Data Wrangling with Python · Unicorns

Date: 20 June 2024

Project completed by: Anton Starshev
linkedin.com/in/starshev

Context

In this project scenario I am financial data consultant, and an investor has tasked my team with identifying new business opportunities. To help them decide which future companies to invest in, I need to provide a list of current businesses valued at more than $1 billion. These are sometimes referred to as "unicorns." My client will use this information to learn about profitable businesses in general.

The investor has asked me to provide them with the following data:

1. Companies in the Hardware industry based in either Beijing, San Francisco or London.
2. Companies in the Artificial Intelligence industry based in London.
3. A list of the top 20 countries sorted by sum of company valuations in each country, excluding United States, China, India and United Kingdom.

The dataset includes a list of businesses and data points, such as the year they were founded, their industry, city, country and continent.

Imports and loads

Imported the relevant Python libraries and modules.

In [87]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import seaborn as sns
```

Loaded the dataset into a DataFrame.

The dataset provided is in the form of a csv file named `Unicorn_Companies.csv` and contains a subset of data on unicorn companies.

In [91]:
```
df_companies = pd.read_csv("Unicorn_Companies.csv")
```

Data exploration

Displayed the first 10 rows of the data to understand how the dataset is structured.

In [92]:
```
df_companies.head(10)
```

	Company	Valuation	Date	Industry	City	Country/Region	Continent	Year	Fundin

Out[92]:

			Joined					Founded	
0	Bytedance	$180B	4/7/17	Artificial intelligence	Beijing	China	Asia	2012	$8
1	SpaceX	$100B	12/1/12	Other	Hawthorne	United States	North America	2002	$7
2	SHEIN	$100B	7/3/18	E-commerce & direct-to-consumer	Shenzhen	China	Asia	2008	$2
3	Stripe	$95B	1/23/14	Fintech	San Francisco	United States	North America	2010	$2
4	Klarna	$46B	12/12/11	Fintech	Stockholm	Sweden	Europe	2005	$4
5	Canva	$40B	1/8/18	Internet software & services	Surry Hills	Australia	Oceania	2012	$572
6	Checkout.com	$40B	5/2/19	Fintech	London	United Kingdom	Europe	2012	$2
7	Instacart	$39B	12/30/14	Supply chain, logistics, & delivery	San Francisco	United States	North America	2012	$3
8	JUUL Labs	$38B	12/20/17	Consumer & retail	San Francisco	United States	North America	2015	$14
9	Databricks	$38B	2/5/19	Data management & analytics	San Francisco	United States	North America	2013	$3

Checked statistical properties of the dataset.

In [74]:
```
df_companies.describe()
```

Out[74]:

	Year Founded
count	1074.000000
mean	2012.895717
std	5.698573
min	1919.000000
25%	2011.000000
50%	2014.000000
75%	2016.000000
max	2021.000000

Checked the size of the dataset.

In [75]:
```
df_companies.shape
```

Out[75]:
```
(1074, 10)
```

Checked the data types of the columns.

In [76]:
```
df_companies.dtypes
```

Out[76]:
```
Company             object
Valuation           object
Date Joined         object
Industry            object
City                object
Country/Region      object
Continent           object
Year Founded         int64
Funding             object
Select Investors    object
dtype: object
```

Data pre-processing

In order to answer the investor's questions, some data preprocessing steps were required. The first step was to add a new column to the dataframe containing just the year each company became a unicorn company. This new column was named as Year Joined.

In [96]:
```
df_companies.insert(loc = 3, column = 'Year Joined', value =
                    pd.to_datetime(df_companies['Date Joined'],
                                   format = '%m/%d/%y').dt.year)

df_companies.head()
```

Out[96]:

	Company	Valuation	Date Joined	Year Joined	Industry	City	Country/Region	Continent	Year Founded	Fun
0	Bytedance	$180B	4/7/17	2017	Artificial intelligence	Beijing	China	Asia	2012	
1	SpaceX	$100B	12/1/12	2012	Other	Hawthorne	United States	North America	2002	
2	SHEIN	$100B	7/3/18	2018	E-commerce & direct-to-consumer	Shenzhen	China	Asia	2008	
3	Stripe	$95B	1/23/14	2014	Fintech	San Francisco	United States	North America	2010	
4	Klarna	$46B	12/12/11	2011	Fintech	Stockholm	Sweden	Europe	2005	

Since the data in the 'Valuation' column was a string that starts with a `$` and ends with a `B`, it had to be converted to a numeric datatype to allow calculations. I defined a function called `str_to_num()` that accepts an argument x (a string in the format of the values contained in the 'Valuation' column) and returns the integer of the number represented by the input string.

In [97]:
```
def str_to_num(x):
    x = x.strip('$B')
    x = int(x)
    return x
```

Next, I used this function to create a new column called 'valuation_num' that represents the 'Valuation' column as an integer value.

In [98]:
```
df_companies['valuation_num'] = df_companies['Valuation'].apply(str_to_num)

df_companies.head()
```

Out[98]:

	Company	Valuation	Date Joined	Year Joined	Industry	City	Country/Region	Continent	Year Founded	Fun
0	Bytedance	$180B	4/7/17	2017	Artificial intelligence	Beijing	China	Asia	2012	
1	SpaceX	$100B	12/1/12	2012	Other	Hawthorne	United States	North America	2002	
2	SHEIN	$100B	7/3/18	2018	E-commerce & direct-to-consumer	Shenzhen	China	Asia	2008	
3	Stripe	$95B	1/23/14	2014	Fintech	San Francisco	United States	North America	2010	
4	Klarna	$46B	12/12/11	2011	Fintech	Stockholm	Sweden	Europe	2005	

Checked if any values are missing.

In [100…
```
df_companies.isnull().sum()
```

Out[100]:
```
Company              0
Valuation            0
Date Joined          0
Year Joined          0
Industry             0
City                16
Country/Region       0
Continent            0
Year Founded         0
Funding              0
Select Investors     1
valuation_num        0
dtype: int64
```

Checked all rows with missing values to understand their nature.

In [103…
```
df_companies.loc[df_companies.isnull().any(axis = 1)]
```

Out[103]:

	Company	Valuation	Date Joined	Year Joined	Industry	City	Country/Region	Continent	Fo
12	FTX	$32B	7/20/21	2021	Fintech	NaN	Bahamas	North America	

170	HyalRoute	$4B	5/26/20	2020	Mobile & telecommunications	NaN	Singapore	Asia
242	Moglix	$3B	5/17/21	2021	E-commerce & direct-to-consumer	NaN	Singapore	Asia
251	Trax	$3B	7/22/19	2019	Artificial intelligence	NaN	Singapore	Asia
325	Amber Group	$3B	6/21/21	2021	Fintech	NaN	Hong Kong	Asia
382	Ninja Van	$2B	9/27/21	2021	Supply chain, logistics, & delivery	NaN	Singapore	Asia
541	Advance Intelligence Group	$2B	9/23/21	2021	Artificial intelligence	NaN	Singapore	Asia
629	LinkSure Network	$1B	1/1/15	2015	Mobile & telecommunications	Shanghai	China	Asia
811	Carousell	$1B	9/15/21	2021	E-commerce & direct-to-consumer	NaN	Singapore	Asia
848	Matrixport	$1B	6/1/21	2021	Fintech	NaN	Singapore	Asia
880	bolttech	$1B	7/1/21	2021	Fintech	NaN	Singapore	Asia
889	Carro	$1B	6/14/21	2021	E-commerce & direct-to-consumer	NaN	Singapore	Asia
893	Cider	$1B	9/2/21	2021	E-commerce & direct-to-consumer	NaN	Hong Kong	Asia
980	NIUM	$1B	7/13/21	2021	Fintech	NaN	Singapore	Asia

986	ONE	$1B	12/8/21	2021	Internet software & services	NaN	Singapore	Asia
994	PatSnap	$1B	3/16/21	2021	Internet software & services	NaN	Singapore	Asia
1061	WeLab	$1B	11/8/17	2017	Fintech	NaN	Hong Kong	Asia

It turned out that the most of the missing values are related to Singapore and Hong Kong. So I filled the missing «City» values with the same names as given in the 'Country/region' column.

In [104…
```
df_companies.loc[df_companies['Country/Region'] == 'Singapore',
                 'City'] = df_companies.loc[df_companies['Country/Region'] ==
                                            'Singapore', 'City'].fillna('Singapore')
df_companies.loc[df_companies['Country/Region'] == 'Hong Kong',
                 'City'] = df_companies.loc[df_companies['Country/Region'] ==
                                            'Hong Kong', 'City'].fillna('Hong Kong')
df_companies.loc[df_companies['Country/Region'] == 'Bahamas',
                 'City'] = df_companies.loc[df_companies['Country/Region'] ==
                                            'Bahamas', 'City'].fillna('Bahamas')

df_companies.isnull().sum()
```

Out[104]:
```
Company             0
Valuation           0
Date Joined         0
Year Joined         0
Industry            0
City                0
Country/Region      0
Continent           0
Year Founded        0
Funding             0
Select Investors    1
valuation_num       0
dtype: int64
```

The only remaining row with the missing value was erased from the dataset.

In [105…
```
df_companies.dropna(inplace = True, ignore_index = True)
df_companies.isnull().sum()
```

Out[105]:
```
Company             0
Valuation           0
Date Joined         0
Year Joined         0
Industry            0
City                0
Country/Region      0
Continent           0
Year Founded        0
Funding             0
```

```
Select Investors    0
valuation_num       0
dtype: int64
```

Providing the requested information

Using the conditional logic, I filtered the data only related to the Hardware and Artificial Intelligence industries, as well as located only in the given list of cities for each industry.

In [108…

```
df_companies[(
    (df_companies['Industry'] == 'Hardware') & (df_companies['City'].isin(
    ['Beijing', 'San Francisco', 'London']))
                                            ) | (
        (df_companies['Industry'] == 'Artificial intelligence') &
        (df_companies['City'] == 'London')
    )]
```

Out[108]:

	Company	Valuation	Date Joined	Year Joined	Industry	City	Country/Region	Continent	Yea Founde
36	Bitmain	$12B	7/6/18	2018	Hardware	Beijing	China	Asia	201
43	Global Switch	$11B	12/22/16	2016	Hardware	London	United Kingdom	Europe	199
147	Chipone	$5B	12/16/21	2021	Hardware	Beijing	China	Asia	200
844	Density	$1B	11/10/21	2021	Hardware	San Francisco	United States	North America	201
872	BenevolentAI	$1B	6/2/15	2015	Artificial intelligence	London	United Kingdom	Europe	201
922	Geek+	$1B	11/21/18	2018	Hardware	Beijing	China	Asia	201
1039	TERMINUS Technology	$1B	10/25/18	2018	Hardware	Beijing	China	Asia	201
1045	Tractable	$1B	6/16/21	2021	Artificial intelligence	London	United Kingdom	Europe	201

For each country I summed the valuations of all companies in that country, then sorted the results in

descending order by summed valuation. Assigned the results to a variable called 'national_valuations'.

In [109…
```
national_valuations = df_companies.groupby(
    'Country/Region')['valuation_num'].sum().sort_values(ascending = False).reset_index(

national_valuations.head(15)
```

Out[109]:

	Country/Region	valuation_num
0	United States	1933
1	China	695
2	India	196
3	United Kingdom	195
4	Germany	72
5	Sweden	63
6	Australia	56
7	France	55
8	Canada	49
9	South Korea	41
10	Israel	39
11	Brazil	37
12	Bahamas	32
13	Indonesia	28
14	Singapore	21

To meet the needs of the stakeholder, I removed the United States, China, India and the United Kingdom and reassigned the result to a variable called 'national_valuations_no_big4'.

In [110…
```
national_valuations_no_big4 = national_valuations[~national_valuations
                                                  ['Country/Region'].isin([
    'United States', 'China', 'India', 'United Kingdom'])]

national_valuations_no_big4.reset_index(inplace = True)
national_valuations_no_big4.head(10)
```

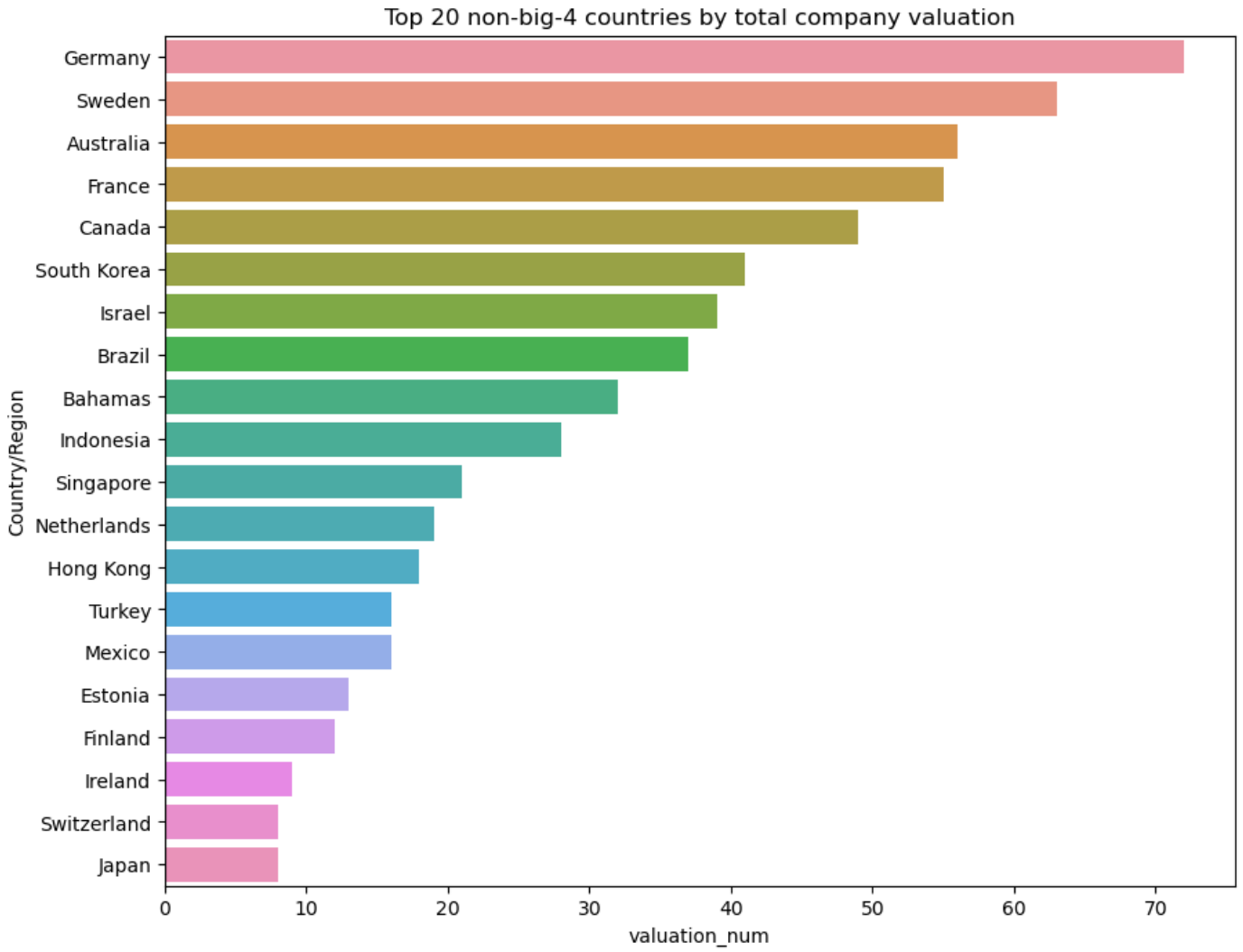
Out[110]:

	index	Country/Region	valuation_num
0	4	Germany	72
1	5	Sweden	63
2	6	Australia	56
3	7	France	55
4	8	Canada	49
5	9	South Korea	41
6	10	Israel	39
7	11	Brazil	37
8	12	Bahamas	32
9	13	Indonesia	28

Created barplot for top 20 non-big-4 countries

In [112…

```
plt.figure(figsize=(10,8))
sns.barplot(y = 'Country/Region', x = 'valuation_num',
            data = national_valuations_no_big4.head(20))
plt.title('Top 20 non-big-4 countries by total company valuation')
plt.show()
```



Summary of findings

- 8 companies met the stated criteria (in Hardware and Artificial Intelligence)
- The sorted data indicates that 4 countries with highest total company valuations are the United States, China, India and the United Kingdom, exactly those considered as outliers by the investor
- Valuation sum per country is visualized by the size of circles around the map, where Europe has the highest concentration of unicorn companies
- Top-5 among the non-big-4 countries with the highest company valuations are Germany, Sweden, Australia, France and Canada

Acknowledgment

This is a lab based on workplace scenario proposed within the syllabus of *Google Advanced Data Analytics Professional Certificate* on Coursera.

I would like to express gratitude to Google and Coursera for supporting the educational process and providing the opportunity to refine and showcase skills acquired during the courses by completing real-life scenario portfolio projects, such as this.

Dataset Reference

Bhat, M.A. *Unicorn Companies*