Computer Lab 2

Kevin Neville, Gustav Sternelöv

17 april 2016

Multinomial model with Dirichlet prior

1.a)

Prior for θ

$$\theta \sim \frac{1}{B(\alpha)} \prod Y_i^{\alpha_i+1}$$
$$\propto \prod Y_i^{\alpha_i+1}$$

$$\frac{n!}{Y_1! \cdot ... Y_k!} P_1^{Y_1} \cdot ... P_k^{Y_k}$$
$$\propto \prod P_i^{Y_i}$$

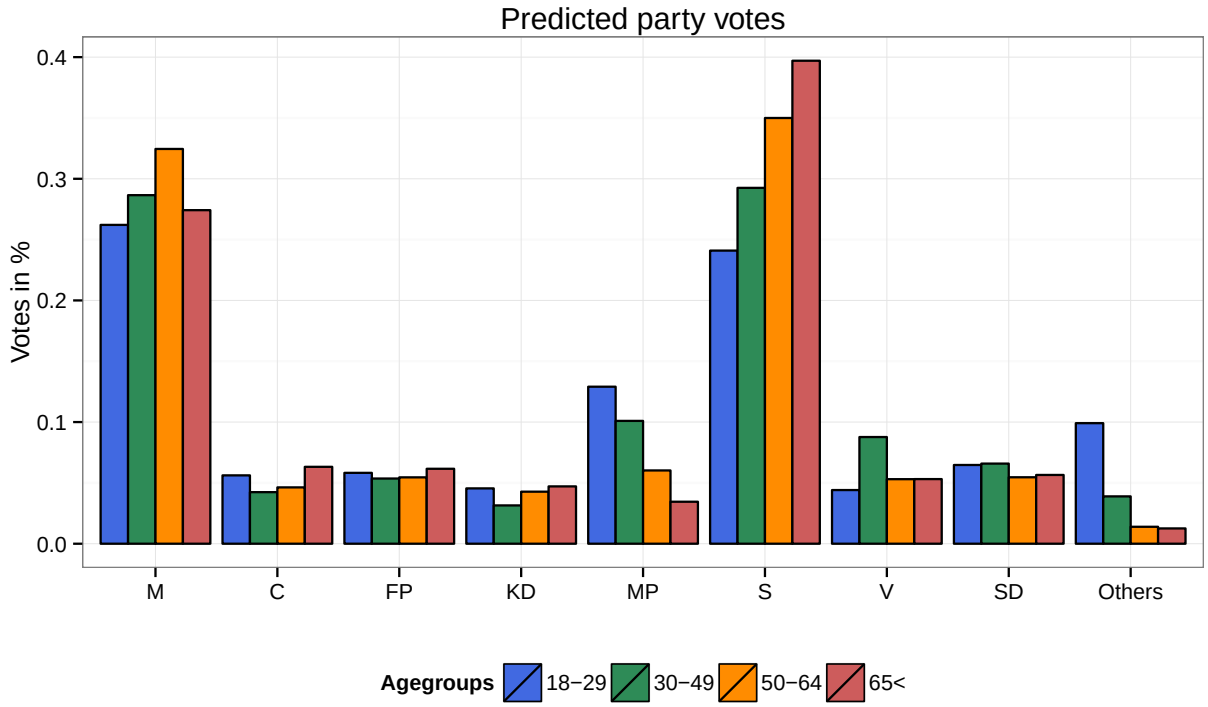
Posterior

$$\prod P_i Y_i^{Y_i+\alpha_i-1}$$

Which is a $Dirichlet(\alpha_1 + y_{i1}, ..., \alpha_k + y_{ik})$.

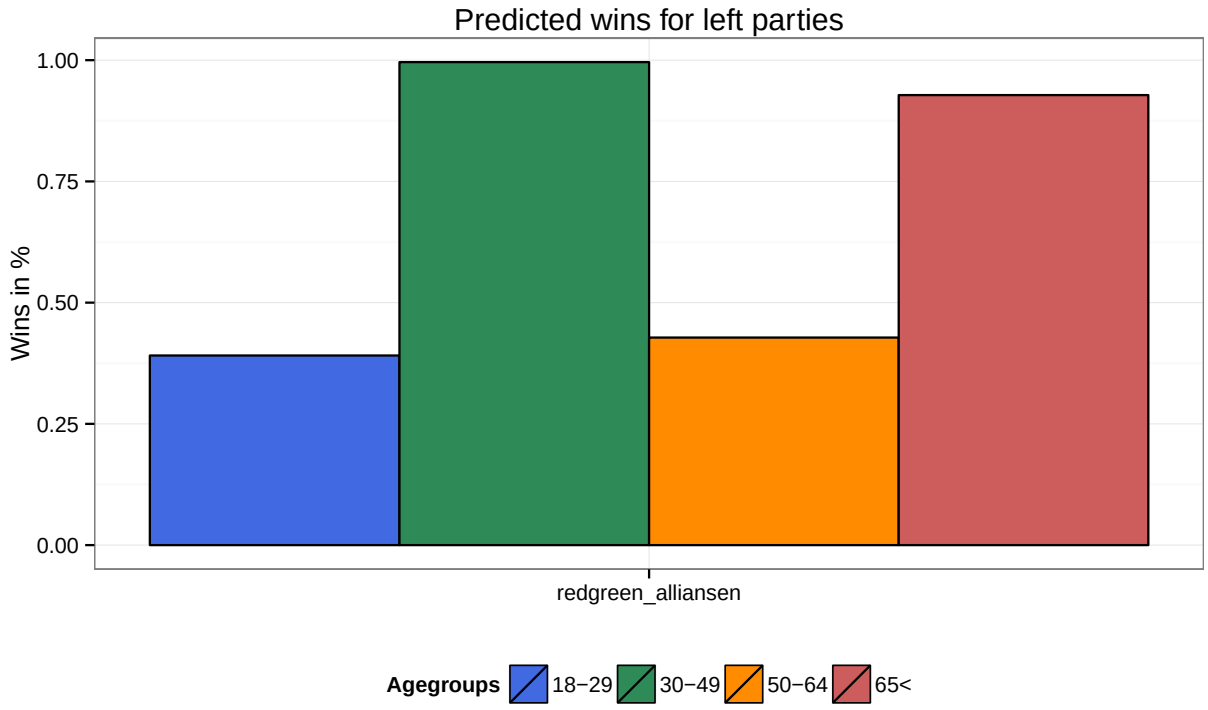
1.b)

The voting behaviour in different age groups is compared with the grouped bar plot below.



The clearest differences between the age groups can be seen for *MP* and *S*. It is predicted that it is much more common that younger people votes for *MP* than older people. Regarding the votes for *S* is the case that the party is predicted to be much more popular amongst the older population than the younger.

1.3



The posterior probability that the Red- Greens (S, MP, V) will win against the Alliance (M, C, FP, KD) is shown for all four age groups. Those in the age span of 30-49 and 65< are more in favor of the Red-Greens than the two other age spans. The Red-Greens has the lowest probability of winning in the age span of 18-29.

1.4

Given the total population in each age group is the predicted probability that the Red-Greens will win the election 94.5 %. Hence, in nearly all of the simulated draws of the election did the Red-Greens win.

Assignment 2 - Linear and polynomial regression

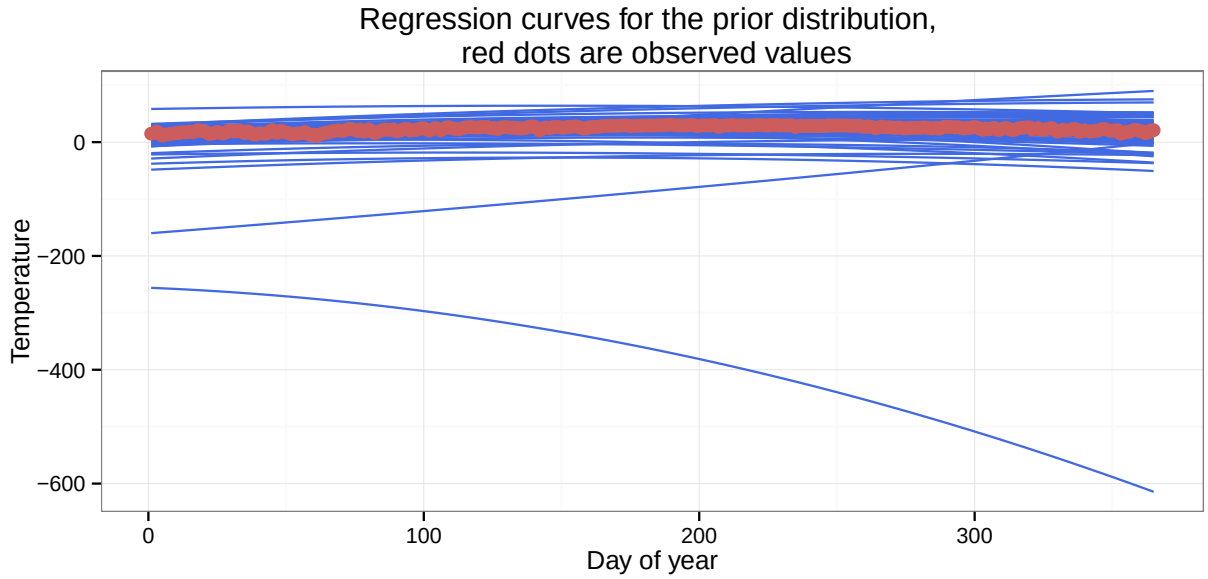
a-b)

To get some information about what values that can be reasonable to use as priors is a quadratic model using plain least squares fitted. A summary of the results from this model is shown in the output below:

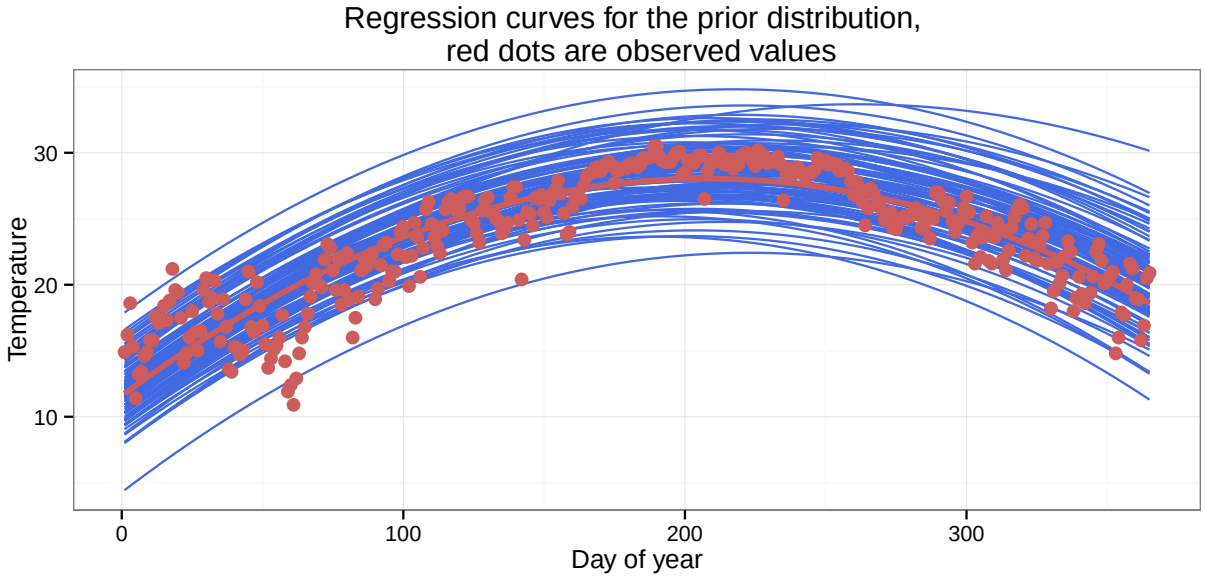
```
##
## Call:
## lm(formula = temp ~ time + I(time^2), data = JapanTemp)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.9271 -1.2670  0.4481  1.4537  6.8899
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.5826     0.3445   33.62   <2e-16 ***
## time         57.8302     1.5864   36.45   <2e-16 ***
## I(time^2)   -50.8155     1.5321  -33.17   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.182 on 362 degrees of freedom
## Multiple R-squared:  0.7948, Adjusted R-squared:  0.7937
## F-statistic: 701.1 on 2 and 362 DF,  p-value: < 2.2e-16
```

By using the results from above the prior β vector is set to (11.58,58,-50) and the σ_0^2 prior is set to 122.82, the average sum of squares with 11.58 as the mean value. Regarding the degrees of freedom and the Ω_0 hyperparameter are more of a trial and error approach applied. First they are set to 1 and 1, which together with the values of the other hyperparameters gave the following results.



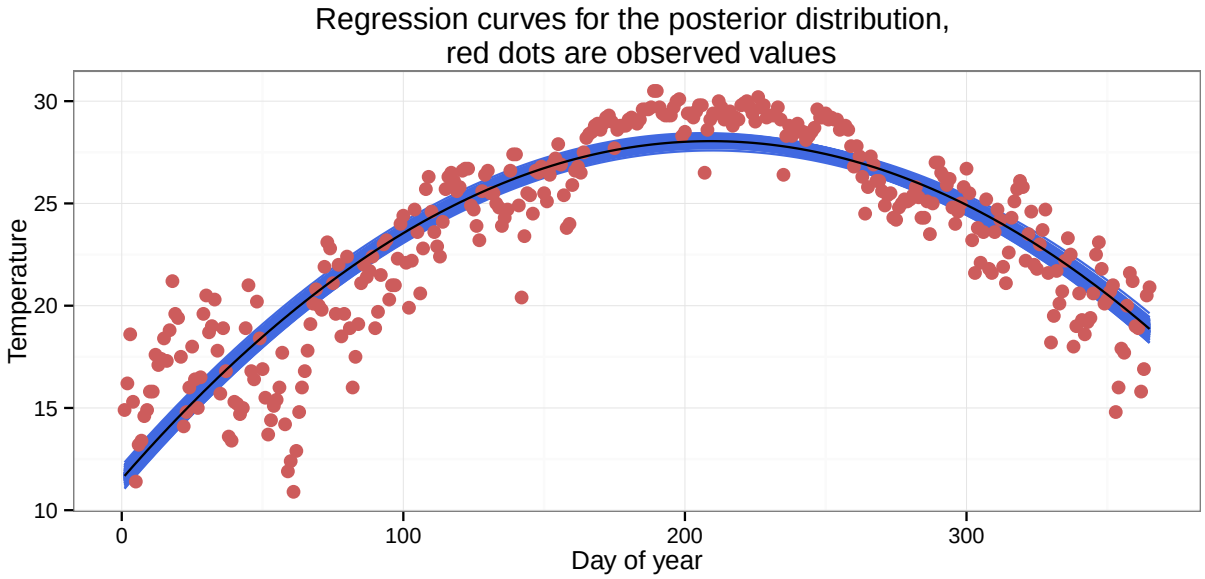
100 draws are made from the joint prior of all parameters and for each draw is a regression curve computed. As can be seen above are the regression curves given by the prior not so well fitted to data and in many cases quite far off the temperatures that can be expected. Hence, the prior hyperparameters v_0 and Ω_0 needs to be given other, more sensible, values. This is done by testing some different values for the respective parameters and it is concluded that v_0 equal to 10 and Ω_0 equal to 30 gives a reasonable prior distribution.



The regression curves obtained from the updated prior distribution are thought to be reasonable since they rather well agrees with our prior belief.

c)

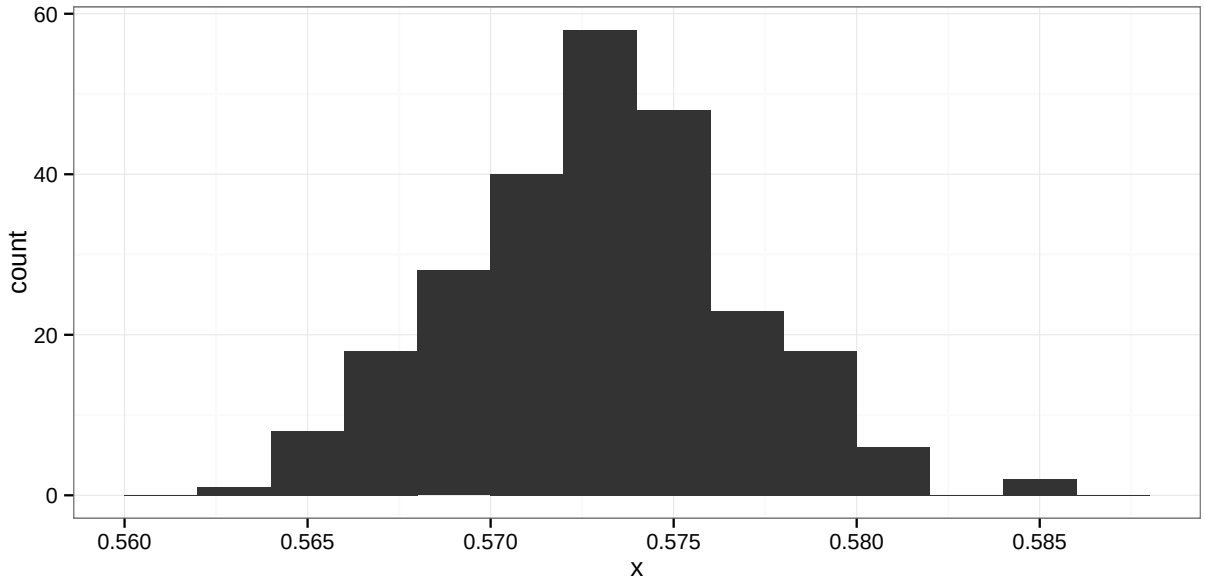
The joint posterior distriubtion for the β_0, β_1, β_2 and σ^2 is derived and then simulations are generated from it. 250 draws are simulated from the posterior distribution and the obtained regression curves are plotted in the graph below. The black line is the mean of all curves.



The regression curves obtained by the draws from the joint posterior distribution are thought to be reasonable. In general the curves follows the observed data rather well. The only exception is for the period with highest temperatures where the model consistently underestimates the temperature.

d)

The time when *E(temp|time)* is maximal is given by setting $\frac{\partial f}{\partial Time} = 0$, where f is $\beta_0 + \beta_1 * time + \beta_2 * time^2$. This returns the expression $time = -\frac{\beta_1}{2\beta_2}$. By using the simulations in *c)* are values simulated from the posterior distribution of the day with highest temperature. The histogram for this distribution is plotted below.



As can be seen in the histogram does the time in the posterior vary from around 0.565 to 0.585. This is approximately equal to the days 206-214.

e)

A suitable prior would be one that regulates so that the parameters not overfitts data. That problaby is a prior that regularizes the coefficients in the model quite heavily.

When the model has polynomials up to order 7, the aim is to regularize the parameters of higher orders. That since models with polynomials of high orders tends to be overfitted to data.

The regularization can be accomplished by setting the prior β parameters to zero and Ω_0 to a sufficiently high value.The effect of this is that each variable, or coefficient, needs to be important. Otherwise its parameter value will be close to zero and the importance of the parameter low.

Appendix

R-code

```
# Reading data
age18_29 <- c(M=208, C=45, FP=46, KD=35, MP=110,
  S=189, V=34, SD=53, Others=88)

age30_49 <- c(M=403, C=58, FP=74, KD=42, MP=146,
              S=413, V=127, SD=93, Others=57)
```

```
age50_64 <- c(M=370, C=51, FP=60, KD=47, MP=67,
              S=401, V=59, SD=61, Others=15)

age65 <- c(M=383, C=89, FP=86, KD=65, MP=45,
              S=567, V=74, SD=79, Others=17)

df <- cbind(age18_29,age30_49,age50_64,age65) # Put everything in one matrix
prior <- c(alpha1=30, alpha2=6, alpha3=7, alpha4=6, alpha5=7, # Reading prior
alpha6=30, alpha7=6, alpha8=6, alpha9=2)
df <- as.data.frame(cbind(df, prior)) # Make a data.frame of all data and given priors



#install.packages("gtools") # Needed for simulating from Drichlet.
library(gtools)

mylist <- list(age18_29=c(),age30_49 =c(),age50_64=c(),age65=c()) # Create a list to fill with data for


# Creates a list with 1000 simulations of probabbilites of the partys for each age group.
for(i in 1:4){
  set.seed(311015)
  temp <- as.data.frame(rdirichlet(1000, df[,i] + df$prior))
  colnames(temp) <- c("M", "C", "FP", "KD", "MP", "S", "V", "SD", "Others")
  mylist[[i]] <- list(Prob=temp)
}
library(reshape2)
library(ggplot2)

# Setting up the required data.frame for ggplot.
gg_groupedBar <- data.frame(NA)
for(i in 1:4){
  for(j in 1:1000){
gg_groupedBar[i,j] <- c(colMeans(mylist[[i]]$Prob), agegroup=i)[j]
}}
colnames(gg_groupedBar) <- c("M", "C", "FP", "KD", "MP", "S", "V", "SD", "Others", "agegroup")
gg_groupedBar <- melt(gg_groupedBar,id.vars = "agegroup")

# Plottings
ggplot(data=gg_groupedBar[1:36,], aes(x=variable, y=value, fill=factor(agegroup))) +
  geom_bar(stat="identity", position=position_dodge(), colour="black") +
  labs(title="Predicted party votes", x="", y="Votes in %") +
  scale_fill_manual(values=c("royalblue","seagreen","darkorange","indianred"),
                    name="Agegroups",
                    labels=c("18-29", "30-49", "50-64", "65<")) +
  theme_bw() + theme(legend.position="bottom")
# Redgreen vs Alliansen. Caluclating amount of simulations in favor of redgreen wins.
redgreen_alliansen <- c()
for(i in 1:4){
  for(j in 1:1000){
    redgreen_alliansen[j] <- sum(mylist[[i]]$Prob[j,c(1:4)]) < sum(mylist[[i]]$Prob[j,c(5:7)])
  }
  mylist[[i]]$redgreen_alliansen <- mean(redgreen_alliansen)
```

```
}

# Setting up the ggplot required dataframe.
gg_redgreen_alliansen <- data.frame(agegroup=1:4, variable=rep("redgreen_alliansen", 4),
                                    value=c(mylist$age18_29$redgreen_alliansen, mylist$age30_49$redgreen
                                      mylist$age50_64$redgreen_alliansen, mylist$age65$redgreen_allianse

# Plotting
ggplot(data=gg_redgreen_alliansen, aes(x=variable, y=value, fill=factor(agegroup))) +
  geom_bar(stat="identity", position=position_dodge(), colour="black") +
  labs(title="Predicted wins for left parties", x="", y="Wins in %") +
  scale_fill_manual(values=c("royalblue","seagreen","darkorange","indianred"),
                    name="Agegroups",
                    labels=c("18-29", "30-49", "50-64", "65<")) +
  theme_bw() + theme(legend.position="bottom")
# Simulate amount of voters for each age group.
set.seed(311015)
numVoter <- as.data.frame(t(rmultinom(1000,6300000, c(0.2, 0.3, 0.3, 0.2))))
colnames(numVoter) <- c("age18_29", "age30_49", "age50_64", "age65")

# Calculate amount of voters for each party and age group using the
# simulated number of voters from numVoter.
Prob_numVoter <- as.data.frame(matrix(NA, nrow = 1000, ncol=9))
colnames(Prob_numVoter) <- c("M", "C", "FP", "KD", "MP", "S", "V", "SD", "Others")
for(i in 1:4){
  for(j in 1:1000) {
    Prob_numVoter[j,] <- mylist[[i]][[1]][j,]*numVoter[j,i]
  }
  mylist[[i]]$Prob_numVoter <- Prob_numVoter
}


total_win <- c()
# Calculate how many times rÃ¶dgrÃ¶na wins over allisansen.
for(j in 1:1000){
  temp <- mylist[[1]]$Prob_numVoter[j,] + mylist[[2]]$Prob_numVoter[j,] +
  mylist[[3]]$Prob_numVoter[j,] + mylist[[4]]$Prob_numVoter[j,]
total_win[j] <- sum(temp[1:4]) < sum(temp[5:7])
}

mylist$Total <- mean(total_win)
library(gtools)
library(ggplot2)
library(plyr)
library(dplyr)
JapanTemp <- read.delim("C:/Users/Gustav/Documents/Machine-Learning/Lab 6/JapanTemp.dat", sep="", heade
ClassicLM <- lm(temp ~ time+ I(time^2), data=JapanTemp)
summary(ClassicLM)
library(geoR)
library(mvtnorm)
regLine <- data.frame(matrix(vector(), 365, 100))
set.seed(311015)
for(i in 1:100){
```

```
  sigma0 <- rinvchisq(1, df = 1, scale = 122.82)
  priorCoef <- rmvnorm(n=1, mean = c(11.58,58,-50), sigma = diag(x=sigma0/5, 3, 3))
  regLine[,i] <- priorCoef[1] + priorCoef[2] * JapanTemp$time + priorCoef[3] * JapanTemp$time^2
}
require(reshape2)
regLine_m <- melt(regLine)
regLine_m$x <- rep(1:365, 100)
ggplot()+geom_line(data=regLine_m,aes(x=x,y=value,group=variable),col="royalblue") +  geom_line(data=da
            col="indianred", size=1.25) + theme_bw() +
  geom_point(data=JapanTemp,aes(x=1:365, y=temp),col="indianred", size=3)+
  ggtitle("Regression curves for the prior distribution, \n red dots are observed values") + xlab("Day
regLine <- data.frame(matrix(vector(), 365, 100))
set.seed(311015)
for(i in 1:100){
  sigma0 <- rinvchisq(1, df = 10, scale = 122.82)
  priorCoef <- rmvnorm(n=1, mean = c(11.58,58,-50), sigma = diag(x=sigma0/30, 3, 3))
  regLine[,i] <- priorCoef[1] + priorCoef[2] * JapanTemp$time + priorCoef[3] * JapanTemp$time^2
}
regLine_m <- melt(regLine)
regLine_m$x <- rep(1:365, 100)
ggplot()+geom_line(data=regLine_m,aes(x=x,y=value,group=variable),col="royalblue") + geom_line(data=data
            col="indianred", size=1.25) + theme_bw() +
  geom_point(data=JapanTemp,aes(x=1:365, y=temp),col="indianred", size=3)+
  ggtitle("Regression curves for the prior distribution, \n red dots are observed values") + xlab("Day
X <- as.matrix(data.frame(int=rep(1, 365), x=JapanTemp$time,x2=JapanTemp$time^2))
omega0 <- diag(x=30, 3,3)
beta0 <- (as.matrix(c(11.58,58,-50)))
v0 <- 10
s0 <- 122.82
betaHat <- solve(t(X)%*%X) %*%
  t(X) %*% JapanTemp[,2]
omegaNew <- t(X)%*%X + omega0
betaNew <- (solve(t(X)%*%X + omega0)) %*%
  ((t(X)%*%X%*%betaHat)+(omega0%*%beta0))
vNew <- v0 + nrow(JapanTemp)
vNew_sNew <- v0*s0 + t(JapanTemp[,2])%*%JapanTemp[,2] +
  t(beta0)%*%omega0%*%(beta0)- t(betaNew)%*%omegaNew%*%(betaNew)
sNew <- vNew_sNew/vNew
PosteriorLine <- data.frame(matrix(vector(), 365, 250))
PosteriorCoef <- data.frame(matrix(vector(), 250, 3))
set.seed(311015)
for(i in 1:250){
  sigma0 <- rinvchisq(1, df = vNew, scale = sNew)
  PosteriorCoef[i,] <- rmvnorm(n=1, mean = betaNew, sigma = as.numeric(sigma0) * solve(omegaNew))
  PosteriorLine[,i] <- PosteriorCoef[i,1]+PosteriorCoef[i,2]*JapanTemp$time+PosteriorCoef[i,3]*JapanTem
}
PosteriorLine_m <- melt(PosteriorLine)
PosteriorLine_m$x <- rep(1:365, 250)
Posterior <- colMeans(PosteriorCoef)
PosteriorLine1 <- data.frame(y=Posterior[1]+Posterior[2]*JapanTemp$time+
                             Posterior[3]*JapanTemp$time^2, x=JapanTemp$time)
ggplot()+geom_line(data=PosteriorLine_m,aes(x=x,y=value,group=variable),col="royalblue")  +
  theme_bw()+geom_point(data=JapanTemp,aes(x=1:365, y=temp),col="indianred", size=3) +
```

```
  ggtitle("Regression curves for the posterior distribution, \n red dots are observed values") + xlab("
  geom_line(data=PosteriorLine1,aes(x=1:365,y=y),col="black")
DayMax <- data.frame(x=-(PosteriorCoef[,2] / (2*PosteriorCoef[,3])))
ggplot(DayMax, aes(x)) + geom_histogram(binwidth=0.002) + theme_bw() +
  scale_x_continuous(breaks=c(0.56,0.565,0.57,0.575,0.58,0.585))
## NA
```