LNL_HW_week5

Patrick Kelly

Saturday, February 14, 2015

This assignment helps understanding Poisson regression and Negative Binomial regression.

Recent outbreak of measles caused debated about necessity of vaccination.

The data file for this project contains CDC data on immunization coverage for MMR for all U.S. states and the national immunization rate.

The data for outbreaks in each state were simulated using the assumption that after vaccination 90% of vaccinated children become immuned.

Another assumption was made that 100% of not immunized people get infected when exposed to the virus.

The data for this project are in the file `MeaslesImmunizationCoverageAndOutbreaks.csv`

Read the data.

```
measles.data<-read.csv(file="C:/Users/Patrick/Documents/R/UChicago/Linear_NonLinear/MeaslesImmunization

measles.data<-as.data.frame(measles.data)
```

Fit Poisson Regression

```
measles.poisson.model<-glm(Outbreaks~Coverage,family=poisson,data=measles.data)
summary(measles.poisson.model)
```

```
##
## Call:
## glm(formula = Outbreaks ~ Coverage, family = poisson, data = measles.data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.51676  -0.53776   0.01017   0.62217   1.38757
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   5.5118     0.1746  31.560  < 2e-16 ***
## Coverage     -0.7828     0.1963  -3.987 6.69e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 52.549  on 53  degrees of freedom
## Residual deviance: 37.549  on 52  degrees of freedom
## AIC: 400.72
##
## Number of Fisher Scoring iterations: 3
```

```
names(measles.poisson.model)
```

```
##  [1] "coefficients"      "residuals"         "fitted.values"
##  [4] "effects"           "R"                 "rank"
##  [7] "qr"                "family"            "linear.predictors"
## [10] "deviance"          "aic"               "null.deviance"
## [13] "iter"              "weights"           "prior.weights"
## [16] "df.residual"       "df.null"           "y"
## [19] "converged"         "boundary"          "model"
## [22] "call"              "formula"           "terms"
## [25] "data"              "offset"            "control"
## [28] "method"            "contrasts"         "xlevels"
```

```
measles.poisson.model$linear.predictors
```

```
##        1        2        3        4        5        6        7        8
## 4.806506 4.815117 4.820597 4.822945 4.828425 4.810420 4.853476 4.806506
##        9       10       11       12       13       14       15       16
## 4.773626 4.767364 4.798677 4.798677 4.786935 4.817465 4.808854 4.794763
##       17       18       19       20       21       22       23       24
## 4.784586 4.811985 4.834688 4.836254 4.809637 4.773626 4.763449 4.833905
##       25       26       27       28       29       30       31       32
## 4.806506 4.803374 4.815900 4.845648 4.796329 4.818248 4.768146 4.790066
##       33       34       35       36       37       38       39       40
## 4.830774 4.783020 4.761884 4.821380 4.851910 4.829208 4.833122 4.786935
##       41       42       43       44       45       46       47       48
## 4.763449 4.822945 4.796329 4.819814 4.794763 4.790849 4.814334 4.848779
##       49       50       51       52       53       54
## 4.809637 4.878527 4.801809 4.817465 4.872264 5.079717
```

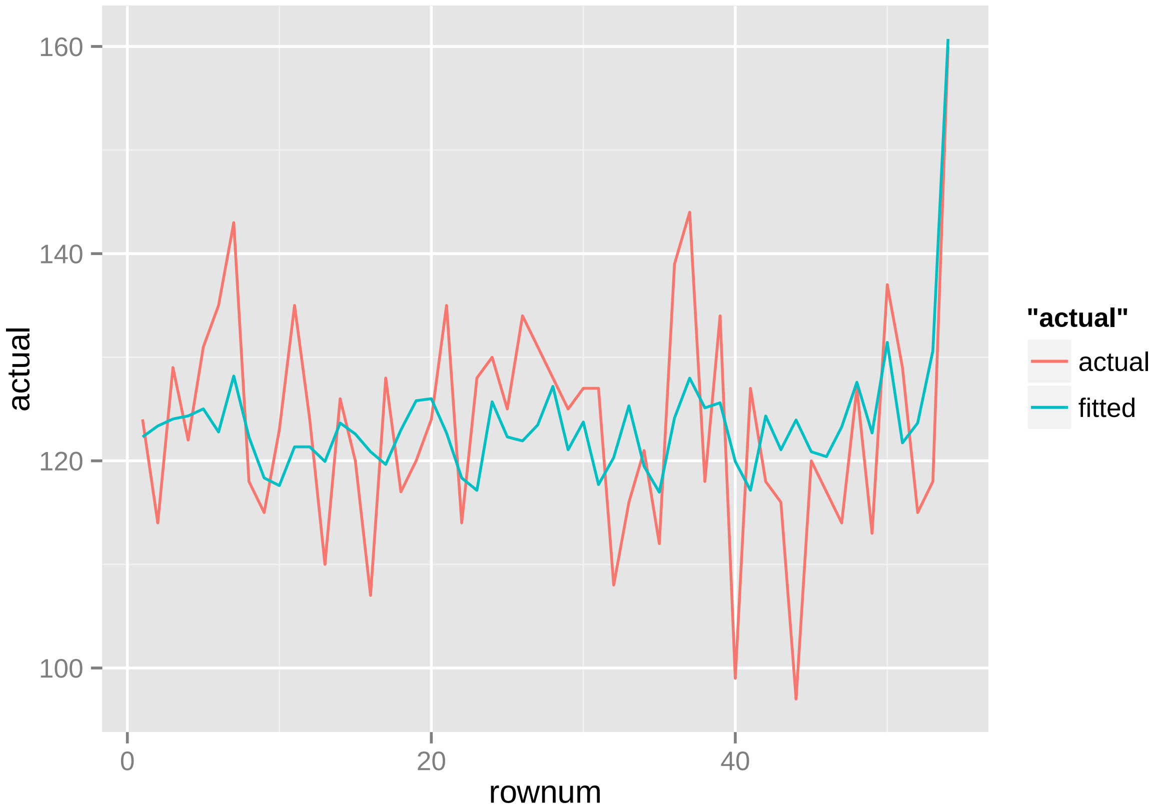
Compare the Outbreaks data with the fitted values.

```
library(ggplot2)

compare.results <- data.frame(rownum = c(1:length(measles.data$Outbreaks)), actual = measles.data$Outbre
compare.results
```

```
##    rownum actual   fitted
## 1       1    124 122.3035
## 2       2    114 123.3612
## 3       3    129 124.0391
## 4       4    122 124.3307
## 5       5    131 125.0139
## 6       6    135 122.7831
## 7       7    143 128.1852
## 8       8    118 122.3035
## 9       9    115 118.3476
## 10     10    123 117.6088
## 11     11    135 121.3498
## 12     12    124 121.3498
## 13     13    110 119.9332
## 14     14    126 123.6513
## 15     15    120 122.5911
## 16     16    107 120.8757
## 17     17    128 119.6518
## 18     18    117 122.9755
## 19     19    120 125.7993
## 20     20    124 125.9964
## 21     21    135 122.6871
## 22     22    114 118.3476
## 23     23    128 117.1493
## 24     24    130 125.7009
## 25     25    125 122.3035
## 26     26    134 121.9211
## 27     27    131 123.4578
## 28     28    128 127.1856
## 29     29    125 121.0651
## 30     30    127 123.7481
## 31     31    127 117.7009
## 32     32    108 120.3093
## 33     33    116 125.3079
## 34     34    121 119.4646
## 35     35    112 116.9660
## 36     36    139 124.1362
## 37     37    144 127.9847
## 38     38    118 125.1118
## 39     39    134 125.6025
## 40     40     99 119.9332
## 41     41    127 117.1493
## 42     42    118 124.3307
## 43     43    116 121.0651
## 44     44     97 123.9420
## 45     45    120 120.8757
## 46     46    117 120.4035
## 47     47    114 123.2647
## 48     48    127 127.5845
## 49     49    113 122.6871
## 50     50    137 131.4369
## 51     51    129 121.7304
## 52     52    115 123.6513
## 53     53    118 130.6163
## 54     54    160 160.7286
```

```
ggplot(compare.results, aes(rownum)) + geom_line(aes(y = actual, colour = "actual")) + geom_line(aes(y
```



```
#compare.results.long <- melt(compare.results, id="rownum")
#ggplot(data=compare.results.long,
#       aes(x=rownum, y=value, colour=variable)) +
#       geom_line()
```

Check that the link in our case is logarithmic.

```
log(measles.poisson.model$fitted.values)
```

```
##        1        2        3        4        5        6        7        8
## 4.806506 4.815117 4.820597 4.822945 4.828425 4.810420 4.853476 4.806506
##        9       10       11       12       13       14       15       16
## 4.773626 4.767364 4.798677 4.798677 4.786935 4.817465 4.808854 4.794763
##       17       18       19       20       21       22       23       24
## 4.784586 4.811985 4.834688 4.836254 4.809637 4.773626 4.763449 4.833905
##       25       26       27       28       29       30       31       32
## 4.806506 4.803374 4.815900 4.845648 4.796329 4.818248 4.768146 4.790066
##       33       34       35       36       37       38       39       40
## 4.830774 4.783020 4.761884 4.821380 4.851910 4.829208 4.833122 4.786935
##       41       42       43       44       45       46       47       48
## 4.763449 4.822945 4.796329 4.819814 4.794763 4.790849 4.814334 4.848779
##       49       50       51       52       53       54
## 4.809637 4.878527 4.801809 4.817465 4.872264 5.079717
```

Interpretation of the model.

What if coverage changes by 1%?

Then the change of intensity measured in percentage points is

```
(1-exp(measles.poisson.model$coef[2]*.01))*100
```

```
##  Coverage
## 0.7797844
```

Fit Negative Binomial regression

The standard `glm()` function does not have functionality to fit negative binomial distribution. But the package MASS does have it.

```
library(MASS)
```

Learn how to use the package to fit negative binomial regression. Fit the model. Interpret the results. Compare the fit with the fitted Poisson regression model.

First, fit the negative binomial where the disperions (k) = 1

```
measles.negbinom.model <- glm(Outbreaks~Coverage, negative.binomial(1), data=measles.data)
measles.negbinom.model
```

```
##
## Call:  glm(formula = Outbreaks ~ Coverage, family = negative.binomial(1),
##     data = measles.data)
##
## Coefficients:
## (Intercept)     Coverage
##      5.5110      -0.7819
##
## Degrees of Freedom: 53 Total (i.e. Null);  52 Residual
## Null Deviance:       0.4218
## Residual Deviance: 0.3082    AIC: 632.5
```

Now, fit the negative binomial but let the model find the optimal dispersion

```
measles.negbinom.model2 <- glm.nb(Outbreaks~Coverage, data=measles.data)
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
measles.negbinom.model2
```

```
## 
## Call:  glm.nb(formula = Outbreaks ~ Coverage, data = measles.data, init.theta = 2934552.331, 
##     link = log)
## 
## Coefficients:
## (Intercept)     Coverage  
##      5.5118      -0.7828  
## 
## Degrees of Freedom: 53 Total (i.e. Null);  52 Residual
## Null Deviance:       52.55 
## Residual Deviance: 37.55     AIC: 402.7
```

```r
summary(measles.negbinom.model2)
```

```
## 
## Call:
## glm.nb(formula = Outbreaks ~ Coverage, data = measles.data, init.theta = 2934552.331, 
##     link = log)
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -2.51671  -0.53774   0.01017   0.62215   1.38754  
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)   5.5118     0.1747  31.559  < 2e-16 ***
## Coverage     -0.7828     0.1963  -3.987 6.69e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for Negative Binomial(2934552) family taken to be 1)
## 
##     Null deviance: 52.547  on 53  degrees of freedom
## Residual deviance: 37.547  on 52  degrees of freedom
## AIC: 402.72
## 
## Number of Fisher Scoring iterations: 1
## 
## 
##               Theta:  2934552 
##           Std. Err.:  59526304 
## Warning while fitting theta: iteration limit reached 
## 
##  2 x log-likelihood:  -396.725
```

```r
#compare AIC values of both binomial models --- the second model, using glm.nb, is better
AIC(measles.negbinom.model,measles.negbinom.model2)
```

```
##                         df      AIC
## measles.negbinom.model   2 632.4603
## measles.negbinom.model2  3 402.7247
```

```r
#now compare AIC values btwn the Poisson model and neg. binom. model
AIC(measles.poisson.model,measles.negbinom.model2)
```

```
##                         df      AIC
## measles.poisson.model    2 400.7239
## measles.negbinom.model2  3 402.7247
```

Interpret the results. Compare the fit with the fitted Poisson regression model

Initially when observing the output from the Poisson model, we notice that the null deviance is not greater than the null degrees of freedom. This is a first quick check that suggests that there is not overdispersion. But we explore further through fitting the negative binomial model anyway. Through the use of `glm.nb` we notice that it does not converge (iteration limit reached) - this also suggests that there is not overdispersion in the distribution.

In further evaluating The negative binomial model we observe a significant negative correlation between the coverage and outbreaks (coeff = -.78), similar to the Poisson model. Both model fits are producing very similiar results but all other evidence suggests that the Poisson distribution / model is a better fit. This is further confirmed when comparing both models AIC values - again they are very simliar but the Poisson model is slightly lower (400.72 vs 402.72).

In conclusion, the Poisson model is the most appropriate fit and the Coverage is a significant indep. variable w/ coeff -.783.