Poisson regression

aqf

February 2022

Example of a Poisson process

Suppose the probability that a drug produces a certain side effect is p = 0.1% and n = 1,000 patients in a clinical trial receive the drug. What is the probability 0 people experience the side effect?

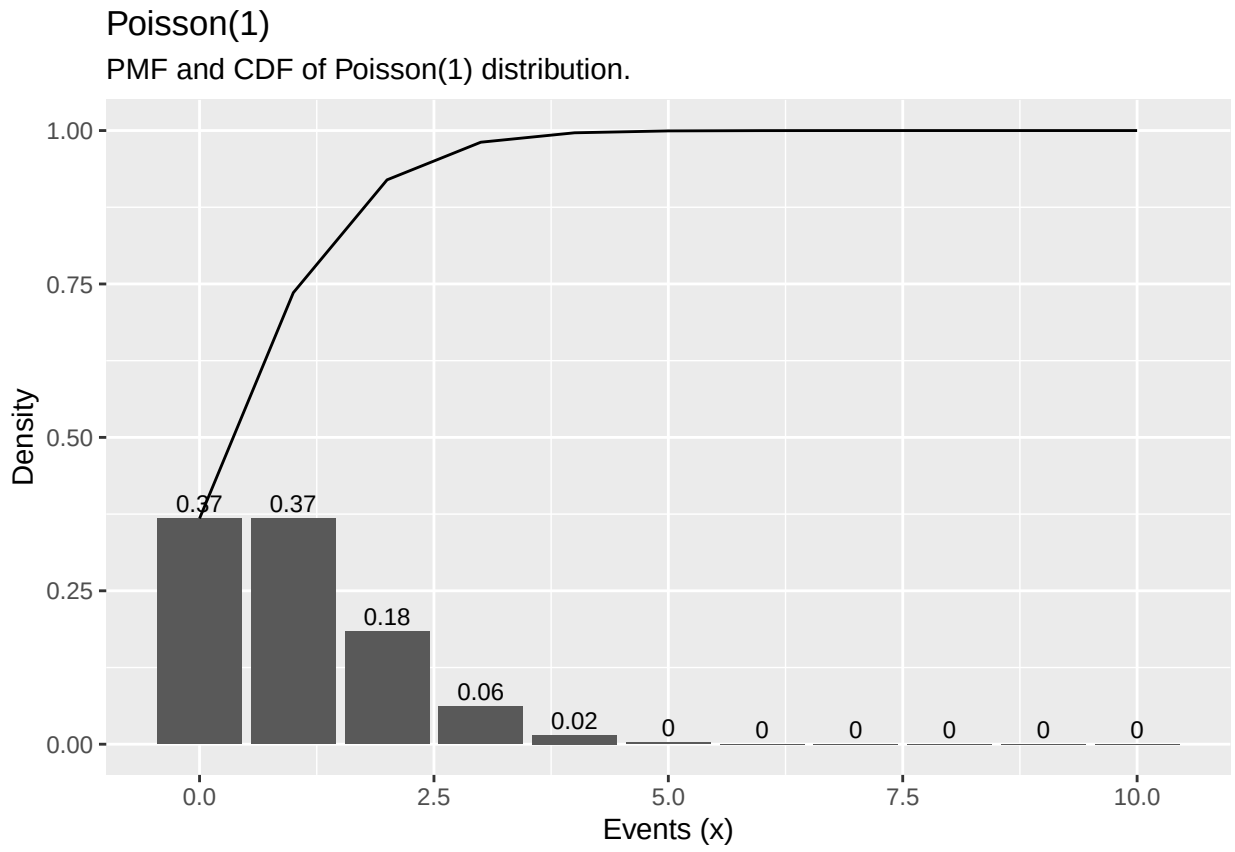
```
# The expected value is np
1000 * .001
```

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

```
# The probability of measuring 0 when the expected value is 1
dpois(x = 0, lambda = 1000 * .001)
```

```
## [1] 0.3678794
```

```
library(ggplot2)
library(dplyr)
options(scipen = 999, digits = 2) # sig digits

x <- 0:10
density <- dpois(x = x, lambda = 1000 * .001)
prob <- ppois(q = x, lambda = 1000 * .001, lower.tail = TRUE)
df <- data.frame(x, density, prob)

ggplot(df, aes(x = x, y = density)) +
  geom_col() +
  geom_text(
    aes(label = round(density,2), y = density + 0.01),
    position = position_dodge(0.9),
    size = 3,
    vjust = 0
  ) +
  labs(title = "Poisson(1)",
       subtitle = "PMF and CDF of Poisson(1) distribution.",
       x = "Events (x)",
       y = "Density") +
  geom_line(data = df, aes(x = x, y = prob))
```



The distribution gives high probability when the number of events is zero or one and decreases rapidly as the number of events grows.

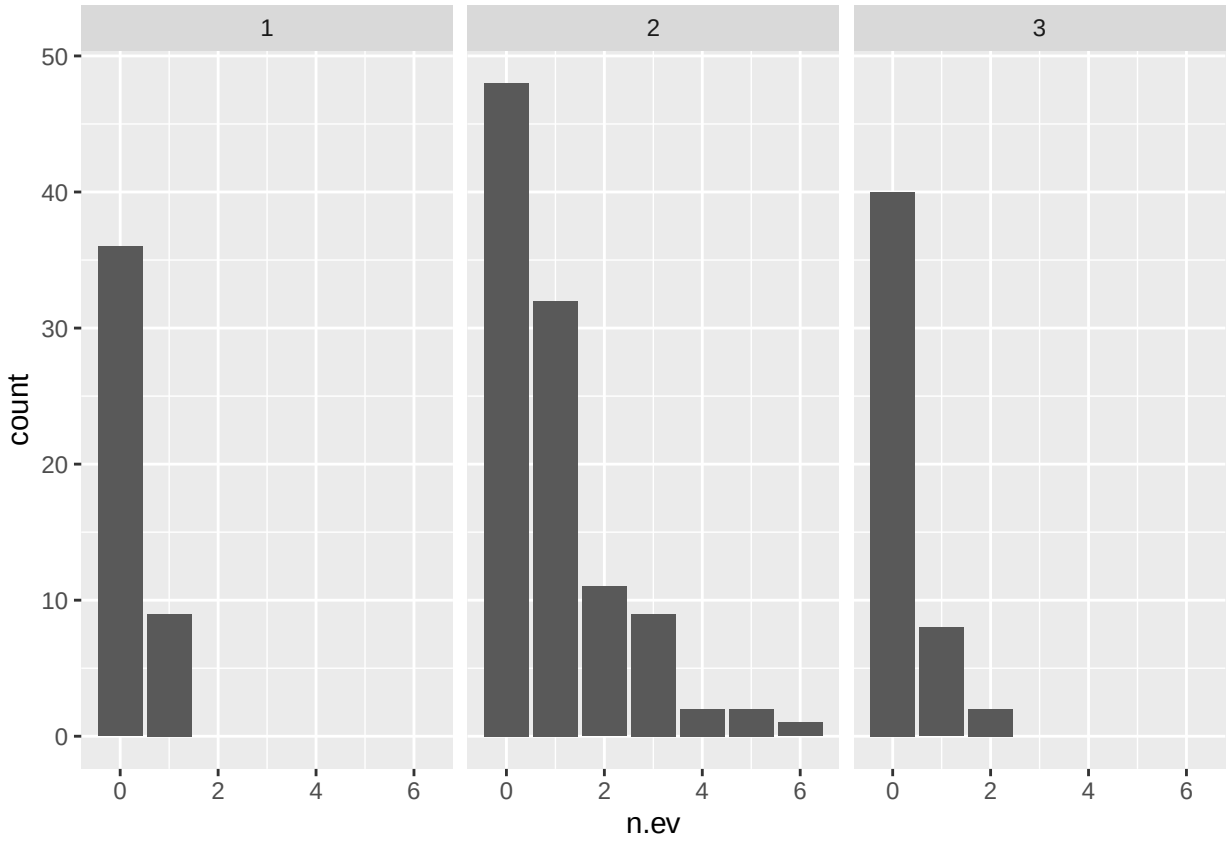
Poisson regression

Poisson Regression Assumptions

- Poisson response variable
- Independence
- The mean of a poisson random variable must be equal to its variance
- Linearity the log of the mean rate, $\log(\lambda)$, must be a linear function of the explanatory variable

After fitting a Poisson regression model, we need to check for overdispersion.

```
# explore whether the response is a Poisson process
dt %>%
  ggplot(aes(n.ev)) +
  geom_bar() +
  facet_grid(cols = vars(ind))
```



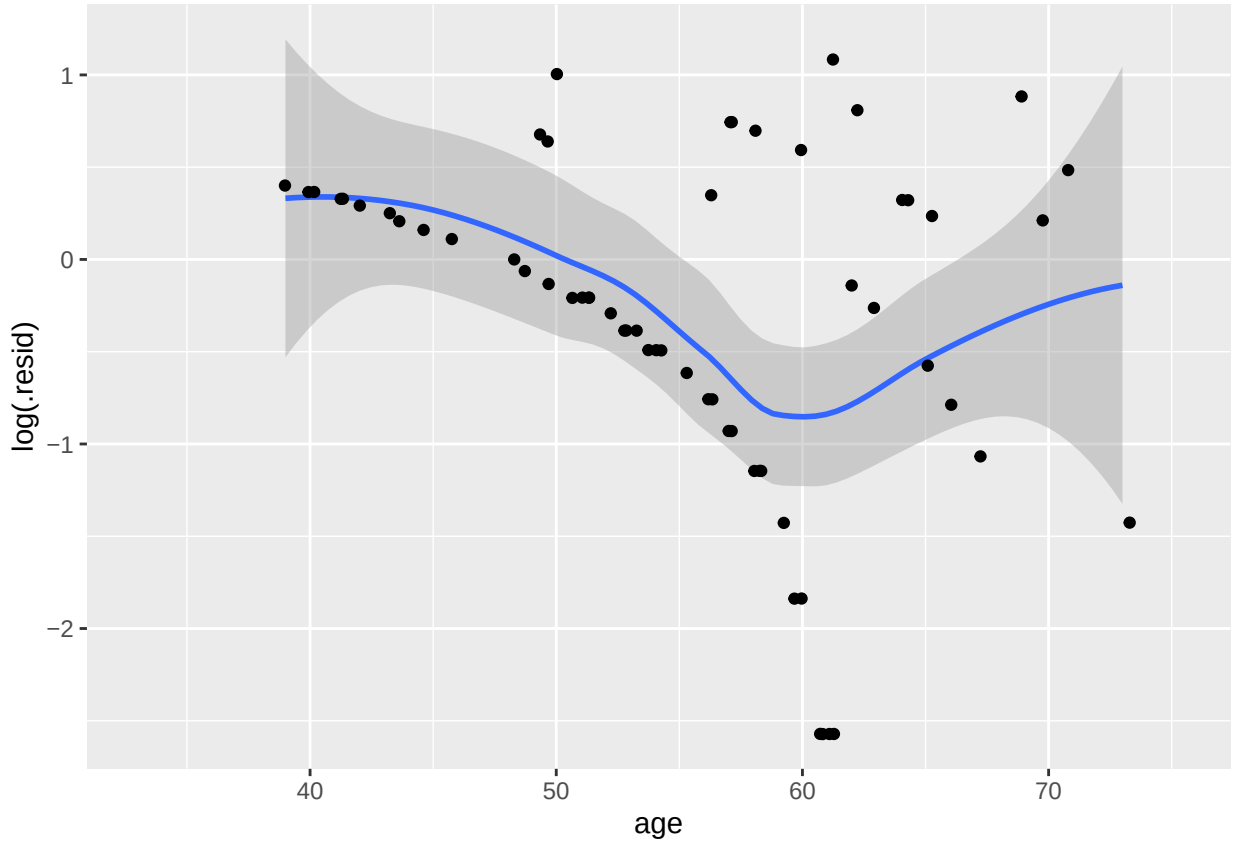

```
# inspect means and variances
dt %>%
  group_by(ind) %>%
  summarise(mean = mean(n.ev, na.rm = T), variance = var(n.ev))
```

```
## # A tibble: 3 x 3
##     ind  mean variance
##   <int> <dbl>    <dbl>
## 1     1  0.2     0.164
## 2     2  1       1.63
## 3     3  0.24    0.268
```

```
# linearity of the log(mean)
mdl1 <- glm(n.ev ~ age, family = "poisson", data = dt)
fortify(mdl1) %>%
  ggplot(aes(age, log(.resid))) + geom_smooth() + geom_jitter()
```

```
## Warning in log(.resid): NaNs produced

## Warning in log(.resid): NaNs produced

## Warning in log(.resid): NaNs produced

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## Warning: Removed 139 rows containing non-finite values (stat_smooth).

## Warning: Removed 139 rows containing missing values (geom_point).
```




```
# obtain model summary
summary(mdl1)
```

```
##
## Call:
## glm(formula = n.ev ~ age, family = "poisson", data = dt)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.185  -0.907  -0.600   0.325   2.953
##
## Coefficients:
##             Estimate Std. Error z value            Pr(>|z|)
## (Intercept) -5.33353    0.59126   -9.02 <0.0000000000000002 ***
## age          0.08617    0.00968    8.90 <0.0000000000000002 ***
## ---
## 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: 204.02  on 198  degrees of freedom
## AIC: 384.1
##
## Number of Fisher Scoring iterations: 6
```

```
# Dispersion parameter rule of thumb less than 1
sum(residuals(mdl1, type ="pearson")^2)/mdl1$df.residual
```

```
## [1] 1.2
```

```
# Make a table to calculate 95% CIs)
cov.mdl1 <- vcovHC(mdl1, type = "HC0") # covariance matrix of mdl1
std.err <- sqrt(diag(cov.mdl1))      # standard error calculation
r.est <- cbind(Estimate = coef(mdl1), "Robust SE" = std.err,
               "Pr(>|z|)" = 2 * pnorm(abs(coef(mdl1)/std.err), lower.tail = FALSE),
               LL = coef(mdl1) - 1.96 * std.err,
               UL = coef(mdl1) + 1.96 * std.err)
r.est
```

```
##             Estimate Robust SE               Pr(>|z|)     LL    UL
## (Intercept)   -5.334    0.5933 0.0000000000000000025 -6.496 -4.17
## age            0.086    0.0098 0.0000000000000000183  0.067  0.11
```

```
# alternatively
gtsummary::tbl_regression(mdl1)
```

```
## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in code chunk header.
```

Characteristic	**log(IRR)**	**95% CI**	**p-value**
age	0.09	0.07, 0.11	<0.001

```
# Run a chi-squared test to check goodness-of-fit
pchisq(mdl1$deviance, mdl1$df.residual, lower.tail = FALSE)
```

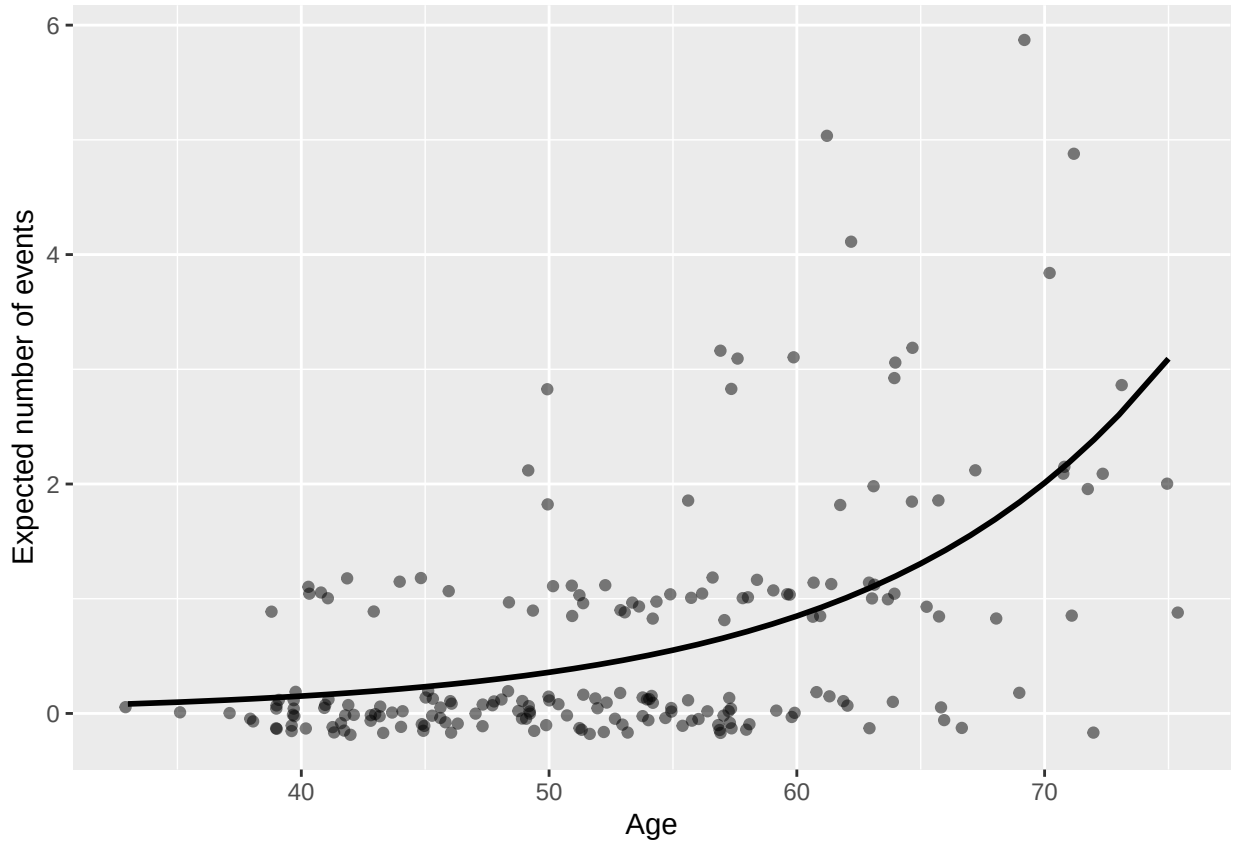
```
## [1] 0.37
```

```
# Suppose we want to drop one independent variable and compare model fit
mdl2 <- update(mdl1, . ~ . -age)
anova(mdl2, mdl1, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: n.ev ~ 1
## Model 2: n.ev ~ age
##   Resid. Df Resid. Dev Df Deviance            Pr(>Chi)
## 1       199        288
```

```
## 2       198        204  1     83.7 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Calculate expected counts
dt$phat <- predict(mdl1, type = "response")
dt <- dt[with(dt, order(age)), ]
 # and represent them graphically
ggplot(dt, aes(x = age, y = phat)) +
  geom_point(aes(y = n.ev), alpha = .5, position = position_jitter(h = .2)) +
  geom_line(size = 1) +
  labs(x = "Age", y = "Expected number of events")
```



Quasi-poisson

When overdispersion for a variable is present use quasi-Poisson

Overdispersion suggests that there is more variation in the response than the model implies. Under a Poisson model, we would expect the means and variances of the response to be about the same in various groups. Without adjusting for overdispersion, we use incorrect, artificially small standard errors leading to artificially small p-values for model coefficients. We may also end up with artificially complex models.

We can estimate a dispersion parameter, ϕ, by dividing the model deviance by its corresponding degrees of freedom; i.e., $\hat{\phi} = \frac{\sum(\text{Pearson residuals})^2}{n-p}$ where p is the number of model parameters.

```
mdllos <- glm(LOS ~ Age + Death + Organisation, data = LOS_model, family = "poisson")
summary(mdllos)
```

```
##
## Call:
## glm(formula = LOS ~ Age + Death + Organisation, family = "poisson",
##     data = LOS_model)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -3.507  -1.046  -0.264   0.734   6.075
##
## Coefficients:
##                Estimate Std. Error z value             Pr(>|z|)
## (Intercept)      0.8019     0.0661   12.13 < 0.0000000000000002 ***
## Age              0.0133     0.0010   13.26 < 0.0000000000000002 ***
## Death            0.2352     0.0625    3.76              0.00017 ***
## Organisation.L  -0.0290     0.0845   -0.34              0.73147
## Organisation.Q  -0.2229     0.0830   -2.69              0.00724 **
## Organisation.C   0.0283     0.0835    0.34              0.73481
## Organisation^4   0.0913     0.0831    1.10              0.27153
## Organisation^5  -0.1774     0.0835   -2.12              0.03364 *
## Organisation^6  -0.0562     0.0828   -0.68              0.49719
## Organisation^7  -0.0215     0.0821   -0.26              0.79325
## Organisation^8   0.1056     0.0818    1.29              0.19679
## Organisation^9  -0.1354     0.0800   -1.69              0.09058 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 752.23  on 299  degrees of freedom
## Residual deviance: 522.05  on 288  degrees of freedom
## AIC: 1507
##
## Number of Fisher Scoring iterations: 5
```

```
# Rule of thumb
sum(residuals(mdllos, type ="pearson")^2)/mdllos$df.residual
```

```
## [1] 2
```

```
# Alternatively, with {glmmTMB}
#altmd <- glmmTMB::glmmTMB(LOS ~ Age + Death, data = LOS_model, family = poisson)
#glmmTMB::fitTMB(altmd)
#glmmTMB::sigma(fitTMB(altmd))
```

In the absence of overdispersion, we expect the dispersion parameter estimate to be 1.0. The estimated dispersion parameter here is much larger than 1.0 (2.019) indicating overdispersion (extra variance) that should be accounted for. The larger estimated standard errors in the quasi-Poisson model reflect the adjustment.

```
quasiLOS <- glm(LOS ~ Age + Death + Organisation, data = LOS_model, family = "quasipoisson")
summary(quasiLOS)
```

```
##
## Call:
## glm(formula = LOS ~ Age + Death + Organisation, family = "quasipoisson",
##     data = LOS_model)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -3.507  -1.046  -0.264   0.734   6.075
##
## Coefficients:
##                Estimate Std. Error t value             Pr(>|t|)
## (Intercept)     0.80188    0.09393    8.54  0.00000000000000081 ***
## Age             0.01330    0.00143    9.33 < 0.0000000000000002 ***
## Death           0.23517    0.08886    2.65               0.0086 **
## Organisation.L -0.02900    0.12010   -0.24               0.8094
## Organisation.Q -0.22286    0.11792   -1.89               0.0598 .
## Organisation.C  0.02830    0.11874    0.24               0.8118
## Organisation^4  0.09134    0.11805    0.77               0.4397
## Organisation^5 -0.17741    0.11868   -1.49               0.1361
## Organisation^6 -0.05622    0.11767   -0.48               0.6332
## Organisation^7 -0.02151    0.11661   -0.18               0.8538
## Organisation^8  0.10559    0.11625    0.91               0.3645
## Organisation^9 -0.13539    0.11369   -1.19               0.2347
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 2)
##
##     Null deviance: 752.23  on 299  degrees of freedom
## Residual deviance: 522.05  on 288  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

Negative binomial regression

An exlpicit likelihood alternative to quasi-poisson in the case of overdispersion which allows for an additional parameter. As a result, it yields a more flexible model.

```
library(MASS)
negbnLOS <- glm.nb(LOS ~ Age + Death + Organisation, data = LOS_model)
summary(negbnLOS)
```

```
##
## Call:
## glm.nb(formula = LOS ~ Age + Death + Organisation, data = LOS_model,
##     init.theta = 6.145006808, link = log)
##
```

```
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.685  -0.838  -0.206   0.517   4.167
##
## Coefficients:
##                Estimate Std. Error z value             Pr(>|z|)
## (Intercept)     0.77144    0.08353    9.24 < 0.0000000000000002 ***
## Age             0.01350    0.00133   10.13 < 0.0000000000000002 ***
## Death           0.33142    0.08836    3.75              0.00018 ***
## Organisation.L -0.02516    0.11386   -0.22              0.82513
## Organisation.Q -0.17413    0.11291   -1.54              0.12302
## Organisation.C  0.06337    0.11349    0.56              0.57657
## Organisation^4  0.09188    0.11294    0.81              0.41590
## Organisation^5 -0.16073    0.11340   -1.42              0.15637
## Organisation^6 -0.07064    0.11297   -0.63              0.53178
## Organisation^7 -0.02835    0.11250   -0.25              0.80101
## Organisation^8  0.10967    0.11258    0.97              0.32998
## Organisation^9 -0.12324    0.11140   -1.11              0.26859
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(6.1) family taken to be 1)
##
##     Null deviance: 421.64  on 299  degrees of freedom
## Residual deviance: 289.29  on 288  degrees of freedom
## AIC: 1442
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  6.15
##           Std. Err.:  1.13
##
##  2 x log-likelihood:  -1416.12
```

Exponentiation for Age gives 1% increase for LOS for each year added. Similarly for Death, 39% longer length of stay was observed in patients who died.

Since there are multiple levels in the Organisation variable, we should adjust for multiple comparisons using Tukey's honestly significant difference

```
## mdllos <- glm(LOS ~ Age + Death + Organisation, data = LOS_model, family = "poisson")
# multicomp::glht(model, linfct)
# linfct is a specification of a linear hypothesis
# mcp(var = "method")
summary(glht(mdllos, linfct = mcp(Organisation = "Tukey")))
```

```
## Warning in RET$pfunction("adjusted", ...): Completion with error > abseps

## Warning in RET$pfunction("adjusted", ...): Completion with error > abseps

## Warning in RET$pfunction("adjusted", ...): Completion with error > abseps

## Warning in RET$pfunction("adjusted", ...): Completion with error > abseps
```

```
## Warning in RET$pfunction("adjusted", ...): Completion with error > abseps

## Warning in RET$pfunction("adjusted", ...): Completion with error > abseps

## Warning in RET$pfunction("adjusted", ...): Completion with error > abseps

##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: glm(formula = LOS ~ Age + Death + Organisation, family = "poisson",
##     data = LOS_model)
##
## Linear Hypotheses:
##                         Estimate Std. Error z value Pr(>|z|)
## Trust2 - Trust1 == 0  -0.10220    0.12039   -0.85    0.998
## Trust3 - Trust1 == 0   0.11477    0.11502    1.00    0.992
## Trust4 - Trust1 == 0   0.07102    0.11626    0.61    1.000
## Trust5 - Trust1 == 0   0.31601    0.11011    2.87    0.114
## Trust6 - Trust1 == 0   0.08863    0.11617    0.76    0.999
## Trust7 - Trust1 == 0   0.01647    0.11455    0.14    1.000
## Trust8 - Trust1 == 0  -0.00115    0.11712   -0.01    1.000
## Trust9 - Trust1 == 0   0.06764    0.11539    0.59    1.000
## Trust10 - Trust1 == 0 -0.08279    0.12005   -0.69    1.000
## Trust3 - Trust2 == 0   0.21697    0.12026    1.80    0.732
## Trust4 - Trust2 == 0   0.17322    0.12138    1.43    0.919
## Trust5 - Trust2 == 0   0.41821    0.11565    3.62    0.011 *
## Trust6 - Trust2 == 0   0.19083    0.12123    1.57    0.861
## Trust7 - Trust2 == 0   0.11867    0.12029    0.99    0.993
## Trust8 - Trust2 == 0   0.10105    0.12234    0.83    0.998
## Trust9 - Trust2 == 0   0.16985    0.12057    1.41    0.925
## Trust10 - Trust2 == 0  0.01941    0.12508    0.16    1.000
## Trust4 - Trust3 == 0  -0.04376    0.11591   -0.38    1.000
## Trust5 - Trust3 == 0   0.20124    0.10982    1.83    0.714
## Trust6 - Trust3 == 0  -0.02614    0.11575   -0.23    1.000
## Trust7 - Trust3 == 0  -0.09830    0.11489   -0.86    0.998
## Trust8 - Trust3 == 0  -0.11593    0.11698   -0.99    0.993
## Trust9 - Trust3 == 0  -0.04713    0.11517   -0.41    1.000
## Trust10 - Trust3 == 0 -0.19756    0.11987   -1.65    0.824
## Trust5 - Trust4 == 0   0.24500    0.11118    2.20    0.453
## Trust6 - Trust4 == 0   0.01762    0.11685    0.15    1.000
## Trust7 - Trust4 == 0  -0.05454    0.11619   -0.47    1.000
## Trust8 - Trust4 == 0  -0.07217    0.11811   -0.61    1.000
## Trust9 - Trust4 == 0  -0.00337    0.11619   -0.03    1.000
## Trust10 - Trust4 == 0 -0.15381    0.12089   -1.27    0.960
## Trust6 - Trust5 == 0  -0.22738    0.11105   -2.05    0.564
## Trust7 - Trust5 == 0  -0.29954    0.10988   -2.73    0.163
## Trust8 - Trust5 == 0  -0.31716    0.11228   -2.82    0.128
## Trust9 - Trust5 == 0  -0.24837    0.11058   -2.25    0.425
## Trust10 - Trust5 == 0 -0.39880    0.11539   -3.46    0.020 *
## Trust7 - Trust6 == 0  -0.07216    0.11612   -0.62    1.000
```

```
## Trust8 - Trust6 == 0  -0.08978    0.11794   -0.76    0.999
## Trust9 - Trust6 == 0  -0.02099    0.11597   -0.18    1.000
## Trust10 - Trust6 == 0 -0.17142    0.12070   -1.42    0.921
## Trust8 - Trust7 == 0  -0.01763    0.11704   -0.15    1.000
## Trust9 - Trust7 == 0   0.05117    0.11542    0.44    1.000
## Trust10 - Trust7 == 0 -0.09926    0.12003   -0.83    0.998
## Trust9 - Trust8 == 0   0.06880    0.11722    0.59    1.000
## Trust10 - Trust8 == 0 -0.08164    0.12188   -0.67    1.000
## Trust10 - Trust9 == 0 -0.15044    0.11991   -1.25    0.963
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

When there are statistically significant differences, interaction terms must be considered.

Zero-inflated models

In settings with excess zero counts, the Poisson model is fitted with an additional parameter α which corresponds to the proportion of zeroes.

```
library(pscl)
```

```
## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis
```

```
poisalc <- glm(death ~ agein + alcohol + drug.use, data = poissonlshtm, family = "poisson")
summary(poisalc)
```

```
##
## Call:
## glm(formula = death ~ agein + alcohol + drug.use, family = "poisson",
##     data = poissonlshtm)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.349  -0.656  -0.495  -0.359   1.939
##
## Coefficients:
##             Estimate Std. Error z value            Pr(>|z|)
## (Intercept) -6.59753    0.45089  -14.63 <0.0000000000000002 ***
## agein        0.08768    0.00842   10.41 <0.0000000000000002 ***
## alcohol      0.28228    0.10919    2.59              0.0097 **
## drug.use     0.03073    0.10188    0.30              0.7630
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
```

```
##     Null deviance: 1125.21  on 1657  degrees of freedom
## Residual deviance:  968.16  on 1654  degrees of freedom
## AIC: 1748
##
## Number of Fisher Scoring iterations: 6
```

```
zimalc <- zeroinfl(death ~ agein + alcohol + drug.use, data = poissonlshtm)
check_zeroinflation(poisalc, tolerance = 0.05)
```

```
## # Check for zero-inflation
##
##    Observed zeros: 1272
##   Predicted zeros: 1329
##             Ratio: 1.04

## Model seems ok, ratio of observed and predicted zeros is within the tolerance
##   range.
```

```
summary(zimalc)
```

```
## Warning in sqrt(diag(object$vcov)): NaNs produced

##
## Call:
## zeroinfl(formula = death ~ agein + alcohol + drug.use, data = poissonlshtm)
##
## Pearson residuals:
##    Min     1Q Median     3Q    Max
## -0.954 -0.464 -0.350 -0.254  3.853
##
## Count model coefficients (poisson with log link):
##             Estimate Std. Error z value            Pr(>|z|)
## (Intercept) -6.59753    0.45067  -14.64 <0.0000000000000002 ***
## agein        0.08768    0.00842   10.42 <0.0000000000000002 ***
## alcohol      0.28228    0.10919    2.59              0.0097 **
## drug.use     0.03073    0.10188    0.30              0.7630
##
## Zero-inflation model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    7.653        NaN     NaN      NaN
## agein         -0.527        NaN     NaN      NaN
## alcohol        0.016        NaN     NaN      NaN
## drug.use      -1.590   1808.356       0        1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 35
## Log-likelihood: -870 on 8 Df
```

```
# Vuong test
vuong(zimalc, poisalc)
```

```
## Vuong Non-Nested Hypothesis Test-Statistic:
## (test-statistic is asymptotically distributed N(0,1) under the
##  null that the models are indistinguishible)
## -------------------------------------------------------------
##                Vuong z-statistic             H_A             p-value
## Raw                        -0.35 model2 > model1                 0.4
## AIC-corrected       -1589003.83 model2 > model1 <0.0000000000000002
## BIC-corrected       -5889933.60 model2 > model1 <0.0000000000000002
```