**#Step1 Biodiversity Project**

import codecademylib

from matplotlib import pyplot as plt

import pandas as pd

species = pd.read\_csv('species\_info.csv')

print species.head()

**#Step2**

species\_count = species.scientific\_name.nunique()

print species\_count

species\_type = species.category.unique()

print species\_type

conservation\_statuses = species.conservation\_status.unique()

print conservation\_statuses

**#Step3Analyse Species Conservation Status**

species.fillna('No Interventio', inplace = True)

conservation\_counts = species.groupby('conservation\_status').scientific\_name.nunique().reset\_index()

print conservation\_counts

#the tables shows us only 180/5500 are concern about their future. However groupby doesn t take in count the value none

**#Step5 Plotting Conservation Status By Species**

protection\_counts = species.groupby('conservation\_status')\

.scientific\_name.nunique().reset\_index()\

.sort\_values(by='scientific\_name')

plt.figure(figsize=(10, 4))

ax = plt.subplot()

plt.bar(range(len(protection\_counts)),protection\_counts.scientific\_name.values)

ax.set\_xticks(range(len(protection\_counts)))

ax.set\_xticklabels(protection\_counts.conservation\_status.values)

plt.ylabel('Number of Species')

plt.title('Conservation Status by Species')

plt.show()

plt.savefig('my\_plotting\_conservation\_Status\_by\_Species.png')

**#Step6 target the endangered species -- aftert identification we set a new framework between category and endanger**

species['is\_protected']=species.conservation\_status.apply(fonction)

print species.head(5)

category\_counts = species.groupby(['category','is\_protected']).scientific\_name.nunique().reset\_index()

category\_pivot = category\_counts.pivot(

columns = 'is\_protected',

index= 'category',

values= 'scientific\_name').reset\_index()

#Bird&fish are the most concerned about this situation in contradiction with the plants

**#Step7: Make the dateframe clearer and more efficient( by renaming columns and addin**g percent)

category\_pivot.columns = ['category','not\_protected','protected']

category\_pivot['percent\_protected'] = (category\_pivot.protected/(category\_pivot.protected+category\_pivot.not\_protected)) \* 100

print category\_pivot.head(7)

**#step8:Chi\_Squared Test**

from scipy.stats import

contingency = [[30, 146],

[75, 413]]

chi2\_contingency

chi2, pval, dof, expected = chi2\_contingency(contingency)

print pval

#the pvalue>0.05, so the difference is not significant

contingency\_reptile\_mammal = [[30, 146],

[5, 73]]

pval\_reptile\_mammal = chi2\_contingency(contingency\_reptile\_mammal)[1]

print(pval\_reptile\_mammal)

**#Step10 Observations DataFrame**

observations = pd.read\_csv('observations.csv')

print observations.head()

scientific\_counts = observations.scientific\_name.nunique()

**#Step11 In Search Of Sheep**

species['is\_sheep'] = species.common\_names.apply(lambda x: 'Sheep' in x)

species\_is\_sheep = species[species.is\_sheep]

print species\_is\_sheep

sheep\_species= species[(species.is\_sheep)&(species.category == 'Mammal')]

print sheep\_species

**#Step12: Merging Sheep and Observation DataFrames**

sheep\_observations = sheep\_species.merge(observations)

print sheep\_observations.head()

obs\_by\_park = sheep\_observations.groupby('park\_name').observations.sum().reset\_index()

print obs\_by\_park.head(150)

**#Step13: Plotting Sheep Sightings**

plt.figure(figsize=(16, 4))

ax = plt.subplot(1,1,1)

plt.bar(range(len(obs\_by\_park)),obs\_by\_park.observations.values)

ax.set\_xticks(range(len(obs\_by\_park)))

ax.set\_xticklabels(obs\_by\_park.park\_name.values)

plt.ylabel('Number of Observations')

plt.title('Observations of Sheep per Week')

plt.show()

**#Step14 Foot and Mouth Reduction Effort**

baseline = 15

minimum\_detectable\_effect = 100\*5./15

sample\_size\_per\_variant = 870

yellowstone\_weeks\_observing = sample\_size\_per\_variant/507.

bryce\_weeks\_observing = sample\_size\_per\_variant/250.