Osvaldo Quesada R.

Universidad Castro Carazo

Portafolio de evidencias

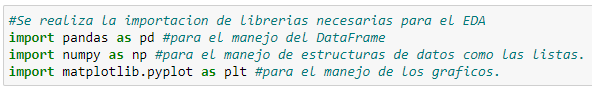
Análisis de Datos

# Semana 7

## Síntesis de los contenidos abordados en la sesión y evidencias

Repaso de contenidos:

**Importación de librerías:**

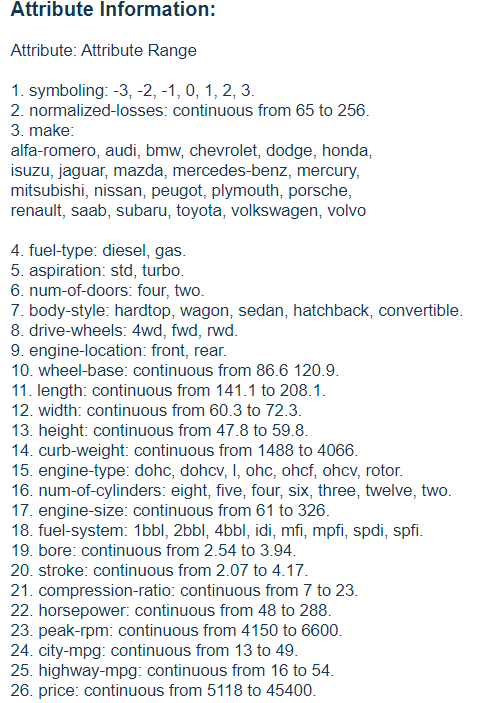


**Carga de Datos:**

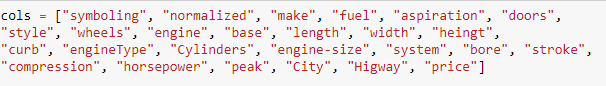
1. Identificar el set de datos (txt, csv,xlsx,BD).

<https://archive.ics.uci.edu/ml/datasets/Automobile>

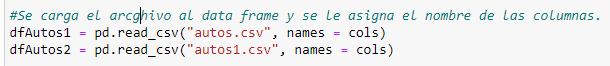
1. Verificar los encabezados.



* Se crea un array unidimensional con los nombres de las columnas:



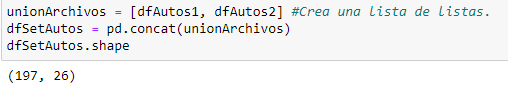
1. Crear estructura en memoria para manejo de los datos (Dataframe)



1. Conocer las estadísticas y datos generales del DataFrame.

* Meta data: descripción de valores que tiene cada columna, interpretación de los datos.
* dfAutos1.info() : muestra el conteo de datos no nulos, tipo de dato, columnas, etc.
* dfAutos1.describe() : muestra las medidas estadísticas.

1. Unión de Archivos de Datos cuando son 2 data set:



.shape: muestra las dimensiones generales de los dos archivos unidos (observaciones y columnas).

print (dfAutos1.head(10)) #Para visualizar todas las columnas si hay ocultas.

**Manejo de datos faltantes o perdidos**:

1. Identificar si hay valores faltantes (NaN, null, ?,””).

* verificar que el tipo de dato corresponda con la columna.
* Identificar como se muestran los valores faltantes y estandarizarlos a NaN.



# el “?” puede ser remplazado por el signo o letra que venga como nulo.

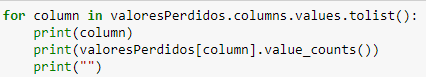
# Se remplaza el “?” por el nan y ese cambio queda aplicado en el set de datos con el inplace.

1. Contar los NaN faltantes por variable / columna.



#Se crea una variable para almacenar el conteo de valores nulos.

#Usando valoresPerdidos.head(5) se puede ver los primeros 5 valores nulos en verdadero.



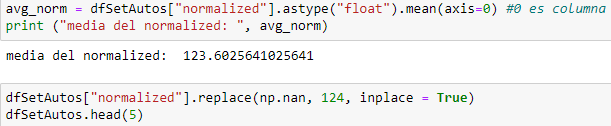
# para cada columna en valoresPerdidos, convierta los valores a una lista.

#imprima la columna

#imprima la cantidad de valores contados en la columna.

1. Imputación de valores:

* Imputación de valores numéricos con la media (promedio).



#busque en la columna normalized, conviértalo en tipo float y calcule la media. Axis 0 = columna. Axis 1 = fila.

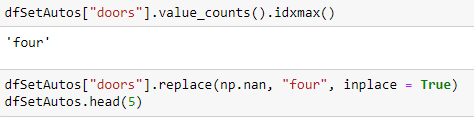
#Luego se sustituye los valores nan con el resultado de la media:

* Imputación valores String con dato más frecuente.

#Se busca el valor más común.

#Se cuenta todo lo de la columna y devuelve el que tuvo mayor conteo.

#Se sustituye los valores nan con el resultado.



* Eliminar observaciones o columnas.

Si más del 90% de los datos en la columna faltan, se puede eliminar la columna.

Si en la columna solo hay pocos (3 o 4) se puede eliminar la fila del faltante si no se pueden recuperar los datos.



#borrar los na en la columna Price y se resetea el índice para que quede consecutivo.

## Desarrollo de actividades y lecturas complementarias

Lectura sobre normalización de datos.

* Escalado absoluto máximo: cambia la escala de cada característica entre -1 y 1 dividiendo cada observación por su valor absoluto máximo.

Se utilizan los métodos de Pandas .max() y .abs().

* Escalado mínimo-máximo: Re escala la característica a un rango fijo de [0,1] restando el valor mínimo de la característica y luego dividiéndolo por el rango. Se usan los métodos. min() y .max().
* El método de puntuación z (Estandarización): Transforma los datos en una distribución con una media de 0 y una desviación estándar de 1.

Cada valor estandarizado se calcula restando la media de la característica correspondiente y luego dividiendo por la desviación estándar.

Se utilizan los métodos mean() y std(). El puntaje z generalmente varía de -3.00 a 3.00.

* El escalado robusto: Se escala cada característica del conjunto de datos restando la mediana y luego dividiendo por el rango intercuartílico.

El rango intercuartílico (RIC) se define como la diferencia entre el tercer y el primer cuartil y representa el 50% central de los datos.

## Lecciones aprendidas

* Repaso completo de los temas abordados en clases previas.
* Diferentes formas para imputar valores ya sean numéricos o string.
* Como cambiar columnas numéricas a una escala común.

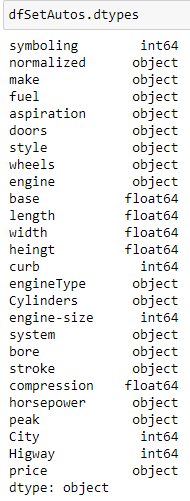
# Semana 8

## Síntesis de los contenidos abordados en la sesión y evidencias

**Estandarización de datos:**

**Verificar dominio de las variables y ajustarlo de ser necesario.**

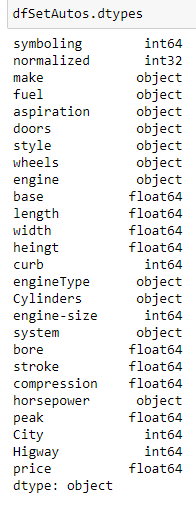
* Verificar los tipos de datos en cada columna con el comando .dtypes.



* Normalización los datos:



#Se toman solo las columnas del DataSet que se quieren normalizar y se verifica si ahora tienen el mismo tipo de dato.



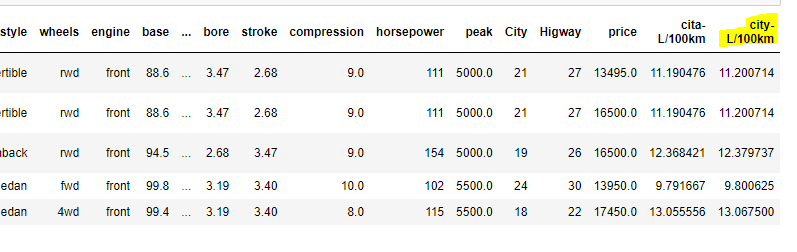
**Estandarización de los valores (convertir los datos en medidas conocidas)**

* El gasto de combustible en ciudad y carretera están en MPG (millas por galón) y se debe convertir a litros y kilómetros.

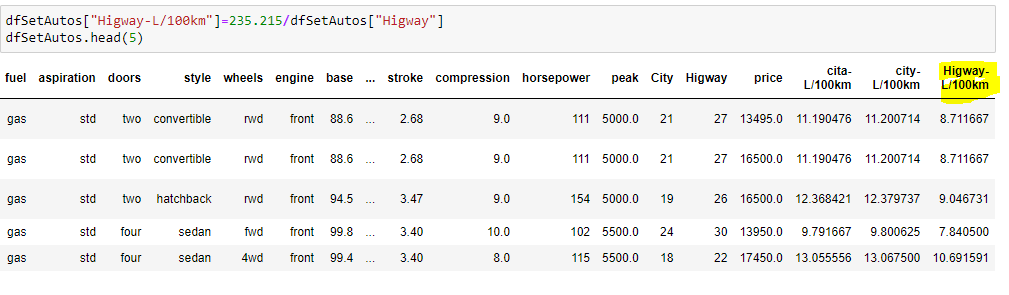
1 milla x galón=235.215 litros por 100 km

* Para estandarizar los datos se crea una columna nueva para no modificar el set original de datos.





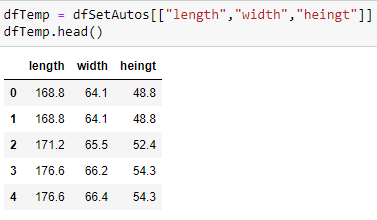
* Se realiza el mismo procedimiento con la columna de higway.



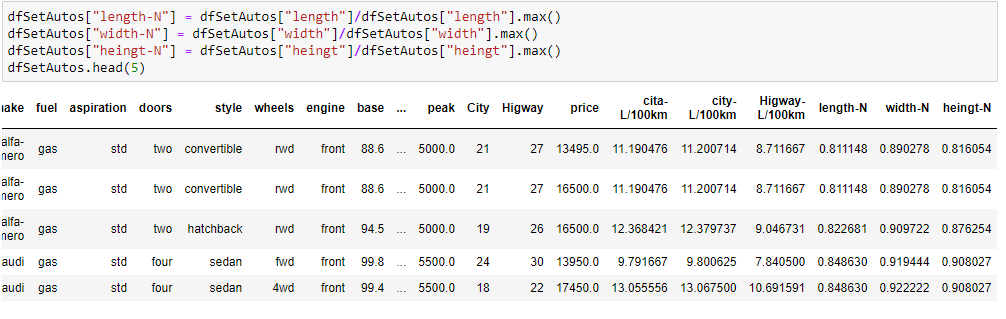
**Normalizar los datos para que los valores queden entre 0 y1 para que sean comparables:**



# Valores no comparables ya que son diferentes, pero puede ser necesario realizar una comparación.

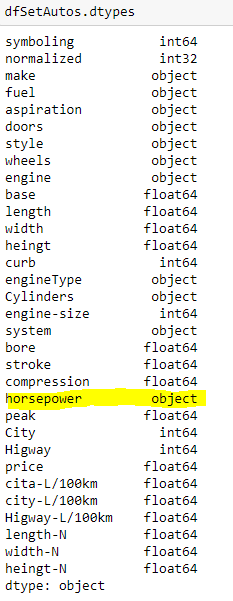


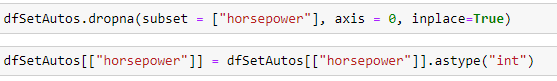
* Se deben convertir los valores entre 0 y 1 para poder comparar los valores.
* Cada valor se divide entre el valor máximo.

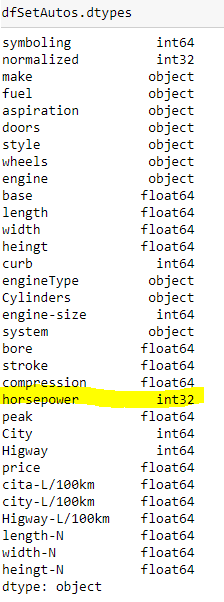


**Creacion de Binnings (rangos):**

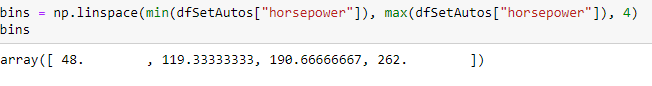
* Se crean rangos para dividir en grupos más pequeños (categorías) una columna.
* Se elige la columna de “horsepower” y se eliminan los datos faltantes ya que son pocos porque el tipo de dato aparece como object.
* Los datos se convierten a entero.



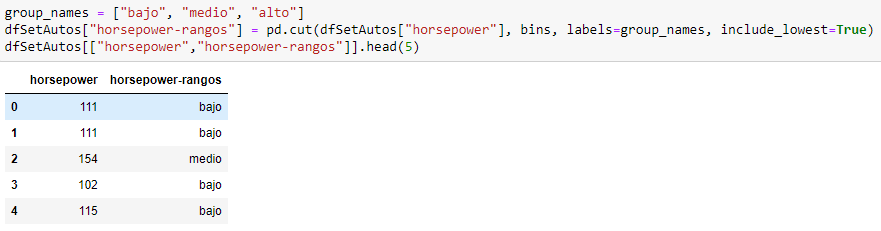




* Se crean los rangos:

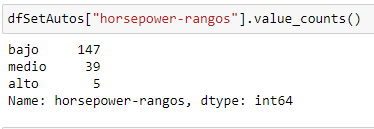


#Se crean 3 rangos por lo que se divide en 4.



#Se crean los nombres de los rangos, se corta la columna en los rangos creados y se incluye el valor mas bajo para que se considere.

#Se muestra la columna de horsepower a la par de la columna con los rangos junto con los primeros 5.



#Se cuentan los valores por categoría.

## Desarrollo de actividades y lecturas complementarias

NA

## Lecciones aprendidas

* Estandarización de datos en medidas conocidas.
* Normalización de datos para que sean comparables entre sí (entre 0 y 1).
* Creacion de rangos (binnings) para dividir en categorías datos numéricos.

# Semana 9

## Síntesis de los contenidos abordados en la sesión y evidencias

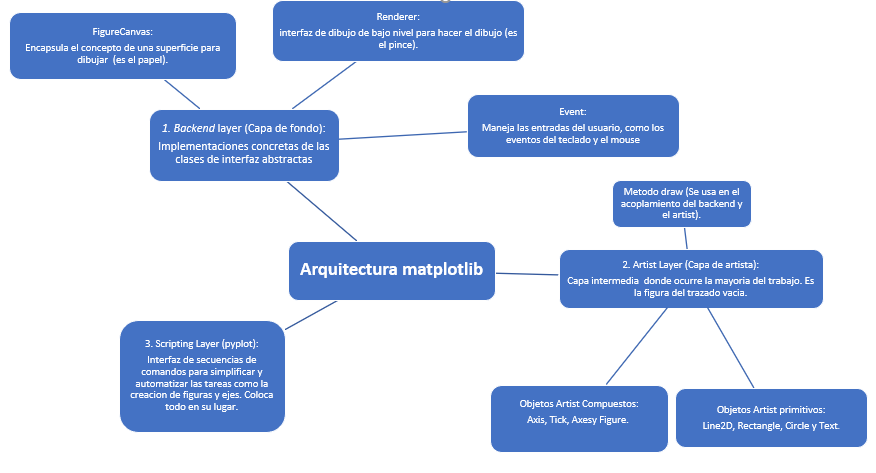
Clase asincrónica

## Desarrollo de actividades y lecturas complementarias

**Matplotlib**

Biblioteca de trazado basada en Python con soporte completo para 2D y soporte limitado para gráficos 3D.

**Arquitectura matplotlib:**



**Refactorización del back-end:**

La cantidad de métodos requeridos en la API de backend se redujo a:

* draw\_path: Dibuja polígonos compuestos, formados por líneas y segmentos de Béezier.
* draw\_image: dibuja imágenes de trama.
* draw\_text: Dibuja texto con las propiedades de fuente dadas.
* get\_text\_width\_height\_descent: Dada una cadena de texto, devuelve sus métricas.

La lista completa de métodos API de back-end opcionales es:

* draw\_markers: Dibuja un conjunto de marcadores.
* draw\_path\_collection: Dibuja una colección de caminos.
* draw\_quad\_mesh: Dibuja una malla cuadrilátera.

**Transformación:**

Cada artist tiene un nodo de transformación que sabe cómo transformar de un sistema de coordenadas a otro.

Estos sistemas de coordenadas incluyen:

* **datos:** los valores originales de datos sin procesar.
* **ejes:** el espacio definido por un rectángulo de ejes particular.
* **figura:** el espacio que contiene la figura completa.
* **visualización:** las coordenadas físicas utilizadas en la salida (por ejemplo, puntos en PostScript, píxeles en PNG).

**The Polyline Pipeline (El conducto polilínea):**

Serie de pasos que se realizan para pasar de los datos sin procesar a la línea dibujada en la pantalla.

* **Transformación:** Las coordenadas se transforman de coordenadas de datos a coordenadas de figuras.
* **Manejar datos faltantes:** la matriz de datos puede tener partes en las que faltan datos o no son válidos. El usuario puede indicar esto configurando esos valores en NaN o usando matrices de numpy enmascaradas (masked arrays).
* **Clipping (recorte):** los puntos fuera de los límites de la figura pueden aumentar el tamaño del archivo al incluir puntos invisibles. También, los valores de coordenadas muy grandes o pequeños pueden causar errores de desbordamiento lo que da como resultado una salida distorsionada.
* **Snapping:** las líneas perfectamente verticales y horizontales pueden verse borrosas debido al antialiasing cuando sus centros no están alineados con el centro de un píxel.
* **Simplificación: A**l trazar gráficos muy densos, es posible que muchos de los puntos de la línea no sean visibles. La inclusión de estos puntos en el gráfico aumenta el tamaño del archivo y puede alcanzar límites en la cantidad de puntos permitidos. Por lo tanto, se eliminan todos los puntos que se encuentran exactamente en la línea entre sus dos puntos vecinos

**Texto matemático:**

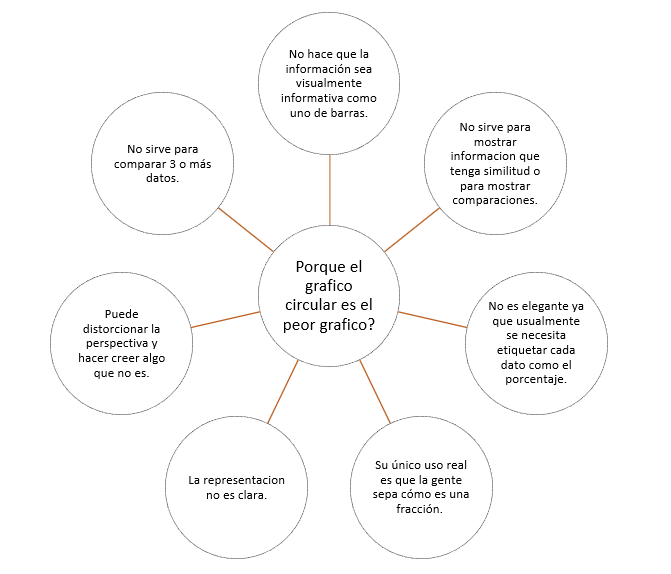
Es útil colocar expresiones matemáticas con formato enriquecido directamente en el gráfico para los usuarios.

Donald Knuth's TeX typesetting system:

Es la sintaxis más utilizada para las expresiones matemáticas. Es una forma de convertir la entrada en un lenguaje de texto sin formato como este: \sqrt{\frac{\delta x}{\delta y}}

**Pruebas de regresión:**

El script de prueba actual de matplotlib genera una serie de gráficos, pero en lugar de requerir una intervención manual, esos gráficos se comparan automáticamente con las imágenes de referencia. Todas las pruebas se ejecutan dentro del marco de pruebas, lo que hace que sea muy fácil generar un informe de las pruebas que fallaron.



## 3. Lecciones aprendidas

* Como es la arquitectura de la librería Matplotlib y cómo funciona.
* Desventajas del grafico circular para visualizar información.

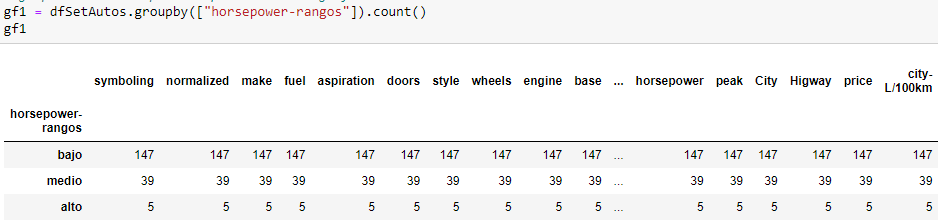
# Semana 10

## Síntesis de los contenidos abordados en la sesión y evidencias

**Visualización de datos**

Visualización de datos usando matplotlib para crear las gráficas:

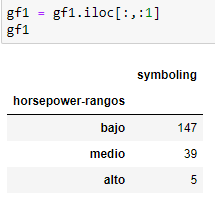
* Agrupación de datos para crear la gráfica:



.iloc: Se utiliza para seleccionar filas y columnas. Sintaxis: data.iloc[<filas>, <columnas>

El primer parametro es la fila hasta N cantidad de filas. Por ejemplo [0:10] (desde la fila 0 a la 10) si se pone vacío [:] significa todo.

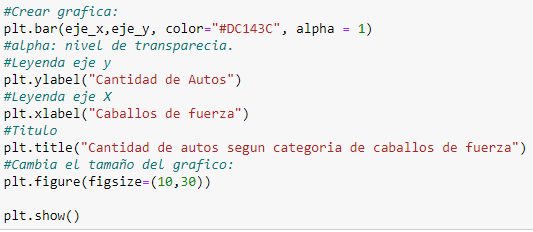
El segundo parámetro representa la columna. [:1] (Significa que va desde el principio a la primera columna)



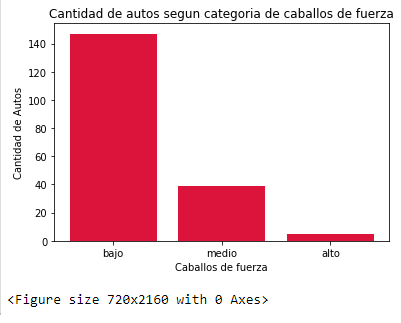
* Creación de arreglos para hacer el grafico con los ejes:



* Creación de la gráfica:



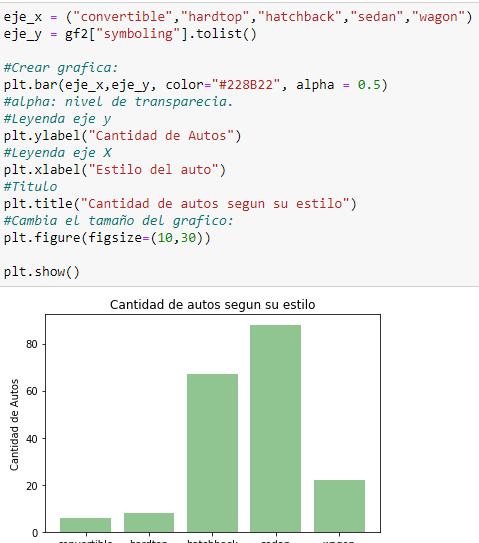
El .bar se usa para hacer el grafico vertical, si fuera horizontal es .barh



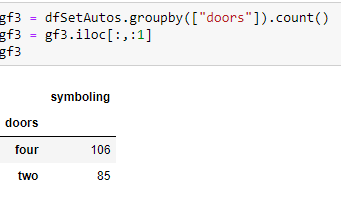
* Agrupación para la creación del segundo grafico:



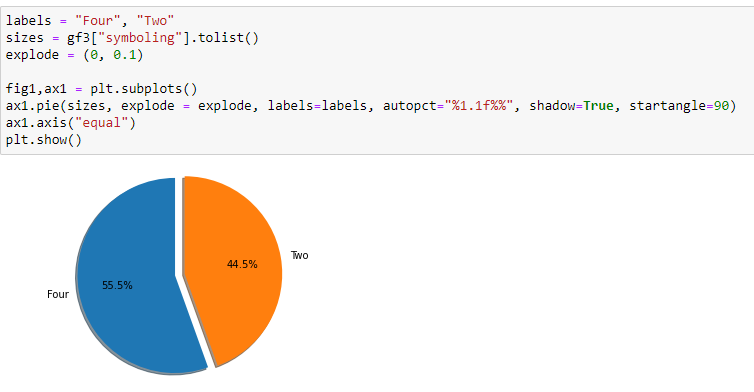
* Creación del segundo grafico:



* Agrupación para el tercer grafico (circular):



* Creación del tercer grafico:



El explode se usa para separación de las partes con respecto a la otra.

Autopct: formato de los datos en valor porcentual.

## Desarrollo de actividades y lecturas complementarias

NA

## Lecciones aprendidas

* Uso de la librería de Matplotlib para crear gráficos.
* Creación de diferentes tipos de gráficos de acuerdo a la necesidad.
* Agrupación de los datos con .groupby() para la creación del gráfico.

# Semana 11

## Síntesis de los contenidos abordados en la sesión y evidencias

## 2. Desarrollo de actividades y lecturas complementarias

## 3. Lecciones aprendidas

# Semana 12

## Síntesis de los contenidos abordados en la sesión y evidencias

## 2. Desarrollo de actividades y lecturas complementarias

## 3. Lecciones aprendidas