

Poisson regression

Overview

Review and continuation of logistic regression

Poisson regression

If there is time: Data cleaning and wrangling with the tidyverse

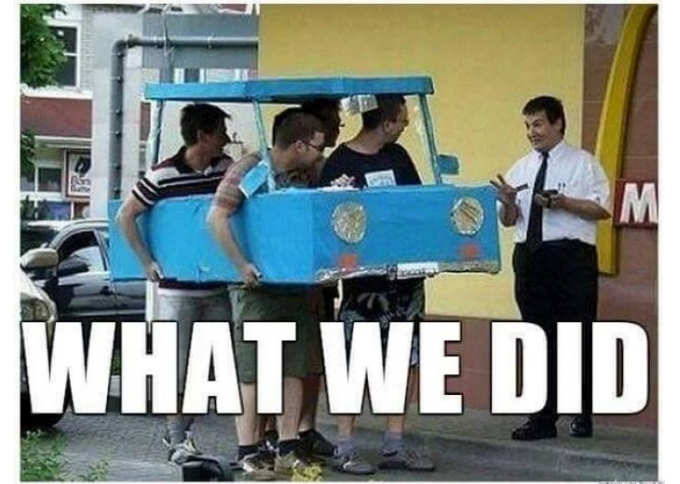
- Processing dates
- Manipulating strings
- Reshaping data

Announcement

Homework 9 is due Sunday November 26th

- i.e., the Sunday at the end of the Thanksgiving break

Keep thinking about your final project!



Very quick review of logistic regression

In **logistic regression** we try to predict if a case belongs to one of two categories

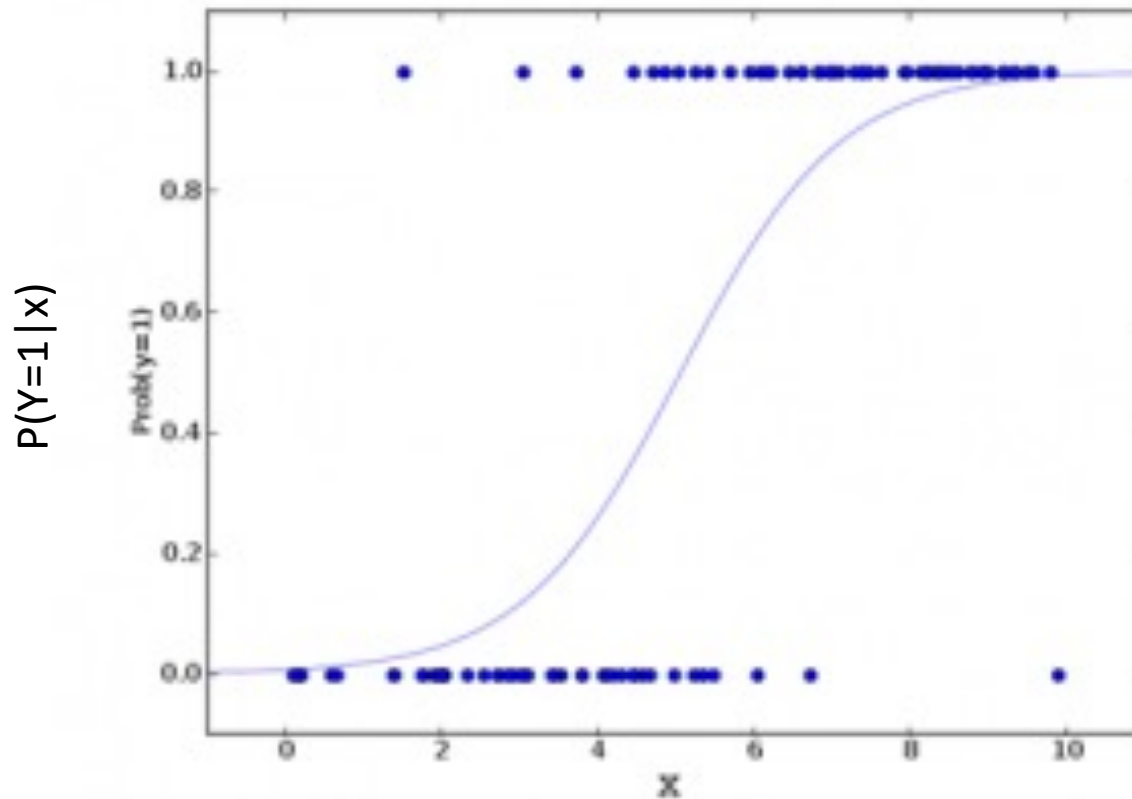
To do this we model the log-odds as a linear function of predictor variables:

$$\log\left(\frac{P(Y=1|x)}{1-P(Y=1|x)}\right) = \beta_0 + \beta_1 \cdot x$$

If we write the above equation in terms of the probability of being in class a we get:

$$P(Y = 1|x) = \frac{\exp(\beta_0 + \beta_1 \cdot x_1)}{1 + \exp(\beta_0 + \beta_1 \cdot x_1)} = \frac{e^{\beta_0 + \beta_1 \cdot x_1}}{1 + e^{\beta_0 + \beta_1 \cdot x_1}}$$

Very quick review of logistic regression



$$P(Y = 1|x_1) = \frac{e^{\beta_0 + \beta_1 \cdot x_1}}{1 + e^{\beta_0 + \beta_1 \cdot x_1}}$$

Very quick review of logistic regression

We can easily extend our logistic regression model to include multiple explanatory variables

$$\log\left(\frac{P(Y=1|x)}{1-P(Y=1|x)}\right) = \hat{\beta}_0 + \hat{\beta}_1 \cdot x_1 + \hat{\beta}_2 \cdot x_2 + \dots + \hat{\beta}_k \cdot x_k$$

When using a categorical predictor, x_2 , in a logistic regression model, the exponential of the regression coefficient $e^{\hat{\beta}_2}$ is the **odds ratio**

- Tells us how many times greater the odds are when $x_2 = 1$ vs. when $x_2 = 0$

We can fit logistic regression models in R using the `glm()` function

```
> lr_fit <- glm(rank_name ~ salary_tot, data = assistant_full_data, family = "binomial")
```

Questions?



Let's continue exploring logistic regression in R...

Poisson regression

Summary of linear regression

We can summarize the linear regression model as:

$$Y_i = \mu_i + \varepsilon_i \quad \text{where } \varepsilon_i \sim N(0, \sigma_\varepsilon)$$

$$\mu_i = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k$$

Equivalently, $Y_i \sim N(\beta_0 + \beta_1 x_1 + \dots + \beta_k x_k, \sigma_\varepsilon)$

Generalized linear models

We can summarize the linear regression model as:

$$Y_i = \mu_i + \varepsilon_i \quad \text{where } \varepsilon_i \sim N(0, \sigma_\varepsilon)$$
$$\mu_i = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k$$

In generalized linear models, we generalize the model to:

$$Y_i \sim f(y|\theta_i) \quad \text{where } f(y|\theta_i) \text{ is some probability distribution}$$
$$\theta_i = g^{-1}(\beta_0 + \beta_1 x_1 + \dots + \beta_k x_k)$$

g^{-1} is called an "inverse link function"
Links "linear predictor" to parameters

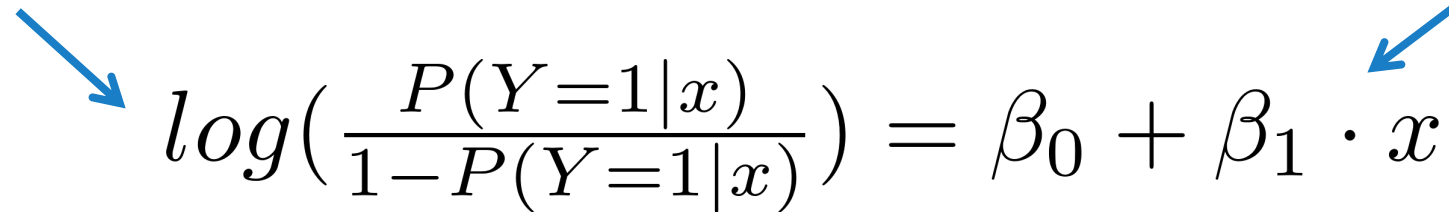
We choose a particular "family" of distributions (e.g., Poisson, binomial, etc.)

Example: logistic regression

In logistic regression we model whether a case belongs to one of two categories

- $P(Y = 0 \mid \mathbf{x})$ or $P(Y = 1 \mid \mathbf{x})$

The logit function (log-odds) is a "link function"

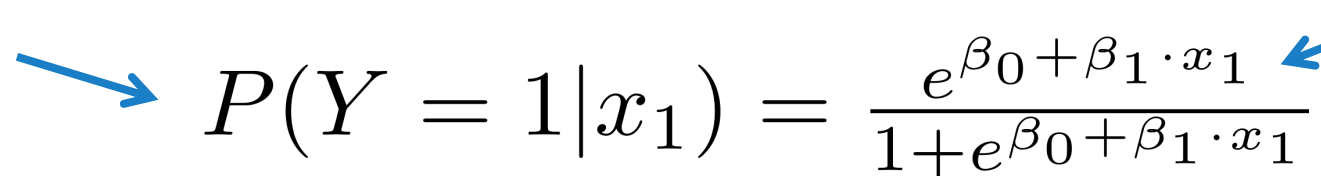


The diagram shows the equation $\log\left(\frac{P(Y=1|x)}{1-P(Y=1|x)}\right) = \beta_0 + \beta_1 \cdot x$. A blue arrow points from the text "The logit function (log-odds) is a 'link function'" to the log function. Another blue arrow points from the text "'Linear predictor'" to the right-hand side of the equation, $\beta_0 + \beta_1 \cdot x$.

$$\log\left(\frac{P(Y=1|x)}{1-P(Y=1|x)}\right) = \beta_0 + \beta_1 \cdot x$$

Inverse link function
(logistic function)

Solving for $P(Y = 1 \mid x)$



The diagram shows the equation $P(Y = 1|x_1) = \frac{e^{\beta_0 + \beta_1 \cdot x_1}}{1 + e^{\beta_0 + \beta_1 \cdot x_1}}$. A blue arrow points from the text "Solving for P(Y = 1 | x)" to the left-hand side of the equation, $P(Y = 1|x_1)$.

$$P(Y = 1|x_1) = \frac{e^{\beta_0 + \beta_1 \cdot x_1}}{1 + e^{\beta_0 + \beta_1 \cdot x_1}}$$

Family is Bernoulli distribution
(binomial with $n = 1$)



The diagram shows the equation $Y_i \sim \text{Bernoulli}(P(Y = 1|x))$. A blue arrow points from the text "Family is Bernoulli distribution (binomial with n = 1)" to the left-hand side of the equation, Y_i .

$$Y_i \sim \text{Bernoulli}(P(Y = 1|x))$$

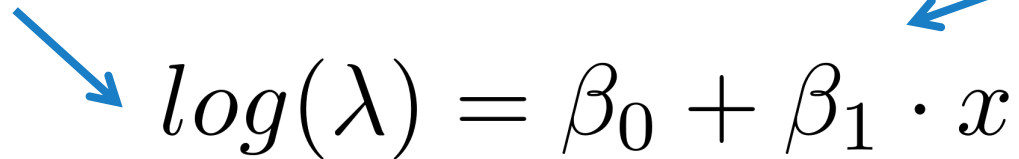
R: `glm_fit <- glm(y ~ x, family = binomial(link = logit))`

Poisson regression

In Poisson regression we model counts

- i.e., integer values: 0, 1, 2, 3, ...

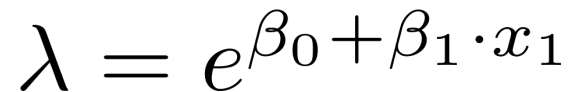
The log is the "link function"



The diagram shows the equation $\log(\lambda) = \beta_0 + \beta_1 \cdot x$. A blue arrow points from the text "The log is the 'link function'" to the $\log(\lambda)$ term. Another blue arrow points from the text "'Linear predictor'" to the right-hand side of the equation, $\beta_0 + \beta_1 \cdot x$.

$$\log(\lambda) = \beta_0 + \beta_1 \cdot x$$

Solving for λ



The diagram shows the equation $\lambda = e^{\beta_0 + \beta_1 \cdot x_1}$. A blue arrow points from the text "Solving for λ " to the λ term. Another blue arrow points from the text "Inverse link function (exponential function)" to the right-hand side of the equation, $e^{\beta_0 + \beta_1 \cdot x_1}$.

$$\lambda = e^{\beta_0 + \beta_1 \cdot x_1}$$

Inverse link function
(exponential function)

Family is Poisson distributions



The diagram shows the equation $Y_i \sim \text{Poisson}(\lambda)$. A blue arrow points from the text "Family is Poisson distributions" to the Y_i term.

$$Y_i \sim \text{Poisson}(\lambda)$$

R: `glm_fit <- glm(y ~ x, family = Poisson(link = log))`

Poisson distributions

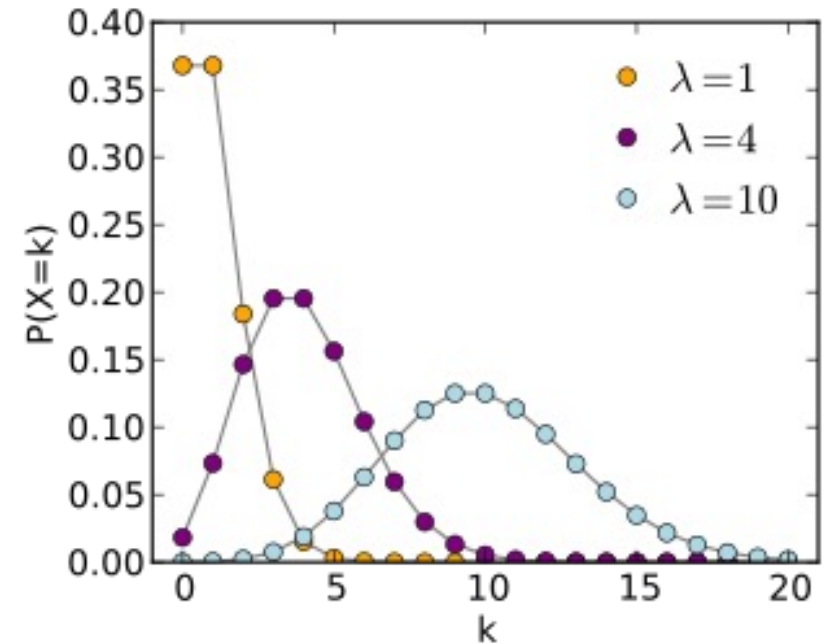
A Poisson distribution is a probability distribution over non-negative integers

- i.e., over values 0, 1, 2, 3, ...

Poisson distributions have a single parameter λ

$$X \sim \text{Pois}(\lambda)$$

$$P(X = k) = \frac{\lambda^k}{k!} e^{-\lambda}, \quad k = 0, 1, 2, \dots$$



- Density: `dpois()`
- Cumulative distribution: `ppois()`
- Random number: `rpois()`

Poisson processes

Poisson distributions models the number of outcomes that have occurred from a **Poisson process**

A **Poisson process** is a stochastic process where:

- Events (random outcome) occur at a fixed rate (λ)
- Every event is independent of the other events

Examples of Poisson processes?



Copyright © 2012 Vectis Auctions. All Rights Reserved.

Side note: Maximum likelihood estimate (MLE)

When building regression models, we need a way to estimate parameters

The "true" underlying model is:

$$y = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \dots + \beta_k \cdot x_k + \epsilon$$

We estimate coefficients using a data set to make predictions \hat{y}

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 \cdot x_1 + \hat{\beta}_2 \cdot x_2 + \dots + \hat{\beta}_k \cdot x_k$$

For GLMs, the maximum likelihood estimates (MLE) is used to estimate the regression coefficients:

- MLEs find the parameters that make the data as likely as possible
 - (For linear regression with normal errors, MLE gives the same coefficient estimates as least squares)

Example: Roy Kent saying f#ck

Ted Lasso was a Apple TV+ series that aired from July 2021 to March 2023

One of the main characters on the show was Roy Kent, who tended to say f#ck frequently

In different episodes of the show Roy was:

- A coach
- Dated Keeley Jones

Let's use Poisson regression to assess if Roy had a tendency to say f#ck more when he was **coaching** and/or when he was **dating** Keeley



Example from season 2

Let's try it in R...

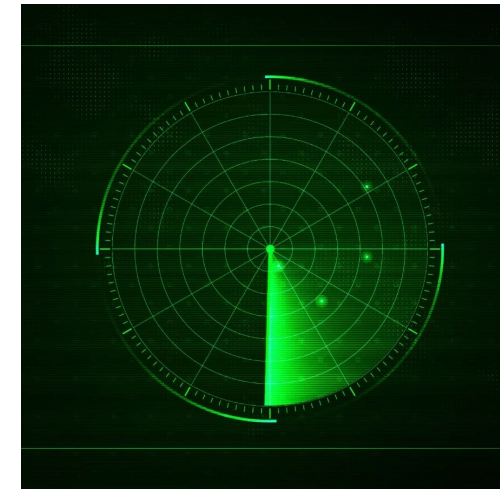
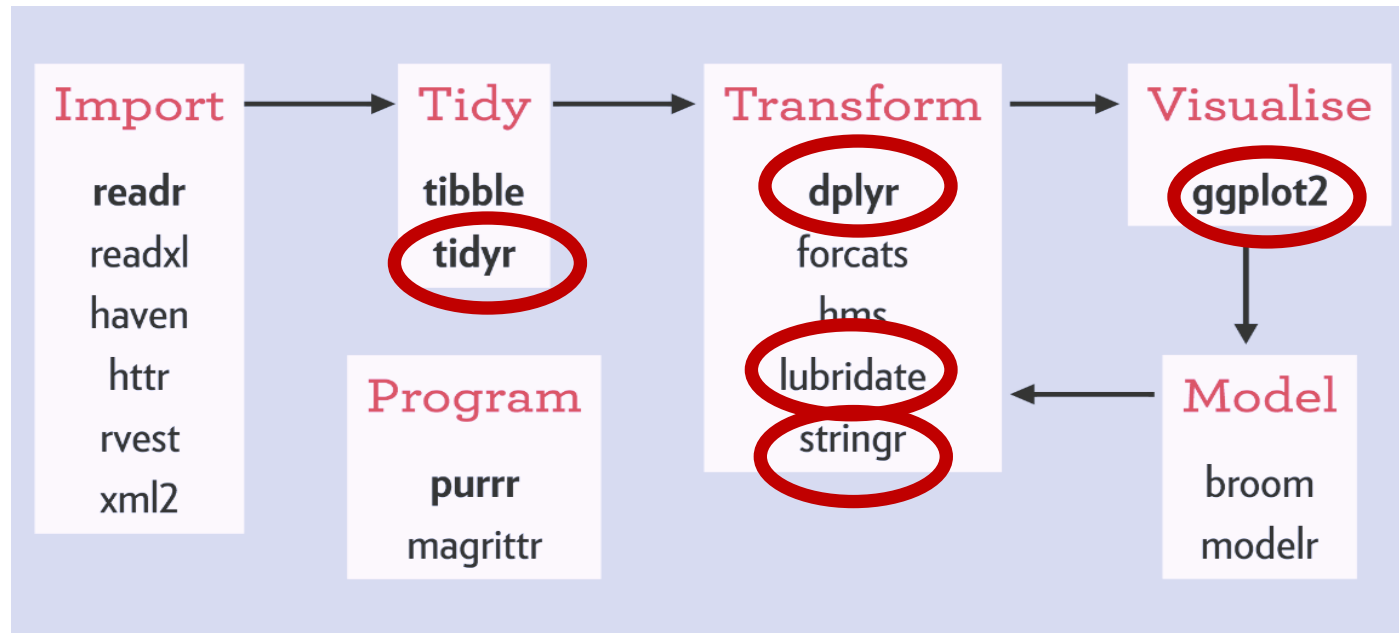


Tidyverse packages useful for your projects

Tidyverse packages useful for your projects

The packages share a common design philosophy

- Most written by Hadley Wickham



The [posit cheat sheets](#) can be very useful

lubridate



lubridate is a package that makes it easier to process dates

- `library(lubridate)` `# install.packages(lubridate)`

There are functions to parse strings and numbers into dates

```
some_date <- ymd(19790816)  
current_date <- mdy("11/16/23")
```

There are functions to extract information from dates:

```
year(current_date)
```

stringr



stringr is a package for manipulate character strings

- `library(stringr)`

There are many useful functions in the stringr package

- Perhaps we can discuss in more detail later in the semester...

Particularly useful is a function for replacing parts of a string:

- `str_replace_all("one fish, two fish, red fish, blue fish", "fish", "cat")`

You can use **regular expressions** to make the string match much more powerful

Let's try it in R...

tidyr for pivoting data

Wide vs. Long data

Plotting data using ggplot requires that data is in the right format

- i.e., requires data transformations

Often this involves converting data from a **wide format** to **long format**

Wide data

Person	Age	Height
Bob	32	72
Alice	24	65
Steve	64	70

Narrow data

Person	name	value
Bob	Age	32
Bob	Height	72
Alice	Age	24
Alice	Height	65
Steve	Age	64
Steve	Height	70

`library(tidyr)`

tidyr::pivot_longer()

pivot_longer(df, cols) converts data from **wide** to **long**

- Takes multiple columns and converts them into two columns: name and value
 - Column names become categorical variable levels of a new variable called **name**
 - The data in rows become entries in a variable called **value**

Wide data

Person	Age	Height
Bob	32	72
Alice	24	65
Steve	64	70



Long data

Person	name	value
Bob	Age	32
Bob	Height	72
Alice	Age	24
Alice	Height	65
Steve	Age	64
Steve	Height	70

tidyr::pivot_wider()

pivot_wider(df, names_from, values_from) converts data from long to wide

- Turns the levels of categorical data into columns in a data frame

Narrow data

person	name	value
Bob	Age	32
Bob	Height	72
Alice	Age	24
Alice	Height	65
Steve	Age	64
Steve	Height	70



Wide data

Person	Age	Height
Bob	32	72
Alice	24	65
Steve	64	70

Let's try it in R...