# Statistical Modelling 2

CID:01724711

March 2023

## 1. Data Analysis

### 1.1 Introduction

In this report, we analyse data from a study into the development of children. The study compared the number of blocks that children were able to stack on top of one another. The team who collected the data are interested in how children's ability to stack blocks changes as they age. The data set we are using contains 4 variables:

- Number - the number of blocks that the child could successfully stack.
- Age - the child's age in completed years.
- Shape - the shape of blocks being stacked, either Cube or Cylinder.
- Child - categorical identifier for the child.

Each child was allowed two attempts for each shape, so there should be four rows for each child. For the purpose of this report we neglect the dependence between observations, treating them as independent. This is of course a limitation of the analysis.

Pros of treating observations as independent:

1. Simplicity: Treating observations as independent can simplify the analysis and make it easier to interpret the results.
2. Computational efficiency: Methods that assume independent observations may be faster and more computationally efficient than those that account for dependence, which can be an advantage for large datasets or computationally intensive models.
3. Transparency: Treating observations as independent is a common assumption in many statistical models, and can make the results more transparent and easier to communicate to non-experts.

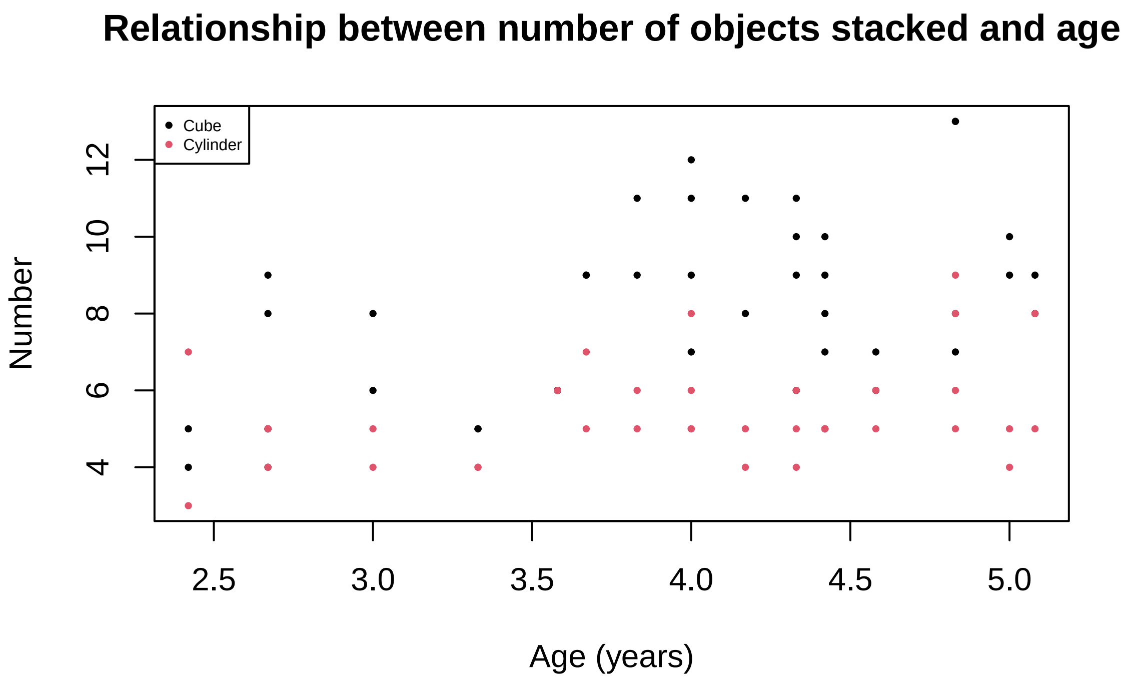
Cons of treating observations as independent:

1. Loss of information: By ignoring the dependence between observations, we may be losing valuable information about the correlation structure of the data, and potentially biasing our estimates of the model parameters and standard errors.
2. Incorrect inferences: Ignoring the dependence between observations can lead to incorrect statistical inferences, such as inflated significance levels or incorrect estimates of the effect sizes.
3. Misspecification: Ignoring the dependence between observations can lead to model misspecification, which can result in biased or inconsistent estimates.

In the context of the data we assume within-subject correlation is small and also the research question is focused on the effects of the predictor variables rather than the correlation structure, hence treating observations as independent may be reasonable.

### 1.2 Exploratory Analysis

First we begin by plotting the number of objects stacked against the age of the child to see the relationship between the two.



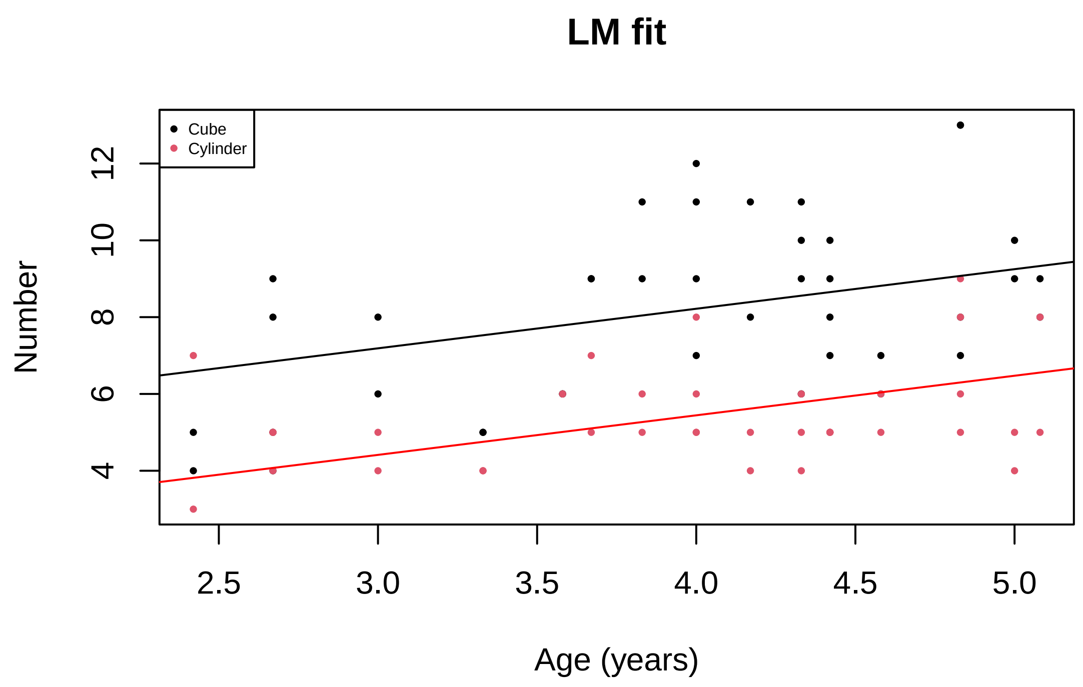
We observe that the trend seems to be that the number of cubes stacked is greater than the number of cylinders stacked for children of the same age and we also a weak positive corellation between the number of objects stacked, both cube and cylinder, and the age of the child.

## 1.3 Fitting initial linear model

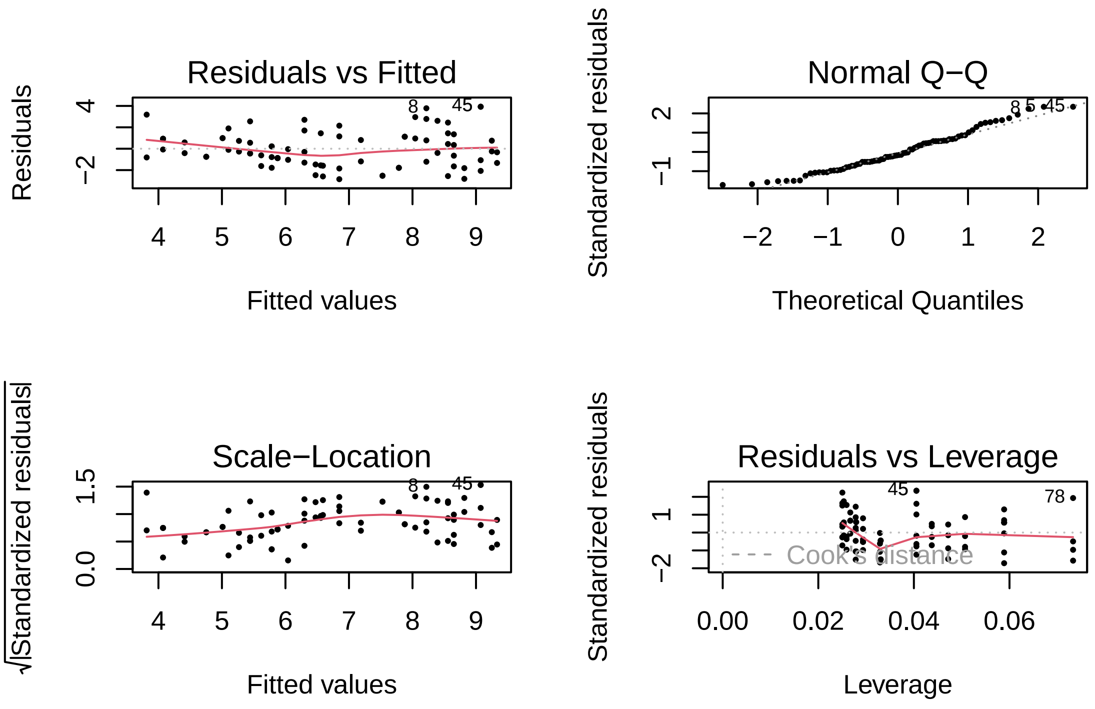
We fit the linear model suggested by the experimental team below:

$$\text{Number} \sim \text{Age} + \text{Shape}$$

It makes sense to fit an initial linear model from the exploratory analysis since the data looks reasonably well placed to be on a straight line for both cubes and cylinders. We can write this as $Y = X\beta + \epsilon$ where we assume iid $\epsilon_i \sim N(0,1)$ and have $Y = Number$ with design matrix $X = (\underline{1}, Age, I_{Shape='Cylinder'})$. We fit the model with the belief that children will linearly improve stacking both objects with age and that the constant rate of improvement, $\beta_1$ should not differ between shapes but that there should be a constant difference in the value of number of shapes stacked between the two shapes, $\beta_2$. We are able to plot the fit of the model on our data. We display lines for both cubes and cylinders.



We find adjusted $R^2$ to be 0.464, relatively low and so indicating potential poor performance. We now show the diagnostic plots for the model.



**The model performs approximately well. The residuals are quite closely distributed about 0 and small in absolute value. But the Q-Q plot displays a rather heavy tail, so empirical quantiles are not matching the theoretical ones. The scale-plot shows a weak correlation between mean and standard variance. We would expect this line to be flat since then the modelling assumption of constant variance would hold. Lastly, we see that all points have relatively small Cook's distance, implying small leverages and residuals.**
**The model is definitely improvable. We may hope to find a model where we do not see a modelling assumption violated and improved performance in the other diagnostic plots. We move onto fitting a different model in the next section that we hope to be a better one.**

## 1.4 Fitting a general linear model

**We now fit a poisson general linear model (GLM) using the canonical link. We do this because a Poisson GLM can be a suitable choice for modeling count data in situations where the response variable is a non-negative integer (i.e., counts), like we have.**

**We first explain the numerical scheme used to fit the model before we fit a few different Poisson GLM models with various linear predictors to choose the best performing one.**
**The key idea is to maximise the log likelihood. For this we can use the Iterative weighted least squares (IWLS) algorithm.**

1. **Given a current estimate $\hat{\beta}$, form the linear predictor $\hat{\eta}$ and the fitted values $\hat{\mu}$**
2. **Form the adjusted dependent variable:**
$$z_i = \hat{\eta_i} + (y_i - \hat{\mu_i})\frac{\partial \eta}{\partial \mu}|_{\mu=\hat{\mu_i}}$$
3. **Form the estimated weights $\tilde{w_{ii}}$ by:**
$$\tilde{w_{ii}}^{-1} = \left(\frac{\partial \eta}{\partial \mu}\right)^2 V(\mu)\frac{\partial \eta}{\partial \mu}|_{\mu=\hat{\mu_i}}$$
4. **Regress $z_i$ on $x_i$ with weights $\tilde{w_{ii}}$ and obtain the new estimate $\hat{\beta}$.**
5. **Repeat steps 1 to 4 until convergence.**

**We reform the pdf into exponential family form and calculate all necessary parts for the algorithm**
**For independent $Y_1, ..., Y_n$ where $Y_i \sim$ Poisson($\lambda_i$),**

$$f(y_i; \lambda_i) = \frac{e^{-\lambda_i}\lambda_i^{y_i}}{y_i!} = exp(\frac{y(\theta) - b(\theta)}{a(\phi)} + c(y, \phi))$$

**where we have (working with the identity link):**

$$\begin{aligned}
\mu_i &\equiv E(Y_i) = \lambda_i \\
\theta_i &= \log(\lambda_i) \\
a(\phi) &= 1 \\
b(\theta_i) &= \lambda_i = \exp(\theta_i) \\
b'(\theta_i) &= \exp(\theta_i) \\
b''(\theta_i) &= \exp(\theta_i) \\
V(\mu_i) &= b''(\theta_i) = \exp(\theta_i) = \lambda_i \\
\eta_i &= \mu_i = \lambda_i \\
\frac{\partial \eta_i}{\partial \mu_i} &= 1 \\
z_i &= \hat{\eta_i} + y_i - \hat{\lambda_i} = y_i \\
w_{ii}^{-1} &= \hat{\lambda_i} = \tilde{w_{ii}}^{-1} \text{ as } \phi = 1
\end{aligned}$$

**In a Poisson GLM with canonical link, the expected response of the $i$th observation**

$$E[Y_i] = \lambda_i = \exp(\eta_i) = \exp(X_i\beta)$$

**We can easily write down the log likelihood as a function of $\beta = (\beta_1, \ldots, \beta_p)$,**

$$l(\beta) = -\sum_{i=1}^{n}\lambda_i + \sum_{i=1}^{n} y_i \log(\lambda_i) - \sum_{i=1}^{n}\log(y_i!)$$

**Note that**

$$\frac{d\lambda_i}{d\beta_j} = X_{ij}\exp(X_i\beta) = X_{ij}\lambda_i.$$

**This means that we can write down the gradient of the log likelihood with respect to $\beta$ explicitly. The $j$th entry is**

$$[\nabla l]_j = -\sum_{i=1}^{n} X_{ij}\lambda_i + \sum_{i=1}^{n} y_i X_{ij} = \sum_{i=1}^{n}(y_i - \lambda_i)X_{ij},$$

**or in matrix notation**

$$\nabla l = X^T(y - \lambda).$$

**Differentiating the $j$the entry of the gradient with respect to $\beta_k$ gives the ($j$,$k$) entry of the hessian matrix of second partial derivatives**

$$[\nabla l]_{jk} = -X_{ik}X_{ik}\lambda_i,$$

or in matrix notation

$$\nabla^2 l = -X^T W X,$$

where $W$ is the diagonal matrix with $i$th entry $\lambda_i$.

We code the functions for the gradient and the hessian.

```r
grad_pois<-function(beta,X,y){
  lambda<-exp(X%*%beta)
  t(X)%*%(y-lambda)
}
hess_pois<-function(beta,X,y){
    lambda<-as.vector(exp(X%*%beta))
    -t(X)%*%diag(lambda)%*%X
}
```

We illustrate the code for Newtons algorithm with fitting $log(\mu_i) = \beta_0 + \beta_1 Age_i + \beta_2 Shape_i$
We implement the stopping criterion to be a check if the change in the log-likelihood or coefficients is below the tolerance value, and if so, exit the loop. A vector of zeros is a sensible guess for the starting value of beta in a Poisson GLM because it assumes that there is no relationship between the predictors and the response variable. This can be a reasonable starting point in many cases, especially when the predictors are not strongly related to the response variable or when there is no prior knowledge about the expected relationship between the variables.

```r
#fit $log(\mu_i) = \beta_0 + \beta_1x_{1i} + \beta_2x_{2i}$
# Define initial values
Y <- Number
type <- as.integer(data$Shape == 'Cylinder')
X <- cbind(1,Age,type)
beta <- c(0,0,0)
beta_store <- matrix(nrow=25, ncol=3)
tolerance <- 1e-6
max_iterations <- 25
iter <- 1
# Iterate until convergence or maximum number of iterations reached
while(iter <= max_iterations) {
  # Store current beta values
  beta_store[iter,] <- beta
  # Compute gradient and Hessian
  grad <- grad_pois(beta,X,Y)
  hess <- hess_pois(beta,X,Y)
  # Compute change in coefficients and log-likelihood
  beta_change <- solve(hess, grad)
  llh_current <- sum(dpois(Y, exp(X %*% beta), log = TRUE))
  llh_new <- sum(dpois(Y, exp(X %*% (beta - beta_change)), log = TRUE))
  # Stop if change in log-likelihood or coefficients is below tolerance
  if(abs(llh_new - llh_current) < tolerance || max(abs(beta_change)) < tolerance) {
    break
  }
  # Update beta values and iteration counter
  beta <- beta - beta_change
  iter <- iter + 1
}
# View estimated coefficients
print(beta)
# View number of iterations to converge
print(iter)
```

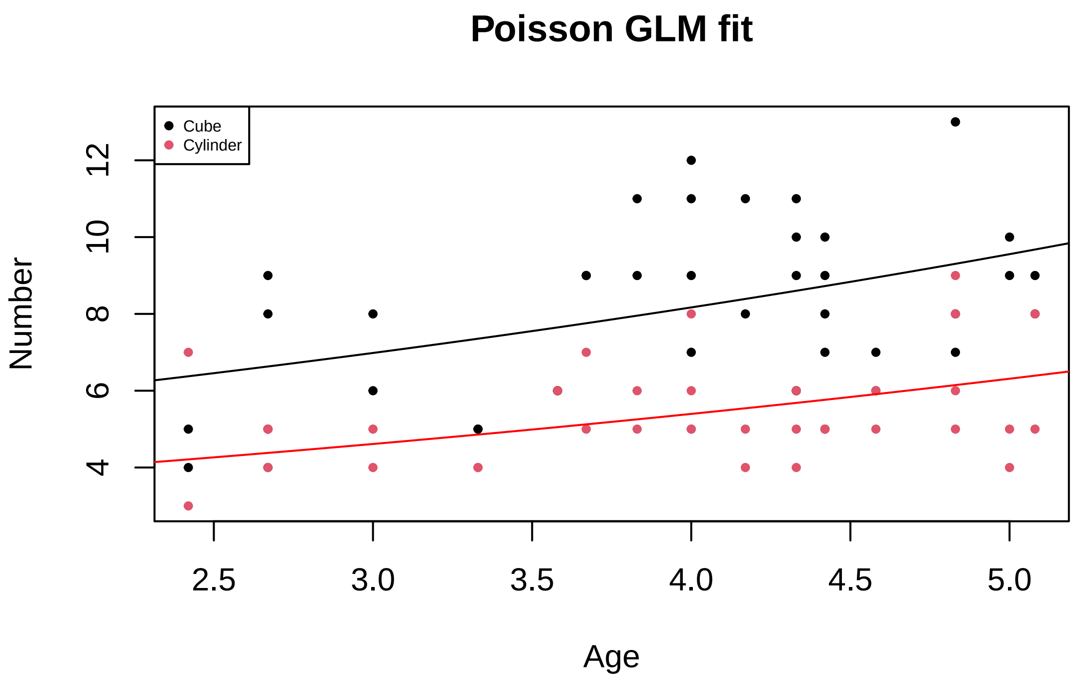
We now fit a few different poisson GLM with various linear predictors. We summarize their performances in the table.

| Linear Predictor | Deviance residual range | AIC |
|---|---|---|
| Age + Shape | 2.3557 | 334.09 |
| Age | 3.3509 | 354.94 |
| Age + Shape + Shape*Age | 2.3699 | 335.28 |
| Age + Age **2 + Shape + Shape | 2.3557 | 334.09 |

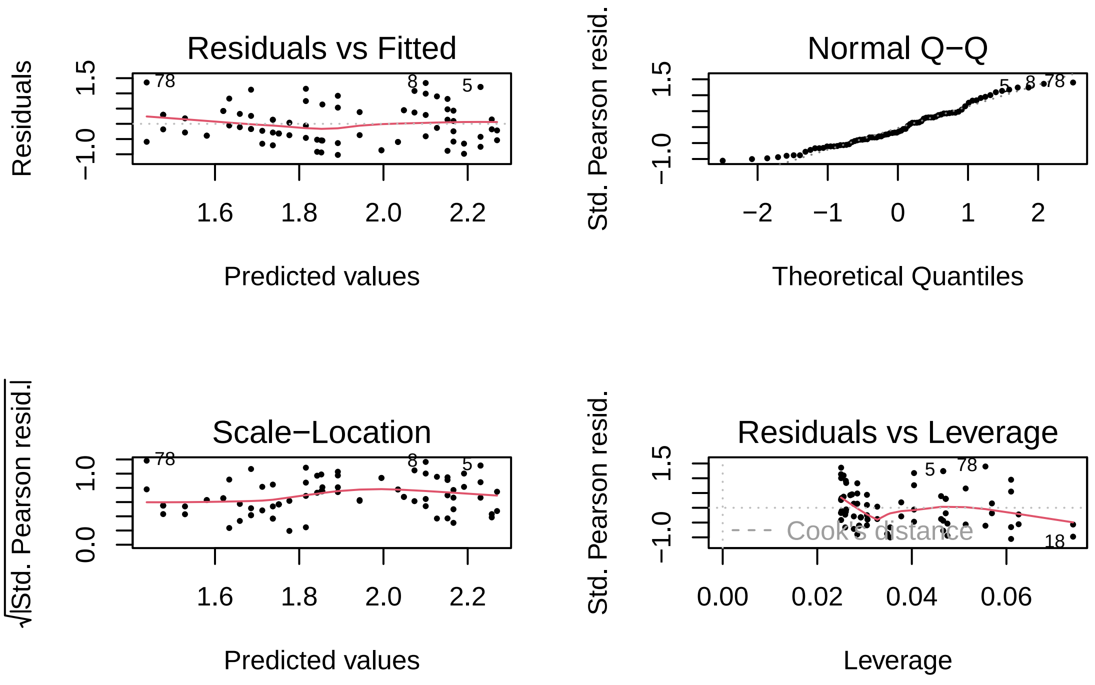
We observe that the two models have equally best performance. Namely the ones with linear predictors Age + Shape and Age + Age **2 + Shape. They have the smallest AIC score by a fairly large gap and are equal deviance residual range to the others to 1 decimal place. We choose to proceed with the Age + Shape linear predictor solely by reasoning of Occam's Razor. We find the model parameters and their standard errors. (we use glm in R since we have shown that we can implement the model from first principles)

| | beta0 | beta1 | beta2 |
|---|---|---|---|
| Value | 1.47292 | 0.15684 | -0.41468 |
| SE | 0.23641 | 0.05674 | 0.08768 |

We plot the new model and observe a slightly better looking fit to the data. We also see that there is no longer an assumption that the rate of improvement of stacking objects with age should not differ between shapes and that there should be a constant difference in the value of number of shapes stacked between the two shapes.



We can check the diagnostic plots for the new model and decide if this model is more appropriate for the data than the original linear model.



We observe that the residual plot also shows a good even flat line of best fit around 0 as with the linear model but the size of the residuals is far less now. We observe a slightly less heavy tailing in the Q-Q plot with marginally more equal looking empirical and theoretical quantiles. The scale-plot shows almost no correlation between mean and standard variance. This is much better than the linear model as we would expect this line to be flat since then the modelling assumption of constant variance would hold. Lastly, we see that all points have relatively small Cook's distance, implying small leverages and residuals and display a similar pattern to the linear model. From these observations we notice an improvement in performance from the linear model and so would prefer to use this general linear model.

## 1.5 Hypothesis testing

We would now like to determine whether there is evidence that children find one of the two shapes of block easier to stack. We use the anova() function to compare the fit of two Poisson GLM models, Y ~ Age and Y ~ Age + Shape, to determine if there is evidence that children find one of two shapes of blocks easier to stack.

**The anova() function is being used with the test = "Chisq" argument to perform a likelihood ratio test (LRT) between the two models. The LRT compares the fit of the two models by comparing their log-likelihoods. If the difference in the log-likelihoods between the two models is large enough, it indicates that one model is a significantly better fit to the data than the other.**

**In this case, we can assume that Y ~ Age represents a Poisson GLM model that assumes that the probability of a child successfully stacking a block does not depend on the shape of the block. On the other hand, Y ~ Age + Shape represents a Poisson GLM model that assumes that the probability of a child successfully stacking a block does depend on the shape of the block.**

**By comparing the fit of these two models using an LRT, we can test the hypothesis that the shape of the block does not affect the probability of a child successfully stacking it. If the LRT indicates that myglm2 provides a significantly better fit to the data than myglm1, we can conclude that there is evidence that children find one of the two shapes of block easier to stack.**

**Essentially we are doing a LRT with** $H_0 : \beta_2 = 0, H_1 : \beta_2 \neq 0$

```
## Analysis of Deviance Table
##
## Model 1: Y ~ Age
## Model 2: Y ~ Age + Shape
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        78     52.971
## 2        77     30.120  1   22.851 1.75e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

**When we perform a likelihood ratio test (LRT) using the anova() function with test = "Chisq", we obtain a p-value that represents the probability of observing a test statistic as extreme as the one we observed, assuming the null hypothesis is true.**

**In this case, the p-value is 1.75e-06. This means that if the null hypothesis were true (i.e., if the shape of blocks does not affect the probability of children successfully stacking them), the probability of observing a test statistic as extreme as the one we observed is very small (less than 0.001%).**

**If we set a significance level of 0.01 (which is a common choice), we would reject the null hypothesis if the p-value is less than 0.01. Since the p-value of 1.75e-06 is much smaller than 0.01, we would reject the null hypothesis and conclude that there is strong evidence that the shape of blocks does affect the probability of children successfully stacking them. In fact we would even reject the null hypothesis at a significance level of 0.00001.**

**In summary, a p-value of 1.75e-06 indicates very strong evidence against the null hypothesis, which suggests that the shape of blocks does affect the probability of children successfully stacking them.**

## 1.6 Predictions with Confidence Intervals

**For given covariates** $x_*$**, we have** $\hat{\eta_*}$ **with variance** $x_*^T(X^TWX)^{-1}x_*$**. Therefore, an approximate confidence interval can be constructed using the normal distribution. To obtain an approximate confidence interval in terms of** $\mu$ **we can use the inverse of the link function to transform the end points. As with the linear models, we can use the predict function in R for GLMs. More precisely, an approximate 95% confidence interval for the mean response of a prediction with covariates** $x_*$ **is**

$$((g^{-1}(\hat{\eta_*} - 1.96\sqrt{x_*^T(X^TWX)^{-1}x_*}), (g^{-1}(\hat{\eta_*} - 1.96\sqrt{x_*^T(X^TWX)^{-1}x_*}))$$

**where** $g^{-1}$ **is the inverse of the link function** $g$**. In our case we have** $g^{-1}(x) = e^x$**. The value 1.96 is the approximate 97.5th percentile of the standard normal distribution, which is used to construct a 95% confidence interval for a parameter estimate assuming the distribution is normal. We produce a table of predictions along with 95% confidence interval lower and upper bounds.**

| Age | Number Cylinder | Cylinder CI LB | Cylinder CI UB | Number Cube | Cube CI LB | Cube CI UB |
|---|---|---|---|---|---|---|
| 3 | 4.61 | 3.86 | 5.51 | 6.98 | 5.95 | 8.19 |
| 4 | 5.40 | 4.72 | 6.17 | 8.17 | 7.33 | 9.11 |
| 5 | 6.31 | 5.32 | 7.48 | 9.56 | 8.21 | 11.12 |

**Based on the table, it appears that as age increases, the mean number of blocks stacked for both the cylinder and cube shapes also increases. For example, the mean number of cylinder blocks stacked increases from 4.61 at age 3 to 6.31 at age 5, and the mean number of cube blocks stacked increases from 6.98 at age 3 to 9.56 at age 5. The confidence intervals for these means also suggest that we can be reasonably certain that the true mean number of blocks stacked falls within the stated range. For example, the 95% confidence interval for the mean number of cylinder blocks stacked at age 3 is 3.86 to 5.51, suggesting that the true mean is likely to be within this range with 95% confidence. Overall, the results suggest that stacking skills improve with age, and that children may find it easier to stack cube-shaped blocks compared to cylinder-shaped blocks.**

## 2. Summary and Conclusion

**We have shown that the Poisson GLM is a more appropriate model for the data and problem at hand than the initial linear model in section 1.4 We observed very good performance of the model on the data in the diagnostic plots. However, there are some limitations of the Poisson GLM that we should be aware of:**

1. **Non-independent observations: It is possible that the stacking performance of children within each age group may be correlated due to similarities in developmental stages or learning environments, which violates the assumption of independent observations in the Poisson GLM.**
2. **Lack of control for confounding variables: The study did not control for potential confounding variables such as motor skills, attention span, or previous experience with stacking tasks, which may impact the stacking performance of children and could lead to biased estimates in the Poisson GLM.**
3. **Non-linearity in the relationship between covariates and response variable: The Poisson GLM assumes a linear relationship between the log of the mean response and the covariates. However, there may be non-linear relationships between the covariates and the response variable, which would require alternative models such as polynomial regression or generalized additive models.**
4. **Overdispersion: There may be overdispersion in the data due to unobserved heterogeneity or variability in the responses, which violates the assumption of equal mean and variance in the Poisson distribution. This can be addressed using alternative models such as Negative Binomial or quasi-Poisson models.**

**We can draw the following conclusions in the context of the problem:**

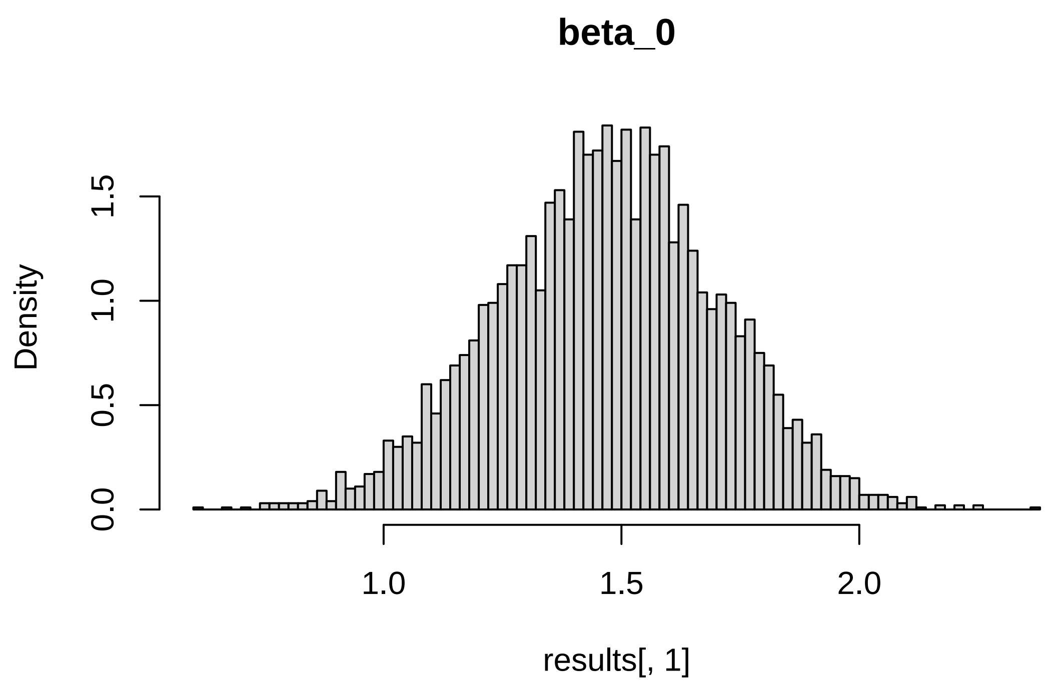
1. **There is a significant positive association between the number of blocks stacked and the age of the child, which suggests that as children grow older, they tend to stack more blocks.**
2. **There is evidence that the shape of the block has an impact on the number of blocks stacked. Specifically, the cube shape appears to be easier to stack compared to the cylinder shape, as evidenced by the significant negative coefficient of the "Shape Cylinder" predictor in the Poisson GLM.**

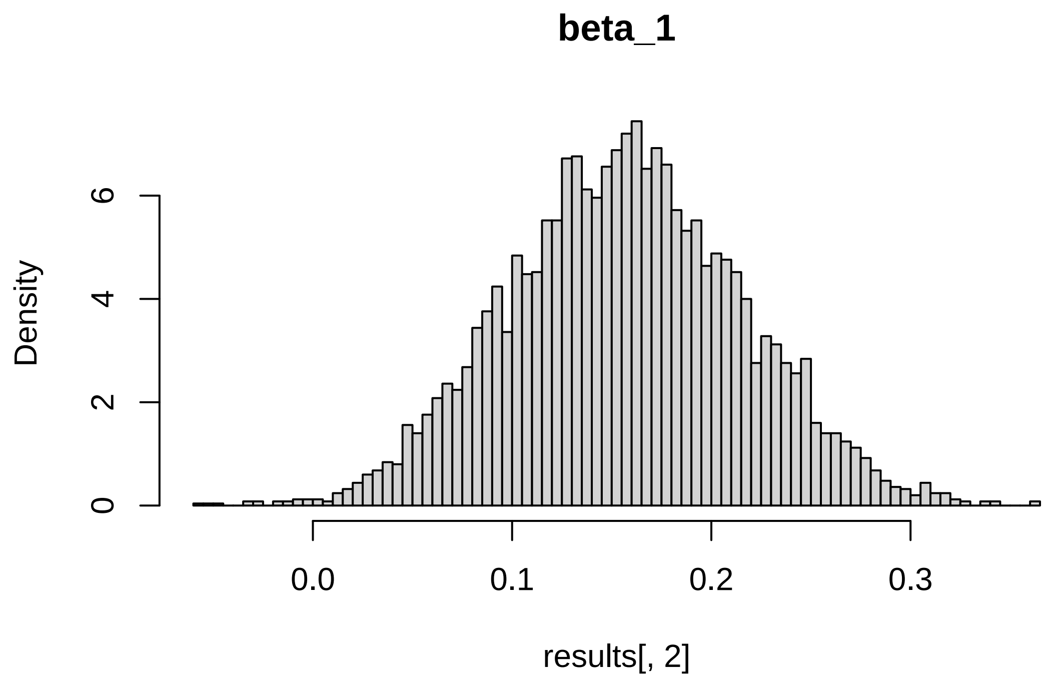
**Mastery material**

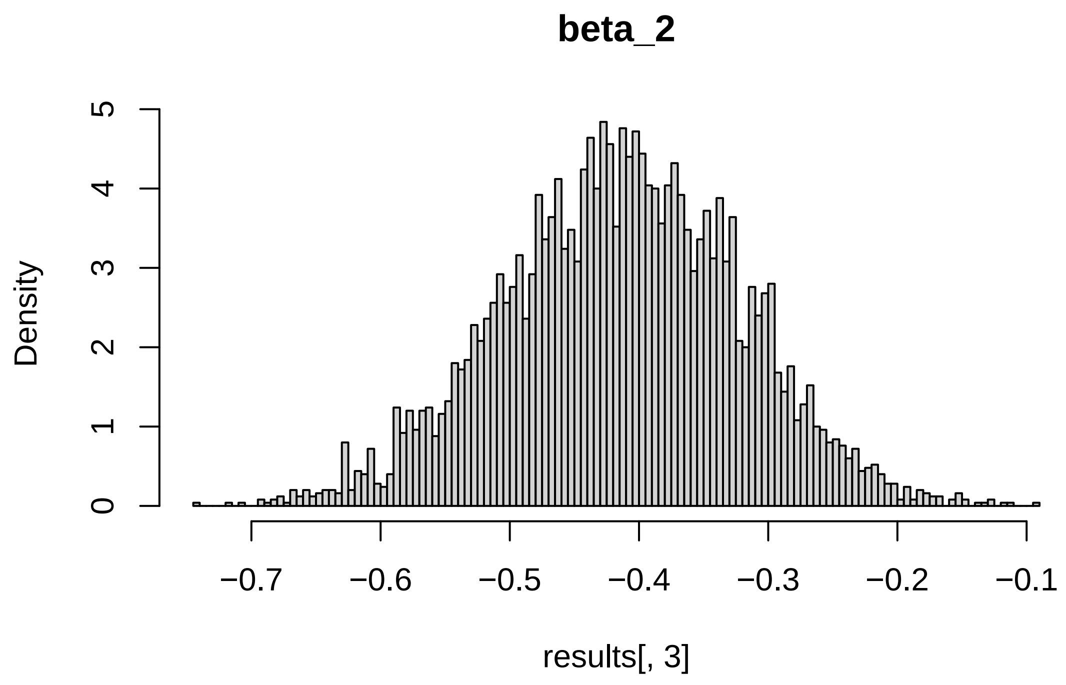
**The commonly used asymptotic approximations for the sampling distribution of the maximum likelihood estimators and scaled deviance in the Poisson GLM are based on the central limit theorem.**

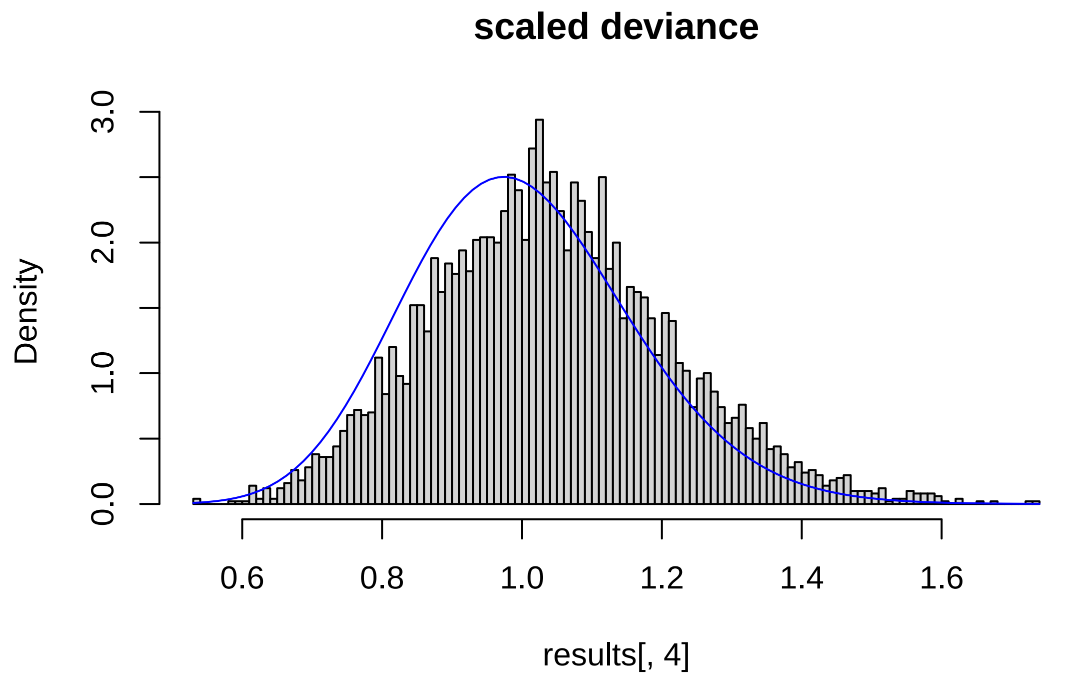
**According to the central limit theorem, under certain regularity conditions, the distribution of the maximum likelihood estimator (MLE) of a parameter approaches a normal distribution with mean equal to the true parameter value and variance equal to the inverse of the Fisher information. Similarly, the scaled deviance statistic, which is used to test the goodness of fit of the model, approaches a chi-squared distribution with degrees of freedom equal to the difference between the number of observations and the number of estimated parameters.**

**We set the 'true beta' to the beta in myglm1 and the the true scaled deviance to that of myglm1 also. Every simulation we create random Age and Shape data of size 80 as in the dataset. We find the mean using the predict() with the random data and the 'true model'. We then fit a poisson glm with predictor as a random poisson sample from the means we obtain and covariates as this simulations Age and Shape data. Each simulation we exctract this fit's MLE beta and scaled deviance values. What we hope to see is over a large number of simulations the histograms of the beta values and the scaled deviance values would match those as the theoretical patterns we expect from our understanding of asymptotic distributions of beta and the scaled deviance statistic.**









**We observe the correct patterns. For the beta coefficients, all three histograms look normally distributed and are clearly centred at the true coefficient values (1.4729174,0.1568450,-0.4146818). We calculate the inverse fisher information and the variances of the beta coefficient simulations.**

| | $\beta_0$ | $\beta_1$ | $\beta_2$ |
|---|---|---|---|
| **Inverse of fisher information** | **0.055891952** | **0.003219592** | **0.007687733** |
| **Simulation Variance** | **0.053920396** | **0.003424756** | **0.008212600** |

**For the scaled deviance we see the simulation histogram approaches a chi-squared distribution with degrees of freedom equal to the difference between the number of observations and the number of estimated parameters, namely 80-3=77. We may conclude that these are suitable asymptotic approximations for the sampling distribution of the maximum likelihood estimators and scaled deviance in this problem.**

Appendix

```
#load the data
data <- read.csv("/Users/nikolaikrokhin/Downloads/1724711.csv")
head(data)
```

```
##   Child Number Shape  Age
## 1     B      9  Cube 5.00
## 2     C      8  Cube 4.42
## 3     D      9  Cube 4.33
## 4     E     10  Cube 4.33
## 5     F     13  Cube 4.83
## 6     G     10  Cube 4.42
```

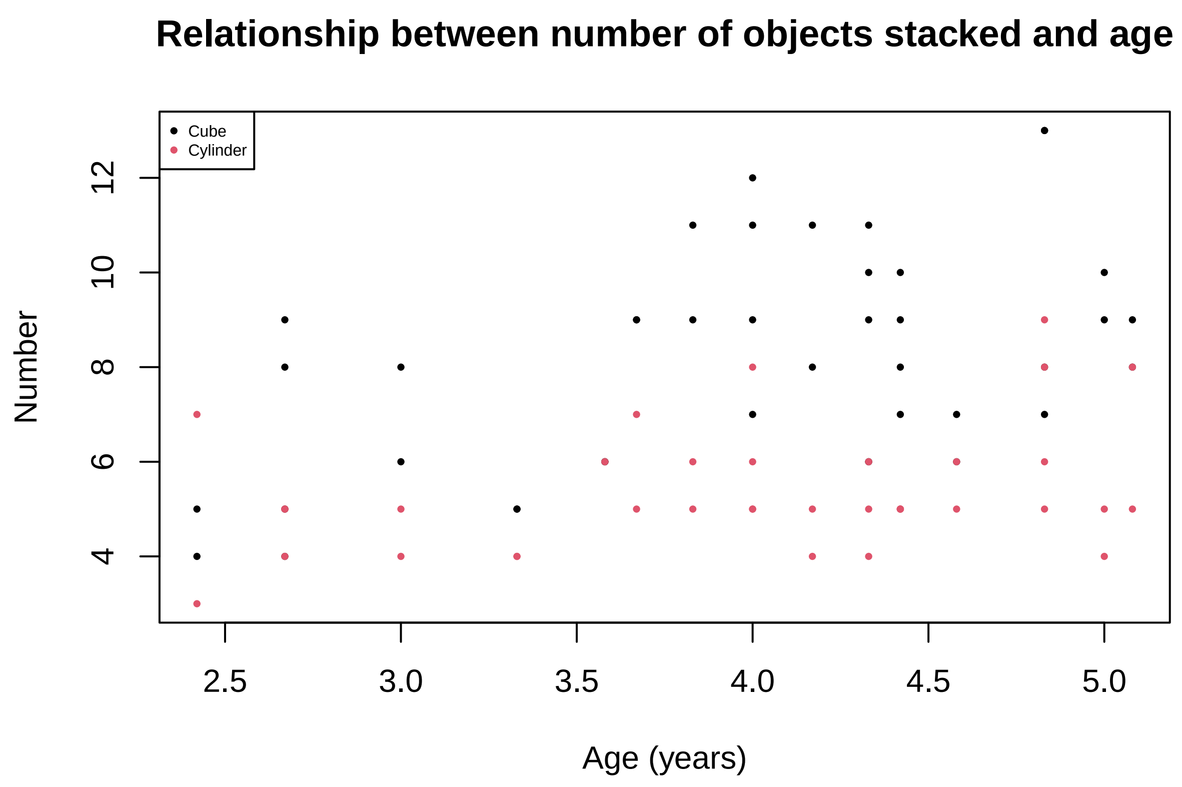
```
#provide summary
summary(data)
```

```
##     Child               Number          Shape                Age
##  Length:80          Min.   : 3.000   Length:80          Min.   :2.420
##  Class :character   1st Qu.: 5.000   Class :character   1st Qu.:3.518
##  Mode  :character   Median : 6.000   Mode  :character   Median :4.085
##                     Mean   : 6.787                      Mean   :3.958
##                     3rd Qu.: 8.250                      3rd Qu.:4.460
##                     Max.   :13.000                      Max.   :5.080
```

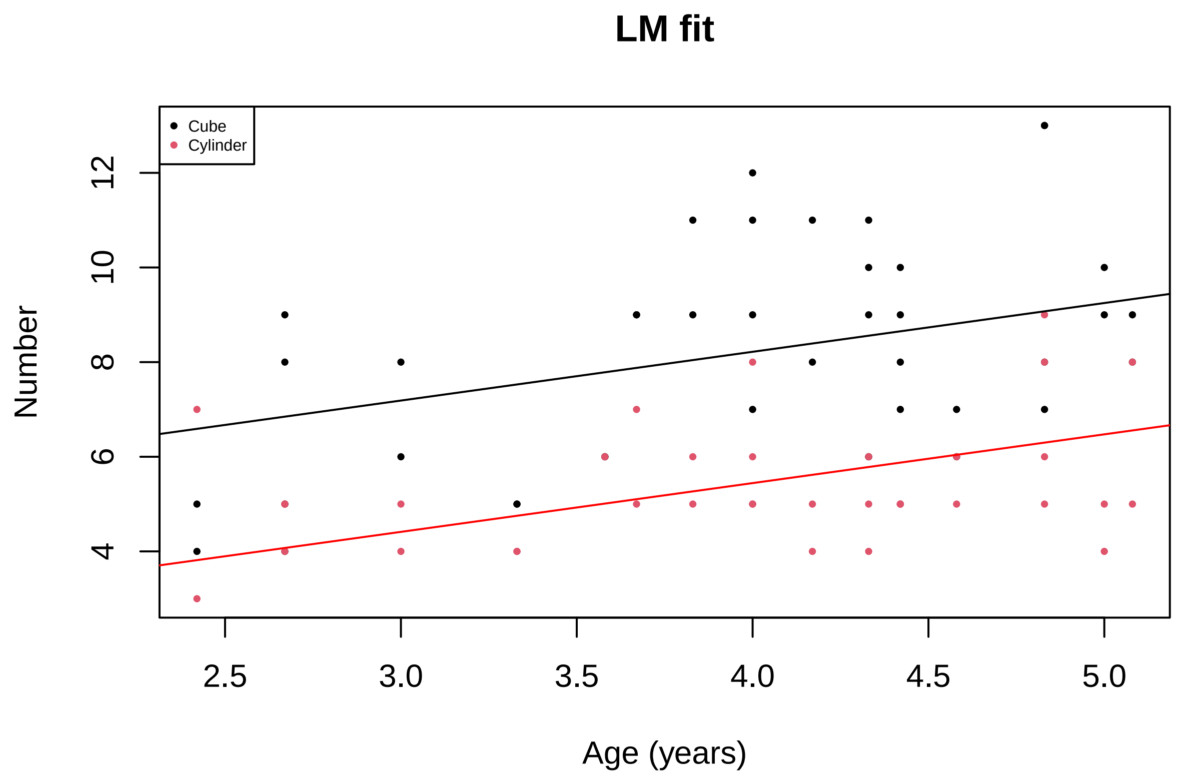
```
data$Shape <- as.factor(data$Shape)
summary(data)
```

```
##     Child               Number            Shape         Age
##  Length:80          Min.   : 3.000   Cube    :40   Min.   :2.420
##  Class :character   1st Qu.: 5.000   Cylinder:40   1st Qu.:3.518
##  Mode  :character   Median : 6.000                 Median :4.085
##                     Mean   : 6.787                 Mean   :3.958
##                     3rd Qu.: 8.250                 3rd Qu.:4.460
##                     Max.   :13.000                 Max.   :5.080
```

```
#exploratory plots
Number <- data$Number
Age <- data$Age
Shape <- data$Shape
plot(Age,Number,pch=16,col=factor(Shape), main="Relationship between number of objects stacked and age",xlab="Ag
legend("topleft",
       legend = levels(factor(Shape)),
       pch = 16,
       cex=.5,
       col = factor(levels(factor(Shape))))
```



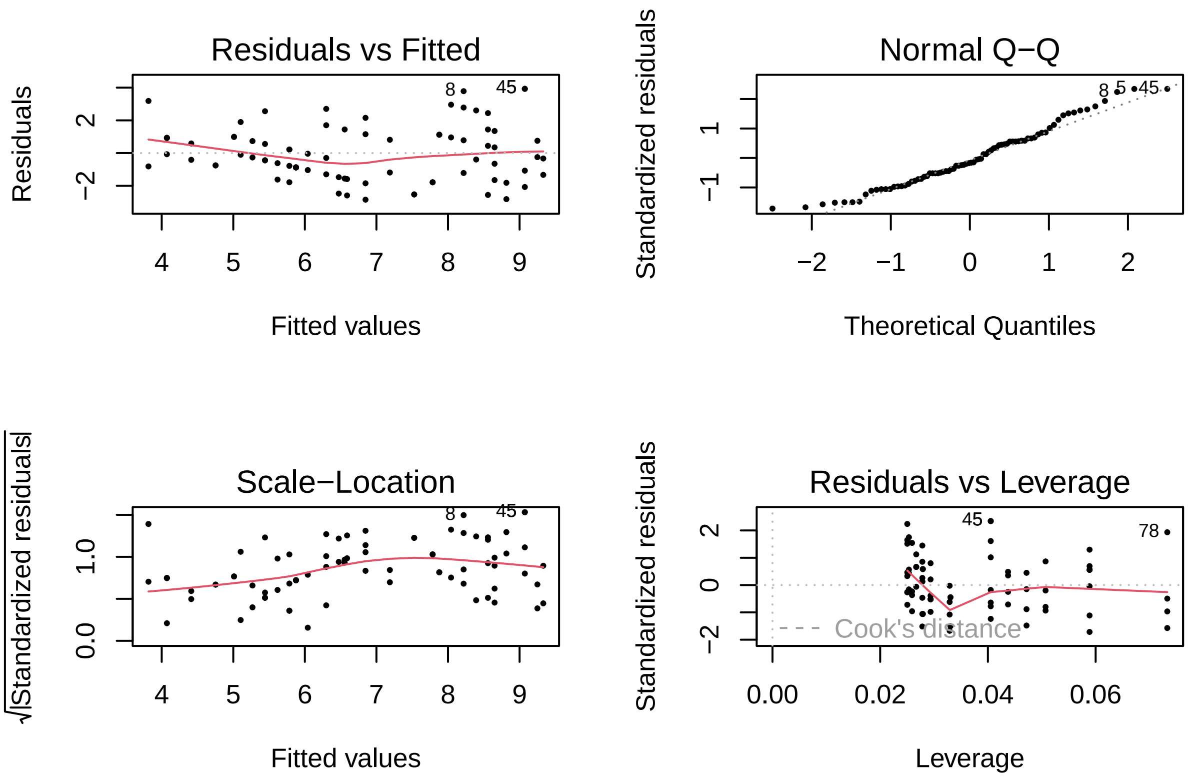
```
plot(Age,Number,pch=16,col=factor(Shape), main="LM fit", xlab='Age (years)',cex=.5)
legend("topleft",
       legend = levels(factor(Shape)),
       pch = 16,
       cex=.5,
       col = factor(levels(factor(Shape))))
abline(betahat[1]+1*betahat[3],betahat[2],,col='red')
abline(betahat[1]+0*betahat[3],betahat[2])
```



```
#fit lm to check what it should be
Shape <- data$Shape
mylm <- lm(Number~Age+Shape,data=data)
summary(mylm)
```

```
##
## Call:
## lm(formula = Number ~ Age + Shape, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8476 -1.2384 -0.2834  0.9896  3.9263
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     4.0960     1.0047   4.077  0.00011 ***
## Age             1.0306     0.2445   4.215 6.72e-05 ***
## ShapeCylinder  -2.7750     0.3825  -7.254 2.74e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.711 on 77 degrees of freedom
## Multiple R-squared:  0.4776, Adjusted R-squared:  0.464
## F-statistic:  35.2 on 2 and 77 DF,  p-value: 1.392e-11
```

```
par(mfrow=c(2,2))
plot(mylm,pch=16,cex=.5)
```



```
grad_pois<-function(beta,X,y){
  lambda<-exp(X%*%beta)
  t(X)%*%(y-lambda)
}

hess_pois<-function(beta,X,y){
    lambda<-as.vector(exp(X%*%beta))
    -t(X)%*%diag(lambda)%*%X
}
```

```
#fit $log(\mu_i) = \beta_0 + \beta_1x_{1i} + \beta_2x_{2i}$

# Define initial values
Y <- Number
type <- as.integer(data$Shape == 'Cylinder')
X <- cbind(1,Age,type)
beta <- c(0,0,0)
beta_store <- matrix(nrow=25, ncol=3)
tolerance <- 1e-6
max_iterations <- 25
iter <- 1

# Iterate until convergence or maximum number of iterations reached
while(iter <= max_iterations) {
  # Store current beta values
  beta_store[iter,] <- beta

  # Compute gradient and Hessian
  grad <- grad_pois(beta,X,Y)
  hess <- hess_pois(beta,X,Y)

  # Compute change in coefficients and log-likelihood
  beta_change <- solve(hess, grad)
  llh_current <- sum(dpois(Y, exp(X %*% beta), log = TRUE))
  llh_new <- sum(dpois(Y, exp(X %*% (beta - beta_change)), log = TRUE))

  # Stop if change in log-likelihood or coefficients is below tolerance
  if(abs(llh_new - llh_current) < tolerance || max(abs(beta_change)) < tolerance) {
    break
  }

  # Update beta values and iteration counter
  beta <- beta - beta_change
  iter <- iter + 1
}

# View estimated coefficients
print(beta)
```

```
##            [,1]
##       1.4729173
## Age   0.1568450
## type -0.4146819
```

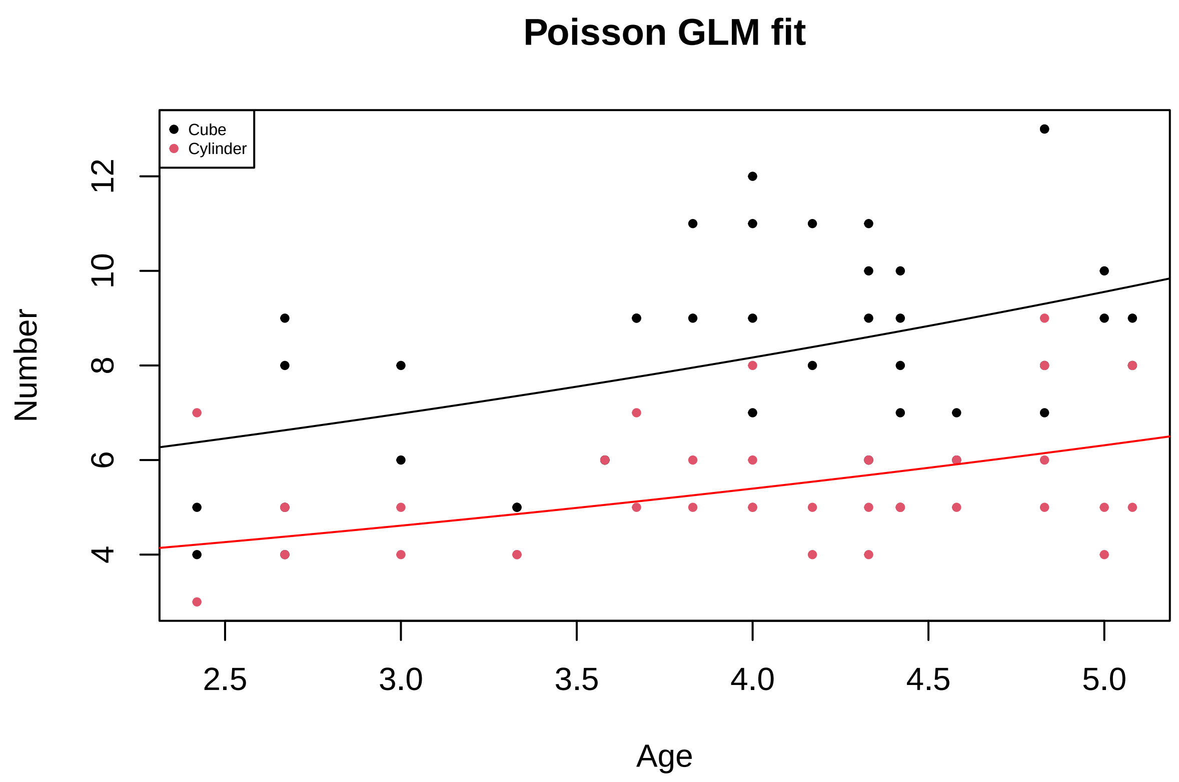
```
# View number of iterations to converge
print(iter)
```

```
## [1] 11
```

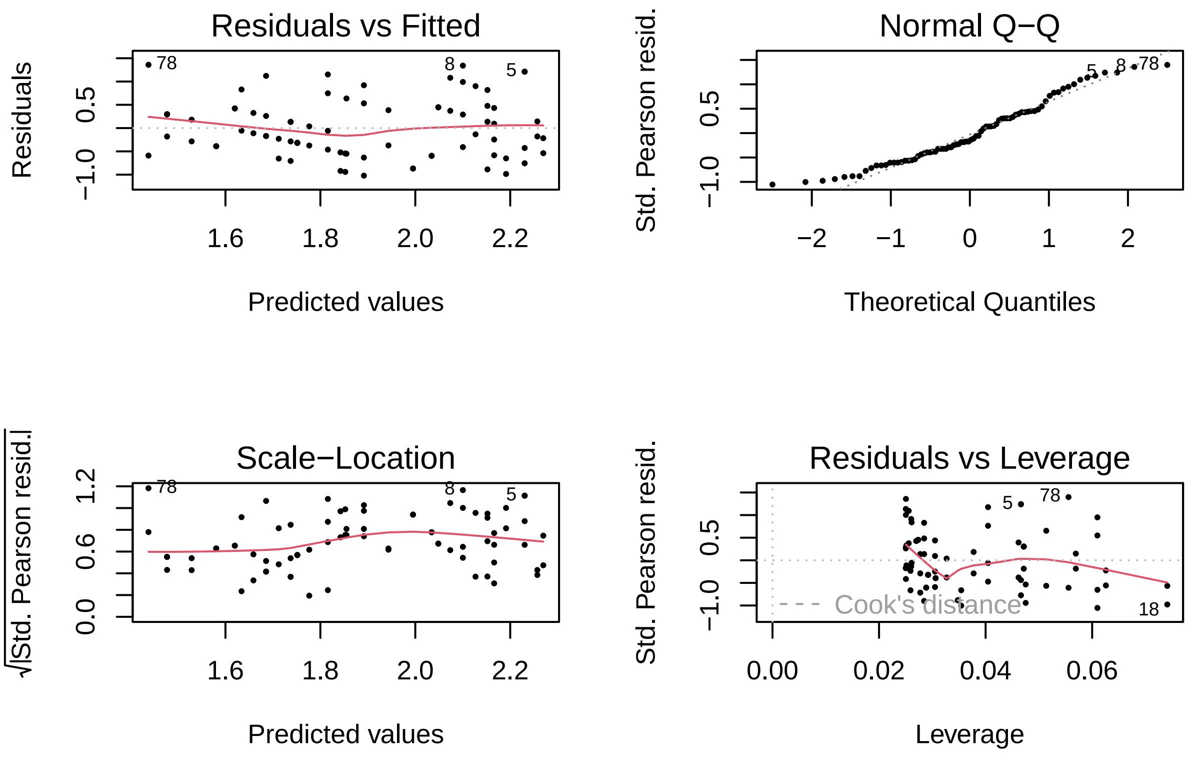
```
#verification of correct implementation
myglm1<-glm(Y ~ Age + Shape, family = "poisson")
summary(myglm1)
```

```
##
## Call:
## glm(formula = Y ~ Age + Shape, family = "poisson")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1036  -0.4939  -0.1540   0.4096   1.2521
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.47292    0.23641   6.230 4.66e-10 ***
## Age            0.15684    0.05674   2.764  0.00571 **
## ShapeCylinder -0.41468    0.08768  -4.730 2.25e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 60.797  on 79  degrees of freedom
## Residual deviance: 30.120  on 77  degrees of freedom
## AIC: 334.09
##
## Number of Fisher Scoring iterations: 4
```

```
#plot glm fit
plot(Age,Number,pch=19,col=factor(Shape), main="Poisson GLM fit",cex=.5)
legend("topleft",
       legend = levels(factor(Shape)),
       pch = 19,
       cex=.5,
       col = factor(levels(factor(Shape))))
xs = seq(2,6,by=0.1)
lines(x = xs, y = exp(beta[1]+beta[2]*xs+1*beta[3]),col='red')
lines(x = xs, y = exp(beta[1]+beta[2]*xs+0*beta[3]))
```



```
#verification of correct implementation
myglm1<-glm(Y ~ Age + Shape, family = "poisson")
par(mfrow=c(2,2))
plot(myglm1,pch=16,cex=.5)
```



```
#fit $log(\mu_i) = \beta_0 + \beta_1x_{1i}$
myglm2<-glm(Y ~ Age, family = "poisson")
summary(myglm2)
```

```
##
## Call:
## glm(formula = Y ~ Age, family = "poisson")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5456  -0.6747  -0.1887   0.4980   1.8053
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.28692    0.23383   5.504 3.72e-08 ***
## Age          0.15684    0.05674   2.764  0.00571 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 60.797  on 79  degrees of freedom
## Residual deviance: 52.971  on 78  degrees of freedom
## AIC: 354.94
##
## Number of Fisher Scoring iterations: 4
```

```
#fit $log(\mu_i) = \beta_0 + \beta_1x_{2i}$
myglm3<-glm(Y ~ Shape, family = "poisson")
summary(myglm3)
```

```
##
```

```
## Call:
## glm(formula = Y ~ Shape, family = "poisson")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6222  -0.4214  -0.1743   0.2839   1.5526
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    2.10108    0.05530   37.99  < 2e-16 ***
## ShapeCylinder -0.41468    0.08768   -4.73 2.25e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 60.797  on 79  degrees of freedom
## Residual deviance: 37.946  on 78  degrees of freedom
## AIC: 339.92
##
## Number of Fisher Scoring iterations: 4
```

```
#fit $log(\mu_i) = \beta_0 + \beta_1x_{1i}+\beta_2x_{2i} \beta_3x_{1i}x_{2i}$
myglm4<-glm(Y ~ Age + Shape + Shape*Age, family = "poisson")
summary(myglm4)
```

```
##
## Call:
## glm(formula = Y ~ Age + Shape + Shape * Age, family = "poisson")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1097  -0.4448  -0.1628   0.3801   1.2602
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)         1.302134   0.306052   4.255 2.09e-05 ***
## Age                 0.198880   0.073873   2.692   0.0071 **
## ShapeCylinder       0.006493   0.474435   0.014   0.9891
## Age:ShapeCylinder -0.104121   0.115425  -0.902   0.3670
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 60.797  on 79  degrees of freedom
## Residual deviance: 29.309  on 76  degrees of freedom
## AIC: 335.28
##
## Number of Fisher Scoring iterations: 4
```

```
#fit $log(\mu_i) = \beta_0 + \beta_1x_{1i}+\beta_2x_{1i} ** 2 + \beta_3x_{2i}$
myglm5<-glm(Y ~ Age + Age **2 + Shape, family = "poisson")
summary(myglm5)
```

```
##
## Call:
## glm(formula = Y ~ Age + Age^2 + Shape, family = "poisson")
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -1.1036  -0.4939  -0.1540   0.4096   1.2521
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     1.47292    0.23641   6.230 4.66e-10 ***
## Age             0.15684    0.05674   2.764  0.00571 **
## ShapeCylinder -0.41468    0.08768  -4.730 2.25e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 60.797  on 79  degrees of freedom
## Residual deviance: 30.120  on 77  degrees of freedom
## AIC: 334.09
##
## Number of Fisher Scoring iterations: 4
```

```
anova(myglm2,myglm1,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: Y ~ Age
## Model 2: Y ~ Age + Shape
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        78     52.971
## 2        77     30.120  1   22.851 1.75e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Age <- c(3,4,5,3,4,5)
Shape <- c("Cube", "Cube", "Cube", "Cylinder", "Cylinder", "Cylinder")
preds_x <- data.frame(Age,Shape)
preds <- predict(myglm1, newdata = preds_x, type = "link", se.fit = TRUE)
critval <- qnorm(0.975)
upr <- preds$fit + (critval * preds$se.fit)
lwr <- preds$fit - (critval * preds$se.fit)
fit <- preds$fit
Number_Cube <- exp(fit[c(1,2,3)])
Number_Cylinder <- exp(fit[c(4,5,6)])
Number_Cube_CI_LB <-exp(lwr[c(1,2,3)])
Number_Cylinder_CI_LB <-exp(lwr[c(4,5,6)])
Number_Cube_CI_UB <-exp(upr[c(1,2,3)])
Number_Cylinder_CI_UB <-exp(upr[c(4,5,6)])
Predictions <- data.frame("Age"=c(3,4,5),Number_Cylinder,Number_Cylinder_CI_LB,Number_Cylinder_CI_UB,Number_Cube
row.names(Predictions) <- NULL
print(Predictions)
```

```
##   Age Number_Cylinder Number_Cylinder_CI_LB Number_Cylinder_CI_UB Number_Cube
## 1   3        4.612503              3.863001              5.507423    6.982817
## 2   4        5.395772              4.721548              6.166273    8.168599
## 3   5        6.312051              5.324871              7.482244    9.555743
##   Number_Cube_CI_LB Number_Cube_CI_UB
## 1          5.953949          8.189477
## 2          7.328444          9.105071
## 3          8.214131         11.116481
```

```r
Number <- data$Number
Age <- data$Age
Shape <- data$Shape
myglm1 <- glm(Number ~ Age + Shape, data = data, family = poisson(link = "log"))

# Summarize the model
summary(myglm1)
```

```
##
## Call:
## glm(formula = Number ~ Age + Shape, family = poisson(link = "log"),
##     data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1036  -0.4939  -0.1540   0.4096   1.2521
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.47292    0.23641   6.230 4.66e-10 ***
## Age            0.15684    0.05674   2.764  0.00571 **
## ShapeCylinder -0.41468    0.08768  -4.730 2.25e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 60.797  on 79  degrees of freedom
## Residual deviance: 30.120  on 77  degrees of freedom
## AIC: 334.09
##
## Number of Fisher Scoring iterations: 4
```

```r
n_sim <- 5000
n <- 80
age_range <- c(min(Age), max(Age))
shape_values <- c("Cube", "Cylinder")

sim_and_fit <- function(Age, Shape){
  newdata <- data.frame(Age, Shape)
  means <- predict(myglm1, newdata = newdata, type = "response")
  y <- rpois(n, means)

  fit <- glm(y ~ Age + Shape, data = newdata, family = poisson(link = "log"))

  return(list(as.numeric(coef(fit)[1]), as.numeric(coef(fit)[2]), as.numeric(coef(fit)[3]), as.numeric(deviance(
}

results <- matrix(NA, nrow = n_sim, ncol = 4)

for(i in 1:n_sim){
  Age_i <- runif(n, age_range[1], age_range[2])
  Shape_i <- sample(shape_values, n, replace=TRUE)

  result <- sim_and_fit(Age_i, Shape_i)

  results[i, ] <- unlist(result)
}

summary(results)
```

```
##        V1               V2                 V3                V4
##  Min.   :0.4515   Min.   :-0.04577   Min.   :-0.7717   Min.   :0.4510
##  1st Qu.:1.3212   1st Qu.: 0.11601   1st Qu.:-0.4761   1st Qu.:0.9175
##  Median :1.4778   Median : 0.15562   Median :-0.4157   Median :1.0277
##  Mean   :1.4748   Mean   : 0.15585   Mean   :-0.4169   Mean   :1.0358
##  3rd Qu.:1.6301   3rd Qu.: 0.19539   3rd Qu.:-0.3564   3rd Qu.:1.1429
##  Max.   :2.2881   Max.   : 0.43691   Max.   :-0.1043   Max.   :1.7459
```

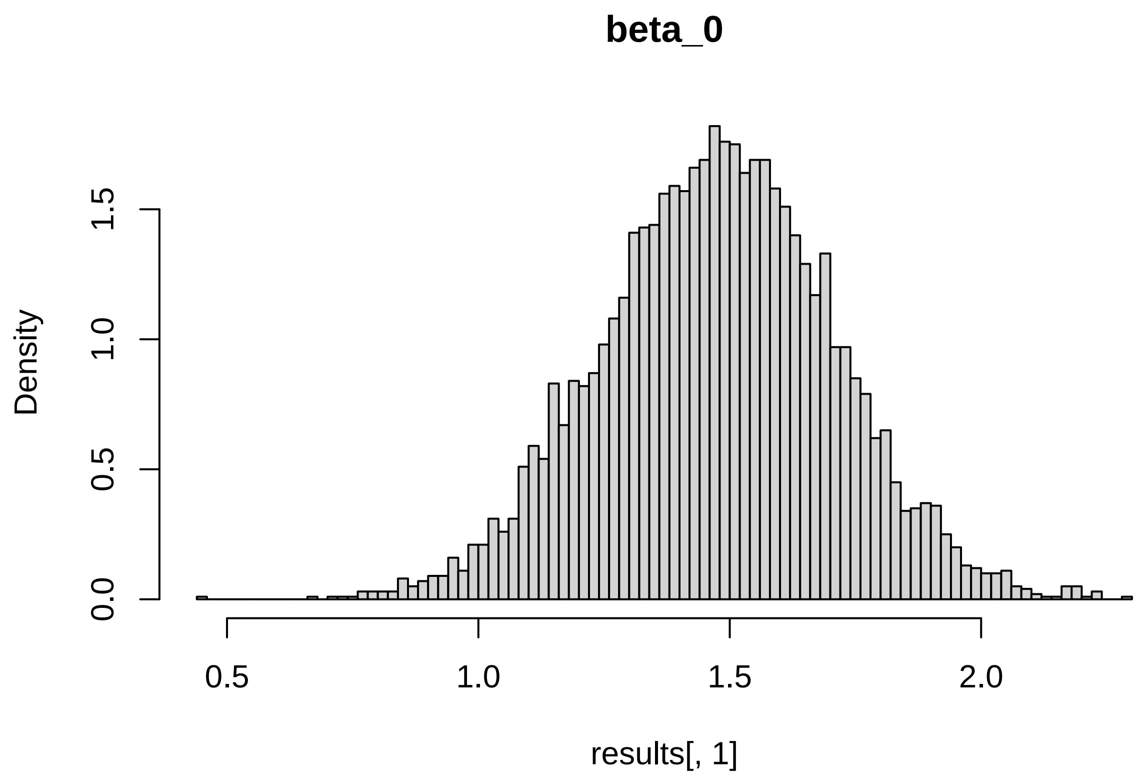
```
coef(myglm1)
```

```
##   (Intercept)           Age ShapeCylinder
##     1.4729174     0.1568450    -0.4146818
```

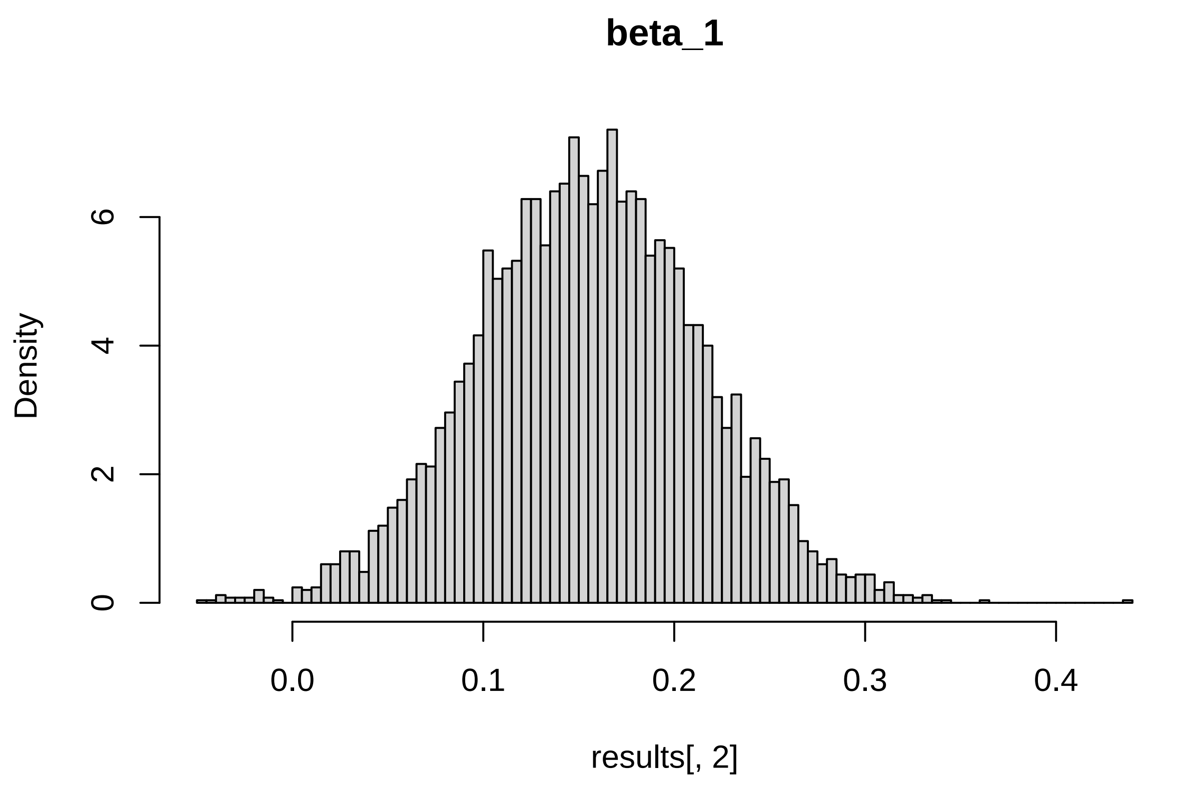
```
deviance(myglm1)/df.residual(myglm1)
```

```
## [1] 0.3911699
```

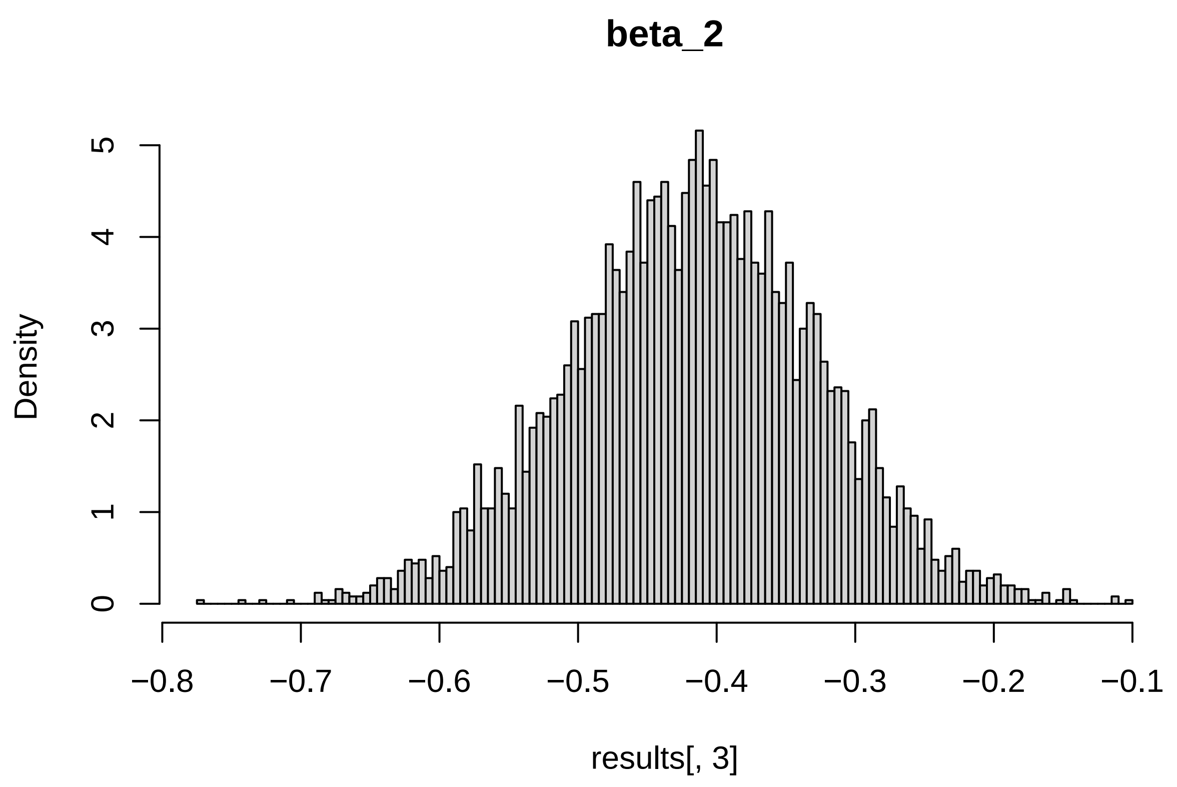
```
hist(results[, 1], breaks=100, freq = FALSE, main = "beta_0")
```



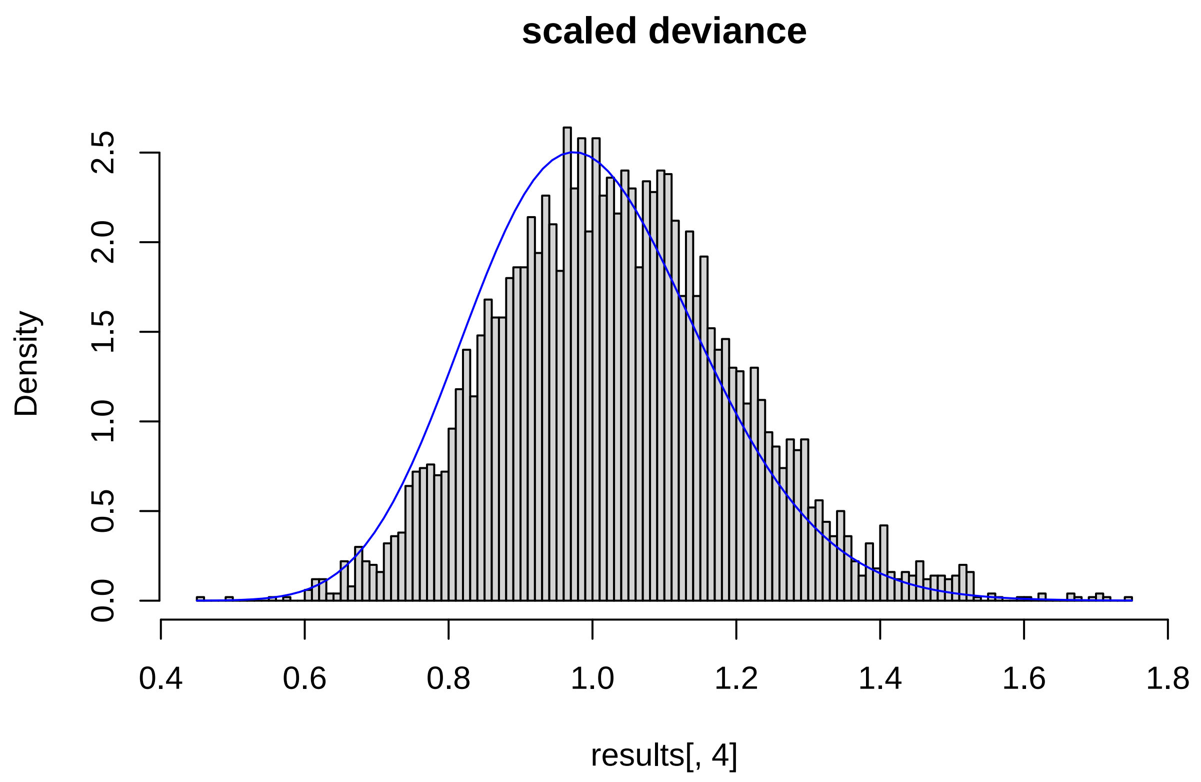
```r
hist(results[, 2], breaks=100, freq = FALSE, main = "beta_1")
```



```r
hist(results[, 3], breaks=100, freq = FALSE, main = "beta_2")
```



```
hist(results[, 4], breaks=100, freq = FALSE, main = "scaled deviance")
curve(77*dchisq(x*77, df = 77),add=TRUE,col='blue')
```



```
diag(var(results))
```

```
## [1] 0.053584822 0.003452901 0.008001573 0.028583663
```

```
diag(vcov(myglm1))
```

```
##   (Intercept)           Age ShapeCylinder
##   0.055891952   0.003219592   0.007687733
```