Poisson regression (Chapter 14.13)

Department of Statistics, University of South Carolina

Stat 705: Data Analysis II

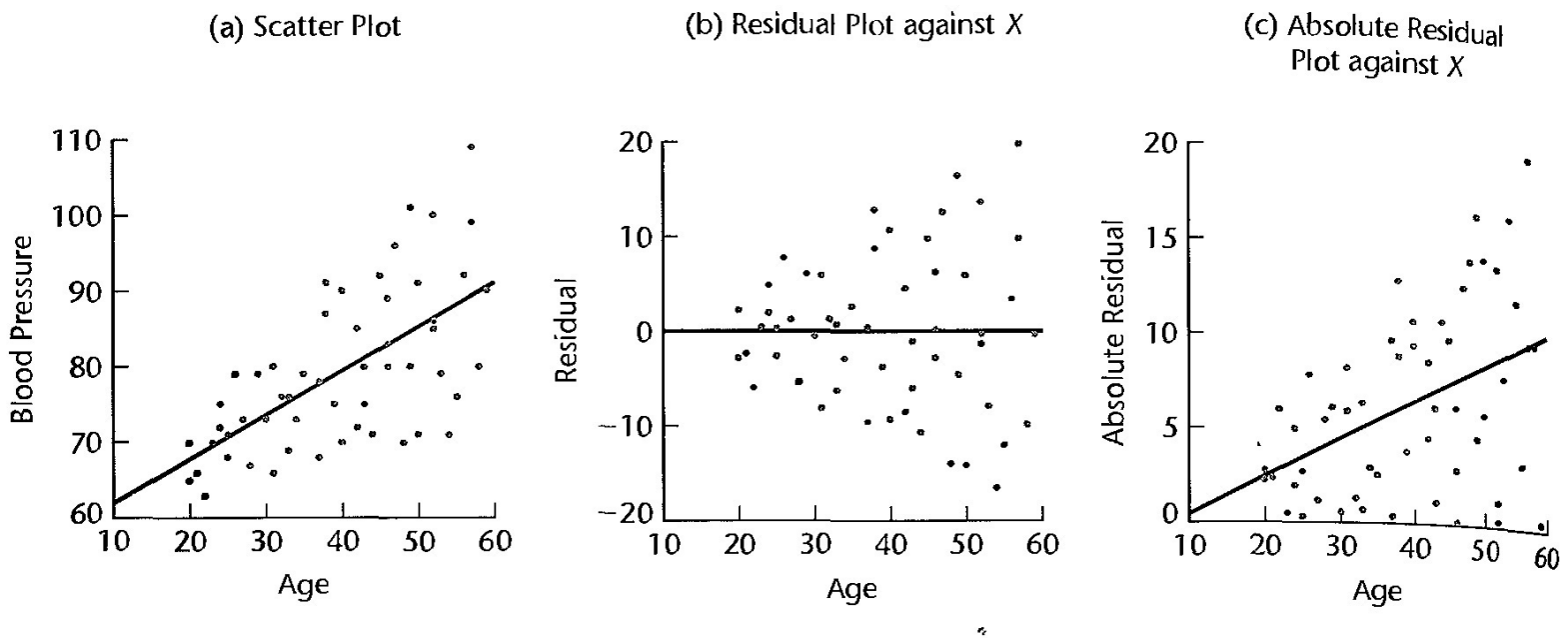
Outline

- Poisson Regression Model
- Interpretation of Regression Coefficients
- Modeling Rates
- Over-dispersion

Poisson regression

- Regular regression data $\{(\mathbf{x}_i, Y_i)\}_{i=1}^n$, but now Y_i is a positive integer, often a count: new cancer cases in a year, number of monkeys killed, etc.
- For Poisson data, $\text{var}(Y_i) = E(Y_i)$; variability increases with predicted values. In regular OLS regression, this manifests itself in the "megaphone shape" for r_i versus $\hat{Y}_i$.
- If you see this shape, consider whether the data could be Poisson. (Blood pressure data p.428)
- Any count, or positive integer could potentially be approximately Poisson. In fact, binomial data where n_i is really large, is approximately Poisson.

Blood Pressure Data: Megaphone Shape



Log and identity links

Let $Y_i \sim \text{Pois}(\mu_i)$.
The **log-link** relating μ_i to $\mathbf{x}_i'\boldsymbol{\beta}$ is used most often:

$$Y_i \sim \text{Pois}(\mu_i), \quad \log \mu_i = \beta_0 + x_{i1}\beta_1 + \cdots + x_{i,p-1}\beta_{p-1},$$

yielding what is commonly called the **Poisson regression** model.

The **identity** link can also be used

$$Y_i \sim \text{Pois}(\mu_i), \quad \mu_i = \beta_0 + x_{i1}\beta_1 + \cdots + x_{i,p-1}\beta_{p-1}.$$

Both can be fit in PROC GENMOD.

Interpretation for log-link

The log link $\log(\mu_i) = \mathbf{x}_i'\boldsymbol{\beta}$ is most common:

$$Y_i \sim \text{Pois}(\mu_i), \quad \mu_i = e^{\beta_0+\beta_1 x_{i1}+\cdots+\beta_k x_{ik}},$$

or simply $Y_i \sim \text{Pois}\left(e^{\beta_0+\beta_1 x_{i1}+\cdots+\beta_k x_{ik}}\right)$.

Say we have $k = 3$ predictors. The mean satisfies

$$\mu(x_1, x_2, x_3) = e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\beta_3 x_3}.$$

Then increasing x_2 to $x_2 + 1$ gives

$$\mu(x_1, x_2 + 1, x_3) = e^{\beta_0+\beta_1 x_1+\beta_2(x_2+1)+\beta_3 x_3} = \mu(x_1, x_2, x_3)e^{\beta_2}.$$

In general, increasing x_j by one, but holding the other predictors the constant, increases the mean by a factor of e^{β_j}.

Example: Crab mating

Data on female horseshoe crabs.

- C = color (1,2,3,4=light medium, medium, dark medium, dark).
- S = spine condition (1,2,3=both good, one worn or broken, both worn or broken).
- W = carapace width (cm).
- Wt = weight (kg).
- Sa = number of satellites (additional male crabs besides her nest-mate husband) nearby.

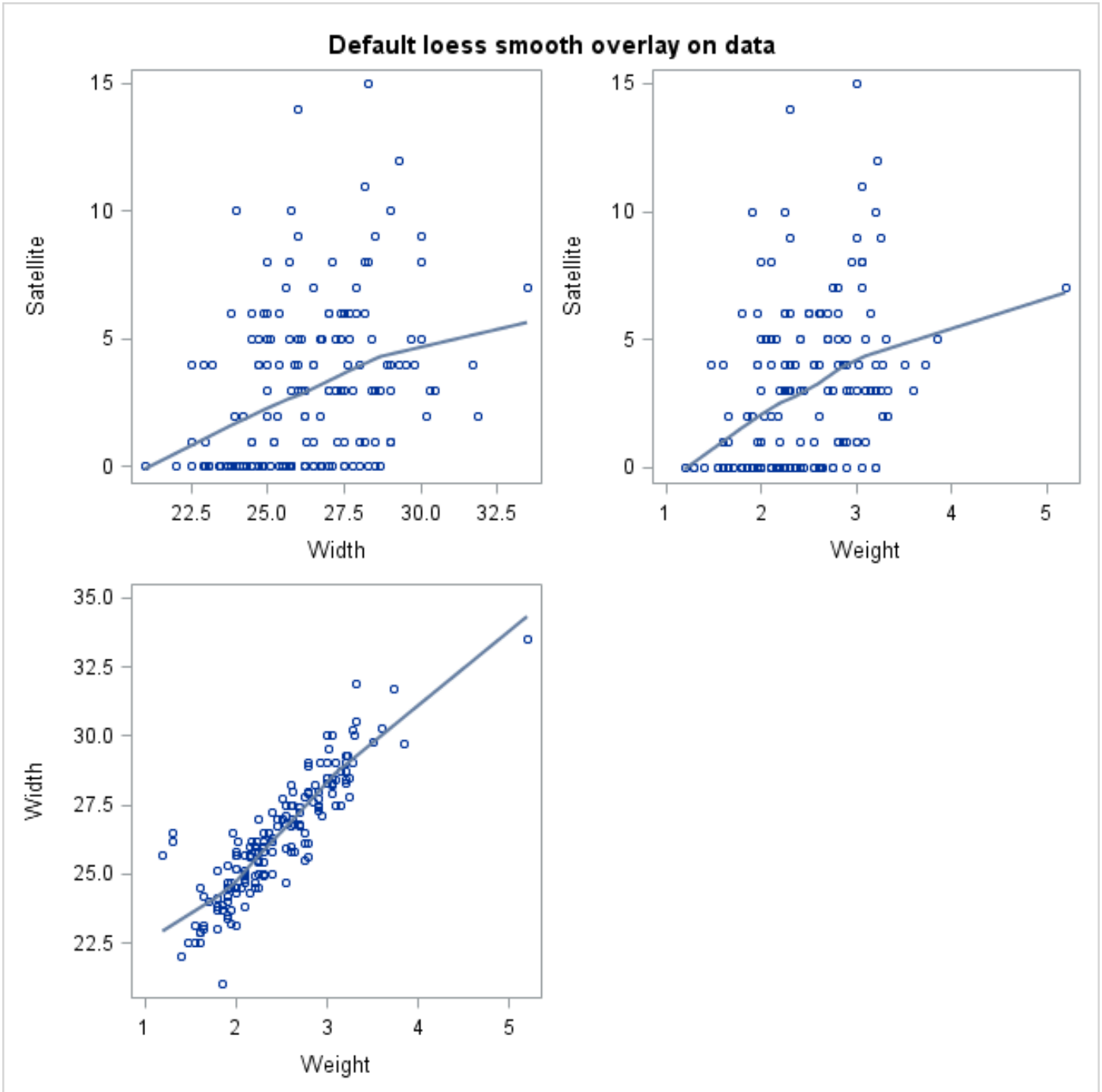
Using logistic regression we explored whether a female had *one or more* satellites. Using Poisson regression we can model the actual *number* of satellites directly.

Looking at the data...

We initially examine width as a predictor for the number of satellites. A raw scatterplot of the numbers of satellites versus the predictors does not tell us much. Superimposing a smoothed fit helps & shows an approximately linear trend in weight.

Note that variability increases with width and weight!

```
options nodate;
proc sgscatter data=crabs;
  title "Default loess smooth on top of data";
  plot satell*(width weight) width*weight / loess;
```


Default loess smooth overlay on data
Satellite
Width
Satellite
Weight
Width
Weight

Three competing models using width as predictor

We'll fit three models using `proc genmod`.

$$Sa_i \sim \text{Pois}(e^{\beta_0+\beta_1 W_i}),$$

$$Sa_i \sim \text{Pois}(\beta_0 + \beta_1 W_i),$$

and

$$Sa_i \sim \text{Pois}(e^{\beta_0+\beta_1 W_i+\beta_2 W_i^2}).$$

SAS code

SAS code:

```
data crabs; input color spine width satellite
weight;
   weight=weight/1000; color=color-1;
   width_sq=width*width;
datalines;
3  3  28.3  8  3050
4  3  22.5  0  1550
...et cetera...
5  3  27.0  0  2625
3  2  24.5  0  2000
;

*Problems with residuals seen here apply to other models as well;
proc genmod data=crabs plots=all;
   model satellite = width / dist=poi link=log ;
proc genmod data=crabs;
   model satellite = width / dist=poi link=identity ;
proc genmod data=crabs;
   model satellite = width width_sq / dist=poi link=log ;
run;
```

R output

```
Call:
glm(formula = satell ~ weight, family = poisson(link = log),
    data = crabdata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9307  -1.9981  -0.5627   0.9298   4.9992

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.284e-01  1.789e-01  -2.394   0.0167 *
weight       5.893e-04  6.502e-05   9.064   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 632.79  on 172  degrees of freedom
Residual deviance: 560.87  on 171  degrees of freedom
AIC: 920.16

Number of Fisher Scoring iterations: 5
```

R output

```
Call:
glm(formula = satell ~ weight, family = poisson(link = identity),
    data = crabdata, start = c(-11, 0.55))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0483  -1.8455  -0.6009   0.8750   4.9259

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.5982483  0.3885754  -6.687 2.28e-11 ***
weight       0.0022638  0.0001825  12.402  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 632.79  on 172  degrees of freedom
Residual deviance: 543.40  on 171  degrees of freedom
AIC: 902.7

Number of Fisher Scoring iterations: 9
```

R output

```
 Call:
glm(formula = satell ~ weight + wsq, family = poisson(link = "log"),
    data = crabdata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0743  -1.8574  -0.6145   0.9158   4.9796

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.283e+00  5.707e-01  -4.000 6.32e-05 ***
weight       1.912e-03  3.955e-04   4.834 1.34e-06 ***
wsq         -2.229e-07  6.773e-08  -3.291 0.000998 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 632.79  on 172  degrees of freedom
Residual deviance: 547.01  on 170  degrees of freedom
AIC: 908.31

Number of Fisher Scoring iterations: 6
```

Inference

- Write down the fitted equation for the Poisson mean from each model.
- How are the regression effects interpreted in each case?
- How would you pick among models? Recall

$$\text{AIC} = -2[L(\hat{\boldsymbol{\beta}}; \mathbf{y}) - p].$$

For log-link quadratic, identity-link linear, and log-link linear we have

$$\begin{aligned} -2(-458.082 - 2) &= 920.16, \\ -2(-449.349 - 2) &= 902.70, \\ -2(-451.16 - 3) &= 908.31. \end{aligned}$$

- Are there any potential problems with any of the models? How about prediction? Are there any potential problems common to *all* these models? We will return to this topic later.

Interpretation

```
Call:
glm(formula = satell ~ weight, family = poisson(link = log),
    data = crabdata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9307  -1.9981  -0.5627   0.9298   4.9992

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.284e-01  1.789e-01  -2.394   0.0167 *
weight       5.893e-04  6.502e-05   9.064   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 632.79  on 172  degrees of freedom
Residual deviance: 560.87  on 171  degrees of freedom
AIC: 920.16

Number of Fisher Scoring iterations: 5
```

Interpretation

```
Call:
glm(formula = satell ~ weight, family = poisson(link = log),
    data = crabdata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9307  -1.9981  -0.5627   0.9298   4.9992

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.284e-01  1.789e-01  -2.394   0.0167 *
weight       5.893e-04  6.502e-05   9.064   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 632.79  on 172  degrees of freedom
Residual deviance: 560.87  on 171  degrees of freedom
AIC: 920.16

Number of Fisher Scoring iterations: 5
```

Since $\widehat{\beta} > 0$, the wider the female crab, the greater expected number of male satellites on the multiplicative order as $e^{0.164} = 1.18$. More specifically, for one unit increase in the width, the number of Sa will increase and it will be multiplied by 1.18.

GLM Model for Rates

- Sometimes counts are collected over different intervals or areas of time, space...
- For example, we may have numbers of new cancer cases per *month* from some counties, and per *year* from others.
- If time periods are the same for all data, then μ_i is the mean count per time period.
- Otherwise we specify μ_i as a rate per unit time period and have data in the form $\{(\mathbf{x}_i, Y_i, t_i)\}_{i=1}^n$ where t_i is the amount of time that the Y_i accumulates over.

$$\begin{aligned} \log(\mu/t) &= \alpha + \beta x \\ \log\mu - \log(t) &= \alpha + \beta x \\ \log(\mu) &= \alpha + \beta x + \log(t) \end{aligned}$$

Offsets

$$g(\mu) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k + \log(t_i)$$

- Random component: Response Y has a Poisson distribution and t is index of the time or space. The expected value of rate Y/t is $E(Y/t) = \mu/t$
- Systematic component: linear predictor
- Link function: log(Y/t)
- Model: $Y_i \sim \text{Pois}(t_i \mu_i)$.
- For the log-link we have

$$Y_i \sim \text{Pois}\left(e^{\mathbf{x}_i'\boldsymbol{\beta} + \log(t_i)}\right).$$

$\log(t_i)$ is called an *offset*–a covariate with fixed coefficient 1.

Ache monkey hunting

Data on the number of capuchin monkeys killed by $n = 47$ Ache hunters over several hunting trips were recorded; there were 363 total records. The hunting process involves splitting into groups,

chasing monkeys through the trees, and shooting arrows straight up. Let Y_i be the total number of monkeys killed by hunter i of

age a_i $(i = 1, \ldots, 47)$ over several hunting trips lasting different amounts of days; total number of days is t_i. Let μ_i be hunter i's kill rate (per day).

$$Y_i \sim \text{Pois}(\mu_i t_i),$$

where

$$\log \mu_i = \beta_0 + \beta_1 a_i + \beta_2 a_i^2.$$

A quadratic effect is included to accommodate a "leveling off" effect or possible decline in ability with age. Of interest is when hunting ability is greatest; hunting prowess contributes to a man's status within the group.

Aiming for...

...dinner!

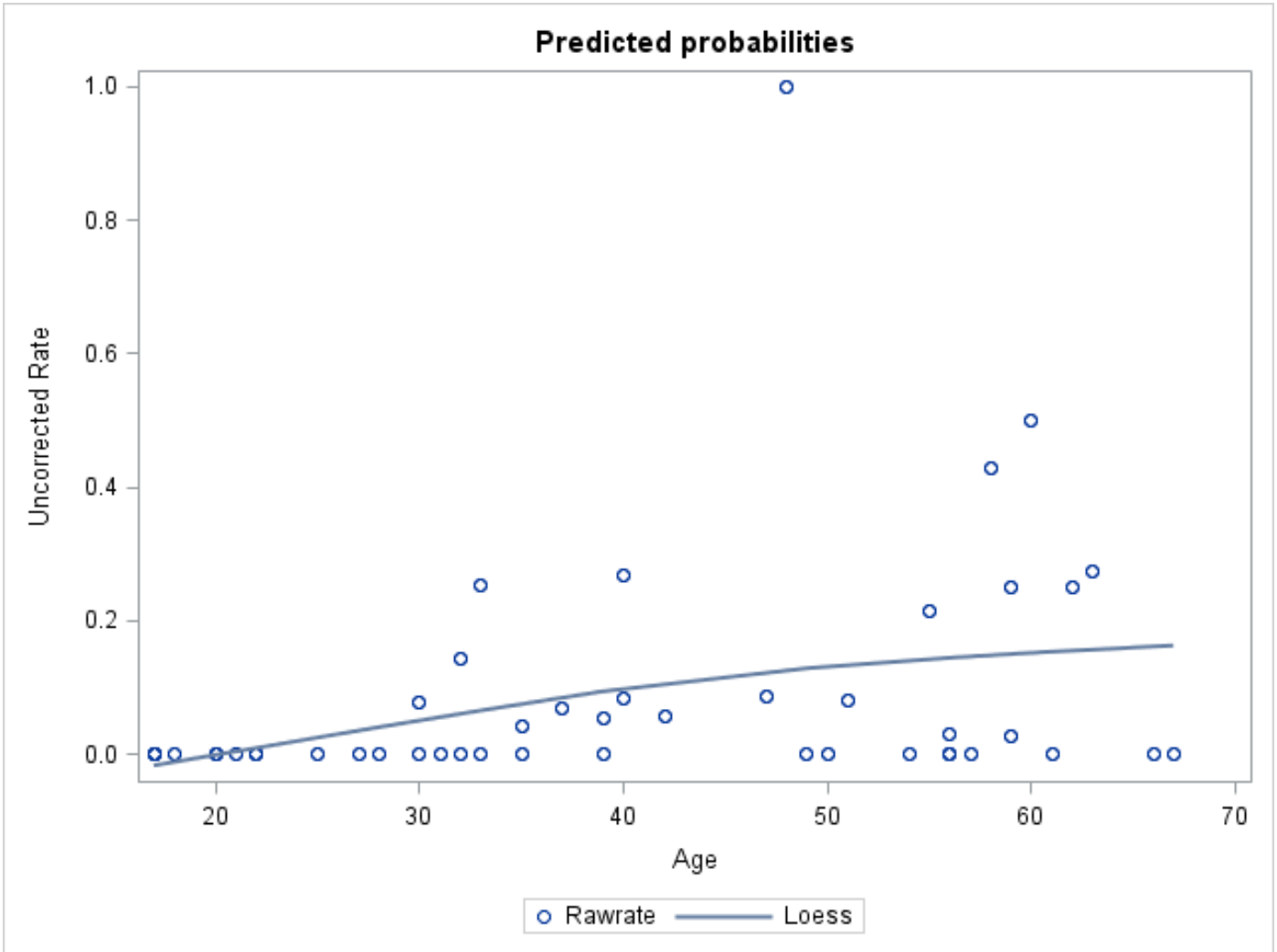
SAS code

```
data ache; input age kills days @@; logdays=log(days); rawrate=kills/days;
datalines;
67       0        3 66       0       89 63      29      106 60       2        4
61       0       28 59       2       73 58       3        7 57       0       13
56       0        4 56       3      104 55      27      126 54       0       63
51       7       88 50       0        7 48       3        3 49       0       56
47       6       70 42       1       18 39       0        4 40       7       83
40       4       15 39       1       19 37       2       29 35       2       48
35       0       35 33       0       10 33      19       75 32       9       63
32       0       16 31       0       13 30       0       20 30       2       26
28       0        4 27       0       13 25       0       10 22       0       16
22       0       33 21       0        7 20       0       33 18       0        8
17       0        3 17       0       13 17       0        3 56       0       62
62       1        4 59       1        4 20       0       11
;
proc sgscatter data=ache; * not weighted by how many days...;
 plot rawrate*age / loess;

proc genmod data=ache;
 model kills=age age*age / dist=poisson link=log offset=logdays;
 output out=out p=p reschi=r;

proc sgscatter data=out;
 plot r*(p age) / loess; run;
```

Raw rates with loess smooth



SAS output

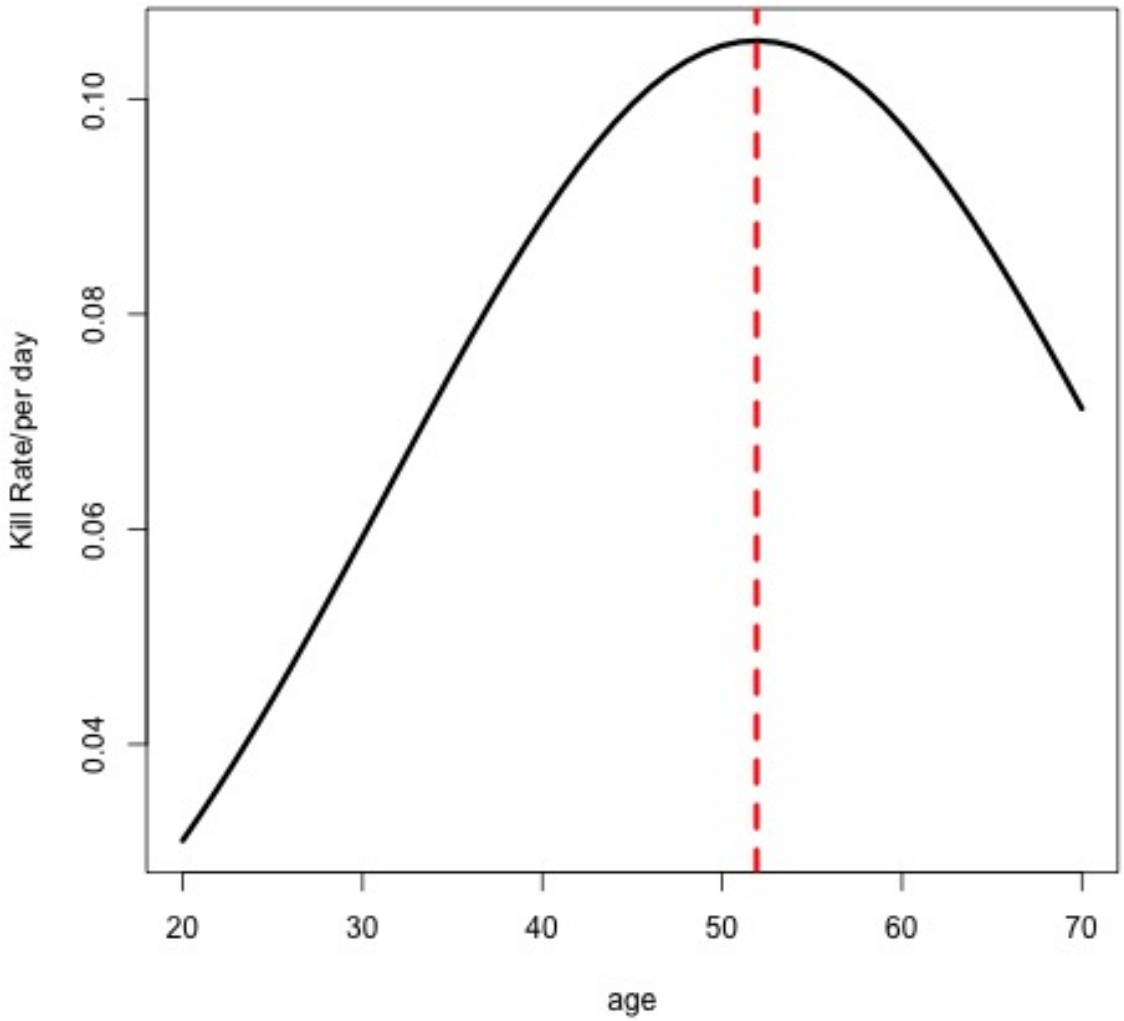
Analysis Of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept	1	-5.4842	1.2448	-7.9240	-3.0445	19.41	<.0001
age	1	0.1246	0.0568	0.0134	0.2359	4.82	0.0281
age*age	1	-0.0012	0.0006	-0.0024	0.0000	3.78	0.0520
Scale	0	1.0000	0.0000	1.0000	1.0000		

The fitted *monkey kill rate* is

$$\mu(a) = \exp(-5.4842 + 0.1246a - 0.0012a^2).$$

At what age, typically, is monkey hunting ability maximized?


Kill Rate/per day
0.04
0.06
0.08
0.10
20
30
40
50
60
70
age

Goodness of fit

The Pearson residual is

$$r_{P_i} = \frac{Y_i - \hat{\mu}_i}{\sqrt{\hat{\mu}_i}}.$$

As in logistic regression, the sum of these gives the Pearson GOF statistic

$$X^2 = \sum_{i=1}^{n} r_{P_i}^2.$$

$X^2 \sim \chi^2_{n-p}$ when the regression model fits. The alternative is the "saturated model."

The deviance statistic is

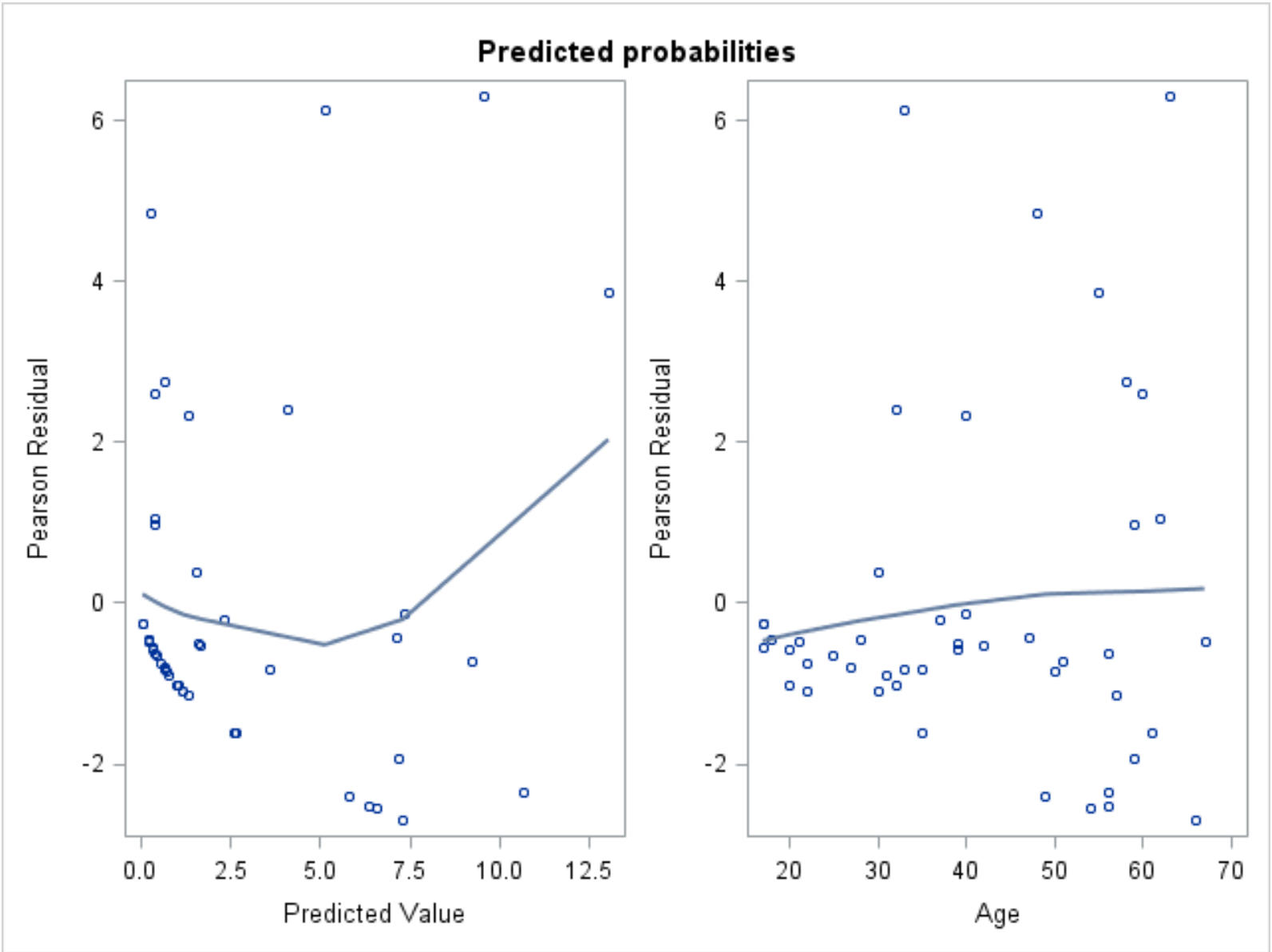
$$D^2 = -2 \sum_{i=1}^{n} \left[Y_i \log(\hat{\mu}_i / Y_i) + (Y_i - \hat{\mu}_i) \right].$$

Replace $\hat{\mu}_i$ by $\hat{\mu}_i t_i$ when offsets are present. $D^2 \sim \chi^2_{n-p}$ when the regression model fits. Page 621 defines "deviance residual" dev_i.

Diagnostics

- From SAS we can get Cook's distance c_i (cookd), leverage h_i (h), predicted $\hat{Y}_i = e^{\mathbf{x}_i'\hat{\boldsymbol{\beta}}}$ (p) Pearson residual r_{P_i} (reschi; variance < 1), studentized Pearson residual r_{SP_i} (stdreschi; variance $= 1$).
- Residual plots show some issue when counts Y_i close to zero.
- We can do smoothed versions; see SAS code for Ache hunting data.

The model doesn't fit very well; $\text{var}(r_{P_i}) \gg 1$...



Overdispersion–blocking

The variability in the Pearson residuals is much higher than what we should see; there are many poorly fit observations. This extra-Poisson variability is often referred to as "overdispersion."

Alternative model could be, *blocking* on individuals to reduce variability. The Ache hunters actually took part in many hunting trips, i.e. there are repeated measures on each hunter. We can instead consider hunting trip j from hunter i of length L_{ij} days, and posit a mixed model

$$Y_{ij} \sim \text{Pois}(\lambda_{ij} L_{ij}), \quad \log(\lambda_{ij}) = \beta_0 + \beta_1 a_i + \beta_2 a_i^2 + u_i,$$

where

$$u_1, \ldots, u_{47} \overset{iid}{\sim} N(0, \sigma^2)$$

are random *hunter ability* effects.

This model, fit in `proc glimmix`, reduces variability by appropriately blocking the repeated measures on hunter. We'll fit this model in our lecture on GLMM's.

Estimation of GLMs

Consider $Y_1, \ldots, Y_n$ be n independent samples from the exponential family distribution in canonical form,

$$f_{Y_i}(y_i; \theta_i) = \exp\{y_i b(\theta_i) + c(\theta_i) + d(y_i)\},$$

and let

$$\eta_i \equiv g(\mu_i) = \sum_{j=1}^{p} x_{ij}\beta_j,$$

where $\mu_i = \mathrm{E}[Y_i]$. The log-likelihood is

$$\begin{aligned} \log \mathcal{L}(\boldsymbol{\beta}; \mathbf{y}) &= \sum_{i=1}^{n} y_i b(\theta_i) + \sum_{i=1}^{n} c(\theta_i) + \sum_{i=1}^{n} d(y_i) \\ &\equiv \sum_{i=1}^{n} l_i \end{aligned}$$

GLM Estimation

The score function ($\frac{\partial}{\partial \beta_j} \log \mathcal{L}(\boldsymbol{\beta}; \mathbf{y})$) is (using the chain rule)

$$\begin{aligned} U_j \equiv \frac{\partial}{\partial \beta_j} \log \mathcal{L}(\boldsymbol{\beta}; \mathbf{y}) &= \sum_{i=1}^{n} \frac{\partial}{\partial \beta_j} l_i \\ &= \sum_{i=1}^{n} \frac{\partial \eta_i}{\partial \beta_j} \cdot \frac{\partial \mu_i}{\partial \eta_i} \cdot \frac{\partial \theta_i}{\partial \mu_i} \cdot \frac{\partial}{\partial \theta_i} l_i. \end{aligned}$$

After some calculation,...
for exponential families in canonical form, the score function is

$$U_j = \sum_{i=1}^{n} \left[\frac{y_i - \mu_i}{\mathrm{Var}[Y_i]} \cdot x_{ij} \cdot \frac{\partial \mu_i}{\partial \eta_i} \right],$$

Alternatively, it can be also written as:

$$U_j = \sum_{i=1}^{n} \left[\frac{y_i - \mu_i}{\mathrm{Var}[Y_i]} \cdot \frac{\partial \mu_i}{\partial \beta_j} \right], \qquad j = 1, \ldots, p.$$

GLM Estimation

Let H (Hessian) denotes the second derivatives of $\log \mathcal{L}$ with respect to $\boldsymbol{\beta}$ ($\frac{\partial^2}{\partial \beta_j \partial \beta_k} \log \mathcal{L}(\boldsymbol{\beta}; \mathbf{y})$). It can be shown that

$$\mathrm{E}[\mathbf{H}] = -\mathcal{J}.$$

For exponential families in canonical form, the Fisher information is:

$$\mathcal{J} = \mathbf{X}'\mathbf{W}\mathbf{X}$$

where $\mathbf{W}$ is the $n \times n$ diagonal matrix with

$$\mathbf{W}_{ii} = \frac{1}{\mathrm{Var}[Y_i]} \left(\frac{\partial \mu_i}{\partial \eta_i} \right)^2.$$

Poisson Regression Inference

Let $Y_i \sim \text{Poisson}(\mu_i)$ with $\log \mu_i = \sum_{j=1}^{p} x_{ij}\beta_j$, $i = 1, \ldots, n$ (all Y_i's independent). In this case,

$$\begin{aligned} U_j &= = \sum_{i=1}^{n} \frac{y_i - \exp\left\{\sum_{k=1}^{p} x_{ik}\beta_k\right\}}{\exp\left\{\sum_{k=1}^{p} x_{ik}\beta_k\right\}} \cdot x_{ij} \cdot \exp\left\{\sum_{k=1}^{p} x_{ik}\beta_k\right\} \\ U_j &= \sum_{i=1}^{n} \left(y_i - \exp\left\{\sum_{k=1}^{p} x_{ik}\beta_k\right\}\right) x_{ij} \end{aligned}$$

So, $\hat{\boldsymbol{\beta}}$ does not have a closed form.

Overdispersion & Quasi-Likelihood

If $Y \sim Poisson(\mu)$, then $E(Y) = \mu$ and $Var(Y) = \mu$. Almost always in count data, $Var(Y) \neq \mu$.
To account for dispersion we set $Var[Y_i] = \phi V_i(\mu_i)$.
Recall that the score function for a GLM is given by,

$$U_j = \sum_{i=1}^{n} \left[\frac{y_i - \mu_i}{\mathrm{Var}[Y_i]} \cdot \frac{\partial \mu_i}{\partial \beta_j} \right],$$

We "plug-in" $Var[Y_i] = \phi V_i(\mu_i)$ in the score function, and the **quasi-score** function can be obtained:

$$\mathbf{U}(\boldsymbol{\beta}; \mathbf{y}) = \mathbf{D}'\mathbf{V}^{-1}(\mathbf{y} - \boldsymbol{\mu})/\phi,$$

where $\mathbf{D}$ is the $n \times p$ matrix with $(i,j)^{th}$ entry $\partial \mu_i / \partial \beta_j$, $\mathbf{V}$ is the $n \times n$ diagonal matrix with i^{th} diagonal entry $V(\mu_i)$,
$\boldsymbol{y} = (y_1, \ldots, y_n)$, and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$.

Overdispersion & Quasi-Likelihood

In general, the quasi-score may not correspond to a true likelihood. Let

$$Q(\boldsymbol{\beta}; \mathbf{y}) = \sum_{i=1}^{n} \int_{y_i}^{\mu_i} \frac{y_i - t}{\phi V(t)} dt.$$

Note that

$$\frac{\partial}{\partial \boldsymbol{\beta}} Q(\boldsymbol{\beta}; \mathbf{y}) = \mathbf{U}(\boldsymbol{\beta}; \mathbf{y}).$$

For this reason, the function Q is called the *quasi-likelihood*, or, more accurately, the *log quasi-likelihood*. It turns out that Q behaves somewhat similarly to the log-likelihood function in a fully parametric setting.

ϕ can be estimated as $\phi = \frac{\chi^2}{n-p}$ (sum over Pearson residuals)

Using Quasi-Likelihood in R for Overdispersion

```
> fit1<-glm( Kills ~ Age + Agesq, offset=logd, data=dat, family="poisson")
> summary(fit1)

Call:
glm(formula = Kills ~ Age + Agesq, family = "poisson", data = dat,
    offset = logd)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7592  -0.9019  -0.6265  -0.3716   4.4430

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.4207759  1.2314767  -4.402 1.07e-05 ***
Age          0.1222860  0.0564406   2.167   0.0303 *
Agesq       -0.0011823  0.0006188  -1.911   0.0560 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 426.35  on 362  degrees of freedom
Residual deviance: 416.56  on 360  degrees of freedom
AIC: 595.09

Number of Fisher Scoring iterations: 6
```

Using Quasi-Likelihood in R for Overdispersion

```
> fit2<-glm( Kills ~ Age + Agesq, offset=logd, data=dat, family="quasipoisson")
> summary(fit2)

Call:
glm(formula = Kills ~ Age + Agesq, family = "quasipoisson", data = dat,
    offset = logd)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7592  -0.9019  -0.6265  -0.3716   4.4430

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.420776   1.697658  -3.193  0.00153 **
Age          0.122286   0.077807   1.572  0.11691
Agesq       -0.001182   0.000853  -1.386  0.16661
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for quasipoisson family taken to be 1.900413)

    Null deviance: 426.35  on 362  degrees of freedom
Residual deviance: 416.56  on 360  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 6
```

Overdispersion–Negative binomial regression

Another simple solution is to assume a distribution for which the variance can be larger than the mean, e.g., the negative binomial. SAS uses the following parameterization for the dispersion parameter k, which yields $V(Y) = \mu + k\mu^2$:

$$f(y) = \frac{\Gamma(y + 1/k)}{\Gamma(y+1)\Gamma(1/k)} \frac{(k\mu)^y}{(1+k\mu)^{y+1/k}}, \; y = 0, 1, \ldots$$

$k = 1/r$ and $\mu = rp/q$ for the more familiar parameterization of the Negative Binomial distribution.

Negative binomial regression

We can model a negative binomial regression with log link in PROC GENMOD

```
proc genmod data=crabs;
   model satellite = width / dist=nb link=log ;
   run;
```

Model Information

Data Set	WORK.CRAB
Distribution	Negative Binomial
Link Function	Log
Dependent Variable	Satellite

Criteria For Assessing Goodness Of Fit

Criterion	DF	Value	Value/DF
Deviance	171	195.8112	1.1451
Pearson Chi-Square	171	144.7507	0.8465
Log Likelihood		154.3889	

Analysis Of Parameter Estimates

Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	-14.0525	1.2642	-6.5303	-1.5747	10.28	0.0013
width	1	0.1921	0.0476	0.0987	0.2854	16.27	<0.0001
Dispersion	1	1.1055	0.1971	0.7795	1.5679		

NOTE: The negative binomial dispersion parameter was estimated by maximum likelihood.

Overdispersion-Zero-Inflated Poisson

Excessive zeroes in a data set can be one source of overdispersion; this appears to be the case for both the Ache data and the horseshoe crab data (though not the only source of over-dispersion in either case).
We use a mixture model, including a distribution degenerate at 0:

$$Y \sim F,\ F = \gamma F_0 + (1-\gamma)F_\mu$$

For $F_0, P_{F_0}(Y = 0) = 1$, while F_μ is a Pois(μ) distribution.

Overdispersion-ZIP

PROC NLIN in SAS can actually optimize a likelihood, provided the contribution to the likelihood for each value of Y is provided.

$$\begin{aligned} P_F(Y=0) &= \gamma P(Y=0|F_0) + (1-\gamma)P(Y=0|F_\mu) \\ &= \gamma + (1-\gamma)exp(-\mu(\mathbf{x})) \end{aligned}$$

$$\begin{aligned} P_F(Y=y) &= \gamma P(Y=y|F_0) + (1-\gamma)P(Y=y|F_\mu) \\ &= (1-\gamma)exp(-\mu(\mathbf{x}))\mu(\mathbf{x})^y/y!,\ y=1,2,3,\ldots \end{aligned}$$

Horseshoe Crab ZIP model

ZIP (and ZINB) models can actually be fit in GENMOD now. Here is a likelihood-based approach instead.

```
proc nlmixed data=crabs;
*Starting values.  For p0, use the frequency table;
*For b0 and b1, use fit from Poisson regression model;
parms p0=.5 b0=-3.0 b1=0.2;
mu0=exp(b0+b1*width);
if satellite=0 then do;
prob=p0+(1-p0)*exp(-mu0);
loglike=log(prob);
end;
else loglike=log(1-p0)+satellite*log(mu0)-mu0-lgamma(satellite+1);
model satellite~general(loglike);
run;
```

Horseshoe Crab ZIP model

In PROC GENMOD, it is straightforward to model $\gamma(\mathbf{x})$, the probability that Y is sampled from distribution F_0, as a logistic regression model.

```
proc genmod data=crabs;
  model satellite = width /dist=zip obstats;
  zeromodel  /link = logit; *we need to transform the answer to recover gamma;
run;

*We can add covariates to a logit model for whether or not a zero is observed;
proc genmod data=crabs;
  model satellite = width /dist=zip obstats;
  zeromodel width /link = logit;
run;
```

Generalized Linear Models (GLMs)

$$g(\mu) = \eta = \mathbf{X}\boldsymbol{\beta}$$

Model	Random	Link	Systematic
Linear Regression	Normal	Identify	Mixed
ANOVA	Normal	Identify	Categorical
ANCOVA	Normal	Identify	Mixed
Logistic Regression	Binomial	Logit	Mixed
Loglinear	Poisson	Log	Categorical
Poisson	Poisson	Log	Mixed
Multinomial Response	Multinomial	Generalized Logit	Mixed