BAIN'24

Bayesian Analysis and Inference

STAMATICS
IITK

Project Mentor:

Lavesh Gupta

Project Mentees:

Sambuddha Chakrabarti
Parv Jain
Shrey Solanki +100 more

Project Progress

01 Week One - Thinking Probabilistically

02 Week Two - Bayes Theorem

03 Week Three - Programming Probabilistically

Week One - Thinking Probabilistically

Intro and Probability Basics (But Through Code!)

Basic introduction about Bayesian modelling and data cleaning using Python .Basic terms in probablity were recapitulated along with coding part . For this< different libraries such as numpy,arviz,matplotlib,preliuz etc were used which was very fascinating.

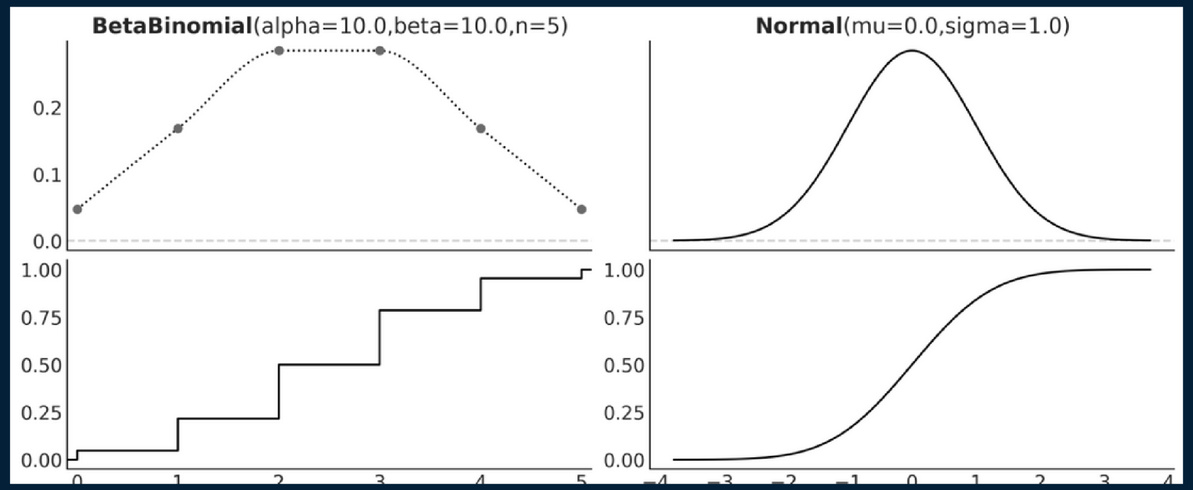
Random variable and Probability distribution

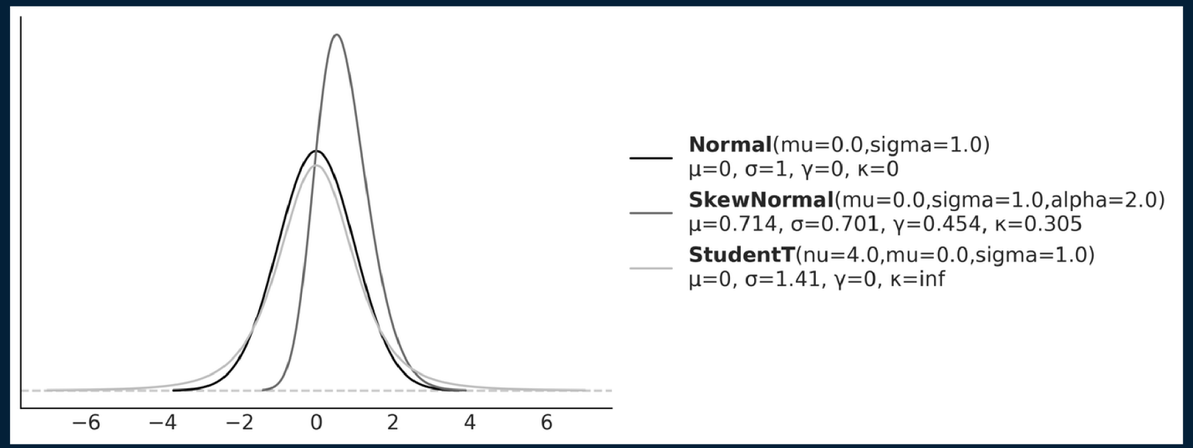
Studied about random variables and its distribution. plotted different graphs using librraries such numpy and matplotlib. Preliz is an another library which helps in plotting the gaussian plots quickly. Examples of distribution includes poisson distribution, binomial distribution etc.

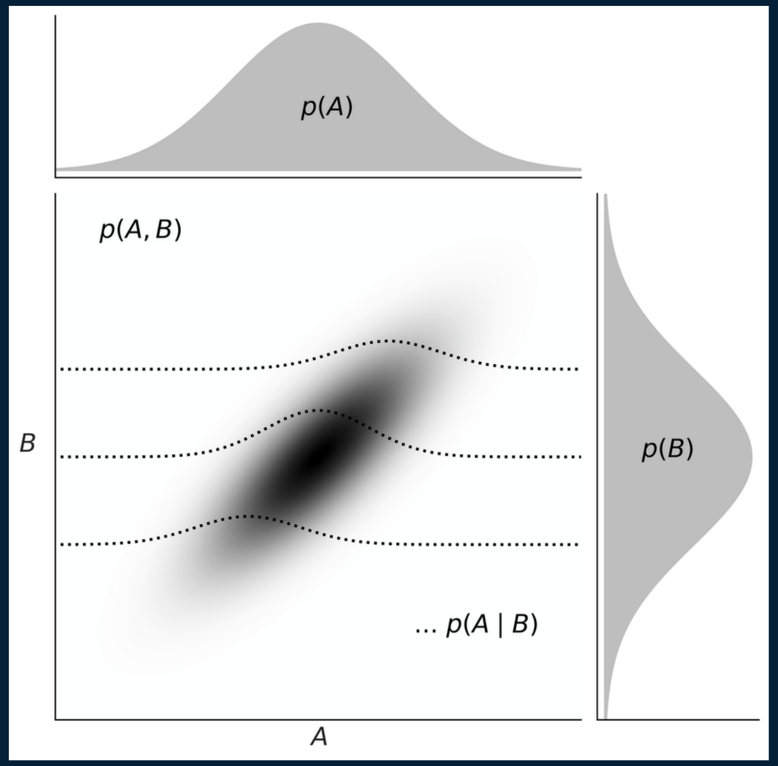
Plotting graphs of continuous random variable,expected value &CDF

With introduction of conditional probability we dived more deep into coding part of probability and plotted different graphs of continuous random variable, cumulative distribution function and expected value.

Week One - Thinking Probabilistically







Assignment 1

+ Code + Text | Copy to Drive | Connect | Gemini

```
[ ] !pip install pymc==5.8.0 arviz==0.16.1 bambi==0.13.0 pymc-bart==0.5.2 kulprit==0.0.1 preliz==0.3.6 nutpie==0.9.1
```

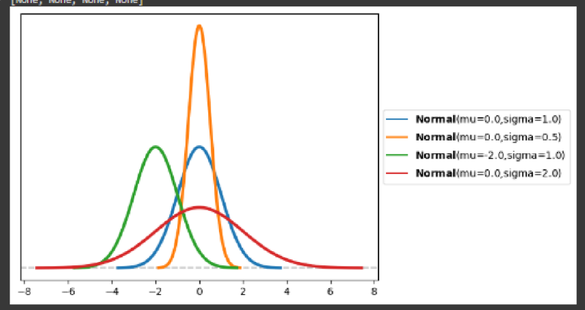
Show hidden output

```
[ ] import arviz as az  # For Bayesian data analysis and visualization
    import matplotlib.pyplot as plt  # For plotting
    import numpy as np  # For numerical operations
    from scipy.special import binom, beta  # For binomial and beta distributions
    import preliz as pz # defining priors
    from cycler import cycler
    import math
```

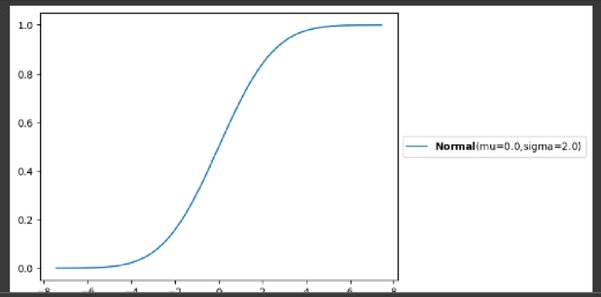
```
[ ] mus = [0., 0., -2.,0]
    sigmas = [1, 0.5, 1,2]
    for mu, sigma in zip(mus, sigmas):
        ax = pz.Normal(mu, sigma).plot_pdf()

    [line.set_linewidth(3.) for line in ax.get_lines()[1::2]]
```

```
[None, None, None, None]
```




```
[ ] mu=0
    sigma=2
    ax=pz.Normal(mu, sigma).plot_cdf()
```



102°F Clear | Search | 8:48 PM 6/16/2024

colab.research.google.com/drive/12PH_cz79XgBgJq9gTYun7Fq9tt0pNd6r#scrollTo=RqinTikSWivK

Copy of Week 1 – Assignment.ipynb

File Edit View Insert Runtime Tools Help Changes will not be saved

+ Code + Text Copy to Drive

Connect + Gemini

Q1. Suppose you have a jar with 4 jelly beans: 2 are strawberry-flavored, 1 is blueberry-flavored, and 1 is cinnamon-flavored. You draw one jelly bean at random from the jar.

```
a. What is the sample space for this experiment?

b. We define event A as the jelly bean drawn is strawberry-flavored and event B as The jelly bean drawn is not cinnamon-flavored. What are the pro

c. Are events A and B mutually exclusive? Why or why not?
```

```
S={'s','b','c'}
A={'s'}
B={'s','b'}
prob={
    's':2/4,
    'b':1/4,
    'c':1/4
}
#01234568789/*-+.
# b. A 1/2, B 3/4
# c. No, they are not mutually exclusive.
```

Q2. Previously, we defined a **Python function P** to compute the probability of an event using the naive definition of probability. Generalize that function to compute the probability of events when they are not all equally likely. Use this new function to compute the probability of events A and B from the previous exercise.

Hint: you can pass a third argument with the probability of each event.

```
def P(S,A,prob):
  if set(A).issubset(set(S)): # set A is an event
    p=0
    for a in A:
      p += prob[a]
    return p
  else:
    return 0
P(S,B,prob)
```

```
0.75
```

Q3. Use PreliZ to explore different parameters for the Gaussian distributions. Use the methods `plot_pdf`, `plot_cdf`, and `plot_interactive`. We discussed the probability mass/density functions and the cumulative density function. But there are other ways to represent functions like the percentile point function ppf. Using the `plot_ppf` method of PreliZ, plot the percentile point function for the BetaBinomial and Gaussian distributions. Can you explain how the ppf is related to the cdf and pmf/pdf?

```
!pip install pymc==5.8.0 arviz==0.16.1 bambi==0.13.0 pymc-bart==0.5.2 kulprit==0.0.1 preliz==0.3.6 nutpie==0.9.1
```

Show hidden output

```
import arviz as az  # For Bayesian data analysis and visualization
import matplotlib.pyplot as plt  # For plotting
import numpy as np  # For numerical operations
from scipy.special import binom, beta  # For binomial and beta distributions
import preliz as pz # defining priors
from cycler import cycler
import math
```

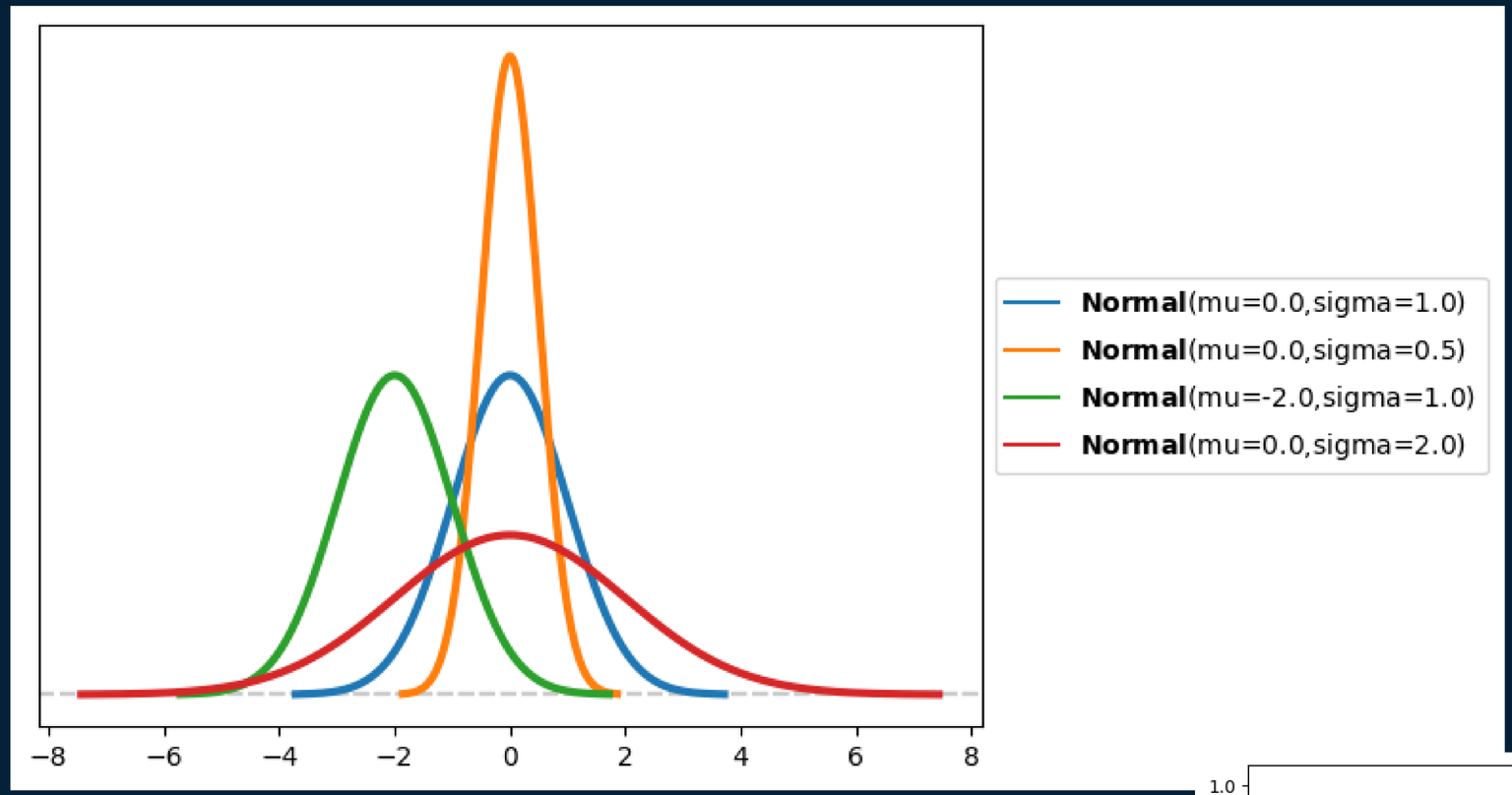
```
mus = [0., 0., -2.,0]
sigmas = [1, 0.5, 1,2]
```

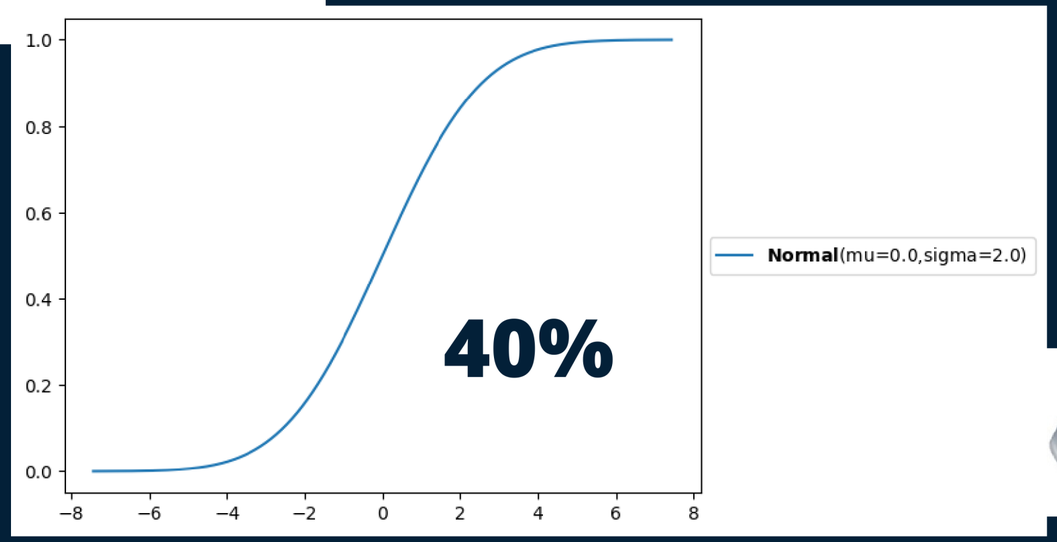
40%

102°F Clear

Search

8:54 PM 6/16/2024


Normal(mu=0.0,sigma=1.0)
Normal(mu=0.0,sigma=0.5)
Normal(mu=-2.0,sigma=1.0)
Normal(mu=0.0,sigma=2.0)
−8
−6
−4
−2
0
2
4
6
8


1.0
0.8
0.6
0.4
0.2
0.0
−8
−6
−4
−2
0
2
4
6
8
Normal(mu=0.0,sigma=2.0)
40%

Week Two - Bayes Theorem

Introduction to Prior, Posterior and Likelihood

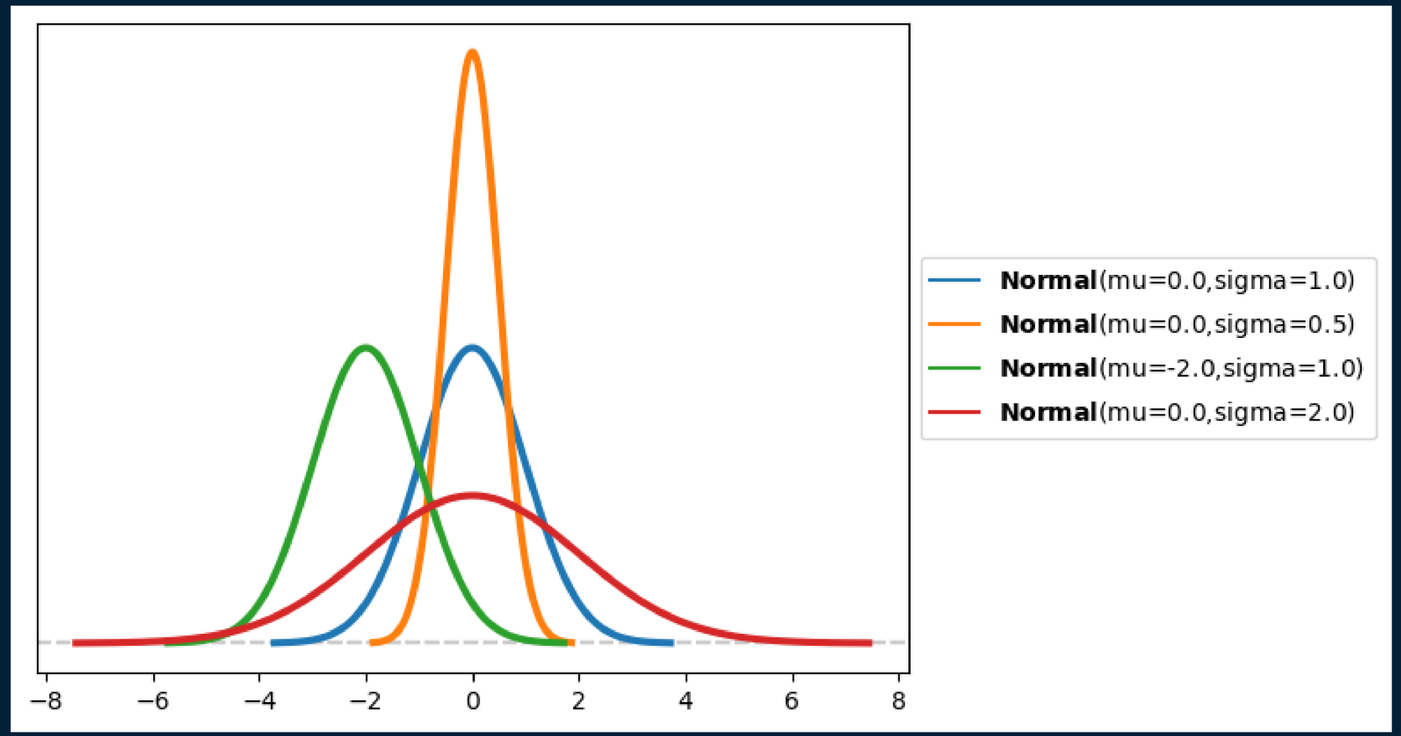
Introduced to the concept of prior, the interpretation of what we know about the model, as well as the role of likelihood in determining the posterior distribution

Single Parameter Inference

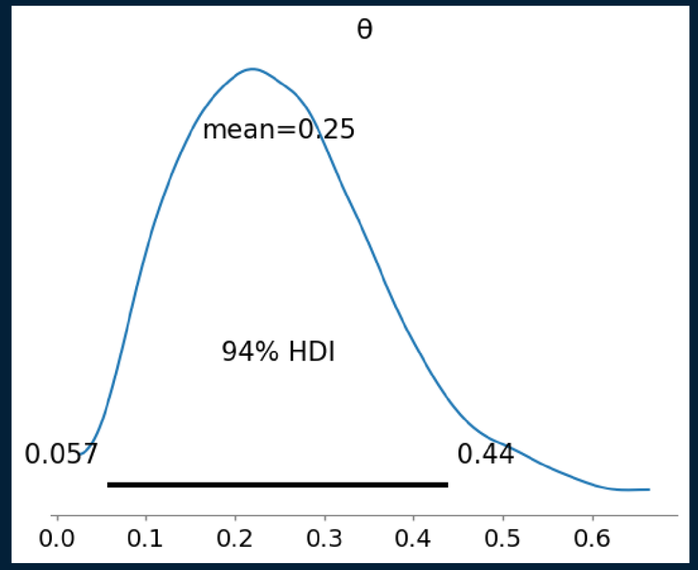
Observed Bayesian estimation of a single parameter, i.e. the coin-flipping task, by modeling the prior using the Beta distribution, the conjugate of Binomail distribution, which is used for the Likelihood. Compared distributions to get the posterior distribution and estimate the expected outcomes

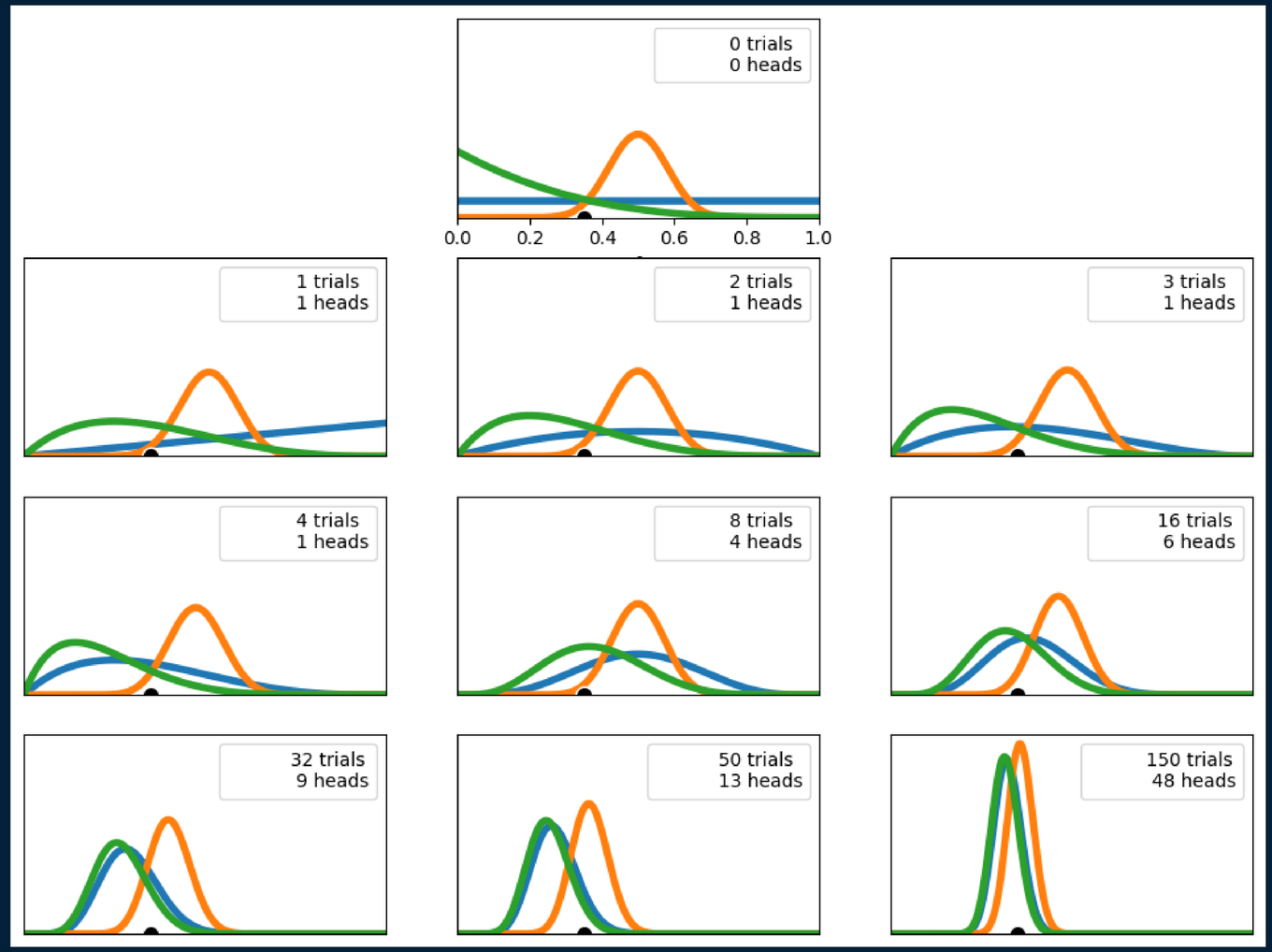
Visualizing distributions

Used the PyMC and ArViz libraries of python to visualize distributions of the prior and the posterior, and study them to choose a suitable prior and the convergence of posterior to the best value, using HDI


−8
−6
−4
−2
0
2
4
6
8
Normal(mu=0.0,sigma=1.0)
Normal(mu=0.0,sigma=0.5)
Normal(mu=-2.0,sigma=1.0)
Normal(mu=0.0,sigma=2.0)

Week Two - Bayes Theorem





Week Three - Programming Probabilistically

Introduction to PyMC library

Learned several methods of making Bayesian inferences using Python library **PyMC.** The library helps in separating the modelling and inference with a lot of inbuilt functions

Sampling and Inference

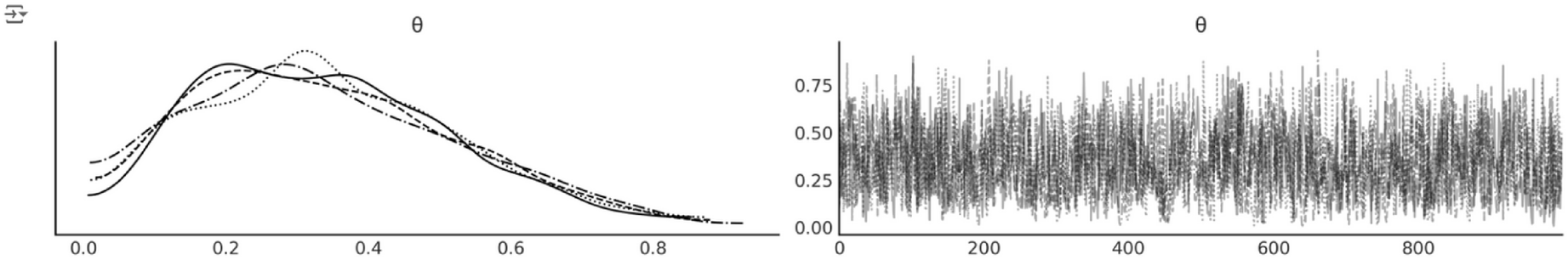
Learned sampling of real data by given parameters.
Experimented with different ways of inference like rank bars, parameter tables, HDI etc. Learned about **Savage-Dickey density ratio** and **Reigion of Practical Equivalence** which are ways to make conclusions from received posterior.

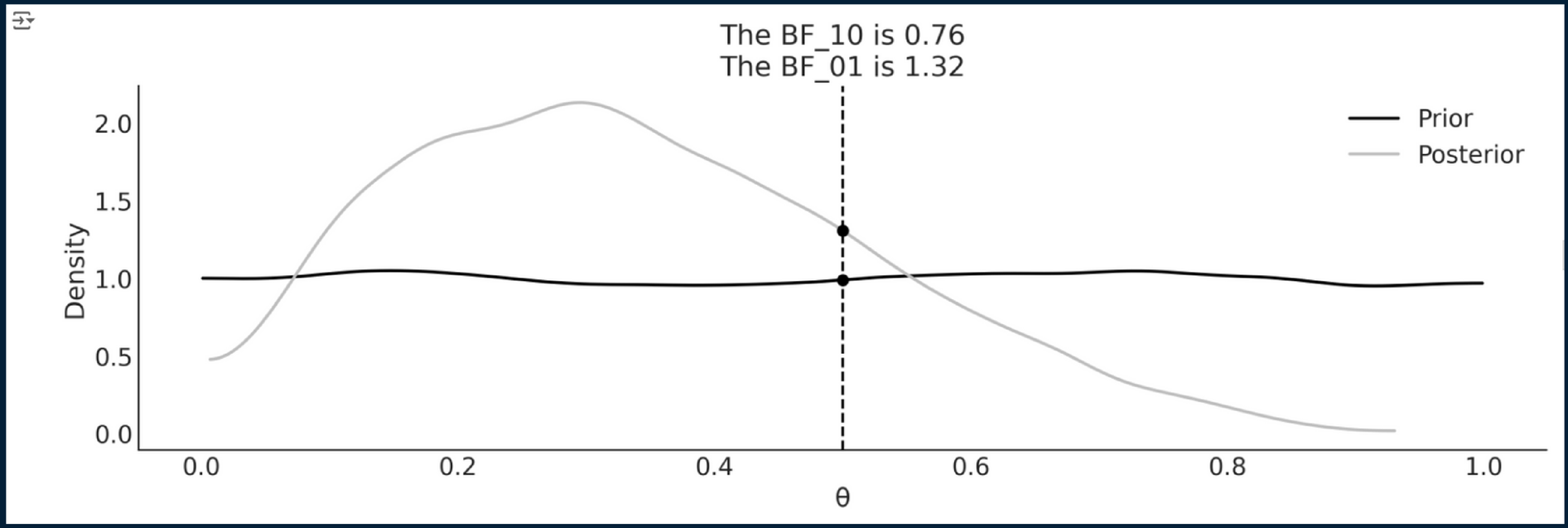
Loss functions

Used the PyMC and ArViz libraries of python to visualize distributions of the prior and the posterior, and study them to choose a suitable prior and the convergence of posterior to the best value, using HDI

Week Three - Programming Probabilistically

```
az.plot_trace(idata)
# plt.savefig("../fig/idata_trace.png")
```





Assignment 1 Submissions:

87 responses

Assignment 2 Submissions:

71 responses

Thank You