## Práctica de Pandas con datos

1. Dados los archivos .txt, users, ratings y movies. Ejecutar el siguiente código y escribir como comentario, detallando qué realiza cada bloque. Tener en cuenta que las rutas de acceso pueden ser otras. Por último, genera el código de agrupamiento y agregación necesario para calcular: suma, cuenta, media, desviación estándar, utilizando las funciones de numpy (ej: np.sum)

**read_table**: permite leer datos de un archivo txt:

In [ ]:
```
import pandas as pd

userHeader = ['user_id', 'gender', 'age', 'ocupation', 'zip']
users = pd.read_table('archs/dataset/users.txt', engine='python', sep='::',
                      header=None, names=userHeader)
```

In [ ]:
```
ratingHeader = ['user_id', 'movie_id', 'rating', 'timestamp']
ratings = pd.read_table('archs/dataset/ratings.txt', engine='python', sep='::',
                        header=None, names=ratingHeader)
```

In [ ]:
```
mergeRatings = pd.merge(users, ratings, on='user_id')
```

In [ ]:
```
mergeRatings = mergeRatings.drop(['user_id','zip','timestamp','ocupation'], axis=1)
```

In [ ]:
```
movieHeader = ['movie_id', 'title', 'genders']
movies = pd.read_table('archs/dataset/movies.txt', engine='python', sep='::', header=None,
                       names=movieHeader, encoding='latin-1')
```

In [ ]:
```
movies[movies.title.str.contains("Exorcist")]
```

In [ ]:
```
merge = pd.merge(mergeRatings, movies)
```

In [ ]:
```
merge.groupby('gender').size().plot(kind='bar', fontsize=10, rot=45, color='turquoise')
```

In [ ]:
```
merge["Género"] = merge["genders"].str.split('|', n=1, expand= True)[0]
```

In [ ]:
```
colors = ['magenta','tan','mediumseagreen','orange','blueviolet', 'gold', 'salmon', 'limegreen']
merge.groupby('Género').size().plot(kind='bar', color=colors)
```

In [ ]:
```
info1000 = merge.loc[1000]
```

In [ ]:
```
info7_12 = merge[7:12]
```

In [ ]:
```
numberRatings = merge.groupby('title').size().sort_values(ascending=False)
numberRatings[:5]
```

In [ ]:
```
avgRatings = merge.groupby(['movie_id', 'title']).mean()
avgRatings['rating'][:10]
```

In [ ]:
```
dataRatings = merge.groupby(['movie_id', 'title'])['rating'].agg(['mean', 'sum', 'count', 'std'])
dataRatings[:10]
```

2. El archivo `cotizacion.csv` contiene cotizaciones de acciones de empresas con las siguientes columnas:
'nombre' (nombre de la empresa),
'Final' (precio de la acción al cierre de bolsa),
'Máximo' (precio máximo de la acción durante la jornada),
'Mínimo' (precio mínimo de la acción durante la jornada),
'volumen' (Volumen al cierre de bolsa),
'Efectivo' (capitalización al cierre en miles de euros).
Construir una función que construya un DataFrame a partir del archivo con el formato anterior y devuelva otro DataFrame con el mínimo, el máximo y la media de dada columna.

3. El archivo autos.xlsx contiene datos de precios de autos y stock. Construye el código necesario que emita el precio mínimo, el máximo y promedio.

4. El archivo comercio_interno.csv contiene información sobre el comercio interno desde la década del 90. Escribe un programa que:

a. Muestre por pantalla las dimensiones del Data Frame, el número de datos que contiene, los nombres de sus columnas y filas, los tipos de datos de las columnas, las 10 primeras filas y las 10 últimas filas.
b. Muestre por pantalla un gráfico de los datos de empleo por provincia y su relación con la columna valor.
c. Muestre por pantalla la columna alcance_nombre ordenada alfabéticamente.
d. Muestre un gráfico de la actividad_producto_nombre agrupados en relación al valor
e. Sume por alcance_nombre los valores de los años 2009 al 2019
f. Muestre un gráfico con el siguiente agrupamiento:

```
df.groupby('actividad_producto_nombre')['valor'].mean().plot(kind='pie',
                                            legend='Reverse',
                                            autopct='%0.2f %%',
                                            fontsize=6,
                                            labels=None,
                                            pctdistance=1.05)
```

g. Muestre un gráfico de la actividad_producto_nombre en la provincia de Mendoza del año 2015 al 2019

5. El archivo salarios muestra distintas categorías, antigüedad, salarios, etc.:

a. Calcula el mínimo, máximo y promedio de antigüedad.
b. Construye el código necesario para emitir un gráfico que muestre los porcentajes de cada cargo.
c. Genera el código de agrupamiento y agregación necesario para calcular: suma, media y desviación estándar del salario, utilizando las funciones de numpy (ej: np.sum)

6. El archivo titanic.csv contiene información sobre los pasajeros del Titanic. Escribir un programa con los siguientes requisitos:

a. Generar un DataFrame con los datos del archivo.
b. Emitir las dimensiones del DataFrame, el número de datos que contiene, los nombres de columnas, filas y los tipos de datos.
c. Emitir los datos del pasajero con identificador 56.
d. Emitir los nombres de las personas que iban en primera clase ordenadas alfabéticamente.
e. Emitir el porcentaje de personas que sobrevivieron y murieron.
f. Emitir el porcentaje de personas que sobrevivieron en cada clase.
g. Eliminar del DataFrame los pasajeros con edad desconocida.
h. Emitir la edad media de las mujeres que viajaban en cada clase.
i. Agregar una nueva columna booleana para conocer si el pasajero era menor de edad o no.
j. Emitir el porcentaje de menores y mayores de edad que sobrevivieron en cada clase.