Biodiversity Capstone Project

import codecademylib

#solely for codecademy

import pandas as pd

#pandas is is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

from matplotlib import pyplot as plt

# Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.

# Loading the Data

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

print species.head(5)

#.head() prints just the first few rows of data – you can specify how many rows you want to see.

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

species\_type = species.category.unique()

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

#nunique() versus unique() = nun returns a number where as un returns a list of unique strings.

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

#lists by string “conservation status” with a number of each (nunique())

#reset\_index() makes a new index (or primary key column)

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

#fillna replaces a ‘nan’ with whatever string argument you want.

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

print conservation\_counts\_fixed

import codecademylib

import pandas as pd

from matplotlib import pyplot as plt

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

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

species['is\_protected'] = species.conservation\_status != 'No Intervention'

print(species.head(45))

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

print(category\_counts.head())

category\_pivot = category\_counts.pivot(columns = 'is\_protected', index = 'category', values = 'scientific\_name').reset\_index()

print(category\_pivot)

import codecademylib

import pandas as pd

from matplotlib import pyplot as plt

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

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

species['is\_protected'] = species.conservation\_status != 'No Intervention'

#print(species.head(45))

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

#print(category\_counts.head())

category\_pivot = category\_counts.pivot(columns = 'is\_protected', index = 'category', values = 'scientific\_name').reset\_index()

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)

import codecademylib

import pandas as pd

from matplotlib import pyplot as plt

# Loading the Data

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

# print species.head()

# Inspecting the DataFrame

species\_count = len(species)

species\_type = species.category.unique()

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

# Analyze Species Conservation Status

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

# print conservation\_counts

# Analyze Species Conservation Status II

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

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

# Plotting Conservation Status by Species

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

.scientific\_name.count().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')

labels = [e.get\_text() for e in ax.get\_xticklabels()]

print ax.get\_title()

# plt.show()

species['is\_protected'] = species.conservation\_status != 'No Intervention'

#sets up a Boolean – so it is either ‘True’ or ‘False’.

category\_counts = species.groupby(['category', 'is\_protected'])\

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

# print category\_counts.head()

category\_pivot = category\_counts.pivot(columns='is\_protected', index='category', values='scientific\_name').reset\_index()

#pivots the data

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

#renames the columns

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

#new column with percent protected calculated.

print category\_pivot.head()

category not\_protected protected percent\_protected

0 Amphibian 72 7 8.860759

1 Bird 413 75 15.368852

2 Fish 115 11 8.730159

3 Mammal 146 30 17.045455

4 Non-vascular Plant 328 5 1.501502

5 Reptile 73 5 6.410256

6 Vascular Plant 4216 46 1.079305

Mammals are NOT significantly different from amphibians, birds & fish but are from the rest.

contingency = [[30, 146],[75, 413]]

from scipy.stats import chi2\_contingency

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

print pval

pval\_bird\_mammal= 0.6875

pval\_reptile\_mammal = 0.03835

pval\_vascular\_plant\_mammal = 1.44e-55

#contingency number are for the first pval (mammals versus birds…used chi-squared because

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

import codecademylib

import pandas as pd

from matplotlib import pyplot as plt

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

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

species['is\_protected'] = species.conservation\_status != 'No Intervention'

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

#new csv being loaded

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

# this line searches for the word ‘sheep’ in the column listed (common\_name)

print(species.head(5))

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

#this somehow returns only those things in the column where

print species\_is\_sheep

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

#now we select all rows that have both a ‘true’ for sheep AND a category of mammal.

print sheep\_species

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

#this line merges the two dataframes observations & sheep\_species

print sheep\_observations.head()

scientific\_name park\_name observations category common\_names conservation\_status is\_protected is\_sheep

0 Ovis canadensis Yellowstone National Park 219 Mammal Bighorn Sheep, Bighorn Sheep Species of Concern True True

1 Ovis canadensis Bryce National Park 109 Mammal Bighorn Sheep, Bighorn Sheep Species of Concern True True

2 Ovis canadensis Yosemite National Park 117 Mammal Bighorn Sheep, Bighorn Sheep Species of Concern True True

3 Ovis canadensis Great Smoky Mountains National Park 48 Mammal Bighorn Sheep, Bighorn Sheep Species of Concern True True

4 Ovis canadensis sierrae Yellowstone National Park 67 Mammal Sierra Nevada Bighorn Sheep Endangered True True

5 Ovis canadensis sierrae Yosemite National Park 39 Mammal Sierra Nevada Bighorn Sheep Endangered True True

6 Ovis canadensis sierrae Bryce National Park 22 Mammal Sierra Nevada Bighorn Sheep Endangered True True

7 Ovis canadensis sierrae Great Smoky Mountains National Park 25 Mammal Sierra Nevada Bighorn Sheep Endangered True True

8 Ovis aries Yosemite National Park 126 Mammal Domestic Sheep, Mouflon, Red Sheep, Sheep (Feral) No Intervention False True

9 Ovis aries Great Smoky Mountains National Park 76 Mammal Domestic Sheep, Mouflon, Red Sheep, Sheep (Feral) No Intervention False True

10 Ovis aries Bryce National Park 119 Mammal Domestic Sheep, Mouflon, Red Sheep, Sheep (Feral) No Intervention False True

11 Ovis aries Yellowstone National Park 221 Mammal Domestic Sheep, Mouflon, Red Sheep, Sheep (Feral) No Intervention False True

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

#grouping by park name and summing the number of observations.

print obs\_by\_park

park\_name observations

0 Bryce National Park 250

1 Great Smoky Mountains National Park 149

2 Yellowstone National Park 507

3 Yosemite National Park 282

Plotting graph of Sheep Observations by Park

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

ax = plt.subplot()

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()