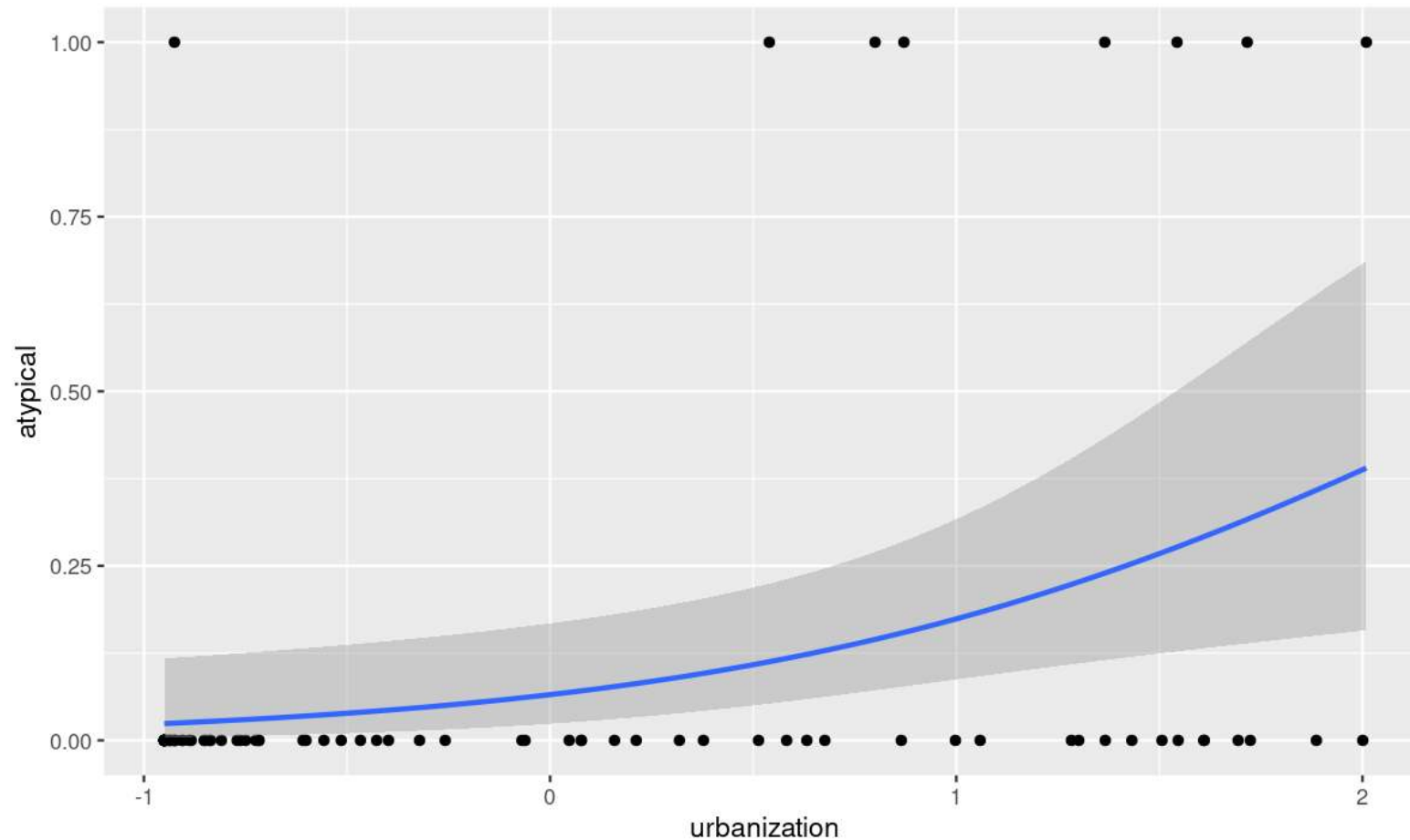


# Even more stats...

Generalized linear models,  
Other advanced models,  
Non-parametric stats



# Getting started (again)

Open RStudio

Open your NRI project

Open a **new** script for today:

File > New File > R Script

Make sure to load packages at the top:

```
library(tidyverse)
```

# Reference Material

This lecture covers A HUGE subject area

It is not comprehensive

It is a place to start, with references to guide you later on

# Generalized linear models

Chapter 13, "The R Book" by Michael J. Crawley.

(Freely available online through University of Manitoba Library)

DHARMa Tutorial

(Many great examples of model checking)

# Generalized linear models

## Gaussian Family (Normal data)

```
lm(y ~ x1 * x2, data = my_data)
```

## Poisson Family (Count data)

```
glm(counts ~ x1 * x2, family = "poisson", data = my_data)
```

## Negative Binomial Family (Overdispersed Count data)

```
MASS::glm.nb(counts ~ x1 * x2, data = my_data)
```

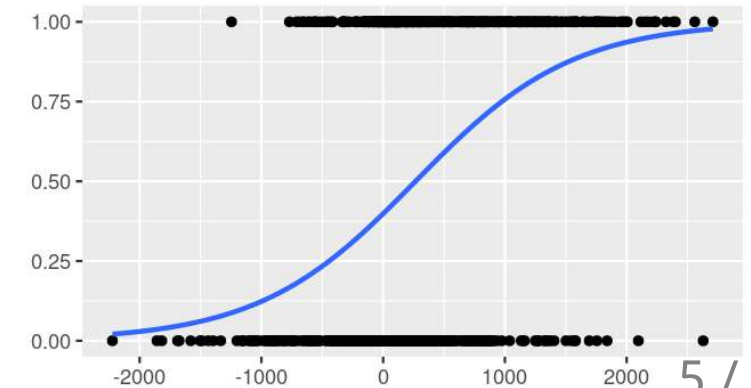
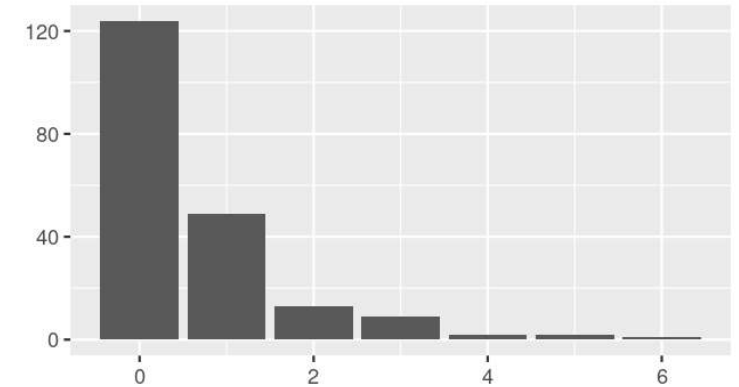
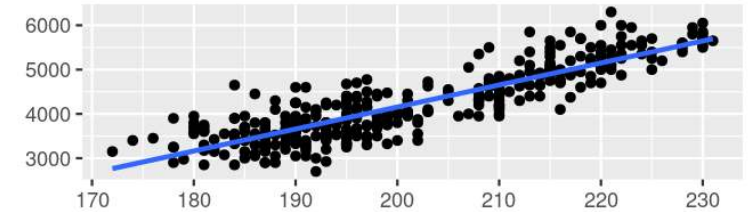
## Binomial Family

Binary, 0/1, Logistic Regression

```
glm(y ~ x1 * x2, family = "binomial", data = my_data)
```

Proportion with binary outcomes (10 yes, 5 no)

```
glm(cbind(Yes, No) ~ x1 * x2, family = "binomial", data = my_data)
```



# Count Data - Poisson Family

## Get the data

```
p <- read_csv("https://stats.idre.ucla.edu/stat/data/poisson_sim.csv")
p <- mutate(p, program = factor(prog, levels = 1:3,
                                labels = c("General", "Academic", "Vocational")))
p
```

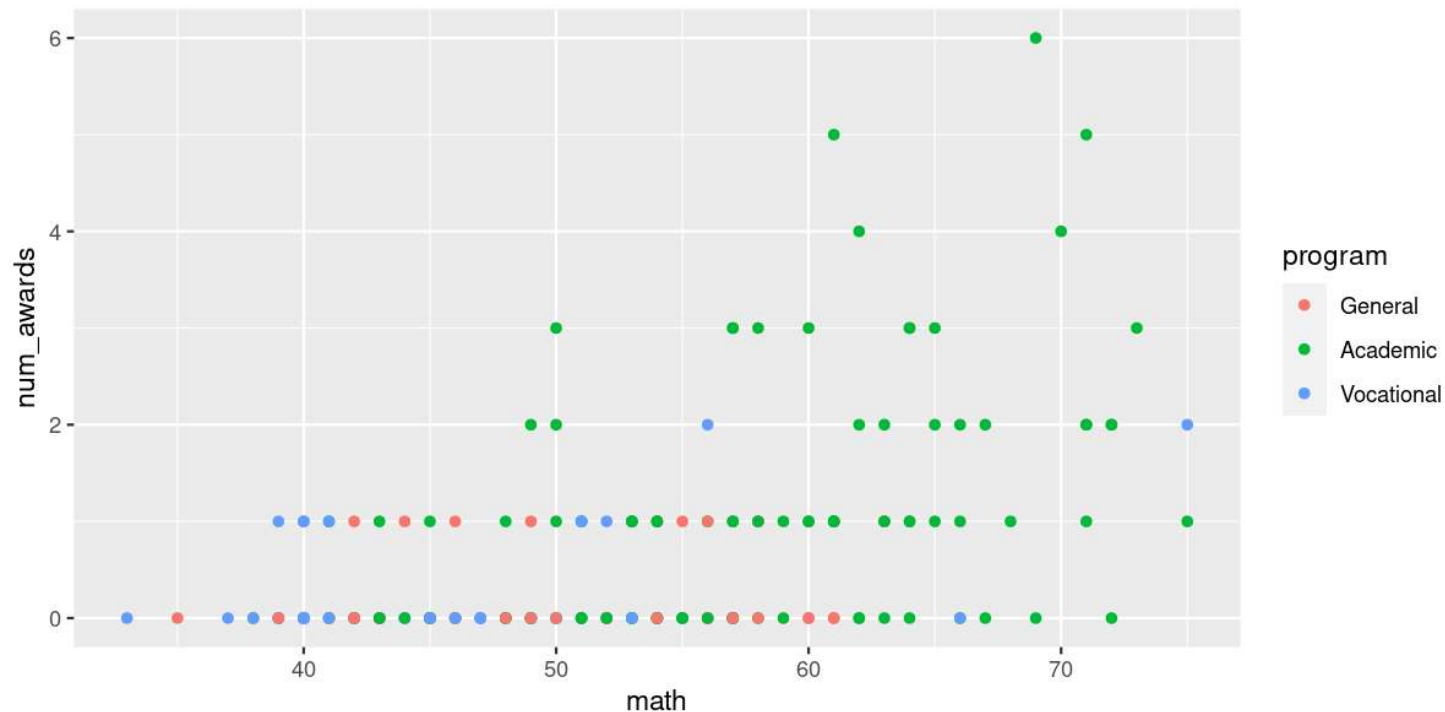
```
## # A tibble: 200 x 5
##       id num_awards  prog  math program
##   <dbl>    <dbl> <dbl> <dbl> <fct>
## 1     45         0     3    41 Vocational
## 2    108         0     1    41 General
## 3     15         0     3    44 Vocational
## 4     67         0     3    42 Vocational
## 5    153         0     3    40 Vocational
## 6     51         0     1    42 General
## 7    164         0     3    46 Vocational
## 8    133         0     3    40 Vocational
## 9      2         0     3    33 Vocational
## 10    53         0     3    46 Vocational
## # ... with 190 more rows
```

How do marks in **math** as well as **program** of study influence the number of awards a student receives (**num\_awards**)?

# Count Data - Poisson Family

## Look at the data

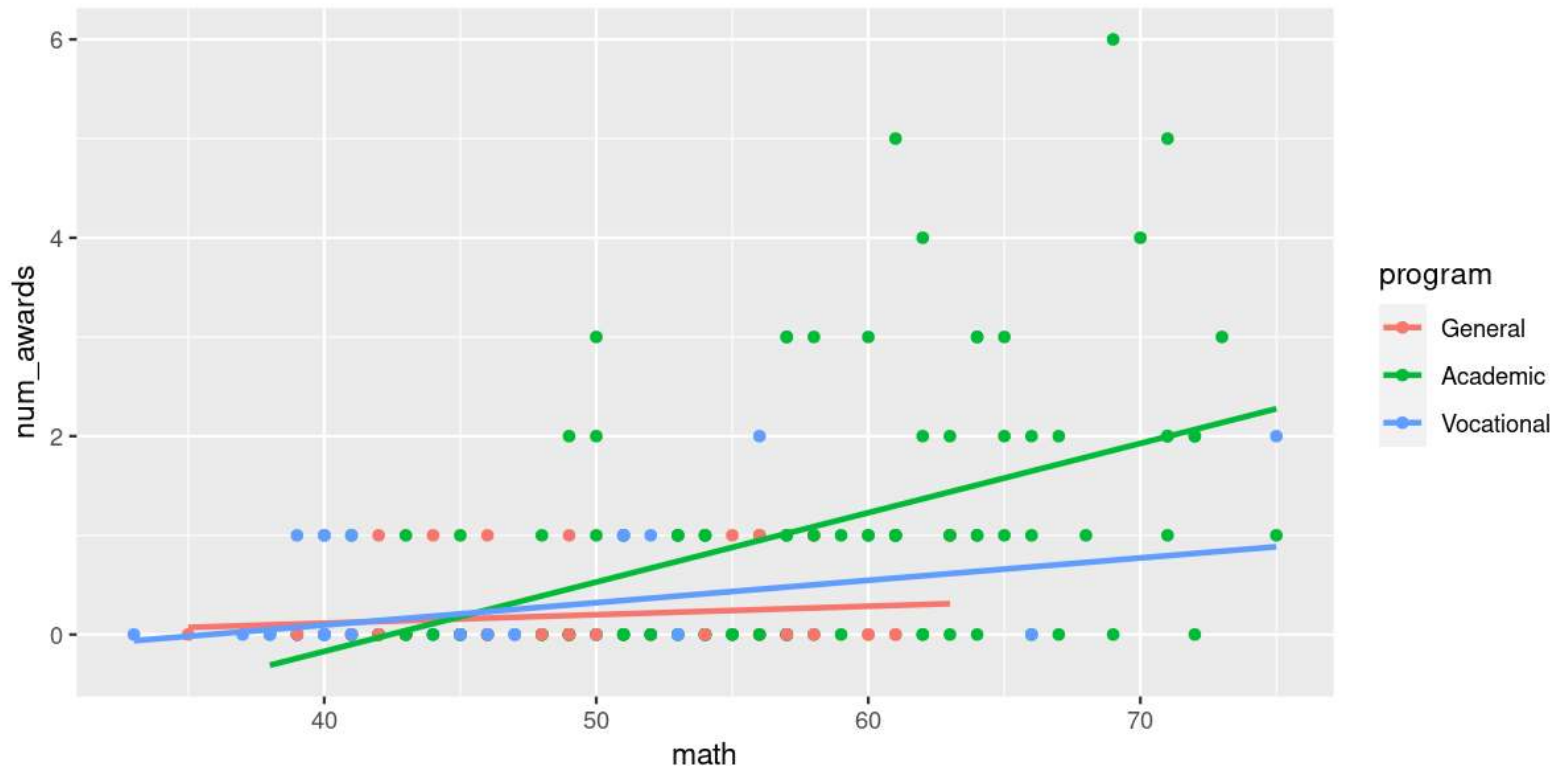
```
ggplot(data = p, aes(x = math, y = num_awards, colour = program)) +  
  geom_point()
```



# Count Data - Poisson Family

## Look at the data

```
ggplot(data = p, aes(x = math, y = num_awards, colour = program)) +  
  geom_point() +  
  stat_smooth(method = "glm", se = FALSE)
```

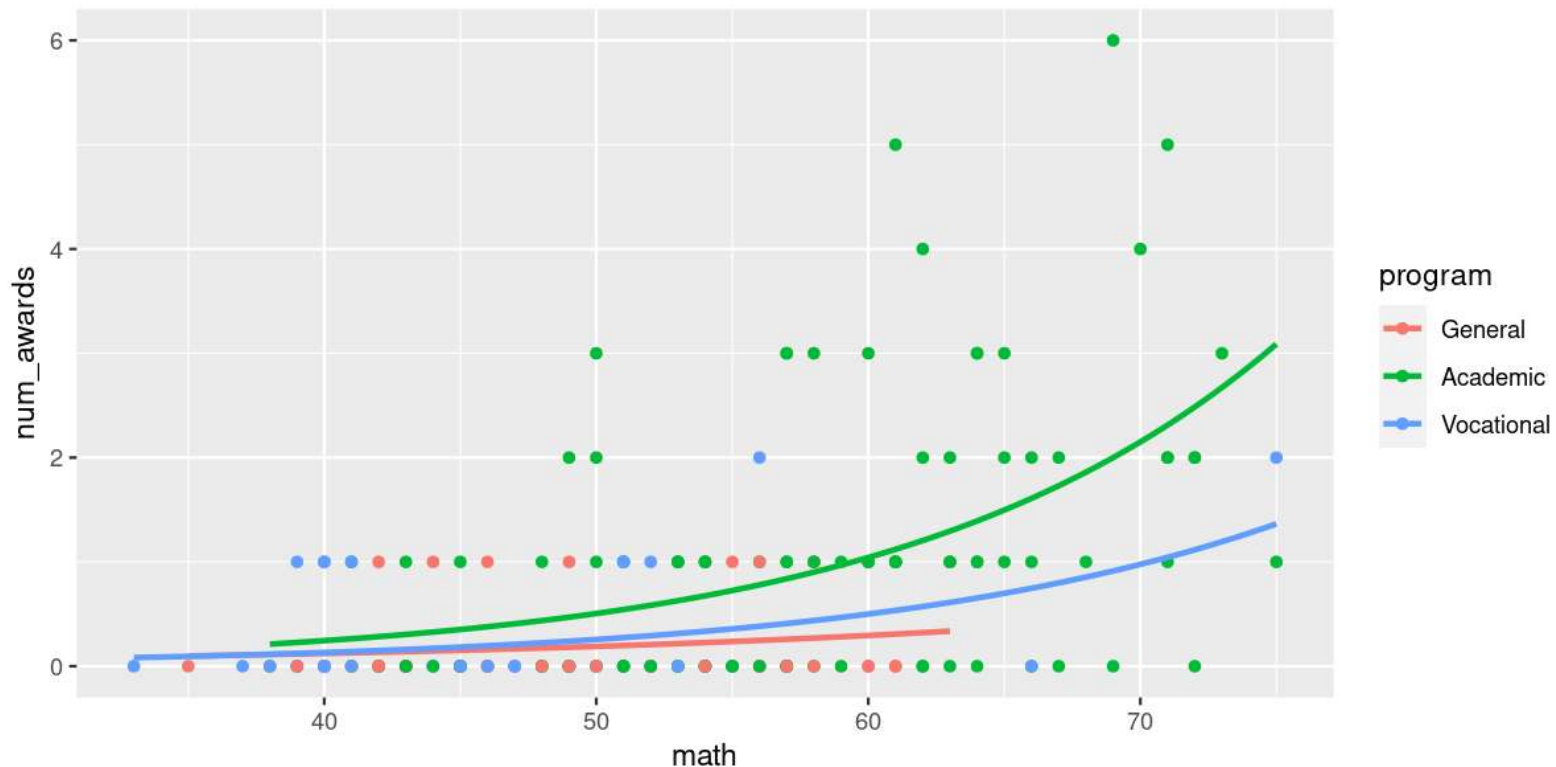




# Count Data - Poisson Family

## Look at the data

```
ggplot(data = p, aes(x = math, y = num_awards, colour = program)) +  
  geom_point() +  
  stat_smooth(method = "glm", method.args = list(family = "poisson"), se = FALSE)
```



# Count Data - Poisson Family

## Run model

```
m <- glm(num_awards ~ math + program, family = "poisson", data = p)
```

# Count Data - Poisson Family

## Run model

```
m <- glm(num_awards ~ math + program, family = "poisson", data = p)
```

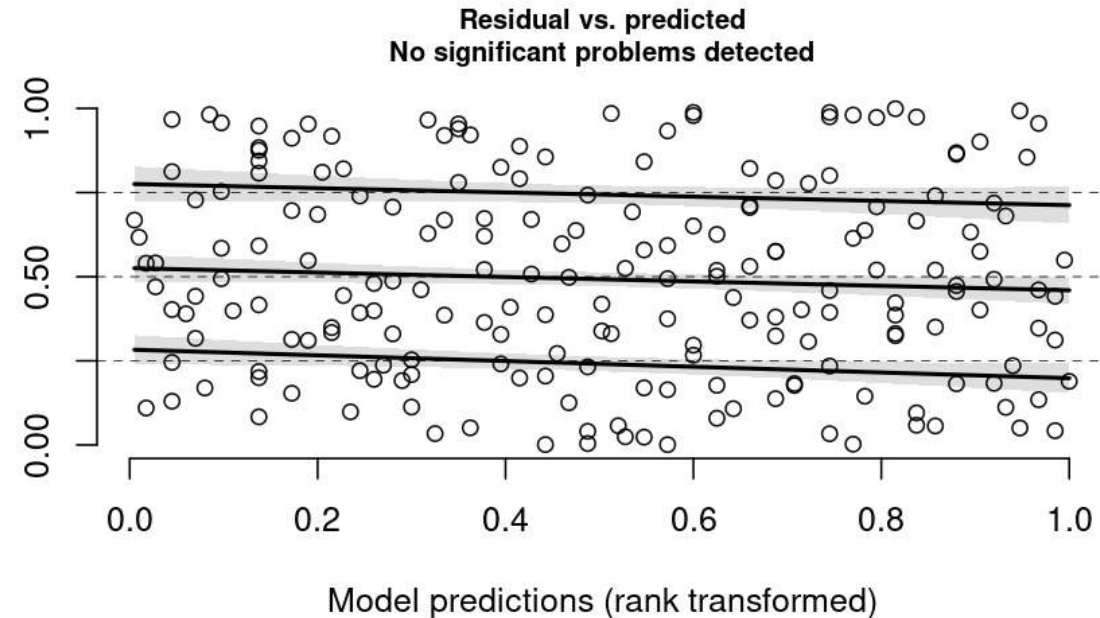
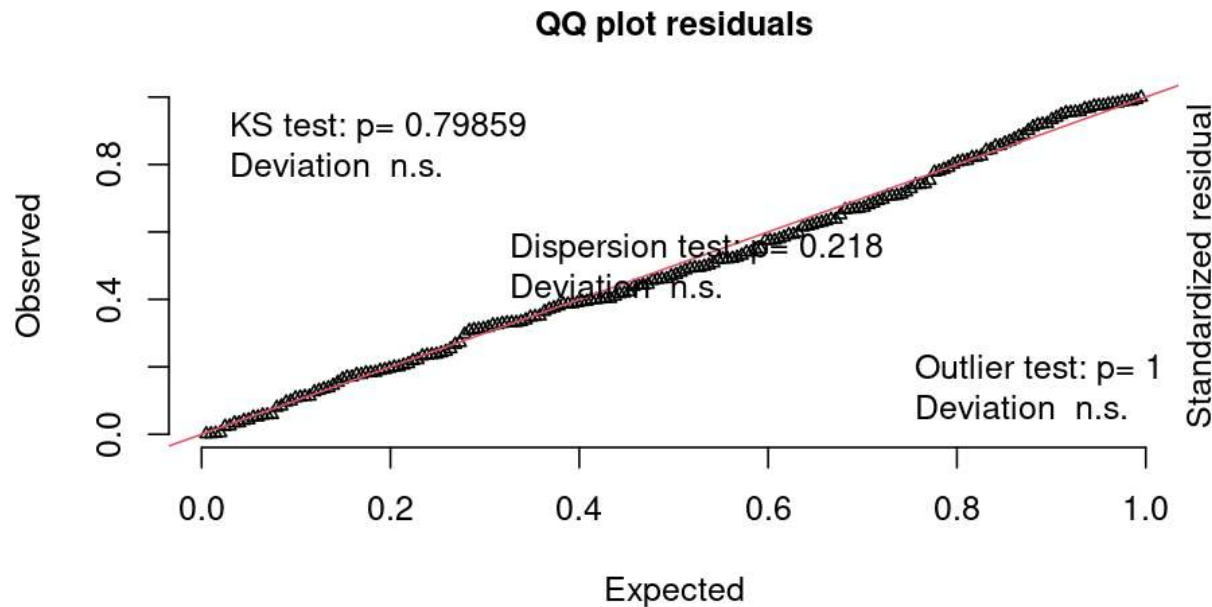
## Model Diagnostics

- More complicated in GLMs
- Here use **DHARMA** package
- Use **simulateResiduals()** to simulate scaled residuals from your model
  - *If model is correctly specified, scaled residuals follow a flat (uniform) distribution (no matter the model)*
- Use **plot = TRUE** to produce diagnostic plots to see if simulated match expectation
- **n = 1000** isn't strictly necessary but runs more simulations to produce more stable results
- These diagnostics apply to almost **any** test (not just Poisson)

```
library(DHARMA)  
r <- simulateResiduals(m, n = 1000, plot = TRUE)
```

# Count Data - Poisson Family - Model Diagnostics

DHARMA residual diagnostics



- Tests **Uniformity** with Kolmogorov-Smirnov (KS) test
  - (do the residuals match a Uniform distribution?)
- Tests for **Over/Underdispersion** with Dispersion Test
- Tests for more **Outliers** than expected with Outlier test
- **No real problems here**
  - (straight line, all tests non-significant)

- Checks distribution of residuals (using visual and quantiles)
- Dotted lines show expected quantiles
- Black lines show simulated quantiles
- Outliers would show up as red stars
- **No real problems here**
  - (straight lines match expected quantiles)

# Count Data - Poisson Family - Model Diagnostics

## Zero-inflation

- Will often pop up as iffy residuals
- Overdispersion can lead to false positives
- Here not a problem

```
testZeroInflation(m, plot = FALSE)
```

```
##  
##      DHARMA zero-inflation test via comparison to expected zeros with simulation under H0 =  
##      fitted model  
##  
## data:  simulationOutput  
## ratioObsSim = 1.0131, p-value = 0.832  
## alternative hypothesis: two.sided
```

# Count Data - Poisson Family

```
summary(m)
```

```
## Call:
## glm(formula = num_awards ~ math + program, family = "poisson",
##      data = p)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.2043  -0.8436  -0.5106   0.2558   2.6796
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -5.24712    0.65845  -7.969 1.60e-15 ***
## math           0.07015    0.01060   6.619 3.63e-11 ***
## programAcademic  1.08386    0.35825   3.025 0.00248 **
## programVocational 0.36981    0.44107   0.838 0.40179
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 287.67  on 199  degrees of freedom
## Residual deviance: 189.45  on 196  degrees of freedom
```

## Traditional check for overdispersion:

- Look at deviance (189.45) vs. df (196)

```
deviance(m) / df.residual(m)
```

```
## [1] 0.9665797
```

- Almost 1 (which it is expected to be)

```
pchisq(deviance(m), df.residual(m),
        lower.tail = FALSE)
```

```
## [1] 0.6182274
```

- Test shows no significant overdispersion

# Count Data - Poisson Family

```
summary(m)
```

```
## Call:
## glm(formula = num_awards ~ math + program, family = "poisson",
##      data = p)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.2043  -0.8436  -0.5106   0.2558   2.6796
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -5.24712    0.65845  -7.969 1.60e-15 ***
## math           0.07015    0.01060   6.619 3.63e-11 ***
## programAcademic  1.08386    0.35825   3.025 0.00248 **
## programVocational 0.36981    0.44107   0.838 0.40179
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 287.67  on 199  degrees of freedom
## Residual deviance: 189.45  on 196  degrees of freedom
```

## Interpreting Results

- Significantly more awards received with higher marks in math
  - i.e. Number of awards increases by 0.07 log-counts per 1 unit increase in Math mark
- Significantly more awards received in Academic Program compared to General
  - i.e. Number of awards greater by 1.08 log-counts for Academic compared to General
- No difference in amount of awards received in Vocational vs. General Program

# Count Data - Poisson Family

```
summary(m)
```

```
## Call:
## glm(formula = num_awards ~ math + program, family = "poisson",
##      data = p)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.2043  -0.8436  -0.5106   0.2558   2.6796
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -5.24712    0.65845  -7.969 1.60e-15 ***
## math           0.07015    0.01060   6.619 3.63e-11 ***
## programAcademic  1.08386    0.35825   3.025  0.00248 **
## programVocational 0.36981    0.44107   0.838  0.40179
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 287.67  on 199  degrees of freedom
## Residual deviance: 189.45  on 196  degrees of freedom
```



OH WTF.



# Count Data - Poisson Family

```
summary(m)
```

```
## Call:
## glm(formula = num_awards ~ math + program, family = "poisson",
##      data = p)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.2043  -0.8436  -0.5106   0.2558   2.6796
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.24712    0.65845  -7.969 1.60e-15 ***
## math          0.07015    0.01060   6.619 3.63e-11 ***
## programAcademic  1.08386    0.35825   3.025 0.00248 **
## programVocational 0.36981    0.44107   0.838 0.40179
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 287.67  on 199  degrees of freedom
## Residual deviance: 189.45  on 196  degrees of freedom
```

## Interpreting Results

- Convert to ratios with  $e^{\text{est}}$

```
coef(m)
```

```
##              (Intercept)  -5.247
##                   math     0.070
##      programAcademic     1.084
##      programVocational    0.370
```

```
exp(coef(m))
```

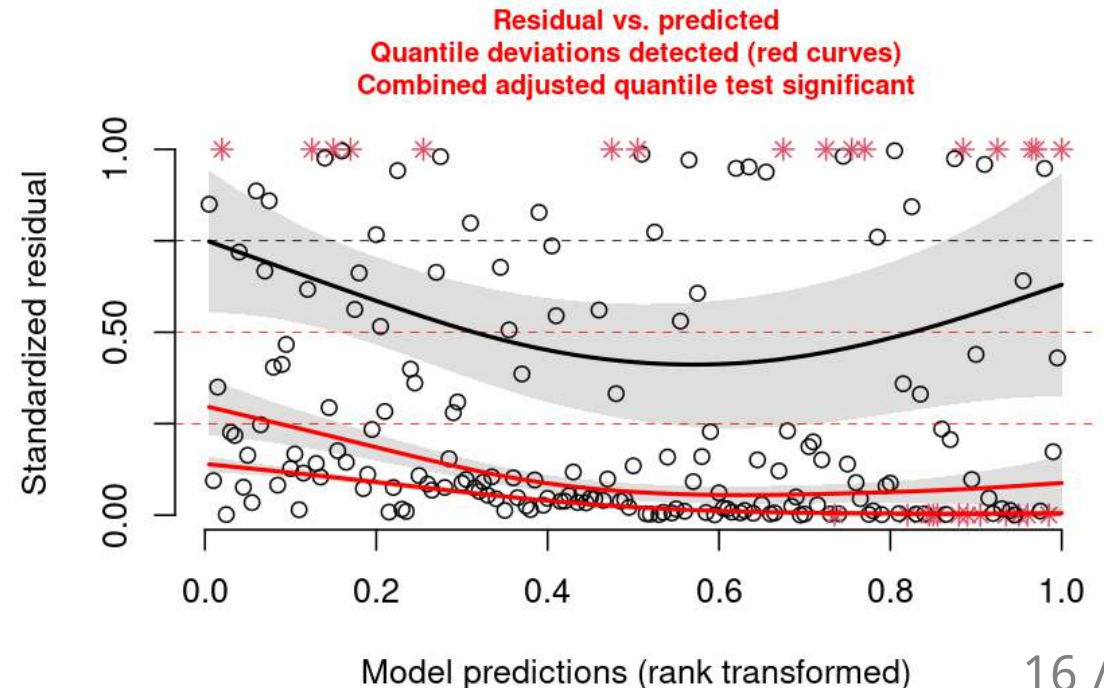
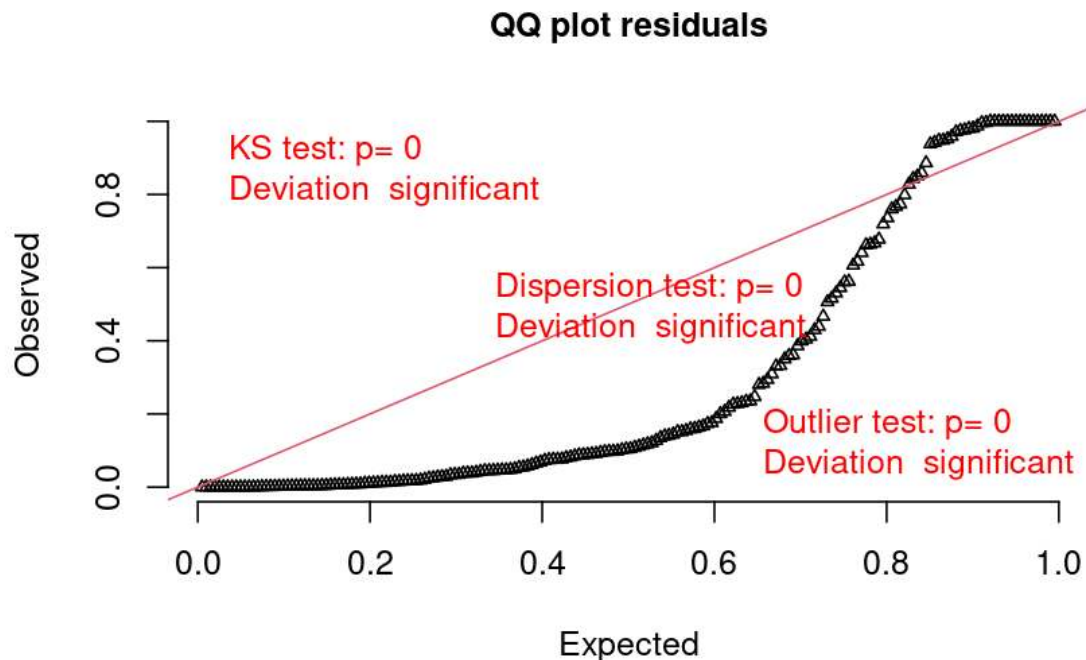
```
##              (Intercept)    0.005
##                   math     1.073
##      programAcademic     2.956
##      programVocational    1.447
```

- No. awards increases by 1.07 *times* per 1 unit increase in Math mark
- No. awards received by Academic is 2.96 *times* greater than in General program

# Overdispersion

- Overdispersion when data is spread out more than distribution would be (longer tails)
- Results in **highly** significant findings that are **not** valid!!
- Simulated residuals run from 0 to 1, but here more residuals around 0 and 1 (longer tails)

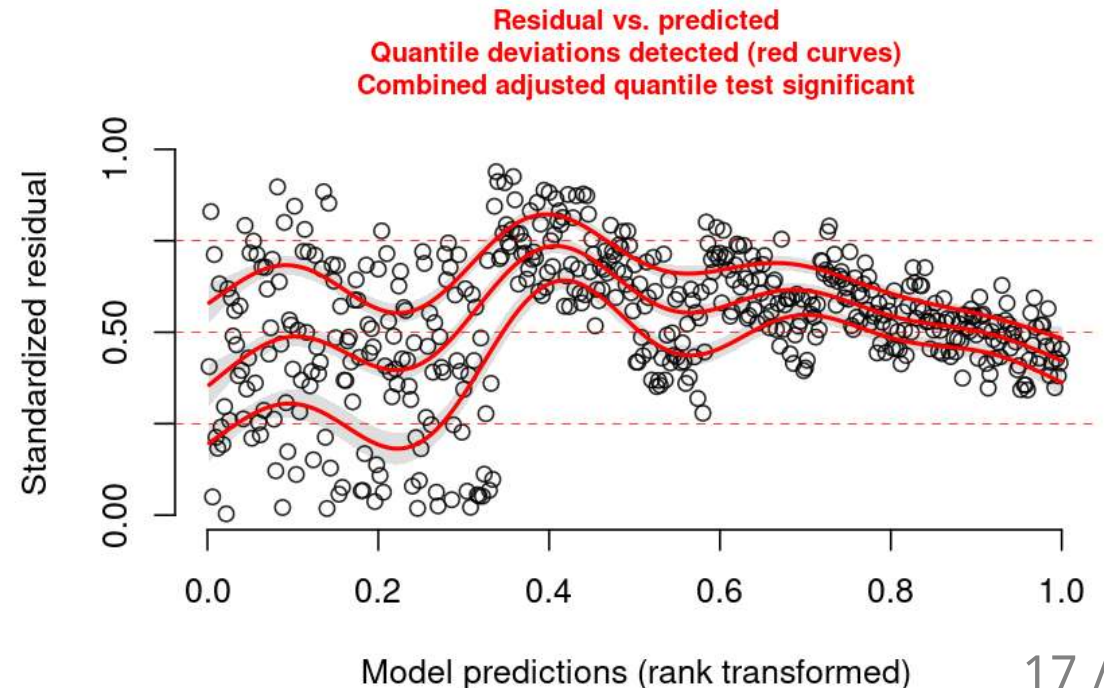
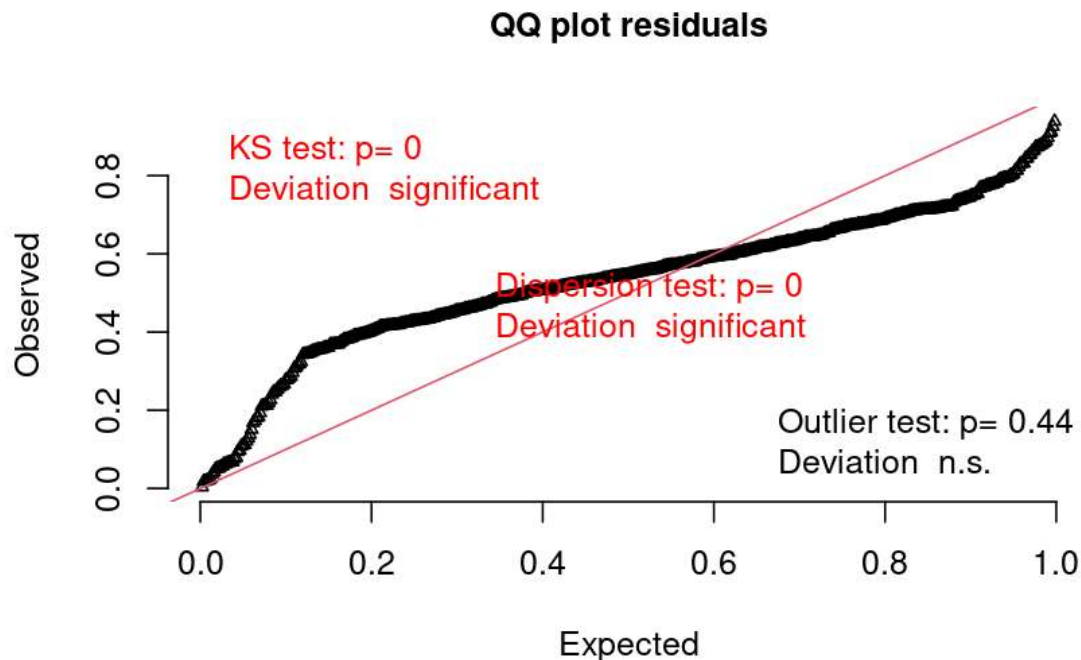
DHARMA residual diagnostics



# Underdispersion

- Underdispersion is less common
- When data is gathered towards the centre more than distribution would be (shorter tails)
- These simulated residuals run from 0 to 1
- Here, more residuals around 0.5 (shorter tails)

DHARMA residual diagnostics



# Overdispersed Count Data - Negative Binomial

```
quine <- MASS::quine
```

## Poisson GLM

```
m1 <- glm(Days ~ Sex, data = quine, family =  
"poisson")  
summary(m1)
```

```
## Coefficients:  
##              Estimate Std. Error z value Pr(>|z|)  
## (Intercept)  2.72294    0.02865  95.030  < 2e-16  
***  
## SexM         0.16490    0.04080   4.041 5.31e-05  
***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05  
'.' 0.1 ' ' 1  
##  
## (Dispersion parameter for poisson family taken to  
be 1)  
##  
##      Null deviance: 2073.5  on 145  degrees of  
freedom
```

## Negative Binomial GLM

```
m2 <- MASS::glm.nb(Days ~ Sex, data = quine)  
summary(m2)
```

```
## Coefficients:  
##              Estimate Std. Error z value Pr(>|z|)  
## (Intercept)  2.7229    0.1116  24.395  <2e-16  
***  
## SexM         0.1649    0.1656   0.996   0.319  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05  
'.' 0.1 ' ' 1  
##  
## (Dispersion parameter for Negative Binomial(1.0741)  
family taken to be 1)  
##  
##      Null deviance: 169.50  on 145  degrees of  
freedom  
## Residual deviance: 168.51  on 144  degrees of  
freedom
```

# Overdispersed Count Data - Negative Binomial

```
quine <- MASS::quine
```

## Poisson GLM

```
m1 <- glm(Days ~ Sex, data = quine, family =  
"poisson")  
summary(m1)
```

```
## Coefficients:  
##              Estimate Std. Error z value Pr(>|z|)  
## (Intercept)  2.72294    0.02865  95.030  < 2e-16  
***  
## SexM          0.16490    0.04080   4.041 5.31e-05  
***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05  
'.' 0.1 ' ' 1  
##  
## (Dispersion parameter for poisson family taken to  
be 1)  
##  
##      Null deviance: 2073.5  on 145  degrees of  
freedom
```

## Negative Binomial GLM

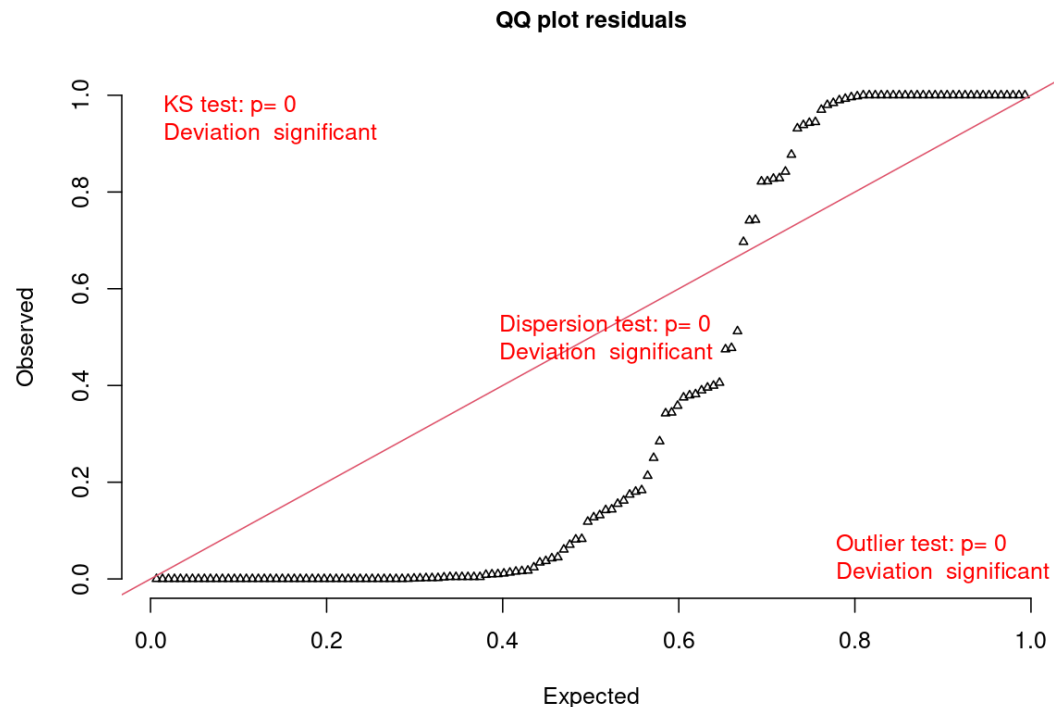
```
m2 <- MASS::glm.nb(Days ~ Sex, data = quine)  
summary(m2)
```

```
## Coefficients:  
##              Estimate Std. Error z value Pr(>|z|)  
## (Intercept)  2.7229    0.1116  24.395  <2e-16  
***  
## SexM          0.1649    0.1656   0.996   0.319  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05  
'.' 0.1 ' ' 1  
##  
## (Dispersion parameter for Negative Binomial(1.0741)  
family taken to be 1)  
##  
##      Null deviance: 169.50  on 145  degrees of  
freedom  
## Residual deviance: 168.51  on 144  degrees of  
freedom
```

# Overdispersed Count Data - Negative Binomial

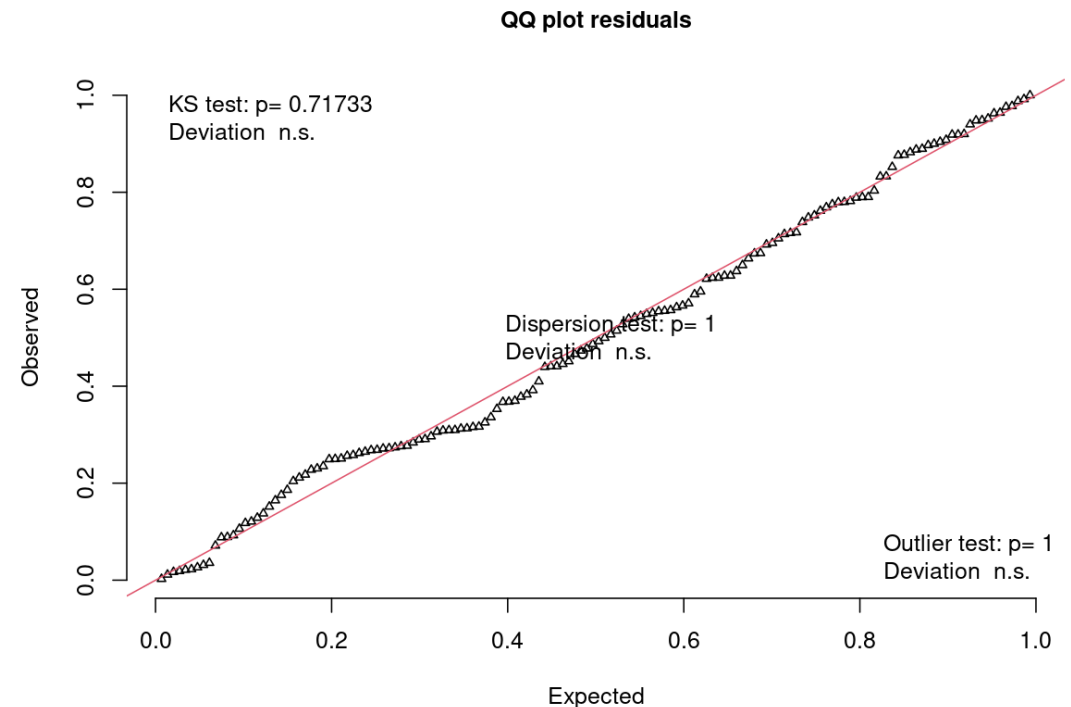
## Poisson GLM

```
r <- simulateResiduals(m1)
plotQQunif(r) # Just the first Uniformity Plot
```



## Negative Binomial GLM

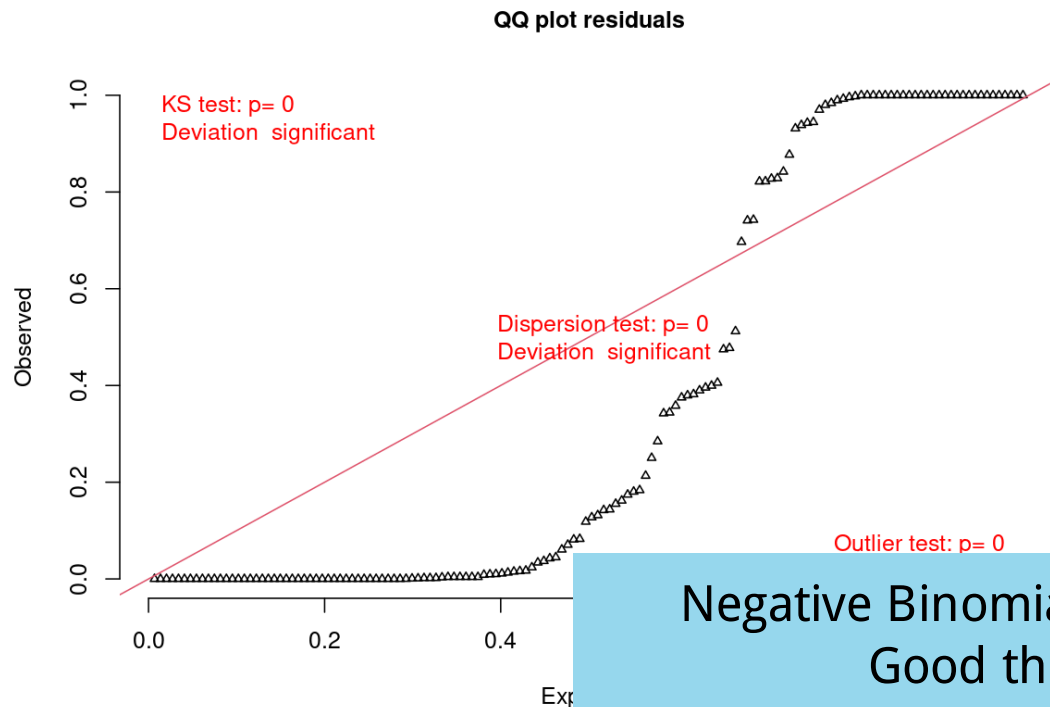
```
r <- simulateResiduals(m2)
plotQQunif(r) # Just the first Uniformity Plot
```



# Overdispersed Count Data - Negative Binomial

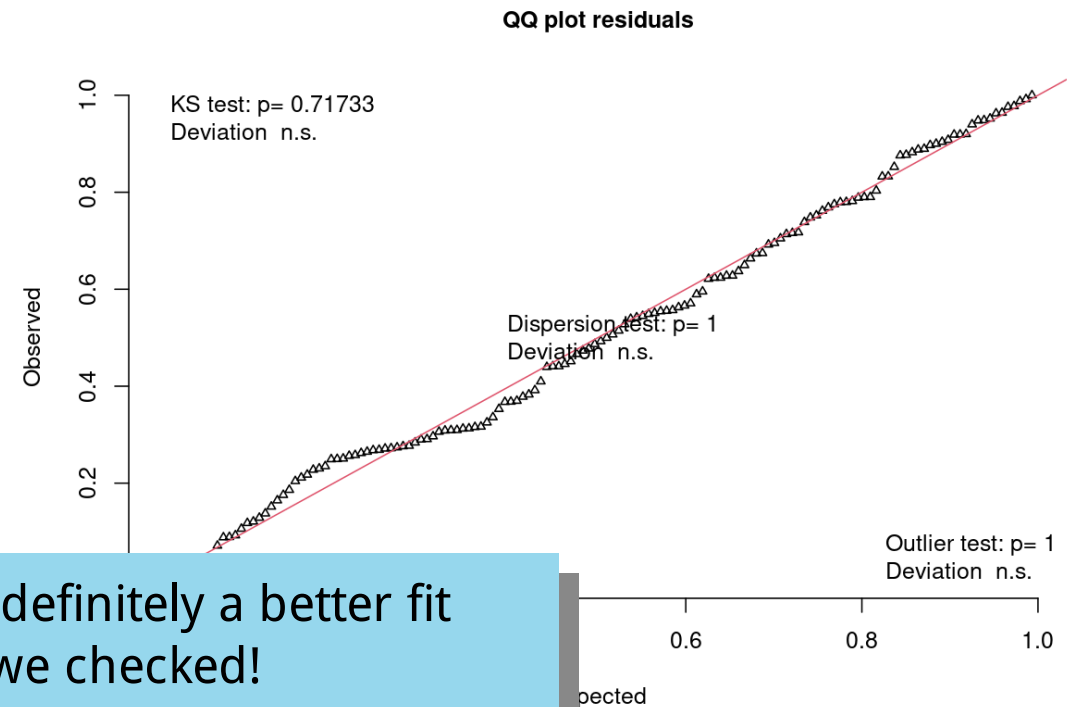
## Poisson GLM

```
r <- simulateResiduals(m1)
plotQQunif(r) # Just the first Uniformity Plot
```



## Negative Binomial GLM

```
r <- simulateResiduals(m2)
plotQQunif(r) # Just the first Uniformity Plot
```



Negative Binomial is definitely a better fit  
Good thing we checked!

# Your Turn!

We have the **crabs** dataset

```
crabs <-  
read_csv("https://steffilazerte.ca/NRI_Labs/data/crabs.csv")
```

## Background

- Horseshoe crabs form pairs for spawning (mating)
- But extra, unattached males crowd around and try to get involved (Satellite males)



## Your Job

You're interested in the effect of female size (**width**) on the number of male **satellites**

- Look at your data (make a plot)
- Run a **glm()** for count data
- Check your diagnostics. Do you have a problem? Check for overdispersion and zero-inflation
- Apply an overdispersion fix
- Check your diagnostics. Do you have a problem? Check for zero-inflation



# Zero-inflated Models (Advanced example! Above and beyond!)



- **glmmTMB()** function from **glmmTMB** package
- Allows modeling zero-inflation (**ziformula** says zeros are a function of the variable **weight**)

```
crabs <- read_csv("https://steffilazerte.ca/NRI_Labs/data/crabs.csv")
library(glmmTMB)
m <- glmmTMB(satellites ~ weight, ziformula = ~ weight , family = "nbinom2", data = crabs)
summary(m)
```

```
## Overdispersion parameter for nbinom2 family (): 4.96
##
## Conditional model:
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.8979     0.3051   2.943  0.00325 **
## weight        0.2171     0.1118   1.942  0.05217 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Zero-inflation model:
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.7546     0.9837   3.817 0.000135 ***
## weight       -1.9123     0.4320  -4.426 9.59e-06 ***
## ---
```

# Zero-inflated Models (Advanced example! Above and beyond!)



- **glmmTMB()** function from **glmmTMB** package
- Allows modeling zero-inflation (**ziformula** says zeros are a function of the variable **weight**)

```
crabs <- read_csv("https://steffilazerte.ca/NRI_Labs/data/crabs.csv")
library(glmmTMB)
m <- glmmTMB(satellites ~ weight, ziformula = ~ weight, family = "nbinom2", data = crabs)
summary(m)
```

```
## Overdispersion parameter for nbinom2 family (): 4.96
```

```
##
```

```
## Conditional model:
```

|                | Estimate | Std. Error | z value | Pr(> z )   |
|----------------|----------|------------|---------|------------|
| ## (Intercept) | 0.8979   | 0.3051     | 2.943   | 0.00325 ** |
| ## weight      | 0.2171   | 0.1118     | 1.942   | 0.05217 .  |

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'
```

```
##
```

```
## Zero-inflation model:
```

|                | Estimate | Std. Error | z value | Pr(> z )     |
|----------------|----------|------------|---------|--------------|
| ## (Intercept) | 3.7546   | 0.9837     | 3.817   | 0.000135 *** |
| ## weight      | -1.9123  | 0.4320     | -4.426  | 9.59e-06 *** |

```
## ---
```

"Regular" effects  
Heavier females have more satellite males  
(P = 0.052)

# Zero-inflated Models (Advanced example! Above and beyond!)



- **glmmTMB()** function from **glmmTMB** package
- Allows modeling zero-inflation (**ziformula** says zeros are a function of the variable **weight**)

```
crabs <- read_csv("https://steffilazerte.ca/NRI_Labs/data/crabs.csv")
library(glmmTMB)
m <- glmmTMB(satellites ~ weight, ziformula = ~ weight , family = "nbinom2", data = crabs)
summary(m)
```

```
## Overdispersion parameter for nbinom2 family (): 4.96
```

```
##
```

```
## Conditional model:
```

```
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.8979     0.3051   2.943  0.00325 **
## weight        0.2171     0.1118   1.942  0.05217 .
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'
```

```
##
```

```
## Zero-inflation model:
```

```
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.7546     0.9837   3.817 0.000135 ***
## weight       -1.9123     0.4320  -4.426 9.59e-06 ***
```

```
## ---
```

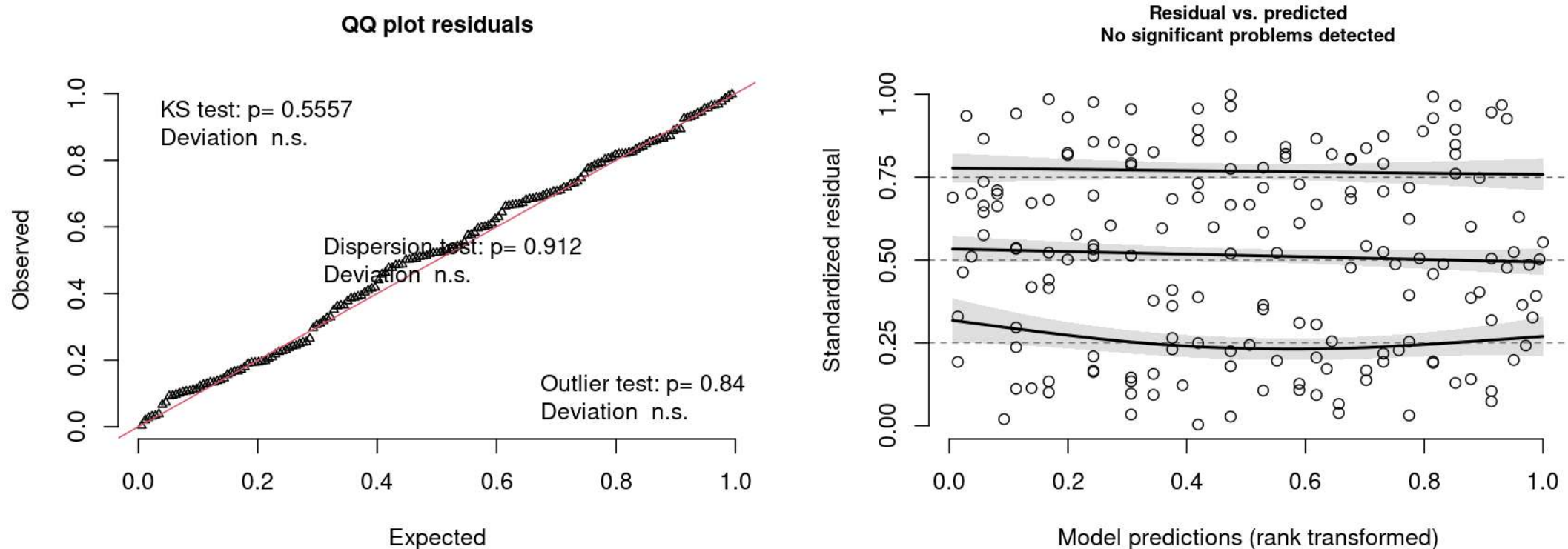
"Zero-inflation" effects  
more zeros counts as weight decreases  
What we expected!

# Zero-inflated Models (Advanced example! Above and beyond!)

```
r <- simulateResiduals(m, plot = TRUE)
```



DHARMA residual diagnostics

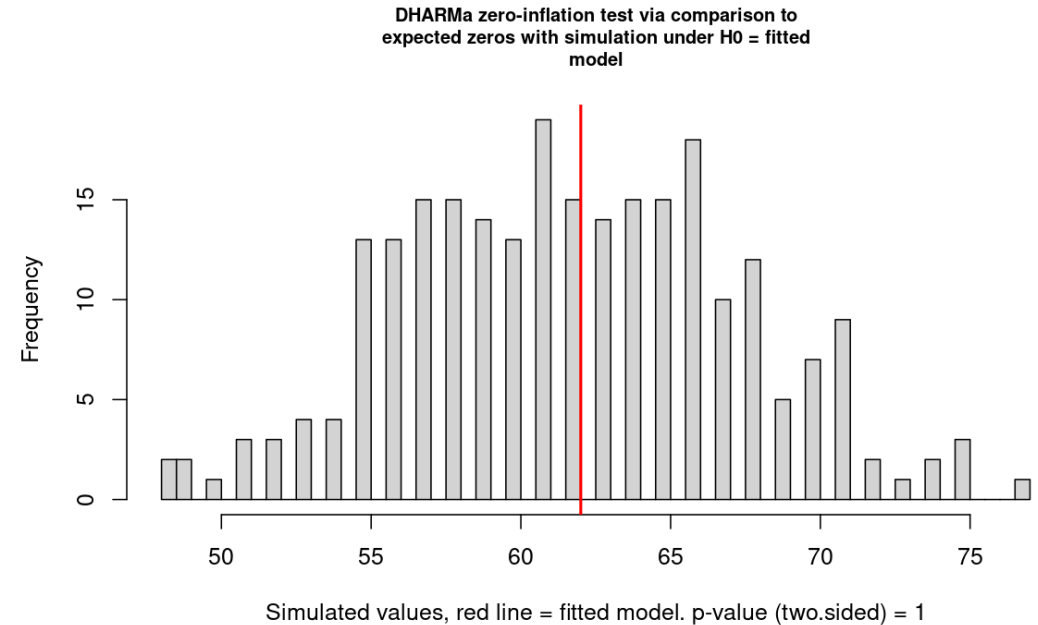


# Zero-inflated Models (Advanced example! Above and beyond!)



```
testZeroInflation(m)
```

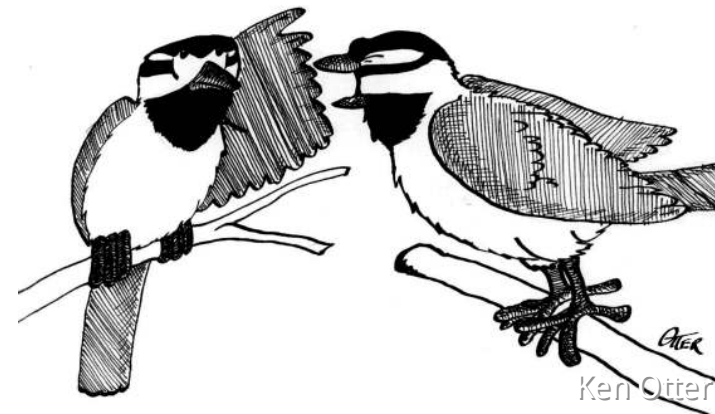
```
##
##      DHARMA zero-inflation test via comparison to
##      expected zeros with simulation under H0 =
##      fitted model
##
## data:  simulationOutput
## ratioObsSim = 1.0016, p-value = 1
## alternative hypothesis: two.sided
```



# Binary Data (0/1) - Binomial Family (logistic regression)

## Get the data

- Mountain chickadees atypical songs by urbanization
- Negative **urbanization** more rural
- Positive **urbanization** more urban
- **atypical\_c** atypical singer (1) or 'normal' singer (0)



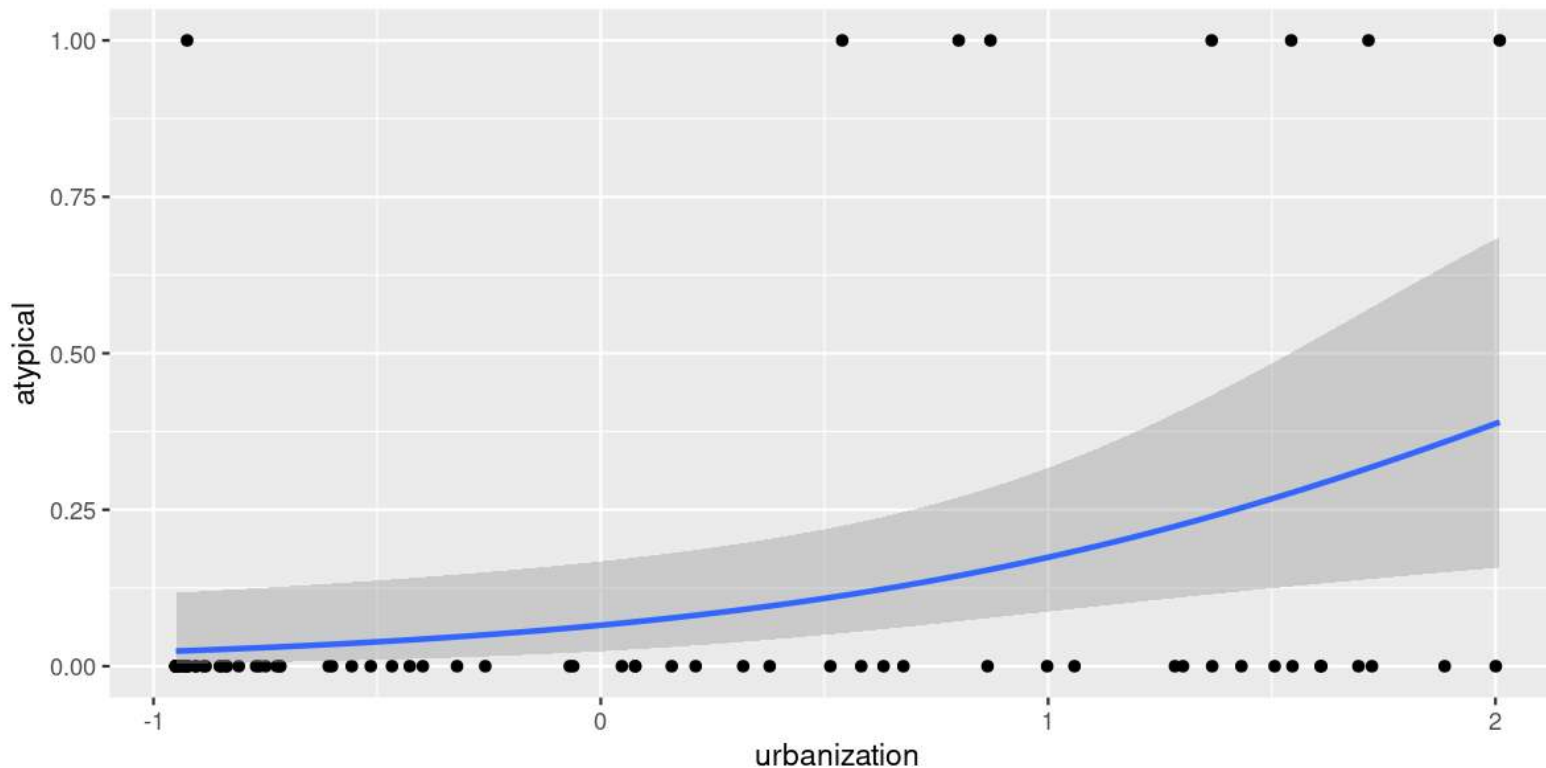
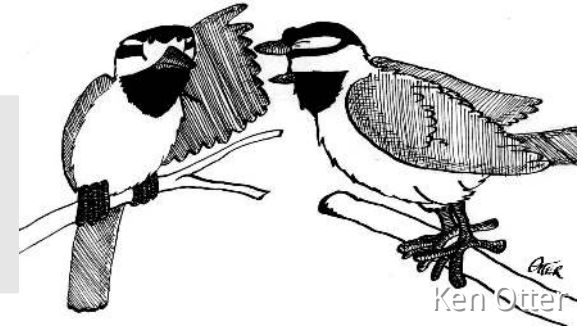
```
atypical <- read_csv("https://steffilazerte.ca/NRI_Labs/data/atypical.csv")
atypical
```

```
## # A tibble: 78 x 2
##   atypical urbanization
##   <dbl>         <dbl>
## 1         1         0.540
## 2         0         0.582
## 3         0        -0.950
## 4         0        -0.950
## 5         0         0.513
## 6         0         1.69
```

# Binary Data (0/1) - Binomial Family (logistic regression)

## Look at the data

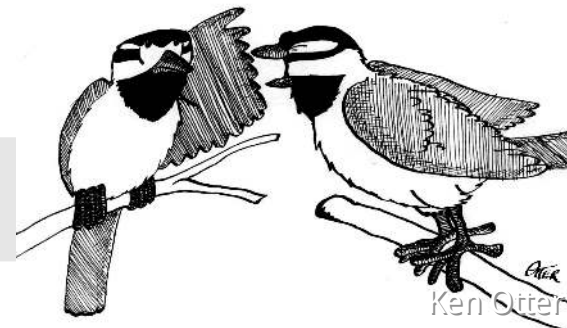
```
ggplot(data = atypical, aes(x = urbanization, y = atypical)) +  
  geom_point() +  
  stat_smooth(method = "glm", method.args = list(family = "binomial"))
```



# Binary Data (0/1) - Binomial Family (logistic regression)

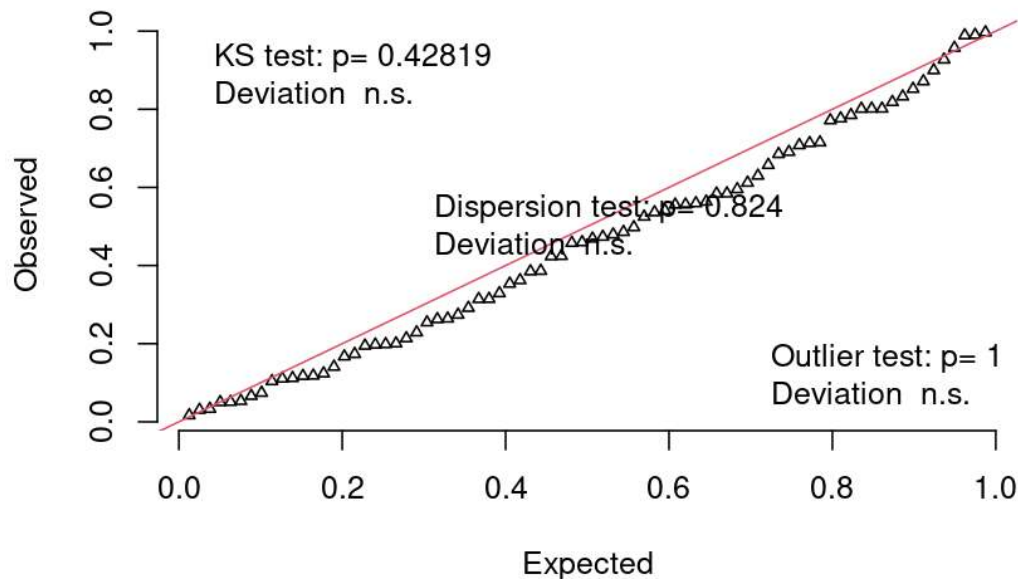
## Run model and check diagnostics

```
m <- glm(atypical ~ urbanization, family = "binomial", data = atypical)
r <- simulateResiduals(m, plot = TRUE)
```

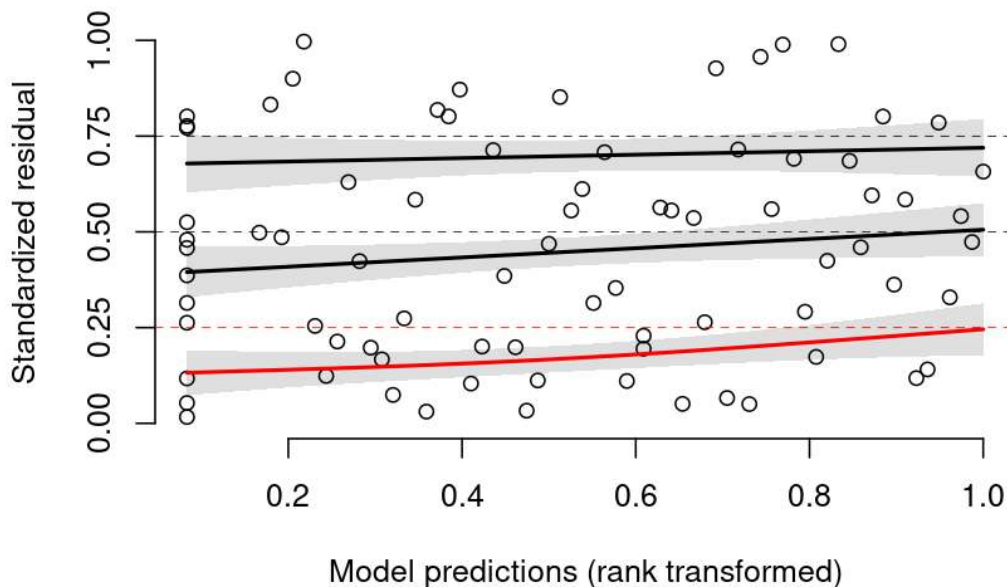


DHARMA residual diagnostics

QQ plot residuals



Residual vs. predicted  
Quantile deviations detected (red curves)  
Combined adjusted quantile test n.s.





# Binary Data (0/1) - Binomial Family (logistic regression)

```
summary(m)
```

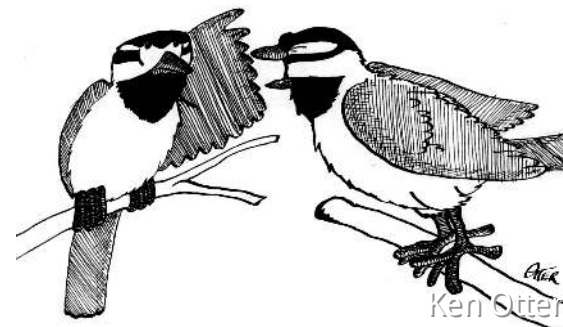
```
##
## Call:
## glm(formula = atypical ~ urbanization, family = "binomial",
data = atypical)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.9907  -0.4460  -0.2500  -0.2210   2.7201
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -2.6572     0.5380  -4.939 7.85e-07 ***
## urbanization    1.1000     0.4209   2.613 0.00897 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 51.586  on 77  degrees of freedom
## Residual deviance: 43.203  on 76  degrees of freedom
## AIC: 47.203
```

## Interpreting Results

```
exp(coef(m))
```

```
## (Intercept) urbanization
##  0.07014254   3.00406665
```

E.g., The odds of being an atypical singer increase by a factor of 3 (x3 times more likely) for every unit increase in Habit Urbanization.



# Binary Outcomes - Binomial Family

Proportion with binary outcomes (e.g., 10 yes, 5 no)

## Get the data

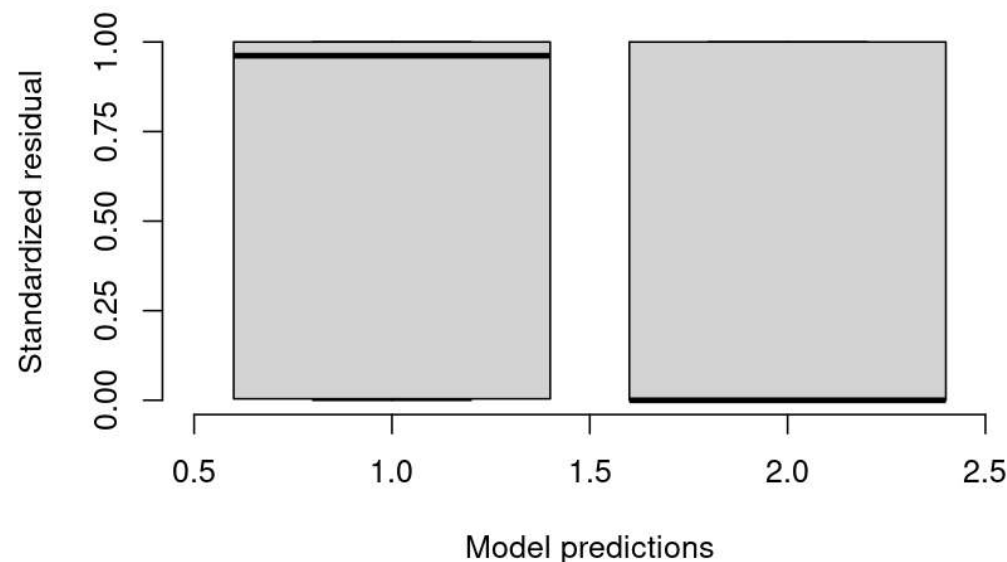
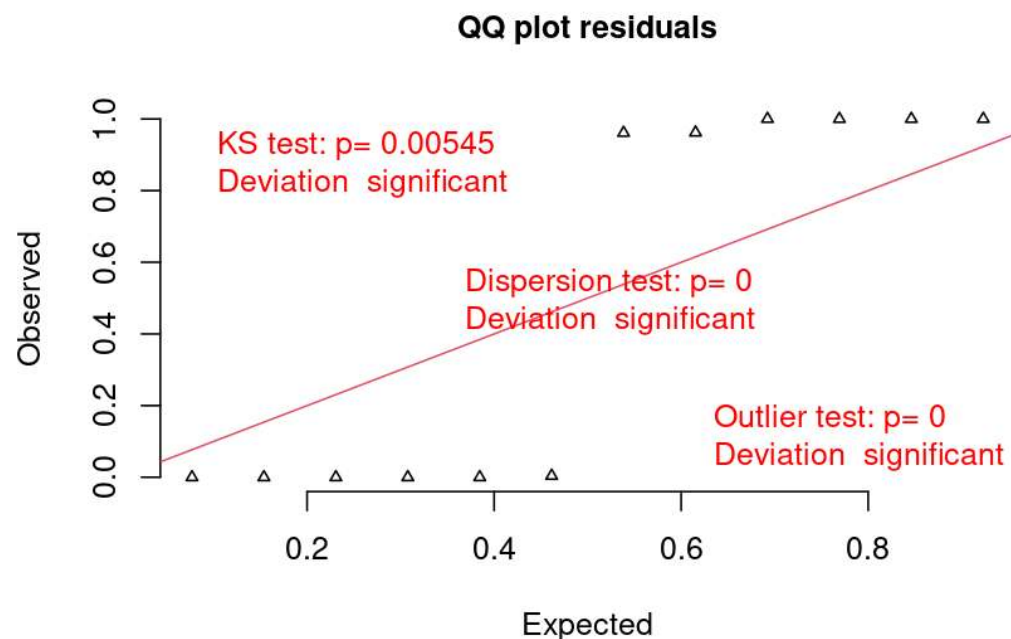
```
admissions <- as.data.frame(UCBAdmissions)
admissions <- pivot_wider(admissions, names_from = Admit, values_from = Freq)
admissions
```

```
## # A tibble: 12 x 4
##   Gender Dept Admitted Rejected
##   <fct> <fct>    <dbl>    <dbl>
## 1 Male   A         512      313
## 2 Female A          89       19
## 3 Male   B        353      207
## 4 Female B         17        8
## 5 Male   C        120      205
## 6 Female C        202      391
## 7 Male   D        138      279
## 8 Female D        131      244
## 9 Male   E         53      138
```

# Binary Outcomes - Binomial Family

## Run the model and check diagnostics

```
m <- glm(cbind(Admitted, Rejected) ~ Gender, family = "binomial", data = admissions)
plotQQunif(m)
plotResiduals(m, asFactor = TRUE) # to ensure Gender is treated as category
```



# Binary Outcomes - Binomial Family

```
summary(m)
```

```
##
## Call:
## glm(formula = cbind(Admitted, Rejected) ~ Gender, family =
## "binomial",
##      data = admissions)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -16.7915   -4.7613   -0.4365    5.1025   11.2022
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.22013    0.03879  -5.675 1.38e-08 ***
## GenderFemale -0.61035    0.06389  -9.553  < 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: 877.06  on 11  degrees of freedom
## Residual deviance: 783.61  on 10  degrees of freedom
```

## Traditional check for overdispersion

- Look at deviance (783.61) vs. df (10)

```
deviance(m) / df.residual(m)
```

```
## [1] 78.3607
```

- Very large (definitely not close to 1)

```
pchisq(deviance(m), df.residual(m),
        lower.tail = FALSE)
```

```
## [1] 6.892992e-162
```

- Test shows significant overdispersion

# Binary Outcomes - Binomial Family

```
summary(m)
```

```
##
## Call:
## glm(formula = cbind(Admitted, Rejected) ~ Gender, family =
## "binomial",
##      data = admissions)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -16.7915   -4.7613   -0.4365    5.1025   11.2022
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.22013    0.03879  -5.675 1.38e-08 ***
## GenderFemale -0.61035    0.06389  -9.553 < 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: 877.06  on 11  degrees of freedom
## Residual deviance: 783.61  on 10  degrees of freedom
```

## Traditional check for overdispersion

- Look at deviance (783.61) vs. df (10)

```
deviance(m) / df.residual(m)
```

```
## [1] 78.3607
```

- Very large (definitely not close to 1)

```
pchisq(deviance(m), df.residual(m),
        lower.tail = FALSE)
```

```
## [1] 6.892992e-162
```

- Test shows significant overdispersion

Okay, how do we deal with this?

# Binary Outcomes - Quasi-binomial Family for overdispersion

```
m <- glm(cbind(Admitted, Rejected) ~ Gender, family = "quasibinomial", data = admissions)
summary(m)
```

```
##
## Call:
## glm(formula = cbind(Admitted, Rejected) ~ Gender, family = "quasibinomial",
##      data = admissions)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -16.7915   -4.7613   -0.4365    5.1025   11.2022
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.2201     0.3281  -0.671   0.517
## GenderFemale  -0.6104     0.5404  -1.129   0.285
##
## (Dispersion parameter for quasibinomial family taken to be 71.52958)
##
##      Null deviance: 877.06  on 11  degrees of freedom
## Residual deviance: 783.61  on 10  degrees of freedom
## AIC: NA
```

# Binary Outcomes - Mixed models for overdispersion

- Observation-level random effects (i.e assign each observation an ID number)

```
library(lme4)
admissions <- mutate(admissions, ID = 1:n())
m <- glmer(cbind(Admitted, Rejected) ~ Gender + (1|ID), family = "binomial", data = admissions)
```

# Binary Outcomes - Mixed models for overdispersion

- Observation-level random effects (i.e assign each observation an ID number)

```
library(lme4)
admissions <- mutate(admissions, ID = 1:n())
m <- glmer(cbind(Admitted, Rejected) ~ Gender + (1|ID), family = "binomial", data = admissions)
summary(m)$coefficients
```

| ##              | Estimate   | Std. Error | z value   | Pr(> z )  |
|-----------------|------------|------------|-----------|-----------|
| ## (Intercept)  | -0.6508845 | 0.4932596  | -1.319558 | 0.1869827 |
| ## GenderFemale | 0.1747094  | 0.7016048  | 0.249014  | 0.8033499 |



# **Packages and References for Other Advanced Models**

# Packages and References

## (Generalized) Linear Mixed Models (LMM, GLMM)

- Also called generalized linear mixed effects models (GLME, LME)
- **lme4** - More advanced, crossed-random factors, Generalized (**glmer()**) and Gaussian (**lmer()**)
- **nlme** - Older but can specify auto-correlation structures, only Gaussian (**lme()**)
- **glmmTMB** - Zero-inflated models and other distributions

## References

- [Ben Bolker's GLMM FAQ](#).
- [Chapter 19, "The R Book" by Michael J. Crawley](#).  
(Freely available online through University of Manitoba Library)
- [Mixed Effects Models and Extensions in Ecology with R by Alain Zuur](#)  
(Freely available online through University of Manitoba Library)
- [Generalized linear mixed models: a practical guide for ecology and evolution](#), 2009, Trends in ecology and evolution

# Packages and References

## General Additive Models (GAM)

- **mgcv** package (`gam()`, `gamm()`)
- **gamm4** package (`gamm4()`)

## References

- [Chapter 19, "The R Book" by Michael J. Crawley](#)

(Freely available online through University of Manitoba Library)

- "Generalized Additive Models: An Introduction with R" by Simon N. Wood

(Hard-copy available from University of Manitoba Library)

# Packages and References

## Generalized Estimating Equations (GEE)

- **gee** package (**gee()**)
- **geepack** package (**geeglm()**)

## References

- [The R package geepack for Generalized Estimating equations](#), Journal of Statistical Software, 2005
- [geepack Manual](#)

# **Non-parametric Statistics**

# Non-parametric Statistics

## Wilcoxon Rank Sum (Mann-Whitney) Test

```
air <- filter(airquality, Month %in% c(5, 8))
```

Is there a difference in air quality between May (5th month) and August (8th month)?

```
wilcox.test(Ozone ~ Month, data = air, exact = FALSE)
```

```
##  
##      Wilcoxon rank sum test with continuity correction  
##  
## data:  Ozone by Month  
## W = 127.5, p-value = 0.0001208  
## alternative hypothesis: true location shift is not equal to 0
```

Yes!

# Non-parametric Statistics

## Kruskal-Wallis Rank Sum Test

Is there a difference in air quality among months?

```
kruskal.test(Ozone ~ Month, data = airquality)
```

```
##  
##      Kruskal-Wallis rank sum test  
##  
## data:  Ozone by Month  
## Kruskal-Wallis chi-squared = 29.267, df = 4, p-value = 6.901e-06
```

Yes, there is at least one month that is different from the rest.