Soft-k-means example

- **Compiles Stan code**

```
SetDirectory["~/GitHub/MathematicaStan/Examples/Cluster/"];
Needs["CmdStan`"];
StanCompile["soft-k-means.stan"] (* CAVEAT: takes some time *)
```

```
make: '/home/pix/GitHub/MathematicaStan/Examples/Cluster/soft-k-means' is up to date.
```

Runs generated executable

```
StanRunVariational["soft-k-means"]
```

```
method = variational
  variational
    algorithm = meanfield (Default)
      meanfield
    iter = 10000 (Default)
    grad_samples = 1 (Default)
    elbo_samples = 100 (Default)
    eta = 1 (Default)
    adapt
      engaged = 1 (Default)
      iter = 50 (Default)
    tol_rel_obj = 0.01 (Default)
    eval_elbo = 100 (Default)
    output_samples = 1000 (Default)
id = 0 (Default)
data
  file = /home/pix/GitHub/MathematicaStan/Examples/Cluster/soft-k-means.data.R
init = 2 (Default)
random
  seed = 635825645
output
  file = /home/pix/GitHub/MathematicaStan/Examples/Cluster/output.csv
  diagnostic_file =  (Default)
  refresh = 100 (Default)


This is Automatic Differentiation Variational Inference.

(EXPERIMENTAL ALGORITHM: expect frequent updates to the procedure.)

Gradient evaluation took 0.000253 seconds
1000 iterations under these settings should take 0.253 seconds.
Adjust your expectations accordingly!

Begin eta adaptation.
Iteration:   1 / 250 [  0%]  (Adaptation)
Iteration:  50 / 250 [ 20%]  (Adaptation)
Iteration: 100 / 250 [ 40%]  (Adaptation)
Iteration: 150 / 250 [ 60%]  (Adaptation)
Iteration: 200 / 250 [ 80%]  (Adaptation)
Success! Found best value [eta = 1] earlier than expected.

Begin stochastic gradient ascent.
  iter       ELBO   delta_ELBO_mean   delta_ELBO_med   notes
   100     -8e+02             1.000            1.000
   200     -8e+02             0.502            1.000
   300     -8e+02             0.335            0.004   MEDIAN ELBO CONVERGED

Drawing a sample of size 1000 from the approximate posterior...
COMPLETED.
```

Imports data and variable manipulations

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

- **Prints header data (20 first variables)**

```
Take[StanImportHeader[output],20]
```

```
{{lp__, 1}, {mu.1.1, 2}, {mu.2.1, 3}, {mu.3.1, 4}, {mu.4.1, 5}, {mu.5.1, 6}, {mu.1.2, 7},
 {mu.2.2, 8}, {mu.3.2, 9}, {mu.4.2, 10}, {mu.5.2, 11}, {mu.1.3, 12}, {mu.2.3, 13}, {mu.3.3, 14},
 {mu.4.3, 15}, {mu.5.3, 16}, {mu.1.4, 17}, {mu.2.4, 18}, {mu.3.4, 19}, {mu.4.4, 20}}
```

- **Extract mu for sample 6**

```
StanVariable["mu", output, 6] // MatrixForm
```

$$\begin{pmatrix} 0.229876 & -0.319952 & -0.342192 & -0.20772 & -0.432373 & -2.66269 & 0.823331 & -0.18624 \\ 1.15016 & -0.199826 & 0.65275 & 0.93393 & 1.83303 & 0.0179777 & 1.13992 & 0.058709 \\ -0.730597 & 0.280598 & 1.27422 & -0.524396 & 1.66405 & 0.075491 & 0.0758345 & 0.24539 \\ 1.2511 & 1.6941 & 1.21554 & 1.46439 & -0.470623 & -1.24124 & -0.280543 & 0.483635 \\ -0.353924 & 0.0704483 & 2.29614 & -1.14478 & -0.00308209 & -0.481008 & -0.299382 & 1.23815 \end{pmatrix}$$

```
StanVariable["mu.2.3",output,6]
```

```
{0.65275}
```

- **Extracts the whole column of sample for mu.2.3 (only prints the first 10)**

```
Take[StanVariableColumn["mu.2.3", output], 10] // MatrixForm
```

$$\begin{pmatrix} 0.626577 \\ 0.642296 \\ 0.613605 \\ 0.566757 \\ 0.361053 \\ 0.65275 \\ 0.137155 \\ 0.497903 \\ 0.528791 \\ 0.729885 \end{pmatrix}$$

- **Computes mean and standard deviation for all variables**

```
StanVariableFunc["mu", output, Mean] // MatrixForm
StanVariableFunc["mu", output, StandardDeviation] // MatrixForm
```

$$\begin{pmatrix} 0.0660412 & -0.0310603 & -0.365404 & 0.155462 & -0.739835 & -2.59777 & 0.757785 & 0.0215279 \\ 1.30083 & 0.15199 & 0.636815 & 0.82174 & 1.69953 & 0.200379 & 1.00365 & 0.182425 \\ -0.531263 & 0.464708 & 0.373633 & -0.425722 & 1.56698 & -0.0890072 & 0.172922 & 0.327113 \\ 1.38314 & 1.72403 & 0.914609 & 0.758489 & -0.574967 & -1.42689 & -0.316061 & 0.197621 \\ -0.658575 & 0.118586 & 2.06253 & -1.37109 & -0.170198 & -0.381739 & -0.211064 & 0.730086 \end{pmatrix}$$

$$\begin{pmatrix} 0.204028 & 0.182921 & 0.174298 & 0.202643 & 0.215292 & 0.20043 & 0.199626 & 0.179017 \\ 0.233909 & 0.225535 & 0.215279 & 0.289487 & 0.181643 & 0.212475 & 0.21113 & 0.219633 \\ 0.318148 & 0.206751 & 0.270064 & 0.289944 & 0.338243 & 0.220693 & 0.30345 & 0.22364 \\ 0.308764 & 0.263655 & 0.314065 & 0.342926 & 0.271846 & 0.282111 & 0.25822 & 0.319209 \\ 0.153389 & 0.200542 & 0.193284 & 0.191812 & 0.179134 & 0.171717 & 0.18514 & 0.176691 \end{pmatrix}$$