June 18, 2023

[1]:
```
! python3 -m pip install pystan
! pip install nest_asyncio
```

```
Requirement already satisfied: pystan in /opt/conda/lib/python3.10/site-packages
(3.7.0)
Requirement already satisfied: aiohttp<4.0,>=3.6 in
/opt/conda/lib/python3.10/site-packages (from pystan) (3.8.4)
Requirement already satisfied: clikit<0.7,>=0.6 in
/opt/conda/lib/python3.10/site-packages (from pystan) (0.6.2)
Requirement already satisfied: httpstan<4.11,>=4.10 in
/opt/conda/lib/python3.10/site-packages (from pystan) (4.10.1)
Requirement already satisfied: numpy<2.0,>=1.19 in
/opt/conda/lib/python3.10/site-packages (from pystan) (1.23.5)
Requirement already satisfied: pysimdjson<6.0.0,>=5.0.2 in
/opt/conda/lib/python3.10/site-packages (from pystan) (5.0.2)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.10/site-
packages (from pystan) (67.7.2)
Requirement already satisfied: attrs>=17.3.0 in /opt/conda/lib/python3.10/site-
packages (from aiohttp<4.0,>=3.6->pystan) (22.2.0)
Requirement already satisfied: charset-normalizer<4.0,>=2.0 in
/opt/conda/lib/python3.10/site-packages (from aiohttp<4.0,>=3.6->pystan) (3.1.0)
Requirement already satisfied: multidict<7.0,>=4.5 in
/opt/conda/lib/python3.10/site-packages (from aiohttp<4.0,>=3.6->pystan) (6.0.4)
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in
/opt/conda/lib/python3.10/site-packages (from aiohttp<4.0,>=3.6->pystan) (4.0.2)
Requirement already satisfied: yarl<2.0,>=1.0 in /opt/conda/lib/python3.10/site-
packages (from aiohttp<4.0,>=3.6->pystan) (1.9.2)
Requirement already satisfied: frozenlist>=1.1.1 in
/opt/conda/lib/python3.10/site-packages (from aiohttp<4.0,>=3.6->pystan) (1.3.3)
Requirement already satisfied: aiosignal>=1.1.2 in
/opt/conda/lib/python3.10/site-packages (from aiohttp<4.0,>=3.6->pystan) (1.3.1)
Requirement already satisfied: crashtest<0.4.0,>=0.3.0 in
/opt/conda/lib/python3.10/site-packages (from clikit<0.7,>=0.6->pystan) (0.3.1)
Requirement already satisfied: pastel<0.3.0,>=0.2.0 in
/opt/conda/lib/python3.10/site-packages (from clikit<0.7,>=0.6->pystan) (0.2.1)
Requirement already satisfied: pylev<2.0,>=1.3 in
/opt/conda/lib/python3.10/site-packages (from clikit<0.7,>=0.6->pystan) (1.4.0)
Requirement already satisfied: appdirs<2.0,>=1.4 in
```

```
/opt/conda/lib/python3.10/site-packages (from httpstan<4.11,>=4.10->pystan)
(1.4.4)
Requirement already satisfied: marshmallow<4.0,>=3.10 in
/opt/conda/lib/python3.10/site-packages (from httpstan<4.11,>=4.10->pystan)
(3.19.0)
Requirement already satisfied: webargs<9.0,>=8.0 in
/opt/conda/lib/python3.10/site-packages (from httpstan<4.11,>=4.10->pystan)
(8.2.0)
Requirement already satisfied: packaging>=17.0 in
/opt/conda/lib/python3.10/site-packages (from
marshmallow<4.0,>=3.10->httpstan<4.11,>=4.10->pystan) (23.1)
Requirement already satisfied: idna>=2.0 in /opt/conda/lib/python3.10/site-
packages (from yarl<2.0,>=1.0->aiohttp<4.0,>=3.6->pystan) (3.4)
Requirement already satisfied: nest_asyncio in /opt/conda/lib/python3.10/site-
packages (1.5.6)
```

[2]:
```
!pip install arviz
```

```
Requirement already satisfied: arviz in /opt/conda/lib/python3.10/site-packages
(0.15.1)
Requirement already satisfied: setuptools>=60.0.0 in
/opt/conda/lib/python3.10/site-packages (from arviz) (67.7.2)
Requirement already satisfied: matplotlib>=3.2 in
/opt/conda/lib/python3.10/site-packages (from arviz) (3.6.3)
Requirement already satisfied: numpy>=1.20.0 in /opt/conda/lib/python3.10/site-
packages (from arviz) (1.23.5)
Requirement already satisfied: scipy>=1.8.0 in /opt/conda/lib/python3.10/site-
packages (from arviz) (1.9.3)
Requirement already satisfied: packaging in /opt/conda/lib/python3.10/site-
packages (from arviz) (23.1)
Requirement already satisfied: pandas>=1.3.0 in /opt/conda/lib/python3.10/site-
packages (from arviz) (1.5.3)
Requirement already satisfied: xarray>=0.21.0 in /opt/conda/lib/python3.10/site-
packages (from arviz) (2023.5.0)
Requirement already satisfied: h5netcdf>=1.0.2 in
/opt/conda/lib/python3.10/site-packages (from arviz) (1.2.0)
Requirement already satisfied: typing-extensions>=4.1.0 in
/opt/conda/lib/python3.10/site-packages (from arviz) (4.5.0)
Requirement already satisfied: xarray-einstats>=0.3 in
/opt/conda/lib/python3.10/site-packages (from arviz) (0.5.1)
Requirement already satisfied: h5py in /opt/conda/lib/python3.10/site-packages
(from h5netcdf>=1.0.2->arviz) (3.8.0)
Requirement already satisfied: contourpy>=1.0.1 in
/opt/conda/lib/python3.10/site-packages (from matplotlib>=3.2->arviz) (1.0.7)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.10/site-
packages (from matplotlib>=3.2->arviz) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
/opt/conda/lib/python3.10/site-packages (from matplotlib>=3.2->arviz) (4.39.3)
```

```
Requirement already satisfied: kiwisolver>=1.0.1 in
/opt/conda/lib/python3.10/site-packages (from matplotlib>=3.2->arviz) (1.4.4)
Requirement already satisfied: pillow>=6.2.0 in /opt/conda/lib/python3.10/site-
packages (from matplotlib>=3.2->arviz) (9.5.0)
Requirement already satisfied: pyparsing>=2.2.1 in
/opt/conda/lib/python3.10/site-packages (from matplotlib>=3.2->arviz) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in
/opt/conda/lib/python3.10/site-packages (from matplotlib>=3.2->arviz) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.10/site-
packages (from pandas>=1.3.0->arviz) (2023.3)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.10/site-
packages (from python-dateutil>=2.7->matplotlib>=3.2->arviz) (1.16.0)
```

[81]:
```
import logging
import stan
import random
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
# import stan
```

[82]:
```
import arviz as az
#https://qiita.com/roki18d/items/7039e6260ec3dd4646b2
#https://qiita.com/msx222/items/3da9ad50c46bcf44f3b7
```

[83]:
```
import seaborn as sns
```

[84]:
```
# notebook
# https://qiita.com/msx222/items/37da015a06758a05c990

import nest_asyncio
nest_asyncio.apply()
```

[85]:
```
import tqdm
```

[86]:
```
from IPython.display import clear_output
```

1

p **n** **N**

N

1.1

N
$ p(N|n) p(n | N) * p(N) $

- $p(n|N)$
  n N p
  $p(n|N)$ $binom(N, p)$
  n

  $p(n|N)$ $normal(Np, Np(1-p))$

- p(N)

  (p6)

  $p(N)$ $normal(n/p, 10000)$

  N

2 stan

- Stan
  - python string
  - string build stan ( )

※
2021 pystan 3.0 update

2.1 stan

2.1.1 stan

-
  - data
    *
  - parameters
    *
  - model
    * ...

[87]:
```
stan_file = """
data {
  int n; //
  real p; //
}
parameters { // parameter
  real N; //
}
```

```
model { //     model
  N ~ normal(n/p, 10000); //
  n ~ normal(N*p, sqrt((1 - p) * p * N) );//n

}

"""
```

[88]:
```
#   .stan
with open("sample_code.stan", "w") as r:
    r.write(stan_file)
    r.close()
```

[89]:
```
def sampling(data_num, sampling_p, stan_file, fit_sampling_num):

    stan_data = { 'n':data_num, 'p': sampling_p} #stan
    posterior = stan.build(stan_file, data=stan_data, random_seed=1)
    fit = posterior.sample(num_chains=4, num_samples= fit_sampling_num) #fit␣
 ↪   MCMC

    # pystan        arviz
    idata = az.from_pystan(posterior=fit, posterior_model=posterior)
    res = az.hdi(idata,hdi_prob=0.95).N.values #     95%
    return fit, res
```

3

fit sampling

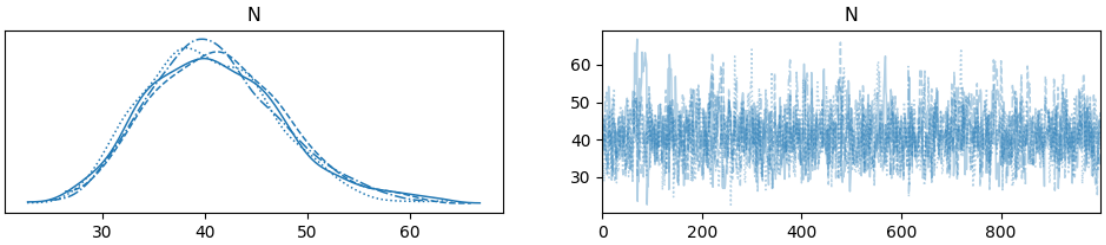
3.1

[90]:
```
#
n = 20
p = 0.5

duplicate = 1000
```

[91]:
```
fit, res = sampling(n, p, stan_file, duplicate)
clear_output()
```

[92]:
```
fig = az.plot_trace(fit)
```



[93]:
```
res #      95
```

[93]:
```
array([28.64294774, 53.95113193])
```

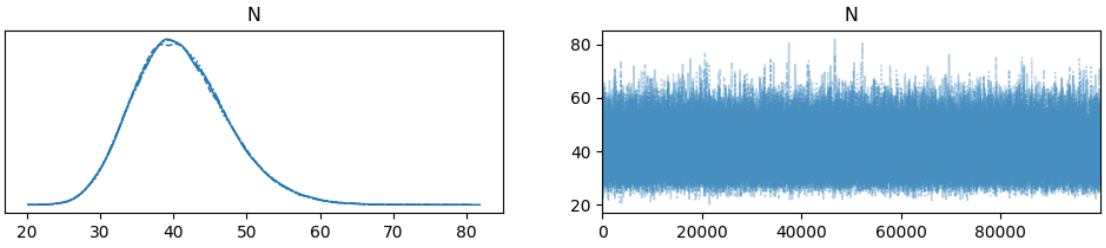
3.2

[94]:
```
#
n = 20
p = 0.5

duplicate = 100000
```

[95]:
```
fit_large, res_large = sampling(n, p, stan_file, 100000)
clear_output()
```

[96]:
```
fig = az.plot_trace(fit_large)
```



[97]:
```
res_large #      95
```

[97]:
```
array([28.84856878, 53.92251555])
```

3.3

- N $(20 \ / \ 0.5 = 40)$
-
  1000

- fig MCMC sampling step N

4 2

- N = [n * 2 ~ n * 10] - 0.5 100 - n

N p n N
N 0 n/p * 10

[98]:
```
import random
```

[99]:
```
import pandas as pd
```

[100]:
```
n = 20
p = 0.5

duplicate = 100
```

4.1

[101]:
```
def sample(N):
    '''
       N
     0.5    N


    input
        N:int

    retrun
        cnt:
    '''

    cnt = 0
    for i in range(N):
        r = random.random() #

        if r < 0.5:
            '''
              0.5
             0.5
            '''
            cnt += 1

    return cnt
```

[102]:
```
def experiment():
    '''
       N n   n / p 10
           N      duplicate
             n

    return
        df:pandas dataframe
            N:
            n_num:       n
    '''

    dic = {}
    for i in range(n, n * 2 * 10):
        cnt  = 0

        for j in range(duplicate):
            res = sample(i)

            if res == n:
                cnt +=1

        dic[i] = cnt
    df = pd.DataFrame([list(dic.keys()),list(dic.values())]).T.
↪rename(columns={0: 'N',1: 'n_num'})

    return df
```

[103]:
```
df = experiment()
total_data = df["n_num"].sum()
print(total_data)
```

```
200
```

[104]:
```
df
```

[104]:

	N	n_num
0	20	0
1	21	0
2	22	0
3	23	0
4	24	0
..	…	…
375	395	0
376	396	0
377	397	0
378	398	0
379	399	0

[380 rows x 2 columns]

4.2

[105]:
```
fit, res = sampling(n,  p, stan_file, 1000)
clear_output()
```

[106]:
```
res
```

[106]:
```
array([28.64294774, 53.95113193])
```

4.2.1

28 53 95

4.3

100
95 95

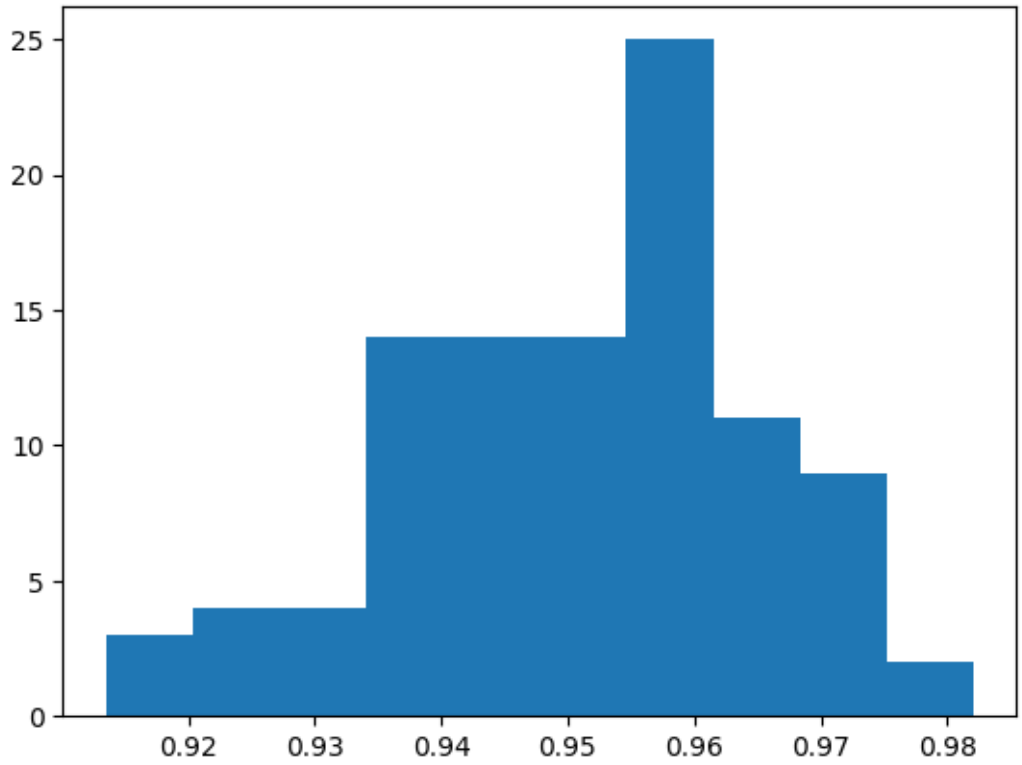
[107]:
```
li = []
for i in tqdm.tqdm(range(100)):
    df = experiment()
    num_cd_interval = df[(df["N"] > res[0]) & (df["N"] < res[1])]["n_num"].
 ↪sum() # N            n
    total_data = df["n_num"].sum() #     n
    li.append(num_cd_interval / total_data) #     n     N
```

```
100%|      | 100/100 [01:32<00:00,  1.08it/s]
```

[110]:
```
r = np.array(li)
print("mean...{}".format(r.mean()))
plt.hist(r)
```

```
mean…0.9511314888505886
```

[110]:
```
(array([ 3.,  4.,  4., 14., 14., 14., 25., 11.,  9.,  2.]),
 array([0.91351351, 0.92036575, 0.927218  , 0.93407024, 0.94092248,
        0.94777472, 0.95462696, 0.9614792 , 0.96833145, 0.97518369,
        0.98203593]),
 <BarContainer object of 10 artists>)
```



[111]:
```
r.mean()
```

[111]: 0.9511314888505886

4.4 n

n

[112]:
```
n = 40
p = 0.5

duplicate = 100
```

[113]:
```
fit, res = sampling(n,  p, stan_file, 1000)
clear_output()
```

[114]:
```
li = []
for i in tqdm.tqdm(range(100)):
    df = experiment()
    num_cd_interval = df[(df["N"] > res[0]) & (df["N"] < res[1])]["n_num"].
 ↪sum() #
    total_data = df["n_num"].sum()
```

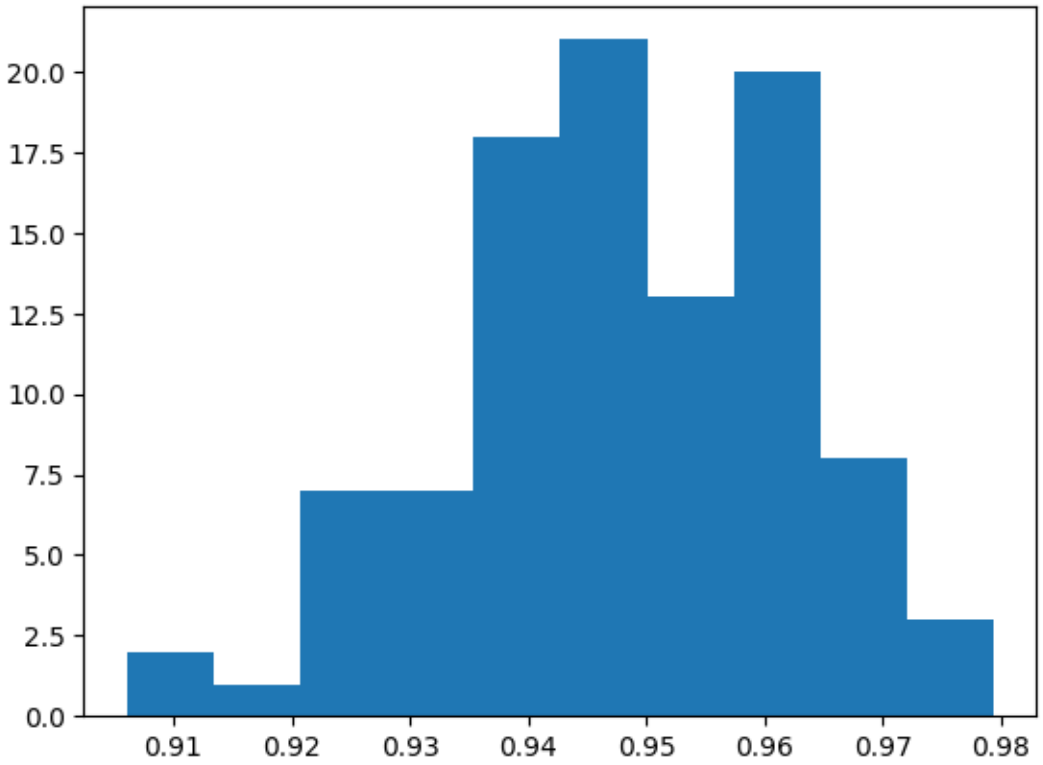
```
    li.append(num_cd_interval / total_data)
```

```
100%|      | 100/100 [06:20<00:00,  3.80s/it]
```

[115]:
```
r = np.array(li)
print("mean...{}".format(r.mean()))
plt.hist(r)
```

```
mean…0.9479293029867005
```

[115]:
```
(array([ 2.,  1.,  7.,  7., 18., 21., 13., 20.,  8.,  3.]),
 array([0.90610329, 0.9134311 , 0.92075892, 0.92808673, 0.93541455,
        0.94274236, 0.95007018, 0.957398  , 0.96472581, 0.97205363,
        0.97938144]),
 <BarContainer object of 10 artists>)
```



95

5 n

n

[116]:
```
import time
start_ts = time.time()
idx_li = []
li = []
for i in tqdm.tqdm(range(12, 1000)):
    fit, res = sampling(i, 0.5, stan_file, 1000)
    idx_li.append(i)
    li.append(res)

end_ts = time.time()
clear_output()
```

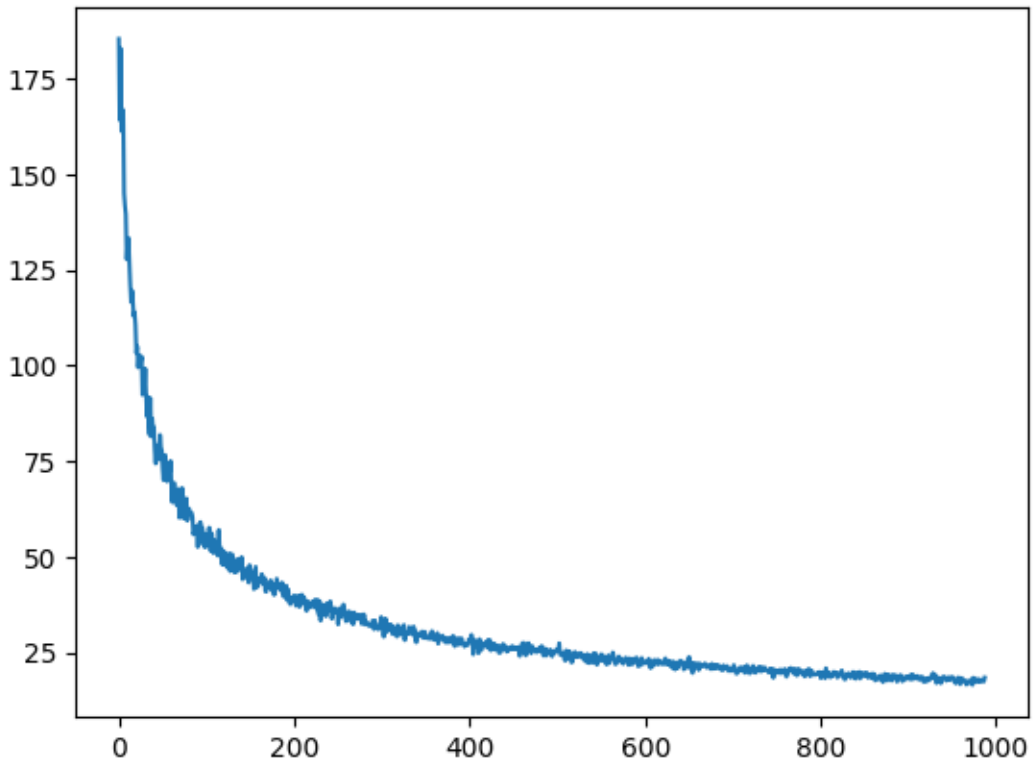
[117]:
```
print("   {} ".format(int((end_ts - start_ts)/60)))
```

```
  5
```

[118]:
```
stacked = np.stack(li)
idx_ = np.array(idx_li)
```

[119]:
```
plt.plot((stacked[:,1] - idx_*2) /idx_*2 * 100) #     (%)
```

[119]:
```
[<matplotlib.lines.Line2D at 0x7f117f694250>]
```



[ ]: