In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
C:\Users\ranig\anaconda3\lib\site-packages\pandas\core\computation\expressions.py:21:
UserWarning: Pandas requires version '2.8.4' or newer of 'numexpr' (version '2.8.1' cu
rrently installed).
  from pandas.core.computation.check import NUMEXPR_INSTALLED
C:\Users\ranig\anaconda3\lib\site-packages\pandas\core\arrays\masked.py:60: UserWarnin
g: Pandas requires version '1.3.6' or newer of 'bottleneck' (version '1.3.4' currently
installed).
  from pandas.core import (
<frozen importlib._bootstrap>:228: RuntimeWarning: scipy._lib.messagestream.MessageStr
eam size changed, may indicate binary incompatibility. Expected 56 from C header, got
64 from PyObject
```

In [3]:
```
df=pd.read_csv('dataset_1.csv')
```

In [4]:
```
df.shape
```

Out[4]:
```
(100000, 50)
```

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

Out[5]:
```
Index(['gbifID', 'datasetKey', 'occurrenceID', 'kingdom', 'phylum', 'class',
       'order', 'family', 'genus', 'species', 'infraspecificEpithet',
       'taxonRank', 'scientificName', 'verbatimScientificName',
       'verbatimScientificNameAuthorship', 'countryCode', 'locality',
       'stateProvince', 'occurrenceStatus', 'individualCount',
       'publishingOrgKey', 'decimalLatitude', 'decimalLongitude',
       'coordinateUncertaintyInMeters', 'coordinatePrecision', 'elevation',
       'elevationAccuracy', 'depth', 'depthAccuracy', 'eventDate', 'day',
       'month', 'year', 'taxonKey', 'speciesKey', 'basisOfRecord',
       'institutionCode', 'collectionCode', 'catalogNumber', 'recordNumber',
       'identifiedBy', 'dateIdentified', 'license', 'rightsHolder',
       'recordedBy', 'typeStatus', 'establishmentMeans', 'lastInterpreted',
       'mediaType', 'issue'],
      dtype='object')
```

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

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[4]:

	gbifID	datasetKey	occurrenceID	kingdom	
0	2423051671	50c9509d-22c7-4a22-a47d-8c48425ef4a7	https://www.inaturalist.org/observations/32655983	Animalia	
1	2423108310	50c9509d-22c7-4a22-a47d-8c48425ef4a7	https://www.inaturalist.org/observations/32869995	Animalia	
2	2423173864	50c9509d-22c7-4a22-a47d-8c48425ef4a7	https://www.inaturalist.org/observations/33126934	Plantae	Tr
3	2423231193	50c9509d-22c7-4a22-a47d-8c48425ef4a7	https://www.inaturalist.org/observations/33361861	Plantae	Tr
4	3408104323	50c9509d-22c7-4a22-a47d-8c48425ef4a7	https://www.inaturalist.org/observations/33628063	Fungi	Ba

5 rows × 50 columns

gbifID: A unique identifier for each record in the Global Biodiversity Information Facility (GBIF) database. datasetKey: The unique identifier for the dataset within GBIF from which the record originates. occurrenceID: A unique identifier for the occurrence record, which can link to individual biodiversity observations. kingdom: The highest taxonomic rank, categorizing life forms into major groups (e.g., Animalia, Plantae, Fungi). phylum: A taxonomic rank below the kingdom and above class, representing major lineages within each kingdom. class: A taxonomic rank below phylum, used to group organisms with similar characteristics. order: A taxonomic rank below class, grouping similar families together. family: A taxonomic rank below order, which groups similar genera. genus: A taxonomic rank below family, grouping species that are structurally similar or related. species: The specific name of the organism, representing the basic unit of biological classification. taxonRank: The rank of the taxon in the hierarchy (e.g., species, genus, family). scientificName: The full scientific name of the organism, typically consisting of the genus and species. verbatimScientificName: The original scientific name as it appeared in the data, which might include author citation details. countryCode: The ISO country code indicating the country where the observation was made. stateProvince: The name of the state, province, or equivalent administrative unit where the observation occurred. occurrenceStatus: Indicates whether the organism was observed as present, absent, or in another status. publishingOrgKey: A unique identifier for the organization that published the dataset. decimalLatitude: The latitude of the observation in decimal degrees. decimalLongitude: The longitude of the observation in decimal degrees. coordinateUncertaintyInMeters: The uncertainty of the coordinates in meters, representing the potential error in the location. eventDate: The date when the observation occurred. day: The day of the month when the observation occurred. month: The month when the observation occurred. year: The year when the observation occurred. taxonKey: A unique identifier for the taxon in the GBIF taxonomy database. speciesKey: A unique identifier for the species in the GBIF taxonomy database. basisOfRecord: The type of record (e.g., preserved specimen, observation, human observation, machine observation). institutionCode: A code representing the institution where the record originated (e.g., museum, research institution). collectionCode: The collection code within the institution, indicating which specific collection the record belongs to. catalogNumber: The catalog number associated with the specimen or observation. identifiedBy: The person or organization that identified the species or taxon. dateIdentified: The date when the species or taxon was identified. license: The license under which the data is published (e.g., Creative Commons licenses). rightsHolder: The person or organization that holds the rights to the data. recordedBy: The person or organization that recorded the occurrence. lastInterpreted: The date when the record was last interpreted by GBIF. mediaType: The type of media associated with the record (e.g., image, audio). issue: Any issues or flags associated with the record, such as missing data, errors in location, or taxonomic misidentification.

In [5]:
```
df.isnull().sum()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[5]:
```
gbifID                                     0
datasetKey                                 0
occurrenceID                               0
kingdom                                    0
phylum                                    13
class                                   1343
order                                   3067
family                                    61
genus                                     92
species                                 1290
infraspecificEpithet                   96824
taxonRank                                  0
scientificName                             0
verbatimScientificName                     0
verbatimScientificNameAuthorship      100000
countryCode                              304
locality                              100000
stateProvince                            482
occurrenceStatus                           0
individualCount                       100000
publishingOrgKey                           0
decimalLatitude                          217
decimalLongitude                         217
coordinateUncertaintyInMeters          19430
coordinatePrecision                   100000
elevation                             100000
elevationAccuracy                     100000
depth                                 100000
depthAccuracy                         100000
eventDate                                  0
day                                        0
month                                      0
year                                       0
taxonKey                                   0
speciesKey                              1290
basisOfRecord                              0
institutionCode                            0
collectionCode                             0
catalogNumber                              0
recordNumber                          100000
identifiedBy                             416
dateIdentified                           416
license                                    0
rightsHolder                               0
recordedBy                                 0
typeStatus                            100000
establishmentMeans                    100000
lastInterpreted                            0
mediaType                               3726
issue                                      0
dtype: int64
```

In [6]:
```
df.info()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 50 columns):
 #   Column                            Non-Null Count   Dtype
---  ------                            --------------   -----
 0   gbifID                            100000 non-null  int64
 1   datasetKey                        100000 non-null  object
 2   occurrenceID                      100000 non-null  object
 3   kingdom                           100000 non-null  object
 4   phylum                            99987 non-null   object
 5   class                             98657 non-null   object
 6   order                             96933 non-null   object
 7   family                            99939 non-null   object
 8   genus                             99908 non-null   object
 9   species                           98710 non-null   object
 10  infraspecificEpithet              3176 non-null    object
 11  taxonRank                         100000 non-null  object
 12  scientificName                    100000 non-null  object
 13  verbatimScientificName            100000 non-null  object
 14  verbatimScientificNameAuthorship  0 non-null       float64
 15  countryCode                       99696 non-null   object
 16  locality                          0 non-null       float64
 17  stateProvince                     99518 non-null   object
 18  occurrenceStatus                  100000 non-null  object
 19  individualCount                   0 non-null       float64
 20  publishingOrgKey                  100000 non-null  object
 21  decimalLatitude                   99783 non-null   float64
 22  decimalLongitude                  99783 non-null   float64
 23  coordinateUncertaintyInMeters     80570 non-null   float64
 24  coordinatePrecision               0 non-null       float64
 25  elevation                         0 non-null       float64
 26  elevationAccuracy                 0 non-null       float64
 27  depth                             0 non-null       float64
 28  depthAccuracy                     0 non-null       float64
 29  eventDate                         100000 non-null  object
 30  day                               100000 non-null  int64
 31  month                             100000 non-null  int64
 32  year                              100000 non-null  int64
 33  taxonKey                          100000 non-null  int64
 34  speciesKey                        98710 non-null   float64
 35  basisOfRecord                     100000 non-null  object
 36  institutionCode                   100000 non-null  object
 37  collectionCode                    100000 non-null  object
 38  catalogNumber                     100000 non-null  int64
 39  recordNumber                      0 non-null       float64
 40  identifiedBy                      99584 non-null   object
 41  dateIdentified                    99584 non-null   object
 42  license                           100000 non-null  object
 43  rightsHolder                      100000 non-null  object
 44  recordedBy                        100000 non-null  object
 45  typeStatus                        0 non-null       float64
 46  establishmentMeans                0 non-null       float64
 47  lastInterpreted                   100000 non-null  object
 48  mediaType                         96274 non-null   object
 49  issue                             100000 non-null  object
dtypes: float64(15), int64(6), object(29)
memory usage: 38.1+ MB
```

In [7]:
```
# find out the null columns is max or 100 percent null columns find it
null_col=df.columns[df.isnull().sum()>=50000]
```

In [8]:
```
# this 11 columns is empty of drop this columns
df=df.drop(null_col,axis=1)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[9]: (100000, 38)

In [10]:
```
df.dtypes
```

Out[10]:
```
gbifID                            int64
datasetKey                       object
occurrenceID                     object
kingdom                          object
phylum                           object
class                            object
order                            object
family                           object
genus                            object
species                          object
taxonRank                        object
scientificName                   object
verbatimScientificName           object
countryCode                      object
stateProvince                    object
occurrenceStatus                 object
publishingOrgKey                 object
decimalLatitude                 float64
decimalLongitude                float64
coordinateUncertaintyInMeters   float64
eventDate                        object
day                               int64
month                             int64
year                              int64
taxonKey                          int64
speciesKey                      float64
basisOfRecord                    object
institutionCode                  object
collectionCode                   object
catalogNumber                     int64
identifiedBy                     object
dateIdentified                   object
license                          object
rightsHolder                     object
recordedBy                       object
lastInterpreted                  object
mediaType                        object
issue                            object
dtype: object
```

In [11]:
```
columns=df.columns
columns
```

Out[11]:
```
Index(['gbifID', 'datasetKey', 'occurrenceID', 'kingdom', 'phylum', 'class',
       'order', 'family', 'genus', 'species', 'taxonRank', 'scientificName',
       'verbatimScientificName', 'countryCode', 'stateProvince',
       'occurrenceStatus', 'publishingOrgKey', 'decimalLatitude',
       'decimalLongitude', 'coordinateUncertaintyInMeters', 'eventDate', 'day',
       'month', 'year', 'taxonKey', 'speciesKey', 'basisOfRecord',
       'institutionCode', 'collectionCode', 'catalogNumber', 'identifiedBy',
       'dateIdentified', 'license', 'rightsHolder', 'recordedBy',
       'lastInterpreted', 'mediaType', 'issue'],
      dtype='object')
```

In [12]:
```
# remove  this columns is same information this columns
for i in columns:
    nuique_val_col=len(df[i].unique())
    print(f'{i}\n:{nuique_val_col}')
    if nuique_val_col<=3:
        df=df.drop(i,axis=1)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

gbifID
:100000
datasetKey
:1
occurrenceID
:100000
kingdom
:7
phylum
:38
class
:103
order
:469
family
:2177
genus
:11979
species
:26677
taxonRank
:6
scientificName
:28485
verbatimScientificName
:28673
countryCode
:220
stateProvince
:1846
occurrenceStatus
:1
publishingOrgKey
:1
decimalLatitude
:97093
decimalLongitude
:97088
coordinateUncertaintyInMeters
:5529
eventDate
:97664
day
:31
month
:12
year
:58
taxonKey
:28485
speciesKey
:26677
basisOfRecord
:1
institutionCode
:1
collectionCode
:1
catalogNumber
:100000
identifiedBy
:33530
dateIdentified
:99503
license
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
rightsHolder
:39712
recordedBy
:39712
lastInterpreted
:99423
mediaType
:53
issue
:21
```

In [13]:
```
df=df.drop(columns=['gbifID','occurrenceID'],axis=1)
```

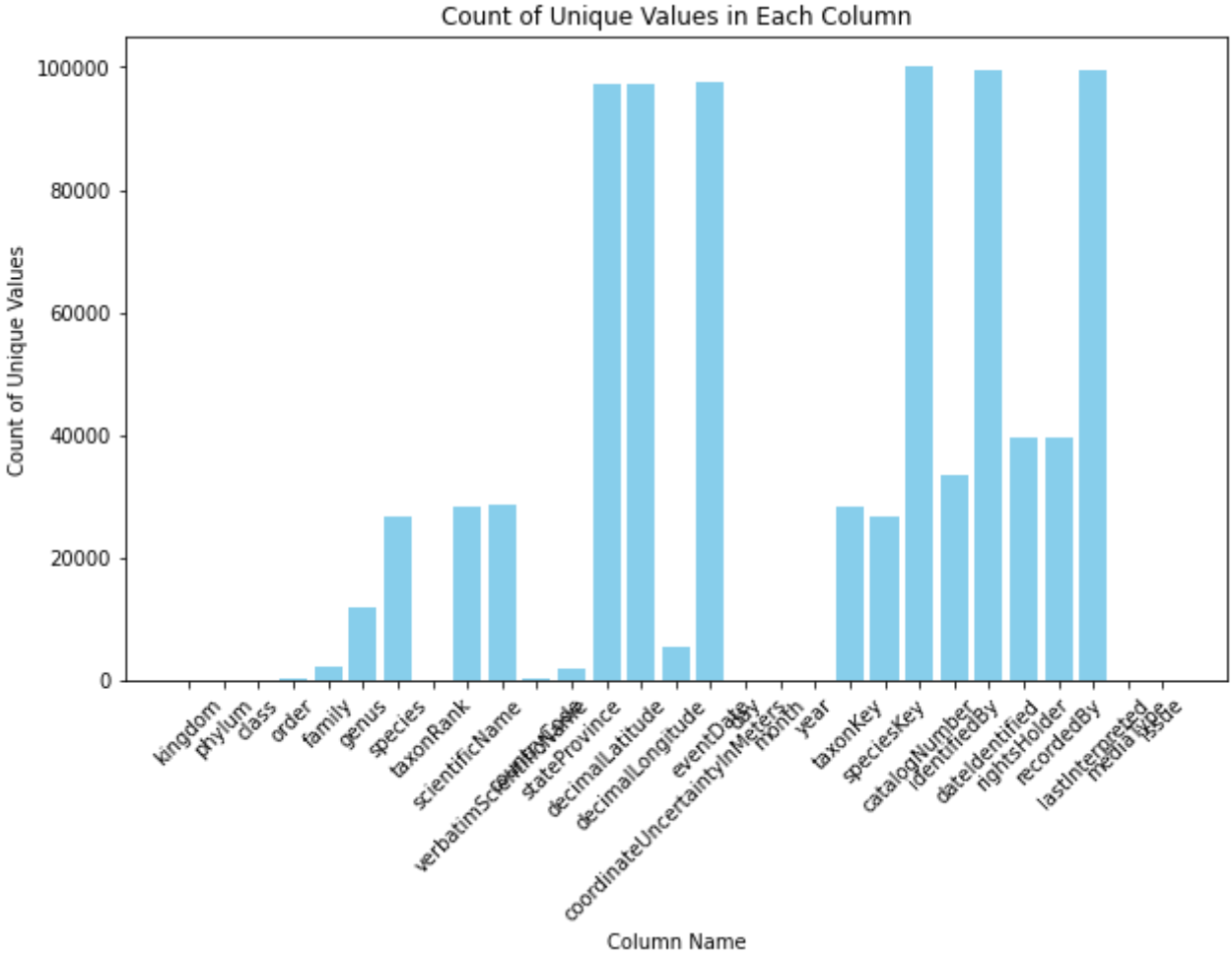
In [14]:
```
columns=df.columns
```

In [15]:
```
import matplotlib.pyplot as plt

# Create a dictionary to store the count of unique values for each column
unique_val_count = {}

# Iterate over the columns
for i in columns:
    nuique_val_col = len(df[i].unique())
    unique_val_count[i] = nuique_val_col
    print(f'{i}\n:{nuique_val_col}')
    if nuique_val_col <= 3:
        df = df.drop(i, axis=1)

# Create a bar plot of the unique values
plt.figure(figsize=(10, 6))
plt.bar(unique_val_count.keys(), unique_val_count.values(), color="skyblue")
plt.xlabel('Column Name')
plt.ylabel('Count of Unique Values')
plt.title('Count of Unique Values in Each Column')
plt.xticks(rotation=45)
plt.show()
plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

kingdom
:7
phylum
:38
class
:103
order
:469
family
:2177
genus
:11979
species
:26677
taxonRank
:6
scientificName
:28485
verbatimScientificName
:28673
countryCode
:220
stateProvince
:1846
decimalLatitude
:97093
decimalLongitude
:97088
coordinateUncertaintyInMeters
:5529
eventDate
:97664
day
:31
month
:12
year
:58
taxonKey
:28485
speciesKey
:26677
catalogNumber
:100000
identifiedBy
:33530
dateIdentified
:99503
rightsHolder
:39712
recordedBy
:39712
lastInterpreted
:99423
mediaType
:53
issue
:21

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [16]:
```
columns=df.columns
columns
```

Out[16]:
```
Index(['kingdom', 'phylum', 'class', 'order', 'family', 'genus', 'species',
       'taxonRank', 'scientificName', 'verbatimScientificName', 'countryCode',
       'stateProvince', 'decimalLatitude', 'decimalLongitude',
       'coordinateUncertaintyInMeters', 'eventDate', 'day', 'month', 'year',
       'taxonKey', 'speciesKey', 'catalogNumber', 'identifiedBy',
       'dateIdentified', 'rightsHolder', 'recordedBy', 'lastInterpreted',
       'mediaType', 'issue'],
      dtype='object')
```

In [17]:
```
df.shape
```

Out[17]:
```
(100000, 29)
```

In [18]:
```
df.info()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 29 columns):
 #   Column                        Non-Null Count   Dtype
---  ------                        --------------   -----
 0   kingdom                       100000 non-null  object
 1   phylum                        99987 non-null   object
 2   class                         98657 non-null   object
 3   order                         96933 non-null   object
 4   family                        99939 non-null   object
 5   genus                         99908 non-null   object
 6   species                       98710 non-null   object
 7   taxonRank                     100000 non-null  object
 8   scientificName                100000 non-null  object
 9   verbatimScientificName        100000 non-null  object
 10  countryCode                   99696 non-null   object
 11  stateProvince                 99518 non-null   object
 12  decimalLatitude               99783 non-null   float64
 13  decimalLongitude              99783 non-null   float64
 14  coordinateUncertaintyInMeters 80570 non-null   float64
 15  eventDate                     100000 non-null  object
 16  day                           100000 non-null  int64
 17  month                         100000 non-null  int64
 18  year                          100000 non-null  int64
 19  taxonKey                      100000 non-null  int64
 20  speciesKey                    98710 non-null   float64
 21  catalogNumber                 100000 non-null  int64
 22  identifiedBy                  99584 non-null   object
 23  dateIdentified                99584 non-null   object
 24  rightsHolder                  100000 non-null  object
 25  recordedBy                    100000 non-null  object
 26  lastInterpreted               100000 non-null  object
 27  mediaType                     96274 non-null   object
 28  issue                         100000 non-null  object
dtypes: float64(4), int64(5), object(20)
memory usage: 22.1+ MB
```

In [19]:
```
df.isnull().sum()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[19]:
```
kingdom                            0
phylum                            13
class                           1343
order                           3067
family                            61
genus                             92
species                         1290
taxonRank                          0
scientificName                     0
verbatimScientificName             0
countryCode                      304
stateProvince                    482
decimalLatitude                  217
decimalLongitude                 217
coordinateUncertaintyInMeters  19430
eventDate                          0
day                                0
month                              0
year                               0
taxonKey                           0
speciesKey                      1290
catalogNumber                      0
identifiedBy                     416
dateIdentified                   416
rightsHolder                       0
recordedBy                         0
lastInterpreted                    0
mediaType                       3726
issue                              0
dtype: int64
```

rows missing value handle

In [20]:
```
# coordinateUncertaintyInMeters this rows is 19430 rows is NA so remove this  empty r
df=df.dropna(subset=['coordinateUncertaintyInMeters'],axis=0)
```

In [21]:
```
df.isnull().sum()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[21]:
```
kingdom                            0
phylum                            11
class                           1144
order                           2661
family                            51
genus                             77
species                         1039
taxonRank                          0
scientificName                     0
verbatimScientificName             0
countryCode                      290
stateProvince                    437
decimalLatitude                  217
decimalLongitude                 217
coordinateUncertaintyInMeters      0
eventDate                          0
day                                0
month                              0
year                               0
taxonKey                           0
speciesKey                      1039
catalogNumber                      0
identifiedBy                     357
dateIdentified                   357
rightsHolder                       0
recordedBy                         0
lastInterpreted                    0
mediaType                       2842
issue                              0
dtype: int64
```

In [22]:
```
# find out the numerical columns and categorical columns
num_col=[]
cat_col=[]
for i in columns:
    if df[i].dtypes=='object':
        cat_col.append(i)
    else:
        num_col.append(i)
```

In [23]:
```
# Fill missing values in numerical columns with the fill using  mean of respective co
for col in num_col:
    df[col].fillna(df[col].mean(), inplace=True)

# Fill missing values in categorical columns with the mode and their rspective column
for col in cat_col:
    df[col].fillna(df[col].mode()[0], inplace=True)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\1927919631.py:3: FutureWarning: A val
ue is trying to be set on a copy of a DataFrame or Series through chained assignment u
sing an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because th
e intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({c
ol: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the
operation inplace on the original object.


  df[col].fillna(df[col].mean(), inplace=True)
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\1927919631.py:7: FutureWarning: A val
ue is trying to be set on a copy of a DataFrame or Series through chained assignment u
sing an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because th
e intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({c
ol: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the
operation inplace on the original object.


  df[col].fillna(df[col].mode()[0], inplace=True)
```

In [24]:
```
df.isnull().sum()
```

Out[24]:
```
kingdom                        0
phylum                         0
class                          0
order                          0
family                         0
genus                          0
species                        0
taxonRank                      0
scientificName                 0
verbatimScientificName         0
countryCode                    0
stateProvince                  0
decimalLatitude                0
decimalLongitude               0
coordinateUncertaintyInMeters  0
eventDate                      0
day                            0
month                          0
year                           0
taxonKey                       0
speciesKey                     0
catalogNumber                  0
identifiedBy                   0
dateIdentified                 0
rightsHolder                   0
recordedBy                     0
lastInterpreted                0
mediaType                      0
issue                          0
dtype: int64
```

In [25]:
```
df.shape
```

Out[25]:
```
(80570, 29)
```

univariate analysis

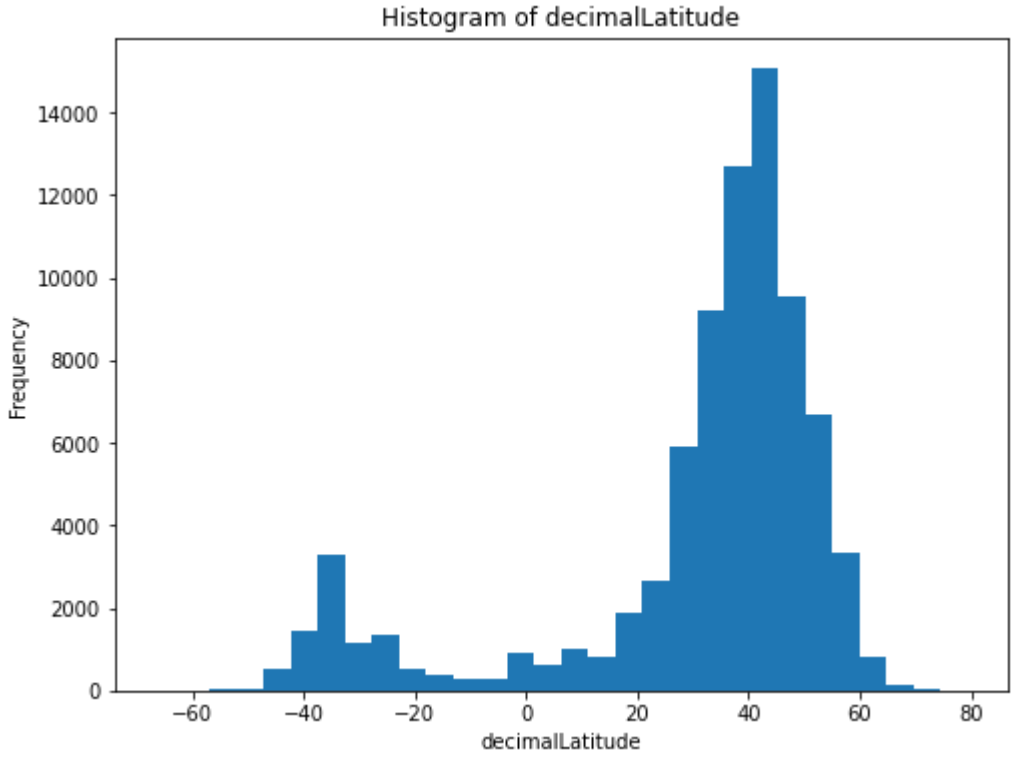
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

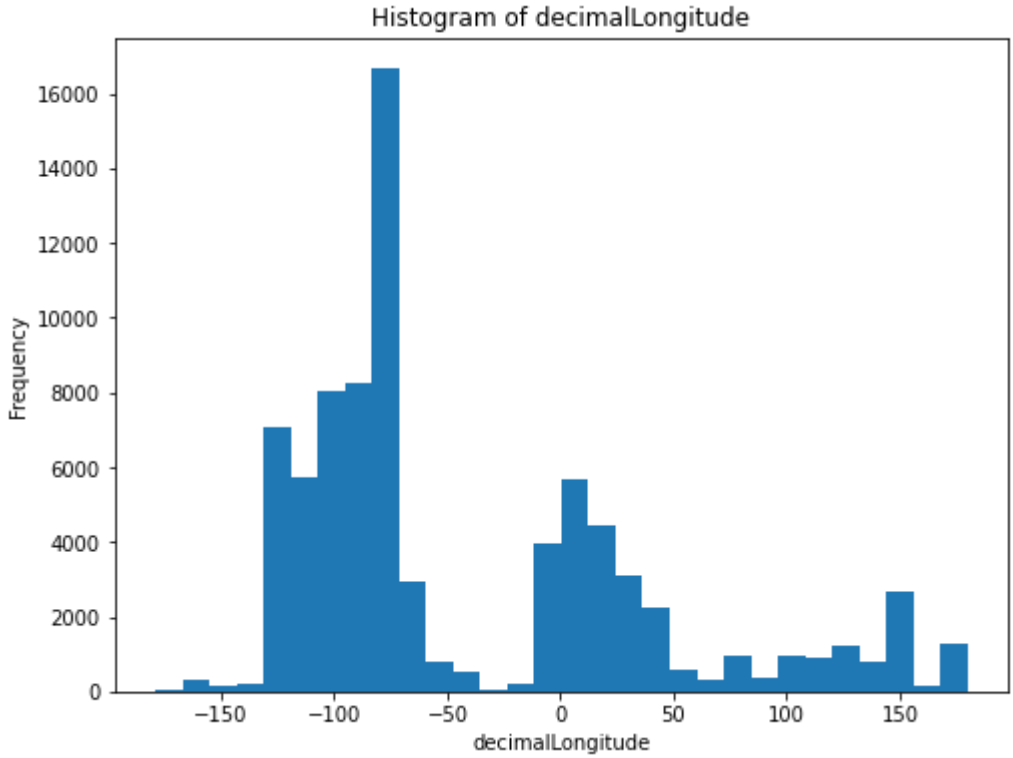
In [26]:
```python
# Numerical columns: Descriptive statistics and plots
print("Descriptive Statistics for Numerical Columns:")
print(df[num_col].describe())
```

```
Descriptive Statistics for Numerical Columns:
       decimalLatitude  decimalLongitude  coordinateUncertaintyInMeters  \
count     80570.000000      80570.000000                   8.057000e+04
mean         31.003953        -39.352729                   5.962562e+03
std          25.073817         79.751012                   8.782091e+04
min         -66.674369       -179.199142                   1.000000e+00
25%          29.519292        -96.799978                   8.000000e+00
50%          38.884382        -76.531791                   3.100000e+01
75%          45.426494         12.045415                   2.690000e+02
max          79.386183        179.998902                   1.150145e+07

                day         month          year      taxonKey    speciesKey  \
count  80570.000000  80570.000000  80570.000000  8.057000e+04  8.057000e+04
mean      15.828398      6.393273   2020.266476  4.253757e+06  4.146885e+06
std        8.973259      2.779570      3.143644  2.465209e+06  2.393786e+06
min        1.000000      1.000000   1935.000000  2.393000e+03  1.000850e+06
25%        8.000000      4.000000   2019.000000  2.480953e+06  2.480651e+06
50%       16.000000      6.000000   2021.000000  3.085413e+06  3.042291e+06
75%       24.000000      8.000000   2022.000000  5.371685e+06  5.331779e+06
max       31.000000     12.000000   2023.000000  1.238014e+07  1.236798e+07

       catalogNumber
count   8.057000e+04
mean    9.208620e+07
std     5.222475e+07
min     5.711000e+03
25%     4.742888e+07
50%     9.234465e+07
75%     1.371369e+08
max     1.840407e+08
```

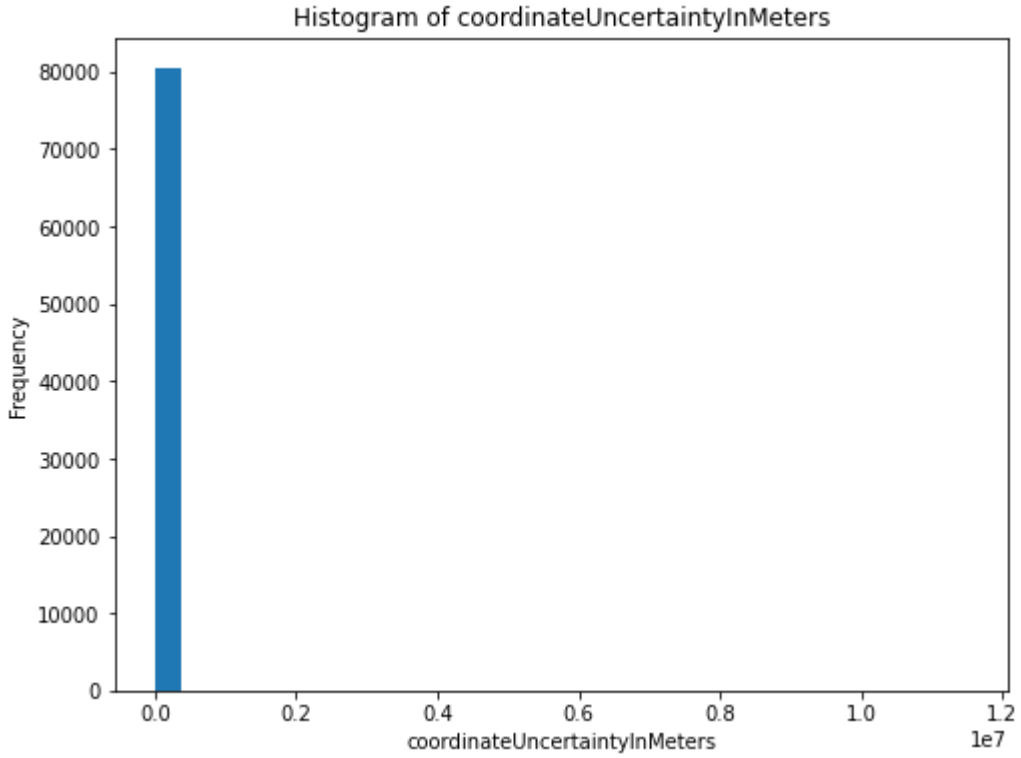
In [27]:
```python
# Plot histograms for each numerical column
import matplotlib.pyplot as plt

# Plot histograms for each numerical column
for col in num_col:
    plt.figure(figsize=(8, 6))
    plt.hist(df[col],bins=30)
    plt.title(f'Histogram of {col}')
    plt.xlabel(col)
    plt.ylabel('Frequency')
    plt.show()
```

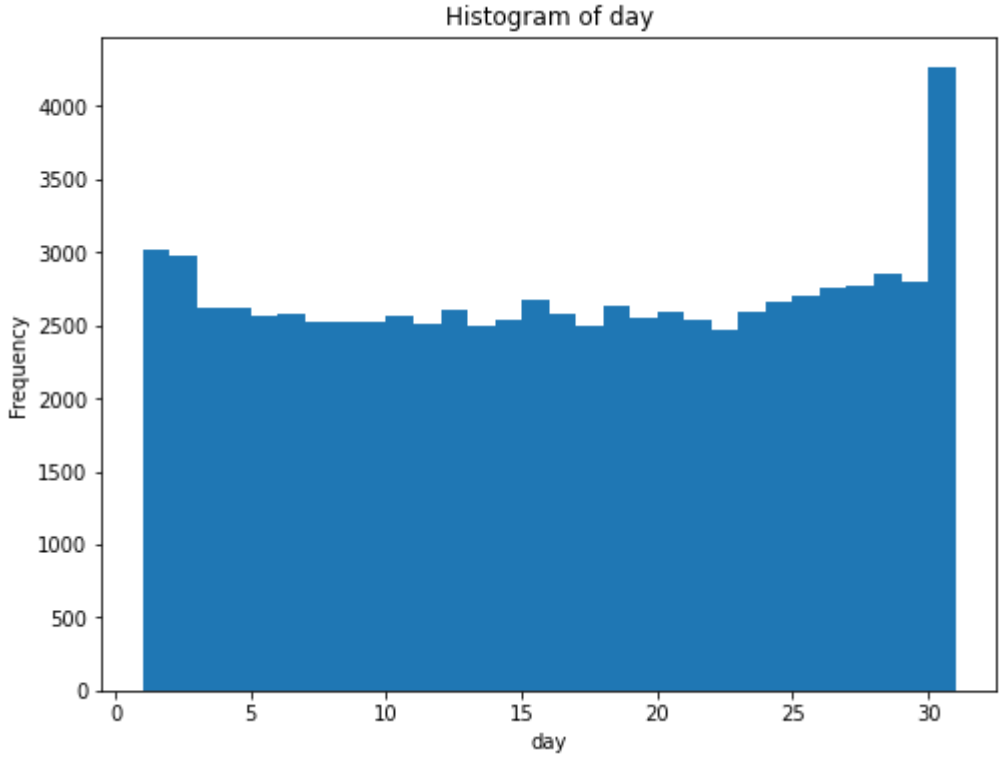
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



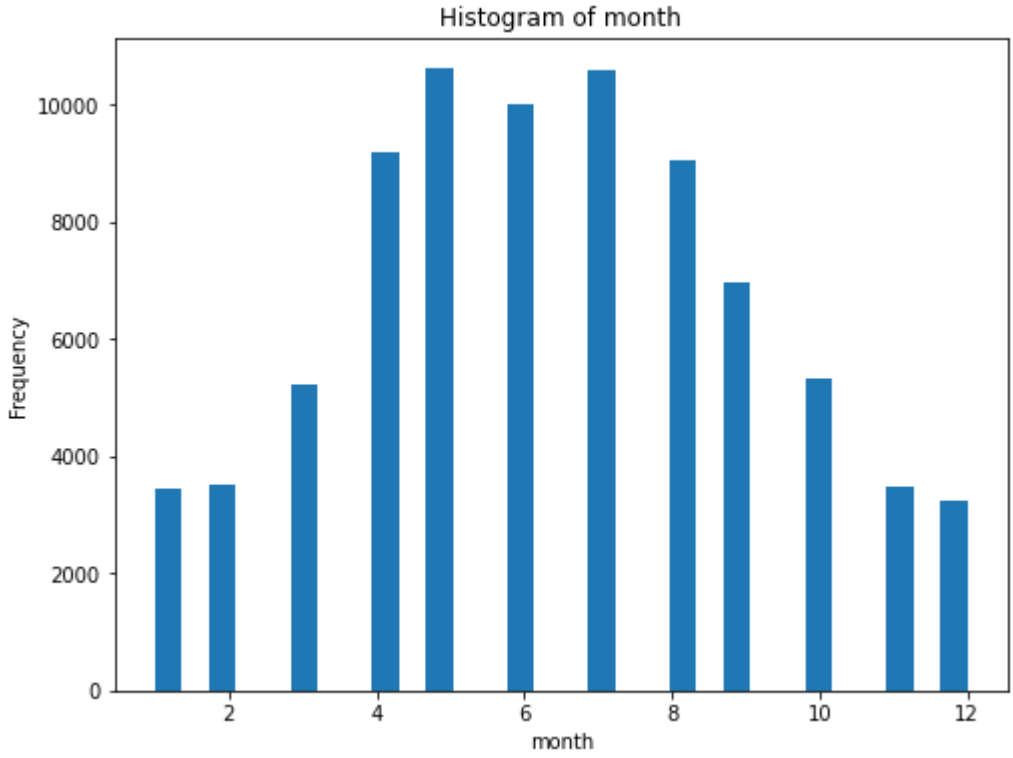


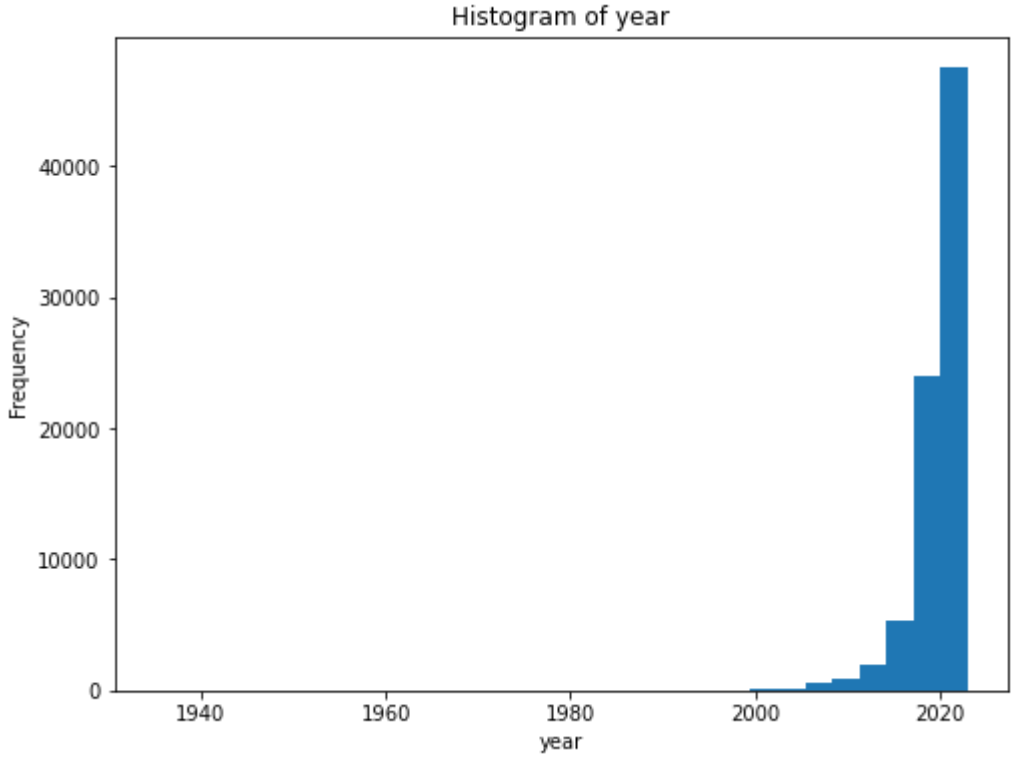
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



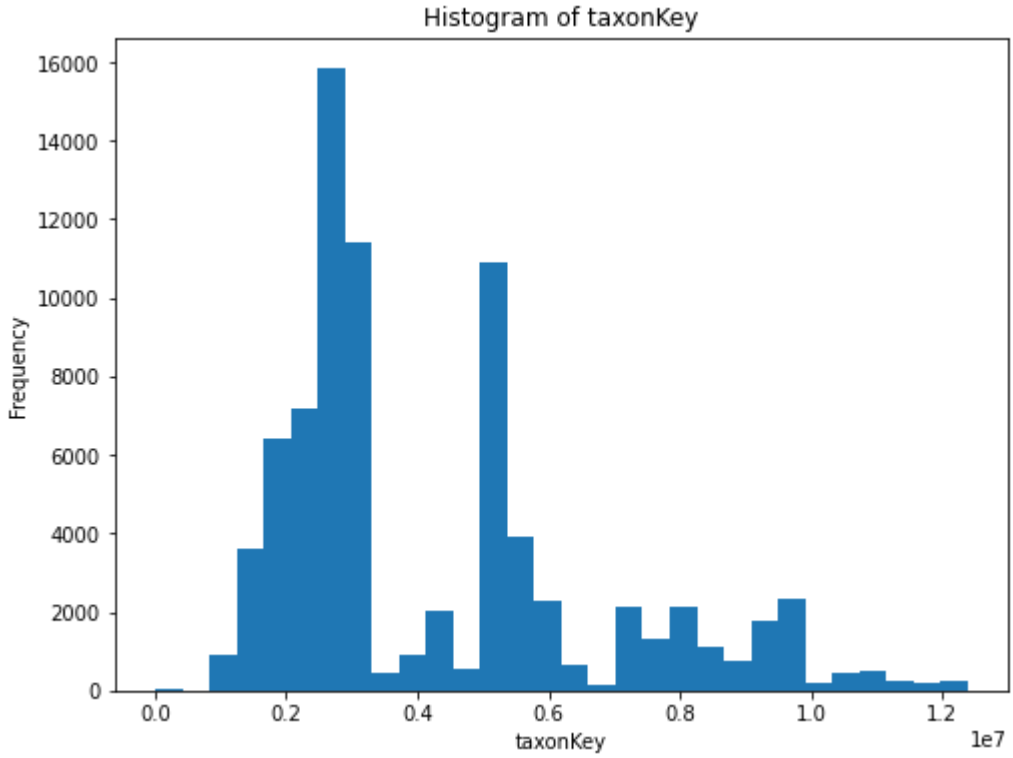


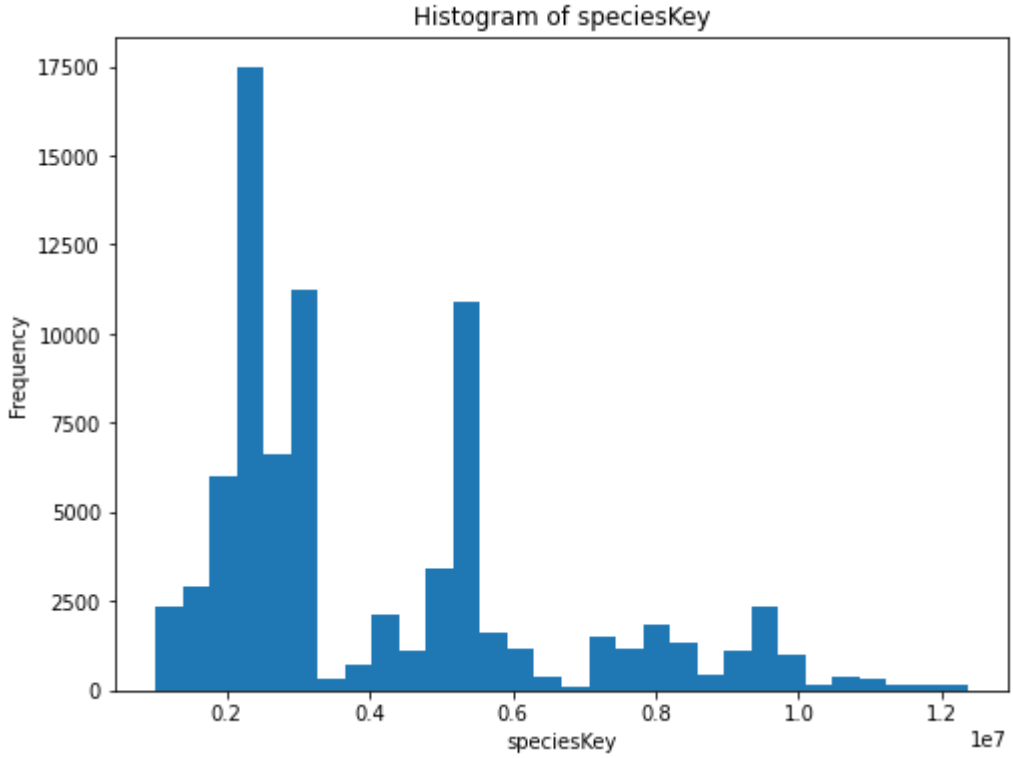
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



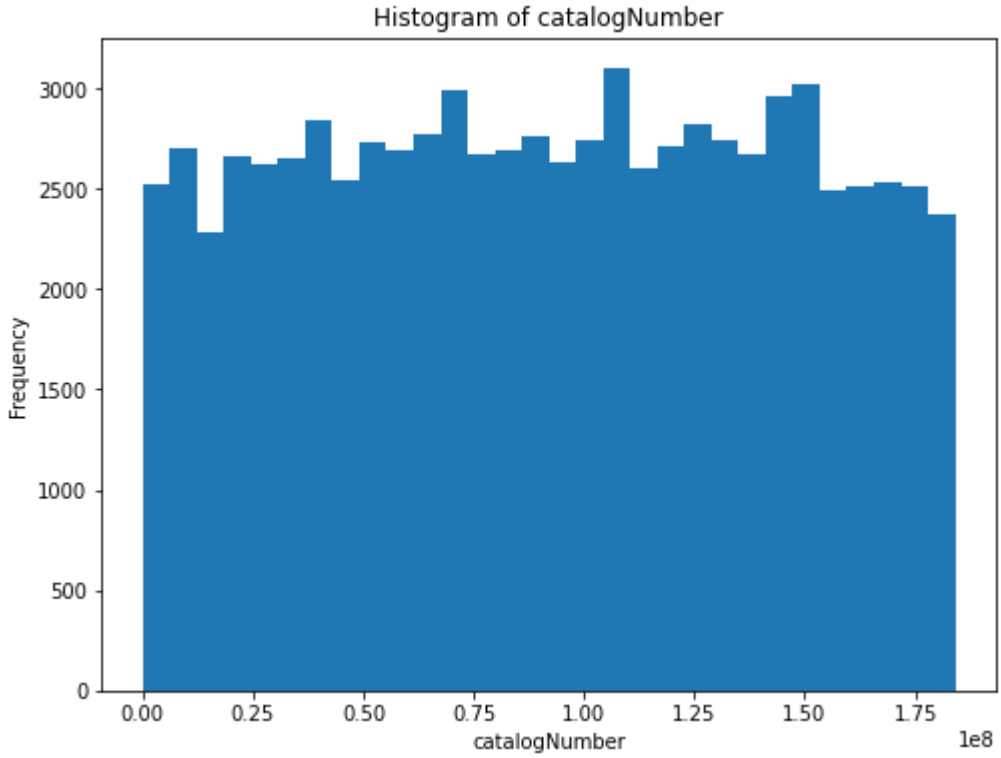


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

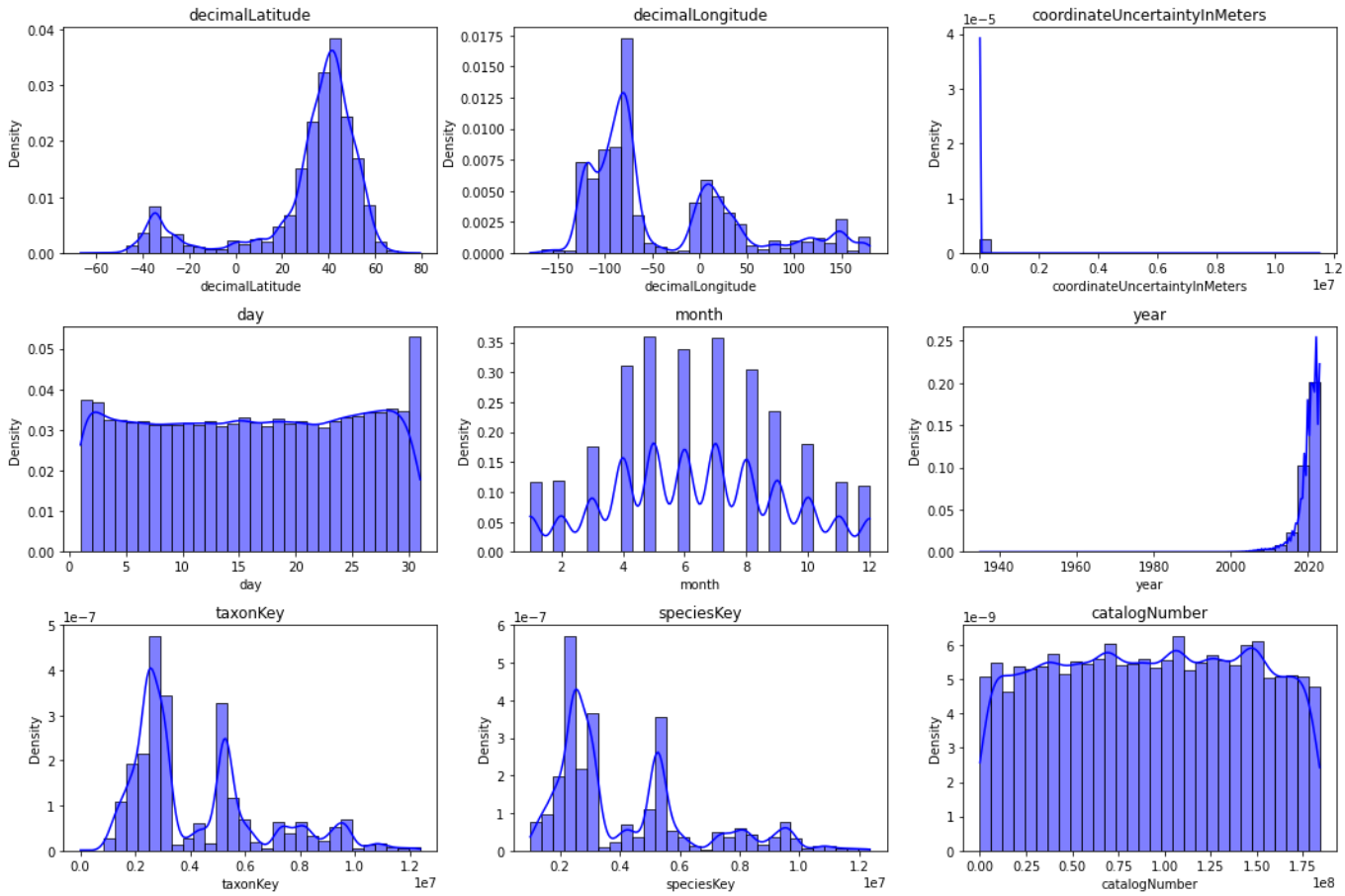




Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



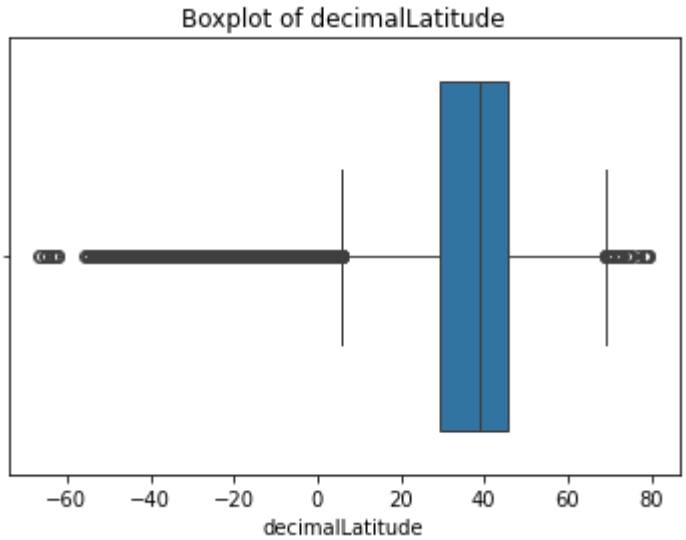
In [28]:
```
# Assuming num_col is defined and df is your DataFrame
fig, axs = plt.subplots(3,3, figsize=(15,10))
axs = axs.flatten()
for i, col in enumerate(num_col):
    sns.histplot(df[col], ax=axs[i], bins=30, kde=True, color='blue', stat='density')
    axs[i].set_title(col)
plt.tight_layout()
plt.show()
```



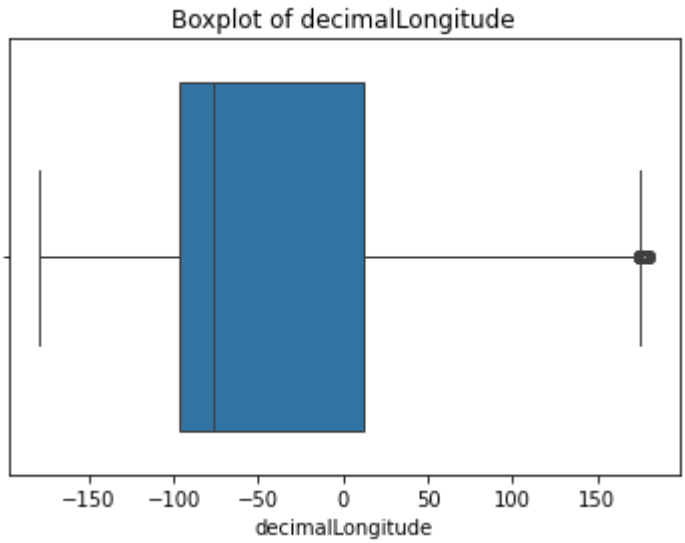
In [29]:
```
# find out the outlier of numerical data sets using boxplot

for i in num_col:
    # Calculate Q1, Q3, and IQR
    Q1 = df[i].quantile(0.25)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
    Q3 = df[i].quantile(0.75)
    IQR = Q3 - Q1

    # Define the outlier range
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    # Find outliers
    outliers = df[(df[i] < lower_bound) | (df[i] > upper_bound)]

    # Display the boxplot
    sns.boxplot(x=df[i])
    plt.title(f'Boxplot of {i}')
    plt.show()

    # Print information about outliers
    print(f"Column: {i}")
    print(f"Number of outliers: {outliers.shape[0]}")
    print(f"Outliers:\n{outliers[i].values}\n")
```

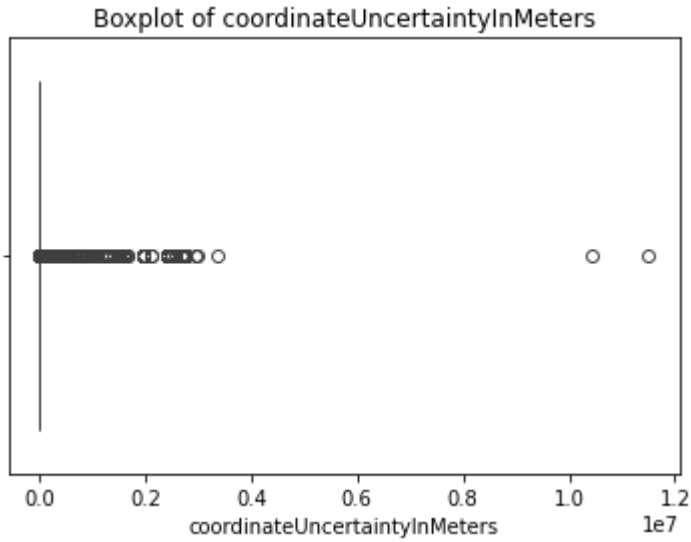



```
Column: decimalLatitude
Number of outliers: 10725
Outliers:
[-17.923595 -20.140961 -37.829741 ... -33.681979 -23.897306 -45.605386]
```



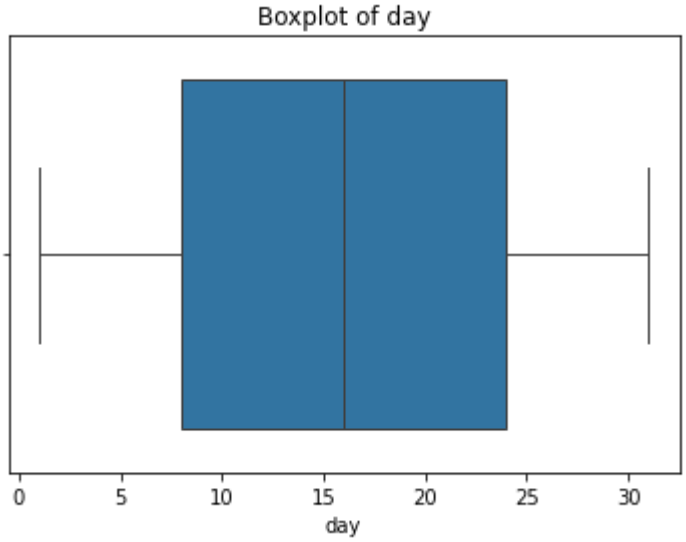
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
Column: decimalLongitude
Number of outliers: 196
Outliers:
[176.252445 175.922867 176.127987 178.146928 175.350005 176.418933
 176.004877 175.316742 176.054489 177.185415 175.500996 176.195984
 175.319842 176.500033 176.124272 176.896169 176.89109  175.578504
 175.747458 175.488911 177.229299 177.416667 176.073157 176.098029
 175.71353  179.088033 176.695519 175.627184 175.759705 177.3874
 175.65732  175.815606 175.683917 175.549592 175.745768 176.310563
 177.317543 175.665645 178.456847 175.544002 175.782328 176.017042
 176.078347 175.458577 177.176157 176.015048 175.665482 175.927303
 175.743624 176.216935 176.192347 175.838546 177.657546 177.885954
 175.556882 176.0864   175.618315 175.530409 176.006479 176.896244
 178.065032 175.423917 175.681817 176.277588 177.215719 175.566371
 175.50413  175.701235 175.747087 175.4328   178.090482 176.002307
 175.845488 175.697141 175.885348 176.768131 177.425369 177.320816
 176.303456 175.506728 175.595484 176.262199 175.487018 177.034145
 176.104674 177.315249 176.896184 175.953098 175.710799 175.680107
 177.178115 175.64131  176.2163   176.855835 175.325509 175.409441
 175.757699 177.18506  175.663943 176.896447 175.813111 175.693725
 177.194521 176.239752 176.149755 176.170309 175.715255 175.690551
 178.548313 175.826062 176.219083 175.677983 177.348827 175.953265
 177.215581 176.712631 176.072148 175.752045 176.756069 176.990385
 176.32808  177.081145 175.612291 175.875245 175.548862 179.998902
 175.554803 176.245536 175.812149 175.747777 176.58195  179.317077
 175.393581 176.284717 176.036962 175.691492 175.772898 176.218108
 175.684159 176.251066 175.809964 178.096726 177.415466 178.450343
 177.108869 178.034517 177.884943 177.793578 175.549948 175.686063
 176.089408 175.587416 176.072008 176.219114 175.617557 176.907949
 175.627306 175.921833 175.868797 176.928991 175.56851  175.850113
 176.977529 175.627352 175.358084 176.113684 175.950822 175.561501
 177.07607  175.318948 175.863308 177.624396 175.516165 175.909477
 176.333099 176.996457 176.49809  175.341111 175.356347 175.562653
 176.176461 177.886039 175.338725 175.397887 176.382939 175.70752
 176.909144 176.267307 176.190491 175.629909 175.732065 175.629761
 175.677382 175.476395 176.171689 176.122018]
```

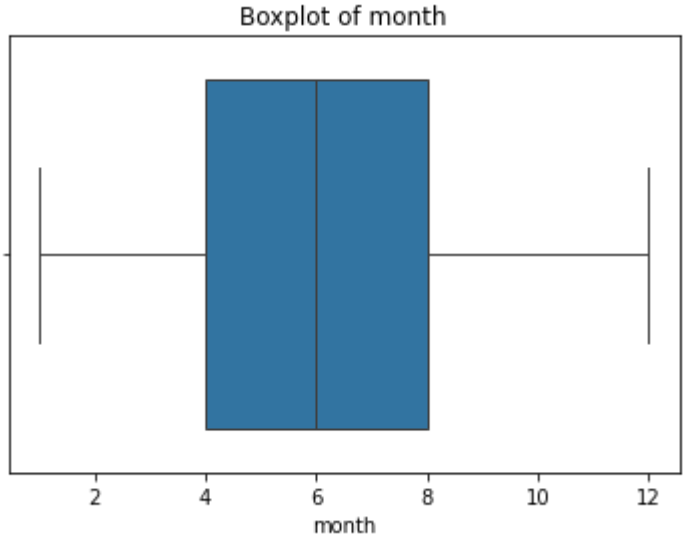



```
Column: coordinateUncertaintyInMeters
Number of outliers: 15569
Outliers:
[12018.   876. 27222. ...  2703.   782.  3758.]
```

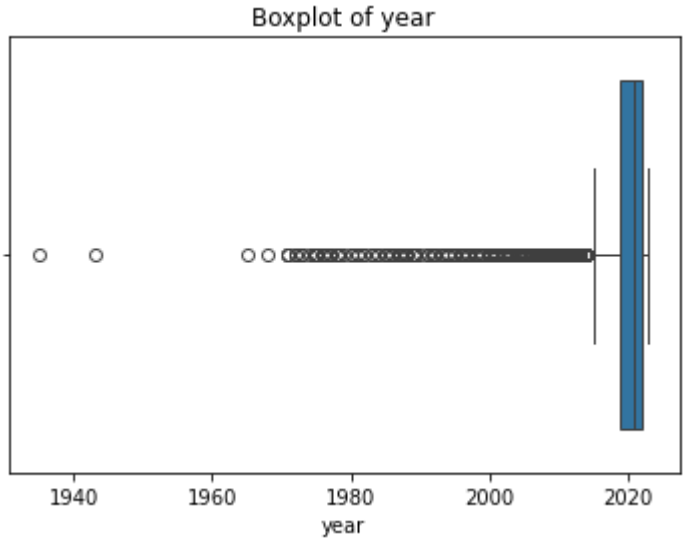
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js




```
Column: day
Number of outliers: 0
Outliers:
[]
```

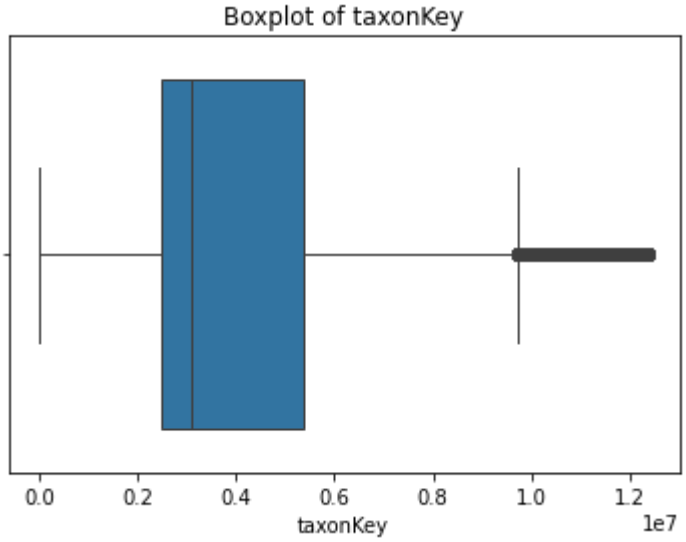



```
Column: month
Number of outliers: 0
Outliers:
[]
```

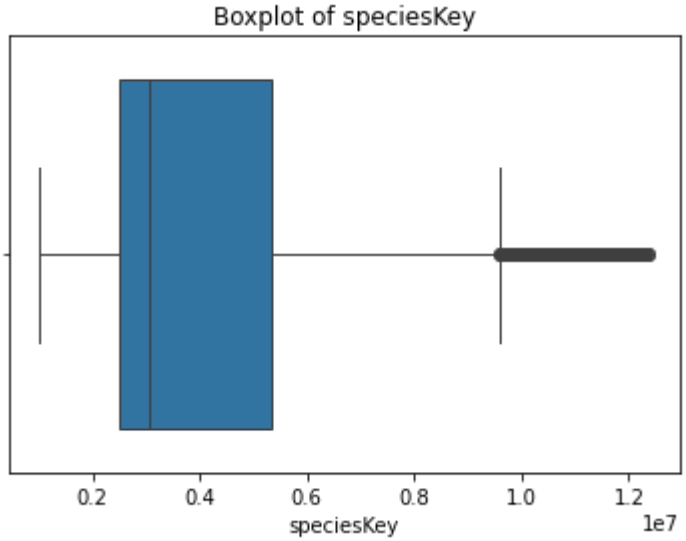



```
Column: year
Number of outliers: 3692
Outliers:
[2013 2011 2014 ... 2009 2008 2013]
```

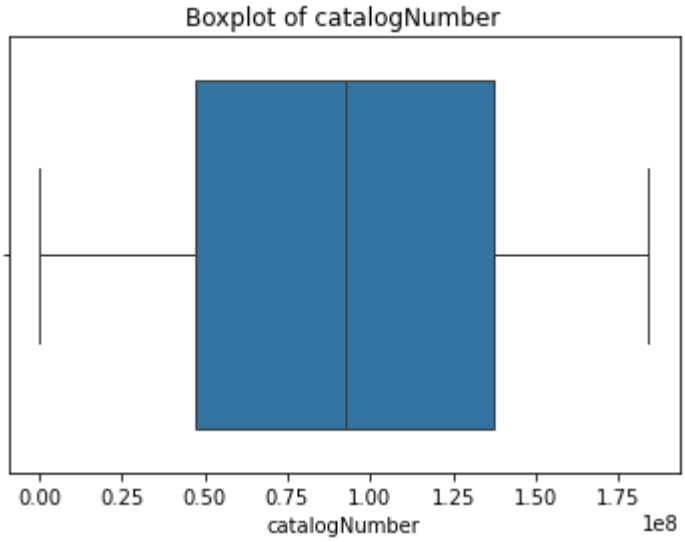
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js




```
Column: taxonKey
Number of outliers: 2719
Outliers:
[10671194  9780339 10902734 ... 10710678 10413588 10840565]
```




```
Column: speciesKey
Number of outliers: 3060
Outliers:
[10671194. 11534744.  9616058. ... 10710678. 10413588. 10840565.]
```




```
Column: catalogNumber
Number of outliers: 0
Outliers:
[]
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

In [30]:
```python
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Calculate the number of rows and columns for the subplots
n_cols = 3
n_rows = (len(num_col) + n_cols - 1) // n_cols

# Create a figure and axis object
fig, axs = plt.subplots(n_rows, n_cols, figsize=(15,10))
axs = axs.flatten()

# Iterate over each numerical column
outlier_counts = []
for i, col in enumerate(num_col):
    # Calculate Q1, Q3, and IQR
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1

    # Define the outlier range
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    # Find outliers
    outliers = df[(df[col] < lower_bound) | (df[col] > upper_bound)]

    # Display the boxplot
    sns.boxplot(x=df[col], ax=axs[i])
    axs[i].set_title(f'Boxplot of {col}')

    # Print information about outliers
    print(f"Column: {col}")
    print(f"Number of outliers: {outliers.shape[0]}")
    print(f"Outliers:\n{outliers[col].values}\n")

    # Store the outlier count
    outlier_counts.append(outliers.shape[0])

plt.tight_layout()
plt.show()

# Create a dataframe with the column names and outlier counts
outlier_df = pd.DataFrame({'Column': num_col, 'Number of Outliers': outlier_counts})

# Create a bar plot
plt.figure(figsize=(10,6))
sns.barplot(x='Column', y='Number of Outliers', data=outlier_df)
plt.title('Number of Outliers by Column')
plt.xlabel('Column')
plt.ylabel('Number of Outliers')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

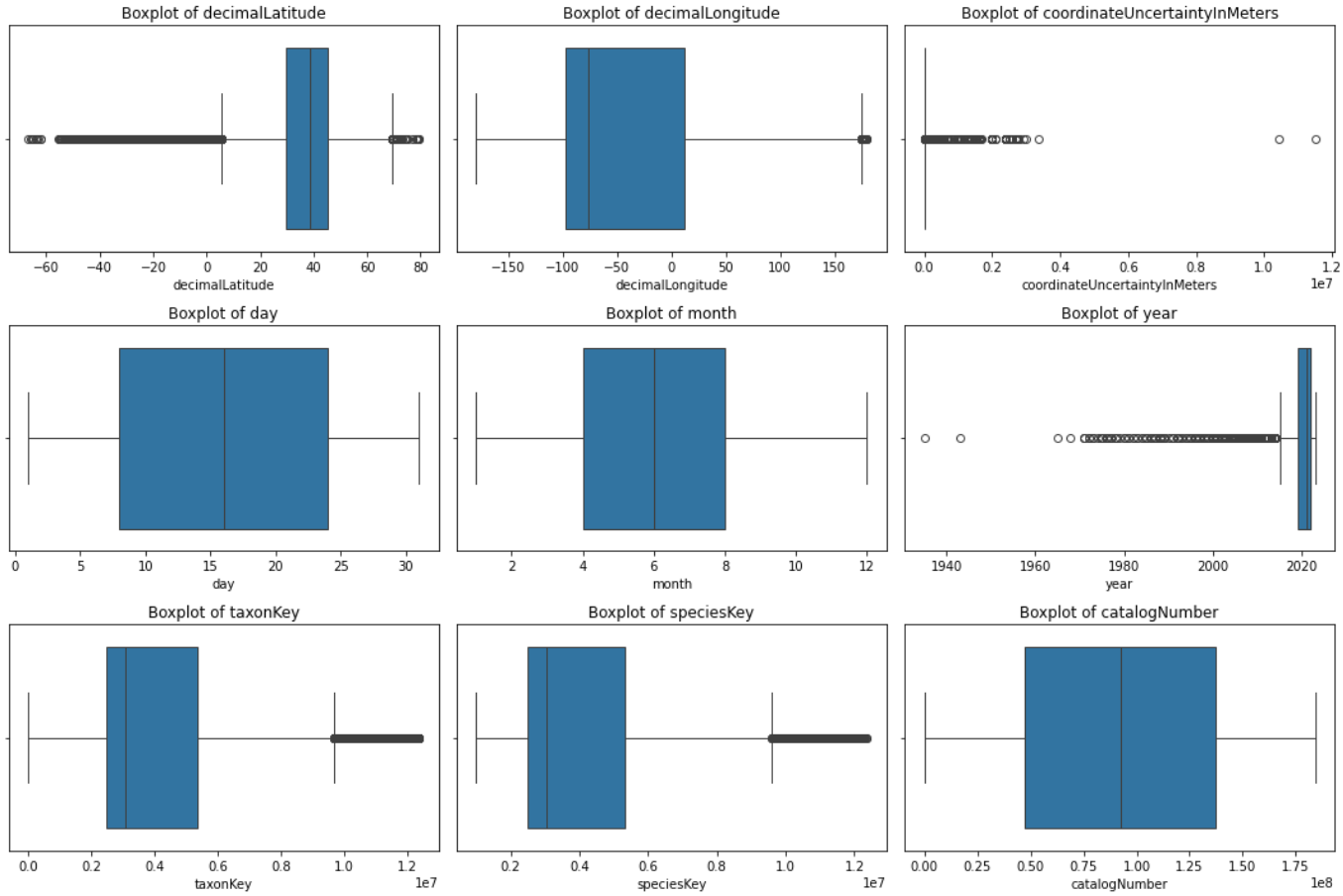
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

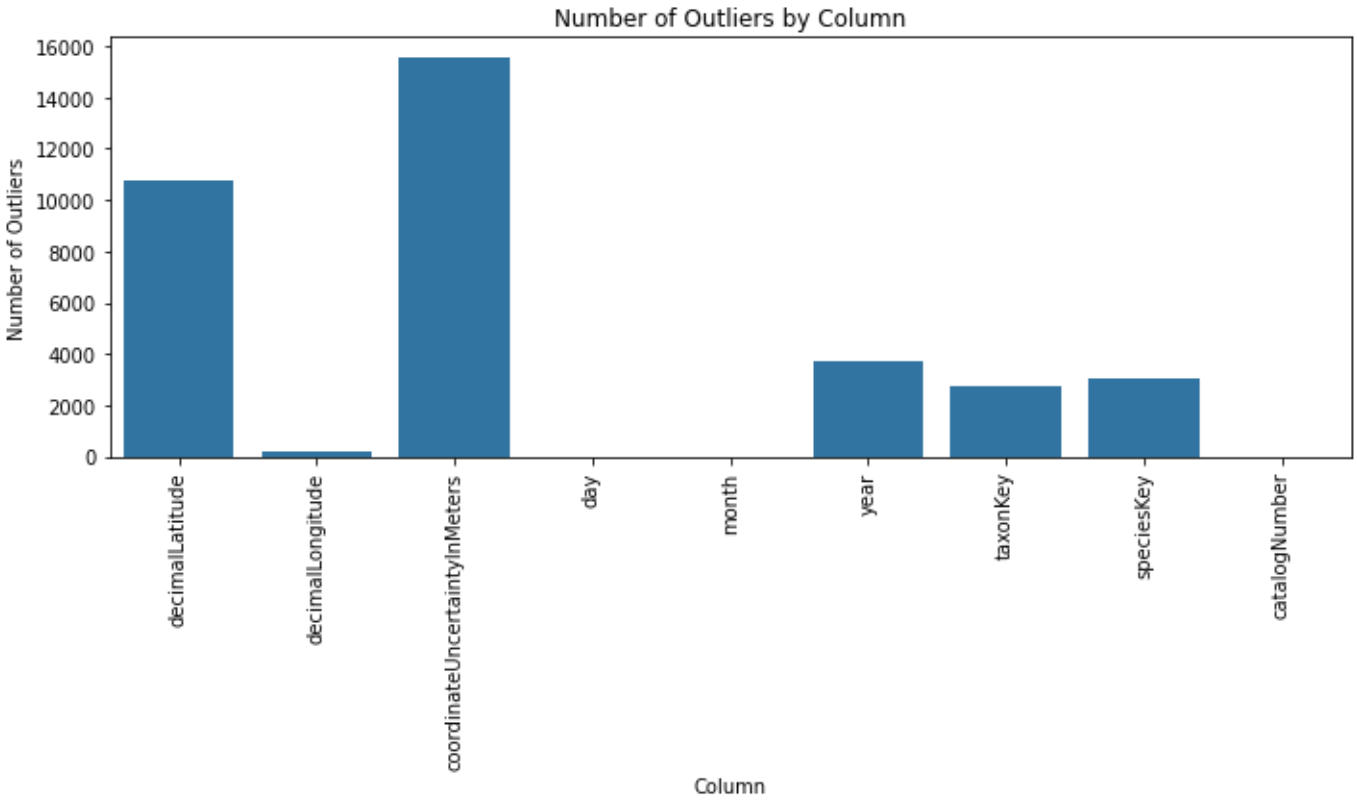
```
Column: decimalLatitude
Number of outliers: 10725
Outliers:
[-17.923595 -20.140961 -37.829741 ... -33.681979 -23.897306 -45.605386]

Column: decimalLongitude
Number of outliers: 196
Outliers:
[176.252445 175.922867 176.127987 178.146928 175.350005 176.418933
 176.004877 175.316742 176.054489 177.185415 175.500996 176.195984
 175.319842 176.500033 176.124272 176.896169 176.89109  175.578504
 175.747458 175.488911 177.229299 177.416667 176.073157 176.098029
 175.71353  179.088033 176.695519 175.627184 175.759705 177.3874
 175.65732  175.815606 175.683917 175.549592 175.745768 176.310563
 177.317543 175.665645 178.456847 175.544002 175.782328 176.017042
 176.078347 175.458577 177.176157 176.015048 175.665482 175.927303
 175.743624 176.216935 176.192347 175.838546 177.657546 177.885954
 175.556882 176.0864   175.618315 175.530409 176.006479 176.896244
 178.065032 175.423917 175.681817 176.277588 177.215719 175.566371
 175.50413  175.701235 175.747087 175.4328   178.090482 176.002307
 175.845488 175.697141 175.885348 176.768131 177.425369 177.320816
 176.303456 175.506728 175.595484 176.262199 175.487018 177.034145
 176.104674 177.315249 176.896184 175.953098 175.710799 175.680107
 177.178115 175.64131  176.2163   176.855835 175.325509 175.409441
 175.757699 177.18506  175.663943 176.896447 175.813111 175.693725
 177.194521 176.239752 176.149755 176.170309 175.715255 175.690551
 178.548313 175.826062 176.219083 175.677983 177.348827 175.953265
 177.215581 176.712631 176.072148 175.752045 176.756069 176.990385
 176.32808  177.081145 175.612291 175.875245 175.548862 179.998902
 175.554803 176.245536 175.812149 175.747777 176.58195  179.317077
 175.393581 176.284717 176.036962 175.691492 175.772898 176.218108
 175.684159 176.251066 175.809964 178.096726 177.415466 178.450343
 177.108869 178.034517 177.884943 177.793578 175.549948 175.686063
 176.089408 175.587416 176.072008 176.219114 175.617557 176.907949
 175.627306 175.921833 175.868797 176.928991 175.56851  175.850113
 176.977529 175.627352 175.358084 176.113684 175.950822 175.561501
 177.07607  175.318948 175.863308 177.624396 175.516165 175.909477
 176.333099 176.996457 176.49809  175.341111 175.356347 175.562653
 176.176461 177.886039 175.338725 175.397887 176.382939 175.70752
 176.909144 176.267307 176.190491 175.629909 175.732065 175.629761
 175.677382 175.476395 176.171689 176.122018]

Column: coordinateUncertaintyInMeters
Number of outliers: 15569
Outliers:
[12018.   876. 27222. ...  2703.   782.  3758.]

Column: day
Number of outliers: 0
Outliers:
[]

Column: month
Number of outliers: 0
Outliers:
[]

Column: year
Number of outliers: 3692
Outliers:
[2013 2011 2014 ... 2009 2008 2013]

Column: taxonKey
Number of outliers: 2719
Outliers:
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
10678 10413588 10840565]
```

```
Column: speciesKey
Number of outliers: 3060
Outliers:
[10671194. 11534744.  9616058. ... 10710678. 10413588. 10840565.]

Column: catalogNumber
Number of outliers: 0
Outliers:
[]
```





In [31]:
```
outlier_df
```

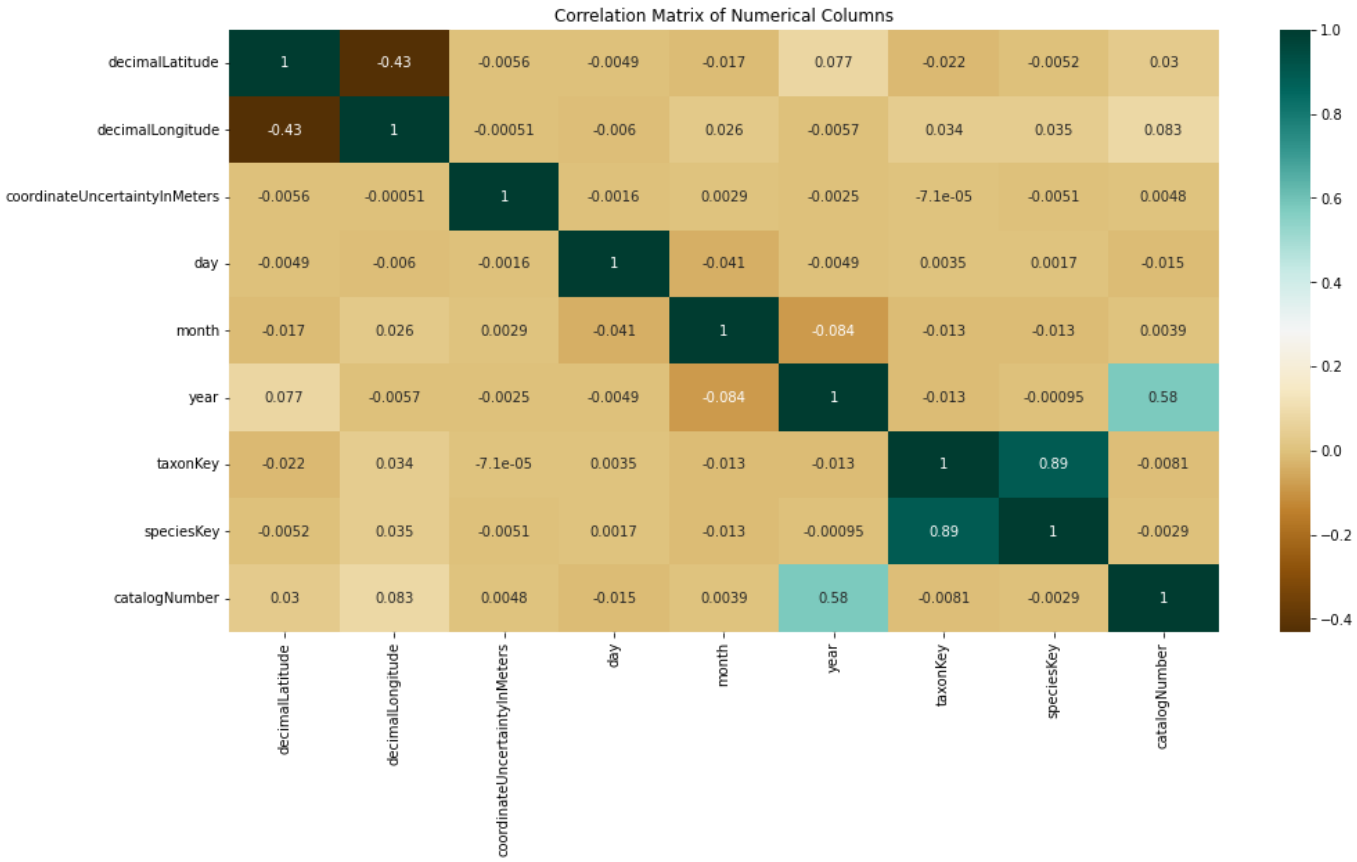
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[31]:

	Column	Number of Outliers
0	decimalLatitude	10725
1	decimalLongitude	196
2	coordinateUncertaintyInMeters	15569
3	day	0
4	month	0
5	year	3692
6	taxonKey	2719
7	speciesKey	3060
8	catalogNumber	0

In [32]:

```
# corrrleation plot of num columns
plt.figure(figsize=(16, 8))
correlation_matrix=df[num_col].corr()
sns.heatmap(correlation_matrix,annot=True,cmap='BrBG')
plt.title('Correlation Matrix of Numerical Columns')

plt.show()
```



In [33]:

```
cat_col
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[33]:
```
['kingdom',
 'phylum',
 'class',
 'order',
 'family',
 'genus',
 'species',
 'taxonRank',
 'scientificName',
 'verbatimScientificName',
 'countryCode',
 'stateProvince',
 'eventDate',
 'identifiedBy',
 'dateIdentified',
 'rightsHolder',
 'recordedBy',
 'lastInterpreted',
 'mediaType',
 'issue']
```

In [34]:
```python
# Frequency count
print("Frequency Counts for Categorical Columns:")
for col in cat_col:
    print(f"\n{col} Frequency Count:")
    print(df[col].value_counts())
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
Frequency Counts for Categorical Columns:

kingdom Frequency Count:
kingdom
Animalia     48414
Plantae      29914
Fungi         2109
Chromista       70
Protozoa        46
Bacteria        14
Viruses          3
Name: count, dtype: int64

phylum Frequency Count:
phylum
Tracheophyta           29579
Chordata               25311
Arthropoda             21481
Basidiomycota           1574
Mollusca                1180
Ascomycota               533
Bryophyta                222
Cnidaria                 191
Echinodermata            160
Marchantiophyta           74
Ochrophyta                67
Mycetozoa                 43
Porifera                  27
Annelida                  26
Rhodophyta                24
Chlorophyta               15
Platyhelminthes           15
Charophyta                11
Cyanobacteria              8
Bryozoa                    5
Proteobacteria             5
Ctenophora                 2
Onychophora                2
Nematoda                   2
Pisuviricota               2
Euglenozoa                 2
Myzozoa                    2
Rotifera                   1
Brachiopoda                1
Oomycota                   1
Amoebozoa                  1
Mucoromycota               1
Firmicutes                 1
Entomophthoromycota        1
Name: count, dtype: int64

class Frequency Count:
class
Magnoliopsida     24866
Insecta           19435
Aves              17774
Liliopsida         4120
Mammalia           2392
                  ...
Peronosporea          1
Petromyzonti          1
Charophyceae          1
Rhynchonellata        1
Stenolaemata          1
Name: count, Length: 96, dtype: int64
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
order Frequency Count:
order
Lepidoptera       12144
Passeriformes      8150
Asterales          3744
Lamiales           2678
Coleoptera         2554
                  ...
Percopsiformes        1
Hysteriales           1
Acorales              1
Dothideales           1
Diphysciales          1
Name: count, Length: 446, dtype: int64

family Frequency Count:
family
Asteraceae       3494
Nymphalidae      2190
Anatidae         1823
Fabaceae         1797
Rosaceae         1177
                 ...
Pupillidae          1
Miturgidae          1
Rousseaceae         1
Mimetidae           1
Turritellidae       1
Name: count, Length: 2044, dtype: int64

genus Frequency Count:
genus
Anas          569
Ardea         423
Turdus        408
Bombus        403
Buteo         366
             ...
Scutus          1
Leucospora      1
Boltonia        1
Illinigina      1
Ciulfina        1
Name: count, Length: 10883, dtype: int64

species Frequency Count:
species
Anas platyrhynchos         1399
Apis mellifera              271
Danaus plexippus            219
Harmonia axyridis           218
Passer domesticus           198
                           ...
Rhenea michii                 1
Ranunculus monophyllus        1
Achillea clavennae            1
Argyresthia goedartella       1
Ciulfina biseriata            1
Name: count, Length: 23358, dtype: int64

taxonRank Frequency Count:
taxonRank
SPECIES       76879
SUBSPECIES     2111
GENUS           962
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
FAMILY           77
FORM             66
Name: count, dtype: int64

scientificName Frequency Count:
scientificName
Anas platyrhynchos Linnaeus, 1758                        334
Apis mellifera Linnaeus, 1758                            265
Harmonia axyridis (Pallas, 1773)                         218
Danaus plexippus (Linnaeus, 1758)                        215
Ardea herodias Linnaeus, 1758                            196
                                                        ...
Megarhyssa greenei Viereck, 1911                           1
Polypremum procumbens L.                                   1
Clepsis melaleucanus (Walker, 1863)                        1
Herpetogramma aquilonalis Handfield & Handfield, 2021      1
Ciulfina biseriata Westwood, 1889                          1
Name: count, Length: 24856, dtype: int64

verbatimScientificName Frequency Count:
verbatimScientificName
Anas platyrhynchos      334
Apis mellifera          265
Harmonia axyridis       218
Danaus plexippus        215
Ardea herodias          196
                       ...
Timandra amaturaria       1
Lactuca biennis           1
Planuncus tingitanus      1
Papaver walpolei          1
Ciulfina biseriata        1
Name: count, Length: 25011, dtype: int64

countryCode Frequency Count:
countryCode
US    37580
CA     6890
RU     4222
AU     3396
MX     2423
      ...
TO        1
CC        1
QA        1
ST        1
BN        1
Name: count, Length: 215, dtype: int64

stateProvince Frequency Count:
stateProvince
California              7210
Texas                   5007
Ontario                 3020
Florida                 2192
England                 2034
                        ...
Saint Thomas Lowland       1
Boa Vista                  1
Daman and Diu              1
Iași                       1
Gyeongsangbuk-do           1
Name: count, Length: 1741, dtype: int64

eventDate Frequency Count:
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
2019-04-27T00:00:00    9
2020-06-14T00:00:00    7
2020-07-15T00:00:00    7
2020-04-26T00:00:00    7
2017-06-01T00:00:00    6
                      ..
2023-01-19T16:09:00    1
2012-03-28T21:53:00    1
2020-05-05T08:27:00    1
2022-07-27T15:15:05    1
2019-12-31T23:32:00    1
Name: count, Length: 78927, dtype: int64

identifiedBy Frequency Count:
identifiedBy
CK Kelly                    588
Sam Kieschnick              176
John D Reynolds             164
Tony Rebelo                 160
Jon McIntyre                157
                           ...
wewantashrubbery              1
myco_log                      1
Renato Machado de Sobral      1
jaga1242                      1
Alain Tio                     1
Name: count, Length: 29921, dtype: int64

dateIdentified Frequency Count:
dateIdentified
2017-01-01T20:43:23    359
2022-03-27T06:08:57      2
2020-10-26T02:21:13      2
2022-11-08T09:36:29      2
2021-09-01T22:17:56      2
                       ...
2022-06-16T16:56:32      1
2022-06-14T23:48:48      1
2022-06-13T12:11:23      1
2022-06-11T23:30:31      1
2020-02-09T02:34:20      1
Name: count, Length: 80155, dtype: int64

rightsHolder Frequency Count:
rightsHolder
CK Kelly           256
Tony Rebelo        192
John D Reynolds    171
DW                 158
Jon McIntyre       154
                  ...
shaynolds            1
shazza17             1
mathewdotson         1
lcpaige              1
Alain Tio            1
Name: count, Length: 35396, dtype: int64

recordedBy Frequency Count:
recordedBy
CK Kelly           256
Tony Rebelo        192
John D Reynolds    171
DW                 158
Jon McIntyre       154
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
shaynolds            1
shazza17             1
mathewdotson         1
lcpaige              1
Alain Tio            1
Name: count, Length: 35396, dtype: int64

lastInterpreted Frequency Count:
lastInterpreted
2023-09-21T16:30:03.415Z    3
2023-09-21T13:29:41.013Z    2
2023-09-21T16:34:57.001Z    2
2023-09-21T16:14:57.189Z    2
2023-09-21T15:55:47.578Z    2
                           ..
2023-09-21T16:01:44.796Z    1
2023-09-21T16:01:36.631Z    1
2023-09-21T16:01:29.283Z    1
2023-09-21T16:01:13.316Z    1
2023-09-21T13:47:12.604Z    1
Name: count, Length: 80196, dtype: int64

mediaType Frequency Count:
mediaType
StillImage
53348
StillImage;StillImage
14016
StillImage;StillImage;StillImage
6897
StillImage;StillImage;StillImage;StillImage
3458
StillImage;StillImage;StillImage;StillImage;StillImage
1053
StillImage;StillImage;StillImage;StillImage;StillImage;StillImage
517
Sound
380
StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage
284
StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImag
e
184
StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImag
e;StillImage
114
StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImag
e;StillImage;StillImage
66
StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImag
e;StillImage;StillImage;StillImage
50
StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImag
e;StillImage;StillImage;StillImage;StillImage
39
StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImag
e;StillImage;StillImage;StillImage;StillImage;StillImage
21
StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImag
e;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillIm
age;StillImage;StillImage;StillImage;StillImage
16
Sound;Sound
15
StillImage;Sound
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Sound;StillImage
13
StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImag
e;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage
11
StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImag
e;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillIm
age
10
StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImag
e;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage
9
StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImag
e;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillIm
age;StillImage;StillImage
6
StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImag
e;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillIm
age;StillImage;StillImage;StillImage
5
StillImage;Sound;StillImage
4
Sound;StillImage;StillImage
4
StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImag
e;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillIm
age;StillImage
4
StillImage;StillImage;StillImage;Sound
3
StillImage;StillImage;Sound
3
StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImag
e;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillIm
age;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;Still
Image;StillImage
2
StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImag
e;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillIm
age;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;Still
Image;StillImage;StillImage;StillImage
2
StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImag
e;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillIm
age;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage
2
Sound;Sound;Sound;Sound;Sound;Sound
2
StillImage;StillImage;Sound;StillImage
2
Sound;Sound;Sound;Sound
2
StillImage;StillImage;Sound;StillImage;StillImage
2
StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImag
e;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillIm
age;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;Still
Image;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage
1
StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImag
e;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillIm
age;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;Still
Image;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;Sti
llImage;StillImage;StillImage 1
Sound;Sound;StillImage;Sound;StillImage

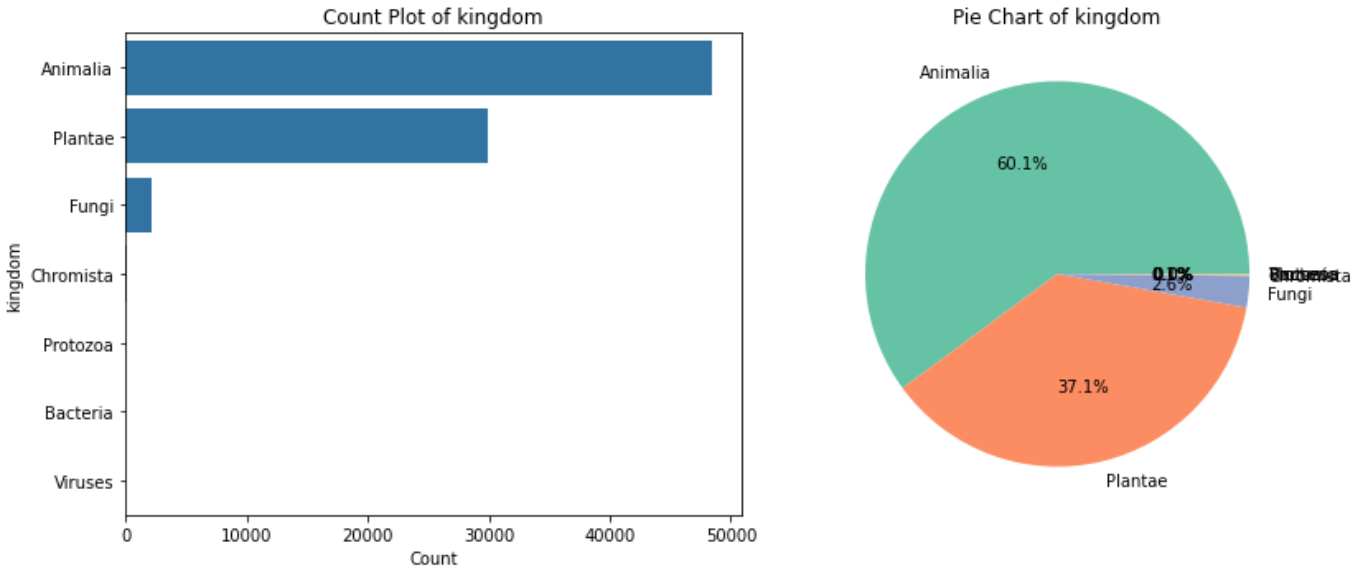
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

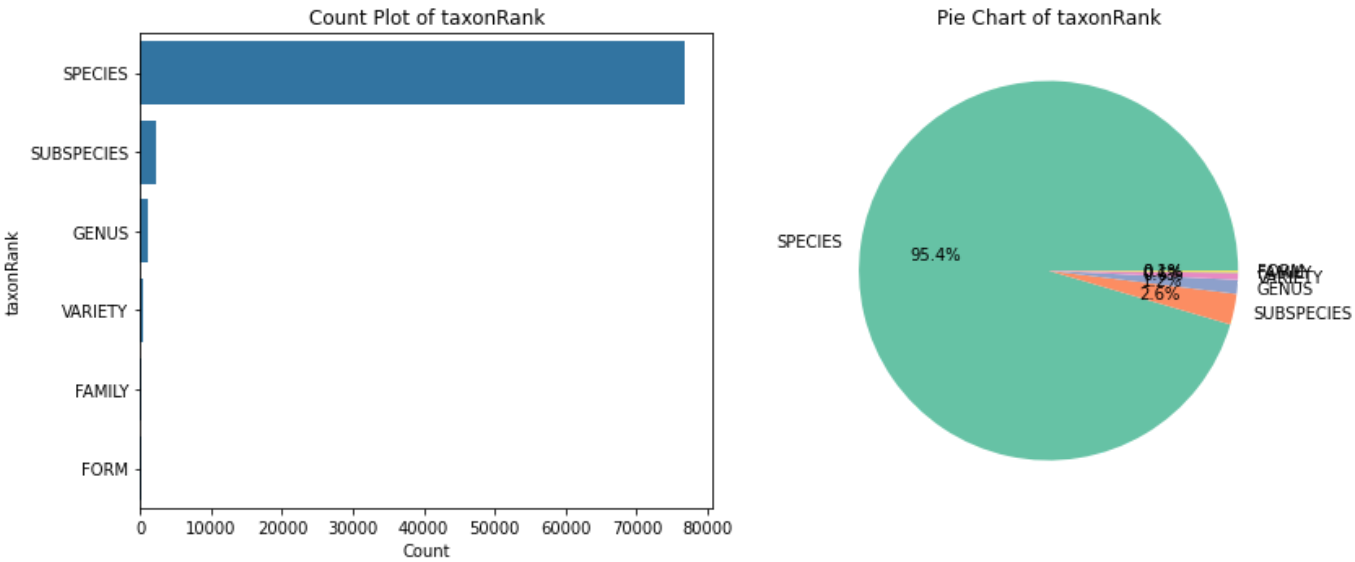
```
Sound;Sound;Sound
1
Sound;StillImage;StillImage;StillImage
1
StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImag
e;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillIm
age;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage
1
StillImage;Sound;StillImage;Sound;StillImage;StillImage;StillImage;StillImage;Sound
1
Sound;StillImage;StillImage;StillImage;StillImage;StillImage
1
StillImage;StillImage;StillImage;Sound;StillImage
1
StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImag
e;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillIm
age;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;StillImage;Still
Image
1
Sound;StillImage;StillImage;Sound
1
StillImage;StillImage;StillImage;StillImage;Sound;StillImage
1
Name: count, dtype: int64

issue Frequency Count:
issue
COORDINATE_ROUNDED;CONTINENT_DERIVED_FROM_COORDINATES;TAXON_MATCH_TAXON_ID_IGNORED
73783
CONTINENT_DERIVED_FROM_COORDINATES;TAXON_MATCH_TAXON_ID_IGNORED
5182
COORDINATE_ROUNDED;CONTINENT_DERIVED_FROM_COORDINATES;TAXON_MATCH_TAXON_ID_IGNORED;TAX
ON_MATCH_HIGHERRANK                655
COORDINATE_ROUNDED;CONTINENT_DERIVED_FROM_COORDINATES;TAXON_MATCH_TAXON_ID_IGNORED;TAX
ON_MATCH_FUZZY                     218
TAXON_MATCH_TAXON_ID_IGNORED
217
COORDINATE_ROUNDED;TAXON_MATCH_TAXON_ID_IGNORED
202
COORDINATE_ROUNDED;COUNTRY_DERIVED_FROM_COORDINATES;TAXON_MATCH_TAXON_ID_IGNORED
110
CONTINENT_DERIVED_FROM_COORDINATES;TAXON_MATCH_TAXON_ID_IGNORED;TAXON_MATCH_HIGHERRANK
63
COORDINATE_ROUNDED;COUNTRY_COORDINATE_MISMATCH;CONTINENT_DERIVED_FROM_COORDINATES;TAXO
N_MATCH_TAXON_ID_IGNORED            62
COORDINATE_ROUNDED;COUNTRY_DERIVED_FROM_COORDINATES;CONTINENT_DERIVED_FROM_COORDINATE
S;TAXON_MATCH_TAXON_ID_IGNORED       21
CONTINENT_DERIVED_FROM_COORDINATES;TAXON_MATCH_TAXON_ID_IGNORED;TAXON_MATCH_FUZZY
20
COUNTRY_DERIVED_FROM_COORDINATES;TAXON_MATCH_TAXON_ID_IGNORED
15
COORDINATE_ROUNDED;CONTINENT_DERIVED_FROM_COORDINATES;TAXON_MATCH_TAXON_ID_IGNORED;MUL
TIMEDIA_DATE_INVALID                 8
COORDINATE_ROUNDED;TAXON_MATCH_TAXON_ID_IGNORED;TAXON_MATCH_HIGHERRANK
4
COORDINATE_ROUNDED;CONTINENT_DERIVED_FROM_COORDINATES;TAXON_MATCH_TAXON_ID_IGNORED;MUL
TIMEDIA_URI_INVALID                  3
COUNTRY_COORDINATE_MISMATCH;CONTINENT_DERIVED_FROM_COORDINATES;TAXON_MATCH_TAXON_ID_IG
NORED                                3
TAXON_MATCH_TAXON_ID_IGNORED;TAXON_MATCH_FUZZY
2
COORDINATE_ROUNDED;COUNTRY_COORDINATE_MISMATCH;TAXON_MATCH_TAXON_ID_IGNORED
1
CONTINENT_DERIVED_FROM_COORDINATES;TAXON_MATCH_TAXON_ID_IGNORED;MULTIMEDIA_DATE_INVALI
D                                    1
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

In [35]:

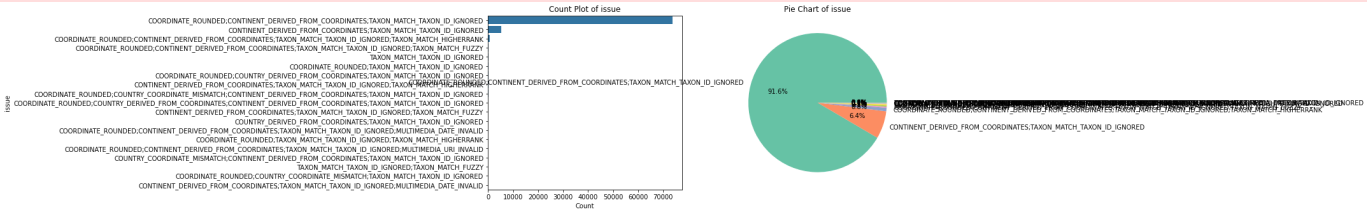
```
# Filter categorical columns with 10 or fewer unique values
filtered_cat_col = [col for col in cat_col if df[col].nunique() <=30]

# Loop through the filtered columns and plot count plot and pie chart
for col in filtered_cat_col:
    # Frequency count for the column
    value_counts = df[col].value_counts()

    # Plot count plot
    plt.figure(figsize=(12, 5))

    # Count plot
    plt.subplot(1, 2, 1)
    sns.countplot(y=df[col], order=value_counts.index)
    plt.title(f'Count Plot of {col}')
    plt.xlabel('Count')
    plt.ylabel(col)

    # Pie chart
    plt.subplot(1, 2, 2)
    value_counts.plot.pie(autopct='%1.1f%%', startangle=0, colors=sns.color_palette("
    plt.title(f'Pie Chart of {col}')
    plt.ylabel('')  # Hide the y-label in the pie chart

    # Adjust layout
    plt.tight_layout()
    plt.show()
```





Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

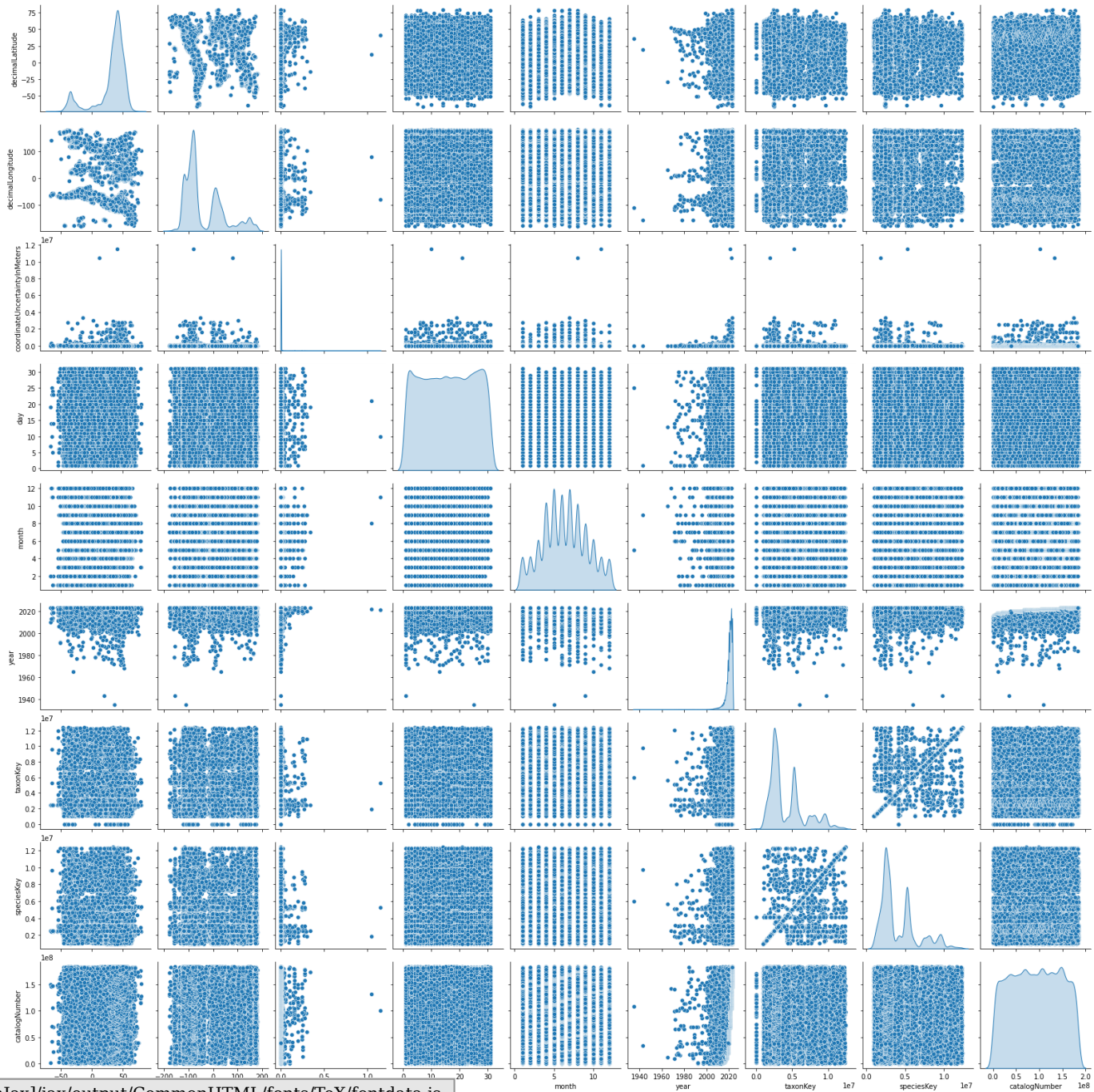
```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\1031336515.py:26: UserWarning: Tight
layout not applied. The left and right margins cannot be made large enough to accommod
ate all Axes decorations.
  plt.tight_layout()
```



Bivariate analysis

numerical-numerical

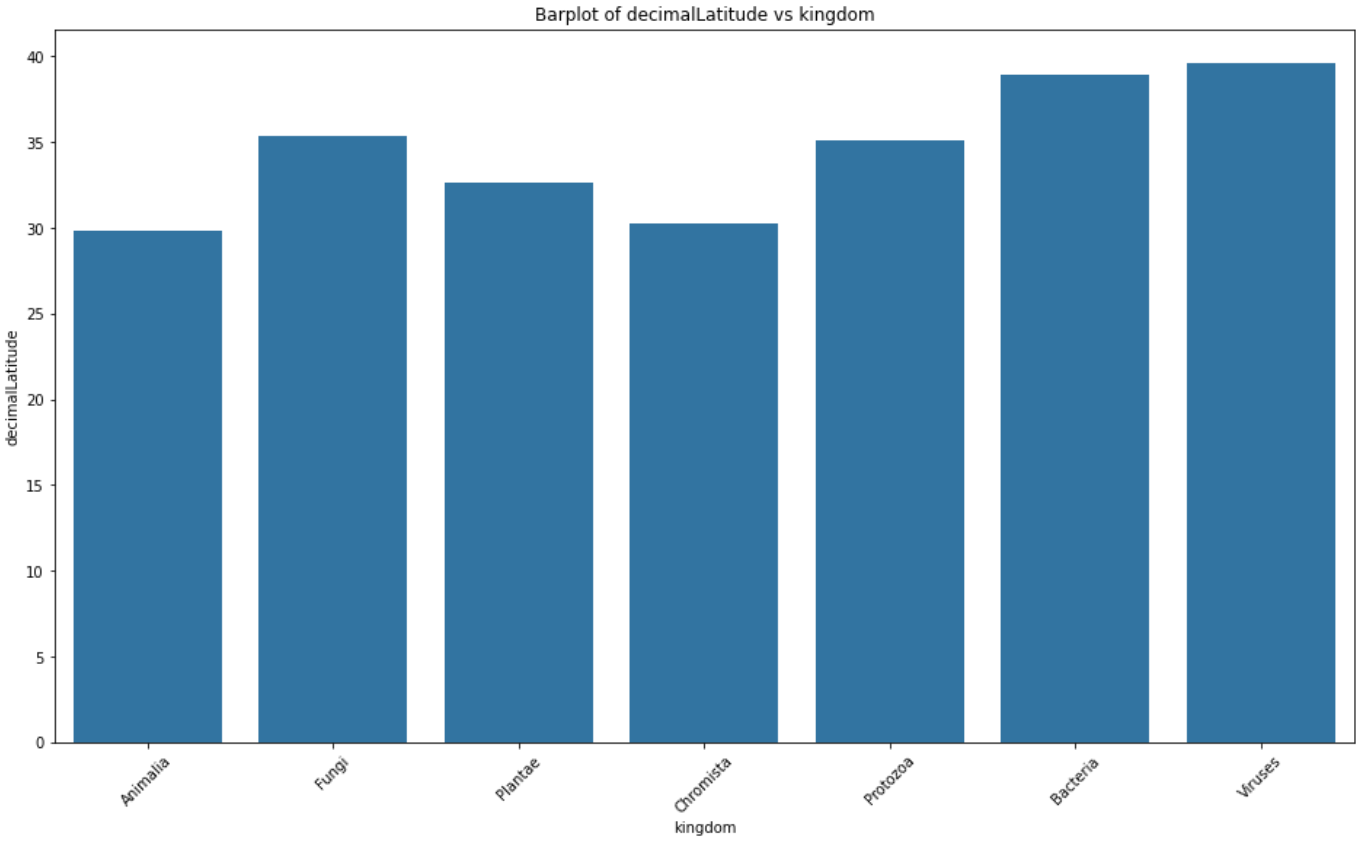
In [36]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming num_col is a list of numerical column names in your dataframe df
sns.pairplot(df[num_col], diag_kind="kde")
plt.show()
```



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

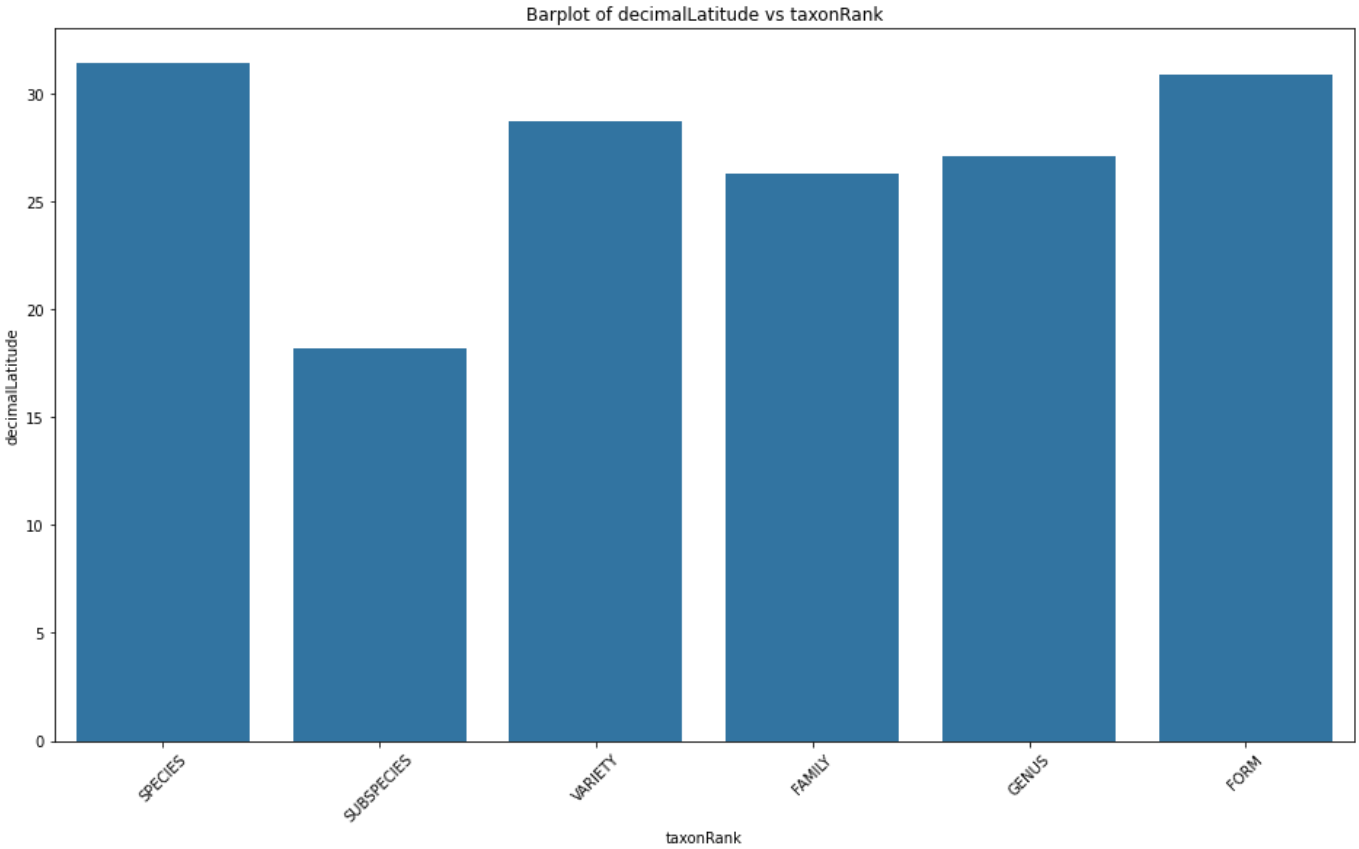
In [37]:
```
# Loop through each numerical and categorical column
for i in num_col:
    for j in filtered_cat_col:
        plt.figure(figsize=(16, 9))

        # Barplot - Use categorical variable 'j' on x-axis, and numerical 'i' on y-ax
        sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence inte

        plt.title(f'Barplot of {i} vs {j}')
        plt.xlabel(j)
        plt.ylabel(i)
        plt.xticks(rotation=45)  # Rotate x-axis labels for better readability if nec
        plt.show()
```

```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```



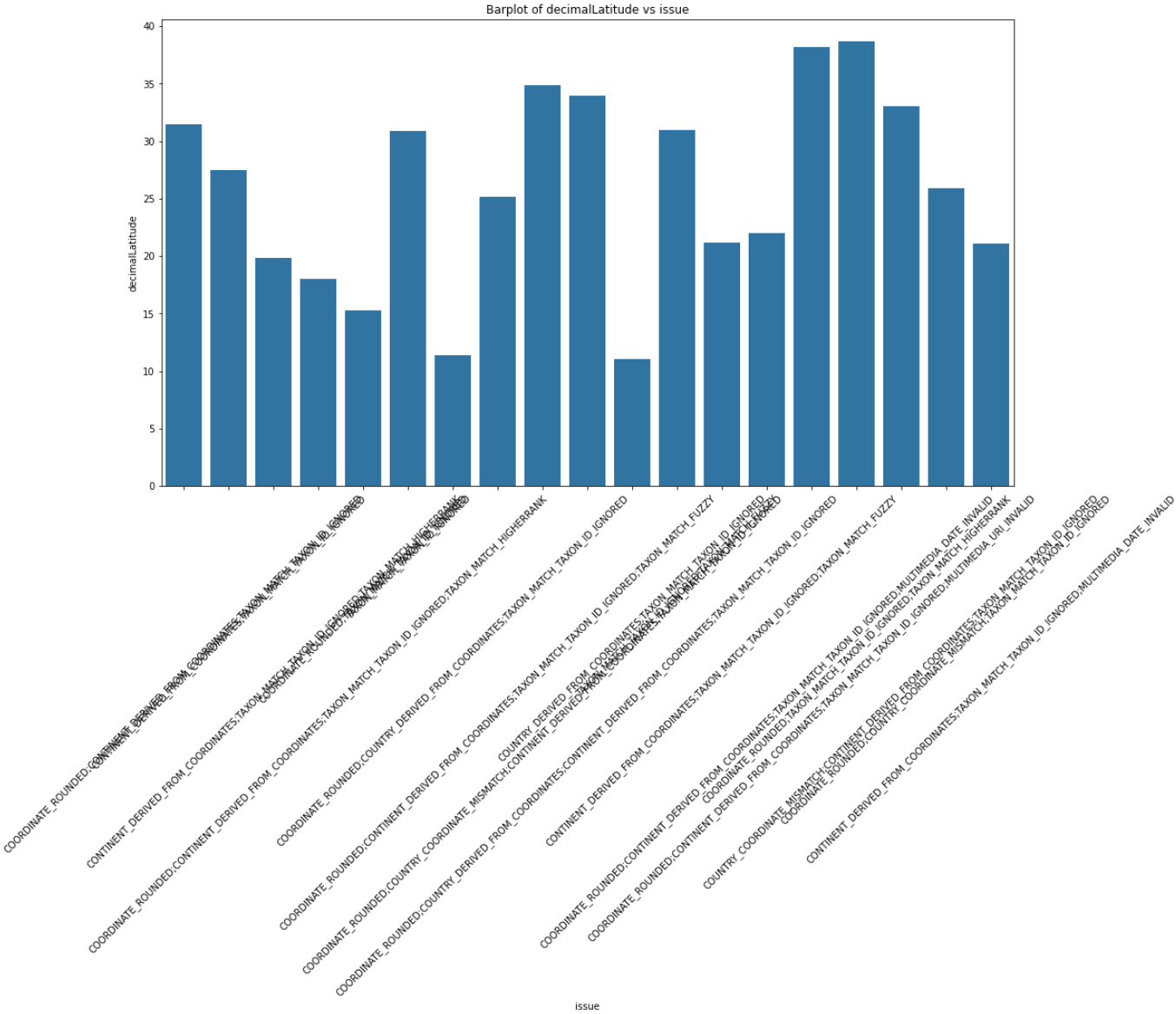

```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

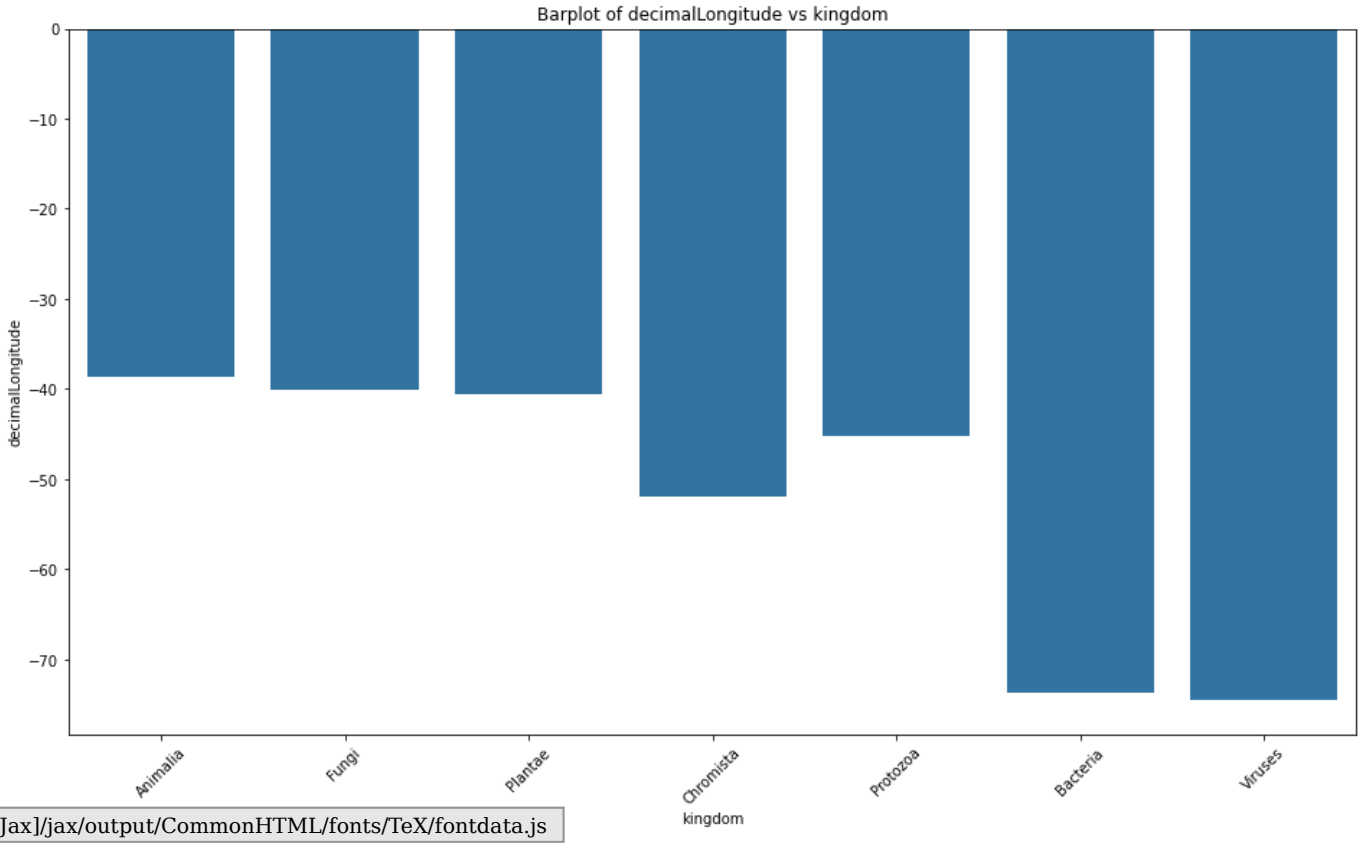



```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

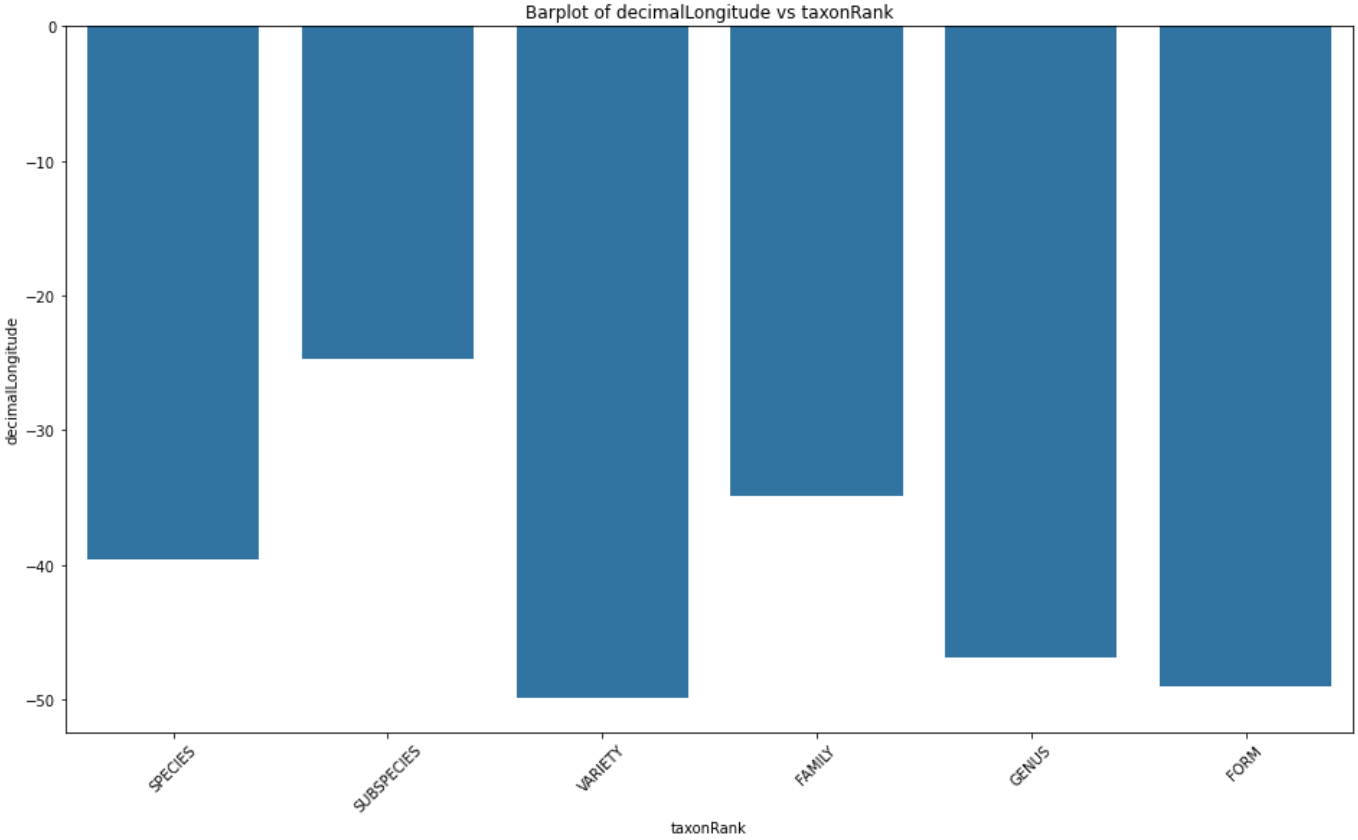



```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```



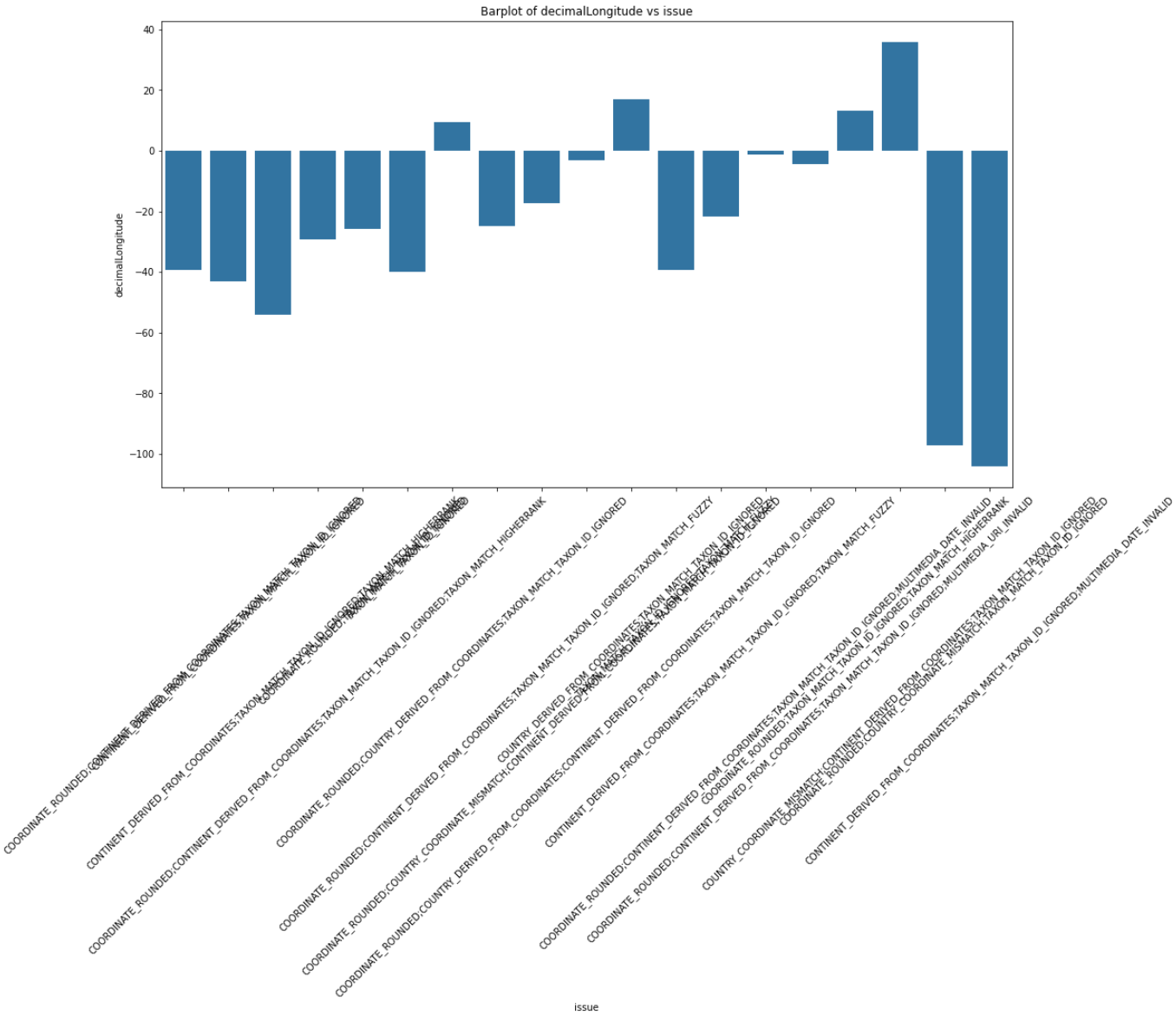
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```

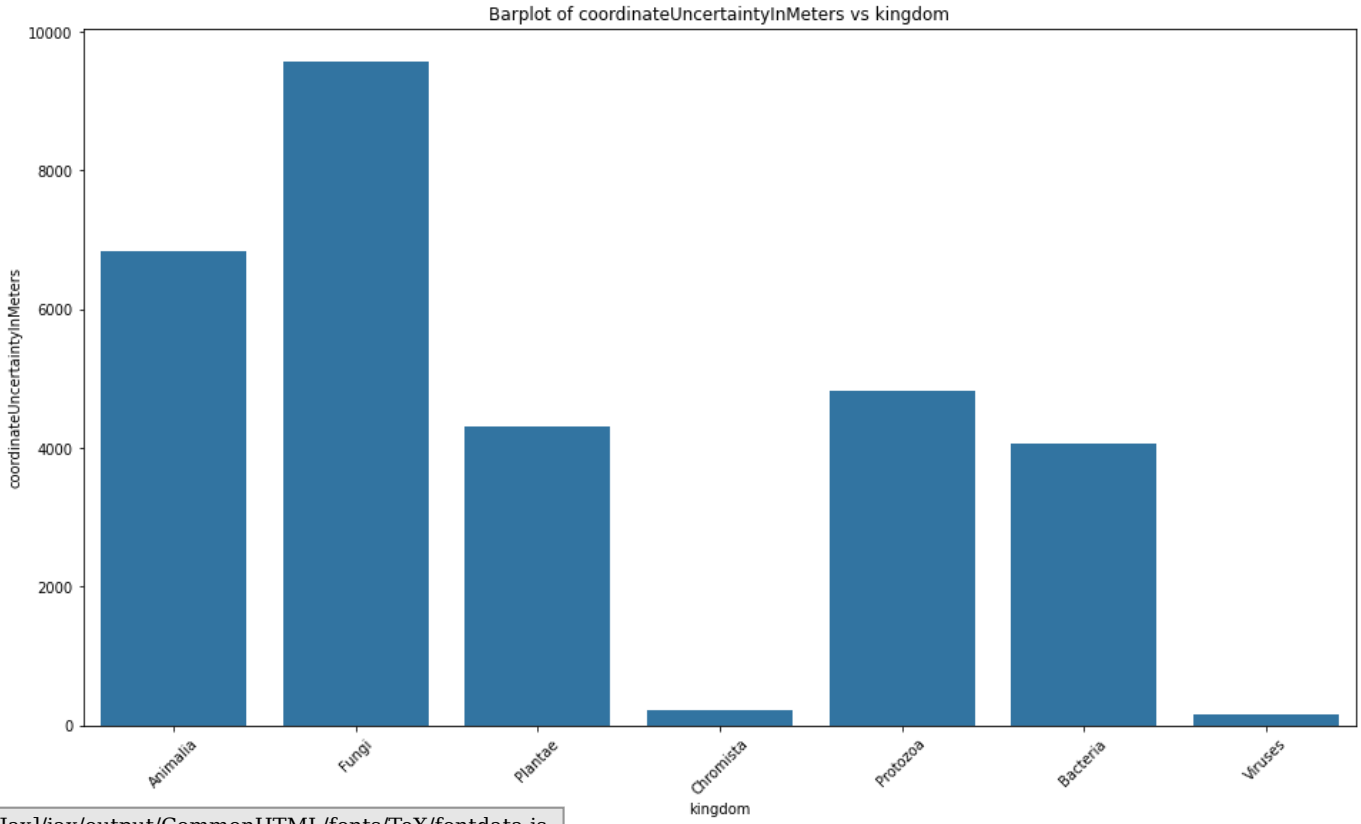



```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

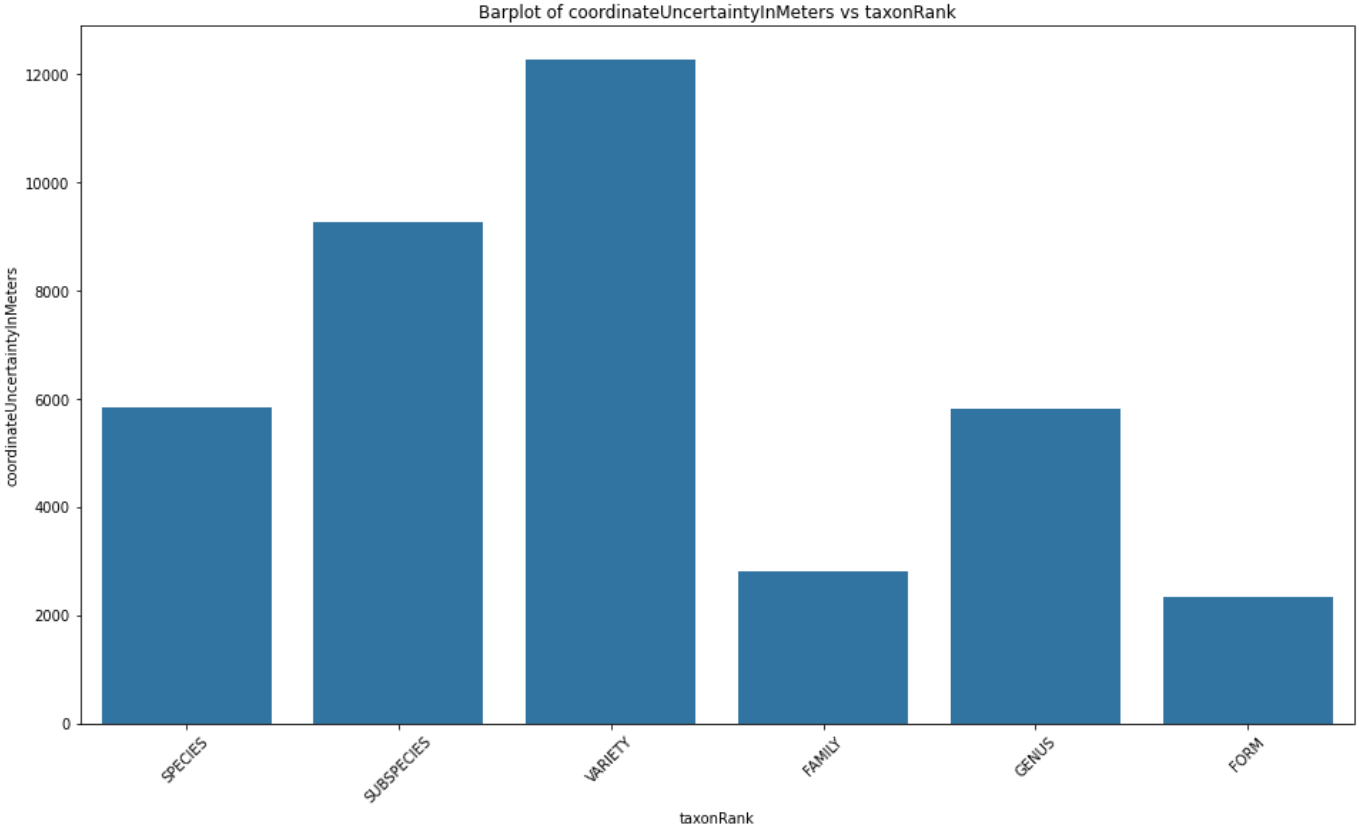



```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```



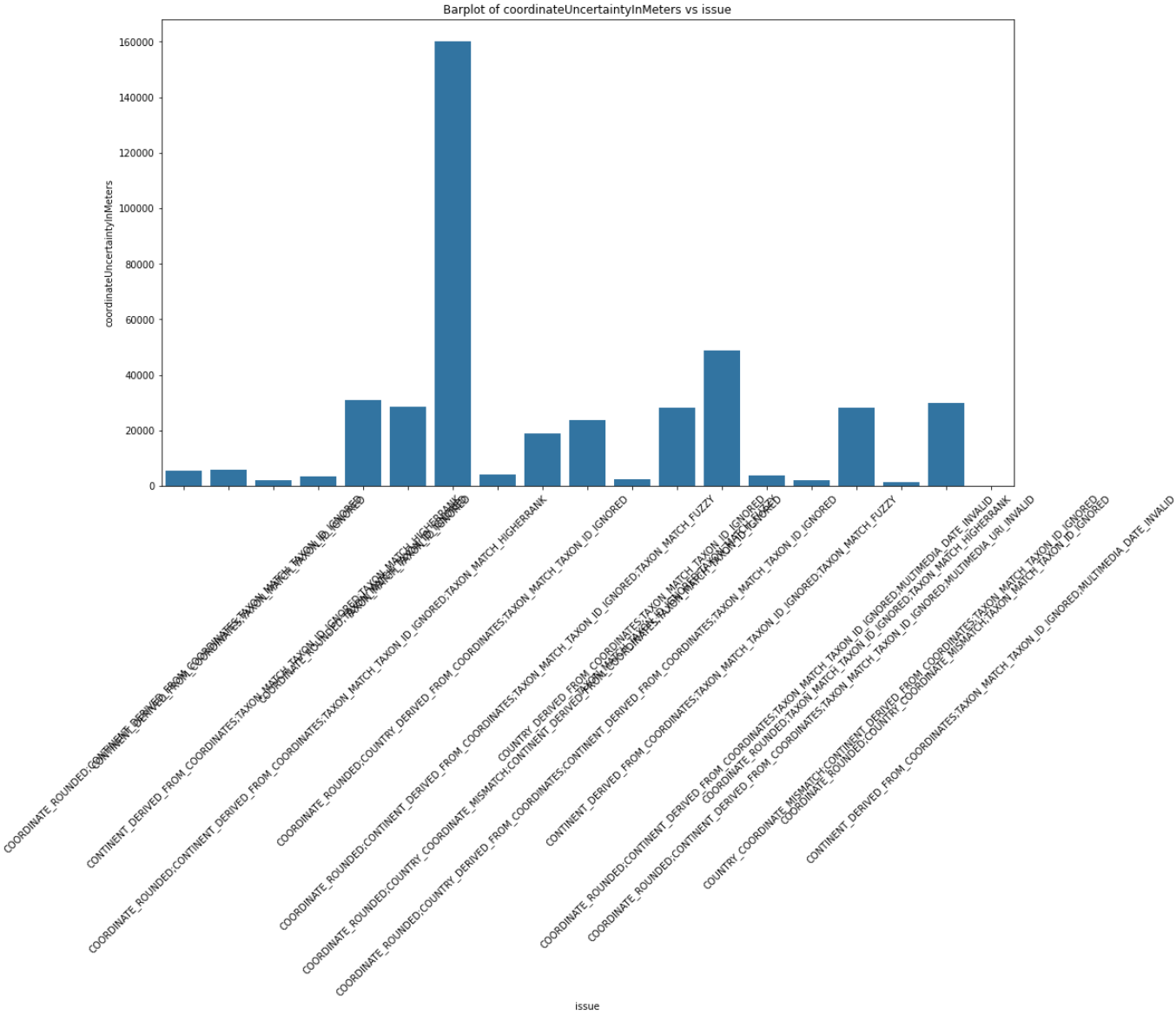
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```



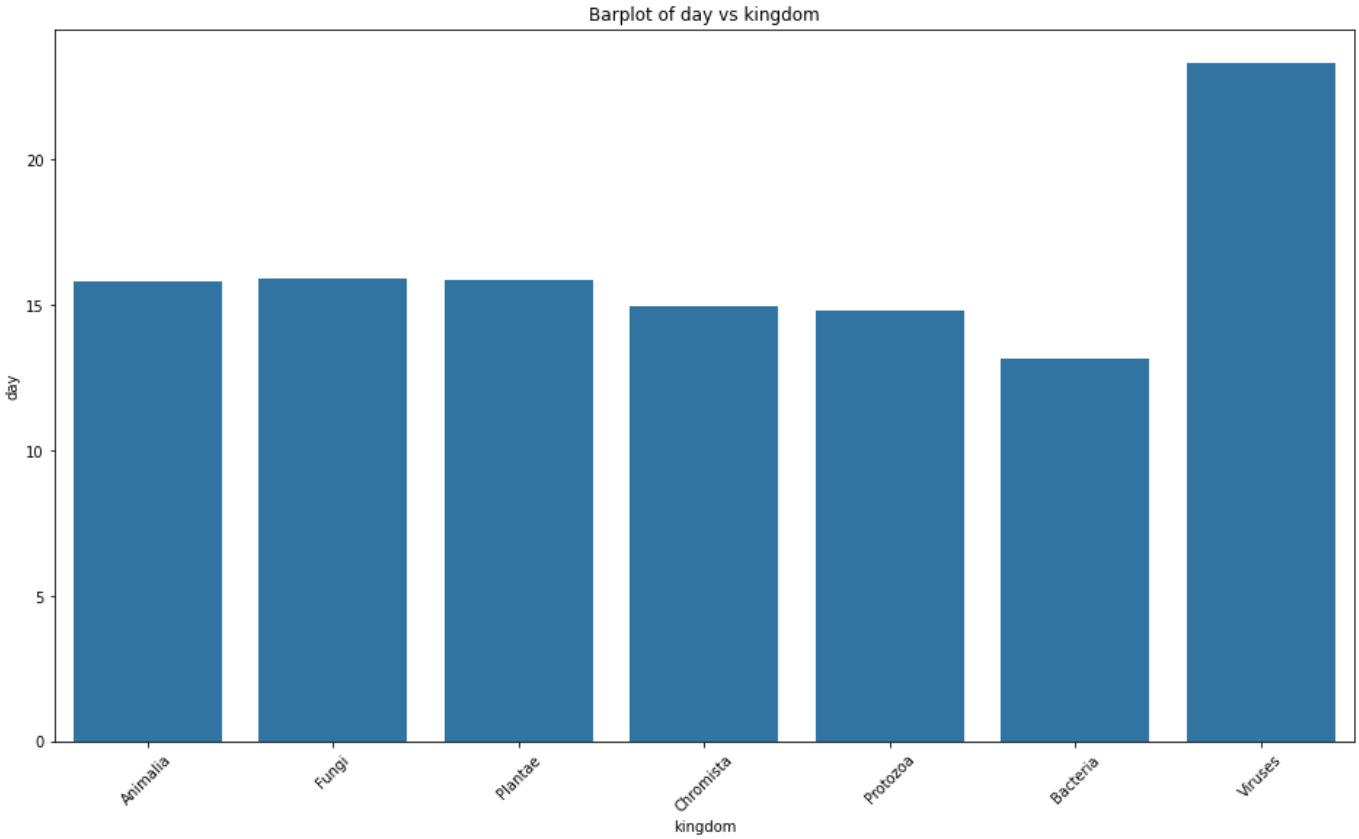

```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

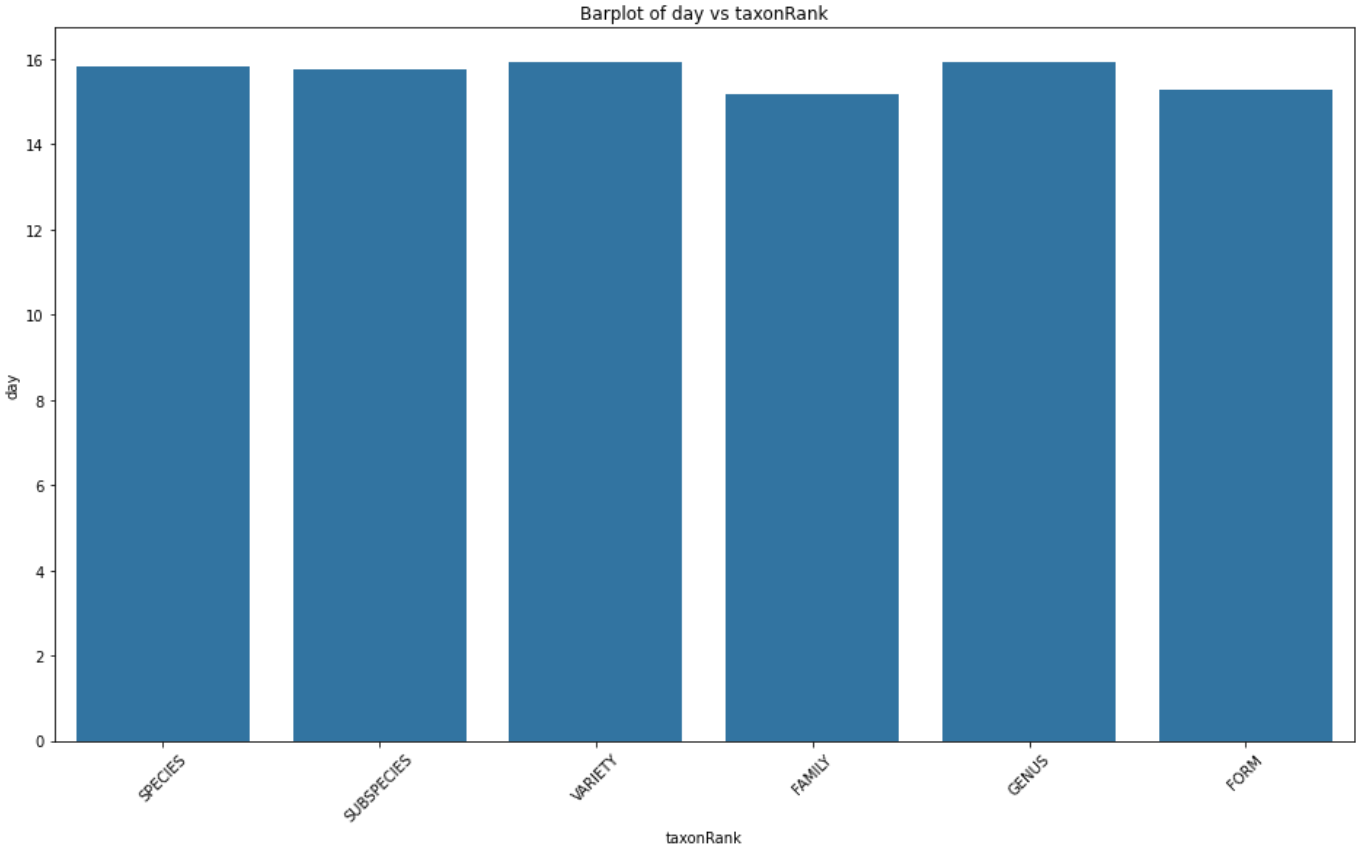



```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

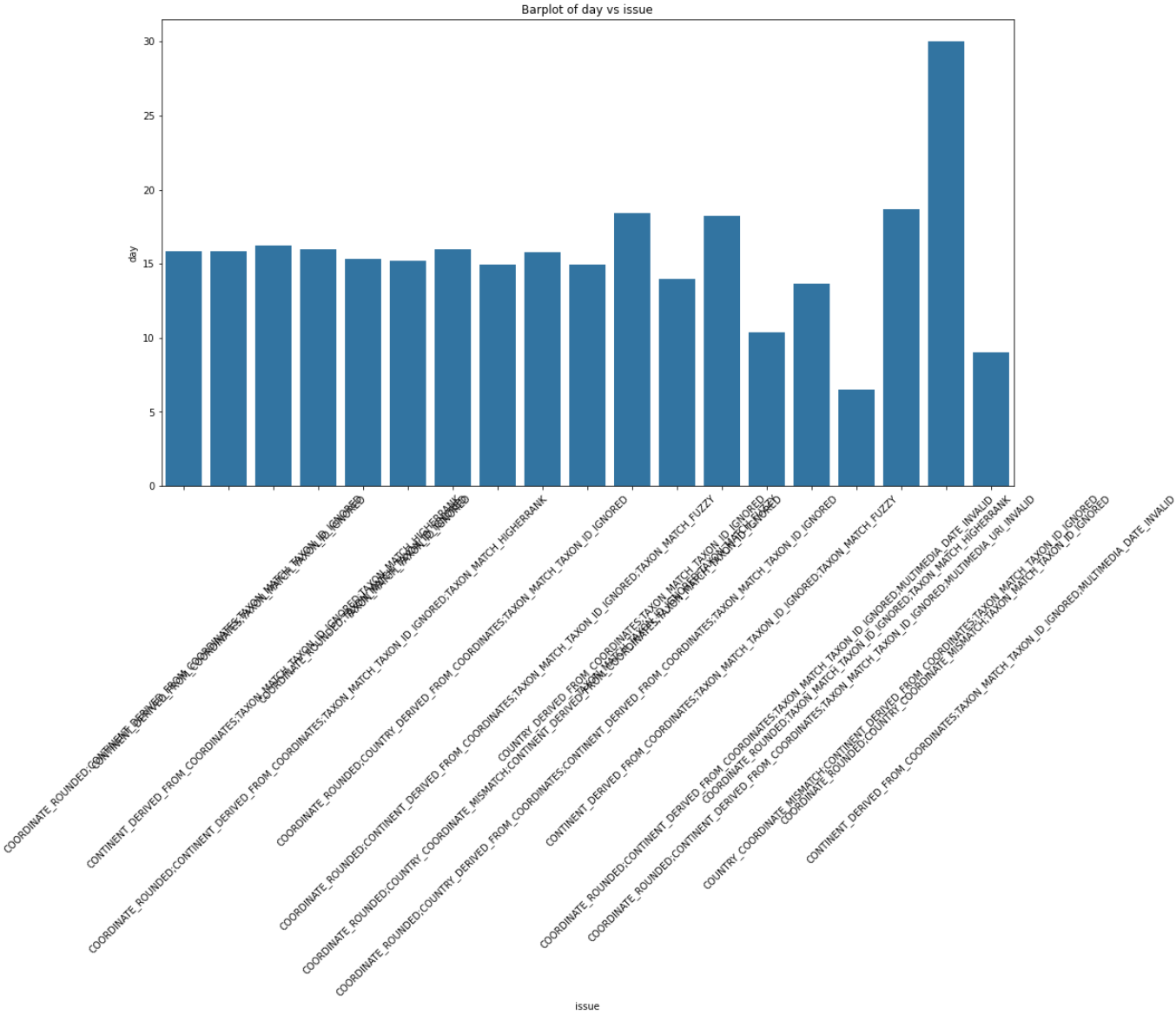



```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```



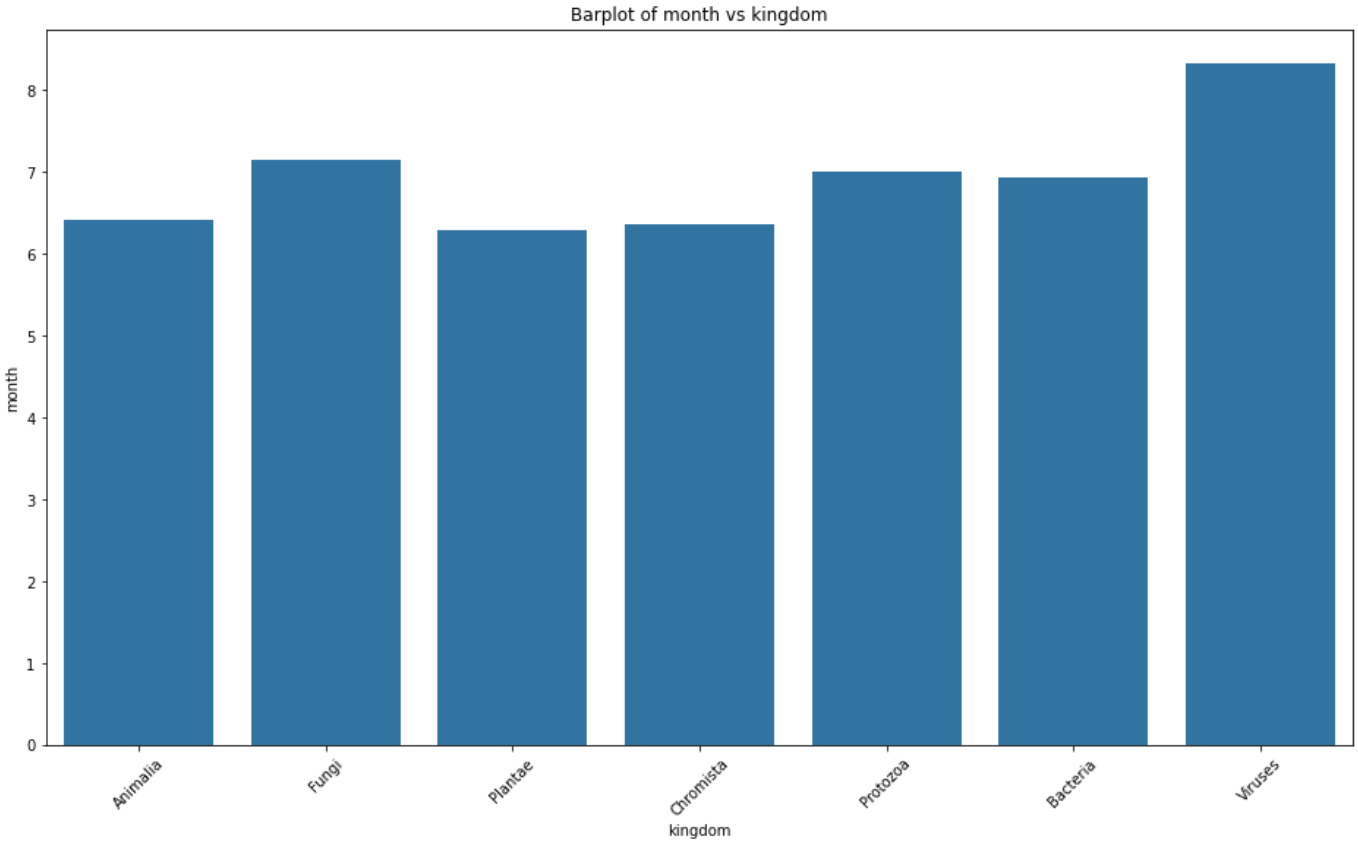

```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

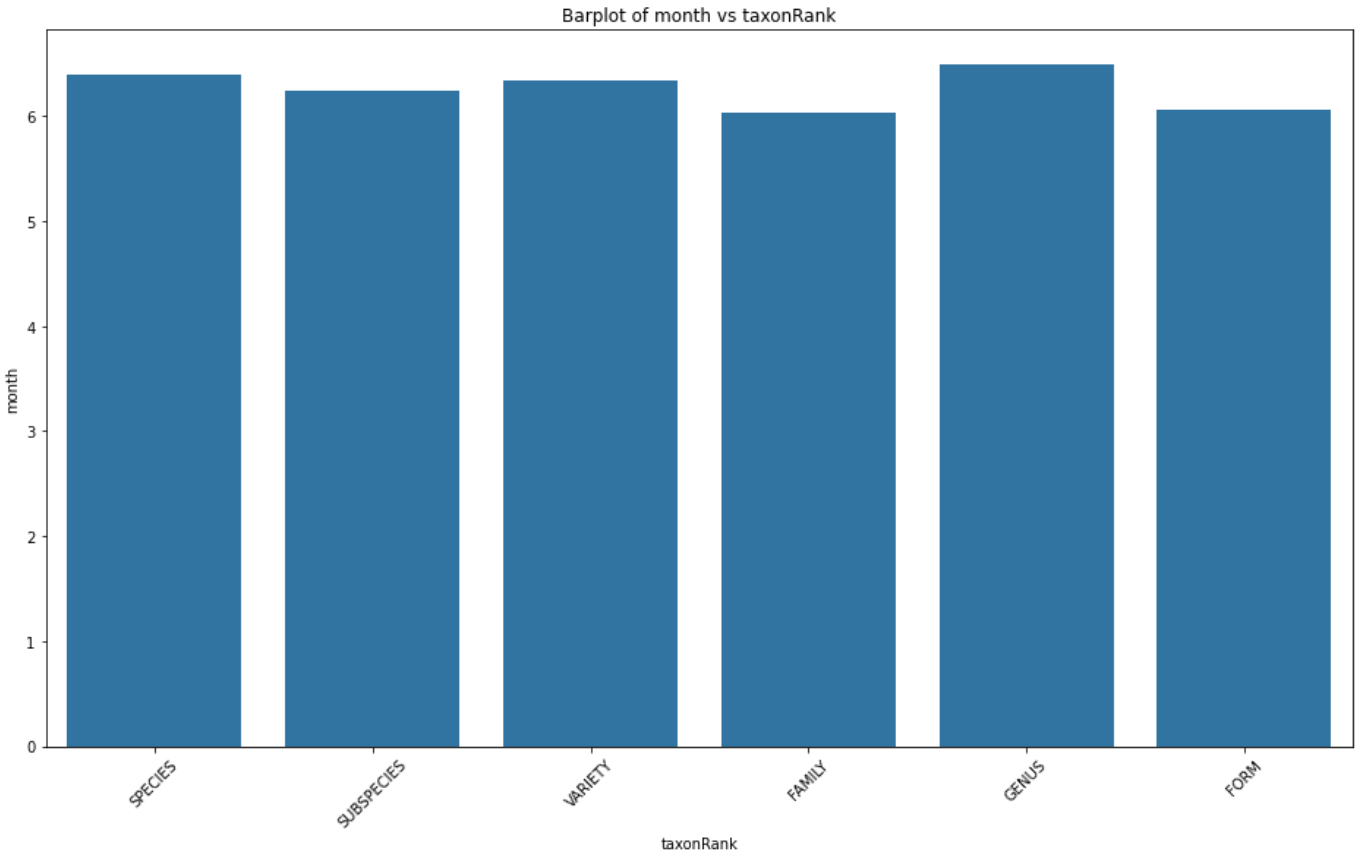



```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

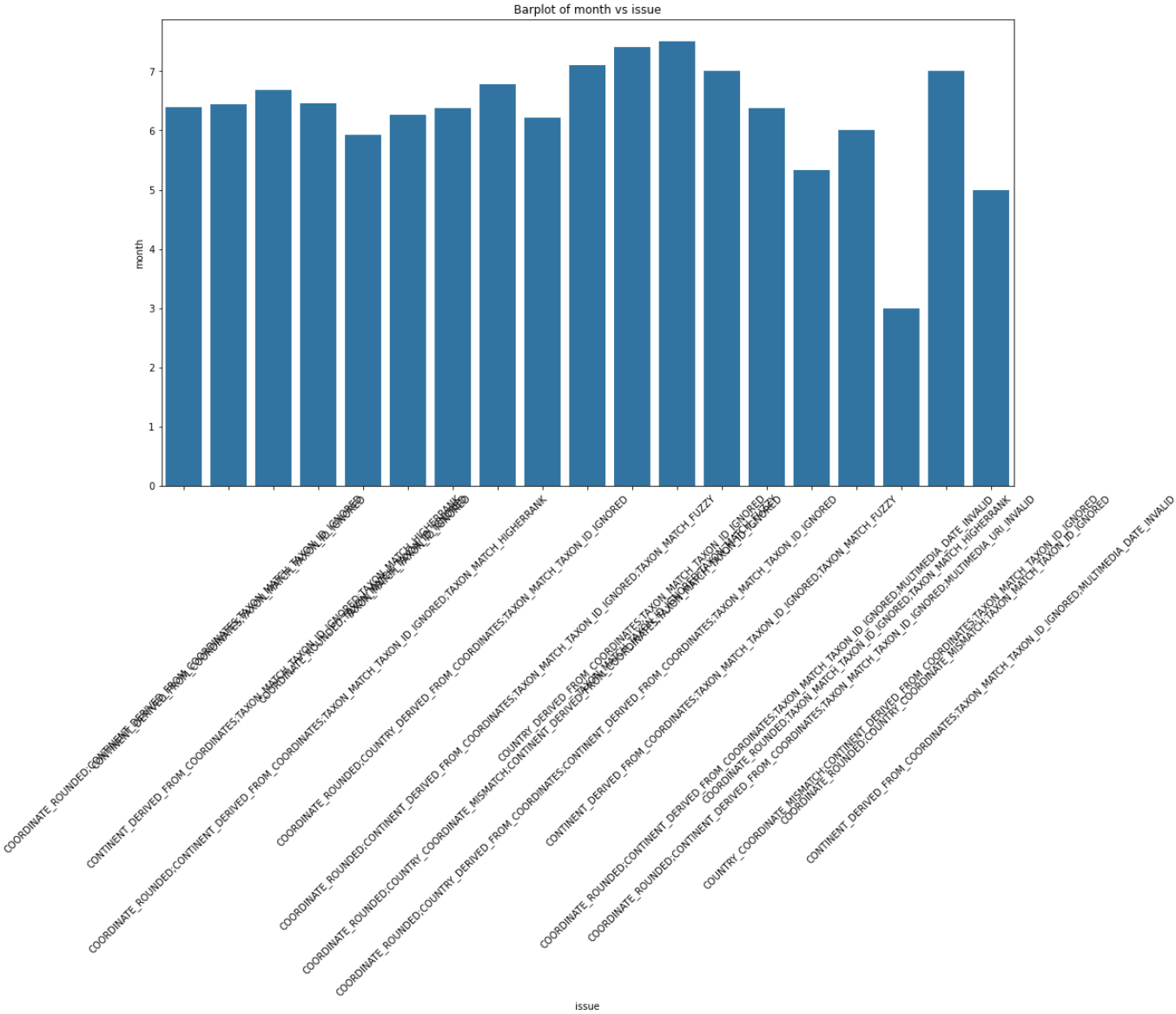



```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```

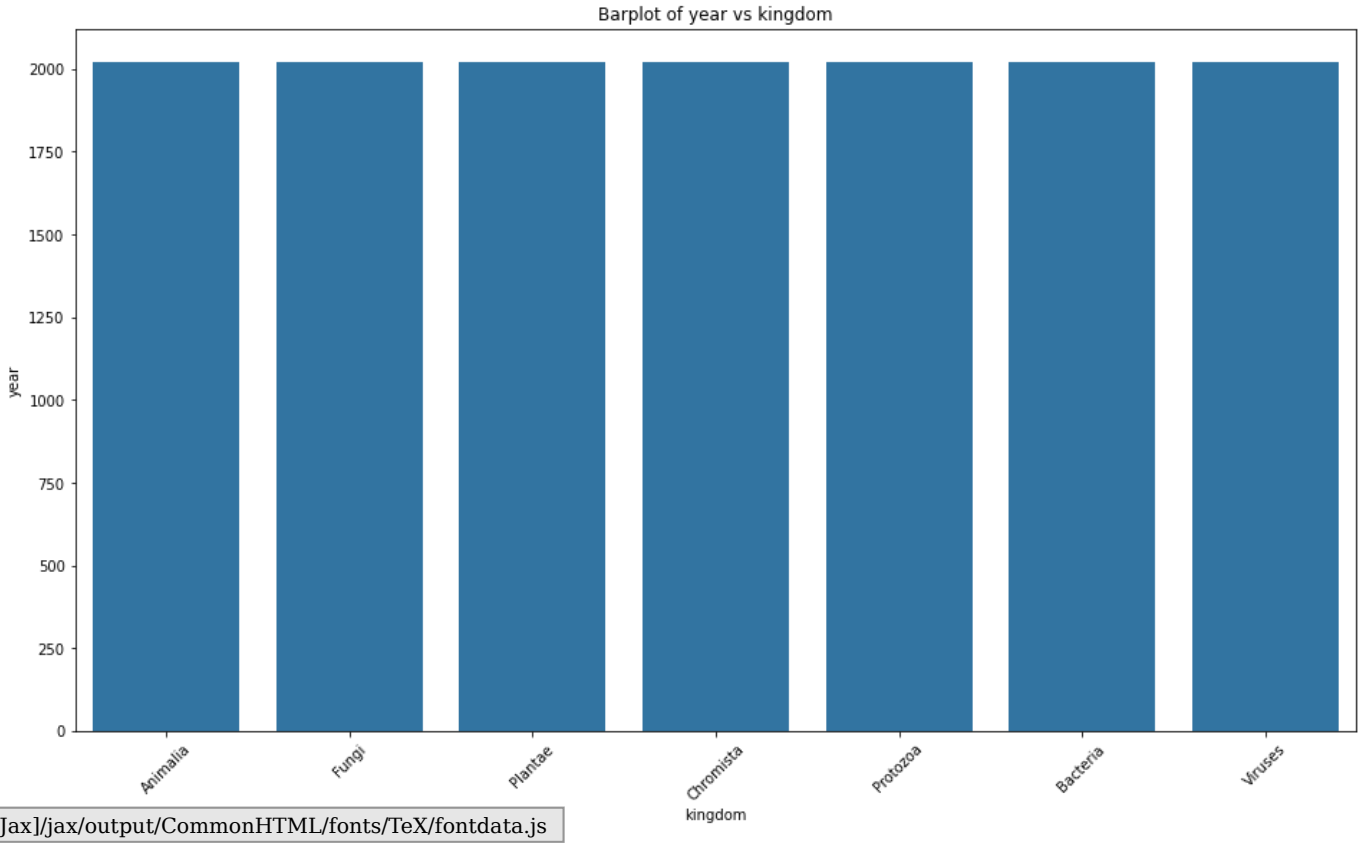



```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

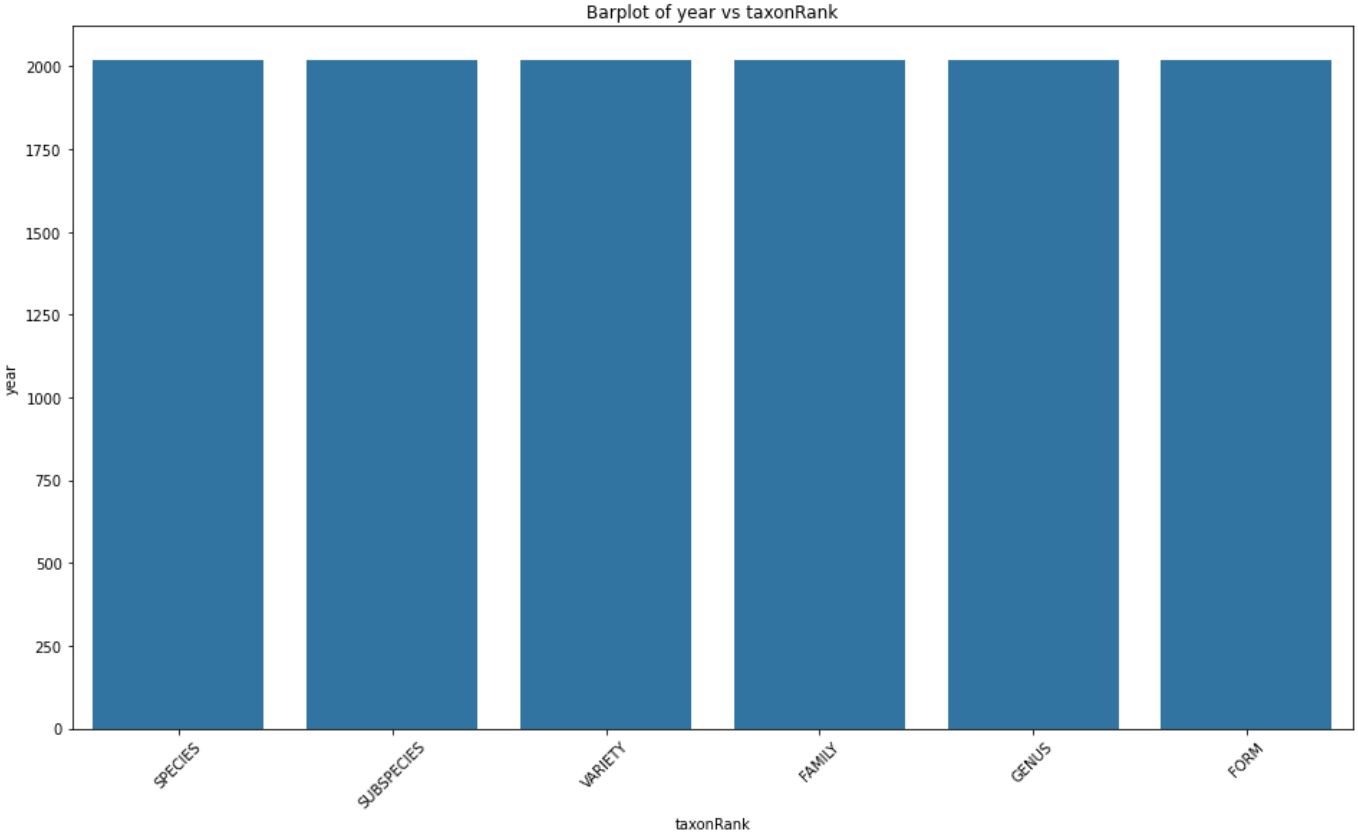



```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```



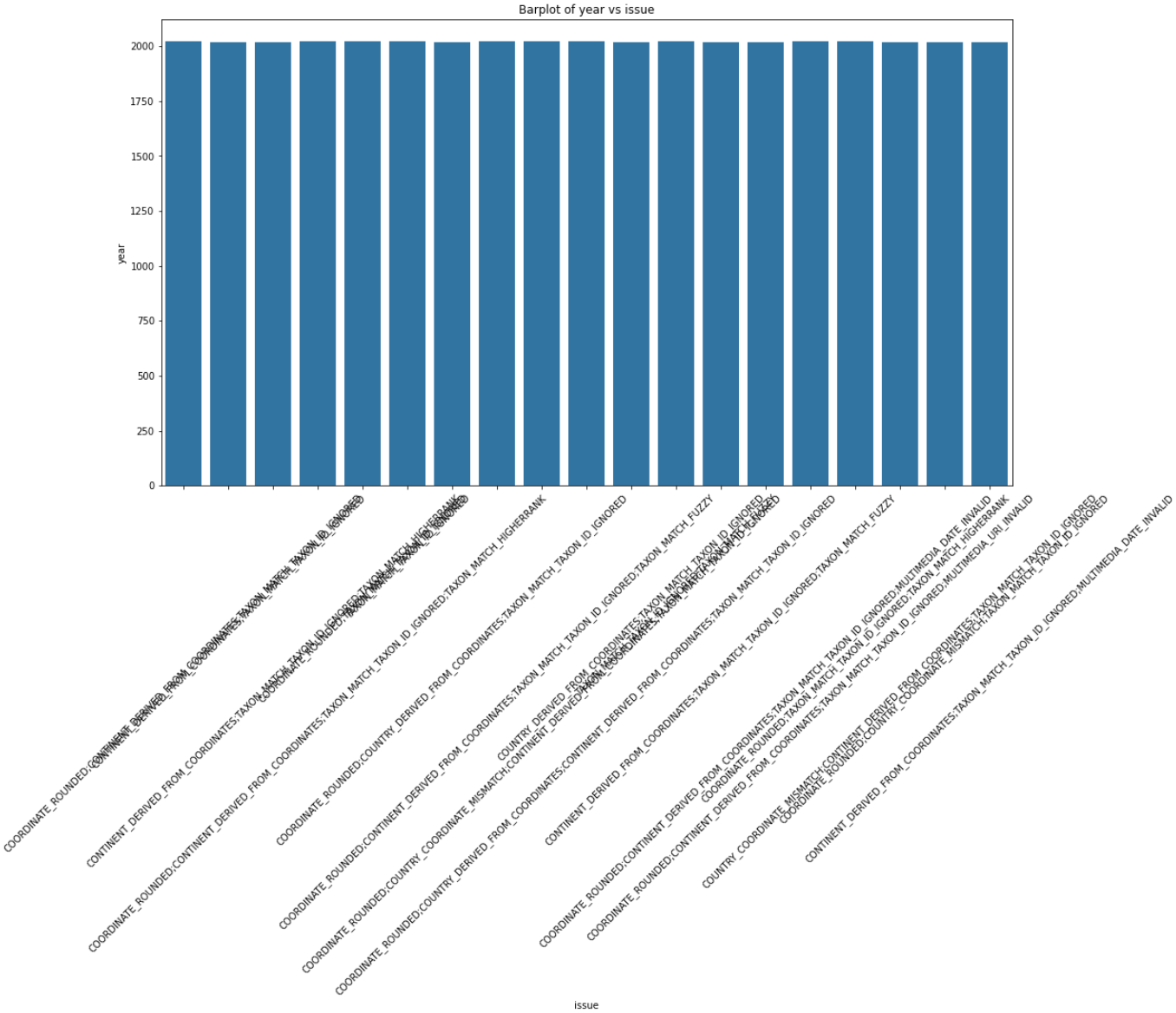
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```

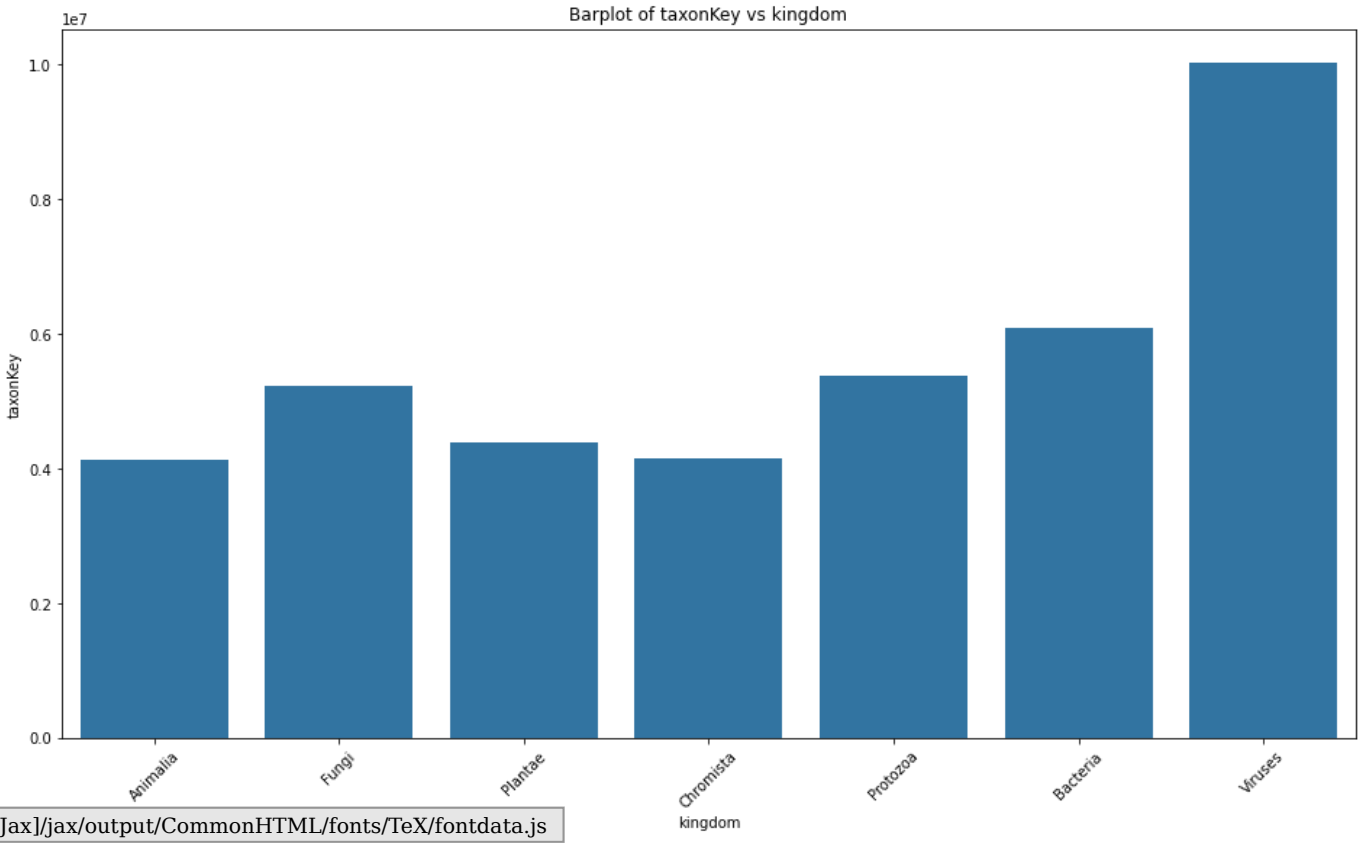



```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

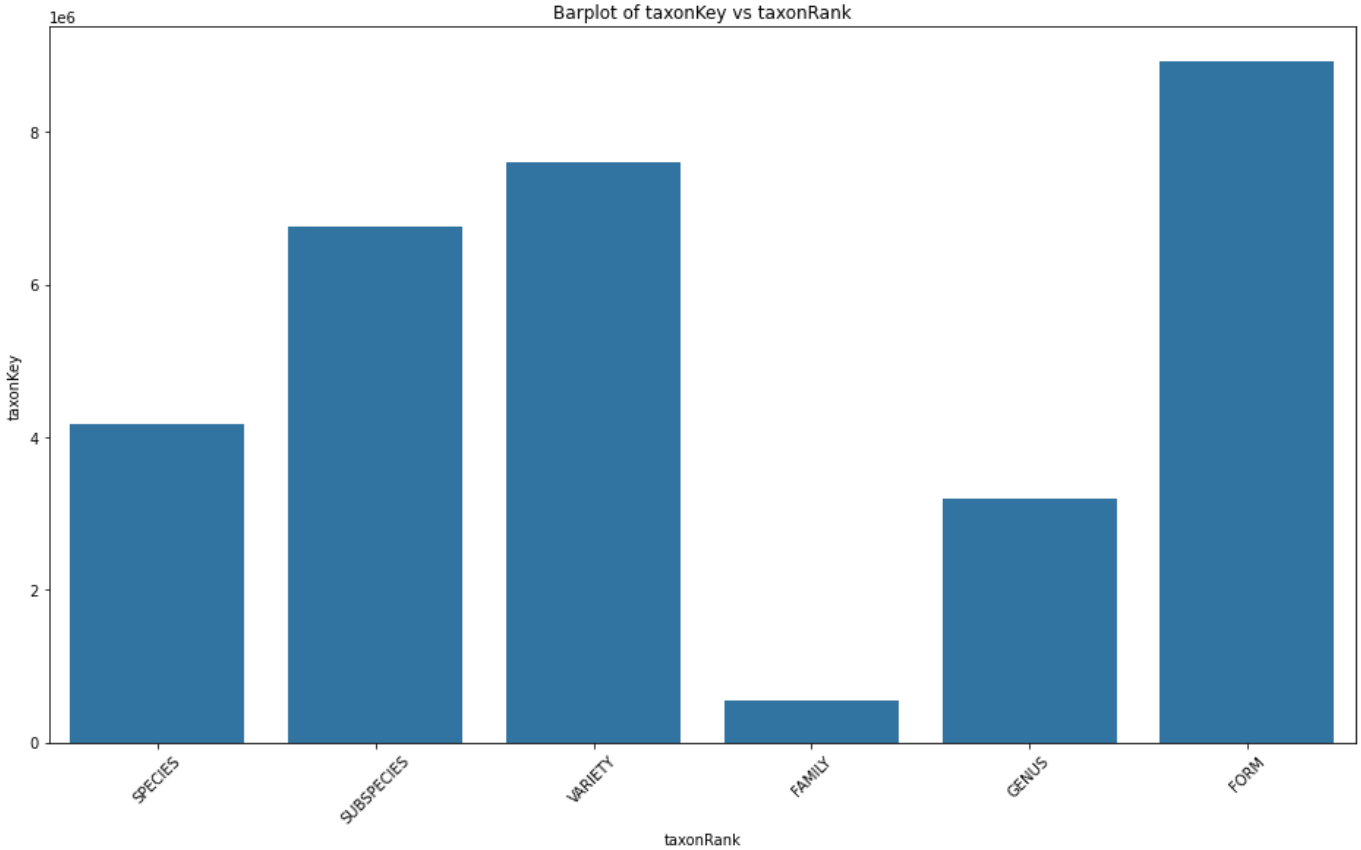



```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```



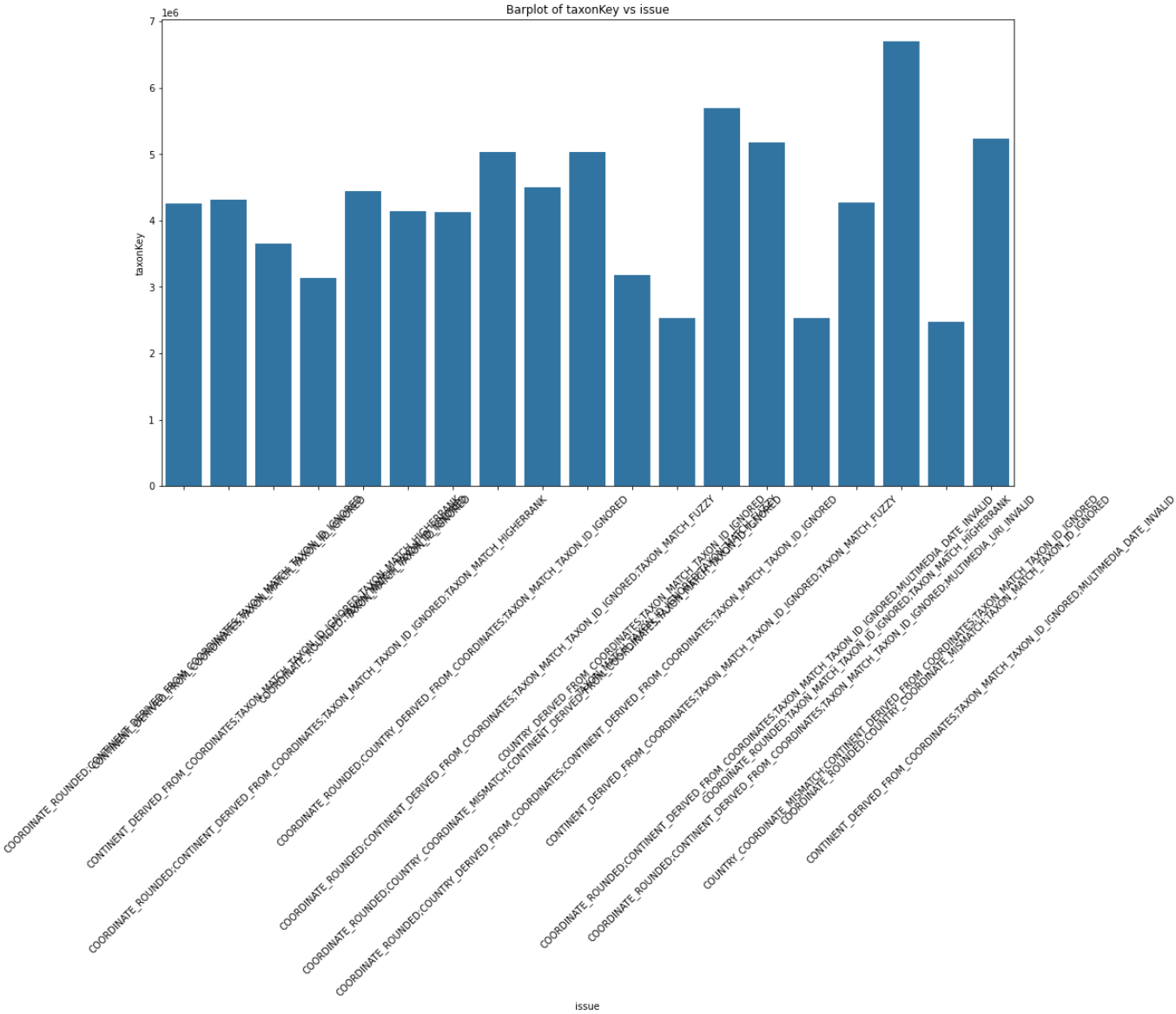
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```

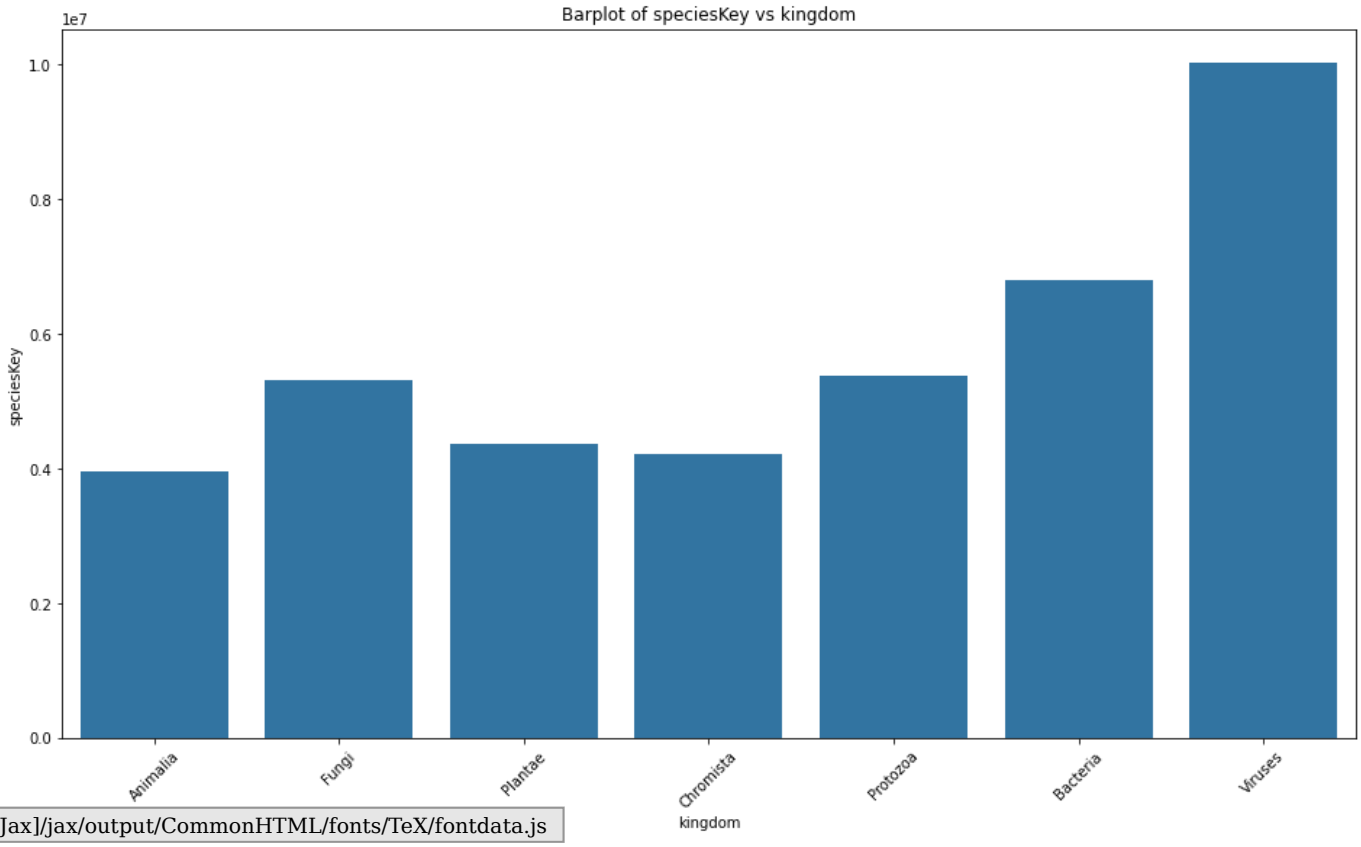



```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

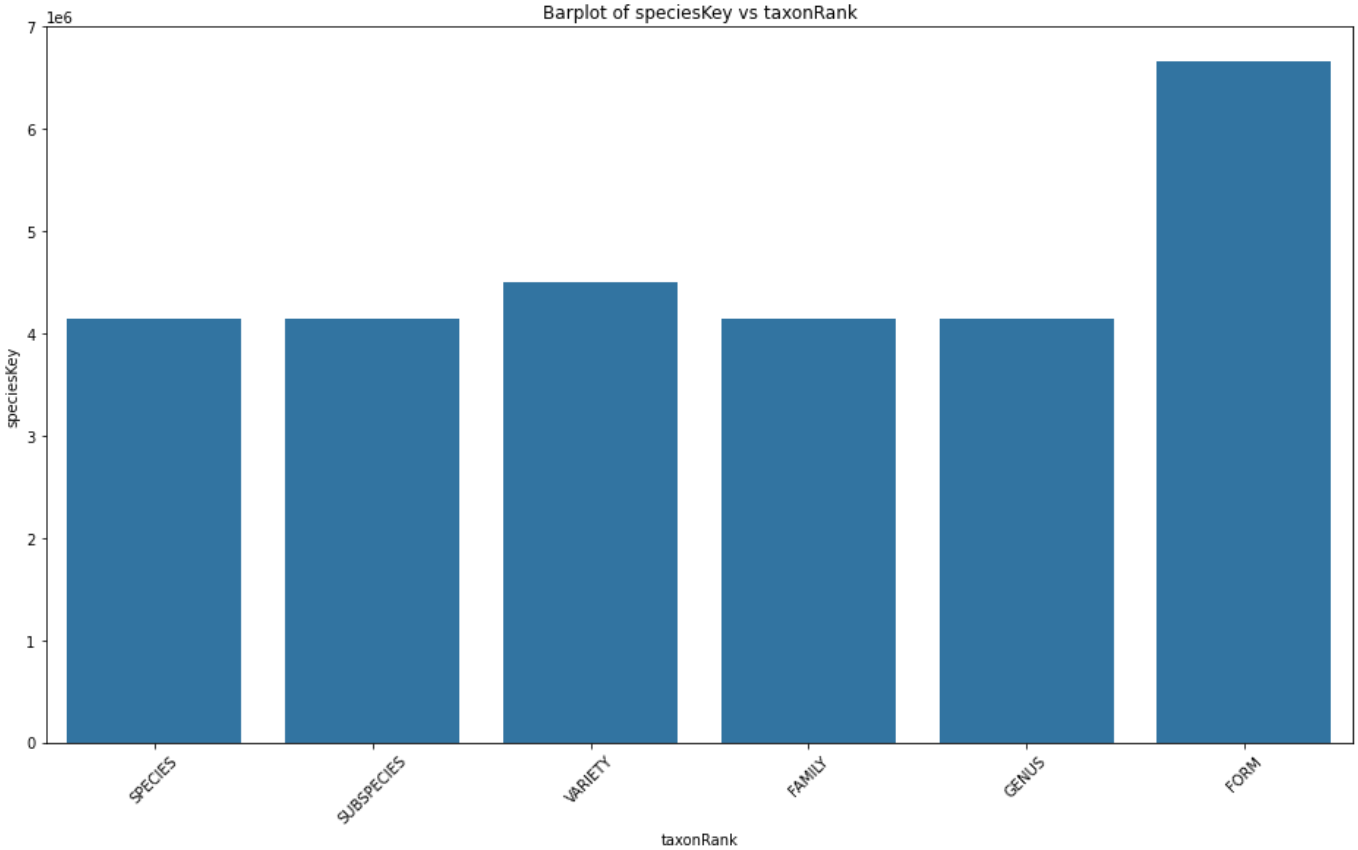



```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```



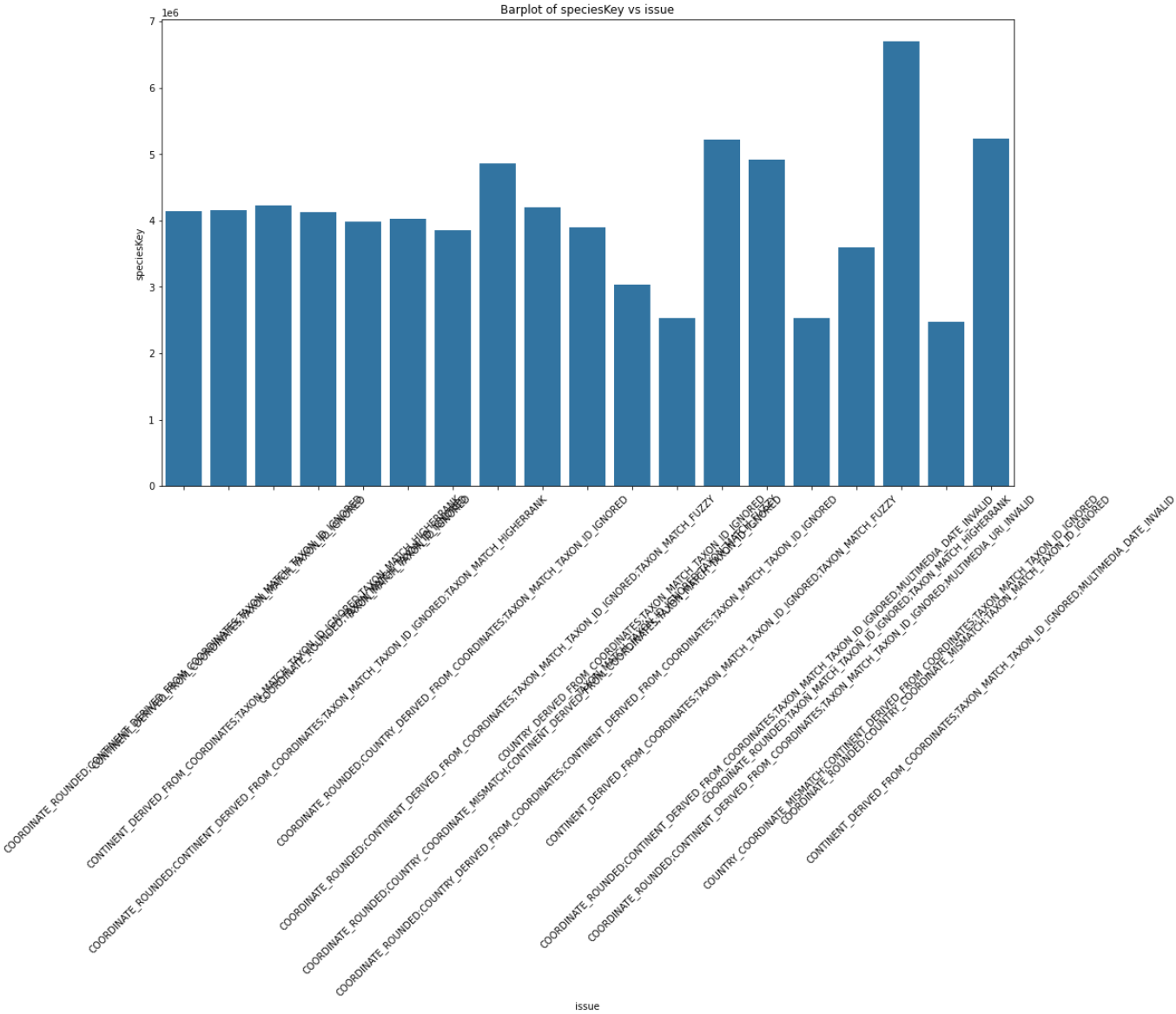
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```

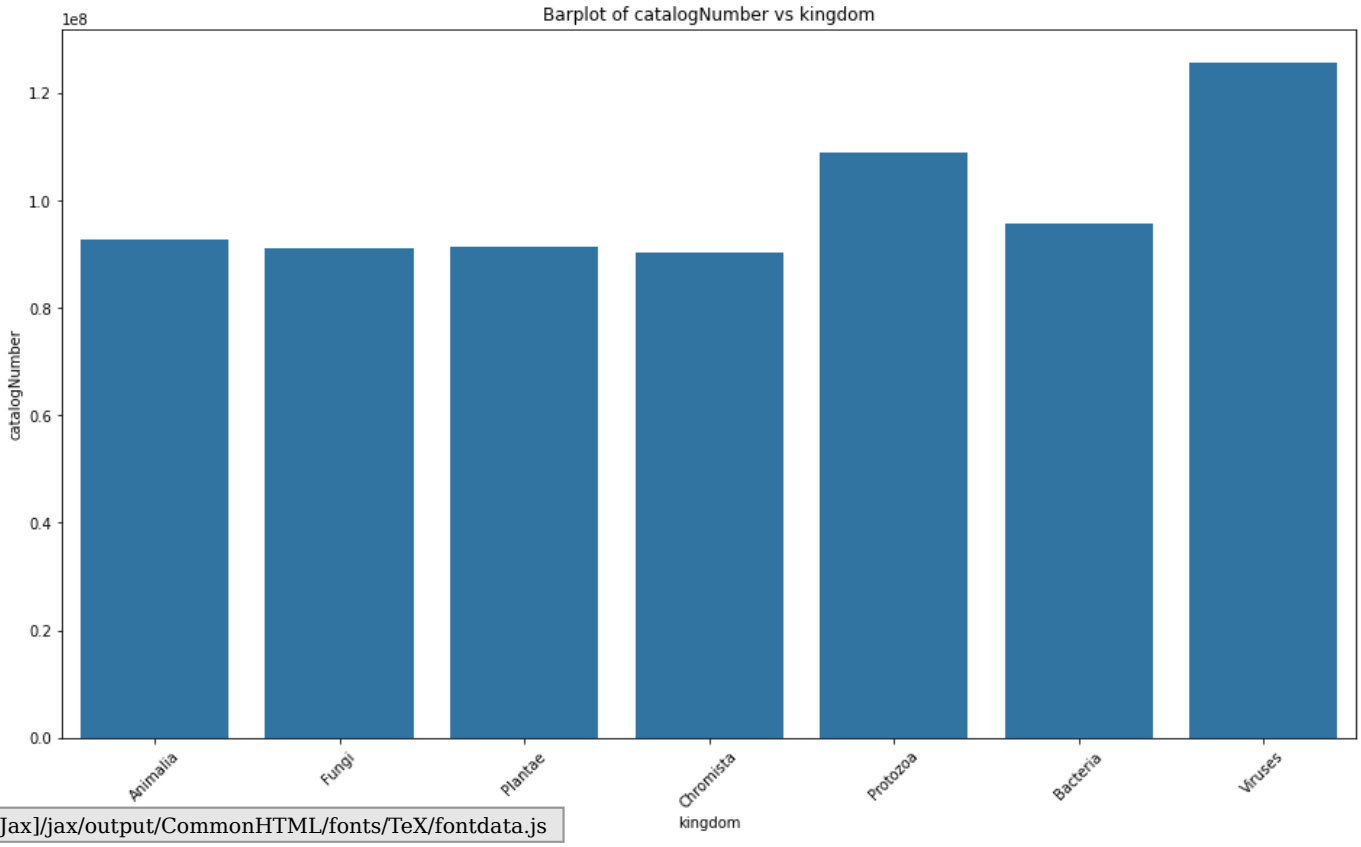



```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

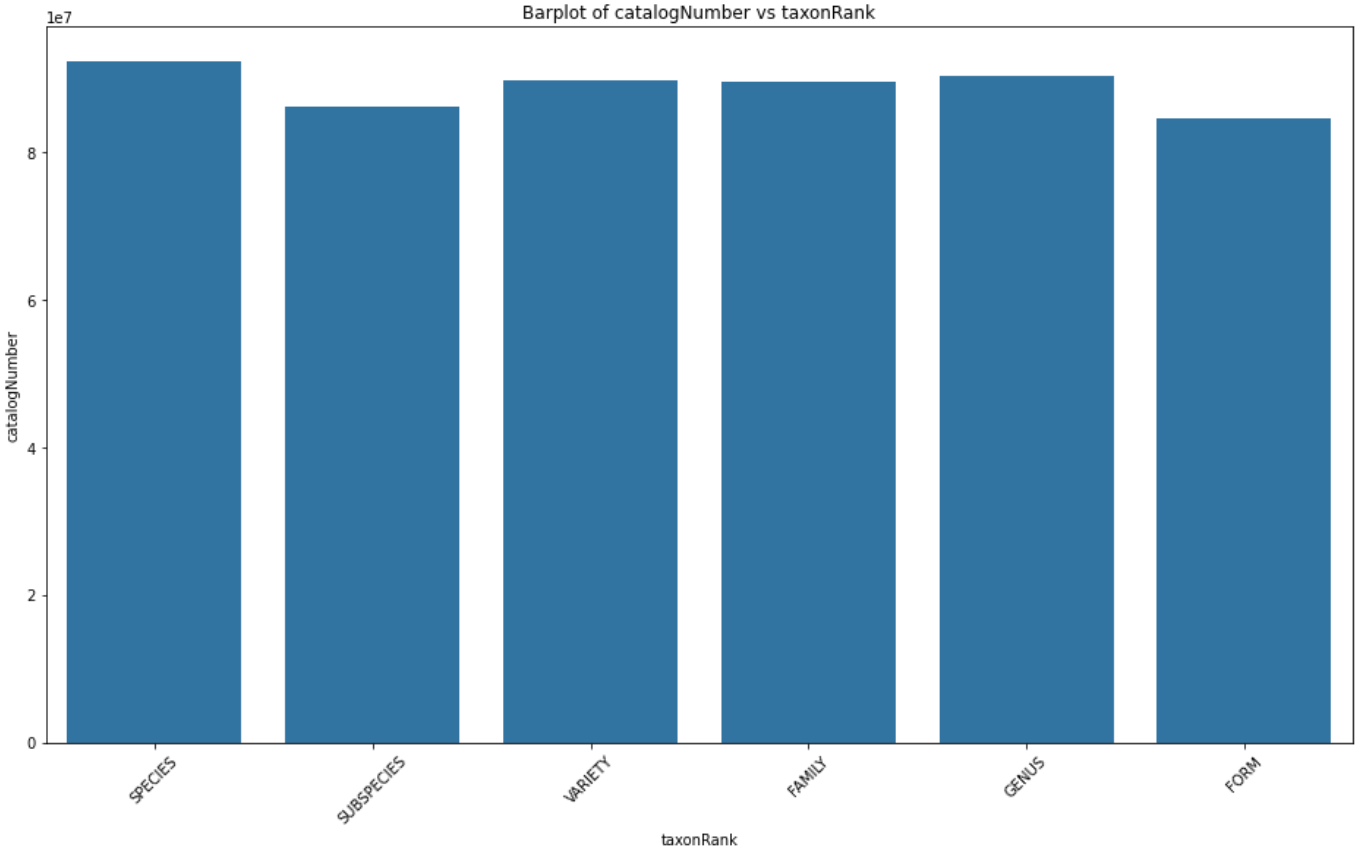



```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```



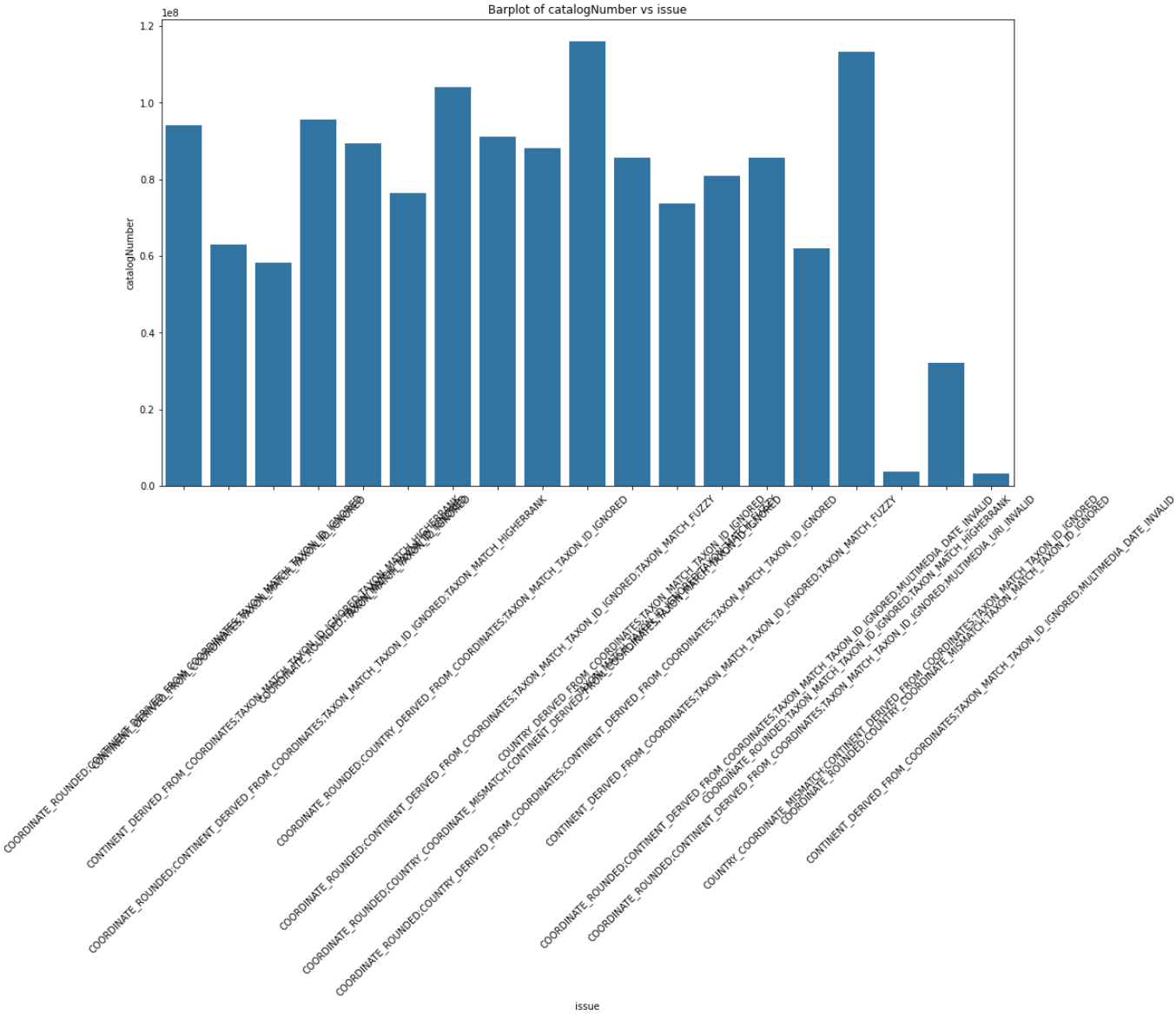
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```




```
C:\Users\ranig\AppData\Local\Temp\ipykernel_6644\4188629591.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x=df[j], y=df[i], ci=None)  # ci=None removes the confidence interval ba
rs if not needed
```

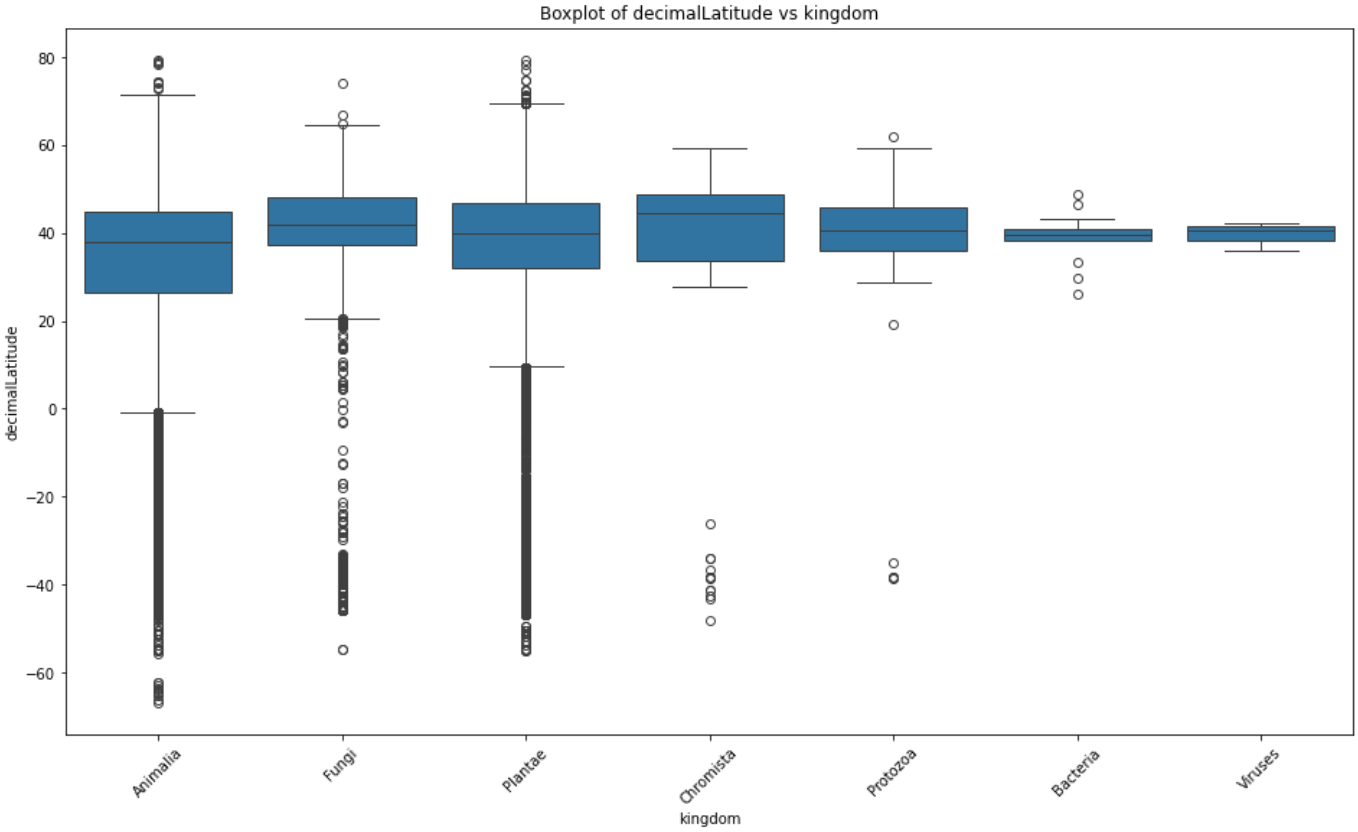

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

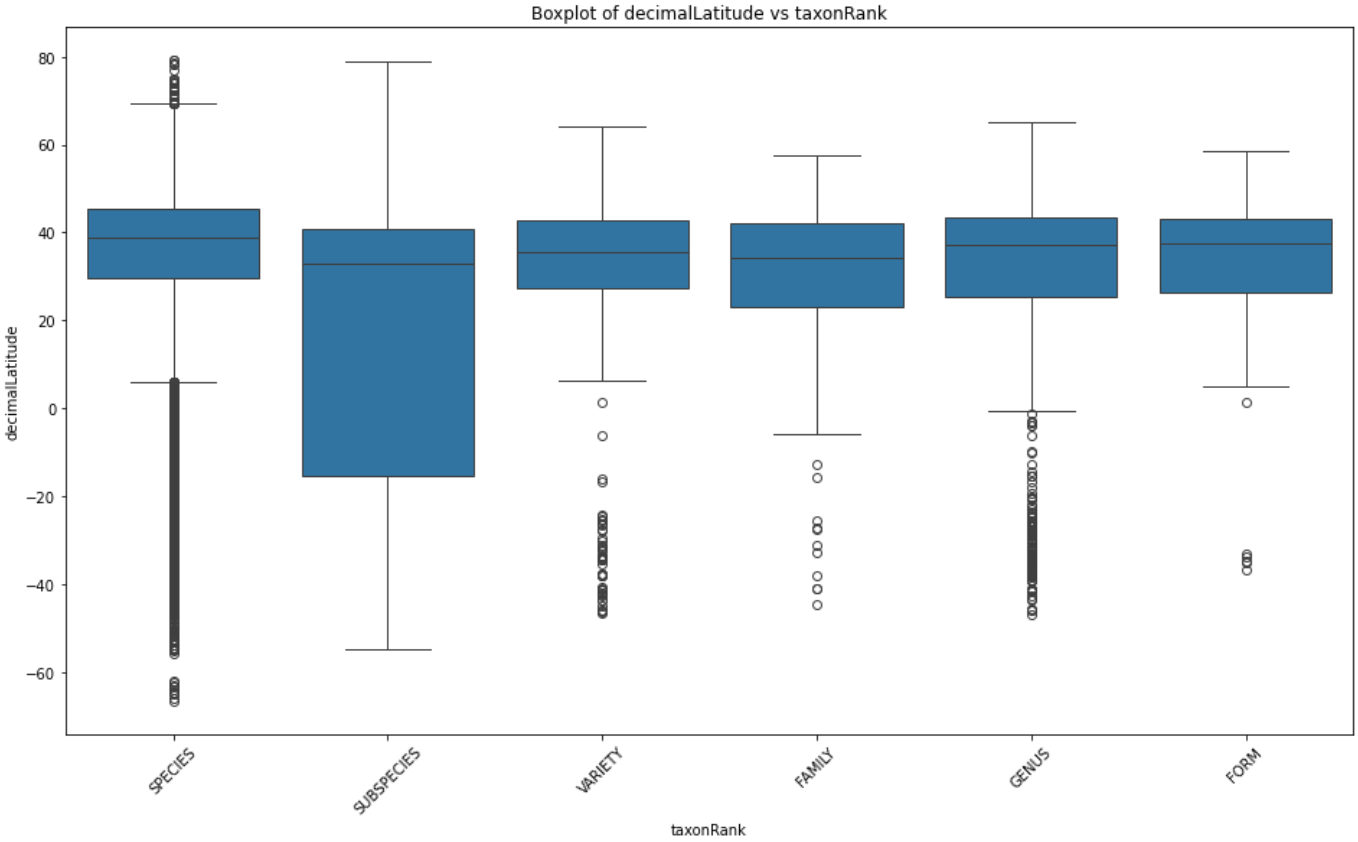


In [38]:

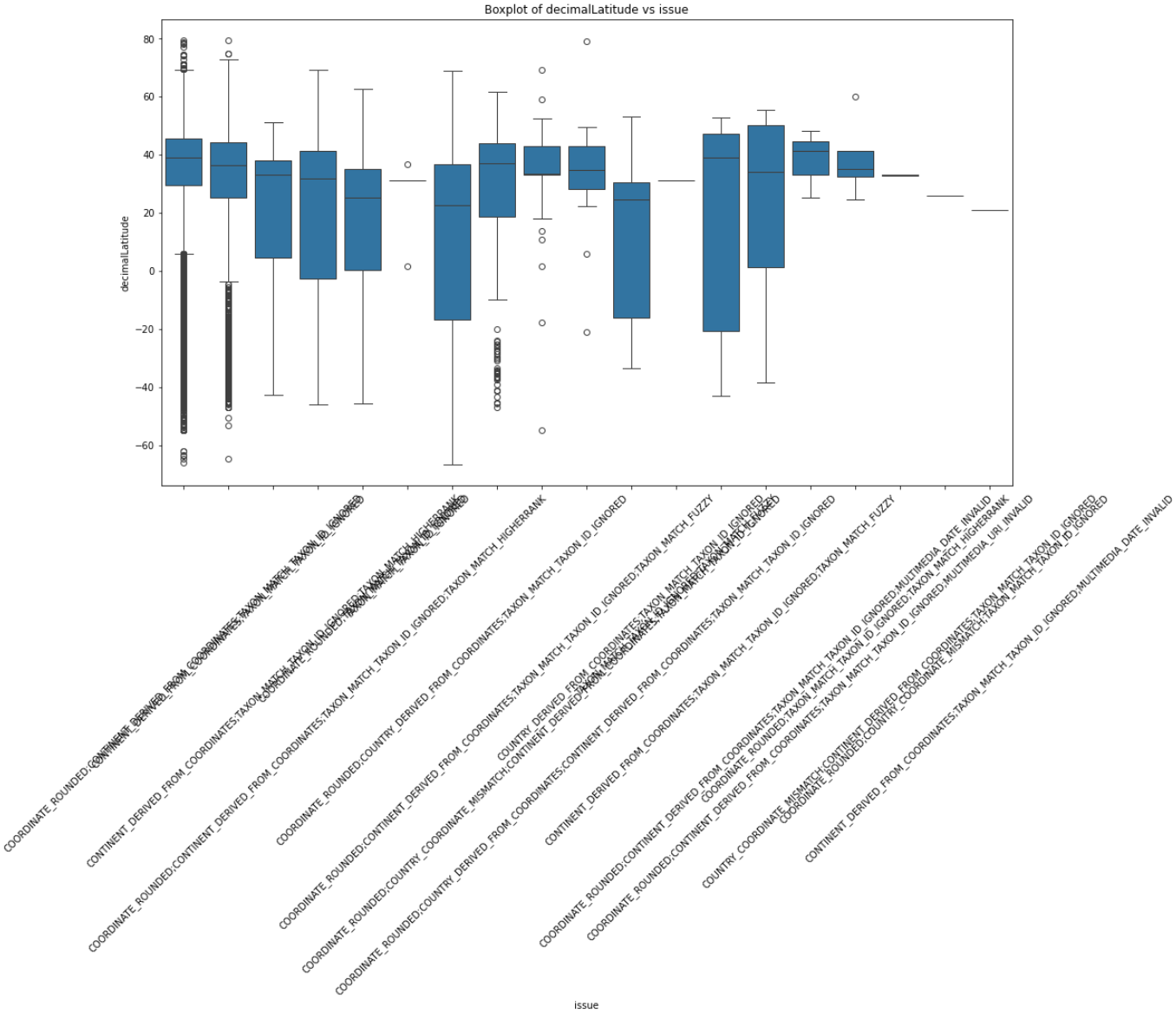
```
# box plot of nummerical and categorical columns
# Loop through each numerical and categorical column
for i in num_col:
    for j in filtered_cat_col:
        plt.figure(figsize=(16, 9))

        # Boxplot - Use categorical variable 'j' on x-axis, and numerical 'i' on y-ax
        sns.boxplot(x=df[j], y=df[i])

        plt.title(f'Boxplot of {i} vs {j}')
        plt.xlabel(j)
        plt.ylabel(i)
        plt.xticks(rotation=45)  # Rotate x-axis labels for better readability if nec
        plt.show()
```

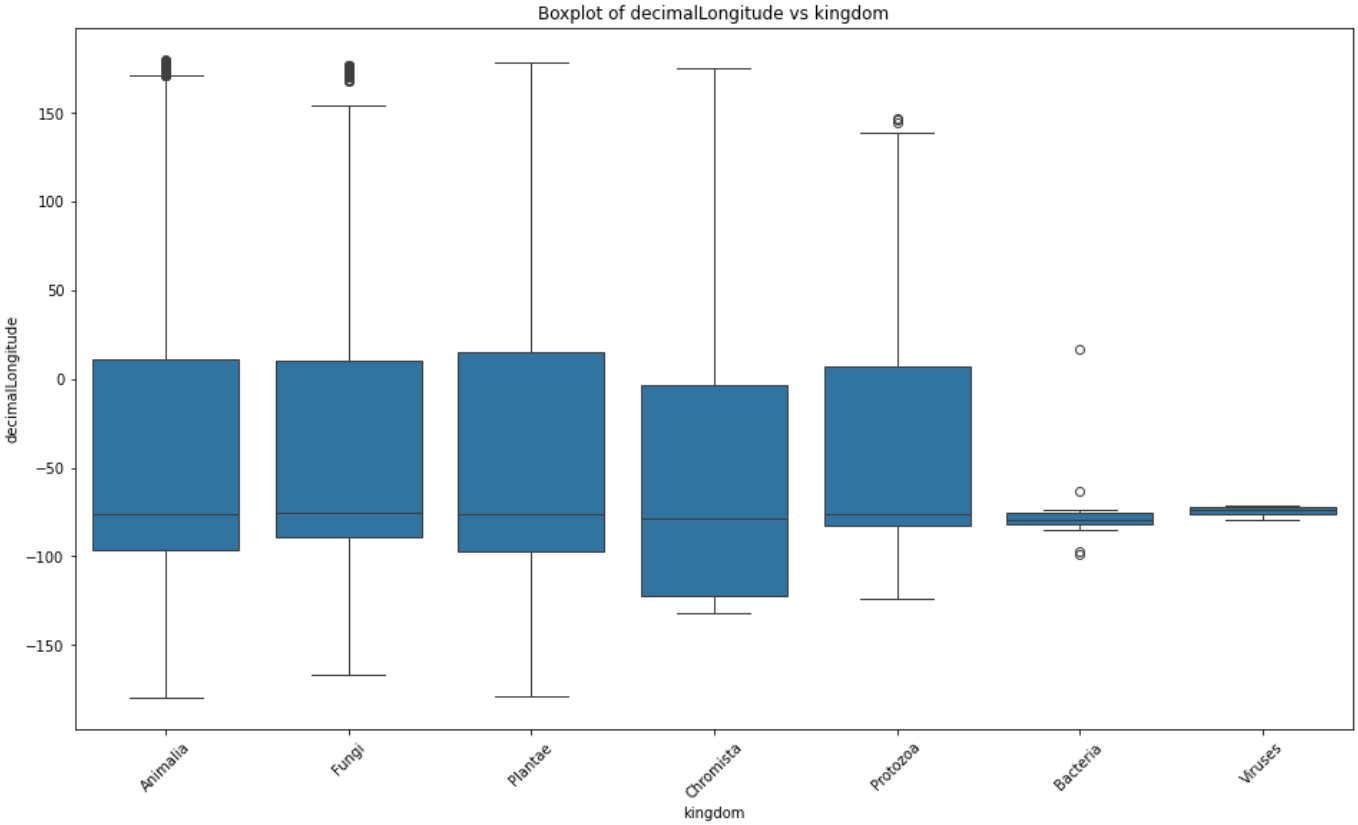
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



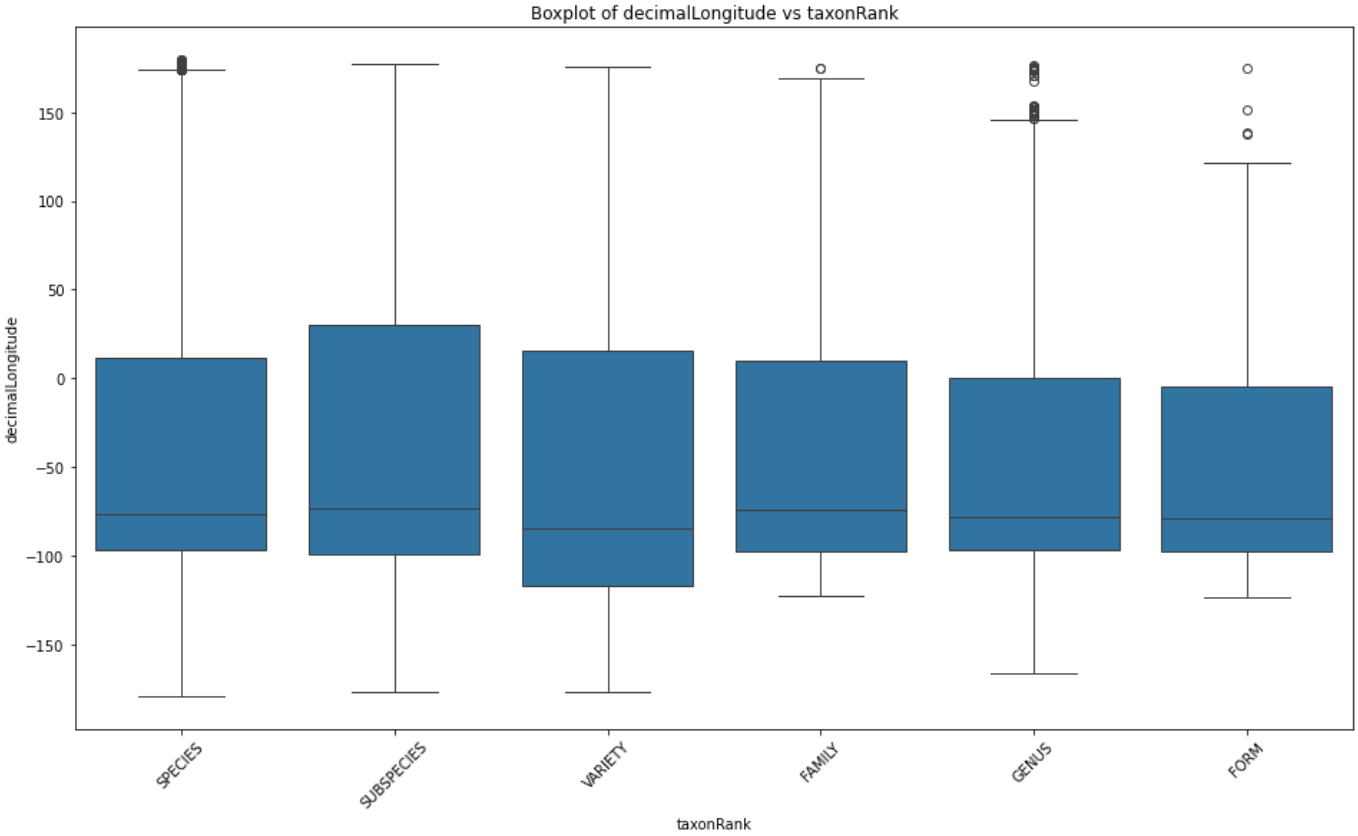


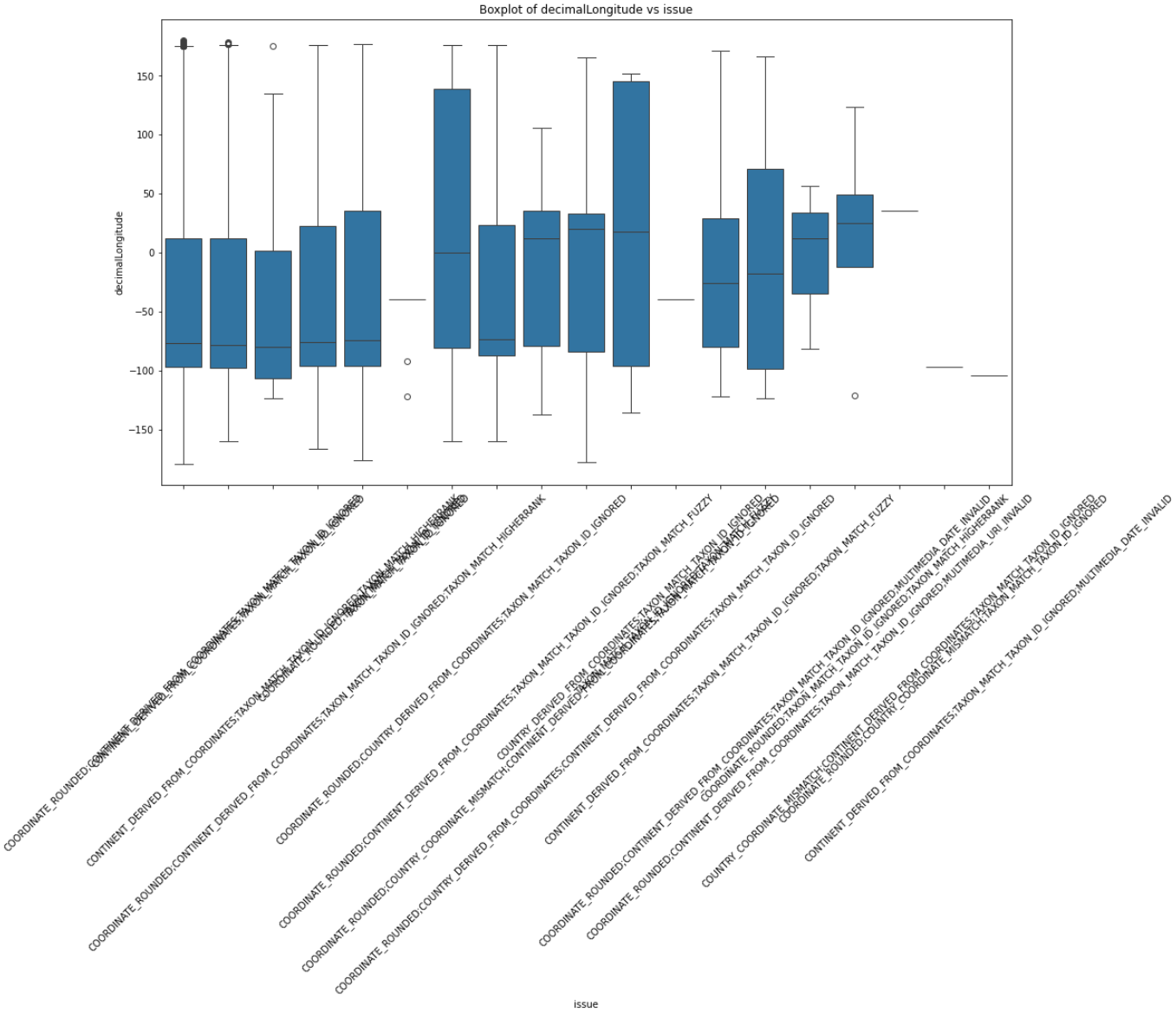
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



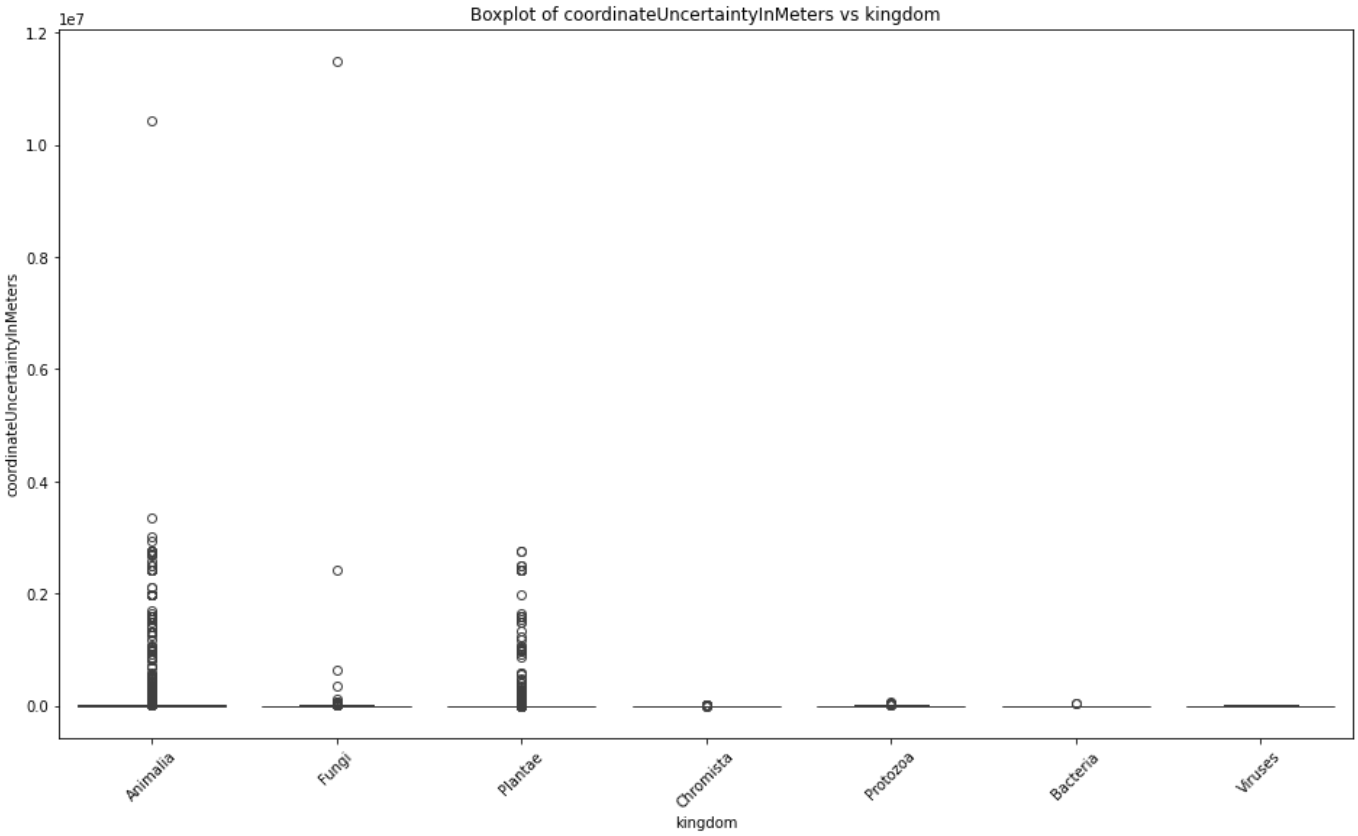


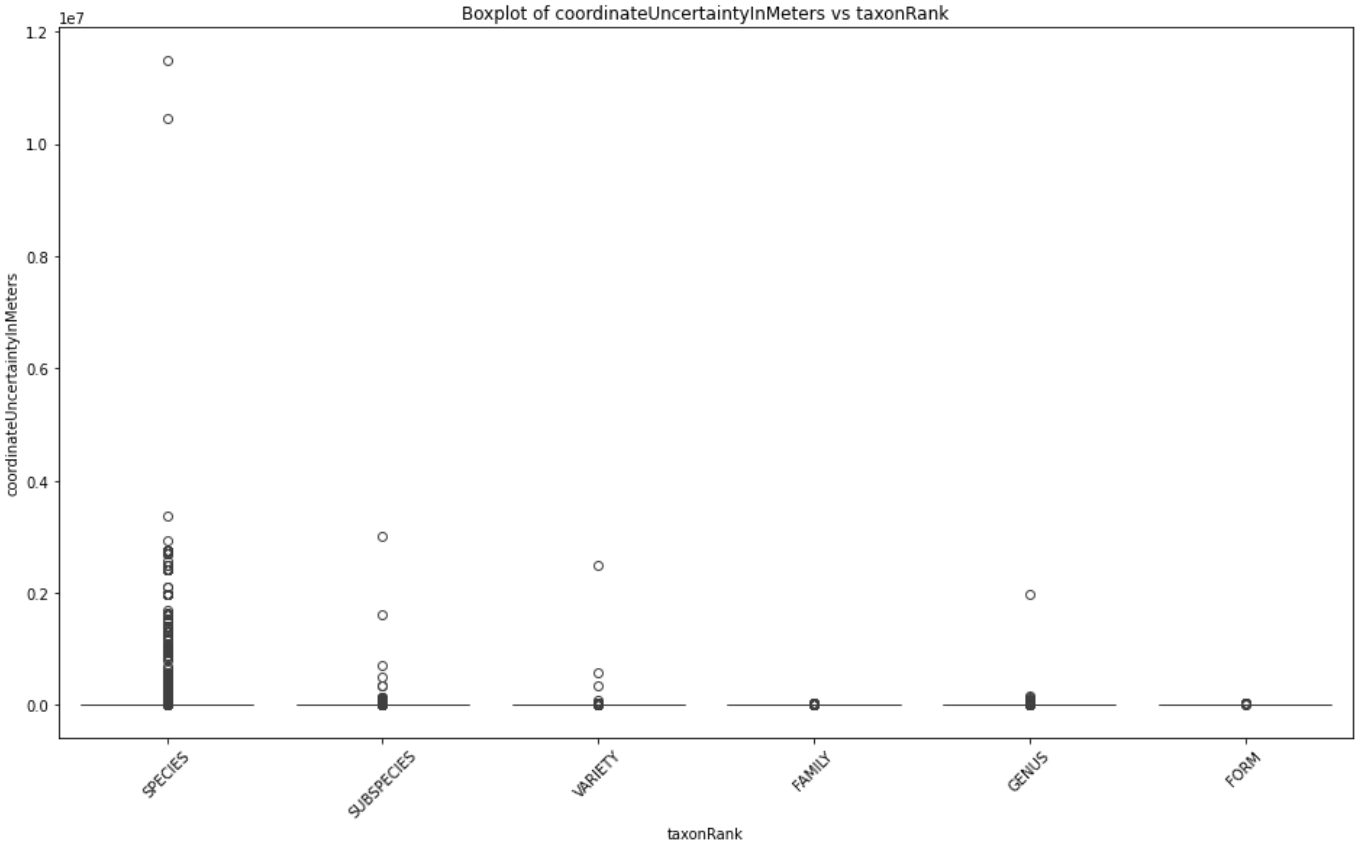
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



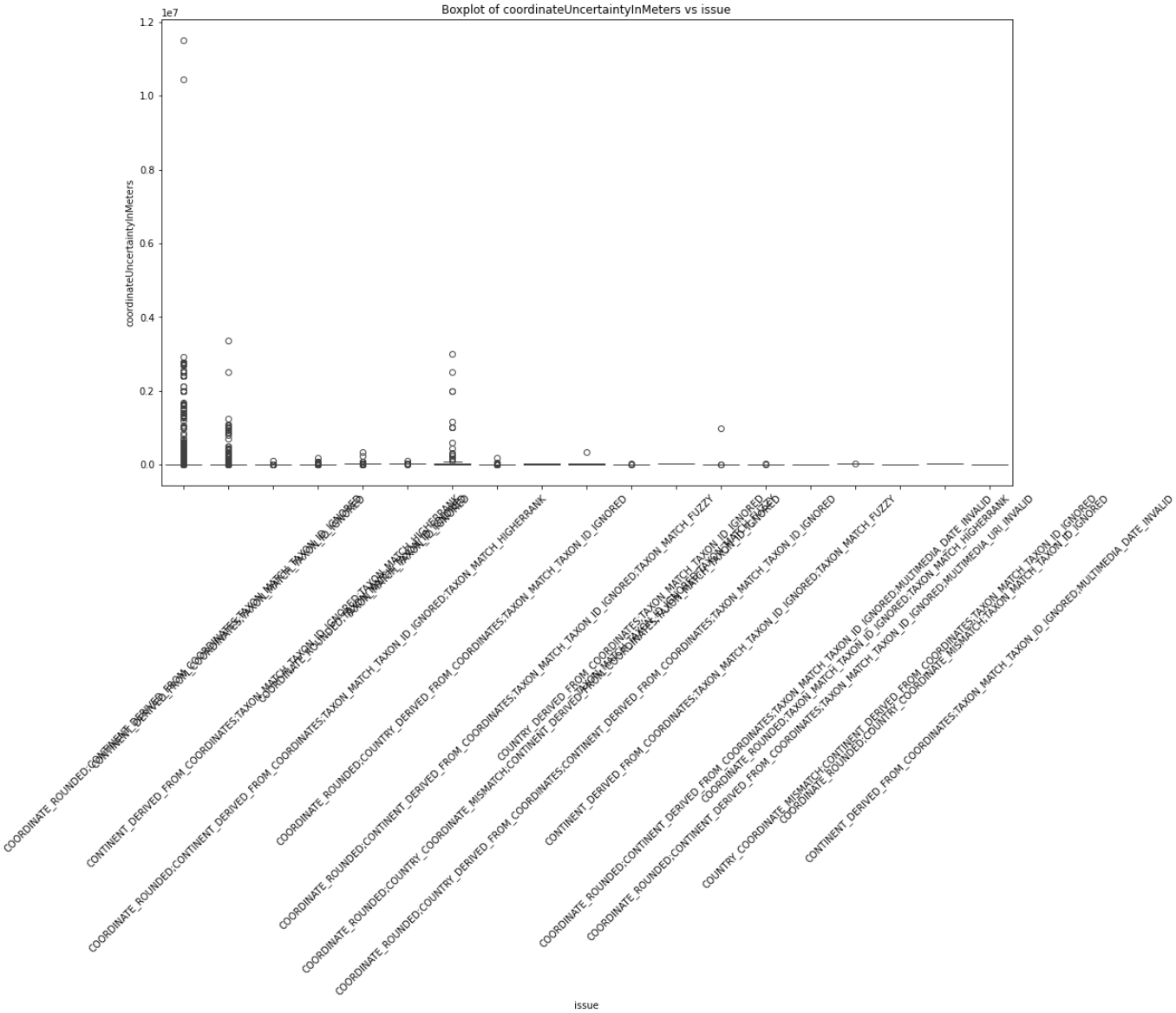


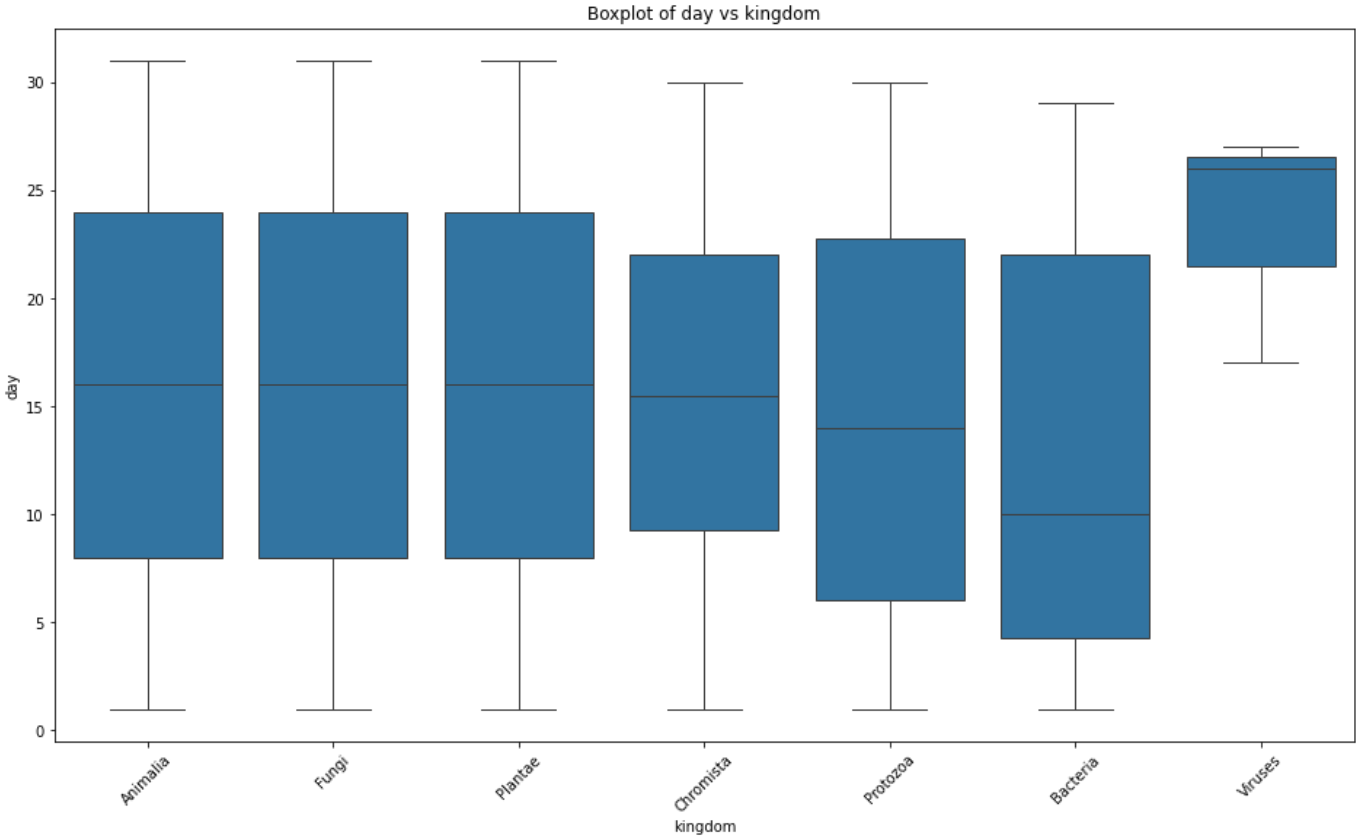
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



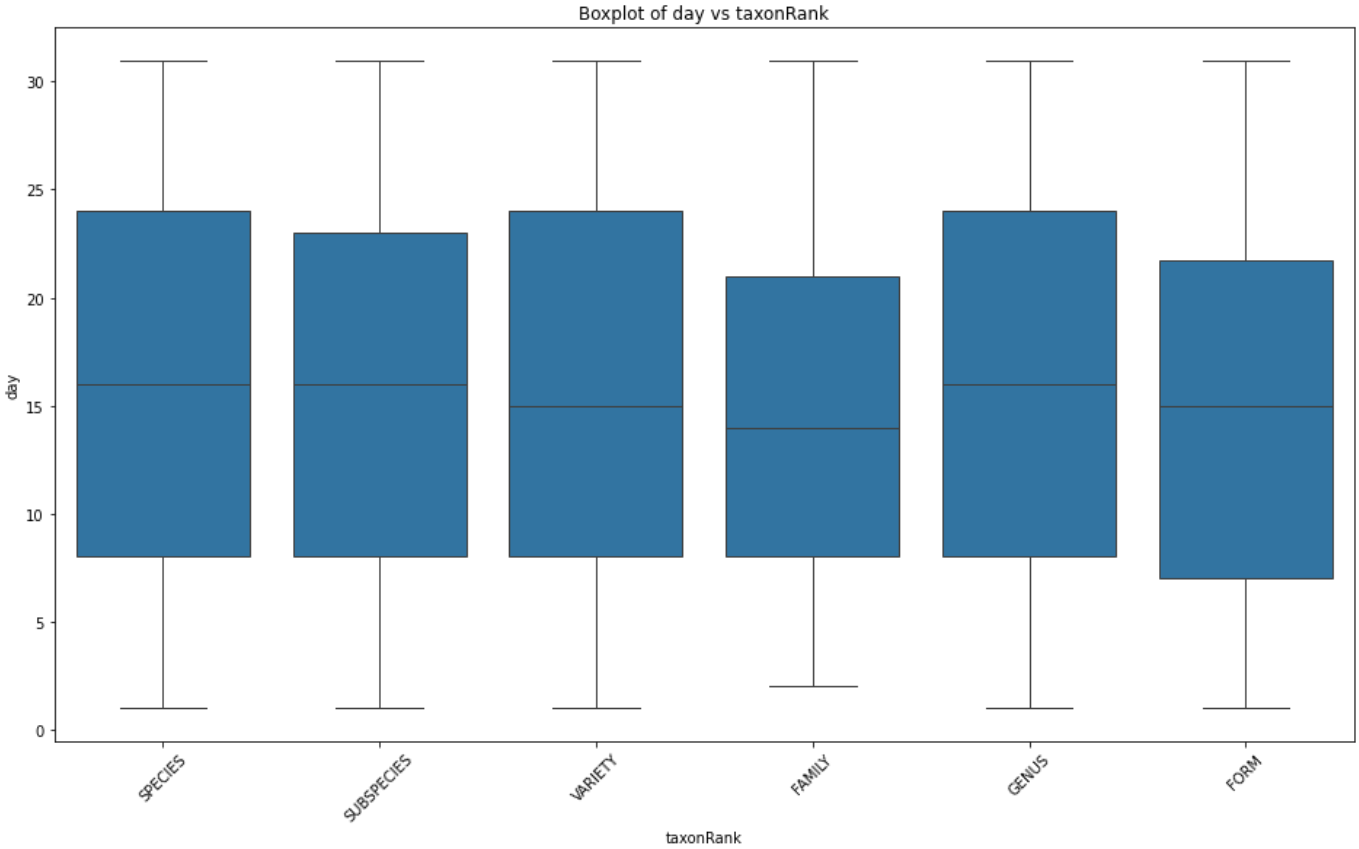


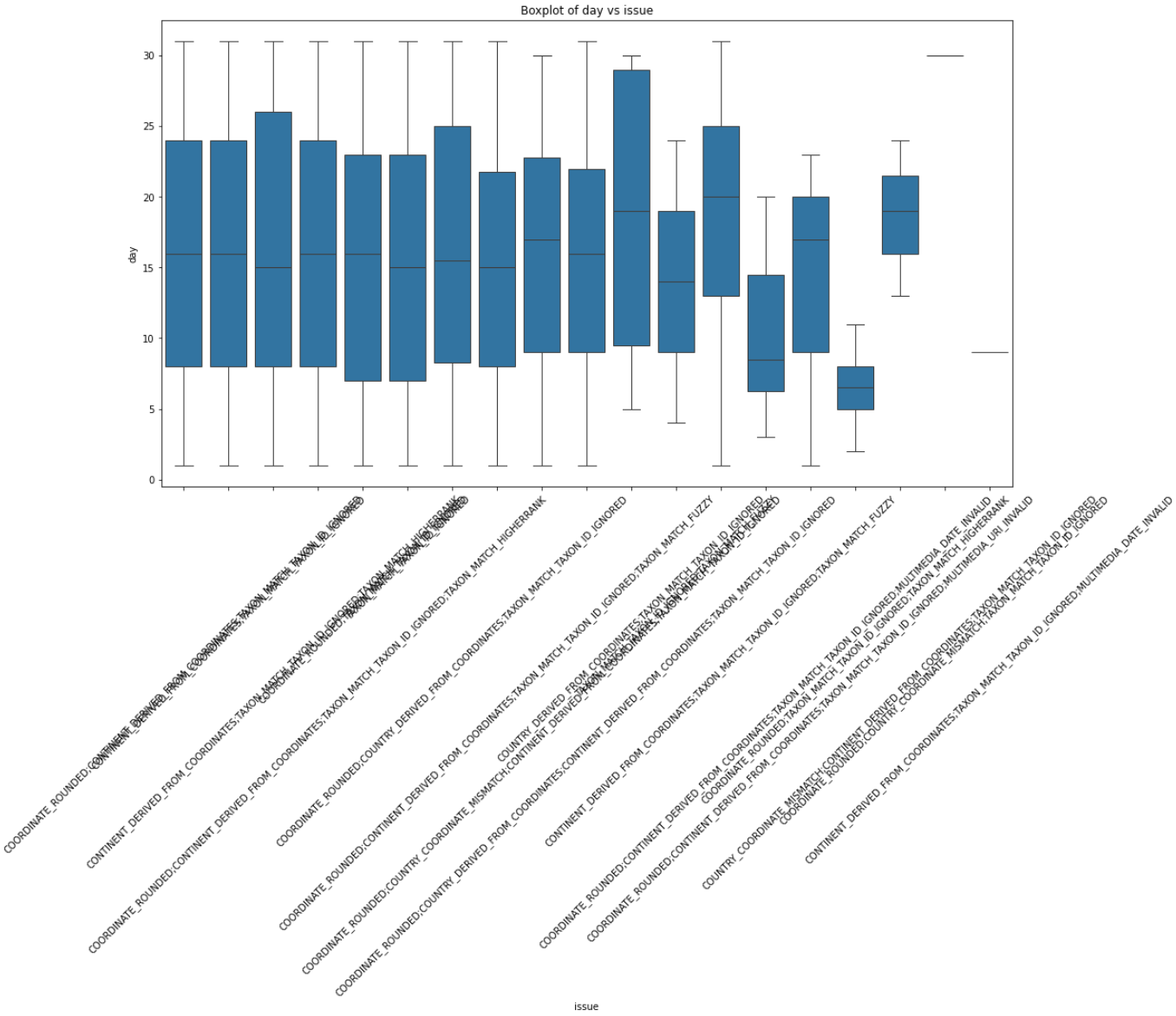
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



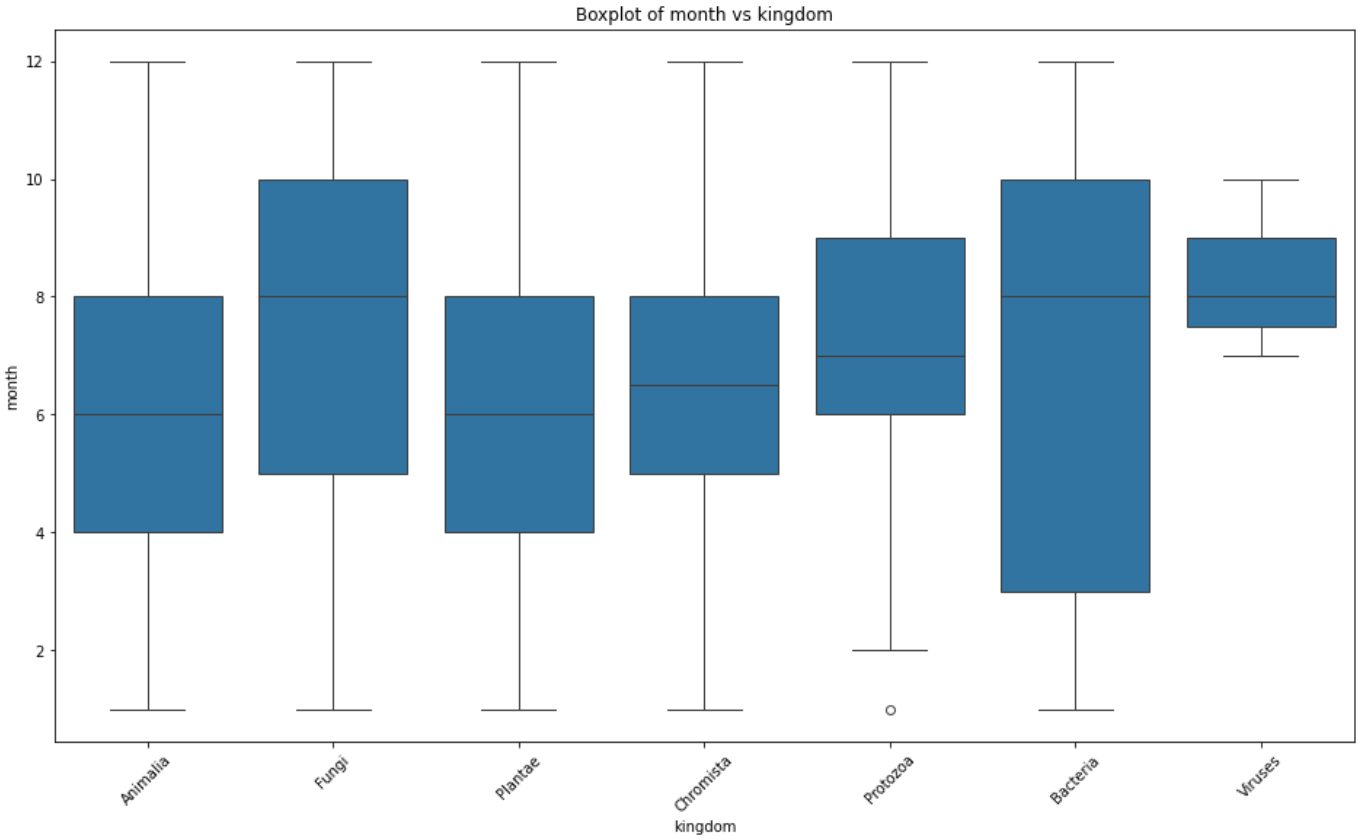


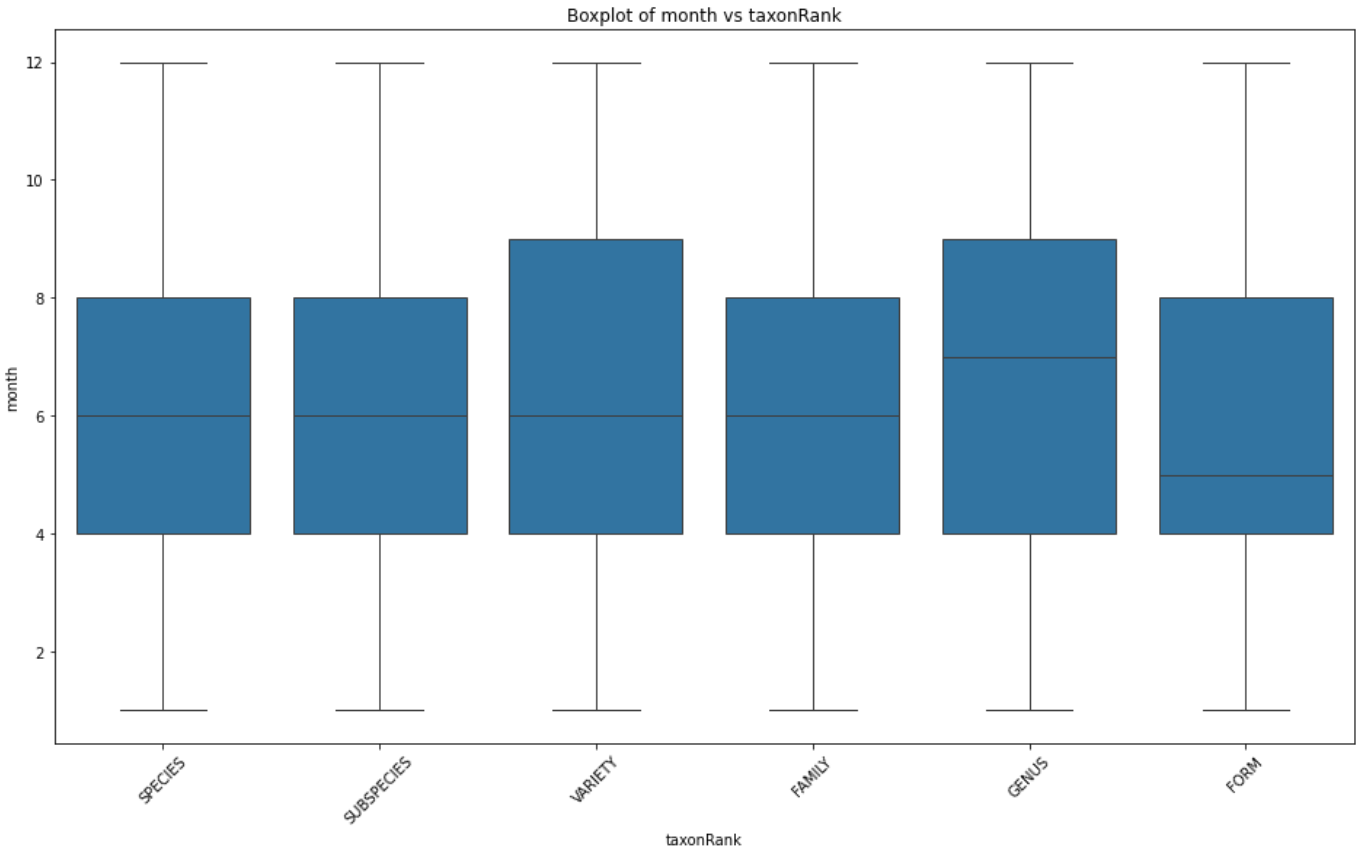
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



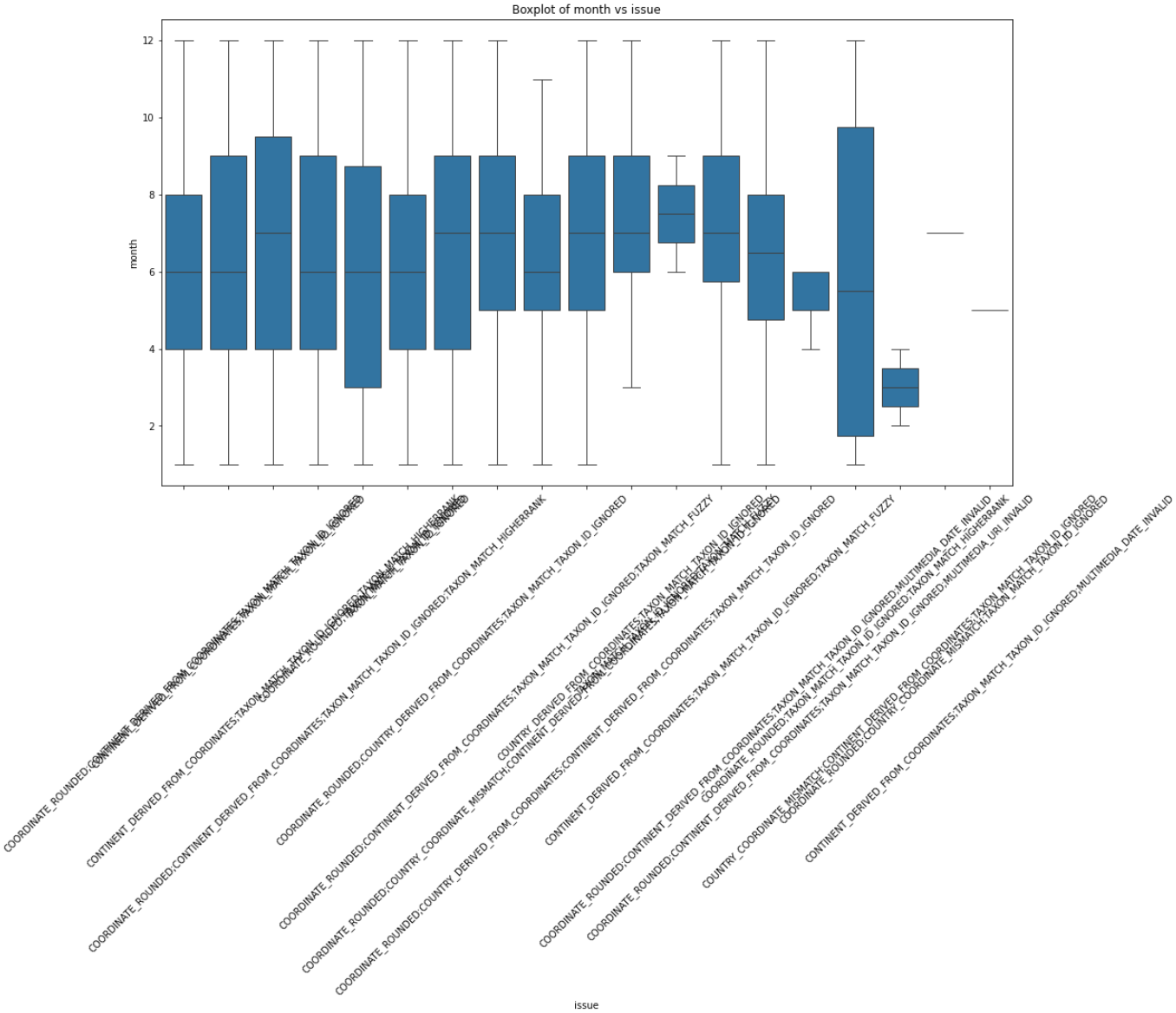


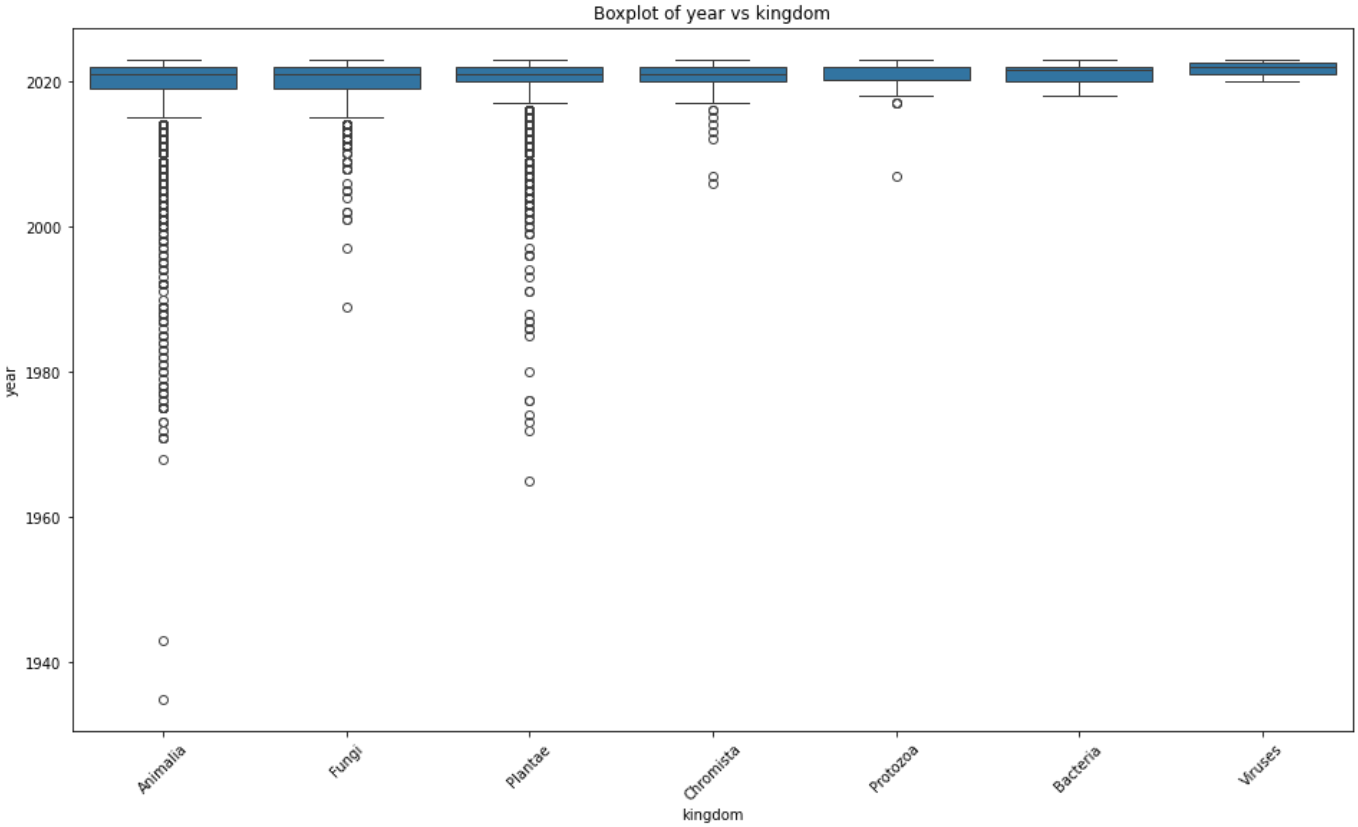
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



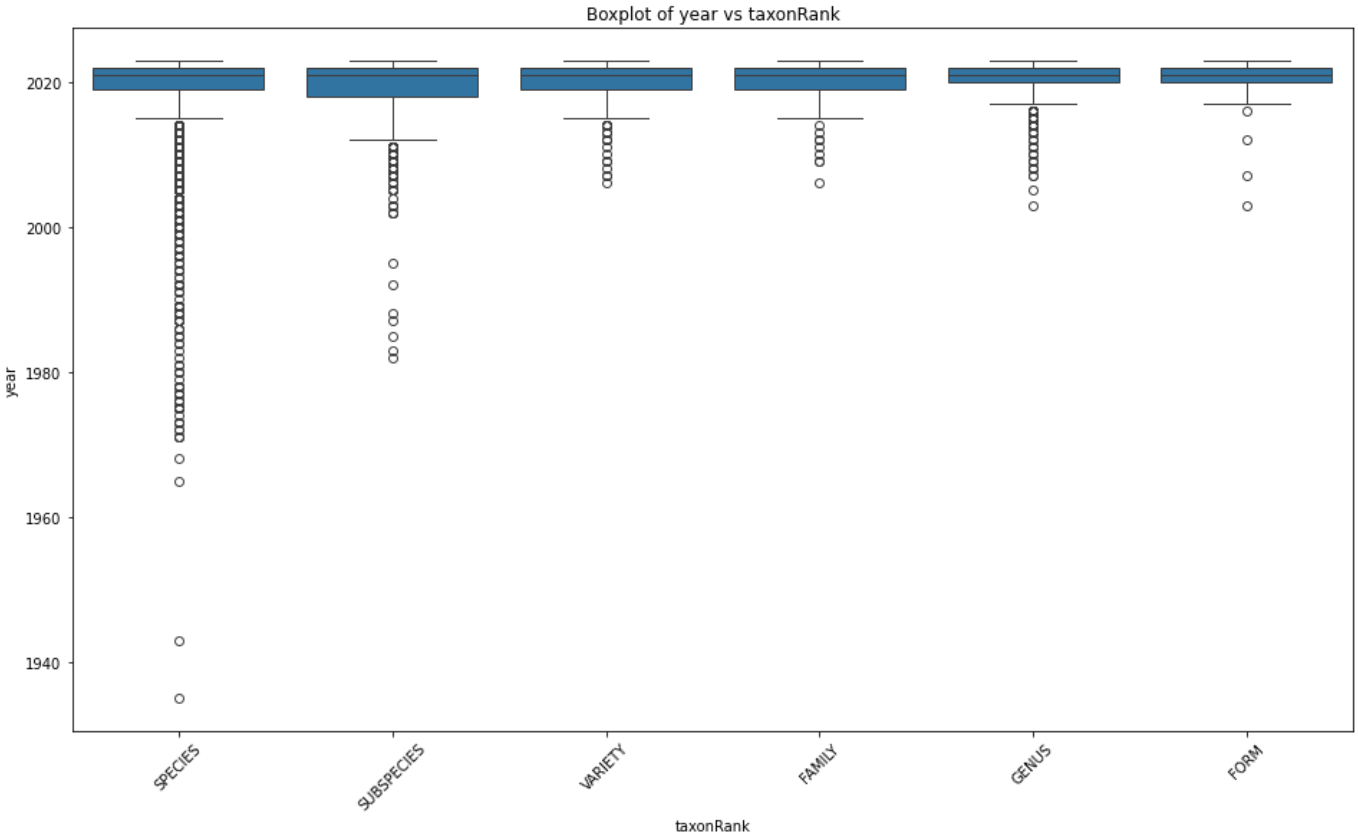


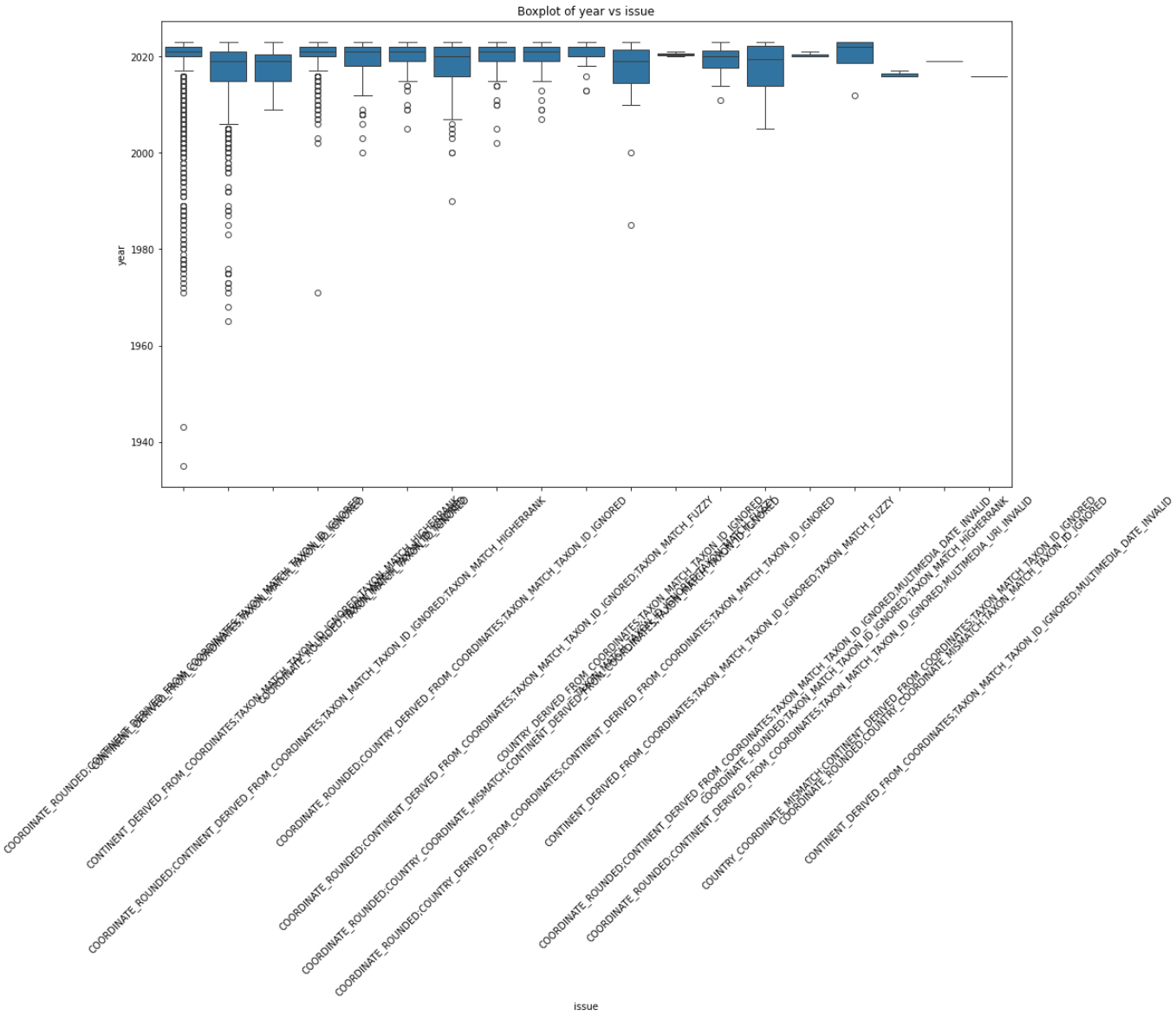
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



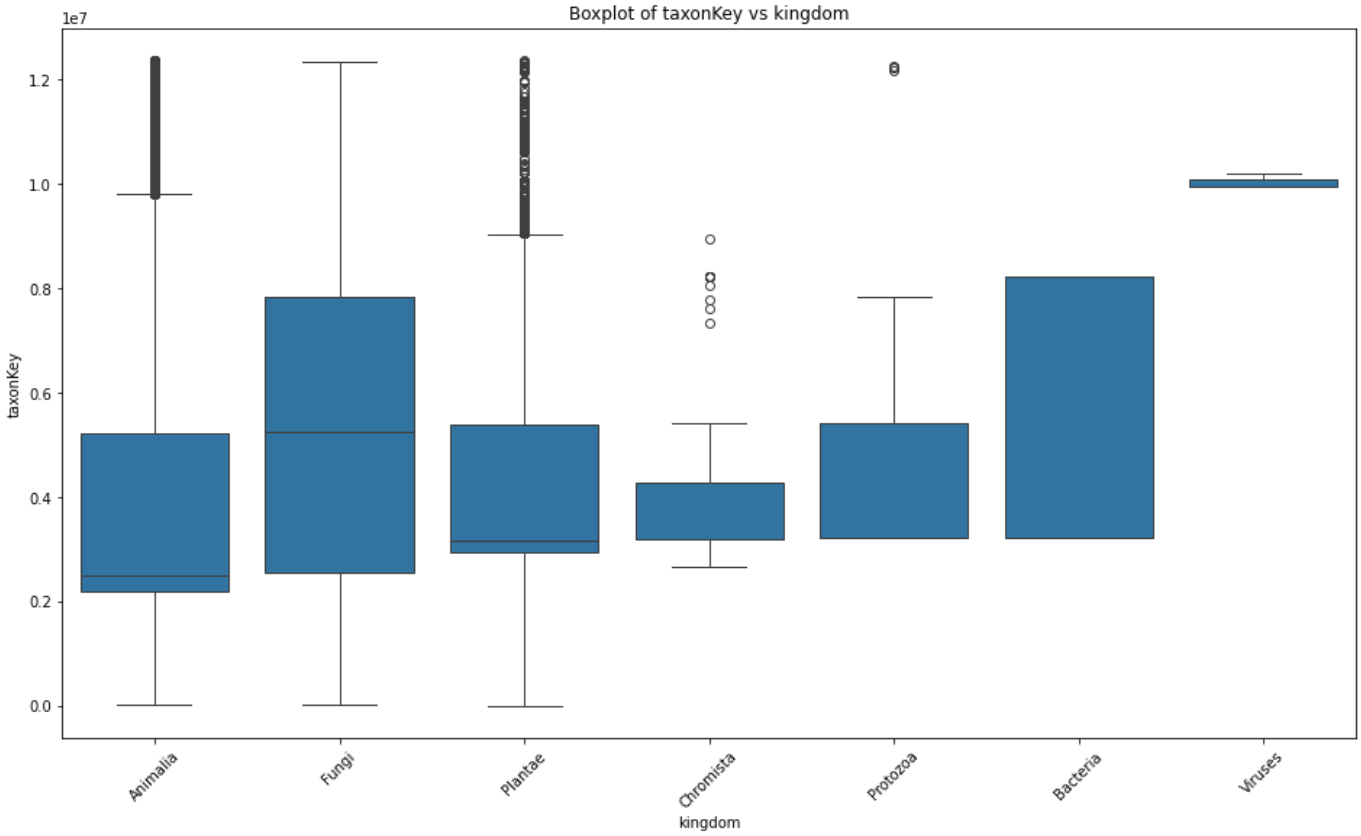


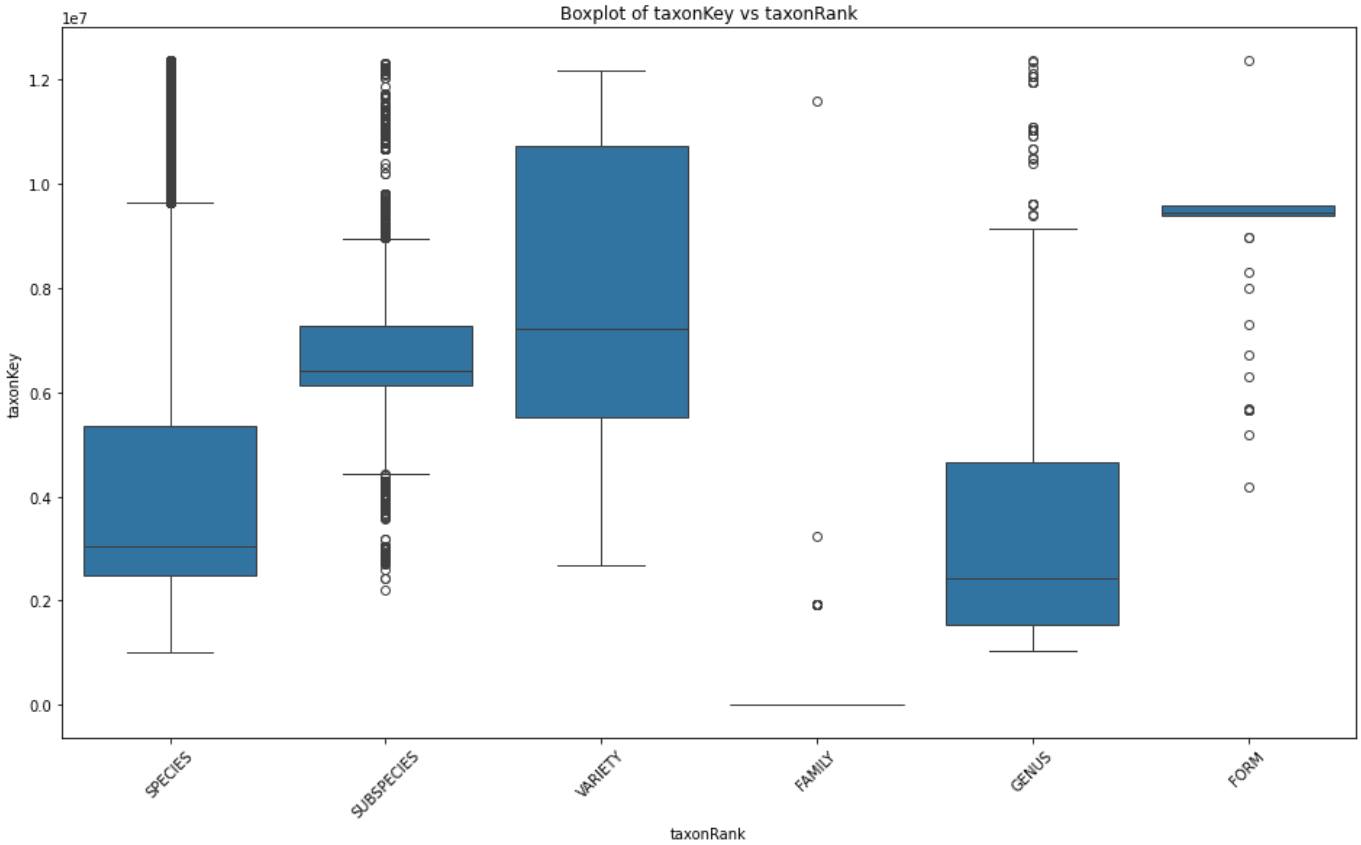
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



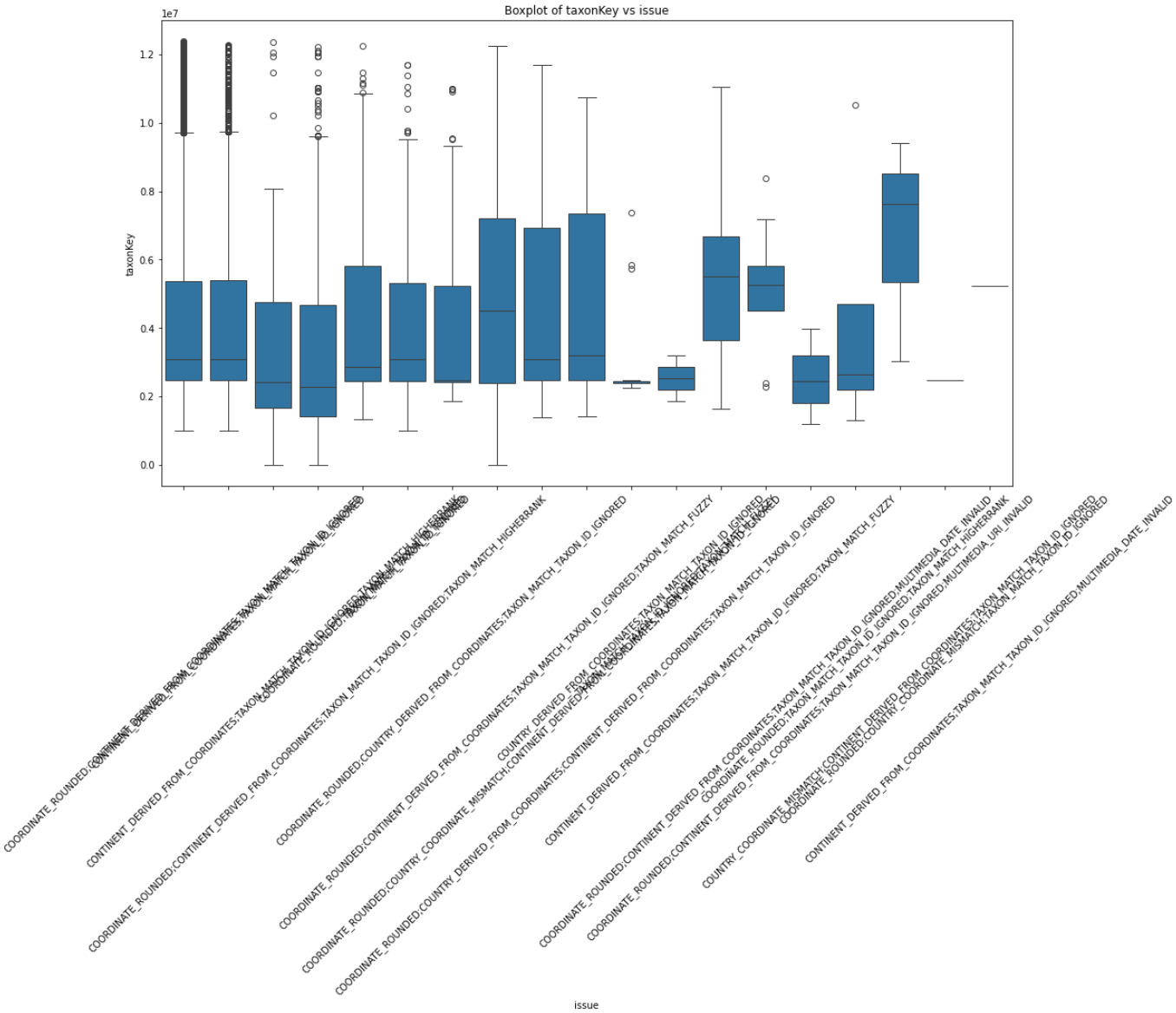


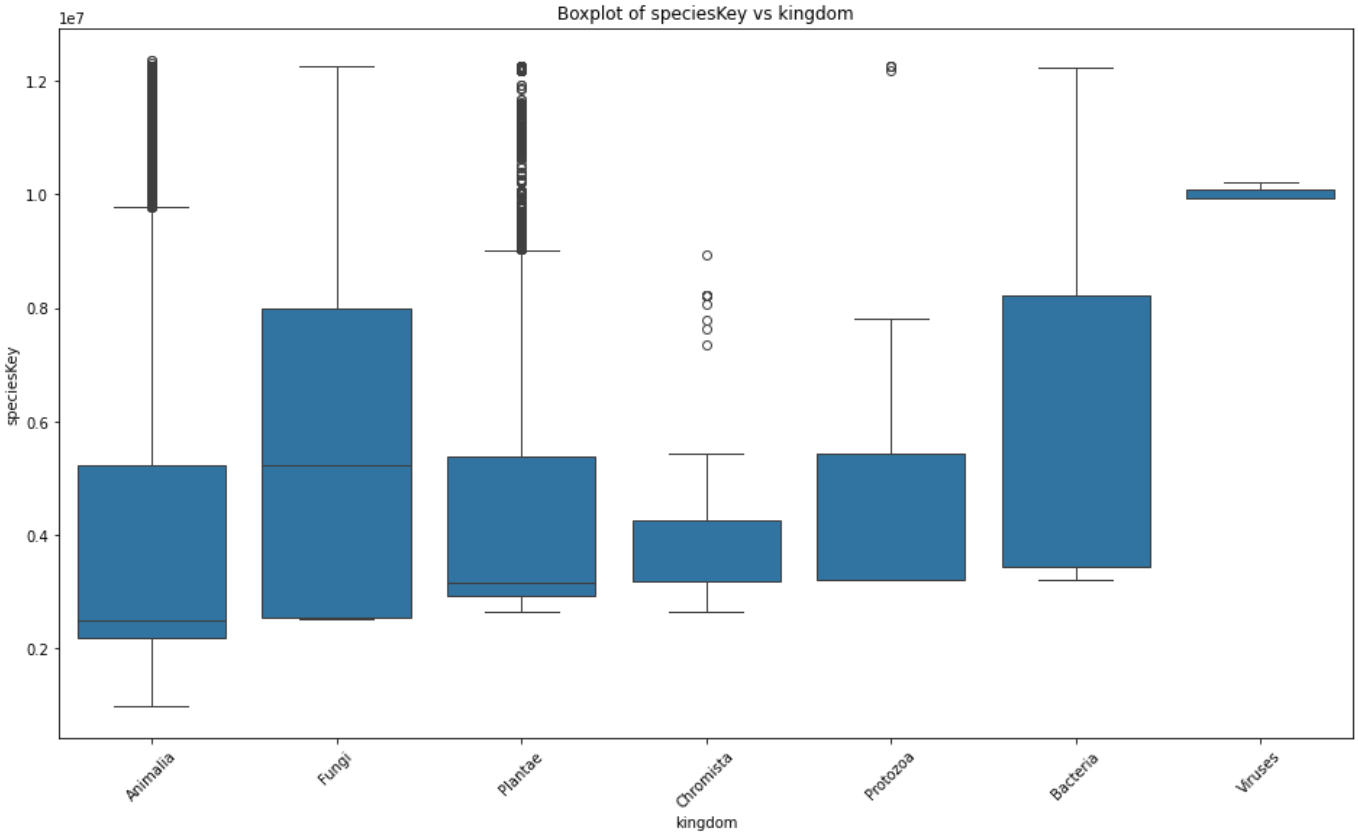
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



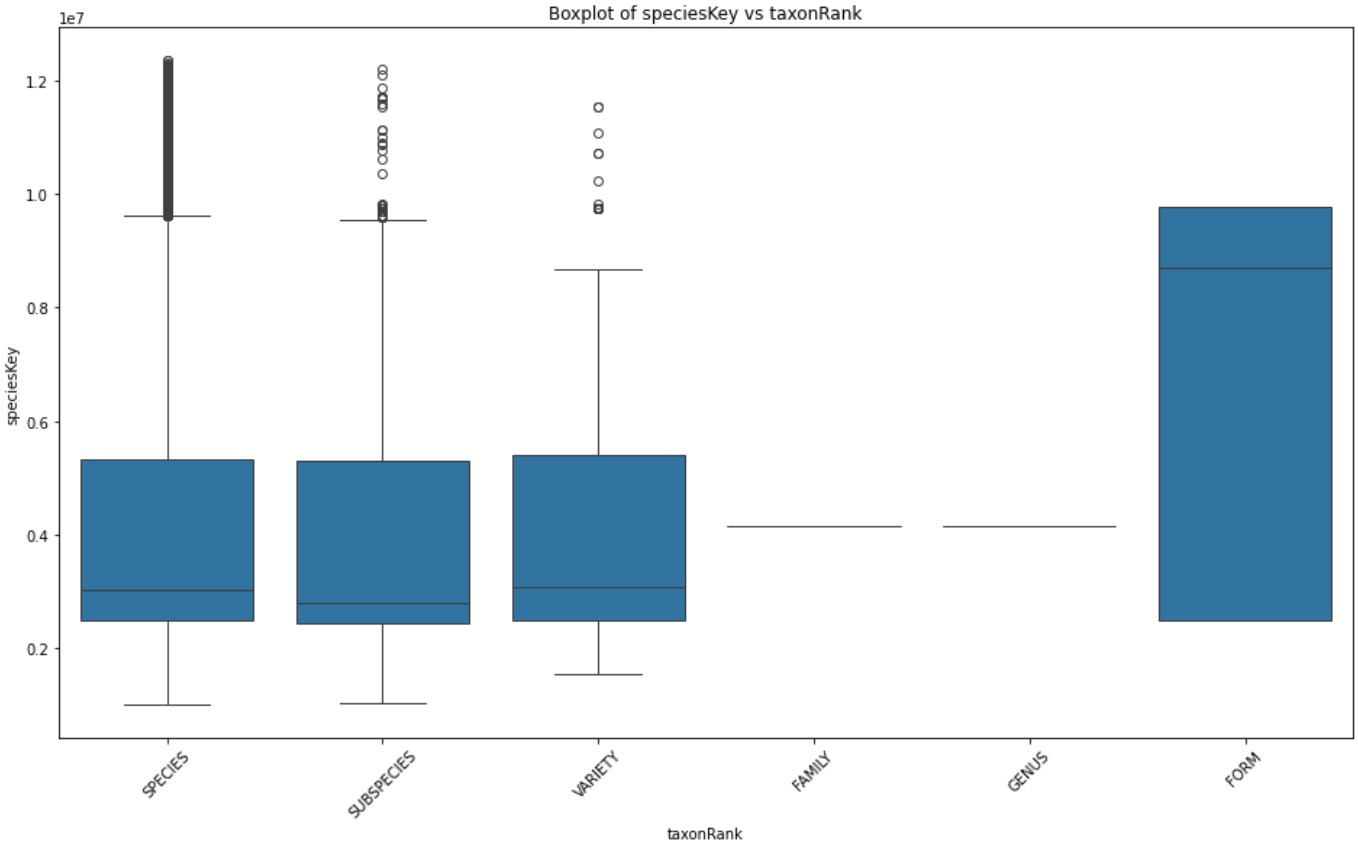


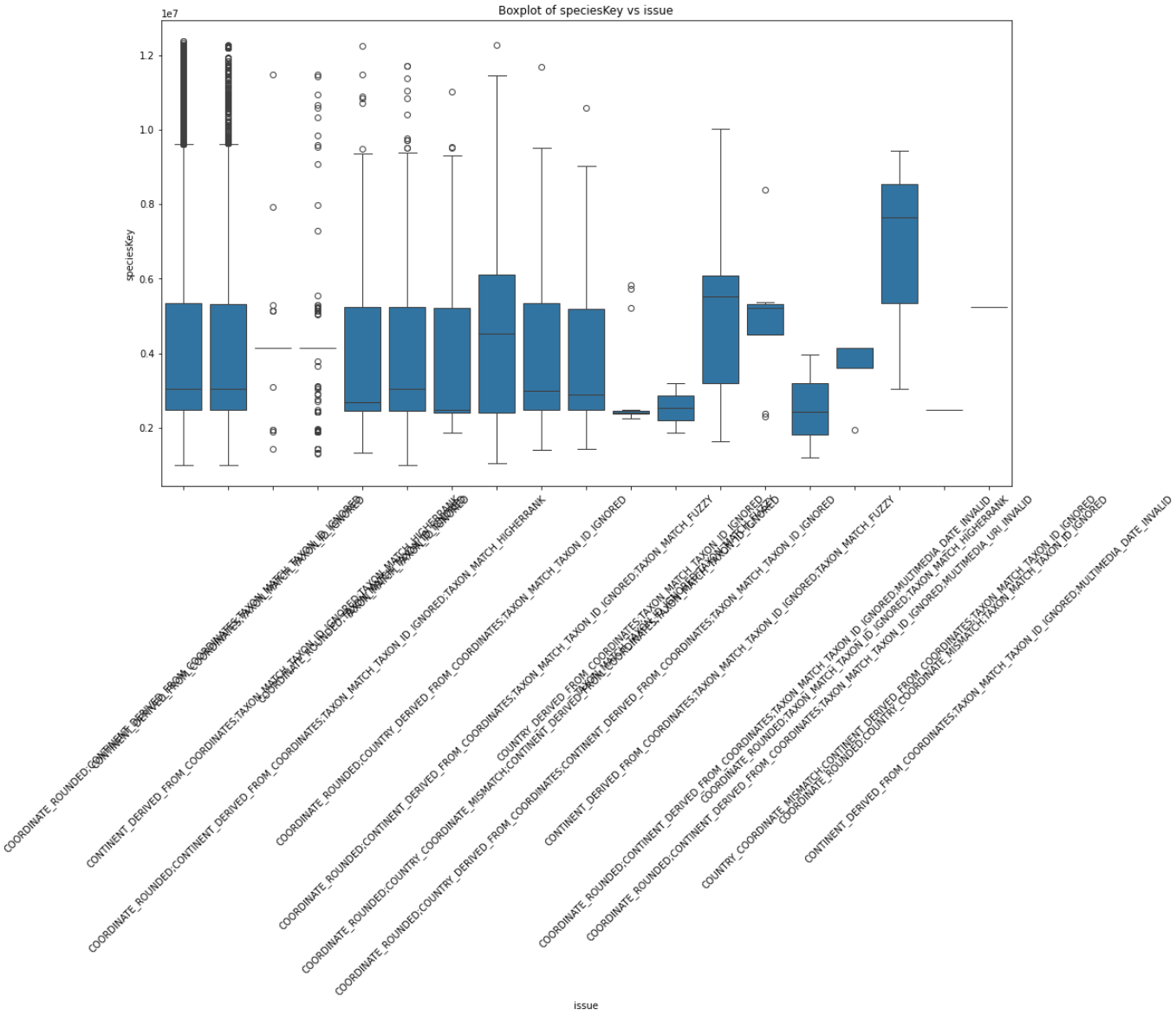
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



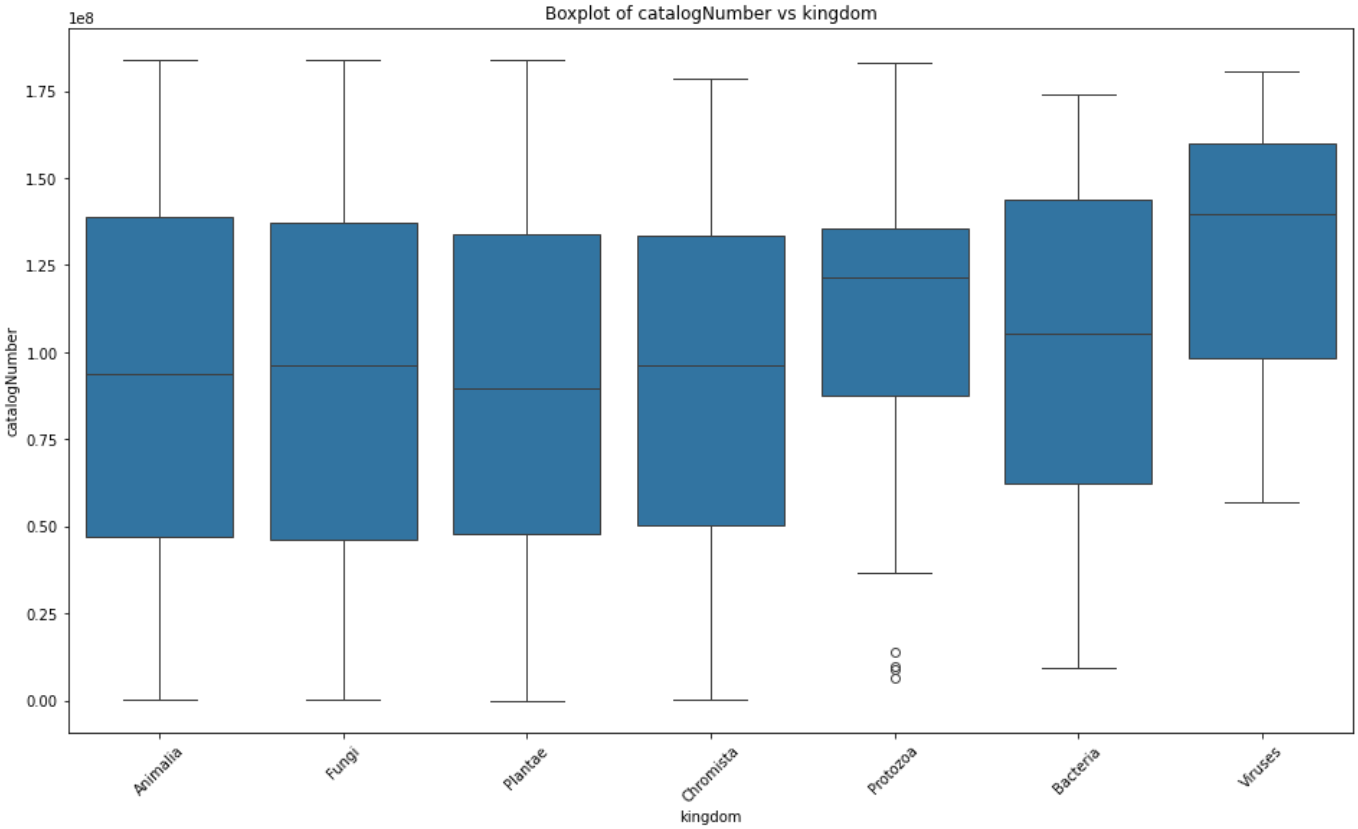


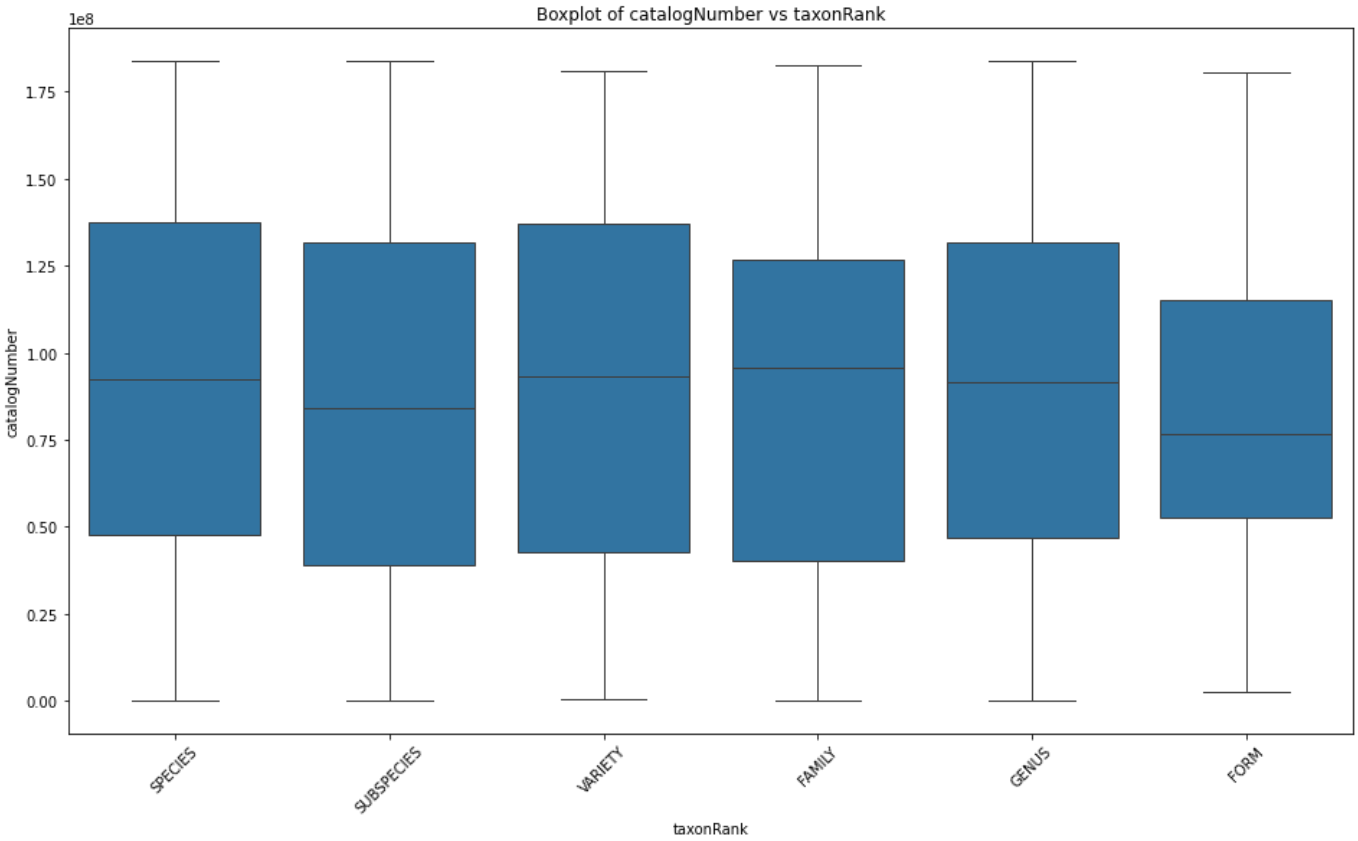
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



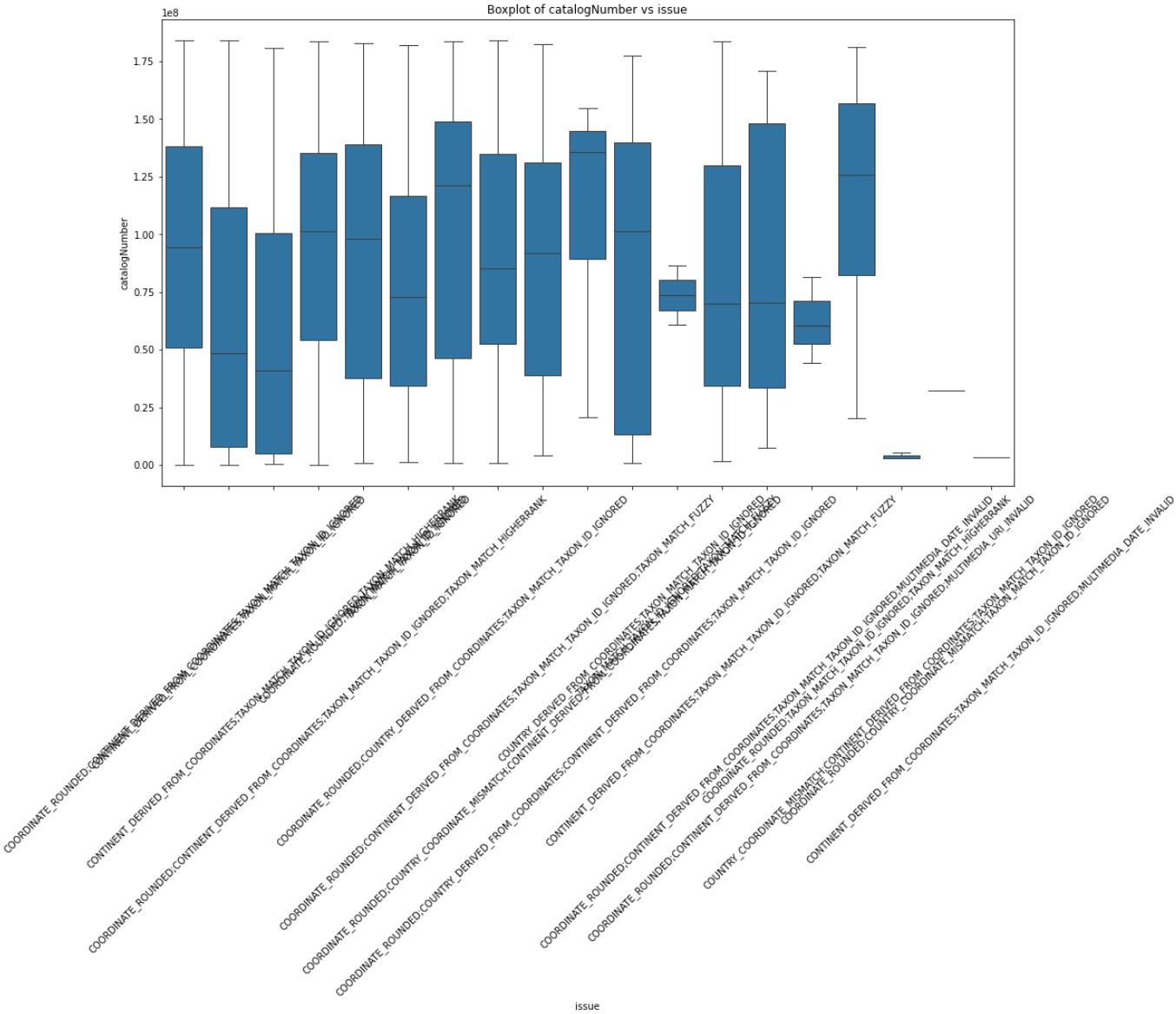


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js





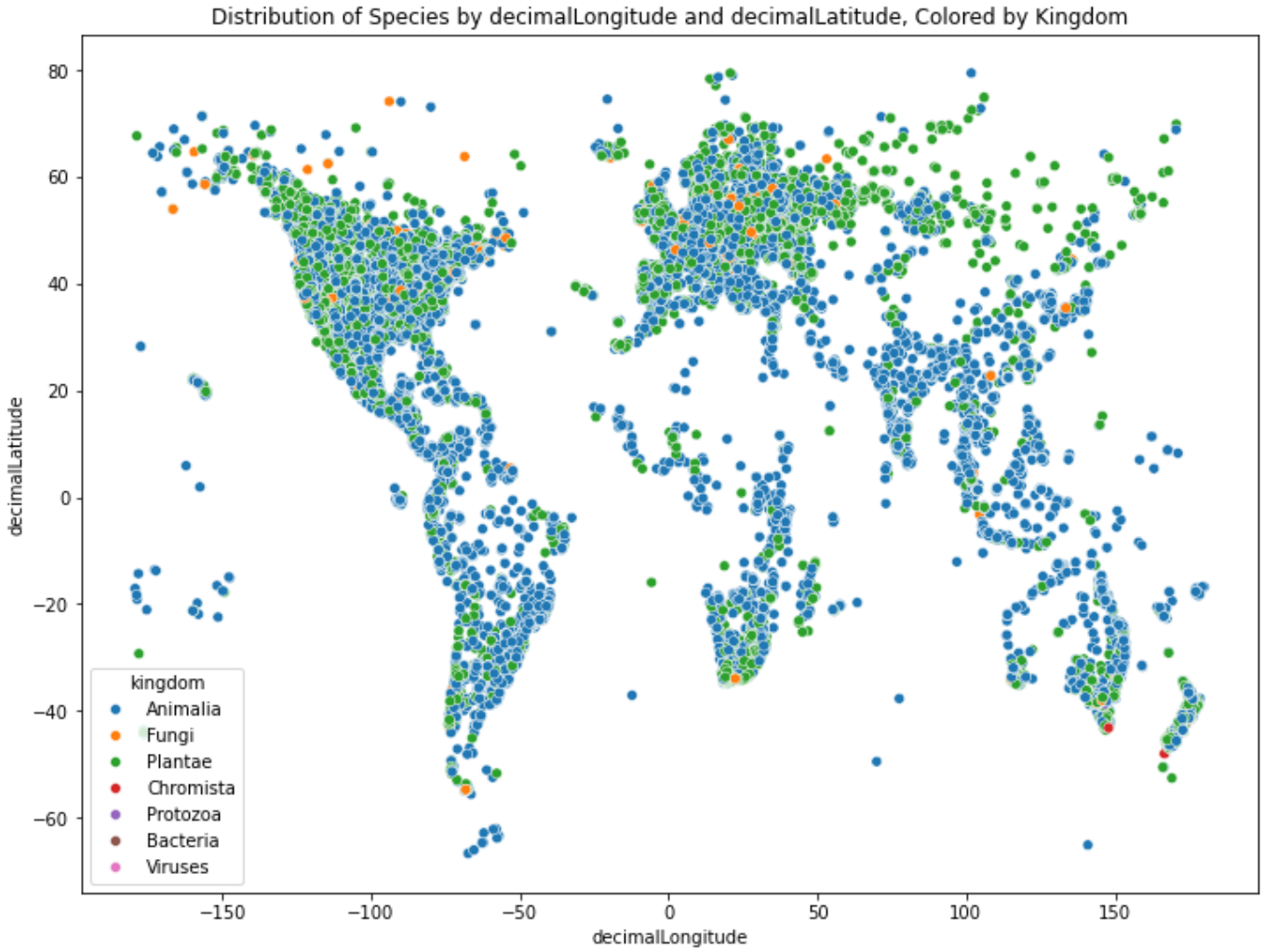
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



multivariate analysis

In [39]:
```
plt.figure(figsize=(12,9))
sns.scatterplot(data=df, x=df['decimalLongitude'], y=df['decimalLatitude'], hue=df['k
plt.title('Distribution of Species by decimalLongitude and decimalLatitude, Colored b
plt.show()
```

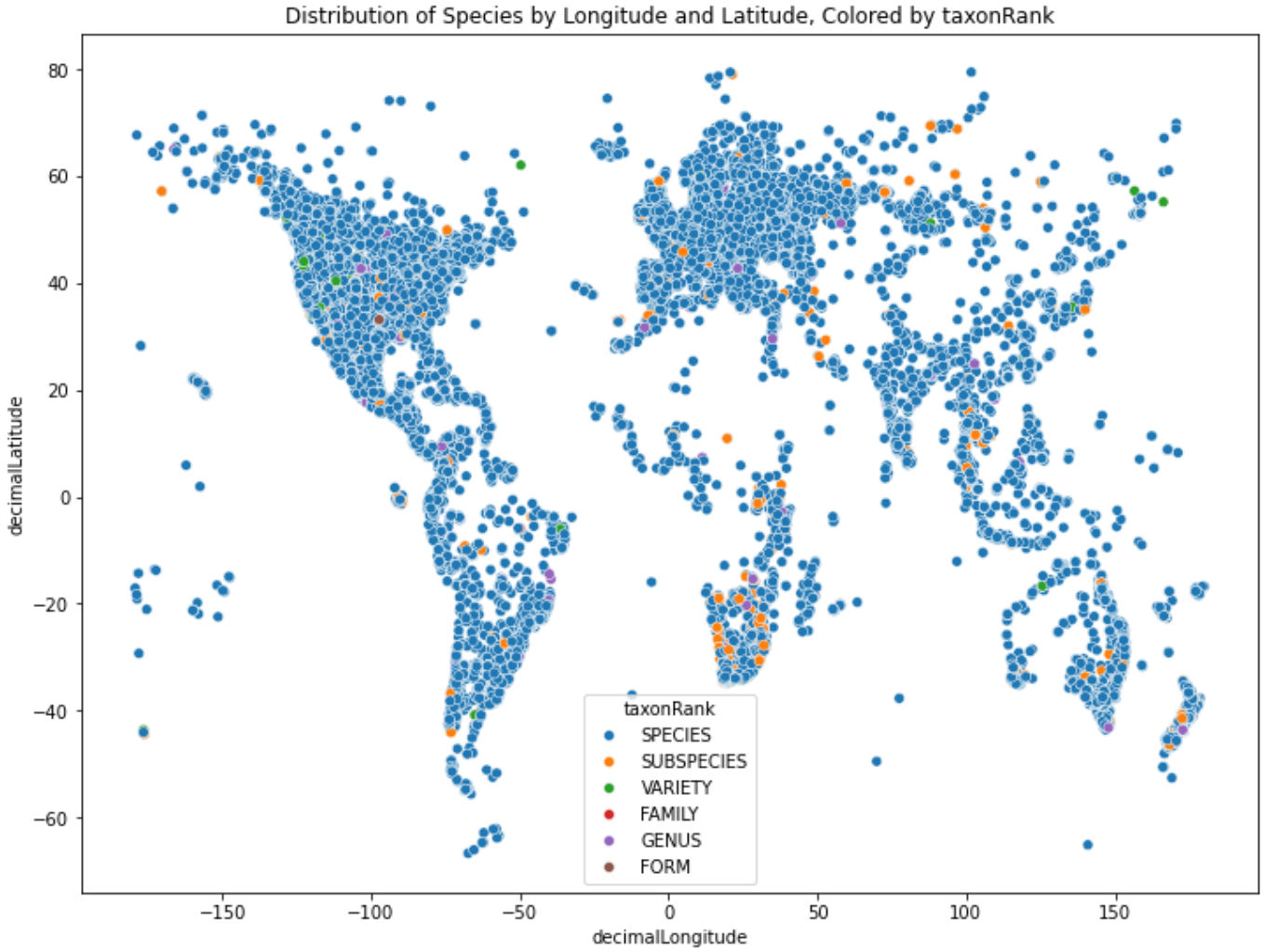
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [40]:
```
plt.figure(figsize=(12,9))
sns.scatterplot(data=df,x=df['decimalLongitude'],y=df['decimalLatitude'],hue=df['taxo
plt.title('Distribution of Species by Longitude and Latitude, Colored by taxonRank')
```

Out[40]:
```
Text(0.5, 1.0, 'Distribution of Species by Longitude and Latitude, Colored by taxonR
ank')
```

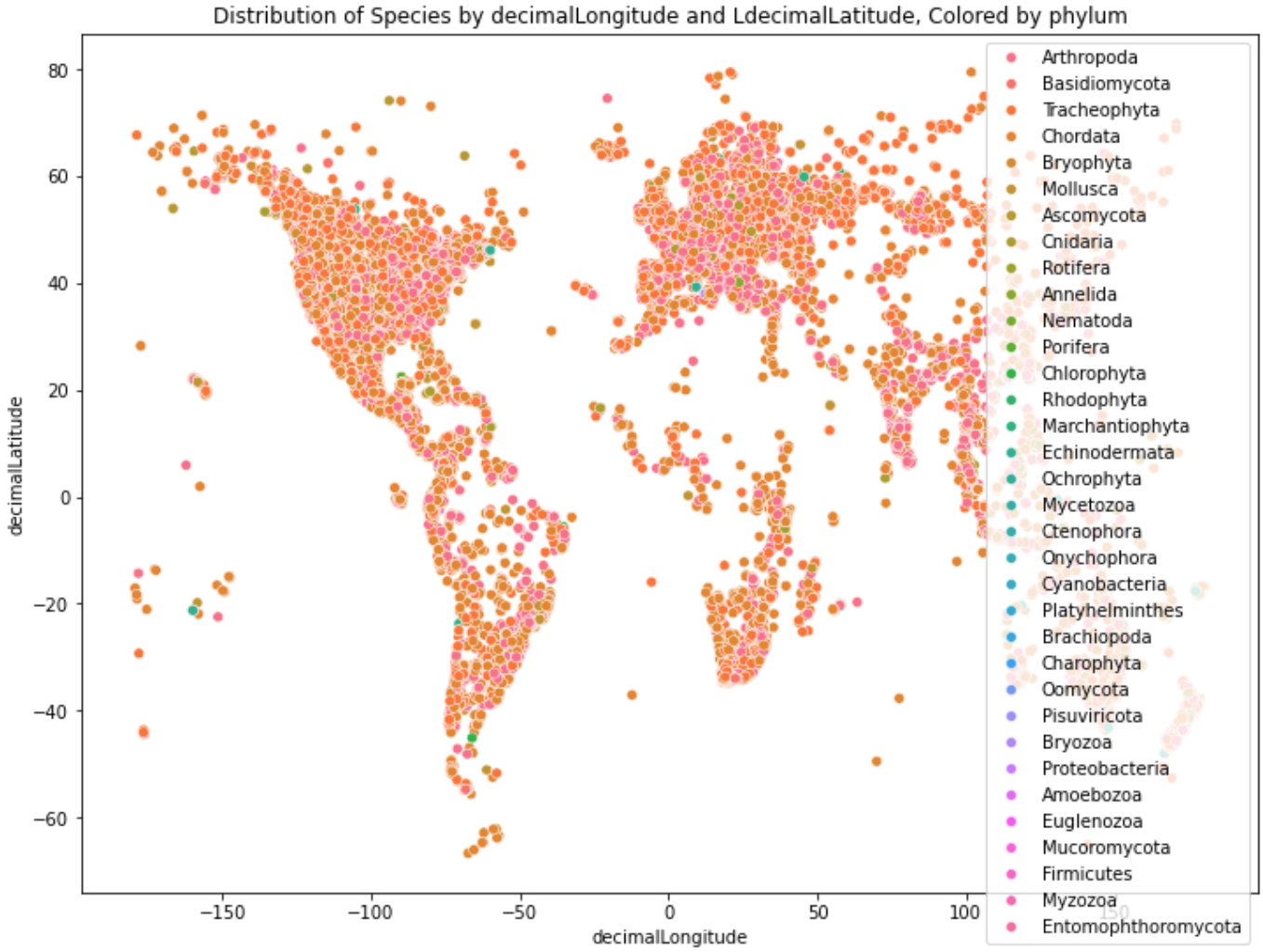
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [41]:
```
plt.figure(figsize=(12,9))
sns.scatterplot(data=df,x=df['decimalLongitude'],y=df['decimalLatitude'],hue=df['phyl
plt.title('Distribution of Species by decimalLongitude and LdecimalLatitude, Colored
plt.legend()
```

Out[41]:
```
<matplotlib.legend.Legend at 0x24a382f3eb0>
```

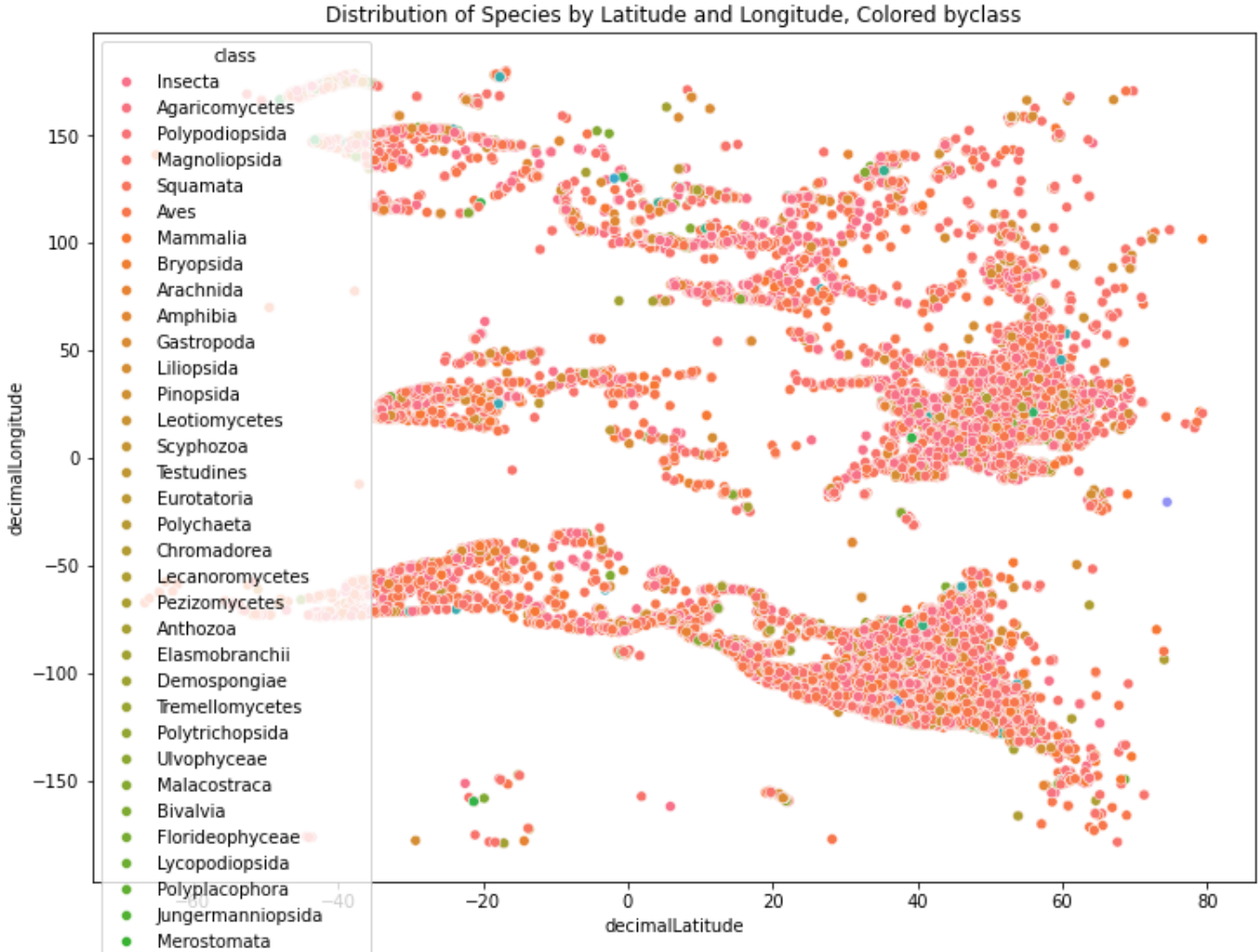
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [42]:
```
plt.figure(figsize=(12,9))
sns.scatterplot(data=df,x=df['decimalLatitude'],y=df['decimalLongitude'],hue=df['clas

plt.title('Distribution of Species by Latitude and Longitude, Colored byclass')
```

Out[42]:
```
Text(0.5, 1.0, 'Distribution of Species by Latitude and Longitude, Colored byclass')
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

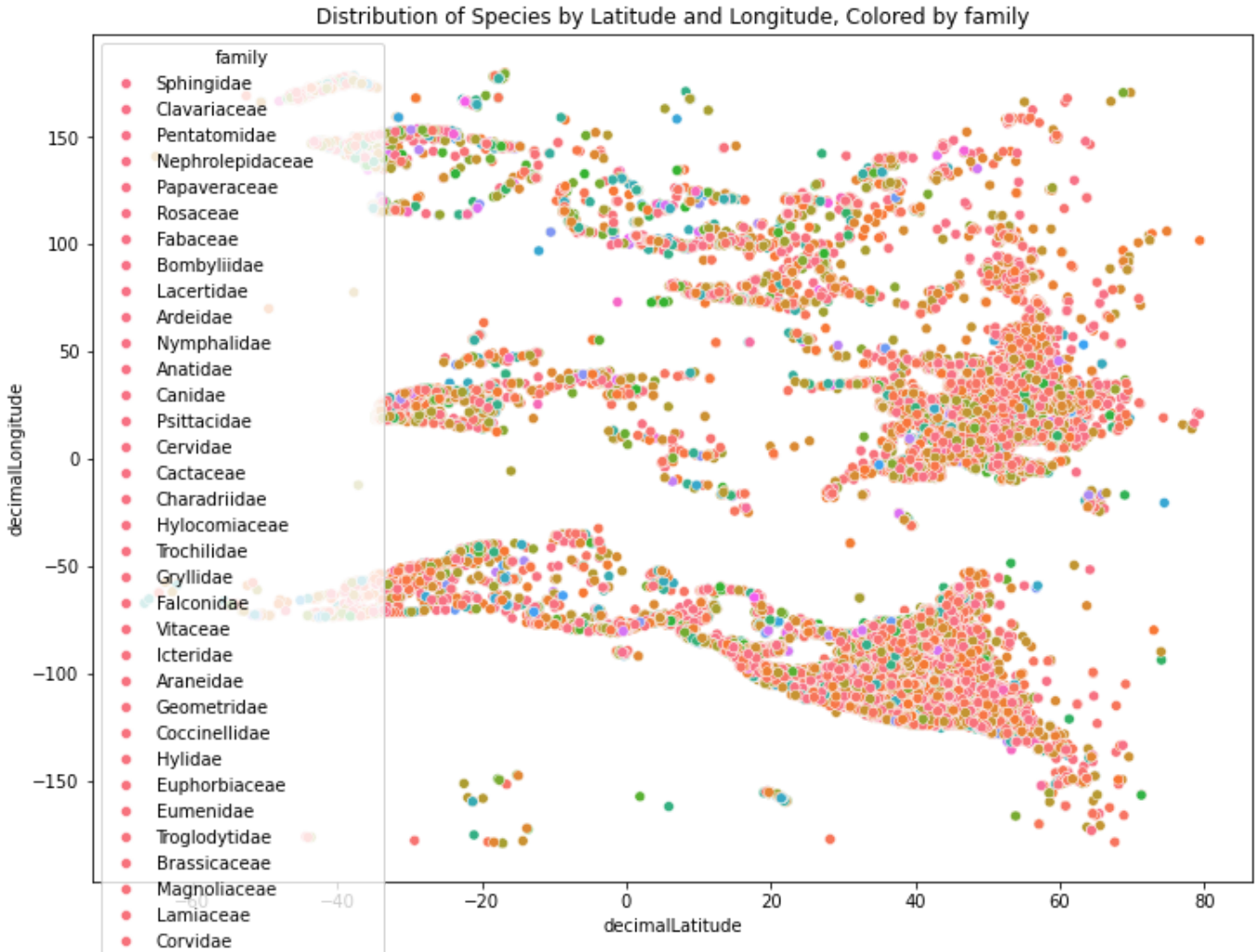


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

In [43]:
```
plt.figure(figsize=(12,9))
sns.scatterplot(data=df,x=df['decimalLatitude'],y=df['decimalLongitude'],hue=df['fami
plt.title('Distribution of Species by Latitude and Longitude, Colored by family')
```

Out[43]:
```
Text(0.5, 1.0, 'Distribution of Species by Latitude and Longitude, Colored by famil
y')
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Taxaceae
- Colchicaceae
- Mantidae
- Accipitridae
- Hymenogastraceae
- Araceae
- Asteraceae
- Pezizellaceae
- Paridae
- Mustelidae
- Tricholomataceae
- Chrysomelidae
- Sagittariidae
- Passerellidae
- Scolopacidae
- Turdidae
- Dryopteridaceae
- Viscaceae
- Ciconiidae
- Orthotrichaceae
- Pomacentridae
- Philomycidae
- Crassulaceae
- Certhiidae
- Acrochordidae
- Bromeliaceae
- Orchidaceae
- Phacellophoridae
- Solanaceae
- Lycaenidae
- Polemoniaceae
- Haematopodidae
- Anacardiaceae
- Formicidae
- Apocynaceae
- Cracticidae
- Cardinalidae
- Columbidae
- Schizophyllaceae
- Amanitaceae
- Bulimulidae
- Helicidae
- Picidae
- Violaceae
- Scarabaeidae
- Aristolochiaceae
- Sphaerodactylidae
- Muscicapidae
- Ursidae
- Percidae
- Orobanchaceae
- Colubridae
- Suillaceae
- Cicadellidae
- Emydidae
- Herpestidae
- Convolvulaceae
- Syrphidae
- Scaridae
- Caprifoliaceae
- Theridiidae
- Scincidae
- Ambystomatidae
- Tortricidae
- Tetrigidae
- Rhopalidae
- Polyporaceae
- Vespidae
- Verbenaceae
- Polypodiaceae
- Phalacrocoracidae
- Thraupidae
- Bolbitiaceae
- Philodinidae
- Carabidae
- Thalassematidae
- Parulidae
- Sciuridae
- Tyrannidae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Boraginaceae
- Lygodiaceae
- Odontophoridae
- Oleaceae
- Plantaginaceae
- Anguinidae
- Cerambycidae
- Tephritidae
- Felidae
- Sapindaceae
- Begoniaceae
- Teloschistaceae
- Otariidae
- Santalaceae
- Chamaeleonidae
- Liliaceae
- Primulaceae
- Helvellaceae
- Sarraceniaceae
- Syngnathidae
- Suidae
- Scrophulariaceae
- Lestidae
- Cornaceae
- Caryophyllaceae
- Salicaceae
- Humiriaceae
- Eriophyidae
- Lythraceae
- Labridae
- Pieridae
- Cuculidae
- Cyprinidae
- Nemanthidae
- Geraniaceae
- Plakobranchidae
- Pterophoridae
- Agaricaceae
- Reduviidae
- Sittidae
- Salviniaceae
- Moraceae
- Gekkonidae
- Proteaceae
- Bombycillidae
- Urolophidae
- Chalinidae
- Tremellaceae
- Cathartidae
- Passeridae
- Pinaceae
- Polytrichaceae
- Linyphiidae
- Thymelaeaceae
- Ulvaceae
- Parmeliaceae
- Juncaceae
- Gentianaceae
- Palaemonidae
- Malaconotidae
- Stomolophidae
- Dironidae
- Gruidae
- Rubiaceae
- Mytilidae
- Leiocephalidae
- Cynipidae
- Curculionidae
- Coreidae
- Phasianidae
- Viburnaceae
- Tenebrionidae
- Acrocephalidae
- Dactyloidae
- Motacillidae
- Pandionidae
- Siphonariidae
- Passifloraceae
- Coenagrionidae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Gomphaceae
- Pyralidae
- Ulidiidae
- Acrididae
- Myrtaceae
- Elaeocarpaceae
- Saturniidae
- Podicipedidae
- Lycopodiaceae
- Iguanidae
- Dioscoreaceae
- Samlidae
- Onagraceae
- Arecaceae
- Berberidaceae
- Malvaceae
- Laniidae
- Dermestidae
- Threskiornithidae
- Hirundinidae
- Ischnochitonidae
- Polyceridae
- Pelliaceae
- Anguidae
- Gaviidae
- Sclerosomatidae
- Phocidae
- Lecythidaceae
- Anhingidae
- Epialtidae
- Matutidae
- Pandoridae
- Bovidae
- Mimidae
- Romaleidae
- Mutillidae
- Uloboridae
- Mycenaceae
- Bignoniaceae
- Amaranthaceae
- Tachinidae
- Capparaceae
- Betulaceae
- Urticaceae
- Lyonetiidae
- Cupressaceae
- Commelinaceae
- Cettiidae
- Heliconiaceae
- Centrarchidae
- Numididae
- Dasypodidae
- Phalangiidae
- Unionidae
- Phascolarctidae
- Pelecanidae
- Oedemeridae
- Cacatuidae
- Saxifragaceae
- Sylviidae
- Galbulidae
- Andrenidae
- Crabronidae
- Ptilogonatidae
- Acanthaceae
- Margarodidae
- Dasyatidae
- Procellariidae
- Turbinidae
- Asphodelaceae
- Anacampserotaceae
- Strigidae
- Salamandridae
- Achatinidae
- Cladoniaceae
- Cyperaceae
- Salticidae
- Hydrobatidae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Polygonaceae
- Testudinidae
- Onocleaceae
- Lucanidae
- Rhytismataceae
- Merulinidae
- Furnariidae
- Pomacanthidae
- Aquifoliaceae
- Heliozelidae
- Recurvirostridae
- Pholcidae
- Elateridae
- Plethodontidae
- Procyonidae
- Ophioglossaceae
- Amylocorticiaceae
- Loganiaceae
- Celastraceae
- Petauridae
- Mullidae
- Cosmetidae
- Melanthiaceae
- Irciniidae
- Corduliidae
- Sapotaceae
- Porcellanidae
- Auriculariaceae
- Namaceae
- Lutjanidae
- Loasaceae
- Astacidae
- Asilidae
- Lentibulariaceae
- Phrynosomatidae
- Diprionidae
- Laetiporaceae
- Limulidae
- Fouquieriaceae
- Cicadidae
- Vireonidae
- Polygalaceae
- Aegithalidae
- Buprestidae
- Porcellionidae
- Alcedinidae
- Malpighiaceae
- Cardiidae
- Menyanthaceae
- Montiaceae
- Anisopodidae
- Carangidae
- Alcidae
- Microhylidae
- Alaudidae
- Cracidae
- Butomaceae
- Amaryllidaceae
- Halictidae
- Ranidae
- Calopterygidae
- Trochanteriidae
- Aeshnidae
- Cistaceae
- Melittidae
- Blechnaceae
- Oxalidaceae
- Asterinidae
- Elephantidae
- Viperidae
- Xanthonychidae
- Sclerodermataceae
- Iridaceae
- Maluridae
- Zygophyllaceae
- Poeciliidae
- Estrildidae
- Fucaceae
- Meropidae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Saururaceae
- Endocladiaceae
- Sphecidae
- Castoridae
- Cecidomyiidae
- Armadillidiidae
- Ploceidae
- Stratiomyidae
- Ceratophyllaceae
- Omphalotaceae
- Hericiaceae
- Emberizidae
- Fomitopsidaceae
- Centrolenidae
- Candelariaceae
- Grossulariaceae
- Macropodidae
- Lasiocampidae
- Asteliaceae
- Miridae
- Calliphoridae
- Sarcomyxaceae
- Gomphidae
- Talpidae
- Hypericaceae
- Cantharidae
- Simaroubaceae
- Plumbaginaceae
- Aizoaceae
- Upupidae
- Scombridae
- Tachyglossidae
- Cercopithecidae
- Nyctaginaceae
- Lobariaceae
- Megachilidae
- Melastomataceae
- Notodontidae
- Atelidae
- Chromodorididae
- Pectinidae
- Elapidae
- Phytolaccaceae
- Dilleniaceae
- Aplysiidae
- Carcharhinidae
- Platanaceae
- Myocastoridae
- Delphinidae
- Pteropodidae
- Curtisiaceae
- Grapsidae
- Podargidae
- Stichopodidae
- Emballonuridae
- Sesarmidae
- Alismataceae
- Gracillariidae
- Agelenidae
- Limnephilidae
- Tenthredinidae
- Giraffidae
- Pycnonotidae
- Synodontidae
- Rallidae
- Hemieuryalidae
- Equidae
- Splachnidiaceae
- Dictyotaceae
- Paradoxosomatidae
- Gorgonocephalidae
- Juglandaceae
- Ramphastidae
- Hydnaceae
- Cisticolidae
- Oecophoridae
- Regulidae
- Petaluridae
- Lepadidae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Campephagidae
- Aspleniaceae
- Pyrochroidae
- Corydalidae
- Fasciolariidae
- Pontederiaceae
- Passalidae
- Sciaenidae
- Linaceae
- Gloeophyllaceae
- Griseliniaceae
- Rhipiduridae
- Acanthizidae
- Cycloramphidae
- Marattiaceae
- Enidae
- Macromiidae
- Sesiidae
- Marasmiaceae
- Thelypteridaceae
- Lembophyllaceae
- Potamogetonaceae
- Diapensiaceae
- Grifolaceae
- Melongenidae
- Mimallonidae
- Pisaniidae
- Sparidae
- Hydrophyllaceae
- Menispermaceae
- Droseraceae
- Posidoniaceae
- Hamamelidaceae
- Staphylinidae
- Acanthochitonidae
- Cerithiidae
- Acanthosomatidae
- Lycosidae
- Uranoscopidae
- Gymnophthalmidae
- Schisandraceae
- Erysiphaceae
- Hypoxylaceae
- Paulowniaceae
- Parnassiaceae
- Uraniidae
- Scutelleridae
- Dendrodorididae
- Vaejovidae
- Cypraeidae
- Lomentariaceae
- Eremobatidae
- Cheilodactylidae
- Cleridae
- Zingiberaceae
- Fistulinaceae
- Megalaimidae
- Scapaniaceae
- Pteromalidae
- Mniaceae
- Phrymaceae
- Balsaminaceae
- Dolichopodidae
- Timaliidae
- Dytiscidae
- Aequoreidae
- Scirtidae
- Lycidae
- Muscidae
- Phyllodactylidae
- Scutigeridae
- Gyrinidae
- Trogonidae
- Bombinatoridae
- Mactridae
- Pteridaceae
- Erinaceidae
- Lycoperdaceae
- Pyrgotidae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Loranthaceae
- Zosteraceae
- Scoliidae
- Agamidae
- Salmonidae
- Frullaniaceae
- Pycnoporellaceae
- Corytophanidae
- Pyxicephalidae
- Cordiaceae
- Dicaeidae
- Agromyzidae
- Cotingidae
- Tritoniidae
- Pipidae
- Ulmaceae
- Ginglymostomatidae
- Pleurotaceae
- Peltigeraceae
- Hemiphractidae
- Bruniaceae
- Rhaphidophoridae
- Flabellinidae
- Smilacaceae
- Aphididae
- Dryophthoridae
- Aphalaridae
- Cricetidae
- Carcinidae
- Eriocottidae
- Clinidae
- Hypnaceae
- Hygromiidae
- Veronicellidae
- Chelydridae
- Ptychadenidae
- Calcariidae
- Pisauridae
- Discinaceae
- Goniodorididae
- Dairidae
- Sulidae
- Depressariidae
- Pezizaceae
- Zygaenidae
- Bondarzewiaceae
- Heliotropiaceae
- Triglidae
- Ustilaginaceae
- Psathyrellaceae
- Phylloscopidae
- Fissurellidae
- Chaetodontidae
- Bothriuridae
- Meliphagidae
- Ramalinaceae
- Aramidae
- Cichlidae
- Phyllidiidae
- Lybiidae
- Phallaceae
- Onchidorididae
- Physalacriaceae
- Tetragnathidae
- Lennoaceae
- Leptoceridae
- Physaraceae
- Attelabidae
- Tropiduridae
- Aplodontiidae
- Rhyparochromidae
- Olividae
- Pyrrhocoridae
- Megapodiidae
- Cortinariaceae
- Patellidae
- Gasterosteidae
- Metteniusaceae
- Chrysopidae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Idoteidae
- Nartheciaceae
- Irpicaceae
- Pseudocheiridae
- Ischnodermataceae
- Haliotidae
- Lecanoraceae
- Pythonidae
- Dendronotidae
- Erethizontidae
- Crepidotaceae
- Pachybolidae
- Latridae
- Gleicheniaceae
- Mazaceae
- Siricidae
- Haemulidae
- Sparassidae
- Lonchodidae
- Annonaceae
- Balanophoraceae
- Gorgoniidae
- Megalopygidae
- Hypochilidae
- Membracidae
- Fulgoridae
- Mitridae
- Fregatidae
- Kerriidae
- Vespertilionidae
- Gesneriaceae
- Strophariaceae
- Portunidae
- Ostraciidae
- Phalangeridae
- Tetraodontidae
- Myliobatidae
- Cleomaceae
- Limacidae
- Plutellidae
- Platystomatidae
- Burseraceae
- Catostomidae
- Haplotrematidae
- Pelodryadidae
- Meloidae
- Bryconidae
- Sagartiidae
- Balitoridae
- Parastacidae
- Caviidae
- Apodidae
- Euphaeidae
- Adelidae
- Serranidae
- Pseudophasmatidae
- Momotidae
- Calostomataceae
- Oriolidae
- Trichoniscidae
- Petroicidae
- Burhinidae
- Rutaceae
- Balanidae
- Hypodontiaceae
- Aphrophoridae
- Cheloniidae
- Acroporidae
- Varanidae
- Pardalotidae
- Craugastoridae
- Peramelidae
- Pyrgomorphidae
- Mephitidae
- Meesiaceae
- Pachycephalidae
- Myricaceae
- Gerridae
- Musaceae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Petromyzontidae
- Russulaceae
- Muricidae
- Ammodytidae
- Dicksoniaceae
- Nolidae
- Arcidae
- Volutidae
- Bittacidae
- Tupaiidae
- Milleporidae
- Sabellidae
- Podocarpaceae
- Gryllotalpidae
- Clethraceae
- Synthemistidae
- Kyphosidae
- Krameriaceae
- Hydnangiaceae
- Euteliidae
- Trachelipodidae
- Balaenopteridae
- Ebenaceae
- Ostreidae
- Theaceae
- Anostostomatidae
- Ficidae
- Drepanidae
- Hydrocharitaceae
- Flatidae
- Melyridae
- Asterolecaniidae
- Oxyopidae
- Pompilidae
- Gecarcinucidae
- Chelidae
- Rajidae
- Goodeniaceae
- Plataspidae
- Nanodeaceae
- Jacanidae
- Trinchesiidae
- Therevidae
- Arcyriaceae
- Diplocystidiaceae
- Lottiidae
- Caudinidae
- Icmadophilaceae
- Octopodidae
- Leiothrichidae
- Coryphellidae
- Pleurobrachiidae
- Teiidae
- Podocnemididae
- Hydrangeaceae
- Gryllacrididae
- Balaenidae
- Botryosphaeriaceae
- Typhaceae
- Cranaidae
- Culicidae
- Phoenicopteridae
- Actiniidae
- Drilliidae
- Osmundaceae
- Tingidae
- Buthidae
- Orectolobidae
- Penaeaceae
- Limoniidae
- Typhlopidae
- Locustellidae
- Simmondsiaceae
- Riodinidae
- Facelinidae
- Veneridae
- Ovalipidae
- Phlyctidaceae
- Julidae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Micropezidae
- Oxycarenidae
- Marantaceae
- Cossidae
- Ocypodidae
- Helodermatidae
- Forficulidae
- Strongylocentrotidae
- Rhodomelaceae
- Cosmopterigidae
- Portulacaceae
- Cymodoceaceae
- Chrysididae
- Lophocoleaceae
- Tetraclitidae
- Euscorpiidae
- Rhizophoraceae
- Hygrophoropsidaceae
- Combretaceae
- Amphinomidae
- Viduidae
- Arthroleptidae
- Lithobiidae
- Didelphidae
- Blaberidae
- Alucitidae
- Pseudoclitocybaceae
- Pottiaceae
- Stereaceae
- Ophichthidae
- Yponomeutidae
- Davalliaceae
- Chthamalidae
- Acanthuridae
- Bradypodidae
- Lunulariaceae
- Fundulidae
- Pluteaceae
- Theraphosidae
- Cheirogaleidae
- Hadruridae
- Physaliidae
- Hyperoliidae
- Tineidae
- Crotaphytidae
- Mogoplistidae
- Zopheridae
- Platycnemididae
- Phyllophoridae
- Leiosauridae
- Glomeridae
- Umbridae
- Conopidae
- Glareolidae
- Holocentridae
- Turneraceae
- Cuniculidae
- Cervantesiaceae
- Resedaceae
- Lepisosteidae
- Mantispidae
- Gobiidae
- Naucoridae
- Tetrachondraceae
- Lepismatidae
- Monarchidae
- Triviidae
- Limacinidae
- Haminoeidae
- Physciaceae
- Coenobitidae
- Ixodidae
- Lardizabalaceae
- Toxopneustidae
- Arthoniaceae
- Argidae
- Ectobiidae
- Tayassuidae
- Neckeraceae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Pucciniastraceae
- Brachyceridae
- Ripiphoridae
- Pandanaceae
- Baetidae
- Chactidae
- Alseuosmiaceae
- Discodorididae
- Geotrupidae
- Proteidae
- Dipterocarpaceae
- Hippidae
- Bryaceae
- Dicranaceae
- Ephedraceae
- Scathophagidae
- Stilbaceae
- Lauriidae
- Alydidae
- Promeropidae
- Aytoniaceae
- Bartheletiaceae
- Urotrygonidae
- Dendrobatidae
- Cabombaceae
- Tabanidae
- Scorpionidae
- Diapheromeridae
- Pandalidae
- Umbilicariaceae
- Meliaceae
- Cottidae
- Cystopteridaceae
- Rhipiceridae
- Buphagidae
- Rhacophoridae
- Psyllidae
- Hyaenidae
- Pittosporaceae
- Munididae
- Eresidae
- Peripatidae
- Pittidae
- Phylloxeridae
- Erotylidae
- Littorinidae
- Corynidae
- Cassidae
- Graphostromataceae
- Hydrophilidae
- Blenniidae
- Charopidae
- Dictynidae
- Ariidae
- Thoridae
- Synlestidae
- Leucobryaceae
- Hepialidae
- Mordellidae
- Cydnidae
- Molossidae
- Asteriidae
- Sphyraenidae
- Cinclidae
- Meripilaceae
- Eumastacidae
- Procaviidae
- Margaritiferidae
- Achariaceae
- Cirrhitidae
- Xyridaceae
- Plagiotheciaceae
- Ruppiaceae
- Tytonidae
- Lessoniaceae
- Zosteropidae
- Entomobryidae
- Gobiesocidae
- Ebretiaceae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Escalloniaceae
- Psammobiidae
- Hormuridae
- Arionidae
- Tripterygiidae
- Myrmecophagidae
- Columbellidae
- Neanuridae
- Hydroleaceae
- Pertusariaceae
- Chaoboridae
- Clavariadelphaceae
- Auriscalpiaceae
- Alligatoridae
- Ricciaceae
- Muraenidae
- Tessaratomidae
- Antennariidae
- Sicariidae
- Xantusiidae
- Echinometridae
- Austrosiphonidae
- Tateidae
- Grallariidae
- Restionaceae
- Cancridae
- Nectariniidae
- Coraciidae
- Garryaceae
- Odacidae
- Juncaginaceae
- Tubiferaceae
- Psychidae
- Heteroziidae
- Zamiaceae
- Pseudothelphusidae
- Monacanthidae
- Achilidae
- Holothuriidae
- Hypocreaceae
- Triozidae
- Aglajidae
- Nostocaceae
- Remizidae
- Trechaleidae
- Perlidae
- Phyllocladaceae
- Boidae
- Diogenidae
- Indriidae
- Moronidae
- Kinosternidae
- Styelidae
- Pocilloporidae
- Gelatinodiscaceae
- Anobiidae
- Myobatrachidae
- Morchellaceae
- Naticidae
- Solaropsidae
- Antrodiaetidae
- Rhinobatidae
- Pollicipedidae
- Scyliorhinidae
- Apionidae
- Ephemeridae
- Calyptraeidae
- Trigonidiidae
- Varunidae
- Athyriaceae
- Dorididae
- Radulaceae
- Heterodontidae
- Xyloryctidae
- Pupillidae
- Dacrymycetaceae
- Otididae
- Geoemydidae
- Trochidae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Montiniaceae
- Staphyleaceae
- Meruliaceae
- Stylidiaceae
- Ochotonidae
- Hymenophyllaceae
- Antilocapridae
- Phrynichidae
- Sargassaceae
- Hystricidae
- Cymatiidae
- Ariolimacidae
- Histeridae
- Sciomyzidae
- Capitonidae
- Paradisaeidae
- Thesiaceae
- Paguridae
- Oniscidae
- Cercopidae
- Casuariidae
- Pallaviciniaceae
- Viviparidae
- Anthomyiidae
- Pseudocerotidae
- Spheniscidae
- Rostratulidae
- Soricidae
- Scorpaenidae
- Ricaniidae
- Callicladiaceae
- Vombatidae
- Frankeniaceae
- Caulerpaceae
- Balistidae
- Eremiaphilidae
- Diomedeidae
- Haloragaceae
- Cheliferidae
- Rhytididae
- Ranellidae
- Selaginellaceae
- Nepticulidae
- Cleidothaeridae
- Zanclidae
- Hygrophoraceae
- Venturiaceae
- Panaceae
- Hybosoridae
- Laminariaceae
- Hydropsychidae
- Nidulariaceae
- Liolaemidae
- Trionychidae
- Ichneumonidae
- Donacidae
- Cryptobranchidae
- Climaciaceae
- Mopaliidae
- Gomphidiaceae
- Metridiidae
- Pharidae
- Isanthidae
- Cerrenaceae
- Gunneraceae
- Lumbricidae
- Evaniidae
- Atherinopsidae
- Stemonitidaceae
- Rhineuridae
- Mycetopodidae
- Eriococcidae
- Rickenellaceae
- Psilotaceae
- Apterygidae
- Aniliidae
- Ophidiasteridae
- Delphacidae
- Gymnosporangiaceae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Cucumaridae
- Pseudoxyrhophiidae
- Apogonidae
- Plexauridae
- Lethrinidae
- Terebratellidae
- Characeae
- Calycanthaceae
- Momphidae
- Gecarcinidae
- Phyllanthaceae
- Pleurobranchidae
- Apatelodidae
- Tapiridae
- Anthribidae
- Tettigometridae
- Climacteridae
- Bacillidae
- Hymenochaetaceae
- Psammophiidae
- Lonchopteridae
- Pellorneidae
- Faviidae
- Xylariaceae
- Caliciaceae
- Acrobatidae
- Symplocaceae
- Monimiaceae
- Lyonsiidae
- Cronartiaceae
- Phleogenaceae
- Hippopotamidae
- Nabidae
- Aulacomniaceae
- Caprimulgidae
- Mycteridae
- Martyniaceae
- Philosciidae
- Prunellidae
- Stenopodidae
- Diadematidae
- Ethmiidae
- Corallinaceae
- Ptilonorhynchidae
- Sarcosomataceae
- Opisthocomidae
- Streptaxidae
- Coleosporiaceae
- Phylliidae
- Channidae
- Ceramiaceae
- Pinguipedidae
- Sabellariidae
- Taphrinaceae
- Xystodesmidae
- Characidae
- Pareidae
- Porellaceae
- Lauxaniidae
- Pilumnidae
- Cancellariidae
- Phasmatidae
- Trapeliaceae
- Isonychiidae
- Camaenidae
- Ptilidiaceae
- Ctenidae
- Zygnemataceae
- Ophiocordycipitaceae
- Scythrididae
- Polychrotidae
- Clupeidae
- Petalophyllaceae
- Serpulidae
- Leptodactylidae
- Anisolabididae
- Sphagnaceae
- Rheidae
- Thuidiaceae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Marchantiaceae
- Trogossitidae
- Polygyridae
- Scaphiopodidae
- Cyatheaceae
- Stichodactylidae
- Lymnaeidae
- Rhinocerotidae
- Hexabranchidae
- Tegulidae
- Scoliciosporaceae
- Anhimidae
- Oenosandridae
- Argyresthiidae
- Lecideaceae
- Nepidae
- Lithodidae
- Veliidae
- Herbertaceae
- Eleutherodactylidae
- Lachesillidae
- Melianthaceae
- Tonicellidae
- Gadidae
- Rhyacotritonidae
- Dendrophylliidae
- Notonectidae
- Dysderidae
- Nacellidae
- Anatinellidae
- Derbidae
- Dendrasteridae
- Gnaphosidae
- Ovulidae
- Argonautidae
- Crangonidae
- Chordariaceae
- Anguillidae
- Blissidae
- Polleniidae
- Cordulegastridae
- Sarcoscyphaceae
- Tethyidae
- Cyclanthaceae
- Renillidae
- Melampsoraceae
- Andrognathidae
- Ascalaphidae
- Peleopodidae
- Clathrinidae
- Oxychilidae
- Zachrysiidae
- Sphenopidae
- Bucerotidae
- Capnellidae
- Agonidae
- Rhinocryptidae
- Lestoideidae
- Pimelodidae
- Cerionidae
- Adrianichthyidae
- Empusidae
- Pannariaceae
- Megalonychidae
- Calophyllaceae
- Tethydidae
- Cadlinidae
- Dreissenidae
- Petrosiidae
- Schizoporaceae
- Talinaceae
- Tephromelataceae
- Lepidoziaceae
- Phyllostomidae
- Stereocaulaceae
- Tropaeolaceae
- Entolomataceae
- Discidae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Grimmiaceae
- Corydiidae
- Cytinaceae
- Anthelidae
- Rhynchocinetidae
- Chitonidae
- Thespidae
- Triopsidae
- Sphaerophragmiaceae
- Thamnophilidae
- Pachygronthidae
- Ariophantidae
- Eirenidae
- Batillariidae
- Linderniaceae
- Myrmeleontidae
- Planorbidae
- Priacanthidae
- Palinuridae
- Delesseriaceae
- Paralichthyidae
- Chloranthaceae
- Cerianthidae
- Graphidaceae
- Gerrhosauridae
- Choreutidae
- Tubariaceae
- Aegiridae
- Pucciniaceae
- Tectariaceae
- Elonidae
- Planaxidae
- Sialidae
- Sebastidae
- Oonopidae
- Cyttariaceae
- Solenopeziaceae
- Loricariidae
- Calliostomatidae
- Geomitridae
- Cimbicidae
- Eschrichtiidae
- Codiaceae
- Cordycipitaceae
- Achatocarpaceae
- Trichiaceae
- Steccherinaceae
- Cunoniaceae
- Aulopidae
- Ulmaridae
- Hemerobiidae
- Leptochitonidae
- Aulostomidae
- Pennariidae
- Cryphaeaceae
- Stenopsocidae
- Amblystegiaceae
- Osmylidae
- Galatheidae
- Pleuronectidae
- Capniidae
- Amelidae
- Chironemidae
- Pylaisiaceae
- Triportheidae
- Nitidulidae
- Paeoniaceae
- Mugilidae
- Alstroemeriaceae
- Geastraceae
- Hexagrammidae
- Labrisomidae
- Arixeniidae
- Leptotyphlopidae
- Molluginaceae
- Oziidae
- Diatrypaceae
- Myopsocidae
- Cordylidae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Brentidae
- Woodsiaceae
- Styracaceae
- Heliornithidae
- Acanaloniidae
- Basellaceae
- Orsodacnidae
- Geoplanidae
- Dasyproctidae
- Belostomatidae
- Costariaceae
- Myrrhinidae
- Blattidae
- Sphaerocarpaceae
- Cyrenidae
- Agaonidae
- Esocidae
- Bankeraceae
- Caliscelidae
- Calophyidae
- Petrospongiaceae
- Limnodynastidae
- Eublepharidae
- Didymiidae
- Zoropsidae
- Caricaceae
- Empididae
- Cebidae
- Nyssaceae
- Chinchillidae
- Cerococcidae
- Hymenopodidae
- Clastopteridae
- Cochlicopidae
- Tridactylidae
- Mellitidae
- Caloscyphaceae
- Himanthaliaceae
- Pteroclididae
- Brachytheciaceae
- Phaethontidae
- Potamidae
- Callionymidae
- Haliplidae
- Pallopteridae
- Scolopendridae
- Aplysinidae
- Boliniaceae
- Cannaceae
- Callitrichidae
- Batrachoididae
- Lyophyllaceae
- Archaeobalanidae
- Canellaceae
- Sepiidae
- Pinnidae
- Colletidae
- Capnodiaceae
- Fissidentaceae
- Gelsemiaceae
- Pergidae
- Terebridae
- Limnanthaceae
- Arbaciidae
- Callyspongiidae
- Parmacellidae
- Psychopsidae
- Agriolimacidae
- Rivetinidae
- Cirratulidae
- Titanoecidae
- Phacidiaceae
- Hydroptilidae
- Strombidae
- Eupterotidae
- Philodromidae
- Albuginaceae
- Stercorariidae
- Pipridae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Bryoxiphiaceae
- Pestalotiopsidaceae
- Thiaridae
- Trichonotidae
- Saccotheciaceae
- Tiphiidae
- Clavelinidae
- Loliginidae
- Acoraceae
- Nemipteridae
- Dromaiidae
- Liparidae
- Moringaceae
- Bernieridae
- Potamididae
- Tellinidae
- Eriocaulaceae
- Autostichidae
- Callaeatidae
- Cyprinodontidae
- Bothidae
- Tanypezidae
- Potyviridae
- Thorectidae
- Piperaceae
- Ochnaceae
- Araucariaceae
- Palpimanidae
- Membraniporidae
- Hysteriaceae
- Mesothuriidae
- Oreasteridae
- Loxsomataceae
- Thalassianthidae
- Leotiaceae
- Agariciidae
- Cephaloziaceae
- Costaceae
- Tecophilaeaceae
- Stiphidiidae
- Grubbiaceae
- Ranixalidae
- Artheneidae
- Amaurobiidae
- Aphredoderidae
- Mictyridae
- Meessiidae
- Toxopidae
- Radulomycetaceae
- Chyphotidae
- Diadumenidae
- Polyphysaceae
- Cidaridae
- Fistulariidae
- Inachoididae
- Cyaneidae
- Porpitidae
- Nepenthaceae
- Polydesmidae
- Batrachylidae
- Nemastomatidae
- Spirobolidae
- Zoarcidae
- Largidae
- Rhabdoweisiaceae
- Arminidae
- Lemindidae
- Picrodendraceae
- Eulophidae
- Urocystidaceae
- Dinolestidae
- Stomopneustidae
- Monodactylidae
- Terapontidae
- Anemiaceae
- Nassariidae
- Luidiidae
- Metzgeriaceae
- Menuridae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Nogodinidae
- Asteropseidae
- Siganidae
- Rhinotermitidae
- Glaucostegidae
- Blastobasidae
- Calceolariaceae
- Corynocarpaceae
- Achiridae
- Melandryidae
- Termitidae
- Physidae
- Cibotiaceae
- Geophilidae
- Tarsiidae
- Polydnaviridae
- Goniasteridae
- Amiidae
- Coniocybaceae
- Geocoridae
- Subergorgiidae
- Fossombroniaceae
- Miturgidae
- Rousseaceae
- Sparassidaceae
- Pyronemataceae
- Hypoxidaceae
- Ypsolophidae
- Lacturidae
- Mimetidae
- Tofieldiaceae
- Ampulicidae
- Nothofagaceae
- Gastrodontidae
- Rhynchitidae
- Lithoglyphidae
- Mormoopidae
- Entodontaceae
- Parechinidae
- Homalopsidae
- Platysteiridae
- Musophagidae
- Philesiaceae
- Triaenonychidae
- Pelagiidae
- Salvadoraceae
- Paxillaceae
- Trichechidae
- Arhynchobatidae
- Panopeidae
- Aeolidiidae
- Phalangopsidae
- Ephippidae
- Bucconidae
- Conidae
- Argiolestidae
- Scouleriaceae
- Artamidae
- Panorpidae
- Hemitripteridae
- Ampullariidae
- Pempheridae
- Sphaerophoraceae
- Echinasteridae
- Orthalicidae
- Galaxiidae
- Glyphipterigidae
- Pentaphylacaceae
- Succineidae
- Rhizocarpaceae
- Tetrarogidae
- Mydidae
- Scotocercidae
- Etmopteridae
- Incurvariidae
- Filistatidae
- Osphronemidae
- Stathmopodidae
- Rhizangiidae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Cuthonidae
- Caryocaraceae
- Helicarionidae
- Entylomataceae
- Hypopterygiaceae
- Lysmatidae
- Plagiogyriaceae
- Tremellodendropsidaceae
- Podosporaceae
- Chilodontaidae
- Beesoniidae
- Rathouisiidae
- Erythropalaceae
- Leucosiidae
- Ascidiidae
- Hormosiraceae
- Synaptidae
- Ptychomitriaceae
- Lindsaeaceae
- Anthicidae
- Rhizobiaceae
- Pelecinidae
- Inocybaceae
- Scatopsidae
- Dactylopteridae
- Baeomycetaceae
- Pylaisiadelphaceae
- Dromiidae
- Glycymerididae
- Dicroglossidae
- Janolidae
- Xylophagidae
- Putranjivaceae
- Conocephalaceae
- Clusiaceae
- Pitheciidae
- Scytosiphonaceae
- Synanceiidae
- Ditrichaceae
- Alsodidae
- Bostrichidae
- Ganodermataceae
- Opistognathidae
- Parazoanthidae
- Epaphroditidae
- Carposinidae
- Hymedesmiidae
- Corcoracidae
- Nelumbonaceae
- Caesionidae
- Sclerotiniaceae
- Pedaliaceae
- Aiptasiidae
- Mesodesmatidae
- Braconidae
- Uromycladiaceae
- Rhinocricidae
- Chironomidae
- Nemopteridae
- Aongstroemiaceae
- Bibionidae
- Flabellinopsidae
- Eucnemidae
- Ortheziidae
- Pyrosomatidae
- Dasyuridae
- Crambeidae
- Peucedramidae
- Mikronegeriaceae
- Cyclophoridae
- Ascodichaenaceae
- Tetratomidae
- Udoteaceae
- Epipyropidae
- Monocleaceae
- Prophalangopsidae
- Eriphiidae
- Pygopodidae
- Cheiracanthiidae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Anyphaenidae
- Aegithinidae
- Corinnidae
- Issidae
- Psocidae
- Cominellidae
- Marsileaceae
- Tinamidae
- Clausiliidae
- Triakidae
- Naemateliaceae
- Neritidae
- Atherospermataceae
- Lesquereusiidae
- Nectriaceae
- Majidae
- Cephalotaxaceae
- Sarcobataceae
- Cimicidae
- Lucinidae
- Zoanthidae
- Castniidae
- Carphodactylidae
- Pterocladiaceae
- Aleyrodidae
- Onychiuridae
- Fontinalaceae
- Actinocyclidae
- Temnopleuridae
- Scytomonadidae
- Platycephalidae
- Eleotridae
- Helicodontidae
- Laboulbeniaceae
- Punctidae
- Hynobiidae
- Mortierellaceae
- Lichinaceae
- Orbiliaceae
- Pomatostomidae
- Phrynidae
- Siliquariidae
- Acholeplasmataceae
- Boletinellaceae
- Nandidae
- Pseudomelatomidae
- Alytidae
- Catostylidae
- Centrolophidae
- Hydrometridae
- Cimeliidae
- Hedwigiaceae
- Nephropidae
- Boreidae
- Pythidae
- Mysidae
- Casuarinaceae
- Bataceae
- Mycosphaerellaceae
- Cyphellaceae
- Megasporaceae
- Ephydridae
- Calycidorididae
- Aplodactylidae
- Clionaidae
- Echinidae
- Phalacridae
- Lobophylliidae
- Acarosporaceae
- Aglaopheniidae
- Heleomyzidae
- Mycocaliciaceae
- Pamphiliidae
- Daphniidae
- Philinidae
- Macrophthalmidae
- Zonitidae
- Phyllotopsidaceae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Elachistidae
- Velutinidae
- Salpingidae
- Oecobiidae
- Camelidae
- Drosophilidae
- Bullidae
- Engraulidae
- Buccinidae
- Myuriaceae
- Xanthidae
- Ptychopteridae
- Ochrolechiaceae
- Cucujidae
- Chlorospleniaceae
- Gavialidae
- Phanerochaetaceae
- Liocranidae
- Caeciliusidae
- Erythrotrichiaceae
- Oreohelicidae
- Tetranychidae
- Ceratiomyxaceae
- Phycidae
- Trombidiidae
- Petropedetidae
- Douglasiidae
- Cryphonectriaceae
- Megophryidae
- Stenocephalidae
- Varroidae
- Balaenicipitidae
- Periplomatidae
- Phocoenidae
- Soleidae
- Tischeriidae
- Dotillidae
- Superstitioniidae
- Hermaeidae
- Niphatidae
- Phoeniculidae
- Anseranatidae
- Stichaeidae
- Ximeniaceae
- Valloniidae
- Sematophyllaceae
- Armadillidae
- Acarnidae
- Centropomidae
- Coriariaceae
- Heterogastridae
- Endomychidae
- Phacidae
- Alestidae
- Phyllachoraceae
- Carpiliidae
- Nanomantidae
- Viverridae
- Perophoridae
- Phyllodocidae
- Thaumatoneuridae
- Psophodidae
- Hormathiidae
- Geomyidae
- Antedonidae
- Bixaceae
- Sepsidae
- Hysterangiaceae
- Mesotaeniaceae
- Tripedaliidae
- Peripsocidae
- Pelobatidae
- Rhincodontidae
- Gelastocoridae
- Lecithoceridae
- Plagiochilaceae
- Ophioparmaceae
- Lomisidae
- Meandrinidae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Welwitschiaceae
- Oleandraceae
- Leucettidae
- Belonidae
- Myochamidae
- Polymastiidae
- Desidae
- Peraceae
- Stegostomatidae
- Durvillaeaceae
- Hybotidae
- Aotidae
- Gerreidae
- Silvanidae
- Chrysobalanaceae
- Noctilucaceae
- Rhinidae
- Scheuchzeriaceae
- Tipulidae
- Atyidae
- Iteaceae
- Ophiotrichidae
- Ancillariidae
- Cystopeltidae
- Cryptopidae
- Doidae
- Triphoridae
- Phengodidae
- Clavicipitaceae
- Gonyaulacaceae
- Ophiocomidae
- Lymexylidae
- Limidae
- Chondrillidae
- Plumulariidae
- Eratoidae
- Psoraceae
- Rissoinidae
- Atypidae
- Schizaeaceae
- Grammatidae
- Keratoisididae
- Blaniulidae
- Myidae
- Isognomonidae
- Hemiprocnidae
- Stromateidae
- Noetiidae
- Peripatopsidae
- Babyloniidae
- Coryphaenidae
- Plerogyridae
- Stylostomidae
- Typhulaceae
- Masaridae
- Coptopterygidae
- Megapodagrionidae
- Noteridae
- Nyctibiidae
- Mitrospingidae
- Mithrodiidae
- Borsoniidae
- Lemuridae
- Diplodactylidae
- Onchidiidae
- Pelomedusidae
- Sagdidae
- Nemesiidae
- Isostictidae
- Sarothruridae
- Actinidiaceae
- Saldidae
- Microbotryaceae
- Microhexuridae
- Isididae
- Bedelliidae
- Atherinidae
- Phyllophoraceae
- Comandraceae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Corystidae
- Pteromalini
- Machilidae
- Datiscaceae
- Polyphemidae
- Eurypygidae
- Hernandiaceae
- Didemnidae
- Amphisbaenidae
- Thelephoraceae
- Eupleridae
- Spirulidae
- Corallimorphidae
- Olindiidae
- Scytodidae
- Leucospidae
- Alcyonidiidae
- Darwinellidae
- Lamprophiidae
- Pseudomonadaceae
- Lypusidae
- Tragulidae
- Monocentridae
- Ptinidae
- Hodotermitidae
- Stolotermitidae
- Pseudodistomidae
- Umbraculidae
- Leptophlebiidae
- Bucculatricidae
- Aradidae
- Pandeidae
- Filobasidiaceae
- Entomophthoraceae
- Sclerodactylidae
- Argophyllaceae
- Aspergillaceae
- Talitridae
- Horneridae
- Miomantidae
- Macrosphenidae
- Embiotocidae
- Alariaceae
- Rivulidae
- Tapinellaceae
- Rhinolophidae
- Hemiramphidae
- Aplustridae
- Palmariaceae
- Cycadaceae
- Gnomoniaceae
- Cepheidae
- Columelliaceae
- Axinellidae
- Psilidae
- Chloropseidae
- Incrustoporiaceae
- Arkyidae
- Haemodoraceae
- Clusiidae
- Dacryobolaceae
- Cerastidae
- Athoracophoridae
- Chalcididae
- Stenopelmatidae
- Aracanidae
- Corambidae
- Cyrillaceae
- Cardiopteridaceae
- Splachnaceae
- Chlorociboriaceae
- Bdellidae
- Sepiolidae
- Heptageniidae
- Gliridae
- Hippolytidae
- Plocamiaceae
- Acanthisittidae
- Poritidae

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

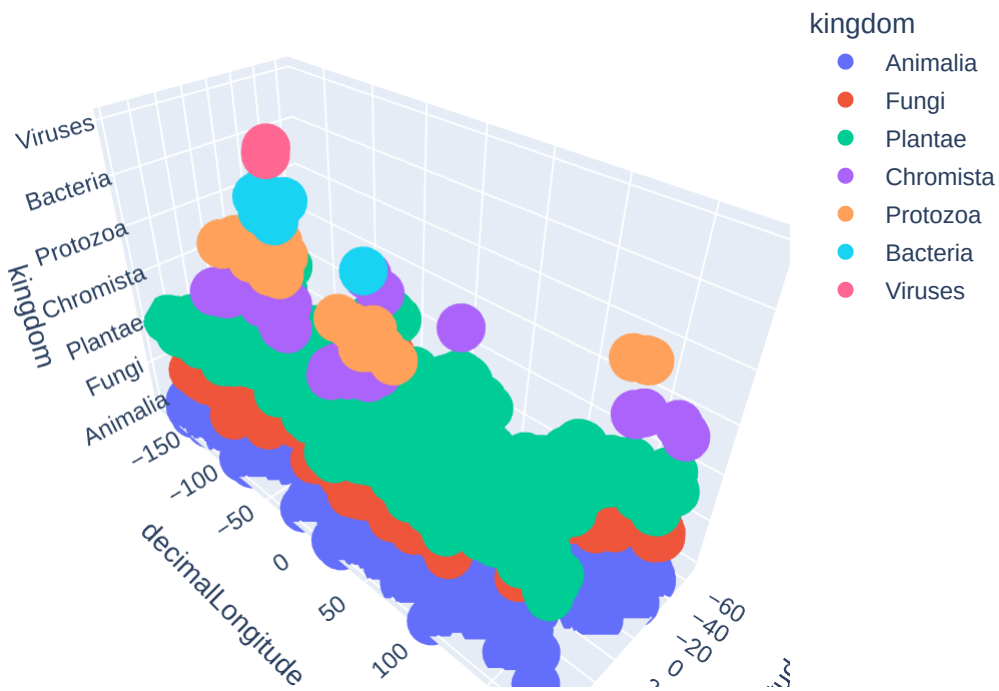
- Myrmecophilidae
- Collemataceae
- Aididae
- Heteractidae
- Echinosteliaceae
- Coniophoraceae
- Pseudeurotiaceae
- Psolidae
- Endromidae
- Pennantiaceae
- Leiodidae
- Eriocraniidae
- Rhizostomatidae
- Adelgidae
- Molidae
- Pholadidae
- Eledonidae
- Amphibulimidae
- Cixiidae
- Peniophoraceae
- Echinarachniidae
- Turbinellidae
- Odontobutidae
- Valvatidae
- Sebacinaceae
- Cirolanidae
- Chaenopsidae
- Cylisticidae
- Sapygidae
- Megomphicidae
- Lecanographaceae
- Scopidae
- Seirococcaceae
- Pamphagidae
- Lotidae
- Sabiaceae
- Winteraceae
- Amphibolidae
- Sphaeromatidae
- Tonnidae
- Diphysciaceae
- Malesherbiaceae
- Turritellidae

In [44]:

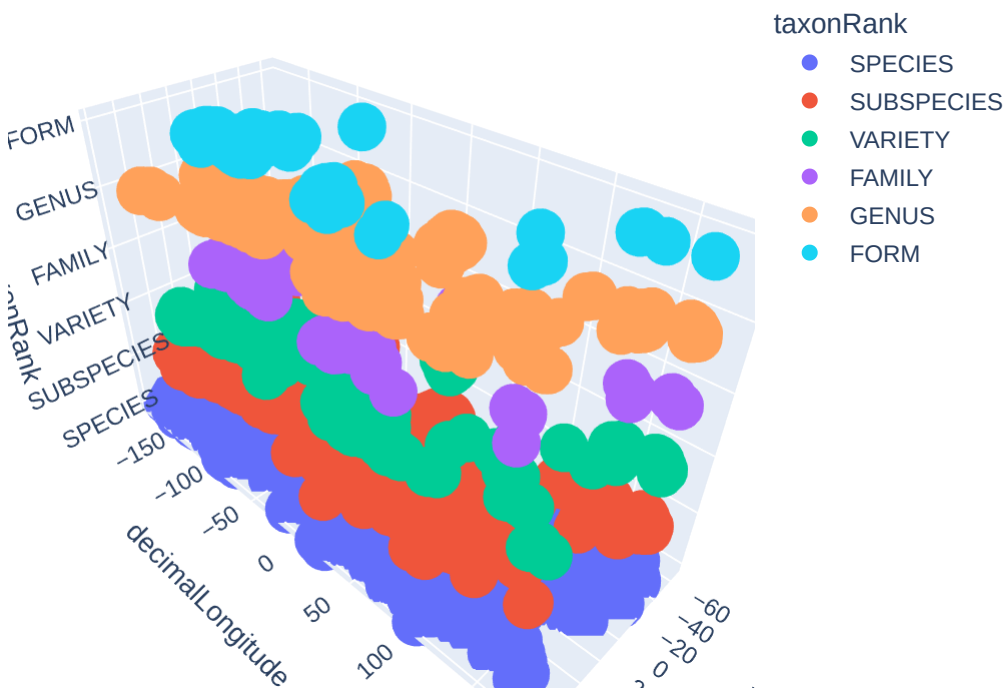
```
import plotly.express as px

# 3D Scatter Plot 1: Using 'kingdom' for color
fig1 = px.scatter_3d(df,
                     x='decimalLatitude',
                     y='decimalLongitude',
                     z='kingdom',
                     color='kingdom',
                     title='3D Scatter Plot: Decimal Latitude vs Decimal Longitude vs
fig1.show()

# 3D Scatter Plot 2: Using 'taxonRank' for color
fig2 = px.scatter_3d(df,
                     x='decimalLatitude',
                     y='decimalLongitude',
                     z='taxonRank',
                     color='taxonRank',
                     title='3D Scatter Plot: Decimal Latitude vs Decimal Longitude vs
fig2.show()

# 3D Scatter Plot 3: Using 'phylum' for color
fig3 = px.scatter_3d(df,
                     x='decimalLatitude',
                     y='decimalLongitude',
                     z='phylum',
                     color='phylum',
                     title='3D Scatter Plot: Decimal Latitude vs Decimal Longitude vs
fig3.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

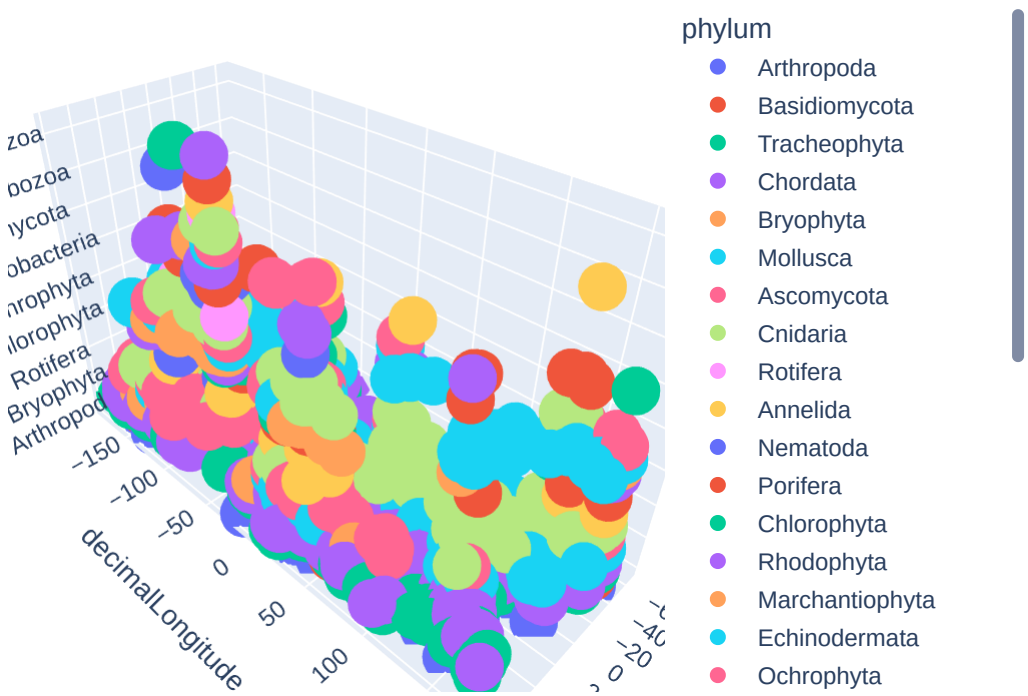
3D Scatter Plot: Decimal Latitude vs Decimal Longitude vs Kingdom



3D Scatter Plot: Decimal Latitude vs Decimal Longitude vs Taxon Rank



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [45]:
```
num_col
```

Out[45]:
```
['decimalLatitude',
 'decimalLongitude',
 'coordinateUncertaintyInMeters',
 'day',
 'month',
 'year',
 'taxonKey',
 'speciesKey',
 'catalogNumber']
```

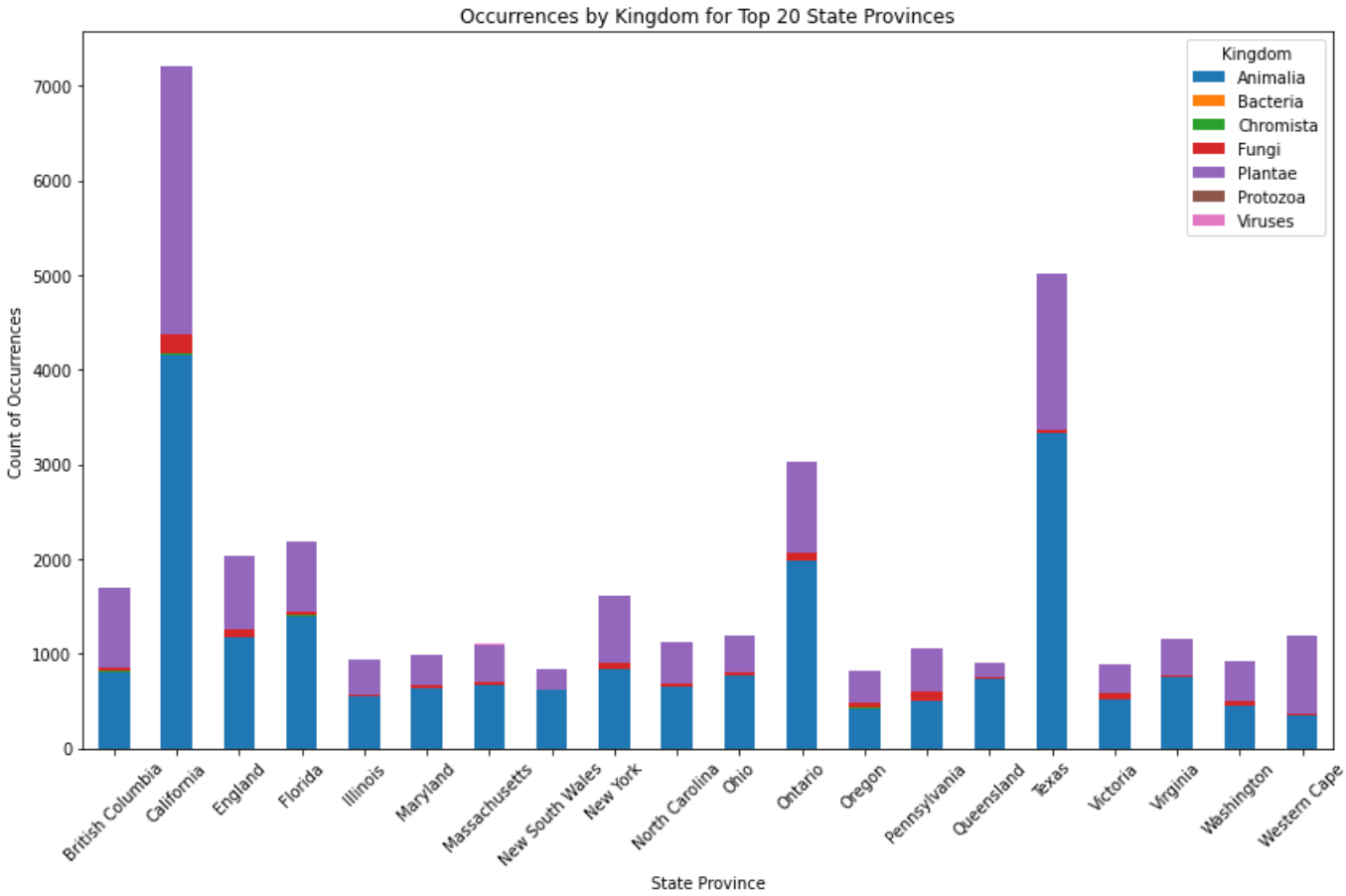
In [46]:
```
df['stateProvince'].value_counts()
```

Out[46]:
```
stateProvince
California              7210
Texas                   5007
Ontario                 3020
Florida                 2192
England                 2034
                        ...
Saint Thomas Lowland       1
Boa Vista                  1
Daman and Diu              1
Iași                       1
Gyeongsangbuk-do           1
Name: count, Length: 1741, dtype: int64
```

In [47]:
```
import matplotlib.pyplot as plt

# Group by kingdom and stateProvince, then count occurrences
occurrence_counts = df.groupby(['kingdom', 'stateProvince']).size().reset_index(name=
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
# Get the top 20 State Provinces based on total occurrences
top_provinces = occurrence_counts.groupby('stateProvince')['count'].sum().nlargest(20

# Filter the dataset for top 20 provinces
top_occurrences = occurrence_counts[occurrence_counts['stateProvince'].isin(top_provi

# Pivot the data for stacked bar plot
pivot_data = top_occurrences.pivot(index='stateProvince', columns='kingdom', values='

# Stacked bar plot
pivot_data.plot(kind='bar', stacked=True, figsize=(12, 8))

# Plot settings
plt.title('Occurrences by Kingdom for Top 20 State Provinces')
plt.xlabel('State Province')
plt.ylabel('Count of Occurrences')
plt.xticks(rotation=45)
plt.legend(title='Kingdom')

# Adjust layout and show plot
plt.tight_layout()
plt.show()
```

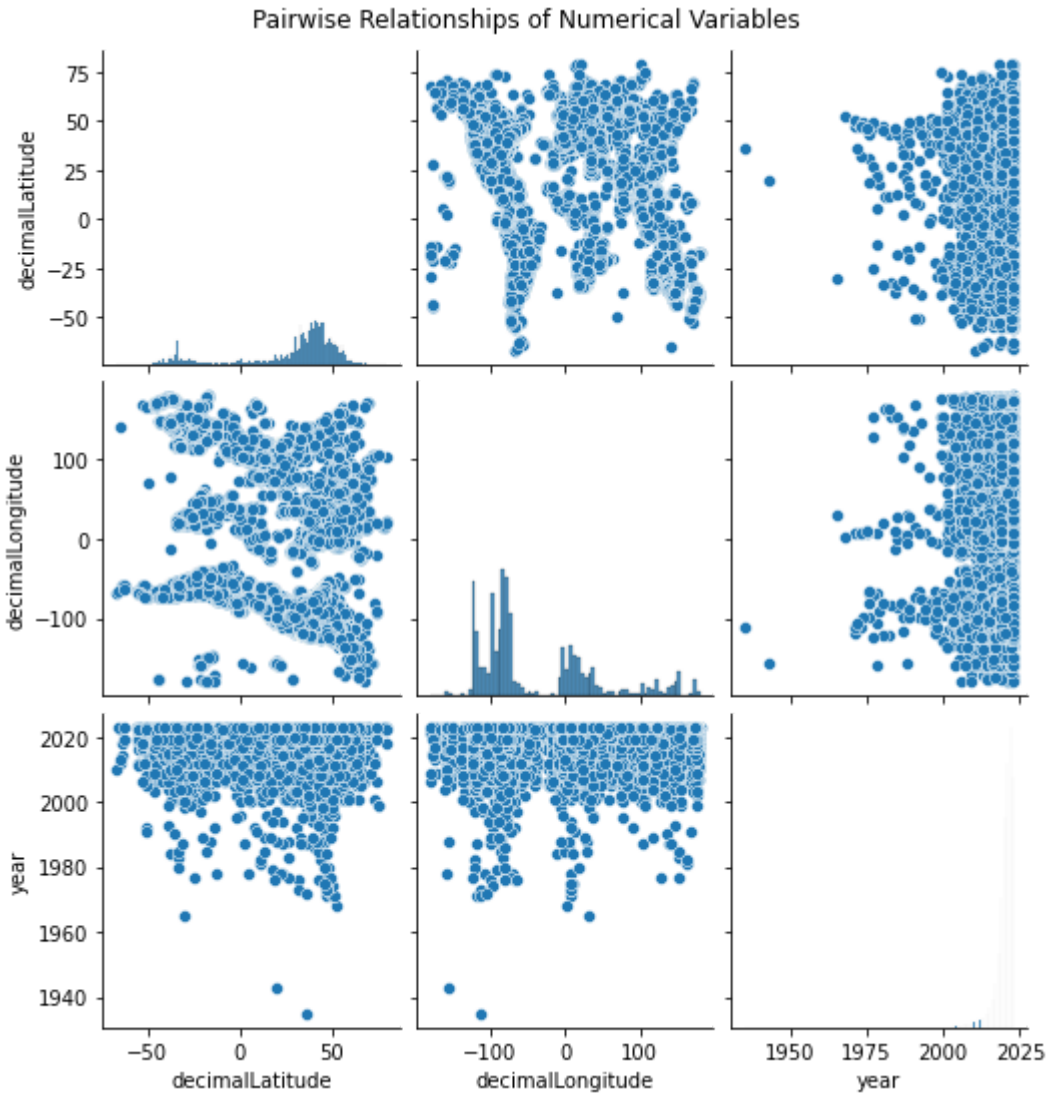


In [48]:

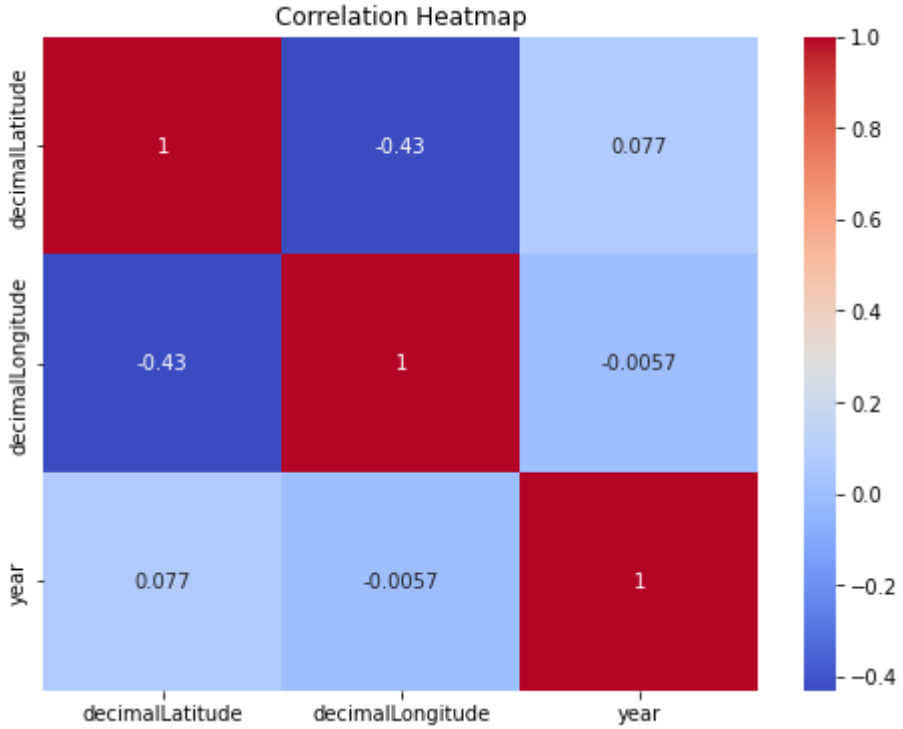
```
import seaborn as sns
import matplotlib.pyplot as plt
#import geopandas as gpd

# Filtering non-missing value columns
filtered_data = df[['decimalLatitude', 'decimalLongitude', 'year', 'kingdom', 'phylum

# 1. Pairwise Relationships for Numerical Columns
sns.pairplot(filtered_data[['decimalLatitude', 'decimalLongitude', 'year']])
plt.suptitle("Pairwise Relationships of Numerical Variables", y=1.02)
plt.show()

# 2. Correlation Heatmap for Numerical Variables
plt.figure(figsize=(8,6))
                                    lLatitude', 'decimalLongitude', 'year']].corr(), an
```

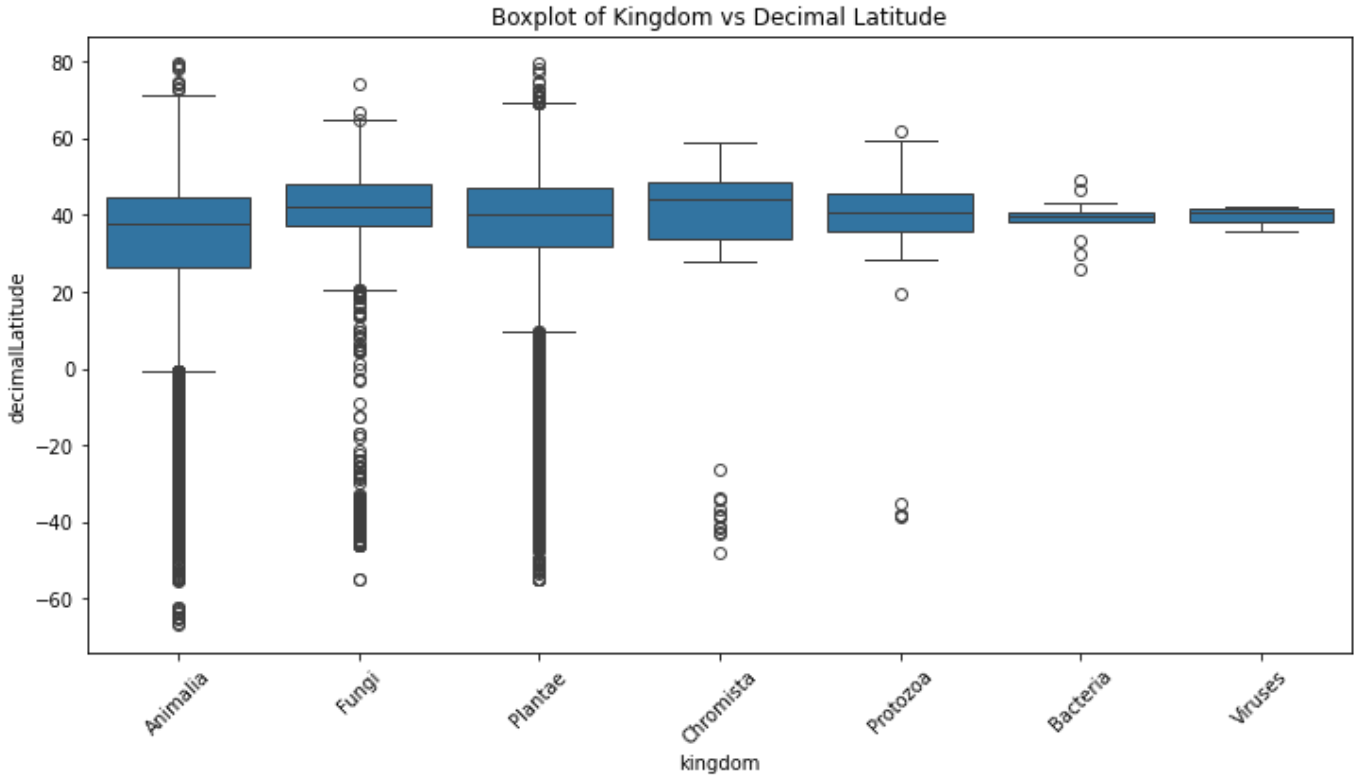
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
plt.title("Correlation Heatmap")
plt.show()

# 3. Boxplot and Violin Plot for Categorical-Numerical Relationship (e.g., kingdom vs
plt.figure(figsize=(12,6))
sns.boxplot(x='kingdom', y='decimalLatitude', data=filtered_data)
plt.title("Boxplot of Kingdom vs Decimal Latitude")
plt.xticks(rotation=45)
plt.show()

plt.figure(figsize=(12,6))
sns.violinplot(x='kingdom', y='decimalLatitude', data=filtered_data)
plt.title("Violin Plot of Kingdom vs Decimal Latitude")
plt.xticks(rotation=45)
plt.show()

# 4. Geospatial Visualization using Longitude and Latitude
plt.figure(figsize=(10,6))
plt.scatter(filtered_data['decimalLongitude'], filtered_data['decimalLatitude'], alph
plt.title("Geospatial Distribution of Observations")
plt.xlabel("Longitude")
plt.ylabel("Latitude")
plt.show()

# 5. Temporal Trends in Number of Records per Year
plt.figure(figsize=(10,5))
sns.countplot(x='year', data=filtered_data)
plt.title("Number of Records by Year")
plt.xticks(rotation=45)
plt.show()
```

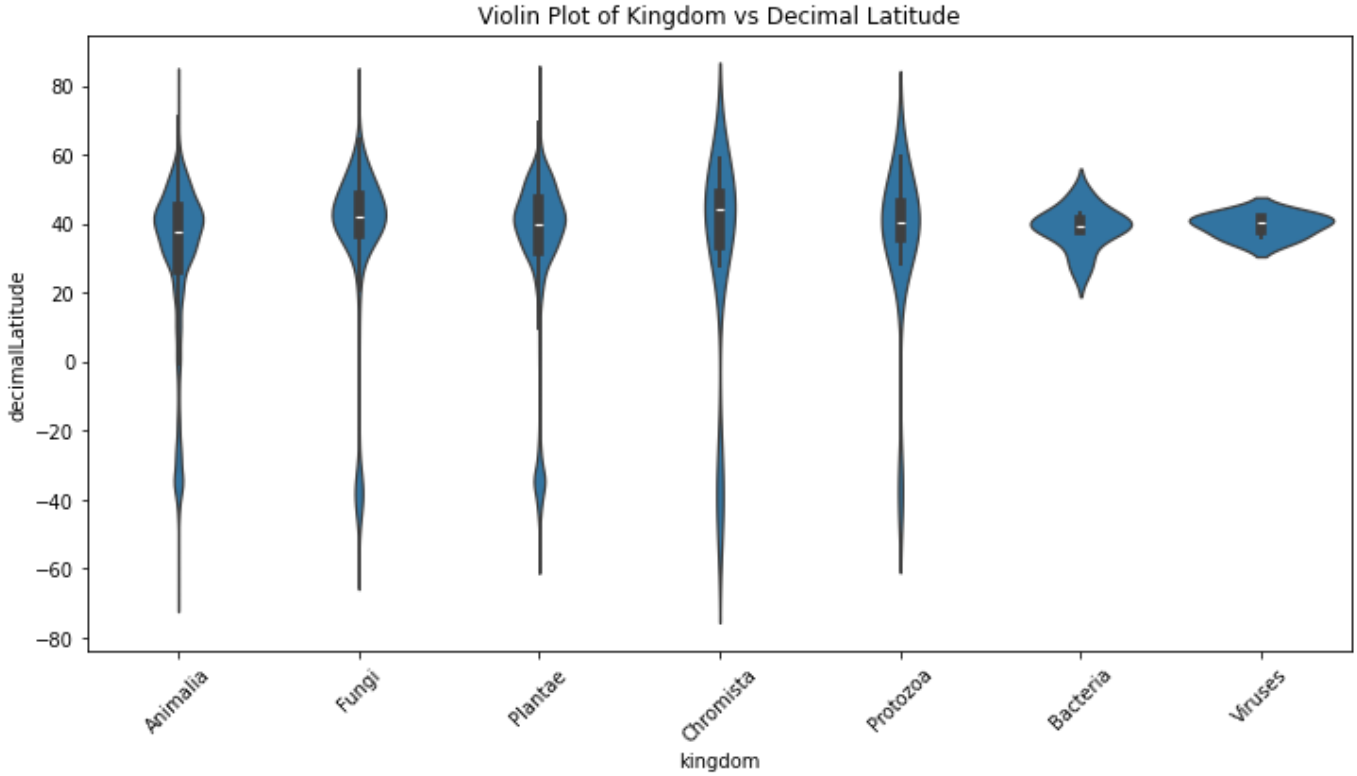


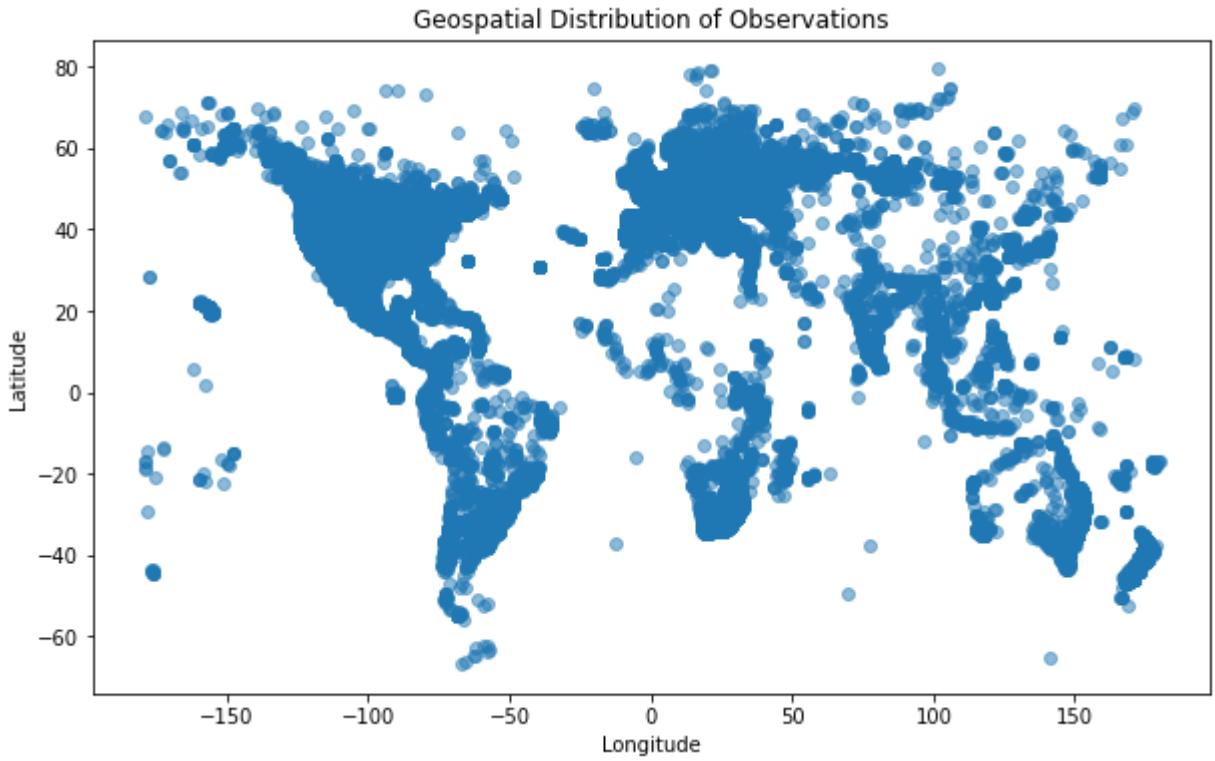
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



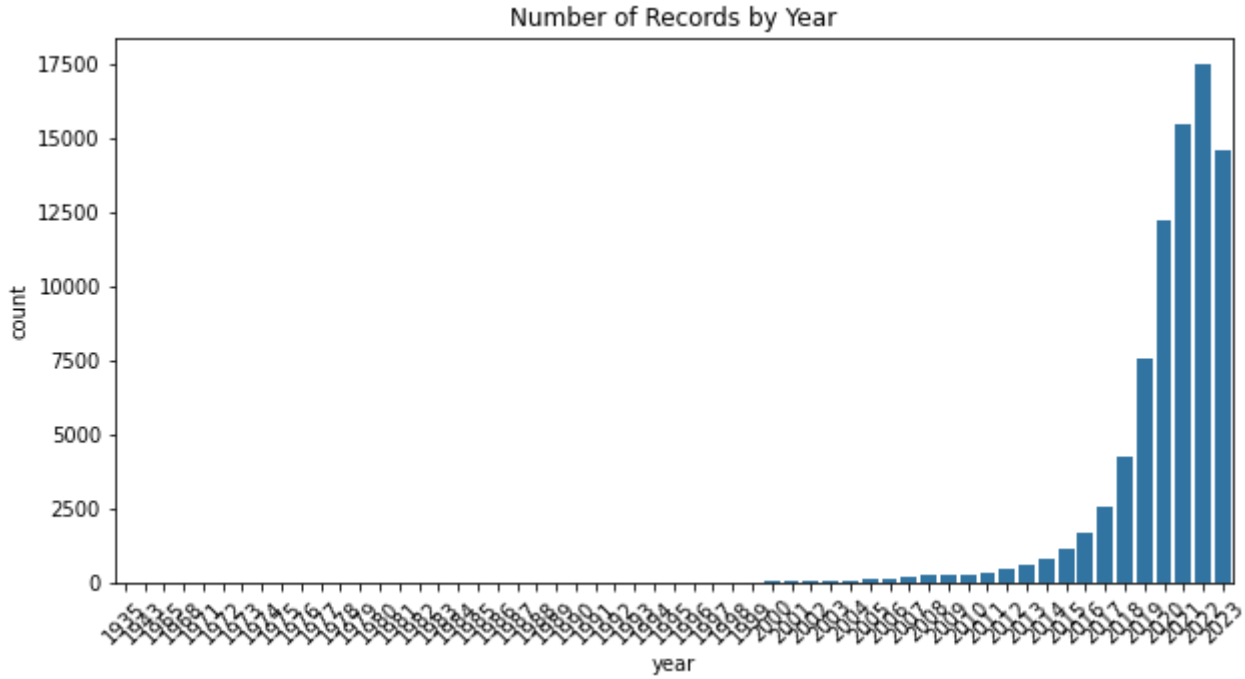


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js





Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [49]:
```
df.columns
```

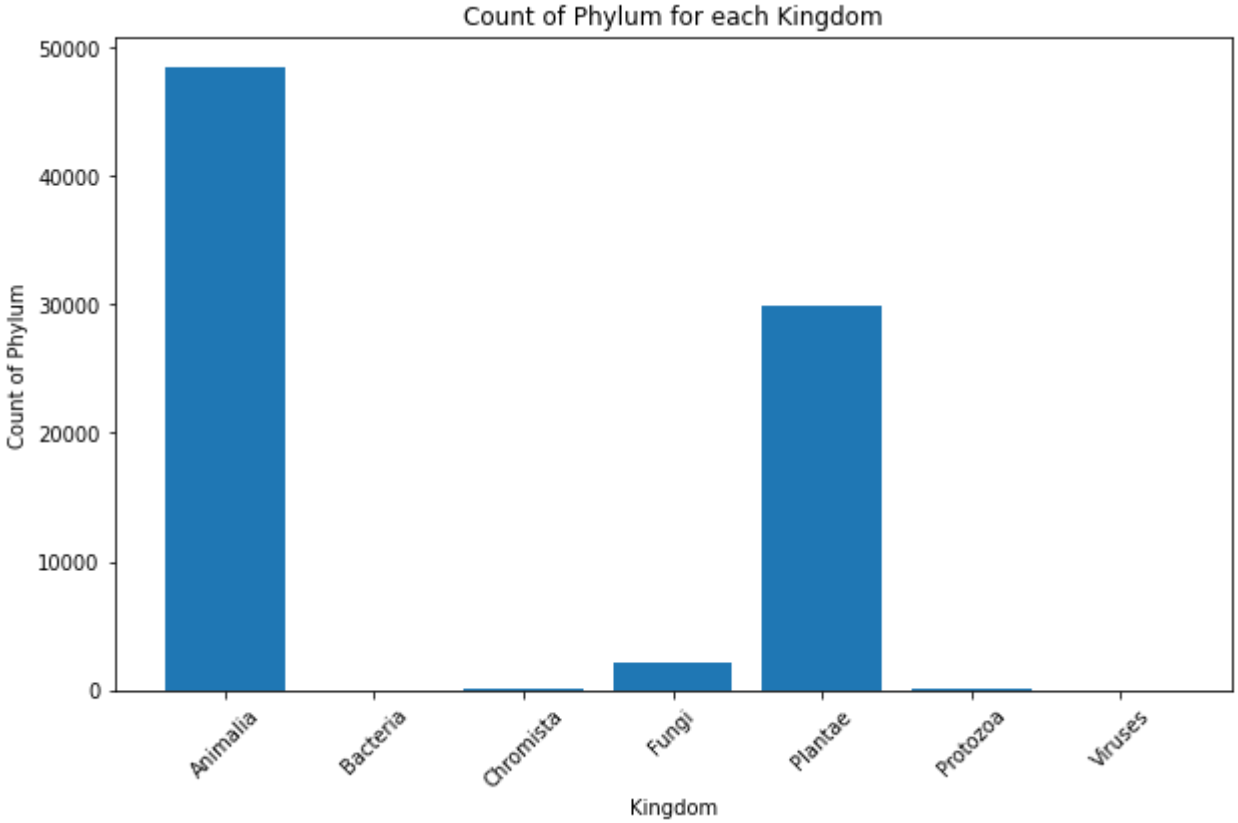
Out[49]:
```
Index(['kingdom', 'phylum', 'class', 'order', 'family', 'genus', 'species',
       'taxonRank', 'scientificName', 'verbatimScientificName', 'countryCode',
       'stateProvince', 'decimalLatitude', 'decimalLongitude',
       'coordinateUncertaintyInMeters', 'eventDate', 'day', 'month', 'year',
       'taxonKey', 'speciesKey', 'catalogNumber', 'identifiedBy',
       'dateIdentified', 'rightsHolder', 'recordedBy', 'lastInterpreted',
       'mediaType', 'issue'],
      dtype='object')
```

In [50]:
```
group_df=df.groupby('kingdom')['phylum'].count()
group_df
```

Out[50]:
```
kingdom
Animalia     48414
Bacteria        14
Chromista       70
Fungi         2109
Plantae      29914
Protozoa        46
Viruses          3
Name: phylum, dtype: int64
```

In [51]:
```
grouped_df=df.groupby('kingdom')['phylum'].count()
plt.figure(figsize=(10, 6))
plt.bar(grouped_df.index, grouped_df.values)
plt.xlabel("Kingdom")
plt.ylabel("Count of Phylum")
plt.title("Count of Phylum for each Kingdom")
plt.xticks(rotation=45)
plt.show()
grouped_df
```

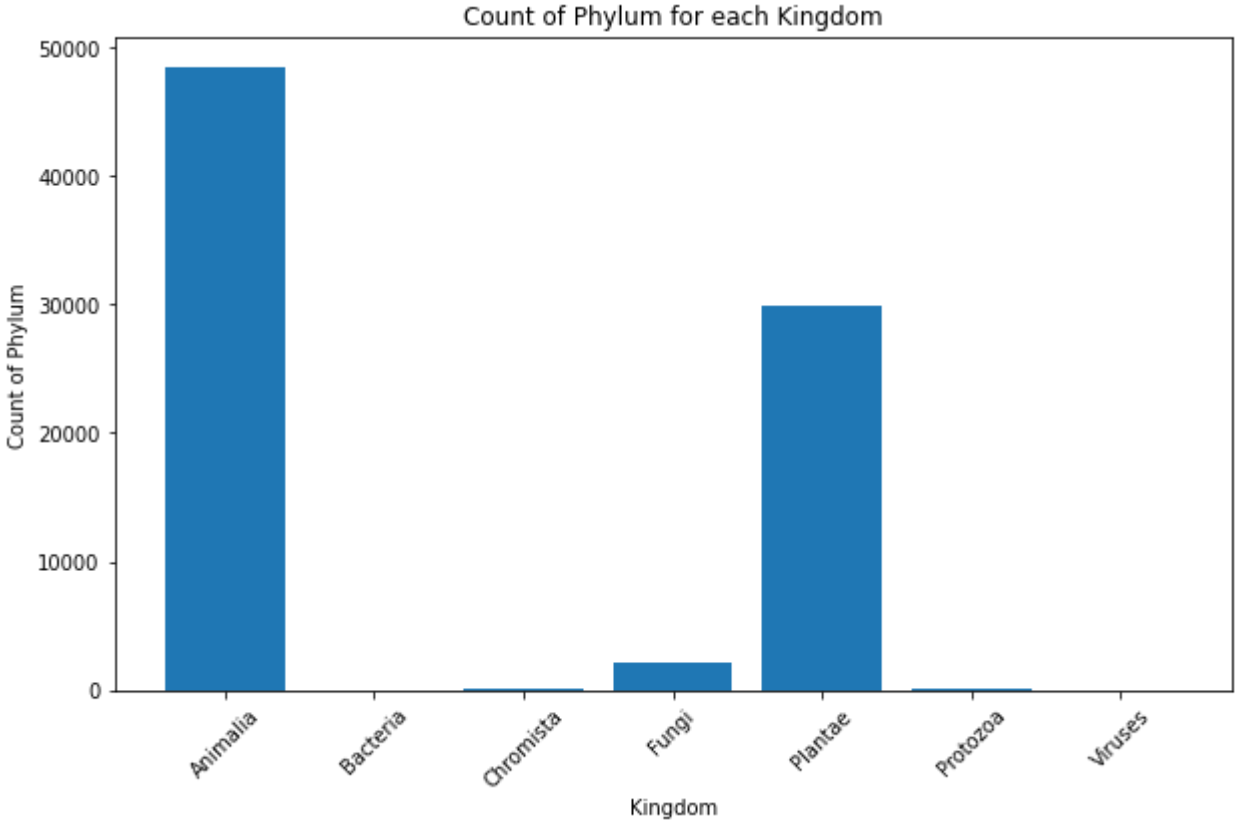
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



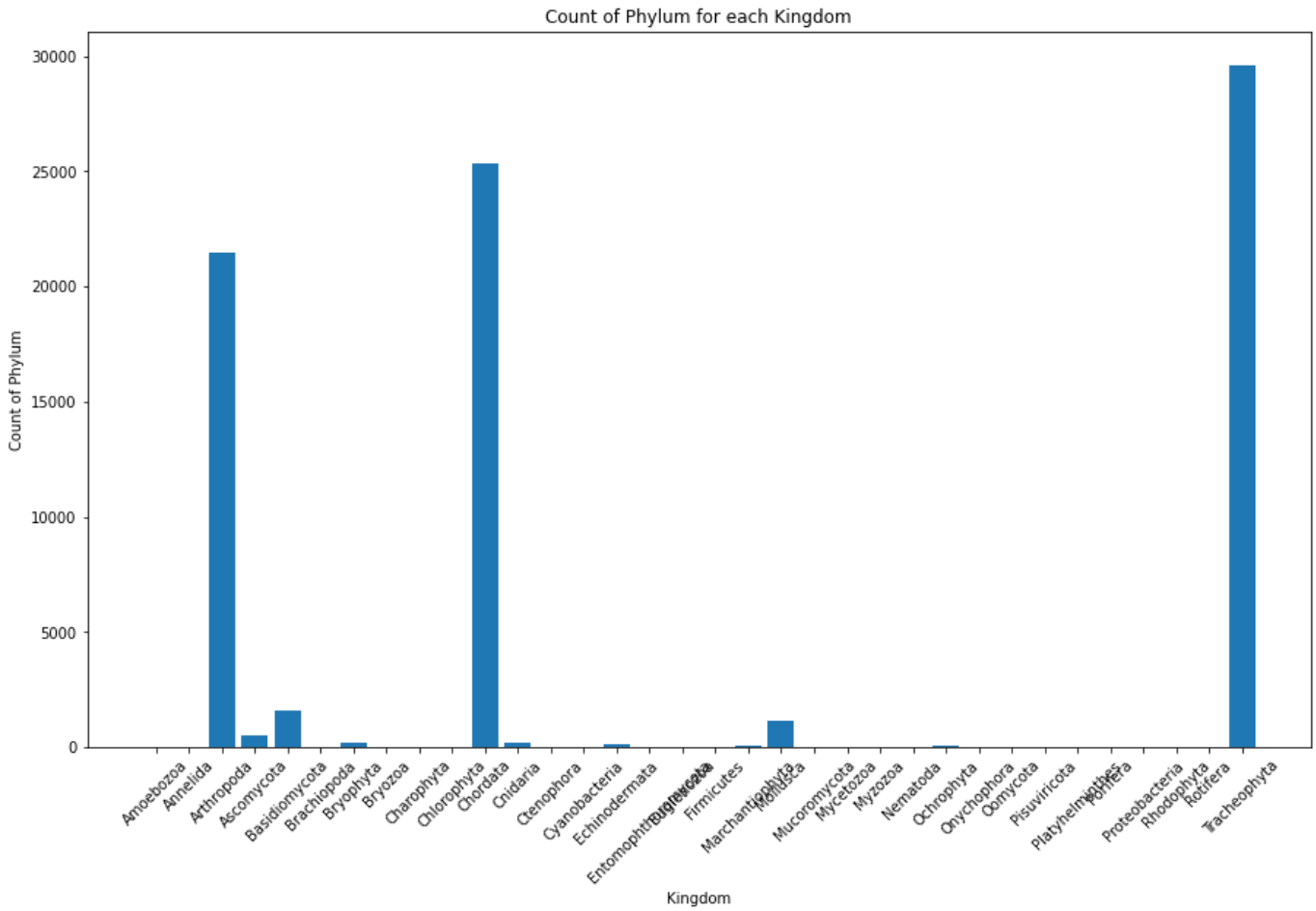
Out[51]:
```
kingdom
Animalia     48414
Bacteria        14
Chromista       70
Fungi         2109
Plantae      29914
Protozoa        46
Viruses          3
Name: phylum, dtype: int64
```

In [52]:
```
grouped_df=df.groupby('kingdom')['phylum'].count()
plt.figure(figsize=(10, 6))
plt.bar(grouped_df.index, grouped_df.values)
plt.xlabel("Kingdom")
plt.ylabel("Count of Phylum")
plt.title("Count of Phylum for each Kingdom")
plt.xticks(rotation=45)
plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



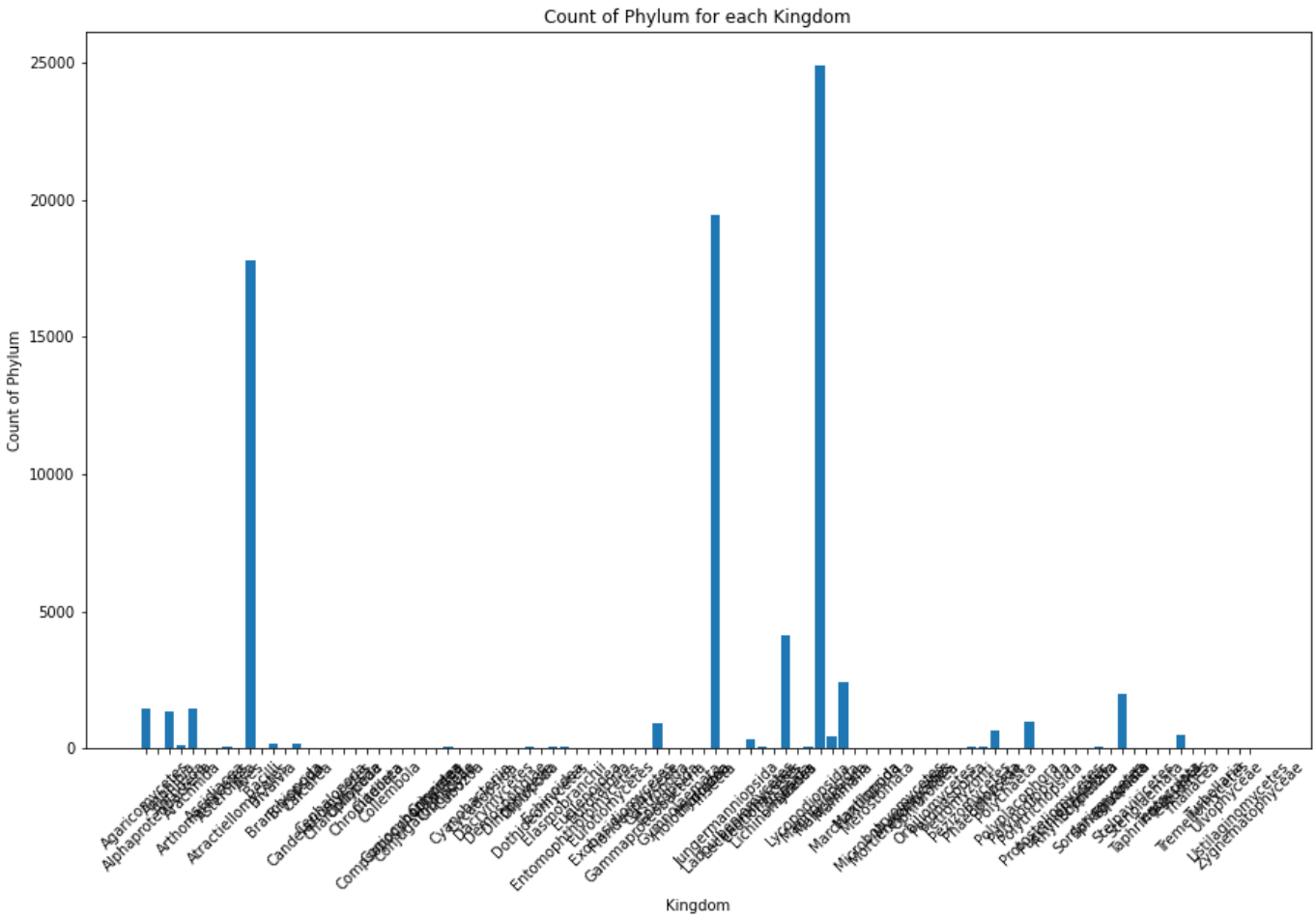
In [53]:
```
grouped_df=df.groupby('phylum')['class'].count()
plt.figure(figsize=(15, 9))
plt.bar(grouped_df.index, grouped_df.values)
plt.xlabel("Kingdom")
plt.ylabel("Count of Phylum")
plt.title("Count of Phylum for each Kingdom")
plt.xticks(rotation=45)
plt.tight_layout
plt.show()
```



In [54]:
```
der'].count()
```

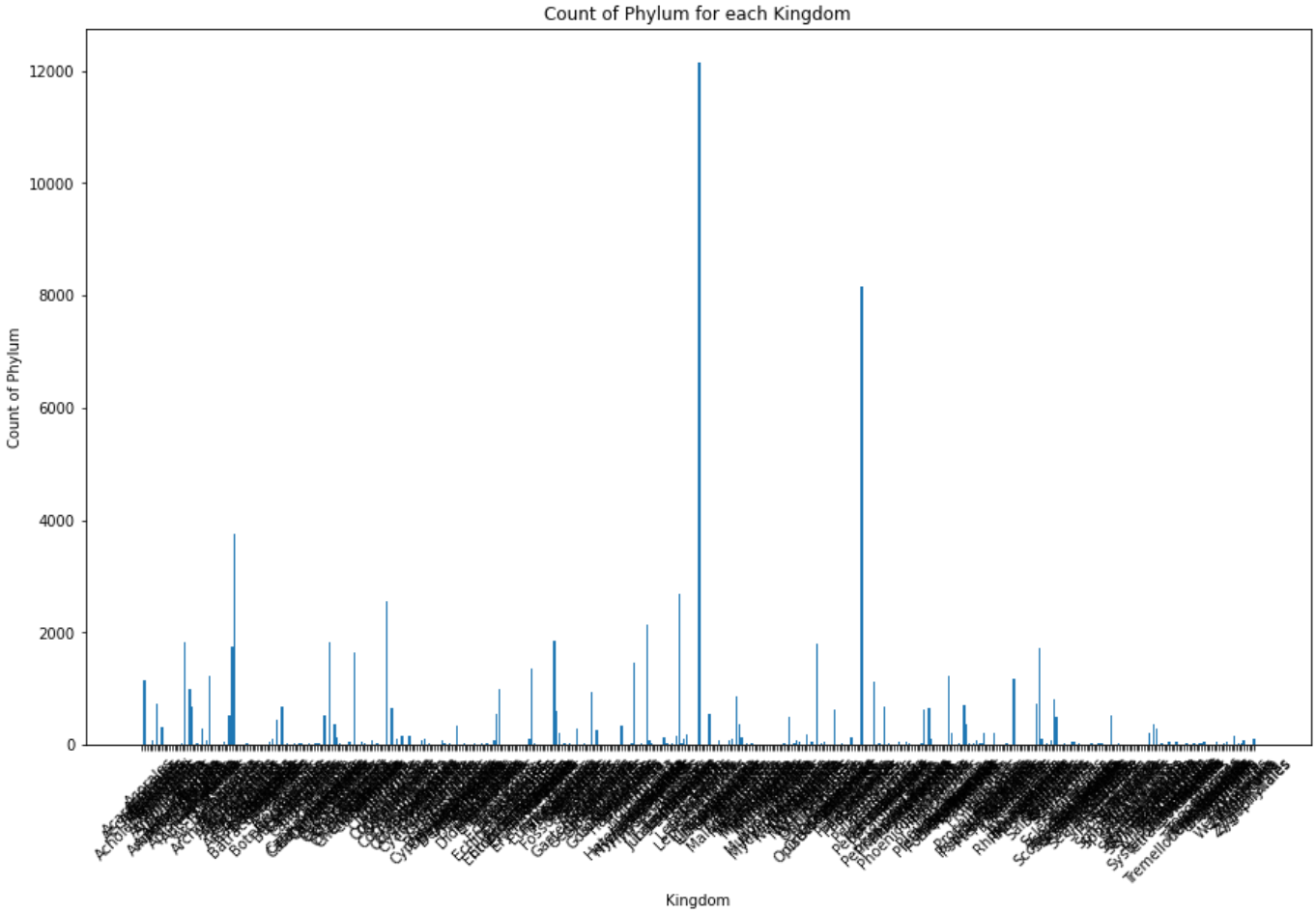
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
plt.figure(figsize=(15, 9))
plt.bar(grouped_df.index, grouped_df.values)
plt.xlabel("Kingdom")
plt.ylabel("Count of Phylum")
plt.title("Count of Phylum for each Kingdom")
plt.xticks(rotation=45)
plt.tight_layout
plt.show()
```



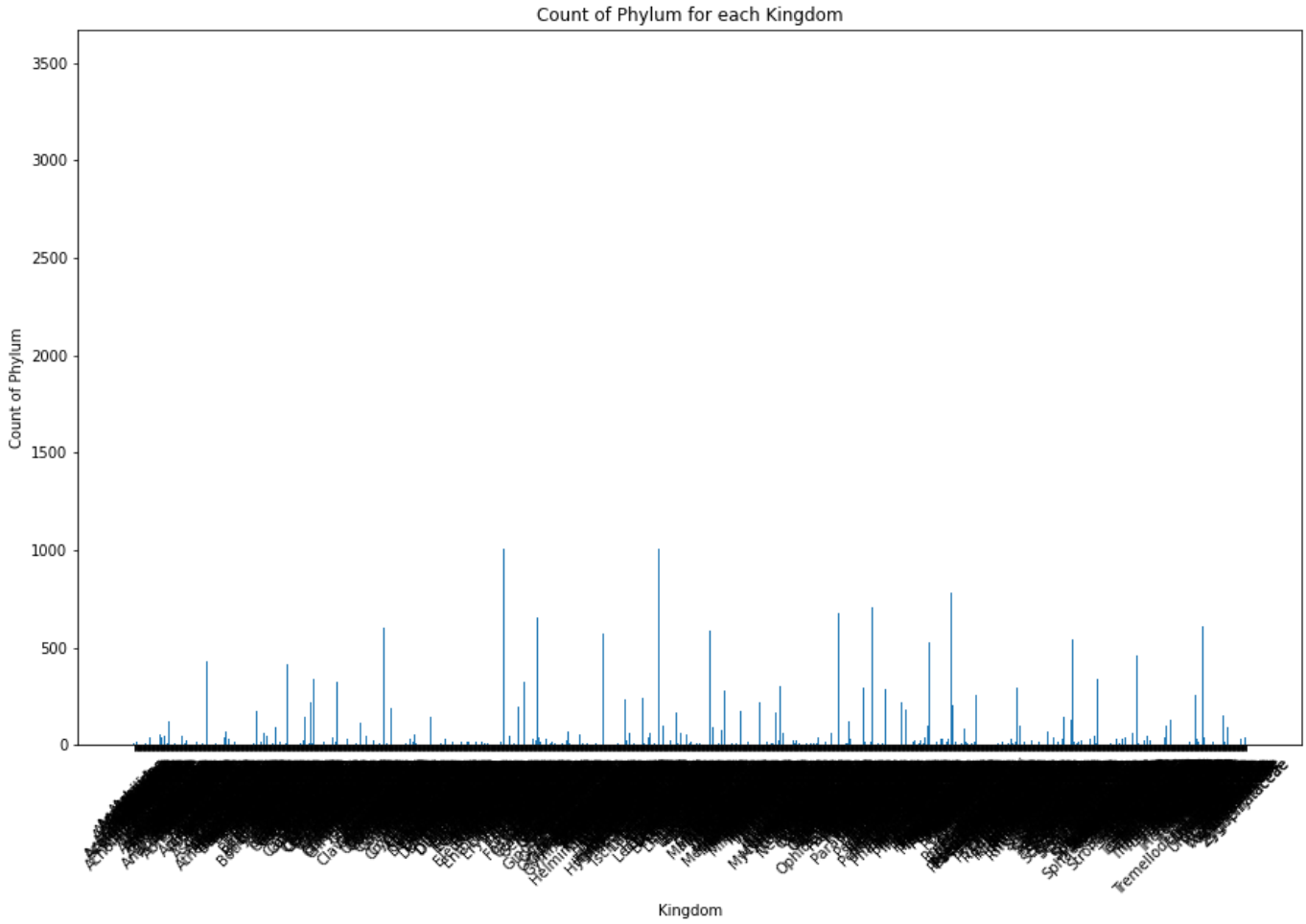
In [55]:
```
grouped_df=df.groupby('order')['family'].count()
plt.figure(figsize=(15, 9))
plt.bar(grouped_df.index, grouped_df.values)
plt.xlabel("Kingdom")
plt.ylabel("Count of Phylum")
plt.title("Count of Phylum for each Kingdom")
plt.xticks(rotation=45)
plt.tight_layout
plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



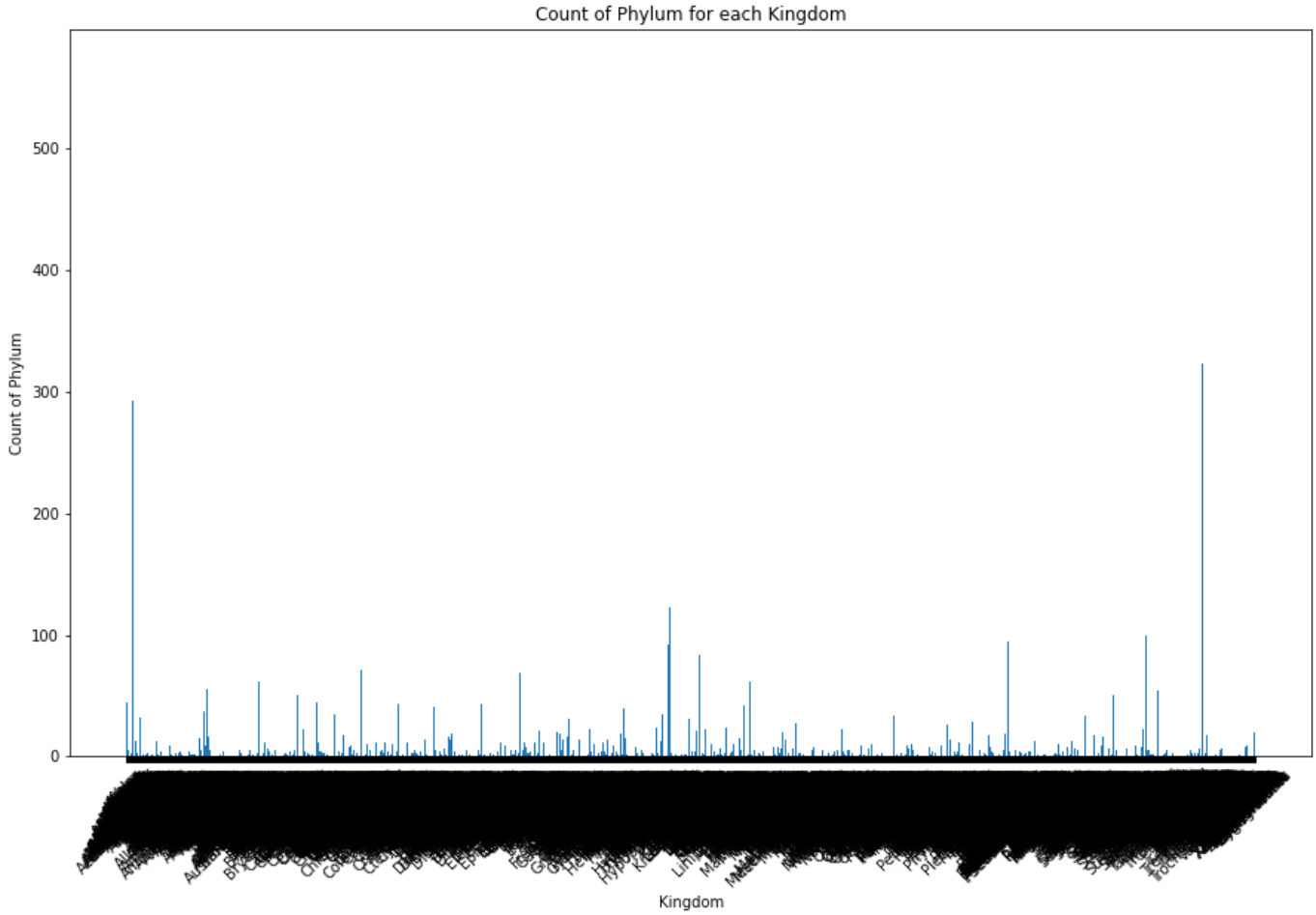
In [56]:
```
grouped_df=df.groupby('family')['genus'].count()
plt.figure(figsize=(15, 9))
plt.bar(grouped_df.index, grouped_df.values)
plt.xlabel("Kingdom")
plt.ylabel("Count of Phylum")
plt.title("Count of Phylum for each Kingdom")
plt.xticks(rotation=45)
plt.tight_layout
plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

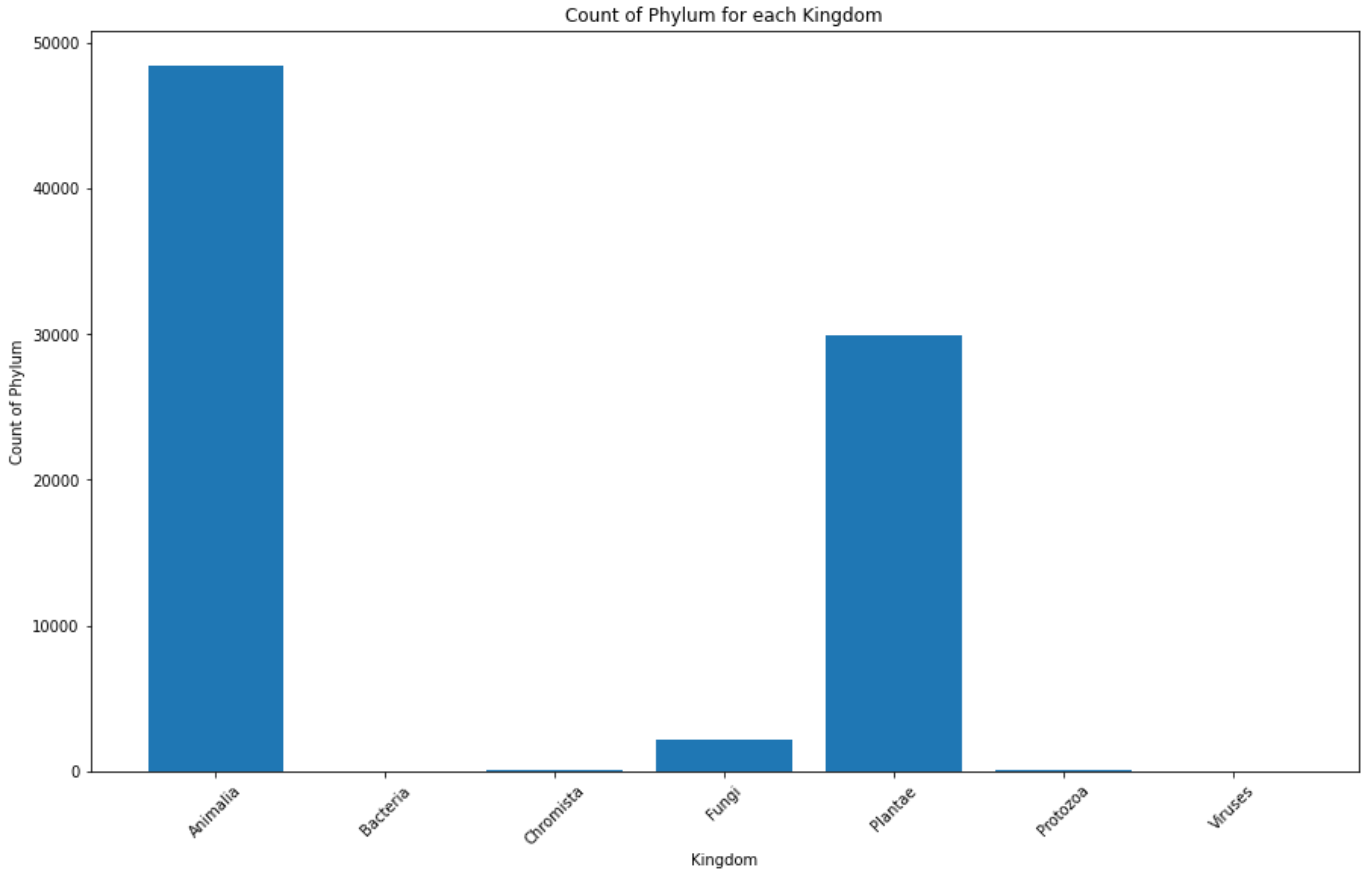


In [57]:
```
grouped_df=df.groupby('genus')['species'].count()
plt.figure(figsize=(15, 9))
plt.bar(grouped_df.index, grouped_df.values)
plt.xlabel("Kingdom")
plt.ylabel("Count of Phylum")
plt.title("Count of Phylum for each Kingdom")
plt.xticks(rotation=45)
plt.tight_layout
plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

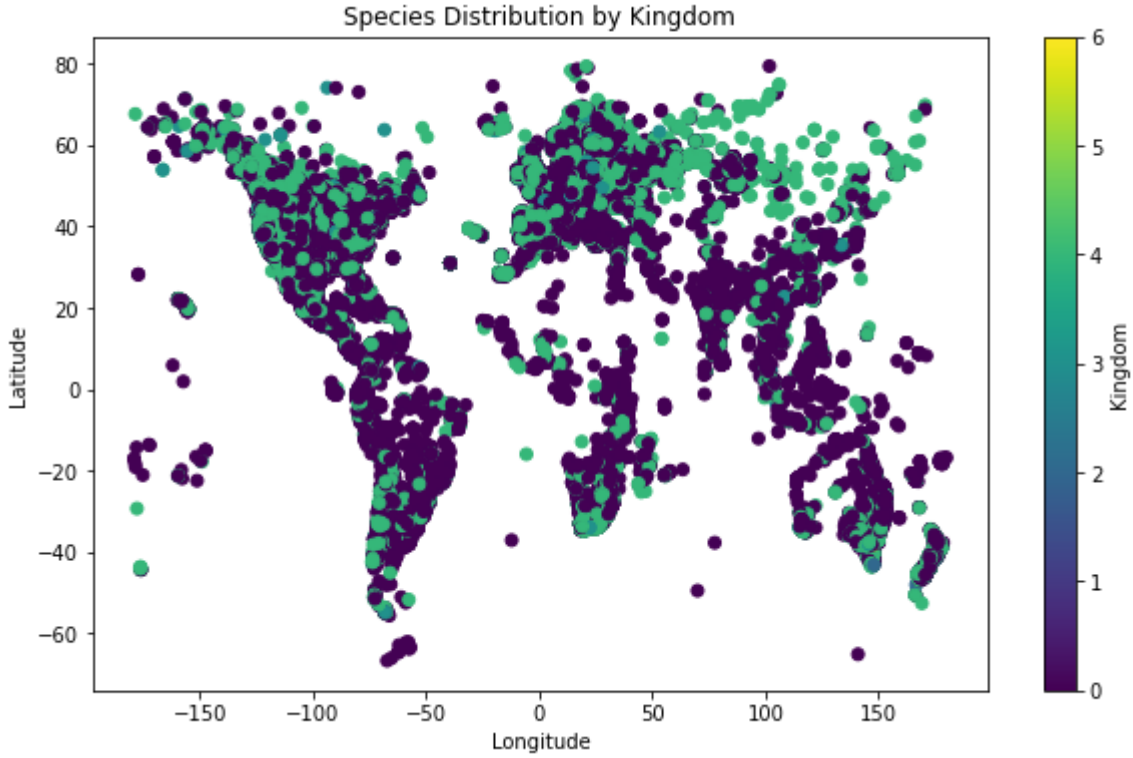


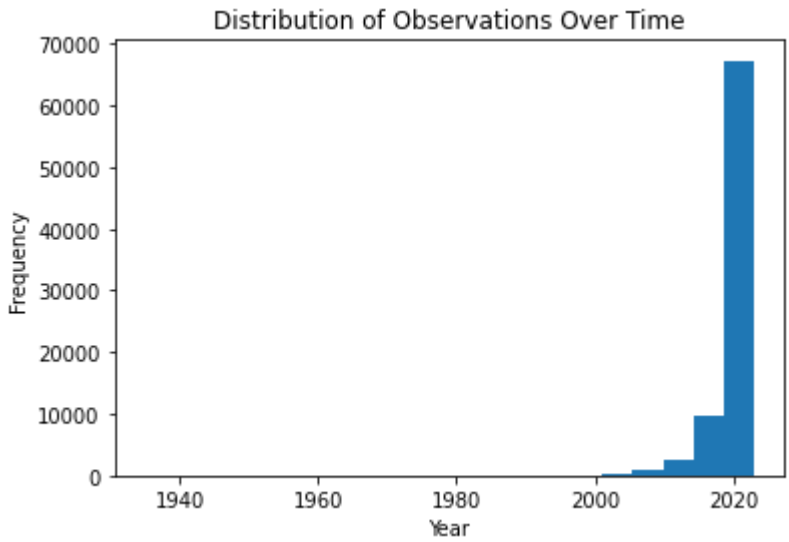
In [58]:
```
grouped_df=df.groupby('kingdom')['countryCode'].count()
plt.figure(figsize=(15, 9))
plt.bar(grouped_df.index, grouped_df.values)
plt.xlabel("Kingdom")
plt.ylabel("Count of Phylum")
plt.title("Count of Phylum for each Kingdom")
plt.xticks(rotation=45)
plt.tight_layout
plt.show()
```



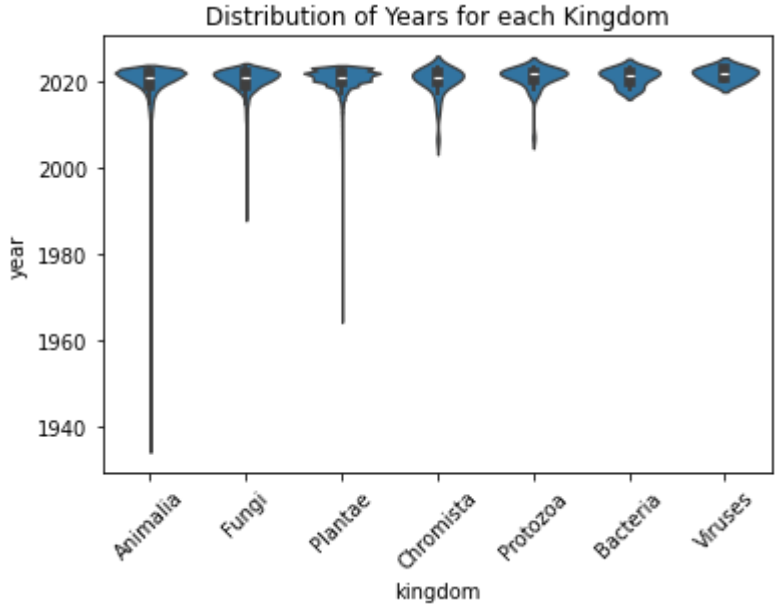
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

In [59]:
```
# Scatter plot example using matplotlib
plt.figure(figsize=(10, 6))
plt.scatter(df['decimalLongitude'], df['decimalLatitude'], c=df['kingdom'].astype('ca
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.title('Species Distribution by Kingdom')
plt.colorbar(ticks=df['kingdom'].astype('category').cat.codes.unique(), label='Kingdo
plt.show()

# Histogram example using matplotlib
plt.hist(df['year'], bins=20)
plt.xlabel('Year')
plt.ylabel('Frequency')
plt.title('Distribution of Observations Over Time')
plt.show()

# Violin plot using seaborn
sns.violinplot(x='kingdom', y='year', data=df)
plt.xticks(rotation=45)
plt.title('Distribution of Years for each Kingdom')
plt.show()
```





Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [60]:
```
df.columns
```

Out[60]:
```
Index(['kingdom', 'phylum', 'class', 'order', 'family', 'genus', 'species',
       'taxonRank', 'scientificName', 'verbatimScientificName', 'countryCode',
       'stateProvince', 'decimalLatitude', 'decimalLongitude',
       'coordinateUncertaintyInMeters', 'eventDate', 'day', 'month', 'year',
       'taxonKey', 'speciesKey', 'catalogNumber', 'identifiedBy',
       'dateIdentified', 'rightsHolder', 'recordedBy', 'lastInterpreted',
       'mediaType', 'issue'],
      dtype='object')
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js