R Markdown with Other Engines

Yingqi Jing

December 26, 2019

Contents

List of Tables
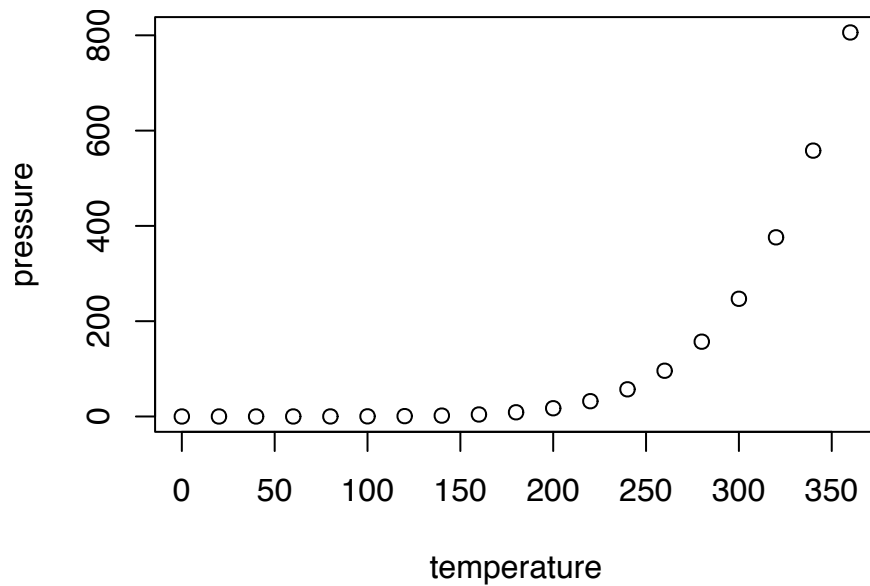
List of Figures

Figure S1: Relationship between temperature and presure

S1 R Markdown

S1.1 Including Plots

You can also embed plots, for example:

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

S1.1.1 subsubsection

```r
summary(cars)
```

S2 Python code chunk

```python
x = 'hello, python world!'
print(x.split(' '))
```

```
['hello,', 'python', 'world!']
```

S3 C++ code chunk

```cpp
#include <Rcpp.h>
using namespace Rcpp;
// [[Rcpp::export]]
NumericVector timesTwo(NumericVector x) {
  return x * 2;
}
```

```r
timesTwo(10) # test function in R chunk or console
```

```
[1] 20
```

S4 Bash or Fish script

```
echo "Hello Bash"
```

```
echo "Hello Fish"
```

S5 Stan code chunk

We can assign the stan code to a variable ($model1$), and can use this later in the R code chunk.

```
parameters {
  real y[2];
}
model {
  y[1] ~ normal(0, 1);
  y[2] ~ double_exponential(0, 2);
}
```

```
fit <- sampling(model1, chains = 1)
```

```
SAMPLING FOR MODEL '78478c2aa59249012e782886a3af321e' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 1e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.1 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 0.013726 seconds (Warm-up)
Chain 1:                0.013279 seconds (Sampling)
Chain 1:                0.027005 seconds (Total)
Chain 1:
```

```
print(fit)
```

```
Inference for Stan model: 78478c2aa59249012e782886a3af321e.
1 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=1000.

      mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
y[1]  0.04    0.04 0.98 -1.97 -0.56  0.05  0.64  1.97   510 1.00
y[2]  0.00    0.13 2.84 -6.21 -1.34 -0.03  1.30  6.11   466 1.00
```

```
lp__  -1.45    0.08 1.31 -5.06 -1.87 -1.12 -0.57 -0.10    289 1.01
```

Samples were drawn using NUTS(diag_e) at Thu Dec 26 20:22:45 2019.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).