

Tarea por calificar

Pregunta 1: Utilice yfinance para extraer datos bursátiles

Restablezca el índice, guarde y visualice las cinco primeras filas del marco de datos tesla_data utilizando la función head. Suba una captura de pantalla de los resultados y el código del principio de la Pregunta 1 a los resultados de abajo.


```
# 1. Instalar la librería yfinance
!pip install lxml==4.6.4
# !pip install requests==2.26.0

# 2. Importar las librerías necesarias
import yfinance as yf

# 3. Extraer los datos bursátiles de Tesla (TSLA)
tesla_data = yf.download('TSLA', start='2020-01-01', end='2024-12-01')

# 4. Restablecer el índice
tesla_data.reset_index(inplace=True)

# 5. Visualizar las primeras 5 filas
tesla_data.head()
```

 Requirement already satisfied: lxml==4.6.4 in /usr/local/lib/python3.10/dist-packages (4.6.4)
[*****100%*****] 1 of 1 completed

Price	Date	Adj Close	Close	High	Low	Open	Volume	
Ticker		TSLA	TSLA	TSLA	TSLA	TSLA	TSLA	
0	2020-01-02	28.684000	28.684000	28.713333	28.114000	28.299999	142981500	
1	2020-01-03	29.534000	29.534000	30.266666	29.128000	29.366667	266677500	
2	2020-01-06	30.102667	30.102667	30.104000	29.333332	29.364668	151995000	
3	2020-01-07	31.270666	31.270666	31.441999	30.224001	30.760000	268231500	
4	2020-01-08	32.809334	32.809334	33.232666	31.215334	31.580000	467164500	

Pasos siguientes:

[Ver gráficos recomendados](#)[New interactive sheet](#)

Pregunta 2

```
# 1. Instalar las librerías necesarias
!pip install requests beautifulsoup4 pandas

# 2. Importar las librerías
import requests
from bs4 import BeautifulSoup
import pandas as pd

# 3. Definir la URL de la página con los datos financieros de Tesla
url = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/revenue.htm'

# 4. Realizar la solicitud HTTP a la página
response = requests.get(url)

# 5. Parsear el contenido HTML con BeautifulSoup
soup = BeautifulSoup(response.content, 'html.parser')

# 6. Encontrar la tabla de ingresos de Tesla
table = soup.find_all('table')[0]

# 7. Crear una lista para almacenar las filas de datos
rows = []

# 8. Iterar sobre las filas de la tabla para extraer los datos
for row in table.find_all('tr')[1:]: # Omitir la primera fila de encabezado
    cols = row.find_all('td')
    cols = [ele.text.strip() for ele in cols]
    rows.append(cols)

# 9. Crear un DataFrame a partir de los datos extraídos
tesla_revenue = pd.DataFrame(rows, columns=['Year', 'Revenue'])

# 10. Mostrar las últimas 5 filas del DataFrame usando tail()
tesla_revenue.tail()
```

 Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (2.32.3)
 Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-packages (4.12.3)
 Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (2.2.2)
 Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests) (3.4.0)
 Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests) (3.10)
 Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests) (2.2.3)
 Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests) (2024.8.30)
 Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (from beautifulsoup4) (2.6)
 Requirement already satisfied: numpy>=1.22.4 in /usr/local/lib/python3.10/dist-packages (from pandas) (1.26.4)
 Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
 Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
 Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
 Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)

	Year	Revenue	
8	2013	\$2,013	
9	2012	\$413	
10	2011	\$204	
11	2010	\$117	
12	2009	\$112	

Pregunta 3

```

# Código Pregunta 1
!pip install lxml==4.6.4
# !pip install requests==2.26.0

# Código Pregunta 1
import yfinance as yf



# Desarrollo Pregunta 2
# 1. Extraer los datos bursátiles de GameStop (GME)
gme_data = yf.download('GME', start='2020-01-01', end='2024-12-01')

# 2. Restablecer el índice
gme_data.reset_index(inplace=True)

# 3. Visualizar las primeras 5 filas
gme_data.head()

```

↻ [*****100%*****] 1 of 1 completed

Price	Date	Adj Close	Close	High	Low	Open	Volume	
Ticker		GME	GME	GME	GME	GME	GME	
0	2020-01-02	1.5775	1.5775	1.6175	1.5175	1.5350	17814400	
1	2020-01-03	1.4700	1.4700	1.5625	1.4600	1.5525	14175600	
2	2020-01-06	1.4625	1.4625	1.4775	1.4000	1.4500	13579200	
3	2020-01-07	1.3800	1.3800	1.4575	1.3600	1.4425	20912000	
4	2020-01-08	1.4300	1.4300	1.4625	1.3525	1.3725	22517600	

Pasos siguientes:

 [Ver gráficos recomendados](#)

[New interactive sheet](#)

Pregunta 4

```
# 1. Definir la URL de la página con los datos financieros de GameStop (GME)
url = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/stock.html'

# 2. Realizar la solicitud HTTP a la página
response = requests.get(url)

# 3. Parsear el contenido HTML con BeautifulSoup
soup = BeautifulSoup(response.content, 'html.parser')

# 4. Encontrar la tabla de ingresos de GameStop
table = soup.find('table')

# 5. Crear una lista para almacenar las filas de datos
rows = []

# 6. Iterar sobre las filas de la tabla para extraer los datos
for row in table.find_all('tr')[1:]: # Omitir la primera fila de encabezado
    cols = row.find_all('td')
    cols = [ele.text.strip() for ele in cols]
    rows.append(cols)

# 7. Crear un DataFrame a partir de los datos extraídos
gme_revenue = pd.DataFrame(rows, columns=['Year', 'Revenue'])

# 8. Mostrar las últimas 5 filas del DataFrame usando tail()
gme_revenue.tail()
```



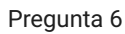
	Year	Revenue	
11	2009	\$8,806	
12	2008	\$7,094	
13	2007	\$5,319	
14	2006	\$3,092	
15	2005	\$1,843	

Pregunta 5

```
# 1. Importar las librerías necesarias
import matplotlib.pyplot as plt
```

```
# 2. Crear la función make_graph
def make_graph(data):
    # Crear una gráfica de la columna "Close" (precio de cierre)
    plt.figure(figsize=(10, 6))
    plt.plot(data['Close'], label='Precio de Cierre')
    plt.title('Precio de las Acciones de Tesla (TSLA)', fontsize=16)
    plt.xlabel('Fecha')
    plt.ylabel('Precio de Cierre (USD)')
    plt.legend(loc='upper left')
    plt.grid(True)
    plt.show()
```

```
# 3. Llamar a la función make_graph con los datos de Tesla
make_graph(tesla_data)
```



The chart displays the closing price of GameStop (GME) stock over a two-year period. The y-axis represents the closing price in USD, with major gridlines at 40, 60, and 80. The x-axis shows the timeline from January 2020 to January 2022. A legend indicates that the green line represents the 'Precio de Cierre' (Closing Price). The price starts around \$40 in early 2020, rises to a peak of approximately \$85 in early 2021, then drops sharply to around \$40 by mid-2021. It remains relatively stable between \$40 and \$50 until late 2021, when it spikes again to nearly \$50 before ending the period around \$45.