Programa: Introducción a Python [Código del curso]

Claudia Ranocchia

Año academico 2024-2025

Índice

Módulo 1 Conceptos básicos de Python

Este módulo cubre los fundamentos de Python, proporcionando una base sólida para la programación en este lenguaje. Se explicarán conceptos clave como la sintaxis, estructuras de datos y funciones, que son esenciales para el desarrollo de cualquier aplicación en Python. El objetivo es familiarizar con las herramientas básicas del lenguaje para que puedan comenzar a escribir código de manera efectiva.

1.1. Introducción a Python e instalación

En esta sección se introduce Python como un lenguaje de programación versátil y fácil de aprender. Se destacará su creciente popularidad en ciencia de datos, inteligencia artificial y desarrollo web. También se cubrirán los pasos para instalar Python en diferentes sistemas operativos y se explicarán brevemente las herramientas recomendadas para programar en Python, como el uso de editores de código (Visual Studio Code, PyCharm) y Jupyter Notebooks.

1.2. Sintaxis, variables, tipos de datos, operaciones simples

Se presenta la sintaxis básica de Python, haciendo hincapié en su simplicidad y legibilidad. Se explicarán conceptos fundamentales como la declaración de variables, los tipos de datos más comunes (enteros, flotantes, cadenas) y las operaciones matemáticas y lógicas simples que pueden realizarse con ellos.

1.3. If y varios bucles

Esta sección introduce las estructuras de control de flujo, como las sentencias condicionales (if, elif, else) y los bucles (for, while). Se aprenderá a escribir código que ejecute ciertas acciones en función de condiciones específicas y cómo repetir tareas de manera eficiente utilizando bucles. Se incluirán ejemplos prácticos y ejercicios para reforzar estos conceptos.

1.4. Listas, tuplas, conjuntos, diccionarios

Aquí se cubren las estructuras de datos más importantes de Python: listas, tuplas, conjuntos y diccionarios. Se aprenderá cómo crear, manipular y acceder a estos contenedores de datos, así como las diferencias clave entre ellos. Estos conceptos son fundamentales para trabajar con grandes volúmenes de datos y para organizar información de manera estructurada.

1.5. Funciones y lambdas

Se introduce el concepto de funciones en Python, destacando su importancia para organizar y reutilizar el código. Se explicará cómo definir y llamar funciones, así como los diferentes tipos de parámetros que pueden utilizarse. Además, se presentarán las funciones anónimas (lambdas), que permiten escribir funciones pequeñas y sencillas en una sola línea. Los ejemplos prácticos ayudarán a ilustrar el uso de funciones en diversos contextos.

1.6. Gestión de archivos

Esta sección se centra en la lectura y escritura de archivos en Python. Se explicarán las operaciones básicas de gestión de archivos, como abrir, leer, escribir y cerrar archivos. Se también aprenderá cómo trabajar con diferentes tipos de archivos (texto, CSV, JSON) y la importancia de manejar excepciones para evitar errores comunes durante la manipulación de archivos.

Módulo 2 Numpy

2.1. Introducción e instalación

Numpy es una biblioteca fundamental para la manipulación de array en Python. Para instalar Numpy, puedes usar el siguiente comando:

```
pip install numpy
```

2.2. Indexación de arrays

En Numpy, los elementos de un array pueden ser accedidos mediante la indexación. Puedes seleccionar un único elemento, un rango de elementos o incluso usar indexación basada en condiciones.

2.3. Cortar matrices

Cortar matrices (slicing) es una técnica útil para seleccionar subconjuntos de una matriz. Puedes extraer filas, columnas o porciones específicas de un array utilizando índices de inicio y fin.

2.4. Tipos de datos

Cada array en Numpy tiene un tipo de dato (dtype). Estos tipos de datos definen el tipo de los elementos del array (por ejemplo, `int`, `float`, `complex`, etc.). Es posible especificar o cambiar el tipo de dato de un array.

2.5. Ver y copiar

Al crear una vista (view) o una copia de una matriz, es importante comprender las diferencias. Una vista es simplemente una referencia a los mismos datos en memoria, mientras que una copia crea un nuevo array con una copia independiente de los datos.

2.6. Shape y reshape

El `shape` de una matriz indica sus dimensiones. La función `reshape()` te permite cambiar la forma de una matriz sin alterar los datos subyacentes.

2.7. Iteración de matrices

Numpy ofrece varias formas de iterar sobre los elementos de una matriz. Puedes usar bucles simples o métodos avanzados como `nditer()` para una iteración más eficiente.

2.8. Fusión de varias matrices

Puedes combinar matrices con funciones como `np.concatenate()` o `np.vstack()` y `np.hstack()` para fusionar matrices vertical u horizontalmente.

2.9. Dividir una matriz

Para dividir una matriz en múltiples submatrices, Numpy proporciona funciones como `np.split()` y `np.hsplit()` o `np.vsplit()`.

2.10. Buscar, rellenar y ordenar

Numpy permite realizar operaciones de búsqueda en matrices, rellenar matrices con valores específicos y ordenar los elementos mediante las funciones `np.searchsorted()`, `np.fill()` y `np.sort()`.

2.11. La clase aleatoria

La clase `numpy.random` proporciona una variedad de métodos para generar números aleatorios y realizar simulaciones.

2.12. Funciones universales

Las funciones universales (ufuncs) en Numpy son funciones que operan sobre arrays de manera eficiente. Estas incluyen operaciones matemáticas básicas como suma, resta, multiplicación, y muchas otras.

Módulo 3 Pandas para el análisis de datos

Aquí tienes la subsección creada en LaTeX, con cada elemento de la lista convertido en subsección, igual que antes, ahora con el tema que me has indicado:

3.1. Introducción e instalación

Pandas es una biblioteca fundamental para la manipulación y análisis de datos en Python. Se usa comúnmente para trabajar con datos estructurados, como tablas. Para instalar Pandas, puedes usar el siguiente comando:

```
pip install pandas
```

3.2. Conjuntos de datos

En Pandas, los datos se gestionan principalmente mediante dos estructuras: Series y DataFrames. Los DataFrames son similares a las tablas de bases de datos o Excel, lo que permite una manipulación eficiente de conjuntos de datos grandes.

3.3. Qué son los marcos de datos

Un DataFrame es una estructura bidimensional, similar a una tabla, que contiene columnas de datos etiquetadas. Es uno de los objetos más importantes en Pandas y permite una fácil manipulación, filtrado y análisis de datos.

3.4. Carga desde CSV y JSON

Pandas ofrece funciones rápidas para cargar datos desde archivos CSV y JSON usando `pd.read_csv()` y `pd.read_json()`, respectivamente. Estas funciones permiten leer datos y almacenarlos en DataFrames para su posterior análisis.

3.5. Lectura y selección de datos

Una vez que los datos están cargados en un DataFrame, puedes leer y seleccionar subconjuntos de datos mediante el uso de índices, nombres de columnas o condiciones. Esto se hace comúnmente con la notación de corchetes o usando métodos como `.loc` y `.iloc`.

3.5.1. LOC vs ILOC

Los métodos `.loc` y `.iloc` permiten seleccionar datos en un DataFrame. `.loc` selecciona datos basándose en etiquetas (nombres de columnas o índices), mientras que `.iloc` selecciona datos según su posición numérica.

3.6. Iterar el marco de datos

Puedes iterar sobre un DataFrame fila por fila o columna por columna usando bucles o métodos como `.iterrows()` y `.itertuples()` para acceder a los datos de manera eficiente.

3.7. Ordenar datos

Los datos en un DataFrame pueden ser ordenados por los valores de una o más columnas usando el método `.sort_values()`, lo que permite organizar los datos de acuerdo con diferentes criterios.

3.8. Añadir, eliminar y mover columnas

Es sencillo añadir nuevas columnas a un DataFrame, eliminarlas o cambiar su posición. Puedes usar la notación de corchetes para añadir columnas, el método `.drop()` para eliminar y reordenar las columnas moviéndolas directamente en el DataFrame.

3.9. Guardar datos en un archivo CSV

Después de procesar los datos, puedes guardarlos en un archivo CSV con el método `.to_csv()`, que exporta los datos del DataFrame de manera eficiente a un formato legible para otros programas.

3.10. Filtrar el marco de datos con condiciones

Puedes filtrar un DataFrame aplicando condiciones lógicas a las columnas, lo que te permitirá seleccionar subconjuntos específicos de datos según las características deseadas.

3.11. Editar el marco de datos

Pandas permite realizar ediciones directas en los datos, como modificar valores, reemplazar datos faltantes o actualizar información incorrecta en un DataFrame usando métodos como `.replace()` y `.update()`.

3.12. Limpiar datos

La limpieza de datos es una tarea esencial para el análisis. Pandas proporciona funciones útiles como `.dropna()` para eliminar filas o columnas con datos faltantes, o `.fillna()` para reemplazar valores faltantes con un valor especificado.

Módulo 4 Metodos para analisis econometrica

4.1. Introducción a statsmodels

Introducción al módulo statsmodels.api. Para instalar statsmodel puedes usar el siguiente comando:

```
pip install statsmodels
```

4.2. Pandas dataframe con statsmodel

Uso de DataFrames con pandas en statsmodels.

4.3. Mínimos Cuadrados Ordinarios (OLS)

Definición y lógica de los mínimos cuadrados. Implementación básica de un modelo OLS usando statsmodels.

Módulo 5 Introducción a Matplotlib

5.1. Introducción a Matplotlib

Matplotlib es una de las bibliotecas más populares de Python para la visualización de datos. Proporciona una amplia variedad de gráficos para representar datos de manera visual. Para instalar Matplotlib, puedes utilizar el siguiente comando:

```
pip install matplotlib
```

5.2. Personalización de gráficos

Matplotlib permite una personalización completa de los gráficos, incluyendo títulos, etiquetas en los ejes, leyendas, colores y estilos de línea. Estas opciones se gestionan con funciones como `plt.title()`, `plt.xlabel()`, `plt.ylabel()` y `plt.legend()`.

5.3. Gráfico de barras

Los gráficos de barras son útiles para representar y comparar cantidades en diferentes categorías. Se pueden crear fácilmente en Matplotlib usando el método `plt.bar()` para gráficos de barras verticales o `plt.barh()` para gráficos de barras horizontales.

5.4. Gráfico circular

Un gráfico circular (o gráfico de pastel) es ideal para mostrar la proporción de partes respecto a un todo. Matplotlib proporciona el método `plt.pie()` para crear gráficos circulares, que pueden personalizarse con colores, etiquetas y porcentajes.

5.5. Gráfico de áreas

Un gráfico de áreas es una variante de un gráfico de líneas donde el área entre la línea y el eje se rellena con color. Este tipo de gráfico se genera en Matplotlib con el método `plt.fill_between()`.

5.6. Histogramas

Los histogramas muestran la distribución de datos continuos dividiéndolos en intervalos (bins). Puedes crear un histograma con Matplotlib utilizando `plt.hist()` para visualizar la frecuencia de los datos en diferentes rangos.

5.7. Gráfico de dispersión

Los gráficos de dispersión (scatter plots) se utilizan para visualizar relaciones entre dos variables. En Matplotlib, se crean con `plt.scatter()`, permitiendo visualizar correlaciones o distribuciones de puntos en el plano.

5.8. Fecha y hora en gráficos

Matplotlib permite trabajar con datos de fechas y horas de manera sencilla. Puedes formatear y visualizar fechas en los ejes de un gráfico utilizando `matplotlib.dates` y funciones como `plt.gcf().autofmt_xdate()` para ajustar el formato de las fechas.

5.9. Gráficos de subtramas múltiples

Para mostrar múltiples gráficos en una sola figura, Matplotlib proporciona la función `plt.subplot()`, que permite crear una cuadrícula de gráficos en una única ventana, facilitando la comparación visual de distintos conjuntos de datos.

Calendario

Aula: XXX

Horario: 17:00-19:00.

Semana 1

Fecha	Tema	Duración
15/01/2025	Módulo 1: Conceptos básicos de Python 1.1-1.3	2 horas
16/01/2025	Módulo 1: Conceptos básicos de Python 1.4-1.6	2 horas

Semana 2

Fecha	Tema	Duración
23/01/2025	Módulo 2: Numpy	2 horas

Semana 3

Fecha	Tema	Duración
30/01/2025	Módulo 3: Pandas para el análisis de datos	2 horas

Semana 4

Fecha	Tema	Duración
06/02/2025	Módulo 4: Metodos para analisis econometrica	1 hora
06/02/2025	Módulo 5: Introducción a Matplotlib	1 hora

Semana 5

Fecha	Tema	Duración
12/02/2025	Dudas y preguntas	2 horas