

```
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
```

```
In [ ]: F = 5 #Agata
L = 9 #Swatowska
```

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=0,
                                     chains = 1)
```

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

INFO:cmdstanpy:CmdStan done processing.

```
In [ ]: df = samples.draws_pd()
df
```

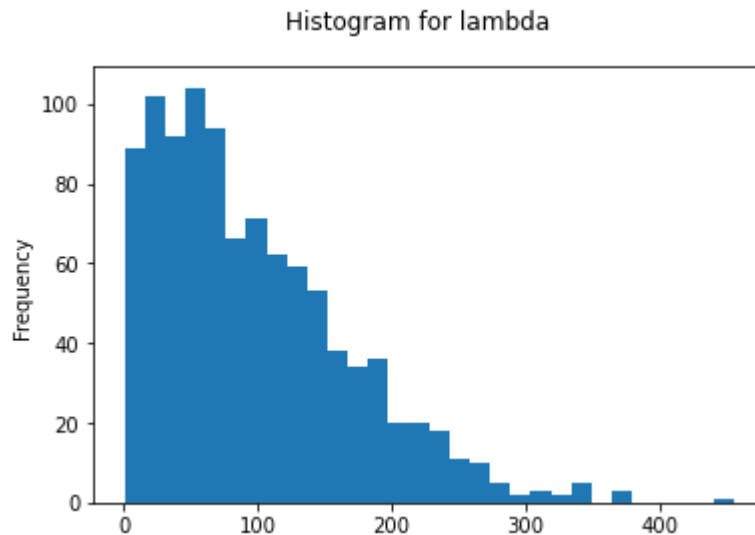
```
Out[ ]:
```

	lp__	accept_stat__	lambda	y_sim[1]	y_sim[2]	y_sim[3]	y_sim[4]	y_sim[5]
0	0.0	0.0	6.08035	6.0	5.0	4.0	6.0	4.0
1	0.0	0.0	64.41750	64.0	64.0	69.0	76.0	63.0
2	0.0	0.0	87.65640	90.0	87.0	77.0	111.0	75.0
3	0.0	0.0	71.77300	54.0	66.0	68.0	73.0	63.0
4	0.0	0.0	153.68400	149.0	167.0	172.0	143.0	160.0
...
995	0.0	0.0	7.21813	7.0	14.0	5.0	11.0	7.0
996	0.0	0.0	28.24400	26.0	27.0	22.0	31.0	27.0
997	0.0	0.0	67.83360	65.0	75.0	69.0	75.0	66.0
998	0.0	0.0	82.55480	80.0	84.0	61.0	82.0	76.0
999	0.0	0.0	193.87900	184.0	201.0	196.0	187.0	218.0

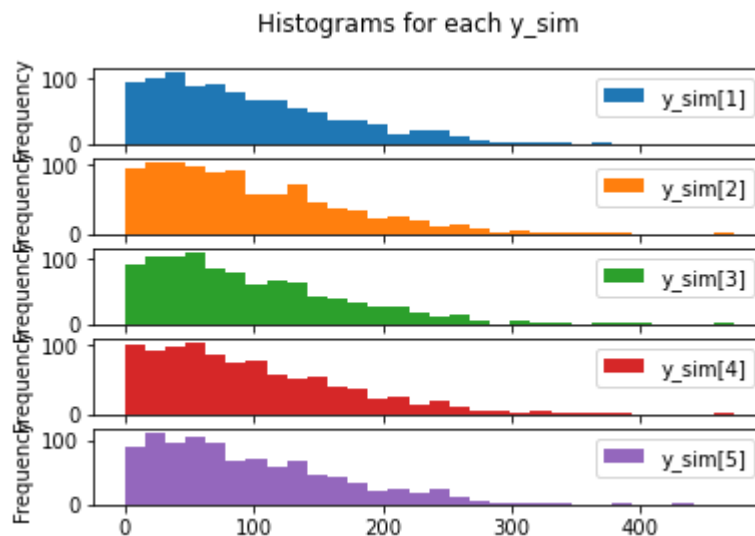
1000 rows × 8 columns

```
In [ ]: lambdas = df['lambda']

lambdas.plot.hist(bins=30)
plt.suptitle("Histogram for lambda")
plt.show()
```



```
In [ ]: df_y = df.drop(df.columns[0:3], axis=1)
df_y.plot.hist(subplots=True, bins=30)
plt.suptitle("Histograms for each y_sim")
plt.show()
```



Excercise 2 - Constraints on the data

```
In [ ]: bern1 = CmdStanModel(stan_file='code_2.stan')
# when using 'y':[0,2] it retrun range error but with 'y':[0,1] its ok
samp_bern1 = bern1.sample(data={'N':2, 'y':[0,1]})
```

```
INFO:cmdstanpy:found newer exe file, not recompiling
INFO:cmdstanpy:CmdStan start processing
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 []:

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

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

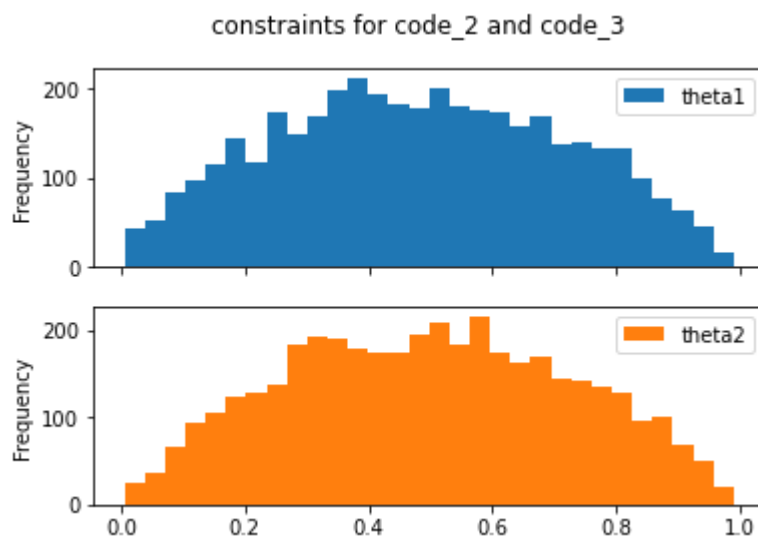
```
chain 1 | ██████████ | 00:00 Sampling completed
chain 2 | ██████████ | 00:00 Sampling completed
chain 3 | ██████████ | 00:00 Sampling completed
chain 4 | ██████████ | 00:00 Sampling completed
```

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

In []:

```
theta1 = samp_bern1.stan_variable('theta')
theta2 = samp_bern2.stan_variable('theta')

df2= pd.DataFrame({'theta1': theta1, 'theta2': theta2})
df2.plot.hist(subplots=True, bins=30)
plt.suptitle("constraints for code_2 and code_3")
plt.show()
```



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_warmup=1000,
        out_gamma1.diagnose())
```

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

```
chain 1 | ████████ | 00:00 Iteration: 4300 / 7000 [ 61%] (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.
```

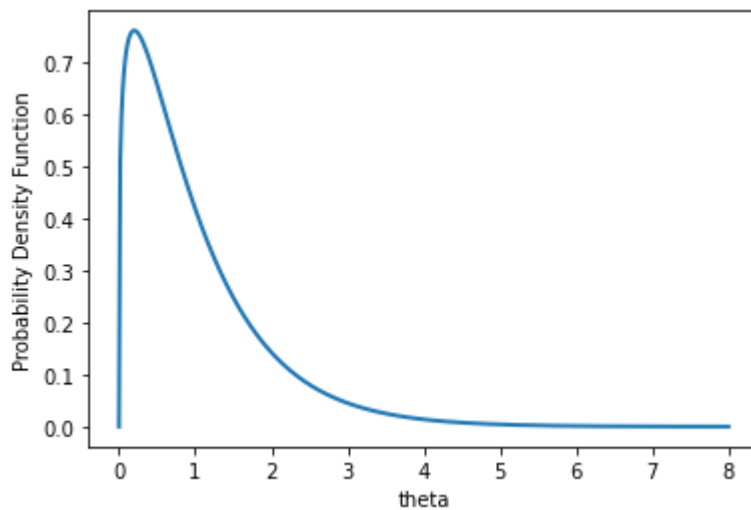
```
Out[ ]: "Processing csv files: C:\\DataAnalytics_VSC\\DataAnalytics\\lab2\\samples\\code_4-20220309194310_1.csv, C:\\DataAnalytics_VSC\\DataAnalytics\\lab2\\samples\\code_4-20220309194310_2.csv, C:\\DataAnalytics_VSC\\DataAnalytics\\lab2\\samples\\code_4-20220309194310_3.csv, C:\\DataAnalytics_VSC\\DataAnalytics\\lab2\\samples\\code_4-20220309194310_4.csv\n\nChecking sampler transitions treedepth.\nTreedepth satisfactory for all transitions.\n\nChecking sampler transitions for divergences.\n8638 of 24000 (35.99%) 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()
```



Constrained parameter

```
In [ ]: model_gm2 = CmdStanModel(stan_file='code_5.stan')
out_gamma2 = model_gm2.sample(output_dir='samples', iter_sampling=6000, iter_warmup=1000,
out_gamma2.diagnose())
```

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

```
chain 1 | ████████ | 00:00 Iteration: 3500 / 7000 [ 50%] (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.
```

```
Out[ ]: "Processing csv files: C:\\DataAnalytics_VSC\\DataAnalytics\\lab2\\samples\\code_5-20220
309194326_1.csv, C:\\DataAnalytics_VSC\\DataAnalytics\\lab2\\samples\\code_5-20220309194
326_2.csv, C:\\DataAnalytics_VSC\\DataAnalytics\\lab2\\samples\\code_5-20220309194326_3.
csv, C:\\DataAnalytics_VSC\\DataAnalytics\\lab2\\samples\\code_5-20220309194326_4.csv\\n
\\nChecking sampler transitions treedepth.\\nTreedepth satisfactory for all transitions.\\n
\\nChecking sampler transitions for divergences.\\n1 of 24000 (0.00%) transitions ended wi
th a divergence.\\nThese divergent transitions indicate that HMC is not fully able to exp
lore the posterior distribution.\\nTry increasing adapt delta closer to 1.\\nIf this does
n't remove all divergences, try to reparameterize the model.\\n\\nChecking E-BFMI - sample
r transitions HMC potential energy.\\nE-BFMI satisfactory.\\n\\nEffective sample size satis
factory.\\n\\nSplit R-hat values satisfactory all parameters.\\n\\nProcessing complete.\\n"
```

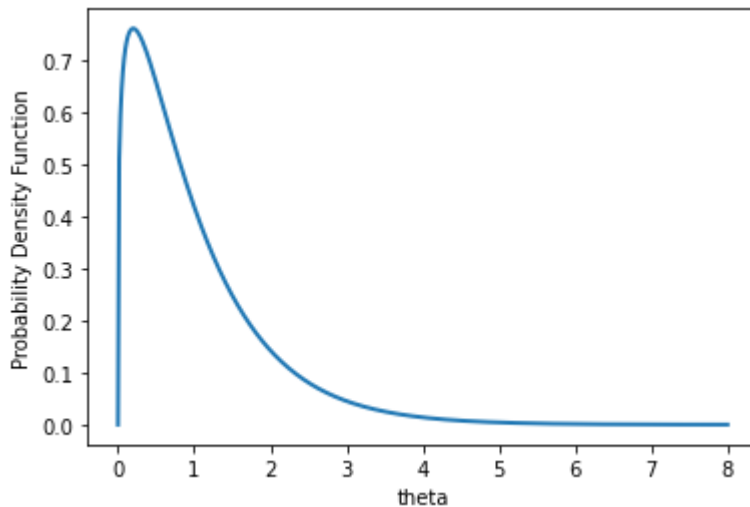
```
In [ ]: N=500
xs = np.linspace(0,8,N)
pdfs = stats.gamma.pdf(xs, 1.25, scale = 1 / 1.25)

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()
```



Excercise 4 - Selection of parameters using equation solving

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

F = 5 # number of letters in the first name
L = 9 # number of letters in the last name
y0 = 1 # 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_warmup=0,
tunes.draws_pd())
```

```
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[ ]:   lp__  accept_stat__  sigma
0  0.0          0.0  2.71757
```

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 = 27051970 # integer, your date of birth in the DDMMYYYY format without leading ze
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 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 Sampling completed
chain 2 | ██████████ | 00:00 Sampling completed
chain 3 | ██████████ | 00:00 Sampling completed
chain 4 | ██████████ | 00:00 Sampling completed
```

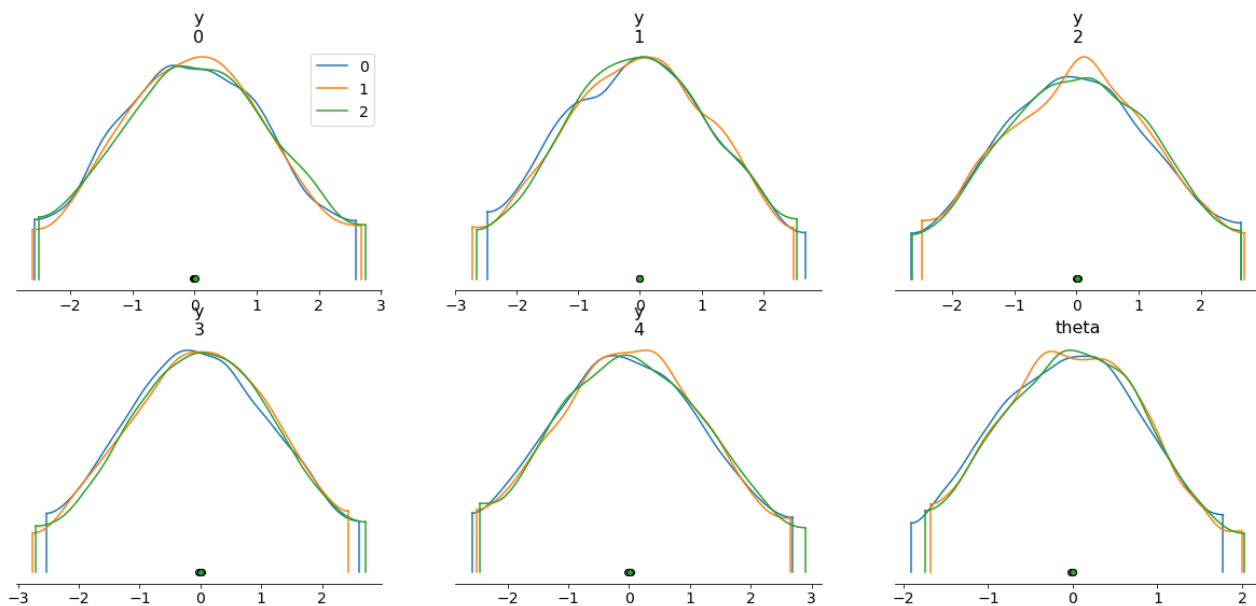
```
INFO:cmdstanpy:CmdStan done processing.
INFO:cmdstanpy:CmdStan start processing
```

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

```
chain 1 | ██████████ | 00:00 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_3)
# investigate the output and plot histogram of mean_y variable
print(mean_of_y.draws().shape)
df = mean_of_y.draws_pd()
print(type(df), df.shape)
df.columns
```

```
INFO:cmdstanpy:found newer exe file, not recompiling
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] done processing
INFO:cmdstanpy:Chain [2] done processing
INFO:cmdstanpy:Chain [3] done processing
INFO:cmdstanpy:Chain [4] done processing
(1000, 4, 1)
<class 'pandas.core.frame.DataFrame'> (4000, 1)
```

```
Out[ ]: Index(['mean_y'], dtype='object')
```

```
In [ ]: df.plot.hist(bins=50)
```

```
Out[ ]: <AxesSubplot:ylabel='Frequency'>
```

