```
species = pd.read_csv('species_info.csv')
print(species.head())
```

```
species_count = species.scientific_name.nunique()
print(species_count) #A: 5541
```

```
species_type = species.category.unique()
print(species_type) #Total = 7
```

```
conservation_statuses = species.conservation_status.unique()
print(conservation_statuses) #Count = 4 + nan (5)
```

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

```
#group by won't count NaN. The below will add these to a new categoru 'No Intervention'
species.fillna('No Intervention', inplace = True)
conservation_counts_fixed =
species.groupby('conservation_status').scientific_name.nunique().reset_index()
print(conservation_counts_fixed)
```

```
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.scientific_name)),
protection_counts.scientific_name)

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

plt.show()
plt.savefig('Conservation Status by Species.png')
```

```
#Create a new column in species called is_protected
species['is_protected'] = species.conservation_status != 'No Intervention'

#Now group by both category and is_protected
category_counts =
species.groupby(['category','is_protected']).scientific_name.nunique().reset_index()
print(category_counts.head())

#Pivot data to make it easier to view
category_pivot = category_counts.pivot(
  columns = 'is_protected',
  index = 'category',
  values = 'scientific_name').reset_index()
print(category_pivot)
```

```
#Chi-Squared Test for significance
#Contigency table:
contingency = [[30,146], [75,413]]
#Run test for Mammals vs Birds:
_, pval, _, _ = chi2_contingency(contingency)
print(pval) #A: 0.687594809666 = not significant

#Contigency table:
contingency = [[30,146],[5,73]]
#Run test for Mammals vs Reptiles:
_, pval_reptile_mammal, _, _ = chi2_contingency(contingency)
print(pval_reptile_mammal) #A: 0.0383555902297 = significant
```

```
#Use apply and a lambda function to create a new column in species called is_sheep which is True if the common_names contains 'Sheep', and False otherwise.
species['is_sheep'] = species.common_names.apply(lambda x: True if 'Sheep' in x else False)
```

```
#Select the rows of species where is_sheep is True
species_is_sheep = species.is_sheep == True
print(species_is_sheep.head())
```

```
#Many of the results are actually plants. Select the rows of species where is_sheep is True and category is Mammal
sheep_species = species[(species.is_sheep == True) & (species.category == 'Mammal')]
print(sheep_species)
```

```
#Now merge sheep_species with observations to get a DataFrame with observations of sheep.
sheep_observations = pd.merge(sheep_species, observations)
print(sheep_observations.head())
```

```
#How many total sheep sightings (across all three species) were made at each national park? Use groupby to get the sum of observations for each park_name
obs_by_park = sheep_observations.groupby('park_name').observations.sum().reset_index()
print(obs_by_park) #=This is the total number of sheep observed in each park over the past 7 days
```

```
plt.figure(figsize=(16,4))
ax = plt.subplot()
plt.bar(range(len(obs_by_park)), obs_by_park.observations)

ax.set_xticks(range(len(obs_by_park)))
ax.set_xticklabels(obs_by_park.park_name)
plt.ylabel('Number of Observations')
plt.title('Observations of Sheep per Week')

plt.show()
```

```
#They want to be able to detect reductions of at least 5 percentage
#The only information that the scientists currently have is that last year it was recorded that 15% of sheep at Bryce National Park have foot and mouth disease. Using this value and the sample size calculator in the browser window on the right, you will need to calculate the number of sheep that they would need to observe from each park to make sure their foot and mouth percentages are significant. Use the default level of significance (90%).
#What is the baseline percentage of this sample size determination?
baseline = 15

#Calculate "Minimum Detectable Effect".
#if we wanted to observe an x% change with confidence, our minimum detectable effect would be equal to 100 * x / baseline.
minimum_detectable_effect = 100*5./15

sample_size_per_variant = 870

#507 = Yellstone observations over 7 days
yellowstone_weeks_observing = sample_size_per_variant/507.
print(yellowstone_weeks_observing) #A: 1.71597633136 weeks

#250 = Bryce observations over 7 days
bryce_weeks_observing = sample_size_per_variant/250.
print(bryce_weeks_observing) #A: 3.48 weeks
```