



Una revisión de experiencias y recursos educativos para aprender economía y finanzas con Python

Francisco Salas-Molina¹, David Pla-Santamaria², Ana García-Bernabeu³ y Natalia Utrero-González⁴

¹ Universitat Politècnica de València, frasamo@upv.es

² Universitat Politècnica de València, dplasan@upv.es

³ Universitat Politècnica de València, angarber@upv.es

⁴ Universitat Politècnica de València, nmutrgon@upv.es

How to cite: F. Salas-Molina, D. Pla-Santamaria, A. García-Bernabeu y N. Utrero-González. 2023. Una revisión de experiencias y recursos educativos para aprender economía y finanzas con Python. En libro de actas: *IX Congreso de Innovación Educativa y Docencia en Red*. Valencia, 13 – 14 de julio de 2023. <https://doi.org/10.4995/INRED2023.2023.16511>

Abstract

In this paper, we review different educational experiences in which programming language Python is used to teach/learn economics and finance. The emphasis is placed on the resources used and the challenges faced by teachers. A critical analysis on the advantages and disadvantages is provided to facilitate the choice of the best teaching methods in the context of quantitative economics and finance. As a result, the paper summarizes how economics and finance can be taught using Python. Furthermore, the paper contributes to enhance the knowledge about computer driven economic learning and to close the gap between computer science and economics.

Keywords: Python, programming, learning, economics, finance.

Resumen

En este artículo revisamos diferentes experiencias educativas en las que se utiliza el lenguaje de programación Python para enseñar/aprender economía y finanzas. El énfasis está puesto en los recursos utilizados y los desafíos a los que se enfrentan los docentes. Se realiza un análisis crítico de las ventajas y desventajas para facilitar la elección de los mejores métodos de enseñanza en el contexto de la economía y las finanzas. Como resultado, el documento resume cómo se puede enseñar economía y finanzas usando Python. Además, el trabajo contribuye a mejorar el conocimiento sobre el aprendizaje económico basado en la tecnología informática y a cerrar la brecha entre la informática y la economía.

Keywords: Python, programación, aprendizaje, economía, finanzas.

1 Introducción

Uno de los ámbitos de estudios universitarios más importantes por número de estudiantes es el de economía y finanzas. Según la *Estadística de estudiantes* publicada por el Ministerio de Universidades de España (Ministerio de Universidades, 2022), el número total de estudiantes en el curso 2021-2022 de *Economía y afines*, que comprende grados en *Administración y gestión de empresas*, *Economía*, y *Otra educación comercial y empresarial*, fue de 202,198 estudiantes. Solamente el ámbito de *Administración y gestión de empresas*, con 132,151 estudiantes, es el primero en número de estudiantes por encima de *Ingeniería*, con 131,690 estudiantes, y *Derecho*, con 109,934 estudiantes. En la Comunidad Valenciana, el número total de estudiantes del ámbito de *Economía y afines* fue de 17,366, lo que representa un 8.6 % del total nacional siendo la tercera comunidad autónoma en número de estudiantes por detrás de Madrid, Cataluña y Andalucía.

En el contexto de la economía y las finanzas, el aprendizaje basado en el uso intensivo de la tecnología informática generalmente se considera una metodología innovadora que proporciona a los estudiantes unas habilidades muy deseadas por los empleadores (Kuroki, 2021). Esto es así porque tanto la economía como las finanzas son cada vez más campos interdisciplinarios que involucran no solo los aspectos sociales sino también múltiples aspectos cuantitativos cuya comprensión solo es posible mediante herramientas propias de la informática. Algunos ejemplos de las habilidades requeridas son el pensamiento computacional, la capacidad analítica o la simulación de posibles escenarios.

Una forma adecuada de introducir el pensamiento computacional en la instrucción de estudiantes de economía y finanzas es el uso de un lenguaje de programación. Aunque los economistas dedican una cantidad considerable de su tiempo a codificar y ejecutar software (Aruoba & Fernández-Villaverde, 2015; Coleman et al., 2021; Duarte et al., 2020), la herramienta predominante en las clases prácticas de muchas universidades sigue siendo la hoja de cálculo. Sin embargo, algunos lenguajes de programación como Python están empezando a ser utilizados tanto en la enseñanza/aprendizaje de economía y finanzas (Jenkins, 2022; Kuroki, 2021). Python es un lenguaje de programación de propósito general creado en 1989 por Guido van Rossum. Según Stack Overflow Trends (Stackoverflow, 2022), Python es el lenguaje más popular en términos del porcentaje de preguntas que hacen los desarrolladores sobre los problemas de programación que encuentran en su trabajo. Entre las razones de su popularidad, se puede destacar que Python es gratuito, de código abierto y apto para principiantes.

El resultado principal de este trabajo es una revisión sistemática de las experiencias educativas y otros recursos sobre el uso de Python como herramienta tecnológica para el aprendizaje de la economía y las finanzas a nivel universitario. Además de servir como referencia para futuras investigaciones, el trabajo de revisión nos ayudará a identificar cuáles son los elementos comunes en estas experiencias, y cuáles son los problemas a los que se han enfrentado los docentes en su implementación.

Además de esta introducción, la estructura de este documento es la siguiente. La Sección 2 plantea los objetivos de este trabajo. La Sección 3 presenta una revisión sistemática de diferentes experiencias educativas. La Sección 4 contiene recursos, un análisis de ventajas e inconvenientes y acciones para resolver las dificultades identificadas. Finalmente, la Sección 5 concluye este artículo.

2 Objetivos

Para contribuir a una mejor comprensión de cómo se puede enseñar economía y finanzas utilizando Python, este trabajo pretende identificar aquellas experiencias educativas en las que se haya hecho uso de Python como lenguaje de programación para mejorar el aprendizaje en el ámbito de la economía y las finanzas en el contexto de la enseñanza universitaria. De este objetivo principal, se derivan dos objetivos adicionales como son: 1) analizar las ventajas e inconvenientes identificados en las experiencias recogidas; y 2) poner a disposición de los docentes interesados una serie de recursos sobre el uso de Python en el aprendizaje de economía y finanzas.

3 Desarrollo de la innovación: una revisión sistemática

En esta sección, revisamos las principales características de las experiencias educativas y los recursos útiles existentes en la literatura sobre el uso de Python como lenguaje de programación en el contexto de la economía y las finanzas. Para ello, seguimos la metodología propuesta por Medina López et al. (2010) basada en cinco pasos: 1) identificación del campo de estudio y período a analizar; 2) selección de las fuentes de información; 3) realización de la búsqueda; 4) gestión y depuración de los resultados; y 5) análisis de los resultados.

3.1 Identificación del campo de estudio y período a analizar

El campo que será objeto de estudio en esta revisión sistemática es el uso del lenguaje de programación Python en un contexto de enseñanza y aprendizaje de la economía y las finanzas en la enseñanza universitaria. A partir de la identificación del campo de estudio, se plantean los siguientes términos clave (*keywords*) en inglés: *Python, economics, finance, education, learning*.

Cuando la información disponible así lo permita, se indicará si dentro de la enseñanza universitaria un determinado trabajo está más enfocado a estudios de grado o a estudios de postgrado. En relación con el período analizado, cabe indicar que Python apareció en 1989 con lo que la descripción de cualquier experiencia educativa utilizando este lenguaje de programación ha de ser necesariamente posterior a ese año y se extenderá hasta el año 2022. En realidad, todos los trabajos identificados son posteriores al año 2000.

3.2 Selección de las fuentes de información

Una vez establecido el campo de estudio y el período a analizar, debemos considerar las diferentes fuentes de información. Con el objetivo acotar la búsqueda a un entorno académico contrastado, las fuentes de información utilizadas se limitan a artículos de revistas, actas de congresos, webs educativas y libros relacionados con el campo de estudio. No se establecen distinciones en relación a la indexación de las revistas en las bases de datos.

3.3 Realización de la búsqueda

Los criterios de búsqueda seleccionados para la realización de la búsqueda en bases de datos como Web of Science (www.webofscience.com) y Google Scholar (scholar.google.es) son los siguientes:

1. La presencia del término *Python* en el título o subtítulo de la publicación.
2. Además, la presencia de términos en el título que acoten el campo de estudio como *economics*, *finance*, *economist*, *financial*, *econometrics*.
3. Además, que traten sobre experiencias educativas, metodologías docentes, recursos para el aprendizaje.

La realización de una búsqueda automática que incluya los tres criterios de búsqueda anteriores entraña algunas dificultades técnicas que acabarían por descartar numerosas publicaciones que consideramos relevantes. Por este motivo, se ha optado por realizar una búsqueda manual de publicaciones que traten sobre experiencias educativas, metodologías docentes, y recursos para el aprendizaje de la economía y las finanzas entre todos los resultados de una búsqueda en bases de datos que incluyan en el título las palabras clave mencionadas en los puntos 1 y 2.

3.4 Gestión y depuración de los resultados

Además de lo mencionado en la sección anterior respecto a la temática educativa, el proceso de gestión y depuración de las publicaciones identificadas incluye la clasificación de las publicaciones obtenidas atendiendo a los siguientes criterios:

- Tipo: artículo, congreso, libro, web.
- Nivel: grado o postgrado.
- Experiencia educativa: sí o no, en función de si se describe o no una experiencia educativa en la publicación.

De acuerdo con la estrategia de búsqueda y los criterios de clasificación establecidos, la Tabla 1 recoge las publicaciones identificadas. La tabla se ordena en primer lugar por el hecho de recoger alguna experiencia educativa en el ámbito de la economía o las finanzas y, posteriormente, por año de publicación.

3.5 Análisis de los resultados

Una vez identificadas las publicaciones sobre economía y finanzas que utilizan Python como recurso de aprendizaje, en esta sección revisamos las principales características de las referencias con un doble objetivo. En primer lugar, conocer con mayor profundidad en qué medida se está utilizando el lenguaje de programación Python en el ámbito de la economía y las finanzas a nivel educativo de grado y postgrado (incluyendo en este último nivel la investigación). En segundo lugar, establecer el estado del arte de las innovaciones o experiencias educativas innovadoras que se basan en el uso de Python para el aprendizaje de la economía y las finanzas.

Teniendo en cuenta que Python se creó en 1989, el tiempo transcurrido hasta que aparecen primeros trabajos que combinan Python y la economía en un sentido amplio es notable. De hecho hasta 2009,

Tabla 1: Comparativa de las publicaciones sobre economía y finanzas con Python.

Referencia	Tipo	Nivel	Experiencia educativa
Pugh (2014)	Congreso	Postgrado	Sí
Salas-Molina y Pla-Santamaria (2018)	Artículo	Postgrado	Sí
Kuroki (2021)	Artículo	Grado	Sí
Jenkins (2022)	Artículo	Grado	Sí
Choirat y Seri (2009)	Artículo	Grado	No
Stachurski (2009)	Libro	Postgrado	No
Fletcher y Gardner (2009)	Libro	Postgrado	No
Sargent y Stachurski (2014)	Web	Postgrado	No
Hilpisch (2014)	Libro	Postgrado	No
Hilpisch (2015)	Libro	Postgrado	No
Hilpisch (2020)	Libro	Postgrado	No
Hilpisch (2021)	Libro	Postgrado	No
Yan (2014)	Libro	Postgrado	No
Weiming (2015)	Libro	Postgrado	No
Salas-Molina y Pla-Santamaria (2017)	Libro	Postgrado	No
De Prado (2018)	Libro	Postgrado	No
Oosterlee y Grzelak (2019)	Libro	Postgrado	No
Dixon et al. (2020)	Libro	Postgrado	No
Sarmas et al. (2020)	Libro	Postgrado	No
Hilpisch (2021)	Libro	Postgrado	No
Sheppard (2021)	Libro	Postgrado	No
Coleman et al. (2021)	Artículo	Postgrado	No
Kelliher (2022)	Libro	Postgrado	No

no aparecen los primeros trabajos de Python aplicado en la economía como Choirat y Seri (2009) y Stachurski (2009). Choirat y Seri (2009) proponen el uso de Python en el área de econometría como una nueva alternativa a otros lenguajes de programación como R (Cribari-Neto & Zarkos, 1999), Octave (Eddelbuettel, 2000) y Pearl (Baiocchi, 2003). Los autores presentan conceptos básicos de Python e ilustran su uso mediante un ejemplo de simulación de Monte Carlo (Glasserman, 2004). Este documento no describe una experiencia educativa, sino que se refiere a cómo los economistas pueden utilizar un nuevo lenguaje de programación con sus ventajas e inconvenientes. Por otro lado, Stachurski (2009) propone el uso de Python como herramienta clave para estudiantes de posgrado o investigadores en economía cubriendo una amplia gama de temas en dinámica económica como simulación, teoría de la estabilidad y programación dinámica. Finalmente, aunque Sheppard (2021) se centra en el uso de Python para la investigación, el autor afirma que el trabajo también puede ser útil para estudiantes que requieran una plataforma versátil para econometría, estadística o análisis numérico general.

Sargent y Stachurski (2014) crearon un proyecto de aprendizaje en línea particularmente interesante que presenta una serie completa de lecciones sobre economía cuantitativa utilizando los lenguajes de programación Python y Julia. En el momento de escribir este artículo, la versión actual de la web incluye 150 lecciones agrupadas en tres áreas principales: 1) Programación en Python para economía y finanzas (19 lecciones); 2) Economía cuantitativa con Python (81 lecciones); y 3) Economía cuantitativa avanzada con Python (50 lecciones).

En Fletcher y Gardner (2009), encontramos también una de las primeras propuestas de uso de Python para el modelado financiero. Este libro no está enfocado a una audiencia educativa sino más bien a analistas cuantitativos y científicos. Otro grupo de manuales enfocados principalmente a una audiencia educativa son los escritos por Hilpisch (2014, 2015, 2020, 2021). Hilpisch (2014) proporciona ejemplos de grandes instituciones financieras como Bank of America y Merrill Lynch como usuarios estratégicos de Python. Además, el autor también afirma que muchos programas de postgrado en finanzas usan Python para recorrer el camino de la teoría financiera a la práctica. El enfoque del autor es particularmente interesante por la importancia que confiere al aspecto práctico en el que la implementación viene antes que los detalles teóricos. Hilpisch (2014) cubre conceptos básicos como instalación, tipos de datos y estructuras de datos, herramientas matemáticas, visualización, operaciones de entrada/salida, integración de Excel y varios temas financieros como series temporales, optimización de carteras, simulación Monte Carlo (Glasserman, 2004), y valoración de derivados. Por otro lado, Hilpisch (2015) está centrado en el análisis de productos derivados, Hilpisch (2020) analiza el papel clave que la inteligencia artificial tiene en las finanzas, y Hilpisch (2021) es una introducción básica a la teoría financiera en el que se vincula el aspecto teórico financiero, con el concepto analítico o matemático y, finalmente, con la versión práctica mediante su implementación con Python. Otros libros introductorios similares que cubren varios temas sobre Python para finanzas son Weiming (2015) y Yan (2014) y Lewinson (2022).

La relación de la inteligencia artificial y el aprendizaje automático con la economía y las finanzas es cada vez más evidente. Como menciona Gogas y Papadimitriou (2021), el aprendizaje automático aplicado a problemas económicos se remonta al trabajo de Lee y Lee (1974). Más recientemente, se pueden encontrar ejemplos del uso del aprendizaje automático en un contexto de investigación o de la industria financiera en De Prado (2018). En este libro, el autor afirma que el aprendizaje automático financiero ahora debe considerarse como un tema distinto y que, algún día, el aprendizaje automático dominará las finanzas hasta el punto en que invertir ya no significará apostar. Además, el autor proporciona el código de Python que utilizó para resolver la mayoría de los problemas financieros que se recogen en el libro. Oosterlee y Grzelak (2019) también combinan la modelización matemática y las finanzas computacionales con numerosos ejercicios tanto en Python como en Matlab. Otro libro escrito para estudiantes de posgrado avanzados y académicos en el campo del aprendizaje automático para finanzas cuantitativas es Dixon et al. (2020). Los autores presentan el aprendizaje automático como una extensión de varios temas de la economía cuantitativa, como la econometría financiera y la programación dinámica, con énfasis en representaciones algorítmicas novedosas de datos, regularización y otras técnicas predictivas. Los autores también proporcionan ejemplos de código de Python para ayudar a los lectores a comprender las metodologías y las aplicaciones.

Desde el trabajo inicial de Markowitz (1952), el problema de la selección de carteras está presente en muchos programas de investigación y formación en gestión financiera. El lector interesado puede encontrar diferentes enfoques para resolver el problema de selección de carteras usando Python tanto en Hilpisch (2014) como en Sargent y Stachurski (2014). Una contribución reciente a este tema es Sarmas et al. (2020), en la que los autores presentan el pseudocódigo de una serie de métodos de análisis de decisión multicriterio para abordar el problema de selección de carteras.

Aunque el foco está puesto en los métodos multicriterio en lugar de la implementación de Python, los autores dedican un capítulo a la implementación de los algoritmos propuestos en Python y proporcionan una referencia¹ a un repositorio de GitHub para el código fuente. Otra propuesta reciente de Kelliher (2022) cubre distintos aspectos de las finanzas cuantitativas y gestión de inversiones. El objetivo principal del libro es ayudar al lector a comprender la teoría financiera subyacente mediante la aplicación de conceptos teóricos en Python.

En este punto centramos nuestra atención en las experiencias educativas que relatan el uso del lenguaje de programación Python en el ámbito del aprendizaje de la economía y las finanzas. Las primeras experiencias educativas fueron descritas por Ketcheson (2014)² y Pugh (2014)³ en la *13th Python in Science Conference*. Aunque la propuesta de Ketcheson (2014) no está dirigida específicamente a estudiantes de economía y finanzas (por este motivo no está recogida en la Tabla 1), es conveniente mencionarla por su relación directa con el objetivo de relatar experiencias educativas con Python en el ámbito universitario. El trabajo plantea un curso de métodos numéricos con Notebooks como medio para combinar matemáticas, explicaciones en texto, código y visualización en un contexto interactivo. Sobre el aspecto educativo, el autor parte de la premisa de que los estudiantes aprenden haciendo más que escuchando cosas. Posteriormente sugiere el uso del aprendizaje reflexivo (Ernst et al., 2017; Scott et al., 2018) como la forma de acción más apropiada para enseñar métodos numéricos. Después de describir las principales características sobre el uso de Notebooks de Python (Pérez & Granger, 2007; Rossant, 2014), el autor comparte algunas lecciones aprendidas:

- Los estudiantes deben escribir código desde el principio.
- El profesor debe ayudar a los alumnos a descubrir conceptos por sí mismos.
- El profesor debe adaptar la dificultad al nivel de los estudiantes.
- El profesor debe aumentar gradualmente la complejidad.

Finalmente, Ketcheson (2014) también señala algunos inconvenientes como la dificultad para cubrir todo el material debido al tiempo requerido para los cursos que involucran programación, la imposibilidad de que el instructor interactúe individualmente si el número de estudiantes es mayor a 20, y algunos problemas de instalación del programa necesario para desarrollar el curso.

Por otro lado, el trabajo de Pugh (2014) es una propuesta específica sobre el uso de Python para la investigación y la enseñanza de la economía. El autor argumenta que, junto con la teoría y la experimentación, el modelado computacional y la simulación se han convertido en el tercer pilar de la investigación científica. El autor propone un curso⁴ sobre métodos numéricos diseñado para estudiantes e investigadores de posgrado en la Escuela de Economía de la Universidad de Edimburgo. El autor recalca el hecho de que los libros de texto estándar de macroeconomía y microeconomía a nivel de posgrado, como Mas-Colell et al. (1995) y Romer (2012), no mencionan que se necesitan métodos numéricos para resolver incluso los modelos económicos básicos. Una motivación adicional para el uso de métodos numéricos en economía es que la mayoría de los modelos económicos, en particular los modelos económicos dinámicos, presentan no linealidades y restricciones vinculan-

¹<https://github.com/epu-ntua/Multicriteria-Portfolio-Construction-with-Python>

²<http://www.youtube.com/watch?v=OaP6LiZuaFM>

³<http://www.youtube.com/watch?v=xHkGW115X8k>

⁴<https://github.com/davidrpugh/numerical-methods>

tes que los hacen analíticamente intratables. El curso sobre métodos computacionales propuesto por el autor está organizado en torno a tres componentes principales: un conjunto de sesiones de laboratorio, un seminario intensivo de Python de una semana de duración y un curso avanzado de formación de doctorado. El autor reconoce el uso de material de Sargent y Stachurski (2014) para dos sesiones introductorias y concluye afirmando que se sorprendió por el entusiasmo de los estudiantes de posgrado en economía por aprender métodos computacionales.

Un aspecto de la gestión financiera que no está cubierto en los trabajos de Sargent y Stachurski (2014) y Hilpisch (2014) es el problema de la gestión de efectivo. La gestión de efectivo tiene como objetivo encontrar un equilibrio entre lo que se mantiene en efectivo y lo que se asigna a otras inversiones a cambio de un rendimiento. En Salas-Molina y Pla-Santamaria (2018), los autores describen una experiencia educativa sobre el uso de Python para la gestión de efectivo en un programa de Máster de Administración de Empresas. Los autores proponen un método de enseñanza basado en la implementación (codificación) del modelo económico basado en los siguientes pasos: 1) seleccionar el concepto a enseñar; 2) definir el modelo que describe la relación entre las variables de interés dentro del concepto seleccionado; 3) codificar o implementar el modelo; y 4) experimentar con la implementación del modelo. Los autores también refieren diferentes problemas de instalación por parte de los alumnos y la dificultad de abordar un tema novedoso como es el problema del manejo de efectivo con una nueva herramienta como los Notebooks.

Otra experiencia educativa en el contexto de la enseñanza de la microeconomía ha sido descrita por Kuroki (2021). El autor utiliza Google Colaboratory (Google, 2022) como una forma de evitar problemas de instalación informados por Ketcheson (2014) y Salas-Molina y Pla-Santamaria (2018). De hecho, Google Colaboratory permite a los usuarios escribir código Python en un navegador web con el único requisito de crear una cuenta de usuario y sin preocuparse por los problemas de instalación. El autor cubre temas de optimización avanzada como la maximización de la utilidad, el equilibrio general, la minimización de costes, el mercado competitivo, el monopolio, la discriminación de precios, el modelo de Cournot, el modelo de Stackelberg y el modelo de Bertrand. El curso está dirigido a estudiantes de grado de teoría microeconómica. Una de las cosas más interesantes de la propuesta de Kuroki (2021) es la discusión sobre los motivos por los que tanto profesores como estudiantes deberían aprender Python. En relación con los profesores, el autor argumenta que Python es un recurso de código abierto que puede reducir los costes para los estudiantes y los departamentos. Además, los profesores pueden complementar la metodología tradicional de papel y lápiz con un enfoque computacional como método alternativo para resolver problemas de optimización. Por el lado de los estudiantes, el autor sugiere que las habilidades de programación pueden hacer que los estudiantes puedan encontrar trabajo con mayor facilidad después de graduarse, ya que muchos trabajos en finanzas valoran la competencia computacional. En la misma línea que Sargent y Stachurski (2014), el autor afirma que Python parece estar reemplazando el uso de las hojas de cálculo en muchas empresas de banca y finanzas.

En Jenkins (2022) se propone otro curso⁵ de nivel de grado en macroeconomía computacional basado en Python. El autor afirma que los estudiantes inscritos en el curso trabajaron conceptos como la optimización dinámica y la simulación de modelos dinámicos lineales, procesos estocásticos básicos, modelos de ciclos económicos reales y modelos de ciclos económicos nekeynesianos. El curso no presupone experiencia previa en programación porque los estudiantes realizan una introducción a la programación en Python. El autor recuerda que el objetivo no es que los estudiantes de economía se conviertan en programadores informáticos sino que adquieran las habilidades necesarias

⁵<https://github.com/letsgoexploring/computational-macroeconomics>

para resolver problemas específicos. Los estudiantes también se familiarizaron con las bibliotecas de Python e hicieron un uso extensivo de Notebooks. En la misma línea que Kuroki (2021), el autor se refiere a alguna evidencia sobre la posibilidad de obtener un mejor salario por aquellos estudiantes con conocimientos de programación informática.

Finalmente, aunque no corresponde al área de economía y finanzas, Gutiérrez et al. (2016) describen un proyecto de innovación educativa en ingeniería naval que puede servir para complementar el trabajo de revisión de experiencias educativas en educación superior con Python. El objetivo de este proyecto es mejorar los resultados de aprendizaje de los estudiantes, incorporando metodologías activas de aprendizaje como es la programación en el lenguaje Python. Los autores esperan que este tipo de metodologías favorezcan la interdisciplinariedad y la coordinación curricular de tipo horizontal y vertical así como desarrollar un conjunto de recursos útiles para el aprendizaje. A modo de resumen, la Tabla 2 recoge las experiencias educativas revisadas así como las conclusiones extraídas por los autores.

Tabla 2: Experiencias educativas con Python.

Referencia	Nivel	Área	Lecciones aprendidas
Ketcheson (2014)	Postgrado	Matemáticas	Escribir código desde el principio y adaptar la dificultad a los alumnos.
Pugh (2014)	Postgrado	Economía	Los estudiantes muestran deseo por aprender modelos computacionales
Gutiérrez et al. (2016)	Grado	Ingeniería naval	Fomento de la interdisciplinariedad y la coordinación curricular
Salas-Molina y Pla-Santamaria (2018)	Postgrado	Finanzas	Aplicar Python a un modelo conocido para facilitar el aprendizaje
Kuroki (2021)	Grado	Microeconomía	Colaboratory para evitar problemas de instalación
Jenkins (2022)	Grado	Macroeconomía	Los alumnos valoraron positivamente el curso según una encuesta.

4 Resultados

En esta sección se presentan los resultados principales que se derivan de la revisión sistemática realizada en la sección anterior. Más concretamente, se recoge una serie de recursos útiles para profesores interesados en utilizar Python en el aprendizaje de la economía y las finanzas, se realiza un análisis de las ventajas e inconvenientes que se pueden plantear y, finalmente, se plantean algunas propuestas para tratar de resolver las dificultades identificadas.

Sin centrarse en ninguna disciplina académica, Guzdial (2008) establece que los educadores tienen la responsabilidad de hacer que el pensamiento computacional esté disponible para todos. Sin embargo, la enseñanza del pensamiento computacional puede requerir diferentes enfoques y responder a preguntas difíciles como el grado de comprensión informática que tienen los estudiantes que no

son informáticos (por ejemplo, aquellos matriculados en estudios de economía y finanzas), qué les resulta interesante, qué herramientas son más accesibles y cómo se deben impartir las clases. Planteando las ventajas y desventajas destacadas por las experiencias educativas revisadas en la Sección 3, a continuación intentamos dar algunas respuestas a las preguntas anteriores.

El pensamiento computacional es un tipo de pensamiento analítico (Wing, 2006, 2008). Como consecuencia, esta habilidad es de aplicación universal para todos, no solo para los informáticos. El pensamiento computacional no es más que un enfoque para resolver problemas, diseñar sistemas y comprender el comportamiento humano. En consecuencia, no cabe duda de que los estudiantes de economía y finanzas tienen que resolver problemas, diseñar sistemas y comprender el comportamiento humano como parte fundamental de su formación. Una forma de lograr este objetivo es mediante el uso de Python como lenguaje de programación en el aula.

Como primer resultado derivado de la revisión de la literatura, la Tabla 3 recoge algunos de los recursos disponibles que pueden resultar de utilidad para los profesores interesados:

Tabla 3: Recursos educativos de Python en economía y finanzas.

Referencia	Área	Elementos clave
Stachurski (2009)	Dinámica económica	Rigor metodológico, basado en ejemplos, capítulo introductorio a la programación.
Sargent y Stachurski (2014)	Economía cuantitativa	Amplia extensión, nivel avanzado, gratuito, acceso al código.
Hilpisch (2014)	Finanzas	Introductorio, generalista pero con numerosos ejemplos prácticos.
Hilpisch (2015)	Derivados	Avanzado, específico para profundizar en simulación y valoración de opciones.
Salas-Molina y Pla-Santamaria (2017)	Gestión tesorería	Introductorio, simulación de modelos y previsión series temporales.
De Prado (2018)	Finanzas	Avanzado, centrado en aprendizaje automático.
Hilpisch (2021)	Teoría financiera	Introductorio, basado en la conexión entre finanzas, matemáticas y Python.
Lewinson (2022)	Finanzas	Introductorio, incluye conceptos de aprendizaje automático.

Por otro lado, algunas de las ventajas del uso de Python que conviene recordar son las siguientes. Python es un lenguaje de programación de código abierto muy popular, fácil de aprender y accesible sin coste para los estudiantes (Guo, 2014). Otra clara ventaja para los estudiantes es la posibilidad de desarrollar un conjunto de habilidades que se valoran en muchas ofertas de trabajo (Kuroki, 2021). De manera similar, Jenkins (2022) cita a Blumenstyk (2016) indicando que poseer habilidades de programación informática aumenta los salarios de los recién graduados. Sin embargo, también advierte que un curso de microeconomía no es la mejor elección para aquellos estudiantes que deseen usar Python para el análisis de datos.

El uso de los Notebooks como una herramienta interactiva que combina código, texto, matemáticas, figuras, imágenes, videos y animaciones es recomendado por Ketcheson (2014), Kuroki (2021) y Salas-Molina y Pla-Santamaria (2018). Una ventaja adicional sobre el uso de los Notebooks es que su uso no se limita a Python porque también se pueden usar otros lenguajes de programación como R. Ketcheson (2014) destaca el uso exitoso de los Notebooks para cursos universitarios de un semestre y tutoriales breves de 1 a 3 días. Sin embargo, el uso de Python en estudios de economía y finanzas no está exento de dificultades. Ketcheson (2014) señala algunas dificultades para que los estudiantes comiencen a usar los Notebooks debido a problemas de instalación. Salas-Molina y Pla-Santamaria (2018) también informan problemas de instalación, incluso cuando a los estudiantes se les proporcionó previamente un breve tutorial sobre cómo instalar el software requerido en sus propias computadoras portátiles.

Otros inconvenientes sobre el uso de Notebooks destacados por Ketcheson (2014) incluyen los siguientes: 1) el material cubierto en el curso es sustancialmente menor que en el caso de un método tradicional debido al hecho de que incluso los algoritmos simples requieren mucho tiempo para los estudiantes; 2) dificultades para gestionar clases grandes; 3) diferentes resultados obtenidos cuando el Notebook no se ejecuta de forma secuencial; 4) problemas para abrir Notebooks; 5) impedimentos para desarrollar programas largos; y 6) falta de interactividad.

En relación con los aspectos metodológicos del uso de Python como recurso didáctico, Kuroki (2021) destaca el hecho de que el uso de una forma alternativa de enseñanza puede complementar el enfoque tradicional de papel y lápiz. Jenkins (2022) se refiere a el equilibrio entre el tiempo dedicado a desarrollar los conocimientos informáticos y el tiempo dedicado a tratar temas de economía. Sin embargo, el autor comprobó mediante una encuesta que los estudiantes tenían un fuerte deseo de aprender habilidades informáticas a pesar de que partían de una experiencia limitada en programación. Salas-Molina y Pla-Santamaria (2018) describen cómo la enseñanza de un concepto novedoso para los estudiantes, como el problema de gestión de tesorería, utilizando una herramienta novedosa como los Notebooks, hizo que los estudiantes se enfrentaran a un doble desafío. En este sentido, parece una estrategia más adecuada utilizar conceptos conocidos como paso introductorio para comenzar a utilizar cualquier herramienta computacional en el aula.

Como resultado adicional del análisis de las experiencias educativas sobre el uso de Python en los campos de la economía y las finanzas, la Tabla 4 recoge un resumen de ventajas e inconvenientes. Parece claro que el pensamiento computacional y la tecnología de la información son habilidades útiles que pueden ayudar a los estudiantes a encontrar un buen trabajo. Debido a su popularidad, coste nulo, facilidad de uso para los principiantes, y la creciente disponibilidad de recursos útiles, Python es una buena opción a considerar. Los problemas de instalación, la cantidad de tiempo requerido para aprender los conceptos básicos de Python, el tiempo requerido para implementar incluso algoritmos simples y la necesidad de restringir las clases a un número reducido de estudiantes también son aspectos a considerar.

Tabla 4: Resumen de pros y contras del uso de Python para la educación en economía y finanzas.

Ventajas	Inconvenientes
Código abierto y gratuito	Problemas de instalación
Fácil para principiantes	Tiempo requerido para aprender conceptos básicos
Desarrollar habilidades computacionales	Tiempo requerido para implementar algoritmos
Complemento a la enseñanza tradicional	Necesario tener pocos alumnos
Disponibilidad de recursos	Diferentes habilidades informáticas de los estudiantes

Como propuestas para resolver las dificultades identificadas, a continuación se plantean algunas decisiones que se podrían tomar para facilitar el uso de Python en el aprendizaje de economía y finanzas. En primer lugar, una opción adecuada para evitar problemas de instalación es el uso de una plataforma web que proporcione servicios de Notebooks como Google Colaboratory o Microsoft Azure. El tiempo requerido para aprender conceptos básicos y para implementar algoritmos se podría paliar mediante una metodología de aprendizaje inverso (Prieto Martín, 2017; Sams & Bergmann, 2014) basado en la revisión previa de materiales escritos y audiovisuales para el aprendizaje asíncrono por parte de los alumnos. En este sentido, las clases presenciales se destinarían a resolver dificultades planteadas por los alumnos o casos prácticos de aplicación de los materiales revisados en casa. Esta metodología de aprendizaje inverso también puede servir de ayuda para reducir el impacto por la presencia de un elevado número de alumnos en el curso e incluso para potenciar un aprendizaje más personalizado que intente afrontar las diferentes habilidades informáticas de los estudiantes.

5 Conclusiones

Este artículo contribuye a resumir el estado del arte acerca de cómo se puede enseñar economía y finanzas utilizando el lenguaje de programación Python. Las referencias identificadas, los recursos señalados así como las ventajas y los inconvenientes destacados representan puntos de partida para la toma de decisiones por parte de los profesores del área de la economía y las finanzas.

En primer lugar, las experiencias educativas identificadas plantean no solo las ventajas que Python puede aportar a los alumnos (como el desarrollo del pensamiento computacional y su gratuidad) sino también los retos a los que los profesores se han enfrentado. En este sentido, se puede destacar los problemas de instalación y el tiempo necesario para un aprendizaje básico como los mayores problemas identificados. Sin embargo, el uso de las nuevas plataformas de trabajo con Python en la web y de metodologías de aprendizaje inverso pueden ser soluciones relativamente sencillas de implantar. Estas conclusiones y otras que de estas nuevas experiencias se pudieran derivar serían fácilmente transferibles a otros contextos educativos en los que los aspectos cuantitativos tengan un papel relevante como la ingeniería, las matemáticas, la física, la química o la biología.

Por otro lado, con el fin de facilitar la tarea de los profesores que consideran el uso de Python para enseñar economía y finanzas, este artículo presenta una serie de recursos disponibles para los profesores que quieran recurrir a Python como herramienta de aprendizaje. Se recogen diferentes áreas de estudio dentro de la economía y las finanzas, distintos niveles y una breve reseña de sus características principales para permitir a los docentes un acceso más rápido y una evaluación más sencilla de los recursos disponibles.

Referencias bibliográficas

- Aruoba, S. B., & Fernández-Villaverde, J. (2015). A comparison of programming languages in macroeconomics. *Journal of Economic Dynamics and Control*, 58, 265-273.
- Baiocchi, G. (2003). Managing econometric projects using Perl. *Journal of Applied Econometrics*, 18(3), 371-378.

- Blumenstyk, G. (2016). Liberal-arts majors have plenty of job prospects, if they have some specific skills, too. *The Chronicle of Higher Education*.
- Choirat, C., & Seri, R. (2009). Econometrics with Python. *Journal of Applied Econometrics*, 24(9), 698-704.
- Coleman, C., Lyon, S., Maliar, L., & Maliar, S. (2021). Matlab, Python, Julia: What to Choose in Economics? *Computational Economics*, 58(4), 1263-1288.
- Cribari-Neto, F., & Zarkos, S. G. (1999). R: Yet another econometric programming environment. *Journal of Applied Econometrics*, 14(3), 319-329.
- De Prado, M. L. (2018). *Advances in financial machine learning*. John Wiley & Sons.
- Dixon, M. F., Halperin, I., & Bilokon, P. (2020). *Machine learning in Finance* (Vol. 1406). Springer.
- Duarte, V., Duarte, D., Fonseca, J., & Montecinos, A. (2020). Benchmarking machine-learning software and hardware for quantitative economics. *Journal of Economic Dynamics and Control*, 111, 103796.
- Eddelbuettel, D. (2000). Econometrics with Octave. *Journal of Applied Econometrics*, 15(5), 531-542.
- Ernst, D. C., Hodge, A., & Yoshinobu, S. (2017). What is inquiry-based learning. *Notices of the AMS*, 64(6), 570-574.
- Fletcher, S., & Gardner, C. (2009). *Financial modelling in Python*. John Wiley & Sons.
- Glasserman, P. (2004). *Monte Carlo methods in financial engineering* (Vol. 53). Springer.
- Gogas, P., & Papadimitriou, T. (2021). Machine learning in economics and finance. *Computational Economics*, 57(1), 1-4.
- Google. (2022). Google Colaboratory [<https://colab.research.google.com/>, accessed: 2022-09-01].
- Guo, P. (2014). Python Is Now the Most Popular Introductory Teaching Language at Top U.S. Universities. *Communications of the ACM*.
- Gutiérrez, L. M. G., Pita, J. L. C., & Burgos, D. E. (2016). Implantación de metodologías de cálculo a través del lenguaje Python para asignaturas impartidas en las titulaciones de la ETSIN. *Modelling in Science Education and Learning*, 9(1), 151-160.
- Guzdial, M. (2008). Education Paving the way for computational thinking. *Communications of the ACM*, 51(8), 25-27.
- Hilpisch, Y. (2014). *Python for Finance: Analyze Big Financial Data*. O'Reilly Media.

- Hilpisch, Y. (2015). *Derivatives analytics with Python: data analysis, models, simulation, calibration and hedging*. John Wiley & Sons.
- Hilpisch, Y. (2020). *Artificial Intelligence in Finance*. O'Reilly Media.
- Hilpisch, Y. (2021). *Financial Theory with Python*. O'Reilly Media.
- Jenkins, B. C. (2022). A Python-based undergraduate course in computational macroeconomics. *The Journal of Economic Education*, 53(2), 126-140.
- Kelliher, C. (2022). *Quantitative Finance with Python: A Practical Guide to Investment Management, Trading, and Financial Engineering*. CRC Press.
- Ketcheson, D. I. (2014). Teaching numerical methods with IPython notebooks and inquiry-based learning. *Proceedings of the 13th Python in Science Conference*.
- Kuroki, M. (2021). Using Python and Google Colab to teach undergraduate microeconomic theory. *International Review of Economics Education*, 38, 100225.
- Lee, S. C., & Lee, E. T. (1974). Fuzzy sets and neural networks. *Journal of Cybernetics*, 42(2), 83-103.
- Lewinson, E. (2022). *Python for Finance Cookbook*. Packt Publishing Ltd.
- Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, 7(1), 77-91.
- Mas-Colell, A., Whinston, M. D., Green, J. R., et al. (1995). *Microeconomic theory*. Oxford University Press.
- Medina López, C., Alfalla Luque, R., & Marín García, J. A. (2010). Una propuesta metodológica para la realización de búsquedas sistemáticas de bibliografía. *Working Papers on Operations Management*, 1(2), 13-30.
- Ministerio de Universidades. (2022). Estadística de estudiantes [<https://www.universidades.gob.es/estadistica-de-estudiantes/>, accessed: 2023-05-23].
- Oosterlee, C. W., & Grzelak, L. A. (2019). *Mathematical modeling and computation in finance: with exercises and Python and Matlab computer codes*. World Scientific.
- Pérez, F., & Granger, B. E. (2007). IPython: a system for interactive scientific computing. *Computing in Science & Engineering*, 9(3).
- Prieto Martín, A. (2017). *Flipped Learning: aplicar el modelo de aprendizaje inverso*. Narcea Ediciones.
- Pugh, D. R. (2014). Python for research and teaching economics. *Proceedings of the 13th Python in Science Conference*.

- Romer, D. (2012). *Advanced Macroeconomics* (4th). McGraw-Hill.
- Rossant, C. (2014). *IPython interactive computing and visualization cookbook*. Packt Publishing.
- Salas-Molina, F., & Pla-Santamaria, D. (2018). Coding oriented learning in economics, business and finance. *Modelling in Science Education and Learning*, 11(1), 55-64.
- Salas-Molina, F., & Pla-Santamaria, D. (2017). *Gestión de tesorería con Python*. Editorial Universitat Politècnica de València.
- Sams, A., & Bergmann, J. (2014). *Dale la vuelta a tu clase: lleva tu clase a cada estudiante, en cualquier momento y cualquier lugar*. Ediciones SM España.
- Sargent, T. J., & Stachurski, J. (2014). Lectures in Quantitative Economics [<https://quantecon.org/>, accessed: 2022-09-03].
- Sarmas, E., Xidonas, P., Doukas, H., et al. (2020). *Multicriteria portfolio construction with Python*. Springer.
- Scott, D. M., Smith, C., Chu, M.-W., & Friesen, S. (2018). Examining the efficacy of inquiry-based approaches to education. *Alberta Journal of Educational Research*, 64(1), 35-54.
- Sheppard, K. (2021). Introduction to Python for econometrics, statistics and data analysis (5.^a ed.). *Self-published, University of Oxford*.
- Stachurski, J. (2009). *Economic dynamics: theory and computation*. MIT Press.
- Stackoverflow. (2022). Stack Overflow Trends [<https://insights.stackoverflow.com/trends>, accessed: 2022-09-01].
- Weiming, J. M. (2015). *Mastering Python for Finance*. Packt Publishing.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717-3725.
- Yan, Y. (2014). *Python for finance*. Packt Publishing.