STATISTICS USING PYTHON

July 7, 2022

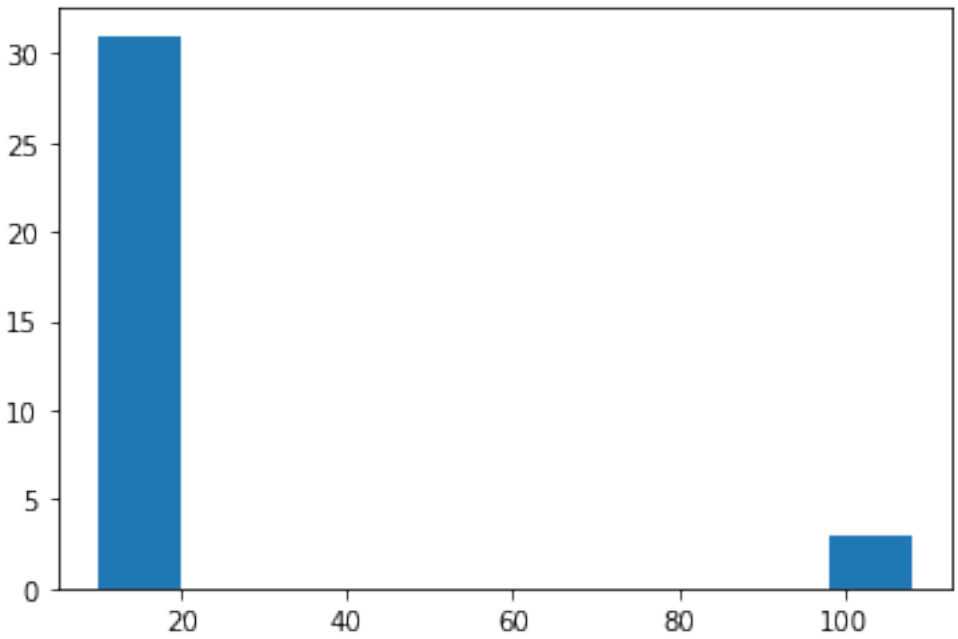
[ ]:
```
#statistics applied in python :
```

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

[3]:
```
dataset =␣
↪[11,10,12,14,12,15,14,13,15,102,12,14,17,19,107,10,13,12,14,12,108,12,11,14,13,15,10,15,12,
```

[4]:
```
plt.hist(dataset)
#when you look there is an outlier
```

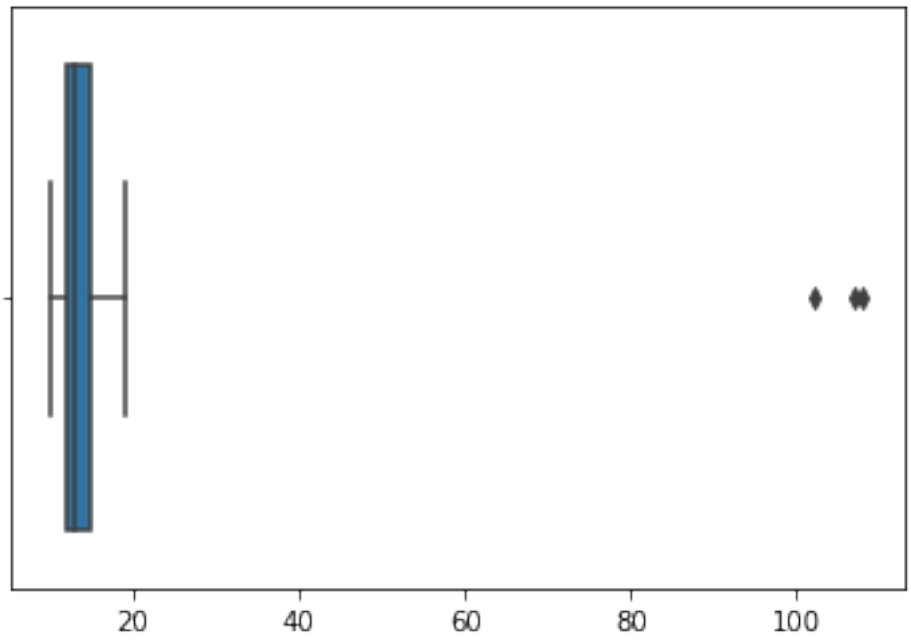
[4]:
```
(array([31.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  3.]),
 array([ 10. ,  19.8,  29.6,  39.4,  49.2,  59. ,  68.8,  78.6,  88.4,
         98.2, 108. ]),
 <BarContainer object of 10 artists>)
```



[6]:
```
sns.boxplot(dataset)
#we can see towards right there is a big outlier
```

```
C:\Users\91936\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

[6]: <AxesSubplot:>



[ ]:
```
#performing z test using python :
```

[ ]:
```
# suppose the IQ in a certain population is normally distributed with a mean =␣
 ↪100 , std dev = 15
# the researcher wants to know if a new drig affeccts iq levels , so he␣
 ↪recruits 20 patients and tries it.
# solve the following question using z test and find if it causes a significant␣
 ↪difference in iq levels.
```

[8]:
```
from statsmodels.stats.weightstats import ztest as ztest

#iq levels of 20 patients
```

```
data = [88,92,94,94,96,97,97,99,99,105,109,109,109,110,112,112,113,114,115]
ztest(data,value = 100)
#in the output the first value is the z test value and the second value is p␣
 ↪value of significnce value
#the p value is 0.08 compare it with alpha value if its less than alpha value␣
 ↪reject the null hypothesis
```

[8]:
```
(1.6976532110033402, 0.0895732335076963)
```

[ ]:
```
# T TEST
```

[20]:
```
ages =␣
 ↪[10,20,35,50,28,40,55,18,16,55,30,25,43,18,30,28,14,24,16,17,32,35,26,27,65,18,43,23,21,20,
#calculating the mean value
ages_mean = np.mean(ages)
print("the mean of the ages is : ",ages_mean)
#so here we are not given the population std deviation so we are taking a␣
 ↪sample and finding the t test if the mean value is similar to that of t test
sample_size = 10
ages_sample = np.random.choice(ages,sample_size)#we picked 10 choices
print("the random 10 samples are : ",ages_sample)
from scipy.stats import ttest_1samp
ttest_1samp(ages_sample,30) #30 is the mean we got earlier and we are comparing␣
 ↪it with the random 10 samples if the mean is similar or not
#if the p value is lesser than aplha accept the null hyporthesis
```

```
the mean of the ages is :  30.34375
the random 10 samples are :  [30 17 16 35 23 18 23 18 27 35]
```

[20]:
```
Ttest_1sampResult(statistic=-2.5283861426780696, pvalue=0.0323208437243828)
```

[19]:
```
np.mean(ages_sample)
```

[19]:
```
29.6
```

[ ]:
```

```