In [1]:

```python
# 1.fetching information about species from the GBIF AP

import requests
import pandas as pd

# Base URL for the GBIF API
url = 'https://api.gbif.org/v1/species/search'

# Define the parameters for the search (e.g., searching for "Puma concolor")
params = {
    'q': 'Puma concolor',
    'rank': 'species',
}

# Send a GET request to the API
response = requests.get(url, params=params)

# Check if the request was successful
if response.status_code == 200:
    species_data = response.json()

    # Convert the results to a DataFrame
    df_species = pd.DataFrame(species_data['results'])

    # Display the first few rows of species data
    print("Species Data:")
    print(df_species.head())
else:
    print(f"Error fetching data: {response.status_code}")
```

```
Species Data:
         key    nameKey                              datasetKey   nubKey  \
0  164366659  9541879.0  e7250318-b8ac-4780-b2c8-da946f4792da  2435099
```

```
1  157169053  9541879.0  081304be-3a8b-436d-a9b4-185b6cdda870  2435099
2  116892593        NaN  71667154-257d-4d8e-a2a5-711aaf9b2d74  2435099
3  104061094        NaN  fab88965-e69d-4491-a04d-e3198b626e52  2435099
4  212466002        NaN  accaeedb-7e50-4a42-8ac0-714073d05311  2435099

   parentKey parent   kingdom    phylum      order   family  ...  \
0  164366658   Puma  Animalia  Chordata  Carnivora  Felidae  ...
1  157251292   Puma  Animalia  Chordata  Carnivora  Felidae  ...
2  116892589   Puma       NaN       NaN  Carnivora  Felidae  ...
3  104061090   Puma   Metazoa  Chordata  Carnivora  Felidae  ...
4  212471935   Puma  Animalia  Chordata  Carnivora  Felidae  ...

      threatStatuses descriptions  \
0  [NEAR_THREATENED]           []
1                 []           []
2                 []           []
3                 []           []
4                 []           []

                                  vernacularNames  \
0                                              []
1                                              []
2  [{'vernacularName': 'puma', 'language': 'eng'}]
3                   [{'vernacularName': 'puma'}]
4                                              []

                           higherClassificationMap  synonym     class  \
0  {'164365843': 'Animalia', '164365844': 'Chorda...    False  Mammalia
1  {'157251234': 'Animalia', '157251235': 'Chorda...    False  Mammalia
2  {'116891946': 'Mammalia', '116892052': 'Carniv...    False  Mammalia
3  {'103832354': 'Metazoa', '103882489': 'Chordat...    False  Mammalia
4  {'212471346': 'Animalia', '212471364': 'Chorda...    False  Mammalia

   acceptedKey  accepted  basionymKey basionym
0          NaN       NaN          NaN      NaN
1          NaN       NaN          NaN      NaN
2          NaN       NaN          NaN      NaN
3          NaN       NaN          NaN      NaN
4          NaN       NaN          NaN      NaN

[5 rows x 40 columns]
```

In [2]:

```python
# 2. fetching occurrence data for a species, allowing me to find where certain spe
import requests
import pandas as pd

# Base URL for GBIF occurrence search API
```

```
url = "https://api.gbif.org/v1/occurrence/search"

# Define parameters for the search (e.g., finding occurrences of 'Puma concolor')
params = {
    'scientificName': 'Puma concolor',
    'limit': 10
}

# Send a GET request to the API
response = requests.get(url, params=params)

# Check if the request was successful
if response.status_code == 200:
    occurrence_data = response.json()

    # Convert the results to a DataFrame
    df_occurrence = pd.DataFrame(occurrence_data['results'])

    # Display the first few rows of occurrence data
    print("Occurrence Data:")
    print(df_occurrence.head())
else:
    print(f"Error fetching data: {response.status_code}")
```

```
Occurrence Data:
          key                              datasetKey  \
0  4510345615  50c9509d-22c7-4a22-a47d-8c48425ef4a7
1  4924177840  50c9509d-22c7-4a22-a47d-8c48425ef4a7
2  4510372143  50c9509d-22c7-4a22-a47d-8c48425ef4a7
3  4510335239  50c9509d-22c7-4a22-a47d-8c48425ef4a7
4  4510165116  50c9509d-22c7-4a22-a47d-8c48425ef4a7

                      publishingOrgKey                          installationKey  \
```

```
0  28eb1a3f-1c15-4a95-931a-4af90ecb574d  997448a8-f762-11e1-a439-00145eb45e9a
1  28eb1a3f-1c15-4a95-931a-4af90ecb574d  997448a8-f762-11e1-a439-00145eb45e9a
2  28eb1a3f-1c15-4a95-931a-4af90ecb574d  997448a8-f762-11e1-a439-00145eb45e9a
3  28eb1a3f-1c15-4a95-931a-4af90ecb574d  997448a8-f762-11e1-a439-00145eb45e9a
4  28eb1a3f-1c15-4a95-931a-4af90ecb574d  997448a8-f762-11e1-a439-00145eb45e9a

                 hostingOrganizationKey publishingCountry     protocol  \
0  28eb1a3f-1c15-4a95-931a-4af90ecb574d                 US  DWC_ARCHIVE
1  28eb1a3f-1c15-4a95-931a-4af90ecb574d                 US  DWC_ARCHIVE
2  28eb1a3f-1c15-4a95-931a-4af90ecb574d                 US  DWC_ARCHIVE
3  28eb1a3f-1c15-4a95-931a-4af90ecb574d                 US  DWC_ARCHIVE
4  28eb1a3f-1c15-4a95-931a-4af90ecb574d                 US  DWC_ARCHIVE

                     lastCrawled                      lastParsed  crawlId  ...  \
0  2024-10-18T20:49:13.203+00:00  2024-10-19T13:20:47.596+00:00      492  ...
1  2024-10-18T20:49:13.203+00:00  2024-10-19T13:20:05.769+00:00      492  ...
2  2024-10-18T20:49:13.203+00:00  2024-10-19T13:50:18.609+00:00      492  ...
3  2024-10-18T20:49:13.203+00:00  2024-10-19T13:21:56.812+00:00      492  ...
4  2024-10-18T20:49:13.203+00:00  2024-10-19T13:22:56.714+00:00      492  ...

                                      occurrenceID taxonID catalogNumber  \
0  https://www.inaturalist.org/observations/19563...   42007     195637303
1  https://www.inaturalist.org/observations/19577...  143589     195779125
2  https://www.inaturalist.org/observations/19579...   42007     195792898
3  https://www.inaturalist.org/observations/19580...   42007     195804747
4  https://www.inaturalist.org/observations/19606...   42007     196060253

   institutionCode       eventTime  http://unknown.org/captive  \
0      iNaturalist  13:47:00-07:00                        wild
1      iNaturalist  15:07:00-07:00                        wild
2      iNaturalist  18:25:00-08:00                        wild
3      iNaturalist  17:33:00-08:00                        wild
4      iNaturalist  10:39:00-08:00                        wild

   identificationID  lifeStage  infraspecificEpithet  occurrenceRemarks
0         440102394        NaN                   NaN                NaN
1         440542997      Adult               couguar                NaN
2         440584493        NaN                   NaN                NaN
3         440618271        NaN                   NaN                NaN
4         441358517        NaN                   NaN    Odd drag marks.

[5 rows x 88 columns]
```

In [3]:

```python
# 3. fetching information about the countries available in the GBIF API. This can
import requests
import pandas as pd

# Base URL for GBIF country enumeration
url = 'https://api.gbif.org/v1/enumeration/country'

# Send a GET request to fetch the country list
response = requests.get(url)

# Check if the request was successful
if response.status_code == 200:
```

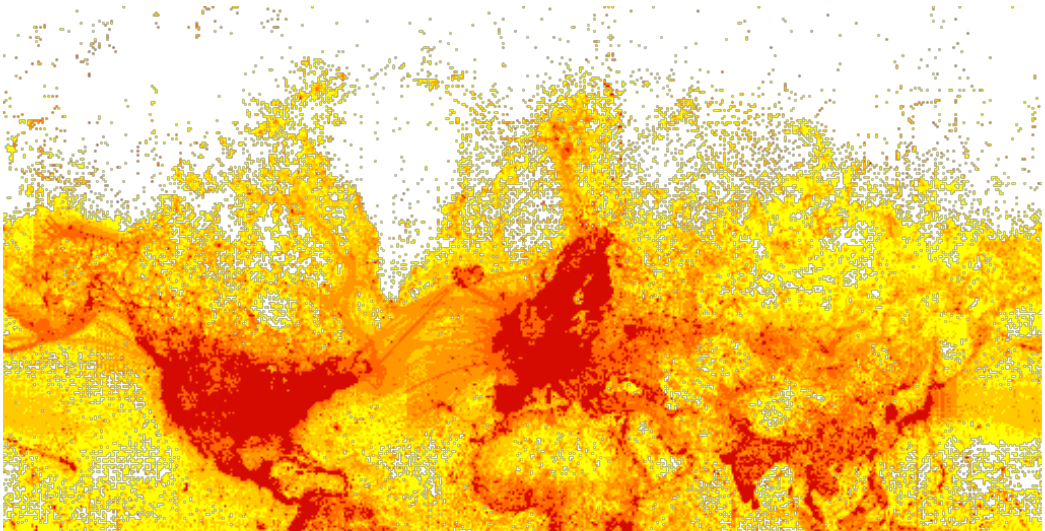
```
    country_data = response.json()

    # Convert the results to a DataFrame
    df_countries = pd.DataFrame(country_data)

    # Display the first few rows of country data
    print("Country Data:")
    print(df_countries.head())
else:
    print(f"Error fetching data: {response.status_code}")
```

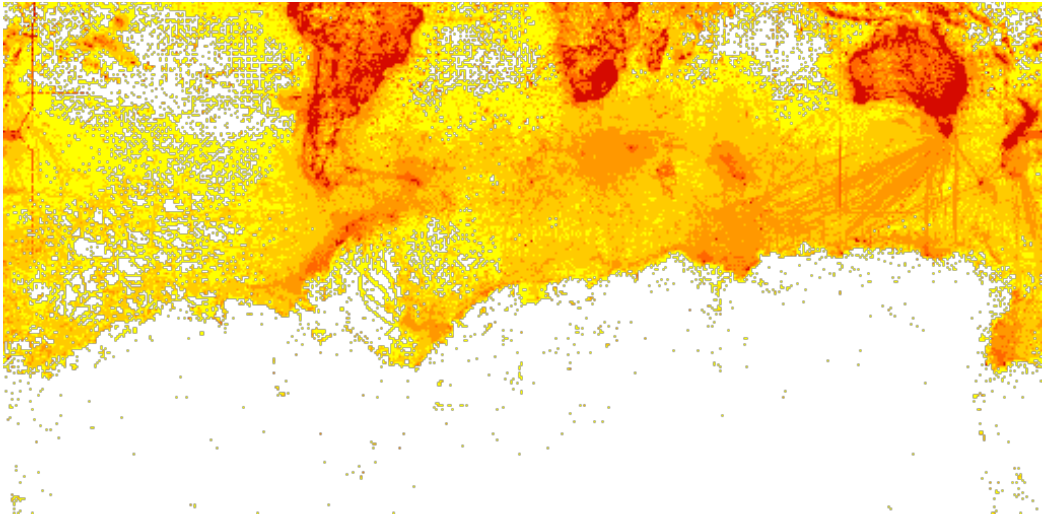
```
Country Data:
  iso2 iso3 isoNumerical           title gbifRegion        enumName
0   AF  AFG            4     Afghanistan       ASIA     AFGHANISTAN
1   AX  ALA          248   Åland Islands     EUROPE   ALAND_ISLANDS
2   AL  ALB            8         Albania     EUROPE         ALBANIA
3   DZ  DZA           12         Algeria     AFRICA         ALGERIA
4   AS  ASM           16  American Samoa    OCEANIA  AMERICAN_SAMOA
```

In [4]:

```
# 4. using the Maps API to generate a map for species occurrences. This will retu
import requests
from IPython.display import Image

# Base URL for the GBIF Maps API
url = 'https://api.gbif.org/v2/map/occurrence/density/{z}/{x}/{y}@1x.png'

# Example parameters for generating the map (you can modify these)
params = {
    'style': 'classic.point',
    'taxonKey': 2435099,  # Puma concolor taxonKey
    'mode': 'GEO_CENTROID',
    'srs': 'EPSG:4326',
    'x': 0,
    'y': 0,
    'z': 0,
}

# Construct the full URL
map_url = url.format(z=params['z'], x=params['x'], y=params['y'])

# Display the map image
Image(url=map_url)
```

Out[4]:





In [5]:

```python
# literature API use to search for peer-reviewed articles that cite GBIF data. Her
import requests
import pandas as pd

# Base URL for GBIF literature search API
url = 'https://api.gbif.org/v1/literature/search'

# Define parameters for the search
params = {
    'q': 'GBIF',
    'limit': 10
}

# Send a GET request to the API
response = requests.get(url, params=params)

# Check if the request was successful
if response.status_code == 200:
```

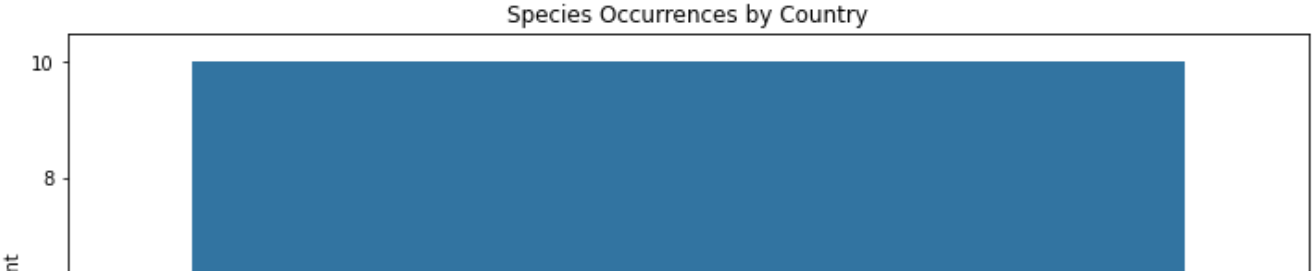
```
    literature_data = response.json()

    # Convert the results to a DataFrame
    df_literature = pd.DataFrame(literature_data['results'])

    # Display the first few rows of literature data
    print("Literature Data:")
    print(df_literature.head())
else:
    print(f"Error fetching data: {response.status_code}")
```

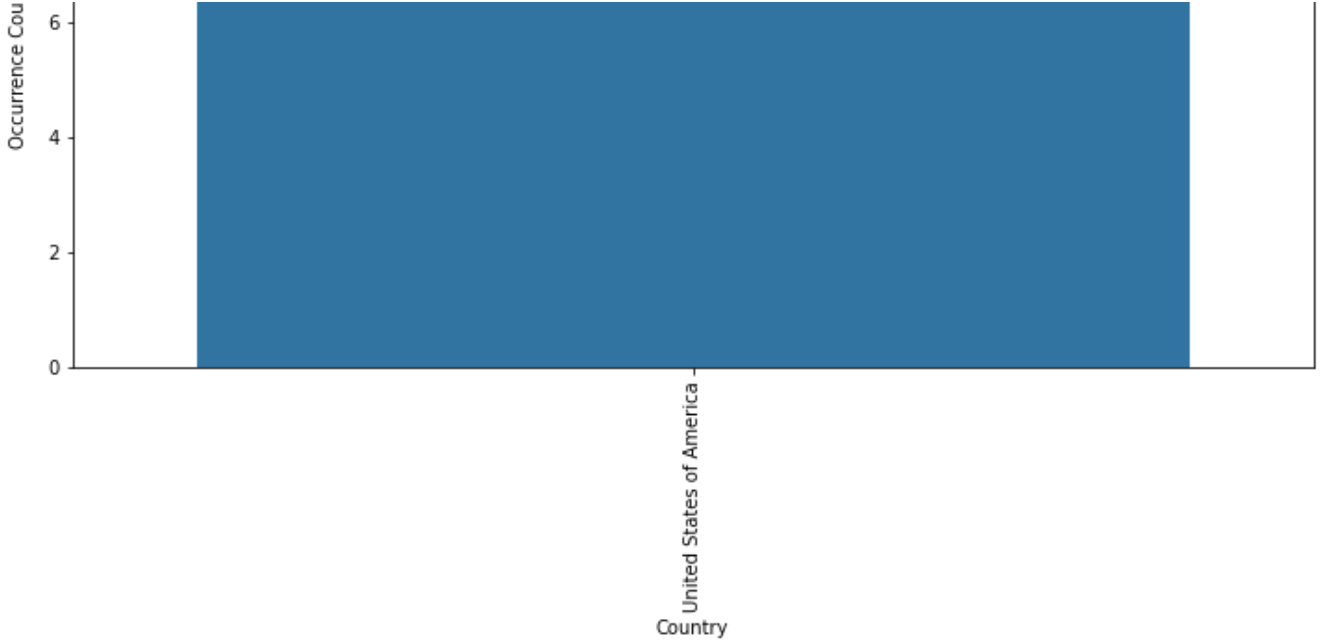
```
Literature Data:
   discovered                                            authors  \
0  2019-06-21  [{'firstName': 'Donald', 'lastName': 'Hobern'}...
1  2015-09-08  [{'firstName': 'Markus', 'lastName': 'Opperman...
2  2019-08-13  [{'firstName': 'Juan Miguel', 'lastName': 'Gon...
3  2022-08-26  [{'firstName': 'John Thomas', 'lastName': 'Wal...
4  2017-07-07  [{'firstName': '大澤', 'lastName': '剛士'}, {'firs...

  countriesOfCoverage countriesOfResearcher                         added  \
0                  []              [AU, DK]  2020-11-24T15:47:07.559+00:00
1                  []                  [DE]  2020-11-24T15:49:59.201+00:00
2                  []          [ES, NL, DK]  2020-11-24T15:47:41.845+00:00
3                  []                  [DK]  2022-09-06T12:41:48.515+00:00
4                  []                  [JP]  2020-11-24T15:51:51.519+00:00

                     published   day gbifDownloadKey gbifOccurrenceKey  \
0  2019-06-21T00:00:00.000+00:00  21.0              []                []
1  2015-01-07T00:00:00.000+00:00   7.0              []                []
2  2019-08-08T00:00:00.000+00:00   8.0              []                []
3  2022-08-23T00:00:00.000+00:00  23.0              []                []
4  2016-01-01T00:00:00.000+00:00   NaN              []                []

  gbifTaxonKey  ...           publisher         relevance  \
0           []  ...  Pensoft Publishers     [GBIF_AUTHOR]
1           []  ...                 NaN      [GBIF_CITED]
2           []  ...  Pensoft Publishers  [GBIF_MENTIONED]
3           []  ...  Pensoft Publishers  [GBIF_DISCUSSED]
4           []  ...       一般社団法人 日本生態学会  [GBIF_DISCUSSED]


                                        source  \
0     Biodiversity Information Science and Standards
1  Database : the journal of biological databases...
2     Biodiversity Information Science and Standards
3     Biodiversity Information Science and Standards
4                                     日本生態学会誌

                                          tags  \
0  [2019, AU, Biodiversity_science, DK, GBIF_auth...
1  [2015, DE, GBIF_cited, Germany, open_access:fa...
2  [2019, DK, Data_management, ES, GBIF_mentioned...
3  [2022, DK, GBIF_discussed, Taxonomy, citation_...
4  [2016, Data_management, GBIF_discussed, JP, Ja...

                                          title                  topics  \
0  An alliance for biodiversity knowledge: Rethin...  [BIODIVERSITY_SCIENCE]
1  GBIS: the information system of the German Gen...                      []
2  Facing e-Biodiversity Challenges Together: GBI...       [DATA_MANAGEMENT]
3    Finding Data Gaps in the GBIF Backbone Taxonomy             [TAXONOMY]
4          日本における生物多様性情報概況 –生物多様性情報概況GBIOの和訳公開と国内動向–       [
DATA_MANAGEMENT]
```

```
DATA_MANAGEMENT]

                       modified  \
0  2022-05-25T15:03:25.891+00:00
1  2020-11-24T15:49:59.201+00:00
2  2022-05-25T15:03:41.879+00:00
3  2022-09-06T12:41:48.515+00:00
4  2022-05-25T14:44:53.852+00:00

                                          websites  year  \
0            [https://doi.org/10.3897/biss.3.37324]  2019
1  [http://database.oxfordjournals.org/content/20...  2015
2            [https://doi.org/10.3897/biss.3.38554]  2019
3            [https://doi.org/10.3897/biss.6.91312]  2022
4  [https://www.jstage.jst.go.jp/article/seitai/6...  2016

                                          abstract
0  There has been major progress over the last tw...
1  The German Federal ex situ Genebank of Agricul...
2  The collaboration between LifeWatch ERIC and D...
3  AbstractWhen publishers supply GBIF (Global Bi...
4  2013 年、生物多様性情報学の世界的な現状と課題をまとめた地球規模生物多様性情報概況（Gl...

[5 rows x 35 columns]
```

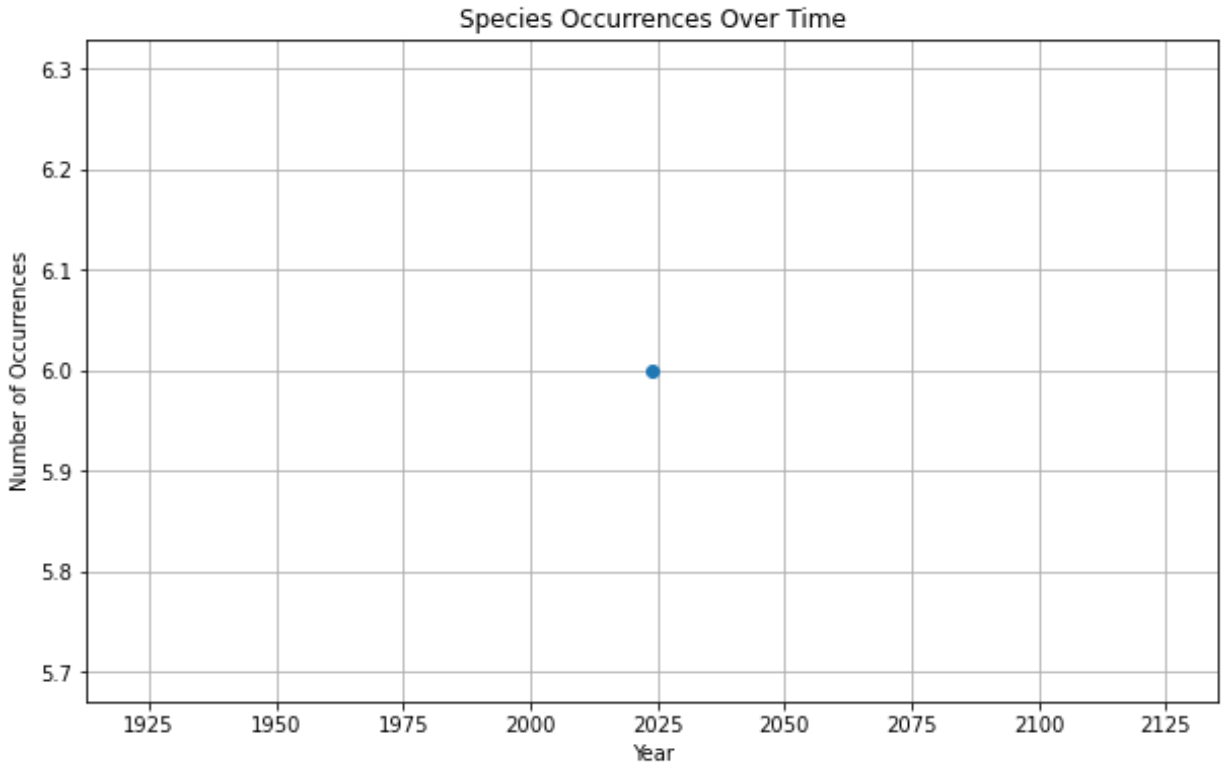
In [26]:

```python
#  Heatmap of Species Occurrences by Country

import matplotlib.pyplot as plt
import seaborn as sns

# Assuming `df_occurrence` contains occurrence data with 'country' and 'occurrence
occurrence_counts = df_occurrence.groupby('country')['key'].count().reset_index()
occurrence_counts.columns = ['Country', 'Occurrence Count']

# Plot heatmap
plt.figure(figsize=(12,6))
sns.barplot(x='Country', y='Occurrence Count', data=occurrence_counts)
plt.xticks(rotation=90)
plt.title('Species Occurrences by Country')
plt.show()
```



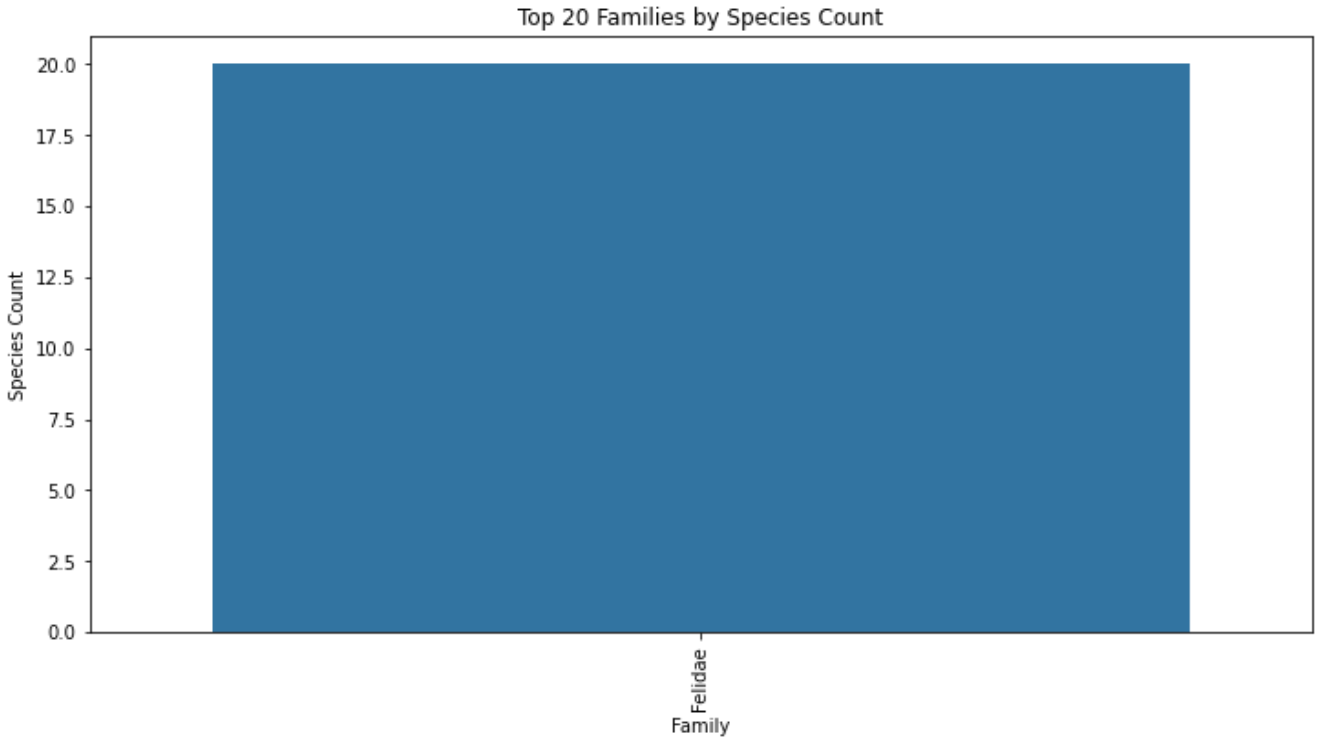


In [27]:
```
# Time Series of Occurrences Over Time
# Assuming df_occurrence has a 'year' column
occurrence_by_year = df_occurrence.groupby('year')['key'].count().reset_index()
occurrence_by_year.columns = ['Year', 'Occurrence Count']

# Plot time series
plt.figure(figsize=(10,6))
plt.plot(occurrence_by_year['Year'], occurrence_by_year['Occurrence Count'], marke
plt.title('Species Occurrences Over Time')
plt.xlabel('Year')
plt.ylabel('Number of Occurrences')
plt.grid(True)
plt.show()
```



In [28]:
```
# Bar Plot of Species by Family
import seaborn as sns

# Assuming df_species has a 'family' column
```

```
# Assuming df_species has a 'family' column
species_by_family = df_species['family'].value_counts().reset_index()
species_by_family.columns = ['Family', 'Species Count']

# Plot bar chart
plt.figure(figsize=(12,6))
sns.barplot(x='Family', y='Species Count', data=species_by_family.head(20))  # Lim
plt.xticks(rotation=90)
plt.title('Top 20 Families by Species Count')
plt.show()
```



In [ ]: