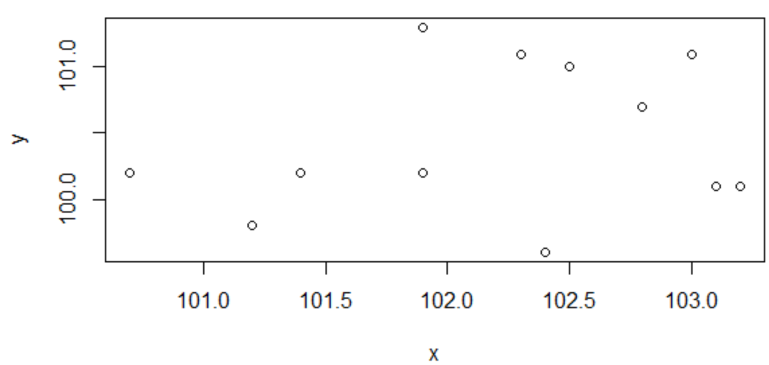
Midterm Math525

Haya Toumy

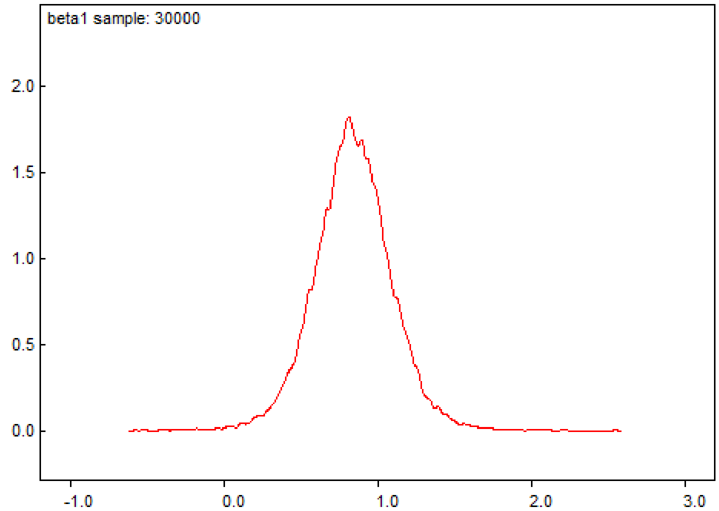
October 16, 2014

Part a:
I plotted the raw data in R and that yielded:

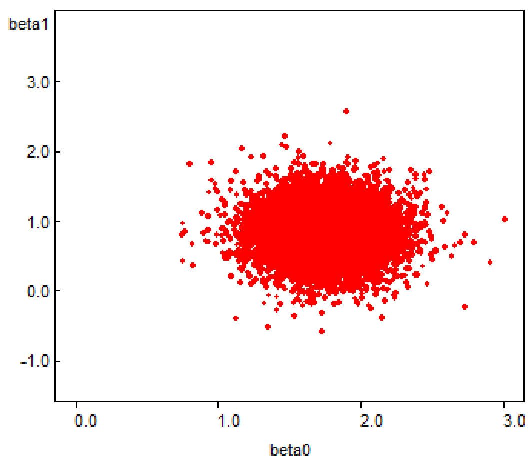


The linear model doesn't appear plausible for the first look at the raw data plot, but we need a "goodness-of-fit" measure (like R^2) to make a better decision; running a standard linear regression of Z on the X-mean(x) returns an $R^2 = 0.5987$; so I conclude that a linear model is plausible.

Part b:
Fitting the given SLR model in WinBUGS (after finding the Z_i values in R), and running 30'000 iterations after 1000 burn-ins, I got the following posterior for β_1:



And a bi-variate scatter plot for the joint posterior of $\beta = (\beta_0, \beta_1)$:

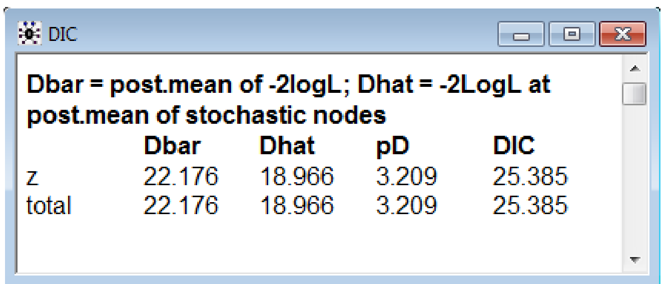


There's no correlation between between β_0 and β_1 because the X's are already centered around their mean in our simple linear regression model.

WinBUGS also gives us the 95% of β_1 which was: (0.3481,1.338), with β_1 mean: 0.8407 and standard deviation: 0.2491.

I can't set a $H_0 : \beta_1 = 0$ and reject it to confirm the investigators' suggestion because I set a vague flat prior on β_1, so testing the R_{01} will give 1; not useful.
But by running: cor(x,y) in R, I had: 0.223615; a positive correlation means the higher x is, the higher z will be; so according to that I confirm the investigators' suggestion that reduction in temperature is related to the initial temperature.

For the DIC and the P_D, WinBUGS returned: $P_D = 3.209$ and DIC=25.385

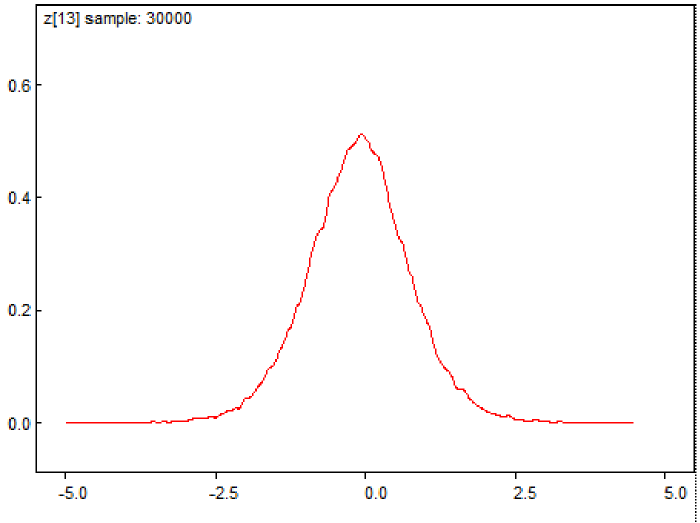


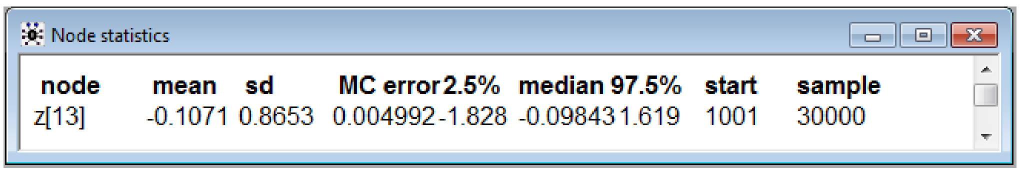
DIC

Dbar = post.mean of -2logL; Dhat = -2LogL at post.mean of stochastic nodes

	Dbar	**Dhat**	**pD**	**DIC**
z	22.176	18.966	3.209	25.385
total	22.176	18.966	3.209	25.385

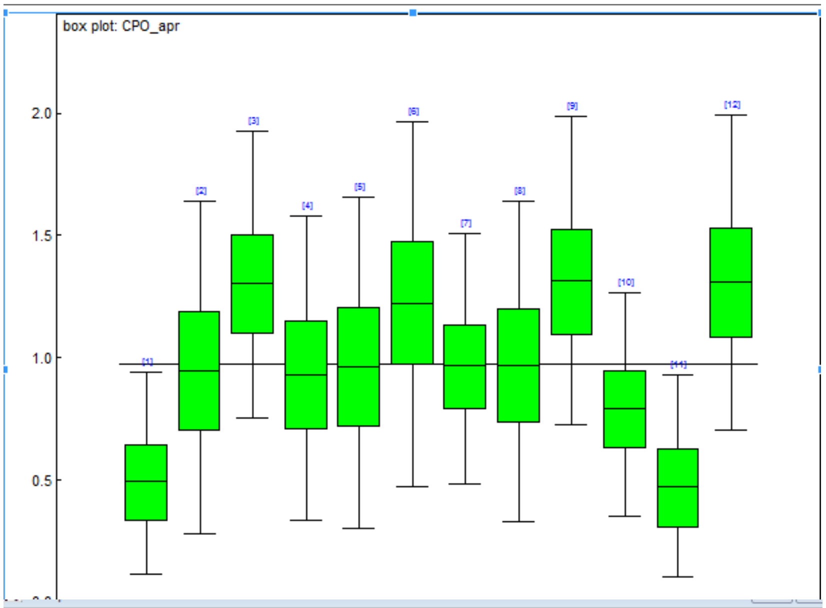
Part c:
I predicted Z after the arrival of the new x_{n+1} in WinBUGS for 30'000 iterations after 1000 burn-ins and monitored Z. (code below).
The density of z_{13} and the mean and standard deviation (the point estimates) as WinBUGS returned are:

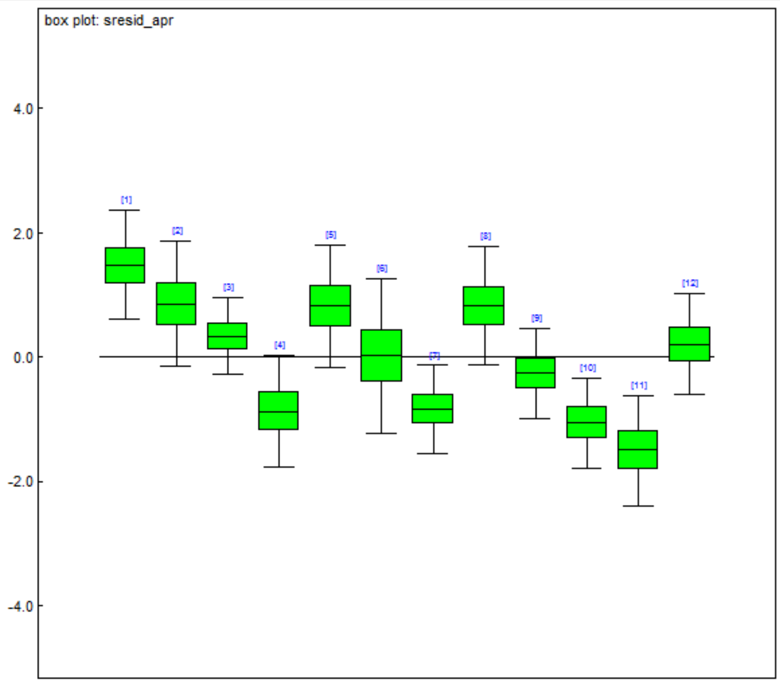



Node statistics

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
z[13]	-0.1071	0.8653	0.004992	-1.828	-0.09843	1.619	1001	30000

Part d:
Following the book's code in WinBUGS again, but I modified it by finding the standarized X's and Z's in R instead and used them in WinBUGS data directly, then ran 20'000 iterations after 1000 burn-ins. I got the following Box-plots:





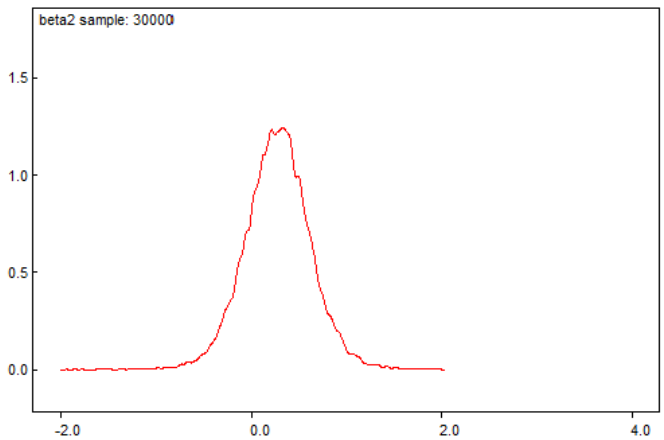
I don't recognize any astray outliers, but most of them on the far end (I tried to change from step=1.5 to step=2 in the WinBUGS code and got similar results); we need more data, I assume this is not enough sample size to decide on outliers.

Checking DIC and p_D again, gives: DIC=28.683 and p_D=3.193

Part e:

The model is now: $Z_i = \beta_0 + \beta_1(x_i - \bar{x}) + \beta_2(x_i - \bar{x})^2 + \epsilon_i$

Running this in WinBUGS for 30'000 iterations after 1000 burn-ins gave this posterior density for β_2:

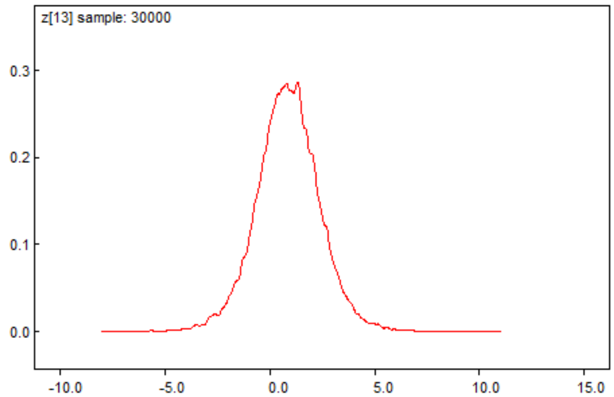


It's normal distribution with mean:0.2769 and sd:0.3475. As I did for β_1 earlier, I took the coda from WinBUGS and put it in R and fitted the distribution to normal and it returned the same mean and standard deviation I got in WinBUGS.

Using the DIC tool in WinBUGS, I got: p_D=4.338 and DIC=26.867

Lower number for DIC than the previous model means the model has enhanced, we have a better fitting model to our data after adding the quadratic term; the data justifies that because we have some entries a little far from the mean, and the new squared distance included them in the model, (both the over and below the line).

To see how that impacts Part c, I predicted z_{13} again for $x_{n+1} = 100$ under this new SLR model; and that done in WinBUGS and checked in R (code below) gave the following normal density and statistics:



Node statistics

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
z[13]	0.868	1.513	0.008822	-2.173	0.8645	3.885	1001	30000

Yes, the second prediction is the justified by the data because I got a positive (mean) reduction in temperature, which accords with the rest of the Z vector $z_1 ... z_{12}$, while the first prediction yielded a negative number (negative mean, with small sd) for z_{13} which doesn't make sense, since it means the patient temperature (most likely) went up after taking aspirin; and that is contradicted by the rest of the Z vector values.

The codes used:

```
##R code for Part a:
#----------------------------
rm(list=ls())
data<-read.table("aspirin_data.txt")
x<-data$V2
y<-data$V3
n<-length(x)
plot(x,y)
##############################
##############################
##WinBUGS code for Part b:
#------------------------------------
model{
  for( i in 1:n) {
    xmean[i] <- (x[i]-mean(x[]))
    z[i] ~ dnorm(mu[i] , tau)

   mu[i] <- beta0+ beta1*xmean[i]
    }

#  beta0 ~ dnorm(0, 0.001)   # for later use
#  beta1 ~ dnorm(0, 0.001)

  beta0 ~ dflat()     # for max comparability to frequentest results
  beta1 ~ dflat()

 tau ~ dgamma(0.1, 0.1)   #  p_D = 3.203,   DIC = 25.365
 sigma <- 1/sqrt(tau)
}  #  end of BUGS code

#Data:
list(x = c( 102.4, 103.2, 101.9, 103.0,
101.2, 100.7, 102.5, 103.1, 102.8, 102.3,
101.9, 101.4),
z=c(2.8, 3.1, 1.7, 1.9, 1.4, 0.5, 1.5, 3.0,
2.1, 1.2, 0.6, 1.2),
          n = 12)
```

```
# Inits:
list( beta0 = 0, beta1 = 1, tau = 1)   # for gamma prior on tau

##############################################################
#R code for fitting beta1 posterior:
#-------------------------------------------
b1post<-read.table("b1post.txt")
library(MASS)
ppost<-b1post$v2
fitdistr(ppost,"normal")
#same as we got in WinBUGS
######################################################################
##WinBUGS code for Part c:
#-----------------------------

model{
  for( i in 1:n) {
    xmean[i] <- (x[i]-mean(x[]))
    z[i] ~ dnorm(mu[i] , tau)

   mu[i] <- beta0+ beta1*xmean[i]

    }

  beta0 ~ dflat()      # for max comparability to frequentist results
  beta1 ~ dflat()

 tau ~ dgamma(0.1, 0.1)    #  p_D = 3.203,    DIC = 25.365
 sigma <- 1/sqrt(tau)


}  #  end of BUGS code

#Data:
list(x = c( 102.4, 103.2, 101.9, 103.0, 101.2,
100.7, 102.5, 103.1, 102.8, 102.3, 101.9, 101.4, 100),
z=c(2.8, 3.1, 1.7, 1.9, 1.4, 0.5, 1.5, 3.0, 2.1,
 1.2, 0.6, 1.2, NA),
```

```
        n = 13)

# Inits:

list( beta0 = 0, beta1 = 1, tau = 1)   #  for gamma prior on tau


#Note about carrying out the model here and in prediction in Part d:
#for some reason, it didn't accept the initials, so I generated ones,
#updated for 1000 iterations as burn-ins, then loaded these initials
#and monitored Z, updated for 30'000 iterations and collected
#information as usual.

##########################################################################
#R code for distribution of z13:
#--------------------------------------
#predictive was found in WinBUGS
partc<-read.table("part-c-prediction.txt")
newz<-partc$V2
library(MASS)
fitdistr(newz,"normal")
#it's indeed normal. results are consistant with what found in WinBUGS
z13<-pnorm(0.0001,-0.120042920,0.868969448)
z13
#######################################################################
##WinBUGS code for Part d:
#--------------------------------
#standardizing the data in R:
#---------------
lz<-c(scale(z))
lx<-c(scale(x))
lx
lz

#now the WinBUGS code:
#-------------------------
#the new data set is the standardized data found in R
#from the linear regression model we find these:
```

```
 beta1<- beta1 / sd(x)
   beta0 <- beta0 - b1 * mean(x)

model{
  for( i in 1:n) {
z[i]~dnorm(mu[i],tau)
mu[i]<-beta0+beta1*x[i]
    }

#  beta0 ~ dnorm(0, 0.001)   # for later use
#  beta1 ~ dnorm(0, 0.001)

  beta0 ~ dflat()     # for max comparability to frequentist results
  beta1 ~ dflat()

 tau ~ dgamma(0.1, 0.1)   #  p_D = 3.203,   DIC = 25.365
 sigma <- 1/sqrt(tau)

for( i in 1:n) {
# standardized residuals without the "leave-one-out", and taking the
# the *ratio* before posterior averaging.
#Results tend to be too close to 0.
   sresid_apr[i] <- (z[i] - mu[i]) / sigma

# flag observations with sresid_apr values bigger than 1.5 as outliers:
   outlier_apr[i] <- step(sresid_apr[i] - 1.5) +
        step(-(sresid_apr[i] +1.5) )
# approximate CPO:
   CPO_apr[i] <-sqrt(tau)* exp(-tau/2* (z[i] - mu[i])* (z[i] -mu[i]))
   }
}  #  end of BUGS code

#Data:
list(x = c( 0.2503559, 1.2517794, -0.3755338, 1.0014235,
-1.2517794, -1.8776690, 0.3755338, 1.1266014, 0.7510676, 0.1251779,
-0.3755338, -1.0014235),
 z=c(1.20805061, 1.55320793, -0.05752622, 0.17257866,
```

```
        -0.40268354, -1.43815549,  -0.28763110, 1.43815549,
        0.40268354, -0.63278842, -1.32310305, -0.63278842),
         n = 12)

# Inits:
list( beta0 = 0, beta1 = 1, tau = 1)   #  for gamma prior on tau
############################################################
##########################################################################
##WinBUGS code for Part e:
#--------------------------------

model{
  for( i in 1:n) {
    xmean[i] <- (x[i]-mean(x[]))
    z[i] ~ dnorm(mu[i] , tau)

   mu[i] <- beta0+ beta1*xmean[i] +beta2*(xmean[i]*xmean[i])

    }

  beta0 ~ dflat()      # for max comparability to frequentist results
  beta1 ~ dflat()
  beta2 ~ dflat()

 tau ~ dgamma(0.1, 0.1)   #  p_D = 3.203,   DIC = 25.365
 sigma <- 1/sqrt(tau)

}  #  end of BUGS code

#Data:
list(x = c( 102.4, 103.2, 101.9, 103.0, 101.2, 100.7,
102.5, 103.1, 102.8, 102.3, 101.9, 101.4),
z=c(2.8, 3.1, 1.7, 1.9, 1.4, 0.5, 1.5, 3.0,
                       2.1, 1.2, 0.6, 1.2),
         n = 12)

# Inits:
list( beta0 = 0, beta1 = 1, beta2 = 1, tau = 1)  #for gamma prior on tau
```

```
################################################################################
##WinBUGS code for Part e - predicting z13:
#-------------------------------------------------

model{
  for( i in 1:n) {
    xmean[i] <- (x[i]-mean(x[]))
    z[i] ~ dnorm(mu[i] , tau)

   mu[i] <- beta0+ beta1*xmean[i] +beta2*(xmean[i]*xmean[i])

    }

#  beta0 ~ dnorm(0, 0.001)
#  beta1 ~ dnorm(0, 0.001)
# beta2 ~ dnorm(0, 0.001)

  beta0 ~ dflat()
  beta1 ~ dflat()
  beta2 ~ dflat()

 tau ~ dgamma(0.1, 0.1)   #  p_D = 3.203,   DIC = 25.365
 sigma <- 1/sqrt(tau)


}  #  end of BUGS code


#Data:
list(x = c( 102.4, 103.2, 101.9, 103.0, 101.2, 100.7,
102.5, 103.1, 102.8, 102.3, 101.9, 101.4, 100),
z=c(2.8, 3.1, 1.7, 1.9, 1.4, 0.5, 1.5, 3.0, 2.1,
1.2, 0.6, 1.2, NA),
        n = 13)

# Inits:
```

```
list( beta0 = 0, beta1 = 1, beta2 = 1, tau = 1) #for gamma prior on tau
###########################################################################
#R code for Part 2, fitting posteriors:
#for beta2:
b2post<-read.table("b2post-part-e.txt")
library(MASS)
#by looking at b2post from the -working environement-:
b2<-b2post$V2
fitdistr(b2,"normal")

#for the z13 prediction:
z13e<-read.table("part-e-prediction.txt")
library(MASS)
zz<-z13e$V2
fitdistr(zz,"normal")
###########################################################################
```