Biodiversity for the National Parks

From: Codecademy - Introduction to Data Analysis

By: David Brooks

ORIGINAL SAMPLE DATA FROM SPECIES_INFO.CSV

- DATA FROM THE CSV FILE HAS INFO ON ANIMALS FROM DIFFERENT NATIONAL PARK AROUND THE COUNTRY
- DATA INCLUDED: CATEGORY, SCIENTIFIC NAME, COMMON NAME, CONSERVATION STATUS FOR ALL ANIMALS
- TO ANALYZE THIS DATA WE USED PYTHON PANDAS

	category	scientific_name	common_names	conservation_status
0	Mammal	Clethrionomys gapperi gapperi	Gapper's Red-Backed Vole	nan
1	Mammal	Bos bison	American Bison, Bison	nan
2	Mammal	Bos taurus	Aurochs, Aurochs, Domestic Cattle (Feral), Domesticated Cattle	nan
3	Mammal	Ovis aries	Domestic Sheep, Mouflon, Red Sheep, Sheep (Feral)	nan
4	Mammal	Cervus elaphus	Wapiti Or Elk	nan

```
import codecademylib
from matplotlib import pyplot as plt
import pandas as pd

species = pd.read_csv('species_info.csv')

print(species.head())
```

INSPECTING THE DATA FRAME

- WANTING TO KNOW MORE ABOUT DATA IN THE CSV FILE WE INSPECT THE COLUMNS
  - HOW MANY UNIQUE SPECIES OF ANIMALS IN THE DATA FRAME
  - HOW MANY UNIQUE TYPES OF SPECIES IN CATEGORY
  - WHAT ARE THE DIFFERENT TYPES OF CONSERVATION STATUSES

```
species_count = species.scientific_name.nunique()
species_type = species.category.unique()
conservation_statuses = species.conservation_status.unique()
```

ANALYZING SPECIES

- We thought it would be interesting to see how many species fall into each conversation statuses.
- After doing that we thought the wasn't an accurate representation of all the data since there are a lot of Null spaces in conversation status column. So we did the following to fix that.

```
conservation_counts =
species.groupby('conservation_status').scientific_name.nunique().reset_index()
print(conservation_counts)
```

	conservation_status	scientific_name
0	Endangered	15
1	In Recovery	4
2	Species of Concern	151
3	Threatened	10

```
species.fillna('No Intervention', inplace = True)
conservation_counts_fixed =
species.groupby('conservation_status').scientific_name.nunique().reset_index()
print(conservation_counts_fixed)
```

	conservation_status	scientific_name
0	Endangered	15
1	In Recovery	4
2	No Intervention	5363
3	Species of Concern	151
4	Threatened	10

MAKING A BAR GRAPH WITH THE DATA

- SINCE WE HAD A TABLE OF DATA FROM THE CONSERVATION STATUES WE MADE A GRAPH WITH THE INFO.

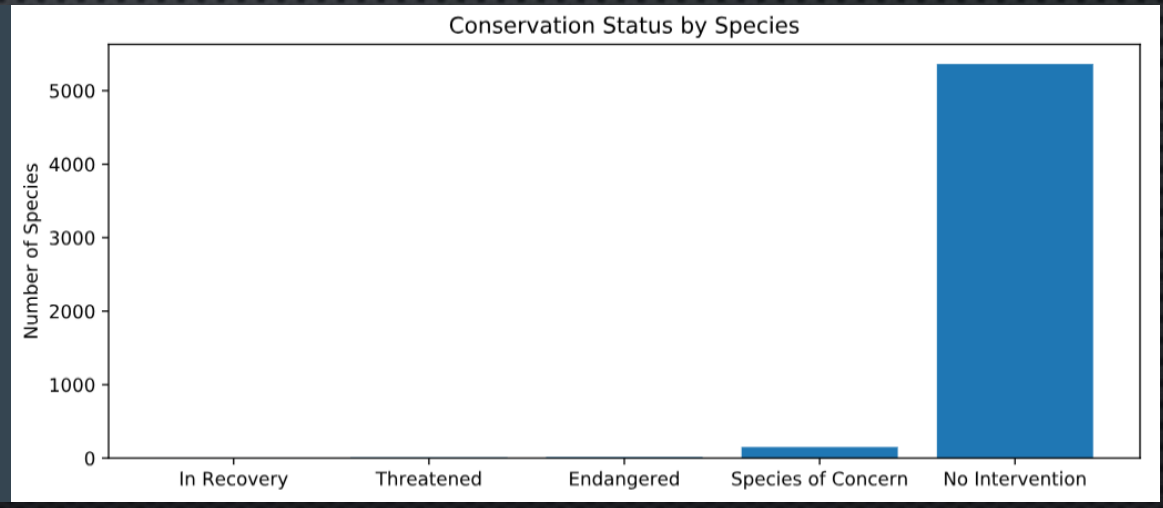
```
import codecademylib
import pandas as pd
from matplotlib import pyplot as plt

species = pd.read_csv('species_info.csv')

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

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.conservation_status)),protection_counts.scientific_name)
ax.set_xticks(range(len(protection_counts.conservation_status)))
ax.set_xticklabels(protection_counts.conservation_status)
plt.ylabel('Number of Species')
plt.title('Conservation Status by Species')
plt.show()
```



INVESTIGATING ENDANGERED SPECIES

- ARE CERTAIN TYPES OF SPECIES MORE LIKELY TO BE ENDANGERED?
- TO ANSWER THIS QUESTION WE WILL NEED TO DO SOME MORE ANALYZATION OF THE DATA AND FORM A PIVOT TABLE.

```
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'

category_counts = species.groupby(['category','is_protected']).scientific_name\
.count().reset_index().sort_values(by='scientific_name')

print(category_counts.head())

category_pivot = category_counts.pivot(\
            columns = 'is_protected',\
            index = 'category',\
            values = 'scientific_name').reset_index()

print(category_pivot)

category_pivot.columns=['category','not_protected','protected']
category_pivot['percent_protected'] = category_pivot.protected /
(category_pivot.protected + category_pivot.not_protected)

print(category_pivot)
```

	category	not_protected	protected	percent_protected
0	Amphibian	73	7	0.087500
1	Bird	442	79	0.151631
2	Fish	116	11	0.086614
3	Mammal	176	38	0.177570
4	Nonvascular Plant	328	5	0.015015
5	Reptile	74	5	0.063291
6	Vascular Plant	4424	46	0.010291

CHI-SQUARED TEST

- ARE CERTAIN TYPES OF SPECIES MORE LIKELY TO BE ENDANGERED?
- WITH THE PIVOT TABLE WE ARE MADE WE ARE NOW MORE EQUIPPED TO ANSWER THIS QUESTION.
- WE RAN A CHI-SQUARED TEST TO SEE IF THERE ARE SIGNIFICANT RELATION BETWEEN CATEGORY OF SPECIES

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

chi,pval,dof,excepted=chi2_contingency(contingency)
print(pval)

contingency2 = [[5,73],
                [30,146]]

chi2,pval_reptile_mammal,dof2,excepted2=chi2_contingency(contingency2)
print(pval_reptile_mammal)
```

```
('Pvalue for Birds and Mammals: ', 0.68759480966613362)
Pvalue is not significant because it is higher 0.05
('Pvalue for Reptiles and Mammals: ', 0.038355590229698977)
Pvalue is significant because it is below 0.05
```

OBSERVATIONS

- We were given another dataframe to work with in a file called observation.csv
- After loading it the data it contains is the scientific name, park name, and number of observations at that park.

```
import codecademylib
import pandas as pd
from matplotlib import pyplot as plt

observations = pd.read_csv('observations.csv')
print(observations.head())
```

	scientific_name	park_name	observations
0	Vicia benghalensis	Great Smoky Mountains National Park	68
1	Neovison vison	Great Smoky Mountains National Park	77
2	Prunus subcordata	Yosemite National Park	138
3	Abutilon theophrasti	Bryce National Park	84
4	Githopsis specularioides	Great Smoky Mountains National Park	85

IN SEARCH OF SHEEP

- We wanted to investigate the movement of sheep with the two dataframes.
- The steps we needed to do that were
  - Make a new column in species called is_sheep
  - Find all the results where mammal is True and is_sheep is True
  - Merge obersation dataframe with the True statements
  - Group that by park names to see the numbers of sheep observed at each park

```
species['is_sheep'] = species.common_names.apply(lambda x: True if 'Sheep' in
x else False)

species_is_sheep = species[species.is_sheep == True]
print(species_is_sheep.head())

sheep_species = species[(species.is_sheep == True) & (species.category
=='Mammal')]
print(sheep_species)

sheep_observations = pd.merge(observations,sheep_species)
print(sheep_observations.head())

obs_by_park =
sheep_observations.groupby('park_name').observations.sum().reset_index()
print(obs_by_park)
```

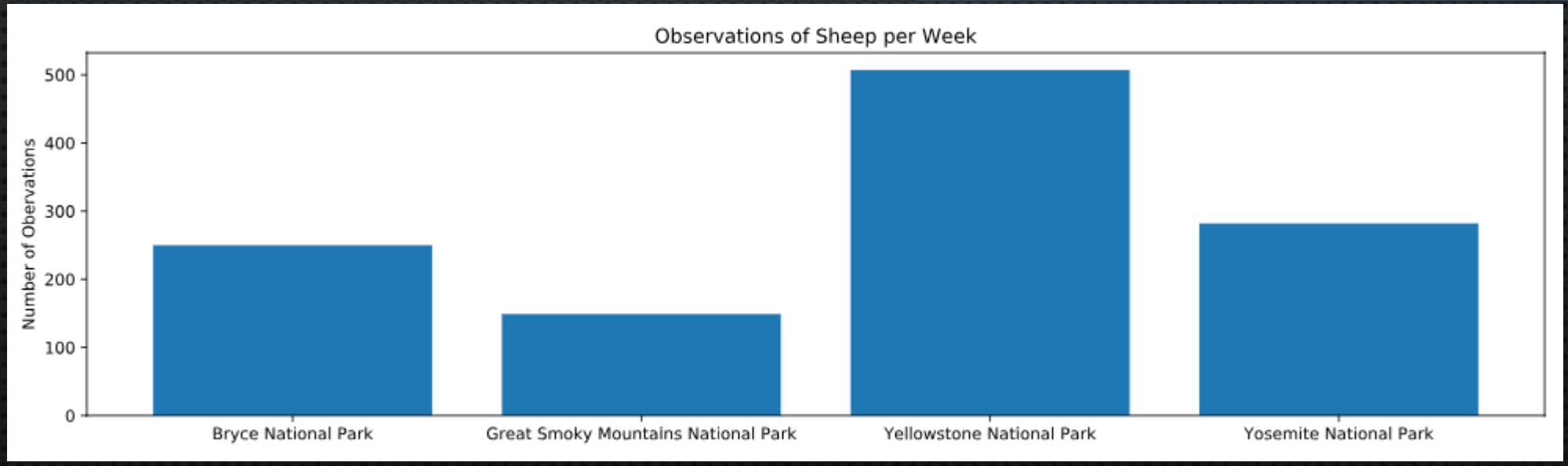
	park_name	observations
0	Bryce National Park	250
1	Great Smoky Mountains National Park	149
2	Yellowstone National Park	507
3	Yosemite National Park	282

PLOTTING SHEEP SIGHTINGS

- WITH THE INFORMATION ON SHEEP OBSERVATIONS IN EACH PARK WE THINK IT WOULD BE BEST TO PLOT OUT A BAR GRAPH WITH AT DATA.

```
from matplotlib import pyplot as plt

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 Obervations')
plt.title('Observations of Sheep per Week')
plt.show()
```



Foot and Mouth Reduction Effort - Sample Size Determination

- Rangers have been running a program to reduce the Foot and Mouth disease in the park, we would like to find out the sample size that the test would need using a sample size calculator.
- To do that we need the Baseline percentage, Min Detectable Effect, and Statistical significance rate.

```
baseline = 15
minimum_detectable_effect = 100*5/15
print 'Min Detectable Effect is: ', minimum_detectable_effect

sample_size_per_variant = 890
print 'Sample Size per Variant is: ', sample_size_per_variant

yellowstone_weeks_observing = sample_size_per_variant/507
print 'Weeks needed at Yellowstone: ', yellowstone_weeks_observing

bryce_weeks_observing = sample_size_per_variant/250
print 'Weeks needed at Bryce: ', bryce_weeks_observing
```

```
Min Detectable Effect is:  33
Sample Size per Variant is:  890
Weeks needed at Yellowstone:  1.75542406312
Weeks needed at Bryce:  3.56
```