Bird Migration Group Project

<u>Group Members:</u> Veronika Kermoshchuk, Maria Coughlin, Sara Conway, Caitlin Larkin

In [ ]:
```
#!pip install geopy --user codio
#!pip3 install haversine --user
!pip install geopy --user
!pip install haversine --user
plt.offline.init_notebook_mode (connected = True)
```

Importing Libraries: ¶

In [ ]:
```
# Importing libraries
import pandas as pd
import matplotlib.pyplot as plt
from plotly import tools
import chart_studio.plotly
from plotly.offline import init_notebook_mode, iplot
init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.figure_factory as ff
from IPython.display import HTML, Image
import plotly.express as px
px.set_mapbox_access_token(open(".mapbox_token").read())
from geopy.geocoders import Nominatim
import ipywidgets as widgets
from ipywidgets import interact, interact_manual
import haversine as hs
import plotly as plt
```

Data Set:

In [ ]:
```
birds = pd.read_csv("bird_tracking.csv")
birds = birds.assign(date_time = lambda d: pd.to_datetime(d['date_time'], form
at = "%Y-%m-%d"))
birds['month'] = pd.to_datetime(birds['date_time']).dt.month
birds['hour'] = pd.to_datetime(birds['date_time']).dt.hour
birds.head()
```

Eric:

In [ ]:
```
Eric = birds[birds['bird_name'] == "Eric"].reset_index()
Eric.head(2)
```

Nico:

In [ ]:
```
Nico = birds[birds['bird_name'] == "Nico"].reset_index()
Nico.head(2)
```

Sanne:

In [ ]:
```
Sanne = birds[birds['bird_name'] == "Sanne"].reset_index()
Sanne.reset_index().head(2)
```

Migration Pattern for Birds

In [ ]:
```
bird_names = list(birds.bird_name.unique())
bird_names.append("ALL")
```

In [ ]:
```
@interact(birdName = bird_names)
def scatter_by_features(birdName):
    if birdName == "ALL":
        selected_df = birds
    else:
        selected_df = birds[birds.bird_name == birdName]
    px.set_mapbox_access_token(open(".mapbox_token").read())
    fig = px.scatter_mapbox(selected_df, lat="latitude", lon="longitude", colo
r = "bird_name",
                  color_continuous_scale=px.colors.cyclical.IceFire, size_max=
5, zoom=3)
    fig.show()
```

For this project we used:

Departure month: August
Arrival month: April
Winter month: January/February

~

Migration Path of Eric

In [ ]:
```
fig = px.scatter_mapbox(Eric, lat="latitude", lon="longitude", color = "bird_n
ame",
                  color_continuous_scale=px.colors.cyclical.IceFire, size_max=
5, zoom=4)
fig.show()
```

Eric's Starting location:

In [ ]:
```
Eric_lat_start = str(Eric['latitude'].iloc[0])
Eric_lon_start = str(Eric['longitude'].iloc[0])

geolocator = Nominatim(user_agent = 'geoapiExercises')
location = geolocator.reverse(Eric_lat_start+","+Eric_lon_start)
address = location.raw['address']
country = address.get('country', '')
country
```

Eric's starting location is France

Eric's location in January:

In [ ]:
```
winter = Eric[Eric['month'] == 1].reset_index()
```

In [ ]:
```
Eric_lat_winter = str(winter['latitude'].iloc[0])
Eric_lon_winter = str(winter['longitude'].iloc[0])

geolocator = Nominatim(user_agent = 'geoapiExercises')
location = geolocator.reverse(Eric_lat_winter+","+Eric_lon_winter)
address = location.raw['address']
country = address.get('country', '')
country
```

> **Eric's location during the winter month January is Morocco**

Eric's ending location after migration:

In [ ]:
```
Eric_lat_end = str(Eric['latitude'].iloc[Eric.shape[0]-1])
Eric_lon_end = str(Eric['longitude'].iloc[Eric.shape[0]-1])

geolocator = Nominatim(user_agent = 'geoapiExercises')
location = geolocator.reverse(Eric_lat_end+","+Eric_lon_end)
address = location.raw['address']
country = address.get('country', '')
country
```

> **Eric's ending location is Belgium after the migration**

At the beginning Eric is located in France. Then during the cold months he migrates and stays at Morocco. Finally he returns in April and settles in Belgium after his migration.

Eric's first location in January

We used Eric's first location in January so that we can have one of the possible points of location where he is in Morocco.

In [ ]:
```
first_loc = Eric[Eric['month'] == 1].head(1)
first_loc
```

The approximate distances that Eric traveled

In [ ]:
```
loc1 = (float(Eric_lat_start), float(Eric_lon_start))
loc2 = (first_loc['latitude'].iloc[0], first_loc['longitude'].iloc[0])
to_dist = hs.haversine(loc1,loc2)
print("The distance that Eric traveled from France to Morocco is about:", roun
d(hs.haversine(loc1,loc2),2), "KM")
```

In [ ]:
```
loc1 = (first_loc['latitude'].iloc[0], first_loc['longitude'].iloc[0])
loc2 = (float(Eric_lat_end), float(Eric_lon_end))
from_dist = hs.haversine(loc1,loc2)
print("The distance that Eric traveled from Morocco to Belgium is about:", rou
nd(hs.haversine(loc1,loc2),2), "KM")
```

In [ ]:
```
total_dist_Eric = to_dist + from_dist
print("The total distance that Eric traveled according to this dataset is abou
t:", round(total_dist_Eric,2), "KM")
```

~ ~

Migration Path of Nico

In [ ]:
```
fig = px.scatter_mapbox(Nico, lat="latitude", lon="longitude", color = "bird_n
ame",
                  color_continuous_scale=px.colors.cyclical.IceFire, size_max=
5, zoom=2)
fig.show()
```

Nico's Starting location:

In [ ]:
```
Nico_lat_start = str(Nico['latitude'].iloc[0])
Nico_lon_start = str(Nico['longitude'].iloc[0])

geolocator = Nominatim(user_agent = 'geoapiExercises')
location = geolocator.reverse(Nico_lat_start+","+Nico_lon_start)
address = location.raw['address']
country = address.get('country', '')
country
```

Nico's starting location is Belgium

Nico's location in February:

In [ ]:
```
winter = Nico[Nico['month'] == 2].reset_index()
```

In [ ]:
```
Nico_lat_winter = str(winter['latitude'].iloc[0])
Nico_lon_winter = str(winter['longitude'].iloc[0])

geolocator = Nominatim(user_agent = 'geoapiExercises')
location = geolocator.reverse(Nico_lat_winter+","+Nico_lon_winter)
address = location.raw['address']
country = address.get('country', '')
country
```

Nico's location during the winter month February is Senegal, Africa

Nico's ending location after migration:

In [ ]:
```
Nico_lat_end = str(Nico['latitude'].iloc[Nico.shape[0]-1])
Nico_lon_end = str(Nico['longitude'].iloc[Nico.shape[0]-1])

geolocator = Nominatim(user_agent = 'geoapiExercises')
location = geolocator.reverse(Nico_lat_end+","+Nico_lon_end)
address = location.raw['address']
country = address.get('country', '')
country
```

Nico's ending location is Belgium after migration

At the beginning Nico is located in Belgium. Then during the cold months he migrates and stays at Senegal. Finally he returns in April and settles in Belgium after his migration.

Nico's first location in February

In [ ]:
```
first_loc_Nico = Nico[Nico['month'] == 2].head(1)
first_loc_Nico
```

The approximate distances that Nico traveled

In [ ]:
```
loc1 = (float(Nico_lat_start), float(Nico_lon_start))
loc2 = (first_loc_Nico['latitude'].iloc[0], first_loc_Nico['longitude'].iloc
[0])
to_dist_Nico = hs.haversine(loc1,loc2)
print("The distance that Nico traveled from Belgium to Senegal is about:", rou
nd(hs.haversine(loc1,loc2),2), "KM")
```

In [ ]:
```
loc1 = (first_loc_Nico['latitude'].iloc[0], first_loc_Nico['longitude'].iloc
[0])
loc2 = (float(Nico_lat_end), float(Nico_lon_end))
from_dist_Nico = hs.haversine(loc1,loc2)
print("The distance that Nico traveled from Senegal to Belgium is about:", rou
nd(hs.haversine(loc1,loc2),2), "KM")
```

In [ ]:
```
total_dist_Nico = to_dist_Nico + from_dist_Nico
print("The total distance that Nico traveled according to this dataset is abou
t:", round(total_dist_Nico,2), "KM")
```

~ ~

Migration Path of Sanne

In [ ]:
```
fig = px.scatter_mapbox(Sanne, lat="latitude", lon="longitude", color = "bird_
name",
                  color_continuous_scale=px.colors.cyclical.IceFire, size_max=
5, zoom=2)
fig.show()
```

Sanne's Starting location:

In [ ]:
```
Sanne_lat_start = str(Sanne['latitude'].iloc[0])
Sanne_lon_start = str(Sanne['longitude'].iloc[0])

geolocator = Nominatim(user_agent = 'geoapiExercises')
location = geolocator.reverse(Sanne_lat_start+","+Sanne_lon_start)
address = location.raw['address']
country = address.get('country', '')
country
```

Sanne's starting location is France

Sanne's location in January:

In [ ]:
```
winter = Sanne[Sanne['month'] == 1].reset_index()
```

In [ ]:
```
Sanne_lat_winter = str(winter['latitude'].iloc[0])
Sanne_lon_winter = str(winter['longitude'].iloc[0])

geolocator = Nominatim(user_agent = 'geoapiExercises')
location = geolocator.reverse(Sanne_lat_winter+","+Sanne_lon_winter)
address = location.raw['address']
country = address.get('country', '')
country
```

Sanne's location during the winter month January is Senegal, Africa

Sanne's ending location after migration:

In [ ]:
```
Sanne_lat_end = str(Sanne['latitude'].iloc[Sanne.shape[0]-1])
Sanne_lon_end = str(Sanne['longitude'].iloc[Sanne.shape[0]-1])

geolocator = Nominatim(user_agent = 'geoapiExercises')
location = geolocator.reverse(Sanne_lat_end+","+Sanne_lon_end)
address = location.raw['address']
country = address.get('country', '')
country
```

Sanne's ending location is Belgium after migration

At the beginning Sanne is located in France. Then during the cold months he migrates and stays at Senegal. Finally he returns in April and settles in Belgium after his migration.

Sanne's first location in January

In [ ]:
```
first_loc_Sanne = Sanne[Sanne['month'] == 1].head(1)
first_loc_Sanne
```

The approximate distances that Sanne traveled

In [ ]:
```
loc1 = (float(Sanne_lat_start), float(Sanne_lon_start))
loc2 = (first_loc_Sanne['latitude'].iloc[0], first_loc_Sanne['longitude'].iloc
[0])
to_dist_Sanne = hs.haversine(loc1,loc2)
print("The distance that Sanne traveled from France to Senegal is about:", rou
nd(hs.haversine(loc1,loc2),2), "KM")
```

In [ ]:
```
loc1 = (first_loc_Sanne['latitude'].iloc[0], first_loc_Sanne['longitude'].iloc
[0])
loc2 = (float(Sanne_lat_end), float(Sanne_lon_end))
from_dist_Sanne = hs.haversine(loc1,loc2)
print("The distance that Sanne traveled from Senegal to Belgium is about:", ro
und(hs.haversine(loc1,loc2),2), "KM")
```

In [ ]:
```
total_dist_Sanne = to_dist_Sanne + from_dist_Sanne
print("The total distance that Sanne traveled according to this dataset is abo
ut:", round(total_dist_Sanne,2), "KM")
```

~ ~

Speed Differences Across All Birds

In [ ]:
```
bird_mean_speed = birds.groupby('bird_name')['speed_2d'].mean().to_frame()
bird_mean_speed
```

Based on the information, Eric is the slowest bird since he has the lowest average speed compared to all the other birds. On the other hand Nico seems to be the fastest bird of all since he has the highest average speed.

~ ~

Total Distance Traveled Across All Birds

In [ ]:
```
print("Total distance traveled:")
print()
print("Eric:", round(total_dist_Eric,2))
print("Nico:", round(total_dist_Nico,2))
print("Sanne:", round(total_dist_Sanne,2))
```

~ ~

Summary of the Results

Using the geopy library, our team was able to locate each birds location based on their longitude and latitude coordinates. Using this information, we focused on following the flight path and migration pattern for every bird. In addition to this information, our team also discovered the average speed of each bird and the total distance that each of them traveled.

Eric

> Our team uncovered that Eric started his migration in France, flew to Morrocco during winter, and settled in Belgium after his migration in the spring.
>
> The total distance of these three locations was also the shortest compared to the other two birds. Eric only traveled a total of 4884.99 KM. This is because the distance that Eric traveled from France to Morocco is about 2331.25 KM and the distance Eric traveled from Morocco to Belgium is about 2553.73 KM.
>
> Furthermore, Eric proved to be the slowest bird as his average speed was only 2.300545 km/h.
>
> It is interesting to note that Eric is the only bird that migrated to Morocco.

Nico

> Our team uncovered that Nico started his migration in Belgium, flew to Senegal, Africa, during winter, and returned to Belgium after his migration in the spring.
>
> The total distance of these three locations was also the second farthest compared to the other two birds. Nicco traveled a total of 8543.76 KM. This is because the distance that Nicco traveled from Belgium to Senegal, Africa, is about 4241.48 KM and the distance Nicco traveled from Senegal, Africa, to Belgium is about 4302.27 KM.
>
> Furthermore, Nicco proved to be the fastest bird as his average speed was 2.908726 km/h.
>
> It is interesting to note that Nico's migration route varied both times from his migration from Belgium to Senegal, Africa, and back.

Sanne

Our team uncovered that Sanne started his migration in France, flew to Senegal, Africa, during winter, and then flew to Belgium after his migration in the spring.

The total distance of these three locations was also the farthest compared to the other two birds. Sanne traveled a total of 8855.28 KM. This is because the distance that Sanne traveled from France to Senegal, Africa, is about 4343.33 KM and the distance Sanne traveled from Senegal, Africa, to Belgium is about 4511.95 KM.

Furthermore, Sanne's average speed was neither the fastest, nor slowest, as his average speed was 2.450434 km/h.

It is interesting to note that Sanne appears to fly a migration route that travels the most over ocean water compared to the other two birds.