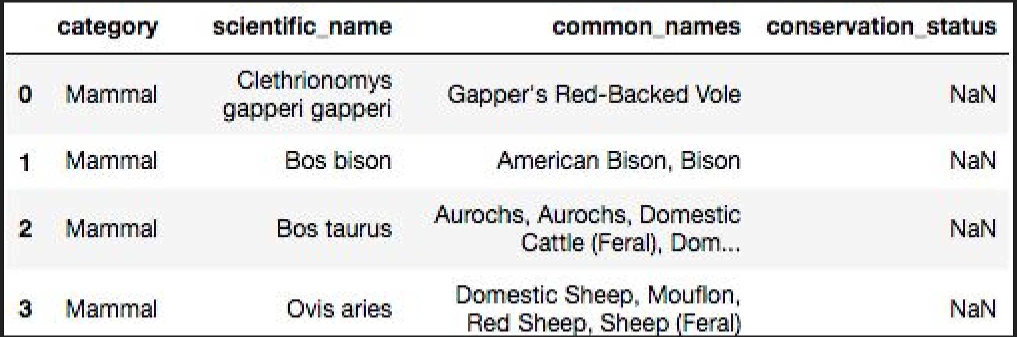
Biodiversity for National Parks

Taylor Marucci
1/21/18

What is it?

- Python 2.7
- Libraries: matplotlib, pandas and scipy.stats
- Access, organize, manipulate, visualize and analyze inspired-by-real data concerning biodiversity in National Parks
- Chi Squared Test Example
- Identify medical trial length at different parks

Original DataFrame: species1 = pd.read_csv(‘species-info.csv’)



	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

category options:

Mammal
Bird
Reptile
Amphibian
Fish
Vascular Plant
Nonvascular Plant

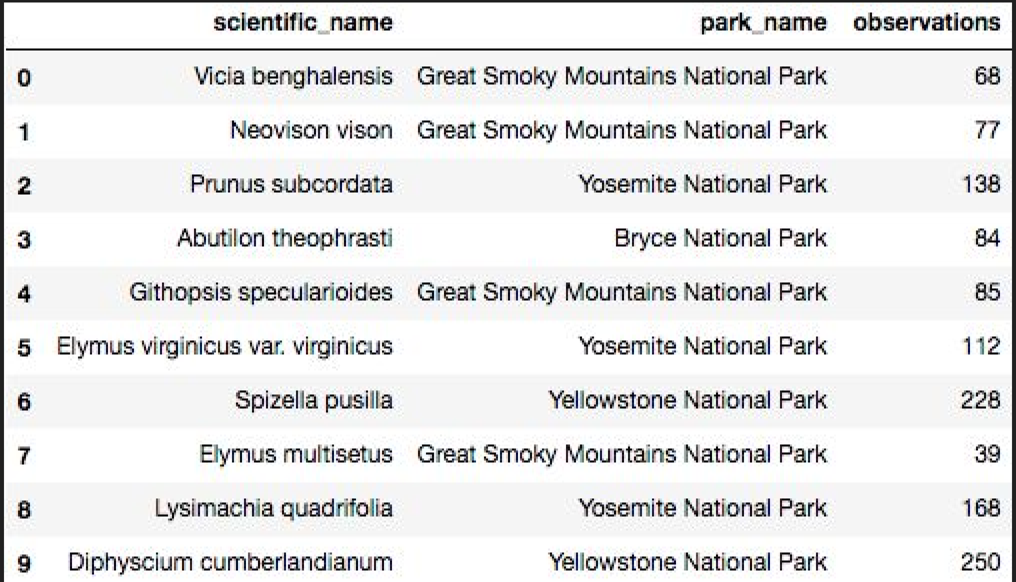
scientific_name:

species1.scientific_name.nunique()
= 5,541

conservation_status options:

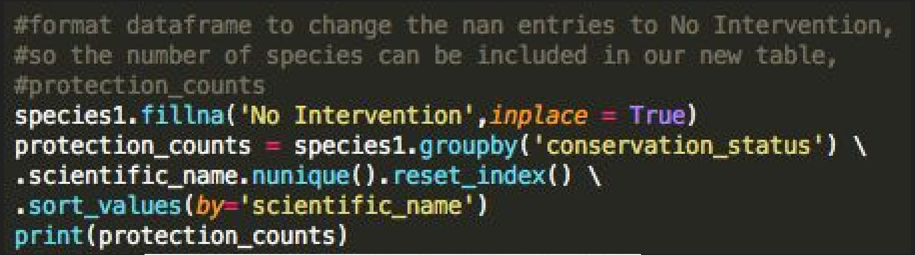
NaN
Species of Concern
Endangered
Threatened
In Recovery

Original DataFrame: observations = pd.read_csv(‘observations.csv’)

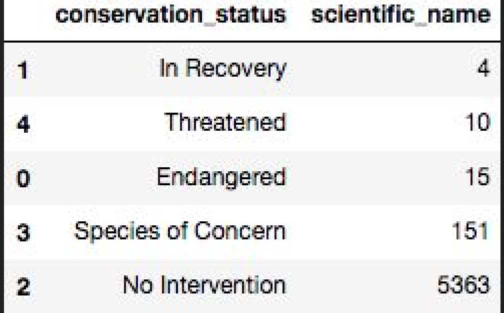


	scientific_name	park_name	observations
0	Vicia benghalensis	Great Smoky Mountains National Park	68
1	Neovison vison	Great Smoky Mountains National Park	77
2	Prunus subcordata	Yosemite National Park	138
3	Abutilon theophrasti	Bryce National Park	84
4	Githopsis specularioides	Great Smoky Mountains National Park	85
5	Elymus virginicus var. virginicus	Yosemite National Park	112
6	Spizella pusilla	Yellowstone National Park	228
7	Elymus multisetus	Great Smoky Mountains National Park	39
8	Lysimachia quadrifolia	Yosemite National Park	168
9	Diphyscium cumberlandianum	Yellowstone National Park	250

Info Overview

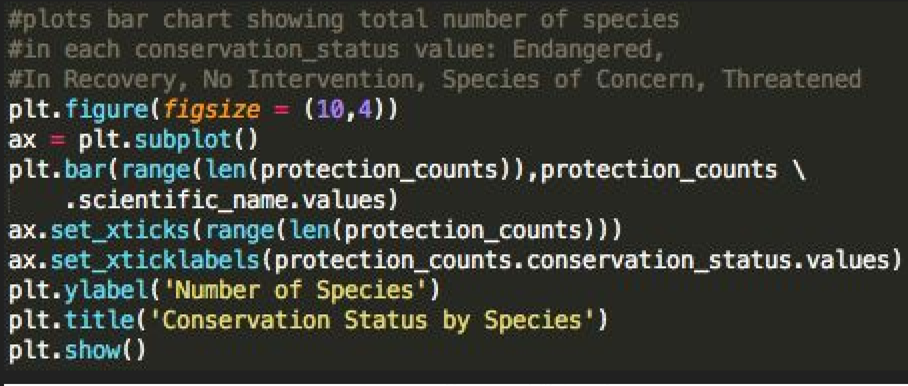


```
#format dataframe to change the nan entries to No Intervention,
#so the number of species can be included in our new table,
#protection_counts
species1.fillna('No Intervention',inplace = True)
protection_counts = species1.groupby('conservation_status') \
.scientific_name.nunique().reset_index() \
.sort_values(by='scientific_name')
print(protection_counts)
```

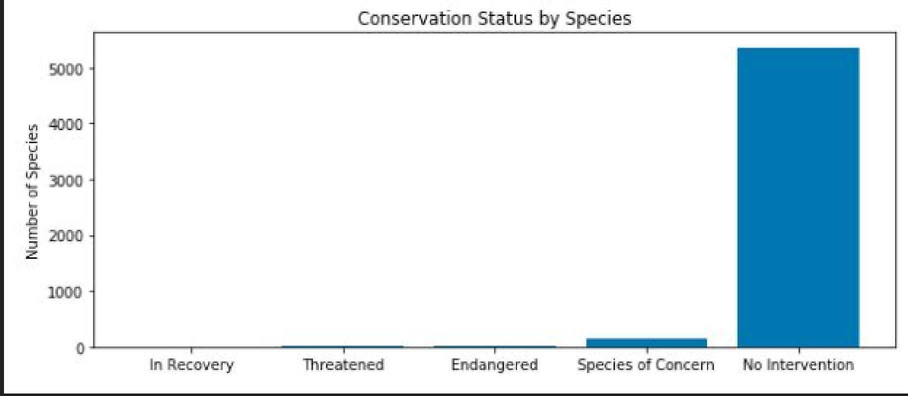


	conservation_status	scientific_name
1	In Recovery	4
4	Threatened	10
0	Endangered	15
3	Species of Concern	151
2	No Intervention	5363

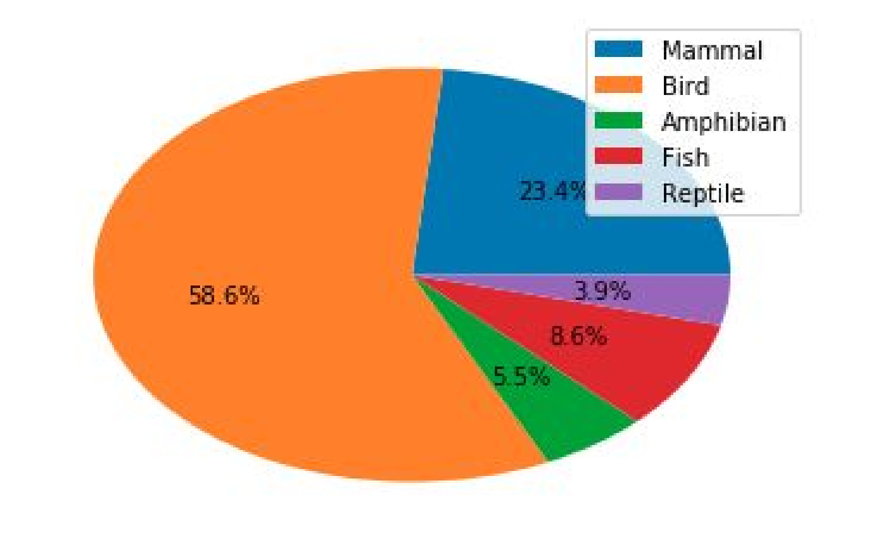
- 3.2% of 5,541 all species are protected
- 13.8% of 946 species of animals are protected



```
#plots bar chart showing total number of species
#in each conservation_status value: Endangered,
#In Recovery, No Intervention, Species of Concern, Threatened
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()
```



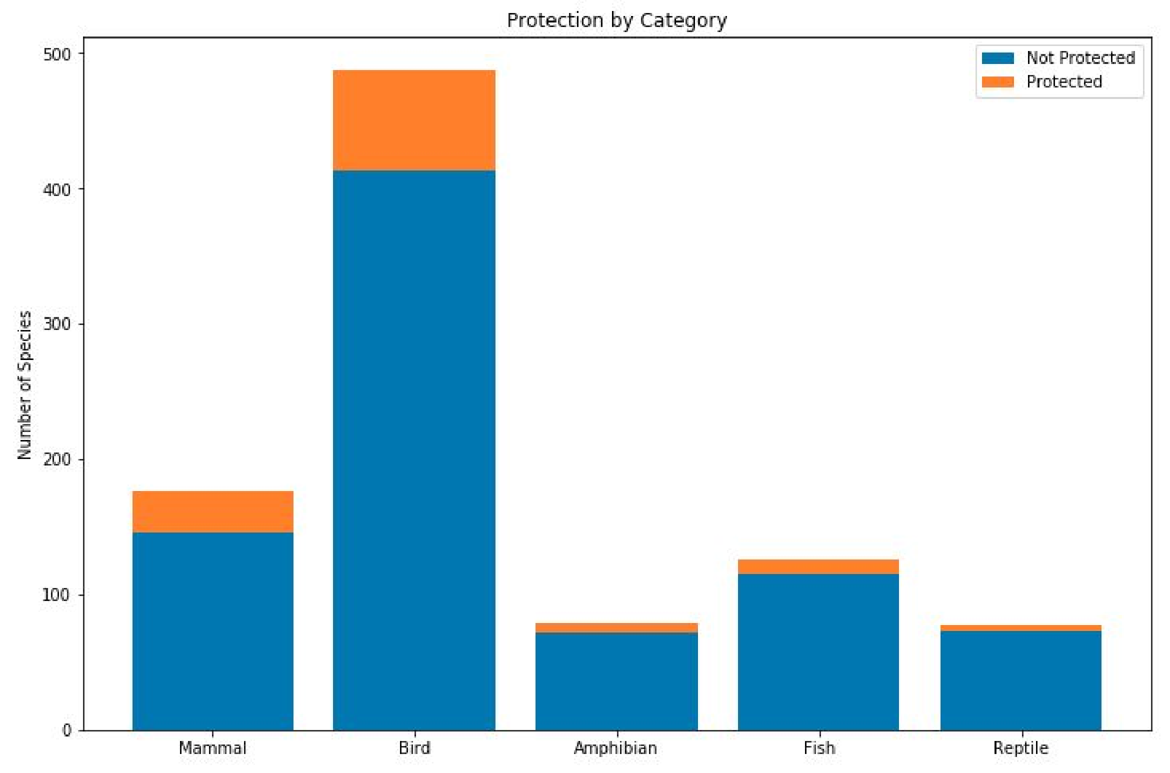
Percent of protected by category:

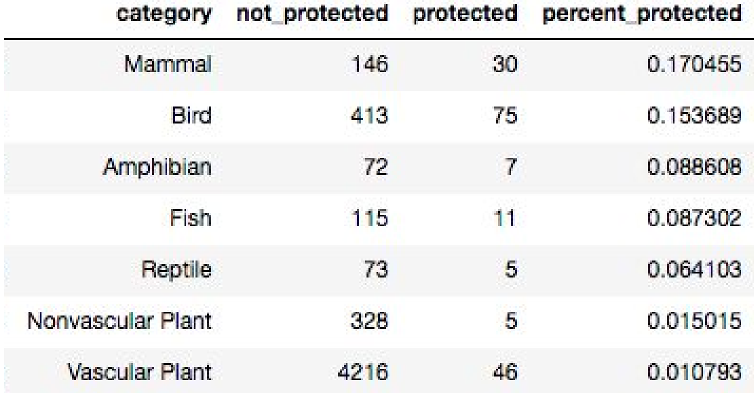


Are species of mammals more likely to be protected than birds?

Are species of reptiles more likely to be protected than mammals?

Protection by Category





category	not_protected	protected	percent_protected
Mammal	146	30	0.170455
Bird	413	75	0.153689
Amphibian	72	7	0.088608
Fish	115	11	0.087302
Reptile	73	5	0.064103
Nonvascular Plant	328	5	0.015015
Vascular Plant	4216	46	0.010793

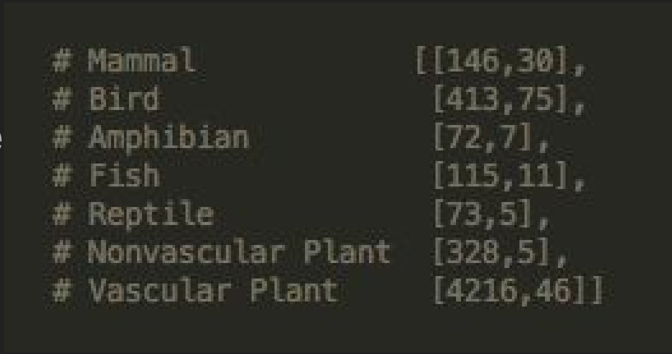
Percent_protected mammal:

= 30 / (146+30)

= .17 = 17%

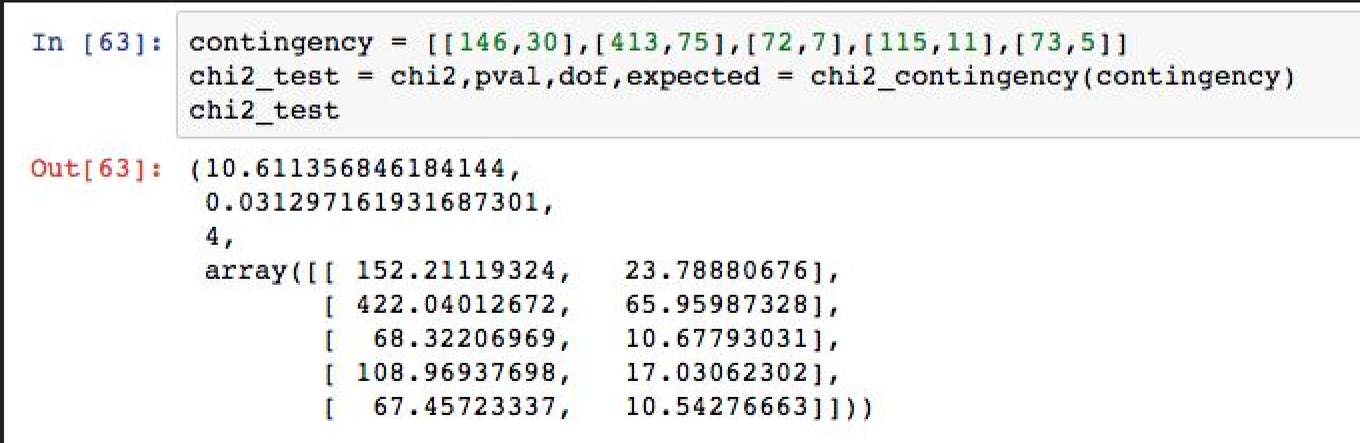
Chi Squared Test

- Scipy.stats.chi2_contingency to test independancies of variables in a contingency table
- Pval > 0.05 - cannot reject Null Hypothesis
- Pval < 0.05 - can reject Null Hypothesis

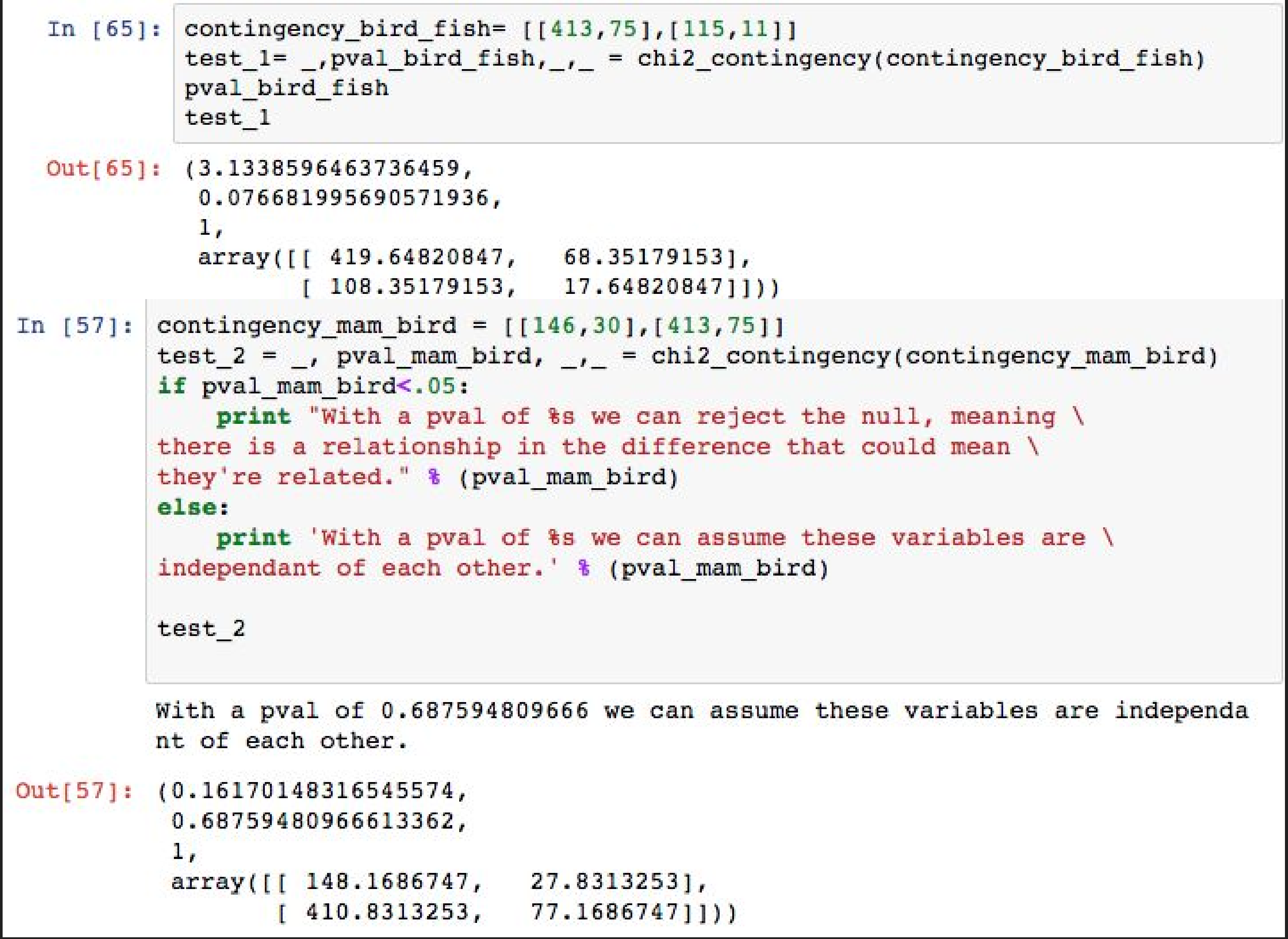


```
# Mammal              [[146,30],
# Bird                 [413,75],
# Amphibian            [72,7],
# Fish                 [115,11],
# Reptile              [73,5],
# Nonvascular Plant    [328,5],
# Vascular Plant       [4216,46]]
```

- From Chi Distribution Table our critical value = 7.78 with $\alpha = 0.10$,
- and dof = 4
- 10.6 > 7.78
- Reject Null



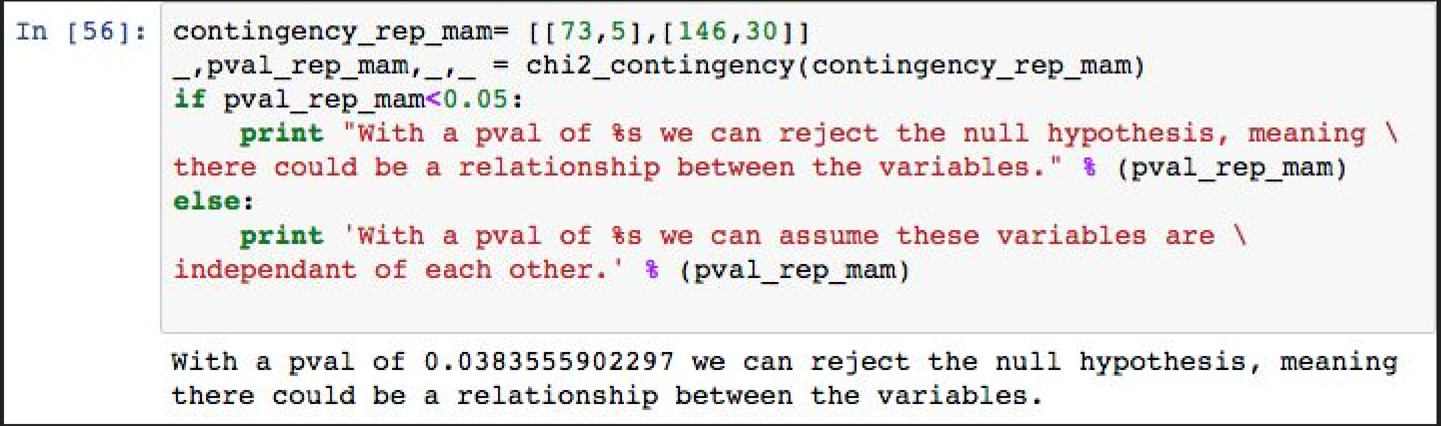
```
In [63]: contingency = [[146,30],[413,75],[72,7],[115,11],[73,5]]
         chi2_test = chi2,pval,dof,expected = chi2_contingency(contingency)
         chi2_test

Out[63]: (10.611356846184144,
          0.031297161931687301,
          4,
          array([[ 152.21119324,   23.78880676],
                 [ 422.04012672,   65.95987328],
                 [  68.32206969,   10.67793031],
                 [ 108.96937698,   17.03062302],
                 [  67.45723337,   10.54276663]]))
```



```
In [65]: contingency_bird_fish= [[413,75],[115,11]]
         test_1= _,pval_bird_fish,_,_ = chi2_contingency(contingency_bird_fish)
         pval_bird_fish
         test_1

Out[65]: (3.1338596463736459,
          0.076681995690571936,
          1,
          array([[ 419.64820847,   68.35179153],
                 [ 108.35179153,   17.64820847]]))
```

```
In [57]: contingency_mam_bird = [[146,30],[413,75]]
         test_2 = _, pval_mam_bird, _,_ = chi2_contingency(contingency_mam_bird)
         if pval_mam_bird<.05:
             print "With a pval of %s we can reject the null, meaning \
         there is a relationship in the difference that could mean \
         they're related." % (pval_mam_bird)
         else:
             print 'With a pval of %s we can assume these variables are \
         independant of each other.' % (pval_mam_bird)

         test_2

         With a pval of 0.687594809666 we can assume these variables are independa
         nt of each other.

Out[57]: (0.16170148316545574,
          0.68759480966613362,
          1,
          array([[ 148.1686747,   27.8313253],
                 [ 410.8313253,   77.1686747]]))
```

- Mammals are more likely to be protected
- Leads to more questions:
  - Do reptiles actually have less diseases or do they have different importance level?
  - What is our quality of observation data?

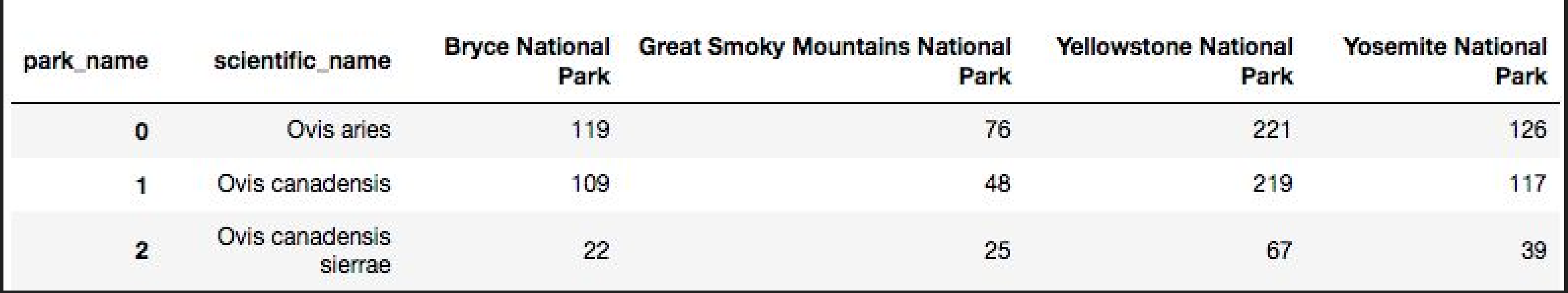


In [56]:

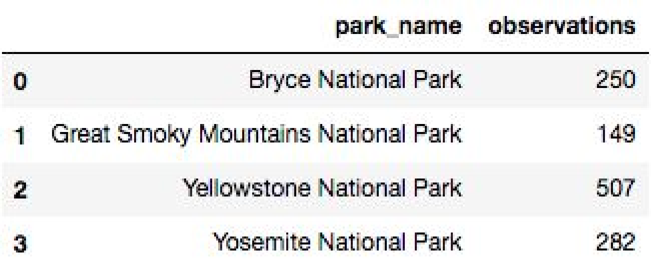
```
contingency_rep_mam= [[73,5],[146,30]]
_,pval_rep_mam,_,_ = chi2_contingency(contingency_rep_mam)
if pval_rep_mam<0.05:
    print "With a pval of %s we can reject the null hypothesis, meaning \
there could be a relationship between the variables." % (pval_rep_mam)
else:
    print 'With a pval of %s we can assume these variables are \
independant of each other.' % (pval_rep_mam)
```

```
With a pval of 0.0383555902297 we can reject the null hypothesis, meaning
there could be a relationship between the variables.
```

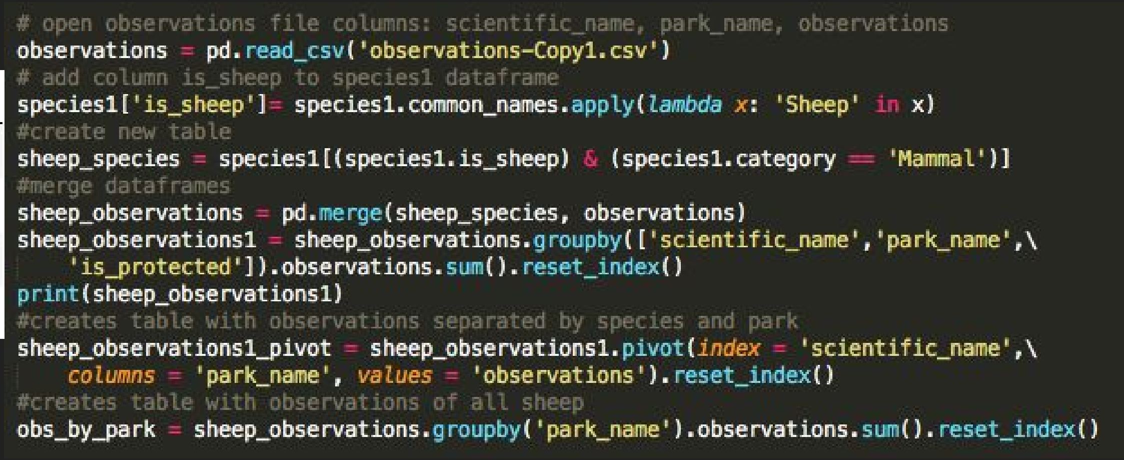
Observations of Sheep at National Parks



park_name	scientific_name	Bryce National Park	Great Smoky Mountains National Park	Yellowstone National Park	Yosemite National Park
0	Ovis aries	119	76	221	126
1	Ovis canadensis	109	48	219	117
2	Ovis canadensis sierrae	22	25	67	39

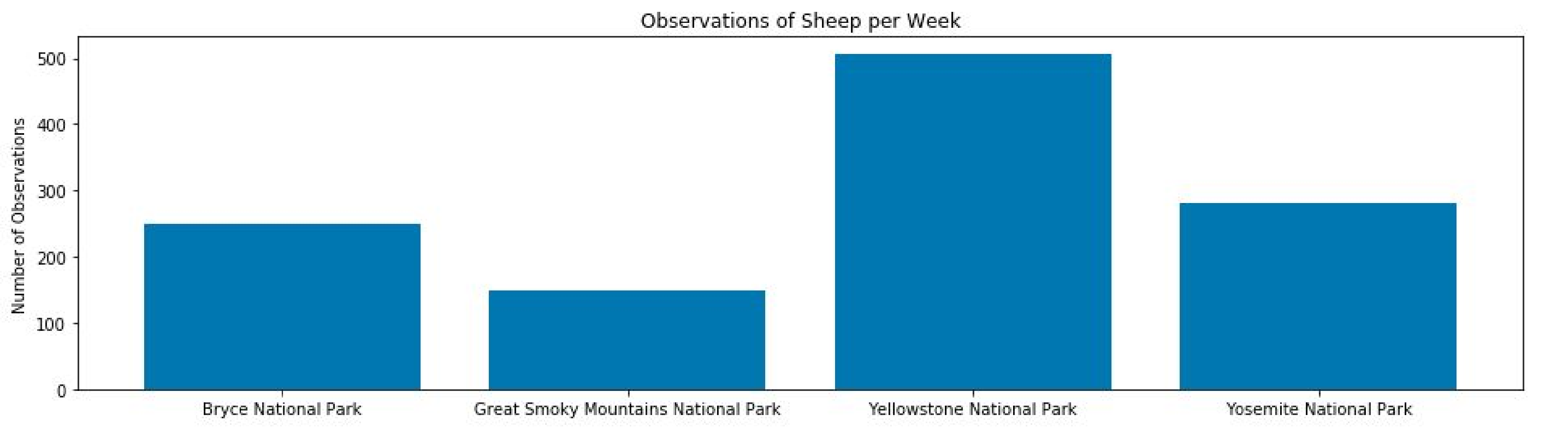


	park_name	observations
0	Bryce National Park	250
1	Great Smoky Mountains National Park	149
2	Yellowstone National Park	507
3	Yosemite National Park	282



```
# open observations file columns: scientific_name, park_name, observations
observations = pd.read_csv('observations-Copy1.csv')
# add column is_sheep to species1 dataframe
species1['is_sheep']= species1.common_names.apply(lambda x: 'Sheep' in x)
#create new table
sheep_species = species1[(species1.is_sheep) & (species1.category == 'Mammal')]
#merge dataframes
sheep_observations = pd.merge(sheep_species, observations)
sheep_observations1 = sheep_observations.groupby(['scientific_name','park_name',\
    'is_protected']).observations.sum().reset_index()
print(sheep_observations1)
#creates table with observations separated by species and park
sheep_observations1_pivot = sheep_observations1.pivot(index = 'scientific_name',\
    columns = 'park_name', values = 'observations').reset_index()
#creates table with observations of all sheep
obs_by_park = sheep_observations.groupby('park_name').observations.sum().reset_index()
```

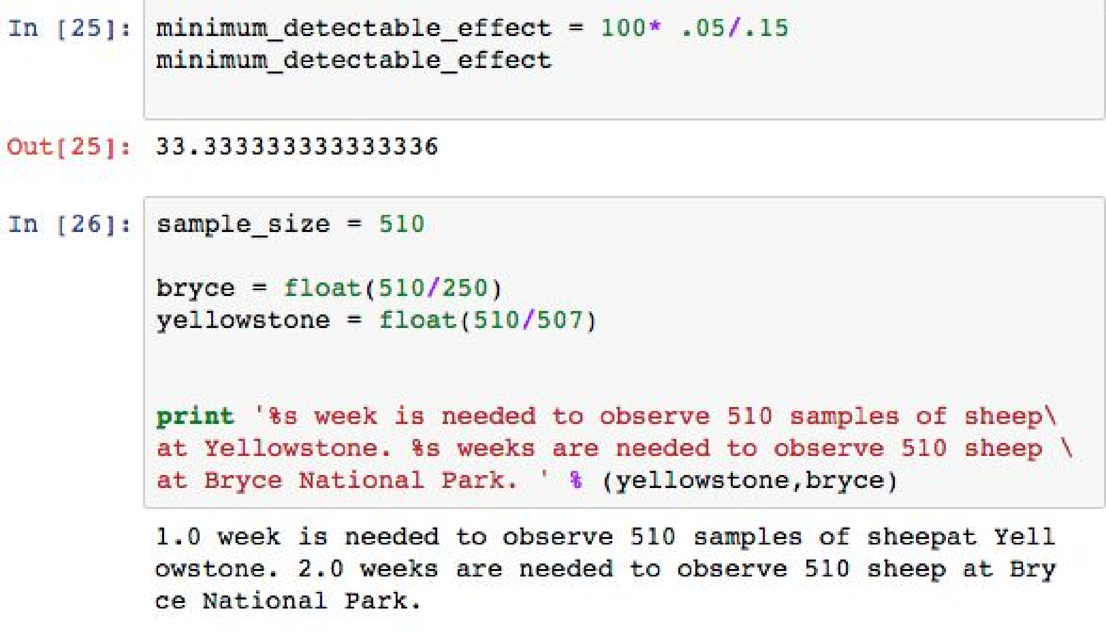
How long will the trial be?



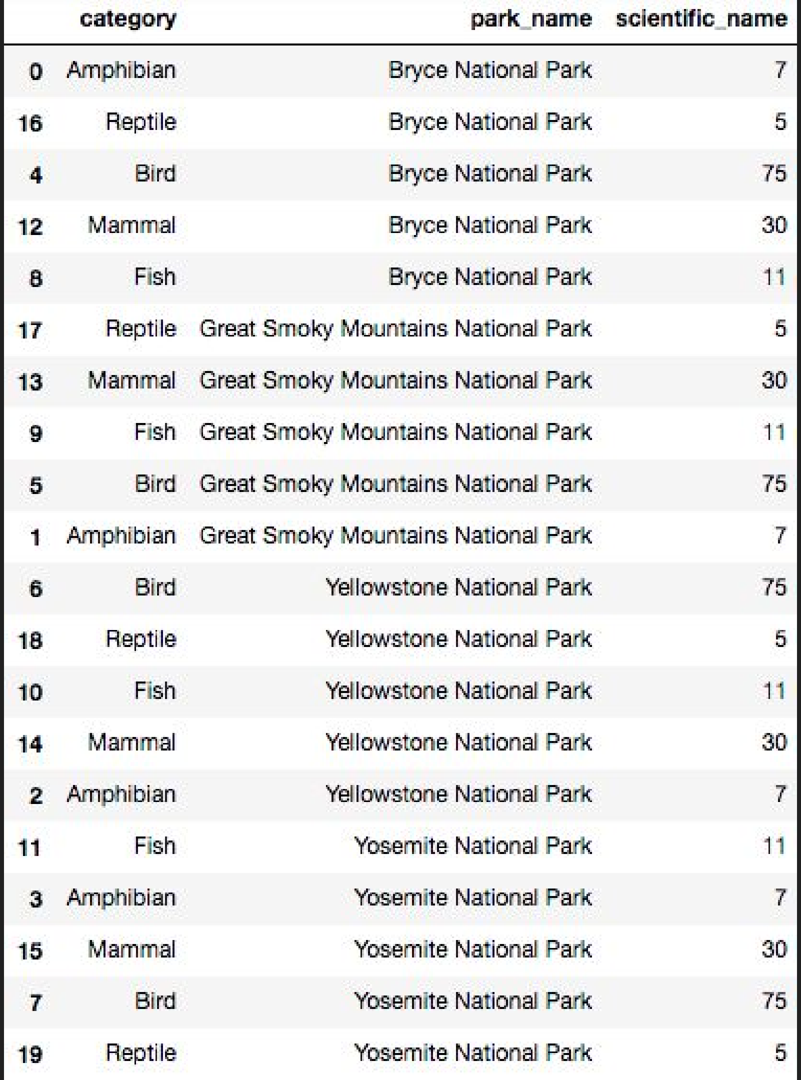
- We know that 15% of sheep have a disease at Bryce National Park.
- Yellowstone ran a program that brought their percentage of diseased sheep from 15% to 10%.
- We'd like to be able to detect reduction of at least 5 percentage points.

Use a sample size calculator

- To find sample size:
  - Baseline conversion rate
    - 15%
  - Minimum detectable effect
    - 33.33%
  - Statistical Significance
    - 90%

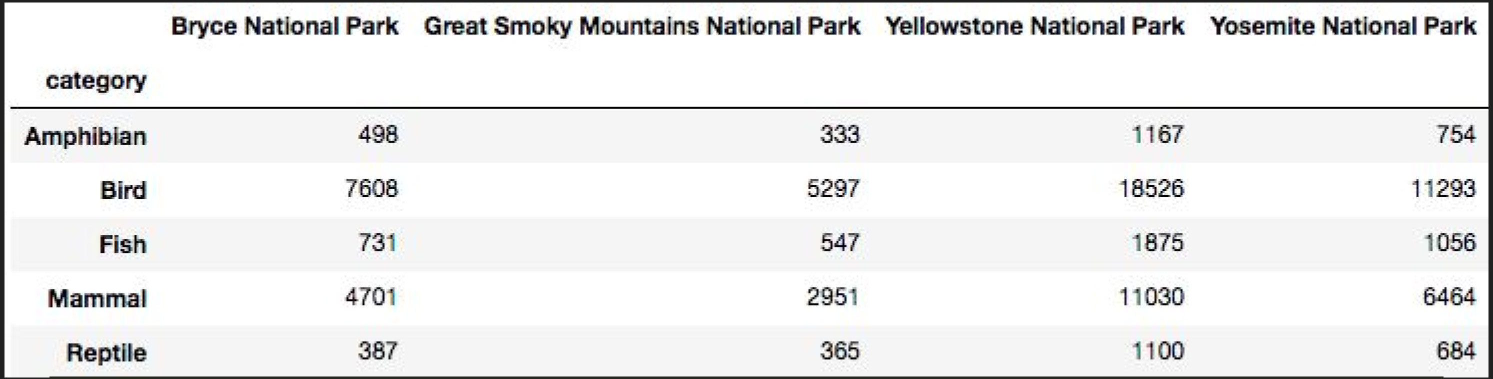


```
In [25]: minimum_detectable_effect = 100* .05/.15
         minimum_detectable_effect

Out[25]: 33.333333333333336

In [26]: sample_size = 510

         bryce = float(510/250)
         yellowstone = float(510/507)


         print '%s week is needed to observe 510 samples of sheep\
         at Yellowstone. %s weeks are needed to observe 510 sheep \
         at Bryce National Park. ' % (yellowstone,bryce)

         1.0 week is needed to observe 510 samples of sheepat Yell
         owstone. 2.0 weeks are needed to observe 510 sheep at Bry
         ce National Park.
```

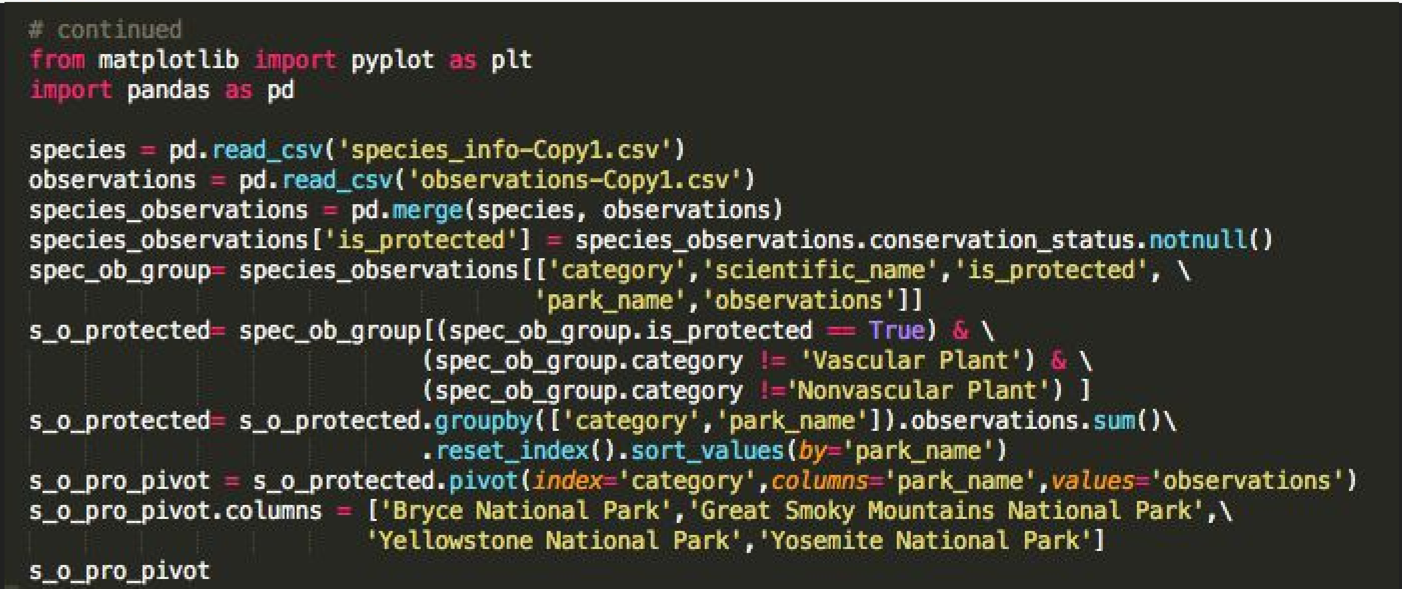


	category	park_name	scientific_name
0	Amphibian	Bryce National Park	7
16	Reptile	Bryce National Park	5
4	Bird	Bryce National Park	75
12	Mammal	Bryce National Park	30
8	Fish	Bryce National Park	11
17	Reptile	Great Smoky Mountains National Park	5
13	Mammal	Great Smoky Mountains National Park	30
9	Fish	Great Smoky Mountains National Park	11
5	Bird	Great Smoky Mountains National Park	75
1	Amphibian	Great Smoky Mountains National Park	7
6	Bird	Yellowstone National Park	75
18	Reptile	Yellowstone National Park	5
10	Fish	Yellowstone National Park	11
14	Mammal	Yellowstone National Park	30
2	Amphibian	Yellowstone National Park	7
11	Fish	Yosemite National Park	11
3	Amphibian	Yosemite National Park	7
15	Mammal	Yosemite National Park	30
7	Bird	Yosemite National Park	75
19	Reptile	Yosemite National Park	5

Observations of Protected Species per Park



category	Bryce National Park	Great Smoky Mountains National Park	Yellowstone National Park	Yosemite National Park
Amphibian	498	333	1167	754
Bird	7608	5297	18526	11293
Fish	731	547	1875	1056
Mammal	4701	2951	11030	6464
Reptile	387	365	1100	684



```
# continued
from matplotlib import pyplot as plt
import pandas as pd

species = pd.read_csv('species_info-Copy1.csv')
observations = pd.read_csv('observations-Copy1.csv')
species_observations = pd.merge(species, observations)
species_observations['is_protected'] = species_observations.conservation_status.notnull()
spec_ob_group= species_observations[['category','scientific_name','is_protected', \
                                     'park_name','observations']]
s_o_protected= spec_ob_group[(spec_ob_group.is_protected == True) & \
                             (spec_ob_group.category != 'Vascular Plant') & \
                             (spec_ob_group.category !='Nonvascular Plant') ]
s_o_protected= s_o_protected.groupby(['category','park_name']).observations.sum()\
                             .reset_index().sort_values(by='park_name')
s_o_pro_pivot = s_o_protected.pivot(index='category',columns='park_name',values='observations')
s_o_pro_pivot.columns = ['Bryce National Park','Great Smoky Mountains National Park',\
                         'Yellowstone National Park','Yosemite National Park']
s_o_pro_pivot
```

resources

https://docs.scipy.org/doc/scipy-0.15.1/reference/generated/scipy.stats.chi2_contingency.html

http://www.ling.upenn.edu/~clight/chisquared.htm

https://www.optimizely.com/sample-size-calculator/