

# Week 6: Further GLM Topics

## MATH-516 Applied Statistics

Tomas Masak

March 27th 2023

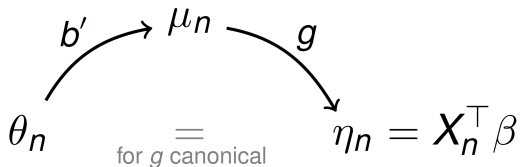
## Section 1

# Uncertainty Quantification in GLMs

# Confidence Interval

- $\sqrt{N}(\hat{\beta} - \beta) \rightarrow \mathcal{N}_p(0, I^{-1}(\theta))$  [Wald]
  - the Fisher information matrix can be consistently estimated, then...
  - easy to obtain CIs for any  $\beta_j$
  - easy to obtain CI for  $c^\top \beta$  for any  $c \in \mathbb{R}^p$
- $2[\ell(\hat{\beta}) - \ell(\beta)] \rightarrow \chi_p^2$  [LR]
  - invert numerically to obtain confidence region for the whole beta  $\beta$
  - similarly invert the model-submodel test to obtain confidence regions for some entries of  $\beta$

# Graph of GLM



- asymptotic Gaussianity for  $\beta$  and hence for  $\eta_n$ 
  - on the linear predictor scale, things are roughly Gaussian
- unless  $g$  is identity (which makes sense only for the Gaussian linear model), no Gaussianity for the modelled mean  $\mu_n$ 
  - no Gaussianity on the response scale

# Prediction Interval

- new observation  $(Y_*, X_*)^\top$  with  $Y_*$  unknown
- goal: construct interval  $(L_{Y_*}, U_{Y_*})$  depending on the fitted GLM and  $X_*$  such that  $P(Y_* \in (L_{Y_*}, U_{Y_*})) = 1 - \alpha$
- [Wald] provides  $(L_{\eta_*}, U_{\eta_*})$  such that  $P(\eta_* \in (L_{\eta_*}, U_{\eta_*})) = 1 - \alpha$
- $\Rightarrow P(\mu_* \in (g^{-1}(L_{\eta_*}), g^{-1}(U_{\eta_*}))) = 1 - \alpha$
- if  $Y_*$  is distributed according to a certain distribution, the prediction interval is given by quantiles of that distribution
- $\Rightarrow$  run  $(g^{-1}(L_{\eta_*}), g^{-1}(U_{\eta_*}))$  through the quantile function (of the response distribution estimated by the GLM) and report the minimum and maximum value as the prediction interval  $(L_{Y_*}, U_{Y_*})$ 
  - this is conservative, but it is not easy to do better because, unlike Gaussian linear models, other GLMs do not have distribution of the “error” independent of that of  $\hat{\eta}_n$
  - replace this step by Monte Carlo?
- prediction intervals are fairly useless for binary data (there CI for  $\eta_*$  is enough)

# Sources of Uncertainty in Prediction

- ① uncertainty in the model
- ② uncertainty in the model parameters
- ③ uncertainty in the new observation
- we try to remove source 1 by careful model building and diagnostics
  - we act like if we have succeeded
- sources 2 and 3 are independent for Gaussian linear model, but not for other GLMs
  - and we don't know what the form of the dependence is, so we conservatively take the worst case
- Monte Carlo simulation?
  - often people simulate only from the fitted model (i.e. parametric bootstrap), but that ignores source 2 of uncertainty
  - simulating for given  $X_*$  the whole  $\beta \mapsto \eta_* \mapsto \mu_{star} \mapsto$  "new sample" path by starting from the asymptotic distribution of  $\beta$  is better for moderate/low sample sizes

## Section 2

### Logistic vs. Log-linear Model

# Logistic vs. Log-linear Model

If  $Y$  is binary, then there is a certain equivalence between logistic and log-linear models:

- let  $Z = (Z_1, \dots, Z_p)^\top$  be all other variables
- the logistic model  $Y \sim Z$  is equivalent to the log-linear model  $\text{freq} \sim Y * Z + (Z)^\wedge p$ 
  - $Y * Z$  are interactions between  $Y$  and all  $Z_1, \dots, Z_p$
  - $(Z)^\wedge p$  denotes the full interaction term between all  $Z_1, \dots, Z_p$
- coefficients of the logistic model are exactly those corresponding to  $Y * Z$  in the log-linear model
  - including standard errors and everything else ... this is because the extra parameters of the log-linear model are fitting counts and they can be shown asymptotically independent of those that fit probabilities
- logistic regression pushes frequencies out of consideration, it does not care about the distribution of  $Z$ 's or their relationship
  - which we also often do not care about, so using logistic can be the simpler way to go



# Logistic vs. Log-linear Model

- interpretation is the same in terms of probabilities (odds and odds ratios), but interpretation of the whole model is slightly different - the difference between the two types of asymptotics
  - what can we predict?
- when  $Y$  wouldn't be binary, the log-linear model  $\text{freq} \sim Y * Z + (Z)^{\wedge} p$  would be equivalent to the *proportional odds model* (a multi-class generalization of logistic regression)
  - and including three-way interactions including  $Y$  in the log-linear model would lead to a more general model (with non-proportional odds)

# Example: Premier League Data

```
Call:
glm(formula = score ~ (home_flag * covid)^2, family = poisson,
    data = Data)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.7509	-1.5890	-0.2788	0.5635	4.4586

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	0.23309	0.03518	6.626	3.45e-11 ***
home_flag	0.19401	0.04751	4.084	4.43e-05 ***
covid1	0.06115	0.05655	1.081	0.2796
covid2	0.03331	0.05704	0.584	0.5592
home_flag:covid1	-0.18620	0.07851	-2.372	0.0177 *
home_flag:covid2	-0.04622	0.07754	-0.596	0.5511

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 3624.8 on 2799 degrees of freedom  
Residual deviance: 3600.9 on 2794 degrees of freedom  
AIC: 8597.7

Number of Fisher Scoring iterations: 5

Call:

```
glm(formula = home_flag ~ covid, family = binomial, data = Data2)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.261	-1.241	1.096	1.115	1.174

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	0.19401	0.04751	4.084	4.43e-05 ***
covid1	-0.18620	0.07851	-2.372	0.0177 *
covid2	-0.04622	0.07754	-0.596	0.5511

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 5367.5 on 3883 degrees of freedom  
Residual deviance: 5361.8 on 3881 degrees of freedom  
AIC: 5367.8

Number of Fisher Scoring iterations: 3

## What the models predict:

- Poisson (a.k.a. log-linear): how many goals will be scored in a given match
- binomial (a.k.a. logistic): was a given goal scored at home or away

# Example: Premier League Data

- the log-linear model targets goal frequencies
  - the intercept and the two covid-related coefs estimate expected baseline (away and before covid) goals
  - the other three coefficients show how the expected goals change when we move home or into/past covid
    - $e^{0.19} \approx 1.21$  is the proportional change in the expected number of goals when a team plays at home as opposed to playing away ... frequency interpretation
    - $e^{0.19} \approx 1.21$  is the odds of scoring at home against scoring away before covid ... probability interpretation
- the logistic model targets probabilities of goal being scored away/home
  - the intercept provides the baseline (before covid) probability of a goal being scored at home
    - $e^{0.19} \approx 1.21$  is the odds of success (scoring at home) against failure (scoring away) before covid ... the only and arguably a bit weird interpretation
  - the covid-related coefs show how the probability of a goal being scored at home changes when we move into/past covid

## Section 3

### GLMs for Positive Response

# Main Areas for GLM

There are three exemplary situations where a (Gaussian) linear model is inadequate:

- binary response
  - Bernoulli distribution is the only viable one
  - but still, the GLM can be wrong, e.g. due to overdispersion
- frequency (count) response
  - Poisson distribution is arguably the most natural one
  - negative binomial distribution is another option
    - related to overdispersed Poisson
    - has a quadratic variance function
- continuous positive response
  - many options for the response distribution here. . .

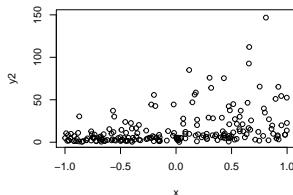
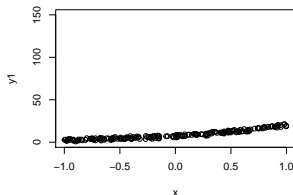
# Positive Response

Several options for the response distribution:

- ① Gaussian modelled as a GLM with a log-link
  - here the response can be technically negative
- ② log-normal
  - take a logarithm of the response and model it as Gaussian
- ③ Gamma
- ④ inverse-Gaussian

# Gaussian with a log-link vs. log-normal

```
N <- 200; set.seed(517)
x <- runif(N,-1,1)
beta0 <- 2; beta1 <- 1 #intercept and slope
y1 <- rnorm(N) + exp(beta0+beta1*x)
y2 <- exp(rnorm(N)+beta0+beta1*x)
plot(x,y1,ylim=c(0,150)); plot(x,y2,ylim=c(0,150))
```



# Some Fun on Stack Exchange

## How to specify a lognormal distribution in the glm family argument in R?

Asked 11 years ago   Modified 1 year, 5 months ago   Viewed 50k times



25

Simple question: How to specify a lognormal distribution in the GLM family argument in R? I could not find how this can be achieved. Why is `lognormal` not an option in the family argument?



12

Lognormal is not an option because the log-normal distribution is not in the [exponential family](#) of distributions. Generalized linear models can only fit distributions from the exponential family.

I'm less clear why exponential is not an option, as the exponential distribution *is* in the exponential family (as you might hope).

- 11 The lognormal is in the exponential family - it even says so in the very link you gave! See the second sentence [here](#), and see [this table](#), right above "Inverse Gaussian".

Do you have a reference for the statement that "Generalized linear models can only fit distributions from the exponential family"? – [Henrik](#) May 13, 2018 at 11:20

I would think that the people who wrote `glm` would have checked the validity of this before including the option `family=gaussian(link=log)`. – [abalter](#) Mar 5, 2020 at 19:28

@abalter `family=gaussian(link=log)` is not the same as lognormal regression. This is using the link function on the mean value as the log. Not to be confused with a log transformation of the response.

– [Therkel](#) Dec 13, 2021 at 10:06



# Distribution of a Positive Response

Recall that in exponential family:  $\text{var}(Y) = \varphi V(\mathbb{E}Y)$

- $V(\cdot)$  is the variance function

Gaussian with a log-link	log-normal	Gamma	inverse-Gaussian
$Y \sim \mathcal{N}(\cdot, \cdot)$	$\log(Y) \sim \mathcal{N}(\cdot, \cdot)$	$Y \sim \Gamma(\cdot, \cdot)$	$Y \sim IG(\cdot, \cdot)$
$\log(\mathbb{E}Y) = X^\top \beta$	$\mathbb{E} \log(Y) = X^\top \beta$	$\mathbb{E}Y = \frac{1}{X^\top \beta}$	$\mathbb{E}Y = \frac{1}{\sqrt{X^\top \beta}}$
$V(\mathbb{E}y) = 1$	$V(\mathbb{E}Y) = \mathbb{E}Y$	$V(\mathbb{E}Y) = (\mathbb{E}Y)^2$	$V(\mathbb{E}Y) = (\mathbb{E}Y)^3$

# Example: Permeability of Building Materials

- permeability (time needed for water particles to get through a material) of 81 sheets produced on 3 different machines over 9 days measured
- 2 factors as regressors (day and machine)
  - does permeability differ for the different machines?
  - does day matter?
- we will use log-links for all the model to facilitate the same interpretation

```
library(GLMsData)
data(perm)
perm$Day <- as.factor(perm$Day)
fit1_loglink <- glm(Perm ~ Mach * Day, data=perm,
                    family=gaussian(link="log"))
fit2_lognormal <- lm(log(Perm) ~ Mach * Day, data=perm)
fit3_gamma <- glm(Perm ~ Mach * Day, data=perm,
                  family=Gamma(link="log"))
fit4_igauss <- glm(Perm ~ Mach * Day, data=perm,
                  family=inverse.gaussian(link="log"))
```

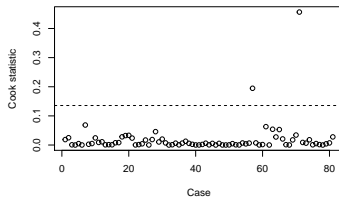
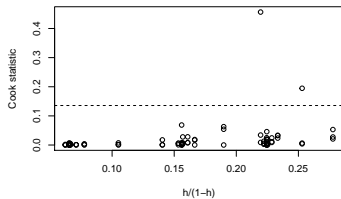
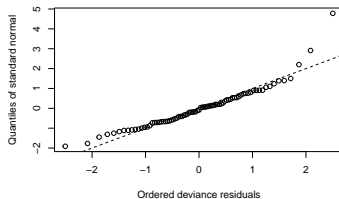
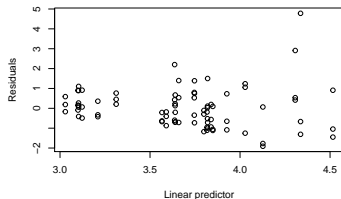
# Example: Permeability of Building Materials

- aiming for simplicity, model-submodel tests lead us to the following models:
  - notice how day matters in only two of them

```
fit1_loglink <- glm(Perm ~ Mach + Day, data=perm,  
                    family=gaussian(link="log"))  
fit2_lognormal <- lm(log(Perm) ~ Mach, data=perm)  
fit3_gamma <- glm(Perm ~ Mach + Day, data=perm,  
                  family=Gamma(link="log"))  
fit4_igauss <- glm(Perm ~ Mach, data=perm,  
                   family=inverse.gaussian(link="log"))
```

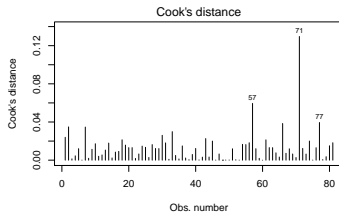
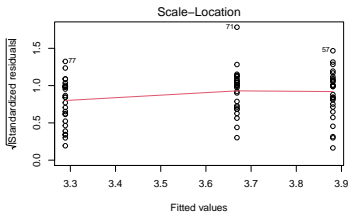
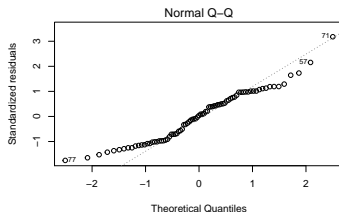
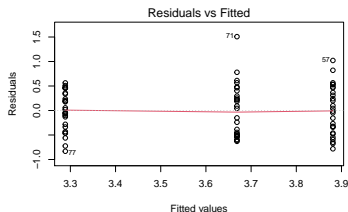
# Example: Permeability of Building Materials

```
library(boot)
glm.diag.plots(fit1_loglink)
```



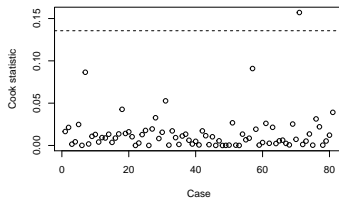
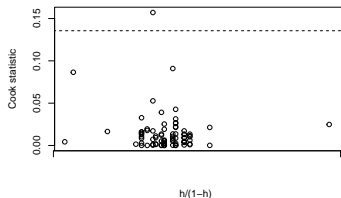
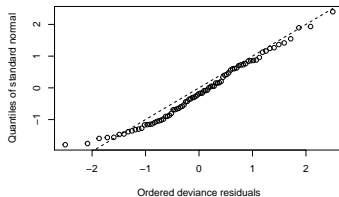
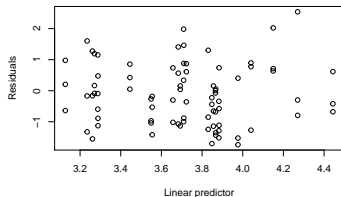
# Example: Permeability of Building Materials

```
par(mfrow=c(2,2))  
plot(fit2_lognormal,1:4)
```



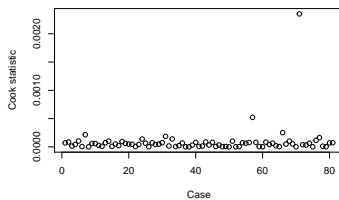
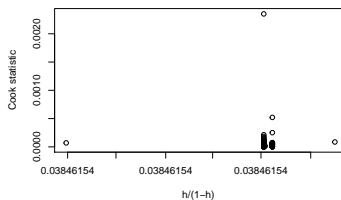
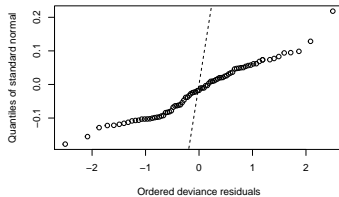
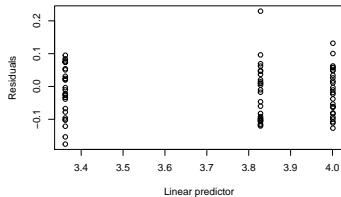
# Example: Permeability of Building Materials

```
glm.diag.plots(fit3_gamma)
```



# Example: Permeability of Building Materials

```
glm.diag.plots(fit4_igauss)
```



```
hist(residuals(fit4_igauss))
```

## Example: Permeability of Building Materials

- Gaussian model with a log-link is clearly wrong
- log-normal model is not that problematic, but the residual plots are not great
- Gamma and inverse Gaussian models are both alright
  - the inverse Gaussian QQ plot shows that the axes are flipped in boot's implementation of diagnostic plots, otherwise that plot does not display a problematic behavior!
- how do we choose between Gamma and inverse Gaussian?
  - Gaussian distribution describes a Brownian motion's level at a fixed time, which is why the hitting-time (the time it takes the Brownian motion - with a drift - to reach a fixed level) distribution is called inverse Gaussian
  - hence inverse Gaussian is likely a good model for permeability (since Brownian motion is the most common model for random movement of particles over time, so assuming a uniform material with microscopic pores... otherwise Gamma distribution as the hitting time of a Poisson process might be more appropriate for a coarse-grained material, where particles travel by jumping from grain to grain with an exponential waiting time)