BDA - Assignment 9

Anonymous

Contents:

Exercise 1

In [1]:

```python
import matplotlib.pyplot as plt
import arviz as az
import numpy as np
import pystan
import csv
# read data

file = open(r".\factory.txt", 'r')
temp = file.read().splitlines()

quality = [[] for i in range(5)]
qi = 0
for line in temp:
    values = line.split(" ")
    while True:
        try:
            values.remove("")
        except:
            break
    for i in range(len(values)):
        quality[qi].append(int(values[i]))
    qi += 1
```

Exercise 1.1

In [35]:

```python
hierarchical_model = '''
data {
    int<lower=0> N;
    int<lower=0> J;
    matrix[N, J] y;
}

parameters {
    real mu;
    real<lower=0> sigma;
    real theta[J];
    real<lower=0> sigmaJ;

    real new_machine;
}


model {
    // priors
    mu ~ normal(0, 10);
    sigma ~ inv_chi_square(1);
    for (j in 1:J)
        theta[j] ~ normal(mu, sigma);
    new_machine ~ normal(mu, sigma);
    sigmaJ ~ inv_chi_square(1);

    // likelihood
    for (j in 1:J){
        y[,j] ~ normal(theta[j], sigmaJ);
        y[,j] ~ normal(new_machine, sigmaJ);
    }
}

generated quantities {
    real ypred[J];
    real ypednew;

    for (j in 1:J) {
        ypred[j] = normal_rng(theta[j] , sigmaJ);
    }

    ypednew = normal_rng(new_machine , sigmaJ);
}
'''
```

In [36]:

```python
data = {
    "N": len(quality),
    "J": len(quality[0]),
    "y": quality
}

posterior_hierarchical = pystan.StanModel(model_code=hierarchical_model)
hierarchical_fit = posterior_hierarchical.sampling(data=data)
```

```
INFO:pystan:COMPILING THE C++ CODE FOR MODEL anon_model_1754689cd079db4b0facadba10e7e4
af NOW.
```

In [50]:

```python
def utility(y_pred):
    good = []
    bad = []
    for prediction in y_pred:
        if prediction >= 85:
            good.append(prediction)
        else:
            bad.append(prediction)
    failure_prob = len(bad) / len(y_pred)
    return - failure_prob * 106 + (1 - failure_prob) * 94


utility(y_pred)

y_pred_machines = {}
machines = ["ypred[1]", "ypred[2]", "ypred[3]", "ypred[4]", "ypred[5]", "ypred[6]", "y
for machine in machines:
    y_pred_machines[machine] = hierarchical_fit.extract(machine)[machine]

utility_machines = {}
index = 1
for key, value in y_pred_machines.items():
    utility_machines[key] = round(utility(value),2)
    print(f"The expected utility computed for machine {index} is {utility_machines[key
    index += 1
```

```
The expected utility computed for machine 1 is -46.05
The expected utility computed for machine 2 is 66.3
The expected utility computed for machine 3 is 4.7
The expected utility computed for machine 4 is 78.25
The expected utility computed for machine 5 is 11.55
The expected utility computed for machine 6 is -2.55
The expected utility computed for machine 7 is 31.3
```

Exercise 1.2

In [51]:

```python
ranked_machines = dict(sorted(utility_machines.items(), key=lambda item: item[1]))
print(f"The ranking with the associated expected utility values looks like:")
ranked_machines
```

```
The ranking with the associated expected utility values looks like:
```

Out[51]:

```
{'ypred[1]': -46.05,
 'ypred[6]': -2.55,
 'ypred[3]': 4.7,
 'ypred[5]': 11.55,
 'ypednew': 31.3,
 'ypred[2]': 66.3,
 'ypred[4]': 78.25}
```

We can therefore see that machine number 4 is the one that would return the highest expected utility.

An expected utility above 0 for a new product in a machine (given that we're dealing with profit) means that that specific machine is bringing a profit to the company. Therefore, in my case we can see that machines 1 and 6 would not bring any profit, whereas machines 2 3 4 and 5 would be good to be bought.

Exercise 1.3

This was already done in the code above, and we can find the utility below:

In [52]:

```python
print(f"The expected utility for the 7th machine is {utility_machines['ypednew']}")
```

```
The expected utility for the 7th machine is 31.3
```

Exercise 1.4

We can see that the expected utility for the 7th machine is positive, and additionally it's ranked 3rd as the one bringing most profit out of the 7. We can therefore say that the owner could buy this machine, given the lack of any prior information about the machine and the heavy reliance on data.

Exercise 1.5

Done