

1. Información del equipo pedagógico y horarios de atención a estudiantes

Profesor magistral: Lucas Gómez Tobón
Correo electrónico: l.gomez@uniandes.edu.co

Asistente docente: Santiago Neira Hernández
Correo electrónico: s.neira10@uniandes.edu.co
Horario de atención a estudiantes: Agendar por correo.
Lugar de atención a estudiantes: Vía Zoom.

2. Descripción del curso

Este curso busca desarrollar habilidades de programación en el contexto de análisis de datos. Está dirigido a un público no familiarizado con la programación, o al menos no necesariamente en el contexto de análisis de datos. Se cubre desde una introducción rápida al ecosistema Python y a la sintaxis de sus estructuras de datos y control, seguido de tutoriales introductorios sobre manipulación, exploración y análisis de datos con paquetes ampliamente usados, como Numpy, Pandas, Seaborn, entre otros.

Este curso podrá darle herramientas a los estudiantes para trabajar en las clases en donde se requiera manipular datos, así como con el desarrollo de las tesis aplicadas.

3. Resultados de aprendizaje

Al final del curso los estudiantes estarán en capacidad de:

- Realizar operaciones matemáticas vectoriales y matriciales con numpy.
- Limpieza y manipulación de bases de datos.
- Crear visualizaciones de datos con las librerías matplotlib, seaborn y plotly.
- Automatizar tareas.
- Realizar análisis estadísticos descriptivos.

4. Cronograma

Semana	Semana 1
Objetivos de la semana	<ul style="list-style-type: none">• Comprender qué es Python y cuáles son sus características como lenguaje de programación.• Entender las diferencias entre Python y Anaconda.• Entender qué son los Integrated Development Environments (IDEs).

	<ul style="list-style-type: none"> • Aprender a instalar Python y sus demás requerimientos funcionales. • Aprender a manipular los objetos básicos de Python: tipos numéricos, secuencias, diccionarios, conjuntos y booleanos. • Instalación y cargue de librerías.
Descripción	Se hará una introducción al lenguaje de programación Python y a las demás herramientas tecnológicas que complementarán su uso, como lo son git y los IDEs.
Evaluación	Taller práctico los últimos 30 minutos de la clase. Este no es evaluativo pero recibe retroalimentación del profesor durante la sesión.

Semana	Semana 2
Objetivos de la semana	<ul style="list-style-type: none"> • Programar listas. • Programar tuplas. • Entender la sintaxis de los diccionarios y programarlos. • Realizar operaciones matriciales y vectoriales.
Descripción	Familiarizarse con los diferentes tipos de iterables (listas, tuplas y diccionarios). Además se repasarán conceptos de algebra lineal utilizando Python.
Evaluación	Taller práctico los últimos 30 minutos de la clase. Este no es evaluativo pero recibe retroalimentación del profesor durante la sesión.

Semana	Semana 3
Objetivos de la semana	<ul style="list-style-type: none"> • Programar condicionales: if, elif y else. • Programar loops (ciclos): for y while. • Identificar cuándo es necesario utilizar un loop o un condicional.
Descripción	Python es muy útil para automatizar procesos, pero para poder hacerlo es importante dominar los conceptos de Control Flow. A partir de condicionales podrá crear códigos contingentes para diferentes escenarios y con los loops podrá recorrer iterables para aplicar un mismo código. Esto es especialmente útil cuando necesite repetir tareas sobre diferentes elementos.
Evaluación	Taller práctico los últimos 30 minutos de la clase. Este no es evaluativo pero recibe retroalimentación del profesor durante la sesión.

Semana	Semana 4
Objetivos de la semana	<ul style="list-style-type: none"> • Entender el enunciado del Taller 1.
Descripción	Se presentará el enunciado del primer taller el cual se comenzará a resolver en clase para apoyar a los estudiantes resolviendo dudas.
Evaluación	Taller 1. Operaciones matriciales y vectoriales en Python. Se entrega la siguiente semana antes del comienzo de la siguiente sesión.

Semana	Semana 5
---------------	----------

Objetivos de la semana	<ul style="list-style-type: none"> • Aprender a programar funciones • Identificar cuándo es necesario construir funciones • Aprender a documentar funciones según los estándares de Google • Construir lambda functions • Utilizar el concepto de recursividad para construir funciones más eficientes.
Descripción	Las funciones permiten construir un código modular, más ordenado y fácilmente reproducible. Enseñaremos cómo dividir una tarea en varias funciones y como construir un pipeline de desarrollo adecuado.
Evaluación	Taller práctico los últimos 30 minutos de la clase. Este no es evaluativo pero recibe retroalimentación del profesor durante la sesión.

Semana	Semana 6
Objetivos de la semana	<ul style="list-style-type: none"> • Simular el comportamiento de un dado. • Identificar como partir un problema complejo en pequeños componentes para ser modelados. • Construir intervalos de confianza a partir de simulaciones. • Definir supuestos para una simulación.
Descripción	Se mostrará como se pueden combinar todos los conceptos aprendidos hasta el momento para responder una pregunta compleja: ¿Cuántos sobres de monas se deben comprar para llenar el álbum del mundial con 90% de confianza?
Evaluación	Taller práctico los últimos 30 minutos de la clase. Este no es evaluativo pero recibe retroalimentación del profesor durante la sesión.

Semana	Semana 7
Objetivos de la semana	<ul style="list-style-type: none"> • Construir dataframes • Filtrar filas y columnas según condiciones • Crear variables • Renombrar columnas • Calcular principales estadísticos de un dataframe • Importar datos externos: csv, exceles, txt, entre otros. • Pegar filas y columnas. • Pegar data frames: inner, left, right, outer joins • Exportar data frames • Group by • Manipulación de fechas
Descripción	Introducción a la librería pandas: Manipulación de bases de datos para su limpieza
Evaluación	Taller práctico los últimos 30 minutos de la clase. Este no es evaluativo pero recibe retroalimentación del profesor durante la sesión.

Semana	Semana 8
---------------	----------

Objetivos de la semana	<ul style="list-style-type: none"> Entender el enunciado del Taller 2.
Descripción	Se presentará el enunciado del segundo taller el cual se comenzará a resolver en clase para apoyar a los estudiantes resolviendo dudas.
Evaluación	Taller 2. Limpieza de bases de datos. Se entrega la semana 11

Semana	Semana 9 – SEMANA DE RECESO
Objetivos de la semana	
Descripción	
Evaluación	

Semana	Semana 10 – SEMANA SANTA
Objetivos de la semana	
Descripción	
Evaluación	

Semana	Semana 11
Objetivos de la semana	<ul style="list-style-type: none"> Construcción de scatter plots Construcción de line plots Construcción de bar plots Construcción de gráficos por grupos
Descripción	Introducción a las visualizaciones. Librerías matplotlib y seaborn
Evaluación	Taller práctico los últimos 30 minutos de la clase. Este no es evaluativo pero recibe retroalimentación del profesor durante la sesión.

Semana	Semana 12
Objetivos de la semana	<ul style="list-style-type: none"> Introducción a plotly Construcción de mapas para visualizar variables socioeconómicas Gráficos de densidad Identificación de hotspots
Descripción	Introducción a las visualizaciones espaciales e interactivas.
Evaluación	Taller práctico los últimos 30 minutos de la clase. Este no es evaluativo pero recibe retroalimentación del profesor durante la sesión.

Semana	Semana 13
Objetivos de la semana	<ul style="list-style-type: none"> Estimación de regresión por MCO de forma manual. Multiplicación de matrices. Cálculo de matriz var-cov.
Descripción	Introducción a la econometría en Python
Evaluación	Taller práctico los últimos 30 minutos de la clase. Este no es evaluativo pero recibe retroalimentación del profesor durante la sesión.
Sesión sincrónica	

Semana	Semana 14
Objetivos de la semana	<ul style="list-style-type: none"> • Estimación de regresión por MCO de forma manual. • Multiplicación de matrices. • Cálculo de matriz var-cov. • Aprender a especificar modelos • Aprender a exportar resultados • Aprender a crear gráficas de ajustes
Descripción	Introducción a la econometría en Python
Evaluación	Taller práctico los últimos 30 minutos de la clase. Este no es evaluativo pero recibe retroalimentación del profesor durante la sesión.

Semana	Semana 14
Objetivos de la semana	<ul style="list-style-type: none"> • Entender el enunciado del Taller 3.
Descripción	Se presentará el enunciado del tercer taller el cual se comenzará a resolver en clase para apoyar a los estudiantes resolviendo dudas.
Evaluación	Taller 3. Visualizaciones. Se entrega la siguiente semana antes del comienzo de la siguiente sesión.

Semana	Semana 15
Objetivos de la semana	<ul style="list-style-type: none"> • Entender los conceptos básicos de html • Aprender a hacer un get request • Identificar páginas estáticas y dinámicas • Hacer consultas específicas para construir tablas
Descripción	Introducción al web scraping en Python
Evaluación	Taller práctico los últimos 30 minutos de la clase. Este no es evaluativo pero recibe retroalimentación del profesor durante la sesión.

Semana	Semana 16
Objetivos de la semana	<ul style="list-style-type: none"> • Aprender como los computadores entienden el lenguaje • Aprender pipeline de limpieza: eliminar stopwords, lematización, stemming, homogenización de formato. • Construir nubes de palabras • Expresiones regulares
Descripción	Introducción al NLP
Evaluación	Taller práctico los últimos 30 minutos de la clase. Este no es evaluativo pero recibe retroalimentación del profesor durante la sesión.

Semana	Semana 17
Objetivos de la semana	<ul style="list-style-type: none"> • Entendimiento de los datos espaciales y sus diferentes formatos • Introducción a la librería geopandas • Visualización
Descripción	Introducción a los datos espaciales
Evaluación	

Semana	Semana 18
Objetivos de la semana	<ul style="list-style-type: none"> Entender el enunciado del Taller 4.
Descripción	Se presentará el enunciado del cuarto taller el cual se comenzará a resolver en clase para apoyar a los estudiantes resolviendo dudas.
Evaluación	Taller 4. Webscraping y limpieza de texto. Se entrega en la siguiente semana.

5. Metodología

Todas las sesiones comenzarán con 10 minutos de respuesta a preguntas y de socialización de respuesta de las tareas no evaluativas. Luego les seguirán 50 minutos de explicación del tema acompañados de ejercicios sincrónicos y finalmente la clase tendrá de 20 a 30 minutos de ejercicios para que los estudiantes resuelvan guiados en menor medida por el profesor. Las sesiones se realizarán por Zoom.

Nota: En caso de que alguna sesión se cruce con un festivo, se enviará el contenido de la clase y se hará un video explicativo para cubrir los temas de la sesión de forma asincrónica. Para la solución de las dudas habrá un foro y también horarios de atención.

6. Evaluaciones

El principal criterio para la evaluación del curso son los cuatro talleres que se comenzarán en clase y luego serán enviados para terminar autónomamente. La calificación final será determinada de la siguiente manera:

- Taller #1: 30%
- Taller #2: 20%
- Taller #3: 20%
- Taller #4: 20%
- Participación: 10%

Los talleres en clase se evaluarán teniendo en cuenta: 1) Organización del código/script, 2) Que el código genere un resultado correcto y 3) Número de errores en el código. Los talleres deben ser presentados a través de Bloque Neón. Para aprobar el curso cada estudiante debe alcanzar una nota igual o superior a 3.00. Las calificaciones definitivas de las materias serán numéricas de uno punto cinco (1,50) a cinco punto cero (5,00), en unidades, décimas y centésimas. Según los artículos 62 y 63 del Reglamento general de estudiantes de pregrado, el estudiante tendrá ocho días hábiles después de la entrega de la evaluación calificada para presentar un reclamo. El profesor responderá al reclamo en los diez días hábiles siguientes. Si el estudiante considera que la respuesta no concuerda con los criterios de evaluación podrá solicitar un segundo calificador al Consejo de la Facultad en los ochos días hábiles posteriores a la recepción de la decisión del profesor.