Inference and overdispersion

Last time: modeling article publication

We are interested in analyzing the number of articles published by biochemistry PhD students. The data contains the following variables:

- `art`: articles published in last three years of Ph.D.
- `fem`: gender (recorded as male or female)
- `mar`: marital status (recorded as married or single)
- `kid5`: number of children under age six
- `phd`: prestige of Ph.D. program
- `ment`: articles published by their mentor in last three years

$$Articles_i \sim Poisson(\lambda_i)$$

$$\log(\lambda_i) = \beta_0 + \beta_1 Female_i + \beta_2 Married_i + \beta_3 Kids_i + \beta_4 Prestige_i + \beta_5 Mentor_i$$

Recap: confidence interval

$$Articles_i \sim Poisson(\lambda_i)$$

$$\log(\lambda_i) = \beta_0 + \beta_1 Female_i + \beta_2 Married_i + \beta_3 Kids_i + \beta_4 Prestige_i + \beta_5 Mentor_i$$

Confidence interval for β_4:

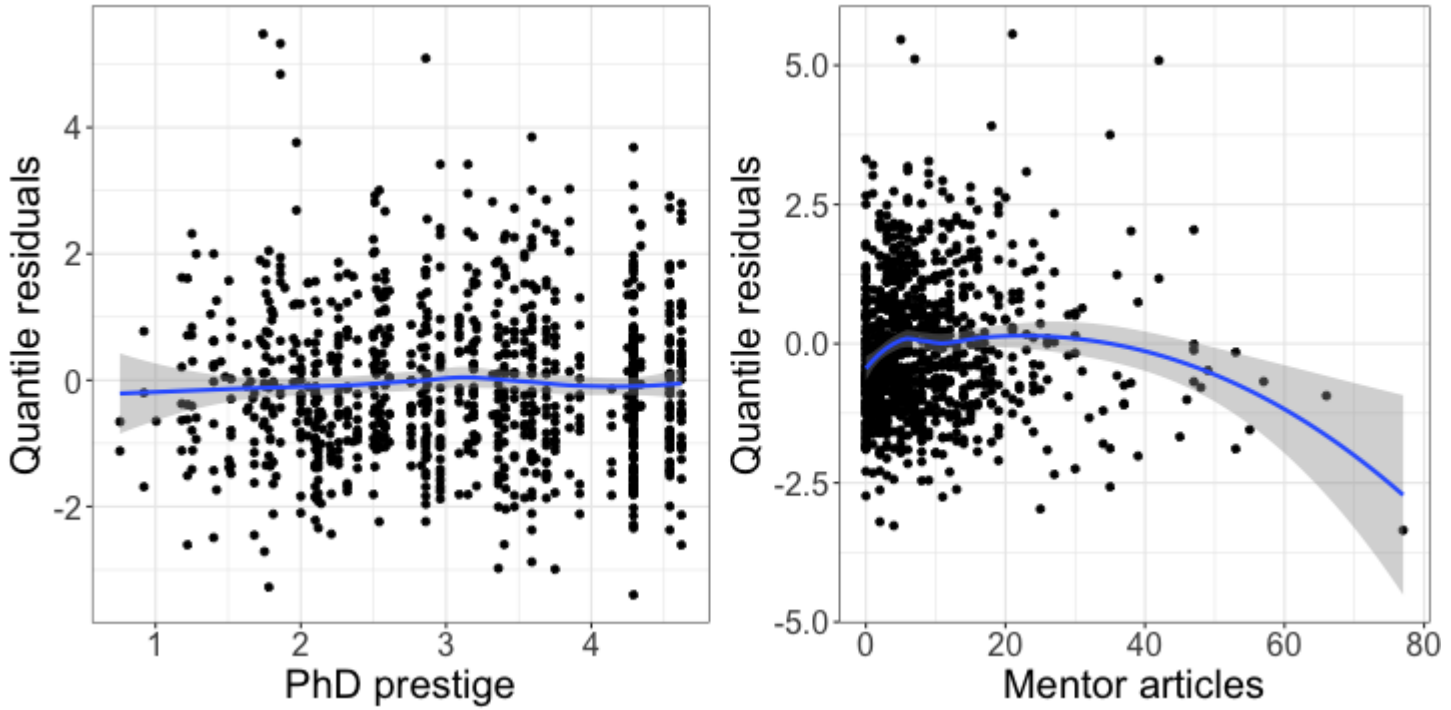
Recap: assumptions

$$Articles_i \sim Poisson(\lambda_i)$$

$$\log(\lambda_i) = \beta_0 + \beta_1 Female_i + \beta_2 Married_i + \beta_3 Kids_i + \beta_4 Prestige_i + \beta_5 Mentor_i$$

> What assumptions is this Poisson regression model making, and how do we check those assumptions?

Quantile residual plots



Do the model assumptions seem reasonable?

Goodness-of-fit test

```
summary(m1)
```

```
...
## (Intercept)  0.304617   0.102981   2.958   0.0031 **
## femWomen    -0.224594   0.054613  -4.112 3.92e-05 ***
## marMarried   0.155243   0.061374   2.529   0.0114 *
## kid5        -0.184883   0.040127  -4.607 4.08e-06 ***
## phd          0.012823   0.026397   0.486   0.6271
## ment         0.025543   0.002006  12.733  < 2e-16 ***
##
##     Null deviance: 1817.4  on 914  degrees of freedom
## Residual deviance: 1634.4  on 909  degrees of freedom
...
```

> How do I perform the goodness-of-fit test for this regression model?

Goodness-of-fit test

```
summary(m1)
```

```
...
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1817.4  on 914  degrees of freedom
## Residual deviance: 1634.4  on 909  degrees of freedom
...
```

```
pchisq(1634, 909, lower.tail=F)
```

```
## [1] 4.916386e-44
```

> Why might the model not be a good fit to the data?

Overdispersion

Overdispersion occurs when the response Y has higher variance than we would expect if Y followed a Poisson distribution.

Formally, let

$$\phi = \frac{\text{Variance}}{\text{Mean}}$$

> What should ϕ be if there is no overdispersion?

Estimating overdispersion

The *Pearson residual* for observation i is defined as

$$e_{(P)i} = \frac{Y_i - \widehat{\lambda}_i}{\sqrt{\widehat{\lambda}_i}}$$

$$\widehat{\phi} = \frac{\sum_{i=1}^{n} e^2_{(P)i}}{n - p}$$

- $p =$ number of parameters in model

Example: Estimating overdispersion

```
# fit the model
m1 <- glm(art ~ ., data = articles,
          family = poisson)

# get Pearson residuals
pearson_resids <- resid(m1, "pearson")

# estimate dispersion parameter
phihat <- sum(pearson_resids^2)/(915 - 6)
phihat
```

```
## [1] 1.828984
```

> What problems do you think it causes to assume the mean and variance are the same, when they are not?

Exploring effects of overdispersion

Overdispersion can affect our ability to perform inference. For example, if we create a confidence interval using a Poisson regression model, but the variance is actually bigger than the mean, then our confidence interval will be too narrow.

> How could we perform a simulation study to assess the impact of overdispersion on the coverage of Poisson regression confidence intervals? Discuss with a neighbor, then we will discuss as a group.

Class activity

https://sta214-s23.github.io/class_activities/ca_lecture_21.html

Class activity

Confidence interval coverage when underlying data is truly Poisson:

```
n <- 1000
nsim <- 500
contains_beta <- rep(0, nsim)
for(i in 1:nsim){
  x <- rnorm(n, sd = 0.5)
  y1 <- rpois(n, lambda = exp(x))

  m1 <- glm(y1 ~ x, family = poisson)

  upper <- summary(m1)$coefficients[2,1] +
      1.96*summary(m1)$coefficients[2,2]
  lower <- summary(m1)$coefficients[2,1] -
      1.96*summary(m1)$coefficients[2,2]

  contains_beta[i] <- upper > 1 && lower < 1
}

mean(contains_beta)
```

```
## [1] 0.958
```

Class activity

Coverage when there is overdispersion:

```
n <- 1000
nsim <- 500
contains_beta <- rep(0, nsim)
for(i in 1:nsim){
  x <- rnorm(n, sd = 0.5)
  y2 <- rnbinom(n, size=0.5, mu=exp(x))

  m2 <- glm(y2 ~ x, family = poisson)

  upper <- summary(m2)$coefficients[2,1] +
      1.96*summary(m2)$coefficients[2,2]
  lower <- summary(m2)$coefficients[2,1] -
      1.96*summary(m2)$coefficients[2,2]

  contains_beta[i] <- upper > 1 && lower < 1
}

mean(contains_beta)
```

```
## [1] 0.666
```

Class activity

> How does coverage change as I decrease the amount of overdispersion in the data?

Class activity

```
n <- 1000
nsim <- 500
contains_beta <- rep(0, nsim)
for(i in 1:nsim){
  x <- rnorm(n, sd = 0.5)
  y2 <- rnbinom(n, size=10, mu=exp(x))

  m2 <- glm(y2 ~ x, family = poisson)

  upper <- summary(m2)$coefficients[2,1] +
      1.96*summary(m2)$coefficients[2,2]
  lower <- summary(m2)$coefficients[2,1] -
      1.96*summary(m2)$coefficients[2,2]

  contains_beta[i] <- upper > 1 && lower < 1
}

mean(contains_beta)
```

```
## [1] 0.926
```

Handling overdispersion

Overdispersion is a problem because our standard errors (for confidence intervals and hypothesis tests) are too low.

> If we think there is overdispersion, what should we do?

Adjusting the standard error

- In our data, $\widehat{\phi} = 1.83$
- This means our variance is 1.83 times bigger than it should be
- So our standard error is $\sqrt{1.83} = 1.35$ times bigger than it should be

Adjusting the standard error in R

```
m2 <- glm(art ~ ., data = articles,
          family = quasipoisson)
```

```
...
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.304617   0.139273   2.187 0.028983 *
## femWomen    -0.224594   0.073860  -3.041 0.002427 **
## marMarried   0.155243   0.083003   1.870 0.061759 .
## kid5        -0.184883   0.054268  -3.407 0.000686 ***
## phd          0.012823   0.035700   0.359 0.719544
## ment         0.025543   0.002713   9.415  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 1.829006)
...
```

- Allowing ϕ to be different from 1 means we are using a *quasi-likelihood* (in this case, a *quasi-Poisson*)

Calculating a confidence interval

```
...
## (Intercept)  0.304617   0.139273   2.187 0.028983 *
## femWomen    -0.224594   0.073860  -3.041 0.002427 **
## marMarried   0.155243   0.083003   1.870 0.061759 .
## kid5        -0.184883   0.054268  -3.407 0.000686 ***
## phd          0.012823   0.035700   0.359 0.719544
## ment         0.025543   0.002713   9.415  < 2e-16 ***
...
```

New confidence interval for β_4:

$$0.0128 \pm t^*_{n-p} \cdot 0.0357$$

```
qt(0.025, df=909, lower.tail=F)
```

```
## [1] 1.962577
```

$$0.0128 \pm 1.96 \cdot 0.0357 = (-0.0572,\ 0.0828)$$

Adjusting the standard error in R

Poisson:

```
...
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.304617   0.102981   2.958   0.0031 **
## femWomen    -0.224594   0.054613  -4.112 3.92e-05 ***
## marMarried   0.155243   0.061374   2.529   0.0114 *
## kid5        -0.184883   0.040127  -4.607 4.08e-06 ***
...
```

Quasi-Poisson:

```
...
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.304617   0.139273   2.187 0.028983 *
## femWomen    -0.224594   0.073860  -3.041 0.002427 **
## marMarried   0.155243   0.083003   1.870 0.061759 .
## kid5        -0.184883   0.054268  -3.407 0.000686 ***
...
```

Back to simulations

```
n <- 1000
nsim <- 500
contains_beta <- rep(0, nsim)
for(i in 1:nsim){
  x <- rnorm(n, sd = 0.5)
  y2 <- rnbinom(n, size=0.5, mu=exp(x))

  m2 <- glm(y2 ~ x, family = poisson)

  upper <- summary(m2)$coefficients[2,1] +
      1.96*summary(m2)$coefficients[2,2]
  lower <- summary(m2)$coefficients[2,1] -
      1.96*summary(m2)$coefficients[2,2]

  contains_beta[i] <- upper > 1 && lower < 1
}

mean(contains_beta)
```

```
## [1] 0.63
```

Adjusting for overdispersion

```
n <- 1000
nsim <- 500
contains_beta <- rep(0, nsim)
for(i in 1:nsim){
  x <- rnorm(n, sd = 0.5)
  y2 <- rnbinom(n, size=0.5, mu=exp(x))

  m2 <- glm(y2 ~ x, family = quasipoisson)

  upper <- summary(m2)$coefficients[2,1] +
      qt(0.025, n-2, lower.tail = F)*summary(m2)$coefficients[2,2]
  lower <- summary(m2)$coefficients[2,1] -
      qt(0.025, n-2, lower.tail = F)*summary(m2)$coefficients[2,2]

  contains_beta[i] <- upper > 1 && lower < 1
}

mean(contains_beta)
```

```
## [1] 0.906
```