Unicorn companies data set

1. # Sort `companies` and display the first 10 rows of the resulting Df.

# To arrange the data from latest to earliest `Year Founded`

🡪companies.sort\_values(by="Year Founded", ascending=False).head(10)

2. Determine the number of companies founded each year

# 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)

3. Convert the Date Joined column to datetime –

companies["Date Joined"] = pd.to\_datetime(companies["Date Joined"])

4 month joined column

companies["Month Joined"] = companies["Date Joined"].dt.month\_name()

Determine how many years it took for companies to reach unicorn status

companies["Years To Join"] = companies["Date Joined"].dt.year - companies["Year Founded"]

get more insight on the year of that interests you

companies\_2021 = companies[companies["Date Joined"].dt.year == 2021]

5. # After identifying the time interval that interests you, proceed with the following:

# Insert a `Week Joined` column into `companies\_2021.`

companies\_2021.insert(3, "Week Joined", companies\_2021["Date Joined"].dt.strftime('%Y-W%V'), True)

# Group `companies\_2021` by `Week Joined`.

# Aggregate by counting companies that joined per week of 2021..

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()

# 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-Q%q')

# 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"].apply(lambda v: v.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"})

6.

Create the box plot

# to visualize the distribution of how long it took companies to become unicorns, with respect to the month they joined

sns.boxplot(x=companies['Month Joined'], y=companies['Years To Join'], order=month\_order, showfliers=False)

Create bar plot to visualize the average number of years it took companies to reach unicorn status with respect to when they were founded.

sns.barplot(x=companies["Year Founded"], y=companies["Years To Join"], ci=False)