7 Generalized Linear Models (GLM)

7.1 When the response is not normal

For the linear we assumed

$$y_i|x_i \sim N(\beta_0 + \beta_i x_i, \sigma)$$

However, $y_i|x_i$ may not follow a normal distribution. For this cases, we need a Generalized Linear Model (GLM).

The responds could also follow one of the following distributions:

- Normal (continuous)
- Poisson or negative binomial (count data)
- Gamma (positive continuous)
- Bernoulli (binary data)

7.1.1 Example for count data

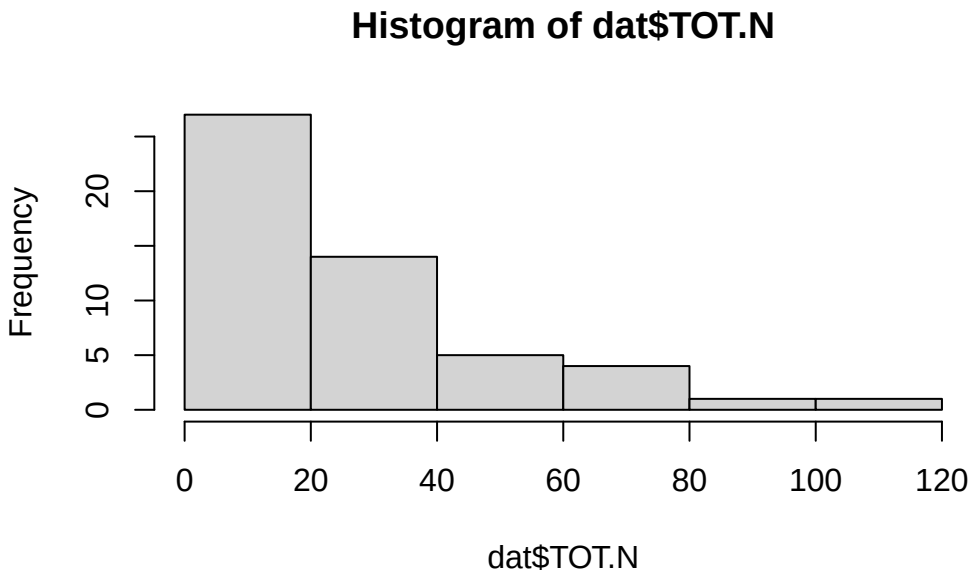
Amphibian road kills

```
dat <- read.table(here::here("data/RoadKills.txt"), header = TRUE)
head(dat)
```

```
  Sector      X      Y BufoCalamita TOT.N S.RICH OPEN.L  OLIVE MONT.S   MONT
1      1 260181 256546            5    22      3 22.684 60.333  0.000  0.653
2      2 259914 256124            1    14      4 24.657 40.832  0.000  0.161
3      3 259672 255688           40    65      6 30.121 23.710  0.258 10.918
4      4 259454 255238           27    55      5 50.277 14.940  1.783 26.454
5      5 259307 254763           67    88      4 43.609 35.353  2.431 11.330
6      6 259189 254277           56   104      7 31.385 17.666  0.000 43.678
  POLIC SHRUB  URBAN WAT.RES L.WAT.C L.D.ROAD L.P.ROAD D.WAT.RES D.WAT.COUR
1 4.811 0.406  7.787   0.043   0.583 3330.189    1.975   252.113    735.000
2 2.224 0.735 27.150   0.182   1.419 2587.498    1.761   139.573    134.052
```

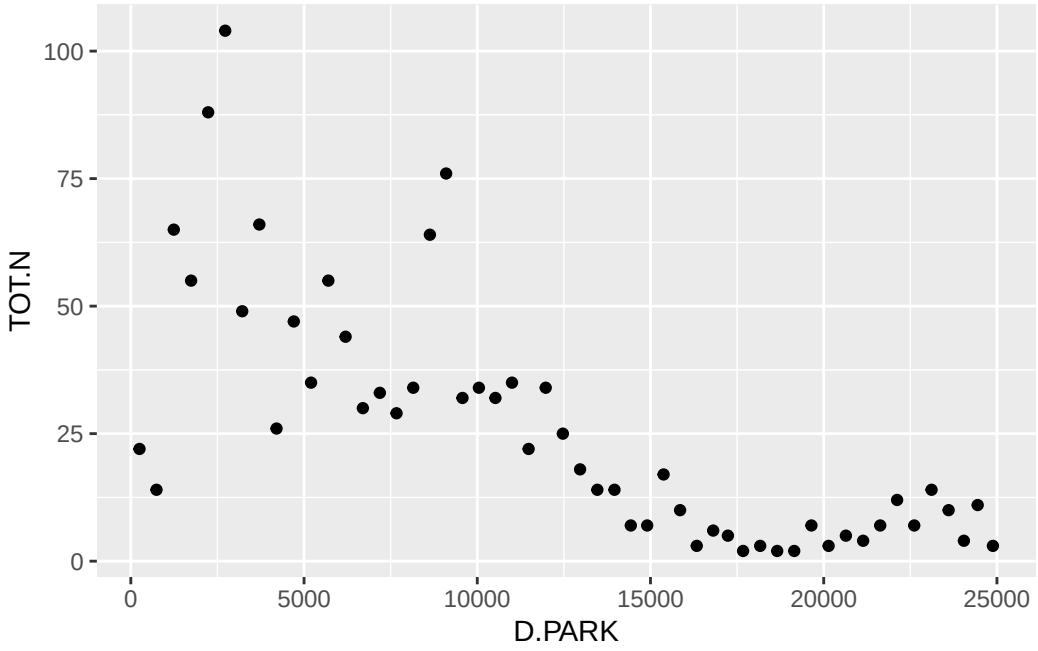
```
3 1.946 0.474 28.086   0.453   2.005 2149.651    1.250    59.168    269.029
4 0.625 0.607  0.831   0.026   1.924 4222.983    0.666   277.842     48.751
5 0.791 0.173  2.452   0.000   2.167 2219.302    0.653   967.808    126.102
6 0.054 0.325  2.730   0.039   2.391 1005.629    1.309   560.000    344.444
    D.PARK N.PATCH  P.EDGE L.SDI
1  250.214     122 553.936 1.801
2  741.179      96 457.142 1.886
3 1240.080      67 432.360 1.930
4 1739.885      63 421.292 1.865
5 2232.130      59 407.573 1.818
6 2724.089      49 420.289 1.799
```

```
hist(dat$TOT.N)
```



We want to model the response (`TOT.N`) as a Poisson distribution with rate λ that changes according to the distance to a park.

```
ggplot(dat, aes(D.PARK, TOT.N)) + geom_point()
```



7.1.2 Example for pressence absence data

Tuberculosis in wild boar, this can be seen as a Bernoulli random variable (a boar is either infected or not).

```
dat <- read.table(here::here("data/Boar.txt"), header = TRUE)
table(dat$Tb)
```

```
  0   1
314 285
```

```
ggplot(dat, aes(LengthCT, Tb)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm") +
  geom_smooth(method = "glm",
              method.args = list(family = "binomial"), col = "red")
```

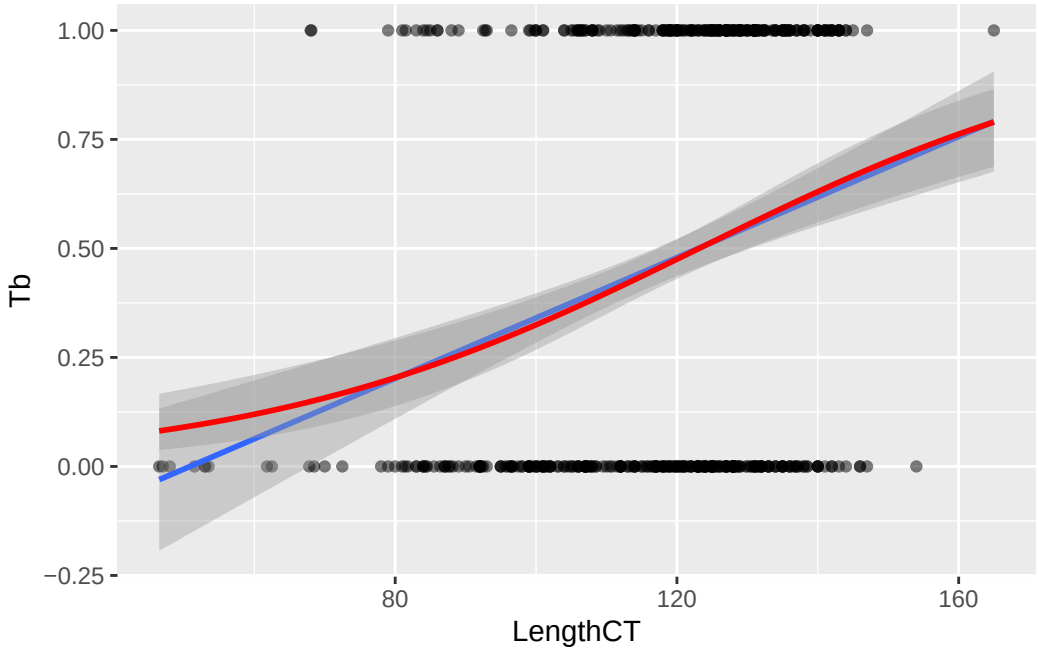
```
`geom_smooth()` using formula = 'y ~ x'
```

```
Warning: Removed 149 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
`geom_smooth()` using formula = 'y ~ x'
```

```
Warning: Removed 149 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 149 rows containing missing values or values outside the scale range
(`geom_point()`).
```



7.2 Structure of GLMs

A GLM consists of three components:

1. The distribution of the response.
2. Specification of the systematic part of the model (i.e., the linear predictor), usually denoted as η.
3. A link $g(\cdot)$ between the expectation of the response distribution and the systematic part of the model. This is also called the **link function**.

Some common link functions are:

Table 15.1 Some Common Link Functions and Their Inverses

Link	$\eta_i = g(\mu_i)$	$\mu_i = g^{-1}(\eta_i)$
Identity	μ_i	η_i
Log	$\log_e \mu_i$	e^{η_i}
Inverse	μ_i^{-1}	η_i^{-1}
Inverse-square	μ_i^{-2}	$\eta_i^{-1/2}$
Square-root	$\sqrt{\mu_i}$	η_i^2
Logit	$\log_e \frac{\mu_i}{1-\mu_i}$	$\frac{1}{1+e^{-\eta_i}}$
Probit	$\Phi^{-1}(\mu_i)$	$\Phi(\eta_i)$
Log-log	$-\log_e[-\log_e(\mu_i)]$	$\exp[-\exp(-\eta_i)]$
Complementary log-log	$\log_e[-\log_e(1-\mu_i)]$	$1-\exp[-\exp(\eta_i)]$

NOTE: μ_i is the expected value of the response; η_i is the linear predictor; and $\Phi(\cdot)$ is the cumulative distribution function of the standard-normal distribution.

Figure 7.1: From Fox 2016 Applied Regression Analysis and Gerneralized Linear Models; page 419.

Table 15.2 Canonical Link, Response Range, and Conditional Variance Function for Exponential Families

Family	*Canonical Link*	*Range of* Y_i	$V(Y_i\|\eta_i)$
Gaussian	Identity	$(-\infty, +\infty)$	ϕ
Binomial	Logit	$\frac{0,1,\ldots,n_i}{n_i}$	$\frac{\mu_i(1-\mu_i)}{n_i}$
Poisson	Log	$0,1,2,\ldots$	μ_i
Gamma	Inverse	$(0,\infty)$	$\phi\mu_i^2$
Inverse-Gaussian	Inverse-square	$(0,\infty)$	$\phi\mu_i^3$

NOTE: ϕ is the dispersion parameter, η_i is the linear predictor, and μ_i is the expectation of Y_i (the response). In the binomial family, n_i is the number of trials.

Figure 7.2: From Fox 2016 Applied Regression Analysis and Gerneralized Linear Models; page 421.

Note, the linear model is a special case of a GLM with the identity function as the link function.

7.3 Example Poisson GLM

Count data occurs whenever the response is a count of some quantities. For example number of accidents, number of individuals, number of years, etc.

A possible model for such data is the Poisson distribution, with the following probability mass function:

$$f(y;\lambda) = \frac{\lambda^y \exp(-\lambda)}{y!}$$

With:

- $y \geq 0$ and y being an integer, and
- λ being the rate and $\lambda \geq 0$.

Note, that for a Poisson distribution: $E(y) = Var(y) = \lambda$.

1. The distribution of the response: **Poisson distribution**
2. Specification of the systematic part of the model $\eta_i = \beta_0 + \beta_1 x_i$
3. Link function: log

That means:

- $\eta_i = \beta_0 + \beta_1 x_i$
- $\log(\lambda_i) = \eta_i$ or $\lambda_i = \exp(\eta_i)$.
- Finally, $y_i \sim P(\lambda_i)$

In R there is a function called `glm`.

```
dat <- read.table(here::here("data/RoadKills.txt"), header = TRUE)
pr1 <- glm(TOT.N ~ D.PARK, data = dat, family = poisson(link = "log"))
summary(pr1)
```

```
Call:
glm(formula = TOT.N ~ D.PARK, family = poisson(link = "log"),
    data = dat)

Coefficients:
```

```
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  4.316e+00  4.322e-02   99.87   <2e-16 ***
D.PARK      -1.059e-04  4.387e-06  -24.13   <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: 1071.4  on 51  degrees of freedom
Residual deviance:  390.9  on 50  degrees of freedom
AIC: 634.29

Number of Fisher Scoring iterations: 4
```

How many road kills would we expect 1000 m away from a park?

```
coef(pr1) %*% rbind(1, 1000)
```

```
         [,1]
[1,] 4.210634
```

This is on the linear scale, we have to apply $g(\cdot)^{-1}$ to get back at the response scale. Since we used the log link function, the inverse can be obtained with exp.

```
exp(coef(pr1) %*% rbind(1, 1000))
```

```
         [,1]
[1,] 67.39928
```

Note, that the `predict()` function also has an argument `type` that can take the value `'link'` or `'response'`.

By default type is set to `"link"`:

```
predict(pr1, data.frame(D.PARK = 1000))
```

```
       1
4.210634
```

```
predict(pr1, data.frame(D.PARK = 1000), type = "response")
```

```
       1
67.39928
```

7.3.1 Interpretation

1. Transform coefficients
2. Using figures

```
pr0 <- glm(TOT.N ~ D.PARK, data = dat, family = poisson())
```

7.3.1.1 Transform coefficients

We used the model

$$\begin{aligned}\log \mu &= \beta_0 + \beta_1 x_1 \\ \mu &= \exp(\beta_0 + \beta_1 x_1) \\ &= \exp(\beta_0) \times \exp(\beta_1 x_1) \\ &= \exp(\beta_0) \times \exp(\beta_1)^{x_1}\end{aligned}$$

This means that the effect of β_1 is multiplicative. Increasing x_1 by one unit, changes y by the factor $\exp(\beta_1)$.

```
p1 <- predict(pr0, data.frame(D.PARK = 1001), type = "response")
p2 <- predict(pr0, data.frame(D.PARK = 1002), type = "response")
p2 / p1
```

```
        1
0.9998942
```

This is the same as

```
exp(coef(pr0)[2])
```

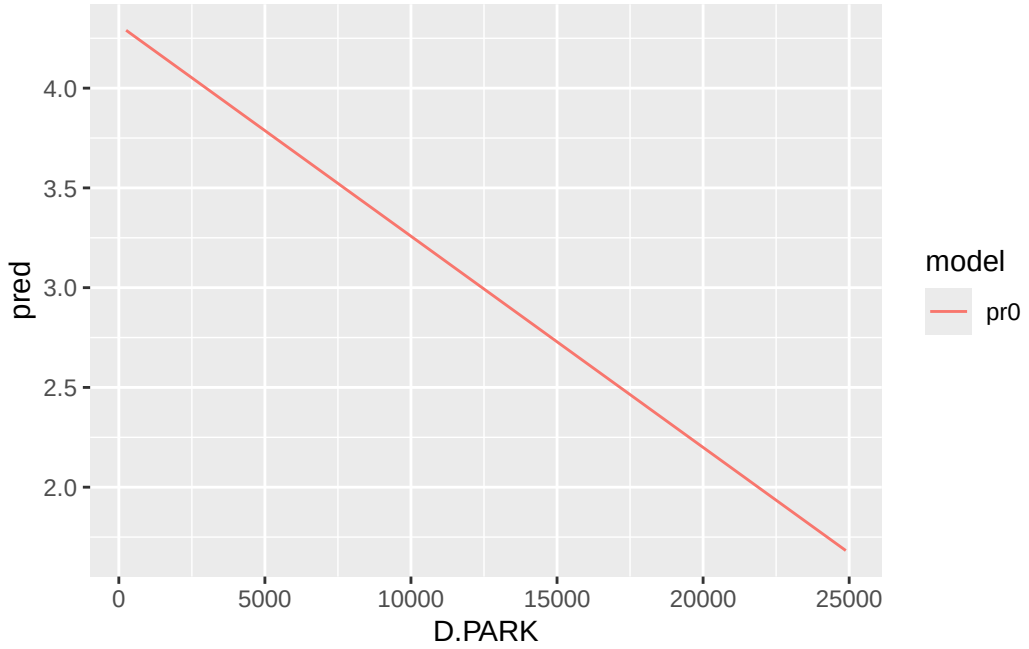
```
   D.PARK
0.9998942
```

7.3.1.2 Using figures

Note, I use here the package `modelr` for making predictions. This is between `ggeffects` and the base R `predict()` function.

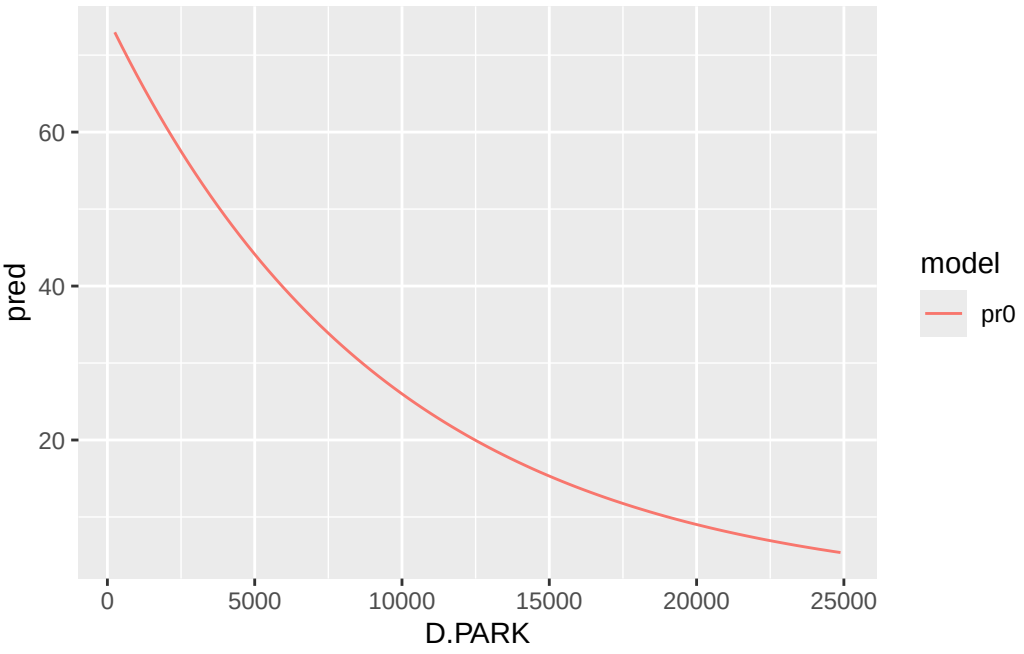
I use first the function `data_grid()` as a more convient alternative to `expand.grid()`. Then `gather_predictions()` is slightly more convinient than `predict()`.

```
library(modelr)
data_grid(dat,
          D.PARK = seq_range(D.PARK, 100),
          N.PATCH = typical(N.PATCH),
          S.RICH = typical(S.RICH)) %>%
  gather_predictions(pr0) %>%
  ggplot(aes(D.PARK, pred, col = model)) + geom_line()
```



We can also specify the response type.

```
data_grid(dat,
          D.PARK = seq_range(D.PARK, 100),
          N.PATCH = typical(N.PATCH),
          S.RICH = typical(S.RICH)) %>%
  gather_predictions(pr0, type = "response") %>%
  ggplot(aes(D.PARK, pred, col = model)) + geom_line()
```



7.4 The logistic regression

The logistic regression can be use, if we have a response that can take on two values: 0 or 1, yes or no, male or female, ...

What we model, is the expectation π of a binomial distribution.

1. The distribution of the response: **Binomial distribution**
2. Specification of the systematic part of the model $\eta_i = \beta_0 + \beta_1 x_i$
3. Link function: *logit*

Thus, we have

$$logit(\pi_i) = \eta_i$$

this means that

$$\pi_i = \frac{\exp(\eta_i)}{1 + \exp(\eta_i)}$$

7.4.1 Interpretation of parameters

For the logistic regression, the coefficients can be interpreted in terms of odds ratio.

```
dat <- read.table(here::here("data/Boar.txt"), header = TRUE) %>%
  mutate(sex = factor(SEX, labels = c("male", "female")))
m1 <- glm(Tb ~ sex + LengthCT, data = dat, family = binomial())
summary(m1)
```

```
Call:
glm(formula = Tb ~ sex + LengthCT, family = binomial(), data = dat)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.02584    0.70309  -5.726 1.03e-08 ***
sexfemale   -0.20098    0.19101  -1.052    0.293
LengthCT     0.03359    0.00578   5.811 6.21e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 681.25  on 493  degrees of freedom
Residual deviance: 640.12  on 491  degrees of freedom
  (163 observations deleted due to missingness)
AIC: 646.12

Number of Fisher Scoring iterations: 4
```

The odds are defined as

$$\frac{\pi_i}{1-\pi_i}$$

Using the inverse logit $1/(1+\exp(-\eta))$ we get

$$\frac{1/(1+\exp(-\beta_0+\beta_1 x))}{1/(1+\exp(-\beta_0+\beta_1(x+1))} = \exp(\beta_1)$$

- If $\beta_1 = 0$ the odds do not change.
- If $\beta_1 < 0$ the odds decrease.
- If $\beta_1 > 0$ the odds increase.

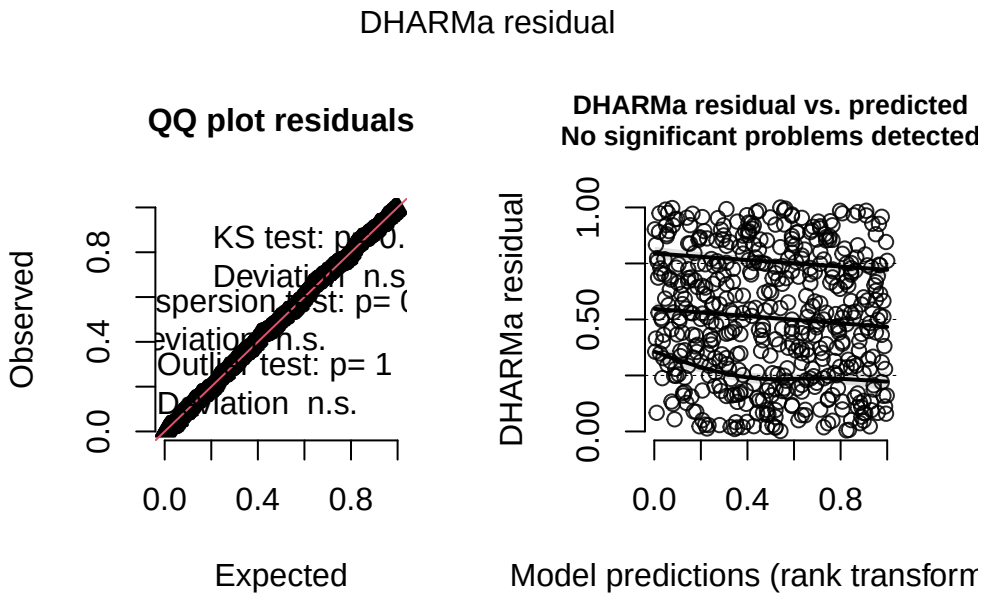
7.4.2 How well does the model fit the data?

Residual diagnostics is not straight-forward any more. One options to check residuals are simulations from the model (these are called randomized quantile residuals). The package `DHARMa` implements this.

```
library(DHARMa)
```

```
This is DHARMa 0.4.7. For overview type '?DHARMa'. For recent changes, type news(package = 'D
```

```
plot(simulateResiduals(m1))
```



Exercise 1: Fitting a GLM

Linden and Rohloff (2015) studied white-headed woodpeckers in California (USA) using point counts. The file `woodp_Occ.csv` contains data from three visits at 66 sites. For each site and visit we know the if a woodpecker was present or not (columns `y.1`, `y.2`, `y.3`), the day of the year (columns `date.1`, `date.2`, `date.3`) and the snag density (the number of tree snags per ha; column `snags`). Tree snags are standing but dead or dying trees. Only use the first session of data collection for this exercise (i.e., `y.1`).

1. What would be a suitable model for this kind of data (i.e., think of the distribution of y)?
2. Fit the model that you decided on in 1).
3. Give an interpretation of the model results.

4. Create a suitable plot to communicate the results.
5. Validate your model.
6. *Optional*: Can you fit the same model with Stan?