Untitled

Wuhao Wang(wuhwa469)

4/26/2022

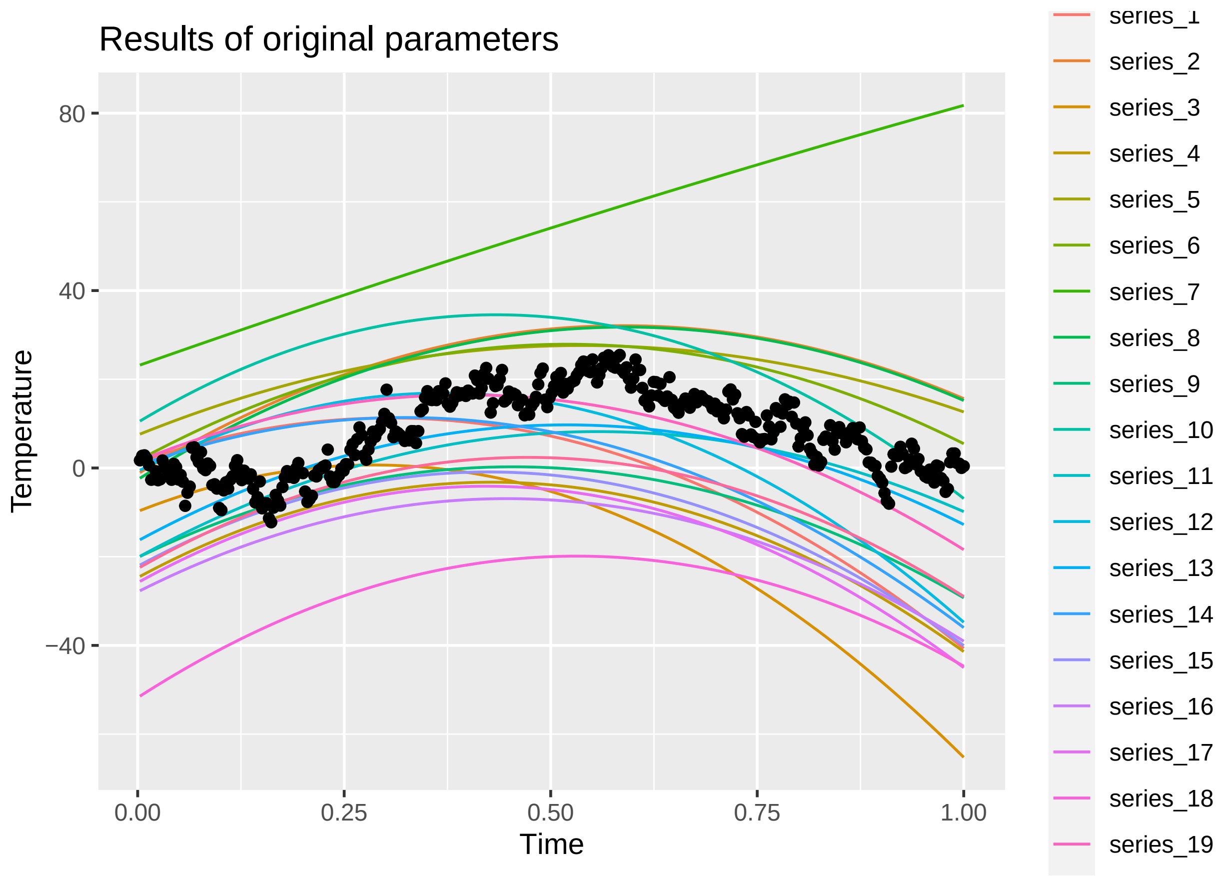
```
library(ggplot2)
library(LaplacesDemon)
library(reshape2)
library(mvtnorm)
library(gridExtra)
```

Question 1 Liner and polynomial regression

a)

```
tempdata <- read.table("TempLambohov.txt",header = TRUE)
y <- tempdata$temp
# x = (beta0,beta1,beta2)
X <- cbind(1, tempdata$time, tempdata$time**2)
n_obs <- nrow(X)

mu0 <- c(-10, 100, -100)
omega0<- 0.02*diag(3)
v0 <- 3
sigma0 <- 2

N <- 20
coe_prior <- matrix(ncol = 3, nrow = N)
for (i in 1:N) {
  #from L5 slides
  sigma  <- LaplacesDemon::rinvchisq(1 ,v0, sigma0)
  beta <- MASS::mvrnorm(1, mu0, sigma*solve(omega0))

  coe_prior[i,1:3] <- beta
}
df <- as.data.frame(cbind(tempdata$time, X%*%t(coe_prior)))
cnames <- c("x")
for (i in 1:N) {
  cnames[1+i] <- paste0("series_",i)
}
colnames(df) <- cnames
df <- melt(df, id.vars = "x")
p1a <- ggplot(df)+
  geom_line(aes(x = x, y = value, color = variable)) +
```

```
  geom_point(data = tempdata, aes(x = time, y = temp)) +
  ggtitle("Results of original parameters") + ylab("Temperature") + xlab("Time")
p1a
```

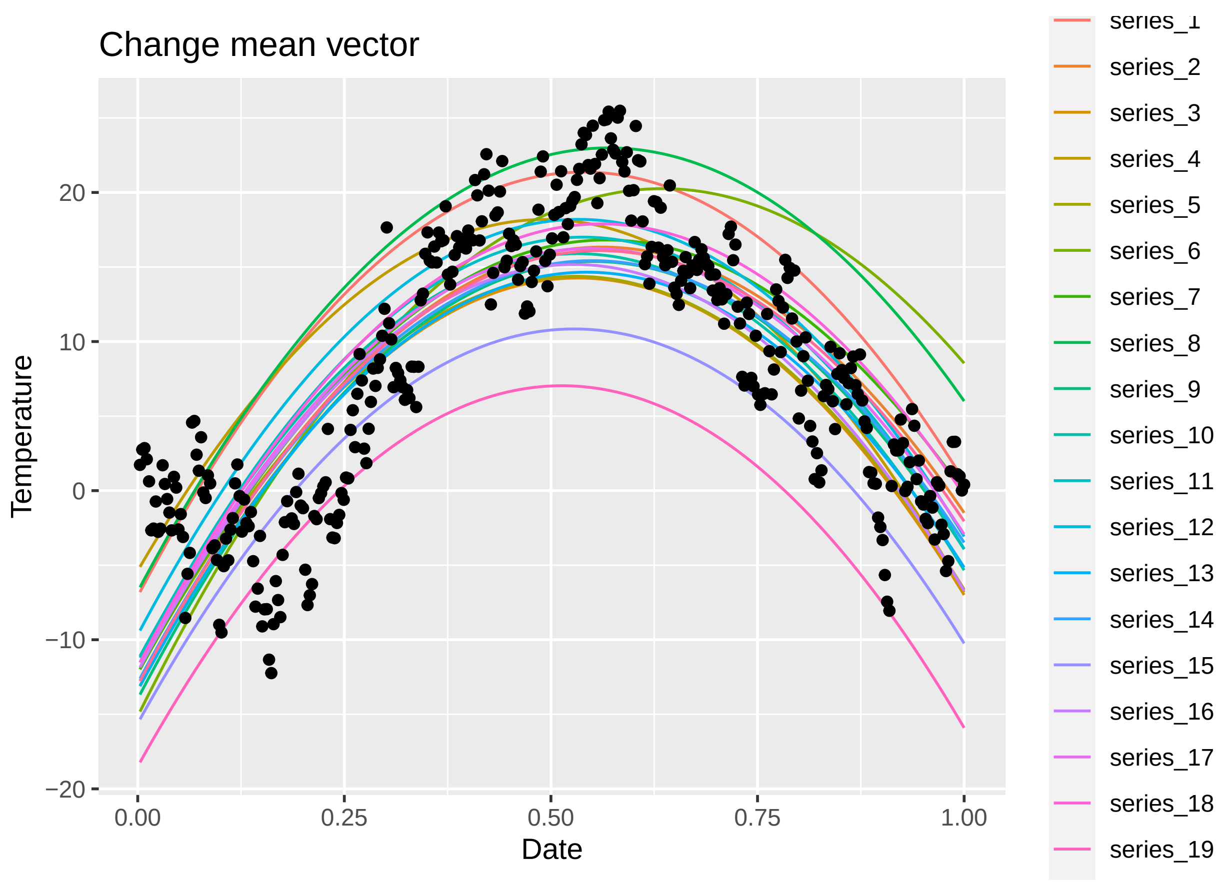


From the picture above, I can see that the simulations can not match the observations. So there are some problems with original parameters. Consider that this model is a liner model, so I can use `lm()` function to find optimal estimated coefficient of liner model.

By applying `lm()` , I find that the estimated parameters for ($\beta_0, \beta_1, \beta_2$) are (-11.927,103.418,-95.207). So I can apply this vector to μ_0 . Besides, I can also see that the draws are quite separate, so maybe turn σ_0 to smaller value would help. I tried different value of σ_0, if σ_0 is too small, the curve can not match some top and bottom value, finally I set σ_0 to 0.1. And when it comes to ω_0 ,changing the value of it will work reversly on curve(bigger number cause narrower curve), and finally we set it to 0.025*`I` .

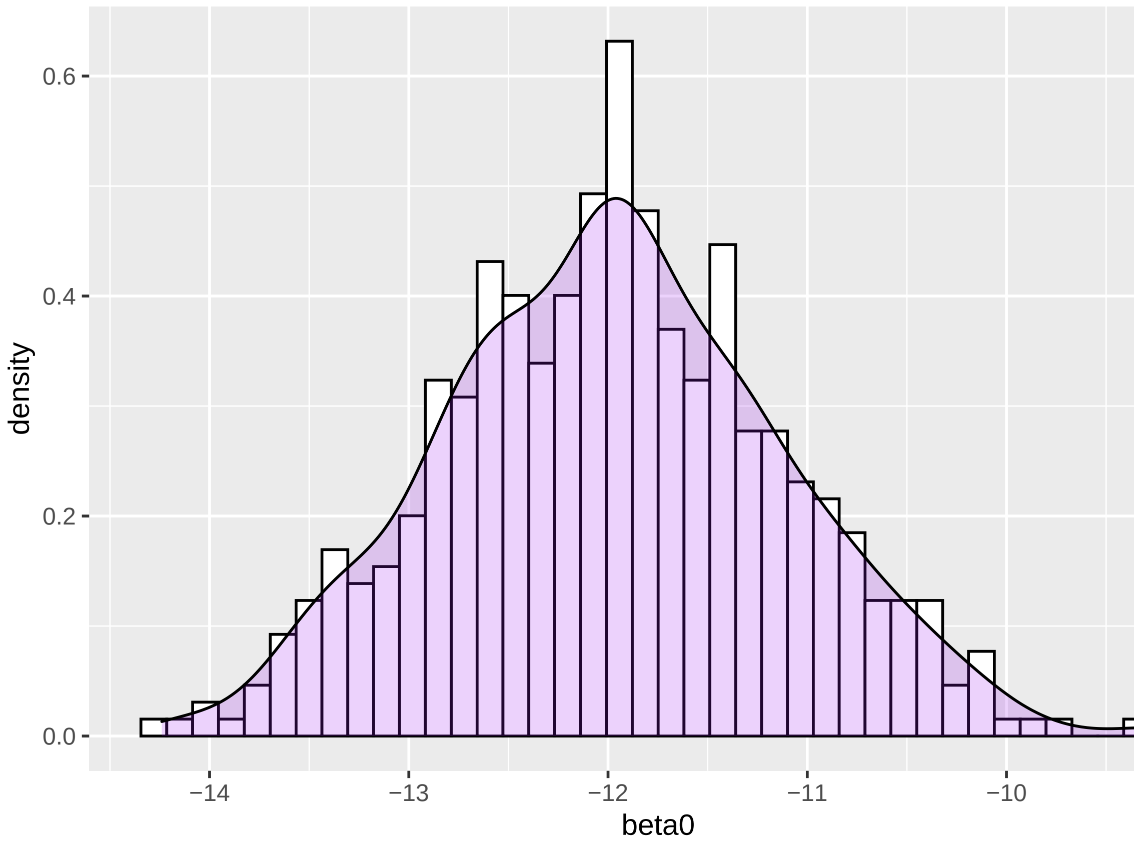
```
tempdata <- read.table("TempLambohov.txt",header = TRUE)
y <- tempdata$temp
# x = (beta0,beta1,beta2)
X <- cbind(1, tempdata$time, tempdata$time**2)
n_obs <- nrow(X)

mu0 <- c(-11.927,103.418,-95.207)
omega0<- 0.025*diag(3)
v0 <- 3
sigma0 <- 0.1

N <- 20
```

```
coe_prior <- matrix(ncol = 3, nrow = N)
for (i in 1:N) {
  #from L5 slides
  sigma  <- LaplacesDemon::rinvchisq(1 ,v0, sigma0)
  beta <- MASS::mvrnorm(1, mu0, sigma*solve(omega0))

  coe_prior[i,1:3] <- beta
}
df <- as.data.frame(cbind(tempdata$time, X%*%t(coe_prior)))
cnames <- c("x")
for (i in 1:N) {
  cnames[1+i] <- paste0("series_",i)
}
colnames(df) <- cnames
df <- melt(df, id.vars = "x")
p1a <- ggplot(df)+
  geom_line(aes(x = x, y = value, color = variable)) +
  geom_point(data = tempdata, aes(x = time, y = temp)) +
  ggtitle("Change mean vector") + ylab("Temperature") + xlab("Date")
p1a
```



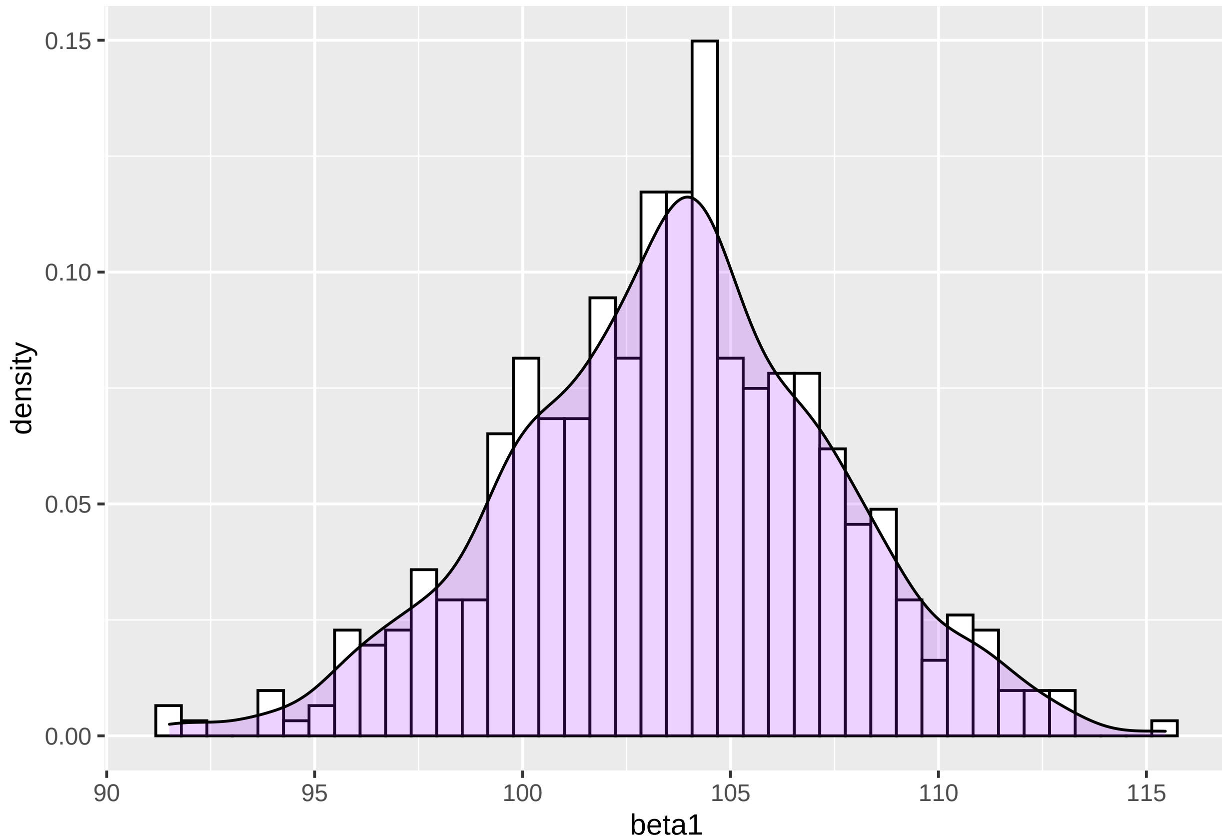
b)

```
k <- 3 #  nr of regression coefficients
beta_hat <- solve(t(X)%*%X)%*%t(X)%*%y
mu_n <- solve(t(X)%*%X+omega0)%*%(t(X)%*%X%*%beta_hat+omega0%*%mu0)
omega_n  <- t(X) %*% X+omega0
v_n <- v0+n_obs
sigma_n <- (v0*sigma0+(t(y)%*%y+t(mu0)%*%omega0%*%mu0-t(mu_n)%*%omega_n%*% mu_n))/v_n

df1 <- as.data.frame(
  mvtnorm::rmvt(n = 500, delta = mu_n, df = n_obs-k,
                sigma = as.numeric(sigma_n) * solve(t(X) %*% X))
  )

df2 <- LaplacesDemon::rinvchisq(n = 1000, v_n, sigma_n)
df <- cbind(df1, df2)
cnames <- c("beta0", "beta1", "beta2", "sigma")
colnames(df) <- cnames
ggplot(df,aes(beta0)) +
    geom_histogram(aes(y = ..density..),
                   colour = "black",
                   fill   = "white",
                   bins   = 40) +
    geom_density(alpha = .2, fill = "purple")
```

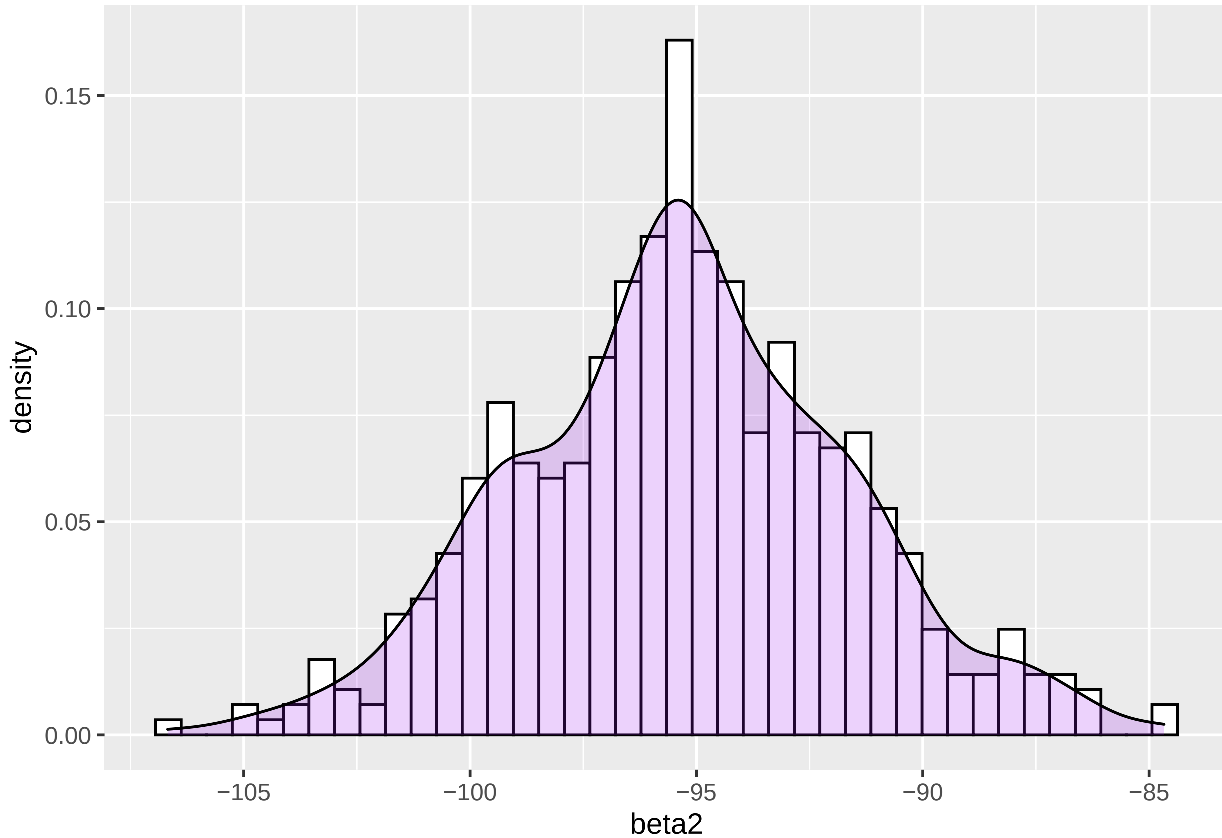


i. marginal posterior

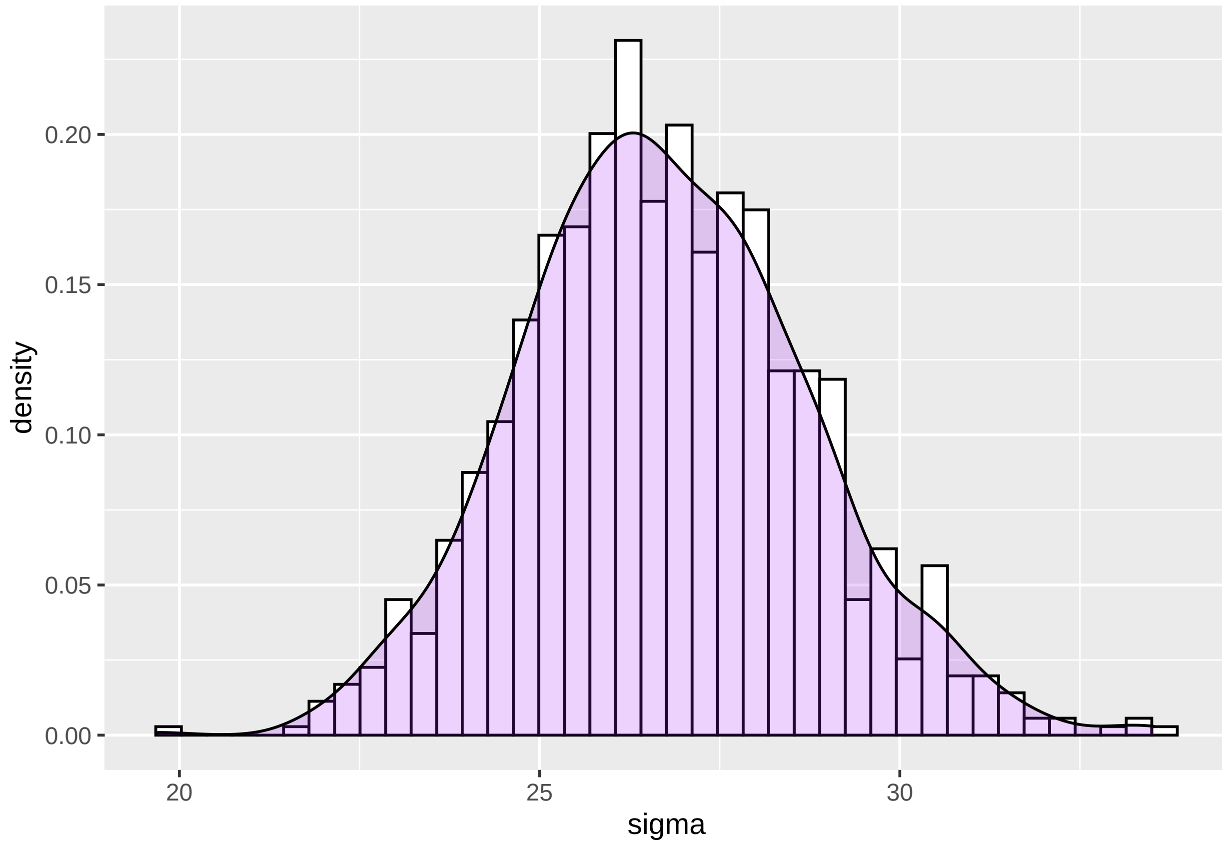
```r
ggplot(df,aes(beta1)) +
    geom_histogram(aes(y = ..density..),
                   colour = "black",
                   fill   = "white",
                   bins   = 40) +
    geom_density(alpha = .2, fill = "purple")
```



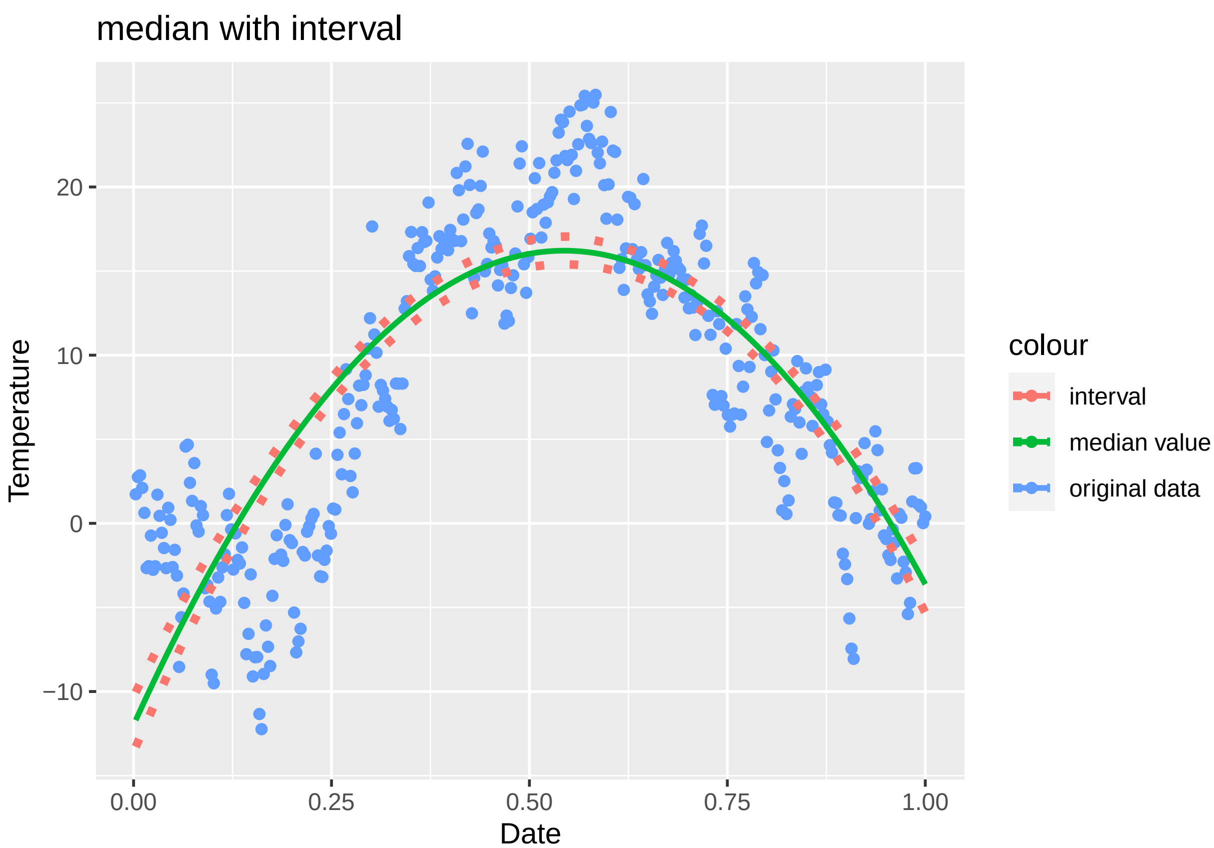

```r
ggplot(df,aes(beta2)) +
    geom_histogram(aes(y = ..density..),
                   colour = "black",
                   fill   = "white",
                   bins   = 40) +
    geom_density(alpha = .2, fill = "purple")
```




```r
ggplot(df,aes(sigma)) +
    geom_histogram(aes(y = ..density..),
                   colour = "black",
                   fill   = "white",
                   bins   = 40) +
    geom_density(alpha = .2, fill = "purple")
```




```
df = df[,1:3]
column_median <- apply(df,2,median)
res1b <- column_median%*%t(X)
pre <- as.matrix(df) %*% t(X) # regression function
pre_interval <- data.frame(nrow = n_obs, nrow = 2)
colnames(pre_interval) <- c("i0.025","i0.975")

for(i in 1:n_obs){
  data_t <- pre[,i]
  pre_interval[i,] <- quantile(data_t, probs = c(0.025,0.975))
}

df1b <- cbind(tempdata, t(res1b), pre_interval)
ggplot(df1b) +
  geom_point(aes(x = time, y = temp, color = "original data")) +
  geom_line(aes(x = time, y = t(res1b), color = "median value"),size = 1) +
  geom_line(aes(x = time, y = i0.025, color = "interval"), linetype = "dotted", size = 1.5) +
  geom_line(aes(x = time, y = i0.975, color = "interval"), linetype = "dotted", size = 1.5) +
  ggtitle('median with interval')+
  ylab("Temperature") + xlab("Date")
```



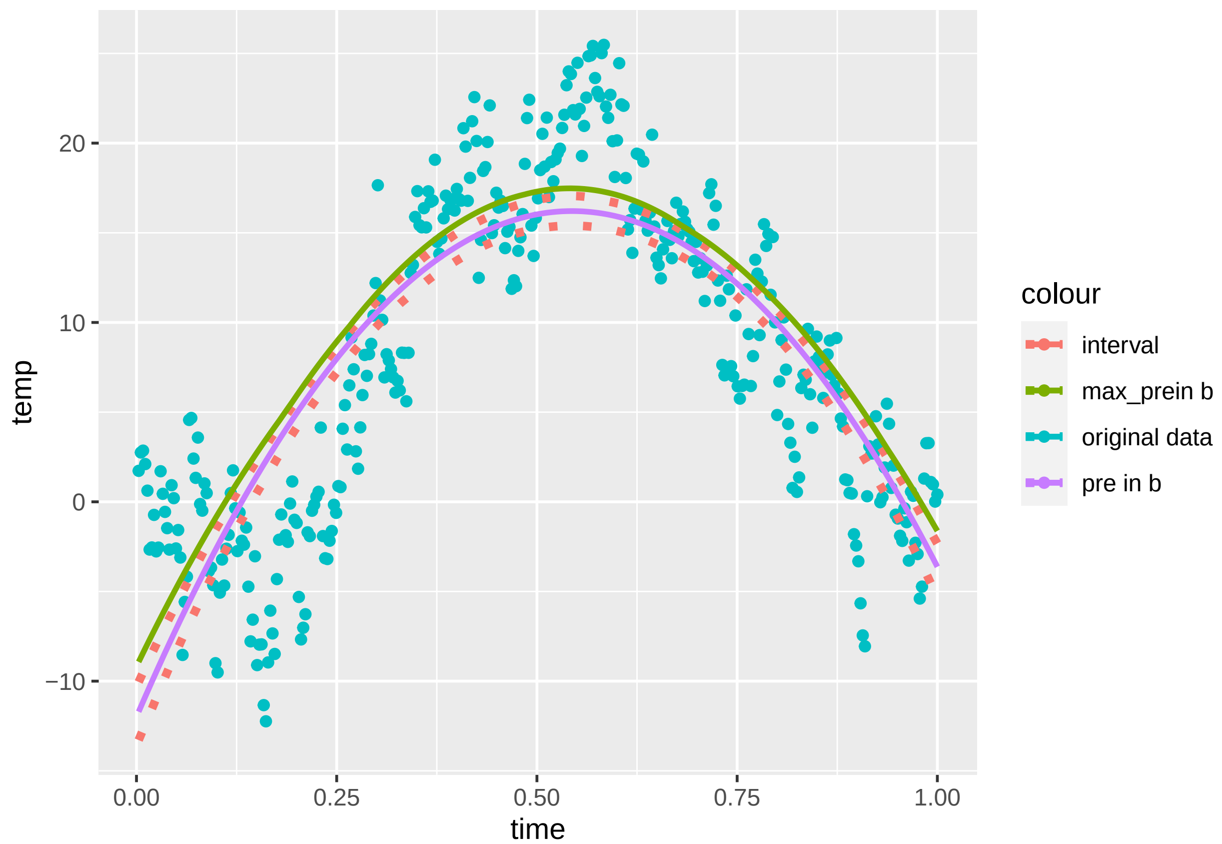
ii.

The result above show the posterior median and its (2.5%-97.5%) credible interval. From the plot, we can see that most observations are out of interval. We should not expect this interval will contain most observations because this interval is actually measure how good the beta is (since there is no ϵ in the model). If we have reasonable ϵ in our model, maybe we can witness that most data is within the interval.

c)

By selecting the maximum results from b), we can easily plot the results as show below.

```
maximum_pre <- c()
for(i in 1:365){
  maximum_pre <- c(maximum_pre,max(pre[,i]))
}
df1c <- cbind(tempdata, t(res1b), pre_interval, maximum_pre)
ggplot(df1c) +
  geom_point(aes(x = time, y = temp, color = "original data")) +
  geom_line(aes(x = time, y = t(res1b),color = "pre in b"),size = 1) +
  geom_line(aes(x = time, y = i0.025, color = "interval"), linetype = "dotted", size = 1.5) +
  geom_line(aes(x = time, y = i0.975, color = "interval"), linetype = "dotted", size = 1.5) +
  geom_line(aes(x = time,y = maximum_pre, color = "max_prein b"), linetype = "solid", size = 1)
```



d)

From the course slides, the proper prior of beta might be gaussian.

$\beta_i \mid \sigma^2 \sim N(\mu_0, \frac{\sigma^2}{\lambda}), \quad where \;\; \Omega_0 = \lambda \cdot I_3,$

Question 2 Posterior approximation for classification with logistic regression

a)

```
wdata <- read.table("WomenAtWork.dat", header = TRUE)
Probit <- 0
Covs <- c(2:8)
lambda <- 1
Nobs <- dim(wdata)[1]
y <- wdata$Work
X <- as.matrix(wdata[,Covs])
Xnames <- colnames(X)
Npar <- dim(X)[2]

mu <- as.matrix(rep(0,Npar))
Sigma <- (1/lambda)*diag(Npar)
```

```r
LogPostLogistic <- function(betas,y,X,mu,Sigma){
  linPred <- X%*%betas;
  logLik <- sum( linPred*y - log(1 + exp(linPred)) );
  #if (abs(logLik) == Inf) logLik = -20000; # Likelihood is not finite, stear the optimizer away from h
  logPrior <- dmvnorm(betas, mu, Sigma, log=TRUE);

  return(logLik + logPrior)
}

LogPostProbit <- function(betas,y,X,mu,Sigma){
  linPred <- X%*%betas;
  SmallVal <- .Machine$double.xmin
  logLik <- sum(y*log(pnorm(linPred)+SmallVal) + (1-y)*log(1-pnorm(linPred)+SmallVal) )
  logPrior <- dmvnorm(betas, mu, Sigma, log=TRUE);
  return(logLik + logPrior)
}

# Select the initial values for beta
initVal <- matrix(0,Npar,1)
if (Probit==1){
  logPost = LogPostProbit;
} else{
  logPost = LogPostLogistic;
}

opt <- optim(initVal,logPost,gr=NULL,y,X,mu,Sigma,method=c("BFGS"),control=list(fnscale=-1),hessian=TRUI

names(opt$par) <- Xnames # Naming the coefficient by covariates
approxPostStd <- sqrt(diag(-solve(opt$hessian))) # Computing approximate standard deviations.
names(approxPostStd) <- Xnames # Naming the coefficient by covariates
print('The posterior mode is:')
```

```
## [1] "The posterior mode is:"
```

```r
print(opt$par)
```

```
##              [,1]
## [1,]  0.21430601
## [2,] -0.03361233
## [3,]  0.18433780
## [4,]  0.12177139
## [5,] -0.05851682
## [6,] -1.34335510
## [7,] -0.04986357
## attr(,"names")
## [1] "Constant"    "HusbandInc"  "EducYears"   "ExpYears"    "Age"
## [6] "NSmallChild" "NBigChild"
```

```r
print('The approximate posterior standard deviation is:')
```

```
## [1] "The approximate posterior standard deviation is:"
```

```
approxPostStd <- sqrt(diag(-solve(opt$hessian)))
print(approxPostStd)
```

```
## [1] 0.85177040 0.01938511 0.07294511 0.02967776 0.02122354 0.37201051 0.13317877
```

```
child_feature_data <- as.data.frame(
  mvtnorm::rmvnorm(n = 1000, mean = opt$par, sigma = -solve(opt$hessian))
  )[,6]


CI_0_025 <- quantile(child_feature_data, probs = c(0.025,0.975))[1]
CI_0_975 <- quantile(child_feature_data, probs = c(0.025,0.975))[2]
interval <- c(CI_0_025,CI_0_975)
print('The 95% equal tail interval is :')
```

```
## [1] "The 95% equal tail interval is :"
```

```
cat(interval)
```

```
## -2.027204 -0.6616174
```

From the result of posterior mode, we can see that `NSmallChile` has great negative impact on womenWork (-1.34), so we can say that this feature is important for the probability that a women works.

And by applying the built-in function`glm()`, we can get coefficients from maximum likelihood estimation, the results are similar to what we have above.

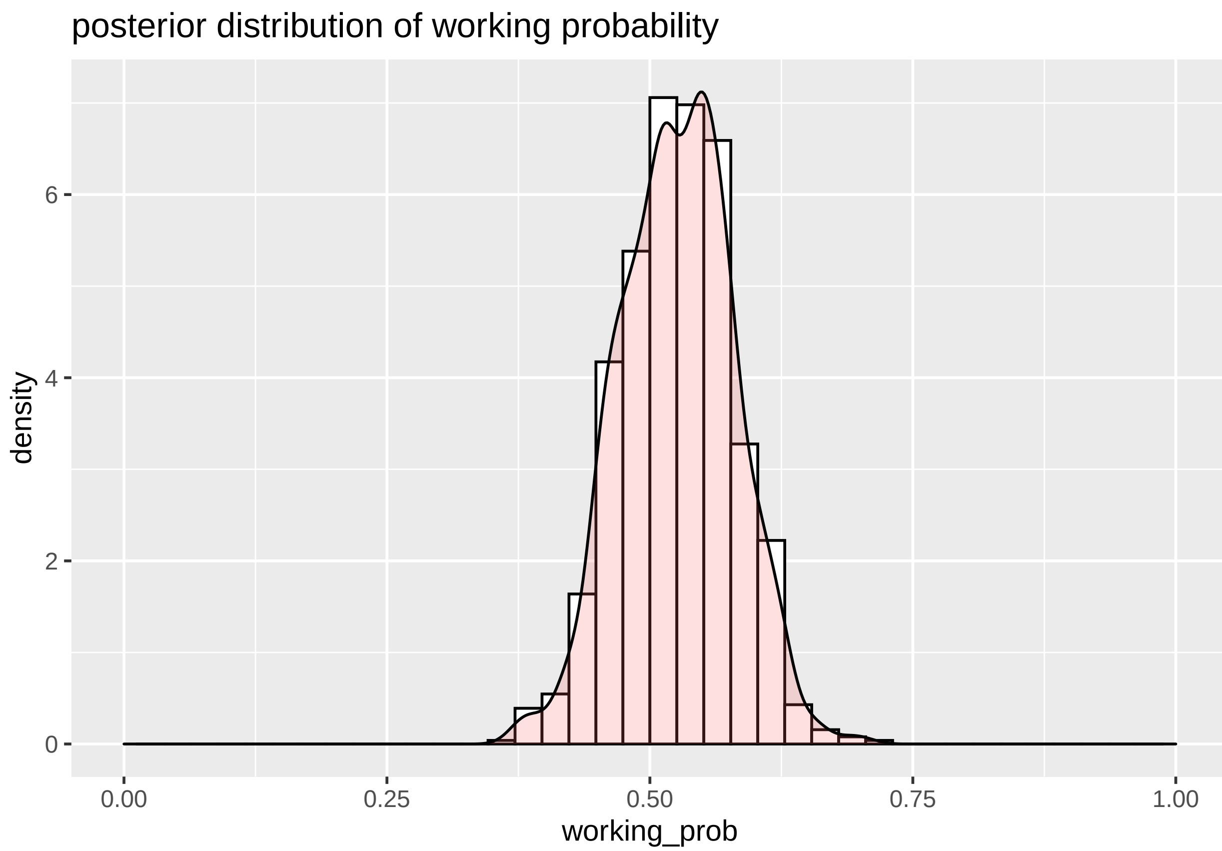
Constant HusbandInc EducYears ExpYears Age NSmallChild NBigChild

1.12242734 -0.03425216 0.17650532 0.12317305 -0.07475060 -1.64598118 -0.08973248

b)

```
situation <- c(1,20,12,8,43,0,2)
set.seed(123)
df <- as.data.frame(rmvnorm(n = 1000, mean = opt$par, sigma = -solve(opt$hessian)))
draw <- function(situation,df)
{
  samples <<- as.data.frame(t(situation %*% t(df)))
  res <- as.data.frame(1/(1+exp(-samples)))
  colnames(res) <- "working_prob"
  res$work <- ifelse(res$working_prob < 0.5, "not work", "work")
  res$work_label <- ifelse(res$working_prob < 0.5, 0, 1)
  res$nr <- c(1:nrow(res))
  res_2b <<- res
}
draw(situation,df)
ggplot(data = res_2b, aes(x = working_prob)) +
  geom_histogram(aes(y = ..density..),
                 colour = "black",
                 fill   = "white",
```

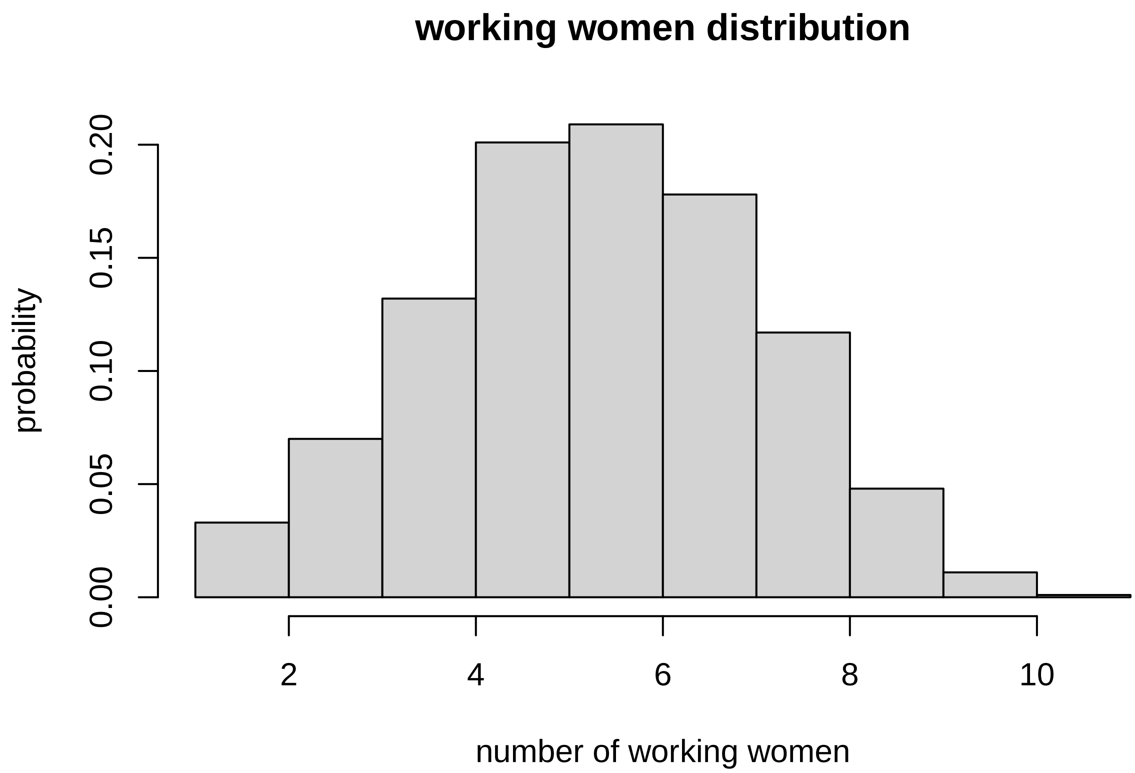
```
                 bins    = 40) +
  geom_density(alpha = .2, fill = "#FF6666") +
  ggtitle("posterior distribution of working probability") +
  xlim(c(0,1))
```



From the plot, we can see that the mean probability of working is a little higher than 0.5 which means this women is more likely to work.

c)

```
situation <- c(1,20,12,8,43,0,2)
N_obs <- 11
draw <- function(situation,n,means,covar,n_obs)
{
  betas <- rmvnorm(n,mean=means,sigma = covar)
  situation <- as.matrix(situation)
  samples <- betas%*%situation
  prob <- exp(samples)/(1+exp(samples))
  posterior <- sapply(prob,FUN=function(x){rbinom(1,n_obs,x)})
  return(posterior)
}
posterior <- draw(situation,1000,opt$par,-solve(opt$hessian),N_obs)
hist(posterior,breaks = 8,freq = FALSE,main = "working women distribution",xlab='number of working wome
```


working women distribution
probability
0.00
0.05
0.10
0.15
0.20
2
4
6
8
10
number of working women