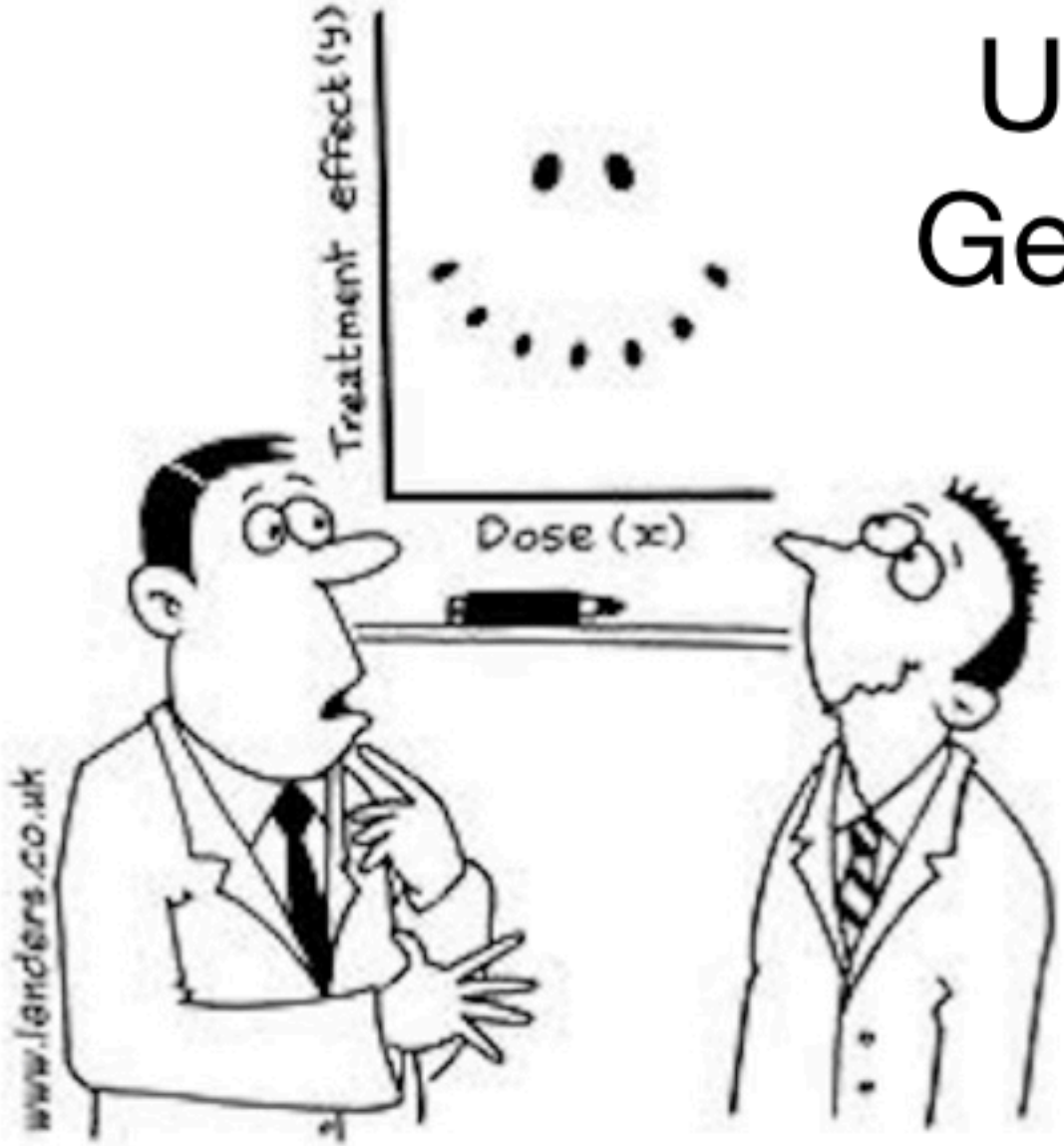
Unit 5, Lecture 1
Generalized Linear Models



"It's a non-linear pattern with outliers.....but for some reason I'm very happy with the data."

Plan & Goals for Today

Plan: I assume you have seen logistic regression, a specific case of something called "Generalized Linear Regression"

We will look at another specific family member, Poisson regression, and discuss some specifics to that case. This is a form of linear regression when the outcome is *count data.*

Trigger warning: We will be talking about stop-and-frisk and policing behavior

We will then *briefly* discuss a general framework, and touch upon instances of this general framework.

Goals:

- ★ Introduce a different kind of regression modeling
- ★ Give overview of an important case of modeling counts as outcomes
- ★ Discuss *overdispersion*, an important type of model misspecification for some models

Note: No multilevel modeling today—we will add that in next class.

Poisson Regression, a special case of Generalized Linear Models

Case study: Police stops by ethnic group

- ★ We have records of "stop and frisk" stops in New York over a period of time.
- ★ Our ideal question of interest is whether the rate of police stops is higher or lower for certain groups and precincts than we would predict given amount of crime.
- ★ We don't have these measures, so we attempt to get at this question indirectly.
- ★ Units *i* are precincts & race/ethnic group combos. (75 precincts.)

1=black, 2=hispanic, 3=white

Outcome y_i is number of stops of that group in the precinct

```
> head( stops )
  precinct eth past.arrests stops   pop
1        1   1          980   202  1720
2        1   2          295   102  1368
3        1   3          381    81 23854
4        2   1          753   132  2596
```

Example from Gelman & Hill text

Thinking through our overall estimation strategy

★ We can watch any given precinct. It has

- Number of people in our different race/ethnic groups (we assume constant).
- Amount of crime for a set period of time (for each of our groups).
- Number of police stops for a set period of time.

★ The number of stops are *count data.*

★ Goal: is the number of stops relatively larger, given crime levels, for some groups?

A Toy, Related, Example

★ Let *i* index precincts in our city

★ y_i is number of stop-and-frisks in a year

★ We are *counting* events in each precint.

★ The longer we wait, the more we will count.

★ We want to estimate the ***rate:*** the number of stop-and-frisks per unit time.

★ This is often modeled as a **Poisson distribution**

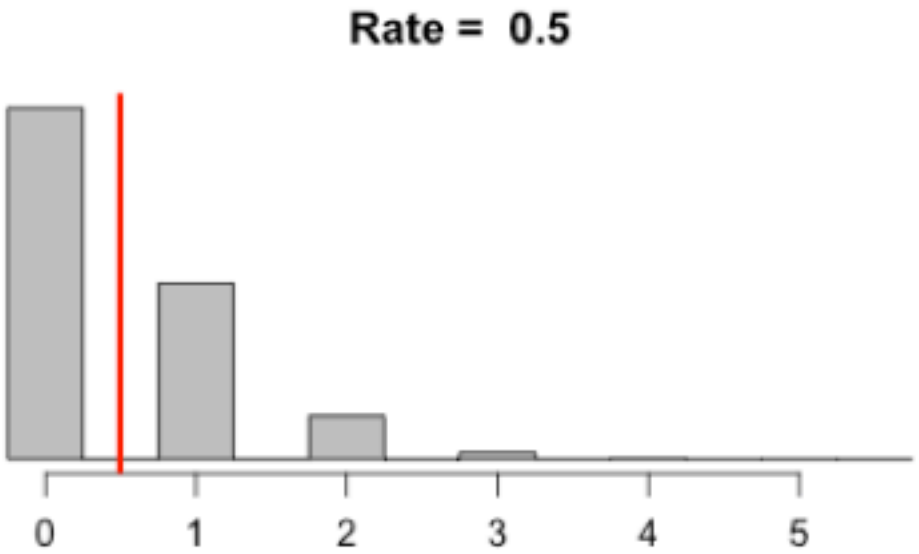
$$Y_i \sim Poisson(\theta_i)$$

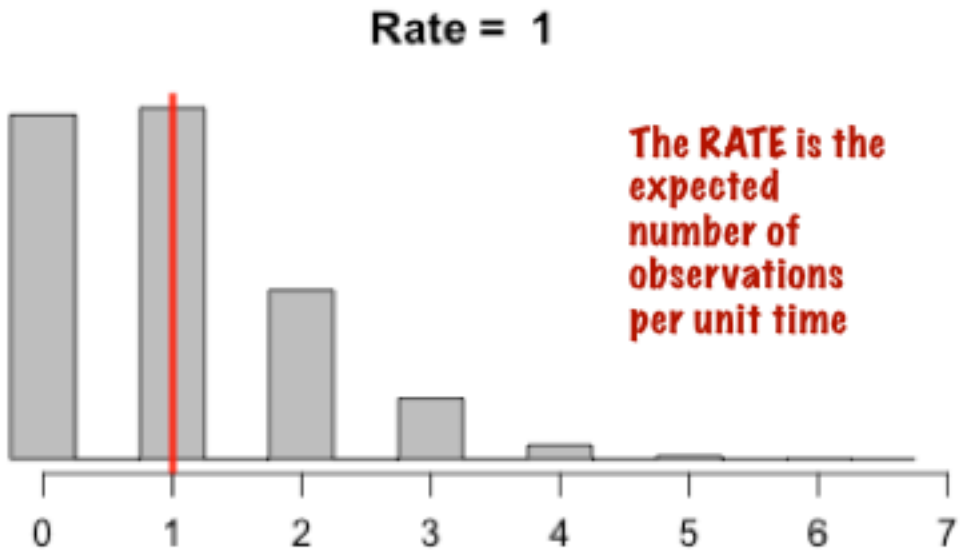
θ_i is the rate: how many we expect per unit time

General features of (Poisson) count data

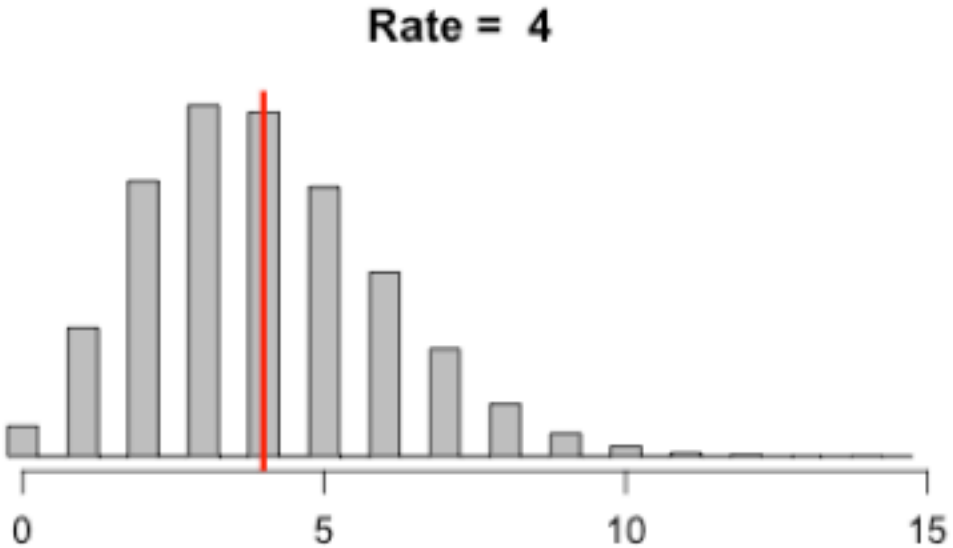
- ★ Your cases are the units being observed (the precints).
- ★ Your outcome is the number of times something happens to that case (the # stop-and-frisks).
- ★ You believe the longer you wait, the more you get.
- ★ For each unit, at any given moment, the chance of seeing an event is the same as any other moment.
- ★ This chance is the ***rate***- which is what you want to estimate.

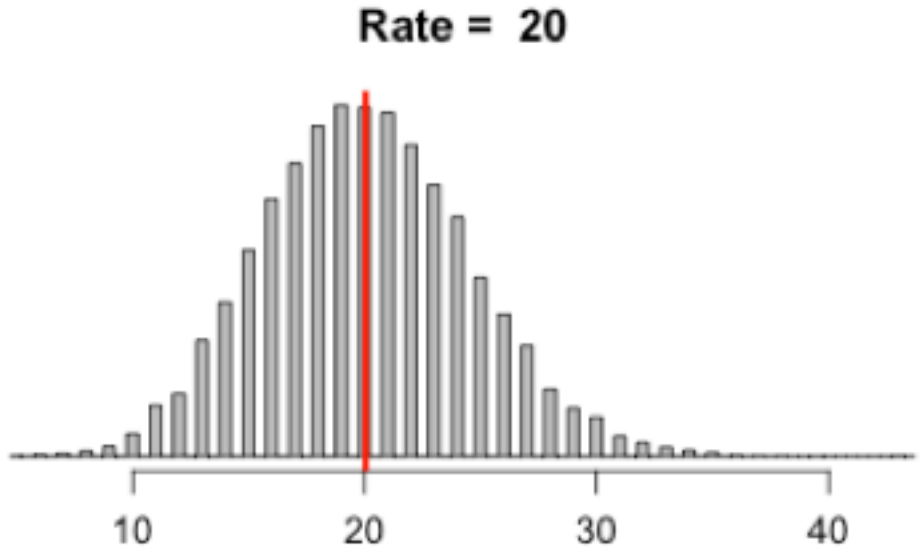
Four Poisson Distributions





The RATE is the expected number of observations per unit time





Y1 = rpois(10000, lambda=20)

Fun Poisson Facts

If

$$Y_i \sim Poisson(\theta_i)$$

then:

- ★ Y_i is going to be a nonnegative integer
- ★ The mean (expected value) of Y_i is $E[Y_i] = \theta_i$
- ★ The variance of Y_i is $Var[Y_i] = \theta_i$
- ★ Note the mean and the variance are *tied together.* This is not like a normal distribution!
- ★ Poisson distributions are indexed by a *single parameter*
- ★ This is similar to the Binomial distribution (total heads in n coin flips).

Predicting the Rate

We might want to predict the stop-and-frisk rate of each precinct with

- ★An intercept (overall average)
- ★X_1, a measure of policing intensity
- ★X_2, a dummy for being in an industrial area

These variables are hypothetical: we do not have them for the real analysis.

We could regress the counts on these using OLS (a linear model)

Or we can do something more fancy.

The Poisson Regression Model

We model the connection between our observed outcome and a rate parameter:

$$Y_i \sim Poisson(\theta_i)$$

And then we model our rate parameter as a function of our covariates:

$$\theta_i = \exp(X_i\beta)$$

Outcome is a count e.g., of puppies

Our linear predictor. This part looks like OLS

Our inverse "link function" exp(x) is e^x

Interpretation involves fiddling with the link

Say we have:

$$Y_i \sim Poisson(\exp(2.8 + 0.012X_{i1} - 0.20X_{i2}))$$

How do we interpret these coeffiicents?

Our model says the rate (stops/unit time) is this:

$$\theta_i = \exp(2.8 + 0.012X_{i1} - 0.20X_{i2})$$
$$= e^{2.8} e^{0.012X_{i1}} e^{-0.20X_{i2}}$$

Recall X_{i2} is our industrial zone dummy.

Impact of industrial zone is a scaling on average (rate): $e^{-0.20 \cdot 1} = 0.82$

Exposure: how much opportunity to see events

The longer we watch, the more stop-and-frisks we see.

Say we have observed different intersections for different lengths of time.

Time, here, is *exposure*. We expand our model:

$$Y_i \sim Poisson(u_i\theta_i) \qquad \hat{Y}_i = u_i\theta_i$$

u_i = exposure of precinct i

This hat-Y is our predicted (expected) number of observations

We often talk about *offset instead:*

$$\hat{Y}_i = \exp(\log(u_i) + X_i\beta)$$

offset term

We expect to see rate X exposure events

How define exposure in the police stops context?

★ Exposure is a scaling factor telling us how much outcome we should expect to see (more exposure = more outcome)

★ We choose: *exposure* u_i is the number of **arrests** by people of that group in that precinct in the previous year (as recorded by DCJS)

- Why arrests? We want to (attempt to) control for baseline level of crime from that group in that precinct.

1=black, 2=hispanic, 3=white

```
> head( stops )
  precinct eth past.arrests stops   pop
1        1   1          980   202  1720
2        1   2          295   102  1368
3        1   3          381    81 23854
4        2   1          753   132  2596
```

Outcome y_i is number of stops of that group in the precinct

Our data has one observation per precinct by racial group

Fitting in R: Model 1 (no predictors)

```
> fit.1 <- glm (stops ~ 1, family=poisson,
                offset=log(past.arrests), data=stops )
> display(fit.1)

            coef.est coef.se
(Intercept) -0.59     0.00
---
  n = 225, k = 1
  residual deviance = 46120.3, null deviance = 46120.3
(difference = 0.0)
```

We tell glm we are fitting a Poisson.

We use our exposure for the offset.

We get a grand rate parameter out.

Fitting in R: Model 2 (ethnic group)

Our data has one observation per precinct by racial group

```
> fit.2 <- glm (stops ~ factor(eth), family=poisson,
                offset=log(past.arrests), data=stops )
> display(fit.2)
```

1=black, 2=hispanic, 3=white

```
             coef.est coef.se
(Intercept)   -0.59     0.00
factor(eth)2   0.07     0.01
factor(eth)3  -0.16     0.01
---
  n = 225, k = 3
  residual deviance = 45437.4, null deviance = 46120.3
(difference = 682.9)
```

Twice the difference in likelihoods compared to null model (we can use this for a likelihood ratio test)

$$D \sim \chi^2_m$$

so $E[D] = m$

We expect to see 1 unit per parameter added. This is a huge difference.

Now we add precinct

```
> fit.3 <- glm (stops ~ factor(eth) +  factor(precinct),
                family=poisson,
                offset=log(past.arrests), data=stops )
> display(fit.3)
```

```
                    coef.est coef.se
(Intercept)          -1.38      0.05
factor(eth)2          0.01      0.01
factor(eth)3         -0.42      0.01
factor(precinct)2    -0.15      0.07
factor(precinct)3     0.56      0.06
...
factor(precinct)74    1.15      0.06
factor(precinct)75    1.57      0.08
---
  n = 225, k = 77
  residual deviance = 3427.1, null de
(difference = 42693.1)
```

What does this mean?

An enormous drop!
Improved fit!
Different precincts have different relative stop rates (beyond their arrest rates)

The likelihood ratio tests

```
> library( lmtest )
> lrtest( fit.1, fit.2, fit.3)
Likelihood ratio test

Model 1: stops ~ 1
Model 2: stops ~ factor(eth)
Model 3: stops ~ factor(eth) + factor(precinct)
  #Df   LogLik Df    Chisq Pr(>Chisq)
1   1 -23913.4
2   3 -23572.0  2   682.91  < 2.2e-16 ***
3  77  -2566.9 74 42010.21  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'
0.1 ' ' 1
```

Compare the 682.91 to the difference in deviance on Model 2.

Model 3 is being compared to Model 2 here, NOT the base model 1.

Interpretation of results

Stops by police are compared to number of arrests in prior year.

If coefficient for “hispanic” or “white” is greater than 1 then this group is stopped relatively more often, compared to their arrest rate, than blacks.

Same argument for precincts (as compared to precinct 1, the arbitrarily chosen baseline).

Over dispersion

Model misspecification issue: Heteroskedasticity

The Poisson model says **variance depends on rate**

Our model gives us, for each observation, what we would have expected:

$$\hat{Y}_i = u_i\hat{\theta}_i \text{ with } \hat{\theta}_i = \exp(X_i\hat{\beta}_i)$$

Our model says that

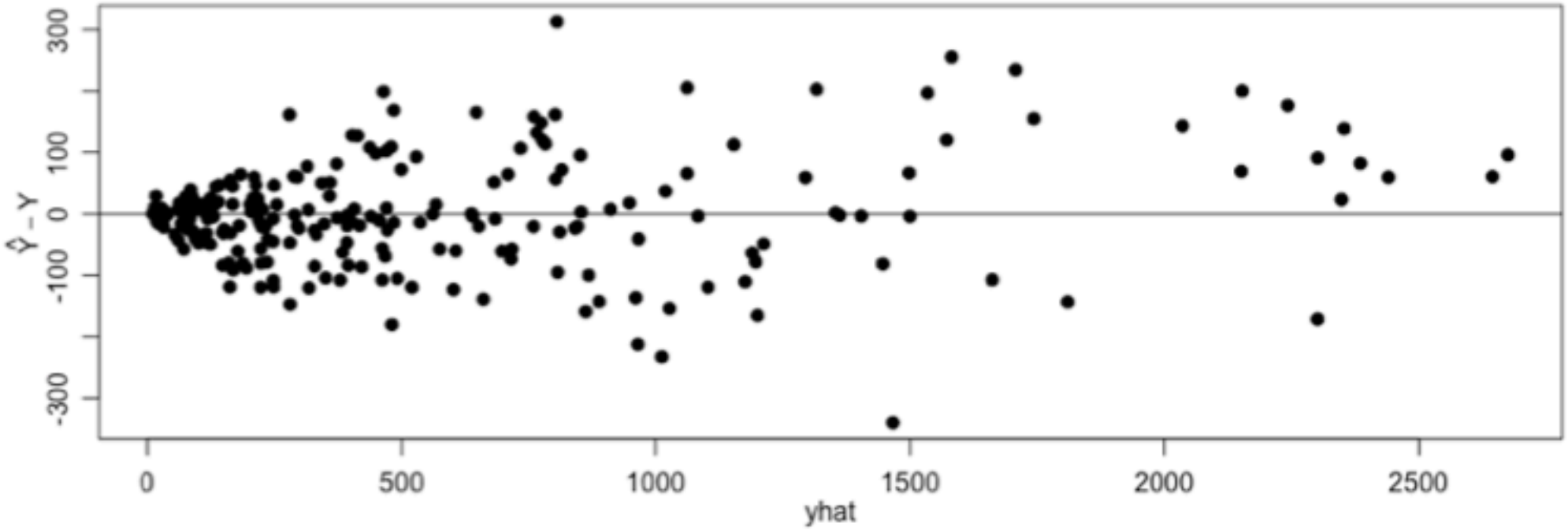
$$var[Y_i|\hat{Y}_i] = \hat{Y}_i = u_i\hat{\theta}_i$$

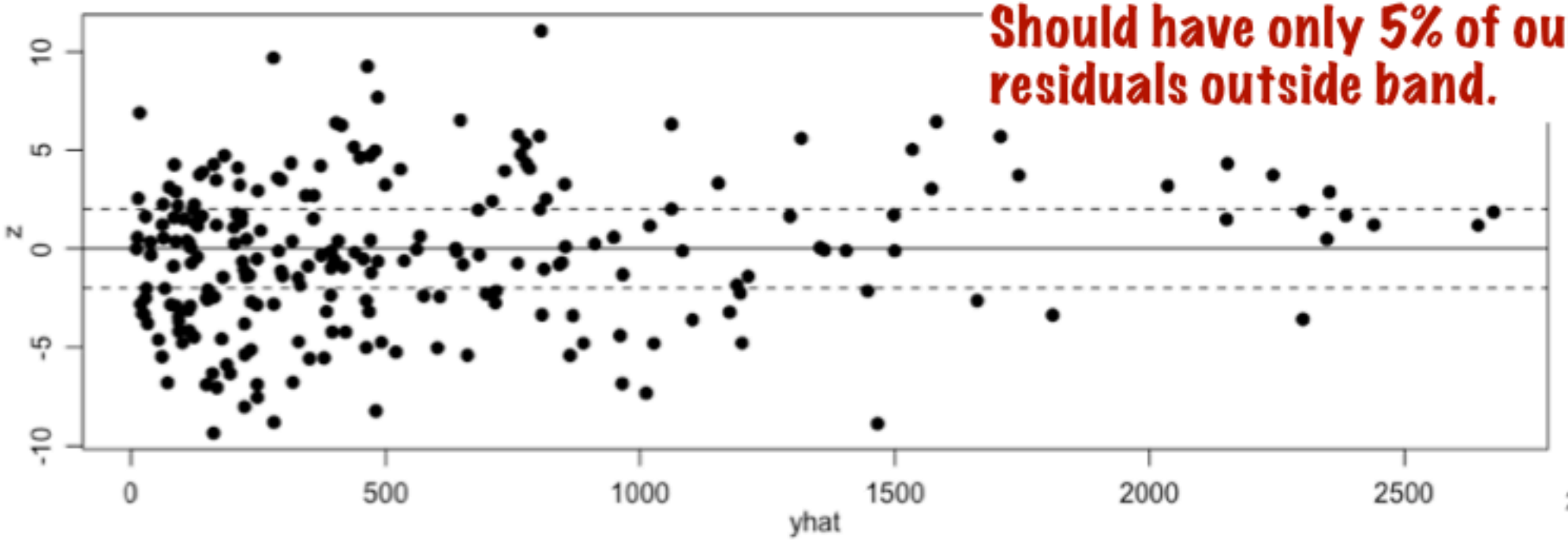
So our expected error will be different for different observations. We can align them with **Standardized Residuals:**

These should be all Normal(0, 1)!

$$z_i = \frac{y_i - \hat{y}_i}{sd(Y_i)} = \frac{y_i - u_i\hat{\theta}_i}{\sqrt{u_i\hat{\theta}_i}}$$

For police stops:
Raw residuals at top, standardized at bottom





Should have only 5% of our residuals outside band.

Over-dispersion

The bunnies are over-dispersed—too many of them are outside our expectations (the black circle).

This can mess up our estimation since there is more variability than expected.

But since we are dealing with bunnies, we have to account for this variability —it is part of the data.

All these bunnies are not where we expect

We expect some bunnies outside our region, but not too many.

Too many means overdispersion.

Our region where we expect bunnies

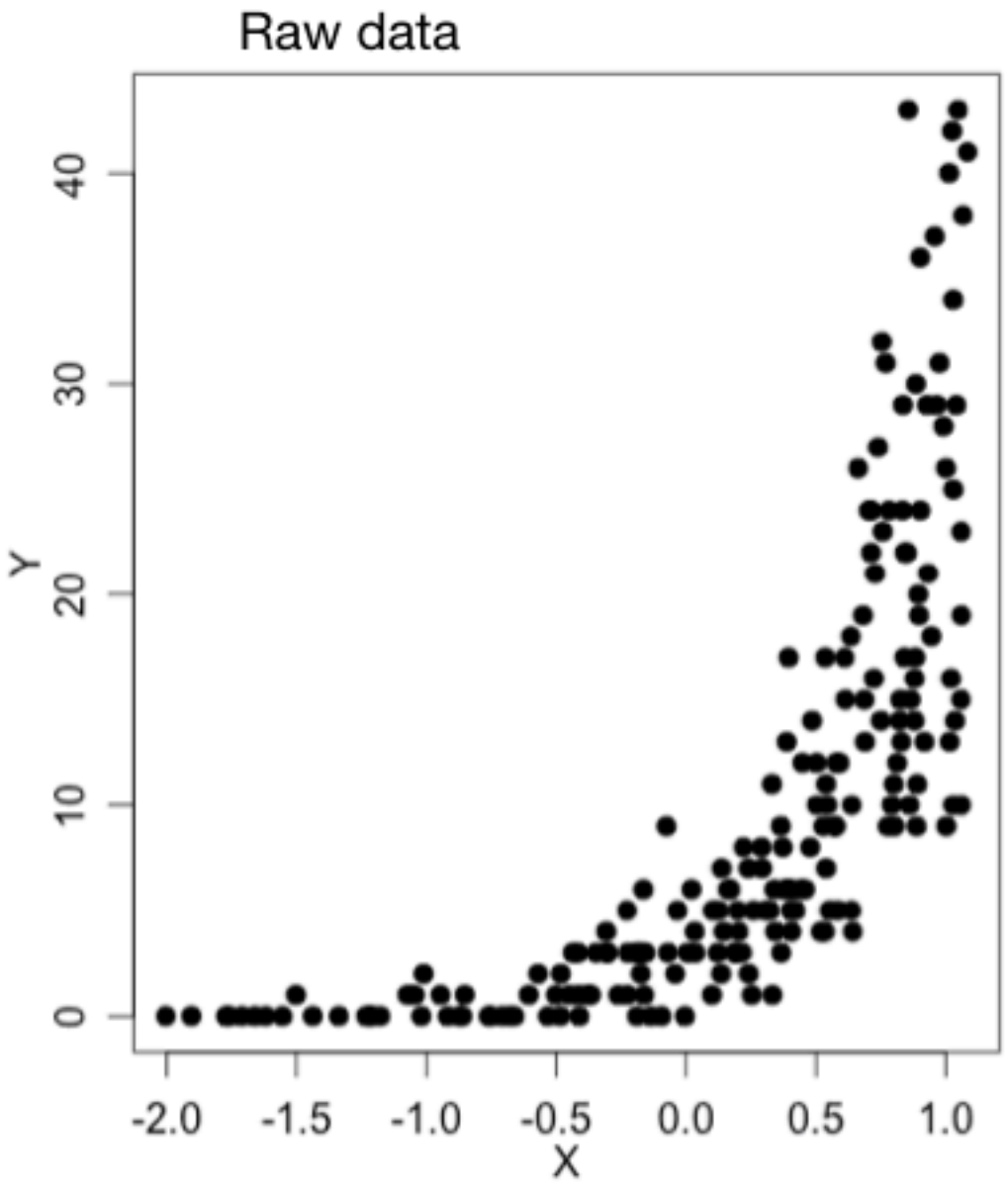
Exploring Overdispersion with a Quick Simulation Example

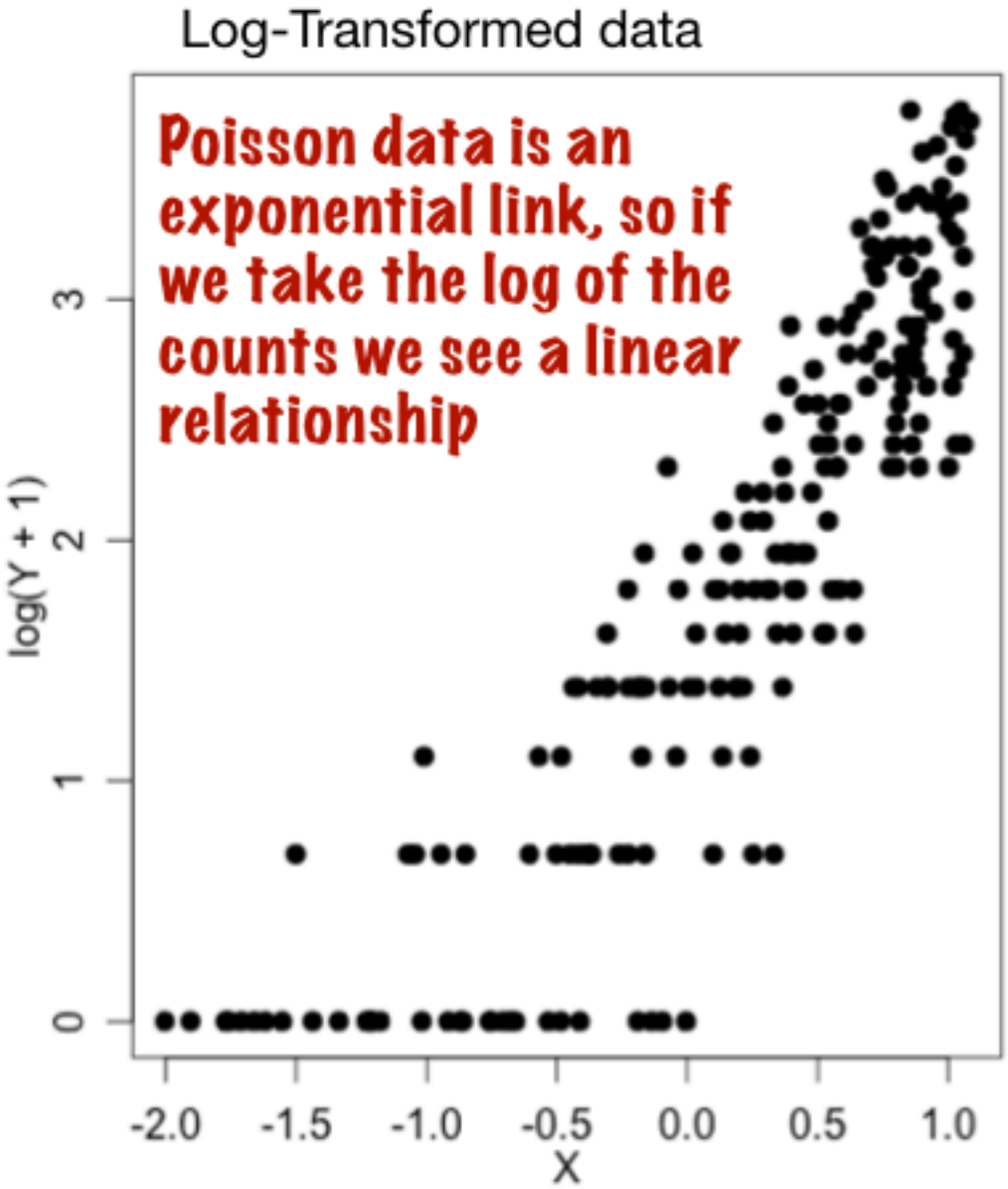
```
N = 200 # sample size

# Our covariate.
# Log makes the covariate linearly related to E[Y]
X = log( runif( N, 0.1, 3 ) )

# How much each unit gets "exposed"
exposure = runif( N, 0.5, 2 )

# our rate
rate = 1 + 2 * X

# our outcome
Y = rpois( N, lambda = exposure * exp( rate ) )

qplot( X, Y )
```

runif() - Make random numbers from 0.1 to 3, uniformly distributed

$$Y_i \sim Poisson(u_i \theta_i)$$

Our raw simulated data





Looking at our residuals

```
> fit.1 <- glm ( Y ~ X, family=poisson,
                 offset=log(exposure), data=df )

> display( fit.1 )

            coef.est coef.se
(Intercept) 1.04     0.05
X           1.97     0.06
—
  n = 200, k = 2
  residual deviance = 185.2, null deviance = 1835.2 (diff
>
> df$Y.hat = predict( fit.1, type = "response" )
> head( df$Y.hat )
[1] 13.4367010 23.1112149  4.8488740 25.5756308  0.575266

> # Make standardized z-scores
> z = with( df, (Y - Y.hat) / sqrt( Y.hat ) )
```

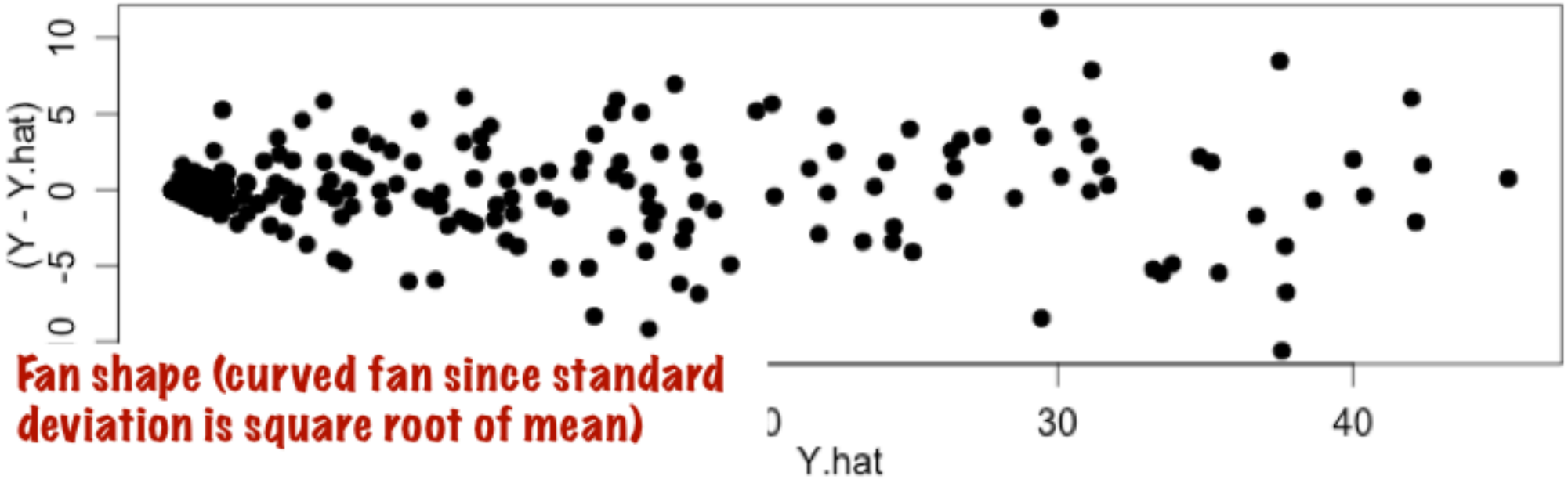
We fit the correct model. We can then see what our residuals look like when we are doing things right.

Predictions are not integers!

We are dividing by estimated variance under Poisson assumption

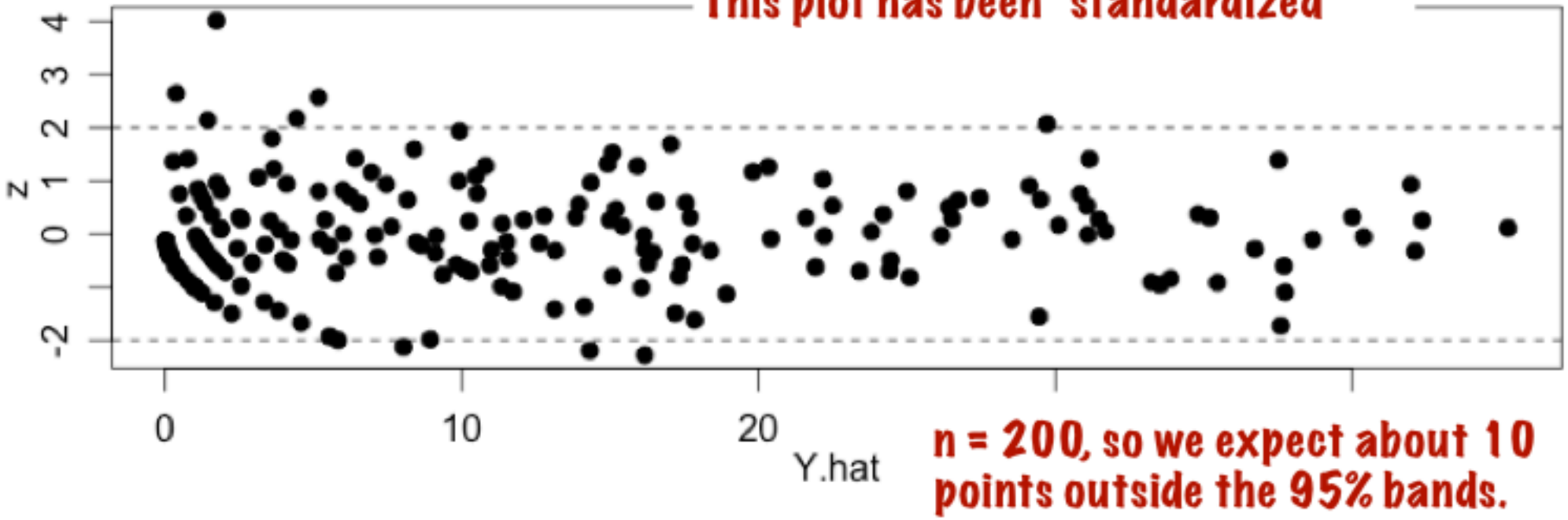
From Simulated Data

Raw residuals at top, standardized at bottom



Fan shape (curved fan since standard deviation is square root of mean)

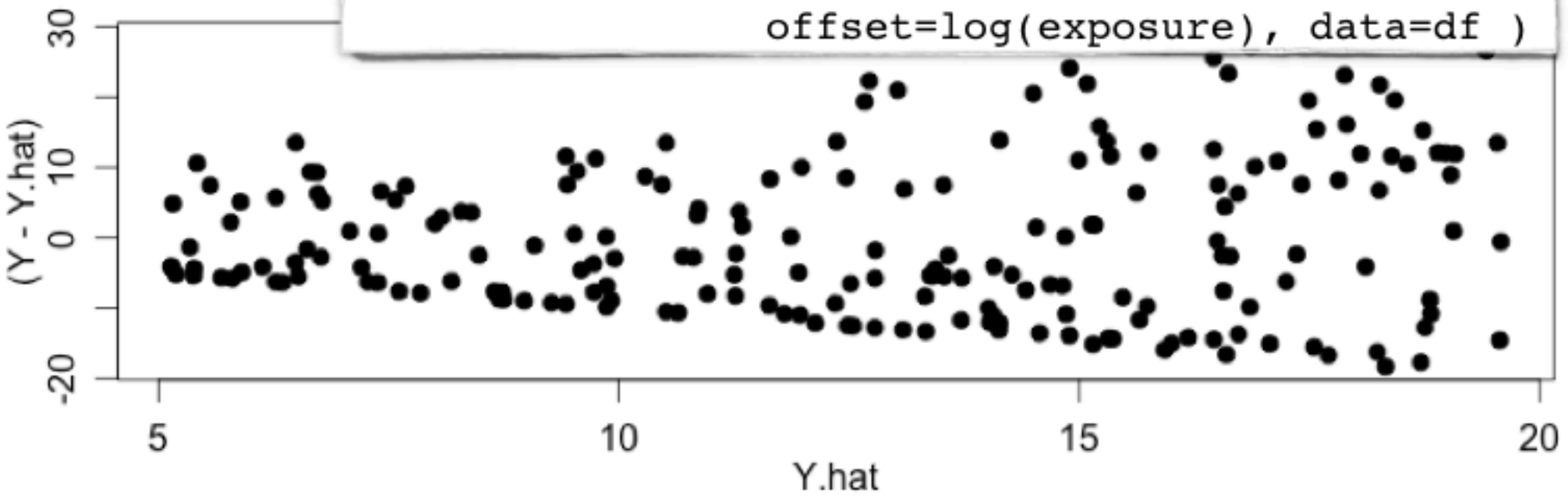
This plot has been "standardized"



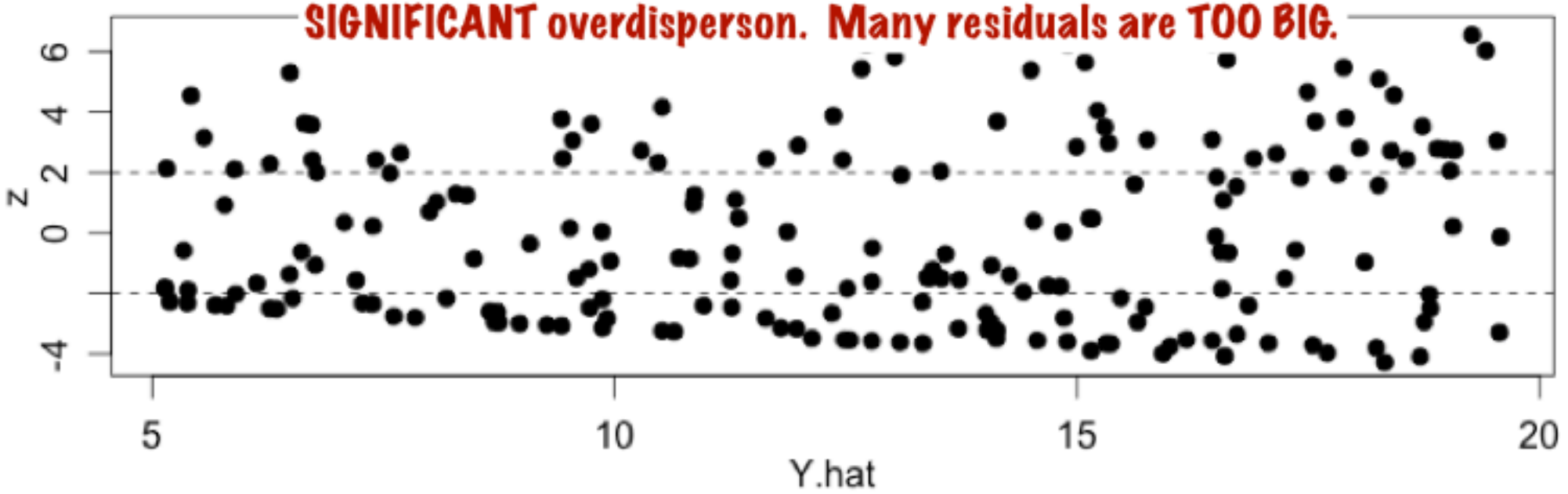
n = 200, so we expect about 10 points outside the 95% bands.

Residuals after Fitting our fake data without X

```
> fit.3 <- glm ( Y ~ 1, family=poisson,
                 offset=log(exposure), data=df )
```



SIGNIFICANT overdisperson. Many residuals are TOO BIG.



What is Overdisperson?

- ★ The variance in our data might not align with the variance in the model if our model is misspecified.
- ★ If the observations are spread out more than what the rates would suggest, we are **over dispersed.**
- ★ As we saw, this can happen if we are missing a covariate that predicts the rate.
- ★ We can see this by examining our standardized residuals. We ask

Are you Standard Normal?

Examining overdispersion by hand in our Police Stops example

```
> yhat = predict( fit.3, type="response" )
> head( yhat )
        1         2         3         4         5
246.82622  75.06056  63.11321 163.39124 122.09941  61.50
```

```
> z = ( stops$stops - yhat ) / sqrt( yhat )
```

```
> n = nrow(stops)
> k = 75 + 3 - 1
> D = sum( z^2 )
> D
[1] 3238.988
```

D is the sum of the squared standardized residuals (so a bunch of standard normals added up).

This is our estimated standard deviation. It should be one!

```
> sqrt( D / ( n - k ) )
[1] 4.678146
```

We can do a hypothesis test for overdispersion. D should be chi squared with n-k degrees of freedom.

```
> pchisq( D, n-k, lower.tail = FALSE )
[1] 0
```

P value of 0? Strong evidence of overdispersion.

Why overdispersion is bad

The regression **standard errors of our fixed effects** will be off by about the square root of the overdispersion ratio

In our case this is about 4.7!!!

All inferences (e.g., calculations of *t*-statistics) will be wildly off.

It is NOT GOOD to have your standard error be 4.7 times too big.

How you can get overdispersion

Easiest way:

Missing a predictor variable of the rate

In this case, your **actual rate varies for each unit beyond what you are modeling**. You have extra variance, which means more dispersion.

Another way:

Measurement error in exposure or predictor

Again, you are unable to predict the real mean of the observations well, and this uncertainty translates to more dispersion.

See simulation study script for example of both of these ways.

The overdispersed Poisson (almost)Model

$$y_i \sim \text{overdispersed Pois}(u_i \exp(X_i \beta), \omega)$$

ω "Omega" is our overdispersion parameter.

This is not yet a fully specified model.

We need to specify the *link* between our outcome and the parameters and predictors.

One link often used is the

negative-binomial distribution

Warning: negative-binomial model use
different parameters a, b
with a mean of a/b and overdispersion of 1 + 1/b

Fitting a Poisson model with overdispersion

quasipoisson is another choice.
Just think of it as a Poisson.

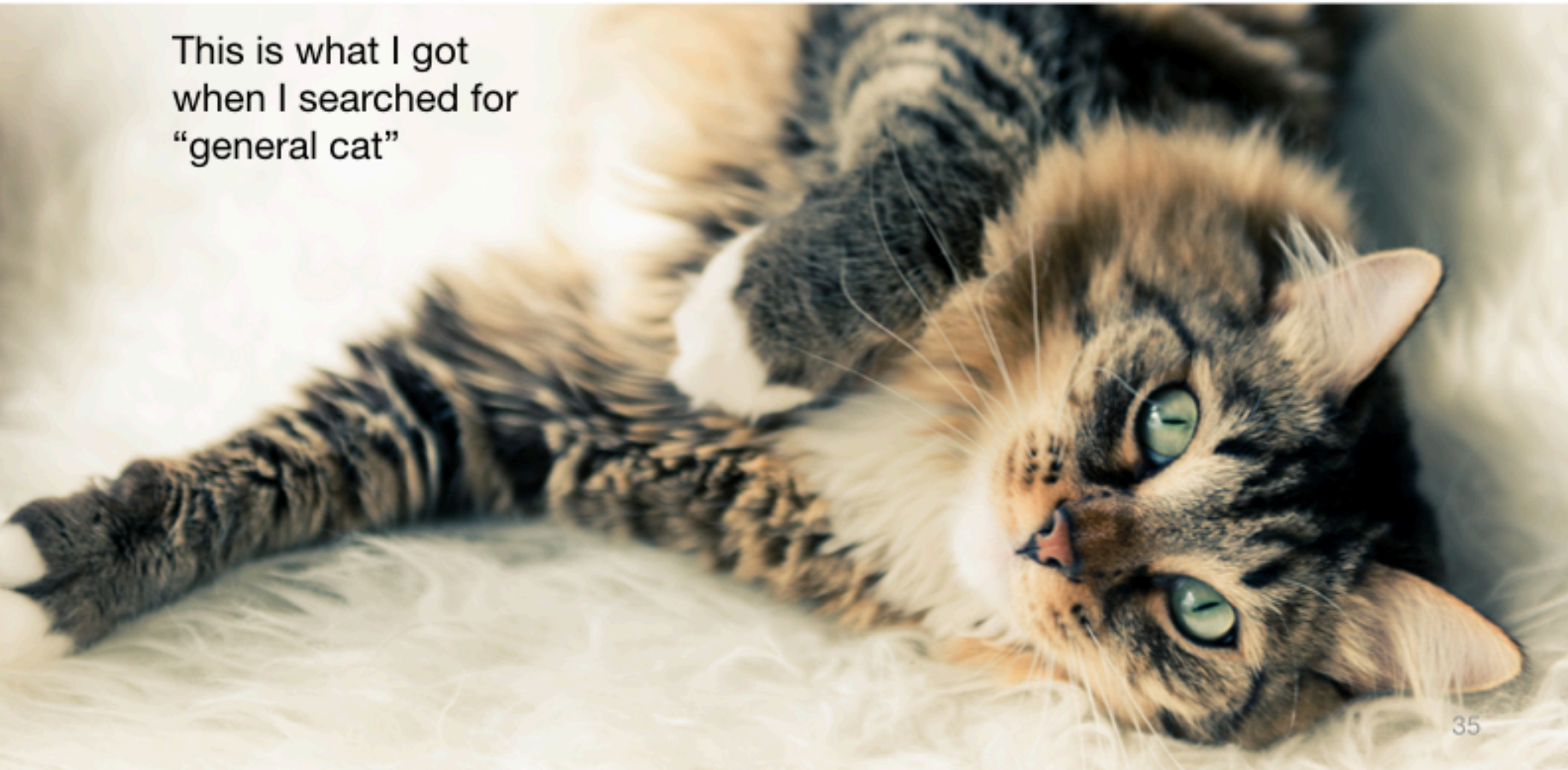
```
> fit.4 <- glm (stops ~ factor(eth) + factor(precinct),
          family=quasipoisson,
          offset=log(past.arrests), data=stops )
> display(fit.4)

                    coef.est coef.se
(Intercept)          -1.38     0.24
factor(eth)2          0.01     0.03
factor(eth)3         -0.42     0.04
factor(precinct)2    -0.15     0.35
factor(precinct)3     0.56     0.27
...
factor(precinct)74    1.15     0.27
factor(precinct)75    1.57     0.35
---
  n = 225, k = 77
  residual deviance = 3427.1, null deviance = 46120.3
(difference = 42693.1)
  overdispersion parameter = 21.9
```

We can believe these standard errors now.

(And they are substantially larger.)

Generalized Linear Models (A framework for a wide variety of special regression models)

G&H Chapter 6. Also see RH&S 10.1-10.5 or
G&H Chapter 5 on Logistic Models (a special case)



Anatomy of a Generalized Linear Model

Many models have the following canonical form:

- ★ Data vector (outcomes) $y_1, \ldots, y_n$
- ★ A n x p predictor matrix *X*, usually including an intercept.
- ★ A *linear predictor* $X_i\beta$
- ★ A *link function g* giving $\hat{y}_i = g^{-1}(X_i\beta)$
  - These are predicted (expected) *y* given the *linear predictor.*
- ★ We expect the actual *y* to vary about these predictions with a specific data distribution $p(y_i|\hat{y}_i)$
- ★ Possibly other parameters (variances, overdispersions, cutpoints)

Example 1: Linear Regression

Link Function $\hat{y}_i = g^{-1}(X_i\beta) = X_i\beta$ ← Our linear predictor

Data Distribution $p(y_i|\hat{y}_i) = N(\hat{y}_i, \sigma_y^2)$

Comments:

The link is the *identity* link (or no real link at all).

$$g(x) = x$$

We have an extra variance term for the data distribution (the classic σ^2).

Example 2: Poisson Regression

Link Function $\hat{y}_i = g^{-1}(X_i\beta) = \exp(X_i\beta)$

Data Distribution $p(y_i|\hat{y}_i) = Poisson(\hat{y}_i)$

Comments:

The exponent makes everything multiplicative in interpretation.

It also makes all predictions non-negative.

We can think of the exposure as a predictor (as an offset)

Example 3: Logistic Regression

Link Function

$$\hat{y}_i = g^{-1}(X_i\beta) = logit^{-1}(X_i\beta) = \frac{\exp(X_i\beta)}{1+\exp(X_i\beta)}$$

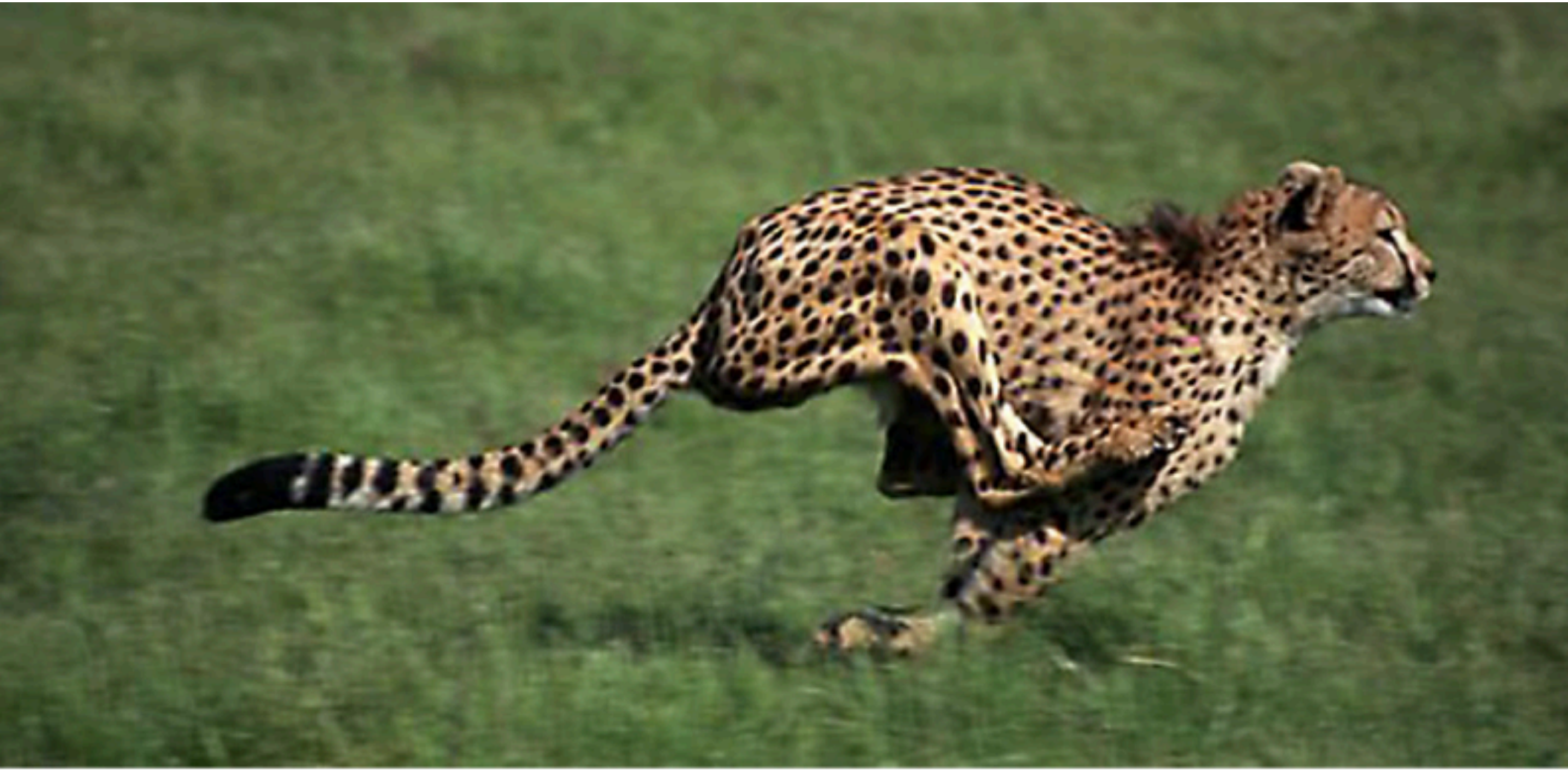
Data Distribution

$$p(y_i|\hat{y}_i) = Bernoulli(\hat{y}_i)$$

Comments:

The y-hat are the predicted y, which is the same as the *probability* of getting a success.

(So you don't get 0s and 1s from your predictions)



A fast recap

Summary

Generalized regression allows for different structures between outcome and covariates

It also allows for different types of outcome, so we can better model specific aspects of data.

For example, we can model count with an exponential relationship between covariates and rate.

Some of these models require the tweak of allowing for **over dispersion**, to account for measurement error and missing predictors in our model.

Appendix
Another logistic model: the Logistic-binomial model

Reading: see R&B examples

Also see G&H 6.3

A version of the logistic model

The logistic-binomial is a form of logistic regression.

You have **units**, and each unit has some fixed number of **trials** (e.g., number of coins to flip).

You are counting the number of heads.

The chance of each coin in a unit has the same chance p_i

Death Sentences Overturned

Data cover 34 states across 23 years (1973-1995)

Units are the 450 state-years (for those state-years with death penalties)

We then have

$$y_i \sim Binomial(n_i, p_i)$$

$$p_i = logit^{-1}(X_i\beta)$$

Each state-year is modeled as n_i coin flips with each coin having chance p_i of heads.

(n are cases, heads is overturned)

Overdispersion for Binomial Logistic

A binomial has variance dictated by the proportion

$$var[y_i] = np_i(1 - p_i)$$

We can then check if all the *y* are about 1 standard deviation from their respective y-hats (just like with Poisson earlier).

We again standardize

$$z_i = \frac{y_i - n_i\hat{p}_i}{\sqrt{n_i\hat{p}_i(1 - \hat{p}_i)}}$$

In practice, overdispersion happens all the time. (Flips are not the same or are not fully independent, is the usual explanation.)

Two forms of logistic regression?

Binary data is special case of count data. We just have $n_i = 1$ for each case!

However, no overdispersion is possible in this individual success-failure model.

Conversely, we could model count data as binary data if we replicated each case i n_i times.

This expansion would underscore the assumption of independence of the individual trials within a larger case.

Over-dispersion can take within-unit dependence into account, so is probably a better choice.