Generalized Linear Model: Binomial & Quasi-binomial

Ruben Eduardo Montano Claure

The data here are counts of surviving trout eggs out of a total number of eggs buried at one of five stream locations and retrieved and counted for surviving eggs at four different time periods. See help(troutegg, package="faraway") for more information on these data.

Uploading Data

```
data(troutegg, package="faraway")
str(troutegg)
```

```
## 'data.frame':    20 obs. of  4 variables:
##  $ survive : int  89 106 119 104 49 94 91 100 80 11 ...
##  $ total   : int  94 108 123 104 93 98 106 130 97 113 ...
##  $ location: Factor w/ 5 levels "1","2","3","4",..: 1 2 3 4 5 1 2 3 4 5 ...
##  $ period  : Factor w/ 4 levels "4","7","8","11": 1 1 1 1 1 2 2 2 2 2 ...
```

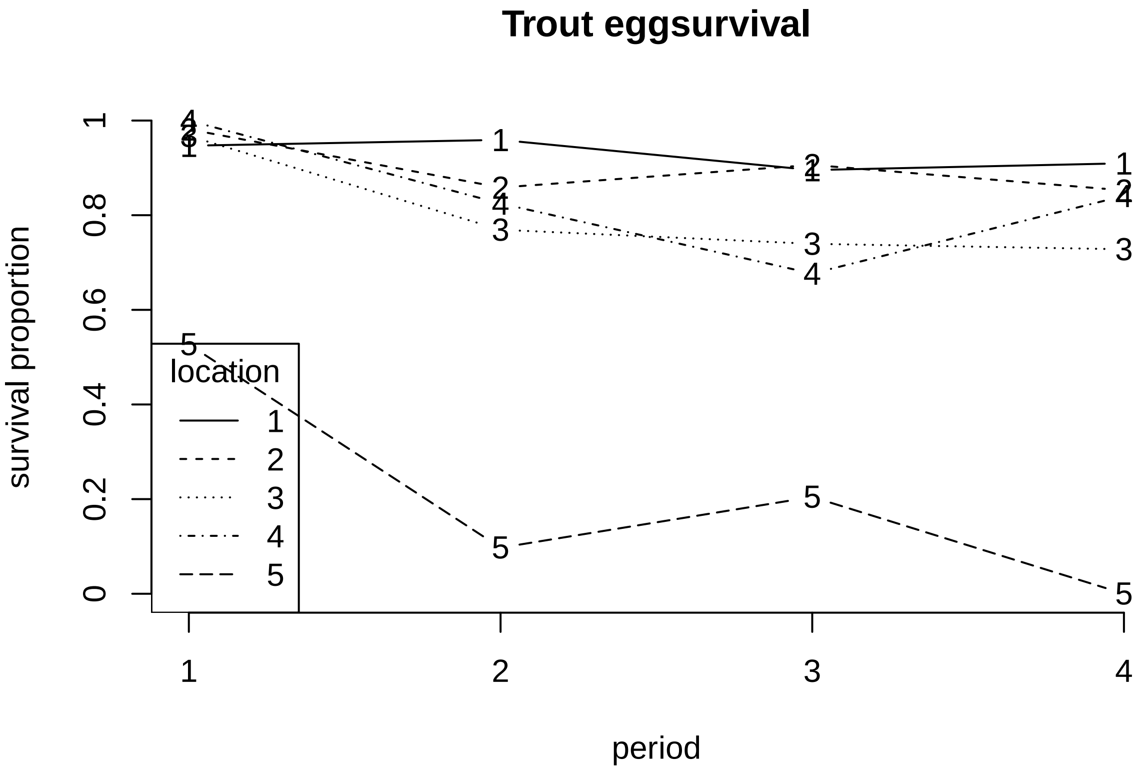
Table

```
ftable(xtabs(cbind(survive,total) ~location+period,troutegg))
```

```
##                  survive total
## location period
## 1        4            89    94
##          7            94    98
##          8            77    86
##          11          141   155
## 2        4           106   108
##          7            91   106
##          8            87    96
##          11          104   122
## 3        4           119   123
##          7           100   130
##          8            88   119
##          11           91   125
## 4        4           104   104
##          7            80    97
##          8            67    99
##          11          111   132
## 5        4            49    93
##           7             11   113
##           8             18    88
##           11             0   138
```

Plot

```
plot(survive/total ~as.numeric(period),data=troutegg,subset=troutegg$location==1,pch="1",type="b",
     ylim=c(0,1), ylab="survival proportion", xlab="period",main="Trout eggsurvival",axes=FALSE)
axis(side=1, at=1:4,labels=1:4)
attach(troutegg)
axis(side=2, at=pretty(survive/total),
labels=pretty(survive/total))
detach(troutegg)
lines(survive/total ~as.numeric(period),data=troutegg,subset=troutegg$location==2,type="b",pch="2",lty=2
lines(survive/total ~as.numeric(period),data=troutegg,subset=troutegg$location==3,type="b",pch="3",lty=3
lines(survive/total ~as.numeric(period),data=troutegg,subset=troutegg$location==4,type="b",pch="4",lty=4
lines(survive/total ~as.numeric(period),data=troutegg,subset=troutegg$location==5,type="b",pch="5",lty=5
legend("bottomleft",legend=1:5,lty=1:5,title="location")
```



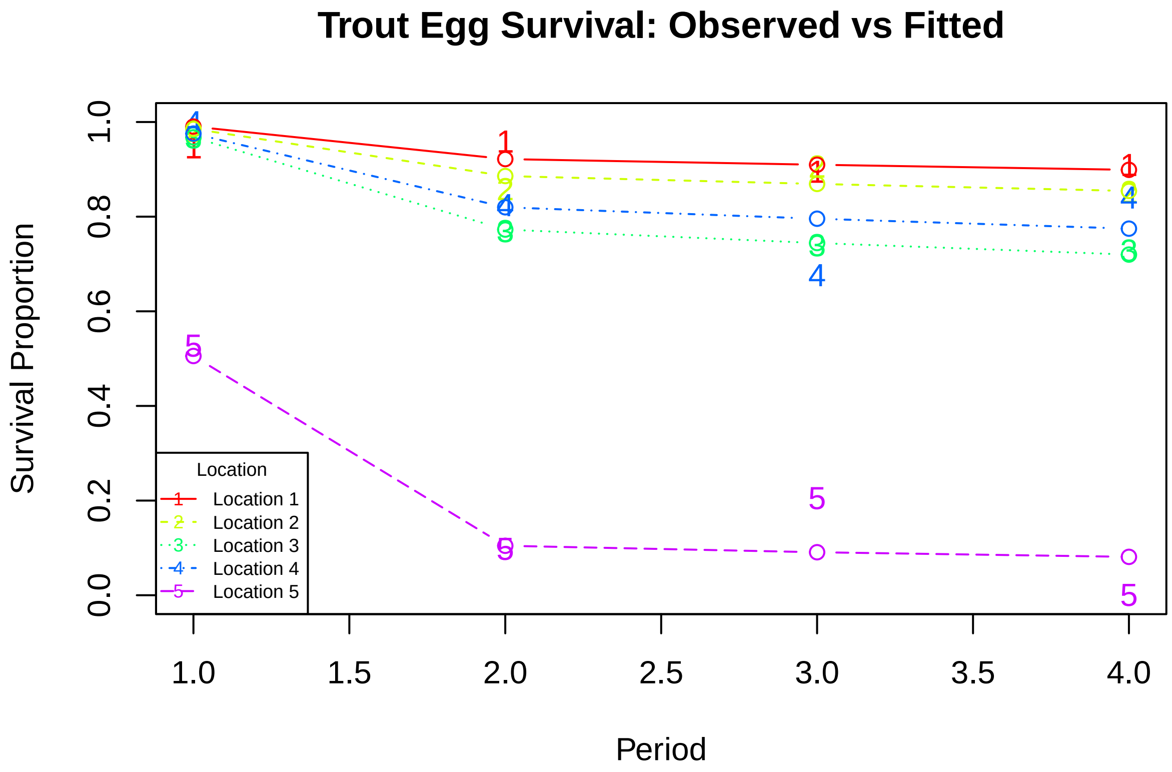
1. Binomial GLM with logit link using a linear predictor of the location and period factors

```r
binGLM <- glm(survive/total ~ location + period,
              weights = total,
              family = binomial(link = "logit"),
              data = troutegg)

summary(binGLM)
```

```
##
## Call:
## glm(formula = survive/total ~ location + period, family = binomial(link = "logit"),
##     data = troutegg, weights = total)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   4.6358     0.2813  16.479  < 2e-16 ***
## location2    -0.4168     0.2461  -1.694   0.0903 .
## location3    -1.2421     0.2194  -5.660 1.51e-08 ***
## location4    -0.9509     0.2288  -4.157 3.23e-05 ***
## location5    -4.6138     0.2502 -18.439  < 2e-16 ***
## period7      -2.1702     0.2384  -9.103  < 2e-16 ***
## period8      -2.3256     0.2429  -9.573  < 2e-16 ***
## period11     -2.4500     0.2341 -10.466  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1021.469  on 19  degrees of freedom
## Residual deviance:   64.495  on 12  degrees of freedom
## AIC: 157.03
##
## Number of Fisher Scoring iterations: 5
```

2. Scatterplot of empirical proportions vs. location and period

```r
troutegg$fitted <- fitted(binGLM)

plot(survive/total ~ as.numeric(period), data = troutegg, type = "n",
     ylim = c(0, 1), ylab = "Survival Proportion", xlab = "Period",
     main = "Trout Egg Survival: Observed vs Fitted")

# Colors and line types
cols <- rainbow(length(unique(troutegg$location)))

# Plotting observed points
for (i in 1:length(unique(troutegg$location))) {
  loc_data <- subset(troutegg, location == unique(location)[i])
  points(survive/total ~ as.numeric(period), data = loc_data,
         pch = as.character(i), col = cols[i])
}

# Adding fitted probabilities with lines within each location
for (i in 1:length(unique(troutegg$location))) {
  loc_data <- subset(troutegg, location == unique(location)[i])
  ord <- order(loc_data$period)
  lines(loc_data$fitted[ord] ~ as.numeric(loc_data$period[ord]),
        type = "b", lty = i, col = cols[i])
}

legend("bottomleft", legend = paste("Location", unique(troutegg$location)),
       col = cols, pch = as.character(1:length(cols)), lty = 1:length(cols),
       title = "Location", cex = 0.6)
```



The numbers labeled in the plot (1–5) represent the observed proportions of survival, while the open circles are the model-predicted probabilities. The connecting lines between the fitted points show the model's trend for each location.

Location 1: The model slightly overestimates survival in period 1 and underestimates in period 2.

Location 2: Shows slight overestimation in period 2 and underestimation in period 3.

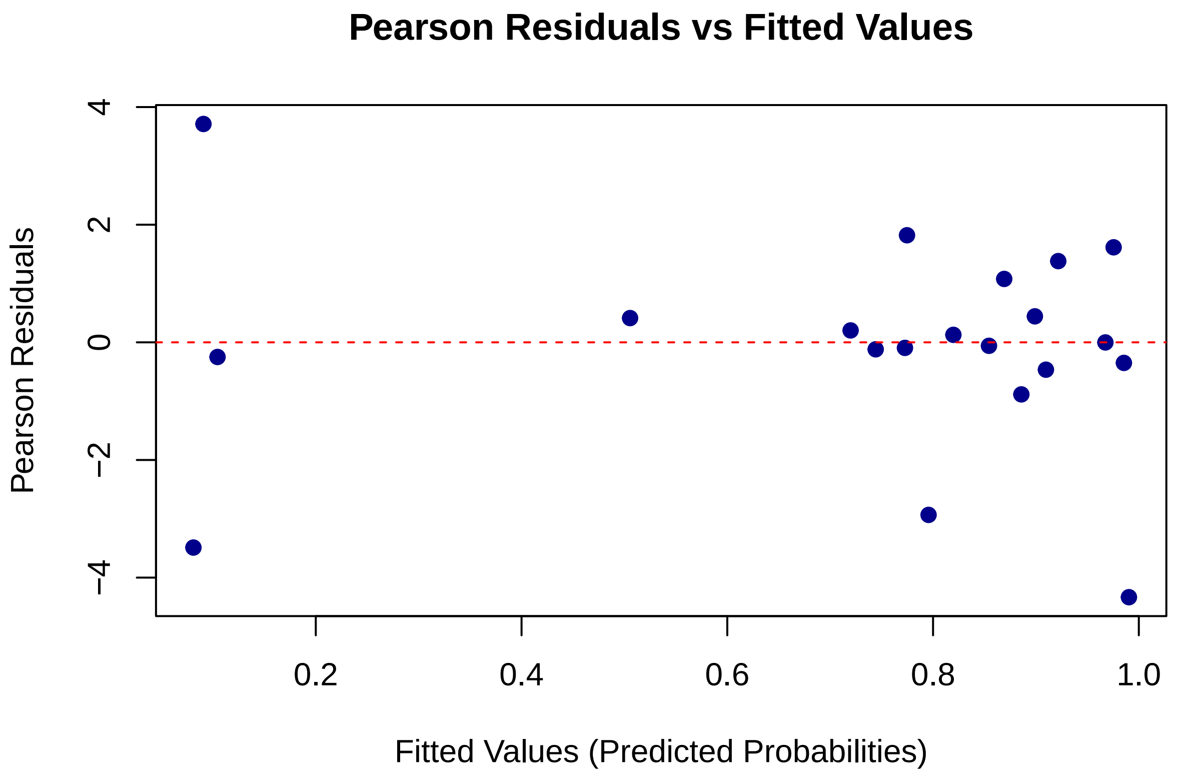
Location 3: The model fits well across all periods.

Location 4: Overestimation occurs in period 3, followed by underestimation in period 4.

Location 5: The model underestimates in period 3 and overestimates in period 4.

3. ploting Pearson residuals versus fitted values.

```r
pearson_res <- residuals(binGLM, type = "pearson")

fitted_vals <- fitted(binGLM)

# Plot
plot(fitted_vals, pearson_res,
     xlab = "Fitted Values (Predicted Probabilities)",
     ylab = "Pearson Residuals",
     main = "Pearson Residuals vs Fitted Values",
     pch = 19, col = "darkblue")
abline(h = 0, lty = 2, col = "red")
```



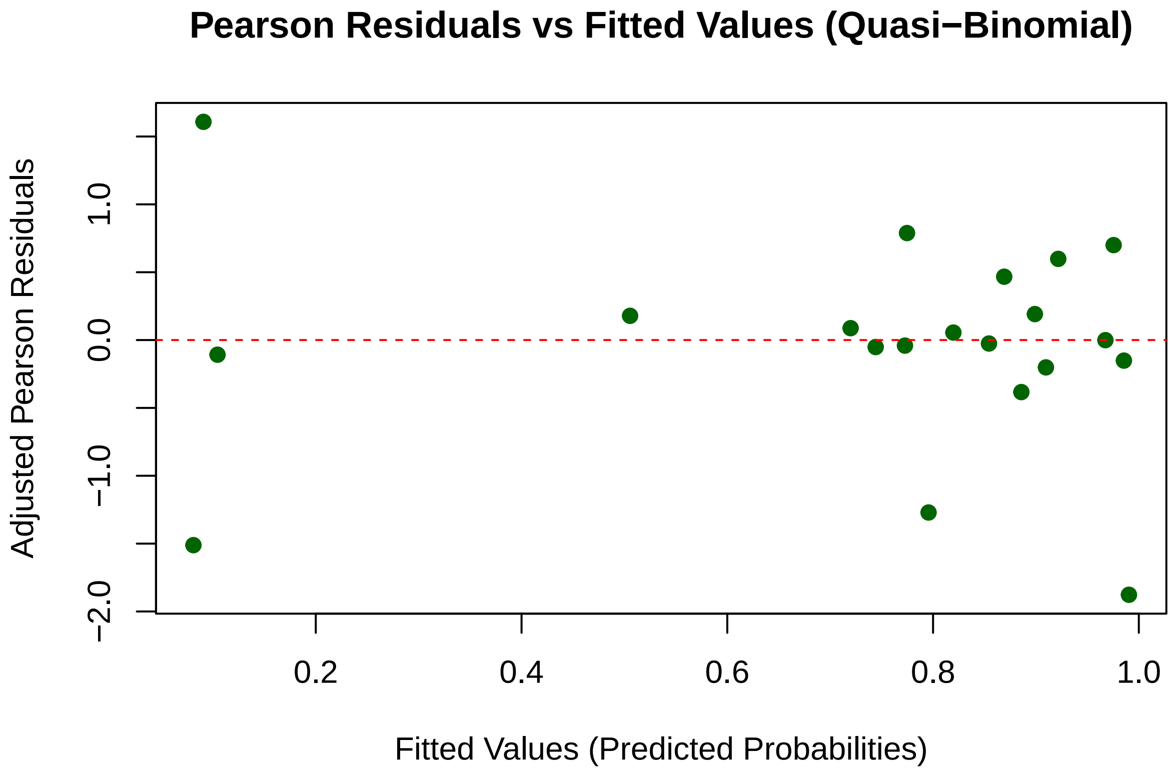
- If the variance model is correct, expecct Pearson residuals fall between -2 and +2
- Pearson residual plot shows that while most residuals fall within the expected range of ±2, there are a few points that exceed ±3. This suggests that the observed variability is larger than what the binomial model accounts for (overdispersion). Therefore, a better fit might be achieved by allowing for overdispersion (e.g., using a quasibinomial).

4. Quasi-binomial with logit link

```r
#quasi-binomial GLM
quasi_fit <- glm(survive/total ~ location + period,
                 weights = total,
                 family = quasibinomial(link = "logit"),
                 data = troutegg)

summary(quasi_fit)
```

```
## 
## Call:
## glm(formula = survive/total ~ location + period, family = quasibinomial(link = "logit"), 
##     data = troutegg, weights = total)
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   4.6358     0.6495   7.138 1.18e-05 ***
## location2    -0.4168     0.5682  -0.734 0.477315    
## location3    -1.2421     0.5066  -2.452 0.030501 *  
## location4    -0.9509     0.5281  -1.800 0.096970 .  
## location5    -4.6138     0.5777  -7.987 3.82e-06 ***
## period7      -2.1702     0.5504  -3.943 0.001953 ** 
## period8      -2.3256     0.5609  -4.146 0.001356 ** 
## period11     -2.4500     0.5405  -4.533 0.000686 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for quasibinomial family taken to be 5.330358)
## 
##     Null deviance: 1021.469  on 19  degrees of freedom
## Residual deviance:   64.495  on 12  degrees of freedom
## AIC: NA
## 
## Number of Fisher Scoring iterations: 5
```

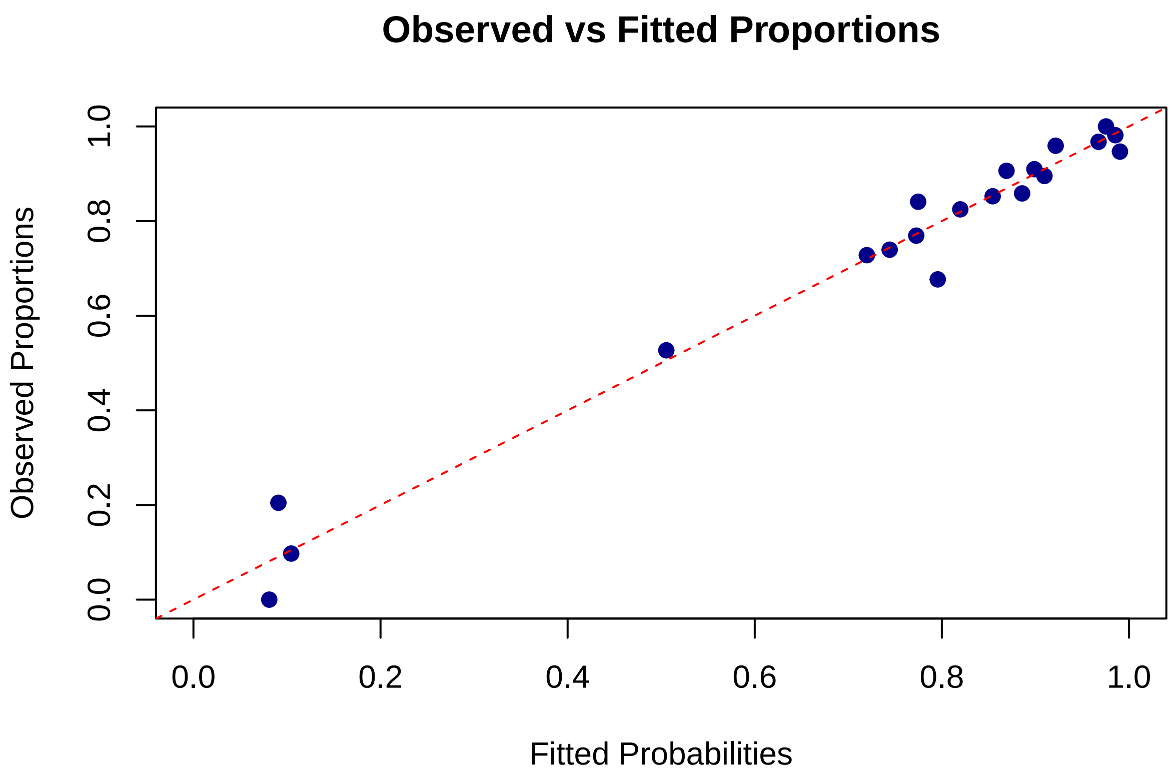
```
# Getting Pearson Residuals and Adjusting Them
pearson_raw <- residuals(quasi_fit, type = "pearson")

dispersion <- summary(quasi_fit)$dispersion

pearson_adj <- pearson_raw / sqrt(dispersion)

fitted_vals <- fitted(quasi_fit)


# Ploting Adjusted Pearson Residuals vs Fitted Values
plot(fitted_vals, pearson_adj,
     xlab = "Fitted Values (Predicted Probabilities)",
     ylab = "Adjusted Pearson Residuals",
     main = "Pearson Residuals vs Fitted Values (Quasi-Binomial)",
     pch = 19, col = "darkgreen")
abline(h = 0, lty = 2, col = "red")
```



The plot of adjusted Pearson residuals versus fitted values from the quasi-binomial model shows no strong patterns or evidence of systematic deviation. All residuals fall within the range of approximately −2 to +2, which is well within the expected bounds under a well-specified variance model. This is an improvement over the standard binomial model, where few residuals exceeded ±3, indicating overdispersion. Therefore, the quasi-binomial variance model is more appropriate for this data and provides a better fit by accounting for the extra-binomial variation.

5. Plot of observed proportions vs. the corresponding fitted probabilities.

```r
# Observed proportions
observed <- troutegg$survive / troutegg$total

# Fitted probabilities from the quasi-binomial model
fitted_probs <- fitted(quasi_fit)

# Plot
plot(fitted_probs, observed,
     xlab = "Fitted Probabilities",
     ylab = "Observed Proportions",
     main = "Observed vs Fitted Proportions",
     pch = 19, col = "darkblue",
     xlim = c(0, 1), ylim = c(0, 1))
abline(0, 1, col = "red", lty = 2)  # reference line y = x
```

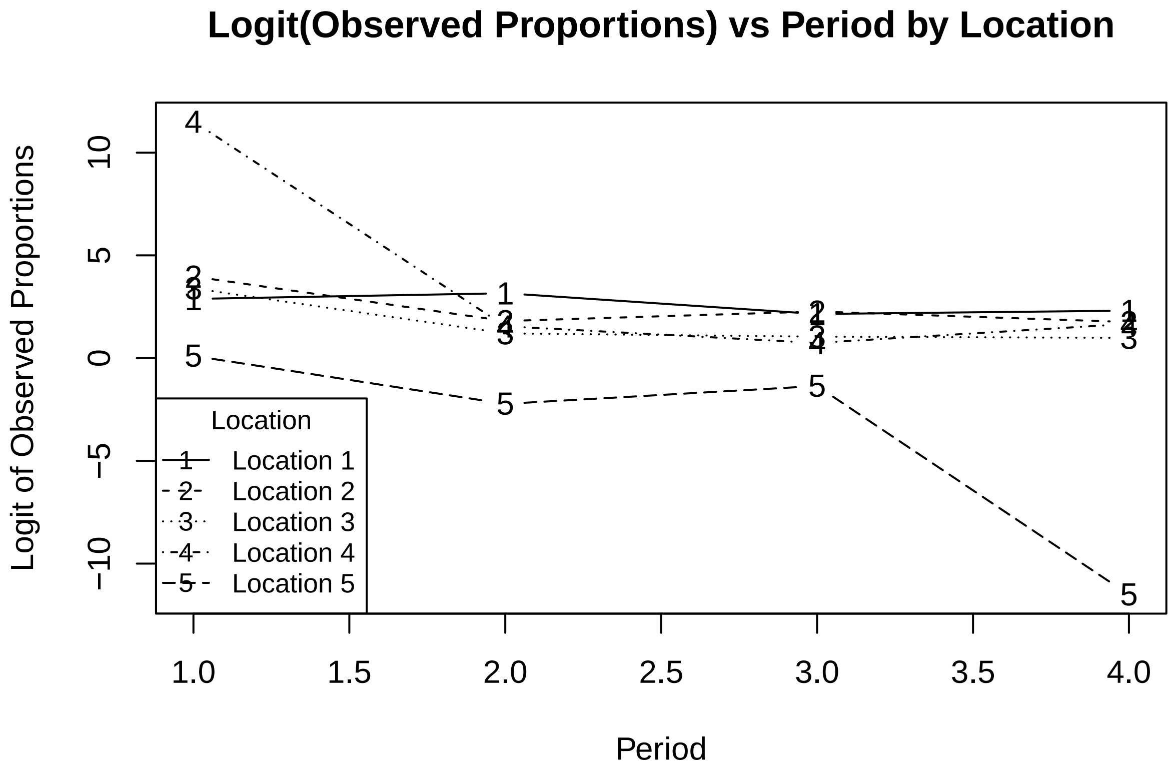


The plot of observed proportions versus fitted probabilities shows that the model's predictions align well with the actual observed data. Most points fall close to the reference line, thus, the probability model appears to fit the data well overall.

6. Does there appear to be any interaction between location and period?

```
# observed proportions
troutegg$prop <- troutegg$survive / troutegg$total

# Avoiding log(0) or log(1) with epsilon
epsilon <- 1e-5
troutegg$prop_adj <- pmin(pmax(troutegg$prop, epsilon), 1 - epsilon)

# logit of observed proportions
troutegg$logit_prop <- log(troutegg$prop_adj / (1 - troutegg$prop_adj))

# Plot
plot(logit_prop ~ as.numeric(period), data = troutegg, type = "n",
     ylab = "Logit of Observed Proportions",
     xlab = "Period",
     main = "Logit(Observed Proportions) vs Period by Location")

# Adding lines per location
locations <- unique(troutegg$location)
line_types <- 1:length(locations)
```

```r
for (i in seq_along(locations)) {
  loc_data <- subset(troutegg, location == locations[i])
  lines(logit_prop ~ as.numeric(period), data = loc_data,
        lty = line_types[i], type = "b", pch = as.character(locations[i]))
}

legend("bottomleft", legend = paste("Location", locations),
       lty = line_types, pch = as.character(locations), title = "Location", cex = 0.8)
```



The lines for different locations are not parallel and in some cases cross or diverge, indicating that the effect of period on survival is not consistent across locations. This suggests the presence of an interaction between location and period. In particular, Locations 4 and 5 show very different trends compared to the others, supporting the idea that the main effects model may be insufficient to fully capture the structure of the data.

7. Computing a point estimate and 95% confidence interval for the odds ratio of the last period vs the first period (period 11 vs. period 4) using the (non-saturated) quasibinomial fit and comparing these to the (non-saturated) binomial fit using nlWaldTest::nlConfint.

```r
library(nlWaldTest)

levels(troutegg$period)
```

```
## [1] "4"  "7"  "8"  "11"
```

```
# Binomial model
nlWaldTest::nlConfint(binGLM, texts = c("exp(b[4])"))
```

```
##                value     2.5 %    97.5 %
## exp(b[4]) 0.3864073 0.2131588 0.5596557
```

```
# Quasi-binomial model
nlWaldTest::nlConfint(quasi_fit, texts = c("exp(b[4])"))
```

```
##                value       2.5 %    97.5 %
## exp(b[4]) 0.3864073 -0.01358124 0.7863958
```

The estimated odds ratio for survival in period 11 vs period 4 is 0.386, indicating that the odds of survival in the last period are approximately 61% lower than in the first period.

In the quasi-binomial model, the 95% confidence interval for the odds ratio is approximately [0, 0.786], suggesting a non-significant effect due to the larger variance estimate. The binomial model, in contrast, gives a tighter and statistically significant interval of [0.213, 0.560]. This highlights the impact of variance modeling: the quasi-binomial model adjusts for overdispersion, resulting in more conservative inference.