Análisis de Datos

Presentado por: Dr. Octavio Augusto Muñoz Román

22 de mayo 2021

¿Qué es el análisis de datos?

Un proceso de **inspección**, **limpieza**, **transformación** y **modelado de datos** con el objetivo de **descubrir información útil**, **informar conclusiones** y **respaldar la toma de decisiones**.

Herramientas para el manejo y análisis de datos

Auto-managed closed tools

Qlik

tableau

looker

Zoho Analytics

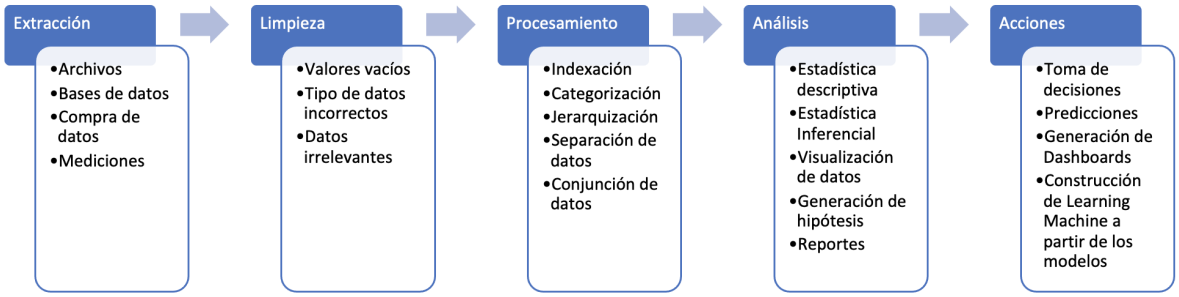
Programming Languages

python

R

julia

Proceso para el análisis de datos



¿Por qué la estadística?

La estadística es la manera de tratar y enteder los datos y poderles dar una interpertación

coherente.

Los datos los podemos representar de maneras diferentes:

- numéricamente (tablas)
- gráficamente

Los datos pueden venir de diferentes fuentes y pueden ser clasificados en dos grandes tipos

- orgánicos: son aquellos que provienen de la naturaleza o de una actividad del ser humano y se registran casí a tiempo real y que por lo general son procesados por un sistema computacional
    - Ejemplos: Datos deportivos, transacciones financieras, historial de vistas de Netflix, sensores de temperatura, etc.
- diseñados: Es cuando usamos las muestras poblacionales para no tener que trabajar la población completa.

La estadística puede tener diferentes puntos de vista llamadas perspectivas dependiendo del propósito.

- Arte de resumir los datos
- Ciencia de la incertidumbre
- Toma de decisiones
- Predicciones

Software a utilizar

Para este taller estaremos usando una herramienta Web llamada Google Colab (https://colab.research.google.com)

y el archivo de datos que estaremos usando se encuentra en https://raw.githubusercontent.com/omunozgit/panda/main/temblor.csv

Extrayendo los datos

Primero que nada introduciremos las librerias que necesitaremos para trabajar

import pandas as pd #Manejo de datos

import matplotlib.pyplot as graf #visualización de gráficos

In [ ]:
```
import pandas as pd
import matplotlib.pyplot as graf
```

- **Indicamos la ruta del origen de los datos**

url = "ruta_del_documento"

- **Cargamos los datos del archivo.**

Para cargar el archivo tenemos dos opciones

a) Solo indicamos el nombre del archivo

pd.read_csv(url)

b) Indimcamos el nombre del archivo y la columna que servirá como índice

pd.read_csv(url)

pd.read_csv(url, index_col = 0)

c) Reconvertir un tipo de dato en otro

df["columna"] = pd.to_numeric(df['columna'], errors='coerce').fillna(0)

In [ ]:
```
url = "https://raw.githubusercontent.com/omunozgit/panda/main/temblor.csv"
df = pd.read_csv(url)

df["Deep"] = pd.to_numeric(df['Deep'], errors='coerce').fillna(0)
```

- **Mostramos como se conforma el dataFrame**

La información que nos muestra este comando es:

La columna índice, el nombre de las columnas (Tantas como tenga el archivo), cantidad de columnas con y sin datos datos y el tipo de dato de cada columna.

In [ ]:
```
df.info()
```

- **Mostramos la información del total de renglones y columnas que tiene el dataFrame con datos**

Usamos la instrucción **shape** del dataframe

In [ ]:
```
df.shape
```

- **Mostramos la información de la tabla completa**

In [ ]:
```
df
```

Visualización numérica de los datos

- **Mostramos el primeros registros de la tabla**

usando el método **head()** del dataframe. Recuerda que los registros comienzan con el indice cero

head() devuelve los primeros 5 registrso de la tabla

head(n) devuelve los "n" registros indicados, de arriba hacia abajo

In [ ]:
```
df.head()
```

- **Mostramos los últimos registrso de la tabla**

usando el método **tail()**

tail() devuelve los últimos 5 registrso de la tabla

tail(n) devuelve los "n" registros indicados, de abajo hacia arriba

In [ ]:
```
df.tail()
```

Mostrando ciertas columnas del dataFrame

Usaos el dataframe **df["nombre_columna"]**

Usamos la instrucción **loc[: ,["nombre_columna"]]**

Si queremos mostrar solo algunos renglones usamos **loc[n:m, ["nombre_columna"]]**, donde "n" representa el índice del renglon donde quiero comenzar y "m" represanta el índice del renglón donde quiero terminar.

NOTA: El índice siempre es la primera columna

In [ ]:
```
df["Km"]
```

In [ ]:
```
df.loc[:,["Deep"]]
```

In [ ]:
```
df.loc[:4,["Entidad"]]
```

In [ ]:
```
df.loc[df["Entidad"]=="QRO"],["Magnitud"]
```

In [ ]:
```
df["Magnitud"].describe()
```

In [ ]:
```
df.loc[:,["Magnitud"]].std()
```

In [ ]:
```
df.agg({"Deep":["min","max","mean","median","std"]})
```

In [ ]:
```
df.loc[df["Entidad"]=="BC"].agg({"Deep":["min","max","mean","median","std"]})
```

In [ ]:
```
df.groupby(["Anio","Mes"])["Magnitud"].mean()
```

In [ ]:
```
lista = ["Magnitud","Deep","Km"]
df.groupby(["Anio","Mes"])[lista].mean()
```

In [ ]:
```
df.groupby(["Anio","Mes"]).agg({"Magnitud":["min","max","mean","median","std"
```

In [ ]:
```
df.groupby(["Entidad","Localidad"])["Magnitud"].describe()
```

Visualización gráfica de los datos

- **Grafiquemos un histograma**

In [ ]:
```
df["Magnitud"].hist(bins = 10, grid=False)
graf.show()
```

In [ ]:
```
df["Magnitud"].plot.kde(bw_method=10)
graf.show()
```

In [ ]:
```
n1 = df.loc[0:30,["Km","Magnitud"]]
n1.plot(figsize=(10, 5), grid=True)
graf.title("Grafica de Km")
graf.xlabel("Dia")
graf.ylabel("KM")
graf.show()
```

In [ ]:
```
df.plot.scatter(x="Deep", y="Magnitud")
graf.show()
```

In [ ]:
```
#https://matplotlib.org/stable/tutorials/colors/colormaps.html
df.plot.scatter(x="Deep", y="Magnitud", c = "Km", colormap="Reds")
graf.show()
```

In [ ]:
```
df.loc[df["Mes"]==1,["Magnitud","Deep"]].plot.box(figsize=(15, 8), subplots=T
graf.show()
```

In [ ]:
```
df.loc[:,["Magnitud","Deep"]].plot.box(figsize=(15, 3))
graf.show()
```

In [ ]:
```
df.loc[:,["Magnitud","Deep"]].plot.box(figsize=(15, 3), subplots=True)
graf.show()
```

In [ ]:
```
df.loc[df["Entidad"]=="MICH"].groupby(["Entidad","Localidad"])["Magnitud"].me
graf.show()
df.loc[df["Entidad"]=="GRO"].groupby(["Entidad","Localidad"])["Magnitud"].mea
graf.show()
df.loc[df["Entidad"]=="CHIS"].groupby(["Entidad","Localidad"])["Magnitud"].me
graf.show()
```

In [ ]:
```
df.loc[df["Entidad"]=="MICH"].groupby(["Entidad","Localidad"])["Magnitud"].me
graf.show()
df.loc[df["Entidad"]=="GRO"].groupby(["Entidad","Localidad"])["Magnitud"].mea
graf.show()
df.loc[df["Entidad"]=="CHIS"].groupby(["Entidad","Localidad"])["Magnitud"].me
graf.show()
```

In [ ]:
```
df.loc[df["Entidad"]=="MICH"].groupby(["Entidad","Magnitud","Localidad"])["Ma
graf.show()
```