```
# BTC-USDT Cotizacion de CoinMarketCap
# Date,Open,High,Low,Close,Adj Close,Volume
# mes 2014-10-01 -2022-06-10
```

Análisis de precio Anual y Mensual de Bitcoin

Un poco de analisis con Python y Pandas,
La tabla corresponde a la accion del precion de BTC anual y mensual.

Data

Open : Apertua de cotizacion del precio durante un periodo,

Close: cierre del precio durante un periodo

High : precio mas alto durante ese periodo

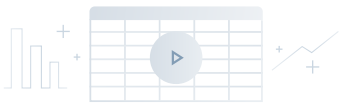
Low : precio mas bajo durante ese periodo

Adj Close: ajuste de cierre

Volume: Volumen de negociacion durante ese periodo

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

```
!ls /datasets/coinmarketcap-mescorte1-2014-10-01
```



Run the app to see the outputs
Press the run button in the top right corner

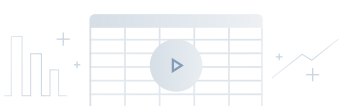
```
df = pd.read_csv('/work/BTC-USD-CoinMarketCap.csv')
```

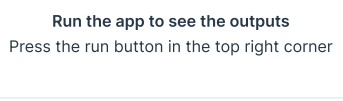
Acción del Precio

```
print(df.columns)
print(df.shape)
print(df.index)
print(df.index.name)
# df.dtypes
```


Run the app to see the outputs
Press the run button in the top right corner

```
round(df,2)
```





Run the app to see the outputs
Press the run button in the top right corner

```
df['Date'] = pd.to_datetime(df['Date'])
```

```
df_year = pd.DataFrame()
años = [2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022]
df_year = df.groupby(pd.Grouper(key='Date', axis=0, freq='Y', sort=True)).mean()
df_year = df_year.assign(Date=años)
df_year = df_year[['Date', 'Open', 'High','Low', 'Close', 'Volume']]
df_year.set_index(pd.Index([0,1,2,3,4,5,6,7,8]),inplace = True)
df_year = round(df_year,2)
```

Promedio de precio de compra anual

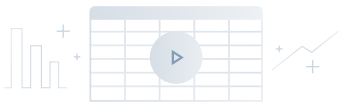
```
df_year
```


Run the app to see the outputs
Press the run button in the top right corner

```
plt.figure(figsize=(14, 8))
with plt.style.context('Solarize_Light2'):
    # plt.plot(df['Date'], df['Open'], "r")
    # plt.plot(df['Date'], df['Close'], "b")
    plt.plot(df['Date'], df['Low'],"g")
    plt.plot(df['Date'], df['High'],"c")

    # Number of accent colors in the color scheme
    plt.title('Accion del Precio')
    plt.xlabel('Tiempo', fontsize=15)
    plt.ylabel('Precio', fontsize=15)

plt.show()
```

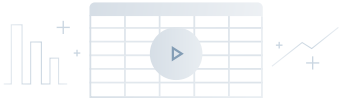


Run the app to see the outputs
Press the run button in the top right corner

Valor porcentual del precio Rendimiento Anual

```
year_percen = pd.DataFrame()
year_percen = df_year.diff()
year_percen = year_percen/df_year.shift()*100
year_percen = year_percen.assign(Date=años)
year_percen = round(year_percen,2)
```

```
year_percen
```

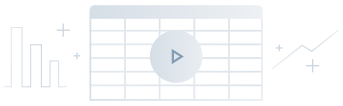


Run the app to see the outputs
Press the run button in the top right corner

```
y_pos = year_percen['Date']
performance = year_percen['Open']
# error = np.random.rand(len(year_percen))
fig, ax = plt.subplots(figsize=(15, 3))
ax.set(xlim=[-60, 550], xlabel='Total Revenue', ylabel='Company')
hbars = ax.barh(y_pos, performance, align='center')
ax.set_yticks(y_pos, labels=year_percen['Date'])
ax.invert_yaxis()  # labels read top-to-botto
ax.set_xlabel('porcentajes%')
ax.set_title('How fast do you want to go today?')

ax.bar_label(hbars, fmt='%.2f')

plt.show()
```



Run the app to see the outputs
Press the run button in the top right corner

Valor porcentual Rendimiento mes vs mes

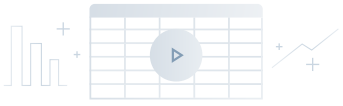
Delladado valor porcentual del mes de octubre

```
precioVsMonth = df[df['Date'].dt.month == 10]
precioVsMonth = round(precioVsMonth)
precioVsMonth
```


Run the app to see the outputs
Press the run button in the top right corner

```
dif = precioVsMonth['Open'].diff()
dif = dif/precioVsMonth['Open'].shift()
dif = dif*100
```

```
dif
```

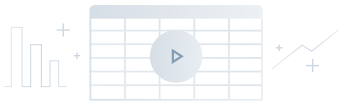


Run the app to see the outputs
Press the run button in the top right corner

Análisis del Volumen de negociación mes

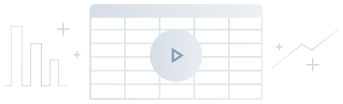
Este análisis permite intuir la cantidad de negociación que se generan por mes, permite ver cierto interés en negociaciones dado que el bitcoin tiene un historial como activo cíclico como se ve en los mercados de energía esto se debe a que el bitcoin cada 4 años la cantidad de moneda generadas se reduce a la mitad.

```
dfv = df.loc[df.index[0:],['Date','Volume']]
# dfv
plt.figure(figsize=(14, 8))
plt.scatter(x=dfv['Date'], y=dfv['Volume'] , color='blue')
plt.xlabel("Fechas", size=16)
# plt.ylim([0, 5])
plt.ylabel("Volumen de negociación", size=15)
# plt.show()
# plt.title("Bubble Plot", size=18)
```



Run the app to see the outputs

Press the run button in the top right corner



Run the app to see this chart

Press the run button in the top right corner