ISYE 6420: Bayesian Statistics **Spring 2020**

Homework #6

Instructor: Roshan Vengazhiyil, Brani Vidakovic *Name:* Nick Korbit, *gtID:* 903263968

Problem 1

a) Let x_1 be the time after injection and y be the temperature. We model y as a normally distributed variable with mean μ and precision τ:

$$y \sim \mathcal{N}(\mu, \tau)$$

We then model μ as a linear regression:

$$\hat{\mu} = \alpha + \beta x_1$$

We also set non-informative for α, β and tau:

$$\alpha \sim \mathcal{N}(0, 0.001)$$
$$\beta \sim \mathcal{N}(0, 0.001)$$
$$\tau \sim \mathcal{G}a(0.001, 0.001)$$

Given observed 10 data points we specify an OpenBUGS model as

```
# Training
for (i in 1:n) {
mu[i] <- alpha + beta*time[i]
temp[i] ~ dnorm(mu[i], tau)
}

# Priors
tau ~ dgamma(0.001, 0.001)
alpha ~ dnorm(0, 0.001)
beta ~ dnorm(0, 0.001)
```

In the observed data points we have one missing value for x_1. Assuming that missingness happened at random and noticing that x_1 are arranged in the ascending order, we can put a uniform distribution $\mathcal{U}(56, 70)$ on the missing 5th value:

```
time[5] ~ dunif(56, 70)
```

Then we specify calculation for R^2. Since we have 2 independent variables, we adjust SSE by $(n-2)$:

```
# R^2
sigma2 <- 1/tau
sse <- (n-2)*sigma2
for(i in 1:n){
ctemp[i] <- temp[i] - mean(temp[])
}
sst <- inprod(ctemp[], ctemp[])
BR2 <- 1 - sse/sst
BR2adj <- 1 - (n-1) * sigma2 / sst
```

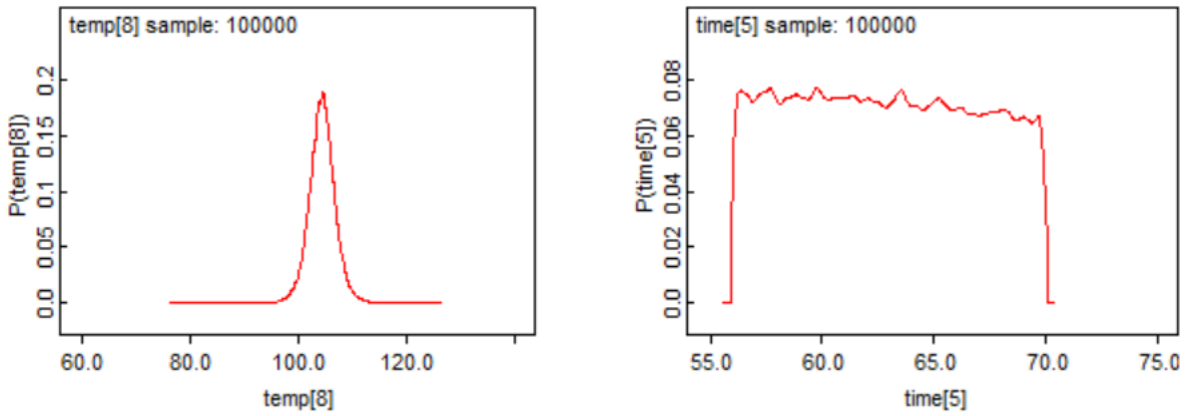
We initialize values for the priors as:

```
list(alpha = 0, beta = 0, tau=1)
```

Let's now run an OpenBUGS simulation. We burn the first 10000 observation and update the model with the next 100000 samples. First, we investigate stats:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
BR2	-0.2934	1.021	0.007075	-2.61	-0.04438	0.5685	10001	100000
BR2adj	-0.4551	1.149	0.007959	-3.061	-0.1749	0.5146	10001	100000
alpha	105.2	2.394	0.03242	100.2	105.3	109.6	10001	100000
beta	-0.009939	0.03742	5.083E-4	-0.07837	-0.01145	0.0691	10001	100000
mu[1]	105.0	1.571	0.02025	101.7	105.1	107.8	10001	100000
mu[2]	104.9	1.318	0.01621	102.1	105.0	107.3	10001	100000
mu[3]	104.7	0.9033	0.008187	102.9	104.8	106.4	10001	100000
mu[4]	104.7	0.79	0.004337	103.0	104.7	106.2	10001	100000
mu[5]	104.6	0.7872	0.001969	103.0	104.6	106.1	10001	100000
mu[6]	104.5	0.8484	0.003744	102.8	104.5	106.2	10001	100000
mu[7]	104.5	0.8822	0.004648	102.8	104.5	106.3	10001	100000
mu[8]	104.5	0.9418	0.006071	102.6	104.5	106.4	10001	100000
mu[9]	104.4	1.06	0.008521	102.4	104.4	106.6	10001	100000
mu[10]	104.3	1.534	0.01655	101.3	104.2	107.5	10001	100000
tau	0.264	0.1436	7.339E-4	0.06172	0.2388	0.6085	10001	100000
temp[8]	104.5	2.493	0.008733	99.52	104.5	109.5	10001	100000
time[5]	62.85	4.024	0.01407	56.33	62.78	69.63	10001	100000

We notice that both R^2 and R^2_{adj} have negative values, meaning that in principle we are better off with just setting an average of y_i as our prediction. We have also automatically inferred values for the missing data – x_5 and y_8, with y_8 averaging at 104.5:



The 95% credible set for the slope β is around $(-0.08, 0.07)$, so that 0 is inside the set.

b) We now expand our model with a new vaiable – time after injection squared. So that:

$$\hat{\mu} = \alpha + \beta_1 x_1 + \beta_2 x_1^2$$

Or in OpenBUGS terms:

```
# Training
for (i in 1:n) {
mu[i] <- alpha + beta1*time[i] + beta2*time2[i]
temp[i] ~ dnorm(mu[i], tau)
}

# Priors
tau ~ dgamma(0.001, 0.001)
alpha ~ dnorm(0, 0.001)
beta1 ~ dnorm(0, 0.001)
beta2 ~ dnorm(0, 0.001)
```

We also model missing values as uniform distributions:

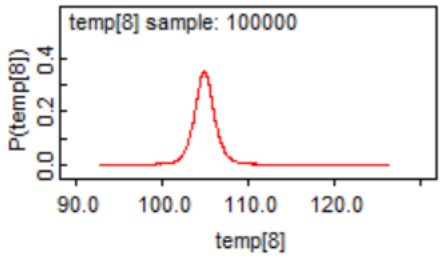
```
time[5] ~ dunif(56, 70)
time2[5] ~ dunif(3136, 4900)
```

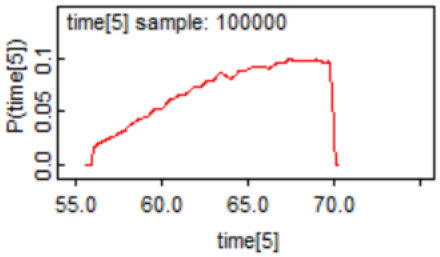
Since we have 3 independent variables, we adjust SSE by $(n-3)$. Let's now run an OpenBUGS simulation. We burn the first 10000 observation and update the model with the next 100000 samples. First, we investigate stats:

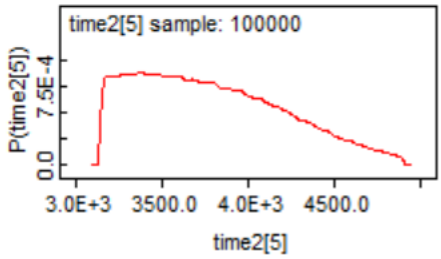
	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
BR2	0.6382	0.3369	0.006076	-0.1722	0.7288	0.9031	10001	100000
BR2adj	0.5348	0.4331	0.007812	-0.5071	0.6513	0.8755	10001	100000
alpha	97.05	2.724	0.1129	91.09	97.25	101.9	10001	100000
beta1	0.3232	0.09899	0.004272	0.1473	0.3156	0.5402	10001	100000
beta2	-0.002919	8.317E-4	3.525E-5	-0.004742	-0.002857	-0.001439	10001	100000
mu[1]	103.1	1.001	0.031	101.0	103.2	105.0	10001	100000
mu[2]	104.4	0.7138	0.0131	102.9	104.4	105.8	10001	100000
mu[3]	105.8	0.5852	0.01166	104.7	105.8	107.1	10001	100000
mu[4]	106.0	0.5971	0.0162	104.9	106.0	107.3	10001	100000
mu[5]	106.7	1.084	0.01162	104.5	106.7	108.8	10001	100000
mu[6]	105.4	0.5615	0.01373	104.3	105.3	106.6	10001	100000
mu[7]	105.2	0.5561	0.01228	104.2	105.2	106.4	10001	100000
mu[8]	104.9	0.5549	0.009623	103.8	104.8	106.0	10001	100000
mu[9]	104.2	0.5863	0.004265	103.1	104.2	105.4	10001	100000
mu[10]	101.2	1.131	0.02792	98.86	101.2	103.4	10001	100000
tau	1.015	0.6232	0.008689	0.1847	0.8907	2.562	10001	100000
temp[8]	104.9	1.362	0.01005	102.2	104.9	107.7	10001	100000
time[5]	64.42	3.586	0.02903	57.15	64.8	69.74	10001	100000
time2[5]	3783.0	423.1	3.86	3166.0	3732.0	4685.0	10001	100000

We notice that means for both R^2 and R^2_{adj} are much higher now, 0.64 and 0.53 respectively. So including a time squared feature is a good idea and that alone significantly boosts model performance.

We have also automatically inferred values for the missing data – x_5, x_5^2 and y_8, with y_8 averaging at 104.9:







The 95% credible set for the slope β_1 is around $(0.15, 0.54)$ and $(-0.005, -0.001)$ for beta β_2. So that 0 is not inside the sets.

Note: the full OpenBUGS code is available at *rabbits1.odc* and *rabbits2.odc* in the attached archive.

Problem 2

Assuming observed times are exponentially distributed with non-informative priors, let's first define the OpenBUGS model:

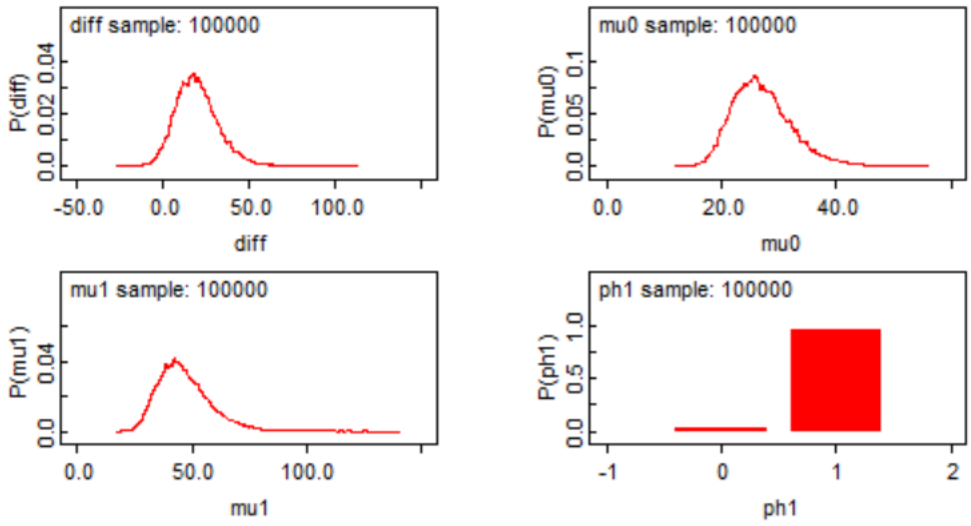
```
for(i in 1:n) {
time[i] ~ dexp(lambda[i])I(time.cen[i],)
lambda[i] <- exp(beta0 + beta1*group[i])
}

beta0 ~ dnorm(0.0, 0.0001)
beta1 ~ dnorm(0.0, 0.0001)
```

We also can derive the estimators for the expected time until recurrence (the bigger value, the better treatment is). μ_0 is the mean time for placebo (no treatment) and μ_1 is the mean time for chemotherapy:

```
mu0 <- exp(-beta0)
mu1 <- exp(-beta0-beta1)
diff <- mu1-mu0
ph1 <- step(diff)
```

Let's now run an OpenBUGS simulation. We start with burning the first 10000 observation and update the model with the next 100000 samples. First, we investigate the stats:

	mean	sd	MC_error	val2.5pc	val5.0pc	median	val95.0pc	val97.5pc	start	sample
diff	19.74	12.8	0.1862	-1.816	1.172	18.56	42.37	48.55	10001	100000
mu0	27.09	5.231	0.07825	18.71	19.72	26.45	36.58	39.21	10001	100000
mu1	46.84	11.74	0.1319	29.25	31.21	45.07	68.22	74.37	10001	100000
ph1	0.9621	0.1909	0.002158	0.0	1.0	1.0	1.0	1.0	10001	100000



We notice that the expected time until cancer recurrence is far more larger in case of the chemotherapy – 46.8 months vs 27.1 months (placebo treatment). Constructing a 90% credible set for the difference in means $\mu_1 - \mu_0$ will yield us $(1.17, 36.6)$.

Next, we test the hypothesis $H_1 : \mu_1 - \mu_0$. The OpenBUGS simulation estimates the probability of H_1 as 96.2%, so that we accept H_1 hypothesis.

Since 90% credible set for the difference in means is always positive and the probability of $H_1 : \mu_1 - \mu_0$ hypothesis is 96.2%, we can conclude that chemotherapy is beneficial for the patient. The treatment extends

the time to recurrence by 19 months on average.

Note: the full OpenBUGS code is available at *bladderc0.odc* in the attached archive.

References

[1] Engineering Biostatistics: An Introduction using MATLAB and WinBUGS. Brani Vidakovic - Wiley Series in Probability and Statistics.