Generalized Linear Models in R

Markus Gesmann

21 November 2017

Agenda

Generalized linear models (GLMs) are a flexible generalization of linear models, with applications in many disciplines.

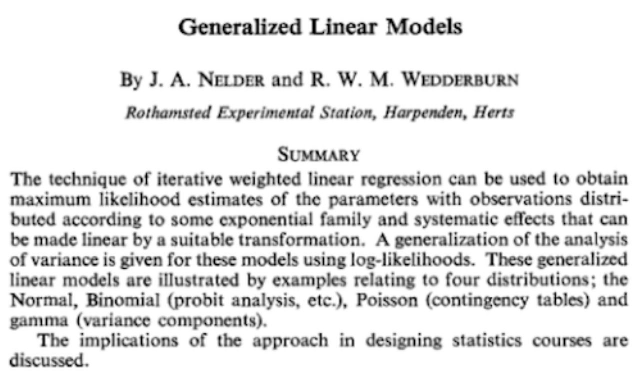
This talk will give an introduction to GLMs from a distribution-centric point of view.

Using a small toy data set we will discuss how different assumptions about the data generating process lead to different assumptions about the distribution of the data (response variable) and therefore different GLMs.

The problem GLMs solved

- Linear regression of data generated from distributions other than the Gaussian
- First published in 1972 by John Nelder and Robert Wedderburn, as a unifying approach for many different distributions

Original Abstract



Generalized Linear Models

By J. A. Nelder and R. W. M. Wedderburn

Rothamsted Experimental Station, Harpenden, Herts

Summary

The technique of iterative weighted linear regression can be used to obtain maximum likelihood estimates of the parameters with observations distributed according to some exponential family and systematic effects that can be made linear by a suitable transformation. A generalization of the analysis of variance is given for these models using log-likelihoods. These generalized linear models are illustrated by examples relating to four distributions; the Normal, Binomial (probit analysis, etc.), Poisson (contingency tables) and gamma (variance components).

The implications of the approach in designing statistics courses are discussed.

Figure 1: Nelder, John; Wedderburn, Robert (1972). “Generalized Linear Models”. Journal of the Royal Statistical Society. Series A (General). Blackwell Publishing. 135 (3): 370–384.

The original authors of GLM

Figure 2: John Nelder (1924 - 2010) and Robert Wedderburn (1947 - 1975). Photos courtesy of Wikipedia and Cambridge Statistical Laboratory

Example problem: Selling ice cream

Figure 3: How many units of ice cream should I stock?

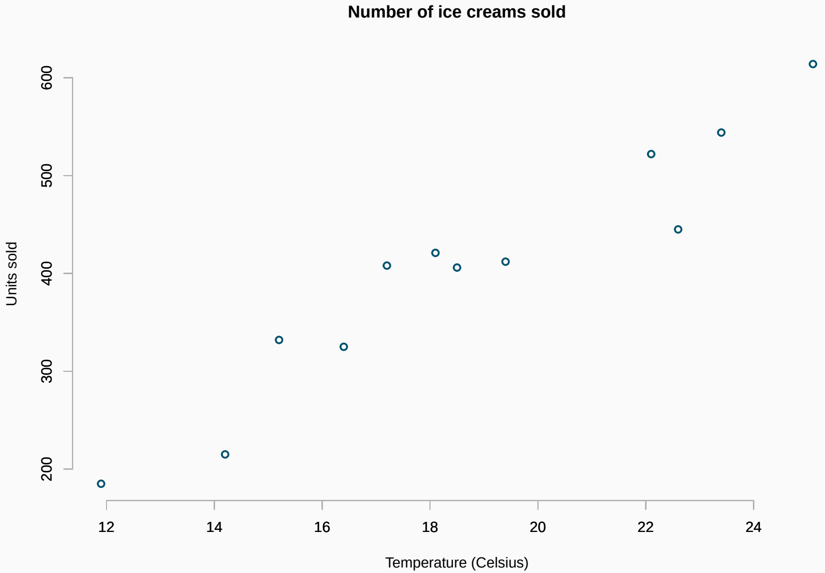
The challenge

- Create a model that predicts the number of ice creams sold for different temperatures
- Ensure model behaves well when the temperature drops to 0ºC and for a very hot summer's day at 35ºC.

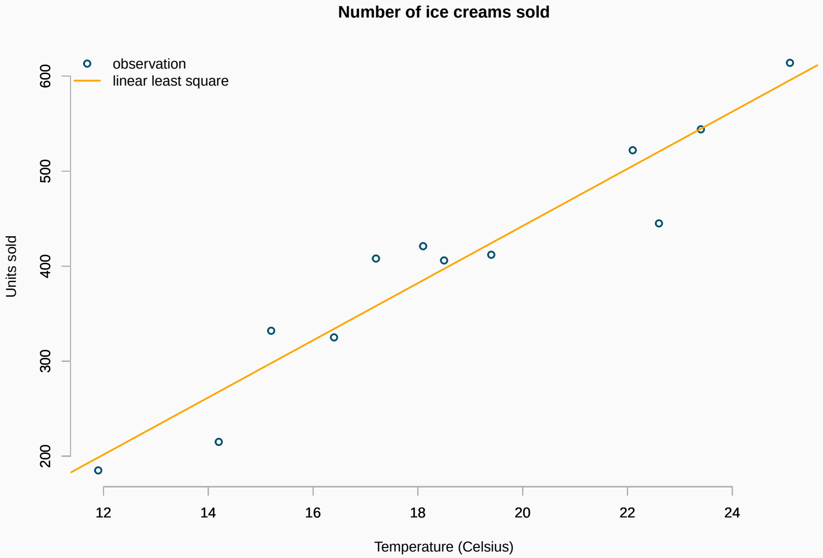
Available data

```r
icecream <- data.frame(
  temp=c(11.9, 14.2, 15.2, 16.4, 17.2, 18.1,
         18.5, 19.4, 22.1, 22.6, 23.4, 25.1),
  units=c(185L, 215L, 332L, 325L, 408L, 421L,
          406L, 412L, 522L, 445L, 544L, 614L)
  )
```

Plot of ice cream data



Use a ruler and pencil



Model 1: Gaussian GLM

Linear regression

Let's start with a probability distribution centric description of the data.

I believe the observation y_i was drawn from a Normal distribution with a mean μ_i, depending on the temperature x_i and a constant variance σ^2 across all temperatures.

$$y_i \sim \mathcal{N}(\mu_i, \sigma^2), \tag{1}$$

$$\mathbb{E}[y_i] = \mu_i = \alpha + \beta x_i \text{ for all } i \tag{2}$$

The first equation describes the distribution of the response and the second equation how the distribution parameter μ_i is linked in a linear way.

Model 1: Gaussian GLM

```r
library(arm) # for 'display' function only
lin.mod <- glm(units ~ temp, data=icecream,
               family=gaussian(link="identity"))
display(lin.mod)
```

```
glm(formula = units ~ temp, family = gaussian(link = "identity"),
    data = icecream)
            coef.est coef.se
(Intercept) -159.47    54.64
temp          30.09     2.87
---
  n = 12, k = 2
  residual deviance = 14536.3, null deviance = 174754.9 (difference = 160218.6)
  overdispersion parameter = 1453.6
  residual sd is sqrt(overdispersion) = 38.13
```

Model 1: Gaussian GLM Plot

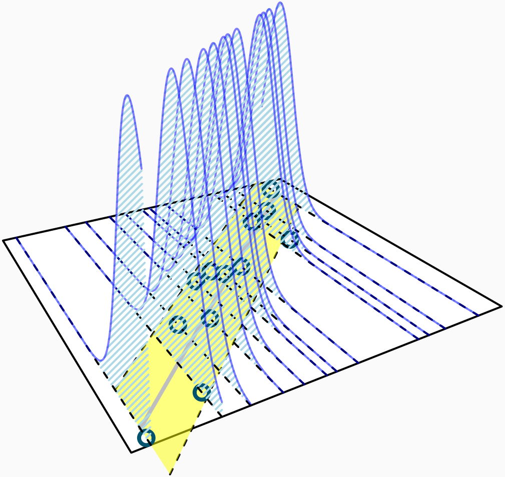


Figure 4:

$y_i \sim \mathcal{N}(\mu_i, \sigma^2)$ with $\mu_i = -159.5 + 30.1\, x_i$ and $\sigma = 38.1$

Model 1: Critique

Although the linear model looks fine in the range of temperatures observed, it doesn't make much sense at 0ºC. The intercept is at -159, which would mean that customers return on average 159 units of ice cream on a freezing day. Well, I don't think so.

Model 2: Log-transformed Gaussian GLM

Model 2: Log-transformed Gaussian GLM

Ok, perhaps I can transform the data first. Ideally I would like ensure that the transformed data has only positive values. So, let's model the ice cream sales on a logarithmic scale.

$$\log(y_i) \sim \mathcal{N}(\mu_i, \sigma^2) \tag{3}$$

$$\mathbb{E}[\log(y_i)] = \mu_i = \alpha + \beta x_i \tag{4}$$

This model implies that I believe the sales follow a log-normal distribution: $y_i \sim \log \mathcal{N}(\mu_i, \sigma^2)$.

Model 2: Lognormal distribution

The log-normal distribution is skewed to the right, which means that I regard higher sales figures more likely than lower sales figures.

Although the model is still linear on a log-scale, I have to remember to transform the predictions back to the original scale because $\mathbb{E}[\log(y_i)] \neq \log(\mathbb{E}[y_i])$.

$$\begin{aligned} y_i &\sim \log \mathcal{N}(\mu_i, \sigma^2) \qquad (5) \\ \mathbb{E}[y_i] &= \exp(\mu_i + \sigma^2/2) = \exp(\alpha + \beta x_i + \sigma^2/2) \qquad (6) \end{aligned}$$

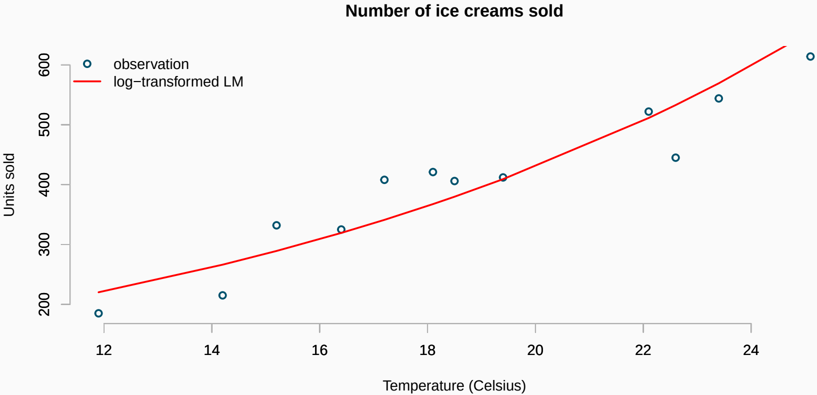
Model 2: Log-transformed Gaussian GLM

```
log.lin.mod <- glm(log(units) ~ temp, data=icecream,
                 family=gaussian(link="identity"))
display(log.lin.mod)
```

```
glm(formula = log(units) ~ temp, family = gaussian(link = "identity"),
    data = icecream)
            coef.est coef.se
(Intercept) 4.40     0.20
temp        0.08     0.01
---
  n = 12, k = 2
  residual deviance = 0.2, null deviance = 1.4 (difference = 1.2)
  overdispersion parameter = 0.0
  residual sd is sqrt(overdispersion) = 0.14
```

Model 2: Log-transformed Gaussian GLM

```
log.lin.sig <- summary(log.lin.mod)$dispersion
log.lin.pred <- exp(predict(log.lin.mod) + 0.5*log.lin.sig)
```



Model 2: Log-transformed Gaussian GLM

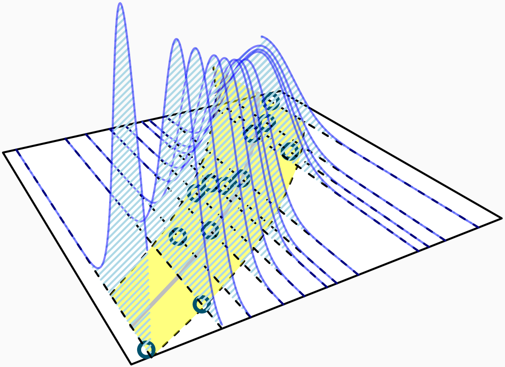


Figure 5: $y_i \sim \log \mathcal{N}(\mu_i, \sigma^2),\ \mu_i = 4.4 + 0.08\, x_i$ and $\mathbb{E}[y_i] = \exp(4.4 + 0.08\, x_i + \frac{1}{2}\, 0.14^2)$

Model 2: Critque

This plot looks a little better than the previous linear model and it predicts that I would sell, on average, 82 ice creams when the temperature is 0ºC:

```
exp(coef(log.lin.mod)[1] + 0.5 * log.lin.sig)
```

```
(Intercept)
    82.3945
```

But the assumed model distribution generates real numbers, while ice cream sales only occur in whole numbers.

Model 3: Poisson GLM

Model 3: Poisson GLM

The classic approach for count data is the Poisson distribution.

The Poisson distribution has only one parameter, here μ_i, which is also its expected value. The canonical link function for μ_i is the logarithm, i.e. the logarithm of the expected value is regarded a linear function of the predictors.

This is distinctively different to the log-transformed linear model above, where the original data was transformed, not the expected value of the data.

Model 3: Poisson GLM

Although the expected value of my observation is a real number, the Poisson distribution will generate only whole numbers, in line with the actual sales.

$$y_i \sim \text{Poisson}(\mu_i) \tag{7}$$

$$\mathbb{E}[y_i] = \mu_i = \exp(\alpha + \beta x_i) = \exp(\alpha)\exp(\beta x_i) \tag{8}$$

$$\log(\mu_i) = \alpha + \beta x_i \tag{9}$$

Note, the exponential function turns the additive scale into a multiplicative one.

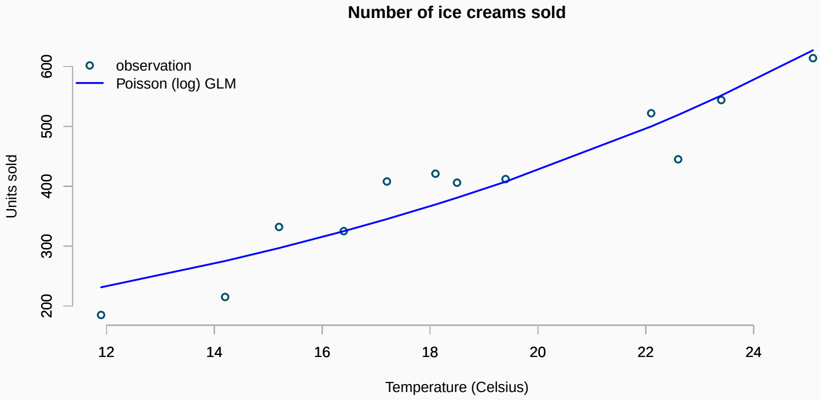
Model 3: Poisson GLM

```r
pois.mod <- glm(units ~ temp, data=icecream,
              family=poisson(link="log"))
display(pois.mod)
```

```
glm(formula = units ~ temp, family = poisson(link = "log"), data = icecream)
            coef.est coef.se
(Intercept) 4.54     0.08
temp        0.08     0.00
---
  n = 12, k = 2
  residual deviance = 60.0, null deviance = 460.1 (difference = 400.1)
```

Model 3: Poisson GLM

```
pois.pred <- predict(pois.mod, type="response")
```



Model 3: Poisson GLM

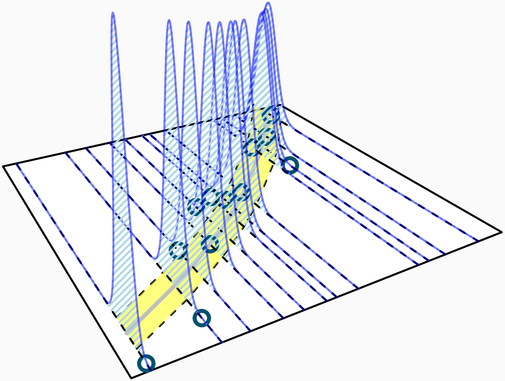


Figure 6:

$y_i \sim \text{Poisson}(\mu_i)$, $\log(\mu_i) = 4.54 + 0.08\, x_i$ and $\mathbb{E}[y_i] = \mu_i$

Model 3: Critque

So far, so good. My model is in line with my observations. Additionally, it will not predict negative sales and if I would simulate from a Poisson distribution I will always only get whole numbers back.

However, my model will also predict that I should expect to sell over 1000 ice creams when the temperature reaches 32ºC:

```
predict(pois.mod, newdata=data.frame(temp=32), type="response")
```

```
       1 
1056.651 
```

Model 4: Binomial GLM

Model 4: Binomial GLM

Ok, let's me think about the problem this way: If I have 800 potential sales then I'd like to understand the proportion sold at a given temperature.

This suggests a binomial distribution for the number of successful sales out of 800.

The key parameter for the binomial distribution is the probability of success, the probability that someone will buy ice cream as a function of temperature.

Model 4: Binomial GLM

Dividing my sales statistics by 800 would give me a first proxy for the probability of selling all ice cream.

Therefore I need an S-shape curve that maps the sales statistics into probabilities between 0 and 100%.

A canonical choice is the logistic function:

$$\text{logistic}(u) = \frac{e^u}{e^u + 1} = \frac{1}{1 + e^{-u}}$$

Model 4: Binomial GLM

The Binomial GLM can be described as

$$
\begin{aligned}
y_i &\sim \mathsf{Binom}(n, \mu_i) && (10)\\
\mathsf{logit}(\mu_i) &= \alpha + \beta x_i && (11)\\
\mathbb{E}[y_i] &= \mu_i = \mathsf{logit}(\alpha + \beta x_i) && (12)\\
& && (13)
\end{aligned}
$$

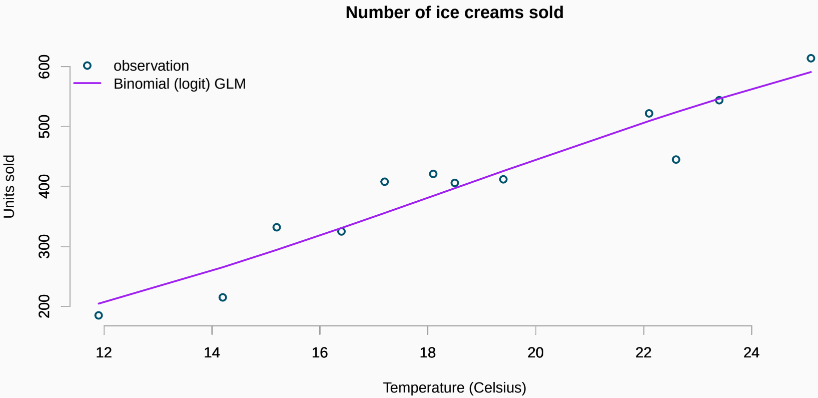
Model 4: Binomial GLM

```r
market.size <- 800
icecream$opportunity <- market.size - icecream$units
bin.glm <- glm(cbind(units, opportunity) ~ temp, data=icecream,
    family=binomial(link = "logit"))
display(bin.glm)
```

```
glm(formula = cbind(units, opportunity) ~ temp, family = binomial(link = "logit
    data = icecream)
            coef.est coef.se
(Intercept) -2.97     0.11
temp         0.16     0.01
---
  n = 12, k = 2
  residual deviance = 84.4, null deviance = 909.4 (difference = 825.0)
```

Model 4: Binomial GLM

```
bin.pred <- predict(bin.glm, type="response")*market.size
```



Model 4: Binomial GLM

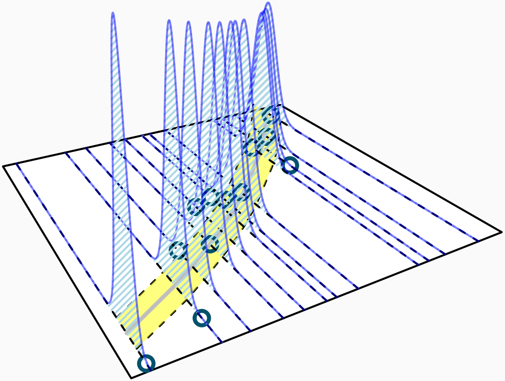


Figure 7: $y_i \sim$ Binom(n, μ_i), logit$(\mu_i) =$ $-2.97 + 0.16\, x_i$ and $\mathbb{E}[y_i] = \mu_i =$ logit$^{-1}(\mu_i)$

Model 4: Critque

This model doesn’t look too bad at all!

I can predict sales at 0ºC and 35ºC using the inverse of the logistic function, which is given in R as `plogis`:

```
plogis(coef(bin.glm)[1])*market.size
```

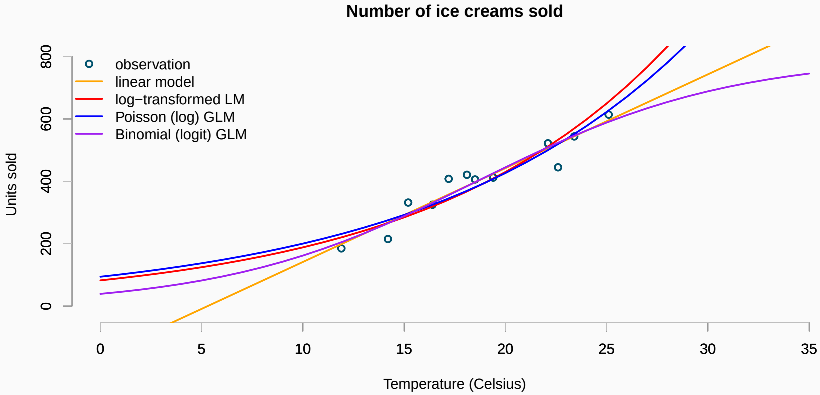
```
(Intercept)
   39.09618
```

```
plogis(coef(bin.glm)[1] +  coef(bin.glm)[2]*35)*market.size
```

```
(Intercept)
   745.7449
```

Summary

Let's bring all the models together into one graph, with temperatures ranging from 0 to 35ºC.



Conclusions

Fitting a model to data requires more than just applying an algorithm. In particular it is worth to think about:

- the range of expected values: are they bounded or range from $-\infty$ to ∞?
- the type of observations: do I expect real numbers, whole numbers or proportions?
- how to link the distribution parameters to the observations

Further observations

There are many aspects of GLMs which I haven't touched on here, such as:

- all the above models incorporate a fixed level of volatility. However, in practice, the variability of making a sale at low temperatures might be significantly different than at high temperatures. Check the residual plots and consider an over-dispersed model.
- I used so called canonical link functions in my models, which have nice theoretical properties, but other choices are possible as well.
- the distribution has to be chosen from the exponential family, e.g. Normal, Gamma, Poisson, binomial, Tweedie, etc.

Further readings and more details

Books:

- Generalized Linear Models, *P. McCullagh and J.A.Nelder*
- An Introduction to Generalized Linear Models, *Annette J. Dobson and Adrian Barnett*
- Data Analysis Using Regression and Multilevel/ Hierarchical Models, *Andrew Gelman and Jennifer Hill*

On my blog:

- Generalized Linear Models in R
- Visualising theoretical distributions of GLMs
- Bayesian regression models using Stan in R

The End. Thanks.