STA2201H Methods of Applied Statistics II

Monica Alexander

Week 12: Misc

Overview

- Sampling distributions (A3)
- A selection of things that are modeling extensions to what you already know
- Some more R coding projects and where to find help
- Exam

Sampling distributions

Sampling distributions

- How does the Bayes estimator compare to the MLE?
- We can check sampling properties, which refer to behavior under hypothetically repeatable surveys or experiments.
- Estimates versus estimators:
  - After observing the sample, $\hat{\mu}_{MLE}$ and $\hat{\mu}_{Bayes}$ are point estimates for μ
  - Before observing the data, the estimators $\hat{\mu}_{MLE}$ and $\hat{\mu}_{Bayes}$ are unknown quantitative outcomes (because they depend on the yet-to-be-observed data), so we think of them as a random variable with a probability distribution.

Sampling distributions

- The sampling distribution of an estimator $\hat{\mu}$ refers to the point estimates that would be obtained if new data sets were obtained.
- Note: the sampling distribution of μ_{Bayes} is not the posterior distribution of μ^*
- In this example, with $y_1, y_2, \ldots, y_n|\mu^*, \sigma^2 \sim N\left(\mu^*, \sigma^2\right)$, Sampling distribution for $\hat{\mu}_{\text{Bayes}} = \frac{\mu_0 + n\bar{y}}{n+1}$

$$\hat{\mu}_{\text{Bayes}}|\mu^* \sim N\left(E\left[\hat{\mu}_{\text{Bayes}}|\mu^*\right], \text{Var}\left[\hat{\mu}_{\text{Bayes}}|\mu^*\right]\right)$$

$$E\left[\hat{\mu}_{\text{Bayes}}|\mu^*\right] = \frac{\mu_0 + n\mu^*}{n+1}$$

$$\text{Var}\left[\hat{\mu}_{\text{Bayes}}|\mu^*\right] = \frac{n}{(n+1)^2}\sigma^2$$

Cool things you can do with these modeling skills

Extending the data model: error around observations

Extending the data model

So far we've thought about (most) models in terms of

$$y_i = X\beta + \varepsilon_i$$

with $\varepsilon_i \sim N(0, \sigma_y^2)$ and then we put a prior on σ_y^2 and estimate it in the model.

Equivalently, $y_i \sim N(X\beta, \sigma_y^2)$

- This is the likelihood or **data model** for y_i.
- Can read as "observed data is truth plus some error", and then we model the "truth" (i.e. expected Y) with a (generalized) linear model
- But we often have some info about ε_i

Data model and measurement error

$$y_i = X\beta + \varepsilon_i$$

- e.g. if y_i is from a representative survey, we have sampling error, which is a function of the size of the sample
- we also have stochastic error based on size of population
  - e.g. if looking at death probabilities could assume normal approximation to Binomial and calculate standard errors as $\sqrt(p \cdot (1-p))/n$
  - side note, this is how the variance approximation to Kaplain Meier works
- we may also have non-sampling error, other sources of bias that we may have info on
  - non-response
  - e.g. recall bias in surveys

Measurement error

- The power of data models in a Bayesian context is that we can account for different sources of error, and combine inputs on measurement error with other sources of error that need to be estimated
- E.g. $y_i \sim N(X\beta, \sigma_i^2)$ with $\sigma_i^2 = (\sigma_i^{\text{sampling}} + \sigma^{\text{bias}})^2$ where $\sigma_i^{\text{sampling}}$ is known and σ^{bias} is to be estimated

Example

'Combining social media and survey data to nowcast migrant stocks in the United States' https://arxiv.org/abs/2003.02895

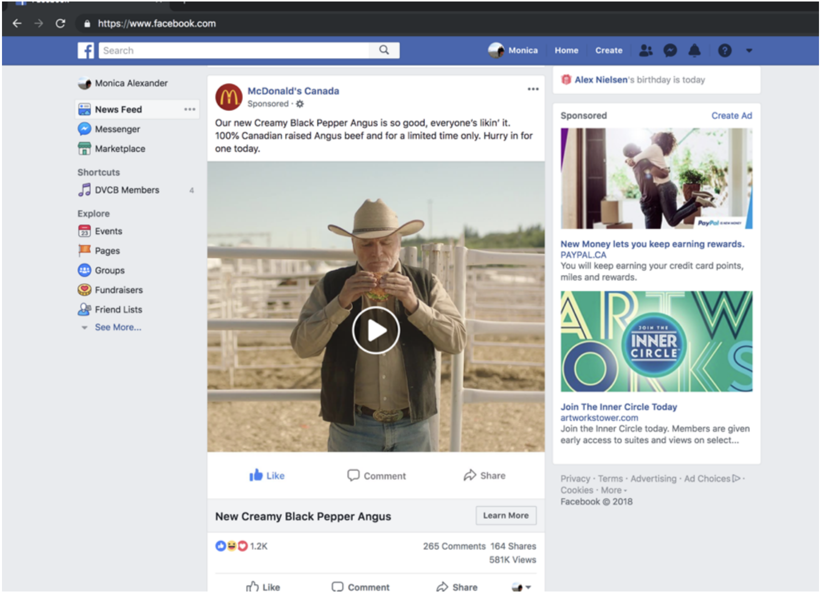
Measuring migration

- Is important, but hard
- Data are usually limited, and data that do exist are delayed
- E.g. in the US, good quality data from nationally representative survey, but 1-2 year delay in release

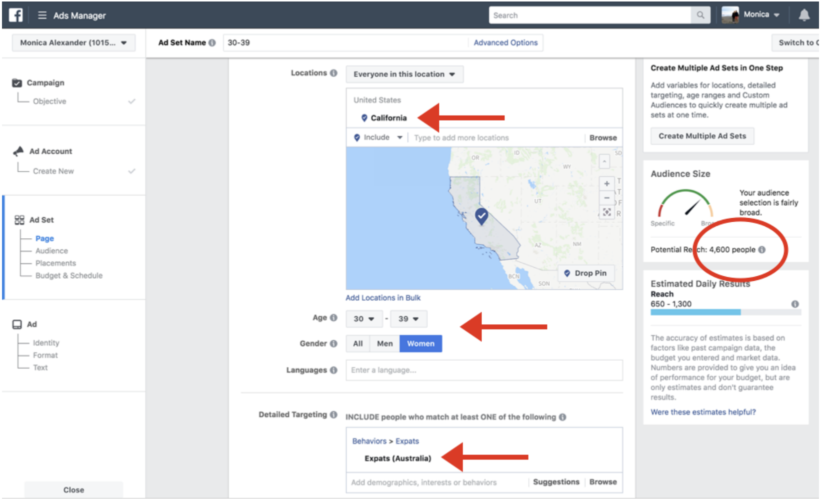
Social media as a data source

- Users of social media as their own population, with births, deaths and movements
- Big samples, updated in essentially real time
- Non-representative, self-report bias, confidentiality

Demographic data from Facebook's Advertsing Platform



Demographic data from Facebook's Advertsing Platform

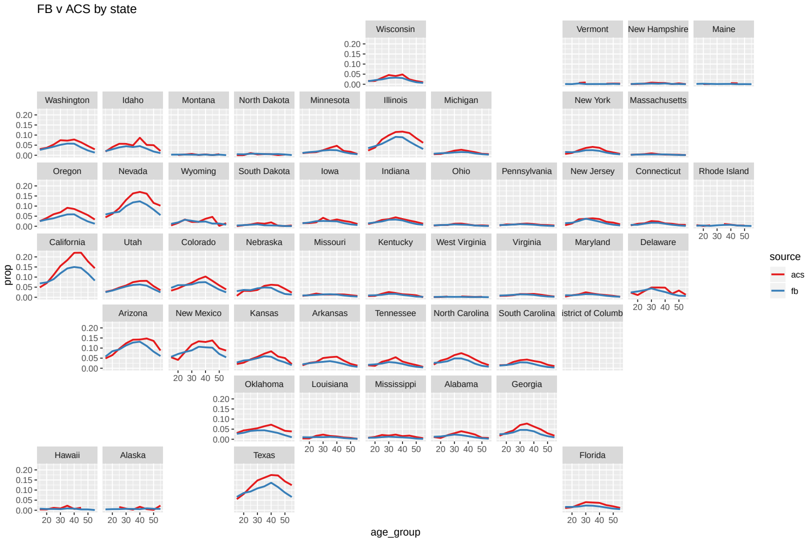


X

Demographic data from Facebook's Advertsing Platform

- Use API to automate data collection
- Collecting waves of data for $\sim$ 2 years

Problems and promises



Goals

1. Adjust for biases in Facebook data to effectively use up-to-date information on migration patterns
2. Incorporate longer time series of information from American Community Survey
3. Combine data in a probabilistic way; incorporate uncertainty in data

Step 3 is achieved through combining both sources in a Bayesian hierarchical framework, with a **data model** that allows for different types of uncertainty around the two different types of data

The model

$$\log p_{xts} \sim N(\log \mu_{xts}, \sigma^2)$$

Relating data to 'truth'

$$p_{xts} = \begin{cases} \textbf{from ACS}, & \textbf{if } 2001 \le t \le 2016 \\ p^*_{xts} \textbf{ (FB estimate)}, & \textbf{if } t \ge 2017 \end{cases}$$

Source of data

$$\sigma^2 = \begin{cases} \sigma_s^2, & \textbf{if } 2001 \le t \le 2016 \\ \sigma_s^2 + \sigma_{bias}^2 + \sigma_{ns}^2, & \textbf{if } t \ge 2017 \end{cases}$$

Uncertainty around data (varies based on source)

$$\begin{aligned} \log \mu_{xts} &= \beta_{ts,1} \cdot Z_{x,1} + \beta_{ts,2} \cdot Z_{x,2} + \varepsilon_{xts} \\ \beta_{ts,p} &= \Phi_{t,p} + \delta_{ts,p} \\ \delta_{t,s,p} &\sim N(\delta_{t-1,s,p}, \sigma_p^2) \\ \varepsilon_{x,t,s} &\sim N(\rho_{xs}\varepsilon_{x,t-1,s}, \sigma_{x,s}^2) \\ \log \sigma_{x,s}^2 &\sim N(\xi_x, \psi_x) \end{aligned}$$

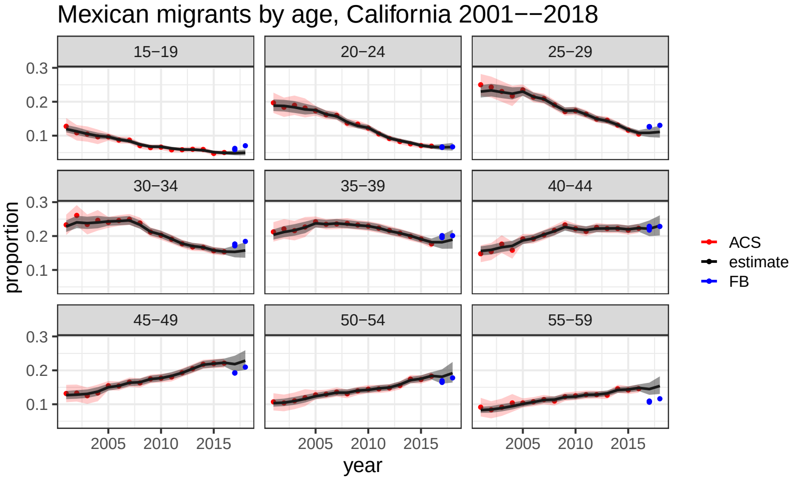
Time series model

$$\log p^*_{xts} = \beta_0 + \beta_1 \log p^{FB}_{xts} + \beta \mathbf{X}$$

x

Facebook bias adjustment model

Illustrative results



More than one data model

More than one data model

- So far we've just thought about a potential data model for the outcome y_i
- But often we may want incorporate/adjust for/allow for measurement error in covariates or other components of the model
- Hard to think about / adjust for in a classical set-up, but a relatively straight-forward extension in a Bayesian hierarchical set-up: can allow for multiple data models on both outcome and explanatory component(s) E.g.

$$
\begin{aligned}
y_i &\sim N\left(f\left(\phi_i\right) + \text{ blah }, \sigma^2_{y,i}\right) \\
X_i &\sim N\left(\phi_i, \sigma^2_{\phi,i}\right)
\end{aligned}
$$

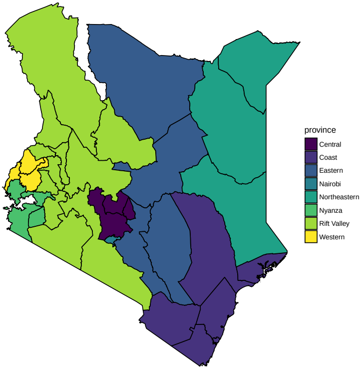
E.g y_i is graduating grade, ϕ_i is IQ, blah are other covariates, X_i is score on IQ test

Example

Work in progress: Estimating sub-national populations by age and sex in low-income countries

- In general we don't have a good idea of the basic characteristics of people by sub-national area in low income countries
- Infrequent censuses, no vital registration systems, political and economic instabilities
- But important to know for public health planning and provisions (disease and mortality risk varies by age, e.g. COVID-19)

Sub-national populations in Kenya



- Estimates do exist (e.g. WorldPop project)
- But age distributions take info from last census and apply to total population counts
- For Kenya, this was in 2009: huge changes in internal migration
- Under-estimate the relative elderly burden in rural areas

Model set-up

We know that (for adult populations)

$$P_{a,t} = P_{a-1,t-1} - D_{a-1,t-1} + I_{a-1,t-1} - O_{a-1,t-1}$$

- P is population, D are deaths, I is in-migration and O is out-migration
- This is an identity; we can't create or remove adults of age > 0 in any other way
- If we had exact data on each of these components, then no worries
- But even when we do have data, these components often don't line up (measurement error)
- For Kenya, we have P from censuses (1979, 1989, 1999, 2009), we don't really know anything else

Model set-up

Goal: estimate population by age in 47 counties in Kenya for 2019

Let's write as

$$
\begin{aligned}
\log y_i &\sim N\left(\log p_{a[i],t[i],r[i]}, \sigma_y^2[i]\right) \\
p_{a,t,r} &= p_{a-1,t-1,r} \cdot \left[(1 - q_{a-1,t-1,r}) + \phi_{a-1,t-1,r}\right]
\end{aligned}
$$

Notation:

- y_i are observed population counts, with measurement error $\sigma_y^2[i]$ (currently just sampling error based on the fact we are using a 10% sample of the census)
- $p_{a,t,r}$ is "true" population at age a time t and in region r
- The notation $a[i], t[i], r[i]$ means the a, t, r of the ith observation
- $q_{a,t,r}$ is the probability of death for a particular age
- $\phi_{a,t,r}$ is net-migration (as a proportion of population)

Model set-up

$$\begin{aligned} \log y_i &\sim N\left(\log p_{a[i],t[i],r[i]}, \sigma_y^2[i]\right) \\ p_{a,t,r} &= p_{a-1,t-1,r} \cdot [(1 - q_{a-1,t-1,r}) + \phi_{a-1,t-1,r}] \end{aligned}$$

- The first line is our data model (relating observed data to underlying but observed quantity of interest)
- The second line is a process model (i.e. describes the process of population growth over time)
- To get $p_{a,t,r}$ we need estimates for $q_{a,t,r}$ and $\phi_{a,t,r}$
- Don't have complete data for these components so need to model

Migration

For migration, we have one data observation.

- From the 2009 census, people were asked “where did you live 1 year ago?”
- Can use this to get an observation for in- and out- (therefore net) migration for each region in 2008, call it $M_{a,2008,r}$
- How to include in model? Do we just assume $\phi_{a,2008,r} = M_{a,2008,r}$? How does this help us get other years?

Migration

Just like we have a data model and process model for population, we can have a data model and process model for migration

Let's model the $\phi_{a,t,r}$ as a random walk:

$$\begin{aligned} M_{a,t,r} &\sim N(\phi_{a,t,r}, 0.05^2) \\ \phi_{a,t,r} &\sim N(\phi_{a,t-1,r}, \sigma_\phi^2) \end{aligned}$$

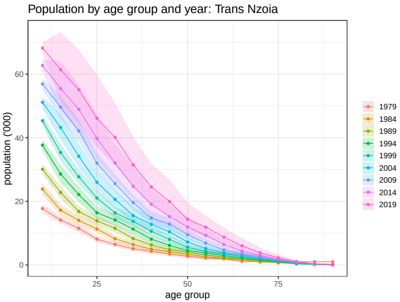
- The data model says we expect the observed net migration to be +/- 5% of the true net migration

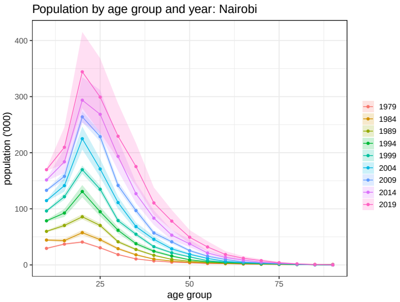
Now we have two data models

$$
\begin{aligned}
\log y_i &\sim N\left(\log p_{a[i],t[i],r[i]}, \sigma_y^2[i]\right) \\
p_{a,t,r} &= p_{a-1,t-1,r} \cdot \left[(1 - q_{a-1,t-1,r}) + \phi_{a-1,t-1,r}\right] \\
q_{a,t,r} &= \ldots \\
M_{a,t,r} &\sim N(\phi_{a,t,r}, 0.05^2) \\
\phi_{a,t,r} &\sim N(\phi_{a,t-1,r}, \sigma_\phi^2) \\
\text{other stuff} &= \ldots
\end{aligned}
$$

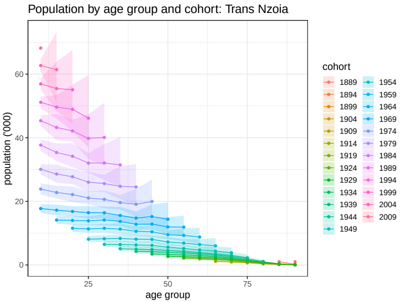
- There's a model on the death probabilities ($q_{a,t,r}$), too (basis functions derived from national mortality, ask if interested)
- "other stuff" is things like constraining the sum of our sub-national populations to be (close to) pre-published national population estimates from the UN

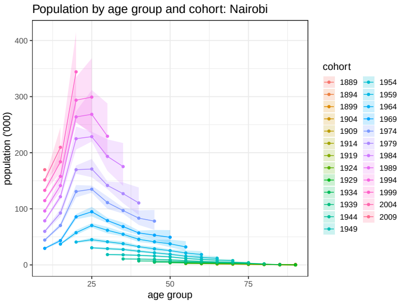
Preliminary results





Preliminary results: by cohort





Multi-level regression and post-stratification (MRP)

Dealing with non-representative surveys

- We generally want to use responses from surveys to form estimates of groups of interest (e.g. national, state-level opinions)
- To do this we need to ensure that the characteristics of the people surveyed are similar to the group of interest
- But getting representative survey responses is expensive
- Even if you have a good sampling frame, not guaranteed you will get a representative set of responses (people don't have phones, or don't answer them)
- Often better to over-sample people of interest, and the re-weight (post-stratify) to get representative estimates

Me trying to stay relevant

We have 25 people in our class. Let's say 12 of you did undergrad at UofT (48%), and the other 13 did undergrad somewhere else.

Say I was interested in the the proportion of graduate students at UofT that use TikTok.

- I did a survey of our class, and out of 25 people, 10 people use TikTok and 15 do not.
- Of the people who did undergrad at UofT, 4 person uses TikTok, and of those who didn't, 6 people use TikTok.

Based on our class survey, I could conclude that $10/25 = 40\%$ of graduate students use TikTok.

Post-stratification

- But say we knew that of all UofT grad students, 25% actually did undergrad at UofT.
- This is much lower than the proportion in our class
- A better estimate based on our survey, then, could be to post-stratify based on undergrad institution
- So our estimate of $\Pr(\text{TikTok}) = 4/12 * 0.25 + 6/13 * 0.75 =$ 43%

Post-stratify based on more characteristics

- It might make even more sense to post-stratify on other characteristics, like gender, age, undergraduate degree
- These are characteristics we might expect to be associated with TikTok usage
- But as we choose more post-stratifying variables, the cell count (i.e. the number of people in each group) gets smaller
- So our estimates become more uncertain

A more robust approach

Instead of taking raw counts by group, we could model the probability of using TikTok (y_i) in a hierarchical (multi-level regression), with covariates such as age, gender, undergrad degree, e.g.

$$\text{logit}^{-1}(Pr(y_i = 1)) = \beta_0 + \beta_1 \text{gender}_i + \beta_2 \text{age}_i + \beta_3 \text{degree}_i + \beta_4 \text{institution}_i$$

with $\beta_2 \sim (0, \sigma^2_{\text{age}})$ and $\beta_3 \sim (0, \sigma^2_{\text{degree}})$ i.e. the effects of age and degree are modeled hierarchically (too few groups with gender and institution).

- Why the hierarchical/multi-level set-up? Remember from radon, lip cancer, etc, that estimates for groups with small counts get shrunk toward the global mean, effectively placing less weight on the outcomes for groups where we have less information

MRP

$$\text{logit}^{-1}(Pr(y_i = 1)) = \beta_0 + \beta_1 \text{gender}_i + \beta_2 \text{age}_i + \beta_3 \text{degree}_i + \beta_4 \text{institution}_i$$

- Our model gives us **predicted** probabilities for TikTok usage by group.
- Once we have those, we can post-stratify as before to get a population-level (UofT-wide) estimate (and uncertainty)
- So the difference is we are using modeled proportions rather than raw proportions from the data
- Note that you must have info on post-stratification counts (cross-tabulated)! So in this example, need to know UofT counts by age, gender, undergrad degree, undergrad institution
- MRP often used to predict voting behavior, with post-stratification info coming from the census

MRP: further reading

- A famous paper, using surveys of Xbox users: Wang et al., "Forecasting elections with non-representative polls"
- Here's a worked example, where I used data from a survey about changing name after marriage: https://www.monicaalexander.com/posts/2019-08-07-mrp/

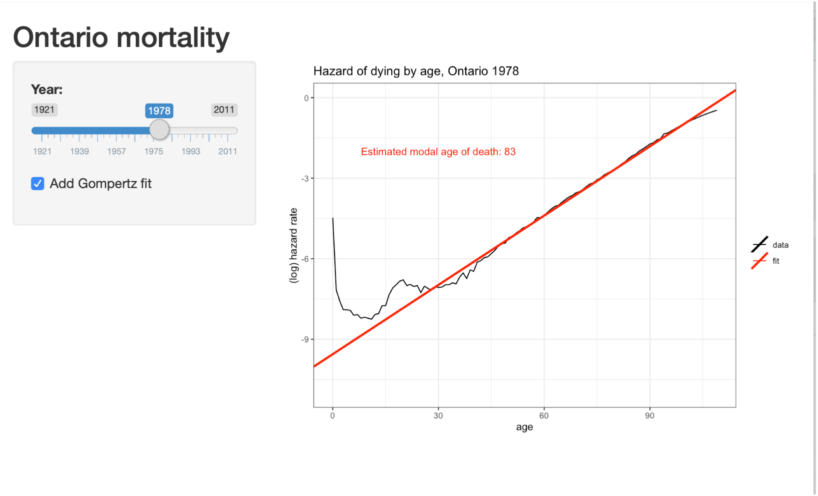
Cool things you can do with these coding skills

R Shiny

- An R package to create interactive web-based applications to visualize results
- Essentially inserts your R code (ggplot or otherwise) into functions that create an interactive interface so the user can change inputs based on a widget (slider, dropdown, etc)
- Can then host on the web for free with `shinyapps.io` or your own server

A simple example

https://monica-alexander.shinyapps.io/example_shiny/



more examples: https://shiny.rstudio.com/

A simple example

```r
library(tidyverse)
d <- read_rds("data/ON_mortality.RDS")

# Define a server for the Shiny app
function(input, output) {

  # Fill in the spot we created for a plot
  output$hazardPlot <- renderPlot({

    p <- d %>%
      mutate(age = as.numeric(age)) %>%
      filter(year==input$year) %>%
      ggplot(aes(age, log(hx))) +
      geom_line(aes(color = "data")) +
      #scale_y_log10() +
      theme_bw() +
      ylab("(log) hazard rate") +
      ggtitle(paste0("Hazard of dying by age, Ontario ", input$year )) +
      scale_color_manual(name = "", values = c("data" = "black", "fit" = "red")) +
      ylim(c(-11,0))

      if(input$addGompertz==FALSE){
        p
      }

    ...

  })
}
```

A simple example

```
library(tidyverse)
d <- read_rds("data/ON_mortality.RDS")

# Use a fluid Bootstrap layout
fluidPage(

  # Give the page a title
  titlePanel("Ontario mortality"),

  sidebarLayout(

    sidebarPanel(
      sliderInput("year",
                  "Year:",
                  value = 1960,
                  min = min(d$year),
                  max = max(d$year), sep = ""),
      checkboxInput("addGompertz", "Add Gompertz fit", FALSE)
      ),

    # Create a spot for the plot
    mainPanel(
      plotOutput("hazardPlot")
    )

  )
)
```

Blogdown

Websites with blogdown

- Consider making a website, if you don't have one already!
- If you are on the job market (academic or otherwise) people will Google you. It's a useful way to partially control what they see.
- Even before you're on the market, good to have, to build up a profile

Blogdown

- Blogdown is an R package that let's you create websites in RMarkdown
- Built by people at RStudio so nicely integrated
- Builds on website templates from Hugo (https://gohugo.io/)

Example wesbites built with blogdown:

- Mine: https://www.monicaalexander.com/
- Julia Silge: https://juliasilge.com/
- Sharla Gelfand: https://sharla.party/
- Alex Stringer: https://alexstringer.ca/

High-level steps

1. Create a new folder with an RStudio project. Best to make it a git repo also (e.g. my_website) because it will be easier to get online later
2. Choose a Hugo theme, hugo-academic is common one to start with. Then in Rstudio type

```
blogdown::new_site(theme = "gcushen/hugo-academic")
```

This will download a bunch of files into your folder and begin "serving" your site locally (i.e. within RStudio)

3. Add your own basic content. Some of this will be editing the `config.toml` file that got downloaded. You can also add a headshot (in the `static/img` folder).

High-level steps

4. Add more detailed content.

- If you look at the the `content` folder, for hugo-academic there are some markdown (.md, similar to .Rmd) files called things like `about.md`, `publications.md` etc. Can edit as neccessary.
- If you want to add blog posts written in RMarkdown, you can add them in the `content/post` folder.
- This step will invole a lot of playing around and editing to get things how you want. There's lots of help online, and I've included some good blog posts and resources below.
- To come back your website once you've closed R Studio, open the RStudio project, then type 'blogdown:::serve_site()' into the console to serve your site and then continue editing.

5. Make your website public

- Commit and push to GitHub. Then two options: deploy using Netlify or GitHub Pages

Blogdown: further resources

Lots of good resources out there, here's a selection

- https://bookdown.org/yihui/blogdown/
- https://alison.rbind.io/post/2017-06-12-up-and-running-with-blogdown/
- https://masalmon.eu/2020/02/29/hugo-maintenance/
- https://djnavarro.net/post/starting-blogdown/

Exam etc

Thank you :)