Bayesian network for parameter estimation
By Neelu Verma
(MP19AI002)

February 2, 2022

[75]:
```
# importing the required libraries

import matplotlib.pyplot as plt
import numpy as np
import math
```

[76]:
```
# taking horizontal values
horizontal=17
Total=27
```

[77]:
```
# defining range of theta
theta_range = np.linspace(0,1,27)
theta_range[16]
```

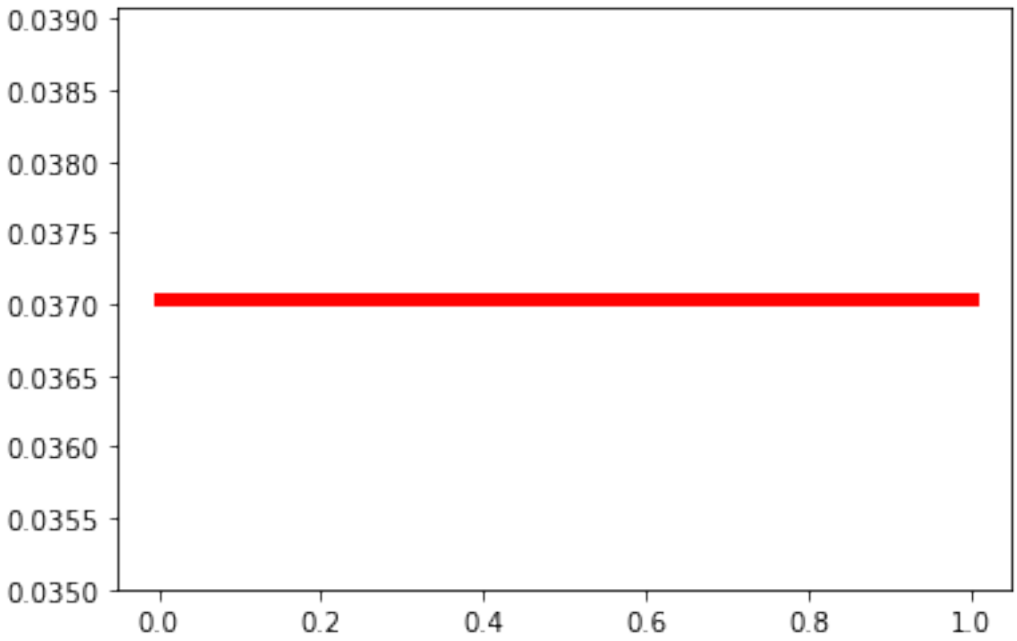
[77]:
```
0.6153846153846154
```

[78]:
```
# calculating prior
#prior = stats.beta.pdf(x = range, a=horizontal, b=Total)
prior=np.full(27,1/27)
prior
```

[78]:
```
array([0.03703704, 0.03703704, 0.03703704, 0.03703704, 0.03703704,
       0.03703704, 0.03703704, 0.03703704, 0.03703704, 0.03703704,
       0.03703704, 0.03703704, 0.03703704, 0.03703704, 0.03703704,
       0.03703704, 0.03703704, 0.03703704, 0.03703704, 0.03703704,
       0.03703704, 0.03703704, 0.03703704, 0.03703704, 0.03703704,
       0.03703704, 0.03703704])
```

[79]:
```
#plotting the prior

plt.plot(theta_range,prior,'r', linewidth=5,)
```

[79]:
```
[<matplotlib.lines.Line2D at 0x2726f3e98b0>]
```

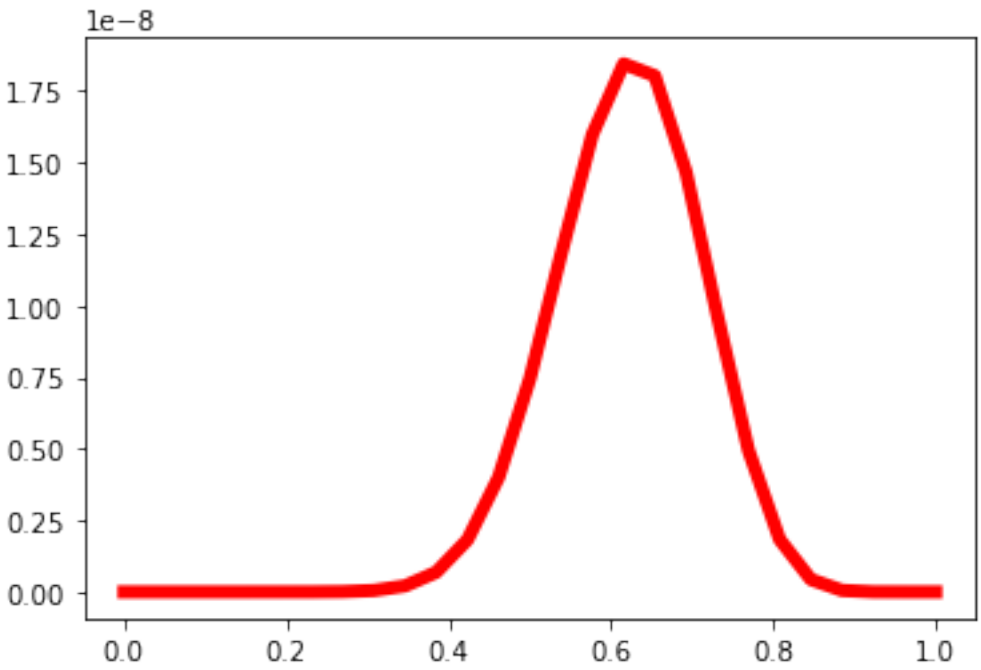


[80]:
```
#P_data_theta= stats.binom.pmf(k = horizontal, n = Total, p = theta_range)

P_data_theta=(theta_range**horizontal) *  ((1-theta_range)**(Total-horizontal))
P_data_theta
```

[80]:
```
array([0.00000000e+00, 5.95826331e-25, 5.19208685e-20, 3.34240672e-17,
       2.85079682e-15, 7.95064982e-14, 1.08290614e-12, 8.91091487e-12,
       5.02313019e-11, 2.10054253e-10, 6.86940990e-10, 1.82099118e-09,
       4.00941623e-09, 7.45058060e-09, 1.17952913e-08, 1.59660638e-08,
       1.84399318e-08, 1.80208953e-08, 1.46639868e-08, 9.67188716e-09,
       4.95155987e-09, 1.83293751e-09, 4.34015352e-10, 5.20361748e-11,
       1.86041859e-12, 3.63663532e-15, 0.00000000e+00])
```

[81]:
```
# plotting P_data_theta

plt.plot(theta_range,P_data_theta, 'r', linewidth=5)
```

[81]:
```
[<matplotlib.lines.Line2D at 0x2726db8cb80>]
```



[82]:
```
# Calculating p_data

p_data=(math.factorial(horizontal) * math.factorial(Total-(horizontal)))/ (math.
 ↪factorial(Total+1))
p_data
```

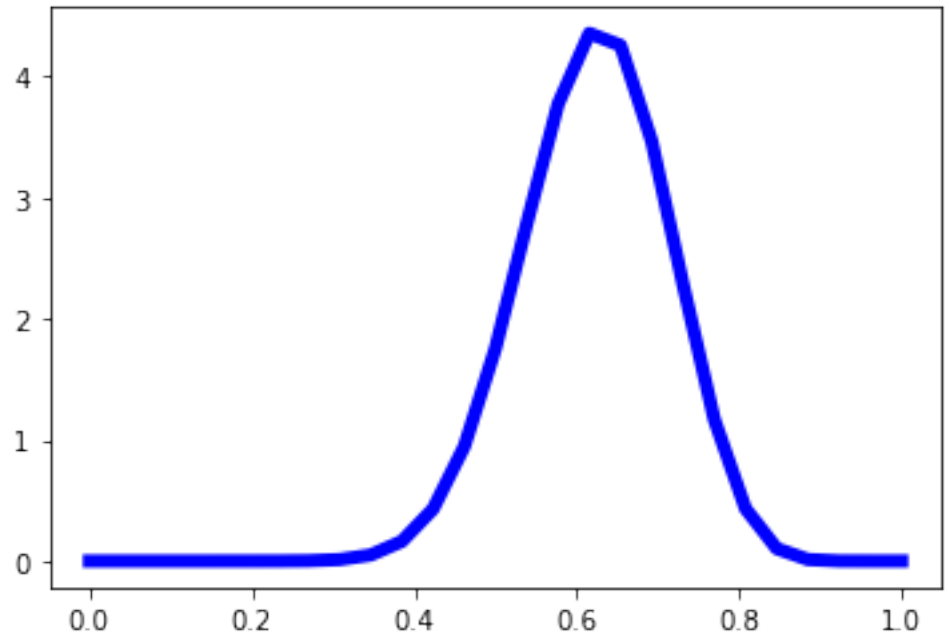
[82]:
```
4.233413844397826e-09
```

[83]:
```
# calculating and printing posterior

posterior = (P_data_theta * 1)/ p_data
posterior
```

[83]:
```
array([0.00000000e+00, 1.40743701e-16, 1.22645388e-11, 7.89529878e-09,
       6.73403764e-07, 1.87807054e-05, 2.55799736e-04, 2.10490049e-03,
       1.18654362e-02, 4.96181712e-02, 1.62266439e-01, 4.30147217e-01,
       9.47088185e-01, 1.75994620e+00, 2.78623630e+00, 3.77143941e+00,
       4.35580656e+00, 4.25682344e+00, 3.46386802e+00, 2.28465430e+00,
       1.16963757e+00, 4.32969130e-01, 1.02521362e-01, 1.22917760e-02,
       4.39460600e-04, 8.59031376e-07, 0.00000000e+00])
```

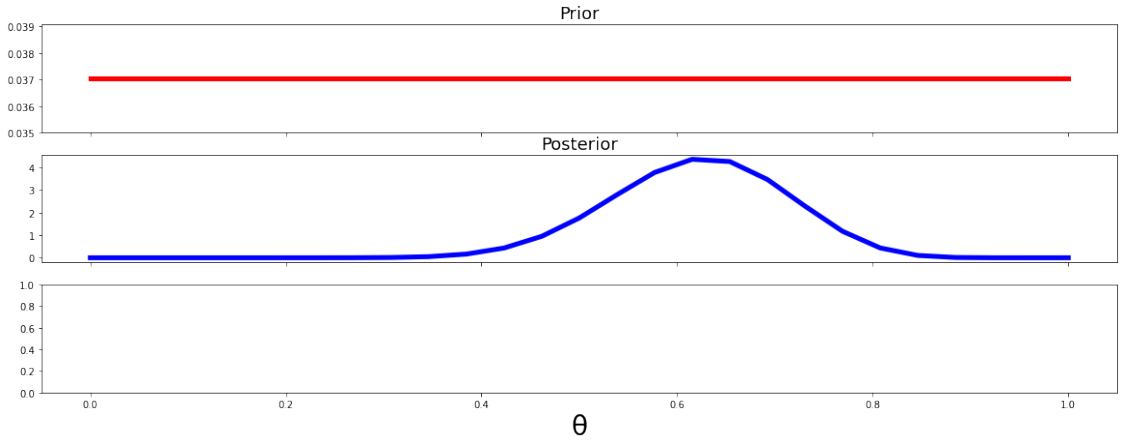
[84]:
```
# plotting theta range of posterior

plt.plot(theta_range, posterior, label='Posterior', linewidth=5, color='b')
```

[84]: [<matplotlib.lines.Line2D at 0x2726dba0a00>]



[85]:
```
fig, axes = plt.subplots(3, 1, sharex=True, figsize=(20,7))
plt.xlabel('', fontsize=28)

axes[0].plot(theta_range, prior, label="Prior", linewidth=5, color='r')
axes[0].set_title("Prior", fontsize=18)

axes[1].plot(theta_range, posterior, label='Posterior', linewidth=5, color='b')
axes[1].set_title("Posterior", fontsize=18)

#plt.show()
```

[85]: Text(0.5, 1.0, 'Posterior')



1 1. The posterior probability distribution for

[86]:
```
#updation in theta
theta=0
for i in range(27):
  theta+=theta_range[i]* posterior[i]
```

[87]:
```
print("The posterior probability distribution for :", theta)
```

```
The posterior probability distribution for : 16.13793101600681
```

2 2. Estimated value of

[88]:
```
print("Estimated value of :",theta.mean())
```

```
Estimated value of : 16.13793101600681
```

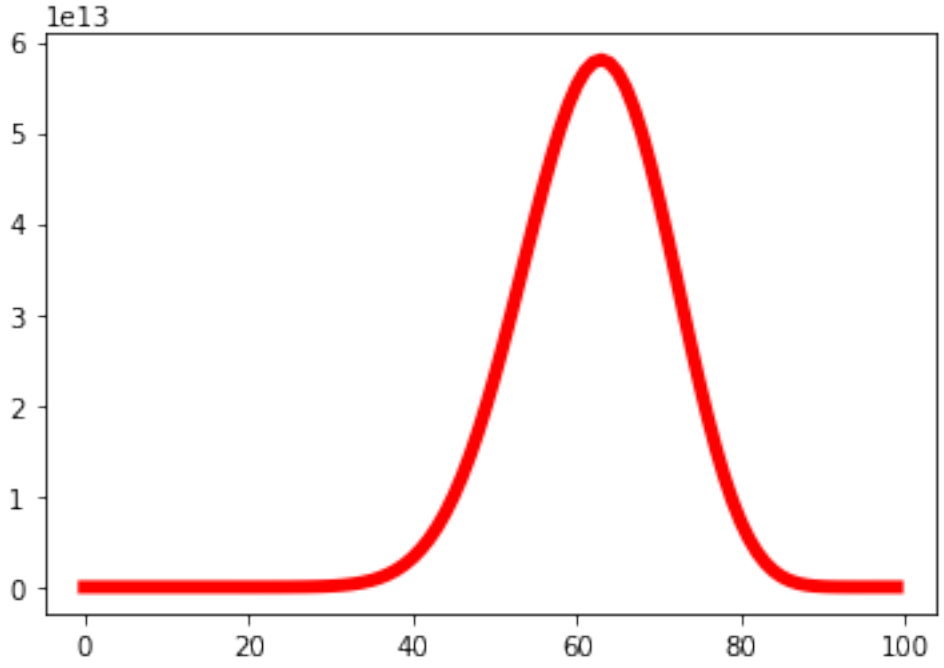
[89]:
```
###################-------------Finish-----------##############
```

3 Analytical Approach.

[52]:
```
import matplotlib.pyplot as plt
import numpy as np
import math
from math import factorial
p = []


for theta in np.arange(0, 1, 0.01):
```

```
    pdf = (factorial(28)/factorial(17)*(factorial(10)))*pow(theta, 17) *␣
 ↪pow(1-theta, 10)
    p.append(pdf)
plt.plot(p,'r',linewidth=5)

estimated_theta =(17+1)/(27+2)
mle_theta =17/27

print("Estiated theta = ",estimated_theta)
print("MLE_theta = ", mle_theta)
```

```
Estiated theta =  0.6206896551724138
MLE_theta =  0.6296296296296297
```



[ ]:
```
#########---------------Thankyou----------------------###############
```