Generalized Linear Models: A Brief Intro

Dr. Michael Fix
mfix@gsu.edu

Georgia State University

10 April 2025

Note: The slides are distributed for use by students in POLS 8810. Please do not reproduce or redistribute these slides to others without express permission from Dr. Fix.

When the Linear Model Fails

- Much of the data we are interested in, causes major issues with the linear model due to, among other things:
  - Non-linearities between **X** and **Y**
  - DVs that are noncontinuous or bounded
  - Issue with residuals
- So what do we do?

When the Linear Model Fails

- Generalized linear models are just that. They allow us to build from the classical linear model
- The most basic GLMs allow us to model non-linear relationships, noncontinuous DVs, and $E(\mathbf{u}) \neq 0$
- Some GLMs allow for correlation between $\mathbf{X}$ and $\mathbf{u}$

GLMs are NOT a Panacea

- While some GLMs can accomodate correlation between $\mathbf{X}$ and $\mathbf{u}$, cases (i.e. columns of $\mathbf{X}$) must be uncorrelated, thus GLMs do not handle time-series data or spatially correlated data any better than the classical linear model
- Requires a single error terms, so models with a more complicated error structure can be problematic (at least with basic GLMs)
- GLMS are fully parametric. This means the researcher MUST correctly define the form of the likelihood function
- We should still avoid pitfalls such as stargazing, data mining, etc

Some Definitions

GLMs require us think think in probabilistic terms. Moving forward requires some definitions:

Probability Density Function (PDF): $f(y)$ or a probabilistic function about the distribution of a random variable, Y, over a defined range

Probability Mass Function (PMF): $P(Y = y)$. Discrete case version of the PDF. The probability that a random variable, Y, takes on some realization y.

Thinking Careful about Distributional Forms

- Given the importance of selecting the appropriate distribution, the question becomes how to select from among the dozens of known statistical distributions
- Information about our dependent variable helps us narrow down our choices to a given family of distributions:
  - Is the dependent variable continuous or discrete?
  - Is the depend value truncated a a given value (e.g. 0)
- Our choice of distribution reflects (in part) our level of uncertainty about the functional form of the relationship between $\mathbf{X}$ and the $\mathbf{y}$.
- This is an important decision that requires careful thought, examination of various plots and other preliminary data analysis techniques, and knowledge of the nature of the dependent variable.

Linear Model(s)

$$Y_i = \mathbf{X}_i \boldsymbol{\beta} + u_i \tag{1}$$

$$\mathrm{E}(Y_i) \equiv \boldsymbol{\mu}_i = \mathbf{X}_i \boldsymbol{\beta} \tag{2}$$

The “Generalized” Part

$$g(\boldsymbol{\mu}_i) = \mathbf{X}_i\boldsymbol{\beta}. \tag{3}$$

$$\begin{aligned} \boldsymbol{\eta}_i =& \mathbf{X}_i\boldsymbol{\beta} \qquad (4) \\ =& g(\boldsymbol{\mu}_i) \qquad (5) \end{aligned}$$

$$\begin{aligned} \boldsymbol{\mu}_i =& g^{-1}(\boldsymbol{\eta}_i) \qquad (6) \\ =& g^{-1}(\mathbf{X}_i\boldsymbol{\beta}) \qquad (7) \end{aligned}$$

GLMs

Random component:

$$\mathrm{E}(Y_i) = \boldsymbol{\mu}_i. \tag{8}$$

Systematic component:

$$\boldsymbol{\eta}_i = \mathbf{X}_i \boldsymbol{\beta} \tag{9}$$

"Link function":

$$g(\boldsymbol{\mu}_i) = \boldsymbol{\eta}_i \tag{10}$$

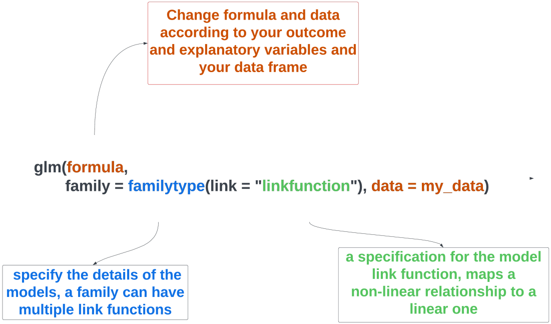
or

$$g^{-1}(\boldsymbol{\eta}_i) = \boldsymbol{\mu}_i. \tag{11}$$

GLM in R

- *glm()* function (in base R, so you do not need any packages)
- But, there are also packages like *glm2* or *glmnet* which might be helpful for advanced stuff (like penalized ML)

Structure of GLM code



Change formula and data according to your outcome and explanatory variables and your data frame

```
glm(formula,
    family = familytype(link = "linkfunction"), data = my_data)
```

specify the details of the models, a family can have multiple link functions

a specification for the model link function, maps a non-linear relationship to a linear one

Structure of GLM code

- For instance, the following code runs OLS:

$$
\begin{aligned}
&glm(Y \sim X_1 + X_2,\\
&\qquad family = \textbf{gaussian}(link = \textbf{``identity''}),\\
&\qquad data = my\_data)
\end{aligned}
\tag{12}
$$

- By changing **family type** and **link function**, you will get different estimators

GLM Family Quick Guide

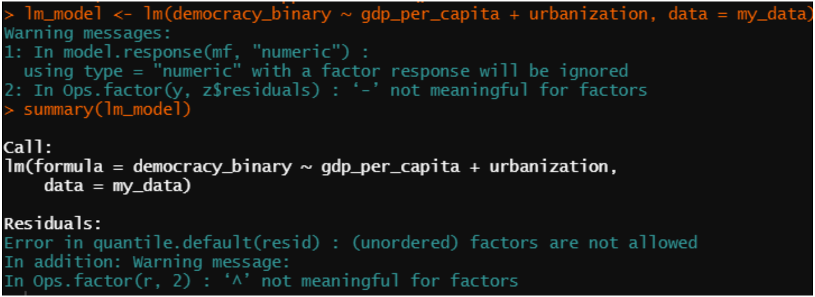
Family	**Default Link Function**
binomial	(link = “logit”)
gaussian	(link = “identity”)
Gamma	(link = “inverse”)
inverse.gaussian	(link = “$1/mu^2$”)
poisson	(link = “log”)
quasi	(link = “identity”, variance = “constant”)
quasibinomial	(link = “logit”)
quasipoisson	(link = “log”)

Toy Model

- V-Dem data
- Democracy (binary) explained by GDP per capita and urbanization

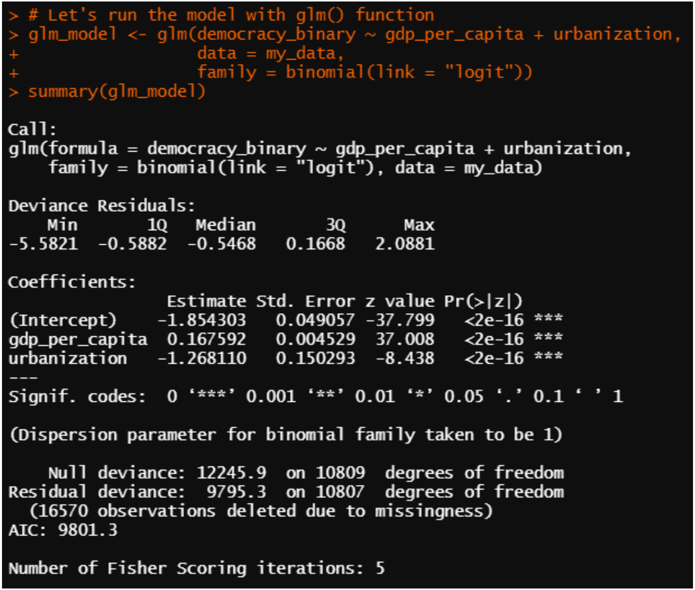
Let’s run using lm() and see what message we get

Oh no, R is confused!



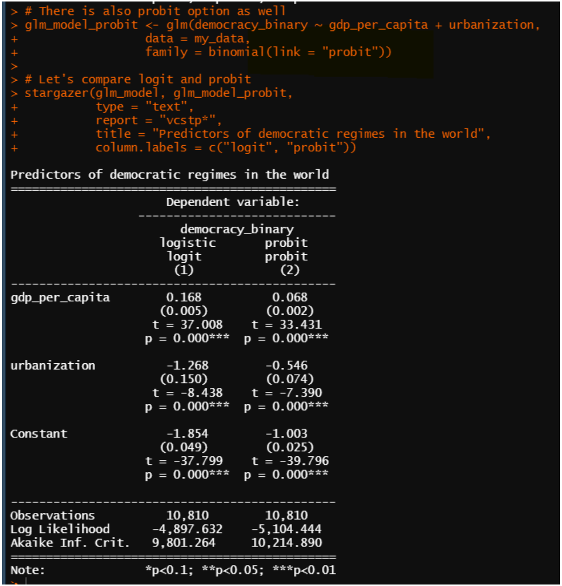
```
> lm_model <- lm(democracy_binary ~ gdp_per_capita + urbanization, data = my_data)
Warning messages:
1: In model.response(mf, "numeric") :
  using type = "numeric" with a factor response will be ignored
2: In Ops.factor(y, z$residuals) : ‘-’ not meaningful for factors
> summary(lm_model)

Call:
lm(formula = democracy_binary ~ gdp_per_capita + urbanization,
    data = my_data)

Residuals:
Error in quantile.default(resid) : (unordered) factors are not allowed
In addition: Warning message:
In Ops.factor(r, 2) : ‘^’ not meaningful for factors
```

Using glm(), we can run things smoothly



```
> # Let's run the model with glm() function
> glm_model <- glm(democracy_binary ~ gdp_per_capita + urbanization,
+                  data = my_data,
+                  family = binomial(link = "logit"))
> summary(glm_model)

Call:
glm(formula = democracy_binary ~ gdp_per_capita + urbanization,
    family = binomial(link = "logit"), data = my_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.5821  -0.5882  -0.5468   0.1668   2.0881

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    -1.854303   0.049057 -37.799   <2e-16 ***
gdp_per_capita  0.167592   0.004529  37.008   <2e-16 ***
urbanization   -1.268110   0.150293  -8.438   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 12245.9  on 10809  degrees of freedom
Residual deviance:  9795.3  on 10807  degrees of freedom
  (16570 observations deleted due to missingness)
AIC: 9801.3

Number of Fisher Scoring iterations: 5
```

probit link and its comparison with *logit*



```
> # There is also probit option as well
> glm_model_probit <- glm(democracy_binary ~ gdp_per_capita + urbanization,
+                  data = my_data,
+                  family = binomial(link = "probit"))
>
> # Let's compare logit and probit
> stargazer(glm_model, glm_model_probit,
+           type = "text",
+           report = "vcstp*",
+           title = "Predictors of democratic regimes in the world",
+           column.labels = c("logit", "probit"))

Predictors of democratic regimes in the world
==============================================
                      Dependent variable:
                  ----------------------------
                        democracy_binary
                     logistic       probit
                      logit         probit
                       (1)            (2)
----------------------------------------------
gdp_per_capita        0.168          0.068
                     (0.005)        (0.002)
                    t = 37.008    t = 33.431
                   p = 0.000***  p = 0.000***

urbanization          -1.268        -0.546
                     (0.150)        (0.074)
                    t = -8.438    t = -7.390
                   p = 0.000***  p = 0.000***

Constant              -1.854        -1.003
                     (0.049)        (0.025)
                   t = -37.799    t = -39.796
                   p = 0.000***  p = 0.000***

----------------------------------------------
Observations            10,810      10,810
Log Likelihood      -4,897.632    -5,104.444
Akaike Inf. Crit.   9,801.264     10,214.890
==============================================
Note:              *p<0.1; **p<0.05; ***p<0.01
```