Julia Krysiak

The goal of this excercise is to create a binomial model, use prior predictive distributions to select a proper prior and then using posterior predictive distributions.

We will be considering a following case:

Typical effects of administering certain vaccine result in an allergic reaction in 20% cases on average. This is what we know a priori

We are testing a new vaccine, for which we performed a trial on 50 patients registering 7 allergic reactions. We want to verify what is the allergic reaction probability for new trial and what is the probability that it is lower than for normal vacine. We assume that each patient treatment is exchangeable.

In [ ]:
```python
from cmdstanpy import CmdStanModel
import numpy as np
import matplotlib.pyplot as plt
import arviz as az
```

```
/usr/local/lib/python3.9/site-packages/tqdm/auto.py:22: TqdmWarning: IProgress no
t found. Please update jupyter and ipywidgets. See https://ipywidgets.readthedoc
s.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
INFO:cmdstanpy:found newer exe file, not recompiling
```

Modeling prior predictive distribution

Because prior predictive distribution does not use data it can often be realized just using random number generators.

1. Create a Stan model, which will consist of only generated quantites block.
2. In this block define integer variable N for number of trials equal to 50, integer variable y for number of successes and real p for probability of allergic reaction. Remember to use necessary constraints.
3. Assign a prior for p (appropriate random number generator) that will represent our knowledge about typical cases. You can do it either analytically or by using simulations.
4. Sample from binomial distribution, that will use N and sampled p to generate number of allergic reactions y.
5. Generate 1000 samples (pair of p an y) by calling appropriate method in cmdstanpy. Remeber to set fixed_param=True.
6. Compute ratio of allergic reactions for each sample and create a histogram.
7. Verify if mean of the ratio is consistent with prior knowledge, otherwise modify prior parameters. Describe your reasoning in the report.

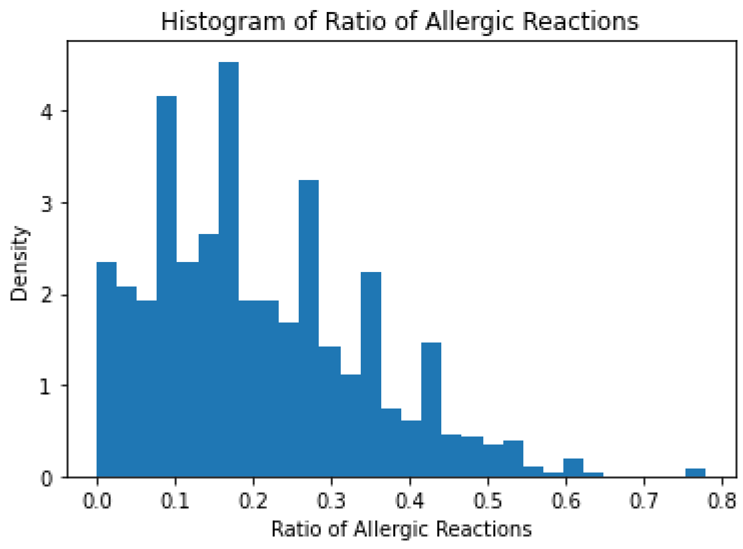
```stan
generated quantities {
    int<lower=0, upper=50> N = 50;
    int<lower=0, upper=N> y;
    real<lower=0, upper=1> p;

    p = beta_rng(2, 8);
    y = binomial_rng(N, p);
}
```

In [ ]:
```
model = CmdStanModel(stan_file='lab3_ex1.stan')
samples = model.sample(fixed_param=True, iter_sampling=1000, chains=1)
```

```
INFO:cmdstanpy:found newer exe file, not recompiling
INFO:cmdstanpy:CmdStan start processing
chain 1 |██████████| 00:00 Sampling completed

INFO:cmdstanpy:CmdStan done processing.
```

In [ ]:
```
ratios = samples.stan_variable('y') / samples.stan_variable('N')
```

In [ ]:
```
plt.hist(ratios, bins=30, density=True)
plt.xlabel('Ratio of Allergic Reactions')
plt.ylabel('Density')
plt.title('Histogram of Ratio of Allergic Reactions')
plt.show()
```



In [ ]:
```
mean_ratio = np.mean(ratios)
print("Mean ratio of allergic reactions:", mean_ratio)
```

```
Mean ratio of allergic reactions: 0.20574
```

The mean ratio is around 20% so I assume that the parameters are correct.

Modeling posterior and posterior predictive distribution

1. Create a Stan model in which:

- N and y are appropriately defined in the data block.
- p is defined in the parameters block.
- binomial likelihood and prior (selected using prior predictive distribution) are defined in the model block.

- new integer variable y_pred in the generated quantities block.
- sample y_pred using values of parameter p and input variable N

2. Generate default number of samples from appropriate method in cmdstanpy.
3. Compute ratio of predicted allergic reactions for each sample and create a histogram.
4. Compute the expected value and 94% density interval of the predicted ratio, compare it with expected value and 94% density interval of parameter p. Use arviz package.
5. Compute the probability that ratio is lower than the average probability from traditional vaccines (count the number of simulated ratios that are smaller).

```
data{
    int<lower=0> N;
    int<lower=0, upper=N> y;
}
parameters {
   real<lower=0, upper=1> p;
}
model {
   p ~ beta(2, 8);
   y ~ binomial(N, p);
}
generated quantities {
   int<lower=0, upper=N> y_pred;
   y_pred = binomial_rng(N, p);
}
```

In [ ]:
```
model2 = CmdStanModel(stan_file='lab3_ex2.stan')
```

```
INFO:cmdstanpy:compiling stan file /home/DA/lab3_ex2.stan to exe file /home/DA/la
b3_ex2
INFO:cmdstanpy:compiled model executable: /home/DA/lab3_ex2
```

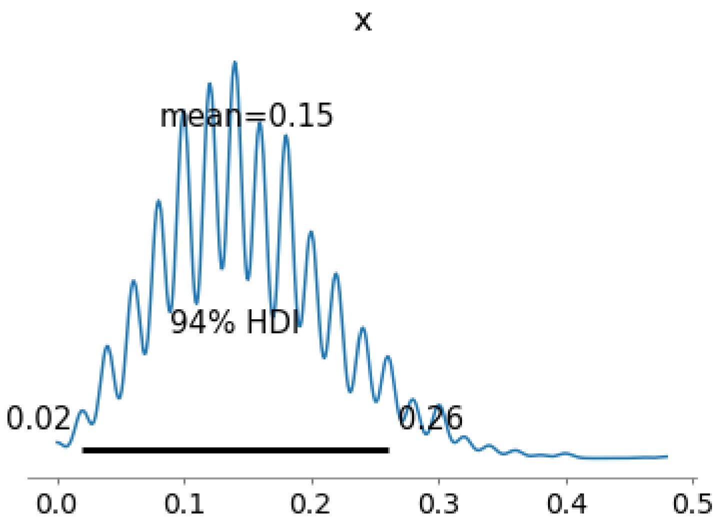
In [ ]:
```
fit = model2.sample(data={'N': 50, 'y': 7})
samples = fit.stan_variables()
```

```
INFO:cmdstanpy:CmdStan start processing
chain 1 |          | 00:00 Status


chain 1 |█          | 00:00 Status


chain 1 |██████████| 00:00 Sampling completed
chain 2 |██████████| 00:00 Sampling completed
chain 3 |██████████| 00:00 Sampling completed
chain 4 |██████████| 00:00 Sampling completed
```

```
INFO:cmdstanpy:CmdStan done processing.
```

In [ ]:
```
N_values = np.full_like(samples['y_pred'], 50)

predicted_ratio = samples['y_pred']  / N_values
```

In [ ]:
```
az.plot_posterior(predicted_ratio, hdi_prob=0.94, point_estimate='mean')
plt.show()
```



In [ ]:
```
expected_value = predicted_ratio.mean()
hdi_94 = az.hdi(predicted_ratio, hdi_prob=0.94)

print("Expected value of predicted ratio:", expected_value)
print("94% HDI of predicted ratio:", hdi_94)
```

```
Expected value of predicted ratio: 0.149935
94% HDI of predicted ratio: [0.02 0.26]
```

In [ ]:
```
p_samples = samples['p']

# Compute expected value and 94% density interval of parameter p
p_expected_value = p_samples.mean()
p_hdi_94 = az.hdi(p_samples, hdi_prob=0.94)

print("Expected value of parameter p:", p_expected_value)
print("94% HDI of parameter p:", p_hdi_94)
```

```
Expected value of parameter p: 0.14960788619999998
94% HDI of parameter p: [0.0626283 0.232148 ]
```

In [ ]:
```
average_p_traditional = 0.2  # for instance
lower_ratio_probability = (predicted_ratio < average_p_traditional).mean()

print("Probability that ratio is lower than the average probability from traditi
```

```
Probability that ratio is lower than the average probability from traditional vac
cines: 0.7465
```