exe3_lab3_bayesian learning

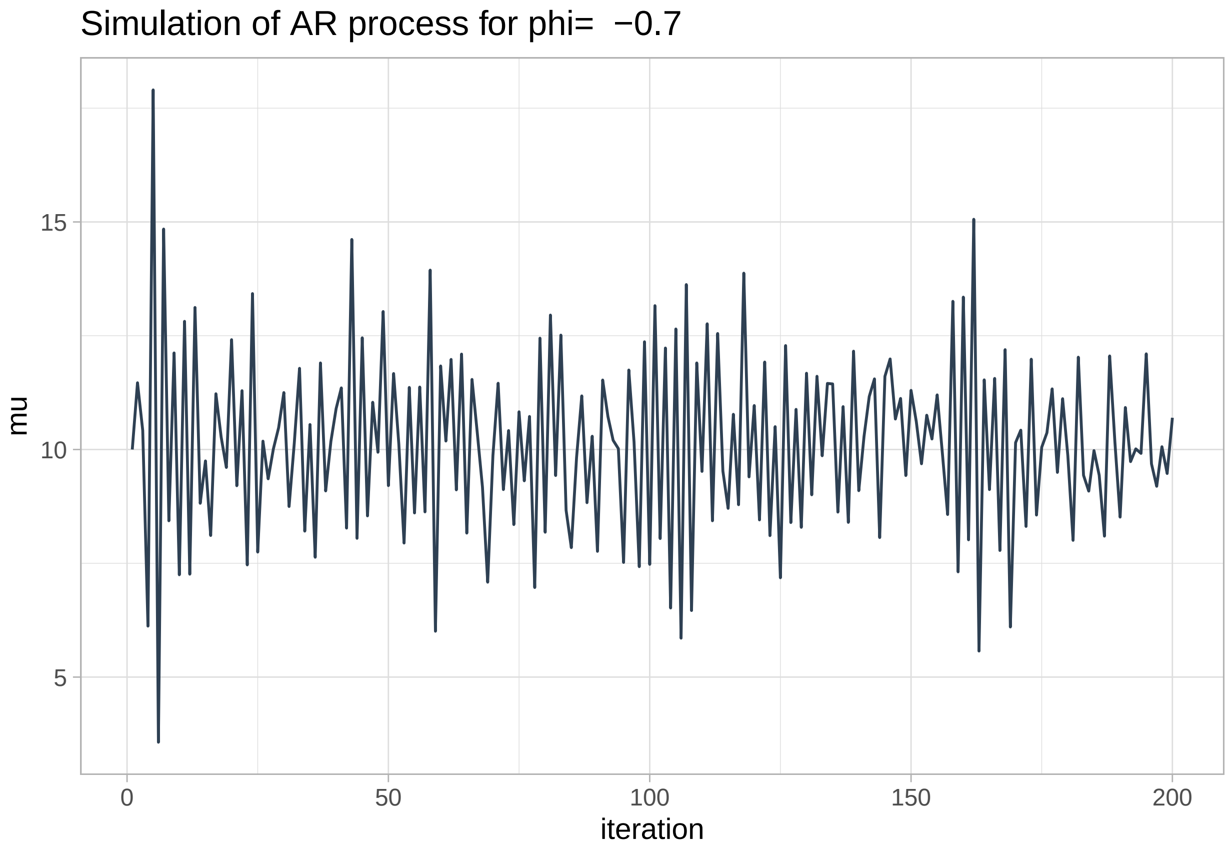
zahra jalilpour

2021-06-05

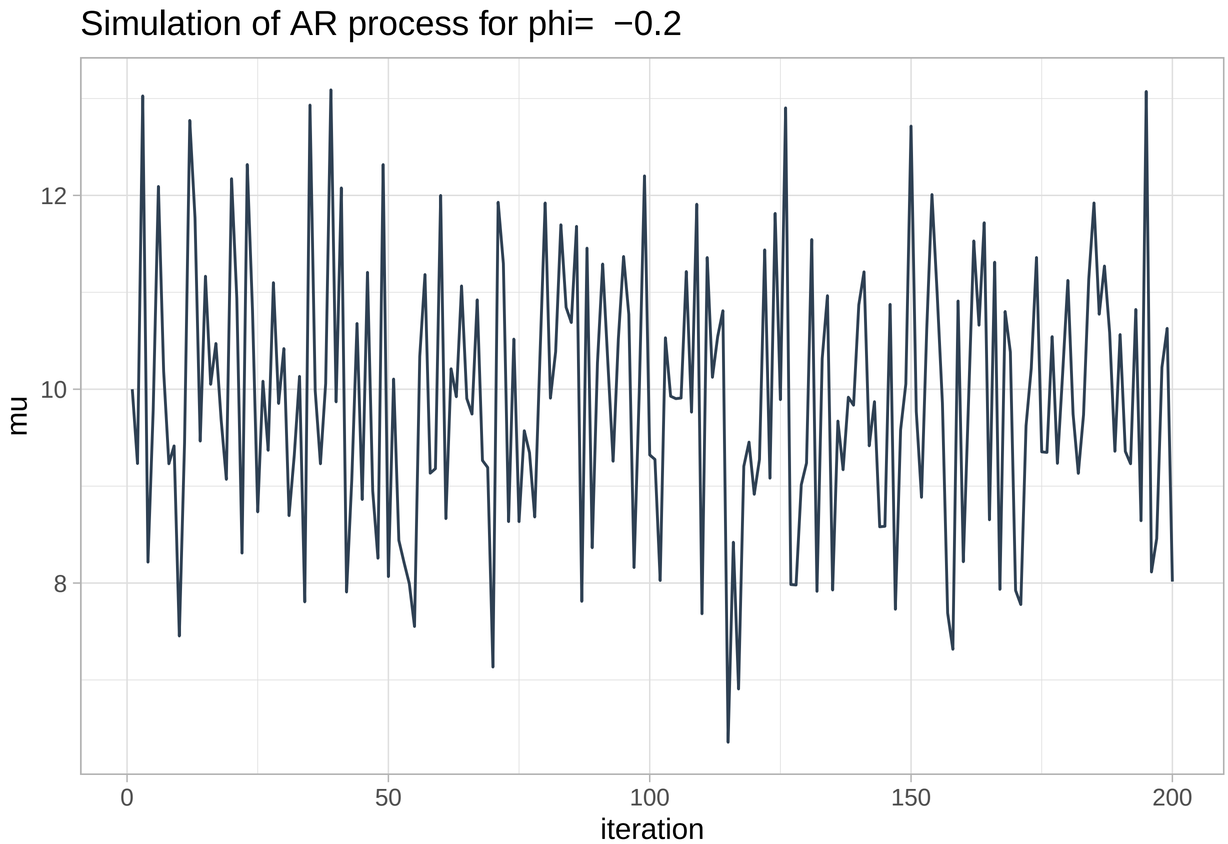
3

a

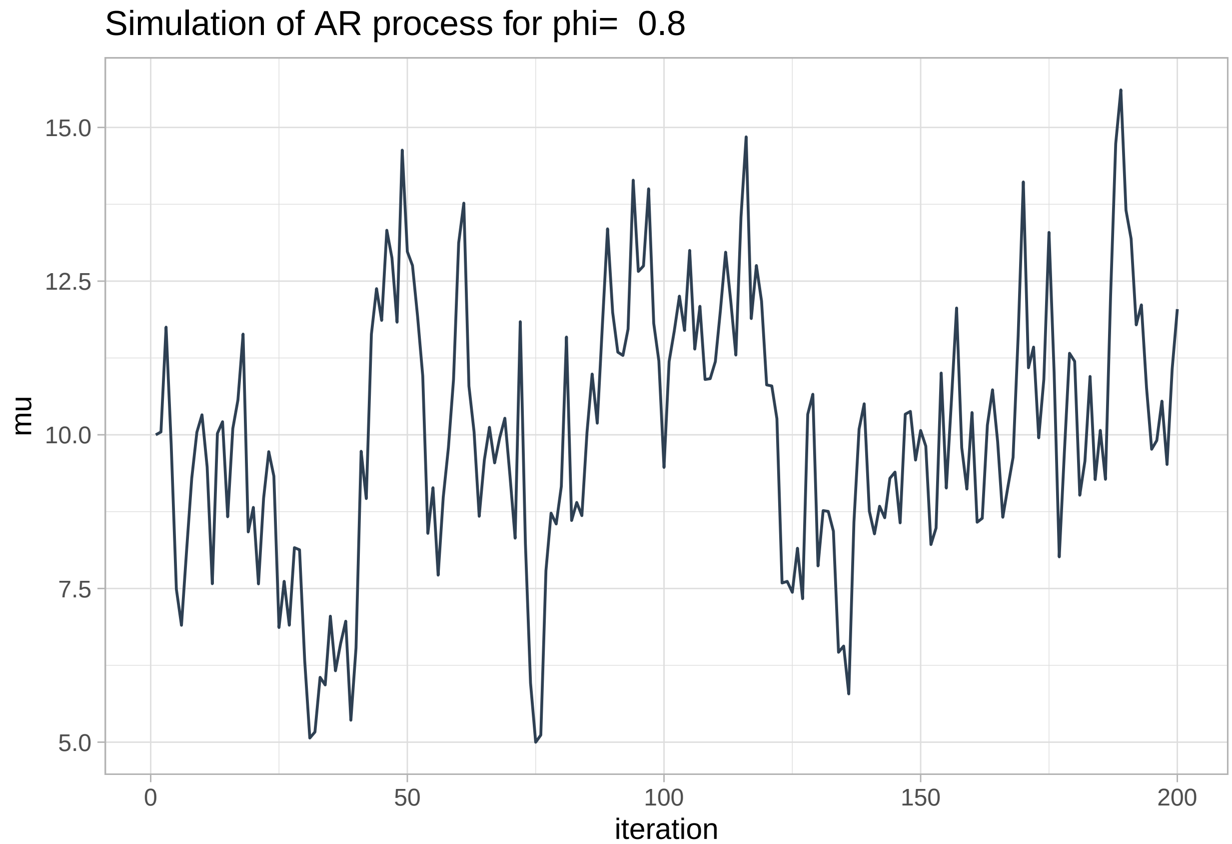
```
T<-200
mu<- 20
sigma_sq<- 4
AR_process <- function(mu, phi,  T,sigma_sq){
  x<- rep(0, T)
  x[1]<- mu

  for (t in 1:(T-1)){
    x[t+1] <- mu + phi * (x[t] - mu) + rnorm(1, mean=0, sd=sqrt(sigma_sq))
  }

return(x)
}
simulation_AR <- function(mu, phi,  T,sigma_sq ){
  sim <- AR_process(mu, phi,  T,sigma_sq)
  AR_df <- data.frame(x=1:T, y=sim)
  AR_plot <- ggplot(AR_df)+
    geom_line(aes(x=x, y=y), color= "#2E4053")+
    labs(title= paste("Simulation of AR process for phi= ",phi), x="iteration", y="mu",color="legend")+
    theme_light()

  return(AR_plot)
}

plot1 <- simulation_AR(mu= 10, phi=-0.7, T=200, sigma_sq=2)
plot2 <- simulation_AR(mu= 10, phi=-0.2, T=200, sigma_sq=2)
plot3 <- simulation_AR(mu= 10, phi=0.8, T=200, sigma_sq=2)
plot4 <- simulation_AR(mu= 10, phi=1, T=200, sigma_sq=2)
par(mfrow=c(3,3))
plot1
```


Simulation of AR process for phi= −0.7
mu
15
10
5
0
50
100
150
200
iteration

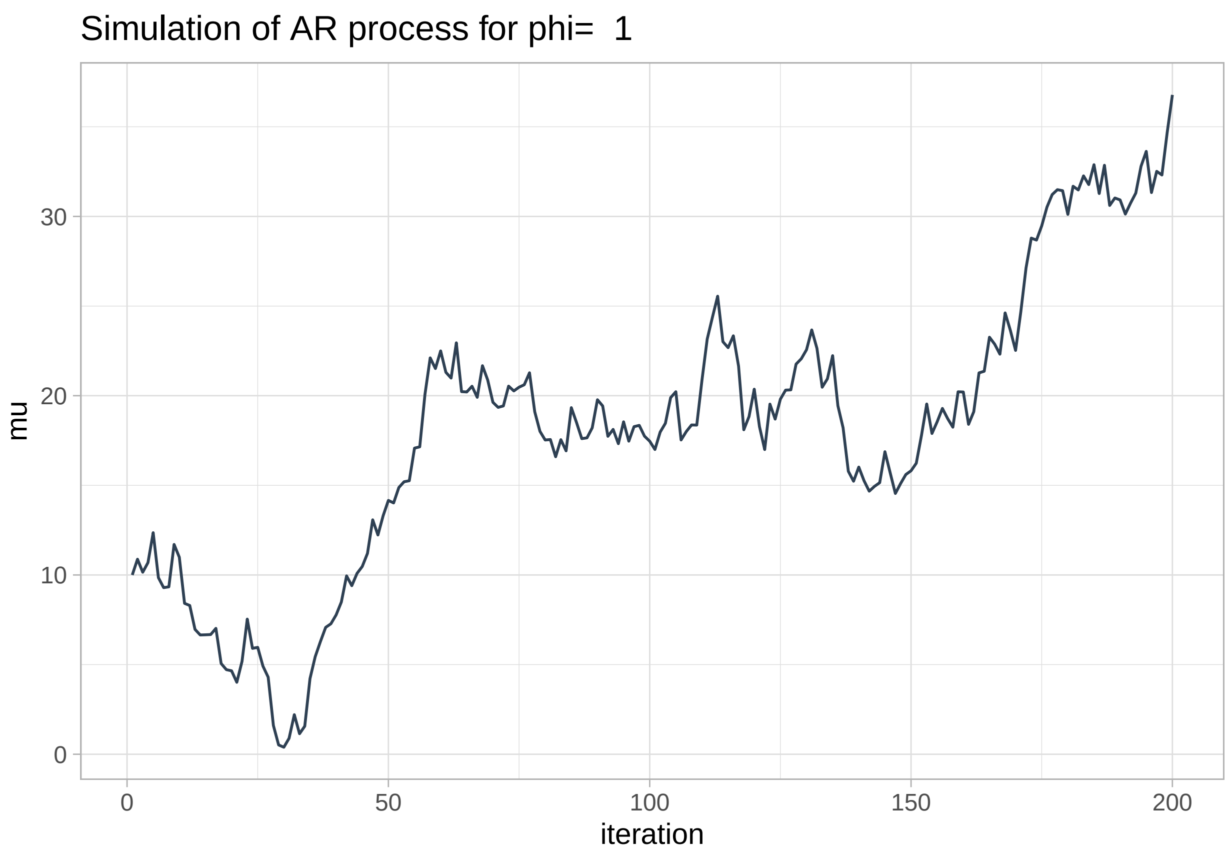
```
plot2
```


Simulation of AR process for phi= −0.2
mu
12
10
8
0
50
100
150
200
iteration

```
plot3
```


Simulation of AR process for phi= 0.8
15.0
12.5
10.0
7.5
5.0
mu
0
50
100
150
200
iteration

```
plot4
```



b

```
T<-200
mu<- 20
sigma_sq<- 4
AR_process <- function(mu, phi,  T,sigma_sq){
  x<- rep(0, T)
  x[1]<- mu

  for (t in 1:(T-1)){
    x[t+1] <- mu + phi * (x[t] - mu) + rnorm(1, mean=0, sd=sqrt(sigma_sq))
  }

return(x)
}



X = AR_process(mu = 20, T = 200, phi = 0.3, sigma_sq = 4)
Y = AR_process(mu = 20, T = 200, phi = 0.9, sigma_sq = 4)
# Defining Stan model

stanModel = '
data {
```

```
  int<lower=0> N;
  vector[N] y;
}
parameters {
  real mu;
  real phi;
  real<lower=0> sigma_sq;
}
model {
  mu~normal(0,100);
  sigma_sq ~ scaled_inv_chi_square(1,2);
  for (t in 2:N){
    y[t] ~ normal(mu+(y[t-1]-mu)*phi,sigma_sq);
  }
}'

stan_ModelX = stan(model_code = stanModel,
                   data = list(N = length(X), y = X),
                   warmup = 1000,
                   iter = 2000,chains=1)
```

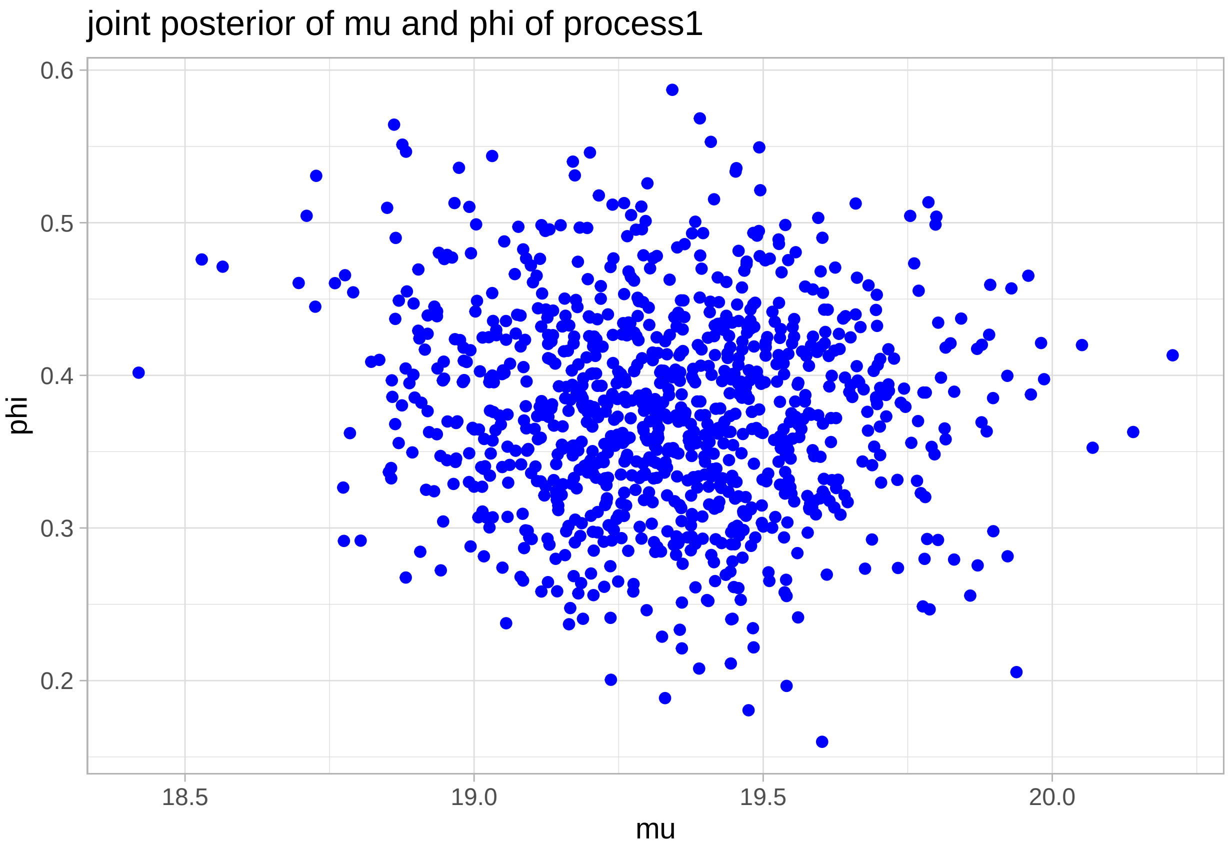
```
##
## SAMPLING FOR MODEL '921503b0102e145fb3979540254edde9' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## 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: 0.535 seconds (Warm-up)
## Chain 1:                0.134 seconds (Sampling)
## Chain 1:                0.669 seconds (Total)
## Chain 1:
```

```
stan_ModelY = stan(model_code = stanModel,
                   data = list(N = length(Y), y = Y),
                   warmup = 1000,
                   iter = 2000,chains=1)
```

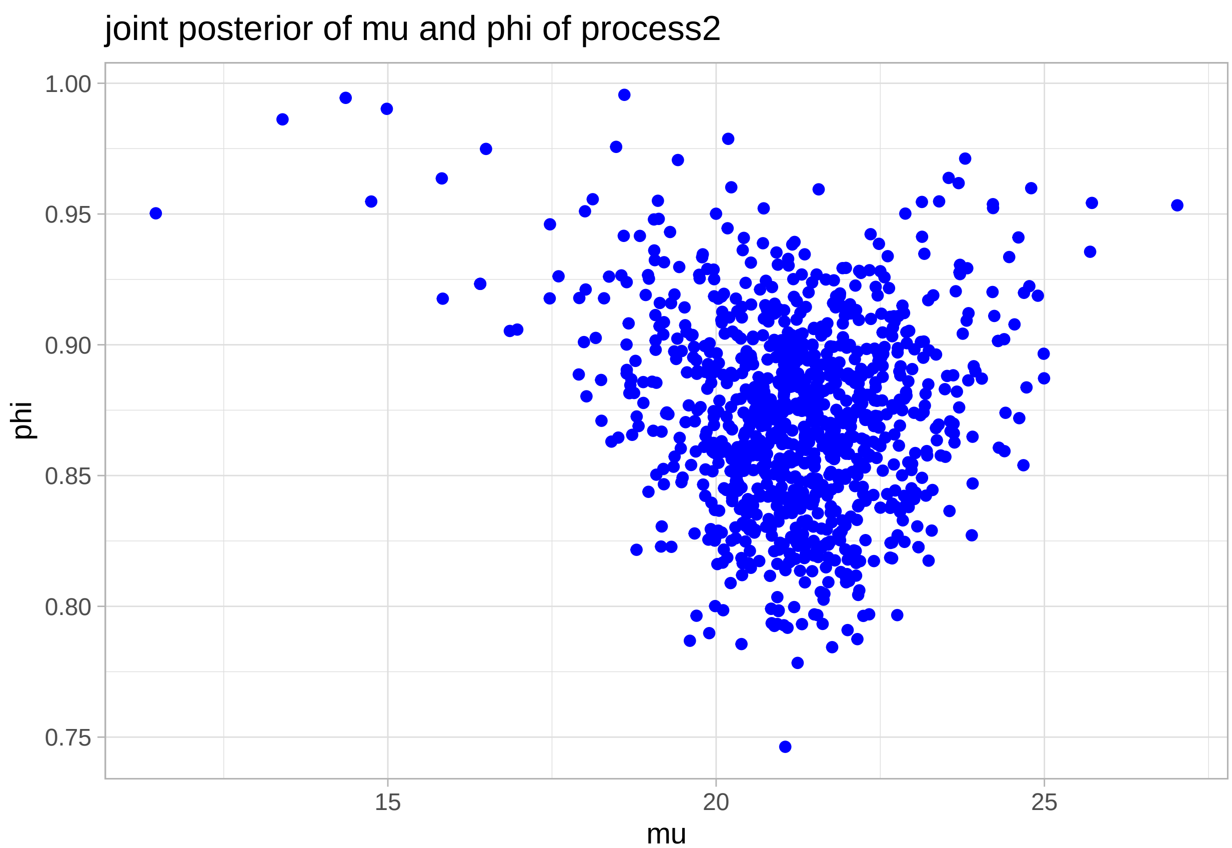
```
##
```

```
## SAMPLING FOR MODEL '921503b0102e145fb3979540254edde9' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 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: 0.289 seconds (Warm-up)
## Chain 1:                0.151 seconds (Sampling)
## Chain 1:                0.44 seconds (Total)
## Chain 1:
```

```
# extract posterior sample
posterior_X = extract(stan_ModelX)
posterior_paramsX = As.mcmc.list(stan_ModelX)
posterior_Y = extract(stan_ModelY)
posterior_paramsY = As.mcmc.list(stan_ModelY)



new_df <- data.frame(mu_x=posterior_X$mu, phi_x=posterior_X$phi, mu_y=posterior_Y$mu, phi_y=posterior_Y$
ggplot(new_df)+
  geom_point(aes(x=mu_x, y= phi_x),color="blue")+
  labs(title="joint posterior of mu and phi of process1", y="phi", x="mu", color="Legend")+
  theme_light()
```



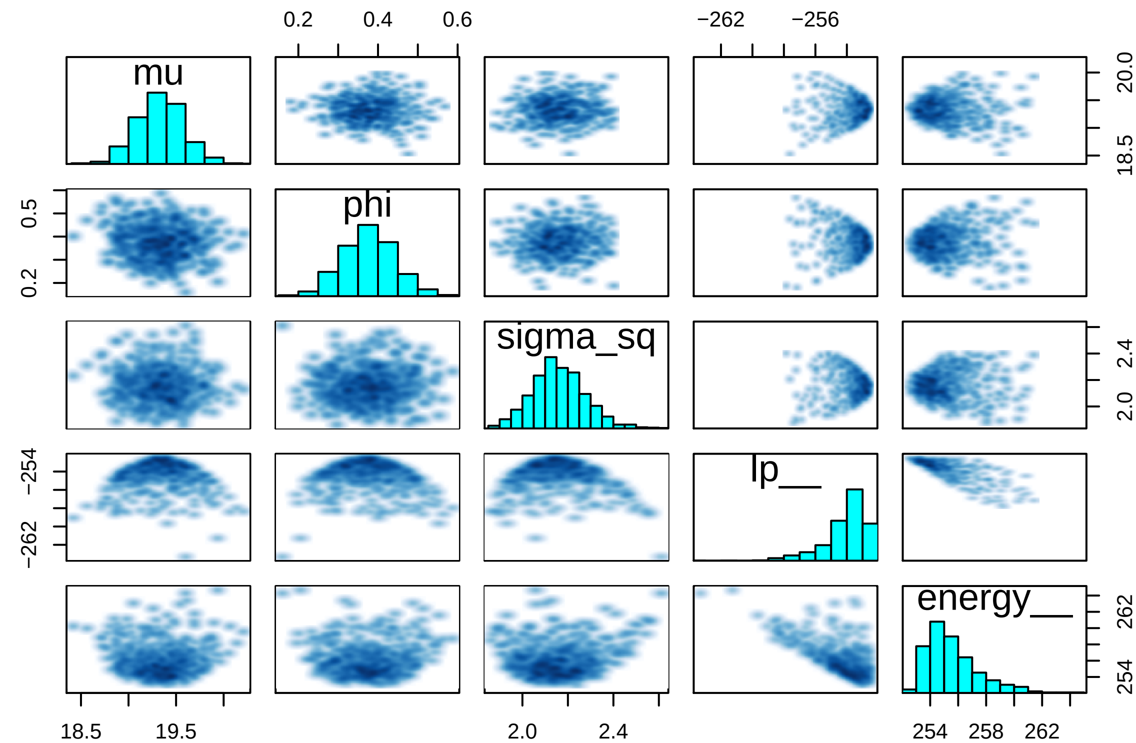

```
ggplot(new_df)+
  geom_point(aes(x=mu_y, y= phi_y),color="blue")+
  labs(title="joint posterior of mu and phi of process2", y="phi", x="mu", color="Legend")+
  theme_light()
```




```
stan_ModelX
```

```
## Inference for Stan model: 921503b0102e145fb3979540254edde9.
## 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
## mu          19.33    0.01 0.25   18.86   19.17   19.33   19.48   19.82   735
## phi          0.38    0.00 0.07    0.25    0.33    0.38    0.42    0.51  1106
## sigma_sq     2.16    0.00 0.12    1.94    2.08    2.15    2.23    2.40  1041
## lp__      -254.06    0.06 1.34 -257.68 -254.66 -253.67 -253.11 -252.57   452
##           Rhat
## mu        1.00
## phi       1.00
## sigma_sq 1.00
## lp__      1.01
##
## Samples were drawn using NUTS(diag_e) at Sat Jun 05 16:43:05 2021.
## 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).
```

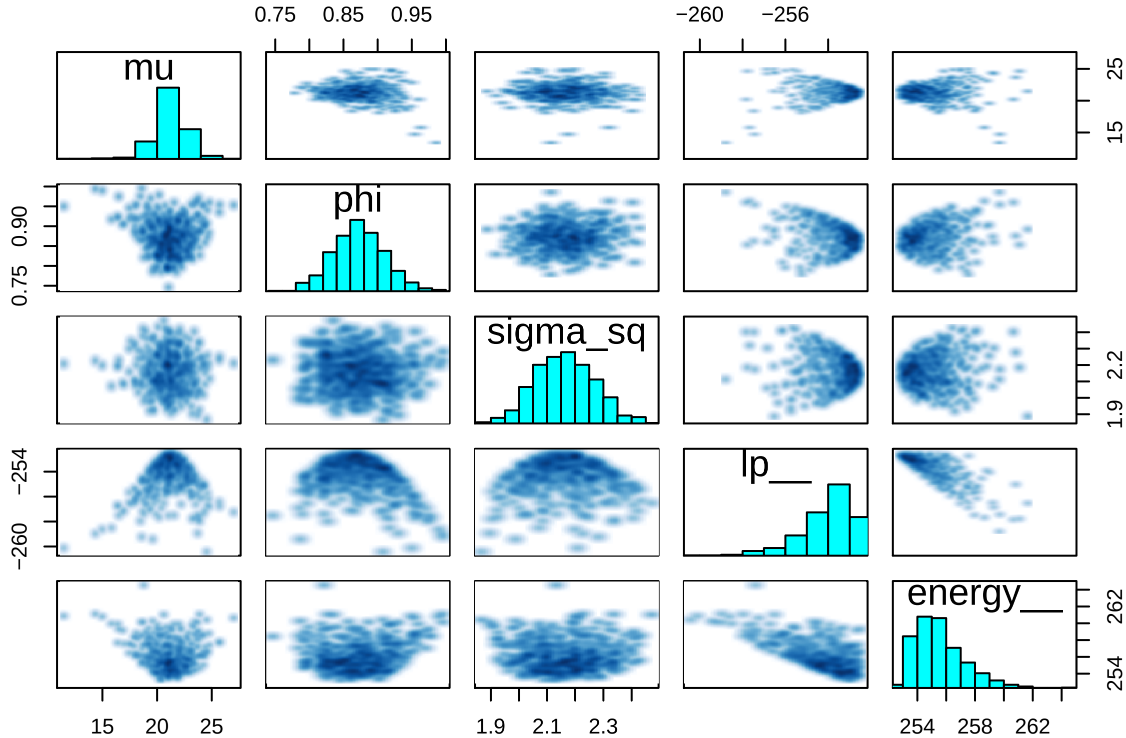
```
pairs(stan_ModelX)
```




```
stan_ModelY
```

```
## Inference for Stan model: 921503b0102e145fb3979540254edde9.
## 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
## mu         21.22    0.08 1.47   18.29   20.45   21.24   22.05   24.05   369
## phi         0.87    0.00 0.04    0.80    0.85    0.87    0.90    0.95   494
## sigma_sq    2.16    0.00 0.10    1.97    2.09    2.16    2.24    2.37   887
## lp__     -254.03    0.07 1.25 -257.39 -254.62 -253.71 -253.08 -252.61   284
##          Rhat
## mu       1.01
## phi      1.00
## sigma_sq 1.00
## lp__     1.00
##
## Samples were drawn using NUTS(diag_e) at Sat Jun 05 16:43:06 2021.
## 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).
```

```
pairs(stan_ModelY)
```



mu
phi
sigma_sq
lp__
energy__
0.75
0.85
0.95
−260
−256
25
15
0.90
0.75
2.2
1.9
−254
−260
262
254
15
20
25
1.9
2.1
2.3
254
258
262