GR5065 HW5 Answer Key

1 Auto Insurance Claims

This question was based on this paper.

```r
FILE <- "telematics_syn-032021.csv"
if (!file.exists(FILE)) {
  ROOT <- "http://www2.math.uconn.edu/~valdez/"
  download.file(paste0(ROOT, FILE), destfile = FILE)
}
UBI <- readr::read_csv(FILE, show_col_types = FALSE)
```

```r
library(dplyr)
UBI <- mutate(UBI, # change the units to something sane
              Insured.age = Insured.age / 10,
              Credit.score = Credit.score / 100,
              Annual.miles.drive = Annual.miles.drive / 10000,
              Years.noclaims = Years.noclaims / 10,
              Total.miles.driven = Total.miles.driven / 10000,
              Brake.06miles = Brake.06miles / 100,
              Brake.08miles = Brake.08miles / 100,
              Brake.09miles = Brake.09miles / 100,
              Brake.11miles = Brake.11miles / 100,
              Brake.12miles = Brake.12miles / 100,
              Brake.14miles = Brake.14miles / 100) %>%
  rename(Pct.drive.rush_am = `Pct.drive.rush am`,
         Pct.drive.rush_pm = `Pct.drive.rush pm`)
```

```r
incl <- sample(1:nrow(UBI), size = 10000, replace = FALSE)
training <- slice(UBI, incl)
UBI <- slice(UBI, -incl)
```

```r
incl <- sample(1:nrow(UBI), size = 10000, replace = FALSE)
testing <- slice(UBI, incl)
```

1.1 Poisson Model

```r
library(brms)
```

```
Warning: package 'Rcpp' was built under R version 4.2.1
```

```r
options(mc.cores = parallel::detectCores())
```

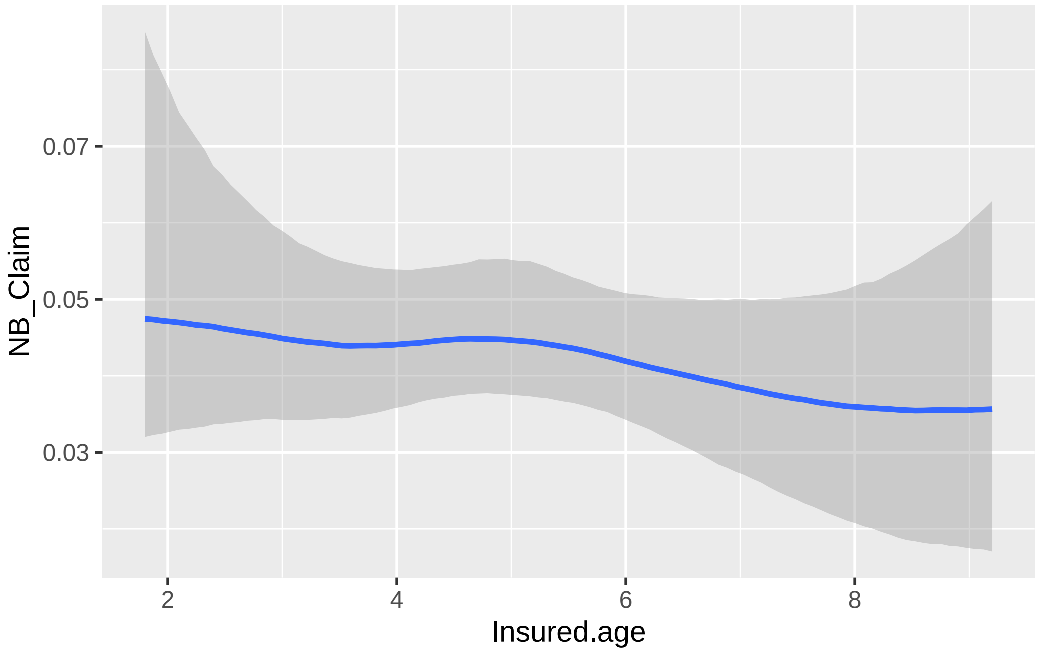
```r
poisson_prior <-
  prior(normal(-8, 2),  class = "Intercept") +
  prior(normal(0, 0.5), class = "b") +
  prior(exponential(1), class = "sds")
```

```r
post <- brm(NB_Claim ~ s(Insured.age) + Insured.sex + Credit.score +
              Annual.miles.drive + Marital + Years.noclaims +
              Pct.drive.wkend + Pct.drive.rush_am + Pct.drive.rush_pm +
              Brake.06miles + Brake.08miles + Brake.09miles +
              Brake.11miles + Brake.12miles + Brake.14miles +
              offset(log(Duration)),
            data = training,
            family = poisson,
            prior = poisson_prior,
            save_pars = save_pars(all = TRUE),
            seed = 20230427,
            control = list(adapt_delta = 0.999)) # eliminate divergences
```

```
Compiling Stan program...
```

```
Start sampling
```

```
plot(conditional_effects(post, "Insured.age"))
```



1.2 Zero Inflated Poisson Model

Some people do not drive their cars every day, in which case they would not have auto insurance claims for traffic accidents on those days. Thus, there could be zeros in the outcome due to not driving much and / or zeros in the outcome due to driving carefully.

```
post_ZI <- brm(
  bf(NB_Claim ~ s(Insured.age) + Insured.sex + Credit.score +
                Annual.miles.drive + Marital +
                Pct.drive.wkend + Pct.drive.rush_am + Pct.drive.rush_pm +
                Brake.06miles + Brake.08miles + Brake.09miles +
                Brake.11miles + Brake.12miles + Brake.14miles +
                offset(log(Duration)),
     zi ~ Years.noclaims),
  data = training,
  family = zero_inflated_poisson,
  prior = poisson_prior +
    prior(normal(-0.05, 0.1), class = "b",
          coef = "Years.noclaims", dpar = "zi"),
  save_pars = save_pars(all = TRUE),
  seed = 20230427,
  control = list(adapt_delta = 0.999)) # eliminate divergences
```

```
Compiling Stan program...
```

```
Start sampling
```

1.3 Model Comparison

```
loo_post <- loo(post, moment_match = TRUE)
```

```
Warning: Some Pareto k diagnostic values are slightly high. See help('pareto-k-diagnostic')
```

```
loo_post_ZI <- loo(post_ZI, moment_match = TRUE)
```

```
Warning: Some Pareto k diagnostic values are slightly high. See help('pareto-k-diagnostic')
```

```
loo_compare(loo_post, loo_post_ZI)
```

```
        elpd_diff se_diff
post_ZI  0.0       0.0   
post    -0.8       1.3   
```

These two models have very similar potential to predict future data. If you had to choose one, it would be the zero-inflated one but if your audience were unfamiliar with zero-inflated models, it would not hurt to go with the plain Poisson model.

```
log_lik(post_ZI, newdata = testing) %>%
  apply(MARGIN = 2, FUN = loo::logMeanExp) %>%
  sum
```

```
[1] -1786.937
```

```
loo_post_ZI
```

```
Computed from 4000 by 10000 log-likelihood matrix

         Estimate    SE
elpd_loo  -1806.3  64.8
p_loo        10.6   0.8
looic      3612.6 129.6
------
Monte Carlo SE of elpd_loo is 0.1.

Pareto k diagnostic values:
                         Count Pct.    Min. n_eff
(-Inf, 0.5]   (good)     9988  99.9%   765
 (0.5, 0.7]   (ok)         12   0.1%   755
   (0.7, 1]   (bad)         0   0.0%   <NA>
   (1, Inf)   (very bad)    0   0.0%   <NA>

All Pareto k estimates are ok (k < 0.7).
See help('pareto-k-diagnostic') for details.
```

The ELPD of the zero-inflated model is about -1800 with a standard error of 65, so we would anticipate any particular dataset of the same size to have a LPD around there. The LPD of the training data was -1787, which should make it clear that splitting the original data into training and test was not only unnecessary — because we can estimate the ELPD just fine from the data that was conditioned on — but also that holding out the testing data represents a wasted opportunity to obtain a more precise posterior distribution.

2 Pandemic Election

Data were recently released that allow researchers to join the dataset from the post-2020 American National Election Survey (ANES) with the General Social Survey (GSS) panel survey where the same people were first asked questions in either 2016 or 2018 and then asked a largely identical set of questions a few months before the 2020 election.

```
suppressPackageStartupMessages(library(haven))
if (!file.exists("GSS_2020_panel_stata_1a.zip")) {
  download.file("https://gss.norc.org/Documents/stata/GSS_2020_panel_stata_1a.zip",
                destfile = "GSS_2020_panel_stata_1a.zip")
  unzip("GSS_2020_panel_stata_1a.zip")
}

GSS <- as_factor(read_dta("gss2020panel_r1a.dta"))
```

```
unzip("anes_timeseries_2020_gss_stata_20220408.zip") # creates a few files
ANES <- as_factor(read_dta("anes_timeseries_2020_gss_stata_20220408.dta"))
```

```
ANES_GSS <- inner_join(ANES, GSS, by = c(YEARID = "yearid"))
```

```
ANES_GSS <- ANES_GSS %>%
  mutate(natsci_1 = natsci_1b, # a factor
         natsci_1 = if_else(is.na(natsci_1), natsci_1a, natsci_1),
         degree_2 = factor(degree_2,
                           levels = c("less than high school",
                                      "high school",
                                      "associate/junior college",
                                      "bachelor's",
                                      "graduate"), ordered = TRUE),
         age_2 = as.numeric(as.character(age_2)),
         y = factor(V202383x, ordered = TRUE,
                    levels = c("1. Health benefits much greater than risks",
                               "2. Health benefits moderately greater than risks",
                               "3. Health benefits slightly greater than risks",
                               "4. No difference",
                               "5. Risks slightly greather than health benefits",
                               "6. Risks moderately greater than health benefits",
                               "7. Risks much greater than health benefits"))) %>%
  filter(!is.na(y), !is.na(natsci_1),
         !is.na(age_2), !is.na(sex_2), !is.na(degree_2))
```

2.1 Proportional Odds Model

```
ord <- brm(y ~ age_2 + sex_2 + mo(degree_2) + (age_2 | natsci_2),
           data = ANES_GSS,
           family = cumulative,
           prior = prior(normal(0, 1), class = "Intercept") +
                   prior(normal(0, 0.5), class = "b") +
                     prior(exponential(1), class = "sd"))
```

```
Warning: Rows containing NAs were excluded from the model.
```

```
Compiling Stan program...
```

```
Start sampling
```

```
Warning: There were 997 divergent transitions after warmup. See
https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
to find out why this is a problem and how to eliminate them.
```

```
Warning: Examine the pairs() plot to diagnose sampling problems
```

```
Warning: The largest R-hat is NA, indicating chains have not mixed.
Running the chains for more iterations may help. See
https://mc-stan.org/misc/warnings.html#r-hat
```

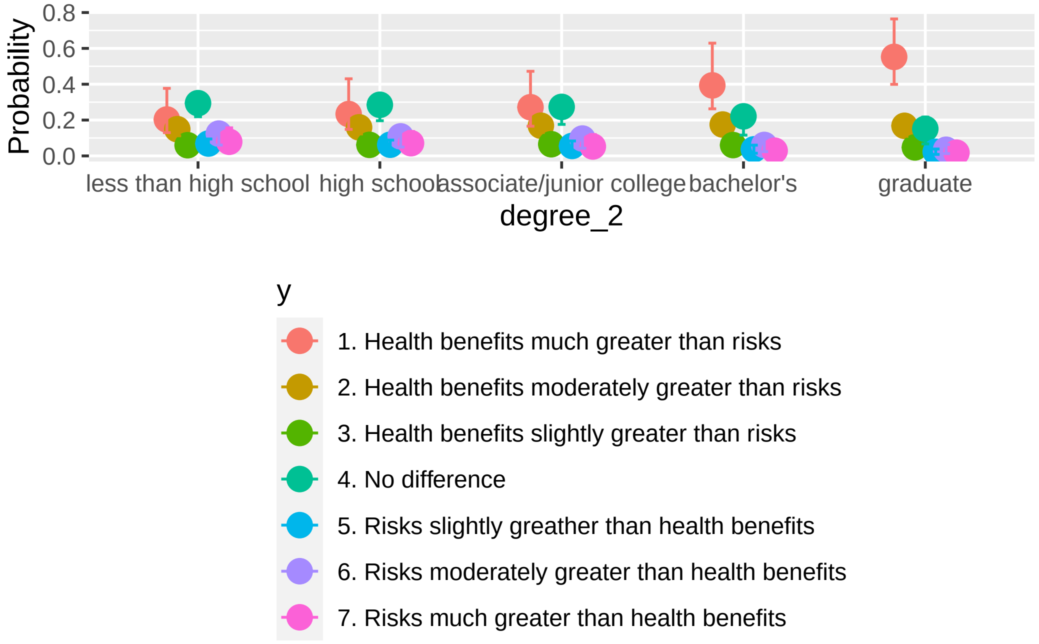
```
Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and median
Running the chains for more iterations may help. See
https://mc-stan.org/misc/warnings.html#bulk-ess
```

```
Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tai
Running the chains for more iterations may help. See
https://mc-stan.org/misc/warnings.html#tail-ess
```

I left the warnings here so you can see what they look like, but in any actual research, you should take steps to eliminate them because they imply that Stan had difficulty drawing from the posterior distribution that you defined.

2.2 Visualization

```r
library(ggplot2)
plot(conditional_effects(ord, "degree_2", categorical = TRUE),
     theme = theme(legend.position = "bottom",
                   legend.direction = "vertical"))
```



2.3 Frequentism

Although Frequentist hierarchical models are often used in some subfields, they are only justified from a Frequentist perspective if the grouping variable has been randomized as part of the research design. That is not the case when `natsci_2` (or a lot of other situations where Frequentist hierarchical models are used).

This is important because the likelihood function maximized by Frequentist hierarchical models first integrates the a_j and b_j variables out of the original likelihood function to leave an integrated likelihood function that only depends on the common parameters. In other words, the MLEs are not conditional on the grouping structure.

In contrast, Bayesian models, including hierarchical models, condition on the available data, including any grouping structure that may be present. The a_j and b_j are just as much parameters as any other unknown, and we can draw from the posterior distribution of all the unknowns conditional on the data.