```
html"""
<style>
    main {
        margin: 0 auto;
        max-width: 2000px;
        padding-left: max(160px, 10%);
        padding-right: max(160px, 10%);
    }
</style>
"""
```

```
using Pkg
```

```
begin
    using CSV    , DataFrames    ,
    NamedTupleTools
    using InferenceObjects
    using StanSample
end
```

```
stan_schools = """
data {
    int<lower=0> J;
    real y[J];
    real<lower=0> sigma[J];
}

parameters {
    real mu;
    real<lower=0> tau;
    real theta_tilde[J];
}

transformed parameters {
    real theta[J];
    for (j in 1:J)
        theta[j] = mu + tau *
theta_tilde[j];
}

model {
    mu ~ normal(0, 5);
    tau ~ cauchy(0, 5);
    theta_tilde ~ normal(0, 1);
    y ~ normal(theta, sigma);
}

generated quantities {
    vector[J] log_lik;
    vector[J] y_hat;
    for (j in 1:J) {
        log_lik[j] = normal_lpdf(y[j] |
theta[j], sigma[j]);
        y_hat[j] = normal_rng(theta[j],
sigma[j]);
    }
}
""";
```

```
data = Dict(
    "J" => 8,
    "y" => [28.0, 8.0, -3.0, 7.0, -1.0,
1.0, 18.0, 12.0],
    "sigma" => [15.0, 10.0, 16.0, 11.0,
9.0, 11.0, 10.0, 18.0]
);
```

```
begin
    m_schools =
    SampleModel("eight_schools",
    stan_schools)
    rc = stan_sample(m_schools; data,
    save_warmup=true)
end;
```

```
/var/folders/l7/pr04h0650q5dvqttnvs8s
2c00000gn/T/jl_oNldP3/eight_schools.s
tan updated.
```

```
InferenceData
▶ posterior
▶ posterior_predictive
▶ log_likelihood
▶ sample_stats
▶ observed_data
▶ warmup_posterior
▶ warmup_posterior_predictive
▶ warmup_sample_stats
▶ warmup_log_likelihood
```

```
if success(rc)

    idata = inferencedata(m_schools;
log_likelihood_var=:log_lik,
posterior_predictive_var=:y_hat)
    idata = merge(idata, from_namedtuple(;
observed_data = namedtuple(data)))
else
    @warn "Sampling failed."
end
```

To see more details, click on any of the triangles above or specify group as shown below.

```
Dataset with dimensions: Dim{:draw}, Dim{:chair
and 4 layers:
  :theta_tilde Float64 dims: Dim{:draw}, Dim{:c
  :mu          Float64 dims: Dim{:draw}, Dim{:c
  :tau         Float64 dims: Dim{:draw}, Dim{:c
  :theta       Float64 dims: Dim{:draw}, Dim{:c

with metadata Dict{String, Any} with 1 entry:
  "created_at" => "2022-12-21T07:31:35.401"
```

```
idata.posterior
```

```
Dataset with dimensions: Dim{:sigma_dim_1}, Dim
and 3 layers:
  :sigma Float64 dims: Dim{:sigma_dim_1} (8)
  :J     Int64 dims:
  :y     Float64 dims: Dim{:y_dim_1} (8)

with metadata Dict{String, Any} with 1 entry:
  "created_at" => "2022-12-21T07:31:37.032"
```

```
if :observed_data in propertynames(idata)
    idata.observed_data
end
```

	sigma_dim_1	y_dim_1	sigma	J	y
1	1	1	15.0	8	28.0
2	2	1	10.0	8	28.0
3	3	1	16.0	8	28.0
4	4	1	11.0	8	28.0
5	5	1	9.0	8	28.0
6	6	1	11.0	8	28.0
7	7	1	10.0	8	28.0
8	8	1	18.0	8	28.0
9	1	2	15.0	8	8.0
10	2	2	10.0	8	8.0
	more				
64	8	8	18.0	8	12.0

```
DataFrame(idata.observed_data)
```

```
(:theta_tilde, :mu, :tau, :theta)
```

```julia
keys(idata.posterior)
```

```
post_schools =
```

	mu	tau	theta_tilde.1	theta_til
1	3.52231	4.7065	0.0271936	-0.0177
2	6.63695	2.74395	0.908655	-0.9163
3	6.01454	0.752124	1.35384	2.16331
4	9.47459	1.31025	0.817419	1.93847
5	3.66894	1.59465	-0.215355	1.87681
6	-0.59466	1.35707	-0.17536	-2.0038
7	1.55325	4.38612	-0.321583	-1.1820
8	7.44546	2.65873	0.46783	1.76013
9	5.69612	6.10343	1.63567	0.90004
10	4.37191	2.16671	1.43176	1.37798
more				
4000	4.43746	7.08411	0.422206	1.31232

```julia
post_schools = read_samples(m_schools,
:dataframe; start=1001)
```

```
posterior_schools =
```

	draw	chain	theta_tilde_dim_1	theta_di
1	1001	1	1	1
2	1002	1	1	1
3	1003	1	1	1
4	1004	1	1	1
5	1005	1	1	1
6	1006	1	1	1
7	1007	1	1	1
8	1008	1	1	1
9	1009	1	1	1
10	1010	1	1	1
more				
256000	2000	4	8	8

```
posterior_schools =
DataFrame(idata.posterior)
```

```
(16384000, 12)
```

```
DataFrame(inferencedata(m_schools).posteri
or) |> size
```

	draw	chain	school	y_hat_dim_1	log_
1	1001	1	1	1	1
2	1002	1	1	1	1
3	1003	1	1	1	1
4	1004	1	1	1	1
5	1005	1	1	1	1
6	1006	1	1	1	1
7	1007	1	1	1	1
8	1008	1	1	1	1
9	1009	1	1	1	1
10	1010	1	1	1	1
more					
2048000	2000	4	8	8	8

```
DataFrame(inferencedata(m_schools; dims=
(theta=[:school], theta_tilde=
[:school])).posterior)
```

```
(2048000, 11)
```

```
DataFrame(inferencedata(m_schools; dims=
(theta=[:school], theta_tilde=
[:school])).posterior) |> size
```