

STA2201H Methods of Applied Statistics II

Monica Alexander

Week 1: Introduction

Overview

- ▶ Introductions
- ▶ Course outline and goals
- ▶ Tools
- ▶ GLMs review (start)
- ▶ Lab: Git, tidyverse, R Markdown/Quarto

Introductions

Instructor

- ▶ Monica Alexander
- ▶ Email: monicaalexander@utoronto.ca.
- ▶ Office hours time: 2-3pm Wednesdays, 700 University Level 9 Room 9135

TA

- ▶ Michael Chong
- ▶ Email: myc.chong@mail.utoronto.ca.

Course outline and goals

Web

- ▶ Github: <https://github.com/MJAlexander/applied-stats-2024>
 - ▶ slides, labs, and data will be put here
- ▶ Quercus page
 - ▶ go here to submit assignments (not labs)
 - ▶ links to discussion boards
 - ▶ class announcements will be put here

Course outline

- ▶ Topics will include generalized linear models, Bayesian inference, hierarchical models, generalized additive models involving non-parametric smoothing, temporal models, model evaluation and selection. We will also cover some core statistical computing techniques.
- ▶ A large focus of the outcomes on this course will also be on reproducible research, identifying and dealing with data and modeling issues, and model interpretation and communication.
- ▶ The focus in terms of methods is advanced regression techniques, fit using Bayesian inference. The focus in terms of coding/computation is becoming more comfortable and adept at efficient, reproducible coding and workflows (data, analysis, reporting and communicating results)

Course outline

- ▶ Throughout the course we will be using R in all examples, labs and homework assignments.
- ▶ Each week will be a lecture (~1.5hrs) then a lab

Assessment

- ▶ Lab exercises, 8 in total, 2.5% each
 - ▶ Due 9am the following Monday
 - ▶ **Hand in via git**
 - ▶ Practice of concepts covered in the lecture
- ▶ Three assignments, 10% each
 - ▶ Mostly data analysis, R heavy
 - ▶ **Hand in via Quercus**
- ▶ Mid-term, 15%
 - ▶ After reading week
 - ▶ Assesses week 1-6
 - ▶ Short answer
- ▶ Research project 35%
 - ▶ Pick a dataset, research question and statistical approach (that is covered in class)
 - ▶ Research proposal (7.5%)
 - ▶ Research paper (20 %)
 - ▶ Presentation last week of class (7.5%)

Expectations

We will be doing applied statistics in the truest sense of the term

- ▶ Understand main ideas behind important techniques for applied statistics
- ▶ Coding in R (and in particular, the tidyverse, ggplot)
- ▶ Dealing with real data!
- ▶ R markdown or Quarto
- ▶ Git (terminal or desktop, not direct file upload)
- ▶ Code readability
- ▶ Clear communication of methods, findings, limitations
 - ▶ Data exploration is part of this!
- ▶ Aim for reproducible research

Expect a lot of coding, some “open ended-ness”

Research project

- ▶ The goal is to write a short applied statistics paper in the academic style
- ▶ Intro/background (some reference to previous literature/work!), data, methods, results, discussion, limitation
- ▶ Increased length does not equal increased quality
- ▶ (Increased number of graphs does not equal increased quality of EDA)
- ▶ Should be written in RMarkdown/Quarto (ideally self-contained, but if data/code too big/slow, qmd should call scripts in a reproducible way)

Research project tips

- ▶ Start thinking about it now
- ▶ We will look at regression techniques to deal with
 - ▶ a range of outcomes (continuous, binary, categorical, counts)
 - ▶ nested groups (e.g. individuals within schools within districts within provinces)
 - ▶ non-representative surveys
 - ▶ time series (can have missing data! can have multiple observations!)
 - ▶ measurement error
- ▶ Think of a question → find a dataset → if you can't find a dataset then maybe change your question :)
- ▶ Must be different to AS1 projects
- ▶ Please try and avoid Kaggle and UCI

Course roadmap

Subject to change depending on time and priorities.

Planned lecture content:

- ▶ Generalized linear models recap
- ▶ Bayesian inference
- ▶ Visualizing the Bayesian workflow and model checks
- ▶ Multilevel models
- ▶ Non-linear/ non-parametric models (Penalized splines)
- ▶ Temporal models / dealing with correlation
- ▶ Time/interest permitting: text analysis?

Roadmap

Planned lab content:

- ▶ Rmarkdown/Quarto, git
- ▶ Tidyverse
- ▶ EDA, data viz
- ▶ RShiny
- ▶ Stan
- ▶ Probably: web scraping
- ▶ Maybe: Extracting data from API (e.g. Facebook Spotify/Genius)

Motivating example

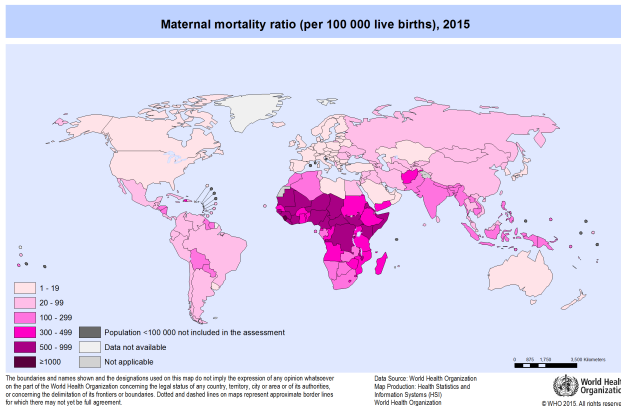
Global estimation of the causes of maternal death

- ▶ **Maternal mortality:** the death of a woman while pregnant or within 42 days of termination of pregnancy, from any cause related to or aggravated by the pregnancy.
- ▶ Very important indicator of health and development of a country
- ▶ Part of the Sustainable Development Goals (3.1)



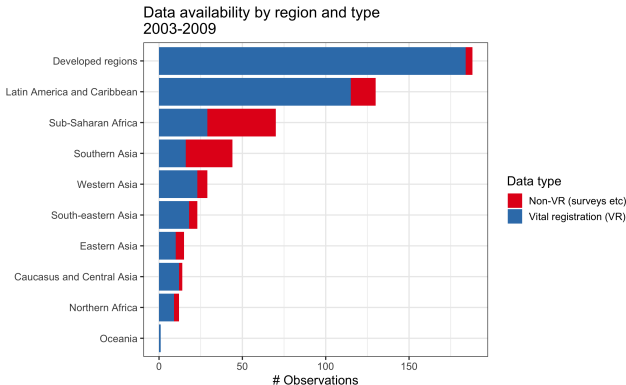
Global estimation of the causes of maternal death

- ▶ Large variation in maternal mortality ratio (deaths per 100,000 births) across the world (highest: 1150; lowest: 2)
- ▶ In order to reduce number of deaths, need to know underlying causes
- ▶ But this is difficult information to obtain/estimate



How do we get information on causes of (maternal) death?

- ▶ In high-income countries and some middle-income countries: civil registration systems
- ▶ In low-income countries: ???
 - ▶ surveys (why is this hard?)
 - ▶ facility-based administrative data
 - ▶ other specialized studies



How do we get information on causes of (maternal) death?

- ▶ If we had complete coverage of all deaths and a reliable way of classifying cause of death, then we could just count deaths and call it a day
- ▶ But in most countries (particularly high-burden countries) we have very little information, and what we do have is full of problems
- ▶ → Use statistical methods to obtain as reliable estimates as possible

Issues

To name a few:

- ▶ Years with no data
- ▶ Only some causes observed (even in high-income countries)
- ▶ Non-representative data (subnational, facility-based)
- ▶ Cause of death classification issues (death not witnessed, definition changes, differences across countries etc)
- ▶ Under/over-reporting (especially abortion)
- ▶ Not all civil registration systems are high quality
- ▶ Low death counts (~ 25 deaths in Australia)

Intro to statistical set-up

Notation:

- ▶ observations $i = 1, \dots, n$
- ▶ d_i is total number of maternal deaths for the i th observation
- ▶ observed maternal deaths $\mathbf{y}_i = (y_{i,1}, \dots, y_{i,7})$
- ▶ $y_{i,j}$ is the number of deaths due to cause j for the i th observation
- ▶ cause groups $j = 1, \dots, 7$ corresponding to {ABO, EMB, HEM, SEP, DIR, IND, HYP}

Intro to statistical set-up

Think of deaths as a stochastic process:

- ▶ Given total number of maternal deaths d_i , the probability of a death is due to cause j is $p_{i,j}$. This is a Multinomial distribution, with 7 categories:

$$\mathbf{y}_i \sim \text{Multinomial}(d_i, \mathbf{p}_i)$$
$$\mathbf{p}_i = (p_{i,1}, \dots, p_{i,7})$$

- ▶ We observe $y_{i,j}$ and d_i
- ▶ We are interested in estimating \mathbf{p}_i . These will help us get estimates for the 'true' proportions \mathbf{p}_c for countries $c = 1, \dots, 193$ (UN member countries)

Intro to statistical set-up

$$\mathbf{y}_i \sim \text{Multinomial}(d_i, \mathbf{p}_i)$$

$$\mathbf{p}_i = (p_{i,1}, \dots, p_{i,7})$$

Put a model on \mathbf{p}_i :

- ▶ Transform to ensure probabilities sum to 1
- ▶ Model can include effects/adjustments for different things e.g. region, data quality, temporal changes, subnational adjustments. . .
- ▶ This is a (Bayesian) hierarchical model. We will learn about these!

More info, see paper: <https://arxiv.org/abs/2101.05240>

Maternal mortality summary

- ▶ Real world problem, working with WHO and statisticians, epidemiologists, clinicians, public health officials
- ▶ So many data problems
- ▶ Data complexities lead to relatively complex models
- ▶ Substantive area knowledge helps to understand data issues
- ▶ Results have big impact (policy, \$\$\$): need to be careful, transparent with assumptions, reproducible

Tools

Tools

- ▶ R
- ▶ Tidyverse
- ▶ RMarkdown/Quarto
- ▶ git

R

We will be using R in this course. Pros:

- ▶ Free
 - ▶ reproducibility
 - ▶ portability
- ▶ Open
 - ▶ large community
 - ▶ lots of packages
 - ▶ lots of help

RStudio:

- ▶ IDE for R that makes using R a lot nicer and easier
- ▶ If you haven't already got it, download the free version here:
<https://posit.co/download/rstudio-desktop/>

Tidyverse

- ▶ R Packages contain R functions, the documentation that describes how to use them, and sample data.
- ▶ The 'tidyverse' is "an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures."
<https://www.tidyverse.org/>
- ▶ ggplot probably the most well known
- ▶ Style of coding fundamentally different to base R.
- ▶ A lot of other packages now produce output objects in the 'tidy' form

RMarkdown

- ▶ Markdown is plain text formatting syntax that can be converted into lots of different outputs (eg HTML, PDF)
- ▶ R Markdown allows you to combine Markdown (for the report writing) and embedded R chunks, which are dynamically updated when the document is compiled
- ▶ R code can be in chunks or inline (e.g the fourth root of π is 0.7853982)
- ▶ These slides are written in RMarkdown and knitted to PDF (beamer)

```
282 - ## RMarkdown
283
284 - Markdown is plain text formatting syntax that can be converted into lots of different outputs (eg HTML, PDF)
285 - R Markdown allows you to combine Markdown (for the report writing) and embedded R chunks, which are dynamically updated when
the document is compiled
286 - R code can be in chunks or inline (e.g the fourth root of  $\pi$  is `r pi*(1/4)`)
287 - These slides are written in RMarkdown and knitted to PDF (beamer)
288
289 \begin{figure}
290 \includegraphics[width = 0.8\textwidth]{turtles.png}
291 \end{figure}
```

RMarkdown

- ▶ Good reproducibility tool
- ▶ Can do most things you can do in LaTeX (writing math is the same)
- ▶ You are expected to write up assignments in RMarkdown or Quarto

Quarto

- ▶ The newer version of R Markdown
- ▶ Mostly the same, a few bits and pieces improved
- ▶ You can use either R Markdown or Quarto to do labs / assignments

git

- ▶ git is a version control system (think a more complicated Dropbox)
- ▶ Designed for software engineers, but useful for all sorts of code
- ▶ Useful for both collaborative and solo projects
- ▶ GitHub is useful place to host open source projects

GLMs recap

Motivating examples

Outcomes we may be interested in investigating (in relation to other explanatory variables):

- ▶ Police stop and frisks in NYC
- ▶ Infant deaths in the US
- ▶ Who voted for the Liberal party v other party
- ▶ Who voted Liberal, Conservatives, LDP
- ▶ Concentration of drug at particular times after ingestion

The take-away: none of these are Normal.

General linear models

Let's start with a recap of general linear models. We observe y_1, y_2, \dots, y_n which are realizations of the random variables Y_1, Y_2, \dots, Y_n

In linear models the y_i 's have two pieces:

1. A **systematic part**, with the form

$$E(\mathbf{Y}|\mathbf{X}) = \mu = \mathbf{X}\beta$$

2. A **random part**, where errors are assumed to be i.i.d such that $E[\epsilon] = 0$ and $var[\epsilon] = \sigma^2$. We usually further assume that errors are Normal with constant variance σ^2 .

Multiple linear regression

One of the most common examples of a general linear model.

Goal: we are trying to measure the association between response/outcome/dependent variable Y_i and one or more explanatory variables/covariates $X_{i,1}, X_{i,2}, \dots, X_{i,k}$

- ▶ The conditional expectation function (CEF)
 $E(Y_i | X_{i,1}, X_{i,2}, \dots, X_{i,k})$ describes the expected value (population mean) of Y_i given values of the variables $X_{i,1}, X_{i,2}, \dots, X_{i,k}$.
- ▶ The CEF has two properties that make it a good measure of statistical association:
 - ▶ The CEF decomposition property
 - ▶ The CEF ANOVA property

Multiple linear regression

MLR is a model for the CEF:

$$\begin{aligned} Y_i &= E(Y_i \mid X_{i,1}, X_{i,2}, \dots, X_{i,k}) + \varepsilon_i \\ &= \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \dots + \beta_k X_{ik} + \varepsilon_i \end{aligned}$$

Specifically, the most basic MLR model is a simple linear function of the X 's and associated parameters β .

An aside: what can this model tell us?

$$\begin{aligned} Y_i &= E(Y_i \mid X_{i,1}, X_{i,2}) + \varepsilon_i \\ &= \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \varepsilon_i \end{aligned}$$

The MLR assumptions

1. no model misspecification
2. there is independent variation in all of the explanatory variables
 - ▶ In other words, none of the explanatory variables are constants, and there are no perfect linear relationships among the explanatory variables
 - ▶ e.g. can't have $X_{i1} = X_{i2} + X_{i3}$
3. All variables are from a simple random sample
 - ▶ This assumption implies that all members of a population have an equal probability of selection, that all possible samples of size n have an equal probability of selection, and that each observation is independent of all the others

The MLR assumptions

4. The variance of $\varepsilon_i = Y_i - E(Y_i | X_{i1}, X_{i2}, \dots, X_{ik})$ is the same across all values of the explanatory variables
i.e. $\text{Var}(\varepsilon_i | X_{i1}, X_{i2}, \dots, X_{ik}) = \sigma^2$
 - ▶ This is called homoskedasticity
5. The normality assumption $\varepsilon_i = Y_i - E(Y_i | X_{i1}, X_{i2}, \dots, X_{ik})$ is normally distributed

Assumption 1-4 are Gauss-Markov assumptions

Estimation

Minimizing the sum of squared residuals

$$S(\beta) = \sum_{i=1}^n \left(y_i - \mathbf{x}_i^T \beta \right)^2 = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta)$$

Leads to the MLR-OLS estimator

$$\hat{\beta} = \left(\mathbf{X}^T \mathbf{X} \right)^{-1} \mathbf{X}^T \mathbf{y}$$

Sampling distribution of the MLR-OLS estimator

- Under the Gauss Markov and normality assumptions, the OLS estimator, $\hat{\beta}_k$ is normally distributed with a mean equal to

$$E(\hat{\beta}_k) = \beta_k$$

and variance

$$\text{Var}(\hat{\beta}_k) = \frac{\sigma^2}{\sum_i (X_{ik} - \bar{X}_{ik})^2 (1 - R_k^2)}$$

We can use this property for inference: The sampling distribution of standard error standardized estimator follows a t-distribution with $n - (k + 1)$ degrees of freedom.

General linear models

$$\begin{aligned}Y_i &\sim N(\mu_i, \sigma^2) \\ \mu_i &= \mathbf{X}_i^T \beta\end{aligned}$$

General linear models are not appropriate when

- ▶ The range of Y is restricted
- ▶ The variance of Y depends on the mean

Generalized Linear Models extend the classical set-up to allow for a wider range of distributions. Introduced by Nelder and Wedderburn (1972) [Later, GAMs in 1990].

Generalized linear models

Generalized linear models

GLMs have an additional piece on top of the classical linear models:

1. **random component:** $Y_i \sim$ some distribution with $E[Y_i|\mathbf{X}_i] = \mu_i$
 2. **systematic component:** $\mathbf{X}_i^T \beta$
 3. The **link function** that links the random and systematic components $g(u_i) = \mathbf{X}_i^T \beta$
- ▶ Set-up is almost the same, particularly in terms of specifying a good linear predictor $\mathbf{X}_i^T \beta$
 - ▶ Just need to think about the link and the distribution of the outcome

GLMs

$$\begin{aligned}Y_i &\sim G(\mu_i, \phi) \\ E[Y_i | \mathbf{X}_i] &= \mu_i \\ g(\mu_i) &= \mathbf{X}_i^T \beta\end{aligned}$$

► ϕ is the scale parameter.

What can Y be distributed as? In principle, anything. In practice (and original formulation), distributions come from the **exponential family**.

Exponential Family

Exponential Family

The random variable Y belongs to the exponential family of distributions if its support does not depend upon any unknown parameters and its density or probability mass function takes the form

$$p(y|\theta, \phi) = \exp \left(\frac{y\theta - b(\theta)}{\phi} + c(y, \phi) \right)$$

- ▶ $\theta = h(\mu)$ depends on the expected value of y and is the **canonical parameter**
- ▶ ϕ is the scale parameter (if known: one-parameter family)
- ▶ b and c are arbitrary functions

Example: Poisson distribution

$$p(y|\theta, \phi) = \exp \left(\frac{y\theta - b(\theta)}{\phi} + c(y, \phi) \right)$$

Poisson:

$$p(y|\mu) = \frac{\mu^y e^{-\mu}}{y!}$$

Write as

$$p(y|\mu) = \exp \{y \log \mu - \mu - \log y!\}$$

- ▶ $\theta = \log \mu$
- ▶ $b(\theta) = e^\theta$
- ▶ $c(y, \phi) = -\log y!$
- ▶ Note that the scale parameter $\phi = 1$ so the variance is entirely determined by the mean

Example: Normal distribution

$$p(y|\theta, \phi) = \exp \left(\frac{y\theta - b(\theta)}{\phi} + c(y, \phi) \right)$$

Normal:

$$p(y|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{(y - \mu)^2}{2\sigma^2} \right\}$$

Write as

$$p(y|\mu, \sigma^2) = \exp \left\{ \frac{y\mu - \frac{1}{2}\mu^2}{\sigma^2} - \frac{1}{2} \left[\frac{y^2}{\sigma^2} + \log(2\pi\sigma^2) \right] \right\}$$

- ▶ $\theta = \mu$
- ▶ $b(\theta) = \frac{1}{2}\theta^2$
- ▶ $\phi = \sigma^2$
- ▶ $c(y, \phi) = -\frac{1}{2} \left[\frac{y^2}{\sigma^2} + \log(2\pi\sigma^2) \right]$

Other examples

Other common examples:

- ▶ Binomial
- ▶ Gamma
- ▶ Negative binomial
- ▶ Inverse Gaussian

Lab

Lab this week

- ▶ Setting up git, with a small exercise
- ▶ Intro to tidyverse and ggplot

Git

- ▶ Git is a version control system
- ▶ The system tracks changes you make to git repositories ('repos')
- ▶ Think of repos as folders
- ▶ In order for file versions to be tracked, they need to be **committed** to the git repo
- ▶ Think of committing as like saving, but with slightly more steps

GitHub

<https://github.com/>

- ▶ A hosting service for git repos
- ▶ You can sign up for free, and host an unlimited number of public or private repos
- ▶ You will be submitting lab exercises via GitHub, so you need to set up an account!

The simplest Git/GitHub workflow

- ▶ New repo on GitHub
- ▶ Clone onto local computer
- ▶ Do work on local computer
- ▶ Save
- ▶ Add and commit to git repo
- ▶ Push to GitHub (this means your new work will appear on the GitHub website)

If you are working on your own, on one computer, this is it!

Git/GitHub

- ▶ If you are working on a couple of different computers / servers, you may also need to **pull** from GitHub to update any new work done elsewhere
- ▶ Git is designed for collaborative work. More complicated workflows have branches, pull requests, merges (more later)

Steps on GitHub

Monica to now demonstrate

- ▶ creating a new repository
- ▶ Adding Monica and Michael as collaborators
 - ▶ MJAlexander
 - ▶ michael-chong

Then using the terminal (or GitHub Desktop?):

- ▶ cloning to your computer
- ▶ doing some work
- ▶ git status
- ▶ add, commit, push

Disclaimer: I use the terminal. You are welcome to use the GitHub Desktop: <https://desktop.github.com/> But do not use direct file upload.

This week's lab assessment (git part)

- ▶ Make a repo and add me and Michael as collaborators
- ▶ Add a text file with your name, program (e.g. statistics Masters), and your favorite thing to do in Toronto
- ▶ Push changes to GitHub