Count data and the Poisson distribution

Frank Edwards

Count data

- Counts are cumulative totals of the number of incidences of some event, generally across time or place

Count data

- Counts are cumulative totals of the number of incidences of some event, generally across time or place
- Counts are positive integers $\in [0, \infty]$

Count data

- Counts are cumulative totals of the number of incidences of some event, generally across time or place
- Counts are positive integers $\in [0, \infty]$

Counts as extensions of binary data

- Counts can be thought of as repeated binary trials
- $\sum y_i$ where y is equal to 1 or 0 provides a count
- Generally, we could treat `sum(y==1) + sum(y==0)` or `nrow(y)` as the exposure, or denominator for a rate. Why?

An example of count data

```
## data from https://github.com/adror1/nwslR or
## devtools::install_github('adror1/nwslR')
library(nwslR)
data("player")
data("fieldplayer_overall_season_stats")
head(player, n = 2)
```

```
## # A tibble: 2 x 5
##   person_id player        nation pos   name_other
##       <dbl> <chr>         <chr>  <chr> <chr>
## 1       342 Marisa Abegg  USA    DF    <NA>
## 2       117 Danesha Adams USA    FW,MF <NA>
```

```
head(fieldplayer_overall_season_stats, n = 2)
```

```
## # A tibble: 2 x 14
##   person_id season nation pos   team_id    mp starts   min   gls   ast    pk
##       <int>  <dbl> <chr>  <chr> <chr>   <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
## 1       342   2013 USA    DF    WAS         5      4    NA     0     0     0
## 2       117   2013 USA    FW,MF NJ         20     20    NA     3     3     1
## # i 3 more variables: p_katt <dbl>, crd_y <dbl>, crd_r <dbl>
```

```
# check the help files with ?(fieldplayer_overall_season_stats) for codebook
```

make a joined table with players names

```r
### attaching names
dat <- fieldplayer_overall_season_stats %>%
    left_join(player) %>%
    filter(!(is.na(min)))

### check to ensure that the dimensions are what we want
nrow(dat) == nrow(fieldplayer_overall_season_stats)
```

```
## [1] FALSE
```

Approaches to modeling count data

The Poisson model

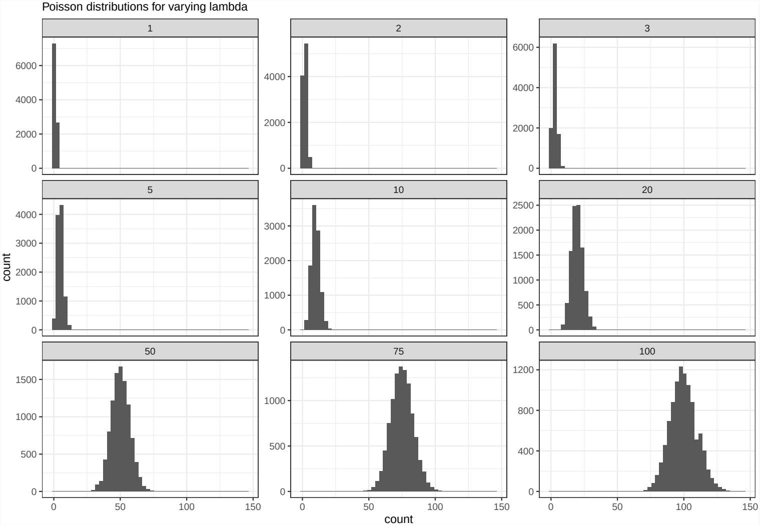
Where y is a non-negative integer (count)

$$y \sim Poisson(\lambda)$$

$$E(y) = \bar{y} = \lambda$$

$$Var(y) = \lambda$$

Shape of the Poisson distribution for varying Lambda parameters



Let’s look at each Poisson variable

```
pois_demo %>%
    group_by(lambda) %>%
    summarise(mean = mean(count), variance = var(count))
```

```
## # A tibble: 9 x 3
##   lambda   mean variance
##    <dbl>  <dbl>    <dbl>
## 1      1   1.01     1.01
## 2      2   2.00     1.99
## 3      3   2.97     2.94
## 4      5   5.01     5.00
## 5     10  10.0     10.0
## 6     20  20.0     20.3
## 7     50  49.9     50.0
## 8     75  75.0     72.8
## 9    100 100.     101.
```

Poisson models as a GLM

For a count variable y, we can specify a Poisson GLM with a log link function

$$y \sim Poisson(\lambda)$$

$$\lambda = e^{\beta_0 + \beta_1 x_1 \cdots \beta_n x_n}$$

Poisson models as a GLM

For a count variable y, we can specify a Poisson GLM with a log link function

$$y \sim Poisson(\lambda)$$

$$\lambda = e^{\beta_0 + \beta_1 x_1 \cdots \beta_n x_n}$$

What is $\log(\lambda)$ equal to?

Poisson models as a GLM

$$E(y|x) = e^{\lambda}$$

$$log(E(y|x)) = \lambda = X\beta$$

Poisson models as a GLM

$$E(y|x) = e^{\lambda}$$

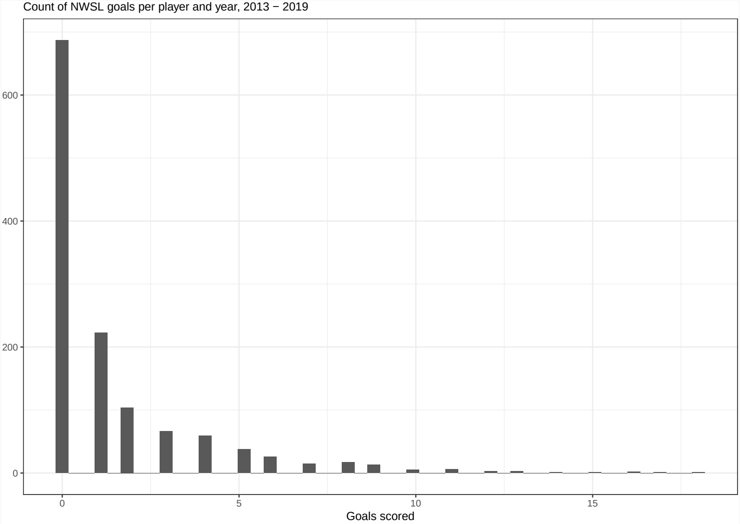
$$log(E(y|x)) = \lambda = X\beta$$

if a GLM is defined as $g(\mu) = X\beta$ with link function g, what is the link function for the Poisson GLM?

Model NWSL data using a Poisson GLM

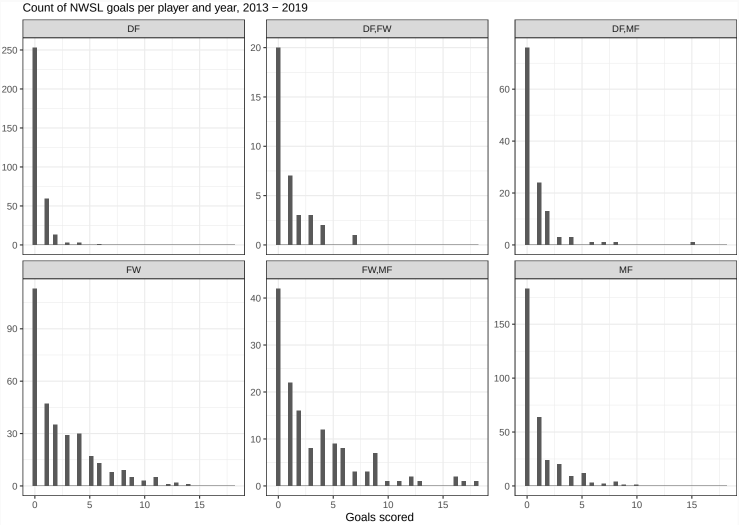
Goal scoring

```
ggplot(dat, aes(x = gls)) + geom_histogram(bins = 50) + labs(x = "Goals scored",
    y = "", subtitle = "Count of NWSL goals per player and year, 2013 - 2019")
```



Goal scoring

```r
ggplot(dat, aes(x = gls)) + geom_histogram(bins = 50) + facet_wrap(~pos, scales = "free_y") +
    labs(x = "Goals scored", y = "", subtitle = "Count of NWSL goals per player and year, 2013 - 2019")
```



Modeling goals

```r
goals_0 <- glm(gls ~ pos, data = dat, family = "poisson")

broom::tidy(goals_0)
```

```
## # A tibble: 6 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)   -1.09     0.0945    -11.5  1.32e- 30
## 2 posDF,FW       1.11     0.190       5.88 4.23e-  9
## 3 posDF,MF       0.947    0.135       7.01 2.43e- 12
## 4 posFW          2.01     0.101      19.9  6.96e- 88
## 5 posFW,MF       2.27     0.106      21.5  3.18e-102
## 6 posMF          1.17     0.109      10.8  4.33e- 27
```

So how many goals does our model expect for each position?

We could just do the math: $\lambda_i = E(y_i|X) = e^{\beta_0 + \beta_1 x_1 \ldots \beta_n x_n}$

```
exp(coef(goals_0))
```

```
## (Intercept)    posDF,FW    posDF,MF       posFW    posFW,MF       posMF
##   0.3373494   3.0466270   2.5786876   7.4293576   9.6392600   3.2212517
```

And because $e^{a+b} = e^a \times e^b$

Expected goals for a forward under model 0 are $e^{\beta_0} \times e^{\beta_3}$

```
# intercept is in row 1, b3 is in row 4
exp(coef(goals_0)[4]) * exp(coef(goals_0)[1])
```

```
##    posFW
## 2.506289
```

So how many goals does our model expect for each position?

Or we could have R handle everything using simulation

```
sim_dat <- data.frame(pos = unique(dat$pos))
sim_dat <- sim_dat %>%
    mutate(e_gls = predict(goals_0, newdata = sim_dat, type = "response"))

sim_dat
```

```
##     pos     e_gls
## 1    FW 2.5062893
## 2 DF,MF 0.8699187
## 3    DF 0.3373494
## 4    MF 1.0866873
## 5 DF,FW 1.0277778
## 6 FW,MF 3.2517986
```

Regression generates conditional means

Not coincidentally:

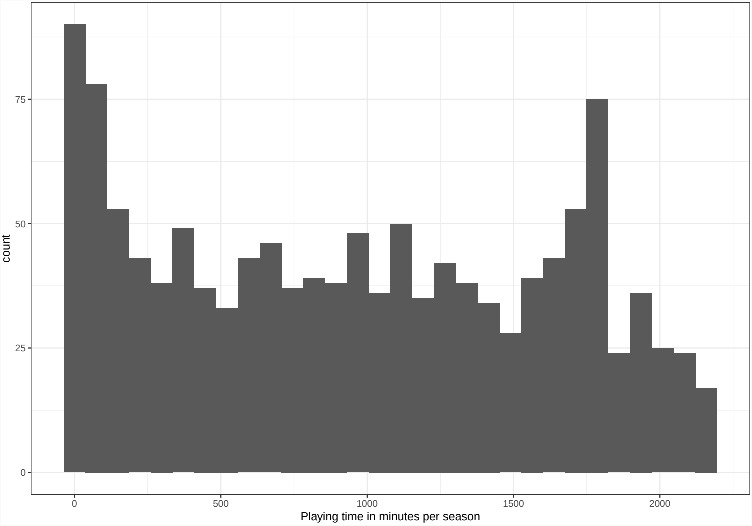
```
dat %>%
    group_by(pos) %>%
    summarize(gls = mean(gls))
```

```
## # A tibble: 6 x 2
##   pos     gls
##   <chr> <dbl>
## 1 DF    0.337
## 2 DF,FW 1.03
## 3 DF,MF 0.870
## 4 FW    2.51
## 5 FW,MF 3.25
## 6 MF    1.09
```

Fitting a more complex model

Let's look at playing time as a predictor

```
ggplot(dat, aes(x = min)) + geom_histogram() + labs(x = "Playing time in minutes per season")
```



How does playing time impact scoring?

Theory time

How does playing time impact scoring?

As an opportunity structure - more time = more chances

Theory time

How does playing time impact scoring?

As an opportunity structure - more time = more chances

Is playing time likely to have the same effect on goal scoring for each position?

How does playing time impact scoring?

As an opportunity structure - more time = more chances

Is playing time likely to have the same effect on goal scoring for each position?

Let's evaluate two models:

$$m1 : E(goals|position, minutes) = e^{\beta_0+\beta_1 position+\beta_2 minutes}$$

$$m2 : E(goals|position, minutes) = e^{\beta_0+\beta_1 position\times\beta_2 minutes}$$

Estimate the models

```
goals_1 <- glm(gls ~ pos + min, data = dat, family = "poisson")

goals_2 <- glm(gls ~ pos * min, data = dat, family = "poisson")
```

Check our fits

```r
broom::tidy(goals_1)
```

```
## # A tibble: 7 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept) -2.76    0.113         -24.3  1.31e-130
## 2 posDF,FW     1.28    0.190           6.74 1.55e- 11
## 3 posDF,MF     0.834   0.135           6.17 6.83e- 10
## 4 posFW        2.30    0.101          22.7  3.86e-114
## 5 posFW,MF     2.42    0.106          22.9  4.22e-116
## 6 posMF        1.28    0.109          11.8  3.67e- 32
## 7 min          0.00128 0.0000413      31.1  4.94e-212
```

Check our fits

```r
broom::tidy(goals_2)
```

```
## # A tibble: 12 x 5
##    term          estimate std.error statistic  p.value
##    <chr>            <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)  -2.26      0.253       -8.92  4.62e-19
##  2 posDF,FW      1.45      0.421        3.44  5.80e- 4
##  3 posDF,MF      0.631     0.397        1.59  1.12e- 1
##  4 posFW         1.81      0.267        6.77  1.30e-11
##  5 posFW,MF      1.58      0.290        5.45  5.10e- 8
##  6 posMF         0.814     0.293        2.78  5.45e- 3
##  7 min           0.000941  0.000165     5.69  1.26e- 8
##  8 posDF,FW:min -0.000149  0.000286    -0.520 6.03e- 1
##  9 posDF,MF:min  0.000149  0.000253     0.588 5.56e- 1
## 10 posFW:min     0.000333  0.000176     1.89  5.87e- 2
## 11 posFW,MF:min  0.000579  0.000189     3.06  2.18e- 3
## 12 posMF:min     0.000318  0.000190     1.67  9.50e- 2
```

Compare goodness of fit with AIC

```r
AIC(goals_0, goals_1, goals_2)
```

```
##         df      AIC
## goals_0  6 4802.068
## goals_1  7 3701.602
## goals_2 12 3695.366
```

```r
# AIC is an adjusted measure for the log-likelihood of the model
logLik(goals_0)
```

```
## 'log Lik.' -2395.034 (df=6)
```

```r
logLik(goals_1)
```

```
## 'log Lik.' -1843.801 (df=7)
```

```r
logLik(goals_2)
```

```
## 'log Lik.' -1835.683 (df=12)
```

Let’s look at model expectations with simulation

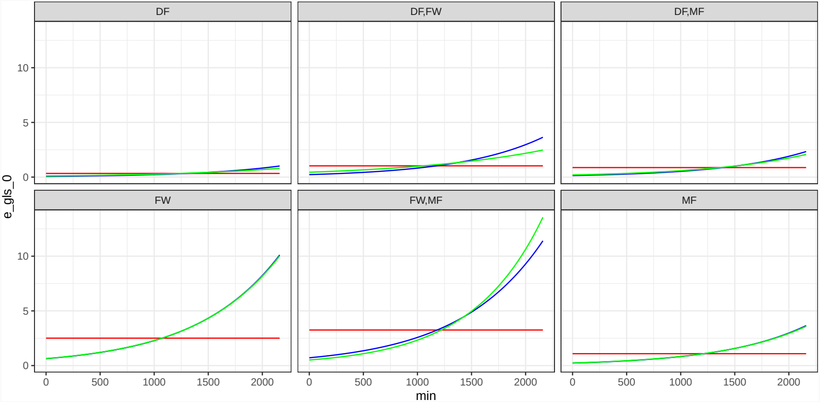
```r
pos <- unique(dat$pos)
min <- 0:max(dat$min)
sim_dat <- expand_grid(pos, min)
# whoa that's a big object oh well!
sim_dat <- sim_dat %>%
    mutate(e_gls_0 = predict(goals_0, newdata = sim_dat, type = "response"), e_gls_1 = predict(goals_1,
        newdata = sim_dat, type = "response"), e_gls_2 = predict(goals_2, newdata = sim_dat,
        type = "response"))

sim_dat
```

```
## # A tibble: 12,966 x 5
##    pos     min e_gls_0 e_gls_1 e_gls_2
##    <chr> <int>   <dbl>   <dbl>   <dbl>
##  1 FW        0    2.51   0.632   0.640
##  2 FW        1    2.51   0.633   0.641
##  3 FW        2    2.51   0.634   0.641
##  4 FW        3    2.51   0.634   0.642
##  5 FW        4    2.51   0.635   0.643
##  6 FW        5    2.51   0.636   0.644
##  7 FW        6    2.51   0.637   0.645
##  8 FW        7    2.51   0.638   0.646
##  9 FW        8    2.51   0.639   0.646
## 10 FW        9    2.51   0.639   0.647
## # i 12,956 more rows
```

Now to visualize our model predictions

```
ggplot(sim_dat, aes(x = min)) + geom_line(aes(y = e_gls_0), color = "red") + geom_line(aes(y = e_gls_1),
    color = "blue") + geom_line(aes(y = e_gls_2), color = "green") + facet_wrap(~pos)
```



Advantages of the Poisson distribution for regression

1. Constrained to non-negative integers
2. Variance scales with the expectation of y
3. Relatively simple to interpret

Homework

1. Visualize the distribution of goals across players for the 2019 season (your choice on geom)
2. Define a linear predictor for goals made during a season, where the players' position is the only predictor.
3. Estimate this model with a Normal likelihood (OLS)
4. Estimate this model with a Poisson likelihood (family = "poisson")
5. Generate predictions for each position for both models
6. Compare the predictions. Which model do you prefer? Why?