GR5065 Homework 5

Ben Goodrich

Due March 30, 2021 at 8PM New York Time

```
# call the set.seed function once here to make the knitting conditionally deterministic
```

1 Social and political correlates of Covid-19

Read this paper, which was written in July of 2020, although we will use updated versions of the variables in their dataset through February 2021. This code is based on that in https://github.com/wzb-ipi/rep_corona/

First, we download the most current version of the dataset, correct the missingness in the `checks_veto` predictor, and rescale total population (`pop_tot`) back into people (as opposed to millions of people):

```
library(readr)
library(dplyr)
df <- suppressWarnings(read_csv("https://wzb-ipi.github.io/corona/df_full.csv",
                                col_types = cols(X1 = col_skip(),
                                date = col_date(format = "%Y-%m-%d"),
                                infections_ebola = col_integer()))) %>%
  mutate(checks_veto = ifelse(checks_veto < 0, NA, checks_veto),
         pop_tot = pop_tot * 10^6)
data_date <- max(df$date_rep, na.rm = TRUE)
df_today  <- df %>% filter(as.Date(date_rep) == data_date)
```

Next, we get the names of the variables in each of the families of predictors.

```
measures <- read_csv("https://raw.githubusercontent.com/wzb-ipi/rep_corona/master/measures.csv") %>%
  filter(include == 1)

families <- c("state_cap_vars", "pol_account_vars", "social_vars",
              "econ_vars",  "phys_vars", "epi_vars", "health_sys_vars")

for(v in families){
  assign(v, filter(measures, family == v)$vars %>% as.character)
  assign(paste0(v, "_labels"), filter(measures, family == v)$labels %>% as.character)
 }

controls <- c("pop_tot_log", "share_older", "healthcare_qual", "health_exp_pc", "detect_index")
controls_labels <- c("Total population (logged)", "Share 65+", "Healthcare quality index (GHSI)",
                     "Healthcare spending/capita", "Health data quality")
```

The variables in those families are explained in more detail in section 9.3 of the paper but briefly are:

```r
cbind(state_cap_vars, state_cap_vars_labels)
```

```
##      state_cap_vars        state_cap_vars_labels
## [1,] "gov_effect"          "Government effectiveness"
## [2,] "state_fragility"     "State fragility"
## [3,] "bureaucracy_corrupt" "Public sector corruption"
## [4,] "pandemic_prep"       "Pandemic preparedness"
## [5,] "infection"           "Ebola/SARS/MERS exposure"
## [6,] "checks_veto"         "Veto players"
## [7,] "federal_ind"         "Index of federalism"
```

```r
cbind(pol_account_vars, pol_account_vars_labels)
```

```
##       pol_account_vars pol_account_vars_labels
##  [1,] "vdem_libdem"    "Liberal democracy"
##  [2,] "pr"             "PR electoral system"
##  [3,] "vdem_mecorrpt"  "Media independence"
##  [4,] "oil"            "Oil rents (% of GDP)"
##  [5,] "electoral_pop"  "Electoral populism"
##  [6,] "woman_leader"   "Women leaders"
##  [7,] "dist_anyelec"   "Electoral pressure"
##  [8,] "polar_rile"     "Party polarization (MARPOR)"
##  [9,] "pos_gov_lr"     "Left-Right Government"
```

```r
cbind(social_vars, social_vars_labels)
```

```
##      social_vars       social_vars_labels
## [1,] "trust_gov"       "Institutional trust"
## [2,] "al_etfra"        "Ethnic fractionalization"
## [3,] "al_religfra"     "Religious fractionalization"
## [4,] "gini"            "Income GINI"
## [5,] "trust_people"    "Interpersonal trust"
## [6,] "migration_share" "Share foreign born"
## [7,] "share_powerless" "Prop. of marginalized groups"
```

```r
cbind(econ_vars, econ_vars_labels)
```

```
##      econ_vars    econ_vars_labels
## [1,] "gdp_pc"     "GDP per capita (PPP)"
## [2,] "trade"      "Trade (share of GDP)"
## [3,] "fdi"        "FDI (net inflows, USD)"
## [4,] "air_travel" "Air travel (passengers carried)"
```

```r
cbind(phys_vars, phys_vars_labels)
```

```
##      phys_vars         phys_vars_labels
## [1,] "pop_density_log" "Population density (log)"
## [2,] "pop_tot_log"     "Total population (logged)"
## [3,] "precip"          "Precipitation (mm/month)"
## [4,] "temp_mean"       "Temperature (Celsius)"
## [5,] "urban"           "Urban popularion (percent)"
```

```r
cbind(epi_vars, epi_vars_labels)
```

```
##      epi_vars            epi_vars_labels
## [1,] "share_older"       "Share 65+"
## [2,] "resp_disease_prev" "Respiratory disease prevalence"
```

```r
cbind(health_sys_vars, health_sys_vars_labels)
```

```
##      health_sys_vars    health_sys_vars_labels
## [1,] "share_health_ins" "Share with health insurance"
## [2,] "hosp_beds_pc"     "Hospital beds / capita (GHSI)"
## [3,] "detect_index"     "Health data quality"
## [4,] "health_index"     "Health sector robustness (GHSI)"
```

```r
cbind(controls, controls_labels)
```

```
##      controls          controls_labels
## [1,] "pop_tot_log"     "Total population (logged)"
## [2,] "share_older"     "Share 65+"
## [3,] "healthcare_qual" "Healthcare quality index (GHSI)"
## [4,] "health_exp_pc"   "Healthcare spending/capita"
## [5,] "detect_index"    "Health data quality"
```

1.1 Frequentist Inference

On page 36 (regarding Figure 11) the authors state

> We calculate confidence intervals using robust standard errors …[c]alculated using lm_robust from the estimatr package for R.

What Frequentist inference are the authors interested in making?

1.2 Clustered Standard Errors

The motivation for this estimator of the standard errors is that the data-generating process may not be $y_n = \mu_n + \epsilon_n$ with $\epsilon_n \sim \mathcal{N}(0, \sigma)$ but rather something else, such as $\epsilon_n \sim \mathcal{N}(0, \sigma_n)$. Frequents do not estimate each σ_n, and indeed cannot estimate σ_n consistently, because no matter how large is N, there is only one observation with which to estimate each σ_n. A justification for this estimator of the standard errors is given here.

What are the limitations of this estimator of the standard errors in the context of the authors' models of national deaths due to coronavirus?

1.3 Leave-One-Out Cross Validation

Figure 12 in the paper is constructed by (literal) leave-one-out cross validation, where the n-th country is individually omitted from the dataset, point estimates of the coefficients are estimated using all the other counties and Ordinary Least Squares, and $e^{\widehat{y_n}}$ is taken as the point prediction for the n-th observation.

Recreate (something akin to) Figure 12 (which will not be exact because the data in `df_today` are updated and you do not need to worry about labeling the points) using the `lm` function in R, where the outcome is called `deaths_cum_log` and the five predictors are listed above.

1.4 Posterior Distributions

LASSO is used at various points in the paper, especially in section 6, which is further explained in the appendix. Essentially, the authors include the five control variables along with predictors from the political or social families and see which coefficient estimates produced by LASSO are non-zero. The authors conclude on page 43 that

> the addition of political and social variables does not improve over models that include only simple demographic and health indicators. Overall, the results from this analysis suggest that, to date, these political and social variables have little explanatory power over and above simple demographic and health indicators.

We are going to do something similar, except using `stan_glm`, a binomial likelihood, informative priors, and comparisons of the Expected Log Predictive Density. The best way to estimate a binomial model using `stan_glm` is by specifying a matrix of deaths and non-deaths to the left of the `~`, like `stan_glm(cbind(deaths_cum, pop_tot - deaths_cum) ~ ?, data = df_today, family = binomial, prior_intercept = ??, prior = ???)` Doing so models the (log-odds of the) probability of death due to covid in a country, where the number of independent trials is given by the total population. Thus, $\eta_n = \frac{\ln \mu_n}{\ln(1-\mu_n)}$ is the log-odds of death due to covid and α is expectation for a country with average values on all the predictors. However, `pop_tot_log` should *not* be included as a predictor because the total population is already conditioned on in the binomial likelihood.

Estimate four binomial models for cumulative deaths using `stan_glm`

1. With the four control variables only
2. With the four control variables and the political accountability variables
3. With the four control variables and the social variables
4. With the four control variables, the political accountability variables, and the social variables.

In each case, the prior on the coefficients should be based on the theoretical considerations given in section 3 of the paper.

1.5 Model Comparison

Which of the four models has the highest estimated Expected Log Predictive Density? What set of four non-negative weights (that adds up to 1) on the predictions from each of the four models maximizes the Expected Log Predictive Density?

1.6 Overfitting

For the best of the four models, assess whether it is likely to do a worse job of predicting future data than past data. In other words, judge the extent to which the best model overfits the past data.

2 Count Models

Read https://osf.io/ux9et even though it is Frequentist.

The data on the number of times a video was watched on YouTube can be loaded with

```r
youtube <- read_csv("https://osf.io/25sz9/download")
```

The main two predictors are `scol`, which is a measure of how accurate the video is, and `age2`, which is a measure of how long the video has been available. The outcome variable is called `views2`.

2.1 Log-Likelihood

Draw once from you prior distribution of α, β_1, β_2 and ϕ, where the conditional variance of the n-th observation under a negative binomial process is $\mu_n + \frac{\mu_n^2}{\phi}$ and $\mu_n = e^{\eta_n}$ and $\eta_n = \alpha + \beta_1 \times \mathrm{scol} + \beta_2 \times \mathrm{age2}$. You should center the two predictors so that α is the expectation of η_n for a video of average accuracy and age.

What number is the log-likelihood of these realizations of α, β_1, β_2 and ϕ, evaluated over all the observations in the `youtube` dataset? You should use the `dnbinom` function where `mu` is a vector of conditional expectations and `size` is ϕ.

2.2 Prior Predictive Distribution

Draw 1000 times from the prior predictive distribution of YouTube views for each of the observations in the `youtube` dataset. Show that your prior predictive distribution is reasonable by verifying it puts only a little bit of probability on values greater than a half-million views.

2.3 Posterior Distribution

Call `stan_glm.nb` in the rstanarm package (which is equivalent to `stan_glm` with `family = neg_binomial_2`) to estimate a generalized linear model for YouTube views. How would you describe your posterior beliefs about β_1.

2.4 Splines

Call `stan_gamm4` in the rstanarm package with `family = neg_binomial2` to estimate a model where the logarithm of the expected number of views is a smooth but non-linear function of `scol` and / or `age2`. How would you describe your posterior beliefs about this relationship?

2.5 Model Comparison

Does the generalized linear model or the smooth but non-linear model have a higher Expected Log Predictive Density for future YouTube videos? Are there any observations that are too influential on the posterior distribution for the Pareto-Smooth Importance Sampling estimator to be valid?