Task 1 Done

December 5, 2025

[8]:

```
from bs4 import BeautifulSoup
import requests
import pandas as pd

url = "https://en.wikipedia.org/wiki/
 ↪List_of_public_corporations_by_market_capitalization"
headers = {
    "User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) "
                  "AppleWebKit/537.36 (KHTML, like Gecko) "
                  "Chrome/120.0.0.0 Safari/537.36"
}

response = requests.get(url, headers=headers)
soup = BeautifulSoup(response.text, "lxml")

tables = soup.find_all("table", class_="wikitable")
print("Number of wikitable tables found:", len(tables))

Market_Value = tables[1]
row_s = Market_Value.find_all("tr")
data = []

for row in row_s[1:]:
    cols = row.find_all(["th", "td"])
    cols = [c.get_text(strip=True) for c in cols]
    data.append(cols)

df = pd.DataFrame(data, columns=[
    "Rank", "First quarter", "Second quarter", "Third quarter",
    "Fourth quarter", "", "", "", ""
])

df
```

```
Number of wikitable tables found: 27
```

[8]:

```
  Rank First quarter                      Second quarter Third quarter  \
0    1                                 Apple3,337,000[43]
```

```
1    2                         Microsoft2,791,000[45]
2    3                            Nvidia2,644,000[44]
3    4                            Amazon2,016,000[46]
4    5                          Alphabet1,895,000[47]
5    6                              Meta1,460,000[48]
6    7                Berkshire Hathaway1,140,000[49]
7    8                               Tesla833,529[51]
8    9                            Broadcom787,247[50]
9   10                           Eli Lilly782,950[53]

                    Fourth quarter
0              Nvidia3,850,000[44]                Nvidia4,542,000[44]
1           Microsoft3,700,000[45]             Microsoft3,850,000[45]
2               Apple3,060,000[43]                 Apple3,794,000[43]
3              Amazon2,330,000[46]              Alphabet2,975,000[47]
4            Alphabet2,150,000[47]                Amazon2,341,000[46]
5                Meta1,860,000[48]                  Meta1,845,000[48]
6            Broadcom1,300,000[50]              Broadcom1,589,000[50]
7                TSMC1,170,000[52]                 Tesla1,478,000[51]
8  Berkshire Hathaway1,050,000[49]                  TSMC1,448,000[52]
9               Tesla1,020,000[51]    Berkshire Hathaway1,086,000[49]
```

[9]:
```
url = "https://en.wikipedia.org/wiki/
 ↪List_of_public_corporations_by_market_capitalization"
headers = {
    "User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) "
                  "AppleWebKit/537.36 (KHTML, like Gecko) "
                  "Chrome/120.0.0.0 Safari/537.36"
}

response = requests.get(url, headers=headers)
soup = BeautifulSoup(response.text, "lxml")

tables = soup.find_all("table", class_="wikitable")
print("Number of wikitable tables found:", len(tables))

Market_Value = tables[2]
row_s = Market_Value.find_all("tr")
data = []

for row in row_s[1:]:
    cols = row.find_all(["th", "td"])
    cols = [c.get_text(strip=True) for c in cols]
    data.append(cols)

df = pd.DataFrame(data, columns=[
    "Rank", "First quarter", "Second quarter", "Third quarter",
    "Fourth quarter", "", "", "", ""
])

df
```

```
Number of wikitable tables found: 27
```

[9]:

```
  Rank First quarter                 Second quarter Third quarter  \
0    1                       Microsoft3,126,000[45]
1    2                           Apple2,648,000[43]
2    3                          Nvidia2,259,000[44]
3    4                        Alphabet1,893,000[47]
4    5                          Amazon1,874,000[46]
5    6                            Meta1,238,000[48]
6    7                Berkshire Hathaway912,130[49]
7    8                         Eli Lilly739,660[53]
8    9                              TSMC705,690[52]
9   10                          Broadcom614,220[50]

                  Fourth quarter                                     \
0             Apple3,322,000[45]           Microsoft3,543,000[43]
1         Microsoft3,230,000[43]               Apple3,198,000[45]
2            Nvidia3,182,000[44]              Nvidia2,979,000[44]
3          Alphabet2,267,000[47]            Alphabet2,058,000[47]
4            Amazon2,011,000[46]              Amazon1,956,000[46]
5              Meta1,279,000[48]                Meta1,448,000[48]
6                TSMC901,390[52]    Berkshire Hathaway993,020[49]
7  Berkshire Hathaway879,670[49]                  TSMC900,670[52]
8           Eli Lilly815,210[53]                 Tesla835,810[51]
9            Broadcom747,360[50]              Broadcom805,670[50]


0             Apple3,785,000[43]
1            Nvidia3,289,000[44]
2         Microsoft3,134,000[45]
3          Alphabet2,331,000[47]
4            Amazon2,307,000[46]
5              Meta1,478,000[48]
6             Tesla1,296,000[51]
7          Broadcom1,087,000[50]
8              TSMC1,024,000[52]
9  Berkshire Hathaway978,890[49]
```

[10]:

```
url = "https://en.wikipedia.org/wiki/
 ↪List_of_public_corporations_by_market_capitalization"
headers = {
    "User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) "
                  "AppleWebKit/537.36 (KHTML, like Gecko) "
                  "Chrome/120.0.0.0 Safari/537.36"
}

response = requests.get(url, headers=headers)
soup = BeautifulSoup(response.text, "lxml")

tables = soup.find_all("table", class_="wikitable")
print("Number of wikitable tables found:", len(tables))

Market_Value = tables[3]
row_s = Market_Value.find_all("tr")
data = []

for row in row_s[1:]:
    cols = row.find_all(["th", "td"])
    cols = [c.get_text(strip=True) for c in cols]
    data.append(cols)

df = pd.DataFrame(data, columns=[
    "Rank", "First quarter", "Second quarter", "Third quarter",
    "Fourth quarter", "", "", "", ""
])

df
```

```
Number of wikitable tables found: 27
```

[10]:

```
  Rank First quarter                 Second quarter Third quarter  \
0    1                           Apple2,609,000[43]
1    2                       Microsoft2,146,000[45]
2    3                        Alphabet1,332,000[47]
3    4                          Amazon1,058,000[46]
4    5                            Nvidia686,090[44]
5    6                Berkshire Hathaway677,770[49]
6    7                             Tesla656,420[51]
7    8                              Meta549,480[54]
8    9                              TSMC482,410[52]
9   10                              Visa473,870[55]

                  Fourth quarter                                     \
0             Apple3,050,000[43]               Apple2,677,000[43]
1         Microsoft2,532,000[45]           Microsoft2,346,000[45]
2          Alphabet1,530,000[47]            Alphabet1,662,000[47]
3            Amazon1,337,000[46]              Amazon1,312,000[46]
4            Nvidia1,044,000[44]              Nvidia1,074,000[44]
5               Tesla829,670[51]                 Tesla794,200[51]
6  Berkshire Hathaway745,010[49]                  Meta772,490[48]
```

```
7                 Meta735,450[54]    Berkshire Hathaway769,260[49]
8                 TSMC523,410[52]             Eli Lilly509,890[53]
9                 Visa497,370[55]                  Visa480,990[55]
```

```
0              Apple2,994,000[43]
1          Microsoft2,795,000[45]
2           Alphabet1,764,000[47]
3             Amazon1,570,000[46]
4             Nvidia1,223,000[44]
5                 Meta909,000[48]
6                Tesla789,930[51]
7  Berkshire Hathaway783,550[49]
8            Eli Lilly553,370[53]
9                 TSMC539,390[52]
```

[11]:

```
url = "https://en.wikipedia.org/wiki/
 ↪List_of_public_corporations_by_market_capitalization"
headers = {
    "User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) "
                  "AppleWebKit/537.36 (KHTML, like Gecko) "
                  "Chrome/120.0.0.0 Safari/537.36"
}

response = requests.get(url, headers=headers)
soup = BeautifulSoup(response.text, "lxml")

tables = soup.find_all("table", class_="wikitable")
print("Number of wikitable tables found:", len(tables))

Market_Value = tables[4]
row_s = Market_Value.find_all("tr")
data = []

for row in row_s[1:]:
    cols = row.find_all(["th", "td"])
    cols = [c.get_text(strip=True) for c in cols]
    data.append(cols)

df = pd.DataFrame(data, columns=[
    "Rank", "First quarter", "Second quarter", "Third quarter",
    "Fourth quarter", "", "", "", ""
])

df
```

```
Number of wikitable tables found: 27
```

[11]:

```
  Rank First quarter                 Second quarter Third quarter  \
0    1                           Apple2,850,000[43]
1    2                       Microsoft2,311,000[45]
2    3                        Alphabet1,846,000[47]
3    4                          Amazon1,659,000[46]
4    5                           Tesla1,114,000[51]
5    6                Berkshire Hathaway779,150[49]
6    7                            Nvidia684,880[44]
7    8                              Meta605,250[54]
8    9                              TSMC540,670[52]
9   10                      UnitedHealth479,830[56]

                  Fourth quarter                                      \
0             Apple2,212,000[43]               Apple2,221,000[43]
1         Microsoft1,920,000[45]           Microsoft1,737,000[45]
2          Alphabet1,435,000[47]            Alphabet1,254,000[47]
3            Amazon1,080,000[46]              Amazon1,151,000[46]
4               Tesla697,660[51]                 Tesla831,150[51]
5  Berkshire Hathaway602,450[49]    Berkshire Hathaway596,410[49]
6        UnitedHealth481,870[56]          UnitedHealth472,410[56]
7   Johnson & Johnson467,090[57]     Johnson & Johnson429,500[57]
8             Tencent445,990[59]                  Visa374,380[55]
9                Meta436,390[54]                  Meta364,650[60]


0             Apple2,066,000[43]
1         Microsoft1,787,000[45]
2          Alphabet1,145,000[47]
3             Amazon856,940[46]
4  Berkshire Hathaway681,770[49]
5        UnitedHealth495,370[56]
6   Johnson & Johnson461,840[57]
7           ExxonMobil454,240[58]
8                Visa439,950[55]
9             Tencent405,090[59]
```

[12]:

```python
import numpy as np
N_A = np.nan
#Fixing the data to my liking
#APPLE
Apple_MV = np.array([2850000, 2212000, 2221000, 2066000, 2609000, 3050000,␣
 ↪2677000, 2994000, 2648000, 3322000, 3198000, 3785000, 3337000, 3060000,␣
 ↪3794000])
print(f'Apple Market Value:', Apple_MV)

#Microsoft
```

```
Microsoft_MV = np.array([2311000, 1920000, 1737000, 1787000, 2146000, 2532000,␣
 ↪2346000, 2795000, 3126000, 3230000, 3543000, 3134000, 2791000, 3700000,␣
 ↪3850000])
print(f'Microsoft Market Value:', Microsoft_MV)

#Alphabet
Alphabet_MV = np.array([1846000, 1435000, 1254000, 1145000, 1332000, 1530000,␣
 ↪1662000, 1764000, 1893000, 2267000, 2058000, 2331000, 1895000, 2150000,␣
 ↪2975000])
print(f'Alphabet Market Value:', Alphabet_MV)

#Amazon
Amazon_MV = np.array([1659000, 1080000, 1151000, 856940, 1058000, 1337000,␣
 ↪1312000, 1570000, 1874000, 2011000, 1956000, 2307000, 2016000, 2330000,␣
 ↪2341000])
print(f'Amazon Market Value:', Amazon_MV)

#Tesla
Tesla_MV = np.array([1114000, 697660, 831150, N_A, 656420, 829670, 794200,␣
 ↪789930, N_A, N_A, 835810, 1296000, 833529, 1020000, 1478000])
print(f'Tesla Market Value:', Tesla_MV)

#Berkshire Hathaway
Berkshire_Hathaway_MV = np.array([779150, 602450, 596410, 681770, 677770,␣
 ↪745010, 769260, 783550, 912130, 879670, 993020, 978890, 1140000, 1050000,␣
 ↪1086000])
print(f'Berkshire Hathaway Market Value:', Berkshire_Hathaway_MV)

#Nvidia
Nvidia_MV = np.array([684880, N_A, N_A, N_A, 686090, 1044000, 1074000, 1223000,␣
 ↪2259000, 3182000, 2979000, 3289000, 2644000, 3850000, 4542000])
print(f'Nvidia Market Value:', Nvidia_MV)

#Meta
Meta_MV = np.array([605250, 436390, 364650, N_A, 549480, 735450, 772490,␣
 ↪909000, 1238000, 1279000, 1448000, 1478000, 1460000, 1860000, 1845000])
print(f'Meta_Market Value:', Meta_MV)

#TSMC
TSMC_MV = np.array([540670, N_A, N_A, N_A, 482410, 523410, N_A, 539390, 705690,␣
 ↪901390, 900670, 1024000, N_A, 1170000, 1448000])
print(f'TSMC Market Value:', TSMC_MV)

#UnitedHealth
UnitedHealth_MV = np.array([479830, 481870, 472410, 495370, N_A, N_A, N_A, N_A,␣
 ↪N_A, N_A, N_A, N_A, N_A, N_A, N_A])
```

```
print(f'UnitedHealth Market Value:', UnitedHealth_MV)

#Johnson & Johnson
Johnson_and_Johnson_MV= np.array([N_A, 467090, 429500, 461840, N_A, N_A, N_A,␣
 ↪N_A, N_A, N_A, N_A, N_A, N_A, N_A, N_A])
print(f'TSMC Market Value:', TSMC_MV)

#Tencent
Tencent_MV = np.array([N_A, 445990, N_A, 405090, N_A, N_A, N_A, N_A, N_A, N_A,␣
 ↪N_A, N_A, N_A, N_A, N_A])
print(f'Johnson and Johnson Market Value:', Johnson_and_Johnson_MV)

#Visa
Visa_MV = np.array([N_A, N_A, 374380, 439950, 473870, 497370, 480990, N_A, N_A,␣
 ↪N_A, N_A, N_A, N_A, N_A, N_A])
print(f'Visa Market Value:', Visa_MV)

#ExxonMobil
ExxonMobil_MV = np.array([N_A, N_A, N_A, 454240, N_A, N_A, N_A, N_A, N_A, N_A,␣
 ↪N_A, N_A, N_A, N_A, N_A])
print(f'ExxonMobil Market Value:', ExxonMobil_MV)

#Eli Lilly
Eli_Lilly_MV = np.array([N_A, N_A, N_A, N_A, N_A, N_A, 509890, 553370, 739660,␣
 ↪815210, N_A, N_A, 782950, N_A, N_A])
print(f'Eli Lilly Market Value:', Eli_Lilly_MV)

#Broadcom
Broadcom_MV = np.array([N_A, N_A, N_A, N_A, N_A, N_A, N_A, N_A, 61422, 747360,␣
 ↪805670, 1087000, 787247, 1300000, 1589000])
print(f'Broadcom Market Value:', Broadcom_MV)


Time = np.array([0.25, 0.50, 0.75, 1.0, 1.25, 1.50, 1.75, 2.0, 2.25, 2.50, 2.
 ↪75, 3.0, 3.25, 3.50, 3.75])
```

```
Apple Market Value: [2850000 2212000 2221000 2066000 2609000 3050000 2677000
2994000 2648000
 3322000 3198000 3785000 3337000 3060000 3794000]
Microsoft Market Value: [2311000 1920000 1737000 1787000 2146000 2532000 2346000
2795000 3126000
 3230000 3543000 3134000 2791000 3700000 3850000]
Alphabet Market Value: [1846000 1435000 1254000 1145000 1332000 1530000 1662000
1764000 1893000
 2267000 2058000 2331000 1895000 2150000 2975000]
Amazon Market Value: [1659000 1080000 1151000  856940 1058000 1337000 1312000
1570000 1874000
```

```
 2011000 1956000 2307000 2016000 2330000 2341000]
Tesla Market Value: [1114000.  697660.  831150.      nan  656420.  829670.
794200.  789930.
      nan      nan  835810. 1296000.  833529. 1020000. 1478000.]
Berkshire Hathaway Market Value: [ 779150  602450  596410  681770  677770
745010  769260  783550  912130
  879670  993020  978890 1140000 1050000 1086000]
Nvidia Market Value: [ 684880.      nan      nan      nan  686090. 1044000.
1074000. 1223000.
 2259000. 3182000. 2979000. 3289000. 2644000. 3850000. 4542000.]
Meta_Market Value: [ 605250.  436390.  364650.      nan  549480.  735450.
772490.  909000.
 1238000. 1279000. 1448000. 1478000. 1460000. 1860000. 1845000.]
TSMC Market Value: [ 540670.      nan      nan      nan  482410.  523410.
nan  539390.
  705690.  901390.  900670. 1024000.      nan 1170000. 1448000.]
UnitedHealth Market Value: [479830. 481870. 472410. 495370.     nan     nan
nan     nan     nan
     nan     nan     nan     nan     nan     nan]
TSMC Market Value: [ 540670.      nan      nan      nan  482410.  523410.
nan  539390.
  705690.  901390.  900670. 1024000.      nan 1170000. 1448000.]
Johnson and Johnson Market Value: [    nan 467090. 429500. 461840.     nan
nan     nan     nan     nan
     nan     nan     nan     nan     nan     nan]
Visa Market Value: [    nan     nan 374380. 439950. 473870. 497370. 480990.
nan     nan
     nan     nan     nan     nan     nan     nan]
ExxonMobil Market Value: [    nan     nan     nan 454240.     nan     nan
nan     nan     nan
     nan     nan     nan     nan     nan     nan]
Eli Lilly Market Value: [    nan     nan     nan     nan     nan     nan 509890.
553370. 739660.
 815210.     nan     nan 782950.     nan     nan]
Broadcom Market Value: [     nan      nan      nan      nan      nan      nan
nan      nan
   61422.  747360.  805670. 1087000.  787247. 1300000. 1589000.]
```

[13]:

```
# Core AI Leaders
Nvidia_MV = np.array([684880, N_A, N_A, N_A, 686090, 1044000, 1074000, 1223000,␣
 ↪2259000, 3182000, 2979000, 3289000, 2644000, 3850000, 4542000])

Microsoft_MV = np.array([2311000, 1920000, 1737000, 1787000, 2146000, 2532000,␣
 ↪2346000, 2795000, 3126000, 3230000, 3543000, 3134000, 2791000, 3700000,␣
 ↪3850000])
```

```
Alphabet_MV = np.array([1846000, 1435000, 1254000, 1145000, 1332000, 1530000, 1662000, 1764000, 1893000, 2267000, 2058000, 2331000, 1895000, 2150000, 2975000])

Apple_MV = np.array([2850000, 2212000, 2221000, 2066000, 2609000, 3050000, 2677000, 2994000, 2648000, 3322000, 3198000, 3785000, 3337000, 3060000, 3794000])

Meta_MV = np.array([605250, 436390, 364650, N_A, 549480, 735450, 772490, 909000, 1238000, 1279000, 1448000, 1478000, 1460000, 1860000, 1845000])


# AI investors, and integrators and users
Amazon_MV = np.array([1659000, 1080000, 1151000, 856940, 1058000, 1337000, 1312000, 1570000, 1874000, 2011000, 1956000, 2307000, 2016000, 2330000, 2341000])

Tesla_MV = np.array([1114000, 697660, 831150, N_A, 656420, 829670, 794200, 789930, N_A, N_A, 835810, 1296000, 833529, 1020000, 1478000])

Broadcom_MV = np.array([N_A, N_A, N_A, N_A, N_A, N_A, N_A, N_A, 61422, 747360, 805670, 1087000, 787247, 1300000, 1589000])

TSMC_MV = np.array([540670, N_A, N_A, N_A, 482410, 523410, N_A, 539390, 705690, 901390, 900670, 1024000, N_A, 1170000, 1448000])

Tencent_MV = np.array([N_A, 445990, N_A, 405090, N_A, N_A, N_A, N_A, N_A, N_A, N_A, N_A, N_A, N_A, N_A])



# Limitted AI Integration
Berkshire_Hathaway_MV = np.array([779150, 602450, 596410, 681770, 677770, 745010, 769260, 783550, 912130, 879670, 993020, 978890, 1140000, 1050000, 1086000])

UnitedHealth_MV = np.array([479830, 481870, 472410, 495370, N_A, N_A, N_A, N_A, N_A, N_A, N_A, N_A, N_A, N_A, N_A])

Johnson_and_Johnson_MV= np.array([N_A, 467090, 429500, 461840, N_A, N_A, N_A, N_A, N_A, N_A, N_A, N_A, N_A, N_A, N_A])

Visa_MV = np.array([N_A, N_A, 374380, 439950, 473870, 497370, 480990, N_A, N_A, N_A, N_A, N_A, N_A, N_A, N_A])
```

```
ExxonMobil_MV = np.array([N_A, N_A, N_A, 454240, N_A, N_A, N_A, N_A, N_A, N_A,␣
 ↪N_A, N_A, N_A, N_A, N_A])

Eli_Lilly_MV = np.array([N_A, N_A, N_A, N_A, N_A, N_A, 509890, 553370, 739660,␣
 ↪815210, N_A, N_A, 782950, N_A, N_A])
```

[ ]:

```

```