ISYE 6420: Final

aelhabr3

1. Time to Second Birth.

Instructions

…

Response

We fit the following linear regression model.

$$\text{time } = \beta_0 + \beta_1 \text{ death } + \beta_2 \text{ mage } .$$

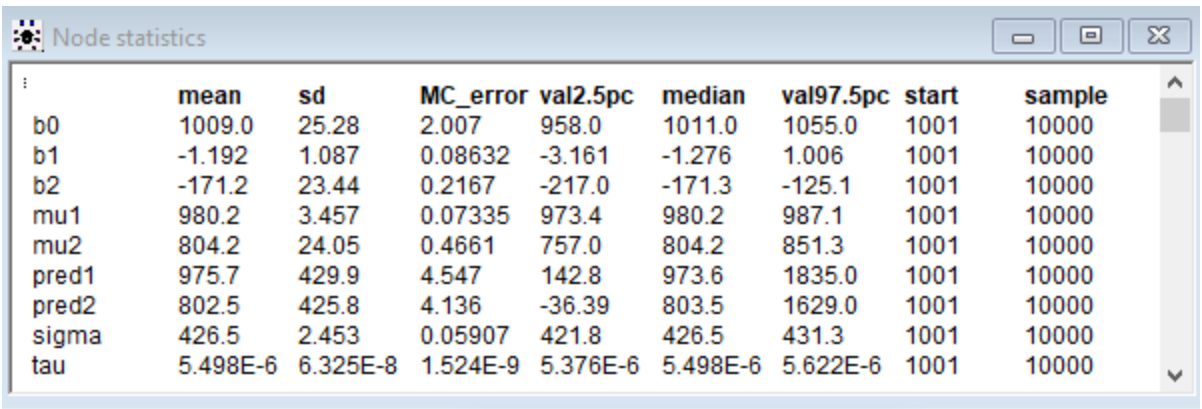
The model code below is written so as to answer all parts of the question. The full model can be found in “q1.odc”. (Note that we need to do anything “special” for the indicator variable `death`, i.e. transform its values to a 1-2 binary pair. It was found that the results are consistent either way.)

```
library(tidyverse)
```

Below is the model code. The data is excerpted for the sake of readability. The full model can be found in “q1.odc”.

```
model{
  for(i in 1:n) {
    mu[i] <- b0 + b1 * mage[i] + b2 * death[i]
    time[i] ~ dnorm(mu[i], tau)
  }

  # priors
  b0 ~ dnorm(0, 0.001)
  b1 ~ dnorm(0, 0.001)
  b2 ~ dnorm(0, 0.001)
  tau ~ dgamma(0.001, 0.001)
  sigma <- 1 / sqrt(tau)

  mu1 <- b0 + b1 * (24) + b2 * (0)
  pred1 ~ dnorm(mu1, tau)

  mu2 <- b0 + b1 * (28) + b2 * (1)
  pred2 ~ dnorm(mu2, tau)
}

# inits
list(b0 = -1, b1 = -1, b2 = 0, tau = 1, pred1 = 0, pred2 = 0)


# data
...
```

Below is a summary of the output.



	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b0	1009.0	25.28	2.007	958.0	1011.0	1055.0	1001	10000
b1	-1.192	1.087	0.08632	-3.161	-1.276	1.006	1001	10000
b2	-171.2	23.44	0.2167	-217.0	-171.3	-125.1	1001	10000
mu1	980.2	3.457	0.07335	973.4	980.2	987.1	1001	10000
mu2	804.2	24.05	0.4661	757.0	804.2	851.3	1001	10000
pred1	975.7	429.9	4.547	142.8	973.6	1835.0	1001	10000
pred2	802.5	425.8	4.136	-36.39	803.5	1629.0	1001	10000
sigma	426.5	2.453	0.05907	421.8	426.5	431.3	1001	10000
tau	5.498E-6	6.325E-8	1.524E-9	5.376E-6	5.498E-6	5.622E-6	1001	10000

We see that the estimated model is as follows.

$$\text{time } = (1009) + (-1.192)\ \text{death } + (-171.2)\text{mage}\ .$$

a

From the output shown above, we see that the slope (i.e. the posterior estimate of the mean) of β_2 (for death) is -171.2. Its 95% CS is [-217, -125.1]. This variable is significant (in this case, significantly negative) because its 95% CS does not contain zero. (Yes, this is not exactly the most

robust manner in which to determine significance, but it is usually a very good heuristic.)

b

We observe that the posterior estimate of the mean of β_1 (for mage) is -1.192, and that its 95% CS is [-3.161, 1.006]. We may say that this variable is not significant (with respect to its relationship with time) because its 95% CS is not completely negative.

c

From the output shown above, we see that the predicted time between births for Helga (corresponding to `pred1`) is 975.7 with a 95% CS of [142.8, 1835]. Notably, this CS is very wide, but it does not include zero, so we can be somewhat confident in it (and be mindful of its imprecision).

d

From the output shown above, we see that the predicted time between births for Emma (corresponding to `pred2`) is 802.5 with 95% CS is [-36.39, 1629]. As with the prediction for Helga, this prediction's CS is very wide. It even includes zero (which does not make sense in the context of this problem)! Thus, we should be wary of the lack of robustness of this prediction.

2. Tasmanian Clouds.

Instructions

…

Response

We implement a comprehensive Bayesian additive two-way ANOVA analysis on the response `diff` to estimate and test the effects of factors `season` and `seeded`.

a

Below is the model code, not including any interaction term between `seeded` and `season`. The full model can be found in "q2a.odc".

```
model{
  for(i in 1:n) {
    diff[i] ~ dnorm(mu[i], tau)
    mu[i] <- mu0 + alpha[seeded[i]] + beta[season[i]]
  }

  # STZ (sum-to-zero) constraints
  alpha[1] <- -sum(alpha[2:za])
  beta[1] <- -sum(beta[2:zb])

  # priors
  mu0 ~ dnorm(0, 0.0001)
  for(i in 2:za) {
    alpha[i] ~ dnorm(0, 0.0001)
  }
  for(i in 2:zb) {
    beta[i] ~ dnorm(0, 0.0001)
  }
  tau ~ dgamma(0.001, 0.001)
  sigma <- 1 / sqrt(tau)

  # pairwise comparisons
  for(i in 1:(za-1)) {
    for(j in (i+1):za) {
      ca[i, j] <- alpha[i] - alpha[j]
    }
  }
  for(i in 1:(zb-1)) {
    for(j in (i+1):zb) {
      cb[i, j] <- beta[i] - beta[j]
    }
  }
}

# inits
list(
  mu0 = 0,
  alpha = c(NA, 0),
  beta = c(NA, 0, 0, 0),
  tau = 1
)

# data
...
```

Note the following about the model.

- It is very similar to that of the “simvastatin.odc” example.
- The “U” and “S” for `seeded` have been re-coded as 1 and 2, and “Spring”, “Summer”, “Autumn”, and “Winter” for `season` have been re-coded as 1, 2, 3, and 4, respectively.

Below is a summary of the output.

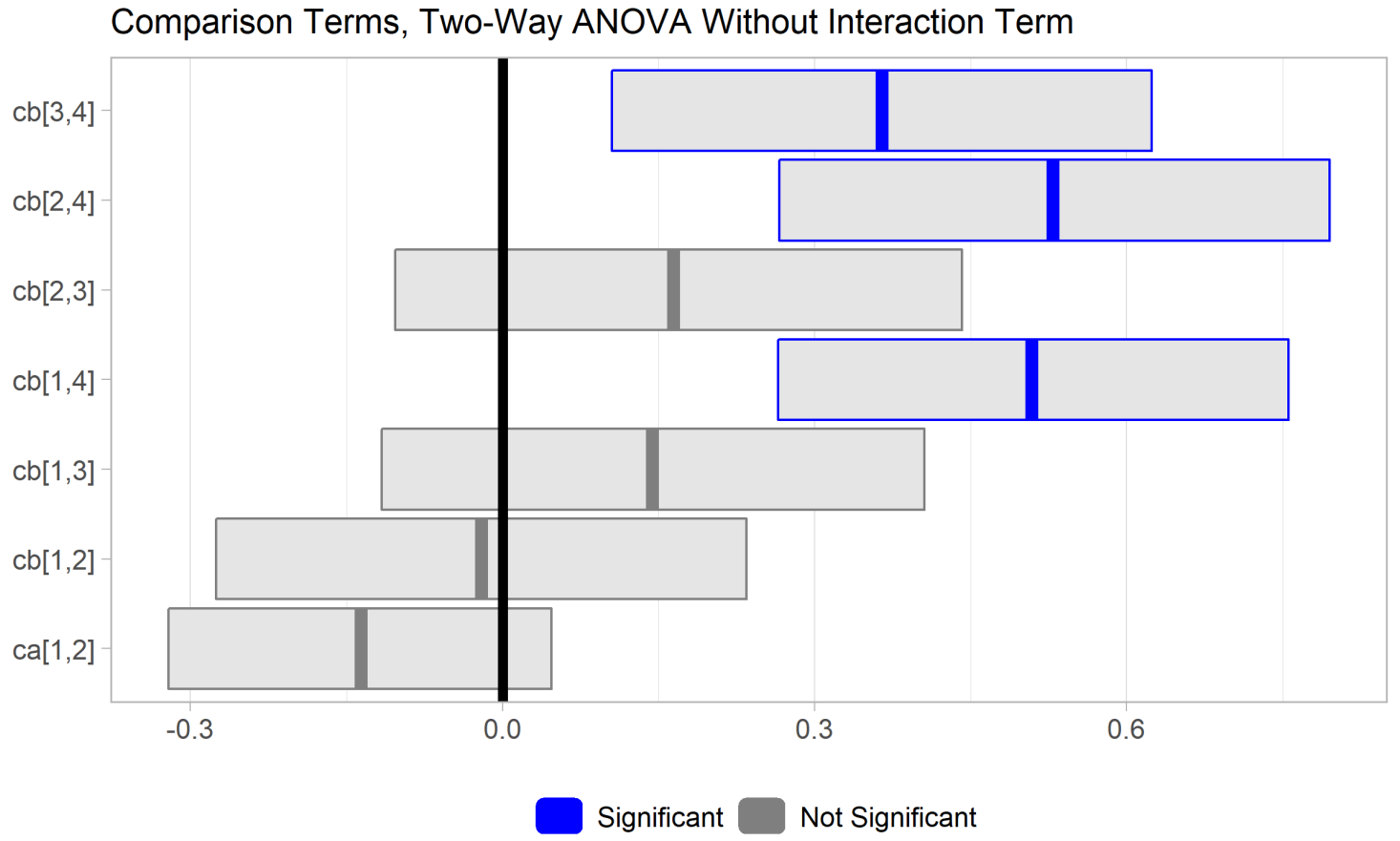
Node statistics

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
alpha[1]	-0.06797	0.04672	4.456E-4	-0.1607	-0.0677	0.02348	1001	10000
alpha[2]	0.06797	0.04672	4.456E-4	-0.02335	0.0677	0.1607	1001	10000
beta[1]	0.1582	0.0771	5.895E-4	0.008243	0.1586	0.3108	1001	10000
beta[2]	0.1785	0.08417	0.001235	0.01736	0.1778	0.3478	1001	10000
beta[3]	0.014	0.08388	0.001186	-0.152	0.01463	0.1784	1001	10000
beta[4]	-0.3507	0.07817	0.001114	-0.5039	-0.3506	-0.1979	1001	10000
ca[1,2]	-0.1359	0.09344	8.911E-4	-0.3213	-0.1354	0.04696	1001	10000
cb[1,2]	-0.02036	0.1311	0.001438	-0.2751	-0.01852	0.2347	1001	10000
cb[1,3]	0.1442	0.1316	0.001467	-0.1161	0.1443	0.4056	1001	10000
cb[1,4]	0.5089	0.1243	0.001299	0.2651	0.5089	0.7558	1001	10000
cb[2,3]	0.1645	0.1399	0.002091	-0.1033	0.1638	0.4418	1001	10000
cb[2,4]	0.5292	0.1333	0.002043	0.266	0.5278	0.7949	1001	10000
cb[3,4]	0.3647	0.132	0.001902	0.1053	0.3651	0.6238	1001	10000
mu0	0.007276	0.04686	4.827E-4	-0.0845	0.007969	0.09797	1001	10000
sigma	0.4824	0.03429	3.448E-4	0.422	0.4801	0.5565	1001	10000
tau	4.361	0.6111	0.006073	3.229	4.338	5.617	1001	10000

In the output above we are primarily interested in the comparison terms `ca` and `cb`. The `ca` terms quantify the difference in paired `seeded` values—either seeded “S” or unseeded “U”. Here, there are only two possible values for `seeded`, so there is only one `ca` term, i.e. `ca[1, 2]`. (Really, an array data structure is not needed for the comparison of `seeded` terms, but it is used anyways to maintain consistency with the `season` comparison terms, of which there are more than one.) The `cb` terms quantify the difference between pairs of `season` values. For example, `cb[1, 2]` quantifies the difference between “Spring” and “Summer”; and `cb[1, 3]` quantifies the difference between “Spring” and “Autumn”; etc.

We consider those comparison terms with a 95% CS that is either completely positive or completely negative to be significant. (If we see that `val2.5pc > 0`, then we say that the comparison is significantly different in a positive way, favoring the first term; similarly, If we see that `val97.5pc < 0`, then we say that the comparison is significantly different in a negative way, favoring the second term.)

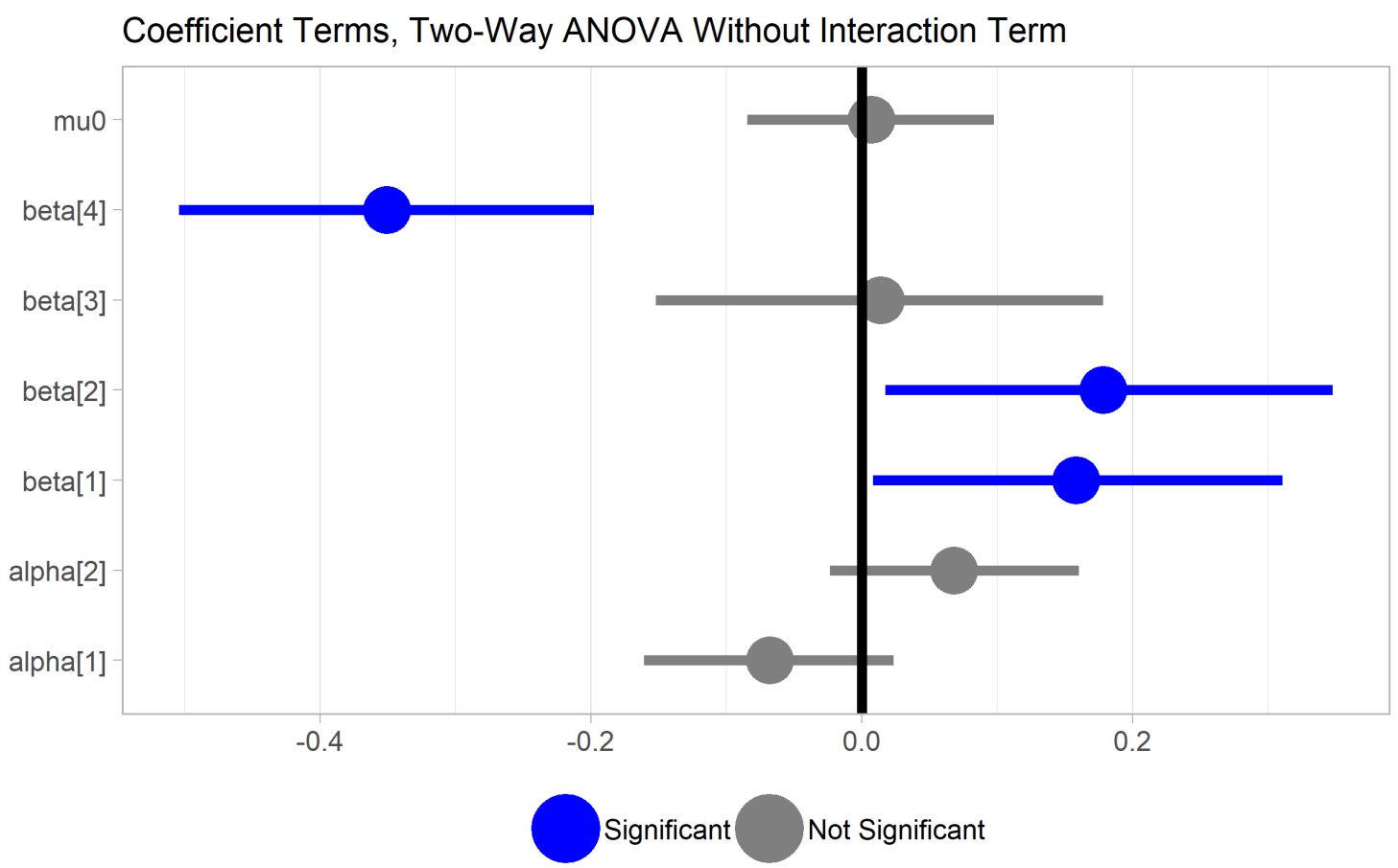
To assist with interpreting the comparison terms, below is a plot illustrating the posterior means and 95% CS of the two-way paired comparisons. (The interior crossbars illustrate the mean and the box bounds illustrate the 95% CS range.)



We observe the following:

- **The three `season` comparison terms involving "Winter" are significantly positive: (1) `cb[1,4]` (i.e. the difference between "Spring" and "Winter"); (2) `cb[2 4]` (i.e. the difference between "Summer" and "Winter"); (3) `cb[3,4]` (i.e. the difference between "Autumn" and "Winter").**
- **Two terms have negative posterior means but are not significant: `ca[1,2]` and `cb[1,2]`.**
- **The remaining two terms—involving `season` comparisons with "Autumn"—have positive posterior means but are not significant: `cb[1,3]` and `cb[2,3]`.**

Of course, we should also look at the estimates of the `alpha` and `beta` terms, corresponding to `seeded` and `season` respectively.



We observe the following.

- **The `alpha` terms are insignificant because the CS includes zero. Also, note that `alpha[1]` and `alpha[2]` —representing "U" (unseeded) and "S" (seeded) respectively—are essentially inverses of one another.**
- **Three `beta` terms are significant. `beta[1]` and `beta[1]` —representing "Spring" and "Summer" respectively—are significantly positive; `beta[4]` — representing "Winter—is significantly negative.**
- **`beta[3]` —representing "Autumn"—is close to zero and is insignificant. Still, it is significantly different from `beta[4]`, as indicated before with `cb[3,4]`. (Note that its CS does not overlap with that of `beta[4]`.)**
- **The intercept term `mu0` is nearly zero and is insignificant.**

b

Below is the code for the model including the interaction term (i.e. `alpha.beta`). The full model can be found in "q2b.odc".

```
model{
  for(i in 1:n) {
    diff[i] ~ dnorm(mu[i], tau)
    mu[i] <- mu0 + alpha[seeded[i]] + beta[season[i]] + alpha.beta[seeded[i], season[i]]
  }

  # STZ (sum-to-zero) constraints
  alpha[1] <- -sum(alpha[2:za])
  beta[1] <- -sum(beta[2:zb])
  for(i in 1:za) {
    alpha.beta[i, 1] <- -sum(alpha.beta[i, 2:zb])
  }
  for(i in 2:zb) {
    alpha.beta[1, i] <- -sum(alpha.beta[2:za, i])
  }

  # priors
  mu0 ~ dnorm(0, 0.0001)
  for(i in 2:za) {
    alpha[i] ~ dnorm(0, 0.0001)
  }
  for(i in 2:zb) {
    beta[i] ~ dnorm(0, 0.0001)
  }
  for(i in 2:za) {
    for(j in 2:zb) {
      alpha.beta[i, j] ~ dnorm(0, 0.0001)
    }
  }
  tau ~ dgamma(0.001, 0.001)
  sigma <- 1 / sqrt(tau)

  # pairwise comparisons
  for(i in 1:(za-1)) {
    for(j in (i+1):za) {
      ca[i, j] <- alpha[i] - alpha[j]
    }
  }
  for(i in 1:(zb-1)) {
    for(j in (i+1):zb) {
      cb[i, j] <- beta[i] - beta[j]
    }
  }
}

# inits
list(
  mu0 = 0,
  alpha = c(NA, 0),
  beta = c(NA, 0, 0, 0),
  alpha.beta = structure(.Data = c(NA, NA, NA, NA, NA, 0, 0, 0), .Dim = c(2, 4)),
  tau = 1
)
```

```
# data
...
```

Note that this code is very similar to that from (a) above, with additions made to include the interaction term `alpha.beta`.

Below is a summary of the output.

Node statistics

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
alpha[1]	-0.07068	0.04626	4.684E-4	-0.1611	-0.07062	0.0211	1001	10000
alpha[2]	0.07068	0.04626	4.684E-4	-0.02099	0.07064	0.1612	1001	10000
alpha.beta[1,1]	0.07164	0.07566	4.816E-4	-0.07649	0.07194	0.2179	1001	10000
alpha.beta[1,2]	0.1492	0.08277	0.001164	-0.0127	0.1489	0.3109	1001	10000
alpha.beta[1,3]	-0.1617	0.08291	0.001194	-0.3254	-0.1614	4.982E-4	1001	10000
alpha.beta[1,4]	-0.05911	0.07778	0.001158	-0.2135	-0.05879	0.09407	1001	10000
alpha.beta[2,1]	-0.07164	0.07566	4.816E-4	-0.2175	-0.07193	0.07711	1001	10000
alpha.beta[2,2]	-0.1492	0.08277	0.001164	-0.3107	-0.1489	0.01271	1001	10000
alpha.beta[2,3]	0.1617	0.08291	0.001194	-4.62E-4	0.1614	0.3257	1001	10000
alpha.beta[2,4]	0.05911	0.07778	0.001158	-0.09403	0.0588	0.2136	1001	10000
beta[1]	0.158	0.07563	5.178E-4	0.01126	0.1586	0.3046	1001	10000
beta[2]	0.1769	0.08263	0.001101	0.01502	0.1773	0.3409	1001	10000
beta[3]	0.01344	0.08263	0.001181	-0.1499	0.01315	0.1774	1001	10000
beta[4]	-0.3484	0.07902	0.001128	-0.5045	-0.3484	-0.193	1001	10000
ca[1,2]	-0.1414	0.09252	9.367E-4	-0.3222	-0.1412	0.04219	1001	10000
cb[1,2]	-0.01889	0.129	0.001345	-0.2739	-0.01923	0.2311	1001	10000
cb[1,3]	0.1446	0.1283	0.00136	-0.1045	0.1438	0.3947	1001	10000
cb[1,4]	0.5064	0.1241	0.00125	0.261	0.5046	0.7499	1001	10000
cb[2,3]	0.1635	0.137	0.001922	-0.1031	0.1634	0.4335	1001	10000
cb[2,4]	0.5253	0.1324	0.001869	0.2686	0.526	0.7902	1001	10000
cb[3,4]	0.3618	0.133	0.002047	0.09833	0.3618	0.6275	1001	10000
mu0	0.006773	0.04677	4.839E-4	-0.08487	0.00609	0.09864	1001	10000
sigma	0.4751	0.03392	3.335E-4	0.4131	0.4733	0.5478	1001	10000
tau	4.498	0.6375	0.006222	3.332	4.465	5.862	1001	10000

We observe the that all interaction terms are insignificant. (Each interaction term's CS includes zero.) Also, we see that the posterior means and 95% CS of all the `seeded` and `season` terms are only trivially affected. (There is no use in re-creating the same plots shown in (a).)

3. Miller Lumber Company Customer Survey.

Instructions

…

Response

We fit the following Poisson regression model.

$$\begin{aligned}\log(\text{customers}) \quad &\sim \quad \beta_0 + \beta_1\ \text{hunits} \ + \beta_2\ \text{aveinc} + \\ &\qquad \beta_3\ \text{aveage} \ + \beta_4\ \text{distcomp} \ + \beta_5\ \text{diststore}\end{aligned}$$

The model code below is written so as to answer parts (a) and (c). (In particular, `lambdastar` and `pred1` correspond to the mean response and prediction asked for in (c). The full model can be found in "q3ac.odc".

```
model {
  for(i in 1:n){
    customers[i] ~ dpois(lambda[i])
    log(lambda[i]) <- b0 + b1 * hunits[i] + b2 * aveinc[i] + b3 * avehage[i] + b4 * distcomp[i] + b5 * dists
tore[i]
  }

  # prediction
  log(lambdastar) <- b0 + b1 * (0.72) + b2 * (7) + b3 * (6) + b4 * (4.1) + b5 * (8)
  pred1 ~ dpois(lambdastar)

  # priors
  b0 ~ dnorm(0, 0.01)
  b1 ~ dnorm(0, 0.01)
  b2 ~ dnorm(0, 0.01)
  b3 ~ dnorm(0, 0.01)
  b4 ~ dnorm(0, 0.01)
  b5 ~ dnorm(0, 0.01)
}

# inits
list(b0 = 0, b1 = 0, b2 = 0, b3 = 0, b4 = 0, b5 = 0, pred1 = 0)


# data
# ...
```

Note the following about the model.

- As advised by the announcement made on Canvas, the values of the variable hunits (for home units) has been scaled down by a factor of 1,000 (i.e. divided by 1,000) and the values of the variable aveinc (for salaries) has been scaled down by a factor of 10,000.
- Following a second recommendation from the Canvas announcement, we make the β terms not "too noninformative", defining them as `dnorm(0, 0.01)` instead of something like `dnorm(0, 0.000001)`.

Below is a summary of the output.

Node statistics

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b0	2.941	0.208	0.002764	2.531	2.943	3.348	1001	10000
b1	0.6072	0.1421	0.00174	0.3245	0.6064	0.8827	1001	10000
b2	-0.1173	0.0209	2.665E-4	-0.159	-0.1171	-0.0767	1001	10000
b3	-0.003717	0.001787	1.984E-5	-0.007173	-0.00373	-1.959E-4	1001	10000
b4	0.1684	0.02581	3.061E-4	0.1174	0.1684	0.2187	1001	10000
b5	-0.1288	0.01629	2.124E-4	-0.1603	-0.1291	-0.09686	1001	10000
lambdastar	9.007	0.6621	0.008573	7.758	8.982	10.36	1001	10000
pred1	9.025	3.114	0.03013	4.0	9.0	16.0	1001	10000

a

As shown in the output above, **the coefficient estimates (i.e. the β posterior means) are as follows.**

- β_1 = 0.6072
- β_2 = -0.1173
- β_3 = -0.003717
- β_4 = 0.1684
- β_5 = -0.1288

All terms seem to be significant, given that their 95% CS are either completely negative or completely positive.

b

In order to identify the “best” two covariates from the set of five, we implement stochastic search variable selection (SSVS), as discussed in Unit 9 and exemplified in “Haldssvs” from the Unit 9 exercises. The theory behind this process is as follows (specified for Poisson regression instead of traditional linear regression).

$$\begin{aligned} \log(\lambda) &= \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k, \\ \beta_i &= \delta \alpha_i, \\ \alpha_i &\sim \mathcal{N}orm(0, \tau), \\ \delta_i &\sim \mathcal{B}ern(p_i). \end{aligned}$$

where p_i is the probability that the variable x_i is in the model. Note that the indicator δ_i terms are used simply to determine whether a β_i term is included in a given model. That is, if $\delta_i = 0$, then β_i is not included; and if $\delta_i = 1$, then β_i is included in the model and its estimate is that of α_i. In the end, we choose the model with the greatest a posteriori probability.

The (initial) model code is shown below. (This is actually a two-step process, given the ambiguity of the initial results.) The full model files are “q3b1.odc” and “q3b2.odc”.

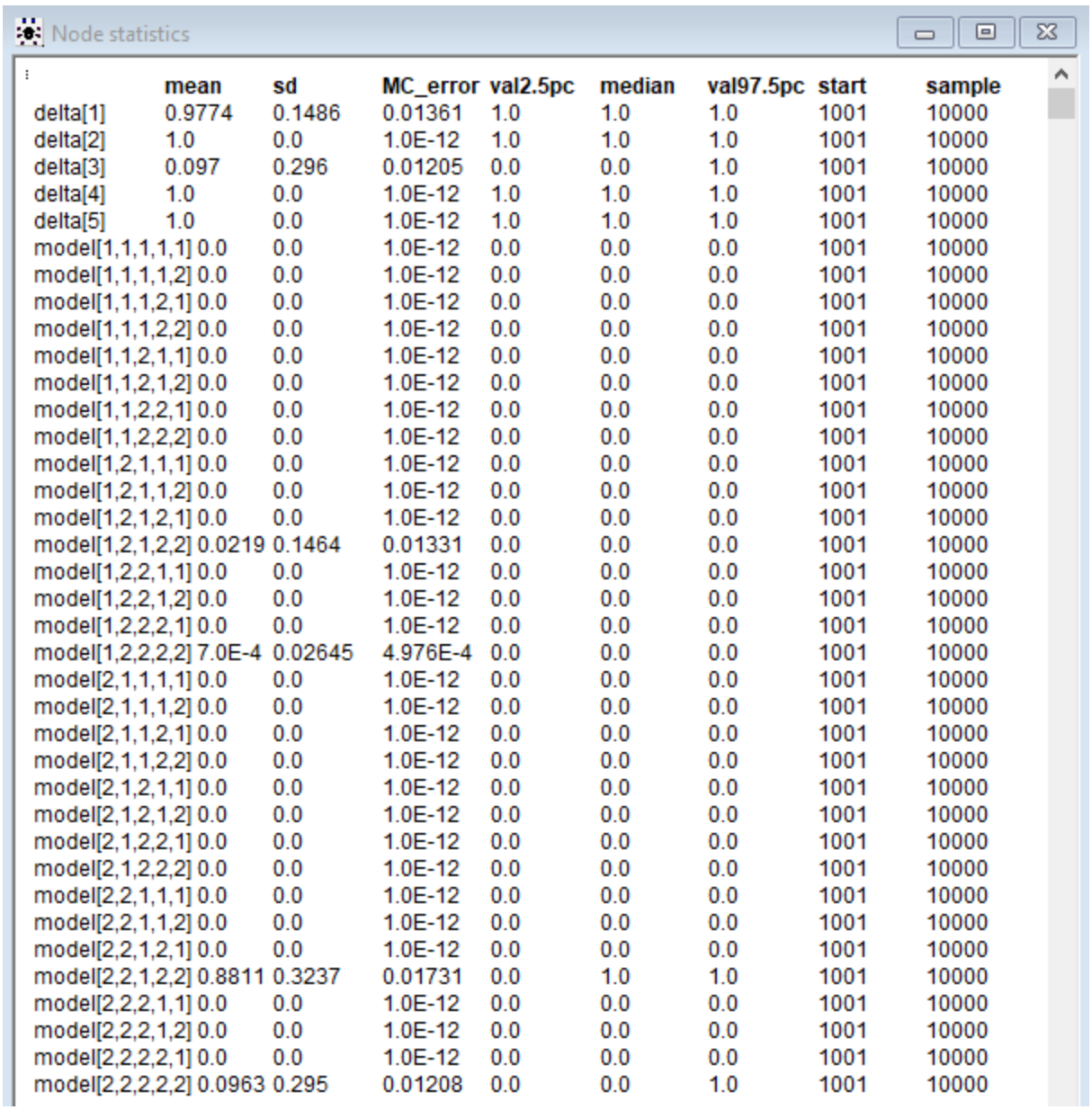
```
model {
  for(i in 1:n){
    y[i] ~ dpois(lambda[i])
    log(lambda[i]) <- int + inprod(x.c[i,], beta[])
  }
  int ~ dnorm(0,0.001)

  # SSVS prior
  for(j in 1:p){
    delta[j] ~ dbern(0.5)
    alpha[j] ~ dnorm(0, 0.1)
    beta[j] <- delta[j] * alpha[j]
  }

  # model probabilities
  for(j1 in 1:2){
    for(j2 in 1:2){
      for(j3 in 1:2){
        for(j4 in 1:2){
          for(j5 in 1:2){
            model[j1, j2, j3, j4, j5] <- equals(delta[1], j1 - 1) * equals(delta[2], j2 - 1) * equals(delta
[3], j3 - 1) * equals(delta[4], j4 - 1) * equals(delta[5], j5 - 1)
          }
        }
      }
    }
  }

  # data centering
  for(i in 1:n){
    for(j in 1:p){
      x.c[i,j] <- (x[i, j] - mean(x[, j])) / sd(x[, j])
    }
  }
}

# inits
list(
  delta = c(1, 1, 1, 1, 1),
  alpha = c(0, 0, 0, 0, 0),
  int = 1
)

# data
...
```

Note the following about the model.

- For the sake of convenience, we rename the features `hunits`, …, `diststore` to be `x[,1]`, …, `x[,5]` with corresponding β estimates `b1`, …, `b5`.

- For our implementation, we define $\tau = 0.01$ (shown in the stochastic process equations above) so that the α_i estimates are relatively noninformative. (This maintains the premise that we should make our priors "not too noninformative".)
- We center the data—creating the set of `x.c[]` variables. This is a standard pre-processing step in most variable selection procedures.
- We use the values `2` and `1` instead of `1` and `0` as our binary pair for included and non-included terms. (This is convenient for OpenBUGs.)

Below is a summary of the output.


Node statistics

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
delta[1]	0.9774	0.1486	0.01361	1.0	1.0	1.0	1001	10000
delta[2]	1.0	0.0	1.0E-12	1.0	1.0	1.0	1001	10000
delta[3]	0.097	0.296	0.01205	0.0	0.0	1.0	1001	10000
delta[4]	1.0	0.0	1.0E-12	1.0	1.0	1.0	1001	10000
delta[5]	1.0	0.0	1.0E-12	1.0	1.0	1.0	1001	10000
model[1,1,1,1,1]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[1,1,1,1,2]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[1,1,1,2,1]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[1,1,1,2,2]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[1,1,2,1,1]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[1,1,2,1,2]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[1,1,2,2,1]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[1,1,2,2,2]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[1,2,1,1,1]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[1,2,1,1,2]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[1,2,1,2,1]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[1,2,1,2,2]	0.0219	0.1464	0.01331	0.0	0.0	0.0	1001	10000
model[1,2,2,1,1]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[1,2,2,1,2]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[1,2,2,2,1]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[1,2,2,2,2]	7.0E-4	0.02645	4.976E-4	0.0	0.0	0.0	1001	10000
model[2,1,1,1,1]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[2,1,1,1,2]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[2,1,1,2,1]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[2,1,1,2,2]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[2,1,2,1,1]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[2,1,2,1,2]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[2,1,2,2,1]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[2,1,2,2,2]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[2,2,1,1,1]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[2,2,1,1,2]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[2,2,1,2,1]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[2,2,1,2,2]	0.8811	0.3237	0.01731	0.0	1.0	1.0	1001	10000
model[2,2,2,1,1]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[2,2,2,1,2]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[2,2,2,2,1]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[2,2,2,2,2]	0.0963	0.295	0.01208	0.0	0.0	1.0	1001	10000

There is quite a bit to look at here. To provide some guidance for interpreting these results, take the following examples.

- Note that `model[2, 2, 1, 2, 2]` has a posterior `mean` estimate of 0.8811. This tells us that the model with `x[,1]` (for `hunits`, corresponding to the first `2`), with `x[,2]` (for `aveinc`, corresponding to the second `2`), without `x[,3]` (for `avehage`, corresponding the `1`), with `x[,4]` (for `distcomp`, corresponding to the third `2`) and with `x[,5]` (for `diststore`, corresponding to the fourth `2`) is visited 88% of the time by the MCMC procedure.
- Note that `delta[1]` has a posterior `mean` estimate of 0.9774. This means that the `x1` term (for `hunits`) is selected 98% of the time by the MCMC procedure.

Overall, we observe that there are many model combinations with posterior `mean` estimates (of probability) equal to 0. Since the only combinations that have nonzero probabilities include three or more terms, we must repeat the SSVS process. Since the `delta` terms corresponding to `x[,2]`, `x[,4]`, `x[,5]` equally have the highest posterior `mean` s, we reduce our data set to just these three terms.

Since the model code is nearly identical, it is not shown here. The only non-trivial change to note is that the terms `x[,2]`, `x[,4]`, `x[,5]` are "re-indexed" to `x[,1]`, `x[,2]`, `x[,3]`. Please consult the "q3b2.odc" file to see the implementation.

Below are the results of this second SSVS procedure.

Node statistics

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
delta[1]	0.9047	0.2936	0.02488	0.0	1.0	1.0	1001	10000
delta[2]	1.0	0.0	1.0E-12	1.0	1.0	1.0	1001	10000
delta[3]	1.0	0.0	1.0E-12	1.0	1.0	1.0	1001	10000
model[1,1,1]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[1,1,2]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[1,2,1]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[1,2,2]	0.0953	0.2936	0.02488	0.0	0.0	1.0	1001	10000
model[2,1,1]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[2,1,2]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[2,2,1]	0.0	0.0	1.0E-12	0.0	0.0	0.0	1001	10000
model[2,2,2]	0.9047	0.2936	0.02488	0.0	1.0	1.0	1001	10000

We have exactly one two-covariate model— `model[1,2,2]` —with a nonzero posterior `mean`, which makes our choice of two-covariate model straightforward. (Of course, if we were able to choose three variables, then we could have chosen all three.). **The terms included in this "best" model (according to SSVS) are `distcomp` and `diststore`.**

c

The mean response for the new data (`hunits = 720`, `aveinc = 70000`, `aveage = 6`, `distcomp = 4.1`, and `diststore = 8`) correspond to `lambdastar` from the output shown above (a). Similarly, the prediction corresponds to `pred1`.

There we see that the mean response is 9.007 with a 95% CS of [7.758, 10.36]. The prediction is 9.025 with a 95% CS of [4, 16]. Note that the prediction has a larger CS than that of the mean response. This is because the prediction's CS accounts for additional prediction—the uncertainty of a point estimate prediction. (We explored this concept in question 2(c) of HW6.)