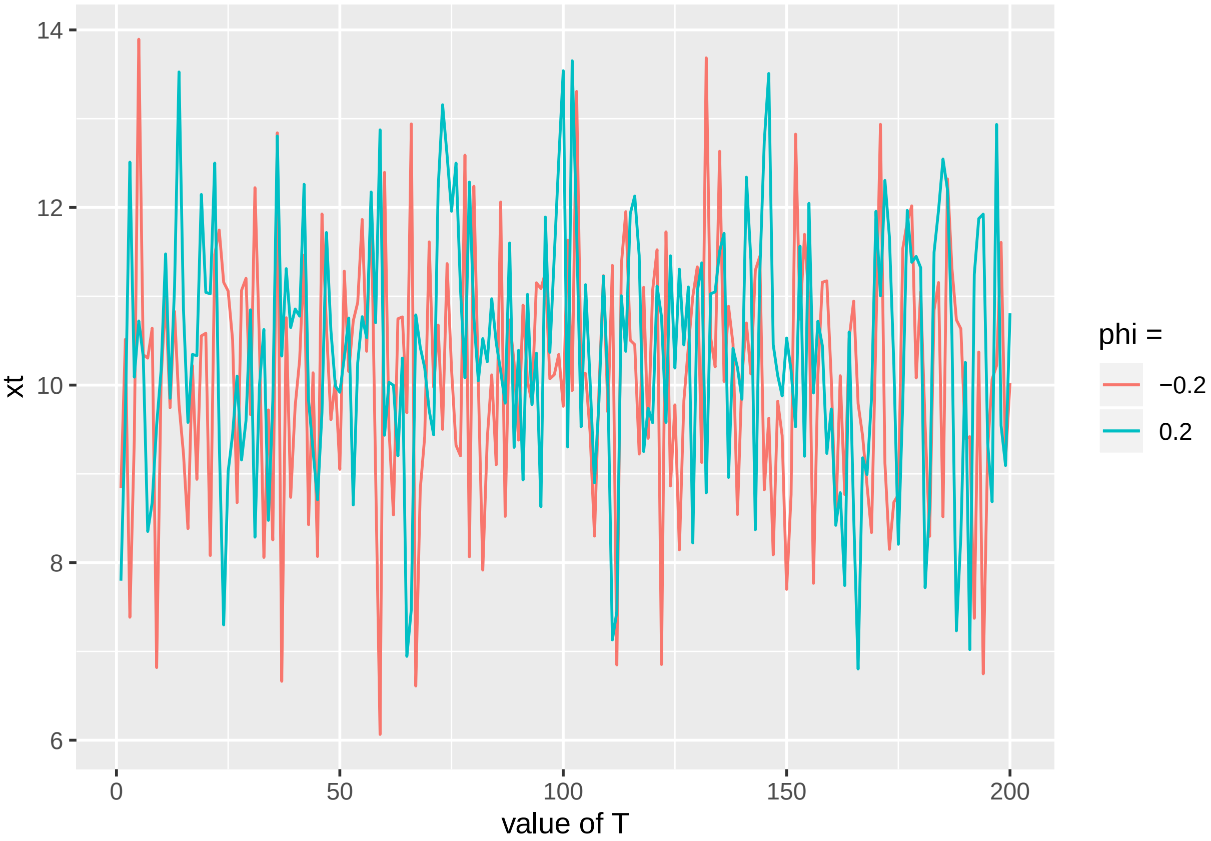
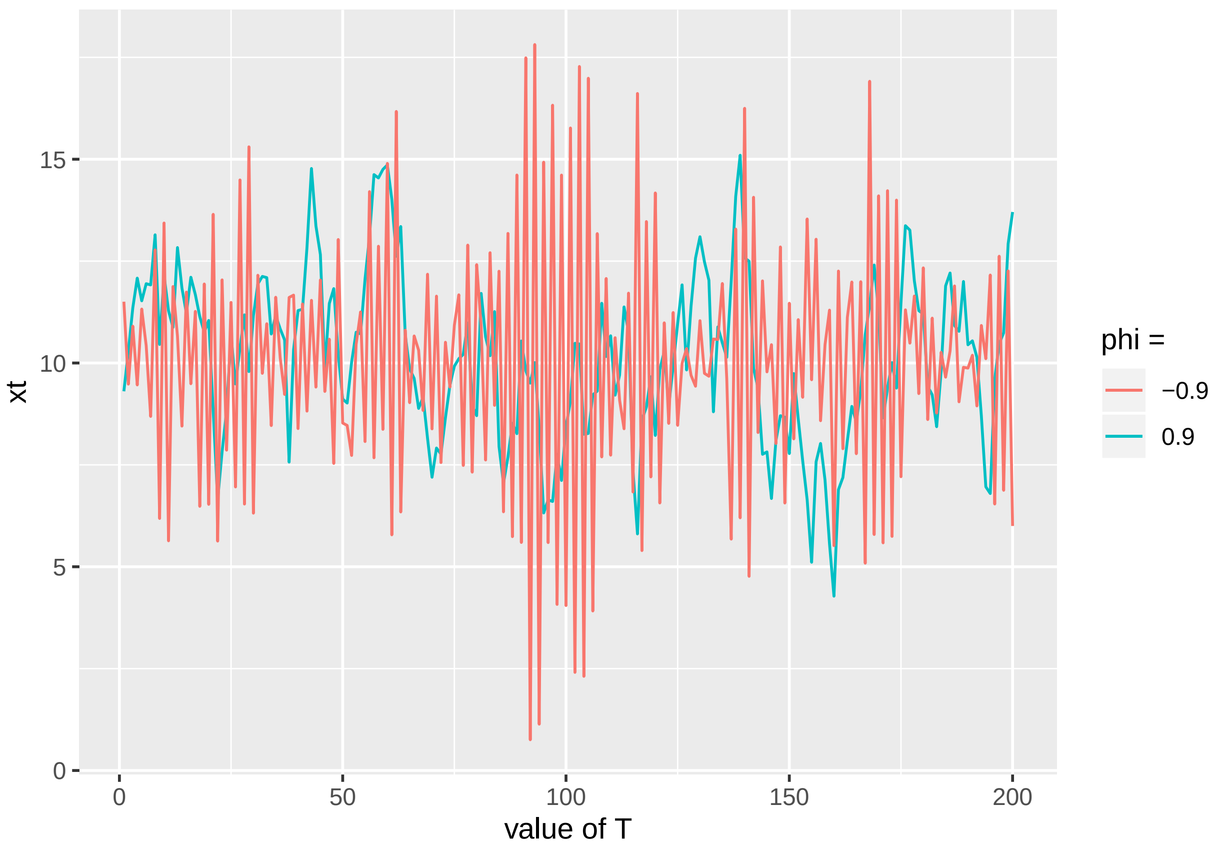
Lab 4 - TDDE07

Axel Holmberg (axeho681), Wilhelm Hansson (wilha431)

1

a)

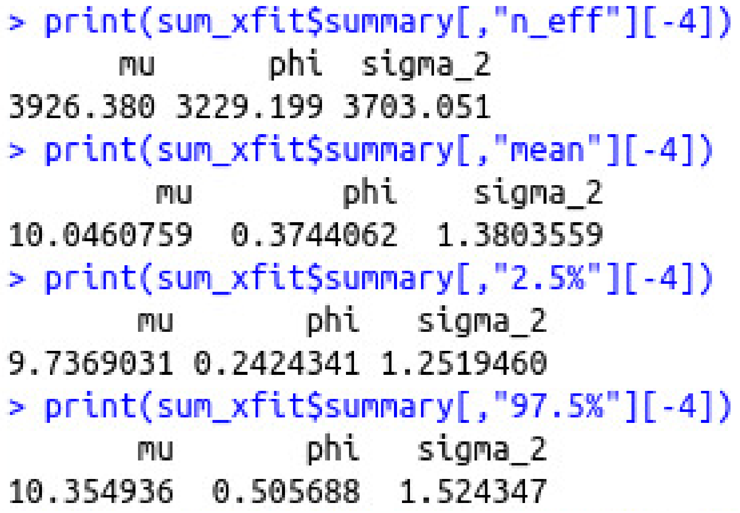




As one can see in the plot above, the observed variance of x_t is lower for lower values of ϕ. One can also see a difference between large positive and large negative values of ϕ.

b)

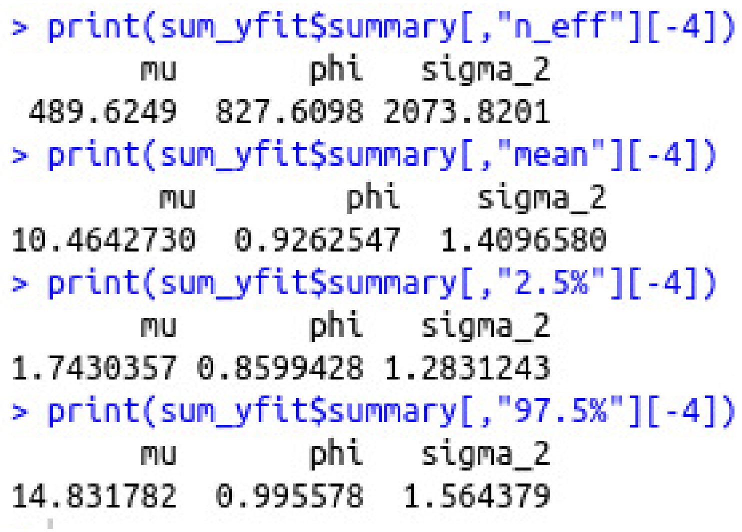
i)



```
> print(sum_xfit$summary[,"n_eff"][-4])
      mu      phi  sigma_2 
3926.380 3229.199 3703.051 
> print(sum_xfit$summary[,"mean"][-4])
        mu        phi    sigma_2 
10.0460759  0.3744062  1.3803559 
> print(sum_xfit$summary[,"2.5%"][-4])
       mu       phi   sigma_2 
9.7369031 0.2424341 1.2519460 
> print(sum_xfit$summary[,"97.5%"][-4])
       mu       phi   sigma_2 
10.354936  0.505688  1.524347 
```

Figure 1: Output for x

We are almost able to estimate the true values. As the true value of $\mu = 10$, which is within the confidence interval and close to the mean. This also goes for $\sigma = \sqrt{2}$ and is estimated to be 1.38 and $\phi = 0.3$ and is estimated to be 0.375.



```
> print(sum_yfit$summary[,"n_eff"][-4])
       mu       phi   sigma_2 
 489.6249  827.6098 2073.8201 
> print(sum_yfit$summary[,"mean"][-4])
        mu        phi    sigma_2 
10.4642730  0.9262547  1.4096580 
> print(sum_yfit$summary[,"2.5%"][-4])
       mu       phi   sigma_2 
1.7430357 0.8599428 1.2831243 
> print(sum_yfit$summary[,"97.5%"][-4])
       mu       phi   sigma_2 
14.831782  0.995578  1.564379 
```

Figure 2: Output for y

This estimation is not as good. This is particularly the case for μ. μ is estimated to be about 10.5, however the 95 % CI for μ has a lower limit of 1.74 and an upper limit of 14.83. Comparing this with xfit which has a 95 % CI for μ with a lower limit of 9.7 and an upper limit of 10.35, which is much narrower. Although, the estimation for σ and ϕ is almost as good as for xfit. Here $\phi = 0.95$.

ii)

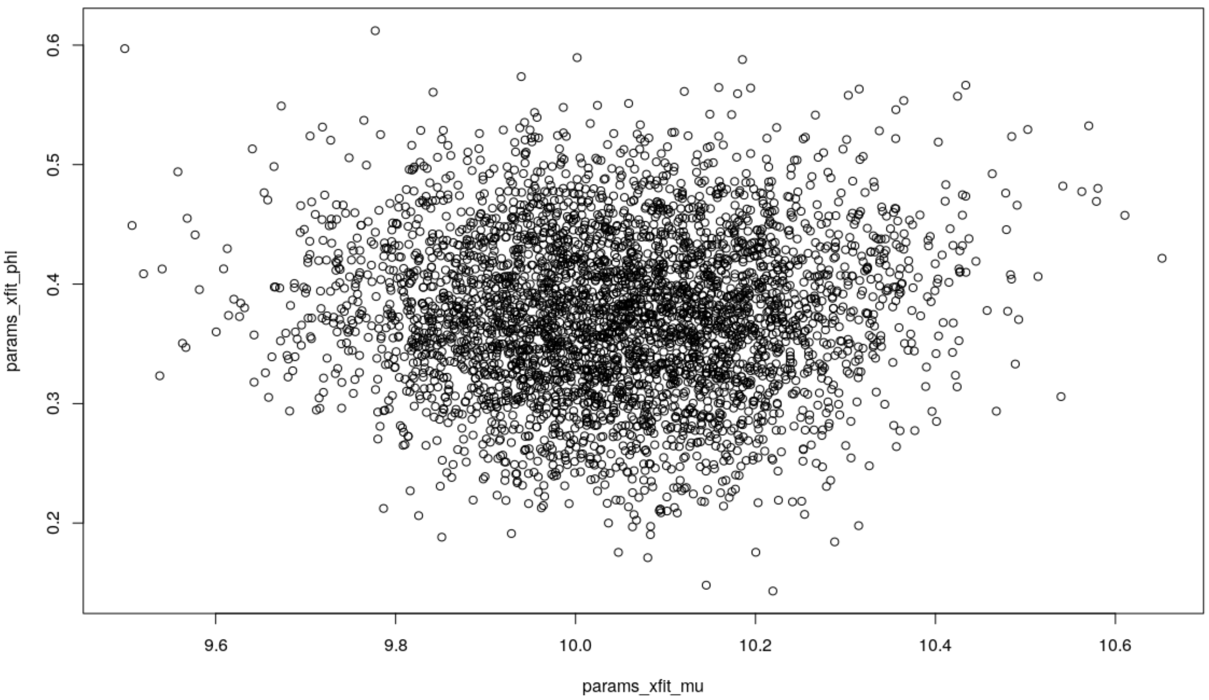


Figure 3: Convergence for of mu and phi for x

The plot above shows the joint posterior of ϕ and μ for data based on $\phi = 0.3$. As one can see there is a high density in the middle of the cluster of the sampled values. The middle of the cluster is the mean of each of the paramaters, which can be seen above in i).

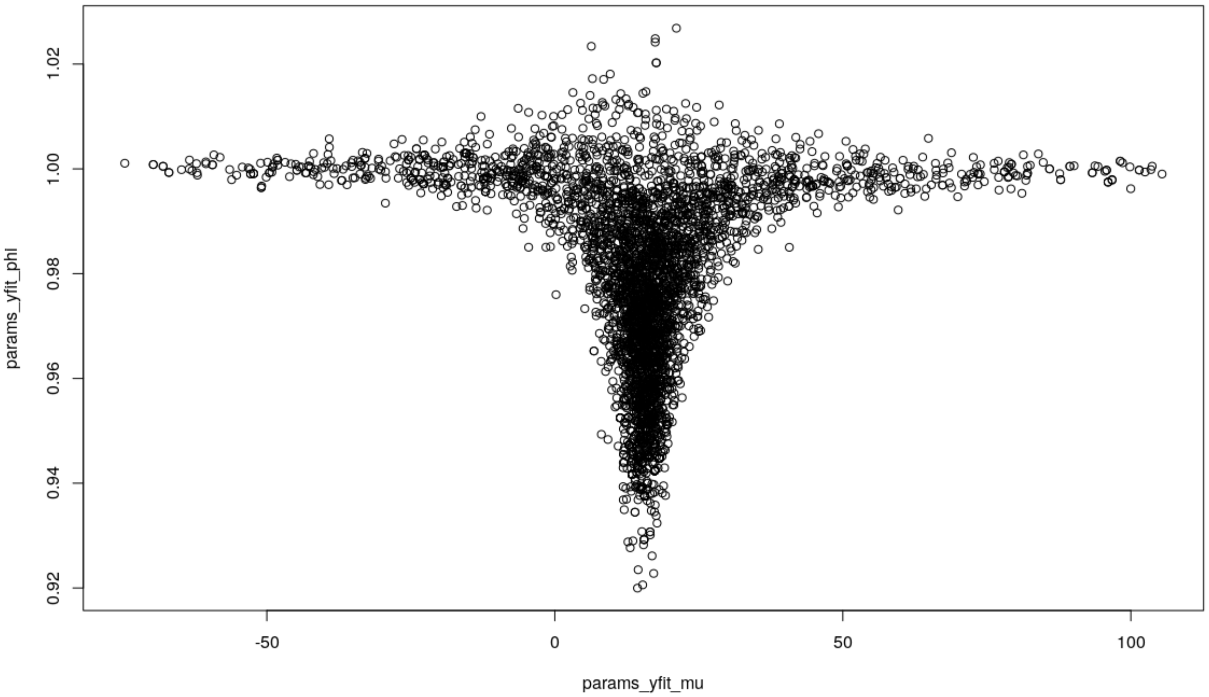


Figure 4: Convergence for of mu and phi for y

The plot above shows the joint posterior of ϕ and μ for data based on $\phi = 0.95$. Here one can see that the sampled values of ϕ is more uniformly distributed between about 1 and 0.9 while the majority of the sampled values for μ are all almost the same value, except a small portion of large outliers.

c)

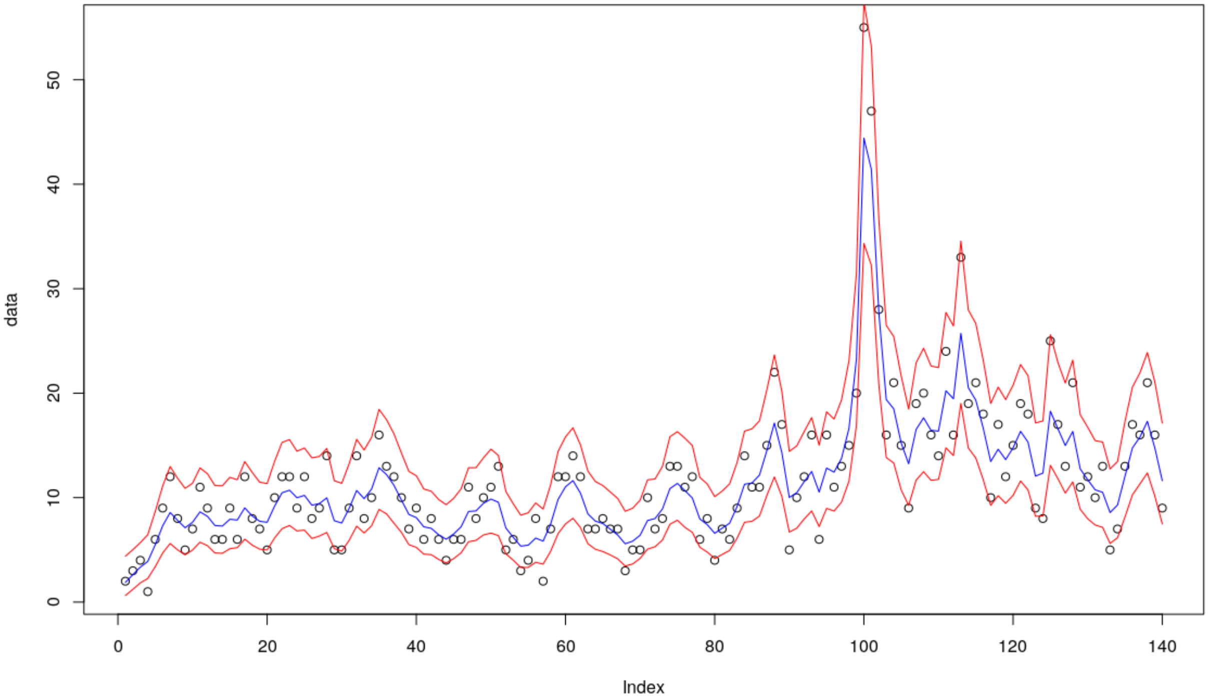


Figure 5: Data points, posterior mean and 95% confidence interval

d)

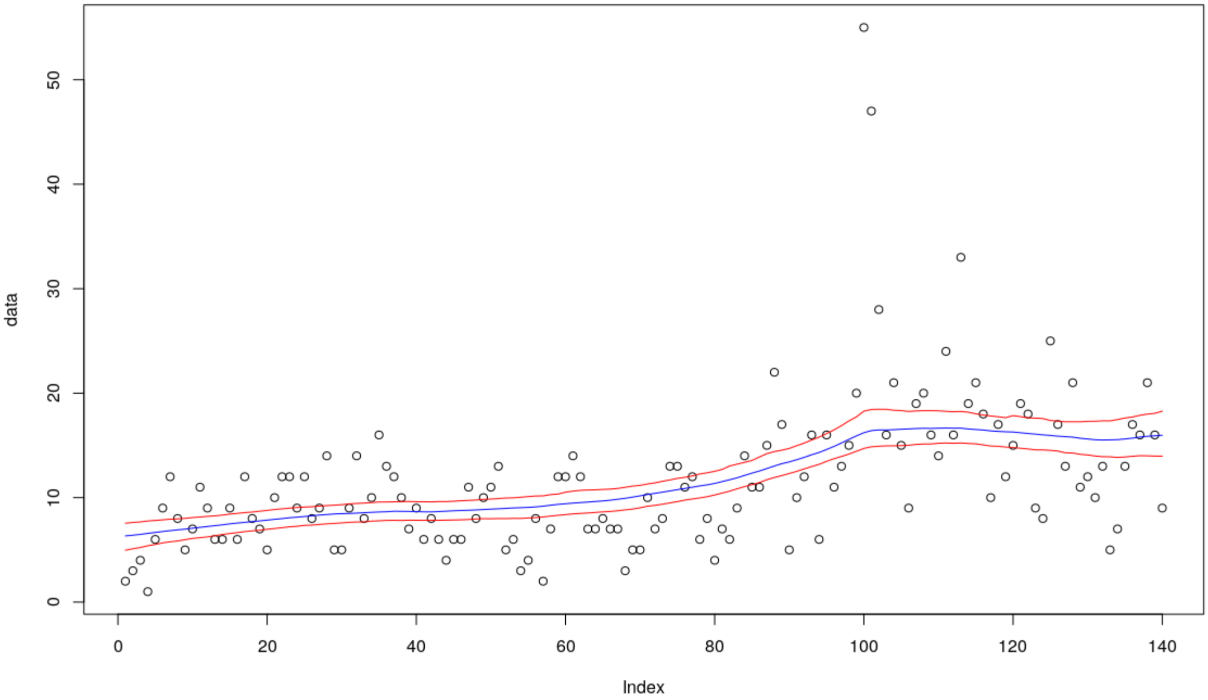


Figure 6: Data points, posterior mean and 95% confidence interval with prior smoothing to simga squared

The posterior has changed as it is smoother and outliers doesn't affect as much as it did before. The $\sigma^2 \sim Scale - inv - \chi^2(N, 0.1)$ where $N = \# \; of \; observations \; in \; the \; data \; set.$

Appendix for code

```r
library(rstan)
options(mc.cores = parallel::detectCores())
rstan_options(auto_write = TRUE)
#a)
T <- 200
mu <- 10
sigma_2 <- 2

phi <- seq(-1,1,0.05)

AR_func <- function(mu, phi, x_prev, e_mu, e_sigma_sq) {
  err <- rnorm(1, e_mu, sqrt(e_sigma_sq))
  print(err)
  #err <- 0
  return(mu + phi * (x_prev - mu) + err)
}


calced_val <- matrix(nrow = T,ncol=length(phi))
xt <- mu
#xt <- rnorm(1,mu,)
for (i in 1:length(phi)) {
  for (t in 1:T) {
    xt <- AR_func(mu,phi[i],xt,0,sigma_2)
    calced_val[t,i] <- xt
  }
  temp_mean <- mean(calced_val[,i])
  plot(c(1:T),calced_val[,i],type="b",main=paste("phi =",phi[i],"mean =",temp_mean),ylim=c(4,16))
  Sys.sleep(0.5)
}


i <- 2
plot(c(1:T),calced_val[,i],type="b",main=paste("phi =",phi[i]))


# b)

## i)

x_t <- mu
y_t <- mu
x <- c()
y <- c()
for (t in 1:T) {
  x_t <- AR_func(mu,0.3,x_t,0,sigma_2)
  y_t <- AR_func(mu,0.95,y_t,0,sigma_2)
  x <- rbind(x,x_t)
  y <- rbind(y,y_t)
}

## Creates a stan model for given dist and function
```

```
StanModel <-
'data {
int<lower=0> N;
vector[N] x;
}
parameters {
real mu;
real phi;
real<lower=0> sigma_2;
}
model {
for (n in 2:N)
  x[n] ~ normal(mu + phi * (x[n-1] - mu), sigma_2);
}'


xfit <- stan(model_code=StanModel, data=list(x=c(x), N=T))
sum_xfit <- summary(xfit)

print(sum_xfit$summary[,"n_eff"][-4])
print(sum_xfit$summary[,"mean"][-4])
print(sum_xfit$summary[,"2.5%"][-4])
print(sum_xfit$summary[,"97.5%"][-4])


yfit <- stan(model_code=StanModel, data=list(x=c(y), N=T))
sum_yfit <- summary(yfit)
print(sum_yfit$summary[,"n_eff"][-4])
print(sum_yfit$summary[,"mean"][-4])
print(sum_yfit$summary[,"2.5%"][-4])
print(sum_yfit$summary[,"97.5%"][-4])

## ii)

params_xfit_mu <- extract(xfit)$mu
params_xfit_phi <- extract(xfit)$phi

plot(params_xfit_mu, params_xfit_phi)

# The plot above shows the joint posterior of $\phi$ and $\mu$ for data based on $\phi=0.3$. As one can
# The middle of the cluster is the mean of each of the paramaters, which can be seen above in i).

params_yfit_mu <- extract(yfit)$mu
params_yfit_phi <- extract(yfit)$phi

plot(params_yfit_mu, params_yfit_phi)

# The plot above shows the joint posterior of $\phi$ and $\mu$ for data based on $\phi=0.95$. Here one
# while the majority of the sampled values for $\mu$ are all almost the same value, except a small port


# c)
```

```
data <- read.delim("campy.dat", header = TRUE, sep = "\n")[, 1]


PoisStanModel <-
'data {
int<lower=0> N;
int c[N];
}
parameters {
real mu;
real phi;
real<lower=0> sigma_2;
vector[N] x;
}
model {


for (n in 2:N)
  x[n] ~ normal(mu + phi * (x[n-1] - mu), sigma_2);

for (n in 1:N)
  c[n] ~ poisson(exp(x[n]));
}'

poisfit <- stan(model_code=PoisStanModel, data=list(c=c(data), N=length(c(data))))
sum_poisfit <- summary(poisfit)$summary[-c(1,2,3,144),]

poisfit_upper <- sum_poisfit[,"97.5%"]
poisfit_lower <- sum_poisfit[,"2.5%"]
poisfit_mean <- sum_poisfit[,"mean"]

plot(data)
lines(exp(poisfit_upper), col="red")
lines(exp(poisfit_lower), col="red")
lines(exp(poisfit_mean), col="blue")

# d)


PoisStanModel2 <-
  'data {
int<lower=0> N;
int c[N];

}
parameters {
real mu;
real phi;
real<lower=0> sigma_2;
vector[N] x;
}
model {
```

```
sigma_2 ~scaled_inv_chi_square(N, 0.1);
for (n in 2:N)
  x[n] ~ normal(mu + phi * (x[n-1] - mu), sigma_2);

for (n in 1:N)
  c[n] ~ poisson(exp(x[n]));
}'

poisfit2 <- stan(model_code=PoisStanModel2, data=list(c=c(data), N=length(c(data))))
sum_poisfit2 <- summary(poisfit2)$summary[-c(1,2,3,144),]



poisfit_upper2 <- sum_poisfit2[,"97.5%"]
poisfit_lower2 <- sum_poisfit2[,"2.5%"]
poisfit_mean2 <- sum_poisfit2[,"mean"]

plot(data)
lines(exp(poisfit_upper2), col="red")
lines(exp(poisfit_lower2), col="red")
lines(exp(poisfit_mean2), col="blue")

# The posterior has changed as it is smoother and outliers doesn't affect as much as it did before.
```