

# Beamer Class Demonstration<sup>1</sup>

Me Myself and I

IQSS

March 29, 2018

---

<sup>1</sup>for beamer

# Outline

## Beamer Features

### Some of Gary's Examples

## Other features

### Structural Features

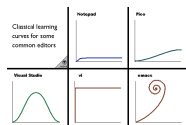
## More Features

### Blocks

# What's this course about?

- Specific statistical methods for many research problems
  - How to learn (or create) new methods
  - Inference:  
Using facts you know to learn about facts you don't know
- How to write a publishable scholarly paper
- All the practical tools of research — theory, applications, simulation, programming, word processing, plumbing, whatever is useful
- ~→ Outline and class materials:

[j.mp/G2001](http://j.mp/G2001)



- The syllabus gives topics, not a weekly plan.
- We will go as fast as possible subject to everyone following along
- We cover different amounts of material each week

# How much math will you scare us with?

- All math requires two parts: **proof** and **concepts & intuition**
- Different classes emphasize:
  - **Baby Stats**: dumbed down proofs, vague intuition
  - **Math Stats**: rigorous mathematical proofs
  - **Practical Stats**: deep concepts and intuition, proofs when needed
    - Goal: how to do empirical research, in depth
    - Use rigorous statistical theory — when needed
    - Insure we understand the intuition — always
    - Always traverse from theoretical foundations to practical applications
    - Includes “how to” computation
    - $\rightsquigarrow$  Fewer proofs, more concepts, better practical knowledge
- Do you have the background for this class? **A Test: What's this?**

$$b = (X'X)^{-1}X'y$$

# Systematic Components: Examples



- $E(Y_i) \equiv \mu_i = X_i\beta = \beta_0 + \beta_1 X_{1i} + \dots + \beta_k X_{ki}$
- $\Pr(Y_i = 1) \equiv \pi_i = \frac{1}{1+e^{-x_i\beta}}$
- $V(Y_i) \equiv \sigma_i^2 = e^{x_i\beta}$
- Interpretation:
  - Each is a **class of functional forms**
  - Set  $\beta$  and it picks out one **member of the class**
  - $\beta$  in each is an “effect parameter” vector, with different meaning

# Negative Binomial Derivation

Recall:

one two three

# Negative Binomial Derivation

Recall:

$$\Pr(A|B) = \frac{\Pr(AB)}{\Pr(B)} \implies \Pr(AB) = \Pr(A|B) \Pr(B)$$

one two three

# Negative Binomial Derivation

Recall:

$$\Pr(A|B) = \frac{\Pr(AB)}{\Pr(B)} \implies \Pr(AB) = \Pr(A|B) \Pr(B)$$

one two three

$$\text{NegBin}(y|\phi, \sigma^2) = \int_0^\infty \text{Poisson}(y|\lambda) \times \text{gamma}(\lambda|\phi, \sigma^2) d\lambda$$



# Negative Binomial Derivation

Recall:

$$\Pr(A|B) = \frac{\Pr(AB)}{\Pr(B)} \implies \Pr(AB) = \Pr(A|B) \Pr(B)$$

one two three

$$\begin{aligned}\text{NegBin}(y|\phi, \sigma^2) &= \int_0^\infty \text{Poisson}(y|\lambda) \times \text{gamma}(\lambda|\phi, \sigma^2) d\lambda \\ &= \int_0^\infty \mathbb{I}(y, \lambda|\phi, \sigma^2) d\lambda\end{aligned}$$

# Negative Binomial Derivation

Recall:

$$\Pr(A|B) = \frac{\Pr(AB)}{\Pr(B)} \implies \Pr(AB) = \Pr(A|B) \Pr(B)$$

one two three

$$\begin{aligned}\text{NegBin}(y|\phi, \sigma^2) &= \int_0^\infty \text{Poisson}(y|\lambda) \times \text{gamma}(\lambda|\phi, \sigma^2) d\lambda \\ &= \int_0^\infty \mathbb{I}(y, \lambda|\phi, \sigma^2) d\lambda \\ &= \frac{\Gamma\left(\frac{\phi}{\sigma^2-1} + y_i\right)}{y_i! \Gamma\left(\frac{\phi}{\sigma^2-1}\right)} \left(\frac{\sigma^2-1}{\sigma^2}\right)^{y_i} (\sigma^2)^{\frac{-\phi}{\sigma^2-1}}\end{aligned}$$

# Outline

Beamer Features

Some of Gary's Examples

Other features

Structural Features

More Features

Blocks

# Structural Features

## Levels of Structure

- usual  $\text{\LaTeX}$  `\section`, `\subsection` commands
- ‘frame’ environments provide slides
- ‘block’ environments divide slides into logical sections
- ‘columns’ environments divide slides vertically (example later)
- overlays (à la prosper) change content of slides dynamically

## Example (Overlay Alerts)

On the first overlay, **this text** is highlighted (or *alerted*).

On the second, this text is.

# Structural Features

## Levels of Structure

- usual  $\text{\LaTeX}$  `\section`, `\subsection` commands
- ‘frame’ environments provide slides
- ‘block’ environments divide slides into logical sections
- ‘columns’ environments divide slides vertically (example later)
- overlays (à la prosper) change content of slides dynamically

## Example (Overlay Alerts)

On the first overlay, this text is highlighted (or *alerted*).

On the second, **this text** is.

## Code blocks

```
# Say hello in R  
hello <- function(name) paste("hello", name)
```

# Code blocks

```
# Say hello in R  
hello <- function(name) paste("hello", name)
```

```
# Say hello in Python  
def hello(name):  
    return("Hello" + " " + name)
```

# Code blocks

```
# Say hello in R  
hello <- function(name) paste("hello", name)
```

```
# Say hello in Python  
def hello(name):  
    return("Hello" + " " + name)
```

```
-- Say hello in Haskell  
hello name = "Hello" ++ " " ++ name
```



# Code blocks

```
# Say hello in R
hello <- function(name) paste("hello", name)
```

```
# Say hello in Python
def hello(name):
return("Hello" + " " + name)
```

```
-- Say hello in Haskell
hello name = "Hello" ++ " " ++ name
```

```
/* Say hello in C */
#include <stdio.h>
int main()
{
    char name[256];
    fgets(name, sizeof(name), stdin);
    printf("Hello %s", name);
    return(0);
}
```

# Alerts

- First level alert
- Second level alert
- Third level alert
- Fourth level alert
- Fifth level alert

# Outline

- Beamer Features

  - Some of Gary's Examples

- Other features

  - Structural Features

- More Features

  - Blocks

# Other Features

## Levels of Structure

- Clean, extensively customizable visual style
- Hyperlinks ([click here](#))
- No weird scaling prosper
  - slides are 96 mm × 128 mm
  - text is 10-12pt on slide
  - slide itself magnified with Adobe Reader/xpdf/gv to fill screen
- pgf graphics framework easy to use
- include external JPEG/PNG/PDF figures
- output directly to pdf: no PostScript hurdles
- detailed User's Manual (with good presentation advice, too)

# Theorems and Proofs

The proof uses *reductio ad absurdum*.

## Theorem

*There is no largest prime number.*

## Proof.

1. Suppose  $p$  were the largest prime number.
2. Consider the number  $p + 1$ .
3.  $p + 1$  is greater than 1, thus divisible by some prime number not in the first  $p$  numbers.



# Theorems and Proofs

The proof uses *reductio ad absurdum*.

## Theorem

*There is no largest prime number.*

## Proof.

1. Suppose  $p$  were the largest prime number.
2. Let  $q$  be the product of the first  $p$  numbers.
3.  $q + 1$  is greater than 1, thus divisible by some prime number not in the first  $p$  numbers.



# Theorems and Proofs

The proof uses *reductio ad absurdum*.

## Theorem

*There is no largest prime number.*

## Proof.

1. Suppose  $p$  were the largest prime number.
2. Let  $q$  be the product of the first  $p$  numbers.
3. Then  $q + 1$  is not divisible by any of them.
4. But  $q + 1$  is greater than 1, thus divisible by some prime number not in the first  $p$  numbers.



# Blocks

## Normal block

A **set** consists of elements.

## Alert block

$2 = 2$ .

## Example block

The set  $\{1, 2, 3, 5\}$  has four elements.