



# Girls Who Code en casa

**Zona de recreo de datos**  
Manipulación de datos en Python

## Descripción de la actividad

La visualización de datos es el proceso de transformar en arte gráfico datos que contienen grandes cantidades de información en forma de números, palabras o símbolos. Estas obras de arte nos relatan una historia de los datos. Las distintas representaciones pueden destacar o realzar información importante para un público. Por ejemplo, al buscar el número de [casos de COVID-19](#), puedes ver esta información de varias maneras: como [mapa coroplético](#), tabla, gráfico de barras y muchas más.

En esta actividad, aprenderás cómo representar datos mediante diversos tipos de gráficos: lineales, de barras, circulares, histogramas y de dispersión utilizando Python. Específicamente, veremos datos sobre proyectos de [Kickstarter](#), una plataforma de financiamiento colectivo (lo que también se conoce como crowdfunding). Hoy en día, los expertos en datos asumen varios roles al trabajar con datos. Primero, deben depurar los datos, lo que significa eliminar los datos incorrectos o buscar datos duplicados; luego, analizan los datos utilizando diversos tipos de algoritmos; y, finalmente, presentan los datos a las partes interesadas (dependiendo de los datos, las partes interesadas podrían ser responsables de políticas, líderes de negocios, investigadores, médicos, el público en general, etc.).

Si quieres conocer más acerca de cómo manipular los datos de Kickstarter, consulta nuestra actividad:

[Zona de recreo de datos.](#)

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!

## Objetivos del aprendizaje

Al finalizar esta actividad, serás capaz de:

- ◆ Utilizar distintos parámetros del método `plot()` en pandas para crear gráficos personalizados.
- ◆ Determinar la mejor representación gráfica según el tipo de datos.
- ◆ Entender cómo navegar por Kaggle y Jupyter Notebook para explorar un conjunto de datos.

## Materiales

- ◆ [Datos de Kaggle sobre Kickstarter](#)
- ◆ [Proyecto de muestra de representación de datos](#)
- ◆ **Opcional:** Si finalizaste nuestra actividad de [Zona de recreo de datos](#), consulta el [Proyecto de muestra de zona de recreo de datos](#) para ver cómo podrías representar tus resultados finales.

## 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: Fernanda Viégas



Fuente de la imagen: [Google](#)

¿Cómo transformas datos o información en arte? Cuando has querido presentar información, tal vez hayas creado diagramas de Venn, gráficos de barras, tablas o incluso utilizando imágenes para representar los datos. En el trabajo de Fernanda Viégas, ella experimenta con el uso del aprendizaje automático para transformar los datos en hermosas visualizaciones.

Fernanda comenzó su trayectoria como diseñadora gráfica tradicional. Al conocer el Laboratorio de Medios de MIT, encontró inspiración en los artistas que conoció, que combinaban arte y tecnología. Comenzó a aprender a programar y luego fundó

[Flowing Media](#), una empresa naciente que se especializa en el uso de la visualización de datos para expresar ideas y contar historias. Fue a través de esta lente que [Google](#) comenzó a reconocer su trabajo y le ofreció empleo para dirigir el equipo [PAIR](#) (People and Artificial Intelligence Research, Investigación de Personas e Inteligencia Artificial).

Cada día, en el mundo se generan unos 2.5 trillones de bytes de datos y solo un 30 a 40 % es accesible para investigación. ¡Son muchos datos! Fernanda usa [aprendizaje automático](#) e [inteligencia artificial \(AI\)](#) para ordenar los datos, aprender de ellos y entenderlos mejor. Después, aplica un marco de arte y diseño para transformar los datos en formas que nos permiten visualizarlos y comprenderlos con mayor facilidad. Parte de su trabajo en Google consiste en desarrollar la biblioteca [TensorFlow](#) en JavaScript, para facilitarles a otros el uso del aprendizaje automático para la visualización de datos por sí mismos sin que tengan que escribir código desde cero.

Ve este [video](#) para conocer más sobre el impacto que el trabajo de Fernanda tiene en la manera en que vemos los datos.

¿Quieres saber más sobre Fernanda? Visita el [sitio web de Fernanda](#), donde destaca ejemplos de su asombroso trabajo. También puedes ver este [video](#), donde habla un poco acerca de sus antecedentes y de su trabajo en Google.

### Reflexión

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



**CREATIVIDAD**

Fernanda combina sus dos pasiones, el diseño gráfico y las ciencias de la computación, para forjar su propia trayectoria profesional de visualización de datos. ¿Cuáles son dos o más cosas que te interesan y que parezcan no tener nada en común? ¿Podrías pensar en actividades que conecten tus intereses?

Comparte tus respuestas con un familiar o amigo. Anima a otras personas para que lean sobre Fernanda 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.

Si ya has finalizado nuestra actividad [Zona de recreo de datos](#), puedes omitir los pasos 1 a 3.

### 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. Los macrodatos pueden ocupar unos pocos terabytes (TB) de espacio, mientras que los datos normales por lo general solo ocupan decenas o centenas de gigabytes (GB). ¡Es casi 100 veces más que los datos normales! Los macrodatos por lo general requieren mucho esfuerzo para “depurarlos” o dejarlos listos para uso e interpretación. Por ejemplo, piensa en cómo podrías enviarle a una amiga un mensaje de texto con una frase sencilla, como “OK”. Tal vez escribas “okay”, “ok”, “k”, “kay”, y mucho 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.

- ◆ [Statistical & Cognitive Biases in Data Science: What is Bias?](#) de Elder Research
- ◆ [Machine Learning and Human Bias Video](#) de Google.
- ◆ [Dealing with Bias in Artificial Intelligence](#) de The New York Times



## Paso 1: ¿Qué son los macrodatos? (cont.)

### 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 representar los datos, debemos examinar el conjunto de datos que seleccionemos para esta actividad. Es importante entender qué datos se representan y ver si existe algún sesgo.

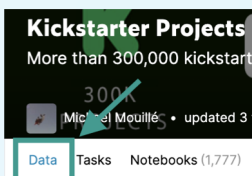
### Proyectos de Kickstarter (1 minuto)



**Kickstarter** es una plataforma de financiamiento colectivo donde una persona o una empresa pequeña pueden financiar proyectos creativos mediante inversiones comunitarias. 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?* Estos datos de Kickstarter contienen valores [categóricos](#), es decir, que pueden agruparse en categorías, y numéricos. Lo elegimos porque presenta una buena perspectiva de distintos tipos de datos. En esta actividad, te guiaremos por un ejemplo básico de *algunas* de las cosas que pueden hacer para visualizar ambos tipos de valores.

### Configurar la vista (5 a 8 minutos)



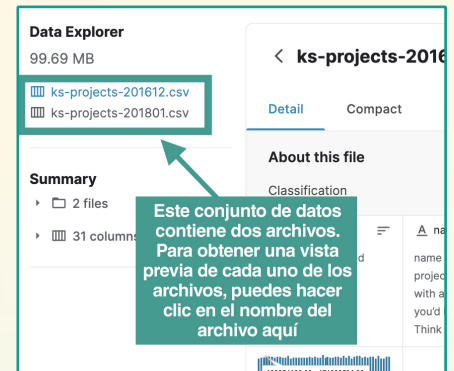
¡Sumerjámonos en los datos! Para comenzar, abramos el [conjunto de datos de Kickstarter en Kaggle](#). Asegúrate de estar en la pestaña **Data** (Datos). Deberás notar que la palabra “Data” en la barra de encabezado superior está en color [azul](#) y [subrayada](#). Luego, avanza hasta la sección **Data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

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

En el lado izquierdo, notarás que hay dos archivos, `ks-projects-201612.csv` y `ks-projects-201801.csv`.

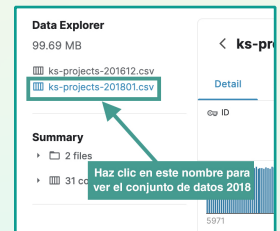
- ◆ `ks-projects-201612.csv`: Este conjunto de datos contiene todos proyectos de Kickstarter lanzados antes de diciembre de 2016.
- ◆ `ks-projects-201801.csv`: Este conjunto de datos contiene todos proyectos de Kickstarter lanzados antes de enero de 2018.

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.

- ◆ **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.

### Características de proyectos de Kickstarter (1 minuto)

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

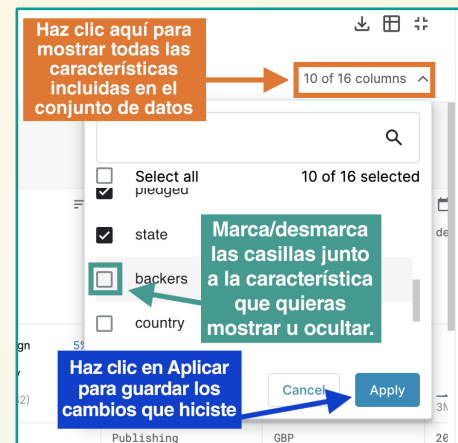
Nombre	Color de ojos	Color de cabello	Fecha de nacimiento
Reshma Saujani	Castaño	Castaño	18 de noviembre de 1975

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

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

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

Los expertos en datos usan diversas herramientas y lenguajes para procesar y analizar datos, como [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 [métodos](#) 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.

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

### Crear una libreta nueva (5 a 8 minutos)

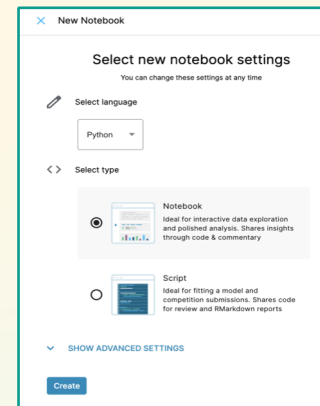
Regresemos a nuestro [conjunto de datos de Kickstarter](#).

- **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 **minimizarlo**, haz clic en el **ícono 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. Asegúrate de que el nombre indique de cierta manera lo que lograrás con el proyecto, como “Representación de exploración de datos”. Tal vez quieras incluir tu nombre en el título; asegúrate de incluir solo tus iniciales o tu nombre de pila y la inicial del apellido.

### 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"><li>◆ <code>import</code>: Esta palabra clave le indica a la computadora que usaremos una biblioteca de Python.</li><li>◆ <code>as</code>: Esta palabra clave le asigna un sobrenombre al paquete. Es un paso opcional, pero puede facilitar la programación.</li><li>◆ <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>.</li><li>◆ <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>.</li></ul>

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`.



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

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.

#### 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
<pre>pd.read_csv("filename")</pre>	<ul style="list-style-type: none"><li>◆ <code>pd</code>: 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.</li><li>◆ <code>.</code>: Este símbolo le indica a la computadora que estamos usando un método.</li><li>◆ <code>read_csv()</code>: 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.</li><li>◆ <code>"filename"</code> (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.</li></ul>

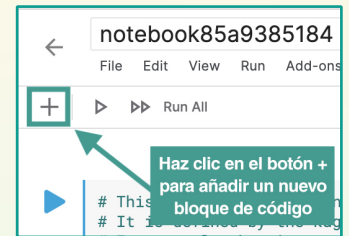


### Paso 3: Introducción a Python y Pandas (cont.)

- **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: ZGAC Arts Capsule For ET	Narrative Film	Film & Video	USD
2	1000004038 Where is Hank?	Narrative Film	Film & Video	USD
3	1000007540 ToshiCapital Rekordz Needs Help to Complete Album	Music	Music	USD
4	1000011046 Community Film Project: The Art of Neighborhood...	Film & Video	Film & Video	USD
...	...	...	...	...
378656	999976400 ChienTruk Nationwide Charity Drive 2014 (Canceled)	Documentary	Film & Video	USD
378657	999977640 The Tribe	Narrative Film	Film & Video	USD
378658	999986353 Walls of Remedy: New Iranian Romantic Comedy I...	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"><li>◆ <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.</li><li>◆ <code>info()</code>: Este método en pandas obtiene información sobre el conjunto de datos, incluidas las características, filas y tipo de datos.</li></ul>

- **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            374864 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 4: Entender datos en Pandas (15 a 20 minutos)

Antes de que comencemos a aprender cómo programar gráficos visuales, tomemos unos momentos para explorar los diferentes tipos de datos y cómo la biblioteca pandas procesa un conjunto de datos.



### Datos categóricos y numéricos (2 a 4 minutos)

Los datos pueden clasificarse en dos tipos: **categóricos** o **numéricos**. Los **datos categóricos** son aquellos que pueden dividirse en categorías o grupos, mientras que los **datos numéricos** son aquellos que pueden expresarse en forma de número. Los datos numéricos deben contener datos que representen un valor continuo (en el que no se omitan números) dentro de cierto rango. En ocasiones, los números pueden representar categorías; por ejemplo, una encuesta en la que te piden que califiques tu experiencia con un número del 1 al 5. Aquí podemos ver que la experiencia está *agrupada* por los números del 1 al 5. Tómate un momento e intenta clasificar los siguientes ejemplos como categóricos o numéricos.

- ◆ Longitud de gatos en centímetros
- ◆ Calificaciones finales de una clase (A, B, C, D o F)
- ◆ Representación de los meses del año mediante números (1 a 12)
- ◆ Cantidad de sueño en horas
- ◆ Colores favoritos de las personas

Haz una pausa aquí antes de consultar las soluciones más abajo.

- ◆ Longitud de gatos en centímetros → **Numérico**
- ◆ Calificaciones finales de una clase (A, B, C, D o F) → **Categórico**
- ◆ Representación de los meses del año mediante números (1 a 12) → **Categórico**
- ◆ Cantidad de sueño en horas → **Numérico**
- ◆ Colores favoritos de las personas → **Categórico**

### DataFrame de Pandas (5 a 8 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 `[]`. Si estás familiarizada con las **matrices** en JavaScript y las **listas** en Python, es un concepto muy similar.

The diagram illustrates a DataFrame structure with rows and columns. The columns are labeled: Name, Team, Number, Position, and Age. The rows are indexed from 0 to 6. The data is as follows:

	Name	Team	Number	Position	Age
0	Avery Bradley	Boston Celtics	0.0	PG	25.0
1	John Holland	Boston Celtics	30.0	SG	27.0
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN
6	Evan Turner	Boston Celtics	11.0	SG	27.0

Arrows indicate that the rows are labeled 'Filas' and the columns are labeled 'Columnas'. A pink box highlights the data area, labeled 'Datos'.

Fuente de la imagen: [GeeksforGeeks](https://www.geeksforgeeks.org/)

Con un total de **15 características** o columnas, tal vez queramos centrarnos en solo algunas de las características del conjunto de datos. Practiquemos cómo podemos usar los símbolos `[]` para seleccionar características específicas.

## Paso 4: Entender datos en Pandas (cont.)

- **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.
- **Usa los símbolos `[]` en nuestra variable de conjunto de datos, `ds`, para seleccionar la característica `'state'` (estado).** 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 las características deseadas *dentro* de los corchetes.

PYTHON	DESCRIPCIÓN
<code>ds["state"]</code>	<ul style="list-style-type: none"><li>◆ <code>ds</code>: la variable que almacena el conjunto de datos de Kickstarter.</li><li>◆ <code>[]</code>: este símbolo se usa para acceder a información del conjunto de datos.</li><li>◆ <code>"state"</code>: Indicamos qué característica queremos seleccionar <i>dentro</i> de los símbolos <code>[]</code>. Las características deben incluirse entre comillas, ya que son nombres.</li></ul>

- **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	CONSEJOS DE DEPURACIÓN
<pre>0      fallido 1      fallido 2      fallido 3      fallido 4      cancelado ... 378656 cancelado 378657 fallido 378658 fallido 378659 fallido 378660 fallido</pre>	<ul style="list-style-type: none"><li>◆ Nombre <code>'ds'</code> 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 <b>ejecutes todos</b> 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.</li><li>◆ ¿Usaste comillas para delimitar los nombres de las características? Verifica que todos los nombres de características estén entre comillas. Asegúrate de <i>teclear</i> todas las líneas de código, ya que las comillas no se copian con el carácter correcto al programar.</li><li>◆ ¿Escribiste correctamente los nombres y las variables? Verifica que todos los nombres de características estén bien escritos. Recuerda que Python distingue entre mayúsculas y minúsculas.</li><li>◆ ¿Hay corchetes <code>[]</code> de más? Tal vez notes que, al teclear un símbolo <code>[</code>, la libreta agrega automáticamente el corchete de cierre <code>]</code>. 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.</li></ul>

## Paso 5: Conocer el método `plot()` (2 minutos)

La biblioteca pandas es una poderosa biblioteca de Python, que te ofrece herramientas para analizar, manipular e incluso visualizar información en un conjunto de datos. Nos permite presentar el método `plot()` de pandas.

PYTHON	DESCRIPCIÓN
<code>ds.plot()</code>	<ul style="list-style-type: none"><li>◆ <code>ds</code>: la variable que almacena el conjunto de datos de Kickstarter.</li><li>◆ <code>plot()</code>: este método en pandas obtiene información del conjunto de datos y lo presenta en forma de gráfico. Usamos el método <code>plot()</code> con nuestra variable, <code>ds</code>, NO con toda la biblioteca pandas. La razón de esto es que queremos crear un gráfico con la información almacenada en nuestra variable de conjunto de datos.</li></ul>

El método `plot()` es muy poderoso: ayuda a tomar los datos almacenados en el conjunto de datos y convertirlos en diferentes gráficos, por ejemplo, de barras, lineales, de caja, de dispersión, histogramas y muchos más. Para especificar el tipo de gráfico e identificar los datos que deben mostrarse, el método `plot()` utiliza varios **parámetros** o entradas. En esta actividad, resaltaremos el uso de algunos de estos parámetros, en particular el atributo `kind` (tipo). El parámetro `kind` (tipo) recibe una entrada de secuencia o palabra que nos permite indicarle a la computadora qué tipo de gráfico queremos mostrar. Algunos de los valores de entrada posibles son 'pie' (circular), 'bar' (barras), 'scatter' (dispersión), etc. Para conocer más sobre las diferentes entradas que recibe el método `plot()`, puedes explorar la [documentación oficial de pandas](#). Usaremos este método en los siguientes pasos para crear gráficos de datos categóricos y numéricos.

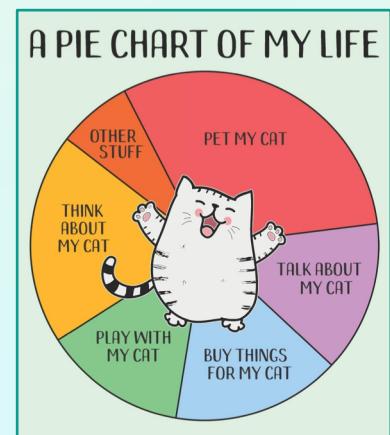
## Paso 6: Mostrar datos categóricos (20 a 30 minutos)

Recuerda que los **datos categóricos** son información que puede agruparse en categorías. En esa sección, revisaremos diferentes tipos de gráficos apropiados para mostrar datos categóricos. Destacaremos las características clave de cada gráfico y luego lo programaremos en Python utilizando el método `plot()`.

### Gráfico circular (2 minutos)

Un **gráfico circular** es un tipo de gráfico en forma de círculo donde los datos ocupan cierto porcentaje del círculo, representados como una rebanada (similar a una rebanada de pizza). Analicemos los componentes de un gráfico circular:

- ◆ **Título:** por lo general aparece en la parte superior del gráfico y nos indica qué se describe.
- ◆ **Rebanada:** cada rebanada representa un porcentaje de datos específicos con respecto a todo el conjunto de datos. En ocasiones, se muestra el porcentaje junto a cada rebanada.
- ◆ **Leyenda:** nos indica qué color representa qué valor de datos. Algunas veces, aparece como una lista a un costado del gráfico o como rótulos individuales junto a cada rebanada.



**Preguntas que debes considerar al seleccionar un gráfico circular:**

- ◆ ¿Tienes datos categóricos?
- ◆ ¿Es importante comparar el número de puntos de datos en una categoría con el conjunto de datos en general?
- ◆ ¿Tienes muchas categorías? Considera el tamaño de cada rebanada. Si hay muchas rebanadas pequeñas, podría ser difícil leerlas en un gráfico circular.

**Fuente de la imagen:** [Amazon](#)



### Elaborar gráficos circulares en Pandas (5 a 8 minutos)

Vuelve a abrir tu libreta en Kaggle. Te guiaremos para usar el método `plot()` en pandas para mostrar un gráfico con el atributo `kind` (tipo). Nuestro conjunto de datos de Kickstarter representa la información de un proyecto en una fila, pero no contiene filas específicas que nos informen acerca de *todos* los proyectos en el conjunto de datos. Por ejemplo, ¿qué pasaría si quisiéramos conocer el número de proyectos clasificados como Film and Video (cine y video)? ¿O la cantidad de proyectos en cada estado? Para obtener esta información, usaremos el método `value_counts()`, que nos permite conocer el detalle de los proyectos en una característica específica. El método `value_counts()` nos ayuda a agrupar valores según ciertas características, como conocer el número de proyectos en cada categoría, para que podamos usar el método `plot()` para mostrar un gráfico representativo de todo el conjunto de datos.

En este ejemplo, te guiaremos para crear un gráfico circular que muestre el detalle de los proyectos según su `estado`. La característica `state` es el estado actual de un proyecto de Kickstarter: exitoso, fallido, suspendido, cancelado, activo o no definido.

- **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.
- **Selecciona la característica `state` del conjunto de datos.** Recuerda que almacenamos la referencia a nuestro conjunto de datos en una variable llamada `ds`. Después, incluiremos la característica `"state"` entre corchetes `[]` para indicar que queremos seleccionar únicamente esta columna del conjunto de datos.
- **Usa el método `value_counts()` en el conjunto de datos filtrado.**

PYTHON	DESCRIPCIÓN
<code>ds["state"].value_counts()</code>	<ul style="list-style-type: none"> <li>◆ <code>ds["state"]</code>: Queremos conocer el detalle de los proyectos por estado (<code>state</code>). 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.</li> <li>◆ <code>.</code>: este símbolo le indica a la computadora que estamos usando un método.</li> <li>◆ <code>value_counts()</code>: Este método devuelve el número de entidades (o filas) que tienen un valor único de la característica.</li> </ul>

- **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"> <li>◆ ¿Necesitas volver a ejecutar todos los bloques de código?</li> <li>◆ ¿Usaste comillas para delimitar los nombres?</li> <li>◆ ¿Escribiste correctamente los nombres y las variables?</li> </ul> <p>Recuerda que Python distingue entre mayúsculas y minúsculas.</p> <ul style="list-style-type: none"> <li>◆ ¿Hay corchetes <code>[]</code> de más?</li> </ul>
fallido	197719												
exitoso	133956												
cancelado	38779												
no definido	3562												
activo	2799												
suspendido	1846												

## Paso 6: Mostrar datos categóricos (cont.)


- **Almacena los datos obtenidos en una variable nueva.** Queremos transformar los datos que obtuvimos de la característica `state` en un gráfico circular. Es mucho más fácil hacerlo si los almacenamos en una variable. Puedes asignarle cualquier nombre a la variable, solo asegúrate de que el nombre de la variable sea descriptivo y que siga la convención camelCase (combinación de minúsculas y mayúsculas). En nuestro ejemplo, le asignamos el nombre `projState` a nuestra variable. Asegúrate de tener esto en cuenta si le asignaste otro nombre a la variable.

```
projState = ds['state'].value_counts()
```

- **Usa el método `plot()` en el conjunto de datos de estado y asigna el parámetro `kind (tipo)` a `'pie'` (circular).** Esta es la línea de código que le indica a la computadora que genere un gráfico circular basado en los datos de la característica `state` que acabamos de almacenar en la variable `projState`.

PYTHON	DESCRIPCIÓN
<pre>projState.plot(kind = 'pie')</pre>	<ul style="list-style-type: none"><li>◆ <code>projState</code>: Específicamente, queremos mostrar el detalle de los estados de proyecto en la forma de un gráfico circular.</li><li>◆ <code>.</code>: este símbolo le indica a la computadora que estamos usando un método.</li><li>◆ <code>plot()</code>: este método en pandas obtiene información del conjunto de datos y lo presenta en forma de gráfico.</li><li>◆ <code>kind = 'pie'</code>: usamos el parámetro (o entrada) <code>kind</code> y le asignamos un valor igual al nombre del gráfico que queremos mostrar. En este caso, mostramos un gráfico circular, por lo que ajustamos el valor a <code>'pie'</code>. El tipo de gráfico debe estar entre comillas, pues es el <i>nombre</i> del gráfico.</li></ul>

- **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:

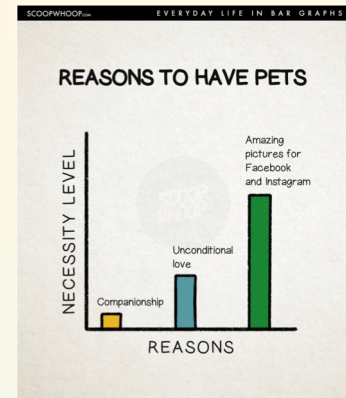
RESULTADOS	CONSEJOS DE DEPURACIÓN
	<ul style="list-style-type: none"><li>◆ ¿Necesitas volver a ejecutar todos los bloques de código?</li><li>◆ ¿Usaste comillas para delimitar los nombres?</li><li>◆ ¿Escribiste correctamente los nombres y las variables? Recuerda que Python distingue entre mayúsculas y minúsculas.</li><li>◆ ¿Colocaste el método entre paréntesis <code>()</code>?</li></ul>

## Paso 6: Mostrar datos categóricos (cont.)

### Gráfico de barras (2 minutos)

Un **gráfico de barras** es un gráfico que muestra los datos mediante el uso de barras de distintas alturas. Analicemos algunos de los componentes de un gráfico de barras:

- **Título**
- **Barras:** cada barra representa una categoría en los datos. La altura del gráfico de barras depende de lo que se mide en el otro eje. Debe haber una pequeña separación entre las barras.
- **Orientación:** en el ejemplo de la derecha, vemos el uso de barras verticales. Los gráficos de barras también pueden contener barras horizontales.



Fuente de la imagen: [ScoopWoop](#)

**Preguntas que debes considerar al seleccionar un gráfico circular:**

- ◆ ¿Tienes datos categóricos?
- ◆ ¿Es importante mostrar la cantidad exacta en cada categoría? Los gráficos de barras muestran los **valores reales** de cada categoría, mientras que los gráficos circulares muestran el *porcentaje* de cada categoría con respecto a todo el conjunto de datos.

### Crear gráficos de barras en Pandas (5 a 8 minutos)

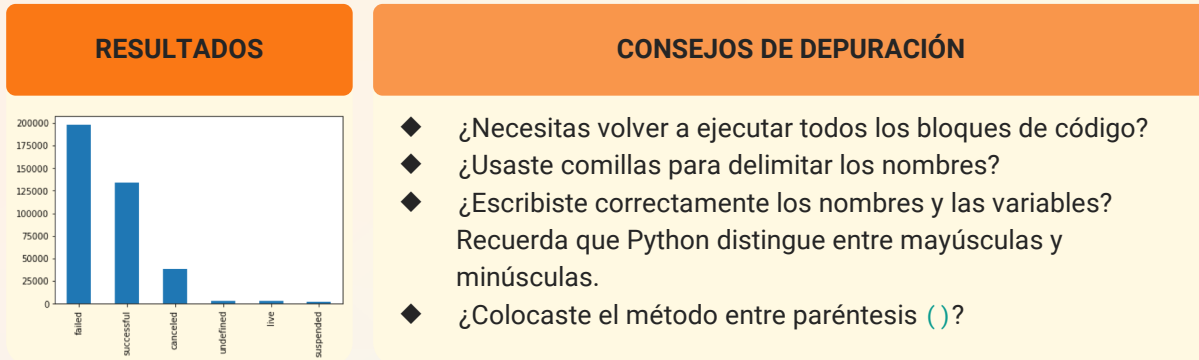
En este ejemplo, presentaremos el estado de los proyectos de Kickstarter en forma de gráfico de barras. De esta manera, podrás comparar fácilmente los gráficos de barras y circulares. Esta vez, al usar el método `plot()`, le asignaremos al parámetro `kind` el valor `"bar"`.

- **Agrega un nuevo bloque de código al final de la libreta.**
- **Usa el método `plot()` en el conjunto de datos de estado y asigna el parámetro `kind` (tipo) a `'bar'` (barra).** Recuerda que almacenamos en la variable `projState` el conjunto de datos que contiene el número de proyectos en cada estado. Tal vez le hayas asignado otro nombre a esta variable.

PYTHON	DESCRIPCIÓN
<pre>projState.plot(kind = 'bar')</pre>	<ul style="list-style-type: none"><li>◆ <code>projState</code>: específicamente, queremos mostrar el detalle del estado de los proyectos.</li><li>◆ <code>.</code>: este símbolo le indica a la computadora que estamos usando un método.</li><li>◆ <code>plot()</code>: este método en pandas obtiene información del conjunto de datos y lo presenta en forma de gráfico.</li><li>◆ <code>kind = 'bar'</code>: usamos el parámetro (o entrada) <code>kind</code> y le asignamos un valor igual al nombre del gráfico que queremos mostrar. En este caso, mostramos un gráfico de barras, por lo que ajustamos el valor a <code>'bar'</code>. El tipo de gráfico debe estar entre comillas, pues es el <i>nombre</i> del gráfico.</li></ul>

## Paso 6: Mostrar datos categóricos (cont.)

- **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:

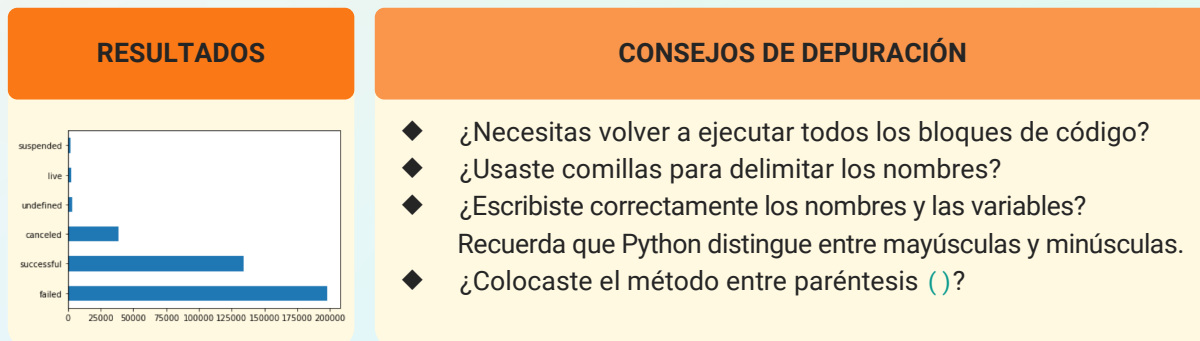


¿Quieres cambiar este gráfico a uno de barras horizontales? Es muy sencillo: esta vez, al usar el método `plot()`, le asignaremos al parámetro `kind` el valor `"barh"`.

- **Agrega un nuevo bloque de código al final de la libreta.**
- **Usa el método `plot()` en el conjunto de datos de estado y asigna el parámetro `kind` (tipo) a `'barh'` (barra horizontal).**

```
projState.plot(kind = 'barh')
```

- **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:



## Comparar gráficos circulares y de barras (5 a 8 minutos)

Tómate un momento para revisar los gráficos de barras y circular que creaste y realiza las siguientes preguntas:

### PREGUNTA 1

A partir de los gráficos,  
¿a qué conclusión puedes  
llegar acerca de los datos?

### PREGUNTA 2

¿Qué gráfico destaca más  
tus conclusiones?

## Paso 6: Mostrar datos categóricos (cont.)

Tómate **2 minutos** para escribir todas las conclusiones que puedas a partir de los gráficos que creaste. Tal vez te resulte útil organizar tus ideas en la forma de una tabla, como la siguiente.

<b>CONCLUSIÓN</b> <i>A partir de los gráficos, ¿a qué conclusión puedes llegar acerca de los datos?</i>	<b>MEJOR GRÁFICO</b> <i>¿Qué gráfico destaca más tu conclusión?</i>

Haz una pausa aquí antes de revelar abajo algunas de las conclusiones a las que llegamos.

Pueden hacerse muchas conclusiones acerca de los datos y puedes revisar algunas de las que creamos. Tus respuestas sobre qué gráfico expresa mejor la conclusión podrían ser ligeramente distintas, y eso está bien. Lo más importante es que puedas justificar tu decisión.

<b>CONCLUSIÓN</b> <i>A partir de los gráficos, ¿a qué conclusión puedes llegar acerca de los datos?</i>	<b>MEJOR GRÁFICO</b> <i>¿Qué gráfico destaca más tu conclusión?</i>
La mayoría de los proyectos tienen estado "failed" (fallido).	Gráfico circular
Casi 200,000 de los proyectos de Kickstarter concluyeron con estado "failed" (fallido).	Gráfico de barras
Cerca del 50 % de los proyectos de Kickstarter fracasan.	Gráfico circular
El número de proyectos no definidos, activos y suspendidos es casi igual.	Gráfico de barras

Recuerda que los gráficos circulares son muy buenos a la hora de ver el detalle de todos los proyectos y cómo se comparan en forma porcentual. Los gráficos de barras son muy buenos para ver el detalle de los proyectos y sus valores exactos. La selección de un gráfico de barras horizontal o vertical depende de tus preferencias. Ambos muestran los mismos datos, pero con orientación diferente.



## Paso 7: Mostrar datos numéricos (25 a 35 minutos)

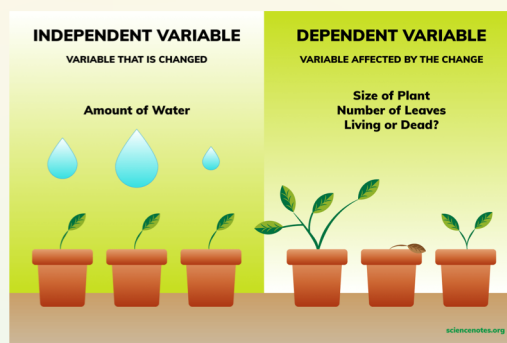


Recuerda que los **datos numéricos** son aquellos que pueden expresarse en forma de número. Deben tener un rango continuo (lo que significa que todos los números en el rango están representados). Al considerar datos numéricos gráficos, debemos identificar las variables **independientes** y **dependientes**.

### Variables independientes y dependientes (2 minutos)

Un gráfico muestra la comparación de dos variables. Por ejemplo, el gráfico circular y el gráfico de barras que creamos compararon el número y el estado de proyectos de Kickstarter. Llamamos a las dos variables que se comparan variable independiente y variable dependiente. La **variable independiente** consiste en datos cuyo cambio es controlado o predecible. En nuestro ejemplo anterior, la variable o los datos que se mantienen constantes son los diversos tipos de estados (es decir, activo, fallido, suspendido, etc.). La **variable dependiente** es el valor que cambia *según* el valor de la variable independiente. En nuestro ejemplo, sería el número de proyectos, ya que depende del estado del proyecto.

Al comparar valores numéricos, la variable independiente se muestra en el **eje x** u horizontal, mientras que la variable dependiente se muestra en el **eje y** o vertical. Debemos especificar estos campos al representar valores numéricos gráficamente en pandas.



Fuente de la imagen: [Science Notes](https://www.sciencenotes.org)

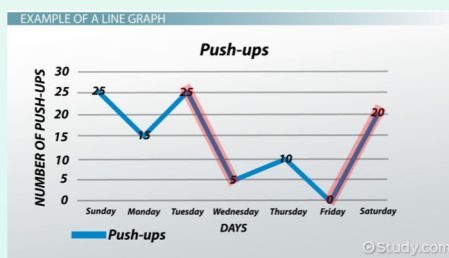
### Gráficos lineales (2 minutos)

Un **gráfico lineal o de líneas** es un gráfico que muestra datos que cambian con el tiempo. Un gráfico lineal utiliza puntos para representar puntos de datos específicos conectados por una línea para mostrar la tendencia con el paso del tiempo. Un gráfico lineal puede usarse para representar datos categóricos y numéricos. Un gráfico lineal tiene ciertos componentes clave:

- ◆ **Título**
- ◆ **Eje X:** en este eje horizontal incluimos la **variable independiente** de los datos.
- ◆ **Eje Y:** en este eje vertical incluimos la **variable dependiente** de los datos.
- ◆ **Tendencia:** En un gráfico lineal, todos los puntos de datos individuales están conectados mediante una línea. Esto ayuda a destacar la tendencia de los datos y ver fácilmente los incrementos y reducciones. En el gráfico de arriba vemos que la línea se realza en color rojo en los días cuando se presenta una tendencia descendente. Algunas veces verás gráficos lineales que distinguen las tendencias mediante colores, o todo el gráfico muestra el mismo color.

### Preguntas a considerar al seleccionar un gráfico lineal:

- ◆ ¿Tienes una variable independiente que mide tiempo? (Por ejemplo, días en una semana, horas en un día, meses un año, minutos en una hora)
- ◆ ¿Es importante ver la tendencia entre los puntos de datos?



Fuente de la imagen: [Study.com](https://www.study.com)

## Crear gráficos de barras en Pandas (5 a 8 minutos)

Tómate un momento para pensar en qué característica podría usarse para representar el *tiempo* en nuestro conjunto de datos de Kickstarter. Tanto la característica `launched` (lanzamiento) y `deadline` (fecha límite) se miden como tiempo. Para este ejemplo, compararemos el monto de la *meta* de los proyectos de Kickstarter a la fecha de `lanzamiento`. Recuerda que debemos considerar qué característica es la variable independiente y cuál es la variable dependiente. Piensa por tu cuenta antes de revelar nuestra respuesta abajo.

- ◆ **Gráfico:** Comparación de la fecha de lanzamiento y el monto de la meta para proyectos de Kickstarter
- ◆ **Variable independiente:** `deadline` (fecha límite). Aquí, el tiempo se mantiene constante.
- ◆ **Variable dependiente:** `goal` (meta). El monto que un proyecto de Kickstarter establece como meta puede variar según el tiempo, por lo cual es la variable dependiente.

En esta ocasión, al usar el método `plot()`, tenemos que establecer los parámetros `x` y `y` como las características que queremos mostrar, y el parámetro `kind` (tipo) como `"line"` (línea).

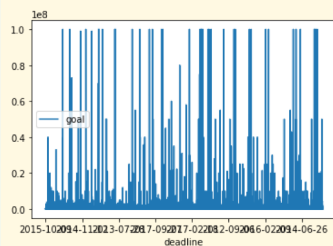
Por defecto, el método `plot()` genera un gráfico lineal, por lo que es opcional especificar el parámetro `kind` (tipo) como `"line"` (línea).

- **Agrega un nuevo bloque de código al final de la libreta.**
- **Usa el método `plot()` en el conjunto de datos y establece los siguientes parámetros:**
  - ◆ parámetro `x` igual a la variable independiente.
  - ◆ parámetro `y` igual a la variable dependiente.
  - ◆ parámetro `kind` igual a `'line'` (línea).

```
ds.plot(x='deadline', y='goal', kind = 'line')
```

- **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

- ◆ ¿Necesitas volver a ejecutar todos los bloques de código?
- ◆ ¿Usaste comillas para delimitar los nombres?
- ◆ ¿Escribiste correctamente los nombres y las variables?  
Recuerda que Python distingue entre mayúsculas y minúsculas.
- ◆ ¿Colocaste el método entre paréntesis `()`?

Tal vez notes que este gráfico lineal no nos dice mucho acerca de los datos, ya que hay mucha información capturada en el conjunto de datos. Este gráfico lineal representó los 378 661 como un solo punto en este gráfico. ¡Son muchísimos! Estamos tratando de entender qué es cada gráfico y como programarlo en pandas; no tenemos que incluir todos los proyectos en el gráfico.

## Paso 7: Mostrar datos numéricos (cont.)

Exploremos el uso del método `head()`, que selecciona únicamente las primeras filas (o entidades) del conjunto de datos. Lo usaremos para mostrar los 100 primeros proyectos del conjunto de datos. También practicaremos aquí el encadenamiento de métodos para agrupar los métodos `head()` y `plot()`.

- **Agrega un nuevo bloque de código al final de la libreta.**
- **Usa el método `head()` con el parámetro 100 para obtener solamente los 100 primeros proyectos del conjunto de datos.**

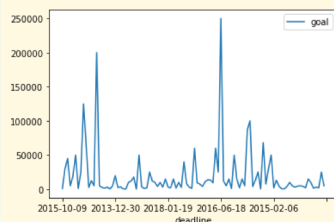
PYTHON	DESCRIPCIÓN
<code>ds.head(100)</code>	<ul style="list-style-type: none"><li>◆ <code>ds</code>: esta es la variable donde almacenamos el conjunto de datos.</li><li>◆ <code>.</code>: este símbolo le indica a la computadora que estamos usando un método.</li><li>◆ <code>head()</code>: este método de pandas obtiene las primeras entidades (o filas) del conjunto de datos.</li><li>◆ <code>100</code>: el método <code>head()</code> recibe un número entero como entrada, que especifica el número de filas que se seleccionarán. En nuestro ejemplo, solo queremos las 100 primeras filas.</li></ul>

- **Encadena el método `plot()` al conjunto de datos y especifica los parámetros `x`, `y` y `kind`:**

```
ds.head(100).plot(x='deadline', y='goal', kind = 'line')
```

- **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

- ◆ ¿Necesitas volver a ejecutar todos los bloques de código?
- ◆ ¿Usaste comillas para delimitar los nombres?
- ◆ ¿Escribiste correctamente los nombres y las variables? Recuerda que Python distingue entre mayúsculas y minúsculas.
- ◆ ¿Colocaste el método entre paréntesis `()`?

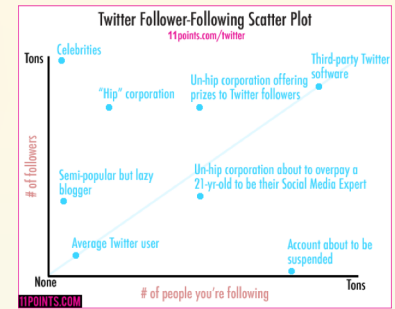
## Paso 7: Mostrar datos numéricos (cont.)

### Gráfico de dispersión (2 minutos)

Un **gráfico de dispersión** es uno que utiliza puntos para representar cada punto de datos al comparar dos variables. Es similar a un gráfico lineal en que cada dato es representado por un punto, pero los puntos no están conectados por una línea. Los gráficos de dispersión por lo general se usan para gráficos que contienen muchos datos, a fin de determinar si existe una tendencia o correlación general. Las tendencias pueden tener forma de línea, de diversas curvas o no existir.

Analicemos los componentes de un gráfico de dispersión:

- **Título**
- **Eje X:** en este eje horizontal incluimos la **variable independiente** de los datos.
- **Eje Y:** en este eje vertical incluimos la **variable dependiente** de los datos.
- **Tendencia:** los puntos o datos individuales no están conectados. En lugar de ello, la tendencia por lo general intenta resumir o agrupar los datos para realzar una relación específica. Si la mejor forma de describir la tendencia es como una línea, es probable que hayas escuchado el término “línea de mejor ajuste” para describirla.



Fuente de la imagen: [11](#).

### Crear gráficos de dispersión en Pandas (3 a 5 minutos)

Uno de los mayores retos que notamos en el gráfico lineal fue que las líneas que conectaban los puntos dificultaban ver una tendencia general de los datos. Intentemos de nuevo comparar el monto de la **goal** de un proyecto de Kickstarter con la fecha de **launched**. En esta ocasión, al usar el método `plot()`, tenemos que establecer los parámetros **x** y **y** como las características que queremos mostrar, y el parámetro **kind** (tipo) como **"scatter"** (dispersión).

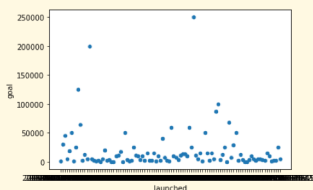
- **Agrega un nuevo bloque de código al final de la libreta.**
- **Usa el método `plot()` en el conjunto de datos y establece los siguientes parámetros:**
  - ◆ **parámetro `x` igual a la variable independiente.**
  - ◆ **parámetro `y` igual a la variable dependiente.**
  - ◆ **parámetro `kind` igual a `'scatter'` (dispersión).**

```
ds.plot(x='deadline', y='goal', kind = 'scatter')
```

Tal vez notes que Kaggle tarda bastante tiempo en ejecutar un gráfico de dispersión para todo el conjunto de datos. Si tarda demasiado, detén la ejecución del bloque de código y usa el método `head()` para filtrar los 100 primeros puntos de datos antes de generar el gráfico de dispersión.

- **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



\*Este gráfico solo muestra los 100 primeros puntos de datos

#### CONSEJOS DE DEPURACIÓN

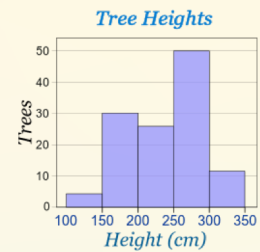
- ◆ ¿Necesitas volver a ejecutar todos los bloques de código?
- ◆ ¿Usaste comillas para delimitar los nombres?
- ◆ ¿Escribiste correctamente los nombres y las variables? Recuerda que Python distingue entre mayúsculas y minúsculas.
- ◆ ¿Colocaste el método entre paréntesis `()`?

Una conclusión a la que podríamos llegar a partir de este gráfico es que la mayoría de los proyectos de Kickstarter corresponden a un valor de la **meta** de menos de 50,000.

## Paso 7: Mostrar datos numéricos (cont.)

### Histograma (2 minutos)

Un **histograma** es un gráfico que muestra los datos mediante barras de distintas alturas. A diferencia de los gráficos de barras, los histogramas agrupan los datos en rangos. Por esta razón, los histogramas solo pueden usarse con **datos numéricos**. Analicemos algunos de los componentes de un histograma:



Fuente de la imagen: [Math is Fun](#)

- ◆ **Título**
- ◆ **Barras:** Cada barra representa un rango de valores en los datos. La altura del gráfico de barras mide la cantidad de puntos de datos dentro del rango. A diferencia de los gráficos de barras, **no hay separación** entre las barras.
- ◆ **Contenedores:** determina el número de barras que deben mostrarse en el gráfico. En pandas el valor predeterminado es de 10 contenedores (o barras).

### Crear histogramas en Pandas (15 a 20 minutos)

En este ejemplo, dividiremos los rangos del monto de la meta en los proyectos de Kickstarter. Para generar un histograma en pandas, podemos usar el método `plot()` y ajustar el parámetro `kind` a `'hist'`, o podemos usar el método `hist()`. Ambos métodos hacen exactamente lo mismo, pero el método `hist()` tiene parámetros específicos que nos permiten personalizar características exclusivas de los histogramas, como el número de **contenedores**.

→ **Agrega un nuevo bloque de código al final de la libreta.**

→ **Usa el método `hist()` con el conjunto de datos y ajusta el parámetro `column` (columna) a la característica `'goal'` (meta).**

PYTHON	DESCRIPCIÓN
<pre>ds.hist(column = 'goal')</pre>	<ul style="list-style-type: none"><li>◆ <code>ds</code>: esta es la variable que almacena el conjunto de datos.</li><li>◆ <code>.</code>: este símbolo le indica a la computadora que estamos usando un método.</li><li>◆ <code>hist()</code>: este método de pandas genera un histograma</li><li>◆ <code>column = 'goal'</code>: usamos el parámetro <code>column</code> (columna) para especificar para qué característica numérica queremos ver el detalle de valores.</li></ul>

→ **Ejecuta el bloque de código.**



#### RESULTADOS

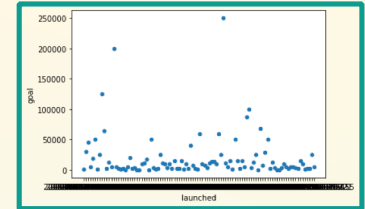
#### CONSEJOS DE DEPURACIÓN

- ◆ ¿Necesitas volver a ejecutar todos los bloques de código?
- ◆ ¿Usaste comillas para delimitar los nombres?
- ◆ ¿Escribiste correctamente los nombres y las variables?  
Recuerda que Python distingue entre mayúsculas y minúsculas.
- ◆ ¿Colocaste el método entre paréntesis `()`?



## Paso 7: Mostrar datos numéricos (cont.)

Tal vez te preguntes por qué solo ves una barra en este histograma. El valor predeterminado para el número de **contenedores** al usar `hist()` es 10. Esto significa que deberíamos ver exactamente 10 barras, ¿pero por qué solo vemos una? Recuerda el gráfico de dispersión, en el que comparamos la meta de los proyectos a lo largo del tiempo. La mayoría de los proyectos tenían un valor de la meta de 50,000 o menos, pero había algunos proyectos con una meta alta de hasta 100,000,000, o baja de hasta 0.01. Cuando pandas intenta dividir este rango en 10 contenedores, cada uno de ellos tiene un rango muy grande. La mayoría de los proyectos corresponderán al primer contenedor que se muestra en el histograma. Hay proyecto en el otro rango, pero son tan pocos que parecen no existir en el histograma. Hagamos un poco de **manipulación de datos** para ver los proyectos con una meta de 5000 o menos. Recuerda que en pandas podemos usar los símbolos `[]` y añadir filtros entre ellos, como lo hicimos cuando quisimos realzar solo ciertas características. En este caso, añadiremos una **sentencia condicional** para obtener valores de `goal` (meta) inferiores a 5000.



- **Agrega un nuevo bloque de código al final de la libreta.**
- **Escribe una condición para la característica `goal` (meta) del conjunto de datos para obtener solo valores inferiores a 5000.**

PYTHON	DESCRIPCIÓN
<code>(ds["goal"] &lt; 5000)</code>	<ul style="list-style-type: none"><li>◆ <code>ds["goal"]</code>: queremos aplicar un filtro al conjunto de datos, de modo que usamos nuestra variable de conjunto de datos, <code>ds</code>, y luego usamos corchetes <code>[]</code> para especificar el filtrado por la característica <code>"goal"</code> (meta).</li><li>◆ <code>&lt; 5000</code>: dado que solo queremos proyectos con un valor de la meta inferior a 5000.</li><li>◆ <code>()</code>: colocamos la sentencia condicional entre paréntesis para que la computadora pueda distinguir la sentencia.</li></ul>

- **Aplica la sentencia condicional que escribiste para el conjunto de datos dentro de los símbolos `[]`.** Aquí, usaremos la variable `ds` y los símbolos `[]` para aplicar los enunciados condicionales del paso anterior. Al añadir el enunciado condicional *dentro* de los corchetes `[]`, le indicamos a la computadora que busque en el conjunto de datos `ds` todas las entidades (o filas) donde el valor de la característica `'goal'` sea menor que 5000.

```
ds[(ds['goal'] < 5000)]
```

- **Almacena este nuevo conjunto de datos filtrado en una nueva variable.** Queremos almacenar este conjunto de datos filtrado en una nueva variable, para que no modifiquemos el conjunto de datos original. En nuestro ejemplo, le asignamos el nombre `under5000` a esta variable.

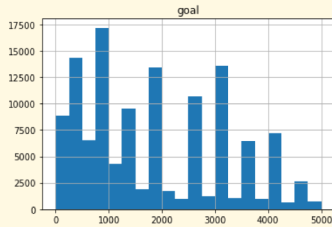
```
under5000 = ds[(ds['goal'] < 5000)]
```

- **Usa el método `hist()` con el conjunto de datos filtrado y ajusta el parámetro `column` (columna) a la característica `'goal'` (meta).**

```
under5000.hist(column='goal')
```

→ Ejecuta el bloque de código.

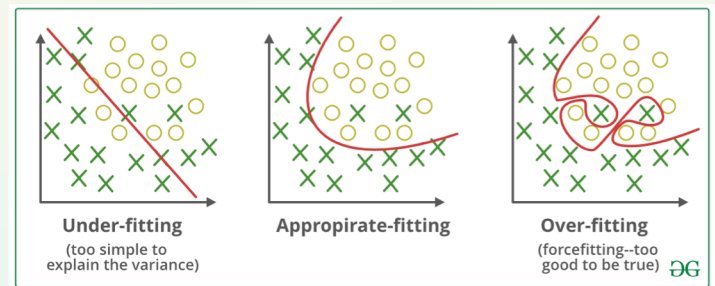
### RESULTADOS



### CONSEJOS DE DEPURACIÓN

- ◆ ¿Necesitas volver a ejecutar todos los bloques de código?
- ◆ ¿Usaste comillas para delimitar los nombres?
- ◆ ¿Escribiste correctamente los nombres y las variables?  
Recuerda que Python distingue entre mayúsculas y minúsculas.
- ◆ ¿Colocaste el método entre paréntesis `()`?
- ◆ ¿Colocaste corchetes `[]` para delimitar el conjunto de datos?
- ◆ ¿Usaste comas para separar los parámetros?

Al incrementar el número de contenedores del gráfico, podemos ver que hay picos de proyectos en diversos montos de meta. Esto no se destacaba tanto cuando usábamos 10 contenedores. Podemos experimentar con parámetros como `bin` (contenedor) para determinar cuáles se ajustan mejor a los datos. Este proceso se conoce como **ajuste**. Hay un equilibrio fino entre un **sobreajuste** o tratar de correlacionar demasiado datos, con lo que incluyen valores atípicos, y un **subajuste**, o no correlacionar datos suficientes, con lo que no se mide de manera correcta la tendencia.



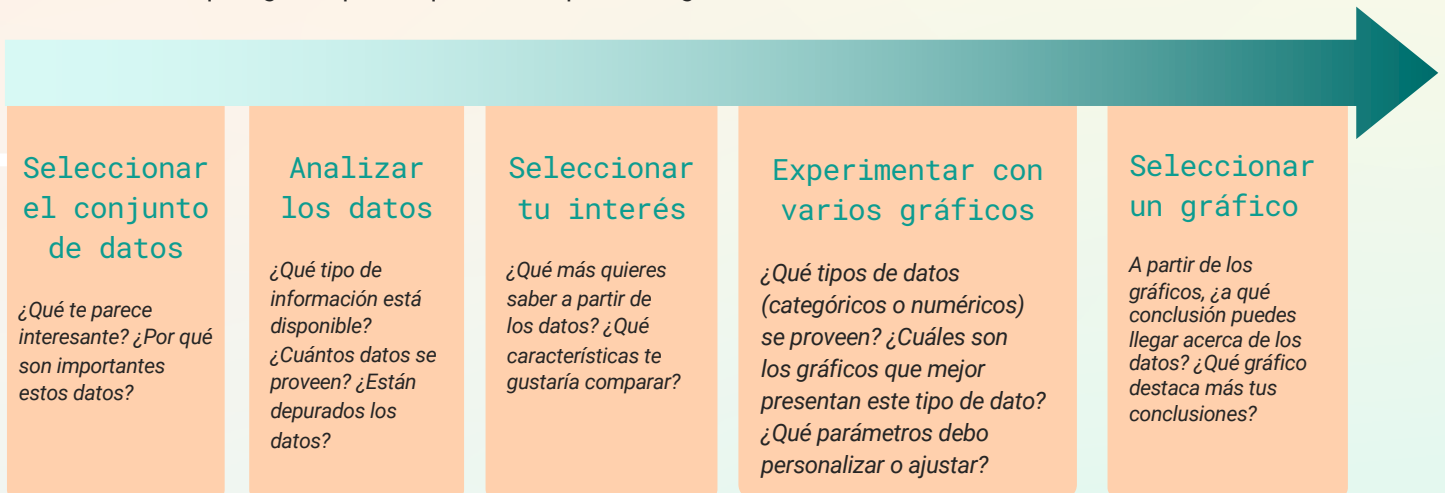
Fuente de la imagen: [GeeksforGeeks](https://www.geeksforgeeks.org/)

## Paso 8: 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 ordenar por **facilidad de uso**. Este puntaje te ayuda determinar qué tan “depurado”, o listo para usarse e interpretarse, está un conjunto de datos.

Al analizar un nuevo conjunto de datos, debes asegurarte de entender profundamente los datos antes de comenzar. He aquí algunos pasos que tal vez quieras seguir.



Una vez que haya seleccionado el conjunto de datos y realizado una lluvia de ideas sobre cómo visualizar los datos, es hora de comenzar el análisis. En esta actividad solo hemos tocado la superficie de algunas de las cosas que [pandas](#) puede hacer para ayudarte a visualizar los datos, pero [pandas](#) es capaz de hacer mucho más. Consulta estos recursos para obtener más información sobre esta poderosa biblioteca.

- ◆ [Documentación de Pandas: Visualización](#)
- ◆ [Real Python: Gráficos con Pandas](#)

Tal vez descubras que tus datos requieran un poco más de actualización o manipulación para llegar más lejos con tu análisis. Te recomendamos que consultes nuestra actividad [Zona de recreo de datos](#) sobre manipulación de datos en Python.

## Extensión 2: Personalizar tus gráficos (5 a 15 minutos)

El método `plot()` contiene varios parámetros. Tomemos un momento para destacar algunos de los parámetros que podría ser interesante incluir.

- ◆ **title:** este parámetro recibe una secuencia, o palabra y la muestra en la parte superior del gráfico como el título. Si tienes varios gráficos, puedes pasar una lista de títulos y pandas los separará de modo correspondiente.
- ◆ **grid:** este parámetro añade líneas de cuadrícula al gráfico. Para añadir líneas de cuadrícula, debes ajustar este parámetro a True (verdadero).
- ◆ **legend:** al ajustar este parámetro a True (verdadero), puedes añadir una leyenda a un lado del gráfico
- ◆ **xlim:** pandas configura automáticamente el gráfico según el mejor rango de valores. Puedes usar este parámetro para restringir el rango de valores en el eje horizontal del gráfico. Debes ingresar el rango en la forma de un par ordenado, usando paréntesis.
- ◆ **ylim:** similar a xlim, el parámetro ylim puede usarse para restringir el rango de valores en el eje vertical del gráfico.
- ◆ **xlabel:** este parámetro recibe una secuencia y agrega una etiqueta al eje horizontal.
- ◆ **ylabel:** este parámetro recibe una secuencia y agrega una etiqueta al eje vertical.
- ◆ **color:** este parámetro recibe una secuencia y cambia el color de los datos en el gráfico. Puedes ver la lista de nombres de colores [aquí](#).

Experimenta con algunos de estos parámetros para personalizar el aspecto del gráfico.

## Extensión 3: Explorar Seaborn (20 a 30 minutos)

Pandas es una herramienta poderosa utilizada por programadores para buscar, filtrar, comparar, modificar y visualizar datos. Sin embargo, tal vez hayas notado que algunos de los gráficos no son muy bonitos.

Permítenos presentarse otra biblioteca de Python: **Seaborn**. Seaborn es una biblioteca creada específicamente para la visualización de datos, lo que significa que, a diferencia de pandas, la mayoría de las funciones y los atributos se especializan en lograr que los gráficos tengan un aspecto más organizado y que los programadores tengan la habilidad de personalización adicional.

Para usar seaborn en tu libreta de Kaggle, primero debemos importar la biblioteca.

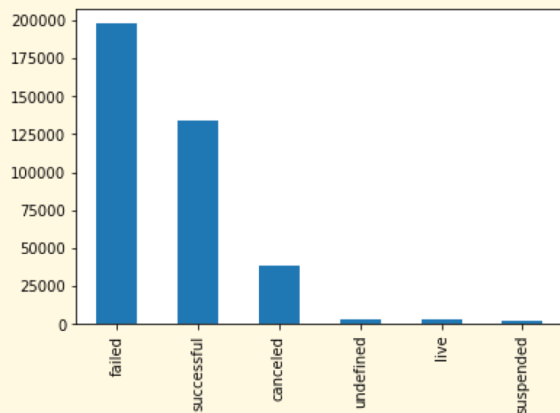
PYTHON	DESCRIPCIÓN
<code>import seaborn as sns</code>	<ul style="list-style-type: none"> <li>◆ <b>import:</b> Esta palabra clave le indica a la computadora que usaremos una biblioteca de Python.</li> <li>◆ <b>as:</b> Esta palabra clave le asigna un sobrenombre al paquete. Es un paso opcional, pero puede facilitar la programación.</li> <li>◆ <b>seaborn/sns:</b> esta biblioteca de Python se usa para visualizar datos. Le hemos asignado a este paquete el sobrenombre de <b>sns</b>.</li> </ul>

## Paso 8: Extensiones (cont.)

Comparemos algunos de los gráficos de barras que creamos para ilustrar el número de proyectos en cada estado:

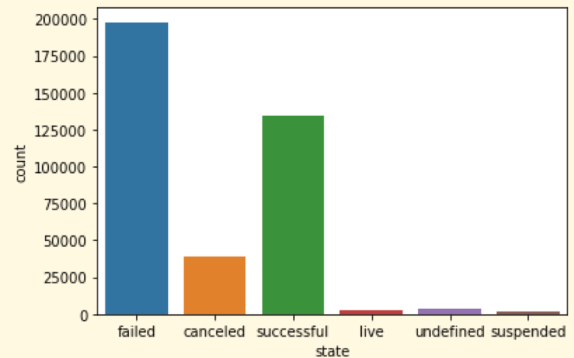
### PANDAS

```
projState = ds['state'].value_counts()  
projState.plot(kind='bar')
```



### SEABORN

```
sns.countplot(x='state', data=ds)
```



Al usar la biblioteca seaborn vemos que cada barra tiene un color diferente, que los ejes x e y ya están etiquetados y que se requieren menos líneas de código. El método `countplot()` recibe un conjunto de datos y calcula la frecuencia de cada valor único en una característica. Al usar pandas, tuvimos que usar la función `value_counts()` para hacer lo mismo.

Tómate un tiempo para explorar la biblioteca [seaborn](#) y algunos de los tipos de gráficos que ofrece. He aquí recursos adicionales, si quieres aprender más:

- [The Ultimate Python Seaborn Tutorial: Gotta Catch 'Em All](#)
- [Data Camp's Python Seaborn Tutorial For Beginner](#)



## Paso 9: ¡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!

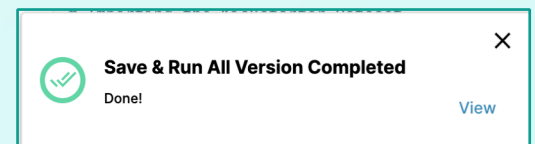
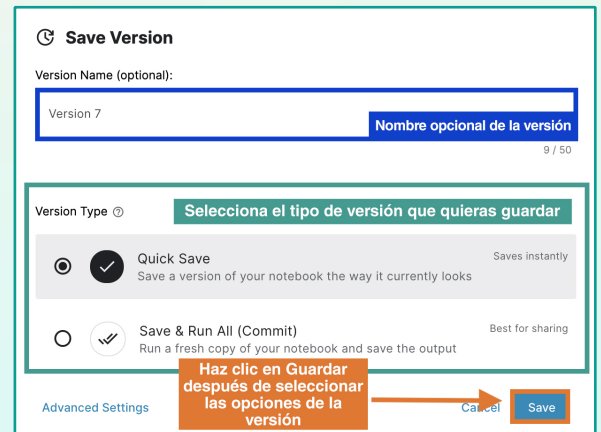
### Guardar tu trabajo (2 a 5 minutos)

- **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.

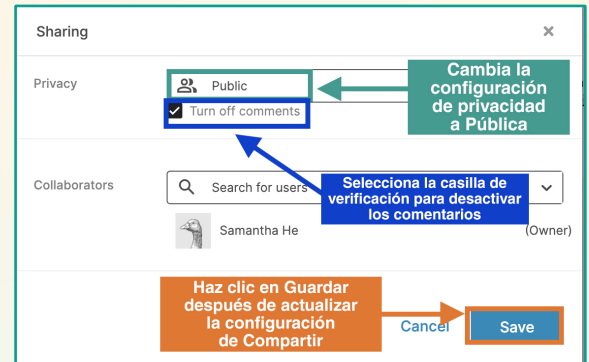
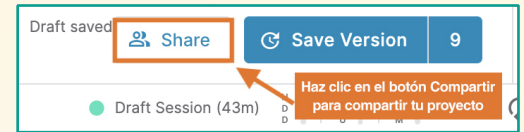
- **Haz clic en el botón Save (Guardar) y espera.** Una vez que hayas confirmado las opciones de la versión que guardarás, haz clic en el botón **Save** (Guardar). Deberás ver una ventana emergente que te indique el estado de la operación de guardado. Tal vez tengas que esperar un minuto a que la versión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



## Paso 9: Compartir tu proyecto de Girls Who Code en casa (cont.)

### Compartir tu trabajo (3 a 5 minutos)

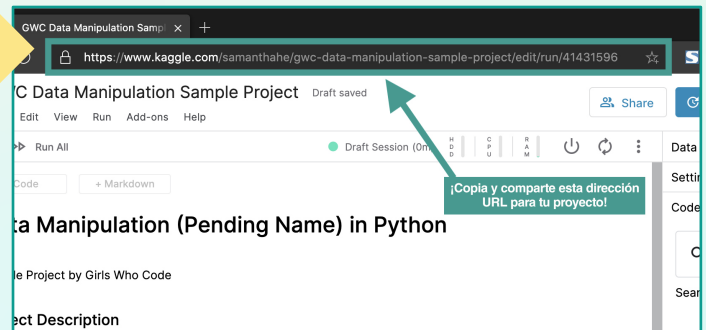
- **Haz clic en el botón Share (Compartir).** En la parte superior derecha de la libreta, junto al botón Save Version (Guardar versión), está el botón Share (Compartir).
- **Cambia la privacidad a pública.** Selecciona la columna desplegable a la derecha del campo Privacy (Privacidad) y selecciona **Public** (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. No olvides etiquetar **@girlswhocode** **#codefromhome**.



#### Nota acerca de colaboradores:

Kaggle te permite invitar a otros usuarios para que programen juntos en tu libreta. Esto es excelente para trabajar en equipo. **¡NO permitas que cualquier persona sea colaboradora!** Selecciona con cuidado con quién compartes los derechos de edición.

#### Enlace al proyecto



¡Espera más proyectos de Girls Who Code en casa!

