In [1]:

```
from mpl_toolkits.mplot3d import Axes3D
import seaborn as sns


from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt # plotting
import numpy as np # linear algebra
import os # accessing directory structure

import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
```

In [2]:

```
df1=pd.read_csv("D:\datasets\startup/acquisitions.csv",low_memory=False)
df2=pd.read_csv("D:\datasets\startup/relationships.csv",low_memory=False)
df3=pd.read_csv("D:\datasets\startup/people.csv",low_memory=False)
df4=pd.read_csv("D:\datasets\startup/funds.csv",low_memory=False)
df5=pd.read_csv("D:\datasets\startup/investments.csv",low_memory=False)
df6=pd.read_csv("D:\datasets\startup/milestones.csv",low_memory=False)
df7=pd.read_csv("D:\datasets\startup/objects.csv",low_memory=False)
df8=pd.read_csv("D:\datasets\startup/offices.csv",low_memory=False)
df9=pd.read_csv("D:\datasets\startup/degrees.csv",low_memory=False)
df10=pd.read_csv("D:\datasets\startup/funding_rounds.csv",low_memory=False)
df11=pd.read_csv("D:\datasets\startup/ipos.csv",low_memory=False)
#df1=pd.read_csv("D:\datasets\startup/acquisitions.csv")
```

In [3]:

```
'''#HoloViews is an open-source Python library designed to make data analysis
and visualization seamless and simple. With HoloViews, you can usually
express what you want to do in very few lines of code,
letting you focus on what you are trying to explore and convey, not on the process of plott

import holoviews as hv
from holoviews import opts
from holoviews.operation.datashader import datashade, bundle_graph

import networkx as nx
```

In [4]:

```
hv.extension('bokeh')

defaults = dict(width=600, height=600, padding=0.1, yaxis=None, xaxis=None, show_frame=Fals
hv.opts.defaults(
    opts.EdgePaths(**defaults), opts.Graph(**defaults), opts.Nodes(**defaults))
```

In [5]:

```
df = df3.merge(df9, on='object_id')
```

In [6]:

```
df['full_name'] = df['first_name'].str.cat(df['last_name'],sep=" ")
```

In [7]:

```
df['institution'] = df['institution'].replace('Harvard Business School' ,'Harvard Universit
df['institution'] = df['institution'].replace('Stanford University Graduate School of Busin
```

In [8]:

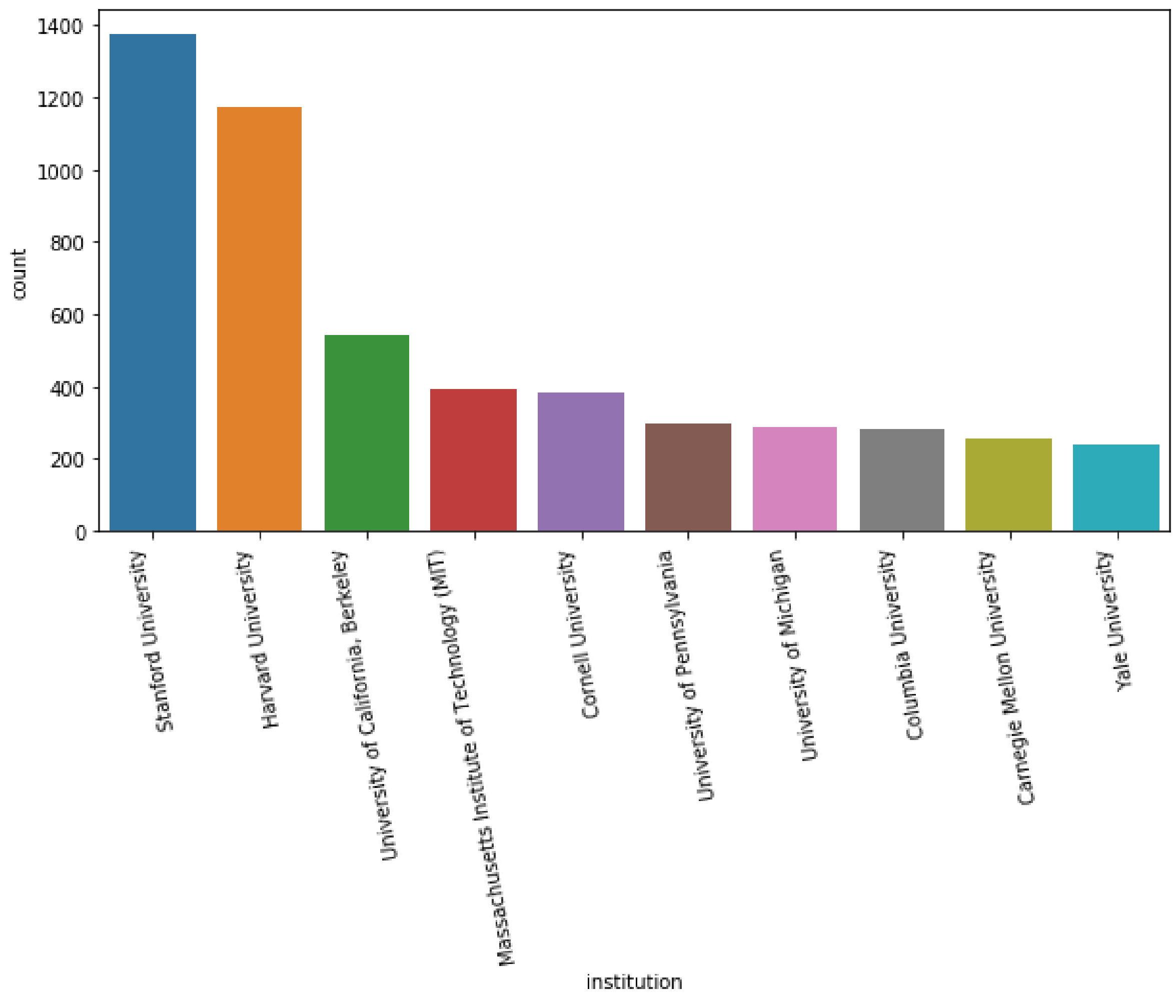
```
df = df[df['affiliation_name'] != 'Unaffiliated']
```

In [9]:

```
df = df[['object_id', 'full_name', 'birthplace', 'institution', 'degree_type', 'subject', '
```

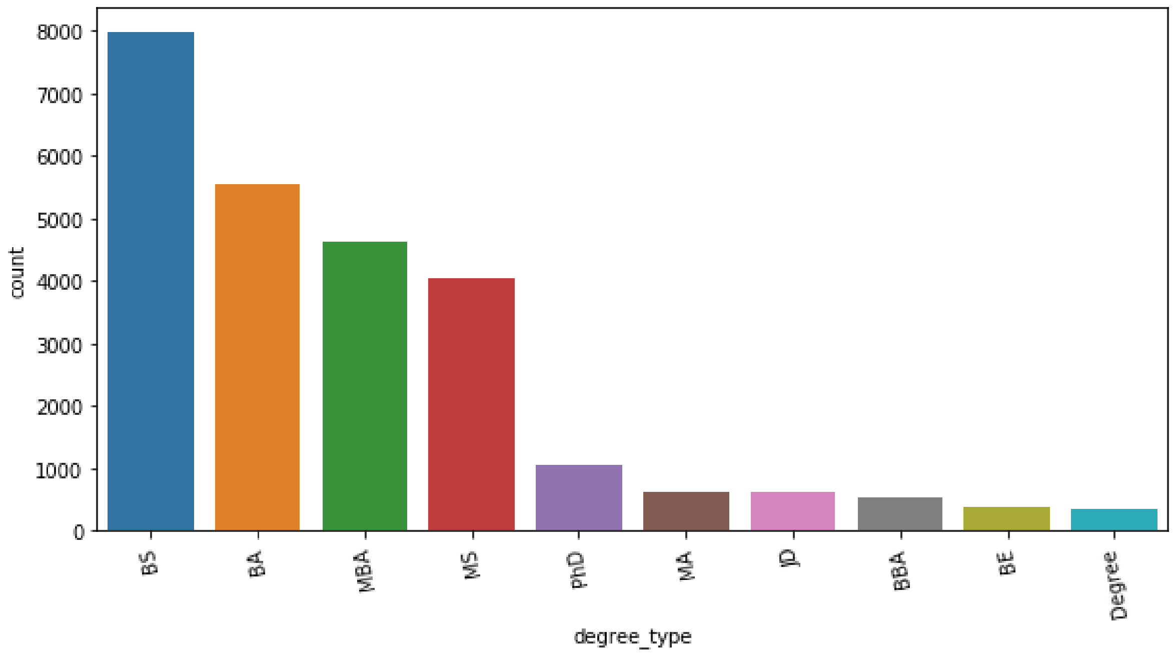
In [10]:

```
def count_plots(df, col_count):
    for i, col in enumerate(df.columns):
        plt.figure(i, figsize=(10,5))
        sns.countplot(x=col, data=df, order=pd.value_counts(df[col]).iloc[:col_count].index
        plt.xticks(rotation=100)

count_columns = df[['institution']]

count_plots(count_columns, 10)
```



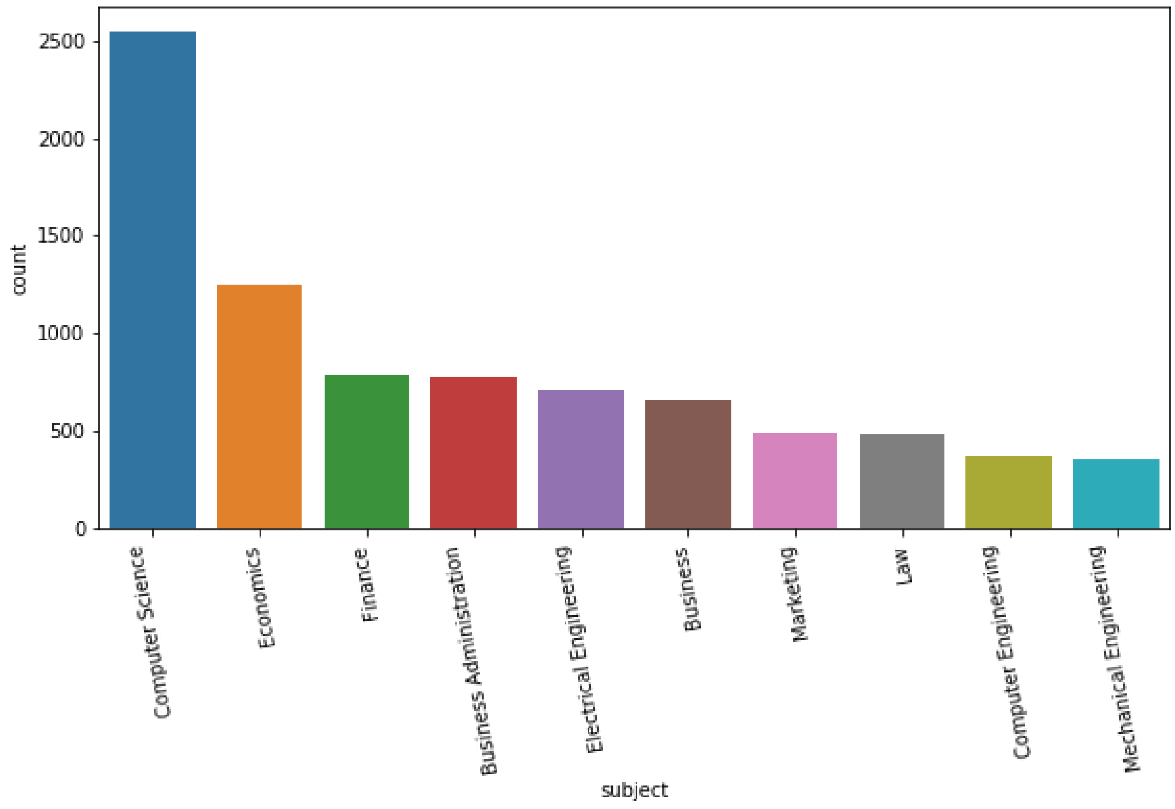
In [11]:

```
def count_plots(df, col_count):
    for i, col in enumerate(df.columns):
        plt.figure(i, figsize=(10,5))
        sns.countplot(x=col, data=df, order=pd.value_counts(df[col]).iloc[:col_count].index
        plt.xticks(rotation=100)

count_columns = df[['degree_type']]

count_plots(count_columns, 10)
```



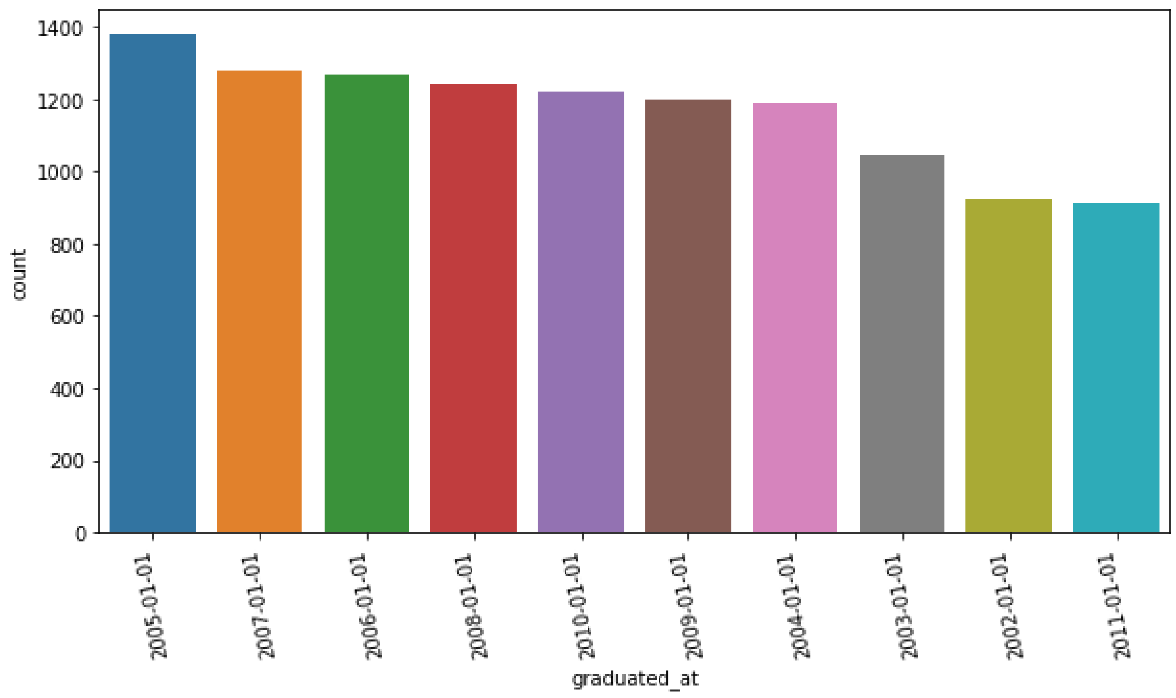
In [12]:

```
def count_plots(df, col_count):
    for i, col in enumerate(df.columns):
        plt.figure(i, figsize=(10,5))
        sns.countplot(x=col, data=df, order=pd.value_counts(df[col]).iloc[:col_count].index
        plt.xticks(rotation=100)

count_columns = df[['subject']]

count_plots(count_columns, 10)
```



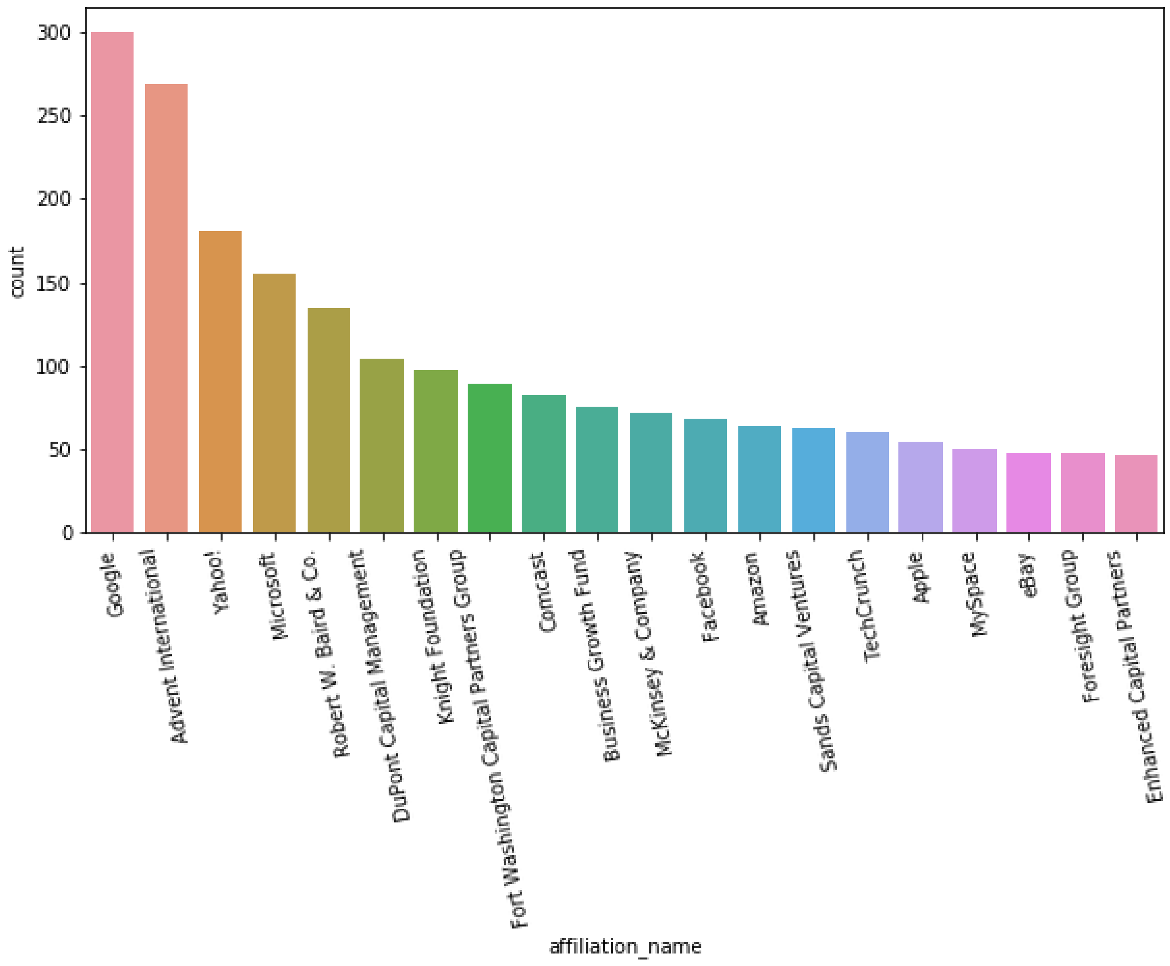
In [13]:

```
def count_plots(df, col_count):
    for i, col in enumerate(df.columns):
        plt.figure(i, figsize=(10,5))
        sns.countplot(x=col, data=df, order=pd.value_counts(df[col]).iloc[:col_count].index
        plt.xticks(rotation=100)

count_columns = df[['graduated_at']]

count_plots(count_columns, 10)
```



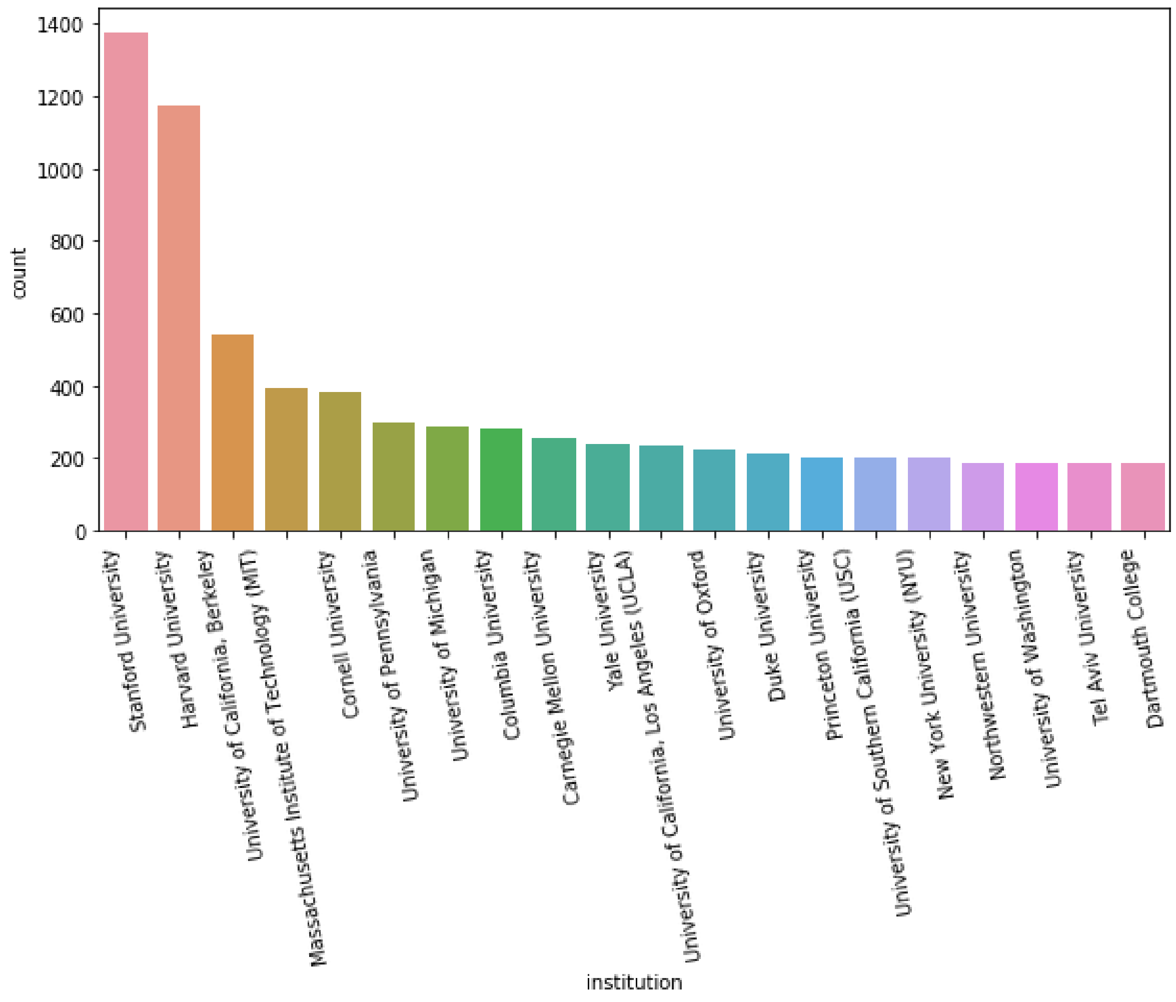
In [14]:

```
def count_plots(df, col_count):
    for i, col in enumerate(df.columns):
        plt.figure(i, figsize=(10,5))
        sns.countplot(x=col, data=df, order=pd.value_counts(df[col]).iloc[:col_count].index
        plt.xticks(rotation=100)

count_columns = df[['affiliation_name']]

count_plots(count_columns, 20)
```



In [15]:

```
def count_plots(df, col_count):
    for i, col in enumerate(df.columns):
        plt.figure(i, figsize=(10,5))
        sns.countplot(x=col, data=df, order=pd.value_counts(df[col]).iloc[:col_count].index
        plt.xticks(rotation=100)

count_columns = df[['institution']]

count_plots(count_columns, 20)
```



In [ ]:

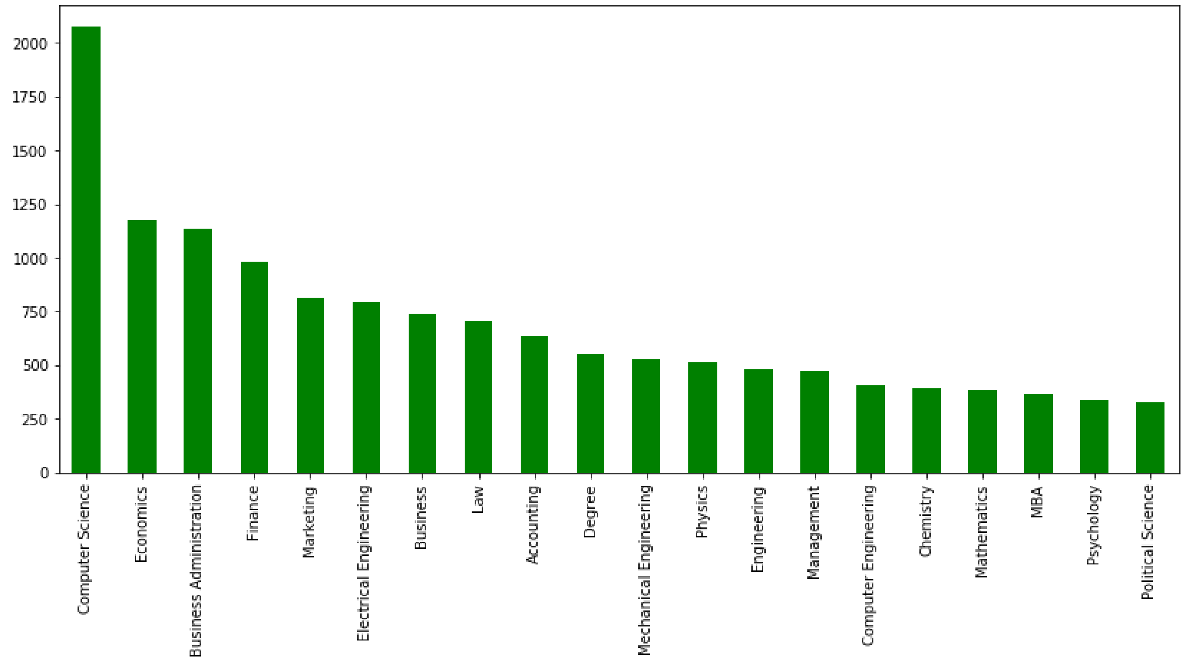
```

```

In [16]:

```
condition_dn = df9.groupby(['subject'])['institution'].nunique().sort_values(ascending=Fals
condition_dn[0:20].plot(kind="bar", figsize = (14,6), fontsize = 10,color="green")
plt.xlabel("", fontsize = 20)
plt.ylabel("", fontsize = 20)
plt.title("", fontsize = 20)
```

Out[16]:

```
Text(0.5, 1.0, '')
```



In [ ]:

```
```

In [ ]:

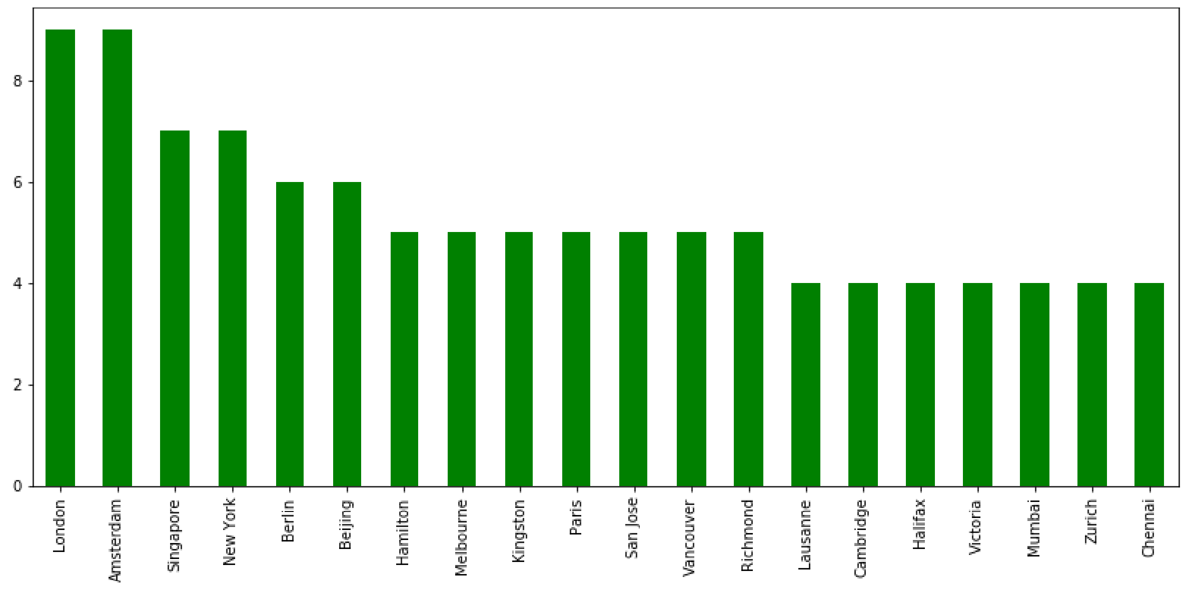
```
```

In [17]:

```
condition_dn = df8.groupby(['city'])['country_code'].nunique().sort_values(ascending=False)
condition_dn[0:20].plot(kind="bar", figsize = (14,6), fontsize = 10,color="green")
plt.xlabel("", fontsize = 20)
plt.ylabel("", fontsize = 20)
plt.title("", fontsize = 20)
```

Out[17]:

```
Text(0.5, 1.0, '')
```

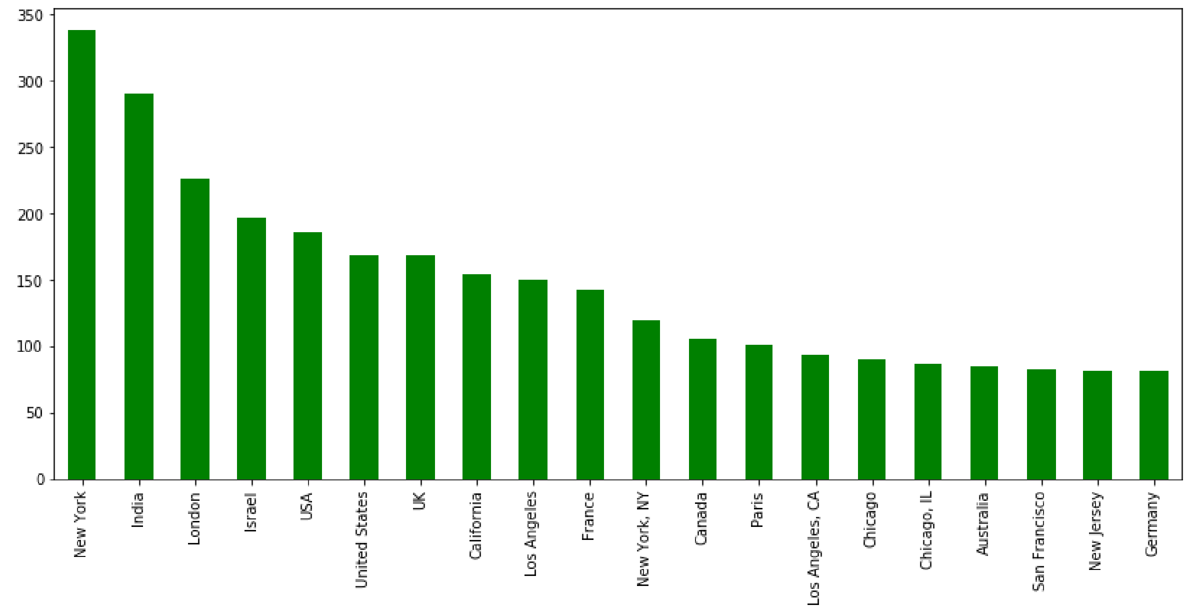


In [18]:

```
condition_dn = df3.groupby(['birthplace'])['affiliation_name'].nunique().sort_values(ascend
condition_dn[0:20].plot(kind="bar", figsize = (14,6), fontsize = 10,color="green")
plt.xlabel("", fontsize = 20)
plt.ylabel("", fontsize = 20)
plt.title("", fontsize = 20)
```

Out[18]:

```
Text(0.5, 1.0, '')
```

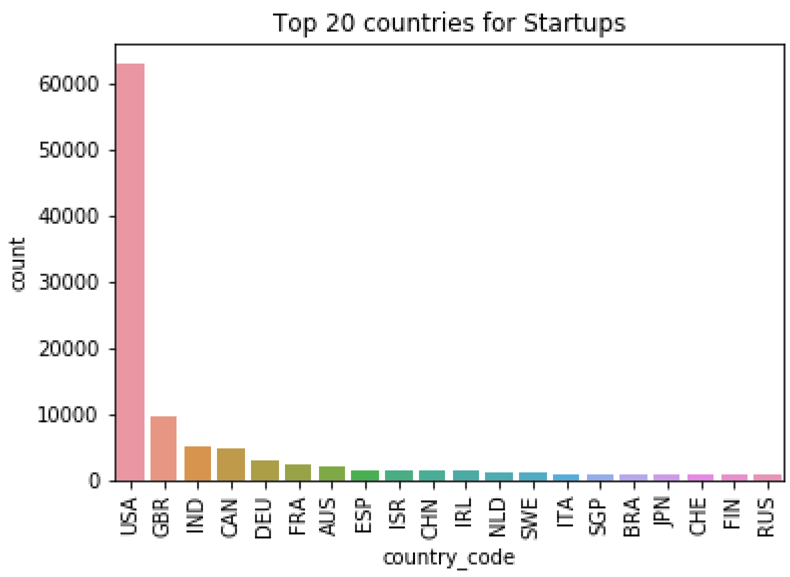


In [19]:

```
sns.countplot(data = df8,x="country_code",order=df8["country_code"].value_counts()[:20].ind
plt.xticks(rotation=90)
plt.title("Top 20 countries for Startups")
```

Out[19]:

```
Text(0.5, 1.0, 'Top 20 countries for Startups')
```

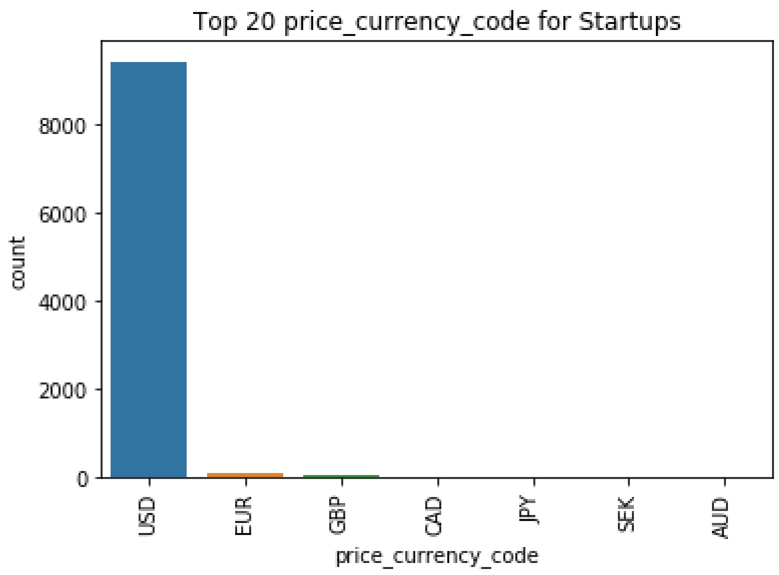


In [20]:

```
sns.countplot(data = df1,x="price_currency_code",order=df1["price_currency_code"].value_cou
plt.xticks(rotation=90)
plt.title("Top 20 price_currency_code for Startups")
```

Out[20]:

```
Text(0.5, 1.0, 'Top 20 price_currency_code for Startups')
```

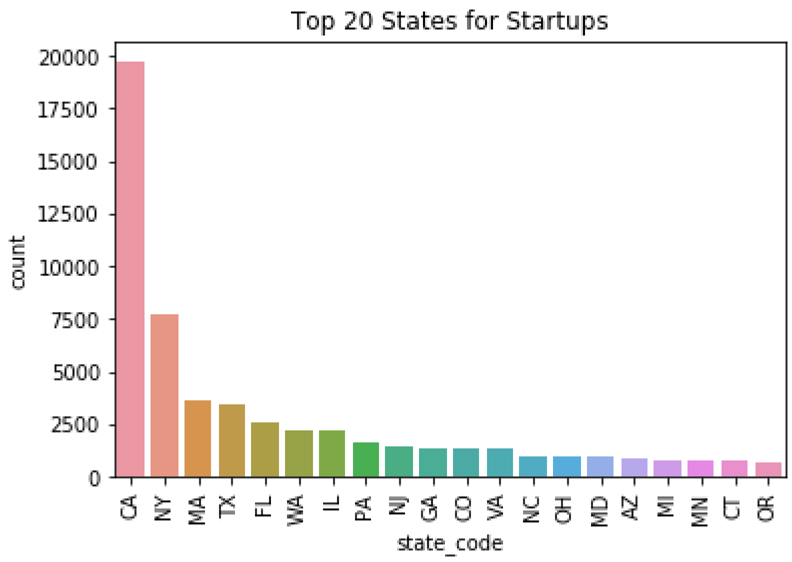


In [21]:

```
sns.countplot(data = df8,x="state_code",order=df8["state_code"].value_counts()[:20].index)
plt.xticks(rotation=90)
plt.title("Top 20 States for Startups")
```

Out[21]:

```
Text(0.5, 1.0, 'Top 20 States for Startups')
```

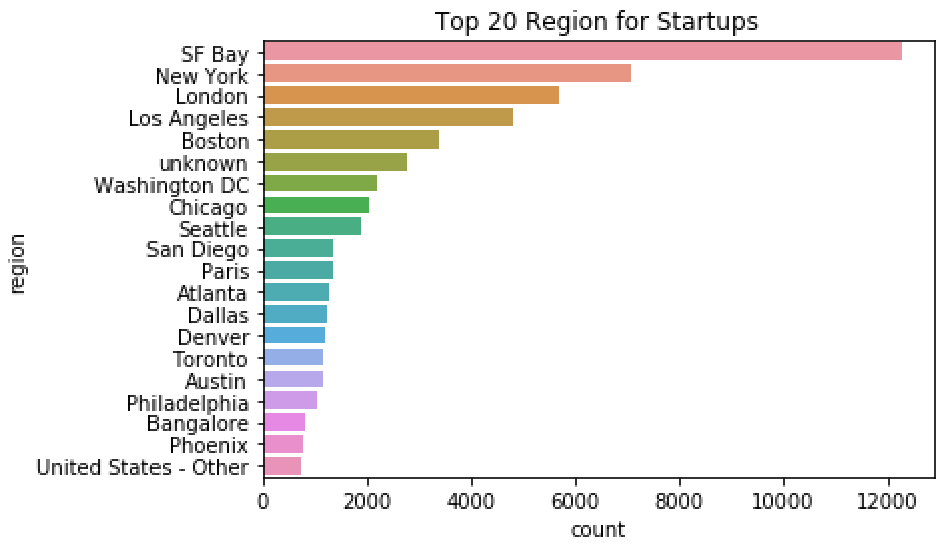


In [22]:

```
sns.countplot(data = df8,y="region",order=df8["region"].value_counts()[:20].index)
plt.title("Top 20 Region for Startups")
```

Out[22]:

```
Text(0.5, 1.0, 'Top 20 Region for Startups')
```

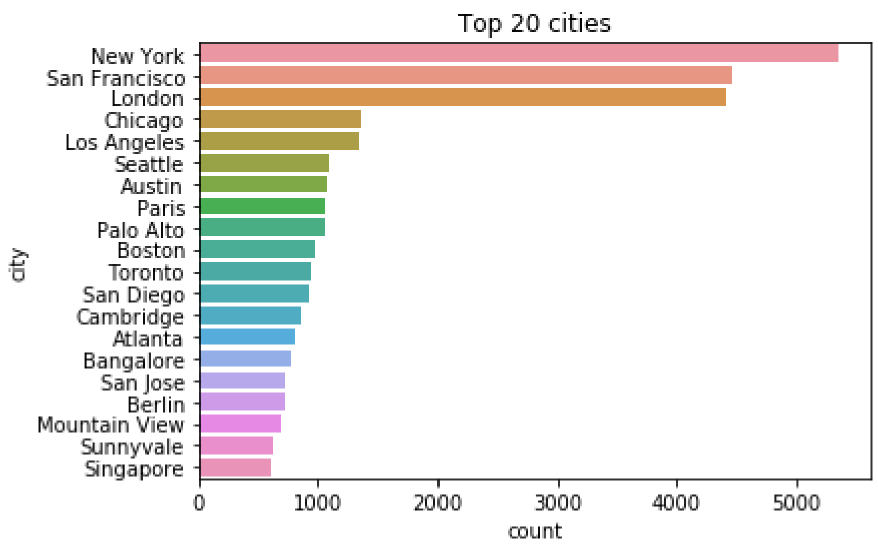


In [23]:

```
sns.countplot(data = df8,y="city",order=df8["city"].value_counts()[:20].index)
plt.title("Top 20 cities")
```

Out[23]:

```
Text(0.5, 1.0, 'Top 20 cities')
```

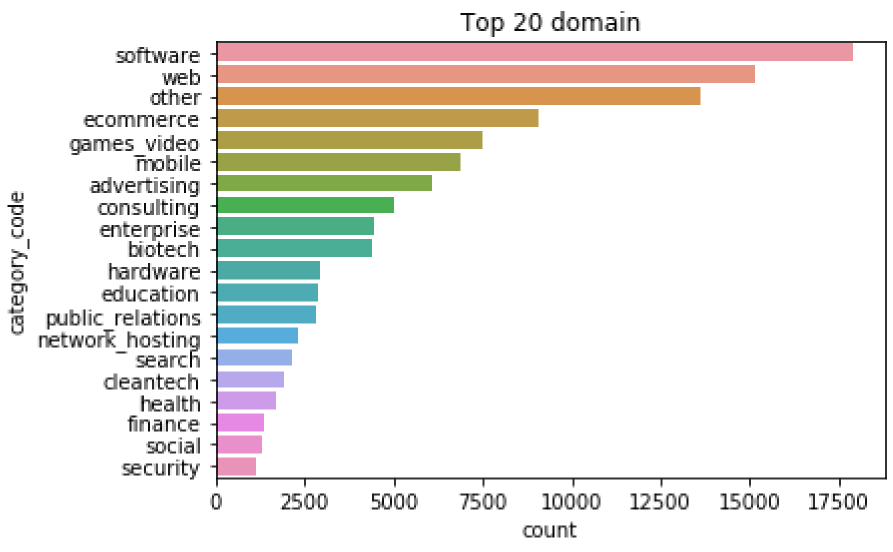


In [24]:

```
sns.countplot(data = df7,y="category_code",order=df7["category_code"].value_counts()[:20].i
plt.title("Top 20 domain")
```

Out[24]:

```
Text(0.5, 1.0, 'Top 20 domain')
```

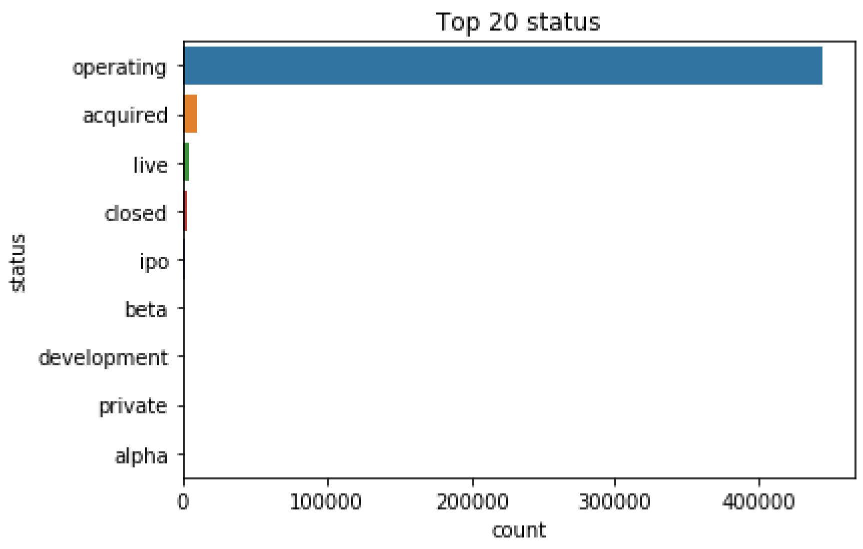


In [25]:

```
sns.countplot(data = df7,y="status",order=df7["status"].value_counts()[:20].index)
plt.title("Top 20 status")
```

Out[25]:

```
Text(0.5, 1.0, 'Top 20 status')
```

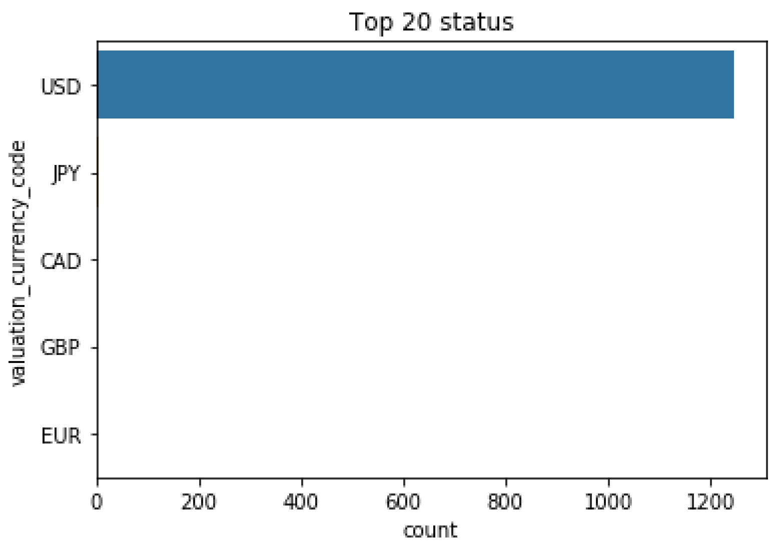


In [26]:

```
sns.countplot(data = df11,y="valuation_currency_code",order=df11["valuation_currency_code"]
plt.title("Top 20 status")
```

Out[26]:

```
Text(0.5, 1.0, 'Top 20 status')
```



In [ ]:

```

```