Bird Migration Project

Sophia Colonello, Patrick Dolan, Adriana Schermaier

Setting up our data

In [35]:
```
#!pip install pandas
#!pip install plotly
#!pip install packaging
#!pip install haversine
#!pip install ipywidgets
```

In [36]:
```
import numpy as np
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())
import datetime as dt
from haversine import haversine, Unit
import ipywidgets as widgets
from ipywidgets import interact, interact_manual
```

In [37]:
```
birds = pd.read_csv('bird_tracking.csv')
```

In [38]:
```
# setting up the date and time columns
birds = birds.assign(date_time = lambda d: pd.to_datetime(d['date_time'], format = '%Y-%m-%d'))

birds = birds.assign(month = lambda d: d['date_time'].dt.month,
                     day = lambda d: d['date_time'].dt.day,
                     hour = lambda d: d['date_time'].dt.hour)

# adding month and day columns as strings for use in the migration and location plots
birds["month_str"] = birds["month"].astype(str)
birds["day_str"] = birds["day"].astype(str)
```

In [39]:
```
#calculating and adding the distance travelled column
birds['distance'] = 0

for row_index in range(birds.shape[0]-1):
    # The if statement is necessary so that when we hit the last value for one bird,
    # it is not used to calculate the first distance for the next bird
    if birds.loc[row_index,'bird_name']==birds.loc[row_index+1,'bird_name']:
        distance = haversine((birds.loc[row_index, 'latitude'],birds.loc[row_index, 'longitude']),
                             (birds.loc[row_index + 1, 'latitude'], birds.loc[row_index + 1, 'longitude']),unit=Unit.MILES)
        birds.loc[row_index + 1, 'distance'] = round(distance,2)
```

In [40]:
```
#fixing the altitude and speed for mapbox display purpose
def fix_alt(alt):
    if alt > 0:
        return np.log2(alt)
    else:
        return 1
def fix_speed(speed):
    if speed > 1:
        return np.log2(speed)
    else:
        return 1

birds['fix_alt'] = birds.apply(lambda x: fix_alt(x.altitude), axis =1)
birds['fix_speed'] = birds.apply(lambda x: fix_speed(x.speed_2d), axis = 1)
```

Bird Activity: Night and Day

Night is defined as any hour >= 18 or <= 5.

We are interested in seeing when and how active each bird is, and the proportions of distance traveled during night vs day.

In [41]:
```
#Seperating datasets into night observations and day observations
nightbird=birds.loc[lambda d: (d['hour'] >= 18) | (d['hour'] <= 5)]
daybird=birds.loc[lambda d: (d['hour'] < 18) & (d['hour'] > 5)]
```

In [42]:
```
daymovement=daybird.groupby('bird_name')['distance'].sum()
daymovement.to_frame()
# Nico travels the most distance during the day
```

Out[42]:

	distance
bird_name	
Eric	6901.81
Nico	9762.27
Sanne	8528.43

In [43]:
```
nightmovement=nightbird.groupby('bird_name')['distance'].sum()
nightmovement.to_frame()
# Sanne travels the most distance during the night
```

Out[43]:

	distance
bird_name	
Eric	3522.38
Nico	7006.93
Sanne	7173.34

In [44]:
```
birddistance=birds.groupby('bird_name')['distance'].sum()
birddistance.to_frame()
# Nico travels the most distance overall
```

Out[44]:

	distance
bird_name	
Eric	10424.19
Nico	16769.20
Sanne	15701.77

In [45]:
```
# Eric:
print(daymovement[0]/birddistance[0])
print(nightmovement[0]/birddistance[0])
```

```
0.662095568097
0.337904431903
```

In [46]:
```
# Nico:
print(daymovement[1]/birddistance[1])
print(nightmovement[1]/birddistance[1])
```

```
0.58215478377
0.41784521623
```

In [47]:
```
# Sanne:
print(daymovement[2]/birddistance[2])
print(nightmovement[2]/birddistance[2])
```

```
0.543150867705
0.456849132295
```

Eric moves 2/3 of his distance during the day, and 1/3rd during the night

Nico moves a little less during the day, a little more during the night. Around 60% daytime

Sanne moves 54% of distance during the day, and 45% during the night

In [48]:
```
def timeTest(hour):
    if ((hour>=18) | (hour<=5)):
        return "night"
    else:
        return "day"
```

In [49]:
```
birds['time'] = birds['hour'].apply(lambda x: timeTest(x))
```

In [50]:
```
bird_list = ['All', 'Eric', 'Nico', 'Sanne']
```

In [51]:
```
@interact(bird=bird_list)
def scatter(bird):
    if bird=='All':
        fig = px.scatter_mapbox(birds,lat="latitude", lon="longitude",
                        color = "time", zoom=2.2)
    else:
        fig = px.scatter_mapbox(birds[birds['bird_name']==bird],lat="latitude", lon="longitude",
                        color = "time", zoom=2.2)
    fig.show()
```

In [52]:
```
# These interactive plots do not show outside of the ipynb file,
# so I have included pictures for these plots in the night_day plot folder
```

The birds all follow a relatively similar pattern: At the start of their migration they travel night and day nonstop and once they reach their destinations, it appears that they start to move more during the day.

In [53]:
```
# Total distance calculated for each month

birds.groupby(['bird_name', 'month'])['distance'].sum().to_frame()
```

Out[53]:

		distance
bird_name	**month**	
Eric	**1**	734.79
	2	338.78
	3	988.73
	4	2858.74
	8	582.32
	9	1104.36
	10	919.07
	11	2521.08
	12	376.32
Nico	**1**	1590.98
	2	1398.48
	3	1437.28
	4	4244.62
	8	801.56
	9	1088.88
	10	962.07
	11	3496.04
	12	1749.29
Sanne	**1**	1615.89
	2	1166.68
	3	1112.48
	4	3942.61
	8	1568.96
	9	3184.70
	10	1052.80
	11	988.41
	12	1069.24

In [54]:
```
# Estimated distance calculated for each month
dist_month = pd.DataFrame([], columns=['bird_name', 'month', 'miles_travelled'])

for bird in birds.bird_name.unique():
    bird_df = birds.loc[lambda d: d['bird_name'] == bird]
    for month in bird_df['month'].unique():
        bird_month = bird_df.loc[lambda d: d['month'] == month].reset_index()
        first = ((bird_month['latitude'][0]),(bird_month['longitude'][0]))
        last = ((bird_month['latitude'][(bird_month.shape[0] - 1)]),
                (bird_month['longitude'][(bird_month.shape[0] - 1)]))
        dist_travelled = round(haversine(first, last, unit=Unit.MILES), 2)
        dist_month = dist_month.append({'bird_name': bird, 'month': month,
                                        'miles_travelled': dist_travelled}, ignore_index = True)

dist_month = dist_month.groupby(['bird_name', 'month'])['miles_travelled'].sum().to_frame()
dist_month
```

Out[54]:

		miles_travelled
bird_name	**month**	
Eric	**1**	0.03
	2	0.02
	3	493.15
	4	1095.08
	8	76.86
	9	17.49
	10	0.53
	11	1508.48
	12	0.09
Nico	**1**	201.76
	2	110.84
	3	130.23
	4	2895.58
	8	36.32
	9	36.30
	10	57.10
	11	2054.38
	12	377.01
Sanne	**1**	23.89
	2	123.79
	3	49.80
	4	2627.02
	8	769.43
	9	1978.44
	10	0.52
	11	0.81
	12	0.24

I have decided to look at estimated distance purposefully here. It better shows whether a bird is staying in the same general area vs traveling long distances.

What we can see is that Sanne and Eric have multiple months where they have 0 miles traveled. This could indicate that they are birds that nest, whereas Nico moves to a new location every month, indicating that he might not nest.

Timeline of Migration and Location

In [55]:
```
@interact(bird = bird_list)
def mapbox_by_bird(bird):

    if bird == 'All':
        bird_df = birds
        label = "bird_name"
    else:
        bird_df = birds.loc[lambda d: d['bird_name'] == bird]
        label = "day_str"

    bird_months = px.scatter_mapbox(bird_df, lat="latitude", lon="longitude", color = "month_str", text = label,
                                    color_discrete_sequence=px.colors.qualitative.Prism, zoom=2.2)
    bird_months.show()
```

In [56]:
```
# These interactive plots do not appear outside of the ipynb file,
# so I have included pictures for these plots in the timeline_location plot folder
```

Eric

Eric leaves from the central area of the Hauts-de-France region of France on November 10th. He stops at the very end of November and stays on the coast of Morocco (around Agadir) from December to March, leaving on the 29th. He finishes his trip back in April, settling in northern Belgium around the coast.

Nico

Nico starts off a little more east than the other birds, by the border of France and Belgium. At the end of October, he moves more toward the central area of the Hauts-de-France region of France, stays there for a bit, and then starts going south on the 16th of November. He keeps travelling pretty much up until March, where he stays on the coast of the Zinguinchor region of Senegal. He then leaves the area around April 8th and goes all the way back north, staying right off the coast of Belgium.

Sanne

Sanne starts off much differently than the other two! He starts like Eric in the central area of the Hauts-de-France region of France, but he starts flying south in August, unlike the other birds who started in November. He continues flying in September, and at the beginning of October, he settles around the coast of the Fatick and Thies region of Senegal and stays there until the beginning of February. He starts heading north on the 6th of February, but most of his distance is covered in April like the other two birds. He then settles just like Eric in northern Belgium by the coast.

Looking at Speed and Altitude

In [57]:
```
#subsets for looking at one direction of migration at a time (generally)
fwbirds = birds[(birds['month'] >= 8) & (birds['month'] <= 12)]
wsbirds = birds[(birds['month'] >= 1) & (birds['month'] <= 5)]
```

In [58]:
```
@interact(mig = ['down', 'up'], var = ['fix_alt', 'fix_speed'])
def mapbox_by_mig_var(mig, var):
    if(mig == 'down'):
        df = fwbirds
    else:
        df = wsbirds
    fig = px.scatter_mapbox(df, lat="latitude", lon="longitude", color = var,
                            text = 'bird_name',
                            color_continuous_scale=px.colors.diverging.Fall, size_max= 5, zoom=2.2)
    fig.show()
```

In [59]:
```
# These interactive plots do not appear outside of the ipynb file,
# so I have included pictures for these plots in the map_speed_altitude plot folder
```

Eric

Eric tends to stay relatively high on the first 3/4 of his journey down (besides parts of Portugal) but flys lower as he approaches his vacation spot. On his way back up, he tends to stick to the lower altitudes over water and higher altitudes over land, again, flying lower as he returns home. As for speed, there are no real distinct differences over his journey other than the speed drops when he approaches his vacation spot and home

Nico

Nico's trip down had him flying high most of the time, except for the times where he was on the coast. On his way back home, Nico follows the same trend as Eric, flying very high over the Spanish peninusla and dipping a little lower over water and the coast. Nico's speed is consistently inconsistent on both his way down and back up.

Sanne

Sanne seems to, on average, follow the trends as before. However overall he seemed to just fly a little lower on the coast and over the water and continued to fly higher over the land. Sanne follows the trend sent before on his way back up, however, he was recorded as flying lower over the water compared to the others. Sanne follows the previous speed trends.

Boxplots

These were just another take on how I separated the migration into two halves and how the speed and altitude differed depending on the months.

In [60]:
```
@interact(var = ['fix_alt', 'speed_2d'])
def box_by_var(var):
    fig = px.box(birds, x = "month" , y = var , color = "bird_name")
    fig.show()
```

In [61]:
```
# These interactive plots do not appear outside of the ipynb file,
# so I have included pictures for these plots in the box_speed_altitude plot folder
```