project2_sta3030

April 19, 2018

```
In [49]: import numpy as np, pandas as pd, matplotlib.pyplot as plt, csv, matplotlib.mlab as ml
         %matplotlib inline
         import scipy.stats as stat
         import timeit
```

```
In [50]: bristol_86 = np.genfromtxt("Bristol_86.csv",delimiter=",", skip_header=1)
```

We are now goign to do the exploratory analysis

```
In [51]: After =bristol_86[:,1]
         Before = bristol_86[:,0]

         difference = Before - After

         BeMean = Before.mean()
         AfMean = After.mean()
         diffMean = difference.mean()
         BeSTD = Before.std()
         AfSTD = After.std()
         diffSTD = difference.std()
```

Now we are going to plot a histogram of the difference

```
In [52]: BeMean

Out[52]: 12.49

In [53]: AfMean

Out[53]: 11.949999999999999

In [54]: diffMean

Out[54]: 0.54000000000000004

In [55]: BeSTD

Out[55]: 1.5977797094718658
```

In [56]: AfSTD

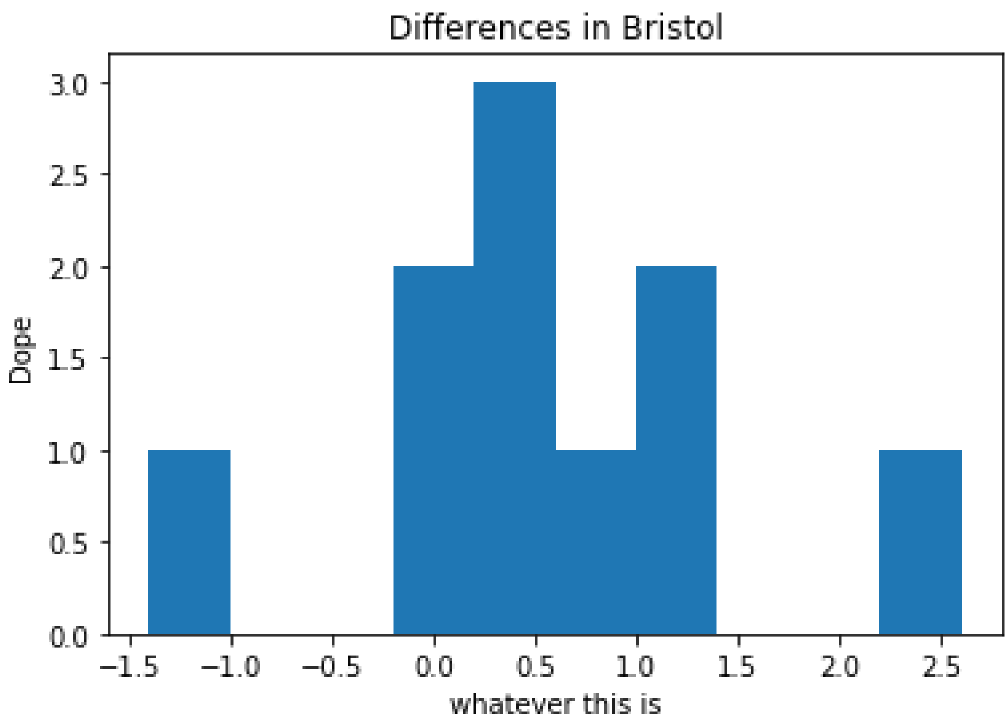
Out[56]: 1.8575521526998913

In [57]: diffSTD

Out[57]: 0.96353515763567288

```
In [58]: plt.hist(difference)
         plt.title("Differences in Bristol")
         plt.xlabel("whatever this is")
         plt.ylabel("Dope")

         #fig = plt.gcf()
```

Out[58]: Text(0,0.5,'Dope')



paired t-test based on collected data

```
In [59]: Obsdiff = diffMean

         stanError = diffSTD/np.sqrt(len(difference))

         t_stat = (diffMean-0)/stanError

         p_val = stat.t.sf(np.abs(t_stat), len(difference)-1)
```

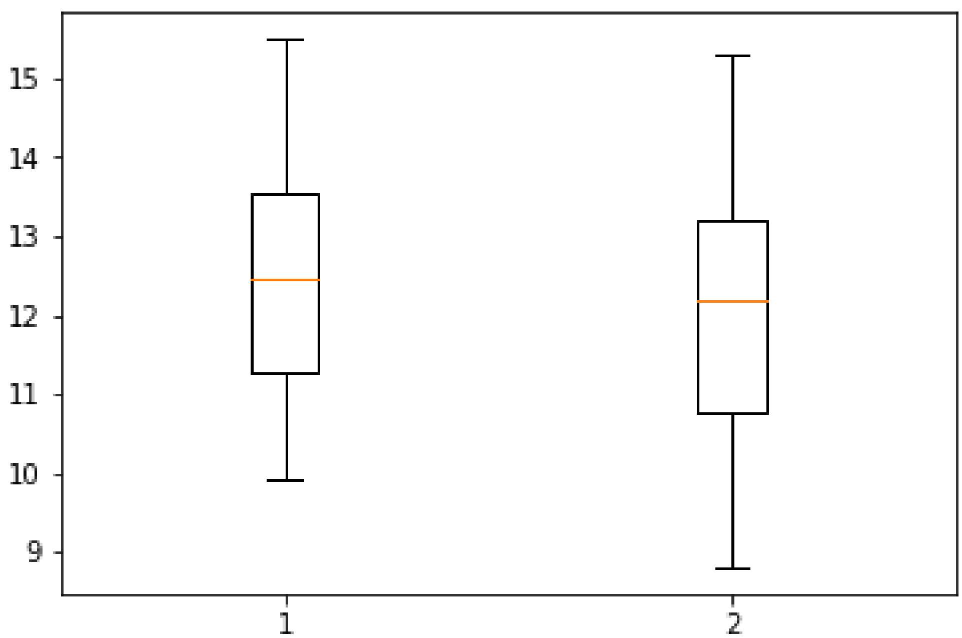
```
In [60]: p_val
```

```
Out[60]: 0.055057065970771 32
```

We are now showing boxplots

```
In [61]: plt.boxplot(bristol_86)
```

```
Out[61]: {'boxes': [<matplotlib.lines.Line2D at 0x104702c50>,
           <matplotlib.lines.Line2D at 0x104716748>],
          'caps': [<matplotlib.lines.Line2D at 0x10470e6a0>,
           <matplotlib.lines.Line2D at 0x10470eac8>,
           <matplotlib.lines.Line2D at 0x104720438>,
           <matplotlib.lines.Line2D at 0x104720860>],
          'fliers': [<matplotlib.lines.Line2D at 0x104716358>,
           <matplotlib.lines.Line2D at 0x1047270f0>],
          'means': [],
          'medians': [<matplotlib.lines.Line2D at 0x10470eef0>,
           <matplotlib.lines.Line2D at 0x104720c88>],
          'whiskers': [<matplotlib.lines.Line2D at 0x104702da0>,
           <matplotlib.lines.Line2D at 0x10470e278>,
           <matplotlib.lines.Line2D at 0x104716ba8>,
           <matplotlib.lines.Line2D at 0x104716fd0>]}
```



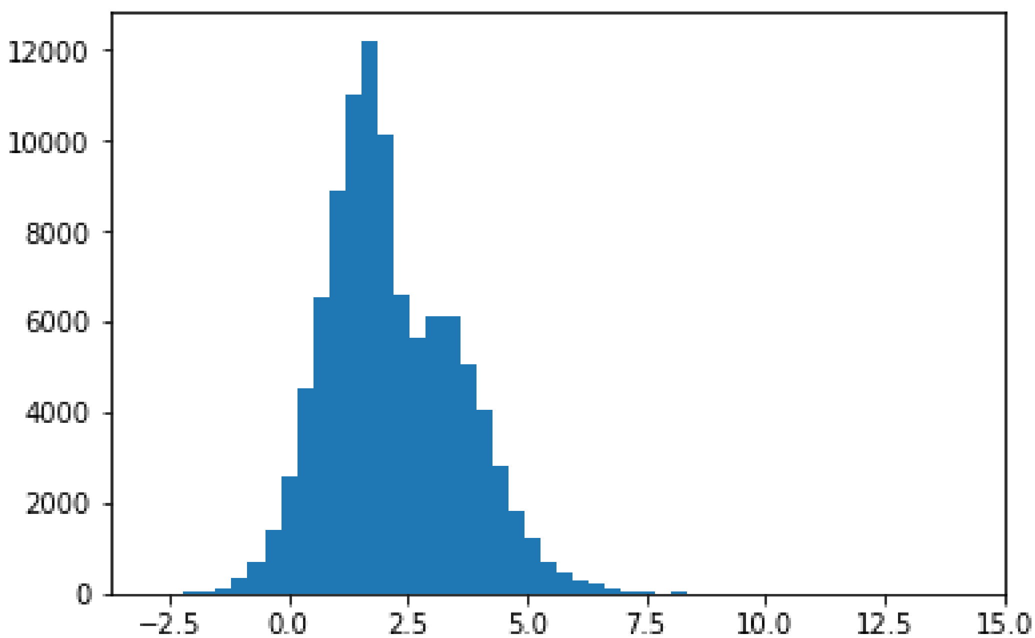
The main event of bootstrapping begins

```
In [62]: bodily = np.random.choice(difference, 10)
```

In [63]:
```
##bootstrap

def bootstrapping(data, totalSamples, output):
    for i in range(1,totalSamples):
        bodily = np.random.choice(data, len(data))
        standErring =np.std(bodily)/np.sqrt(len(bodily))
        tStat = np.mean(bodily)/standErring

        output.append(tStat)
    return (output)
```

In [64]:
```
start = timeit.timeit()
theHistogram = plt.hist(bootstrapping(difference,100000,[]), bins=50)

print (timeit.timeit() - start)
```

```
-0.025635201999648416
```



The mean is normal for 100000 bootstrapped samples.

In [65]:
```
summing =0
p_val = stat.t.sf(np.abs(t_stat), len(difference)-1)
def p_value(output):
    for i in range(0,len(output)):
        if output[i]> abs(p_val):
            ++summing
    return (summing/len(output))
```

In [66]:
```
outing = bootstrapping(difference, 10000,[])
print(p_value(outing))
```

```
0.0
```

Teh sampling errors and bias

In [67]:
```
np.std(outing)

##The bias
_bias = np.mean(outing)- diffMean

##corrected bias

corrBias = diffMean - _bias
```

In [68]:
```
print(_bias)
print(corrBias)
```

```
1.63899638015
-1.09899638015
```