Lab 7

Stephen R. Proulx

Bayesian Statistical Modeling Winter 2024

Lab Exercise, Week 8

When is this lab due? Labs are due on the Thursday after they are assigned. However, in many cases you can complete them during the lab period itself. This assignment is due on Thursday, 3/7/2022.

Load and process the data

Load the dataset. It contains Covid case numbers and deaths for all counties in California.

```
load("SB_covid_data_2022.Rdata")
```

We're going to calculate some new columns that summarize weekly total Covid cases. To do this we need to group by the county, and make sure the data are in order by date. The column `delta.days` is the number of days elapsed since data on COVID cases were first reported last winter.

```
county_ca <- group_by(county_ca,county) %>%
  arrange(delta.days) %>%
  mutate(new_cases = cases-lag(cases),
         week_cases=(cases-lag(cases,7))/7,
         week_old_cases=(lag(cases,7)-lag(cases,14))/7,
         two_week_old_cases=(lag(cases,14)-lag(cases,21))/7,
         three_week_old_cases=(lag(cases,21)-lag(cases,28))/7,
         four_week_old_cases=(lag(cases,28)-lag(cases,35))/7,
         new_deaths = deaths-lag(deaths)) %>%
  ungroup()
```

We are now going to pull out the Santa Barbara county data, but only since day 100. We will also standardize our potential predictor variables.

```
firstday=101

  sb_data<-filter(county_ca,delta.days>firstday,county == "Santa Barbara", state=="California")%>%
  view()
```

Check out the data

This draws a cubic spline fit through the 7-day average case number.

```
firstday=101
maxday=800
ssvals <- smooth.spline(x=sb_data$delta.days, y= sb_data$week_cases, df=15)
spline_data <- tibble( delta.days=ssvals$x, week_cases = ssvals$y)


ggplot(data=sb_data, aes(x=delta.days,y= week_cases/7)) +
  geom_point()+
  geom_line(data=spline_data,color="red")+
  scale_y_continuous( limits=c(1,250) )+
  scale_x_continuous(limits=c(firstday,maxday))+
  labs( x="days since Jan 22" , y="7-day average cases")
```

```
ggplot(data=sb_data, aes(x=delta.days,y=new_deaths )) +
  geom_point()+
  scale_y_continuous( limits=c(-2,40) )+
  scale_x_continuous(limits=c(firstday,maxday))+
  labs( x="days since Jan 22" , y="deaths")
```

Now we'll prune down some of the columns we are keeping and remove any negative numbers (these are due to corrections in the reporting after the fact and sometimes make it appear that people were un-dead)

```
stan_data <- select(sb_data,new_cases, week_cases ,week_old_cases, two_week_old_cases, three_week_old_ca
  mutate(new_deaths= new_deaths * (new_deaths>0) ) %>%
  mutate(NC=standardize(new_cases),
         W0=standardize(week_cases),
         W1=standardize(week_old_cases),
         W2=standardize(two_week_old_cases),
         W3=standardize(three_week_old_cases),
         W4=standardize(four_week_old_cases))%>%
  rowid_to_column(var="index")
```

Example: Fitting the overall mortality with a Poisson

```
m.poiss.overall <- ulam(alist(
   new_deaths ~ dpois(theta),
   log(theta) <- log_theta,
   log_theta ~ dnorm(0,2)
),data=stan_data , log_lik = TRUE)
```

```
precis(m.poiss.overall)
```

Redo it with more chains. Here's the trick to keep from re-compiling. But this only works if you want to run the same specific model. This is helpful, but it doesn't keep all of the information stored in an `ulam` model, so be aware of this limitation.

```
m.poiss.overall.v2 <- stan(fit=m.poiss.overall@stanfit , data=stan_data , chains=4 , iter=3000)
```

```
precis(m.poiss.overall.v2)
```

```
WAIC(m.poiss.overall)
```

How does it compare to our data?

```
sim_out_wide<-rethinking::sim(m.poiss.overall,data=stan_data)

ndays <- nrow(stan_data)

sim_out<-as_tibble(sim_out_wide) %>%
  gather( "index","deaths",1:(ndays)) %>%
  separate(index,c("V","number"),sep=1) %>%
  mutate(number=as.numeric(number))
```

```
sim_summarized <- group_by(sim_out,number) %>%
  summarise(deaths_mean=mean(deaths),
            deaths_lower = quantile(deaths,0.05),
            deaths_upper = quantile(deaths,0.95))%>%
  ungroup()
```

```
ggplot(sim_summarized, aes(x=number,y=deaths_mean))+
  geom_point()+
  geom_errorbar( aes(ymin=deaths_lower,ymax=deaths_upper))+
  geom_point(data=stan_data, aes(x=index,y=new_deaths), color="red")
```

A useful technique for post-processing the samples.

```
tmp<-extract(m.poiss.overall@stanfit) %>% as_tibble()%>%
  mutate(theta=exp(log_theta) )

bayesplot::mcmc_intervals(tmp,pars=c("theta"))
```

Part 1: Poisson model with the week's cases as a predictor

Create a model where the likelihood model for the daily number of new deaths is Poisson and there is an additive model for the log transformed Poisson parameter. Choose priors based on the range of mean Poisson values that could be produced given the range of the predictor variables. Start with a model where the current week's average Covid cases is the only predictor.

Part 2:

Once you have samples for your model, explain how you can have some confidence that the chains are doing a good job of approximating the posterior distribution.

Part 3

Plot the posterior predictions from the model and compare them to the actual data. What do they consistently capture and what do they consistently miss?

Part 4:

Now build a series of models with prior week's average case numbers as predictors. Compare the WAIC values of the different models. What can you conclude? What possible confounds should you consider before making any conclusions?

Bonus du jour: Day of the week effect

There are several issues with when deaths are reported. They are reported when the coroner confirms it as a Covid death, which may be delayed from the actual death. They also may not be reported on some days of the week. Can you test for this effect?