11-05_MF_stats.brew

September 27, 2023

[1]:
```
import numpy as np
import json
import matplotlib.pyplot as plt
%matplotlib inline
```

[2]:
```
with open('brew.json') as f:
  data = np.array(json.load(f))
```

[3]:
```
#data = data[:data.size>>1]
```

[4]:
```
data
```

[4]:
```
array([210.58431408, 202.40094325, 205.8724279 , 213.4453592 ,
       211.20534794, 194.13633272, 205.70053051, 199.09185675,
       199.38068689, 202.46359101, 200.86426143, 208.72564104,
       204.56622635, 200.7300501 , 202.6631794 , 202.00204596,
       208.96447444, 198.76905042, 201.87840621, 194.87542556,
       184.6820611 , 203.92171157, 205.18661719, 195.54700988,
       213.61852774, 191.27380595, 200.2745511 , 198.8768969 ,
       209.19667529, 208.81615262, 200.92968455, 202.26897512,
       194.67328551, 188.11522119, 197.9125271 , 200.93809381,
       207.38174408, 207.21427909, 197.6760391 , 198.1861835 ,
       193.70868221, 191.47989238, 189.76237886, 211.70465237,
       196.94208691, 197.37155419, 192.48322784, 204.66494213,
       190.31661291, 198.72355832, 194.62720063, 202.32141499,
       196.93516917, 192.9162069 , 199.83090663, 202.56999122,
       200.39910333, 201.81483139, 196.19406744, 197.823553  ,
       195.96523731, 197.84268103, 195.12112231, 189.64230439,
       201.06455685, 197.58931438, 190.21880992, 202.77669353,
       194.55620981, 200.31167237, 204.37454337, 200.77389746,
       206.83640411, 192.59104508, 202.41404985, 195.89113945,
       194.7752171 , 196.52690201, 198.13068481, 200.33699205,
       193.00910096, 205.40495892, 202.79397464, 190.78253788,
       208.92951316, 211.37533506, 207.07267743, 198.92045099,
       193.57548427, 206.32671036, 197.58093832, 207.33467042,
       201.24964987, 205.85983422, 202.13819838, 204.23943901,
       200.06300012, 210.71522296, 200.76147256, 202.41193618])
```

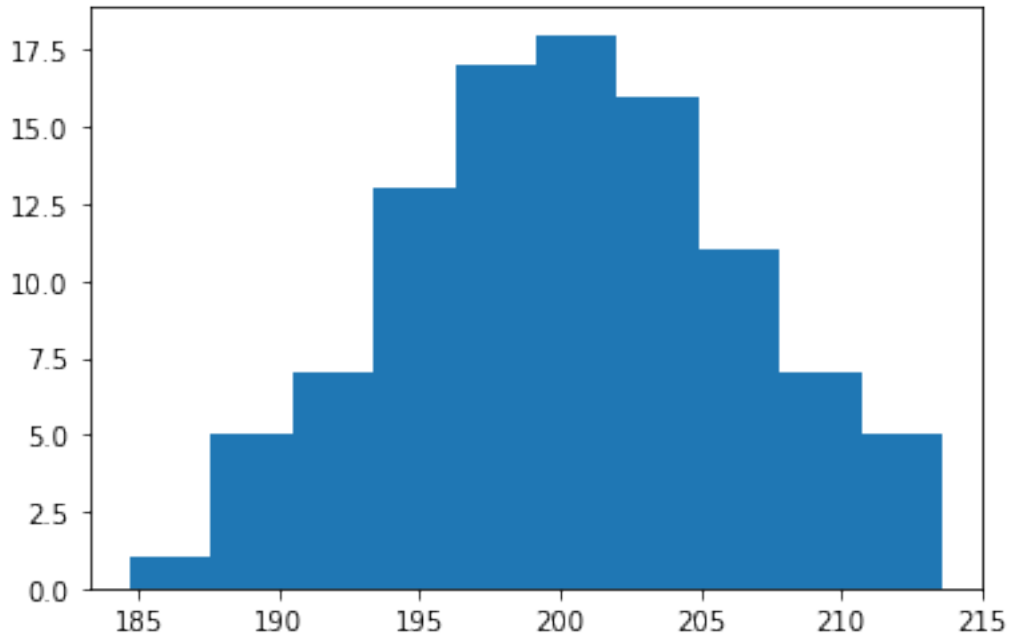
[5]:
```
data.shape
```

[5]:
```
(100,)
```

[6]:
```
mean = np.mean(data)
mean
```

[6]:
```
200.35884809320694
```

[7]:
```
plt.hist(data);
```



[8]:
```
import scipy.stats
```

[9]:
```
std_dev = np.std(data)
std_dev
```

[9]:
```
6.047293468299477
```

[10]:
```
N = data.size
S_n = std_dev / np.sqrt(N)
S_n
```

[10]:
```
0.6047293468299477
```

[11]:
```
confidence = 0.99
df = N - 1
t = scipy.stats.t.ppf(confidence, df)
t
```

[11]: 2.3646058614359737

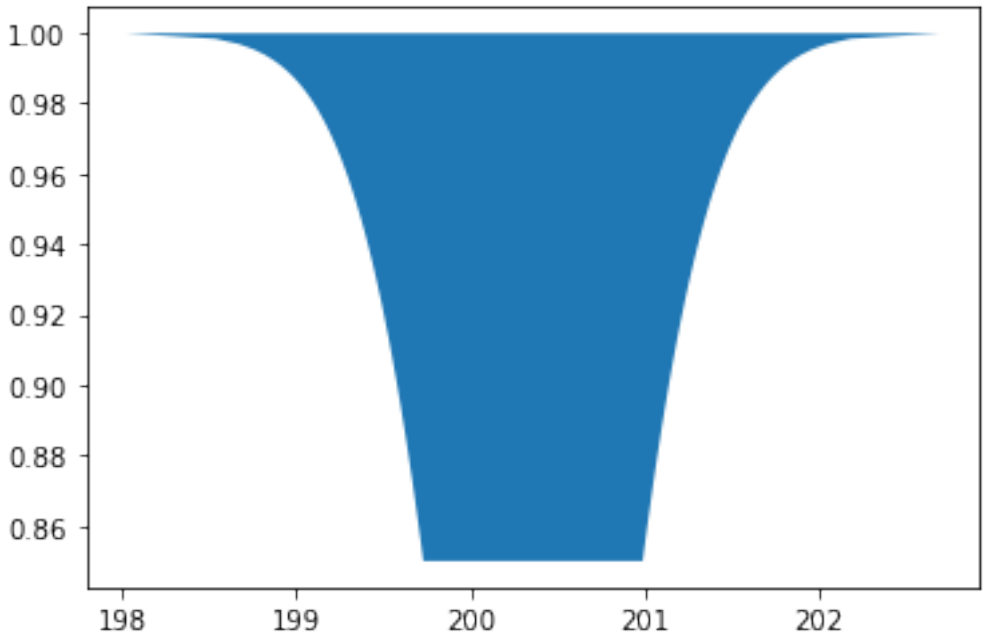
[12]:
```
mean + np.array([-1, 1]) * t * S_n
```

[12]: array([198.92890154, 201.78879465])

[13]:
```
confidences = np.linspace(85, 99.99, 100) / 100
lower_bounds = np.zeros_like(confidences)
upper_bounds = np.zeros_like(confidences)
for i, conf in enumerate(confidences):
  ti = scipy.stats.t.ppf(conf, df)
  lower_bounds[i] = mean - ti * S_n
  upper_bounds[i] = mean + ti * S_n
```

[14]:
```
plt.fill_betweenx(confidences, lower_bounds, upper_bounds)
```

[14]: <matplotlib.collections.PolyCollection at 0x7f18789838e0>



[15]:
```
mean + np.array([-1, 1]) * scipy.stats.norm.ppf(confidence) * S_n
```

[15]: array([198.95203726, 201.76565892])

[ ]: