Statistics Assignment 3

Name: **Danita Anubhuti Prakash**

studentID**: s4745511**

Question 1:

Question 1 : Conjugate Uniform Random Variable Analysis.

Given:

Uniform Random Variables Y_i where $i = 1, \dots n$ and p.d.f for each as:

$$P(y|\theta) = 1/\theta$$

Patero Distribution:

$$p(x|\alpha, x_m) = \begin{cases} \alpha x_m^{\alpha} & x \geq x_m \\ 0 & x < x_m \end{cases}$$

To prove: The posterior distribution of θ.

Proof:

From the Bayes Theorem, we know that

$$P(\theta|y) = P(y|\theta) * P(\theta) / P(y)$$

where $P(y)$ is the marginal likelihood of data. and it can be given as

$$P(y) = \int_0^{x_m} P(y|\theta) * P(\theta) d\theta$$

we can simplify this by substituting the patero distribution, i.e

$$P(y) = \int_0^{x_m} P(y|\theta)^* p(\theta|\alpha x_m) d\theta$$

$$= \int_0^{x_m} \frac{1}{\theta} * (\alpha x_m^{\alpha} / \theta^{\alpha+1}) d\theta$$

$$= \alpha x_m^{\alpha} \int_0^{x_m} \theta^{-\alpha-2} d\theta$$

$$= \alpha x_m^{\alpha} / [(\alpha+1) \cdot \theta^{-\alpha-1}]$$

Now, we can substitute the obtained expression back to bayes theorem to obtain posterior distribution.

$$P(\theta | y) = P(y | \theta) * p(\theta | \alpha, x_m)$$

$$= \cancel{\frac{1}{\theta}\left(\alpha x_m^{\alpha} / [\theta(\alpha+1)]\right)}$$

$$\Rightarrow \frac{1}{\theta} \times \left(\alpha x_m^{\alpha} / \theta^{\alpha+1}\right) / \left(\alpha x_m^{\alpha} / ((\alpha+1)\theta^{-\alpha-1})\right)$$

$$\Rightarrow (\alpha+1) / \theta^{\alpha+2}$$

for $\theta > \max(y)$

Therefore the posterior distribution for θ is a Patero distribution with parameters $(\alpha+1)$ and $\max(y)$.

Question 2:

Question 2:

Given n iid categorical variables Y_i, $(i = 1, \cdots n)$ each with p.d.f $P(y \mid p_1 \cdots p_K) = \prod_{j=1}^{K} p_j$

The prior for $\theta = (p_1 \cdots p_K)$ is the dirichet distribution $\alpha_0^{(1)}, \cdots \alpha_0^{K}$ where α_0^{K} is the hyper-parameter for the j^{th} category.

To prove: ~~Prod the~~ Derive the posterior distribution of θ

Solution:

The likelihood distribution is the joint distribution of the n observations which is the product of individual probabilities.

$$L(\theta \mid y_1 \cdots y_n) = \prod_{j=1}^{K} (p_j^{n_j})$$

where n_j is the number of observations in category j.

The prior distribution is:

$$p(\theta) = \text{Dirichet}(\alpha_0^1, \cdots \alpha_0^K)$$

The posterior distribution is the proportional to the product of the likelihood function and the prior distribution. That is:

$$P(\theta \mid y_1, \cdots y_n) \propto L(\theta \mid y_1 \cdots, y_n) * p(\theta)$$

Taking log on both sides, we get

$$\log(p(\theta \mid y_1 \cdots y_n)) \propto \log(L(\theta \mid y_1 \cdots, y_n)) + \log p(\theta))$$

Using the properties of the logarithm, we can simplify this to :

$$\log(p(\theta | y_1 \cdots, y_n)) \propto \sum_{j=1}^{K} (n_j \times \log(p_j)) + \sum_{j=1}^{K} (a_0^j - 1)$$

where p_j is the probability of category j

The posterior distribution is a dirchlet distribution with updated hyperparameters

$$p(\theta | y_1, \cdots y_n) = \text{Dirichlet}(\alpha_1 \cdots \alpha_K)$$

where p_j is the probability of category j

The posterior distribution is a dirichlet distribution with updated hyperparameters

$$p(\theta | y_1 \cdots y_n) = \text{Dirichlet}(\alpha_1 \cdots \alpha_K)$$

where $\alpha_j = \alpha_0^j + n_j$

so the posterior distribution of θ is a dirichlet distribution with updated hyper-parameters that in-corporate the observed data.

Question 3 a:

Question 3

Given:

$f(x,y) = ce^{-(xy+x+y)}, x>0, y>0$

is a sampling from 2 dimensional pdf. for a normalization constant c, using gibbs sampler. Such that $(X,Y) \sim f$

To prove:

The conditional probability of X given Y=y, and the conditional pdf of Y given X=x.

Solution:

Part-1 :- To find the conditional pdf of X given Y=y we use Bayes Theorem.

$$f(x|y) = f(x,y)/f(y)$$

Part-2 :- To find the conditional pdf of Y given X=x we use Bayes Theorem

$$f(y|x) = f(x,y)/f(x)$$

First we find the joint distribution of $f(x,y)$

$$f(x,y) = ce^{-(xy+x+y)}, x>0, y>0$$

Now we can find the marginal distribution of Y, by integrating out X.

$$f(y) = \int f(x,y)dx = \int ce^{-(xy+x+y)}dx$$

$$= \int ce^{-xy-x-y} dx = ce^{-y}\int e^{-(x+1)y} dx$$

$$= ce^{-y}[-1/y+1].e^{-(x+1)y} + c$$

$$= ce^{-y}/1+y$$

Next, we find the conditional distribution of X given Y=y.

$$f(x|y) = f(x,y)/f(y)$$

$$= ce^{-xy-x-y}/(ce^{-y}/1+y)$$

$$= \cancel{(c)}\, e^{(-x(y+1))}/(1+y)$$

$$= \left[\frac{e^{(-x(y+1))}}{1+y}\right] \cancel{(1+y)}$$

Similarly, the conditional distribution of Y given X=x

$$f(y|x) = f(x,y)/f(x)$$

$$= ce^{-(xy+x+y)}/(ce^{-x}/1+x)$$

$$= \cancel{(c)}\, e^{-y(x+1)}/(e^{-x}/1+x)$$

$$= e^{-y(x+1)}/(1+x)$$

In summary, the conditional pdf of X given Y=y is $e^{-x(y+1)}/(1+y)$

and the conditional pdf of Y given X=x is

$$e^{-y(x+1)}/(1+x)$$

Question 3 b:

Consider a sampling from the 2-dimensional pdf f(x, y) = c e ^ (−(xy+x+y)), x >= 0, y >= 0, for some normalization constant c, using a Gibbs sampler. Let (X, Y) ∼ f. The working code that implements the Gibbs sampler and outputs 1000 points that are approximately distributed according to f is as follows:

```python
import numpy as np

# Define the Gibbs sampler function
def gibbs_sampler(num_samples):
    x, y = 1, 1 # Starting values of x and y
    samples = np.zeros((num_samples, 2)) # Initialize array to store samples
    for i in range(num_samples):
        # Sample x from the conditional distribution p(x|y)
        x = np.random.gamma(2, 1/(y+1))
        # Sample y from the conditional distribution p(y|x)
        y = np.random.gamma(2, 1/(x+1))
        samples[i, :] = [x, y] # Store the sample
    return samples

# Run the Gibbs sampler to generate 1000 samples
samples = gibbs_sampler(1000)

# Print the first 10 samples
print(samples[:10, :])
```

Explanation:

Here, f(x, y) is the joint pdf that we are trying to sample from. The other functions gibbs_sampler(num_samples) decide the conditional pdfs for x and y respectivly by taking the input of number of samples to be considered. After initializing the values with x and y as 1.0 iterations are run based on number of samples. Here Gibbs sampler for 1000 iterations are used. The following is the output of the some of samples.

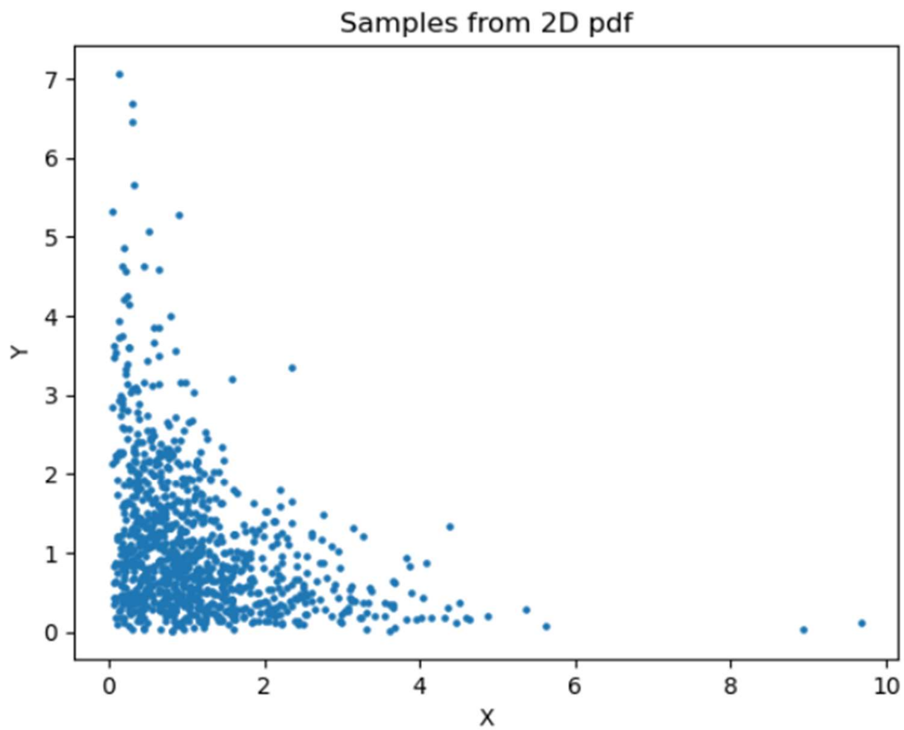
In each iteraton, a new value of x and y is sampled from the function np.random.gamma(2, 1/(y+1)) and np.random.gamma(2, 1/(x+1)). This is then stored in the samples array and returned. Here we do not use the f(x, y) function, but it is implicitly used by the gibbs distibution.

OUTPUT:

```
[[1.28047846 0.74544487]
 [0.88104641 0.70974176]
 [2.13390308 0.30993926]
 [4.50020334 0.37779603]
 [1.66773187 0.57691352]
 [2.49341698 0.56281007]
 [3.14283084 1.30936054]
 [0.46349223 0.70674931]
 [4.3887477  1.33856779]
 [0.55011417 3.12884922]]
```

```python
import matplotlib.pyplot as plt

# Plot the sampled points
plt.scatter(samples[:, 0], samples[:, 1], s=5)
plt.xlabel("X")
plt.ylabel("Y")
plt.title("Samples from 2D pdf")
plt.show()
```



GRAPH EXPLANATION:

It appears that the samples are concentrated in the lower left corner of the plot, with a central region of higher density and upper and lower regions of sparser density. This is in line with the target distribution's behaviour, having a larger probability density in the lower left corner and a decreasing probability density towards the upper and lower portions of the plot.

Question 4 a:

Model 1 assumes a binomial likelihood function and uses a sigmoid function to map the linear predictor to a probability. The model has two parameters alpha – the intercept and beta for the coefficient.

Prior: alpha ~ Uniform(0, 100), beta ~ Uniform(0, 100)

Likelihood: y ~ Binomial(n, sigmoid(alpha + beta*x))

From the data given,

r = Number of yes answers in each district, n = Total number of respondents in each district, j= number of districts

Code:

```
import pymc3 as pm
import numpy as np

# Define the data
r = np.array([10, 40, 90, 160, 150, 120, 70])  # number of YES answers in each district
n = np.array([100, 200, 300, 400, 300, 200, 100])  # total number of responses in each district
J = len(r)  # number of districts

# Define the model
with pm.Model() as model:
    # Priors for hyperparameters
    alpha = pm.Uniform('alpha', lower=0, upper=100)
    beta = pm.Uniform('beta', lower=0, upper=100)

    # Parameters of interest
    theta = pm.Beta('theta', alpha=alpha, beta=beta, shape=J)

    # Likelihood
    r_obs = pm.Binomial('r_obs', n=n, p=theta, observed=r)

    # MCMC settings
    trace_model1 = pm.sample(draws=5000, tune=1000, chains=3)

# Print the summary of the posterior distributions
pm.summary(trace_model1)
```

Explanation:

it uses uniform priors for the hyperparameters alpha and beta, and a beta distribution for the parameter theta_j for each district. The likelihood is defined as a binomial distribution with the observed number of YES answers and the total number of responses in each district.

The pm.sample function runs the MCMC algorithm with 3 chains, each with 5000 draws and a burn-in period of 1000 iterations. The pm.summary function prints the summary statistics of the posterior distributions for each parameter.

From the code we try to sample 3 chains with zero divergance.

100.00% [18000/18000 04:10<00:00 Sampling 3 chains, 0 divergences]

Question 4 b:

Based on the distance from the test centre and the student's level of preparation, this Bayesian linear regression model forecasts the number of right answers a student will provide on a test. The predictors are mapped to a mean response using a linear function under the assumption of a normal likelihood function. Beta0 (the intercept), beta1 (the coefficient for distance), and sigma (the standard deviation of the errors) are the three parameters that make up the model.

Prior:

beta0 ~ Uniform(-10, 10),

beta1 ~ Uniform(-10, 10),

sigma ~ Uniform(0, 100)

Likelihood:

num_yes ~ Binomial(num_answers, sigmoid(beta0 + beta1*distance))

From the data given,

r = Number of yes answers in each district, n = Total number of respondents in each district, j= number of districts

Code:

```
import numpy as np
import pymc3 as pm

# Define the data
n = np.array([100, 200, 300, 400, 300, 200, 100])
r = np.array([10, 40, 90, 160, 150, 120, 70])
d = np.array([7, 6, 5, 4, 3, 2, 1])

# Define the model
with pm.Model() as model:
    # Priors for the parameters
    beta0 = pm.Uniform('beta0', lower=-10, upper=10)
    beta1 = pm.Uniform('beta1', lower=-10, upper=10)
    sigma = pm.Uniform('sigma', lower=0, upper=100)

    # Prior of theta
    theta = pm.math.invlogit(beta0 + beta1 * d)

    # #The likelihood of the number of YES answers in group j, r_j
    likelihood = pm.Binomial('likelihood', n=n, p=theta, observed=r)

    # Define the theta variable and add it to the trace.
    theta_var = pm.Deterministic('theta', theta)

    # Sample from the posterior distribution
    trace_model2 = pm.sample(5000, chains=3, target_accept=0.9)

pm.summary(trace_model2)
```

Explanation:

The goal of the code is to estimate the probability of YES answers- theta as a function of the predictor variables. The prior distribution for theta is specified as the inverse logit transformation of beta0 + beta1*d. The likelihood function is specified using the Binomial distribution with the observed data (r and n) and the probability parameter (theta). A deterministic variable is created for theta (theta_var) using the pm.Deterministic functionmallows the user to calculate and store the value of a variable that is determined by other variables in the model. Finally, the model is fit using MCMC sampling algorithm, and the posterior distributions of the parameters are summarized using pm.summary. The target acceptance rate is set to 0.9 to ensure good mixing and convergence of the MCMC chains.

We use 3 sampling chains with zero divergance.

100.00% [18000/18000 03:42<00:00
Sampling 3 chains, 0 divergences]

Question 4 c:

Here, the columns can be represented as:

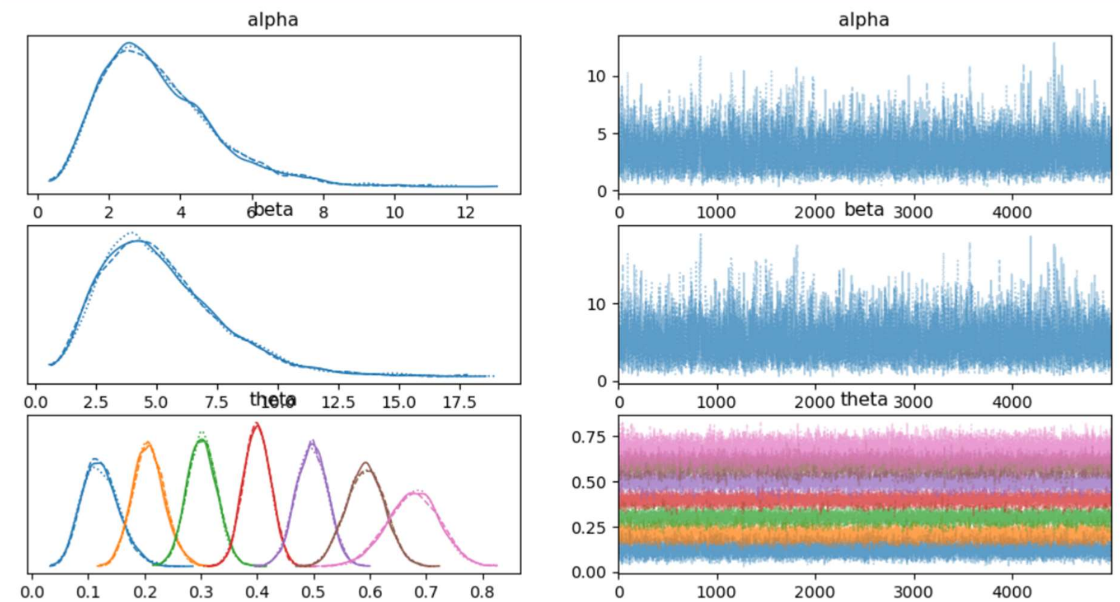
1. mean: the mean of the posterior distribution of the parameter
2. sd: the standard deviation of the posterior distribution of the parameter
3. hdi_3%: the lower bound of the 3% highest density interval (HDI) of the posterior distribution of the parameter

4. hdi_97%: the upper bound of the 97% highest density interval (HDI) of the posterior distribution of the parameter
5. mcse_mean: the estimated standard error of the mean of the parameter
6. mcse_sd: the estimated standard error of the standard deviation of the parameter
7. ess_bulk: the estimated effective sample size of the parameter, taking into account the autocorrelation between samples
8. ess_tail: the estimated effective sample size of the parameter, taking into account the tail of the posterior distribution
9. r_hat: the Gelman-Rubin statistic, which measures the convergence of the chains to the same target distribution.

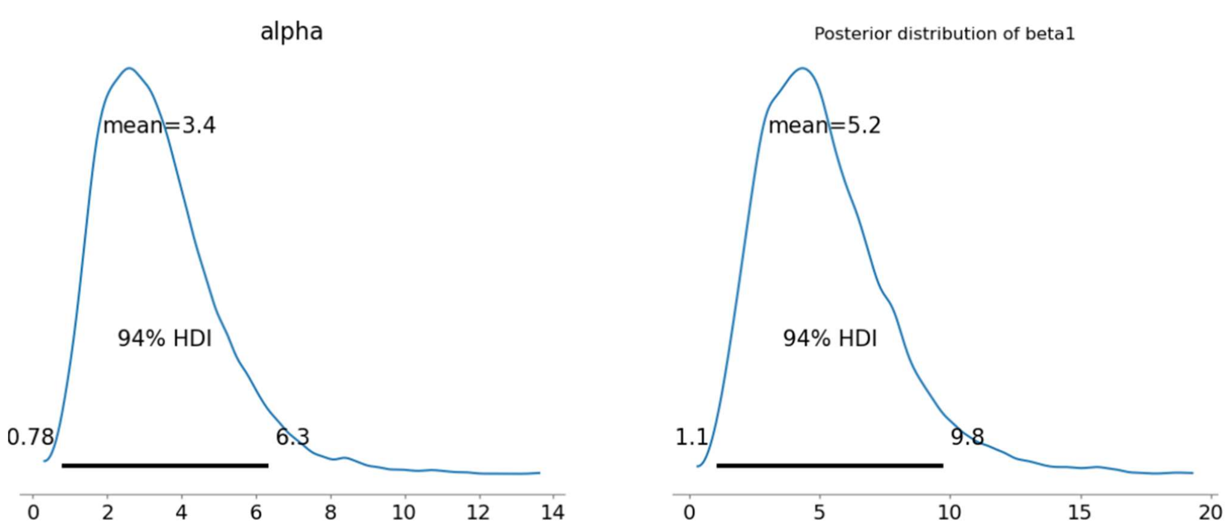
The inference obtained from Model 1 is:

	mean	sd	hdi_3%	hdi_97%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
alpha	3.381	1.630	0.803	6.411	0.022	0.018	6826.0	4374.0	1.0
beta	5.199	2.527	0.998	9.763	0.033	0.027	7068.0	4693.0	1.0
theta[0]	0.123	0.034	0.063	0.186	0.000	0.000	14540.0	9362.0	1.0
theta[1]	0.208	0.028	0.158	0.262	0.000	0.000	17340.0	9570.0	1.0
theta[2]	0.302	0.026	0.254	0.353	0.000	0.000	16944.0	9584.0	1.0
theta[3]	0.400	0.024	0.354	0.445	0.000	0.000	15632.0	10204.0	1.0
theta[4]	0.497	0.029	0.443	0.551	0.000	0.000	18632.0	8737.0	1.0
theta[5]	0.592	0.034	0.523	0.652	0.000	0.000	15803.0	9680.0	1.0
theta[6]	0.676	0.047	0.590	0.762	0.000	0.000	15446.0	8247.0	1.0

From the above model we can infer that



1. Both the sampled of aplha and beta have postive values
2. District 7 i.e thteta[6] has the highest mean (probability of yes) and highest satndard deviation
3. Both Alpha and Beta are left skewed.



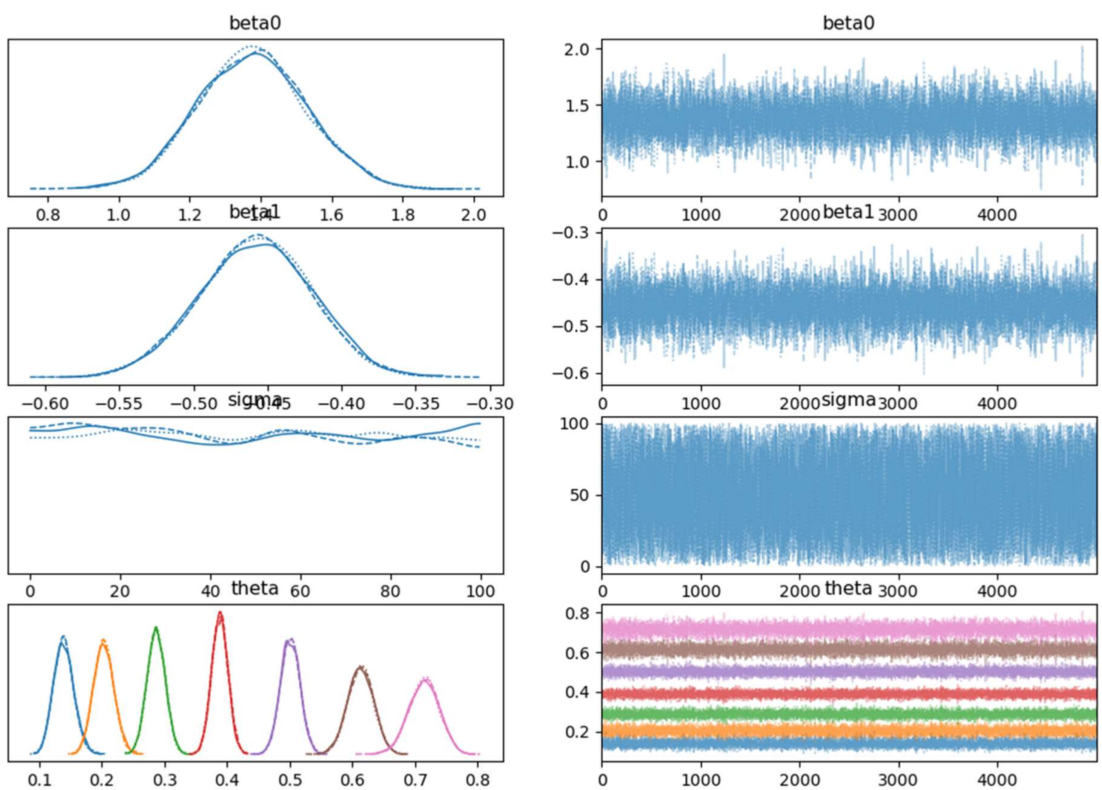
The inference for Model 2 is:

	mean	sd	hdi_3%	hdi_97%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
beta0	1.372	0.152	1.088	1.654	0.002	0.002	4781.0	5680.0	1.0
beta1	-0.457	0.037	-0.525	-0.386	0.001	0.000	4889.0	5510.0	1.0
sigma	50.032	29.180	0.003	94.040	0.322	0.232	7705.0	6196.0	1.0
theta[0]	0.139	0.016	0.110	0.169	0.000	0.000	6346.0	7639.0	1.0
theta[1]	0.203	0.016	0.173	0.232	0.000	0.000	7437.0	8740.0	1.0
theta[2]	0.287	0.014	0.260	0.314	0.000	0.000	10702.0	11099.0	1.0
theta[3]	0.388	0.013	0.364	0.412	0.000	0.000	16530.0	11147.0	1.0
theta[4]	0.500	0.015	0.472	0.530	0.000	0.000	7772.0	9715.0	1.0
theta[5]	0.612	0.020	0.574	0.651	0.000	0.000	5457.0	7237.0	1.0
theta[6]	0.713	0.024	0.667	0.757	0.000	0.000	4935.0	6219.0	1.0

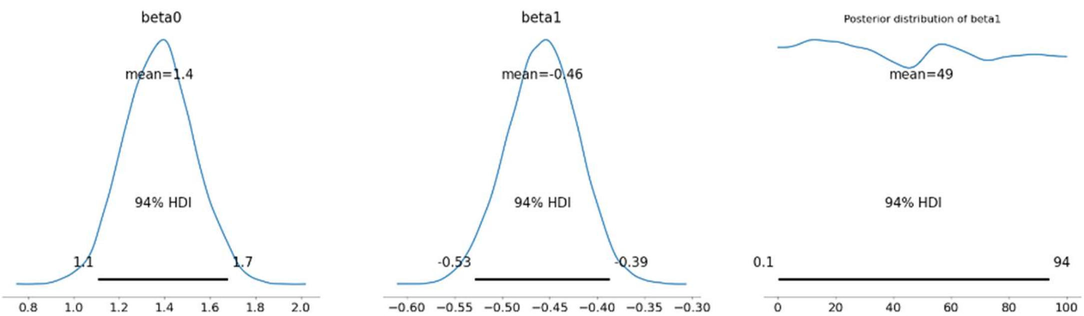
From the above model we can infer that:

1. The samples in beta0 have a positive mean.
2. The samples in beta1 have a negative mean.
3. District 7 i.e theta[6] has the highest mean and standard deviation which means they have the most number of positve (yes) responses.

The distribution can be given as:



The curve of the normal distribution is bell-shaped, which means that the values in the center of the distribution are more likely to occur than the values at the extremes. The plot of the posterior distribution of beta0 and beta1 looks like a bell-shaped curve it this is because the true values of beta0 and beta1 are close to the center of the distribution.



Question 4 d:

In Model 2, β1 represents the effect of the distance between a district and the probability of a 'yes' response. Specifically, β1 measures the change in the log-odds of a positive response for unit increase in distance.

If β1 is positive, then the log-odds of a positive response increase with distance. The probability of a positive response decreases as the distance from the clinic increases. Conversely, if β1 is negative, then the log-odds of a positive response decrease with distance, and the probability of a positive response increases as the distance from the clinic increases.

The estimated value of beta1 is -0.457 with a standard deviation of 0.037. This means that in the logistic regression model, for every one unit increase in the predictor variable d, the log-odds of a "yes" response decrease by an estimated 0.457 units on average. Alternatively, we could say that a one-unit increase in d is associated with a decrease in the odds of a "yes" response by a factor of exp(-0.457) = 0.633 on average.

Therefore, β1 captures the relationship between distance and the response variable, and its sign can indicate whether the clinic's location is a significant factor in determining the probability of a positive response.