3031_A6_Solutions_v3

November 26, 2021

1 Assignment 6

(50 points total + 2 pts for naming/format)

1.1 Part 1

Read the spotify dataset from the file *spotify_data.csv*.

What percentage of all the unique tracks are contributed by the top 3 artists of each genre, where the top artists are based on *artist_popularity*, and the unique tracks are based on unique values of *track_name*? *(8 points for code)*

A typical approach that will **not** work: If you group the data by genre, and filter the top 3 rows by *artist_popularity*, then you may not get 3 unique artists, as one artist can have multiple tracks.

Here is one way to answer this question:

(1) Group the data by genre, artist name and artist popularity. Find the number of unique tracks (by *track_name*) for each group.

(2) The dataset obtained in (1) is at artist-genre level, i.e., each row corresponds to a unique artist-genre combination. Group that dataset by genre, and filter the top 3 rows of each group based on artist popularity.

(3) Sum up the number of unique tracks of the dataset obtained in (2) and divide it by the total number of unique tracks in the original dataset.

Note: (1) The functions *len()* and *unique()* will be useful.
(2) If you can propose a solution that is shorter than the one proposed above, on Monday - 15th Nov, in class, you will get 10% bonus points for this assignment.

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings
#import seaborn as sns

warnings.filterwarnings('ignore')
#sns.set()
```

[2]:
```
data = pd.read_csv('spotify_data.csv')
```

```
[3]: grouped = data.groupby(['genres','artist_name','artist_popularity'])
```

```
[4]: grouped['track_name'].nunique().reset_index()
```

```
[4]:        genres     artist_name  artist_popularity  track_name
     0      country    Alan Jackson                 76          61
     1      country   Blake Shelton                 77          16
     2      country    Brad Paisley                 71          16
     3      country     Brett Young                 73           5
     4      country   Casey Donahew                 60           3
     …         …            …                    …         …
     11949     rock    Óscar Chávez                 51          33
     11950     rock    İkiye On Kala                61           8
     11951     rock                               52          52
     11952     rock           (!)             58          22
     11953     rock                         57          1

     [11954 rows x 4 columns]
```

```
[5]: artist_genre_lvl=grouped['track_name'].nunique().reset_index()
     top3_artists_data = artist_genre_lvl.groupby('genres').apply(lambda x:x.
      ↪sort_values(by = ['artist_popularity'],ascending = False)[0:3])
     top3_artists_data.track_name.sum()/len(data.track_name.unique())
```

```
[5]: 0.047140401953927644
```

The top 3 artisits of each genre contribute to 5% of the total number of tracks.

1.2 Part 2

Read data from the file "Canadian_Fish_Biodiversity.csv" on Canvas. Each row records a unique fishing event from a 2013 sample of fish populations in Ontario, Canada. *(42 points overall)*

```
[6]: cfbdata = pd.read_csv("Canadian_Fish_Biodiversity.csv")

     cfbdata["Species"].nunique()
```

```
[6]: 132
```

1.2.1 Question 1

To analyze the results of these fishing surveys, we need to understand the dynamics of projects, sites, and geographic locations. In large part the following questions deal with missing data. *(16 points total)*

a) Each site (identified by the column *SITEID*) represents a time and place at which fishing events occurred. Sites are grouped into broader projects (identified by the column *Project Name*). We want to understand the scope of these projects.

Using *.groupby*, find the top three projects by number of unique sites. *(2 points for code)*

Hint: The Pandas function *nunique()* may help

```
[7]: cfbdata.groupby("Project Name")["SITEID"].nunique().
      ↪sort_values(ascending=False).head(3)
```

```
[7]: Project Name
     2013 GLAP Survey of Detroit River             220
     2013 Crown Marsh Survey                       146
     2013 Spotted Gar Critical Habitat Survey      131
     Name: SITEID, dtype: int64
```

b) Find the top three and bottom three projects in terms of the proportion of unique sites of the total number of unique sites. *(3 points for code)*

```
[8]: #top 3
     (cfbdata.groupby("Project Name")["SITEID"].nunique()/\
     cfbdata.groupby("Project Name")["SITEID"].count()).sort_values().head(3)
```

```
[8]: Project Name
     2013 Grass Pickerel Twenty Mile Creek            0.047619
     2013 Mussel Fish Community Assessment            0.056452
     2013 Lake Chubsucker Critical Habitat Survey     0.056572
     Name: SITEID, dtype: float64
```

```
[9]: #bottom 3
     (cfbdata.groupby("Project Name")["SITEID"].nunique()/\
     cfbdata.groupby("Project Name")["SITEID"].count()).sort_values().tail(3)
```

```
[9]: Project Name
     2013 Eastern Sand Darter eDNA Survey of Sydenham River    1.0
     2013 Eastern Sand Darter eDNA Survey of Grand River       1.0
     2013 Spotted Gar eDNA Survey                              1.0
     Name: SITEID, dtype: float64
```

c) (i) How many values are missing for the air temperature column? *(1 point for code)*

```
[10]: cfbdata['Air Temperature (C)'].isnull().sum()
```

```
[10]: 808
```

(i) 808 values are missing for the air temperature column.

(ii) Impute the missing values of air temperature with the median air temperature of the corresponding water body (*Waterbody Name*) and month. *(2 points for code)*

(ii)

```
[11]: cfbdata["Air Temperature (C)"] = cfbdata.groupby(["Waterbody␣
       ↪Name",'Month'])["Air Temperature (C)"].apply(lambda x:x.fillna(x.median()))
```

(iii) How many missing values still remain for the air temperature column after the imputation in (ii)? *(1 point for answer)*

```
[12]: cfbdata["Air Temperature (C)"].isnull().sum()
```

```
[12]: 113
```

(iii) 113 missing values still remain after the imputation in (ii)

(iv) We will try to impute the remaining missing values for air temperature. Try impute the remaining missing values of air temperature with the median air temperature of the corresponding project (*Project Name*) and month. *(2 points for code)*

(iv)

```
[13]: cfbdata["Air Temperature (C)"] = cfbdata.groupby(["Project Name",'Month'])["Air␣
       ↪Temperature (C)"].apply(lambda x:x.fillna(x.median()))
```

(v) How many missing values still remain for the air temperature column after the imputation in (iv)? *(1 point for answer)*

```
[14]: cfbdata["Air Temperature (C)"].isnull().sum()
```

```
[14]: 62
```

(v) 62 missing values still remain after the imputation in (iv)

(vi) Find the correlation between air temperature and water temperature. *(1 point for code)*

(vi)

```
[15]: cfbdata["Air Temperature (C)"].corr(cfbdata["Water Temperature (C)"])
```

```
[15]: 0.768184572633517
```

Correlation = 77%

(vii) As you found a high correlation between air temperature and water temperature in (vi), you can use water temperature to estimate the air temperature (using the trendline, like you did in assignment 4). Assuming you already did that, how many missing values will still remain for the air temperature column? *Note: Do not impute the missing values using the trendline, just assume you already did that. (1 point for code)*

(vii) The values for air temperature will remain missing for those observations that have missing values of water temperature.

```
[16]: (cfbdata['Air Temperature (C)'].isnull() & cfbdata['Water Temperature (C)'].
       ↪isnull()).sum()
```

```
[16]: 11
```

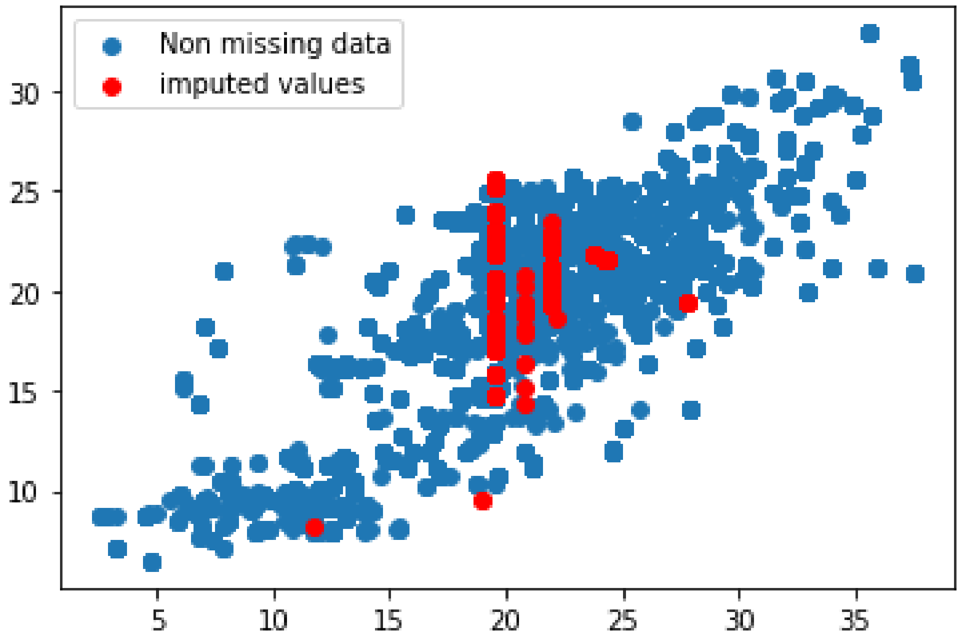
11 missing values will still remain after imputing the missing values using the trendline.

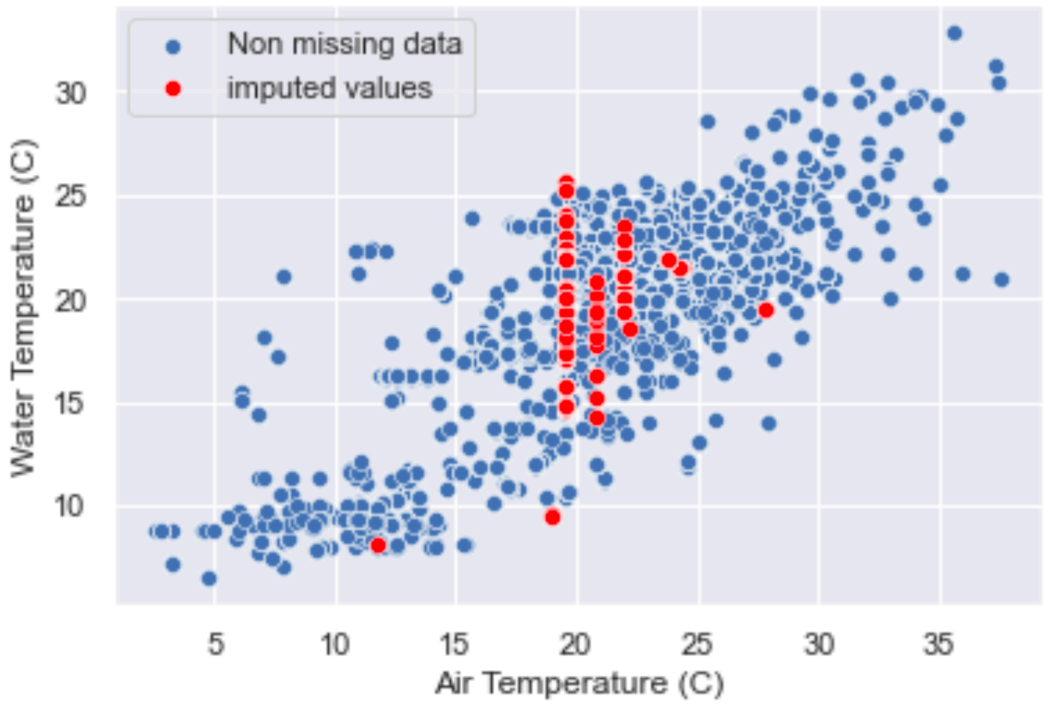
(viii) Make a scatterplot of air temperature against water temperature. Higlight the points for which the air temperature was imputed in (ii) and (iv) with a different color. *(2 points for code)*

(viii)

```
[17]: cfbdata_orig = pd.read_csv("Canadian_Fish_Biodiversity.csv")
```

```
[18]: ind_imputed = np.where(cfbdata_orig['Air Temperature (C)'].isnull() &␣
       ↪~cfbdata['Air Temperature (C)'].isnull())[0]
      plt.scatter(data = cfbdata_orig, x = 'Air Temperature (C)', y = 'Water␣
       ↪Temperature (C)')
      plt.scatter(data = cfbdata.iloc[ind_imputed,:], x = 'Air Temperature (C)', y =␣
       ↪'Water Temperature (C)',color = 'red')
      plt.legend(labels = ['Non missing data','imputed values'])
      plt.show()
```





1.2.2 Question 2

This section begins to investigate the living conditions of fish at different locations and time periods. *(7 points total)*

a) Use a single *.groupby* statement to view the minimum, mean, standard deviation, and maximum air temperature and water temperature for each project during the month of August (use the *Month* column). *(2 points for code)*

```
[19]: cfb_month = cfbdata[cfbdata["Month"]==8]
      cfb_month.groupby("Project Name")[["Air Temperature (C)","Water Temperature␣
       ↪(C)"]].agg(['min','mean','std','max'])
```

```
[19]:                                               Air Temperature (C)            \
                                                                min       mean
      Project Name
      2013 Bridle Shiner Critical Habitat Survey               20.7  24.609091
      2013 Crown Marsh Survey                                  16.4  21.673275
      2013 GLAP Survey of Detroit River                        21.1  24.360619
      2013 Grass Pickerel Niagara Drains                       20.8  25.246154
      2013 Grass Pickerel Twenty Mile Creek                    22.5  25.226190
      2013 Lake Chubsucker Critical Habitat Survey             14.3  21.136106
      2013 Mussel Fish Community Assessment                    23.0  24.535887
      2013 Pugnose Minnow Lake St Clair Drains                 22.2  25.303061
      2013 Species at Risk Assessment                          23.3  24.793939
      2013 Spotted Gar Critical Habitat Survey                 18.6  22.706481
```

```
                                                                  \
                                                 std   max
Project Name
2013 Bridle Shiner Critical Habitat Survey    2.588098  26.5
2013 Crown Marsh Survey                       2.040997  26.2
2013 GLAP Survey of Detroit River             2.555656  28.3
2013 Grass Pickerel Niagara Drains            2.498615  28.2
2013 Grass Pickerel Twenty Mile Creek         2.283649  29.4
2013 Lake Chubsucker Critical Habitat Survey  3.427318  32.6
2013 Mussel Fish Community Assessment         1.203286  26.1
2013 Pugnose Minnow Lake St Clair Drains      1.961982  29.0
2013 Species at Risk Assessment               0.559087  25.0
2013 Spotted Gar Critical Habitat Survey      2.590503  27.7

                                             Water Temperature (C)            \
                                                               min       mean
Project Name
2013 Bridle Shiner Critical Habitat Survey                  20.630  22.052182
2013 Crown Marsh Survey                                     17.430  22.007084
2013 GLAP Survey of Detroit River                           21.646  22.028226
2013 Grass Pickerel Niagara Drains                          17.150  20.105641
2013 Grass Pickerel Twenty Mile Creek                       18.990  22.380238
2013 Lake Chubsucker Critical Habitat Survey                19.300  21.671746
2013 Mussel Fish Community Assessment                       21.800  23.155645
2013 Pugnose Minnow Lake St Clair Drains                    18.970  20.713163
2013 Species at Risk Assessment                             21.960  22.075600
2013 Spotted Gar Critical Habitat Survey                    18.639  21.349769


                                                 std    max
Project Name
2013 Bridle Shiner Critical Habitat Survey    0.503887  22.44
2013 Crown Marsh Survey                       1.240151  23.24
2013 GLAP Survey of Detroit River             0.392402  23.11
2013 Grass Pickerel Niagara Drains            2.426792  23.96
2013 Grass Pickerel Twenty Mile Creek         2.533686  26.87
2013 Lake Chubsucker Critical Habitat Survey  1.619878  25.13
2013 Mussel Fish Community Assessment         1.156345  24.60
2013 Pugnose Minnow Lake St Clair Drains      1.676571  24.22
2013 Species at Risk Assessment               0.080936  22.13
2013 Spotted Gar Critical Habitat Survey      1.978877  24.30
```

b) Make lineplots showing maximum air temperature and water temperature by month and *Region*. To construct *Region*, use *pd.cut* to satisfy the following conditions:

- Rows with a latitude lower than 42.4 should have *Southern* in the *Region* column
- Rows with a latitude between 42.4 and 42.8 should have *Central* in the *Region* column

- Rows with a latitude higher than 42.8 should have *Northern* in the *Region* column

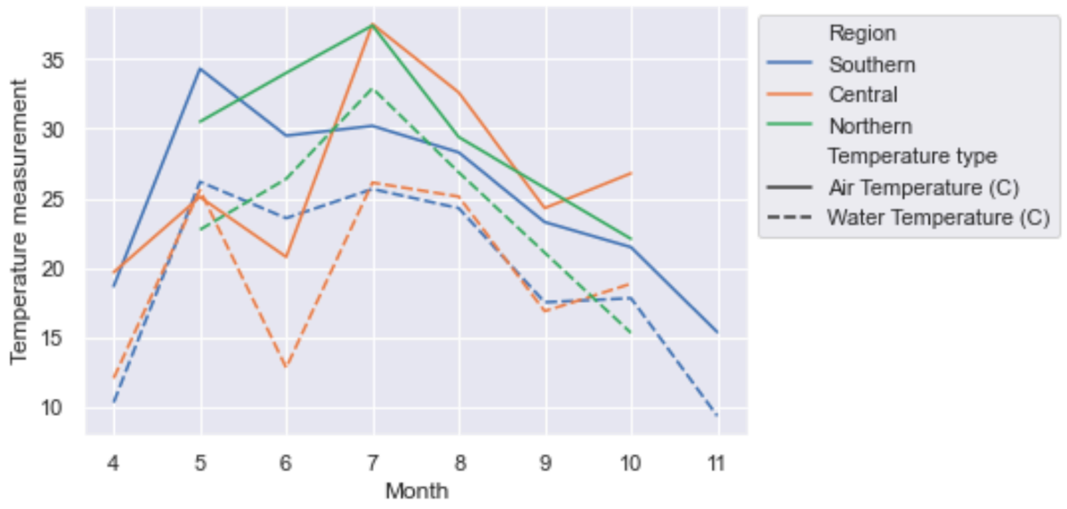
You can have the month on the horizontal axis, the temperature on the vertical axis, different colors for different regions, and different styles (solid line / dotted line) to indicate air/water temperature.

Does anything in the visualization surprise you? Why or why not? *(4 points for code and visualization, 1 point for answer)*

[20]:
```
lat_bins = [40,42.4,42.8,46]
lat_names = ["Southern","Central","Northern"]

cfbdata["Region"] = pd.cut(cfbdata["Start Latitude"],lat_bins,labels=lat_names)
```

[21]:
```
datag = cfbdata.groupby(['Region','Month'],as_index = False)[['Air Temperature␣
↪(C)','Water Temperature (C)']].max()
data_melt = pd.melt(datag, id_vars = ['Region','Month'],var_name = 'Temperature␣
↪type',value_name = 'Temperature measurement')
#sns.boxplot(data = data_melt, x = 'Month',y = 'Temperature measurement', hue =␣
↪'Region',style = 'Temperature type')
#plt.legend(bbox_to_anchor = (1,1))
```



(Sample answer – any reasonable interpretation is acceptable)

I'm surprised that both water and air maximum temperatures in the Northern region are higher than other regions' in June and July—I would've expected bodies of water further north to be generally colder. I'm also surprised that maximum temperatures in the Central region increase from September to October.

1.2.3 Question 3

Finally let's focus on the stars of this survey—the fish, of course. *(19 points total)*

a) Let's continue using our *Region* categorization. Find the top three fish species in each region by number captured. *(3 points for code)*

```
[22]: top3caught = lambda c: c.sort_values(by="Number Captured",ascending=False).
       ↪head(3)

      (cfbdata.groupby(["Region","Species"],as_index=False)["Number Captured"]\
       .sum()).groupby("Region", as_index=False).apply(top3caught)
```

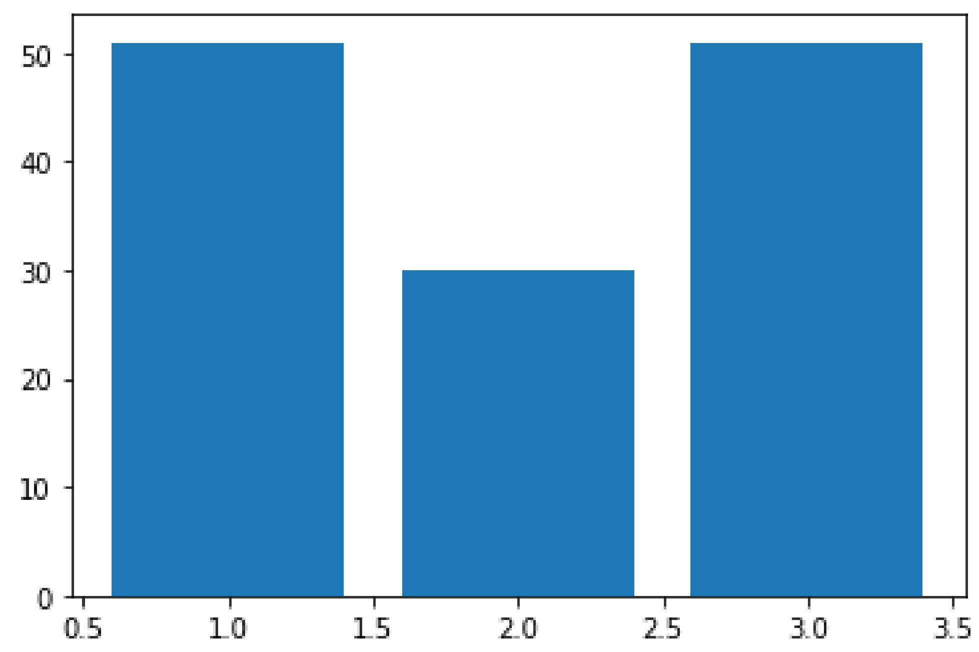
```
[22]:          Region                  Species  Number Captured
      0 63   Southern      Lepomis macrochirus           5072.0
        26   Southern      Dorosoma cepedianum           3559.0
        82   Southern   Neogobius melanostomus           2265.0
      1 195   Central      Lepomis macrochirus           2126.0
        186   Central     Labidesthes sicculus           2029.0
        223   Central       Notropis heterodon           1562.0
      2 346  Northern   Neogobius melanostomus           2522.0
        362  Northern      Notropis volucellus           2104.0
        327  Northern      Lepomis macrochirus           1841.0
```

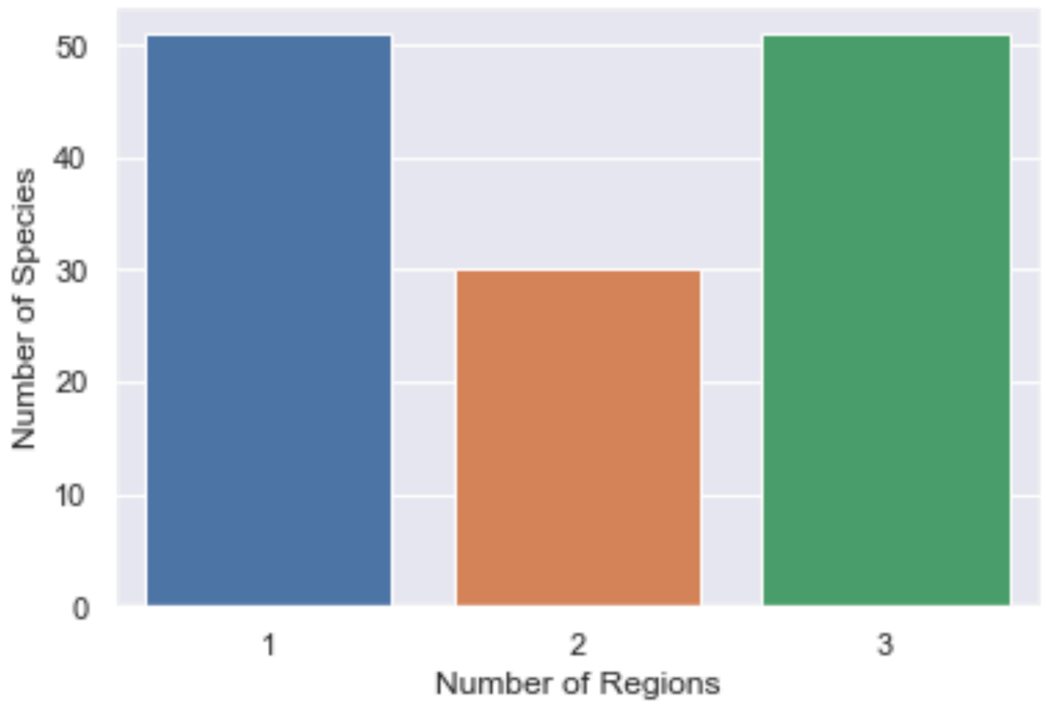
b) Are certain fish only found in some regions? Visualize how many species are in all three regions, how many are in two of three, and how many were only captured in one region. *(3 points for code and visualization)*

```
[23]: region_num = pd.DataFrame(cfbdata.groupby("Species",as_index=False)["Region"].
       ↪nunique()["Region"].value_counts()).reset_index()
      region_num.columns = ["Number of Regions","Number of Species"]
      region_num
```

```
[23]:    Number of Regions  Number of Species
      0                  3                 51
      1                  1                 51
      2                  2                 30
```

```
[24]: plt.bar(x="Number of Regions",height="Number of Species",data=region_num)
      plt.show()
```





c) What percentage of all species are exclusively captured in the Southern region? How about the Northern Region? And the Central region? *(3 points for code)*

[25]:
```
#Method 1
exclusive_species = pd.DataFrame(cfbdata.
 ↪groupby("Species",as_index=False)["Region"].nunique())
exclusive_species = exclusive_species[exclusive_species["Region"]==1]

exclusive_species.head()
```

[25]:
```
                                 Species  Region
5                             Ameiurus sp       1
10  Carassius auratus X Cyprinus carpio       1
12                           Catostomidae       1
16                          Chrosomus eos       1
17                             Clupeidae       1
```

[26]:
```
cfbdata["Species"].nunique()
```

[26]:
```
132
```

[27]:
```
exclusive_obs = cfbdata[cfbdata["Species"].isin(exclusive_species["Species"])]
exclusive_obs.groupby("Region")["Species"].nunique() #/cfbdata["Species"].
 ↪nunique()
```

[27]:
```
Region
Southern    12
Central      6
Northern    33
Name: Species, dtype: int64
```

[28]:
```
#Method 2:
dp = cfbdata.pivot_table(index = 'Species', columns = 'Region', values =␣
 ↪'Number Captured',aggfunc = 'sum',margins = True)
for reg in ['Northern','Southern','Central']:
    print('Region: ',reg,":",((dp[reg] == dp['All']) & (dp[reg]>0)).sum()/
 ↪cfbdata.Species.nunique())
```

```
Region:  Northern : 0.25
Region:  Southern : 0.09090909090909091
Region:  Central : 0.045454545454545456
```

d) Turbidity quantifies the level of cloudiness in liquid. For fish in each of the three regions, is there a correlative relationship between turbidity and # of fish caught? *(2 points for code, 1 point for answer)*

[29]:
```
cfbdata.groupby('Region').apply(lambda x:x['Turbidity (ntu)'].corr(x['Number␣
 ↪Captured']))
```

[29]:
```
Region
Southern   -0.019202
```

```
Central    -0.016327
Northern    0.063456
dtype: float64
```

No, there does not appear to be a correlation.

e) Now let's turn to the length of fish captured, given by *Maximum (mm)* and *Minimum (mm)*. Find the overall maximum and minimum lengths of all fish in each region. Which region has the largest range in captured fish length? *(2 points for code, 1 point for answer)*

[30]:
```
cfbdata.groupby("Region").agg({"Maximum (mm)": 'max', "Minimum (mm)" : 'min'})
```

[30]:
```
          Maximum (mm)  Minimum (mm)
Region
Southern        1130.0           8.0
Central          785.0           9.0
Northern         760.0          10.0
```

The Southern region has the largest range in captured fish length.

f) Find the inverse Simpson index of species counts for each waterbody type (*WaterbodyType*) within each region. Which combination of waterbody type and region has the greatest diversity of fish species? Which has the least?

The inverse Simpson index ($\frac{1}{\lambda}$) is a measure of ecological diversity, for which a larger index number indicates a greater diversity of species. The index is calculated as:

$\frac{1}{\lambda} = 1/(\sum_{i=1}^{R} p_i^2)$

where R is the number of unique species and p_i is the proportion of fish belonging to species i. *(3 points for code, 1 point for answer)*

[31]:
```
#method 1:
wrs_captured = cfbdata.
 ↪groupby(["WaterbodyType","Region","Species"],as_index=False)["Number␣
 ↪Captured"].sum()
wrs_captured = wrs_captured[wrs_captured["Number Captured"]>0]

inv_simpson = lambda s: 1/(np.sum((s/np.sum(s))**2))

wrs_captured.groupby(["WaterbodyType","Region"])["Number Captured"].
 ↪apply(inv_simpson)
```

[31]:
```
WaterbodyType  Region
Lake           Southern     4.094649
               Central      3.854458
               Northern     4.896119
Stream         Southern    13.800772
               Central     12.070477
               Northern    20.306997
Wetland        Southern          NaN
```

```
                 Central      8.954213
                 Northern          NaN
Name: Number Captured, dtype: float64
```

[32]:
```
#Method 2:
def f(x):
    R = x.Species.nunique()
    all_num = x['Number Captured'].sum()
    all_species = x.Species.unique()
    si=0
    for i in all_species:
        icount = x.loc[x['Species']==i,'Number Captured'].sum()
        if icount>0:
            si = si+((icount/all_num)**2)
    return 1/si
```

[33]:
```
cfbdata.groupby(['WaterbodyType','Region']).apply(f)
```

[33]:
```
WaterbodyType  Region
Lake           Southern     4.094649
               Central      3.854458
               Northern     4.896119
Stream         Southern    13.800772
               Central     12.070477
               Northern    20.306997
Wetland        Central      8.954213
dtype: float64
```

According to the inverse Simpson index, Northern streams have the greatest diversity in fish species and Central lakes have the least. The survey didn't include any fish in Southern or Northern wetlands, so we can't describe these bodies of water in terms of fish diversity.