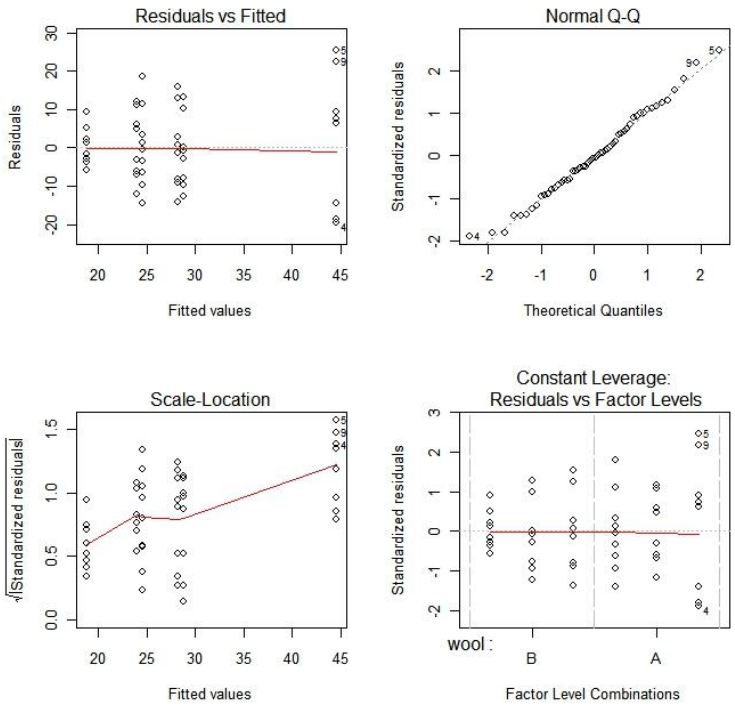
Day 1 Exercises with Solutions

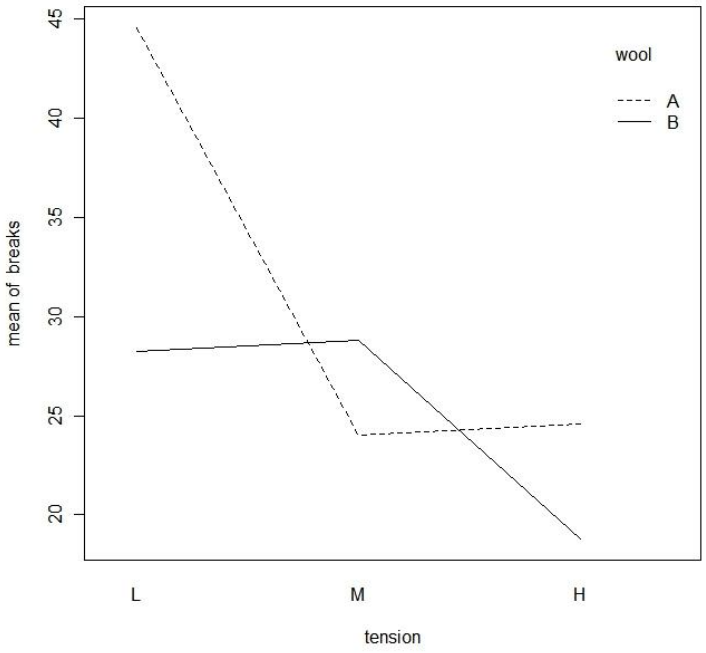
Linear Modeling

1) The **R** data frame **`warpbreaks`** gives the number of **`breaks`** per fixed length of wool during weaving, for two different **`wool`** types, and 3 different weaving **`tensions`**. Using a linear model, establish whether there is evidence that the effect of tension on break rate is dependent on the type of wool. If there is, use **`interaction.plot()`** function to examine the nature of the dependence.

```
> data(warpbreaks)
> head(warpbreaks)

  breaks wool tension
1     26    A       L
2     30    A       L
3     54    A       L
4     25    A       L
5     70    A       L
6     52    A       L

> wm <- lm(breaks~wool*tension,data=warpbreaks)
> par(mfrow=c(2,2))
> plot(wm)  # residual plots are fine
```



```
> anova(wm)
Analysis of Variance Table

Response: breaks
             Df Sum Sq Mean Sq F value    Pr(>F)
wool          1  450.7  450.67  3.7653 0.0582130 .
tension       2 2034.3 1017.13  8.4980 0.0006926 ***
wool:tension  2 1002.8  501.39  4.1891 0.0210442 *
Residuals    48 5745.1  119.69
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## Can see significant interaction in analysis of variance table

> with(warpbreaks,interaction.plot(tension,wool,breaks))
```



Can see the interaction, particularly at medium tension for wools A and B

2) The **R** data frame **`cars`** contains data about the stopping distance and speed of cars when the driver was signaled to stop. It takes a fixed reaction time for drivers to apply their brakes, so the car will travel a distance directly proportional to its speed before beginning to slow. However, an automobile's kinetic energy is proportional to the square of its speed, but the brakes can only dissipate that energy, and slow the car, at a constant rate per unit distance traveled.

   Fit three different linear models to this data. Report the results.

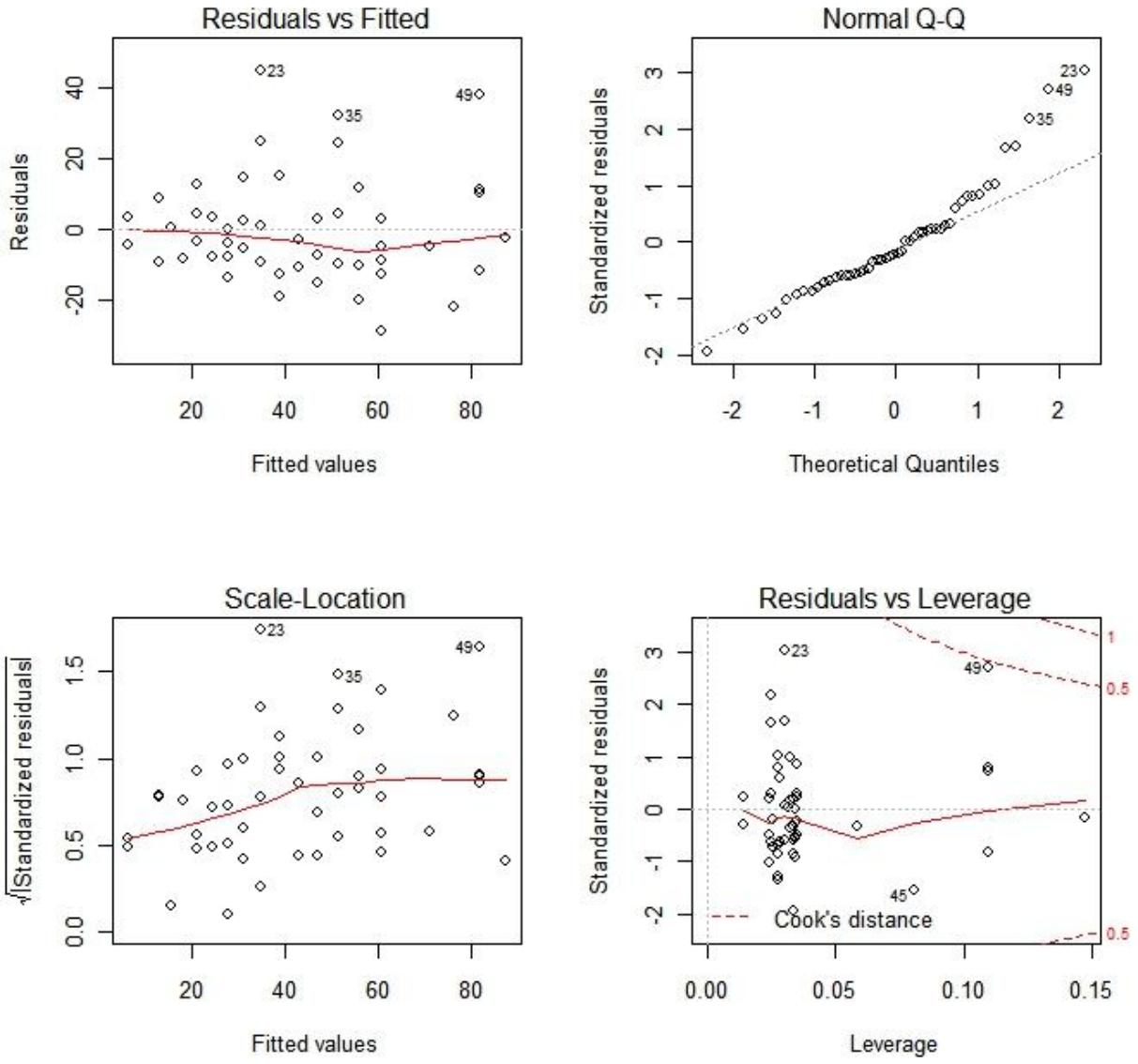
a) dist ~ $\beta_0 + \beta_1(\text{speed}) + \beta_2(\text{speed}^2) + e$
b) dist ~ $\beta_1(\text{speed}) + \beta_2(\text{speed}^2) + e$
c) dist ~ $\beta_1(\text{speed}) + e$

Which model seems to fit better? Why? *cm2 (the second model) fits better….both terms are significant, as is the F-statistic. Further, an anova table indicates that the second model fits better than the third model (the first is a complete misfit). Finally, the second model has the lowest AIC, another indicator of best fit. See solution script output below.*

```
> data(cars)
> head(cars)
  speed dist
1     4    2
2     4   10
3     7    4
4     7   22
5     8   16
6     9   10
> cm1 <- lm(dist ~ speed + I(speed^2),data=cars)
> summary(cm1)

Call:
lm(formula = dist ~ speed + I(speed^2), data = cars)
Residuals:
    Min      1Q  Median      3Q     Max
-28.720  -9.184  -3.188   4.628  45.152

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.47014   14.81716   0.167    0.868
speed        0.91329    2.03422   0.449    0.656
I(speed^2)   0.09996    0.06597   1.515    0.136

Residual standard error: 15.18 on 47 degrees of freedom
Multiple R-squared: 0.6673,  Adjusted R-squared: 0.6532
F-statistic: 47.14 on 2 and 47 DF,  p-value: 5.852e-12

> cm2 <- lm(dist ~ speed + I(speed^2)-1, data=cars)
> summary(cm2)

Call:
lm(formula = dist ~ speed + I(speed^2) - 1, data = cars)

Residuals:
    Min      1Q  Median      3Q     Max
-28.836  -9.071  -3.152   4.570  44.986

Coefficients:
           Estimate Std. Error t value Pr(>|t|)
speed       1.23903    0.55997   2.213  0.03171 *
I(speed^2)  0.09014    0.02939   3.067  0.00355 **
```

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

Residual standard error: 15.02 on 48 degrees of freedom
Multiple R-squared: 0.9133,  Adjusted R-squared: 0.9097
F-statistic: 252.8 on 2 and 48 DF,  p-value: < 2.2e-16

> cm3 <- lm(dist ~ speed-1, data=cars)
> summary(cm3)

Call:
lm(formula = dist ~ speed - 1, data = cars)

Residuals:
    Min      1Q  Median      3Q     Max
-26.183 -12.637  -5.455   4.590  50.181

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
speed   2.9091     0.1414   20.58   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.26 on 49 degrees of freedom
Multiple R-squared: 0.8963,  Adjusted R-squared: 0.8942
F-statistic: 423.5 on 1 and 49 DF,  p-value: < 2.2e-16

> anova(cm2,cm3,test="Chi")
Analysis of Variance Table

Model 1: dist ~ speed + I(speed^2) - 1
Model 2: dist ~ speed - 1
  Res.Df   RSS Df Sum of Sq P(>|Chi|)
1     48 10831
2     49 12954 -1   -2122.7  0.002162 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> AIC(cm2,cm3)
    df      AIC
cm2  3 416.8016
cm3  2 423.7498

> par(mfrow=c(2,2))
```

```
> plot(cm2)
```



You can see from the above residual plots that there are problems for both heteroscedasticity and for non-normality, but that is to be expected with a quadratic term in the model. Perhaps we should run a GLM instead of a linear model.

Using the second model in the above list, estimate the average time that it takes a driver to apply the brakes (there are 5280 feet in a mile).

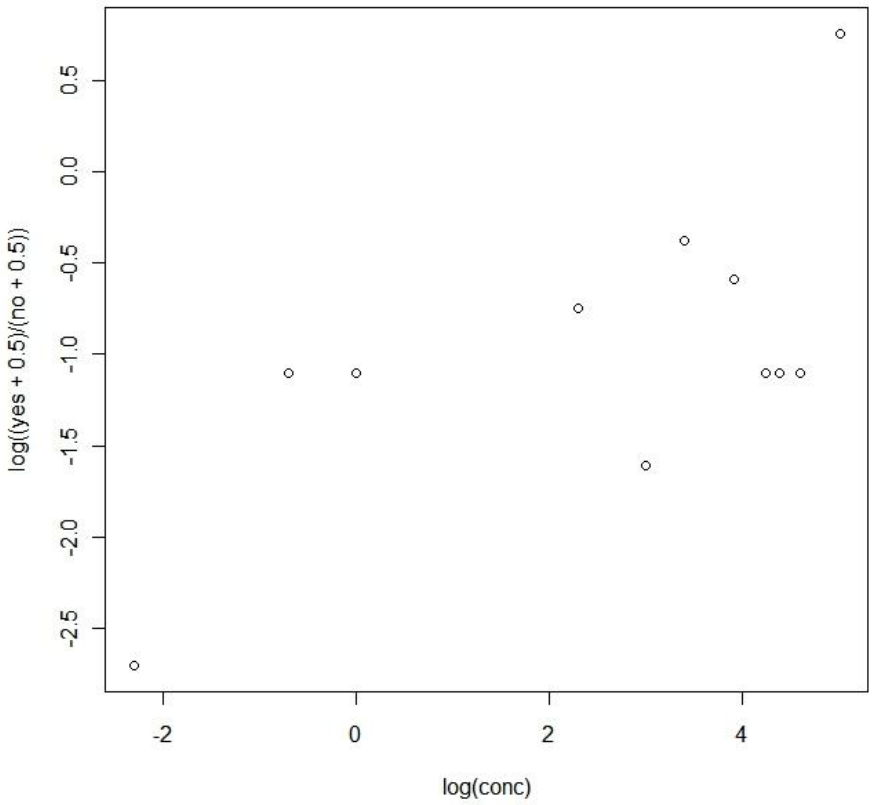
```
> b <- coef(cm2)
> 5280/(b[1]*60^2)
   speed
   1.183722 – on average it takes 1.18 seconds to apply brakes
```

Generalized Linear Modeling

3) The following table shows numbers of occasions when inhibition (i.e., no flow of current across a membrane) occurred within 120 s, for different concentrations of the protein peptide-C. The outcome *yes* implies that inhibition has occurred. Use logistic regression to model the probability of inhibition as a function of protein concentration. Report and plot your results fully. Interpret your results.

conc	0.1	0.5	1	10	20	30	50	70	80	100	150
no	7	1	10	9	2	9	13	1	1	4	3
yes	0	0	3	4	0	6	7	0	0	1	7

```
## Is small data set, can read it in:
> conc <- c(0.1,0.5,1,10,20,30,50,70,80,100,150)
> no <- c(7,1,10,9,2,9,13,1,1,4,3)
> yes <- c(0,0,3,4,0,6,7,0,0,1,7)

## Need to create a variable (n) for the total trials:
> n <- no + yes

## Plot the probability of inhibition by the
## log of the probability of success.
## Note: need to add 0.5 to values of no and yes
## since you have 0 values of yes and the log
## of 0 is undefined

> plot(log(conc),log((yes+0.5)/(no+0.5)))
```




```
## We plot the logit of the observed proportions against log(conc).
## Concentrations are nearer to equally spaced on a scale of relative
## dose, rather than on a scale of dose, suggesting that it might be
## appropriate to work with log(conc). In order to allow plotting
## of cases where no = 0 or yes = 0, we add 0.5 to each count.
## The plot seems consistent with the use of log(conc)
## as the explanatory variable.

## Create the probability of success variable:
> p <- yes/n

## Need to weight the regression by the total successes
## and failures for each level of concentration

> inhibit.glm <- glm(p ~ I(log(conc)), family = binomial, weights = n)
> summary(inhibit.glm)
```

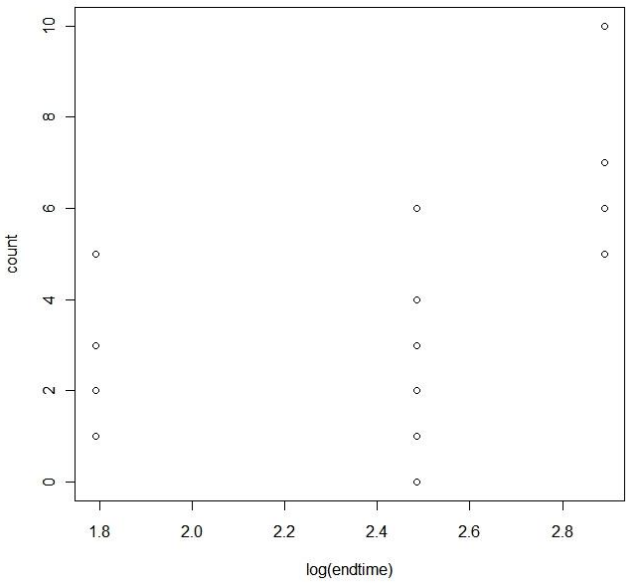
```
Call:
glm(formula = p ~ I(log(conc)), family = binomial, weights = n)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.2510  -1.0599  -0.5029   0.3152   1.3513

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -1.7659     0.5209  -3.390 0.000699 ***
I(log(conc))   0.3437     0.1440   2.387 0.016975 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 16.6834  on 10  degrees of freedom
Residual deviance:  9.3947  on  9  degrees of freedom
AIC: 29.994

Number of Fisher Scoring iterations: 4
```

4) The **R** data frame **ACF1** in the package **DAAG** consists of two columns: `count` and `endtime`. The first column contains the counts of simple aberrant foci (ACFs). These are aberrant aggregations of tube-like structures in the rectal end of 22 rat colons after administration of a dose of the carcinogen azoxymethane. Each rat was sacrificed after 6, 12 or 18 weeks. Create a scatterplot of `count` by (~) `endtime`.

   Run two **glm** models. The first specifies `count` (as the response variable) as predicted by `endtime` (the explanatory variable) and uses a `poisson` family for the distribution. Plot the results. Interpret the results. Then run a second model adding an `endtime^2` term to the right hand side to accommodate a possible quadratic effect. Plot the results. Interpret the results. Compare the two models with an `anova` table. Which model 'fits' better? Why?

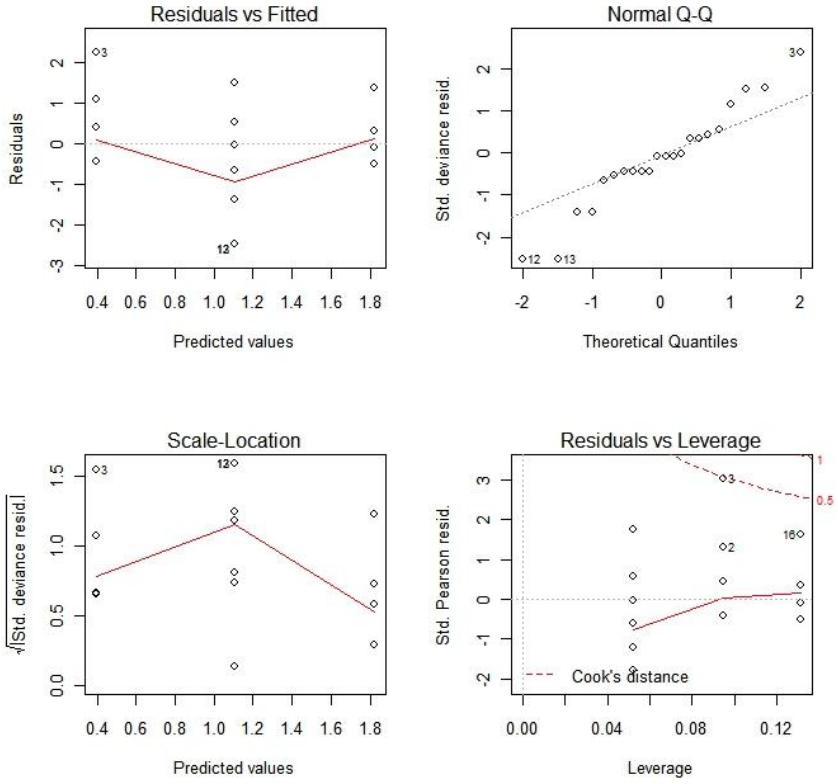
```
> install.packages("DAAG")  ## Might have to install package DAAG
> library("DAAG")
> data(ACF1)
> head(ACF1)

  count endtime
1     1       6
2     3       6
3     5       6
4     1       6
5     2       6
6     1       6
```

```
## Do a plot with log on x-axis to look at relationship:
> plot(count ~ log(endtime), data = ACF1)
```



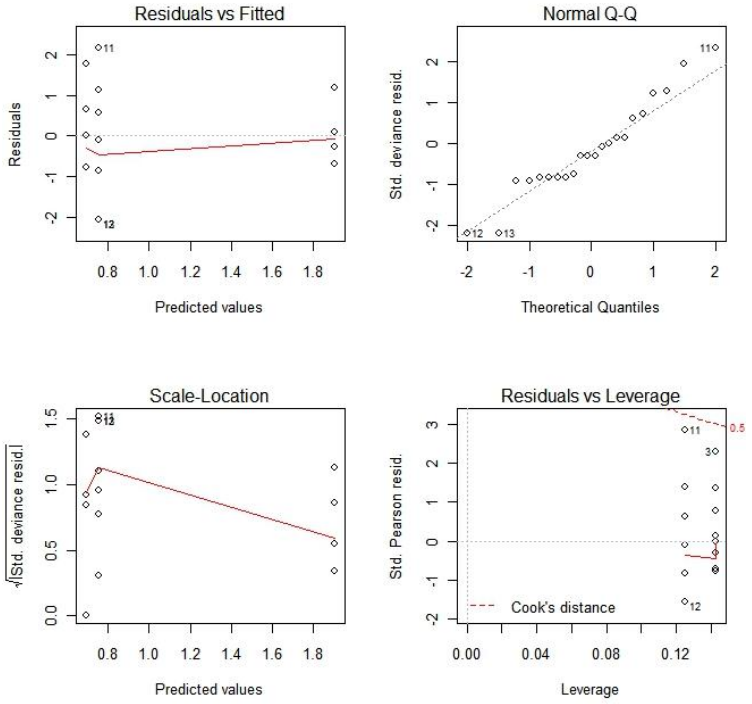

```
## Counts increase with time

## Run the glm for the simple linear model:
> ACF.glm <- glm(count ~ endtime, family = poisson, data = ACF1)
> summary(ACF.glm)

Call:
glm(formula = count ~ endtime, family = poisson, data = ACF1)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.46204  -0.47851  -0.07943   0.38159   2.26332

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.32152    0.40046  -0.803    0.422
endtime      0.11920    0.02642   4.511 6.44e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## We see that the relationship between count and
## endtime is highly significant

(Dispersion parameter for poisson family taken to be 1)

## Residual deviance is a little high but not overly so:
    Null deviance: 51.105  on 21  degrees of freedom
Residual deviance: 28.369  on 20  degrees of freedom
AIC: 92.209

Number of Fisher Scoring iterations: 5
```

```
## Let's look at the residual plots (however, does not make a lot of
## Sense to do this since only have a few predicted values.
> par(mfrow=c(2,2))
> plot(ACF.glm)
```




```
## Quadratic fit may be better, let's try it:

> ACF.glm2 <- glm(count~endtime+I(endtime^2),family=poisson,data = ACF1)
> summary(ACF.glm2)

Call:
glm(formula = count ~ endtime + I(endtime^2), family = poisson,
    data = ACF1)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0615  -0.7834  -0.2808   0.4510   2.1693

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.722364   1.092494   1.577    0.115
endtime      -0.262356   0.199685  -1.314    0.189
I(endtime^2)  0.015137   0.007954   1.903    0.057 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 51.105  on 21  degrees of freedom
Residual deviance: 24.515  on 19  degrees of freedom
AIC: 90.354

Number of Fisher Scoring iterations: 5

## Really not a lot of improvement in the Overdispersion, nor in AIC

## Let's again plot the residuals (again, kind of silly with the
## restricted range of residual values that we have
> par(mfrow=c(2,2))
> plot(ACF.glm2)
```




```
## Cannot tell anything from the residual plots.
## Let's run an anova table on the two glm outputs:
> anova(ACF.glm,ACF.glm2, test="Chisq")
Analysis of Deviance Table

Model 1: count ~ endtime
Model 2: count ~ endtime + I(endtime^2)
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1        20     28.369
2        19     24.515  1   3.8548    0.0496 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## The quadratic appears to fit better by the anova table.
```