

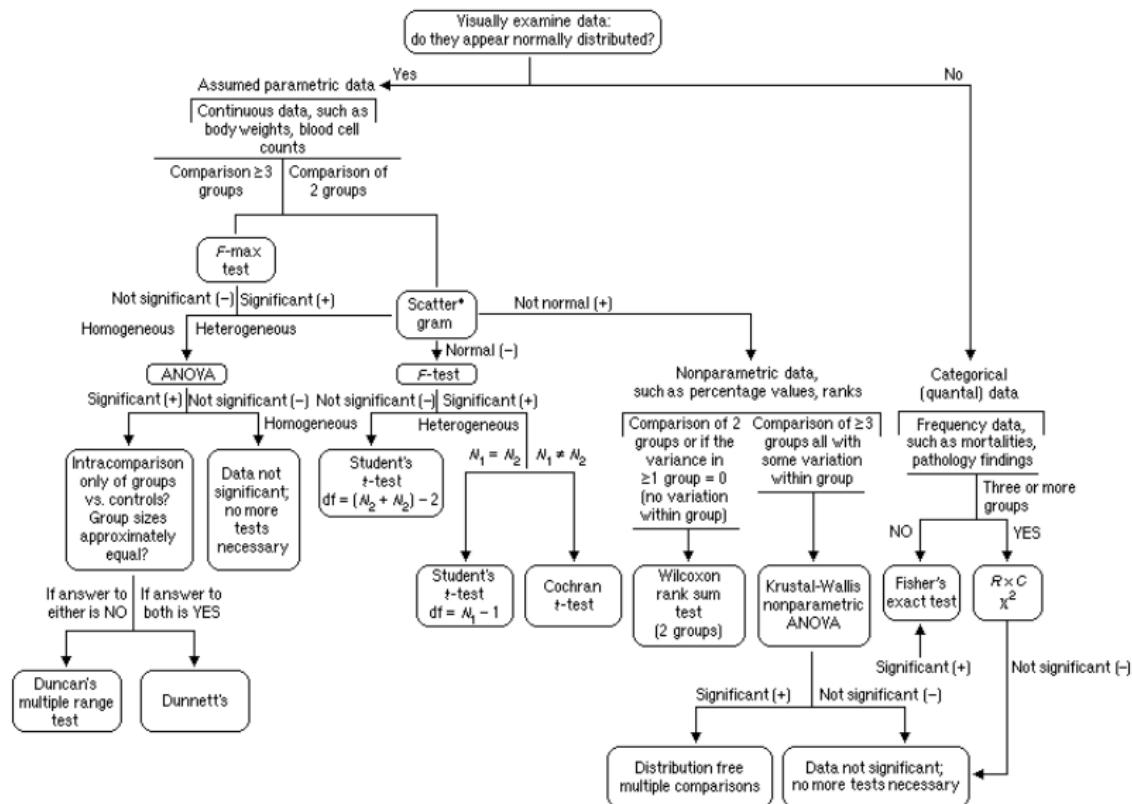
Linear, Generalized, and Mixed/Multilevel models - an introduction with R

Francisco Rodríguez-Sánchez

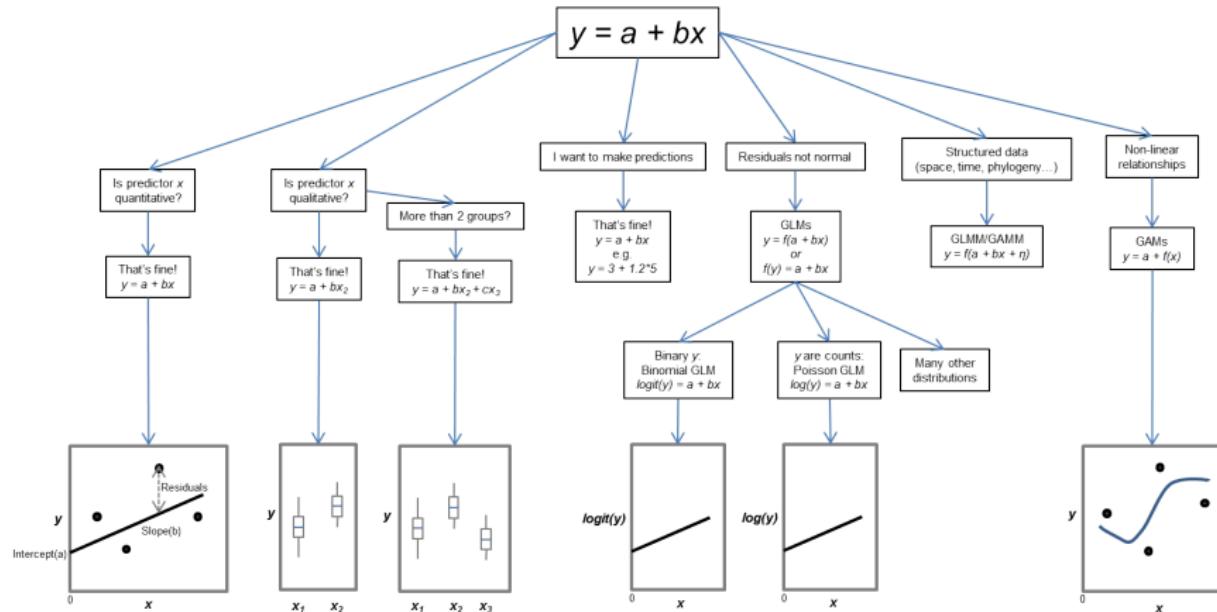
<https://frodriguezsanchez.net>

Introduction to linear models

Modern statistics are easier than this

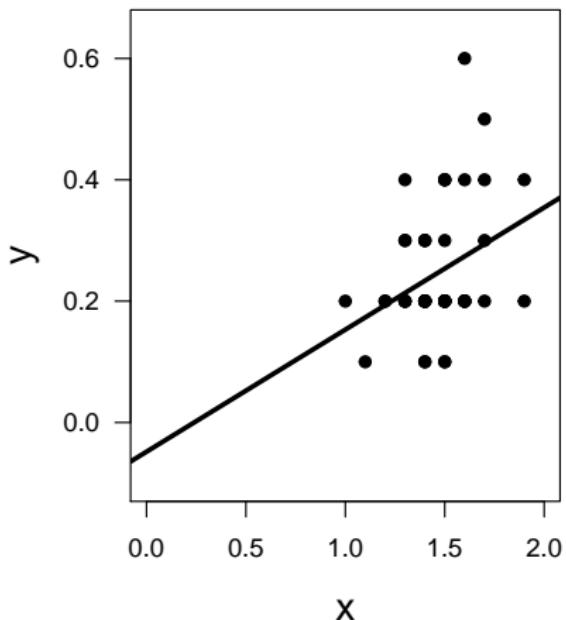


A unified framework



Our unified regression framework

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



Data

y = response variable

x = predictor

Parameters

a = intercept

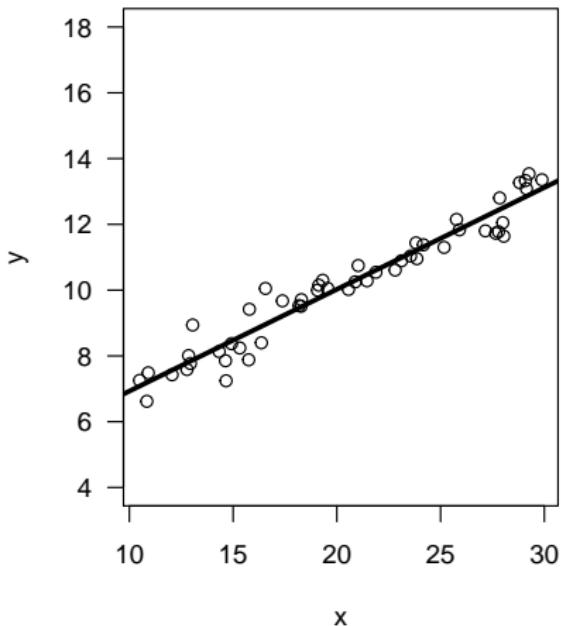
b = slope

σ = residual variation

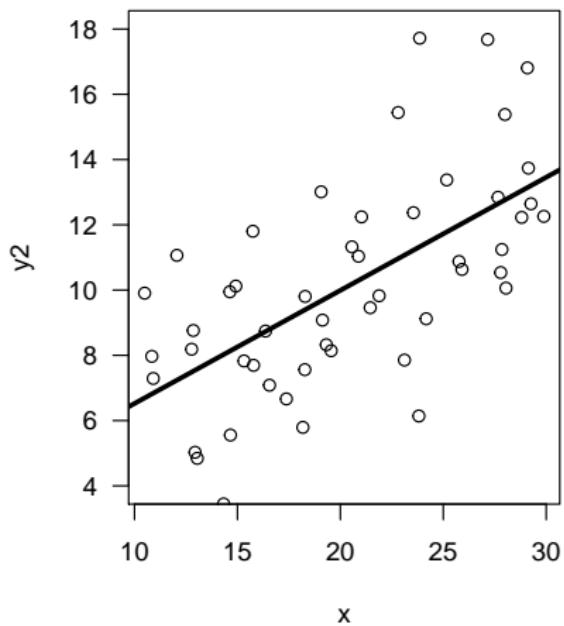
ε = residuals

Residual variation (error)

small



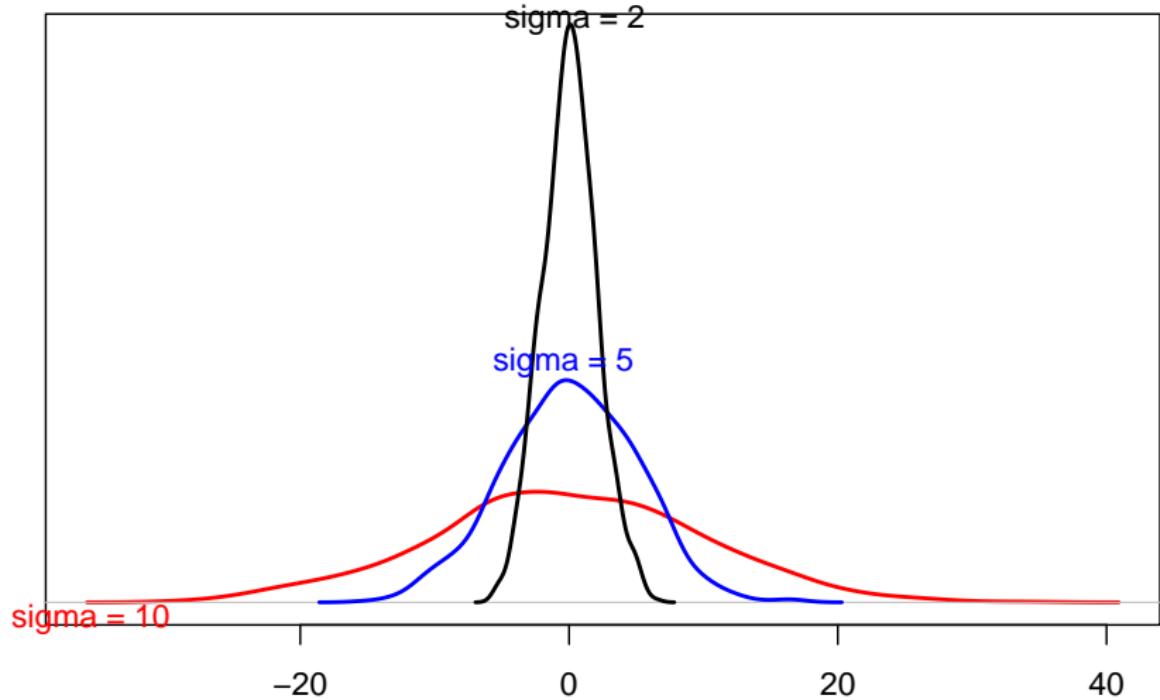
large



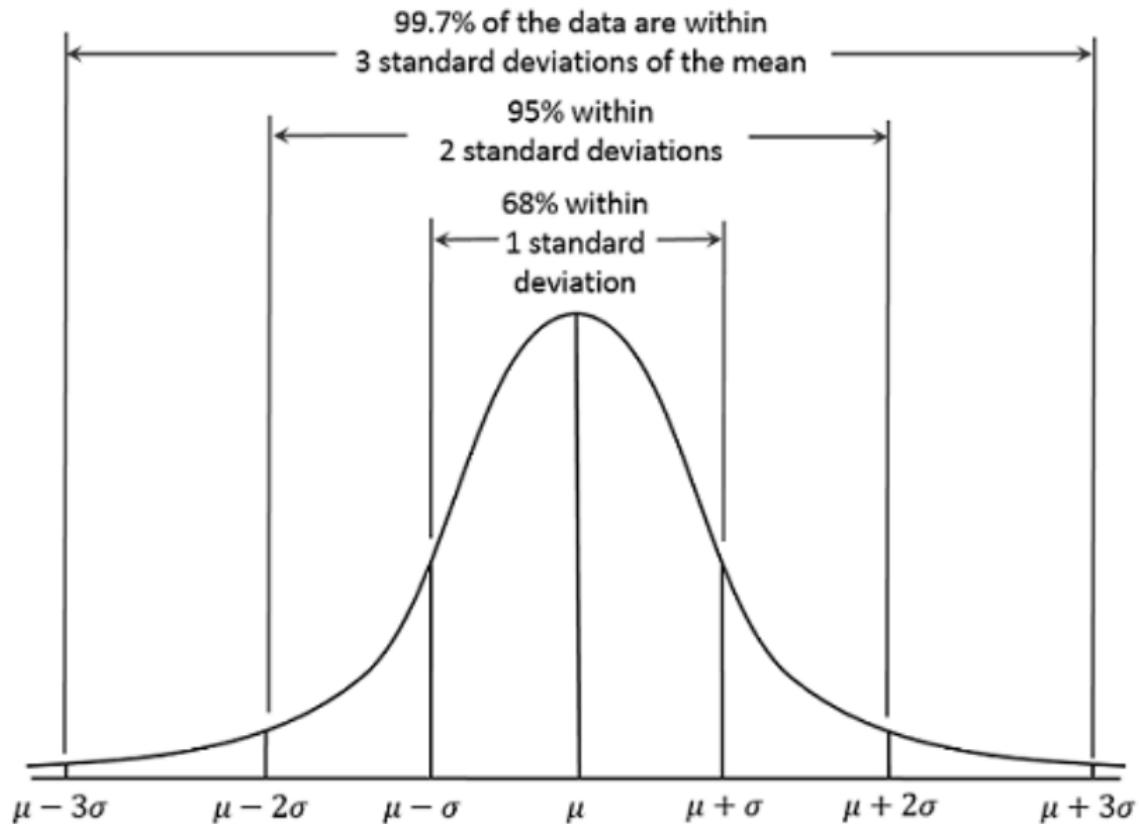
Residual variation

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

Distribution of residuals



In a Normal distribution



Different ways to write same model

$$y_i = a + b x_i + \varepsilon_i$$
$$\varepsilon_i \sim N(0, \sigma^2)$$

$$y_i \sim N(\mu_i, \sigma^2)$$
$$\mu_i = a + b x_i$$
$$\varepsilon_i \sim N(0, \sigma^2)$$

Quiz

<https://pollev.com/franciscorod726>

Linear models

Example dataset: forest trees

- ▶ Download this dataset (or the entire [zip file](#))

```
trees <- read.csv("data/trees.csv")
head(trees)
```

	site	dbh	height	sex	dead
1	4	29.68	36.1	male	0
2	5	33.29	42.3	male	0
3	2	28.03	41.9	female	0
4	5	39.86	46.5	female	0
5	1	47.94	43.9	female	0
6	1	10.82	26.2	male	0

Example dataset: forest trees

- ▶ Download this dataset (or the entire [zip file](#))
- ▶ Import:

```
trees <- read.csv("data/trees.csv")
head(trees)
```

	site	dbh	height	sex	dead
1	4	29.68	36.1	male	0
2	5	33.29	42.3	male	0
3	2	28.03	41.9	female	0
4	5	39.86	46.5	female	0
5	1	47.94	43.9	female	0
6	1	10.82	26.2	male	0

Questions

- ▶ What is the relationship between DBH and height?

Questions

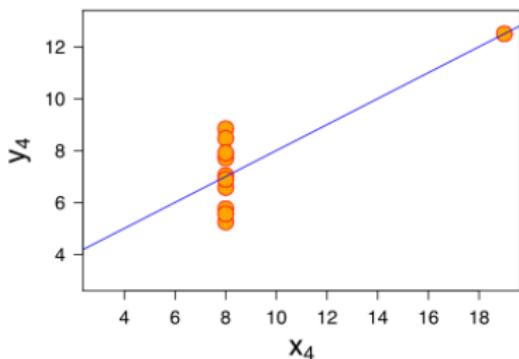
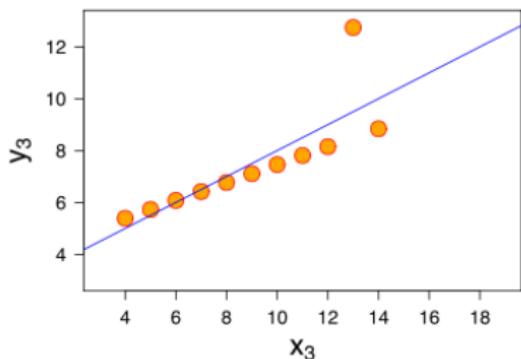
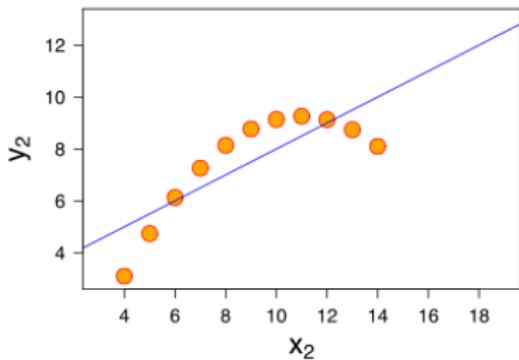
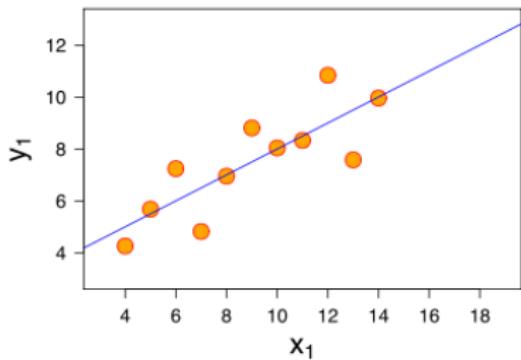
- ▶ What is the relationship between DBH and height?
- ▶ Do taller trees have bigger trunks?

Questions

- ▶ What is the relationship between DBH and height?
- ▶ Do taller trees have bigger trunks?
- ▶ Can we predict height from DBH? How well?

Always plot your data first!

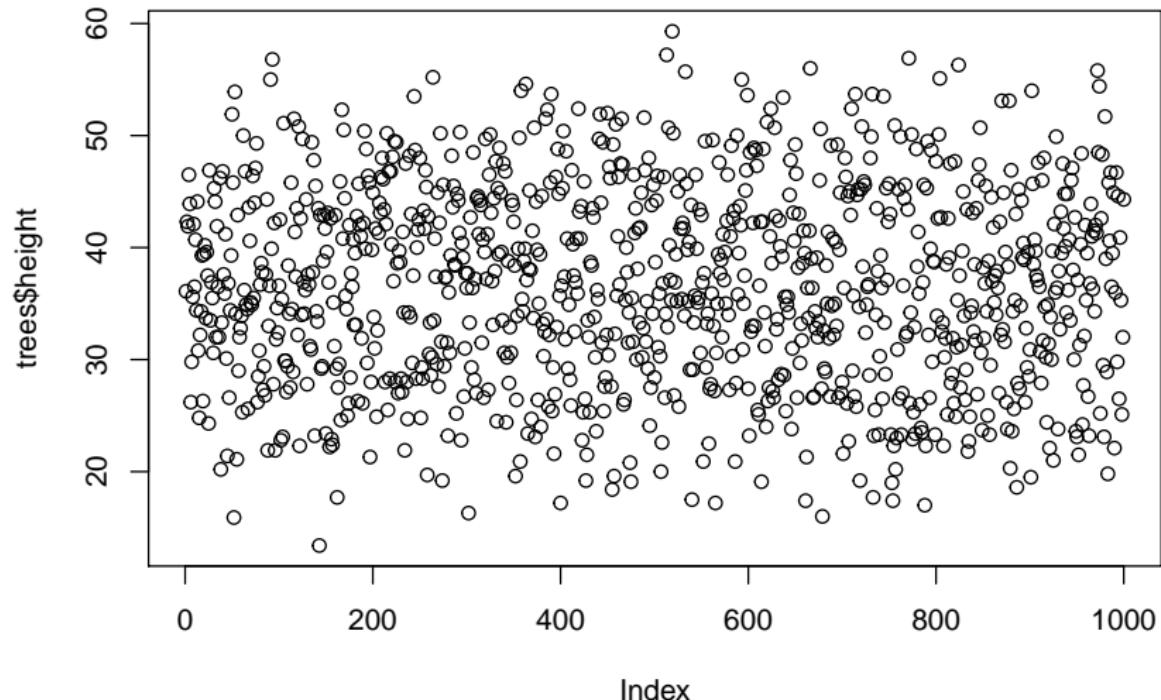
Always plot your data first!



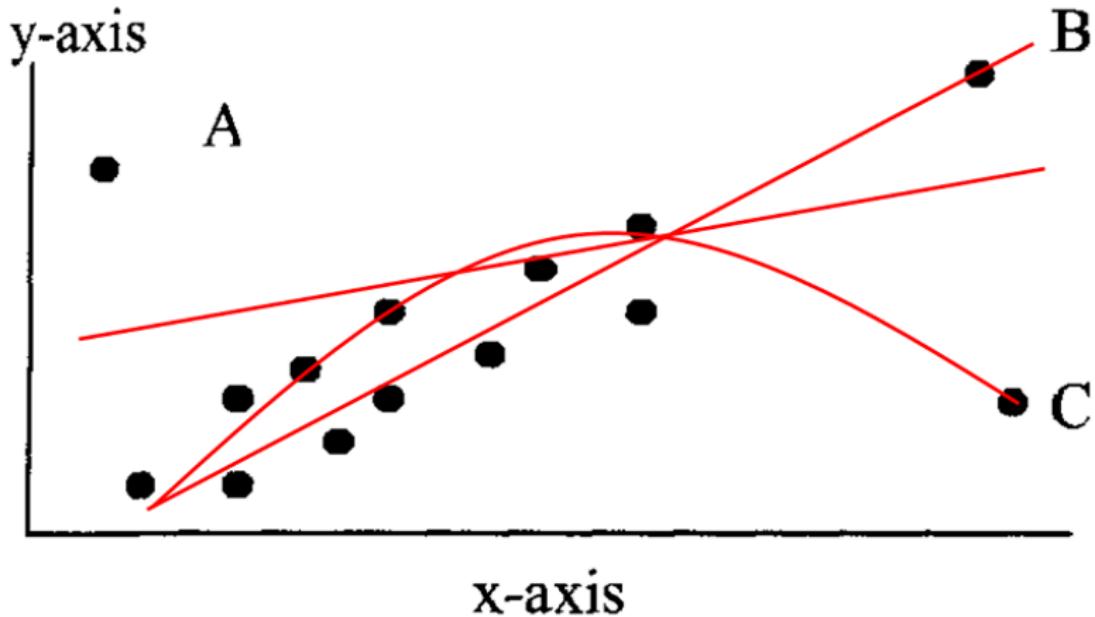
Exploratory Data Analysis (EDA)

Outliers

```
plot(trees$height)
```



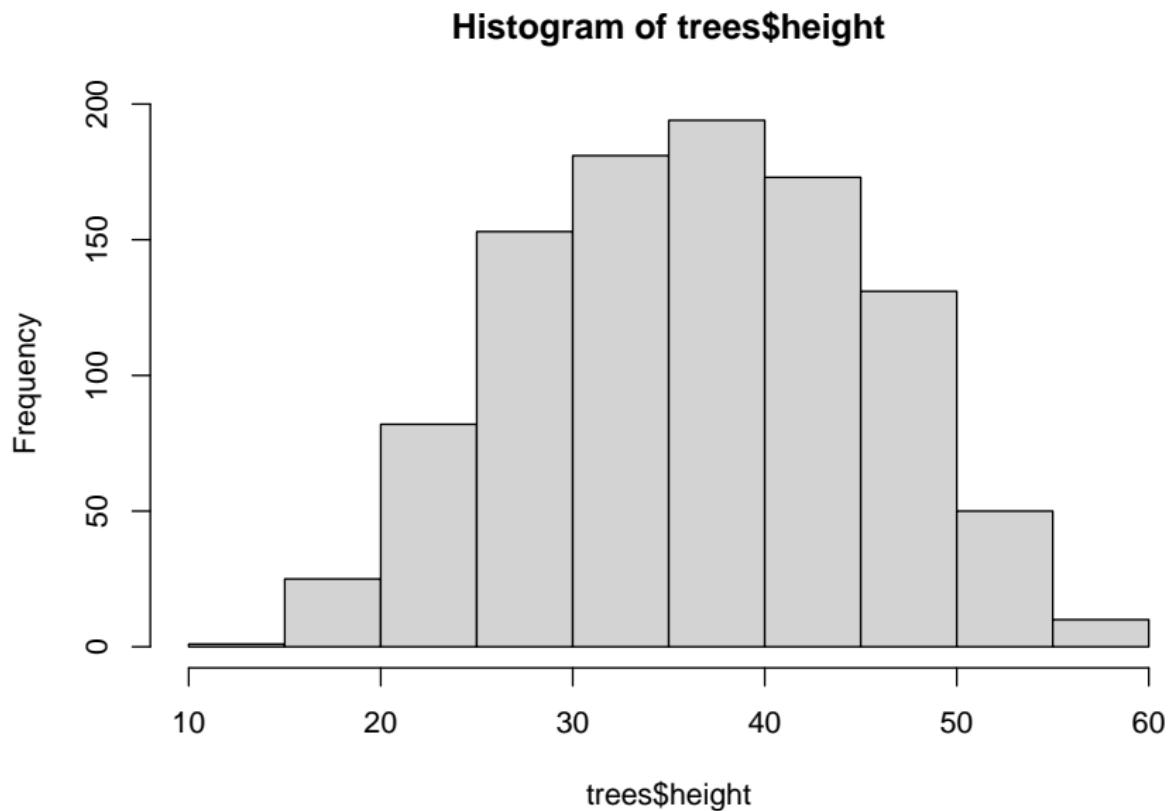
Outliers impact on regression



See <http://rpsychologist.com/d3/correlation/>

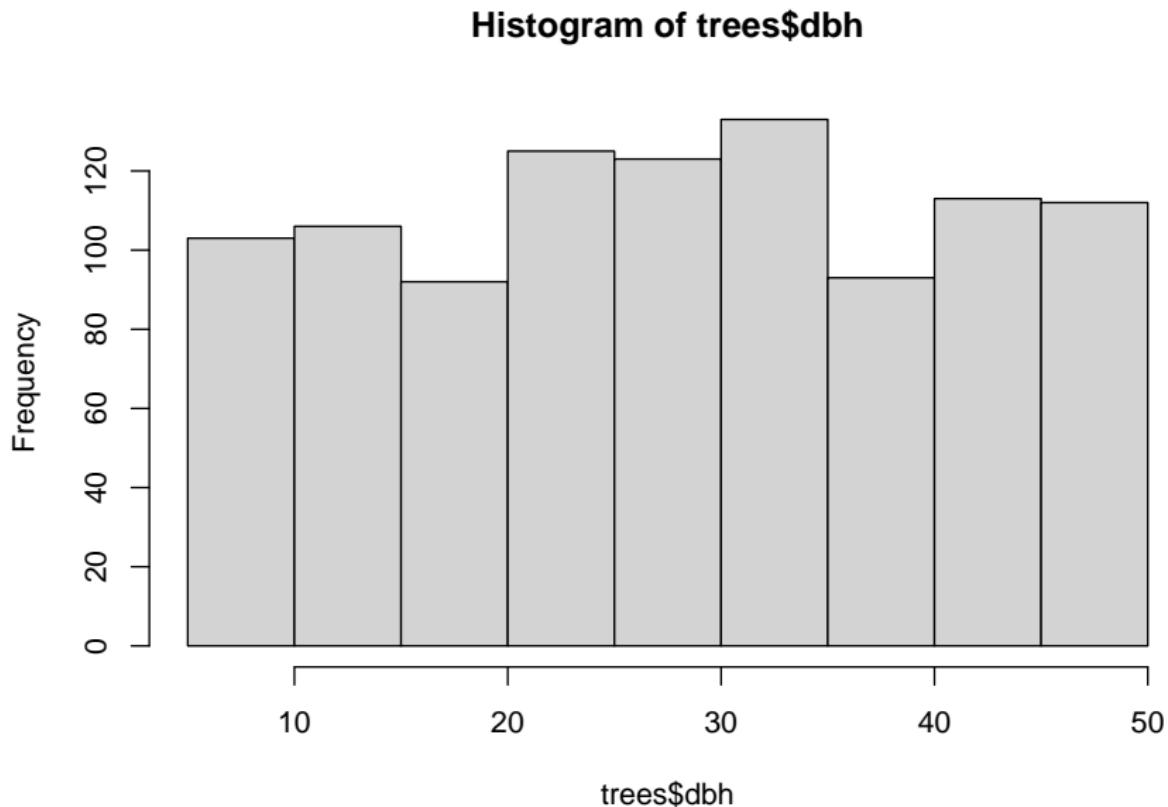
Histogram of response variable

```
hist(trees$height)
```



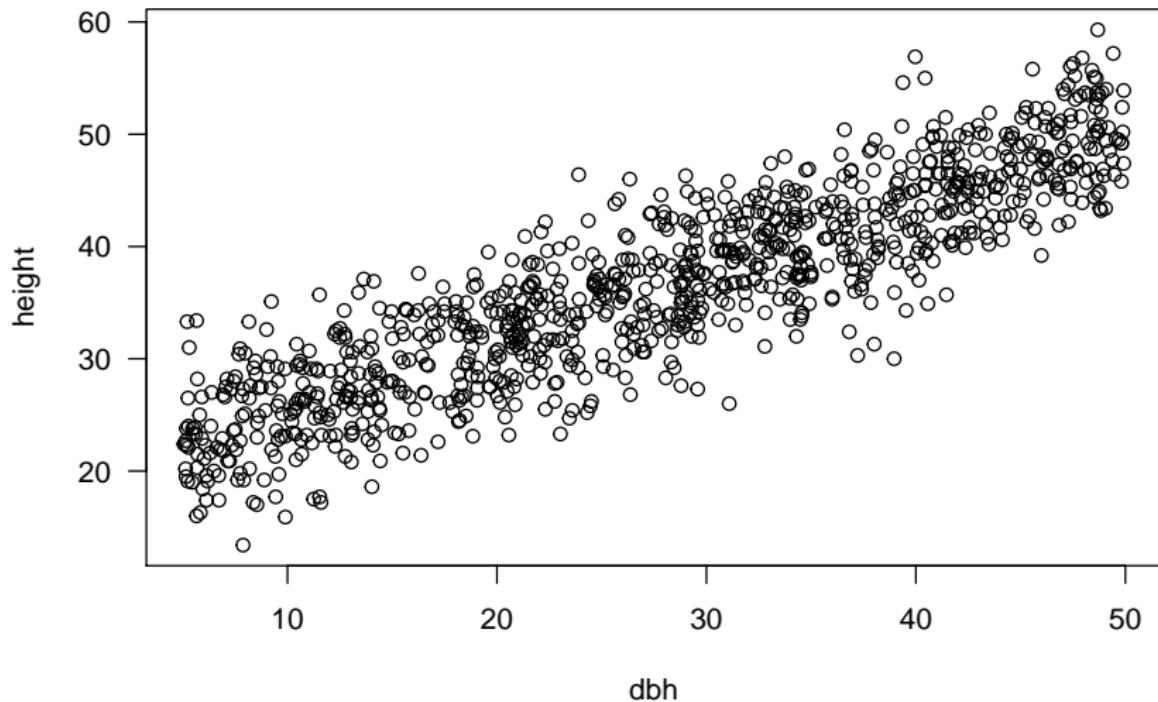
Histogram of predictor variable

```
hist(trees$dbh)
```



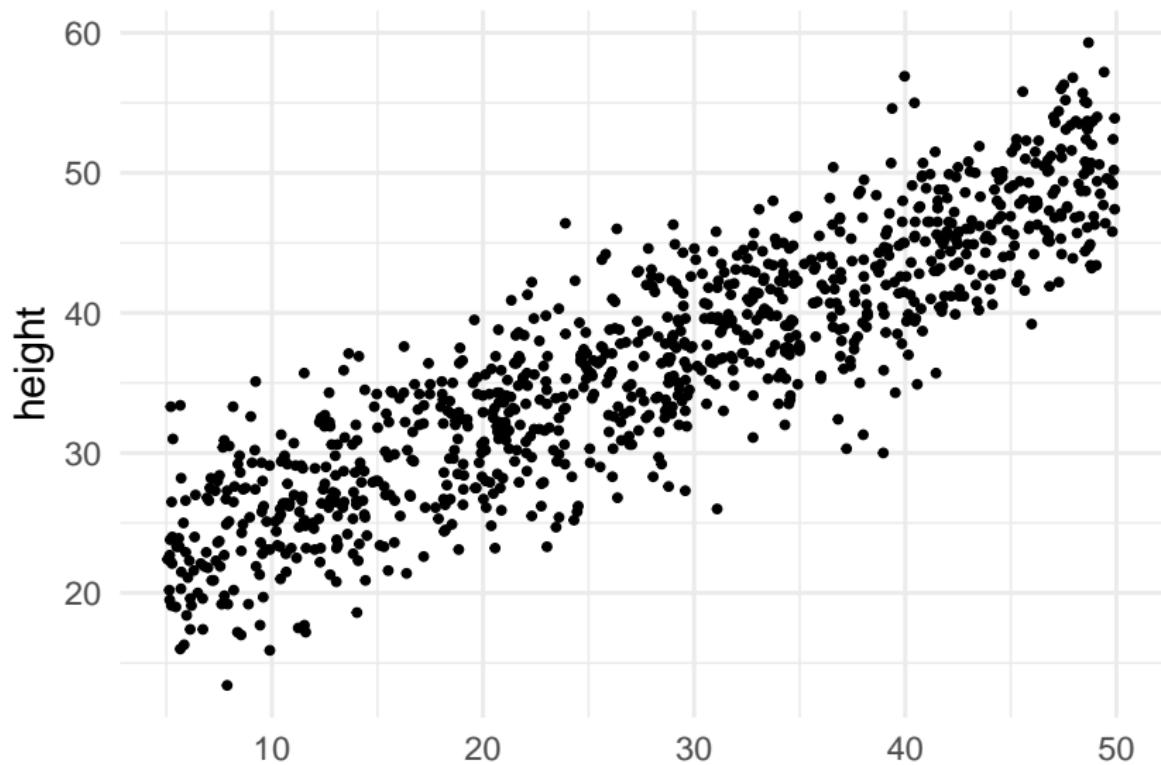
Scatterplot

```
plot(height ~ dbh, data = trees, las = 1)
```



Scatterplot

```
ggplot(trees) +  
  geom_point(aes(dbh, height))
```



Model fitting

Now fit model

Hint: 1m

Now fit model

Hint: `lm`

```
m1 <- lm(height ~ dbh, data = trees)
```

which corresponds to

$$\begin{aligned} Height_i &= a + b \cdot DBH_i + \varepsilon_i \\ \varepsilon_i &\sim N(0, \sigma^2) \end{aligned}$$

Model interpretation

What does this mean?

Call:

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

Residuals:

Min	1Q	Median	3Q	Max
-13.3270	-2.8978	0.1057	2.7924	12.9511

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	19.33920	0.31064	62.26	<2e-16 ***
dbh	0.61570	0.01013	60.79	<2e-16 ***

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

Residual standard error: 4.093 on 998 degrees of freedom

Multiple R-squared: 0.7874, Adjusted R-squared: 0.7871

F-statistic: 3695 on 1 and 998 DF, p-value: < 2.2e-16

Quiz

<https://pollev.com/franciscorod726>

Communicating results

Avoid dichotomania of statistical significance



EDITORIAL • 20 MARCH 2019

It's time to talk about ditching statistical significance

- ▶ 'Never conclude there is 'no difference' or 'no association' just because $p > 0.05$ or CI includes zero'

Avoid dichotomania of statistical significance



EDITORIAL • 20 MARCH 2019

It's time to talk about ditching statistical significance

- ▶ 'Never conclude there is 'no difference' or 'no association' just because $p > 0.05$ or CI includes zero'
- ▶ Estimate and communicate effect sizes and their uncertainty

Avoid dichotomania of statistical significance



EDITORIAL • 20 MARCH 2019

It's time to talk about ditching statistical significance

- ▶ 'Never conclude there is 'no difference' or 'no association' just because $p > 0.05$ or CI includes zero'
- ▶ Estimate and communicate effect sizes and their uncertainty
- ▶ <https://doi.org/10.1038/d41586-019-00857-9>

Communicating results

We found a significant positive relationship between DBH and Height ($p < 0.05$) ($b = 0.61$, $SE = 0.01$).

Presenting model results

```
kable(xtable::xtable(m1), digits = 2)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	19.34	0.31	62.26	0
dbh	0.62	0.01	60.79	0

Presenting model results

```
texreg::texreg(m1, single.row = TRUE)
```

Model 1	
(Intercept)	19.34 (0.31)***
dbh	0.62 (0.01)***
R ²	0.79
Adj. R ²	0.79
Num. obs.	1000

*** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$

Table 2: Statistical models

Retrieving model coefficients

```
coef(m1)
```

	dbh
(Intercept)	19.3391968
	0.6157036

Tidy up model coefficients with broom

```
library(broom)
tidy(m1)

# A tibble: 2 x 5
  term      estimate std.error statistic p.value
  <chr>      <dbl>     <dbl>     <dbl>     <dbl>
1 (Intercept) 19.3      0.311     62.3      0
2 dbh          0.616     0.0101    60.8      0

glance(m1)

# A tibble: 1 x 12
  r.squared adj.r.squared sigma statistic p.value    df logLik     AIC     BIC
  <dbl>          <dbl>    <dbl>     <dbl>     <dbl> <dbl> <dbl> <dbl> <dbl>
1     0.787        0.787   4.09     3695.      0      1 -2827. 5660. 5675.
# ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

Confidence intervals

```
confint(m1)

              2.5 %    97.5 %
(Intercept) 18.7296053 19.948788
dbh          0.5958282  0.635579
```

Using effects package

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

model: height ~ dbh

dbh effect

dbh	5	20	30	40	50
dbh	22.41771	31.65327	37.81030	43.96734	50.12438

Lower 95 Percent Confidence Limits

dbh	5	20	30	40	50
dbh	21.89682	31.35487	37.55287	43.61733	49.61669

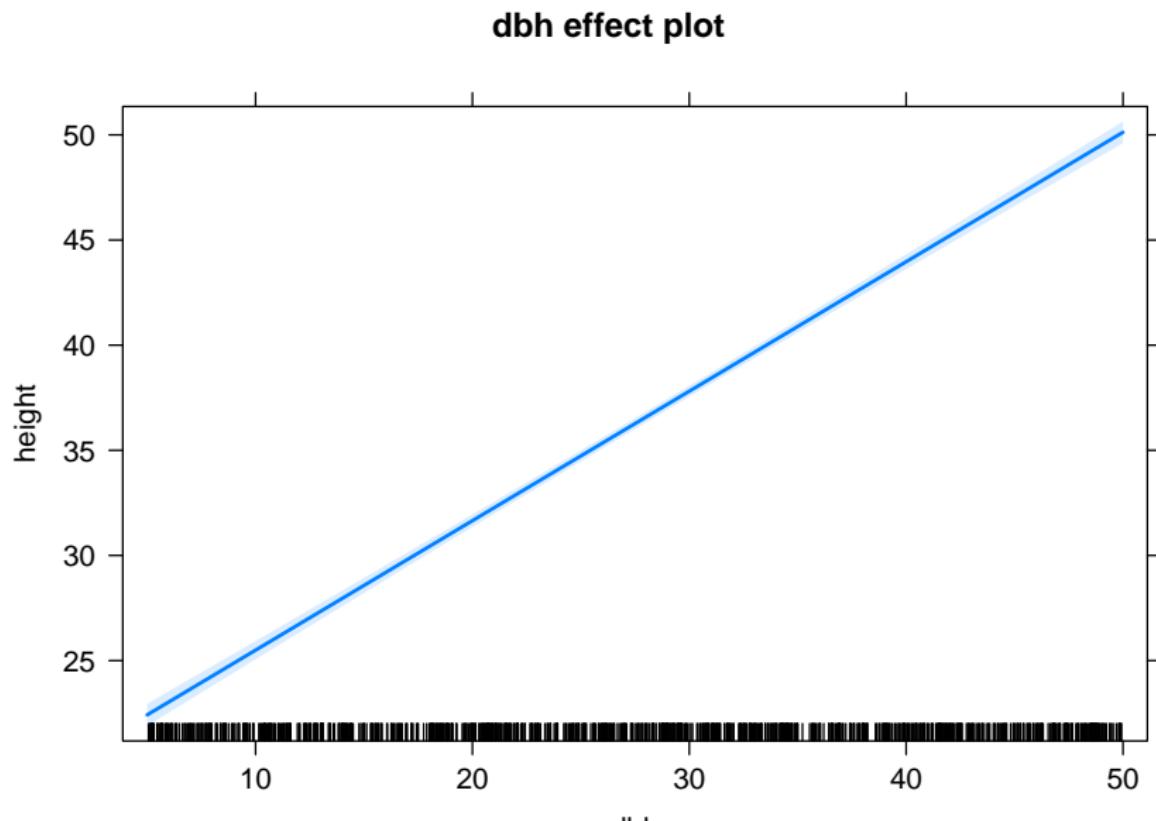
Upper 95 Percent Confidence Limits

dbh	5	20	30	40	50
dbh	22.93861	31.95167	38.06774	44.31735	50.63207

Visualising fitted model

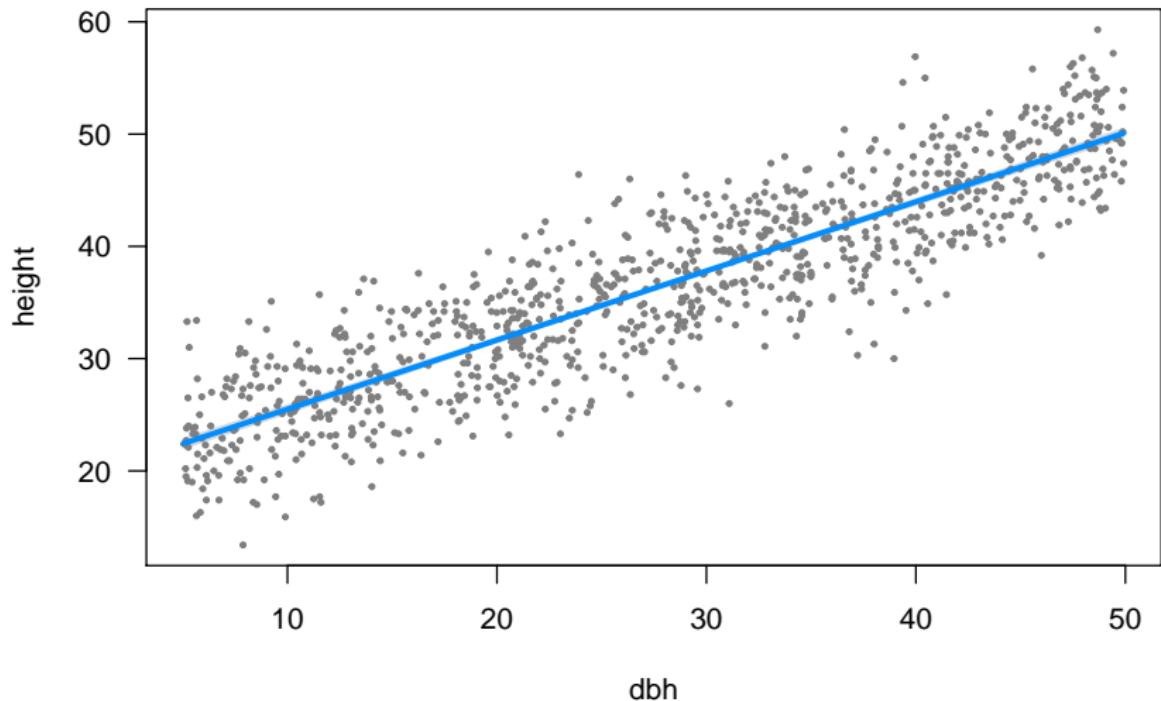
Plot effects

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

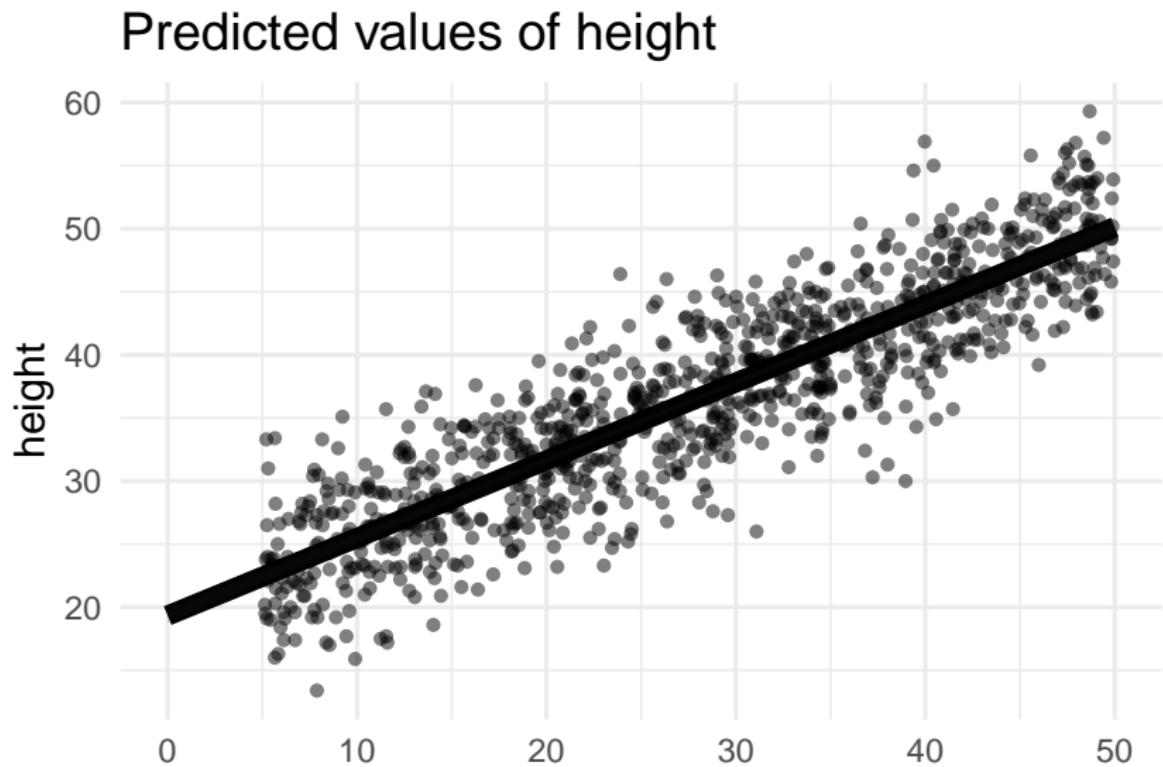


Plot model (visreg)

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



Plot model (sjPlot - ggplot2)



Model checking

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

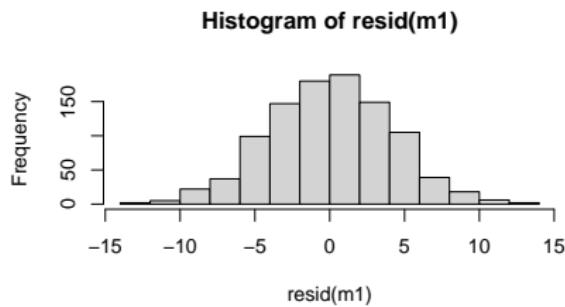
- ▶ Linearity (transformations, GAM...)
- ▶ Residuals:
 - ▶ Independent
 - ▶ Equal variance
 - ▶ Normal

Linear model assumptions

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

Are residuals normal?

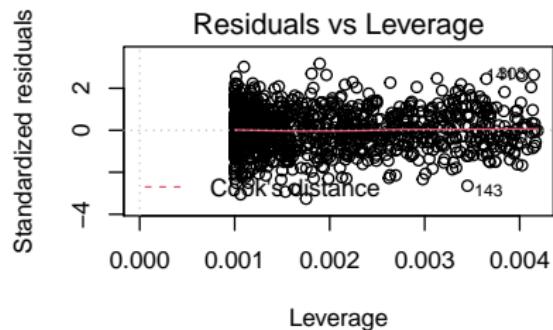
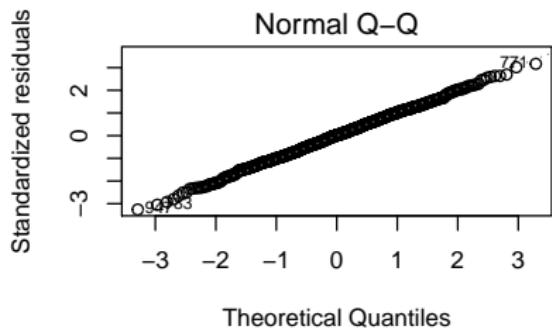
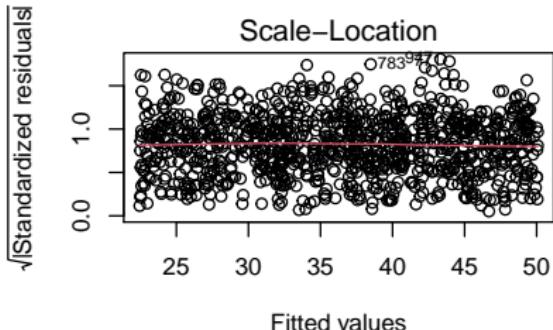
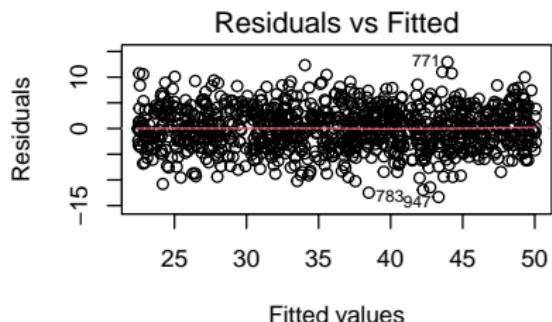
```
hist(resid(m1))
```



```
lm(formula = height ~ dbh, data = trees)
  coef.est coef.se
  (Intercept) 19.34      0.31
  dbh          0.62      0.01
  ---
  n = 1000, k = 2
  residual sd = 4.09, R-Squared = 0.79
```

SD of residuals = 4.09 coincides with estimate of sigma.

Model checking: residuals

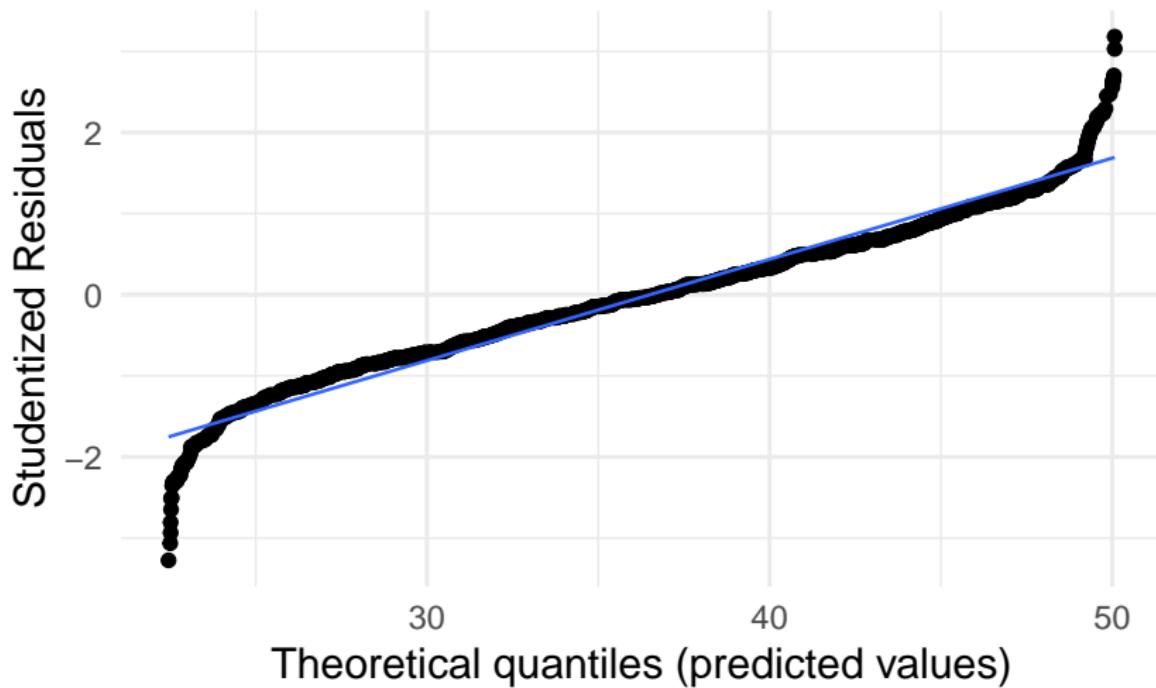


Model checking (sjPlot)

```
plot_model(m1, type = "diag")[[1]]
```

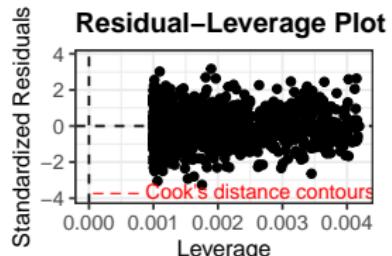
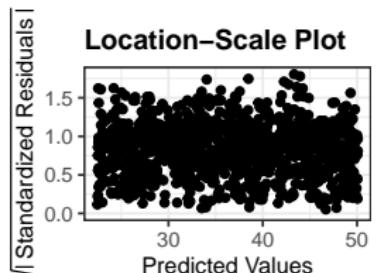
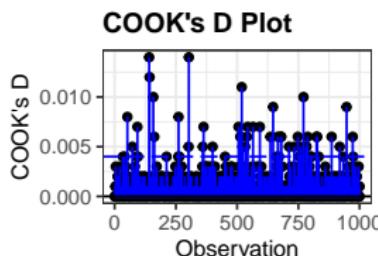
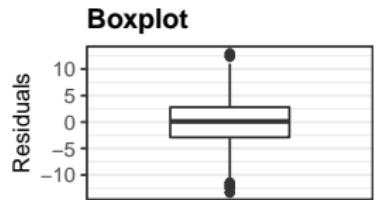
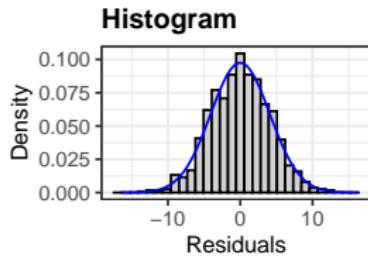
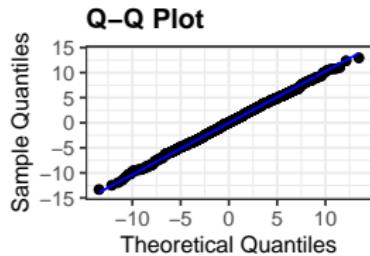
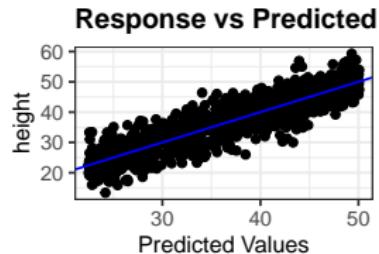
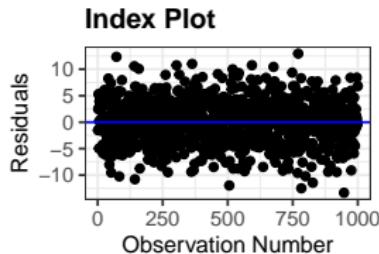
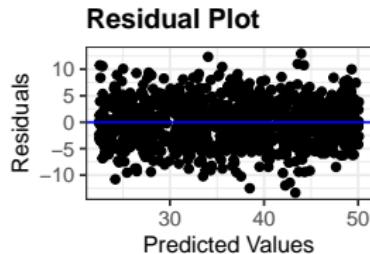
Non-normality of residuals and outliers

Dots should be plotted along the line



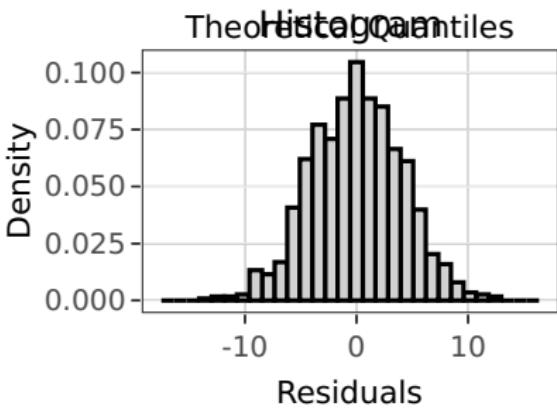
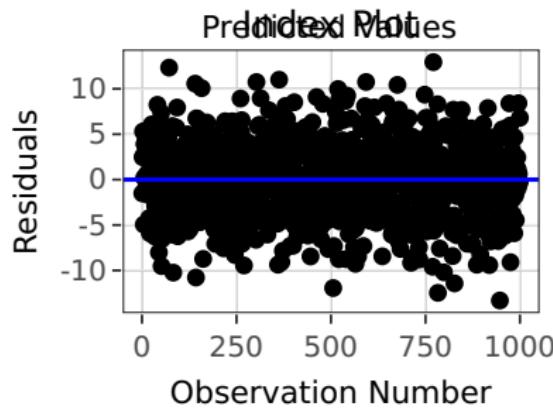
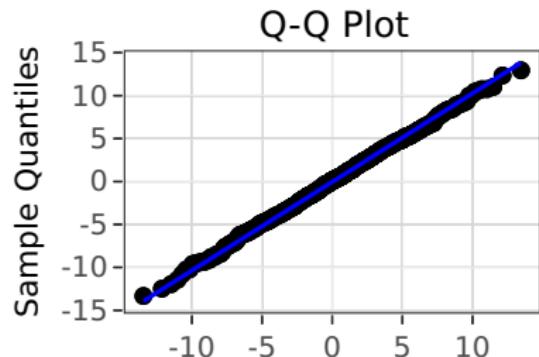
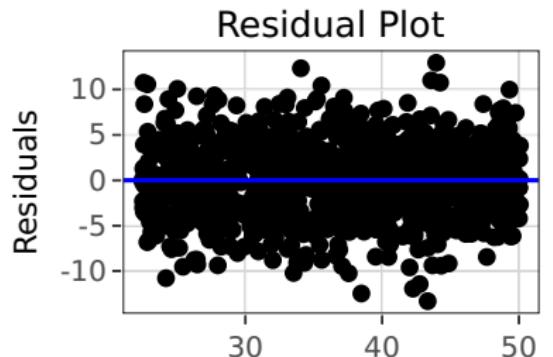
Model checking (ggResidpanel)

```
ggResidpanel:::resid_panel(m1, plots = "all")
```



Interactive model checking (ggResidpanel)

```
ggResidpanel:::resid_interact(m1)
```



Using model for prediction

How good is the model in predicting tree height?

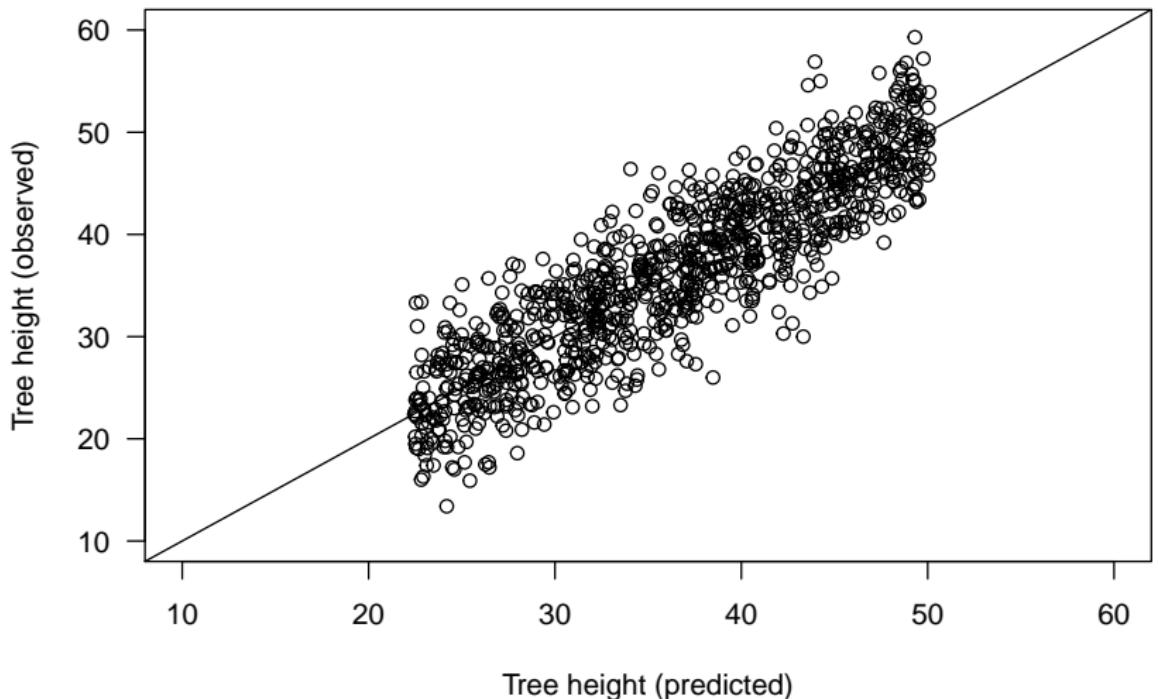
`fitted` gives predictions for each observation

```
trees$height.pred <- fitted(m1)  
head(trees)
```

	site	dbh	height	sex	dead	height.pred
1	4	29.68	36.1	male	0	37.61328
2	5	33.29	42.3	male	0	39.83597
3	2	28.03	41.9	female	0	36.59737
4	5	39.86	46.5	female	0	43.88114
5	1	47.94	43.9	female	0	48.85603
6	1	10.82	26.2	male	0	26.00111

Calibration plot: Observed vs Predicted values

```
plot(trees$height.pred, trees$height,  
      xlab = "Tree height (predicted)", ylab = "Tree height (obse
```



Using fitted model for prediction

Q: Expected tree height if DBH = 39 cm?

```
new.dbh <- data.frame(dbh = c(39))
predict(m1, new.dbh, se.fit = TRUE)
```

```
$fit
```

```
1
```

```
43.35164
```

```
$se.fit
```

```
[1] 0.1715514
```

```
$df
```

```
[1] 998
```

```
$residual.scale
```

```
[1] 4.092629
```

Using fitted model for prediction

Q: Expected tree height if DBH = 39 cm?

```
predict(m1, new.dbh, interval = "confidence")
```

	fit	lwr	upr
1	43.35164	43.01499	43.68828

```
predict(m1, new.dbh, interval = "prediction")
```

	fit	lwr	upr
1	43.35164	35.31344	51.38983

Workflow

- ▶ **Visualise data**

Workflow

- ▶ **Visualise data**
- ▶ **Understand fitted model** (`summary`, `allEffects...`)

Workflow

- ▶ **Visualise data**
- ▶ **Understand fitted model** (`summary`, `allEffects...`)
- ▶ **Visualise model** (`plot(allEffects)`, `visreg`, `plot_model...`)

Workflow

- ▶ **Visualise data**
- ▶ **Understand fitted model** (`summary`, `allEffects...`)
- ▶ **Visualise model** (`plot(allEffects)`, `visreg`, `plot_model...`)
- ▶ **Check model** (`plot`, `resid_panel`, `calibration plot...`)

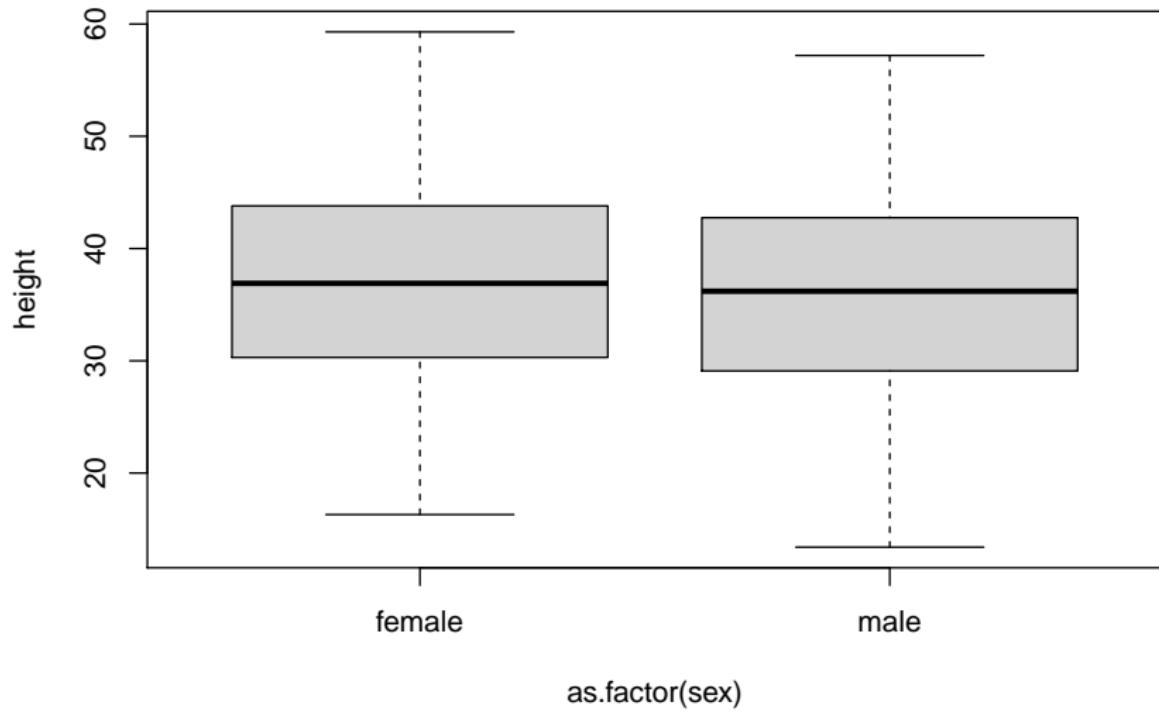
Workflow

- ▶ **Visualise data**
- ▶ **Understand fitted model** (`summary`, `allEffects...`)
- ▶ **Visualise model** (`plot(allEffects)`, `visreg`, `plot_model...`)
- ▶ **Check model** (`plot`, `resid_panel`, `calibration plot...`)
- ▶ **Predict** (`fitted`, `predict`)

Categorical predictors (factors)

Q: Does tree height vary with sex?

```
plot(height ~ as.factor(sex), data = trees)
```



Model height ~ sex

```
m2 <- lm(height ~ sex, data = trees)
```

Call:

```
lm(formula = height ~ sex, data = trees)
```

Residuals:

Min	1Q	Median	3Q	Max
-22.6881	-6.7881	-0.0097	6.7261	22.3687

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	36.9312	0.3981	92.778	<2e-16 ***
sexmale	-0.8432	0.5607	-1.504	0.133

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

Residual standard error: 8.865 on 998 degrees of freedom

Multiple R-squared: 0.002261, Adjusted R-squared: 0.001261

F-statistic: 2.261 on 1 and 998 DF, p-value: 0.133

Linear model with categorical predictors

```
m2 <- lm(height ~ sex, data = trees)
```

corresponds to

$$\begin{aligned} Height_i &= a + b_{male} + \varepsilon_i \\ \varepsilon_i &\sim N(0, \sigma^2) \end{aligned}$$

Model height ~ sex

```
m2 <- lm(height ~ sex, data = trees)
```

Call:

```
lm(formula = height ~ sex, data = trees)
```

Residuals:

Min	1Q	Median	3Q	Max
-22.6881	-6.7881	-0.0097	6.7261	22.3687

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)		
(Intercept)	36.9312	0.3981	92.778	<2e-16 ***		
sexmale	-0.8432	0.5607	-1.504	0.133		

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

Residual standard error: 8.865 on 998 degrees of freedom

Multiple R-squared: 0.002261, Adjusted R-squared: 0.001261

F-statistic: 2.261 on 1 and 998 DF, p-value: 0.133

Quiz

<https://pollev.com/franciscorod726>

Presenting model results

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	36.93	0.40	92.78	0.00
sexmale	-0.84	0.56	-1.50	0.13

Effects: Height ~ sex

Compare CIs

```
summary(allEffects(m2))
```

model: height ~ sex

sex effect

sex

female	male
36.93125	36.08810

Lower 95 Percent Confidence Limits

sex

female	male
36.15012	35.31319

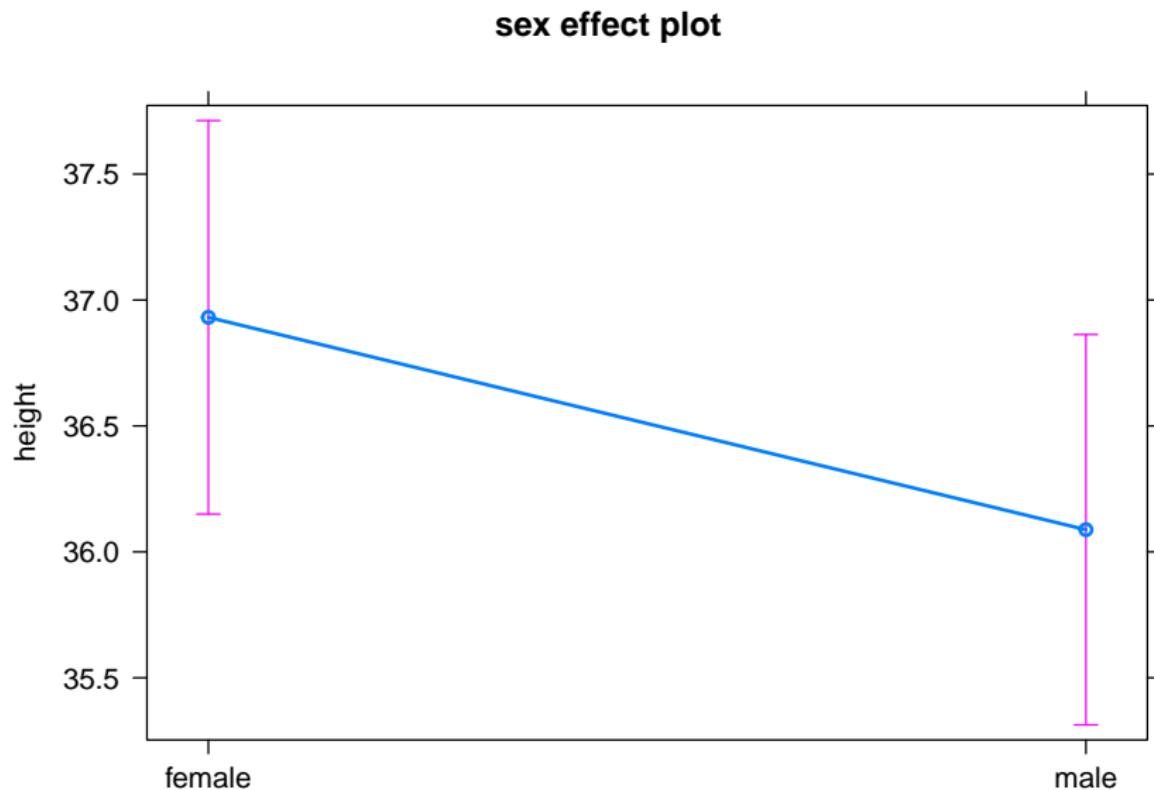
Upper 95 Percent Confidence Limits

sex

female	male
37.71238	36.86300

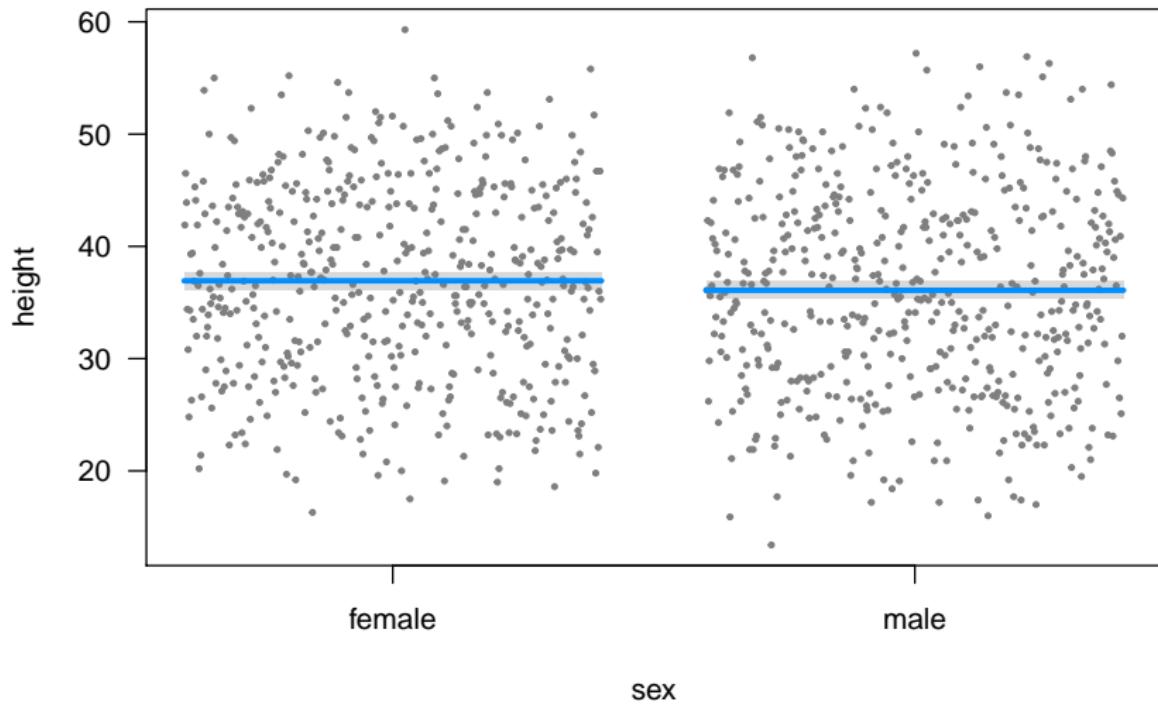
Plot

```
plot(allEffects(m2))
```



Plot (visreg)

```
visreg(m2)
```



Plot model (sjPlot)

```
plot_model(m2, type = "eff")
```

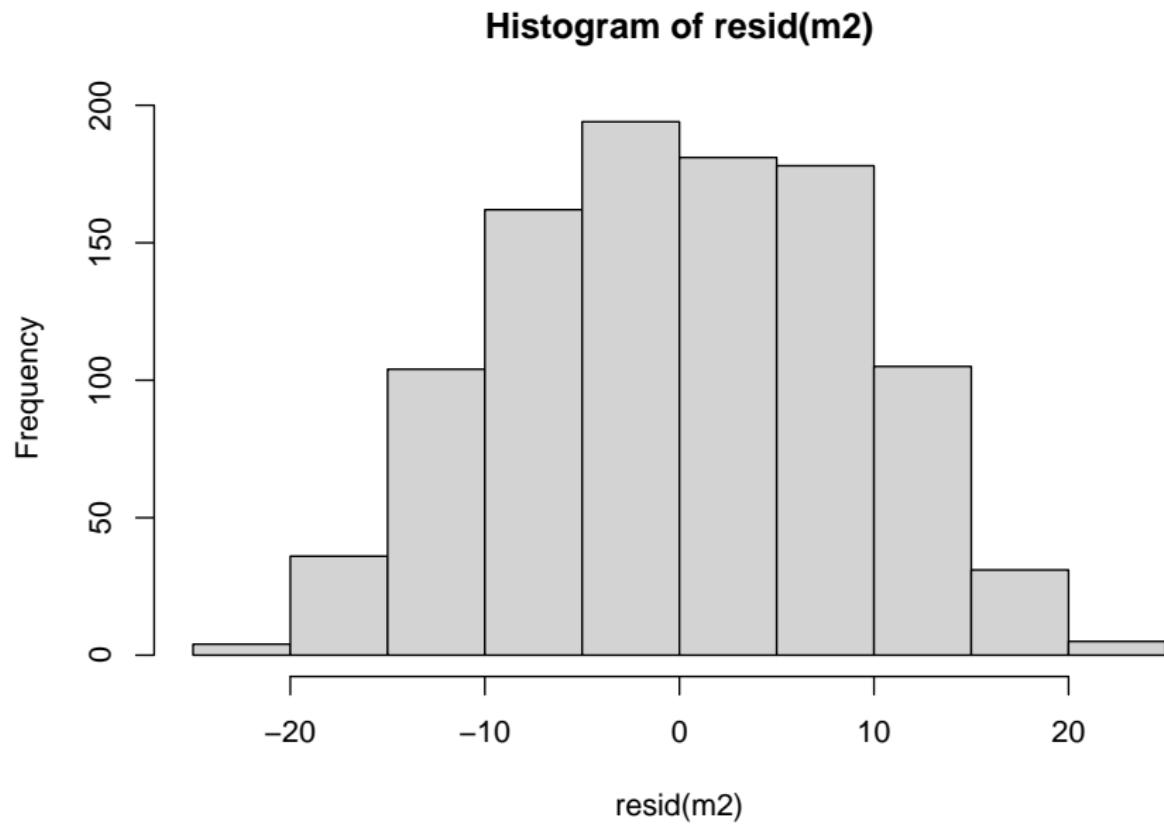
\$sex

Predicted values of height

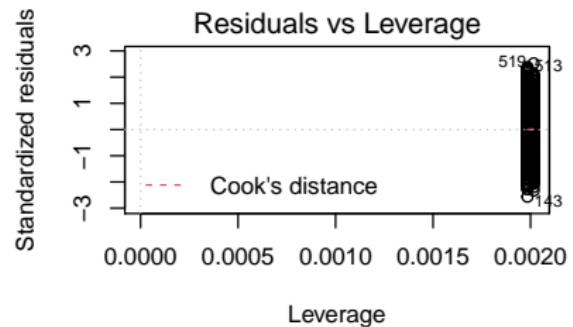
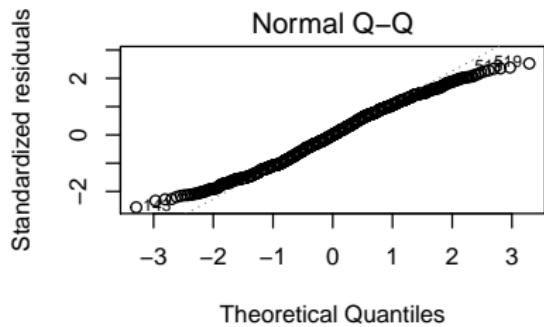
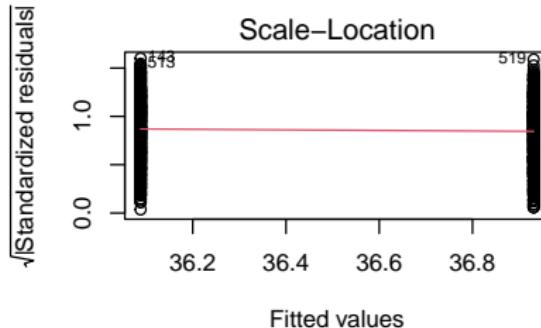
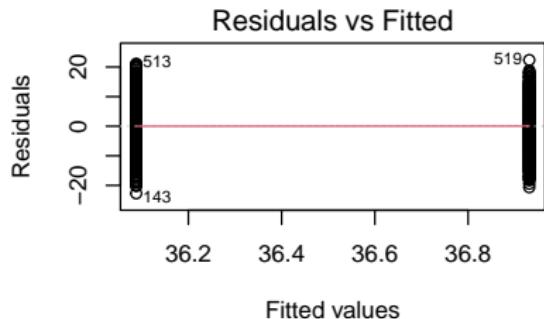


Model checking: residuals

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

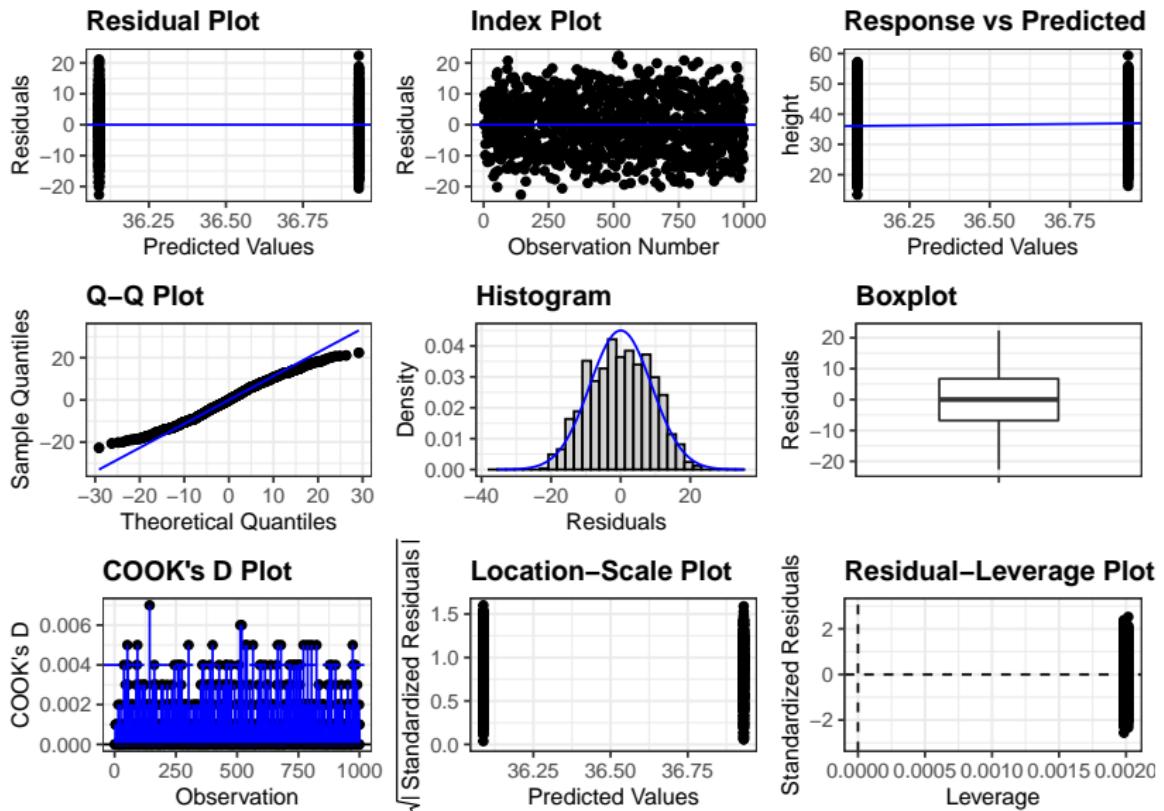


Model checking: residuals



Model checking (ggResidpanel)

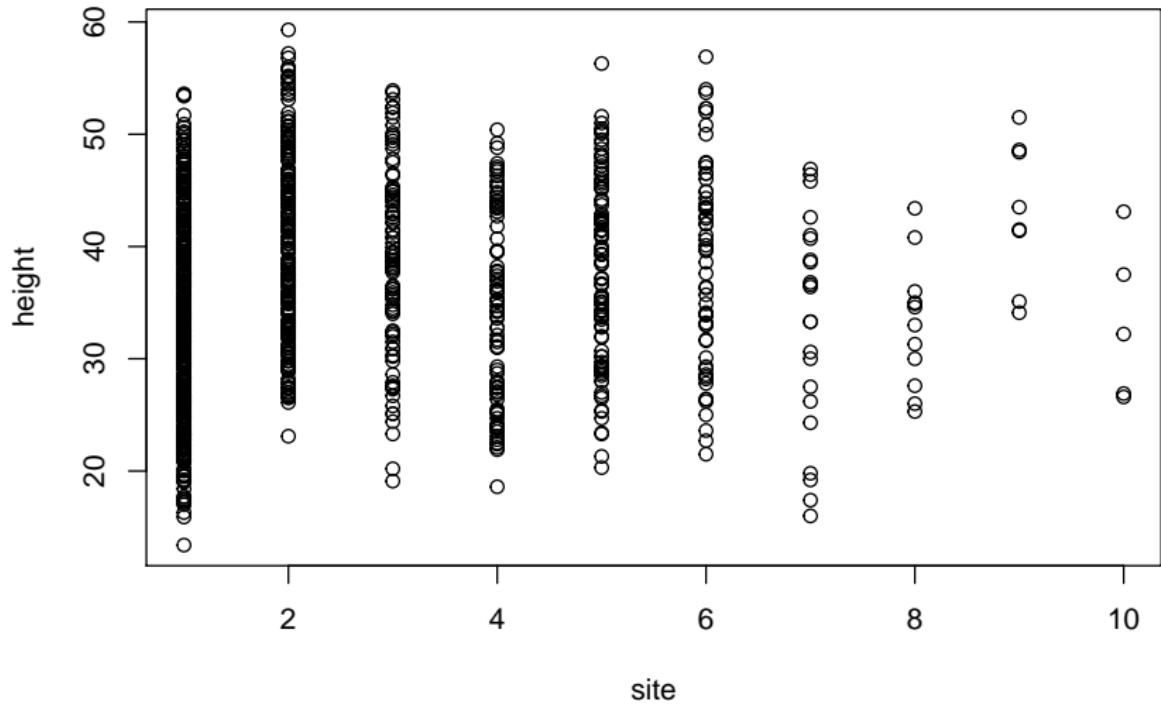
```
ggResidpanel:::resid_panel(m2, plots = "all")
```



Q: Does height differ among field sites?

Plot data first

```
plot(height ~ site, data = trees)
```



Linear model with categorical predictors

```
m3 <- lm(height ~ site, data = trees)
```

$$y_i = a + b_{site2} + c_{site3} + d_{site4} + e_{site5} + \dots + \varepsilon_i$$
$$\varepsilon_i \sim N(0, \sigma^2)$$

Model Height ~ site

All right here?

```
m3 <- lm(height ~ site, data = trees)
```

Call:

```
lm(formula = height ~ site, data = trees)
```

Residuals:

Min	1Q	Median	3Q	Max
-22.4498	-6.7049	0.0709	6.7537	23.0640

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	35.4636	0.4730	74.975	< 2e-16 ***
site	0.3862	0.1413	2.733	0.00639 **

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

Residual standard error: 8.842 on 998 degrees of freedom

Multiple R-squared: 0.007429, Adjusted R-squared: 0.006435

F-statistic: 7.47 on 1 and 998 DF, p-value: 0.006385

site is a factor!

```
trees$site <- as.factor(trees$site)
```

Model Height ~ site

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

Residuals:
    Min      1Q  Median      3Q     Max
-20.4416 -6.9004  0.0379  6.3051 19.7584

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 33.8416    0.4266  79.329 < 2e-16 ***
site2        6.3411    0.7126   8.899 < 2e-16 ***
site3        4.9991    0.9828   5.086 4.36e-07 ***
site4        0.5329    0.9872   0.540  0.58949    
site5        4.3723    0.9425   4.639 3.97e-06 ***
site6        4.7601    1.1709   4.065 5.18e-05 ***
site7       -0.7416    1.8506  -0.401  0.68871    
site8       -0.6832    2.4753  -0.276  0.78258    
site9        9.1709    3.0165   3.040  0.00243 **  
site10      -0.5816    3.8013  -0.153  0.87843    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 8.446 on 990 degrees of freedom
Multiple R-squared: 0.1016, Adjusted R-squared: 0.09344
F-statistic: 12.44 on 9 and 990 DF, p-value: < 2.2e-16

Presenting model results

```
kable(xtable::xtable(m3), digits = 2)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	33.84	0.43	79.33	0.00
site2	6.34	0.71	8.90	0.00
site3	5.00	0.98	5.09	0.00
site4	0.53	0.99	0.54	0.59
site5	4.37	0.94	4.64	0.00
site6	4.76	1.17	4.07	0.00
site7	-0.74	1.85	-0.40	0.69
site8	-0.68	2.48	-0.28	0.78
site9	9.17	3.02	3.04	0.00
site10	-0.58	3.80	-0.15	0.88

Estimated tree heights for each site

```
summary(allEffects(m3))

model: height ~ site

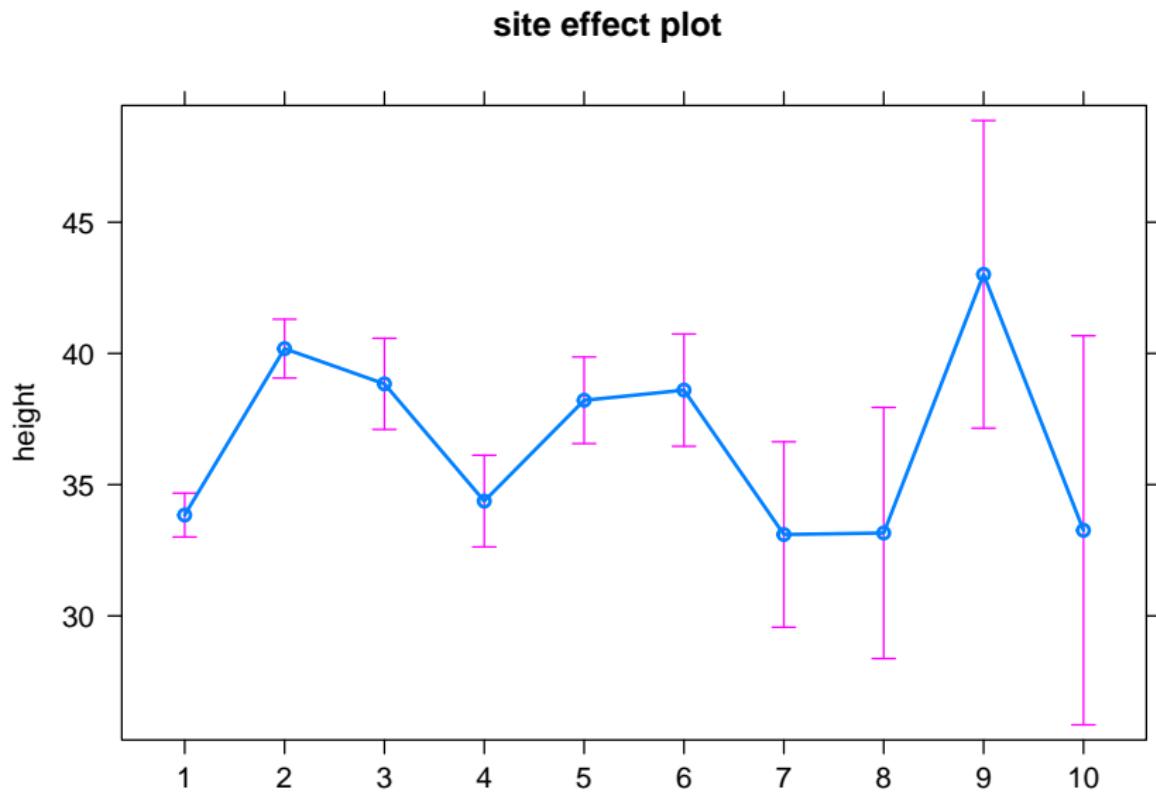
site effect
site
    1      2      3      4      5      6      7      8
33.84158 40.18265 38.84066 34.37444 38.21386 38.60167 33.10000 33.15833
    9      10
43.01250 33.26000

Lower 95 Percent Confidence Limits
site
    1      2      3      4      5      6      7      8
33.00444 39.06264 37.10317 32.62733 36.56463 36.46190 29.56629 28.37367
    9      10
37.15251 25.84764

Upper 95 Percent Confidence Limits
site
    1      2      3      4      5      6      7      8
34.67872 41.30265 40.57814 36.12156 39.86309 40.74143 36.63371 37.94299
    9      10
48.87249 40.67236
```

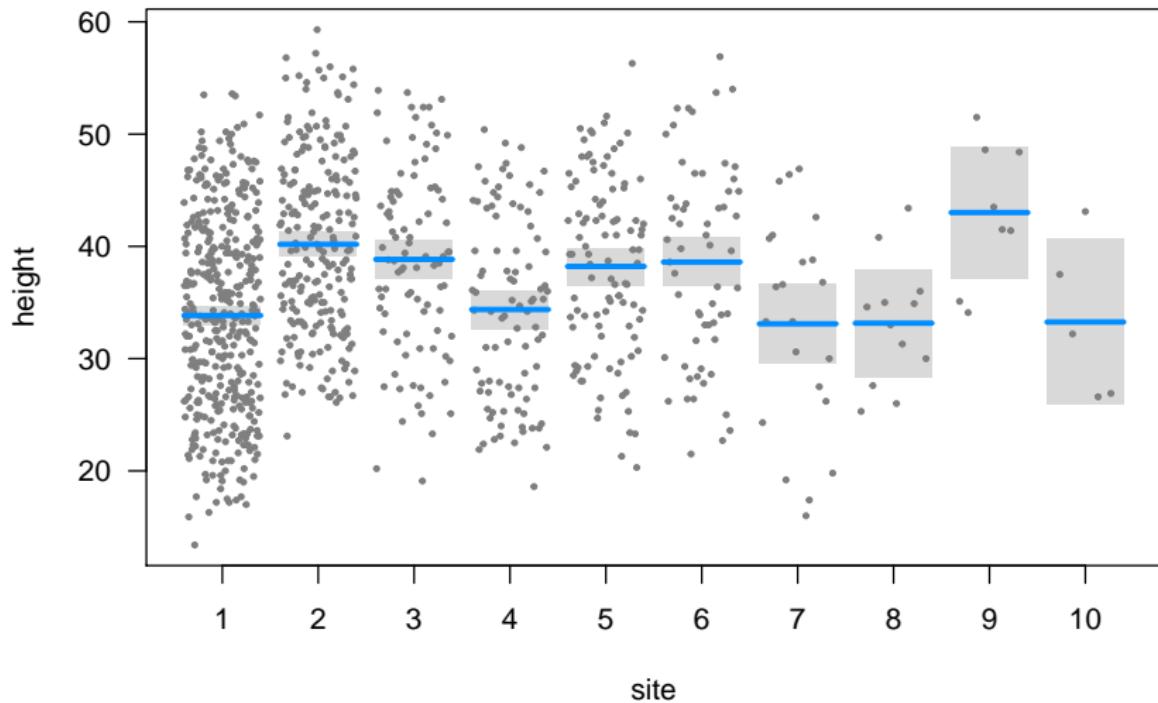
Plot

```
plot(allEffects(m3))
```



Plot (visreg)

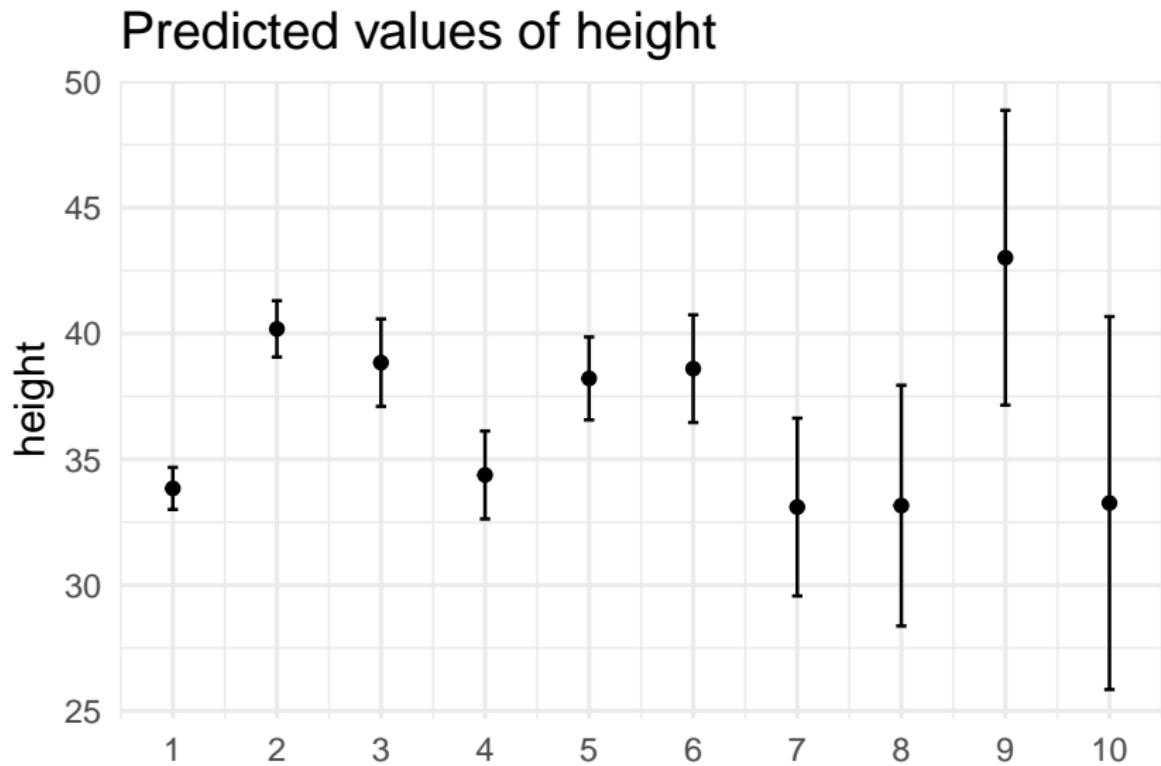
```
visreg(m3)
```



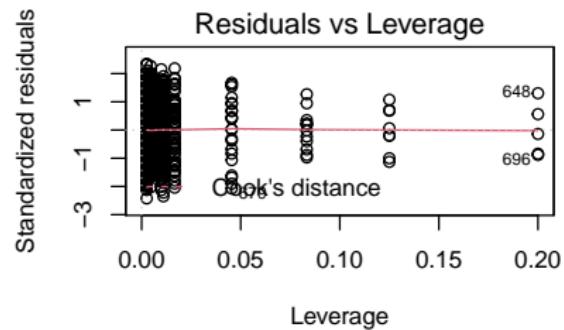
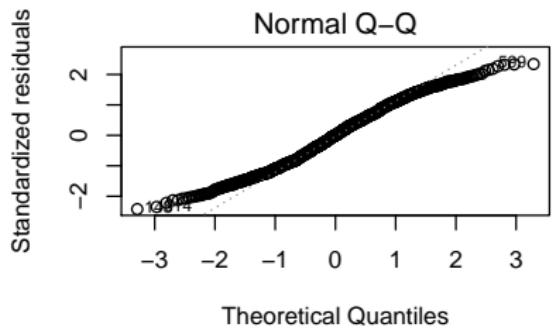
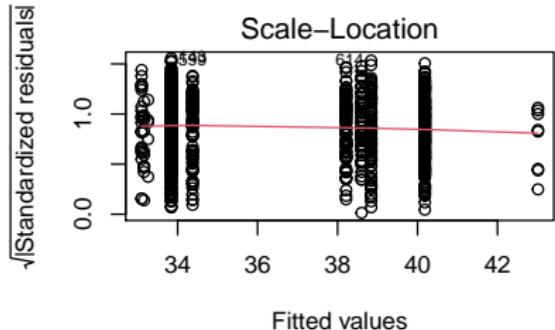
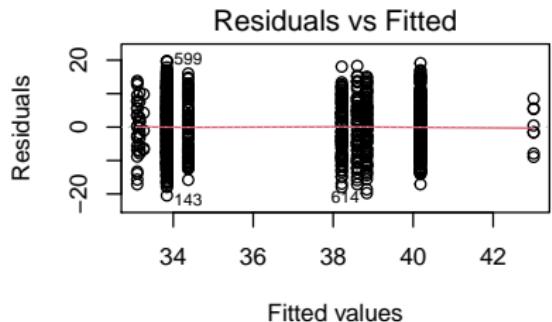
Plot model (sjPlot)

```
plot_model(m3, type = "eff")
```

\$site



Model checking: residuals



Combining continuous and categorical predictors

Predicting tree height based on dbh and site

```
lm(height ~ site + dbh, data = trees)
```

corresponds to

$$y_i = a + b_{site2} + c_{site3} + d_{site4} + e_{site5} + \dots + k \cdot DBH_i + \varepsilon_i$$
$$\varepsilon_i \sim N(0, \sigma^2)$$

Predicting tree height based on dbh and site

Call:

```
lm(formula = height ~ site + dbh, data = trees)
```

Residuals:

Min	1Q	Median	3Q	Max
-10.1130	-1.9885	0.0582	2.0314	11.3320

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	16.699037	0.260565	64.088	< 2e-16 ***
site2	6.504303	0.256730	25.335	< 2e-16 ***
site3	4.357457	0.354181	12.303	< 2e-16 ***
site4	1.934650	0.356102	5.433	6.98e-08 ***
site5	3.637432	0.339688	10.708	< 2e-16 ***
site6	4.204511	0.421906	9.966	< 2e-16 ***
site7	-0.176193	0.666772	-0.264	0.7916
site8	-5.312648	0.893603	-5.945	3.82e-09 ***
site9	5.437049	1.087766	4.998	6.84e-07 ***
site10	2.263338	1.369986	1.652	0.0988 .
dbh	0.617075	0.007574	81.473	< 2e-16 ***

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

Residual standard error: 3.043 on 989 degrees of freedom

Multiple R-squared: 0.8835, Adjusted R-squared: 0.8823

F-statistic: 750 on 10 and 989 DF, p-value: < 2.2e-16

Presenting model results

```
kable(xtable::xtable(m4), digits = 2)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	16.70	0.26	64.09	0.00
site2	6.50	0.26	25.34	0.00
site3	4.36	0.35	12.30	0.00
site4	1.93	0.36	5.43	0.00
site5	3.64	0.34	10.71	0.00
site6	4.20	0.42	9.97	0.00
site7	-0.18	0.67	-0.26	0.79
site8	-5.31	0.89	-5.95	0.00
site9	5.44	1.09	5.00	0.00
site10	2.26	1.37	1.65	0.10
dbh	0.62	0.01	81.47	0.00

Estimated tree heights for each site

```
summary(allEffects(m4))

model: height ~ site + dbh

site effect
site
  1      2      3      4      5      6      7      8
33.90437 40.40868 38.26183 35.83902 37.54181 38.10889 33.72818 28.59173
  9      10
39.34142 36.16771

Lower 95 Percent Confidence Limits
site
  1      2      3      4      5      6      7      8
33.60276 40.00512 37.63569 35.20858 36.94739 37.33787 32.45495 26.86438
  9      10
37.22831 33.49623

Upper 95 Percent Confidence Limits
site
  1      2      3      4      5      6      7      8
34.20599 40.81223 38.88798 36.46947 38.13622 38.87990 35.00141 30.31907
  9      10
41.45454 38.83919

dbh effect
dbh
  5      20      30      40      50
22.38634 31.64246 37.81321 43.98396 50.15471

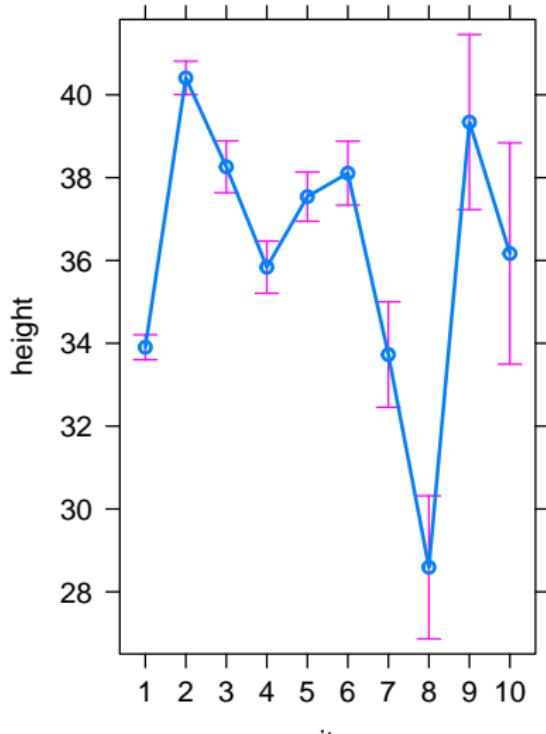
Lower 95 Percent Confidence Limits

```

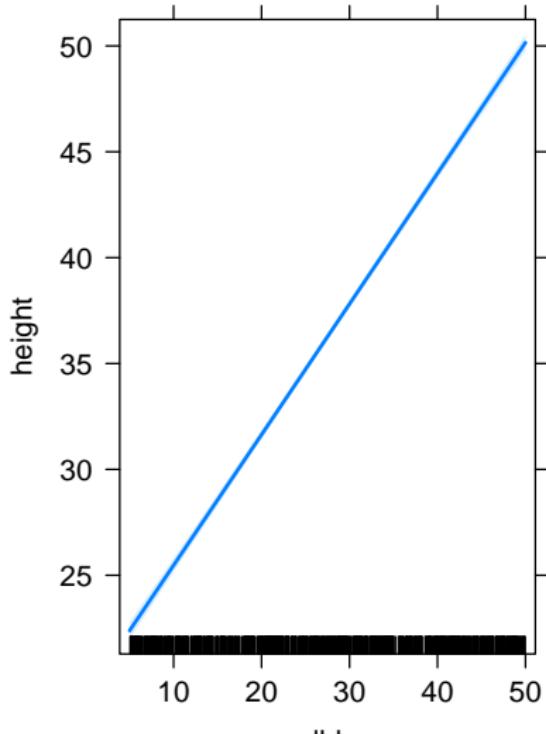
Plot

```
plot(allEffects(m4))
```

site effect plot

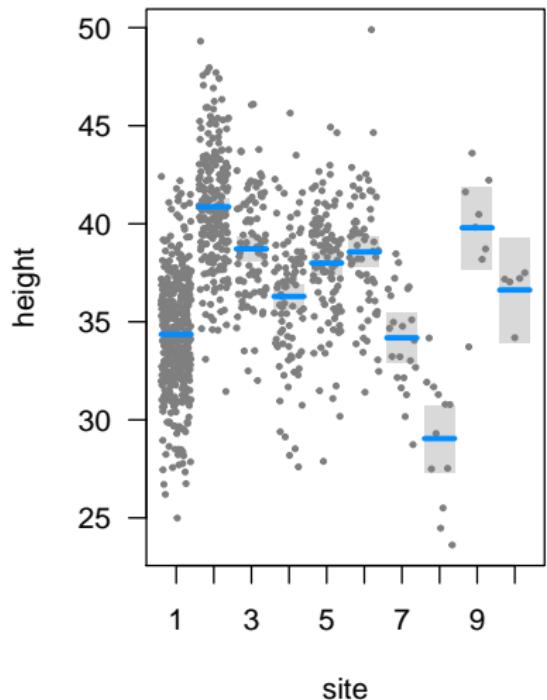


dbh effect plot



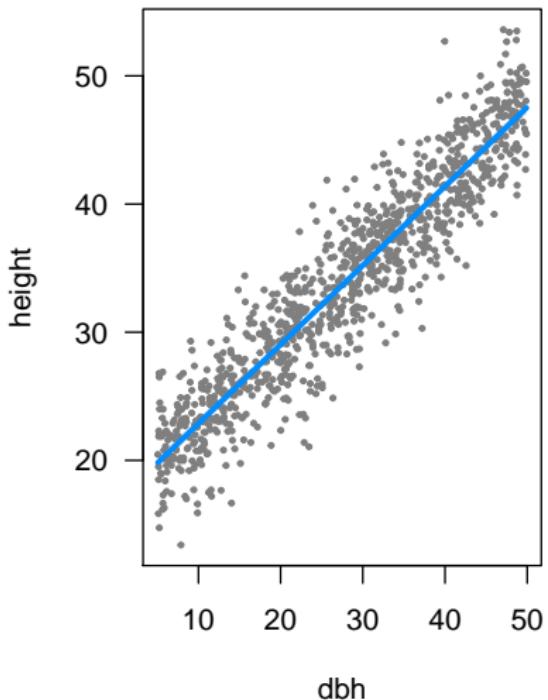
Plot (visreg)

```
visreg(m4)
```



null device

1

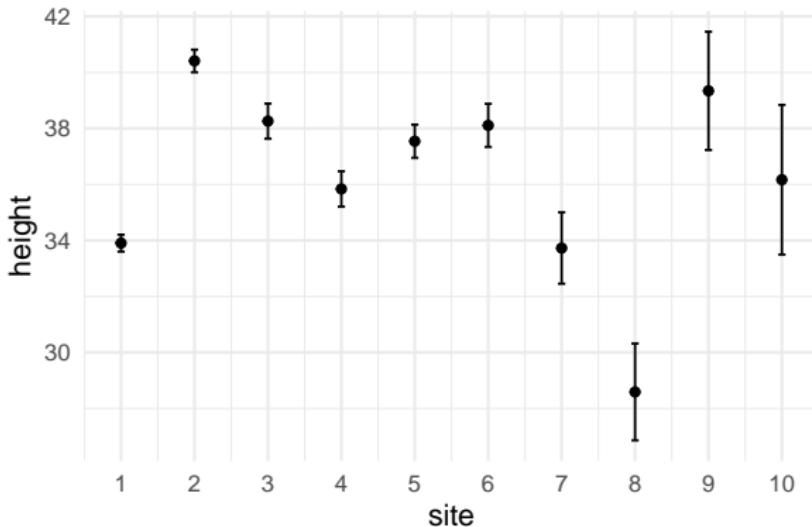


Plot model (sjPlot)

```
plot_model(m4, type = "eff")
```

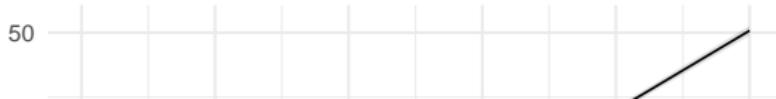
\$site

Predicted values of height



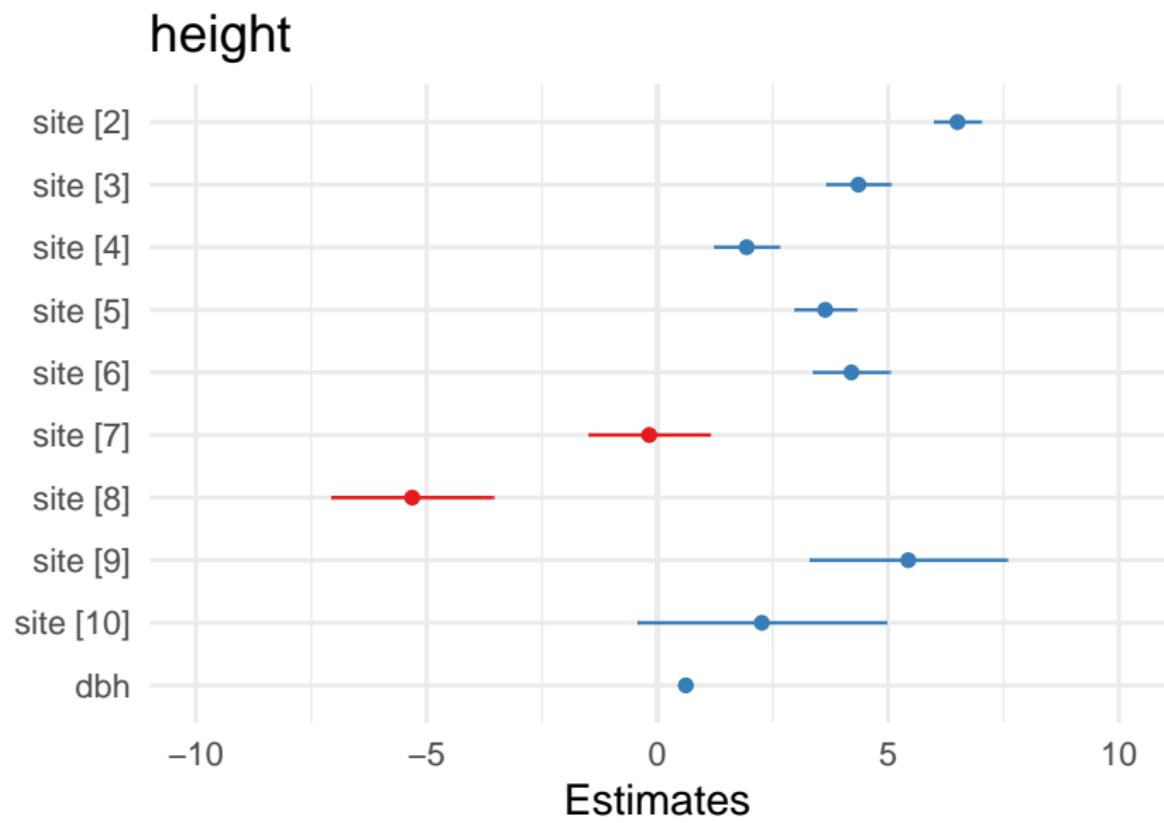
\$dbh

Predicted values of height

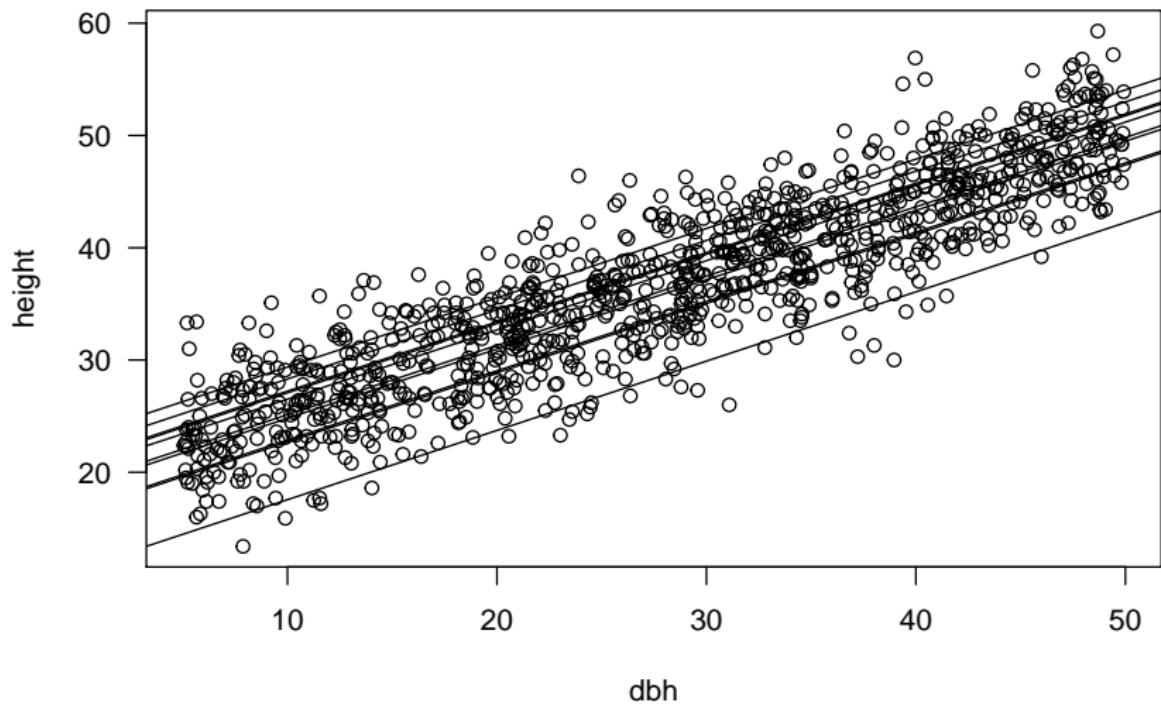


Plot model (sjPlot)

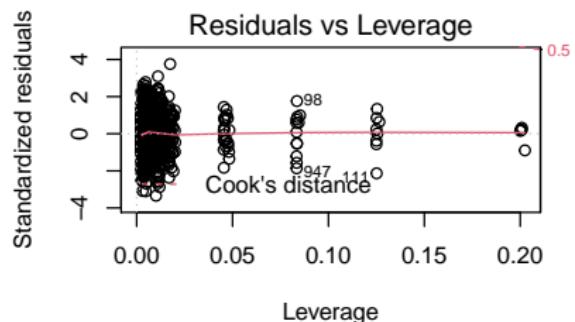
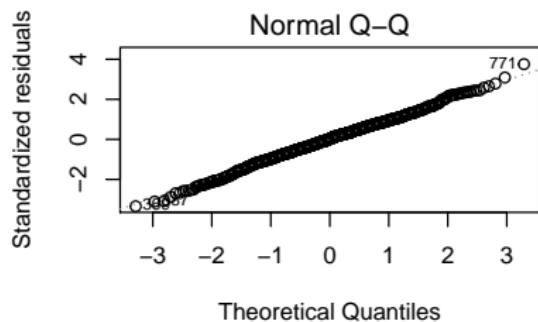
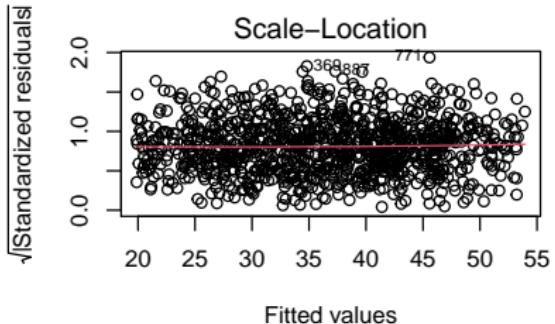
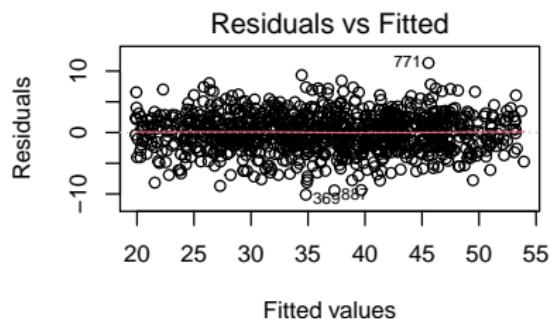
```
plot_model(m4, type = "est")
```



We have fitted model w/ many intercepts and single slope

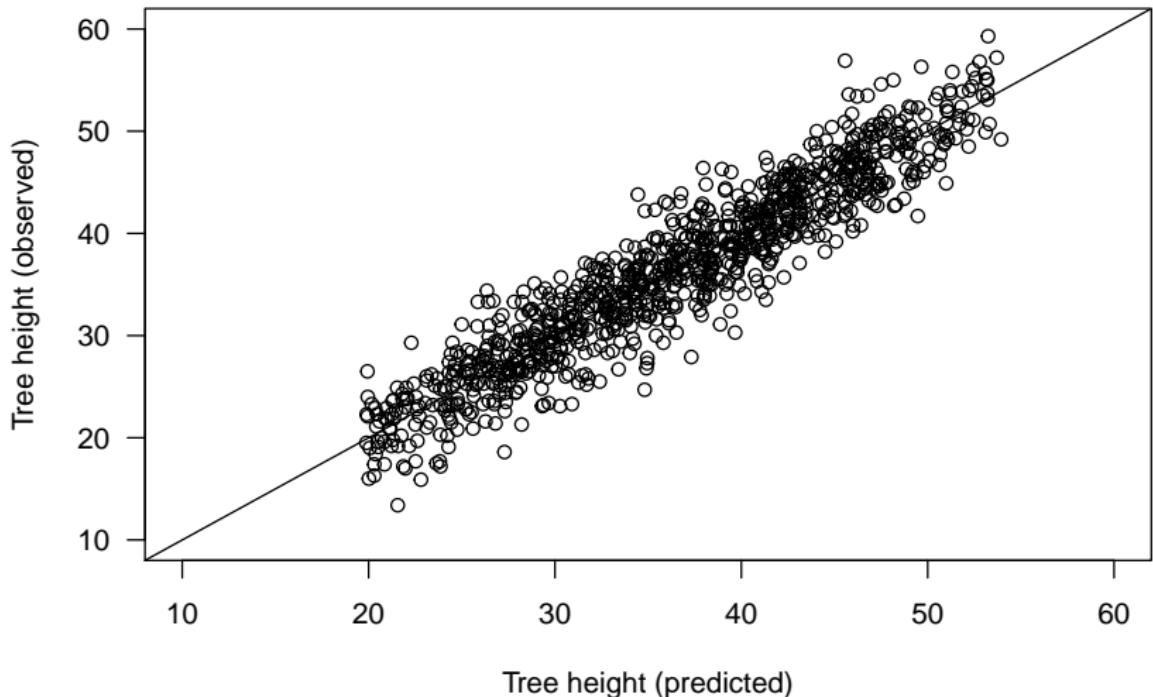


Model checking: residuals



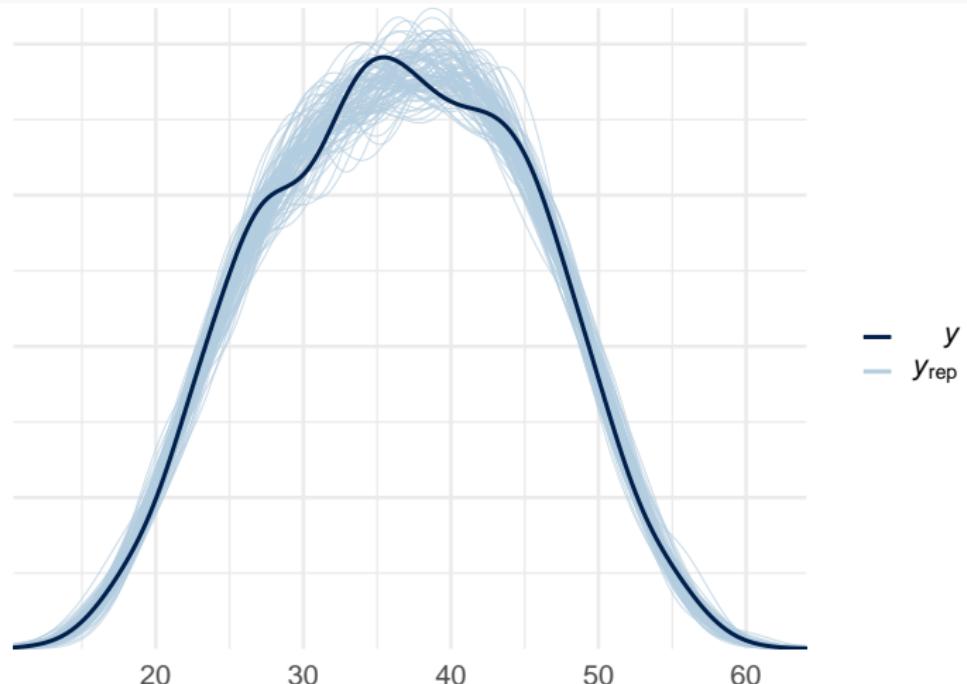
How good is this model? Calibration plot

```
trees$height.pred <- fitted(m4)
plot(trees$height.pred, trees$height, xlab = "Tree height (predicted)", ylab = "Tree height (observed)", main = "Calibration plot")
abline(a = 0, b = 1)
```



Model checking with simulated data

```
library(bayesplot)
sims <- simulate(m4, nsim = 100)
ppc_dens_overlay(trees$height, yrep = t(as.matrix(sims)))
```



Q: Does allometric relationship between DBH
and Height vary among sites?

Model with interactions

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

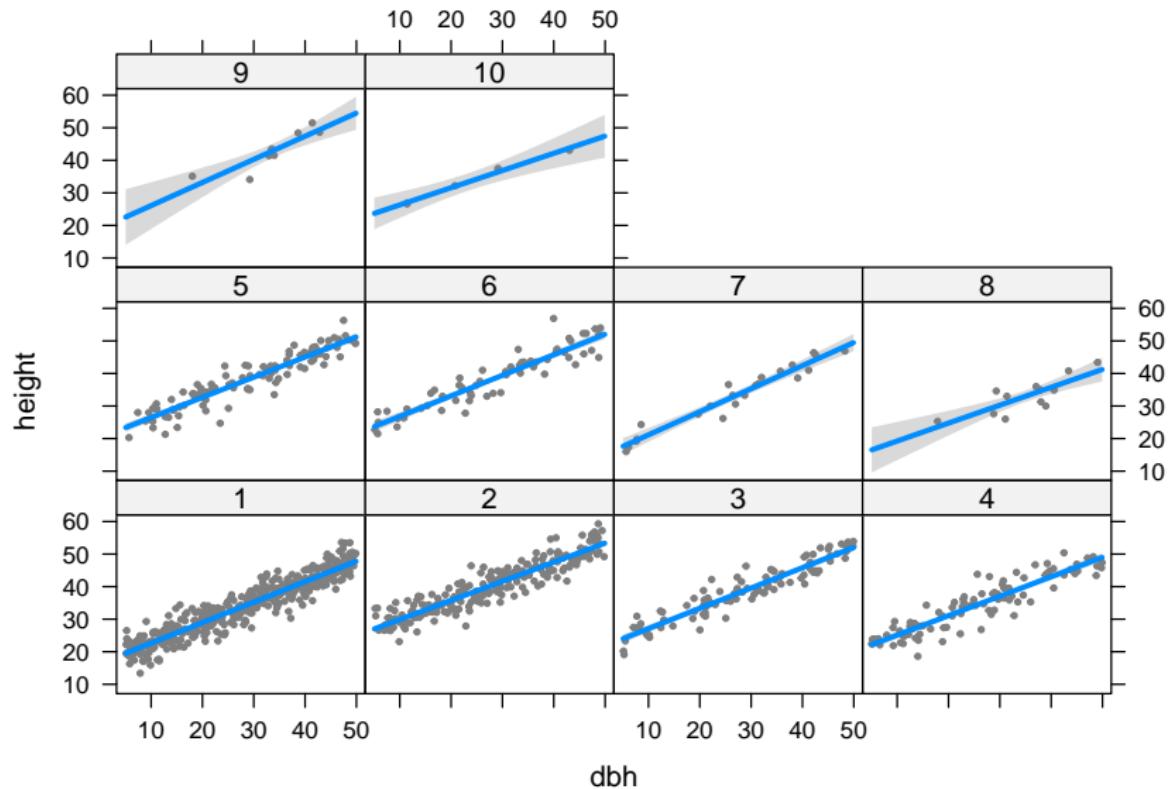
Residuals:
    Min      1Q  Median      3Q     Max
-10.1017 -1.9839  0.0645  2.0486 11.1789

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 16.359437  0.360054 45.436 < 2e-16 ***
site2        7.684781  0.609657 12.605 < 2e-16 ***
site3        4.518568  0.867008  5.212 2.28e-07 ***
site4        2.769336  0.813259  3.405 0.000688 ***
site5        3.917607  0.870983  4.498 7.68e-06 ***
site6        4.155161  1.009379  4.117 4.17e-05 ***
site7       -2.306799  1.551303 -1.487 0.137334
site8       -2.616095  4.090671 -0.640 0.522630
site9        2.621560  5.073794  0.517 0.605492
site10       4.662340  2.991072  1.559 0.119378
dbh          0.629299  0.011722 53.685 < 2e-16 ***
site2:dbh   -0.042784  0.020033 -2.136 0.032950 *
site3:dbh   -0.006031  0.027640 -0.218 0.827312
site4:dbh   -0.031633  0.028225 -1.121 0.262677
site5:dbh   -0.010173  0.027887 -0.365 0.715334
site6:dbh    0.001337  0.032109  0.042 0.966797
site7:dbh    0.079728  0.052056  1.532 0.125951
site8:dbh   -0.079027  0.113386 -0.697 0.485984
site9:dbh    0.081035  0.146649  0.553 0.580679
site10:dbh   -0.101107  0.114520 -0.883 0.377522
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.041 on 980 degrees of freedom
Multiple R-squared:  0.8847,    Adjusted R-squared:  0.8825
F-statistic: 395.7 on 19 and 980 DF,  p-value: < 2.2e-16
```

Does slope vary among forests?

```
visreg(m5, xvar = "dbh", by = "site")
```



Extra exercises

- ▶ [paperplanes](#): How does flight distance differ with age, gender or paper type?

Extra exercises

- ▶ [paperplanes](#): How does flight distance differ with age, gender or paper type?
- ▶ [mammal sleep](#): Are sleep patterns related to diet?

Extra exercises

- ▶ [paperplanes](#): How does flight distance differ with age, gender or paper type?
- ▶ [mammal sleep](#): Are sleep patterns related to diet?
- ▶ [iris](#): Predict petal length ~ petal width and species

Extra exercises

- ▶ [paperplanes](#): How does flight distance differ with age, gender or paper type?
- ▶ [mammal sleep](#): Are sleep patterns related to diet?
- ▶ [iris](#): Predict petal length ~ petal width and species
- ▶ [Penguins data](#): Body mass ~ Flipper length, Bill length ~ Bill depth, differences across sites...

Extra exercises

- ▶ [paperplanes](#): How does flight distance differ with age, gender or paper type?
- ▶ [mammal sleep](#): Are sleep patterns related to diet?
- ▶ [iris](#): Predict petal length ~ petal width and species
- ▶ [Penguins data](#): Body mass ~ Flipper length, Bill length ~ Bill depth, differences across sites...
- ▶ [racing pigeons](#): is speed related to sex?

Generalised Linear Models: Logistic regression

Q: Survival of passengers on the Titanic ~ Class

Read `titanic_long.csv` dataset and fit linear model (`survival ~ class`).

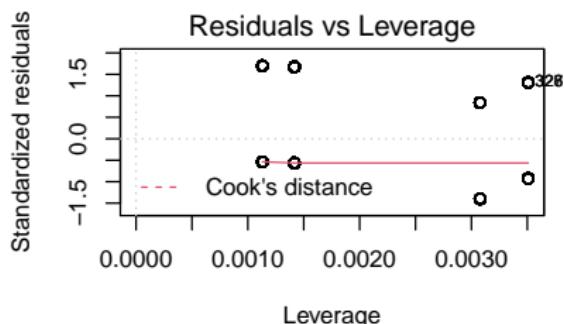
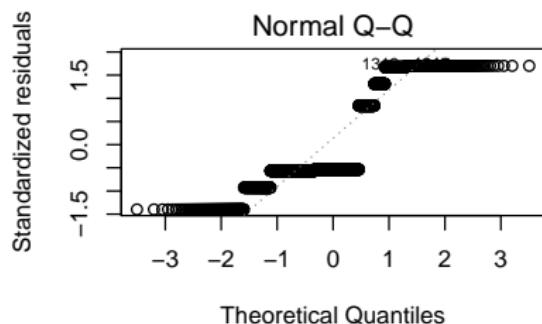
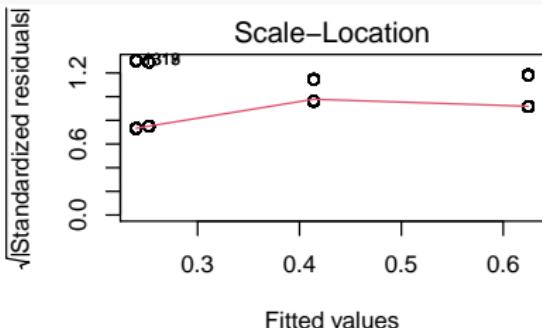
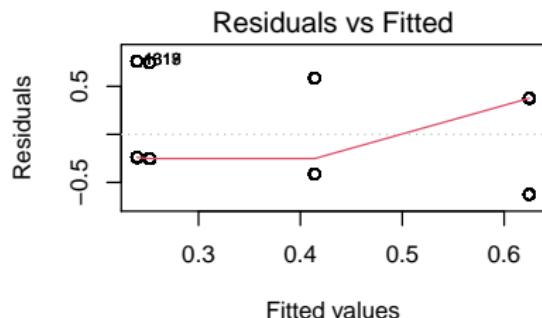
	class	age	sex	survived
1	first	adult	male	1
2	first	adult	male	1
3	first	adult	male	1
4	first	adult	male	1
5	first	adult	male	1
6	first	adult	male	1

Quiz

<https://pollev.com/franciscorod726>

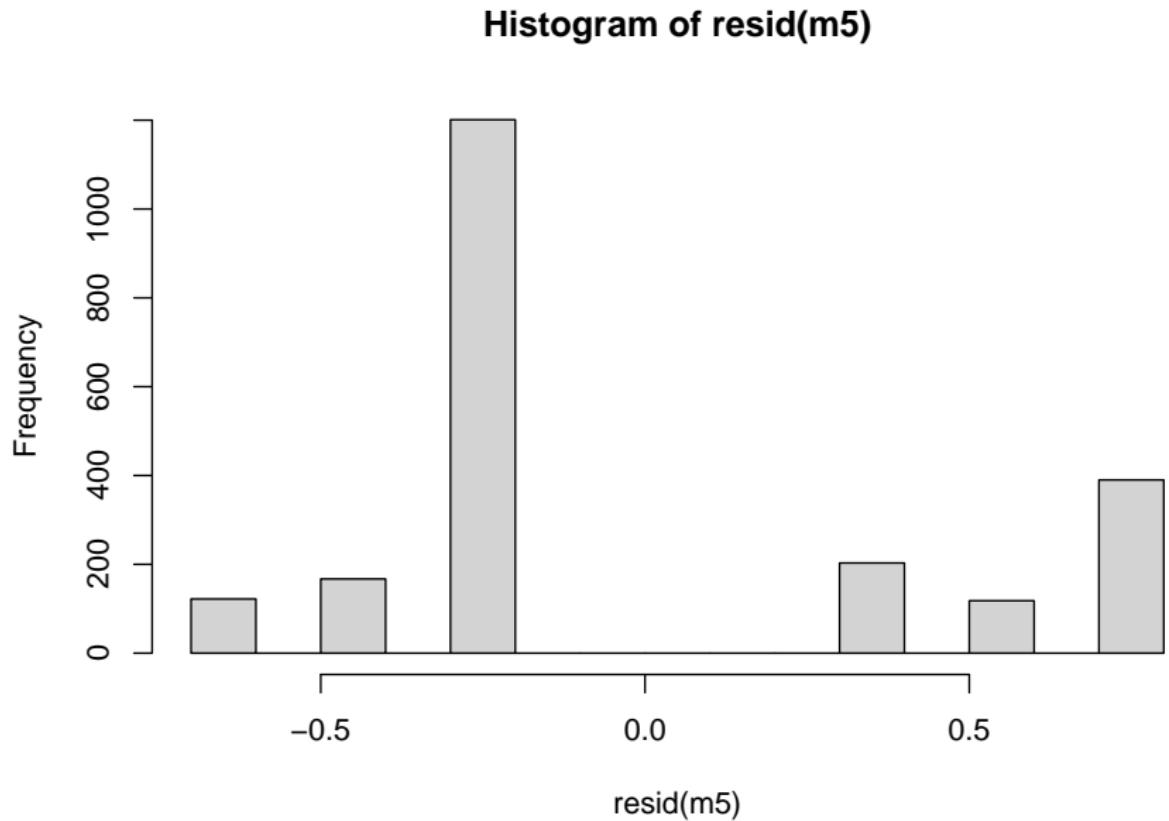
Let's check linear model:

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



```
null device
```

Weird residuals!



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

Binary variables (0/1)

Counts (0, 1, 2, 3, ...)

Generalised Linear Models to the rescue!

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)

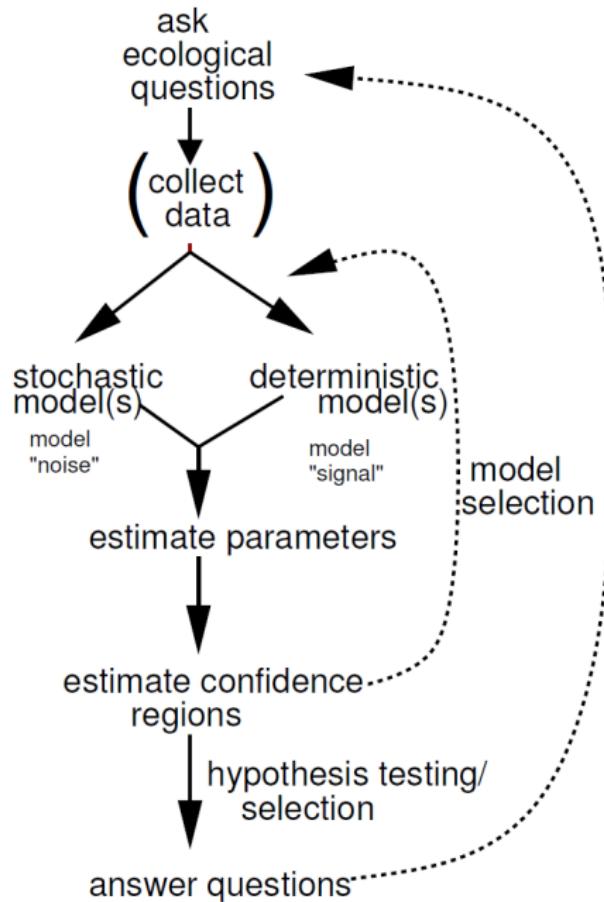
3. Link function

- ▶ 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



Bernoulli - Binomial distribution (Logistic regression)

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

Link function: logit (others possible, see `family`)

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

Then

$$\Pr(\text{alive}) = a + bx$$

$$\text{logit}(\Pr(\text{alive})) = a + bx$$

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

Back to survival of Titanic passengers

How many survived in each class?

```
table(titanic$class, titanic$survived)
```

	0	1
crew	673	212
first	122	203
second	167	118
third	528	178

Back to survival of Titanic passengers (dplyr)

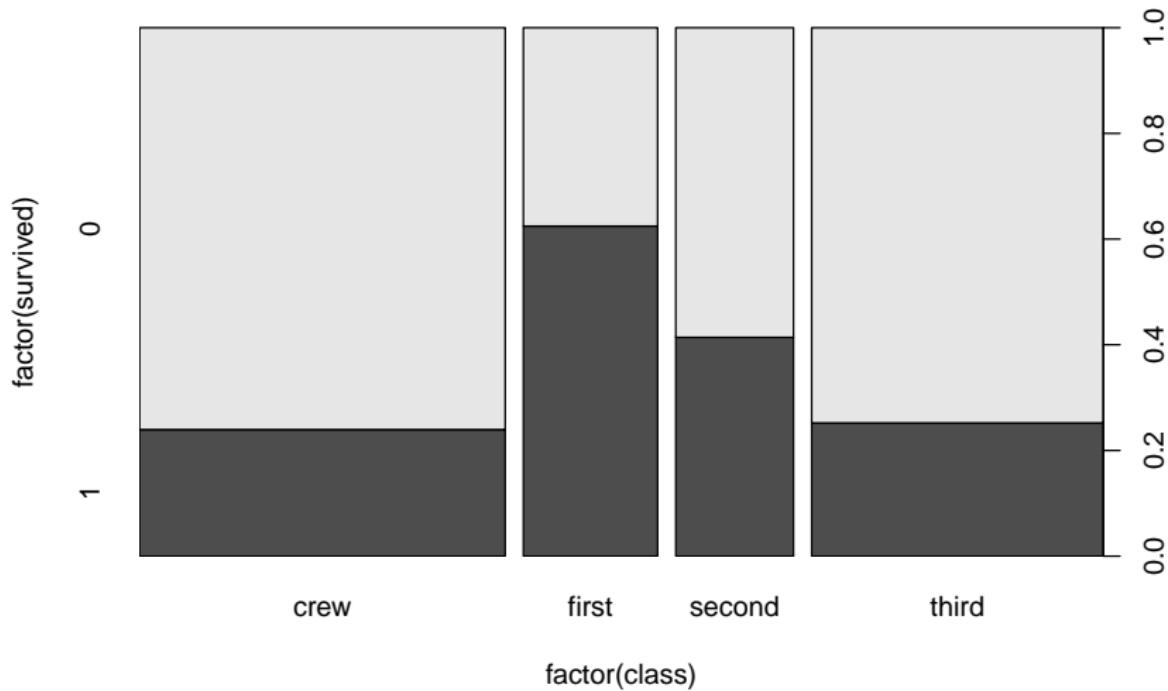
Passenger survival according to class

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

```
# A tibble: 8 x 3
# Groups:   class [4]
  class  survived  count
  <chr>    <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
```

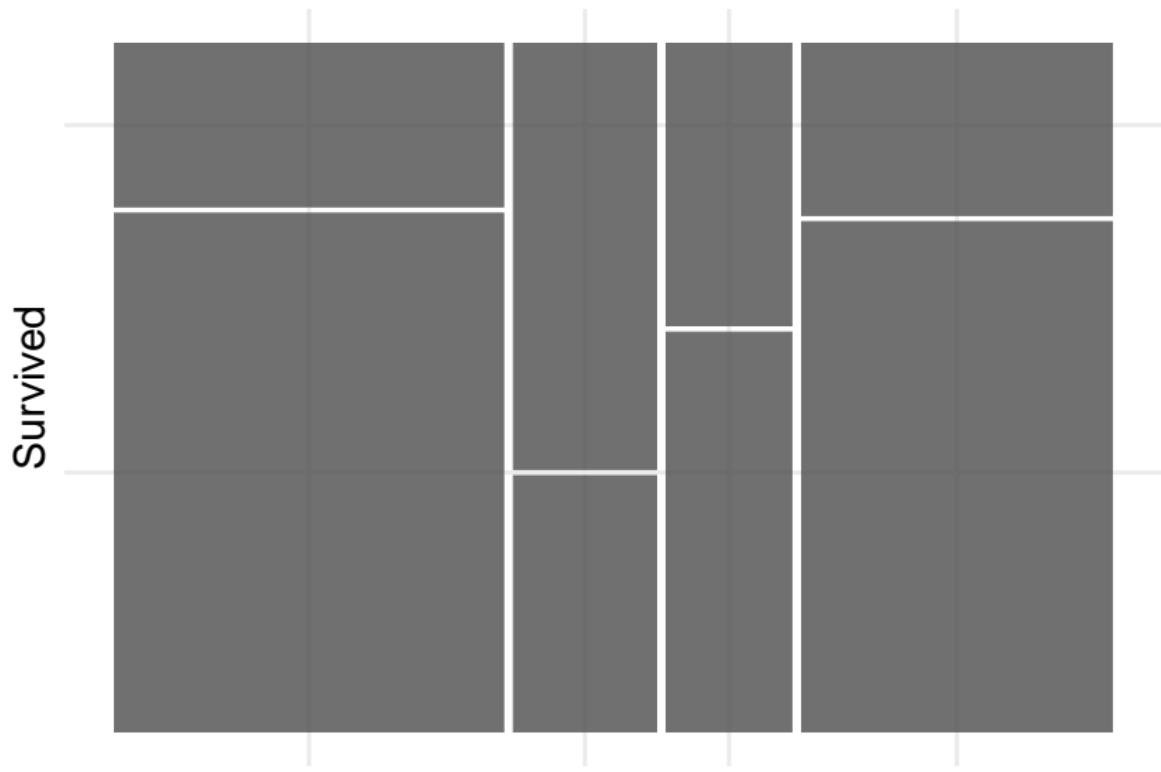
Or graphically...

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



Mosaic plots (ggplot2)

```
ggplot(titanic) +  
  geom_mosaic(aes(x = product(survived, class))) +  
  labs(x = "", y = "Survived")
```



Fitting GLMs in R: `glm`

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

which corresponds to

$$\text{logit}(\Pr(\text{survival})_i) = a + b \cdot \text{class}_i$$

$$\text{logit}(\Pr(\text{survival})_i) = a + b_{\text{first}} + c_{\text{second}} + d_{\text{third}}$$

Fitting GLMs in R: `glm`

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

Call:

```
glm(formula = survived ~ class, family = binomial, 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?

Must add intercept (baseline) to the parameter estimate:

```
plogis(coef(tit.glm)[1] + coef(tit.glm)[2])  
(Intercept)  
0.6246154
```

Again this value matches the data:

```
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
```

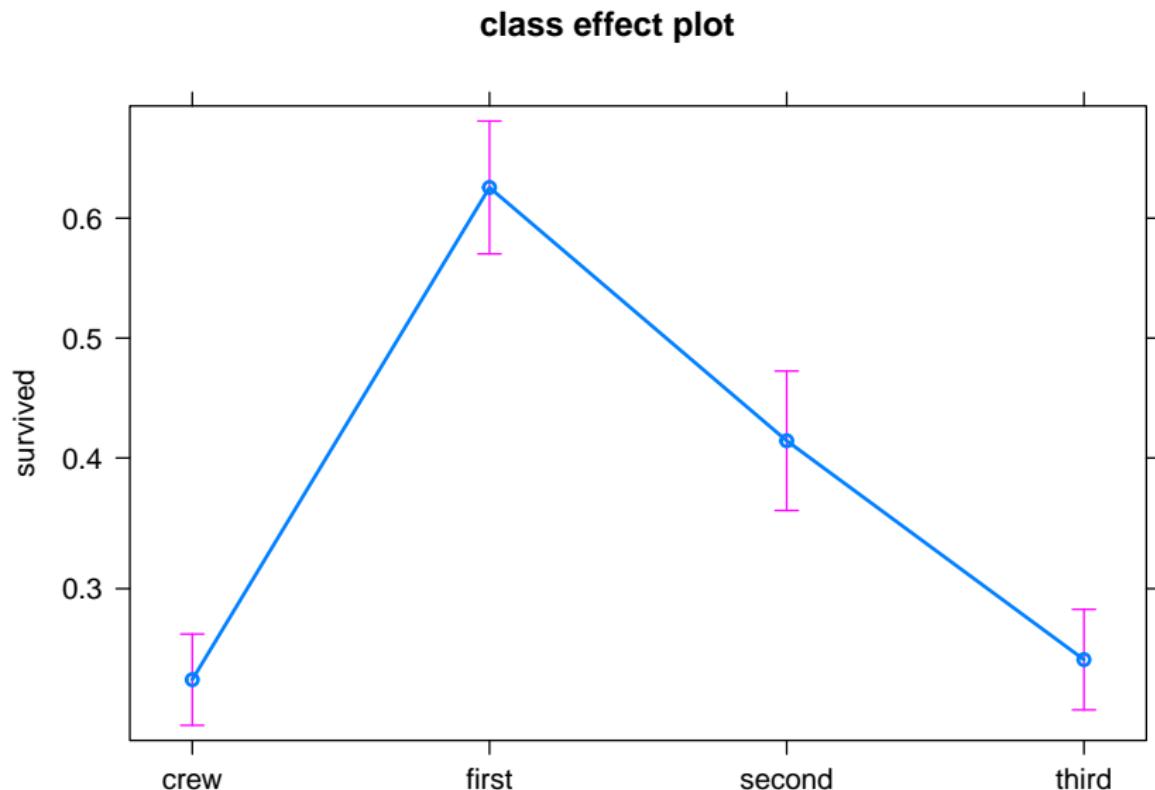
Presenting model results

```
kable(xtable::xtable(tit.glm), digits = 2)
```

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.16	0.08	-14.67	0.00
classfirst	1.66	0.14	11.97	0.00
classsecond	0.81	0.14	5.62	0.00
classthird	0.07	0.12	0.58	0.56

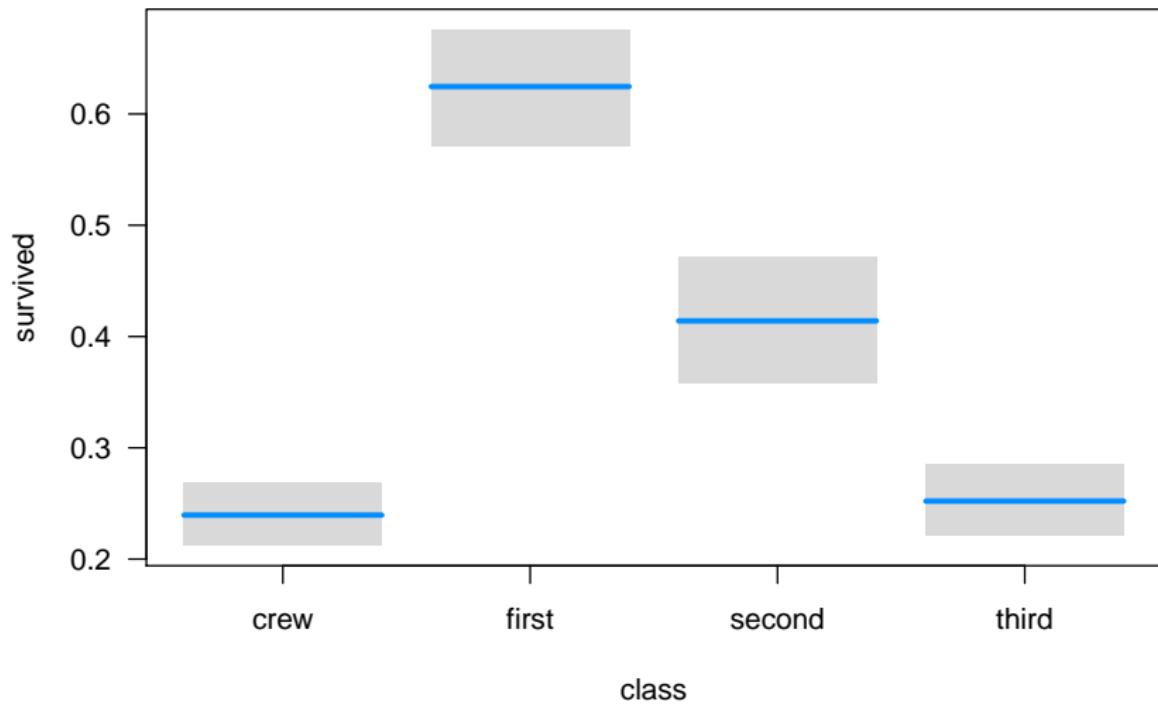
Visualising model: effects package

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



Visualising model: visreg package

```
visreg(tit.glm, scale = "response", rug = FALSE)
```

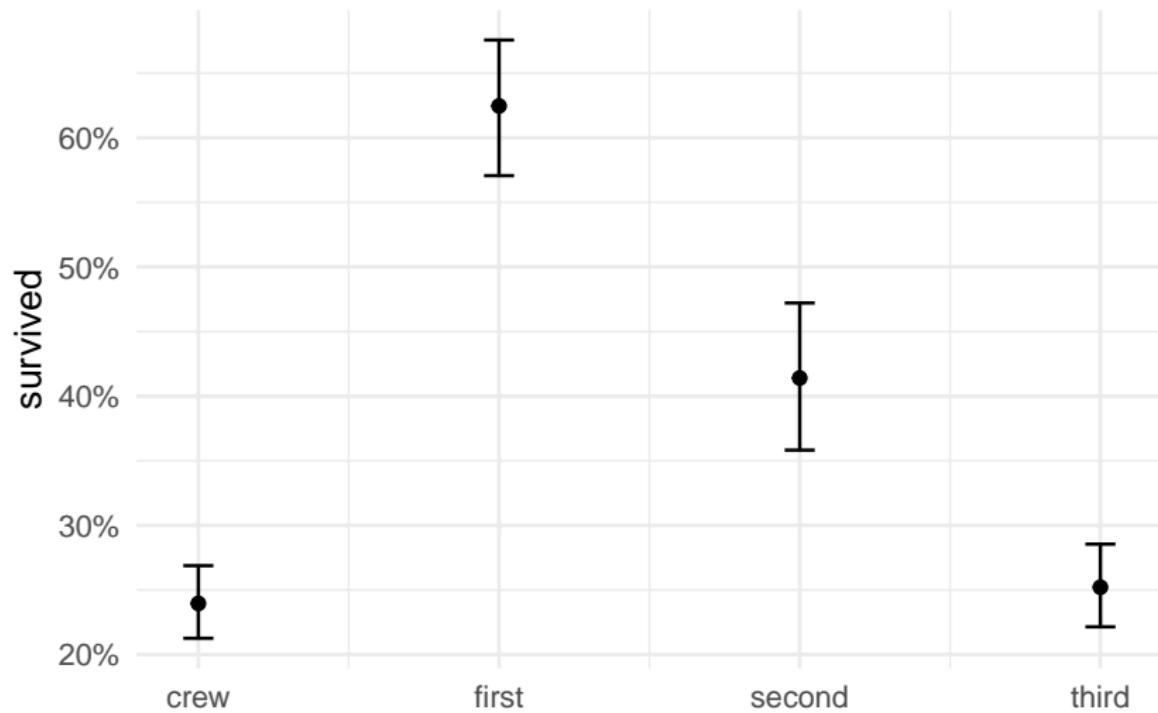


Visualising model: sjPlot package

```
sjPlot::plot_model(tit.glm, type = "eff")
```

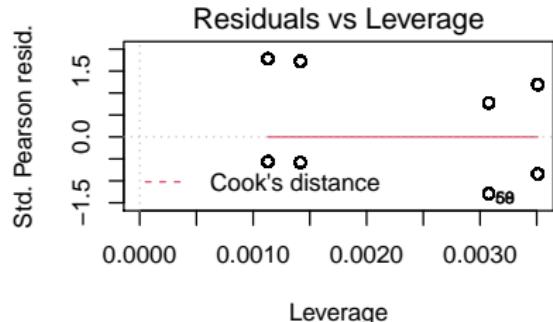
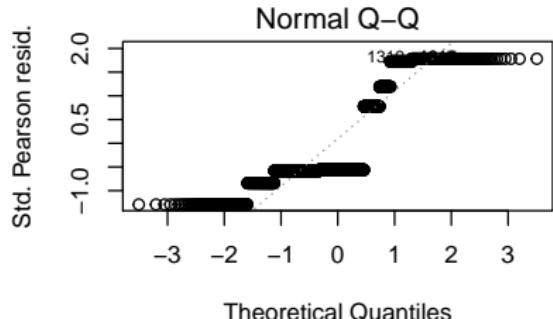
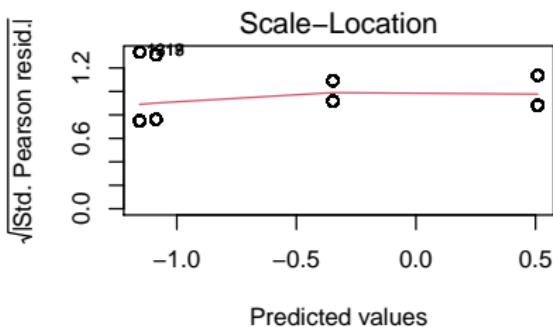
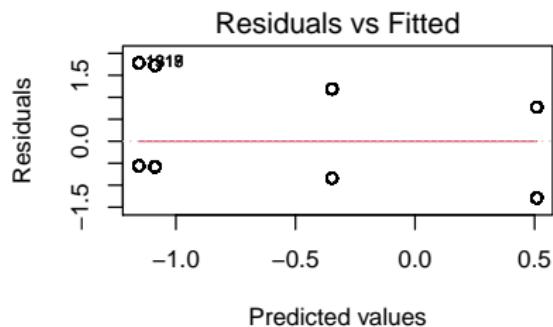
```
$class
```

Predicted probabilities of survived



Logistic regression: model checking

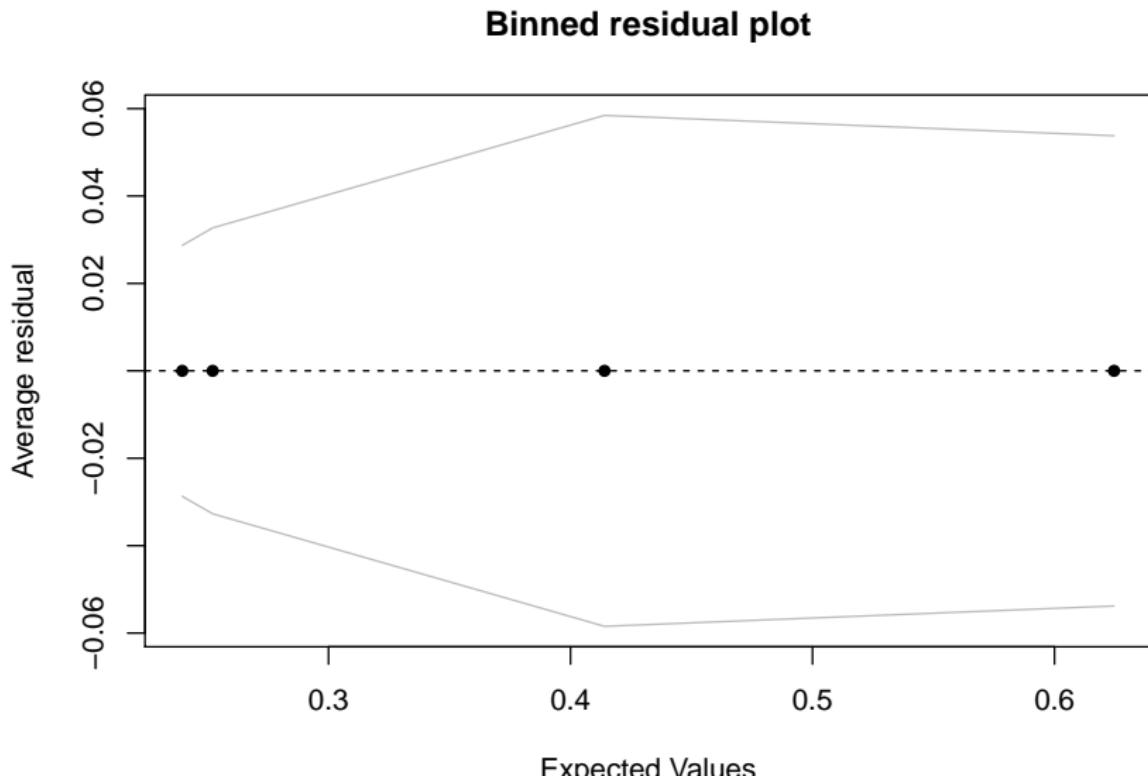
Not very useful



null device

Binned residual plots for logistic regression

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

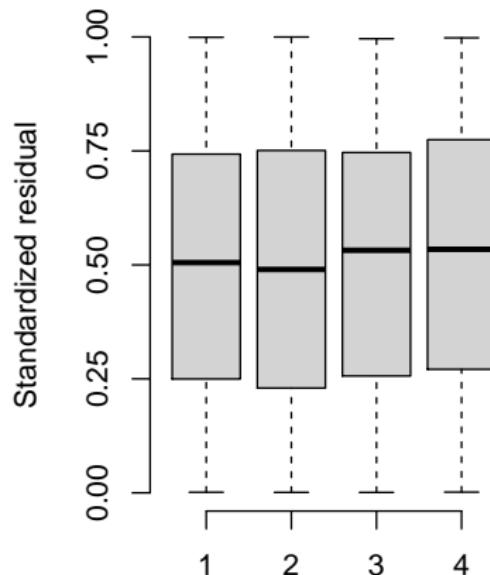
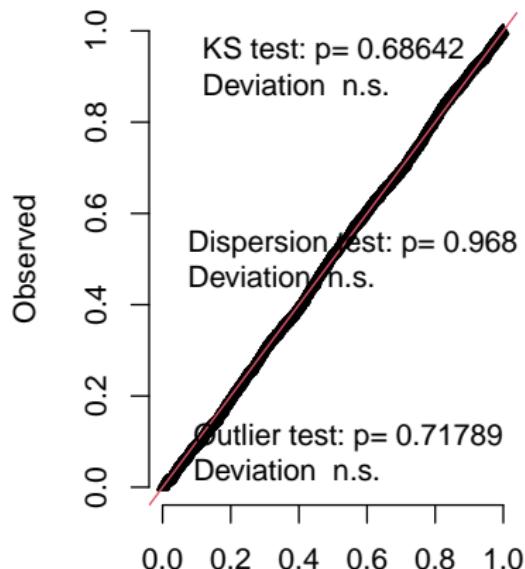


Residual diagnostics with DHARMA

```
library(DHARMA)  
simulateResiduals(tit.glm, plot = TRUE)
```

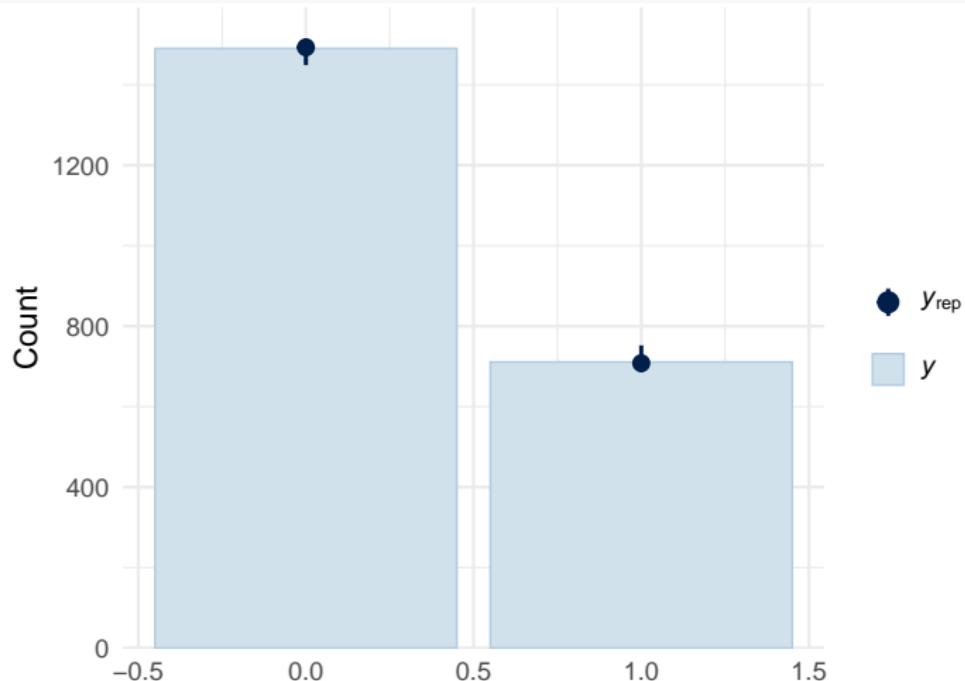
DHARMA residual diagnostics

QQ plot residuals



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(performance)
r2(tit.glm)
```

```
$R2_Tjur
Tjur's R2
0.08650663
```

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

Recapitulating

1. Visualise data

Recapitulating

1. **Visualise data**
2. **Fit model**: `glm`. Don't forget to specify `family`!

Recapitulating

1. **Visualise data**
2. **Fit model**: `glm`. Don't forget to specify `family`!
3. **Examine model**: `summary`

Recapitulating

1. **Visualise data**
2. **Fit model:** `glm`. Don't forget to specify `family`!
3. **Examine model:** `summary`
4. **Back-transform parameters** from *logit* into probability scale
(e.g. `allEffects`)

Recapitulating

1. **Visualise data**
2. **Fit model:** `glm`. Don't forget to specify `family`!
3. **Examine model:** `summary`
4. **Back-transform parameters** from *logit* into probability scale
(e.g. `allEffects`)
5. **Plot model:** `plot(allEffects(model))`, `visreg`, `plot_model...`

Recapitulating

1. **Visualise data**
2. **Fit model:** `glm`. Don't forget to specify `family`!
3. **Examine model:** `summary`
4. **Back-transform parameters** from *logit* into probability scale
(e.g. `allEffects`)
5. **Plot model:** `plot(allEffects(model))`, `visreg`, `plot_model...`
6. **Examine residuals:** `DHARMA::simulateResiduals`.

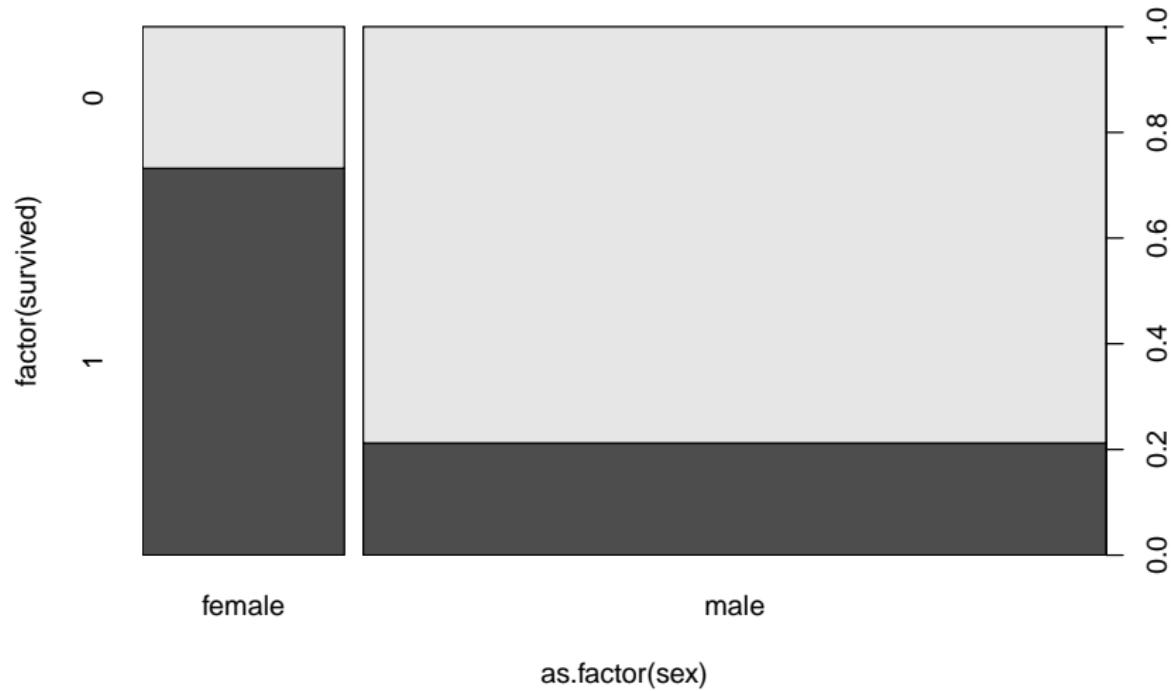
Q: Did men have higher survival than women?

Quiz

<https://pollev.com/franciscorod726>

Plot first

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



Fit model

Call:

```
glm(formula = survived ~ sex, family = binomial, 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
AIC: 2339

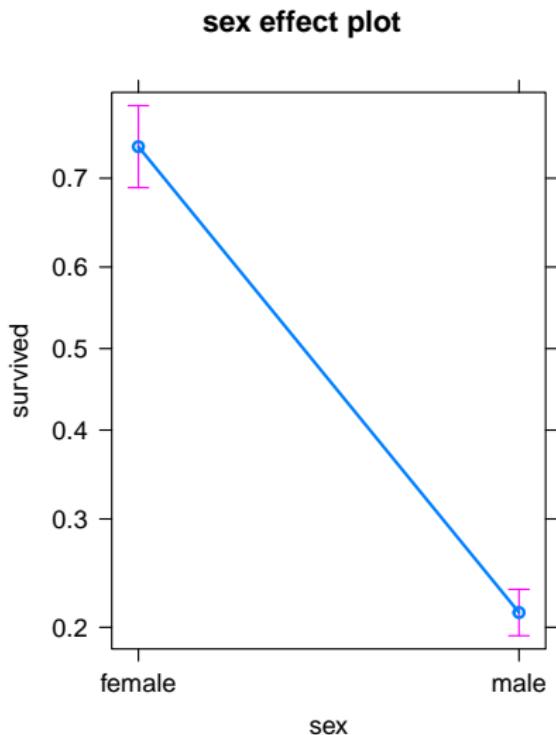
Effects

```
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

```
table(titanic$class, titanic$survived, titanic$sex)
```

```
, , = female
```

	0	1
crew	3	20
first	4	141
second	13	93
third	106	90

```
, , = male
```

	0	1
crew	670	192
first	118	62
second	154	25
third	422	88

Mmmm...

Quiz

<https://pollev.com/franciscorod726>

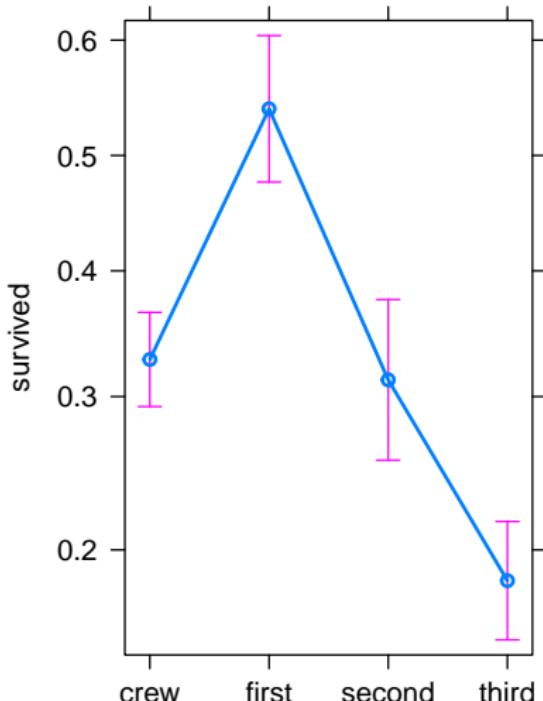
Fit additive model with both factors

```
tit.sex.class <- glm(survived ~ class + sex, family = binomial,  
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
```

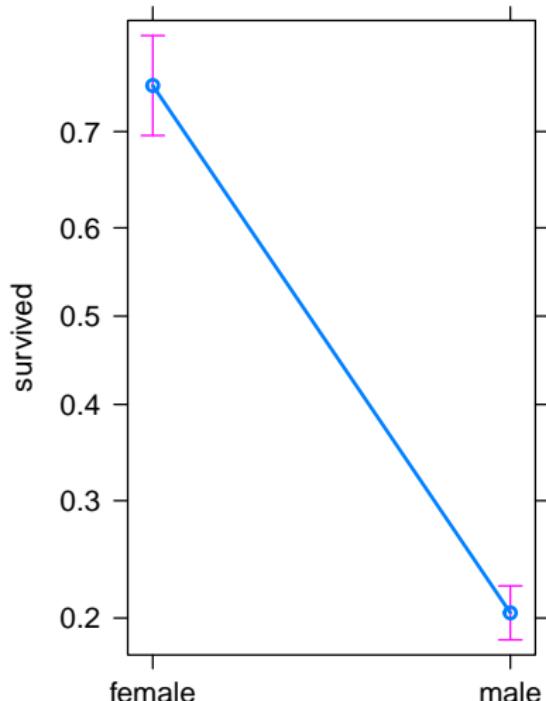
Plot additive model

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

class effect plot



sex effect plot



Fit model with both factors (interactions)

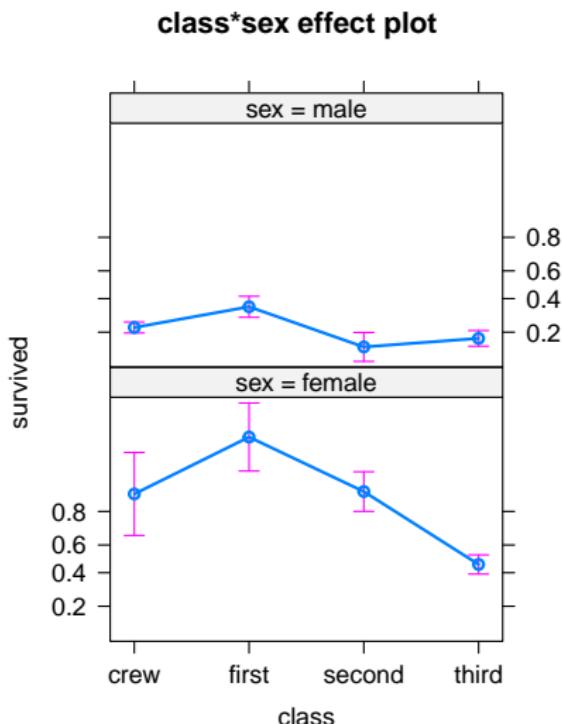
```
tit.sex.class <- glm(survived ~ class * sex, family = binomial,  
glm(formula = survived ~ class * sex, family = binomial, data =  
      coef.est coef.se  
(Intercept)      1.90      0.62  
classfirst       1.67      0.80  
classsecond      0.07      0.69  
classthird      -2.06      0.64  
sexmale          -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

```
model: survived ~ class * sex
```

class*sex effect

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

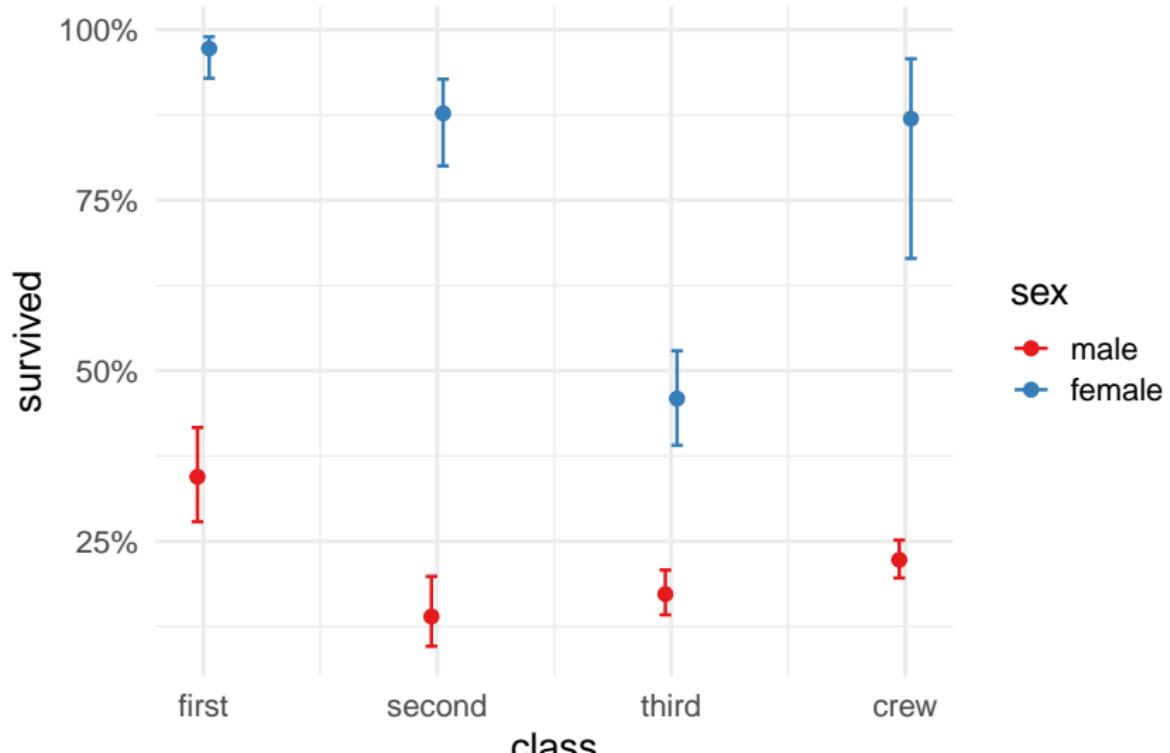


So, **women had higher probability of survival than men, even within the same class.**

Effects (sjPlot)

```
plot_model(tit.sex.class, type = "int")
```

Predicted probabilities of survived



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`.

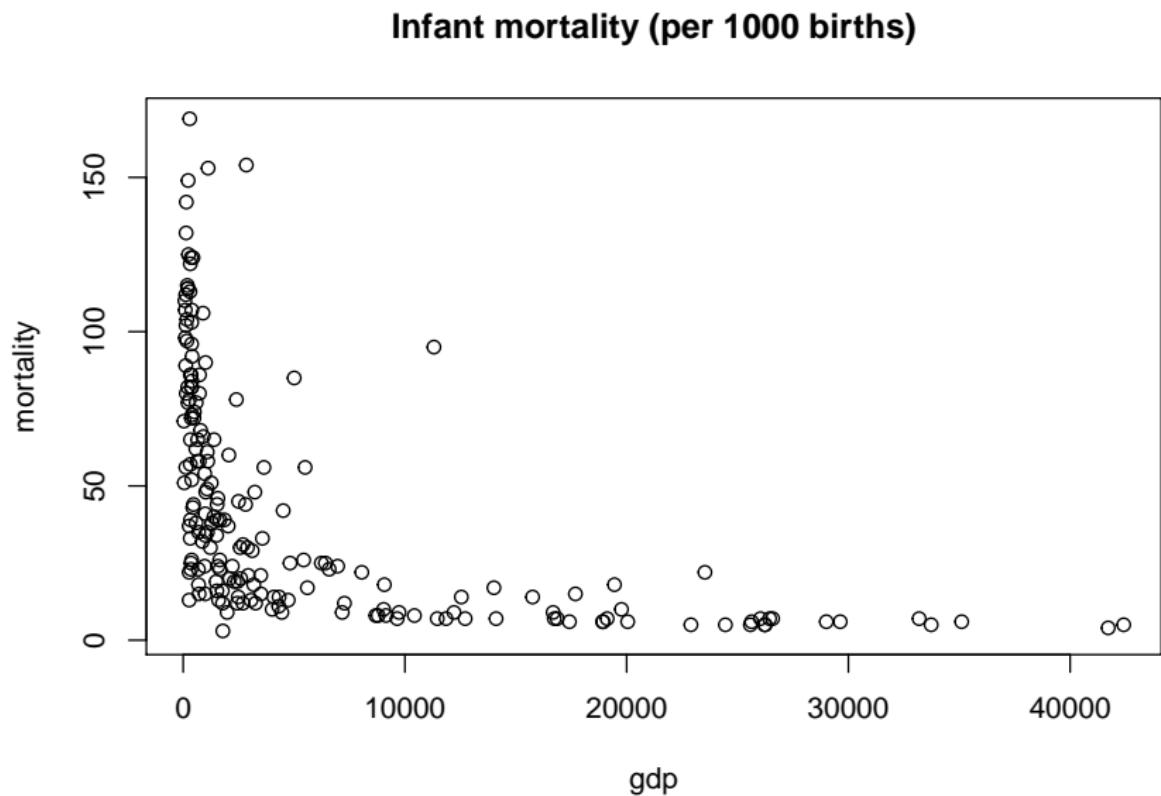
	country	mortality	gdp
Length:	207	Min. : 2.00	Min. : 36
Class :	character	1st Qu.: 12.00	1st Qu.: 442
Mode :	character	Median : 30.00	Median : 1779
		Mean : 43.48	Mean : 6262
		3rd Qu.: 66.00	3rd Qu.: 7272
		Max. : 169.00	Max. : 42416
		NA's : 6	NA's : 10

Q: Is infant mortality related to GDP?

<https://pollev.com/franciscorod726>

EDA

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



Fit model

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

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
```

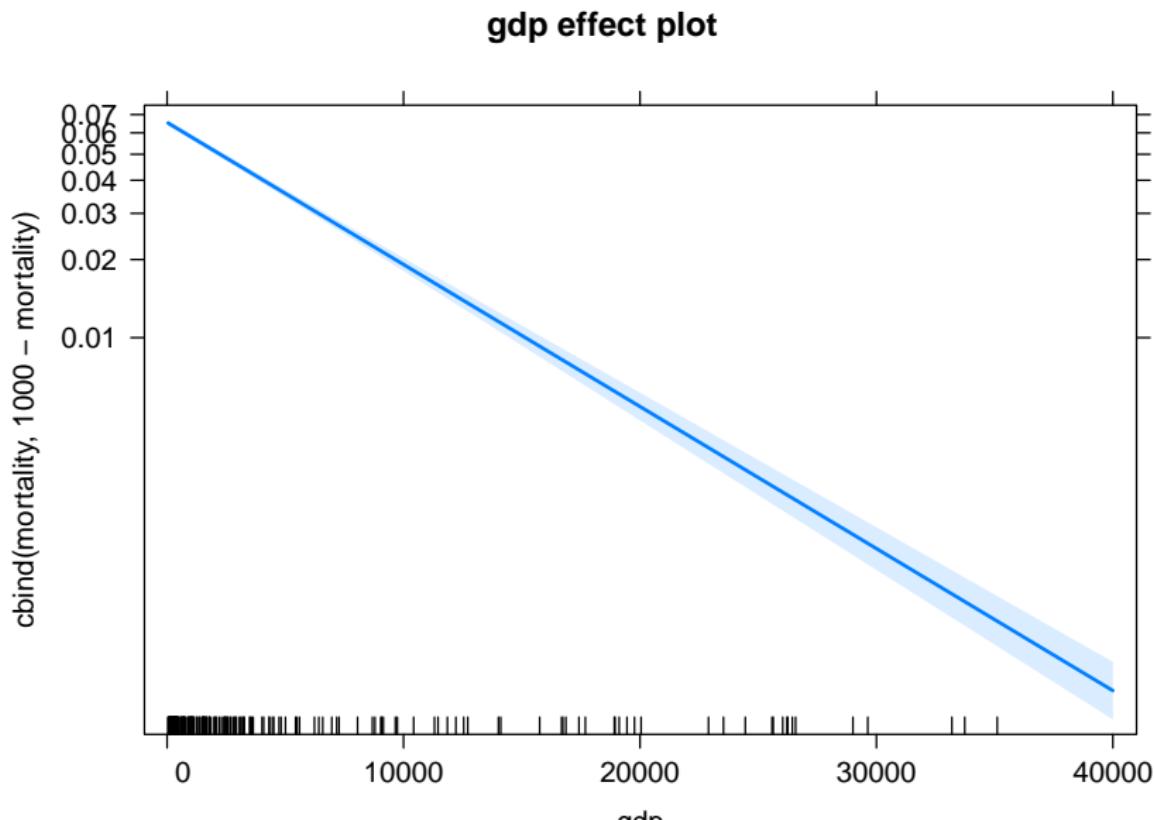
```
gdp effect
```

```
gdp
```

	40	10000	20000	30000	40000
0.0652177296	0.0191438829	0.0054028095	0.0015096074	0.0004206154	

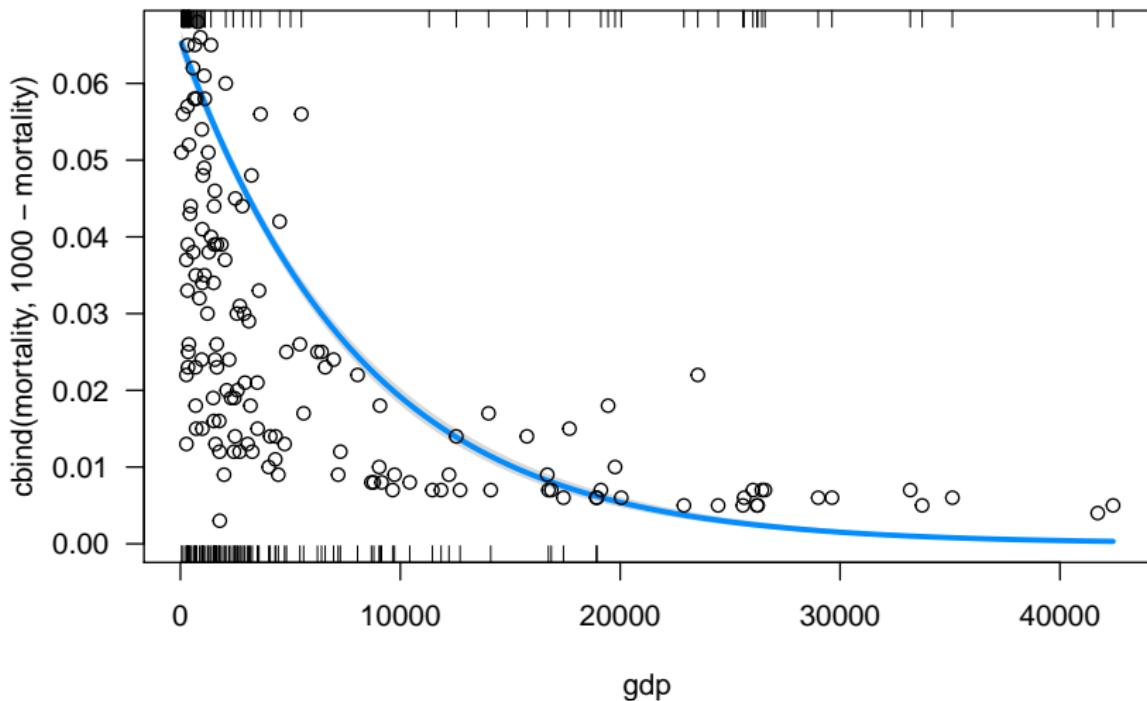
Effects plot

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



Plot model using visreg:

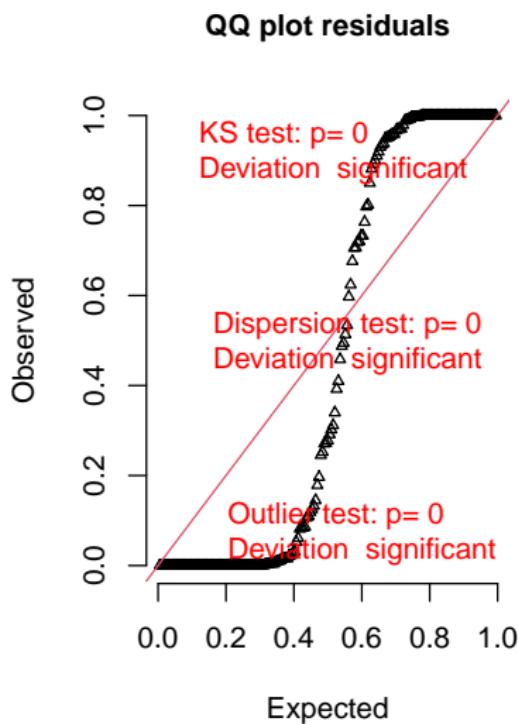
```
visreg(gdp.glm, scale = "response")
points(mortality/1000 ~ gdp, data = gdp)
```



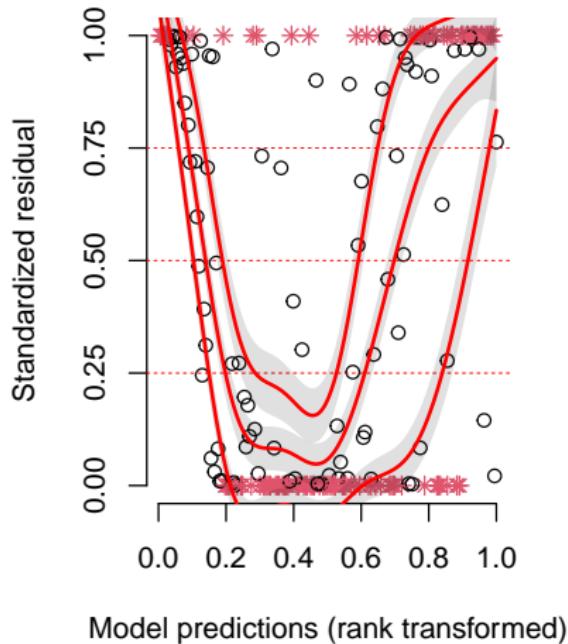
Residuals diagnostics with DHARMA

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

DHARMA residual diagnostics



Residual vs. predicted
Quantile deviations detected (red curves)
Combined adjusted quantile test significant



Overdispersion

Testing for overdispersion (DHARMA)

```
simres <- simulateResiduals(gdp.glm, refit = TRUE)
testDispersion(simres, plot = FALSE)
```

DHARMA nonparametric dispersion test via mean deviance residuals
vs. simulated-refitted

```
data: simres
dispersion = 21, p-value < 2.2e-16
alternative hypothesis: two.sided
```

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
```

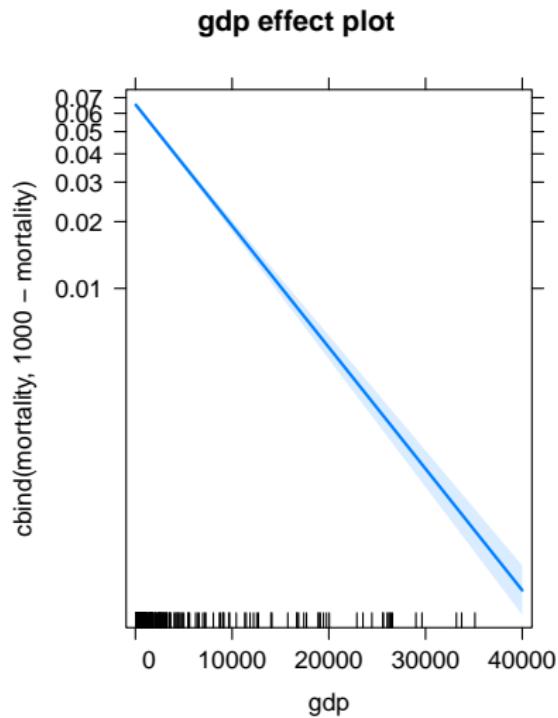
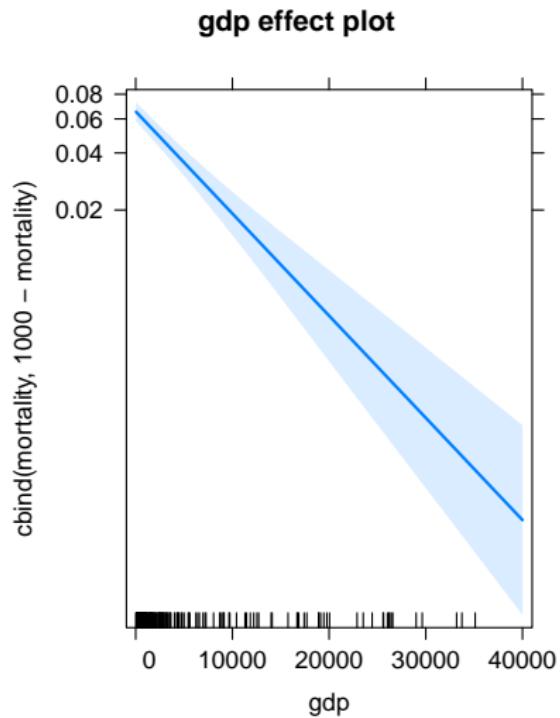
gdp effect

gdp

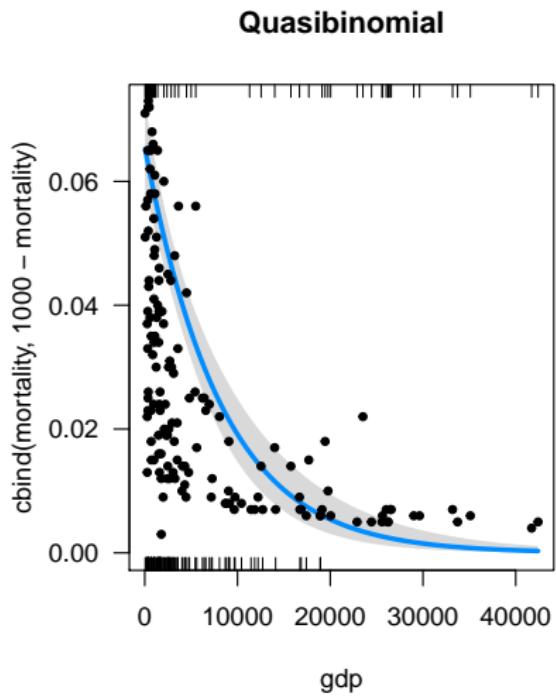
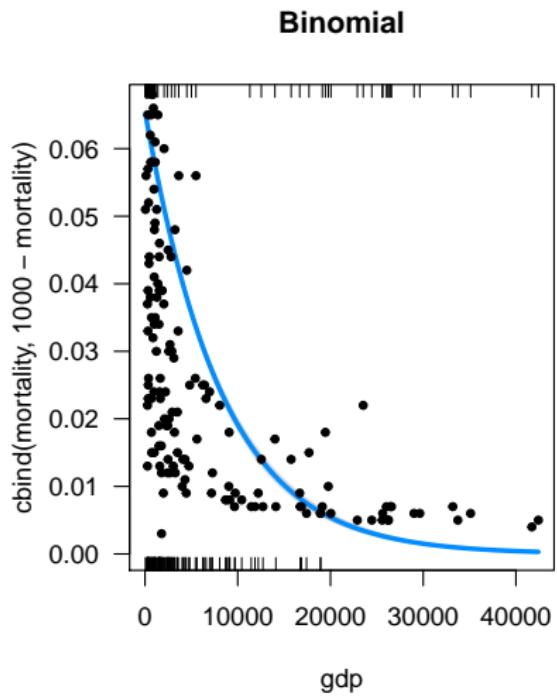
40	10000	20000	30000	40000
----	-------	-------	-------	-------

0.0652177296	0.0191438829	0.0054028095	0.0015096074	0.0004206154
--------------	--------------	--------------	--------------	--------------

But standard errors (uncertainty) do!



Plot model and data



Overdispersion

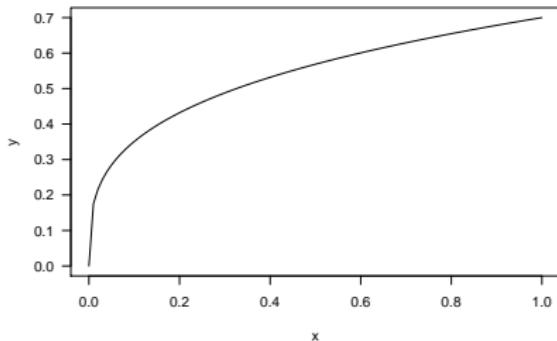
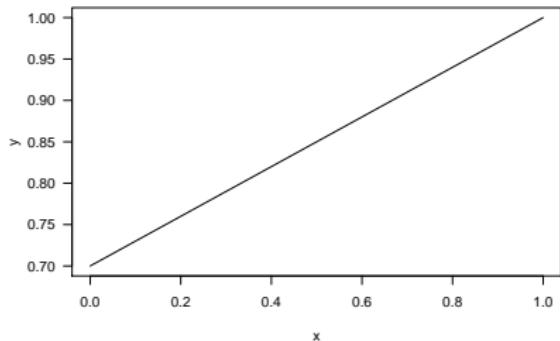
Whenever you fit logistic regression to **proportion** data, check family **quasibinomial**.

Think about the shape of relationships

$$y \sim x + z$$

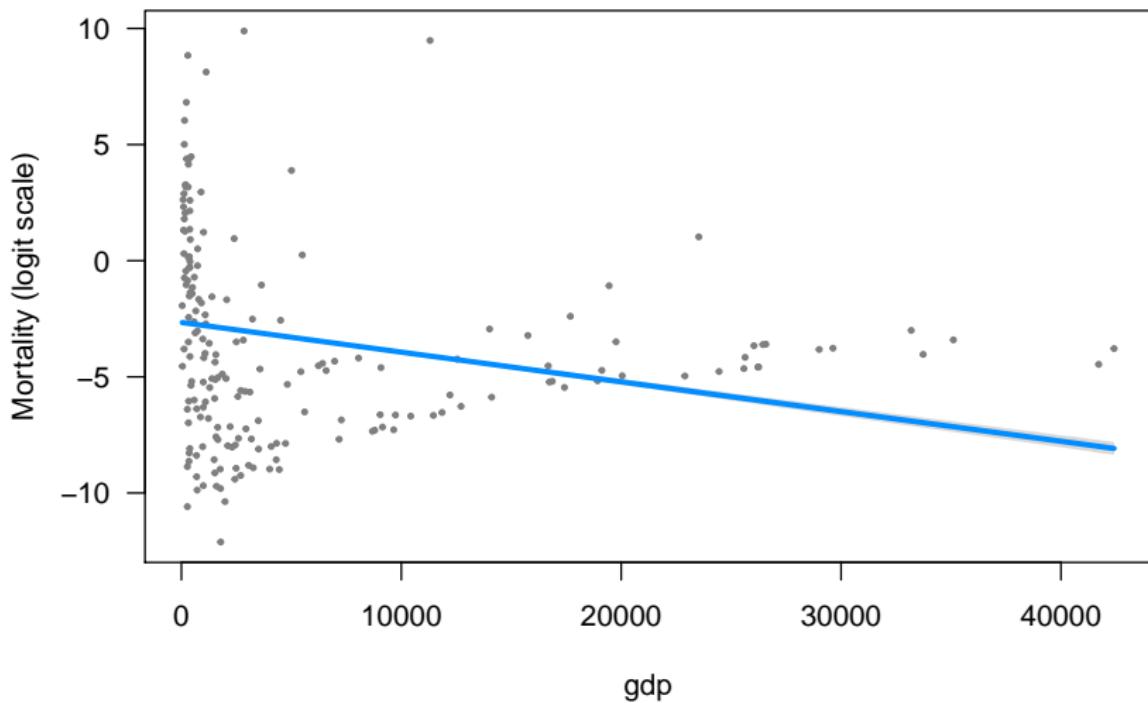
Really? Not everything has to be linear! Actually, it often is not.

Think about shape of relationship. See chapter 3 in Bolker's book.

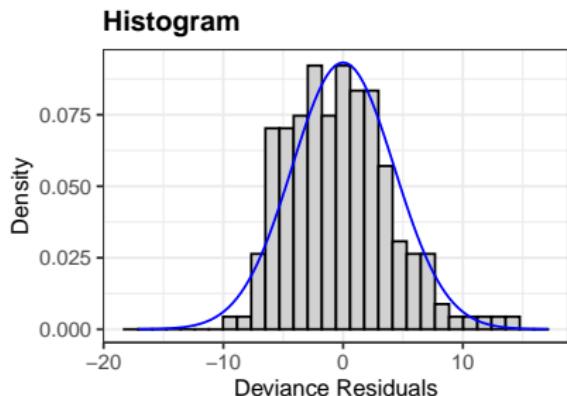
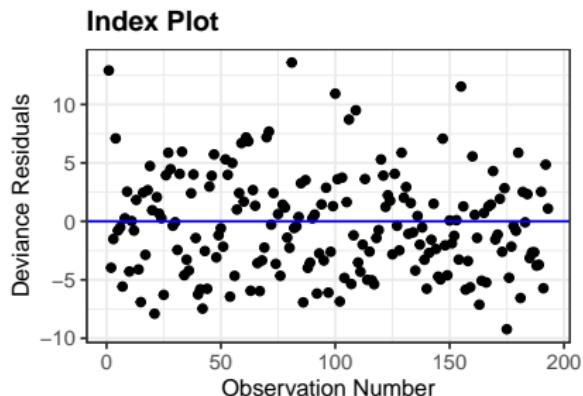
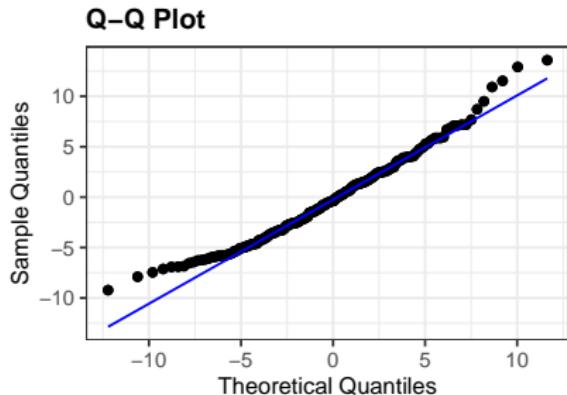
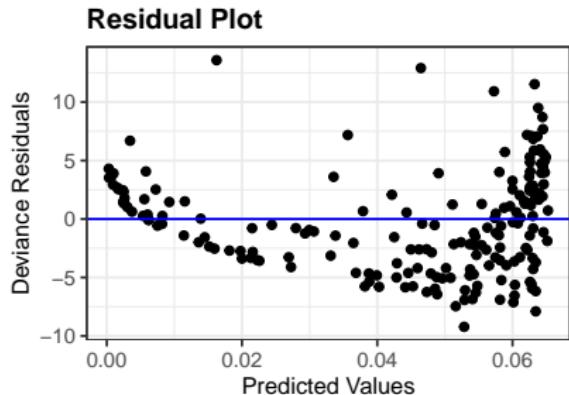


Think about the shape of relationships

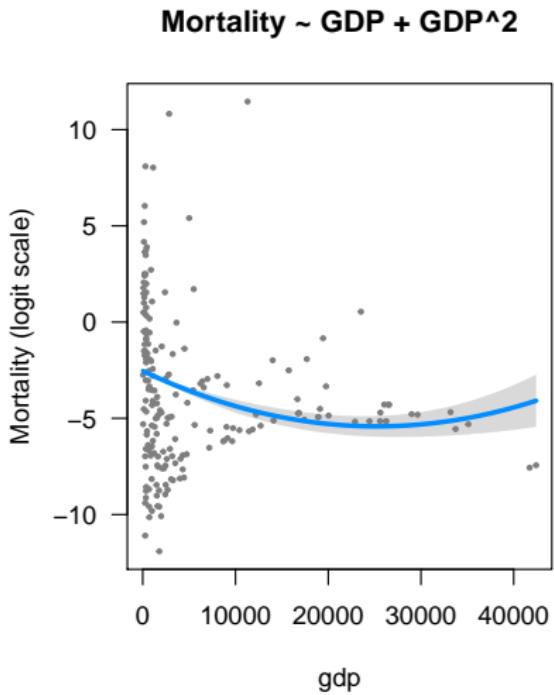
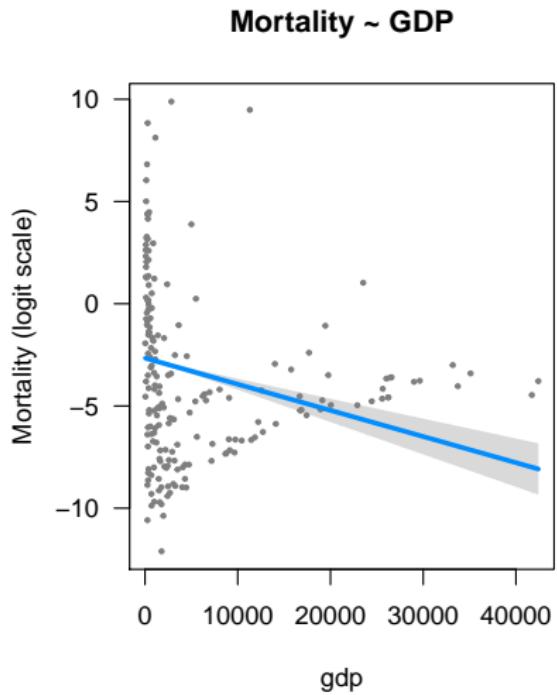
```
visreg(gdp.glm, ylab = "Mortality (logit scale)")
```



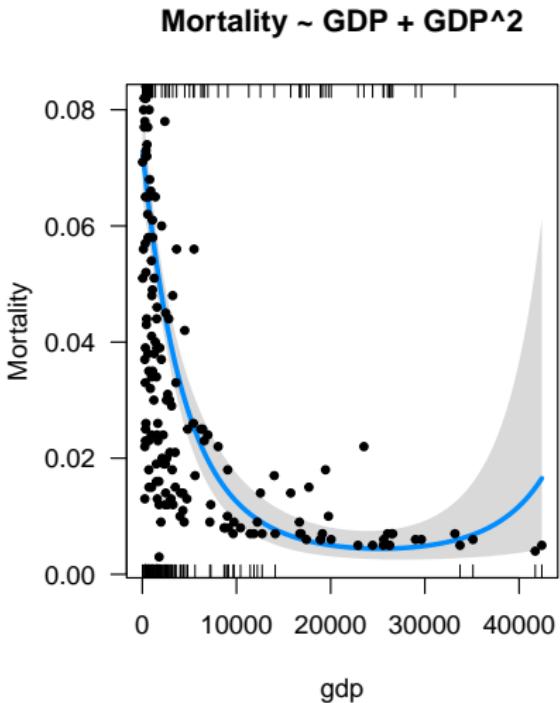
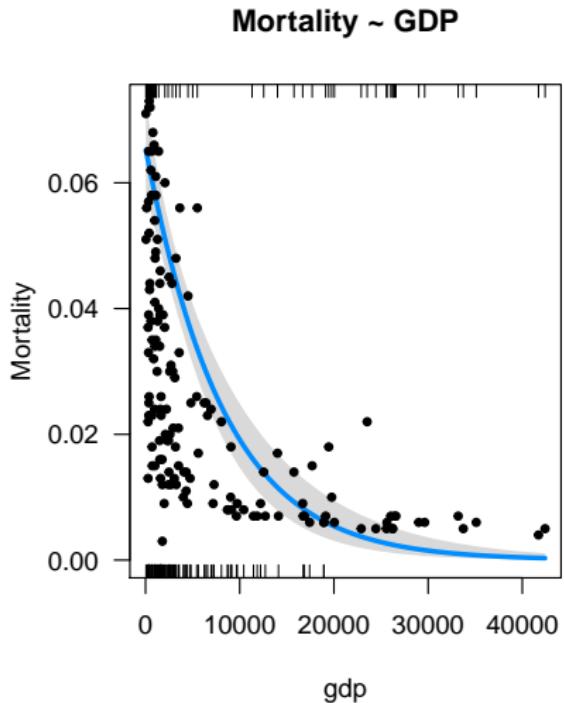
Think about the shape of relationships



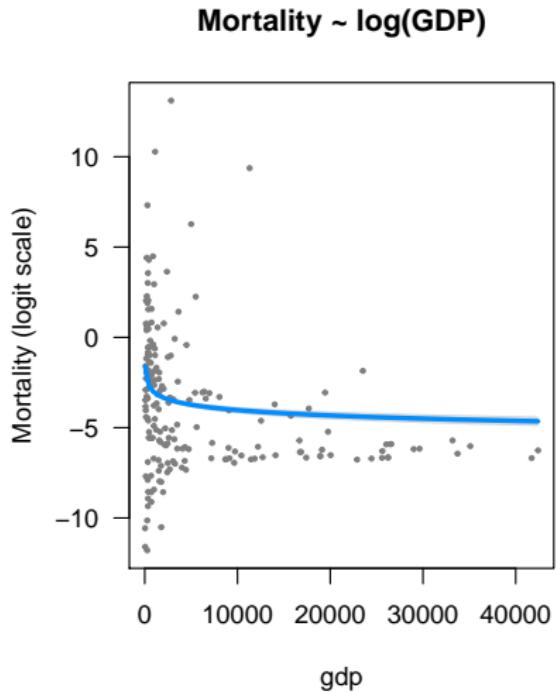
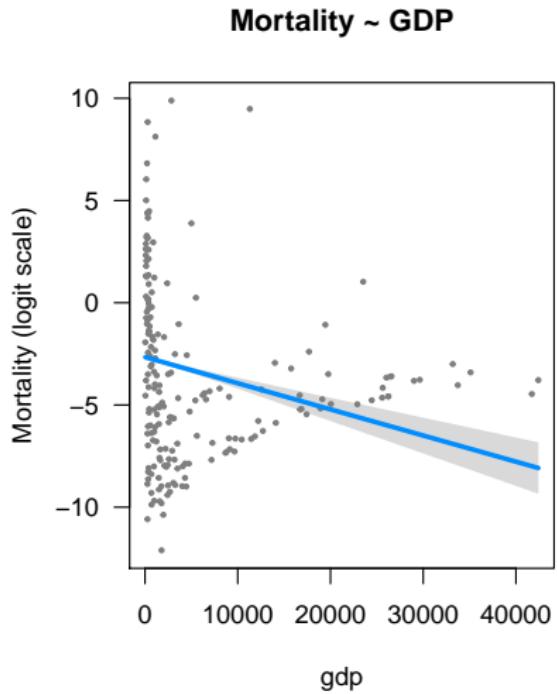
Think about the shape of relationships



Think about the shape of relationships

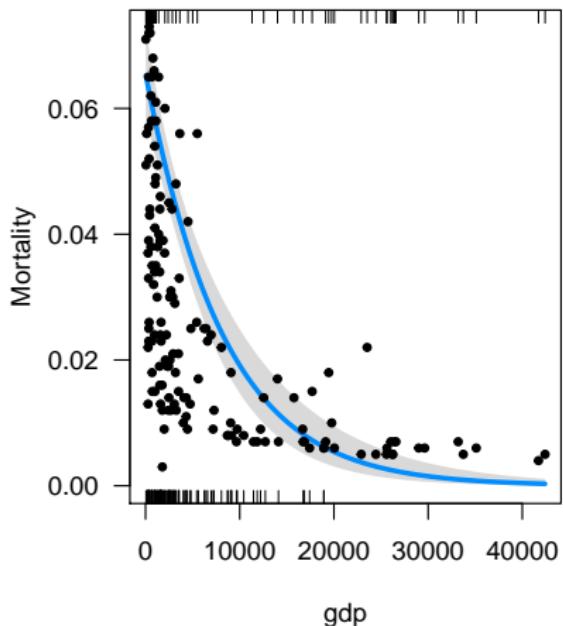


Think about the shape of relationships

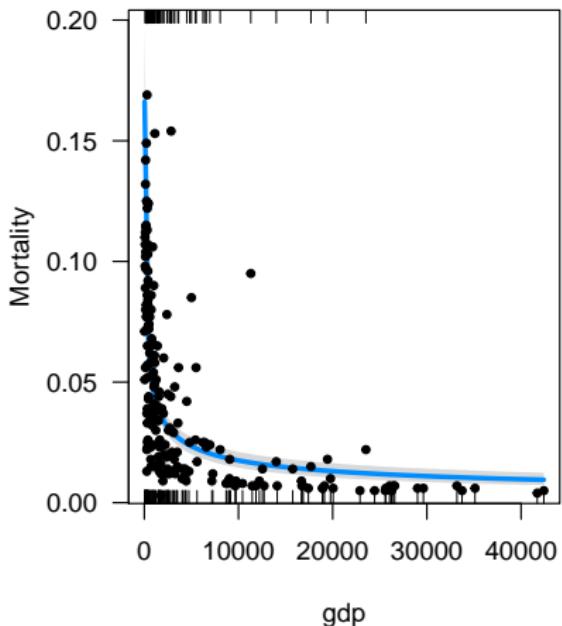


Think about the shape of relationships

Mortality ~ GDP



Mortality ~ log(GDP)



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

More examples

- ▶ seedset.csv: Comparing seed set among plants (Data from [Harder et al. 2011](#))

Seed set among plants

```
seed <- readr::read_csv("data/seedset.csv")
head(seed)

# A tibble: 6 x 6
  species    plant  pcmass fertilized  seeds  ovulecnt
  <chr>      <dbl>   <dbl>      <dbl>   <dbl>      <dbl>
1 ferruginea 2      0          70      52      330
2 ferruginea 2      0.2        321     188      461
3 ferruginea 2      0.485      351     278      435
4 ferruginea 2      0.737      386     301      430
5 ferruginea 2      1          367     342      419
6 ferruginea 3      0          185     39      470

seed$plant <- as.factor(seed$plant)
```

Questions:

<https://pollev.com/franciscorod726>

- ▶ Is seed set related to proportion of outcross pollen (pcmass)?

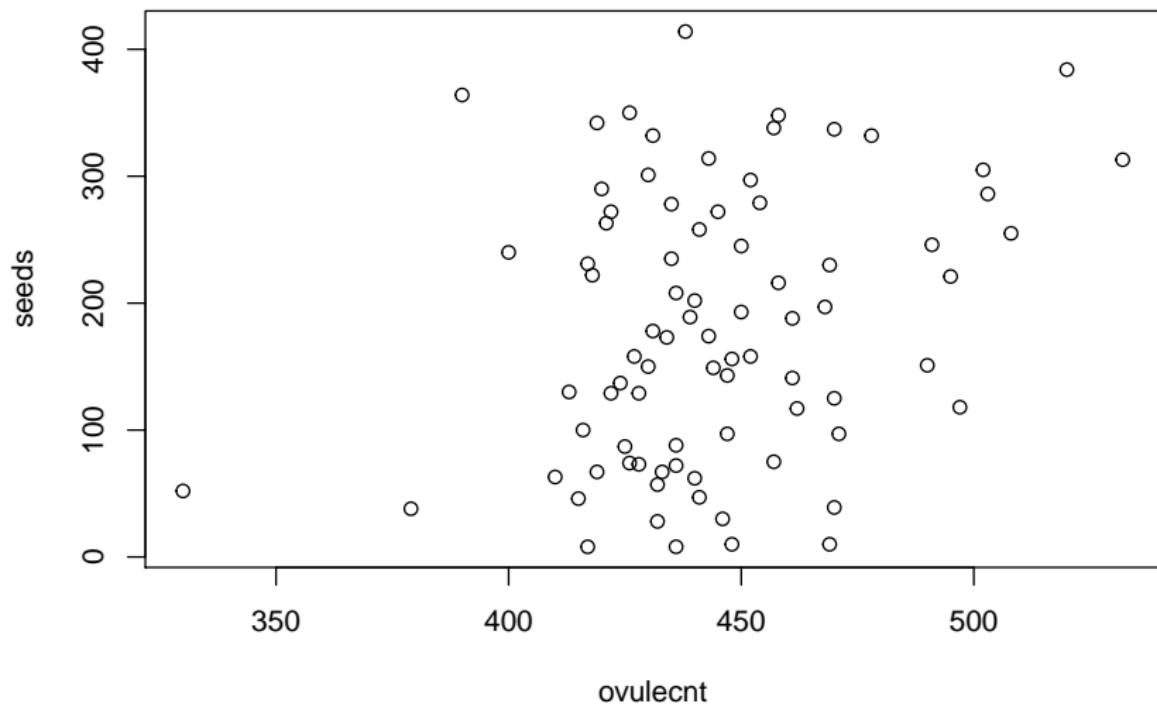
Questions:

<https://pollev.com/franciscorod726>

- ▶ Is seed set related to proportion of outcross pollen (pcmass)?
- ▶ Which plant had lower seed set?

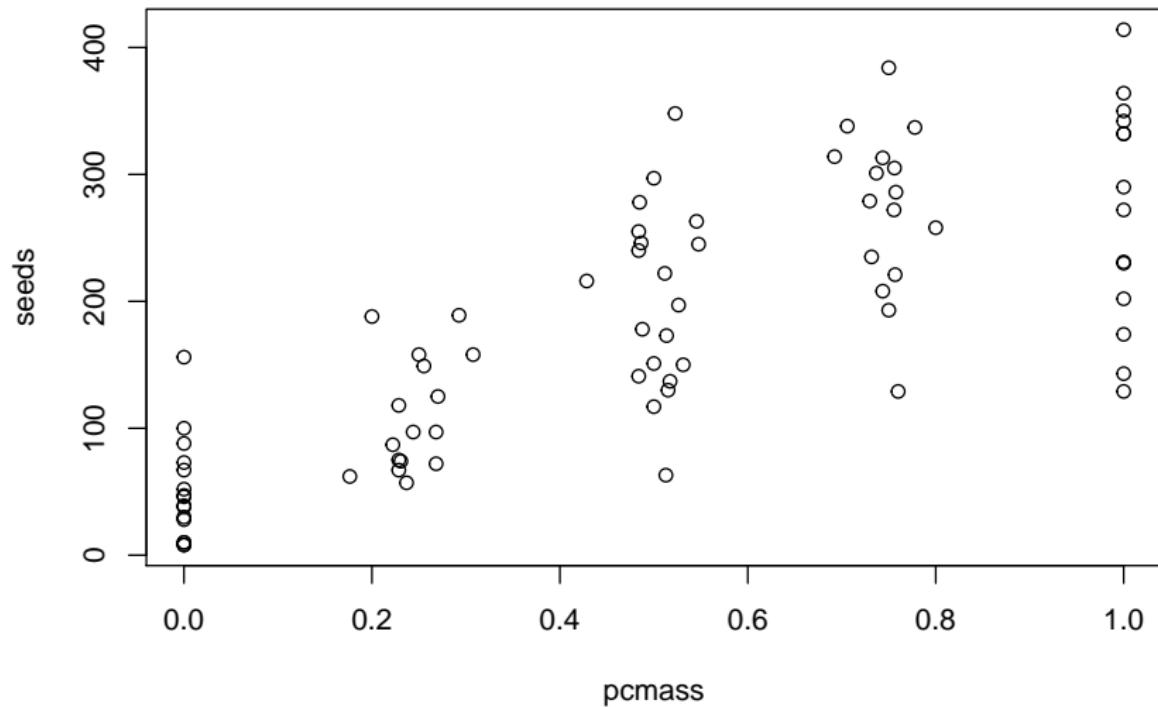
Number of seeds vs Number of ovules

```
plot(seeds ~ ovulecnt, data = seed)
```



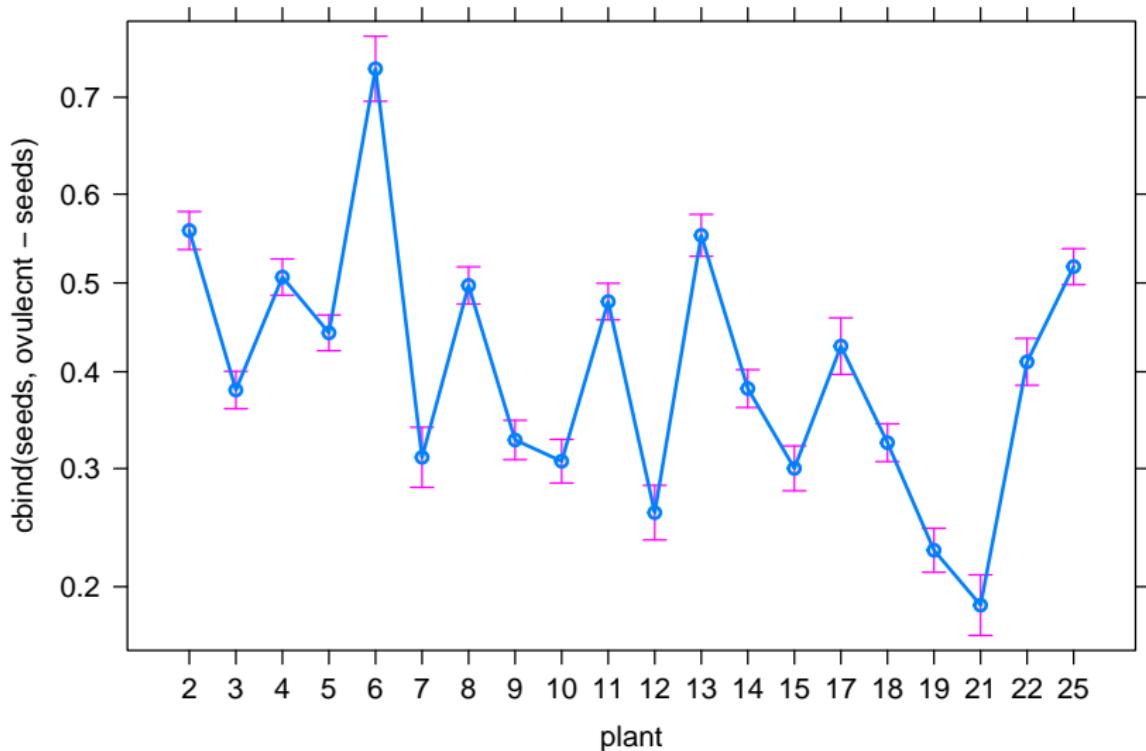
Number of seeds vs Proportion outcross pollen

```
plot(seeds ~ pcmass, data = seed)
```



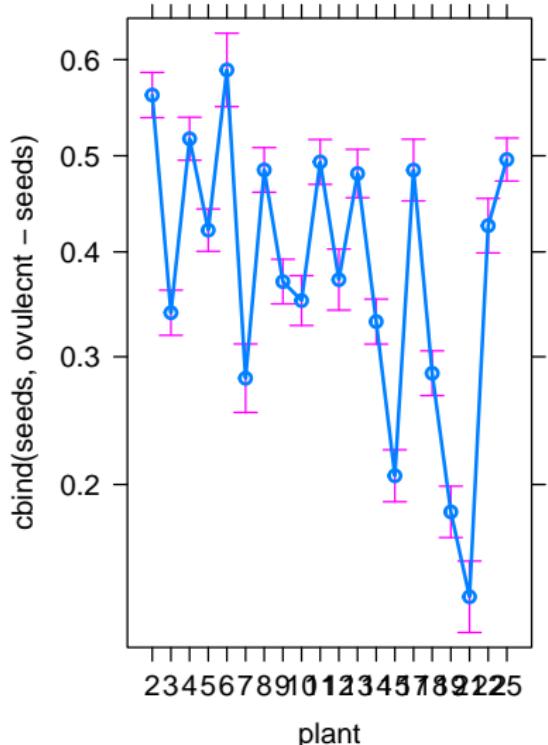
Seed set across plants

plant effect plot

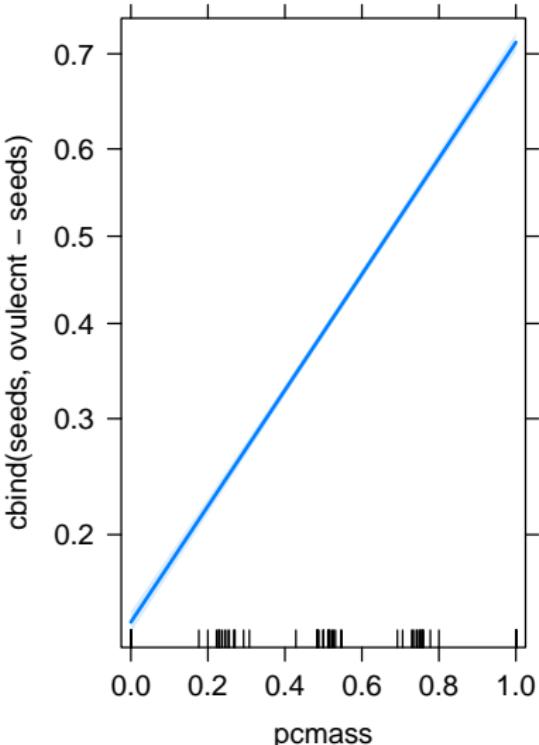


Seed set ~ outcross pollen

plant effect plot



pcmass effect plot



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

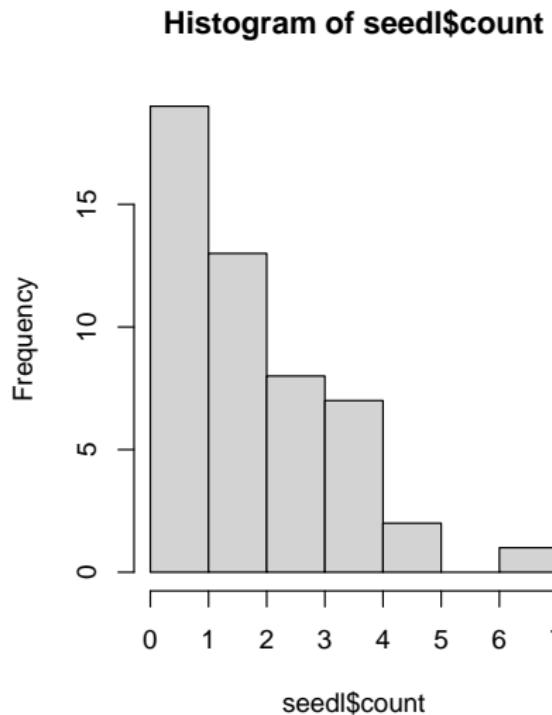
```
seedl <- read.csv("data/seedlings.csv")
```

X	count	row	col	1
Min. : 1.00	Min. :0.00	Min. :1	Min. : 1.0	Min. :
1st Qu.:13.25	1st Qu.:1.00	1st Qu.:2	1st Qu.: 3.0	1st Q
Median :25.50	Median :2.00	Median :3	Median : 5.5	Media
Mean :25.50	Mean :2.14	Mean :3	Mean : 5.5	Mean
3rd Qu.:37.75	3rd Qu.:3.00	3rd Qu.:4	3rd Qu.: 8.0	3rd Q
Max. :50.00	Max. :7.00	Max. :5	Max. :10.0	Max.
area				
Min. : 0.25				
1st Qu.: 0.25				
Median : 0.50				
Mean : 0.62				
3rd Qu.: 1.00				
Max. : 1.00				

EDA

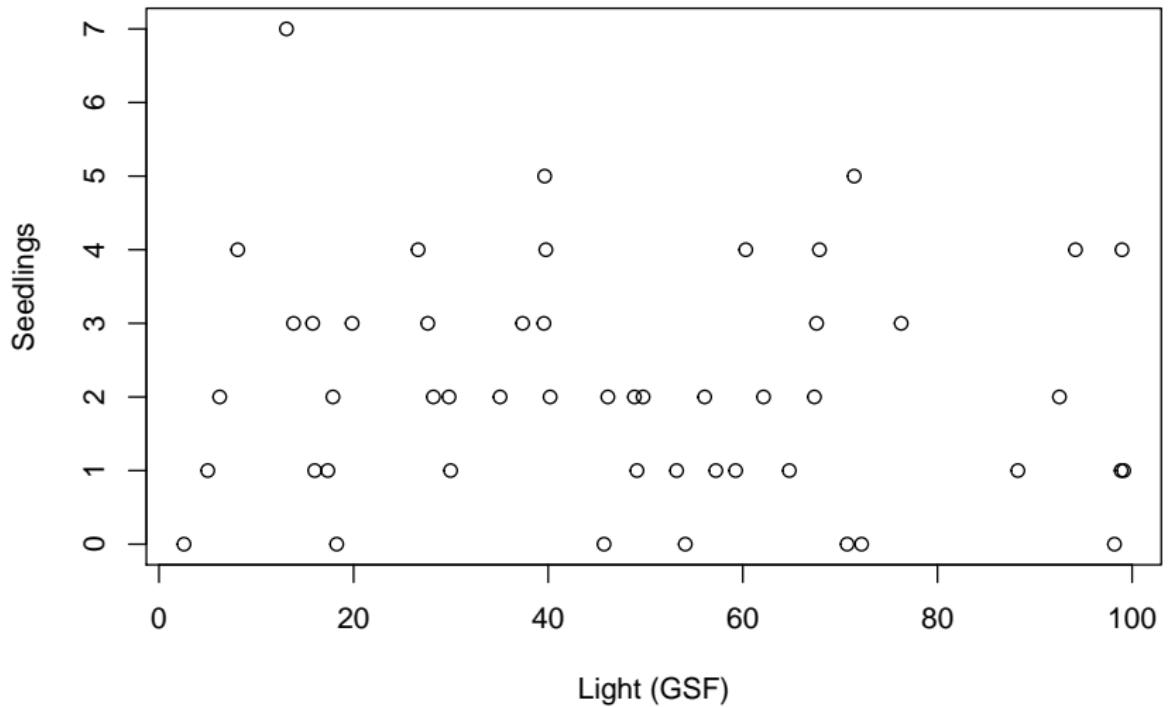
```
table(seed1$count)
```

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



Q: Relationship between Nseedlings and light?

```
plot(seed1$light, seed1$count, xlab = "Light (GSF)", ylab = "See
```



Let's fit model (Poisson regression)

```
seed1.glm <- glm(count ~ light, data = seed1, family = poisson)
summary(seed1.glm)
```

Call:

```
glm(formula = count ~ light, family = poisson, data = seed1)
```

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(seed1.glm)
```

	light
(Intercept)	0.881805022
	-0.002575656

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

```
exp(coef(seed1.glm))
```

	light
(Intercept)	2.4152554
	0.9974277

Using effects package

```
summary(allEffects(seed1.glm))
```

model: count ~ light

light effect

light

	3	30	50	70	100
--	---	----	----	----	-----

2.396665	2.235657	2.123408	2.016794	1.866826
----------	----------	----------	----------	----------

Lower 95 Percent Confidence Limits

light

	3	30	50	70	100
--	---	----	----	----	-----

1.684579	1.795202	1.753373	1.567785	1.228247
----------	----------	----------	----------	----------

Upper 95 Percent Confidence Limits

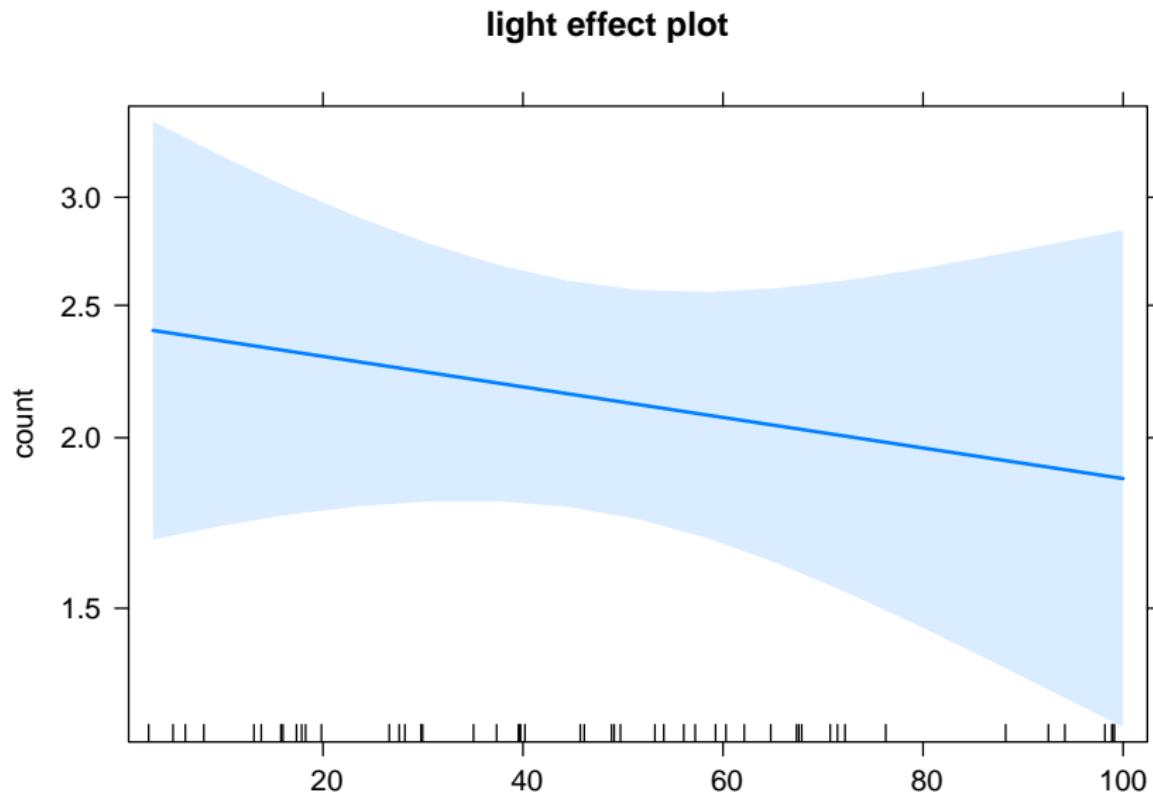
light

	3	30	50	70	100
--	---	----	----	----	-----

3.409754	2.784179	2.571535	2.594398	2.837408
----------	----------	----------	----------	----------

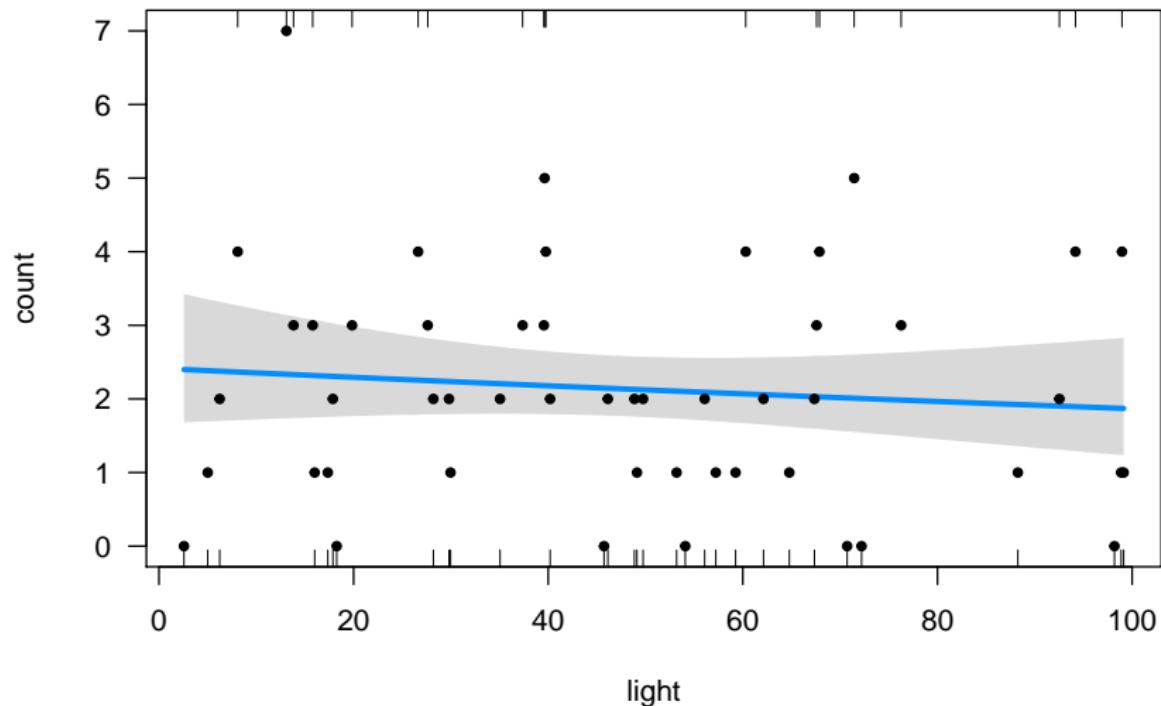
So what's the relationship between Nseedlings and light?

```
plot(allEffects(seed1.glm))
```



Using visreg

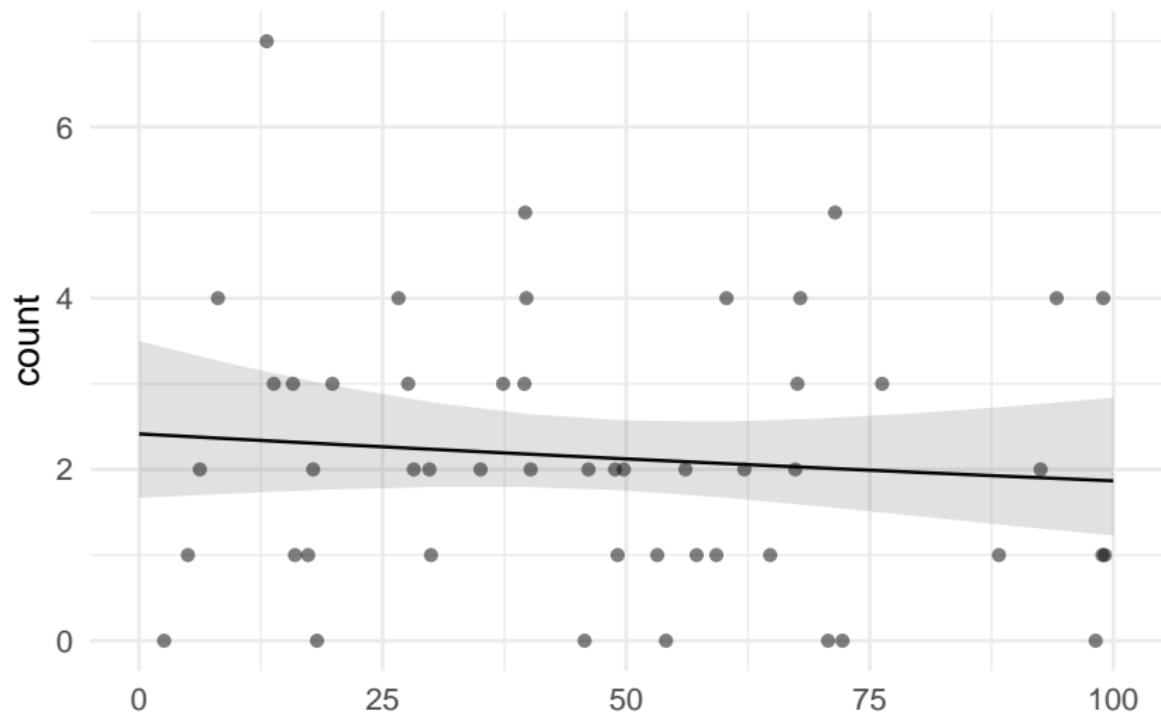
```
visreg(seedl.glm, scale = "response", ylim = c(0, 7))
points(count ~ light, data = seedl, pch = 20)
```



Using sjPlot

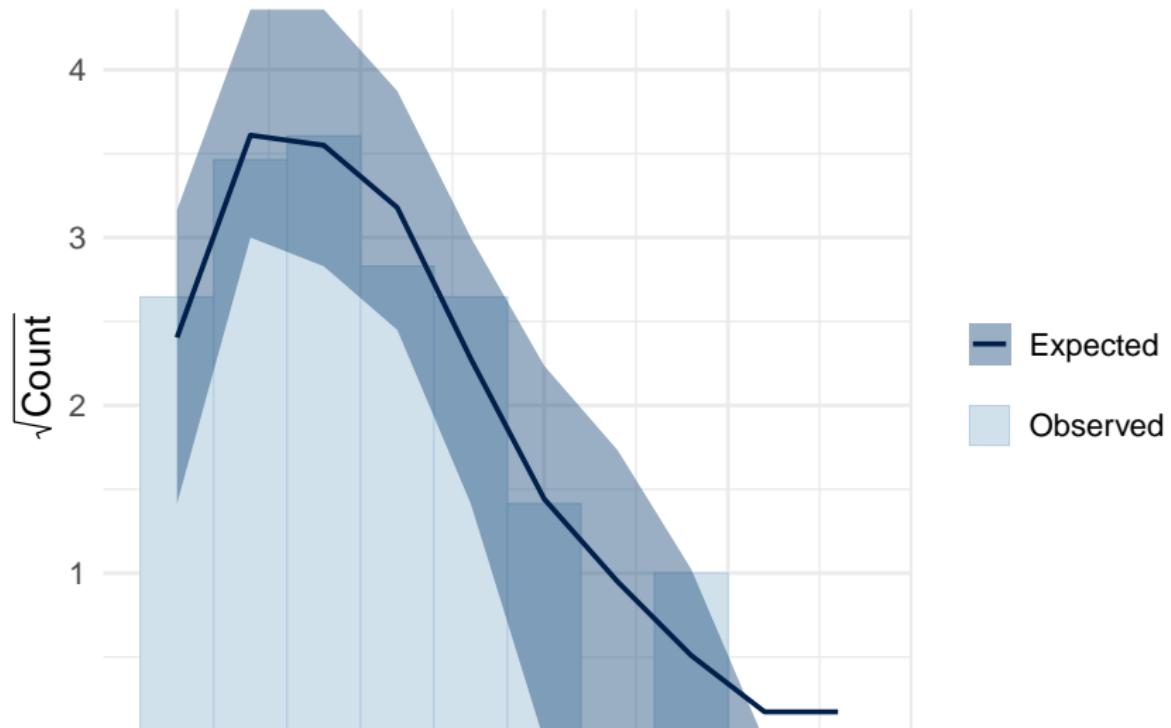
```
sjPlot::plot_model(seed1.glm, type = "eff", show.data = TRUE)  
$light
```

Predicted counts of count

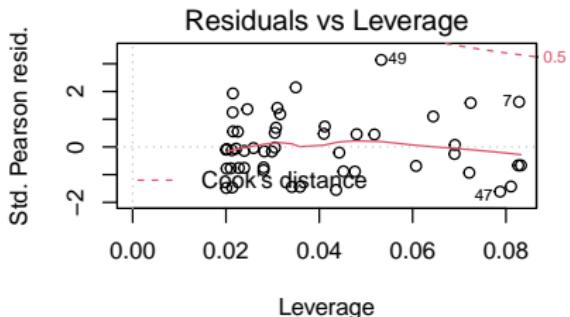
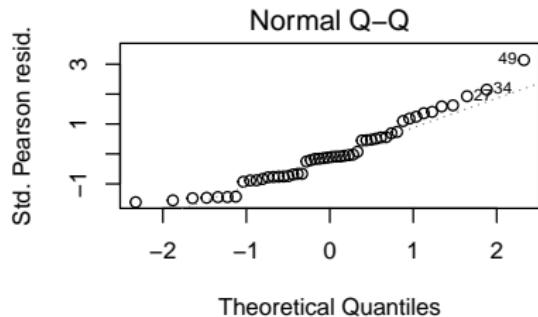
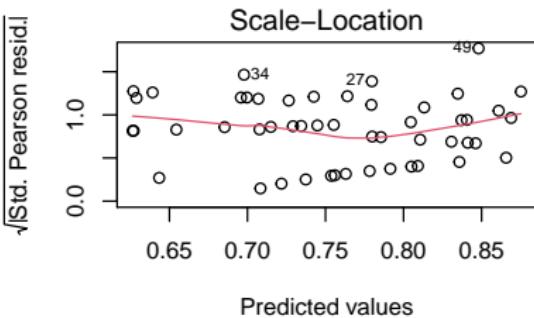
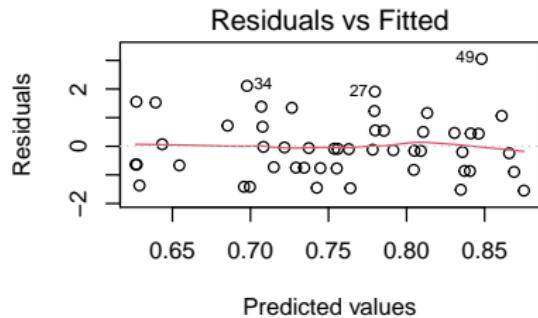


Calibration plot with count data: rootograms

```
sims <- simulate(seed1.glm, nsim = 100)
yrep <- t(as.matrix(sims))
bayesplot::ppc_rootogram(seed1$count, yrep)
```



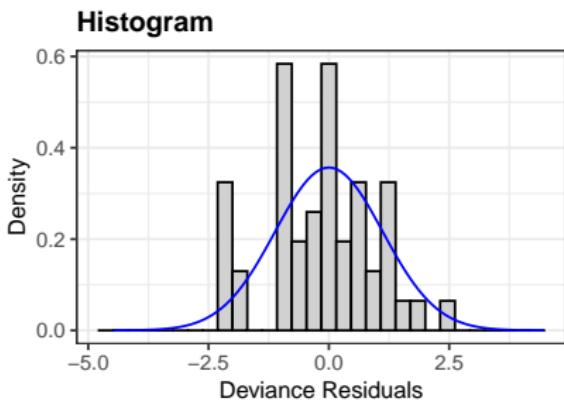
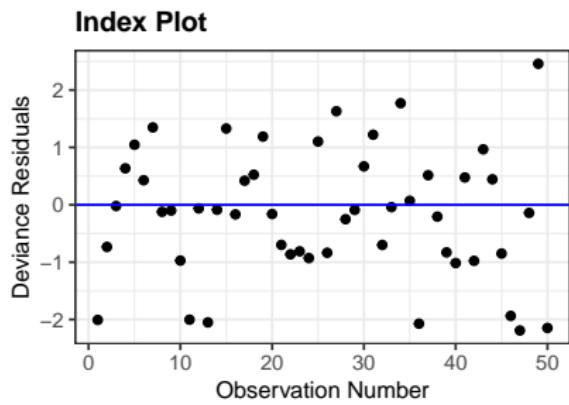
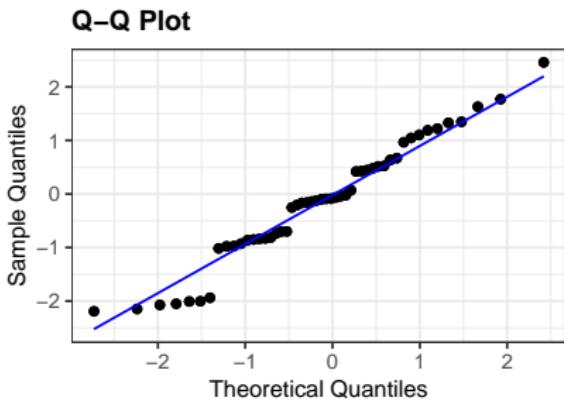
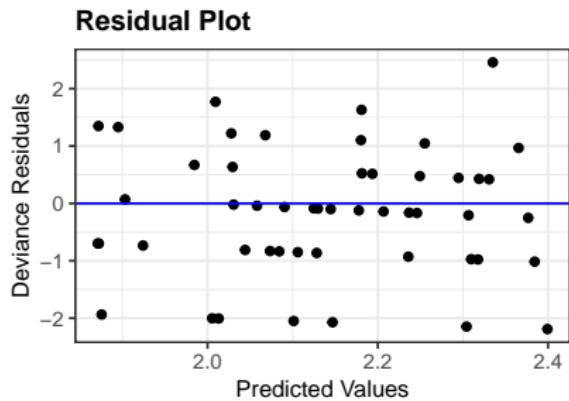
Poisson regression: model checking



null device

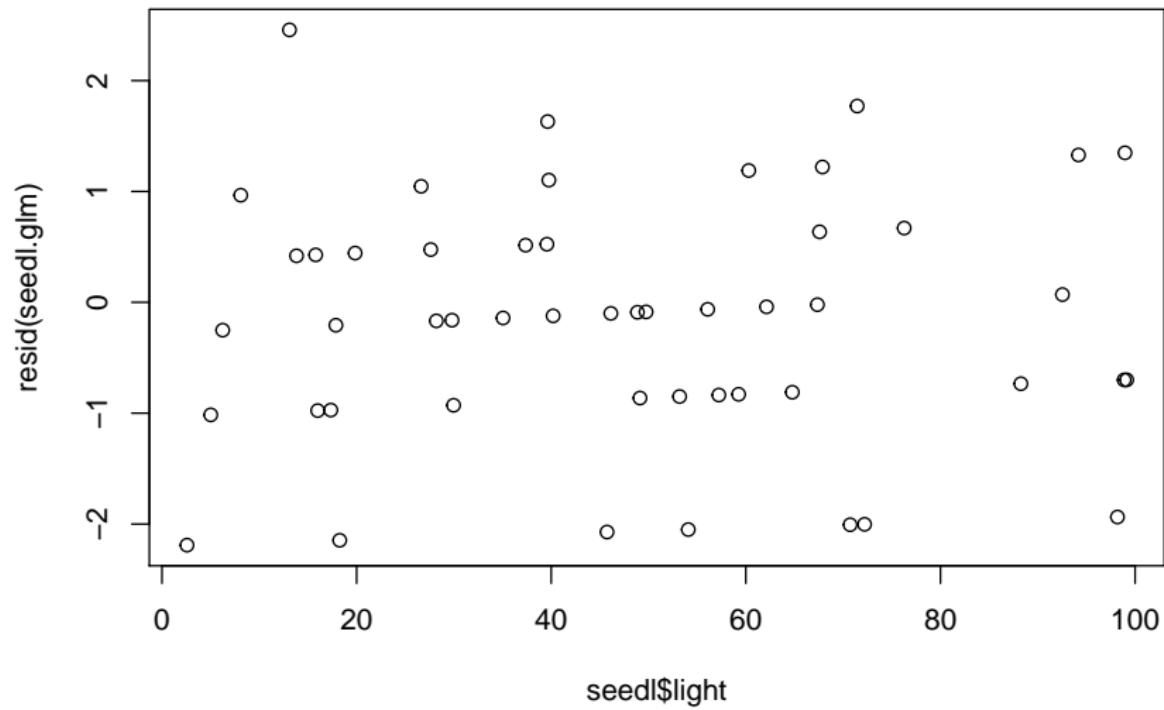
Poisson regression: model checking

```
ggResidpanel:::resid_panel(seed1.glm)
```



Is there pattern of residuals along predictor?

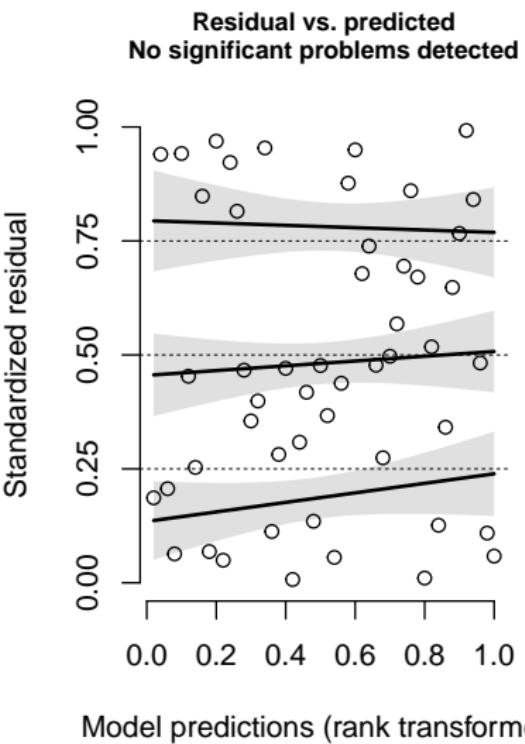
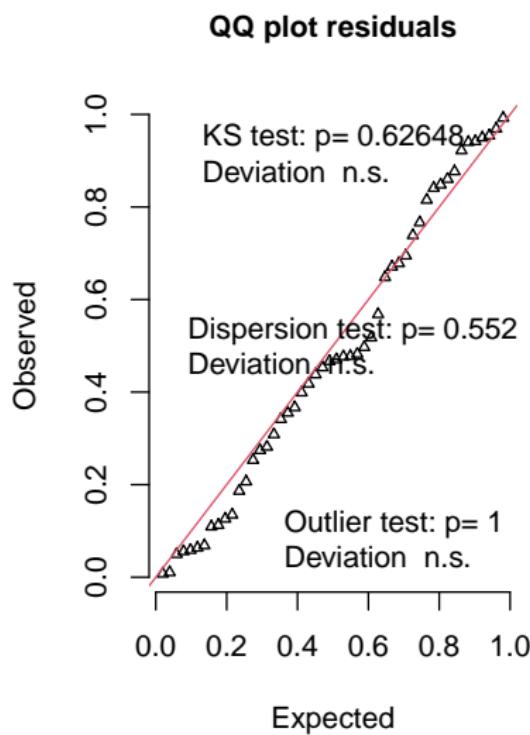
```
plot(seed1$light, resid(seed1.glm))
```



Residuals diagnostics with DHARMA

```
DHARMA:::simulateResiduals(seed1.glm, plot = TRUE)
```

DHARMA residual diagnostics



Poisson regression: Overdispersion

Always check overdispersion with count data

```
simres <- simulateResiduals(seed1.glm, refit = TRUE)
testDispersion(simres, plot = FALSE)
```

DHARMA nonparametric dispersion test via mean deviance residuals
vs. simulated-refitted

```
data: simres
dispersion = 1.1655, p-value = 0.432
alternative hypothesis: two.sided
```

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

AIC: NA

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
----------	----------	----------	----------	----------

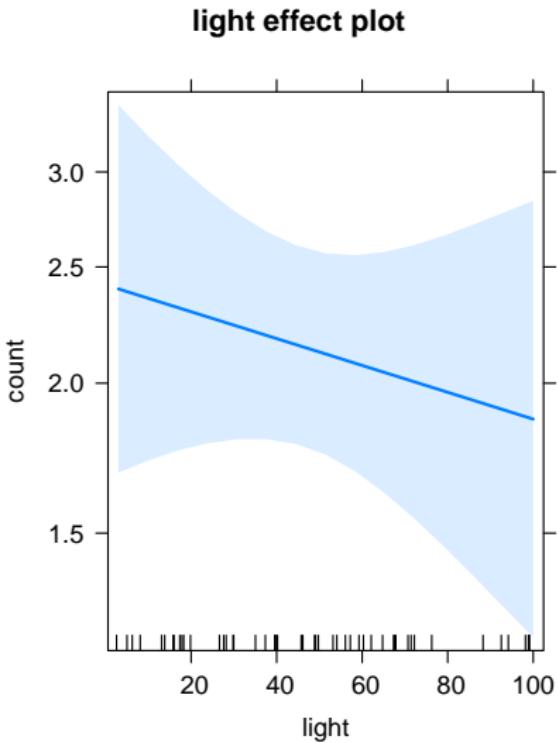
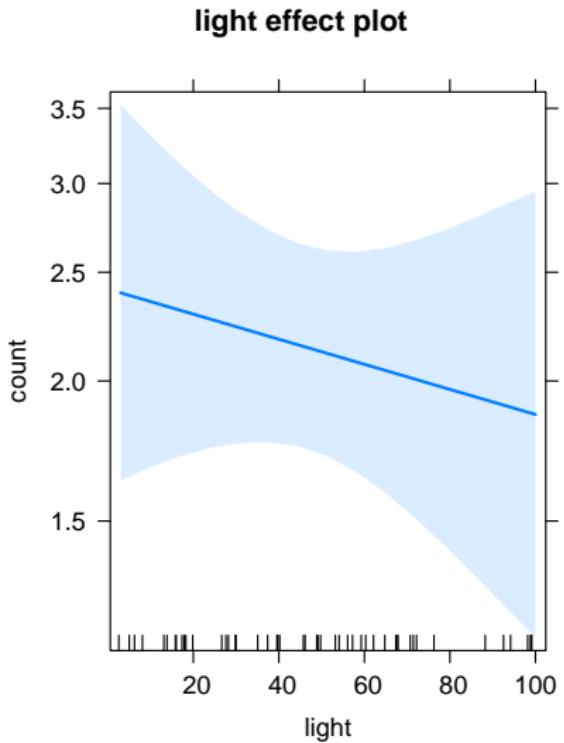
```
model: count ~ light
```

```
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 \sim light

J. R. Statist. Soc. A (1993)
156, Part 3, pp. 379-392

Spurious Correlation and the Fallacy of the Ratio Standard Revisited

By RICHARD A. KRONMAL†

Use offset to standardise response variables in GLMs

```
seed1.offset <- glm(count ~ light, offset = seed1$area, data = s
summary(seed1.offset)
```

Call:

```
glm(formula = count ~ light, family = poisson, data = seed1,
    offset = seed1$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
Residual deviance: 68.535 on 48 degrees of freedom

Note estimates now referred to area units

```
exp(coef(seed1.offset))
```

	light
(Intercept)	1.3491422
	0.9955123

Other examples

- ▶ Infant mortality \sim GDP

Other examples

- ▶ Infant mortality \sim GDP
- ▶ Number of cones consumed by squirrels ([data](#))

Mixed / Multilevel models

Example dataset: trees

- ▶ Data on 1000 trees from 10 sites.

```
head(trees)
```

	site	dbh	height	sex	dead
1	4	29.68	36.1	male	0
2	5	33.29	42.3	male	0
3	2	28.03	41.9	female	0
4	5	39.86	46.5	female	0
5	1	47.94	43.9	female	0
6	1	10.82	26.2	male	0

Example dataset: trees

- ▶ Data on 1000 trees from 10 sites.
- ▶ Trees per site: 4 - 392.

```
head(trees)
```

	site	dbh	height	sex	dead
1	4	29.68	36.1	male	0
2	5	33.29	42.3	male	0
3	2	28.03	41.9	female	0
4	5	39.86	46.5	female	0
5	1	47.94	43.9	female	0
6	1	10.82	26.2	male	0

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
-13.3270	-2.8978	0.1057	2.7924	12.9511

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)							
(Intercept)	19.33920	0.31064	62.26	<2e-16 ***							
dbh	0.61570	0.01013	60.79	<2e-16 ***							

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

Residual standard error: 4.093 on 998 degrees of freedom

Multiple R-squared: 0.7874, Adjusted R-squared: 0.7871

F-statistic: 3695 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:

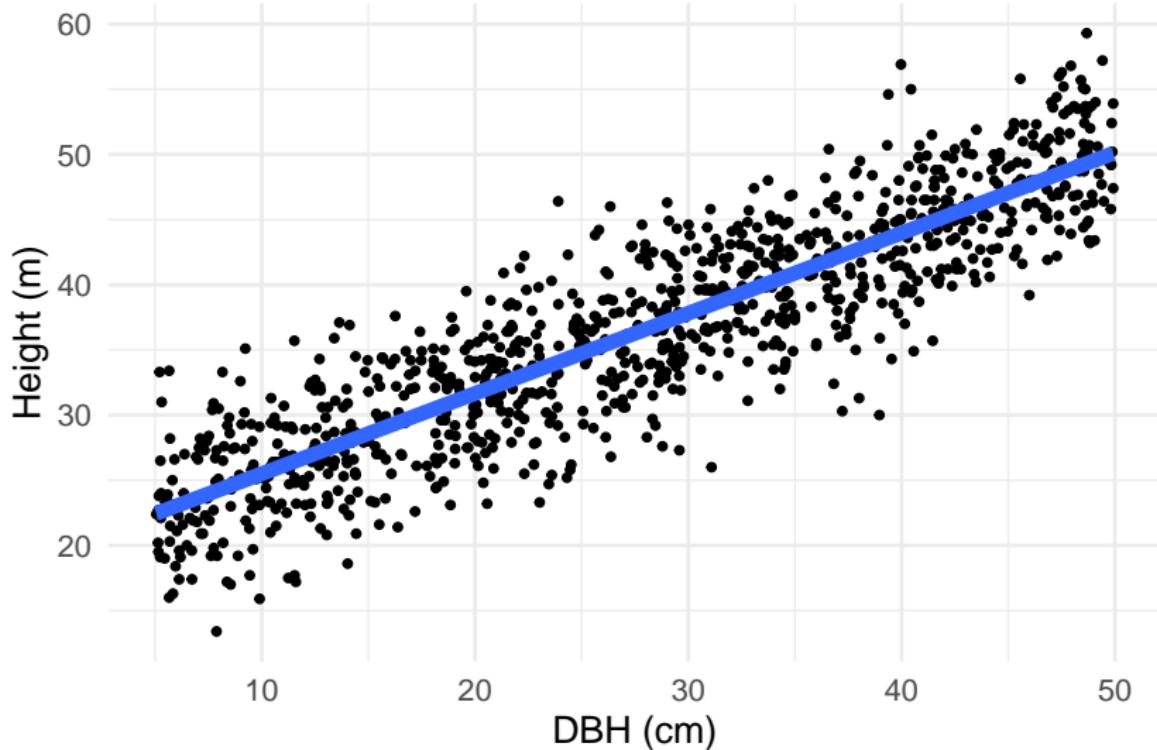
$$Height_i \sim N(\mu_i, \sigma^2)$$
$$\mu_i = \alpha + \beta DBH_i$$

α : expected height when DBH = 0

β : how much height increases with every unit increase of DBH

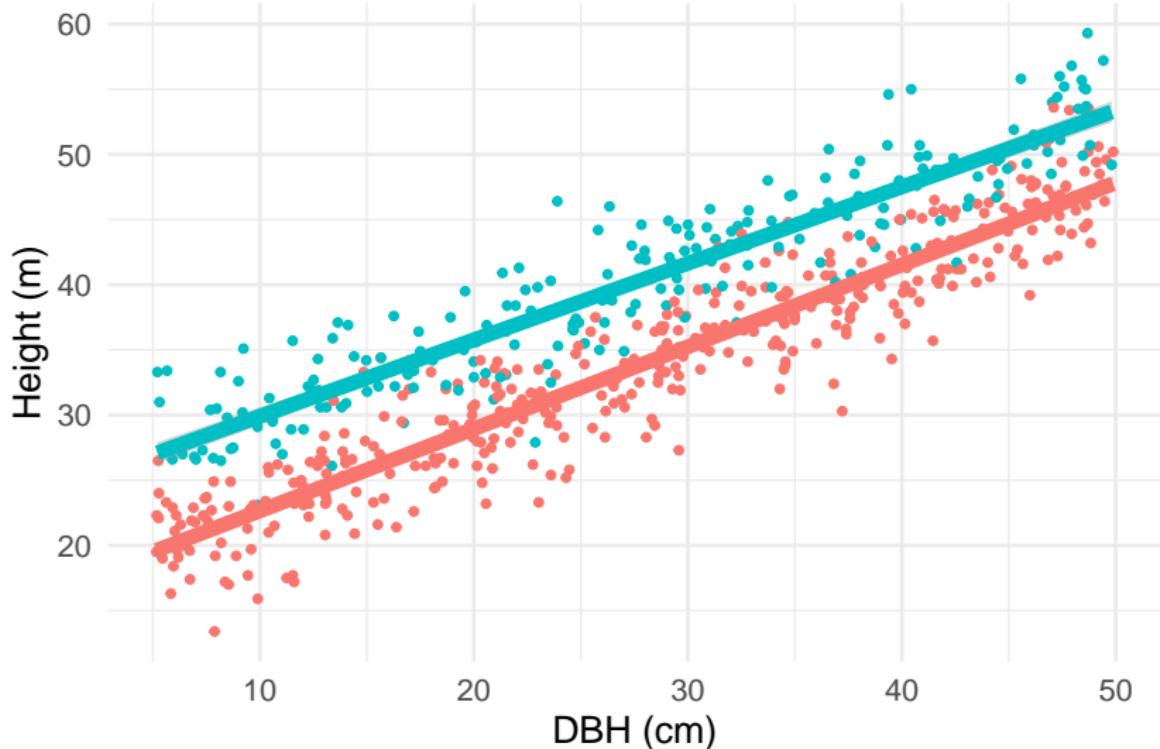
There is only one intercept

Single intercept



What if allometry varies among sites?

Different intercept for each site



Fitting a varying intercepts model with lm

Call:

```
lm(formula = height ~ factor(site) + dbh, data = trees)
```

Residuals:

Min	1Q	Median	3Q	Max
-10.1130	-1.9885	0.0582	2.0314	11.3320

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	16.699037	0.260565	64.088	< 2e-16 ***
factor(site)2	6.504303	0.256730	25.335	< 2e-16 ***
factor(site)3	4.357457	0.354181	12.303	< 2e-16 ***
factor(site)4	1.934650	0.356102	5.433	6.98e-08 ***
factor(site)5	3.637432	0.339688	10.708	< 2e-16 ***
factor(site)6	4.204511	0.421906	9.966	< 2e-16 ***
factor(site)7	-0.176193	0.666772	-0.264	0.7916
factor(site)8	-5.312648	0.893603	-5.945	3.82e-09 ***
factor(site)9	5.437049	1.087766	4.998	6.84e-07 ***
factor(site)10	2.263338	1.369986	1.652	0.0988 .
dbh	0.617075	0.007574	81.473	< 2e-16 ***

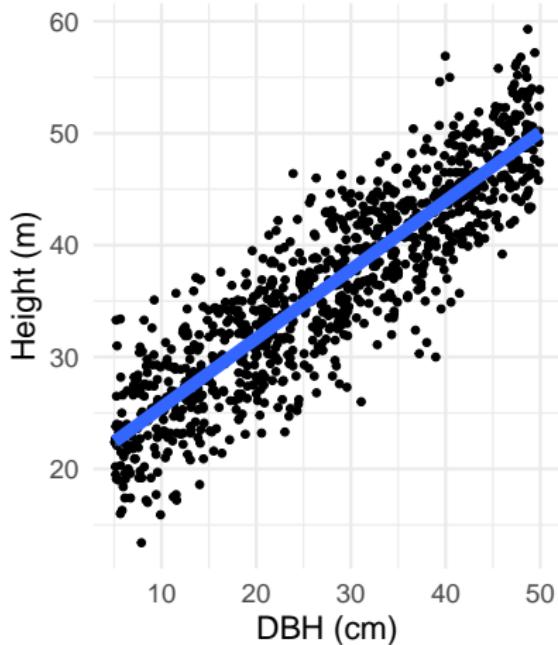
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.043 on 989 degrees of freedom

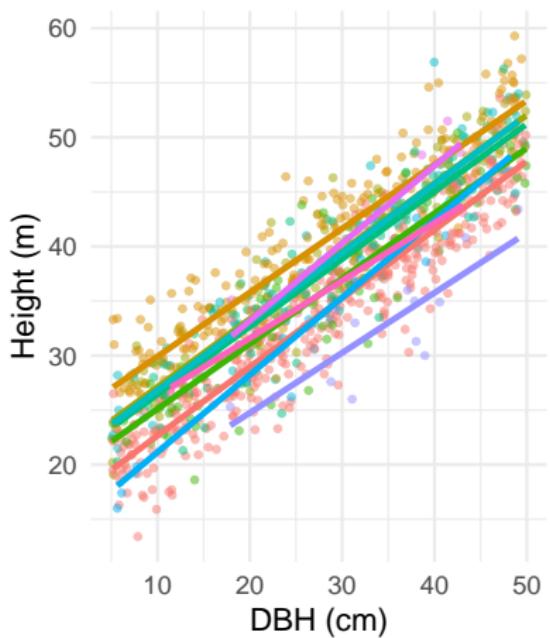
Multiple R-squared: 0.8835, Adjusted R-squared: 0.8823

Single vs varying intercept

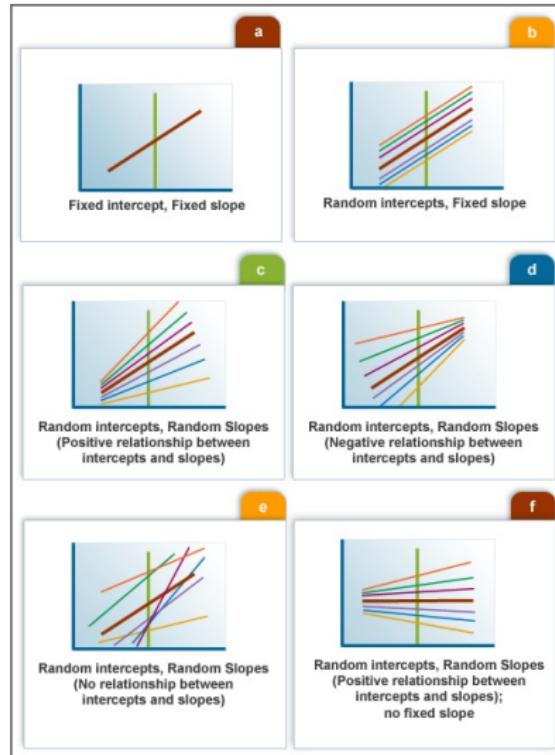
Single intercept



Different intercept for each si

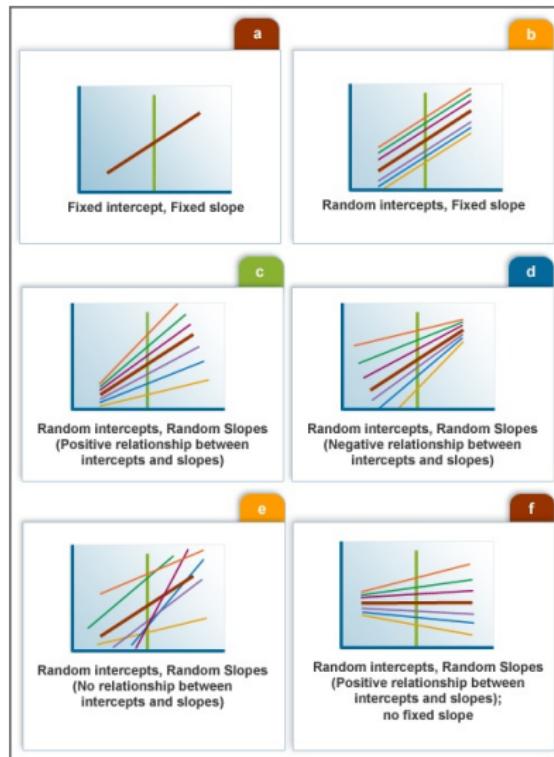


Mixed models enable us to account for variability



► Varying intercepts

Mixed models enable us to account for variability



- ▶ Varying intercepts
- ▶ Varying slopes

Mixed model with varying intercepts

$$\begin{aligned}y_i &= a + \alpha_j + b \cdot x_i + \varepsilon_i \\ \alpha_j &\sim N(0, \tau^2) \\ \varepsilon_i &\sim N(0, \sigma^2)\end{aligned}$$

En nuestro ejemplo:

$$\begin{aligned}Height_i &= a + site_j + b \cdot DBH_i + \varepsilon_i \\ site_j &\sim N(0, \tau^2) \\ \varepsilon_i &\sim N(0, \sigma^2)\end{aligned}$$

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

Hence there's gradient between

- ▶ **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.
 - ▶ $\text{lm}(\text{height} \sim \text{dbh})$
- ▶ **no pooling**: One *independent* intercept for each site.

Hence there's gradient between

- ▶ **complete pooling**: Single overall intercept.
 - ▶ $\text{lm}(\text{height} \sim \text{dbh})$
- ▶ **no pooling**: One *independent* intercept for each site.
 - ▶ $\text{lm}(\text{height} \sim \text{dbh} + \text{site})$

Hence there's gradient between

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

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?

- ▶ Varies by group

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

What is a random effect, really?

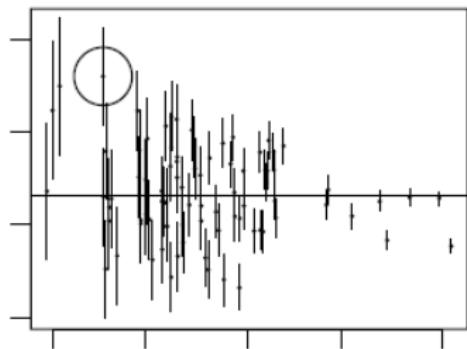
- ▶ Varies by group
- ▶ Variation estimated with **probability model**

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

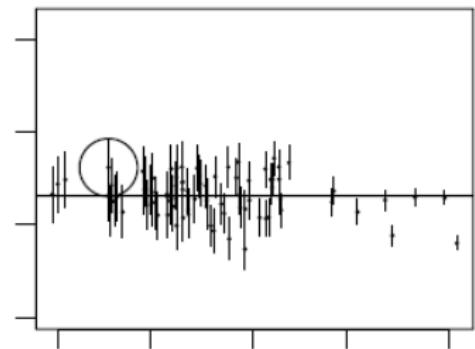
Shrinkage improves parameter estimation

Especially for groups with low sample size

No pooling



Multilevel model



From Gelman & Hill p. 253

Fitting mixed/multilevel models

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

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

```
REML criterion at convergence: 5108.3
```

```
Scaled residuals:
```

Min	1Q	Median	3Q	Max
-3.3199	-0.6607	0.0227	0.6716	3.7328

```
Random effects:
```

Groups	Name	Variance	Std.Dev.
site	(Intercept)	11.195	3.346
	Residual	9.261	3.043

```
Number of obs: 1000, groups: site, 10
```

```
Fixed effects:
```

	Estimate	Std. Error	t value
(Intercept)	19.011468	1.100444	17.28
dbh	0.616927	0.007572	81.47

```
Correlation of Fixed Effects:
```

```
  (Intr)
```

```
  dbh  0.107
```

Retrieve model coefficients

```
coef(mixed)
```

```
$site
  (Intercept)      dbh
1    16.70800 0.6169271
2    23.19162 0.6169271
3    21.04229 0.6169271
4    18.64086 0.6169271
5    20.32995 0.6169271
6    20.88200 0.6169271
7    16.61686 0.6169271
8    11.88302 0.6169271
9    21.84779 0.6169271
10   18.97228 0.6169271
```

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

Broom: model estimates in tidy form

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

```
# A tibble: 4 x 6  
  effect  group    term      estimate std.error statistic  
  <chr>   <chr>    <chr>      <dbl>     <dbl>      <dbl>  
1 fixed    <NA>    (Intercept)    19.0      1.10      17.3  
2 fixed    <NA>      dbh        0.617     0.00757    81.5  
3 ran_pars site    sd__(Intercept)  3.35      NA        NA  
4 ran_pars Residual sd__Observation 3.04      NA        NA
```

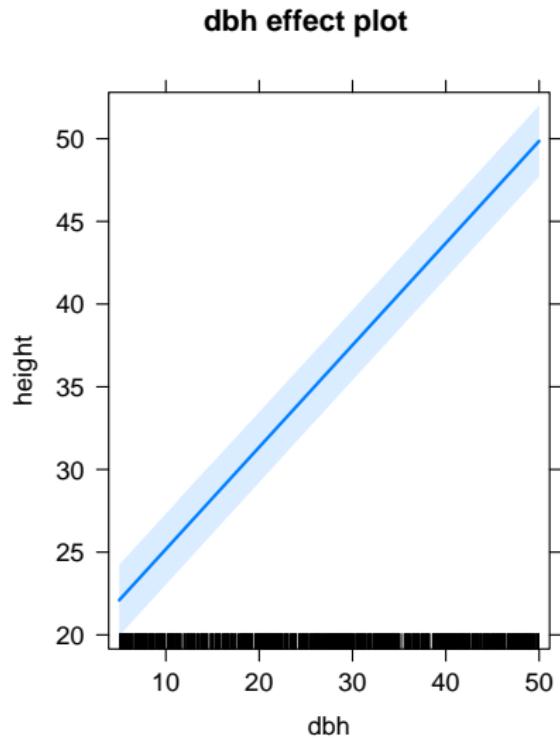
See also [broom.mixed](#)

Visualising model: allEffects

```
model: height ~ dbh
```

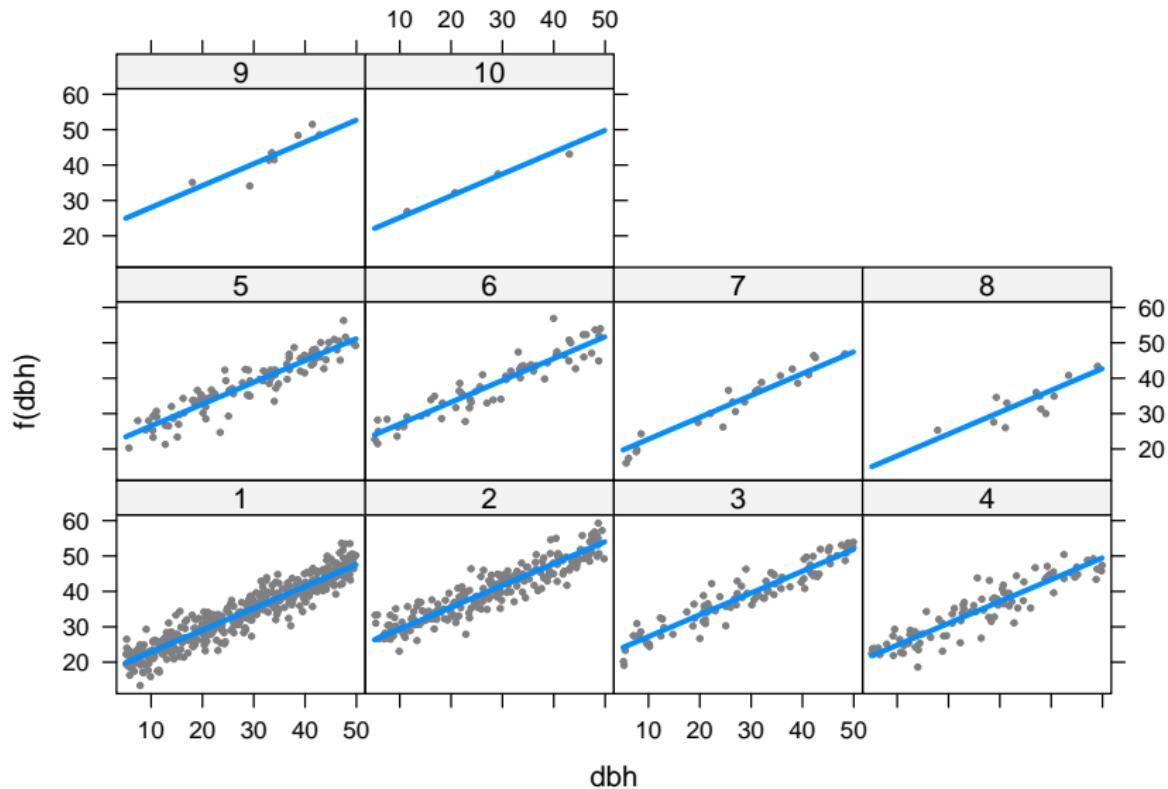
```
dbh effect  
dbh
```

5	20	30	40	50
22.09610	31.35001	37.51928	43.68855	49.85782



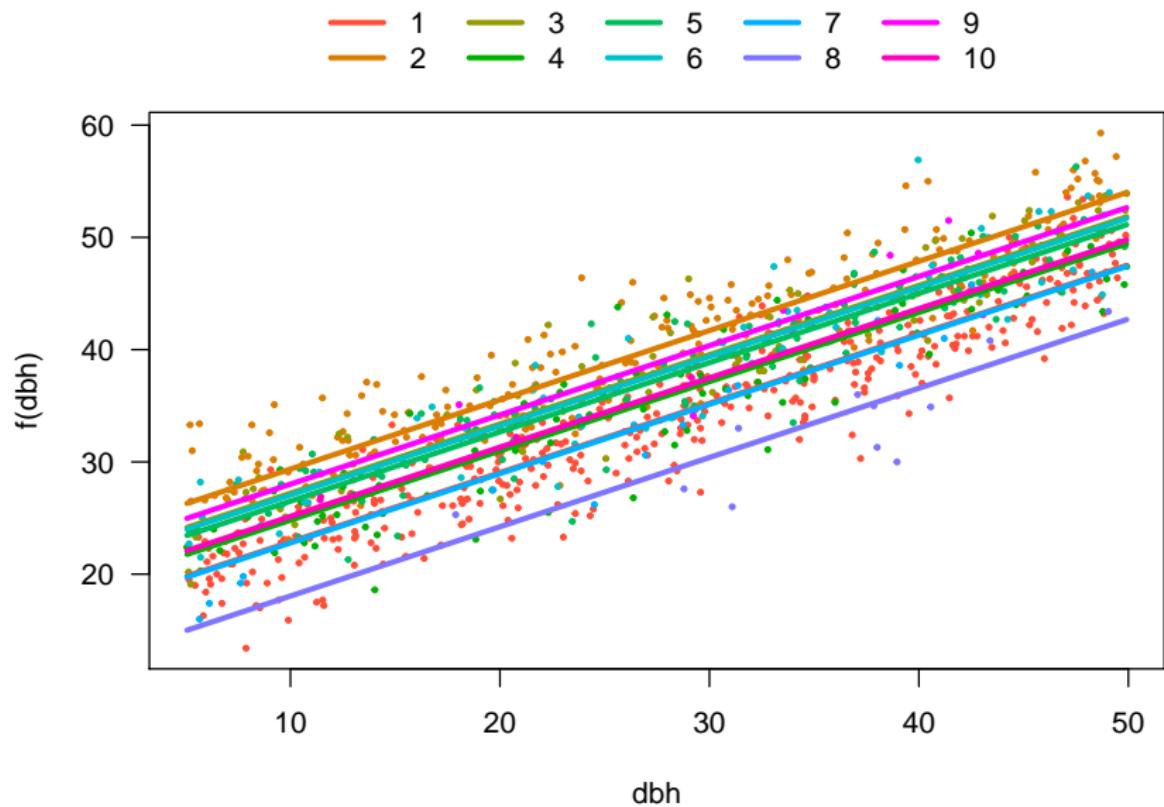
Visualising model: visreg

```
visreg(mixed, xvar = "dbh", by = "site", re.form = NULL)
```



Visualising model

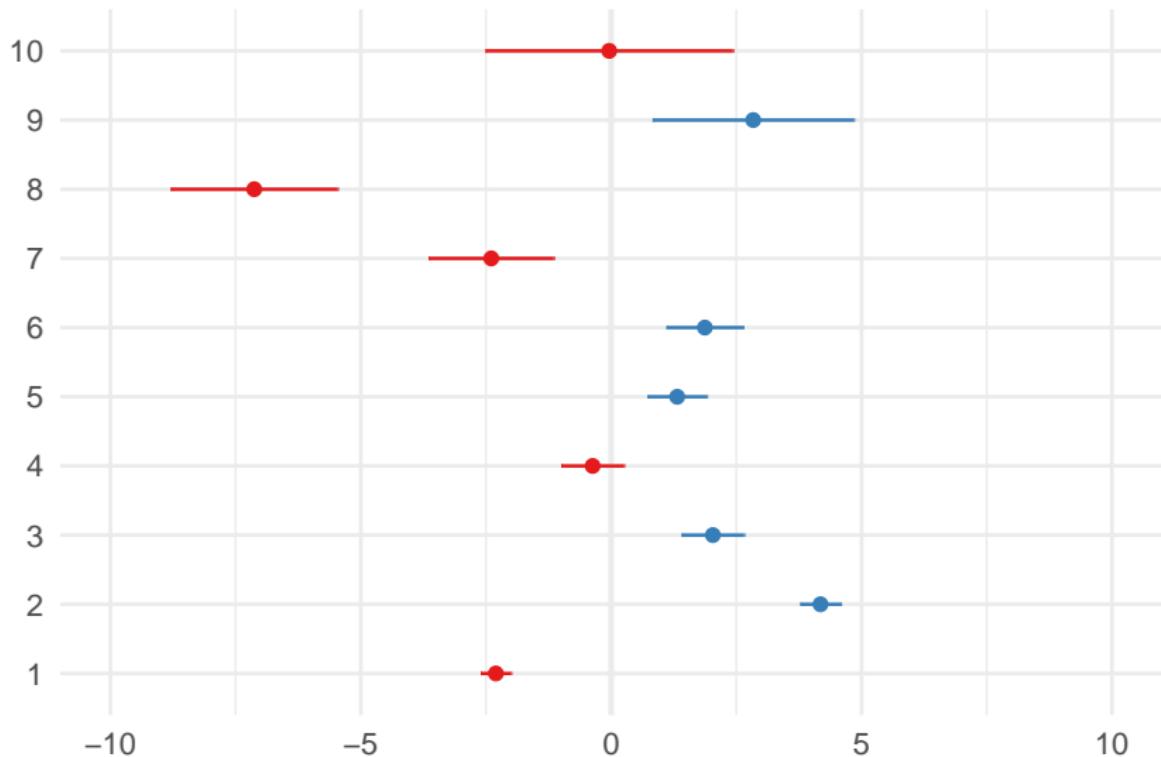
```
visreg(mixed, xvar = "dbh", by = "site", re.form = NULL, overlay
```



Visualising model: sjPlot

```
sjPlot::plot_model(mixed, type = "re")
```

Random effects

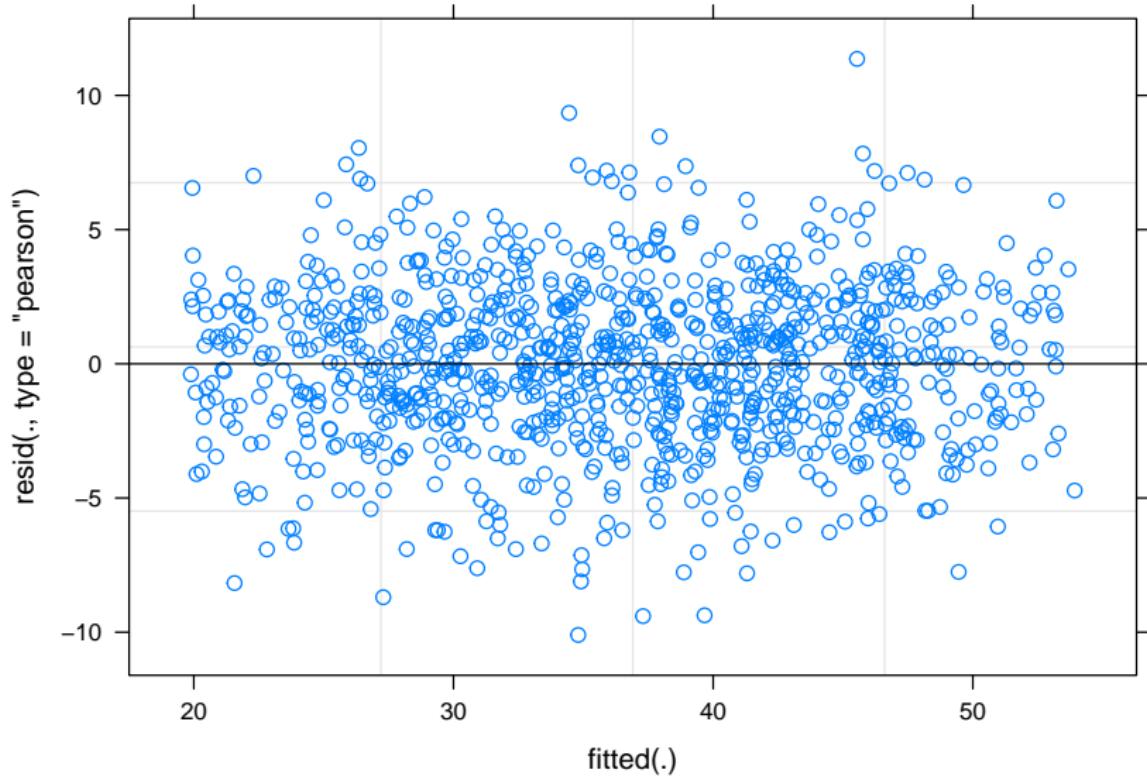


Using merTools to understand fitted model

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

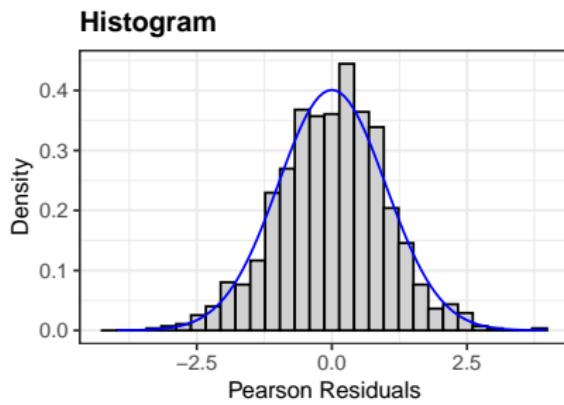
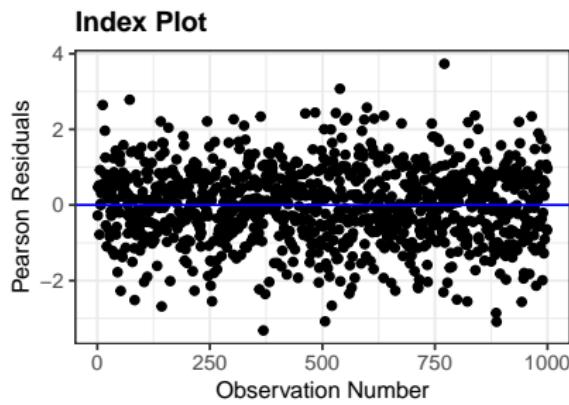
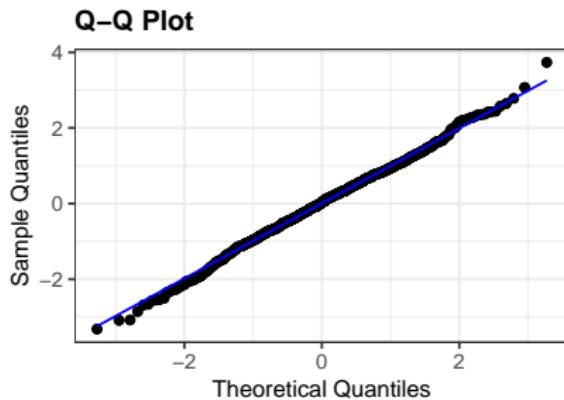
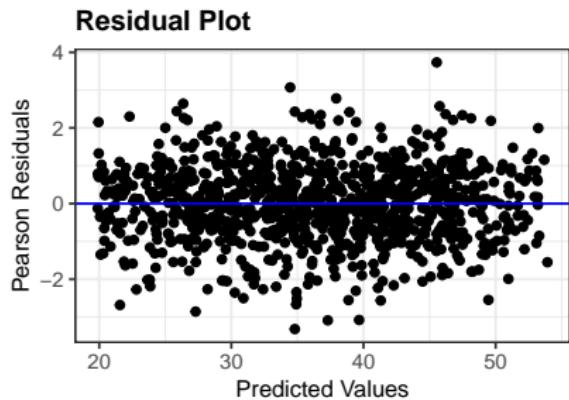
Checking residuals

```
plot(mixed)
```



Checking residuals

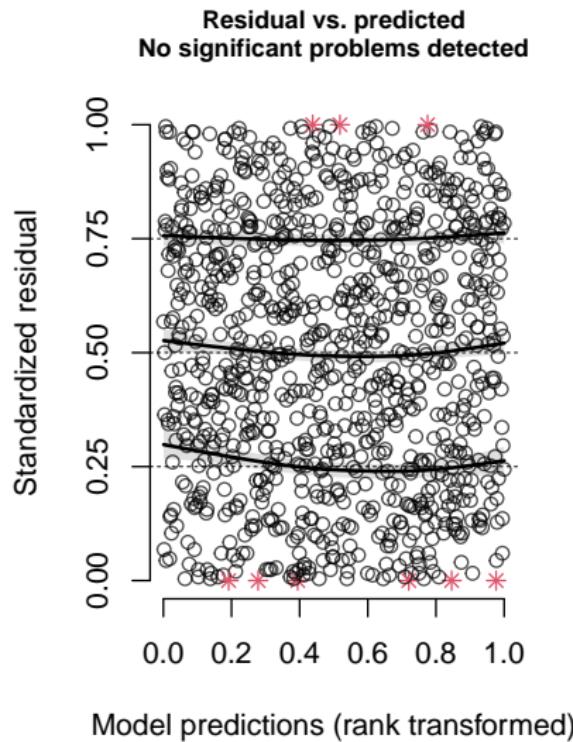
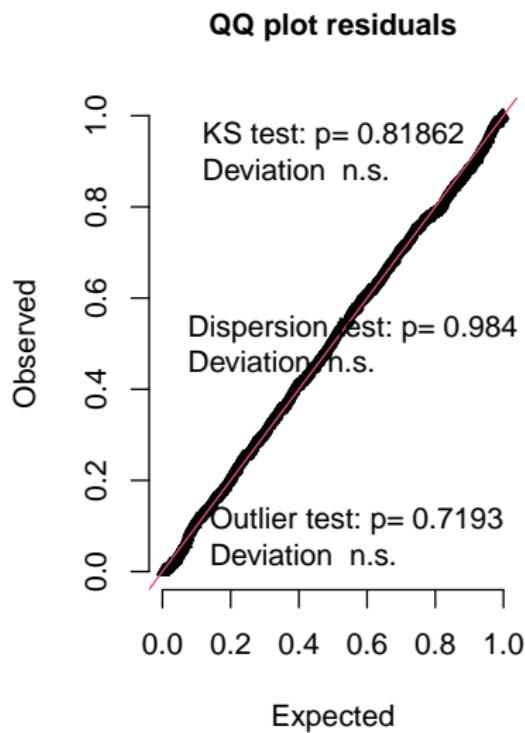
```
ggResidpanel:::resid_panel(mixed)
```



Checking residuals (DHARMA)

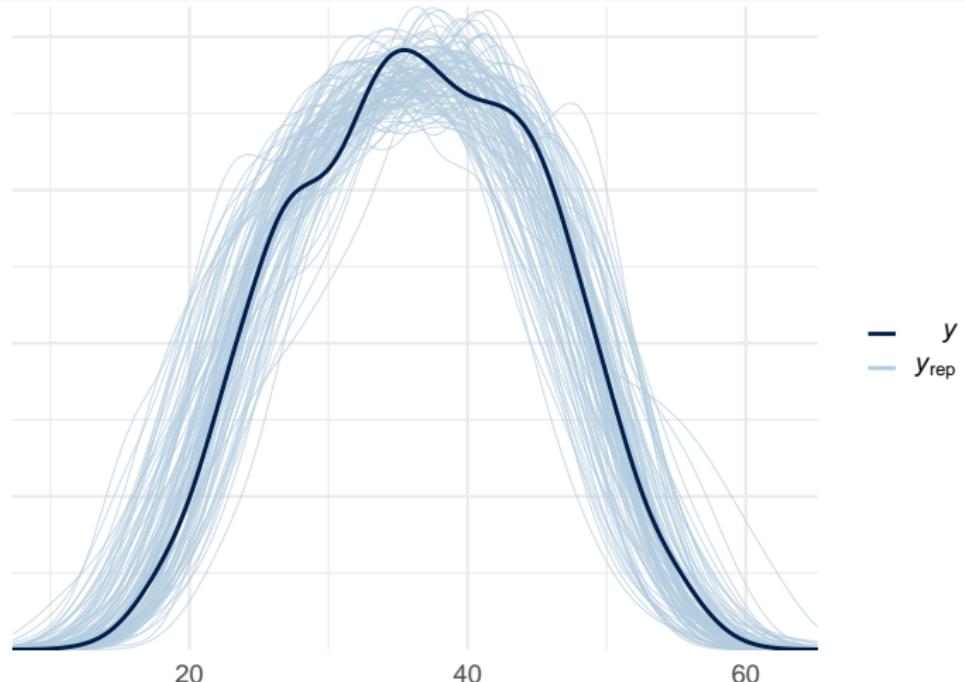
```
DHARMA:::simulateResiduals(mixed, plot = TRUE, use.u = TRUE)
```

DHARMA residual diagnostics



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 (e.g. see [this](#)).

Nakagawa & Schielzeth propose **marginal** (considering fixed effects only) and **conditional** R^2 (including random effects too):

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

	R2m	R2c
[1,]	0.752535	0.8879656

Growing the hierarchy: adding site-level
predictors

Model with group-level predictors

We had:

$$\begin{aligned}y_i &= a + \alpha_j + b \cdot x_i + \varepsilon_i \\ \alpha_j &\sim N(0, \tau^2) \\ \varepsilon_i &\sim N(0, \sigma^2)\end{aligned}$$

Now

$$\begin{aligned}y_i &= a + \alpha_j + b \cdot x_i + \varepsilon_i \\ \alpha_j &\sim N(\mu_j, \tau^2) \\ \mu_j &= \delta \cdot Predictor_j \\ \varepsilon_i &\sim N(0, \sigma^2)\end{aligned}$$

Are height differences among sites related to temperature?

$$Height_i = site_j + b \cdot DBH_i + \varepsilon_i$$

$$site_j \sim N(\mu_j, \tau^2)$$

$$\mu_j = a + \delta \cdot Temperature_j$$

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

Are height differences among sites related to temperature?

```
sitedata <- read.csv("data/sitedata.csv")
```

```
sitedata
```

	site	temp
1	1	15.1
2	2	22.0
3	3	20.1
4	4	20.4
5	5	20.0
6	6	20.1
7	7	17.5
8	8	14.6
9	9	19.2
10	10	16.0

Merging trees and site data

```
trees.full <- merge(trees, sitedata, by = "site")
head(trees.full)
```

	site	dbh	height	sex	dead	temp
1	1	21.05	32.2	male	0	15.1
2	1	46.63	45.9	female	0	15.1
3	1	43.86	45.5	male	0	15.1
4	1	29.03	35.5	male	0	15.1
5	1	6.02	21.1	male	0	15.1
6	1	40.82	38.7	male	0	15.1

Fit multilevel model

```
group.pred <- lmer(height ~ dbh + (1 | site) + temp, data = trees.full)
```

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

```
REML criterion at convergence: 5098.2
```

```
Scaled residuals:
```

Min	1Q	Median	3Q	Max
-3.3247	-0.6517	0.0192	0.6663	3.7268

```
Random effects:
```

Groups	Name	Variance	Std.Dev.
site	(Intercept)	3.158	1.777
	Residual	9.266	3.044

```
Number of obs: 1000, groups: site, 10
```

```
Fixed effects:
```

	Estimate	Std. Error	t value
(Intercept)	-1.730910	4.671330	-0.371
dbh	0.616894	0.007571	81.484
temp	1.115104	0.248000	4.496

```
Correlation of Fixed Effects:
```

(Intr)	dbh
dbh	-0.055
temp	-0.991 0.008

Too strong correlation of parameters!

Centre (and scale) continuous variables

```
mean(sitedata$temp)
```

```
[1] 18.5
```

```
trees.full$temp.c <- trees.full$temp - 18
```

Temperatures now referred as deviations from 18 °C (close to average)

Fit multilevel model

```
group.pred <- lmer(height ~ dbh + (1 | site) + temp.c, data = trees.full)
```

```
Linear mixed model fit by REML ['lmerMod']  
Formula: height ~ dbh + (1 | site) + temp.c  
Data: trees.full
```

```
REML criterion at convergence: 5098.2
```

```
Scaled residuals:
```

Min	1Q	Median	3Q	Max
-3.3247	-0.6517	0.0192	0.6663	3.7268

```
Random effects:
```

Groups	Name	Variance	Std.Dev.
site	(Intercept)	3.158	1.777
	Residual	9.266	3.044

```
Number of obs: 1000, groups: site, 10
```

```
Fixed effects:
```

	Estimate	Std. Error	t value
(Intercept)	18.340954	0.655054	27.999
dbh	0.616894	0.007571	81.484
temp.c	1.115104	0.248000	4.496

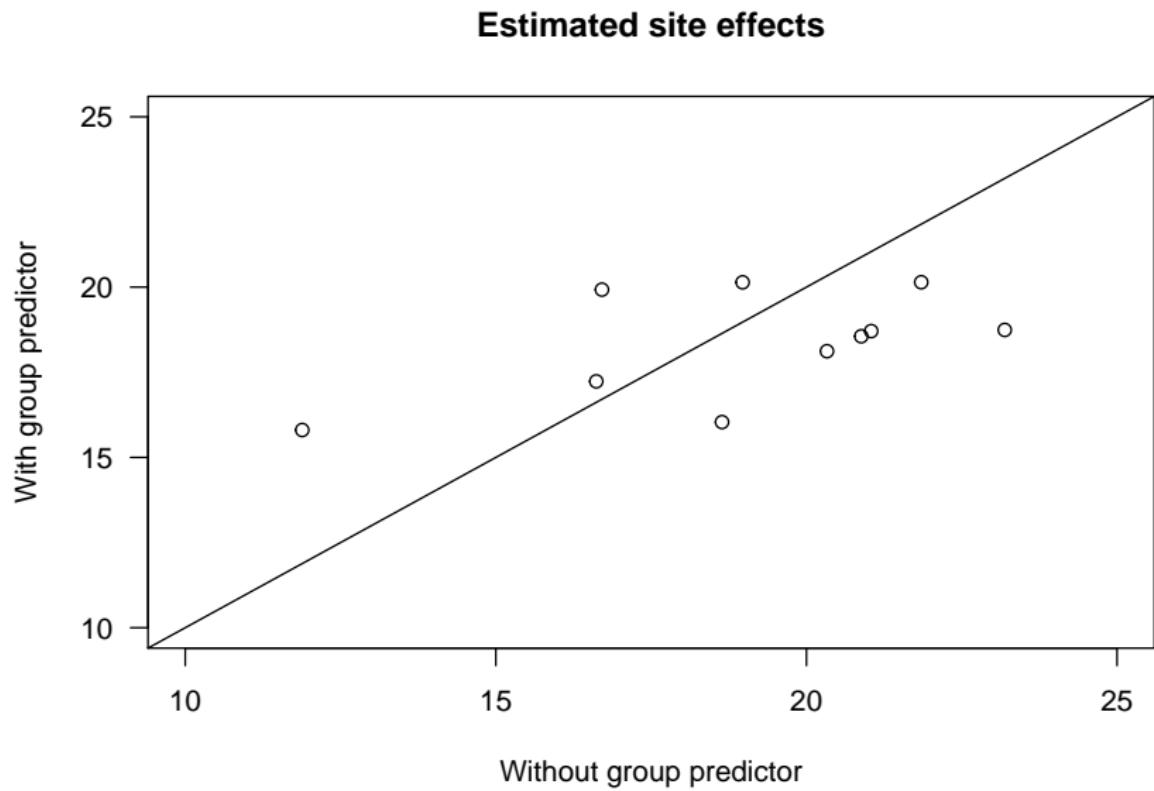
```
Correlation of Fixed Effects:
```

(Intr)	dbh
dbh	-0.333
temp.c	0.008
temp.c	-0.250

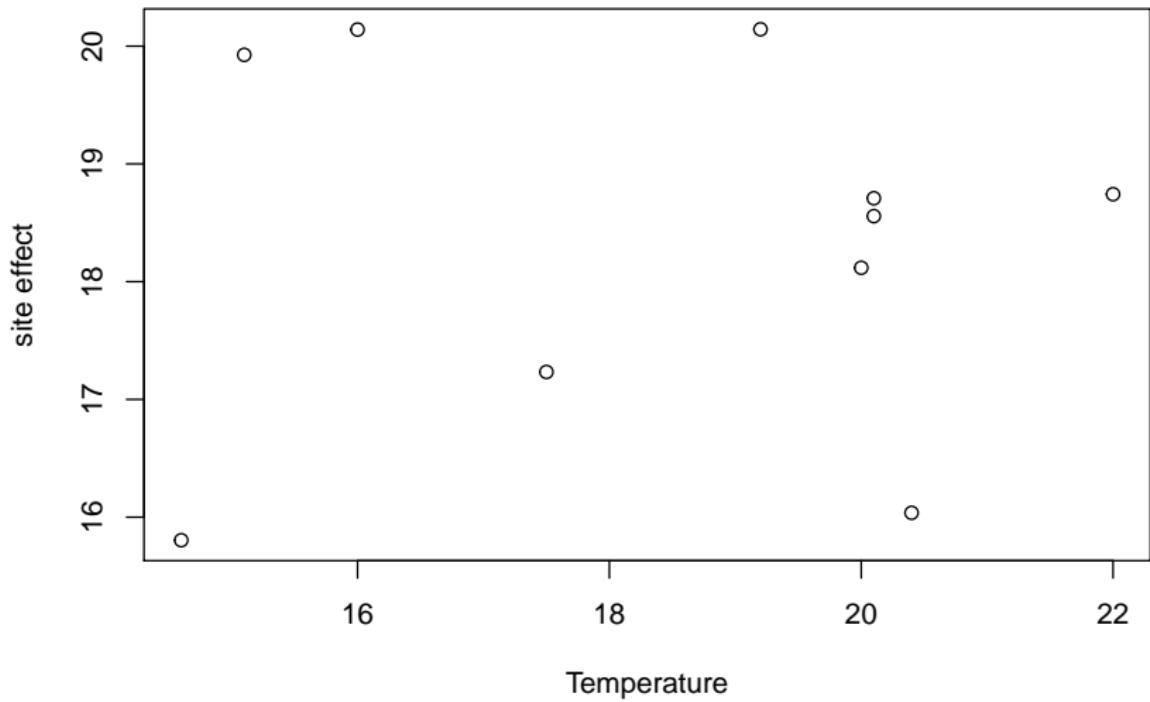
Examine model with merTools

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

Comparing site effects with and without group predictor



Are site effects related to temperature?



Varying intercepts and slopes

Varying intercepts and slopes

There is overall difference in height among sites (different intercepts)

AND

Relationship between DBH and Height varies among sites (different slopes)

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

Varying intercepts and slopes

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

REML criterion at convergence: 5105.1

Scaled residuals:

Min	1Q	Median	3Q	Max
-3.3342	-0.6599	0.0375	0.6916	3.7756

Random effects:

Groups	Name	Variance	Std.Dev.	Corr
site	(Intercept)	1.566e+01	3.95671	
	dbh	3.087e-04	0.01757	-1.00
Residual		9.226e+00	3.03744	

Number of obs: 1000, groups: site, 10

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	18.95272	1.29190	14.67
dbh	0.61837	0.00946	65.37

Correlation of Fixed Effects:

(Intr)	dbh	optimizer (nloptwrap) convergence code: 0 (OK)
	-0.722	

Varying intercepts and slopes

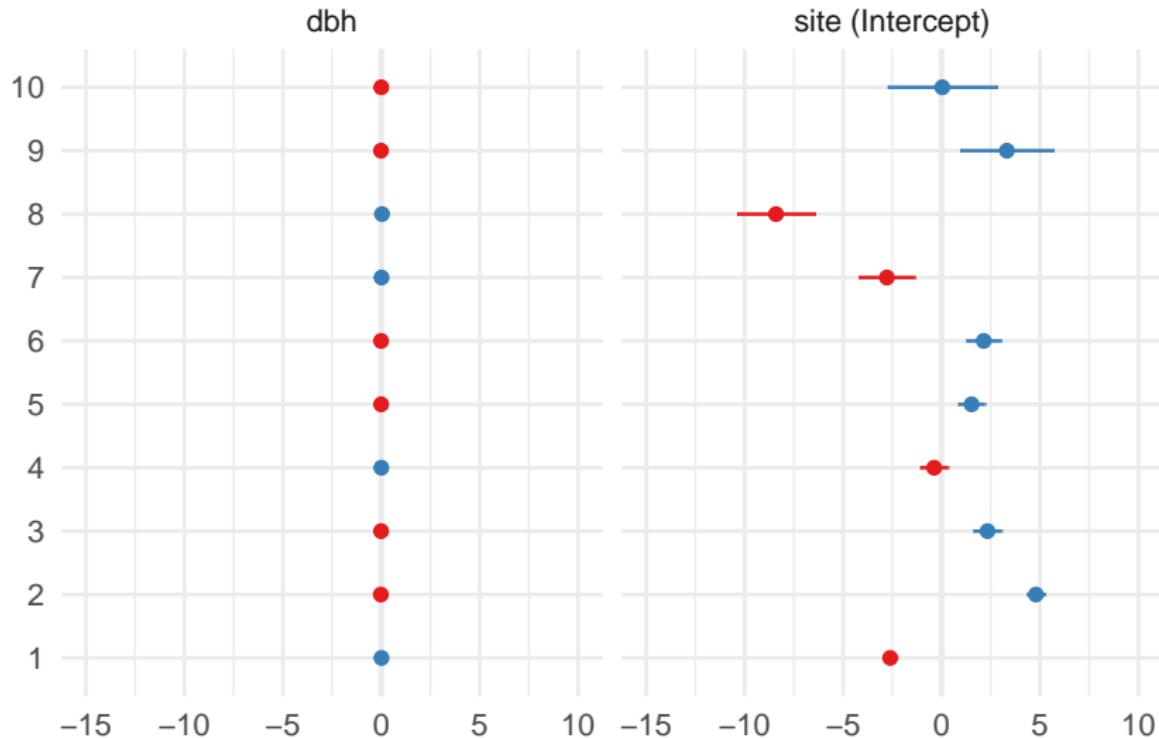
```
$site
  (Intercept)      dbh
1    16.34655 0.6299443
2    23.74733 0.5970814
3    21.28802 0.6080019
4    18.57844 0.6200337
5    20.47961 0.6115916
6    21.09608 0.6088542
7    16.17675 0.6306983
8    10.54681 0.6556978
9    22.27301 0.6036281
10   18.99463 0.6181856

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

Visualising model: sjPlot

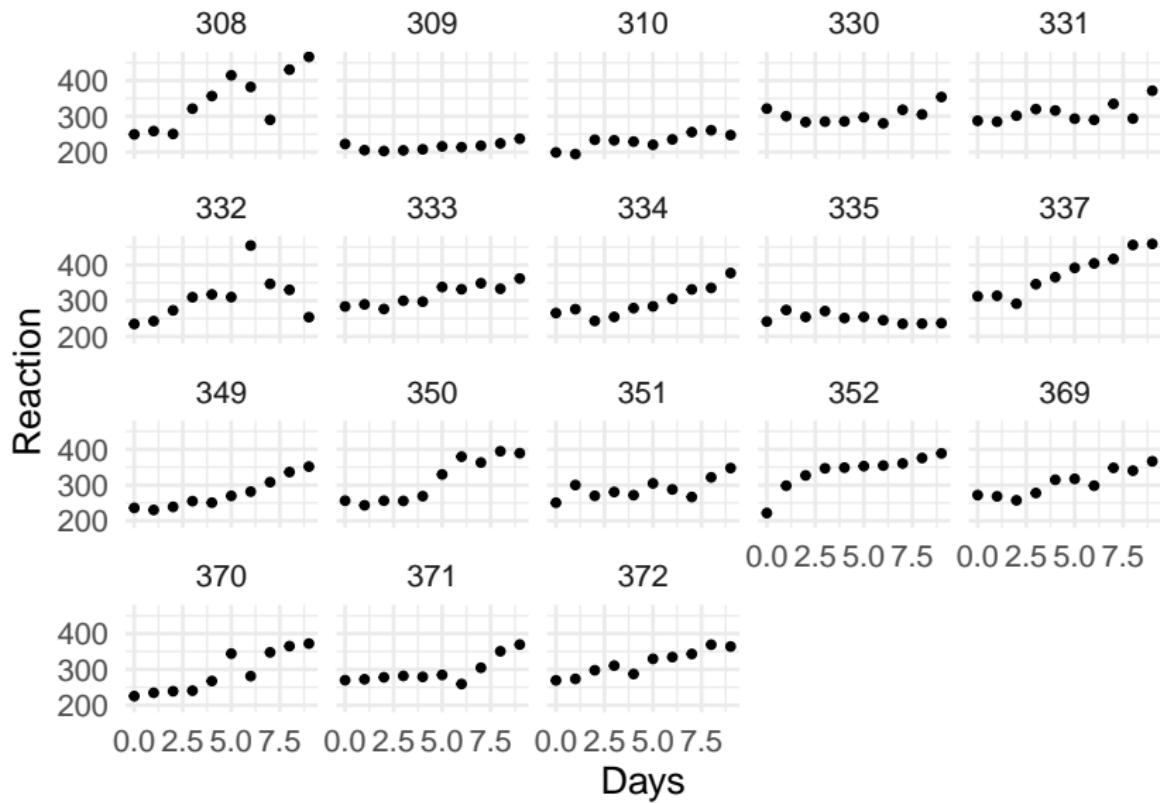
```
plot_model(mixed.slopes, type = "re")
```

Random effects



More examples

sleepstudy (repeated measures)



Varying intercepts and slopes (lme4)

```
sleep <- lmer(Reaction ~ Days + (1+Days|Subject), data = sleepstudy)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: Reaction ~ Days + (1 + Days | Subject)
Data: sleepstudy
```

```
REML criterion at convergence: 1743.6
```

```
Scaled residuals:
```

Min	1Q	Median	3Q	Max
-3.9536	-0.4634	0.0231	0.4634	5.1793

```
Random effects:
```

Groups	Name	Variance	Std.Dev.	Corr
Subject	(Intercept)	612.10	24.741	
	Days	35.07	5.922	0.07
	Residual	654.94	25.592	

```
Number of obs: 180, groups: Subject, 18
```

```
Fixed effects:
```

	Estimate	Std. Error	t value
(Intercept)	251.405	6.825	36.838
Days	10.467	1.546	6.771

```
Correlation of Fixed Effects:
```

(Intr)
Days -0.138

Varying intercepts and slopes (lme4)

```
visreg(sleep, xvar = "Days", by = "Subject", re.form = NULL)
```

Fitting multilevel models (GAMM) with mgcv

```
sgamm <- mgcv:::gam(Reaction ~ s(Days, Subject, k = 3, bs = "fs"),
                     data = sleepstudy, method = "REML")
```

Family: gaussian

Link function: identity

Formula:

Reaction ~ s(Days, Subject, k = 3, bs = "fs")

Parametric coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	295.22	10.49	28.15	<2e-16 ***

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

Approximate significance of smooth terms:

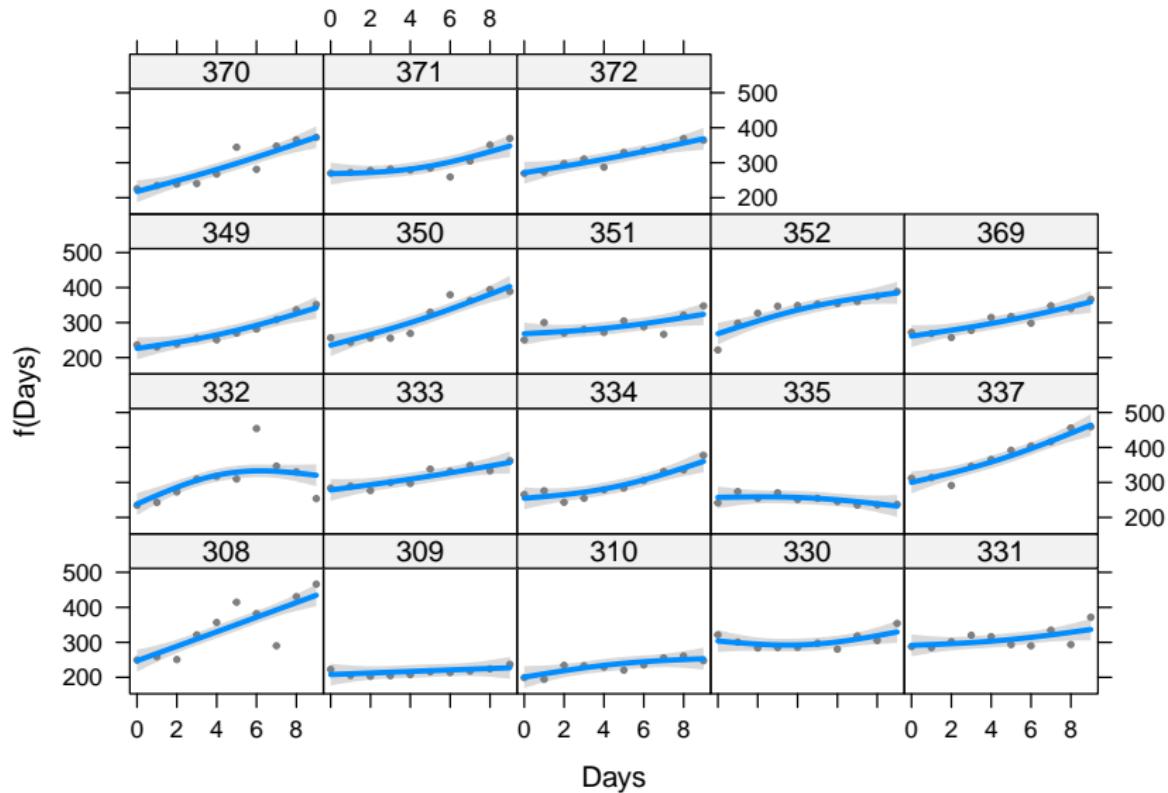
	edf	Ref.df	F	p-value
s(Days,Subject)	42.2	53	16.05	<2e-16 ***

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

R-sq.(adj) = 0.826 Deviance explained = 86.7%

Fitting multilevel models (GAMM) with mgcv

```
visreg(sgamm, xvar = "Days", by = "Subject")
```



Fitting multilevel models (GAMM) with `mgcv`

Hierarchical generalized additive models: an introduction with `mgcv`

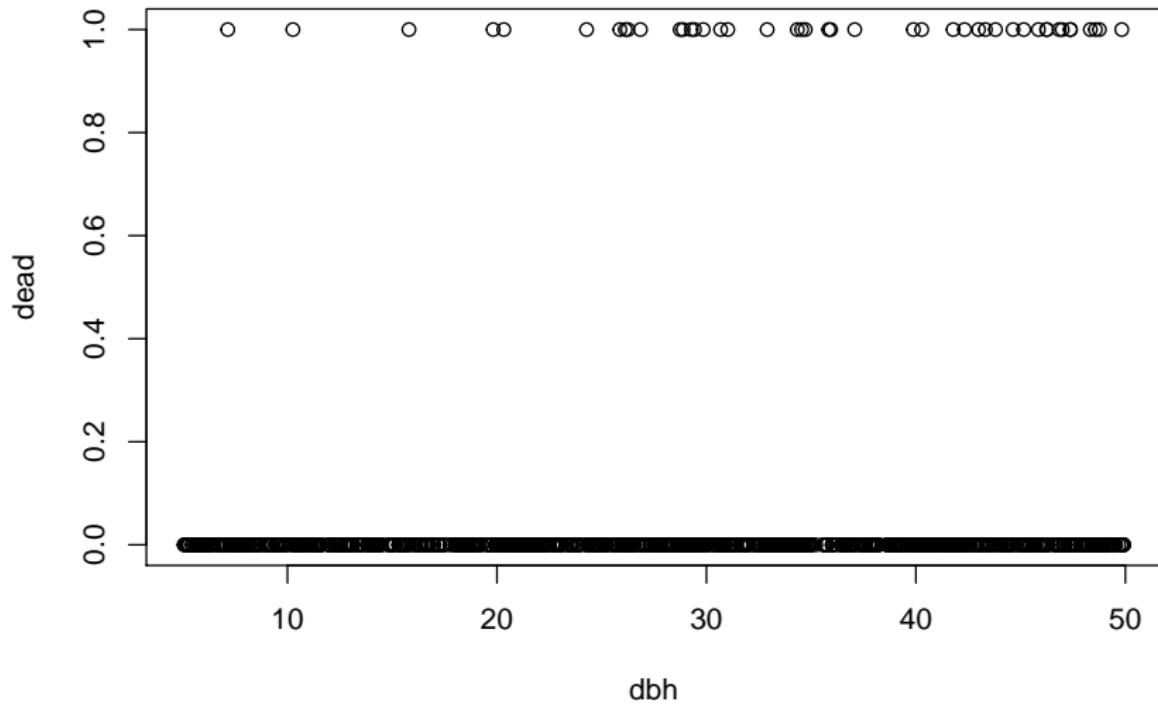
Eric J Pedersen ^{Corresp.} ^{1,2}, David L. Miller ^{3,4}, Gavin L. Simpson ⁵, Noam Ross ⁶

<https://doi.org/10.7287/peerj.preprints.27320v1>

Multilevel logistic regression

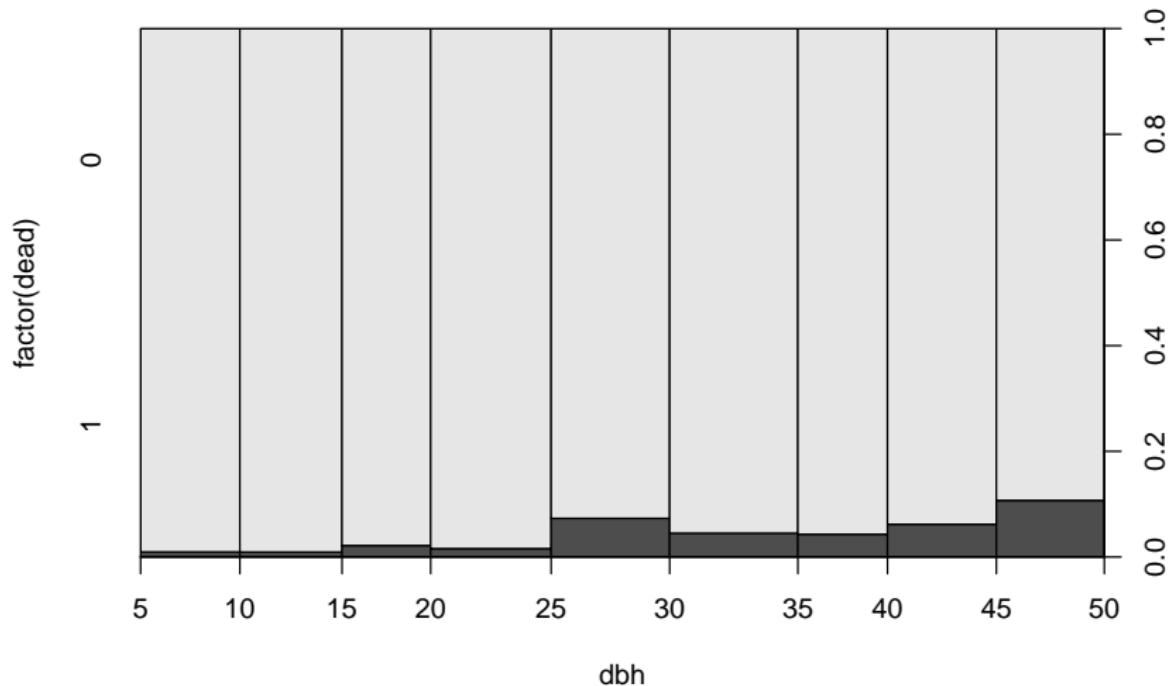
Q: Relationship between tree size and mortality

```
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
-0.4805	-0.3520	-0.2647	-0.1928	2.9690

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)		
(Intercept)	-4.77874	0.50902	-9.388	< 2e-16 ***		
dbh	0.05365	0.01377	3.895	9.82e-05 ***		

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

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 360.91 on 999 degrees of freedom

Residual deviance: 343.69 on 998 degrees of freedom

Logistic regression with *independent* site effects

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

Call:

```
glm(formula = dead ~ dbh + factor(site), family = binomial, data = trees)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-0.6359	-0.3449	-0.2561	-0.1852	2.9763

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-4.80123	0.54985	-8.732	<2e-16	***
dbh	0.05371	0.01381	3.889	0.0001	***
factor(site)2	-0.29692	0.46073	-0.644	0.5193	
factor(site)3	0.21275	0.52799	0.403	0.6870	
factor(site)4	0.39841	0.53025	0.751	0.4524	
factor(site)5	-0.42557	0.64018	-0.665	0.5062	
factor(site)6	0.66861	0.53656	1.246	0.2127	
factor(site)7	0.11862	1.06211	0.112	0.9111	
factor(site)8	0.43899	1.08058	0.406	0.6846	
factor(site)9	-13.63389	840.90382	-0.016	0.9871	
factor(site)10	-13.17148	1042.21823	-0.013	0.9899	

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

(R version 3.6.1 (2019-07-05) -- "Just in Time")

Fit multilevel logistic regression

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

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

      AIC      BIC      logLik deviance df.resid
349.7    364.4    -171.8     343.7      997

Scaled residuals:
    Min      1Q  Median      3Q      Max
-0.3498 -0.2528 -0.1888 -0.1370  9.0031

Random effects:
Groups Name        Variance Std.Dev.
site   (Intercept) 0         0
Number of obs: 1000, groups: site, 10

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.77874   0.50904 -9.388 < 2e-16 ***
dbh         0.05365   0.01377  3.895 9.83e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Retrieve model coefficients

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

```
$site
  (Intercept)      dbh
1 -4.778744 0.05364989
2 -4.778744 0.05364989
3 -4.778744 0.05364989
4 -4.778744 0.05364989
5 -4.778744 0.05364989
6 -4.778744 0.05364989
7 -4.778744 0.05364989
8 -4.778744 0.05364989
9 -4.778744 0.05364989
10 -4.778744 0.05364989
```

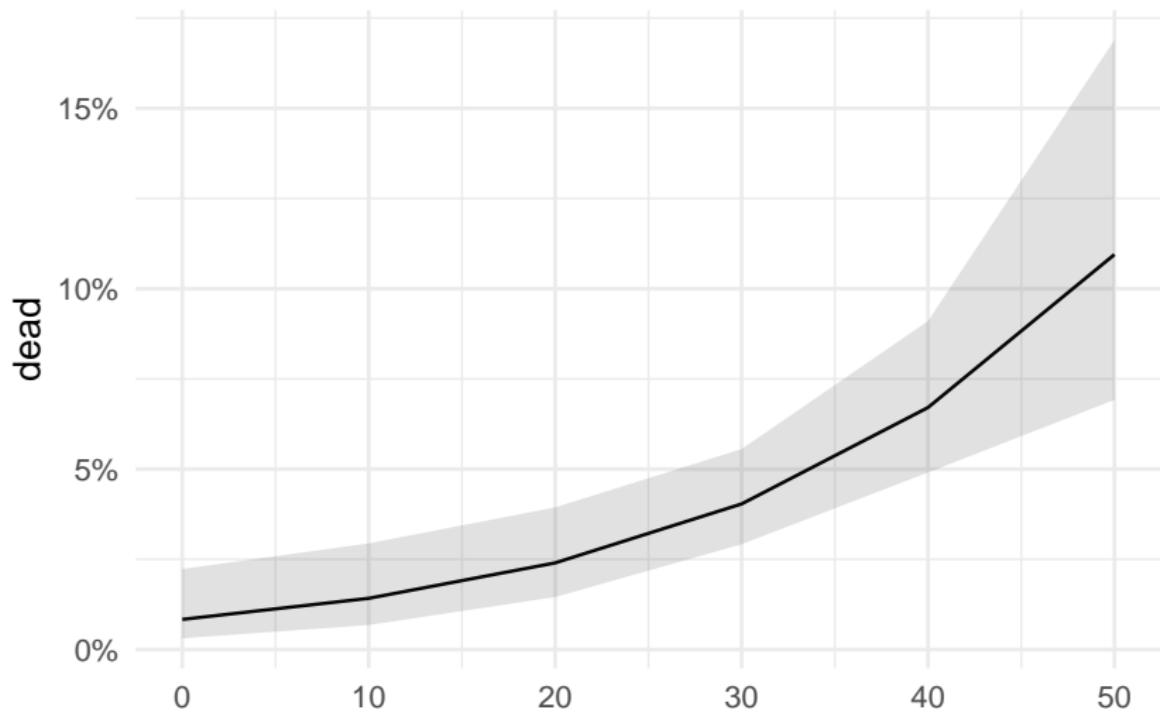
```
attr(,"class")
[1] "coef.mer"
```

Visualising model: sjPlot

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

\$dbh

Predicted probabilities of dead



Poisson multilevel regression

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

- ▶ Varying intercepts

Formula syntax for different models

- ▶ Varying intercepts
 - ▶ $y \sim x + (1 | \text{group})$

Formula syntax for different models

- ▶ Varying intercepts
 - ▶ $y \sim x + (1 | \text{group})$
- ▶ Varying intercepts and slopes

Formula syntax for different models

- ▶ Varying intercepts
 - ▶ $y \sim x + (1 \mid \text{group})$
- ▶ Varying intercepts and slopes
 - ▶ $y \sim x + (1 + x \mid \text{group})$

Formula syntax for different models

- ▶ Varying intercepts
 - ▶ $y \sim x + (1 \mid \text{group})$
- ▶ Varying intercepts and slopes
 - ▶ $y \sim x + (1 + x \mid \text{group})$
- ▶ Varying intercepts, 2 groups (crossed)

Formula syntax for different models

- ▶ Varying intercepts
 - ▶ $y \sim x + (1 \mid \text{group})$
- ▶ Varying intercepts and slopes
 - ▶ $y \sim x + (1 + x \mid \text{group})$
- ▶ Varying intercepts, 2 groups (crossed)
 - ▶ $y \sim x + (1 \mid \text{group1}) + (1 \mid \text{group2})$

Formula syntax for different models

- ▶ Varying intercepts
 - ▶ $y \sim x + (1 \mid \text{group})$
- ▶ Varying intercepts and slopes
 - ▶ $y \sim x + (1 + x \mid \text{group})$
- ▶ Varying intercepts, 2 groups (crossed)
 - ▶ $y \sim x + (1 \mid \text{group1}) + (1 \mid \text{group2})$
- ▶ Varying intercepts, 2 groups (nested)

Formula syntax for different models

- ▶ Varying intercepts
 - ▶ $y \sim x + (1 \mid \text{group})$
- ▶ Varying intercepts and slopes
 - ▶ $y \sim x + (1 + x \mid \text{group})$
- ▶ Varying intercepts, 2 groups (crossed)
 - ▶ $y \sim x + (1 \mid \text{group1}) + (1 \mid \text{group2})$
- ▶ Varying intercepts, 2 groups (nested)
 - ▶ $y \sim x + (1 \mid \text{group/subgroup})$

Formula syntax for different models

- ▶ Varying intercepts
 - ▶ $y \sim x + (1 \mid \text{group})$
- ▶ Varying intercepts and slopes
 - ▶ $y \sim x + (1 + x \mid \text{group})$
- ▶ Varying intercepts, 2 groups (crossed)
 - ▶ $y \sim x + (1 \mid \text{group1}) + (1 \mid \text{group2})$
- ▶ Varying intercepts, 2 groups (nested)
 - ▶ $y \sim x + (1 \mid \text{group/subgroup})$
 - ▶ This is **equivalent** to $y \sim x + (1 \mid \text{group1}) + (1 \mid \text{group2})$ with distinct labelling of group levels.

Formula syntax for different models

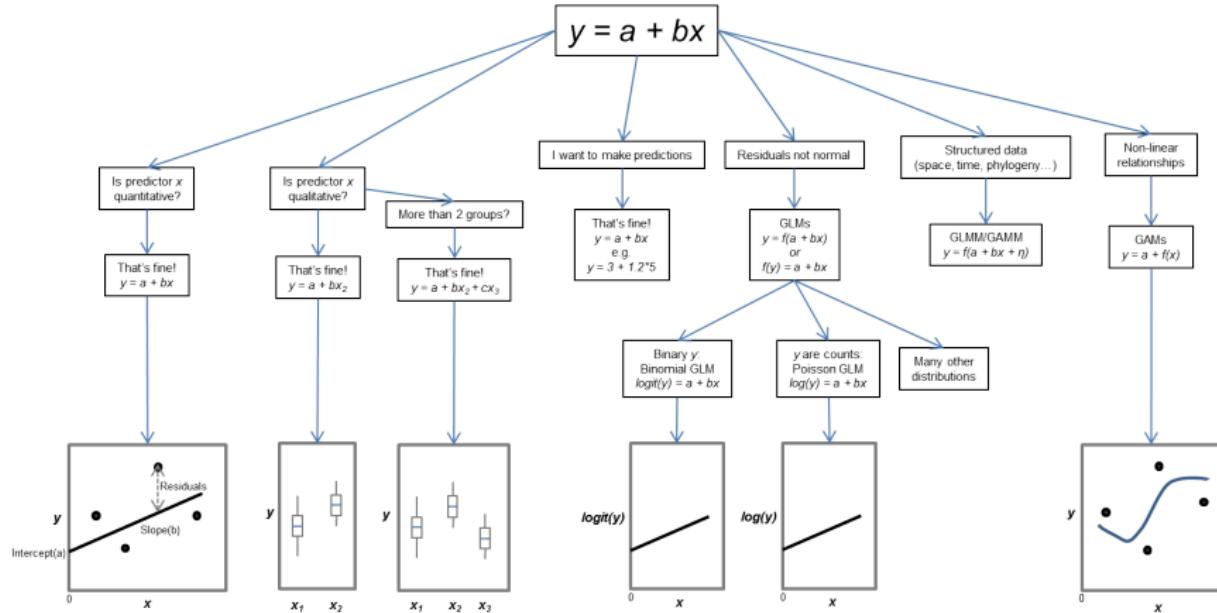
- ▶ Varying intercepts
 - ▶ $y \sim x + (1 \mid \text{group})$
- ▶ Varying intercepts and slopes
 - ▶ $y \sim x + (1 + x \mid \text{group})$
- ▶ Varying intercepts, 2 groups (crossed)
 - ▶ $y \sim x + (1 \mid \text{group1}) + (1 \mid \text{group2})$
- ▶ Varying intercepts, 2 groups (nested)
 - ▶ $y \sim x + (1 \mid \text{group/subgroup})$
 - ▶ This is **equivalent** to $y \sim x + (1 \mid \text{group1}) + (1 \mid \text{group2})$ with distinct labelling of group levels.
- ▶ Varying intercepts and slopes, 2 groups (crossed)

Formula syntax for different models

- ▶ Varying intercepts
 - ▶ $y \sim x + (1 \mid \text{group})$
- ▶ Varying intercepts and slopes
 - ▶ $y \sim x + (1 + x \mid \text{group})$
- ▶ Varying intercepts, 2 groups (crossed)
 - ▶ $y \sim x + (1 \mid \text{group1}) + (1 \mid \text{group2})$
- ▶ Varying intercepts, 2 groups (nested)
 - ▶ $y \sim x + (1 \mid \text{group/subgroup})$
 - ▶ This is **equivalent** to $y \sim x + (1 \mid \text{group1}) + (1 \mid \text{group2})$ with distinct labelling of group levels.
- ▶ Varying intercepts and slopes, 2 groups (crossed)
 - ▶ $y \sim x + (1 + x \mid \text{group1}) + (1 + x \mid \text{group2})$

GLMM FAQ

<https://bbolker.github.io/mixedmodels-misc/glmmFAQ.html>



END



Source code and materials:

<https://github.com/Pakillo/LM-GLM-GLMM-intro>