Generalized Linear Models

Fitting Models to Data, Not Data to Models
Model Fitting Series - With Applications in R

Jesse Ghashti

October 23, 2025

Centre for Scholarly Communication
The University of British Columbia | Okanagan Campus | Syilx Okanagan Nation Territory

Welcome, feel free to get ready while we wait to start...

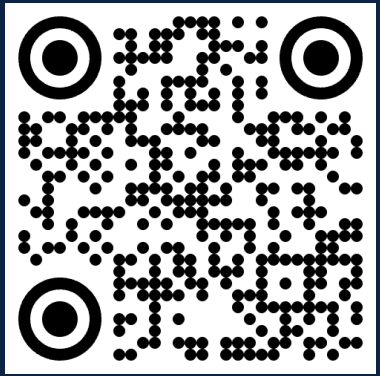
Install Required Packages

```
packages <- c("dplyr", "mgcv", "ggplot2",
              "gratia", "janitor", "tidyr")
toInstall <- packages[!(packages %in%
                          installed.packages()[,"Package"])]
if(length(toInstall)) install.packages(toInstall)

library('dplyr')    # for data wrangling
library('mgcv')     # for modeling
library('ggplot2')  # for fancy plots
library('janitor')  # for data cleaning
library('gratia')   # ggplot-based graphics
library('tidyr')
```

Workshop Series Overview

Session	Topic	Date/Time
1	Simple Linear Regression	Oct 7, 9:00 AM
2	Fitting Linear Models in R	Oct 8, 10:30 AM
3	Multiple Linear Regression in R	Oct 16, 4:00 PM
4	Interaction Terms & Hierarchical Linear Models	Oct 21, 11:00 AM
5	**Generalized Linear Models**	**Oct 23, 4:00 PM**
6	Generalized Additive Models (GAMs)	Oct 28, 11:00 AM
7	Interpreting & Predicting from GAMs	Oct 29, 10:30 AM
8	Hierarchical GAMs	Nov 4, 12:00 PM
9	Penalized Models	Nov 18, 11:00 AM
10	Survival Models	Nov 25, 11:00 AM
11	Nonparametric Models	Dec 2, 11:00 AM

New Here?

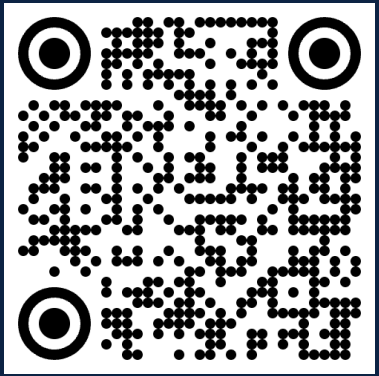


<———————-

New to R? Check out the Fundamentals of R series!

GitHub code and slides for today's workshop (and previous workshops)

———————————>



Alternatively, code/slides available at the bottom of `https://csc-ubc-okanagan.github.io/workshops/`

Last Time (Workshop 4) — Quick Recap

Key Concepts

- Learned about interaction terms: when effects depend on context
- Distinguished fixed effects (specific levels) vs random effects (population samples)
- Built hierarchical models with `s(chick, bs = 're')`
- Visualized interactions with heat maps and faceted plots
- "It depends" (on) can be part of your model.

Today: Generalized Linear Models (GLMs)

Today we will...

- Move beyond normality
- Understand the three components of GLMs
- Learn about link functions and why we need them
- Interpret coefficients on the link scale

Today we require...

```
library('dplyr')   # for data wrangling
library('tidyr')   # for expand_grid
library('mgcv')    # for modeling
library('ggplot2') # for fancy plots
library('gratia')  # for ggplot-based model graphics
library('tidyr')   # for data wrangling
theme_set(theme_bw(base_size = 15))
```

Why Do We Need GLMs?

Limitations of Linear Models (among other things)

Linear models assume:

- Response is continuous ($-\infty$ to $+\infty$)
- Errors are normally distributed
- Variance is constant (homoscedasticity)

But real data often consists of more:

- Count data (0, 1, 2, ...) — can't be negative
- Binary outcomes (yes/no, success/failure)
- Proportions
- Positive continuous (weight, income) — can't be negative

GLMs generalize linear models to handle these situations

The Three Components of a GLM

GLM = Family + Link + Linear Predictor

1. Random Component (Family):

- Distribution of Y: Normal, Poisson, Binomial, Gamma, etc.

2. Systematic Component (Linear Predictor):

- $\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots$

3. Link Function:

- Connects mean of Y to linear predictor: $g(\mu) = \eta$
- Or equivalently: $\mu = g^{-1}(\eta)$

Linear models are GLMs with `Normal family + Identity link`

Common GLM Families and Link Functions

Response Type	Family	Common Link	Link Function
Continuous	Gaussian	Identity	$g(\mu) = \mu$
Count data	Poisson	Log	$g(\mu) = \log(\mu)$
Binary	Binomial	Logit	$g(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$
Positive continuous	Gamma	Log	$g(\mu) = \log(\mu)$
Proportions	Beta	Logit	$g(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$

The link function maps the constrained mean to the unconstrained linear predictor scale

GLM Derivation Example: Poisson Regression

Modelling count data Y_i using a log link

1. Random Component:

$$Y_i \sim \text{Poisson}(\mu_i), \quad \text{with } P(Y_i = y_i) = \frac{e^{-\mu_i}\mu_i^{y_i}}{y_i!}$$

2. Systematic Component: $\eta_i = \beta_0 + \beta_1 x_i$

3. Link Function: $g(\mu_i) = \log(\mu_i) = \eta_i \quad \Rightarrow \quad \mu_i = e^{\eta_i}$

4. Substitution: $\mu_i = e^{\beta_0 + \beta_1 x_i}$

So a one-unit increase in x multiplies the expected count by e^{β_1}.

The link function transforms the mean μ_i from the constrained (positive) scale to an unconstrained linear predictor η_i for estimation.

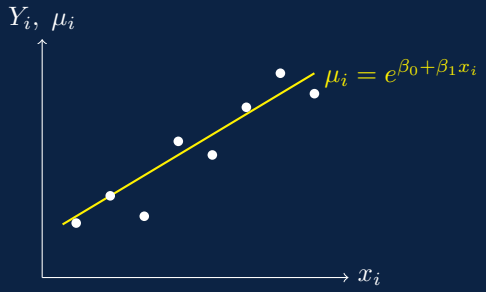
Observed vs Expected: Y_i and μ_i

Building intuition...

- Y_i: observed outcome (random)
- $\mu_i = \mathbb{E}[Y_i]$: model's expected mean
- GLMs model μ_i, not Y_i directly

$$g(\mu_i) = \eta_i = \mathbf{x}_i^\top \boldsymbol{\beta}$$

$$\Rightarrow \mu_i = g^{-1}(\eta_i)$$



Compare Y_is around its expected value μ_i, and the GLM learns $\boldsymbol{\beta}$ to make μ_i as close as possible to the true mean of Y_i.

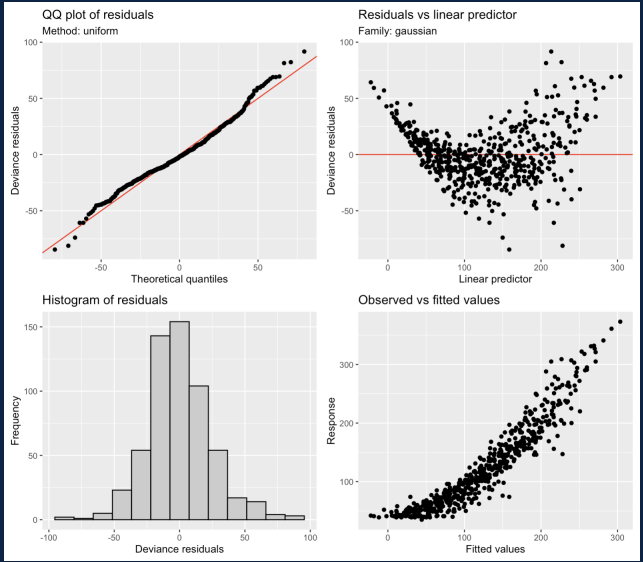
ChickWeight: Linear Model Review

```r
# Best model from last week
m_cw_lm <- gam(
  formula = weight ~
    Time +
    Diet +
    Time:Diet +
    s(Chick, bs = 're'),
  family = gaussian(), # Linear model
  data = ChickWeight,
  method = 'ML')
# appraise(m_cw_lm)
summary(m_cw_lm)
```

Notice the linear model can predict negative weights

```
Family: gaussian
Link function: identity

Formula:
weight ~ Time + Diet + Time:Diet + s(Chick, bs = "re")

Parametric coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  31.5081     5.9358   5.308 1.63e-07 ***
Time          6.7130     0.2584  25.982  < 2e-16 ***
Diet2        -2.8745    10.2342  -0.281    0.779
Diet3       -13.2577    10.2342  -1.295    0.196
Diet4        -0.3983    10.2430  -0.039    0.969
Time:Diet2    1.8961     0.4285   4.425 1.17e-05 ***
Time:Diet3    4.7099     0.4285  10.992  < 2e-16 ***
Time:Diet4    2.9495     0.4340   6.795 2.92e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
           edf Ref.df     F p-value
s(Chick) 41.15     46 9.879  <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.873   Deviance explained = 88.3%
-ML =   2744  Scale est. = 643.72    n = 578
```

Recall our `appraise()`



Why Gamma Distribution for Weight?

Gamma Distribution Properties

- Continuous, positive values only
- Right-skewed
- Variance increases with mean
- Common for:
  - Weights/masses
  - Waiting times
  - Income
  - Rainfall

Log Link Benefits

- Ensures predictions > o
- Multiplicative effects
- Natural for growth processes
- Model: $\mu = \exp(\eta)$
- Interpretation: % changes

From a biological standpoint, growth is often multiplicative — chicks may grow as a percentage of current weight

Fitting a Gamma GLM

```r
# Gamma GLM with log link
m_cw_glm <- gam(
  formula = weight ~
    Time +            # time is linear on link scale
    Diet +            # each diet has different intercept
    Time:Diet +       # interaction on log scale
    s(Chick, bs = 're'),
  family = Gamma(link = 'log'),  # gamma link
  data = ChickWeight,
  method = 'ML')
summary(m_cw_glm)
```

```
Family: Gamma
Link function: log

Formula:
weight ~ Time + Diet + Time:Diet + s(Chick, bs = "re")

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.777906   0.038080  99.210  < 2e-16 ***
Time        0.067876   0.001558  43.578  < 2e-16 ***
Diet2       0.051339   0.065687   0.782  0.43481
Diet3       0.019151   0.065687   0.292  0.77074
Diet4       0.101373   0.065737   1.542  0.12365
Time:Diet2  0.008199   0.002582   3.175  0.00158 **
Time:Diet3  0.022041   0.002582   8.536  < 2e-16 ***
Time:Diet4  0.014585   0.002616   5.576 3.93e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
           edf Ref.df     F p-value
s(Chick) 41.83     46 12.08  <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.907   Deviance explained = 93.2%
-ML = 2493.3  Scale est. = 0.023364  n = 578
```

Coefficients now represent multiplicative effects on the response scale

Interpreting GLM Coefficients

Link Scale vs Response Scale

On the link scale (what R reports):

$$\log(\mu) = \beta_0 + \beta_1 \text{Time} + \beta_2 \text{Diet2} + \cdots$$

On the response scale (actual weights):

$$\mu = \exp(\beta_0 + \beta_1 \text{Time} + \beta_2 \text{Diet2} + \cdots)$$

$$\mu = \exp(\beta_0) \times \exp(\beta_1 \text{Time}) \times \exp(\beta_2 \text{Diet2}) \times \cdots$$

Model Equations with Fitted Coefficients

Gaussian (Identity Link): ***note that I rounded to two decimal places.***

$Y_i \sim \text{Normal}(\mu_i, \sigma^2)$ with link $g(\mu_i) = \mu_i$

$$\mu_i = 31.51 + 6.71\,\text{Time}_i - 2.87\,\text{Diet2}_i - 13.26\,\text{Diet3}_i - 0.40\,\text{Diet4}_i + 1.90\,(\text{Time}_i \times \text{Diet2}_i) + 4.71\,(\text{Time}_i \times \text{Diet3}_i) + 2.95\,(\text{Time}_i \times \text{Diet4}_i) + s(\text{Chick}_i)$$

Gamma (Log Link): *note that I rounded to four decimal places.*

$Y_i \sim \text{Gamma}(\mu_i, \phi)$ with link $g(\mu_i) = \log(\mu_i)$

$$\log(\mu_i) = 3.78 + 0.0679\,\text{Time}_i + 0.0513\,\text{Diet2}_i + 0.0192\,\text{Diet3}_i + 0.1014\,\text{Diet4}_i + 0.0082\,(\text{Time}_i \times \text{Diet2}_i) + 0.0220\,(\text{Time}_i \times \text{Diet3}_i) + 0.0146\,(\text{Time}_i \times \text{Diet4}_i) + s(\text{Chick}_i)$$

Model Equations with Fitted Coefficients Interpretation

- Gaussian model predicts weight directly in original units (grams).
- Gamma model predicts $\log(\mu_i)$, so effects are multiplicative on the mean. For example,

e^{β_1} gives proportional change per unit increase in Time.

Sidenote 1: Where's `Time1`?!

Gaussian (Identity Link)

$Y_i \sim \text{Normal}(\mu_i, \sigma^2)$ with link $g(\mu_i) = \mu_i$

$$\mu_i = 31.51 + 6.71\,\text{Time}_i - 2.87\,\text{Diet2}_i - 13.26\,\text{Diet3}_i - 0.40\,\text{Diet4}_i + 1.90\,(\text{Time}_i \times \text{Diet2}_i) + 4.71\,(\text{Time}_i \times \text{Diet3}_i) + 2.95\,(\text{Time}_i \times \text{Diet4}_i) + s(\text{Chick}_i)$$

- The red above are indicators. This means the value is `1` if the condition is true, and 0 if not.
- For example, if chick i has `diet = 2`, then when we calculate μ_i or $\log(\mu_i)$, $\texttt{Diet2}_i = 1, \texttt{Diet3}_i = 0$ and $\texttt{Diet4}_i = 0$.
- If chick i has `diet = 1`, then the model collapses down to

$$\mu_i = 31.51 + 6.71\texttt{Time}_i \quad \text{and} \quad \log(\mu_i) = 3.78 + 0.0679\texttt{Time}_i.$$

Sidenote 2: We didn't `appraise()` the GLM!

For good reason... see the discussion here:

https://stats.stackexchange.com/questions/121490/interpretation-of-plot-glm-model

TLDR; unless you are very experienced with such models, the diagnostics plots are incredibly difficult to understand.

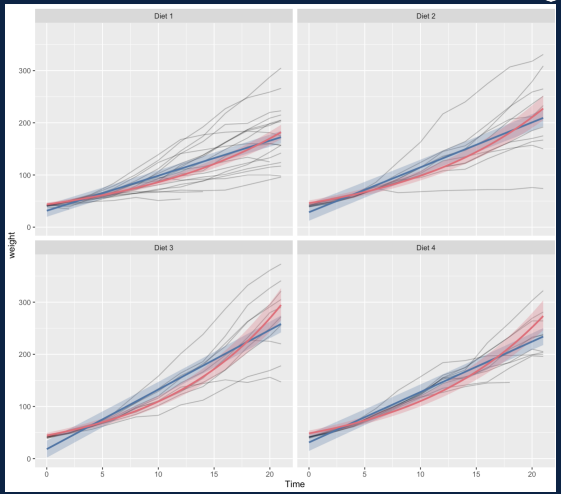
Making Predictions from GLMs

```r
newd <- expand_grid(
  Time = seq(0, 21, length.out = 400),
  Diet = unique(ChickWeight$Diet), # since we are using fixed effects
  Chick = 'new chick') # since we are using random effects

preds <- bind_cols(newd,predict(object = m_cw_lm, newdata = newd, type = 'response', se.fit = TRUE) %>%
                     as.data.frame() %>%
                     rename(lm_fit = fit, lm_se = se.fit) %>%
                     mutate(lm_est = lm_fit, lm_95_lwr = lm_fit - 1.96 * lm_se,
                            lm_95_upr = lm_fit + 1.96 * lm_se),
                   predict(object = m_cw_glm, newdata = newd, type = 'link', se.fit = TRUE) %>%
                     as.data.frame() %>%
                     rename(glm_fit = fit, glm_se = se.fit) %>%
                     mutate(glm_est = exp(glm_fit),
                            glm_95_lwr = exp(glm_fit - 1.96 * glm_se),
                            glm_95_upr = exp(glm_fit + 1.96 * glm_se)))
```

Calculate CIs on the link scale, then transform (bottom three lines of the code above)

Comparing LM vs Gamma GLM

```r
# Plot both model predictions
ggplot(preds) +
  facet_wrap(~ paste('Diet', Diet)) +
  geom_line(aes(Time, weight,
                group = Chick),
            ChickWeight, alpha = 0.3) +
  geom_ribbon(aes(Time,
                  ymin = lm_95_lwr,
                  ymax = lm_95_upr),
              alpha = 0.3, fill = '#4477AA') +
  geom_ribbon(aes(Time,
                  ymin = glm_95_lwr,
                  ymax = glm_95_upr),
              alpha = 0.3, fill = '#EE6677') +
  geom_line(aes(Time, lm_est),
            color = '#4477AA', lwd = 1) +
  geom_line(aes(Time, glm_est),
            color = '#EE6677', lwd = 1)
```



Differences between LM and GLM

Linear Model

- Constant variance
- Can predict negative values
- Additive effects
- Symmetric CIs
- Easier interpretation

Gamma GLM

- Variance increases with mean
- Always positive predictions
- Multiplicative effects
- Asymmetric CIs
- More realistic for growth

Notice in the plot GLM confidence intervals get wider as weight increases. Is this intuitive given the problem?

GLM Practice

Exercise: Explore different families

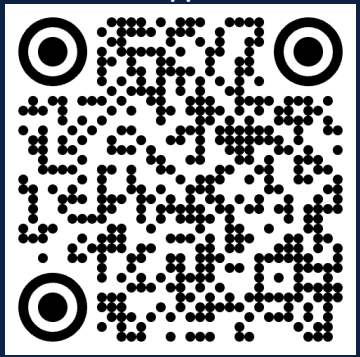
```
# Exercise 1: warpbreaks data (count of breaks)
?warpbreaks
# hint: try family = poisson(link = 'log')

# Exercise 2: mtcars data (mpg is positive continuous)
?mtcars
# hint: try family = Gamma(link = 'log') vs gaussian()

# Here we should try the following:
# 1. Fit both a LM and appropriate GLM
# (try variable selection techniques and interactions)
# 2. Write the fitted model equation of your chosen model
# 3. Plot predictions from both models
# 4. Use appropriate diagnostics and determine which model is a better. Discuss why.
```

Key Takeaways

- GLMs extend linear models: Family + Link + Linear Predictor
- Choose family based on response type (count, binary, positive continuous)
- Link functions map constrained means to unconstrained linear scale
- Interpret coefficients carefully — often on transformed scale
- Always calculate CIs on link scale, then transform
- GLMs can capture variance-mean relationships naturally
- "All models are wrong, but some are useful" (and GLMs are often quite useful)

What's Next?

Additional Questions?
Book an Appointment!



Additional Resources: *An Introduction to Generalized Linear Models* (Dobson & Barnett)

Next Workshop: Generalized Additive Models (GAMs)

October 28, 11:00 AM
– Smooth functions
– Nonlinear relationships
– GLMs + flexibility