

Linear models and generalized linear models

in R

Julien Martin
University of Ottawa

2024-02-04

Linear models

What is a linear regression

Simple linear regression

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon$$
$$\epsilon \sim N(0, \sigma_\epsilon^2)$$

or in distributional notation

$$Y \sim N(\beta_0 + \beta_1 x, \sigma_\epsilon^2)$$

General linear model

$$Y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \dots + \epsilon$$
$$\epsilon \sim N(0, \sigma_\epsilon^2)$$

and

Linear model assumptions

Some are made on the residuals and others on the independent variables. None are made on the (unconditionned) dependent variable.

Residuals are assumed to:

- have a mean of zero
- be independent
- be normally distributed
- be homoscedastic

Independent variables are assumed to:

- have a linear relation with Y
- be measured without error
- to be independent from each other

Maximum likelihood

Technique used for estimating the parameters of a given distribution, using some observed data.

For Example:

Population is known to follow a “normal distribution” but “mean” and “variance” are unknown, MLE can be used to estimate them using a limited sample of the population.

Likelihood vs probability

We maximize the likelihood and make inferences on the probability

Likelihood

$$L(\text{parameters} | \text{data})$$

How likely it is to get those parameters given the data.

Probability

$$P(\text{data} | \text{null parameters})$$

Probability to get the data given the null parameters. Or how probable it is to get those data according to the null model.

Maximum likelihood approach

$$L(\text{parameters} | \text{data}) = \prod_{i=1}^n f(\text{data}_i | \text{parameters})$$

where f is the probability density function of your model.

Working with product is more painful than with sum, we can take the log:

$$\ln(L(\text{parameters} | \text{data})) = \sum_{i=1}^n \ln(f(\text{data}_i | \text{parameters}))$$

Need to solve:

$$\frac{\delta \ln(L(\text{parameters} | \text{data}))}{\delta \text{parameters}} = 0$$

For multiple regression, the parameters β s are given by $\beta = (X^T X)^{-1} X^T y$

Doing linear models in R

Simply use `lm()` function. It works for everything anova, ancova, t-test.

We will use data of sturgeon measurements at different locations in Canada.

```
1 dat <- read.csv("data/lm_example.csv")
2 str(dat)
```

```
'data.frame': 92 obs. of 4 variables:
 $ year    : int  1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 ...
 $ fklnghth: num  41.9 50.2 50.2 47.3 49.6 ...
 $ locate   : chr  "NELSON"      "LOFW"          "LOFW"          "NELSON"      ...
 $ age      : int  11 24 23 20 23 20 23 19 17 14 ...
```

Fitting a model and checking assumptions

First we load the needed packages for:

- data manipulation: `tidyverse`
- fancy plots: `ggplot2`
- type III anova: `car`
- fancy and nicer visual assumptions checks: `performance`
- formal assumptions tests: `lmtest`

```
1 library(car)
2 library(performance)
3 library(lmtest)
4 library(tidyverse)
```

Data exploration

R Code

Plot

```
1 ggplot(data = dat, aes(x = age, y = fklnghth)) +  
2   facet_grid(. ~ locate) +  
3   geom_point() +  
4   stat_smooth(method = lm, se = FALSE) +  
5   stat_smooth(se = FALSE, color = "red") +  
6   labs(  
7     y = "Fork length",  
8     x = "Age"  
9   )
```

Creating log10 transform

```
1 dat <- dat %>%
2   mutate(
3     lage = log10(age),
4     lfkl = log10(fklngh)
5   )
```

Data exploration: with log

Code

Plot

```
1 ggplot(data = dat, aes(x = lage, y = lfk1)) +  
2   facet_grid(. ~ locate) +  
3   geom_point() +  
4   stat_smooth(method = lm, se = FALSE) +  
5   stat_smooth(se = FALSE, color = "red") +  
6   labs(  
7     y = "log 10 Fork length",  
8     x = "Log 10 Age"  
9   )
```

Fit the model

```
1 m1 <- lm(lfkl ~ lage + locate + lage:locate, data = dat)
2 summary(m1)
```

Call:

```
lm(formula = lfkl ~ lage + locate + lage:locate, data = dat)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.09375	-0.01864	-0.00253	0.02090	0.08030

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.24287	0.04370	28.443	< 2e-16 ***
lage	0.31431	0.03292	9.546	3.08e-15 ***
locateNELSON	0.19295	0.06331	3.048	0.00304 **
lage:locateNELSON	-0.14276	0.04902	-2.912	0.00455 **

Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'
	0.1 '	'	1	

Anova for factors

```
1 Anova (m1, type = 3)
```

Anova Table (Type III tests)

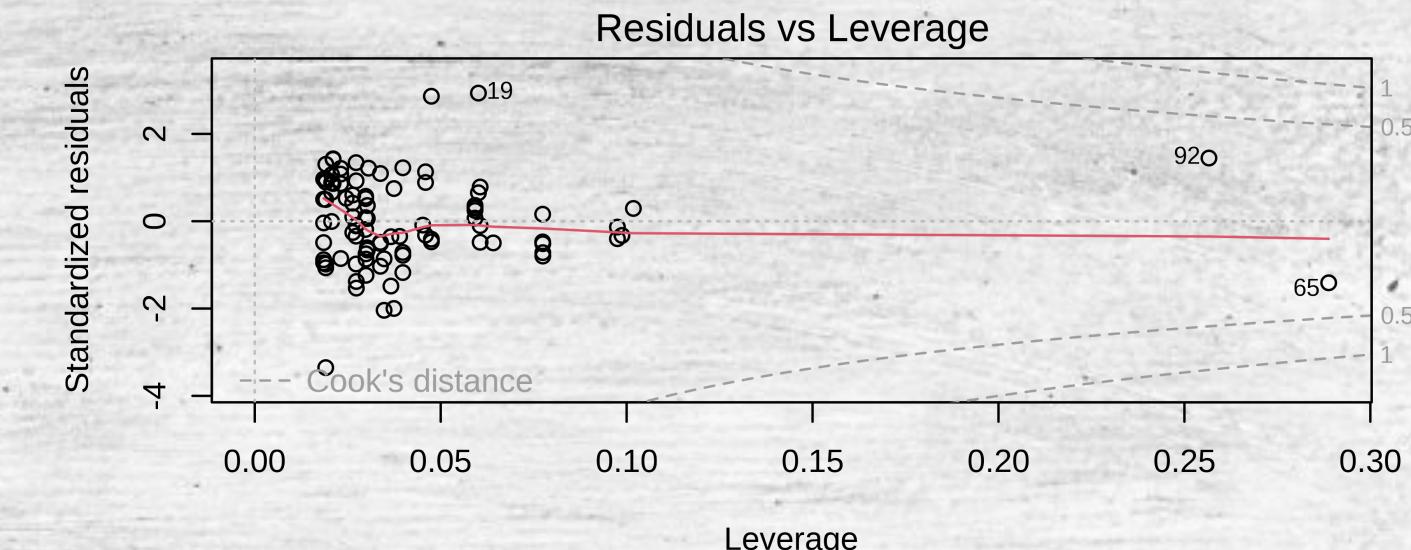
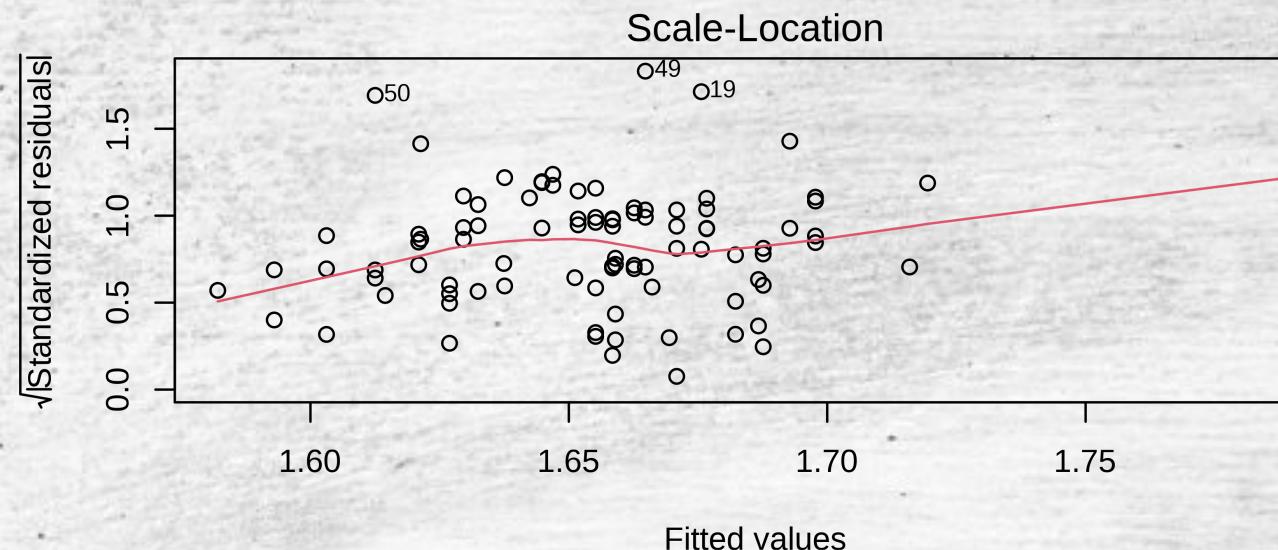
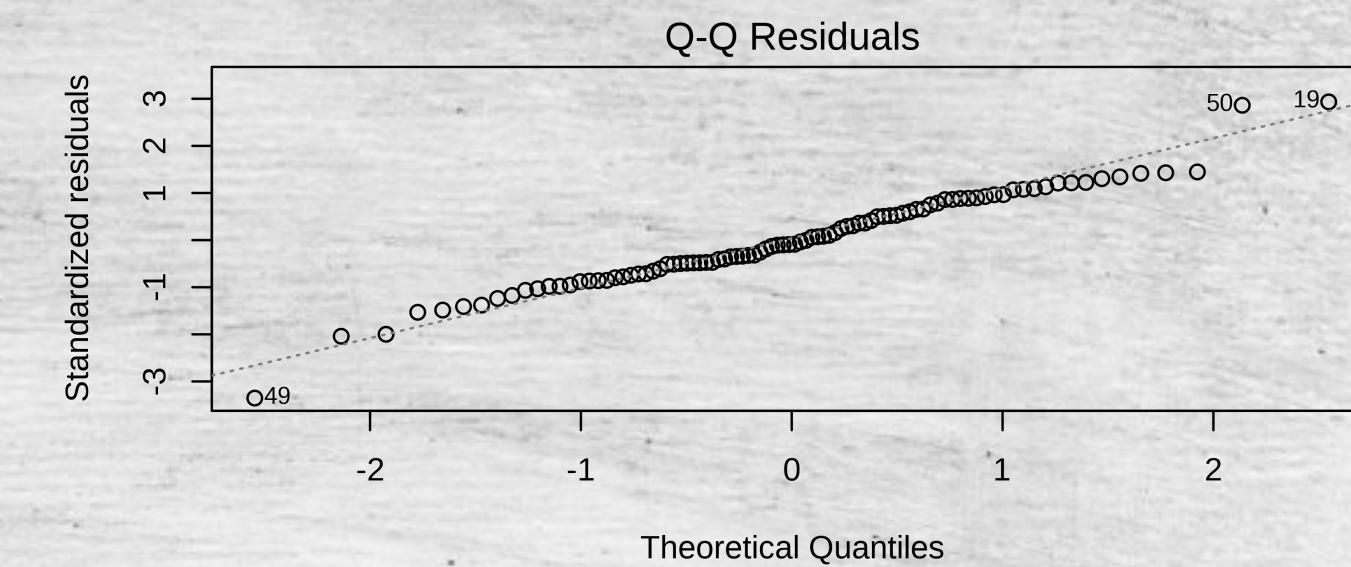
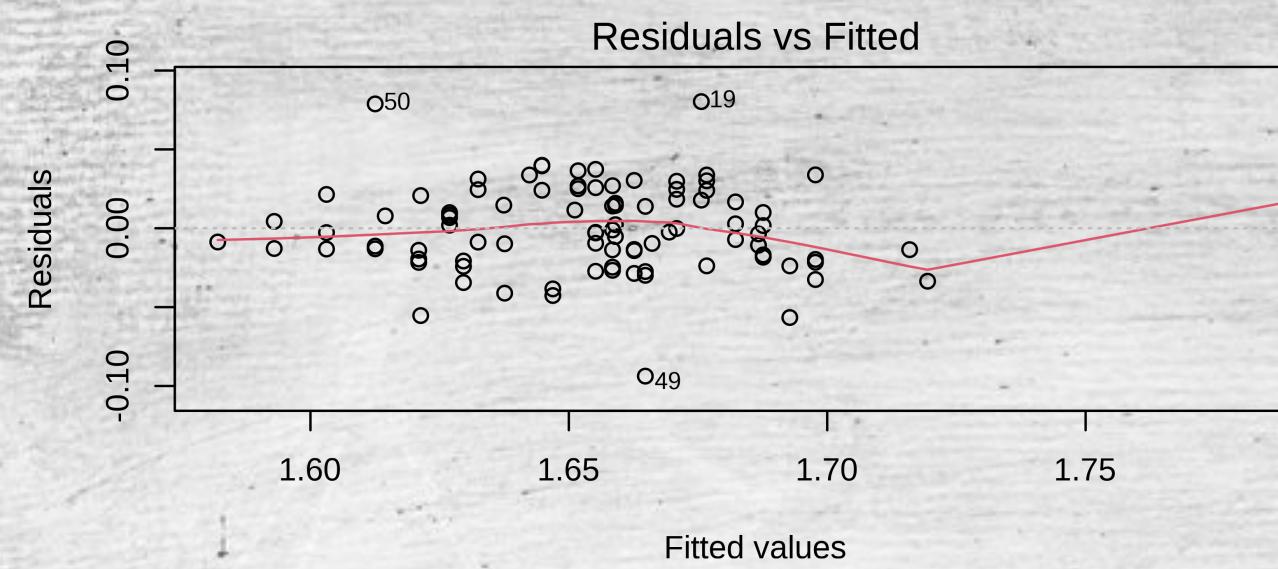
Response: lfk1

	Sum Sq	Df	F value	Pr (>F)	
(Intercept)	0.64467	1	809.0107	< 2.2e-16	***
lage	0.07262	1	91.1310	3.079e-15	***
locate	0.00740	1	9.2901	0.003042	**
lage:locate	0.00676	1	8.4815	0.004546	**
Residuals	0.07012	88			

Signif. codes:	0	'***'	0.001	'**'	0.01 '*' 0.05 '.' 0.1 ' ' 1

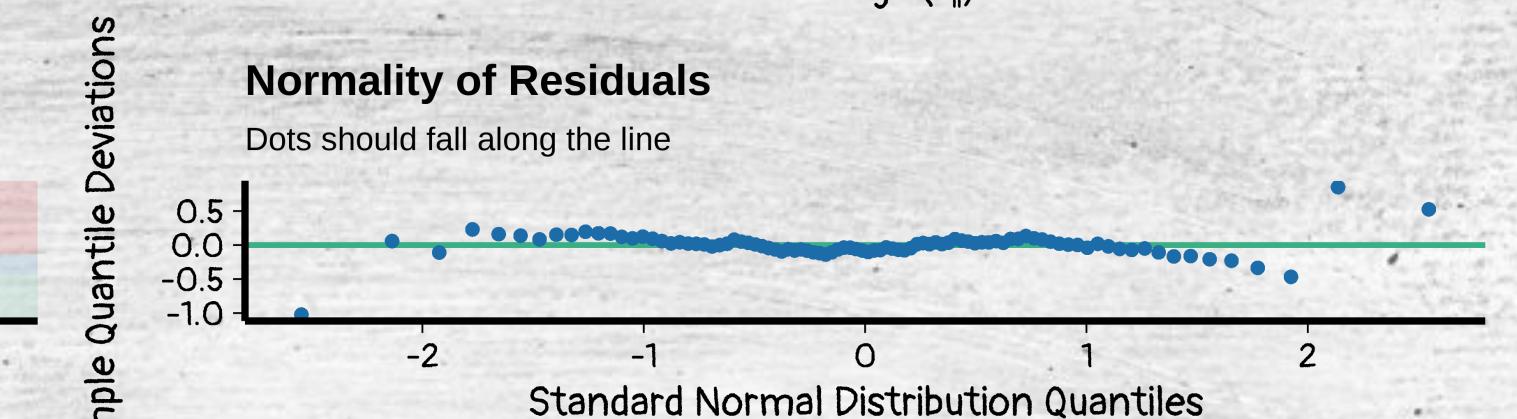
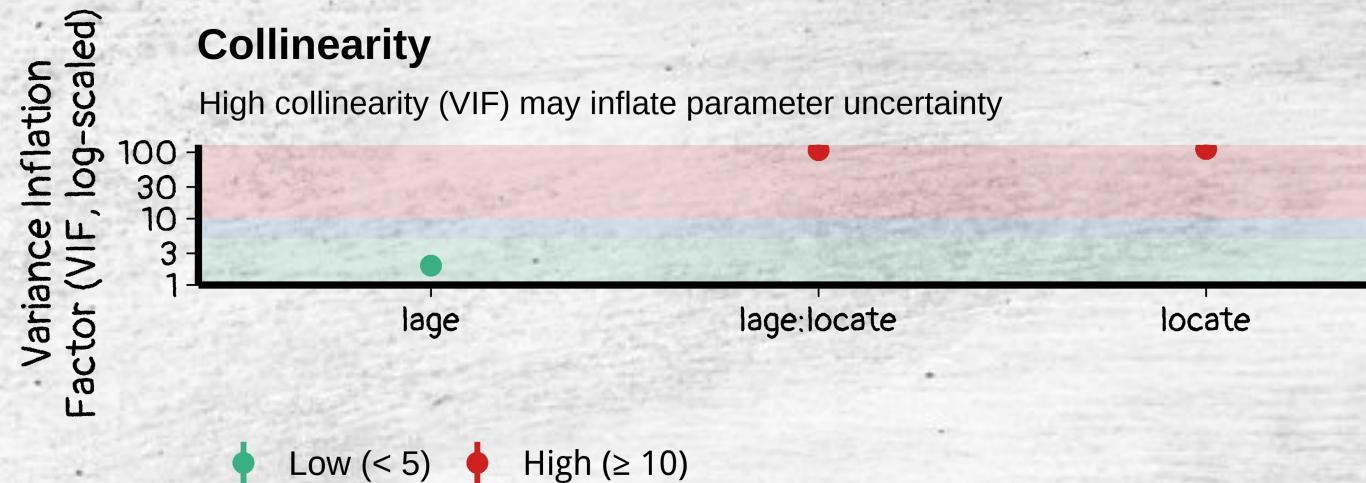
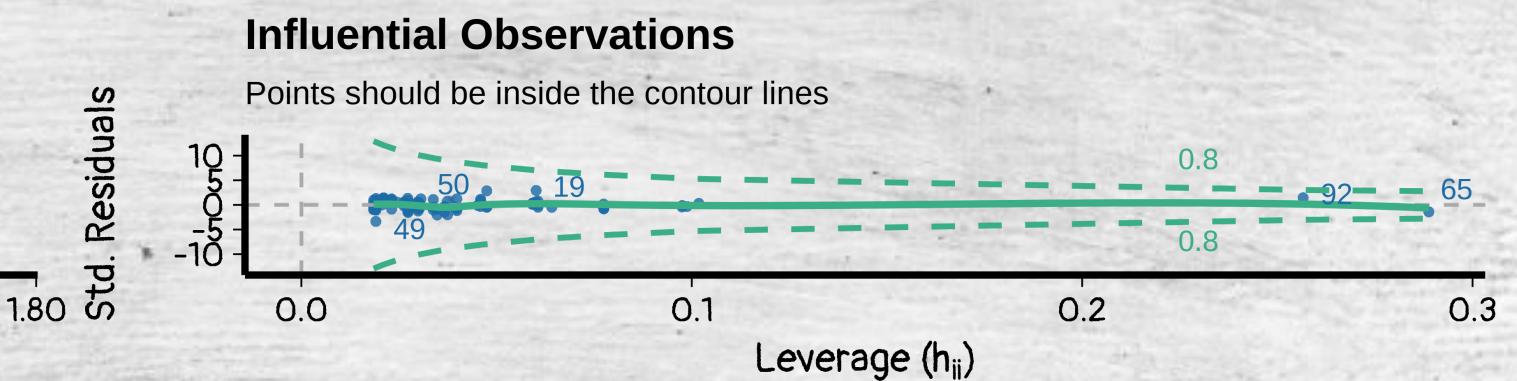
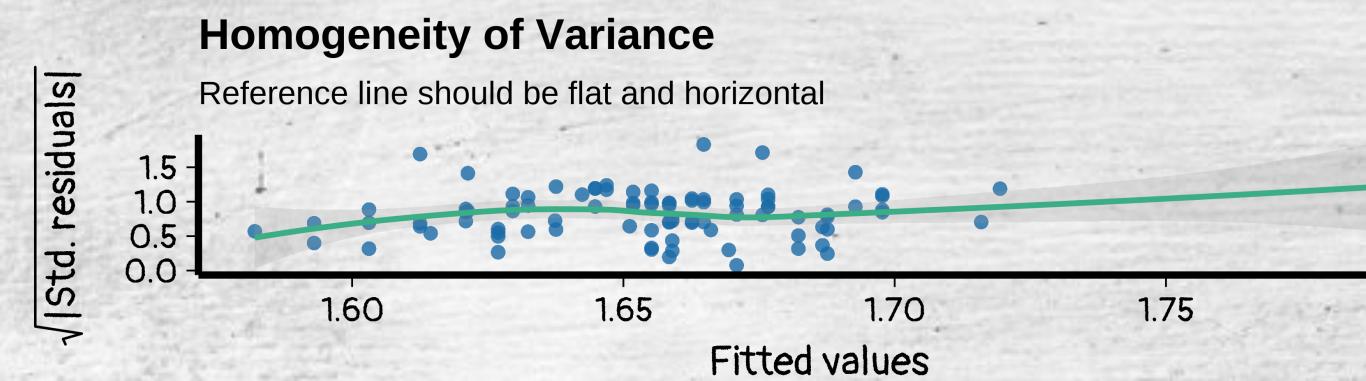
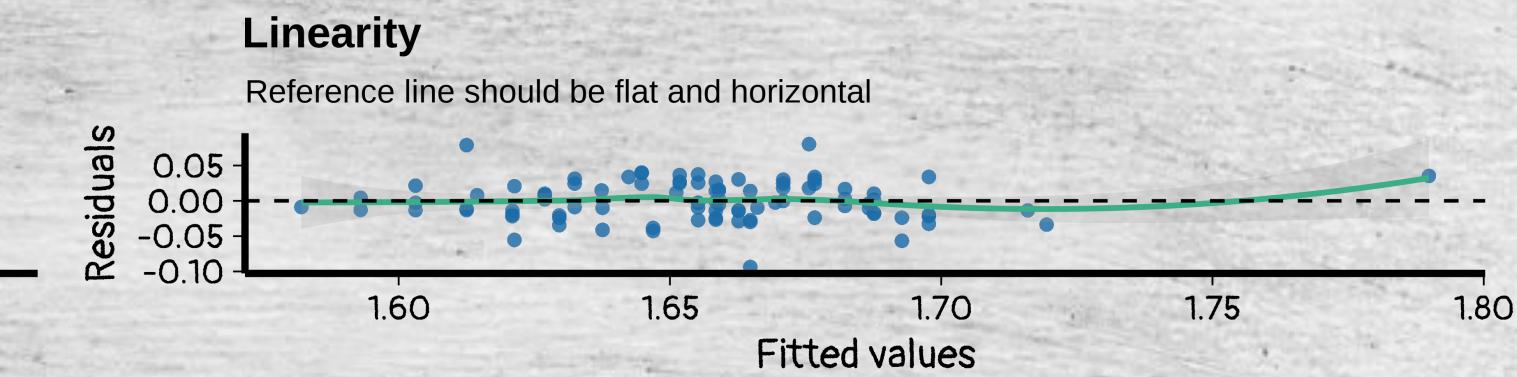
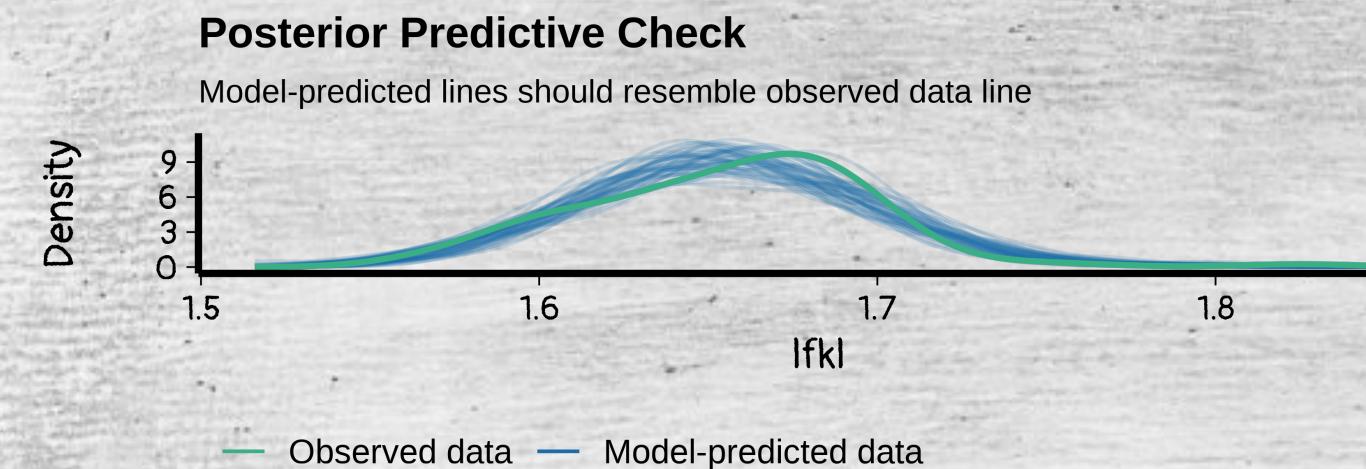
Assumptions (classic)

```
1 par(mfrow = c(2, 2))  
2 plot(m1)
```



Assumptions (Nicer)

```
1 check_model(m1)
```



Formal tests

Normality of residuals

```
1 shapiro.test(residuals(m1))
```

```
Shapiro-Wilk normality test
```

```
data: residuals(m1)
W = 0.97639, p-value = 0.09329
```

Formal tests

Heteroscedasticity

```
1 bptest(m1)
```

```
studentized Breusch-Pagan test
```

```
data: m1  
BP = 1.8366, df = 3, p-value = 0.607
```

Formal tests

Linearity

```
1 resettest(m1, power = 2:3, type = "fitted", data = dat)
```

RESET test

```
data: m1
RESET = 1.6953, df1 = 2, df2 = 86, p-value = 0.1896
```

Generalized linear models

Generalized linear models

An extension to linear models

GLM expresses the transformed conditional expectation of the dependent variable $[Y]$ as a linear combination of the regression variables $[X]$

Model has 3 components

- a dependent variable Y with a response distribution to model it: **Gaussian, Binomial, Bernoulli, Poisson, negative binomial, zero-inflated ..., zero-truncated ...**
- linear predictors (or independent variables)

$$\eta = \beta_0 + \beta_1 X_1 + \dots + \beta_k X_k$$

- a link function such that

$$E(Y|X) = \mu = g^{-1}(\eta)$$

Dependent variable

- when continuous and follows *conditional* normal distribution, called **Linear regression**
- Binary outcomes (success/failure), follows a *Binomial distribution*, called **Logistic regression**
- Count data (number of events), follows a *Poisson*, called **Poisson regression**

Classic link functions

- Identity link (form used in linear regression models)

$$g(\eta) = \mu$$

- Log link (used when μ cannot be negative, e.g. Poisson data)

$$g(\eta) = \log(\mu)$$

Logit link (used when μ is bounded between 0 and 1, e.g. binary data)

$$g(\eta) = \log \left(\frac{\mu}{1 - \mu} \right)$$

Linear regression

- Y: continuous
- Response distribution: Gaussian
- Link function: identity

$$g(\eta) = \mu$$

$$\mu(X_1, \dots, X_k) = \beta_0 + \beta_1 X_1 + \dots + \beta_k X_k$$

Logistic regression

- Y: binary or proportion
- Response distribution: Binomial or bernoulli
- Link function: logit

$$g(\eta) = \ln \left(\frac{\mu}{1 - \mu} \right)$$

$$\mu(X_1, \dots, X_k) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \dots + \beta_k X_k)}}$$

Poisson regression

- Y: discrete variable (integers)
- Response distribution: Poisson or Negative binomial
- Link function: natural logarithm

$$g(\eta) = \ln(\mu)$$

$$\mu(X_1, \dots, X_k) = e^{\beta_0 + \beta_1 X_1 + \dots + \beta_k X_k}$$

Model assumptions

- Easy answer none or really few
- More advanced answer I am not sure, it is complicated
- Just check residuals I as usual
- Technically only 3 assumption:
 - Variance is a function of the mean specific to the distribution used
 - observations are independent
 - linear relation on the latent scale

GLMs do not care if the residual errors are Gaussian as long as the specified mean-variance relationship is satisfied by the data

- what about DHaRMA ? It's complicated

Choosing a link function

A link function should map the structural component from $(-\infty, \infty)$ to the distribution interval (e.g. $(0,1)$ for binomial)

So number of link function possible is extremley large.

Choice of link function heavily influenced by field tradition

For binomial models

- **logit** assume modelling probability of an observation to be one
- **probit** assume binary outcome from a hidden gaussian variable (i.e. threshold model)
- **logit & probit** are really similar, both are symmetric but **probit** tapers faster. **logit** coefficient easier to interpret directly
- **cologlog** not-symmetrical

Logistic regression

Data

Here is some data to play with from a study on bighorn sheep.

We will look at the relation between reproduction and age

Loading and tweaking the data

```
1 mouflon0 <- read.csv("data/mouflon.csv")
2 mouflon <- mouflon0 %>%
3   arrange(age) %>%
4   mutate(
5     reproduction = case_when(
6       age >= 13 ~ 0,
7       age <= 4 ~ 1,
8       .default = reproduction
9     )
10   )
```

First plot

Code

Plot

```
1 bubble <- data.frame(
2   age = rep(2:16, 2),
3   reproduction = rep(0:1, each = 15),
4   size = c(table(mouflon$age, mouflon$reproduction) )
5 ) %>%
6   mutate(size = ifelse(size == 0, NA, size))
7 ggplot(
8   bubble,
9   aes(x = age, y = reproduction, size = size)
10 ) +
11   geom_point(alpha = 0.8) +
12   scale_size(range = c(.1, 20), name = "Nb individuals")
```

Fitting the logistic regression

```
1 m1 <- glm(reproduction ~ age,  
2   data = mouflon,  
3   family = binomial  
4 )  
5 summary(m1)
```

Call:

```
glm(formula = reproduction ~ age, family = binomial, data = mouflon)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	3.19921	0.25417	12.59	<2e-16 ***
age	-0.36685	0.03287	-11.16	<2e-16 ***

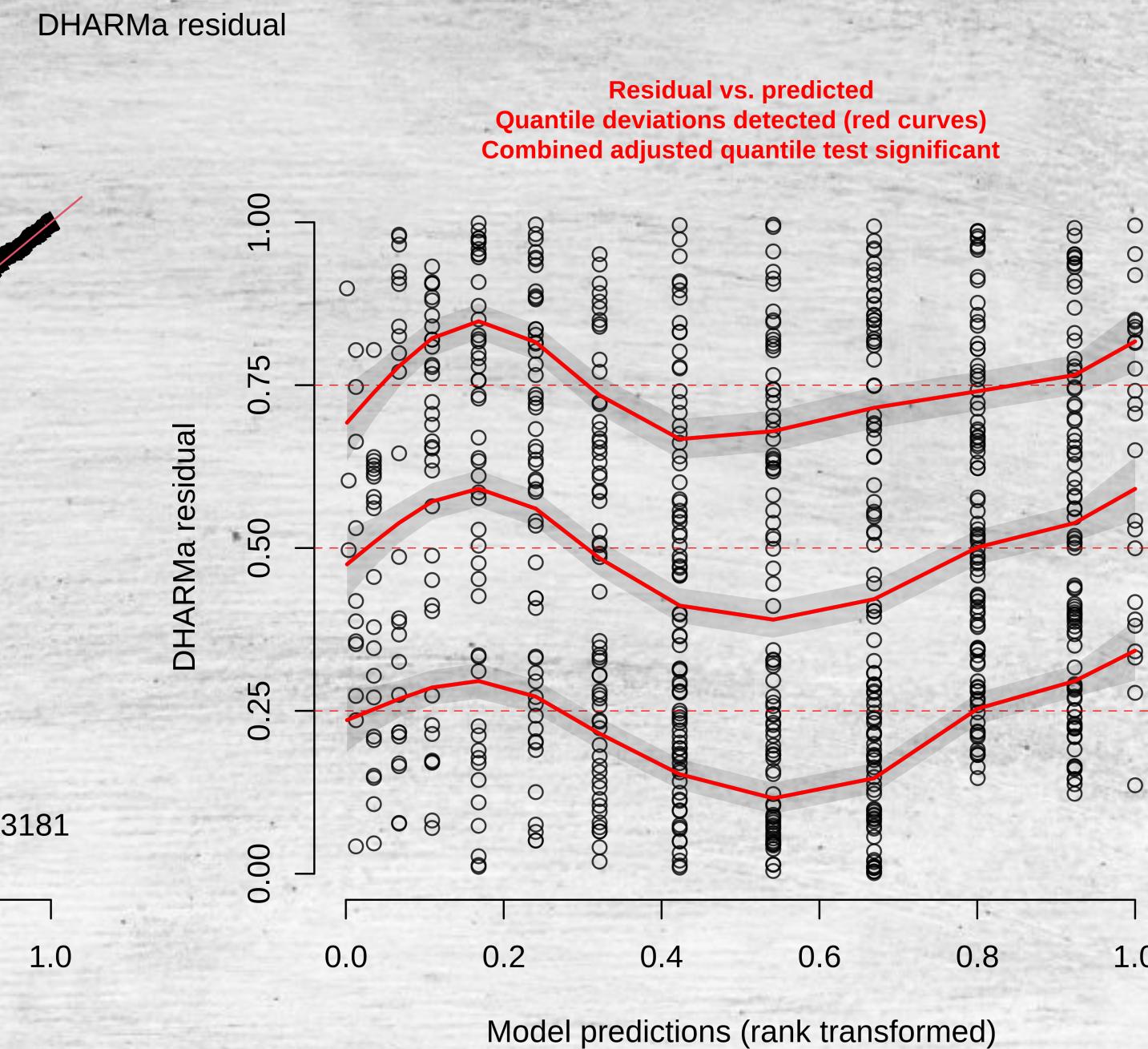
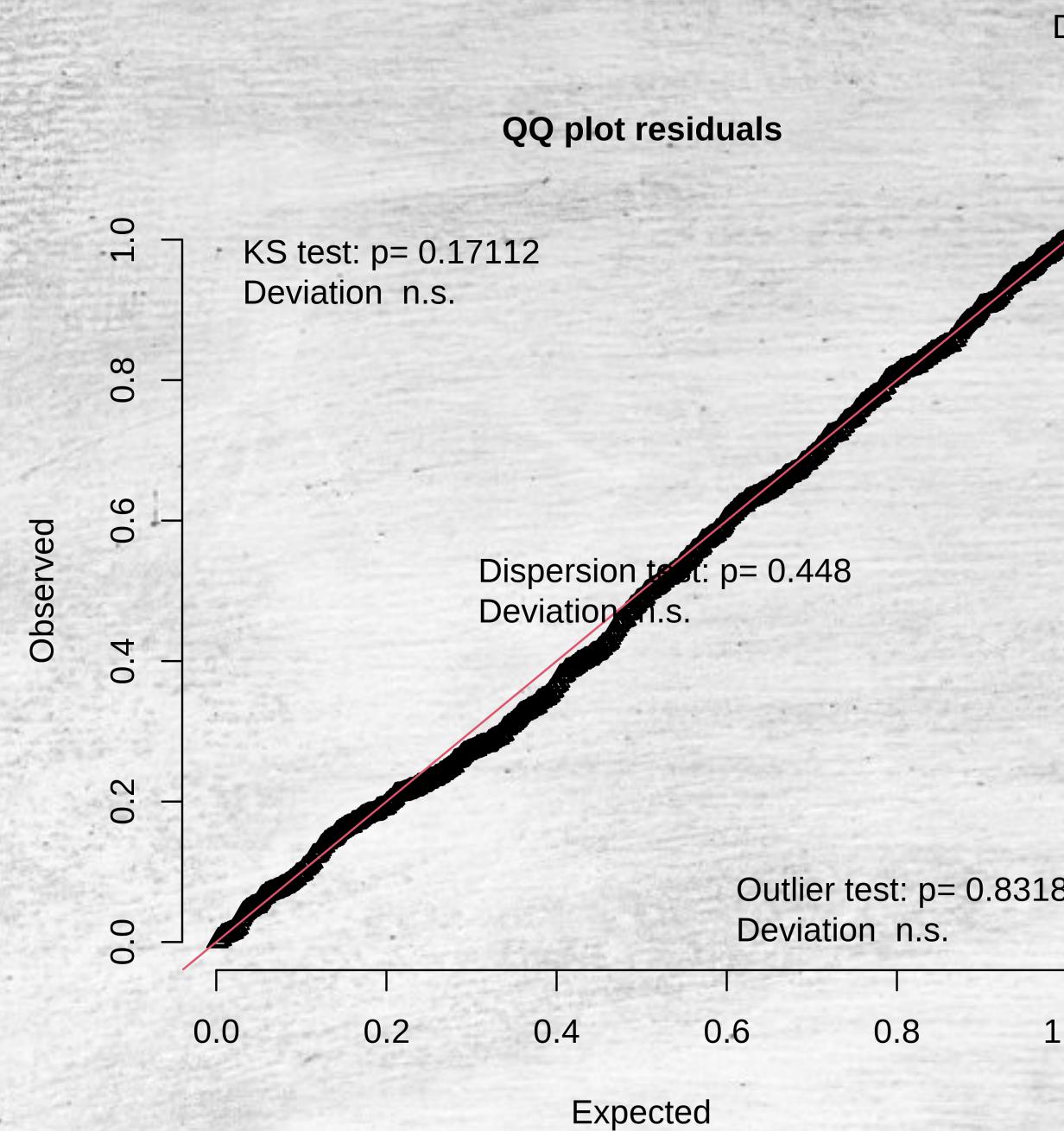
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 928.86 on 715 degrees of freedom
Residual deviance: 767.51 on 714 degrees of freedom
(4 observations deleted due to missingness)

Checking assumptions

```
1 simulationOutput <- simulateResiduals(m1)  
2 plot(simulationOutput)
```

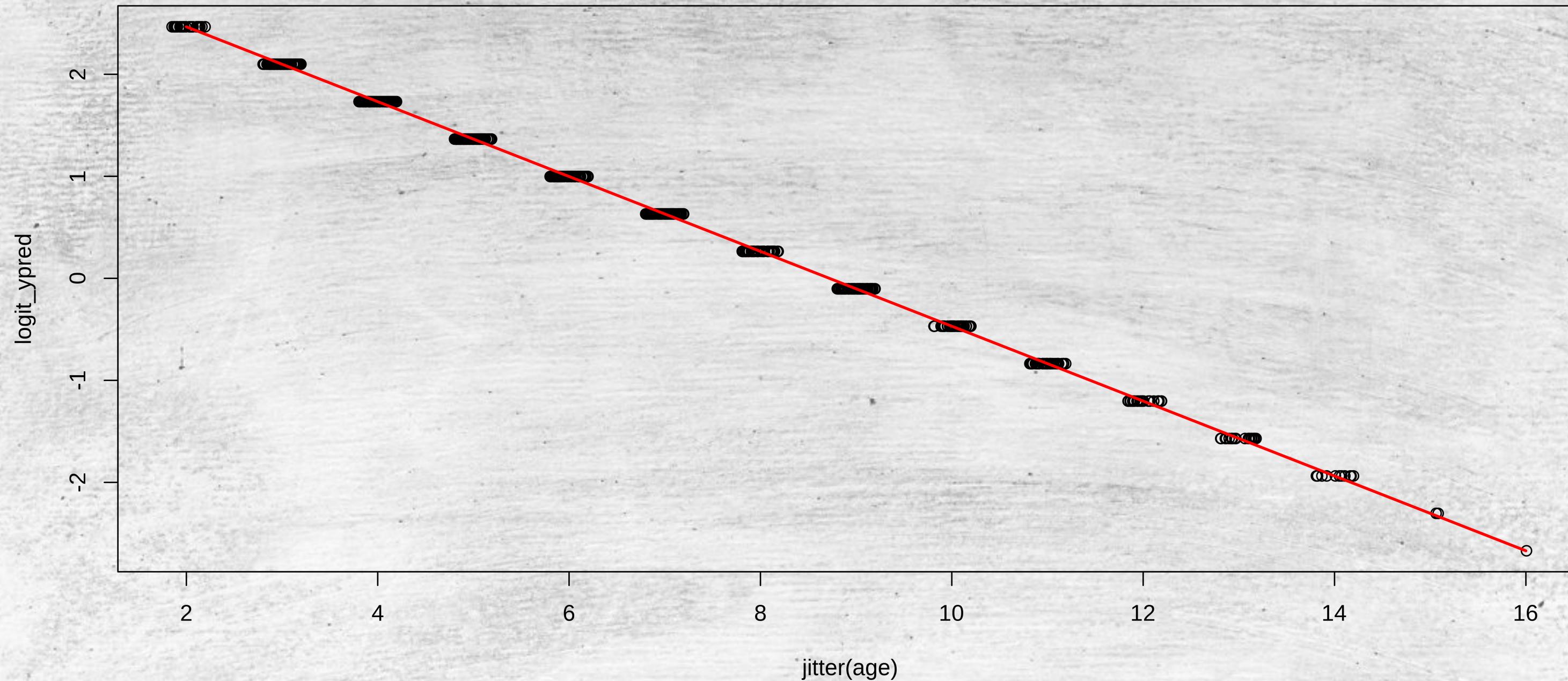


Plotting predictions (latent scale)

plotting the model prediction on the link (latent) scale

```
1 mouflon$logit_ypred <- 3.19921 - 0.36685 * mouflon$age  
2 plot(logit_ypred ~ jitter(age), mouflon)  
3 points(mouflon$age, mouflon$logit_ypred, col = "red", type = "l", lwd =
```

Plotting predictions (latent scale)

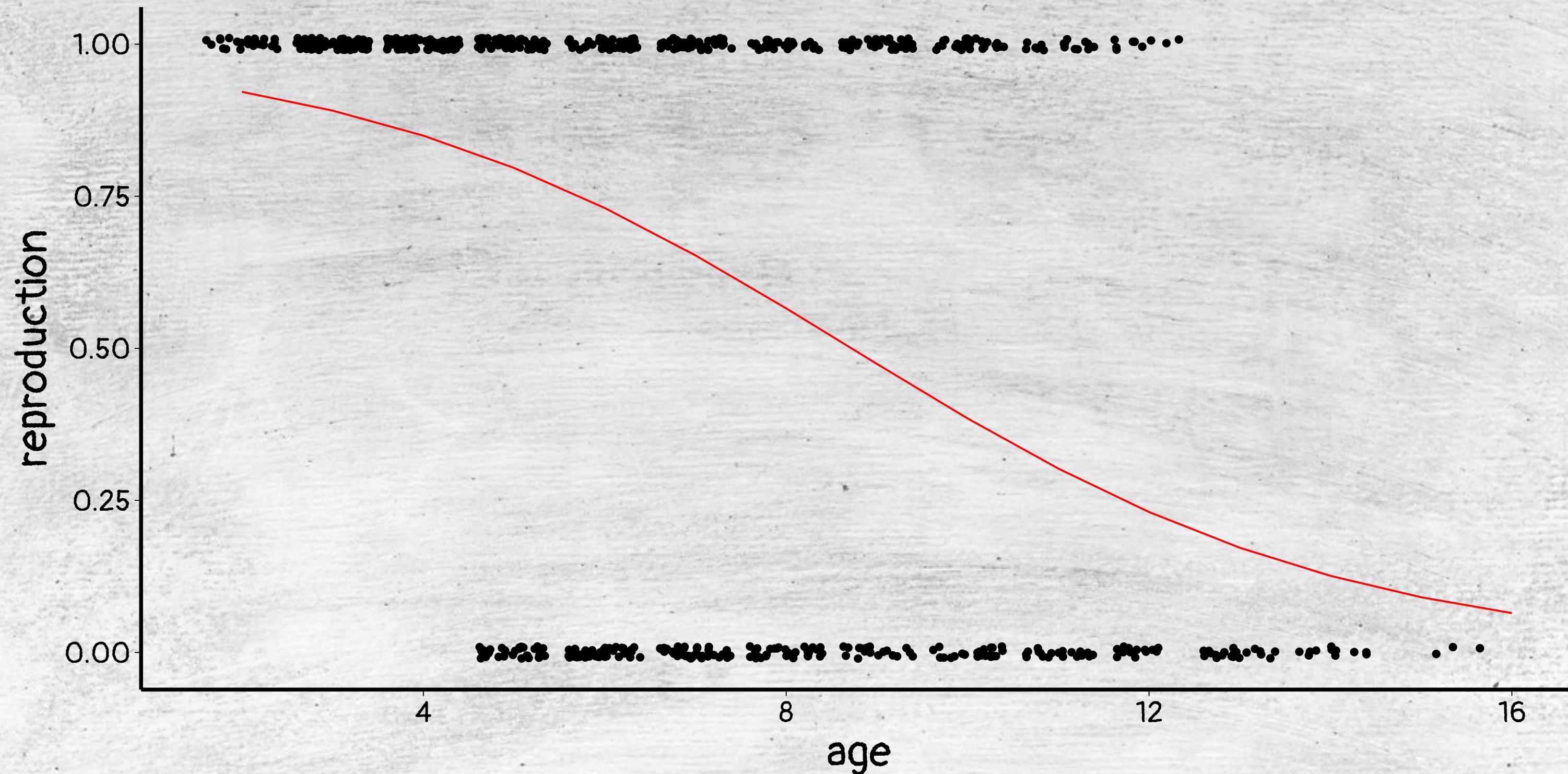


Plotting predictions (obs scale)

plotting on the observed scale

```
1 mouflon$ypred <- exp(mouflon$logit_ypred) / (1 + exp(mouflon$logit_ypre  
2 ggplot(mouflon, aes(x = age, y = reproduction)) +  
3   geom_jitter(height = 0.01) +  
4   geom_line(aes(y=ypred), color = "red")
```

Plotting predictions (obs scale)



Plotting predictions (obs scale)

Code

Plot

but it can be much simpler

```
1 dat_predict <- data.frame(  
2   age = seq(min(mouflon$age), max(mouflon$age), length = 100)  
3 ) %>%  
4   mutate(  
5     reproduction = predict(m1, type = "response", newdata = .)  
6   )  
7  
8 ggplot(mouflon, aes(x = age, y = reproduction)) +  
9   geom_jitter(height = 0.01) +  
10  geom_line(data = dat_predict, aes(x = age, y = reproduction), color =
```

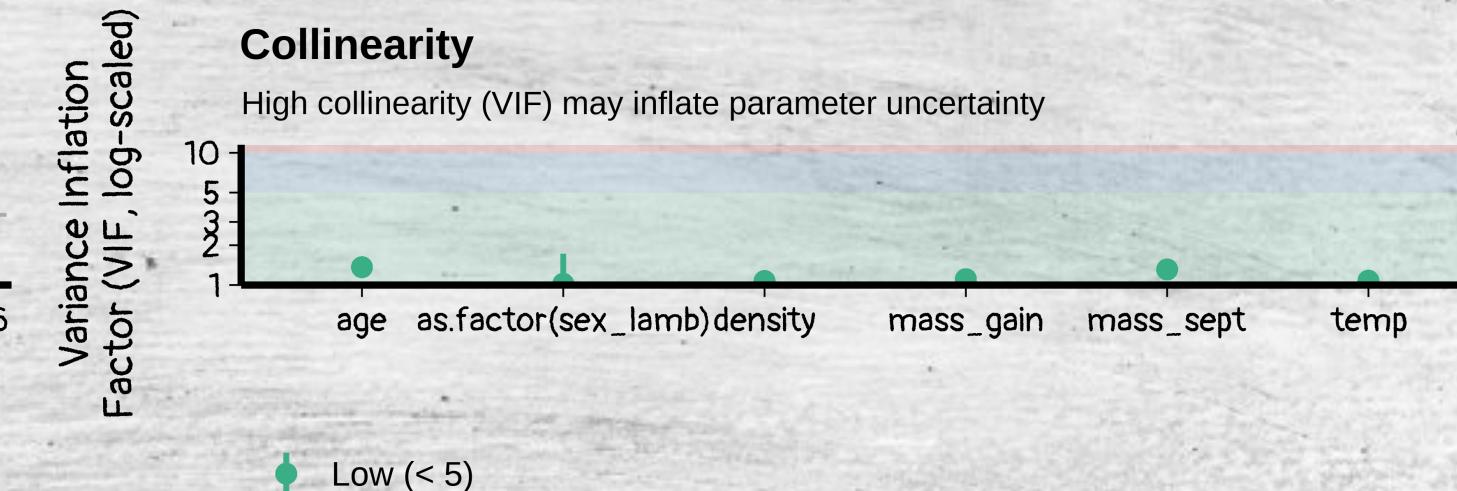
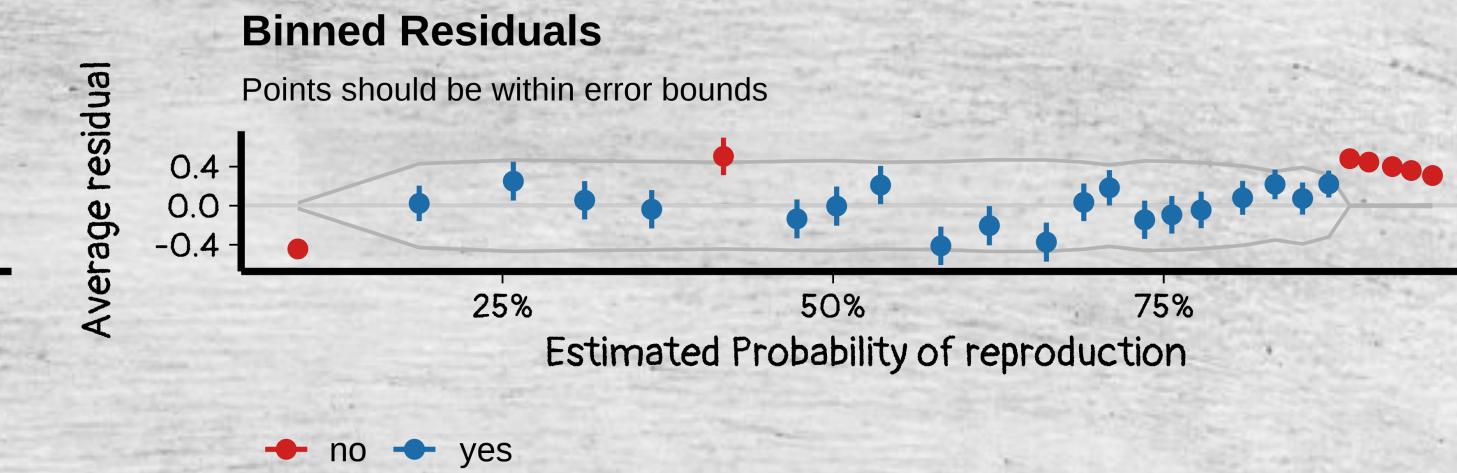
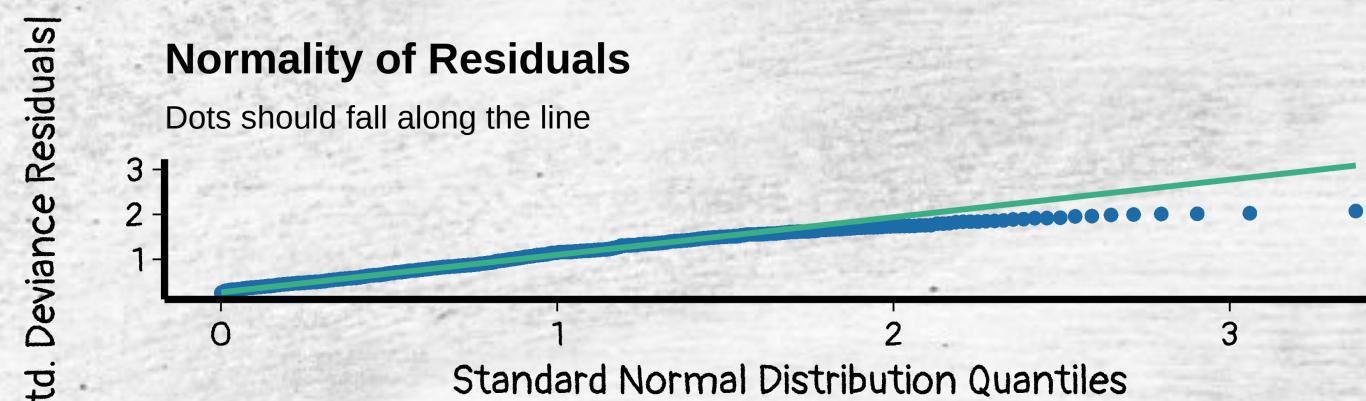
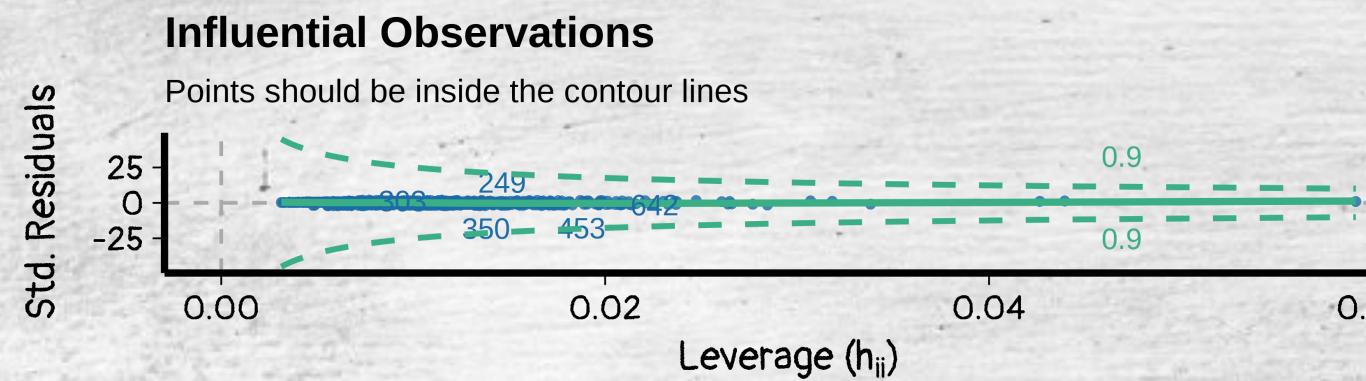
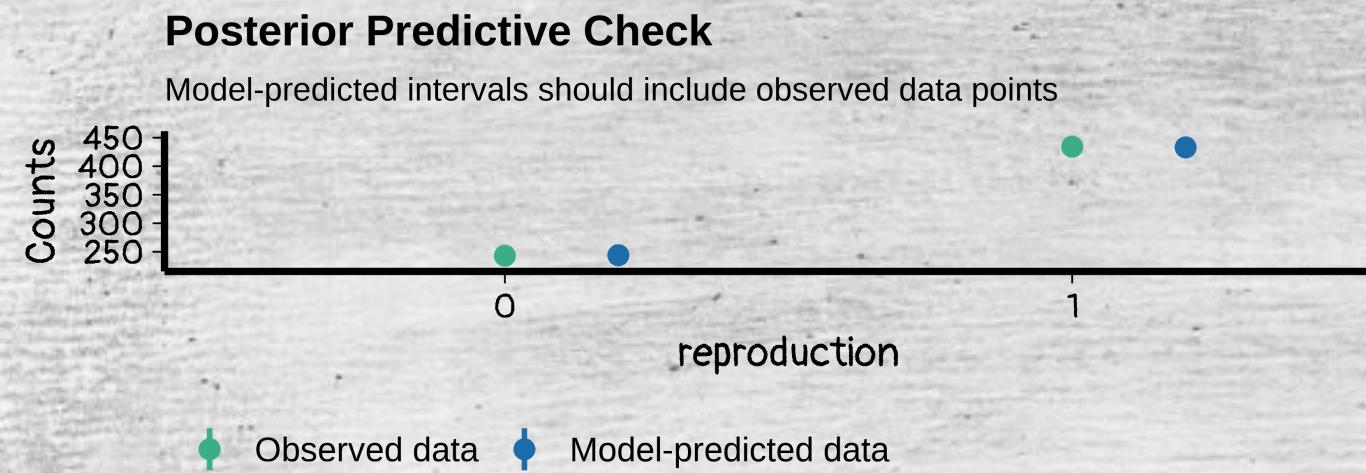
Your turn

we can do the same things with more complex models

```
1 m2 <- glm(  
2   reproduction ~ age + mass_sept + as.factor(sex_lamb) +  
3   mass_gain + density + temp,  
4   data = mouflon,  
5   family = binomial  
6 )
```

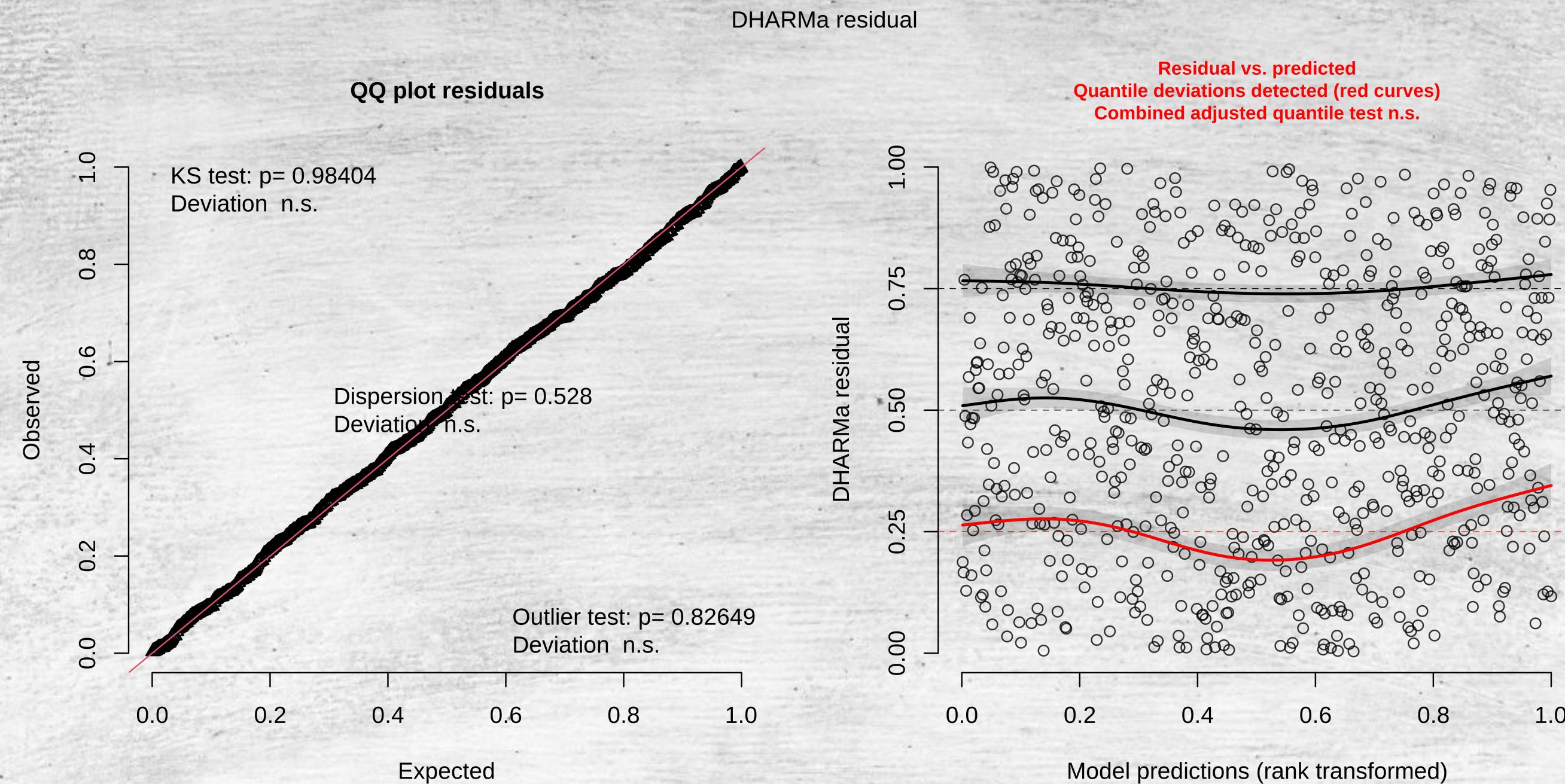
check model

1 check_model (m2)



with DHARMA

```
1 simulationOutput <- simulateResiduals(m2)  
2 plot(simulationOutput)
```



Poisson regression

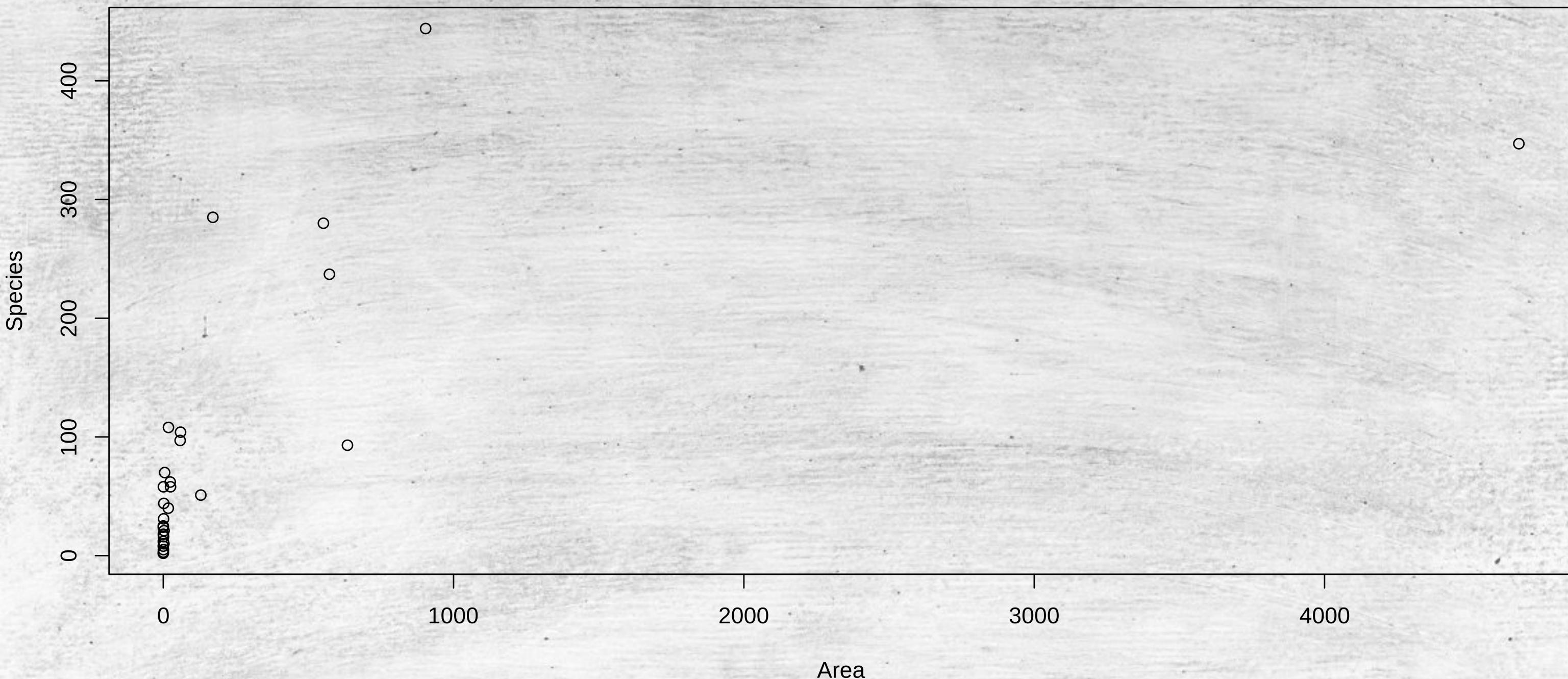
Data

data on galapagos islands species richness

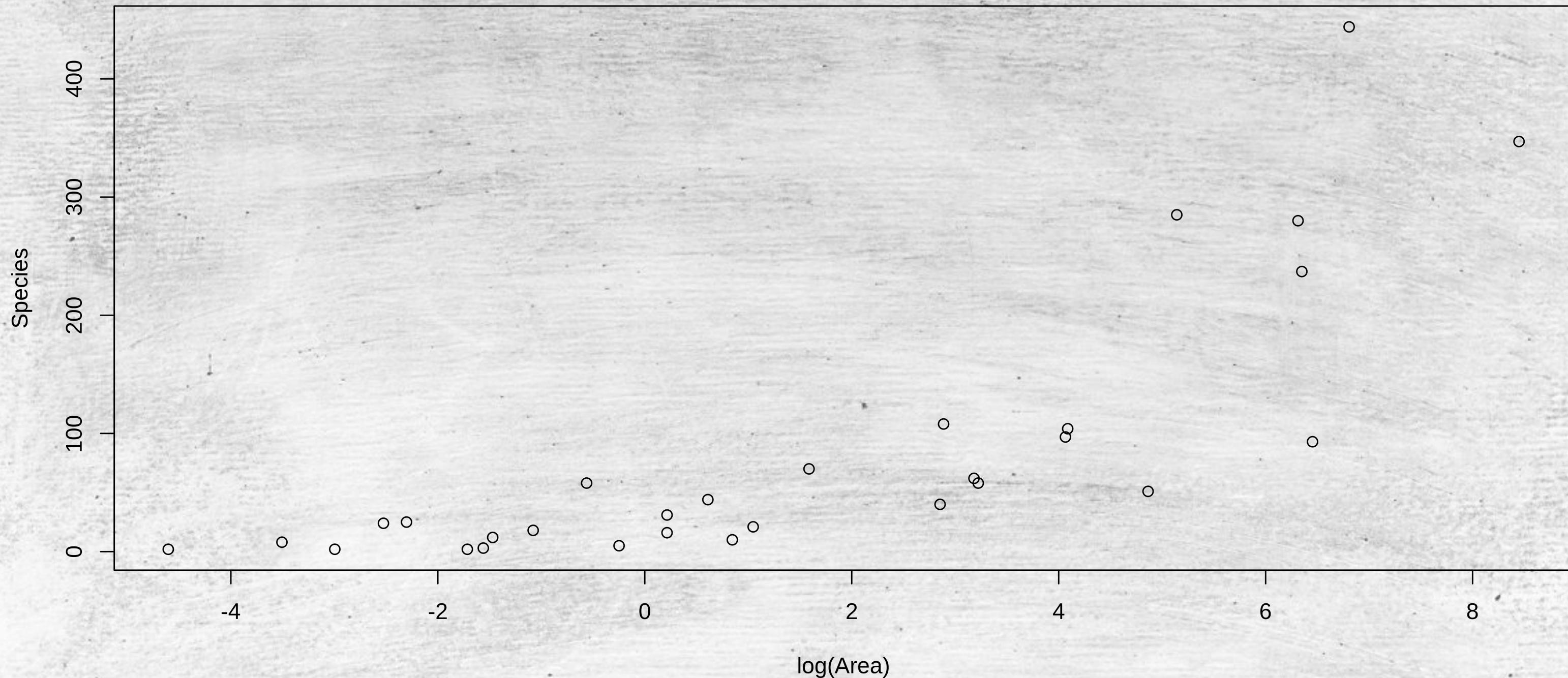
Fit 3 models:

- model of total number of species
- model of proportion of endemics to total
- model of species density

```
1 gala <- read.csv("data/gala.csv")
2 plot(Species ~ Area, gala)
```

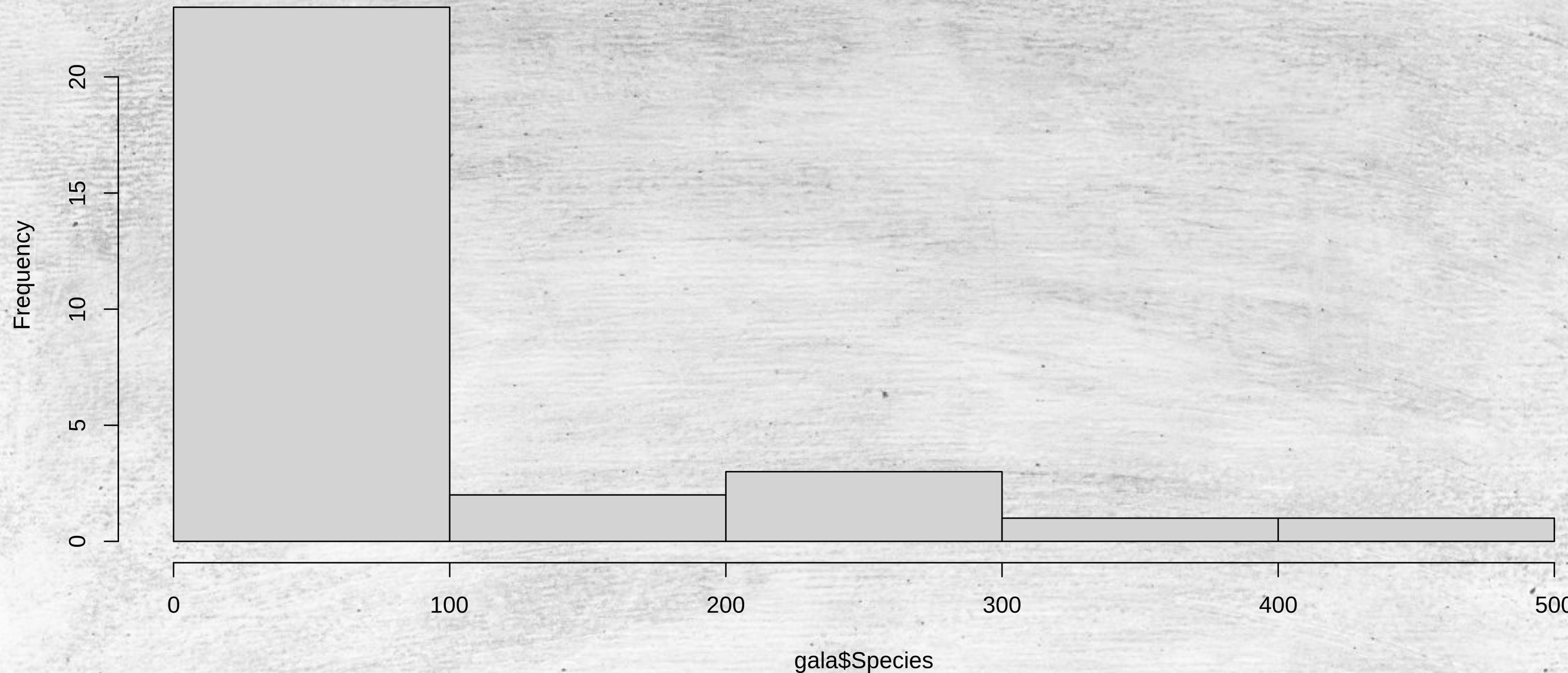


```
1 plot(Species ~ log(Area), gala)
```



```
1 hist(gala$Species)
```

Histogram of gala\$Species



```
1 modpl <- glm(Species ~ Area + Elevation + Nearest, family = poisson, ga  
2 summary(modpl)
```

Call:

```
glm(formula = Species ~ Area + Elevation + Nearest, family = poisson,  
data = gala)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	3.548e+00	3.933e-02	90.211	< 2e-16 ***
Area	-5.529e-05	1.890e-05	-2.925	0.00344 **
Elevation	1.588e-03	5.040e-05	31.502	< 2e-16 ***
Nearest	5.921e-03	1.466e-03	4.039	5.38e-05 ***

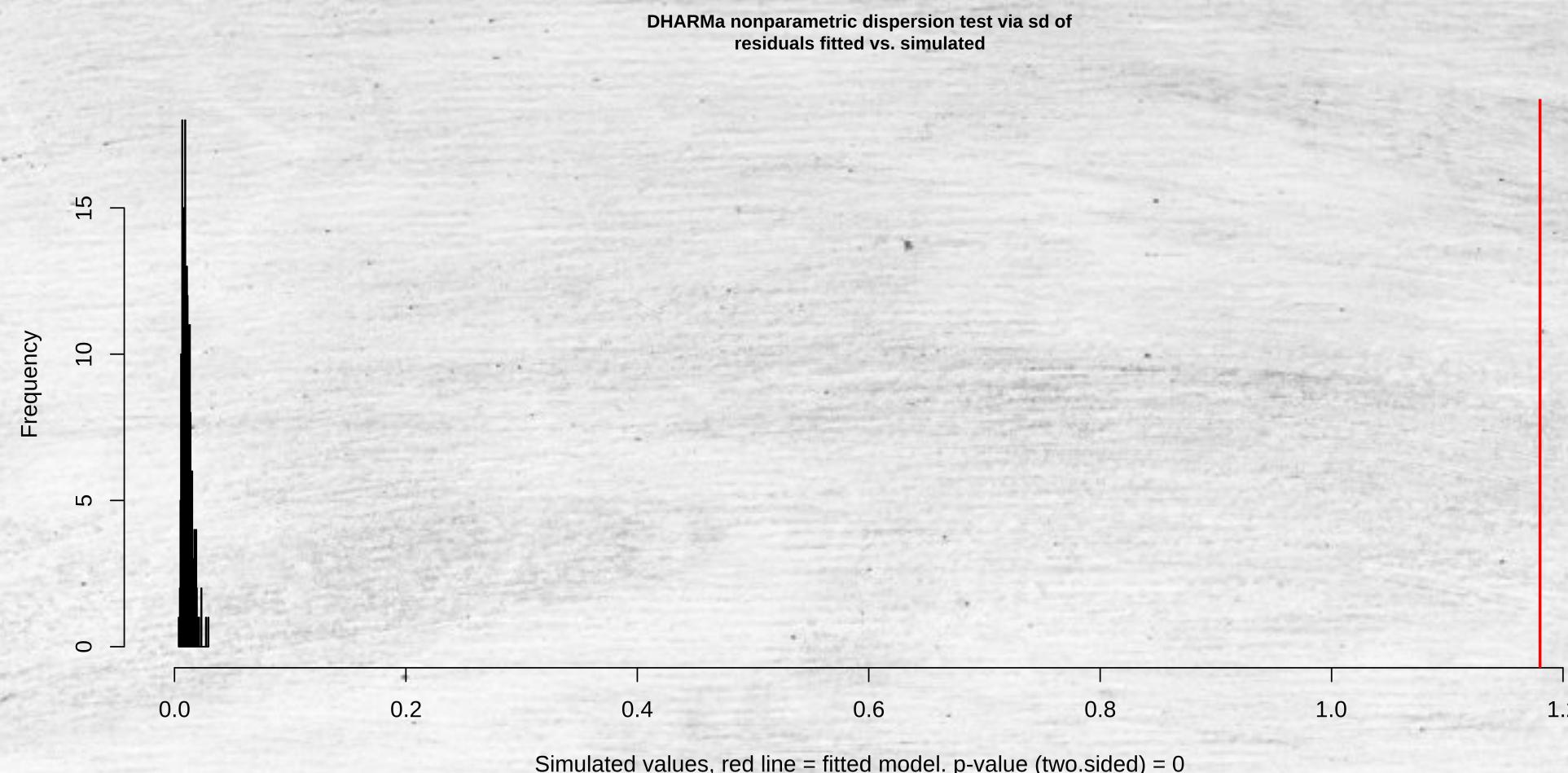
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

```
1 res <- simulateResiduals(modpl)
2 testDispersion(res)
```

DHARMA nonparametric dispersion test via sd of residuals fitted vs.
simulated

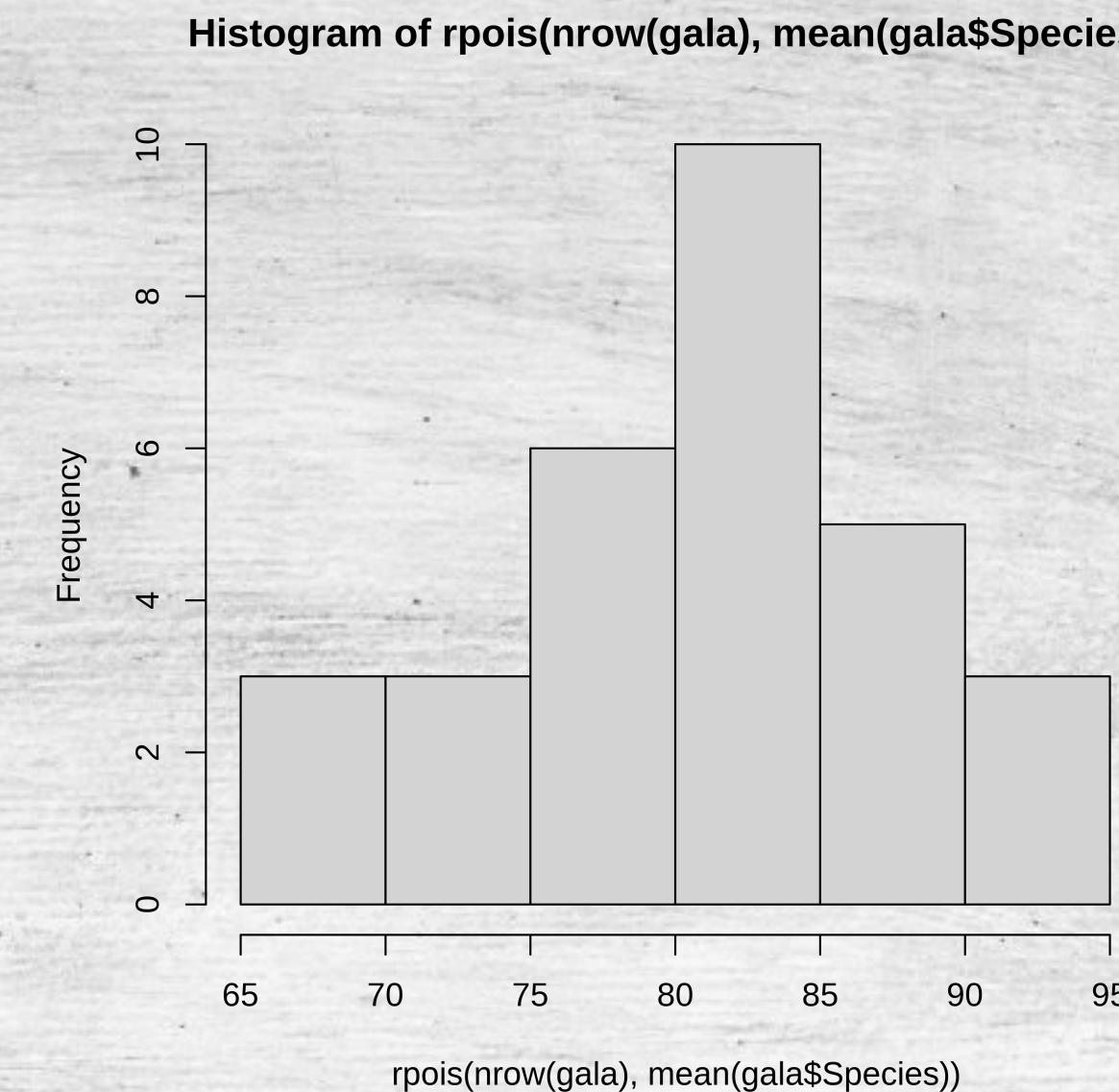
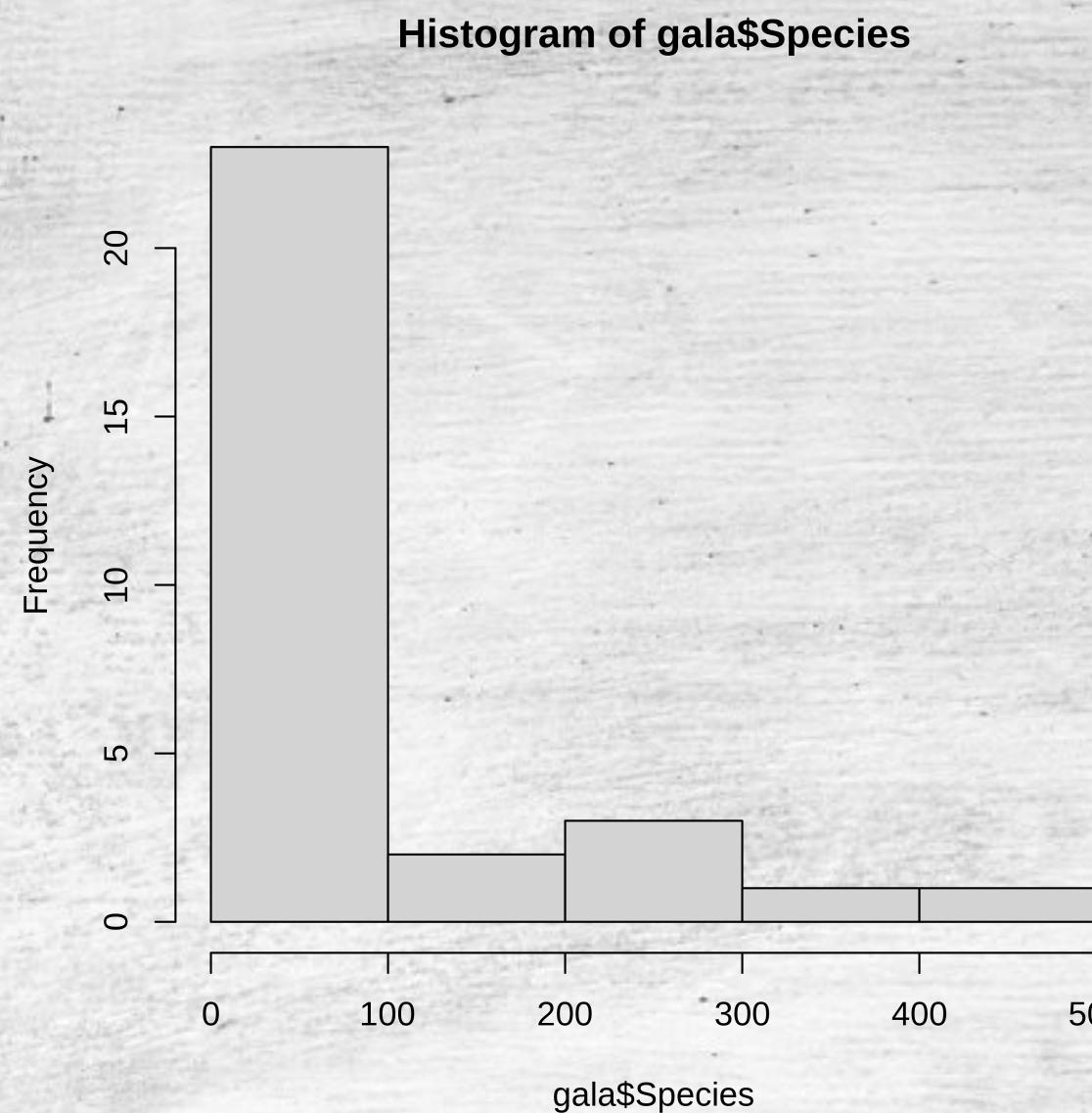
```
data: simulationOutput
dispersion = 110.32, p-value < 2.2e-16
alternative hypothesis: two.sided
```



```
1 c(mean(gala$Species), var(gala$Species))
```

```
[1] 85.23333 13140.73678
```

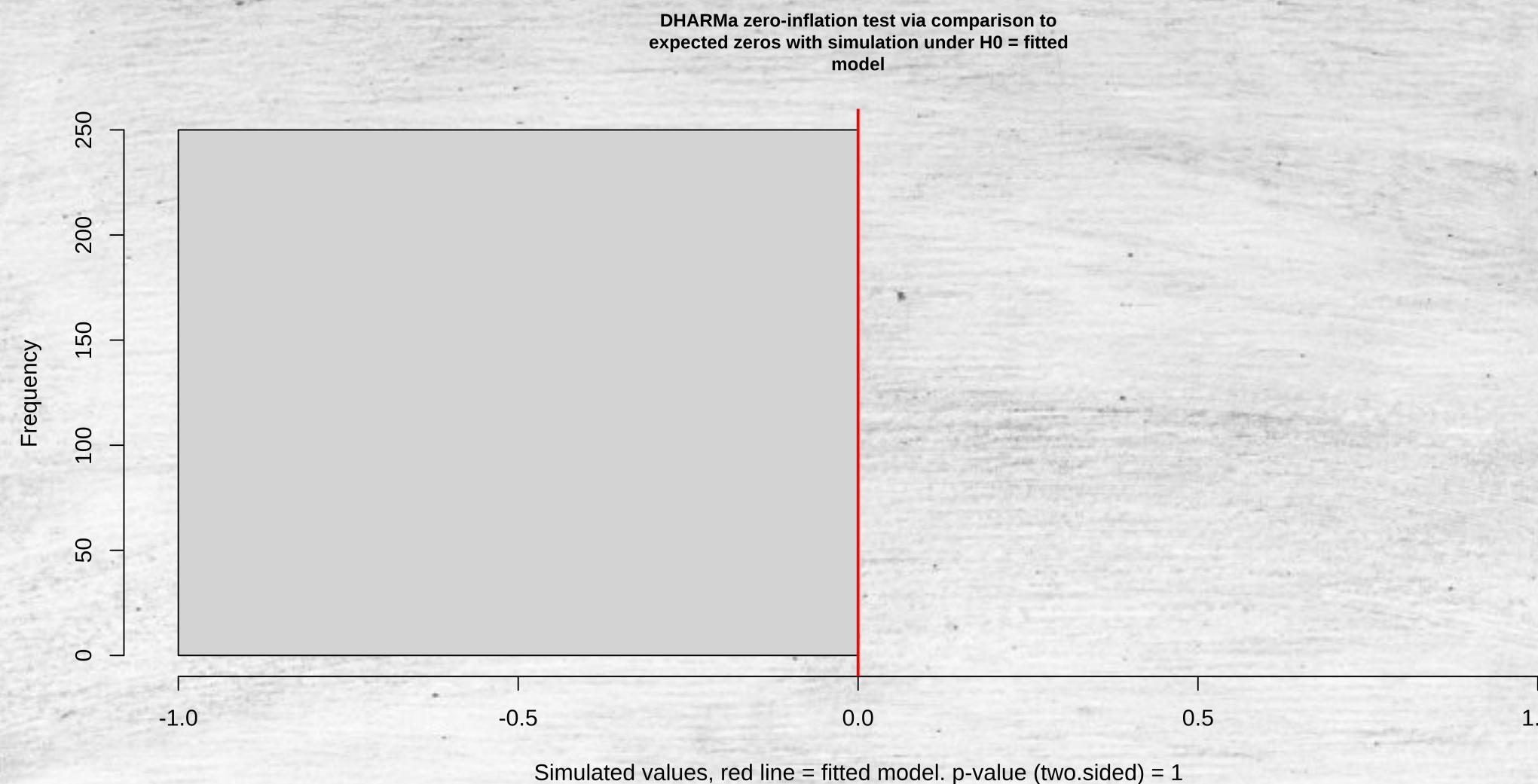
```
1 par(mfrow = c(1, 2))
2 hist(gala$Species)
3 hist(rpois(nrow(gala), mean(gala$Species)))
```



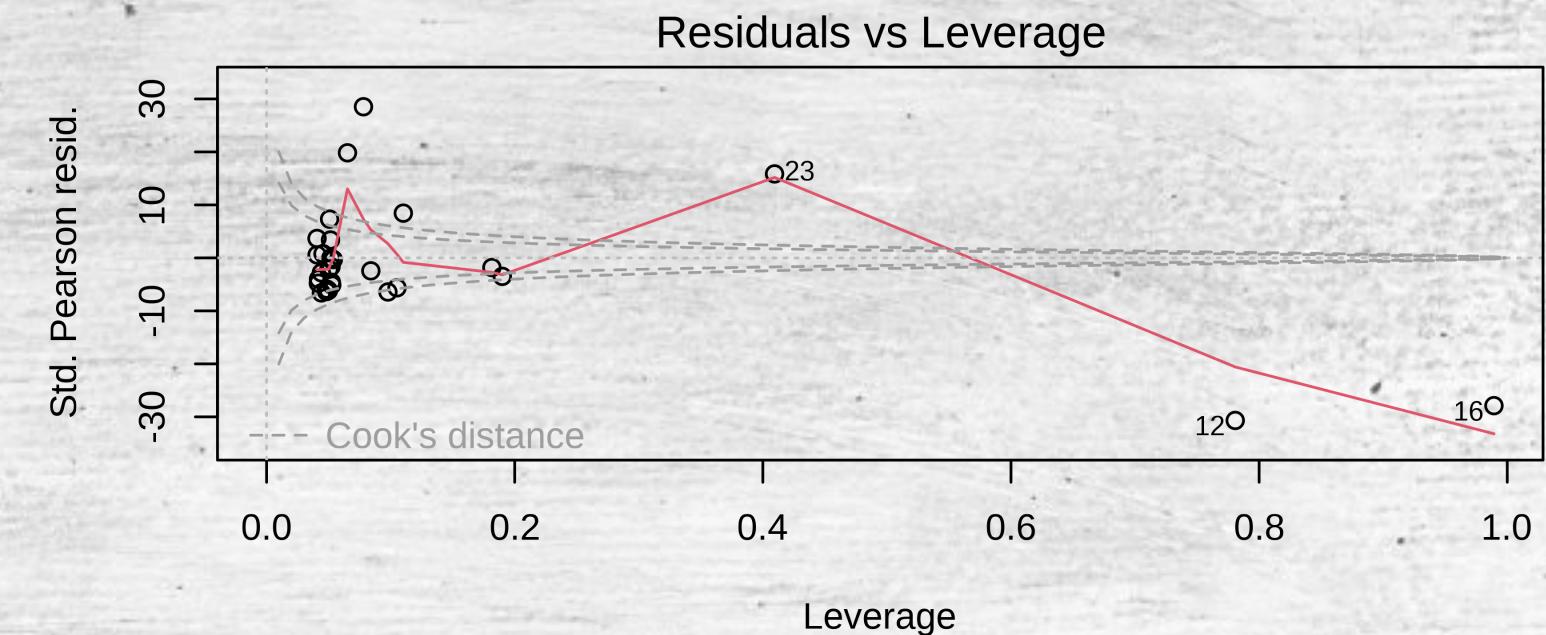
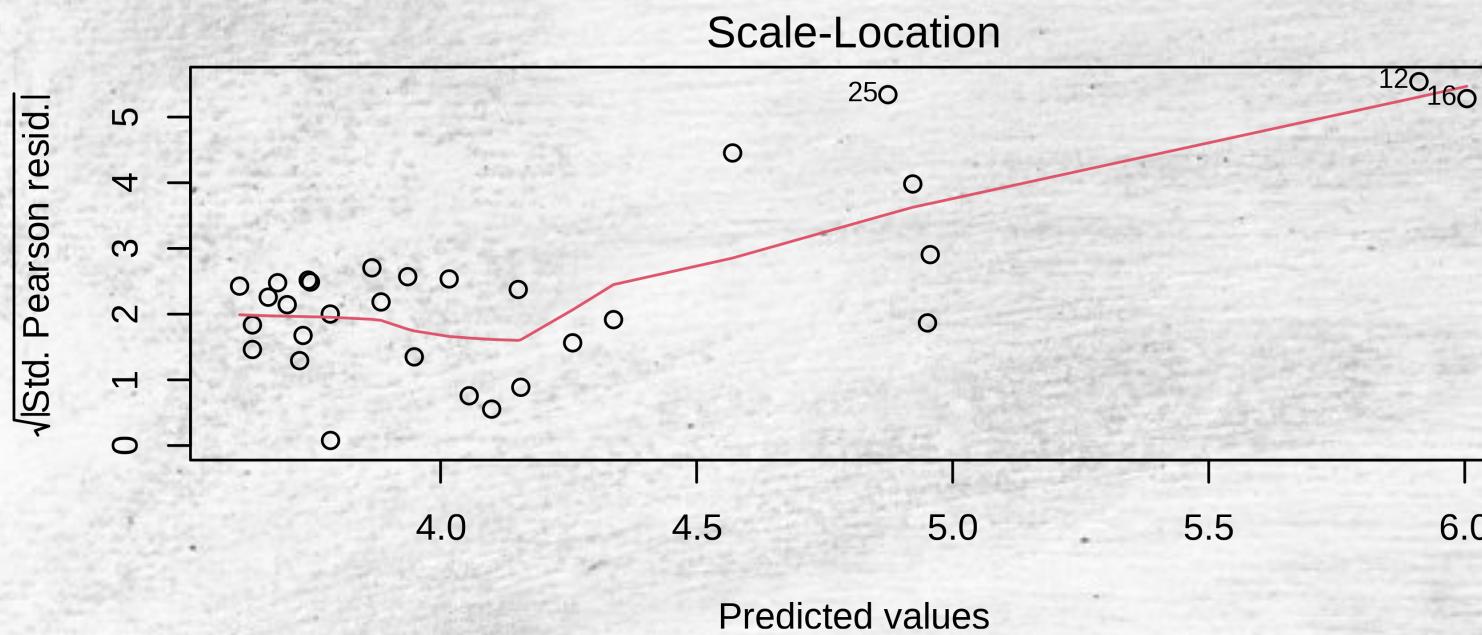
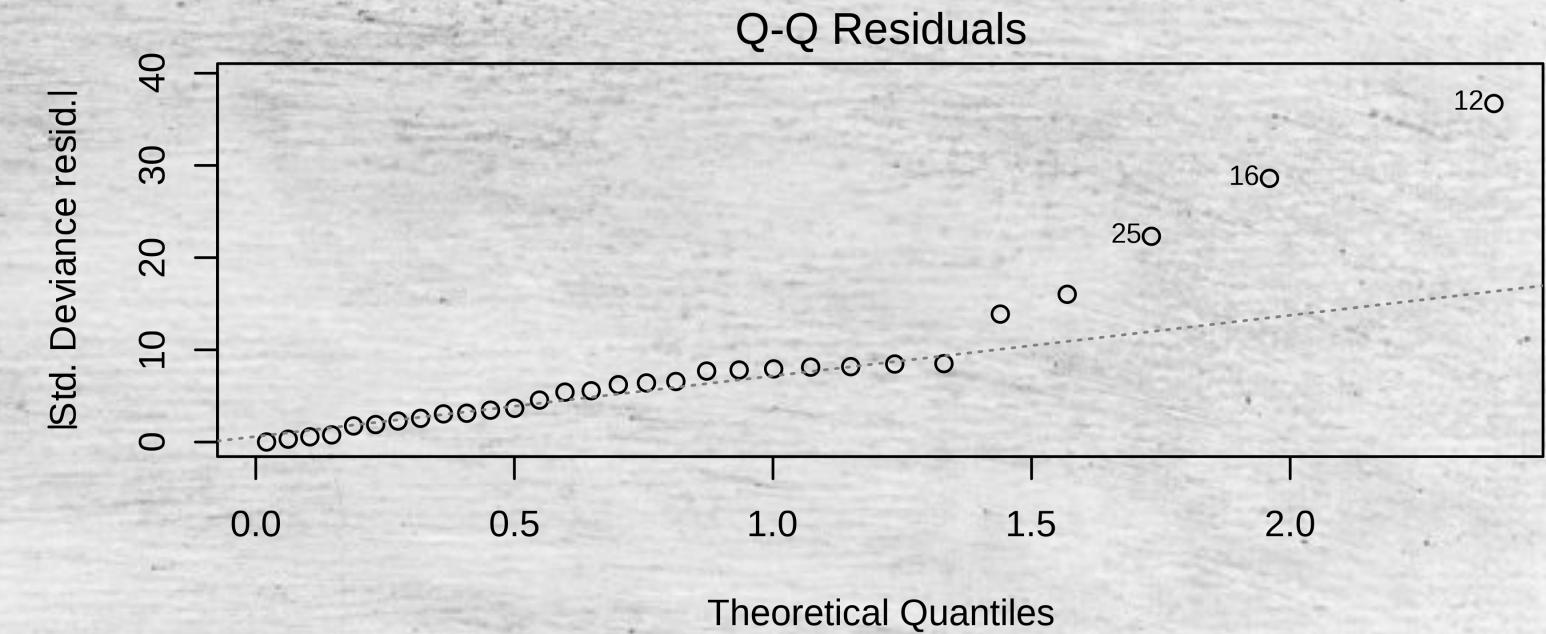
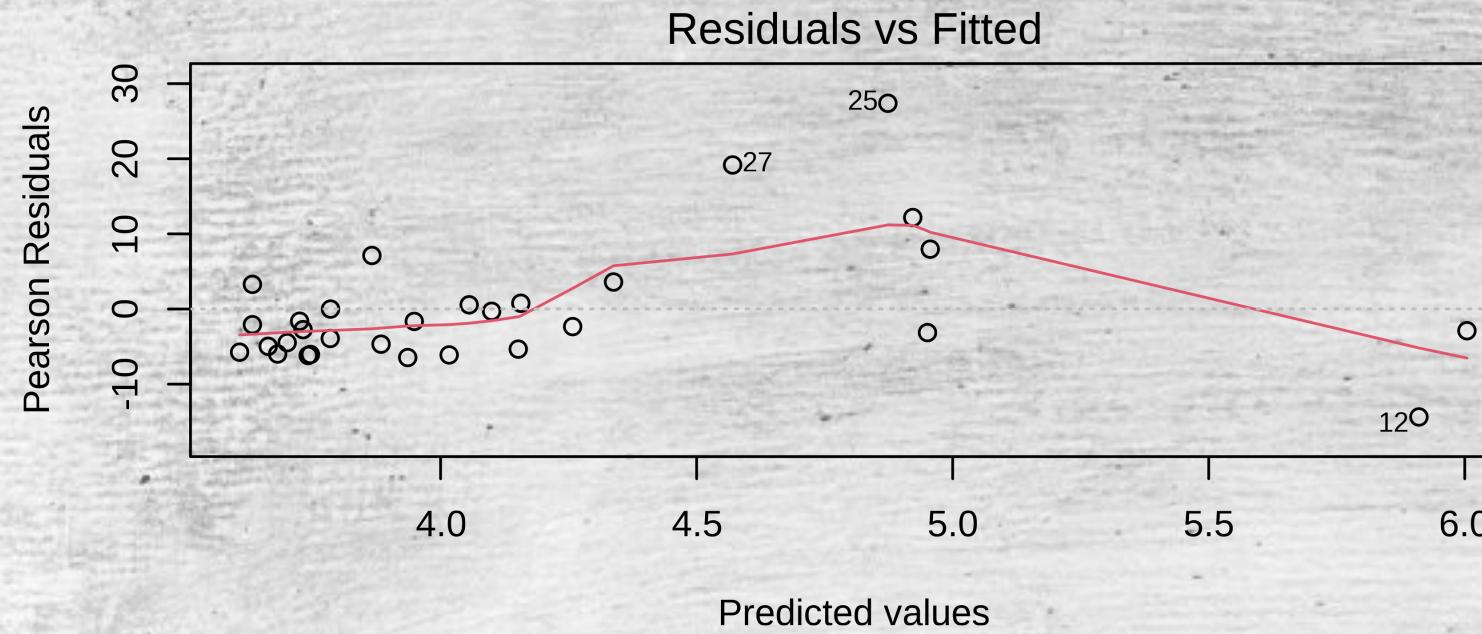
```
1 testZeroInflation(res)
```

DHARMA zero-inflation test via comparison to expected zeros with simulation under H0 = fitted model

```
data: simulationOutput  
ratioObsSim = NaN, p-value = 1  
alternative hypothesis: two.sided
```



```
1 par(mfrow = c(2, 2))  
2 plot(modpl)
```



Happy Coding

