

GLM with a Gamma-distributed Dependent Variable

Paul E. Johnson

October 6, 2014

1 Introduction

I started out to write about why the Gamma distribution in a GLM is useful.

In the end, I've found it difficult to find an example which proves that is true. If you fit a GLM with the correct link and right-hand side functional form, then using the Normal (or Gaussian) distributed dependent variable instead of a Gamma will probably not result in disaster.

However, I did find out something really important, something which is mentioned in Myers, Montgomery, and Vining at several points, but I did not appreciate it until now. The GLM really is different than OLS, even with a Normally distributed dependent variable. Using OLS with "manually transformed" data leads to horribly wrong parameter estimates.

Let y_i be the dependent variable with mean μ_i . Your theory is that the link is the "inverse" $g(\mu_i) = 1/\mu_i$ or the "natural log" $g(\mu_i) = \ln(\mu_i)$. The OLS and GLM estimates will differ for any nonlinear link function, and for any right hand side. For example, put what you want in place of $g(\mu_i)$ in your theory:

$$\ln(\mu_i) = f(X_i, b) = b_0 + b_1/x_i$$

or

$$1/\mu_i = f(X_i, b) = b_0 + b_1/x_i$$

then you estimate a Generalized Linear model with a Gamma distribution with

```
glm(y~I(1/x), family = Gamma(link="log"))
```

or

```
glm(y~I(1/x), family = Gamma(link="inverse"))
```

If you mistakenly use a Normal, as in

```
glm(y~I(1/x), family = gaussian(link="log"))
```

or

```
glm(y~I(1/x), family = gaussian(link = "inverse"))
```

then the estimated b's from the Gamma and Normal models will probably be similar. If your dependent variable is truly Gamma, the Gaussian is "wrong" on a variety of levels, but the predicted values are "about right."

However, if you think you can just transform the variables yourself and run this through an OLS program, as in

```
lm(ln(y)~I(1/x))
```

or

```
lm(I(1/y)~I(1/x))
```

then the parameter estimates will be far from the mark. That happens because you have not transformed the expected values, but rather the observed values.

The only time that the GLM and OLS estimates line up is when the link function is the “identity” function

```
glm(y~I(1/x), family = Gamma(link = identity))
```

will be similar to the OLS estimates from

```
lm(y~I(1/x))
```

What do I conclude from this? You can’t stay with your old friend OLS. You really must learn and understand the GLM. My feeling then is similar to the comment attributed to NBA great Bill Russell: “They scheduled the game. We have to play. We might as well win.” If you have to learn the GLM anyway, and you use it, you might as well use the correct distribution while you are doing it.

2 Review the Gamma Handout

The Gamma handout is available in the Distributions folder of my web site.

To review briefly, let the shape parameter be α_i and scale be β_i . For the i ’th case being considered, we are acting as though there are individualized parameters for each case. It is annoying to keep all of this indexed by i , but sometimes it pays off. The probability density of observing a particular value y_i given parameters α_i and β_i is

$$f(y_i) = \frac{1}{\beta_i^{\alpha_i} \Gamma(\alpha_i)} y_i^{(\alpha_i-1)} e^{-(y_i/\beta_i)} \quad y_i, \alpha_i, \beta_i > 0$$

and

$$E(y_i) = \alpha_i * \beta_i$$

$$Var(y_i) = \alpha_i * \beta_i^2$$

Regression with the gamma model is going to use input variables X_i and coefficients to make a prediction about the mean of y_i , but in actuality we are really focused on the scale parameter β_i . This is so because we assume that α_i is the same for all observations, and so variation from case to case in $\mu_i = \beta_i \alpha$ is due simply to variation in β_i . The shape parameter is just a multiplier (which is equal to the inverse of the “dispersion parameter” ϕ that is defined for all distributions that are members of the exponential family).

3 Note the linkage between mean and variance

The ratio of the mean to the variance is a constant—the same no matter how large or small the mean is. As a result, when the expected value is small—near zero—the variance is small as well. Conversely, when the expected value is larger, the observed scores are less predictable in absolute terms.

$$\frac{Var(y_i)}{E(y_i)} = \frac{\alpha_i \beta_i^2}{\alpha_i \beta_i} = \beta_i$$

If your Gamma variable has an expected value of 100, the variance has to be $100 \cdot \beta_i$. Strange, but true.

The so-called coefficient of variation, which is used in introductory statistics as a summary of variability, is the ratio of standard deviation to mean. It is also a constant

$$CV = \frac{\sqrt{Var(y_i)}}{E(y_i)} = \frac{\sqrt{\alpha_i \cdot \beta_i^2}}{\alpha_i \beta_i} = \frac{\sqrt{\alpha_i} \beta_i}{\alpha_i \beta_i} = \frac{1}{\sqrt{\alpha_i}}$$

If your Gamma variable’s expected value is 100, the standard deviation is $100/\sqrt{\alpha_i}$.

It seems odd (surprising, interesting, possibly mistaken) to me that the ratio Var/E depends on β_i but the ratio of $StdDev/E$ depends on α_i .

The relationship between mean and variance here is different than some other distributions because it is “adjustable”. In contrast, the Poisson or Negative Binomial distributions have no such tuning parameter.

4 Gamma as a member of the Exponential Family

In order to treat this as the basis for a Generalized Linear Model, you act as though α is a known feature, the same for all observations. So we don’t need to write subscripts on α . Then we treat β_i —the scale parameter—as the parameter of interest. That is what we are trying to predict.

Recall the exponential family has this form, $exp[(y_i \cdot \theta_i - c(\theta_i))/\phi + h(y_i, \phi)]$. Rearrange the density for the Gamma as follows:

$$exp\{-y_i/\beta_i + (\alpha - 1)ln(y_i) - ln[\beta_i^{\alpha}] - ln[\Gamma(\alpha)]\}$$

$$exp\{-y_i/\beta_i + (\alpha - 1)ln(y_i) - \alpha ln[\beta_i] - ln[\Gamma(\alpha)]\}$$

$$exp\{-y_i/\beta_i - \alpha ln[\beta_i] + (\alpha - 1)ln(y_i) - ln[\Gamma(\alpha)]\}$$

Now a sneaky math guy trick appears. “Guess” that the natural parameter is

$$\theta_i = -\frac{1}{\alpha\beta_i}$$

Consequently,

$$\frac{-1}{\beta_i} = \theta_i\alpha$$

and

$$\beta_i = -\frac{1}{\theta_i\alpha}$$

Using those findings in the previous expression,

$$exp\{\alpha y_i\theta_i - \alpha ln(-\frac{1}{\theta_i\alpha}) - \alpha ln(\alpha) + (\alpha - 1)ln(y_i) - ln[\Gamma(\alpha)]\}$$

$$exp\{\alpha y_i\theta_i - \alpha ln(-\frac{\alpha}{\theta_i\alpha}) + (\alpha - 1)ln(y_i) - ln[\Gamma(\alpha)]\}$$

$$exp\{\alpha y_i\theta_i - \alpha ln(-\frac{1}{\theta_i}) + (\alpha - 1)ln(y_i) - ln[\Gamma(\alpha)]\}$$

$$exp\{\alpha(y_i\theta_i - ln(-\frac{1}{\theta_i})) + (\alpha - 1)ln(y_i) - ln[\Gamma(\alpha)]\}$$

That was quite a lot of work to find out that $\alpha = 1/\phi$ and that $c(\theta_i) = ln(-1/\theta_i)$. But if we rearrange just one more time, we find the Gamma in the form of the exponential density.

$$exp\{\frac{y_i\theta_i - ln(-1/\theta_i)}{\phi} + (\frac{1 - \phi}{\phi})ln(y_i) - ln[\Gamma(\phi^{-1})]\}$$

Then you can use the GLM Facts described on my GLM handout #1.

GLM Fact #1 states that $\mu_i = dc(\theta_i)/d\theta_i$, and so that implies the Gamma’s μ_i is

$$\frac{dc(\theta_i)}{d\theta_i} = \frac{dln(-1/\theta_i)}{d\theta_i} = -\frac{dln(\theta_i)}{d\theta_i} = -\frac{1}{\theta_i} = \alpha_i\beta_i$$

GLM Fact #2 states that $V(\mu_i) = d^2c(\theta_i)/d\theta_i^2$, and so, in this case,

$$V(\mu_i) = \frac{d}{d\theta_i^2}(-1/\theta) = \frac{1}{\theta_i^2} = \mu^2 = (\alpha\beta_i)^2$$

These findings are internally consistent with what we know already (or can check in textbooks). Recall from the GLM notes that the observed variance of y_i has two components: .

$$Var(y_i) = \phi_i V(\mu_i)$$

For the Gamma, we already know that $E(y_i) = \mu_i = \alpha * \beta$ and $Var(y_i) = \alpha * \beta^2$. The variance function is $V(\mu_i) = \mu_i^2 = \alpha^2 * \beta^2$, and the dispersion parameter ϕ_i must be equal to the reciprocal of the shape parameter $1/\alpha$. You can easily verify that all of these separate pieces work together in a logical way:

$$\begin{aligned} Var(y_i) &= \phi_i V(\mu) = \phi_i \cdot \alpha^2 \beta^2 \\ &= \alpha \beta^2 \text{ where } \phi_i = \frac{1}{\alpha} \end{aligned}$$

Keep in mind, then, that when the GLM routine estimates dispersion- ϕ -it is estimating the reciprocal of the shape parameter.

5 The Reciprocal is the Canonical Link

The canonical link for the GLM with a Gamma-distributed dependent variable is the reciprocal, $1/\mu_i$. That means that the expected value of your observed y_i , ($E(y_i) = \mu_i$), is related to your input variables as, for example,

$$\frac{1}{\mu_i} = b_0 + b_1 x_{1i}$$

Which obviously implies

$$\mu_i = \frac{1}{b_0 + b_1 x_{1i}}$$

Plot that! In Figure 1, you see there is some serious potential for funny business with this function.

6 Why would you want a Gamma-distributed dependent variable?

This is a difficult question. Theoretically, the Gamma should be the right choice when the dependent variable is real-valued on a range from 0 to ∞ . And the Gamma is suitable when you suspect the linkage between mean and variance is “fixed”. If you expect a small value of y_i , you should also expect only a small amount of variability in observed values. Conversely, if you expect a huge value of y_i , you should expect a lot of variability.

However, after some testing, I have developed some doubts about the need to change from a model based on the Normal distribution to a model based on the Gamma. The Gamma may be “theoretically right” but there are several cases in which the old “theoretically wrong” Normal OLS model seems to do about as well.

This is especially true if the Gamma parameters are tuned so that the distribution is symmetrical, but even when it is pretty badly skewed, I find the OLS predictions are as good.

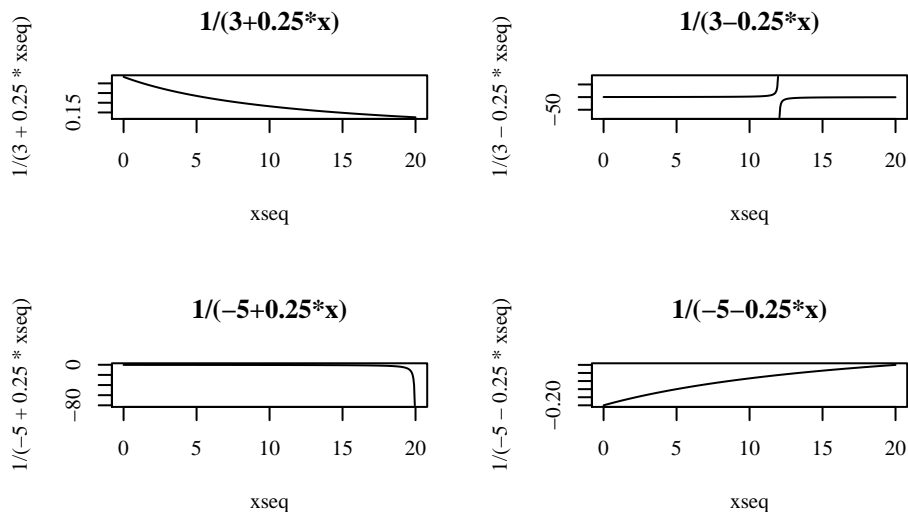
However, I find some cases where using the GLM with a Gamma distribution has a dramatic impact. The differences hinge on the functional form being investigated.

So I’ve prepared some vignettes.

7 Reciprocal Relationship: $\mu_i = b_0 + b_1/x_i$

Some simulated data is presented in Figure2. The line represents the “true” value of μ_i , the expected value of the dependent variable.

Figure 1: Reciprocal Link



7.1 GLM fit for the Reciprocal Model

The Generalized Linear Model can be fit with the “identity” link with these commands.

```
agam <- glm(yobs ~ I(1/xseq), family = Gamma(link = "identity"), control =
  glm.control(maxit=100), start=c(1,65))
library(MASS)
myshape <- gamma.shape(agam)
gampred <- predict(agam, type="response", se=T, dispersion=1/myshape$alpha)
```

(Side note about estimating dispersion: This uses the MASS library’s function `gamma.shape` to calculate a more precise estimate of the gamma distribution’s shape parameter, which is equal to the reciprocal of the GLM’s dispersion $\alpha = 1/\phi$. This is useful because the estimate of the dispersion offered by the default GLM summary command does not take into account the special information about the dispersion that can be calculated by using the Gamma distribution. Not all GLMs have a model-specific, enhanced way to estimate dispersion.)

```
summary(agam, dispersion=1/myshape$alpha)
```

```
Call:
glm(formula = yobs ~ I(1/xseq), family = Gamma(link = "identity"),
    start = c(1, 65), control = glm.control(maxit = 100))
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.5702	-0.8164	-0.2555	0.3364	2.5154

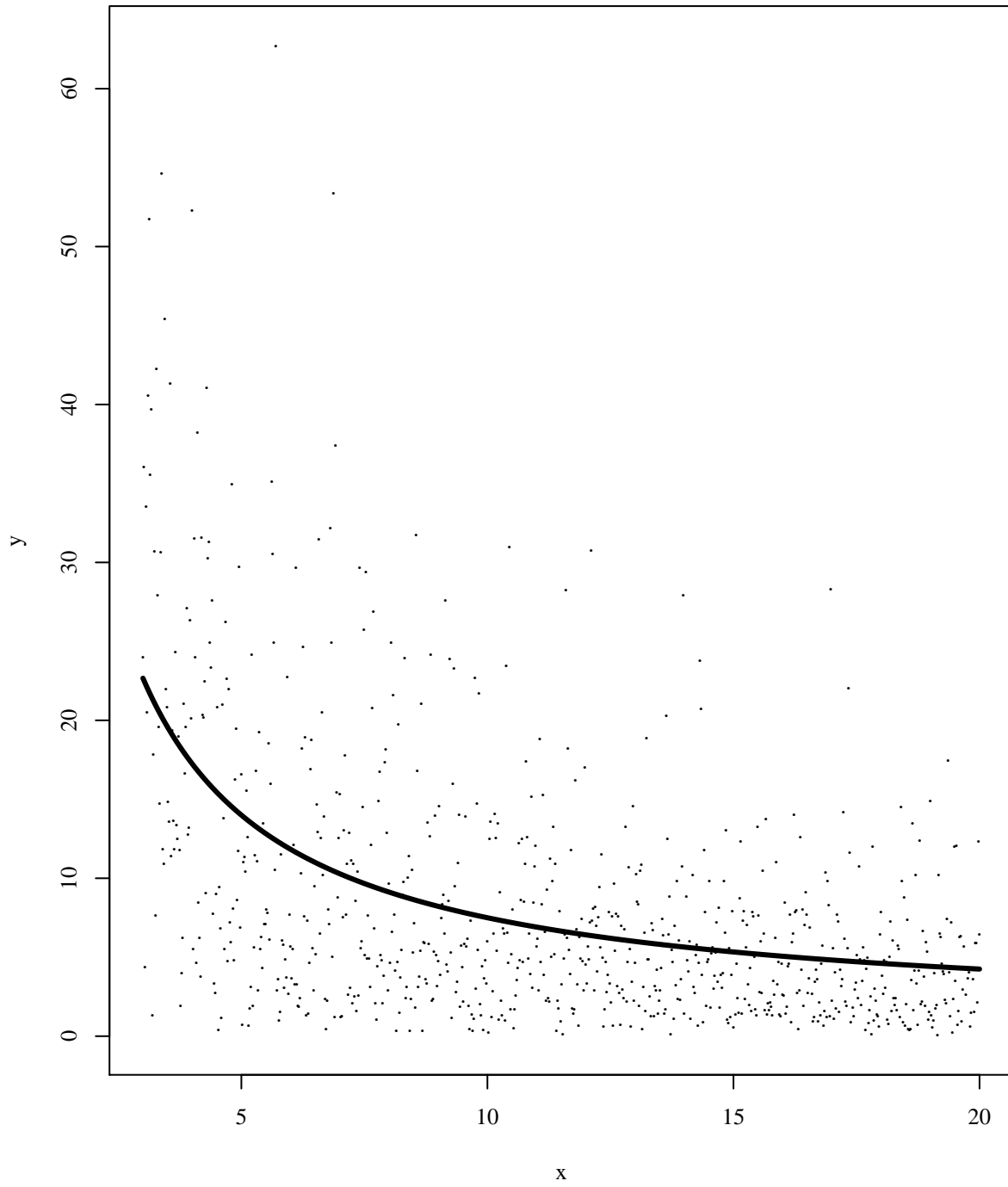
Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.7738	0.4524	1.71	0.0872 .
I(1/xseq)	67.7209	5.4534	12.42	<2e-16 ***

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

Figure 2: Gamma Dependent Variable $\mu_i = 1 + 65/x_i, \text{shape}=1.5$

Gamma DV, mu= 1 + 65/x



```
(Dispersion parameter for Gamma family taken to be 0.6757266)
```

```
Null deviance: 786.66 on 799 degrees of freedom  
Residual deviance: 599.11 on 798 degrees of freedom  
AIC: 4756
```

```
Number of Fisher Scoring iterations: 3
```

7.2 Linear Model Fit with the Normal Distribution

Suppose you make the mistaken assumption that this data is Normally distributed. The default settings of the glm estimator in R lead to estimates for a Normally distributed dependent variable with the identity link.

```
lmmod <- glm(yobs~ I(1/xseq))  
lmpred <- predict(lmmod, se=T)
```

We have asked predict for the standard errors because they are useful for plots shown below.

```
summary(lmmod)
```

```
Call:
```

```
glm(formula = yobs ~ I(1/xseq))
```

```
Deviance Residuals:
```

```
    Min       1Q   Median       3Q      Max  
-21.467  -3.929  -1.500   2.537   49.790
```

```
Coefficients:
```

```
              Estimate Std. Error t value Pr(>|t|)  
(Intercept)    0.3723     0.5212   0.714   0.475  
I(1/xseq)      71.6044     4.0309  17.764 <2e-16 ***
```

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

```
(Dispersion parameter for gaussian family taken to be 55.12723)
```

```
Null deviance: 61387 on 799 degrees of freedom  
Residual deviance: 43992 on 798 degrees of freedom  
AIC: 5482
```

```
Number of Fisher Scoring iterations: 2
```

Please note that you get the same parameter estimate if you put the same relationship through the ordinary least squares regression procedure, lm.

```
lmmod1a <- lm ( yobs~I(1/xseq))  
summary(lmmod1a)
```

```
Call:
```

```
lm(formula = yobs ~ I(1/xseq))
```

```
Residuals:
```

	Min	1Q	Median	3Q	Max
	-21.467	-3.929	-1.500	2.537	49.790
Coefficients:					
	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	0.3723	0.5212	0.714	0.475	
I(1/xseq)	71.6044	4.0309	17.764	<2e-16	***
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					
Residual standard error: 7.425 on 798 degrees of freedom					
Multiple R ² : 0.2834, Adjusted R ² : 0.2825					
F-statistic: 315.6 on 1 and 798 DF, p-value: < 2.2e-16					

7.3 Is the Normal Model “Just as Good”?

No, of course it isn't. It is the wrong model. But if you mistakenly used OLS, would you make a major mistake? In this test case, the answer is no.

1. The parameter estimates from the two models are “about the same.”

$$\text{Gamma GLM} : \hat{\mu}_i = 0.77 + 67.72/x_i$$

$$\text{Normal GLM} : \hat{\mu}_i = 0.37 + 71.60/x_i$$

2. Consequently, the predicted values from the two models are “about the same.”

Consider the plotted lines in Figure 3. It is difficult to distinguish the two lines representing the predicted values. I had a hard time believing that the two lines could actually be so close to one another, so I printed out the first 10 observations of the two models:

```
cbind(glmGamma=gampred$fit [1:10] , glmNormal=lmpred$fit [1:10])
```

	glmGamma	glmNormal
1	23.34747	24.24043
2	23.18850	24.07235
3	23.03175	23.90661
4	22.87719	23.74318
5	22.72475	23.58200
6	22.57440	23.42303
7	22.42610	23.26622
8	22.27980	23.11154
9	22.13546	22.95892
10	21.99305	22.80835

The plotted estimates of the means, along with the “confidence intervals” , are illustrated in Figure 4.

3. If you (mistakenly) choose models by T statistics, you will be wrong.

It upsets me when students say one model is “more significant” to mean that a model has coefficients with bigger t values. In this case, the t value for the coefficient of $1/x_i$ in the Normal model is 17.76, while the comparable value from the Gamma fit is 12.42. That does not mean the Normal model is

Figure 3: Fitted Models for Gamma Dependent Variable

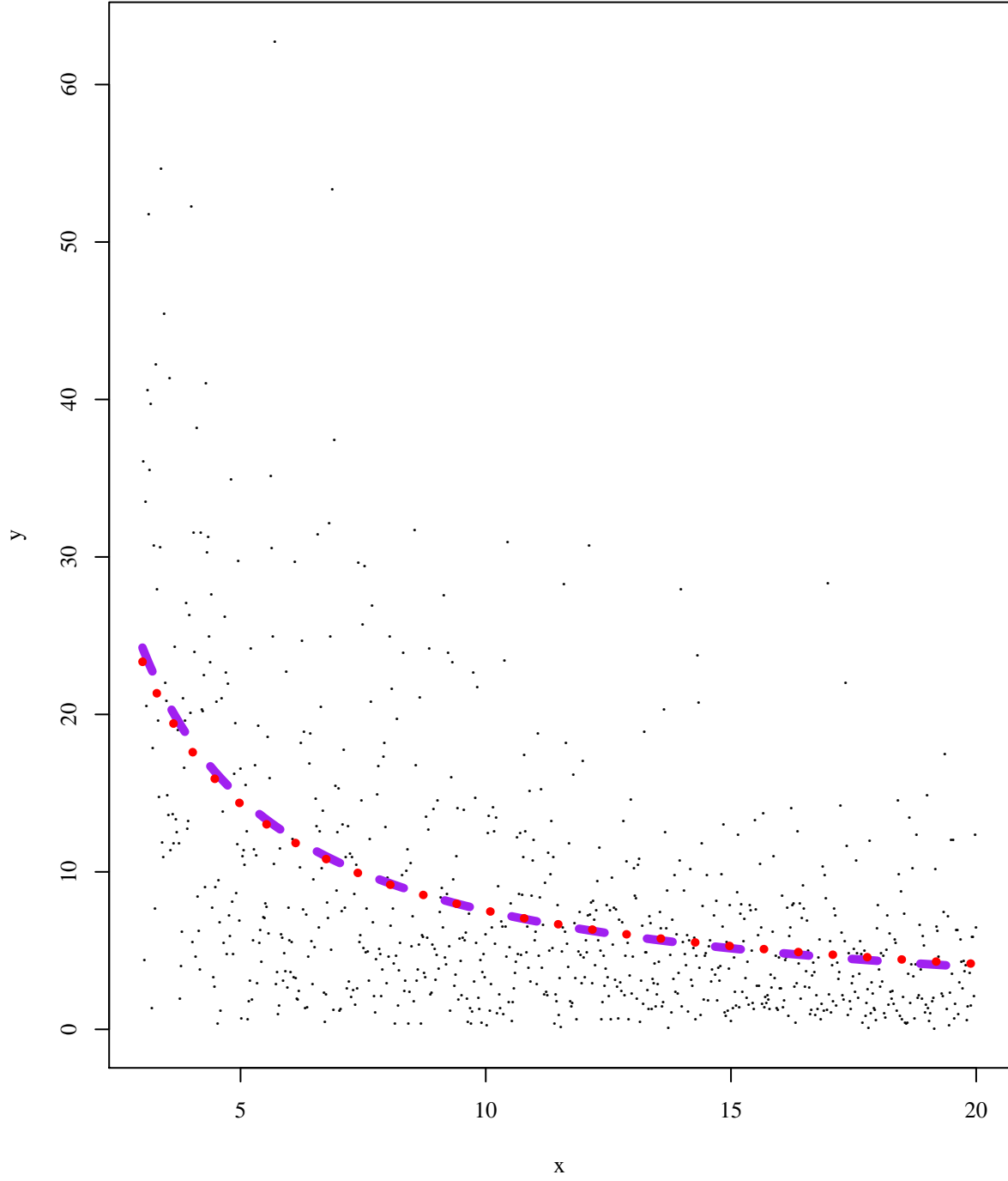
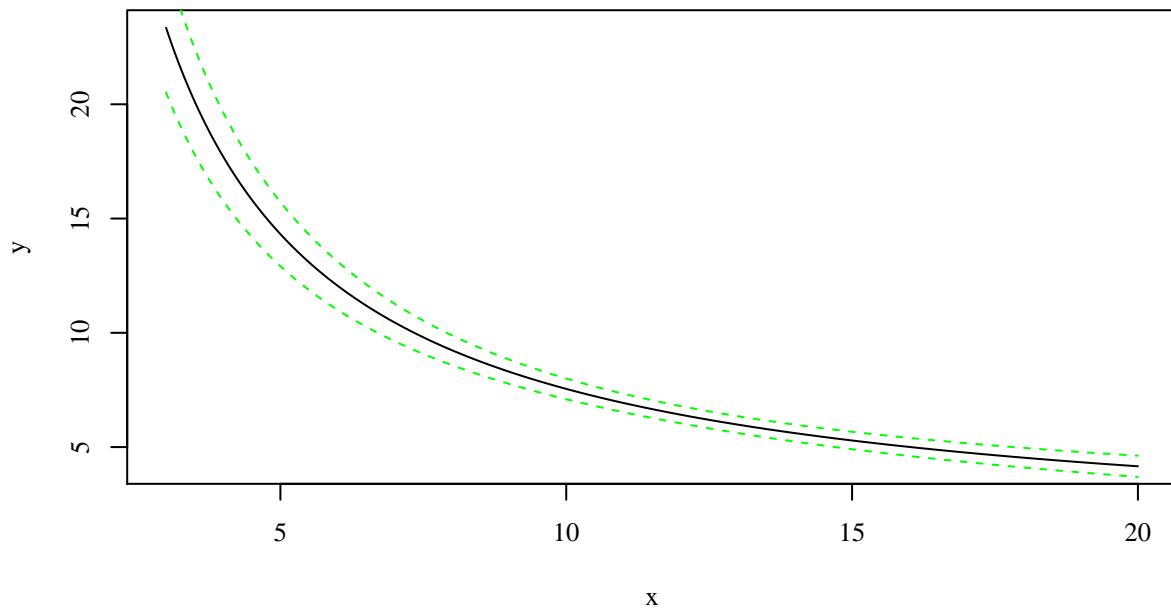


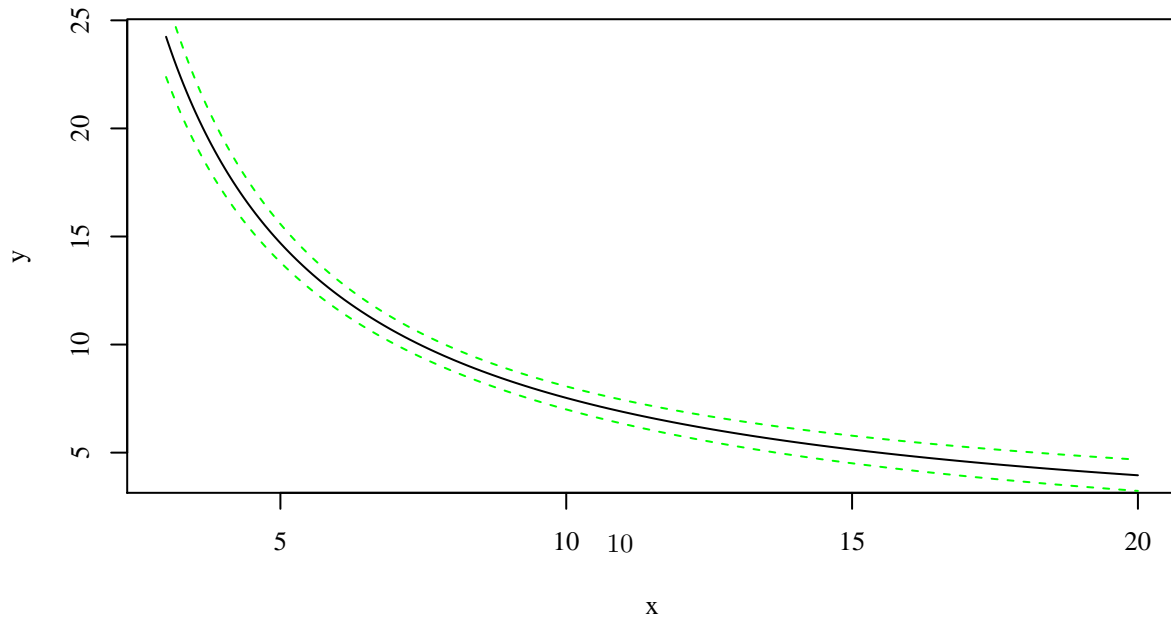
Figure 4: Predicted Values from glm with Gamma and Gaussian Distributions

```
par(mfcol=c(2,1))
plot(xseq, gampred$fit, type="l", xlab="x", ylab="y", main="Gamma, link=identity")
lines(xseq, gampred$fit+2*gampred$se, lty=2, col="green")
lines(xseq, gampred$fit-2*gampred$se, lty=2, col="green")
plot(xseq, lmpred$fit, type="l", xlab="x", ylab="y", main="Gaussian, link=identity")
lines(xseq, lmpred$fit+2*lmpred$se, lty=2, col="green")
lines(xseq, lmpred$fit-2*lmpred$se, lty=2, col="green")
```

Gamma, link=identity



Gaussian, link=identity



better, for many reasons. The fact is that these tests assume you have chosen the correct model and then estimate on the variability of the \hat{b} based on your specification. They do not constitute a way to choose between two models.

4. The Deviance is different: Gamma looks significantly better.

The residual deviance of the Gamma fit is 599.11 on 798 degrees of freedom, and the Akaike Information Criterion is 4765. The residual deviance of the Normal model is 43992 on 798 degrees of freedom, and the AIC is 5482. The model with the smaller AIC is preferred.

8 Michaelson-Morley model: $\mu_i = x_i / (b_1 + b_0 x_i)$

In several fields, scholars have proposed a model with this functional form:

$$\mu_i = \frac{x_i}{b_1 + b_0 x_i}$$

Note: the numbering of the coefficients is not mistaken.

Write it like this and you see why the reciprocal link (the canonical link) makes sense:

$$\frac{1}{\mu_i} = \frac{b_1 + b_0 x_i}{x_i}$$

$$\frac{1}{\mu_i} = b_0 + b_1 \frac{1}{x_i}$$

Write it like this and it should remind you of a logistic regression.

$$\mu_i = \frac{1}{b_0 + b_1(1/x_i)}$$

One should graph that. For $x_i = 0$, this is undefined (just as Gamma density is undefined). For very small values of x_i , say 0.001, you can see the expected value is a very small number. As x_i gets bigger and bigger, the expected value tends to $1/b_0$.

Variants of this are known in ecology, biochemistry, and physics. In R, one finds it discussed as the Michaelson-Morley model.

A hypothetical example of Gamma distributed data with $\mu_i = x_i / (3 + 0.25x_i)$ with the Gamma shape parameter equal to 1.5 is presented in Figure 5.

8.1 GLM fit with a Gamma Variable and Log Link

If one algebraically re-arranged the model as

$$\frac{1}{\mu_i} = b_0 + b_1 \frac{1}{x_i}$$

then one would have to transform the input variable in the regression model, but the glm procedure will handle the transformation of the left hand side. One should write the glm formula as

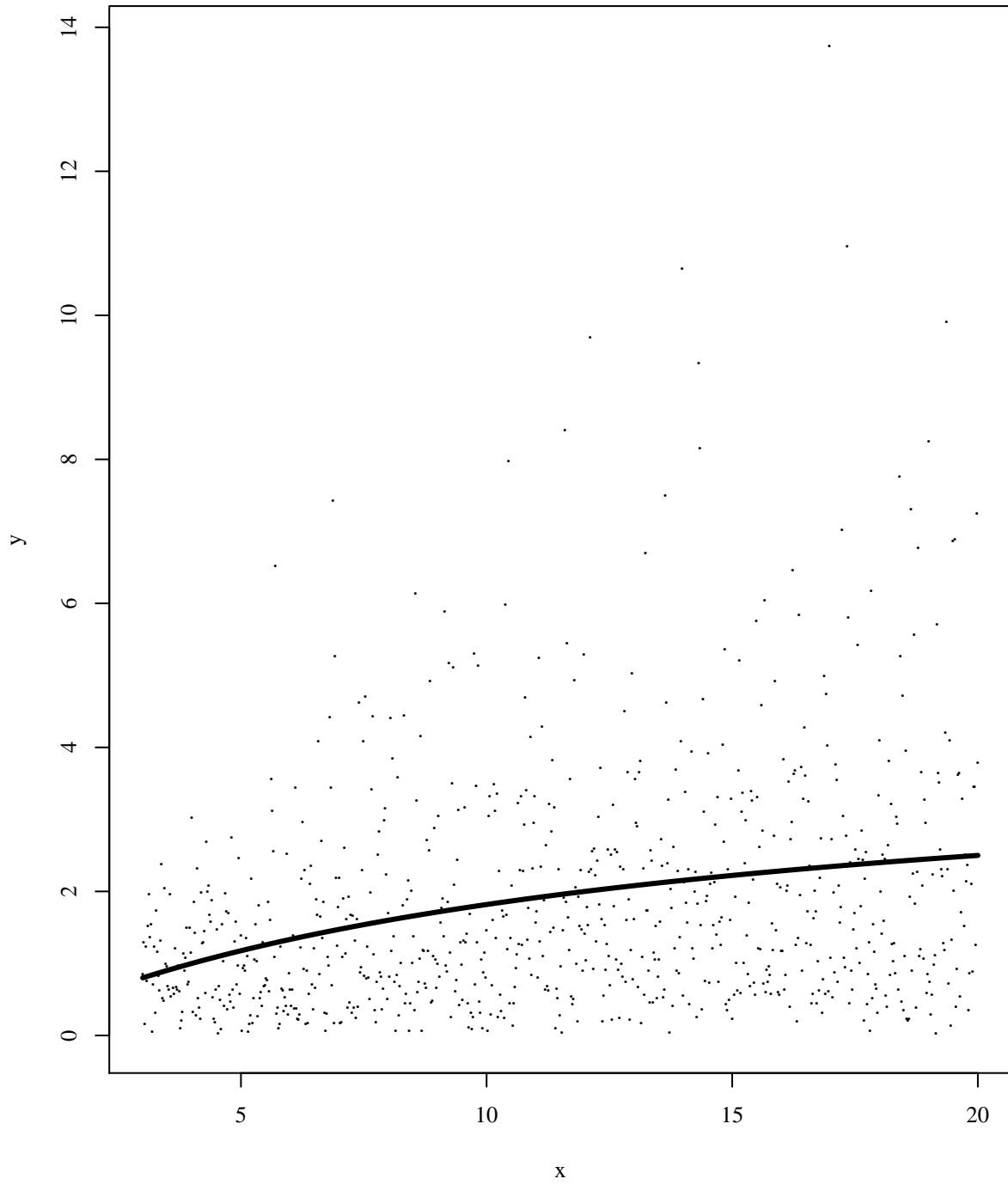
$$y \sim I(1/x)$$

but specify the link as the “inverse”, so that the left hand side is y_i is transformed.

The glm procedure to fit a Gamma distributed dependent variable of this sort is:

```
agam2<-glm(yobs2~ I(1/xseq), family=Gamma(link="inverse"), control=
  glm.control(maxit=100), start=c(2,4))
library(MASS)
myshape2 <- gamma.shape(agam2)
gampred2 <- predict(agam2,type="response", se=T, dispersion=1/myshape2$alpha)
```

Figure 5: Gamma Dependent Variable $\mu_i = x_i/(3 + 0.25x_i)$



This uses the MASS library's `gamma.shape` method to get a better estimate of the dispersion parameter, which is then used in making predictions and also in preparing the summary output. The estimate of the dispersion coefficient affects the standard errors, but not the estimates of the b 's.

```
summary(agam2, dispersion=1/myshape2$alpha)
```

```
Call:
glm(formula = yobs2 ~ I(1/xseq), family = Gamma(link = "inverse"),
     start = c(2, 4), control = glm.control(maxit = 100))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5708  -0.8150  -0.2532   0.3325   2.5156

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.26736     0.03352   7.976 1.51e-15 ***
I(1/xseq)    2.80621     0.34735   8.079 6.54e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Gamma family taken to be 0.6756965)

Null deviance: 654.43  on 799  degrees of freedom
Residual deviance: 599.09  on 798  degrees of freedom
AIC: 2479.4

Number of Fisher Scoring iterations: 7
```

8.2 What if you used OLS?

You can translate this into a form that looks like an ordinary regression model: just “tack on an error term” (recall OLS: expected value of 0, constant variance):

$$\frac{1}{y_i} = b_0 + b_1 \frac{1}{x_i} + e_i$$

and create transformed variables $1/y_i$ and $1/x_i$ and estimate this with OLS. How gauche.

```
lmmod2 <- lm(I(1/yobs2)~I(1/xseq))
lmpred2 <- predict(lmmod2, se=T)
```

There are a number of reasons why you should not do that. It violates the usual OLS assumptions. It assumes the mismatch between the expected and observed is of a very peculiar sort, $E(e_i) = 0$ and constant variance.

The most important reason why you should not fit these parameters with OLS is that the resulting parameter estimates are grossly wrong.

```
summary(lmmod2)
```

```
Call:
lm(formula = I(1/yobs2) ~ I(1/xseq))

Residuals:
    Min       1Q   Median       3Q      Max
```

```
-2.243 -1.002 -0.707 -0.040 34.966
```

```
Coefficients:
```

```
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.8402      0.2020   4.159 3.55e-05 ***
I(1/xseq)      5.9798      1.5627   3.827 0.00014 ***
```

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

```
Residual standard error: 2.878 on 798 degrees of freedom
Multiple R2: 0.01802, Adjusted R2: 0.01679
F-statistic: 14.64 on 1 and 798 DF, p-value: 0.0001401
```

Note, the predicted values from this model are presented on a reciprocal scale, so the predicted values must be transformed as $1/\text{predicted}$ in order to be plotted on the scale of the original, untransformed data.

8.3 But did you really need the Gamma in the glm?

Here we are back to the main question: is it the functional form that is the source of the trouble, or is it the assumed statistical distribution.

The reason that the OLS estimation fails so dramatically is that all of the y_i values are transformed. We really only wanted to represent the transformation $1/\mu_i$, but in the `lm` framework, doing so requires us to transform the observed values $1/y_i$. We really want only to model the transformation of the mean and GLM does that.

Now we can use `glm` with a Gaussian distribution but with an inverse link. And, unlike the model discussed in the previous section, there is a difference between using `lm` and `glm` on the same data.

To estimate the model with GLM,

$$\frac{1}{\mu_i} = b_0 + b_1 \frac{1}{x_i}$$

the following R commands are used:

```
lmmod3 <- glm(yobs2 ~ I(1/xseq), family=gaussian(link="inverse"))
summary(lmmod3)
```

```
Call:
```

```
glm(formula = yobs2 ~ I(1/xseq), family = gaussian(link = "inverse"))
```

```
Deviance Residuals:
```

```
    Min       1Q   Median       3Q      Max
-2.3914 -1.0866 -0.3662  0.5736  11.4282
```

```
Coefficients:
```

```
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.26355      0.03979   6.623 6.47e-11 ***
I(1/xseq)      2.85141      0.50428   5.654 2.18e-08 ***
```

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

```
(Dispersion parameter for gaussian family taken to be 2.702469)
```

```
Null deviance: 2320.6 on 799 degrees of freedom
Residual deviance: 2156.5 on 798 degrees of freedom
AIC: 3069.6
```

Figure 6: Fitted Models for $\mu = x_i/(3 + .25x_i)$ with a Gamma Distributed Variable

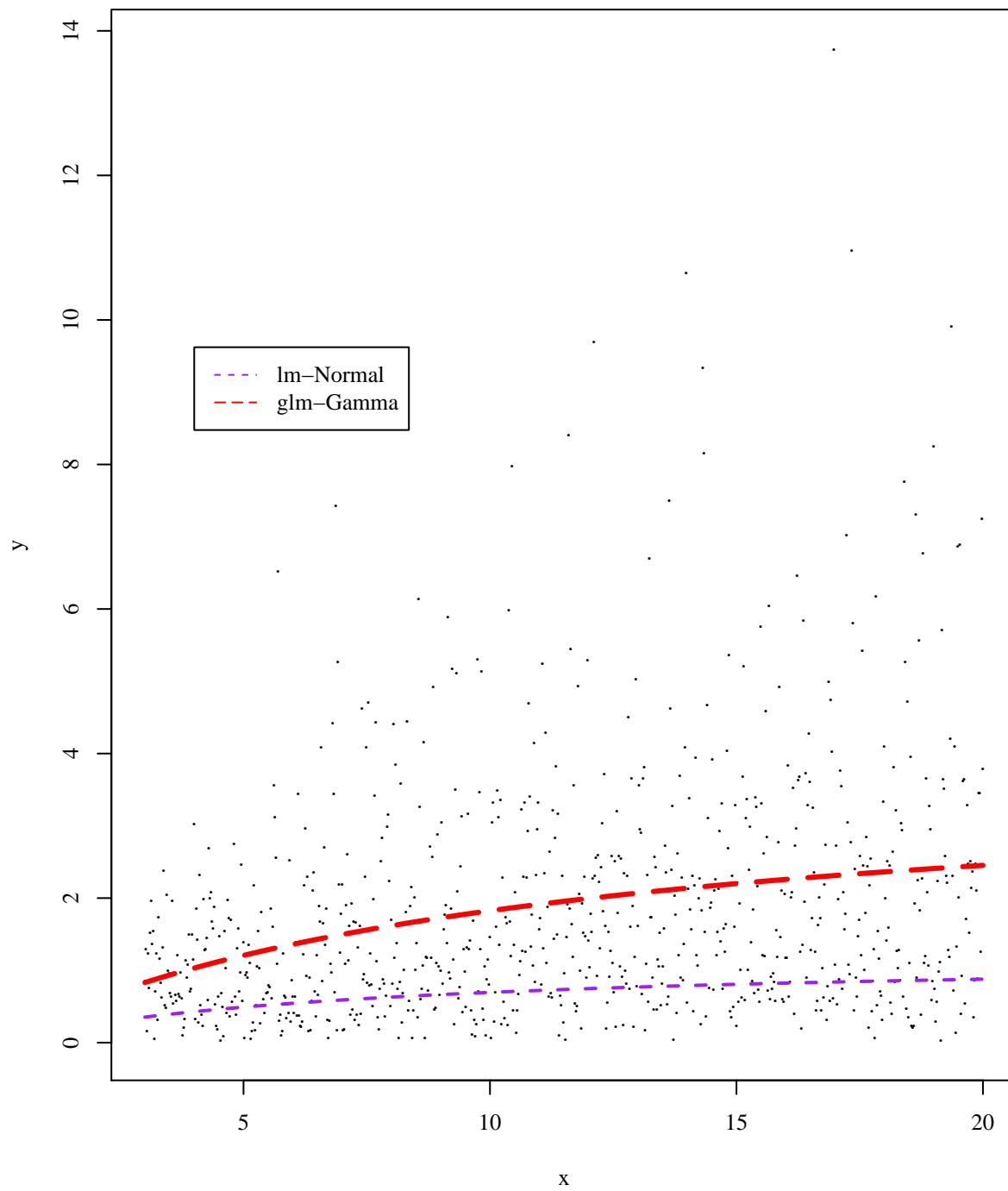
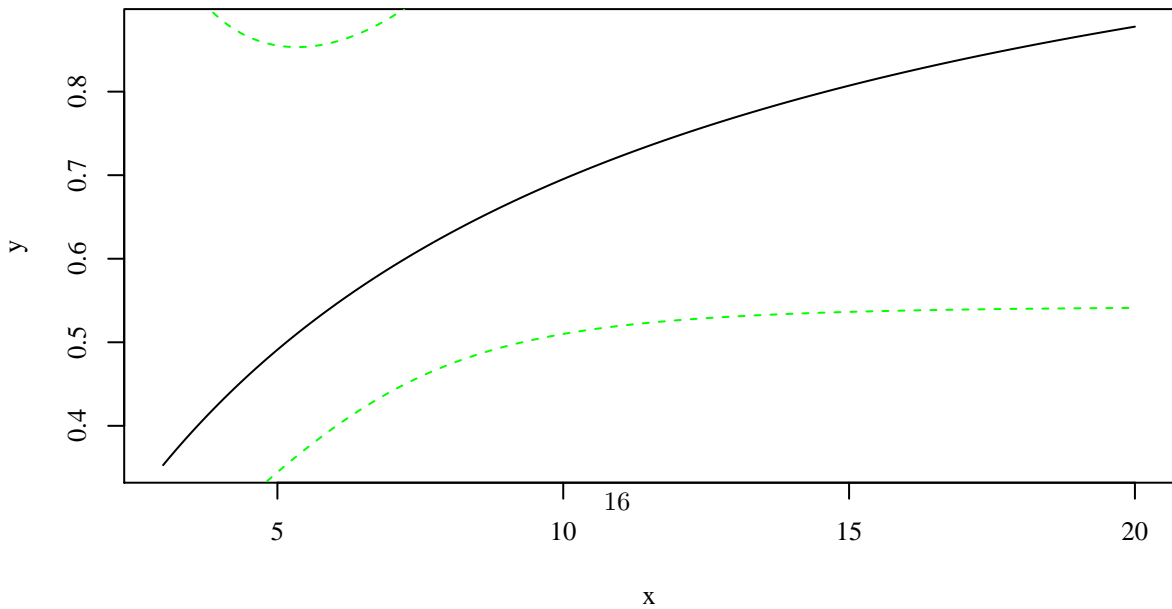
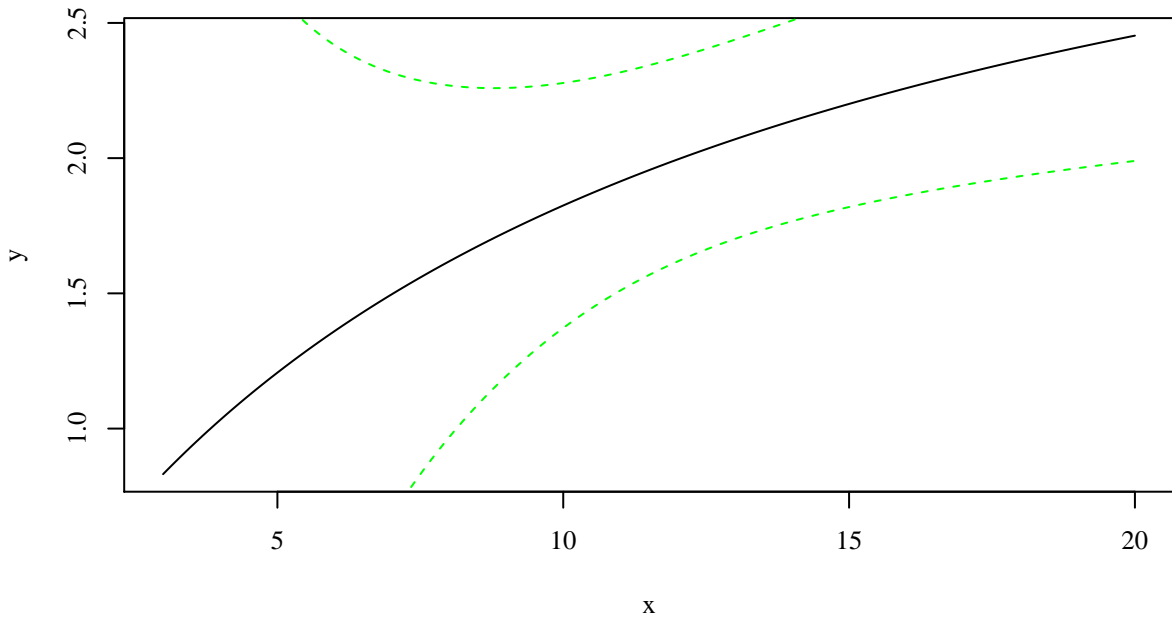


Figure 7: Predicted Values from glm-Gamma and lm-Normal

```
par(mfcol=c(2,1))
plot(xseq, gampred2$fit, type="l", xlab="x", ylab="y", main="Gamma, link=inverse")
lines(xseq, gampred2$fit+2*gampred$se, lty=2, col="green")
lines(xseq, gampred2$fit-2*gampred$se, lty=2, col="green")
plot(xseq, 1/lmpred2$fit, type="l", xlab="x", ylab="y", main="")
ypsem<- lmpred2$fit-1.96*lmpred$se
ypsep<- lmpred2$fit+1.96*lmpred$se
lines(xseq, 1/ypsem, lty=2, col="green")
lines(xseq, 1/ypsep, lty=2, col="green")
```

Gamma, link=inverse



Note that the parameter estimates coincide with the GLM-Gamma estimates of the earlier analysis. Figure 8 plots the predicted values for the glm models fit with the Normal and Gamma distributions. The predicted values once-again coincide in these two models fit with glm.

8.4 In the end, its all about the deviance

The residual deviance for the GLM-Gamma model in section 8.1 was 599.09 on 798 degrees of freedom with an AIC of 2479.4. The GLM-Normal model has deviance of 2156.6 and the AIC was 3069.6.

9 How does Gamma GLM fix that?

You might have noticed that the Gamma is a two-parameter distribution. However, it is not necessary to know α before doing the estimates of the slope coefficients. After fitting the slope coefficients, then the dispersion can be estimated (similar in the way OLS estimates the slopes and then the variance of the error term).

If your theory stated the M-M relationship,

$$\mu_i = \frac{x_i}{b_1 + b_0 x_i}$$

you would re-arrange like so

$$\frac{1}{\mu_i} = b_0 + b_1 \frac{1}{x_i}$$

So your link function is the inverse function (reciprocal). If y_i is distributed as a Gamma with that mean, then the R glm procedure would be invoked as

```
glm(y~I(1/x),family=Gamma(link='inverse'))
```

It is necessary to “manually” transform the right hand side. You are completely free to use any formula you like. This specifies $1/x$ in order to match the M-M model. As long as your formula is additive on the right hand side, one can make any legitimate transformations, such as logs, squares, square roots, sines, or cosines. However, it is not necessary to transform the left hand side in the formula. That is taken care of by specifying the link as the inverse.

The inverse is the canonical link, but R’s glm will fit models that specify the log or identity links. Thus one could theorize that y_i is Gamma-distributed and that its mean is related to the input variables as

$$\log(\mu_i) = b_0 + b_1 x_i$$

This implies, of course, that you think the relationship between your linear predictor η_i and the mean of y_i is exponential

$$\mu_i = \exp[b_0 + b_1 x_i]$$

Or one can have a simple linear relationship with the identity link

$$\mu_i = b_0 + b_1 x_i$$

Figure 8: Fitted Models for $\mu = x_i/(3 + .25x_i)$ with a Gamma Distributed Variable

