

Unit 5 Live Session (Solutions)

Discrete Response Model Part 5



Figure 1: South Hall

Class Announcements

No HW this week

Lab-1 due in 1 week

Roadmap

Rearview Mirror

- Model unordered and ordered categorical response

Today

- Poisson probability model
- Poisson regression model, estimation, and statistical inference
- Model Comparison Criteria, Model Assessment, Goodness of Fit

Looking Ahead

- Univariate and multivariate time-series
- Notion of dependency and stationarity

Start-up Code

```
# Insert the function to *tidy up* the code when they are printed out
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=60), tidy=TRUE)

# Start with a clean R environment
rm(list = ls())

# Load libraries
## Load a set of packages including: broom, cli, crayon, dbplyr , dplyr, dtplyr,forcats,
## googledrive, googlesheets4, ggplot2, haven, hms, httr, jsonlite, lubridate , magrittr,
## modelr, pillar, purrr, readr, readxl, reprex, rlang, rstudioapi, rvest, stringr, tibble,
## tidyverse, xml2
if(!"tidyverse"%in%rownames(installed.packages())) {install.packages("tidyverse")}
library(tidyverse)

## provide useful functions to facilitate the application and interpretation of regression analysis.
if(!"car"%in%rownames(installed.packages())) {install.packages("car")}
library(car)

## provides many functions useful for data analysis, high-level graphics, utility operations like describe()
if(!"Hmisc"%in%rownames(installed.packages())) {install.packages("Hmisc")}
library(Hmisc)

##Tools for performing model selection like AICc()
if(!"gamlr"%in%rownames(installed.packages())) {install.packages("gamlr")}
library(gamlr)

## To generate regression results tables and plots
if(!"finalfit"%in%rownames(installed.packages())) {install.packages("finalfit")}
library(finalfit)

## To produces LaTeX code, HTML/CSS code and ASCII text for well-formatted tables
if(!"stargazer"%in%rownames(installed.packages())) {install.packages("stargazer")}
library(stargazer)

## For overdispersion test
if(!"AER"%in%rownames(installed.packages())) {install.packages("AER")}
```

```
library(AER)

## For negative binomial regression
if(!"MASS"%in%rownames(installed.packages())) {install.packages("MASS")}
library(MASS)
```

Poission Distribution Review

Recall that the Poisson distribution models count data i.e. the number of events between $0, 1, \dots$ for a random variable X . The distribution is:

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

One key assumption of the distribution is that $E(X) = Var(X) = \lambda$, meaning that the mean and variance of the distribution is the same. As we will see this is a limiting assumption when we do Poission Regression.

Poission Regression Review

In Poisson regression we model the log of λ (the mean) as a linear combination of the features:

$$\log(\lambda_i) = \log(E(Y_i|X_i)) = X_i\beta$$

We use maximum likelihood to estimate the coefficients in β assuming that Y follows a Poisson distribution:

$$\max_{\beta} L(\beta|Y_1, \dots, Y_n) = P(Y_1 = y_1, \dots, Y_n = y_n|\beta) = \prod_i \frac{e^{y_i X_i \beta} e^{-e^{X_i \beta}}}{y_i!}$$

The log likelihood has no closed form solution, so we estimate the parameters beta using numerical methods just like in logistic regression.

For each y_i we calculate and predict:

$$\hat{y}_i = E(y_i|X_i) = \lambda_i = e^{X_i \beta}$$

This also means that the $Var(\hat{y}_i) = \lambda_i = e^{X_i \beta}$ so that Poisson regression naturally has heteroskedasticity in the results.

This assumption of equal mean and variance is often not met when actually fitting to data, which results in what is known as overdispersion where the variance in the data is larger than the variance fit in the model. This usually can be remedied by adding more X variables into the model to improve the fit.

Another option is to fit what are known as a quasi poisson model or negative binomial regression model. In both cases, we relax the equal mean and variance assumption by adding an additional parameter to the variance of the response variable, allowing it to be larger than the mean. In quasi poisson regression we set $Var(\hat{y}_i) = \theta \lambda_i$ and in negative binomial regression we set $Var(\hat{y}_i) = \lambda_i + \kappa \lambda_i^2$.

Also, because coefficients are on the log scale, when exponentiated they multiply the expected mean outcome. This is similar to the interpretation in logistic regression except we are not dealing with odds ratios but rather the average outcome.

Case Study: Modeling the Number of Awards

Introduction

Imagine we are trying to model the number of awards earned by students based on the type of programs the student was enrolled in using historical admission data. The awards committee provides a small data sample that includes the score of the final math exam in previous years.

Data Description

The dataset *PossionEx1.csv* contains the following variables:

- num_awards: the number of awards earned by students at a high school in a year
- math: students' scores on their final math exam
- prog: the type of program in which the students were enrolled (1 = “General”, 2 = “Academic” and 3 = “Vocational”).

Descriptive Statistics

Some questions to ask when exploring the dataset:

- What is the number of observations?
- What is the number of variables?
- Are there any redundant variables?
- Are there any missing information?
- Are there any duplicated records?
- Are there any values in each of the variables that seem unreasonable?

```
df <- read.csv("./data/PossionEx1.csv", stringsAsFactors = F, header=TRUE, sep=",")  
  
head(df) %>%  
  knitr::kable()
```

X	id	num_awards	prog	math
1	45	0	Vocational	41
2	108	0	General	41
3	15	0	Vocational	44
4	67	0	Vocational	42
5	153	0	Vocational	40
6	51	0	General	42

```
#str(df)

## convert prog to factor
df$prog = factor(df$prog)

# Checking the number of missing values for each of the variables
# df[!complete.cases(df),]
sapply(df, function(x) sum(is.na(x)))

##          X      id num_awards      prog      math
##          0       0        0       0       0

# Attach the dataset
attach(df)

#describe(df)
```

Univariate Analysis

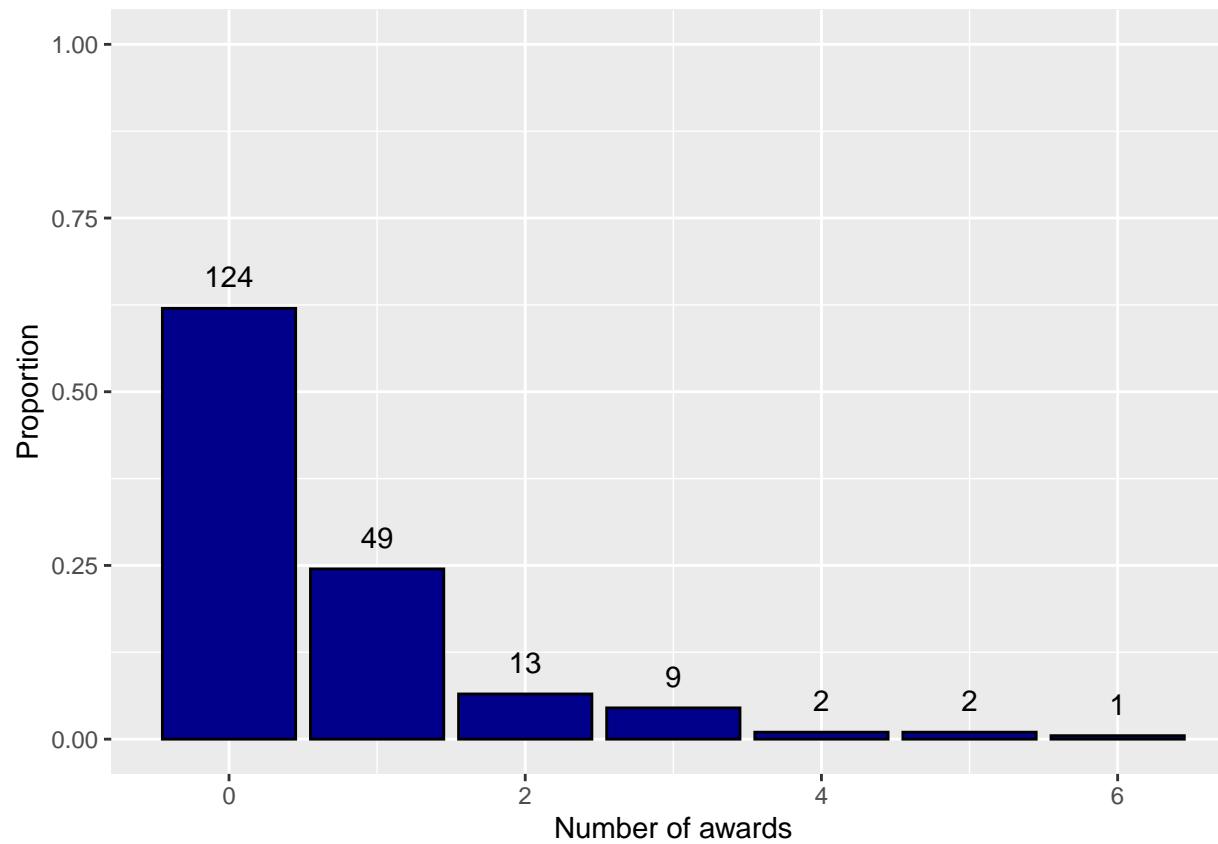
- Use a frequency table and a bar plot to explore the distribution of the response variable(num_awards). What do you learn?

```
df %>%
  count(num_awards) %>%
  mutate(prop = round(prop.table(n), 2)) %>%
  kable(col.names = c('Number of awards', 'N', "Proportion"))
```

Number of awards	N	Proportion
0	124	0.62
1	49	0.24
2	13	0.06
3	9	0.04
4	2	0.01
5	2	0.01
6	1	0.00

```
df %>%
  ggplot(aes(x= num_awards, y = ..prop.., group = 1)) +
  geom_bar(fill = 'DarkBlue', color = 'black') +
  geom_text(stat='count', aes(label=..count..), vjust=-1) +
  xlab("Number of awards") +
  ylab("Proportion") +
  ylim(0,1)

## Warning: The dot-dot notation ('..prop..'') was deprecated in ggplot2 3.4.0.
## i Please use 'after_stat(prop)' instead.
```



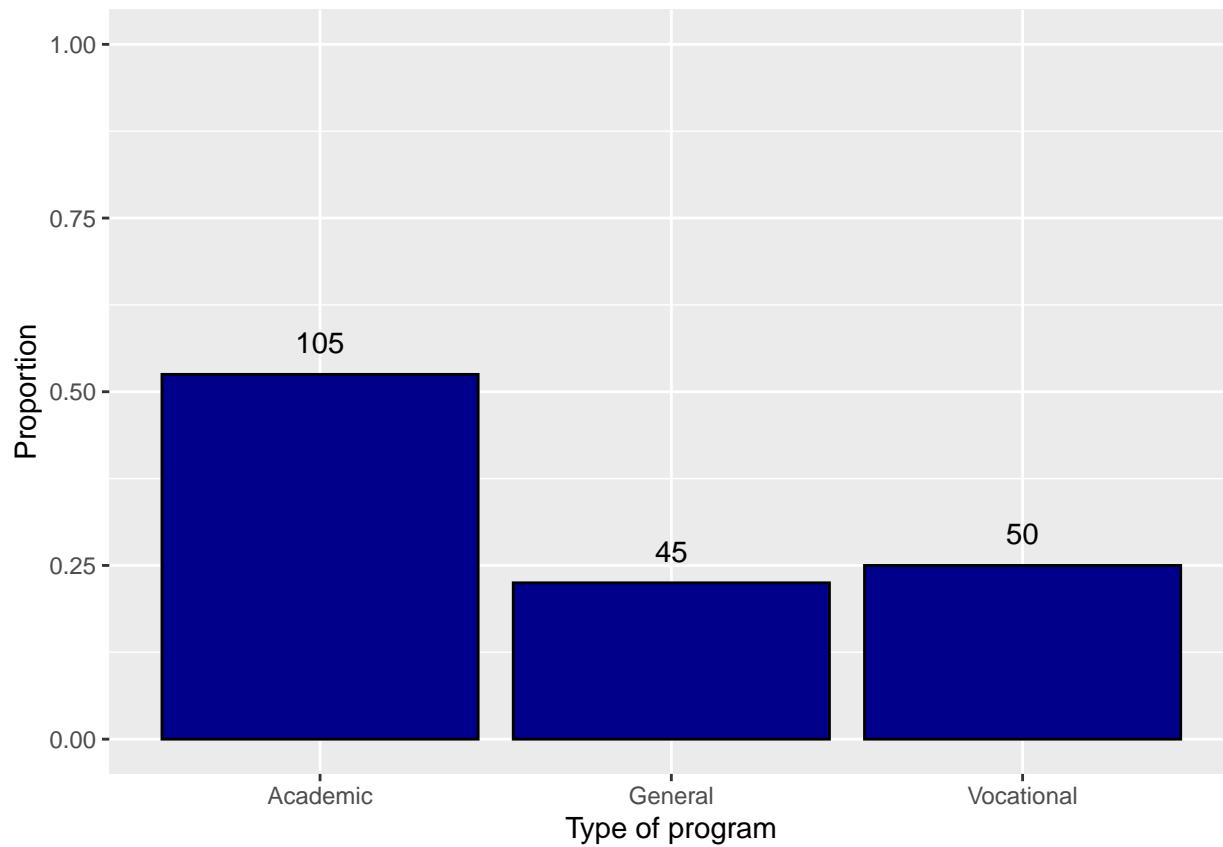
The distribution of the number of awards is pretty skewed with a long right tail, and 62% of the students didn't earn any awards, 24% earned one award, and 14% earned more than one award.

- The prog is the committee's critical explanatory variable of interest. It has three levels: academic, general, and vocational. Use a frequency table and a bar plot to examine its distribution. What do you discover?

```
df %>%
  count(prog) %>%
  mutate(prop = round(prop.table(n), 2)) %>%
  kable(col.names = c(' Type of program', 'N', "Proportion"))
```

Type of program	N	Proportion
Academic	105	0.52
General	45	0.22
Vocational	50	0.25

```
df %>%
  ggplot(aes(x= prog, y = ..prop.., group = 1)) +
  geom_bar(fill = 'DarkBlue', color = 'black') +
  geom_text(stat='count', aes(label=..count..), vjust=-1) +
  xlab("Type of program") +
  ylab("Proportion") +
  ylim(0,1)
```



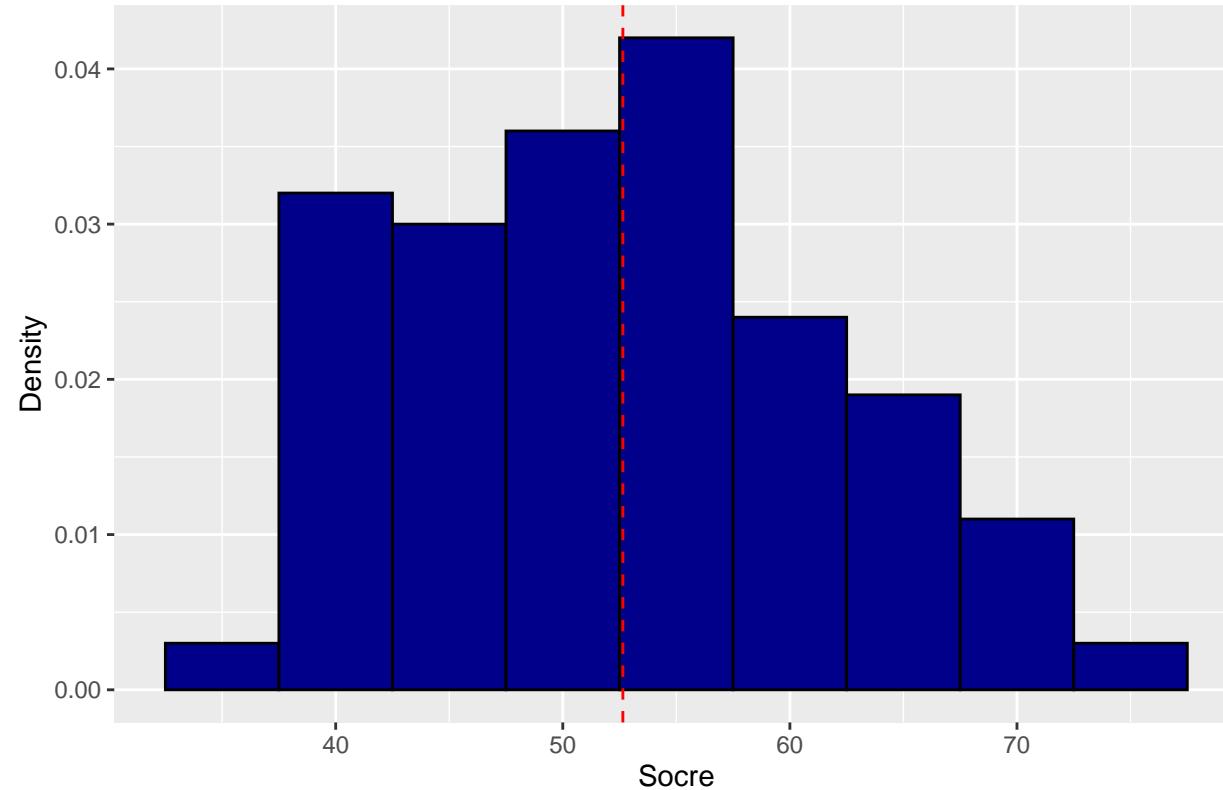
Almost half of the students (52%) enrolled in the academic program, and the other half equally enrolled in general and vocational programs.

- Plot the distribution of math scores. What are the range and average math scores?

```
df %>%
  ggplot(aes(x = math)) +
  geom_histogram(aes(y = ..density..), binwidth = 5 ,fill = "DarkBlue", color = "black") +
  geom_vline(aes(xintercept = mean(math)), color = "red", linetype = "dashed") +
  ggtitle("Distribution of students' math scores") +
  theme(plot.title = element_text(lineheight=1, face="bold")) +
  xlab("Socre") +
```

```
ylab("Density")
```

Distribution of students' math scores

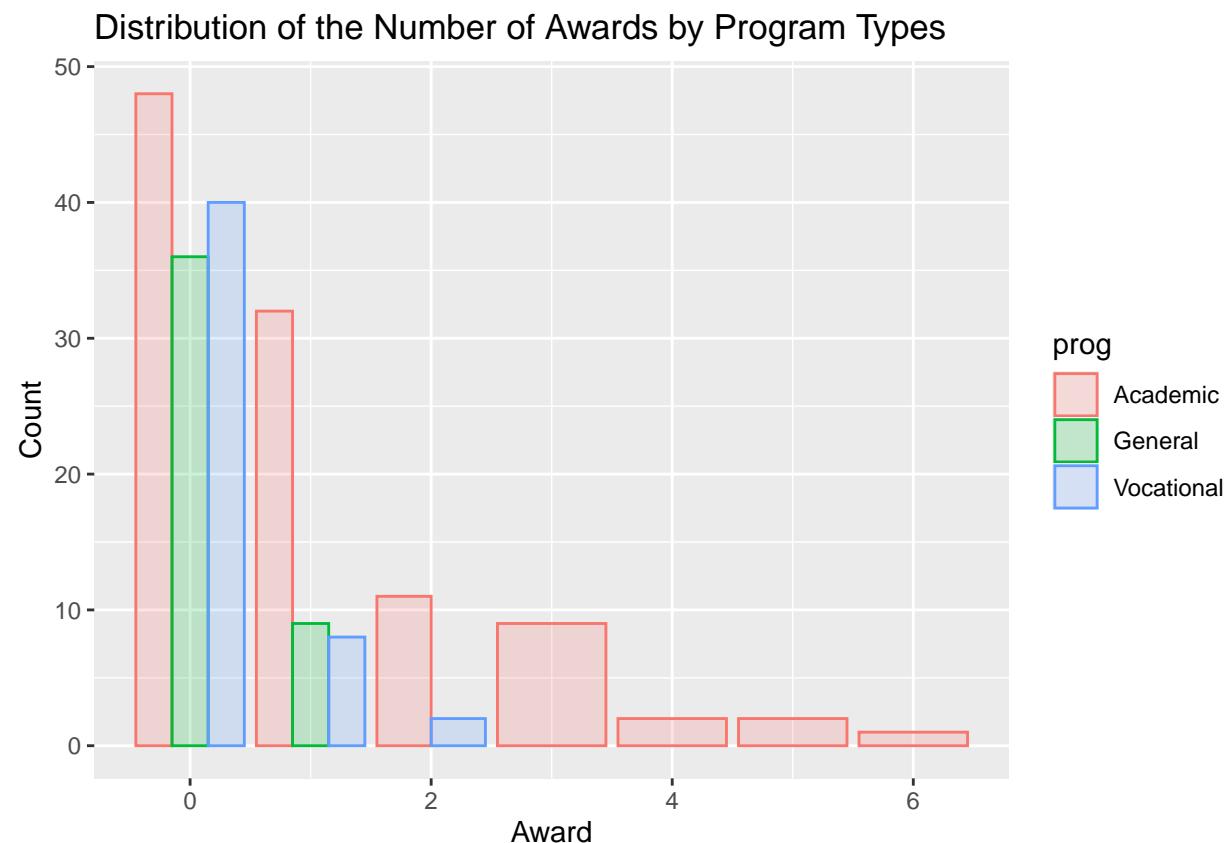


The distribution of math scores is not skewed, with an average of almost 53

Bivariate Analysis

- Examine the associations between the number of awards and program and math scores.
- The graph below shows the distribution of the number of awards by program types. How are awards distributed among different programs?

```
df %>%
  ggplot(aes(x = num_awards)) +
  geom_bar(aes(color = prog, fill = prog), alpha=0.2, position = "dodge" ) +
  ggtitle("Distribution of the Number of Awards by Program Types") +
  xlab("Award") +
  ylab("Count")
```

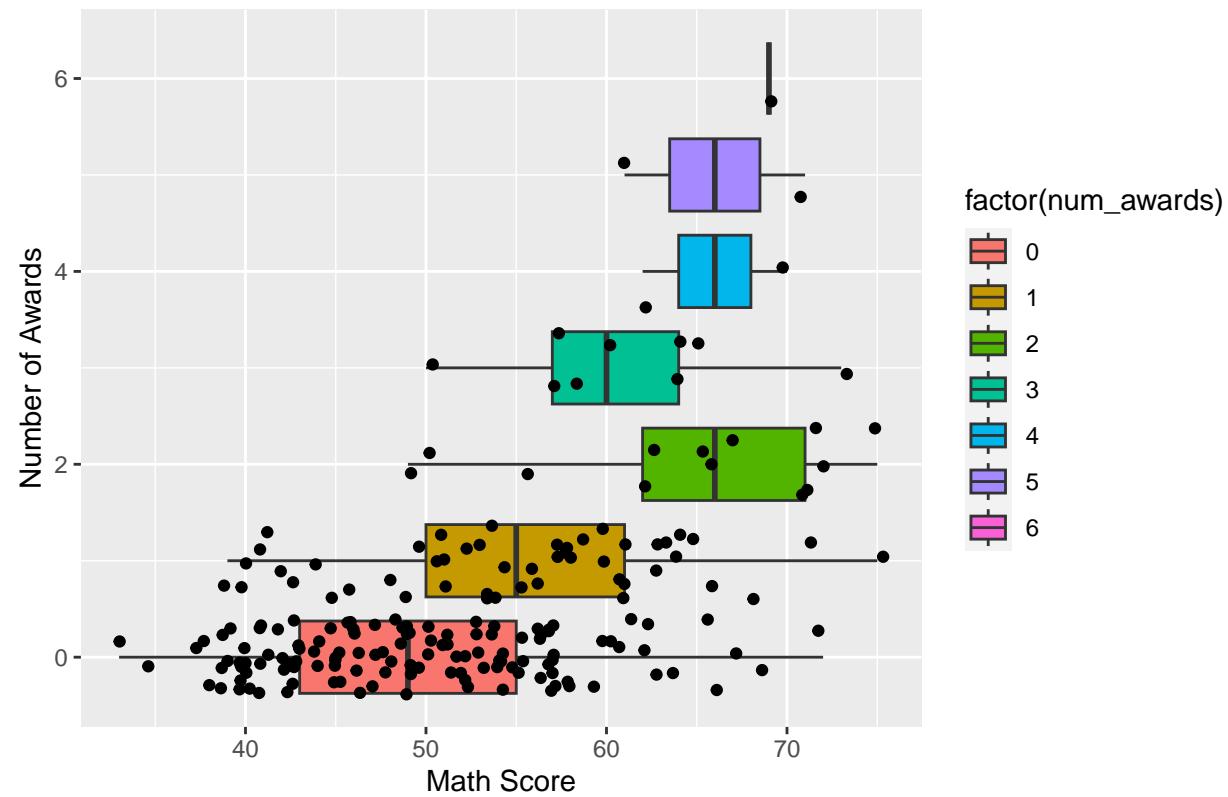


It suggests that students in the academic program eaned more awards, and prog type is a good candidate for predicting the number of awards because the value of the outcome appears to vary by prog.

- The graph below shows the distribution of the number of awards and students' math scores. Is there any clear relationship between them?

```
df %>%
  ggplot(aes(num_awards, math)) +
  geom_boxplot(aes(fill = factor(num_awards))) +
  geom_jitter() +
  coord_flip() +
  ggtitle("Math Score by the Number of Awards") +
  theme(plot.title = element_text(lineheight=1, face="bold")) +
  ylab("Math Score") +
  xlab("Number of Awards")
```

Math Score by the Number of Awards



The box plot shows that students with higher scores on their final math exam earned more awards, so math scores could also be a good predictor of the number of awards.

- Use summary_factorlist() function from the finalfit package to tabulate data. What do you learn from the EDA?

```
dependent <- "num_awards"
explanatory <- c("prog", "math")
df %>%
  summary_factorlist(dependent, explanatory, add_dependent_label = TRUE) %>%
  knitr::kable()
```

Dependent: num awards		unit	value
prog	Academic	Mean (sd)	1.0 (1.3)
	General	Mean (sd)	0.2 (0.4)
	Vocational	Mean (sd)	0.2 (0.5)
math	[33.0,75.0]	Mean (sd)	0.6 (1.1)

Model Development

- Given the specification of a poisson regression model below,

$$\log(\text{mean_num_awards}) = \beta_0 + \beta_1 \text{prog} + \beta_2 \text{math} + u$$

- Estimate and interpret the model results:

```
poisson.mod.1 <- glm(num_awards ~ prog + math, data = df, family = poisson)
summary(poisson.mod.1)
```

```
##
## Call:
## glm(formula = num_awards ~ prog + math, family = poisson, data = df)
##
## 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) -4.16327   0.66288 -6.281 3.37e-10 ***
## progGeneral -1.08386   0.35825 -3.025  0.00248 **
## progVocational -0.71405   0.32001 -2.231  0.02566 *
## math         0.07015   0.01060  6.619 3.63e-11 ***
## ---
## 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
## AIC: 373.5
##
## Number of Fisher Scoring iterations: 6
(exp(coef(poisson.mod.1))-1)*100

##          (Intercept)    progGeneral    progVocational           math
##        -98.444332     -66.171250     -51.034289      7.267164
```

The negative coefficients of the general and vocational programs indicate that the number of wards is lower in these two programs
The positive coefficients of math score indicates that the number of awards is increasing as the math score increases To have a more convenient way to interpret these coefficients, we compute and use percentage changes

Holding program type constant, 1 unit increase in math score increase the mean number of awards by 7%.

Hold math score constant; a student in the general program, on average, receives 66% fewer awards than students in the academic program

Hold math score constant; a student in the vocational program, on average, receives 51% fewer awards than students in the academic program

- Construct and interpret the confidence intervals for each variable.

```
# Confident intervals for the original coefficient estimates
beta.interval <- confint(poisson.mod.1, level = 0.95)

## Waiting for profiling to be done...
# Convert the confidence intervals to percentage change, corresponding to the coefficient estimates

100*(exp(beta.interval)-1)
```

```
##           2.5 %    97.5 %
## (Intercept) -99.585448 -94.418558
## progGeneral   -84.368296 -35.276377
## progVocational -75.106850 -11.852828
## math            5.074217   9.535332
```

With 95% confidence, a 1-unit increase in math score leads to a 5.07% to 9.53% increase in the mean number of awards, holding the program constant.

With 95% confidence, the mean number of awards decreases by 35% to 84% for students in the General program vs. students in the academic program, holding the math score constant.

With 95% confidence, the mean number of awards decrease by 11% to 75% for the student in vocational program vs. student in academic programs, holding the math score constant.

- Test the overall effect of prog,

```
Anova(poisson.mod.1)

## Analysis of Deviance Table (Type II tests)
##
## Response: num_awards
```

```
##      LR Chisq Df Pr(>Chisq)
## prog   14.572  2  0.0006852 ***
## math   45.010  1   1.96e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Based on the p-values, the **prog**, taken together, is a statistically significant predictor of the number of awards.

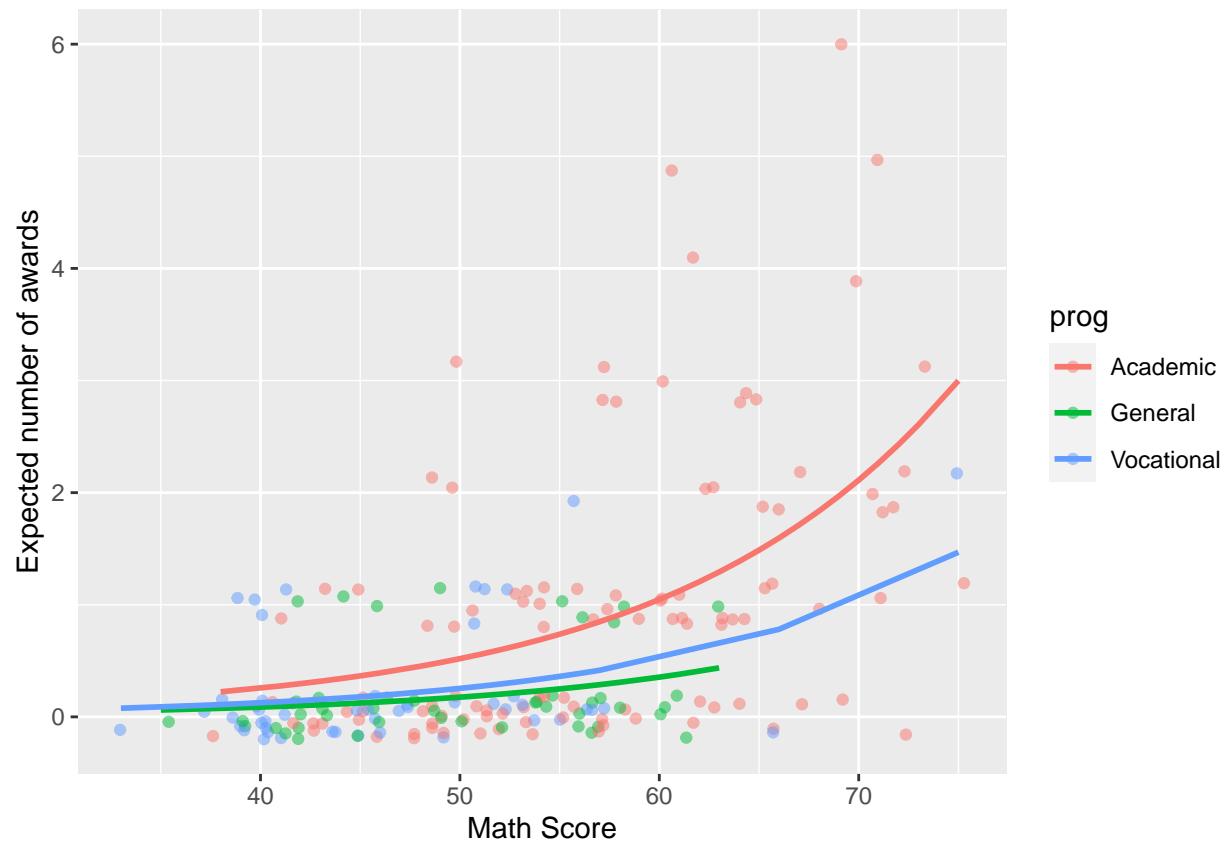
- Plot the fitted values across the three programs and discuss how the number of awards is associated with math scores.

```
## calculate and store predicted values
fitted_values <- predict(poisson.mod.1, type="response")

## Create the plot

ggplot(df, aes(x = math, y = fitted_values, colour = prog)) +
  geom_point(aes(y = num_awards), alpha=.5, position=position_jitter(h=.2)) +
  geom_line(size = 1) +
  labs(x = "Math Score", y = "Expected number of awards")
```

Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use 'linewidth' instead.



This graph indicates that the most awards are earned by students in the academic program, especially if the student has a high math score. Students in the general program earn the lowest number of awards

Model Comparison Criteria

Recall that the general form of most information criteria is:

$$IC(k) = -2\log(L(\hat{\beta}|y_1, \dots, y_n)) + kr$$

Where $\log(L(\hat{\beta}|y_1, \dots, y_n))$ is the log-likelihood of an estimated model, n is the sample size, and r is the number of parameters in the model.

The three most common information criteria are:

$$AIC = IC(2) = -2\log(L(\hat{\beta}|y_1, \dots, y_n)) + 2r$$

$$AIC_c = IC\left(\frac{2n}{n-r-1}\right) = -2\log(L(\hat{\beta}|y_1, \dots, y_n)) + r\frac{2n}{n-r-1} = AIC + \frac{2r(r+1)}{n-r-1}$$

$$BIC = IC(\log(n)) = -2\log(L(\hat{\beta}|y_1, \dots, y_n)) + r\log(n)$$

- Compute these three information criteria for the following three models and rank models based on each criterion.

$$\log(mean_num_awards) = \beta_0 + \beta_1 prog + u$$

$$\log(mean_num_awards) = \beta_0 + \beta_1 math + u$$

$$\log(mean_num_awards) = \beta_0 + \beta_1 prog + \beta_2 math + u$$

```
mod.1 <- glm(num_awards ~ math, data = df, family = poisson)

mod.2 <- glm(num_awards ~ prog, data = df, family = poisson)

mod.3 <- glm(num_awards ~ prog + math, data = df, family = poisson)

## compute AIC
## use generic AIC()
data.frame(mod.1 = AIC(mod.1), k = 2, mod.2 = AIC(mod.2), mod.3 = AIC(mod.3))

##      mod.1    mod.2    mod.3
## 1 384.0762 416.5149 373.5045
```

```

## compute corrected AIC
data.frame(mod.1 = AICc(mod.1), mod.2 = AICc(mod.2), mod.3 = AICc(mod.3))

##      mod.1    mod.2    mod.3
## 1 384.1371 416.6373 373.7096

## compute BIC
data.frame(mod.1 = BIC(mod.1), mod.2 = BIC(mod.2), mod.3 = BIC(mod.3))

##      mod.1    mod.2    mod.3
## 1 390.6728 426.4098 386.6978

```

The model with the lowest AIC, corrected AIC, or BIC score is preferred. The absolute values of these scores do not matter. These scores can be negative or positive.

Based on all these three criteria, the third model with both math and program is the best, and the second model with the only program is the worst model

Model Assessment

Recall that the Pearson residual corrects for the unequal variance in the raw residuals by dividing by the standard deviation:

$$e_m = \frac{y_m - \hat{y}_m}{\sqrt{\widehat{Var}(Y_m)}}$$

Also standardized Pearson residual correct for overestimates the standard deviation of $y_m - \hat{y}_m$:

$$r_m = \frac{y_m - \hat{y}_m}{\sqrt{\widehat{Var}(Y_m - \hat{Y}_m)}} = \frac{y_m - \hat{y}_m}{\sqrt{\widehat{Var}(Y_m) - (1 - h_m)}}$$

where h_m is the mth diagonal element of the hat matrix>

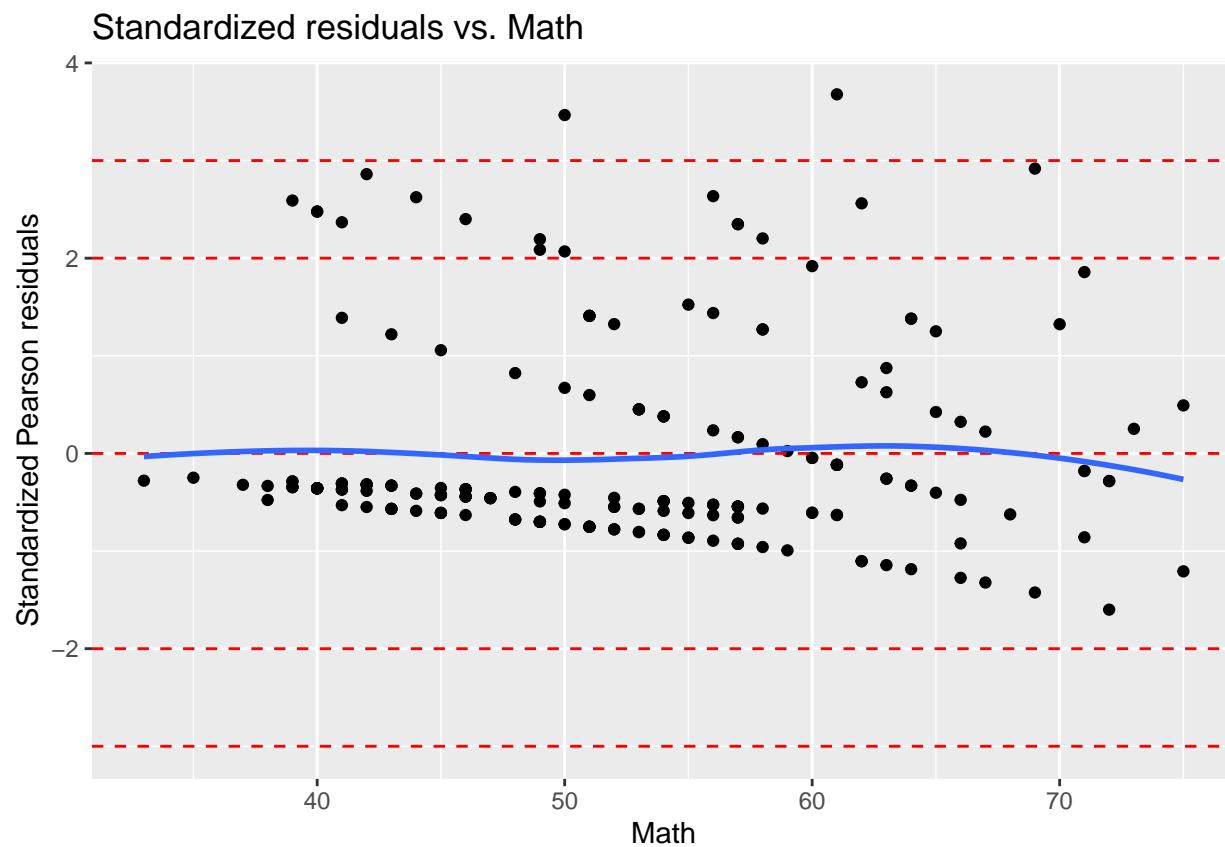
- For the first Poisson model using prog and math as predictors, plot the standardized Pearson residuals against explanatory variables, fitted values, and the linear predictor to assess whether the model assumptions are satisfied.

```
pred <- predict(poisson.mod.1 , type = "response")
res <- residuals(poisson.mod.1 , type = "pearson")
s.res <- rstandard(poisson.mod.1 , type = "pearson")
lin.pred <- poisson.mod.1$linear.predictors

df1 <- data.frame(df, pred ,res , s.res , lin.pred)

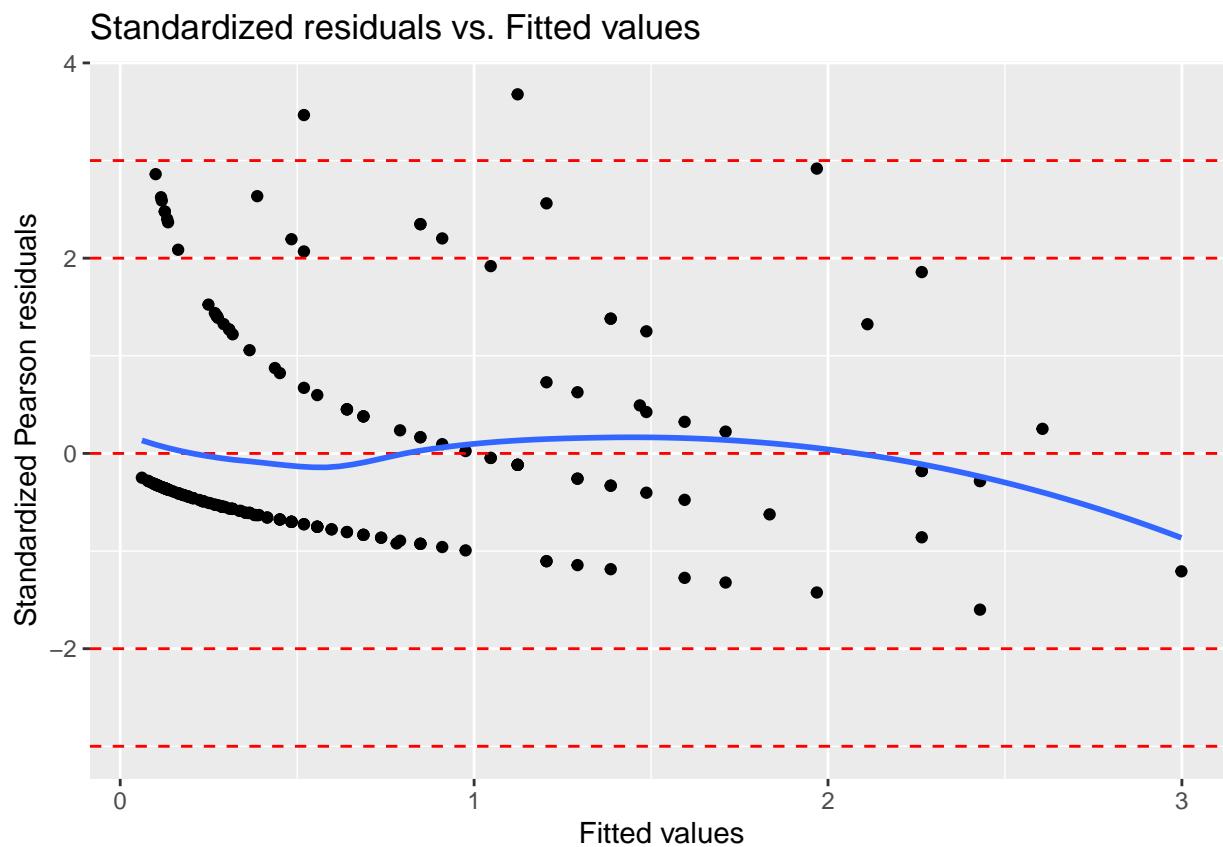
#Standardized Pearson residual vs. math plot

df1 %>%
  ggplot(aes(x = df1$math , y = df1$s.res)) +
  geom_point() +
  geom_hline(yintercept=c(3, 2, 0, -2, -3), color = "red", linetype = "dashed")+
  geom_smooth(se = FALSE) +
  ggtitle("Standardized residuals vs. Math") +
  xlab("Math") +
  ylab("Standardized Pearson residuals")
```



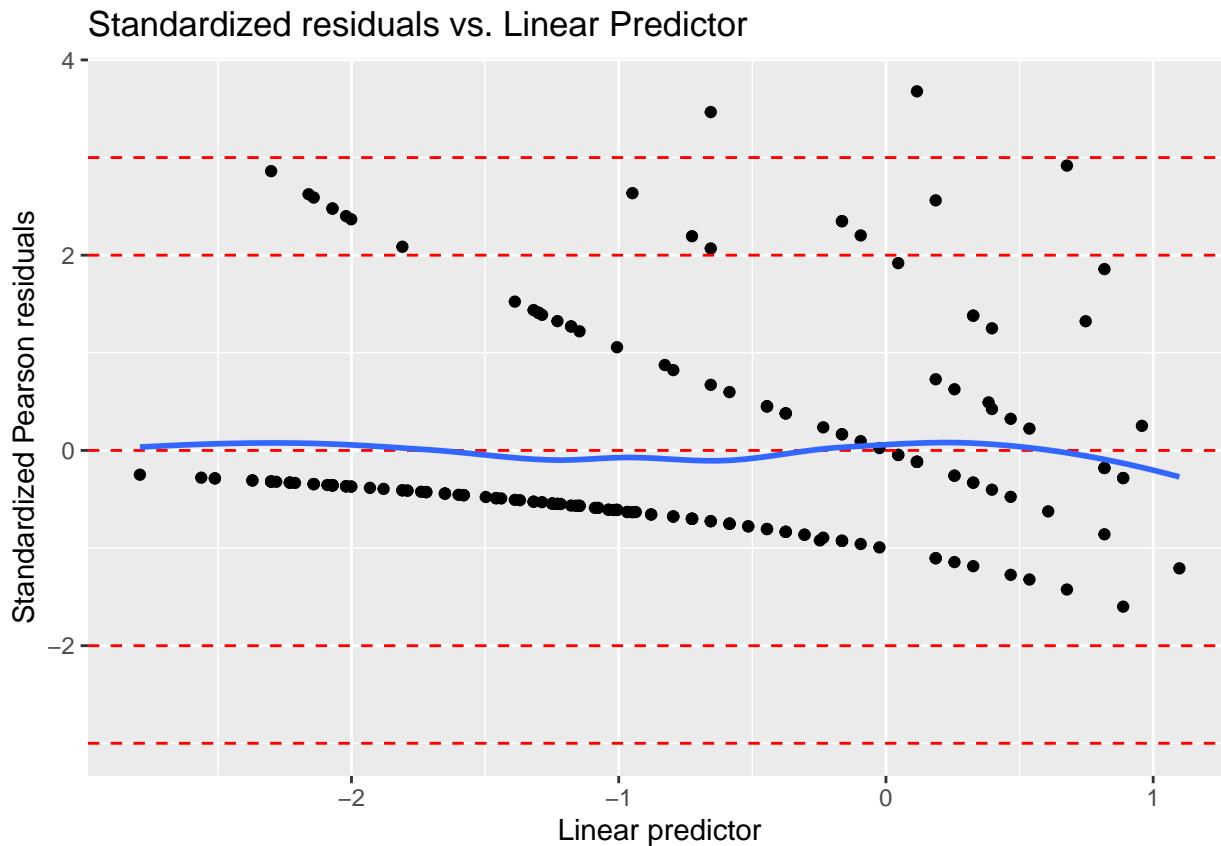
```
#Standardized Pearson residual vs. fitted values

df1 %>%
  ggplot(aes(x = df1$pred , y = df1$s.res)) +
  geom_point() +
  geom_hline(yintercept=c(3, 2, 0, -2, -3), color = "red", linetype = "dashed")+
  geom_smooth(se = FALSE) +
  ggtitle("Standardized residuals vs. Fitted values") +
  xlab("Fitted values") +
  ylab("Standardized Pearson residuals")
```



```
#Standardized Pearson residual vs. linear predictor

df1 %>%
  ggplot(aes(x = df1$lin.pred , y = df1$s.res)) +
  geom_point() +
  geom_hline(yintercept=c(3, 2, 0, -2, -3), color = "red", linetype = "dashed")+
  geom_smooth(se = FALSE) +
  ggtitle("Standardized residuals vs. Linear Predictor") +
  xlab("Linear predictor") +
  ylab("Standardized Pearson residuals")
```



We have four assumptions general assumptions for Poisson regression

- 1- IID data.
- 2- The distribution of the response is the one specified by the model, which here is Poisson.
- 3- The mean of the distribution is linked to the explanatory variables by the link function $g()$ that we specify. In Poisson is is $\log()$.
- 4- The link relates to the explanatory variables in a linear fashion. Here linearity means using a linear combination of the regression parameters.

For iid data, we must discuss how the data is generated and look for possible clustering. Here there could be some clustering among the students who are in the same program or same classroom, or same cohort, which would violate independence.

The plot of the standardized residuals against math shows roughly the same variance throughout the math score range. There is no severe curvature in the plot, suggesting we don't need a transformation or additional polynomial terms.

Similarly, the plots of residuals against the fitted values and linear predictor have roughly constant variance. They show no evident curvature, so we can conclude that the link function fits well here.

We can also use these plots to check for extreme residuals. In this case, there are numerous residuals whose magnitudes are larger than 2 or 3, scattered across the range of math scores. This may be a sign of overdispersion, meaning more variability in the counts than the model estimated. This is an indication that there may be important explanatory variables missing from the model

Goodness of Fit

The Pearson Statistic χ^2 and Residual Deviance D are often used to test the goodness of fit, where the null hypothesis is that our model is correct. Under asymptotic theory, both of these follow a chi-squared distribution with the same degrees of freedom as the residuals from the Poisson model.

We can also use these to test for overdispersion in our model since if our model is a good fit to the data we should not have overdispersion.

Recall:

$$\chi^2 = \sum_{i=1}^n \frac{(y_i - \exp\{\mathbf{X}_i\hat{\beta}\})^2}{\exp\{\mathbf{X}_i\hat{\beta}\}}$$

and

$$D = 2 \sum_{i=1}^n \left[y_i \log\left(\frac{y_i}{\exp\{\mathbf{X}_i\hat{\beta}\}}\right) - (y_i - \exp\{\mathbf{X}_i\hat{\beta}\}) \right]$$

Calculate the Pearson statistic and residual deviance and perform the goodness of fit test.

```
# Calculate Pearson statistic residuals
pearson_stat <- sum(residuals(poisson.mod.1, type = "pearson")^2)

# Get p value associated with the pearson statistic
pearson_p.value <- pchisq(pearson_stat, poisson.mod.1$df.residual)
pearson_p.value

## [1] 0.796014

#Calculate deviance p value
deviance_p.value <- pchisq(poisson.mod.1$deviance, poisson.mod.1$df.residual, lower.tail=FALSE)
deviance_p.value

## [1] 0.6182274
```

Goodness-of-fit statistics test is a more objective measure of the overall fit.

The null hypothesis is that the model is correct against the alternative that it is not.

We can use both Pearson statistic or the residual deviance to perform this test

Here, the two non significant p-values indicate that we fail to reject the null hypothesis that the model is correct

Directly Testing for Over Dispersion

We can use a dispersion test for Poisson regression from the AER package that tests the null hypothesis that $\theta = 1$ vs. not in a regression of the form $Var(Y_i) = (1 + \alpha) * \lambda_i$.

Note that if we set $(1 + \alpha) = \theta$ we get the variance form for a quasipoisson above. So this test is examining whether the variance of our outcome variable appears to come from a Poisson or quasipoisson distribution.

If we reject the null hypothesis due to a small p-value, we have overdispersion if $\alpha > 0$ and underdispersion (smaller variance in reality which is less common) if $\alpha < 0$.

The test itself reports the estimated dispersion value along with a p-value.

```
dispersiontest(poisson.mod.1)
```

```
##  
## Overdispersion test  
##  
## data: poisson.mod.1  
## z = 0.53224, p-value = 0.2973  
## alternative hypothesis: true dispersion is greater than 1  
## sample estimates:  
## dispersion  
## 1.047254
```

Note the dispersion estimate above 1 but lack of significant p-value since it is larger than 0.05. This means we fail to reject the null hypothesis that poission regression is value.

If we reject the null hypothesis in an overdispersion test, that means we should fit a quasipoisson or negative binomial regression.

We can fit quasipoisson directly using glm and specifying the appropriate family. For a negative binomial using regression, we need to use glm.nb from the MASS package.

```
# quasipoisson regression  
  
quasi.poisson <- glm(num_awards ~ prog + math, data = df, family = quasipoisson)
```

Note the dispersion estimate above 1 but lack of significant p-value since it is larger than 0.05. This means we fail to reject the null hypothesis that poission regression is value.

```
# negative binomial regression  
  
neg.binom <- glm.nb(num_awards ~ prog + math, data = df)
```

```

stargazer(poisson.mod.1, quasi.poisson, neg.binom, type="text")

##
## =====
##             Dependent variable:
## -----
##                 num_awards
## Poisson   glm: quasipoisson    negative
##           link = log            binomial
##             (1)          (2)          (3)
## -----
## progGeneral -1.084***     -1.084***     -1.075***
##               (0.358)        (0.373)        (0.367)
## 
## progVocational -0.714**     -0.714**     -0.708**
##                (0.320)        (0.333)        (0.334)
## 
## math         0.070***     0.070***     0.071***
##               (0.011)        (0.011)        (0.012)
## 
## Constant    -4.163***     -4.163***     -4.218***
##               (0.663)        (0.690)        (0.713)
## 
## -----
## Observations      200        200        200
## Log Likelihood -182.752          -182.905
## theta           6.115 (5.577)
## Akaike Inf. Crit. 373.505          373.811
## =====
## Note: *p<0.1; **p<0.05; ***p<0.01

```

This is not surprising given the test results above, but the coefficients from all three models are very similar. Note that quasipoisson and poisson should have identical coefficients since they only differ in the assumption about the dispersion parameter.

Reminders

1. Before the next live session:
 1. Complete and turn in the Lab-1
 2. Complete all videos and reading for unit 6