Pakistan Startup Census Analysis

In [ ]:

```python
import pandas as pd
import numpy as np
import re
```

In [ ]:

```python
df = pd.read_csv("Pakistan Startup Census.csv")
df.head()
```

Out[ ]:

	Name	Location	Tagline	Category	Website	Founded
0	Outnet	Karachi Pakistan	Cloud based SaaS platform for planning procur...	Advertising OOH Outdoor Advertising Data an...	http://www.outnet.com.pk	1st September 2014
1	7Vals	Lahore Pakistan	Understanding and enabling businesses to impro...	Consulting Rails Product Development UI	http://www.7vals.com	2011
2	92Solution	Lahore Pakistan	92 Solution A Pakistani Company Giving his Pa...	Consulting Website & Software Development	http://92solution.com	7th January 2015
3	A2Z Yellow Pages & Info Services	Lahore	Find local Businesses and Services in Pakistan...	Online Business Directory & Portal	https://www.yp-pak.com	30th January 2017
4	AALogics	Karachi Pakistan	NaN	Consulting Software	http://www.aalogics.com	1st August 2014

In [ ]:

```python
df.shape
```

Out[ ]:

```
(433, 7)
```

In [ ]:

```python
df.columns
```

Out[ ]:

```
Index(['Name', 'Location', 'Tagline', 'Category', 'Website', 'Founded',
       'Description'],
      dtype='object')
```

In [ ]:

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

Out[ ]:

```
Name            0
Location        0
Tagline         2
Category        0
Website        15
Founded         0
Description    43
dtype: int64
```

In [ ]:

```
df.duplicated()
```

Out[ ]:

```
0      False
1      False
2      False
3      False
4      False
       ...  
428    False
429    False
430    False
431    False
432    False
Length: 433, dtype: bool
```

In [ ]:

```
duplicates_in_column = df[df.duplicated(subset=['Name'], keep=False)]
duplicates_in_column
```

Out[ ]:

	Name	Location	Tagline	Category	Website	Founded
142	Goodshop.pk	Lahore Pakistan	Goodshop.pk offers Online Shopping in Pakistan...	E-Commerce	https://www.goodshop.pk	1st September 2015
143	Goodshop.pk	Lahore Pakistan	Goodshop.pk offers Online Shopping in Pakistan...	E-Commerce	http://www.goodshop.pk	14th August 2016

In [ ]:

```
# Drop duplicate rows based on the 'Name' column
df_cleaned = df.drop_duplicates(subset=['Name'], keep='last')
 #I am keeping the latest entry that is the last one to consider the correct one
```

In [ ]:

```
duplicates_in_column = df_cleaned[df_cleaned.duplicated(subset=['Name'], keep=False
duplicates_in_column
```

Out[ ]:

Name	Location	Tagline	Category	Website	Founded	Description

In [ ]:

```
df_cleaned.shape
```

```
Out[ ]: (432, 7)
```

```
In [ ]: df_cleaned.dtypes
```

```
Out[ ]: Name           object
        Location       object
        Tagline        object
        Category       object
        Website        object
        Founded        object
        Description    object
        dtype: object
```

```
In [ ]: df_sorted = df_cleaned.sort_values(by='Location')
        df_sorted.head()
```

Out[ ]:

	Name	Location	Tagline	Category	Website	Found
16	Assemblage	214 Block B 13D/2 Gulshan e Iqbal Karachi P...	Assemblage is an online Parenting Community an...	e-Magazine Website	www.assemblagekids.com	16th Ju 20
81	Credvestor	3rd Floor Citiview Naheed Supermarket Building...	An innovative peer-to-peer lending bazaar for ...	Software Finance	http://www.credvestor.com/	1st Augu 20
187	Kaanjo	Amsterdam	We help our clients give a voice to the silent...	Software	http:// kaanjo.co	16 Septemb 20
149	HAT incorporation	Block I North Nazimabad Karachi Pakistan	Hat inco is a software house and a computer in...	Software House Services	http://www.hatinco.com	Janua 20
104	Edev Technologies	Canada	Build great requirements together	Software Product Software Services Software ...	http://www.edevtech.com	19

```
In [ ]: df_sorted.Location.unique()
```

Out[ ]:

```
array(['214  Block B  13D/2 Gulshan e Iqbal Karachi  Pakistan ',
       '3rd Floor Citiview Naheed Supermarket Building Shaheed e Millat road Karac
hi  Pakistan',
       'Amsterdam', 'Block I North Nazimabad Karachi  Pakistan', 'Canada',
       'Dallas  Texas  United States', 'Dubai', 'Dubai  UAE',
       'Dubai  UAE - 1015 Arfa Software Technology Park Lahore Pakistan',
       'Faisalabad  Pakistan', 'Faisalabad Pakistan',
       'Go Logistics  Ground Floor  Palace Cinema Building  Civil Lines  Karachi',
       'Gujranwala  Pakistan', 'Gujrat  Pakistan', 'Gujrat Pakistan',
       'Hyderabad  Pakistan',
       'IBA Center for Entrepreneurial Development  Karachi  Pakistan',
       'Islamabad', 'Islamabad  PAkistan', 'Islamabad  Pakistan',
       'Johar Town  Lahore', 'Karachi', 'Karachi  Pakistan',
       'Karachi  Pakistan ', 'Karachi  Pakistan.', 'Karachi Pakistan',
       'Karachi Pakistan ', 'Karachi Â· Pakistan ', 'Lahore',
       'Lahore   Pakistan', 'Lahore  Manchester', 'Lahore  Pakistan',
       'Lahore  Pakistan ', 'Lahore  Punjab', 'Lahore  Punjab  Pakistan',
       'Lahore Pakistan', 'London  England', 'Los Angeles',
       'M-22  Al-Ameen Tower  Nipa Chowrangi  Karachi Pakistan.',
       'Main Market Gulberg  Lahore  Pakistan', 'Multan  Pakistan',
       'New York', 'Nowshera  Khyber Pakhtunkhwa',
       'Nowshera  Khyber Pakhtunkhwa  Pakistan', 'Pakistan',
       'Palo Alto  CA', 'Peshawar', 'Peshawar  Pakistan',
       'Quetta  Pakistan', 'Raleigh  North Carolina',
       'Rawalpindi  Pakistan', 'Rawalpindi/Islamabad',
       'Roosevelt Avenue Sunnyvale  CA 94085 United States',
       'San Francisco  California', 'Sargodha  Pakistan',
       'Seoul  South Korea', 'Sialkot  Pakistan', 'TIC  NUST  Islamabad',
       'Toronto  ON Canada', 'United States', 'Wah  Pakistan',
       'Wah Cantt  Pakistan', 'White Plains  New York  USA',
       'Z43/44 first floor darul aman society block 7/8 near hill park sharah e fa
isal Karachi  Pakistan',
       'http://www.shoplhr.com', 'karachi', 'karachi  Pakistan'],
      dtype=object)
```

By looking at the data we can see that We have t some Foreign regsitered startups also listed here, Lets seggerate our data first in Pakistan - regsitersd Startup and Foreign Registered Startups, so we are creating a country column in which we tagging our startups as Foreign and Pakistan

In [ ]:

```python
# Function to extract the country name
def extract_country(location):
    if 'Pakistan' in location:
        return 'Pakistan'
    else:
        return 'Foreign'
# df_sorted['Country'] = df_sorted['Location'].apply(lambda x: 'Pakistan' if 'Pakis
```

In [ ]:

```python
df_sorted['Country'] = df_sorted['Location'].apply(extract_country)
# Separate Pakistan-based and Foreign-based startups
pakistan_based = df_sorted[df_sorted['Country'] == 'Pakistan']
foreign_based = df_sorted[df_sorted['Country'] == 'Foreign']
```

In [ ]:

```
pakistan_based
```

Out[ ]:

	Name	Location	Tagline	Category	V
16	Assemblage	214 Block B 13D/2 Gulshan e Iqbal Karachi P...	Assemblage is an online Parenting Community an...	e-Magazine Website	www.assemblagek
81	Credvestor	3rd Floor Citiview Naheed Supermarket Building...	An innovative peer-to-peer lending bazaar for ...	Software Finance	http://www.credvest
149	HAT incorporation	Block I North Nazimabad Karachi Pakistan	Hat inco is a software house and a computer in...	Software House Services	http://www.hatir
163	ILMASOFT	Dubai UAE - 1015 Arfa Software Technology Par...	Educational and security products for schools ...	Software	http://www.ilmas
47	BrandsEgo.Com	Faisalabad Pakistan	Buy with confidence. We are Manufacturer and E...	E-Commerce	https://brandse
...	...	...	...	...	
371	The Books Yard	Sialkot Pakistan	The Books sell Books Online in Pakistan.	Education	https://www.facebook.com/thebo
201	Learn DAE	Wah Pakistan	Learn DAE is a Non-Profit Technical Educationa...	Technical Education	www.learnda
84	Daastan	Wah Cantt Pakistan	Daastan is a for-profit company working for re...	Literature Publishing Marketplace	http://www.daas
422	Pehnji	Z43/44 first floor darul aman society block 7/...	Online shopping from the Pakistan's best marke...	E-Commerce	http://www.peh

	Name	Location	Tagline	Category	\
239	My Mohalla	karachi Pakistan	My Mohalla is an organization that is working ...	Software	http://mymoh

378 rows × 8 columns

We also have found that we have incomplet Locaations in the column, which Only showing the country so we have to drop it because it will cause problem when we will be filltering data for the cities of Pakistan

Extracting Cities fromt the pakistan-based startups

```
In [ ]: def extract_city(location):
            parts = location.split()
            if 'Pakistan' in parts or 'Pakistan.' in parts:
                idx = parts.index('Pakistan') if 'Pakistan' in parts else parts.index('Paki
                if idx >= 0:
                    # Check if the word before 'Pakistan' is a province
                    if parts[idx - 1] in ['Punjab', 'Sindh', 'Balochistan']:
                        city = parts[idx - 2]
                    elif parts[idx - 1] == 'Pakhtunkhwa':
                        # Check if 'Khyber' is present directly before 'Pakhtunkhwa'
                        if parts[idx - 2] == 'Khyber':
                            # Extract the city name before 'Khyber'
                            city = parts[idx - 3]
                        else:
                            city = parts[idx - 2]
                    else:
                        city = parts[idx - 1]
                else:
                    city = None
            else:
                city = None
            return city

        # Apply the function to create a new column 'City'
        pakistan_based['City'] = pakistan_based['Location'].apply(extract_city)

        pakistan_based
```

```
C:\Users\NimZee\AppData\Local\Temp\ipykernel_16672\113054989.py:25: SettingWithCopyW
arning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  pakistan_based['City'] = pakistan_based['Location'].apply(extract_city)
```

Out[ ]:

	Name	Location	Tagline	Category	W
16	Assemblage	214 Block B 13D/2 Gulshan e Iqbal Karachi P...	Assemblage is an online Parenting Community an...	e-Magazine Website	www.assemblagek
81	Credvestor	3rd Floor Citiview Naheed Supermarket Building...	An innovative peer-to-peer lending bazaar for ...	Software Finance	http://www.credvest
149	HAT incorporation	Block I North Nazimabad Karachi Pakistan	Hat inco is a software house and a computer in...	Software House Services	http://www.hatir
163	ILMASOFT	Dubai UAE - 1015 Arfa Software Technology Par...	Educational and security products for schools ...	Software	http://www.ilmas
47	BrandsEgo.Com	Faisalabad Pakistan	Buy with confidence. We are Manufacturer and E...	E-Commerce	https://brandse
...	...	...	...	...	
371	The Books Yard	Sialkot Pakistan	The Books sell Books Online in Pakistan.	Education	https://www.facebook.com/thebo
201	Learn DAE	Wah Pakistan	Learn DAE is a Non-Profit Technical Educationa...	Technical Education	www.learnda
84	Daastan	Wah Cantt Pakistan	Daastan is a for-profit company working for re...	Literature Publishing Marketplace	http://www.daas
422	Pehnji	Z43/44 first floor darul aman society block 7/...	Online shopping from the Pakistan's best marke...	E-Commerce	http://www.peh

	Name	Location	Tagline	Category	\
239	My Mohalla	karachi Pakistan	My Mohalla is an organization that is working ...	Software	http://mymoh

378 rows × 9 columns

```
In [ ]: pakistan_based['City'] = pakistan_based['Location'].apply(extract_city)
        pakistan_based
```

```
C:\Users\NimZee\AppData\Local\Temp\ipykernel_16672\2232085017.py:1: SettingWithCopyW
arning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  pakistan_based['City'] = pakistan_based['Location'].apply(extract_city)
```

Out[ ]:

	Name	Location	Tagline	Category	W
16	Assemblage	214 Block B 13D/2 Gulshan e Iqbal Karachi P...	Assemblage is an online Parenting Community an...	e-Magazine Website	www.assemblagek
81	Credvestor	3rd Floor Citiview Naheed Supermarket Building...	An innovative peer-to-peer lending bazaar for ...	Software Finance	http://www.credvest
149	HAT incorporation	Block I North Nazimabad Karachi Pakistan	Hat inco is a software house and a computer in...	Software House Services	http://www.hatir
163	ILMASOFT	Dubai UAE - 1015 Arfa Software Technology Par...	Educational and security products for schools ...	Software	http://www.ilmas
47	BrandsEgo.Com	Faisalabad Pakistan	Buy with confidence. We are Manufacturer and E...	E-Commerce	https://brandse
...	...	...	...	...	
371	The Books Yard	Sialkot Pakistan	The Books sell Books Online in Pakistan.	Education	https://www.facebook.com/thebo
201	Learn DAE	Wah Pakistan	Learn DAE is a Non-Profit Technical Educationa...	Technical Education	www.learnda
84	Daastan	Wah Cantt Pakistan	Daastan is a for-profit company working for re...	Literature Publishing Marketplace	http://www.daas
422	Pehnji	Z43/44 first floor darul aman society block 7/...	Online shopping from the Pakistan's best marke...	E-Commerce	http://www.peh

	Name	Location	Tagline	Category	W
239	My Mohalla	karachi Pakistan	My Mohalla is an organization that is working ...	Software	http://mymoh

378 rows × 9 columns

In [ ]:
```
# Apply the function to create a new column 'City'
pakistan_based.City.unique()
```

Out[ ]:
```
array(['Karachi', 'Lahore', 'Faisalabad', 'Gujranwala', 'Gujrat',
       'Hyderabad', 'Islamabad', 'Â·', 'Multan', 'Nowshera', 'Pakistan',
       'Peshawar', 'Quetta', 'Rawalpindi', 'Sargodha', 'Sialkot', 'Wah',
       'Cantt', 'karachi'], dtype=object)
```

We still have some issues Like None and repeatation of Karachi Wah and Wah Cantt so lets just check none first and this A and then will make it proper.

In [ ]:
```
# Filter the DataFrame to show rows where 'City' is None
none_cities = pakistan_based[pakistan_based['City'].isnull()]
# Print the rows with None values in the 'City' column
none_cities
```

Out[ ]:

Name	Location	Tagline	Category	Website	Founded	Description	Country	City

In [ ]:
```
# Assuming you want to filter rows containing the word 'Karachi' in the 'Location'
filtered_rows = pakistan_based[pakistan_based['City'].str.contains('Â·', case=False

# Print the filtered rows
filtered_rows
```

Out[ ]:

	Name	Location	Tagline	Category	Website	Founded	Descri
21	AutoExpert	Karachi Â· Pakistan	Preemptive Car Care At your Door	Mobile Â· E-Commerce Â· Automotive Â· Mobile C...	http://www.autoexpert.pk	2015	Our g to e expect fo

Its a data entry error so we know its Karachi so we can directly consider it.

In [ ]:
```
# Replace the 'City' column value with 'Karachi' for the filtered rows
pakistan_based.loc[filtered_rows.index, 'City'] = 'Karachi'
```

```
# Print the DataFrame after the replacement
pakistan_based.City.unique()
```

```
C:\Users\NimZee\AppData\Local\Temp\ipykernel_16672\1469899815.py:2: SettingWithCopyW
arning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  pakistan_based.loc[filtered_rows.index, 'City'] = 'Karachi'
```

```
Out[ ]: array(['Karachi', 'Lahore', 'Faisalabad', 'Gujranwala', 'Gujrat',
               'Hyderabad', 'Islamabad', 'Multan', 'Nowshera', 'Pakistan',
               'Peshawar', 'Quetta', 'Rawalpindi', 'Sargodha', 'Sialkot', 'Wah',
               'Cantt', 'karachi'], dtype=object)
```

Its looking much better noww but we still have some to sort, Like Drop rows which have City name as Pakistan since City is not provided., make Wah and cantt as Wah cantt, and varation of Karchi as small and capital letter to Karachi.

```
In [ ]: # Drop rows where 'City' is 'Pakistan'

        #Keeping it for lateruse foreign and Pakistan comaprasion

        none_city = pakistan_based[pakistan_based['City'] == 'Pakistan']

        pakistan_based = pakistan_based[pakistan_based['City'] != 'Pakistan']

        # Replace 'Wah' with 'Wah Cantt'
        pakistan_based['City'] = pakistan_based['City'].replace('Wah', 'Wah Cantt')

        # Replace 'Cantt' with 'Wah Cantt'
        pakistan_based['City'] = pakistan_based['City'].replace('Cantt', 'Wah Cantt')

        # Replace variations of 'Karachi' with 'Karachi'
        pakistan_based['City'] = pakistan_based['City'].replace(['karachi', 'KARACHI'], 'Ka

        pakistan_based.City.unique()
```

```
C:\Users\NimZee\AppData\Local\Temp\ipykernel_16672\2765083002.py:10: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  pakistan_based['City'] = pakistan_based['City'].replace('Wah', 'Wah Cantt')
C:\Users\NimZee\AppData\Local\Temp\ipykernel_16672\2765083002.py:13: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  pakistan_based['City'] = pakistan_based['City'].replace('Cantt', 'Wah Cantt')
C:\Users\NimZee\AppData\Local\Temp\ipykernel_16672\2765083002.py:16: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  pakistan_based['City'] = pakistan_based['City'].replace(['karachi', 'KARACHI'], 'K
arachi')
```

Out[ ]:

```
array(['Karachi', 'Lahore', 'Faisalabad', 'Gujranwala', 'Gujrat',
       'Hyderabad', 'Islamabad', 'Multan', 'Nowshera', 'Peshawar',
       'Quetta', 'Rawalpindi', 'Sargodha', 'Sialkot', 'Wah Cantt'],
      dtype=object)
```

In [ ]:

```
foreign_based.head()
```

Out[ ]:

	Name	Location	Tagline	Category	Website	Founded
187	Kaanjo	Amsterdam	We help our clients give a voice to the silent...	Software	http:// kaanjo.co	16th September 2016
104	Edev Technologies	Canada	Build great requirements together	Software Product Software Services Software ...	http://www.edevtech.com	1999
95	Dimensional Sys	Dallas Texas United States	Dimensional Systems is an IT services provider...	Consulting Software	http://dimensionalsys.com	1st January 2012
183	Jouple FZ LLC	Dubai	Delivering real-world digital solutions in Mob...	Software	http://www.jouple.com	15th February 2015
55	Careem	Dubai	Careem is a chauffeur cab booking service avai...	Sharing Economy	http://careem.com/	15th June 2012

any occurrence of a city name from the 'pakistan_based' DataFrame in the 'Location' column of the 'foreign_based' DataFrame will be captured, regardless of the case.

In [ ]:

```
# Create a regex pattern to match any word from the 'City' column of 'pakistan_base
city_pattern = '|'.join(pakistan_based['City'].str.lower())

# Filter the 'foreign_based' DataFrame based on the 'Location' containing any word
pakistan_based_foreign = foreign_based[foreign_based['Location'].str.lower().str.co

# Print the DataFrame containing Pakistan-based startups from 'foreign_based'
pakistan_based_foreign.Location.unique()
```

Out[ ]:

```
array(['Go Logistics  Ground Floor  Palace Cinema Building  Civil Lines  Karachi',
       'Islamabad', 'Islamabad  PAkistan', 'Johar Town  Lahore',
       'Karachi', 'Lahore', 'Lahore  Manchester', 'Lahore  Punjab',
       'Nowshera  Khyber Pakhtunkhwa', 'Peshawar', 'Rawalpindi/Islamabad',
       'TIC  NUST  Islamabad', 'karachi'], dtype=object)
```

In [ ]:

```
# Function to find matching city from pakistan_based['City']
def find_matching_city(location):
    for city in pakistan_based['City']:
        # Check if any word from 'City' matches any word in 'Location'
        if any(re.search(r'\b{}\b'.format(re.escape(city.lower())), word.lower()) f
                return city
        return None

# Create 'City' column in pakistan_based_foreign and assign matching city from paki
pakistan_based_foreign['City'] = pakistan_based_foreign['Location'].apply(find_matc

# Print the updated DataFrame
pakistan_based_foreign.City.unique()
```

```
C:\Users\NimZee\AppData\Local\Temp\ipykernel_16672\2970353797.py:10: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  pakistan_based_foreign['City'] = pakistan_based_foreign['Location'].apply(find_mat
ching_city)
```

Out[ ]:

```
array(['Karachi', 'Islamabad', 'Lahore', 'Nowshera', 'Peshawar'],
      dtype=object)
```

In [ ]:

```
# Concatenate pakistan_based_foreign and pakistan_based DataFrames
Pakistan_startups = pd.concat([pakistan_based_foreign, pakistan_based])

# Reset index of the merged DataFrame
Pakistan_startups.reset_index(drop=True, inplace=True)

# Print the updated DataFrame
Pakistan_startups.head()
```

Out[ ]:

	Name	Location	Tagline	Category	Website	Founded
0	Go Rickshaw	Go Logistics Ground Floor Palace Cinema Buil...	GO Rickshaw redefines the entire idea of on-de...	Transportation	https://gorickshaw.pk/	30th October 2015
1	DexterED	Islamabad	Dextered believes in Creative & Automated Asse...	Education	http://dextered.com	23rd March 2015
2	TopSchools.pk	Islamabad	Top Schools is a platform for all schools col...	Web Portal	http://www.topschools.pk	10th January 2014
3	Jobz.pk	Islamabad	Latest Jobs in Pakistan where one can find new...	Job Portal	http://jobz.pk	2000
4	ContentStudio	Islamabad	The easiest way to discover monitor and share...	Software	https://contentstudio.io	2016

In [ ]:

```python
# Update all rows in 'Country' column to 'Pakistan'
Pakistan_startups['Country'] = 'Pakistan'

# Print the updated DataFrame
Pakistan_startups.head()
```

Out[ ]:

	Name	Location	Tagline	Category	Website	Founded
0	Go Rickshaw	Go Logistics Ground Floor Palace Cinema Buil...	GO Rickshaw redefines the entire idea of on-de...	Transportation	https://gorickshaw.pk/	30th October 2015
1	DexterED	Islamabad	Dextered believes in Creative & Automated Asse...	Education	http://dextered.com	23rd March 2015
2	TopSchools.pk	Islamabad	Top Schools is a platform for all schools col...	Web Portal	http://www.topschools.pk	10th January 2014
3	Jobz.pk	Islamabad	Latest Jobs in Pakistan where one can find new...	Job Portal	http://jobz.pk	2000
4	ContentStudio	Islamabad	The easiest way to discover monitor and share...	Software	https://contentstudio.io	2016

In [ ]:

```
# Merge the two DataFrames on 'Name' and 'Location', using an indicator flag
merged = foreign_based.merge(Pakistan_startups[['Name', 'Location']], on=['Name', '

# Filter out the rows present in both DataFrames
foreign_startups = merged[merged['_merge'] == 'left_only']

# Drop the indicator column
foreign_startups.drop('_merge', axis=1, inplace=True)

# Print the filtered foreign startups DataFrame
foreign_startups.head()
```

```
C:\Users\NimZee\AppData\Local\Temp\ipykernel_16672\1361341757.py:8: SettingWithCopyW
arning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  foreign_startups.drop('_merge', axis=1, inplace=True)
```

Out[ ]:

	Name	Location	Tagline	Category	Website	Founded
0	Kaanjo	Amsterdam	We help our clients give a voice to the silent...	Software	http:// kaanjo.co	16th September 2016
1	Edev Technologies	Canada	Build great requirements together	Software Product Software Services Software ...	http://www.edevtech.com	1999
2	Dimensional Sys	Dallas Texas United States	Dimensional Systems is an IT services provider...	Consulting Software	http://dimensionalsys.com	1st January 2012
3	Jouple FZ LLC	Dubai	Delivering real-world digital solutions in Mob...	Software	http://www.jouple.com	15th February 2015
4	Careem	Dubai	Careem is a chauffeur cab booking service avai...	Sharing Economy	http://careem.com/	15th June 2012

In [ ]:

```
# Define a regular expression pattern to match web addresses
web_address_pattern = r'http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\\(\\),]|(?:%[0-9

# Filter rows where 'Location' column contains a web address
rows_with_web_address = foreign_startups[foreign_startups['Location'].str.contains(

# Print rows containing a web address
print("Rows containing web addresses:")
for index, row in rows_with_web_address.iterrows():
    print(row)
```

```
Rows containing web addresses:
Name                                                      Shoplhr
Location                                    http://www.shoplhr.com
Tagline        First online grocery and food delivery service...
Category                                           Online Shopping
Website                                     http://www.shoplhr.com
Founded                                           1st April  2017
Description    If jewellery  clothing  beauty products and ap...
Country                                                    Foreign
Name: 52, dtype: object
```

noww our data is quite clean and contain the city name

In [ ]:

```
# Drop rows containing a web address
foreign_startups = foreign_startups.drop(rows_with_web_address.index)

print("\nUpdated DataFrame after dropping rows with web addresses:")
foreign_startups.head()
```

```
Updated DataFrame after dropping rows with web addresses:
```

Out[ ]:

	Name	Location	Tagline	Category	Website	Founded
0	Kaanjo	Amsterdam	We help our clients give a voice to the silent...	Software	http:// kaanjo.co	16th September 2016
1	Edev Technologies	Canada	Build great requirements together	Software Product Software Services Software ...	http://www.edevtech.com	1999
2	Dimensional Sys	Dallas Texas United States	Dimensional Systems is an IT services provider...	Consulting Software	http://dimensionalsys.com	1st January 2012
3	Jouple FZ LLC	Dubai	Delivering real-world digital solutions in Mob...	Software	http://www.jouple.com	15th February 2015
4	Careem	Dubai	Careem is a chauffeur cab booking service avai...	Sharing Economy	http://careem.com/	15th June 2012

Now we have two properly cleaned Datasets Foreign_startups and Pakistan Startups Lets analysze and visuzlie it

In [ ]:

```
#Merging if requires
Startups = pd.concat([Pakistan_startups, foreign_startups])

# Reset index of the merged DataFrame
Startups.reset_index(drop=True, inplace=True)

# Print the updated DataFrame
Startups.head()
```

Out[ ]:

	Name	Location	Tagline	Category	Website	Founded
0	Go Rickshaw	Go Logistics Ground Floor Palace Cinema Buil...	GO Rickshaw redefines the entire idea of on-de...	Transportation	https://gorickshaw.pk/	30th October 2015
1	DexterED	Islamabad	Dextered believes in Creative & Automated Asse...	Education	http://dextered.com	23rd March 2015
2	TopSchools.pk	Islamabad	Top Schools is a platform for all schools col...	Web Portal	http://www.topschools.pk	10th January 2014
3	Jobz.pk	Islamabad	Latest Jobs in Pakistan where one can find new...	Job Portal	http://jobz.pk	2000
4	ContentStudio	Islamabad	The easiest way to discover monitor and share...	Software	https://contentstudio.io	2016

In [ ]:

```
# Define a regular expression pattern to match years
year_pattern = r'(\b\d{4}\b)'

# Extract founded years using regular expressions
Startups['Founded Year'] = Startups['Founded'].str.extract(year_pattern)
Startups.head(2)
```

Out[ ]:

	Name	Location	Tagline	Category	Website	Founded	Descripti
0	Go Rickshaw	Go Logistics Ground Floor Palace Cinema Buil...	GO Rickshaw redefines the entire idea of on-de...	Transportation	https://gorickshaw.pk/	30th October 2015	N
1	DexterED	Islamabad	Dextered believes in Creative & Automated Asse...	Education	http://dextered.com	23rd March 2015	At Dexte we commit to not c rev

We have issue that we are not provided with the complete year so we are endeed with the null or not provided values

In [ ]:

```
# Check if there are no null values in the 'Founded_Year' column
no_null_founded_year = not Startups['Founded Year'].isnull().any()

# Filter rows where 'Founded_Year' is null
null_founded_year_rows = Startups[Startups['Founded Year'].isnull()]

# Replace null values in 'Founded_Year' with 'Not Provided'
Startups['Founded Year'].fillna('Not Provided', inplace=True)
```

In [ ]:

```
# Count the unique categories and their frequencies
category_counts = Startups['Category'].value_counts()

# Print the unique categories and their frequencies
print(category_counts)
```

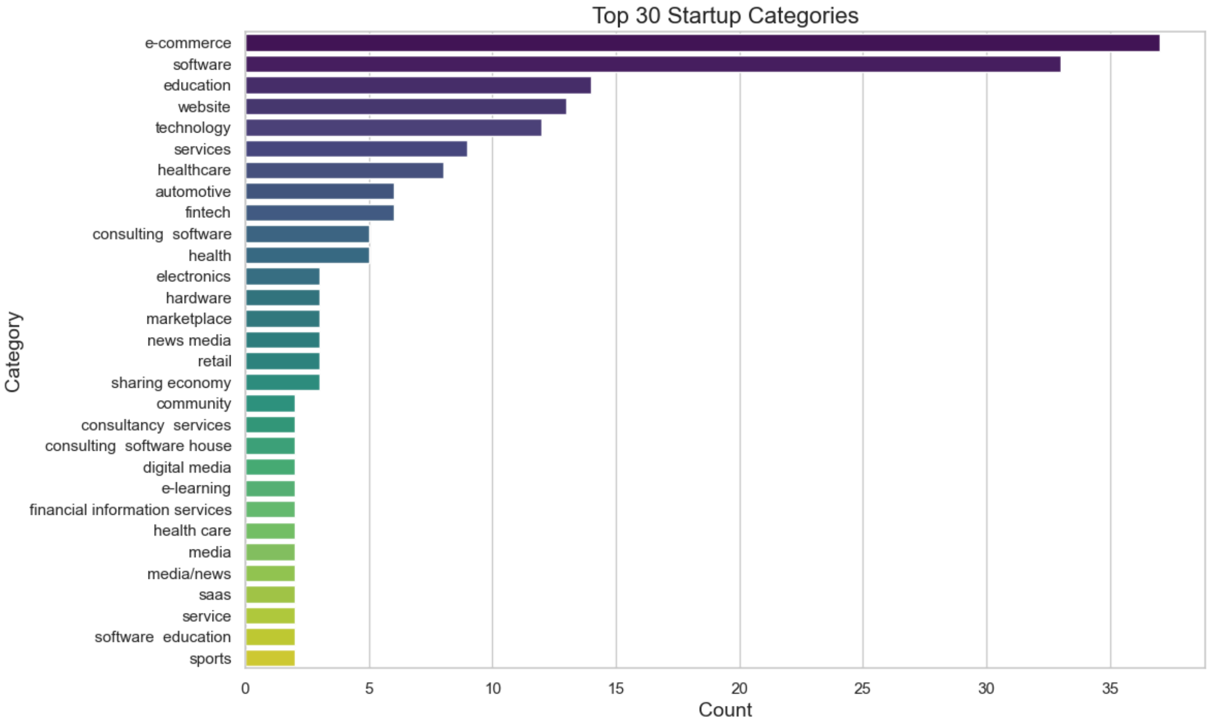
```
Software                                                             31
E-Commerce                                                           28
Education                                                            14
Technology                                                           12
Website                                                              12
                                                                     ..
Discount Store                                                        1
SMS Interaction  Tech                                                 1
Consulting  Software  Application Development  Branding  Web Hosting  1
E-commerce  Online Shopping                                           1
Consulting  Computer Aided Drafting  3D                               1
Name: Category, Length: 276, dtype: int64
```

In [ ]:

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

Startups.Category = Startups.Category.str.lower().str.strip()
```

```
cat_wise = Startups.groupby(Startups.Category).size().nlargest(30)
cat_wise_df = cat_wise.reset_index()
cat_wise_df.columns = ['Category', 'Count']

# Set the style
sns.set(style="whitegrid")

# Create the bar plot
plt.figure(figsize=(12, 8))
sns.barplot(x='Count', y='Category', data=cat_wise_df, palette='viridis')

# Set labels and title
plt.xlabel('Count', fontsize=14)
plt.ylabel('Category', fontsize=14)
plt.title('Top 30 Startup Categories', fontsize=16)

# Show the plot
plt.show()
```

```
C:\Users\NimZee\AppData\Local\Temp\ipykernel_16672\658080512.py:14: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x='Count', y='Category', data=cat_wise_df, palette='viridis')
```



In [ ]:

```
import requests

# Define a DataFrame to store city coordinates
city_coordinates_df = pd.DataFrame(columns=['City', 'Latitude', 'Longitude'])


# Function to get latitude and longitude coordinates using OpenStreetMap Nominatim
def get_coordinates(city):
    # Check if coordinates for the city are already available in the DataFrame
    if city in city_coordinates_df['City'].values:
        return city_coordinates_df.loc[city_coordinates_df['City'] == city, ['Latit
    else:
        # Fetch coordinates from the API
        url = f'https://nominatim.openstreetmap.org/search?city={city}&format=json'
        response = requests.get(url).json()
        if response:
            # Extract latitude and longitude from the response
            latitude = float(response[0]['lat'])
            longitude = float(response[0]['lon'])
            # Add coordinates to the DataFrame
            city_coordinates_df.loc[len(city_coordinates_df)] = [city, latitude, lo
            return latitude, longitude
        else:
            return None, None

# Apply the function to create new columns
#
#
# for latitude and longitude in the Startups DataFrame
Startups['Latitude'], Startups['Longitude'] = zip(*Startups['City'].apply(get_coord
```

In [ ]:
```
Startups.head(3)
```

Out[ ]:

	Name	Location	Tagline	Category	Website	Founded	D
0	Go Rickshaw	Go Logistics Ground Floor Palace Cinema Buil...	GO Rickshaw redefines the entire idea of on-de...	transportation	https://gorickshaw.pk/	30th October 2015	
1	DexterED	Islamabad	Dextered believes in Creative & Automated Asse...	education	http://dextered.com	23rd March 2015	A
2	TopSchools.pk	Islamabad	Top Schools is a platform for all schools col...	web portal	http://www.topschools.pk	10th January 2014	-

In [ ]:
```
import folium
from folium.plugins import HeatMap
import pandas as pd

# Group startups by city and count the number of startups in each city
```

```
startup_counts = Startups['City'].value_counts()

# Create a DataFrame with city and corresponding startup count
city_counts_df = pd.DataFrame({'City': startup_counts.index, 'Count': startup_count

# Get the latitude and longitude of each city
from geopy.geocoders import Nominatim

geolocator = Nominatim(user_agent="startup_heatmap")
city_counts_df['location'] = city_counts_df['City'].apply(lambda x: geolocator.geoc
city_counts_df = city_counts_df.dropna()

# Create a Folium map centered around Pakistan
map_pakistan = folium.Map(location=[30.3753, 69.3451], zoom_start=6)

# Create a HeatMap layer
heat_data = [[row['location'].latitude, row['location'].longitude, row['Count']] fo
HeatMap(heat_data, radius=15).add_to(map_pakistan)

# Display the map
map_pakistan
```

Out[ ]:

In [ ]:

```
import folium
from folium.plugins import MarkerCluster
import pandas as pd

# Group startups by city and count the number of startups in each city
startup_counts = Startups['City'].value_counts()

# Create a DataFrame with city and corresponding startup count
city_counts_df = pd.DataFrame({'City': startup_counts.index, 'Count': startup_count

# Get the latitude and longitude of each city
from geopy.geocoders import Nominatim
```

```
geolocator = Nominatim(user_agent="startup_heatmap")
city_counts_df['location'] = city_counts_df['City'].apply(lambda x: geolocator.geoc
city_counts_df = city_counts_df.dropna()

# Create a Folium map centered around Pakistan
map_pakistan3 = folium.Map(location=[30.3753, 69.3451], zoom_start=6)

# Create a MarkerCluster layer
marker_cluster = MarkerCluster().add_to(map_pakistan3)

# Add markers to the MarkerCluster layer
for index, row in city_counts_df.iterrows():
    folium.Marker(
        location=[row['location'].latitude, row['location'].longitude],
        popup=f"{row['City']}: {row['Count']} startups",
        icon=None,
    ).add_to(marker_cluster)

# Display the map
map_pakistan3
```

Out[ ]:

In [ ]:

```
import folium
import json
import requests
import os

# Function to fetch GeoJSON data for a city
def get_geojson(city_name):
    url = f"https://nominatim.openstreetmap.org/search?city={city_name}&country=Pak
    response = requests.get(url)
    if response.status_code == 200:
        return response.json()
    else:
        print(f"Failed to fetch GeoJSON for {city_name}")
        return None
```

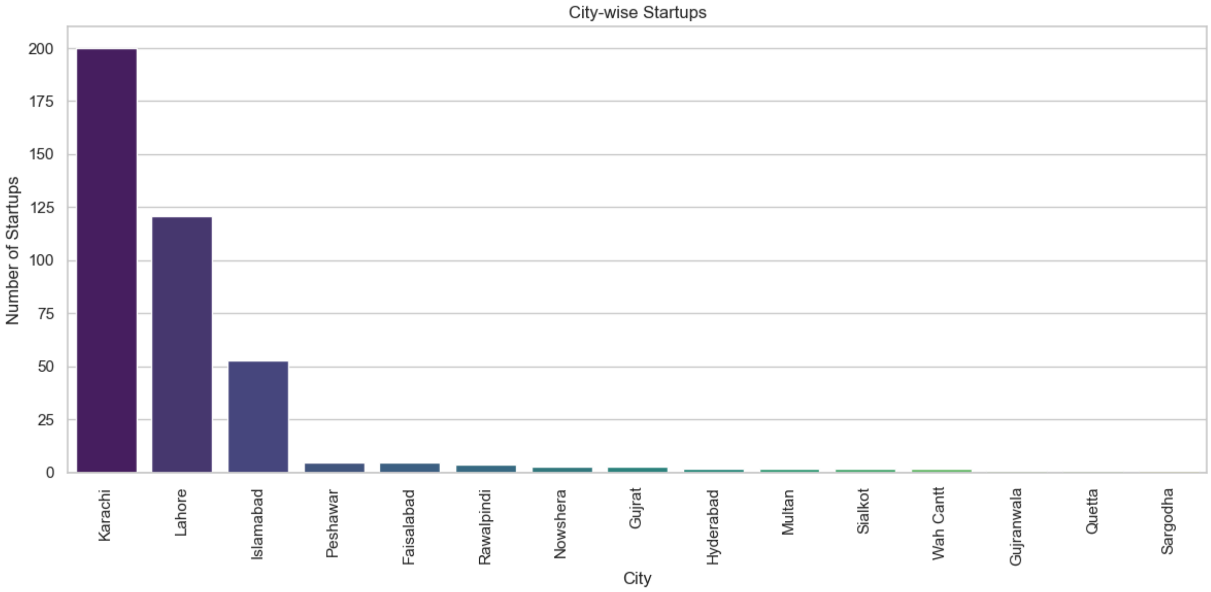
```
# Fetch unique city names from the 'City' column
city_names = Startups["City"].unique()

# Create a directory to save GeoJSON files
geojson_dir = "geojson_files"
os.makedirs(geojson_dir, exist_ok=True)

# Iterate over the city names and fetch/save their GeoJSON boundaries
city_geojson_map = {}
for city_name in city_names:
    geojson_data = get_geojson(city_name)
    if geojson_data:
        city_geojson_map[city_name] = geojson_data
        # Save GeoJSON data to a file
        with open(f"{geojson_dir}/{city_name}.geojson", "w", encoding="utf-8") as f
            json.dump(geojson_data, f)

# Create a Folium map centered around Pakistan
map_pakistan1 = folium.Map(location=[30.3753, 69.3451], zoom_start=6)

# Iterate over city names and add Choropleth layer to the map
for city_name, geojson_data in city_geojson_map.items():
    folium.Choropleth(
        geo_data=geojson_data,
        fill_opacity=0.7,
        line_opacity=0.2,
        name=city_name,
    ).add_to(map_pakistan1)

# Save the map as an HTML file
map_pakistan1.save("startup_counts_map.html")

# Display the map
map_pakistan1
```

Out[ ]:

In [ ]:

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

# Sort the DataFrame by startup count in descending order
city_counts_df_sorted = city_counts_df.sort_values(by='Count', ascending=False)

# Create the bar plot using Seaborn
plt.figure(figsize=(12, 6))
sns.barplot(x='City', y='Count', data=city_counts_df_sorted, palette='viridis')
plt.xlabel('City')
plt.ylabel('Number of Startups')
plt.title('City-wise Startups')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

```
C:\Users\NimZee\AppData\Local\Temp\ipykernel_16672\2012555394.py:9: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x='City', y='Count', data=city_counts_df_sorted, palette='viridis')
```



In [ ]:

```
# Sort the DataFrame by startup count in descending order
city_counts_df_sorted = city_counts_df.sort_values(by='Count', ascending=False)

# Create the bar plot using Plotly
fig = px.bar(city_counts_df_sorted, x='City', y='Count', title='City-wise Startups'
fig.update_layout(xaxis={'categoryorder':'total descending'}, xaxis_title='City', y
fig.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[90], line 5
      2 city_counts_df_sorted = city_counts_df.sort_values(by='Count', ascending=Fal
se)
      4 # Create the bar plot using Plotly
----> 5 fig = px.bar(city_counts_df_sorted, x='City', y='Count', title='City-wise St
artups', labels={'Count': 'Number of Startups', 'City': 'City'}, color='City')
      6 fig.update_layout(xaxis={'categoryorder':'total descending'}, xaxis_title='C
ity', yaxis_title='Number of Startups')
      7 fig.show()

NameError: name 'px' is not defined
```

In [ ]:

```
import plotly.express as px

# Filter out rows with 'Not Provided' in the 'Founded Year' column
startup_year_counts = Startups[Startups['Founded Year'] != 'Not Provided']

# Group startups by founded year and count the number of startups in each year
startup_year_counts = startup_year_counts['Founded Year'].value_counts().reset_inde
startup_year_counts.columns = ['Founded Year', 'Count']

# Sort startup_year_counts DataFrame by 'Founded Year' column in ascending order
startup_year_counts = startup_year_counts.sort_values(by='Founded Year', ascending=

# Create the Plotly graph
fig = px.bar(startup_year_counts, x='Founded Year', y='Count', title='Number of Sta
            labels={'Count': 'Number of Startups', 'Founded Year': 'Founded Year'},
            color='Founded Year', color_continuous_scale=px.colors.sequential.Virid

fig.show()
```

In [ ]:

```
import plotly.express as px
import pandas as pd

# Filter out rows with 'Not Provided' in the 'Founded Year' column
filtered_startups = Startups[Startups['Founded Year'] != 'Not Provided']

# Convert 'Founded Year' column to numeric (assuming it's a numerical column)
filtered_startups['Founded Year'] = pd.to_numeric(filtered_startups['Founded Year']

# Drop rows with NaN values in 'Founded Year' column
filtered_startups = filtered_startups.dropna(subset=['Founded Year'])

# Group startups by founded year and count the number of startups in each year
startup_year_counts = filtered_startups.groupby(['Founded Year', 'Category']).size(

# Sort startup_year_counts DataFrame by 'Count' column in descending order and sele
top_20_categories = startup_year_counts.groupby('Category').sum().sort_values(by='C

# Filter startup_year_counts to include only the top 20 categories
startup_year_counts = startup_year_counts[startup_year_counts['Category'].isin(top_

# Sort startup_year_counts DataFrame by 'Founded Year' column in ascending order
startup_year_counts = startup_year_counts.sort_values(by='Founded Year', ascending=

# Create the Plotly bubble plot
fig = px.scatter(startup_year_counts, x='Founded Year', y='Count', size='Count',
                 title='Number of Startups by Founded Year and Category',
                 labels={'Count': 'Number of Startups', 'Founded Year': 'Founded Ye
                 color='Category', color_discrete_sequence=px.colors.qualitative.Pa

# Add a range slider for selecting the range of years
fig.update_layout(
    xaxis=dict(
        rangeslider=dict(
            visible=True
        ),
        type='linear'  # Set the type of x-axis to linear
    )
)
fig.show()
```

```
C:\Users\NimZee\AppData\Local\Temp\ipykernel_21236\110923047.py:8: SettingWithCopyWa
rning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
```

In [ ]:

```
import plotly.express as px
import pandas as pd

# Filter out rows with 'Not Provided' in the 'Founded Year' column
filtered_startups = Startups[Startups['Founded Year'] != 'Not Provided']

# Convert 'Founded Year' column to numeric (assuming it's a numerical column)
filtered_startups['Founded Year'] = pd.to_numeric(filtered_startups['Founded Year']

# Drop rows with NaN values in 'Founded Year' column
filtered_startups = filtered_startups.dropna(subset=['Founded Year'])

# Group startups by founded year and count the number of startups in each year
startup_year_counts = filtered_startups.groupby('Founded Year').size().reset_index(

# Sort startup_year_counts DataFrame by 'Founded Year' column in ascending order
startup_year_counts = startup_year_counts.sort_values(by='Founded Year', ascending=

# Filter the data for the years from 2000 to 2020
startup_year_counts = startup_year_counts[(startup_year_counts['Founded Year'] >= 2

# Create the Plotly line graph
fig = px.line(startup_year_counts, x='Founded Year', y='Count',
              title='Number of Startups Over Time (2000-2020)',
              labels={'Count': 'Number of Startups', 'Founded Year': 'Founded Year'

# Show the line graph
fig.show()
```

```
C:\Users\NimZee\AppData\Local\Temp\ipykernel_21236\759568864.py:8: SettingWithCopyWa
rning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
```

In [ ]:

```
import plotly.express as px

# Group startups by city and category and count the number of startups in each city
startup_city_category_counts = Startups.groupby(['City', 'Category']).size().reset_

# Sort the DataFrame by count in descending order to focus on the top categories in
startup_city_category_counts = startup_city_category_counts.sort_values(by='Count',

# Filter out the top categories in each city (you can adjust the number as needed)
top_categories_per_city = startup_city_category_counts.groupby('City').head(5)

# Create the Plotly graph
fig = px.bar(top_categories_per_city, x='City', y='Count', color='Category',
             title='Top Categories of Startups in Each City',
             labels={'Count': 'Number of Startups', 'City': 'City', 'Category': 'Ca
```

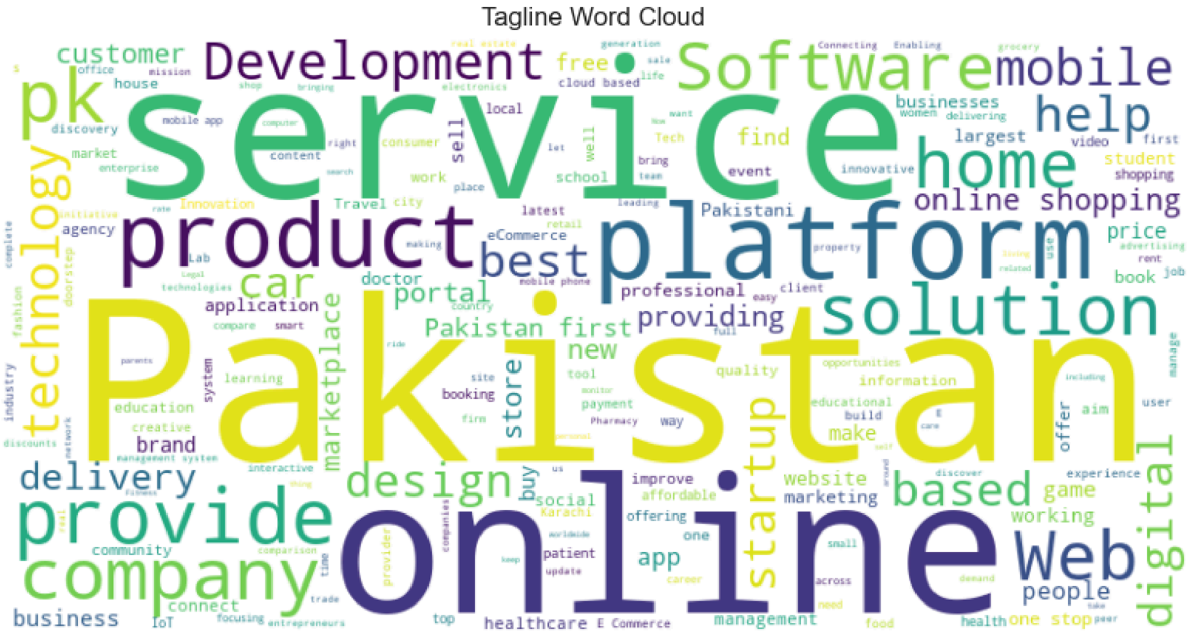
```
# Show the graph
fig.show()
```

In [ ]:

```
import plotly.express as px

# Group startups by city and category and count the number of startups in each grou
startup_counts = Startups.groupby(['City', 'Category']).size().reset_index(name='Co

# Create the treemap
fig = px.treemap(startup_counts, path=['City', 'Category'], values='Count',
                 title='Distribution of Startups by City and Category')

# Show the treemap
fig.show()
```

In [ ]:

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Combine all taglines into a single string
all_taglines = ' '.join(Startups['Tagline'].dropna())

# Create a word cloud object
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(all

# Display the word cloud
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Tagline Word Cloud')
plt.show()
```

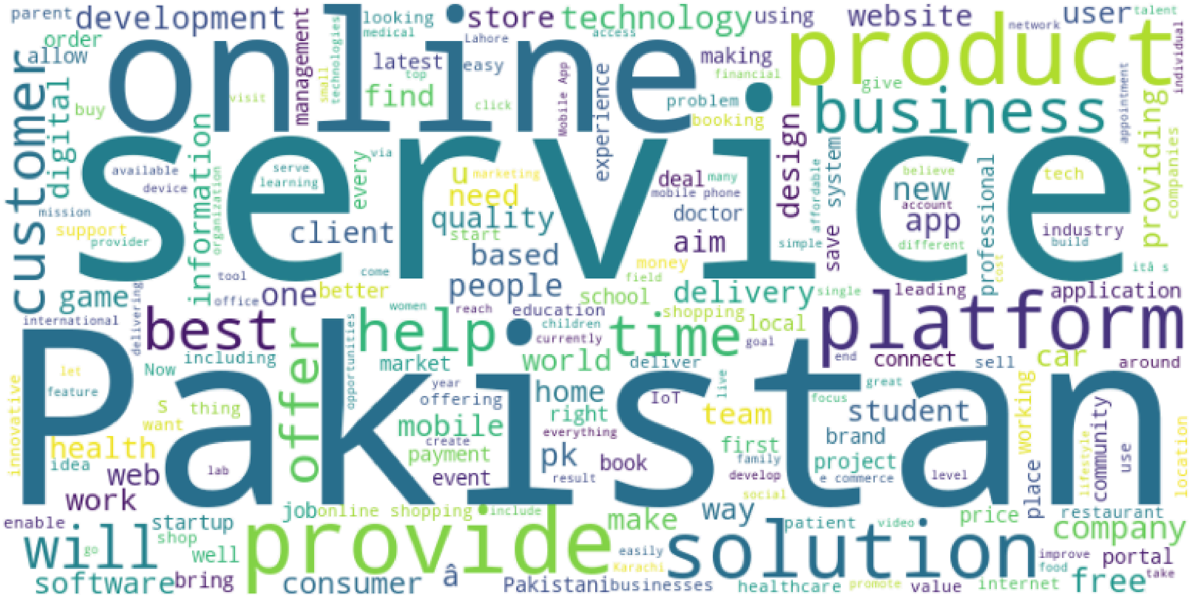


In [ ]:

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
```

```
# Combine all descriptions into a single string
all_descriptions = ' '.join(Startups['Description'].dropna())

# Create a word cloud object
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(all

# Display the word cloud
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Description Word Cloud')
plt.show()
```

Description Word Cloud



```
In [ ]: import pandas as pd
        import plotly.express as px

        # Assuming you have a DataFrame named 'Startups' containing the 'Country' and 'Cate

        # Filter out the top 30 categories
        top_categories = Startups['Category'].value_counts().head(30).index.tolist()
        filtered_startups = Startups[Startups['Category'].isin(top_categories)]

        # Group startups by country and category and count the number of startups in each g
        country_category_counts = filtered_startups.groupby(['Country', 'Category']).size()

        # Create a bar chart using Plotly
        fig = px.bar(country_category_counts, x='Country', y='Count', color='Category',
                     title='Distribution of Startups by Country and Category (Top 30)',
                     labels={'Count': 'Number of Startups'},
                     category_orders={'Country': sorted(country_category_counts['Country'].
                     width=1200, height=600)

        # Show the graph
        fig.show()
```

In [ ]:

```python
import pandas as pd

# Assuming you have a DataFrame named 'Startups'

# Convert 'Founded Year' column to numeric
Startups['Founded Year'] = pd.to_numeric(Startups['Founded Year'], errors='coerce')

# Filter out rows with NaN values in 'Founded Year' column
filtered_startups = Startups.dropna(subset=['Founded Year'])

# Group startups by founding year and count the number of startups founded in each
startup_year_counts = filtered_startups['Founded Year'].value_counts().reset_index(
startup_year_counts.columns = ['Founded Year', 'Count']

# Calculate the Pearson correlation coefficient
correlation = startup_year_counts['Founded Year'].corr(startup_year_counts['Count']

print("Pearson correlation coefficient:", correlation)
```

```
Pearson correlation coefficient: 0.4148754683849889
```

The Pearson correlation coefficient calculated is approximately 0.415.

This value indicates a positive correlation between the founding year and the number of startups founded in each year, although it is not very strong. It suggests that, on average, there is a tendency for the number of startups founded to increase slightly over time, but there may be other factors influencing startup activity as well

In [ ]:

```python
import pandas as pd

# Assuming you have a DataFrame named 'Startups'

# Convert 'Founded Year' column to numeric
Startups['Founded Year'] = pd.to_numeric(Startups['Founded Year'], errors='coerce')

# Drop rows with NaN values in 'Founded Year' column
filtered_startups = Startups.dropna(subset=['Founded Year'])

# Group startups by founding year and count the number of startups founded in each
startup_year_counts = filtered_startups['Founded Year'].value_counts().reset_index(
startup_year_counts.columns = ['Founded Year', 'Count']

# Calculate the Pearson correlation coefficient between 'Founded Year' and 'Count'
correlation_count = startup_year_counts['Founded Year'].corr(startup_year_counts['C

# Convert 'Category' column to numerical values using label encoding
Startups['Category'] = pd.Categorical(Startups['Category'])
Startups['Category Code'] = Startups['Category'].cat.codes

# Calculate the Pearson correlation coefficient between 'Founded Year' and 'Categor
correlation_category = Startups['Founded Year'].corr(Startups['Category Code'])

# Convert 'Country' column to numerical values using label encoding
Startups['Country'] = pd.Categorical(Startups['Country'])
```

```
Startups['Country Code'] = Startups['Country'].cat.codes

# Calculate the Pearson correlation coefficient between 'Founded Year' and 'Country
correlation_country = Startups['Founded Year'].corr(Startups['Country Code'])

print("Pearson correlation coefficient between Founded Year and Count:", correlatio
print("Pearson correlation coefficient between Founded Year and Category:", correla
print("Pearson correlation coefficient between Founded Year and Country:", correlat
```

```
Pearson correlation coefficient between Founded Year and Count: 0.4148754683849889
Pearson correlation coefficient between Founded Year and Category: 0.095079444198436
24
Pearson correlation coefficient between Founded Year and Country: 0.1528715792752295
```

Pearson correlation coefficient between Founded Year and Count:

We can say as the as the time goes on, the number of startups founded in each year tends to increase, but the relationship is not extremely strong.

Pearson correlation coefficient between Founded Year and Category:

There's little to no evidence of a significant linear relationship between these two variables.

Pearson correlation coefficient between Founded Year and Category:

There's little to no evidence of a significant linear relationship between these two variables.