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

Francisco Rodriguez-Sanchez

http://bit.ly/frod_san

Introduction to linear models
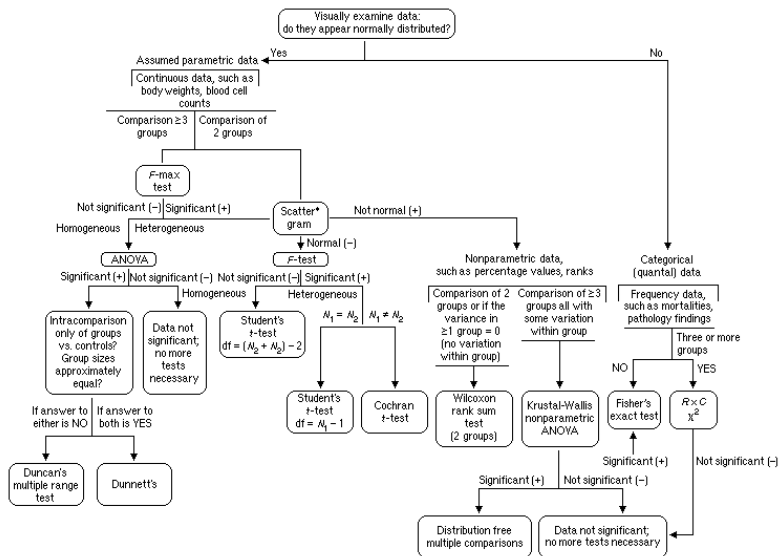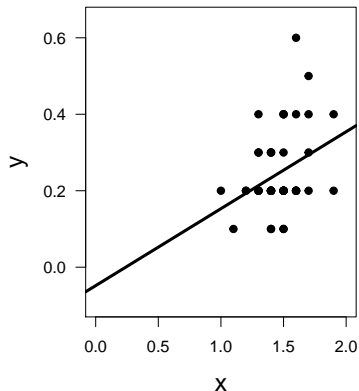
Modern statistics are easier than this

Figure 1:

Our overarching regression framework

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

Data
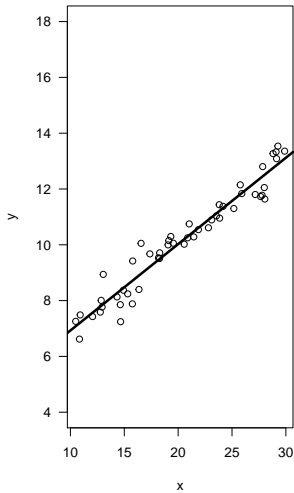y = response variable
x = predictor
Parameters
a = intercept
b = slope
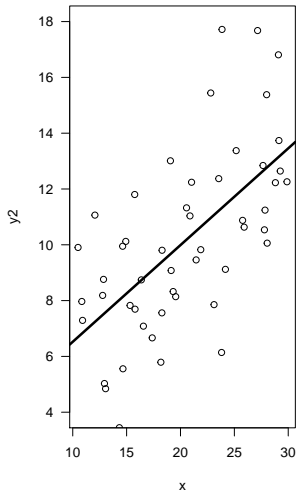σ = residual variation
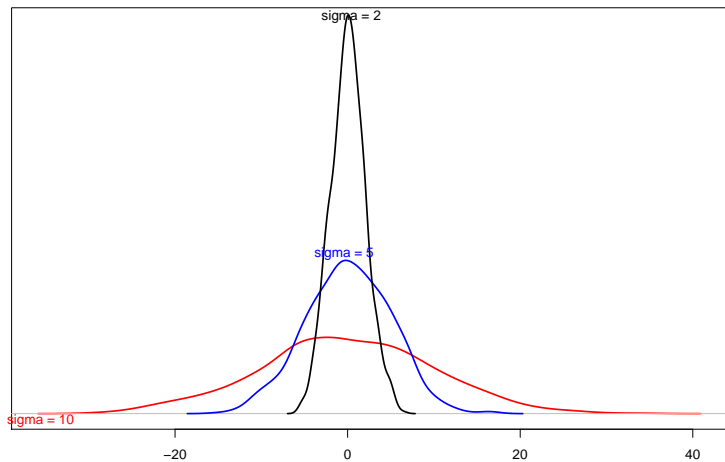ε = residuals

Residual variation (error)

Residual variation

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

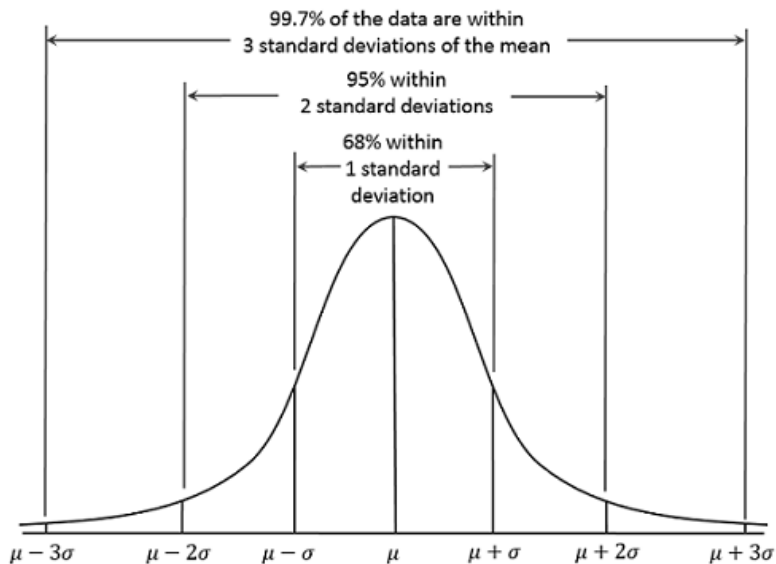**Distribution of residuals**

In a Normal distribution

Figure 2:

Different ways to write same model

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

.

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

Linear models

Example dataset: paper planes flying experiment

```
library(paperplanes)
head(paperplanes)
```

```
# A tibble: 6 x 8
     id hour    person   gender   age plane      paper distance
  <int> <fct>   <chr>    <fct>  <dbl> <chr>      <int>    <dbl>
1     1 [17,18) Roland   male     30. Standard80    80     7.80
2     2 [17,18) Astrid   female   30. Concorde120  120     2.70
3     3 [17,18) Roland   male     30. Standard120  120     9.20
4     4 [17,18) Isabella female   48. Standard120  120     6.00
5     5 [17,18) Fabienne female   17. Standard120  120     7.30
6     6 [17,18) Fabienne female   17. Standard120  120     7.80
```

Questions

- What is the relationship between age and distance flown?

Questions

- ▶ What is the relationship between age and distance flown?
- ▶ Do adults achieve longer distances?

Questions

- What is the relationship between age and distance flown?
- Do adults achieve longer distances?
- Can we predict distance flown from participant's age? How well?
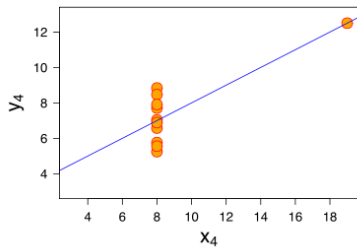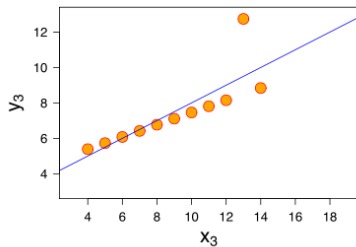
Always plot your data first!

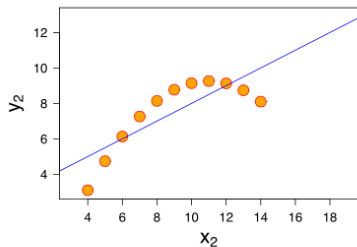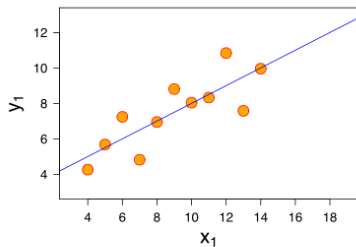
Figure 3:

Exploratory Data Analysis (EDA)

Outliers

```
plot(paperplanes$distance)
```
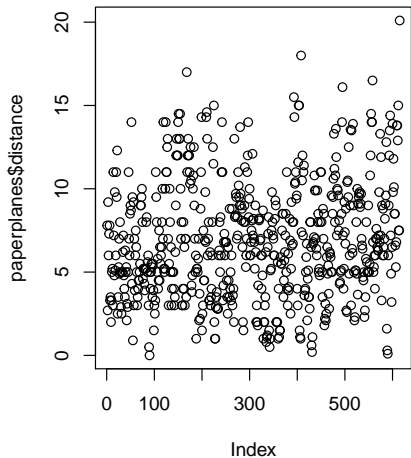
Outliers impact on regression

Figure 4:

See http://rpsychologist.com/d3/correlation/
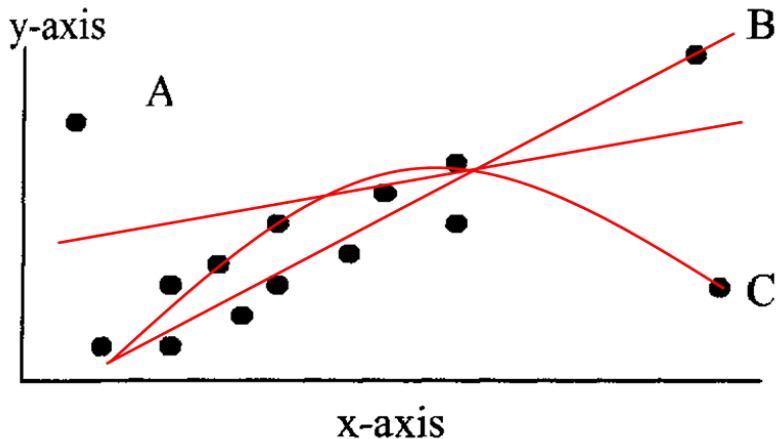
Histogram of response variable

```
hist(paperplanes$distance)
```


Histogram of paperplanes$distance
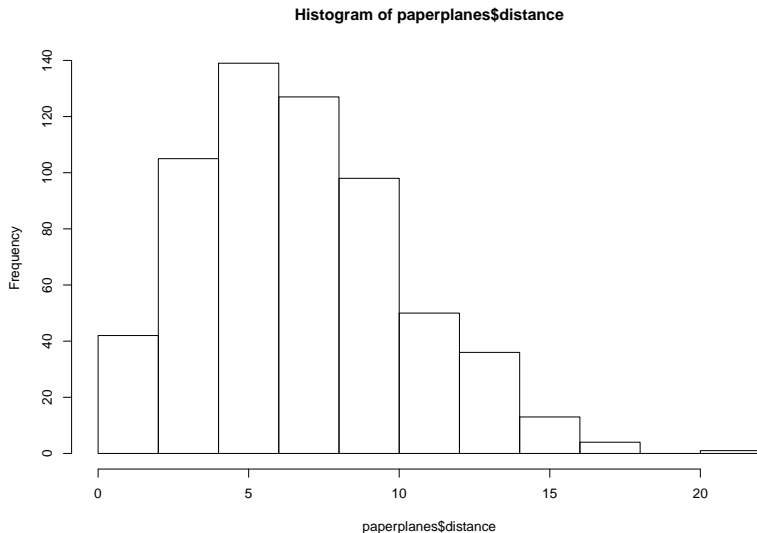
Histogram of predictor variable

```r
hist(paperplanes$age)
```


Histogram of paperplanes$age
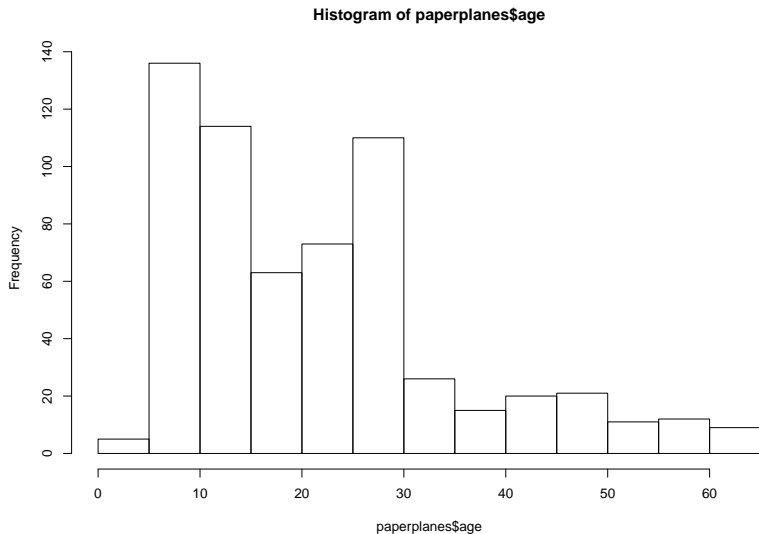
Scatterplot

```
plot(paperplanes$age, paperplanes$distance, las = 1)
```
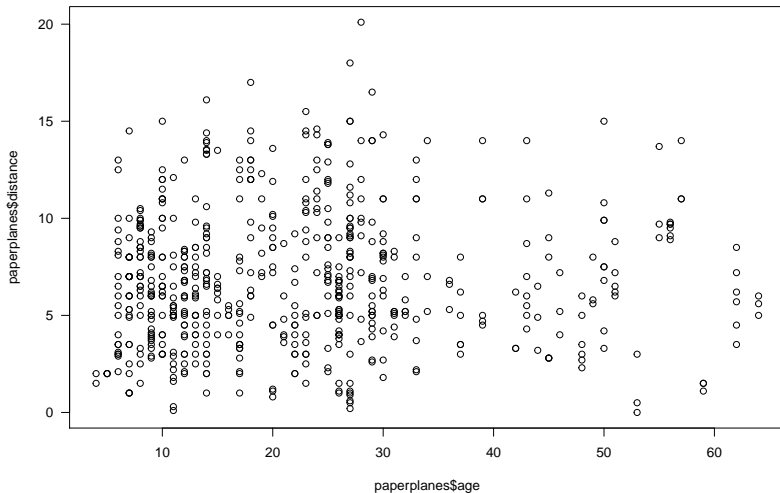
Now fit model

Hint: `lm`

Now fit model

Hint: `lm`

```
m1 <- lm(distance ~ age, data = paperplanes)
```

What does this mean?

```
Call:
lm(formula = distance ~ age, data = paperplanes)

Residuals:
    Min      1Q  Median      3Q     Max
-7.1929 -2.6014 -0.3789  2.1572 13.1658

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.64440    0.26982  24.626   <2e-16 ***
age          0.01035    0.01040   0.996     0.32
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.504 on 613 degrees of freedom
Multiple R-squared: 0.001614,   Adjusted R-squared: -1.434e-05
F-statistic: 0.9912 on 1 and 613 DF,  p-value: 0.3198
```

Retrieving model coefficients

```
coef(m1)
```

```
(Intercept)          age
 6.64439782  0.01034968
```

Tidy up model coefficients with broom

```r
library(broom)
tidy(m1)
```

```
         term     estimate  std.error  statistic     p.value
1 (Intercept) 6.64439782 0.26981611 24.6256529 1.286330e-93
2         age 0.01034968 0.01039555  0.9955879 3.198432e-01
```

```r
glance(m1)
```

```
   r.squared adj.r.squared    sigma statistic   p.value df    l
1 0.001614348  -1.43402e-05 3.503736 0.9911952 0.3198432  2 -164
       AIC      BIC deviance df.residual
1 3291.502 3304.767 7525.289         613
```

Confidence intervals

```
confint(m1)
```

```
                 2.5 %     97.5 %
(Intercept)  6.11452177 7.17427388
age         -0.01006553 0.03076489
```

Using effects package

```
library(effects)
summary(allEffects(m1))

 model: distance ~ age

 age effect
age
       4        20        30        50        60
6.685797 6.851391 6.954888 7.161882 7.265379

 Lower 95 Percent Confidence Limits
age
       4        20        30        50        60
6.223509 6.570601 6.634085 6.528536 6.443633

 Upper 95 Percent Confidence Limits
age
       4        20        30        50        60
7.148084 7.132182 7.275692 7.795228 8.087125
```
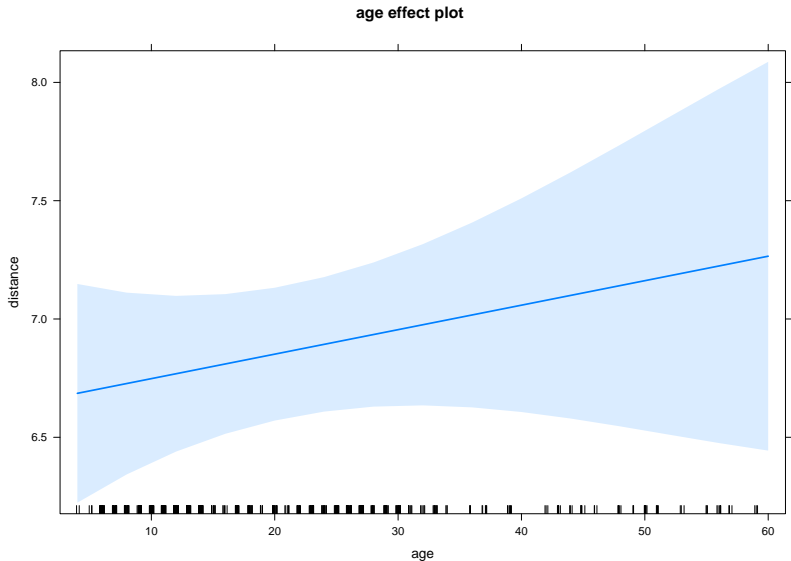
Plot effects

```
plot(allEffects(m1))
```


age effect plot

Plot model (visreg)

```
library(visreg)
visreg(m1)
```
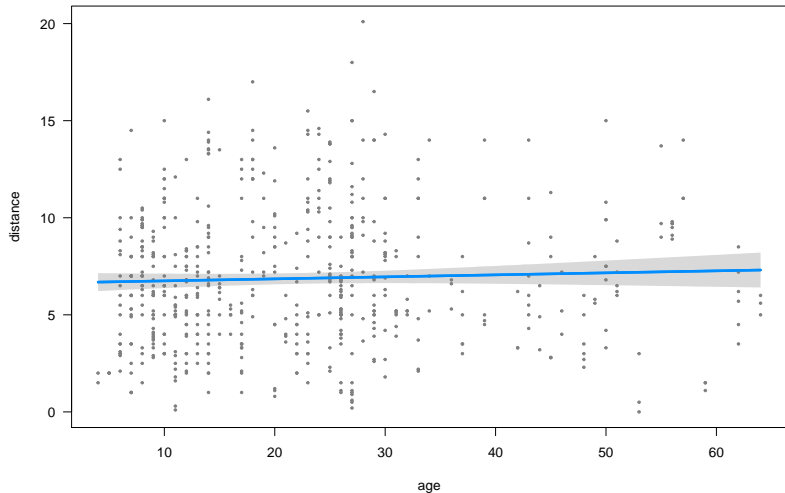
Linear model assumptions

- Linearity (transformations, GAM. . .)

Linear model assumptions

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

Linear model assumptions

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

Linear model assumptions

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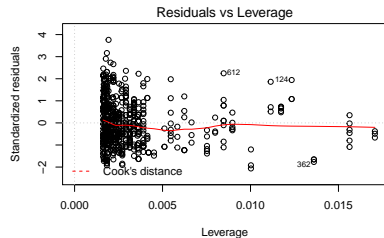- Residuals:
  - Independent
  - Equal variance

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
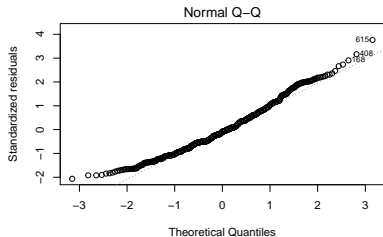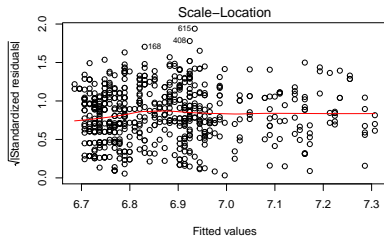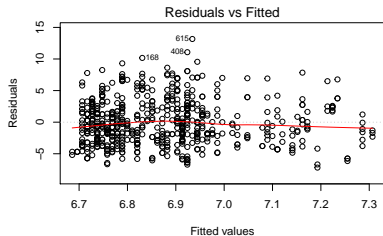
Are residuals normal?

```
hist(resid(m1))
```

Histogram of resid(m1)
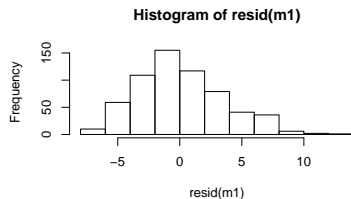

```
lm(formula = distance ~ age, data = paperplane
            coef.est coef.se
(Intercept) 6.64     0.27
age         0.01     0.01
---
n = 615, k = 2
residual sd = 3.50, R-Squared = 0.00
```

SD of residuals = 3.5 coincides with estimate of sigma.

How good is the model in predicting distance?

`fitted` gives predictions for each observation

```
paperplanes$distance.pred <- fitted(m1)
head(paperplanes)
```

```
# A tibble: 6 x 9
     id hour    person   gender   age plane   paper distance distance.pred
  <int> <fct>   <chr>    <fct>  <dbl> <chr>   <int>    <dbl>         <dbl>
1     1 [17,18) Roland   male     30. Standa~    80     7.80          6.95
2     2 [17,18) Astrid   female   30. Concor~   120     2.70          6.95
3     3 [17,18) Roland   male     30. Standa~   120     9.20          6.95
4     4 [17,18) Isabella female   48. Standa~   120     6.00          7.14
5     5 [17,18) Fabienne female   17. Standa~   120     7.30          6.82
6     6 [17,18) Fabienne female   17. Standa~   120     7.80          6.82
```

Calibration plot: Observed vs Predicted values

```
plot(paperplanes$distance.pred, paperplanes$distance, xlab = "Di
```
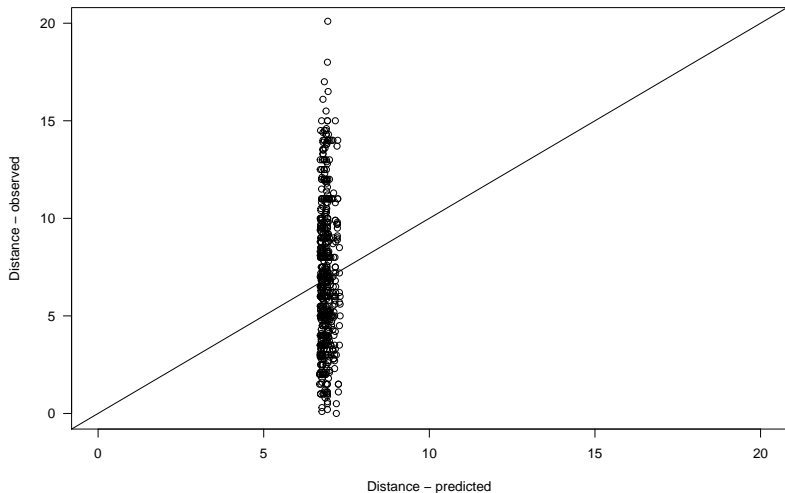
Using fitted model for prediction

Q: Expected distance if age = 30?

```
new.age <- data.frame(age = c(30))
predict(m1, new.age, se.fit = TRUE)

$fit
       1
6.954888

$se.fit
[1] 0.1633552

$df
[1] 613

$residual.scale
[1] 3.503736
```

Using fitted model for prediction

Q: Expected distance if age = 30?

```
new.age <- data.frame(age = c(30))
predict(m1, new.age, se.fit = TRUE, interval = "confidence", lev
```

```
$fit
       fit      lwr      upr
1 6.954888 6.634085 7.275692

$se.fit
[1] 0.1633552

$df
[1] 613

$residual.scale
[1] 3.503736
```

Using fitted model for prediction

Q: Expected distance if age $= 30$?

```
new.age <- data.frame(age = c(30))
predict(m1, new.age, se.fit = TRUE, interval = "prediction", lev

$fit
       fit        lwr      upr
1 6.954888 0.06663211 13.84314

$se.fit
[1] 0.1633552

$df
[1] 613

$residual.scale
[1] 3.503736
```

Important functions

- plot

Important functions

- `plot`
- `summary`

Important functions

- `plot`
- `summary`
- `coef`

Important functions

- `plot`
- `summary`
- `coef`
- `confint`

Important functions

- `plot`
- `summary`
- `coef`
- `confint`
- `fitted`

Important functions

- `plot`
- `summary`
- `coef`
- `confint`
- `fitted`
- `resid`

Important functions

- `plot`
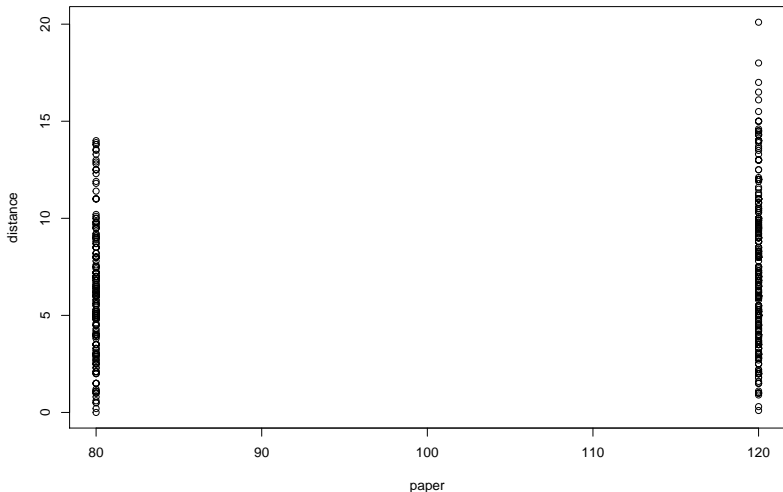- `summary`
- `coef`
- `confint`
- `fitted`
- `resid`
- `allEffects`

Important functions

- `plot`
- `summary`
- `coef`
- `confint`
- `fitted`
- `resid`
- `allEffects`
- `predict`

Categorical predictors (factors)

Q: Does distance vary with paper type?

```
plot(distance ~ paper, data = paperplanes)
```

Model distance ~ paper

All right here?

```r
m2 <- lm(distance ~ paper, data = paperplanes)
```

```
Call:
lm(formula = distance ~ paper, data = paperplanes)

Residuals:
    Min      1Q  Median      3Q     Max
-7.2756 -2.3756 -0.3756  2.2244 12.7244

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.638290   0.750041   4.851 1.56e-06 ***
paper       0.031144   0.007095   4.389 1.34e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.453 on 613 degrees of freedom
Multiple R-squared: 0.03047,    Adjusted R-squared: 0.02889
```

Model distance ~ paper

Paper is a factor!

```r
paperplanes$paper <- as.factor(paperplanes$paper)
```

```
      id              hour          person             gender
 Min.   :  1.0   [19,20) :139   Length:615        female:213
 1st Qu.:154.5   [22,23) :108   Class :character   male  :402
 Median :308.0   [21,22) : 89   Mode  :character
 Mean   :308.0   [18,19) : 86
 3rd Qu.:461.5   [23,Inf): 78
 Max.   :615.0   [17,18) : 75
                 (Other) : 40
     age            plane           paper       distance
 Min.   : 4.00   Length:615       80 :248   Min.   : 0.000
 1st Qu.:11.00   Class :character  120:367   1st Qu.: 4.350
 Median :20.00   Mode  :character           Median : 6.500
 Mean   :22.11                              Mean   : 6.873
 3rd Qu.:28.00                              3rd Qu.: 9.000
 Max.   :64.00                              Max.   :20.100

 distance.pred
```

Model distance ~ paper

```r
m2 <- lm(distance ~ paper, data = paperplanes)
```

```
Call:
lm(formula = distance ~ paper, data = paperplanes)

Residuals:
    Min      1Q  Median      3Q     Max
-7.2756 -2.3756 -0.3756  2.2244 12.7244

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.1298     0.2192  27.958  < 2e-16 ***
paper120      1.2458     0.2838   4.389 1.34e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.453 on 613 degrees of freedom
Multiple R-squared:  0.03047,   Adjusted R-squared:  0.02889
F-statistic: 19.27 on 1 and 613 DF,  p-value: 1.339e-05
```

Linear model with categorical predictors

$$y_i = a + bx_i + \varepsilon_i$$
$$y_i = a + b_{paper120} + \varepsilon_i$$

Model distance ∼ paper

```r
m2 <- lm(distance ~ paper, data = paperplanes)
```

```
Call:
lm(formula = distance ~ paper, data = paperplanes)

Residuals:
    Min      1Q  Median      3Q     Max
-7.2756 -2.3756 -0.3756  2.2244 12.7244

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.1298     0.2192  27.958  < 2e-16 ***
paper120      1.2458     0.2838   4.389 1.34e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.453 on 613 degrees of freedom
Multiple R-squared:  0.03047,   Adjusted R-squared:  0.02889
F-statistic: 19.27 on 1 and 613 DF,  p-value: 1.339e-05
```

Effects: Estimated Distance ~ paper

```
summary(allEffects(m2))

 model: distance ~ paper

 paper effect
paper
      80       120
6.129839 7.375613

 Lower 95 Percent Confidence Limits
paper
      80       120
5.699269 7.021668

 Upper 95 Percent Confidence Limits
paper
      80       120
6.560408 7.729558
```
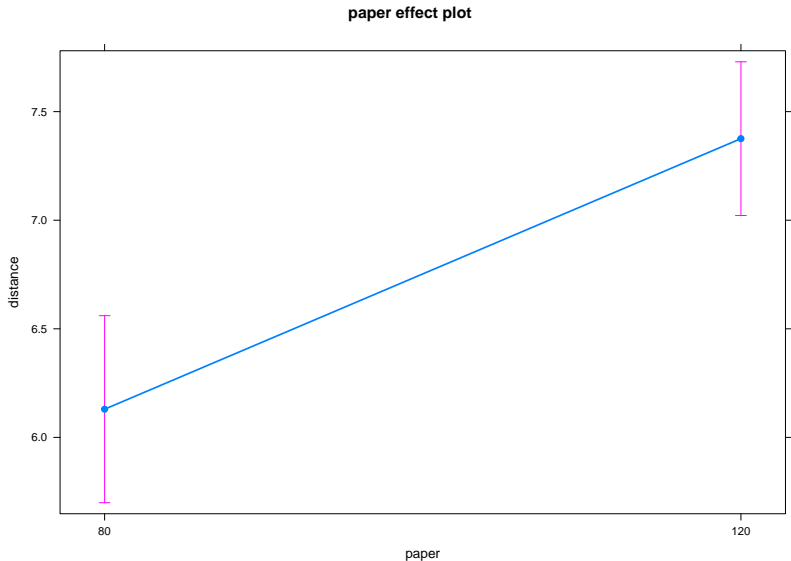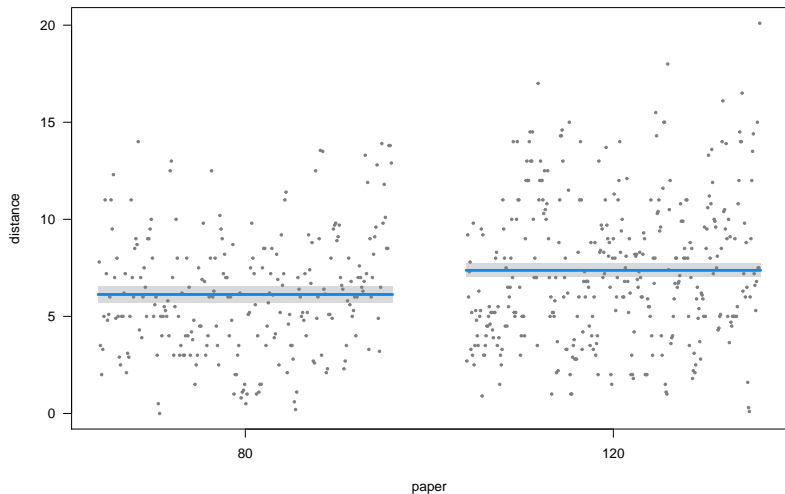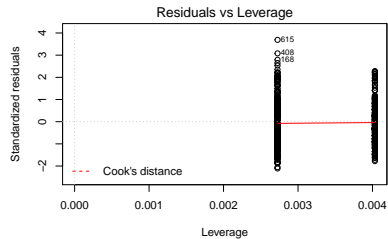
Plot
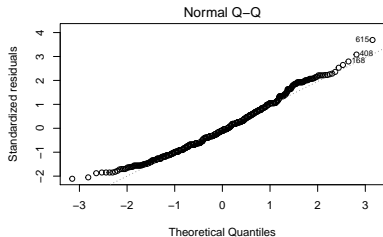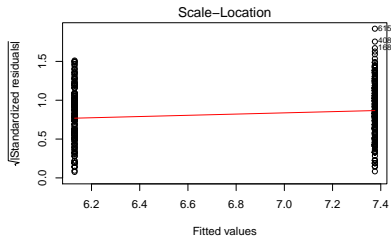
```
plot(allEffects(m2))
```


paper effect plot

Plot (visreg)

```
visreg(m2)
```

Model checking: residuals
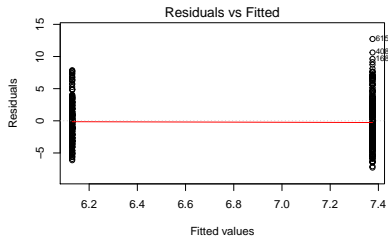
Model checking: residuals

```
hist(resid(m2))
```


Histogram of resid(m2)
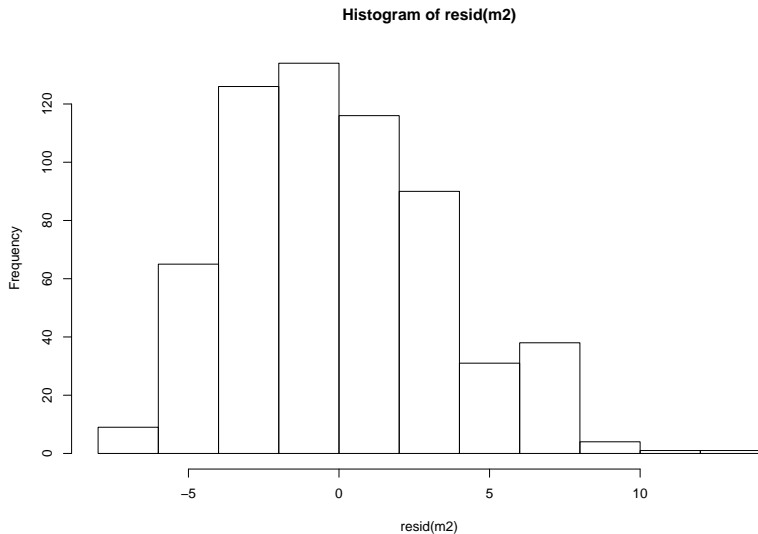
Exercise: Does distance vary with gender?

Combining continuous and categorical predictors

Predicting distance based on age and paper type

$$y_i = a + bx_i + \varepsilon_i$$
$$y_i = a + b_{paper120} + c \cdot age_i + \varepsilon_i$$

Predicting distance based on age and paper type

```
Call:
lm(formula = distance ~ age + paper, data = paperplanes)

Residuals:
    Min      1Q  Median      3Q     Max
-7.1092 -2.4753 -0.3576  2.2523 12.5892

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.69210    0.33641  16.920  < 2e-16 ***
age          0.01774    0.01035   1.714   0.0871 .
paper120     1.32192    0.28683   4.609 4.93e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.447 on 612 degrees of freedom
Multiple R-squared:  0.0351,    Adjusted R-squared:  0.03195
F-statistic: 11.13 on 2 and 612 DF,  p-value: 1.784e-05
```

Estimated distance

```
summary(allEffects(multreg))
```

```
model: distance ~ age + paper

age effect
age
      4        20        30        50        60
6.551921 6.835779 7.013191 7.368014 7.545425

Lower 95 Percent Confidence Limits
age
      4        20        30        50        60
6.093516 6.559431 6.696578 6.738709 6.728156

Upper 95 Percent Confidence Limits
age
      4        20        30        50        60
7.010326 7.112127 7.329803 7.997318 8.362694

paper effect
paper
     80       120
6.084400 7.406318

Lower 95 Percent Confidence Limits
paper
     80       120
5.651366 7.051182

Upper 95 Percent Confidence Limits
```
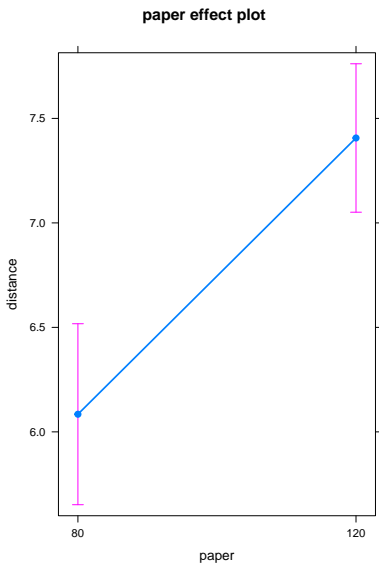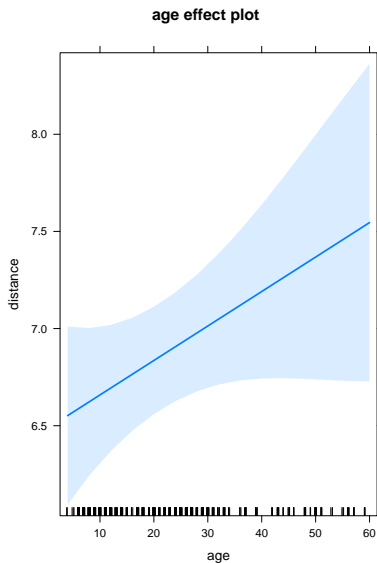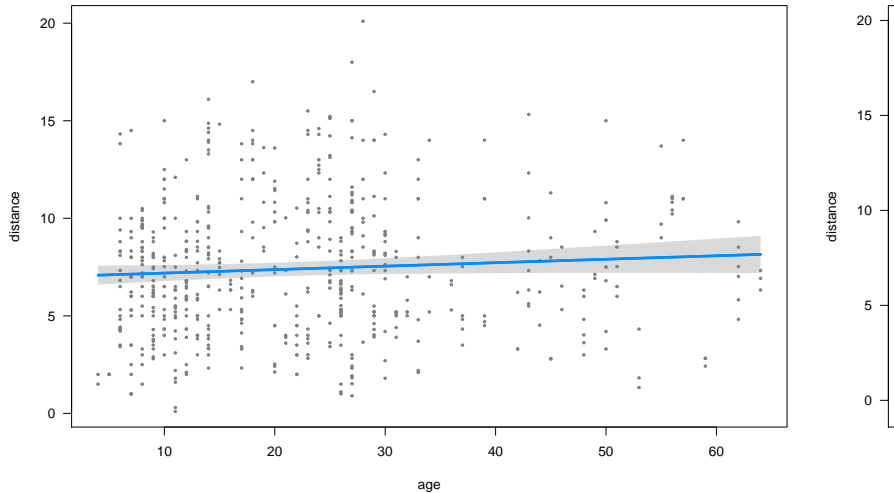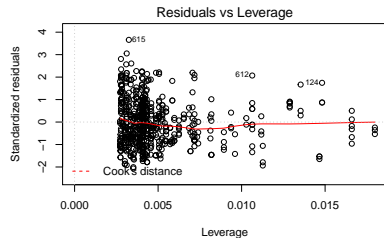
Plot

```
plot(allEffects(multreg))
```

Plot (visreg)

```
visreg(multreg)
```

Model checking: residuals
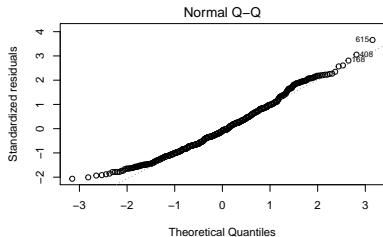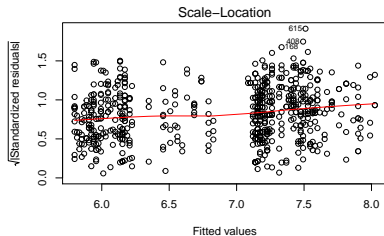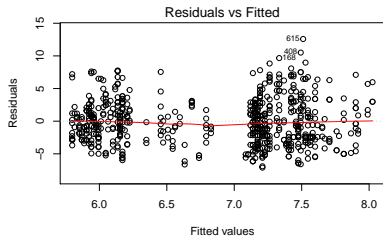
How good is this model? Calibration plot

```
paperplanes$distance.pred <- fitted(multreg)
plot(paperplanes$distance.pred, paperplanes$distance, xlab = "Di
abline(a = 0, b = 1)
```
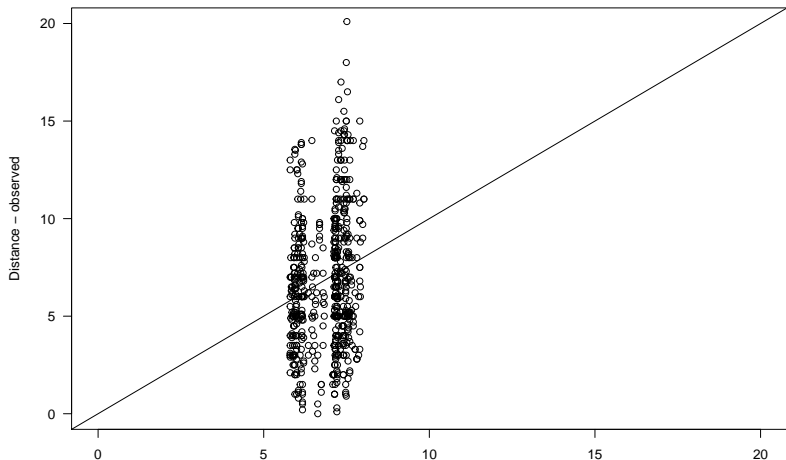
Model checking with simulated data

```
library(bayesplot)
sims <- simulate(multreg, nsim = 100)
ppc_dens_overlay(paperplanes$distance, yrep = t(as.matrix(sims)))
```
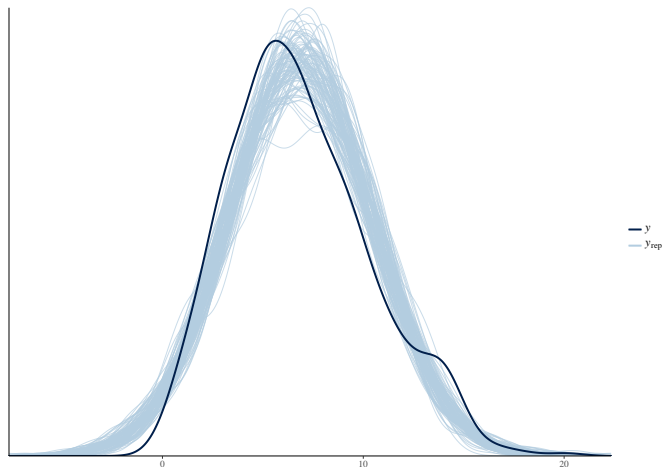
Extra exercises

- mammal sleep: Are sleep patterns related to diet?

Extra exercises

- mammal sleep: Are sleep patterns related to diet?
- iris: Predict petal length $\sim$ petal width and species

Generalised Linear Models: Logistic regression

Q: Survival of passengers on the Titanic ~ Class

Read `titanic_long.csv` dataset.

```
  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)
```
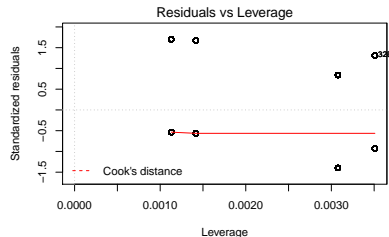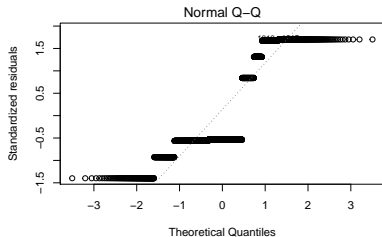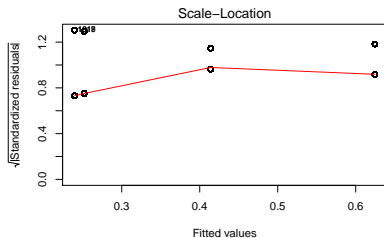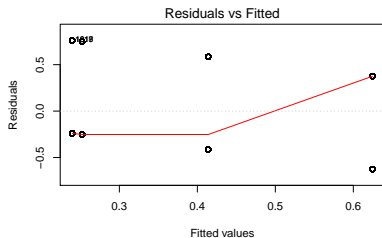
Weird residuals!
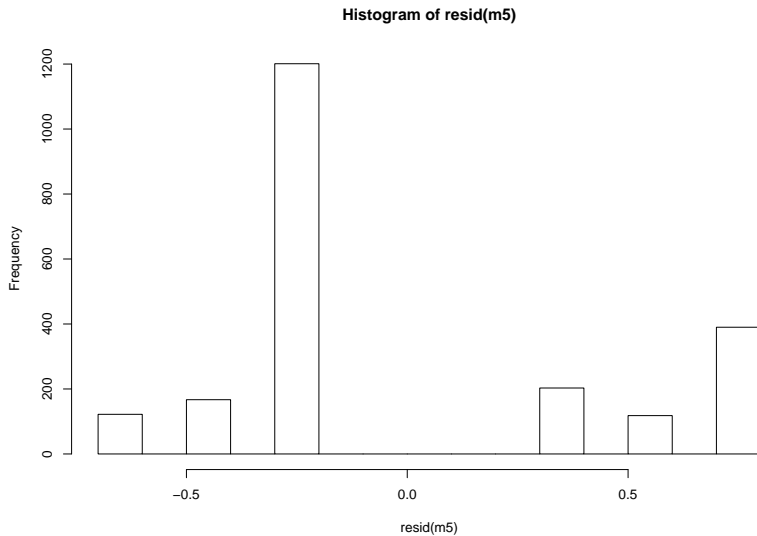
Histogram of resid(m5)

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`.

The modelling process
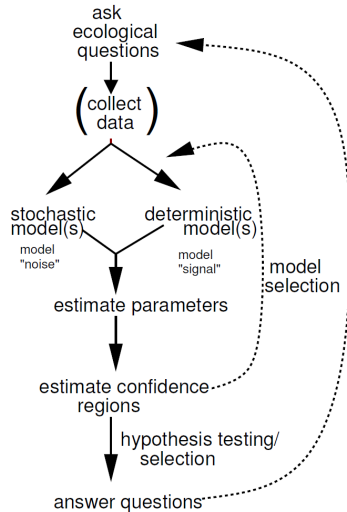
Figure 1.5 Flow of the modeling process.

Figure 5:

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?

Back to survival of Titanic passengers

How many passengers travelled in each class?

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

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

Back to survival of Titanic passengers

How many passengers travelled in each class?

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

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

How many survived?

Back to survival of Titanic passengers

How many passengers travelled in each class?

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

```
  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
```

Back to survival of Titanic passengers

How many passengers travelled in each class?

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

```
  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?

```
as.numeric(tapply(titanic$survived, titanic$class, mean))
```

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

Back to survival of Titanic passengers (dplyr)

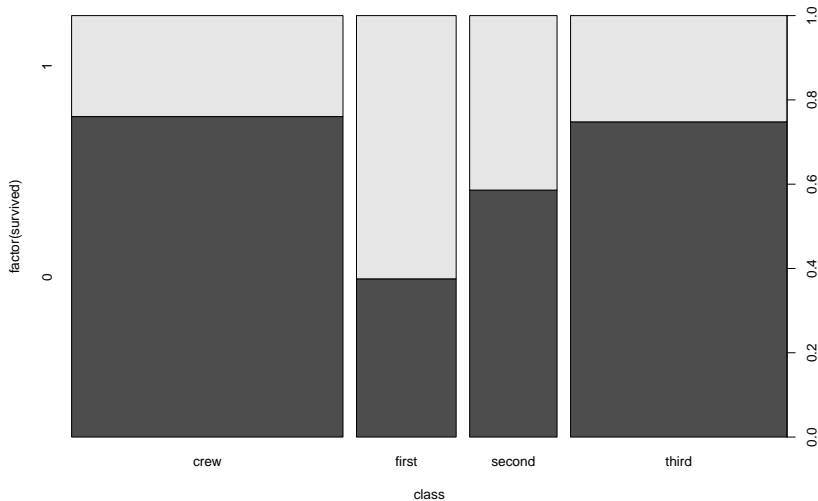Passenger survival according to class

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

```
# A tibble: 8 x 3
# Groups:   class [?]
  class  survived count
  <fct>     <int> <int>
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), count =
n())

Or graphically. . .

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

Fitting GLMs in R: glm

```
tit.glm <- glm(survived ~ class, data = titanic, family = binomial(link = "logit"))
```

```
Call:
glm(formula = survived ~ class, family = binomial(link = "logit"),
    data = titanic)

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 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2769.5  on 2200  degrees of freedom
Residual deviance: 2588.6  on 2197  degrees of freedom
AIC: 2596.6

Number of Fisher Scoring iterations: 4
```

These estimates are in logit scale!

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?

```r
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:

```r
sum(titanic$survived[titanic$class == "first"]) /
  nrow(titanic[titanic$class == "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
```
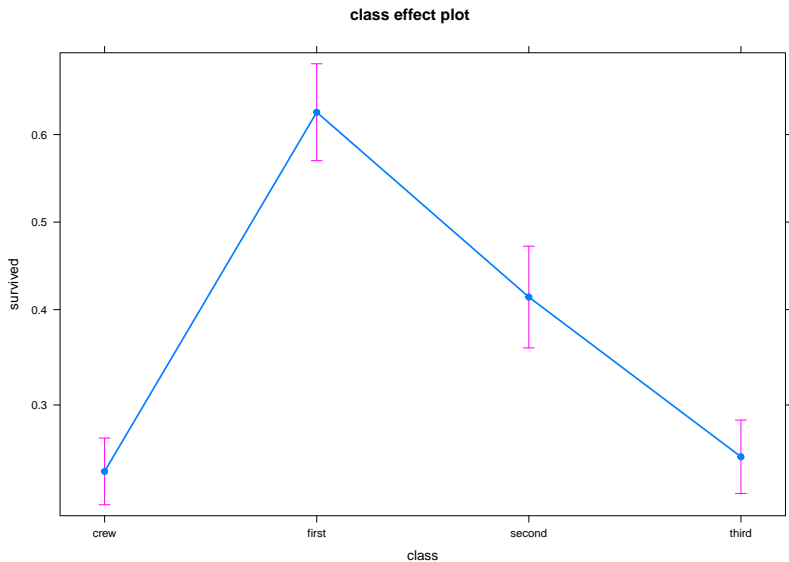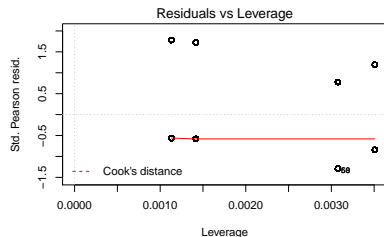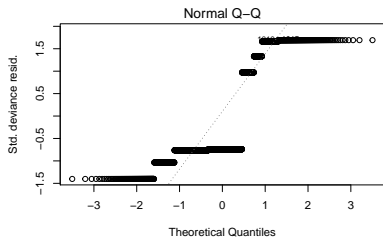
Effects plot

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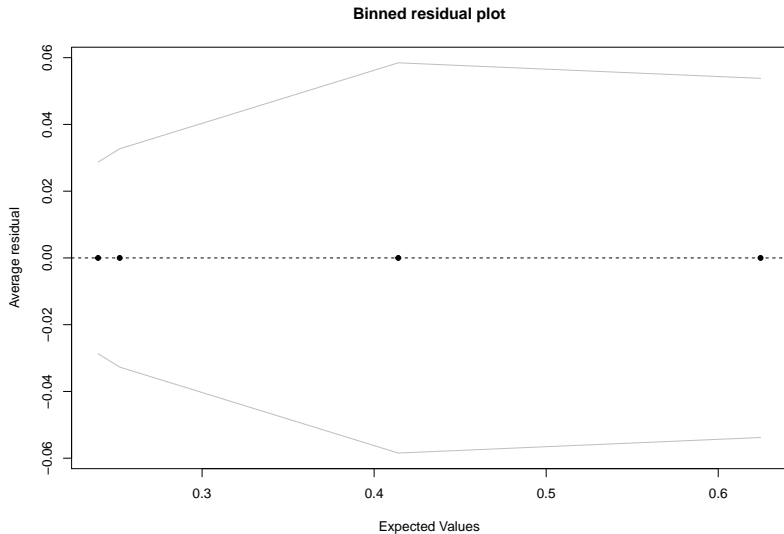

class effect plot

Logistic regression: model checking
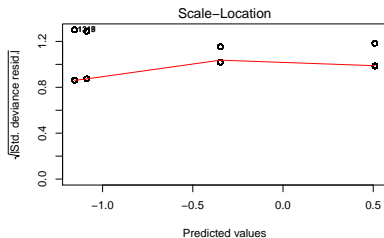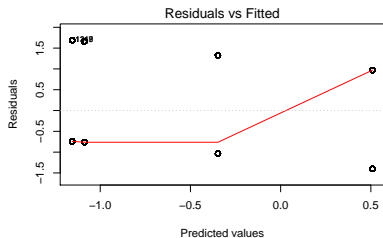
Not very useful.

Binned residual plots for logistic regression
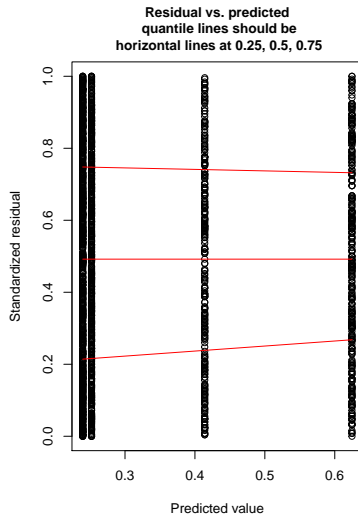
```
predvals <- predict(tit.glm, type="response")
arm::binnedplot(predvals, titanic$survived - predvals)
```


Binned residual plot

Residual diagnostics with DHARMa

```
library(DHARMa)
simulateResiduals(tit.glm, plot = TRUE)
```
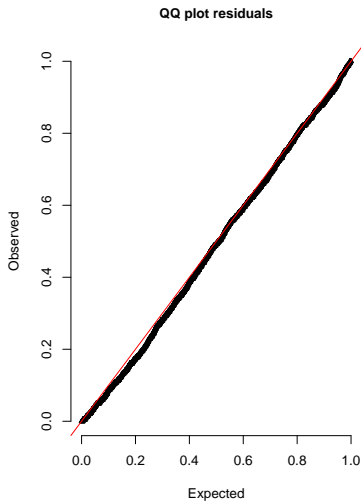
Model checking with simulated data

```
library(bayesplot)
sims <- simulate(tit.glm, nsim = 100)
ppc_bars(titanic$survived, yrep = t(as.matrix(sims)))
```
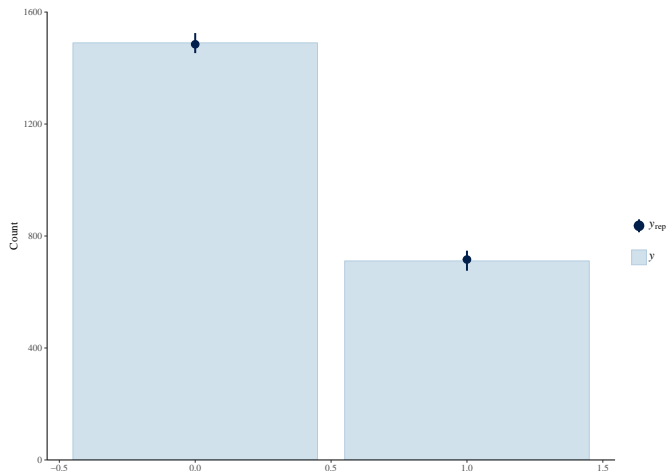
Pseudo R-squared for GLMs

```
library(sjstats)
r2(tit.glm)
```

```
Cox & Snell's R-squared: 0.0789
 Nagelkerke's R-squared: 0.1102
```

But many caveats apply! (e.g. see here and here)

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.
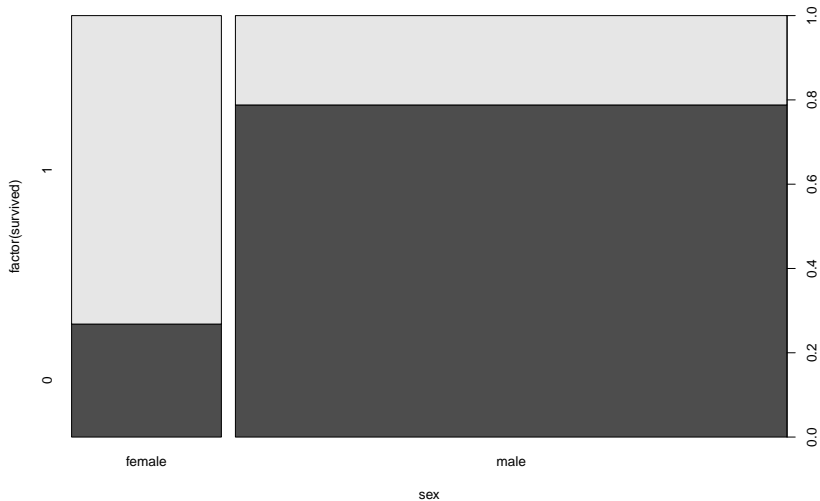7. Plot model: `plot(allEffects(model))`. Or use `visreg`.
8. Examine residuals: use `arm::binnedplot` or `DHARMa::simulateResiduals`.

Q: Did men have higher survival than women?

Plot first

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

Fit model

```
Call:
glm(formula = survived ~ sex, family = binomial(link = "logit"),
    data = titanic)

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 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2769.5  on 2200  degrees of freedom
Residual deviance: 2335.0  on 2199  degrees of freedom
```
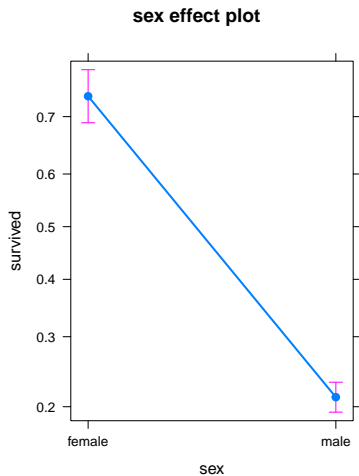
Effects

sex effect plot

```
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), sum)
```

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

Mmmm...

Fit additive model with both factors

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

glm(formula = survived ~ class + sex, family = binomial, data =
            coef.est coef.se
(Intercept)  1.19     0.16
classfirst   0.88     0.16
classsecond -0.07     0.17
classthird  -0.78     0.14
sexmale     -2.42     0.14
---
  n = 2201, k = 5
  residual deviance = 2228.9, null deviance = 2769.5 (difference
```
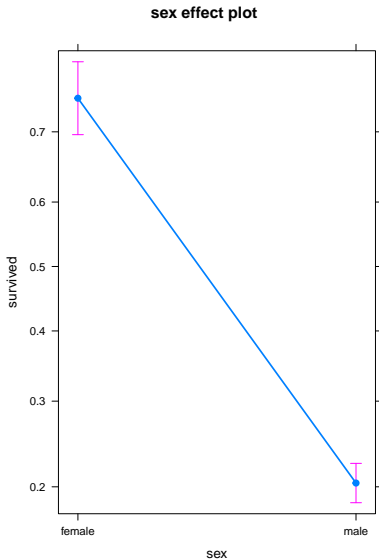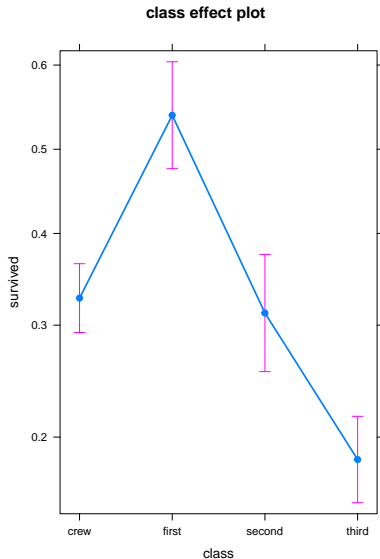
Plot additive model

```
plot(allEffects(tit.sex.class))
```

Fit model with both factors (interactions)

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

glm(formula = survived ~ class * sex, family = binomial, data =
                     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 (difference
```

Effects

class*sex effect plot


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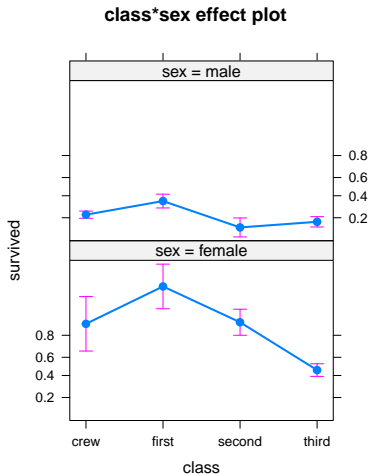
So, women had higher probability of survival than men, even within the same class.

Logistic regression for proportion data

Read Titanic data in different format

Read Titanic_prop.csv data.

```
  X Class    Sex   Age  No Yes
1 1   1st Female Adult   4 140
2 2   1st Female Child   0   1
3 3   1st   Male Adult 118  57
4 4   1st   Male Child   0   5
5 5   2nd Female Adult  13  80
6 6   2nd Female Child   0  13
```

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

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

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

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

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 ' ' 1

(Dispersion parameter for binomial family taken to be 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
Read UN_GDP_infantmortality.csv.

```
          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 (per
```


Infant mortality (per 1000 births)
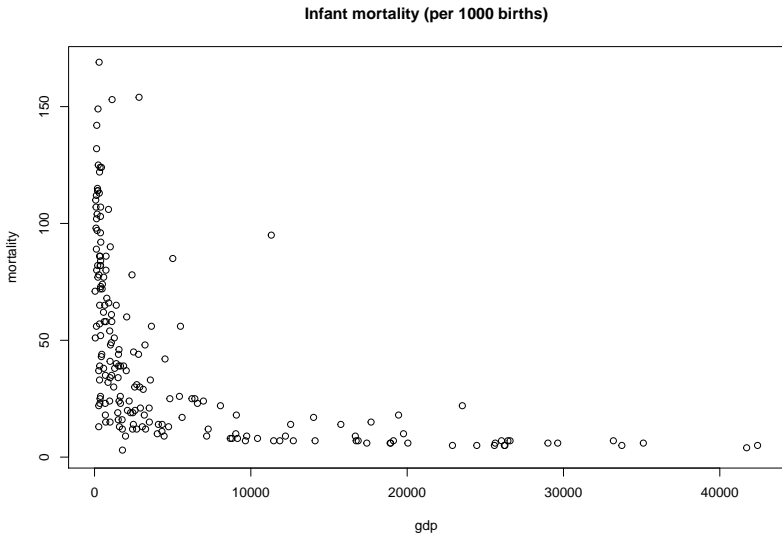
Fit model

```r
gdp.glm <- glm(cbind(mortality, 1000 - mortality) ~ gdp,
               data = gdp, family = binomial(link = "logit"))
```

```
Call:
glm(formula = cbind(mortality, 1000 - mortality) ~ gdp, family =
    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 ' ' 1

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

Effects

```
allEffects(gdp.glm)

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

gdp effect
gdp
          40        10000        20000        30000        40000
0.0652177296 0.0191438829 0.0054028095 0.0015096074 0.0004206154
```
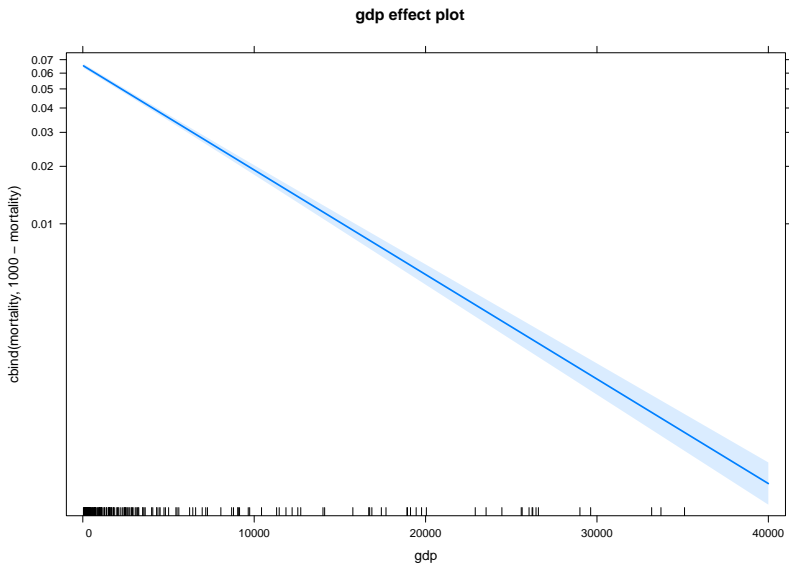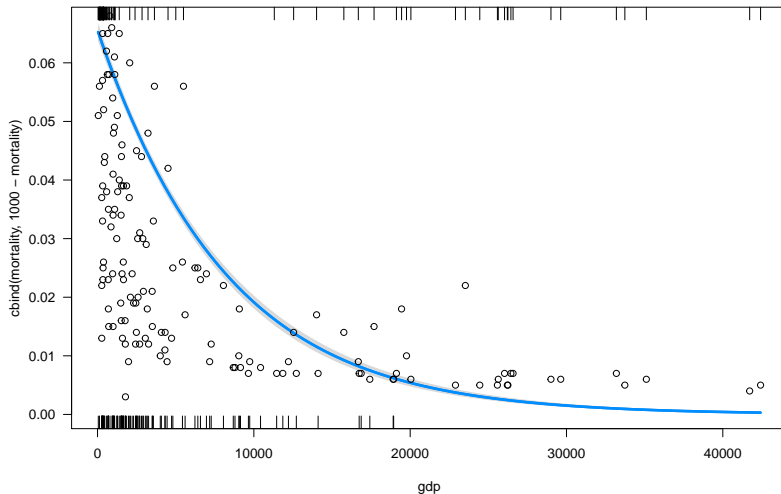
Effects plot

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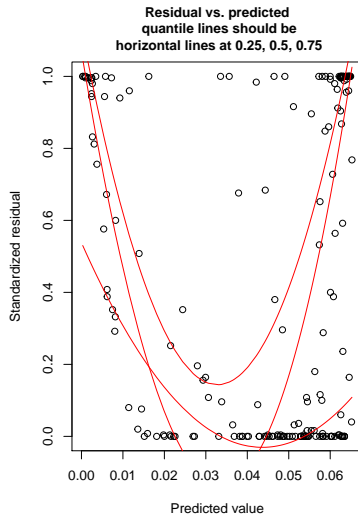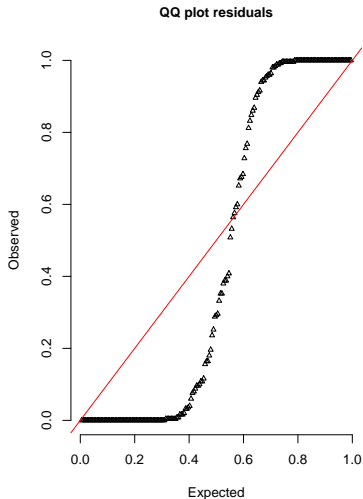
gdp effect plot

Plot model using visreg:

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

Residuals diagnostics with DHARMa

```
simulateResiduals(gdp.glm, plot = TRUE)
```


DHARMa scaled residual plots

Overdispersion

Testing for overdispersion (DHARMa)

```
simres <- simulateResiduals(gdp.glm, refit = TRUE)
testOverdispersion(simres)
```

```
    DHARMa nonparametric overdispersion test via comparison to
    simulation under H0 = fitted model

data:  simres
dispersion = 20.562, p-value < 2.2e-16
alternative hypothesis: overdispersion
```

Overdispersion in logistic regression with proportion data

```
gdp.overdisp <- glm(cbind(mortality, 1000 - mortality) ~ gdp,
                data = gdp, family = quasibinomial)


Call:
glm(formula = cbind(mortality, 1000 - mortality) ~ gdp, family =
    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 ' ' 1

(Dispersion parameter for quasibinomial family taken to be 20.79
```

Mean estimates do not change after accounting for overdispersion

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

gdp effect
gdp
          40         10000        20000        30000        40000
0.0652177296 0.0191438829 0.0054028095 0.0015096074 0.0004206154

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

gdp effect
gdp
          40         10000        20000        30000        40000
0.0652177296 0.0191438829 0.0054028095 0.0015096074 0.0004206154
```
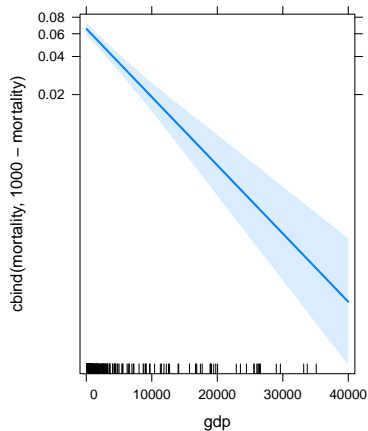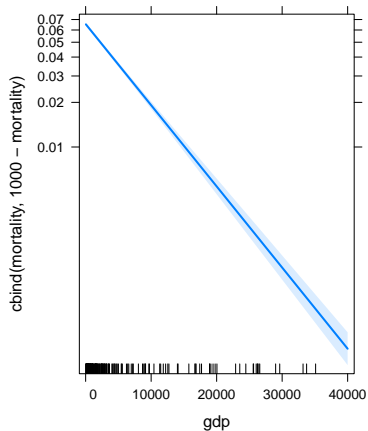
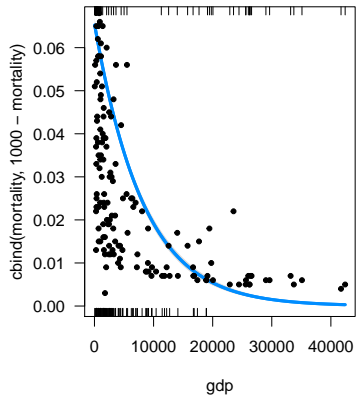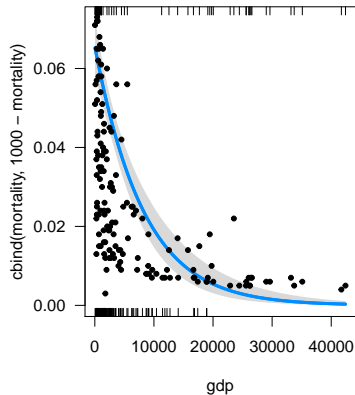# But standard errors (uncertainty) do!
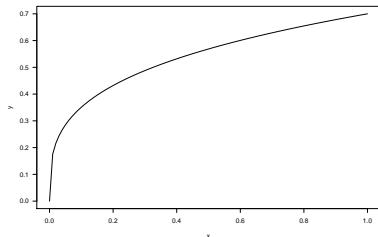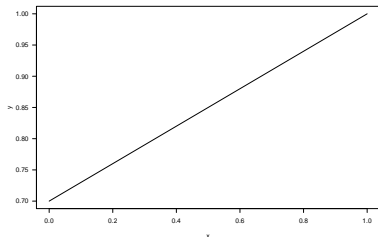
Plot model and data

Overdispersion

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

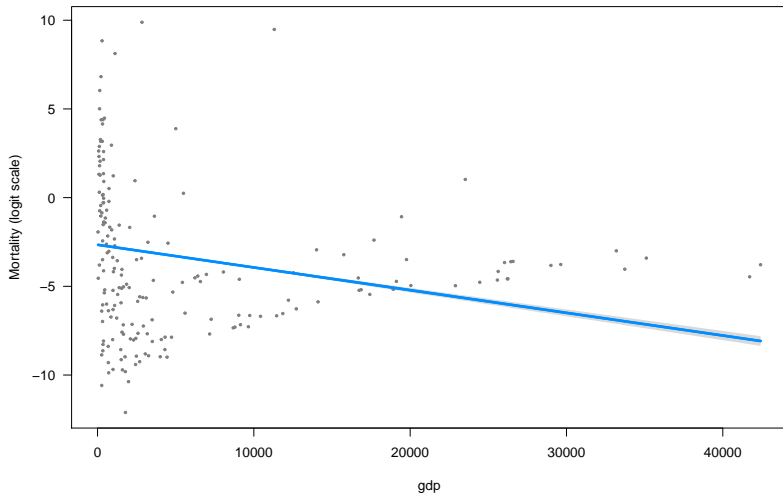# Think about the shape of relationships

y ~ x + z
Really? Not everything has to be linear! Actually, it often is not.
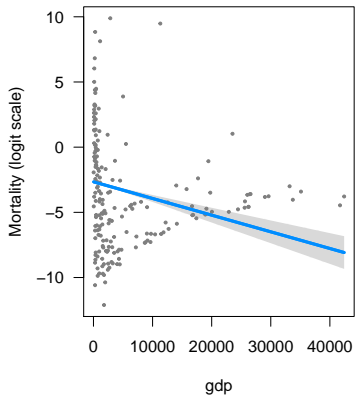**Think** about shape of relationship. See chapter 3 in Bolker's book.

Think about the shape of relationships

```
visreg(gdp.glm, ylab = "Mortality (logit scale)")
```

Think about the shape of relationships
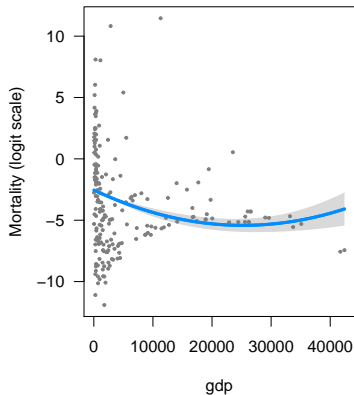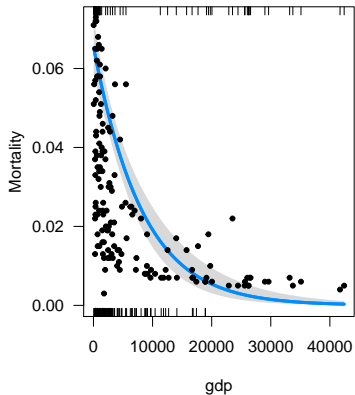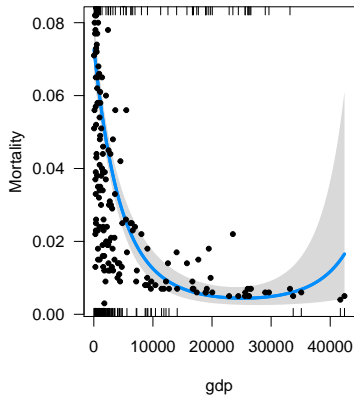
Think about the shape of relationships
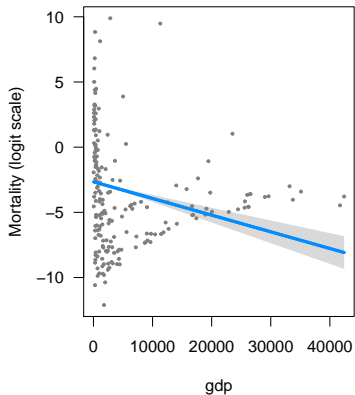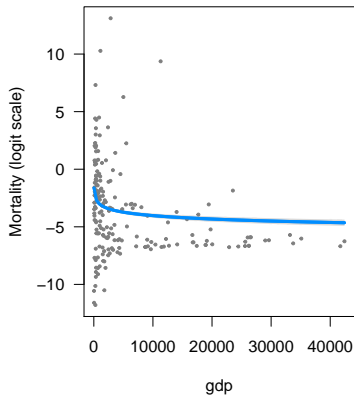
Think about the shape of relationships

Think about the shape of relationships
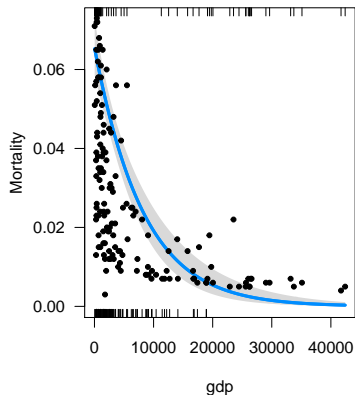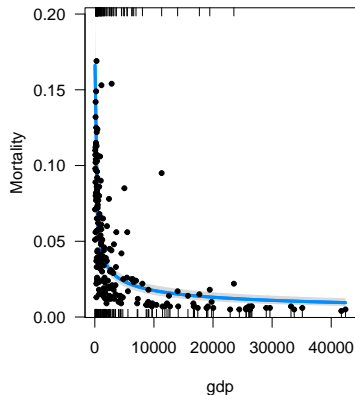
GLM for count data: Poisson regression

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 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            area
 Min.   : 2.571   Min.   :0.25
 1st Qu.:26.879   1st Qu.:0.25
 Median :47.493   Median :0.50
 Mean   :47.959   Mean   :0.62
 3rd Qu.:67.522   3rd Qu.:1.00
 Max.   :99.135   Max.   :1.00
```

EDA

Histogram of seedl$count

```
table(seedl$count)


 0  1  2  3  4  5  7
 7 12 13  8  7  2  1
```
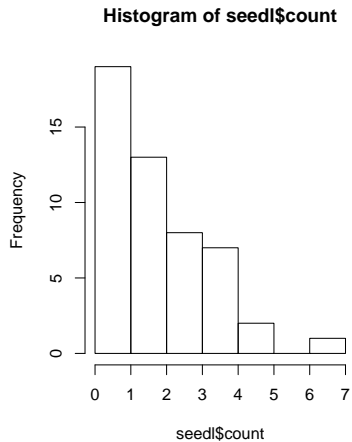
Q: Relationship between Nseedlings and light?

```
plot(seedl$light, seedl$count, las = 1, xlab = "Light (GSF)", yl
```
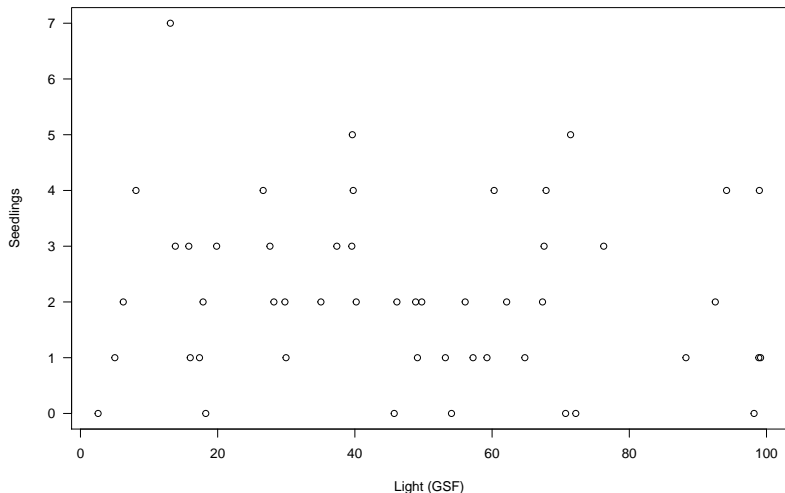
Let's fit model (Poisson regression)

```
seedl.glm <- glm(count ~ light, data = seedl, family = poisson(link="log"))
summary(seedl.glm)


Call:
glm(formula = count ~ light, family = poisson(link = "log"),
    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 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 63.029  on 49  degrees of freedom
Residual deviance: 62.492  on 48  degrees of freedom
AIC: 182.03

Number of Fisher Scoring iterations: 5
```

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))
```


light effect plot
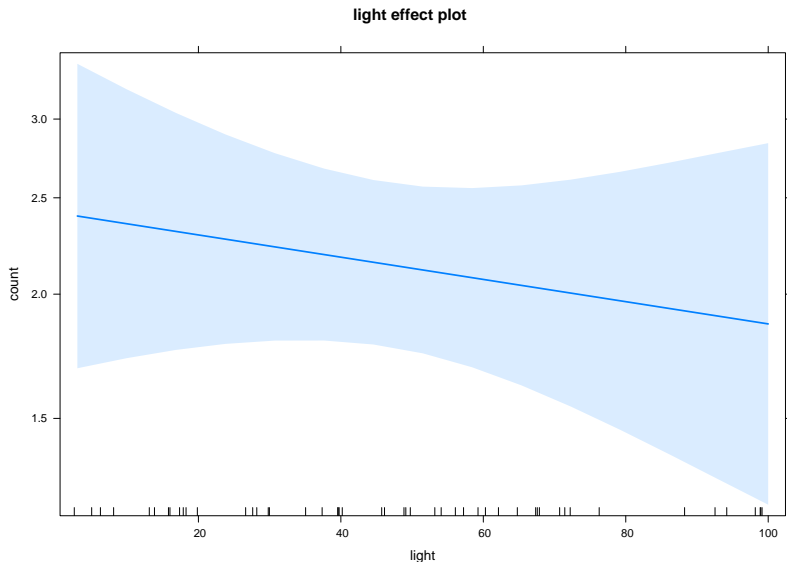
Using visreg

```
visreg(seedl.glm, scale = "response", ylim = c(0, 7))
points(count ~ light, data = seedl, pch = 20)
```
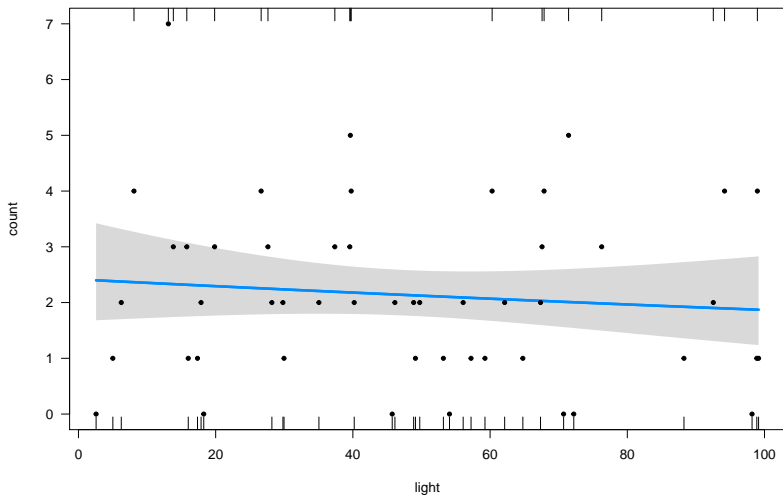
Poisson regression: model checking
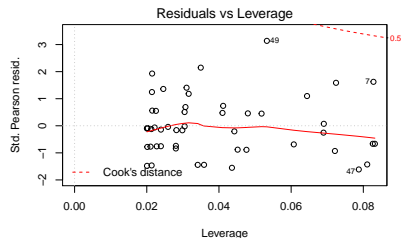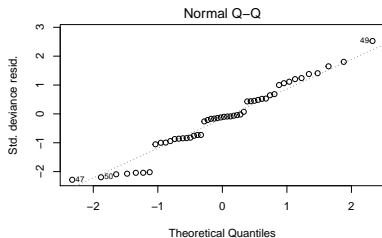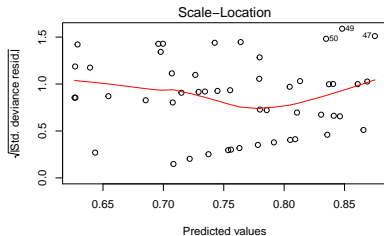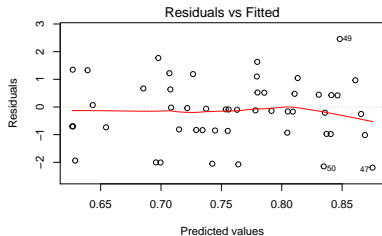
Is there pattern of residuals along predictor?

```
plot(seedl$light, resid(seedl.glm))
```
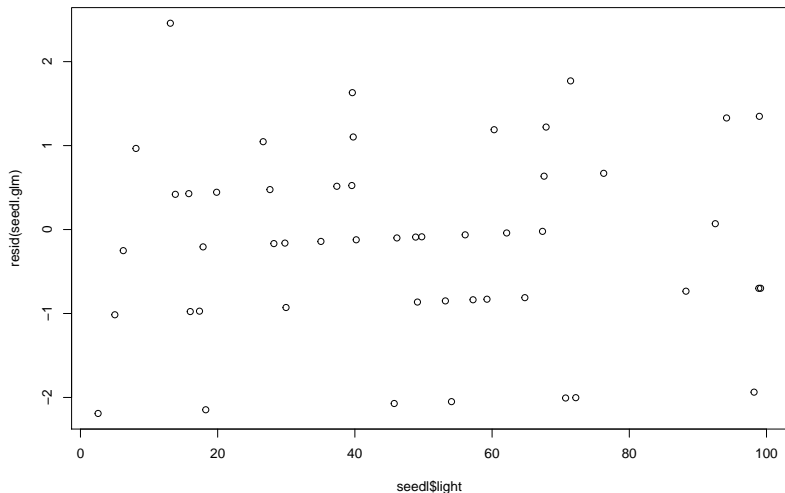
Residuals diagnostics with DHARMa

```
simulateResiduals(seedl.glm, plot = TRUE)
```


DHARMa scaled residual plots
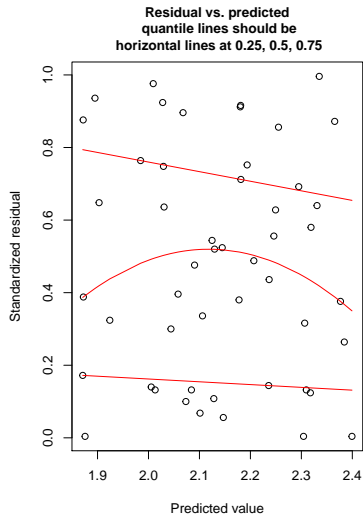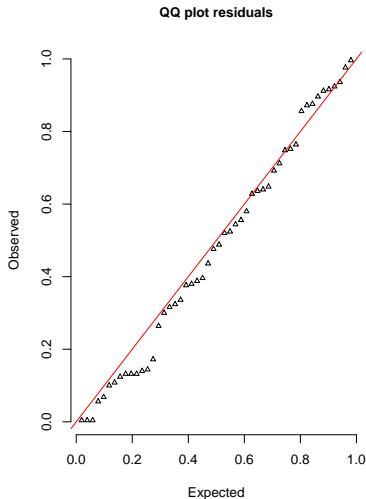
Poisson regression: Overdispersion

Always check overdispersion with count data

```
simres <- simulateResiduals(seedl.glm, refit = TRUE)
testOverdispersion(simres)
```

```
    DHARMa nonparametric overdispersion test via comparison to
    simulation under H0 = fitted model

data:  simres
dispersion = 1.1327, p-value = 0.268
alternative hypothesis: overdispersion
```

Accounting for overdispersion in count data

Use family quasipoisson

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

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 ' ' 1

(Dispersion parameter for quasipoisson family taken to be 1.1349

    Null deviance: 63.029  on 49  degrees of freedom
Residual deviance: 62.492  on 48  degrees of freedom
```

Mean estimates do not change after accounting for overdispersion

```
 model: count ~ light

 light effect
light
       3        30        50        70       100
2.396665 2.235657 2.123408 2.016794 1.866826

 model: count ~ light

 light effect
light
       3        30        50        70       100
2.396665 2.235657 2.123408 2.016794 1.866826
```

But standard errors may change
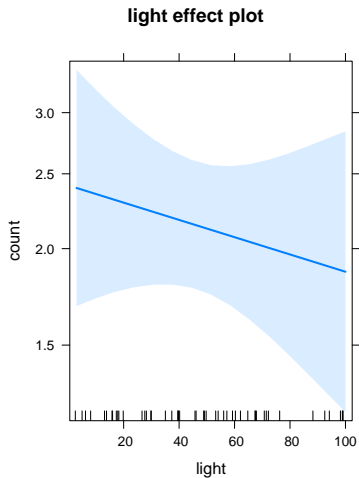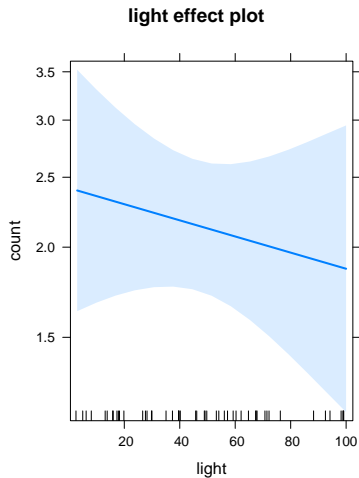
What if survey plots have different area?

Avoid regression of ratios

seedlings/area ~ light

Spurious Correlation and the Fallacy of the Ratio Standard Revisited

By RICHARD A. KRONMAL†

Figure 6:

Use offset to standardise response variables in GLMs

```
seedl.offset <- glm(count ~ light, offset = seedl$area, data = s
summary(seedl.offset)


Call:
glm(formula = count ~ light, family = poisson, data = seedl,
    offset = seedl$area)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6926  -0.8532   0.1491   0.5211   3.1051

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.299469   0.185468   1.615    0.106
light       -0.004498   0.003441  -1.307    0.191

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 70.263  on 49  degrees of freedom
```

Note estimates now referred to area units

```
exp(coef(seedl.offset))

(Intercept)       light
  1.3491422    0.9955123
```

Mixed / Multilevel models

Example dataset: trees

- Data on 1000 trees from 10 plots.

```
head(trees)
```

```
  plot   dbh height    sex dead   dbh.c
1    2 38.85   37.8 female    0   13.85
2    4 26.05   38.1 female    0    1.05
3    5 42.66   50.2 female    0   17.66
4    2 20.72   30.1 female    0   -4.28
5    4 21.83   34.0 female    0   -3.17
6    4  8.23   21.9   male    0  -16.77
```

Example dataset: trees

- Data on 1000 trees from 10 plots.
- Trees per plot: 4 - 392.

```
head(trees)
```

```
  plot   dbh height    sex dead  dbh.c
1    2 38.85   37.8 female    0  13.85
2    4 26.05   38.1 female    0   1.05
3    5 42.66   50.2 female    0  17.66
4    2 20.72   30.1 female    0  -4.28
5    4 21.83   34.0 female    0  -3.17
6    4  8.23   21.9   male    0 -16.77
```

Q: What's the relationship between tree diameter and height?

A simple linear model

```
lm.simple <- lm(height ~ dbh, data = trees)
```

```
Call:
lm(formula = height ~ dbh, data = trees)

Residuals:
    Min      1Q   Median      3Q      Max
-13.7384  -4.7652   0.4759   4.2931  13.5282

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 13.18767    0.41476   31.80   <2e-16 ***
dbh          0.60967    0.01351   45.14   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.549 on 998 degrees of freedom
Multiple R-squared: 0.6712,    Adjusted R-squared: 0.6709
F-statistic:  2038 on 1 and 998 DF,  p-value: < 2.2e-16
```

Remember our model structure

$$y_i \sim N(\mu_i, \sigma^2)$$
$$\mu_i = \alpha + \beta x_i$$
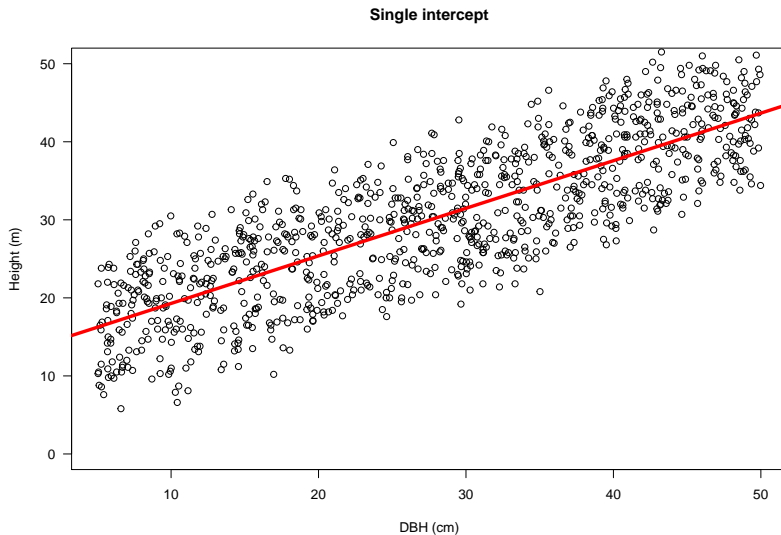
In this case:

$$Height_i \sim N(\mu_i, \sigma^2)$$
$$\mu_i = \alpha + \beta DBH_i$$

α: expected height when DBH $= 0$
β: how much height increases with every unit increase of DBH

There is only one intercept

Single intercept

What if allometry varies among plots?

Different intercept for each plot
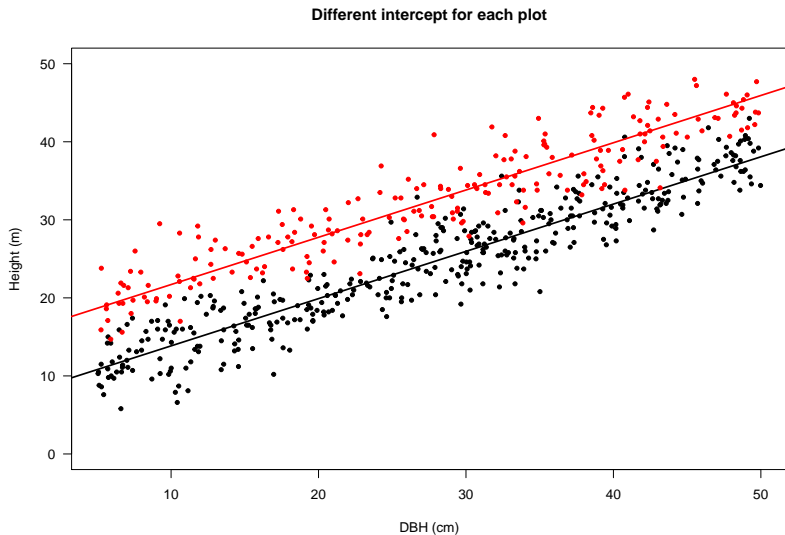
Fitting a varying intercepts model with `lm`

```
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
```

Single vs varying intercept
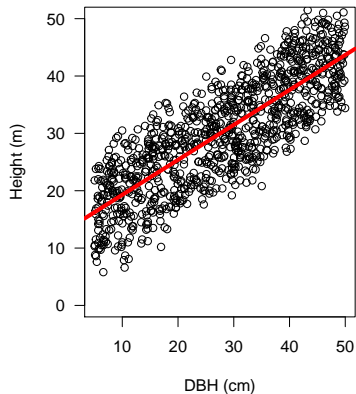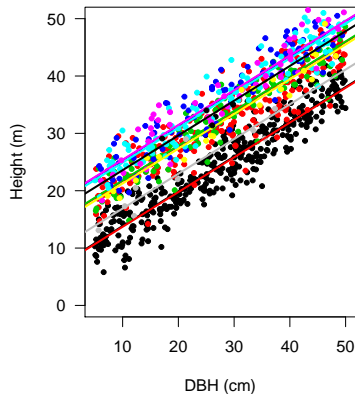
Mixed models enable us to account for variability

▶ Varying intercepts
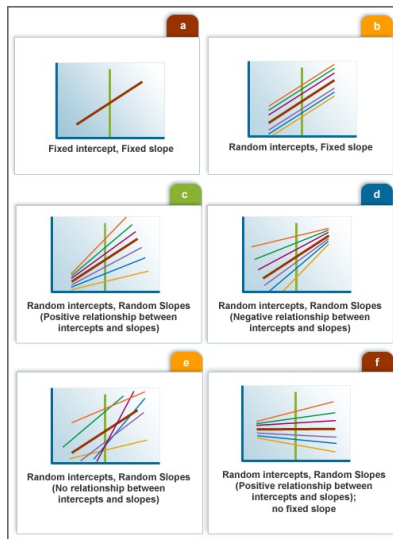
Figure 7:

Mixed models enable us to account for variability

▶ Varying intercepts
▶ Varying slopes
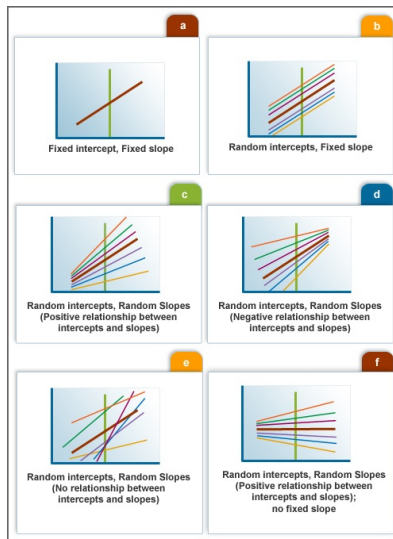
Figure 7:

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:

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

Mixed models estimate varying parameters (intercepts and/or slopes) with pooling among levels (rather than considering them fully independent)

- **complete pooling**: Single overall intercept.

Hence there's gradient between

- **complete pooling**: Single overall intercept.
  - `lm (height ~ dbh)`

Hence there's gradient between

- **complete pooling**: Single overall intercept.
  - `lm (height ~ dbh)`
- **no pooling**: One *independent* intercept for each plot.

Hence there's gradient between

- **complete pooling**: Single overall intercept.
  - `lm (height ~ dbh)`
- **no pooling**: One *independent* intercept for each plot.
  - `lm (height ~ dbh + factor(plot))`

Hence there's gradient between

- **complete pooling**: Single overall intercept.
  - `lm (height ~ dbh)`
- **no pooling**: One *independent* intercept for each plot.
  - `lm (height ~ dbh + factor(plot))`
- **partial pooling**: Inter-related intercepts.

Hence there's gradient between

- **complete pooling**: Single overall intercept.
  - `lm (height ~ dbh)`
- **no pooling**: One *independent* intercept for each plot.
  - `lm (height ~ dbh + factor(plot))`
- **partial pooling**: Inter-related intercepts.
  - `lmer(height ~ dbh + (1 | plot))`

Random vs Fixed effects?

1. Fixed effects constant across individuals, random effects vary.

http://andrewgelman.com/2005/01/25/why_i_dont_use/

Random vs Fixed effects?

1. Fixed effects constant across individuals, random effects vary.
2. Effects are fixed if they are interesting in themselves; random if interest in the underlying population.

http://andrewgelman.com/2005/01/25/why_i_dont_use/

Random vs Fixed effects?

1. Fixed effects constant across individuals, random effects vary.
2. Effects are fixed if they are interesting in themselves; random if interest in the underlying population.
3. Fixed when sample exhausts the population; random when the sample is small part of the population.

http://andrewgelman.com/2005/01/25/why_i_dont_use/

Random vs Fixed effects?

1. Fixed effects constant across individuals, random effects vary.
2. Effects are fixed if they are interesting in themselves; random if interest in the underlying population.
3. Fixed when sample exhausts the population; random when the sample is small part of the population.
4. Random effect if it's assumed to be a realized value of random variable.

http://andrewgelman.com/2005/01/25/why_i_dont_use/

Random vs Fixed effects?

1. Fixed effects constant across individuals, random effects vary.
2. Effects are fixed if they are interesting in themselves; random if interest in the underlying population.
3. Fixed when sample exhausts the population; random when the sample is small part of the population.
4. Random effect if it's assumed to be a realized value of random variable.
5. Fixed effects estimated using least squares or maximum likelihood; random effects estimated with shrinkage.

http://andrewgelman.com/2005/01/25/why_i_dont_use/

What is a random effect, really?

1. Varies by group

Random effects are estimated with partial pooling, while fixed effects are not (infinite variance).

What is a random effect, really?

1. Varies by group
2. Variation estimated with probability model

Random effects are estimated with partial pooling, while fixed effects are not (infinite variance).

Shrinkage improves parameter estimation
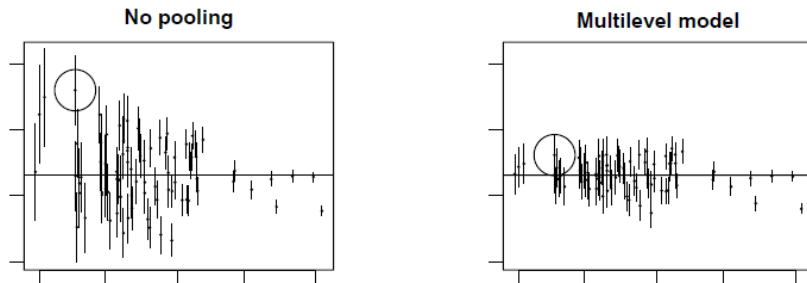
Especially for groups with low sample size

Figure 8:

From Gelman & Hill p. 253

Fitting mixed/multilevel models

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

lmer(formula = height ~ dbh + (1 | plot), data = trees)
            coef.est coef.se
(Intercept) 14.80    1.44
dbh          0.61    0.01

Error terms:
 Groups   Name        Std.Dev.
 plot     (Intercept) 4.45
 Residual             2.89
---
number of obs: 1000, groups: plot, 10
AIC = 5015.6, DIC = 4996.4
deviance = 5002.0
```

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"
```

Broom: model estimates in tidy form

```
library(broom)
tidy(mixed)
```

```
                      term    estimate   std.error statistic    group
1              (Intercept) 14.7981641 1.437421280  10.29494    fixed
2                      dbh  0.6056549 0.007040079  86.02956    fixed
3      sd_(Intercept).plot  4.4535702          NA        NA     plot
4 sd_Observation.Residual  2.8852942          NA        NA Residual
```
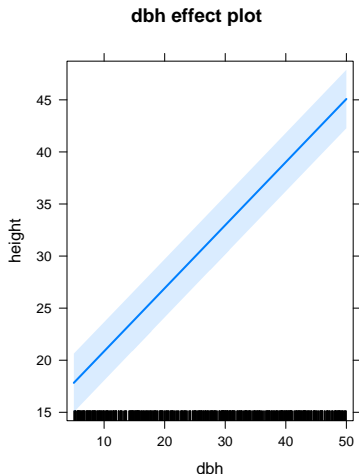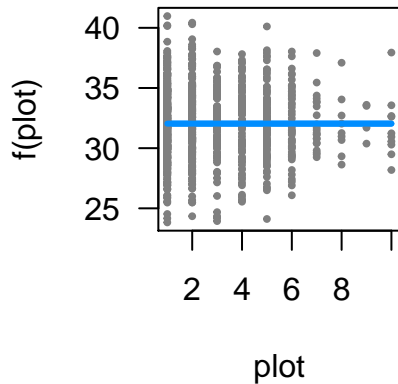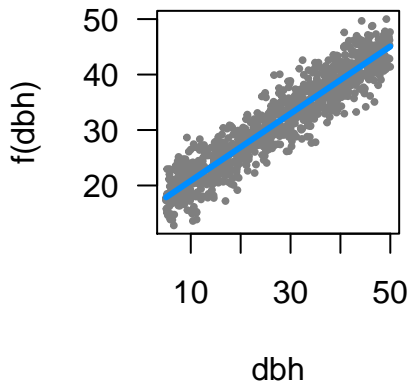
Visualising model: `allEffects`

dbh effect plot

```
model: height ~ dbh

dbh effect
dbh
       5       20       30       40       50
17.82644 26.91126 32.96781 39.02436 45.08091
```
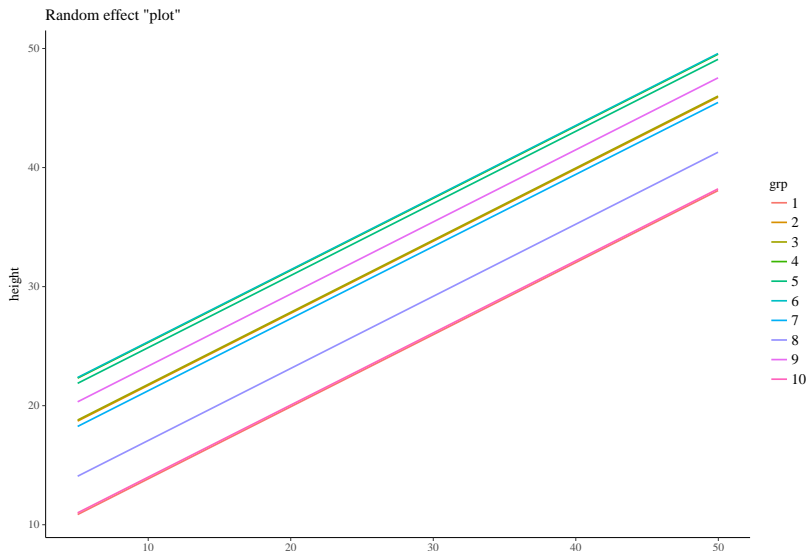
Visualising model: `visreg`

Visualising model: `sjPlot`

```
library(sjPlot)
sjp.lmer(mixed, type = "ri.slope")
```


Random effect "plot"

Visualising model: `sjPlot`
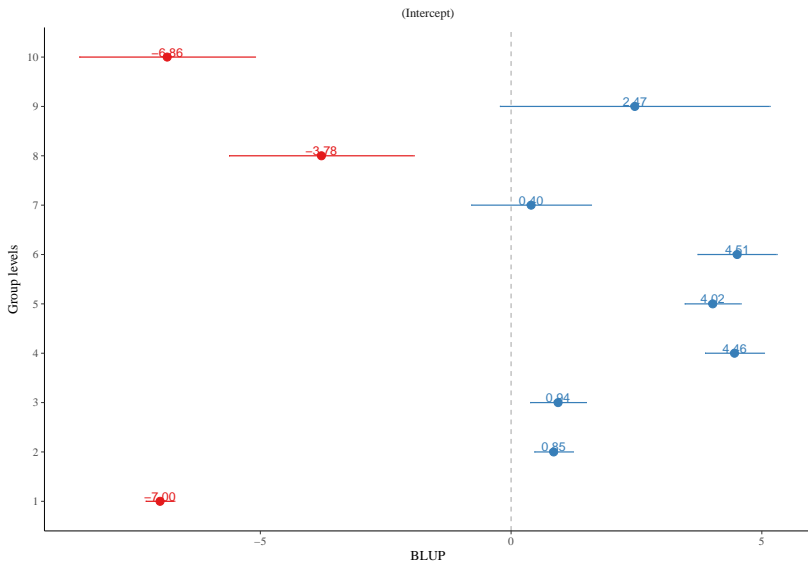
`sjp.lmer`(mixed)

Visualising model: `sjPlot`

```
sjp.lmer(mixed, type = "eff", show.ci = TRUE)
```

Marginal effects of model predictors
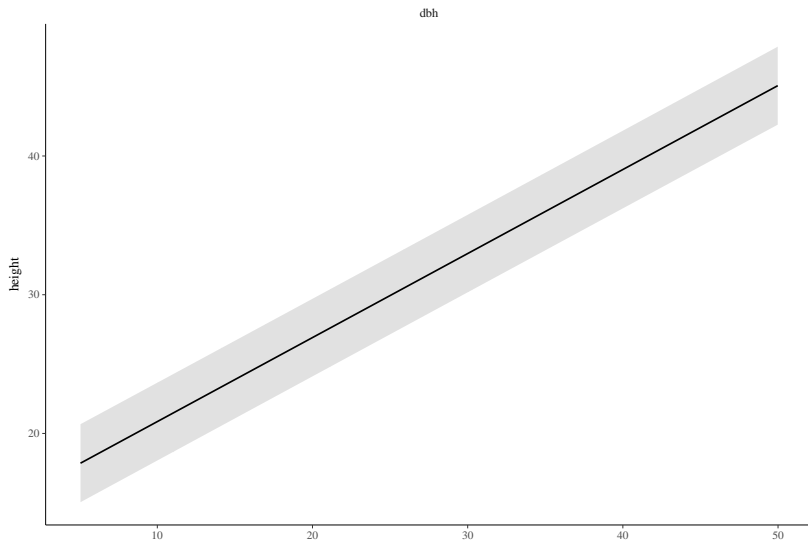
Using merTools to understand fitted model

```r
library(merTools)
shinyMer(mixed)
```

Plotting regression for individual forest plots

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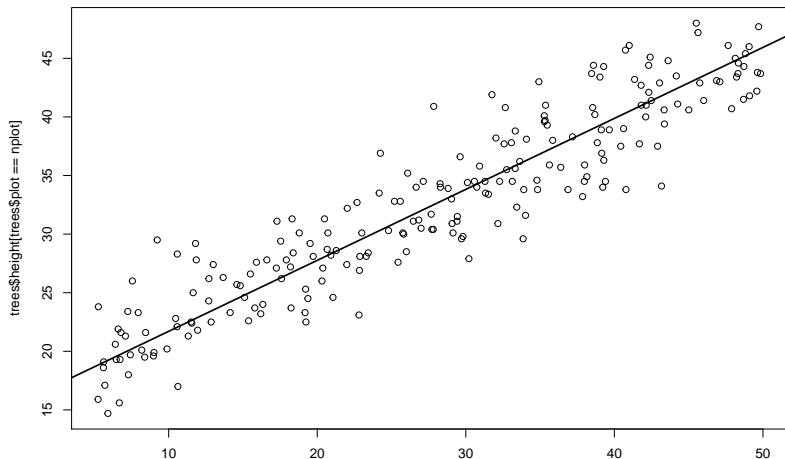
Checking residuals

`plot(mixed)`
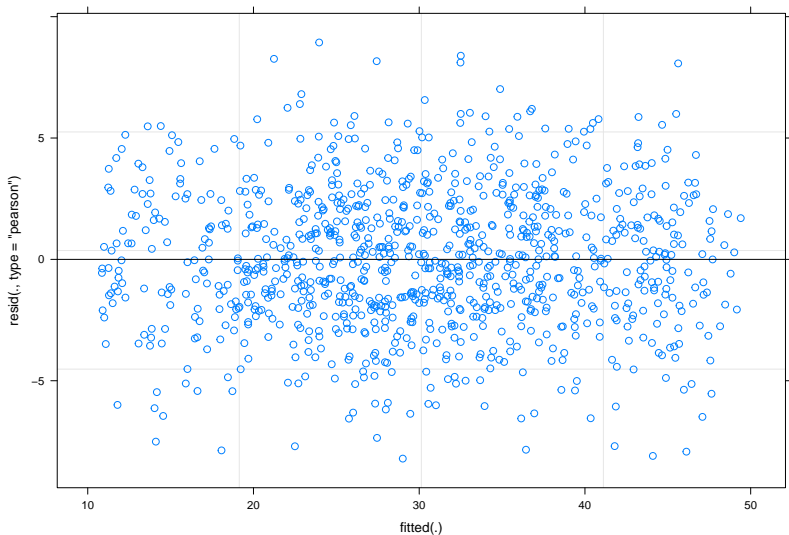
Checking residuals (DHARMa)

```
simulateResiduals(mixed, plot = TRUE, use.u = TRUE)
```


DHARMa scaled residual plots
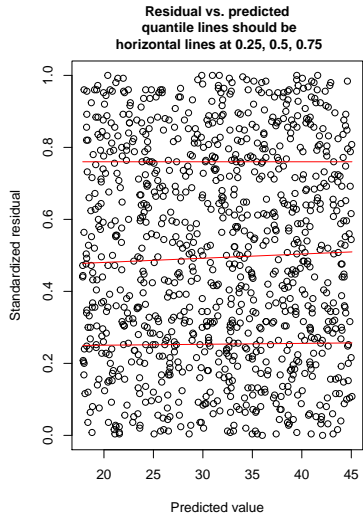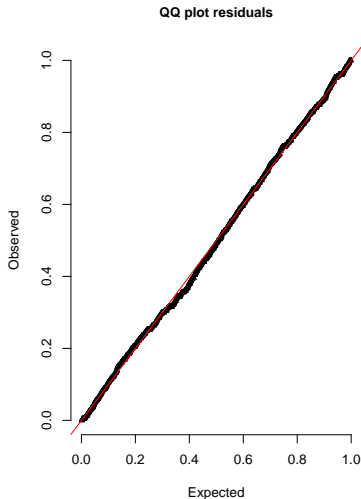
Model checking with simulated data

```
library(bayesplot)
sims <- simulate(mixed, nsim = 100)
ppc_dens_overlay(trees$height, yrep = t(as.matrix(sims)))
```
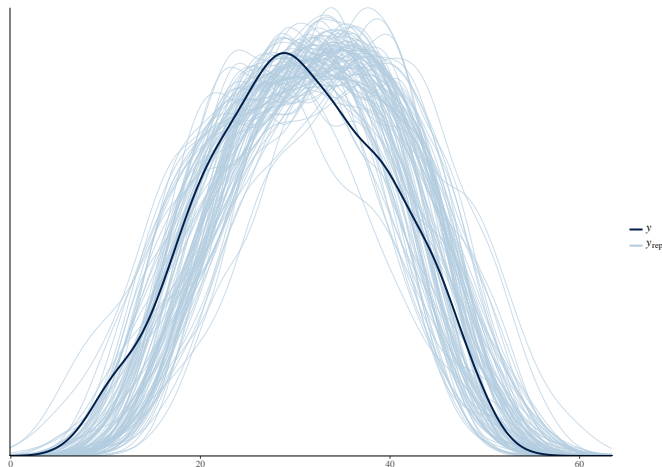
R-squared for GLMMs

Many approaches! Somewhat polemic.
Nakagawa & Schielzeth propose **marginal** (considering fixed effects only)
and **conditional** R^2 (including random effects too):

```
library(MuMIn)
r.squaredGLMM(mixed)


     R2m       R2c
0.6875651 0.9076325
```

Growing the hierarchy: adding plot-level predictors

Model with group-level predictors

We had:

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

Now

$$y_i = a_j + bx_i + \varepsilon_i$$
$$a_j \sim N(\mu_j, \tau^2)$$
$$\mu_j = \gamma + \delta \cdot predictor_j$$
$$\varepsilon_i \sim N(0, \sigma^2)$$

Merging trees and plot data

```
plotdata <- read.csv("data-raw/plotdata.csv")
trees.full <- merge(trees, plotdata, by = "plot")
head(trees.full)
```

```
  plot   dbh height    sex dead   dbh.c temp
1    1 28.63   22.1 female    0    3.63 15.1
2    1 44.71   39.0 female    0   19.71 15.1
3    1 28.31   29.0 female    0    3.31 15.1
4    1 19.33   19.1   male    0   -5.67 15.1
5    1  9.25   12.2 female    0  -15.75 15.1
6    1 30.02   23.1 female    0    5.02 15.1
```

Centre continuous variables

Plot temperatures referred as deviations from 15ºC

```
trees.full$temp.c <- trees.full$temp - 15
```

Fit multilevel model

```
group.pred <- lmer(height ~ dbh + (1 | plot) + temp.c, data = tr
arm::display(group.pred)

lmer(formula = height ~ dbh + (1 | plot) + temp.c, data = trees.
            coef.est coef.se
(Intercept) 11.79    1.75
dbh          0.61    0.01
temp.c       1.07    0.46

Error terms:
 Groups   Name        Std.Dev.
 plot     (Intercept) 3.61
 Residual             2.89
---
number of obs: 1000, groups: plot, 10
AIC = 5012.8, DIC = 4991
deviance = 4996.9
```

Examine model with merTools

```
shinyMer(group.pred)
```

Comparing plot effects with and without group predictor

Estimated plot effects
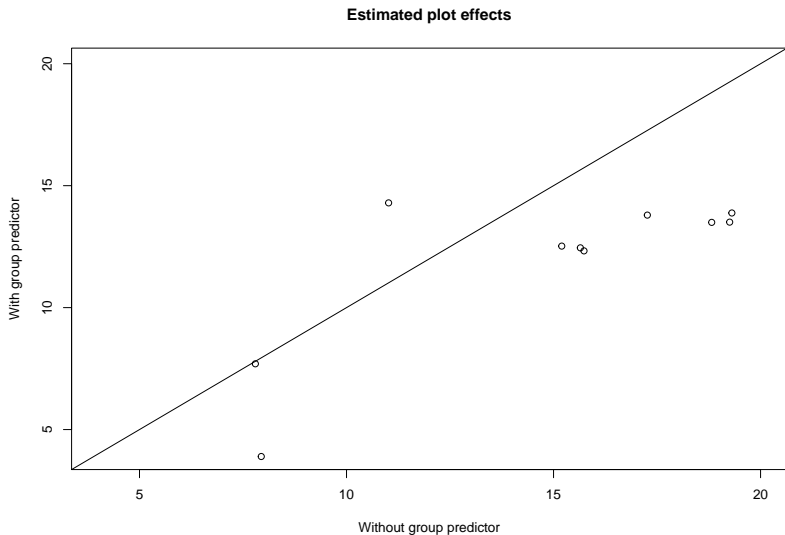
Are plot effects related to temperature?

```
plot(plotdata$temp, coef(group.pred)$plot[,1],
     xlab = "Temperature", ylab = "Plot effect")
```
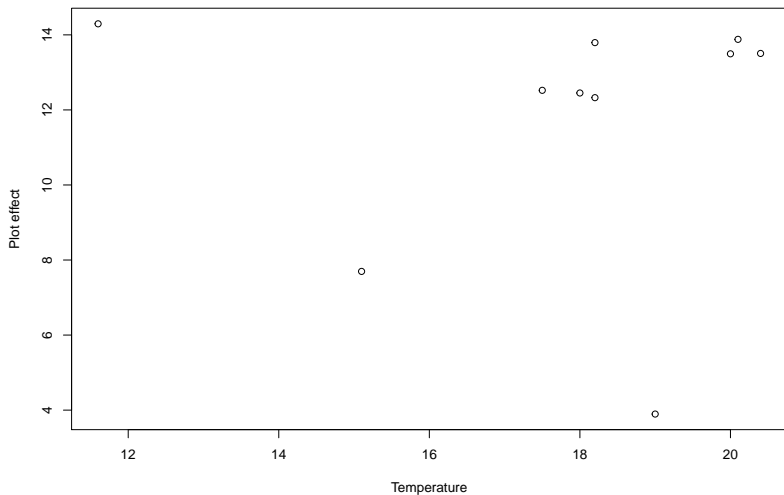
Varying intercepts and slopes

Varying intercepts and slopes

- There is overall difference in height among plots (different intercepts)

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

Varying intercepts and slopes

- There is overall difference in height among plots (different intercepts)
- AND

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

Varying intercepts and slopes

- There is overall difference in height among plots (different intercepts)
- AND
- Relationship between DBH and Height varies among plots (different slopes)

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

Varying intercepts and slopes

```
lmer(formula = height ~ dbh + (1 + dbh | plot), data = trees)
            coef.est coef.se
(Intercept) 14.82    1.48
dbh          0.60    0.01

Error terms:
 Groups    Name        Std.Dev. Corr
 plot      (Intercept) 4.57
           dbh         0.01     -0.41
 Residual              2.88
---
number of obs: 1000, groups: plot, 10
AIC = 5018.6, DIC = 4995.9
deviance = 5001.3
```

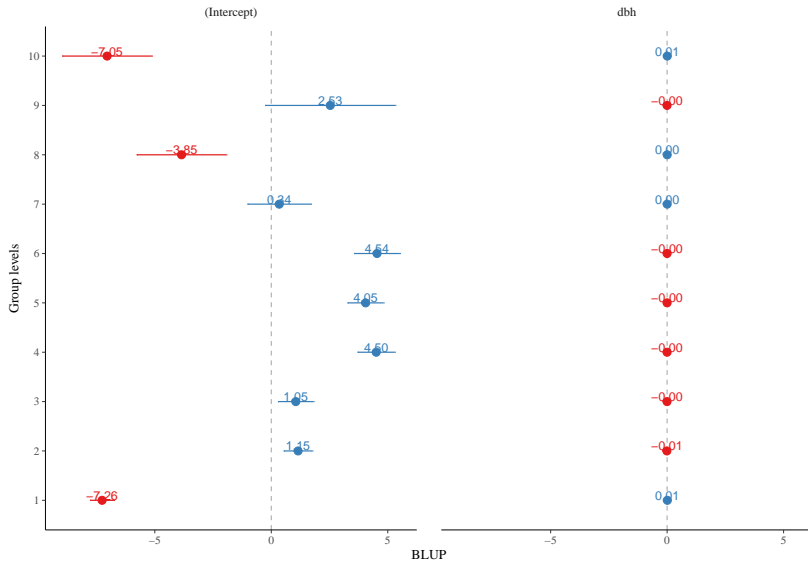## Varying intercepts and slopes

```
$plot
   (Intercept)       dbh
1     7.554578 0.6144452
2    15.966915 0.5942836
3    15.868969 0.6008673
4    19.321161 0.6031855
5    18.866370 0.6039353
6    19.355009 0.6038332
7    15.159258 0.6067449
8    10.965429 0.6080747
9    17.348840 0.6024600
10    7.769135 0.6109349

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

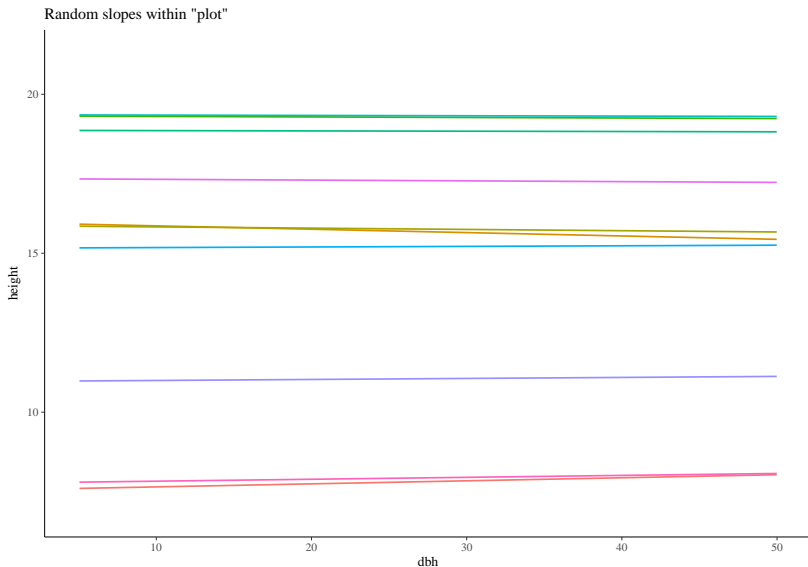# Visualising model: `sjPlot`

`sjp.lmer`(mixed.slopes)

Visualising model: `sjPlot`

```r
sjp.lmer(mixed.slopes, type = "rs.ri")
```


Random slopes within "plot"

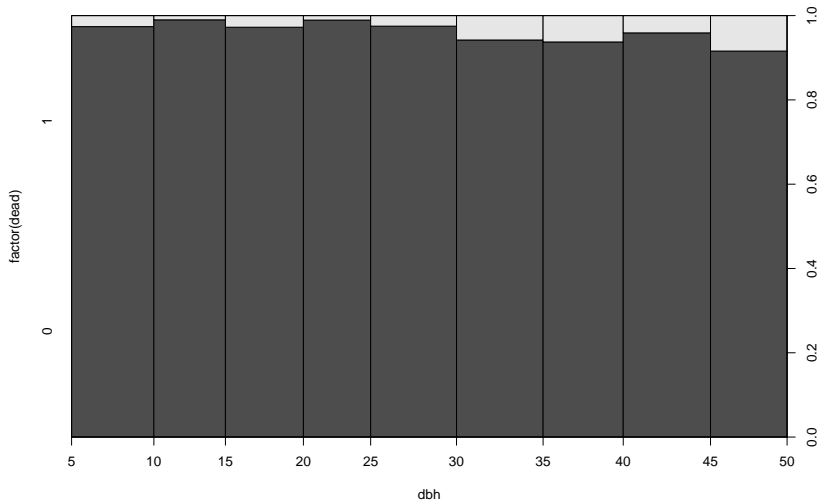# Multilevel logistic regression

Q: Relationship between tree size and mortality

```
plot(dead ~ dbh, data = trees)
```
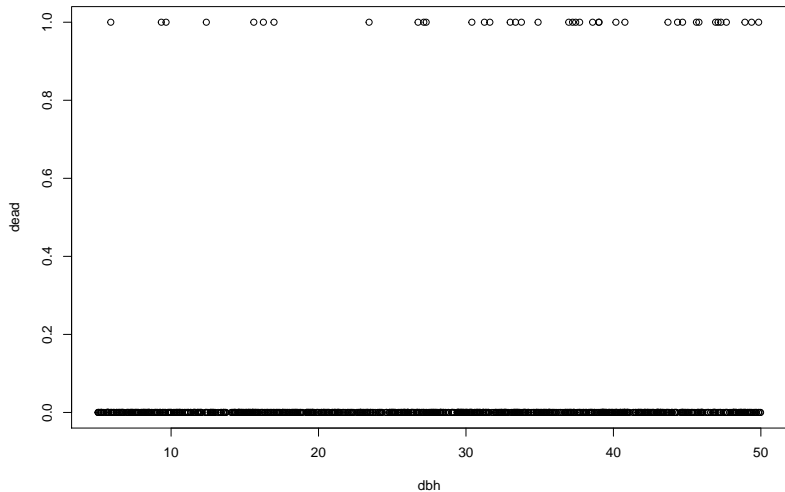
Q: Relationship between tree size and mortality

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

Fit simple logistic regression

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

```
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 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 329.51  on 999  degrees of freedom
```

Logistic regression with *independent* plot effects

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

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

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
factor(plot)6   -0.08426    0.76908  -0.110   0.9128
factor(plot)7    0.03126    1.06064   0.029   0.9765
```

Fit multilevel logistic regression

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

glmer(formula = dead ~ dbh + (1 | plot), data = trees, family =
            coef.est coef.se
(Intercept) -4.47     0.49
dbh          0.04     0.01

Error terms:
 Groups    Name        Std.Dev.
 plot      (Intercept) 0.00
 Residual              1.00
---
number of obs: 1000, groups: plot, 10
AIC = 325.9, DIC = 319.9
deviance = 319.9
```

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"
```
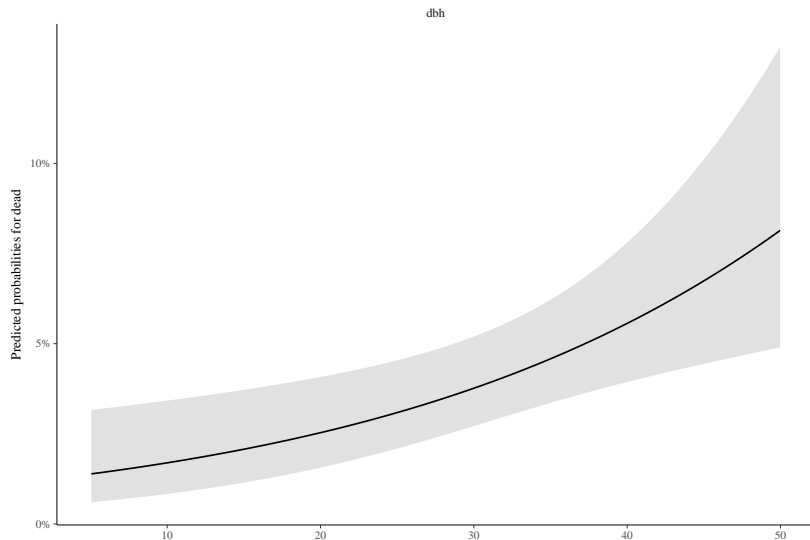
Visualising model: `sjPlot`

```
sjp.glmer(mixed.logis, type = "eff", show.ci = TRUE)
```
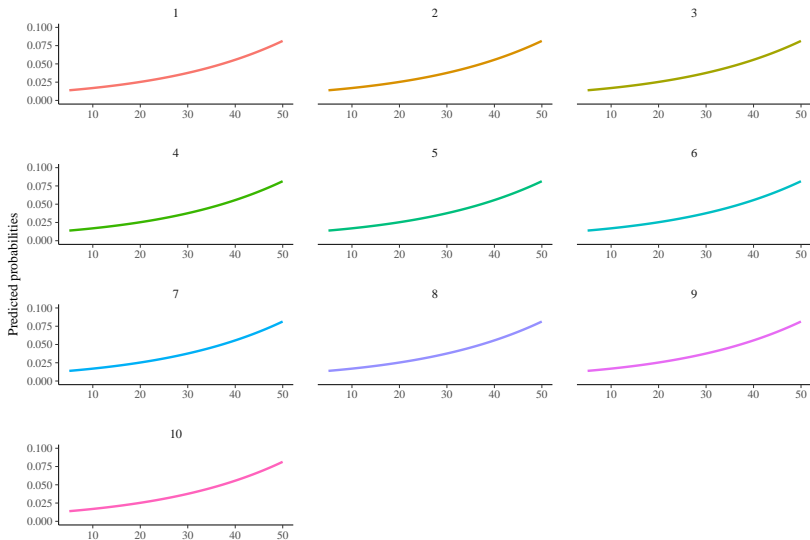

Marginal effects of model predictors

Visualising model: `sjPlot`

```
sjp.glmer(mixed.logis, type = "ri.slope")
```


Predicted probabilities of dbh on dead

Advantages of multilevel models

- Perfect for structured data (space-time)

Advantages of multilevel models

- ► Perfect for structured data (space-time)
- ► Predictors enter at the appropriate level

Advantages of multilevel models

- Perfect for structured data (space-time)
- Predictors enter at the appropriate level
- Accommodate variation in treatment effects

Advantages of multilevel models

- ▶ Perfect for structured data (space-time)
- ▶ Predictors enter at the appropriate level
- ▶ Accommodate variation in treatment effects
- ▶ More efficient inference of regression parameters

Advantages of multilevel models

- Perfect for structured data (space-time)
- Predictors enter at the appropriate level
- Accommodate variation in treatment effects
- More efficient inference of regression parameters
- Using all the data to perform inferences for groups with small sample size

Formula syntax for different models

```
y ~ x + (1 | group) # varying intercepts
y ~ x + (1 + x | group) # varying intercepts and slopes
y ~ x + (1 | group/subgroup) # nested
y ~ x + (1 | group1) + (1 | group2) # varying intercepts, crossed
y ~ x + (1 + x | group1) + (1 + x | group2) # varying intercepts and
slopes, crossed
```

END

Figure 9:

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