

Biodiversity for the National Parks

Capstone Option 2 by Shengcui Cheong
@Toolatetofail (on Slack)

Description of data in species.info.csv

- There are 5,824 record of species across 7 species types in the data
- Of the total record of 5,824 recorded species, 5,541 are unique as retrieved by the following code:

```
species_count = species.scientific_name.nunique()
```

Highlighted code and result for both unique count and count:

```
script.py
1 import codecademylib
2 import pandas as pd
3 from matplotlib import pyplot as plt
4
5 species = pd.read_csv('species_info.csv')
6
7 print species.head()
8 species_count = species.scientific_name.nunique()
9 print species_count
10 species_count2 = species.scientific_name.count()
11 print species_count2
12
```

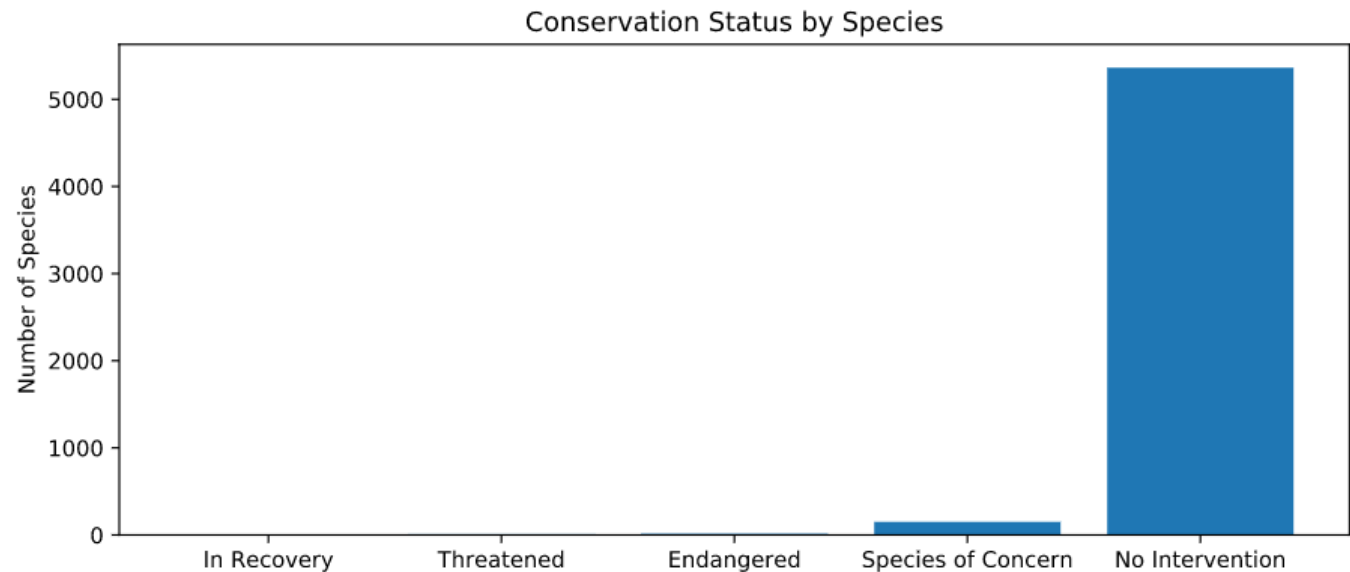
	category	scientific_name	common_names	conservation_status
0	Mammal	Clethrionomys gapperi gapperi	Gapper's Red-Backed Vole	NaN
1	Mammal	Bos bison	American Bison, Bison	NaN
2	Mammal	Bos taurus	Aurochs, Aurochs, Domestic Cattle (Feral), Dom...	NaN
3	Mammal	Ovis aries	Domestic Sheep, Mouflon, Red Sheep,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

5541
5824

Description of data in species.info.csv

- Most of the species are thriving with no need of intervention while some are endangered or threatened. The distribution is as follows:

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



Please note that the species count for these table and charts are 5,543... 2 more than the 5,541 unique count as shown in earlier slide. Perhaps there are unique species categorized in two or more conservation status

Description of data in species.info.csv

- The breakdown of 5,824 species (non-unique) across 7 species types are as follow:

Category	Not Protected	Protected	Total	% protected
Amphibian	73	7	80	8.75%
Bird	442	79	521	15.16%
Fish	116	11	127	8.66%
Mammal	176	38	214	17.76%
Nonvascular Plant	328	5	333	1.50%
Reptile	74	5	79	6.33%
Vascular Plant	4,424	46	4,470	1.03%

```
is_protected    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
```

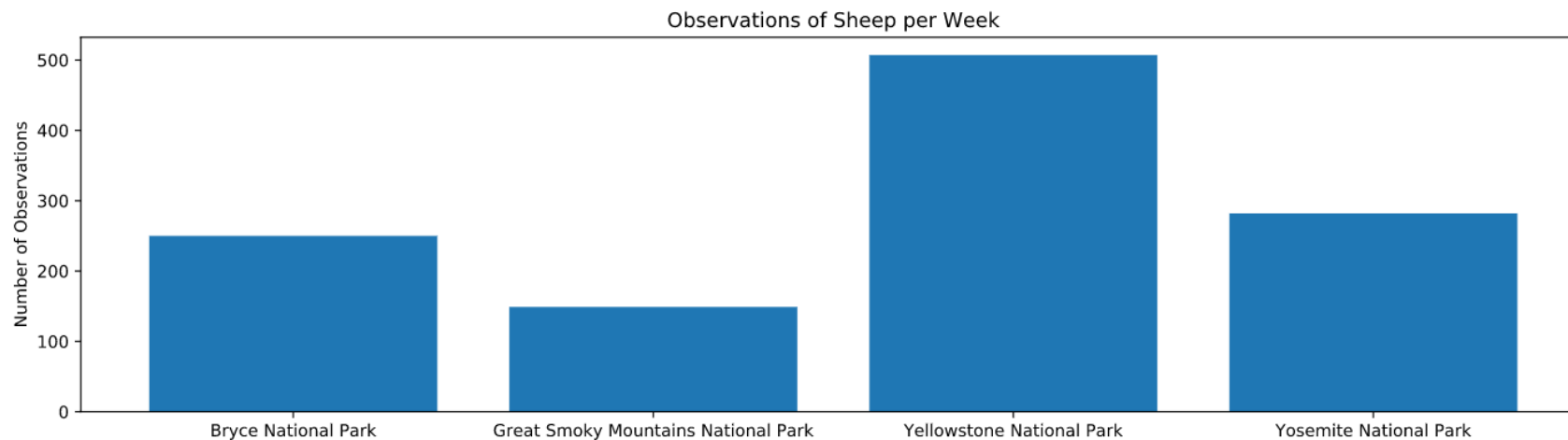
Significance calculation

- From the data in the previous slide, mammal and bird are most threatened.
- Are mammals more likely to be endangered than birds? Or is this due to chance (null hypothesis)? A chi-squared test of significance between them returns a pval of 0.68759 which is more than the 0.05 significance threshold.
- A value higher than 0.05 indicates weak evidence against null hypothesis which means mammal maybe more endangered than bird due to chance.
- A comparative study chi-squared test on mammal vs. reptile yield a pval of 0.0384 which is lower than 0.05 significance threshold. This means it is not due to chance that mammals are more endangered than reptile.
- It is recommended that more conservation and rehabilitation efforts should be channelled to protect birds and mammals as the significance study showed that there are types of species that are more likely to be endangered.

Foot and mouth disease in sheep

- The following is the observation of sheep in 4 national parks in a week

	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



Foot and mouth disease in sheep

- From the study in Bryce National Park, 15% of sheep are known to have foot and mouth disease.
- Therefore this sample sets the baseline at 15%
- As the park rangers want to reduce the % of foot and mouth infection to 10%, data scientist need to be able to measure a drop of 33% (5% out of 15%).
- Hence data scientist should set minimum detectable effect at 33%
- To achieve 90% of confidence level that the park ranger program is indeed effective, the sample size should be at 520.
- If data scientist is to observe at Yellowstone National Park, this will take slightly over a week of observation.
- If data scientist is to observe at Bryce National Park, this will take slightly over two weeks of observation.

Code (Appendix)

Exercise 1 to 4

```
script.py

1 import codecademylib
2 import pandas as pd
3 from matplotlib import pyplot as plt
4
5 species = pd.read_csv('species_info.csv')
6
7 print species.head()
8 species_count = species.scientific_name.nunique()
9 print species_count
10 species_type = species.category.unique()
11 print species_type
12 conservation_statuses = species.conservation_status.unique()
13 print conservation_statuses
14 conservation_counts =
15 species.groupby('conservation_status').scientific_name.nunique().reset_index()
16 print conservation_counts
17 species.conservation_status.fillna('No Intervention', inplace = True)
18 conservation_counts_fixed =
19 species.groupby('conservation_status').scientific_name.nunique().reset_index()
20 print conservation_counts_fixed
```

	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), Dom...	NaN
3	Mammal	Ovis aries	Domestic Sheep, Mouflon, Red Sheep, Sheep (Feral)	NaN
4	Mammal	Cervus elaphus	Wapiti Or Elk	NaN

```
5541
['Mammal' 'Bird' 'Reptile' 'Amphibian' 'Fish' 'Vascular Plant'
 'Nonvascular Plant']
[nan 'Species of Concern' 'Endangered' 'Threatened' 'In Recovery']
```

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

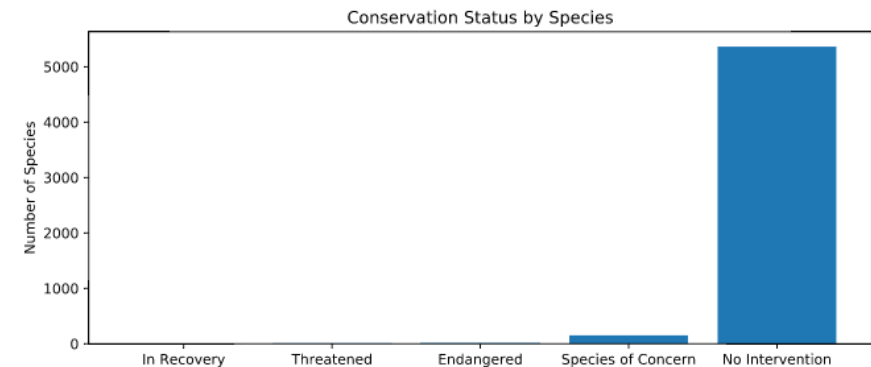
	conservation_status	scientific_name
0	Endangered	15
1	In Recovery	4
2	No Intervention	5363
3	Species of Concern	151
4	Threatened	10

<https://gist.github.com/462869499623775c203031790738a386>

Exercise 5

```
1 import codecademylib
2 import pandas as pd
3 from matplotlib import pyplot as plt
4
5 species = pd.read_csv('species_info.csv')
6
7 species.fillna('No Intervention', inplace = True)
8 protection_counts = species.groupby('conservation_status')\
9     .scientific_name.nunique().reset_index()\
10     .sort_values(by='scientific_name')
11 print protection_counts
12
13 plt.figure(figsize = (10,4))
14 ax = plt.subplot()
15 plt.bar(range(len(protection_counts.conservation_status)),protection_counts.scientific_name)
16 ax.set_xticks([0,1,2,3,4])
17 ax.set_xticklabels(protection_counts.conservation_status)
18 plt.ylabel('Number of Species')
19 plt.title('Conservation Status by Species')
20 plt.show()
```

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



<https://gist.github.com/dbb11bad24e9b20492af25d87056452b>

Exercise 6 to 7

```
1 import codecademylib
2 import pandas as pd
3 from matplotlib import pyplot as plt
4
5 species = pd.read_csv('species_info.csv')
6
7 species.fillna('No Intervention', inplace = True)
8 print species
9
10 species['is_protected'] = species.apply(lambda row:
11     True
12     if row['conservation_status'] != 'No Intervention'
13     else
14     False,
15     axis = 1
16 )
17 print species
18 category_counts = species.groupby(['is_protected', 'category'])
19     ['scientific_name'].count().reset_index()
20 print category_counts.head()
21
22 category_pivot = category_counts.pivot(columns = 'is_protected',
23     index = 'category',
24     values = 'scientific_name'
25     ).reset_index()
26 print category_pivot
27 category_pivot.rename(columns={
28     False: 'not_protected',
29     True: 'protected'},
30     inplace = True
31 )
32 print category_pivot
33 category_pivot['percent_protected'] = category_pivot.protected / (category_pivot.protected
34 + category_pivot.not_protected)
35 print category_pivot
```

Run



```
5812 False
5813 False
5814 False
5815 False
5816 False
5817 False
5818 False
5819 False
5820 False
5821 False
5822 False
5823 False
```

[5824 rows x 5 columns]

is_protected	category	scientific_name
0	False	Amphibian 73
1	False	Bird 442
2	False	Fish 116
3	False	Mammal 176
4	False	Nonvascular Plant 328

is_protected	category	False	True
0	Amphibian	73	7
1	Bird	442	79
2	Fish	116	11
3	Mammal	176	38
4	Nonvascular Plant	328	5
5	Reptile	74	5
6	Vascular Plant	4424	46

is_protected	category	not_protected	protected
0	Amphibian	73	7
1	Bird	442	79
2	Fish	116	11
3	Mammal	176	38
4	Nonvascular Plant	328	5
5	Reptile	74	5
6	Vascular Plant	4424	46

is_protected	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



<https://gist.github.com/7d7698be6a2e35672a3bf3c7f60391b4>

Exercise 8 to 9

```
script.py
30 .sort_values(by='scientific_name')
31
32 # plt.figure(figsize=(10, 4))
33 # ax = plt.subplot()
34 # plt.bar(range(len(protection_counts)),
35 #         protection_counts.scientific_name.values)
36 # ax.set_xticks(range(len(protection_counts)))
37 # ax.set_xticklabels(protection_counts.conservation_status.values)
38 # plt.ylabel('Number of Species')
39 # plt.title('Conservation Status by Species')
40 # labels = [e.get_text() for e in ax.get_xticklabels()]
41 # print ax.get_title()
42 # plt.show()
43
44 species['is_protected'] = species.conservation_status != 'No Intervention'
45
46 category_counts = species.groupby(['category', 'is_protected'])\
47     .scientific_name.count().reset_index()
48
49 # print category_counts.head()
50
51 category_pivot = category_counts.pivot(columns='is_protected', index='category',
52     values='scientific_name').reset_index()
53
54 category_pivot.columns = ['category', 'not_protected', 'protected']
55
56 category_pivot['percent_protected'] = category_pivot.protected / (category_pivot.protected +
57     category_pivot.not_protected)
58
59 print category_pivot
60 contingency = [[30, 146],
61               [75, 413]]
62 from scipy.stats import chi2_contingency
63 chi2, pval, dof, expected = chi2_contingency(contingency)
64 print pval
65 contingency_reptile_mammal = [[5, 73],
66                               [30, 146]]
67 chi2, pval_reptile_mammal, dof, expected = chi2_contingency(contingency_reptile_mammal)
68 print pval_reptile_mammal
```

	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

<https://gist.github.com/4bec4851ba30cecb25b12fe9fba3629e>

Exercise 10 to 12

```
1 import codecademylib
2 import pandas as pd
3 from matplotlib import pyplot as plt
4
5 species = pd.read_csv('species_info.csv')
6 species.fillna('No Intervention', inplace = True)
7 species['is_protected'] = species.conservation_status != 'No Intervention'
8 observations = pd.read_csv('observations.csv')
9 print species.head()
10 print observations.head()
11 species['is_sheep'] = species.apply(lambda row:
12     True
13     if 'Sheep' in row['common_names']
14     else
15     False,
16     axis=1
17 )
18 species_is_sheep = species[species.is_sheep == True]
19 print species_is_sheep
20 sheep_species = species[(species.is_sheep == True) & (species.category == 'Mammal')]
21 print sheep_species
22 sheep_observations = pd.merge(sheep_species, observations)
23 print sheep_observations
24 obs_by_park = sheep_observations.groupby('park_name').observations.sum().reset_index()
25 print obs_by_park
```

Run



	category	scientific_name	common_names	conservation_status
3	Mammal	Ovis aries	Domestic Sheep, Mouflon, Red Sheep, Sheep (Feral)	No Intervention
3014	Mammal	Ovis canadensis	Bighorn Sheep, Bighorn Sheep	Species of Concern
4446	Mammal	Ovis canadensis sierrae	Sierra Nevada Bighorn Sheep	Endangered

5	Mammal	Ovis canadensis	Bighorn Sheep, Bighorn Sheep	Species of Concern
6	Mammal	Ovis canadensis	Bighorn Sheep, Bighorn Sheep	Species of Concern
7	Mammal	Ovis canadensis	Bighorn Sheep, Bighorn Sheep	Species of Concern
8	Mammal	Ovis canadensis sierrae	Sierra Nevada Bighorn Sheep	Endangered
9	Mammal	Ovis canadensis sierrae	Sierra Nevada Bighorn Sheep	Endangered
10	Mammal	Ovis canadensis sierrae	Sierra Nevada Bighorn Sheep	Endangered
11	Mammal	Ovis canadensis sierrae	Sierra Nevada Bighorn Sheep	Endangered

	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

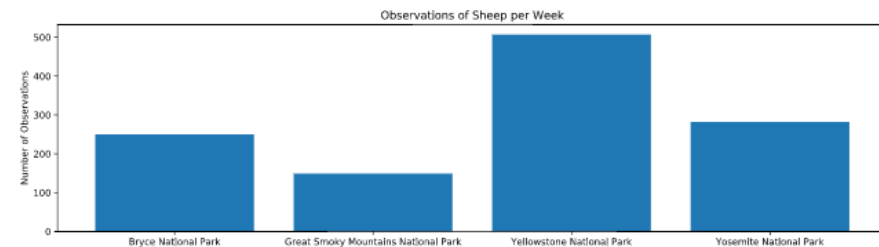


<https://gist.github.com/73a8e5bb2b52147a0ad03b92c3886ba4>

Exercise 13

```
1 import codecademylib
2 import pandas as pd
3 from matplotlib import pyplot as plt
4
5 species = pd.read_csv('species_info.csv')
6 species['is_sheep'] = species.common_names.apply(lambda x: 'Sheep' in x)
7 sheep_species = species[(species.is_sheep) & (species.category == 'Mammal')]
8
9 observations = pd.read_csv('observations.csv')
10
11 sheep_observations = observations.merge(sheep_species)
12
13 obs_by_park = sheep_observations.groupby('park_name').observations.sum().reset_index()
14 print obs_by_park
15
16 plt.figure(figsize=(16,4))
17 ax = plt.subplot()
18 plt.bar(range(len(obs_by_park.park_name)),obs_by_park.observations)
19 ax.set_xticks([0,1,2,3])
20 ax.set_xticklabels(obs_by_park.park_name)
21 plt.ylabel('Number of Observations')
22 plt.title('Observations of Sheep per Week')
23 plt.show()
```

	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



<https://gist.github.com/ad78fa33403cf7f1e82972b68399ffd9>

Exercise 14 to 15

```
1 baseline = 15
2 minimum_detectable_effect = 33
3 sample_size_per_variant = 520
4 yellowstone_weeks_observing = 2
5 bryce_weeks_observing = 3
```

<https://gist.github.com/2e2d861e5adb588e813c74cf650ec9e7>