Assignment 1 - Mosquito - Ben Polasek

January 21, 2019

In [1]:
```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
#load in data
df = pd.read_csv('mosquitos_data.csv')
df
```

Out[2]:

	Response	Treatment
0	27	Beer
1	20	Beer
2	21	Beer
3	26	Beer
4	27	Beer
5	31	Beer
6	24	Beer
7	21	Beer
8	20	Beer
9	19	Beer
10	23	Beer
11	24	Beer
12	28	Beer
13	19	Beer
14	24	Beer
15	29	Beer
16	18	Beer
17	20	Beer
18	17	Beer
19	31	Beer
20	20	Beer
21	25	Beer
22	28	Beer
23	21	Beer
24	27	Beer
25	21	Water
26	22	Water

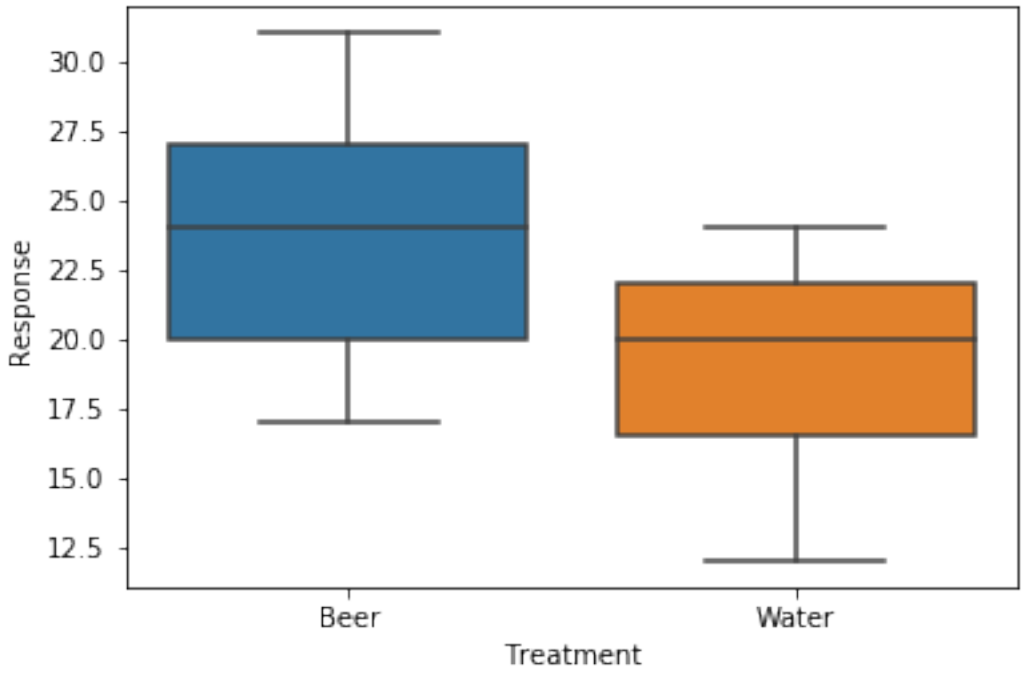
```
27          15     Water
28          12     Water
29          21     Water
30          16     Water
31          19     Water
32          15     Water
33          22     Water
34          24     Water
35          19     Water
36          23     Water
37          13     Water
38          22     Water
39          20     Water
40          24     Water
41          18     Water
42          20     Water
```

In [3]:
```
#seperate treatments for ease of use later
beer = df[df['Treatment'] == 'Beer']
water = df[df['Treatment'] == 'Water']
beer
```

Out[3]:
```
    Response Treatment
0          27      Beer
1          20      Beer
2          21      Beer
3          26      Beer
4          27      Beer
5          31      Beer
6          24      Beer
7          21      Beer
8          20      Beer
9          19      Beer
10         23      Beer
11         24      Beer
12         28      Beer
13         19      Beer
14         24      Beer
15         29      Beer
16         18      Beer
17         20      Beer
18         17      Beer
19         31      Beer
20         20      Beer
21         25      Beer
22         28      Beer
23         21      Beer
24         27      Beer
```

0.1 1. Box plot of treatments

```
In [4]: a = sns.boxplot(x = 'Treatment', y = 'Response', data = df)
```



0.2 2. Q. What does the graph reveal about the data for both groups? Is there an association between beer consumption and attractiveness to mosquitoes?

We can observe that the mean number of mosquitoes is higher for the treatment of beer, and that there is also a slightly higher range or variability for the beer treatment. There seems to be an association between beer consumption and attractiveness to mosquitoes, however, we do not yet know if it is different enough from water consumption to conclude with confidence that beer makes you more attractive to mosquitoes than water

0.3 3. Basic statistic measures for beer and water treatment

```
In [5]: beer.describe()
```

```
Out[5]:          Response
        count   25.000000
        mean    23.600000
        std      4.133199
        min     17.000000
        25%     20.000000
        50%     24.000000
        75%     27.000000
        max     31.000000
```

```
In [6]: beer.median()

Out[6]: Response    24.0
        dtype: float64

In [7]: water.describe()

Out[7]:          Response
        count  18.000000
        mean   19.222222
        std     3.671120
        min    12.000000
        25%    16.500000
        50%    20.000000
        75%    22.000000
        max    24.000000

In [8]: water.median()

Out[8]: Response    20.0
        dtype: float64
```

0.4 4. Explanation

The average number of mosquitoes was higher by ~4 mosquitoes for the beer treatment compared to the water treatment, with the mean number of mosquitoes for the beer treatment being 23.6 and 19.2 for water

The median value was also 4 higher in the beer treatment over water. The medians that were calculated were 24 for the beer treatment and 20 for the water treatment

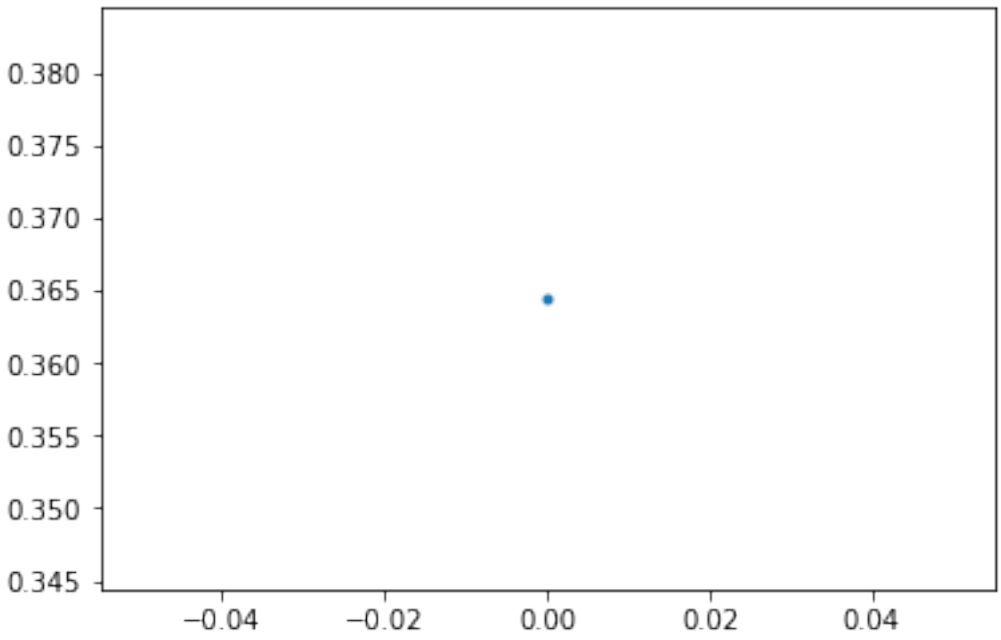
The standard deviation was higher for the beer treatment (4.13) compared to the water tretment (3.67) which is consistent with what we observed from the box plot, that is, the spread of values greater for the beer treatment over the water treatment

0.5 5. Random Permutation Test

```
In [9]: #Step 1. Shuffle Data
        responses = df.Response.values
        treatment = df.Treatment.values
        responses

Out[9]: array([27, 20, 21, 26, 27, 31, 24, 21, 20, 19, 23, 24, 28, 19, 24, 29, 18,
               20, 17, 31, 20, 25, 28, 21, 27, 21, 22, 15, 12, 21, 16, 19, 15, 22,
               24, 19, 23, 13, 22, 20, 24, 18, 20], dtype=int64)

In [10]: df.Treatment.describe()

Out[10]: count       43
         unique       2
         top       Beer
         freq        25
         Name: Treatment, dtype: object
```

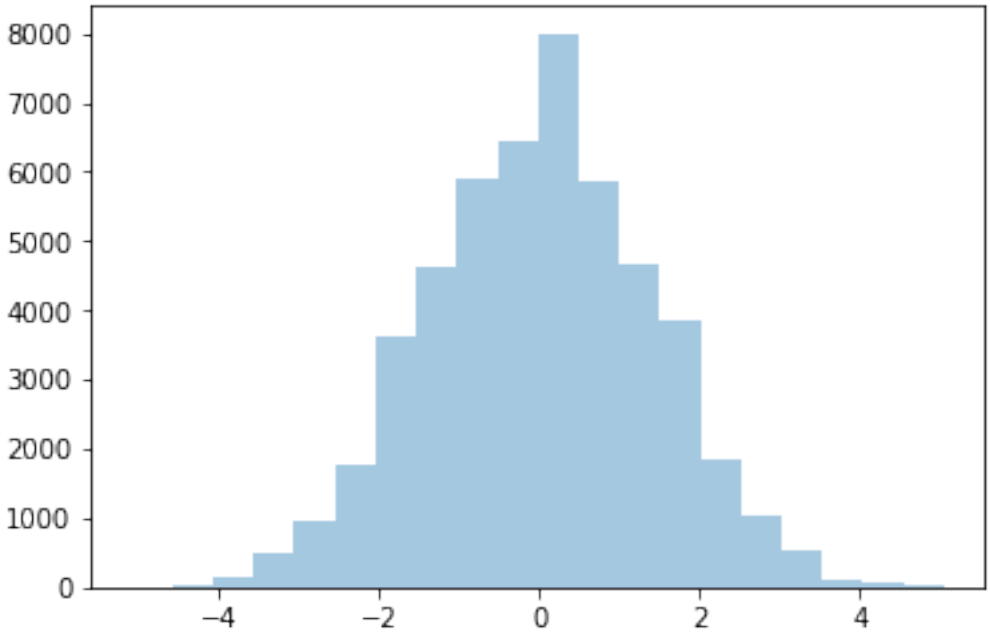
We have 43 samples and 25 are beer, so 18 are water. We can then split our array of shuffled and unlabled responses into two arrays of size 25 and 18 to represent beer and water respectively.

```
In [11]: np.random.shuffle(responses)
         responses

Out[11]: array([28, 19, 20, 24, 27, 24, 19, 19, 20, 18, 13, 27, 24, 28, 27, 22, 20,
                21, 17, 19, 21, 31, 20, 22, 18, 15, 23, 22, 16, 12, 29, 21, 25, 26,
                20, 24, 21, 21, 24, 20, 15, 31, 23], dtype=int64)

In [12]: responses_beer = responses[:25]
         responses_water = responses[25:]

In [13]: responses_beer.size

Out[13]: 25

In [14]: responses_water.size

Out[14]: 18

In [15]: #Step 2. Computer difference in mean
         mean_beer = responses_beer.mean()
         mean_beer

Out[15]: 21.92

In [16]: mean_water = responses_water.mean()
         mean_water

Out[16]: 21.555555555555557

In [17]: mean_diff = mean_beer - mean_water

In [18]: mean_arr= []
         mean_arr.append(mean_diff)
         mean_arr

Out[18]: [0.36444444444444457]

In [19]: #Step 3. Plot it
         plt.plot(mean_arr, '.')

Out[19]: [<matplotlib.lines.Line2D at 0x1fd1eef82e8>]
```



0.5.1 Putting it all together we obtain the following algorithm

```
In [20]: responses = df.Response.values
         treatment = df.Treatment.values
         mean_arr = []
         for i in range (0, 50000):
             np.random.shuffle(responses)
             responses_beer = responses[:25]
             responses_water = responses[25:]

             mean_beer = responses_beer.mean()
             mean_water = responses_water.mean()

             mean_diff = mean_beer - mean_water

             mean_arr.append(mean_diff)

         sns.distplot(mean_arr,bins=20, kde=False)

Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x1fd1ef30eb8>
```


8000
7000
6000
5000
4000
3000
2000
1000
0
−4
−2
0
2
4