Beyond Ordinary Least Squares

Shane T. Mueller shanem@mtu.edu

2019-02-19

# Generalized Linear Model (GLM)

The generalized lineal model is a framework for fitting and testing versions of the linear regression model that are more flexible than ordinary least squares. Unlike non-parametric models, GLM models do not “relax” assumptions-they make alternative assumptions, often permitting us to fit different noise distributions, and different non-linear relationships that we might have otherwise used a transfromation to achieve.

Some suggested reading/resources include:

* R In action, Chapter 13
* Faraway’s, “Extending the linear model with R” Chapter 6
* Modern Applied Statistics in S, Chapter 7

# Assumptions of Ordinary Least Squares (OLS)

Ordinary Least Squares (OLS) regression makes a number of assumptions, which include:

* The effects are linear
* Error is multivariate normal
* Observations are independent
* Variance is uniform (no heteroscedascity)
* Each observation has equal value in estimating the parameters
* (Typically) the best-fitting model is one that minimizes least-squares error

Violations of these assumptions sometimes mean the model is a poor description of the data. At other times, even if the model is reasonable, violating these will impact our inferential statistics–the extent to which we believe our effects are statistically significant. Some of the approaches to this include using transformations to make effects more linear or error more normal. If you have non-independent observations because of repeated measures, you can sometimes use error strata and repeated measures-ANOVA schemes to account for this, or mixed-effects models.

There are a number of ways that have been used to adopt alternate assumptions and processes we use to extend the model.

* The **General Linear Model** typically refers to the entire set of linear regression techniques, including the potential for multiple DVS and multiple IVs, including categorical numerical predictors. Thus, ANOVA, MANOVA, regression, t-tests, etc. are all subclasses of this method.
* **Maximum likelihood estimation** (ML Estimation) is an alternative to least-squares that attempts to find the model parameters that maximize the likelihood of the model. It is especially useful if you have assumed the error is not normal-especially if it is asymmetric. In these cases, least-squares regression will be biased toward the tail of the distribution. Related methods include those that use Bayesian inference.
* **Generalized Linear models** are extensions to linear regression that incorporate distinct error distributions and transformation functions (called linking functions). These include logistic regression, probit regression, poisson regression, log-linear models, and a handful of similar and related regression methods.
* **Robust regression** methods will attempt to fit models having conditional error distributions that are non-normal, often using alternative means for estimating predictors to avoid biases that might otherwise be introduced.
* **Weighted least-squares**, which permit giving different observations different weights.

# Weighted Least Squares

One way we can extend the OLS model is to consider weighted regression. Sometimes, to create a better, more predictive, or more representative model, we might want to weigh some values more than others. Consider the state.x77 data set:

head(state.x77)

 Population Income Illiteracy Life Exp Murder HS Grad Frost
Alabama 3615 3624 2.1 69.05 15.1 41.3 20
Alaska 365 6315 1.5 69.31 11.3 66.7 152
Arizona 2212 4530 1.8 70.55 7.8 58.1 15
Arkansas 2110 3378 1.9 70.66 10.1 39.9 65
California 21198 5114 1.1 71.71 10.3 62.6 20
Colorado 2541 4884 0.7 72.06 6.8 63.9 166
 Area
Alabama 50708
Alaska 566432
Arizona 113417
Arkansas 51945
California 156361
Colorado 103766

 Population Income Illiteracy Life Exp Murder HS Grad Frost Area

Alabama 3615 3624 2.1 69.05 15.1 41.3 20 50708 Alaska 365 6315 1.5 69.31 11.3 66.7 152 566432 Arizona 2212 4530 1.8 70.55 7.8 58.1 15 113417 Arkansas 2110 3378 1.9 70.66 10.1 39.9 65 51945 California 21198 5114 1.1 71.71 10.3 62.6 20 156361

Suppose we want to compute whether murder rate can be predicted based on income, illiteracy, and high school graduation rate

state <- as.data.frame(state.x77)
colnames(state)[c(4,6)] <- c("LifeExp","HS")
state$name <- rownames(state)
model1 <- lm(Murder~Income+Illiteracy+HS,data=state)
summary(model1)

Call:
lm(formula = Murder ~ Income + Illiteracy + HS, data = state)

Residuals:
 Min 1Q Median 3Q Max
-6.0348 -1.7922 0.1427 1.5711 6.9795

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.7936267 4.3865984 0.409 0.685
Income 0.0009038 0.0007918 1.141 0.260
Illiteracy 4.1229869 0.8309778 4.962 1e-05 \*\*\*
HS -0.0611705 0.0718824 -0.851 0.399
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.669 on 46 degrees of freedom
Multiple R-squared: 0.5093, Adjusted R-squared: 0.4773
F-statistic: 15.91 on 3 and 46 DF, p-value: 3.103e-07

In this model, illiteracy predicts murder rate, but income and HS does not. But, states differ substantially in their population. If we consider the larger states more strongly, we can downplay the small states, which might possibly be noisier observations. We can do this by providing the weights argument:

model2 <- lm(Murder~Income+Illiteracy+HS,weights=Population,data=state)
summary(model2)

Call:
lm(formula = Murder ~ Income + Illiteracy + HS, data = state,
 weights = Population)

Weighted Residuals:
 Min 1Q Median 3Q Max
-363.70 -80.86 -29.02 46.77 341.46

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.7255219 4.7203120 -0.154 0.8785
Income 0.0020790 0.0009418 2.207 0.0323 \*
Illiteracy 4.4116511 0.8536518 5.168 4.99e-06 \*\*\*
HS -0.1060555 0.0768549 -1.380 0.1743
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 154 on 46 degrees of freedom
Multiple R-squared: 0.5179, Adjusted R-squared: 0.4864
F-statistic: 16.47 on 3 and 46 DF, p-value: 2.08e-07

We can see that now, income matters. It is hard to understand why, so let’s look at some scatterplots:

library(GGally)
library(dplyr)
ggpairs(select(state,Murder,Population,Income, Illiteracy,HS),progress=F)



These show correlations between Murder rate and income might be impacted by an outlier. We can make the size of the point scale to population size, to see why the weighted model might be doing better:

library(ggplot2)
library(ggrepel)

ggplot(state,aes(x=Income,y=Murder,size=Population)) + geom\_point() + geom\_text\_repel(aes(label=name))



 cor(state$Income,state$Murder)

[1] -0.2300776

Notice how when all states are equally weighted, we don’t see a significant relationship between income and Murder rate. Maybe this is unfair because some states are much larger than others. If we force the deviations for the larger states to have a larger impact, we see a significant (positive) relationship. Notice that the primary relationship between income and murder rate is negative, but this is fighting against a very strong impact of illiteracy–once illiteracy is considered, there was a positive relationship between murder rate and income. This is difficult to interpret completely, because it might alternately be reasonable to include population as a predictor.

# Generalized Linear Regression

Many times, we can suspect from first principles that an ordinary linear regression model will be incorrect, mainly because of the type of outcome/response data we are trying to fit. For example, if your outcome is all ‘true’ or ‘false’, regardless of how your model is made, the errors will all be one of two values–definitely not normally distributed. If we adapt three things about the ‘ordinary least squares’ linear regression model, we can have a lot more flexibility while maintaining reasonable understanding of the relationship.

1. Use Maximum likelihood instead of minimize residuals
2. Permit transforms of the outcome variable (called a ‘link’ function)
3. Allow exponential-family error distributions to permit non-normal residuals.

So, instead of the ‘ordinary least squares’ regression model:

$y=β\_{0}+β\_{1}\*x\_{1}+β\_{2}\*x\_{2}...β\_{n}\*x\_{n}+ϵ$ where $ϵ N(0,σ^{2})$ (i.e., the errors are normally distributed)

we have

$G(y)=β\_{0}+β\_{1}\*x\_{1}+β\_{2}\*x\_{2}...β\_{n}\*x\_{n}+ϵ$

where $ϵ$, the error distribution, can take be any exponential-family dsitribution (including normal, poisson, binomial, and others). Nelder and Wedderburn (1972), who demonstrated how #1 could be used to estimate parameters for when #2 and #3 are adopted, are often credited as introducing and popularizing the notion. In fact, one of the major contributions was showing how weighted least squares could be used iteratively to obtain the maximum likelihood estimate. An slight alternative used by glm in R is called Fisher’s Scoring iterations, which are reported in the output. So, the slight downside of the glm is it can no longer be estimated directly with simple statistics, but it can be estimated numerically using methods a few rounds of regression. Thus, computationally it is typically a little slower than normal regression, but unless you have very large data sets it probably won’t make much of a difference.

McCullagh and Nelder (1989) are one of the most comprehensive books on the topic.

Because the GLM uses Maxmimum Likelihood, it adopts a a generalized notion of variance called `deviance’ which is defined as $−2L\_{max}$. Nelder makes generalizations to the analysis of variance procedures called analysis of deviance. However, you should recognize that deviance, if adjusted appropriately be number of parameters to penalize for model complexity becomes the AIC information metric (and the BIC if also adjusted for the number of observations.)

## Generalized Linear Regression as linear regression.

If the link function is the “identity” function $f(x)=x$ and the error distribution is guassian/normal, we get ordinary least squares (albeit estimated with maximum likeliood)

x <-matrix(runif(40),ncol=4)
y <- 3 + x %\*% c(-1,3.5, 2.2, -4) + rnorm(10)\*2

summary(lm(y~x))

Call:
lm(formula = y ~ x)

Residuals:
 1 2 3 4 5 6 7 8 9
 1.5201 0.1396 -1.7918 -1.6230 1.8895 -1.4353 0.7085 1.7029 -2.4445
 10
 1.3340

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.3483 2.1479 0.628 0.5577
x1 0.6451 2.9962 0.215 0.8380
x2 7.2513 3.2284 2.246 0.0746 .
x3 1.2459 2.3656 0.527 0.6209
x4 -4.0327 2.3927 -1.685 0.1527
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.234 on 5 degrees of freedom
Multiple R-squared: 0.6526, Adjusted R-squared: 0.3747
F-statistic: 2.348 on 4 and 5 DF, p-value: 0.1872

summary(glm(y~x,family=gaussian()))

Call:
glm(formula = y ~ x, family = gaussian())

Deviance Residuals:
 1 2 3 4 5 6 7 8
 1.5201 0.1396 -1.7918 -1.6231 1.8895 -1.4353 0.7085 1.7029
 9 10
-2.4445 1.3340

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.3483 2.1479 0.628 0.5577
x1 0.6451 2.9962 0.215 0.8380
x2 7.2513 3.2284 2.246 0.0746 .
x3 1.2459 2.3656 0.527 0.6209
x4 -4.0327 2.3927 -1.685 0.1527
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 4.992451)

 Null deviance: 71.860 on 9 degrees of freedom
Residual deviance: 24.962 on 5 degrees of freedom
AIC: 49.527

Number of Fisher Scoring iterations: 2

Notice how the estimates of parameter values are the same, but now we get ‘deviance’. Deviance is related to the log-likelihood of the data, rather than the sum-squared error.

## Modeling counts data: Poisson Generalized Linear Regression

Consider data that are counts–usually small whole numbers. For example, you may have a large epidemiological study with 10,000 enrollees; half of whom took a vaccine to prevent an infection that leads to cancer, and another half who did not. At the end of the day, you may have 25 people in the control group, in comparison to 13 in the vaccine group, who were diagnosed with the cancer after 5 years. Did the treatment have an effect? It appears to have reduced the incidence of a low-probability event by about half, which would be great if you could have identified who the twelve people were saved and just give the vaccine to them.

A standard regression model might be fine here, but there are a few reasons why you might entertain a glm. First, counts have a lower value of 0, and can only take on whole numbers–two ways the normal error distribution may be violated. However, the Poisson distribution is designed for this sort of data–it is a discrete distribution predicting the number of times an event would occur given a known rate of occurrence. Also, counts are ratio-scaled, so that a value twice as large as another has a real meaning. This implies a logarithmic transform (link) is reasonable. For , example, suppose you found that women were twice as likely to get the disease as men. You might assume that the treatment would reduce the impact of the disease by 50% in both men and women, which is a linear effect on log(incidence), or a log-link function.

As a worked example, consider the FIFI World Cup data set described in the following paper: <http://www.public.iastate.edu/~curleyb/research.html>

Here, they collected statistics from FIFA World Cup matches dating back to 1930. Let’s consider the total number of yellow and red cards issued over time. This is the sum of three columns.

wc <- read.csv("world\_cup.csv")
head(wc)

 X Year Winners Runnersup Team MP GF GA PEN SOG SW FKR OFF CK W D L
1 1 1930 Uruguay Argentina Romania 2 3 5 0 3 0 0 0 0 1 0 1
2 2 1930 Uruguay Argentina Uruguay 4 15 3 0 15 0 0 0 0 4 0 0
3 3 1930 Uruguay Argentina USA 3 7 6 0 7 0 0 0 0 2 0 1
 Y YCR R FC FS Loc
1 0 0 0 0 0 Uruguay
2 0 0 0 0 0 Uruguay
3 0 0 0 0 0 Uruguay
 [ reached getOption("max.print") -- omitted 3 rows ]

wc$Cards <- wc$Y + wc$YCR + wc$R

According to the data description, the columns are Team *MP = matches played,* GF = goals for, *GA = goals against,* PEN = penalty goals *SOG = shots on goal,* SW = shots wide, *FKR = free kicks received,* OFF = offsides, *CK = corner kicks,* W = wins, *D = draws,* L = losses, *Y = yellow cards,* YCR = second yellow card and red card, *R = red card,* FC= fouls committed, *FS = fouls suffered,* Year =year of world cup, \*Loc =host nation.

To predict cards given, maybe we want to see whether players are more likely to behave badly in response to game situations and rule violations. We can compute the end total points and margin of victory, and also use number of Offsides, fouls committed, and fouls against. We will start with a normal linear regression, using a polynomial of power 3 for to capture the non-linear incerase over time.

wc$margin <- wc$GF-wc$GA
wc$total <- wc$GF+wc$GA

model <- lm(Cards~poly(Year,3)+ margin + total+SOG+OFF+FC+FS,data=wc)

wc$fit1 <- model$fitted.values

ggplot(wc,aes(x=Year,y=Cards))+geom\_point()+geom\_jitter() +
 geom\_point(aes(x=Year,y=fit1),colour="gold",size=2) + theme\_bw()



summary(model)

Call:
lm(formula = Cards ~ poly(Year, 3) + margin + total + SOG + OFF +
 FC + FS, data = wc)

Residuals:
 Min 1Q Median 3Q Max
-7.2262 -1.6405 -0.2274 1.2988 16.2638

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.27728 0.36377 9.009 < 2e-16 \*\*\*
poly(Year, 3)1 56.12785 4.10635 13.669 < 2e-16 \*\*\*
poly(Year, 3)2 1.11873 3.78778 0.295 0.767885
poly(Year, 3)3 -25.97815 3.14389 -8.263 2.35e-15 \*\*\*
margin 0.05211 0.04329 1.204 0.229480
total 0.06229 0.03690 1.688 0.092211 .
SOG 0.08743 0.04521 1.934 0.053851 .
OFF 0.13755 0.05779 2.380 0.017787 \*
FC 0.07549 0.01944 3.883 0.000122 \*\*\*
FS -0.08283 0.02134 -3.882 0.000122 \*\*\*
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.788 on 383 degrees of freedom
Multiple R-squared: 0.6706, Adjusted R-squared: 0.6629
F-statistic: 86.65 on 9 and 383 DF, p-value: < 2.2e-16

Overall, the model fits reasonably well ($R^{2}=.67$). Furthermore, most of the predictors are significant, indicating that total points and shots-on-goal are marginally predictive, but offsides and other fouls are predictive of this ‘bad behavior’ warnings.

However, let’s fit this as a glm. Here, the model is identical, but we specify family=poisson. We don’t need to specify the link as log here, because that is the default, but we did anyway.

model2 <- glm(Cards~poly(Year,3)+margin+total+SOG+OFF+FC+FS,
 family=poisson(link="log"),data=wc)
wc$fitted <- model2$fitted.values

ggplot(wc,aes(x=Year,y=Cards))+geom\_point()+geom\_jitter()+
 geom\_point(aes(x=Year,y=fitted),colour="gold",size=2) + theme\_bw()



summary(model2)

Call:
glm(formula = Cards ~ poly(Year, 3) + margin + total + SOG +
 OFF + FC + FS, family = poisson(link = "log"), data = wc)

Deviance Residuals:
 Min 1Q Median 3Q Max
-3.2874 -0.9027 -0.2344 0.4032 4.9075

Coefficients:
 Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.447006 0.111154 4.021 5.78e-05 \*\*\*
poly(Year, 3)1 34.139050 3.177729 10.743 < 2e-16 \*\*\*
poly(Year, 3)2 -11.823660 2.553596 -4.630 3.65e-06 \*\*\*
poly(Year, 3)3 -2.449453 1.195994 -2.048 0.04056 \*
margin 0.022400 0.006821 3.284 0.00102 \*\*
total 0.026312 0.006100 4.313 1.61e-05 \*\*\*
SOG 0.002654 0.005378 0.494 0.62163
OFF 0.009177 0.006379 1.439 0.15024
FC 0.010630 0.002324 4.574 4.79e-06 \*\*\*
FS -0.010266 0.002491 -4.121 3.78e-05 \*\*\*
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

 Null deviance: 1998.89 on 392 degrees of freedom
Residual deviance: 505.91 on 383 degrees of freedom
AIC: 1566.5

Number of Fisher Scoring iterations: 6

In this case, the results are fairly similar, except SOG/OFF are no longer significant but margin and total are. This happens in part because these predictors all impact the total number of cards as a proportion of baseline, not as an additive value. So, margin and total impact the proportion of red cards, allowing them to have very small impact back when no cards were given, but when 10-15 cards are given per team in the tournament, these might impact by a larger margin.

## Deviance and Comparing Models

The Null deviance is like the total residual error for the intercept-only model, and “Residual deviance” is the RSE for the current model. We can examine and compare models usung these deviance scores, or via the AIC. Suppose we want to see if a 4th order polynomial is better:

model2b <- glm(Cards~poly(Year,4)+margin+total+SOG+OFF+FC+FS,
 family=poisson(link="log"),data=wc)
summary(model2b)

Call:
glm(formula = Cards ~ poly(Year, 4) + margin + total + SOG +
 OFF + FC + FS, family = poisson(link = "log"), data = wc)

Deviance Residuals:
 Min 1Q Median 3Q Max
-3.3457 -0.8464 -0.3531 0.4335 4.8571

Coefficients:
 Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.496790 0.092442 5.374 7.70e-08 \*\*\*
poly(Year, 4)1 30.585464 2.312243 13.228 < 2e-16 \*\*\*
poly(Year, 4)2 -7.258298 1.991481 -3.645 0.000268 \*\*\*
poly(Year, 4)3 -6.806050 1.434959 -4.743 2.11e-06 \*\*\*
poly(Year, 4)4 2.931448 0.850117 3.448 0.000564 \*\*\*
margin 0.022528 0.006795 3.315 0.000916 \*\*\*
total 0.027778 0.006131 4.531 5.87e-06 \*\*\*
SOG 0.000901 0.005402 0.167 0.867537
OFF 0.009266 0.006370 1.455 0.145757
FC 0.010948 0.002322 4.715 2.41e-06 \*\*\*
FS -0.009584 0.002496 -3.839 0.000123 \*\*\*
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

 Null deviance: 1998.89 on 392 degrees of freedom
Residual deviance: 494.79 on 382 degrees of freedom
AIC: 1557.4

Number of Fisher Scoring iterations: 6

AIC(model2)

[1] 1566.515

AIC(model2b)

[1] 1557.393

anova(model2, model2b,test="Chisq")

Analysis of Deviance Table

Model 1: Cards ~ poly(Year, 3) + margin + total + SOG + OFF + FC + FS
Model 2: Cards ~ poly(Year, 4) + margin + total + SOG + OFF + FC + FS
 Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1 383 505.91
2 382 494.79 1 11.122 0.0008531 \*\*\*
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Deviance is often compared using a Chi-squared test, although the AIC measure is also appropriate. Here, the model2b has a lower AIC, and is significantly better than the smaller model–both supporting the suggestion that we use a 4th-order polynomial. Now, could we test specifically whether SOG and OFF matter?

model2c <- glm(Cards~poly(Year,4)+margin+total+FC+FS,
 family=poisson(link="log"),data=wc)
summary(model2c)

Call:
glm(formula = Cards ~ poly(Year, 4) + margin + total + FC + FS,
 family = poisson(link = "log"), data = wc)

Deviance Residuals:
 Min 1Q Median 3Q Max
-3.3338 -0.8641 -0.3493 0.4296 4.8880

Coefficients:
 Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.490023 0.092224 5.313 1.08e-07 \*\*\*
poly(Year, 4)1 30.632610 2.308156 13.271 < 2e-16 \*\*\*
poly(Year, 4)2 -7.228834 1.988595 -3.635 0.000278 \*\*\*
poly(Year, 4)3 -6.812957 1.433144 -4.754 2.00e-06 \*\*\*
poly(Year, 4)4 2.953194 0.845729 3.492 0.000480 \*\*\*
margin 0.023865 0.005733 4.163 3.14e-05 \*\*\*
total 0.028762 0.005061 5.683 1.32e-08 \*\*\*
FC 0.011637 0.002290 5.081 3.76e-07 \*\*\*
FS -0.008519 0.002204 -3.865 0.000111 \*\*\*
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

 Null deviance: 1998.89 on 392 degrees of freedom
Residual deviance: 496.99 on 384 degrees of freedom
AIC: 1555.6

Number of Fisher Scoring iterations: 6

AIC(model2)

[1] 1566.515

AIC(model2b)

[1] 1557.393

AIC(model2c)

[1] 1555.595

anova(model2, model2b,model2c,test="Chisq")

Analysis of Deviance Table

Model 1: Cards ~ poly(Year, 3) + margin + total + SOG + OFF + FC + FS
Model 2: Cards ~ poly(Year, 4) + margin + total + SOG + OFF + FC + FS
Model 3: Cards ~ poly(Year, 4) + margin + total + FC + FS
 Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1 383 505.91
2 382 494.79 1 11.1219 0.0008531 \*\*\*
3 384 496.99 -2 -2.2016 0.3326023
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

This model is better in the AIC sense. But now, it does not differ according to a $χ^{2}$ test from model2b which would suggest the simpler model2c is better.

## Interpreting parameters

In the Poisson model, the parameter estimates are interpretable in terms of their impact on the log-outcome.

$log(y)=β\_{0}+β\_{1}x\_{1}+β\_{2}x\_{2}...$

Which implies:

$y=exp(β\_{0}+β\_{1}x\_{1}+β\_{2}x\_{2}...)$

$y=exp(β\_{0})exp(β\_{1}x\_{1})exp(β\_{2}x\_{2})...$

You can see how y is the product of each $βx$ term exponentiated. But since the $β$ values are available, we can re-write this as each $exp(β)$ value raised to the $x\_{i}$ power. Focusing on just a few of the parameters:

$y=exp(.49)×exp(.02386)^{margin}×exp(.0287)^{total}$

Calculating each exponential, we get:

$y=1.63×1.023^{margin}×1.029^{total}...$

This suggests that for each unit of total, the card rate increases by 2.9%; and for each point difference between winner and loser the card rate increases by 2.3%. Notice that prior to tranform, the beta weight is approximately equal to the the transformed rate -1.0. For small values, this is a well-known engineering approximation; for large values such as the intercept (.49), this no longer holds well (.49 vs .63).

## Overdispersion

On important diagnostic of the GLM is to know whether the deviance is too large. The rule of thumb is that deviance should generally be about the same as the number of the residual degrees of freedom. If it is substantially larger, we should consider a better model, a different model, or accounting for the so-called over-dispersion directly. There are many resources for determining how to deal with this.

In general, the quasi- models allow for fitting of an overdispersion parameter. With the over-dispersion model, we can force the dispersion parameter to 1 (no overdispersion) to see if it makes a difference.

model.od <- glm(Cards ~ poly(Year, 3) + margin + total + FC + FS, family = quasipoisson(link = "log"),
 data = wc)
summary(model.od)

Call:
glm(formula = Cards ~ poly(Year, 3) + margin + total + FC + FS,
 family = quasipoisson(link = "log"), data = wc)

Deviance Residuals:
 Min 1Q Median 3Q Max
-3.2118 -0.9223 -0.2312 0.3917 4.9898

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.437036 0.147188 2.969 0.003172 \*\*
poly(Year, 3)1 34.317515 4.210008 8.151 5.11e-15 \*\*\*
poly(Year, 3)2 -11.863598 3.393383 -3.496 0.000527 \*\*\*
poly(Year, 3)3 -2.390839 1.582698 -1.511 0.131708
margin 0.024944 0.007587 3.288 0.001103 \*\*
total 0.028369 0.006700 4.234 2.87e-05 \*\*\*
FC 0.011364 0.003037 3.741 0.000211 \*\*\*
FS -0.008862 0.002924 -3.030 0.002608 \*\*
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasipoisson family taken to be 1.747136)

 Null deviance: 1998.89 on 392 degrees of freedom
Residual deviance: 508.37 on 385 degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 6

anova(model.od, test = "Chisq")

Analysis of Deviance Table

Model: quasipoisson, link: log

Response: Cards

Terms added sequentially (first to last)

 Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL 392 1998.89
poly(Year, 3) 3 1361.54 389 637.35 < 2.2e-16 \*\*\*
margin 1 71.34 388 566.02 1.661e-10 \*\*\*
total 1 31.25 387 534.77 2.346e-05 \*\*\*
FC 1 10.12 386 524.65 0.01611 \*
FS 1 16.28 385 508.37 0.00227 \*\*
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

anova(model.od, dispersion = 1, test = "Chisq")

Analysis of Deviance Table

Model: quasipoisson, link: log

Response: Cards

Terms added sequentially (first to last)

 Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL 392 1998.89
poly(Year, 3) 3 1361.54 389 637.35 < 2.2e-16 \*\*\*
margin 1 71.34 388 566.02 < 2.2e-16 \*\*\*
total 1 31.25 387 534.77 2.270e-08 \*\*\*
FC 1 10.12 386 524.65 0.001468 \*\*
FS 1 16.28 385 508.37 5.469e-05 \*\*\*
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1