

## CPSC532W - Homework 5

Some important code snippets:

```

1 class Env(dict):
2     """An environment: a dict of {'var': val} pairs, with an outer Env."""
3     def __init__(self, parms=(), args=(), outer=None):
4         self.update(zip(parms, args))
5         self.outer = outer
6         if outer is not None:
7             self.outer = copy.deepcopy(outer)
8     def find(self, var):
9         """Find the innermost Env where var appears."""
10        return self if (var in self) else self.outer.find(var)
11
12 class Procedure(object):
13     """A user-defined Scheme procedure."""
14     def __init__(self, parms, body, env):
15         self.parms, self.body, self.env = parms, body, env
16     def __call__(self, *args):
17         return evaluate_helper(self.body, Env(self.parms, args, self.env))
18
19 def evaluate_helper(x, env):
20     """Evaluate an expression in an environment."""
21     if type(x) is str and x != 'fn': # variable reference
22         try:
23             return env.find(x)[x]
24         except AttributeError:
25             return x
26
27     elif type(x) in [int, float]: # constant
28         return torch.tensor(float(x))
29
30     elif type(x) is torch.Tensor:
31         return x
32
33     op, *args = x
34
35     if op == 'if': # conditional
36         (test, conseq, alt) = args
37         exp = (conseq if evaluate_helper(test, env) else alt)
38         return evaluate_helper(exp, env)
39
40     elif op == 'fn': # procedure
41         (parms, body) = args
42
43         env_inner = Env(outer=env)
44         return Procedure(parms[1:], body, env_inner)
45
46     elif op == 'sample':
47         d = evaluate_helper(args[1], env)
48         return d.sample()
49
50     elif op == 'observe':
51         return evaluate_helper(args[-1], env)
52
53     else: # procedure call

```

```
36     proc = evaluate_helper(op, env)
37     push_address = env.find("push-address")(*args[0][1:]) # push-
address
38     args_noaddres = args[1:]
39     vals = [evaluate_helper(arg, env) for arg in args_noaddres]
40     return proc(*vals)
```

Well, it works! All tests run, as do all three programs.

## 1 Program 1

Test: 1

elapsed time is: 31.9686455

mean is: 100.117

variance is: 9721.29

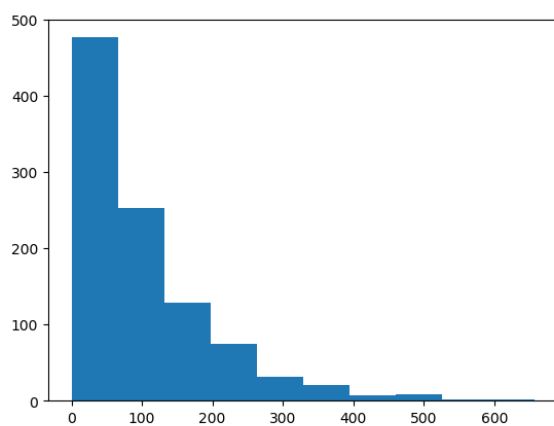


Figure 1: Program 1

## 2 Program 2

Test: 2

elapsed time is: 17.396364499999997

mean is: 0.8550765

variance is: 5.1212597

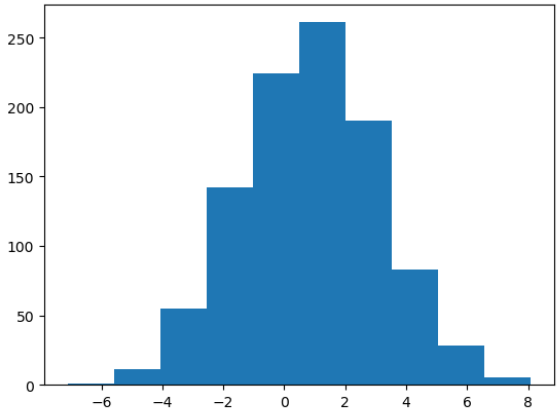


Figure 2: Program 2

3 Program 3

Test: 3  
elapsed time is: 113.54413859999998  
mean is: [1.023, 1.42, 1.478, 1.478, 1.479, 1.467, 1.489, 1.452, 1.478, 1.488, 1.501, 1.496, 1.524, 1.452, 1.485, 1.462, 1.502]  
variance is: [0.680471, 0.5376000000000001, 0.5435160000000001, 0.5655160000000001, 0.5655589999999999, 0.5629109999999999, 0.563879, 0.571696, 0.571516, 0.557856, 0.5379989999999999, 0.5439839999999999, 0.517424, 0.565696, 0.547775, 0.5625560000000001, 0.5319959999999999]

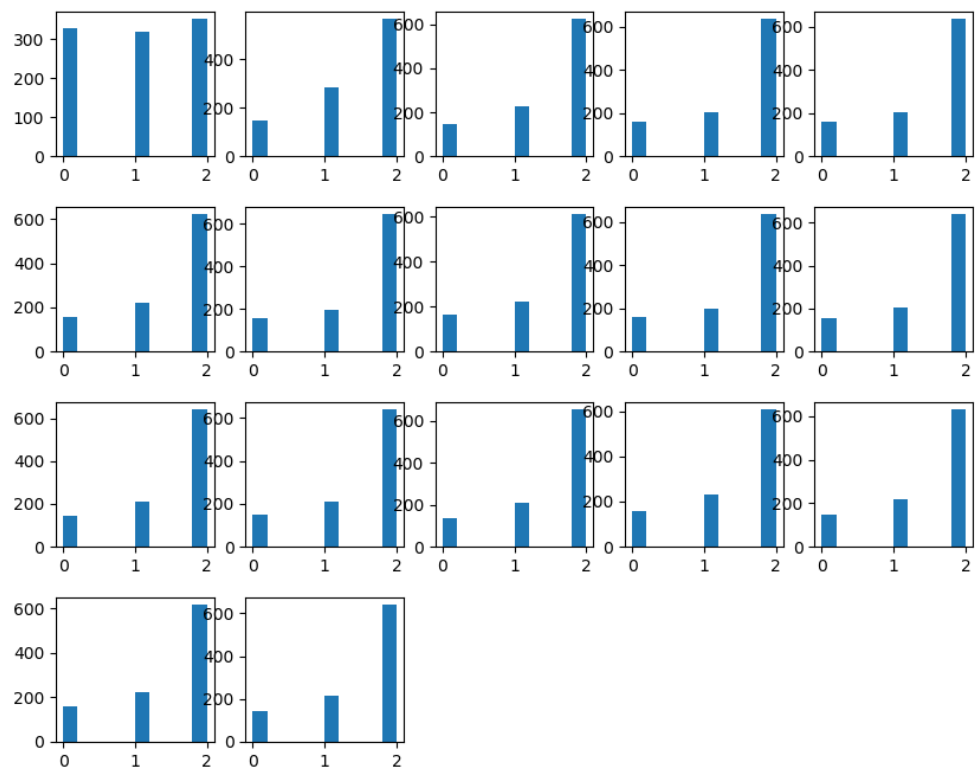


Figure 3: Program 3

Appendix A - Code

<https://github.com/e-vic/cpsc532hw>