CO2 concentration in Mauna Loa station

Week5-ex3, solution

R-template `ex_linear_regression_MaunaLoaCO2_template.Rmd`.

Data file `maunaloa_data.txt`.

This is an example of linear regression and we will analyse the Mauna Loa CO2 data[1]. The data contains monthly concentrations adjusted to represent the 15th day of each month. Units are parts per million by volume (ppmv) expressed in the 2003A SIO manometric mole fraction scale. The "annual average" is the arithmetic mean of the twelve monthly values where no monthly values are missing.

We want to construct and infer with Stan the following model:

$$
\begin{aligned}
y_i &= \mu(x_i) + \epsilon_i \\
\epsilon_i &\sim N(0, \sigma^2) \\
\mu(x_i) &= a + bx_i \\
p(a) &= p(b) \propto 1 \\
\sigma^2 &\sim \text{Inv-Gamma}(0.001, 0.001)
\end{aligned}
$$

where $y_i, i = 1, \cdots, n$ are the reported CO2 values, x_i is time, measured as months from the first observation, a is an intercept, b is the linear weight (slope) and σ^2 is the variance of the "error" terms, ϵ_i, around the linear mean function.

In practice, it is typically advisable to construct the model for standardized observations $\dot{y}_i = (y_i - \text{mean}(y))/\text{std}(y)$ where $\text{mean}(y))$ and $\text{std}(y)$ are the sample mean and standard deviations of y_i values. Similar transformation should be done also for covariates x. You should then sample from the posterior of the parameters $(\dot{a}, \dot{b}, \dot{\sigma}^2)$ corresponding to the standardized data $\dot{y}_i$ and $\dot{x}_i$. After this you have to transform the samples of $\dot{a}, \dot{b}, \dot{\sigma}^2$ to the original scale.

Your tasks are the following:

1. Solve the equations to transform samples of $\dot{a}, \dot{b}, \dot{\sigma}^2$ to the original scale a, b, σ^2.
2. Sample from the posterior of the parameters of the above model using the Maunaloa CO2 data. (You can do this either with transformed or original data so if you didn't get step 1 right you can still proceed with this.) Check the convergence of model parameters and report the results of convergence tests. Visualize the marginal posterior distribution of the model parameters and report their posterior mean and 2.5% and 97.5% posterior quantiles.
3. Discuss how you would interpret the linear mean function $\mu(x)$ and how you would interpret the error terms ϵ_i.
4. Plot a figure where you visualize

- The posterior mean and 95% central posterior interval of the mean function $\mu(x)$ as a function of months from January 1958 to December 2027.
- The posterior mean and 95% central posterior interval of observations y_i as a function of months from January 1958 to December 2027. In case of historical years, consider the distribution of potential replicate observations that have not been measured but could have been measured.

[1] http://cdiac.esd.ornl.gov/ftp/trends/co2/maunaloa.co2

  - plot also the measured observations to the same figure

5. Visualize
   - the Posterior predictive distribution of the mean function, $\mu(x)$ in January 2025 and in January 1958 and the difference between these.
   - the Posterior predictive distribution of observations, y_i in January 2025 and in January 1958 and the difference between these. * Discuss why the distributions of $\mu(x_i)$ and y_i differ

See the R-template for additional instructions.

Solution

1. variable transformations

Now $y_i|a, b, \sigma^2 \sim N(a + bx_i, \sigma^2)$, $i = 1, \dots, n$. We use standardized observations

$$\dot{y}_i = \frac{y_i - mean(y)}{sd(y)} = \frac{y_i - my}{stdy} \tag{1}$$

and

$$\dot{x}_i = \frac{x_i - mean(x)}{sd(x)} = \frac{x_i - mx}{stdx} = \frac{x_i}{stdx} - \frac{mx}{stdx} \tag{2}$$

Note: $X \sim N(\mu, \sigma^2) \Rightarrow cX + d \sim N(c\mu + d, c^2\sigma^2)$

If we use standardized values $\dot{y}_i$ and $\dot{x}_i$, we have $\dot{y}_i|\dot{a}, \dot{b}, \dot{\sigma}^2 \sim N(\dot{\mu}_i, \dot{\sigma}^2)$, where $\dot{\mu}_i = \dot{a} + \dot{b}\dot{x}_i$. Based on the equation (1) we obtain $y_i = \dot{y}_i \cdot stdy + my \sim N(stdy \cdot \dot{\mu}_i + my, stdy^2 \cdot \dot{\sigma}^2)$.

Hence we see that $\sigma^2 = stdy^2 \cdot \dot{\sigma}^2$.

Furthermore with the help of (2) we obtain

$$\begin{aligned} stdy \cdot \dot{\mu}_i + my &= stdy(\dot{a} + \dot{b}\dot{x}_i) + my \\ &= stdy[\dot{a} + \dot{b}(\frac{x_i}{stdx} - \frac{mx}{stdx})] + my \\ &= \frac{stdy \cdot \dot{b}}{stdx} x_i + stdy \cdot \dot{a} - \frac{\dot{b} \cdot mx \cdot stdy}{stdx} + my \end{aligned}$$

and we get $b = \frac{stdy \cdot \dot{b}}{stdx}$ and $a = stdy \cdot \dot{a} - \frac{\dot{b} \cdot mx \cdot stdy}{stdx} + my$.

2. Build and analyze Stan model

Load the needed libraries.

```
library(ggplot2)
```

```
## Warning: replacing previous import 'vctrs::data_frame' by 'tibble::data_frame'
## when loading 'dplyr'
```

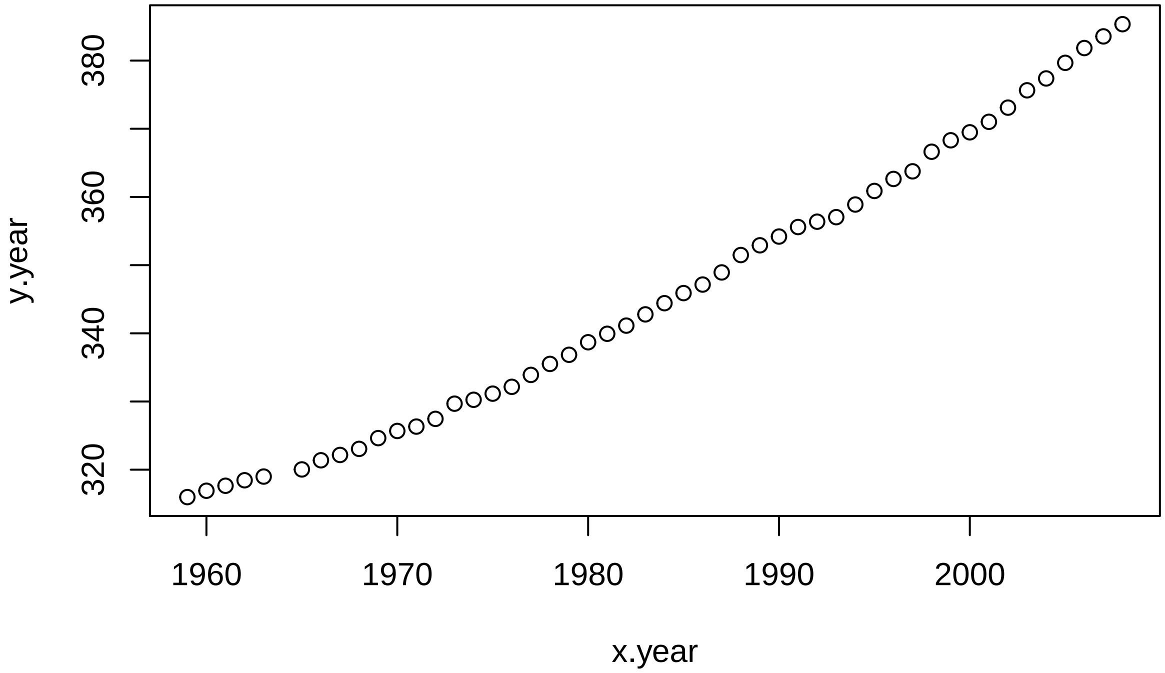
```
library(StanHeaders)
library(rstan)
```

```
## rstan (Version 2.19.3, GitRev: 2e1f913d3ca3)
## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
```

```
set.seed(123)

options(mc.cores = parallel::detectCores())
rstan_options(auto_write = TRUE)
```

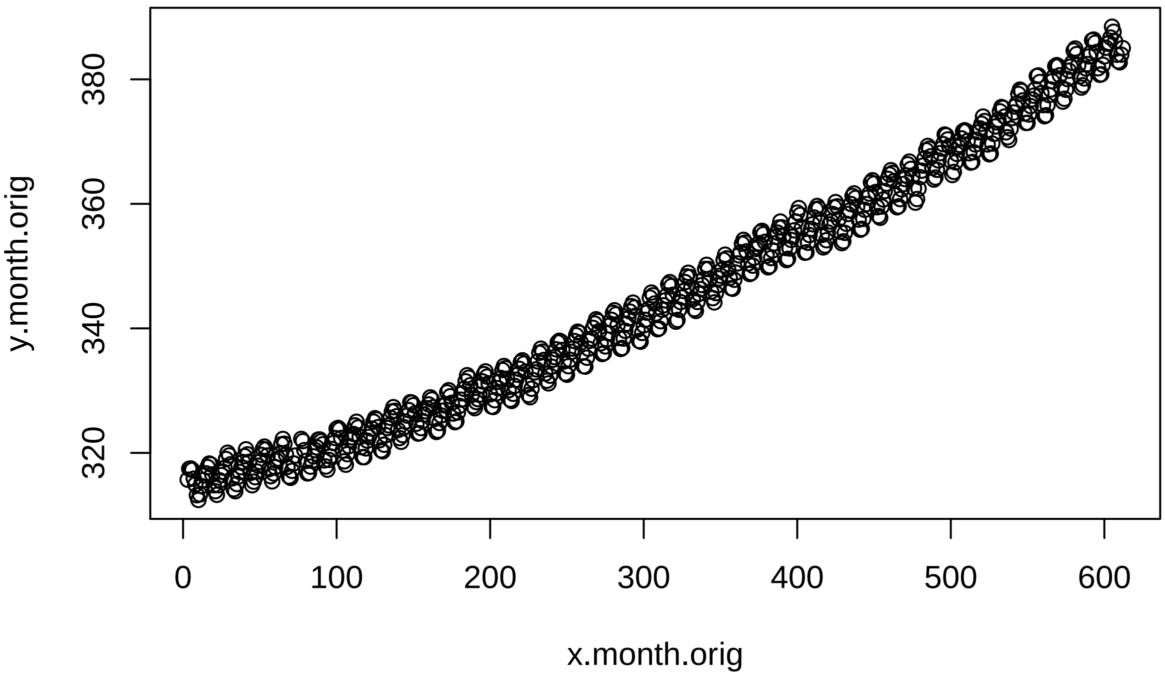
Load the data and explore its properties

```
# Load the data and explore it visually
maunaloa.dat = read.table("maunaloa_data.txt", header=FALSE, sep="\t")
# The columns are
# Year January February ... December Annual average

#  Notice! values -99.99 denote NA

# Let's take the yearly averages and plot them
x.year = as.vector(t(maunaloa.dat[,1]))
y.year = as.vector(t(maunaloa.dat[,14]))
# remove NA rows
x.year = x.year[y.year>0]
y.year = y.year[y.year>0]
plot(x.year,y.year)
```

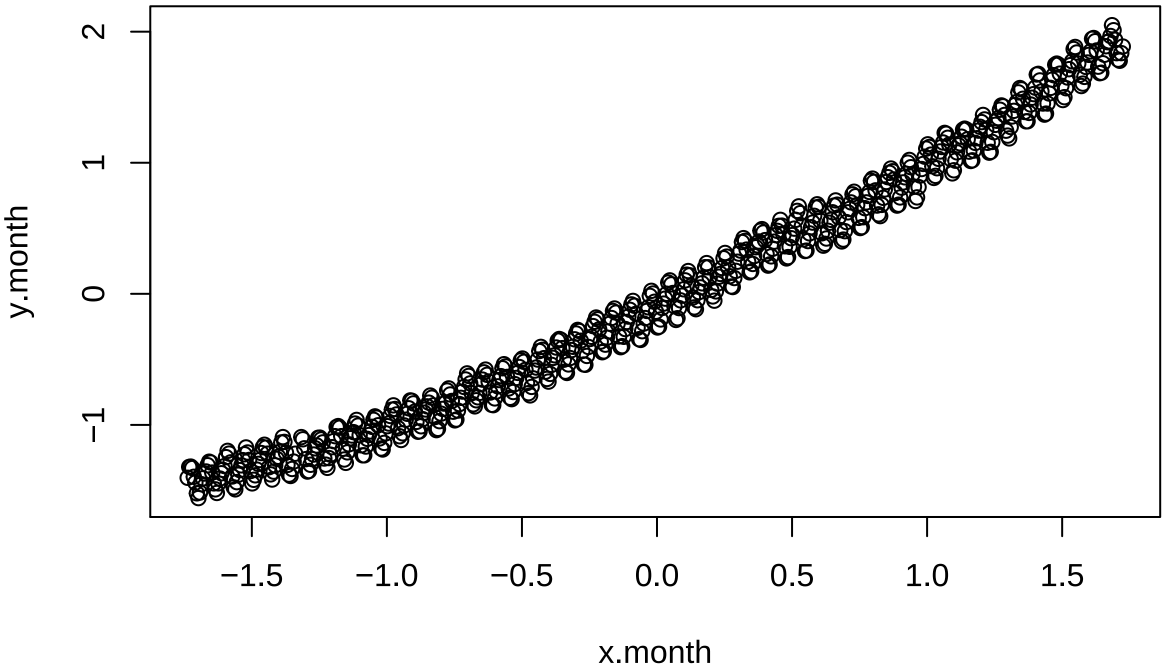


```
# Let's take the monthly values and construct a "running month" vector
y.month.orig = as.vector(t(maunaloa.dat[,2:13]))
x.month.orig = as.vector(seq(1,length(y.month.orig),1))

# remove NA rows
x.month.orig = x.month.orig[y.month.orig>0]
y.month.orig = y.month.orig[y.month.orig>0]
```

```
plot(x.month.orig,y.month.orig)
```



```
# standardize y and x
my = mean(y.month.orig)
stdy = sd(y.month.orig)
y.month = (y.month.orig-my)/stdy

mx = mean(x.month.orig)
stdx = sd(x.month.orig)
x.month = (x.month.orig-mx)/stdx

plot(x.month,y.month)
```



```
# Note, you can make a "crude analysis" of model parameters with the following call to lm
linearMod <- lm(y.month ~ x.month)
print(linearMod)
```

```
##
## Call:
## lm(formula = y.month ~ x.month)
##
## Coefficients:
## (Intercept)      x.month
##  -1.556e-16    9.886e-01
```

Write the model description and set data into list

```
mauna_loa_c02_model = "
data{
  int<lower=0> N; // number of observations
  real y[N];     // observed CO2 values
  real x[N];    // observed times
}
parameters{
  real a;
  real b;
  real<lower=0> sigma2;
}
transformed parameters{
  real<lower=0> sigma;
  real mu[N];

  sigma=sqrt(sigma2);

  for( i in 1 : N ) {
    mu[i] = a + b * x[i];
  }
}
model{
  sigma2 ~ inv_gamma(0.001,0.001);

  for( i in 1 : N ) {
    y[i] ~ normal(mu[i],sigma);
  }
}"

# data list
data <- list (N=length(x.month), y=y.month, x=x.month)
#if you want to see what you get with the original data (not transformed)
#data <- list (N=length(x.month.orig), y=y.month.orig, x=x.month.orig)
```

Now we will start the analysis. Define parameters and set initial values for them. We are going to sample four chains so we need four starting points. It is good practice to set them far apart from each others. We build linear regression model on data in order to get some reasonable initial values for our model parameters. Examine the convergence.

```
# Initial values
init1 <- list (a = 5, b = -10, sigma2 = 0.1)
init2 <- list (a = 0, b = -5, sigma2 = 0.2)
init3 <- list (a = -5, b = 5, sigma2 = 0.3)
init4 <- list (a = 10, b = 10, sigma2 = 0.4)
inits <- list(init1, init2, init3,init4)

## Run the Markov chain sampling with Stan:
set.seed(123)
post=stan(model_code=mauna_loa_c02_model,data=data,warmup=500,iter=2000,chains=4,thin=1,init=inits,cont
```

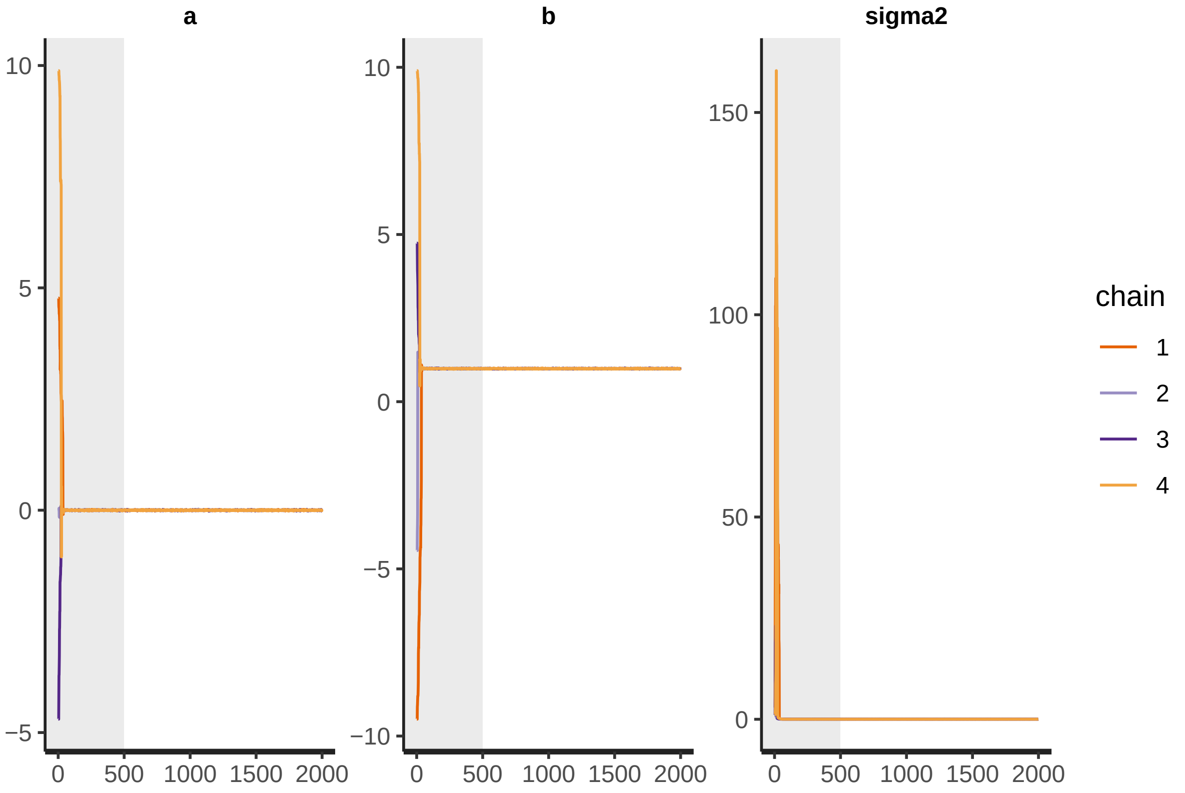
```
## Trying to compile a simple C file
```

```
## Running /usr/lib/R/bin/R CMD SHLIB foo.c
## gcc -std=gnu99 -I"/usr/share/R/include" -DNDEBUG   -I"/home/local/jpvanhat/R/x86_64-pc-linux-gnu-lib
## In file included from /home/local/jpvanhat/R/x86_64-pc-linux-gnu-library/3.6/RcppEigen/include/Eigen
##                  from /home/local/jpvanhat/R/x86_64-pc-linux-gnu-library/3.6/RcppEigen/include/Eigen
##                  from /home/local/jpvanhat/R/x86_64-pc-linux-gnu-library/3.6/StanHeaders/include/sta
##                  from <command-line>:0:
## /home/local/jpvanhat/R/x86_64-pc-linux-gnu-library/3.6/RcppEigen/include/Eigen/src/Core/util/Macros.
##  namespace Eigen {
##  ^~~~~~~~~
## /home/local/jpvanhat/R/x86_64-pc-linux-gnu-library/3.6/RcppEigen/include/Eigen/src/Core/util/Macros.
##  namespace Eigen {
##                  ^
## In file included from /home/local/jpvanhat/R/x86_64-pc-linux-gnu-library/3.6/RcppEigen/include/Eigen
##                  from /home/local/jpvanhat/R/x86_64-pc-linux-gnu-library/3.6/StanHeaders/include/sta
##                  from <command-line>:0:
## /home/local/jpvanhat/R/x86_64-pc-linux-gnu-library/3.6/RcppEigen/include/Eigen/Core:96:10: fatal err
##  #include <complex>
##           ^~~~~~~~~
## compilation terminated.
## /usr/lib/R/etc/Makeconf:168: recipe for target 'foo.o' failed
## make: *** [foo.o] Error 1
```

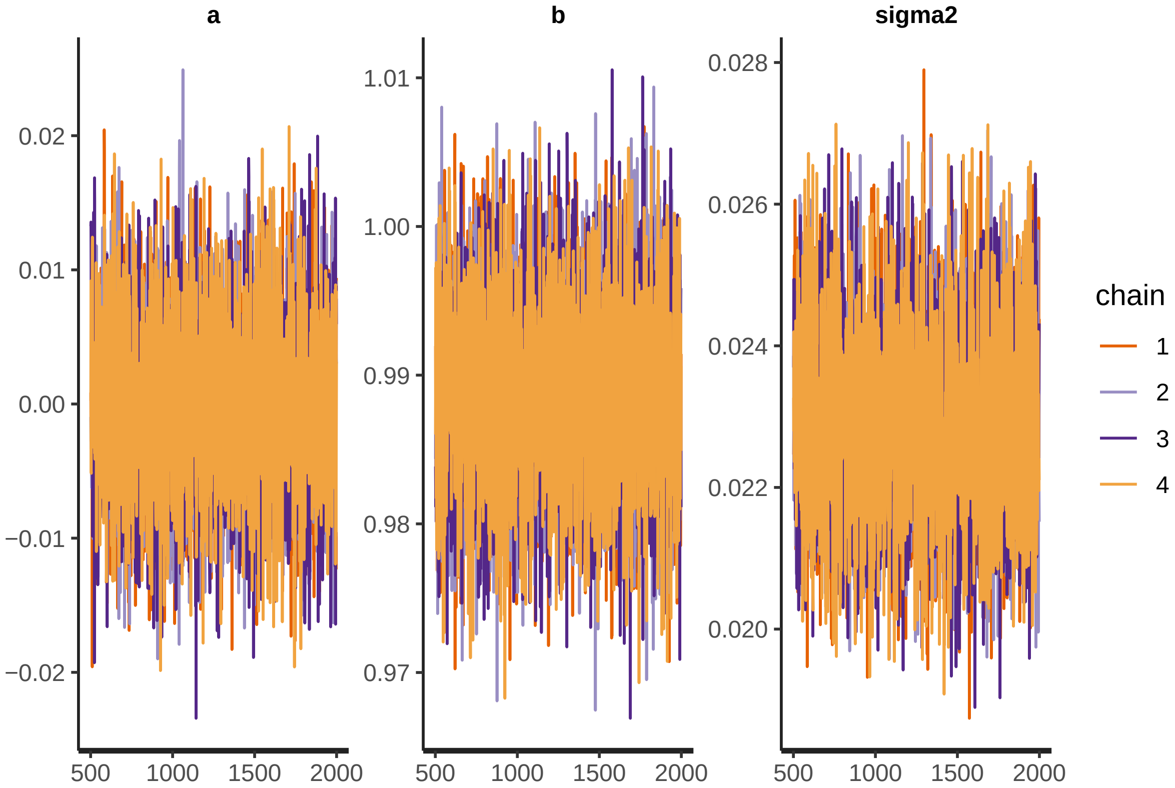
```r
# Check for convergence, see PSRF (Rhat in Stan)
print(post,pars=c("a","b","sigma2"))
```

```
## Inference for Stan model: 237139d61e51ca610fd34064e89729dd.
## 4 chains, each with iter=2000; warmup=500; thin=1;
## post-warmup draws per chain=1500, total post-warmup draws=6000.
##
##        mean se_mean   sd  2.5%  25%  50%  75% 97.5% n_eff Rhat
## a      0.00       0 0.01 -0.01 0.00 0.00 0.00  0.01  6879    1
## b      0.99       0 0.01  0.98 0.98 0.99 0.99  1.00  6828    1
## sigma2 0.02       0 0.00  0.02 0.02 0.02 0.02  0.03  3834    1
##
## Samples were drawn using NUTS(diag_e) at Wed Dec  9 07:45:17 2020.
## 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).
```

```r
#print(post)
plot(post, pars=c("a","b","sigma2"),plotfun= "trace", inc_warmup = TRUE)
```



```r
plot(post, pars=c("a","b","sigma2"), plotfun= "trace", inc_warmup = FALSE)
```



```
#plot(post, pars=c("mu"), plotfun= "trace", inc_warmup = FALSE)
```

Next we visualize and summarize parameter posteriors in original scale. Extract the posterior samples as a matrix. NOTE THAT ABOVE YOU OBTAINED THE PARAMETERS $\dot{a}$, $\dot{b}$ and $\dot{\sigma}^2$. You should continue with the original parameters a, b and σ^2 from now on. So make the needed transformations.

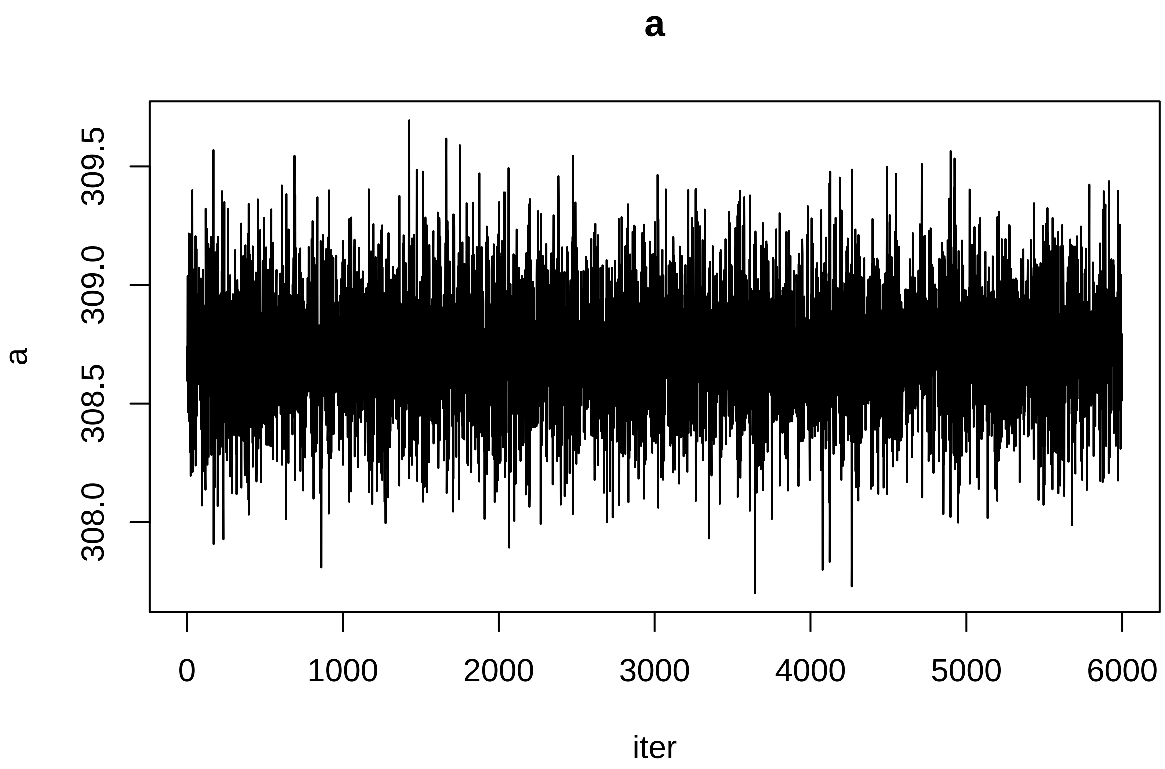
```
post_sample=as.matrix(post, pars =c("a","b","sigma2"))
dim(post_sample)
```

```
## [1] 6000    3
```

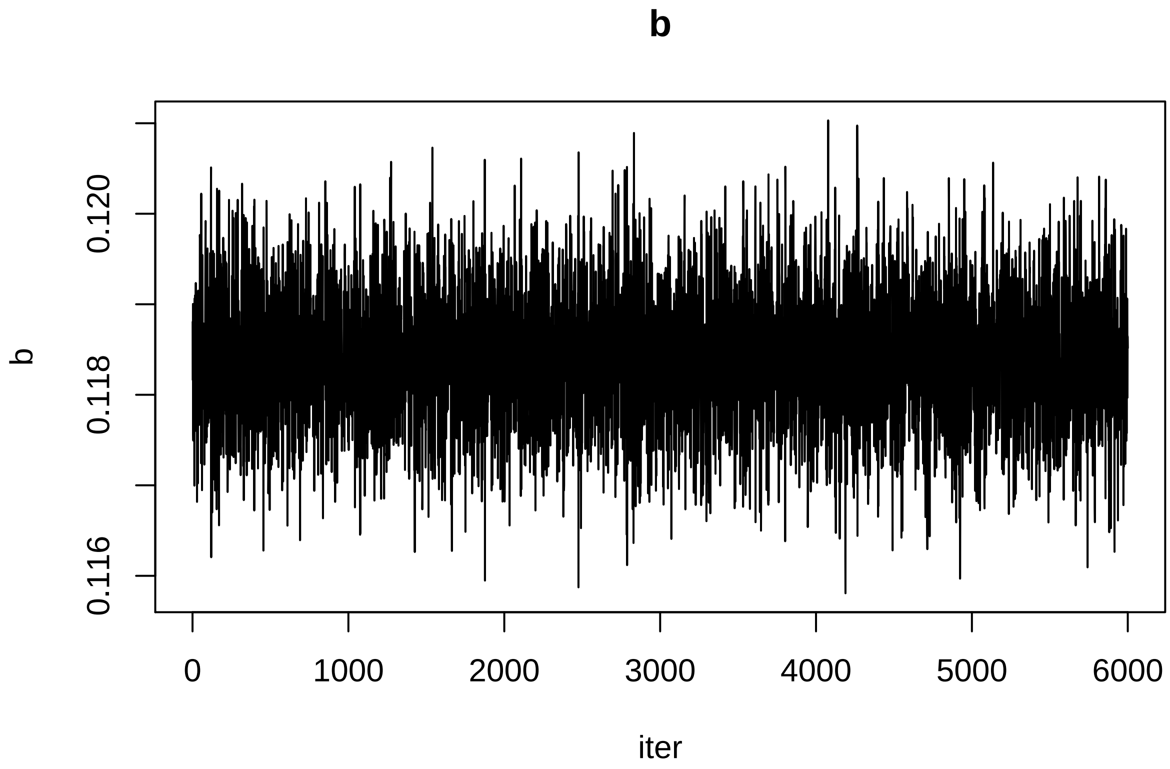
```
#one column contains posterior samples for one variable
a_dot=post_sample[,1]
b_dot=post_sample[,2]
sigma2_dot=post_sample[,3]

a=a_dot*stdy - b_dot*mx*stdy/stdx + my
b=b_dot*stdy/stdx
sigma2=stdy^2*sigma2_dot
post_samples_orig = matrix(c(a,b,sigma2), nrow = 6000, byrow = F)
colnames(post_samples_orig)= c('a','b','sigma2')
# write.table(post_samples_orig, file="param.txt", row.names=FALSE, col.names=TRUE)
```

Now parameter a contains a sample from the posterior $p(a|y, x, n)$ and parameter b contains sample from the posterior $p(b|y, x, n)$. We can now plot sample chains and histograms of them and do the required summaries.

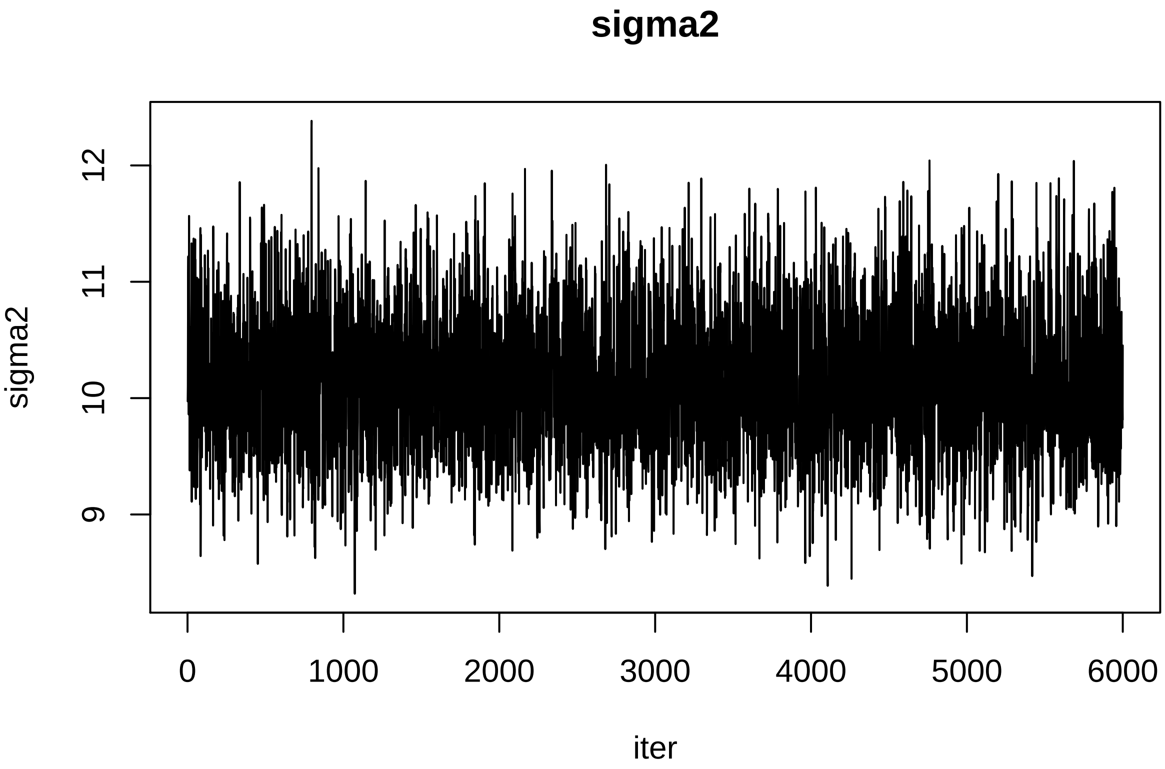
```
#Trace plot of MCMC output to see if the chains have converged for the original parameters
plot(a, main="a", xlab="iter",type="l")
```



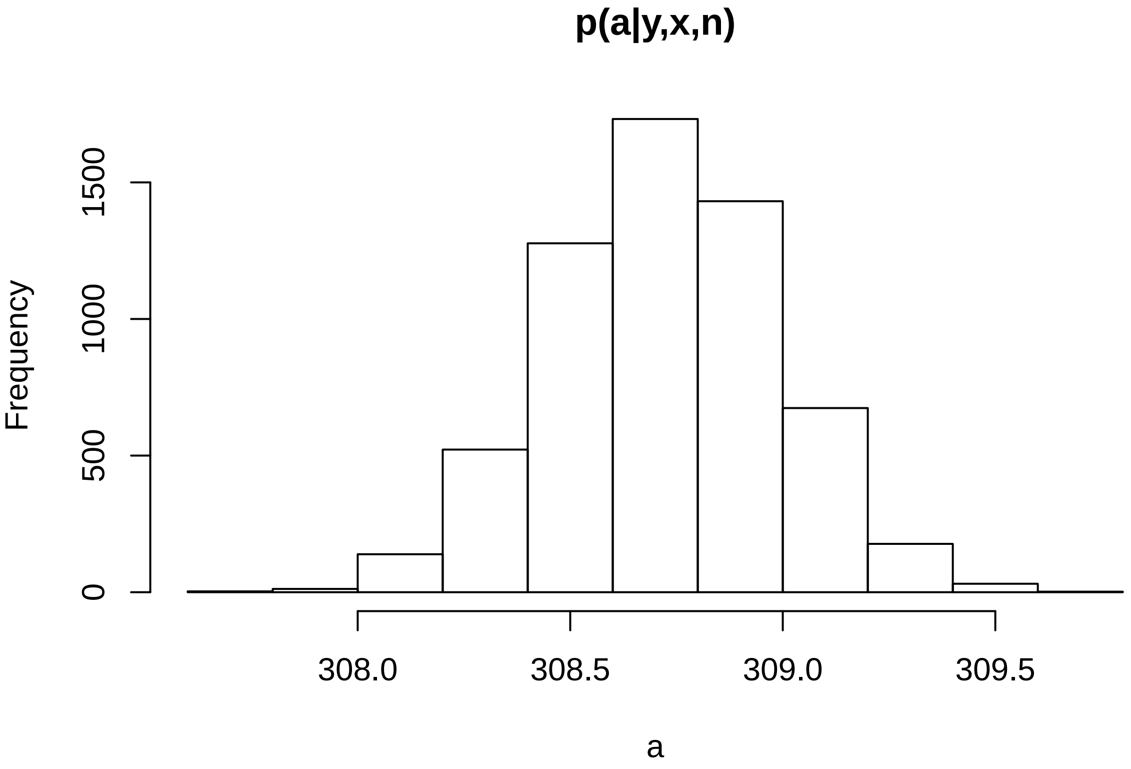
```r
plot(b, main="b", xlab="iter",type="l")
```



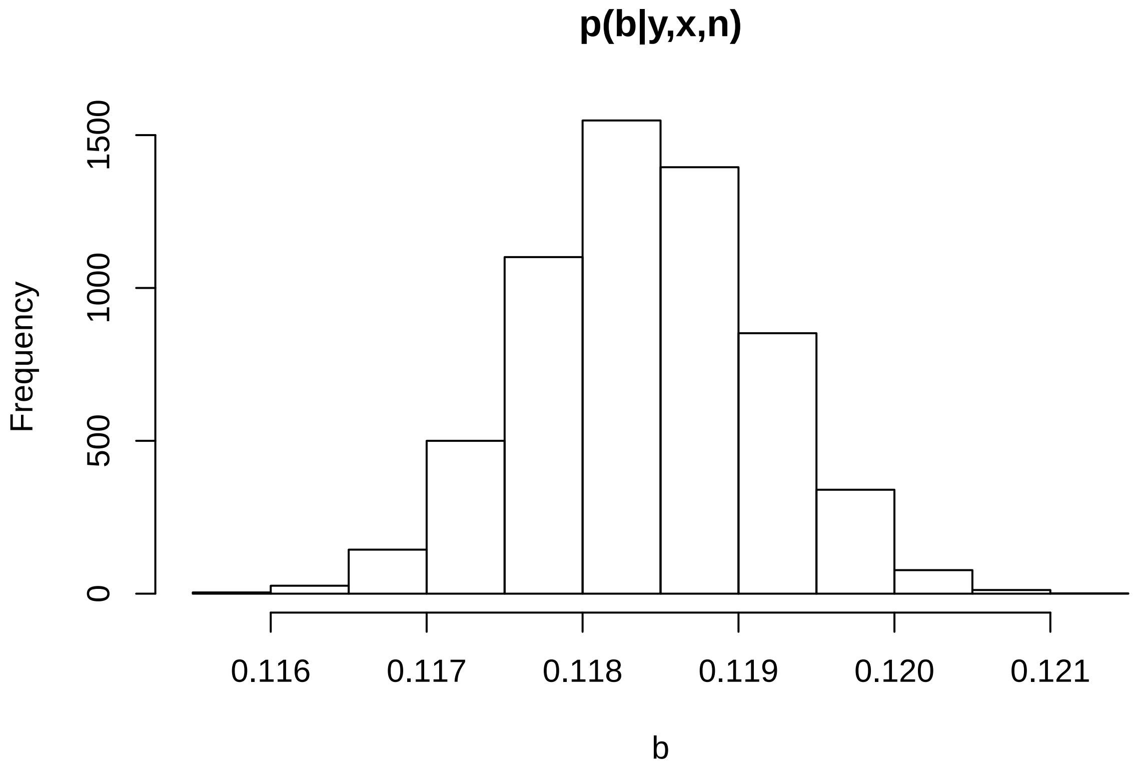
```r
plot(sigma2, main="sigma2", xlab="iter",type="l")
```



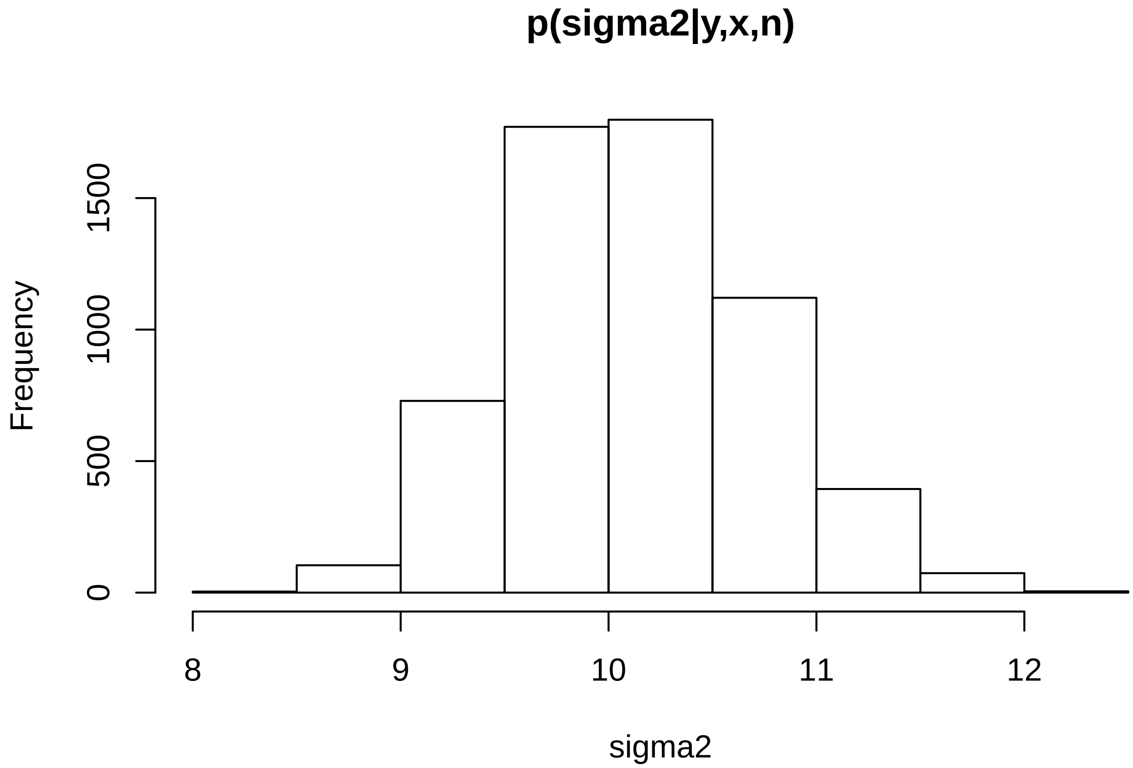
```
#Note, if the chains do not look converged see what is the problem and rerun the model

hist(a, main="p(a|y,x,n)", xlab="a")
```



```r
hist(b, main="p(b|y,x,n)", xlab="b")
```



```r
hist(sigma2, main="p(sigma2|y,x,n)", xlab="sigma2")
```



```
c(mean(a), quantile(a, c(0.025, 0.975)))
```

```
##                2.5%    97.5%
## 308.7220 308.1955 309.2411
```

```
c(mean(b), quantile(b, c(0.025, 0.975)))
```

```
##                 2.5%     97.5%
## 0.1184044 0.1169552 0.1198669
```

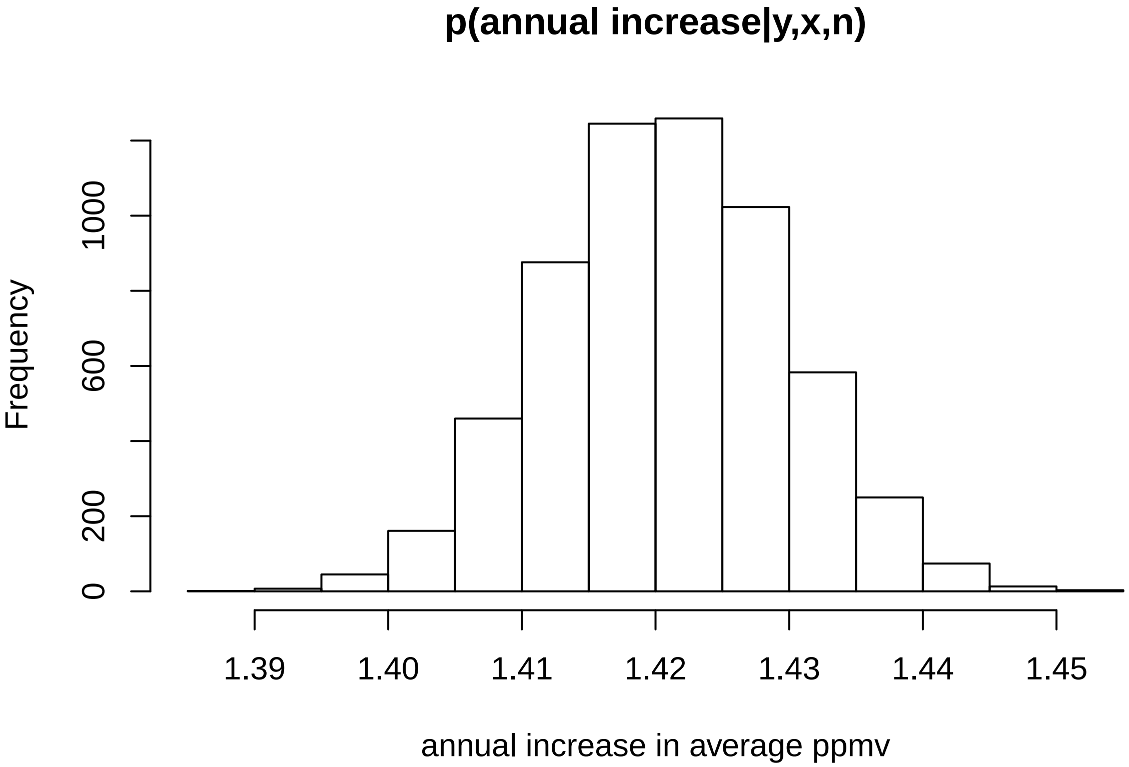
```
c(mean(sigma2), quantile(sigma2, c(0.025, 0.975)))
```

```
##                 2.5%     97.5%
## 10.134447  9.071816 11.354006
```

3. Interpretation of $\mu(x)$ and ϵ_i

The mean function $\mu(x)$ descibes the long term trend in monthly mean CO2 concentrations. For example, we may ask how many ppmvs the CO2 concentration of any month increases per year. The histogram of the posterior distribution of this increase is

```
hist(b*12, main="p(annual increase|y,x,n)", xlab="annual increase in average ppmv")
```

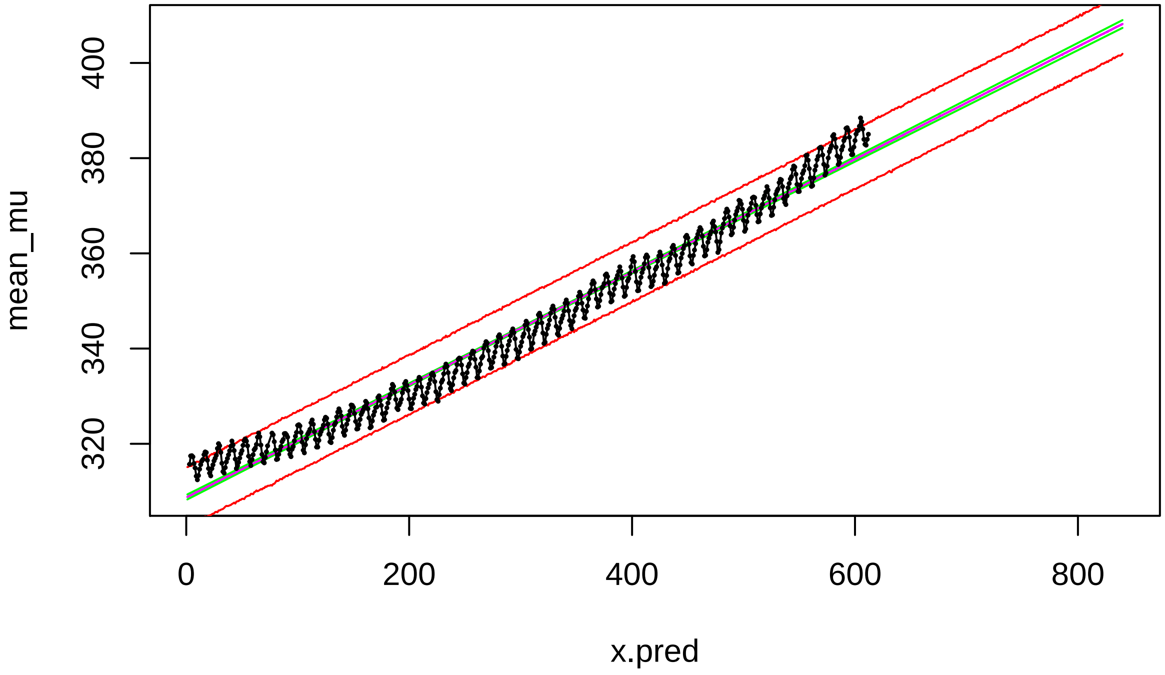


The error term ϵ_i describes monthly variations around this mean as well as possible measurement errors in the measured CO2 concentrations.

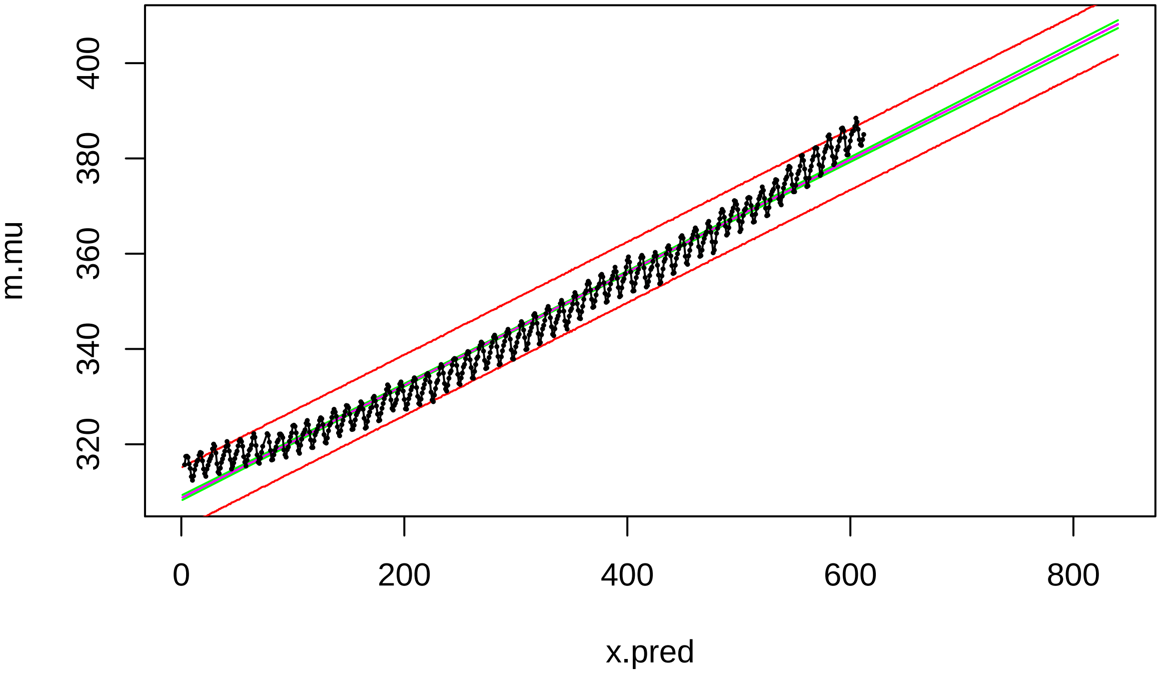
4. visualization of the regression curve

Data covers years from 1958 to 2008. Therefore, we need to construct prediction points and predict the historical and future next 20 years of CO2 concentrations

```
set.seed(123)
x.pred= seq(1,70*12,length=70*12)

mu = matrix(NA,length(x.pred),length(b))      # matrix of posterior samples of mu
y.tilde = matrix(NA,length(x.pred),length(b)) # matrix of posterior samples of y.tilde

mean_mu=rep(NA, length(x.pred))              # posterior mean of mu
int_mu = matrix(NA,length(x.pred),2)         # posterior 95% interval of mu

mean_y=rep(NA, length(x.pred))              # posterior mean of y.tilde
int_y = matrix(NA,length(x.pred),2)         # posterior 95% interval of y.tilde

for (i in 1:length(x.pred)) {
  #mu[i,] = (a + b*x.pred[i])*stdy + my
    mu[i,] = a + b*x.pred[i]
  mean_mu[i]=mean(mu[i,])
  int_mu[i,] = quantile(mu[i,],probs=c(0.025,0.975))
  #y_i = mu_i + e_i and e_i ~ N(0,sigma2)
  y.tilde[i,] =  mu[i,] + rnorm(length(mu[i,]), 0, sqrt(sigma2))
  mean_y[i]=mean(y.tilde[i,])
  int_y[i,] = quantile(y.tilde[i,],probs=c(0.025,0.975))

}

plot(x.pred,mean_mu, type="l",col="blue") #posterior mean for mu(x)
lines(x.pred,int_mu[,1],col="green")
lines(x.pred,int_mu[,2],col="green") # 95% interval of mu(x)
lines(x.pred,mean_y, type="l",col="magenta") #posterior mean for y.tilde
lines(x.pred,int_y[,1],col="red")
lines(x.pred,int_y[,2],col="red") # 95% interval of y.tilde
lines(x.month.orig,y.month.orig)
points(x.month.orig,y.month.orig, cex=0.2)
```



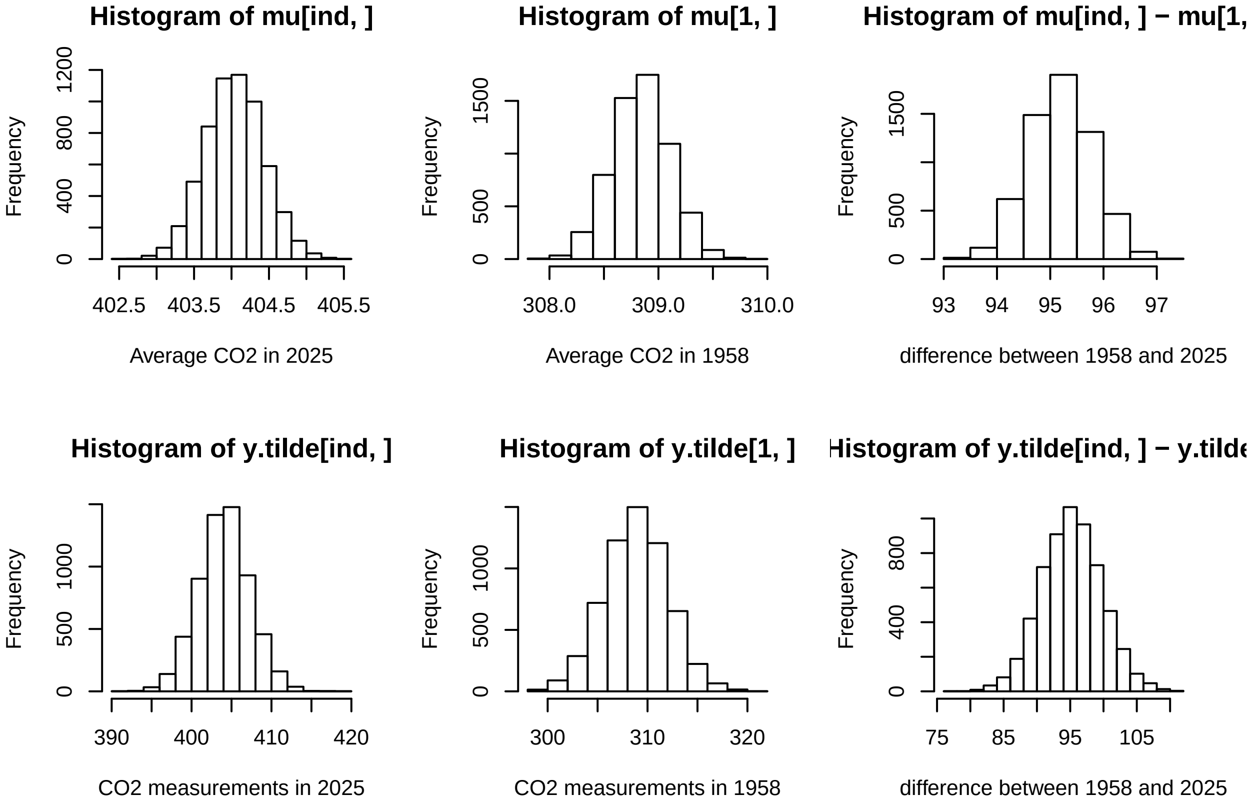
```r
#or you use the information that about 95% of the mass of the normal distribution is
#within mu +- 2sigma
sds.mu = apply(mu,1,sd)
sds.y = apply(y.tilde,1,sd)
m.mu = rowMeans(mu)
m.y = rowMeans(y.tilde)
plot(x.pred,m.mu, type="l",col="blue") #posterior mean for mu(x)
lines(x.pred,m.mu+2*sds.mu, col="green")   # 95% interval of f
lines(x.pred,m.mu-2*sds.mu, col="green")
lines(x.pred,m.y, type="l",col="magenta") #posterior mean for y.tilde
lines(x.pred,m.mu+2*sds.y, col="red")     # 95% interval of y
lines(x.pred,m.mu-2*sds.y, col="red")
lines(x.month.orig,y.month.orig)
points(x.month.orig,y.month.orig, cex=0.2)
```



5. CO2 concentration in January 2025 and 1958

Posterior predictive distribution of the mean function in January 2025, January 1958 and the difference between these. Notice! x=1 corresponds to January 1958

```r
ind = (2025-1958)*12+1
par(mfrow=c(2,3))           # divide plot into 6 subplots
hist(mu[ind,], xlab="Average CO2 in 2025")
hist(mu[1,], xlab="Average CO2 in 1958")
hist(mu[ind,]-mu[1,], xlab="difference between 1958 and 2025")

# Posterior predictive distribution of CO2 measurements  in January 2025, January 1958
# and the difference between these.
hist(y.tilde[ind,], xlab="CO2 measurements in 2025")
hist(y.tilde[1,], xlab="CO2 measurements in 1958")
hist(y.tilde[ind,]-y.tilde[1,], xlab="difference between 1958 and 2025")
```



```
x.pred[ind]
```

```
## [1] 805
```

Grading

Total points 20. Each question gives 4 points as follows: 2 point for a solution that is towards the correct direction. 2 points more if the solution is correct. Note, in questions that require discussion, you should not give full points if the discussion is missing or if it is not adequate.