Unidad 4. Entorno de desarrollo

Por Sarahí Aguilar

Universidad Panamericana ITISI 2016 BDP COM145 Primavera 2022

Introducción a R y Python

Librerías fundamentales de Python

Ambientes de desarrollo de Python

,

Introducción a Anaconda

Introducción a R y Python

Librerías fundamentales de Python

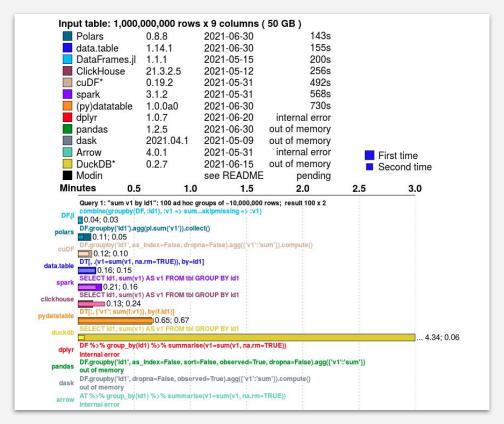
Ambientes de desarrollo de Python

Introducción a Anaconda



Jan 2022	Jan 2021	Change	Programming Language		Ratings	Change
1	3	^	•	Python	13.58%	+1.86%
2	1	•	Э	С	12.44%	-4.94%
3	2	•	<u>\$</u> ,	Java	10.66%	-1.30%
4	4		0	C++	8.29%	+0.73%
5	5		3	C#	5.68%	+1.73%
6	6		VB	Visual Basic	4.74%	+0.90%
7	7		JS	JavaScript	2.09%	-0.11%
8	11	^	ASM	Assembly language	1.85%	+0.21%
9	12	^	SQL	SQL	1.80%	+0.19%
10	13	^	3	Swift	1.41%	-0.02%
11	8	•	php	PHP	1.40%	-0.60%
12	9	~	R	R	1.25%	-0.65%
13	14	^	~GO	Go	1.04%	-0.37%
14	19	*	(Delphi/Object Pascal	0.99%	+0.20%
15	20	*	450	Classic Visual Basic	0.98%	+0.19%
16	16		A	MATLAB	0.96%	-0.19%

Top 16 lenguajes de programación según <u>índice TIOBE</u> en enero del 2022



Database-like ops benchmark por H2O

R es un lenguaje de programación y **entorno** enfocado al **cómputo estadístico** y **graficación**.



R es un lenguaje de programación y **entorno** enfocado al cómputo estadístico y graficación.

El entorno de R es un conjunto integrado de instalaciones de software para la manipulación, el cálculo y la visualización gráfica de datos.

El entorno de R incluye:



Una instalación eficaz de manejo y almacenamiento de datos



Un conjunto de operadores para cálculos en matrices.



Una colección amplia, coherente e integrada de herramientas para el análisis de datos.



Facilidades gráficas para el análisis de datos y visualización en pantalla o en papel.

R es un lenguaje de programación y entorno enfocado al **cómputo estadístico** y graficación.

R fue creado inicialmente con fines de investigación estadística y hoy proporciona una amplia variedad de técnicas estadísticas mediante implementaciones accesibles y simples.

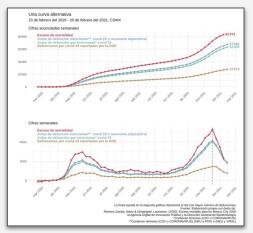
Por ejemplo, modelado lineal y no lineal, pruebas estadísticas clásicas, análisis de series de tiempo, clasificación, clustering, entre otras.



R es un lenguaje de programación y entorno enfocado al cómputo estadístico y graficación.

Facilidad con la que se pueden producir gráficos de alta calidad; una de las fortalezas de R.

En R, es fácil obtener un buen diseño en las visualizaciones de datos pues el usuario conserva el control total de todos los elementos gráficos.



Ejemplo de gráfico producido 100% en R.

La importancia de R



R es gratis.



La **documentación** de R es accesible para todas las áreas de estudio.



La **comunidad** de programadores de R (también conocidos como "useRs") es diversa e incluyente y abierta a ayudar y colaborar.



R es constantemente actualizado.



En enero del 2022, R ocupa el puesto #12 en el índice TIOBE, una medida de **popularidad** de los lenguajes de programación.



R está disponible como **Software Libre** bajo los términos de la Licencia Pública General GNU de The Free Software Foundation.



R es **escalable** a través de sus paquetes.

Python es un lenguaje de programación de **propósito general**, de **alto nivel** e **interpretado**.

- Se tipifica dinámicamente y maneja un sistema de garbage collection.
- Admite múltiples paradigmas de programación, incluida la programación estructurada, orientada a objetos y funcional.
- Su filosofía de diseño enfatiza la legibilidad del código con el uso de sangrías significativas.
- Sus construcciones de lenguaje tienen como objetivo ayudar a los programadores a escribir código claro y lógico para proyectos de pequeña y gran escala.



La importancia de Python

"Python is powerful... and fast; plays well with others; runs everywhere; is friendly & easy to learn; is Open."

python.org

"Python has been an important part of Google since the beginning, and remains so as the system grows and evolves. Today dozens of Google engineers use Python, and we're looking for more people with skills in this language."

Peter Norvig, director de Search Quality en Google

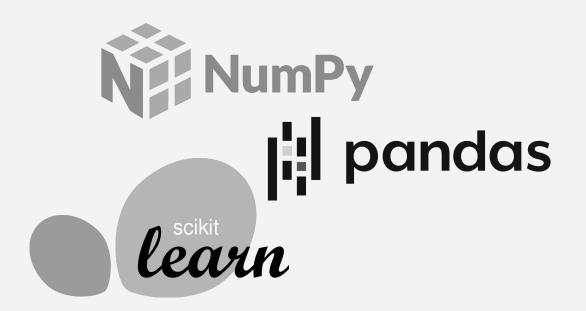


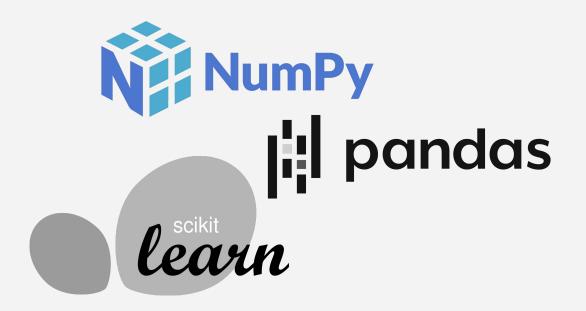
Introducción a R y Python

Librerías fundamentales de Python

Ambientes de desarrollo de Python

Introducción a Anaconda

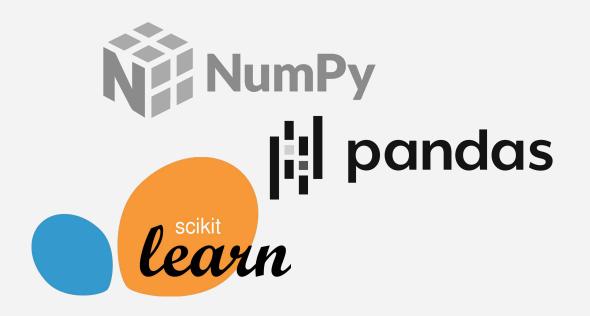




NumPy proporciona, entre otras cosas, soporte para arreglos multidimensionales con operaciones básicas sobre ellos y útiles funciones de álgebra lineal.



Pandas proporciona estructuras de datos de alto rendimiento y herramientas de análisis de datos. La característica clave de Pandas es un objeto DataFrame que es una representación rápida y eficiente para la manipulación de datos con indexación integrada.



Scikit-learn ofrece herramientas simples y eficientes para tareas comunes en machine learning, como clasificación, regresión, agrupación, reducción de dimensionalidad, selección de modelos y preprocesamiento.

Introducción a R y Python

Librerías fundamentales de Python

Ambientes de desarrollo de Python

Introducción a Anaconda



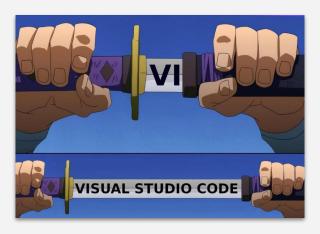
IPython proporciona una arquitectura para la computación interactiva con:

- Un shell interactivo.
- Soporte para visualización de datos interactivos y uso de kits de herramientas GUI.
- Intérpretes flexibles e integrables para cargar en sus propios proyectos.
- Herramientas fáciles de usar y de alto rendimiento para computación paralela.

Ambientes de desarrollo de Python

Entornos de desarrollo integrados

PyCharm
Visual Studio Code
Sublime Text
Vim
Atom
Jupyter Notebook
Jupyter Lab



Ambientes de desarrollo de Python



Colab, o "Collaboratory", permite escribir y ejecutar Python en el navegador.
Requiere cero configuración.
Brinda acceso gratuito a las GPU.
Fácil de compartir.



Introducción a R y Python

Librerías fundamentales de Python

Ambientes de desarrollo de Python

Introducción a Anaconda



Con más de 25 millones de usuarios en todo el mundo, la edición individual de código abierto es la forma más fácil de realizar ciencia de datos Python/R de forma independiente dado el conjunto de herramientas que lo equipa para trabajar con miles de paquetes y bibliotecas de código abierto.



Con conda, puedes crear, exportar, listar, eliminar y actualizar entornos que tengan diferentes versiones de Python y/o paquetes instalados en ellos.

Cambiar o moverse entre entornos se denomina activar el entorno.

También puedes compartir un "archivo de entorno".



Comandos relevantes de conda

```
Crear ambiente: conda create --name mytestenv
Crear ambiente con versión de Python específica: conda create -n myenv python=3.6
Crear ambiente a partir de .yml: conda env create -f environment.yml
Activar ambiente existente: conda activate mytestenv
Imprimir lista de ambientes existentes: conda env list
Exportar .yml de ambiente activo: conda env export > environment.yml
Instalar librería en ambiente activo con versión específica: conda install mytestlib=0.20.3
Instalar múltiples librerías a la vez: conda install numpy pandas scikit-learn matplotlib
seaborn jupyter
Imprimir lista de librerías instaladas en ambiente activo: conda list
```

Introducción a R y Python

Librerías fundamentales de Python

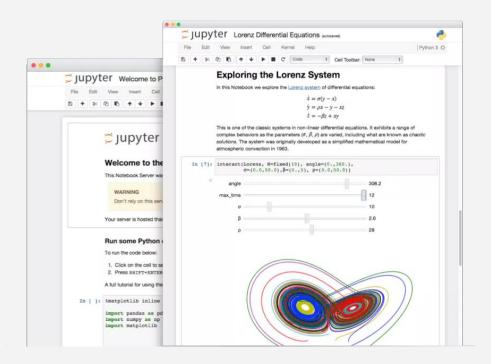
Ambientes de desarrollo de Python

Introducción a Anaconda



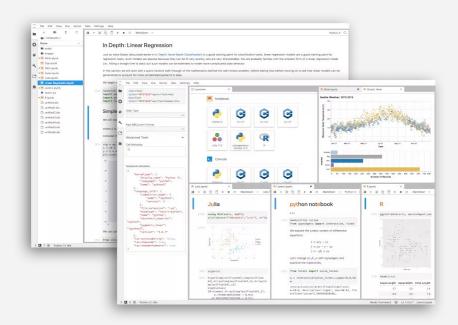
Jupyter es un software libre, de estándares abiertos y de servicios web para computación interactiva en todos los lenguajes de programación

Introducción a Jupyter



Jupyter Notebook es la aplicación web original para crear y compartir documentos computacionales. Ofrece una experiencia simple, optimizada y centrada en documentos.

Introducción a Jupyter



JupyterLab es el último entorno de desarrollo interactivo basado en la web para cuadernos, código y datos. Su interfaz flexible permite a los usuarios configurar y organizar flujos de trabajo en ciencia de datos, computación científica, periodismo computacional y aprendizaje automático. Un diseño modular invita a las extensiones para ampliar y enriquecer la funcionalidad.

Introducción a Jupyter

Comandos relevantes de conda

Instalar Jupyter en ambiente activo: conda install -c anaconda jupyter Instalar NumPy en ambiente activo: conda install numpy Instalar NumPy en ambiente activo: conda install pandas Iniciar jupyter notebook: jupyter notebook

Ejercicios para martes, 15 de febrero:

4.3.

- 1. Crear ambiente de conda a partir de test1environment.yml
- 2. Crear ambiente de conda desde 0 con nombre "test2" con jupyter notebooks, numpy y pandas.
- 3. Importar .yml de test2.4. Resolver los siguientes ejercicios en Python en un Jupyter notebook:
 - 4.1. Escribe una función que dado un string s y un entero no negativo n regrese n copias de s.
 - 4.2. Escribe una función que dado un string s y diferitero no negativo megrese incopias de s.

 4.2. Escribe una función que dado un arreglo de números revise si el producto de dos números del arreglo es par.
 - Escribe una función que dado un arreglo con valores None regrese el arreglo con los valores None reemplazados por el valor no None anterior en el arreglo.
- 4.4. Escribe una función que dada una lista regrese la misma lista ordenada.

Entregables via Moodle: Archivo .yml de test2 importado en punto #3 y Jupyter notebook con ejercicios resueltos del punto #4.