

```
In [156]. # Import library
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

In [162]. companies = pd.read_csv('C:\Users\VP\Desktop\Advance Data Analyst\3. Go beyond Numbers, turning data into insights\2. Module 2\2. what is missing\Files\Unicorn_Companies.csv')
companies.head(10)

Out[162].
Company Valuation Date Joined Industry City Country/Region Continent Year Founded Funding Select Investors
0 ByteDance $180B 4/7/17 Artificial Intelligence Beijing China Asia 2012 $8B Sequoia Capital China, SIG Asia Investments, S...
1 SpaceX $100B 12/1/12 Other Hawthorne United States North America 2002 $7B Founders Fund, Draper Fisher Jurvetson, Rothm...
2 SHEIN $100B 7/31/18 E-commerce & direct-to-consumer Shenzhen China Asia 2008 $2B Tiger Global Management, Sequoia Capital China...
3 Stripe $95B 1/23/14 Fintech San Francisco United States North America 2010 $2B Khosla Ventures, LowercaseCapital, capitalG...
4 Klarna $46B 12/12/11 Fintech Stockholm Sweden Europe 2005 $4B Institutional Venture Partners, Sequoia Capita...
5 Canva $40B 1/8/18 Internet software & services Surry Hills Australia Oceania 2012 $572M Sequoia Capital China, Blackbird Ventures, Ma...
6 Checkout.com $40B 5/2/19 Fintech London United Kingdom Europe 2012 $2B Tiger Global Management, Insight Partners, DST...
7 Instacart $39B 12/30/14 Supply chain, logistics, & delivery San Francisco United States North America 2012 $3B Khosla Ventures, Kleiner Perkins Caufield & By...
8 JULI Labs $38B 12/20/17 Consumer & retail San Francisco North America 2015 $14B Tiger Global Management
9 Databricks $38B 2/5/19 Data management & analytics San Francisco United States North America 2013 $3B Andreessen Horowitz, New Enterprise Associates...

In [164]. # Identify the number of rows and columns in the dataset.
companies.shape

Out[164].
(1074, 10)

In [166]. # Check for duplicates.
companies.drop_duplicates().shape

Out[166].
(1074, 10)

In [168]. # Display the data types of the columns.
companies.dtypes

Out[168].
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

In [170]. # Sort 'companies' and display the first 10 rows of the resulting DataFrame.
# To arrange the data from latest to earliest 'Year Founded'
companies.sort_values(by="Year Founded", ascending=False).head(10)

Out[170].
Company Valuation Date Joined Industry City Country/Region Continent Year Founded Funding Select Investors
782 Phantom $18B 1/13/22 Fintech San Francisco United States North America 2021 $110M Paradigm, Andreessen Horowitz, Jump Capital
814 Yidian Ziun $1B 10/17/17 Mobile & telecommunications Beijing China Asia 2021 $151M Phoenix New Media, Tianjin Hahe Industry Fund
722 GlobalBees $1B 12/28/21 E-commerce & direct-to-consumer New Delhi India Asia 2021 $185M Chirata Ventures, SoftBank Group, Tyfeda Ca...
554 Clickhouse $2B 10/28/21 Data management & analytics Portola Valley United States North America 2021 $300M Lightspeed Venture Partners, Almaz Capital Par...
952 LayerZero Labs $1B 3/30/22 Internet software & services New York United States North America 2021 $143M Andreessen Horowitz, FTX Ventures, Tiger Globa...
314 Flink Food $3B 12/1/21 E-commerce & direct-to-consumer Berlin Germany Europe 2021 $1B Mubadala Capital, Bond, Prosus Ventures
864 Aptos $1B 3/15/22 Internet software & services Palo Alto United States North America 2021 $200M Andreessen Horowitz, Coinbase Ventures, Tiger ...
238 Yuga Labs $4B 3/22/22 Fintech Miami United States North America 2021 $450M Andreessen Horowitz, Thrive Capital, Sound Ven...
775 Joker $1B 12/2/21 E-commerce & direct-to-consumer New York United States North America 2021 $430M GGV Capital, Tiger Global Management, Greycoft
967 Mensa Brands $1B 1/1/2021 Other Bengaluru India Asia 2021 $218M Accel, Falcon Edge Capital, Norwest Venture Pa...

In [172]. # Display each unique year that occurs in the dataset
# along with the number of companies that were founded in each unique year.
companies["Year Founded"].value_counts().sort_values(ascending=False)

Out[172].
Year Founded
2015 155
2016 110
2014 109
2012 85
2013 87
2011 62
2017 74
2018 61
2019 45
2010 40
2019 34
2008 27
2009 25
2007 24
2006 15
2005 14
2000 11
2021 11
2001 9
1999 8
2004 8
2003 8
1998 5
1919 1
1997 1
Name: count, dtype: int64

In [174]. # Plot a histogram
sns.histplot(data=companies, x="Year Founded")
plt.title("Year Founded histogram")

Out[174].
Year Founded histogram
Count
160
140
120
100
80
60
40
20
0
1920 1940 1960 1980 2000 2020
Year Founded

In [180]. # Convert 'Date Joined' column to datetime.
# Update the column with the converted values.
companies["Date Joined"] = pd.to_datetime(companies["Date Joined"])
# Display the data types of the columns in 'companies'
# To confirm that the update actually took place
companies.dtypes

Out[180].
Company object
Valuation object
Date Joined datetime64[ns]
Industry object
City object
Country/Region object
Continent object
Year Founded int64
Funding object
Select Investors object
dtype: object

In [182]. # Obtain the names of the months when companies gained unicorn status.
# Use the result to create a 'Month Joined' column.
companies["Month Joined"] = companies["Date Joined"].dt.month_name()
# Display the first few rows of 'companies'
# To confirm that the new column did get added.
companies.head()

Out[182].
Company Valuation Date Joined Industry City Country/Region Continent Year Founded Funding Select Investors Month Joined
0 ByteDance $180B 2017-04-07 Artificial Intelligence Beijing China Asia 2012 $8B Sequoia Capital China, SIG Asia Investments, S... April
1 SpaceX $100B 2012-12-01 Other Hawthorne United States North America 2002 $7B Founders Fund, Draper Fisher Jurvetson, Rothm... December
2 SHEIN $100B 2018-07-03 E-commerce & direct-to-consumer Shenzhen China Asia 2008 $2B Tiger Global Management, Sequoia Capital China... July
3 Stripe $95B 2014-01-23 Fintech San Francisco United States North America 2010 $2B Khosla Ventures, LowercaseCapital, capitalG... January
4 Klarna $46B 2011-12-12 Fintech Stockholm Sweden Europe 2005 $4B Institutional Venture Partners, Sequoia Capita... December

In [184]. # Determine how many years it took for companies to reach unicorn status.
# Use the result to create a 'Years To Join' column.
companies["Years To Join"] = companies["Date Joined"].dt.year - companies["Year Founded"]
# Display the first few rows of 'companies'
# To confirm that the new column did get added.
companies.head()

Out[184].
Company Valuation Date Joined Industry City Country/Region Continent Year Founded Funding Select Investors Month Joined Years To Join
0 ByteDance $180B 2017-04-07 Artificial Intelligence Beijing China Asia 2012 $8B Sequoia Capital China, SIG Asia Investments, S... April 5
1 SpaceX $100B 2012-12-01 Other Hawthorne United States North America 2002 $7B Founders Fund, Draper Fisher Jurvetson, Rothm... December 10
2 SHEIN $100B 2018-07-03 E-commerce & direct-to-consumer Shenzhen China Asia 2008 $2B Tiger Global Management, Sequoia Capital China... July 10
3 Stripe $95B 2014-01-23 Fintech San Francisco United States North America 2010 $2B Khosla Ventures, LowercaseCapital, capitalG... January 4
4 Klarna $46B 2011-12-12 Fintech Stockholm Sweden Europe 2005 $4B Institutional Venture Partners, Sequoia Capita... December 6

In [186]. # Filter dataset by a year of your interest (in terms of when companies reached unicorn status).
# Save the resulting subset in a new variable.
companies_2021 = companies[companies["Date Joined"].dt.year == 2021]
# Display the first few rows of the subset to confirm that it was created.
companies_2021.head()

Out[186].
Company Valuation Date Joined Industry City Country/Region Continent Year Founded Funding Select Investors Month Joined Years To Join
12 FTX $32B 2021-07-20 Fintech NaN Bahamas North America 2018 $2B Sequoia Capital, Thoma Bravo, Softbank April 3
16 J&T Express $20B 2021-04-07 Supply chain, logistics, & delivery Jakarta Indonesia Asia 2015 $5B Hilhouse Capital Management, Boyu Capital, Se... July 6
24 Blockchain.com $14B 2021-02-17 Fintech London United Kingdom Europe 2011 $490M Lightspeed Venture Partners, Google Ventures, ... February 10
27 OpenSea $13B 2021-07-20 E-commerce & direct-to-consumer New York United States North America 2017 $427M Andreessen Horowitz, Thirty Five Ventures, Sou... July 4
34 Getir $12B 2021-03-26 E-commerce & direct-to-consumer Istanbul Turkey Europe 2015 $2B Tiger Global Management, Sequoia Capital, Revo... March 6

In [188]. # After identifying the time interval that interests you, proceed with the following:
# Step 1. Take the subset that you defined for the year of interest.
# Insert a column that contains the time interval that each date point belongs to, as needed.
# Step 2. Group by the time interval.
# Aggregate by counting companies that joined per interval of that year.
# Save the resulting DataFrame in a new variable.
# Insert a 'Week Joined' column into 'companies_2021'.
companies_2021.insert(3, "Week Joined", companies_2021["Date Joined"].dt.strftime("%W-%V"), True)
# Group 'companies_2021' by 'Week Joined'.
# Aggregate by counting companies that joined per week of 2021.
# Save the resulting DataFrame in a new variable.
companies_by_week_2021 = companies_2021.groupby(by="Week Joined")["Company"].count().reset_index().rename(columns={"Company": "Company Count"})
# Display the first few rows of the new DataFrame to confirm that it was created.
companies_by_week_2021.head()

Out[188].
Week Joined Company Count
0 2021-W01 12
1 2021-W02 9
2 2021-W03 5
3 2021-W04 8
4 2021-W05 4

In [190]. # After identifying the additional year and time interval of interest, proceed with the following:
# Step 1. Filter by the additional year to create a subset that consists of companies that joined in that year.
# Step 2. Concatenate that new subset with the subset that you defined previously.
# Step 3. As needed, add a column that contains the time interval that each date point belongs to,
# in the concatenated DataFrame.
# Step 4. Transform the 'Valuation' column as needed.
# Step 5. Group by the time interval.
# Aggregate by computing average valuation of companies that joined per interval of the corresponding year.
# Save the resulting DataFrame in a new variable.
# Filter by the additional year to create a subset that consists of companies that joined in that year.
companies_2020 = companies[companies["Date Joined"].dt.year == 2020]
# Concatenate the new subset with the subset that you defined previously.
companies_2020_2021 = pd.concat([companies_2020, companies_2021.drop(columns="Week Joined")])
# Add 'Quarter Joined' column to 'companies_2021'.
companies_2020_2021["Quarter Joined"] = companies_2020_2021["Date Joined"].dt.to_period("Q").dt.strftime("%Y-%Qq")
# Convert the 'Valuation' column to numeric by removing '$' and 'B' and casting each value to data type 'float'.
companies_2020_2021["Valuation"] = companies_2020_2021["Valuation"].str.strip("$B").astype(float)
# Group 'companies_2020_2021' by 'Quarter Joined'.
# Aggregate by computing average 'Funding' of companies that joined per quarter of each year.
# Save the resulting DataFrame in a new variable.
companies_by_quarter_2020_2021 = companies_2020_2021.groupby(by="Quarter Joined")["Valuation"].mean().reset_index().rename(columns={"Valuation": "Average Valuation"})
# Display the first few rows of the new DataFrame to confirm that it was created.
companies_by_quarter_2020_2021.head()

Out[190].
Quarter Joined Average Valuation
0 2020-Q1 3.444444
1 2020-Q2 3.777778
2 2020-Q3 3.896552
3 2020-Q4 3.697674
4 2021-Q1 2.760000

In [192]. # Define a list that contains months in chronological order.
month_order = ("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December")
# Print out the list to confirm it is correct.
print(month_order)
('January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December')

In [194]. # Create the box plot to visualize the distribution of how long it took companies to become unicorns, with respect to the month they joined.
# The box plot created shows that companies that became unicorns in the months of September and October have a smaller median value for how long it took to become unicorns.
sns.boxplot(x=companies["Month Joined"],
            y=companies["Years To Join"],
            order=month_order,
            showfliers=False)
# Set the title of the plot.
plt.title("Distribution of years to become unicorn with respect to month joined")
# Rotate labels on the x-axis as a way to avoid overlap in the positions of the text.
plt.xticks(rotation=45, horizontalalignment="right")
# Display the plot.
plt.show()

Out[194].
Distribution of years to become unicorn with respect to month joined
Years to join
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
January February March April May June July August September October November December
Month joined

In [200]. # Set the size of the plot.
plt.figure(figsize=(10, 5))
# Create bar plot to visualize the average number of years it took companies to reach unicorn status
# with respect to when they were founded.
# Use the data from the 'companies' DataFrame.
sns.barplot(x=companies["Year Founded"], y=companies["Years To Join"], errorbar="ci", palette="m")
# Set title
plt.title("Bar plot of years to join with respect to year founded")
# Set x-axis label
plt.xlabel("Year Founded")
# Set y-axis label
plt.ylabel("Years to join unicorn status")
# Make sure the x-axis goes in chronological order by month.
plt.xticks(rotation=45, horizontalalignment="right")
# Display the plot.
plt.show()

Out[200].
Bar plot of years to join with respect to year founded
Years to join unicorn status
100
80
60
40
20
0
1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021
Year founded

In [202]. # Set the size of the plot.
plt.figure(figsize=(10, 5))
# Create bar plot to visualize number of companies that joined per interval for the year of interest.
plt.bar(x=companies_by_week_2021["Week Joined"], height=companies_by_week_2021["Company Count"])
plt.plot()
# Set the x-axis label.
plt.xlabel("Week number")
# Set the y-axis label.
plt.ylabel("Number of companies")
# Set the title.
plt.title("Number of companies that became unicorns per week in 2021")
# Rotate the labels on the x-axis as a way to avoid overlap in the positions of the text.
plt.xticks(rotation=45, horizontalalignment="right", fontsize=8)
# Display the plot.
plt.show()

Out[202].
Number of companies that became unicorns per week in 2021
Number of companies
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
2021-W01 2021-W02 2021-W03 2021-W04 2021-W05 2021-W06 2021-W07 2021-W08 2021-W09 2021-W10 2021-W11 2021-W12 2021-W13 2021-W14 2021-W15 2021-W16 2021-W17 2021-W18 2021-W19 2021-W20 2021-W21 2021-W22 2021-W23 2021-W24 2021-W25 2021-W26 2021-W27 2021-W28 2021-W29 2021-W30 2021-W31 2021-W32 2021-W33 2021-W34 2021-W35 2021-W36 2021-W37 2021-W38 2021-W39 2021-W40 2021-W41 2021-W42 2021-W43 2021-W44 2021-W45 2021-W46 2021-W47 2021-W48 2021-W49 2021-W50 2021-W51 2021-W52
Week number

In [204]. # Using slicing, extract the year component and the time interval that you specified.
# and save them by adding two new columns into the subset.
companies_by_quarter_2020_2021["Quarter Number"] = companies_by_quarter_2020_2021["Quarter Joined"].str[-2:]
companies_by_quarter_2020_2021["Year Joined"] = companies_by_quarter_2020_2021["Quarter Joined"].str[4]
# Set the size of the plot.
plt.figure(figsize=(10, 5))
# Create a grouped bar plot.
sns.barplot(x=companies_by_quarter_2020_2021["Quarter Number"],
            y=companies_by_quarter_2020_2021["Average Valuation"],
            hue=companies_by_quarter_2020_2021["Year Joined"])
plt.plot()
# Set the x-axis label.
plt.xlabel("Quarter number")
# Set the y-axis label.
plt.ylabel("Average valuation (billions of dollars)")
# Set the title.
plt.title("Average valuation of companies that became unicorns per quarter in 2020 vs. 2021")
# Display the plot.
plt.show()

Out[204].
Average valuation of companies that became unicorns per quarter in 2020 vs. 2021
Average valuation (billions of dollars)
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Q1 Q2 Q3 Q4
Quarter number
Year joined
2020
2021

In [ ] . # Key takeaways
# Functions in the pandas library can be used for data manipulation in order to reorganize and structure the data.
# Converting strings that contain dates to datetime format allow you to extract individual components from the data, such as month and year.
# Structuring the data in specific ways allows you to observe more trends and zoom in on parts of the data that are interesting to you.
# Functions in the matplotlib.pyplot module and the seaborn library can be used to create visualizations to gain further insight after structuring the data.
# Findings you share with others
# There were 1074 unicorn companies represented in this dataset.
# 2015 is the year when the most number of unicorn companies were founded.
# Many of the unicorn companies that were founded in 2021 were founded in the United States and belong to "Fintech", "E-commerce & direct-to-consumer", and "Internet software & services" industries.
# The box plot created shows that companies that became unicorns in the months of September and October have a smaller median value for how long it took to become unicorns.
# One of the bar plots created shows that the average valuation of companies that joined in 2021 is highest in the third quarter of the year, whereas the average valuation of companies that joined in 2020 is highest in the first quarter.
# Recommendations you share with stakeholders based on these findings
# According to data analysis that was conducted on a dataset of 1074 unicorn companies, companies that joined in the months of September and October tended to take less time to become unicorns.
# Another finding was that many of the unicorn companies that were founded in 2021 were founded in the United States and belong to "Fintech", "E-commerce & direct-to-consumer", and "Internet software & services" industries. So if the
# It was also discovered that the average valuation of companies that joined in 2021 is higher in the first quarter of the year, and the average valuation of companies that joined in 2020 is the third quarter of the year. When comin
# The data can be analyzed further to gather more insights that are specific to the interests of the investing firm and the stakeholders.
```