Computational Biology Group

D BSSE

ETH zürich

Introduction to Bayesian Statistics with R

8: Exercise solutions

Jack Kuipers

29 November 2022

First we load the tidyverse, brms and set a seed.

```
library(tidyverse); options(dplyr.summarise.inform = FALSE) # suppress summarise warnings
library(brms)
set.seed(42)
```

Exercise 8.1 - Bayesian logistic regression

For Bayesian modelling with `brms` *we can use the* `brm()` *function with* `family = binomial`*, but with a somewhat different syntax for the formula. We separate the number of occurrences from the number of trials (input into the* `trials` *function) with* `|` *and*
`formula = cancers | trials(total) ~ ...`

- *Fit a Bayesian logistic regression model of cancer incidence with* `age_s`*,* `sex`*,* `race`*, and* `registry` *as explanatory variables (no interactions). Include* `I(age_s^2)` *to add a quadratic* `age_s` *term to the model.*
- *Check the model convergence and examine the regression coefficients.*
- *What is the posterior distribution of the probability of having cancer for a 75 year-old Black female from registry 27?*

After we load the data

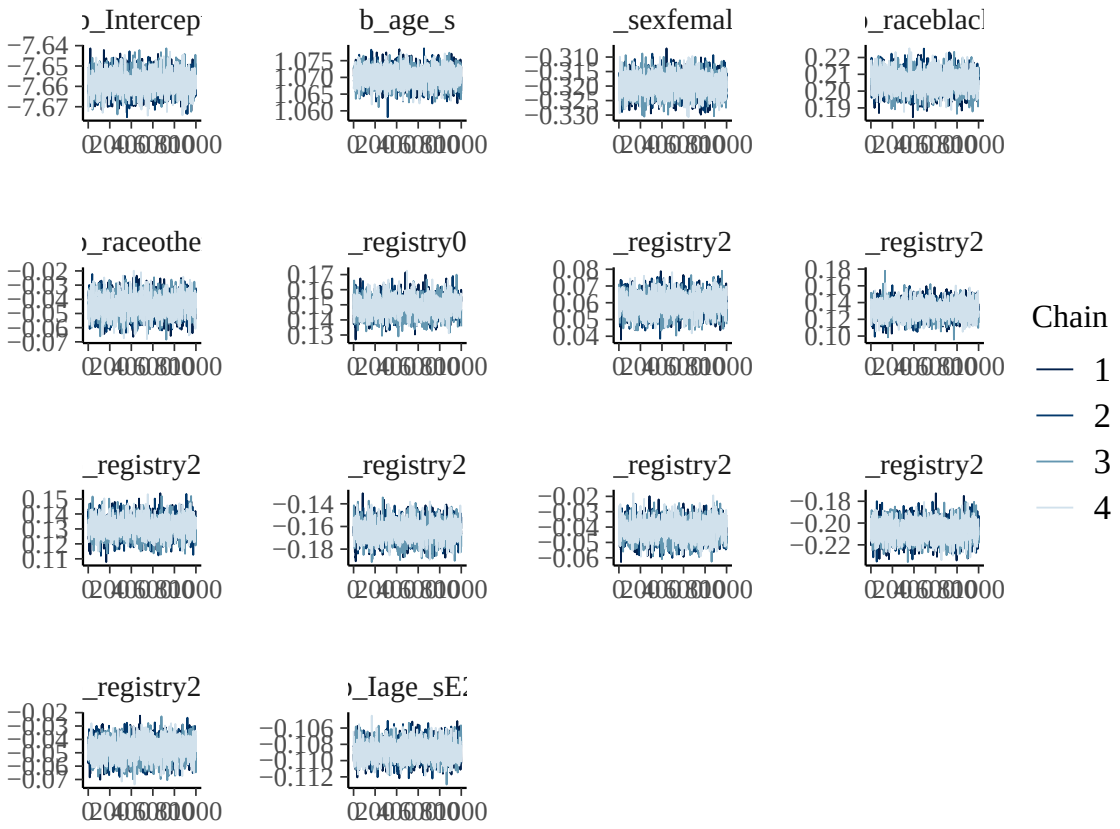
```
load("./data/CRC_Data.RData")
```

we run the model with the following syntax (with our helper function in the background)

```
brmfit_ex8 <- run_model(brm(bf(cancers | trials(total) ~ age_s + sex + race + registry +
  I(age_s^2)), family = binomial, CRC_df), "./brm_models_exercises/logistic_ex8")
```

With so much data, we didn't worry too much about the default priors (especially as we use the rescaled age), and first we check the trace plots

```
mcmc_plot(brmfit_ex8, type = "trace")
```



which all look quite good, as do the $\hat{R}$ values:

```
rhat(brmfit_ex8)
```

```
##  b_Intercept      b_age_s  b_sexfemale  b_raceblack  b_raceother b_registry02
##    1.0015216    1.0003160    0.9992567    1.0002368    1.0005167    1.0007795
## b_registry20 b_registry21 b_registry22 b_registry23 b_registry25 b_registry26
##    1.0009280    0.9998673    1.0008285    1.0006513    1.0002542    1.0010904
## b_registry27   b_Iage_sE2       lprior         lp__
##    1.0002933    1.0002461    1.0015002    1.0014880
```

and the effective sample sizes:

```
summary(brmfit_ex8)$fixed$Bulk_ESS
```

```
##  [1] 1522.641 2942.224 5624.633 3140.737 2030.632 1646.479 1742.972 2103.487
##  [9] 1797.273 2096.491 1789.214 2129.869 1990.103 3181.412
```

```
summary(brmfit_ex8)$fixed$Tail_ESS
```

```
##  [1] 2477.065 3061.518 2881.652 2791.567 2433.148 2310.909 2531.736 2415.921
##  [9] 2629.364 2317.694 2652.390 2671.945 2591.459 2991.736
```

If we look at the regression coefficients

```
fixef(brmfit_ex8)
```

```
##                Estimate   Est.Error        Q2.5       Q97.5
## Intercept  -7.65818507 0.005119319 -7.66820613 -7.64814300
## age_s       1.07008870 0.002562819  1.06503310  1.07499561
## sexfemale  -0.31939029 0.003313728 -0.32606753 -0.31313105
## raceblack   0.20584732 0.005775137  0.19447440  0.21703037
```

```
## raceother  -0.04458297 0.007262838 -0.05879977 -0.02998851
## registry02  0.14878567 0.006030355  0.13701165  0.16035380
## registry20  0.05915472 0.006004347  0.04756327  0.07069752
## registry21  0.13163374 0.009296847  0.11326611  0.15011583
## registry22  0.13142994 0.006189252  0.11953835  0.14352611
## registry23 -0.16373909 0.008493930 -0.18013640 -0.14688746
## registry25 -0.04166973 0.006344099 -0.05402193 -0.02927285
## registry26 -0.20680949 0.008591501 -0.22358324 -0.18985897
## registry27 -0.04785530 0.007448580 -0.06228802 -0.03342212
## Iage_sE2   -0.10871154 0.001083747 -0.11087805 -0.10655925
```

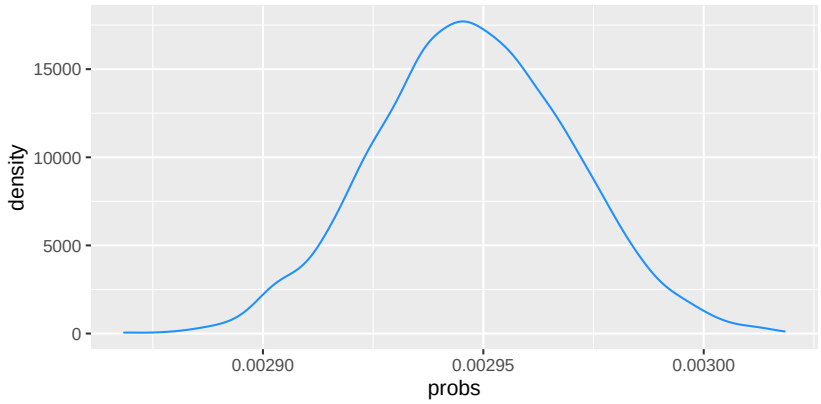
their posteriors are all well away from zero indicating they are strong predictors of cancer incidence.

From these we can extract posterior estimates. For example, for a 75 year old Black female from registry 27, from the posterior samples of the regression coefficients we would have the following mapping to the sampled log-odds

$$b\_Intercept + 2.5 * b\_age\_s + b\_sexfemale + b\_raceblack + b\_registry27 + 2.5^2 * b\_Iage\_sE2$$

From the posterior samples we could therefore create a new column of the log-odds, which we transform back to probabilities with the inverse-logit, or expit, function

```
expit <- function(x) { # inverse logit
  exp(x)/(1 + exp(x))
}
as_draws_df(brmfit_ex8) %>% mutate(log_odds = b_Intercept + 2.5*b_age_s + b_sexfemale +
  b_raceblack + b_registry27 + 2.5^2*b_Iage_sE2,
  probs = expit(log_odds)) %>%
  ggplot(aes(probs)) + geom_density(color = "dodgerblue")
```



The resulting probabilities are mostly between 0.0029 and 0.003, which is closely aligned with the observed frequency of cancer in that stratum:

```
CRC_df %>% filter(age == 75, sex == "female", race == "black", registry == 27) %>%
  mutate(prob = cancers/total)
```

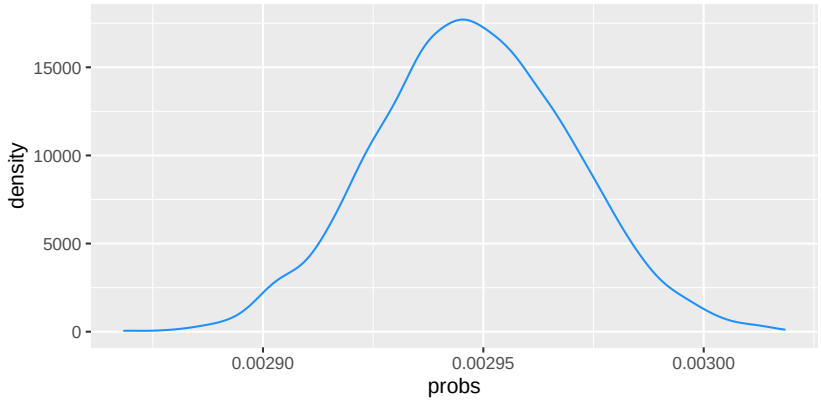
```
## # A tibble: 1 x 9
##     age sex    race  registry cancers noncancers total age_s    prob
##   <dbl> <fct>  <fct> <chr>      <int>      <dbl> <dbl> <dbl>   <dbl>
## 1    75 female black 27           136      47867 48003   2.5 0.00283
```

Rather than doing this by hand, we can use the `posterior_predict` function (or rather the `posterior_linpred` function for the linear modelling part) on new data. For this we make a small data frame of our individual

```
newdata <- data.frame(
  age_s = c(2.5), #transformed age
  sex = factor("female"),
  race = factor("black"),
  registry = c("27"),
  total = 1e5 # dummy value needed to make the functions work
)
```

and pass it into `posterior_linpred` with the argument `transform = TRUE` to output the results in the probability space (rather than the logit-space)

```
posterior_linpred(brmfit_ex8, newdata, transform = TRUE) %>% data.frame(probs = .) %>%
  ggplot(aes(probs)) + geom_density(color = "dodgerblue")
```



Optional Exercise 8.2 - Logistic regression

To run a logistic regression, we can use the `glm()` function with `family = "binomial"` (see details in `?stats::family`) and `formula = cbind(cancers, noncancers) ~ ...`

- *Fit a logistic regression model of cancer incidence with `age_s`, `sex`, `race`, and `registry`, as explanatory variables (no interactions). Examine the model summary and coefficients.*
- *Use `I(age_s^2)` to add a quadratic `age_s` term to the model.*
- *Compare the regression coefficients to the Bayesian model in Exercise 8.1.*
- *Install the `visreg` package, and use `visreg(..., "age_s")` to visualise the fitted slope of `age_s` (x-axis) with respect to the log odds (y-axis). The points are the partial residuals with respect to `age_s`. Does the model fit and `visreg` plot change for the better when including the quadratic term?*

We now run the data through the logistic regression using the syntax above

```
glm_fit <- glm(formula = cbind(cancers, noncancers) ~ age_s + sex + race + registry,
               family = "binomial", data = CRC_df)
summary(glm_fit)
```

```
##
## Call:
## glm(formula = cbind(cancers, noncancers) ~ age_s + sex + race +
##     registry, family = "binomial", data = CRC_df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -12.0480   -1.2495    0.3743    1.5614    8.4957
```

```
##
## Coefficients:
##              Estimate Std. Error   z value Pr(>|z|)
## (Intercept) -7.740603   0.004924 -1572.071  < 2e-16 ***
## age_s        0.873127   0.001333   655.139  < 2e-16 ***
## sexfemale   -0.328223   0.003242  -101.249  < 2e-16 ***
## raceblack    0.217360   0.005805    37.441  < 2e-16 ***
## raceother   -0.036421   0.007428    -4.903 9.43e-07 ***
## registry02   0.151314   0.005807    26.056  < 2e-16 ***
## registry20   0.061408   0.005776    10.632  < 2e-16 ***
## registry21   0.137047   0.009304    14.729  < 2e-16 ***
## registry22   0.127811   0.006039    21.164  < 2e-16 ***
## registry23  -0.153910   0.008205   -18.759  < 2e-16 ***
## registry25  -0.040474   0.006113    -6.621 3.58e-11 ***
## registry26  -0.209244   0.008556   -24.455  < 2e-16 ***
## registry27  -0.046357   0.007399    -6.266 3.71e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 605833  on 3059  degrees of freedom
## Residual deviance:  19925  on 3047  degrees of freedom
## AIC: 35111
##
## Number of Fisher Scoring iterations: 4
```

Each coefficient represents the change in log-odds of cancer for a unit change in the continuous variable `age_s` or a change in level (compared to reference) for the categorical variables.

To include the quadratic term in `age_s`, we use the suggested syntax and obtain the following regression results:

```
glm_fit2 <- glm(formula = cbind(cancers, noncancers) ~ age_s + sex + race + registry +
                  I(age_s^2), family = "binomial", data = CRC_df)
summary(glm_fit2)
```

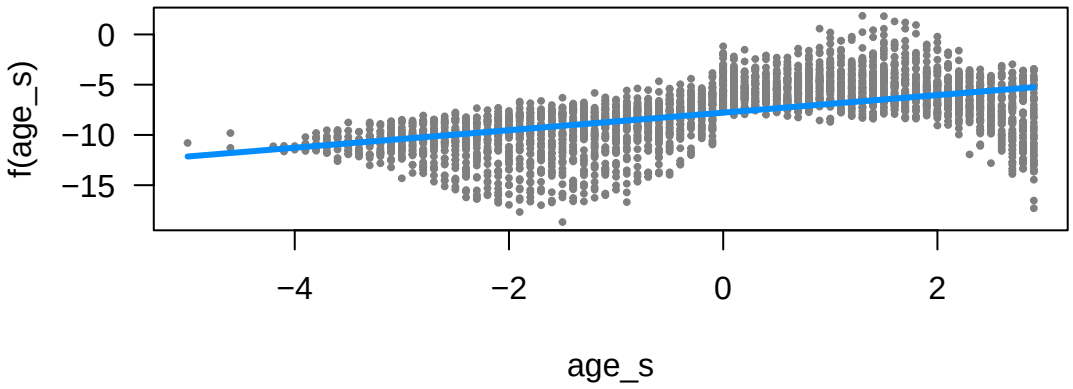
```
##
## Call:
## glm(formula = cbind(cancers, noncancers) ~ age_s + sex + race +
##     registry + I(age_s^2), family = "binomial", data = CRC_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.8771  -0.5817   0.6021   1.5625   5.8589
##
## Coefficients:
##              Estimate Std. Error   z value Pr(>|z|)
## (Intercept) -7.658026   0.005032 -1521.894  < 2e-16 ***
## age_s        1.069954   0.002598   411.804  < 2e-16 ***
## sexfemale   -0.319354   0.003242   -98.517  < 2e-16 ***
## raceblack    0.205940   0.005800    35.504  < 2e-16 ***
## raceother   -0.044624   0.007401    -6.029 1.65e-09 ***
## registry02   0.148727   0.005805    25.620  < 2e-16 ***
## registry20   0.058935   0.005773    10.209  < 2e-16 ***
## registry21   0.131618   0.009281    14.182  < 2e-16 ***
```

```
## registry22   0.131343   0.006037    21.755  < 2e-16 ***
## registry23  -0.163881   0.008204   -19.977  < 2e-16 ***
## registry25  -0.041858   0.006112    -6.849 7.46e-12 ***
## registry26  -0.206759   0.008555   -24.168  < 2e-16 ***
## registry27  -0.048040   0.007392    -6.499 8.10e-11 ***
## I(age_s^2)  -0.108658   0.001084  -100.194  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 605833.5  on 3059  degrees of freedom
## Residual deviance:   8430.5  on 3046  degrees of freedom
## AIC: 23619
##
## Number of Fisher Scoring iterations: 4
```

There are slight changes to all categorical regression coefficients, and an obvious large change for the `age_s` and `(Intercept)` now we have the quadratic term. These regression coefficients are also all very similar to the Bayesian logistic regression in Exercise 8.1.

The quadratic term for `age_s` is a highly significant predictor, suggesting already that the quadratic dependence on `age_s` is a better fit than a linear model. When we visualise the models with `visreg`, without the quadratic term we get the following plot

```
library(visreg)
visreg(glm_fit, xvar = "age_s")
```



where the partial residuals suggest a nonlinear trend in `age`. With the quadratic term, this looks a lot better:

```
visreg(glm_fit2, xvar = "age_s")
```