Bernoulli example

- **Define the working directory and load CmdStan.m**

In[30]:=
```
(* Linux *)
SetDirectory["~/GitHub/MathematicaStan/Examples/Bernoulli"]

(* Windows *)
(* SetDirectory["C:\\Users\\USER_NAME\\Documents\\Mathematica\\STAN\\Examples\\Bernoulli"] *)

Needs["CmdStan`"]
```

Out[30]= /IS006139/home/pix/GitHub/MathematicaStan/Examples/Bernoulli

- **Generate the Bernoulli Stan code and compile it**

In[32]:=
```
stanCode="data {
  int<lower=0> N;
  int<lower=0,upper=1> y[N];
}
parameters {
  real<lower=0,upper=1> theta;
}
model {
  theta ~ beta(1,1);
  for (n in 1:N)
    y[n] ~ bernoulli(theta);
}";
StanCodeExport["bernoulli",stanCode]

(* Compile your code.
 * Caveat: this can take some time
 *)
StanCompile["bernoulli"]
```

Out[33]= bernoulli.stan

Out[34]= make: '/IS006139/home/pix/GitHub/MathematicaStan/Examples/Bernoulli/bernoulli' is up to date.

- **Generate some data and save them (RDump file)**

In[35]:=
```
n=1000;
y=Table[Random[BernoulliDistribution[0.2016]],{i,1,n}];

RDumpExport["bernoulli",{{"N",n},{"y",y}}];
```

- **Run Stan and get result**

In[38]:=
```
StanRunSample["bernoulli"]

output=StanImport["output.csv"];
```

Out[38]=
```
method = sample (Default)
  sample
    num_samples = 1000 (Default)
    num_warmup = 1000 (Default)
```

```
    save_warmup = 0 (Default)
    thin = 1 (Default)
    adapt
      engaged = 1 (Default)
      gamma = 0.050000000000000003 (Default)
      delta = 0.80000000000000004 (Default)
      kappa = 0.75 (Default)
      t0 = 10 (Default)
      init_buffer = 75 (Default)
      term_buffer = 50 (Default)
      window = 25 (Default)
    algorithm = hmc (Default)
      hmc
        engine = nuts (Default)
          nuts
            max_depth = 10 (Default)
        metric = diag_e (Default)
        stepsize = 1 (Default)
        stepsize_jitter = 0 (Default)
id = 0 (Default)
data
  file = /IS006139/home/pix/GitHub/MathematicaStan/Examples/Bernoulli/bernoulli.data.R
init = 2 (Default)
random
  seed = 3382355977
output
  file = /IS006139/home/pix/GitHub/MathematicaStan/Examples/Bernoulli/output.csv
  diagnostic_file =  (Default)
  refresh = 100 (Default)


Gradient evaluation took 7.7e-05 seconds
1000 transitions using 10 leapfrog steps per transition would take 0.77 seconds.
Adjust your expectations accordingly!


Iteration:    1 / 2000 [  0%]  (Warmup)
Iteration:  100 / 2000 [  5%]  (Warmup)
Iteration:  200 / 2000 [ 10%]  (Warmup)
Iteration:  300 / 2000 [ 15%]  (Warmup)
Iteration:  400 / 2000 [ 20%]  (Warmup)
Iteration:  500 / 2000 [ 25%]  (Warmup)
Iteration:  600 / 2000 [ 30%]  (Warmup)
Iteration:  700 / 2000 [ 35%]  (Warmup)
Iteration:  800 / 2000 [ 40%]  (Warmup)
Iteration:  900 / 2000 [ 45%]  (Warmup)
Iteration: 1000 / 2000 [ 50%]  (Warmup)
Iteration: 1001 / 2000 [ 50%]  (Sampling)
Iteration: 1100 / 2000 [ 55%]  (Sampling)
Iteration: 1200 / 2000 [ 60%]  (Sampling)
Iteration: 1300 / 2000 [ 65%]  (Sampling)
Iteration: 1400 / 2000 [ 70%]  (Sampling)
Iteration: 1500 / 2000 [ 75%]  (Sampling)
Iteration: 1600 / 2000 [ 80%]  (Sampling)
Iteration: 1700 / 2000 [ 85%]  (Sampling)
Iteration: 1800 / 2000 [ 90%]  (Sampling)
Iteration: 1900 / 2000 [ 95%]  (Sampling)
```

```
Iteration: 2000 / 2000 [100%]  (Sampling)

 Elapsed Time: 0.226371 seconds (Warm-up)
               0.298571 seconds (Sampling)
               0.524942 seconds (Total)
```

Use the results

List Header

In[40]:=
```
StanImportHeader[output]
```

Out[40]=
```
{{lp__, 1}, {accept_stat__, 2}, {stepsize__, 3}, {treedepth__, 4},
 {n_leapfrog__, 5}, {divergent__, 6}, {energy__, 7}, {theta, 8}}
```

Show sample matrix

In[41]:=
```
Dimensions[StanImportData[output]]
Take[StanImportData[output],3]
```

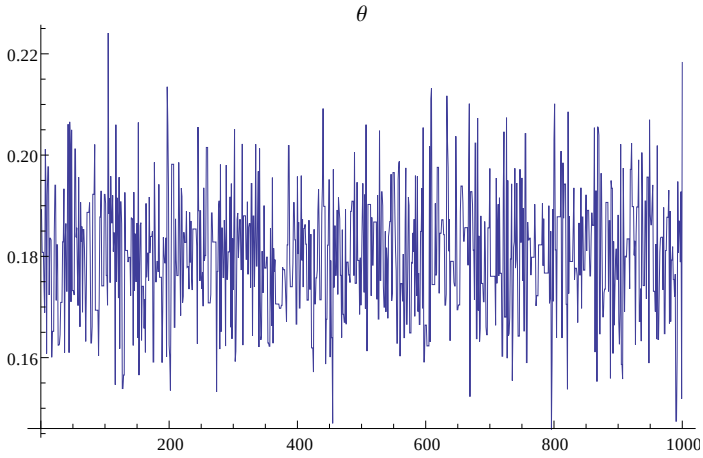
Out[41]=
```
{1000, 8}
```

Out[42]=
```
{{-473.55, 0.505272, 1.80506, 1., 1., 0., 475.647, 0.189243},
 {-473.305, 1., 1.80506, 1., 1., 0., 473.39, 0.180792},
 {-473.305, 0.83586, 1.80506, 1., 1., 0., 473.499, 0.180792}}
```

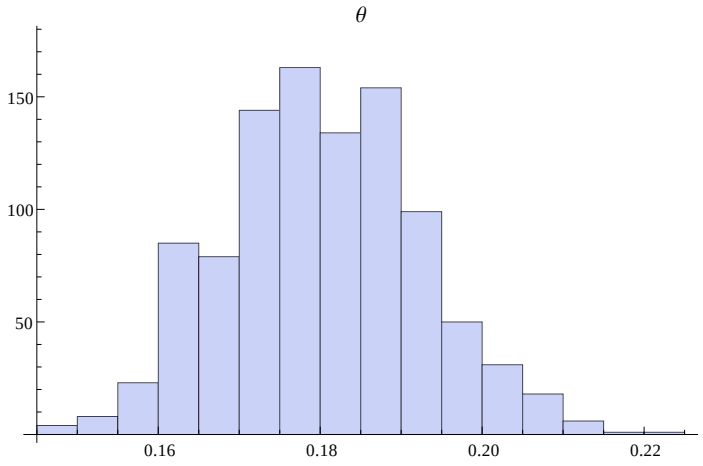
■ Plot θ sample and histogram

In[43]:=
```
ListLinePlot[Flatten[StanVariableColumn["theta", output]], PlotLabel→ "θ"]
Histogram[Flatten[StanVariableColumn["theta", output]], PlotLabel→ "θ"]
```

Out[43]=



Out[44]=



Maximimize likelihood with StanRunOptimize

In[45]:=
```
StanRunOptimize["bernoulli"]
```

Out[45]=
```
method = optimize
  optimize
    algorithm = lbfgs (Default)
      lbfgs
        init_alpha = 0.001 (Default)
        tol_obj = 9.9999999999999998e-13 (Default)
        tol_rel_obj = 10000 (Default)
        tol_grad = 1e-08 (Default)
        tol_rel_grad = 10000000 (Default)
        tol_param = 1e-08 (Default)
        history_size = 5 (Default)
    iter = 2000 (Default)
    save_iterations = 0 (Default)
id = 0 (Default)
data
  file = /IS006139/home/pix/GitHub/MathematicaStan/Examples/Bernoulli/bernoulli.data.R
init = 2 (Default)
random
  seed = 3382357427
output
  file = /IS006139/home/pix/GitHub/MathematicaStan/Examples/Bernoulli/output.csv
  diagnostic_file =  (Default)
  refresh = 100 (Default)

initial log joint probability = -1711.06
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
       6      -471.393   0.000508262   0.000274025           1           1        7
Optimization terminated normally:
  Convergence detected: relative gradient magnitude is below tolerance
```

Options manipulation

In[46]:=
```
StanOptionOptimize[]                                          (* list current options *)
StanSetOptionOptimize["iter",100];                            (* modify the "iter" option *)
StanSetOptionOptimize["output file","output_optimize.csv"];   (* and the "output file" one *)
StanOptionOptimize[]
```

Out[46]=
```
{}
```

Out[47]=
```
{{iter, 100}, {output file, output_optimize.csv}}
```

New run with the new options

In[48]:=
```
StanRunOptimize["bernoulli"]
```

Out[48]=
```
method = optimize
  optimize
    algorithm = lbfgs (Default)
      lbfgs
        init_alpha = 0.001 (Default)
        tol_obj = 9.9999999999999998e-13 (Default)
        tol_rel_obj = 10000 (Default)
        tol_grad = 1e-08 (Default)
        tol_rel_grad = 10000000 (Default)
        tol_param = 1e-08 (Default)
        history_size = 5 (Default)
    iter = 100
    save_iterations = 0 (Default)
id = 0 (Default)
data
  file = /IS006139/home/pix/GitHub/MathematicaStan/Examples/Bernoulli/bernoulli.data.R
init = 2 (Default)
random
  seed = 3382357496
output
  file = output_optimize.csv
  diagnostic_file =  (Default)
  refresh = 100 (Default)

initial log joint probability = -1724.21
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
       6      -471.393   0.000480461   0.000248854           1           1        7
Optimization terminated normally:
  Convergence detected: relative gradient magnitude is below tolerance
```

Overwrite and/or reset option

In[49]:=
```
StanSetOptionOptimize["iter",100]
StanSetOptionOptimize["iter",10]
StanResetOptionOptimize[]
```

Out[49]=
```
{{iter, 100}, {output file, output_optimize.csv}}
```

Out[50]=
```
{{iter, 10}, {output file, output_optimize.csv}}
```

Out[51]=
```
{}
```