

In [1]:

```
import pandas as pd
```

In [2]:

```
df = pd.read_csv('unicorn.csv')
```

In [3]:

```
df.head()
```

Out[3]:

	Company	Valuation(\$B)	Date Joined	Country	City	Industry	Investors	Unname
0	ByteDance	\$140	04-07-2017	China	Beijing	Artificial intelligence	Sequoia Capital China	SIG / Investme
1	SpaceX	\$127	12-01-2012	United States	Hawthorne	Other	Founders Fund	Draper Fis Jurvel
2	SHEIN	\$100	07-03-2018	China	Shenzhen	E-commerce & direct-to-consumer	Tiger Global Management	Sequoia Ca Cl
3	Stripe	\$95	1/23/2014	United States	San Francisco	Fintech	Khosla Ventures	LowercaseCa
4	Canva	\$40	01-08-2018	Australia	Surry Hills	Internet software & services	Sequoia Capital China	Black Vent



```
In [4]:
df.tail()
```

Out[4]:

	Company	Valuation(\$B)	Date Joined	Country	City	Industry	Investors	Unnamed: 11
1199	LeadSquared	\$1	6/21/2022	India	Bengaluru	Internet software & services	Gaja Capital Partners	Stakeholder Capital
1200	FourKites	\$1	6/21/2022	United States	Chicago	Supply chain	logistics	& deliver
1201	VulcanForms	\$1	07-05- 2022	United States	Burlington	Supply chain	logistics	& deliver
1202	SingleStore	\$1	07-12- 2022	United States	San Francisco	Data management & analytics	Google Ventures	Account
1203	Unstoppable Domains	\$1	7/27/2022	United States	Las Vegas	Internet software & services	Boost VC	Draper Associate

```
In [5]:
df.shape
```

Out[5]:

(1204, 12)

```
In [6]:
df.columns
```

Out[6]:

```
Index(['Company', 'Valuation($B)', 'Date Joined', 'Country', 'City',
      'Industry', 'Investors', 'Unnamed: 7', 'Unnamed: 8', 'Unnamed: 9',
      'Unnamed: 10', 'Unnamed: 11'],
      dtype='object')
```

```
In [7]:
df.duplicated().sum()
```

Out[7]:

0

In [8]:

```
df.isnull().sum()
```

Out[8]:

```
Company          0
Valuation($B)    0
Date Joined      0
Country          0
City            17
Industry         0
Investors        1
Unnamed: 7       57
Unnamed: 8      148
Unnamed: 9     1132
Unnamed: 10     1142
Unnamed: 11     1201
dtype: int64
```

In [9]:

```
df = df[['Company', 'Valuation($B)', 'Date Joined', 'Country', 'City',
        'Industry', 'Investors']]
```

In [10]:

```
df.isnull().sum()
```

Out[10]:

```
Company          0
Valuation($B)    0
Date Joined      0
Country          0
City            17
Industry         0
Investors        1
dtype: int64
```

In [11]:

```
df['City'] = df['City'].fillna('Not Available')
df['Investors'] = df['Investors'].fillna('Not Available')
```

In [12]:

```
df.isnull().sum()
```

Out[12]:

```
Company          0
Valuation($B)    0
Date Joined      0
Country          0
City             0
Industry         0
Investors        0
dtype: int64
```

In [13]:

```
df
```

Out[13]:

	Company	Valuation(\$B)	Date Joined	Country	City	Industry	Investors
0	ByteDance	\$140	04-07-2017	China	Beijing	Artificial intelligence	Sequoia Capital China
1	SpaceX	\$127	12-01-2012	United States	Hawthorne	Other	Founders Fund
2	SHEIN	\$100	07-03-2018	China	Shenzhen	E-commerce & direct-to-consumer	Tiger Global Management
3	Stripe	\$95	1/23/2014	United States	San Francisco	Fintech	Khosla Ventures
4	Canva	\$40	01-08-2018	Australia	Surry Hills	Internet software & services	Sequoia Capital China
...
1199	LeadSquared	\$1	6/21/2022	India	Bengaluru	Internet software & services	Gaja Capital Partners
1200	FourKites	\$1	6/21/2022	United States	Chicago	Supply chain	logistics
1201	VulcanForms	\$1	07-05-2022	United States	Burlington	Supply chain	logistics
1202	SingleStore	\$1	07-12-2022	United States	San Francisco	Data management & analytics	Google Ventures
1203	Unstoppable Domains	\$1	7/27/2022	United States	Las Vegas	Internet software & services	Boost VC

1204 rows × 7 columns

In [14]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1204 entries, 0 to 1203
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Company         1204 non-null  object
1   Valuation($B)   1204 non-null  object
2   Date Joined     1204 non-null  object
3   Country         1204 non-null  object
4   City            1204 non-null  object
5   Industry        1204 non-null  object
6   Investors       1204 non-null  object
dtypes: object(7)
memory usage: 66.0+ KB
```

In [15]:

```
df.nunique()
```

Out[15]:

```
Company      1201
Valuation($B) 229
Date Joined   715
Country       49
City          283
Industry      16
Investors     584
dtype: int64
```

In [16]:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [17]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [18]:

```
df['Country'].unique()
```

Out[18]:

```
array(['China', 'United States', 'Australia', 'United Kingdom', 'India',
       'Indonesia', 'Germany', 'Turkey', 'Hong Kong', 'Seychelles',
       'Sweden', 'Mexico', 'Estonia', 'Canada', 'South Korea',
       'Netherlands', 'France', 'Israel', 'Finland', 'Colombia',
       'Belgium', 'Brazil', 'Denmark', 'Switzerland', 'Lithuania',
       'Austria', 'Ireland', 'Singapore', 'Vietnam',
       'United Arab Emirates', 'Argentina', 'Spain', 'Japan',
       'Luxembourg', 'Nigeria', 'Philippines', 'Croatia', 'Senegal',
       'Malaysia', 'Bermuda', 'Norway', 'South Africa', 'Ecuador',
       'Chile', 'Thailand', 'Czech Republic', 'Liechtenstein', 'Italy',
       'London'], dtype=object)
```

In [19]:

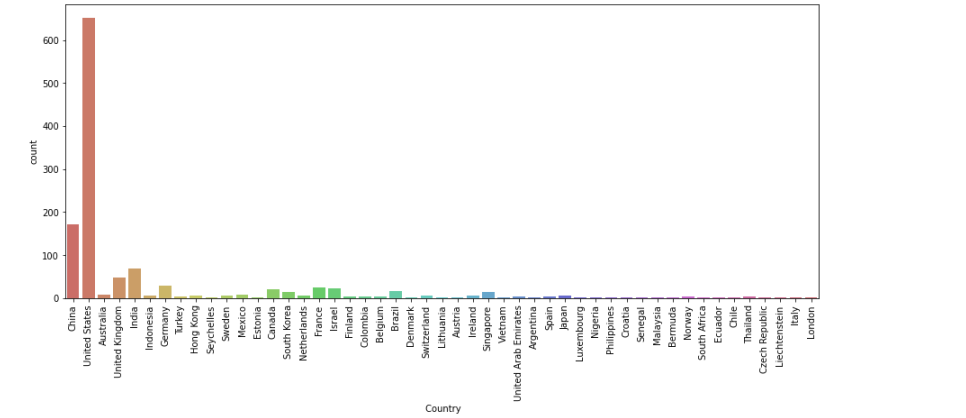
```
df['Country'].value_counts()
```

Out[19]:

United States	651
China	172
India	70
United Kingdom	49
Germany	29
France	25
Israel	23
Canada	20
Brazil	16
Singapore	14
South Korea	14
Mexico	8
Australia	8
Indonesia	7
Hong Kong	7
Sweden	7
Netherlands	7
Ireland	6
Switzerland	6
Japan	6
Spain	5
Finland	4
Norway	4
Belgium	3
Turkey	3
Thailand	3
United Arab Emirates	3
Colombia	3
Chile	2
Philippines	2
South Africa	2
Italy	2
Croatia	2
Lithuania	2
Vietnam	2
Austria	2
Denmark	2
Estonia	2
Nigeria	1
Luxembourg	1
Argentina	1
Senegal	1
Malaysia	1
Bermuda	1
Ecuador	1
Czech Republic	1
Liechtenstein	1
Seychelles	1
London	1

Name: Country, dtype: int64

```
In [20]:  
plt.figure(figsize=(15,6))  
sns.countplot(df['Country'], data = df,  
              palette = 'hls')  
plt.xticks(rotation = 90)  
plt.show()
```



In [21]:

```
df[' City'].unique()
```


Out[21]:

```
array(['Beijing', 'Hawthorne', 'Shenzhen', 'San Francisco', 'Surry Hills',  
      'London', 'Cary', 'Jacksonville', 'Bengaluru', 'Shanghai',  
      'Jakarta', 'Philadelphia', 'New York', 'Munich', 'Waltham',  
      'San Diego', 'Roseville', 'Changsha', 'Istanbul', 'Luohu',  
      'Cheung Sha Wan', 'Walpole', 'Victoria', 'Boston', 'Glendale',  
      'Berlin', 'Palo Alto', 'Stockholm', 'Kirkland', 'Gurugram',  
      'Lerma de Villada', 'Mountain View', 'Fremont', 'Irvine',  
      'Tallinn', 'San Jose', 'San Mateo', 'Chicago', 'Mumbai', 'Irving',  
      'Vancouver', 'Santa Clara', 'Pittsburgh', 'Seoul', 'Hangzhou',  
      'Toronto', 'Changzhou', 'Amsterdam', 'Paris', 'Los Angeles',  
      'Tel Aviv', 'Hoboken', 'Brooklyn', 'Helsinki', 'Pflugerville',  
      'Oakland', 'Bogota', 'Brussels', 'Atlanta', 'Campinas',  
      'Sao Paulo', 'Melbourne', 'Gurgaon', 'Noida', 'Bellevue',  
      'Montreal', 'Copenhagen', 'Geneva', 'Burlingame', 'Vilnius',  
      'Faridabad', 'Seattle', 'Guangzhou', 'Denver', 'Eden Prairie',  
      'Inglewood', 'Sunnyvale', 'Vienna', 'Santa Monica', 'Redwood City',  
      'El Segundo', 'Mississauga', 'Dublin', 'Columbus', 'Carlsbad',  
      'Los Altos', 'San Carlos', 'Miami', 'Chengdu', 'Detroit', 'Bend',  
      'Culver City', 'Carson City', 'Suzhou', 'Marina del Rey',  
      'Duderstadt', 'Not Available', 'San Ramon', 'Redmond',  
      'Schaffhausen', 'Louvain-la-Neuve', 'Englewood Cliffs', 'Wuhan',  
      'Alameda', 'Sacramento', 'Hayward', 'Houston', 'Peterborough',  
      'Kitchener', 'Jersey City', 'Cambridge', 'South Jordan',  
      'Ho Chi Minh City', 'Westerville', 'Lehi', 'Fort Lee', 'Austin',  
      'New Delhi', 'Zhuhai', 'Chatham', 'South San Francisco', 'Bristol',  
      'Scottsdale', 'Dallas', 'Raleigh', 'Louisville', 'Oulu', 'Boulder',  
      'Dubai', 'Nanjing', 'Thane', 'Burlington', 'McLean', 'Sydney',  
      'Madison', 'Rancho', 'Buenos Aires', 'Columbia', 'Foster City',  
      'South Burlington', 'Madrid', 'Unterfoehring', 'Mexico City',  
      'Colchester', 'Maharashtra', 'Pune', 'Bethesda', 'Orlando',  
      'Hong Kong', 'Kowloon', 'Brisbane', 'Tokyo', 'Somerville',  
      'Givatayim', 'Leudelage', 'Goleta', 'Lagos', 'Pennsauken',  
      'Plantation', 'Portola Valley', 'Taguig City', 'Netanya', 'Croix',  
      'Aarhus', 'Sveta Nedelja', 'Bangalore', 'Boca Raton', 'Bruchsal',  
      'Jerusalem', 'Portland', 'Greenwood Village', 'Dakar', 'Hanover',  
      'Selangor', 'Chongqing', 'Hamilton', 'Burnaby', 'Lysaker',  
      'Bryanston', 'Hunan', 'Jiangsu', 'Northbrook', 'Quito',  
      'Santa Barbara', 'Santiago', 'Richmond', 'Draper', 'Cincinnati',  
      'Zephyr Cove', 'Altrincham', 'Bangkok', 'Nashville', 'Curitiba',  
      'Herzliya', 'DC', 'Santa Cruz', 'Hefei', 'Solihull', 'Ghent',  
      'Ambler', 'Pleasanton', 'La Plaine Saint-Denis', 'Central',  
      'Framingham', 'Salt Lake City', 'Tirat Carmel', 'Encinitas',  
      'Barcelona', 'Long Beach', 'Berkeley', 'Chennai',  
      'Port Washington', 'Sherman Oaks', 'Boise', 'Brookline',  
      'Aberdeen', 'Tsuruoka', 'Washington', 'Prague', 'Princeton',  
      'Jaipur', 'Juarez', 'Gloucester', 'Washington DC', 'Tustin',  
      'Qingdao', 'Roubaix', 'Islandia', 'Prilly', 'Monterrey', 'Andheri',  
      'Waterloo', 'Chemnitz', 'Quincy', 'Uttar Pradesh',  
      'Banyeres de Mariola', 'Sarasota', 'Crewe', 'Vaduz',  
      'Tel Aviv-Yafo', 'Milan', 'Scotts Valley', 'Kilkenny', 'Fort Mill',  
      'Lausanne', 'Midrand', 'Manila', 'Espoo', 'Wuxi', 'Charlotte',  
      'Menlo Park', 'Evry', 'Guiyang', 'Leawood', 'Zurich', 'Dongguan',  
      'Vodnjan', 'Englewood', 'Colorado Springs', 'Parana', 'Wilmington',  
      'Ramat Gan', 'Cedar Park', 'United Kingdom', 'Berkeley Heights',  
      'Lincoln', 'Arlington', 'Oslo', 'Montpellier', 'Bellingham',  
      'Herndon', 'Tampa', 'Alexandria', 'Ottawa', 'Petah Tikva',  
      'Hyderabad', 'Milpitas', 'Venice', 'Torrance', 'Kista',  
      'Morrisville', 'Yehud', 'Las Vegas'], dtype=object)
```

In [22]:

```
df[' City'].value_counts()
```

Out[22]:

```
San Francisco    169
New York         116
Beijing          62
Shanghai         43
London           39
...
Leudelange       1
Goleta           1
Lagos            1
Pennsauken       1
Las Vegas        1
Name: City, Length: 283, dtype: int64
```

In [23]:

```
df1 = df[' City'].value_counts()
```

In [24]:

```
df1.to_frame()
```

Out[24]:

	City
San Francisco	169
New York	116
Beijing	62
Shanghai	43
London	39
...	...
Leudelange	1
Goleta	1
Lagos	1
Pennsauken	1
Las Vegas	1

283 rows × 1 columns

In [25]:

```
df1 = df1.reset_index()
```

In [26]:

```
df1
```

Out[26]:

	index	City
0	San Francisco	169
1	New York	116
2	Beijing	62
3	Shanghai	43
4	London	39
...
278	Leudelage	1
279	Goleta	1
280	Lagos	1
281	Pennsauken	1
282	Las Vegas	1

283 rows × 2 columns

In [27]:

```
df1 = df1.rename(columns = {'index':'City', ' City':'Count'})
```

In [28]:

```
df1
```

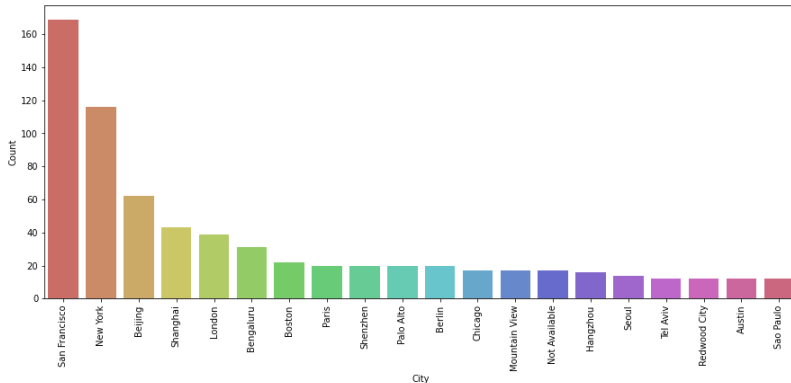
Out[28]:

	City	Count
0	San Francisco	169
1	New York	116
2	Beijing	62
3	Shanghai	43
4	London	39
...
278	Leudelage	1
279	Goleta	1
280	Lagos	1
281	Pennsauken	1
282	Las Vegas	1

283 rows × 2 columns

In [29]:

```
plt.figure(figsize=(15,6))
sns.barpplot(x = df1['City'].head(20), y = df1['Count'].head(20), data = df1,
             palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



In [30]:

```
df[' Industry'].unique()
```

Out[30]:

```
array(['Artificial intelligence', 'Other',
      'E-commerce & direct-to-consumer', 'Fintech',
      'Internet software & services', 'Supply chain',
      'Data management & analytics', 'Edtech', 'Hardware',
      'Consumer & retail', 'Health', 'Auto & transportation',
      'Travel', 'Cybersecurity', 'Mobile & telecommunications',
      'Artificial Intelligence'], dtype=object)
```

In [31]:

```
df[' Industry'].value_counts()
```

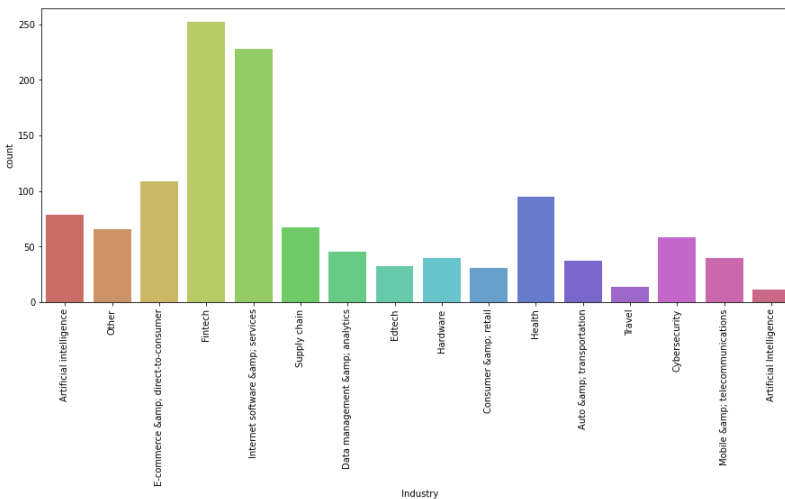
Out[31]:

Fintech	252
Internet software & services	228
E-commerce & direct-to-consumer	109
Health	95
Artificial intelligence	79
Supply chain	67
Other	66
Cybersecurity	58
Data management & analytics	45
Hardware	40
Mobile & telecommunications	40
Auto & transportation	37
Edtech	32
Consumer & retail	31
Travel	14
Artificial Intelligence	11

Name: Industry, dtype: int64

In [32]:

```
plt.figure(figsize=(15,6))
sns.countplot(df[' Industry'], data = df,
              palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



In [33]:

```
df[['Currency', 'Value']] = df[' Valuation($B)'].str.extract(r'([^\d]*)([\d,\.\+]+)')
```

```
In [34]:
df
```

Out[34]:

	Company	Valuation(\$B)	Date Joined	Country	City	Industry	Investors	Curre
0	ByteDance	\$140	04-07-2017	China	Beijing	Artificial intelligence	Sequoia Capital China	
1	SpaceX	\$127	12-01-2012	United States	Hawthorne	Other	Founders Fund	
2	SHEIN	\$100	07-03-2018	China	Shenzhen	E-commerce & direct-to-consumer	Tiger Global Management	
3	Stripe	\$95	1/23/2014	United States	San Francisco	Fintech	Khosla Ventures	
4	Canva	\$40	01-08-2018	Australia	Surry Hills	Internet software & services	Sequoia Capital China	
...	
1199	LeadSquared	\$1	6/21/2022	India	Bengaluru	Internet software & services	Gaja Capital Partners	
1200	FourKites	\$1	6/21/2022	United States	Chicago	Supply chain	logistics	
1201	VulcanForms	\$1	07-05-2022	United States	Burlington	Supply chain	logistics	
1202	SingleStore	\$1	07-12-2022	United States	San Francisco	Data management & analytics	Google Ventures	
1203	Unstoppable Domains	\$1	7/27/2022	United States	Las Vegas	Internet software & services	Boost VC	

1204 rows × 9 columns



```
In [35]:
df['Date Joined'] = pd.to_datetime(df['Date Joined'])
```

```
In [36]:
df['Year'] = df['Date Joined'].dt.year
```

In [37]:

```
df
```

Out[37]:

	Company	Valuation(\$B)	Date Joined	Country	City	Industry	Investors	Currenc
0	ByteDance	\$140	2017-04-07	China	Beijing	Artificial intelligence	Sequoia Capital China	
1	SpaceX	\$127	2012-12-01	United States	Hawthorne	Other	Founders Fund	
2	SHEIN	\$100	2018-07-03	China	Shenzhen	E-commerce & direct-to-consumer	Tiger Global Management	
3	Stripe	\$95	2014-01-23	United States	San Francisco	Fintech	Khosla Ventures	
4	Canva	\$40	2018-01-08	Australia	Surry Hills	Internet software & services	Sequoia Capital China	
...
1199	LeadSquared	\$1	2022-06-21	India	Bengaluru	Internet software & services	Gaja Capital Partners	
1200	FourKites	\$1	2022-06-21	United States	Chicago	Supply chain	logistics	
1201	VulcanForms	\$1	2022-07-05	United States	Burlington	Supply chain	logistics	
1202	SingleStore	\$1	2022-07-12	United States	San Francisco	Data management & analytics	Google Ventures	
1203	Unstoppable Domains	\$1	2022-07-27	United States	Las Vegas	Internet software & services	Boost VC	

1204 rows × 10 columns



In [38]:

```
df['Year'].unique()
```

Out[38]:

```
array([2017, 2012, 2018, 2014, 2019, 2016, 2021, 2022, 2015, 2020, 2013, 2011, 2007], dtype=int64)
```

In [39]:

```
df['Year'].value_counts()
```

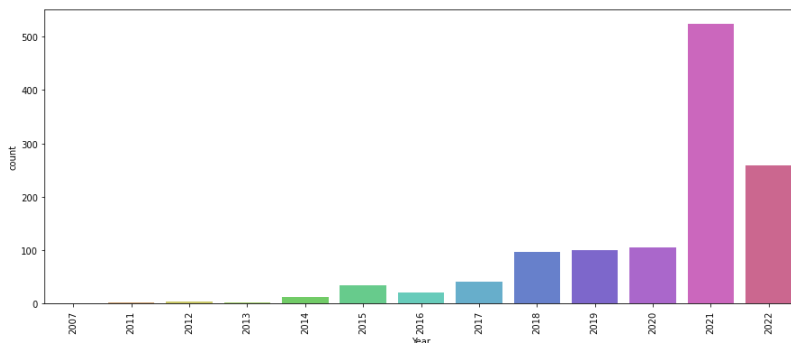
Out[39]:

```
2021    524
2022    258
2020    106
2019    100
2018     97
2017     42
2015     34
2016     21
2014     12
2012      4
2013      3
2011      2
2007      1
```

Name: Year, dtype: int64

In [40]:

```
plt.figure(figsize=(15,6))
sns.countplot(df['Year'], data = df,
              palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



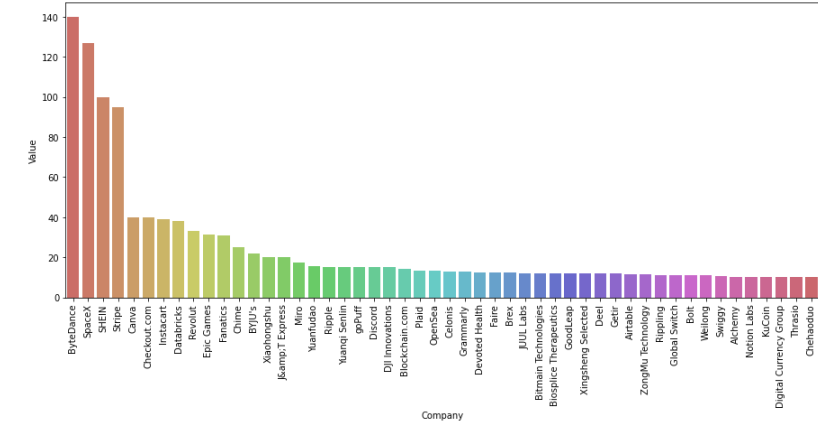

```
In [41]:
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1204 entries, 0 to 1203
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  ---
0    Company                1204 non-null   object
1    Valuation($B)          1204 non-null   object
2    Date Joined            1204 non-null   datetime64[ns]
3    Country                1204 non-null   object
4    City                  1204 non-null   object
5    Industry              1204 non-null   object
6    Investors              1204 non-null   object
7    Currency               1204 non-null   object
8    Value                  1204 non-null   object
9    Year                   1204 non-null   int64
dtypes: datetime64[ns](1), int64(1), object(8)
memory usage: 94.2+ KB
```

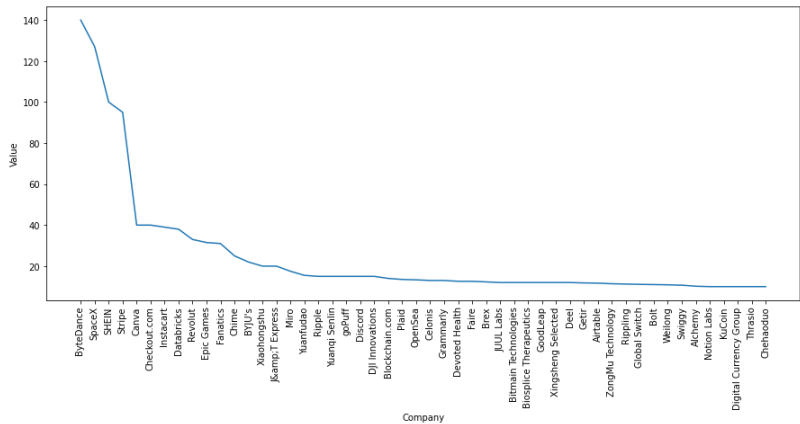
```
In [42]:
df['Value'] = df['Value'].astype(float)
```

```
In [43]:
df.sort_values(by='Value', ascending=False, inplace=True)
```

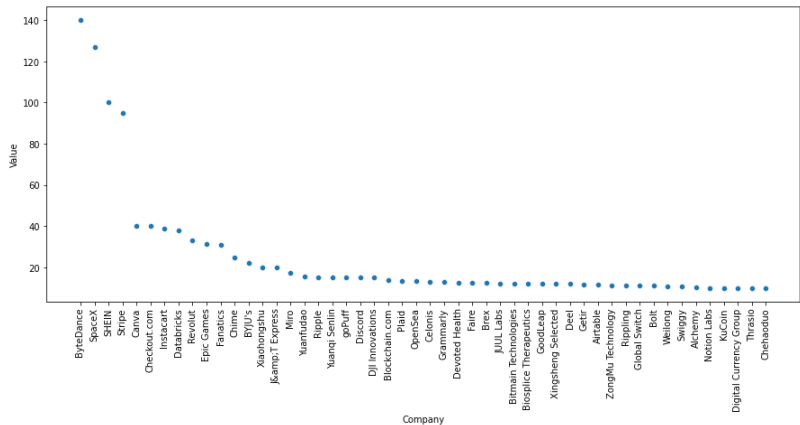
```
In [44]:
plt.figure(figsize=(15,6))
sns.barpplot(x = df['Company'].head(50), y = df['Value'].head(50), data = df,
             palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



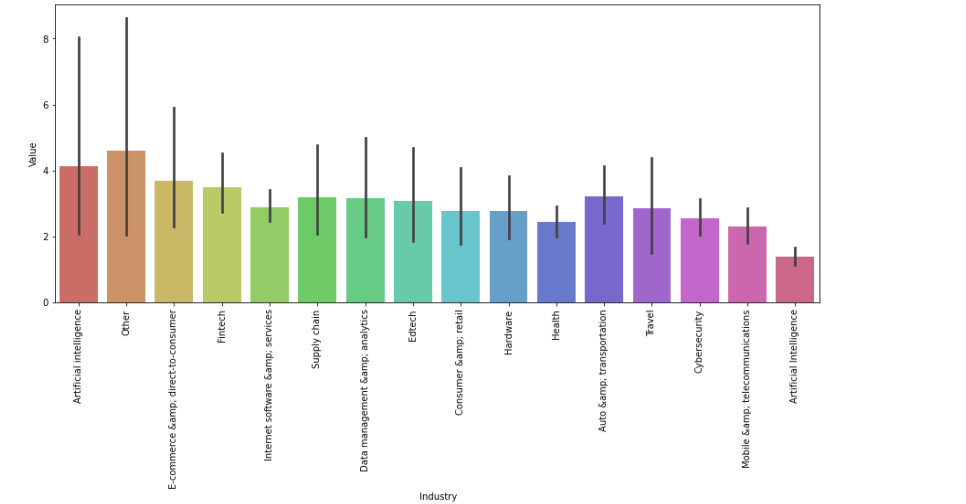
```
In [45]:  
  
plt.figure(figsize=(15,6))  
sns.lineplot(x = df['Company'].head(50), y = df['Value'].head(50), data = df,  
             palette = 'hls')  
plt.xticks(rotation = 90)  
plt.show()
```



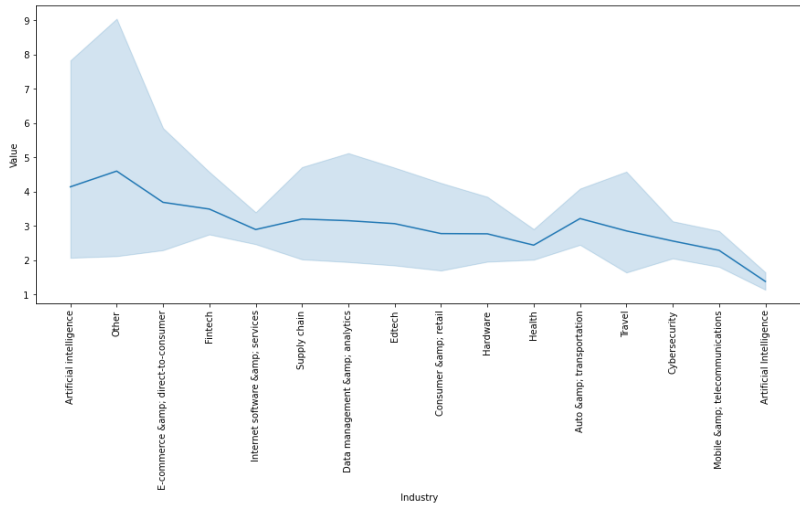
```
In [46]:  
  
plt.figure(figsize=(15,6))  
sns.scatterplot(x = df['Company'].head(50), y = df['Value'].head(50), data = df,  
               palette = 'hls')  
plt.xticks(rotation = 90)  
plt.show()
```



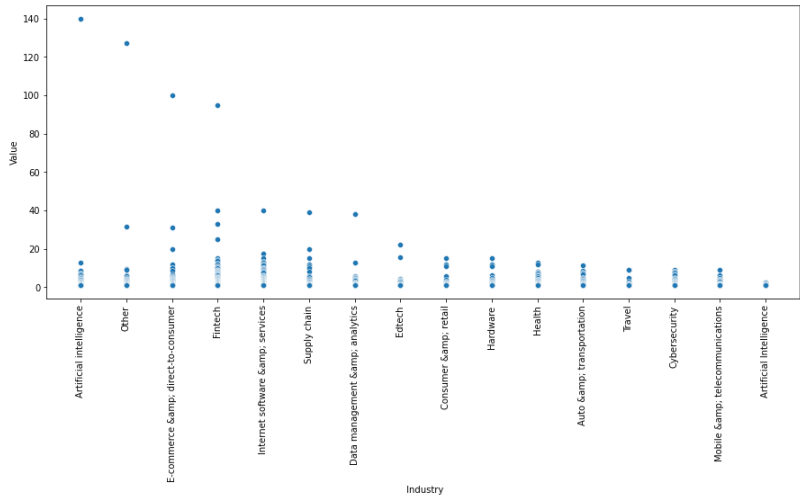
```
In [47]:  
plt.figure(figsize=(15,6))  
sns.barpplot(x = df[' Industry'], y = df['Value'], data = df,  
             palette = 'hls')  
plt.xticks(rotation = 90)  
plt.show()
```



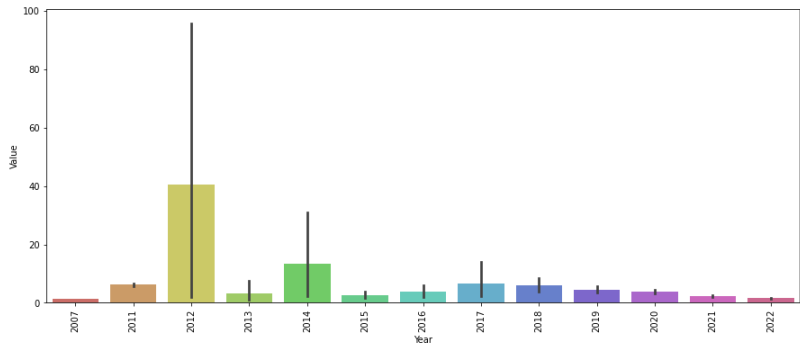
```
In [48]:
plt.figure(figsize=(15,6))
sns.lineplot(x = df[' Industry'], y = df['Value'], data = df,
             palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



```
In [49]:
plt.figure(figsize=(15,6))
sns.scatterplot(x = df[' Industry'], y = df['Value'], data = df,
               palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```

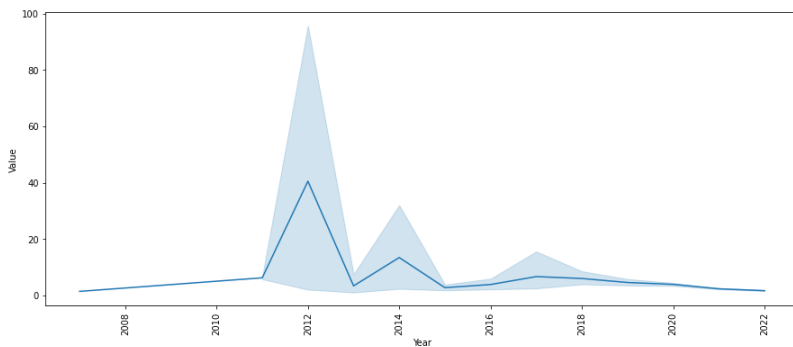


```
In [50]:
plt.figure(figsize=(15,6))
sns.barplot(x = df['Year'], y = df['Value'], data = df,
            palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



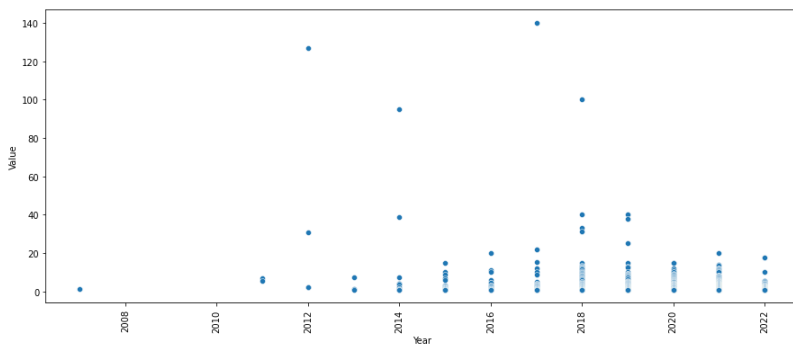
In [51]:

```
plt.figure(figsize=(15,6))
sns.lineplot(x = df['Year'], y = df['Value'], data = df,
             palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```

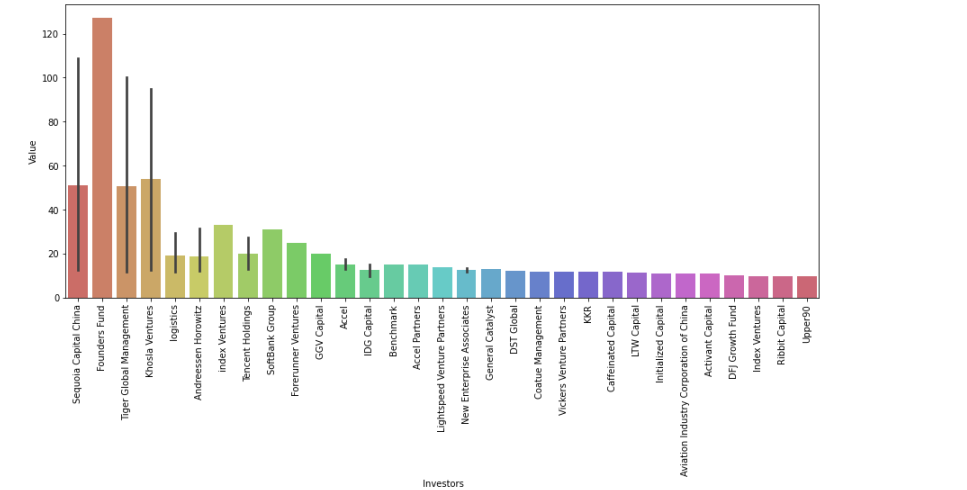


In [52]:

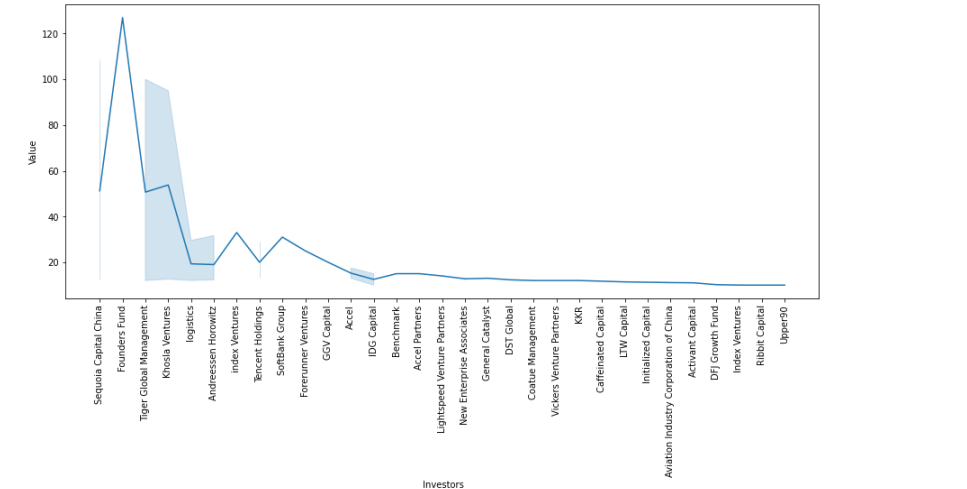
```
plt.figure(figsize=(15,6))
sns.scatterplot(x = df['Year'], y = df['Value'], data = df,
               palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



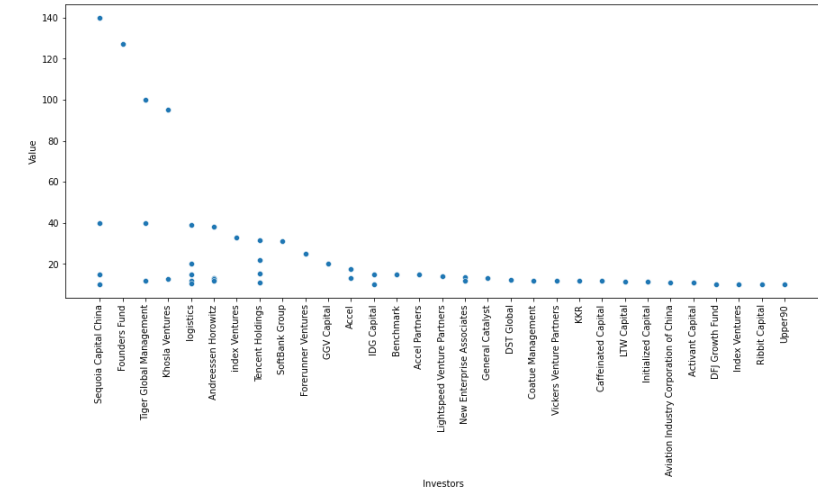
```
In [53]: plt.figure(figsize=(15,6))
sns.barpplot(x = df[' Investors'].head(50), y = df['Value'].head(50), data = df,
             palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



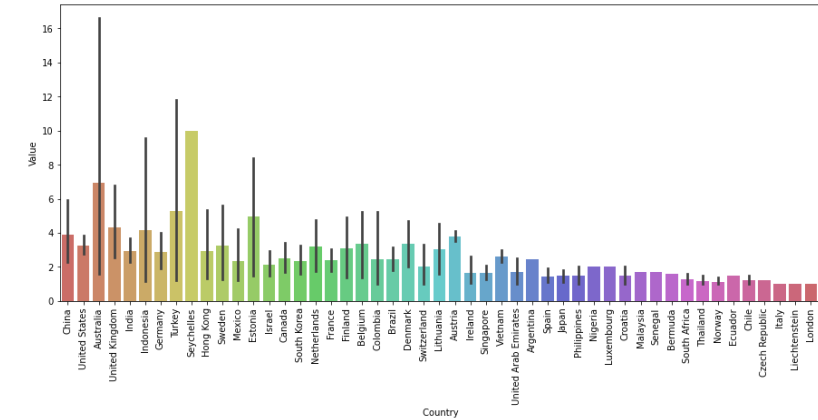
```
In [54]: plt.figure(figsize=(15,6))
sns.lineplot(x = df[' Investors'].head(50), y = df['Value'].head(50), data = df,
             palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



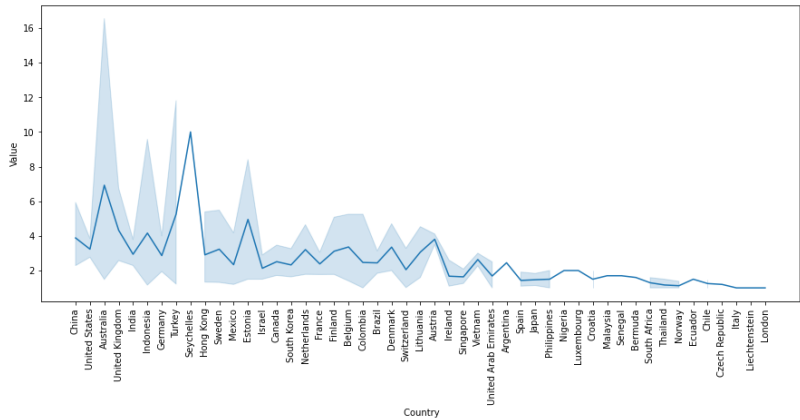
```
In [55]:
plt.figure(figsize=(15,6))
sns.scatterplot(x = df[' Investors'].head(50), y = df['Value'].head(50), data = df,
                palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



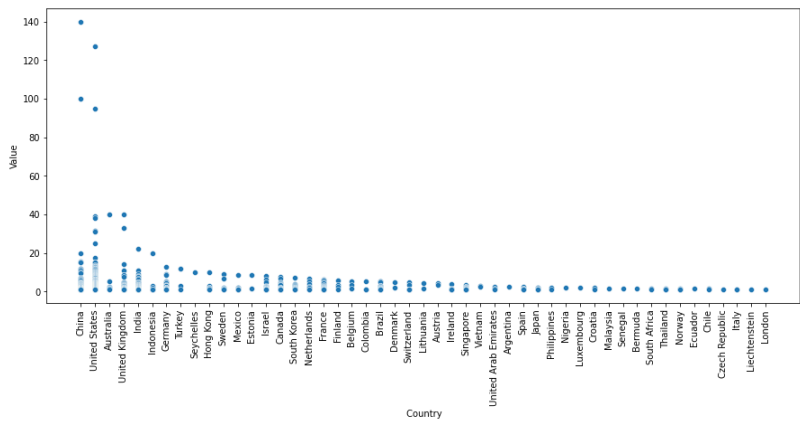
```
In [56]:
plt.figure(figsize=(15,6))
sns.barplot(x = df[' Country'], y = df['Value'], data = df,
            palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```




```
In [57]:  
plt.figure(figsize=(15,6))  
sns.lineplot(x = df[' Country'], y = df['Value'], data = df,  
             palette = 'hls')  
plt.xticks(rotation = 90)  
plt.show()
```

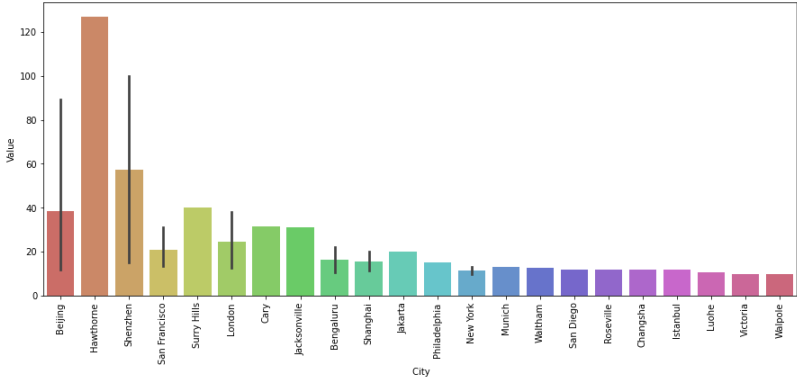


```
In [58]:  
plt.figure(figsize=(15,6))  
sns.scatterplot(x = df[' Country'], y = df['Value'], data = df,  
               palette = 'hls')  
plt.xticks(rotation = 90)  
plt.show()
```



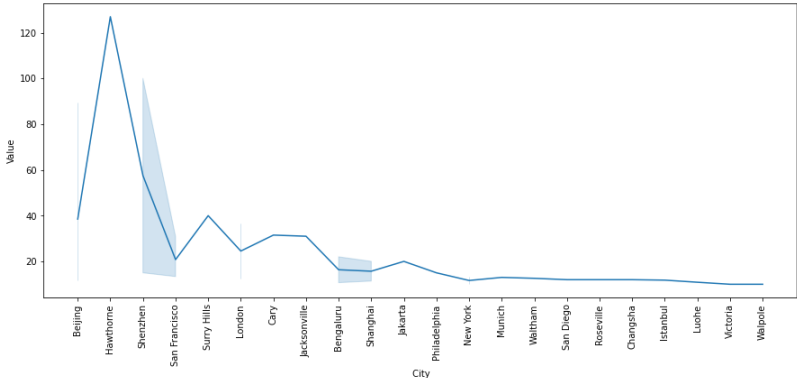
In [59]:

```
plt.figure(figsize=(15,6))
sns.barpplot(x = df[' City'].head(50), y = df['Value'].head(50), data = df,
             palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



In [60]:

```
plt.figure(figsize=(15,6))
sns.lineplot(x = df[' City'].head(50), y = df['Value'].head(50), data = df,
             palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



```
In [61]:  
plt.figure(figsize=(15,6))  
sns.scatterplot(x = df[' City'].head(50), y = df['Value'].head(50), data = df,  
                palette = 'hls')  
plt.xticks(rotation = 90)  
plt.show()
```

