



Girls Who Code en casa

Zona de recreo de datos
Manipulación de datos en Python

Descripción de la actividad

¿Sabías que cerca del 70 % de los datos generados por la internet no se utilizan? Los datos pueden ser increíblemente útiles para las empresas, las cuales, por ejemplo, podrían utilizarlos para sugerir restaurantes en tu aplicación preferida de entrega de comida. Sin embargo los conjuntos de datos solo pueden ser tan valiosos como la diversidad de las personas que los crean. Si no hubiera mujeres y minorías desarrollando la tecnología que define nuestro futuro, los sistemas que dirigen nuestras vidas estarían determinados por los prejuicios de las personas (sobre todo, hombres blancos) que los crearon.

El campo de la [ciencia de datos](#) es vasto y conecta conceptos de inteligencia artificial, minería de datos, macrodatos (también conocidos como datos masivos o big data) y aprendizaje automático. Según [Glassdoor](#), uno de los empleos con mayor demanda es el de experto en datos, ¡con un salario anual base de **\$110,000!** En esta actividad, aprenderás a extraer información de un conjunto de datos para observar posibles tendencias. Con **Python**, aprenderemos a realizar una manipulación básica de datos, filtrando, modificando, eliminando y realizando cálculos en un conjunto de datos. En esta actividad, analizaremos de modo específico datos relacionados con proyectos de [Kickstarter](#), una plataforma de financiamiento colectivo o crowdfunding, para determinar la categoría de proyectos que tiene mayor éxito.

Objetivos del aprendizaje

Al finalizar esta actividad, serás capaz de:

- ◆ Usar la descomposición para determinar cómo utilizar datos para resolver un problema.
- ◆ Guardar, buscar, eliminar y modificar datos con Python.
- ◆ Entender cómo navegar por Kaggle y Jupyter Notebook para explorar un conjunto de datos.

Materiales

- ◆ [Datos de Kaggle sobre Kickstarter](#)
- ◆ [Proyecto de muestra](#)

Conocimientos previos

Antes de comenzar este proyecto, te recomendamos que seas capaz de:

- ◆ Explicar con tus propias palabras qué es una [variable](#) y describir cómo pueden usarse en un programa.
- ◆ Explicar con tus propias palabras qué es un [enunciado condicional](#) y describir cómo pueden usarse en un programa.
- ◆ Explicar con tus propias palabras qué es un [método o función](#) y describir cómo pueden usarse en un programa.
- ◆ Tener experiencia con un lenguaje basado en texto, como JavaScript, Python, Swift, etc.

Si quieres un repaso rápido de Python, te recomendamos que consultes nuestra actividad

[¿Puedo ayudarte?](#)

“Mujeres en tecnología” artículo destacado Theresa Johnson



Fuente de la imagen: [Medium](#)

Un currículum es un documento que presenta las habilidades y los logros de una persona. Es una oportunidad para alardear de tus habilidades y permitir que los futuros empleadores sepan lo asombrosa que eres como persona. Sin embargo, ¿ha oído hablar de un **currículum de fracasos**? Además de mantener y actualizar su currículum normal, Theresa Johnson documenta todos sus fracasos en otro currículum. Ella considera que los errores son experiencias muy valiosas de las que puede aprender. [Aquí](#) puedes conocer más acerca de lo que es un currículum de fracasos y descubrir cómo obtuvo esta idea de [Tina Seelig](#) de Stanford.

La Dra. Theresa Johnson es diseñadora de productos en [Airbnb](#), donde también trabajó como experta en datos. Antes de trabajar en Airbnb, obtuvo un doctorado en Aeronáutica y Astronáutica de Stanford University. Tal vez te preguntes: ¿cómo se relacionan todas estas funciones y campos? Es posible que no logres encontrar una conexión inicial entre la aeronáutica y los datos en Airbnb, pero la manera en que utilizamos y manipulamos los datos es, en esencia, la misma. Durante su tiempo como experta en datos, Theresa trabajó en el análisis de indicadores para optimizar las reseñas de casas en diversos mercados. Ahora, como gerente de producto, aprovecha su experiencia con el aprendizaje automático y la inteligencia artificial para entender los datos de pago y coordinar y comunicar esta información entre varios equipos.

Como mujer de color en la industria tecnológica, Theresa es vocera de la importancia de la diversidad en la industria. Theresa también es presidenta de la junta directiva de [StreetCode Academy](#), una organización sin fines de lucro que provee recursos tecnológicos gratuitos a comunidades de color en Silicon Valley. Esta organización conecta a los estudiantes con computadoras portátiles y cursos de programación, iniciativa empresarial y diseño.

Ve este [video](#) para conocer más sobre Theresa y su trabajo en Airbnb.

Reflexión

Ser una experta informática significa mucho más que simplemente ser buena programando. Toma unos minutos para reflexionar sobre cómo Theresa y su trabajo reflejan las características que todos los verdaderos expertos informáticos deben desarrollar en sí mismos: valentía, resiliencia, creatividad y propósito.



RESILIENCIA

Theresa destaca la importancia del fracaso en su currículum de fracasos. Piensa en una ocasión en la que enfrentaste un fracaso y quedaste decepcionada. ¿Cómo podrías reescribir esta experiencia para que sea más positiva? ¿Cuál fue una cosa que aprendiste acerca de ti que podrías aplicar en tu próximo desafío?

Comparte tus respuestas con un familiar o amigo. Anima a otras personas para que lean sobre Theresa y se unan a la charla.

Paso 1: ¿Qué son los macrodatos? (5 a 10 minutos)

Aproximadamente el 70 % de los datos generados por la internet no son utilizados, ¿pero por qué? En esta sección hablaremos de los macrodatos y por qué son valiosos para empresas en todos los sectores.

Macrodatos (2 minutos)

Los **macrodatos** (también llamados datos masivos o big data) son precisamente eso: ¡muchos datos! Diferenciamos los macrodatos de los datos normales porque los macrodatos se generan con una tasa difícil de mantener y por lo general requieren un gran esfuerzo para “limpiarlos” o prepararlos para su uso e interpretación. Piensa en cómo podrías enviarle a una amiga un mensaje de texto con una frase sencilla, como “OK”. Tal vez envíes mensajes con “okay”, “ok”, “k”, “kay” y muchos más. Esto también podría variar según la manera en que usemos mayúsculas, minúsculas y caracteres especiales. Todas estas opciones posibles generaron datos y necesitamos entrenar a la computadora para que reconozca que todas estas palabras significan lo mismo.



No obstante el enorme volumen de datos disponibles, podría considerarse que gran parte de ellos no pueden usarse, debido a la existencia de sesgo. ¿Qué es el sesgo? En los datos, el sesgo ocurre cuando un resultado en particular podría ser más favorable para cierta consecuencia. Esto podría suceder por diversas razones, una de las cuales es que la información recopilada no es representativa de la población en general. Supongamos que realizas una encuesta entre personas de tu escuela. Si solo les preguntaras a las primeras 20 personas que vieras, ¿crees que los datos serían representativos de toda la escuela? Realmente no. No solo es importante a *quién* encuestaras o sobre *qué* realizarás la encuesta para representar a toda la población, sino también el tipo de información que vas a recopilar. Para obtener un conjunto de datos diversos, es importante que haya diversidad en la población y en el equipo que lleva a cabo el análisis. Si quieres conocer más sobre el sesgo de datos, consulta este [artículo](#) de Elder Research o este [video](#) de Google.

Comenzar con Kaggle (5 a 8 minutos)



Kaggle es un sitio web que contiene datos reales contribuidos por una comunidad en línea. Los datos alojados en este sitio web abarcan desde estadísticas sobre [COVID-19](#), [videos de YouTube](#), [aplicaciones en Google Play](#), [cáncer de seno](#), [precios del aguacate](#) e [incluso Pokémon](#). Es un gran lugar para explorar datos reales que afectan las decisiones que tomamos en la actualidad. En Kaggle, puedes crear una **libreta** asociada con un conjunto de datos en particular y escribir código en el sitio web. Es una excelente manera de organizar tus proyectos y enviar trabajo a proyectos reales en vivo.

- **Crea una cuenta de Kaggle.** Haz clic en este [enlace](#) para crear una cuenta en Kaggle. También puedes hacer clic en el botón **Register** (Registro) en la esquina superior derecha del sitio web de Kaggle para crear una cuenta. *Si eres menor de 13 años, necesitarás el permiso y la dirección de correo electrónico de uno de tus tutores para inscribirte.*



- **Abre el [conjunto de datos de Kickstarter](#).** En la parte superior de este conjunto de datos hay un encabezado principal con el título del conjunto de datos, el creador y cuándo fue la actualización más reciente. Debajo de esto se presentan varias opciones: Data (Datos), Tasks (Tareas), Notebooks (Libretas), Discussion (Discusión), Activity (Actividad) y Metadata (Metadatos). Puedes conocer más sobre la funcionalidad que ofrece Kaggle para cada conjunto de datos [aquí](#).

Paso 2: Explorar el conjunto de datos (10 a 15 minutos)

Antes de comenzar a manipular los datos, debemos tomar tiempo para explorar el conjunto de datos que seleccionamos para esta actividad.

Proyectos de Kickstarter (1 minuto)

Kickstarter es una plataforma de financiamiento colectivo (lo que también se conoce como crowdfunding) donde las pequeñas empresas pueden abrir un proyecto en el sitio web para recaudar una pequeña inversión en su proyecto de la comunidad en línea. Las personas que quieren invertir en el proyecto prometen aportar cierta cantidad. A cambio de su aportación, se les entregan premios al completarse los proyectos. Puedes explorar algunos ejemplos de proyectos en Kickstarter [aquí](#).

Tal vez te preguntes, ¿por qué elegimos este conjunto de datos? En esta actividad, te guiaremos por un ejemplo básico de algunas de las cosas que pueden hacer para manipular datos. Estos datos de Kickstarter contienen valores [categóricos](#), es decir, que pueden agruparse en categorías, y numéricos, lo cual te presenta una perspectiva de cómo podrían estar representados otros conjuntos de datos.

Desglosar el conjunto de datos (5 a 8 minutos)

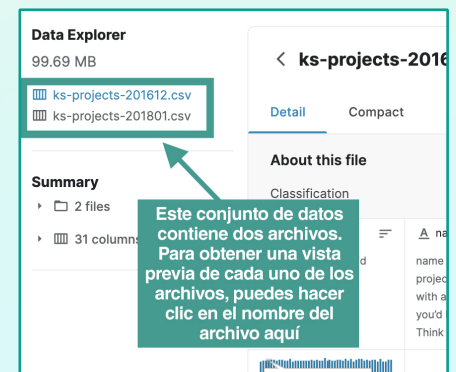


¡Sumerjámonos en los datos! Para comenzar, volvamos a abrir el [conjunto de datos de Kickstarter en Kaggle](#). Asegúrate de estar en la pestaña **Data** (Datos); deberás notar que la palabra “Data” en la barra de encabezado superior está en color [azul](#) y subrayada. Luego, avanza hasta la sección **Data Explorer** (Explorador de datos).

En el lado izquierdo, notarás que hay dos archivos, [ks-projects-201612.csv](#) y [ks-projects-201801.csv](#).


- ◆ [ks-projects-201612.csv](#): Este conjunto de datos contiene todos proyectos de Kickstarter lanzados antes de diciembre de 2016.
- ◆ [ks-projects-201801.csv](#): Este conjunto de datos contiene todos proyectos de Kickstarter lanzados antes de enero de 2018.

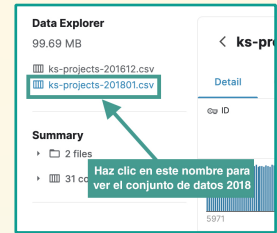
Puesto que queremos trabajar con los datos más recientes, solo usaremos el archivo [ks-projects-201801.csv](#), al que llamaremos el conjunto de datos de 2018.



Dado que [ks-projects-201801.csv](#) contiene todos los proyectos anteriores a enero de 2018, este conjunto de datos contiene **TODOS** los datos en [ks-projects-201612.csv](#) y más.

Paso 2: Explorar el conjunto de datos (cont.)

- ◆ **Abre el conjunto de datos de Kickstarter 2018.** En el **Data Explorer** (Explorador de datos), haz clic en el nombre **ks-projects-201801.csv** en el lado izquierdo. Deberás ver este nombre resaltado en **azul**.
- ◆ **Expande la vista.** Haz clic en el icono de caja  en la esquina superior derecha de la ventana del explorador de datos.
- ◆ **Ocultar la información de resumen.** Haz clic en la flecha a la izquierda del nombre del conjunto de datos. Esto debe cerrar el lado izquierdo y dejar más espacio para ver los datos.



Esta vista nos presenta una perspectiva rápida de los datos y algunos de los valores que contienen. Observa que no se muestra todo el conjunto de datos, pues es muy grande. De hecho, el conjunto de datos completo contiene la información de 375,765 proyectos de Kickstarter. Las **columnas** de este conjunto de datos se conocen como **características** de los datos. Las **características** nos permiten saber qué tipo de información se captura para cada proyecto. Cada **fila** del conjunto de datos representa un proyecto de Kickstarter o **entidad**. Por ejemplo, si tuviéramos un conjunto de datos sobre seres humanos, una persona sería una **entidad** en el conjunto de datos y algunas de las **características** que podríamos incluir serían el nombre de la persona, su color de ojos, su color de cabello, su fecha de nacimiento, etc.

Ejemplo

Nombre	Color de ojos	Color de cabello	Fecha de nacimiento
Reshma Saujani	Castaño	Castaño	18 de noviembre de 1975

Ejemplo de una **entidad**, Reshma Saujani, y algunas de las **características** que la definen.

Características de proyectos de Kickstarter (1 minuto)

Veamos con mayor detalle las **características**, o columnas, en estos datos de Kickstarter. Este conjunto de datos tiene 15 características en total: **ID**, **name** (nombre), **category** (categoría), **main_category** (categoría principal), **currency** (moneda), **deadline** (fecha límite), **goal** (meta), **launched** (lanzamiento), **pledged** (prometido), **state** (estado), **backers** (patrocinadores), **country** (país), **usd pledged** (USD prometido), **usd_pledged_real** (USD prometido real), **usd_goal_real** (USD meta real). La primera fila nos muestra el nombre de la característica y una descripción breve, mientras que la segunda fila presenta una perspectiva de algunos de los valores de cada característica. Tómame un minuto para revisar cada una de las características y su descripción breve.

Nota: En el explorador de datos, solo puedes ver 10 de las 15 columnas; hay un menú desplegable en la esquina superior derecha del conjunto de datos que te permite ver columnas adicionales. Tendrás que desplazarte hacia abajo por el menú para encontrar las 5 columnas adicionales que no se muestran en la vista predeterminada.



Paso 2: Explorar el conjunto de datos (cont.)

Descomposición del problema (5 a 10 minutos)

Ahora que entendemos un poco acerca del conjunto de datos y la información que se nos ha provisto, ¿qué debemos hacer? A partir de aquí, los expertos en datos llevan a cabo una lluvia de ideas sobre las preguntas que quieren responder con el conjunto de datos. Algunas de las posibles preguntas que podríamos considerar son:



- ◆ ¿Existe una relación entre la duración de un proyecto y que sea un éxito?
- ◆ ¿Existe una relación entre la cantidad de patrocinadores (personas que prometen donar) y que sea un éxito?
- ◆ ¿Podemos determinar si un proyecto tendrá éxito, a partir de las promesas de donación actuales?

Hay más preguntas de las que podemos contestar con este conjunto de datos. Si quieres explorar lo que otras personas han hecho con este conjunto de datos, explora las [libretas](#) correspondientes a este conjunto de datos en Kaggle. La pregunta que exploraremos en esta actividad es:

¿QUÉ CATEGORÍA PRINCIPAL DE PROYECTOS TIENE EL PORCENTAJE DE ÉXITO MÁS ALTO?



En esta siguiente parte, te mostraremos como dividir esta pregunta respondiendo a una serie de preguntas adicionales. Este proceso de dividir los problemas grandes en partes más pequeñas se conoce como **descomposición del problema**. Los expertos informáticos con frecuencia utilizan este método para entender mejor el problema y determinar en secciones más pequeñas cómo resolverlo.

PREGUNTA 1

¿Qué características
tendremos que usar para
responder a nuestra pregunta?

PREGUNTA 2

¿Cómo debemos calcular el
porcentaje de éxito?

PREGUNTA 3

¿Qué tareas debemos realizar
para responder a esta
pregunta?

Si lo deseas, haz una **pausa** para realizar una lluvia de ideas sobre cómo podrías responder a estas preguntas, antes de ver nuestras soluciones en la siguiente página.

Paso 2: Explorar el conjunto de datos (cont.)

Pregunta 1: ¿Qué características tendremos que usar para responder a nuestra pregunta?

Características incluidas en el conjunto de datos: `ID`, `name` (nombre), `category` (categoría), `main_category` (categoría principal), `currency` (moneda), `deadline` (fecha límite), `goal` (meta), `launched` (lanzamiento), `pledged` (prometido), `state` (estado), `backers` (patrocinadores), `country` (país), `usd_pledged` (USD prometido), `usd_pledged_real` (USD prometido real), `usd_goal_real` (USD meta real).

Dado que nuestra pregunta se centra en encontrar el porcentaje de proyectos exitosos según su categoría principal, las características que queremos usar son `state` (estado) y `main_category` (categoría principal).

- ◆ Queremos usar `main_category` en lugar de `category` (categoría) porque la característica `main_category` contiene un número fijo de opciones y es más general. Dado que la categoría es más específica, podríamos obtener algunas con pocos proyectos asociados, lo que nos daría resultados imprecisos.
- ◆ Usamos la característica `state` (estado) porque es la única que nos informa si el proyecto tuvo éxito.

Pregunta 2: ¿Cómo debemos calcular el porcentaje de éxito?

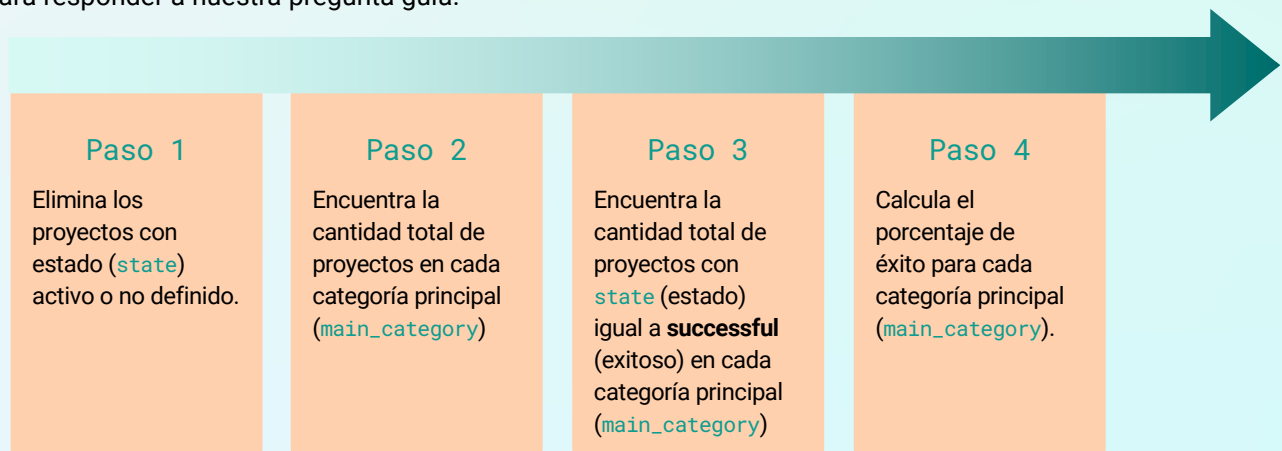
Para calcular el porcentaje de éxito, debemos comparar el número de éxitos con algo. Dado que nos concierne el éxito por categoría principal, debemos calcular el índice de éxito como sigue:

$$\% \text{ of success for category}_{\text{film+video}} = \frac{\# \text{ of successful projects in category}_{\text{film+video}}}{\# \text{ of total projects in category}_{\text{film+video}}} \times 100\%$$

En el ejemplo anterior, vimos cómo calcular el porcentaje de éxito para los proyectos en la categoría de cine y video. Queremos calcular este porcentaje para **cada categoría** del conjunto de datos. Aunque la cantidad de proyectos en cada categoría puede variar, la ecuación debería ser la misma.

Pregunta 3: ¿Qué tareas debemos realizar para responder a esta pregunta?

Al dividir un problema, querrás asegurarte de que las tareas o pasos sean pequeños y manejables. Está bien si aún no sabes cómo resolver algunos de los pasos secundarios, lo importante es tener un punto de partida para comenzar a resolver el problema mayor. Estos son algunos de los pasos más pequeños que queremos resolver para responder a nuestra pregunta guía.



Paso 3: Introducción a Python y Pandas (20 a 25 minutos)

Los expertos en datos utilizan diversas herramientas y lenguajes para procesar y analizar datos. Algunos de estos lenguajes de programación son [R](#), [Structured Query Language \(lenguaje de consulta estructurada o SQL\)](#) y [Python](#). En esta actividad, nos centraremos en el uso del lenguaje de programación Python para ayudarnos a manipular los datos con la ayuda de una biblioteca de Python, [Pandas](#).

Python (2 minutos)



Python es un lenguaje de programación basado en texto, lo que significa que es necesario teclear todos los comandos. Muchos programadores prefieren usar Python porque es fácil de aprender y entender. Python es un lenguaje de **código abierto**, lo que significa que está disponible en forma gratuita para que el público lo utilice y modifique según sea necesario. Hay lineamientos estrictos para aceptar e implementar las actualizaciones al lenguaje, pero cualquier persona puede contribuir a su evolución.

Dado que Python es un lenguaje de código abierto, esto ha permitido que la comunidad desarrolle bibliotecas adicionales. Una **biblioteca** es una colección de comandos y variables. Una biblioteca puede facilitar el escribir código, pues podemos simplemente usar los comandos de la biblioteca para realizar una acción, en lugar de tener que escribir varias líneas de código para hacer lo mismo. En esta actividad, usaremos la biblioteca pandas. Esta biblioteca especial ha sido creada específicamente para que los expertos en datos analicen conjuntos de datos fácilmente sin tener que escribir muchas líneas de código para realizar acciones sencillas, como buscar, filtrar, comparar, modificar o eliminar información de un conjunto de datos. Antes de profundizar en el uso de pandas para realizar acciones con nuestros datos, preparemos nuestro entorno de programación en Kaggle creando una libreta nueva.

Crear una libreta nueva (5 a 8 minutos)

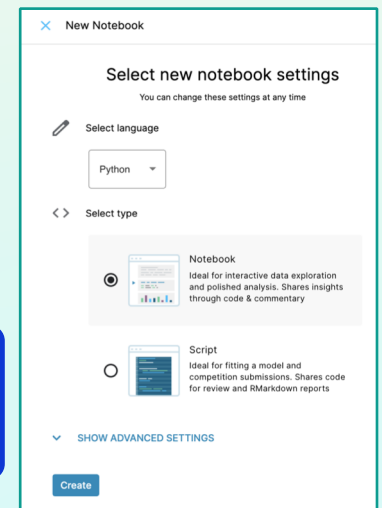
Regresemos a nuestro [conjunto de datos de Kickstarter](#).

- **Crea una libreta nueva.** Haz clic en el botón **New Notebook** (Libreta nueva) debajo de la imagen del encabezado principal a la derecha. Esto deberá llevarte a una pantalla para seleccionar la configuración de la nueva libreta. Asegúrate de que tu libreta tenga lo siguiente:

- ◆ **Lenguaje:** Python
- ◆ **Tipo:** Libreta (Notebook)

Confirma la configuración y haz clic en **Create** (Crear).

Es posible que el conjunto de datos aún se esté mostrando en vista completa. Para **minimizar** el conjunto de datos, haz clic en el **icono de caja** en la esquina superior derecha.



Si ya has programado en Python, esta “libreta” podría tener un aspecto diferente a la programación en Trinket o en otros editores de Python. Kaggle usa una herramienta llamada [Jupyter Notebook](#) para programar en Python. Jupyter Notebook es una herramienta que puede usarse para mostrar código, texto y visualización en un mismo lugar. Es fácil ejecutar bloques de código sin tener que ejecutar todo el programa.

- **Cambia el nombre de tu libreta.** Para hacerlo, haz clic en el nombre predeterminado en la parte superior izquierda de la pantalla y reemplaza el texto con el nuevo título. El título deberá reflejar algo similar a la pregunta que quieres explorar. Por ejemplo, podrías seleccionar “Índice de éxito y categorías para proyectos de Kickstarter”. También podrías optar por incluir tu nombre en el título; asegúrate de incluir solo tus iniciales o tu nombre de pila y la inicial del apellido.

Paso 3: Introducción a Python y Pandas (cont.)

Explorar el código inicial (3 a 5 minutos)

Tomémonos un momento para ver parte del código inicial incluido en la libreta. Tal vez notes varias líneas de código que comienzan con el símbolo `#` en el código inicial. Estas líneas de código son **comentarios de código**. Los programadores las utilizan para organizar su código y hacerlo más legible. En Python, todo texto escrito después de un símbolo `#` se considera un comentario de código y se le da un color verde azulado.



Ahora, veamos las primeras líneas de código, comenzando por la palabra clave `import`.

PYTHON	DESCRIPCIÓN
<pre>import numpy as np import pandas as pd</pre>	<ul style="list-style-type: none">◆ <code>import</code>: Esta palabra clave le indica a la computadora que usaremos una biblioteca de Python.◆ <code>as</code>: Esta palabra clave le asigna un sobrenombre al paquete. Es un paso opcional, pero puede facilitar la programación.◆ <code>numpy/np</code>: Esta biblioteca de Python se usa para realizar operaciones matemáticas con el conjunto de datos. Le hemos asignado a este paquete el sobrenombre de <code>np</code>.◆ <code>pandas/pd</code>: Esta biblioteca de Python se usa para convertir el conjunto de datos a un formato más útil para análisis. Le hemos asignado a este paquete el sobrenombre de <code>pd</code>.

No usaremos la biblioteca `numpy` en esta actividad. A medida que sigas explorando el análisis de datos por tu cuenta, puedes aprender más sobre la manera en que los expertos en datos utilizan la biblioteca `numpy` para mejorar su análisis de un conjunto de datos.

Para finalizar, veamos las últimas líneas de código.

PYTHON	DESCRIPCIÓN
<pre>import os for dirname, _, filenames in os.walk('/kaggle/input'): for filename in filenames: print(os.path.join(dirname, filename))</pre>	Una de las mejores características de la libreta en Kaggle es que puedes usar el conjunto de datos fácilmente, sin tener que realizar pasos adicionales. ¿Cómo? Kaggle ya asocia el conjunto de datos que te interesa con tu libreta. Estas líneas de código opcionales nos permiten saber cuál es el nombre de archivo de nuestro conjunto de datos.

Paso 3: Introducción a Python y Pandas (cont.)

Ejecutar el código (2 minutos)

Primero, haz clic en cualquier lugar dentro del bloque de código. Haz clic en el **botón azul de reproducción** ► que aparece a la izquierda de la ventana. Esto debe ejecutar todas las líneas de código en el bloque de código y mostrar las salidas debajo de la ventana. Al ejecutar estas líneas de código, se deberá obtener la siguiente salida:

```
/kaggle/input/kickstarter-projects/ks-projects-201801.csv  
/kaggle/input/kickstarter-projects/ks-projects-201612.csv
```

Estos son los nombres de archivo de nuestro conjunto de datos. Ahora que entendemos un poco acerca de lo que incluye el código inicial y cómo ejecutar código en nuestra libreta, comencemos a codificar.

Importar el conjunto de datos de Kickstarter (10 a 15 minutos)

Ahora que conocemos los nombres de archivo de nuestro conjunto de datos, necesitamos importar los datos a nuestro programa. En este momento es un archivo aparte, pero necesitamos conectar esta información con nuestro programa de Python. Para hacerlo, usaremos el método `read_csv()` de la biblioteca `pandas` para ayudarnos. Recuerda que un **método o función** es un conjunto de instrucciones (líneas de código) que llevan a cabo una tarea específica. Analicemos la sintaxis de este método.

CSV, que en inglés es una abreviatura de “valores separados por comas”, es un tipo de archivo que se usa para contener datos en forma de texto simple. La biblioteca `pandas` puede leer archivos CSV fácilmente y convertirlos a un formato similar a una tabla que podemos leer y manipular con facilidad.

PYTHON	DESCRIPCIÓN
<code>pd.read_csv("filename")</code>	<ul style="list-style-type: none">◆ pd: Esta palabra clave le indica a la computadora que usaremos un método de la biblioteca <code>pandas</code>. Usamos <code>pd</code> en lugar de <code>pandas</code> porque es el sobrenombre que le asignamos a la biblioteca al importarla al inicio del programa.◆ .: Este símbolo le indica a la computadora que estamos usando un método.◆ read_csv(): Este método de <code>pandas</code> lee un archivo CSV y lo convierte en formato de tabla para que sea más fácil de usar en Python.◆ "filename" (nombre de archivo): Debemos indicarle a la computadora qué archivo abrir. Agrega aquí el <code>filename</code> (nombre de archivo) del archivo CSV. Incluimos el nombre entre comillas (pueden ser sencillas o dobles) dado que es un nombre.

Paso 3: Introducción a Python y Pandas (cont.)

- **Agrega un nuevo bloque de código.** Los bloques de código son una excelente manera de organizar tu código para que corresponda a los subproblemas que definimos al descomponer el problema. Es probable que ya tengas un bloque de código vacío en tu libreta, indicado por un cuadro de color gris claro con el símbolo [] a la izquierda del cuadro.

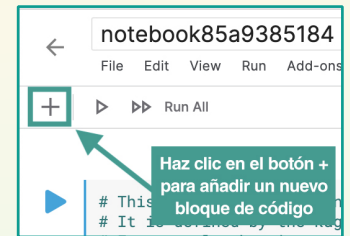
Hay dos opciones para añadir un nuevo bloque de código:

- ◆ Haz clic en el botón + en el menú superior de la libreta.
- ◆ Coloca el puntero del ratón debajo del último bloque de código. Deberás ver una opción que dice "+ Code" (+ código). Selecciona este botón para crear un nuevo bloque de código debajo.

- **Importa el conjunto de datos de Kickstarter.** Ahora que tenemos un nuevo bloque de código, es hora de usar el método `read_csv()` para importar el conjunto de datos. Pero espera, necesitamos el nombre de archivo del conjunto de datos. Recordarás que después de ejecutar el primer bloque, se obtienen los nombres de archivo de nuestro conjunto de datos. De hecho, se obtienen dos nombres para los conjuntos de datos 2016 y 2018. Solo usaremos el conjunto de datos 2018 de Kickstarter, ya que contiene la información más reciente. En el nuevo bloque de código, usa el método `read_csv()` para importar el conjunto de datos 2018. **Copia y pega** el nombre de archivo del conjunto de datos 2018 de la salida del bloque de código anterior.

```
pd.read_csv("/kaggle/input/kickstarter-projects/ks-projects-201801.csv")
```

- **Ejecuta el bloque de código.** Haz clic en el botón de reproducción azul que se encuentra a la izquierda del bloque de código para ejecutar el código. Deberás ver una perspectiva (similar a lo que viste en el explorador de datos) del conjunto de datos.



RESULTADOS

ID	name	category	main_category	currency
0 1000002330	The Songs of Adelaide & Abullah	Poetry	Publishing	GBP
1 1000003930	Greeting From Earth: Z20AC Arts Capsule For ET	Narrative Film	Film & Video	USD
2 1000004038	Where is Hank?	Narrative Film	Film & Video	USD
3 1000007540	ToshCapital Records Needs Help to Complete Album	Music	Music	USD
4 1000011046	Community Film Project: The Art of Neighborhood...	Film & Video	Film & Video	USD
...
378656 999976400	ChknTruk Nationwide Charity Drive 2014 (Cancelled)	Documentary	Film & Video	USD
378657 999977640	The Tribe	Narrative Film	Film & Video	USD
378658 99998353	Walls of Remedy: New Lesbian Romantic Comedy f...	Narrative Film	Film & Video	USD
378659 999987933	BioDefense Education Kit	Technology	Technology	USD
378660 999988282	Nou Renmen Ayiti! We Love Haiti!	Performance Art	Art	USD

378661 rows x 5 columns

CONSEJOS DE DEPURACIÓN

- ◆ Comprueba que hayas incluido el nombre correcto del archivo 2018: `/kaggle/input/kickstarter-projects/ks-projects-201801.csv`
- ◆ Asegúrate de que el nombre de archivo esté entre comillas
- ◆ Recuerda que en Python, ¡la ortografía cuenta! Verifica no solo que todo esté bien escrito, también el uso correcto de mayúsculas y minúsculas.
- ◆ Comprueba que incluyas un punto al invocar un método de pandas.
- ◆ Verifica que no tengas paréntesis () de más. Tal vez notes que al teclear un símbolo (, la libreta añade automáticamente el paréntesis de cierre). Esto podría provocar que agregues paréntesis o corchetes adicionales por accidente.

Paso 3: Introducción a Python y Pandas (cont.)

- **Guarda tu conjunto de datos en una variable llamada `ds`.** ¡Ya casi terminamos! Ahora que tenemos el conjunto de datos enlazado con nuestro programa en Python, debemos almacenarlo en una variable. Recuerda que las [variables](#) se usan para almacenar información (datos) en un programa de computación. Almacena el conjunto de datos en una variable llamada `ds` (abreviatura de conjunto de datos en inglés).

```
ds = pd.read_csv("/kaggle/input/kickstarter-projects/ks-projects-201801.csv")
```

- **Usa el método `info()` para imprimir la información sobre el conjunto de datos.** Podemos usar el método `info()` para obtener una perspectiva rápida de las características y la cantidad de filas en este conjunto de datos.

PYTHON	DESCRIPCIÓN
<code>ds.info()</code>	<ul style="list-style-type: none">◆ <code>ds</code>: Usamos el método <code>info()</code> con nuestra variable, <code>ds</code>, NO con toda la biblioteca pandas. La razón de esto es que queremos obtener información sobre nuestro conjunto de datos específico.◆ <code>info()</code>: Este método en pandas obtiene información sobre el conjunto de datos, incluidas las características, filas y tipo de datos.

- **Ejecuta el bloque de código.** Haz clic en el botón de reproducción azul que se encuentra a la izquierda del bloque de código para ejecutar el código. Esto debe dar como resultado cierta información sobre el conjunto de datos, como el número de entradas (o número de proyectos de Kickstarter) y el número de características. También incluye el [tipo de valor](#) de cada característica (número o palabras) y el número de entradas “**no nulas**” o no vacías.

RESULTADOS

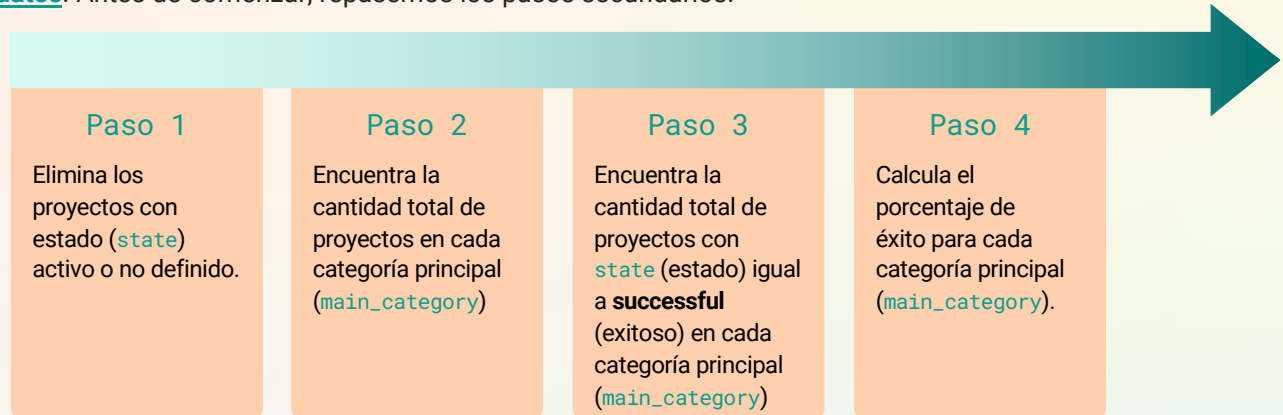
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 378661 entries, 0 to 378660
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype
---  -
0   ID                    378661 non-null  int64
1   name                  378657 non-null  object
2   category              378661 non-null  object
3   main_category         378661 non-null  object
4   currency              378661 non-null  object
5   deadline              378661 non-null  object
6   goal                  378661 non-null  float64
7   launched              378661 non-null  object
8   pledged               378661 non-null  float64
9   state                 378661 non-null  object
10  backers               378661 non-null  int64
11  country               378661 non-null  object
12  usd_pledged           378661 non-null  float64
13  usd_pledged_real      378661 non-null  float64
14  usd_goal_real          378661 non-null  float64
dtypes: float64(5), int64(2), object(8)
memory usage: 43.3+ MB
```

CONSEJOS DE DEPURACIÓN

- ◆ Nombre ‘ds’ no definido: Cada bloque de código en la libreta se ejecuta por separado; por lo tanto, la libreta en ocasiones puede perder su lugar en el programa y requerir que **ejecutes todos** los bloques de código. Para hacerlo, haz clic en el botón de flecha doble ▶ en la parte superior de la libreta para ejecutar todos los bloques de código.
- ◆ Recuerda que en Python, ¡la ortografía cuenta! Verifica no solo que todo esté bien escrito, también el uso correcto de mayúsculas y minúsculas.
- ◆ Comprueba que incluyas un punto al invocar un método.

Paso 3: Modificar los datos (25 a 35 minutos)

Antes de comenzar a calcular el índice de éxito del conjunto de datos, debemos asegurarnos de limpiar los datos. Los expertos en datos primero deben revisar si hay valores que pudieran causar errores en su análisis, como valores duplicados, errores ortográficos, o valores vacíos. Este proceso se conoce como depuración de datos. Antes de comenzar, repasemos los pasos secundarios.



Al abrir el conjunto de datos de Kickstarter en Kaggle, quizá hayas notado que hay un puntaje de **facilidad de uso**. El puntaje de facilidad de uso nos indica cuán *depurados* están los datos. Este conjunto de datos tiene un puntaje de facilidad de uso de 7.9, que es muy alto. Aún debemos llevar a cabo una depuración adicional del conjunto de datos antes de comenzar a calcular el porcentaje de éxito para los datos.

Recuerda que en este conjunto de datos hay proyectos con `estado` de éxito, cancelado, no definido, fallido o activo. Al calcular el índice de éxito de los proyectos, queremos centrarnos en aquellos que ya han concluido. Esto significa que no queremos incluir proyectos considerados **activos** o **no definidos**, pues aún no conocemos su resultado final. Antes de ver cómo lograr esto, aprendamos un poco más sobre la estructura de nuestro conjunto de datos en `pandas`.

DataFrame de Pandas (10 a 15 minutos)

En `pandas`, los datos se almacenan en un objeto similar a una tabla, conocido como **DataFrame** o marco de datos. Almacena los datos de manera similar a cómo los vemos en el explorador de datos de Kaggle. Un DataFrame imita las filas y columnas de la estructura de datos. Para acceder a los datos, usamos símbolos `[]`. Esto es muy similar a las *matrices* en JavaScript y las *listas* en Python.



Con un total de **15 características** o columnas, tal vez no queramos ver todos estos datos. Dado que nos centraremos en solo dos de las características, categoría principal (`main_category`) y estado (`state`), hagamos que el conjunto de datos muestre solo esta información. Construyamos algunos de los componentes para *seleccionar* y mostrar estas columnas.

- **Agrega un nuevo bloque de código al final de la libreta.** Coloca el puntero del ratón debajo del último bloque de código y presiona el botón **+Code** (+código) o presiona el botón **+** en la parte superior izquierda de la libreta.

Paso 4: Modificar los datos (cont.)

- Crea una **lista** con los nombres de las características **deseadas**. Usaremos los símbolos `[]` para añadir los nombres de las características entre los corchetes, separados por comas.

En Python, una **lista** es una estructura de datos ordenada que contiene información. Una lista asigna un número, o **índice**, para que sea más fácil acceder, eliminar o reemplazar valores.

PYTHON	DESCRIPCIÓN
<code>["main_category", "state"]</code>	<ul style="list-style-type: none">◆ <code>[]</code>: los corchetes indican que estamos creando una lista en Python.◆ <code>"main_category", "state"</code>: Incluimos los nombres de las características que nos interesan. Como son nombres, encerramos cada nombre de función entre comillas y separamos los valores mediante comas.

- Usa la **variable de conjunto de datos**, **ds**, para **invocar la lista de nombres de características**. Recuerda que almacenamos la referencia a nuestro conjunto de datos en una variable llamada **ds**. Aquí, usamos los símbolos `[]` para indicar que queremos seleccionar información del conjunto de datos e incluir nuestra lista de características *adentro*.

PYTHON	DESCRIPCIÓN
<code>ds[["main_category", "state"]]</code>	<ul style="list-style-type: none">◆ <code>ds[]</code>: Invocamos la variable ds, que contiene la referencia a nuestro conjunto de datos, y usamos los símbolos <code>[]</code> para indicar que queremos acceder a información del conjunto de datos.◆ <code>["main_category", "state"]</code>: Incluimos esta lista de características para indicarle a la computadora qué columnas queremos del conjunto de datos. Esta información va dentro de los corchetes que aparecen después de ds, la variable que almacena nuestro conjunto de datos.

- **Ejecuta el bloque de código**. Haz clic en el botón de reproducción azul que se encuentra a la izquierda del bloque de código para ejecutar el código. Deberás ver una perspectiva de todos los datos en el conjunto de datos, donde solo se muestran las características **main_category** y **state**. Si el código no se ejecuta de manera correcta, revisa lo siguiente:

RESULTADOS

	main_category	state
0	Publishing	failed
1	Film & Video	failed
2	Film & Video	failed
3	Music	failed
4	Film & Video	canceled
...
378656	Film & Video	canceled
378657	Film & Video	failed
378658	Film & Video	failed
378659	Technology	failed
378660	Art	failed
378661 rows x 2 columns		

CONSEJOS DE DEPURACIÓN

- ◆ Nombre 'ds' no definido: Cada bloque de código en la libreta se ejecuta por separado; por lo tanto, la libreta en ocasiones puede perder su lugar en el programa y requerir que **ejecutes todos** los bloques de código. Para hacerlo, haz clic en el botón de flecha doble `⏮` en la parte superior de la libreta para ejecutar todos los bloques de código.
- ◆ Verifica que todos los nombres de características estén entre comillas.
- ◆ Verifica que todos los nombres de características estén bien escritos. Recuerda que Python distingue entre mayúsculas y minúsculas.
- ◆ Comprueba que no tengas corchetes `[]` de más. Tal vez notes que, al teclear un símbolo `[`, la libreta agrega automáticamente el corchete de cierre `]`. Esto podría provocar que agregues paréntesis o corchetes adicionales por accidente. En total, debes tener dos conjuntos de corchetes: uno alrededor de los nombres de características y el otro alrededor de la lista de características.

Paso 4: Modificar los datos (cont.)

Eliminar los proyectos activos o no definidos (15 a 20 minutos)

Para eliminar los proyectos que no queremos incluir en nuestro análisis, debemos usar sentencias condicionales en Python para indicarle a la computadora qué valores de datos queremos. Para preparar el conjunto de datos para nuestro análisis, queremos eliminar los proyectos que no han finalizado. Esto significa eliminar los proyectos con **estado (state) no definido (undefined)** o **activo (live)**. Antes de eliminar estos proyectos, veamos la cantidad de proyectos en cada estado. Para hacerlo, usaremos el método value_counts().

→ Usa el método `value_counts()` para la característica `state`.

PYTHON	DESCRIPCIÓN
<code>ds["state"].value_counts()</code>	<ul style="list-style-type: none">◆ <code>ds["state"]</code>: Queremos conocer el detalle de los proyectos por estado (state). Usaremos los símbolos <code>[]</code> para seleccionar únicamente la característica <code>state</code> en el conjunto de datos y comillas dobles alrededor de la característica para indicar que es un nombre.◆ <code>.</code>: Este símbolo le indica a la computadora que estamos usando un método.◆ <code>value_counts()</code>: Este método devuelve el número de entidades (o filas) que tienen un valor único de la característica.

→ **Ejecuta el bloque de código.** Haz clic en el botón de reproducción azul que se encuentra a la izquierda del bloque de código para ejecutar el código. Debes obtener un resultado idéntico al detalle presentado a continuación:

RESULTADOS	CONSEJOS DE DEPURACIÓN												
<table><tr><td>fallido</td><td>197719</td></tr><tr><td>exitoso</td><td>133956</td></tr><tr><td>cancelado</td><td>38779</td></tr><tr><td>no definido</td><td>3562</td></tr><tr><td>activo</td><td>2799</td></tr><tr><td>suspendido</td><td>1846</td></tr></table>	fallido	197719	exitoso	133956	cancelado	38779	no definido	3562	activo	2799	suspendido	1846	<ul style="list-style-type: none">◆ El nombre 'ds' no está definido: Cada bloque de código en la libreta se ejecuta por separado; por lo tanto, la libreta en ocasiones puede perder su lugar en el programa y requerir que ejecutes todos los bloques de código. Para hacerlo, haz clic en el botón de flecha doble ⇨ en la parte superior de la libreta para ejecutar todos los bloques de código.◆ Verifica que todos los nombres de características estén entre comillas.◆ Verifica que todos los nombres de características estén bien escritos. Recuerda que Python distingue entre mayúsculas y minúsculas.◆ Comprueba que no tengas corchetes <code>[]</code> de más.
fallido	197719												
exitoso	133956												
cancelado	38779												
no definido	3562												
activo	2799												
suspendido	1846												

Paso 4: Modificar los datos (cont.)

Al intentar eliminar los proyectos **activos** y **no definidos** de nuestro conjunto de datos, podemos comparar el detalle de proyectos con estos valores originales. Para filtrar los datos que no queremos incluir, debemos usar una combinación de sentencias condicionales. Recuerda que los [enunciados condicionales](#) ejecutan código **si** se cumple una condición o conjunto de reglas. Toma un momento para pensar en cómo podemos expresar lo que queremos en la forma de un enunciado condicional **verdadero**. Desglosemos cómo podríamos escribir nuestra regla como un enunciado condicional de Python.

Ejemplo

REGLA	CONDICIÓN DE PYTHON
Ejemplo: Proyectos con estado exitoso	<code>ds["state"] == "successful"</code>

- `ds["state"]`: Queremos aplicar un filtro al conjunto de datos. Usamos nuestra variable de conjunto de datos, `ds`, y luego corchetes `[]` para especificar el filtrado de la característica `"state"`.
- `== "successful"`: Dado que solo queremos proyectos exitosos, usamos el símbolo `==`, que busca si el estado es igual a `"successful"` (exitoso)

Toma un momento para reescribir en la siguiente tabla las demás reglas como enunciados condicionales en Python. ¿Necesitas un repaso de los enunciados condicionales o los operadores de comparación? Consulta este [tutorial de W3 School](#) sobre sentencias condicionales en Python. Dado que queremos aplicar un filtro al conjunto de datos, usamos nuestra variable de conjunto de datos, `ds`, y luego corchetes `[]` para especificar el filtrado de la característica `"state"`.

REGLA	CONDICIÓN DE PYTHON
Proyectos que no tienen estado "undefined" (no definido)	<code>ds["state"]</code>
Proyectos que no tienen estado "live" (activo)	<code>ds["state"]</code>



Haz una **pausa** aquí antes de revelar nuestras soluciones en la siguiente página. Hay muchas maneras en las que puedes escribir un enunciado condicional para representar las reglas que queremos. Nuestra solución solo ofrece una de ellas, pero hay múltiples soluciones.

Paso 4: Modificar los datos (cont.)

Regla n.º 1: Proyectos que no tienen estado "undefined" (no definido)

REGLA	CONDICIÓN DE PYTHON
Proyectos que no tienen estado "undefined" (no definido)	<code>ds["state"] != "undefined"</code>

Como queremos proyectos que no tengan estado "undefined" (no definido), usamos el operador `!=`. De modo alternativo, podrías haber escrito un enunciado condicional que revise si el estado del proyecto es igual a fallido, exitoso, cancelado o suspendido.

Regla n.º 2: Proyectos que no tienen estado "live" (activo)

REGLA	CONDICIÓN DE PYTHON
Proyectos que no tienen estado "undefined" (no definido)	<code>ds["state"] != "undefined"</code>

De modo similar al ejemplo, usamos el operador `!=` para encontrar proyectos con estado distinto de "live" (activo). Ahora que sabemos qué sentencias condicionales incluir, vamos a programarlas en nuestra libreta.

- **Agrega un nuevo bloque de código al final de la libreta.** Coloca el puntero del ratón sobre el último bloque de código y presiona el botón `+Code` (+código) o presiona el botón `+` en la parte superior izquierda de la libreta.
- **Escribe una sentencia condicional para eliminar los proyectos que tenga un estado (state) de "undefined" (no definido) o "live" (activo).** Ya revisamos la sentencia condicional para filtrar los proyectos con `estado (state)` "undefined" (no definido) o "live" (activo), pero para *combinar* estas reglas en **un** enunciado condicional necesitamos usar el operador **and** (y). Simplemente usamos el símbolo `&` para representar "y" en pandas.

El operador **and** (y) representado de manera diferente en pandas que en un enunciado condicional regular de Python. Como estamos usando un operador **and** (y) en el DataFrame, utilizamos el símbolo `&`. Si escribiéramos un enunciado condicional por separado del DataFrame, utilizaríamos en cambio la palabra clave **and**.

```
(ds["state"] != "undefined") & (ds["state"] != "live")
```

- ◆ `&`: esta palabra clave representa "and" (y) y nos ayuda a combinar dos condiciones.
- ◆ `()`: usamos paréntesis alrededor de las sentencias condicionales para distinguir las dos condiciones que queremos revisar.

Paso 4: Modificar los datos (cont.)

- **Aplica el enunciado condicional para filtrar el conjunto de datos.** Aquí, usaremos la variable `ds` y los símbolos `[]` para aplicar los enunciados condicionales del paso anterior. Recuerda que el enunciado condicional debe estar *dentro* de los corchetes `[]`.

```
ds[(ds["state"] != "undefined") & (ds["state"] != "live")]
```

- **Almacena este nuevo conjunto de datos filtrado en una nueva variable llamada `compProj`.** Queremos almacenar este conjunto de datos filtrado en una nueva variable, para que no modifiquemos el conjunto de datos original. Le asignamos el nombre `compProj` a nuestra variable, para representar los proyectos completados.

```
compProj = ds[(ds["state"] != "undefined") & (ds["state"] != "live")]
```

- **Usa el método `value_counts()` para ver el detalle de los proyectos en los datos filtrados.**

```
compProj["state"].value_counts()
```

- **Ejecuta el bloque de código.** Haz clic en el botón de reproducción azul que se encuentra a la izquierda del bloque de código para ejecutar el código. Deberías obtener los siguientes resultados:

RESULTADOS	CONSEJOS DE DEPURACIÓN								
<table><tr><td>fallido</td><td>197719</td></tr><tr><td>Exitoso</td><td>133956</td></tr><tr><td>cancelado</td><td>38779</td></tr><tr><td>suspendido</td><td>1846</td></tr></table>	fallido	197719	Exitoso	133956	cancelado	38779	suspendido	1846	<ul style="list-style-type: none">◆ El nombre 'ds' no está definido: Cada bloque de código en la libreta se ejecuta por separado; por lo tanto, la libreta en ocasiones puede perder su lugar en el programa y requerir que ejecutes todos los bloques de código. Para hacerlo, haz clic en el botón de flecha doble ➡ en la parte superior de la libreta para ejecutar todos los bloques de código.◆ Verifica que todos los nombres de características estén entre comillas.◆ Verifica que todos los nombres de características estén bien escritos. Recuerda que Python distingue entre mayúsculas y minúsculas.◆ Comprueba que no tengas corchetes <code>[]</code> de más.◆ Verifica que cada sentencia condicional esté entre paréntesis <code>()</code>.
fallido	197719								
Exitoso	133956								
cancelado	38779								
suspendido	1846								

Observa que ya no aparecen proyectos con estado "undefined" o "live" en el conjunto de datos filtrado. También deberás notar que la cantidad de proyectos con estado "failed", "successful", "canceled" o "suspended" no ha cambiado. Si se produce un error, revisa lo siguiente

Vaya, ¡ya hemos logrado mucho! Ahora que hemos depurado los datos, es el momento de ponernos la gorra analítica y comenzar a calcular el índice de éxito.

Paso 5: Calcular el porcentaje de éxito (15 a 20 minutos)

Ahora que hemos depurado los datos, podemos comenzar a calcular el porcentaje de éxito. Recordemos los pasos secundarios.



Aunque no lo creas, ya sabes cómo codificar los pasos que faltan. Usaremos una combinación del método `value_counts()` para obtener el número de proyectos en cada categoría y **sentencias condicionales** para filtrar únicamente los proyectos exitosos. No te preocupes, te guiaremos por todos los pasos.

Recuerda: para calcular el porcentaje de éxito, necesitamos comparar el número de proyectos exitosos con el número total de proyectos en cada categoría.

$$\% \text{ of success for category}_{\text{film+video}} = \frac{\# \text{ of successful projects in category}_{\text{film+video}}}{\# \text{ of total projects in category}_{\text{film+video}}} \times 100\%$$

En el ejemplo anterior, vimos cómo calcular el porcentaje de éxito para los proyectos en la categoría de cine y video. Queremos calcular este porcentaje para **cada categoría** del conjunto de datos. Aunque la cantidad de proyectos en cada categoría puede variar, la ecuación debería ser la misma.

Primero, encontraremos el número de proyectos exitosos en cada categoría. Luego, encontraremos el número total de proyectos en cada categoría. Por último, usaremos esos dos valores para calcular el porcentaje de éxito.

Paso 5: Calcular el porcentaje de éxito (cont.)

Número total de proyectos por categoría (3 a 5 minutos)

→ **Agrega un nuevo bloque de código al final de la libreta.** Coloca el puntero del ratón sobre el último bloque de código y presiona el botón **+Code** (+código) o presiona el botón **+** en la parte superior izquierda de la libreta.

→ **Almacena el número total de proyectos por categoría principal (`main_category`) en una variable llamada `totProjCount`.** Usaremos el método `value_counts()` para la característica "`main_category`" del conjunto de datos filtrado, `compProj`. El método `value_counts()` devuelve un objeto `Serie`, que es como una lista de los valores. Esto nos facilitará calcular el porcentaje de éxito de cada categoría con una sola línea de código. Le asignamos el nombre `totProjCount` a la variable, para representar el número total de proyectos.

Anteriormente, usamos el método `value_counts()` para determinar el número de proyectos para cada tipo de `estado`. Ahora, queremos dividir el conjunto de datos en categorías, de modo que añadimos la característica "`main_category`" (categoría principal).

→ **Imprime `totProjCount` y ejecuta el bloque de código.** Este es un paso opcional, pero es buena idea verificar que podamos obtener el número total correcto de proyectos antes de continuar. Usa el método `print()` para imprimir `totProjCount` y luego haz clic en el botón de reproducción de color azul a la izquierda del bloque de código para ejecutar el código.

```
print(totProjCount)
```

RESULTADOS

Película y video	62399
Música	49403
Publicación	39113
Juegos	34943
Tecnología	32189
Diseño	29763
Arte	27959
Comida	24418
Moda	22563
Teatro	10871
Cómics	10743
Fotografía	10730
Manualidades	8733
Periodismo	4724
Baile	3749

CONSEJOS DE DEPURACIÓN

- ◆ Tal vez necesites volver a ejecutar todos los bloques de código. Haz clic en el botón de doble flecha **↺↻** en la parte superior de la libreta.
- ◆ Comprueba que estés usando el conjunto de datos filtrado, `compProj`.
- ◆ Verifica que todos los nombres de características estén entre comillas.
- ◆ Verifica que todos los nombres (características, tabla y métodos) estén bien escritos. Recuerda que Python distingue entre mayúsculas y minúsculas.
- ◆ Comprueba que no tengas corchetes `[]` de más.
- ◆ Comprueba que hayas incluido paréntesis después de usar el método `value_counts`.

Número de proyectos exitosos por categoría (3 a 5 minutos)

- **Agrega un nuevo bloque de código al final de la libreta.** Coloca el puntero del ratón sobre el último bloque de código y presiona el botón **+Code** (+código) o presiona el botón **+** en la parte superior izquierda de la libreta.
- **Almacena en una variable llamada `susProj` una nueva referencia al conjunto de datos que contenga solamente proyectos exitosos.** Filtremos el conjunto de datos `compProj` para obtener solo los proyectos con estado "successful" (exitoso).

```
susProj = compProj[compProj["state"] == "successful"]
```

- **Almacena el número total de proyectos exitosos por categoría principal (`main_category`) en una variable llamada `susProjCount`.** Queremos asegurarnos de utilizar el método `value_counts()` con la característica "`main_category`" (categoría principal) en nuestro conjunto de datos filtrado con solamente proyectos exitosos, `susProj`.

```
susProjCount = susProj["main_category"].value_counts()
```


- **Imprime `susProjCount` y ejecuta el bloque de código.** Este es un paso opcional, pero es buena idea verificar que podamos obtener el número total correcto de proyectos antes de continuar. Usa el método `print()` para imprimir `susProjCount` y luego haz clic en el botón de reproducción de color azul a la izquierda del bloque de código para ejecutar el código.

```
print(susProjCount)
```

RESULTADOS

Música	24197
Película y video	23623
Juegos	12518
Publicación	12300
Arte	11510
Diseño	10550
Teatro	6534
Tecnología	6434
Comida	6085
Cómics	5842
Moda	5593
Fotografía	3305
Baile	2338
Manualidades	2115
Periodismo	1012

CONSEJOS DE DEPURACIÓN

- ◆ Tal vez necesites volver a ejecutar todos los bloques de código. Haz clic en el botón de doble flecha  en la parte superior de la libreta.
- ◆ Comprueba que estés usando el conjunto de datos filtrado, `susProj`.
- ◆ Verifica que todos los nombres de características estén entre comillas.
- ◆ Verifica que todos los nombres (características, tabla y métodos) estén bien escritos. Recuerda que Python distingue entre mayúsculas y minúsculas.
- ◆ Comprueba que no tengas corchetes `[]` de más.
- ◆ Comprueba que hayas incluido paréntesis después de usar el método `value_counts`.

Paso 5: Calcular el porcentaje de éxito (cont.)

Calcular el porcentaje de éxito (3 a 5 minutos)

- **Agrega un nuevo bloque de código al final de la libreta.**
- **Usa las variables `susProjCount` y `totProjCount` para calcular el porcentaje de éxito.** Una de las mejores características de `pandas` es que puedes llevar a cabo el mismo cálculo para varias características con una sola línea de código. Podemos calcular el porcentaje de éxito de cada categoría solo dividir `susProjCount` por `totProjCount`. No olvides multiplicar por 100 para obtener el valor porcentual.

`susProjCount/totProjCount * 100`

- **Ejecuta el bloque de código.** Haz clic en el botón de reproducción azul que se encuentra a la izquierda del bloque de código para ejecutar el código.

RESULTADOS

Arte	41.167424
Cómics	54.379596
Manualidades	24.218482
Baile	62.363297
Diseño	35.446696
Moda	24.788370
Película y video	37.857978
Comida	24.920141
Juegos	35.824056
Periodismo	21.422523
Música	48.978807
Fotografía	30.801491
Publicación	31.447345
Tecnología	19.988195
Teatro	60.104866

CONSEJOS DE DEPURACIÓN

- ◆ Tal vez necesites volver a ejecutar todos los bloques de código. Haz clic en el botón de doble flecha `⏮` en la parte superior de la libreta.
- ◆ Verifica que todos los nombres estén bien escritos. Recuerda que Python distingue entre mayúsculas y minúsculas.
- ◆ No olvides multiplicar por **100** para obtener la respuesta en forma porcentual.

Paso 5: Calcular el porcentaje de éxito (cont.)

Ordenar los resultados (3 a 5 minutos)

Tal vez hayas notado que los resultados finales están en orden alfabético por nombre de la categoría. Puedes ordenar los resultados utilizando el método `sort_values()`. Hagamos unos cambios a nuestro bloque de código anterior para producir los valores ordenados.


- Almacena el porcentaje de éxito en una variable llamada `percentSuccess`.

```
percentSuccess = susProjCount/totProjCount * 100
```

- Use el método `sort_values()` para ordenar los porcentajes.

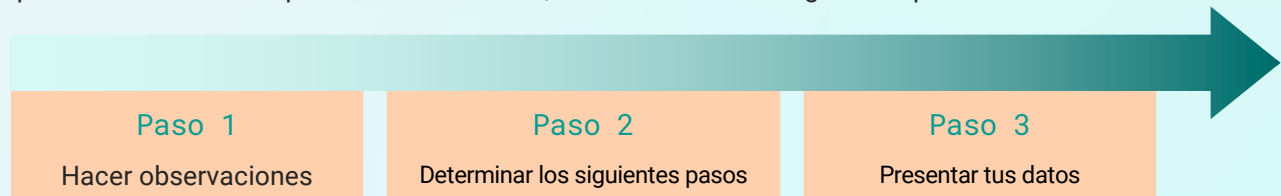
```
percentSuccess.sort_values()
```

- Ejecuta el bloque de código. Haz clic en el botón de reproducción azul que se encuentra a la izquierda del bloque de código para ejecutar el código.

RESULTADOS	CONSEJOS DE DEPURACIÓN																														
<table><tr><td>Tecnología</td><td>19.988195</td></tr><tr><td>Periodismo</td><td>21.422523</td></tr><tr><td>Manualidades</td><td>24.218482</td></tr><tr><td>Moda</td><td>24.788370</td></tr><tr><td>Comida</td><td>24.920141</td></tr><tr><td>Fotografía</td><td>30.801491</td></tr><tr><td>Publicación</td><td>31.447345</td></tr><tr><td>Diseño</td><td>35.446696</td></tr><tr><td>Juegos</td><td>35.824056</td></tr><tr><td>Película y video</td><td>37.857978</td></tr><tr><td>Arte</td><td>41.167424</td></tr><tr><td>Música</td><td>48.978807</td></tr><tr><td>Cómics</td><td>54.379596</td></tr><tr><td>Teatro</td><td>60.104866</td></tr><tr><td>Baile</td><td>62.363297</td></tr></table>	Tecnología	19.988195	Periodismo	21.422523	Manualidades	24.218482	Moda	24.788370	Comida	24.920141	Fotografía	30.801491	Publicación	31.447345	Diseño	35.446696	Juegos	35.824056	Película y video	37.857978	Arte	41.167424	Música	48.978807	Cómics	54.379596	Teatro	60.104866	Baile	62.363297	<ul style="list-style-type: none">◆ Tal vez necesites volver a ejecutar todos los bloques de código. Haz clic en el botón de doble flecha  en la parte superior de la libreta.◆ Verifica que todos los nombres estén bien escritos. Recuerda que Python distingue entre mayúsculas y minúsculas.
Tecnología	19.988195																														
Periodismo	21.422523																														
Manualidades	24.218482																														
Moda	24.788370																														
Comida	24.920141																														
Fotografía	30.801491																														
Publicación	31.447345																														
Diseño	35.446696																														
Juegos	35.824056																														
Película y video	37.857978																														
Arte	41.167424																														
Música	48.978807																														
Cómics	54.379596																														
Teatro	60.104866																														
Baile	62.363297																														

Paso 5: Reflexionar sobre los resultados finales (15 a 20 minutos)

Es sumamente importante realizar una reflexión detallada de tu análisis al hacer inferencias sobre tendencias futuras. Es igual de importante que todos los demás pasos de tu proceso de análisis. A partir de una reflexión, te centrarás en los siguientes pasos:



En esta sección, te guiaremos por una perspectiva del proceso de reflexión que los expertos en datos llevan a cabo después de realizar su análisis de macrodatos. En esta actividad solo te guiaremos por los pasos 1 y 2 del proceso. Mantente atenta la próxima semana, cuando veamos cómo usar diversas técnicas de visualización en Python para presentar tus datos.

Paso 6: Calcular el porcentaje de éxito (cont.)

Hacer observaciones (6 a 10 minutos)

Tómate de **3 a 5 minutos** para ver los resultados finales y documentar tus observaciones. Estamos usando el tiempo como restricción para que te centres en los datos que más sobresalen; siempre puedes volver a ellos más adelante, si lo deseas. Durante este tiempo, deberás anotar las observaciones que hagas con respecto a los datos. Céntrate únicamente en los hechos o datos y procura no realizar observaciones.

OBSERVACIÓN	FUENTE DE DATOS
<i>¿Qué datos sobresalieron? Apégate a los hechos de los datos y no presentes conclusiones en este momento.</i>	<i>¿Dónde hallaste esta información? Incluye el nombre del conjunto de datos o la línea de código.</i>
<i>Ejemplo: La categoría "Film & Video" (Cine y video) tiene la mayor cantidad de proyectos, con un total de 62,399.</i>	<i>Ejemplo: En totProjCount. Recuento total del conjunto de datos filtrado, eliminando los proyectos con estado "undefined" (no definido) y "live" (activo).</i>

Toma un momento para revisar las observaciones que anotaste en la tabla. En esta sección, responderemos a nuestra pregunta principal: **¿Qué categoría principal de proyectos tiene el mayor índice de éxito?**

Ajusta el temporizador a **3 minutos** para responder a nuestra pregunta principal e identificar cualquier otra cosa que te pareció interesante y por qué. Tal vez quieras considerar que crees que dicen los datos acerca de los proyectos de Kickstarter en general y sus categorías. Piensa desde la perspectiva de un patrocinador: si fueras a prometer una donación a un proyecto, ¿qué categoría de proyecto considerarías que tendría potencial de éxito?

Determinar los siguientes pasos (3 minutos)

Ahora que has hecho algunas observaciones sobre los datos, ¿qué sigue en tu trabajo? En esta sección, tomarás tiempo para pensar en algunas de las restricciones que tuviste en tu análisis y otros factores que te gustaría explorar con mayor detalle en este conjunto de datos. ¿Qué información adicional te resultaría útil en tu investigación? Toma **3 minutos** para reflexionar sobre las siguientes preguntas:

PREGUNTA 1	PREGUNTA 2	PREGUNTA 3
¿Cuáles son algunos de los desafíos que enfrentaste al analizar los datos?	¿Qué preguntas adicionales te gustaría responder acerca de los datos?	¿Qué información adicional necesitarías para llegar más lejos con tu análisis?

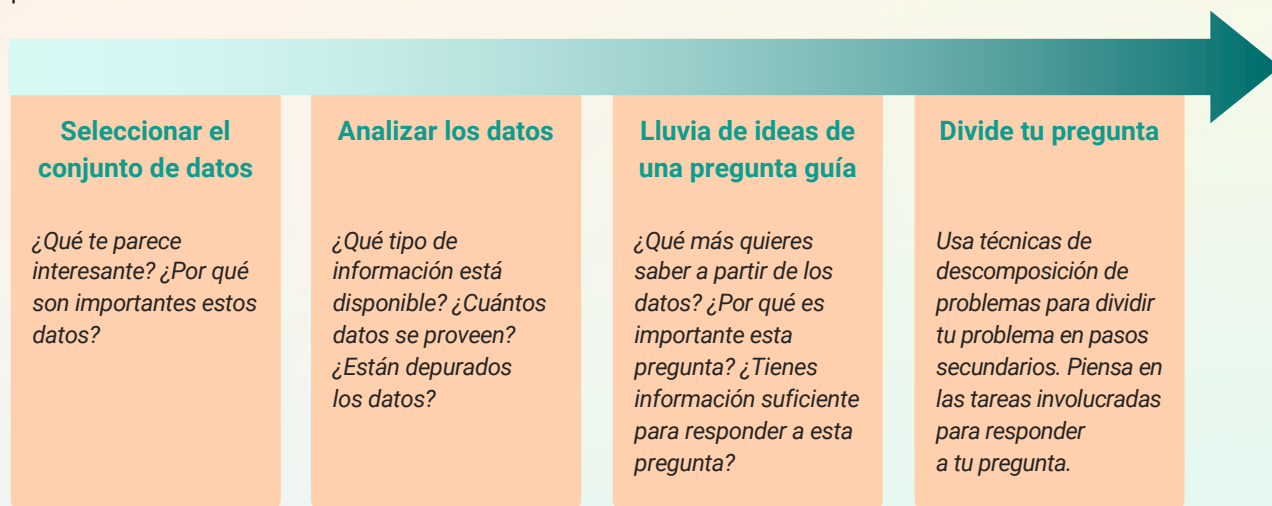
Al trabajar con datos, muchas veces querrás presentar tus resultados con distintos tipos de visualizaciones, como tablas y gráficos. En esta actividad, exploraremos cómo manipular datos, pero mantente atenta a la siguiente actividad, en la que exploraremos técnicas de visualización de datos en Python. Por el momento, toma un descanso y felicítate por ser experta en datos por un día.

Paso 7: Extensiones (5 a 40 minutos)

Extensión 1: Explorar otros conjuntos de datos de Kaggle (30 a 40 minutos)

Hay muchos conjuntos de datos disponibles en Kaggle. Tómate un momento para explorar algunos conjuntos de datos [aquí](#). Tal vez quieras ordenarlos por **facilidad de uso**; recuerda que este puntaje te ayuda a determinar qué tan “depurado” está el conjunto de datos.

Al llevar a cabo tu propio análisis con el conjunto de datos que elijas, querrás seguir un proceso similar al que describimos durante las sesiones de lluvia de ideas en esta actividad.



Una vez que hayas seleccionado el conjunto de datos y realizado una lluvia de ideas sobre las maneras en que puede analizar los datos, es hora de comenzar tu investigación. Algunas investigaciones pueden tardar horas, días o incluso meses. No te desalientes. En esta actividad solo hemos tocado la superficie de algunas de las cosas que **pandas** puede ayudarte a hacer en el análisis de datos, pero **pandas** es capaz de hacer mucho más. Consulta estos recursos para obtener más información sobre esta poderosa biblioteca.

- [Tutorial sobre Pandas de DataCamp: DataFrames en Python](#)
- [Tutorial sobre Pandas en Python de LearnDataSci](#)
- [Tutorial sobre Pandas en Python de Tutorials Point](#)
- [Hoja de referencia para Pandas de DataCamp](#)

Extensión 2: Determinar el porcentaje de fracaso (10 a 15 minutos)

Ahora que has determinado el porcentaje de éxito, ¿qué pasaría si quisiéramos conocer el porcentaje de fracaso? Podemos hacerlo de varias maneras.

- Si vemos el fracaso como lo *opuesto* al éxito y consideramos que solo hay dos estados finales, podemos restar el porcentaje de éxito al 100 %.

```
percentFail = 1 - percentSuccess
```

Recuerda que originalmente usamos un filtro para excluir los proyectos con estado "undefined" y "live".

Recuerda que el conjunto de datos tiene varios estados: successful (exitoso), failed (fallido), canceled (cancelado), undefined (no definido), live (activo) y suspended (suspendido). Al simplemente restar el porcentaje de éxito al 100 %, esto presupone que tratamos los estados "canceled" (cancelado) y "suspended" (suspendido) de igual manera que "failed" (fallido).

- Digamos que queremos que el porcentaje de fracaso solo considere los proyectos con estado "failed" (fallido). Podemos hacer esto simplemente usando el método `value_counts()` en el conjunto de datos.

```
failProj = compProj[compProj["state"]=="failed"]  
failProjCount = failProj["main_category"].value_counts()  
percentFail = failProjCount/totProjCount * 100
```

Extensión 3: Reemplazar los valores en el conjunto de datos (5 a 10 minutos)

Algunas veces, querrás reemplazar valores en el conjunto de datos. Por ejemplo, tal vez quieras combinar ciertas categorías o asegurarte de que ciertos valores sean interpretados igual, por ejemplo, que "Film & Video" sea lo mismo que "Film and Video". Podemos hacer esto fácilmente con `pandas`, usando el método `replace()`. Consulta este [recurso](#) para aprender más sobre las distintas situaciones en las que podrías usar el método de reemplazo.

Paso 8: ¡Compartir tu proyecto de Girls Who Code en casa! (5 a 10 minutos)

Nos encantaría ver tu trabajo y sabemos que a otros también les gustaría. Comparte tu proyecto final con nosotros. No olvides etiquetar [@girlswhocode](#) [#codefromhome](#), ¡y quizá te destaquemos en nuestra cuenta!

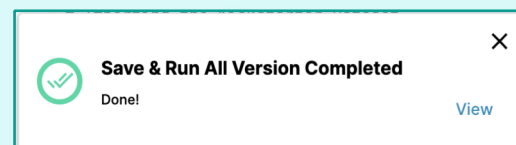
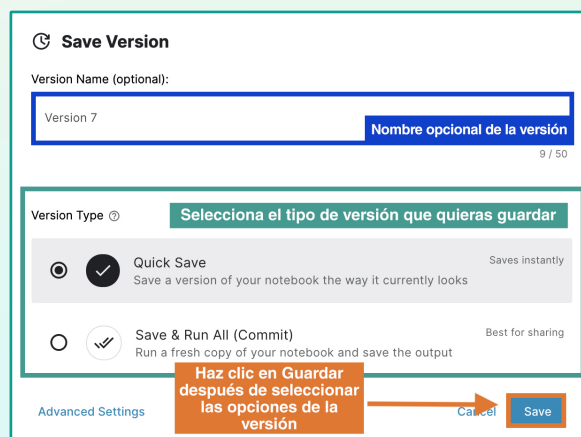
Guardar tu trabajo (2 a 5 minutos)

- **Haz clic en el botón Save Version (Guardar versión).** En la esquina superior derecha de la libreta, tal vez hayas notado el botón **Save Version** (Guardar versión). Esto debe abrir una ventana nueva.
- **Añade un nombre de versión.** Este campo opcional es una buena manera de documentar lo que hiciste en la nueva versión que difiera de las versiones anteriores. Kaggle numerará las versiones automáticamente para que sea fácil ver las versiones viejas.
- **Selecciona el tipo de versión.** Recomendamos seleccionar la opción **Save & Run All** (Guardar y ejecutar todo).

- ◆ **Quick Save (Guardar rápidamente):** Es una excelente manera de guardar tu trabajo de manera muy rápida. Esta versión guardará todo tal como se muestra en la libreta. Esta versión de guardado podría ser problemática si editas la libreta pero no vuelves a ejecutar todos los bloques de código.

- ◆ **Save & Run All (Guardar y ejecutar todo):** Esto guarda una copia reciente de la libreta, ejecutando todos los bloques de código y luego guardando esta versión. Esta siempre es la mejor manera de guardar la libreta, si tienes tiempo.

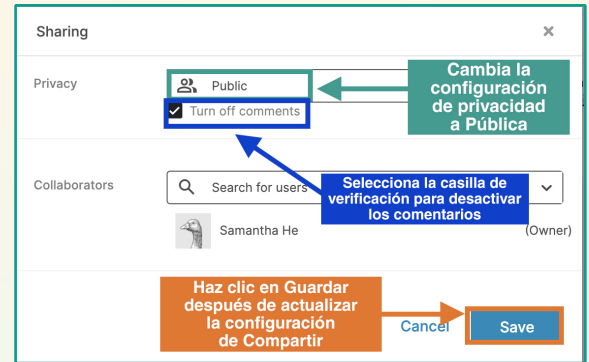
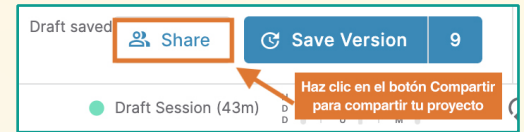
- **Haz clic en el botón Save (Guardar) y espera.** Una vez que hayas confirmado las opciones de la versión que guardarás, haz clic en el botón **Save** (Guardar). Deberás ver una ventana emergente que te indique el estado de la operación de guardado. Tal vez tengas que esperar un minuto a que la versión se guarde por completo.



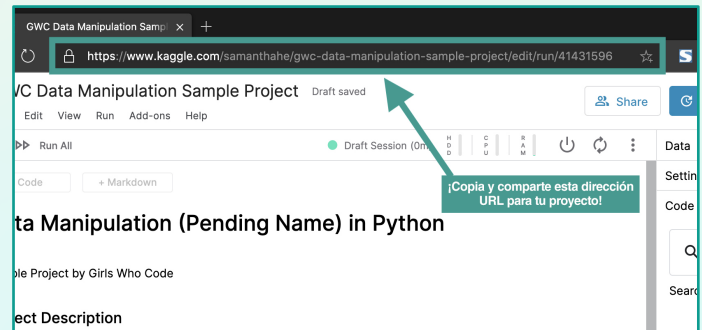
Paso 8: Compartir tu proyecto de Girls Who Code en casa (cont.)

Compartir tu trabajo (3 a 5 minutos)

- **Haz clic en el botón Share (Compartir).** En la parte superior derecha de la libreta, junto al botón Save Version (Guardar versión), está el botón Share (Compartir).
- **Cambia la privacidad a pública.** Selecciona la columna desplegable a la derecha del campo Privacy (Privacidad) y selecciona **Public** (Pública). Aparecerá un mensaje de advertencia para comprobar que sepas que otras personas podrán ver tu proyecto. Selecciona **Ok, make public** (Aceptar, hacer público).
- **Selecciona la casilla "Turn off comments" (Desactivar comentarios).** Haz clic en la casilla de selección a la izquierda de la opción "Turn off comments" (Desactivar comentarios).
- **Haz clic en Save (Guardar).** Una vez que hayas confirmado que toda la configuración esté correcta, haz clic en el botón **Save** (Guardar).
- **Comparte la dirección URL de tu libreta.** Por último, solo tienes que copiar y pegar la dirección URL de tu proyecto con nosotras.



Enlace al proyecto



¡Espera más proyectos de Girls Who Code en casa!

