

```
In [ ]: from cmdstanpy import CmdStanModel
import pandas as pd
import arviz as az
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
```

```
/usr/local/lib/python3.9/site-packages/tqdm/auto.py:22: TqdmWarning: IPProgress not found. Please update jupyter and ipywidgets. See https://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
```

```
In [ ]: F = len("Julia")
L = len("Krysiak")
```

Excercise 1 - Generated Quantities Block

```
In [ ]: gen_quant = CmdStanModel(stan_file='code_1.stan')
```

```
INFO:cmdstanpy:found newer exe file, not recompiling
```

```
In [ ]: samples = gen_quant.sample(data={'M':F},
                                     fixed_param=True,
                                     iter_sampling=1000,
                                     iter_warmup=1,
                                     chains = 1)
```

```
INFO:cmdstanpy:CmdStan start processing
chain 1 | ██████████ | 00:00 Sampling completed
```

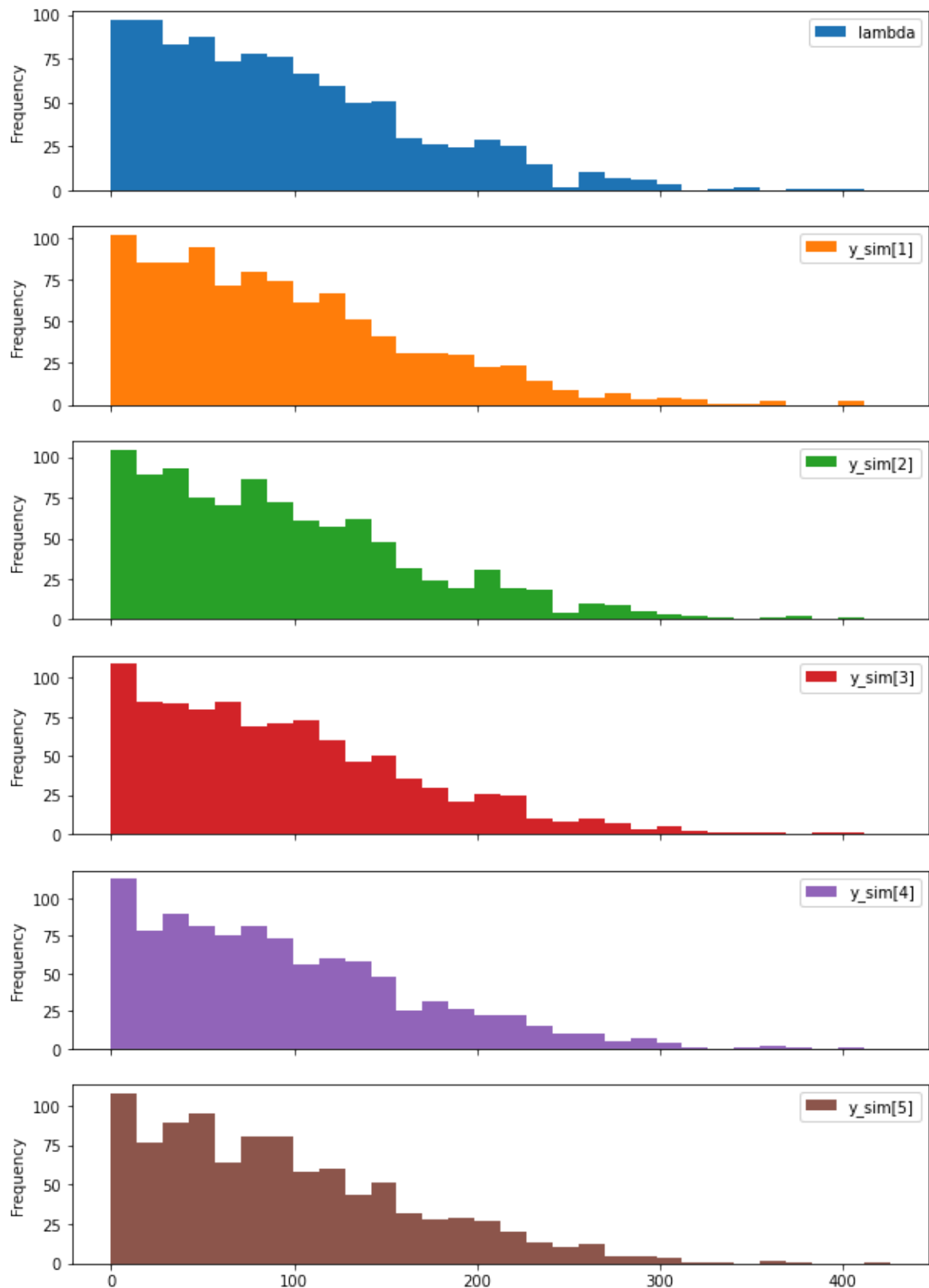
```
INFO:cmdstanpy:CmdStan done processing.
```

```
In [ ]: samples_df = samples.draws_pd()
samples_df.head()
```

```
Out[ ]:   lp__  accept_stat__  lambda  y_sim[1]  y_sim[2]  y_sim[3]  y_sim[4]  y_sim[5]
```

	lp__	accept_stat__	lambda	y_sim[1]	y_sim[2]	y_sim[3]	y_sim[4]	y_sim[5]
0	0.0	0.0	259.6150	263.0	263.0	258.0	256.0	258.0
1	0.0	0.0	160.4310	144.0	161.0	152.0	138.0	156.0
2	0.0	0.0	84.5661	79.0	80.0	111.0	82.0	99.0
3	0.0	0.0	66.6512	63.0	55.0	65.0	66.0	81.0
4	0.0	0.0	223.0360	233.0	209.0	216.0	212.0	208.0

```
In [ ]: samples_df2 = samples_df.drop(samples_df.columns[0:2], axis=1)
samples_df2.plot.hist(subplots=True, bins=30, figsize=(10,15))
plt.show()
```



Excercise 2 - Constraints on the data

```
In [ ]: bern1 = CmdStanModel(stan_file='code_2.stan')
samp_bern1 = bern1.sample(data={'N':2, 'y':[0,2]})
```

```
INFO:cmdstanpy:compiling stan file /home/DA/code_2.stan to exe file /home/DA/code_2
INFO:cmdstanpy:compiled model executable: /home/DA/code_2
INFO:cmdstanpy:CmdStan start processing
chain 1 |          | 00:00 Status

ERROR:cmdstanpy:Chain [4] error: error during processing Operation not permitted
ERROR:cmdstanpy:Chain [2] error: error during processing Operation not permitted
ERROR:cmdstanpy:Chain [1] error: error during processing Operation not permitted
ERROR:cmdstanpy:Chain [3] error: error during processing Operation not permitted
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.
```

[illegible]

5/28

6/28

[illegible]

8/28

9/28

10/28

[illegible]

12/28

[illegible]

[illegible]

15/28

16/28

```
In [ ]: bern2 = CmdStanModel(stan_file='code_3.stan')
samp_bern2 = bern2.sample(data={'N':2, 'y':[0,2]})
```

```

INFO:cmdstanpy:compiling stan file /home/DA/code_3.stan to exe file /home/DA/code_3
INFO:cmdstanpy:compiled model executable: /home/DA/code_3
INFO:cmdstanpy:CmdStan start processing
chain 1 |          | 00:00 Status

ERROR:cmdstanpy:Chain [4] error: error during processing Operation not permitted
ERROR:cmdstanpy:Chain [3] error: error during processing Operation not permitted
ERROR:cmdstanpy:Chain [2] error: error during processing Operation not permitted
ERROR:cmdstanpy:Chain [1] error: error during processing Operation not permitted
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.

```

```

-----
RuntimeError                                Traceback (most recent call last)

```

```

Input In [42], in <cell line: 2>()

```

```

      1 bern2 = CmdStanModel(stan_file='code_3.stan')
----> 2 samp_bern2 = bern2.sample(data={'N':2, 'y':[0,2]})

```

```

File /usr/local/lib/python3.9/site-packages/cmdstanpy/model.py:1089, in CmdStanModel.sample(self, data, chains, parallel_chains, threads_per_chain, seed, chain_ids, inits, iter_warmup, iter_sampling, save_warmup, thin, max_treedepth, metric, step_size, adapt_engaged, adapt_delta, adapt_init_phase, adapt_metric_window, adapt_step_size, fixed_param, output_dir, sig_figs, save_latent_dynamics, save_profile, show_progress, show_console, refresh, time_fmt, force_one_process_per_chain)
    1085         msg = 'Error during sampling:\n{}'.format(runset.get_err_msgs())
    1086         msg = '{}Command and output files:\n{}'.format(
    1087             msg, runset.__repr__()
    1088         )
-> 1089         raise RuntimeError(msg)
    1091     mcmc = CmdStanMCMC(runset)
    1092     return mcmc

```

```

RuntimeError: Error during sampling:

```

```

Exception: code_3_model_namespace::code_3_model: y[2] is 2, but must be less than
or equal to 1.000000 (in '/home/DA/code_3.stan', line 3, column 4 to column 36)
Exception: code_3_model_namespace::code_3_model: y[2] is 2, but must be less than
or equal to 1.000000 (in '/home/DA/code_3.stan', line 3, column 4 to column 36)
Exception: code_3_model_namespace::code_3_model: y[2] is 2, but must be less than
or equal to 1.000000 (in '/home/DA/code_3.stan', line 3, column 4 to column 36)
Exception: code_3_model_namespace::code_3_model: y[2] is 2, but must be less than
or equal to 1.000000 (in '/home/DA/code_3.stan', line 3, column 4 to column 36)Co
mmand and output files:

```

```

RunSet: chains=4, chain_ids=[1, 2, 3, 4], num_processes=4

```

```

cmd (chain 1):

```

```

['/home/DA/code_3', 'id=1', 'random', 'seed=30419', 'data', 'file=/tmp/tmpz7oxv1s3/j110ley1.json', 'output', 'file=/tmp/tmpz7oxv1s3/code_3-20240314113157_1.csv', 'method=sample', 'algorithm=hmc', 'adapt', 'engaged=1']

```

```

retcodes=[1, 1, 1, 1]

```

```

per-chain output files (showing chain 1 only):

```

```

csv_file:

```

```

/tmp/tmpz7oxv1s3/code_3-20240314113157_1.csv

```

```

console_msgs (if any):

```

```

/tmp/tmpz7oxv1s3/code_3-20240314113157_0-stdout.txt

```

For code_2.stan and code_3.stan errors occurred during compiling. For code_2.stan it was caused because the interval was set to $[0,2]$ and the constraints didn't allow that. For code_3.stan an error appeared because of the constraints put on $y[N]$ array. If the interval was set as $[0,1]$ it would work properly.

Excercise 3 - Constraints on parameters

Unconstrained parameters

```
In [ ]: model_gm1 = CmdStanModel(stan_file='code_4.stan')
out_gamma1 = model_gm1.sample(output_dir='samples', iter_sampling=6000, iter_warmu
out_gamma1.diagnose()
```

```
INFO:cmdstanpy:compiling stan file /home/DA/code_4.stan to exe file /home/DA/code_4
INFO:cmdstanpy:compiled model executable: /home/DA/code_4
INFO:cmdstanpy:CmdStan start processing
chain 1 |           | 00:00 Status

chain 1 | ████████ | 00:00 Iteration: 1001 / 7000 [ 14%] (Sampling)

chain 1 | ████████ | 00:00 Iteration: 2200 / 7000 [ 31%] (Sampling)

chain 1 | ████████ | 00:00 Iteration: 2900 / 7000 [ 41%] (Sampling)

chain 1 | ████████ | 00:00 Iteration: 3400 / 7000 [ 48%] (Sampling)

chain 1 | ████████ | 00:01 Iteration: 3900 / 7000 [ 55%] (Sampling)

chain 1 | ████████ | 00:01 Iteration: 4300 / 7000 [ 61%] (Sampling)

chain 1 | ████████ | 00:01 Iteration: 4700 / 7000 [ 67%] (Sampling)

chain 1 | ████████ | 00:01 Iteration: 5100 / 7000 [ 72%] (Sampling)

chain 1 | ████████ | 00:01 Iteration: 5500 / 7000 [ 78%] (Sampling)

chain 1 | ████████ | 00:01 Iteration: 5900 / 7000 [ 84%] (Sampling)

chain 1 | ████████ | 00:01 Iteration: 6300 / 7000 [ 90%] (Sampling)

chain 1 | ████████ | 00:01 Iteration: 6700 / 7000 [ 95%] (Sampling)

chain 1 | ████████ | 00:01 Sampling completed
chain 2 | ████████ | 00:01 Sampling completed
chain 3 | ████████ | 00:01 Sampling completed
chain 4 | ████████ | 00:01 Sampling completed

INFO:cmdstanpy:CmdStan done processing.
```

```
Out[ ]: "Processing csv files: /home/DA/samples/code_4-20240419175916_1.csv, /home/DA/samples/code_4-20240419175916_2.csv, /home/DA/samples/code_4-20240419175916_3.csv, /home/DA/samples/code_4-20240419175916_4.csv\n\nChecking sampler transitions treedepth.\nTreedepth satisfactory for all transitions.\n\nChecking sampler transitions for divergences.\n9489 of 24000 (39.54%) transitions ended with a divergence.\nThese divergent transitions indicate that HMC is not fully able to explore the posterior distribution.\nTry increasing adapt delta closer to 1.\nIf this doesn't remove all divergences, try to reparameterize the model.\n\nChecking E-BFMI - sampler transitions HMC potential energy.\nE-BFMI satisfactory.\n\nEffective sample size satisfactory.\n\nSplit R-hat values satisfactory all parameters.\n\nProcessing complete.\n"
```

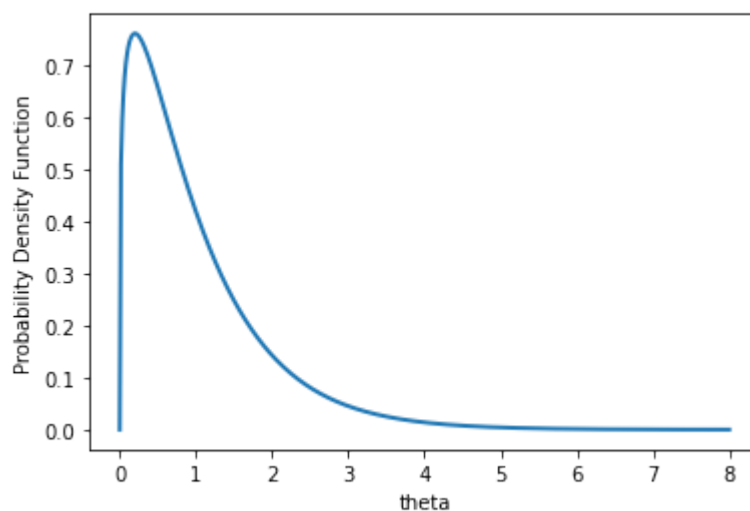
```
In [ ]: N=500
xs = np.linspace(0,8,N)
pdfs = stats.gamma.pdf(xs, 1.25, scale = 1 / 1.25)

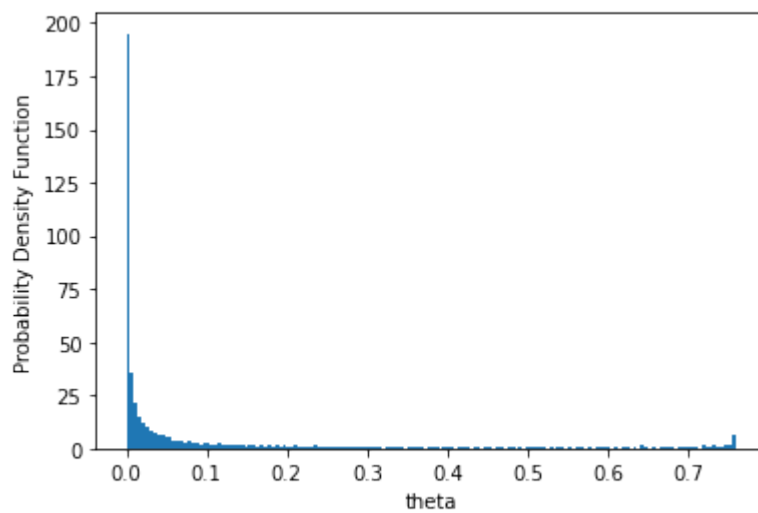
plt.plot(xs, pdfs, linewidth=2)

## add histogram of theta samples with 160 bins

plt.gca().set_xlabel("theta")
plt.gca().set_ylabel("Probability Density Function")
plt.show()

plt.hist(pdfs, bins=160)
plt.gca().set_xlabel("theta")
plt.gca().set_ylabel("Probability Density Function")
plt.show()
```





Constrained parameter

```
In [ ]: model_gm2 = CmdStanModel(stan_file='code_5.stan')
out_gamma2 = model_gm2.sample(output_dir='samples', iter_sampling=6000, iter_warmu
out_gamma2.diagnose()
```



```
INFO:cmdstanpy:compiling stan file /home/DA/code_5.stan to exe file /home/DA/code_5
INFO:cmdstanpy:compiled model executable: /home/DA/code_5
INFO:cmdstanpy:CmdStan start processing
chain 1 |           | 00:00 Status

chain 1 ||           | 00:00 Status

chain 1 |█           | 00:00 Iteration: 1001 / 7000 [ 14%] (Sampling)

chain 1 |██          | 00:00 Iteration: 1800 / 7000 [ 25%] (Sampling)

chain 1 |███         | 00:00 Iteration: 2400 / 7000 [ 34%] (Sampling)

chain 1 |████        | 00:01 Iteration: 2900 / 7000 [ 41%] (Sampling)

chain 1 |█████       | 00:01 Iteration: 3300 / 7000 [ 47%] (Sampling)

chain 1 |██████      | 00:01 Iteration: 3600 / 7000 [ 51%] (Sampling)

chain 1 |███████     | 00:01 Iteration: 3900 / 7000 [ 55%] (Sampling)

chain 1 |████████    | 00:01 Iteration: 4200 / 7000 [ 60%] (Sampling)

chain 1 |█████████   | 00:02 Iteration: 4500 / 7000 [ 64%] (Sampling)

chain 1 |██████████  | 00:02 Iteration: 4800 / 7000 [ 68%] (Sampling)

chain 1 |███████████ | 00:02 Iteration: 5100 / 7000 [ 72%] (Sampling)

chain 1 |███████████ | 00:02 Iteration: 5400 / 7000 [ 77%] (Sampling)

chain 1 |███████████ | 00:02 Iteration: 5700 / 7000 [ 81%] (Sampling)

chain 1 |███████████ | 00:02 Iteration: 6000 / 7000 [ 85%] (Sampling)

chain 1 |███████████ | 00:02 Iteration: 6300 / 7000 [ 90%] (Sampling)

chain 1 |███████████ | 00:02 Iteration: 6600 / 7000 [ 94%] (Sampling)

chain 1 |███████████ | 00:03 Iteration: 6900 / 7000 [ 98%] (Sampling)
```

```
chain 1 | ██████████ | 00:03 Sampling completed
chain 2 | ██████████ | 00:03 Sampling completed
chain 3 | ██████████ | 00:03 Sampling completed
chain 4 | ██████████ | 00:03 Sampling completed
```

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

```
Out[ ]: "Processing csv files: /home/DA/samples/code_5-20240419180030_1.csv, /home/DA/samples/code_5-20240419180030_2.csv, /home/DA/samples/code_5-20240419180030_3.csv, /home/DA/samples/code_5-20240419180030_4.csv\n\nChecking sampler transitions treedepth.\nTreedepth satisfactory for all transitions.\n\nChecking sampler transitions for divergences.\n2 of 24000 (0.01%) transitions ended with a divergence.\nThese divergent transitions indicate that HMC is not fully able to explore the posterior distribution.\nTry increasing adapt delta closer to 1.\nIf this doesn't remove all divergences, try to reparameterize the model.\n\nChecking E-BFMI - sampler transitions HMC potential energy.\nE-BFMI satisfactory.\n\nEffective sample size satisfactory.\n\nSplit R-hat values satisfactory all parameters.\n\nProcessing complete.\n"
```

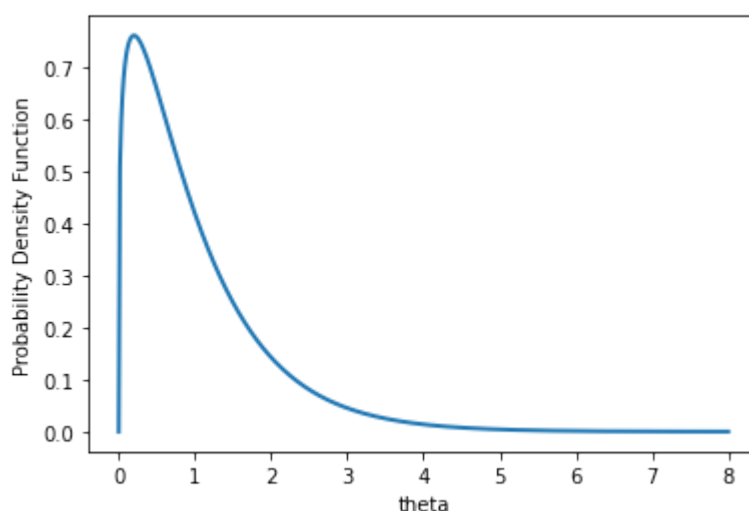
```
In [ ]: N=500
xs = np.linspace(0,8,N)
pdfs = stats.gamma.pdf(xs, 1.25, scale = 1 / 1.25)

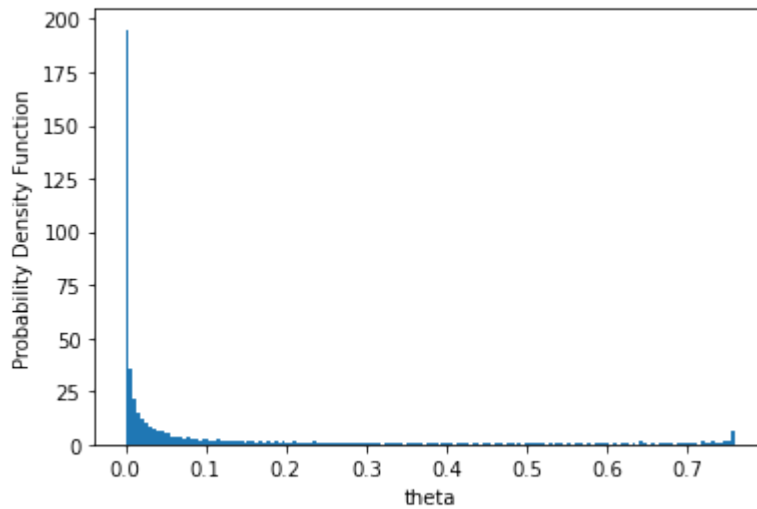
plt.plot(xs, pdfs, linewidth=2)

## add histogram of theta samples from the second model with 160 bins

plt.gca().set_xlabel("theta")
plt.gca().set_ylabel("Probability Density Function")
plt.show()

plt.hist(pdfs, bins=160)
plt.gca().set_xlabel("theta")
plt.gca().set_ylabel("Probability Density Function")
plt.show()
```





Analyzing the diagnostic messages we see `code_4.stan` contains much more divergences than in `code_5.stan` (accordingly 39.54% and 0.01%). This may suggest that adding a constraint on `code_5.stan` improved the model.

Excercise 4 - Selection of parameters using equation solving

```
In [ ]: model_tune = CmdStanModel(stan_file='code_6.stan')

y0 = 1.5 # initial guess for the equation solving

data={'y_guess':[y0],
      'theta':[(F+L)/2]}
tunes = model_tune.sample(data=data, fixed_param=True, iter_sampling=1, iter_war
tunes.stan_variables()
```

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

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

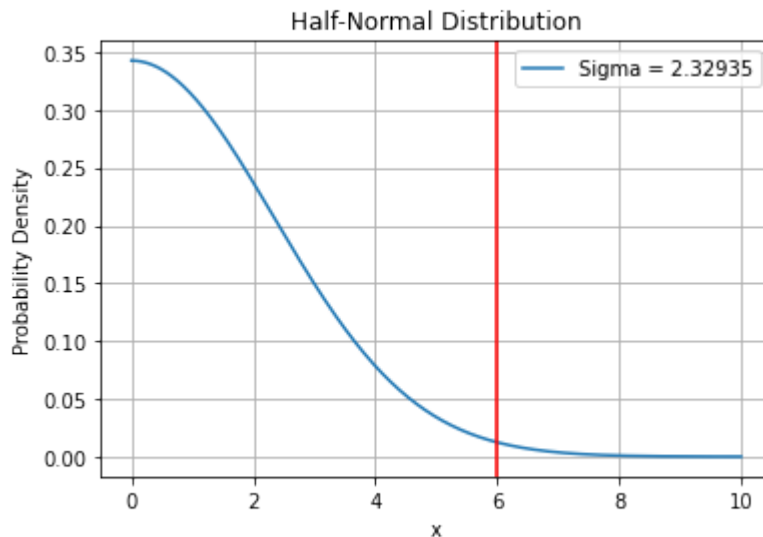
```
Out[ ]: {'sigma': array([2.32935])}
```

```
In [ ]: sigma = 2.32935
x = np.linspace(0, 10, 1000)
x_line = (F+L)/2

# Compute the probability density function (PDF) of the half-normal distribution
pdf = np.sqrt(2 / np.pi) * (1 / sigma) * np.exp(-0.5 * (x / sigma)**2)

# Plot the half-normal distribution
plt.plot(x, pdf, label=f'Sigma = {sigma}')
plt.axvline(x=x_line, color='r')
plt.title('Half-Normal Distribution')
plt.xlabel('x')
plt.ylabel('Probability Density')
plt.legend()
```

```
plt.grid(True)
plt.show()
```



Excercise 5 - different methods of defining models

```
In [ ]: model_samp_st = CmdStanModel(stan_file='code_7.stan')
model_log_target = CmdStanModel(stan_file='code_8.stan')
model_log_target_ind = CmdStanModel(stan_file='code_9.stan')
data = {'N': F}
seed = 6042001 #integer, your date of birth in the DDMMYYYY format without leading zeros
result_1 = model_samp_st.sample(data=data, seed=seed)
result_2 = model_log_target.sample(data=data, seed=seed)
result_3 = model_log_target_ind.sample(data=data, seed=seed)
```

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

```
chain 1 | ████████ | 00:00 Iteration: 1100 / 2000 [ 55%] (Sampling)
```

```
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.
INFO:cmdstanpy:CmdStan start processing
```

```
chain 1 |           | 00:00 Status
```

```
chain 1 | ████████ | 00:00 Iteration: 100 / 2000 [ 5%] (Warmup)
```

```
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.
```

```
INFO:cmdstanpy:CmdStan start processing
```

```
chain 1 |           | 00:00 Status
```

```
chain 1 | ████████ | 00:00 Iteration: 500 / 2000 [ 25%] (Warmup)
```

```
chain 1 | ██████████ | 00:00 Sampling completed
```

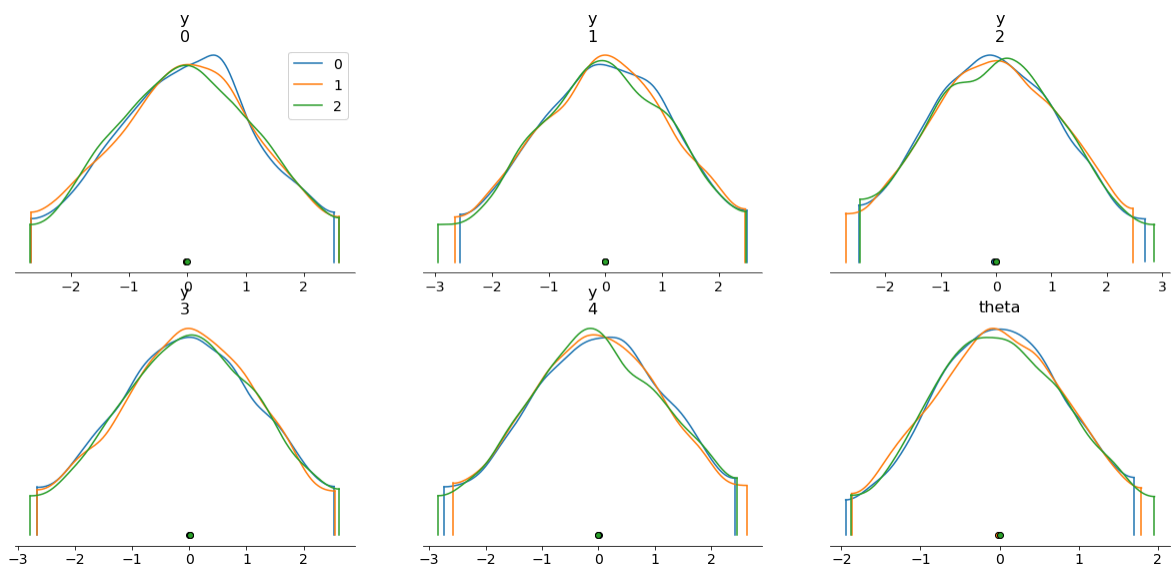
```
chain 2 | ██████████ | 00:00 Sampling completed
```

```
chain 3 | ██████████ | 00:00 Sampling completed
```

```
chain 4 | ██████████ | 00:00 Sampling completed
```

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

```
In [ ]: az.plot_density([result_1,result_2,result_3])
        plt.show()
```



Excercise 6 - generated quantities post sampling

```
In [ ]: model_gq = CmdStanModel(stan_file='code_10.stan')
        # fill in with chosen result from previous exercise
        mean_of_y = model_gq.generate_quantities(data=data,
                                                  mcmc_sample = result_1)
        # investigate the output and plot histogram of mean_y variable
```

```
INFO:cmdstanpy:compiling stan file /home/DA/code_10.stan to exe file /home/DA/cod
e_10
INFO:cmdstanpy:compiled model executable: /home/DA/code_10
INFO:cmdstanpy:Chain [1] start processing
INFO:cmdstanpy:Chain [2] start processing
INFO:cmdstanpy:Chain [3] start processing
INFO:cmdstanpy:Chain [4] start processing
INFO:cmdstanpy:Chain [1] start processing
INFO:cmdstanpy:Chain [2] start processing
INFO:cmdstanpy:Chain [3] start processing
INFO:cmdstanpy:Chain [4] start processing
INFO:cmdstanpy:Chain [2] done processing
INFO:cmdstanpy:Chain [1] done processing
INFO:cmdstanpy:Chain [3] done processing
INFO:cmdstanpy:Chain [4] done processing
```

```
In [ ]: mean_df = mean_of_y.draws_pd()
        for column_name, column_val in mean_df.iteritems():
            plt.figure()
            plt.hist(column_val.values, bins=30)
            plt.title(column_name)
```

