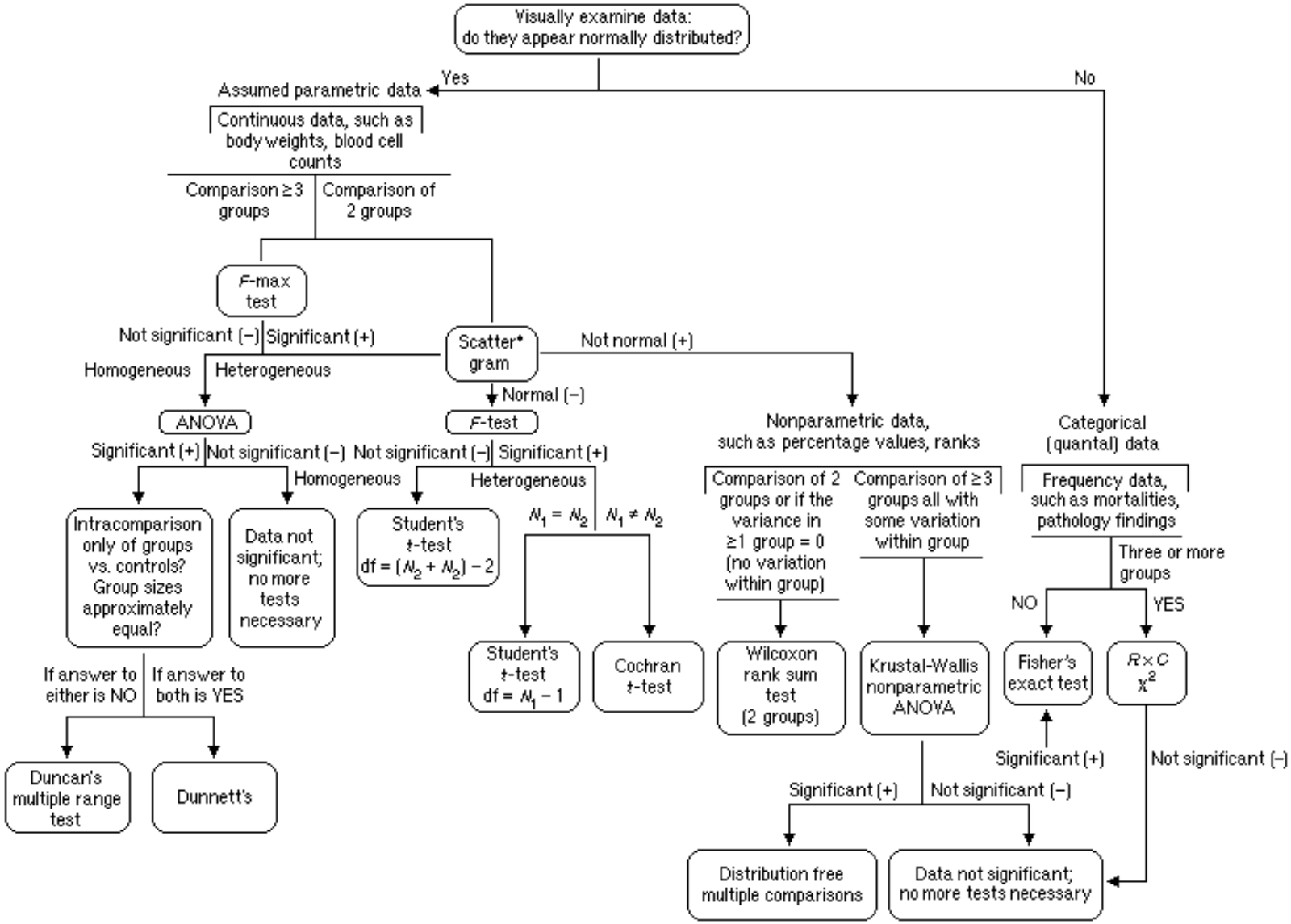
Linear, Generalized, and Mixed/Multilevel models - an introduction with R

Francisco Rodriguez-Sanchez (@frod_san)

December 2014 - January 2015

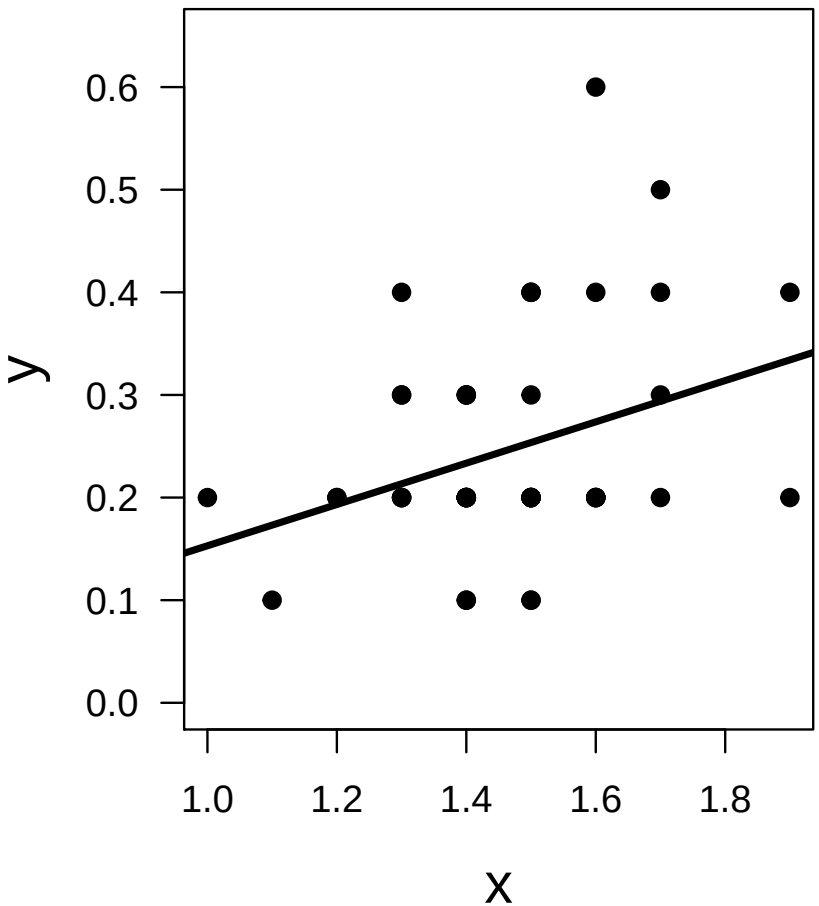
Modern statistics are easier than this



Our overarching regression framework

$$y_i = a + bx_i + \varepsilon_i$$

$$\varepsilon_i \sim N\left(0, \sigma^2\right)$$



Data

y = response variable

x = predictor

Parameters

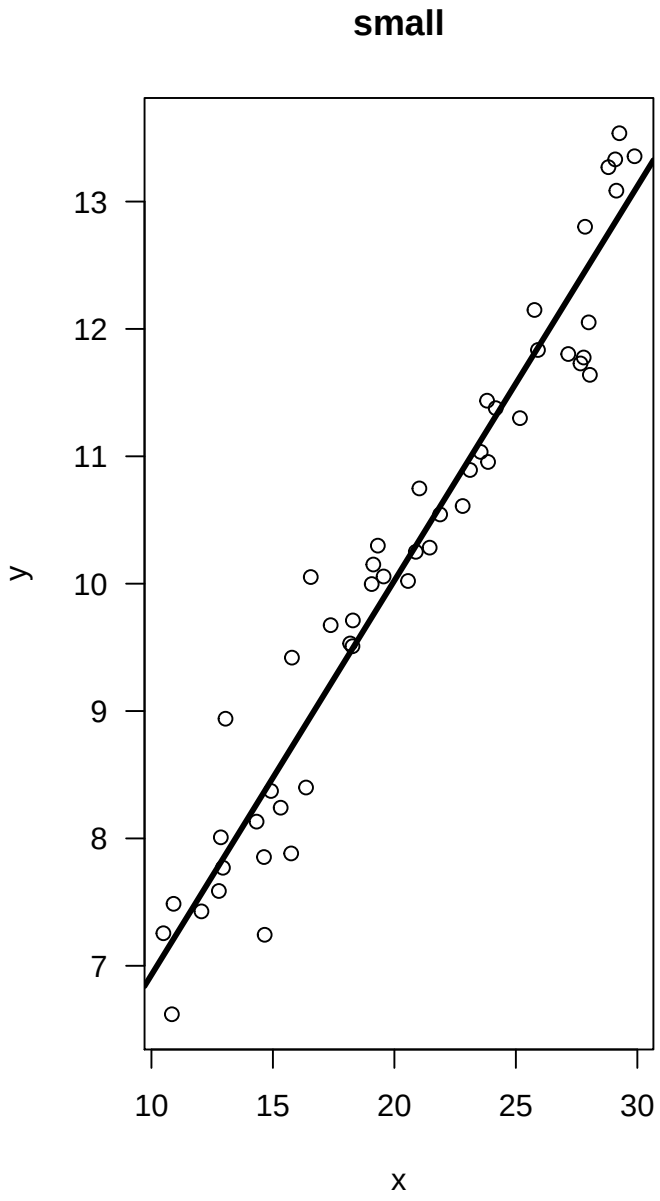
a = intercept

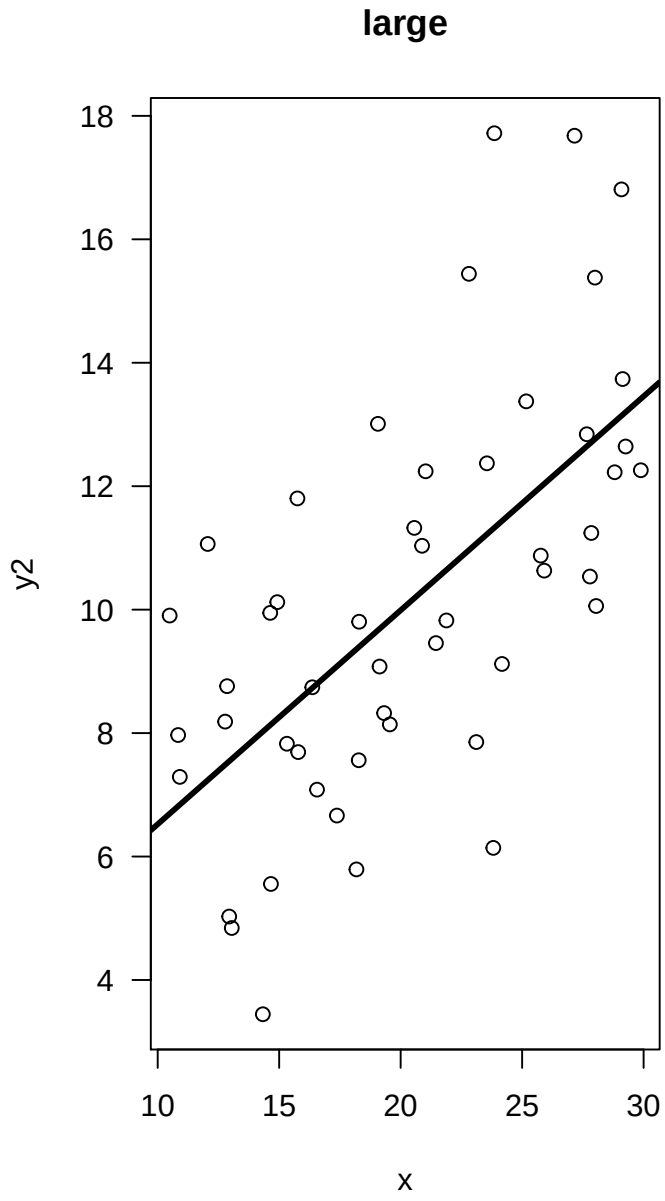
b = slope

σ = residual variation

ε = residuals

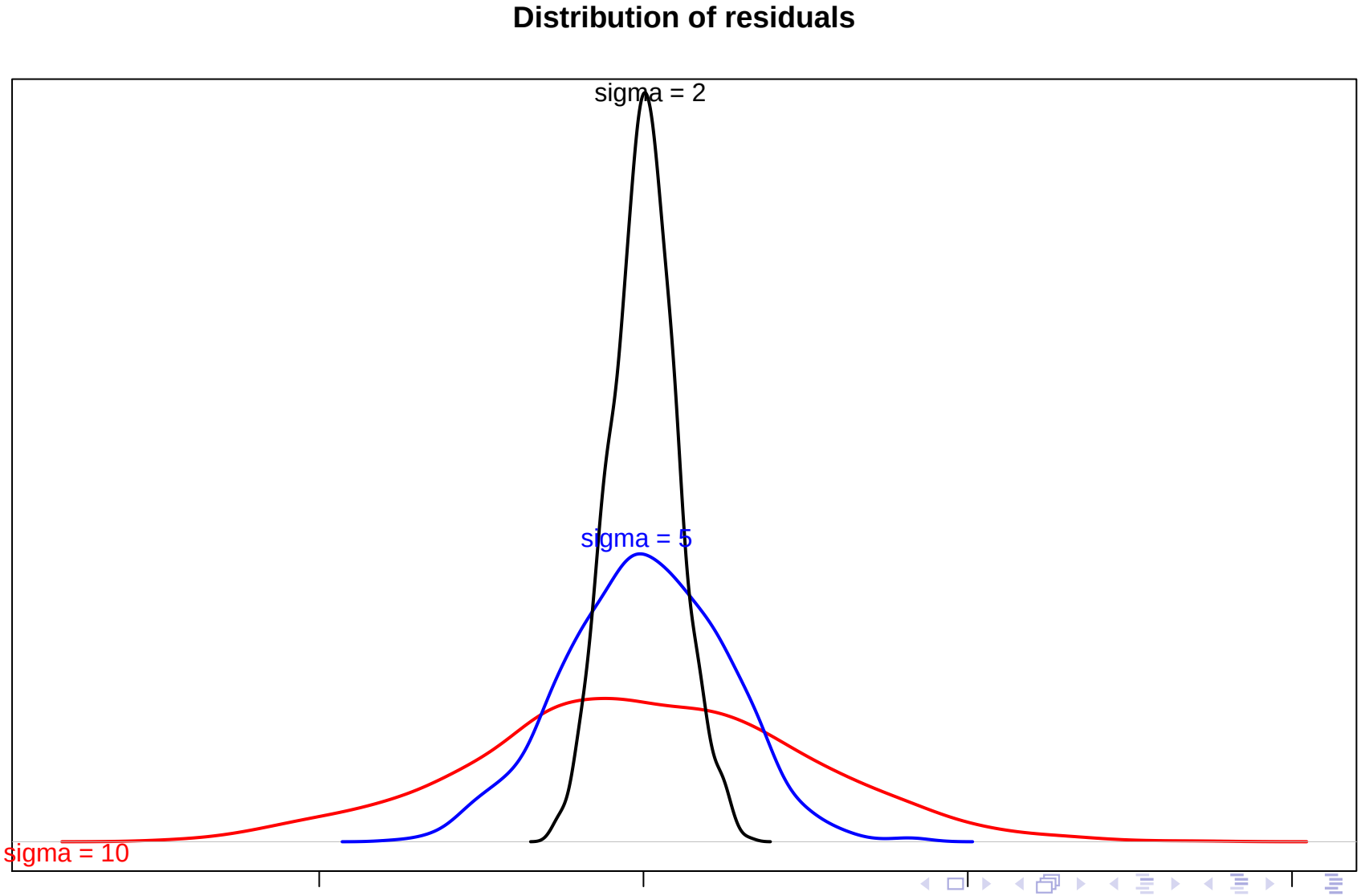
Residual variation (error)





Residual variation

$$\varepsilon_i \sim N\left(0, \sigma^2\right)$$

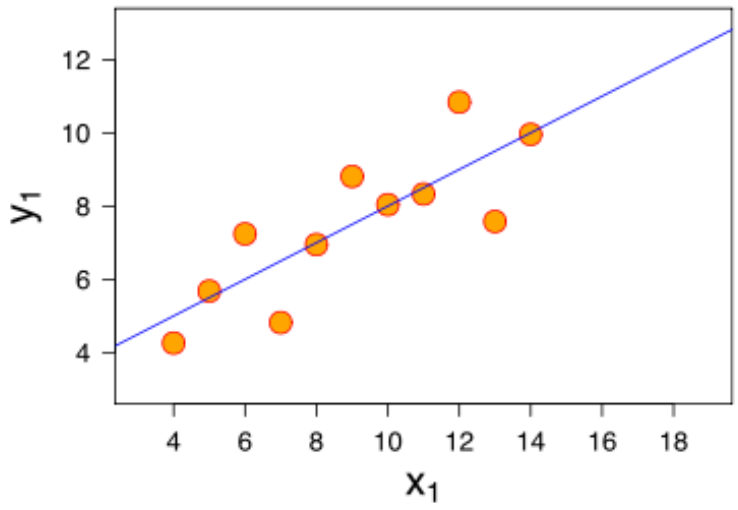


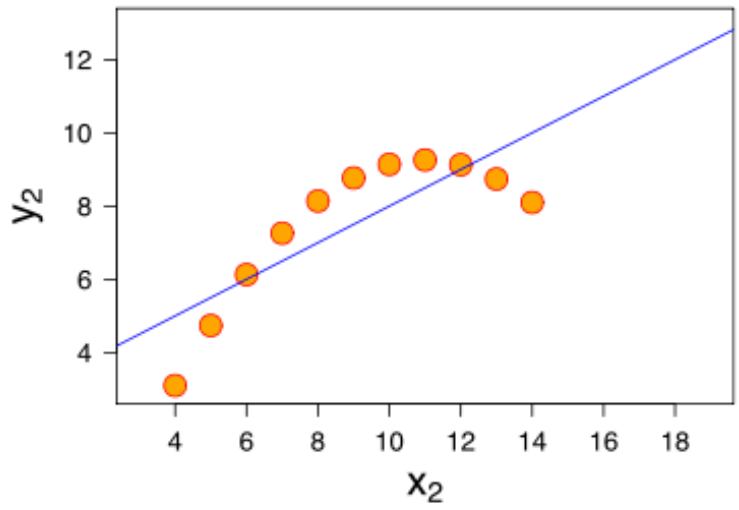
Quick refresher of linear models

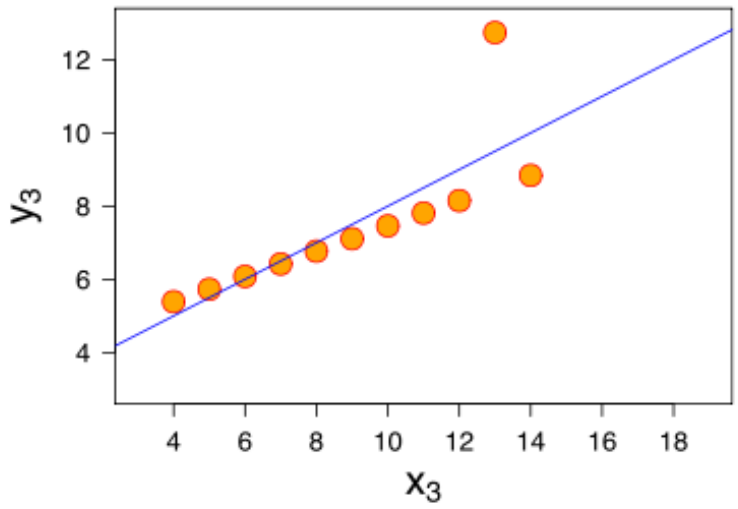
Go to `http://vincentarelbundock.github.io/Rdatasets/datasets.html` and download `iris` dataset.

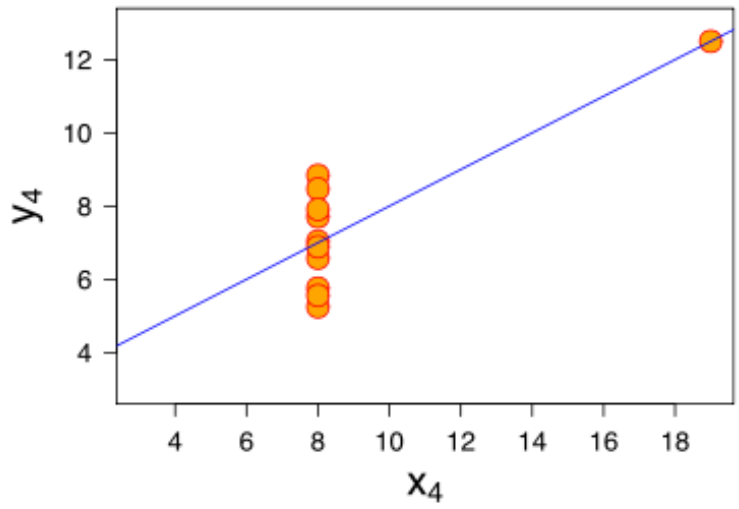
Q: What is the relationship between petal width and length in *Iris setosa*?

Always plot your data first!





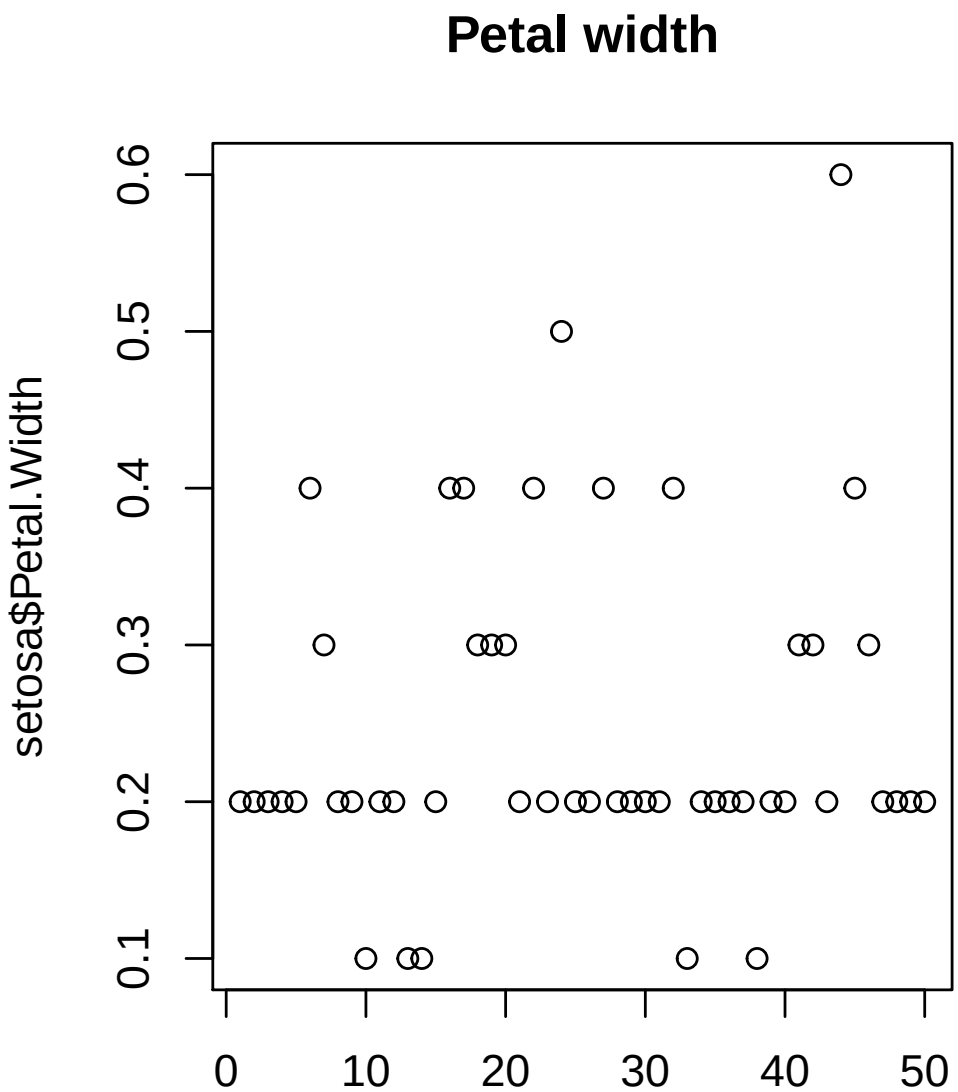




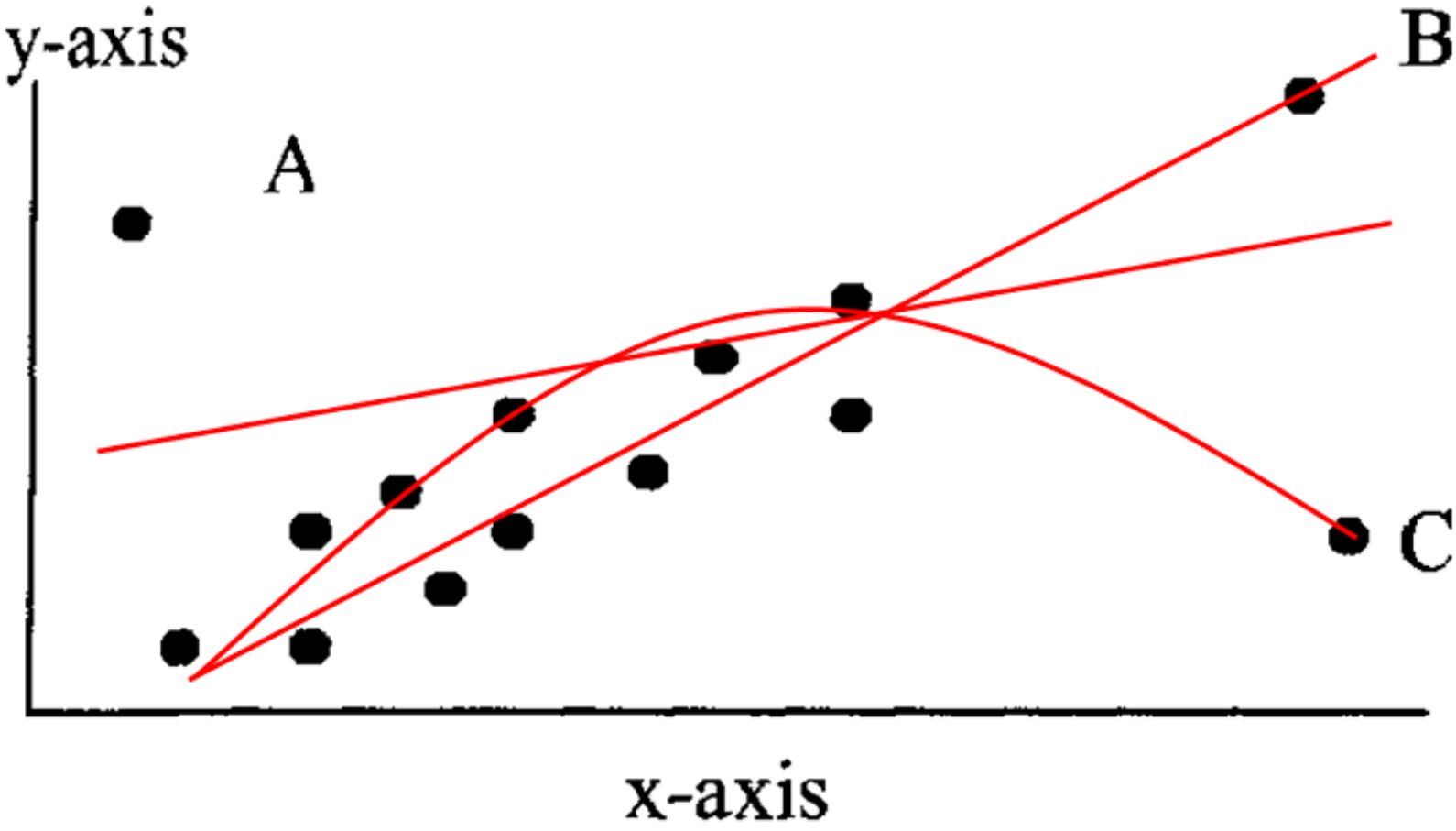
Exploratory Data Analysis (EDA)

Outliers

```
plot(setosa$Petal.Width, main = "Petal width")
```

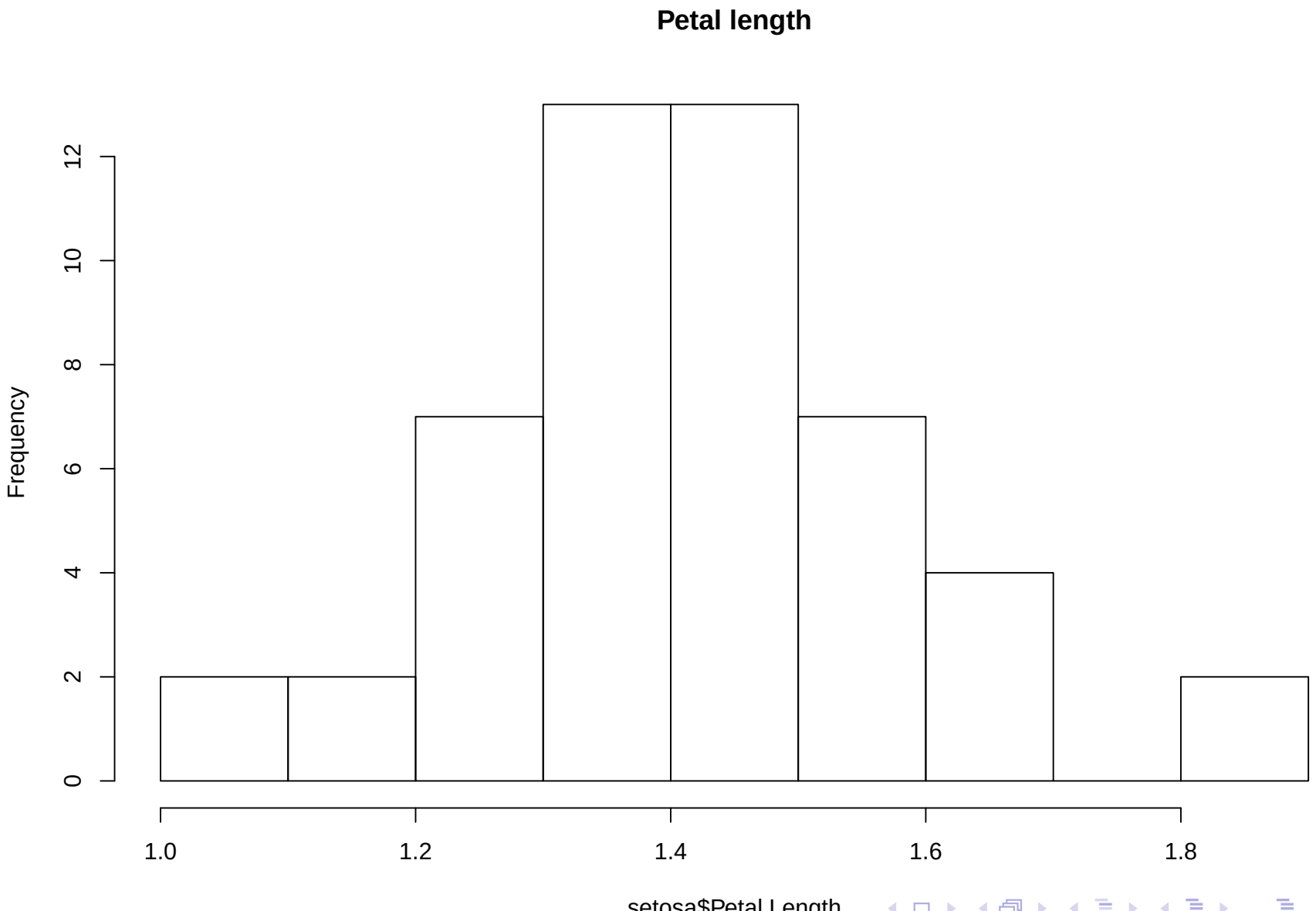


Outliers impact on regression



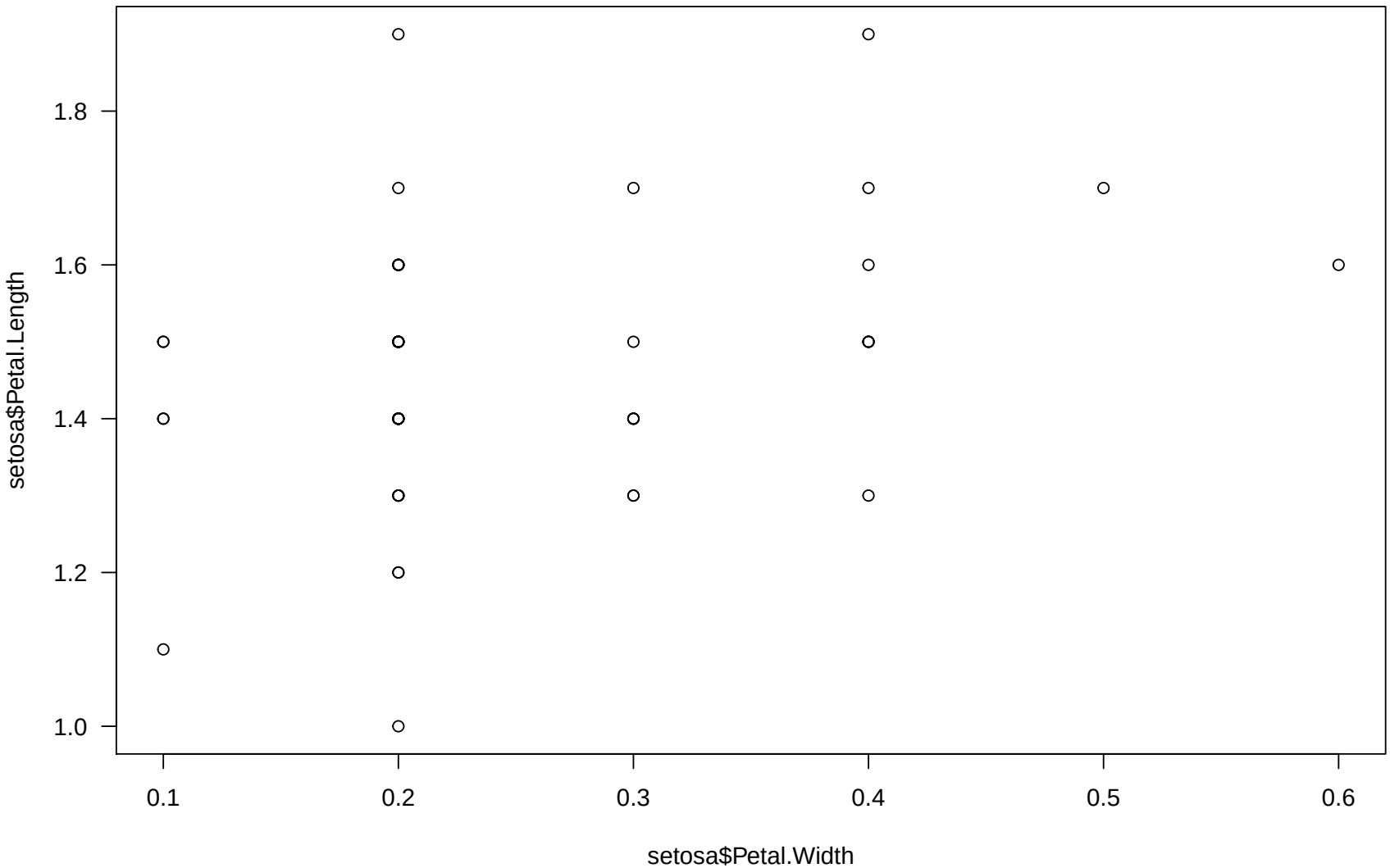
Histogram

```
hist(setosa$Petal.Length, main = "Petal length")
```



Scatterplot

```
plot(setosa$Petal.Width, setosa$Petal.Length, las = 1)
```



Now fit model

Hint: `lm`

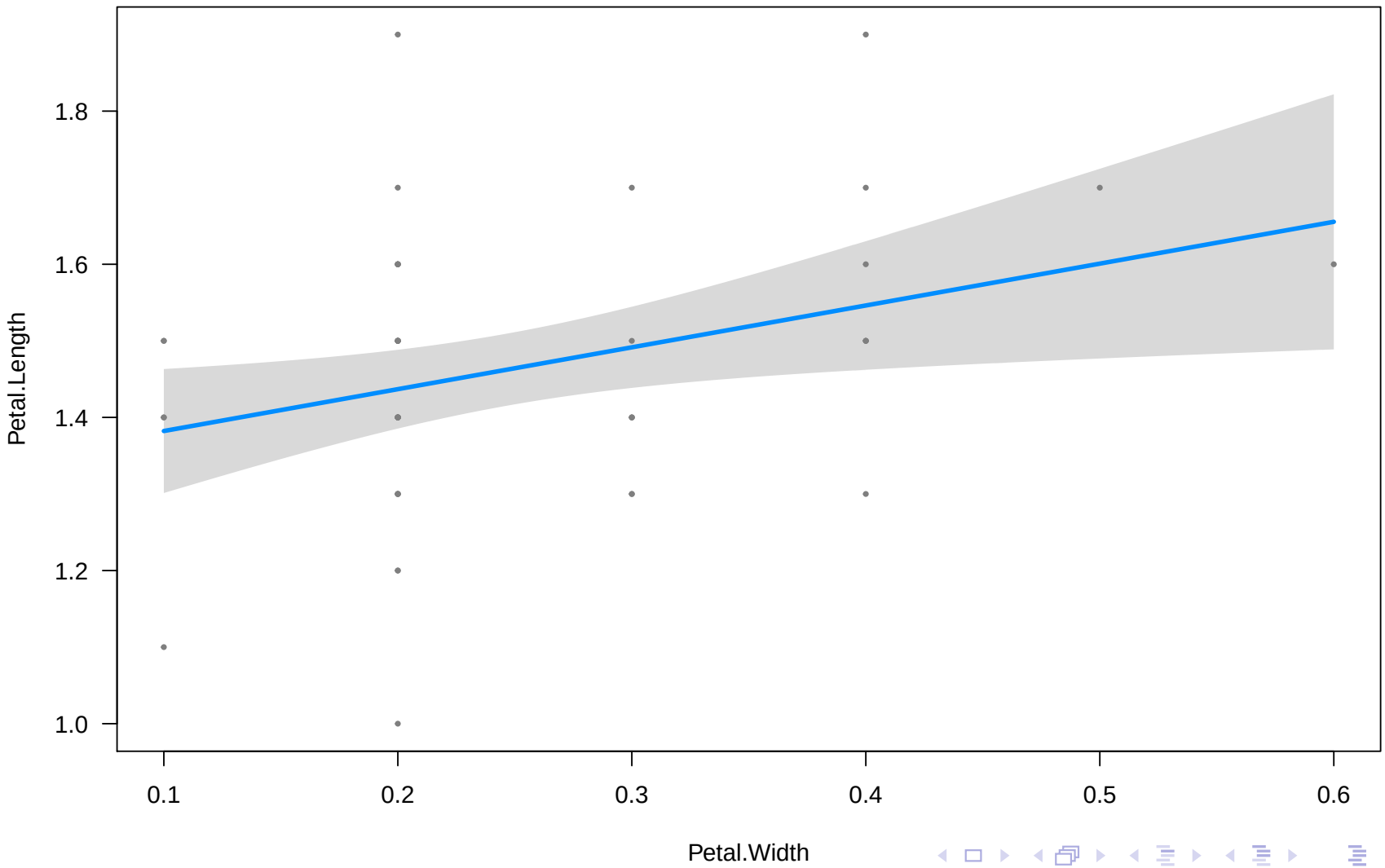
```
m1 <- lm(Petal.Length ~ Petal.Width, data = setosa)
```

What does this mean?

```
Call:
lm(formula = Petal.Length ~ Petal.Width, data = setosa)

Residuals:
     Min       1Q   Median       3Q      Max
-0.43686 -0.09151 -0.03686  0.09018  0.46314

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.32756    0.05996  22.141   <2e-16 ***
Petal.Width  0.54649    0.22439   2.435   0.0186 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '

Residual standard error: 0.1655 on 48 degrees of freedom
Multiple R-squared:   0.11, Adjusted R-squared:  0.09144
F-statistic: 5.931 on 1 and 48 DF,  p-value: 0.01864
```

Plot model

```
library(visreg)
visreg(m1)
```



Linear model assumptions

- Linearity (transformations, GAM. . .)

Linear model assumptions

- Linearity (transformations, GAM. . .)
- Residuals:

Linear model assumptions

- Linearity (transformations, GAM. . .)
- Residuals:
  - Independent

Linear model assumptions

- Linearity (transformations, GAM. . .)
- Residuals:
  - Independent
  - Equal variance

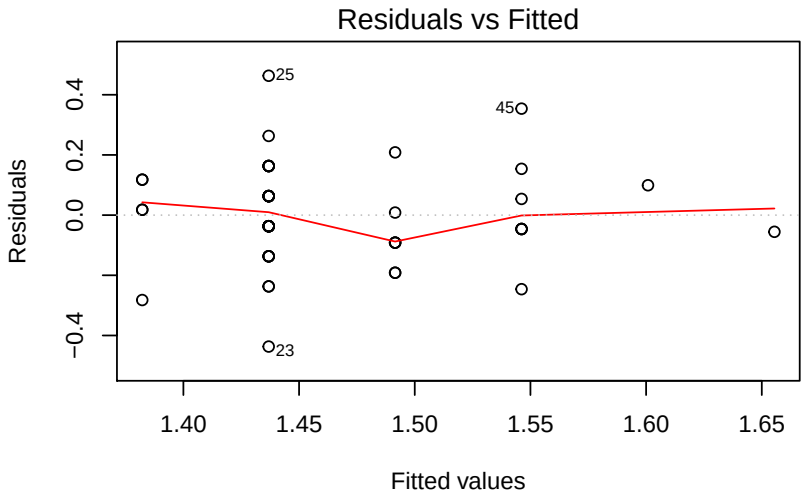
Linear model assumptions

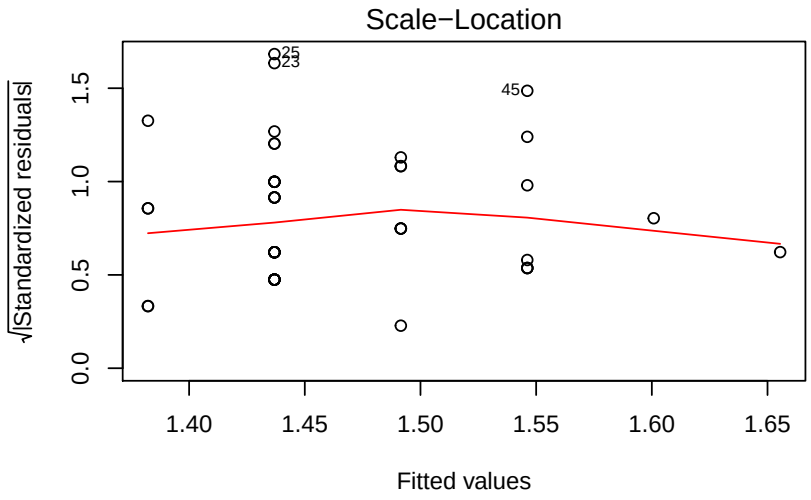
- Linearity (transformations, GAM. . .)
- Residuals:
  - Independent
  - Equal variance
  - Normal

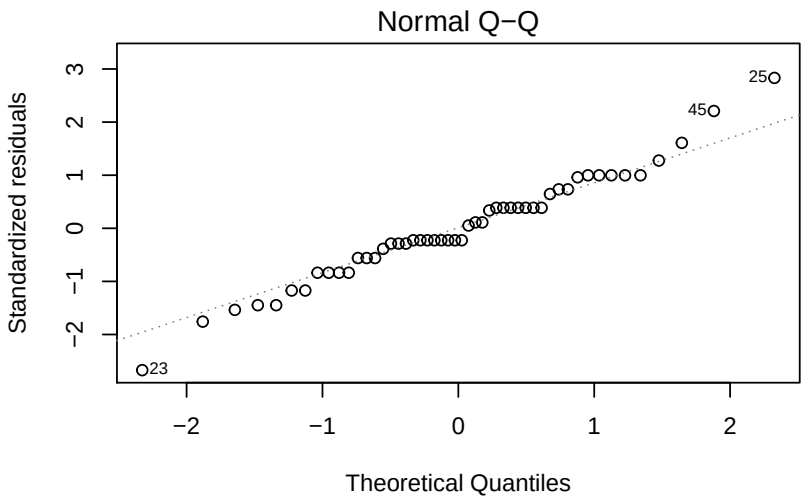
Linear model assumptions

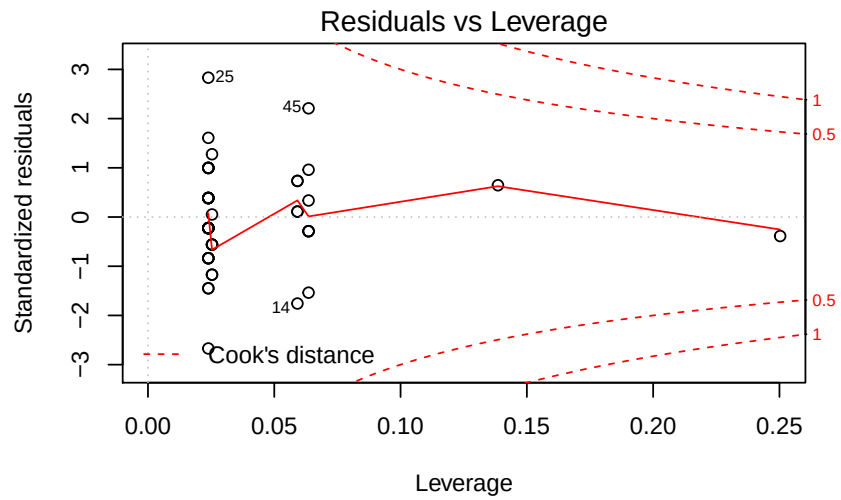
- Linearity (transformations, GAM. . .)
- Residuals:
  - Independent
  - Equal variance
  - Normal
- No measurement error in predictors

Model checking: residuals

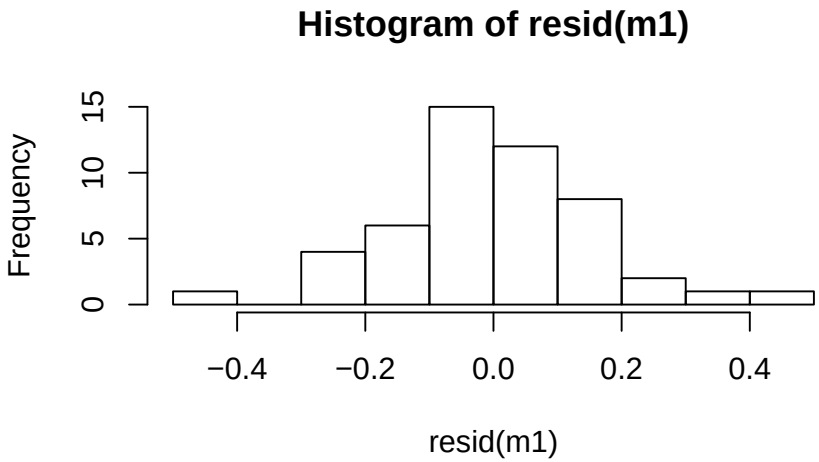








Are residuals normal?

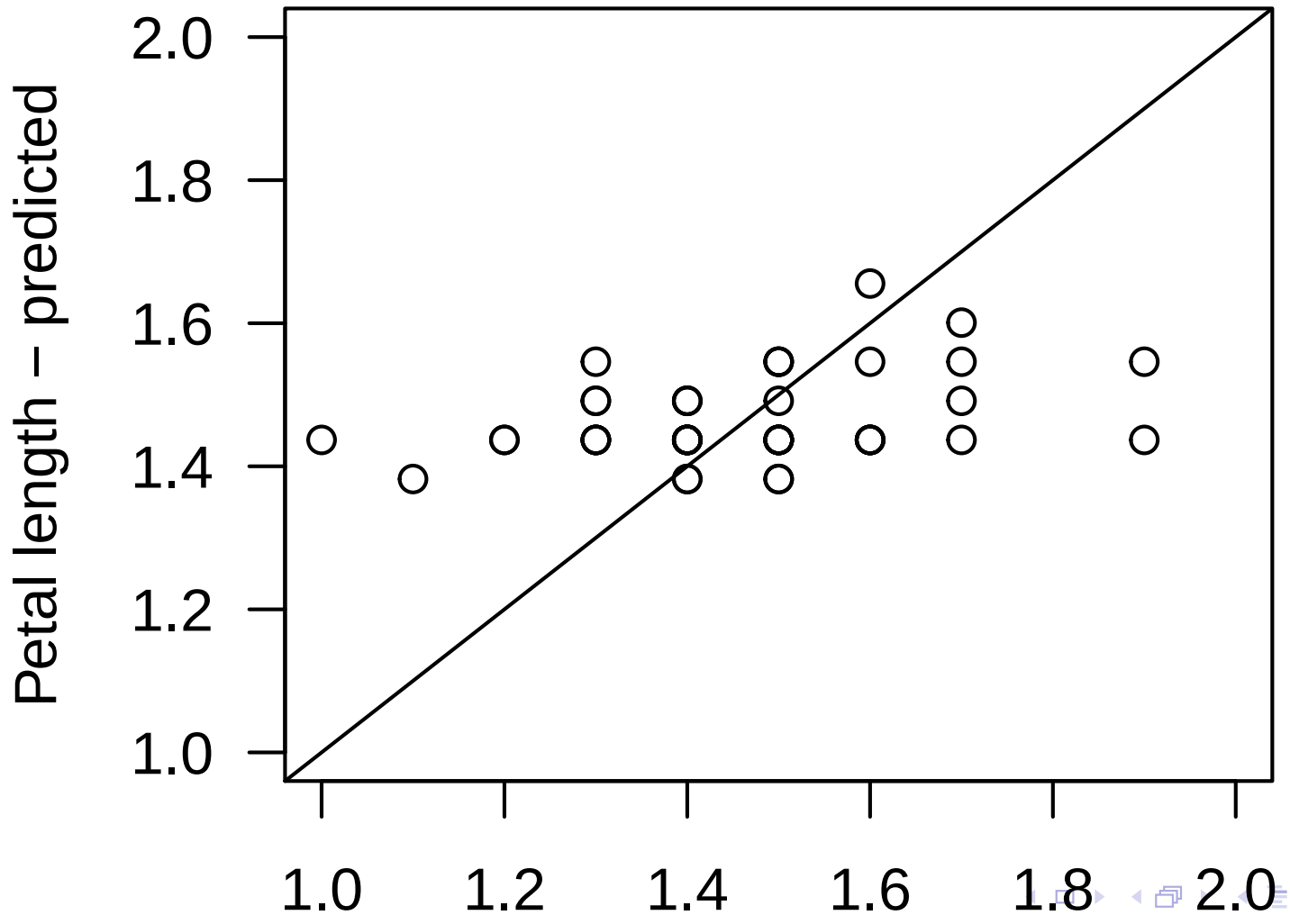



```
lm(formula = Petal.Length ~ Pe
             coef.est coef.se
(Intercept) 1.33     0.06
Petal.Width 0.55     0.22
---
n = 50, k = 2
residual sd = 0.17, R-Squared
```

SD of residuals $= 0.16$ coincides with estimate of `sigma`.

How good is the model in predicting petal length? | Observed vs Predicted values

```
plot(setosa$Petal.Length, fitted(m1), las = 1, xlim = c(1,
    xlab = "Petal length - observed", ylab = "Petal length
```



Categorical predictors (factors)

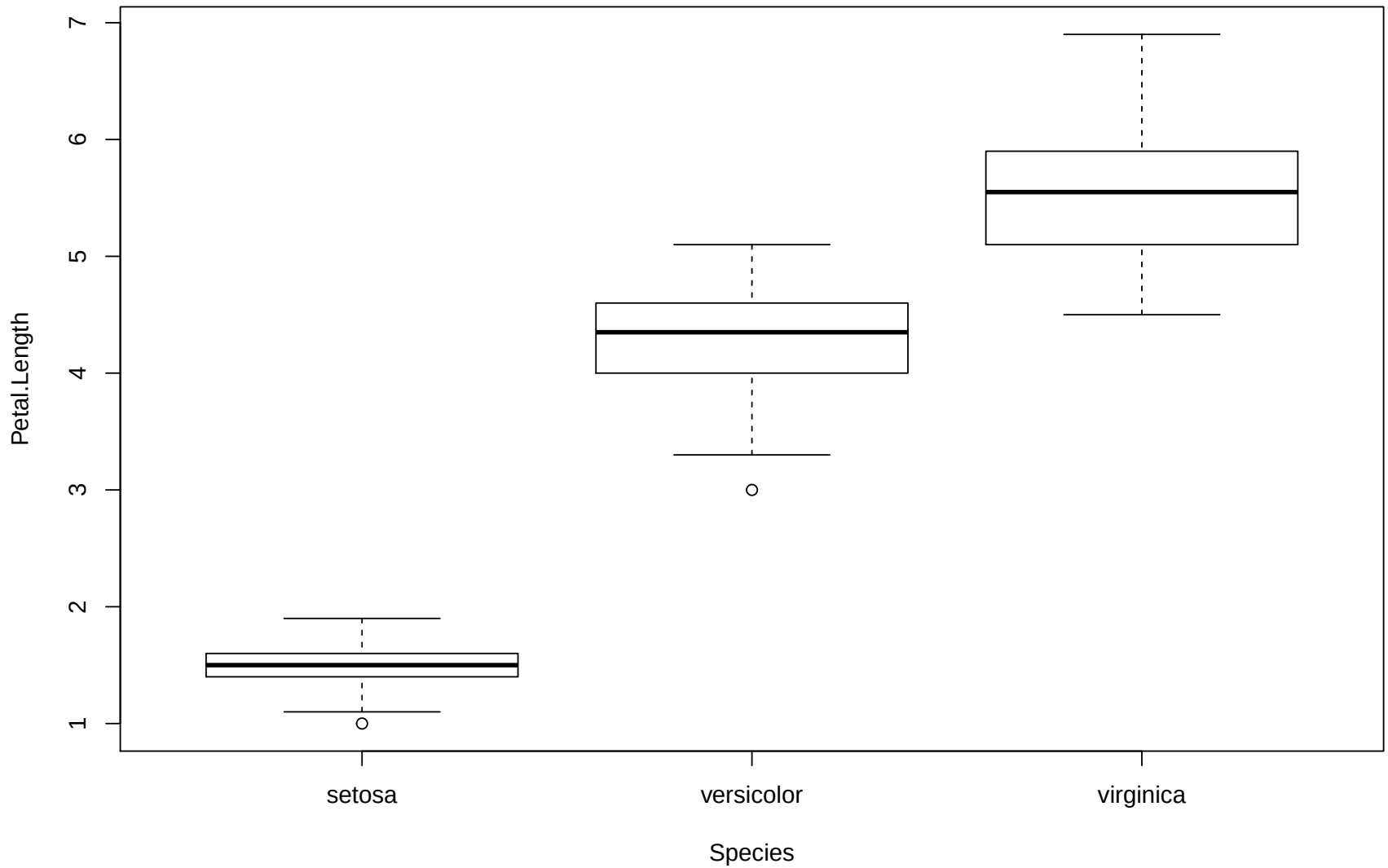
Q: Does petal length vary among *Iris* species?

$$y_i = a + bx_i + \varepsilon_i$$

$$y_i = a + b_{versicolor} + c_{virginica} + \varepsilon_i$$

EDA

```
plot(Petal.Length ~ Species, data = iris)
```



Model

```
m2 <- lm(Petal.Length ~ Species, data = iris)
```

```
Call:
lm(formula = Petal.Length ~ Species, data = iris)

Residuals:
    Min      1Q  Median      3Q     Max
-1.260 -0.258  0.038  0.240  1.348

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        1.46200    0.06086   24.02   <2e-16 ***
Speciesversicolor  2.79800    0.08607   32.51   <2e-16 ***
Speciesvirginica   4.09000    0.08607   47.52   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
```

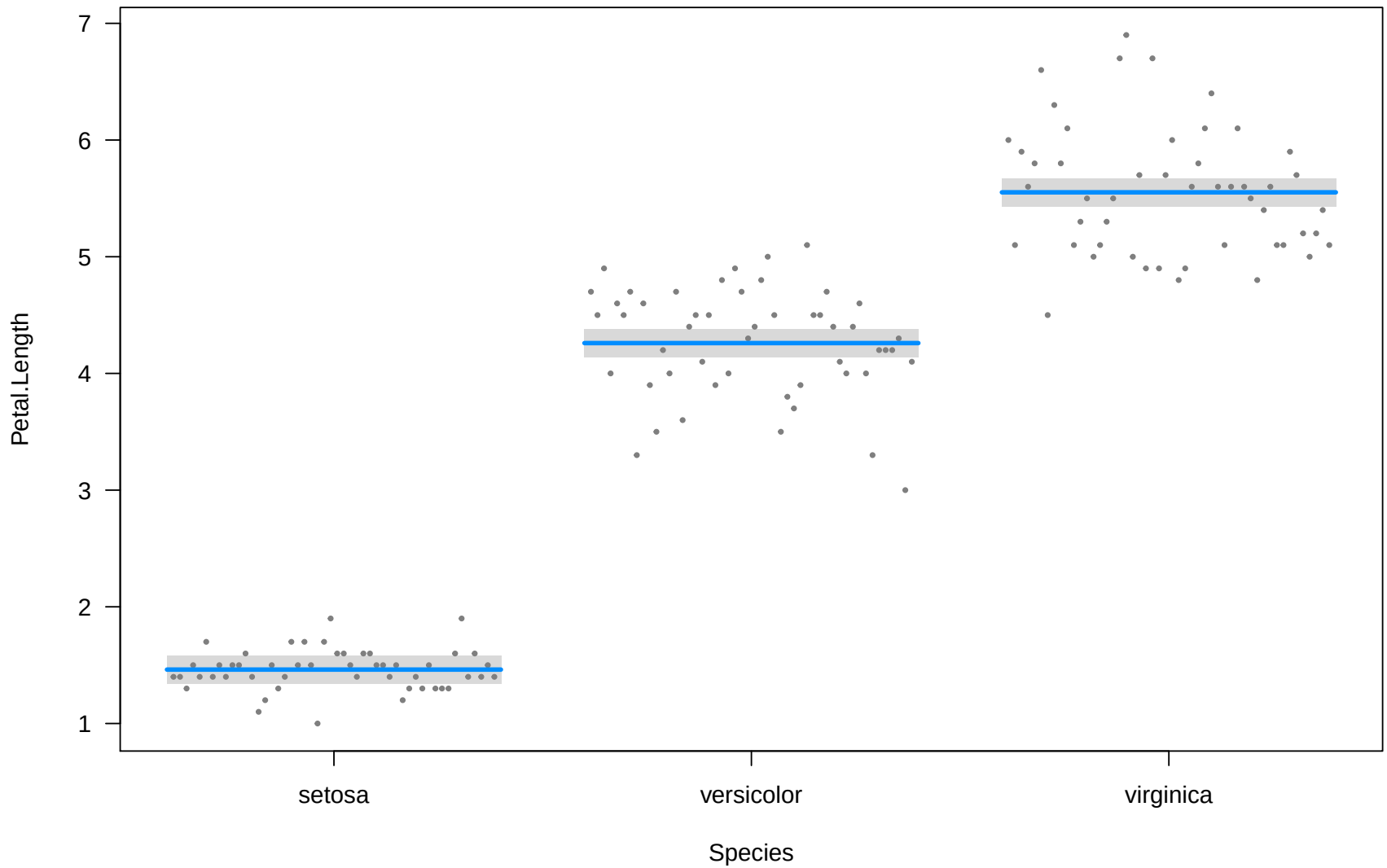
Alternatively, no intercept

```
m3 <- lm(Petal.Length ~ Species - 1, data = iris)
```

```
Call:
lm(formula = Petal.Length ~ Species - 1, data = iris)

Residuals:
    Min      1Q  Median      3Q     Max
-1.260 -0.258  0.038  0.240  1.348

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
Speciessetosa       1.46200    0.06086   24.02   <2e-16 ***
Speciesversicolor   4.26000    0.06086   70.00   <2e-16 ***
Speciesvirginica    5.55200    0.06086   91.23   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
```

Petal length differences across 3 *Iris* species

```
visreg(m3)
```



Generalised Linear Models (GLMs)

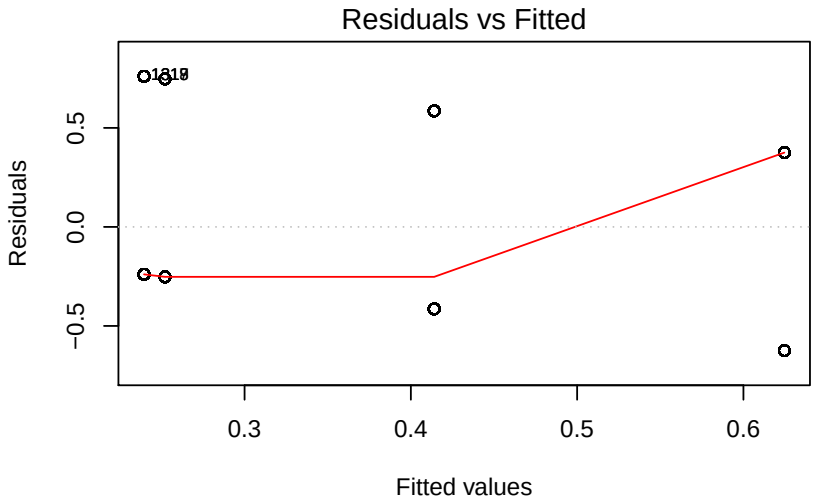
Q: Survival of passengers on the Titanic ~ Class

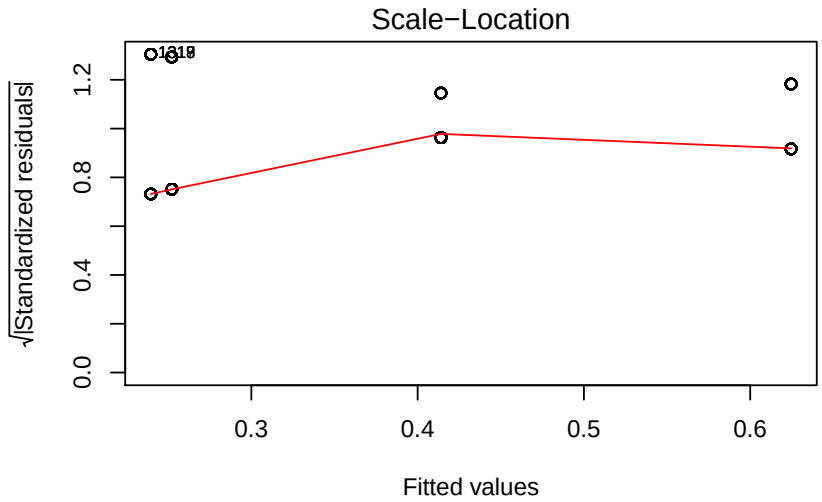
```
titanic <- read.csv("data-raw/titanic_long.csv")
```

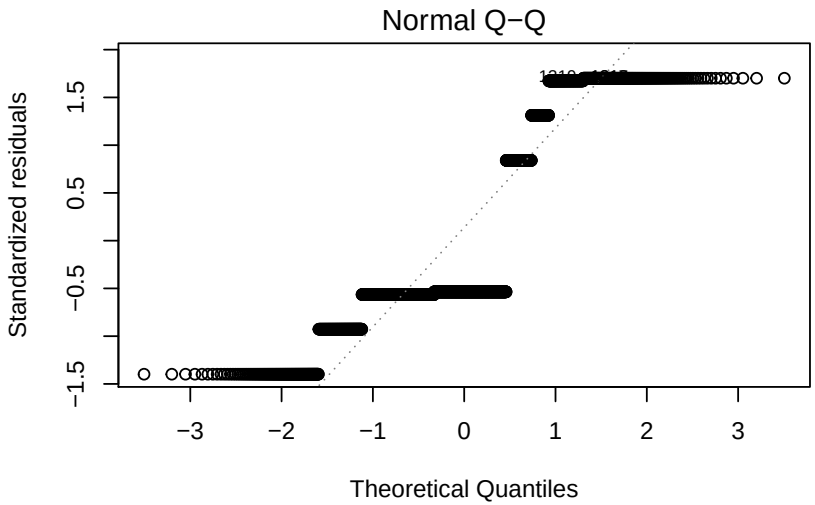
```
  class   age  sex survived
1 first adult male        1
2 first adult male        1
3 first adult male        1
4 first adult male        1
5 first adult male        1
6 first adult male        1
```

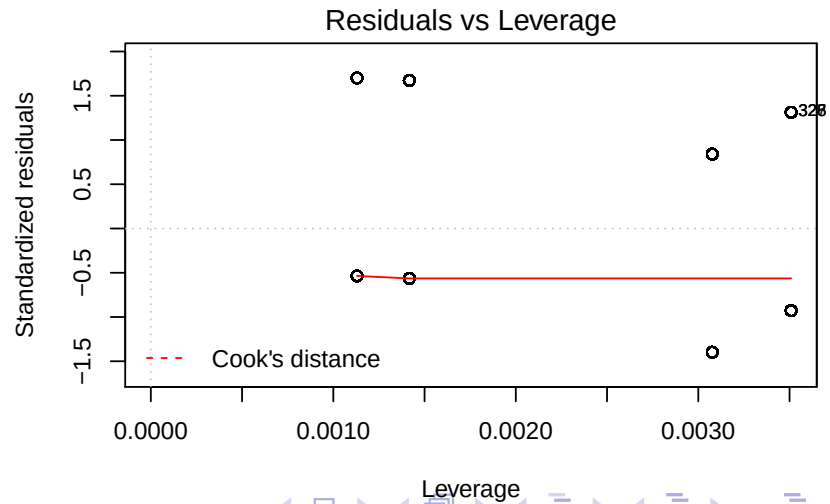
Let's fit linear model:

```
m5 <- lm(survived ~ class, data = titanic)
```

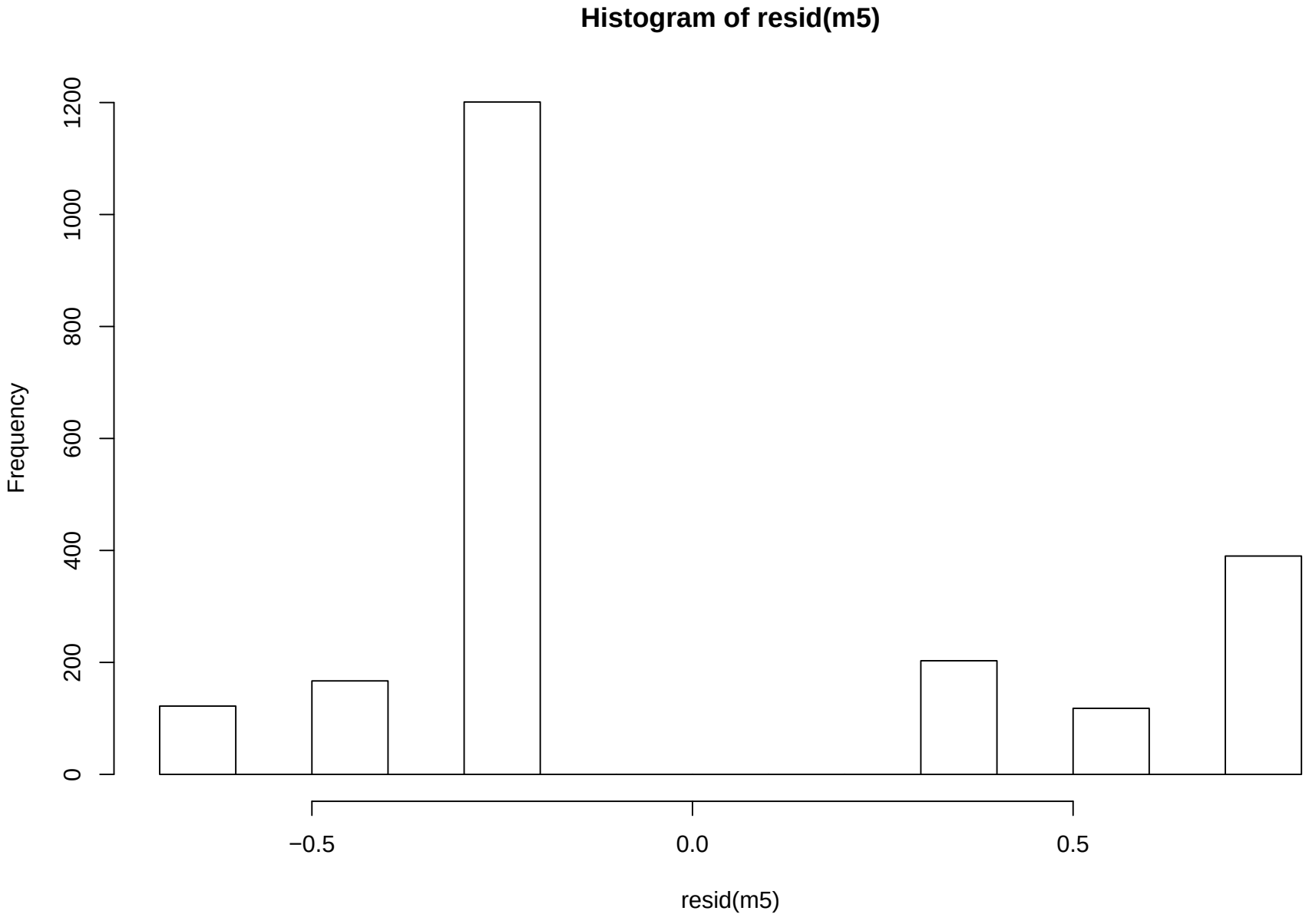








Weird residuals!



What if your residuals are clearly non-normal? And variance not constant (heteroscedasticity)?

- Binary variables (0/1)

What if your residuals are clearly non-normal? And variance not constant (heteroscedasticity)?

- Binary variables (0/1)
- Counts (0, 1, 2, 3, . . .)

Generalised Linear Models

1. **Response variable** - distribution `family`

Generalised Linear Models

1. **Response variable** - distribution `family`
   - Bernouilli - Binomial

Generalised Linear Models

1. **Response variable** - distribution `family`
   - Bernouilli - Binomial
   - Poisson

Generalised Linear Models

1. **Response variable** - distribution `family`
   - Bernouilli - Binomial
   - Poisson
   - Gamma

Generalised Linear Models

1. **Response variable** - distribution `family`
   - Bernouilli - Binomial
   - Poisson
   - Gamma
   - etc

Generalised Linear Models

1. **Response variable** - distribution `family`
   - Bernouilli - Binomial
   - Poisson
   - Gamma
   - etc
2. **Predictors** (continuous or categorical)

Generalised Linear Models

1. **Response variable** - distribution `family`
   - Bernouilli - Binomial
   - Poisson
   - Gamma
   - etc
2. **Predictors** (continuous or categorical)
3. **Link function**

Generalised Linear Models

1. **Response variable** - distribution `family`
   - Bernouilli - Binomial
   - Poisson
   - Gamma
   - etc
2. **Predictors** (continuous or categorical)
3. **Link function**
   - Gaussian: identity

Generalised Linear Models

1. **Response variable** - distribution `family`
   - Bernouilli - Binomial
   - Poisson
   - Gamma
   - etc
2. **Predictors** (continuous or categorical)
3. **Link function**
   - Gaussian: identity
   - Binomial: logit, probit

Generalised Linear Models

1. **Response variable** - distribution `family`
   - Bernouilli - Binomial
   - Poisson
   - Gamma
   - etc
2. **Predictors** (continuous or categorical)
3. **Link function**
   - Gaussian: identity
   - Binomial: logit, probit
   - Poisson: log...

Generalised Linear Models

1. **Response variable** - distribution `family`
   - Bernouilli - Binomial
   - Poisson
   - Gamma
   - etc
2. **Predictors** (continuous or categorical)
3. **Link function**
   - Gaussian: identity
   - Binomial: logit, probit
   - Poisson: log...
   - See `family`.

Bernouilli - Binomial distribution (Logistic regression)

- Response variable: Yes/No (e.g. survival, sex, presence/absence)

$$logit(p) = \ln\left(\frac{p}{1-p}\right)$$

Then

$$Pr(alive) = a + bx$$
$$logit(Pr(alive)) = a + bx$$
$$Pr(alive) = invlogit(a + bx) = \frac{e^{a+bx}}{1 + e^{a+bx}}$$

Bernouilli - Binomial distribution (Logistic regression)

- Response variable: Yes/No (e.g. survival, sex, presence/absence)
- Link function: `logit` (others possible, see `family`).

$$logit(p) = \ln\left(\frac{p}{1-p}\right)$$

Then

$$Pr(alive) = a + bx$$
$$logit(Pr(alive)) = a + bx$$
$$Pr(alive) = invlogit(a + bx) = \frac{e^{a+bx}}{1 + e^{a+bx}}$$

Back to survival of Titanic passengers

How many passengers travelled in each class?

```
tapply(titanic$survived, titanic$class, length)  # or aggr
```

```
  crew  first second  third
   885    325    285    706
```

How many survived?

```
tapply(titanic$survived, titanic$class, sum)
```

```
  crew  first second  third
   212    203    118    178
```

What proportion survived in each class?

```
[1] 0.2395480 0.6246154 0.4140351 0.2521246
```

Back to survival of Titanic passengers (dplyr)

Passenger survival according to class

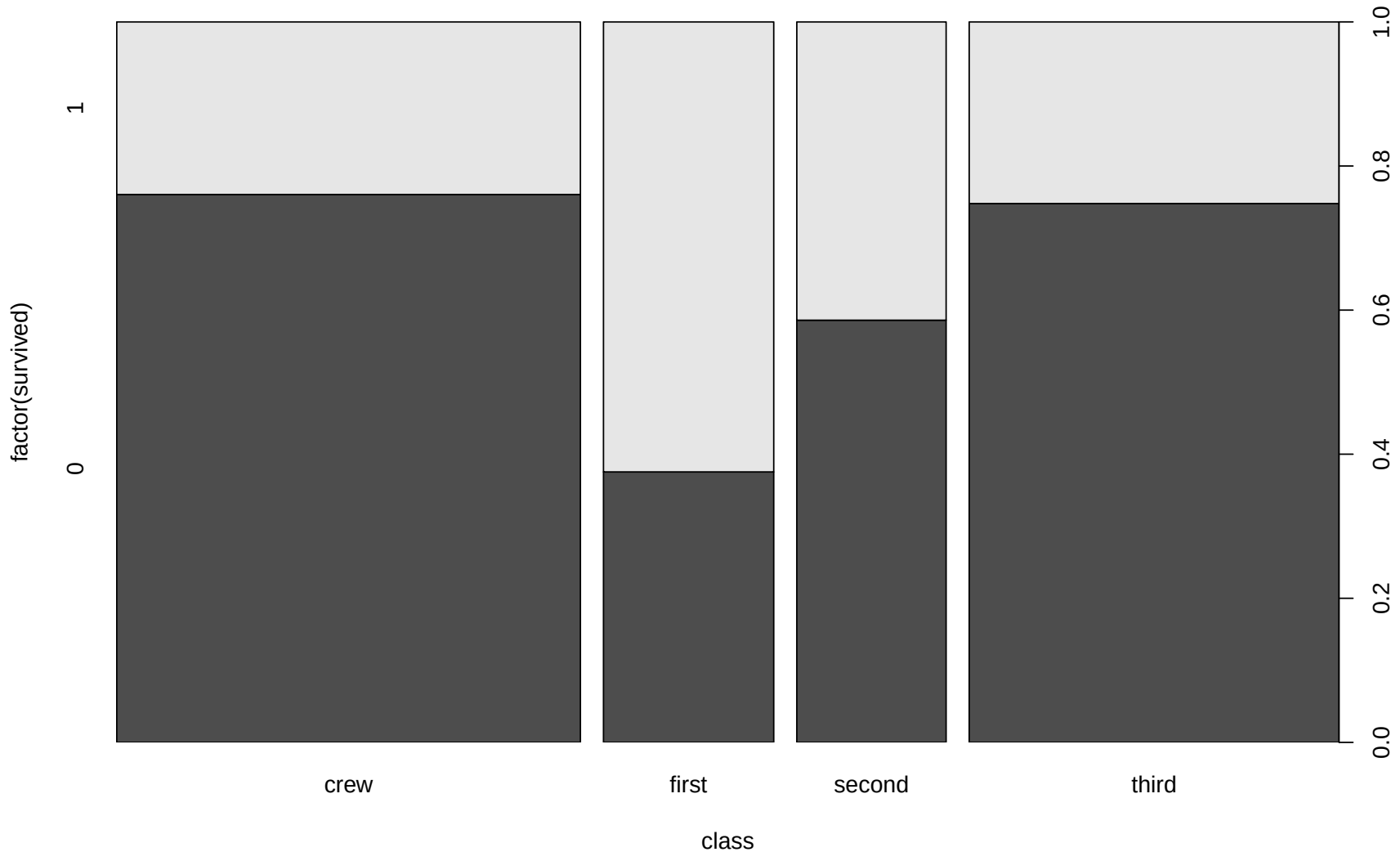
```
library(dplyr)
titanic %>% group_by(class, survived) %>% summarise(count =
```

```
Source: local data frame [8 x 3]
Groups: class

   class survived count
1   crew        0   673
2   crew        1   212
3  first        0   122
4  first        1   203
5 second        0   167
6 second        1   118
7  third        0   528
8  third        1   178
```

Or summarise(group_by(titanic, class, survived),

Or graphically...

```
plot(factor(survived) ~ class, data = titanic)
```



Fitting GLMs in R: glm

```
tit.glm <- glm(survived ~ class, data = titanic, family = b
```

```
Call:
glm(formula = survived ~ class, family = binomial, data = t

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.3999  -0.7623  -0.7401   0.9702   1.6906

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.15516    0.07876 -14.667  < 2e-16 ***
classfirst   1.66434    0.13902  11.972  < 2e-16 ***
classsecond  0.80785    0.14375   5.620 1.91e-08 ***
classthird   0.06785    0.11711   0.579    0.562
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
```

Interpreting logistic regression output

Parameter estimates (logit-scale)

```
 (Intercept)   classfirst classsecond   classthird
-1.15515905   1.66434399  0.80784987   0.06784632
```

We need to back-transform: apply *inverse logit*
Crew probability of survival:

```
plogis(coef(tit.glm)[1])
```

```
(Intercept)
   0.239548
```

Looking at the data, the proportion of crew who survived is

```
[1] 0.239548
```

Q: Probability of survival for 1st class passengers?

```
plogis(coef(tit.glm)[1] + coef(tit.glm)[2])
```

```
(Intercept)
  0.6246154
```

Needs to add intercept (baseline) to the parameter estimate. Again this value matches the data:

```
sum(titanic$survived[titanic$class == "first"])/nrow(titan
    "first", ])
```

```
[1] 0.6246154
```

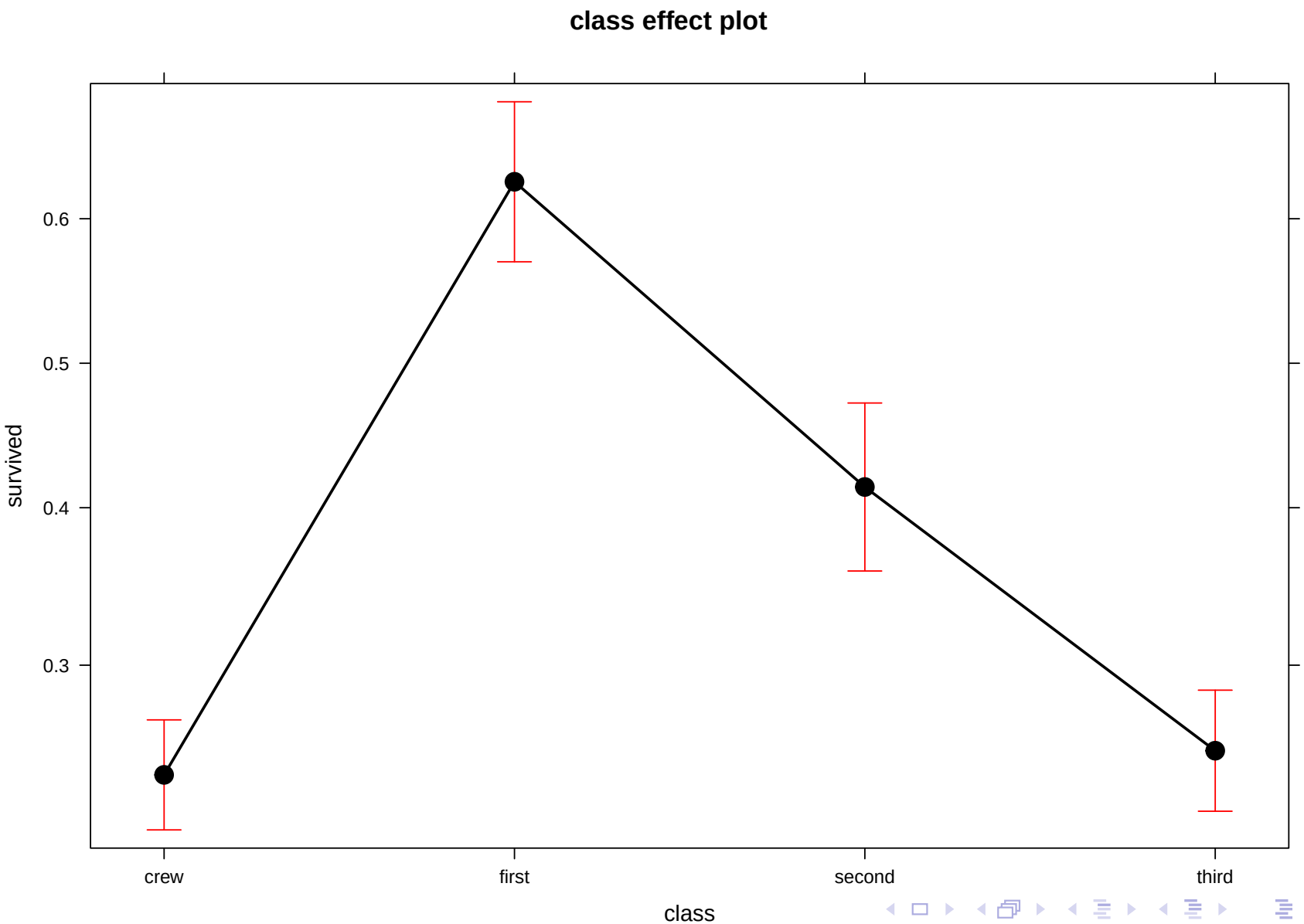
Model interpretation using `effects` package

```
library(effects)
allEffects(tit.glm)
```

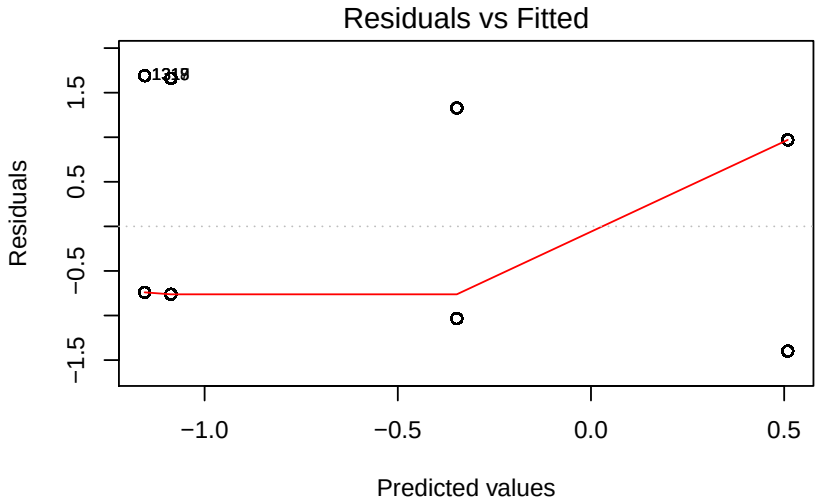
```
 model: survived ~ class

 class effect
class
     crew      first     second      third
0.2395480 0.6246154 0.4140351 0.2521246
```

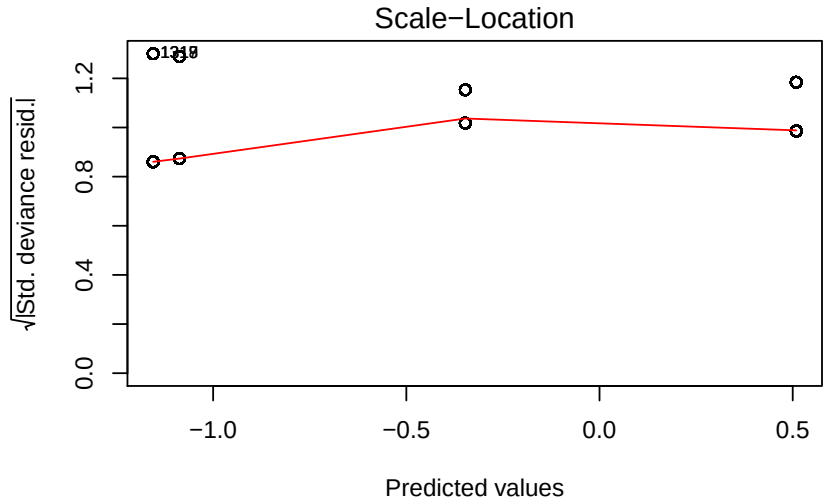
Effects plot

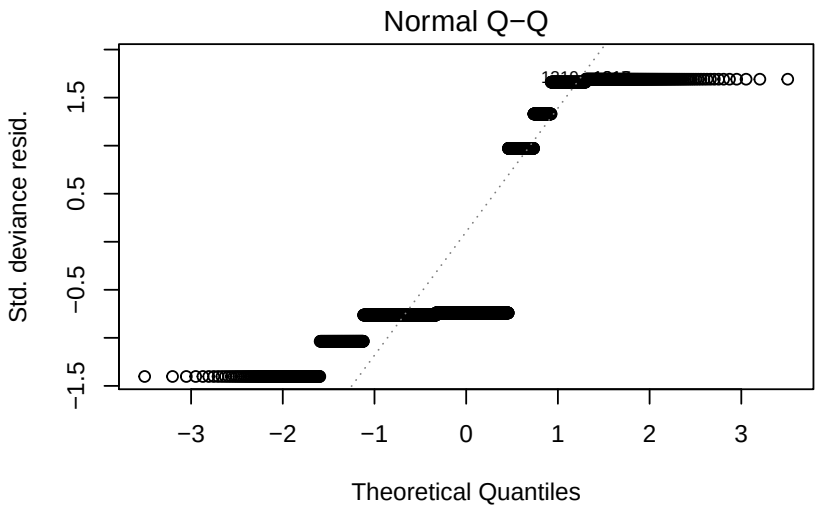
```
plot(allEffects(tit.glm))
```

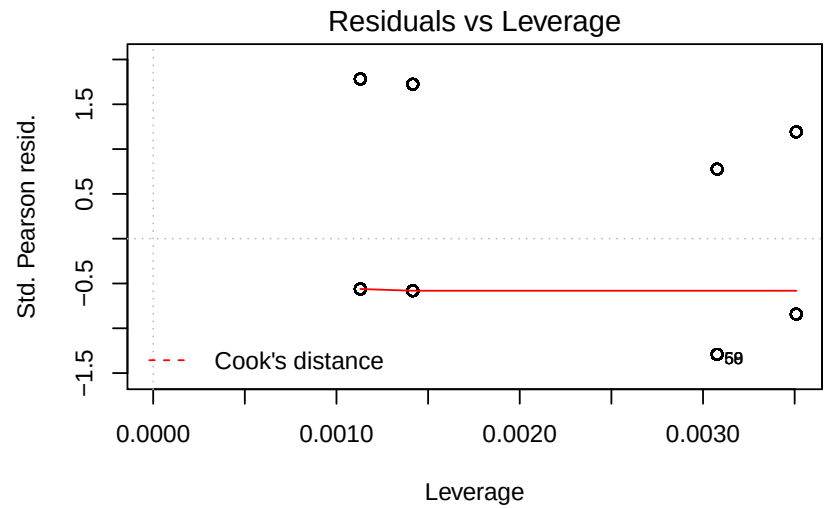


Logistic regression: model checking





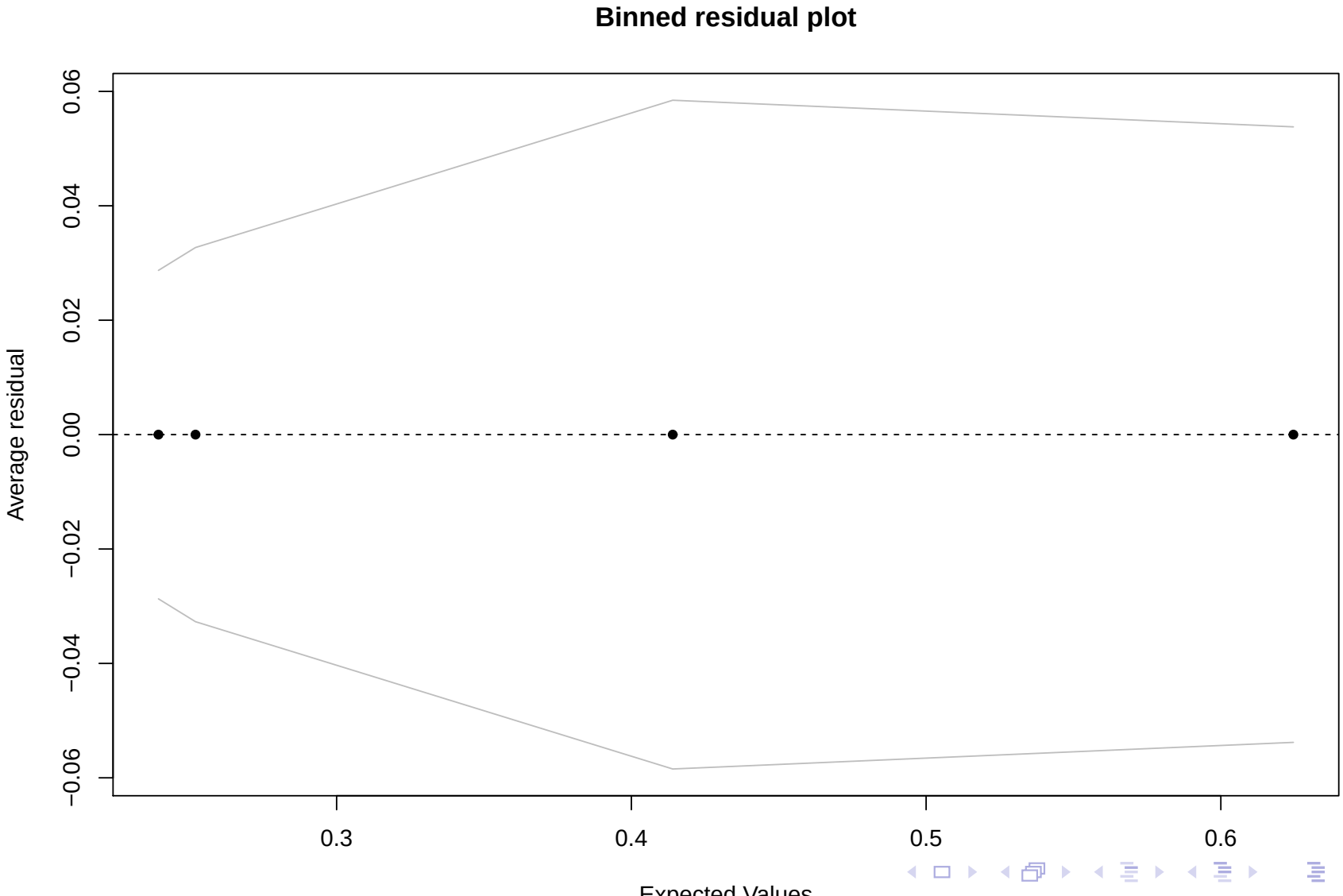




Not very useful.

Binned residual plots for logistic regression

```
predvals <- predict(tit.glm, type = "response")
arm::binnedplot(predvals, titanic$survived - predvals)
```



Recapitulating

1. Import data: `read.table` or `read.csv`

Recapitulating

1. Import data: `read.table` or `read.csv`
2. Check data: `summary`

Recapitulating

1. Import data: `read.table` or `read.csv`
2. Check data: `summary`
3. Plot data: `plot`

Recapitulating

1. Import data: `read.table` or `read.csv`
2. Check data: `summary`
3. Plot data: `plot`
4. Fit model: `glm`. Don't forget to specify `family`!

Recapitulating

1. Import data: `read.table` or `read.csv`
2. Check data: `summary`
3. Plot data: `plot`
4. Fit model: `glm`. Don't forget to specify `family`!
5. Examine models: `summary`

Recapitulating

1. Import data: `read.table` or `read.csv`
2. Check data: `summary`
3. Plot data: `plot`
4. Fit model: `glm`. Don't forget to specify `family`!
5. Examine models: `summary`
6. Use `plogis` to apply back-transformation (*invlogit*) to parameter estimates (`coef`). Alternatively, use `allEffects` from `effects` package.

Recapitulating

1. Import data: `read.table` or `read.csv`
2. Check data: `summary`
3. Plot data: `plot`
4. Fit model: `glm`. Don't forget to specify `family`!
5. Examine models: `summary`
6. Use `plogis` to apply back-transformation (*invlogit*) to parameter estimates (`coef`). Alternatively, use `allEffects` from `effects` package.
7. Plot model: `plot(allEffects(model))`. Or use `visreg`.

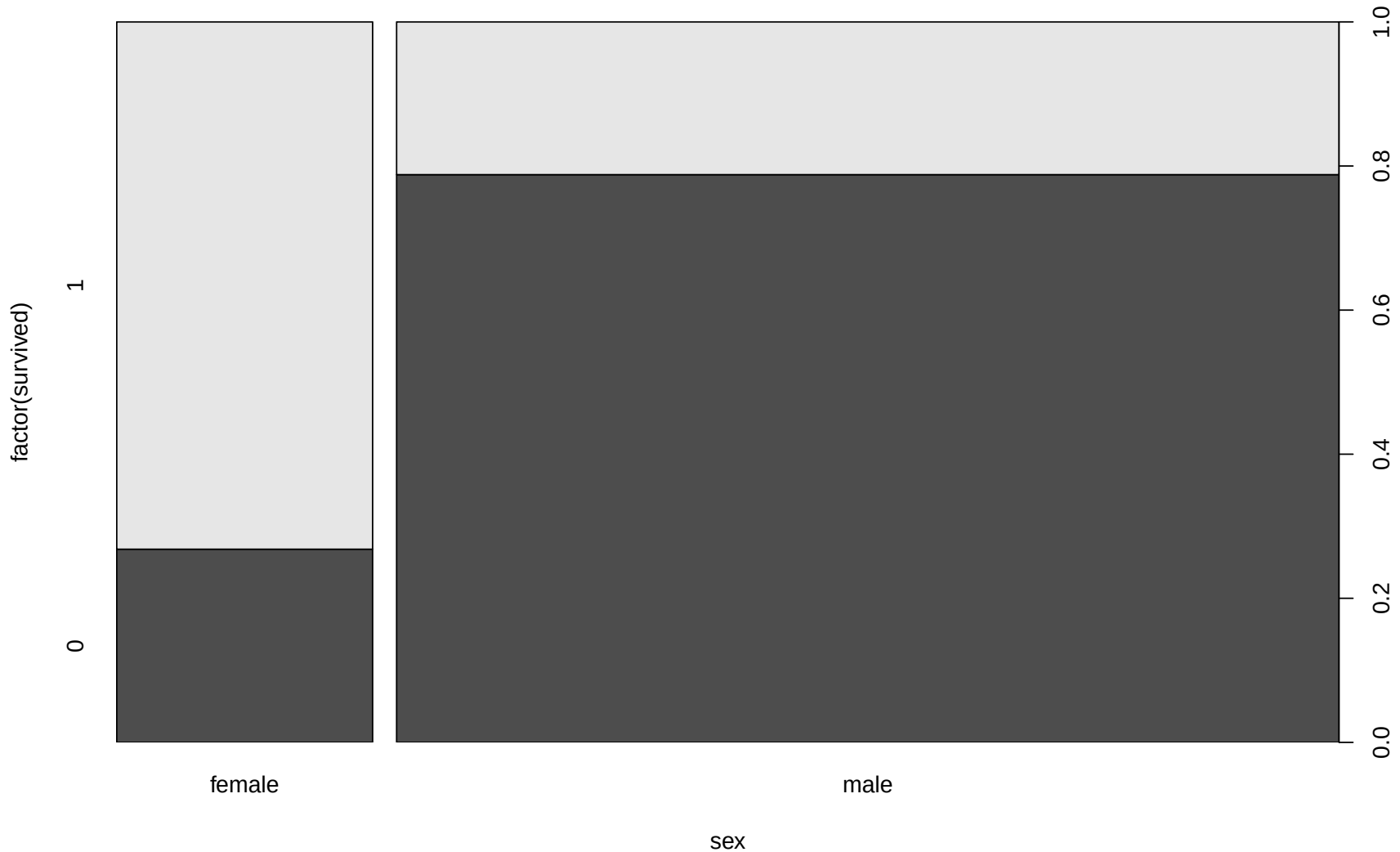
Recapitulating

1. Import data: `read.table` or `read.csv`
2. Check data: `summary`
3. Plot data: `plot`
4. Fit model: `glm`. Don't forget to specify `family`!
5. Examine models: `summary`
6. Use `plogis` to apply back-transformation (*invlogit*) to parameter estimates (`coef`). Alternatively, use `allEffects` from `effects` package.
7. Plot model: `plot(allEffects(model))`. Or use `visreg`.
8. Examine residuals: `binnedplot` from package `arm`. Use `predict` to obtain predicted values for each obs.

Q: Did men have higher survival than women?

Plot first

```
plot(factor(survived) ~ sex, data = titanic)
```



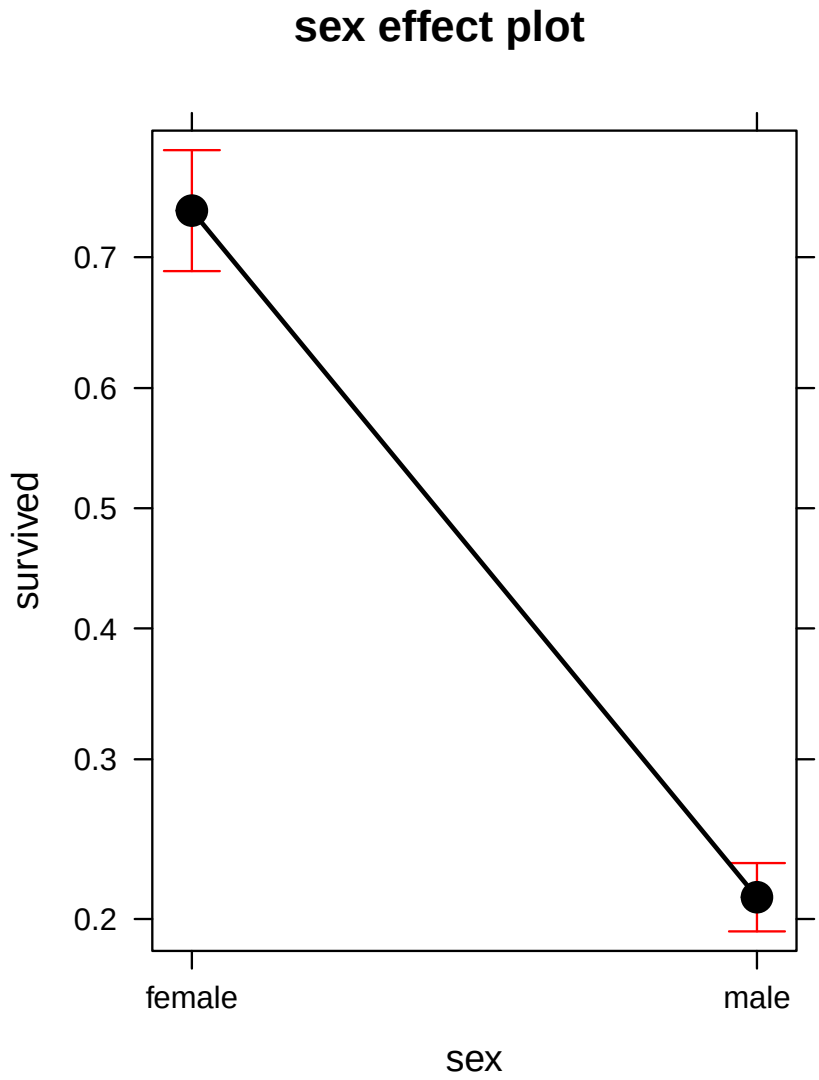
Fit model

```
tit.sex <- glm(survived ~ sex, data = titanic, family = bi
```

```
Call:
glm(formula = survived ~ sex, family = binomial, data = tit

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6226  -0.6903  -0.6903   0.7901   1.7613

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.0044     0.1041   9.645   <2e-16 ***
sexmale      -2.3172     0.1196 -19.376   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '

(Dispersion parameter for binomial family taken to be 1)
```

Effects

```
 model: survived ~ sex

 sex effect
sex
   female      male
0.7319149 0.2120162
```



Q: Did women have higher survival because they travelled more in first class?

Let's look at the data

tapply

```
tapply(titanic$survived, list(titanic$class, titanic$sex),
```

```
       female male
crew       20  192
first     141   62
second     93   25
third      90   88
```

Mmmm...

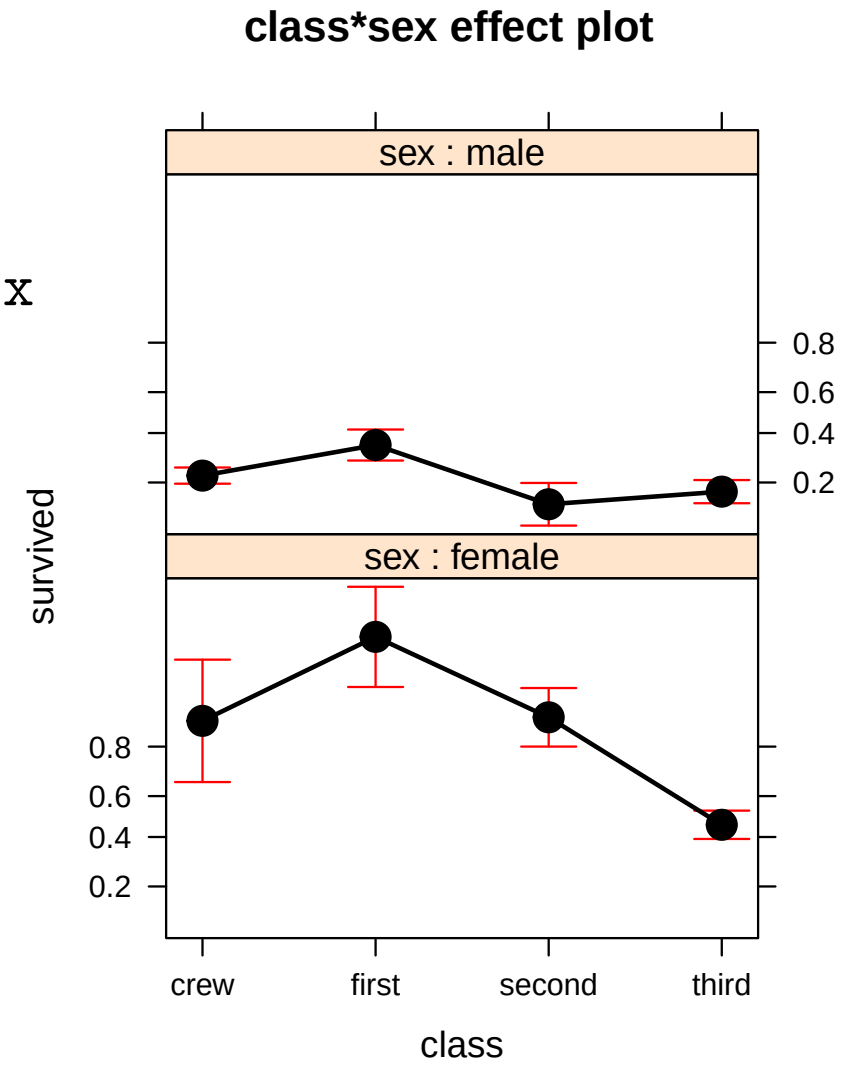
Fit model with both factors (interactions)

```
tit.sex.class <- glm(survived ~ class * sex, data = titanic
```

```
glm(formula = survived ~ class * sex, family = binomial, da
                      coef.est coef.se
(Intercept)            1.90     0.62
classfirst             1.67     0.80
classsecond            0.07     0.69
classthird            -2.06     0.64
sexmale               -3.15     0.62
classfirst:sexmale    -1.06     0.82
classsecond:sexmale   -0.64     0.72
classthird:sexmale     1.74     0.65
---
  n = 2201, k = 8
  residual deviance = 2163.7, null deviance = 2769.5 (diffe
```

Effects

```
model: survived ~ class * sex

class*sex effect
        sex
class         female      male
  crew    0.8695652 0.2227378
  first   0.9724138 0.3444444
  second  0.8773585 0.1396648
  third   0.4591837 0.1725490
```



So, women had higher probability of survival than men, even within the same class.

Logistic regression for proportion data

Read Titanic data in different format

```
tit.prop <- read.csv("http://vincentarelbundock.github.io/R
summary(tit.prop)
```

```
       X             Class       Sex         Age      Survived
 Min.   : 1.00   1st :8   Female:16   Adult:16   No :16   N
 1st Qu.: 8.75   2nd :8   Male  :16   Child:16   Yes:16   1
 Median :16.50   3rd :8                                   N
 Mean   :16.50   Crew:8                                   N
 3rd Qu.:24.25                                            3
 Max.   :32.00                                            N
```

These are the same data, but summarized (see `Freq` variable).

Reshaping data frame

```
library(reshape2)
tit.prop <- dcast(tit.prop, Class + Sex + Age ~ Survived)
```

Load dataset

```
tit.prop <- read.csv("data-raw/Titanic_prop.csv")
```

Use cbind(n.success, n.failures) as response

```
prop.glm <- glm(cbind(Yes, No) ~ Class, data = tit.prop, fa
```

```
Call:
glm(formula = cbind(Yes, No) ~ Class, family = binomial, da

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-9.6404  -0.2915   1.5698   5.0366  10.1516

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.5092     0.1146   4.445 8.79e-06 ***
Class2nd     -0.8565     0.1661  -5.157 2.51e-07 ***
Class3rd     -1.5965     0.1436 -11.114  < 2e-16 ***
ClassCrew    -1.6643     0.1390 -11.972  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
```

Effects

```
 model: cbind(Yes, No) ~ Class

 Class effect
Class
      1st       2nd       3rd      Crew
0.6246154 0.4140351 0.2521246 0.2395480
```

Compare with former model based on raw data:

```
 model: survived ~ class

 class effect
class
     crew     first    second     third
0.2395480 0.6246154 0.4140351 0.2521246
```

Same results!

Logistic regression with continuous predictors

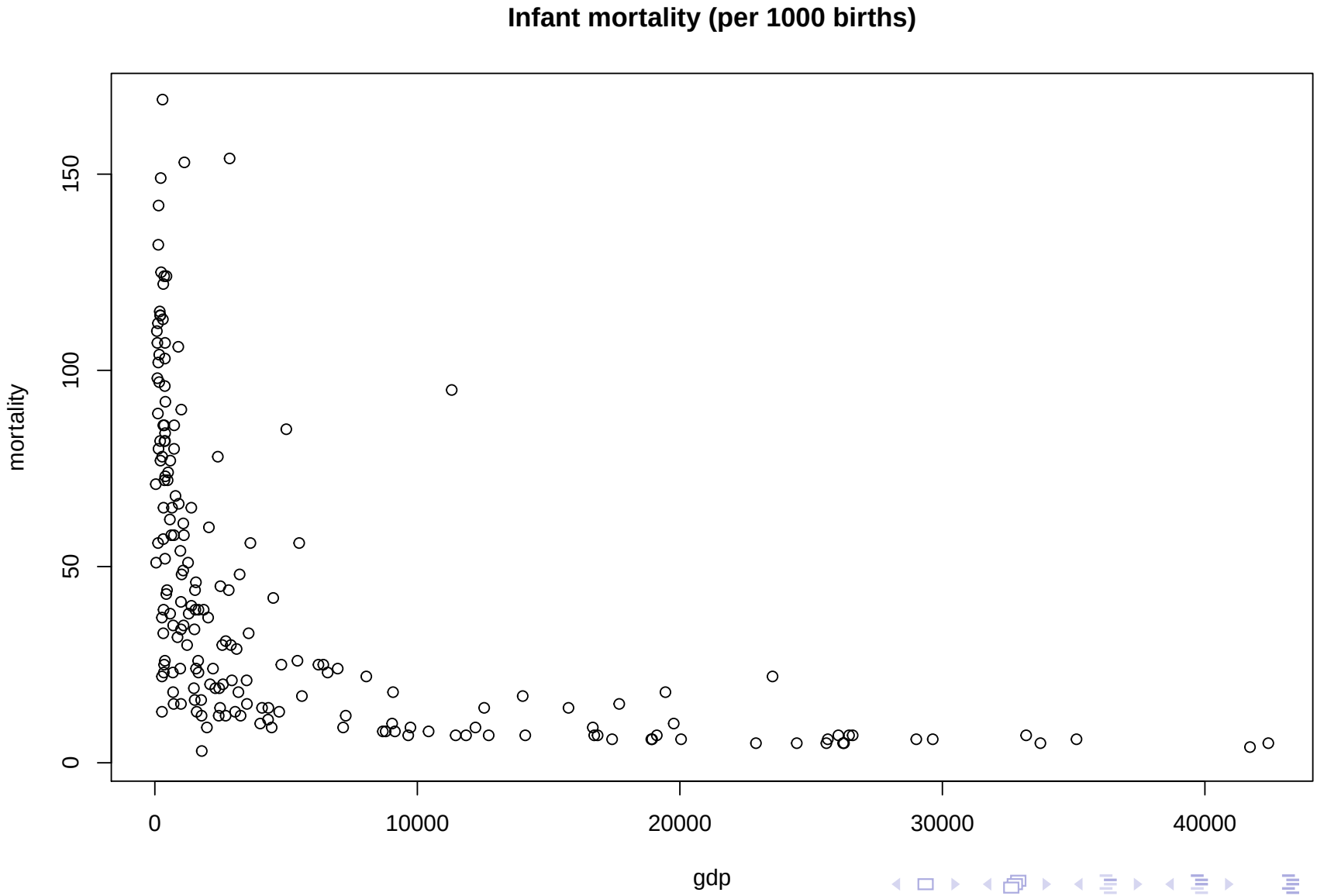
Example dataset: GDP and infant mortality

```
gdp <- read.csv("http://vincentarelbundock.github.io/Rdatas
names(gdp) <- c("country", "mortality", "gdp")
summary(gdp)
```

```
             country       mortality           gdp
 Afghanistan    :  1   Min.   :  2.00   Min.   :   36
 Albania        :  1   1st Qu.: 12.00   1st Qu.:  442
 Algeria        :  1   Median : 30.00   Median : 1779
 American.Samoa:  1   Mean   : 43.48   Mean   : 6262
 Andorra        :  1   3rd Qu.: 66.00   3rd Qu.: 7272
 Angola         :  1   Max.   :169.00   Max.   :42416
 (Other)        :201   NA's   :6        NA's   :10
```

EDA

```
plot(mortality ~ gdp, data = gdp, main = "Infant mortality
```



Fit model

```
gdp.glm <- glm(cbind(mortality, 1000 - mortality) ~ gdp, da
```

```
Call:
glm(formula = cbind(mortality, 1000 - mortality) ~ gdp, fam
    data = gdp)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-9.2230  -3.5163  -0.5697   2.4284  13.5849

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.657e+00  1.311e-02 -202.76   <2e-16 ***
gdp         -1.279e-04  3.458e-06  -36.98   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
```

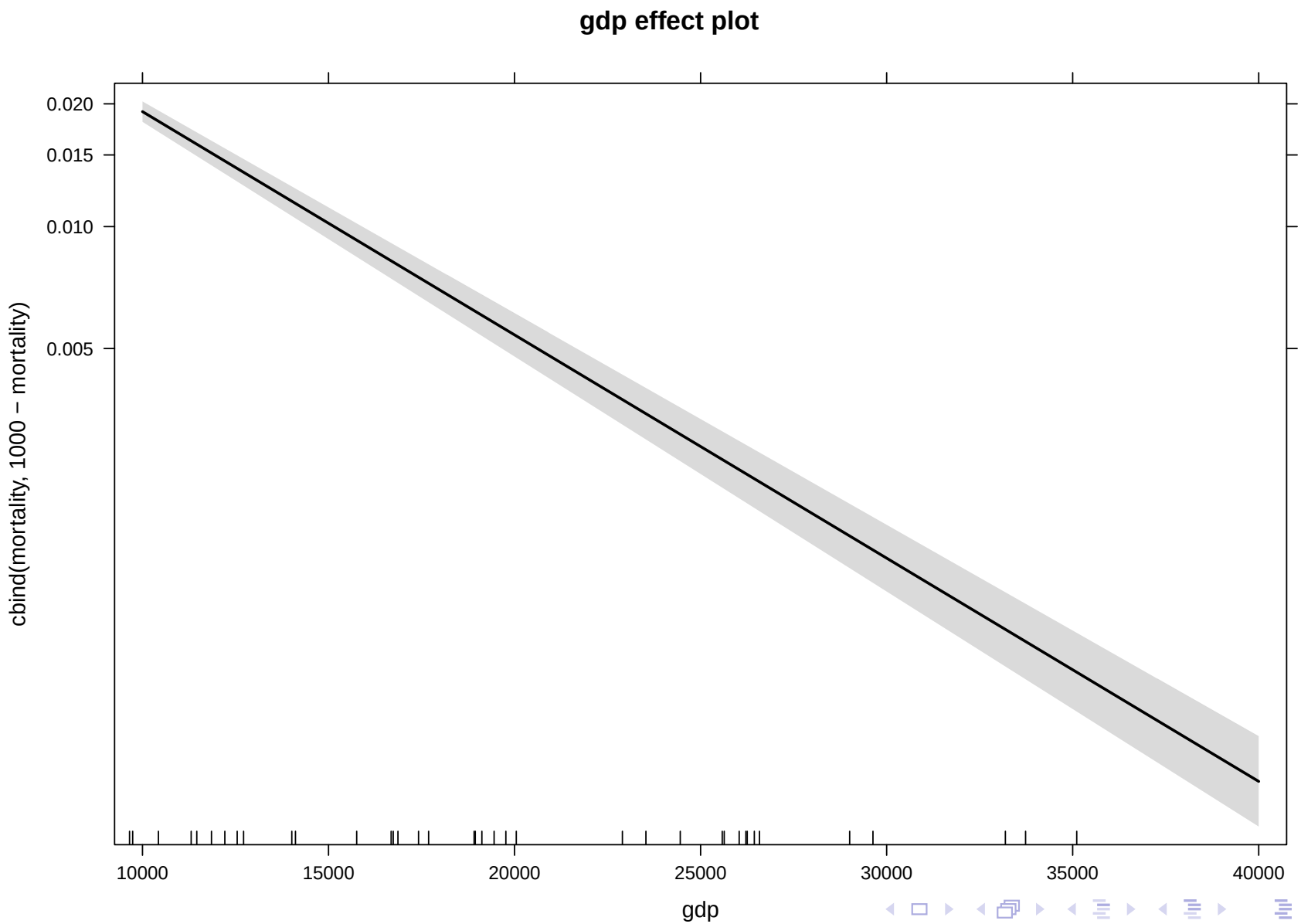
Effects

```
allEffects(gdp.glm)
```

```
 model: cbind(mortality, 1000 - mortality) ~ gdp

 gdp effect
gdp
      10000       20000       30000       40000
0.019143883 0.005402809 0.001509607 0.000420615
```

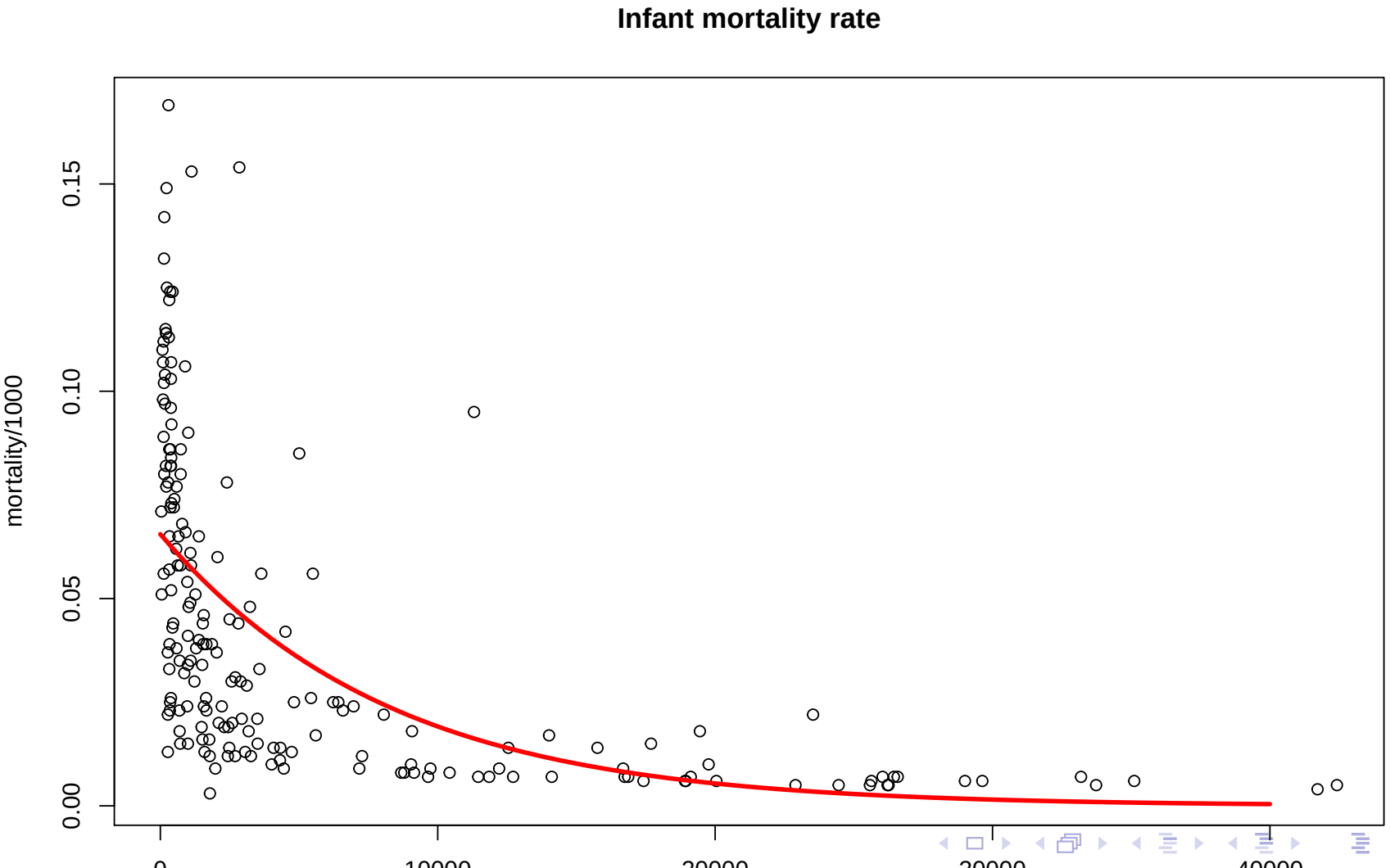
Effects plot

```
plot(allEffects(gdp.glm))
```



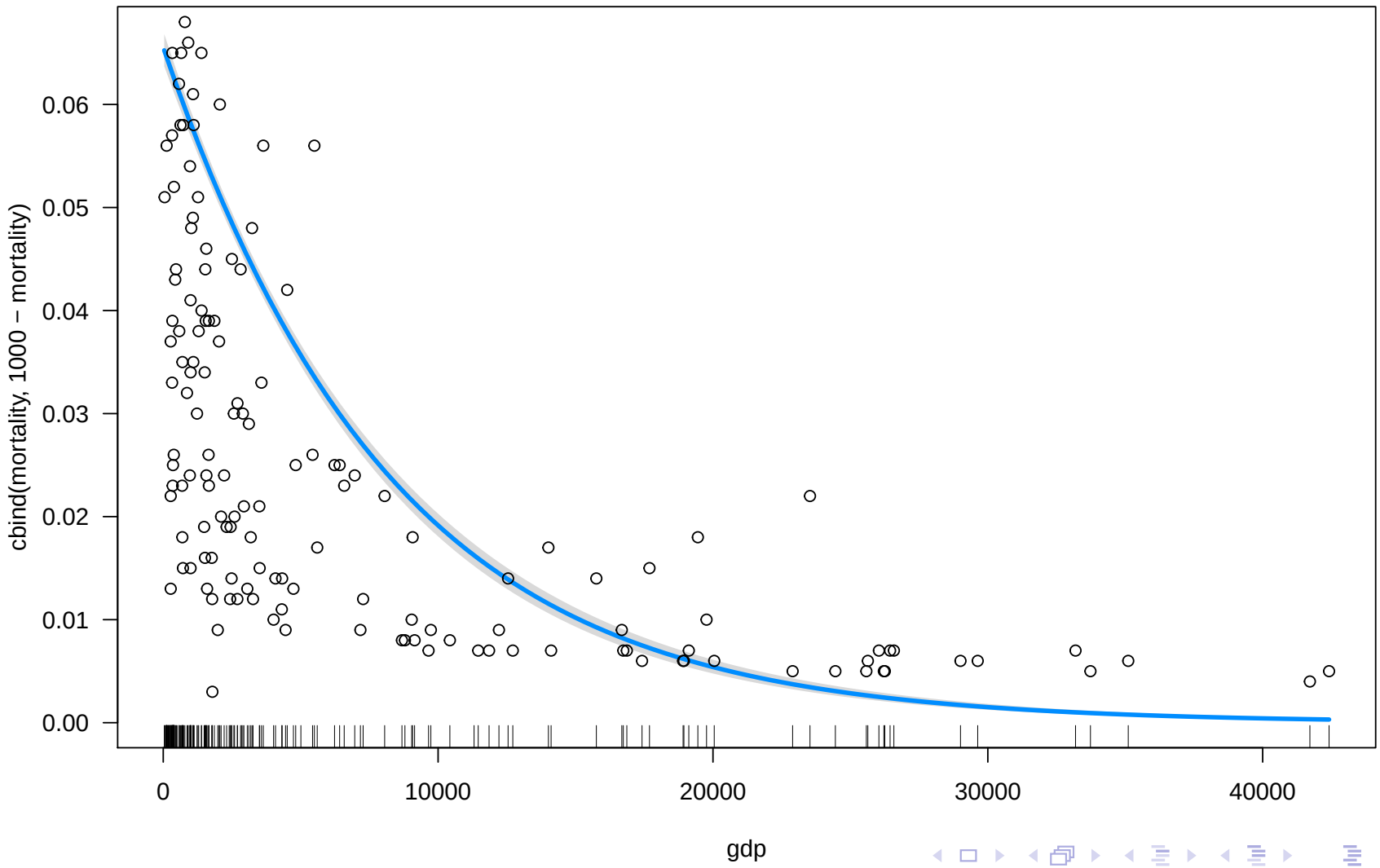
Plot model and data

```
plot(mortality/1000 ~ gdp, data = gdp, main = "Infant morta
curve(plogis(coef(gdp.glm)[1] + coef(gdp.glm)[2] * x), fro
    add = TRUE, lwd = 3, col = "red")
```



Or using visreg:

```
visreg(gdp.glm, scale = "response")
points(mortality/1000 ~ gdp, data = gdp)
```



Overdispersion

Overdispersion in logistic regression with proportion data

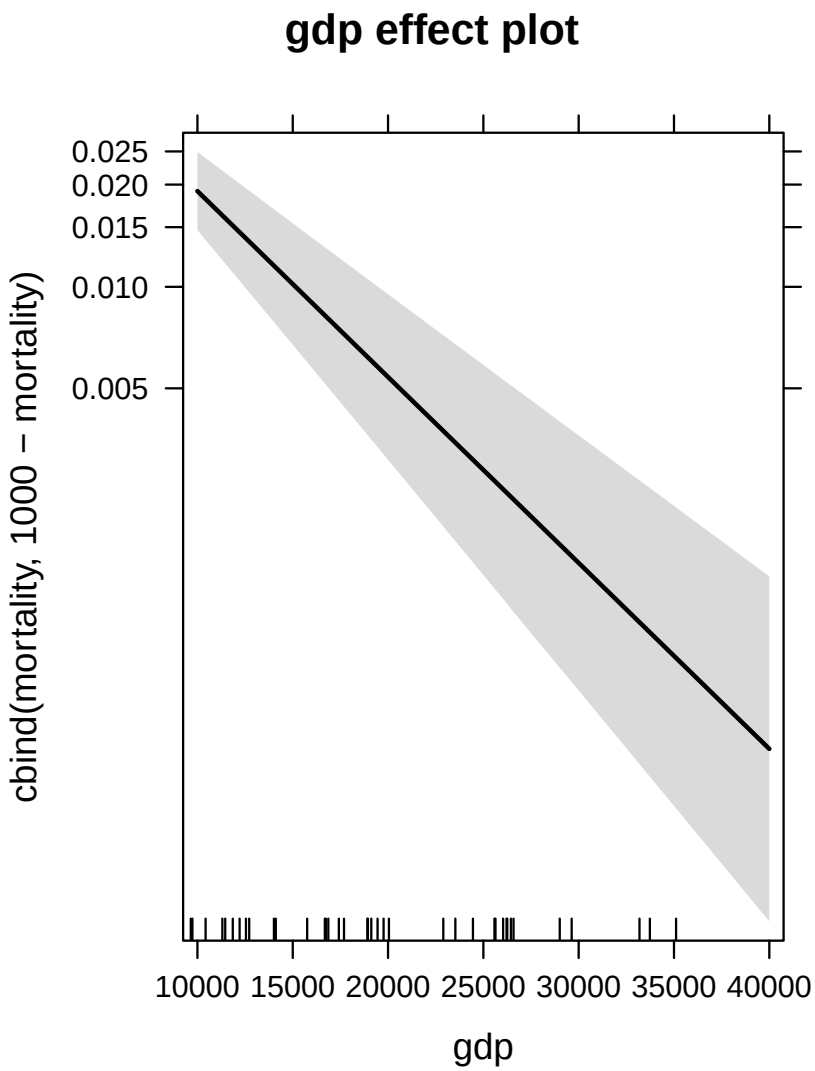
```
gdp.overdisp <- glm(cbind(mortality, 1000 - mortality) ~ gd
```

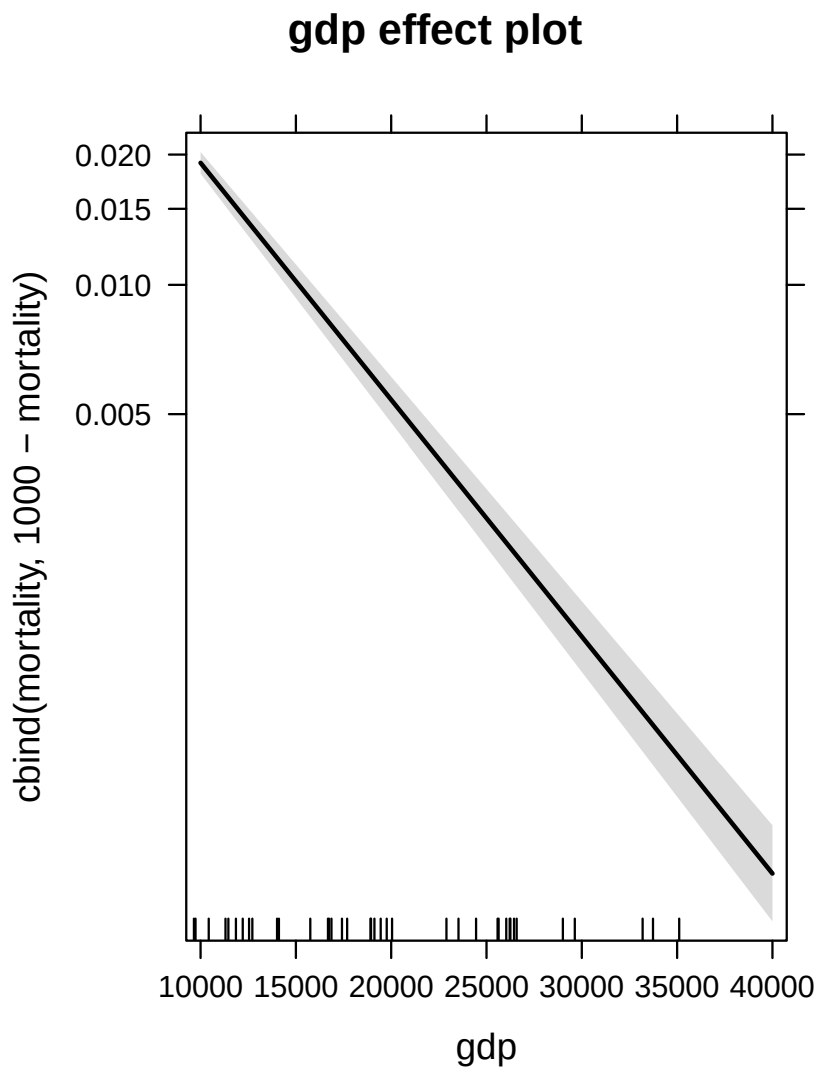
```
Call:
glm(formula = cbind(mortality, 1000 - mortality) ~ gdp, fam
    data = gdp)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-9.2230  -3.5163  -0.5697   2.4284  13.5849

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.657e+00  5.977e-02 -44.465  < 2e-16 ***
gdp         -1.279e-04  1.577e-05  -8.111 5.96e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
```

Mean estimates do not change after accounting for overdispersion

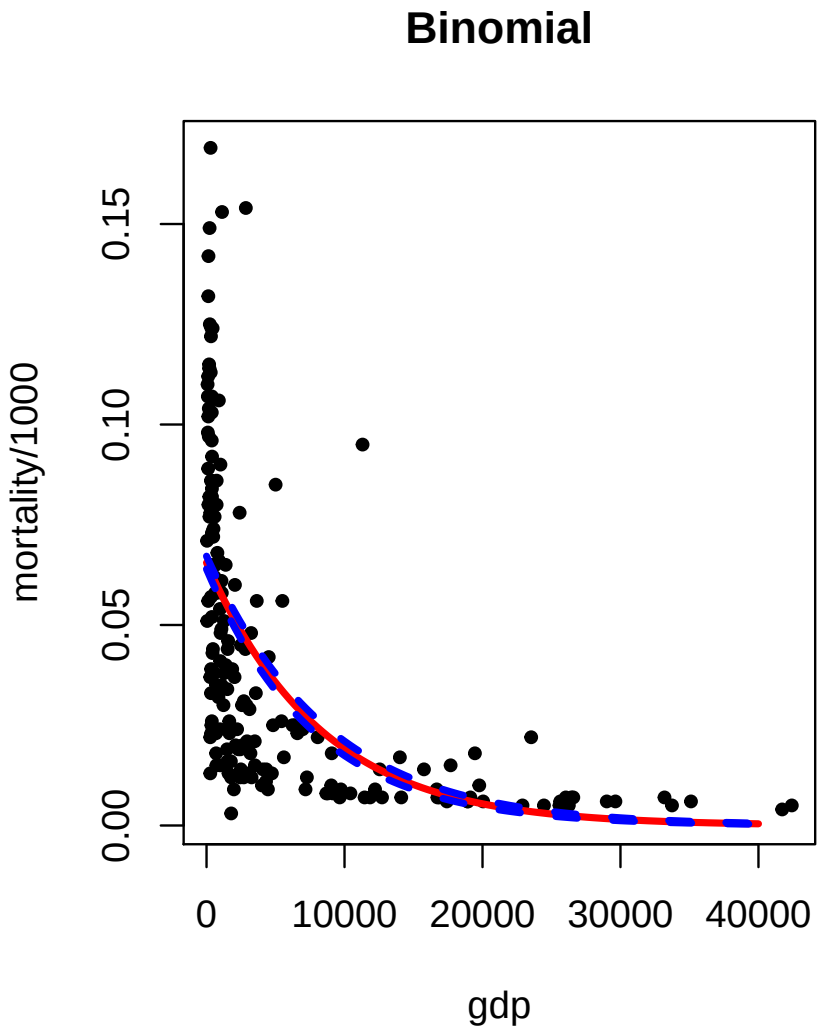
```
 model: cbind(mortality, 1000 - mortality) ~ gdp

 gdp effect
gdp
      10000       20000       30000       40000
0.019143883 0.005402809 0.001509607 0.000420615

 model: cbind(mortality, 1000 - mortality) ~ gdp

 gdp effect
gdp
      10000       20000       30000       40000
0.019143883 0.005402809 0.001509607 0.000420615
```

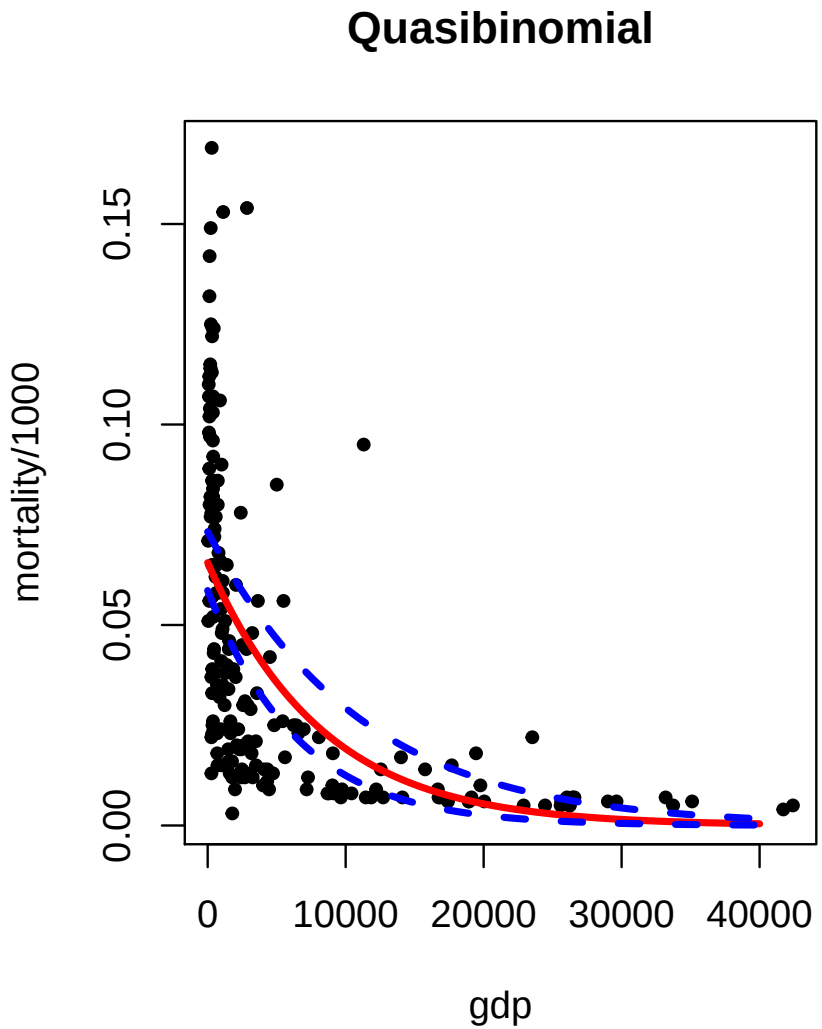
But standard errors (uncertainty) do!





Plot model and data





Overdispersion

Whenever you fit logistic regression to **proportion** data, check family `quasibinomial`.

GLMs for count data: Poisson regression

Types of response variable

- Gaussian: `lm`

Types of response variable

- Gaussian: `lm`
- Bernouilli / Binomial: `glm` (family `binomial` / `quasibinomial`)

Types of response variable

- Gaussian: `lm`
- Bernouilli / Binomial: `glm` (family `binomial` / `quasibinomial`)
- Counts: `glm` (family `poisson` / `quasipoisson`)

Poisson regression

- Response variable: Counts (0, 1, 2, 3...) - discrete

Then

$$log(N) = a + bx$$
$$N = e^{a+bx}$$

Poisson regression

- Response variable: Counts (0, 1, 2, 3...) - discrete
- Link function: `log`

Then

$$log(N) = a + bx$$
$$N = e^{a+bx}$$

Example dataset: Seedling counts in 0.5 m2 quadrats

```
seedl <- read.csv("data-raw/seedlings.csv")
```

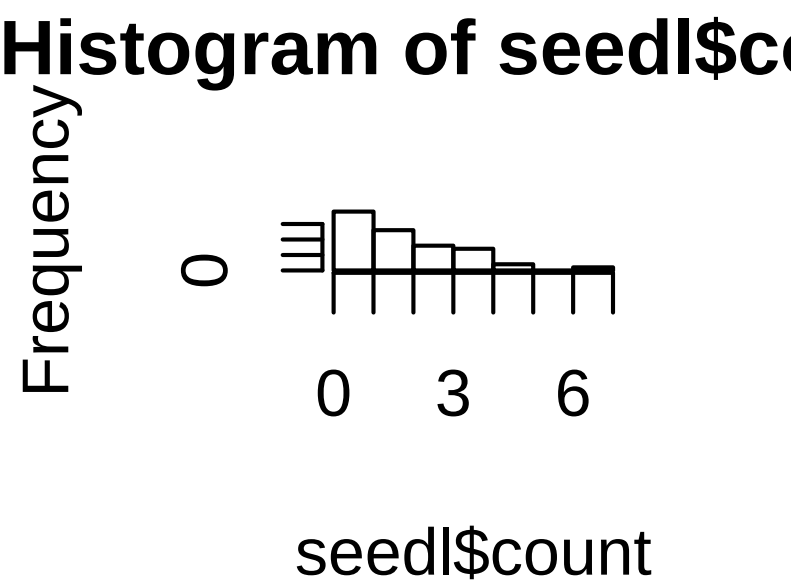
```
       X                count             row              col       
 Min.   : 1.00   Min.   :0.00   Min.   :1   Min.   : 1.0  
 1st Qu.:13.25   1st Qu.:1.00   1st Qu.:2   1st Qu.: 3.0  
 Median :25.50   Median :2.00   Median :3   Median : 5.5  
 Mean   :25.50   Mean   :2.14   Mean   :3   Mean   : 5.5  
 3rd Qu.:37.75   3rd Qu.:3.00   3rd Qu.:4   3rd Qu.: 8.0  
 Max.   :50.00   Max.   :7.00   Max.   :5   Max.   :10.0  
     light       
 Min.   : 2.571  
 1st Qu.:26.879  
 Median :47.493  
 Mean   :47.959  
 3rd Qu.:67.522  
 Max.   :99.135  
```

EDA

```
table(seedl$count)
```

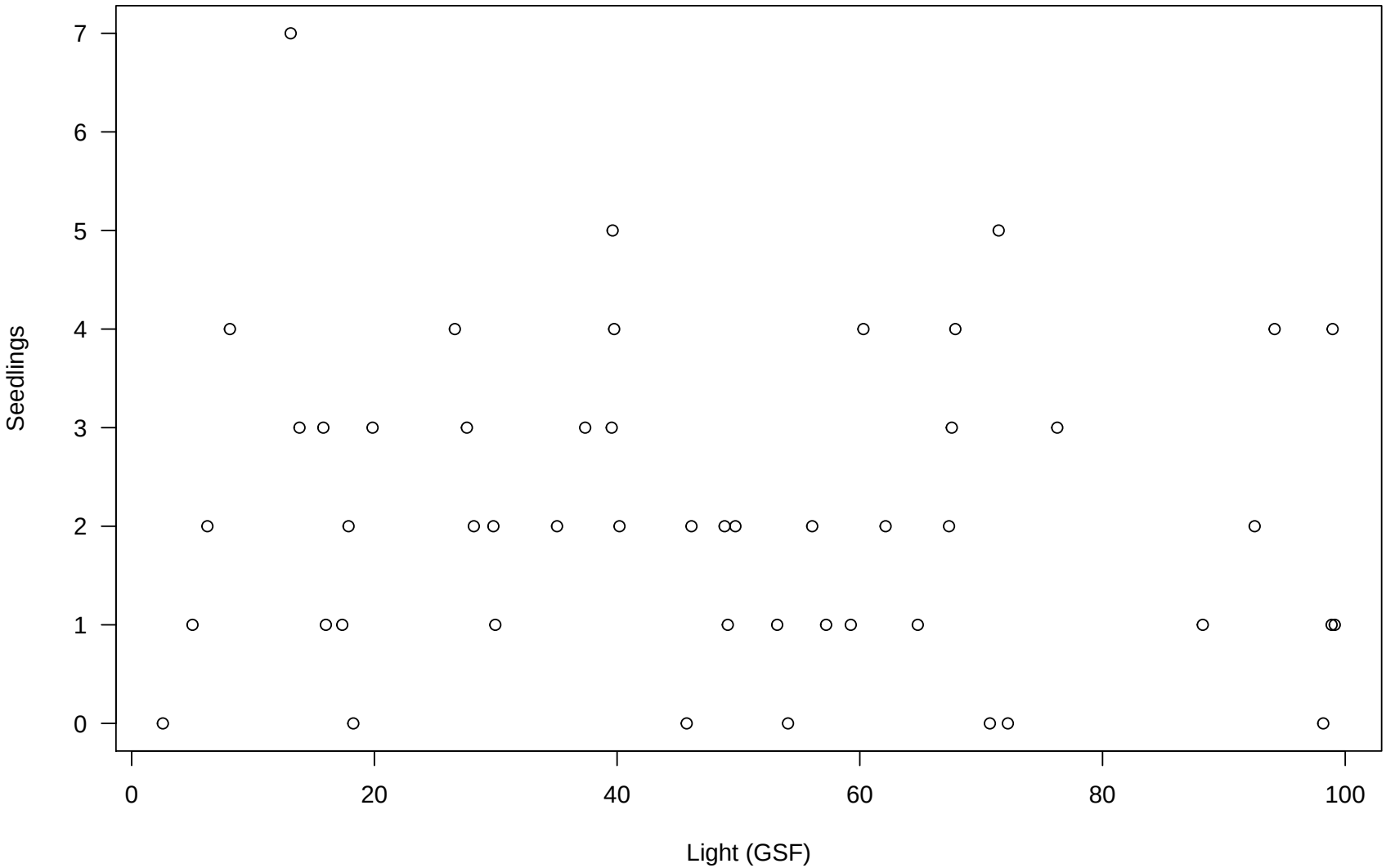
```
 0  1  2  3  4  5  7
 7 12 13  8  7  2  1
```

```
hist(seedl$count)
```



Q: Relationship between Nseedlings and light?

```
plot(seedl$light, seedl$count, las = 1, xlab = "Light (GSF)
```



Let's fit model (Poisson regression)

```
seedl.glm <- glm(count ~ light, data = seedl, family = pois
summary(seedl.glm)
```

```
Call:
glm(formula = count ~ light, family = poisson, data = seedl

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1906  -0.8466  -0.1110   0.5220   2.4577

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.881805   0.188892   4.668 3.04e-06 ***
light       -0.002576   0.003528  -0.730    0.465
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
```

Interpreting Poisson regression output

Parameter estimates (log scale):

```
coef(seedl.glm)
```

```
 (Intercept)        light
0.881805022 -0.002575656
```

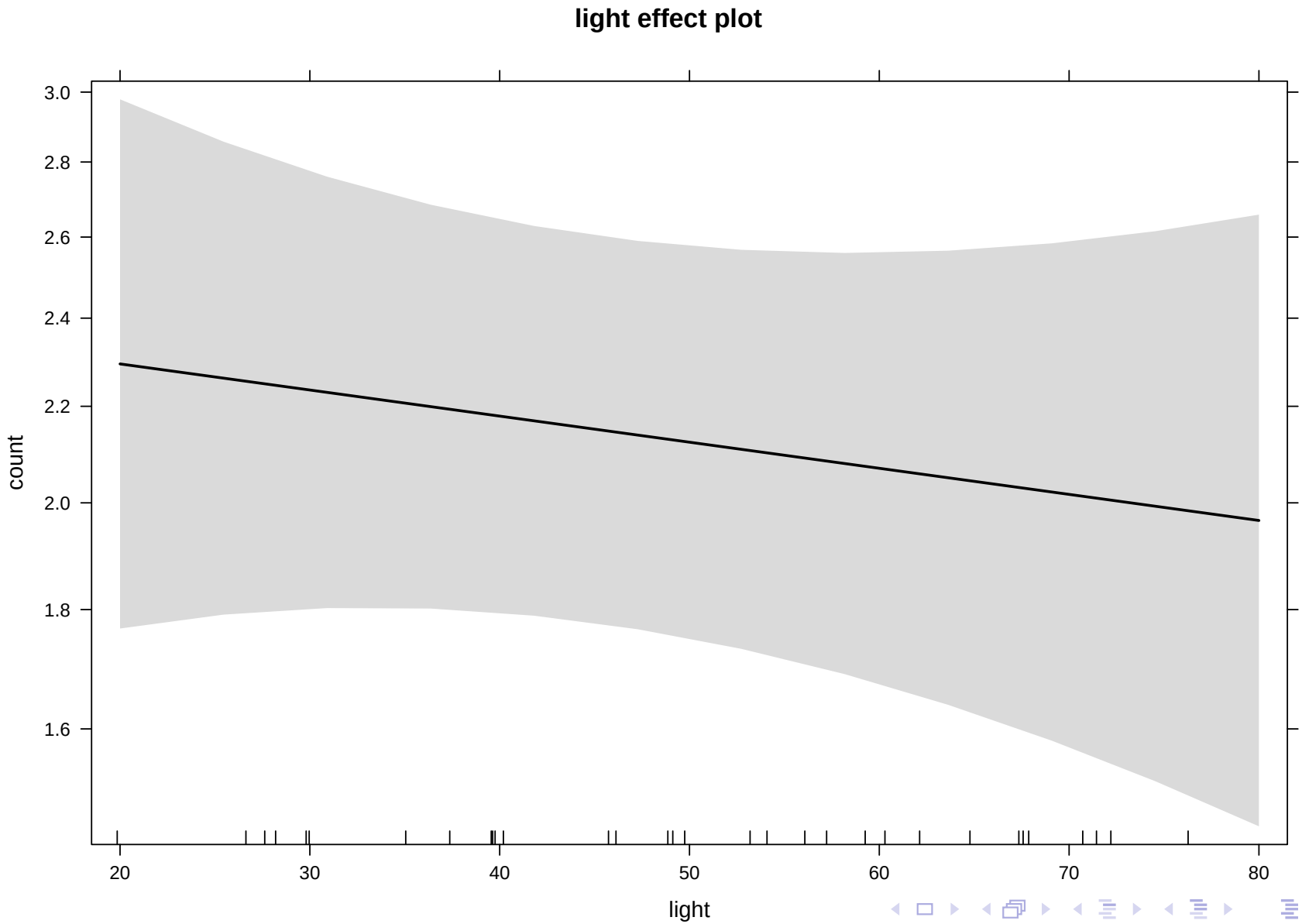
We need to back-transform: apply the inverse of the logarithm

```
exp(coef(seedl.glm))
```

```
(Intercept)       light
  2.4152554   0.9974277
```

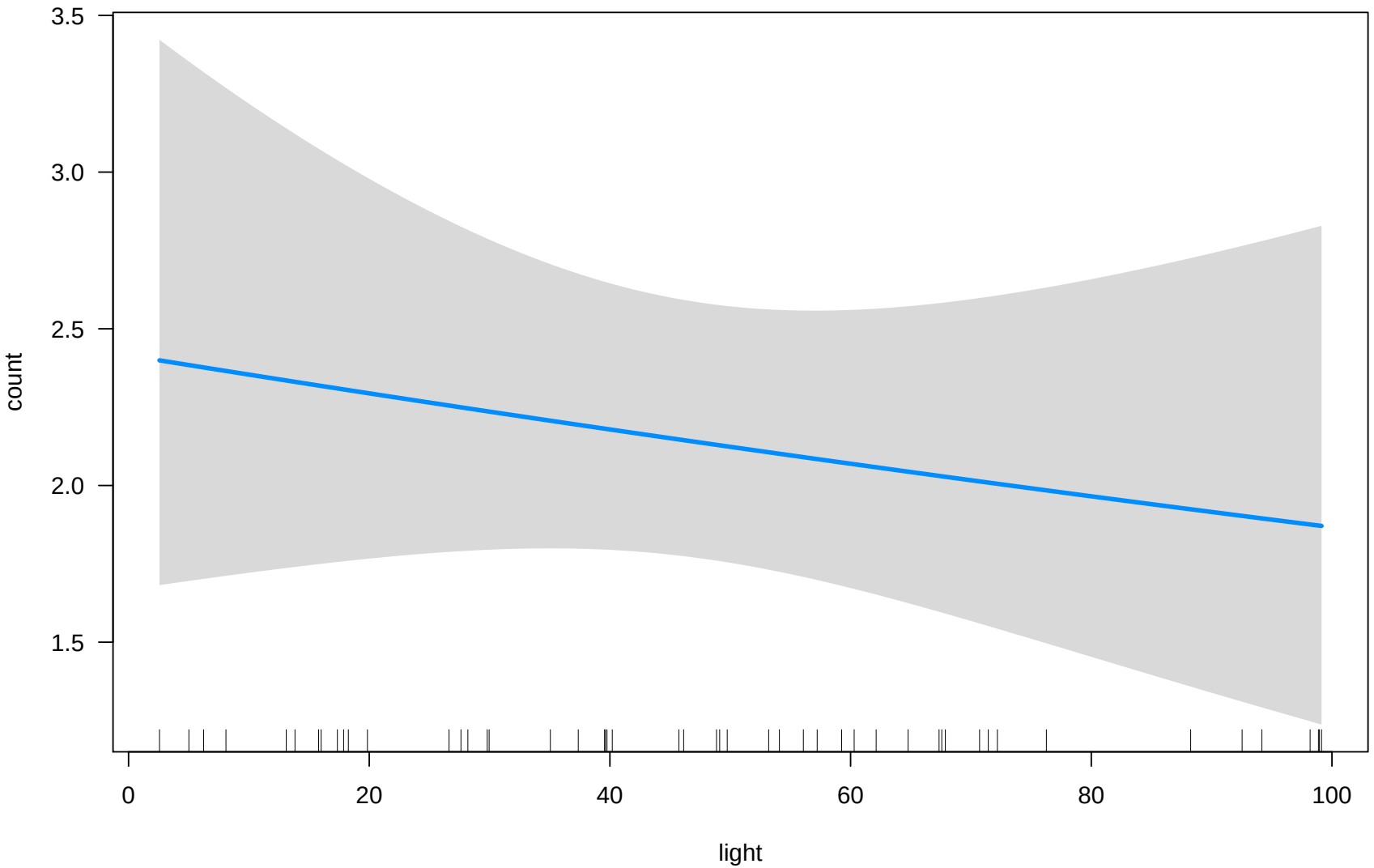
So what's the relationship between Nseedlings and light?

```
plot(allEffects(seedl.glm))
```

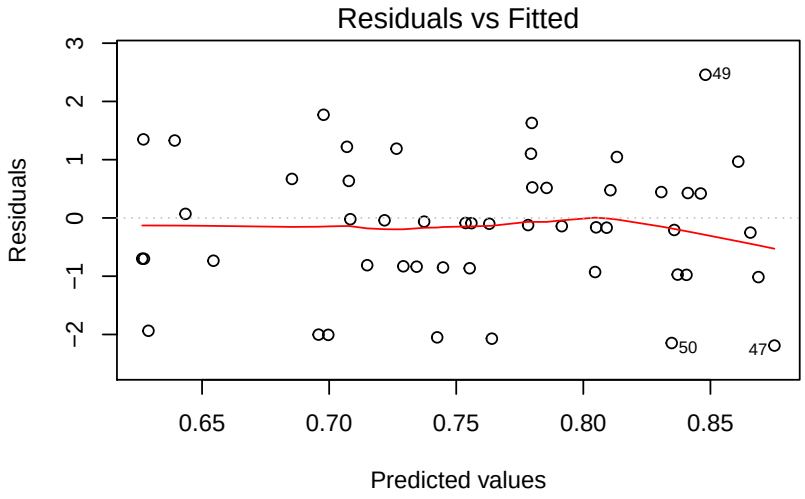


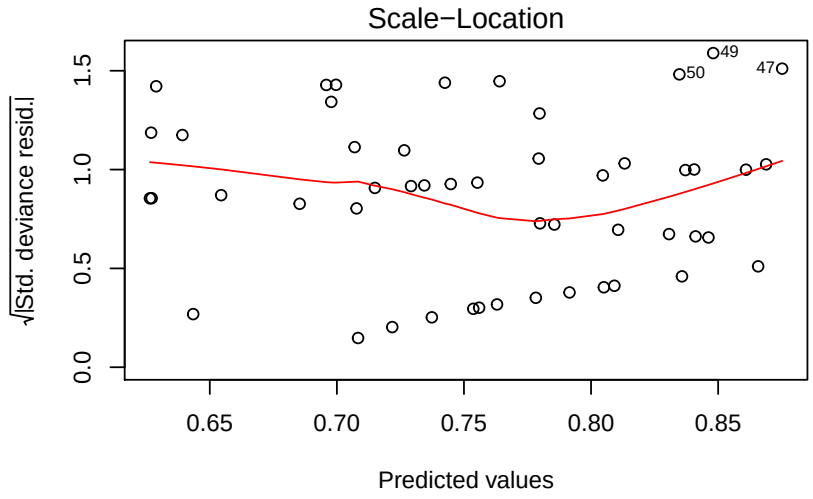
Using visreg

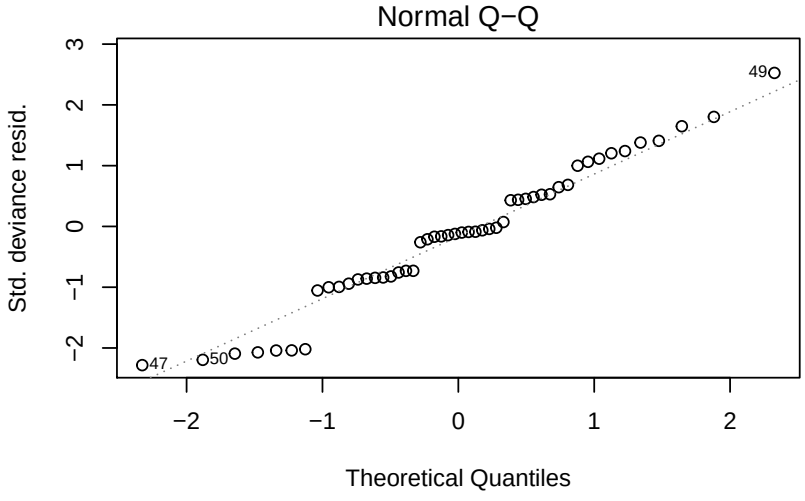
```
visreg(seedl.glm, scale = "response")
```

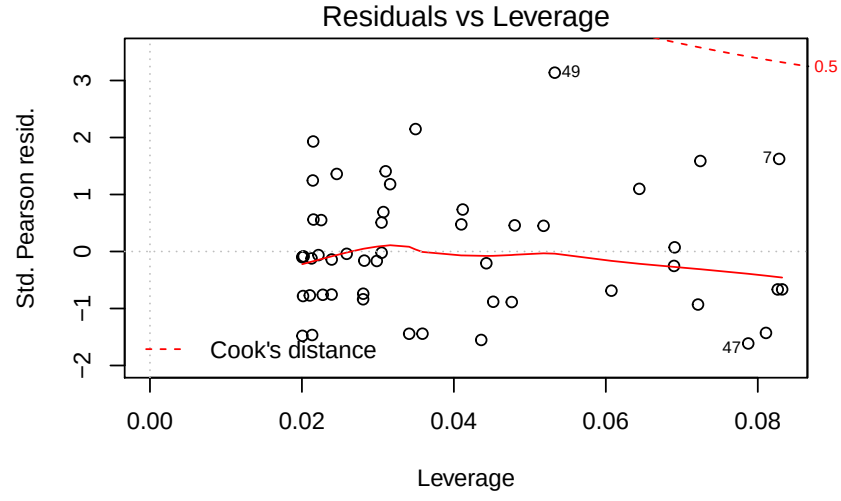


Poisson regression: model checking



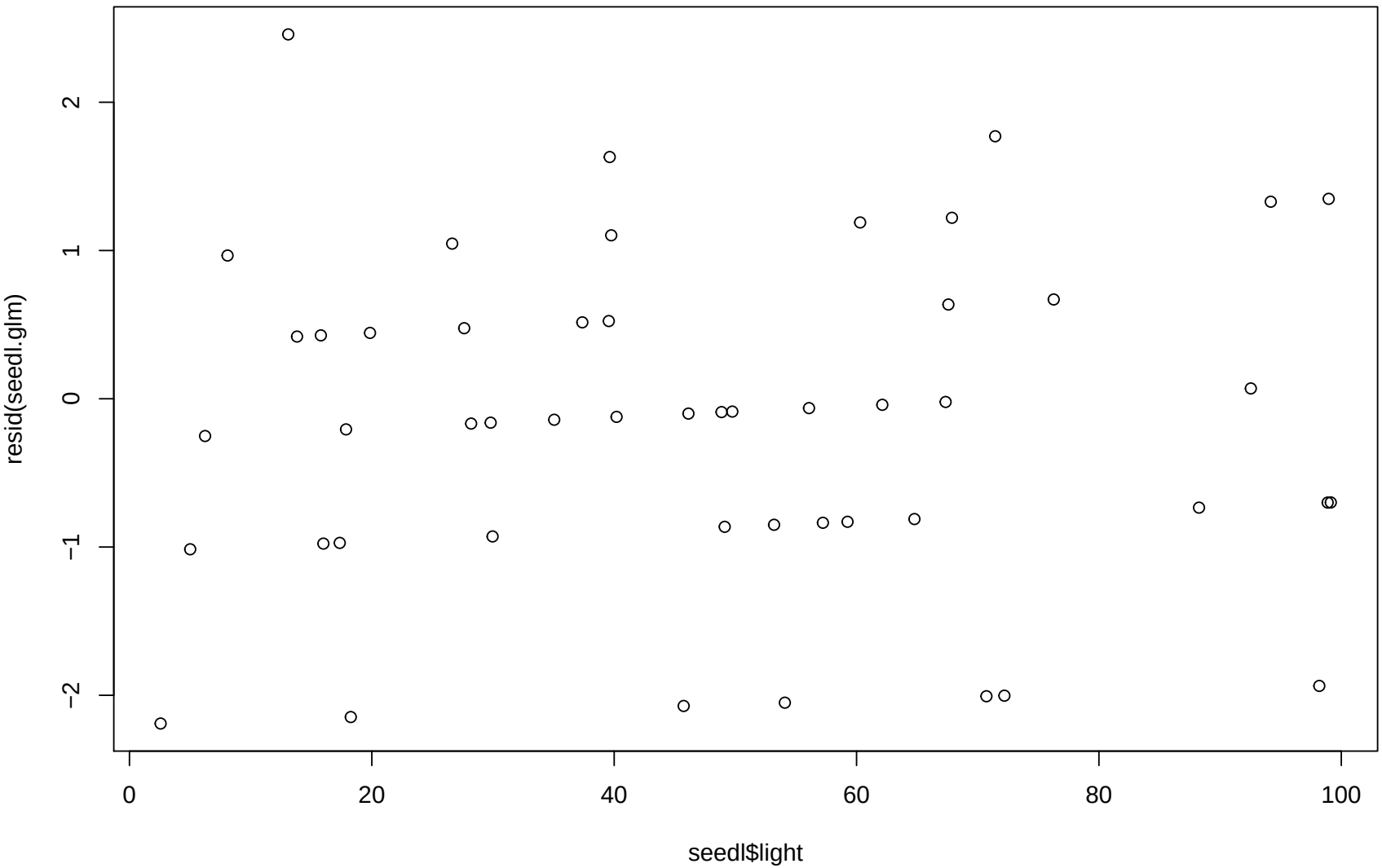






Is there pattern of residuals along predictor?

```
plot(seedl$light, resid(seedl.glm))
```



Poisson regression: Overdispersion

Always check overdispersion with count data

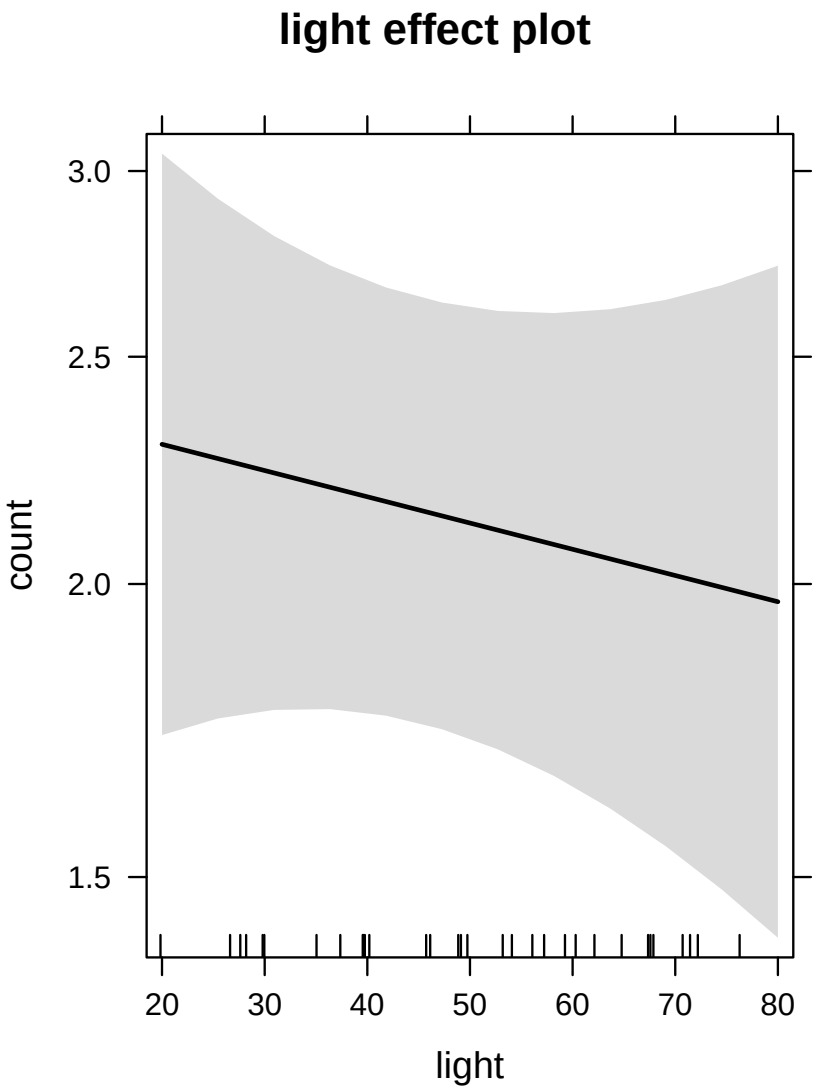
Use family `quasipoisson`

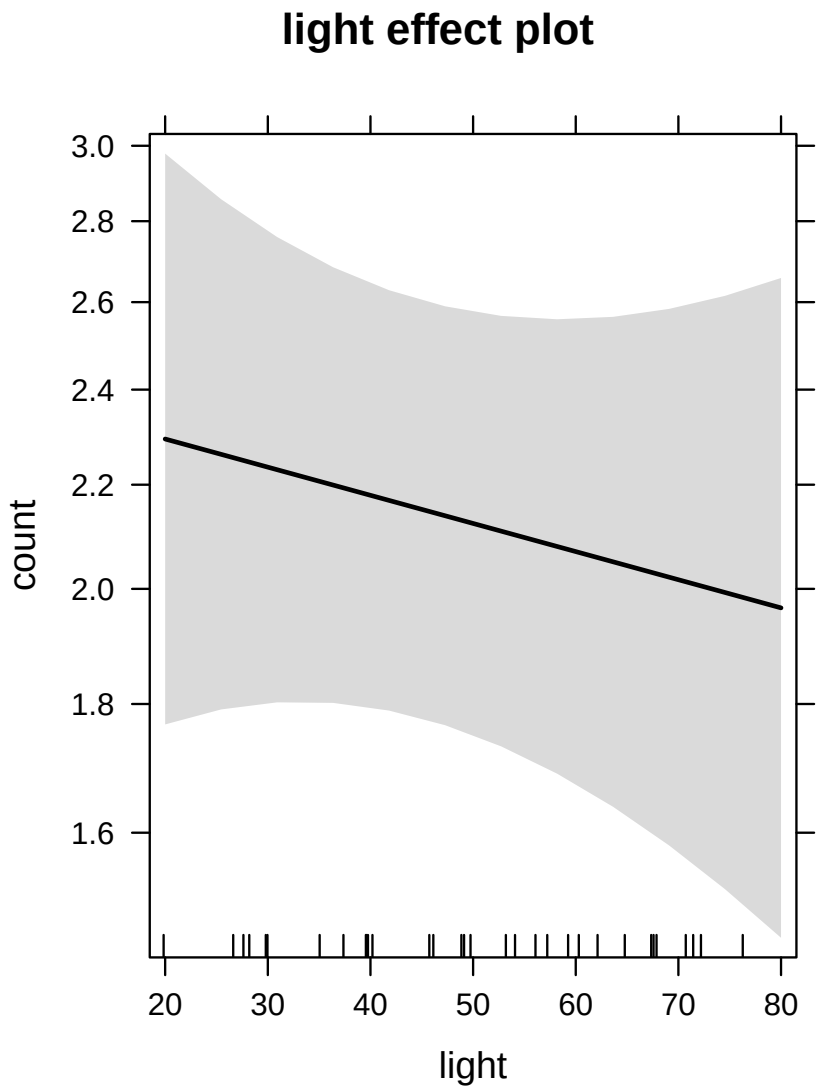
```
Call:
glm(formula = count ~ light, family = quasipoisson, data =

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1906  -0.8466  -0.1110   0.5220   2.4577

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.881805   0.201230   4.382 6.37e-05 ***
light       -0.002576   0.003758  -0.685    0.496
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '

(Dispersion parameter for quasipoisson family taken to be 1
```

Mean estimates do not change after accounting for overdispersion

```
 model: count ~ light

 light effect
light
      20       40       60       80
2.293988 2.178810 2.069414 1.965512

 model: count ~ light

 light effect
light
      20       40       60       80
2.293988 2.178810 2.069414 1.965512
```

But standard errors may change





Mixed / Multilevel Models

Mixed models enable us to account for variability

- Varying intercepts

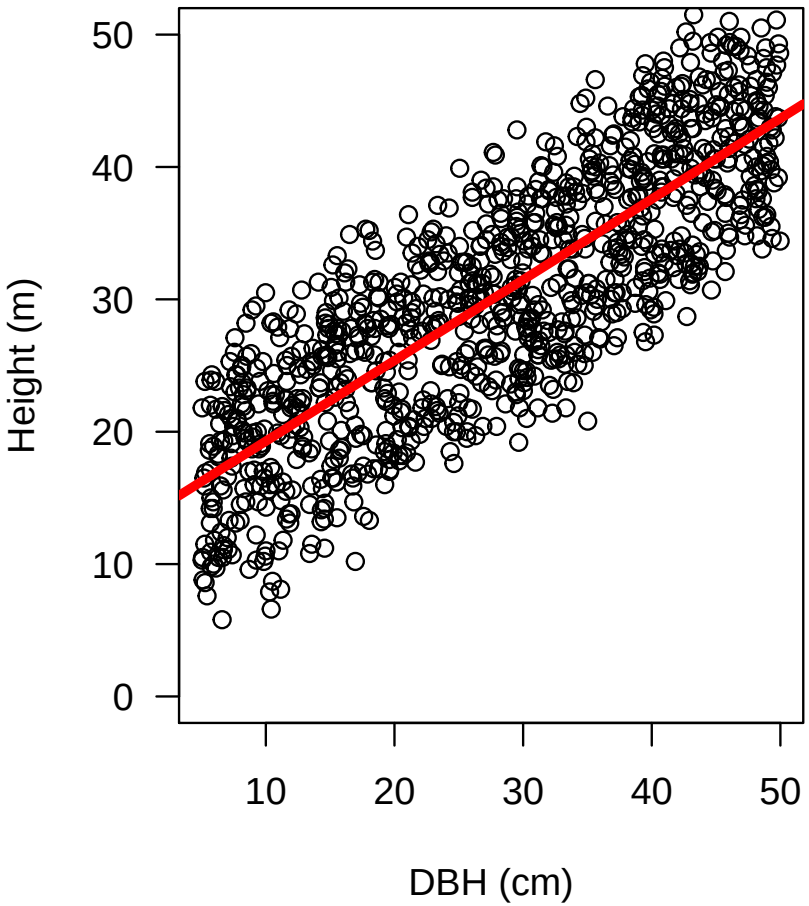
Mixed models enable us to account for variability

- Varying intercepts
- Varying slopes

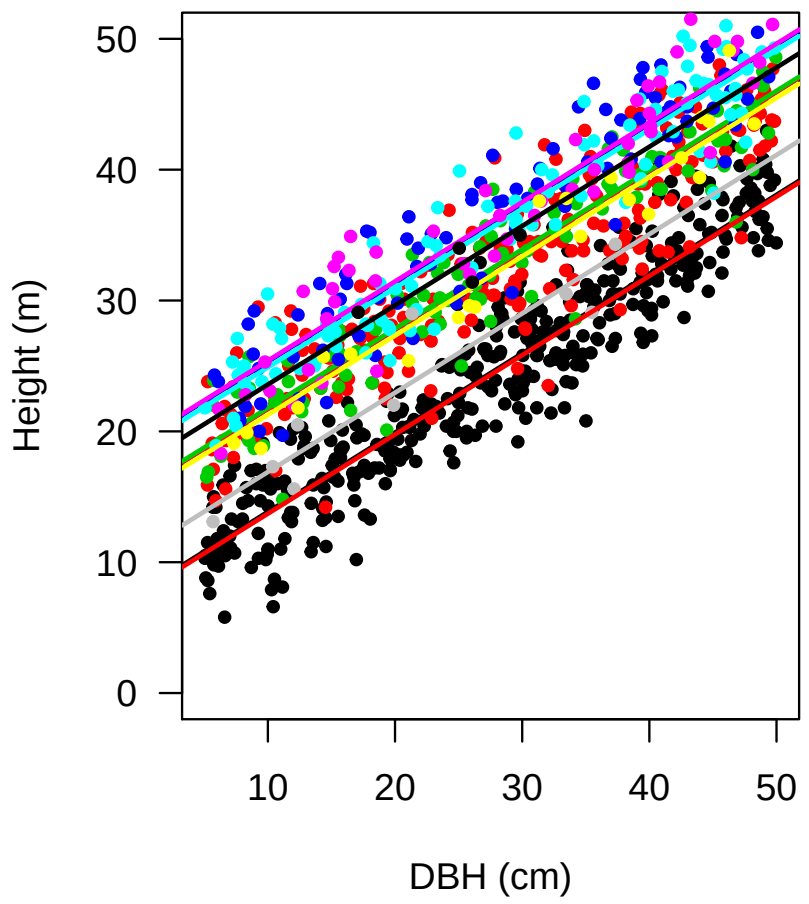
Single vs varying intercept

Dataset: 1000 trees from 10 plots (trees per plot: 4 - 392).

Pooling all plots



Different intercept for each plot



Fitting a varying intercepts model with `lm`

```
lm(formula = height ~ factor(plot) + dbh, data = trees)
                 coef.est coef.se
(Intercept)       7.79     0.24
factor(plot)2     7.86     0.24
factor(plot)3     7.95     0.32
factor(plot)4    11.48     0.33
factor(plot)5    11.05     0.32
factor(plot)6    11.55     0.43
factor(plot)7     7.41     0.63
factor(plot)8     3.05     0.97
factor(plot)9     9.73     1.45
factor(plot)10   -0.14     0.92
dbh               0.61     0.01
---
n = 1000, k = 11
residual sd = 2.89, R-Squared = 0.91
```

Mixed model with varying intercepts

$$y_i = a_j + bx_i + \varepsilon_i$$
$$a_j \sim N\left(0, \tau^2\right)$$
$$\varepsilon_i \sim N\left(0, \sigma^2\right)$$

En nuestro ejemplo:

$$Height_i = plot_j + bDBH_i + \varepsilon_i$$
$$plot_j \sim N\left(0, \tau^2\right)$$
$$\varepsilon_i \sim N\left(0, \sigma^2\right)$$

Fitting mixed/multilevel models

```
library(lme4)
mixed <- lmer(height ~ dbh + (1 | plot), data = trees)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: height ~ dbh + (1 | plot)
   Data: trees

REML criterion at convergence: 5007.6

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.84491 -0.65574 -0.02247  0.69295  3.09733

Random effects:
 Groups   Name        Variance Std.Dev.
 plot     (Intercept) 19.834   4.454
 Residual              8.325   2.885
Number of obs: 1000, groups:  plot, 10
```

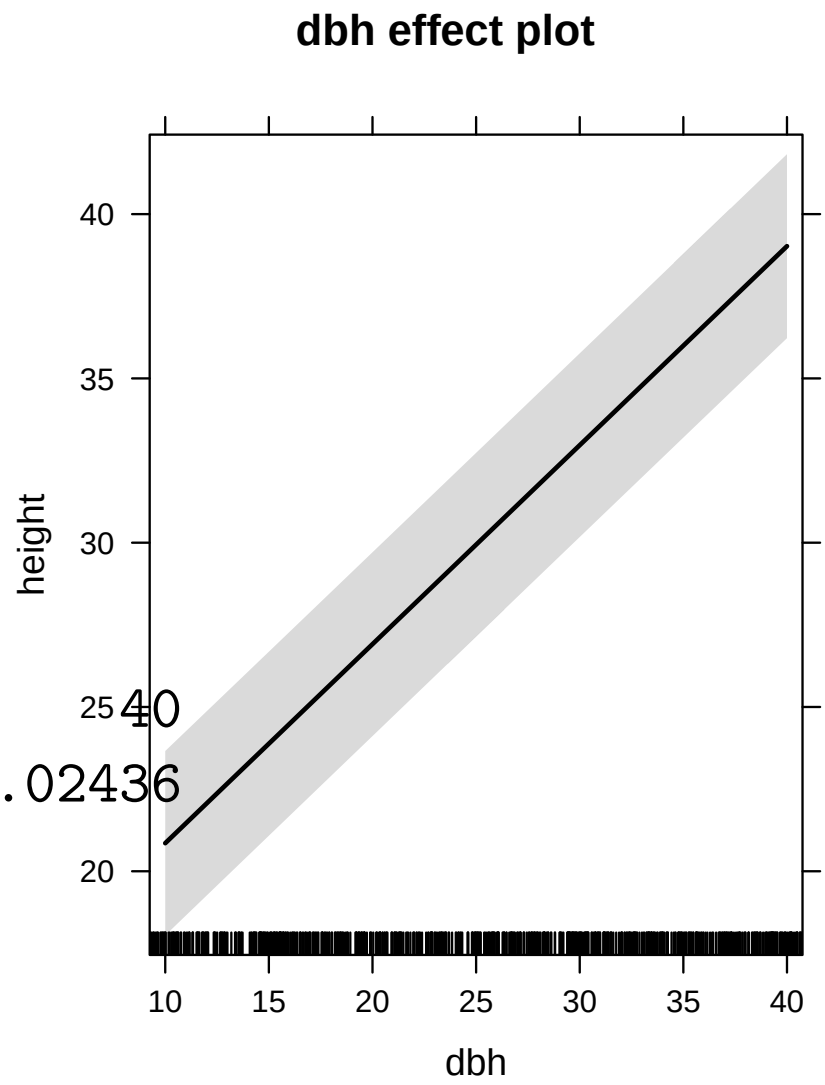
Retrieve model coefficients

```
coef(mixed)
```

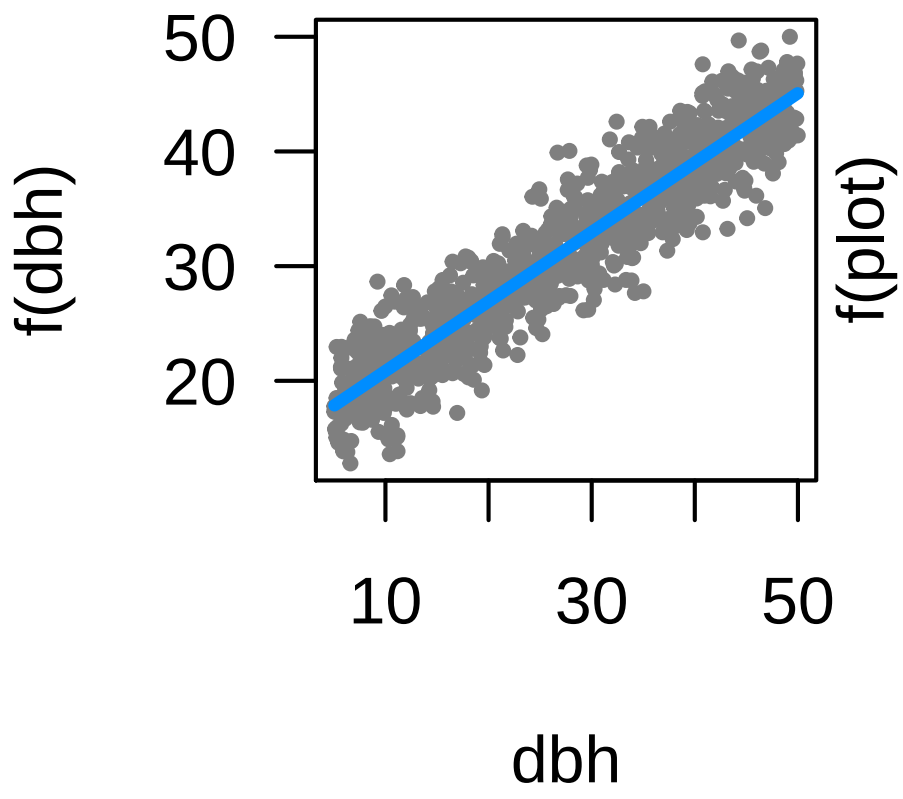
```
$plot
   (Intercept)        dbh
1     7.798373 0.6056549
2    15.647613 0.6056549
3    15.735397 0.6056549
4    19.253661 0.6056549
5    18.819467 0.6056549
6    19.306574 0.6056549
7    15.197908 0.6056549
8    11.016485 0.6056549
9    17.265447 0.6056549
10    7.940715 0.6056549

attr(,"class")
[1] "coef.mer"
```

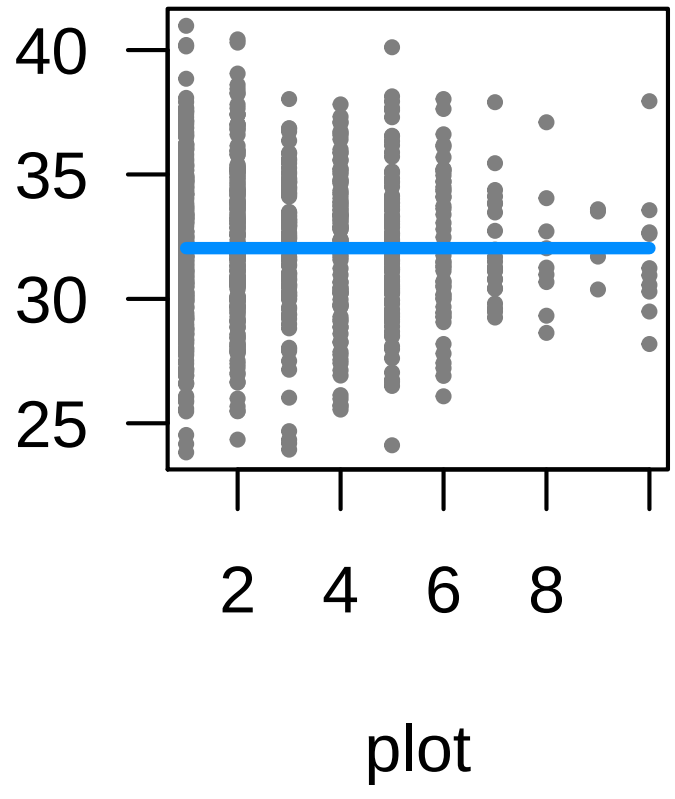
Visualising model: `allEffects`

```
 model: height ~ dbh

 dbh effect
dbh
      10       20       30       40
20.85471 26.91126 32.96781 39.02436
```



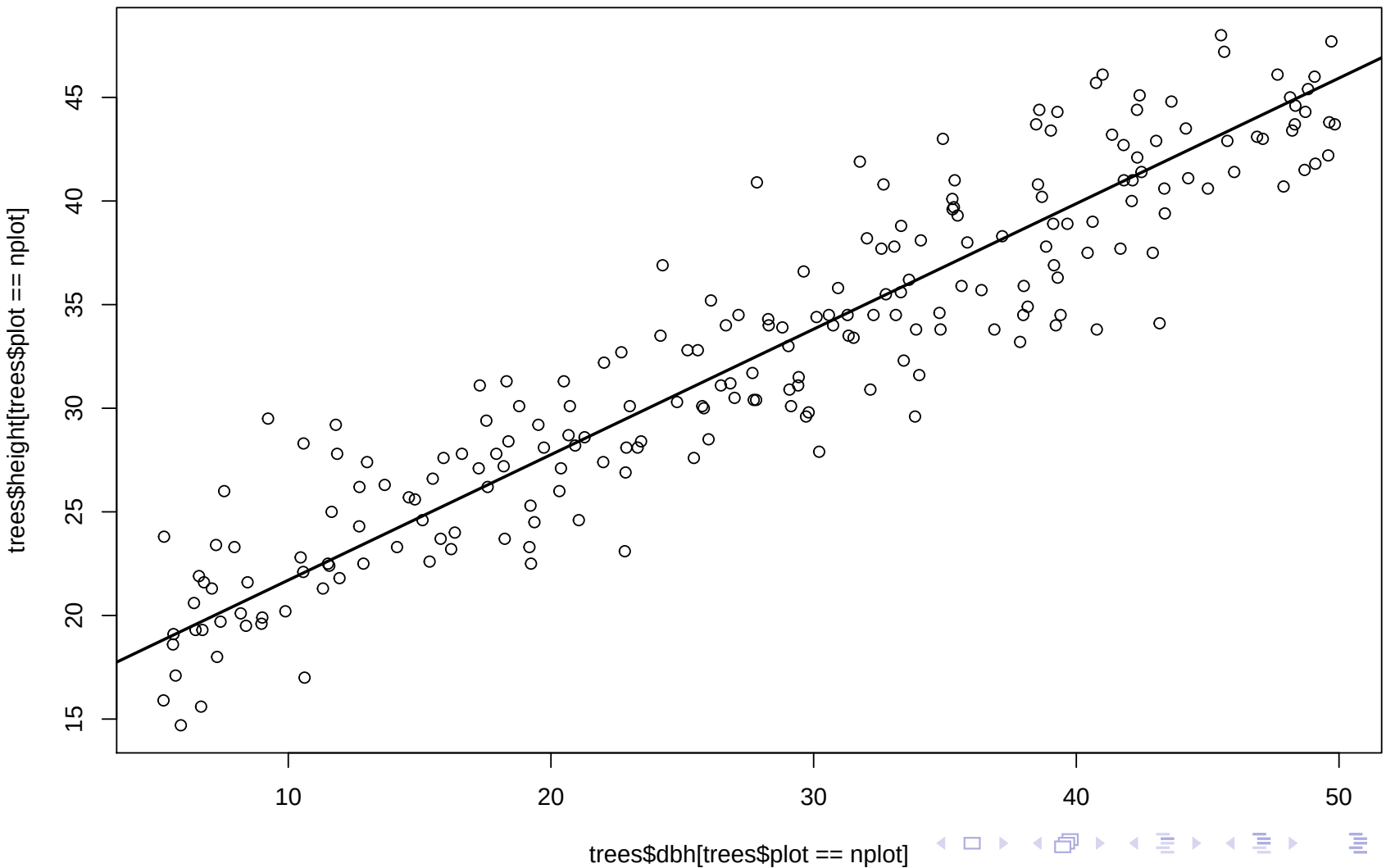
Visualising model: `visreg`





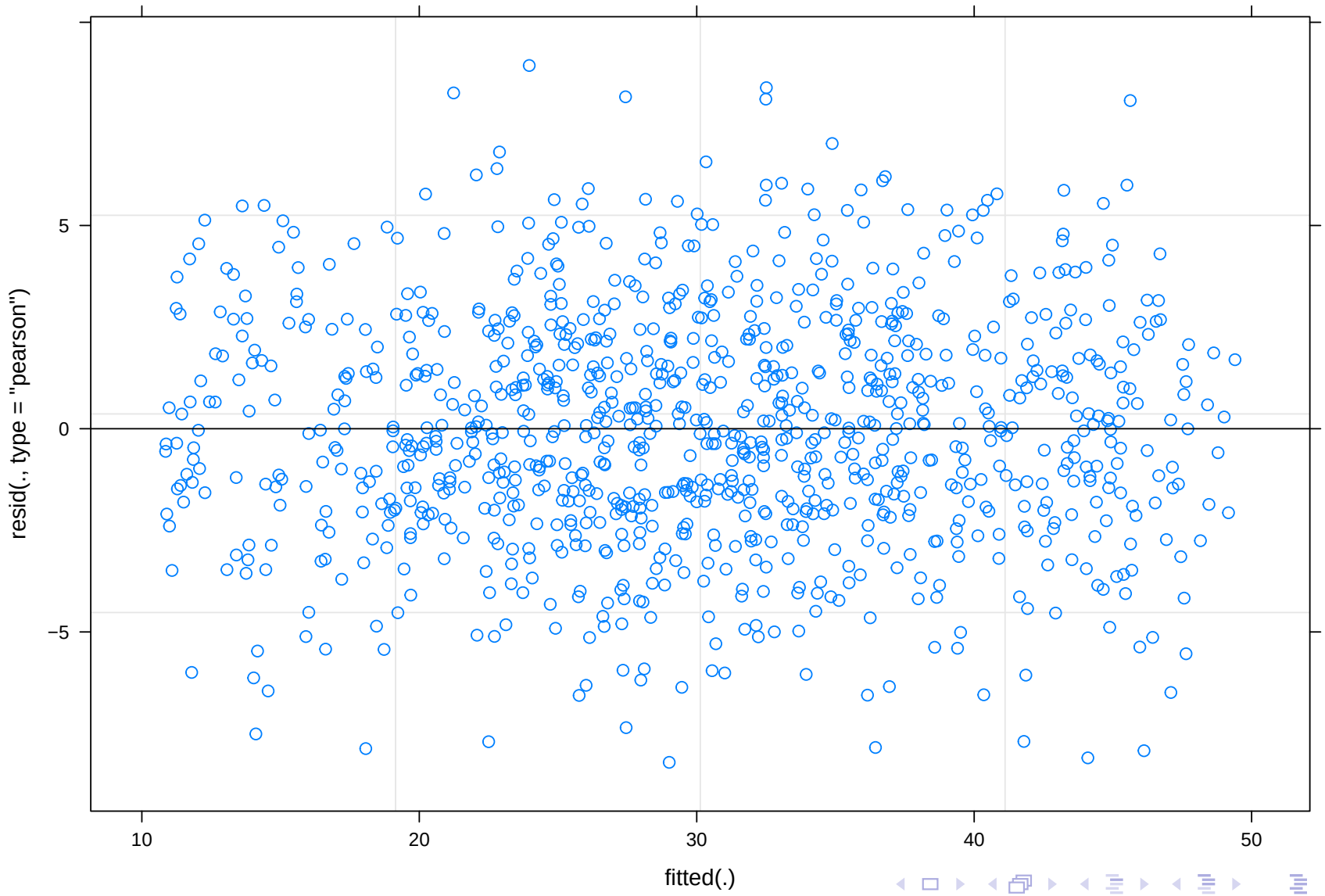
Plotting regression for individual forest plots

```
nplot <- 2
plot(trees$dbh[trees$plot == nplot], trees$height[trees$plo
abline(a = coef(mixed)$plot[nplot, 1], b = coef(mixed)$plot
```



Checking residuals

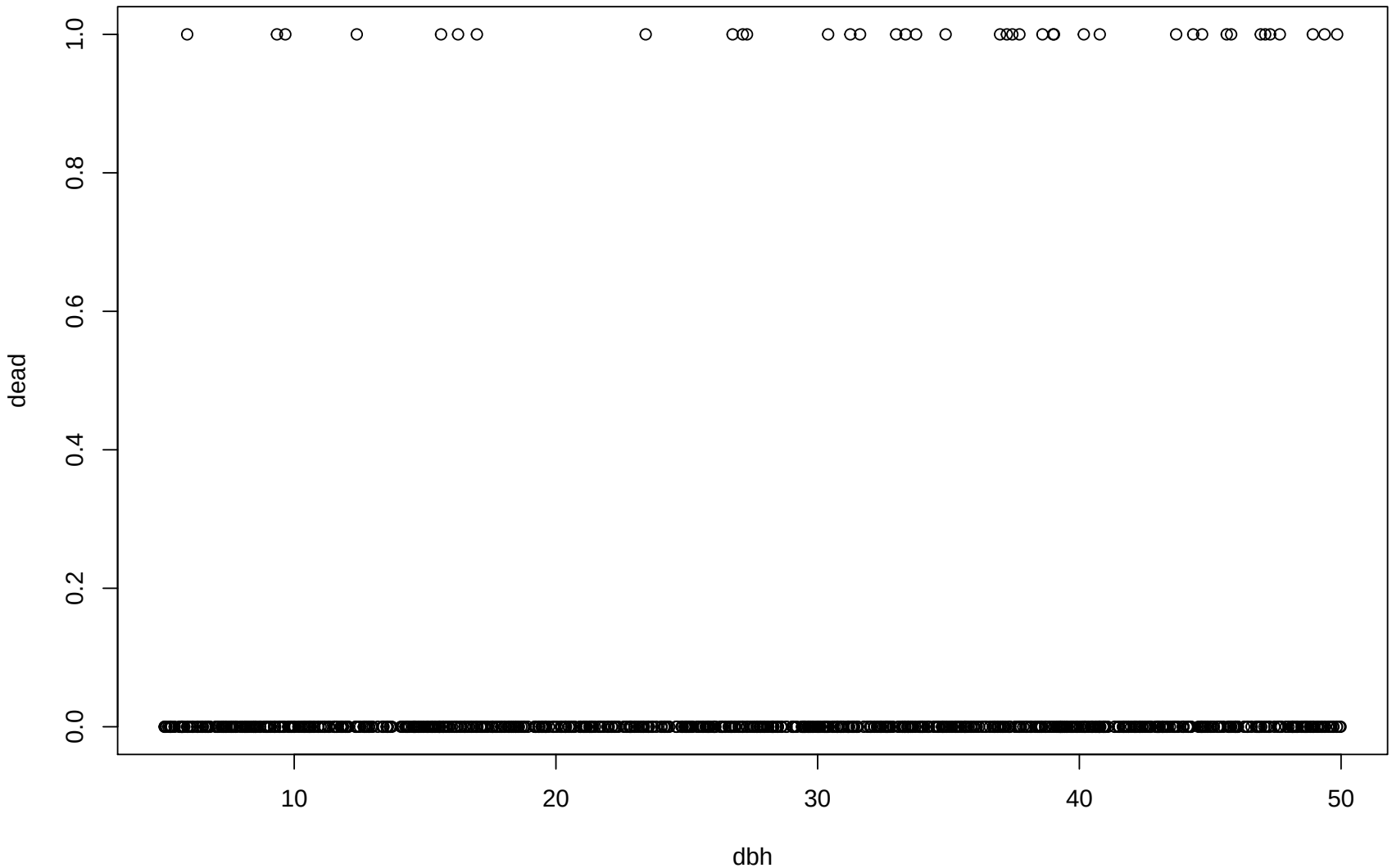
```
plot(mixed)
```



Multilevel logistic regression

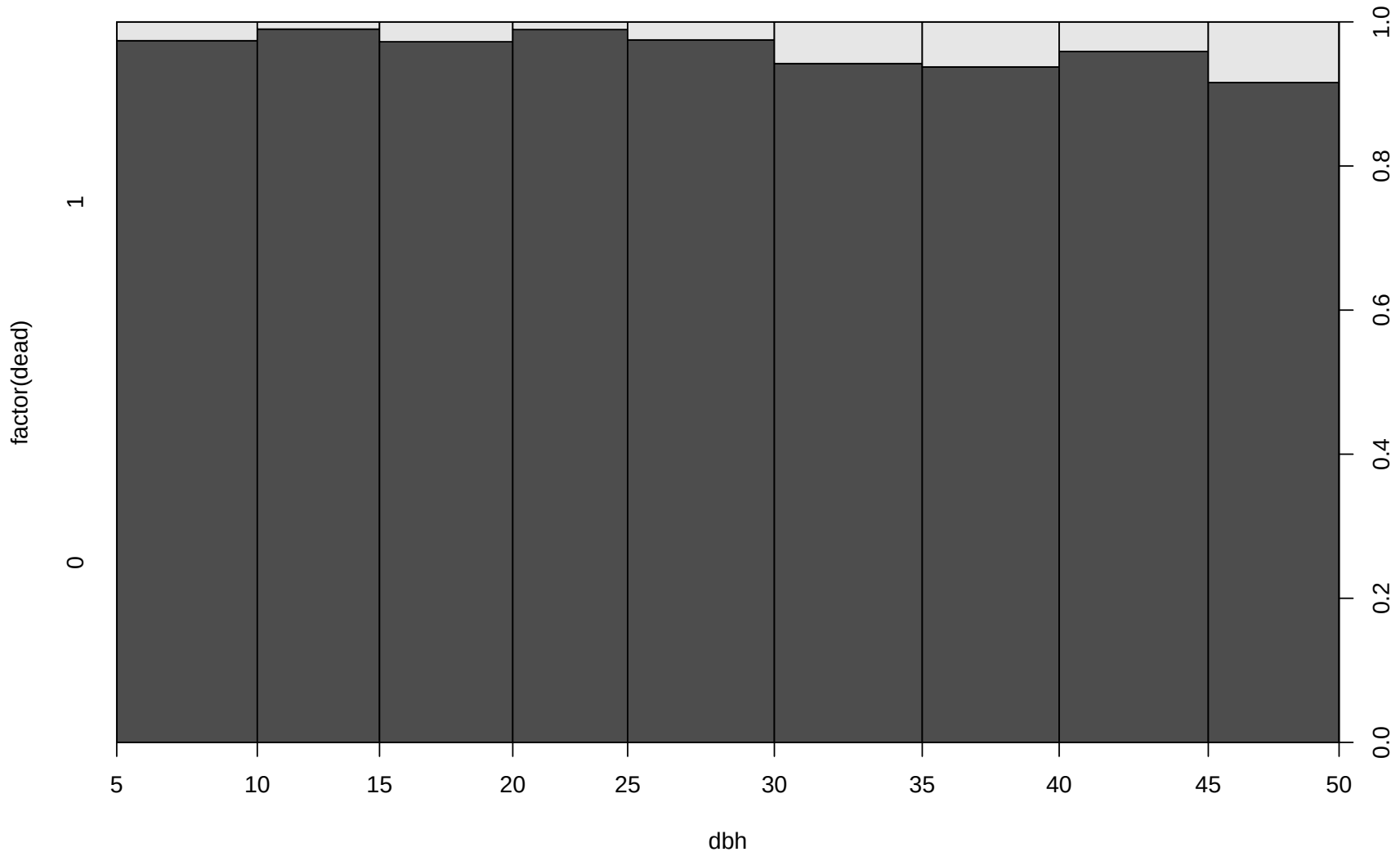
Q: Relationship between tree size and mortality

```
plot(dead ~ dbh, data = trees)
```



Q: Relationship between tree size and mortality

```
plot(factor(dead) ~ dbh, data = trees)
```



Fit simple logistic regression

```
simple.logis <- glm(dead ~ dbh, data = trees, family = binc
```

```
Call:
glm(formula = dead ~ dbh, family = binomial, data = trees)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.4121  -0.3287  -0.2624  -0.2048   2.9127

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.46945    0.49445  -9.039  < 2e-16 ***
dbh          0.04094    0.01380   2.967  0.00301 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '

(Dispersion parameter for binomial family taken to be 1)
```

Fit simple logistic regression (with plots)

```
logis2 <- glm(dead ~ dbh + factor(plot), data = trees, fami
```

```
Call:
glm(formula = dead ~ dbh + factor(plot), family = binomial,

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.5923  -0.3198  -0.2549  -0.1940   2.8902

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)     -4.40106    0.52997  -8.304   <2e-16 ***
dbh              0.04060    0.01386   2.929   0.0034 **
factor(plot)2   -0.59168    0.52132  -1.135   0.2564
factor(plot)3    0.54576    0.47094   1.159   0.2465
factor(plot)4    0.05507    0.57434   0.096   0.9236
factor(plot)5   -0.38312    0.64222  -0.597   0.5508
```

Fit multilevel logistic regression

```
mixed.logis <- glmer(dead ~ dbh + (1 | plot), data = trees
```

```
Generalized linear mixed model fit by maximum likelihood (
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: dead ~ dbh + (1 | plot)
   Data: trees

     AIC      BIC   logLik deviance df.resid
   325.9    340.6   -160.0    319.9      997

Scaled residuals:
    Min      1Q  Median      3Q     Max
-0.2977 -0.2356 -0.1872 -0.1456  8.2792

Random effects:
 Groups Name        Variance Std.Dev.
 plot   (Intercept) 0        0
```

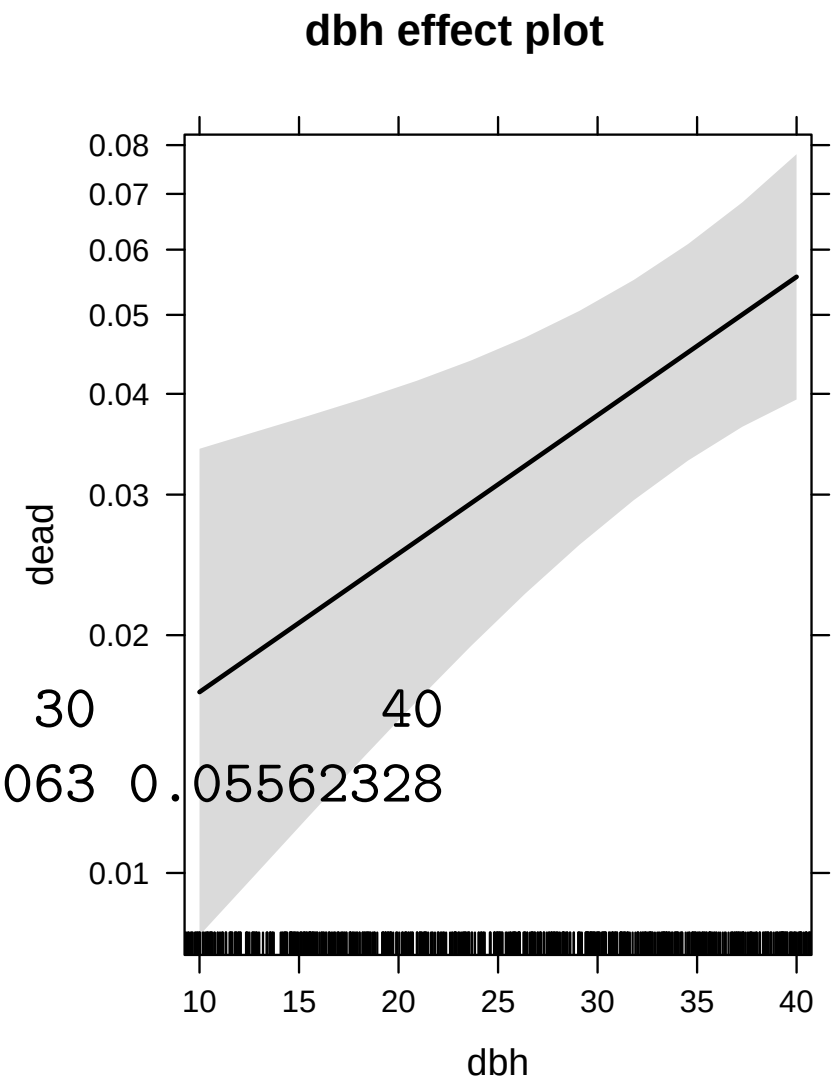
Retrieve model coefficients

```
coef(mixed.logis)
```

```
$plot
   (Intercept)         dbh
1    -4.469446 0.04093806
2    -4.469446 0.04093806
3    -4.469446 0.04093806
4    -4.469446 0.04093806
5    -4.469446 0.04093806
6    -4.469446 0.04093806
7    -4.469446 0.04093806
8    -4.469446 0.04093806
9    -4.469446 0.04093806
10   -4.469446 0.04093806

attr(,"class")
[1] "coef.mer"
```

Visualising model: `allEffects`

```
 model: dead ~ dbh

 dbh effect
dbh
        10         20         30         40
0.01695545 0.02531581 0.03764063 0.05562328
```



END

:)

Source code and materials:
`https://github.com/Pakillo/LM-GLM-GLMM-intro`