t-distribution

August 19, 2018

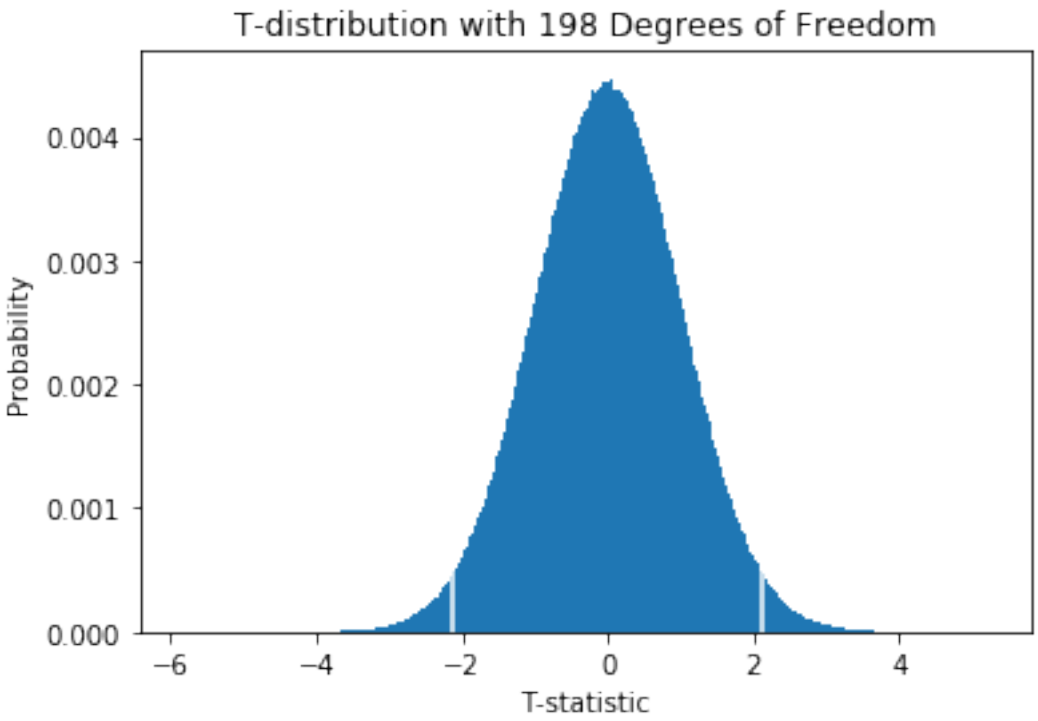
In [10]:
```
# Assignment 1
# Alla Topp
# Visualizing the t-statistics
```

In [16]:
```
import pylab
import random
import scipy
import math
import matplotlib

tStat = -2.13165598142 #
tDist = []
numBins = 1000
for i in range(10000000):
    tDist.append(scipy.random.standard_t(198))

pylab.hist(tDist, bins = numBins,
           weights = pylab.array(len(tDist)*[1.0])/len(tDist))
pylab.axvline(tStat, color = 'w')
pylab.axvline( -tStat, color = 'w')
pylab.title('T-distribution with 198 Degrees of Freedom')
pylab.xlabel('T-statistic')
pylab.ylabel('Probability')
```

Out[16]:
```
Text(0,0.5,'Probability')
```



In [ ]:
```
# Results:
# This plot shows how t-statistics for PED-X lies on the distribution
# Assuming that the sum of the fractions to the left and right of the white lines
# equals the probability of getting a value at least as extrime as the
# observed value (p-value). We might see that the actual population means
# of the treatment and controls are identical.
```

Baye's Theorem

August 19, 2018

In [1]:
```
# Assignment 2
# Alla Topp
# How comfortable you should be about serving mushrooms?
```

In [2]:
```
def Bayes(prior, conditional): # function named after baye`s theorem
    A_cap_B, A_cap_Bp = [conditional[i]*prior[i] for i in range(len(prior))] # A|B, A|
    posterior = A_cap_B /(A_cap_B + A_cap_Bp) # calculates the probability of having n
    return posterior

prior = [0.2, 0.8]  # Probability ANPo, APo
conditional = [0.95, 0.05]  # Sure not poisoned, poisoned

# returns probability of B for given A (probability of having not poisoning mushrooms)
Bayes(prior, conditional)
```

Out[2]:
```
0.8260869565217391
```

In [3]:
```
# Results:
# Probability of not having a poisoning mushrooms is ~83%
# A_cap_B = P(Poisoned)*P(Positive/Poison)
# A_cap_Bp = P(Positive/Not poisoned)*(1-P(poisoned))
# prior is P(A) and posterious is P(A|B)
```