Actividad 1: Sergio Buitrago

Materia: Diplomado de Analisis de datos con Python

1. Importando libreria pandas, cargando fuentes de datos y haciendo analisis descriptivo

Fuente de información: https://raw.githubusercontent.com/shecho30/Diplomado_python/main/Data/imdb.csv

Resumen DataSet: calificación, de las películas por medio de la aplicación IMDB.

```
import pandas as pd
```

```
from google.colab import files
```

```
# files.upload() # Cargar un archivo desde mypc
# df = pd.read_csv('imdb.csv') cargar manual
url = 'https://raw.githubusercontent.com/shecho30/Diplomado_python/main/Data/imdb.csv' #Conec
df = pd.read_csv(url, sep =',') #separar por comas
```

Cargamos el data source de IMDB que tenemos cargados en nuestro repositorio en github

```
df.head(10)
```

```
            Name  Date  Rate   Votes     Genre  Duration   Type  Certificate  Episode:
```

1. Haciendo el analisis de los primeros 10 registros del data source, se puede ver lo que vamos a encontrar en la base de datos, muchos variables tanto numericas como de texto.

```
                                        Crime.
```

```
df.isnull().sum()
```

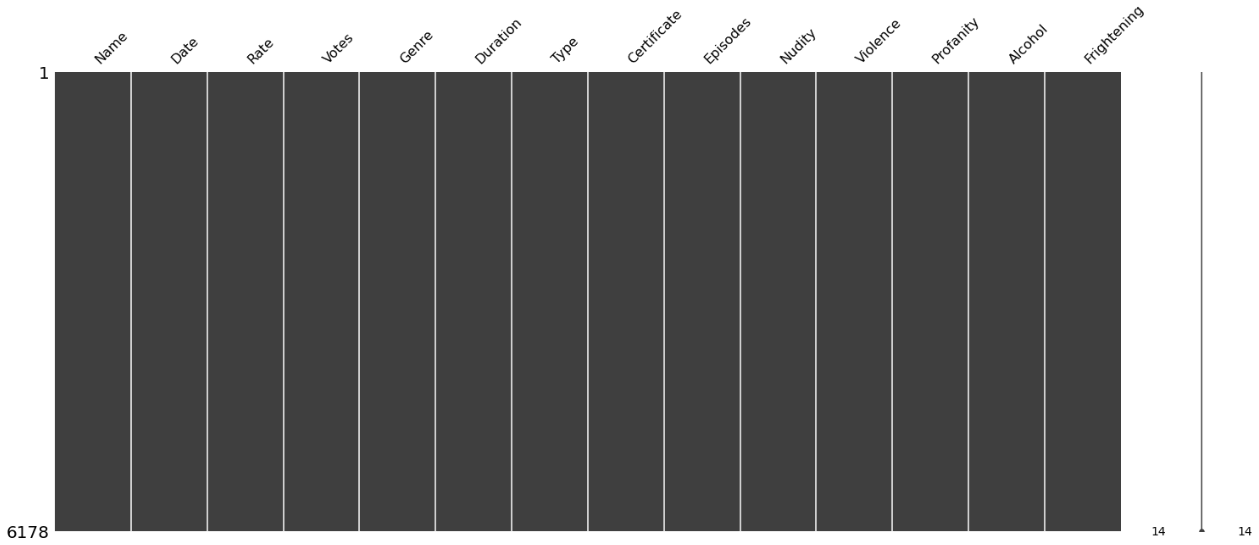
```
     Name           0
     Date           0
     Rate           0
     Votes          0
     Genre          0
     Duration       0
     Type           0
     Certificate    0
     Episodes       0
     Nudity         0
     Violence       0
     Profanity      0
     Alcohol        0
     Frightening    0
     dtype: int64
```

```
!pip install missingno
```

```
     Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/publ
     Requirement already satisfied: missingno in /usr/local/lib/python3.7/dist-packages (0.5
     Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from mis
     Requirement already satisfied: seaborn in /usr/local/lib/python3.7/dist-packages (from m
     Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (fro
     Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from mis
     Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-pac
     Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/li
     Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (f
     Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packag
     Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packag
     Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from 
     Requirement already satisfied: pandas>=0.23 in /usr/local/lib/python3.7/dist-packages (f
     Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (f
```

```
import missingno as msno
msno.matrix(df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa10f647cd0>
```



2. Como vemos no existen datos en blancos, los campos que estan vacios o NAN, los campos que estan en 0 en la columna Episodio, hacen refencia a peliculas.
3. Confirmamos con la libreria missingno para comprobar que no tenemos datos NAN en nuestro dataset.

Analizando el DataSet

```
print(df.head())
```

```
                            Name  Date Rate    Votes  \
0                No Time to Die  2021  7.6  107,163
1                    The Guilty  2021  6.3   64,375
2   The Many Saints of Newark  2021  6.4   27,145
3  Venom: Let There Be Carnage  2021  6.4   30,443
4                          Dune  2021  8.3   84,636

                         Genre Duration  Type Certificate Episodes    Nudity  \
0  Action, Adventure, Thriller     163   Film       PG-13         -      Mild
1       Crime, Drama, Thriller      90   Film           R         -      None
2                 Crime, Drama     120   Film           R         -  Moderate
3    Action, Adventure, Sci-Fi      97   Film       PG-13         -      None
4     Action, Adventure, Drama     155   Film       PG-13         -      None

   Violence Profanity   Alcohol Frightening
0  Moderate      Mild      Mild    Moderate
1      None    Severe      None    Moderate
2    Severe    Severe  Moderate    Moderate
3  Moderate  Moderate      Mild    Moderate
4  Moderate      None      Mild    Moderate
```

```
df.head()
```

	Name	Date	Rate	Votes	Genre	Duration	Type	Certificate	Episodes
0	No Time to Die	2021	7.6	107,163	Action, Adventure, Thriller	163	Film	PG-13	-
1	The Guilty	2021	6.3	64,375	Crime, Drama, Thriller	90	Film	R	-
	The								

```
df.shape
```

```
(6178, 14)
```

4. Con la propiedad shape del dataframe, podemos ver que nuestra tabla tiene un mañao de 14 columnas y 6178 registros.

```
df.dtypes
```

```
Name           object
Date            int64
Rate           object
Votes          object
Genre          object
Duration       object
Type           object
Certificate    object
Episodes       object
Nudity         object
Violence       object
Profanity      object
Alcohol        object
Frightening    object
dtype: object
```

5. Con la propiedad Types del dataframe, podemos ver el nombre de cada serie y el tipo de datos que almacena, donde casi todos son Object.

```
df.describe
```

```
<bound method NDFrame.describe of                                         Name  Date
Rate    Votes  \
0                             No Time to Die  2021  7.6  107,163
1                                 The Guilty  2021  6.3   64,375
2                  The Many Saints of Newark  2021  6.4   27,145
3                Venom: Let There Be Carnage  2021  6.4   30,443
4                                       Dune  2021  8.3   84,636
...                                      ...   ...  ...      ...
6173  The Human Centipede II (Full Sequence)  2011  3.8   37,492
6174                        Double Indemnity  1944  8.3  150,448
6175      Before the Devil Knows You're Dead  2007  7.3  100,668
6176                              Queen Bees  2021  6.0      887
6177                              Death Race  2008  6.3  203,578

                            Genre Duration  Type Certificate Episodes  \
0     Action, Adventure, Thriller     163   Film       PG-13        -
1          Crime, Drama, Thriller      90   Film           R        -
2                    Crime, Drama     120   Film           R        -
3       Action, Adventure, Sci-Fi      97   Film       PG-13        -
4        Action, Adventure, Drama     155   Film       PG-13        -
...                           ...      ...   ...         ...      ...
6173                       Horror      91   Film   Not Rated        -
6174      Crime, Drama, Film-Noir     107   Film      Passed        -
6175       Crime, Drama, Thriller     117   Film           R        -
6176       Comedy, Drama, Romance     100   Film       PG-13        -
6177     Action, Sci-Fi, Thriller     105   Film           R        -

        Nudity  Violence Profanity   Alcohol Frightening
0         Mild  Moderate      Mild      Mild    Moderate
1         None      None    Severe      None    Moderate
2     Moderate    Severe    Severe  Moderate    Moderate
3         None  Moderate  Moderate      Mild    Moderate
4         None  Moderate      None      Mild    Moderate
...        ...       ...       ...       ...         ...
6173    Severe    Severe    Severe      Mild      Severe
6174      None      Mild      None      Mild        Mild
6175    Severe  Moderate    Severe    Severe      Severe
6176      None      None      Mild  Moderate        None
6177      Mild    Severe    Severe      Mild    Moderate

[6178 rows x 14 columns]>
```

```
df['Genre'].value_counts()
```

```
Comedy                      268
Drama                       259
Crime, Drama, Mystery       220
```

```
Comedy, Drama              199
Drama, Romance             189
                          ...
Action, Thriller, War        1
Comedy, Crime, Musical       1
Short, Drama, Romance        1
Animation                    1
Drama, Fantasy, Thriller     1
Name: Genre, Length: 377, dtype: int64
```

6. Con la siguiente formula hacemos un conteo de los generos de las peliculas, donde podemos analizar que la principal es **comedy** seguido de **drama**

```
mov_comedy = df.loc[(df['Genre']=='Comedy')].head(10)
```

```
mov_comedy
```

	Name	Date	Rate	Votes	Genre	Duration	Type	Certificate	Episo
16	Seinfeld	2021	8.8	272,028	Comedy	22	Series	TV-PG	
60	The Office	1993	8.9	475,207	Comedy	22	Series	TV-14	
134	It's Always Sunny in Philadelphia	2021	8.8	201,983	Comedy	22	Series	TV-MA	
136	Superstore	2021	7.8	39,602	Comedy	22	Series	TV-14	
152	Young Sheldon	2021	7.5	41,356	Comedy	30	Series	TV-PG	
190	Schitt's Creek	2017	8.5	97,987	Comedy	22	Series	TV-14	
208	Curb Your Enthusiasm	2001	8.7	111,076	Comedy	28	Series	TV-MA	

7. Podemos guardar en una variable, un dataframe con la informacion filtrada.

Conclusion

Este analisis descriptivo de la fuente de informacion de imbd nos lleva a concluir que es una fuente que nos permitira, sacar los datos estadisticos de las peliculas y series calificadas por la aplicacion de Imdb y nos permitira, conocer y recomendar peliculas dependiendo mis gustos.

Fuente de información 2: https://api.covidtracking.com/v1/us/daily.json

Resumen DataSet: Web Api, de los registros de informacion sobre las personas contagiadas de covid-19

```
url = 'https://api.covidtracking.com/v1/us/daily.json'
```

```
df2 = pd.read_json('https://api.covidtracking.com/v1/us/daily.json')
df2
```

uCurrently	inIcuCumulative	onVentilatorCurrently	...	lastModified	recovered	t
8134.0	45475.0	2802.0	...	2021-03-07T24:00:00Z	NaN	
8409.0	45453.0	2811.0	...	2021-03-06T24:00:00Z	NaN	
8634.0	45373.0	2889.0	...	2021-03-05T24:00:00Z	NaN	
8970.0	45293.0	2973.0	...	2021-03-04T24:00:00Z	NaN	
9359.0	45214.0	3094.0	...	2021-03-03T24:00:00Z	NaN	
...	...	...	...	...	...	
NaN	NaN	NaN	...	2020-01-17T24:00:00Z	NaN	
NaN	NaN	NaN	...	2020-01-16T24:00:00Z	NaN	
NaN	NaN	NaN	...	2020-01-15T24:00:00Z	NaN	
NaN	NaN	NaN	...	2020-01-14T24:00:00Z	NaN	
NaN	NaN	NaN	...	2020-01-13T24:00:00Z	NaN	

```
df2.dtypes
```

```
date                              int64
```

```
states                        int64
positive                    float64
negative                    float64
pending                     float64
hospitalizedCurrently       float64
hospitalizedCumulative      float64
inIcuCurrently              float64
inIcuCumulative             float64
onVentilatorCurrently       float64
onVentilatorCumulative      float64
dateChecked                  object
death                       float64
hospitalized                float64
totalTestResults              int64
lastModified                 object
recovered                   float64
total                         int64
posNeg                        int64
deathIncrease                 int64
hospitalizedIncrease          int64
negativeIncrease              int64
positiveIncrease              int64
totalTestResultsIncrease      int64
hash                         object
dtype: object
```

```
df2['new_date'] = pd.to_datetime(df2['date'], format='%Y%m%d')
df2['new_date']
```

```
0     2021-03-07
1     2021-03-06
2     2021-03-05
3     2021-03-04
4     2021-03-03
         ...
415   2020-01-17
416   2020-01-16
417   2020-01-15
418   2020-01-14
419   2020-01-13
Name: new_date, Length: 420, dtype: datetime64[ns]
```

Haz doble clic (o pulsa Intro) para editar

Ajustamos la variable de fecha, que al principio nos sale en formato string y la convertimos a date.

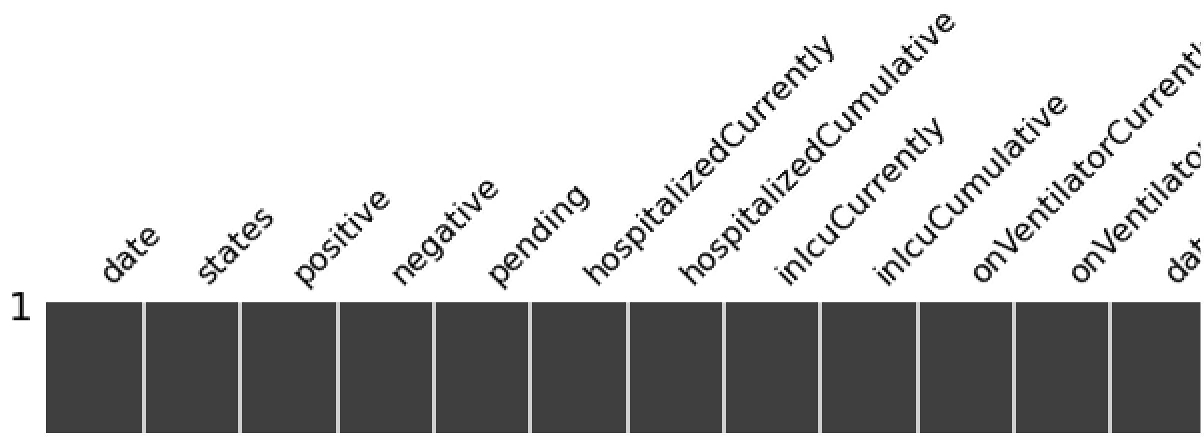
```
df2.isnull().sum()
```

```
date                         0
states                       0
positive                     1
negative                    48
```

```
pending                      51
hospitalizedCurrently        64
hospitalizedCumulative       51
inIcuCurrently               73
inIcuCumulative              72
onVentilatorCurrently        72
onVentilatorCumulative       79
dateChecked                   0
death                        28
hospitalized                 51
totalTestResults              0
lastModified                  0
recovered                   420
total                         0
posNeg                        0
deathIncrease                 0
hospitalizedIncrease          0
negativeIncrease              0
positiveIncrease              0
totalTestResultsIncrease      0
hash                          0
new_date                      0
dtype: int64
```

```
msno.matrix(df2)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa108af7910>
```



Vemos que esta base de datos si tiene valores NAN donde el valor con mas nulos es la Columna "Recovered"

```
df2.shape
```

```
(420, 26)
```

Vemos que este data set tiene 26 columnas y 420 registro

```
df2.dtypes
```

```
date                          int64
states                        int64
positive                    float64
negative                    float64
pending                     float64
hospitalizedCurrently       float64
hospitalizedCumulative      float64
inIcuCurrently              float64
inIcuCumulative             float64
onVentilatorCurrently       float64
onVentilatorCumulative      float64
dateChecked                  object
death                       float64
hospitalized                float64
totalTestResults              int64
lastModified                 object
recovered                   float64
total                         int64
posNeg                        int64
deathIncrease                 int64
hospitalizedIncrease          int64
negativeIncrease              int64
positiveIncrease              int64
totalTestResultsIncrease      int64
hash                         object
```

```
    new_date                    datetime64[ns]
    dtype: object
```

```
df2[['new_date','death']].head(1)
```

	new_date	death
0	2021-03-07	515151.0

```
df2['recovered'].isnull().sum()
df2['recovered']
```

```
    0      NaN
    1      NaN
    2      NaN
    3      NaN
    4      NaN
            ..
    415    NaN
    416    NaN
    417    NaN
    418    NaN
    419    NaN
    Name: recovered, Length: 420, dtype: float64
```

Conclusión:

Este datasource, nos permite tener el historico de registros de muertes y contagios del covid 19

Conclusiones Finales:

Para trabajar en el proyecto he decidido tomar la base de datos de IMDB ya que tiene datos cuantitativos y cualitativos que nos permitan hacer analisis no solo numericos si no tambien categoricos. tambien una ventaja que tiene con los otros data source que vimos es que tiene mas registros.

Productos de pago de Colab - Cancelar contratos

✓ 0 s completado a las 20:42 ● ×