Part II: Generalized Linear Models

Load Packages

Again, we must load the packages that will be used in the first part of this workshop.

```r
library(pastecs, quietly = TRUE)
library(lm.beta,  quietly = TRUE)
library(lmtest,  quietly = TRUE)
library(foreign,  quietly = TRUE)
library(lattice,  quietly = TRUE)
library(lme4,  quietly = TRUE)
library(nlme,  quietly = TRUE)
library(survival,  quietly = TRUE)
library(dplyr,  quietly = TRUE)
library(ggfortify,  quietly = TRUE)
library(survminer,  quietly = TRUE)
library(rms,  quietly = TRUE)
library(MASS, quietly = TRUE)
```

Generalized linear models

A generalized linear model (GLM) has three:

- a random component with mean μ. Generally, the random component is the response variable Y_i.
- a systematic component, η_i, that relates the relates the explanatory variables,

$$\eta_i = \sum_{j=i}^{n} \beta_j x_{ij}$$

- a link function that relates the mean of the random to the systematic component

$$g(\mu) = \eta_i$$

Logistic regression

Logistic regression is a GLM used the model binary (0 or 1) data. The response variable must be binary and is assumed to follow a bernoulli distribution.

That said, logistic regression has the following properties: - a response binary variable, Y_i, that follows a bernoulli distribution with mean π_i. - a systematic component, η_i, that relates the relates the explanatory variables,

$$\eta_i = \sum_{j=1}^{n} \beta_j x_{ij}$$

- a link function that relates the mean of the random to the systematic component

$$\log\left(\frac{\pi_i}{1 - \pi_i}\right) = \sum_{j=i}^{n} \beta_j x_{ij}.$$

$\log\left(\frac{\pi_i}{1-\pi_i}\right)$ is known as the log odds.

Data

Using the iris data, we create binary data. We add the column `Sepal.Width_binary` to iris. If the `Sepal.Width` is greater than the median then the associated value in `Sepal.Width_binary` is 1. Otherwise, `Sepal.Width_binary` is 0.

```
data <- iris
data$Sepal.Width_binary <- ifelse(data$Sepal.Width >= median(data$Sepal.Width), 1, 0)
```

Logistic Regression with only the constant term

Fitting only a constant term, the systematic component is

$$\eta_i = \beta_0$$

```
logit <- glm(Sepal.Width_binary ~ 1, data = data, family = "binomial")
summary(logit)
```

```
##
## Call:
## glm(formula = Sepal.Width_binary ~ 1, family = "binomial", data = data)
##
## Deviance Residuals:
##     Min      1Q   Median      3Q      Max
## -1.3911  -1.3911   0.9778   0.9778   0.9778
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.4895     0.1682    2.91  0.00361 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 199.22  on 149  degrees of freedom
## Residual deviance: 199.22  on 149  degrees of freedom
## AIC: 201.22
##
## Number of Fisher Scoring iterations: 4
```

```
p_avg <- mean(data$Sepal.Width_binary)
log_odds_avg <- log(p_avg/(1-p_avg))
print(log_odds_avg)
```

```
## [1] 0.4895482
```

Logistic Regression with Species

Fitting the species term, the systematic component is

$$\eta_i = 1 + \beta_2 X_{1i} + \beta_3 X_{2i}.$$

where

$$X_{1i} = \begin{cases} 1 & \text{if } i\text{th data point is versicolor} \\ 0 & \text{otherwise} \end{cases}, \ X_{2i} = \begin{cases} 1 & \text{if } i\text{th data point is virginica} \\ 0 & \text{otherwise} \end{cases}$$

```r
logit <- glm(Sepal.Width_binary ~ as.factor(Species), data = data, family = "binomial")
summary(logit)
```

```
##
## Call:
## glm(formula = Sepal.Width_binary ~ as.factor(Species), family = "binomial",
##     data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5373  -0.8782   0.2857   1.0438   1.5096
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   3.1781     0.7215   4.405 1.06e-05 ***
## as.factor(Species)versicolor -3.9318     0.7826  -5.024 5.06e-07 ***
## as.factor(Species)virginica  -2.8553     0.7763  -3.678 0.000235 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 199.22  on 149  degrees of freedom
## Residual deviance: 147.51  on 147  degrees of freedom
## AIC: 153.51
##
## Number of Fisher Scoring iterations: 5
```

Let's compare the results to the average log odds of each Species group

```r
log_odds_avg_fun <- function(data){
  p_avg <- mean(data)
  log_odds_avg <- log(p_avg/(1-p_avg))
  return(log_odds_avg)
}

tapply(data$Sepal.Width_binary,
       data$Species, log_odds_avg_fun)
```

```
##     setosa versicolor  virginica
##  3.1780538 -0.7537718  0.3227734
```

The intercept corresponds to the average log odds of setosa as we would expect. However, the other coefficients do not correspond to the average log odds of the other species. Why?

From the formula, $\eta_i = 1 + \beta_2 X_{2i} + \beta_3 X_{3i}$, the log odds of versicolor actually corresponds to $1 + \beta_2$. The log odds of versicolor actually corresponds to $1 + \beta_3$.

```r
coefficients<-unname(coef(logit))
print(c(coefficients[1],coefficients[1]+coefficients[2],
        coefficients[1]+coefficients[3]))
```

```
## [1]  3.1780537 -0.7537718  0.3227734
```

Logistic Regression with Sepal.Length

Fitting the species term, the systematic component is

$$\eta_i = \beta_3 X_{1i}.$$

where
$$X_{1i} = \begin{cases} 1 & \text{if } i\text{th data point is versicolor} \\ 0 & \text{otherwise} \end{cases}, \; X_{2i} = \begin{cases} 1 & \text{if } i\text{th data point is virginica} \\ 0 & \text{otherwise} \end{cases}$$
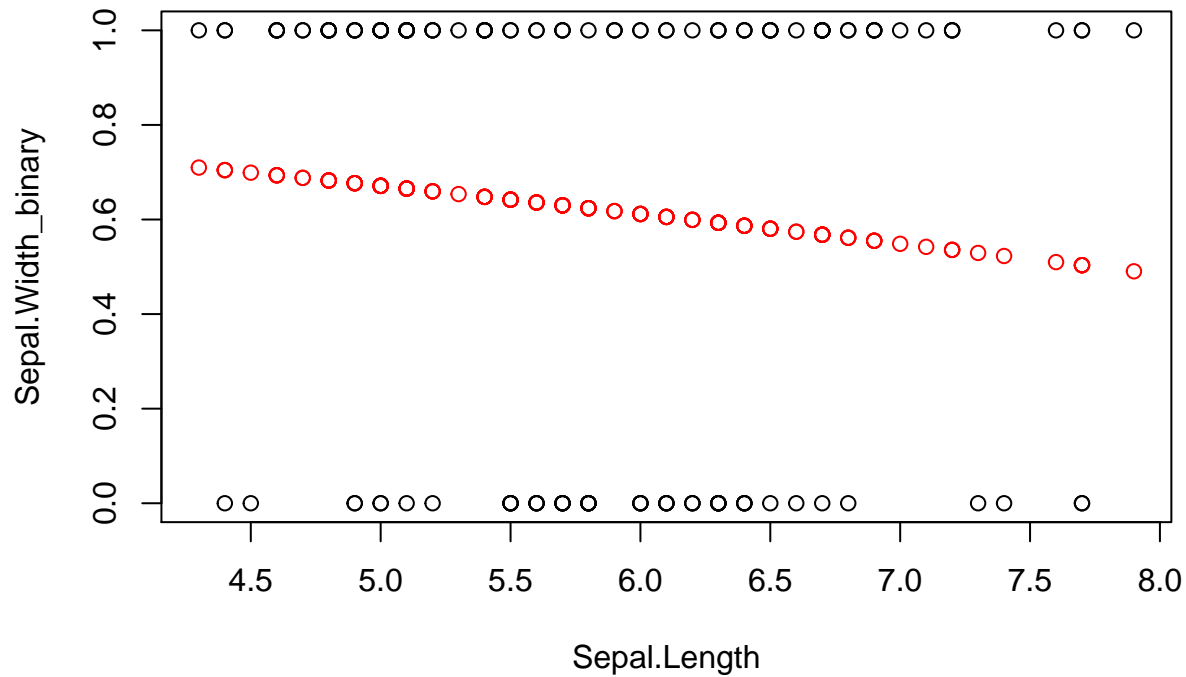
and $X_{3i} = $ Sepal.Length of the ith data point.

```
logit <- glm(Sepal.Width_binary ~ Sepal.Length,
             data = data, family = "binomial")
summary(logit)
```

```
##
## Call:
## glm(formula = Sepal.Width_binary ~ Sepal.Length, family = "binomial",
##     data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5614  -1.3524   0.8883   0.9890   1.1936
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    2.0088     1.2176   1.650    0.099 .
## Sepal.Length  -0.2591     0.2050  -1.264    0.206
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 199.22  on 149  degrees of freedom
## Residual deviance: 197.61  on 148  degrees of freedom
## AIC: 201.61
##
## Number of Fisher Scoring iterations: 4
```

```
plot(Sepal.Width_binary~Sepal.Length, data=data)
points(data$Sepal.Length[order(data$Sepal.Length)],
       logit$fitted[order(data$Sepal.Length)],  col="red")
title(main="Data with Fitted Logistic Regression Line")
```

Data with Fitted Logistic Regression Line

Logistic Regression with Species and Sepal.Length

Fitting the species term, the systematic component is

$$\eta_i = 1 + \beta_2 X_{1i} + \beta_3 X_{2i} + \beta_3 X_{3i}.$$

where

$$X_{1i} = \begin{cases} 1 & \text{if } i\text{th data point is versicolor} \\ 0 & \text{otherwise} \end{cases}, \ X_{2i} = \begin{cases} 1 & \text{if } i\text{th data point is virginica} \\ 0 & \text{otherwise} \end{cases}$$

and $X_{3i} = $ Sepal.Length of the ith data point.

Fitting the logistic model accordingly,

```
logit <- glm(Sepal.Width_binary ~ Species +Sepal.Length,
            data = data, family = "binomial")
summary(logit)
```

```
##
## Call:
## glm(formula = Sepal.Width_binary ~ Species + Sepal.Length, family = "binomial",
##     data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2710  -0.7538   0.2472   0.7020   1.9477
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -4.7988     2.2981  -2.088 0.036784 *
```

```
## Speciesversicolor   -5.6936     0.9686  -5.878 4.16e-09 ***
## Speciesvirginica    -5.4812     1.0879  -5.039 4.69e-07 ***
## Sepal.Length         1.6219     0.4510   3.596 0.000323 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 199.22  on 149  degrees of freedom
## Residual deviance: 131.27  on 146  degrees of freedom
## AIC: 139.27
##
## Number of Fisher Scoring iterations: 6
```

Plot the results for each species, we get that

```
plot(data[data$Species == "setosa", ]$Sepal.Length,
     data[data$Species == "setosa", ]$Sepal.Width_binary,
     xlim=as.matrix(range(data$Sepal.Length)),
     xlab = 'Sepal Length',  ylab= 'Sepal Width binary',
     main= 'Scatter plot of sepal length vs sepal width')

points(data$Sepal.Length[data$Species == "setosa"],
       logit$fitted[data$Species == "setosa"],  pch=15,
       col="red")
```
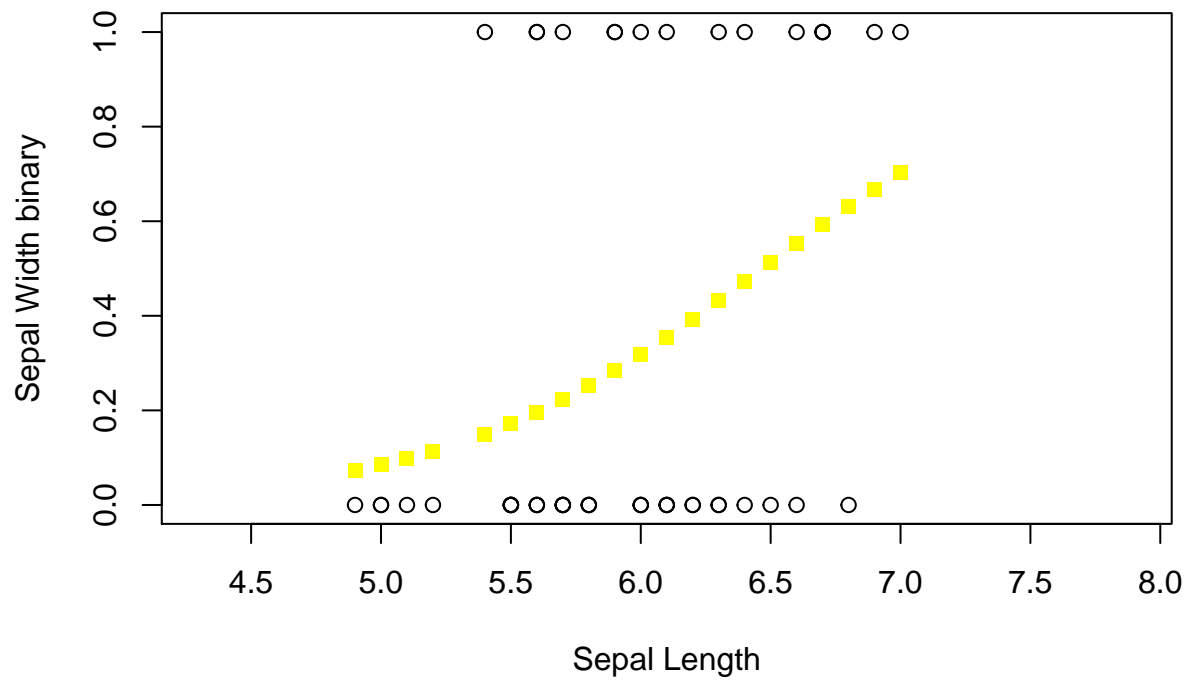
Scatter plot of sepal length vs sepal width
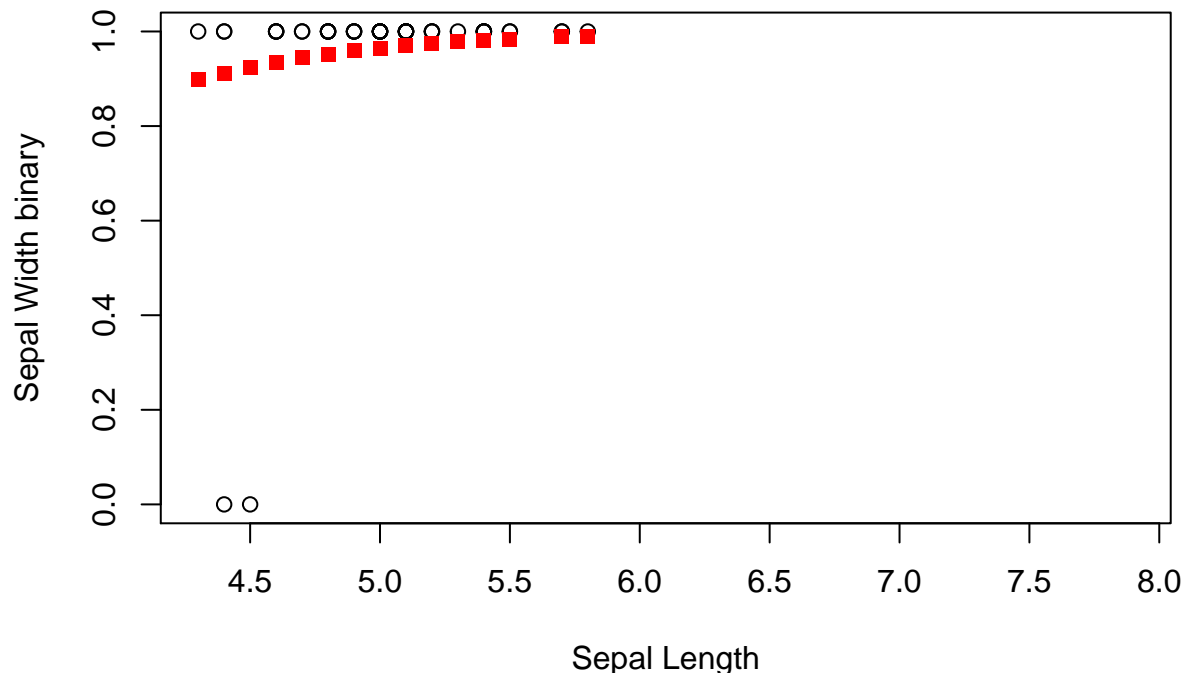

```
plot(data[data$Species == "versicolor", ]$Sepal.Length,
     data[data$Species == "versicolor", ]$Sepal.Width_binary,
     xlim=as.matrix(range(data$Sepal.Length)),
     xlab = 'Sepal Length',  ylab= 'Sepal Width binary',
     main= 'Scatter plot of sepal length vs sepal width')
```

```
points(data$Sepal.Length[data$Species == "versicolor"],
       logit$fitted[data$Species == "versicolor"],  pch=15,
       col="yellow")
```

Scatter plot of sepal length vs sepal width


```
plot(data[data$Species == "virginica", ]$Sepal.Length,
     data[data$Species == "virginica", ]$Sepal.Width_binary,
     xlim=as.matrix(range(data$Sepal.Length)),
     xlab = 'Sepal Length',  ylab= 'Sepal Width binary',
     main= 'Scatter plot of sepal length vs sepal width')

points(data$Sepal.Length[data$Species == "virginica"],
       logit$fitted[data$Species == "virginica"],  pch=15,
       col="blue")
```

Scatter plot of sepal length vs sepal width
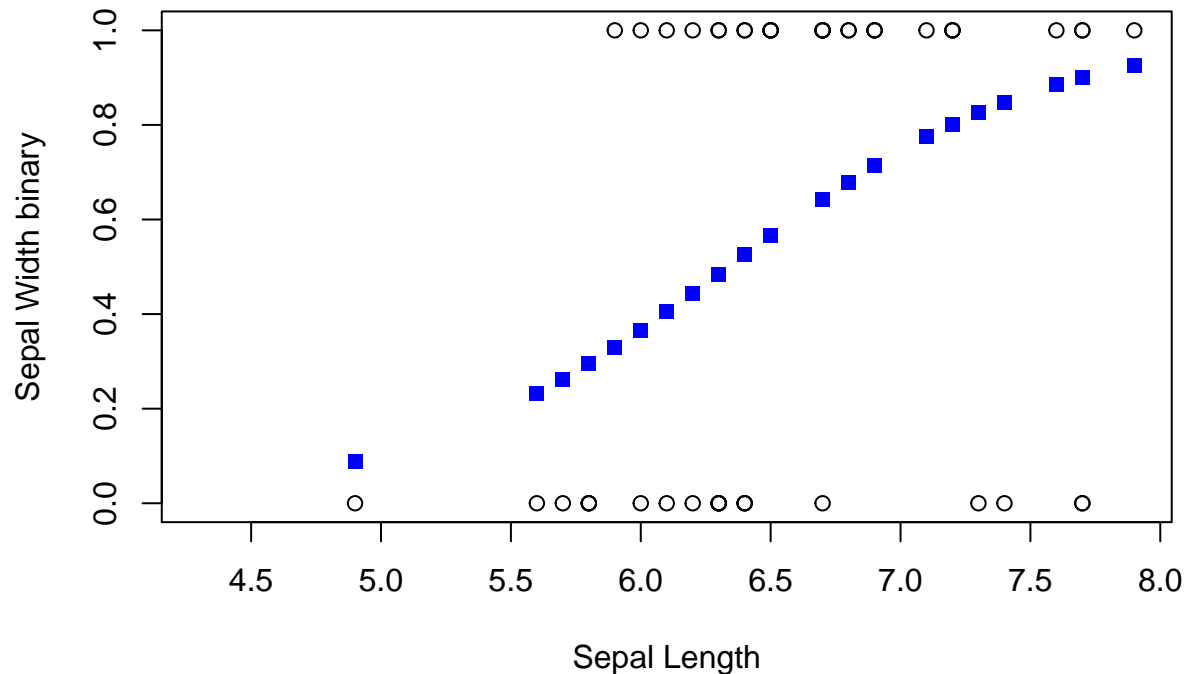
Deviance

For general linear models, we use *deviance* to the compare to two different models. Deviance is the difference in log likelihood of the models multipled by 2.

Saturated Model

Let's consider model in which each data point has its own mean and coefficients. This is called the saturated model. It basically replicates the data at hand.

Using deviance, we can compare our fitted model to a saturated model. If the fitted model is behaves similiar to the saturated model, then the deviance can be well approximated by a chi-squared distribution with $m - n$ degrees of freedom. m is number of the data points and n is number of coefficients in our fitted model.

This satistical property of the deviance allows us perform a hypothesis test

$$H_0 : \text{the fitted model is equivalent to the saturated model}$$

$$H_\alpha : \text{the fitted model is not equivalent to the saturated model}$$

`logit$deviance` is the deviance between saturated model and fitted model. `logit$df.residual` is equal to number of observations minus the number of coefficients in the fitted model. Using this, we can calculate the p value for the hypothesis test above.

```
p_value = 1 - pchisq(logit$deviance, logit$df.residual)
print(p_value)
```

```
## [1] 0.8032738
```

Since the p value is less than 0.05, we fail to reject the null hypothesis. (This is a good thing.)

Null Model

We can also use deviance to determine if our fitted model is better than the null model. The null model is is a model with only a linear term. Like above, we can design a hypothesis test comparing the null model to the fitted model.

$$H_0 = \text{ the fitted model is equivalent to the null model}$$

$$H_\alpha = \text{ the fitted model is not equivalent to the null model}$$

In the limit of large data, it is known that the deviance follows a chi-squared distribution with parameter $n - 1$.

`logit$deviance` is the deviance between saturated model and fitted model. `logit$df.residual` is equal to number of observations minus the number of coefficients in the fitted model.

`logit$null.deviance` is the deviance between saturated model and the null model. `logit$df.null` is the number of observations minus 1.

Using this information, we can calculate the p value for the hypothesis test above.

```
p_value = 1 - pchisq(logit$null.deviance-logit$deviance,
                     logit$df.null-logit$df.residual)
print(p_value)
```

```
## [1] 1.176836e-14
```

Since the p value is less than one, we reject our null hypothesis. (This is a good thing.)

Anova

Sequencial comparison of model terms by deviance

```
anova(logit,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Sepal.Width_binary
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                           149     199.22
## Species       2   51.709       147     147.51 5.910e-12 ***
## Sepal.Length  1   16.239       146     131.27 5.583e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Poisson and Quasi-Poisson Regression

```
attach(epil)
summary(epil)
```

```
##        y                  trt           base           age
##  Min.   :  0.000   placebo  :112   Min.   :  6.00   Min.   :18.00
##  1st Qu.:  2.750   progabide:124   1st Qu.: 12.00   1st Qu.:23.00
##  Median :  4.000                   Median : 22.00   Median :28.00
##  Mean   :  8.254                   Mean   : 31.22   Mean   :28.34
##  3rd Qu.:  9.000                   3rd Qu.: 41.00   3rd Qu.:32.00
##  Max.   :102.000                   Max.   :151.00   Max.   :42.00
##        V4            subject        period         lbase
##  Min.   :0.00   Min.   : 1   Min.   :1.00   Min.   :-1.36249
##  1st Qu.:0.00   1st Qu.:15   1st Qu.:1.75   1st Qu.:-0.66934
##  Median :0.00   Median :30   Median :2.50   Median :-0.06321
##  Mean   :0.25   Mean   :30   Mean   :2.50   Mean   : 0.00000
##  3rd Qu.:0.25   3rd Qu.:45   3rd Qu.:3.25   3rd Qu.: 0.55932
##  Max.   :1.00   Max.   :59   Max.   :4.00   Max.   : 1.86303
##       lage
##  Min.   :-0.42941
##  1st Qu.:-0.18429
##  Median : 0.01242
##  Mean   : 0.00000
##  3rd Qu.: 0.14595
##  Max.   : 0.41789
```

```
sapply(grouseticks, class)
```

```
##     INDEX     TICKS     BROOD    HEIGHT      YEAR  LOCATION   cHEIGHT
##  "factor" "numeric"  "factor" "numeric"  "factor"  "factor" "numeric"
```

```
ggplot(grouseticks,aes(x=grouseticks$TICKS))+
  geom_histogram(binwidth = 1, center = 0.5) +
  scale_x_continuous(breaks=seq(1,max(grouseticks$TICKS), by = 5))+
  ylab("Count")+ xlab("data")+
  ggtitle("Histogram plot of the number of ticks on the heads of red grouse chicks")
```

Histogram plot of the number of ticks on the heads of red grouse chicks
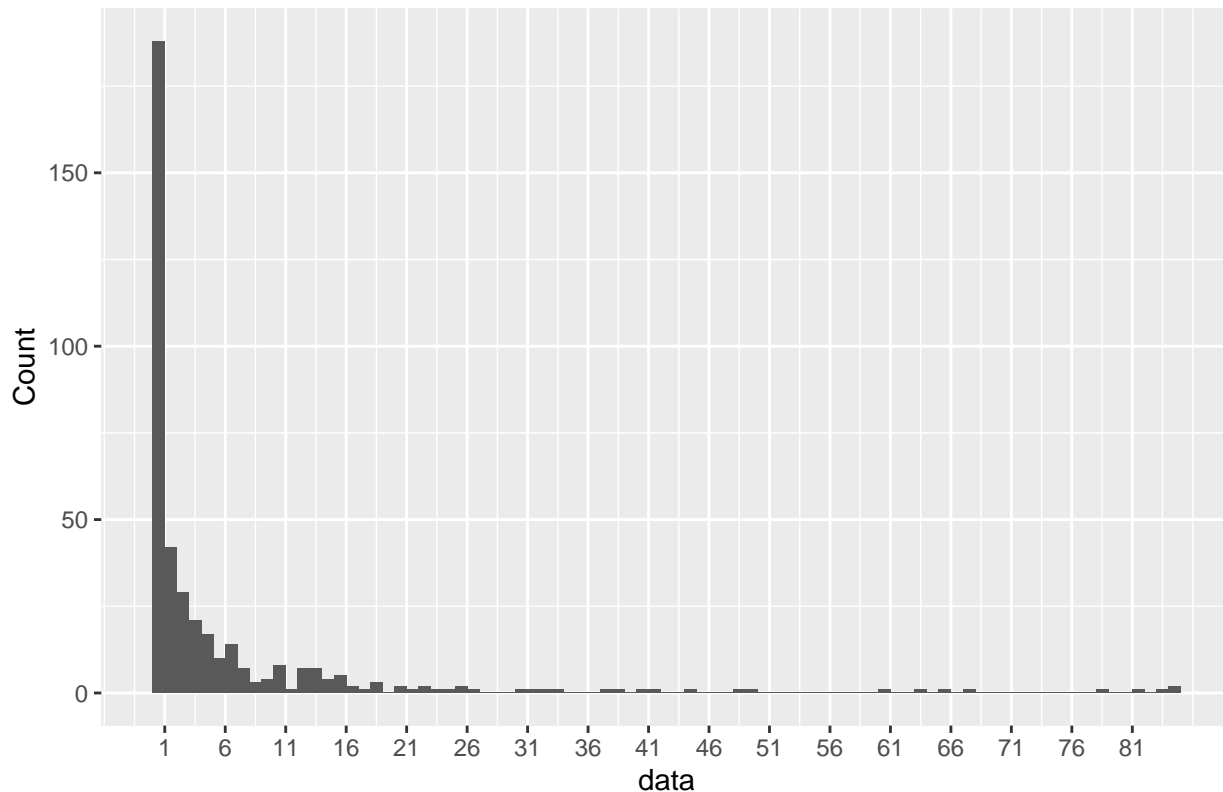

```r
model = glm(TICKS~ 1, family=poisson(link=log),data=grouseticks)
summary(model)
```

```
##
## Call:
## glm(formula = TICKS ~ 1, family = poisson(link = log), data = grouseticks)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.5692  -3.5692  -2.0263  -0.1479  16.8293
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.85156    0.01974   93.81   <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: 5847.5  on 402  degrees of freedom
## Residual deviance: 5847.5  on 402  degrees of freedom
## AIC: 6780.5
##
## Number of Fisher Scoring iterations: 6
```

```r
print(coef(model))
```

```
## (Intercept)
```

```
##     1.851557
```

```
head(data.frame(grouseticks$TICKS,model$fitted))
```

```
##    grouseticks.TICKS model.fitted
## 1                  0     6.369727
## 2                  0     6.369727
## 3                  0     6.369727
## 4                  0     6.369727
## 5                  0     6.369727
## 6                  3     6.369727
```

```
head(model$linear.predictors)
```

```
##        1        2        3        4        5        6
## 1.851557 1.851557 1.851557 1.851557 1.851557 1.851557
```

```
head(exp(model$linear.predictors))
```

```
##        1        2        3        4        5        6
## 6.369727 6.369727 6.369727 6.369727 6.369727 6.369727
```
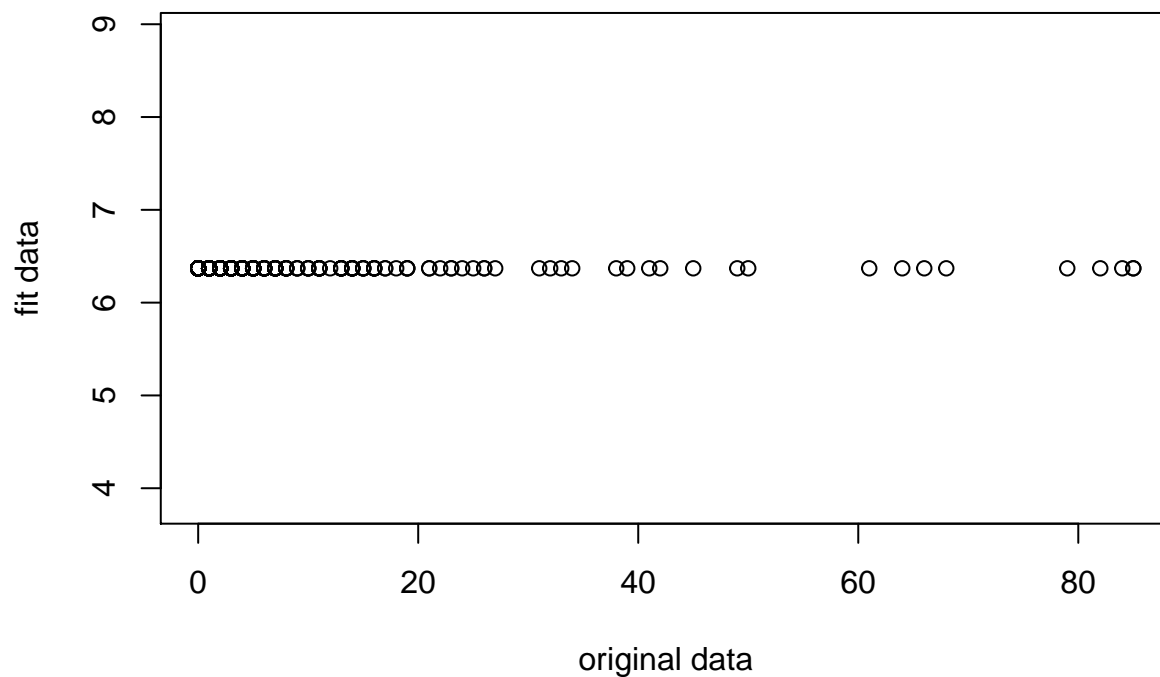
Hypothesis test for goodness of fit

```
print(1-pchisq(model$deviance,model$df.residual))
```

```
## [1] 0
```

plot data comparison

```
df_original = data.frame(data=grouseticks$TICKS)
df_fitted = data.frame(data=model$fitted)
plot(df_original$data,df_fitted$data,ylab='fit data',xlab='original data')
```


add a Covariate to the fit – treatment

```
model = glm(TICKS~ 1 +HEIGHT + LOCATION, family=poisson(link=log),grouseticks)
summary(model)
```

```
##
## Call:
## glm(formula = TICKS ~ 1 + HEIGHT + LOCATION, family = poisson(link = log),
##     data = grouseticks)
##
## Deviance Residuals:
##     Min      1Q   Median      3Q      Max
## -8.9312  -1.2247  -0.5707   0.8354  10.0417
##
## Coefficients: (1 not defined because of singularities)
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.316e+01  1.451e+00   9.067  < 2e-16 ***
## HEIGHT       -2.662e-02  3.533e-03  -7.535 4.89e-14 ***
## LOCATION2    -1.984e+00  5.103e-01  -3.888 0.000101 ***
## LOCATION3    -1.626e+00  2.955e-01  -5.502 3.76e-08 ***
## LOCATION4     7.622e-01  1.139e-01   6.692 2.21e-11 ***
## LOCATION5     7.407e-01  1.732e-01   4.277 1.89e-05 ***
## LOCATION6     5.232e-01  1.395e-01   3.749 0.000177 ***
## LOCATION7     1.871e+00  1.186e-01  15.773  < 2e-16 ***
## LOCATION8     2.450e-01  2.599e-01   0.943 0.345791
## LOCATION9     3.770e-01  1.931e-01   1.953 0.050877 .
## LOCATION10   -8.003e-01  3.517e-01  -2.275 0.022879 *
## LOCATION11    5.256e-01  1.676e-01   3.136 0.001716 **
## LOCATION12   -4.594e-01  3.109e-01  -1.478 0.139467
## LOCATION13   -1.830e-01  1.978e-01  -0.925 0.354810
## LOCATION14    6.519e-02  1.502e-01   0.434 0.664346
## LOCATION15   -1.969e+00  4.660e-01  -4.226 2.38e-05 ***
## LOCATION16    3.344e-03  2.905e-01   0.012 0.990816
## LOCATION17   -2.269e-01  3.325e-01  -0.682 0.494956
## LOCATION18    1.365e-01  2.962e-01   0.461 0.645038
## LOCATION19    1.788e-02  1.886e-01   0.095 0.924480
## LOCATION20    9.944e-01  2.075e-01   4.791 1.66e-06 ***
## LOCATION21   -2.412e-01  5.986e-01  -0.403 0.686956
## LOCATION22   -3.703e-01  2.771e-01  -1.336 0.181465
## LOCATION23    1.449e+00  1.929e-01   7.514 5.75e-14 ***
## LOCATION24    2.989e+00  1.807e-01  16.544  < 2e-16 ***
## LOCATION25   -2.357e-01  2.936e-01  -0.803 0.422112
## LOCATION26    4.559e-01  3.135e-01   1.454 0.145894
## LOCATION27   -3.272e-01  4.514e-01  -0.725 0.468499
## LOCATION28    2.207e-01  2.321e-01   0.951 0.341776
## LOCATION29   -3.793e-01  3.880e-01  -0.978 0.328262
## LOCATION30    3.591e-01  2.862e-01   1.255 0.209620
## LOCATION31   -1.095e+00  4.624e-01  -2.368 0.017869 *
## LOCATION32   -1.608e+01  9.021e+02  -0.018 0.985775
## LOCATION33   -2.433e-01  4.999e-01  -0.487 0.626437
## LOCATION34   -2.310e+00  1.026e+00  -2.252 0.024340 *
## LOCATION35   -1.595e+01  1.276e+03  -0.013 0.990025
## LOCATION36    1.052e+00  2.661e-01   3.953 7.73e-05 ***
## LOCATION37   -1.961e-01  3.930e-01  -0.499 0.617820
## LOCATION38   -1.181e+00  7.518e-01  -1.571 0.116180
## LOCATION39   -4.613e-01  6.326e-01  -0.729 0.465879
```

```
## LOCATION40    1.489e+00  3.086e-01   4.825 1.40e-06 ***
## LOCATION41  -7.070e-01  7.598e-01  -0.930 0.352127
## LOCATION42    4.448e-01  4.255e-01   1.045 0.295838
## LOCATION43  -1.552e+01  1.276e+03  -0.012 0.990291
## LOCATION44  -4.561e-01  6.482e-01  -0.704 0.481638
## LOCATION45  -1.542e+01  7.366e+02  -0.021 0.983300
## LOCATION46    2.641e+00  3.384e-01   7.805 5.94e-15 ***
## LOCATION47  -5.728e-01  6.543e-01  -0.875 0.381397
## LOCATION48    2.524e-01  5.890e-01   0.428 0.668301
## LOCATION49  -1.618e+00  1.048e+00  -1.544 0.122701
## LOCATION50    1.717e+00  3.345e-01   5.135 2.83e-07 ***
## LOCATION51  -1.208e+00  5.943e-01  -2.033 0.042069 *
## LOCATION52  -1.518e+01  7.366e+02  -0.021 0.983559
## LOCATION53    1.658e+00  3.987e-01   4.159 3.20e-05 ***
## LOCATION54  -1.497e+01  6.379e+02  -0.023 0.981283
## LOCATION55    1.895e+00  4.944e-01   3.832 0.000127 ***
## LOCATION56  -1.261e+00  7.990e-01  -1.578 0.114545
## LOCATION57  -3.501e-01  6.905e-01  -0.507 0.612091
## LOCATION58  -1.478e+01  9.021e+02  -0.016 0.986929
## LOCATION59  -1.475e+01  6.379e+02  -0.023 0.981549
## LOCATION60    2.134e+00  4.607e-01   4.633 3.60e-06 ***
## LOCATION61    7.101e-01  7.058e-01   1.006 0.314371
## LOCATION62    8.166e-01  5.360e-01   1.524 0.127616
## LOCATION63          NA          NA      NA        NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 5847.5  on 402  degrees of freedom
## Residual deviance: 1870.9  on 340  degrees of freedom
## AIC: 2927.9
##
## Number of Fisher Scoring iterations: 13
```

```r
print(1-pchisq(model$deviance,model$df.residual))
```

```
## [1] 0
```

```r
df_fitted = data.frame(data=model$fitted)
df_fitted$name = "fitted"
plot(df_original$data,df_fitted$data)
```
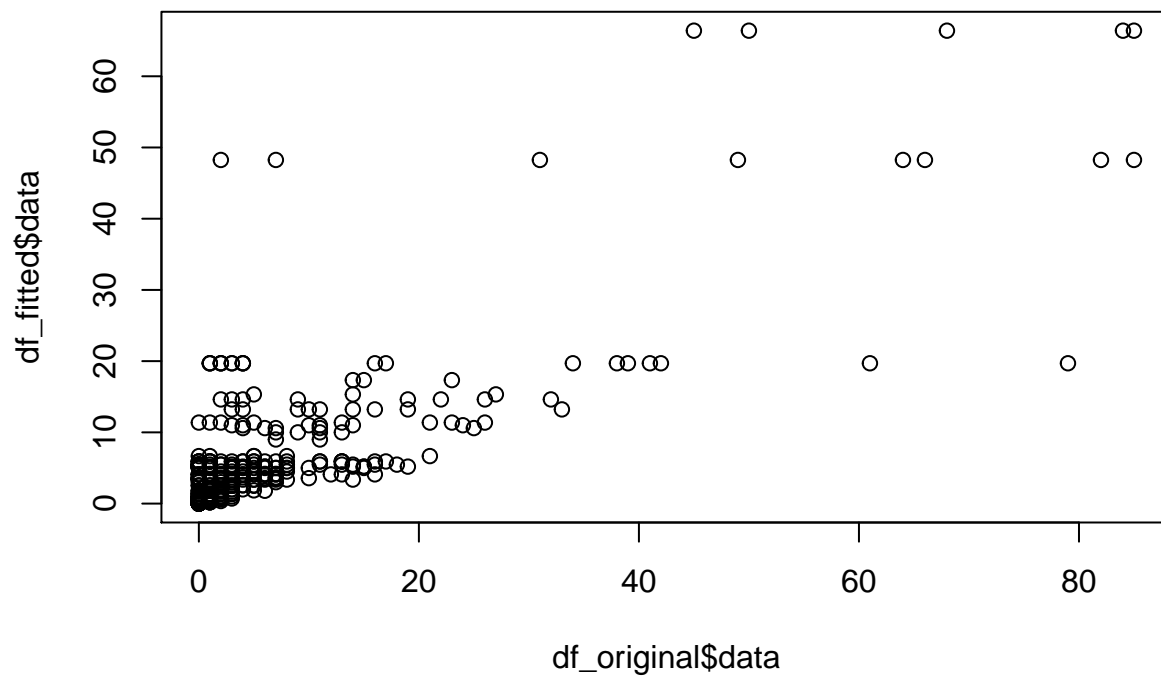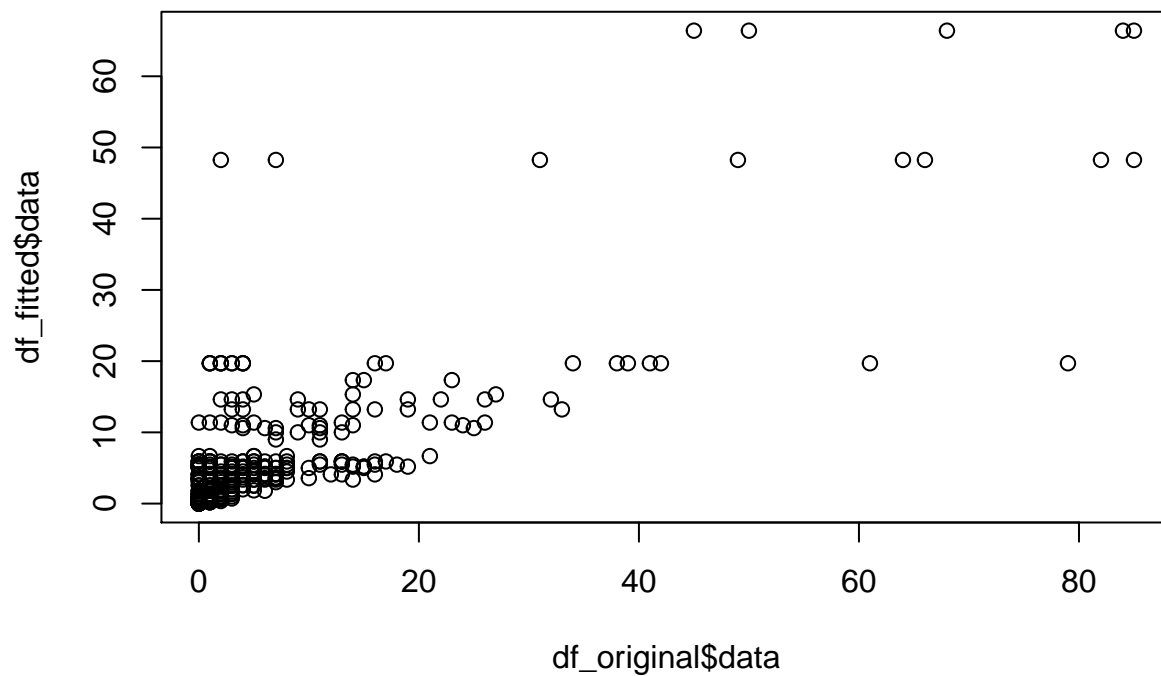
add a Covariate to the fit – treatment, age

```
#model = glm(y ~ 1 +trt*age, family=poisson(link=log),df)
#summary(model)
```

```
print(1-pchisq(model$deviance,model$df.residual))
```

```
## [1] 0
```

```
df_fitted = data.frame(data=model$fitted)
combined=rbind(df_original,df_fitted)
plot(df_original$data,df_fitted$data)
```

Overdispersion might be present – try Quasipossion

```
#model = glm(y ~ 1 +trt*age, family=quasipoisson(link=log),df)
#summary(model)

#print(1-pchisq(model$deviance,model$df.residual))

#df_fitted = data.frame(data=model$fitted)

#combined=rbind(df_original,df_fitted)

#plot(df_original$data,df_fitted$data)
#detach(epil)
```