

2.daphne

```
In [2]: from daphne import daphne
import os, json
import numpy as np
import torch
from torch import tensor
import pandas as pd
import matplotlib.pyplot as plt
```

```
In [3]: def ast_helper(fname,directory):
    sugared_fname = '../prob_prog/hw/hw6/CS532-HW6/{}/{}'.format(directory,fname)
    desugared_ast_json_fname = '/Users/gw/repos/prob_prog/' + sugared_fname.replace('.daphne','.json')
    if os.path.isfile(desugared_ast_json_fname):
        with open(desugared_ast_json_fname) as f:
            ast = json.load(f)
    else:
        #note: the sugared path that goes into daphne desugar should be with respect to the daphne path!
        ast = daphne(['desugar-hoppl-cps', '-i', sugared_fname])

    with open(desugared_ast_json_fname, 'w') as f:
        json.dump(ast, f)
    return ast

i=2
fname = '{}.daphne'.format(i)
exp = ast_helper(fname,directory='programs')
%cat programs/2.daphne
```

```
(defn marsaglia-normal [mean var]
  (let [d (uniform-continuous -1.0 1.0)
        x (sample d)
        y (sample d)
        s (+ (* x x) (* y y))]
    (if (< s 1)
      (+ mean (* (sqrt var)
                  (* x (sqrt (* -2 (/ (log s) s))))))
      (marsaglia-normal mean var)))

(let [mu (marsaglia-normal 1 5)
      sigma (sqrt 2)
      lik (normal mu sigma)]
  (observe lik 8)
  (observe lik 9)
  mu)
```

```
In [4]: import smc
import importlib
importlib.reload(smc)
```

Out[4]: <module 'smc' from '/Users/gw/repos/prob_prog/hw/hw6/CS532-HW6/smc.py'>

```
In [5]: n_particles=3
logZ, particles = smc.SMC(n_particles, exp)
```

```
In SMC step 0, Zs: []
In SMC step 1, Zs: [-8.746667201168327]
In SMC step 2, Zs: [-8.746667201168327, -11.342506857689752]
```

```
In [7]: particle_counts = [1,10,100,1000,10000,100000]
fig, axes = plt.subplots(nrows=len(particle_counts),figsize=(30,20))
# fig.tight_layout()
plt.subplots_adjust(left=None, bottom=None, right=None, top=None, wspace=None, hspace=0.5) # https://stackoverflow.com/a/10944330

for idx, n_particles in enumerate(particle_counts):
    logZ, particles = smc.SMC(n_particles, exp)
    samples_array = np.array([sample.item() for sample in particles])
    mean = samples_array.mean()
    var = samples_array.var()
    pd.Series(samples_array).plot.hist(ax=axes[idx], bins=50, title='Program {} | {} particles | mean {:.3f} | var {:.3f}'.format(2, n_particles, mean, var))
```

```
In SMC step 0, Zs: []
In SMC step 1, Zs: [-20.574844409164438]
In SMC step 2, Zs: [-20.574844409164438, -25.219081845127047]
In SMC step 0, Zs: []
In SMC step 1, Zs: [-3.8916753382007996]
In SMC step 2, Zs: [-3.8916753382007996, -2.641245161237912]
In SMC step 0, Zs: []
In SMC step 1, Zs: [-5.708884842017928]
In SMC step 2, Zs: [-5.708884842017928, -3.9738090087117763]
In SMC step 0, Zs: []
In SMC step 1, Zs: [-5.07073953887065]
In SMC step 2, Zs: [-5.07073953887065, -2.9832770513027356]
In SMC step 0, Zs: []
In SMC step 1, Zs: [-5.4072599330000415]
In SMC step 2, Zs: [-5.4072599330000415, -2.9642962996086535]
In SMC step 0, Zs: []
In SMC step 1, Zs: [-5.382312301483098]
In SMC step 2, Zs: [-5.382312301483098, -2.790267592243308]
```



