Generalized Linear Model: Treatment Coding

Ruben Eduardo Montano Claure

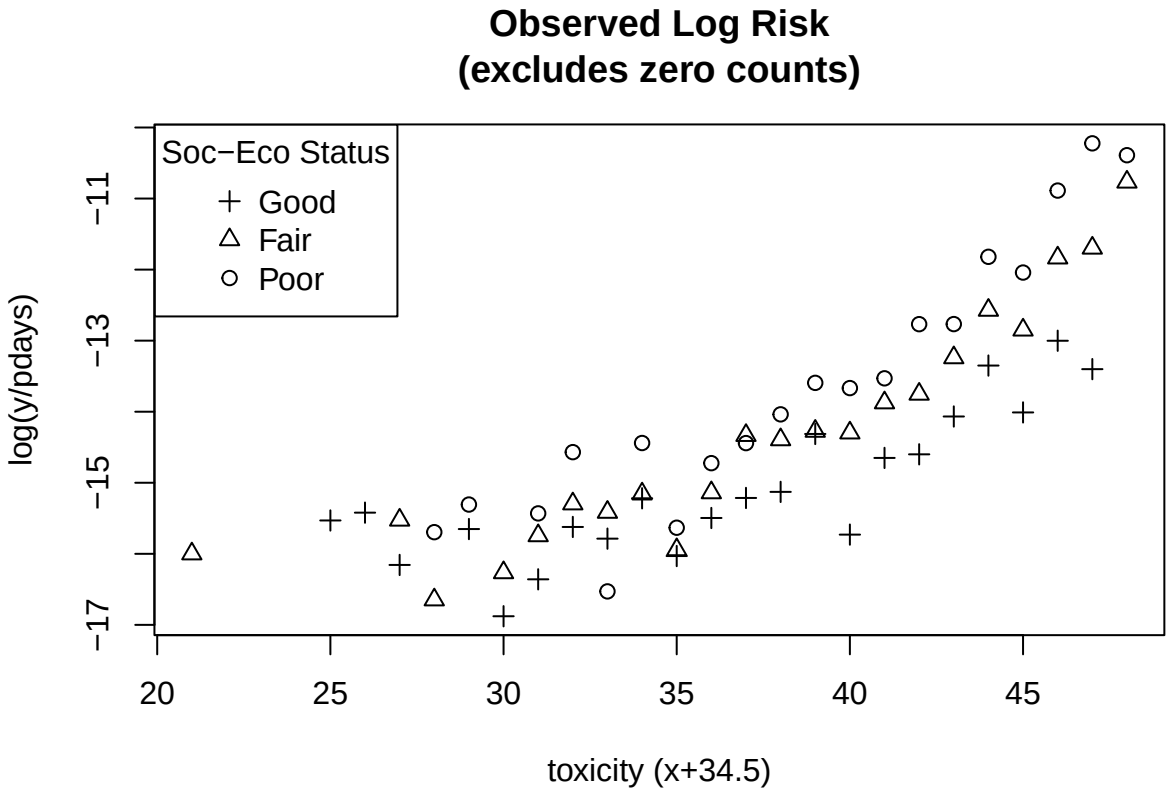
Things to consider before analysis:

The data used here are part of a study of the risk of disease related to the toxicity of a certain substance in various areas and how this risk varies among groups classified according to a measure of health for the areas.

- y_i, disease count in area i (response)
- x_i, measure of the toxicity in area i (no units given) re-centered by subtracting mean toxicity (34.5) (covariate)
- $status_i$, health index for area i (factor covariate with three levels: G is good, F is fair, P is poor)
- $pday_i$, total person-days of exposure to toxicity xi in area i (offset).

Uploading & Visualizying Data

```
dataset<- readRDS(file="df.RDS")
plot(log(y/pdays) ~ I(x+34.5), data=dataset, pch=as.numeric(status),
     xlab="toxicity (x+34.5)", main=c("Observed Log Risk\n(excludes zero counts)"))
legend("topleft", legend=c("Good", "Fair", "Poor"), pch=c(3,2,1), title="Soc-Eco Status")
```



Formula for a generic response (y), continuous covariate (x), and factor (S)

$$y \sim x + I(x^2) + S + S:x + S:I(x^2)$$

$$\log(\mu(x, \text{status}, \beta)) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 f + \beta_4 g + \beta_5 fx + \beta_6 gx + \beta_7 fx^2 + \beta_8 gx^2$$

The coded variable $f = 1$ indicates status $= F(f = 0 for status = P or status = G)$, $g = 1$ indicates status $= G(g = 0 for status = P or status = F)$ and x is the toxicity minus mean toxicity 34.5.

```
# Fitting a GLM
poismod <- glm(y ~ offset(log(pdays)) + x + I(x^2) + status + status:x + status:I(x^2),
    family = "poisson", data = dataset)


summary(poismod)
```

```
##
## Call:
## glm(formula = y ~ offset(log(pdays)) + x + I(x^2) + status +
##     status:x + status:I(x^2), family = "poisson", data = dataset)
##
## Coefficients:
##                   Estimate Std. Error  z value Pr(>|z|)
## (Intercept)    -15.058719   0.124266 -121.181  < 2e-16 ***
```

```
## x                0.210148   0.030393    6.914  4.7e-12 ***
## I(x^2)           0.010783   0.002855    3.778 0.000158 ***
## statusF         -0.261908   0.152014   -1.723 0.084903 .
## statusG         -0.605758   0.168255   -3.600 0.000318 ***
## x:statusF       -0.029103   0.035607   -0.817 0.413741
## x:statusG       -0.090205   0.037776   -2.388 0.016945 *
## I(x^2):statusF  -0.002776   0.003430   -0.809 0.418342
## I(x^2):statusG  -0.003519   0.003896   -0.903 0.366368
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1195.68  on 83  degrees of freedom
## Residual deviance:  113.42  on 75  degrees of freedom
## AIC: 383.64
##
## Number of Fisher Scoring iterations: 5
```

1. Fitted (approximate) risk models for the poor health status group, for the fair group, and for the good group.

```
summary(poismod)$coefficients
```

```
##                     Estimate  Std. Error      z value     Pr(>|z|)
## (Intercept)    -15.058719444 0.124266199 -121.1811383 0.000000e+00
## x                0.210147908 0.030393126    6.9143237 4.701007e-12
## I(x^2)           0.010783151 0.002854503    3.7775933 1.583512e-04
## statusF         -0.261908085 0.152014316   -1.7229172 8.490349e-02
## statusG         -0.605757892 0.168254897   -3.6002393 3.179244e-04
## x:statusF       -0.029102913 0.035607380   -0.8173281 4.137409e-01
## x:statusG       -0.090204580 0.037775687   -2.3879005 1.694493e-02
## I(x^2):statusF  -0.002775707 0.003429758   -0.8093012 4.183419e-01
## I(x^2):statusG  -0.003519122 0.003895863   -0.9032972 3.663682e-01
```

```
#(Intercept) = Beta_0
#x = Beta_1
#I(x^2) = Beta_2
#statusF = Beta_3
#statusG = Beta_4
#x:statusF = Beta_5
#x:statusG = Beta_6
#I(x^2):statusF:= Beta_7
#I(x^2):statusG = Beta_8
```

Summary from above is used to replace the coefficient values from $\beta_0 - \beta_8$ for Poor, Fair and Good status.

2. For Poor (P), substitute $f = 0$ and $g = 0$ from the equation:

$$\log(\mu(x, \text{status}, \beta)) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 f + \beta_4 g + \beta_5 fx + \beta_6 gx + \beta_7 fx^2 + \beta_8 gx^2$$

to obtain the following:

$$\log(\mu_P(x)) = \beta_0 + \beta_1 x + \beta_2 x^2$$

- Exponentiation and replacing with coefficient values:

$$\begin{aligned}\mu_P(x) &= \exp(\beta_0 + \beta_1 x + \beta_2 x^2) \\ &= \exp(-15.058719444 + 0.210147908x + 0.010783151x^2)\end{aligned}$$

3. For Poor (P), substitute $f = 1$ and $g = 0$ from the equation:

$$\log(\mu(x, \text{status}, \beta)) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 f + \beta_4 g + \beta_5 fx + \beta_6 gx + \beta_7 fx^2 + \beta_8 gx^2$$

to obtain the following:

$$\log(\mu_F(x)) = (\beta_0 + \beta_3) + x(\beta_1 + \beta_5) + x^2(\beta_2 + \beta_7)$$

- Exponentiation and replacing with coefficient values:

$$\begin{aligned}\mu_F(x) &= \exp((\beta_0 + \beta_3) + x(\beta_1 + \beta_5) + x^2(\beta_2 + \beta_7)) \\ &= \exp(-15.32063 + 0.18105x + 0.00801x^2)\end{aligned}$$

4. For Poor (P), substitute $f = 0$ and $g = 1$ from the equation:

$$\log(\mu(x, \text{status}, \beta)) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 f + \beta_4 g + \beta_5 fx + \beta_6 gx + \beta_7 fx^2 + \beta_8 gx^2$$

to obtain the following:

$$\log(\mu_G(x)) = (\beta_0 + \beta_4) + x(\beta_1 + \beta_6) + x^2(\beta_2 + \beta_8)$$

- Exponentiation and replacing with coefficient values:

$$\begin{aligned}\mu_G(x) &= \exp((\beta_0 + \beta_4) + x(\beta_1 + \beta_6) + x^2(\beta_2 + \beta_8)) \\ &= \exp(-15.66448 + 0.11994x + 0.00726x^2)\end{aligned}$$

5. Different models to see which terms can be ommited

```
# Model 1: Only a single curve for all status groups

model_1 <- glm(y ~ offset(log(pdays)) + x + I(x^2),
               family = "poisson", data = dataset)

# Model 2: allows curves to have their own 'intercept' parameter for each status group
model_2 <- glm(y ~ offset(log(pdays)) + x + I(x^2) + status,
               family = "poisson", data = dataset)

# Model 3: one that allows the curves to have their own intercept and linear parameters for each status
model_3 <- glm(y ~ offset(log(pdays)) + x + I(x^2) + status + status:x,
               family = "poisson", data = dataset)

#Full Model
```

```r
model_4 <- poismod # from above



# Test (LRT)
anova(model_1, model_2, model_3, model_4, test = "LRT")
```

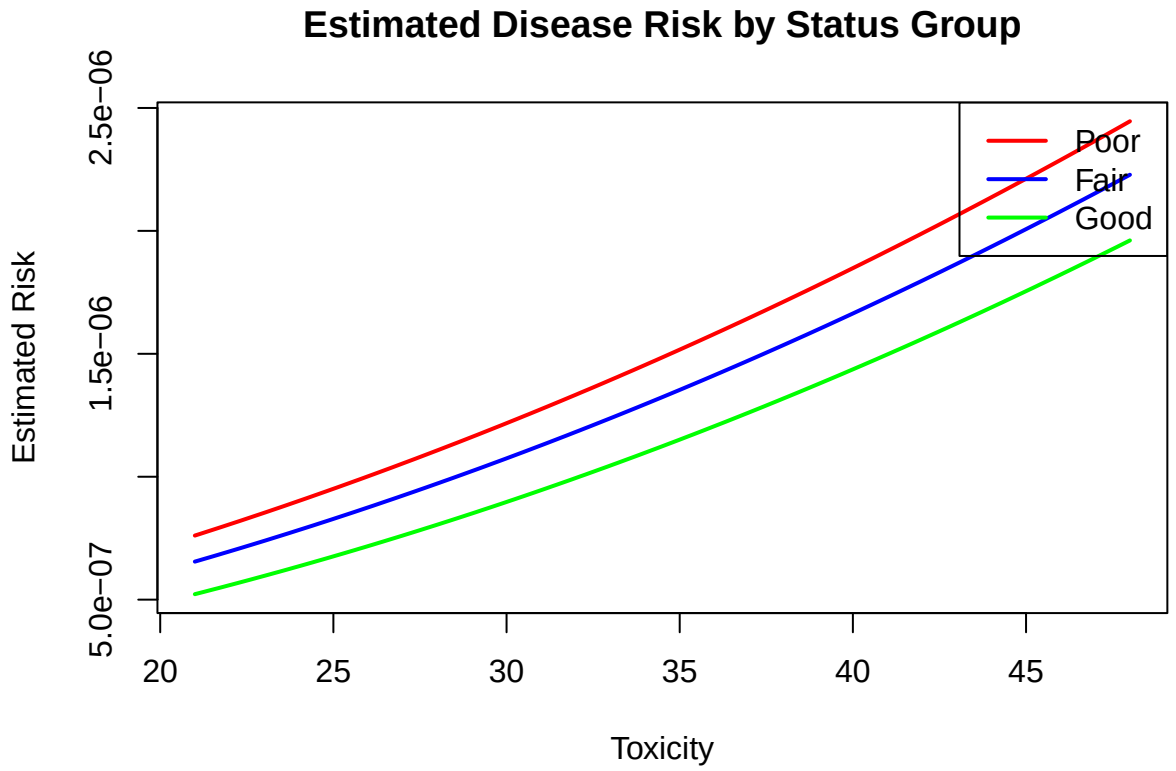
```
## Analysis of Deviance Table
##
## Model 1: y ~ offset(log(pdays)) + x + I(x^2)
## Model 2: y ~ offset(log(pdays)) + x + I(x^2) + status
## Model 3: y ~ offset(log(pdays)) + x + I(x^2) + status + status:x
## Model 4: y ~ offset(log(pdays)) + x + I(x^2) + status + status:x + status:I(x^2)
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        81     363.73
## 2        79     138.64  2  225.086 < 2.2e-16 ***
## 3        77     114.31  2   24.337 5.192e-06 ***
## 4        75     113.42  2    0.884    0.6427
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since Model 4 does not improve over Model 3, remove the quadratic terms from model 4 (full model), leaving to the best model "Model 3". This conclusion is based on the Anova results from above.

6. Using model 3

```r
# Re-centered (actual values)
new_x <- (dataset$x + 34.5)
# Create grid of toxicity values
x.grid <- seq(min(new_x), max(new_x), length=200)

# Use median person-days for predict()
newd <- data.frame(pdays = rep(median(dataset$pdays), 200), x = x.grid)

# Create status factor levels for each group
statusP <- factor(rep("P", 200), levels = levels(dataset$status))
statusF <- factor(rep("F", 200), levels = levels(dataset$status))
statusG <- factor(rep("G", 200), levels = levels(dataset$status))

# Generate new datasets for prediction
newP <- cbind.data.frame(newd, status = statusP)
newF <- cbind.data.frame(newd, status = statusF)
newG <- cbind.data.frame(newd, status = statusG)

# Using the predict function which will give estimated mean counts, E(y | x, status)
predP <- predict(model_3, newdata = newP)
predF <- predict(model_3, newdata = newF)
predG <- predict(model_3, newdata = newG)

# Dividing out pdayi values from the predicted values, to get the requested estimated rate/risk
riskP <- predP / newP$pdays
riskF <- predF / newF$pdays
```

```r
riskG <- predG / newG$pdays

# Plot the estimated risk curves
plot(x.grid, riskP, type = "l", col = "red", lwd = 2, ylim = range(c(riskP, riskF, riskG)),
     xlab = "Toxicity", ylab = "Estimated Risk", main = "Estimated Disease Risk by Status Group")
lines(x.grid, riskF, col = "blue", lwd = 2)
lines(x.grid, riskG, col = "green", lwd = 2)

legend("topright", legend = c("Poor", "Fair", "Good"), col = c("red", "blue", "green"), lwd = 2)
```

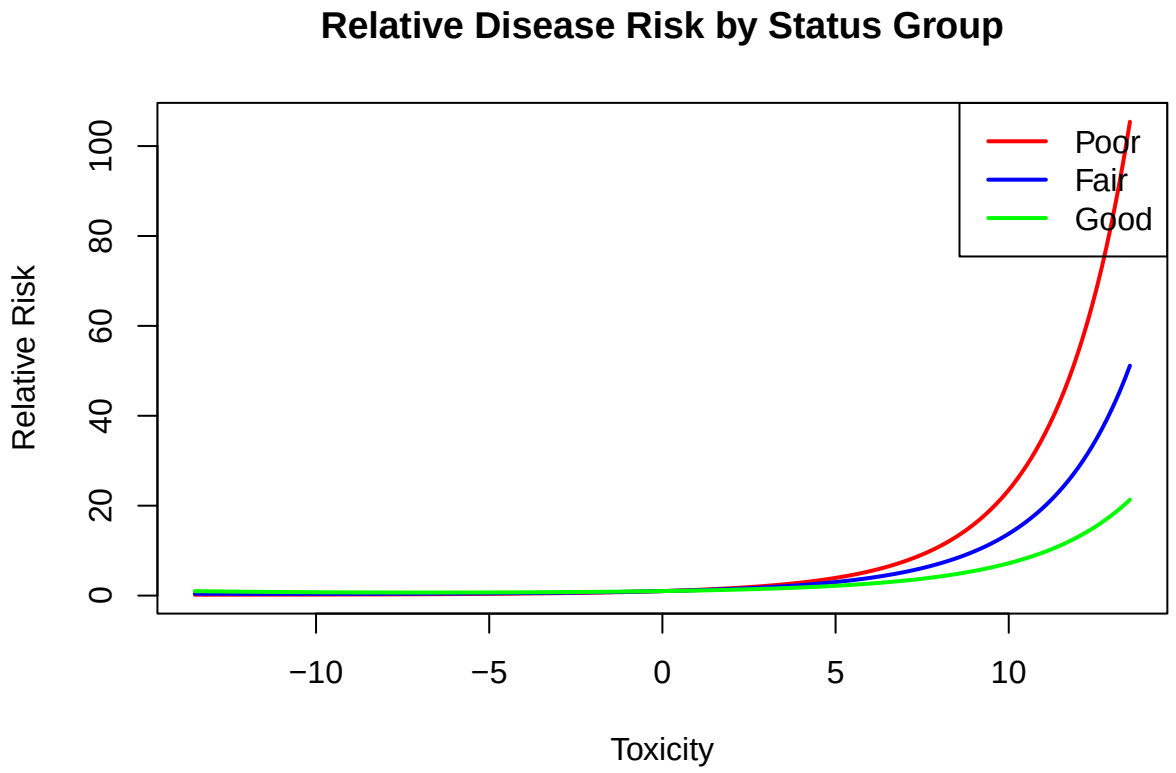


7. Using Relative Risk instead (RR) to visualize the plots better

```r
# Extract baseline risk for each status group at x = 0
baseline_risk_P <- exp(coef(model_3)["(Intercept)"])
baseline_risk_F <- exp(coef(model_3)["(Intercept)"] + coef(model_3)["statusF"])
baseline_risk_G <- exp(coef(model_3)["(Intercept)"] + coef(model_3)["statusG"])
```

```r
# Grid of toxicity values, similarly as above
x.grid <- seq(min(dataset$x), max(dataset$x), length=200)

# Compute relative risk for each group
RR_P <- exp(coef(model_3)["x"] * x.grid + coef(model_3)["I(x^2)"] * x.grid^2)
RR_F <- exp((coef(model_3)["x"] + coef(model_3)["x:statusF"]) * x.grid +
            coef(model_3)["I(x^2)"] * x.grid^2)
```

```r
RR_G <- exp((coef(model_3)["x"] + coef(model_3)["x:statusG"]) * x.grid +
            coef(model_3)["I(x^2)"] * x.grid^2)


# Plot relative risk curves
plot(x.grid, RR_P, type = "l", col = "red", lwd = 2, ylim = range(c(RR_P, RR_F, RR_G)),
     xlab = "Toxicity", ylab = "Relative Risk", main = "Relative Disease Risk by Status Group")
lines(x.grid, RR_F, col = "blue", lwd = 2)
lines(x.grid, RR_G, col = "green", lwd = 2)

# Add a legend
legend("topright", legend = c("Poor", "Fair", "Good"), col = c("red", "blue", "green"), lwd = 2)
```



8. Computing (approximating) a 95% confidence intervals for the ratio of the risk (RR) of the good status group over the risk of poor status group, each at an actual toxicity of 45 (x = 45 − 34.5 = 10.5).

$$RR_{G/P} = \frac{RR_G(x)}{RR_P(x)} = \frac{\exp((\beta_1 + \beta_6)x + \beta_2 x^2)}{\exp(\beta_1 x + \beta_2 x^2)} = \exp(\beta_6 x)$$

```r
library(gmodels)
# Defining the toxicity level at x = 10.5
x_value <- 10.5

# Computing RR ratio using model coefficients
```

```r
# beta 6 (from above) -> x:statusG
log_RR_ratio <- coef(model_3)["x:statusG"] * x_value
RR_ratio <- exp(log_RR_ratio)  # Exponentiation to get the risk ratio

# Use `estimable()` to get confidence interval for log(RR)

ci_log_RR <- estimable(model_3, cm = c("x:statusG" = x_value), conf.int = 0.95)

# Extract numeric confidence interval values (log scale)
ci_values_log <- c(ci_log_RR$Lower.CI, ci_log_RR$Upper.CI)

# Convert log CI to RR CI by exponentiation
ci_RR <- exp(ci_values_log)

# Print results
cat("Estimated RR Ratio (Good/Poor) at x = 10.5:", RR_ratio, "\n")
```

```
## Estimated RR Ratio (Good/Poor) at x = 10.5: 0.2886112
```

```r
cat("95% CI for RR Ratio:", ci_RR, "\n")
```

```
## 95% CI for RR Ratio: 0.1734671 0.4801858
```

9. nlWaldTest::nlConfint to implement the delta method to infer this non-linear function of interest. Report code/output.

```r
names(coef(model_3))
```

```
## [1] "(Intercept)" "x"           "I(x^2)"      "statusF"     "statusG"
## [6] "x:statusF"   "x:statusG"
```

```r
# coefficient 7 is x:statusG, this will be used in the nlWaldTest
```

```r
# Defining the toxicity level at x = 10.5
x_value <- 10.5


# Compute confidence interval using Delta Method
ci_RR_delta <- nlWaldTest::nlConfint(model_3, texts = c("exp(b[7] * 10.5)"))


# Print Delta Method CI
print(ci_RR_delta)
```

```
##                       value     2.5 %    97.5 %
## exp(b[7] * 10.5) 0.2886112 0.1434594 0.4337631
```

Comments for Research Questions 8 and 9

1. Estimated RR: The estimated RR of the good status group compared to the poor status group at toxicity 45 (x = 10.5) is 0.2886 using both gmodels::estimiable and nlWaldTest::nlConfint. This means that individuals in the good status group have approximately 28.86% of the risk of disease compared to those in the poor status group at toxicity level 10.5.

2. Confidence Intervals (CI):

- Using gmodels::estimable: The 95% confidence interval for the RR is [0.1735, 0.4802]
- Using nlWaldTest::nlConfint (Delta Method): The 95% CI is [0.1435, 0.4338]
- Both methods provide similar confidence intervals, indicating a reasonably precise estimate

3. Interpretation:

- Since the confidence intervals do not include 1, one can infer that there is a statistically significant lower risk for the good status group compared to the poor status group at this toxicity level (10.5)
- The intervals suggest some uncertainty, but they reinforce that the good status group consistently has a lower disease risk than the poor status group.