Poisson regression

Chandan Kumar Pandey

2022-10-17

Poisson Distribution

$$P(X = k) = \lambda^k * e^{-\lambda}/k!$$
$$\lambda = e^{\beta 0 + (\beta 1 * X1)}$$

Example for poisson distribution.

1. Number of goal scores in match.
2. Number of car passing by traffic light.

basically,

a. count data
b. Rate

Poisson regression

When to use

1. Response is either is count or rate.
2. Whole number
3. The data may contain a large number of data points for just a few values, thereby making the frequency distribution quite skewed. See for example above histogram.
4. The data may reflect the occurrence of a rare event
5. It can be assumed that there is a certain rate of occurrence of events λ

Data

We will be work on data where math score and number of award is provide. We will fit simple poisson model with one variable.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
maths <- read.csv("data/competition_awards_data.csv")
mod1 <- glm(Awards~Math.Score,data = maths,family = "poisson")
summary(mod1)
```

```
##
## Call:
## glm(formula = Awards ~ Math.Score, family = "poisson", data = maths)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.89269  -0.41546  -0.33027   0.09621   1.31945
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.509355   0.486754  -11.32   <2e-16 ***
## Math.Score   0.076486   0.006069   12.60   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 287.672  on 199  degrees of freedom
## Residual deviance:  39.816  on 198  degrees of freedom
## AIC: 219.87
##
## Number of Fisher Scoring iterations: 5
```

```
library(AER)
```

```
## Loading required package: car
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

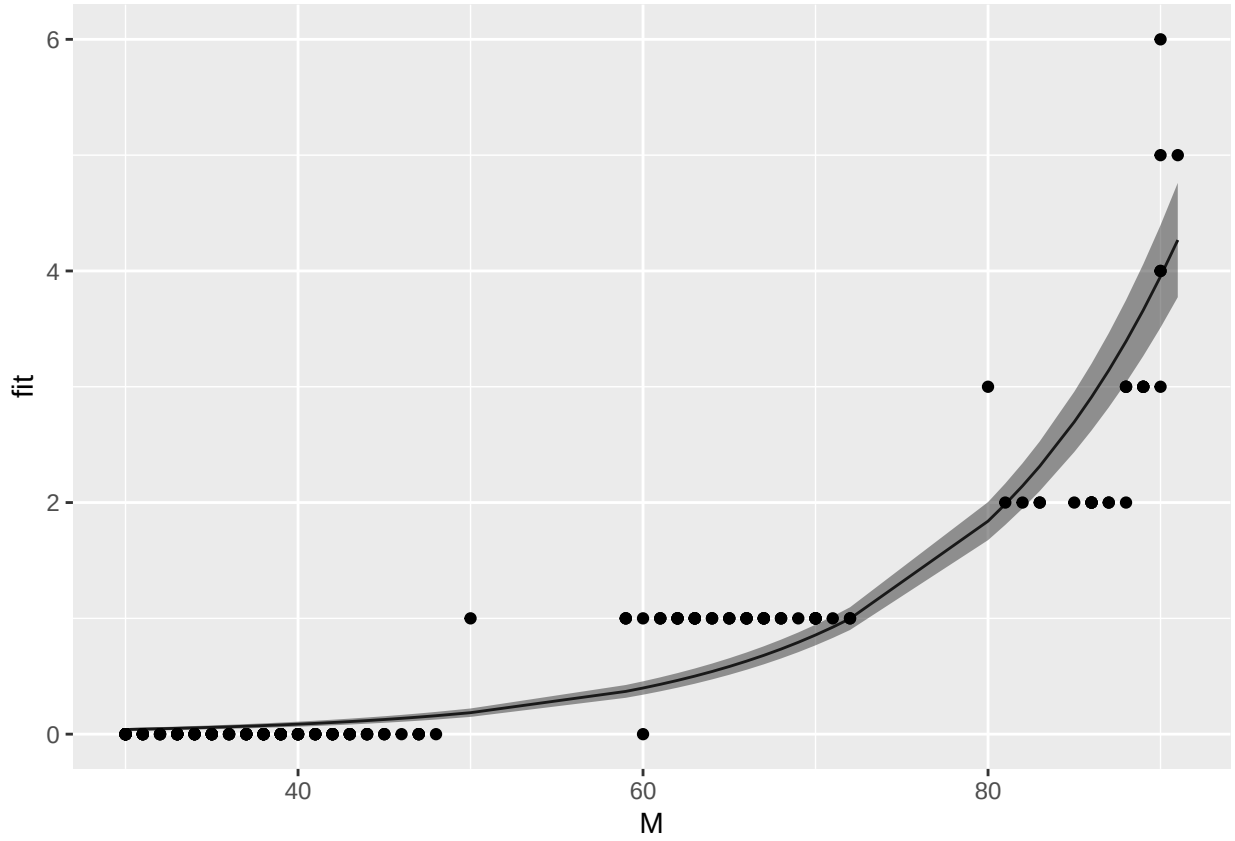
```
## Loading required package: lmtest
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

## Loading required package: survival
```

```
dispersiontest(mod1)
```

```
##
##  Overdispersion test
##
## data:  mod1
## z = -8.4316, p-value = 1
## alternative hypothesis: true dispersion is greater than 1
## sample estimates:
## dispersion
##  0.6187864
```

```
##plotting the model
predicted_value <- predict.glm(mod1,maths,type = "response",se.fit = T)
predicted_value <- predicted_value%>%as.data.frame()
df <- cbind(M=maths$Math.Score,predicted_value)
ggplot(data = df ,aes(x=M,y=fit))+geom_line()+
  geom_ribbon(aes(ymin=fit-se.fit,ymax=fit+se.fit),alpha=0.5)+
  geom_point(data = maths,aes(x=Math.Score,y=Awards))
```



In real life it is very hard to get data which follow this trend, therefore we use negative binomial distribution to model where overdisperssion is higher that 1.10.

Example

Road kill data

```
Road_kill <- read.csv("./data/road_kill.csv",header = T)
## summary()
summary(Road_kill)
```

```
##        X                ID           Class              Order
##  Min.   :    1   Min.   :    1   Length:21512       Length:21512
##  1st Qu.: 5379   1st Qu.: 5379   Class :character   Class :character
##  Median :10756   Median :10756   Mode  :character   Mode  :character
##  Mean   :10756   Mean   :10756
##  3rd Qu.:16134   3rd Qu.:16134
##  Max.   :21512   Max.   :21512
##
##     Family             Genus           Scientific_name    Common_name
##  Length:21512       Length:21512       Length:21512       Length:21512
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
```

```
##
##
##  IUCN_status             Year         Month                Day
##  Length:21512       Min.   :1988   Length:21512       Min.   : 1.00
##  Class :character   1st Qu.:2005   Class :character   1st Qu.:12.00
##  Mode  :character   Median :2012   Mode  :character   Median :18.00
##                     Mean   :2010                      Mean   :17.34
##                     3rd Qu.:2014                      3rd Qu.:23.00
##                     Max.   :2017                      Max.   :31.00
##                     NA's   :190                       NA's   :353
```

```
library(dplyr)
Road_kill_sum <- Road_kill%>%group_by(Class,Year,Month)%>%
  summarise(count=n())%>%as.data.frame()
```

```
## `summarise()` has grouped output by 'Class', 'Year'. You can override using the
## `.groups` argument.
```

```
Road_kill_sum<-na.omit(Road_kill_sum)
class(Road_kill_sum$count)
```

```
## [1] "integer"
```

```
Road_kill_sum$Month <- as.factor(Road_kill_sum$Month)
```

If the number of road kill have increase with years

```
mod1 <- glm(count~Year,data = Road_kill_sum,family = "poisson")
summary(mod1)
```

```
##
## Call:
## glm(formula = count ~ Year, family = "poisson", data = Road_kill_sum)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -8.780  -5.423  -2.715   2.565  36.220
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -39.232595   3.017337   -13.0   <2e-16 ***
## Year          0.021319   0.001501    14.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 20962  on 573  degrees of freedom
## Residual deviance: 20757  on 572  degrees of freedom
## AIC: 23532
##
## Number of Fisher Scoring iterations: 5
```

```r
## check for over dispersion
library(AER)
dispersiontest(mod1)
```

```
##
##  Overdispersion test
##
## data:  mod1
## z = 4.5832, p-value = 2.29e-06
## alternative hypothesis: true dispersion is greater than 1
## sample estimates:
## dispersion
##   48.60764
```

```r
## it is wrong but lets predict it.
value_pred <- predict(mod1,Road_kill_sum,type = "response",se.fit = T)
value_pred <- as.data.frame(value_pred)
value_pred <- cbind(Years = Road_kill_sum$Year,value_pred)
```

Since th dispersion is very high we will used negative binomial

Negetive binomial distribution

```r
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```r
mod_kill_nb <- glm.nb(count ~ Year, data = Road_kill_sum)
summary(mod_kill_nb)
```

```
##
## Call:
## glm.nb(formula = count ~ Year, data = Road_kill_sum, init.theta = 0.9550214435,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1537  -1.0709  -0.4596   0.3815   4.1040
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -35.81447   18.68701  -1.917   0.0553 .
## Year          0.01962    0.00930   2.109   0.0349 *
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.955) family taken to be 1)
##
##     Null deviance: 657.5  on 573  degrees of freedom
## Residual deviance: 652.7  on 572  degrees of freedom
## AIC: 5306.2
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.9550
##           Std. Err.:  0.0523
##
##  2 x log-likelihood:  -5300.1560
```

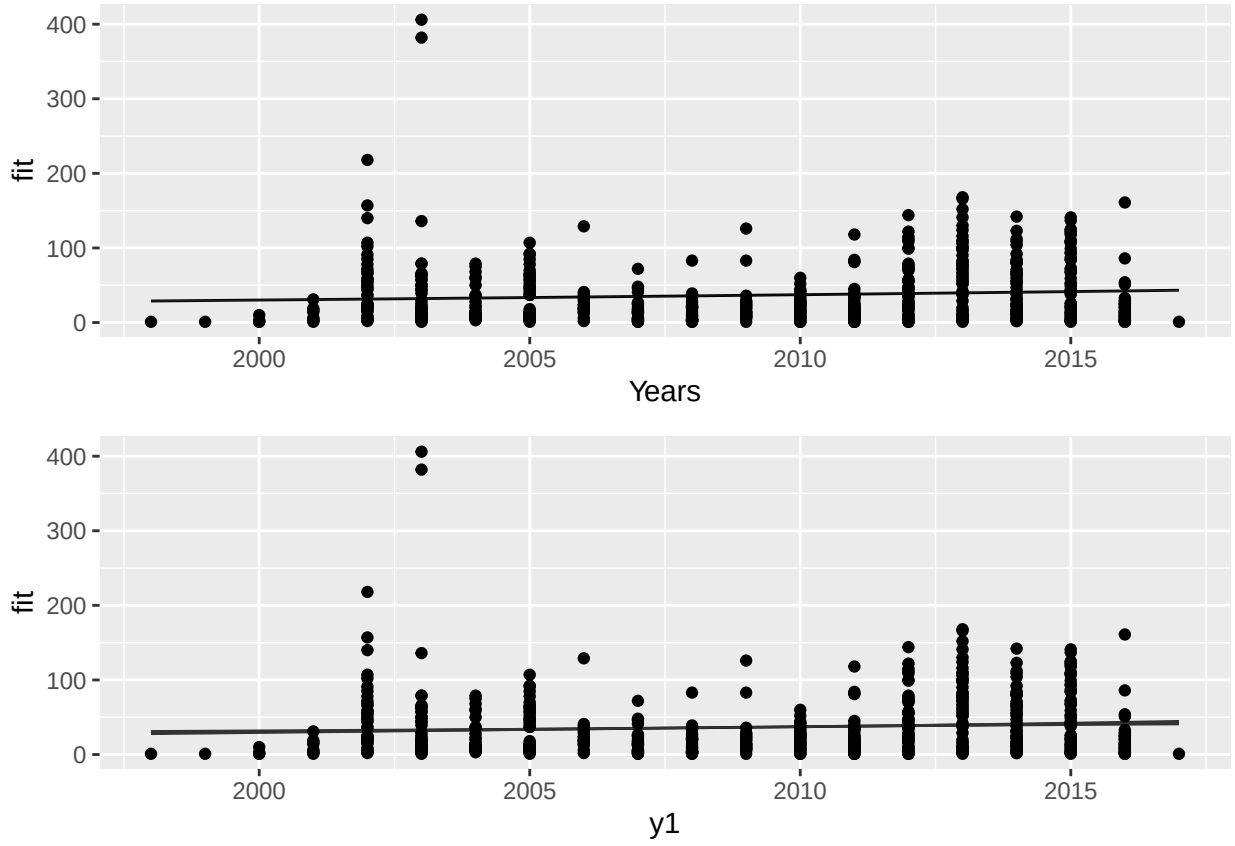
```
value_nb_predict <- predict(mod_kill_nb,Road_kill_sum,type = "response",se.fit = T)
value_nb_predict <- as.data.frame(value_nb_predict)
value_nb_predict <- cbind(y1 = Road_kill_sum$Year,value_nb_predict)
#let plot wrong poisson model, nb model with data points
plot <- ggplot(data =value_pred,aes(x=Years,y=fit))
p_wrog <-plot+ geom_line()+
  geom_ribbon(aes(ymax=fit+se.fit,ymin=fit-se.fit))+
  geom_point(data = Road_kill_sum,aes(x=Year,y=count))



plot1 <- ggplot(data =value_nb_predict,aes(x=y1,y=fit))
P1 <-plot1+ geom_line()+
  geom_ribbon(aes(ymax=fit+se.fit,ymin=fit-se.fit))+
  geom_point(data = Road_kill_sum,aes(x=Year,y=count))

library(gggrid)
```

```
## Loading required package: grid
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
grid.arrange(p_wrog,P1,nrow=2,ncol=1)
```




```
summary(mod_kill_nb)
```

```
##
## Call:
## glm.nb(formula = count ~ Year, data = Road_kill_sum, init.theta = 0.9550214435,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1537  -1.0709  -0.4596   0.3815   4.1040
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -35.81447   18.68701  -1.917   0.0553 .
## Year          0.01962    0.00930   2.109   0.0349 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.955) family taken to be 1)
##
##     Null deviance: 657.5  on 573  degrees of freedom
## Residual deviance: 652.7  on 572  degrees of freedom
## AIC: 5306.2
##
## Number of Fisher Scoring iterations: 1
```

```
##
##
##               Theta:  0.9550
##           Std. Err.:  0.0523
##
##  2 x log-likelihood:  -5300.1560
```

```
mod_kill_nb_add <- glm.nb(count ~ Year+Class, data = Road_kill_sum)
mod_kill_nb_int <- glm.nb(count ~ Year*Class, data = Road_kill_sum)
anova(mod_kill_nb_int)
```

```
## Warning in anova.negbin(mod_kill_nb_int): tests made without re-estimating
## 'theta'
```

```
## Analysis of Deviance Table
##
## Model: Negative Binomial(1.0686), link: log
##
## Response: count
##
## Terms added sequentially (first to last)
##
##
##            Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                         573     730.63
## Year        1    5.346       572     725.28  0.020770 *
## Class       3   14.750       569     710.53  0.002044 **
## Year:Class  3   64.878       566     645.66 5.326e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(mod_kill_nb_add)
```

```
##
## Call:
## glm.nb(formula = count ~ Year + Class, data = Road_kill_sum,
##     init.theta = 0.9740275498, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2318  -1.0394  -0.5069   0.4029   4.6499
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -31.284477  18.781122  -1.666  0.09577 .
## Year            0.017264   0.009342   1.848  0.06459 .
## ClassAves       0.263479   0.133404   1.975  0.04826 *
## ClassMammalia   0.358005   0.125865   2.844  0.00445 **
## ClassReptilia   0.009069   0.134170   0.068  0.94611
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for Negative Binomial(0.974) family taken to be 1)
##
##     Null deviance: 669.80  on 573  degrees of freedom
## Residual deviance: 651.45  on 569  degrees of freedom
## AIC: 5298.8
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.9740
##           Std. Err.:  0.0536
##
##  2 x log-likelihood:  -5286.8150
```

```
summary(mod_kill_nb_int)
```

```
##
## Call:
## glm.nb(formula = count ~ Year * Class, data = Road_kill_sum,
##     init.theta = 1.068570368, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5359  -1.0109  -0.4264   0.3651   3.4842
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         156.46001   42.79567   3.656 0.000256 ***
## Year                 -0.07618    0.02129  -3.579 0.000345 ***
## ClassAves          -411.77499   56.99499  -7.225 5.02e-13 ***
## ClassMammalia      -219.50759   51.94958  -4.225 2.39e-05 ***
## ClassReptilia      -145.72396   57.28661  -2.544 0.010966 *
## Year:ClassAves        0.20501    0.02836   7.230 4.83e-13 ***
## Year:ClassMammalia    0.10944    0.02585   4.234 2.30e-05 ***
## Year:ClassReptilia    0.07254    0.02850   2.545 0.010921 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.0686) family taken to be 1)
##
##     Null deviance: 730.63  on 573  degrees of freedom
## Residual deviance: 645.66  on 566  degrees of freedom
## AIC: 5242.8
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  1.0686
##           Std. Err.:  0.0597
##
##  2 x log-likelihood:  -5224.8260
```

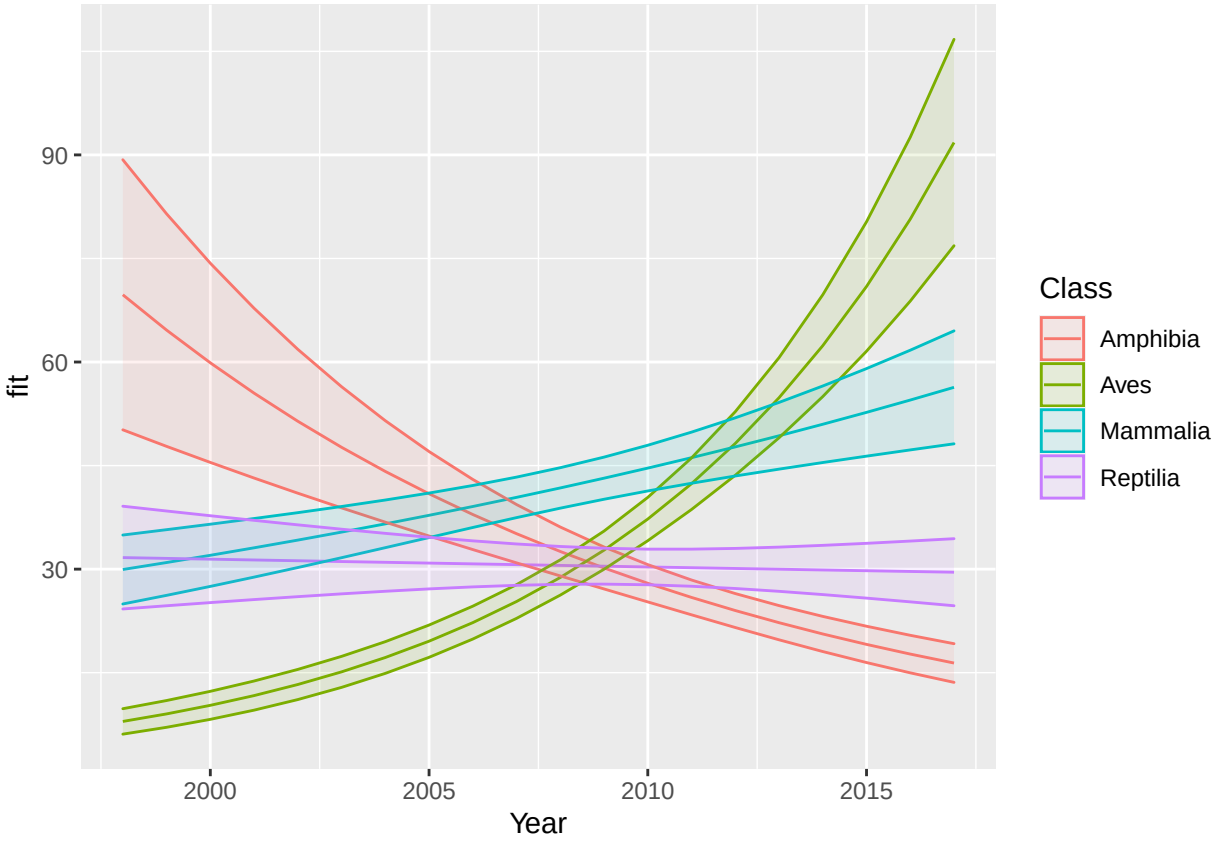
```
AIC(mod_kill_nb,mod_kill_nb_add,mod_kill_nb_int)
```

```
##                  df      AIC
## mod_kill_nb       3 5306.156
## mod_kill_nb_add   6 5298.815
## mod_kill_nb_int   9 5242.826
```

```
library(AICcmodavg)
#define list of models
models <- list(mod_kill_nb,mod_kill_nb_add,mod_kill_nb_int)

#specify model names
mod.names <- c('mod_kill_nb', 'mod_kill_nb_add',"mod_kill_nb_int")

#calculate AIC of each model
aictab(cand.set = models, modnames = mod.names)
```

```
##
## Model selection based on AICc:
##
##                  K    AICc Delta_AICc AICcWt Cum.Wt       LL
## mod_kill_nb_int  9 5243.15       0.00      1      1 -2612.41
## mod_kill_nb_add  6 5298.96      55.82      0      1 -2643.41
## mod_kill_nb      3 5306.20      63.05      0      1 -2650.08
```

```
Year <- unique(Road_kill_sum$Year)
Class<-unique(Road_kill$Class)
Year <- c(Year,Year,Year,Year)
Class <- c(rep(Class[1],20),rep(Class[2],20),rep(Class[3],20),rep(Class[4],20))
newdata <- data.frame(Year,Class)
interactive_pred_value <- predict(mod_kill_nb_int,newdata, type = "response",se.fit = T)
interactive_pred_value <- as.data.frame(interactive_pred_value)
interactive_pred_value <- cbind(interactive_pred_value,newdata)
plot_int <- ggplot(data = interactive_pred_value,
                   aes(x=Year,y=fit,col=Class))
plot_int + geom_line()+
  geom_ribbon(aes(ymax=fit+se.fit,ymin=fit-se.fit,fill=Class),alpha=0.1)
```


90
60
30
fit
2000
2005
2010
2015
Year
Class
Amphibia
Aves
Mammalia
Reptilia