



Diplomado Finanzas Cuantitativas con Python

Cuerpo docente

- Doctor en Matemáticas Mauricio Labadie Martínez. Profesor de Asignatura, Facultad de Ciencias, UNAM. Trader e investigador cuantitativo (quant). Ponente Principal del Diplomado
- Maestro en Ciencia de Datos Omar Rodríguez Torres. CTO (Chief Technology Officer)

TEMARIO

MÓDULO I.- Repaso de Programación, Matemáticas y Finanzas.

TEMA	Subtemas	No. de horas	Ponente(s)
I. Introducción a Python.	I.1. Instalación de Python en una PC a través de la distribución Anaconda.	11	Principal: Omar Rodríguez Auxiliar: Mauricio Labadie

	<p>I.2. Tipos de variables: double, int, string, array, list, dict, etc.</p> <p>I.3. Condicionales e iteraciones: if-then-else, for, while, iteraciones para listas.</p> <p>I.3. Instalación de las librerías que vamos a usar: numpy, pandas, scipy, matplotlib, etc.</p>		
<p>II. Repaso de Matemáticas.</p>	<p>II.1. Álgebra Lineal I: espacios vectoriales, matrices, cambios de base, producto punto, valores y vectores propios.</p> <p>II.2. Cálculo III: curvas, diferencial, matriz hessiana, expansión de Taylor, multiplicadores de Lagrange.</p> <p>II.3. Probabilidad I: variables aleatorias, distribuciones, histogramas, momentos.</p> <p>II.4. Probabilidad II: probabilidad condicional, correlación, covarianza, ley de los grandes números, teorema de limite central.</p> <p>II.5. Inferencia Estadística: estimación paramétrica, máxima verosimilitud, pruebas de hipótesis, p-value.</p> <p>II.6. Procesos estocásticos I: movimiento browniano, ecuaciones diferenciales estocásticas, lema de Itô, movimiento browniano geométrico.</p>	<p>11</p>	<p>Principal: Mauricio Labadie</p> <p>Auxiliar: Omar Rodríguez</p>
<p>III. Repaso de Finanzas.</p>	<p>III.1. Matemáticas Financieras: tasas de interés, valuación de proyectos de inversión.</p> <p>III.2. Mercados Financieros y Valuación de Instrumentos: diferentes tipos de mercados (financieros, de dinero, de divisas,</p>	<p>11</p>	<p>Principal: Mauricio Labadie</p> <p>Auxiliar: Omar Rodríguez</p>

de capitales), cociente de Sharpe, portafolios de Markowitz, frontera eficiente.

III.3. Métodos Cuantitativos en Finanzas: Capital Asset Pricing Model (CAPM), valor en riesgo (VaR), introducción a instrumentos financieros derivados.

MÓDULO II.- Análisis estadístico en Python

TEMA	Subtemas	No. de horas	Ponente(s)
IV. Visualización de distribuciones en Python.	<p>IV.1. Simular variables aleatorias en Python.</p> <p>IV.2. Crear histogramas de variables aleatorias.</p> <p>IV.3. Crear un prototipo de análisis de distribuciones.</p> <p>IV.4. Crear un prototipo del Test de Normalidad de Jarque-Bera.</p>	11	Principal: Mauricio Labadie Auxiliar: Omar Rodríguez
V. Funciones y clases en Python.	<p>V.1. Principios APIE de la programación Orientada Objeto: abstracción, polimorfismo, herencia, encapsulación.</p> <p>V.2. Filosofía de la refactorización: como evitar repetir código.</p> <p>V.3. Usar funciones y clases en Python para refactorizar los prototipos en unas cuantas líneas de código.</p>	11	Principal: Mauricio Labadie Auxiliar: Omar Rodríguez
VI. Datos de mercado reales y plots en Python.	<p>VI.1. Descargar series de tiempo de precios de mercado a través de Yahoo Finance.</p>	11	Principal: Mauricio Labadie

	<p>VI.2. Tablas de datos: aprender a usar los dataframes de la librería pandas.</p> <p>VI.3. Crear plots de las series de tiempos de precios.</p> <p>VI.4. Crear series de tiempo de rendimientos.</p> <p>VI.5. Adaptar los prototipos de distribuciones y de Test de Normalidad a series de tiempo reales.</p>		<p>Auxiliar: Omar Rodríguez</p>
VII. Examen parcial 1.		11	<p>Principal: Mauricio Labadie</p> <p>Auxiliar: Omar Rodríguez</p>

MÓDULO III. Capital Asset Pricing Model (CAPM).

TEMA	Subtemas	No. de horas	Ponente(s)
VIII. Alphas, betas y Capital Asset Pricing Model (CAPM) en Python.	<p>VIII.1. Dedución del CAPM a través de la regresión lineal de un activo con respecto a un mercado dado.</p> <p>VIII.2. Interpretación geométrica y financiera de los coeficientes de la regresión lineal del CAPM: alpha, beta, epsilon.</p> <p>VIII.3. Analizar la geometría de diferentes estrategias de inversión en el espacio alpha - beta - epsilon.</p> <p>VIII.4. Crear un prototipo de cálculo del CAPM para un activo dado.</p>	11	<p>Principal: Mauricio Labadie</p> <p>Auxiliar: Omar Rodríguez</p>

<p>IX. Algoritmos de optimización y de cobertura de portafolios en Python.</p>	<p>IX.1. Comprender los conceptos de delta-neutralidad y beta-neutralidad para portafolios de inversión.</p> <p>IX.2. Solución exacta de cobertura delta y beta neutral de un portafolios dado mediante Álgebra Lineal.</p> <p>IX.3. Entender las limitaciones de la solución exacta desde el punto de vista financiero.</p> <p>IX.4. Proponer una solución no exacta pero financieramente aceptable para la cobertura de un portafolios.</p> <p>IX.5. Aprender a usar la librería <code>scipy.optimize.minimize</code> para minimización sin restricciones.</p> <p>IX.6. Crear un prototipo de cobertura de portafolios de inversión.</p>	<p>11</p>	<p>Principal: Mauricio Labadie</p> <p>Auxiliar: Omar Rodríguez</p>
<p>X. Inversión por factores (Factor Investing) en Python.</p>	<p>X.1. Comprender el concepto financiero de factores o "betas" relativos a un activo o portafolios.</p> <p>X.2. Utilizar los factores más comunes para analizar un activo o portafolios: growth, value, momentum, etc.</p> <p>X.3. Crear un prototipo que, dada una lista de factores, calcule los factores de cualquier activo que se desee.</p>	<p>11</p>	<p>Principal: Mauricio Labadie</p> <p>Auxiliar: Omar Rodríguez</p>
<p>XI. Examen parcial 2.</p>		<p>11</p>	<p>Principal: Mauricio Labadie</p> <p>Auxiliar: Omar Rodríguez</p>

MÓDULO IV. Portafolios de Markowitz y Frontera Eficiente

TEMA	Subtemas	No. de horas	Ponente(s)
XII. Matriz de Varianza-Covarianza en Python.	<p>XII.1. Calcular y analizar la matriz de varianza-covarianza.</p> <p>XII.2. Demostrar el Teorema de Diagonalización de Matrices Simétricas.</p> <p>XII.3. Interpretar los valores propios máximo y mínimo de la matriz de varianza-covarianza como "portafolios especiales".</p> <p>XII.4. Crear un prototipo de cálculo y análisis de la matriz de varianza-covarianza.</p>	11	<p>Principal: Mauricio Labadie</p> <p>Auxiliar: Omar Rodríguez</p>
XIII. Análisis de Componentes Principales (PCA) en Python.	<p>XIII.1. Interpretar la varianza explicada como información que el modelo preserva.</p> <p>XIII.2. Usar el concepto de información para reducir las dimensiones de la matriz de varianza-covarianza.</p> <p>XIII.3. Crear un prototipo del PCA en función del porcentaje mínimo de varianza explicada del modelo.</p>	11	<p>Principal: Mauricio Labadie</p> <p>Auxiliar: Omar Rodríguez</p>
XIV. Examen parcial 3.		11	<p>Principal: Mauricio Labadie</p> <p>Auxiliar: Omar Rodríguez</p>
XV. Optimización de portafolios y Asset Allocation en Python.	XV.1. Portafolios de inversiones "clásicos": máxima varianza, mínima varianza, equi-ponderado, ponderado por volatilidad, etc.	11	<p>Principal: Mauricio Labadie</p>

	<p>XV.2. Aprender a usar la librería <code>scipy.optimize.minimize</code> para minimización con restricciones.</p> <p>XV.3. Crear un prototipo para que, dada una lista de activos y un portafolios clásico deseado, tengamos los pesos o alocación de dicho portafolios.</p>		<p>Auxiliar: Omar Rodríguez</p>
<p>XVI. Portafolios de Markowitz y la “Modern Portfolio Theory” en Python.</p>	<p>XVI.1. Repasar los conceptos de optimización con restricciones y multiplicadores de Lagrange.</p> <p>XVI.2. Interpretar al portafolios de Markowitz como solución a un problema de minimización de varianza con restricciones sobre el rendimiento.</p> <p>XVI.3. Crear un prototipo que, dado un rendimiento esperado, calcule el portafolios de Markowitz correspondiente.</p>	<p>11</p>	<p>Principal: Mauricio Labadie</p> <p>Auxiliar: Omar Rodríguez</p>
<p>XVII. Frontera Eficiente en Python.</p>	<p>XVII.1 Entender la geometría de los portafolios de inversiones en el espacio volatilidad-rendimiento.</p> <p>XVII.2. Entender la frontera eficiente como todos los posibles portafolios de Markowitz.</p> <p>XVII.3. Usar la Frontera Eficiente para optimizar o “mejorar” un portafolios dado.</p> <p>XVII.4. Crear un prototipo de visualización de la frontera eficiente y de portafolios concretos.</p>	<p>11</p>	<p>Principal: Mauricio Labadie</p> <p>Auxiliar: Omar Rodríguez</p>
<p>XVIII. Examen parcial 4.</p>		<p>11</p>	<p>Principal: Mauricio Labadie</p> <p>Auxiliar: Omar Rodríguez</p>

MÓDULO V. Monte Carlo y Derivados en Python

TEMA	Subtemas	No. de horas	Ponente(s)
XIX. Simulaciones de Monte Carlo en Python.	<p>XIX.1. Ley de los grandes números y el teorema de limite central.</p> <p>XIX.2 Intervalos de confianza.</p> <p>XIX.3. Interpretación geométrica de procesos estocásticos usando intervalos de confianza.</p> <p>XIX.4. Definir las simulaciones de Monte Carlo en términos de intervalos de confianza.</p> <p>XIX.5. Crear un prototipo de Monte Carlo que simule trayectorias aleatorias y calcule los intervalos de confianza de la media.</p>	11	<p>Principal: Mauricio Labadie</p> <p>Auxiliar: Omar Rodríguez</p>
XX. Valuación de opciones en Python.	<p>XX.1. Deducción de la Formula de Black-Scholes para valuación de opciones europeas.</p> <p>XX.2. Definición de las "griegas" o sensibilidades.</p> <p>XX.3. Utilización de las griegas para la cobertura de un portafolios de opciones.</p> <p>XX.4. Crear un prototipo de valuación de opciones con dos versiones: solución exacta usando la fórmula de Black-Scholes y solución aproximada con intervalos de confianza usando simulaciones de Monte Carlo.</p>	11	<p>Principal: Mauricio Labadie</p> <p>Auxiliar: Omar Rodríguez</p>
XXI. Examen parcial 5.		11	<p>Principal: Mauricio Labadie</p>

			Auxiliar: Omar Rodríguez
XXII. Productos estructurados en Python.	<p>XXII.1. Definición de productos estructurados.</p> <p>XXII.2. Ejemplo de productos estructurados: opciones barrera, Credit Default Swaps (CDS).</p> <p>XXII.3. Simulación de productos estructurados dependientes de la trayectoria.</p> <p>XXII.4. Crear dos prototipos: pricing de opción barrera y pricing de CDS.</p>	11	<p>Principal: Mauricio Labadie</p> <p>Auxiliar: Omar Rodríguez</p>