# Biodiversity for the National Parks

Capstone Option 2 by Shengcui Cheong

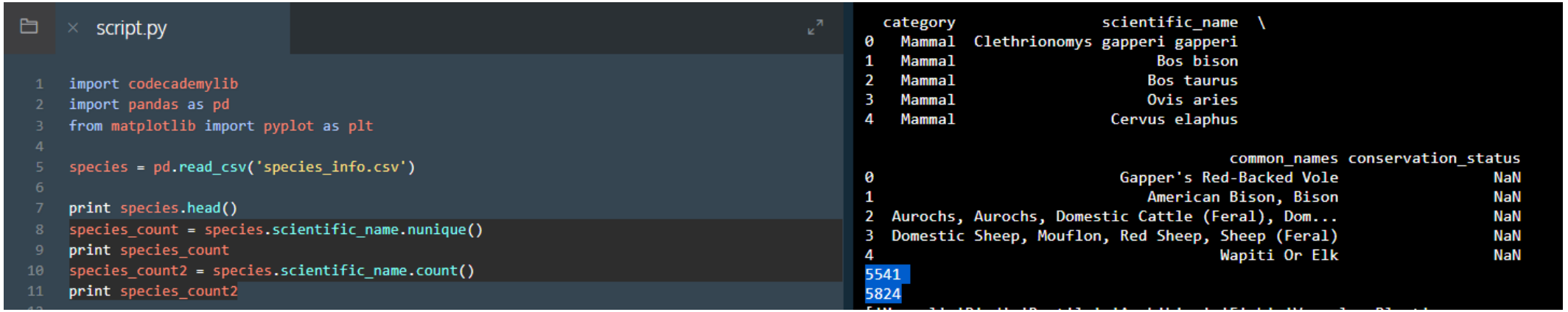
@Toolatetofail (on Slack)

# 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()

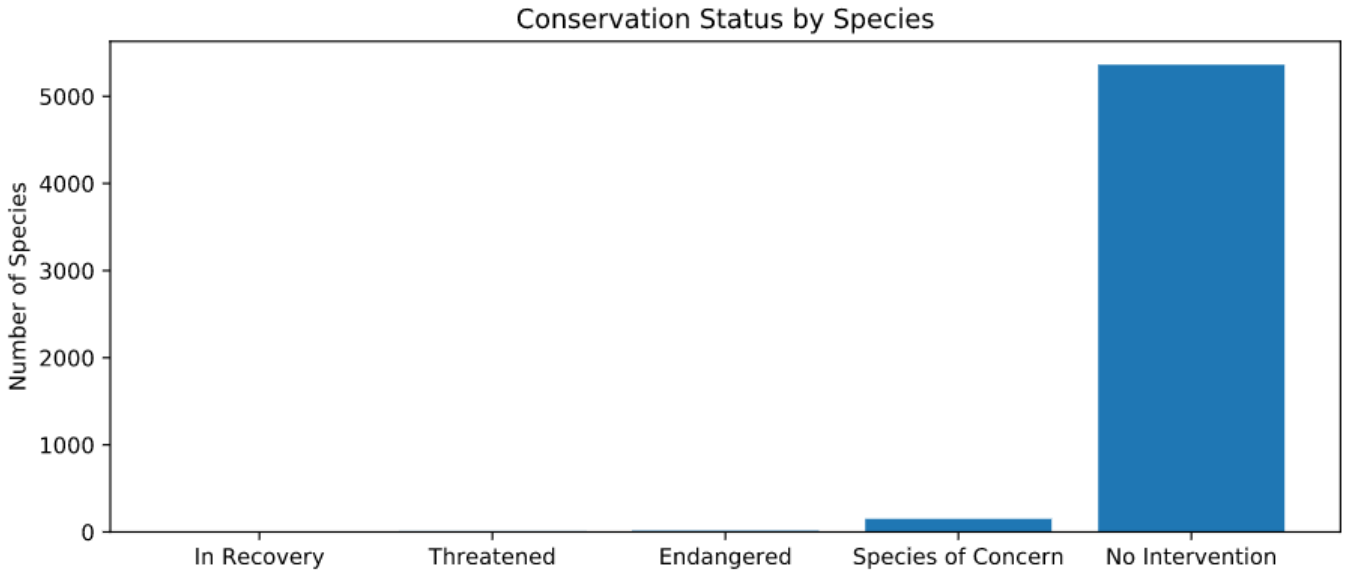
Highlighted code and result for both unique count and count:



# 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

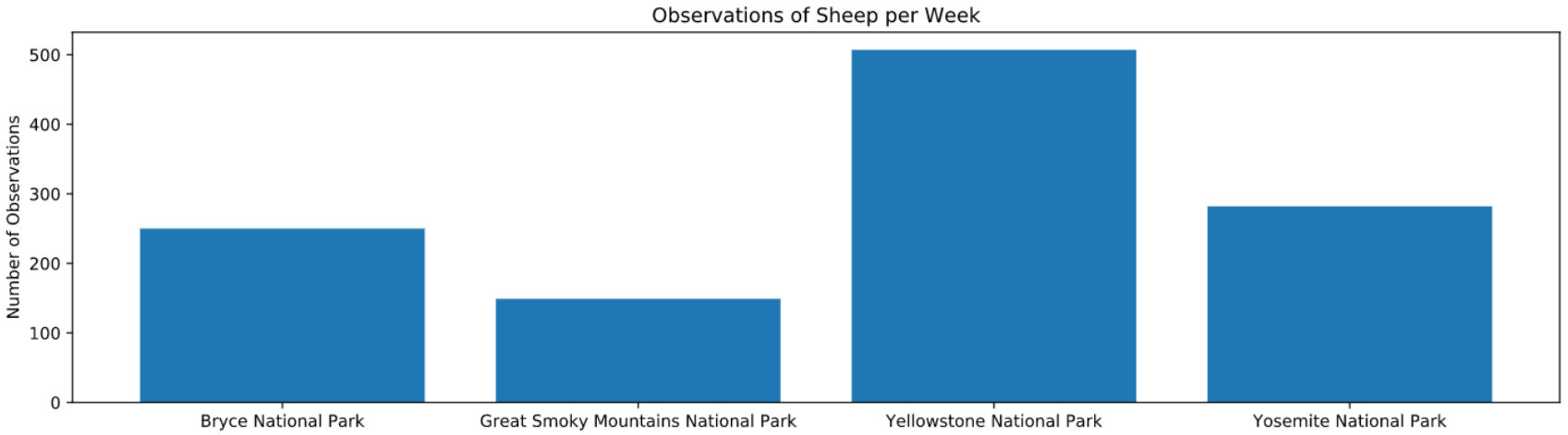
- From the data in the previous slide, mammal and bird are most threatened.
- Are mammals more likely to be endangered than birds? Or is this due to chance (null hypothesis)? A chi-squared test of significance between them returns a pval of 0.68759 which is more than the 0.05 significance threshold.
- 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.

# 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 =
   species.groupby('conservation_status').scientific_name.nunique().reset_index()
15 print conservation_counts
16 species.conservation_status.fillna('No Intervention', inplace = True)
17 conservation_counts_fixed =
   species.groupby('conservation_status').scientific_name.nunique().reset_index()
18 print conservation_counts_fixed
19
```

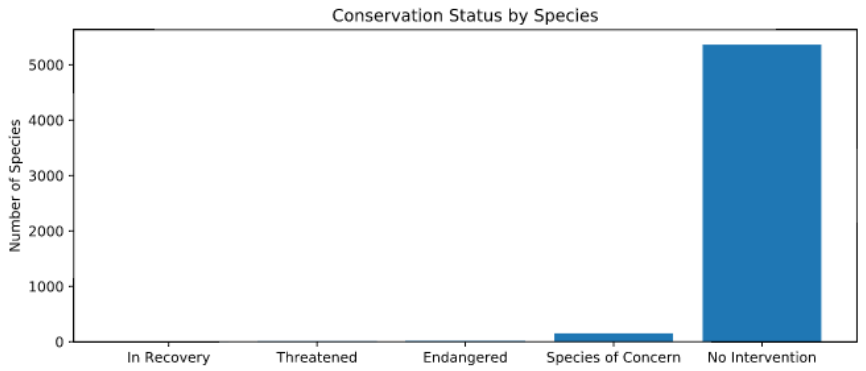
```
  category                scientific_name  \
0   Mammal  Clethrionomys gapperi gapperi
1   Mammal                      Bos bison
2   Mammal                     Bos taurus
3   Mammal                     Ovis aries
4   Mammal                   Cervus elaphus

                                       common_names conservation_status
0                          Gapper's Red-Backed Vole                 NaN
1                             American Bison, Bison                 NaN
2  Aurochs, Aurochs, Domestic Cattle (Feral), Dom...                 NaN
3  Domestic Sheep, Mouflon, Red Sheep, Sheep (Feral)                 NaN
4                                     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/46286949962377 5c203031790738a386

# Exercise 5

```
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')
print protection_counts

plt.figure(figsize = (10,4))
ax = plt.subplot()
plt.bar(range(len(protection_counts.conservation_status)),protection_counts.scientific_name)
ax.set_xticks([0,1,2,3,4])
ax.set_xticklabels(protection_counts.conservation_status)
plt.ylabel('Number of Species')
plt.title('Conservation Status by Species')
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

```python
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)
print species

species['is_protected'] = species.apply(lambda row:
                                        True
                                         if row['conservation_status'] != 'No Intervention'
                                         else
                                         False,
                                        axis = 1
                                        )
print species
category_counts = species.groupby(['is_protected','category'])
['scientific_name'].count().reset_index()
print category_counts.head()

category_pivot = category_counts.pivot(columns = 'is_protected',
                                       index = 'category',
                                       values = 'scientific_name'
                                      ).reset_index()
print category_pivot
category_pivot.rename(columns={
  False:'not_protected',
  True:'protected'},
                      inplace = True
                     )
print category_pivot
category_pivot['percent_protected'] = category_pivot.protected / (category_pivot.protected
+ category_pivot.not_protected)
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

```
    .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'

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()

category_pivot.columns = ['category', 'not_protected', 'protected']

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

print category_pivot
contingency = [[30, 146],
               [75, 413]]
from scipy.stats import chi2_contingency
chi2, pval, dof, expected = chi2_contingency(contingency)
print pval
contingency_reptile_mammal = [[5, 73],
                              [30, 146]]
chi2, pval_reptile_mammal, dof, expected = chi2_contingency(contingency_reptile_mammal)
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
0.687594809666
0.0383555902297
```

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

# Exercise 10 to 12

```
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')
print species.head()
print observations.head()
species['is_sheep'] = species.apply(lambda row:
                                    True
                                    if 'Sheep' in row['common_names']
                                    else
                                     False,
                                     axis=1
                                    )
species_is_sheep = species[species.is_sheep == True]
print species_is_sheep
sheep_species = species[(species.is_sheep == True) & (species.category == 'Mammal')]
print sheep_species
sheep_observations = pd.merge(sheep_species, observations)
print sheep_observations
obs_by_park = sheep_observations.groupby('park_name').observations.sum().reset_index()
print obs_by_park
```

Run

| | category | scientific_name | common_names | conse |
|---|---|---|---|---|
| 3 | Mammal | Ovis aries | Domestic Sheep, Mouflon, Red Sheep, Sheep (Feral) | No Int |
| 3014 | Mammal | Ovis canadensis | Bighorn Sheep, Bighorn Sheep | Speci |
| 4446 | Mammal | Ovis canadensis sierrae | Sierra Nevada Bighorn Sheep | Endan |

| | | | | |
|---|---|---|---|---|
| 5 | Mammal | Ovis canadensis | Bighorn Sheep, Bighorn Sheep | Species |
| 6 | Mammal | Ovis canadensis | Bighorn Sheep, Bighorn Sheep | Species |
| 7 | Mammal | Ovis canadensis | Bighorn Sheep, Bighorn Sheep | Species |
| 8 | Mammal | Ovis canadensis sierrae | Sierra Nevada Bighorn Sheep | Endanger |
| 9 | Mammal | Ovis canadensis sierrae | Sierra Nevada Bighorn Sheep | Endanger |
| 10 | Mammal | Ovis canadensis sierrae | Sierra Nevada Bighorn Sheep | Endanger |
| 11 | Mammal | Ovis canadensis sierrae | Sierra Nevada Bighorn Sheep | Endanger |

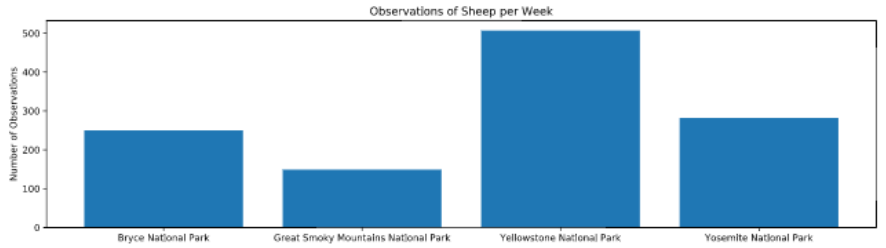
| | park_name | observations |
|---|---|---|
| 0 | Bryce National Park | 250 |
| 1 | Great Smoky Mountains National Park | 149 |
| 2 | Yellowstone National Park | 507 |
| 3 | Yosemite National Park | 282 |

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

# Exercise 13

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

species = pd.read_csv('species_info.csv')
species['is_sheep'] = species.common_names.apply(lambda x: 'Sheep' in x)
sheep_species = species[(species.is_sheep) & (species.category == 'Mammal')]

observations = pd.read_csv('observations.csv')

sheep_observations = observations.merge(sheep_species)

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

plt.figure(figsize=(16,4))
ax = plt.subplot()
plt.bar(range(len(obs_by_park.park_name)),obs_by_park.observations)
ax.set_xticks([0,1,2,3])
ax.set_xticklabels(obs_by_park.park_name)
plt.ylabel('Number of Observations')
plt.title('Observations of Sheep per Week')
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

script.py

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

https://gist.github.com/f7c84ccbd1a9d32465d657f2a83c58bb