**Cheat Sheet - Taller Data Analysis**

Una hoja de atajos o más conocida como “Cheatsheet” es un archivo en el que encontramos un conjunto **conciso/resumido** de notas utilizadas como referencia rápida para apoyar nuestro proceso de aprendizaje.

***Colab***

1. **Colab de Google** nos permite usar “**code cells**” para escribir código, pero también se pueden utilizar “**text cells**”, que son celdas (o filas) para agregar texto a nuestro trabajo e ir documentando.

Para agregar una celda, ya sea de código o texto, basta con presionar el botón +Código o +Texto en la esquina superior izquierda del navegador. Un atajo para agregar una celda de código es Ctrl + M + B.

1. **Cambiar el directorio de trabajo:** En el lado izquierdo del notebook de Colab con el que estás trabajando verás un símbolo de carpeta. Si presionas el botón, se abrirá la carpeta en la que se está trabajando (de tu Drive de Google), que por defecto se llama "sample\_data". Esto se puede cambiar a un nuevo directorio (o carpeta de trabajo) utilizando el código:

**import** os

os.chdir('Dirección\_nueva')

1. Si tu dataset está comprimido, se puede abrir directamente con el siguiente código:

!unzip Dirección\_donde\_quedó\_la\_descarga

1. Si quieres descargar un dataset directamente desde un URL:

!wget Dirección\_url

1. Para descargar archivos desde el Colab:

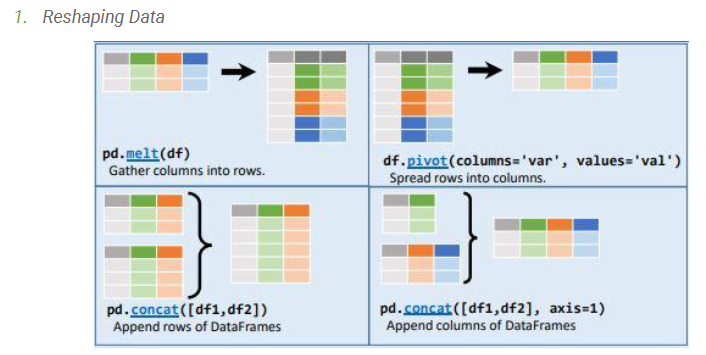
**from** google.colab **import** files

files.download('Dirección\_archivo\_a\_descargar')

***Pandas***

*Manipulación y reestructuración de datasets*

Al trabajar con datasets estructurados o “tidy datasets”, en los que cada variable se almacena como columna y cada observación se almacena en su propia fila, se pueden aplicar una serie de cambios/mejoras/reestructuración según los requerimientos. En estos casos, las operaciones vectorizadas de pandas se aplican de manera eficiente y preservan la estructura de las observaciones. Recuerda que pandas se importa como pd en el ambiente de trabajo.



**pd.melt(df)** Convierte columnas en filas, reuniendo la información.

**pd.pivot(df)**Convierte filas en columnas, “esparciendo” la información.

**pd.concat([df1, df2])** Apila dos datasets (juntando por filas) y los convierte en uno. Para ello, ambos datasets deben tener las mismas columnas.

**pd.concat([df1, df2], axis=1)**Junta dos datasets anexando columnas. Para ello, ambos datasets deben tener las mismas filas.

**df.sort\_values("variable")**Ordena las filas del dataset en base a los valores de la “variable”, por defecto de menor a mayor.

**df.rename(columns={"nombre antiguo": "nombre nuevo"})** Renombra las columnas de un dataset.

**df.sort\_index()** Ordena el índice de un DataFrame.

**df.reset\_index()** Redefine el índice de un DataFrame asignando el número de fila correspondiente, convirtiendo el índice previo en una columna del DataFrame.

**df.drop(columns=["var1", "var2"])** Elimina del DataFrame la lista de columnas entregadas entre paréntesis cuadrado.

**2.** *Subconjuntos de filas*

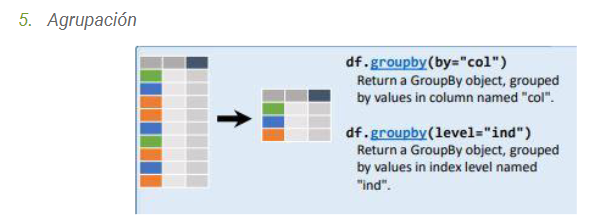
* **df[df.columna2 < 10]** Extrae las filas que cumplen el criterio de la columna definida. En este caso, se extraen las filas en que el valor en la columna 2 es menor a 10.
* **df.drop\_duplicates()** Remueve las filas duplicadas (solo considera columnas).
* **df.sample(frac=0.5)** Selecciona una fracción de datos (en este caso 50%) de manera aleatoria.
* **df.sample(n=10)** Selecciona n filas de manera aleatoria.
* **df.head(n)** Muestra las n primeras filas del DataFrame.
* **df.tail(n)** Muestra las n últimas filas del DataFrame.

**3.** *Subconjuntos de columnas*

* **df[["columna1", "columna2", "columna4"]]** Selecciona múltiples columnas con sus nombres específicos.
* **df["columna2"]** Selecciona una columna específica.
* **df.filter(regex="texto")** Selecciona las columnas en las cuales está presente el “texto” definido.

**4.** *Tablas de resumen (Summarize Data)*

* **df.value\_counts()** Cuenta el número de filas con valores únicos en el DataFrame.
* **df["columna3"].value\_counts()** Cuenta el número de filas con valores únicos de la “columna3”.
* **len(df)** Entrega el número de filas en el DataFrame.
* **df.shape** Entrega una tupla del número de filas y el número de columnas en el DataFrame.
* **df["columna1"].nunique()** Número de valores únicos en la columna 1.
* **df.describe()** Entrega estadísticos descriptivos básicos de cada columna del DataFrame.



* **df.groupby(by="col")** Entrega un objeto agrupado según la columna “col”.
* **df.groupby(level="index")** Entrega un objeto agrupado por los valores del índice del DataFrame.

*Referencias*

Para más detalles y atajos, visitar:

<https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf>

<https://colab.research.google.com/github/Tanu-N-Prabhu/Python/blob/master/Cheat_sheet_for_Google_Colab.ipynb>