

Introducción a la Programación Para Analítica

Robinson Duque, Ph.D

Robinson.duque@correounivalle.edu.co

Lenguajes, Herramientas y Configuración

La eterna pregunta para este tipo de cursos,
¿qué lenguaje de programación utilizar?



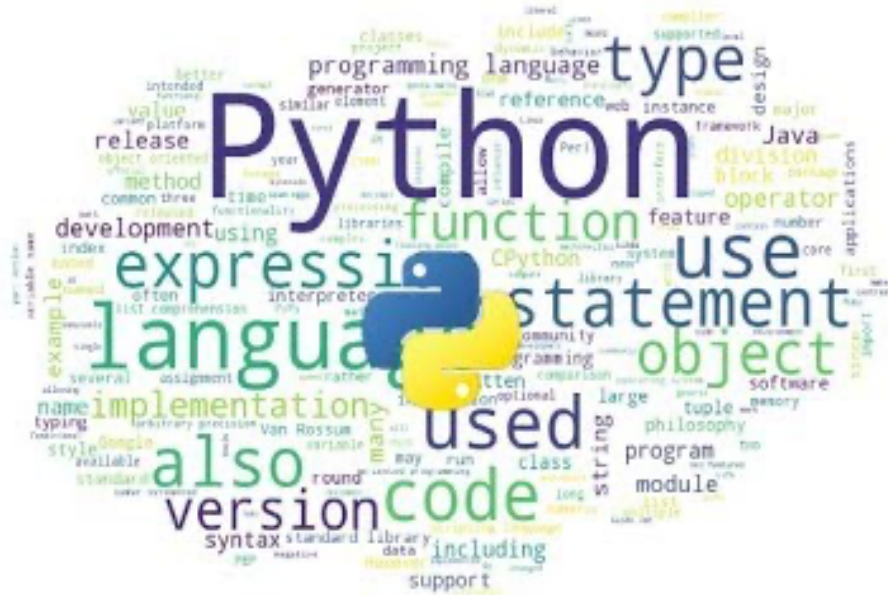
to general



- Es fácil de entender y aprender
- Lenguaje de programación de propósito general (no sólo para crear prototipos, también para construir sistemas y ponerlos en producción)
- Cuando de integración con otros sistemas se trata, existen hoy en día numerosos Frameworks de desarrollo (incluidos Web)
- Múltiples características (análisis exploratorio de datos, adquisición y limpieza de datos, manejo de estadística, visualización, alto rendimiento...)
- Librerías avanzadas en el campo de ciencia de datos (SciPy Ecosystem: IPython, NumPy, Pandas, Matplotlib, Scikit-Learn...)



el y propósito



<https://www.youtube.com/watch?v=L6QeU81HdbA>

- Lenguaje de programación de alto nivel y propósito general (Guido van Rossum, 1991)
- Python 2.0 (liberado en el 2000), Python 3.0 (liberado en el 2008)
- Lenguaje de programación interpretado
- Usado en múltiples plataformas
- Libre y de código abierto
- Orientado a objetos (soporta múltiples paradigmas de programación: procedimental, funcional, OO)
- Tipado dinámico



Librerías Esenciales

NumPy

- Numerical Python. Paquete para realizar cálculos científicos en Python
- Tratamiento de datos homogéneos basados en arreglos (ndarray object)
- Funciones vectorizadas para operaciones matemáticas con arreglos
- Lectura y escritura de conjuntos de datos alojados en disco
- Operaciones de álgebra lineal, generación de números aleatorios, etc
- Interoperabilidad con otros lenguajes como C, C++, Fortran, etc.



Librerías Esenciales

Pandas

- Paquete para realizar cálculos científicos en Python
- Tratamiento de datos heterogéneos y etiquetados
- Basado en (Series y DataFrames)
- Funciones vectorizadas para operaciones matemáticas con arreglos de NumPy
- Lectura y escritura de conjuntos de datos alojados en disco (hojas de cálculo, bases de datos SQL, archivos planos, archivos CSV, etc)
- Operaciones sofisticadas basadas en Índices (reshape, slice, aggregation, selection, filtering, masking, etc)



Librerías Esenciales

Matplotlib

- La librería más popular de Python para producir plots y visualizaciones
- Permite realizar análisis exploratorio de datos

IPython

- Componente básico del conjunto de herramientas del estándar de Python científico
- Provee un ambiente interactivo y exploratorio de computación
- Es una versión enriquecida del Shell de Python que permite codificar programas
- Permite utilizar “cuadernos” científicos basados en HTML a través de un browser
- Ofrece una interfaz para visualización de gráficos
- Ofrece una infraestructura para cálculos paralelos y distribuidos



Librerías Esenciales

SciPy

- Colección de paquetes que ofrecen soporte para un número de problemas en diversos dominios científicos. Algunos son:
 - Solvers para ecuaciones diferenciales
 - Rutinas de álgebra lineal y descomposición de matrices
 - Optimizadores de funciones
 - Distribuciones de probabilidad continuas y discretas, funciones de densidad, tests estadísticos, etc
 - NumPy y SciPy ofrecen una alternativa computacional para MATLAB

Scikit-Learn

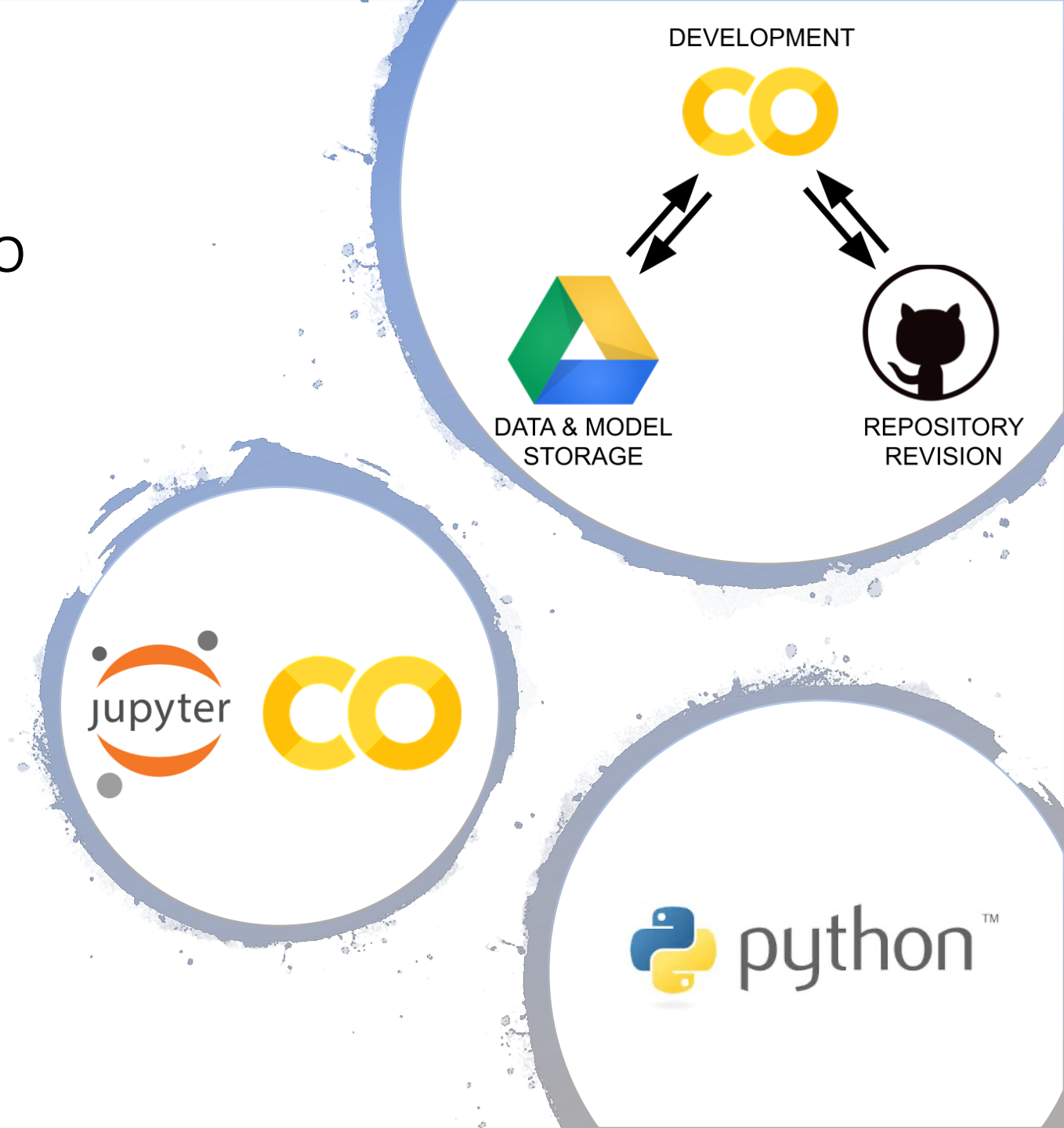
- Soporte para manejo de tareas relacionadas con aprendizaje de máquina

Herramientas para el Curso

Trabajo Online (Colaboratory – Recomendado)

- Disponible en: <https://colab.research.google.com>
- No requiere de configuración previa (sólo para Python)
- Compatible con los “[Jupyter Notebooks](#)” de extensión (*.ipynb)
- Los cuadernos permiten combinar código ejecutable con texto enriquecido (imágenes, HTML, LaTeX, Markdown)
- Almacenamiento en Google Drive
- Acceso gratuito a GPUs y TPUs
- Compartir y dar acceso colaborativo a cuadernos
- Se requiere cuenta de correo electrónico con google
- Filtros con ejemplos de códigos

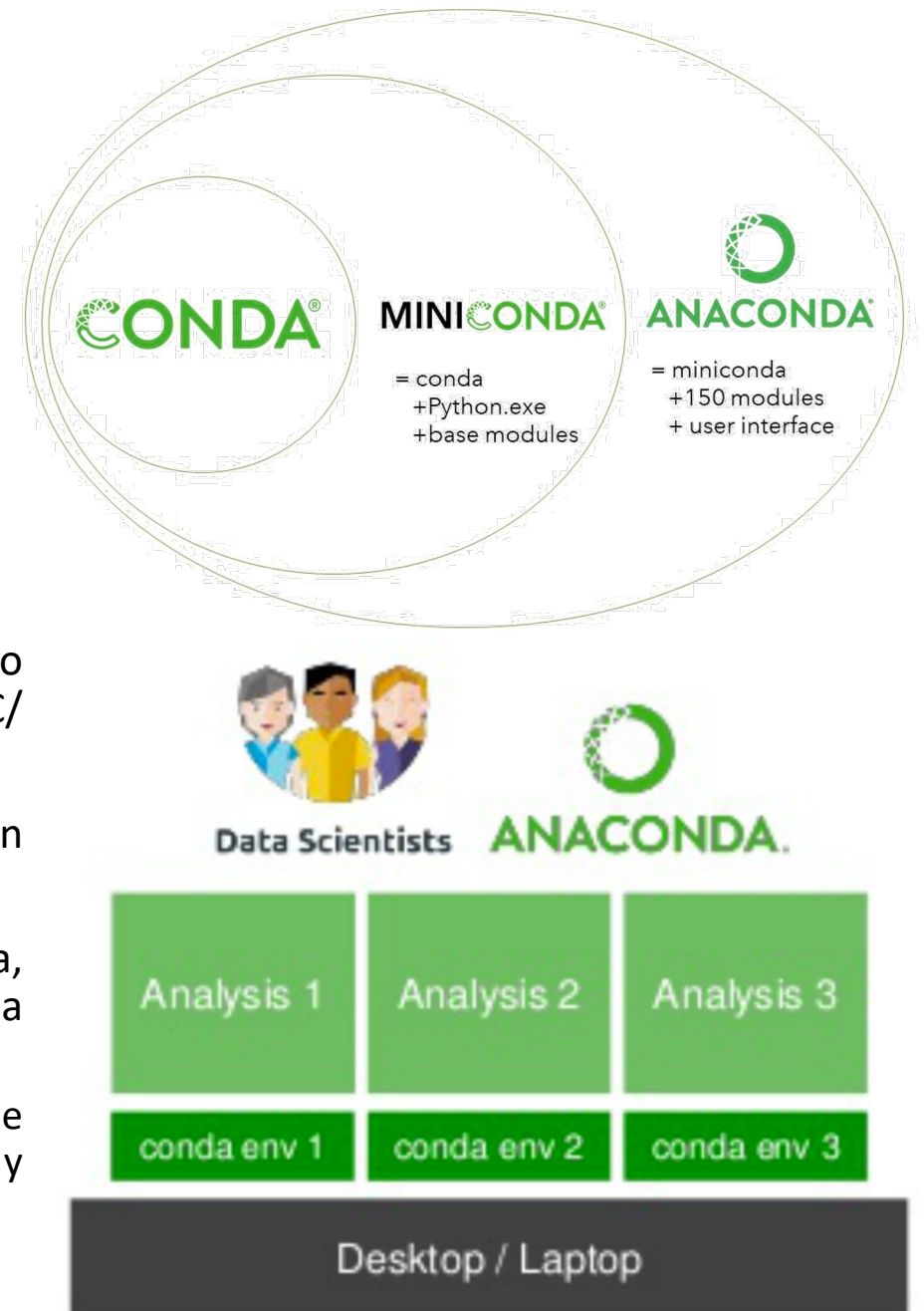
Ver Demostración Rápida



Herramientas para el Curso

Trabajo Offline (Miniconda – Recomendado)

- [Conda](#) es un **gestor de paquetes** y un **sistema de gestión de entornos** de código abierto
- Conda está escrito en Python pero puede gestionar entornos de código para diversos lenguajes (Python, R, Ruby, Lua, Scala, Java, JavaScript, C/C++, FORTRAN...)
- [Anaconda](#) es una distribución de los lenguajes Python y R, utilizada en ciencia de datos y aprendizaje de máquina
- [Miniconda](#) es una versión mínima de Anaconda que incluye Conda, Python y un número pequeño de paquetes útiles. En general Conda nos permitirá:
 - Crear ambientes configurados con diferentes lenguajes de programación (un ambiente es un directorio que contiene paquetes y herramientas específicos incluidas las respectivas dependencias)
 - Instalar fácilmente paquetes y dependencias
 - Cambiar fácilmente de un ambiente a otro



Fuente:

https://angus.readthedocs.io/en/2019/conda_tutorial.html

Herramientas para el Curso



Trabajo Offline (Miniconda – Recomendado)

- Disponible en: <https://docs.conda.io/en/latest/miniconda.html>
- Descargar la versión para Python 3.7 (Windows, MacOSX, Linux)
- Seguir las instrucciones para “[Regular Installation](#)” (acepte los valores por defecto)
- Durante la instalación les será solicitado si desean agregar Conda a las variables de entorno “PATH” en Windows o al iniciar la terminal en Linux y MacOS (sugiero seleccionar la opción):
 - [Ver video Windows](#)
 - [Ver video para MacOS](#)
 - [Ver video para Linux](#)
- Compruebe la instalación abriendo un terminal y ejecute el comando: **conda list**



Herramientas para el Curso



Ejemplo 1: Configuración del Ambiente de Trabajo (Python 3.7)

- `conda create -n mipython python=3.7`
- `conda activate mipython`
- `conda install numpy pandas scikit-learn matplotlib`
- `conda install -c conda-forge notebook`

Ejecutar Jupyter:

- `jupyter notebook`

Ejemplo 2: Configuración de Ambiente (Python 3.7 y R)

- `conda create -n mypyr r-essentials r-base python=3.7`
- `conda activate mypyr`
- `conda install numpy pandas scikit-learn matplotlib`
- `conda install -c conda-forge notebook`

Ejecutar Jupyter:

- `jupyter notebook`



Herramientas para el Curso



Comandos básicos:

- `conda info --env` (Ver lista de ambientes disponibles)
- `conda env list` (Ver lista de ambientes disponibles)
- `conda list` (Ver lista de paquetes instalados en el ambiente activo *)
- `conda deactivate` (Desactivar ambiente actual)
- `conda activate amb` (Activar el ambiente de nombre **amb**)
- `conda create --name py35 python=3.5` (Crear ambiente llamado py35 e instalar Python 3.5)
- `conda create --clone py35 --name py35-2` (Hacer una copia exacta del ambiente py35)
- `conda env remove --name amb` (Eliminar el ambiente **amb**)
- `conda install PACKAGENAME` (Instalar un paquete)
- `conda update PACKAGENAME` (Actualizar un paquete)

[Ver más...](#)