Importing Libraries

In [1]:

```
import pandas as pd
import plotly.express as px
from datetime import datetime
import ast
import warnings
warnings.filterwarnings("ignore")
```

Loading the Dataset

In [2]:

```
User_Data = pd.read_excel('UserData (2).xlsx')
```

```
In [3]: User_Data
```

Out[3]:

	PreferredSponsors	Gender	Country	Degree	Sign Up Date	city	zip	isFromSocialMedia
0	["GlobalShala","Grant Thornton China","Saint L...	Male	Nigeria	Undergraduate Student	2023-07-23T08:05:58.602Z	Owerri	460103	0.0
1	["GlobalShala","Grant Thornton China","Saint L...	Male	India	Undergraduate Student	2023-04-24T09:57:07.405Z	kottayam	686501	0.0
2	["GlobalShala","Illinois Institute of Technolo...	NaN	India	NaN	2022-10-14T17:13:36.303Z	NaN	NaN	0.0
3	["GlobalShala","Grant Thornton China","Saint L...	NaN	Albania	NaN	2023-06-06T12:29:01.772Z	NaN	NaN	1.0
4	["GlobalShala","Grant Thornton China","Saint L...	Female	Ghana	Not in Education	2023-06-15T16:31:42.719Z	Kumasi	AT-1214-9090	0.0
...	...	...	...	...	...	...	...	...
27557	["GlobalShala","Grant Thornton China","Saint L...	Female	Botswana	Undergraduate Student	2023-04-08T05:30:44.705Z	Gaborone	123456	1.0
27558	["GlobalShala","Saint Louis University","Illin...	Male	United States	Undergraduate Student	2023-02-01T20:46:32.637Z	Coppell	75019	0.0
27559	["GlobalShala","Illinois Institute of Technolo...	Male	United States	High School Student	2022-09-22T14:06:56.114Z	Austin	78727	0.0
27560	["GlobalShala","Grant Thornton China","Saint L...	Male	Pakistan	NaN	2023-06-16T04:18:38.811Z	Daraban kalan	29111	1.0
27561	["GlobalShala","Grant Thornton China","Saint L...	Male	Bangladesh	NaN	2023-05-05T04:03:14.765Z	Dhaka	1236	1.0

27562 rows × 8 columns

Printing the first 10 rows

In [4]:
```
User_Data.head(10)
```

Out[4]:

	PreferredSponsors	Gender	Country	Degree	Sign Up Date	city	zip	isFromSocialMedia
0	["GlobalShala","Grant Thornton China","Saint L...	Male	Nigeria	Undergraduate Student	2023-07-23T08:05:58.602Z	Owerri	460103	0.0
1	["GlobalShala","Grant Thornton China","Saint L...	Male	India	Undergraduate Student	2023-04-24T09:57:07.405Z	kottayam	686501	0.0
2	["GlobalShala","Illinois Institute of Technolo...	NaN	India	NaN	2022-10-14T17:13:36.303Z	NaN	NaN	0.0
3	["GlobalShala","Grant Thornton China","Saint L...	NaN	Albania	NaN	2023-06-06T12:29:01.772Z	NaN	NaN	1.0
4	["GlobalShala","Grant Thornton China","Saint L...	Female	Ghana	Not in Education	2023-06-15T16:31:42.719Z	Kumasi	AT-1214-9090	0.0
5	["GlobalShala","Grant Thornton China","Saint L...	Female	India	NaN	2023-07-06T18:49:16.691Z	Chennai	600033	0.0
6	["GlobalShala","Grant Thornton China","Saint L...	NaN	Nigeria	NaN	2023-05-15T21:30:04.370Z	NaN	NaN	1.0
7	["GlobalShala","Grant Thornton China","Saint L...	NaN	United States	NaN	2023-07-26T17:01:59.361Z	NaN	NaN	1.0
8	["GlobalShala","Grant Thornton China","Saint L...	Male	Nigeria	Undergraduate Student	2023-07-27T18:02:17.535Z	Lagos	100278	1.0
9	["GlobalShala","Grant Thornton China","Saint L...	Male	India	High School Student	2023-05-05T04:47:25.446Z	RAS	388570	1.0

Printing the last 10 rows

In [5]:
```
User_Data.tail(10)
```

Out[5]:

	PreferredSponsors	Gender	Country	Degree	Sign Up Date	city	zip	isFromSocialMedia
27552	["GlobalShala","Grant Thornton China","Saint L...	Female	India	Graduate Program Student	2023-05-06T13:41:44.486Z	bangalore	560085	0.0
27553	["GlobalShala","Grant Thornton China","Saint L...	Female	Nigeria	Graduate Program Student	2023-06-13T07:04:59.349Z	Enugu	400107	0.0
27554	["GlobalShala","Grant Thornton China","Saint L...	Female	Pakistan	Graduate Program Student	2023-04-03T10:05:27.051Z	Karachi	75290	0.0
27555	["GlobalShala","Grant Thornton China","Saint L...	Male	India	Undergraduate Student	2023-03-31T18:01:16.166Z	Kadapa distrit	516203	1.0
27556	["Saint Louis University"]	Female	United States	High School Student	2023-05-16T00:34:56.486Z	New Lenox	60451	0.0
27557	["GlobalShala","Grant Thornton China","Saint L...	Female	Botswana	Undergraduate Student	2023-04-08T05:30:44.705Z	Gaborone	123456	1.0
27558	["GlobalShala","Saint Louis University","Illin...	Male	United States	Undergraduate Student	2023-02-01T20:46:32.637Z	Coppell	75019	0.0
27559	["GlobalShala","Illinois Institute of Technolo...	Male	United States	High School Student	2022-09-22T14:06:56.114Z	Austin	78727	0.0
27560	["GlobalShala","Grant Thornton China","Saint L...	Male	Pakistan	NaN	2023-06-16T04:18:38.811Z	Daraban kalan	29111	1.0
27561	["GlobalShala","Grant Thornton China","Saint L...	Male	Bangladesh	NaN	2023-05-05T04:03:14.765Z	Dhaka	1236	1.0

Summary of DataFrame

In [6]:

```
User_Data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27562 entries, 0 to 27561
Data columns (total 8 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   PreferredSponsors   27562 non-null  object 
 1   Gender              18027 non-null  object 
 2   Country             27500 non-null  object 
 3   Degree              16750 non-null  object 
 4   Sign Up Date        27562 non-null  object 
 5   city                18029 non-null  object 
 6   zip                 18027 non-null  object 
 7   isFromSocialMedia   27553 non-null  float64
dtypes: float64(1), object(7)
memory usage: 1.7+ MB
```

In [7]:

```
User_Data.describe()
```

Out[7]:

	isFromSocialMedia
count	27553.000000
mean	0.501252
std	0.500008
min	0.000000
25%	0.000000
50%	1.000000
75%	1.000000
max	1.000000

List of column names

```
In [8]: User_Data.columns
```

```
Out[8]: Index(['PreferredSponsors', 'Gender', 'Country', 'Degree', 'Sign Up Date',
               'city', 'zip', 'isFromSocialMedia'],
              dtype='object')
```

Checking the number rows & columns

```
In [9]: User_Data.shape
```

```
Out[9]: (27562, 8)
```

Finding the number of duplicates

```
In [10]: User_Data.duplicated().sum()
```

```
Out[10]: 0
```

Checking the Null values

```
In [11]: User_Data.isna().sum()
```

```
Out[11]: PreferredSponsors        0
         Gender                9535
         Country                 62
         Degree               10812
         Sign Up Date             0
         city                  9533
         zip                   9535
         isFromSocialMedia        9
         dtype: int64
```

Calculate the percentage of missing values for each column

In [12]:

```
missing_percentage = User_Data.isnull().mean() * 100

# Sort the columns by the highest percentage of missing values
missing_percentage = missing_percentage.sort_values(ascending=False)

# Display the missing percentage for each column
print(missing_percentage)
```

```
Degree                39.227923
Gender                34.594732
zip                   34.594732
city                  34.587476
Country                0.224947
isFromSocialMedia      0.032654
PreferredSponsors      0.000000
Sign Up Date           0.000000
dtype: float64
```

Calculate the total number of rows with missing values

In [13]:

```
total_rows = User_Data.shape[0]
missing_rows = User_Data.isnull().any(axis=1).sum()

# Calculate the percentage of rows with missing values
missing_rows_percentage = (missing_rows / total_rows) * 100

# Print the result
print(f'Total percentage of rows with missing values: {missing_rows_percentage:.2f}%')
```

```
Total percentage of rows with missing values: 39.68%
```

Checking the data types

In [14]:

```
User_Data.dtypes
```

Out[14]:

```
PreferredSponsors     object
Gender                object
Country               object
Degree                object
Sign Up Date          object
city                  object
zip                   object
isFromSocialMedia    float64
dtype: object
```

Type Casting

In [15]:

```
# 1. Fix Data Types
User_Data['Sign Up Date'] = pd.to_datetime(User_Data['Sign Up Date'])
```

Create new columns based on specific sponsors

In [16]:

```
User_Data['Is_Saint_Louis_University'] = User_Data['PreferredSponsors'].apply(lambda x: 'Saint Louis Universit
User_Data['Is_Illinois_Institute_of_Technology'] = User_Data['PreferredSponsors'].apply(lambda x: 'Illinois Ir
```

Filling the Missing values

In [17]:

```
# Process the 'PreferredSponsors' column
User_Data['PreferredSponsors'] = User_Data['PreferredSponsors'].apply(ast.literal_eval)
sponsors_exploded = User_Data.explode('PreferredSponsors')
sponsors_exploded = sponsors_exploded.reset_index(drop=True)

# Process the 'Sign Up Date' column
User_Data['Sign Up Date'] = pd.to_datetime(User_Data['Sign Up Date'], errors='coerce')
User_Data['Sign Up Date (DD-MM-YY)'] = User_Data['Sign Up Date'].dt.strftime('%d-%m-%y')
User_Data['Sign Up Time'] = User_Data['Sign Up Date'].dt.strftime('%H:%M:%S')

# Handling Missing Values
User_Data['Gender'].fillna(User_Data['Gender'].mode()[0], inplace=True)

# Fill missing 'Country' with a placeholder 'Unknown'
User_Data['Degree'].fillna('Unknown', inplace=True)

# Fill missing 'Country' with a placeholder 'Unknown'
User_Data['Country'].fillna('Unknown', inplace=True)

# Fill missing 'city' with the mode (most common value)
User_Data['city'].fillna(User_Data['city'].mode()[0], inplace=True)

# Standardize text data
User_Data['Country'] = User_Data['Country'].str.lower().str.strip()  # Convert to lowercase and remove extra s

# Convert 'isFromSocialMedia' to boolean
User_Data['isFromSocialMedia'] = User_Data['isFromSocialMedia'].astype(bool)

# Encode categorical variables (optional, for ML models)
User_Data = pd.get_dummies(User_Data, columns=['Gender', 'Country'], drop_first=True)

# Flatten the 'PreferredSponsors' column for better analysis (splitting each sponsor into its own row)
User_Data_flattened = User_Data.explode('PreferredSponsors')
```

In [18]:
```
# Extract the country columns
country_columns = [col for col in User_Data.columns if col.startswith('Country_')]

# Convert one-hot encoded country columns back into a single 'Country' column
User_Data['Country'] = User_Data[country_columns].idxmax(axis=1)

# Remove 'Country_' prefix from the new 'Country' column
User_Data['Country'] = User_Data['Country'].str.replace('Country_', '')
```

In [19]:

```
User_Data
```

Out[19]:

	PreferredSponsors	Degree	Sign Up Date	city	zip	isFromSocialMedia	Is_Saint_Louis_University	Is_Illin
0	[GlobalShala, Grant Thornton China, Saint Loui...	Undergraduate Student	2023-07-23 08:05:58.602000+00:00	Owerri	460103	False	True	
1	[GlobalShala, Grant Thornton China, Saint Loui...	Undergraduate Student	2023-04-24 09:57:07.405000+00:00	kottayam	686501	False	True	
2	[GlobalShala, Illinois Institute of Technology...	Unknown	2022-10-14 17:13:36.303000+00:00	Hyderabad	NaN	False	True	
3	[GlobalShala, Grant Thornton China, Saint Loui...	Unknown	2023-06-06 12:29:01.772000+00:00	Hyderabad	NaN	True	True	
4	[GlobalShala, Grant Thornton China, Saint Loui...	Not in Education	2023-06-15 16:31:42.719000+00:00	Kumasi	AT-1214-9090	False	True	
...	...	...	...	...	...	...	...	
27557	[GlobalShala, Grant Thornton China, Saint Loui...	Undergraduate Student	2023-04-08 05:30:44.705000+00:00	Gaborone	123456	True	True	
27558	[GlobalShala, Saint Louis University, Illinois...	Undergraduate Student	2023-02-01 20:46:32.637000+00:00	Coppell	75019	False	True	
27559	[GlobalShala, Illinois Institute of Technology...	High School Student	2022-09-22 14:06:56.114000+00:00	Austin	78727	False	False	
27560	[GlobalShala, Grant Thornton China, Saint Loui...	Unknown	2023-06-16 04:18:38.811000+00:00	Daraban kalan	29111	True	True	
27561	[GlobalShala, Grant Thornton China, Saint Loui...	Unknown	2023-05-05 04:03:14.765000+00:00	Dhaka	1236	True	True	

27562 rows × 183 columns

Verify missing values after Filling

In [20]:
```
# Verify missing valuess
print(User_Data.isna().sum())
```

```
PreferredSponsors                  0
Degree                             0
Sign Up Date                       0
city                               0
zip                             9535
                                ...
Country_virgin islands, u.s.       0
Country_yemen                      0
Country_zambia                     0
Country_zimbabwe                   0
Country                            0
Length: 183, dtype: int64
```

```
In [21]: User_Data
```

Out[21]:

	PreferredSponsors	Degree	Sign Up Date	city	zip	isFromSocialMedia	Is_Saint_Louis_University	Is_Illin
0	[GlobalShala, Grant Thornton China, Saint Loui...	Undergraduate Student	2023-07-23 08:05:58.602000+00:00	Owerri	460103	False	True	
1	[GlobalShala, Grant Thornton China, Saint Loui...	Undergraduate Student	2023-04-24 09:57:07.405000+00:00	kottayam	686501	False	True	
2	[GlobalShala, Illinois Institute of Technology...	Unknown	2022-10-14 17:13:36.303000+00:00	Hyderabad	NaN	False	True	
3	[GlobalShala, Grant Thornton China, Saint Loui...	Unknown	2023-06-06 12:29:01.772000+00:00	Hyderabad	NaN	True	True	
4	[GlobalShala, Grant Thornton China, Saint Loui...	Not in Education	2023-06-15 16:31:42.719000+00:00	Kumasi	AT-1214-9090	False	True	
...	...	...	...	...	...	...	...	
27557	[GlobalShala, Grant Thornton China, Saint Loui...	Undergraduate Student	2023-04-08 05:30:44.705000+00:00	Gaborone	123456	True	True	
27558	[GlobalShala, Saint Louis University, Illinois...	Undergraduate Student	2023-02-01 20:46:32.637000+00:00	Coppell	75019	False	True	
27559	[GlobalShala, Illinois Institute of Technology...	High School Student	2022-09-22 14:06:56.114000+00:00	Austin	78727	False	False	
27560	[GlobalShala, Grant Thornton China, Saint Loui...	Unknown	2023-06-16 04:18:38.811000+00:00	Daraban kalan	29111	True	True	
27561	[GlobalShala, Grant Thornton China, Saint Loui...	Unknown	2023-05-05 04:03:14.765000+00:00	Dhaka	1236	True	True	

27562 rows × 183 columns

In [22]:

```
User_Data.shape
```

Out[22]:

```
(27562, 183)
```

In [23]:

```
User_Data.columns
```

Out[23]:

```
Index(['PreferredSponsors', 'Degree', 'Sign Up Date', 'city', 'zip',
       'isFromSocialMedia', 'Is_Saint_Louis_University',
       'Is_Illinois_Institute_of_Technology', 'Sign Up Date (DD-MM-YY)',
       'Sign Up Time',
       ...
       'Country_unknown', 'Country_uzbekistan',
       'Country_venezuela, bolivarian republic of venezuela',
       'Country_vietnam', 'Country_virgin islands, british',
       'Country_virgin islands, u.s.', 'Country_yemen', 'Country_zambia',
       'Country_zimbabwe', 'Country'],
      dtype='object', length=183)
```

Sort the DataFrame by 'Sign Up Date (DD-MM-YY)'

In [24]:

```
# Sort the DataFrame by 'Sign Up Date (DD-MM-YY)'
User_Data_sorted = User_Data.sort_values(by='Sign Up Date (DD-MM-YY)')

# Reset the index after sorting
User_Data_sorted = User_Data_sorted.reset_index(drop=True)
```

In [25]:

```
User_Data
```

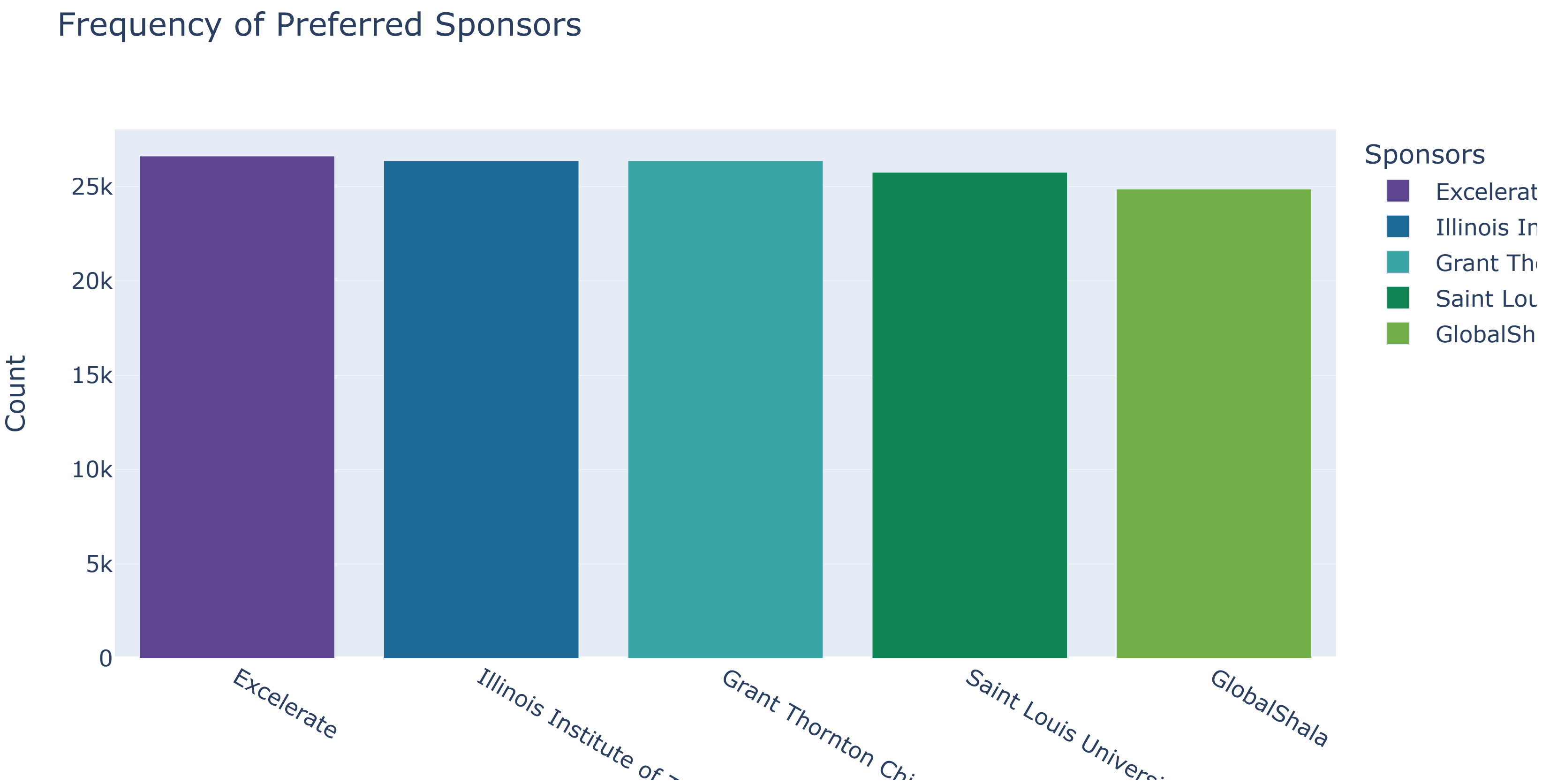
Out[25]:

	PreferredSponsors	Degree	Sign Up Date	city	zip	isFromSocialMedia	Is_Saint_Louis_University	Is_Illin
0	[GlobalShala, Grant Thornton China, Saint Loui...	Undergraduate Student	2023-07-23 08:05:58.602000+00:00	Owerri	460103	False	True	
1	[GlobalShala, Grant Thornton China, Saint Loui...	Undergraduate Student	2023-04-24 09:57:07.405000+00:00	kottayam	686501	False	True	
2	[GlobalShala, Illinois Institute of Technology...	Unknown	2022-10-14 17:13:36.303000+00:00	Hyderabad	NaN	False	True	
3	[GlobalShala, Grant Thornton China, Saint Loui...	Unknown	2023-06-06 12:29:01.772000+00:00	Hyderabad	NaN	True	True	
4	[GlobalShala, Grant Thornton China, Saint Loui...	Not in Education	2023-06-15 16:31:42.719000+00:00	Kumasi	AT-1214-9090	False	True	
...	...	...	...	...	...	...	...	
27557	[GlobalShala, Grant Thornton China, Saint Loui...	Undergraduate Student	2023-04-08 05:30:44.705000+00:00	Gaborone	123456	True	True	
27558	[GlobalShala, Saint Louis University, Illinois...	Undergraduate Student	2023-02-01 20:46:32.637000+00:00	Coppell	75019	False	True	
27559	[GlobalShala, Illinois Institute of Technology...	High School Student	2022-09-22 14:06:56.114000+00:00	Austin	78727	False	False	
27560	[GlobalShala, Grant Thornton China, Saint Loui...	Unknown	2023-06-16 04:18:38.811000+00:00	Daraban kalan	29111	True	True	
27561	[GlobalShala, Grant Thornton China, Saint Loui...	Unknown	2023-05-05 04:03:14.765000+00:00	Dhaka	1236	True	True	

27562 rows × 183 columns

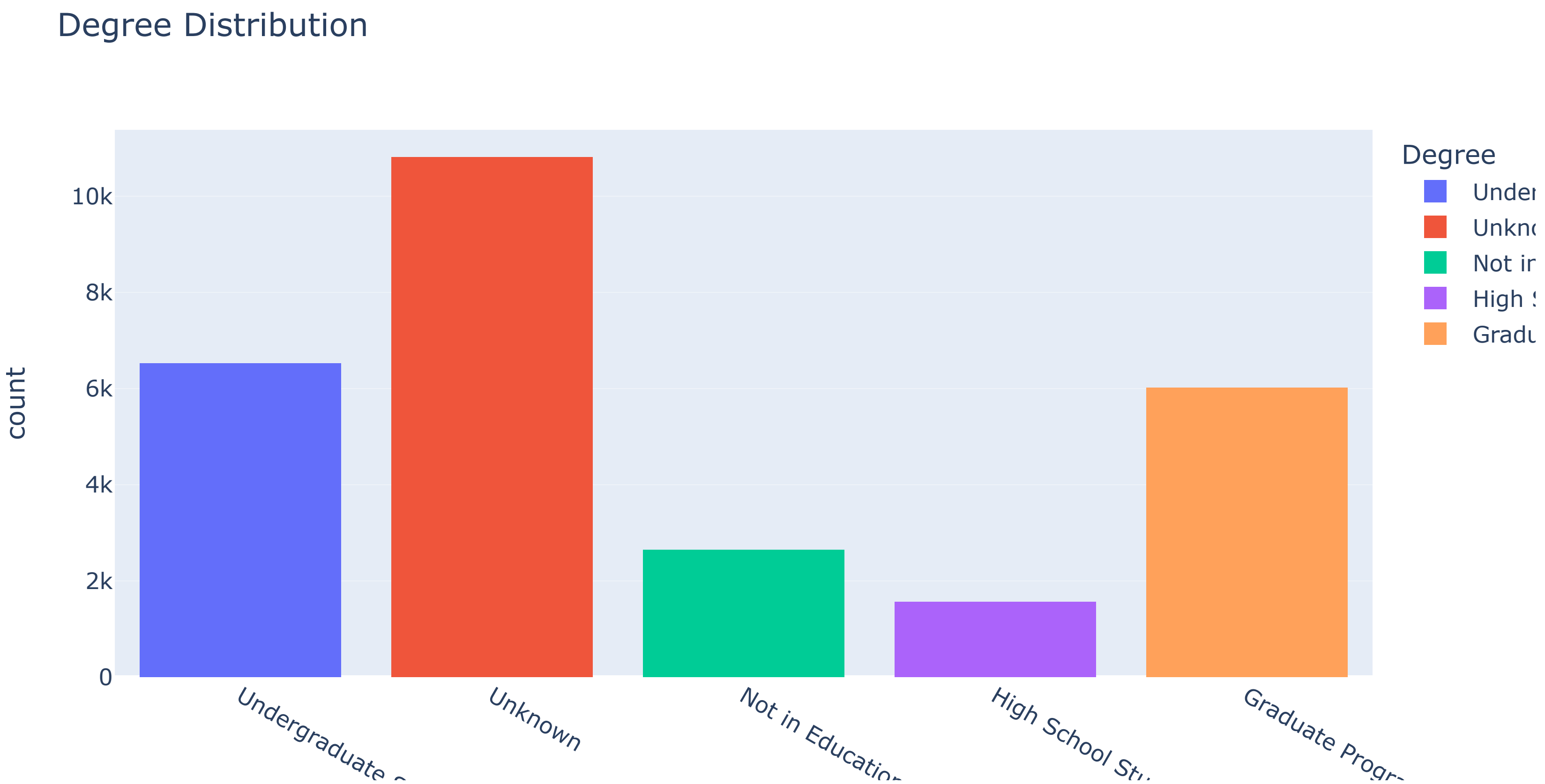
In [26]:

```
# Count how often each sponsor appears
sponsors_list = User_Data['PreferredSponsors'].explode().value_counts().reset_index()
sponsors_list.columns = ['Sponsors', 'Count']

# Add a color column to assign a unique color to each sponsor
sponsors_list['Color'] = sponsors_list['Sponsors']

# Bar plot of Preferred Sponsors with each bar having a different color
fig = px.bar(sponsors_list,
             x='Sponsors',
             y='Count',
             color='Sponsors',  # Assign a different color for each sponsor
             labels={'x': 'Sponsors', 'y': 'Count'},
             title="Frequency of Preferred Sponsors",
             color_discrete_sequence=px.colors.qualitative.Prism)
fig.show()
```


Frequency of Preferred Sponsors
Count
0
5k
10k
15k
20k
25k
Excelerate
Illinois Institute of
Grant Thornton Ch
Saint Louis Univers
GlobalShala
Sponsors
Excelerat
Illinois In
Grant Th
Saint Lou
GlobalSh

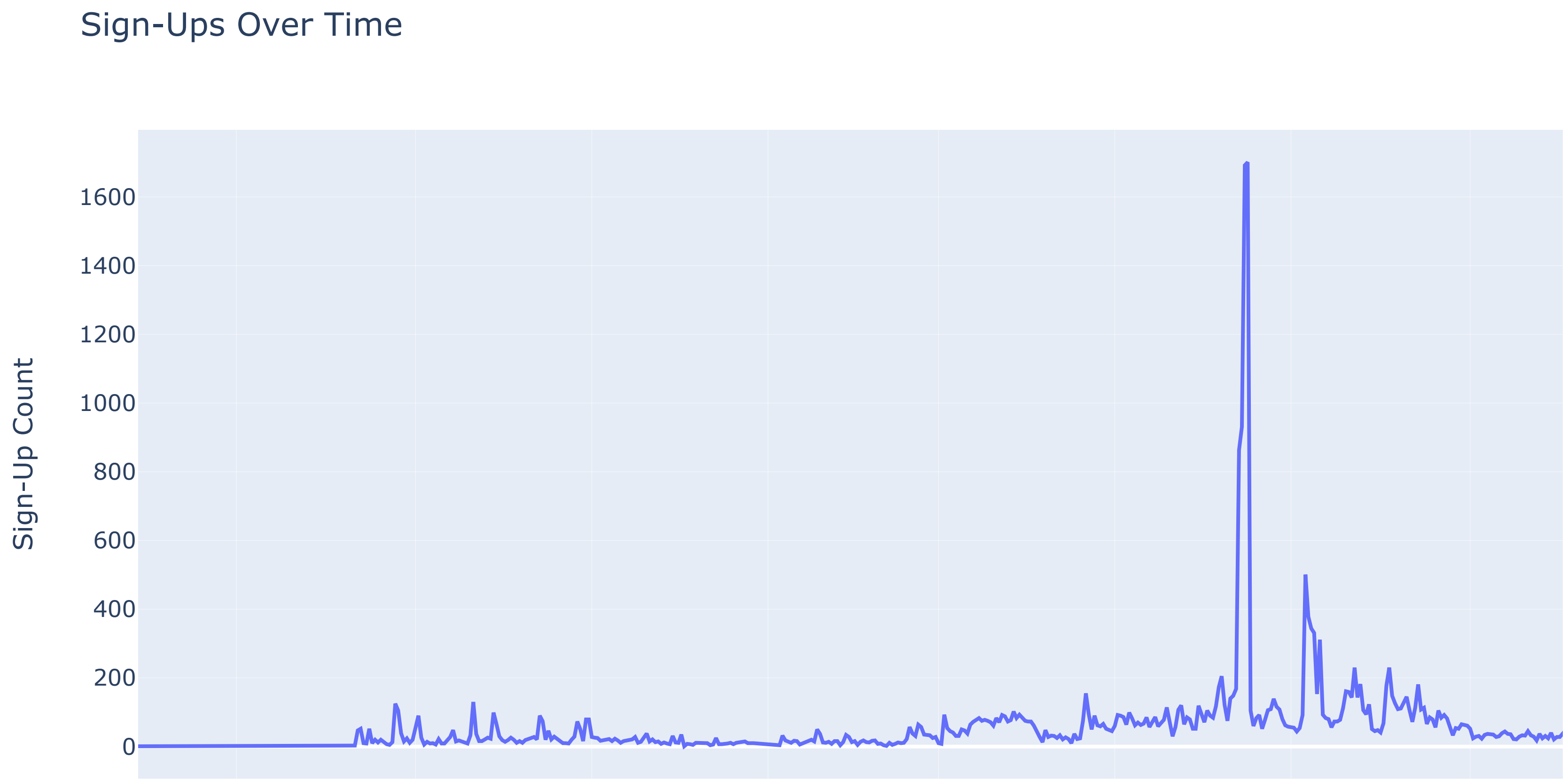
In [27]:

```
# Plot a histogram for Degree distribution
fig = px.histogram(User_Data,
                   x='Degree',
                   title="Degree Distribution",
                   color='Degree')  # Each degree has a different color
fig.show()
```

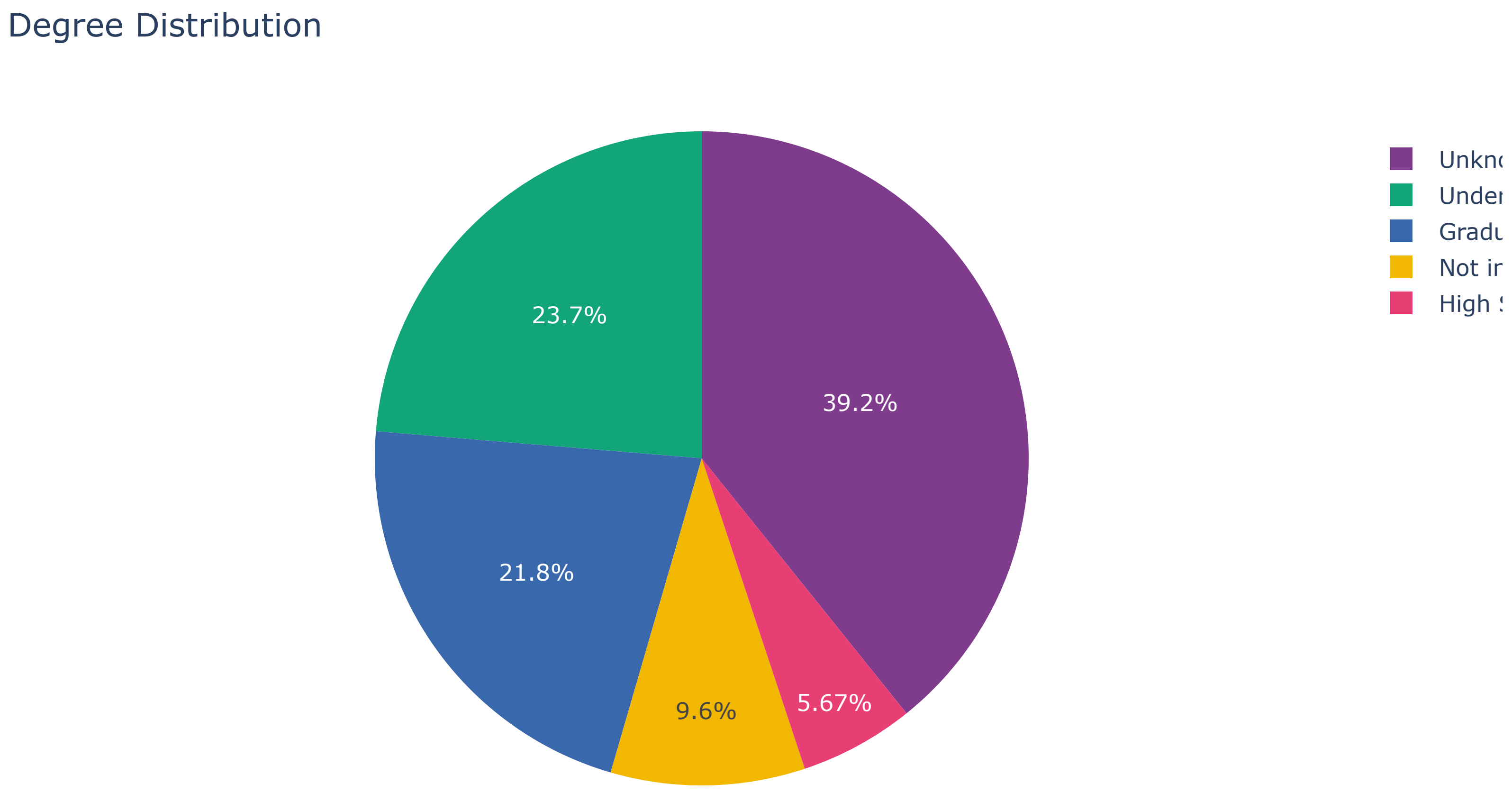


In [28]:

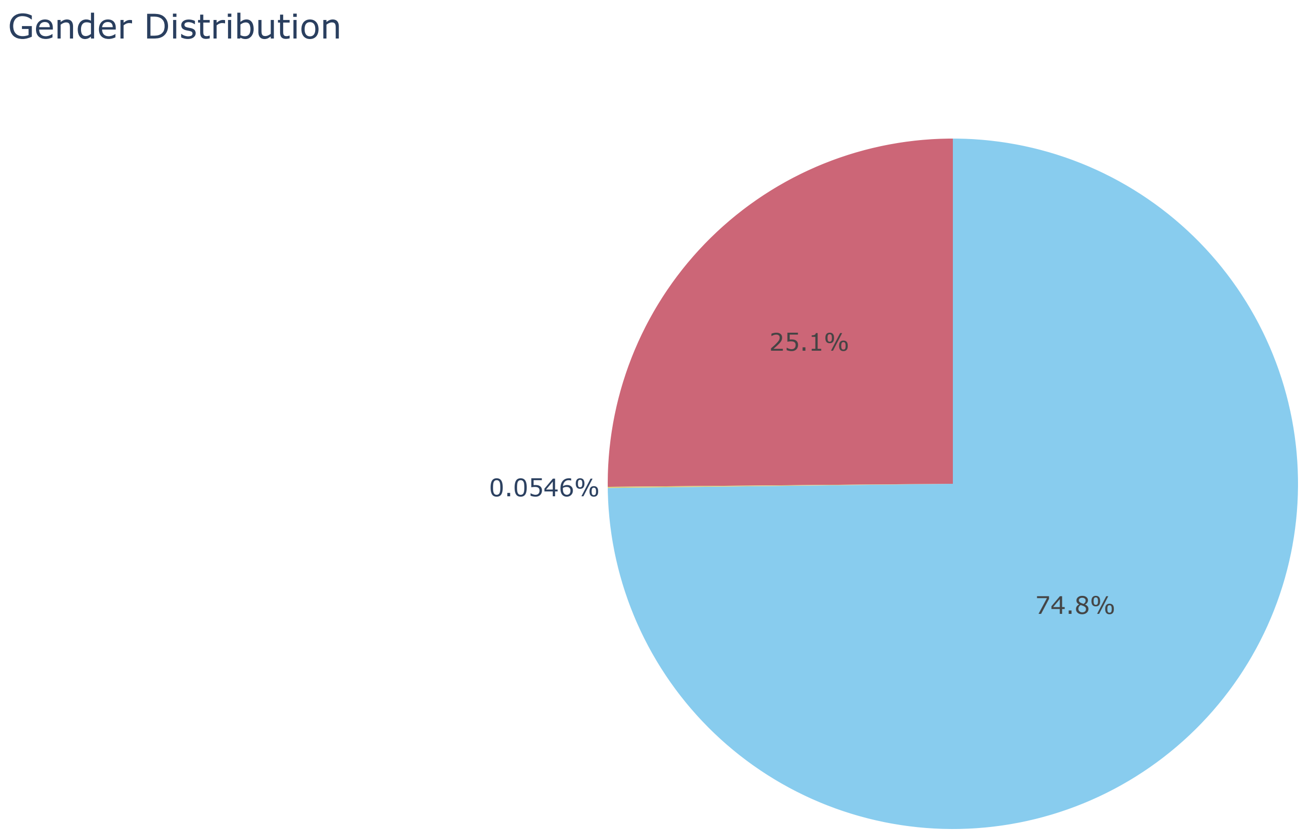
```
# Ensure the 'Sign Up Date' is in datetime format
User_Data['Sign Up Date'] = pd.to_datetime(User_Data['Sign Up Date'])

# Count sign-ups by date
signups_by_date = User_Data.groupby(User_Data['Sign Up Date'].dt.date).size()

# Plot a line chart for sign-ups over time
fig = px.line(x=signups_by_date.index,
              y=signups_by_date.values,
              labels={'x': 'Date', 'y': 'Sign-Up Count'},
              title="Sign-Ups Over Time"
             )
fig.show()
```


Sign-Ups Over Time
Sign-Up Count
1600
1400
1200
1000
800
600
400
200
0

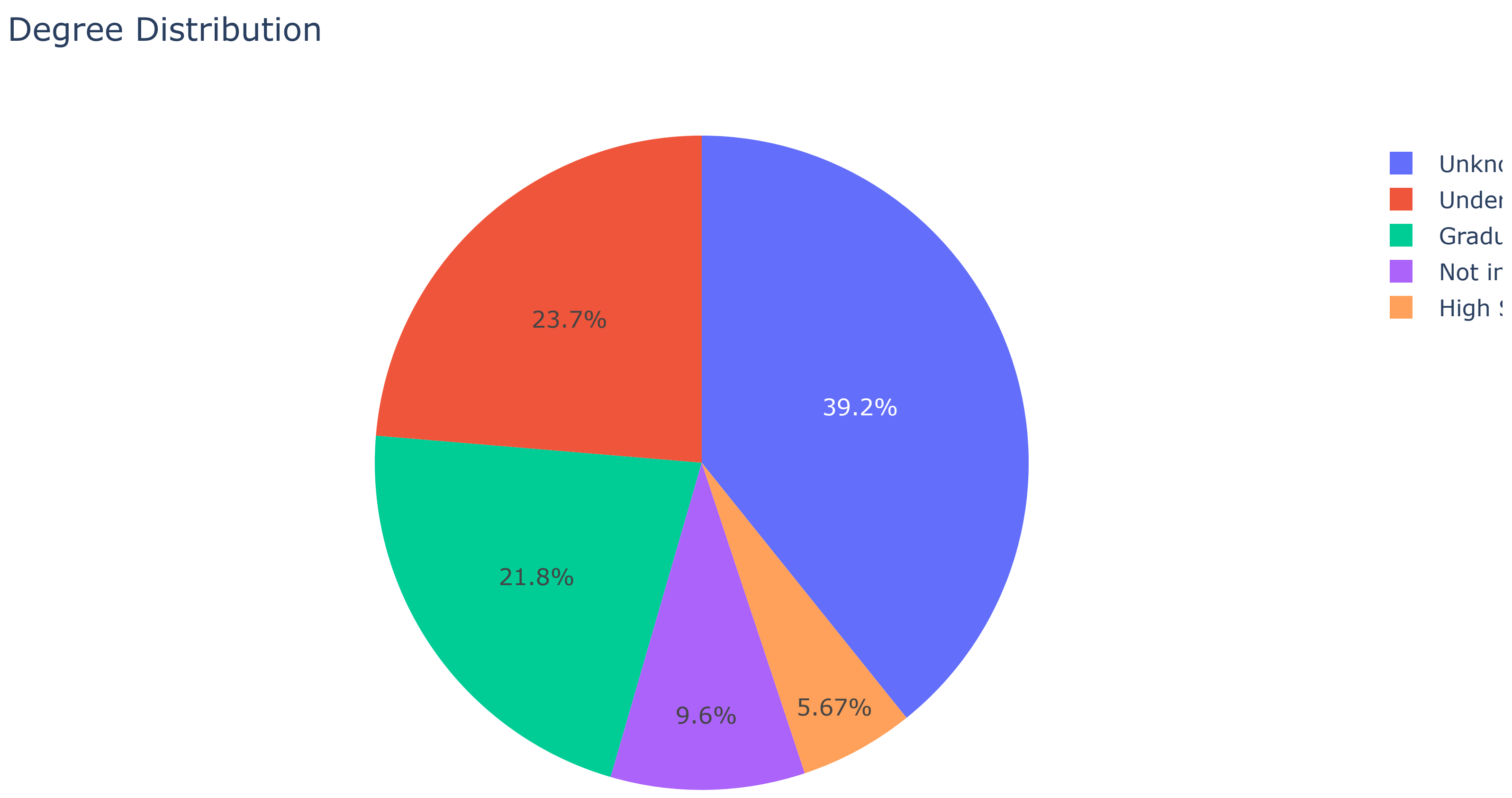
In [29]:
```
# Pie chart of Degree distribution
degree_counts = User_Data['Degree'].value_counts()
fig = px.pie(values=degree_counts.values, names=degree_counts.index,
             title="Degree Distribution", color_discrete_sequence=px.colors.qualitative.Bold)
fig.show()
```



```
In [30]: # Pie chart of Gender distribution
         fig = px.pie(values=[User_Data['Gender_Male'].sum(), User_Data['Gender_Female'].sum(), User_Data['Gender_Other
                      names=['Male', 'Female', 'Other'], title="Gender Distribution",
                      color_discrete_sequence=px.colors.qualitative.Safe)
         fig.show()
```

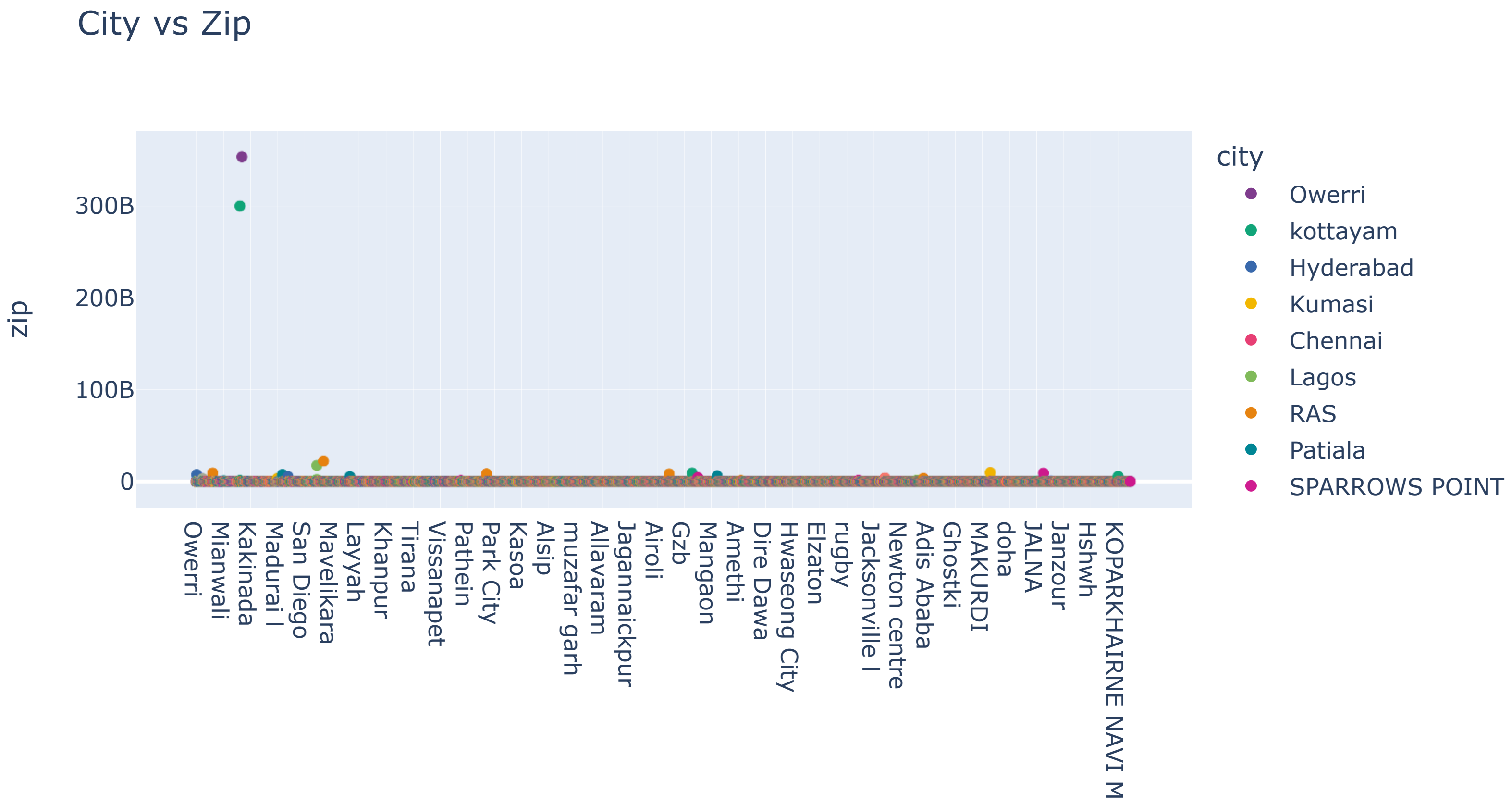


```
In [31]: # Pie chart of Degree distribution
         fig = px.pie(User_Data, names='Degree', title="Degree Distribution")
         fig.show()
```



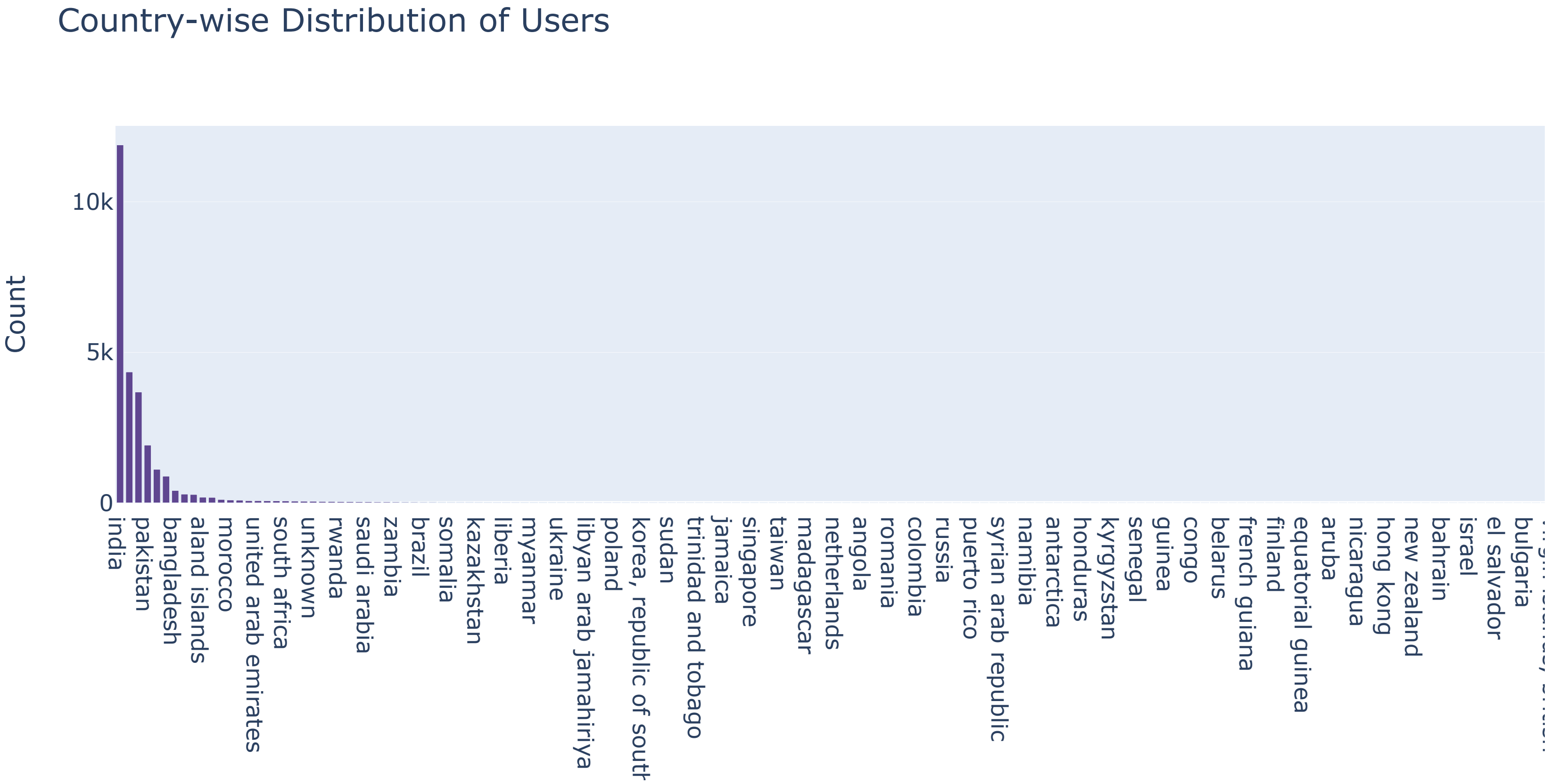
In [32]:

```
# Scatter plot of city vs. zip
fig = px.scatter(User_Data,
                 x='city',
                 y='zip',
                 title="City vs Zip",
                 color='city',
                 color_discrete_sequence=px.colors.qualitative.Bold)  # Bold colors for scatter plot
fig.show()
```



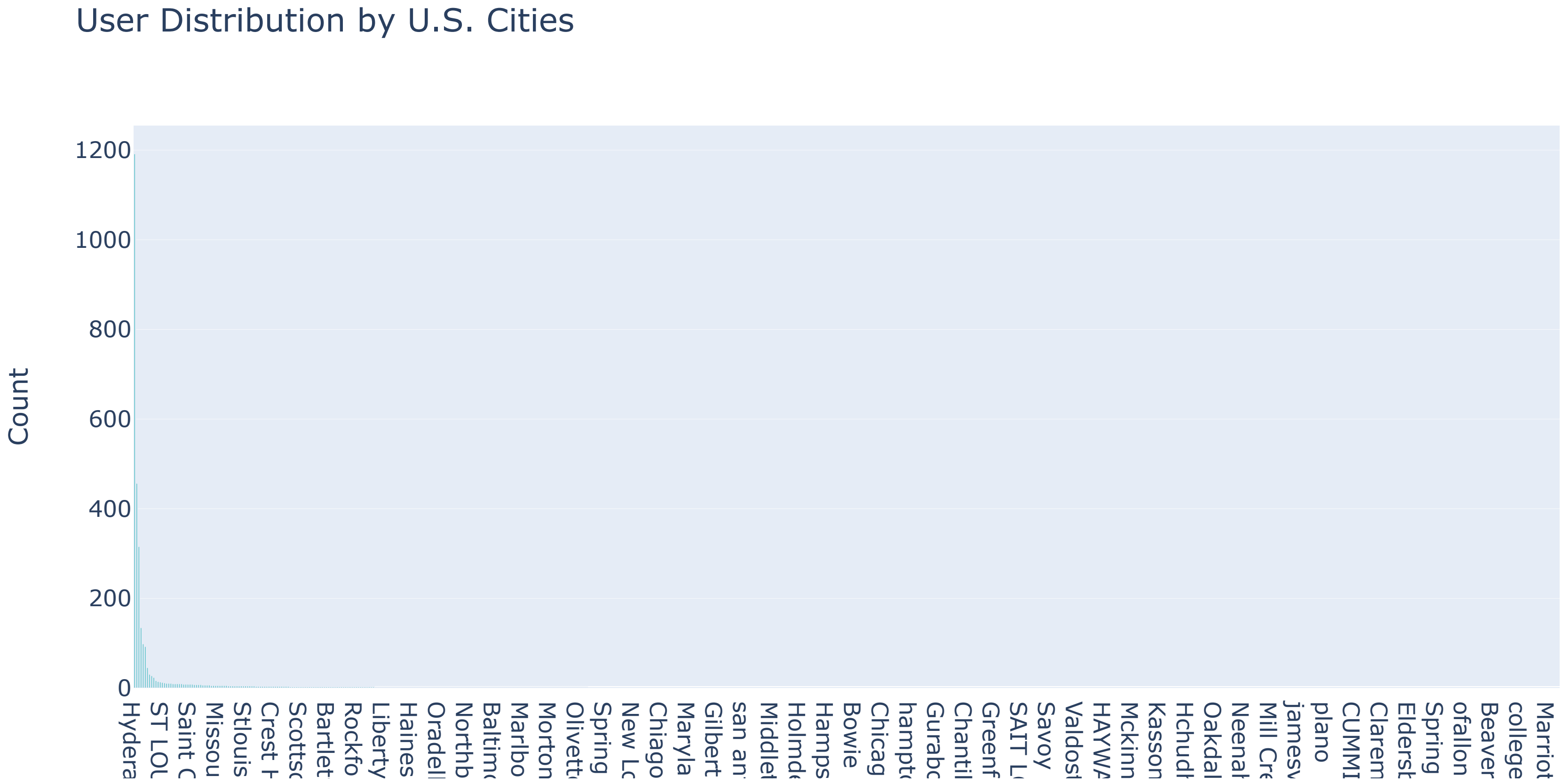
In [33]:

```
# Bar plot for user count by country
country_counts = User_Data['Country'].value_counts()

# Create bar plot
fig = px.bar(country_counts,
             x=country_counts.index,
             y=country_counts.values,
             labels={'x': 'Country', 'y': 'Count'},
             title="Country-wise Distribution of Users",
             color_discrete_sequence=px.colors.qualitative.Prism)

fig.show()
```


Country-wise Distribution of Users
Count
10k
5k
0
india
pakistan
bangladesh
aland islands
morocco
united arab emirates
south africa
unknown
rwanda
saudi arabia
zambia
brazil
somalia
kazakhstan
liberia
myanmar
ukraine
libyan arab jamahiriya
poland
korea, republic of soutr
sudan
trinidad and tobago
jamaica
singapore
taiwan
madagascar
netherlands
angola
romania
colombia
russia
puerto rico
syrian arab republic
namibia
antarctica
honduras
kyrgyzstan
senegal
guinea
congo
belarus
french guiana
finland
equatorial guinea
aruba
nicaragua
hong kong
new zealand
bahrain
israel
el salvador
bulgaria

In [34]:

```python
# Filter data for users from the United States
us_data = User_Data[User_Data['Country_united states'] == 1]

# Count the number of users per city
city_counts = us_data['city'].value_counts()

# Plot the bar plot for U.S. cities
fig = px.bar(city_counts,
             x=city_counts.index,
             y=city_counts.values,
             labels={'x': 'City', 'y': 'Count'},
             title="User Distribution by U.S. Cities",
             color_discrete_sequence=px.colors.qualitative.Pastel)

fig.show()
```


User Distribution by U.S. Cities
Count
1200
1000
800
600
400
200
0
Hydera
ST LOU
Saint C
Missou
Stlouis
Crest H
Scottsc
Bartlet
Rockfo
Liberty
Haines
Oradell
Northb
Baltimo
Marlbo
Morton
Olivett
Spring
New Lo
Chiago
Maryla
Gilbert
san an
Middlet
Holmde
Hamps
Bowie
Chicag
hampto
Gurabo
Chantil
Greenf
SAIT Lo
Savoy
Valdost
HAYWA
Mckinn
Kasson
Hchudh
Oakdal
Neenah
Mill Cre
jamesv
plano
CUMMI
Clarem
Eldersk
Spring
ofallon
Beaver
college
Marriot

In [35]:

```
# List of cities to include
cities_to_include = ['Aurora', 'Chesterfield', 'Chicago', 'Edison', 'MARYLAND HEIGHTS',
                     'Naperville', 'Saint louis', 'saintlouis', 'Skokie', 'St louis']

# Filter the DataFrame to include only the specified cities
filtered_data = User_Data[User_Data['city'].isin(cities_to_include)]

# Display the first few rows of the filtered data to verify
filtered_data.head()
```

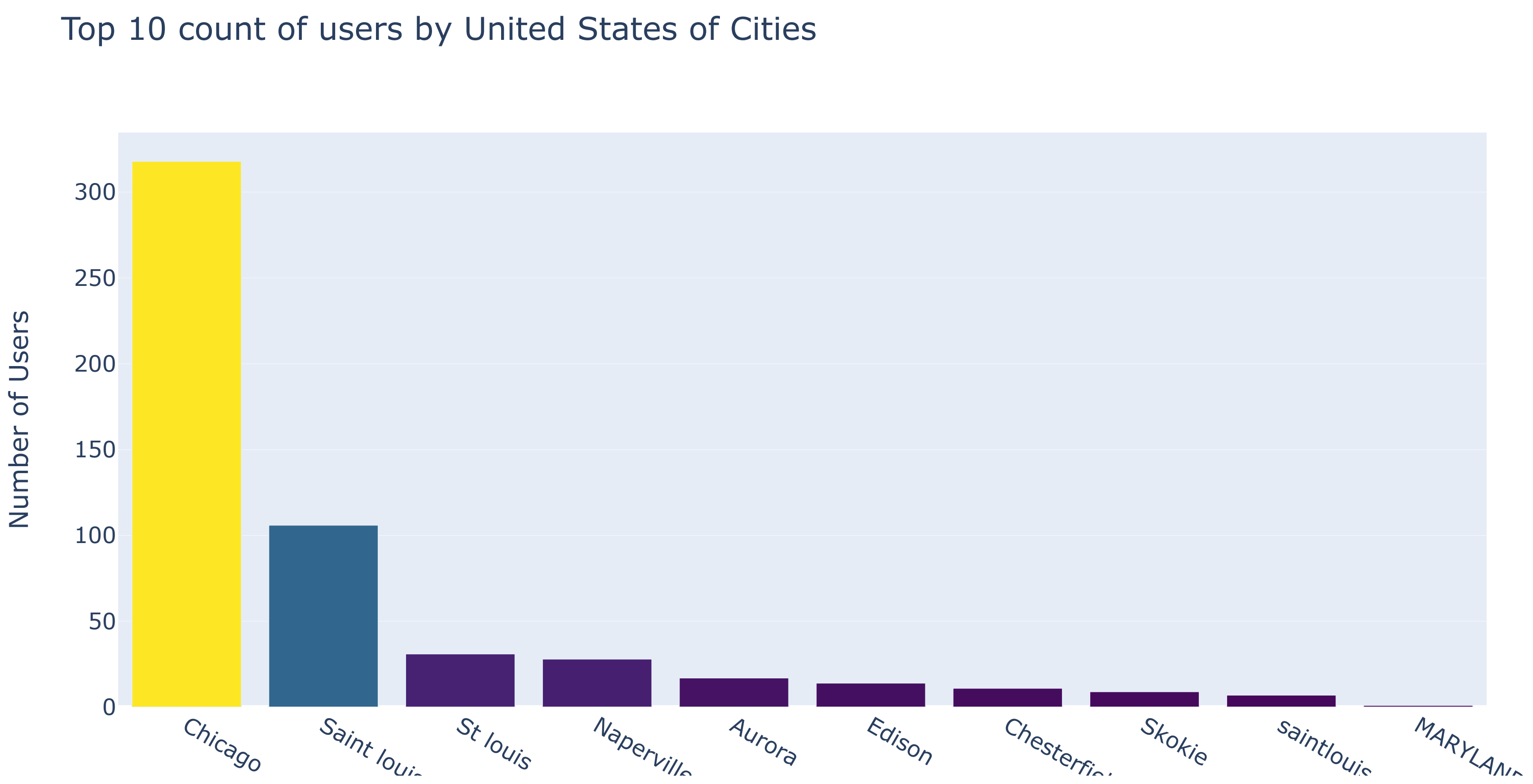
Out[35]:

	PreferredSponsors	Degree	Sign Up Date	city	zip	isFromSocialMedia	Is_Saint_Louis_University	Is_Illinois_I
14	[GlobalShala, Grant Thornton China, Saint Loui...	Graduate Program Student	2023-10-15 01:46:00.107000+00:00	Saint louis	63043	True	True	
39	[GlobalShala, Grant Thornton China, Saint Loui...	Undergraduate Student	2023-04-09 20:35:20.042000+00:00	Chicago	60614-4904	True	True	
45	[Saint Louis University, Excelerate]	Graduate Program Student	2023-08-21 22:28:53.138000+00:00	St louis	63103	False	True	
58	[GlobalShala, Illinois Institute of Technology...	Graduate Program Student	2022-09-16 21:59:13.364000+00:00	Chicago	60616	False	True	
104	[GlobalShala, Saint Louis University, Illinois...	Graduate Program Student	2023-01-06 15:26:36.746000+00:00	St louis	63108	False	True	

5 rows × 183 columns

In [36]:

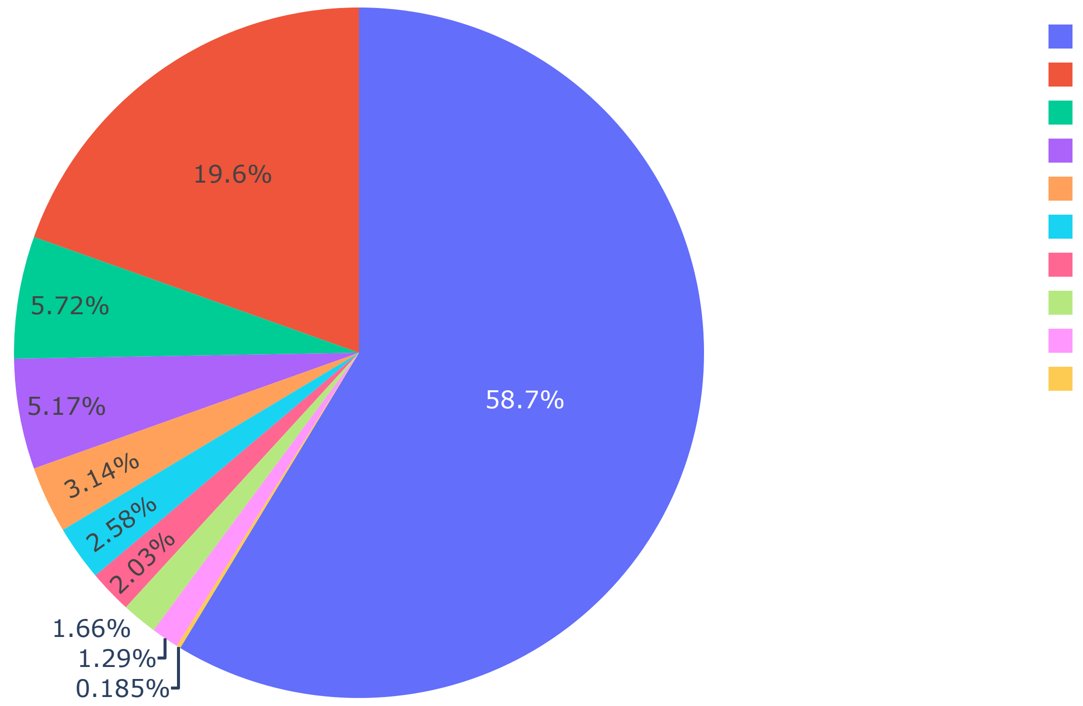
```
# Aggregate data to count users by city
city_counts = filtered_data['city'].value_counts().reset_index()
city_counts.columns = ['city', 'Count']

# Create the bar plot
fig = px.bar(city_counts, x='city', y='Count',
             title='Top 10 count of users by United States of Cities',
             labels={'city': 'City', 'Count': 'Number of Users'},
             color='Count',
             color_continuous_scale=px.colors.sequential.Viridis)

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


Top 10 count of users by United States of Cities
Number of Users
300
250
200
150
100
50
0
Chicago
Saint louis
St louis
Naperville
Aurora
Edison
Chesterfield
Skokie
saintlouis
MARYLAND

In [37]:

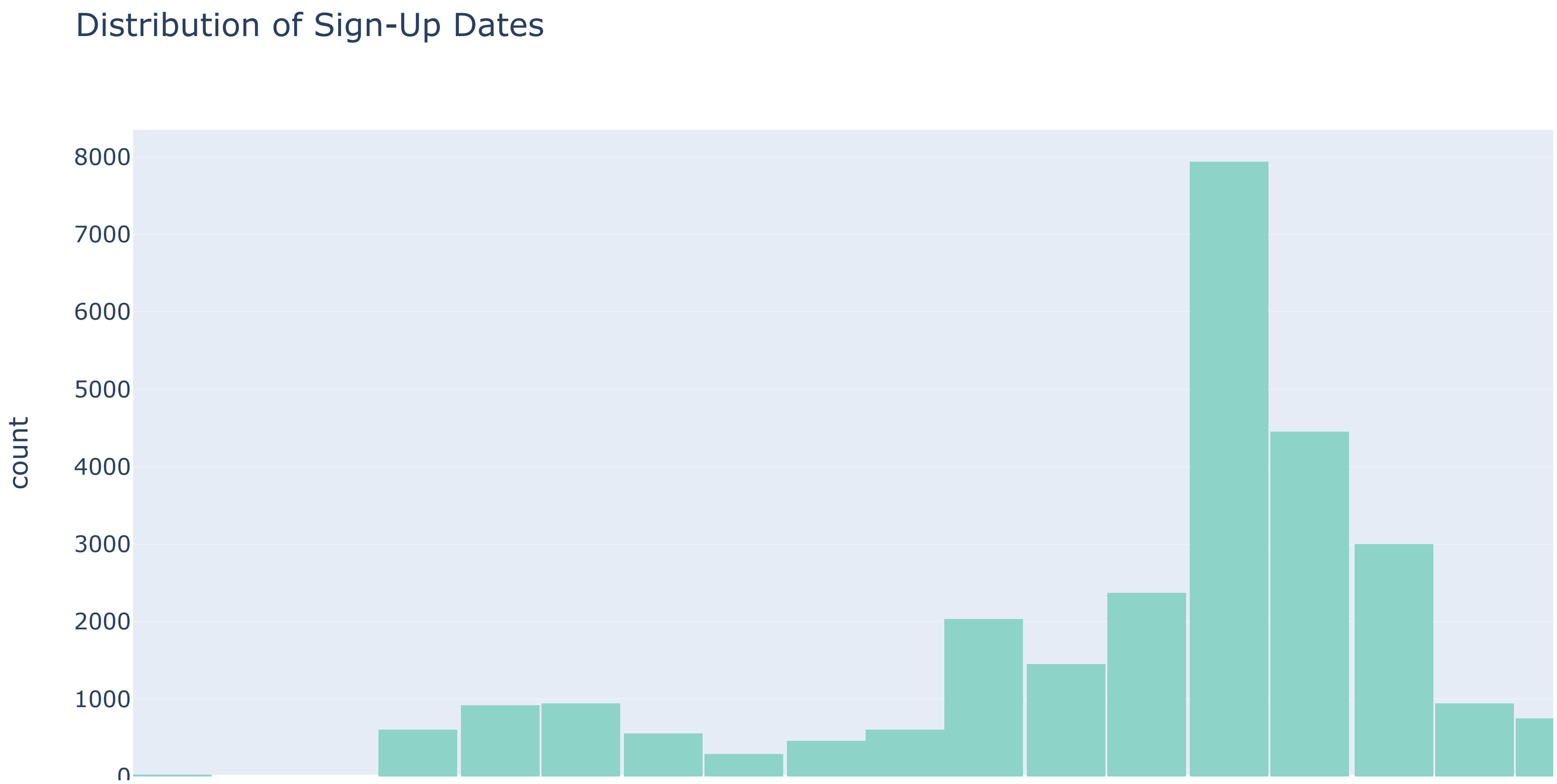
```
# Step 1: Aggregate data to count users by city
city_counts = filtered_data['city'].value_counts().reset_index()
city_counts.columns = ['city', 'Count']

# Step 2: Create the pie chart
fig = px.pie(city_counts, names='city', values='Count',
             title='User Distribution by City',
             labels={'city': 'City', 'Count': 'Number of Users'})

# Step 3: Show the plot
fig.show()
```

User Distribution by City

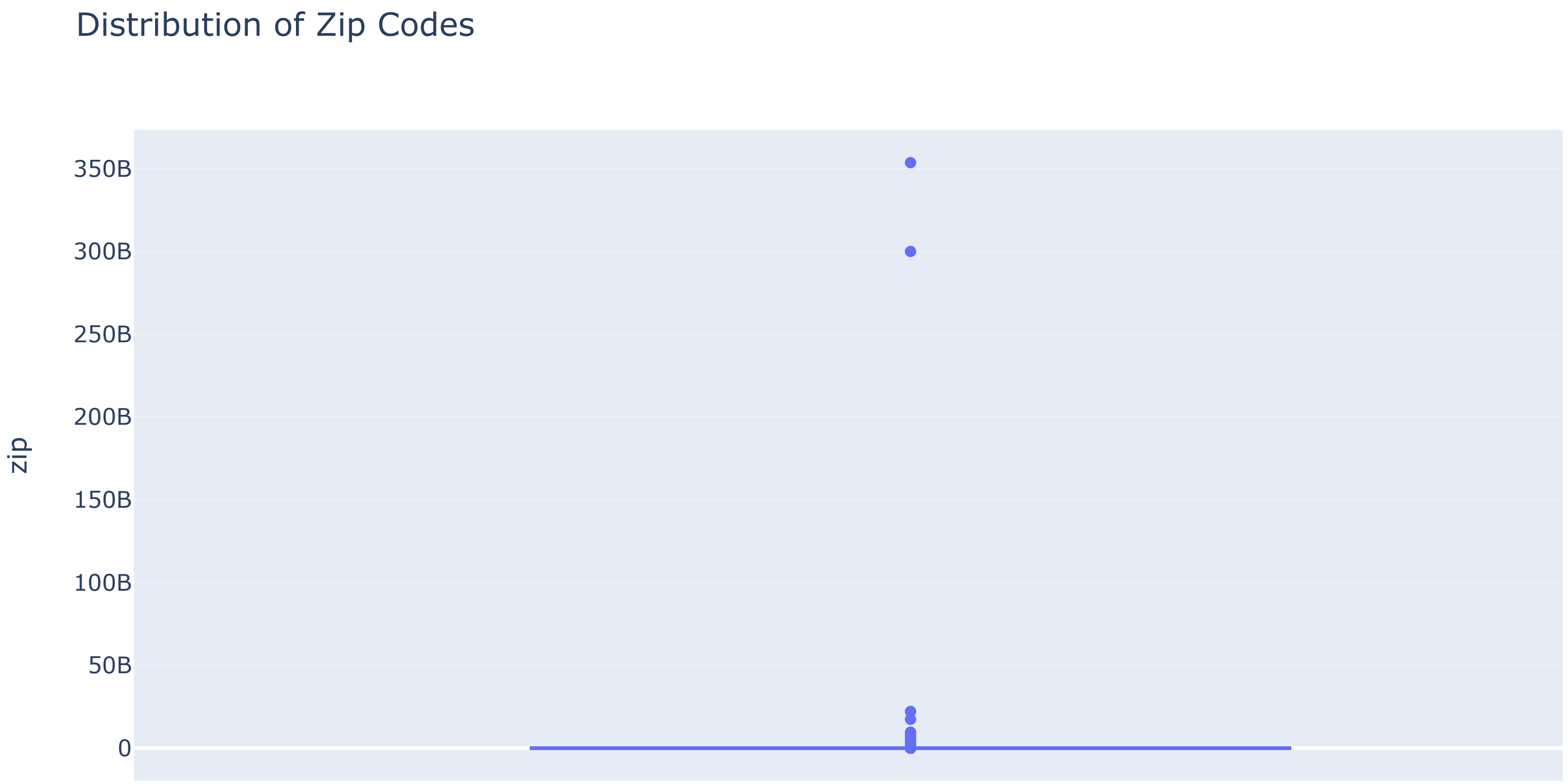


In [38]:

```
# Histogram of sign-up dates
fig = px.histogram(User_Data, x='Sign Up Date',
                   title="Distribution of Sign-Up Dates",
                   nbins=30, color_discrete_sequence=px.colors.qualitative.Set3)
fig.show()
```

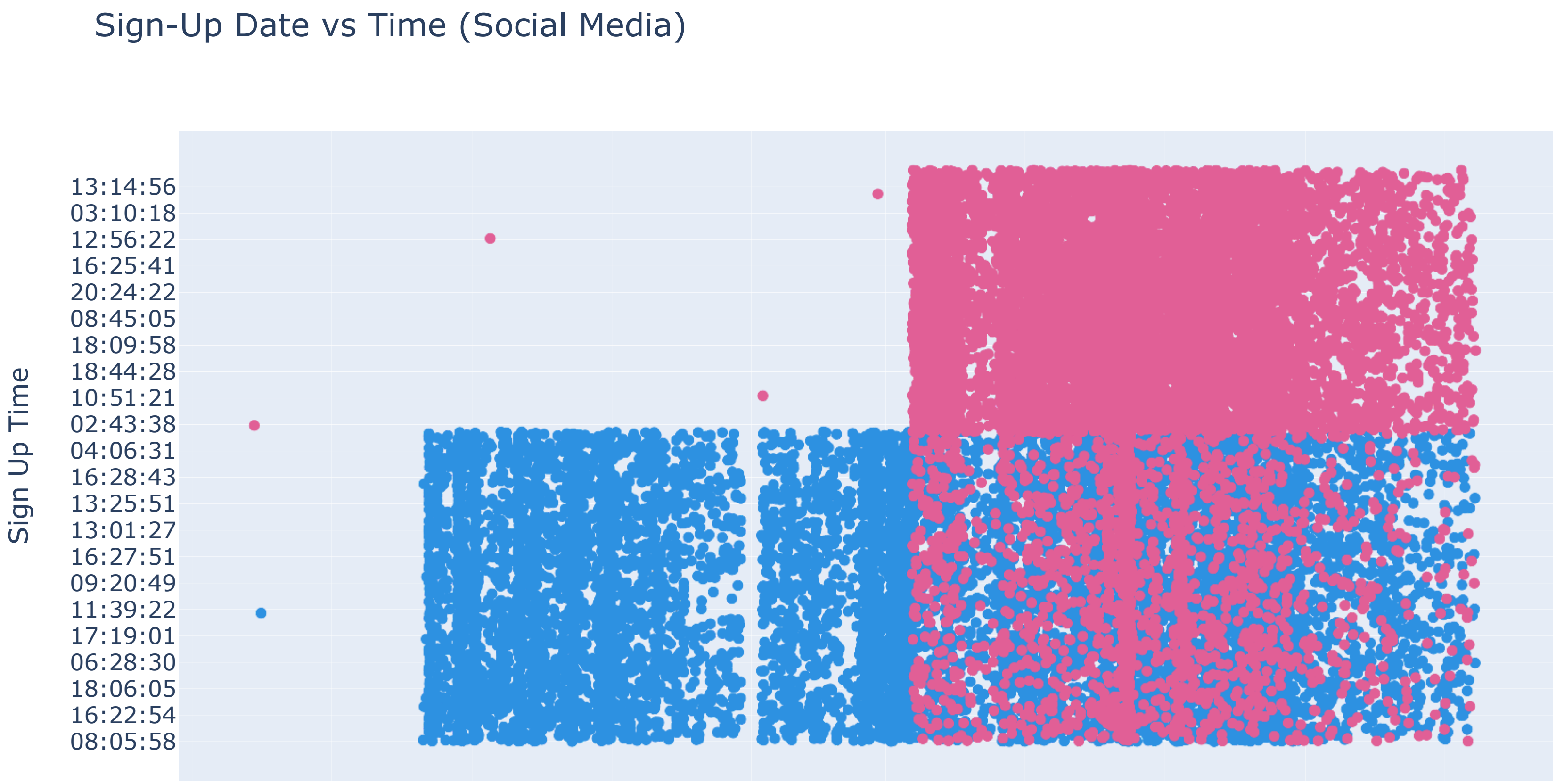


In [39]:
```
# Box plot for zip codes
fig = px.box(User_Data, y='zip', title="Distribution of Zip Codes",
             color_discrete_sequence=px.colors.qualitative.Plotly)
fig.show()
```

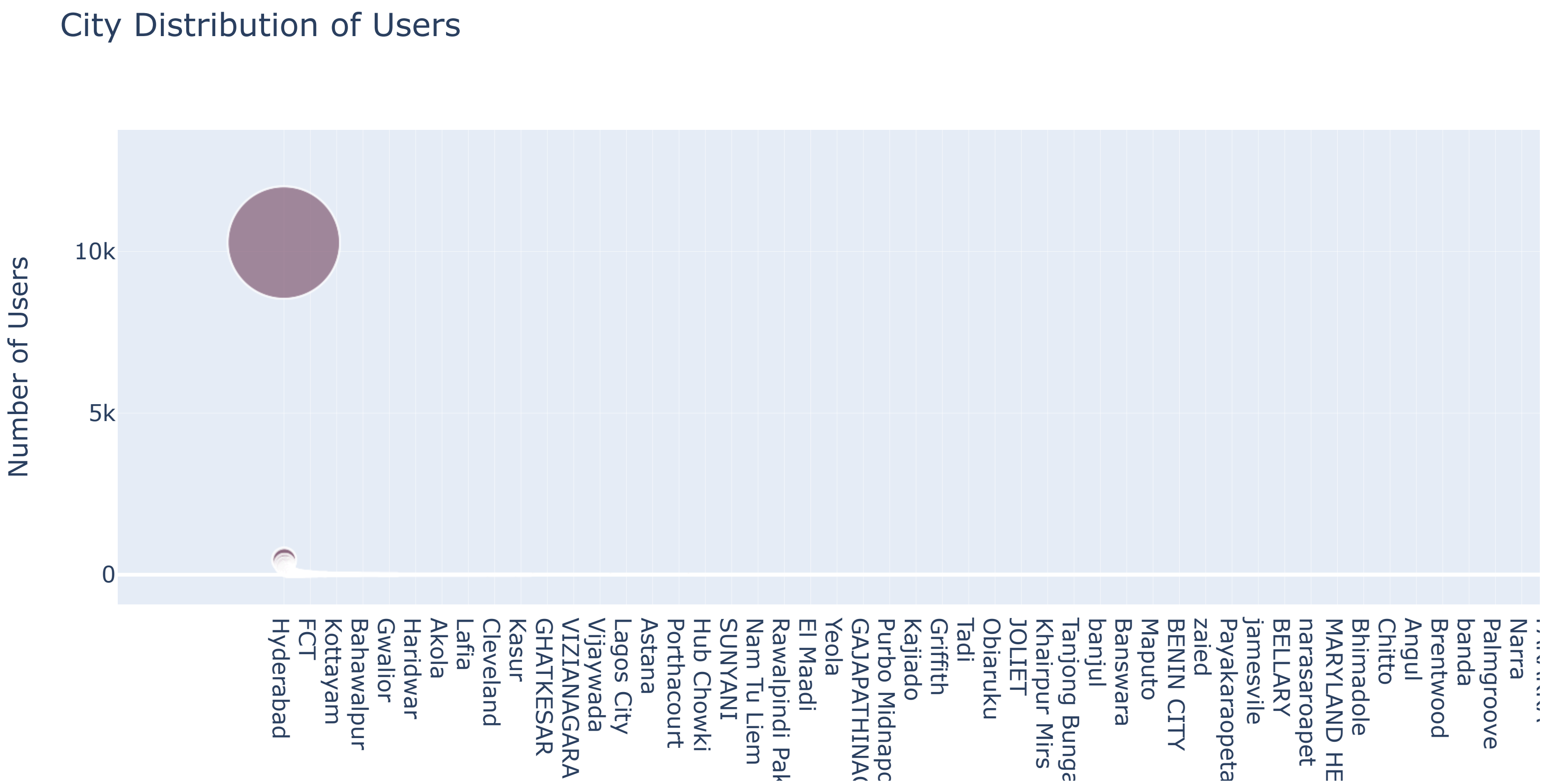


In [40]:

```
# Scatter plot for sign-up date and time
fig = px.scatter(User_Data, x='Sign Up Date', y='Sign Up Time',
                 color='isFromSocialMedia', title="Sign-Up Date vs Time (Social Media)",
                 color_discrete_sequence=px.colors.qualitative.Dark24)
fig.show()
```



In [41]:
```
# City distribution as a bubble chart
city_counts = User_Data['city'].value_counts()
fig = px.scatter(city_counts, x=city_counts.index, y=city_counts.values,
                 size=city_counts.values, title="City Distribution of Users",
                 labels={'x': 'City', 'y': 'Number of Users'},
                 size_max=60, color_discrete_sequence=px.colors.qualitative.Antique)
fig.show()
```



In [42]:

```
User_Data.to_csv('cleaned_User_Data.csv', index=False)
```

In [ ]: