Python, Jupyter Notebooks y Google Colab: potenciando la investigación reproducible en Salud Pública

René Lagos Barrios[1*], Diego Merino Vásquez[2]

[1]Programa de Doctorado en Salud Pública y Grupo de Ciencia de Datos para la Salud Pública
Escuela de Salud Pública, Facultad de Medicina, Universidad de Chile

[2]Programa de Formación de Médicos Especialistas en Salud Pública
Escuela de Salud Pública, Facultad de Medicina, Universidad de Chile

El análisis de datos conlleva múltiples problemas a los que nos enfrentamos frecuentemente en salud pública. Por ejemplo, computadoras lentas al manejar grandes bases de datos (ej. egresos hospitalarios, defunciones, etc.), dificultad para compartir análisis y asegurar que todos usen los mismos datos y código, problemas de configuración de *software* entre diferentes sistemas operativos, y el alto costo de las licencias de *software* estadístico. En los cursos de epidemiología de la Escuela de Salud Pública (ESP) se enseña el uso de Stata, programa estadístico que requiere compra de licencias anuales para que los y las estudiantes lo puedan utilizar. Sin embargo, para funcionarios de una institución pública una licencia individual de Stata cuesta del orden de mil dólares anuales, lo que representa una barrera para reproducir los análisis desarrollados en la ESP[1].

Una alternativa a estos problemas es usar Python y Google Colaboratory (Colab). Colab es una herramienta gratuita y en la nube que permite escribir y ejecutar código Python en el navegador, funcionando en base a un cuaderno de Jupyter Notebook[2]. No requiere instalación, solo una cuenta de Google y un navegador web. Ofrece acceso gratuito a máquinas virtuales de Google con una capacidad de 12 GB de RAM y 50 GB de almacenamiento en disco (aunque para los usuarios que no pagan esta capacidad no está garantizada)[3]. Además, cuenta con la inteligencia artificial (IA) de Google integrada para asistir la generación de código, corrección de errores y documentación de los análisis.

Los cuadernos de Jupyter y Colab se componen de dos bloques fundamentales:

1. Celdas de Texto: Para poner títulos de secciones, explicar metodología y/o fuentes de datos, interpretar resultados y estructurar el análisis, permitiendo usar formato enriquecido (Markdown, LaTeX) e insertar imágenes y enlaces.
2. Celdas de Código: Donde se escribe y ejecuta código Python y se muestran los resultados de este. Al crear una celda de código en nueva, se puede insertar un *prompt* para que Gemini cree el código para la tarea requerida. Mientras que en celdas existentes, con código y resultados previos, se puede solicitar que Gemini explique el código o que lo modifique para ajustar el análisis (ver Figura 1).

Las celdas de texto y de código se insertan y ejecutan en el orden que el usuario quiere, pudiendo volver a celdas anteriores para ajustar parámetros y actualizar los resultados.

Primeros pasos

Para empezar a usar Colab, se debe ir a colab.research.google.com, iniciar sesión con una cuenta de Google, crear un nuevo cuaderno y conectarlo a computador de Google. Colab tiene preinstaladas las principales librerías de ciencia de datos de Python (como `pandas`, `matplotlib`, etc.), pero se pueden instalar otras librerías del repositorio de Python u otro en Github. Ejemplos:

```
import pandas as pd
import numpy as np
import statsmodels.formula.api as smf
import plotly.express as px
import seaborn as sns
import matplotlib.pyplot as plt
```

Una vez iniciada la sesión se puede conectar con Google Drive para acceder a archivos con el comando a continuación, tras lo cual solicitará autentificar la identidad.

```
from google.colab import drive
drive.mount('/content/drive')
```

También es posible configurar los cuadernos de Colab para funcionar con código de R, con lo que se instalan automáticamente algunas de las librerías más comunes (`tidyverse`, `ggplot2`, entre otras).

*rlagos@uchile.cl.

[1]https://www.stata.com/order/new/gov/single-user-licenses/dl/.

[2]https://jupyter.org/.

[3]https://research.google.com/colaboratory/faq.html#resource-limits.

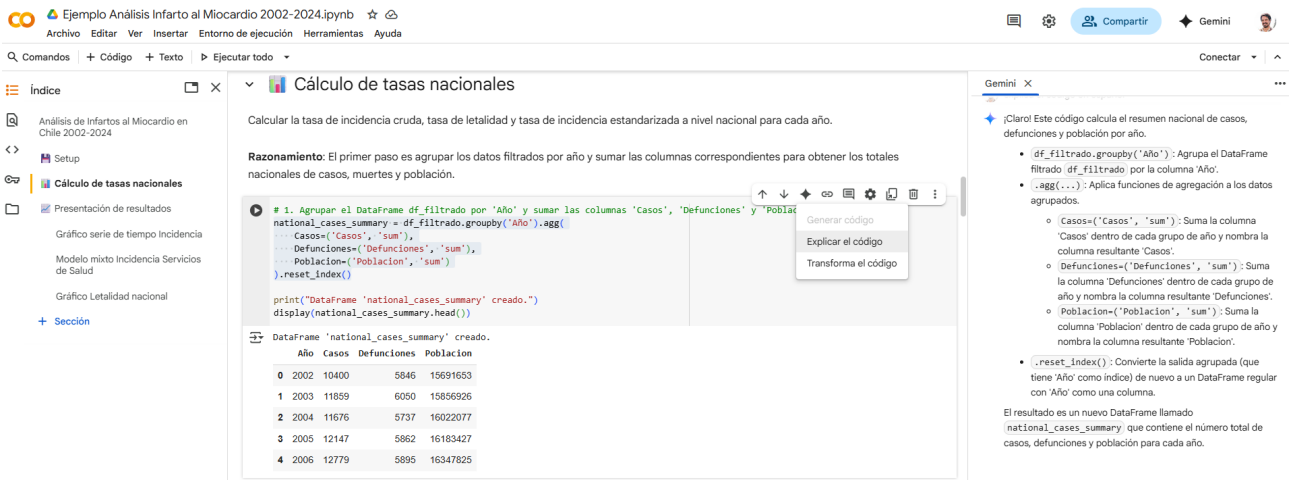

Figura 1. Al seleccionar una celda de código aparece el ícono de Gemini que sirve para explicar o transformar el código. En este caso se seleccionó explicar el código seleccionado, con lo que se abrió un chat con la explicación en la columna derecha donde se puede seguir interactuando con la IA.

Experiencia Práctica

En el proyecto de investigación de primer año de la beca de Salud Pública, se utilizó Colab para llevar a cabo una investigación con datos públicos del DEIS (defunciones y egresos hospitalarios). El objetivo era analizar la tendencia temporal de incidencia y letalidad del infarto al miocardio (IAM) en Chile y los servicios de salud de 2002 a 2024. Se generó un notebook para consolidar los datos y obtener una tabla multidimensional en formato Excel. En otro notebook se cargaron los datos, prepararon para los análisis, se hizo el análisis estadístico descriptivo y se generaron gráficos de las series de tiempo de incidencia y letalidad por Servicio de Salud. Un ejemplo del notebook utilizado para el análisis se encuentra disponible en Análisis Infarto al Miocardio 2002-2024.ipynb. Como se observa en la Figura 1, en la columna izquierda, el cuaderno está dividido en tres secciones: "Setup", donde se cargan los datos y se formatean las columnas para el análisis; "Cálculo de tasas nacionales", donde se calculan las métricas para el análisis; y "Presentación de resultados", donde se grafican las series de tiempo y se modela la incidencia de los servicios de salud con un modelo de regresión mixto. Ingresando al link puede revisar el código, los resultados y volver a reproducirlos por su cuenta.

Beneficios de Usar Colab

Accesibilidad

Colab eliminó las barreras económicas de *software* y *hardware* para desarrollar el proyecto. Python es un lenguaje de programación abierto y se utiliza ampliamente en educación de ciencia de datos por su facilidad de aprendizaje, gracias a su filosofía Zen[4] (Paffenroth & Kong, 2015), siendo un ejemplo de ello el Magíster de Informática Médica de la Facultad de Medicina. Jupyter es un entorno de desarrollo (IDE) abierto que permite programar código y ejecutar análisis en el navegador. Todos los notebooks de Colab se almacenan en el formato de notebook de Jupyter (.ipynb), que es de código abierto y no requiere compra de licencias.

Interactividad

La estructura de los notebooks de Jupyter está orientada a la computación interactiva y la formulación de preguntas y respuestas con datos, lo hace que la herramienta sea intuitiva para aprender métodos computacionales (Granger & Pérez, 2021). Por esta razón, es una herramienta popular para enseñar y compartir código en clases y talleres. En la ESP se utiliza en los cursos "Machine Learning aplicado a la Salud" y "Ciencia de datos para la programación de servicios de salud" para desarrollar talleres de manejo de datos[5].

Colaboración

Los notebooks se comparten como Google Docs, permitiendo la colaboración en tiempo real con tutores o equipos, al igual que las planillas o documentos de Googledocs: comentarios, historial de modificaciones, etc. También se pueden guardar y compartir en Github y aprovechar las funcionalidades de esta plataforma. Esto permite almacenar todo el material de trabajo en la nube y acceder a él desde cualquier computador con acceso a internet.

Reproducibilidad

Al cerrar un *notebook*, las tablas, gráficos y textos no se pierden, por lo que un único archivo contiene

[4] https://peps.python.org/pep-0020/.
[5] https://github.com/rlagosb/taller_eiv.

todo el análisis (texto, código, resultados, gráficos). Es posible ver los códigos con que se generó un análisis y volver a ejecutarlo para verificarlos sin necesidad de instalar *software*. Esto lo hace práctico para difundir los códigos y resultados de tesis e investigaciones realizadas en la ESP.

Uso de Inteligencia Artificial

El uso de IA acelera significativamente el proceso de generar código y solucionar errores. No es necesario copiar y pegar de un chat a otro los códigos o mensajes de error, sino que la IA está inserta en la interfaz de Colab y puede revisar, explicar y corregir el código mientras se edita (Figura 2)[6]. Esto minimiza significativamente el tiempo dedicado a programar y depurar errores y permite destinar más tiempo a la interpretación de los resultados.

Limitaciones y Riesgos

Capacidad computacional

En su versión gratuita, los recursos de máquina virtual no están garantizados y pueden variar durante el día[7]. En caso que esto interrumpa la ejecución de una celda, se debe esperar y volver a ejecutarla. Por otra parte, si se deja de usar el *notebook* más de 90 minutos se desconecta, por lo que se debe volver a ejecutar el código para poder seguir trabajando, lo que puede ser un inconveniente en proyectos de larga duración o que requieren recursos constantes.

Integridad del código

La IA de Colab genera código a partir de un modelo entrenado con grandes cantidades de código, lo que permite aprovechar el conocimiento existente, pero puede reproducir código ineficiente, inadecuado o inseguro. Es responsabilidad del usuario entender el código y verificar que no tenga errores y vulnerabilidades antes de ejecutarlo[8]. Así mismo, al usar repositorios de código abierto es necesario cumplir los requisitos de las licencias, como citar el autor del código original en el caso de licencias CCBY o MIT.

Seguridad de la información

Al usar el *drive* provisto por la Universidad para almacenar los datos y los notebooks, se beneficia de las medidas de seguridad y privacidad institucionales[9].

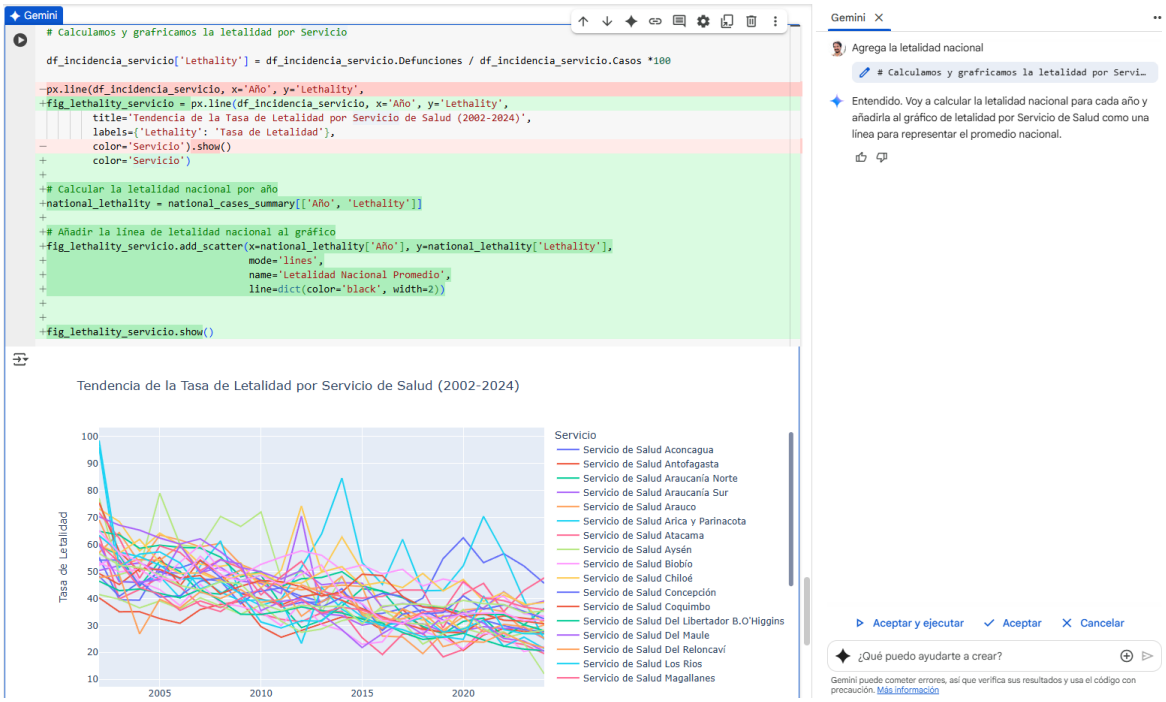

Figura 2. Al seleccionar "Transforma el código" en cualquier celda de código, se abre un chat para dar las instrucciones ("Agregar la letalidad nacional", en este caso) y Gemini edita el código de la celda mostrando en rojo las líneas eliminadas y en verde las líneas agregadas. Seleccionando "Aceptar y ejecutar" se guardan los cambios y se ejecuta el código, seleccionando "Cancelar" se revierten los cambios propuestos y se recupera el código original. La depuración de errores funciona con la misma lógica.

[6] La IA de Colab no tiene acceso a los archivos de Google Drive, pero puede generar código que acceda a ellos a solicitud explícita del usuario (https://research.google.com/colaboratory/faq.html#ai-drive-secrets-access).

[7] En la versión sin costo de Colab, los notebooks se pueden ejecutar durante 12 horas como máximo (https://research.google.com/colaboratory/faq.html#resource-limits).

[8] https://research.google.com/colaboratory/tos_v5.html.

[9] https://vti.uchile.cl/ayudatecnologica/articulo/descripcion-de-herramientas-colaborativas-g-suite-for-educations/.

Sin embargo, Colab no forma parte de los servicios gestionados por la Universidad y todo el código y los datos se procesan en los servidores de Google. Además, las interacciones con la IA son almacenadas y utilizadas por Google[10]. Esto subraya la importancia de asegurar la privacidad de la información personal mediante la anonimización o pseudoanonimización de los datos antes de cargarlos en la nube y de no incluir información personal al interactuar con la IA.

Estadísticas avanzadas

Python tiene librerías avanzadas para manejo de datos y *machine learning*, mientras que para análisis estadísticos tiene lo estándar: estadística descriptiva, tests de hipótesis, regresiones, etc. Sin embargo, para análisis estadísticos avanzados R tiene librerías más desarrolladas y una comunidad estadística más especializada. Si bien Colab se puede configurar para correr código de R, su ecosistema no es tan nativo como con Python, lo que podría implicar limitaciones para usuarios acostumbrados a R o que requieren realizar análisis estadísticos avanzados.

Dependencia de la Conexión a Internet

Dado que Google Colab es una herramienta basada en la nube, su funcionamiento depende completamente de una conexión a internet. En ausencia de conexión, o si esta es intermitente, no es posible acceder a los cuadernos, ejecutar código ni guardar cambios. Esto puede ser una limitación significativa en entornos donde la conectividad es limitada o poco fiable, o para trabajar fuera de línea. La sincronización de los *notebooks* en un computador local permite disminuir la dependencia de una conexión y mantener un respaldo local de los códigos.

Delegación del pensamiento

Existe evidencia que el uso excesivo de IA generativa puede afectar la capacidad de aprendizaje de los/las estudiantes[11] (Deng et al., 2025). Si bien su uso mejora el desempeño académico, también reduce el esfuerzo mental, por lo que podría debilitar la capacidad de análisis riguroso de los código y datos. Para docentes y supervisores, por su parte, plantea el desafío de distinguir entre los aportes originales de los/las estudiantes y de la IA. La Facultad de Medicina de la U. de Chile recomienda declarar el uso de IA en toda producción académica y enfatiza la responsabilidad ética e intelectual de los/las investigadores/as sobre los contenidos finales, mediante la incorporación de la siguiente declaración en informes de investigación y publicaciones científicas (Jerez Yáñez, 2025):

> *Durante el desarrollo de este estudio, se utilizaron herramientas de inteligencia artificial de tipo [especificar herramienta, por ejemplo: ChatGPT 4.0, OpenAI] como apoyo en las siguientes etapas: [revisión de literatura, redacción preliminar, análisis exploratorio, etc.]. Su uso fue supervisado por el equipo investigador, sin delegación de decisiones metodológicas, analíticas o interpretativas. El contenido final ha sido validado por los autores conforme a los principios de integridad académica y autoría responsable.*

Conclusión

Python, Jupyter notebooks y Google Colaboratory (Colab) son herramientas que facilitan experimentar diferentes enfoques de análisis de datos, promueven la claridad al documentar el flujo de trabajo en un solo lugar, y permiten trabajar desde cualquier lugar con acceso a internet. Al ser una herramienta gratuita y colaborativa, facilita la investigación transparente y reproducible en salud pública. Sin embargo, debe utilizarse con una comprensión clara del código que se ejecuta, los términos legales de su uso y las medidas de seguridad básicas para resguardar la privacidad de la información.

Referencias

Deng, R., Jiang, M., Yu, X., Lu, Y., & Liu, S. (2025). Does ChatGPT enhance student learning? A systematic review and meta-analysis of experimental studies. *Computers & Education*, *227*, 105224. https://doi.org/10.1016/j.compedu.2024.105224

Granger, B. E., & Pérez, F. (2021). Jupyter: Thinking and Storytelling With Code and Data. *Computing in Science & Engineering*, *23*(2), 7-14. https : / / doi . org / 10 . 1109 / MCSE . 2021 . 3059263

Jerez Yáñez, O. (2025). *Humanizar la Inteligencia: Orientaciones para un uso ético y transformador de la IA en la educación y la investigación en salud*. Departamento de Educación en Ciencias de la Salud (DECSA), Facultad de Medicina, Universidad de Chile. https://doi.org/10.34720/17ej-j164

Paffenroth, R., & Kong, X. (2015). Python in Data Science Research and Education. En K. Huff & J. Bergstra (Eds.), *Proceedings of the 14th Python in Science Conference* (pp. 164-170). https://doi.org/10.25080/Majora-7b98e3ed-019

[10] https://research.google.com/colaboratory/faq.html#ai-data-collection.
[11] https://observatorio.tec.mx/la-ia-generativa-puede-afectar-el-aprendizaje/.