'alto nivel' y funciones que facilitan trabajar con
# datos estructurados en arreglos tabulares, su objeto básico es el DataFrame
# Con este objeto es posible indexar datos, reformatearlos, reducirlos, separarlos, agr
# ya sea en series o en arreglos tabulares con etiquetas
# Matplotlib: permite realizar gráficas y otras visualizaciones bidimensionales
# IPython: mejora el potencial interactivo de Python. Aunque no proporciona herramienta
# analíticas adicionales, facilita el flujo de trabajo con los scripts
# Jupyter: también es una herramienta interactiva, pero puede operar con otros lenguaje
# facilita intercambiar scripts ejecutables entre usuarios
# Scipy: colección de paquetes de gran utilidad para la programación científica (ODEs)
# Scikit-Learn: paquetería más popular de Python para implementar rutinas predictivas
# statsmodel: paquetería estadística con métodos de la estadística clásica (frequentist
# tanto paramétrica como no-paramétrica, permite análisis inferenciales con estimadores
```

### (3) Instalación de Python y sus interfaces de ejecución mediante comandos (programming shell)

In [4]:

```
# Primer paso: Bajar e instalar Python con el distribuidor Anaconda (edición individual
# https://www.anaconda.com/products/individual
# Versión disponible 3.9. Incluye Jupyter e Spyder (IDE: Interactive Development Enviro
# integra el manejo del script (código) con el Shell (consola de comandos)
# Disponible para Windows, Linux y OS X de Apple
```

```
# Anaconda Navigator: consola desde donde pueden lanzarse distintas paqueterías, alguna  
# disponible en el archivo de programas al hacer la instalación  
# Se puede acceder a la ventana (terminal) de comandos (prompt), a Jupyter y Spyder des  
# icono de Windows al hacer clic en el archivo de Anaconda  
# Si no se usa Spyder, el código puede escribirse en un archivo de texto, como el Noteb  
# Debe guardarse con terminación .py para que el procesador de Python lo reconozca  
# Las librerías que vienen con Anaconda se pueden actualizar de forma muy sencilla con  
# comando insertado en la terminal o ventana de comandos: conda update anaconda
```

## Navegador de Anaconda



https://www.anaconda.com/products/individual

ANACONDA Products Pricing Solutions Resources Partners Blog Company Get Started

# Individual Edition

## Your data science toolkit

With over 25 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for

**Anaconda Individual Edition**

[Download](#)

For Windows  
Python 3.9 • 64-Bit Graphical Installer • 510 MB

This website uses cookies to ensure you get the best experience on our website. [Privacy Policy](#) [Accept](#)

Anaconda Navigator

ANACONDA.NAVIGATOR

Home

Applications on base (root) Channels

 CMD.exe Prompt 0.1.1 Run a cmd.exe terminal with your current environment from Navigator activated. <a href="#">Launch</a>	 DataLab Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team. <a href="#">Launch</a>	 IBM Watson Studio Cloud IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling. <a href="#">Launch</a>	 JupyterLab 3.2.1 An extensible environment for interactive and reproducible computing, based on the Jupyter notebook and Architecture. <a href="#">Launch</a>	 Jupyter Notebook 4.4.0 Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis. <a href="#">Launch</a>	 Powershell Prompt 0.0.1 Run a Powershell terminal with your current environment from Navigator activated. <a href="#">Launch</a>
 Qt Console 5.1.1 PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more. <a href="#">Launch</a>	 Spyder 5.1.5 Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features. <a href="#">Launch</a>	 Glueviz 1.0.0 Multidimensional data visualization across files. Explore relationships within and among related datasets. <a href="#">Install</a>	 Orange 3 3.26.0 Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox. <a href="#">Install</a>	 PyCharm Professional A full-fledged IDE by JetBrains for both Scientific and web Python development. Supports HTML, JS, and SQL. <a href="#">Install</a>	 RStudio 1.1.456 A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks. <a href="#">Install</a>

ANACONDA Commercial Edition  
Commercial grade open-source software.  
[Upgrade](#)

Premium packages and dedicated support.

Documentation

Anaconda Blog

Twitter YouTube GitHub

## (4) Consolas alternativas (shells): terminal y Jupyter

```
In [5]: # Los comandos pueden correrse mediante: prompt, IPython, Notebook de Jupyter, IDE (Spy
# (1) Vía ventana de comandos (Anaconda powershell prompt) pueden ejecutarse Las instru
# por línea:
# Acceder al procesador: >>> python (dar regresar en cada línea)
# >>> a = 5 , >>> print (5) , regresa el valor 5
# Para salir de Python, poner >>> exit() , Para limpiar La ventana, poner >>> clear
# (2) Vía IPython puede abrirse desde el navegador de Anaconda o desde La ventana de co
# escribiendo ipython
# Genero un archivo con el Libro de Notas de Windows y lo guardo en algún folder donde
# tengo scripts de Python.
# Escribo: print ('Hello world') y lo guardo en saludos.py
# Copiar la dirección del archivo en la ventana del navegador de Windows
# Entro a IPython y lo direcciono escribiendo In [1] cd c:\Users\.....
# Checo que efectivamente el archivo ha sido ubicado: In[2] ls *.py (dar retorno)
# Corro el programa: In[3] run saludos.py , regresa el texto: Hello World
```

```
In [6]: # (3) Vía Notebooks en Jupyter. Desde el navegador de Anaconda puede abrirse Jupyter,
# en la terminal escribir jupyter notebook, o bien usar la siguiente URL: http://loc
# Esta interfaz de Python 'sustentada en el browser' permite: (i) escribir comandos,
# (ii) incluir texto formateado, (iii) incorporar visualizaciones y videos
# No requiere de un procesador instalado en la computadora para correrse
# Muy conveniente cuando se empieza a aprender a programar, para crear tutoriales y pa
# intercambiar códigos
# Se hace clic en el botón New en el extremo derecho para crear una Notebook, las cuales
# manejan a través de Internet.
# Van apareciendo celdas (Cells) en las que se insertan los comandos a ejecutar
# Se corren con el botón Run o con las teclas shift + enter
# Para guardar la Notebook, se usa save as y se almacena en la WEB con el identificado
# puede eliminarse con palomita y un clic en la papelera
# También se puede introducir texto, pero hay que cambiar el modo con los comandos.
# Cell > Cell Type > Markdown; Escribe: Este es un texto. Se puede insertar una nueva
# Insert Cell Below, y luego cambiar de modo: Cell Type > Code
```

```
In [7]: a = 5
print(a)
print ('Hello world')
```

```
5
Hello world
```

## (5) Transferencias de archivos:

```
In [8]: # (1) Un Notebook de Jupyter localizado en algún repositorio (github) puede descargarse
# la computadora en algún folder
# (2) Desde la compu puede cargarse en Jupyter haciendo clic en Upload dos veces
# (3) Un notebook en Jupyter puede abrirse con un clic, y bajarse (download) como archi
# *.ipynb o *.py para ser guardada en un folder de la computadora
```

## Jupyter dashboard

Select items to perform actions on them.

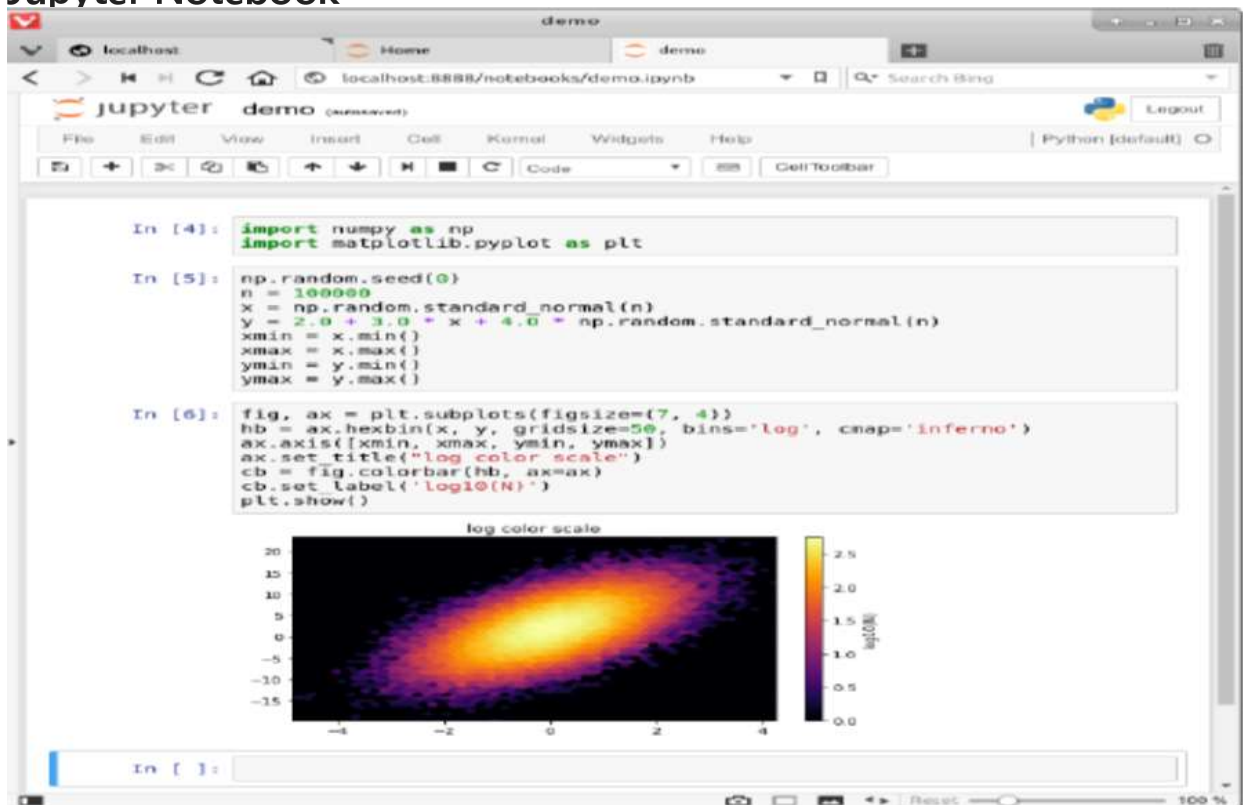
Upload

New



<input type="checkbox"/>	0	Name	Last Modified	File size
<input type="checkbox"/>	/			
<input type="checkbox"/>	Applications		il y a 3 jours	
<input type="checkbox"/>	Desktop		il y a 7 heures	
<input type="checkbox"/>	Documents		il y a 3 jours	
<input type="checkbox"/>	Downloads		il y a 35 minutes	
<input type="checkbox"/>	Dropbox		il y a 2 jours	
<input type="checkbox"/>	Movies		il y a 3 jours	
<input type="checkbox"/>	Music		il y a 7 heures	
<input type="checkbox"/>	Pictures		il y a 3 jours	
<input type="checkbox"/>	Public		il y a 3 jours	
<input type="checkbox"/>	VirtualBox VMs		il y a 6 heures	
<input type="checkbox"/>	VM		il y a 6 heures	
<input type="checkbox"/>	Zotero		il y a 6 heures	

## Jupyter Notebook



## (6) Un ejemplo de Notebook

In [9]: *# Vamos a correr el programa disponible en: [http://matplotlib.org/3.1.1/gallery/pie\\_and\\_polar\\_charts/polar\\_bar.html](http://matplotlib.org/3.1.1/gallery/pie_and_polar_charts/polar_bar.html).  
# Tenemos que incorporar el siguiente Código:*



In [10]:

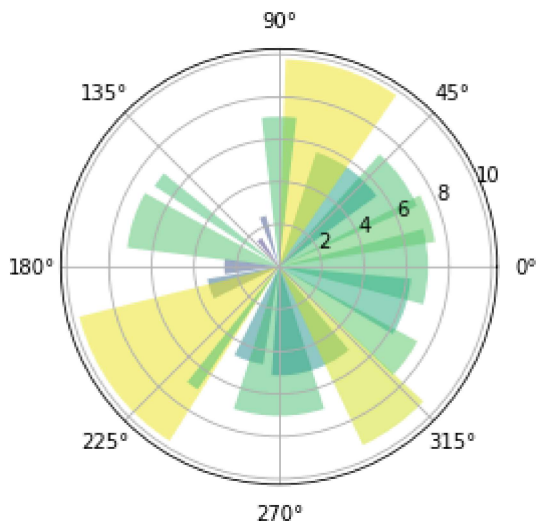
```
# Diagrama de barra utilizando la libreria matplotlib
import numpy as np
import matplotlib.pyplot as plt

# Fixing random state for reproducibility
np.random.seed(19680801)

# Compute pie slices
N = 20
theta = np.linspace(0.0, 2 * np.pi, N, endpoint=False)
radii = 10 * np.random.rand(N)
width = np.pi / 4 * np.random.rand(N)
colors = plt.cm.viridis(radii / 10.)

ax = plt.subplot(111, projection='polar')
ax.bar(theta, radii, width=width, bottom=0.0, color=colors, alpha=0.5)

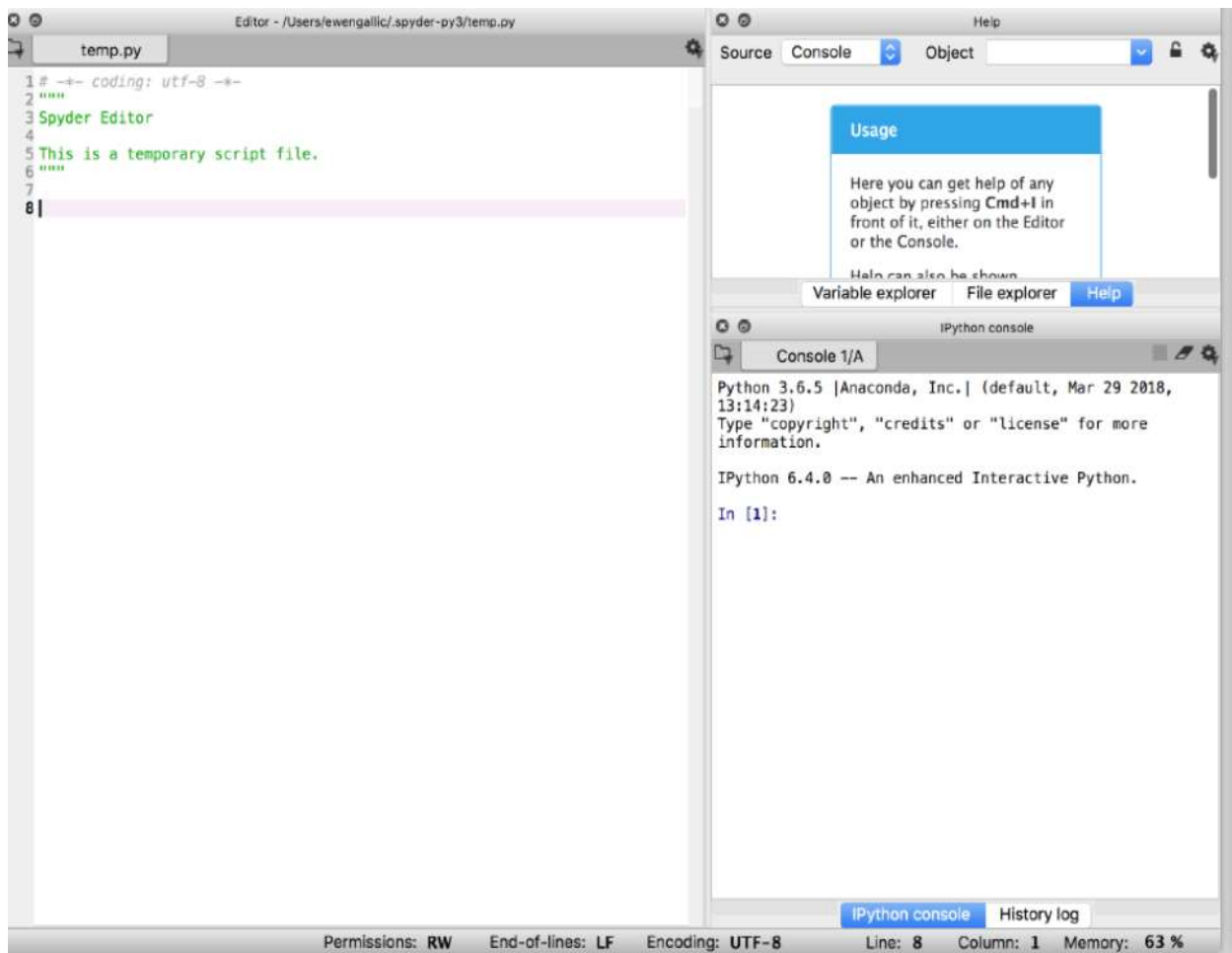
plt.show()
```



## (7) Uso de Spyder

In [11]:

```
# Spyder incluye: un script para escribir códigos, una consola (shell) para ejecutar co
# una ventana con ayudas, explorador de variables, y graficador
# Puede abrirse desde el navegador de Anaconda o desde la barra de tareas
```



```
In [12]: # Cuando se abre Spyder por primera vez se recomienda leer el tutorial
# En el script, comentarios se inician con un #, de lo contrario la línea se interpreta
# de un programa ejecutable (se corre con el run, F5, flecha verde)
# Los resultados de la corrida se observan en la consola
# Desde la consola también se pueden establecer comandos ejecutables, los cuales se cor
# de manera interactiva
# Para eliminar las variables almacenadas en la consola se usa %reset o se hace un clic
# del basurero
# La naturaleza de la variable puede observarse en la ventana del explorador y el lista
# variables se encuentra tecleando dir() en la consola
# En el icono de preferencias puede cambiarse la configuración de la interfaz
# En la sección de script se hace la tabulación de forma automática
# Identifica errores de codificación
```

## (8) Uso del tabulador e introspección

```
In [13]: # Con la tecla del Tab (flechas encontradas horizontales) del teclado es posible comple
# Las instrucciones a insertar en una línea de script
```

```
In [14]: an_apple = 27
an_example = 42
# Checar an<Tab>
# La consola arroja el siguiente texto: an_apple and an_example any
# (dos variables previamente definidas, y una función auto-definida: any)
```

```
In [15]: # El tabulador también nos ayuda a identificar funciones (métodos) que se pueden aplica
# a una variable.
b = [1, 2, 3]
# Checar b.<Tab>
# Lo que arrojaría el siguiente resultado: b.append b.count b.insert b.reverse b.clea
# b.pop b.sort b.copy b.index b.remove
```

```
In [16]: # Cuando a una variable, comando o función se le agrega un signo de interrogación,
# la consola presenta la documentación pertinente Ejemplo: b? print?
print?
```

```
In [18]: b?
```