
GENERALIZED LINEAR MODELS

R LAB

PRODUCED BY: RISHI DEY CHOWDHURY

Table of Contents

1	Brief Overview of Generalized Linear Models	2
1.1	Theory	2
2	Fit Generalized Linear Models in R	3
2.1	Prepare Data	3
2.2	Fit GLM	4
2.3	How GLM is fit?	5
2.4	More on Families and Links	5

About This File

This file was created for the benefit of all teachers and students wanting to learn about using R for Generalized Linear Models

The entirety of the contents within this file, and folder, are free for public use.

Brief Overview of Generalized Linear Models

1.1 Theory

Suppose we want to fit a linear model, just the usual linear model, there are two pieces of information we need to specify i.e.

$$X, \vec{y}$$

where X is the design matrix and \vec{y} is the response we want to model given X .

But when we work with Generalized Linear Models, we need to specify one more thing i.e. the distribution of \vec{y} (in case of simple linear models, we assume that to be Normally distributed).

$$\vec{y} \sim ?$$

One more piece that we need to decide is how is X connected to \vec{y} . In case of simple linear model it was simply,

$$E(\vec{y}) = X\vec{\beta}$$

But, in generalized linear model, we have flexibility of using a **link function** h , thus our model becomes,

$$E(\vec{y}) = h(X\vec{\beta})$$

So, all it one, to fit GLM in R, we will require these two crucial pieces of information i.e. the **family** of distribution \vec{y} follows and the **link function**.

Fit Generalized Linear Models in R

2.1 Prepare Data

Before moving ahead to fit the GLM in R. We will create a toy dataset about mice deaths and dosage of drugs injected (as discussed previously).

```
# The dosage injected
dose = 7:13
n = length(dose)
# To store the number of proportion of mice which died at each dosage level
propdeath = numeric(n)
mat = c()
for(i in 1:n) {
  # Assuming lethal dosage amount to be 10, we simulate 100 mices
  lethal = rnorm(100,mean=10)
  # Find the deaths
  dead = dose[i] >= lethal
  mat = rbind(mat,cbind(dose[i],dead))
  propdeath[i] = mean(dead)
}
# Setting column names
colnames(mat) <- c('dose', 'death')
# Converting to data.frame
dat <- data.frame(mat)
head(dat)
```

```

1      7      0
2      7      0
3      7      0
4      7      0
5      7      0
6      7      0
tail(dat)
695    13      1
696    13      1
697    13      1
698    13      1
699    13      1
700    13      1

```

2.2 Fit GLM

After we have the data ready. We can go ahead with fitting GLM. Fortunately, we don't require any additional package. The required functions for GLM comes with base R. We will use the following commands to fit the GLM.

```

# Fitting GLM for death as response variable and dose as explanatory
# Using binomial family with probit link
fit <- glm(death ~ dose, dat, family = binomial(link = 'probit'))

```

If you are worried about converting deaths into factors i.e. 0s means no death and 1s means death, there is no need for that as when we use glm with binomial family, it treats the deaths as a factor.

You will notice, that we are not free to choose any link function with any family. We have to decide on the family first and then see if the desired link is present in that family. This is because, binomial family does not accept all kinds of links, but only few allowed ones.

Now, we visualize our fit as follows,

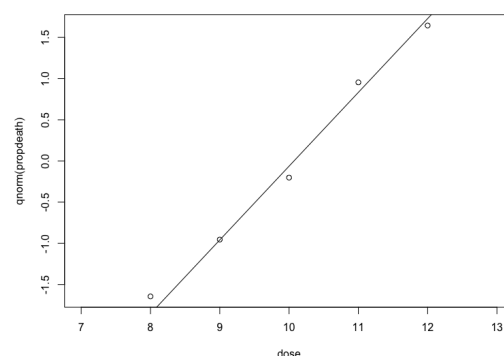


Figure 1: Visualize GLM Fit

```
# Code for the above plot
# We used probit link so we plot propdeath with qnorm vs dose
plot(dose, qnorm(propdeath))
# Display the fit line
abline(fit$coefficients)
```

2.3 How GLM is fit?

It basically uses Maximum Likelihood Estimation along with sophisticated optimization technique instead of trying to fit a line in case of a simple linear regression.

2.4 More on Families and Links

If you are interested in knowing more about the available families and their link functions. You can use the R's documentation to get to know about them as follows,

```
?family
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")
```

The above shows some of the families along with their default link function.