Bayesian Learning

Lab 2

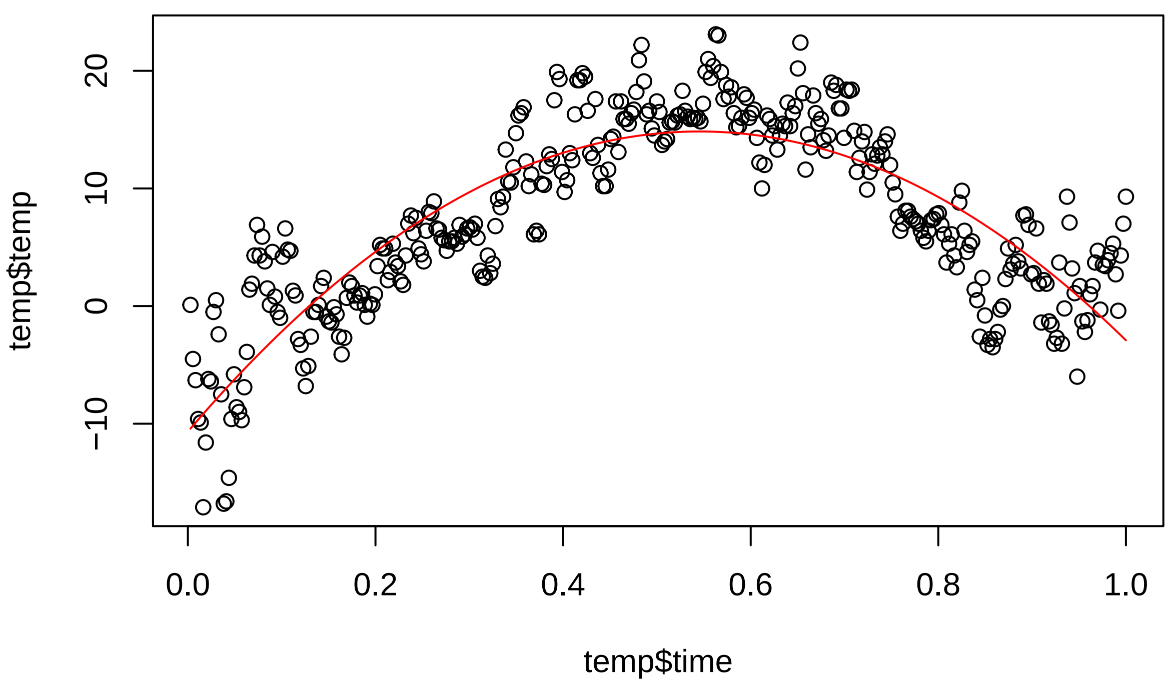
Emil K Svensson and Rasmus Holm

2017-04-26

Question 1

```
temp <- read.table("../data/TempLinkoping2016.txt", header=T)

mod <- lm(temp ~ time + I(time^2), data=temp)
```

```
idx <- order(temp$time)
x <- temp$time[idx]
y <- fitted(mod)[idx]

plot(temp$time, temp$temp)
lines(x, y, col='red', type='l')
```



Prior

$$\begin{aligned}
\sigma^2 &\sim \text{Inv} - \chi^2(\nu_0, \sigma_0^2) \\
\beta|\sigma^2 &\sim \text{N}(\mu_0, \sigma^2\Omega_0^{-1})
\end{aligned}$$

Likelihood

$$\mathbf{y}|\beta, \sigma^2, \mathbf{X} \sim \text{N}(\mathbf{X}\beta, \sigma^2 I_n)$$

Posterior

$$\begin{aligned}
\sigma^2|\mathbf{y} &\sim \text{Inv} - \chi^2(\nu_n, \sigma_n^2) \\
\beta|\sigma^2, \mathbf{y} &\sim \text{N}(\mu_n, \sigma^2\Omega_n^{-1})
\end{aligned}$$

where

$$\begin{aligned}
\mu_n &= (\mathbf{X}^\intercal\mathbf{X} + \Omega_0)^{-1}(\mathbf{X}^\intercal\mathbf{X}\hat{\beta} + \Omega_0\mu_0) \\
\Omega_n &= \mathbf{X}^\intercal\mathbf{X} + \Omega_0 \\
\nu_n &= \nu_0 + n \\
\nu_n\sigma_n^2 &= \nu_0\sigma_0^2 + (\mathbf{y}^\intercal\mathbf{y} + \mu_0^\intercal\Omega_0\mu_0 - \mu_n^\intercal\Omega_n\mu_n)
\end{aligned}$$

a)

```
mu0 <- c(0, 0, 0)
omega0 <- diag(3) * 0.5
nu0 <- 1
sigmasq0 <- 20

hyperparams <- list(mu=mu0, omega=omega0, nu=nu0, sigmasq=sigmasq0)
```

b)

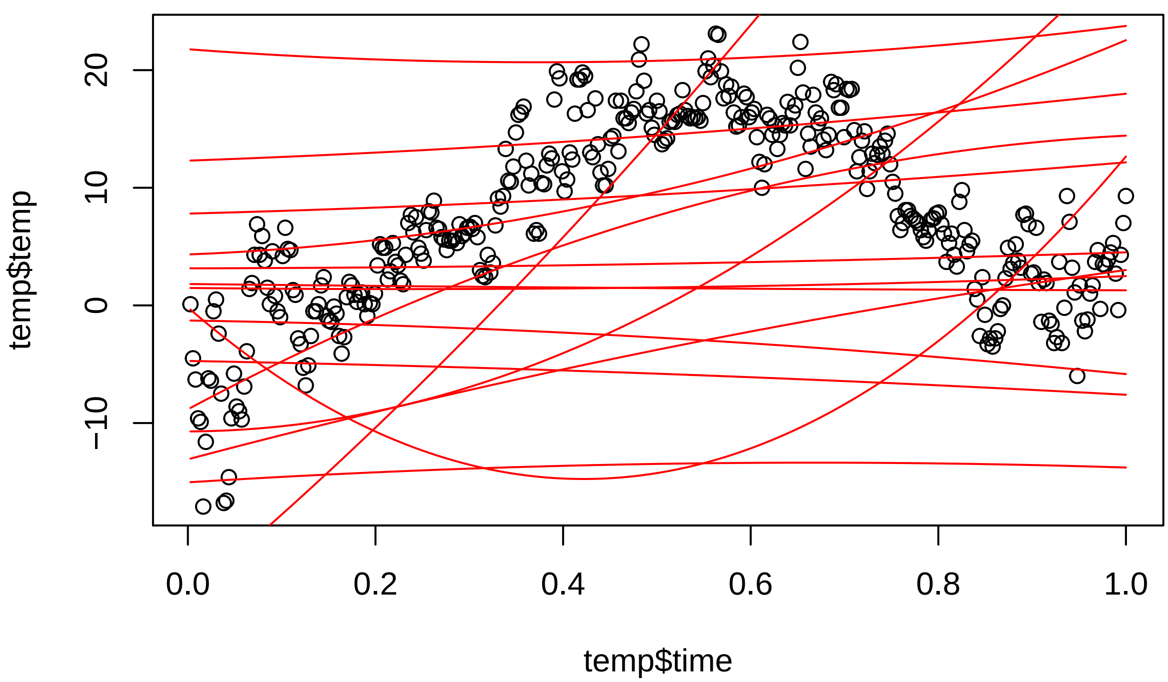
```
library(geoR)
```

```
## ---------------------------------------------------------------
##  Analysis of Geostatistical Data
##  For an Introduction to geoR go to http://www.leg.ufpr.br/geoR
##  geoR version 1.7-5.2 (built on 2016-05-02) is now loaded
## ---------------------------------------------------------------
```

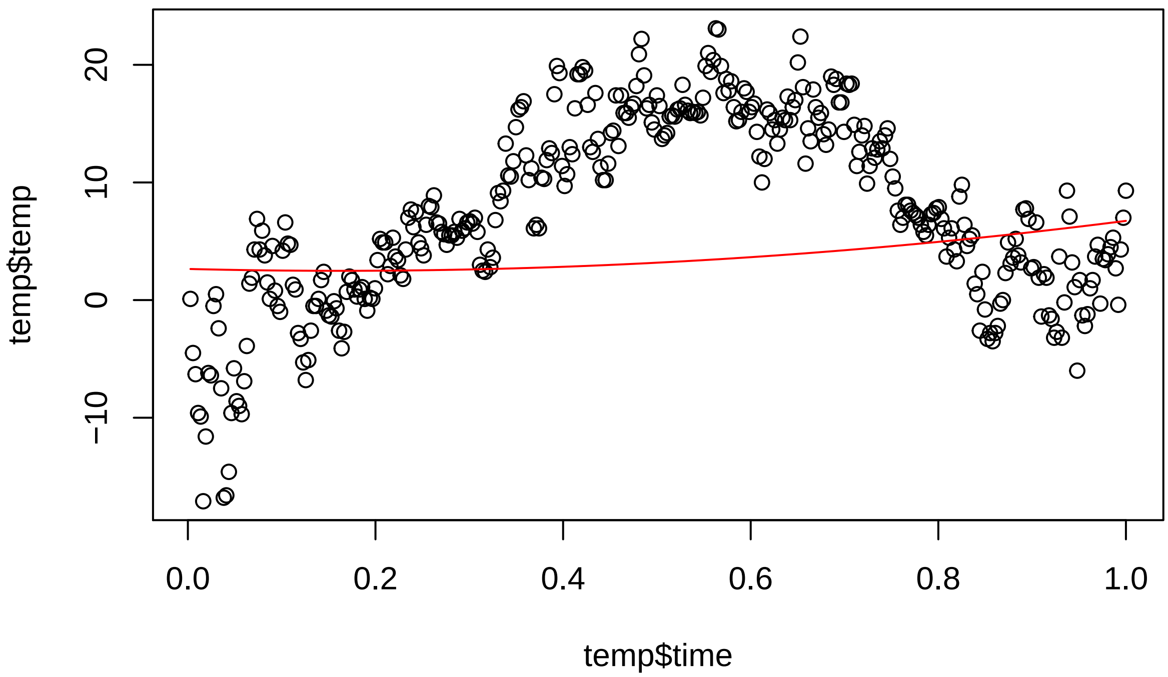
```
library(MASS)

time <- data.frame(rep(1,nrow(temp)), temp$time, temp$time^2)
mtime <- as.matrix(time)
mtemp <- matrix(temp$temp, ncol = 1)

prior_estimate <- function(data, params){
```

```
    sigmasq  <- rinvchisq( n = 1, df = params$nu, scale = params$sigmasq)
    betacoef <- mvrnorm(n = 1, mu = params$mu, Sigma = sigmasq * solve(params$omega) )

    data %*% betacoef
}
```

```
plot(temp$time, temp$temp)

x <- sort(temp$time)

set.seed(12345)

for (i in 1:20){
    y <- prior_estimate(mtime, hyperparams)[order(temp$time)]
    lines(x, y, col='red', type='l')
}
```




```
set.seed(12345)

x <- sort(temp$time)
y <- rowMeans(sapply(1:1000, function(x) prior_estimate(mtime, hyperparams)[order(temp$time)]))

plot(temp$time, temp$temp)
lines(x, y, col='red', type='l')
```

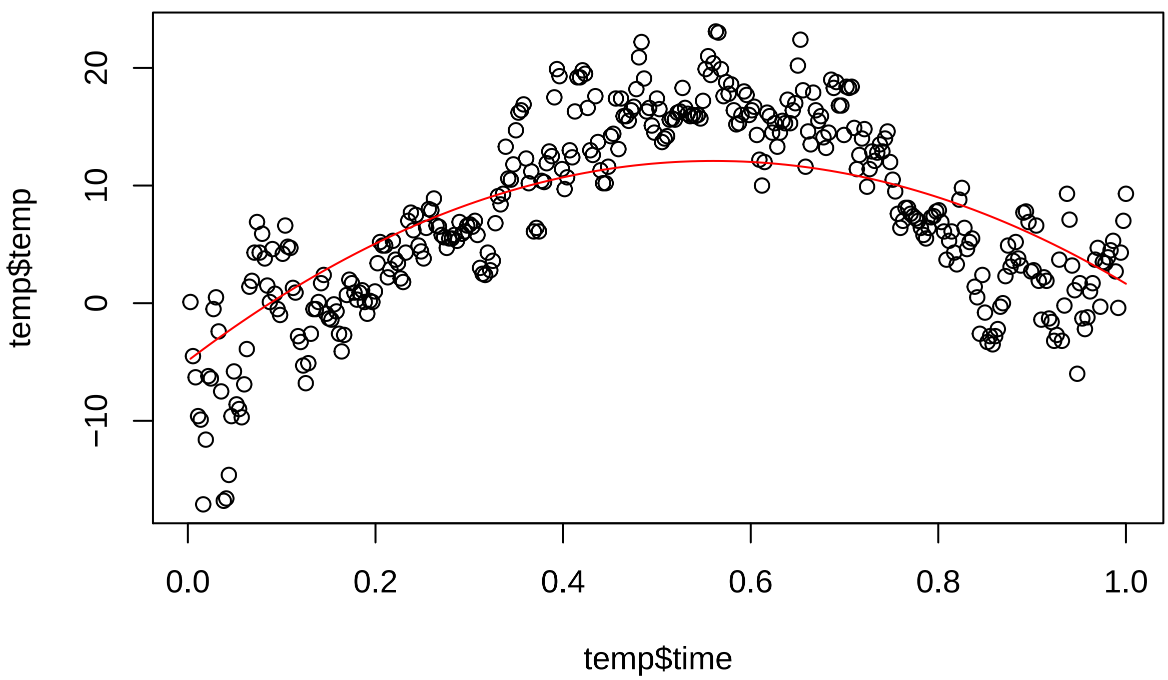


c)

```
posterior_param_sample <- function(X, y, hyperparams){
    XX <- t(X) %*% X
    betahat <- solve(XX) %*% t(X) %*% y
    mun <- solve(XX + hyperparams$omega) %*%
        (XX %*% betahat + hyperparams$omega %*% hyperparams$mu)
    omegan <- XX + hyperparams$omega
    nun <- hyperparams$nu + nrow(X)
    nunsigmasqn <- hyperparams$nu * hyperparams$sigmasq +
        (t(y) %*% y +
         t(hyperparams$mu) %*% hyperparams$omega %*% hyperparams$mu -
         t(mun) %*% omegan %*% mun )
    sigmasqn <- nunsigmasqn / nun

    sigmasq  <- rinvchisq(n = 1, df=nun, scale=sigmasqn)
    beta <- mvrnorm(n = 1, mu = mun, Sigma = as.numeric(sigmasq) * solve(omegan))

    list(beta = beta, sigmasq = sigmasq)
}

posterior_estimate <- function(X, y, hyperparams){
    sample <- posterior_param_sample(X, y, hyperparams)
    X %*% sample$beta
```

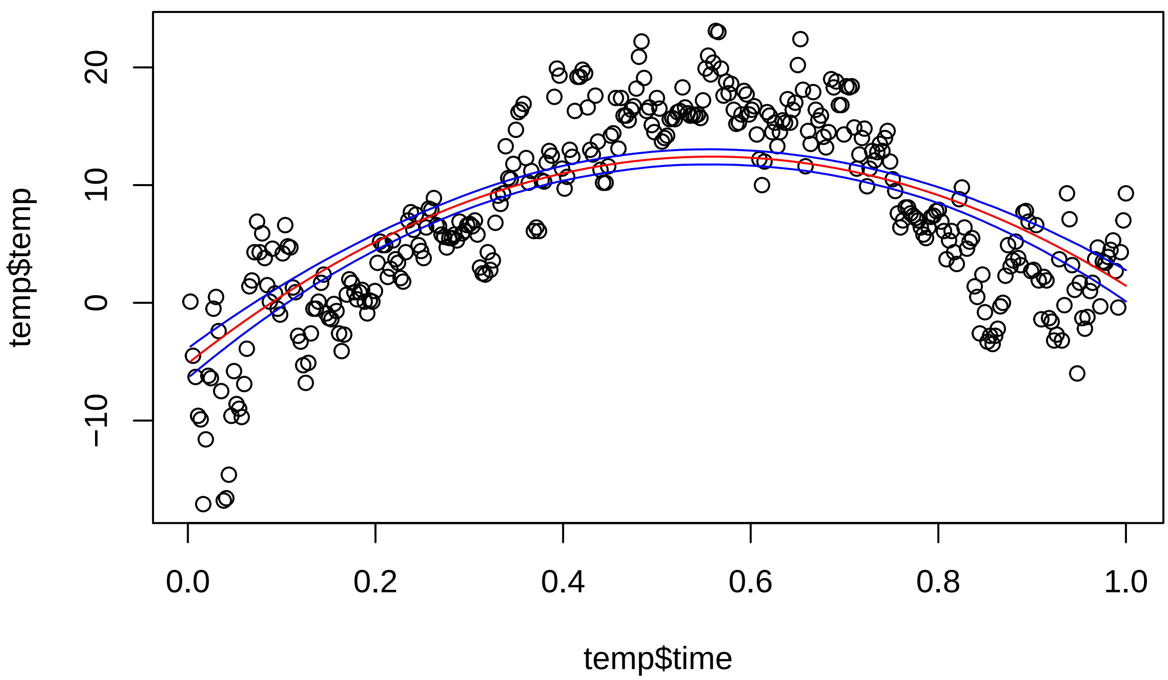
```
}
```

```
plot(temp$time, temp$temp)

set.seed(12345)

idx <- order(temp$time)
x <- temp$time[idx]
y <- posterior_estimate(mtime, mtemp, hyperparams)[idx]

lines(x, y, col='red', type='l')
```




```
set.seed(12345)

ests <- sapply(1:1000, FUN = function(x) posterior_estimate(mtime, mtemp, hyperparams))
cred_interval <- apply(ests, MARGIN = 1, quantile, probs = c(0.05, 0.95))

idx <- order(temp$time)
x <- temp$time[idx]
y1 <- rowMeans(ests)[idx]
y2 <- cred_interval[1,][idx]
y3 <- cred_interval[2,][idx]

plot(temp$time, temp$temp)
lines(x, y1, col='red', type='l')
lines(x, y2, col='blue', type='l')
```

```r
lines(x, y3, col='blue', type='l')
```



d)

```r
set.seed(12345)

betas <- sapply(1:1000, FUN = function(x) posterior_param_sample(mtime, mtemp, hyperparams)$beta)
hot <- mean(-betas[2,] / (2 * betas[3,]))
hot * 366 # July 27, 2016 (Wed)
```

```
## [1] 204.5392
```

e)

Set μ_0 to zeros and a high Ω_0 that expresses a high degree of certainty in our prior.

Question 2

```r
women <- read.table("../data/WomenWorks.txt", header = TRUE)
```

a)

```r
glmModel <- glm(Work ~ 0 + ., data = women, family = binomial)

glmModel
```

```
##
## Call:  glm(formula = Work ~ 0 + ., family = binomial, data = women)
##
## Coefficients:
##    Constant   HusbandInc    EducYears     ExpYears    ExpYears2
##     0.64430     -0.01977      0.17988      0.16751     -0.14436
##         Age  NSmallChild    NBigChild
##    -0.08234     -1.36250     -0.02543
##
## Degrees of Freedom: 200 Total (i.e. Null);  192 Residual
## Null Deviance:       277.3
## Residual Deviance: 222.7     AIC: 238.7
```

```r
summary(glmModel)
```

```
##
## Call:
## glm(formula = Work ~ 0 + ., family = binomial, data = women)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1662  -0.9299   0.4391   0.9494   2.0582
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## Constant     0.64430    1.52307   0.423 0.672274
## HusbandInc  -0.01977    0.01590  -1.243 0.213752
## EducYears    0.17988    0.07914   2.273 0.023024 *
## ExpYears     0.16751    0.06600   2.538 0.011144 *
## ExpYears2   -0.14436    0.23585  -0.612 0.540489
## Age         -0.08234    0.02699  -3.050 0.002285 **
## NSmallChild -1.36250    0.38996  -3.494 0.000476 ***
## NBigChild   -0.02543    0.14172  -0.179 0.857592
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 277.26  on 200  degrees of freedom
## Residual deviance: 222.73  on 192  degrees of freedom
## AIC: 238.73
##
## Number of Fisher Scoring iterations: 4
```

b)

c)