

# Linear, Generalized, and Mixed/Multilevel models in R

---

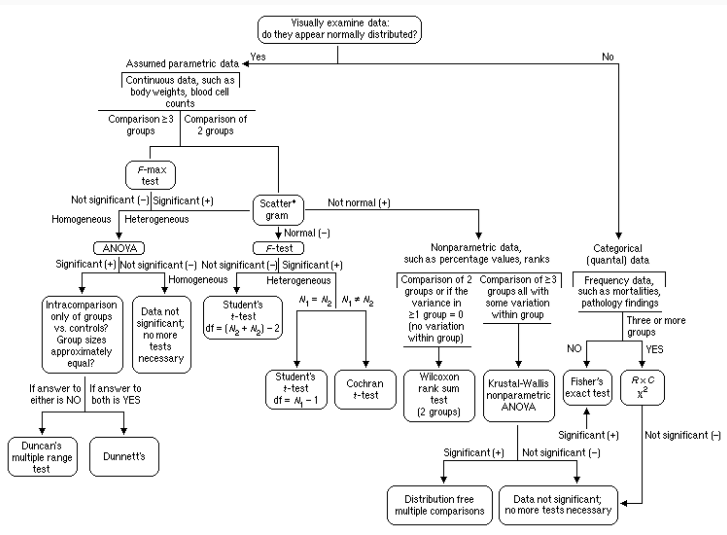
Francisco Rodríguez-Sánchez

<https://frodriguezsanchez.net>

## GLM as unified framework for data analysis

---

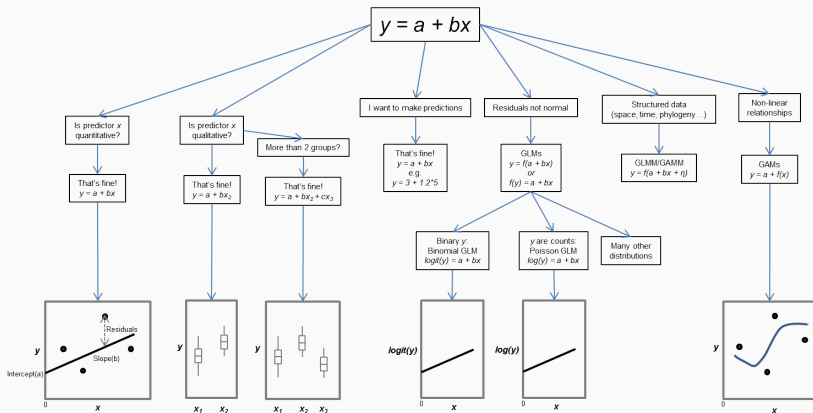
# Statistics as a maze



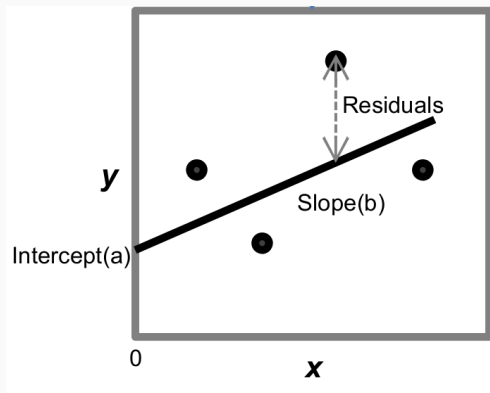
## So many questions

- **Why** should we really use analysis Y over Z?
- What if my data are **not Normal**?
- What if they are **not independent**?
- Why am I getting **different p-values** with different tests?
- What even is a **p-value**?
- How can I take **different factors** into account?
- Can I make **predictions**?

# A unified framework

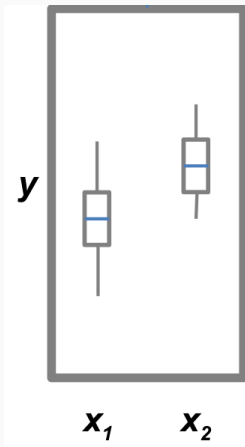


$$y = a + bx$$



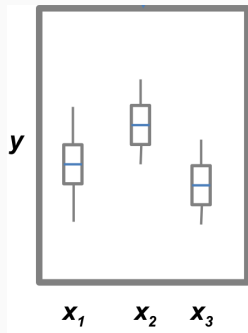
# Is predictor X qualitative?

$$y = a + bx_2$$

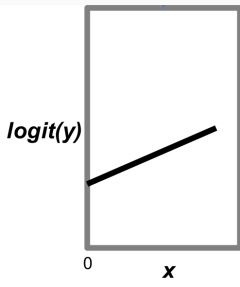


## More than 2 groups?

$$y = a + bx_2 + cx_3$$



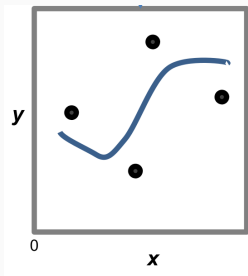
$$y = f(a + bx)$$



$$y = f(a + bx + \eta)$$

## Relationships are not linear

$$y = a + f(x)$$



t-tests

ANOVA

regression

...

are special cases of GLM

# Common statistical tests are linear models

Last updated: 02 April, 2019

See worked examples and more details at the accompanying notebook: <https://lindeloev.github.io/tests-as-linear>

	Common name	Built-in function in R	Equivalent linear model in R	Exact?	The linear model in words	Icon
Simple regression: $lm(y \sim 1 + x)$	<b>y is independent of x</b> P: One-sample t-test N: Wilcoxon signed-rank	t.test(y) wilcox.test(y)	$lm(y \sim 1)$ $lm(\text{signed\_rank}(y) \sim 1)$	✓ for $N \geq 14$	One number (intercept, i.e., the mean) predicts <b>y</b> . - (Same, but it predicts the signed rank of <b>y</b> .)	
	P: Paired-sample t-test N: Wilcoxon matched pairs	t.test(y1, y2, paired=TRUE) wilcox.test(y1, y2, paired=TRUE)	$lm(y2 \sim y1 \sim 1)$ $lm(\text{signed\_rank}(y2 - y1) \sim 1)$	✓ for $N \geq 14$	One intercept predicts the pairwise <b>y<sub>2</sub>-y<sub>1</sub></b> differences. - (Same, but it predicts the signed rank of <b>y<sub>2</sub>-y<sub>1</sub></b> .)	
	<b>y ~ continuous x</b> P: Pearson correlation N: Spearman correlation	cor.test(x, y, method='Pearson') cor.test(x, y, method='Spearman')	$lm(y \sim 1 + x)$ $lm(\text{rank}(y) \sim 1 + \text{rank}(x))$	✓ for $N \geq 10$	One intercept plus <b>x</b> multiplied by a number (slope) predicts <b>y</b> . - (Same, but with ranked <b>x</b> and <b>y</b> .)	
	<b>y ~ discrete x</b> P: Two-sample t-test P: Welch's t-test N: Mann-Whitney U	t.test(y1, y2, var.equal=TRUE) t.test(y1, y2, var.equal=FALSE) wilcox.test(y1, y2)	$lm(y \sim 1 + G_i)^2$ $glm(y \sim 1 + G_i, weights = \dots^y)$ $lm(\text{signed\_rank}(y) \sim 1 + G_i)^2$	✓ ✓ for $N \geq 11$	An intercept for <b>group 1</b> (plus a difference if <b>group 2</b> ) predicts <b>y</b> . - (Same, but with one variance pair group instead of one common.) - (Same, but it predicts the signed rank of <b>y</b> .)	
Multiple regression: $lm(y \sim 1 + x_1 + x_2 + \dots)$	P: One-way ANOVA N: Kruskal-Wallis	aov(y ~ group) kruskal.test(y ~ group)	$lm(y \sim 1 + G_1 + G_2 + \dots + G_n)^2$ $lm(\text{rank}(y) \sim 1 + G_1 + G_2 + \dots + G_n)^2$	✓ for $N \geq 11$	An intercept for <b>group 1</b> (plus a difference if group $\neq 1$ ) predicts <b>y</b> . - (Same, but it predicts the rank of <b>y</b> .)	
	P: One-way ANCOVA	aov(y ~ group + x)	$lm(y \sim 1 + G_1 + G_2 + \dots + G_n + x)^2$	✓	- (Same, but plus a slope on <b>x</b> .) <i>Note: this is discrete AND continuous. ANCOVAs are ANOVAs with a continuous x.</i>	
	P: Two-way ANOVA	aov(y ~ group * sex)	$lm(y \sim 1 + G_1 + G_2 + \dots + G_n + S_1 + S_2 + \dots + S_k + G_1^*S_1 + G_1^*S_2 + \dots + G_n^*S_k)$	✓	Interaction term: changing <b>sex</b> changes the <b>y - group</b> parameters. <i>Note: <math>G_{i,j}</math> is an indicator (0 or 1) for each non-intercept levels of the group variable. Similarly for <math>S_{k,m}</math> for sex. The first line (with <math>G_i</math>) is main effect of group, the second (with <math>S_j</math>) for sex and the third is the group * sex interaction. For two levels (e.g. male/female), line 2 would just be "<math>S_1</math>" and line 3 would be <math>S_1</math> multiplied with each <math>G_i</math>.</i>	[Coming]
Multiple regression: $lm(y \sim 1 + x_1 + x_2 + \dots)$	<b>Counts ~ discrete x</b> N: Chi-square test	chisq.test(groupXsex_table)	<b>Equivalent log-linear model</b> $glm(y \sim 1 + G_1 + G_2 + \dots + G_n + S_1 + S_2 + \dots + S_k + G_1^*S_1 + G_1^*S_2 + \dots + G_n^*S_k, \text{family} = \dots)^2$	✓	Interaction term: (Same as Two-way ANOVA.) <i>Note: Run glm using the following arguments: glm(model, family=poisson())</i> <i>As linear-model, the Chi-square test is <math>\log(y) = \log(\beta) + \log(\alpha) + \log(\beta) + \log(\beta)</math> where <math>\alpha</math> and <math>\beta</math> are proportions. See more info in the accompanying notebook.</i>	Same as Two-way ANOVA
	N: Goodness of fit	chisq.test(y)	$glm(y \sim 1 + G_1 + G_2 + \dots + G_n, \text{family} = \dots)^2$	✓	(Same as One-way ANOVA and see Chi-Square note.)	1W-ANOVA

List of common parametric (P) non-parametric (N) tests and equivalent linear models. The notation  $y \sim 1 + x$  is R shorthand for  $y = 1 + b + a \cdot x$  which most of us learned in school. Models in similar colors are highly similar, but really, notice how similar they all are across colors! For non-parametric models, the linear models are reasonable approximations for non-small sample sizes (see "Exact" column and click links to see simulations). Other less accurate approximations exist, e.g., Wilcoxon for the sign test and Goodness-of-fit for the binomial test. The signed rank function is `function(x) sign(x) * rank(abs(x))`. The variables  $G_i$  and  $S_k$  are "dummy coded" indicator variables (either 0 or 1) exploiting the fact that when  $\Delta x = 1$  between categories the difference equals the slope. Subscripts (e.g.,  $G_2$  or  $y_1$ ) indicate different columns in data. `lm` requires long-format data for all non-continuous models. All of this is exposed in greater detail and worked examples at <https://lindeloev.github.io/tests-as-linear>.

<sup>1</sup> See the note to the two-way ANOVA for explanation of the notation.

<sup>2</sup> Same model, but with one variance per group: `glm(value ~ 1 + G_i, weights = varIdent(form = ~1|group), method="ML")`.



Jonas Kristoffer Lindeløv  
<https://lindeloev.net>

<https://lindeloev.github.io/tests-as-linear/>

With GLM we can analyse  
many different types of data  
using many predictors  
(quantitative & qualitative)

**Unified, coherent framework** for data analysis with many extensions:

- **GLMM** (mixed models): accommodate data structure & variation (space, time, phylogeny)
- **GAMM** (generalised additive models): non-linear relationships
- **Model-based multivariate** statistics
- **Bayesian** modelling

The Generalised Linear Model (GLM) is a particularly reasonable vantage point on statistical analyses, as **many tests and procedures are special cases** of the GLM. The downside of that (and any other) vantage point is that **we first have to climb it**. There are the morass of unfamiliar terminology, the scree slopes of probability and the cliffs of distributions. **The vista, however, is magnificent**. From the GLM, t-test, ANOVA and regression neatly arrange themselves into regular patterns, and we can see the paths leading towards the horizon: to time series analyses, Bayesian statistics, spatial statistics and so forth.

[Dormann 2020](#)

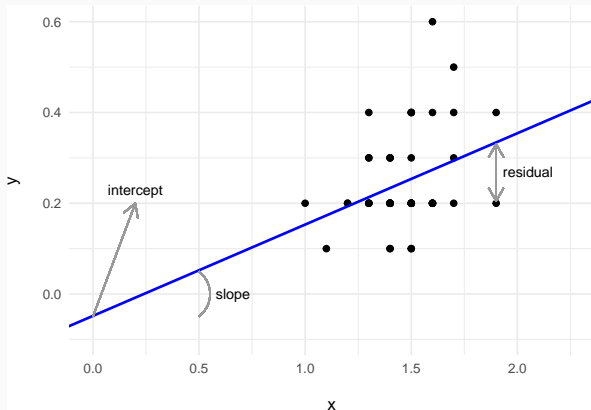
# Introduction to linear models

---

# Our unified regression framework (GLM)

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

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



## Data

$y$  = response variable

$x$  = predictor

## Parameters

$a$  = intercept

$b$  = slope

$\sigma$  = residual variation

$\varepsilon$  = residuals

## What's the intercept?

Expected value of  $y$  when predictors ( $x$ ) = 0

If  $x = 0$ :

- $y = a + b \cdot 0$

- $y = a$

## What's the slope?

How much **y** increases (or decreases) when **x** increases in 1 unit

If we have model

$$y = 0.5 + 2 \cdot x$$

If **x** increases 1 unit, **y** increases **2 units**

- If  $x = 10 \rightarrow y = 0.5 + 2 * 10 = 20.5$
- If  $x = 11 \rightarrow y = 0.5 + 2 * 11 = 22.5$

## Slopes can be negative

If we have model

$$y = 0.5 - 2x$$

If  $x$  increases 1 unit,  $y$  decreases 2 units

- If  $x = 10 \rightarrow y = 0.5 - 2 * 10 = -19.5$
- If  $x = 11 \rightarrow y = 0.5 - 2 * 11 = -21.5$



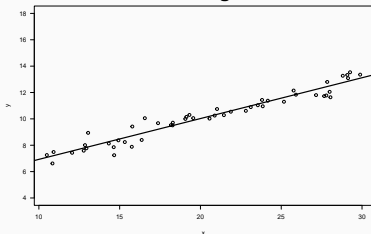
# If sigma is large, residuals are larger

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

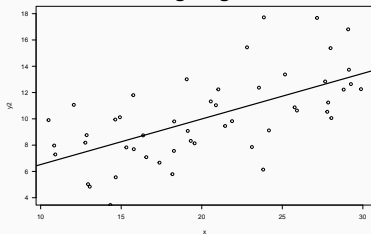
If sigma is larger:

- points farther from regression line
- larger difference of observed - predicted values

**small sigma**



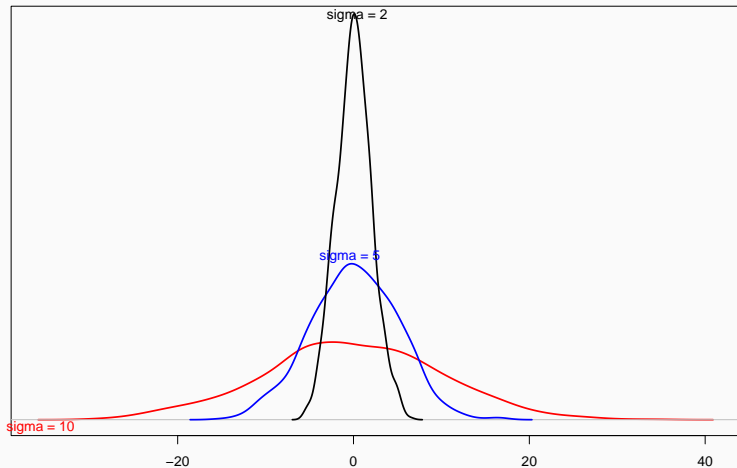
**large sigma**



Residual variation (sigma) is the Std. Dev. of residuals

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

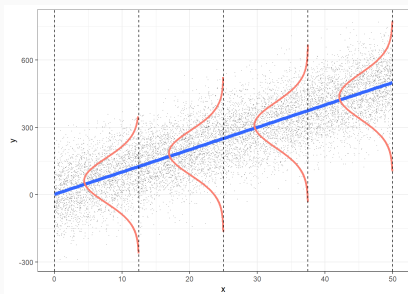
**Distribution of residuals**



# In a general linear model we assume residuals are

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

- Independent
- Normal
- Centred on 0 (no bias)
- Homogeneous variance (*homoscedasticity*)



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

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

<https://pollev.com/franciscorod726>

## Linear models

---

## Example dataset: forest trees

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

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

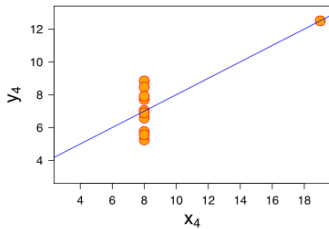
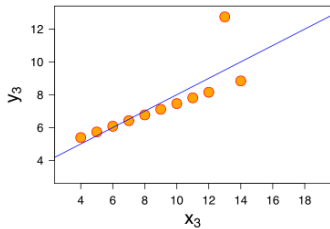
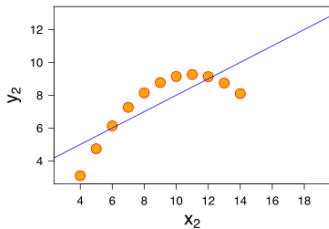
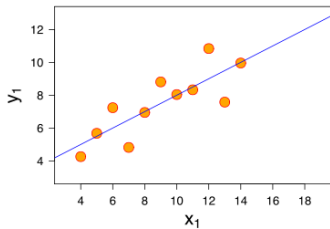
	site	dbh	height	sex	dead
1	4	29.68	36.1	male	0
2	5	33.29	42.3	male	0
3	2	28.03	41.9	female	0
4	5	39.86	46.5	female	0
5	1	47.94	43.9	female	0
6	1	10.82	26.2	male	0

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

Plot your data!

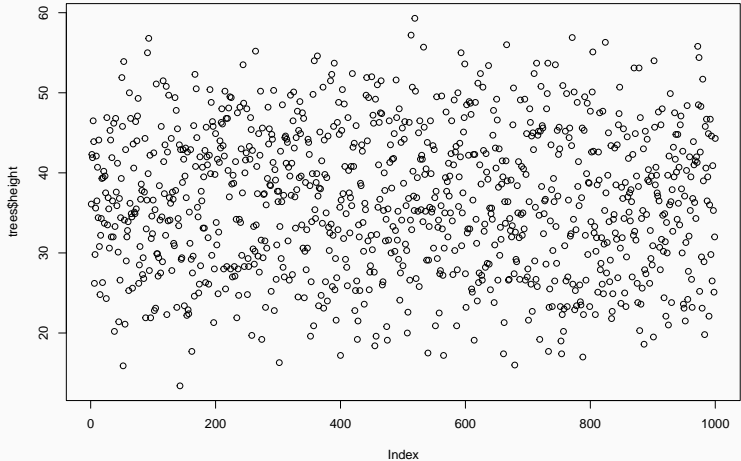
---

# Plot your data!

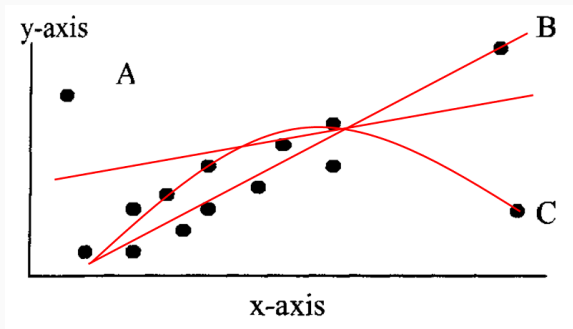


## Outliers

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



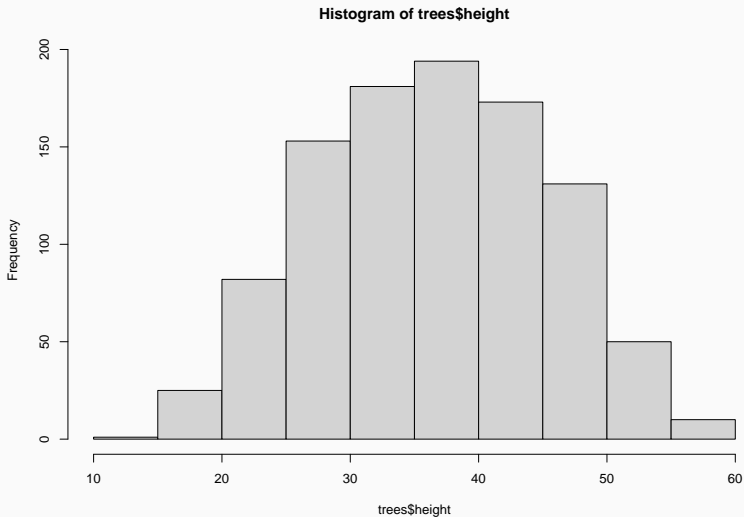
## Outliers impact on regression



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

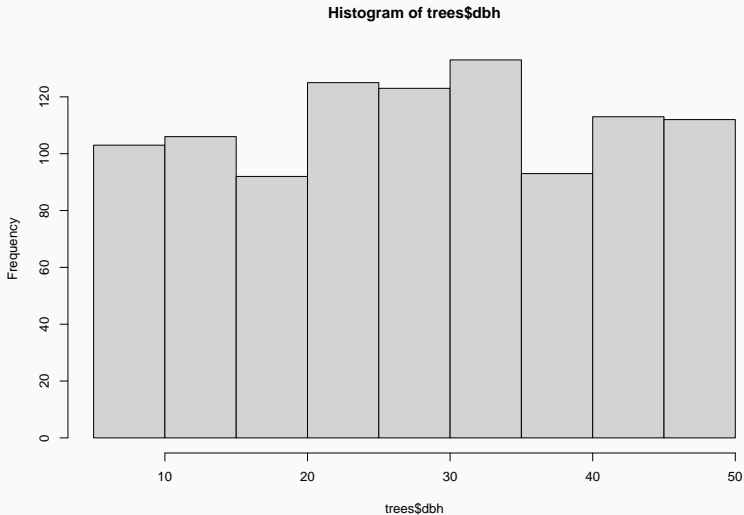
# Histogram of response variable

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



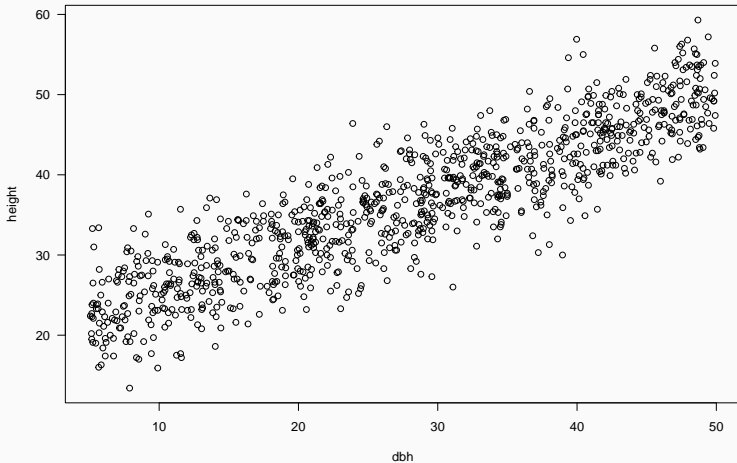
# Histogram of predictor variable

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



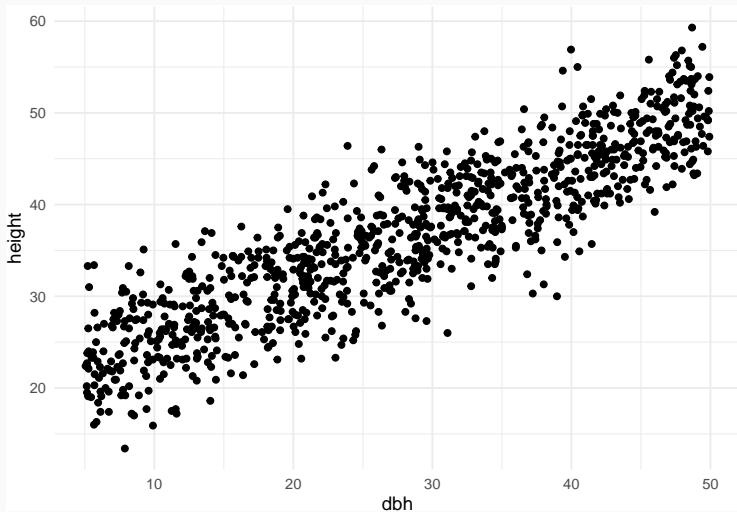
# Scatterplot

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



# Scatterplot

```
ggplot(trees) +  
  geom_point(aes(x = dbh, y = height))
```



## Model fitting

---

Hint: `lm`

Hint: `lm`

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

which corresponds to

$$\text{Height}_i = a + b \cdot \text{DBH}_i + \varepsilon_i$$

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

```
library('equatiomatic')  
m1 <- lm(height ~ dbh, data = trees)  
equatiomatic::extract_eq(m1)
```

$$\text{height} = \alpha + \beta_1(\text{dbh}) + \epsilon \quad (1)$$

```
equatiomatic::extract_eq(m1, use_coefs = TRUE)
```

$$\widehat{\text{height}} = 19.34 + 0.62(\text{dbh}) \quad (2)$$

```
library(texPreview)  
tex_preview(equatiomatic::extract_eq(m1))
```

## Model interpretation

---

## What does this mean?

```
summary(m1)
```

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 ***

---

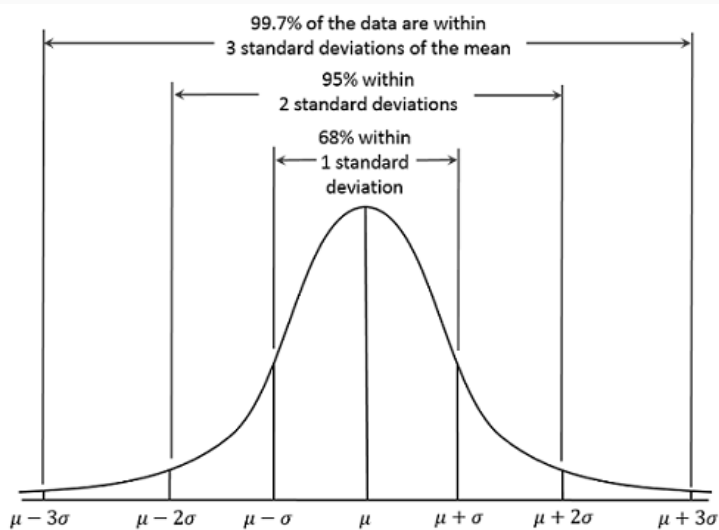
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.093 on 998 degrees of freedom

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

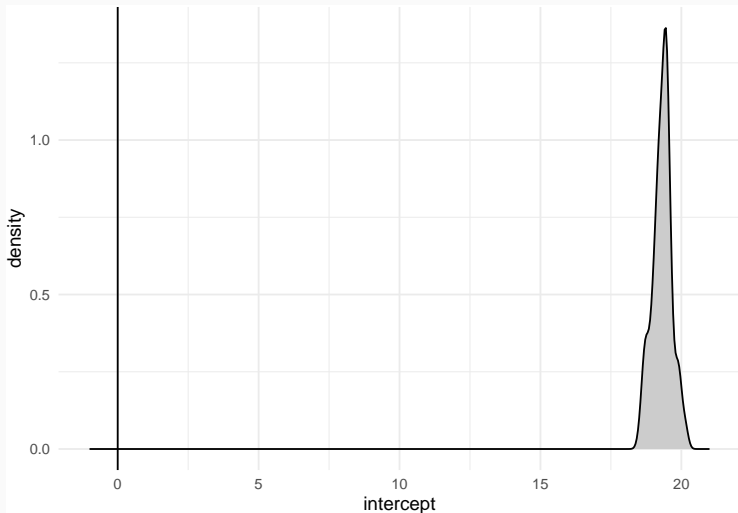
F-statistic: 3695 on 1 and 998 DF, p-value: < 2.2e-16

Remember that in a Normal distribution



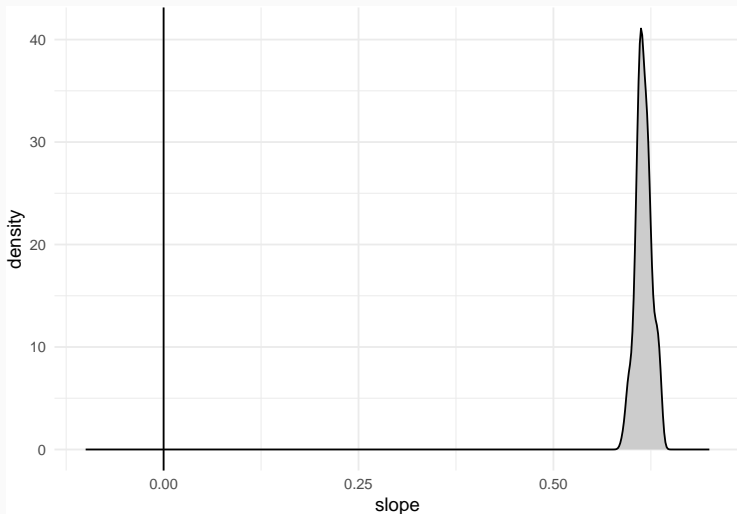
# Estimated distribution of the intercept parameter

Parameter	Coefficient	SE	95% CI	t(998)	p
(Intercept)	19.34	0.31	[18.73, 19.95]	62.26	< .001

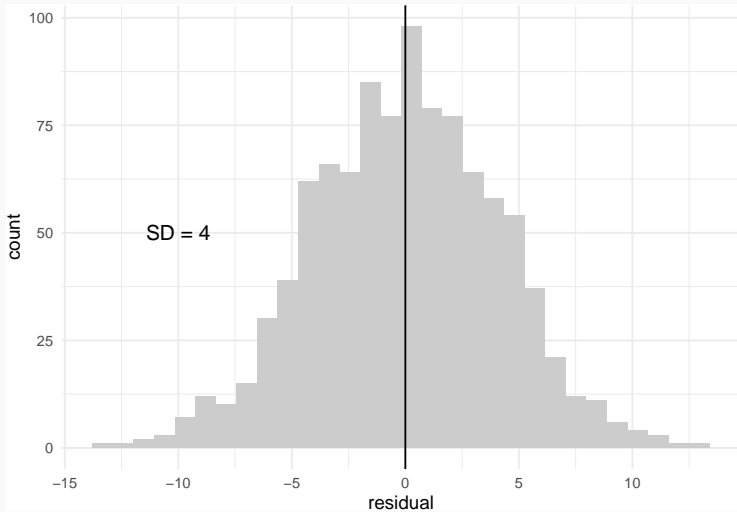


# Estimated distribution of the slope parameter

Parameter	Coefficient	SE	95% CI	t(998)	p
dbh	0.62	0.01	[0.60, 0.64]	60.79	< .001



# Distribution of residuals



$$DF = n - p$$

$n$  = sample size

$p$  = number of estimated parameters

Proportion of 'explained' variance

$$R^2 = 1 - \frac{\textit{Residual Variation}}{\textit{Total Variation}}$$

Accounts for model complexity  
(number of parameters)

$$R_{adj}^2 = 1 - (1 - R^2) \frac{n-1}{n-p-1}$$

<https://pollev.com/franciscorod726>

## Centering continuous predictors

---

- Helps interpretation
- Helps (complex) model convergence
- Centering usually around the mean, but can be any suitable reference point

## Centering continuous predictors

```
mean(trees$dbh)
```

```
[1] 27.88209
```

```
trees$dbh.c <- trees$dbh - 30
```

```
summary(trees$dbh)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
5.06	17.69	28.62	27.88	38.97	49.92

```
summary(trees$dbh.c)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-24.940	-12.312	-1.380	-2.118	8.965	19.920

## Centering continuous predictors

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

Call:

```
lm(formula = height ~ dbh.c, 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)	37.81030	0.13119	288.22	<2e-16 ***
dbh.c	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

## Extracting model info

---

```
coef(m1)
```

```
(Intercept)          dbh  
 19.3391968    0.6157036
```

```
confint(m1)
```

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

## 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.  
# i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

<https://broom.tidymodels.org/>

## Retrieving model parameters with parameters package

```
library('easystats') # parameters package
parameters(m1)
```

Parameter	Coefficient	SE	95% CI	t(998)	p
(Intercept)	19.34	0.31	[18.73, 19.95]	62.26	< .001
dbh	0.62	0.01	[ 0.60, 0.64]	60.79	< .001

<https://easystats.github.io/parameters/>

## Communicating results

---

A screenshot of the top portion of a Nature journal article page. The header is dark red with the 'nature' logo and 'international journal of science' in white. A 'MENU' dropdown is on the left, and a 'Subs' button is on the right. Below the header, the text 'EDITORIAL • 20 MARCH 2019' is displayed. The main title 'It's time to talk about ditching statistical significance' is in a large, dark serif font.

MENU ▾ nature  
international journal of science

Subs

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>

- We found a **significant relationship** between DBH and Height ( $p < 0.05$ ).
- We found a *{significant}* **positive** relationship between DBH and Height  $\{(p < 0.05)\}$  ( $b = 0.61, SE = 0.01$ ).
- (add p-value if you wish)

```
library('report')  
report(m1)
```

We fitted a linear model (estimated using OLS) to predict height with dbh (formula: `height ~ dbh`). The model explains a statistically significant and substantial proportion of variance ( $R^2 = 0.79$ ,  $F(1, 998) = 3695.40$ ,  $p < .001$ , adj.  $R^2 = 0.79$ ). The model's intercept, corresponding to  $\text{dbh} = 0$ , is at 19.34 (95% CI [18.73, 19.95],  $t(998) = 62.26$ ,  $p < .001$ ). Within this model:

- The effect of dbh is statistically significant and positive ( $\beta = 0.62$ , 95% CI [0.60, 0.64],  $t(998) = 60.79$ ,  $p < .001$ ; Std.  $\beta = 0.89$ , 95% CI [0.86, 0.92])

Standardized parameters were obtained by fitting the model on a standardized version of the dataset. 95% Confidence Intervals (CIs) and p-values were computed using a Wald t-distribution approximation.

<https://easystats.github.io/report/>

```
library('gtsummary')  
tbl_regression(m1, intercept = TRUE)
```

Characteristic	Beta	95% CI	p-value
(Intercept)	19	19, 20	<0.001
dbh	0.62	0.60, 0.64	<0.001

Abbreviation: CI = Confidence Interval

<https://www.danielsjoberg.com/gtsummary>

# Generating table with model results: modelsummary

```
library('modelsummary')  
modelsummary(m1, output = 'markdown') # Word, PDF, PowerPoint, png...
```

---

	(1)
(Intercept)	19.339 (0.311)
dbh	0.616 (0.010)
Num.Obs.	1000
R2	0.787
R2 Adj.	0.787
AIC	5660.3
BIC	5675.0
Log.Lik.	-2827.125
F	3695.395
RMSE	4.09

---

<https://modelsummary.com/>

## Generating table with model results: modelsummary

```
modelsummary(m1, fmt = 2,  
             estimate = '{estimate} ({std.error})',  
             statistic = NULL,  
             gof_map = c('nobs', 'r.squared', 'rmse'),  
             output = 'markdown') # Word, PDF, PowerPoint, png...
```

---

	(1)
(Intercept)	19.34 (0.31)
dbh	0.62 (0.01)
Num.Obs.	1000
R2	0.787
RMSE	4.09

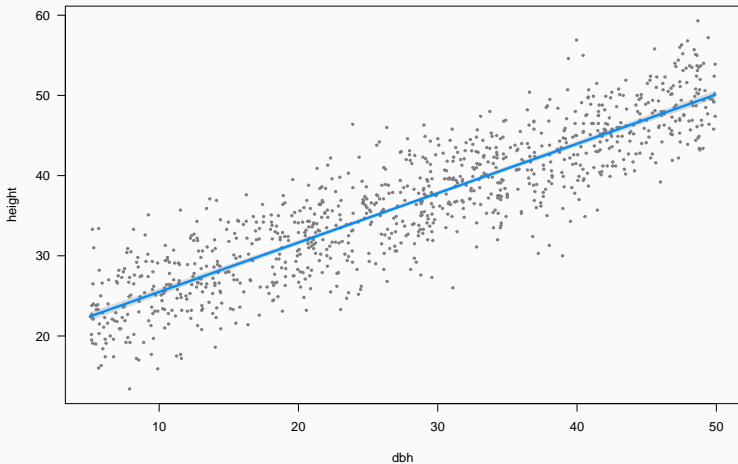
---

## Visualising fitted model

---

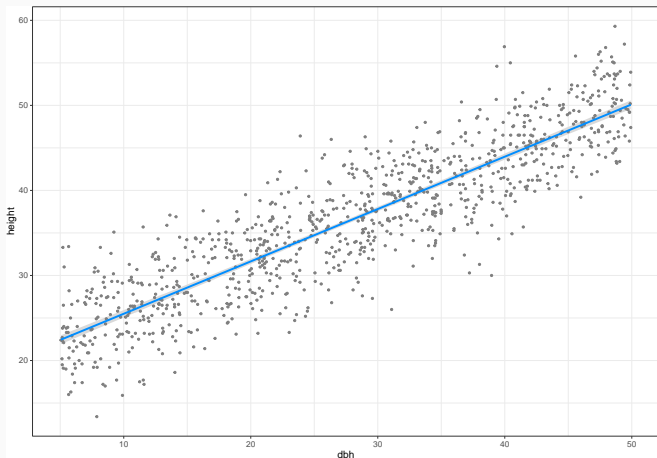
## Plot model: visreg

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



visreg can use ggplot2 too

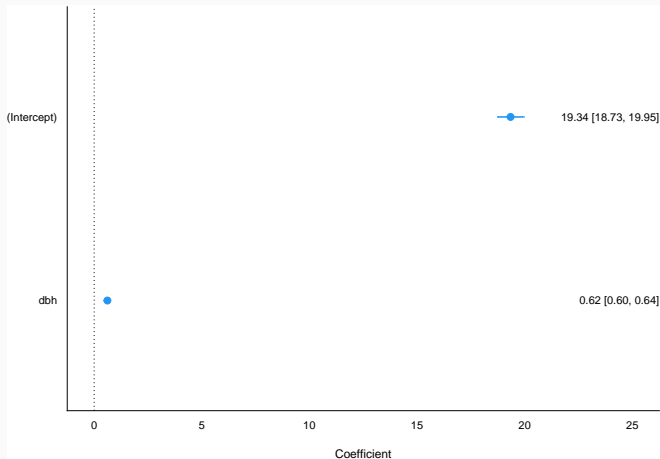
```
visreg(m1, gg = TRUE) + theme_bw()
```



<https://pbreheny.github.io/visreg>

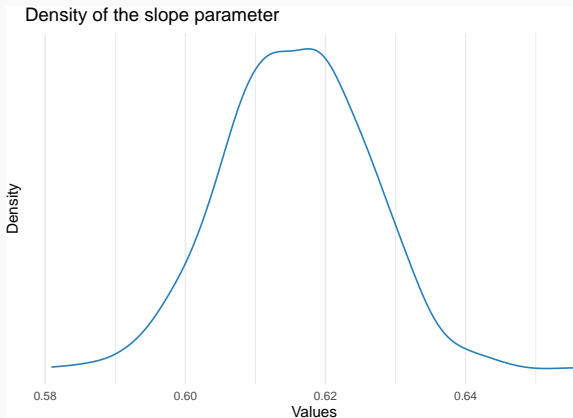
## Plot model parameters with easystats (see package)

```
library('easystats')  
plot(parameters(m1), show_intercept = TRUE, show_labels = TRUE)
```



## Plot parameters' estimated distribution

```
plot(simulate_parameters(m1)) +  
  labs(title = 'Density of the slope parameter')
```

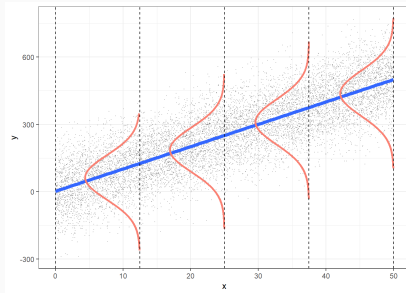


## Model checking

---

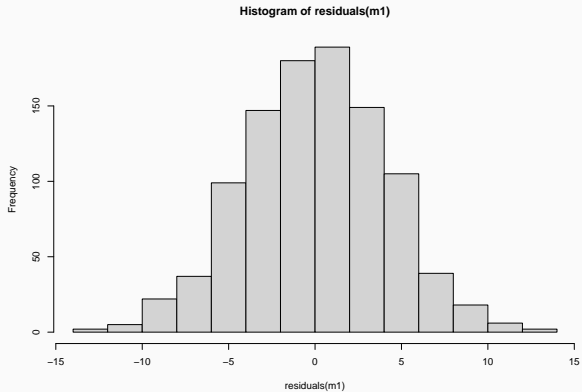
# Linear model assumptions

- **Linearity** (transformations, GAM...)
- **Residuals:**
  - Independent
  - Constant variance
  - Normal
  - Mean close to zero



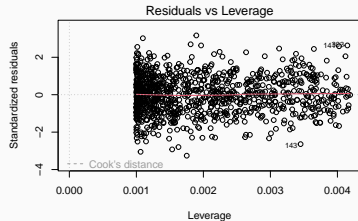
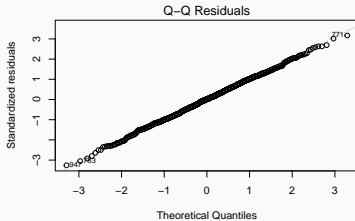
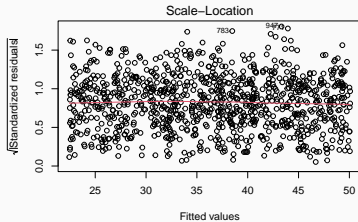
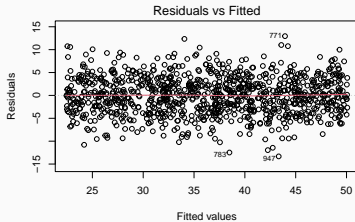
# Are residuals normal?

```
hist(residuals(m1))
```



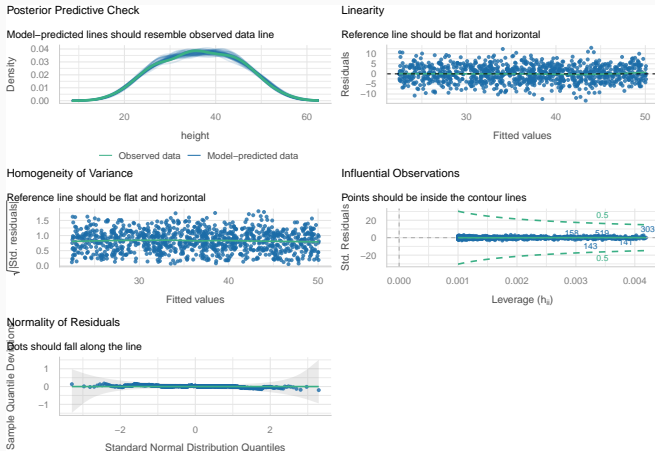
SD = 4.09

# Model checking: `plot(model)`



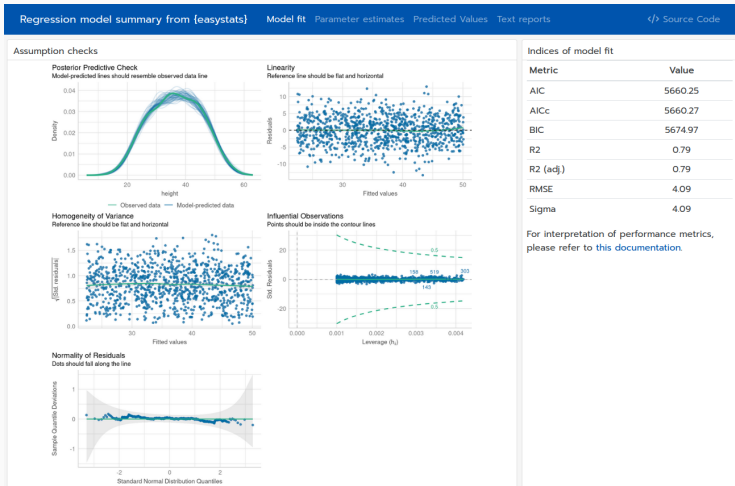
# Model checking with performance (easystats)

```
library('easystats')  
check_model(m1)
```



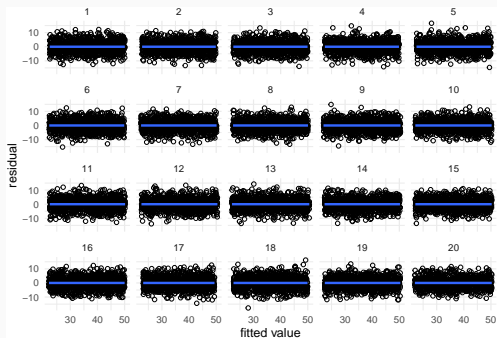
# A dashboard to explore the full model

```
library('easystats')  
model_dashboard(m1)
```



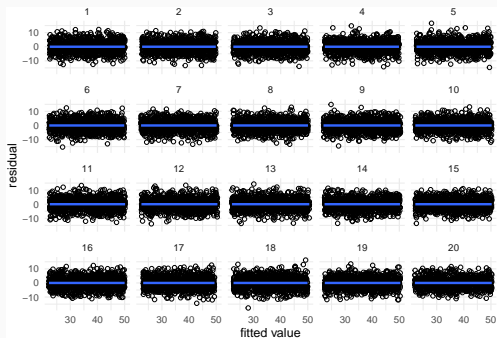
# Can you distinguish model residuals from stochastic noise?

```
library('cannonball') # https://github.com/janhove/cannonball  
lin_plot(parade(m1))
```



# Can you distinguish model residuals from stochastic noise?

```
library('cannonball') # https://github.com/janhove/cannonball  
lin_plot(parade(m1))
```



```
reveal(parade(m1))
```

The true data are in position 10.

## Using model for prediction

---

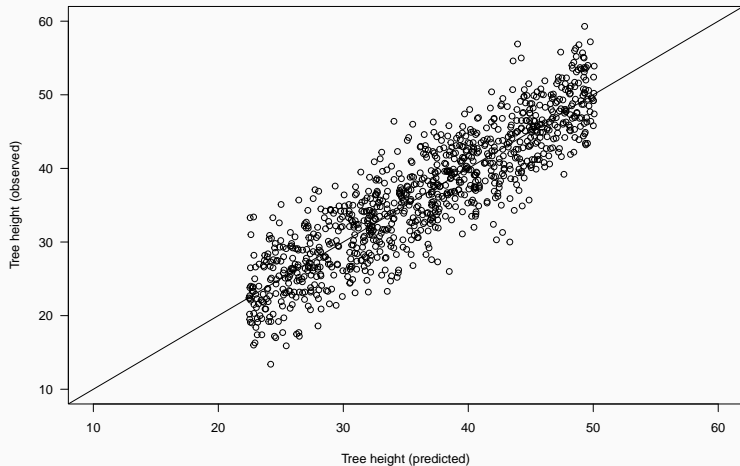
## How good is the model in predicting tree height?

`fitted` gives expected value for each observation

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

	site	dbh	height	sex	dead	dbh.c	height.pred	resid
1	4	29.68	36.1	male	0	-0.32	37.61328	-1.5132797
2	5	33.29	42.3	male	0	3.29	39.83597	2.4640303
3	2	28.03	41.9	female	0	-1.97	36.59737	5.3026313
4	5	39.86	46.5	female	0	9.86	43.88114	2.6188577
5	1	47.94	43.9	female	0	17.94	48.85603	-4.9560274
6	1	10.82	26.2	male	0	-19.18	26.00111	0.1988903

## Calibration plot: Observed vs Predicted values



## Making predictions for new data

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
```

## Confidence vs Prediction Intervals

Q: Expected tree height if DBH = 39 cm?

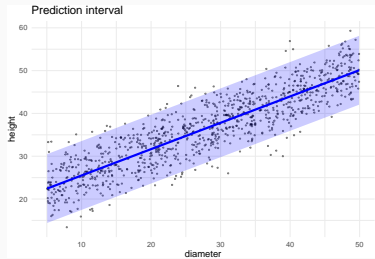
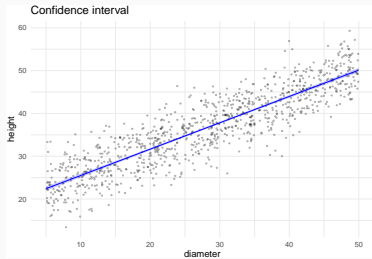
```
predict(m1, new.dbh, interval = 'confidence')
```

```
      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
```

# Confidence vs Prediction Intervals



## Making predictions with easystats

---

## Estimate expected values

```
pred <- estimate_expectation(m1)
```

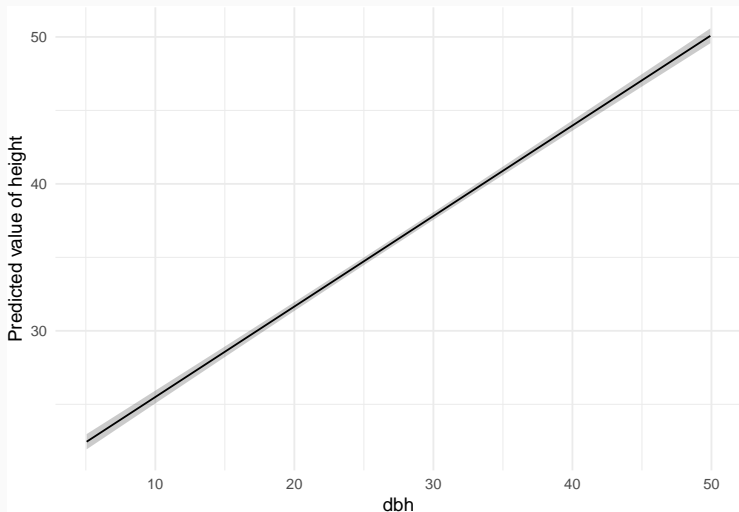
### Model-based Predictions

dbh	Predicted	SE	95% CI	Residuals
29.68	37.61	0.13	[37.36, 37.87]	-1.51
33.29	39.84	0.14	[39.56, 40.11]	2.46
28.03	36.60	0.13	[36.34, 36.85]	5.30
39.86	43.88	0.18	[43.53, 44.23]	2.62
47.94	48.86	0.24	[48.38, 49.33]	-4.96
10.82	26.00	0.22	[25.58, 26.42]	0.20

Variable predicted: height

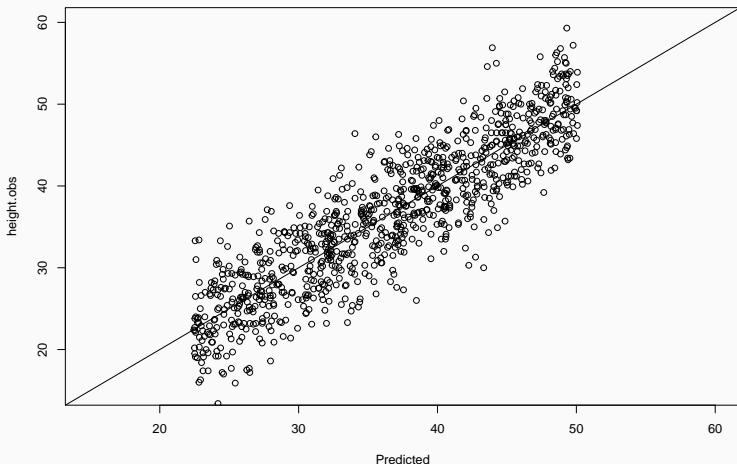
## Expected values given DBH

```
plot(estimate_expectation(m1, by = "dbh"))
```



# Calibration plot: observed vs predicted

```
pred$height.obs <- trees$height  
plot(height.obs ~ Predicted, data = pred, xlim = c(15, 60), ylim = c(15, 60))  
abline(a = 0, b = 1)
```



## Estimate prediction interval

Accounting for residual variation!

```
pred <- estimate_prediction(m1)
head(pred)
```

### Model-based Predictions

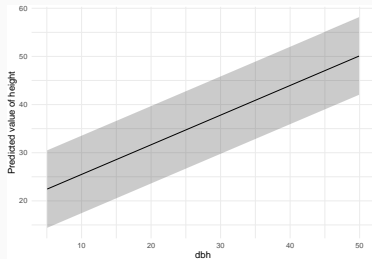
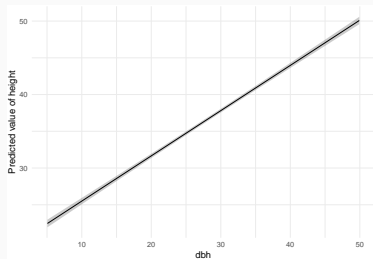
dbh	Predicted	SE	95% CI	Residuals
29.68	37.61	4.09	[29.58, 45.65]	-1.51
33.29	39.84	4.10	[31.80, 47.87]	2.46
28.03	36.60	4.09	[28.56, 44.63]	5.30
39.86	43.88	4.10	[35.84, 51.92]	2.62
47.94	48.86	4.10	[40.81, 56.90]	-4.96
10.82	26.00	4.10	[17.96, 34.04]	0.20

Variable predicted: height

# Confidence vs Prediction interval

```
plot(estimate_expectation(m1, by = dbh))
```

```
plot(estimate_prediction(m1, by = dbh))
```



## Make predictions for new data

```
estimate_expectation(m1, data = data.frame(dbh = 39))
```

Model-based Predictions

dbh	Predicted	SE	95% CI
39	43.35	0.17	[43.01, 43.69]

Variable predicted: height

```
estimate_prediction(m1, data = data.frame(dbh = 39))
```

Model-based Predictions

dbh	Predicted	SE	95% CI
39	43.35	4.10	[35.31, 51.39]

```
performance_cv(m1, method = 'k_fold', metrics = 'common', k = 10)
```

```
# Cross-validation performance (10-fold method)
```

```
RMSE | R2
```

```
-----
```

```
4.1 | 0.79
```

- Visualise data
- Understand fitted model (summary)
- Visualise model (visreg...)
- Check model (plot, check\_model, calibration plot...)
- Predict (predict, estimate\_expectation, estimate\_prediction)

## Categorical predictors (factors)

---

## Example dataset: forest trees

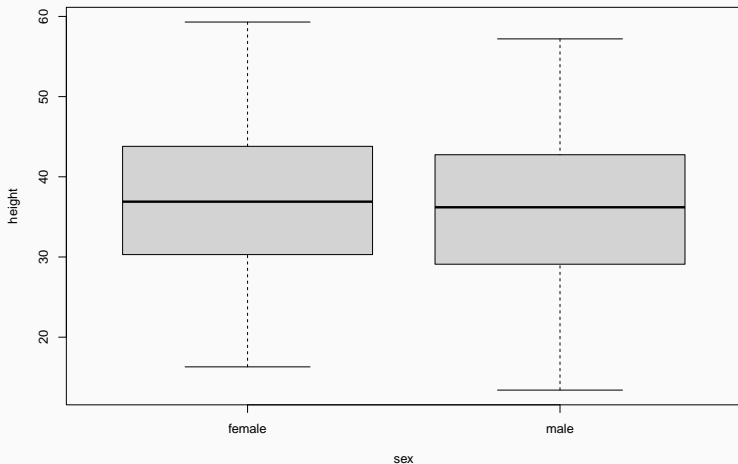
```
trees <- read.csv('data/trees.csv')  
head(trees)
```

	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

Does tree height vary with sex?

Q: Does tree height vary with sex?

```
boxplot(height ~ 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

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

corresponds to

$$\text{Height}_i = a + b_{\text{male}} + \varepsilon_i$$

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

## 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

<https://pollev.com/franciscorod726>

### `report(m2)`

We fitted a linear model (estimated using OLS) to predict height with sex (formula: `height ~ sex`). The model explains a statistically not significant and very weak proportion of variance ( $R^2 = 2.26e-03$ ,  $F(1, 998) = 2.26$ ,  $p = 0.133$ , adj.  $R^2 = 1.26e-03$ ). The model's intercept, corresponding to sex = female, is at 36.93 (95% CI [36.15, 37.71],  $t(998) = 92.78$ ,  $p < .001$ ). Within this model:

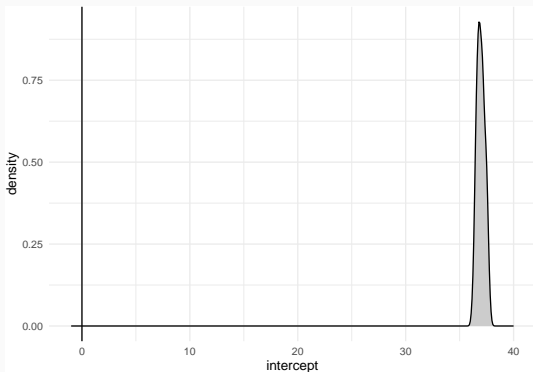
- The effect of sex [male] is statistically non-significant and negative (beta = -0.84, 95% CI [-1.94, 0.26],  $t(998) = -1.50$ ,  $p = 0.133$ ; Std. beta = -0.10, 95% CI [-0.22, 0.03])

Standardized parameters were obtained by fitting the model on a standardized version of the dataset. 95% Confidence Intervals (CIs) and p-values were computed using a Wald t-distribution approximation.

# Estimated distribution of the intercept parameter

Intercept = Height of females

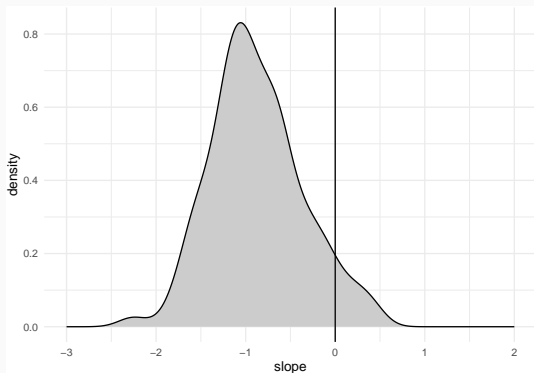
Parameter	Coefficient	SE	95% CI	t(998)	p
(Intercept)	36.93	0.40	[36.15, 37.71]	92.78	< .001



# Estimated distribution of the *beta* parameter

*beta* = height difference of males vs females

Parameter	Coefficient	SE	95% CI	t(998)	p
sex [male]	-0.84	0.56	[-1.94, 0.26]	-1.50	0.133

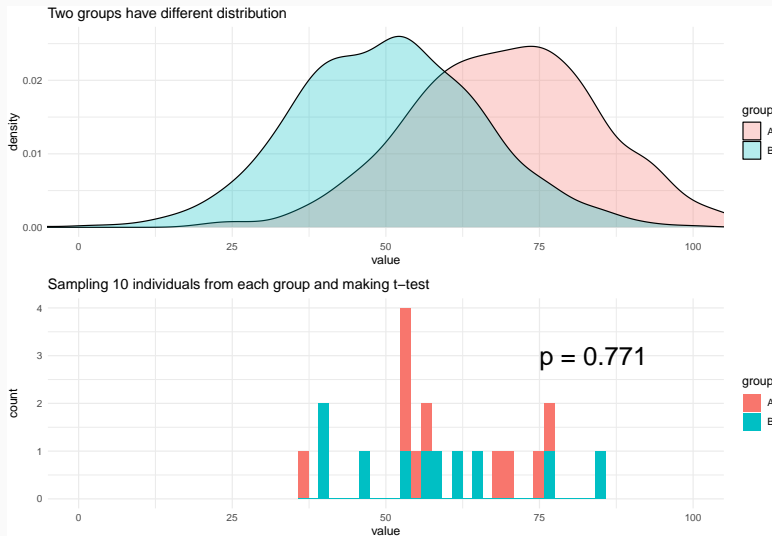


## Short digression on p-values

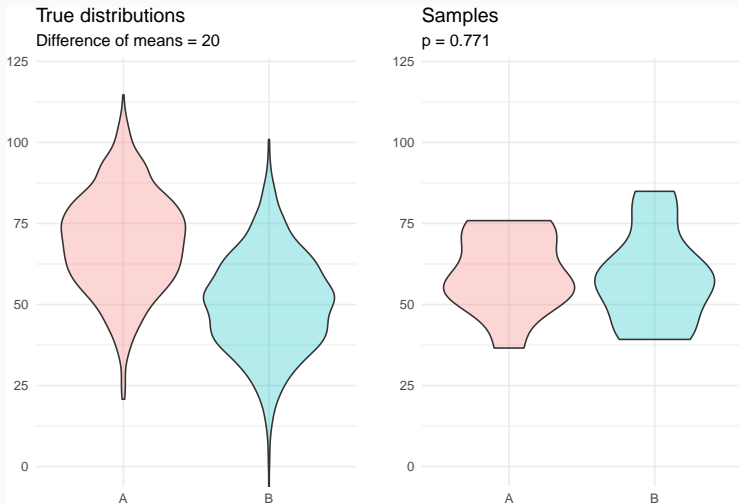
---

‘Not significant’  
does NOT mean  
‘there is no effect’

# 'Not significant' does NOT mean 'they are equal'



# 'Not significant' does NOT mean 'there is no effect'



Failure to reject  $H_0$   $\neq$   $H_0$  is true

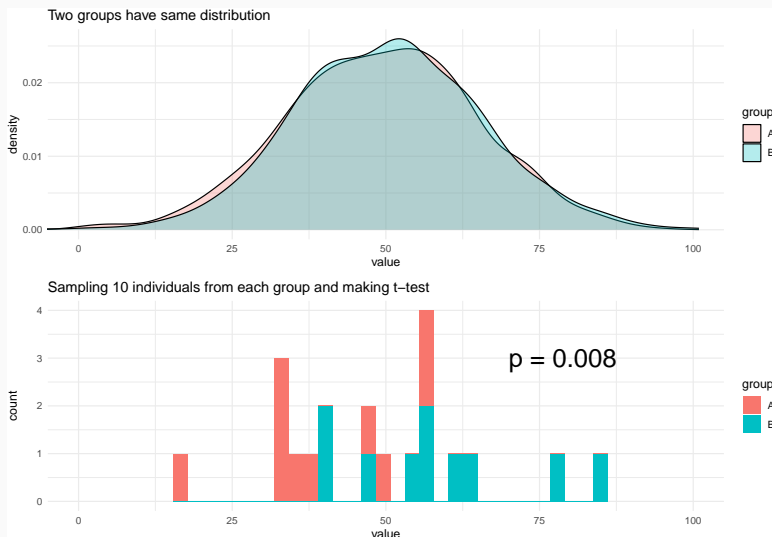
Absence of evidence  $\neq$  Evidence of absence

- “We were **unable to find evidence** against the hypothesis that  $A = B$  **with the current sample size**” ([Harrell](#))
- “Differences between groups were **not statistically clear**” ([Dushoff et al](#))

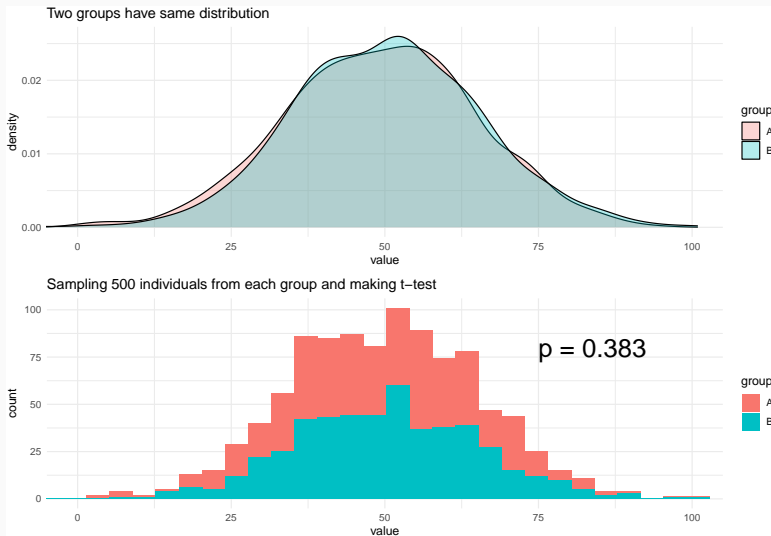
A significant p-value  
does NOT mean  
we found a true difference

# A significant p-value does not mean we found a true difference

Particularly with low sample sizes



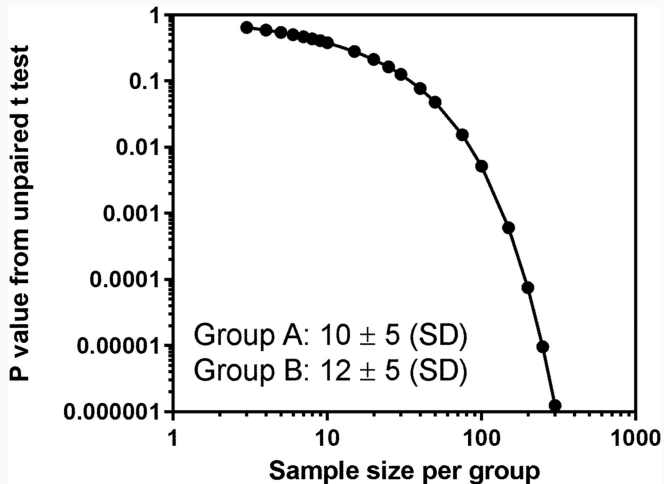
# If sample size was larger...



With low sample size (power),  
significant p-values  
are most likely overestimates

Loken & Gelman 2014, Vasisth et al. 2018

## P-value depends on sample size



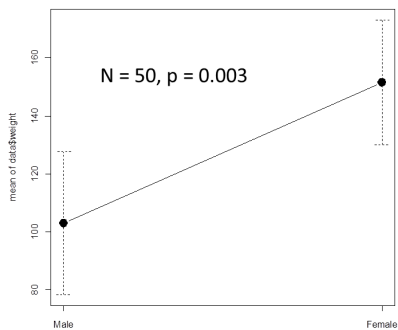
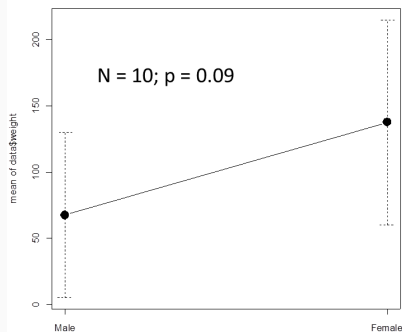
<https://doi.org/10.1002/prp2.93>

Try yourself [here!](#)

# P-value depends on sample size

Same real difference is detected as **significant** or not depending on sample size

Real difference = 40 g



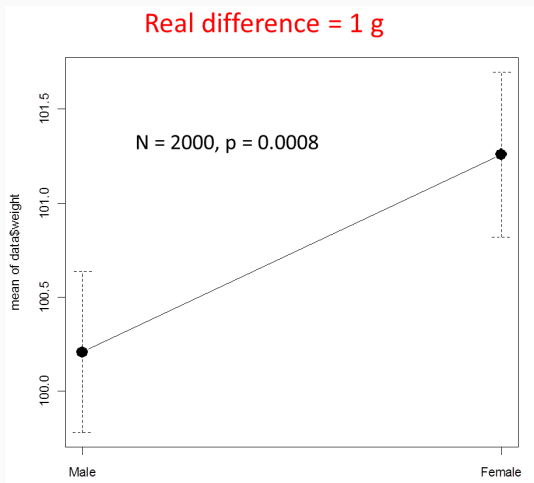
Statistically significant

!=

biologically important

## Statistically significant != biologically important

With big sample size, we can find highly significant but biologically unimportant differences.



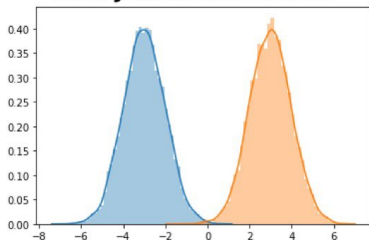
# Statistically significant != biologically important



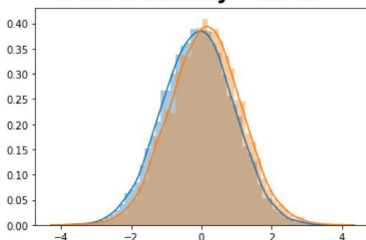
zara weinberg  
@weinberz

friendly reminder about  $p < 0.0001$ :

### What you think it means:



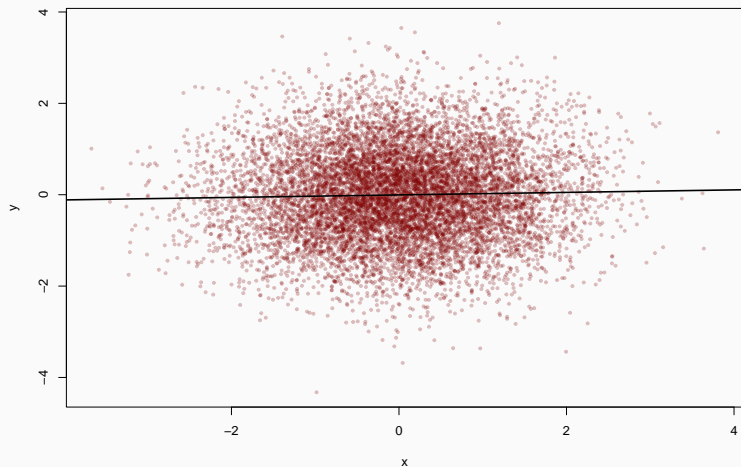
### What it actually means:



<https://twitter.com/weinberz/status/1422405165236178947?s=20>

Statistically significant != biologically important

**$p = 0.005$**



## Statistically significant != biologically important

- Statistically significant = unlikely to be zero
- Good read: *significantly misleading*
- Beyond significant/not significant, look at **effect sizes and their uncertainty**.

- P-values do not measure the **probability of hypothesis** being true, or the probability that the data were produced by **random chance** alone.
- Scientific conclusions or policy decisions should NOT be based only on **whether a p-value passes a specific threshold**.
- P-value, or statistical significance, does not measure the **size of an effect** or the **importance** of a result.
- By itself, a p-value does NOT provide a good **measure of evidence** regarding a model or hypothesis.

<https://doi.org/10.1080/00031305.2016.1154108>

See also [https://lakens.github.io/statistical\\_inferences/01-pvalue.html](https://lakens.github.io/statistical_inferences/01-pvalue.html)

## S-values as alternative to p-values

```
parameters(m2)[2,]
```

Parameter	Coefficient	SE	95% CI	t(998)	p
sex [male]	-0.84	0.56	[-1.94, 0.26]	-1.50	0.133

```
parameters(m2, s_value = TRUE)[2,]
```

Parameter	Coefficient	SE	95% CI	t(998)	s
sex [male]	-0.84	0.56	[-1.94, 0.26]	-1.50	2.91

p-value = 0.133 corresponds to *surprise* of obtaining c. 3 heads in a row when tossing a coin

<https://marginaleffects.com/bonus/svalues.html>

```
library('easystats') # modelbased package
estimate_means(m2)
```

Estimated Marginal Means

sex	Mean	SE	95% CI	t(998)
female	36.93	0.40	[36.15, 37.71]	92.78
male	36.09	0.39	[35.31, 36.86]	91.39

Variable predicted: height

Predictors modulated: sex

```
estimate_contrasts(m2)
```

Marginal Contrasts Analysis

Level1	Level2	Difference	SE	95% CI	t(998)	p
male	female	-0.84	0.56	[-1.94, 0.26]	-1.50	0.133

Variable predicted: height

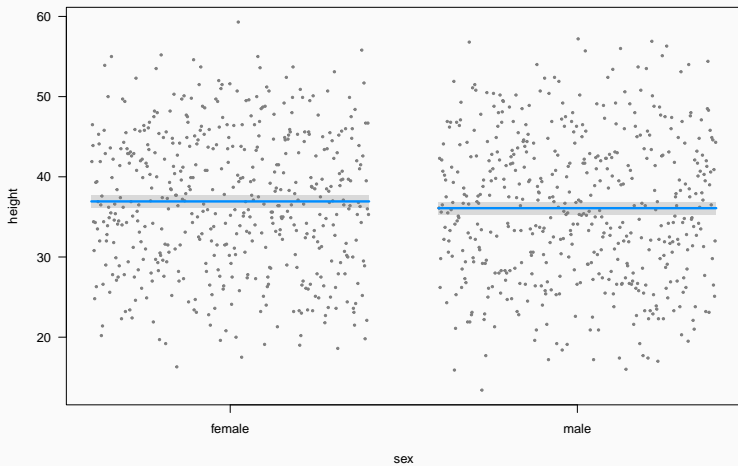
Predictors contrasted: sex

p-values are uncorrected.

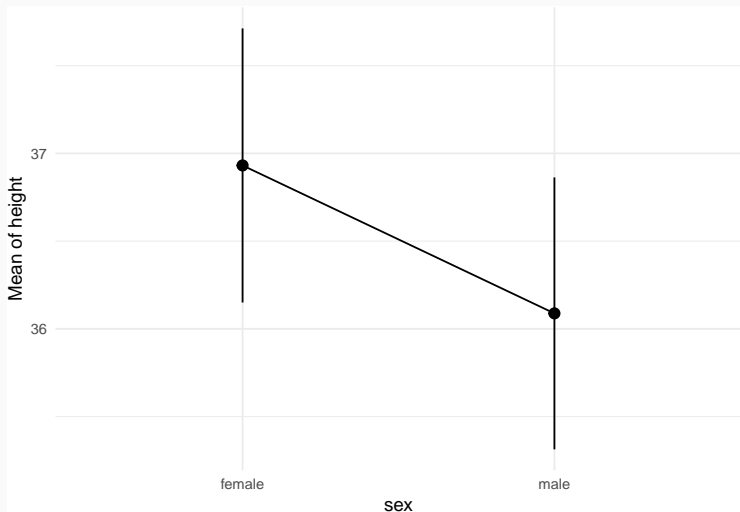
## Visualising the fitted model

---

`visreg(m2)`



```
plot(estimate_means(m2))
```



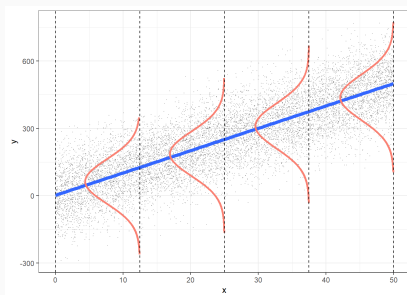
## Model checking

---

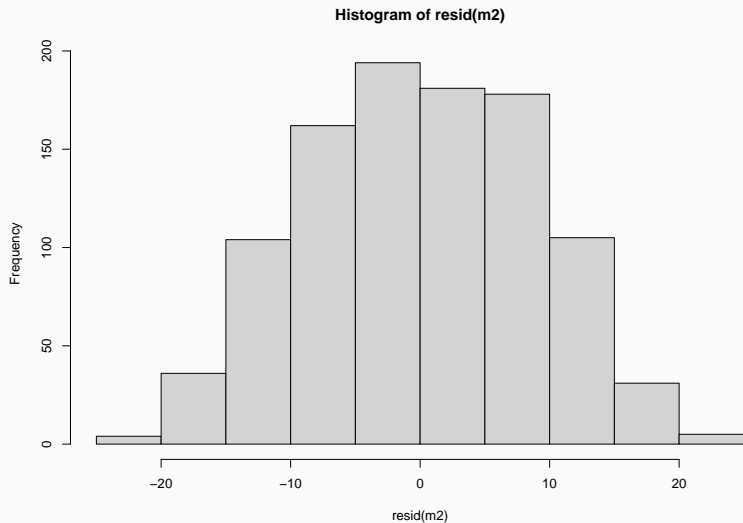
# In a general linear model we assume residuals are

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

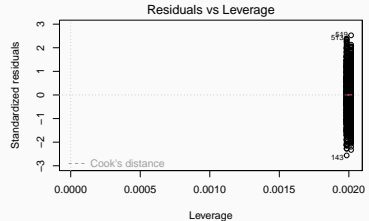
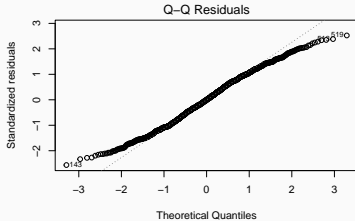
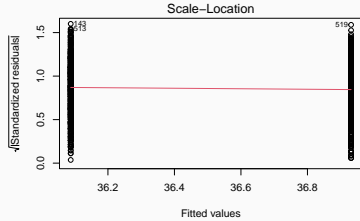
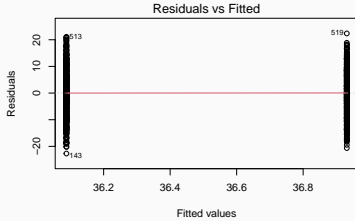
- Independent
- Normal
- Centred on 0 (no bias)
- Homogeneous variance (*homoscedasticity*)



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



# Model checking: residuals

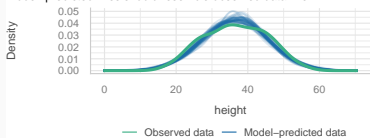


# Model checking

```
library('easystats')  
check_model(m2)
```

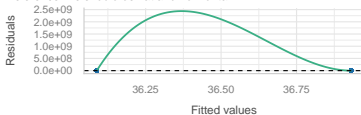
## Posterior Predictive Check

Model-predicted lines should resemble observed data line



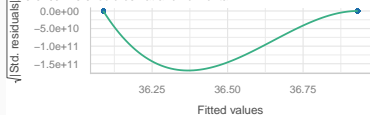
## Linearity

Reference line should be flat and horizontal



## Homogeneity of Variance

Reference line should be flat and horizontal



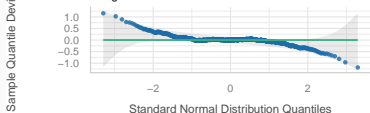
## Influential Observations

Points should be inside the contour lines



## Normality of Residuals

Points should fall along the line



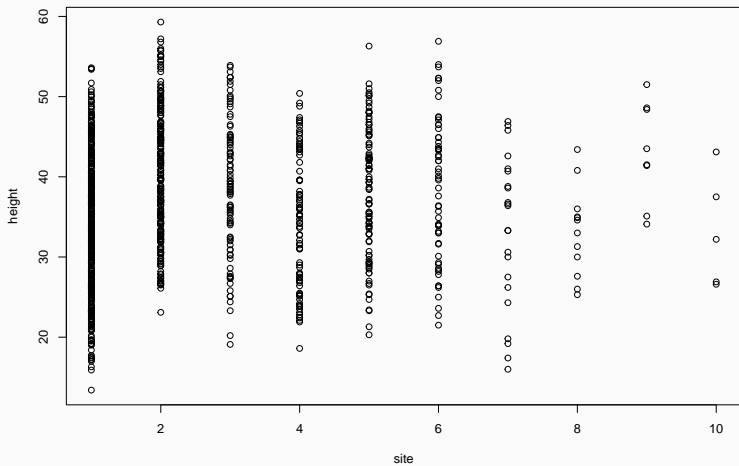
```
model_dashboard(m2)
```

Q: Does height differ among field sites?

---

<https://pollev.com/franciscorod726>

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



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

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

## 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

```
extract_eq(m3)
```

$$\text{height} = \alpha + \beta_1(\text{site}) + \epsilon \quad (3)$$

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

```
trees <- trees |>  
  dplyr::mutate(site = as.factor(site))
```

## Let's check model structure with `equatiomatic`

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

$$\text{height} = \alpha + \beta_1(\text{site}_2) + \beta_2(\text{site}_3) + \beta_3(\text{site}_4) + \beta_4(\text{site}_5) + \beta_5(\text{site}_6) + \beta_6(\text{site}_7) + \beta_7(\text{site}_8) + \beta_8(\text{site}_9) \quad (4)$$

# 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

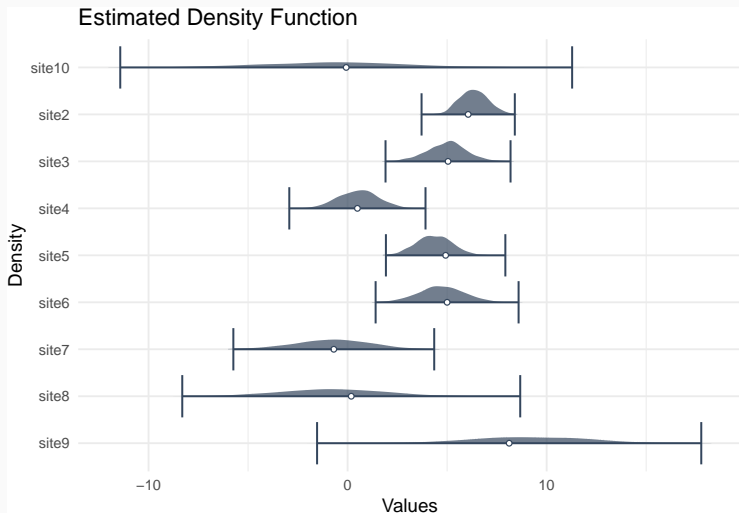
Residual standard error: 8.446 on 990 degrees of freedom

Multiple R-squared: 0.1016, Adjusted R-squared: 0.09344

F-statistic: 12.44 on 9 and 990 DF, p-value: < 2.2e-16

# Estimated parameter distributions

```
plot(simulate_parameters(m3), stack = FALSE)
```



## Estimated tree heights for each site

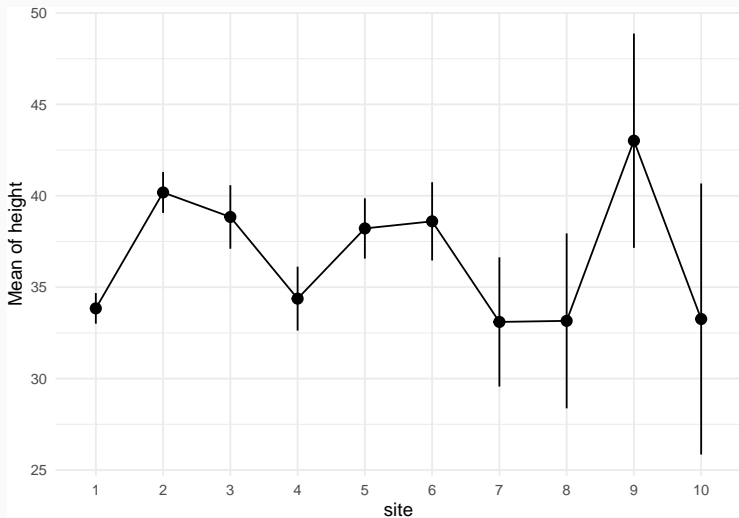
```
estimate_means(m3)
```

Estimated Marginal Means

site	Mean	SE	95% CI	t(990)
1	33.84	0.43	[33.00, 34.68]	79.33
2	40.18	0.57	[39.06, 41.30]	70.40
3	38.84	0.89	[37.10, 40.58]	43.87
4	34.37	0.89	[32.63, 36.12]	38.61
5	38.21	0.84	[36.56, 39.86]	45.47
6	38.60	1.09	[36.46, 40.74]	35.40
7	33.10	1.80	[29.57, 36.63]	18.38
8	33.16	2.44	[28.37, 37.94]	13.60
9	43.01	2.99	[37.15, 48.87]	14.40
10	33.26	3.78	[25.85, 40.67]	8.81

## Plot estimated tree heights for each site

```
plot(estimate_means(m3))
```



# Analysing differences among factor levels

For finer control see `emmeans` package

```
estimate_contrasts(m3)
```

## Marginal Contrasts Analysis

Level1	Level2	Difference	SE	95% CI	t(990)	p
2	1	6.34	0.71	[ 4.94, 7.74]	8.90	< .001
3	1	5.00	0.98	[ 3.07, 6.93]	5.09	< .001
4	1	0.53	0.99	[ -1.40, 2.47]	0.54	0.589
5	1	4.37	0.94	[ 2.52, 6.22]	4.64	< .001
6	1	4.76	1.17	[ 2.46, 7.06]	4.07	< .001
7	1	-0.74	1.85	[ -4.37, 2.89]	-0.40	0.689
8	1	-0.68	2.48	[ -5.54, 4.17]	-0.28	0.783
9	1	9.17	3.02	[ 3.25, 15.09]	3.04	0.002
10	1	-0.58	3.80	[ -8.04, 6.88]	-0.15	0.878
3	2	-1.34	1.05	[ -3.41, 0.73]	-1.27	0.203
4	2	-5.81	1.06	[ -7.88, -3.73]	-5.49	< .001
5	2	-1.97	1.02	[ -3.96, 0.02]	-1.94	0.053
6	2	-1.58	1.23	[ -4.00, 0.83]	-1.28	0.199
7	2	-7.08	1.89	[ -10.79, -3.38]	-3.75	< .001
8	2	-7.02	2.50	[ -11.94, -2.11]	-2.81	0.005
9	2	2.83	3.04	[ -3.14, 8.80]	0.93	0.352
10	2	-6.92	3.82	[ -14.42, 0.57]	-1.81	0.070
4	3	-4.47	1.26	[ -6.93, -2.00]	-3.56	< .001
5	3	-0.63	1.22	[ -3.02, 1.77]	-0.51	0.608
6	3	-0.24	1.40	[ -3.00, 2.52]	-0.17	0.865
7	3	-5.74	2.01	[ -9.68, -1.80]	-2.86	0.004
8	3	-5.68	2.59	[ -10.77, -0.59]	-2.19	0.029
9	3	4.17	3.11	[ -1.94, 10.28]	1.34	0.181
10	3	-5.58	3.88	[ -13.19, 2.03]	-1.44	0.151
5	4	3.84	1.22	[ 1.44, 6.24]	3.14	0.002
6	4	4.23	1.41	[ 1.46, 6.99]	3.00	0.003
7	4	-1.27	2.01	[ -5.22, 2.67]	-0.63	0.526

How different are site 2 and site 9?

```
library('marginaleffects')  
hypotheses(m3, 'site2 = site9')
```

Hypothesis	Estimate	Std. Error	z	Pr(> z )	S	2.5 %	97.5 %
site2=site9	-2.83	3.04	-0.931	0.352	1.5	-8.79	3.13

## Presenting model results

```
parameters(m3)
```

Parameter	Coefficient	SE	95% CI	t(990)	p
(Intercept)	33.84	0.43	[33.00, 34.68]	79.33	< .001
site [2]	6.34	0.71	[ 4.94, 7.74]	8.90	< .001
site [3]	5.00	0.98	[ 3.07, 6.93]	5.09	< .001
site [4]	0.53	0.99	[-1.40, 2.47]	0.54	0.589
site [5]	4.37	0.94	[ 2.52, 6.22]	4.64	< .001
site [6]	4.76	1.17	[ 2.46, 7.06]	4.07	< .001
site [7]	-0.74	1.85	[-4.37, 2.89]	-0.40	0.689
site [8]	-0.68	2.48	[-5.54, 4.17]	-0.28	0.783
site [9]	9.17	3.02	[ 3.25, 15.09]	3.04	0.002
site [10]	-0.58	3.80	[-8.04, 6.88]	-0.15	0.878

## Presenting model results

```
estimate_means(m3)
```

Estimated Marginal Means

site	Mean	SE	95% CI	t(990)
1	33.84	0.43	[33.00, 34.68]	79.33
2	40.18	0.57	[39.06, 41.30]	70.40
3	38.84	0.89	[37.10, 40.58]	43.87
4	34.37	0.89	[32.63, 36.12]	38.61
5	38.21	0.84	[36.56, 39.86]	45.47
6	38.60	1.09	[36.46, 40.74]	35.40
7	33.10	1.80	[29.57, 36.63]	18.38
8	33.16	2.44	[28.37, 37.94]	13.60
9	43.01	2.99	[37.15, 48.87]	14.40
10	33.26	3.78	[25.85, 40.67]	8.81

Variable predicted: height

Predictors modulated: site

## Presenting model results

```
modelsummary(m3, estimate = '{estimate} ({std.error})', statistic = NULL,  
             fmt = 1, gof_map = NA, coef_rename = paste0('site', 1:10), output = 'markdown')
```

---

	(1)
site1	33.8 (0.4)
site2	6.3 (0.7)
site3	5.0 (1.0)
site4	0.5 (1.0)
site5	4.4 (0.9)
site6	4.8 (1.2)
site7	-0.7 (1.9)
site8	-0.7 (2.5)
site9	9.2 (3.0)
site10	-0.6 (3.8)

---

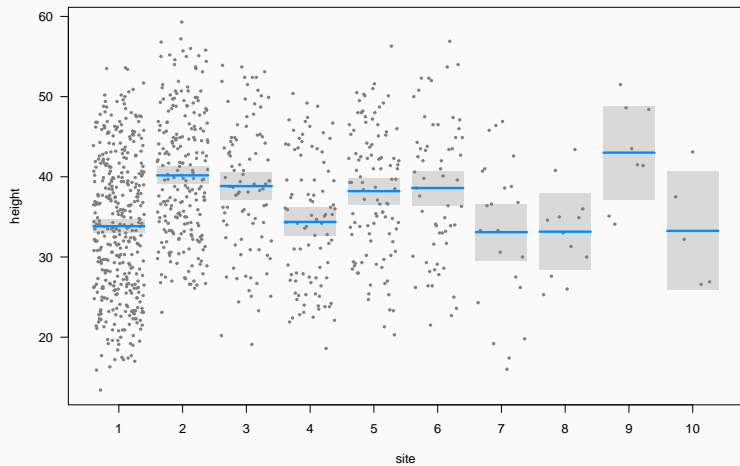
## Presenting model results

```
library('gtsummary')  
tbl_regression(m3)
```

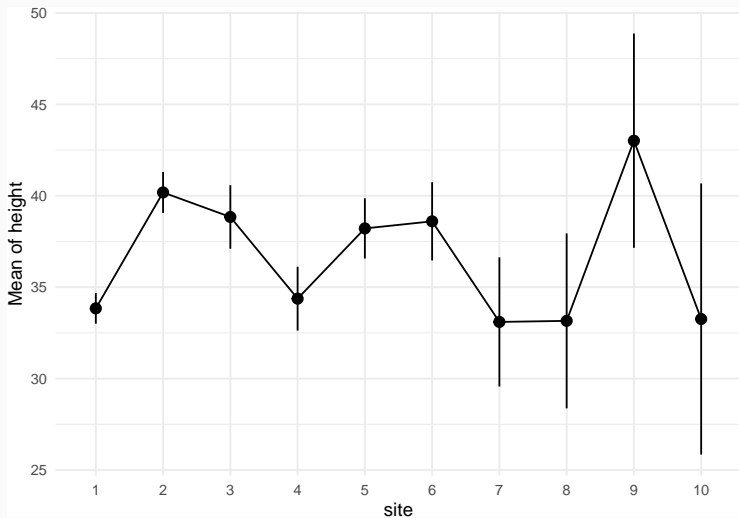
Characteristic	Beta	95% CI	p-value
site			
1	—	—	
2	6.3	4.9, 7.7	<0.001
3	5.0	3.1, 6.9	<0.001
4	0.53	-1.4, 2.5	0.6
5	4.4	2.5, 6.2	<0.001
6	4.8	2.5, 7.1	<0.001
7	-0.74	-4.4, 2.9	0.7
8	-0.68	-5.5, 4.2	0.8
9	9.2	3.3, 15	0.002
10	-0.58	-8.0, 6.9	0.9

Abbreviation: CI = Confidence Interval

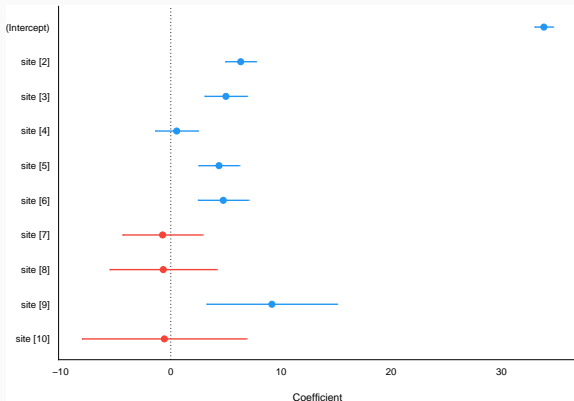
`visreg(m3)`



```
plot(estimate_means(m3))
```



```
plot(parameters(m3), show_intercept = TRUE)
```



## Fit model without intercept

```
m3bis <- lm(height ~ site - 1, data = trees)
```

Call:

```
lm(formula = height ~ site - 1, 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 )	
site1	33.8416	0.4266	79.329	<2e-16	***
site2	40.1826	0.5707	70.404	<2e-16	***
site3	38.8407	0.8854	43.868	<2e-16	***
site4	34.3744	0.8903	38.610	<2e-16	***
site5	38.2139	0.8404	45.469	<2e-16	***
site6	38.6017	1.0904	35.401	<2e-16	***
site7	33.1000	1.8007	18.381	<2e-16	***
site8	33.1583	2.4382	13.599	<2e-16	***
site9	43.0125	2.9862	14.404	<2e-16	***
site10	33.2600	3.7773	8.805	<2e-16	***

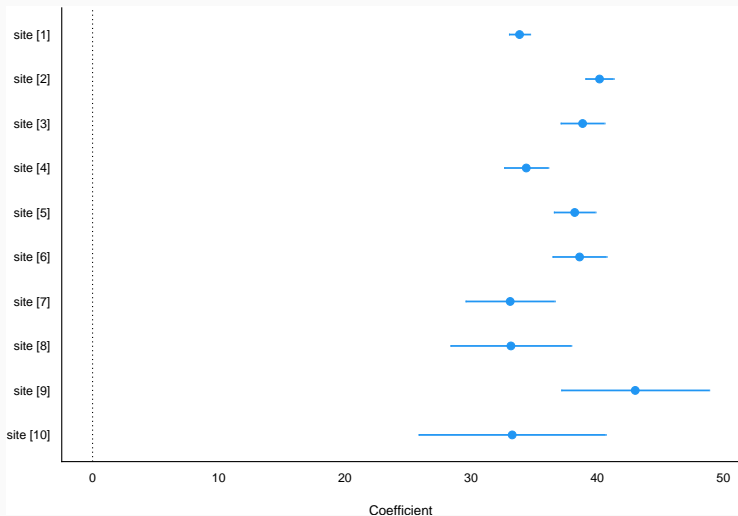
---

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

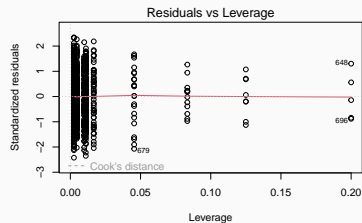
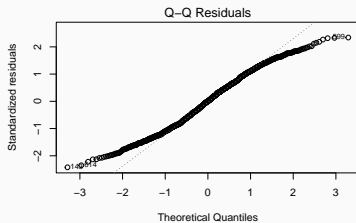
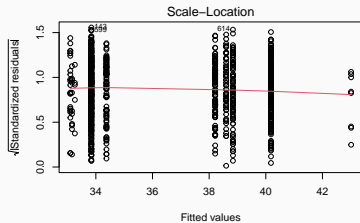
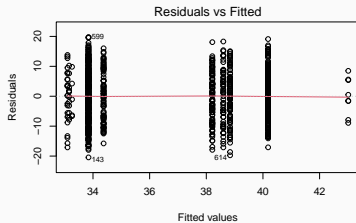
Residual standard error: 8.446 on 990 degrees of freedom

# Model without intercept

```
plot(parameters(m3bis))
```



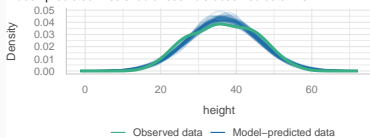
# Model checking: residuals



## check\_model(m3)

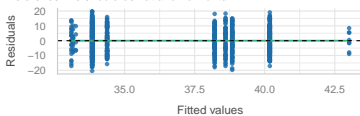
### Posterior Predictive Check

Model-predicted lines should resemble observed data line



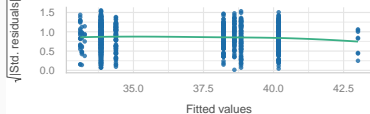
### Linearity

Reference line should be flat and horizontal



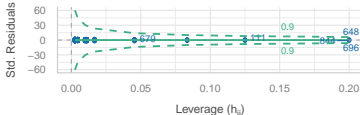
### Homogeneity of Variance

Reference line should be flat and horizontal



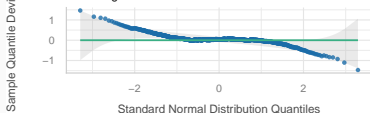
### Influential Observations

Points should be inside the contour lines



### Normality of Residuals

Points should fall along the line



## Combining continuous and categorical predictors

---

## Example dataset: forest trees

```
trees <- read.csv("data/trees.csv")
trees$site <- as.factor(trees$site)
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

```
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

## Let's read the model report...

### `report(m4)`

We fitted a linear model (estimated using OLS) to predict height with site and dbh (formula: `height ~ site + dbh`). The model explains a statistically significant and substantial proportion of variance ( $R^2 = 0.88$ ,  $F(10, 989) = 750.04$ ,  $p < .001$ , adj.  $R^2 = 0.88$ ). The model's intercept, corresponding to `site = 1` and `dbh = 0`, is at 16.70 (95% CI [16.19, 17.21],  $t(989) = 64.09$ ,  $p < .001$ ). Within this model:

- The effect of site [2] is statistically significant and positive (beta = 6.50, 95% CI [6.00, 7.01],  $t(989) = 25.34$ ,  $p < .001$ ; Std. beta = 0.73, 95% CI [0.68, 0.79])
- The effect of site [3] is statistically significant and positive (beta = 4.36, 95% CI [3.66, 5.05],  $t(989) = 12.30$ ,  $p < .001$ ; Std. beta = 0.49, 95% CI [0.41, 0.57])
- The effect of site [4] is statistically significant and positive (beta = 1.93, 95% CI [1.24, 2.63],  $t(989) = 5.43$ ,  $p < .001$ ; Std. beta = 0.22, 95% CI [0.14, 0.30])
- The effect of site [5] is statistically significant and positive (beta = 3.64, 95% CI [2.97, 4.30],  $t(989) = 10.71$ ,  $p < .001$ ; Std. beta = 0.41, 95% CI [0.33, 0.49])
- The effect of site [6] is statistically significant and positive (beta = 4.20, 95% CI [3.38, 5.03],  $t(989) = 9.97$ ,  $p < .001$ ; Std. beta = 0.47, 95% CI [0.38, 0.57])
- The effect of site [7] is statistically non-significant and negative (beta = -0.18, 95% CI [-1.48, 1.13],  $t(989) = -0.26$ ,  $p = 0.792$ ; Std. beta = -0.02, 95% CI [-0.17, 0.13])
- The effect of site [8] is statistically significant and negative (beta = -5.31, 95% CI [-7.07, -3.56],  $t(989) = -5.95$ ,  $p < .001$ ; Std. beta = -0.60, 95% CI [-0.80, -0.40])
- The effect of site [9] is statistically significant and positive (beta = 5.44, 95% CI [3.30, 7.57],  $t(989)$ )

## Presenting model results

```
parameters(m4)
```

Parameter	Coefficient	SE	95% CI	t(989)	p
(Intercept)	16.70	0.26	[16.19, 17.21]	64.09	< .001
site [2]	6.50	0.26	[ 6.00, 7.01]	25.34	< .001
site [3]	4.36	0.35	[ 3.66, 5.05]	12.30	< .001
site [4]	1.93	0.36	[ 1.24, 2.63]	5.43	< .001
site [5]	3.64	0.34	[ 2.97, 4.30]	10.71	< .001
site [6]	4.20	0.42	[ 3.38, 5.03]	9.97	< .001
site [7]	-0.18	0.67	[-1.48, 1.13]	-0.26	0.792
site [8]	-5.31	0.89	[-7.07, -3.56]	-5.95	< .001
site [9]	5.44	1.09	[ 3.30, 7.57]	5.00	< .001
site [10]	2.26	1.37	[-0.43, 4.95]	1.65	0.099
dbh	0.62	7.57e-03	[ 0.60, 0.63]	81.47	< .001

## Estimated tree heights for each site

```
estimate_means(m4)
```

Estimated Marginal Means

site	Mean	SE	95% CI	t(989)
1	33.90	0.15	[33.60, 34.21]	220.59
2	40.41	0.21	[40.01, 40.81]	196.49
3	38.26	0.32	[37.64, 38.89]	119.91
4	35.84	0.32	[35.21, 36.47]	111.55
5	37.54	0.30	[36.95, 38.14]	123.94
6	38.11	0.39	[37.34, 38.88]	96.99
7	33.73	0.65	[32.45, 35.00]	51.98
8	28.59	0.88	[26.86, 30.32]	32.48
9	39.34	1.08	[37.23, 41.45]	36.53
10	36.17	1.36	[33.50, 38.84]	26.57

## Fit model without intercept

```
m4.noint <- lm(height ~ -1 + site + dbh, data = trees)
```

Call:

```
lm(formula = height ~ -1 + site + dbh, data = trees)
```

Residuals:

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

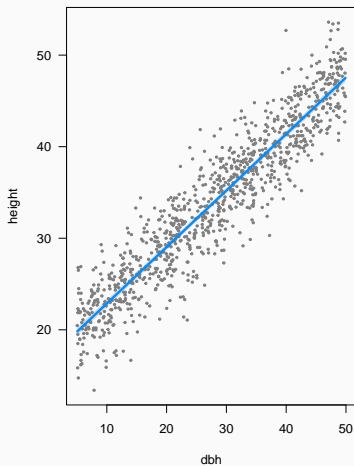
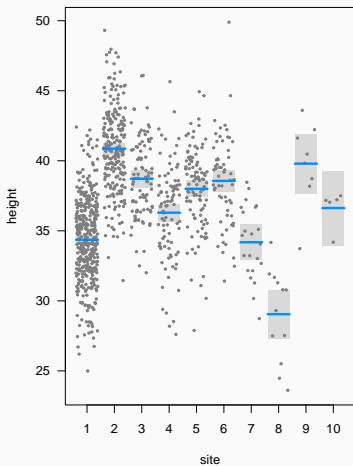
Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
site1	16.699037	0.260565	64.09	<2e-16	***
site2	23.203340	0.292773	79.25	<2e-16	***
site3	21.056494	0.386532	54.48	<2e-16	***
site4	18.633687	0.374456	49.76	<2e-16	***
site5	20.336469	0.373942	54.38	<2e-16	***
site6	20.903548	0.448913	46.56	<2e-16	***
site7	16.522844	0.679936	24.30	<2e-16	***
site8	11.386389	0.918198	12.40	<2e-16	***
site9	22.136086	1.105970	20.02	<2e-16	***
site10	18.962375	1.372158	13.82	<2e-16	***
dbh	0.617075	0.007574	81.47	<2e-16	***
---					

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

# Plot (visreg)

```
visreg(m4)
```

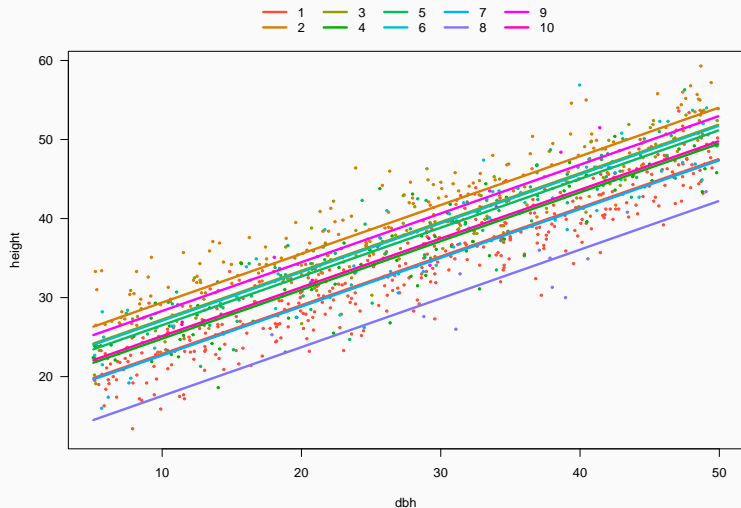


```
null device
```

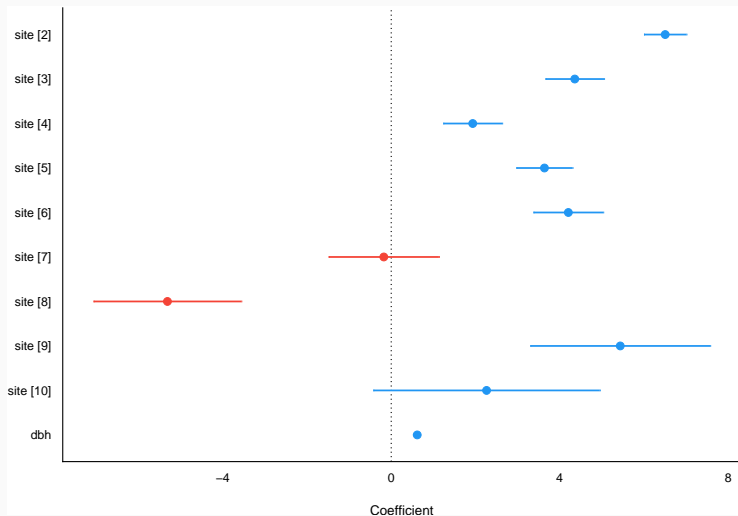
```
1
```

## Plot (visreg)

```
visreg(m4, xvar = 'dbh', by = 'site', overlay = TRUE, band = FALSE)
```



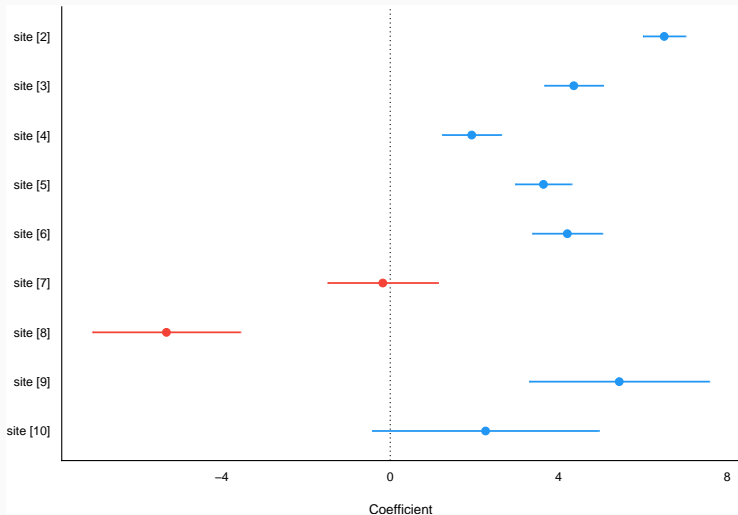
```
plot(parameters(m4))
```



## Plot model (easystats)

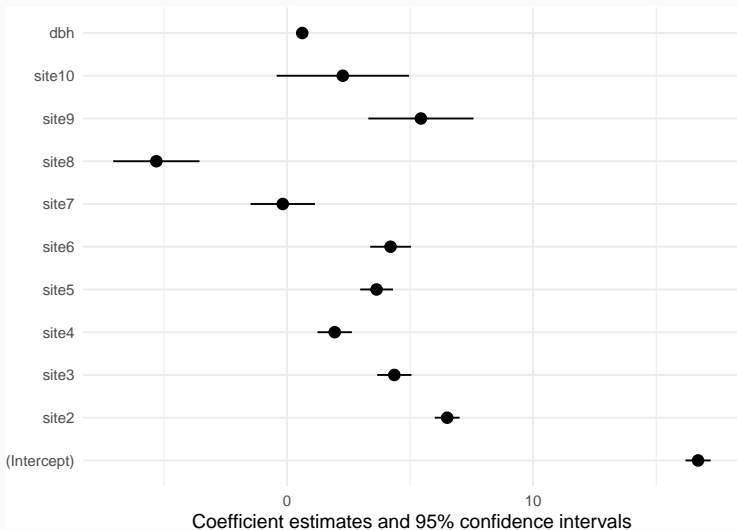
Keeping sites only, dropping 'dbh'

```
plot(parameters(m4, drop = 'dbh'))
```



# Plot model (modelsummary)

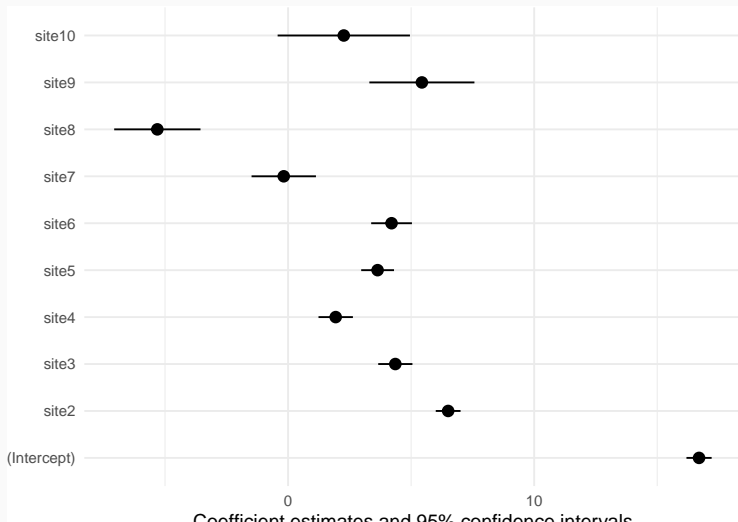
```
modelplot(m4)
```



## Plot model (modelsummary)

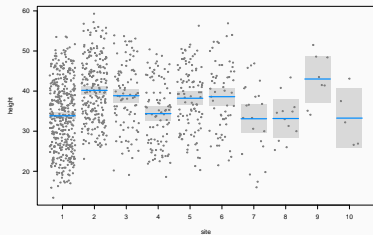
Keeping sites only, dropping 'dbh'

```
modelplot(m4, coef_omit = 'dbh')
```

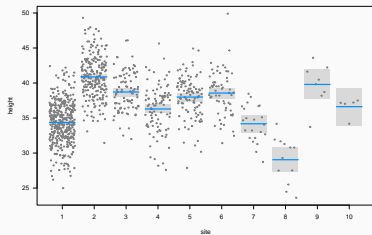


# What happened to site 8?

```
visreg(m3)
```

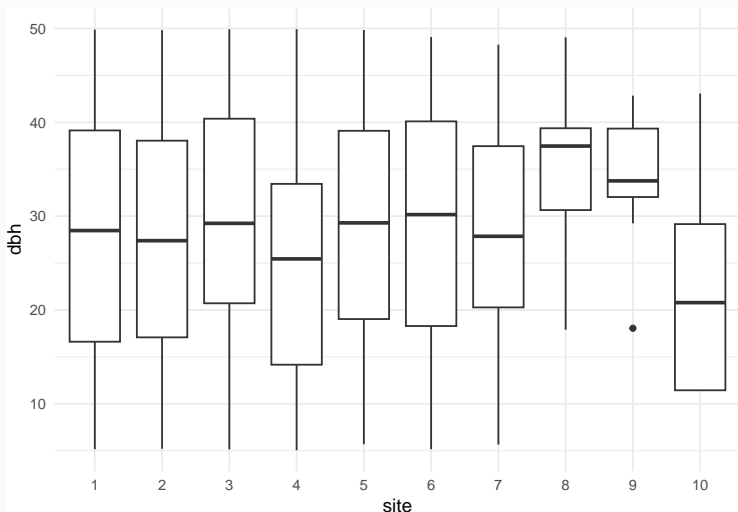


```
visreg(m4, xvar = 'site')
```



# What happened to site 8?

site 8 has the largest diameters



## What happened to site 8?

DBH

```
aggregate(trees$dbh ~ trees$site, FUN = me
```

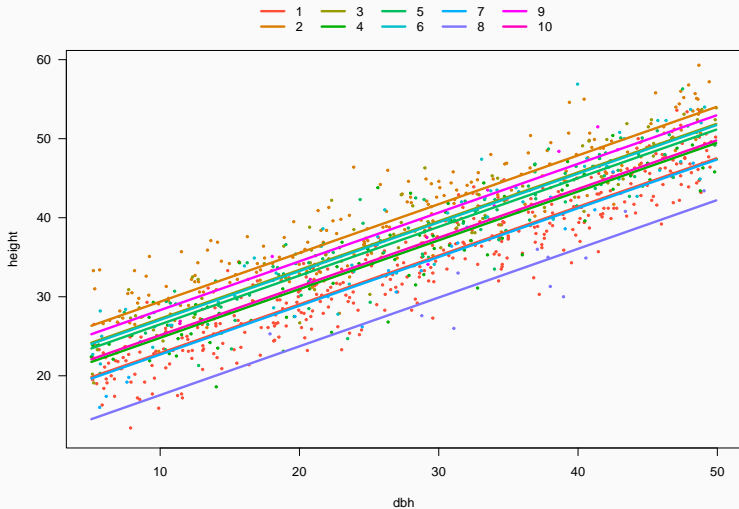
	trees\$site	trees\$dbh
1	1	27.78033
2	2	27.51580
3	3	28.82011
4	4	25.50867
5	5	28.97119
6	6	28.68067
7	7	26.86409
8	8	35.28250
9	9	33.83125
10	10	23.17000

HEIGHT

```
aggregate(trees$height ~ trees$site, FUN =
```

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

# We have fitted model w/ many intercepts and single slope

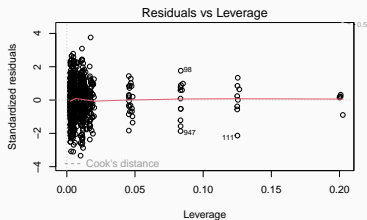
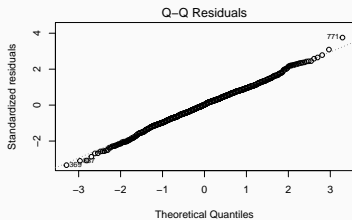
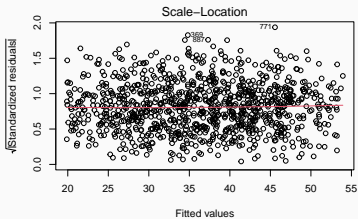
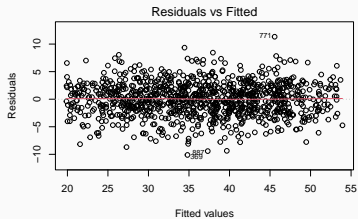


## Slope is the same for all sites

```
parameters(m4, keep = 'dbh')
```

Parameter	Coefficient	SE	95% CI	t(989)	p
dbh	0.62	7.57e-03	[0.60, 0.63]	81.47	< .001

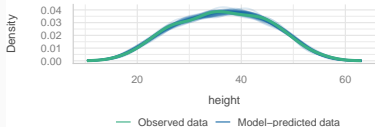
# Model checking: residuals



## check\_model(m4)

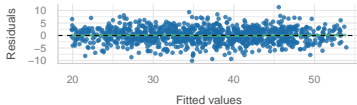
### Posterior Predictive Check

Model-predicted lines should resemble observed data line



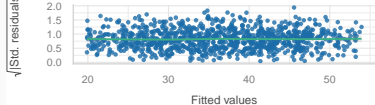
### Linearity

Reference line should be flat and horizontal



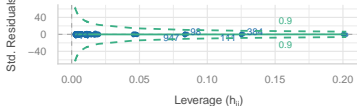
### Homogeneity of Variance

Reference line should be flat and horizontal



### Influential Observations

Points should be inside the contour lines



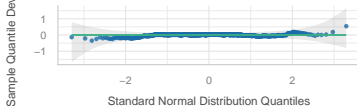
### Collinearity

High collinearity (VIF) may inflate parameter uncertainty



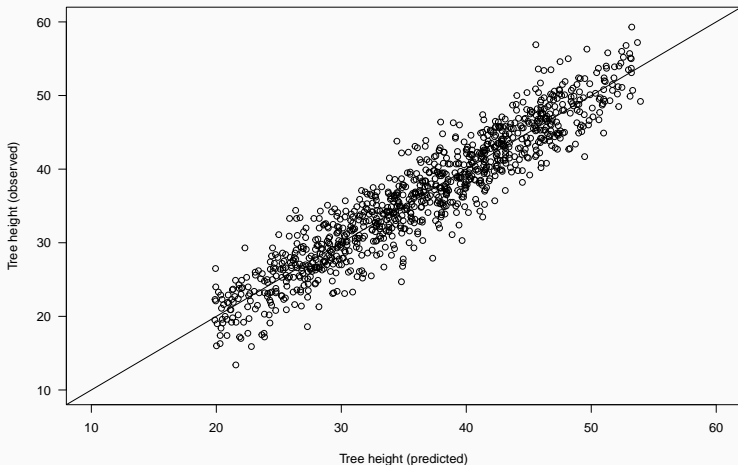
### Normality of Residuals

Plots should fall along the line



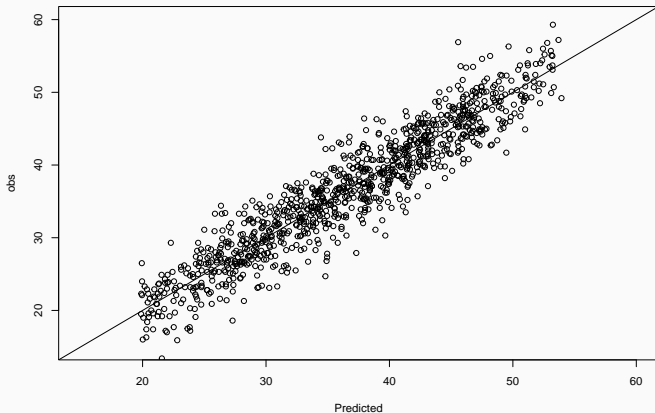
## How good is this model? Calibration plot

```
trees$height.pred <- fitted(m4)
plot(trees$height.pred, trees$height, xlab = 'Tree height (predicted)')
abline(a = 0, b = 1)
```



## How good is this model? Calibration plot (easystats)

```
pred <- estimate_expectation(m4)
pred$obs <- trees$height
plot(obs ~ Predicted, data = pred, xlim = c(15, 60), ylim = c(15, 60),
      abline(a = 0, b = 1))
```

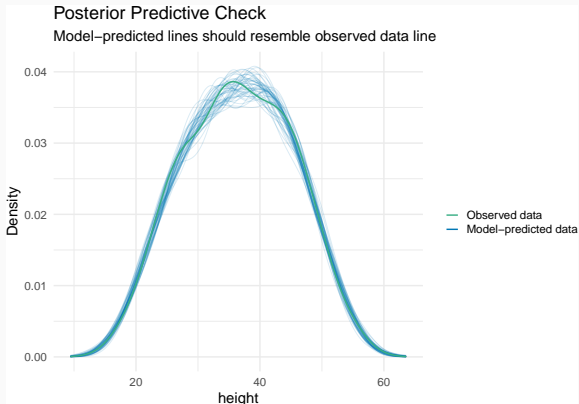


## Posterior predictive checking

Simulating response data from fitted model ( $y_{rep}$ )

and comparing with observed response ( $y$ )

```
performance::check_predictions(m4)
```



## Predicting heights of new trees

---

## Using model for prediction

Expected height of 10-cm diameter tree in each site?

```
trees.10cm <- data.frame(site = as.factor(1:10),  
                          dbh = 10)
```

```
trees.10cm
```

	site	dbh
1	1	10
2	2	10
3	3	10
4	4	10
5	5	10
6	6	10
7	7	10
8	8	10
9	9	10
10	10	10

## Using model for prediction

Confidence interval

```
predict(m4, newdata = trees.10cm, interval = 'confidence')
```

	fit	lwr	upr
1	22.86979	22.46878	23.27079
2	29.37409	28.89388	29.85430
3	27.22724	26.54160	27.91289
4	24.80444	24.13410	25.47477
5	26.50722	25.84952	27.16492
6	27.07430	26.25490	27.89370
7	22.69359	21.39601	23.99117
8	17.55714	15.79282	19.32146
9	28.30683	26.16606	30.44761
10	25.13312	22.45540	27.81085

## Using model for prediction

Prediction interval (accounting for residual variance)

```
predict(m4, newdata = trees.10cm, interval = 'prediction')
```

	fit	lwr	upr
1	22.86979	16.88478	28.85480
2	29.37409	23.38325	35.36493
3	27.22724	21.21645	33.23804
4	24.80444	18.79537	30.81350
5	26.50722	20.49955	32.51489
6	27.07430	21.04678	33.10181
7	22.69359	16.58268	28.80451
8	17.55714	11.33039	23.78388
9	28.30683	21.96314	34.65053
10	25.13312	18.58868	31.67757

## Using model for prediction

Prediction interval (99%)

```
predict(m4, newdata = trees.10cm, interval = 'prediction',  
        level = 0.99)
```

	fit	lwr	upr
1	22.86979	14.998587	30.74098
2	29.37409	21.495225	37.25295
3	27.22724	19.322133	35.13235
4	24.80444	16.901598	32.70727
5	26.50722	18.606216	34.40822
6	27.07430	19.147195	35.00140
7	22.69359	14.656813	30.73037
8	17.55714	9.368019	25.74626
9	28.30683	19.963913	36.64976
10	25.13312	16.526183	33.74007

## Predicting heights of new trees (easystats)

---

## Using model for prediction

Expected height of 10-cm diameter tree in each site?

```
trees.10cm <- data.frame(site = as.factor(1:10),  
                          dbh = 10)
```

```
trees.10cm
```

	site	dbh
1	1	10
2	2	10
3	3	10
4	4	10
5	5	10
6	6	10
7	7	10
8	8	10
9	9	10
10	10	10

## Using model for prediction

Expected height of 10-cm DBH trees at each site

```
pred <- estimate_expectation(m4, data = trees.10cm)
```

Model-based Predictions

site	dbh	Predicted	SE	95% CI
1	10	22.87	0.20	[22.47, 23.27]
2	10	29.37	0.24	[28.89, 29.85]
3	10	27.23	0.35	[26.54, 27.91]
4	10	24.80	0.34	[24.13, 25.47]
5	10	26.51	0.34	[25.85, 27.16]
6	10	27.07	0.42	[26.25, 27.89]
7	10	22.69	0.66	[21.40, 23.99]
8	10	17.56	0.90	[15.79, 19.32]
9	10	28.31	1.09	[26.17, 30.45]
10	10	25.13	1.36	[22.46, 27.81]

Variable predicted: height

## Using model for prediction

Prediction intervals (accounting for residual variance)

```
pred <- estimate_prediction(m4, data = trees.10cm)
```

Model-based Predictions

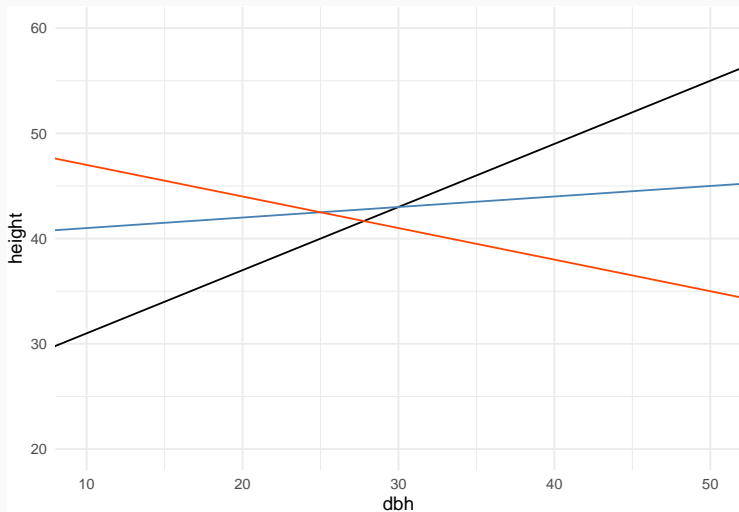
site	dbh	Predicted	SE	95% CI
1	10	22.87	3.05	[16.88, 28.85]
2	10	29.37	3.05	[23.38, 35.36]
3	10	27.23	3.06	[21.22, 33.24]
4	10	24.80	3.06	[18.80, 30.81]
5	10	26.51	3.06	[20.50, 32.51]
6	10	27.07	3.07	[21.05, 33.10]
7	10	22.69	3.11	[16.58, 28.80]
8	10	17.56	3.17	[11.33, 23.78]
9	10	28.31	3.23	[21.96, 34.65]
10	10	25.13	3.33	[18.59, 31.68]

Variable predicted: height

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

---

Does allometric relationship between Height and Diameter 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

---

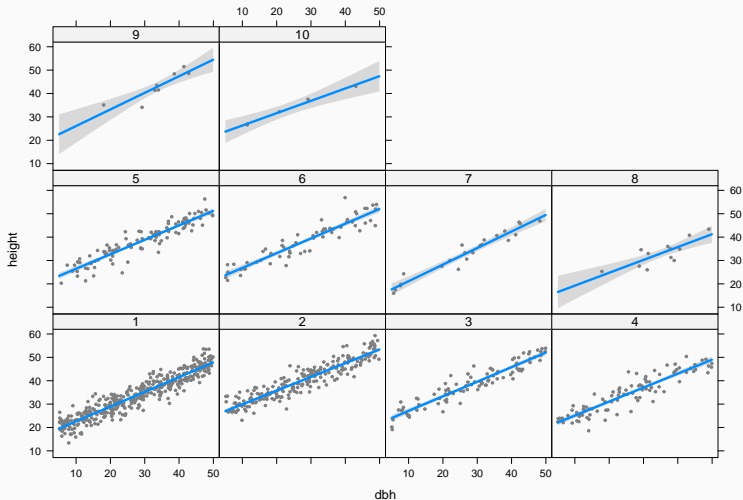
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.041 on 980 degrees of freedom

Multiple R-squared: 0.8847      Adjusted R-squared: 0.8825

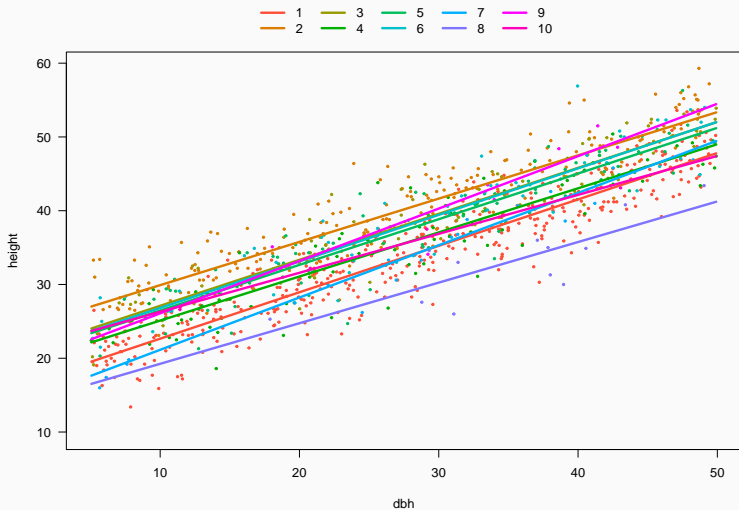
## Does slope vary among sites?

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



## Does slope vary among sites?

```
visreg(m5, xvar = 'dbh', by = 'site', overlay = TRUE, band = FALSE)
```



## Does slope vary among sites?

```
library('marginaleffects')  
hypotheses(m5, '`site9:dbh` = `site10:dbh`')
```

Hypothesis	Estimate	Std. Error	z	Pr(> z )	S	2.5 %	97.5 %
'site9:dbh'='site10:dbh'	0.182	0.185	0.983	0.326	1.6	-0.181	0.545

```
library('modelStudio')
m5.explain <- DALEX::explain(
  m5,
  data = trees,
  y = trees$height)
modelStudio(m5.explain, viewer = 'browser')
```

- [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...

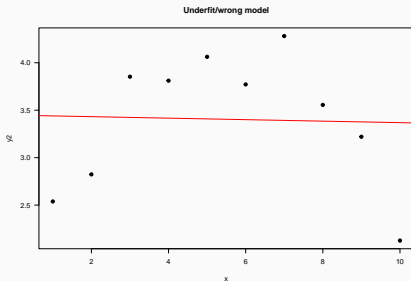
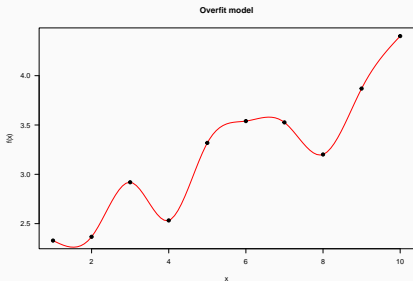
## Variable and model selection

---

# Overfitting and balanced model complexity

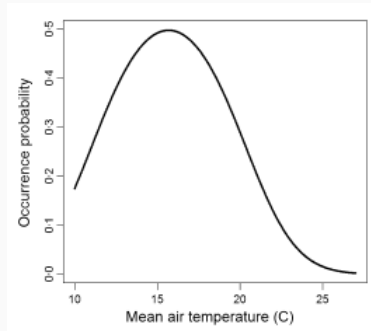
On one hand, we want to **maximise fit**.

On the other hand, we want to **avoid overfitting** and overly complex models.

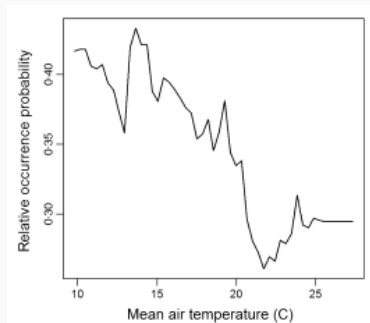


# Overfitting and balanced model complexity

GLMM



Random forests



Wenger & Olden (2012)

## Overfitted models will work badly on new data



- **Cross-validation** (k-fold, leave one out...)
- **Information Criteria:**
  - AIC
  - BIC
  - DIC
  - WAIC...
- All these methods have flaws!

$$AIC = -2 * \text{LogLikelihood} + 2K$$

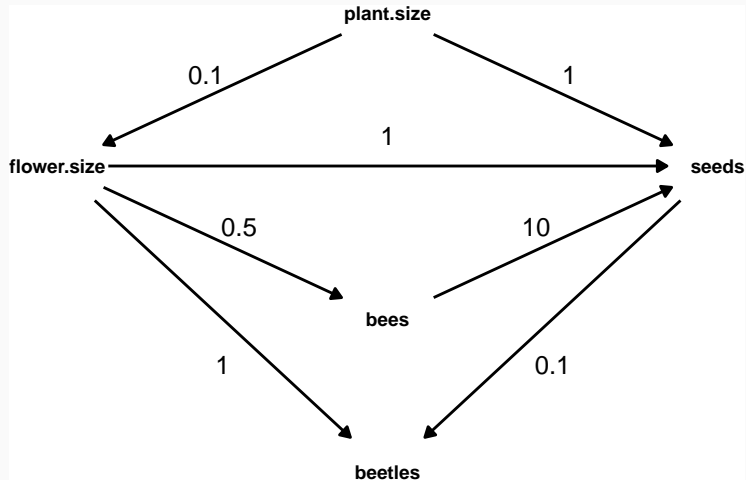
- First term: **model fit**
- **K = number of parameters** (penalisation for model complexity)
- **Lower is better**
- AIC **biased** towards complex models.
- **AICc** recommended with 'small' sample sizes ( $n/p < 40$ ). But see [Richards 2005](#)

- No information criteria is panacea: all have problems.
- They estimate **average** out-of-sample prediction error. But errors can differ substantially within dataset.
- Sometimes better models rank poorly (e.g. see [Gelman et al. 2013](#)). Combine with **thorough model checks**.
- Predictive criteria don't help for **causal inference**.

Predictive criteria don't help for  
causal inference

---

## Predictive criteria don't help for causal inference



## Predictive criteria don't help to choose correct causal model

Making good predictions doesn't require accurate causal model

Model	AIC	R2
m.flower	933.3	0.5
m.flower.plant	913.2	0.6
m.flower.plant.bees	899.1	0.7
m.flower.plant.bees.beetles	829.9	0.8

'Best model' (based on AIC or R2) not good for causal inference

So which variables should enter  
my model?

---

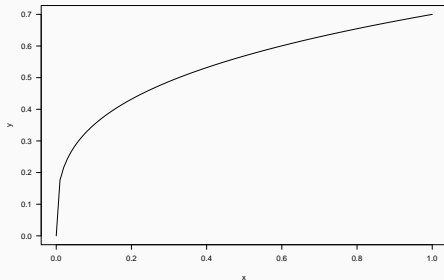
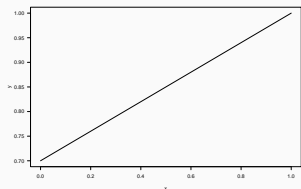
- Choose variables based on **background knowledge**, rather than throwing plenty of them in a fishing expedition.
- **Avoid** (forward) **stepwise** methods!
- Propose **single global model** or small set (< 10 - 20) of **reasonable** candidate models.
- Number of variables **balanced with sample size** (e.g. at least 10 - 30 obs per param)
- For predictors with large effects, **consider interactions**.

# 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.



## Removing predictors

---

- Model w/ highly correlated predictors -> parameter **uncertainty** will increase (SE)
- $VIF > 1$  -> multicollinearity
- $VIF > 5$  or  $10$  indicate strong collinearity
- Should we remove highly collinear predictors?

Collinearity isn't a disease that needs curing

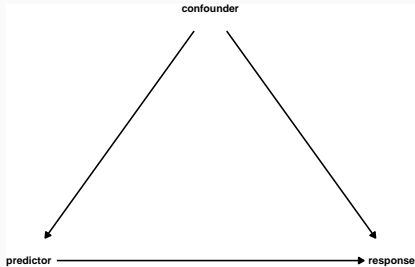
The Meaning of Multiple Regression and the Non-Problem of Collinearity

The VIF Score. What is it Good For? Absolutely Nothing

Additional caution regarding rules of thumb for VIF

# Removing predictors

- If **predictive** goal, just consider predictive performance
- If **explanatory** goal, consider causal structure of your problem (DAG)



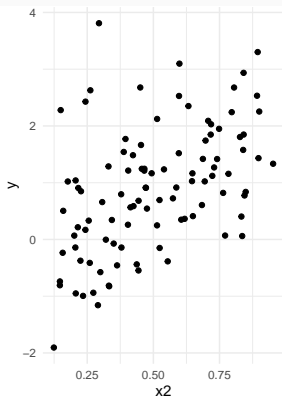
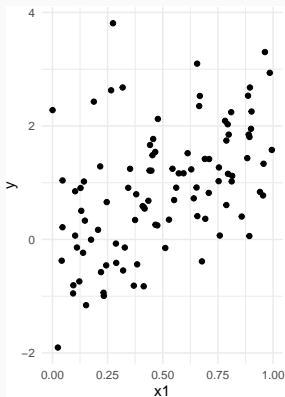
## Stepwise regression has many problems

- Whittingham et al. (2006) **Why do we still use** stepwise modelling in ecology and behaviour? J. Animal Ecology.
- Mundry & Nunn (2009) Stepwise Model Fitting and Statistical Inference: **Turning Noise into Signal Pollution**. Am Nat.
- This includes **stepAIC** (e.g. Dahlgren 2010; Burnham et al 2011; Hegyi & Garamszegi 2011).

## Simpler (best) model provides biased causal estimates

Simulate response depending on two correlated variables ([Hartig 2022](#))

```
x1 = runif(100)
x2 = 0.8*x1 + 0.2*runif(100)
y = x1 + x2 + rnorm(100)
```



# Simpler (best) model provides biased causal estimates

Simulate response depending on two correlated variables [\(Hartig 2022\)](#)

```
fullmodel = lm(y ~ x1 + x2)
```

Call:

```
lm(formula = y ~ x1 + x2)
```

Residuals:

Min	1Q	Median	3Q	Max
-1.8994	-0.6821	-0.1086	0.5749	3.3663

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-0.1408	0.2862	-0.492	0.624
x1	1.2158	1.5037	0.809	0.421
x2	0.8518	1.8674	0.456	0.649

Residual standard error: 0.9765 on 97 degrees of freedom

Multiple R-squared: 0.237, Adjusted R-squared: 0.2212

F-statistic: 15.06 on 2 and 97 DF, p-value: 2.009e-06

## Simpler (best) model provides biased causal estimates

```
simplemodel = MASS::stepAIC(fullmodel, trace = 0)
```

Call:

```
lm(formula = y ~ x1)
```

Residuals:

Min	1Q	Median	3Q	Max
-1.9047	-0.6292	-0.1019	0.6077	3.3394

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-0.04633	0.19670	-0.236	0.814
x1	1.88350	0.34295	5.492	3.13e-07 ***

---

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

Residual standard error: 0.9725 on 98 degrees of freedom

Multiple R-squared: 0.2353, Adjusted R-squared: 0.2275

F-statistic: 30.16 on 1 and 98 DF, p-value: 3.134e-07

Simulating data with 10 random predictors

```
dat <- data.frame(y = rnorm(100),  
                  x = matrix(runif(1000), ncol = 10))
```

y	x.1	x.2	x.3	x.4	x.5	x.6	x.7	x.8	x.9	x.10
-0.1	0.6	0.6	0.3	0.8	0.2	0.0	0.4	0.4	0.3	0.2
0.8	0.4	0.4	0.9	0.2	0.5	0.1	0.6	0.2	0.0	0.0
-0.5	0.0	0.3	0.4	0.3	0.1	0.1	0.9	0.9	0.5	0.8
-0.6	0.7	0.7	0.4	0.5	0.2	0.7	0.7	0.8	0.5	0.3
0.7	0.0	0.6	0.9	0.1	0.2	0.4	0.8	0.6	0.6	0.1
-0.1	0.4	0.2	0.9	0.4	0.6	0.5	0.9	0.1	0.8	0.8

Running `MuMIn::dredge` with 10 random predictors

```
full.model <- lm(y ~ ., data = dat)
dd <- MuMIn::dredge(full.model)
```

Best model:

Parameter	Coefficient	SE	p
(Intercept)	-1.50	0.36	0.00
x.2	0.78	0.36	0.03
x.5	0.59	0.32	0.07
x.6	0.61	0.35	0.09
x.9	0.87	0.34	0.01

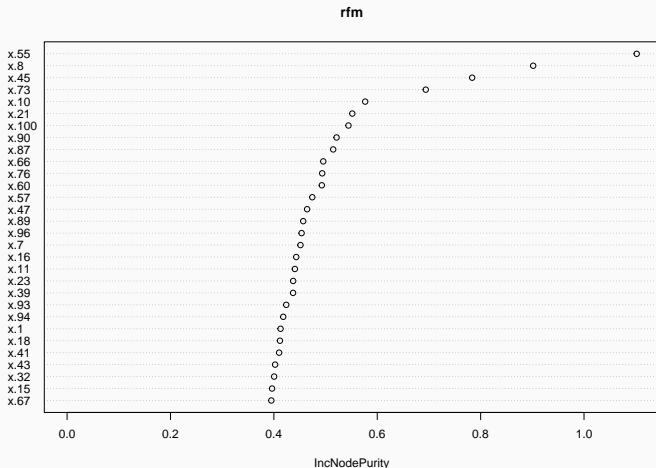
*“Let the computer find out” is a poor strategy and usually reflects the fact that the researcher did not bother to think clearly about the problem of interest and its scientific setting*

Burnham and Anderson 2002

# Variable importance in machine learning

## Random forest on 100 random predictors

```
dat <- data.frame(x = matrix(runif(50000), ncol = 100), y = runif(500))  
rfm <- randomForest::randomForest(y ~ ., data = dat)  
varImpPlot(rfm)
```



- Testing bivariate relationships before building multivariable model
- Removing non-significant predictors

Heinze & Dunkler 2016

- Always **keep 'core' predictors** (based on previous knowledge)
- If ratio sample size/number of predictors is low ( $<10$  EPP), avoid variable selection (too unstable)
- If performing variable selection, always **assess stability** (bootstrap, etc)

Heinze et al 2018

1. Choose meaningful variables
  - Keep good  $n/p$  ratio
2. Generate global model or (small) set of candidate models
  - Avoid stepwise and all-subsets
  - Don't assume linear effects: think about appropriate functional relationships
  - Consider interactions for strong main effects
3. If  $> 1$  model have similar support, consider model averaging (or blending).
4. Always check fitted models thoroughly
5. Always report effect sizes

## Model comparison

---

```
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

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

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

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

```
m4 <- lm(height ~ site*dbh, data = trees)
```

# Compare model performance

```
library('performance')  
compare_performance(m1, m2, m3, m4)
```

```
# Comparison of Model Performance Indices
```

Name	Model	AIC (weights)	AICc (weights)	BIC (weights)	R2
m1	lm	5660.3 (<.001)	5660.3 (<.001)	5675.0 (<.001)	0.787
m2	lm	7206.1 (<.001)	7206.2 (<.001)	7220.9 (<.001)	0.002
m3	lm	7117.3 (<.001)	7117.5 (<.001)	7171.2 (<.001)	0.102
m4	lm	5084.3 (>.999)	5085.2 (>.999)	5187.3 (>.999)	0.885

Name	R2 (adj.)	RMSE	Sigma
m1	0.787	4.089	4.093
m2	0.001	8.856	8.865
m3	0.093	8.404	8.446
m4	0.882	3.011	3.041

# Compare model performance

```
library('see')  
plot(compare_performance(m1, m2, m3, m4))
```

Comparison of Model Indices



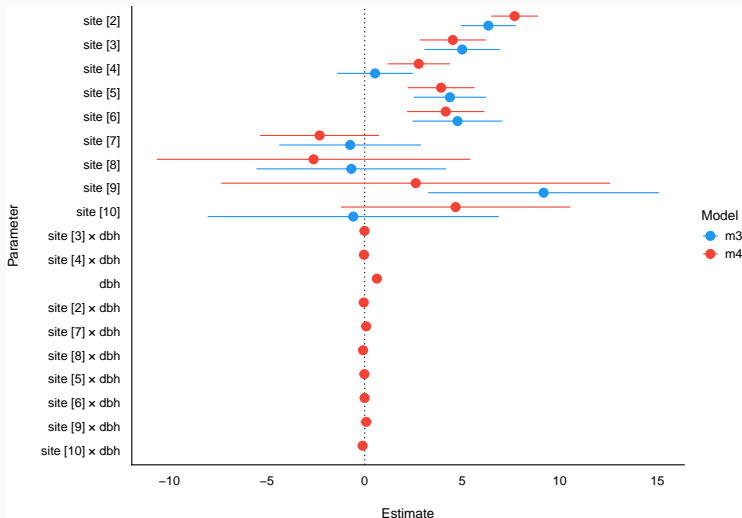
# Compare parameters

```
library('parameters')  
compare_parameters(m3, m4)
```

Parameter	m3	m4
(Intercept)	33.84 (33.00, 34.68)	16.36 ( 15.65, 17.07)
site [2]	6.34 ( 4.94, 7.74)	7.68 ( 6.49, 8.88)
site [3]	5.00 ( 3.07, 6.93)	4.52 ( 2.82, 6.22)
site [4]	0.53 (-1.40, 2.47)	2.77 ( 1.17, 4.37)
site [5]	4.37 ( 2.52, 6.22)	3.92 ( 2.21, 5.63)
site [6]	4.76 ( 2.46, 7.06)	4.16 ( 2.17, 6.14)
site [7]	-0.74 (-4.37, 2.89)	-2.31 ( -5.35, 0.74)
site [8]	-0.68 (-5.54, 4.17)	-2.62 (-10.64, 5.41)
site [9]	9.17 ( 3.25, 15.09)	2.62 ( -7.34, 12.58)
site [10]	-0.58 (-8.04, 6.88)	4.66 ( -1.21, 10.53)
site [3] × dbh		-6.03e-03 ( -0.06, 0.05)
site [4] × dbh		-0.03 ( -0.09, 0.02)
dbh		0.63 ( 0.61, 0.65)
site [2] × dbh		-0.04 ( -0.08, 0.00)
site [7] × dbh		0.08 ( -0.02, 0.18)
site [8] × dbh		-0.08 ( -0.30, 0.14)
site [5] × dbh		-0.01 ( -0.06, 0.04)
site [6] × dbh		1.34e-03 ( -0.06, 0.06)
site [9] × dbh		0.08 ( -0.21, 0.37)
site [10] × dbh		-0.10 ( -0.33, 0.12)

# Compare parameters

```
library('parameters')  
plot(compare_parameters(m3, m4))
```



## Generalised Linear Models

---

## Binomial GLM (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: Did passenger class influence survival?

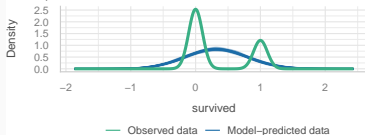
<https://pollev.com/franciscorod726>

# Let's check linear model:

```
m5 <- lm(survived ~ class, data = titanic)
library('easystats')
check_model(m5)
```

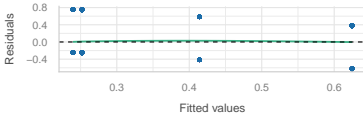
## Posterior Predictive Check

Model-predicted lines should resemble observed data line



## Linearity

Reference line should be flat and horizontal



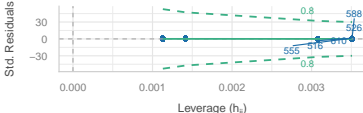
## Homogeneity of Variance

Reference line should be flat and horizontal



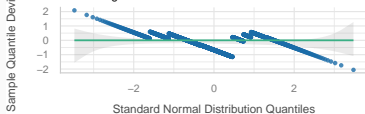
## Influential Observations

Points should be inside the contour lines

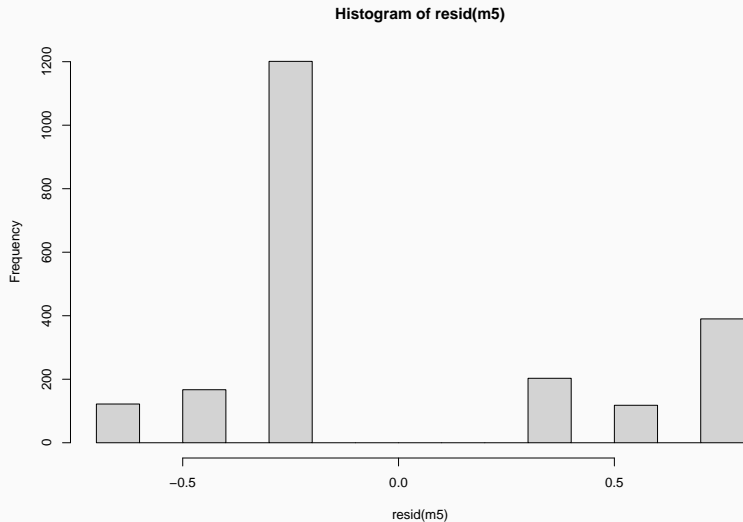


## Normality of Residuals

Points should fall along the line



# Weird residuals!



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

- Binary variables (0/1)
- Counts (0, 1, 2, 3, ...)
- Categories ('small', 'medium', 'large'...)
  
- **Generalised Linear Models to the rescue!**

## 1. Response variable - distribution family

- Bernoulli - Binomial
- Poisson
- Gamma
- etc

## 2. Predictors (continuous or categorical)

## 3. Link function

- Gaussian: identity
- Binomial: logit, probit
- Poisson: log..
- See **family**.

# The modelling process

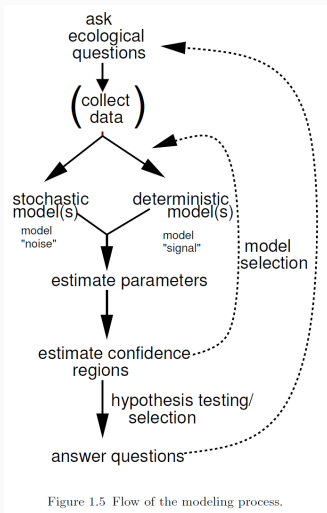


Figure 1.5 Flow of the modeling process.

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

Canonical link function: **logit** (*log odds*), but others possible (see **family**)

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

Then

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

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

## Where is the variance?

In a Gaussian GLM

$$y \sim \text{Normal}(\mu, \sigma)$$

In a Binomial GLM

$$y \sim \text{Binomial}(n, p)$$

**n** = number of trials

**p** = probability of success

$$\text{Var}(y) = np(1 - p)$$

(maximum variance when **p** around 0.5)

## 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

## How many survived in each class? (*dplyr*)

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

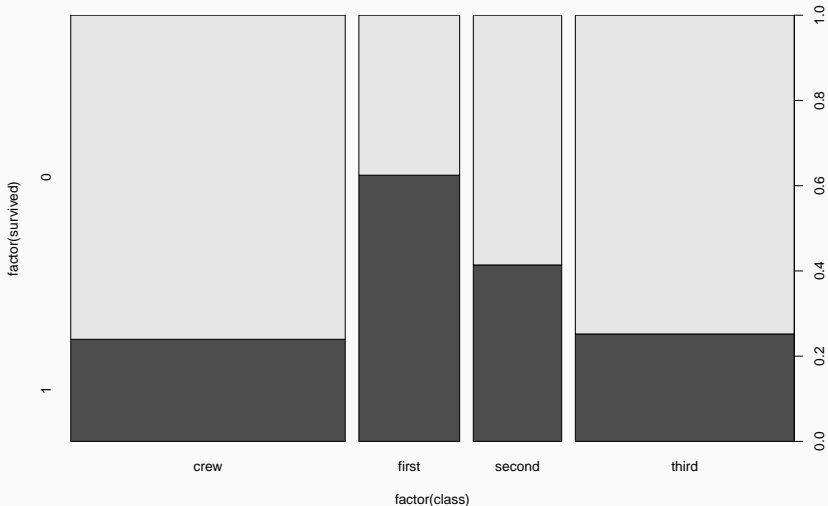
```
# A tibble: 8 x 3
```

```
# 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

## Data visualisation (mosaic plot)

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



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

which corresponds to

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

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

# Interpreting binomial GLM

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

Call:

```
glm(formula = survived ~ class, family = binomial, data = titanic)
```

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 ***
classecond	0.80785	0.14375	5.620	1.91e-08 ***
classthrd	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

We need to **back-transform** (apply *inverse logit*):

- Manually: `plogis`
- Automatically: `easystats`, etc.

## Interpreting logistic regression output (easystats)

```
library('easystats') # 'modelbased' pkg
estimate_means(tit.glm)
```

Estimated Marginal Means

class	Probability	95% CI
crew	0.24	[0.21, 0.27]
first	0.62	[0.57, 0.68]
second	0.41	[0.36, 0.47]
third	0.25	[0.22, 0.29]

Variable predicted: survived

Predictors modulated: class

Predictions are on the response-scale.

## Analysing differences among factor levels (class)

```
estimate_contrasts(tit.glm)
```

### Marginal Contrasts Analysis

Level1	Level2	Difference	SE	95% CI	z	p
first	crew	0.39	0.03	[ 0.33, 0.44]	12.65	< .001
second	crew	0.17	0.03	[ 0.11, 0.24]	5.37	< .001
third	crew	0.01	0.02	[-0.03, 0.06]	0.58	0.563
second	first	-0.21	0.04	[-0.29, -0.13]	-5.31	< .001
third	first	-0.37	0.03	[-0.43, -0.31]	-11.85	< .001
third	second	-0.16	0.03	[-0.23, -0.10]	-4.84	< .001

Variable predicted: survived

Predictors contrasted: class

p-values are uncorrected.

Contrasts are on the response-scale.

```
library('easystats') # 'performance' pkg  
r2(tit.glm)
```

```
# R2 for Logistic Regression  
Tjur's R2: 0.087
```

But there are caveats (e.g. see [here](#) and [here](#))

```
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
classecond	0.81	0.14	5.62	0.00
classthir	0.07	0.12	0.58	0.56

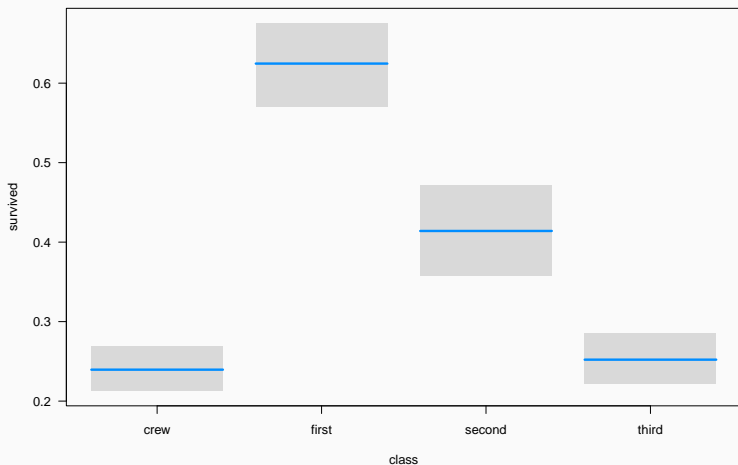
## Presenting model results

```
library('modelsummary')  
modelsummary(tit.glm, output = 'markdown')
```

	(1)
(Intercept)	-1.155 (0.079)
classfirst	1.664 (0.139)
classecond	0.808 (0.144)
classtthird	0.068 (0.117)
Num.Obs.	2201
AIC	2596.6
BIC	2619.3
Log.Lik.	-1294.278
F	57.743
RMSE	0.45

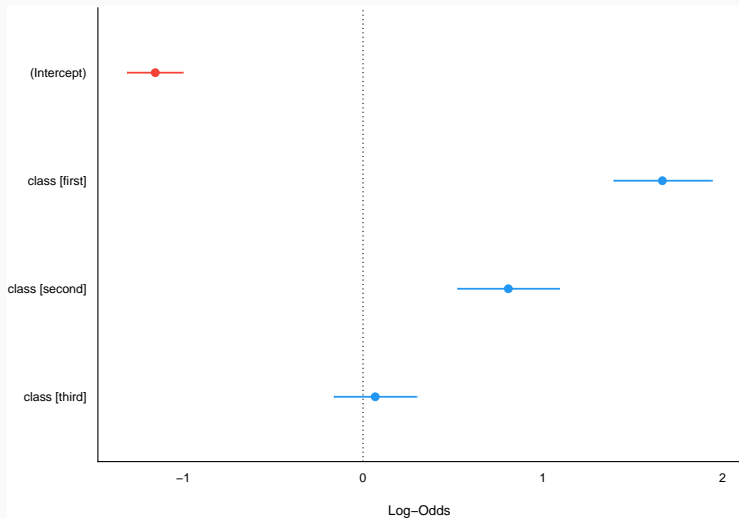
## Visualising model: visreg package

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



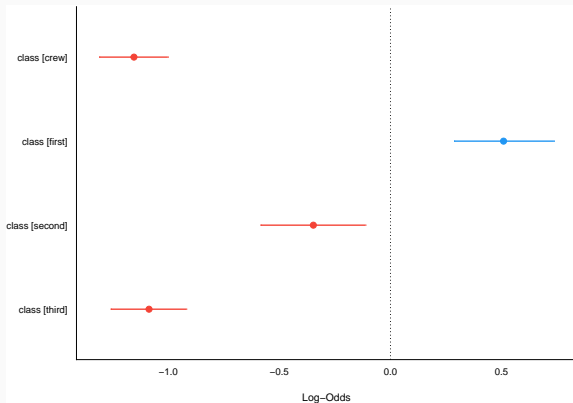
## Visualising model: easystats (see package)

```
plot(parameters(tit.glm), show_intercept = TRUE)
```



## Model without intercept

```
no.intercept <- glm(survived ~ class - 1, family = binomial, data =  
plot(parameters(no.intercept))
```

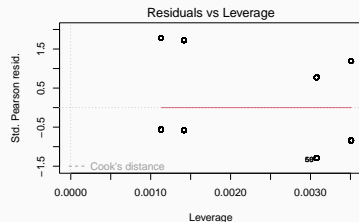
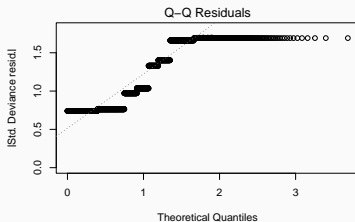
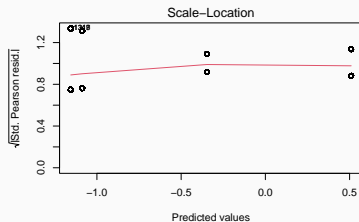
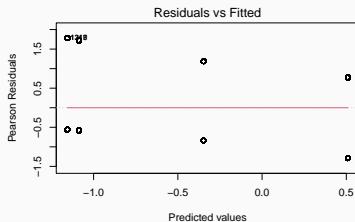


## Model checking

---

# plot(model) not very useful with binomial GLM

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



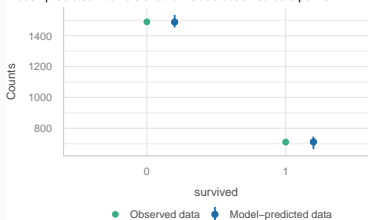
```
null device
```

```
1
```

## check\_model(tit.glm)

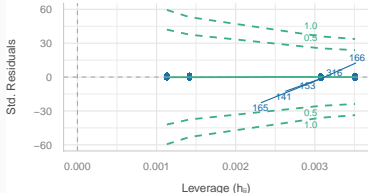
### Posterior Predictive Check

Model-predicted intervals should include observed data points



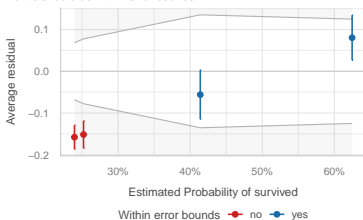
### Influential Observations

Points should be inside the contour lines



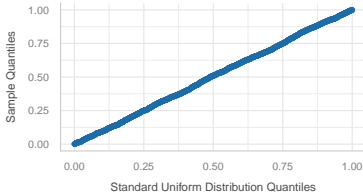
### Binned Residuals

Points should be within error bounds



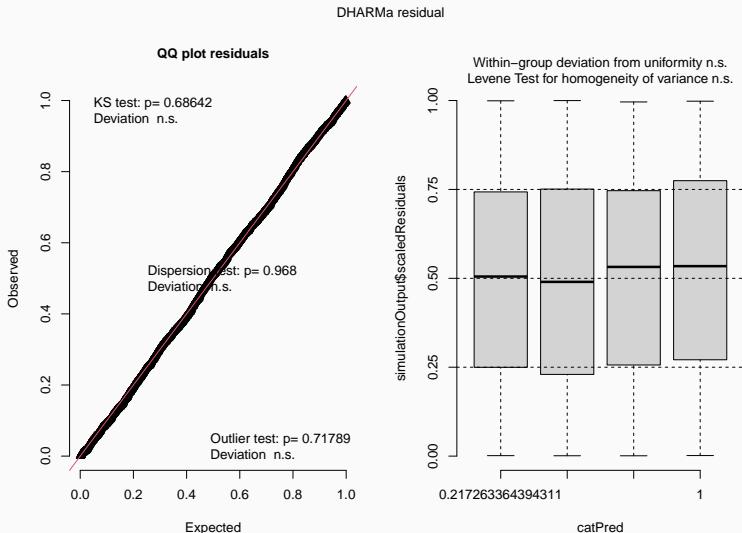
### Distribution of Quantile Residuals

Dots should fall along the line



# Residual diagnostics with DHARMa

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

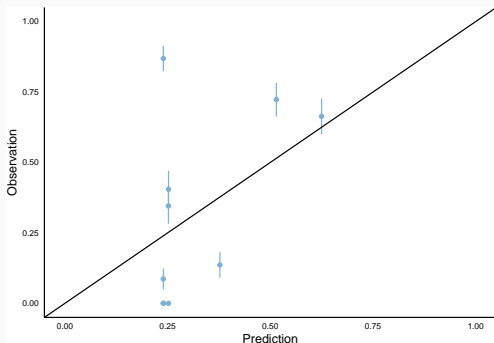


# Calibration plot

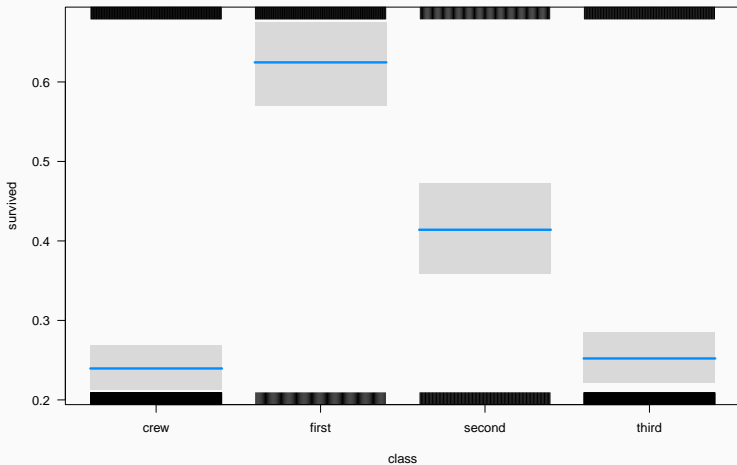
Compares predicted vs observed probabilities (grouped by quantiles)

```
library('predtools')
titanic$surv.pred <- predict(tit.glm, type = 'response')
calibration_plot(data = titanic, obs = 'survived', pred = 'surv.pred',
                 x_lim = c(0,1), y_lim = c(0,1))
```

\$calibration\_plot



Passenger class was important, but lots of unexplained variation



The goal is not to test whether the model's assumptions are 'true', because all models are false.

Rather, the goal is to assess exactly **how the model fails to describe the data**, as a path towards **model comprehension, revision, and improvement**.

Richard McElreath. *Statistical Rethinking*

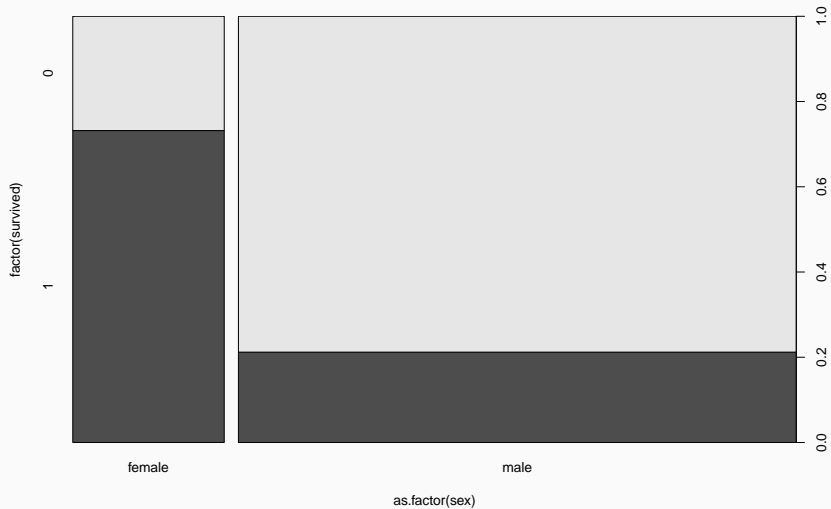
1. Visualise data
2. Fit model: `glm`. Don't forget to specify `family`!
3. Examine model: `summary`
4. Back-transform parameters from *logit* into probability scale  
(`estimate_means`)
5. Plot model: `visreg`, ...
6. Check model: `check_model`, `DHARMA::simulateResiduals`,  
`calibration_plot`

Q: Did men have higher survival  
than women?

---

<https://pollev.com/franciscorod726>

## First, visualise data



Call:

```
glm(formula = survived ~ sex, family = binomial, data = titanic)
```

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

Number of Fisher Scoring iterations: 4

# Model interpretation

```
estimate_means(tit.sex)
```

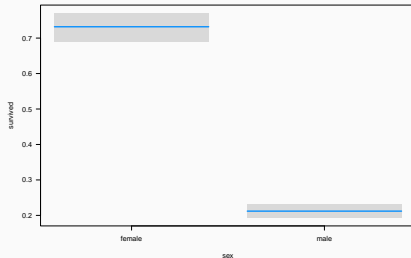
Estimated Marginal Means

sex	Probability	95% CI
female	0.73	[0.69, 0.77]
male	0.21	[0.19, 0.23]

Variable predicted: survived

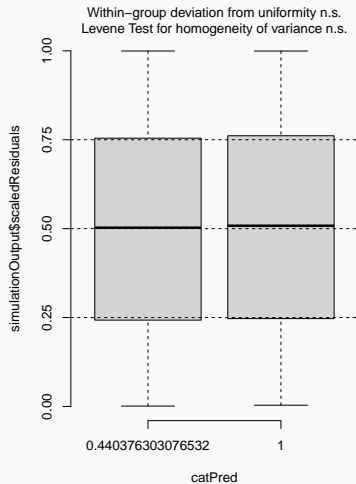
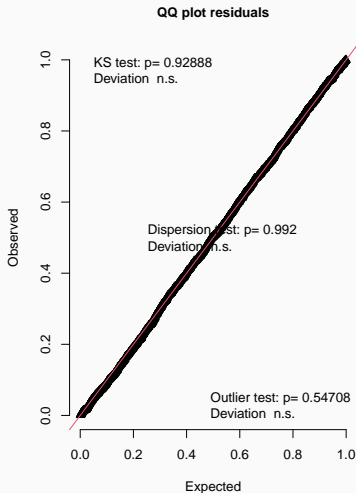
Predictors modulated: sex

Predictions are on the response-scale.



```
simulateResiduals(tit.sex, plot = TRUE)
```

DHARMA residual

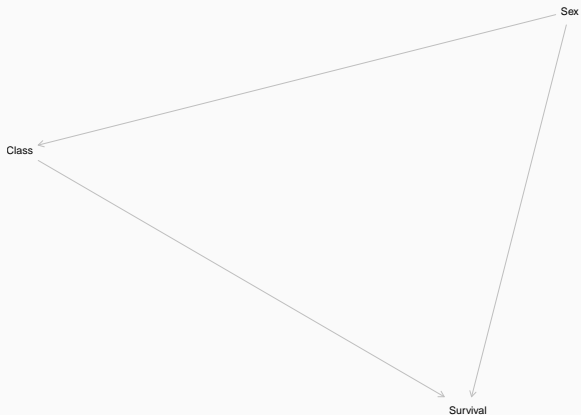


Q: Did women have higher survival because they travelled more in first class?

---

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

Sex is a confounder



## 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

<https://pollev.com/franciscorod726>

## Fit additive model with both factors

Call:

```
glm(formula = survived ~ class + sex, family = binomial, data = titanic)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )	
(Intercept)	1.18740	0.15747	7.541	4.68e-14	***
classfirst	0.88081	0.15697	5.611	2.01e-08	***
classecond	-0.07178	0.17093	-0.420	0.675	
classthird	-0.77742	0.14231	-5.463	4.69e-08	***
sexmale	-2.42133	0.13909	-17.408	< 2e-16	***

---

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

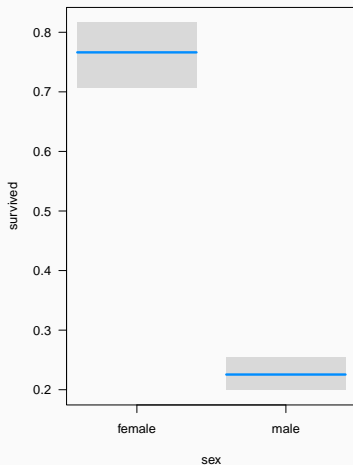
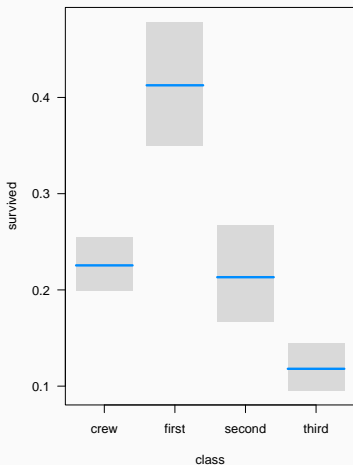
(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2769.5 on 2200 degrees of freedom  
Residual deviance: 2228.9 on 2196 degrees of freedom  
AIC: 2238.9

Number of Fisher Scoring iterations: 4

# Plot additive model

```
visreg(tit.sex.class.add, scale = 'response', rug = FALSE)
```



null device

## Fit model with the interaction of both factors

Call:

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

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )	
(Intercept)	1.89712	0.61914	3.064	0.00218	**
classfirst	1.66535	0.80026	2.081	0.03743	*
classesecond	0.07053	0.68630	0.103	0.91815	
classthird	-2.06075	0.63551	-3.243	0.00118	**
sexmale	-3.14690	0.62453	-5.039	4.68e-07	***
classfirst:sexmale	-1.05911	0.81959	-1.292	0.19627	
classesecond:sexmale	-0.63882	0.72402	-0.882	0.37760	
classthird:sexmale	1.74286	0.65139	2.676	0.00746	**

---

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: 2163.7 on 2193 degrees of freedom

AIC: 2179.7

## Women had higher survival than men, even within the same class

```
estimate_means(tit.sex.class.int)
```

Estimated Marginal Means

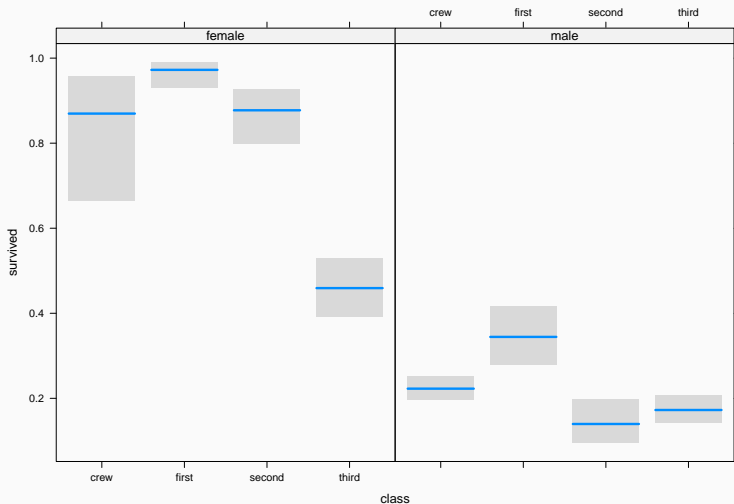
class	sex	Probability	95% CI
crew	female	0.87	[0.66, 0.96]
first	female	0.97	[0.93, 0.99]
second	female	0.88	[0.80, 0.93]
third	female	0.46	[0.39, 0.53]
crew	male	0.22	[0.20, 0.25]
first	male	0.34	[0.28, 0.42]
second	male	0.14	[0.10, 0.20]
third	male	0.17	[0.14, 0.21]

Variable predicted: survived

Predictors modulated: class, sex

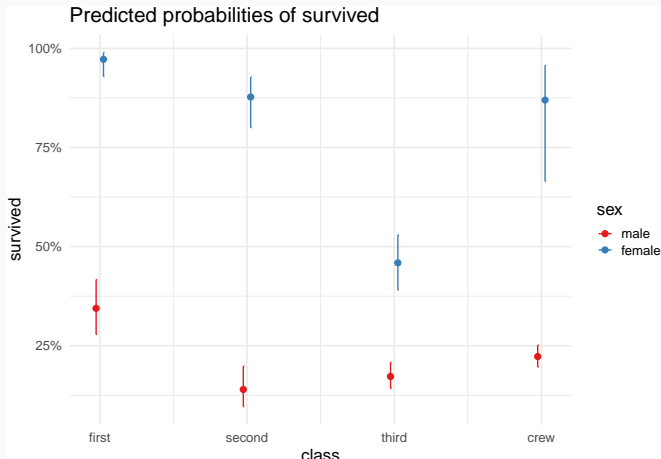
# Women had higher survival than men, even within the same class

```
visreg(tit.sex.class.int, by = 'sex', xvar = 'class', scale = 'response', rug = FALSE)
```



# Visualising model (sjPlot)

```
library('sjPlot')  
plot_model(tit.sex.class.int, type = 'int')
```



# Comparing models

```
library('easystats') # 'performance' pkg
compare_performance(tit.sex.class.add, tit.sex.class.int)
```

```
# Comparison of Model Performance Indices
```

Name	Model	AIC (weights)	AICc (weights)	BIC (weights)
tit.sex.class.add	glm	2238.9 (<.001)	2238.9 (<.001)	2267.4 (<.001)
tit.sex.class.int	glm	2179.7 (>.999)	2179.8 (>.999)	2225.3 (>.999)

Name	Tjur's R2	RMSE	Sigma	Log_loss	Score_log
tit.sex.class.add	0.248	0.405	1	0.506	-Inf
tit.sex.class.int	0.271	0.399	1	0.492	-Inf

Name	Score_spherical	PCP
tit.sex.class.add	0.004	0.671
tit.sex.class.int	0.002	0.681

## Comparing parameters

```
compare_parameters(tit.sex.class.add, tit.sex.class.int)
```

Parameter	tit.sex.class.add	tit.sex.class.int
(Intercept)	1.19 ( 0.88, 1.50)	1.90 ( 0.68, 3.11)
class [first]	0.88 ( 0.57, 1.19)	1.67 ( 0.10, 3.23)
class [second]	-0.07 (-0.41, 0.26)	0.07 (-1.27, 1.42)
class [third]	-0.78 (-1.06, -0.50)	-2.06 (-3.31, -0.82)
sex [male]	-2.42 (-2.69, -2.15)	-3.15 (-4.37, -1.92)
class [first] × sex [male]		-1.06 (-2.67, 0.55)
class [second] × sex [male]		-0.64 (-2.06, 0.78)
class [third] × sex [male]		1.74 ( 0.47, 3.02)
Observations	2201	2201

Is survival related to age?

Are age effects dependent on sex?

## Logistic regression for proportion data

---

## Why not modelling proportion?

- $1/4 = 0.25$
- $250/1000 = 0.25$
- Sample size information is lost!

## 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 = binomial)
```

Call:

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

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)

Null deviance: 671.96 on 13 degrees of freedom  
Residual deviance: 491.06 on 10 degrees of freedom  
AIC: 545.68

Number of Fisher Scoring iterations: 4

```
estimate_means(prop.glm)
```

Estimated Marginal Means

Class	Probability	95% CI
1st	0.62	[0.57, 0.68]
2nd	0.41	[0.36, 0.47]
3rd	0.25	[0.22, 0.29]
Crew	0.24	[0.21, 0.27]

Variable predicted: cbind(Yes, No)

Predictors modulated: Class

Predictors averaged: Yes (44), No (93)

Predictions are on the response-scale.

## Contrasts in survival among classes

### Marginal Contrasts Analysis

Level1	Level2	Difference	SE	95% CI	z	p
2nd	1st	-0.21	0.04	[-0.29, -0.13]	-5.31	< .001
3rd	1st	-0.37	0.03	[-0.43, -0.31]	-11.85	< .001
Crew	1st	-0.39	0.03	[-0.44, -0.33]	-12.65	< .001
3rd	2nd	-0.16	0.03	[-0.23, -0.10]	-4.84	< .001
Crew	2nd	-0.17	0.03	[-0.24, -0.11]	-5.37	< .001
Crew	3rd	-0.01	0.02	[-0.06, 0.03]	-0.58	0.563

Variable predicted: cbind(Yes, No)

Predictors contrasted: Class

Predictors averaged: Yes (44), No (93)

p-values are uncorrected.

Contrasts are on the response-scale.

## Logistic regression with continuous predictors

---

Example dataset: survival\_expt.csv

```
surv <- readr::read_csv('data/survival_expt.csv')  
head(surv)
```

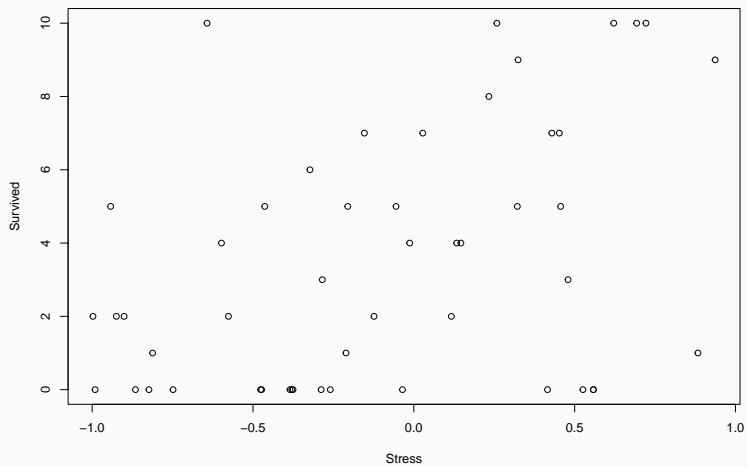
```
# A tibble: 6 x 3
```

```
  Survived Died Stress  
  <dbl> <dbl> <dbl>  
1         7     3  0.452  
2        10     0  0.622  
3         2     8 -0.997  
4        10     0  0.722  
5         0    10  0.558  
6         1     9 -0.211
```

Q: Is survival related to environmental stress?

<https://pollev.com/franciscorod726>

# Visualising data



```
surv.glm <- glm(cbind(Survived, Died) ~ Stress, data = surv, family = binomial)
```

Call:

```
glm(formula = cbind(Survived, Died) ~ Stress, family = binomial,  
    data = surv)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )	
(Intercept)	-0.54005	0.09805	-5.508	3.64e-08	***
Stress	1.23403	0.19079	6.468	9.92e-11	***

---

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

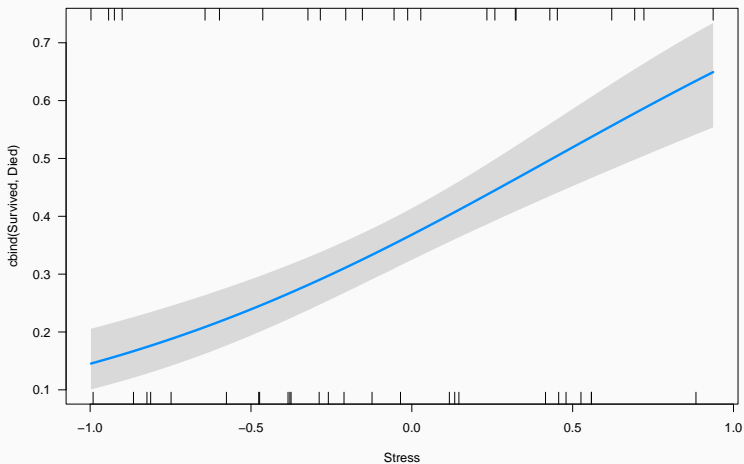
(Dispersion parameter for binomial family taken to be 1)

Null deviance: 323.54 on 49 degrees of freedom  
Residual deviance: 277.29 on 48 degrees of freedom  
AIC: 354.05

Number of Fisher Scoring iterations: 4

## Plot model using visreg:

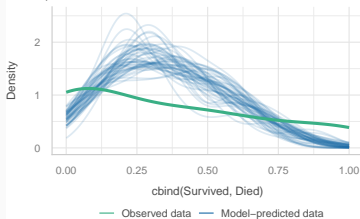
```
visreg(surv.glm, scale = 'response')
```



# Check model (easystats)

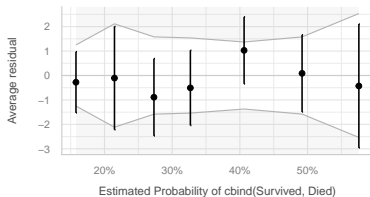
## Posterior Predictive Check

Model-predicted lines should resemble observed data line



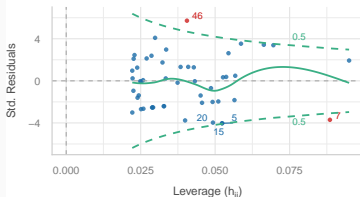
## Binned Residuals

Points should be within error bounds



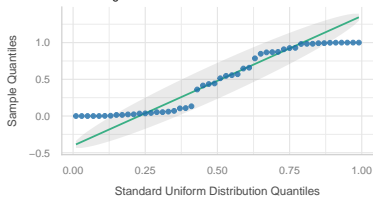
## Influential Observations

Points should be inside the contour lines



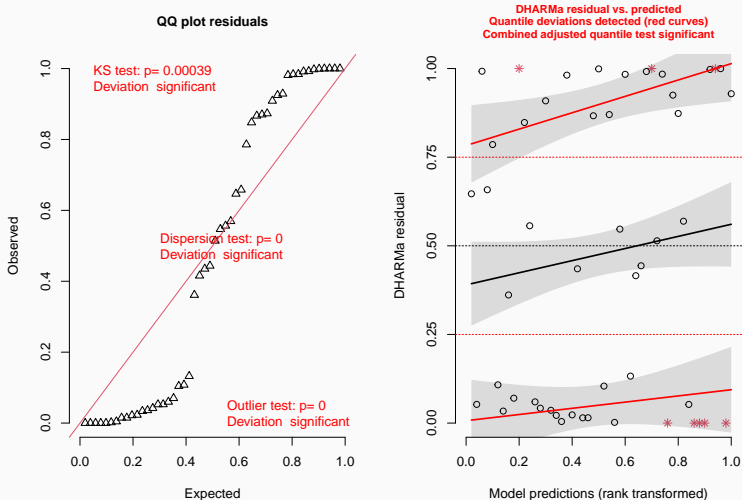
## Distribution of Quantile Residuals

Dots should fall along the line



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

DHARMA residual



# Overdispersion

---

## Overdispersion:

more variation in the data than assumed by statistical model

$$\text{Var}(y) = np(1 - p)$$

## Testing for overdispersion (DHARMA)

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

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

```
data: simres  
dispersion = 4.8201, p-value < 2.2e-16  
alternative hypothesis: two.sided
```

```
check_overdispersion(surv.glm)
```

```
# Overdispersion test
```

```
dispersion ratio = 3.134
```

```
p-value = < 0.001
```

`quasibinomial` allows us to model overdispersed binomial data

## Overdispersion in logistic regression with proportion data

```
surv.overdisp <- glm(cbind(Survived, Died) ~ Stress, data = surv, family = quasibin
```

Call:

```
glm(formula = cbind(Survived, Died) ~ Stress, family = quasibinomial,  
     data = surv)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-0.5400	0.2137	-2.527	0.01484 *
Stress	1.2340	0.4158	2.968	0.00466 **

---

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

(Dispersion parameter for quasibinomial family taken to be 4.74884)

Null deviance: 323.54 on 49 degrees of freedom  
Residual deviance: 277.29 on 48 degrees of freedom  
AIC: NA

Number of Fisher Scoring iterations: 4

# Mean estimates do not change after accounting for overdispersion

But standard errors (uncertainty) do!

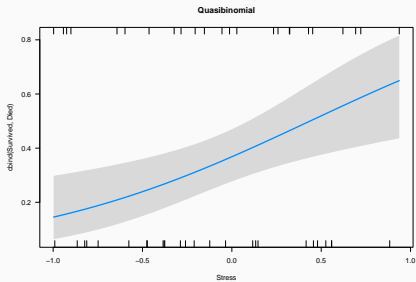
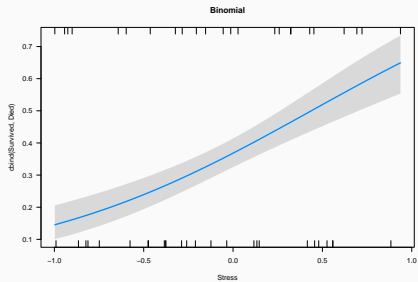
```
parameters(surv.overdisp)
```

Parameter	Log-Odds	SE	95% CI	t(48)	p
(Intercept)	-0.54	0.21	[-0.97, -0.13]	-2.53	0.011
Stress	1.23	0.42	[ 0.45, 2.09]	2.97	0.003

```
parameters(surv.glm)
```

Parameter	Log-Odds	SE	95% CI	z	p
(Intercept)	-0.54	0.10	[-0.73, -0.35]	-5.51	< .001
Stress	1.23	0.19	[ 0.87, 1.62]	6.47	< .001

# But standard errors (uncertainty) do!



- `moth.csv`: Probability of moth predation on trunk trees depending on morph (light/dark) and distance to Liverpool ([Bishop 1972](#))
- `seedset.csv`: Comparing seed set among plants (Data from [Harder et al. 2011](#))

## Moth predation

---

## The industrial revolution and evolution of dark morphs



```
moth <- read.csv('data/moth.csv')
```

	MORPH	DISTANCE	PLACED	REMOVED
1	light	0.0	56	17
2	dark	0.0	56	14
3	light	7.2	80	28
4	dark	7.2	80	20
5	light	24.1	52	18
6	dark	24.1	52	22

## Creating new variable: REMAIN

```
moth$REMAIN <- moth$PLACED - moth$REMOVED
```

	MORPH	DISTANCE	PLACED	REMOVED	REMAIN
1	light	0.0	56	17	39
2	dark	0.0	56	14	42
3	light	7.2	80	28	52
4	dark	7.2	80	20	60
5	light	24.1	52	18	34
6	dark	24.1	52	22	30

## Did some morph have higher predation overall?

Call:

```
glm(formula = cbind(REMOVED, REMAIN) ~ MORPH, family = binomial,  
     data = moth)
```

Coefficients:

```
              Estimate Std. Error z value Pr(>|z|)  
(Intercept) -0.57752    0.09473  -6.097 1.08e-09 ***  
MORPHlight  -0.40331    0.13925  -2.896 0.00377 **
```

---

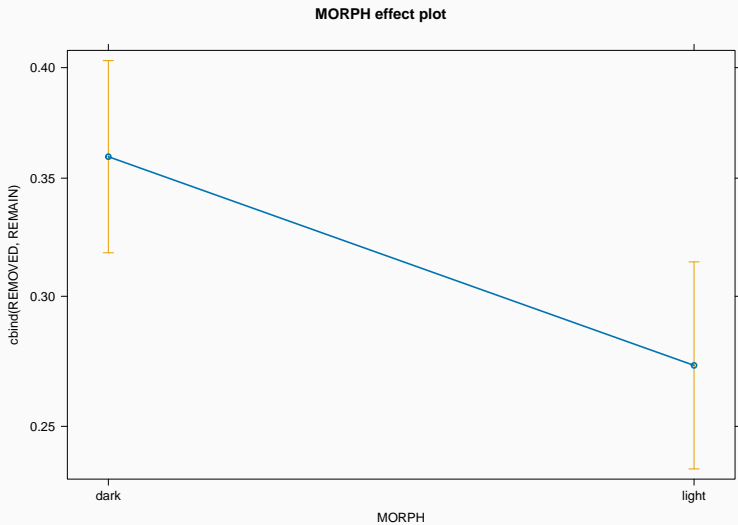
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

```
Null deviance: 35.385  on 13  degrees of freedom  
Residual deviance: 26.936  on 12  degrees of freedom  
AIC: 93.61
```

Number of Fisher Scoring iterations: 4

# Did some morph have higher predation overall?



## Did predation increase farther from city centre?

Call:

```
glm(formula = cbind(REMOVED, REMAIN) ~ DISTANCE, family = binomial,  
     data = moth)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-0.925861	0.136634	-6.776	1.23e-11 ***
DISTANCE	0.005268	0.003984	1.322	0.186

---

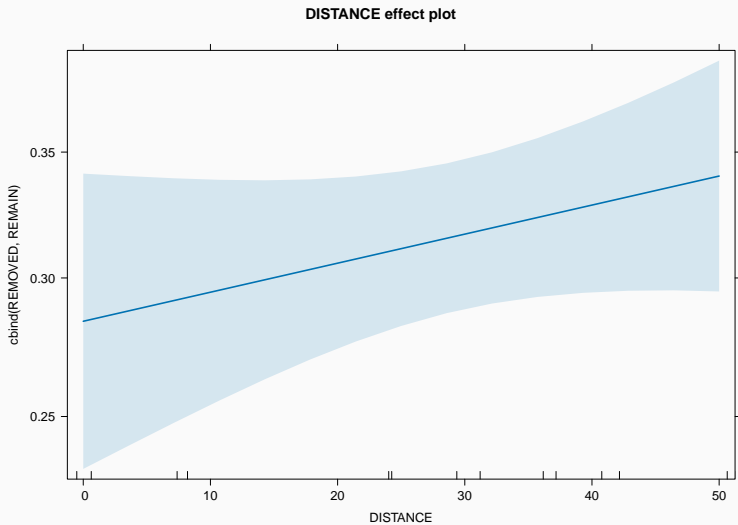
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 35.385 on 13 degrees of freedom  
Residual deviance: 33.626 on 12 degrees of freedom  
AIC: 100.3

Number of Fisher Scoring iterations: 4

# Did predation increase farther from city centre?



# Did dark morph have lower predation in city & light have lower predation in countryside?

Call:

```
glm(formula = cbind(REMOVED, REMAIN) ~ MORPH * DISTANCE, family = binomial,  
     data = moth)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )	
(Intercept)	-1.128987	0.197906	-5.705	1.17e-08	***
MORPHlight	0.411257	0.274490	1.498	0.134066	
DISTANCE	0.018502	0.005645	3.277	0.001048	**
MORPHlight:DISTANCE	-0.027789	0.008085	-3.437	0.000588	***

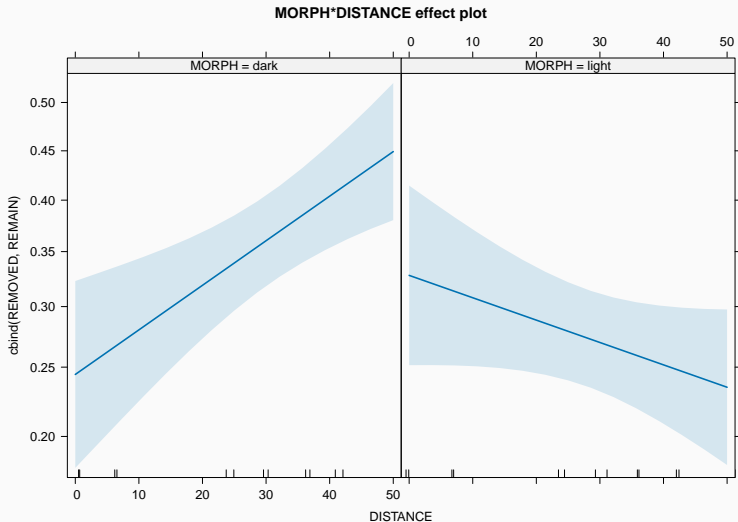
---

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

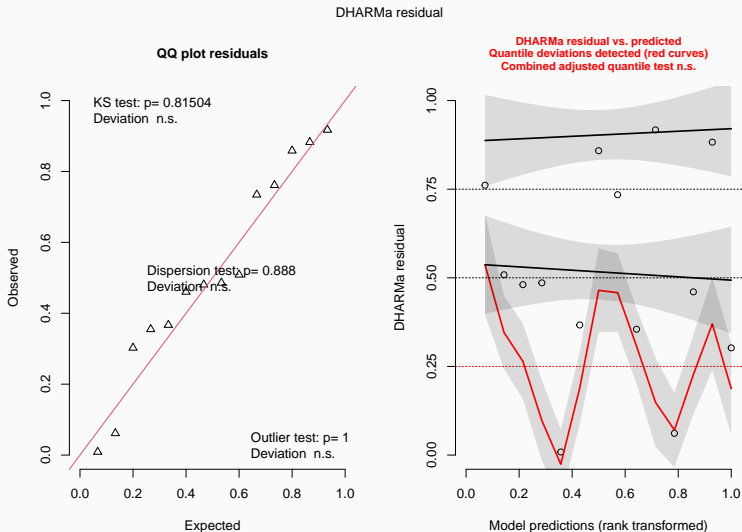
(Dispersion parameter for binomial family taken to be 1)

Null deviance: 35.385 on 13 degrees of freedom  
Residual deviance: 13.230 on 10 degrees of freedom  
AIC: 83.904

Did dark morph have lower predation in city & light have lower predation in countryside?



```
simulateResiduals(pred.int, plot = TRUE)
```



## Seed set among plants

---

# Seed set among plants



## Seed set among plants

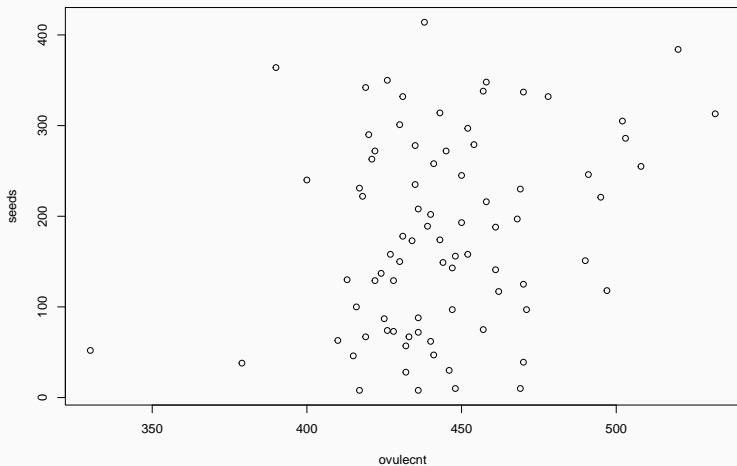
```
# A tibble: 6 x 6
```

```
  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
```

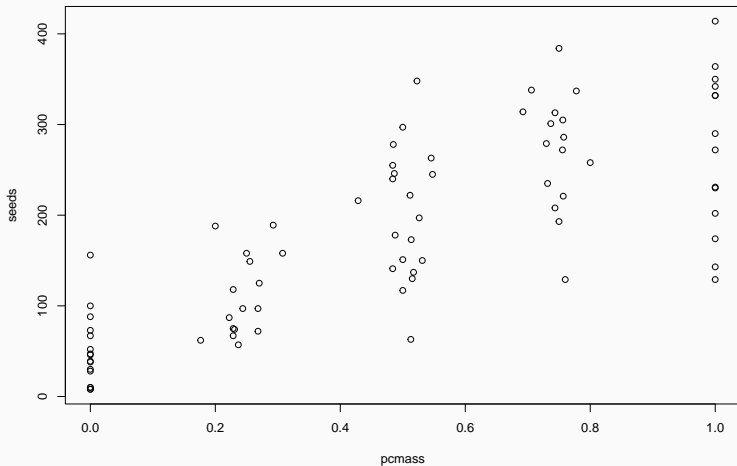
<https://pollev.com/franciscorod726>

- Is seed set related to proportion of outcross pollen ( $pc_{mass}$ )?
- Which plant had lower seed set?

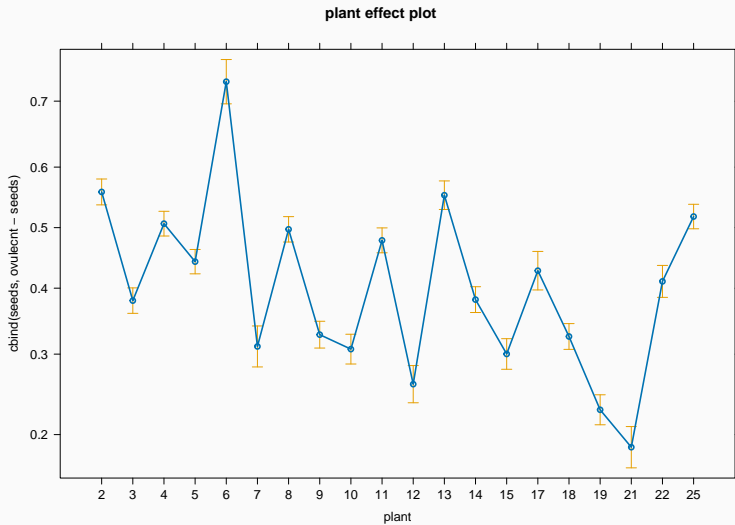
# Number of seeds vs Number of ovules



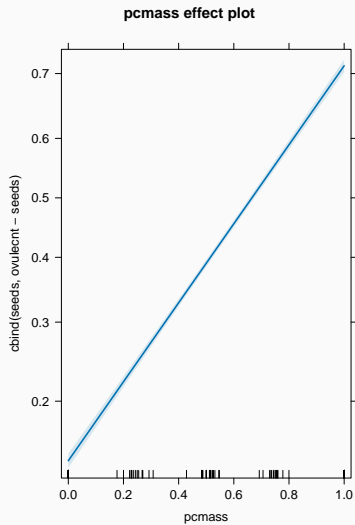
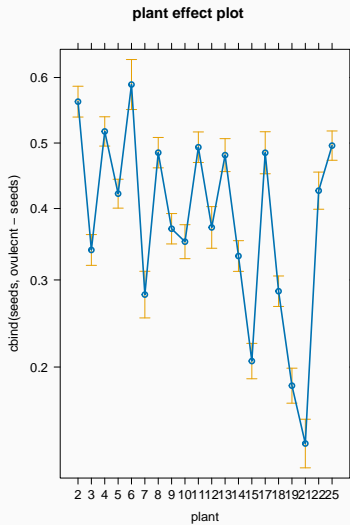
# Number of seeds vs Proportion outcross pollen



# Seed set across plants



# Seed set ~ outcross pollen



## GLM for count data: Poisson regression

---

- **Gaussian:** `lm`
- **Binary:** `glm (family binomial / quasibinomial)`
- **Counts:** `glm (family poisson / quasipoisson)`

- 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

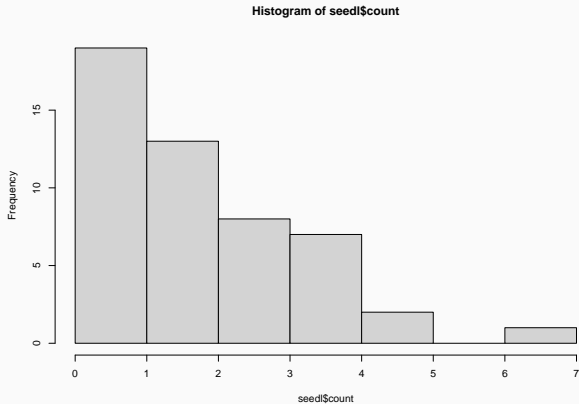
```
seedl <- read.csv('data/seedlings.csv')
```

sample	count	light	area
Min. : 1.00	Min. :0.00	Min. : 2.571	Min. :0.25
1st Qu.:13.25	1st Qu.:1.00	1st Qu.:26.879	1st Qu.:0.25
Median :25.50	Median :2.00	Median :47.493	Median :0.50
Mean :25.50	Mean :2.14	Mean :47.959	Mean :0.62
3rd Qu.:37.75	3rd Qu.:3.00	3rd Qu.:67.522	3rd Qu.:1.00
Max. :50.00	Max. :7.00	Max. :99.135	Max. :1.00

# Exploring the data

```
table(seed$count)
```

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





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

which corresponds to

```
equatiomatic::extract_eq(seedl.glm)
```

$$\log(E(\text{count})) = \alpha + \beta_1(\text{light}) \quad (5)$$

# Interpreting Poisson GLM

Call:

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

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

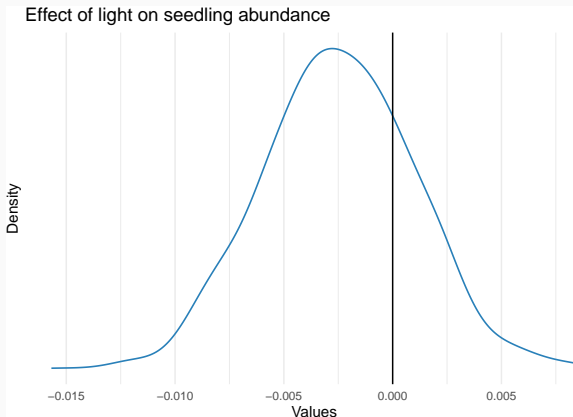
(Dispersion parameter for poisson family taken to be 1)

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

# Estimated distribution of the slope parameter

```
library('parameters')  
plot(simulate_parameters(seedl.glm)) +  
  geom_vline(xintercept = 0) +  
  ggtitle('Effect of light on seedling abundance')
```



## Parameter estimates are in log scale!

Parameter estimates (log scale):

```
coef(seedl.glm)[1]
```

(Intercept)

0.881805

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

```
exp(coef(seedl.glm)[1])
```

(Intercept)

2.415255

```
library('easystats')  
parameters(seedl.glm)
```

Parameter	Log-Mean	SE	95% CI	z	p
(Intercept)	0.88	0.19	[ 0.50, 1.24]	4.67	< .001
light	-2.58e-03	3.53e-03	[-0.01, 0.00]	-0.73	0.465

```
parameters(seedl.glm, exponentiate = TRUE)
```

Parameter	IRR	SE	95% CI	z	p
(Intercept)	2.42	0.46	[1.65, 3.46]	4.67	< .001
light	1.00	3.52e-03	[0.99, 1.00]	-0.73	0.465

## How Nseedlings decrease with light

### Model-based Predictions

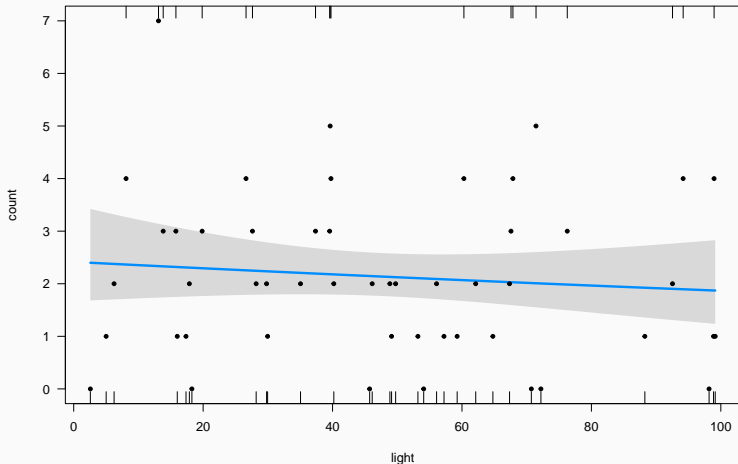
light	Predicted	SE	95% CI
2.57	2.40	0.43	[1.68, 3.42]
13.30	2.33	0.35	[1.74, 3.13]
24.03	2.27	0.28	[1.78, 2.89]
34.76	2.21	0.23	[1.80, 2.71]
45.49	2.15	0.21	[1.78, 2.60]
56.22	2.09	0.22	[1.71, 2.56]
66.95	2.03	0.25	[1.60, 2.58]
77.68	1.98	0.29	[1.48, 2.64]
88.41	1.92	0.34	[1.36, 2.73]
99.14	1.87	0.39	[1.24, 2.83]

Variable predicted: count

Predictors modulated: light

## Visualising how Nseedlings decrease with light

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



```
library('performance')  
r2(seedl.glm)
```

```
# R2 for Generalized Linear Regression  
Nagelkerke's R2: 0.015
```

```
library('report')  
report(seedl.glm)
```

We fitted a poisson model (estimated using ML) to predict count with light (formula: count ~ light). The model's explanatory power is very weak (Nagelkerke's  $R^2 = 0.01$ ). The model's intercept, corresponding to light = 0, is at 0.88 (95% CI [0.50, 1.24],  $p < .001$ ). Within this model:

- The effect of light is statistically non-significant and negative (beta =  $-2.58e-03$ , 95% CI [ $-9.57e-03$ ,  $4.28e-03$ ],  $p = 0.465$ ; Std. beta =  $-0.07$ , 95% CI [ $-0.27$ ,  $0.12$ ])

Standardized parameters were obtained by fitting the model on a standardized version of the dataset. 95% Confidence Intervals (CIs) and p-values were computed using a Wald z-distribution approximation.

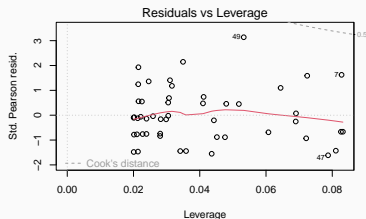
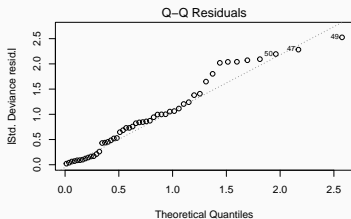
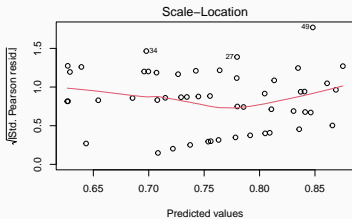
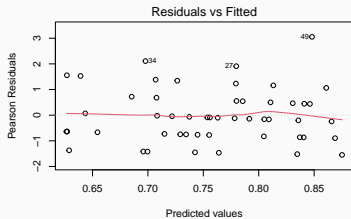
## Model checking

---

- Linearity (log response  $\sim$  predictors)
- Observations are independent
- Mean = Variance

# Checking Poisson GLM

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



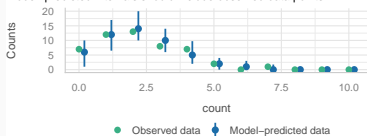
null device

# Checking Poisson GLM

```
check_model(seedl.glm)
```

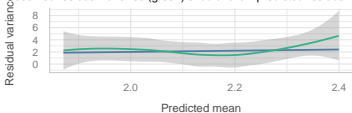
## Posterior Predictive Check

Model-predicted intervals should include observed data points



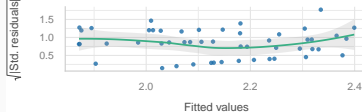
## Misspecified dispersion and zero-inflation

Observed residual variance (green) should follow predicted residual variance



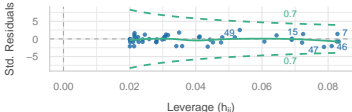
## Homogeneity of Variance

Reference line should be flat and horizontal



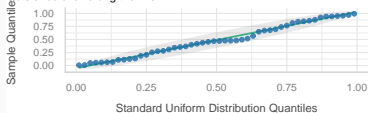
## Influential Observations

Points should be inside the contour lines



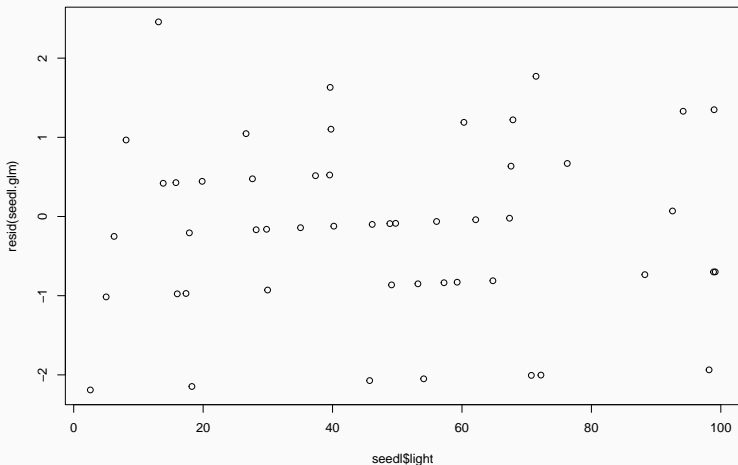
## Distribution of Quantile Residuals

Dots should fall along the line



## Is there pattern of residuals along predictor?

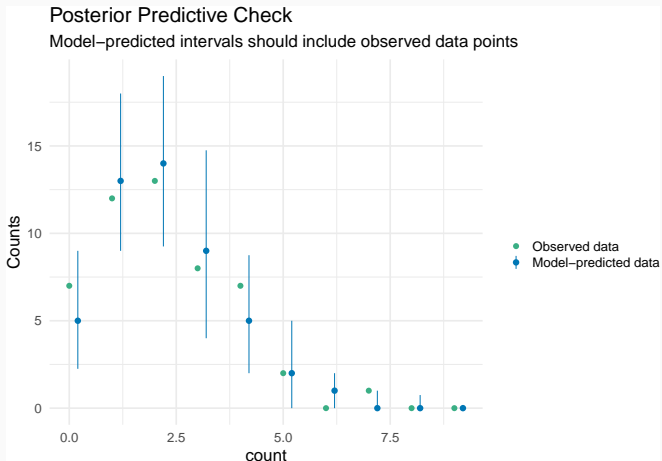
```
plot(seedl$light, resid(seedl.glm))
```



# Posterior predictive checking

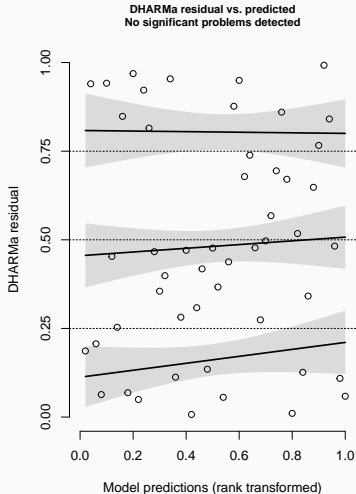
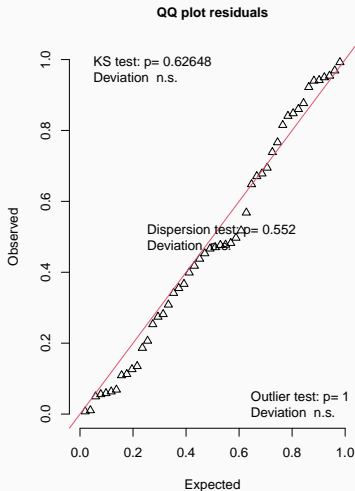
Simulate data from fitted model (`yrep`) and compare with observed data (`y`)

```
check_predictions(seed1.glm)
```



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

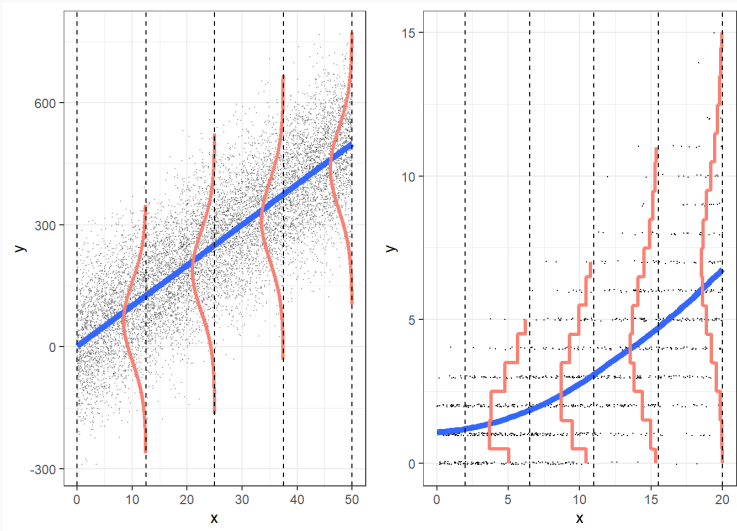
DHARMA residual



## Overdispersion

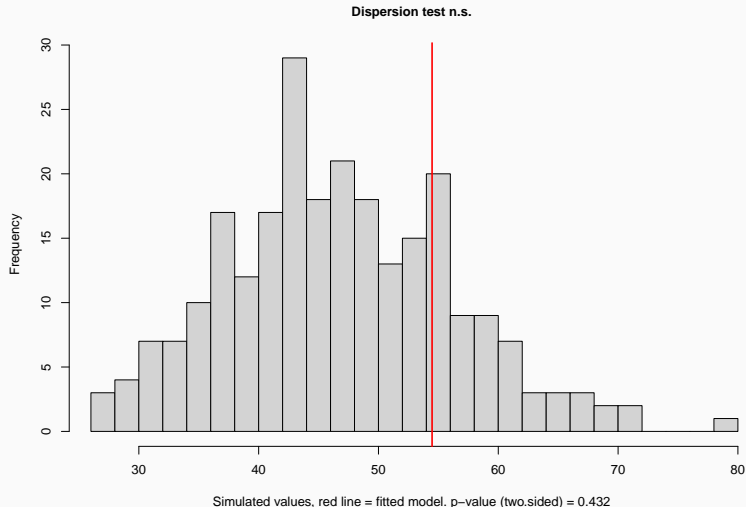
---

# Poisson GLM assumes mean = variance



## Always check overdispersion with count data

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



- Use family `quasipoisson`
- Use negative binomial distribution (`MASS::glm.nb`)
- Include observation-level random effect (e.g. see [Harrison 2014](#))

## Accounting for overdispersion with family quasipoisson

Call:

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

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 1.134907)

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

Number of Fisher Scoring iterations: 5

## Mean estimates do not change after accounting for overdispersion

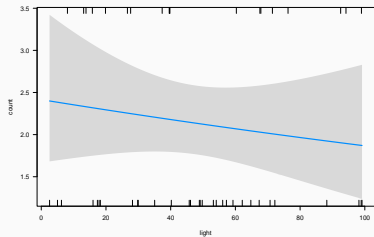
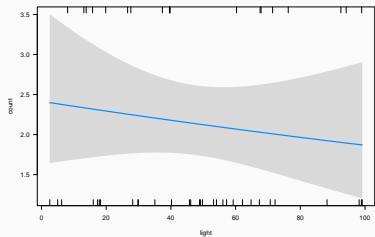
```
parameters(seedl.overdisp)
```

Parameter	Log-Mean	SE	95% CI	t(48)	p
(Intercept)	0.88	0.20	[ 0.47, 1.26]	4.38	< .001
light	-2.58e-03	3.76e-03	[-0.01, 0.00]	-0.69	0.493

```
parameters(seedl.glm)
```

Parameter	Log-Mean	SE	95% CI	z	p
(Intercept)	0.88	0.19	[ 0.50, 1.24]	4.67	< .001
light	-2.58e-03	3.53e-03	[-0.01, 0.00]	-0.73	0.465

## But standard errors may change



## Accounting for overdispersion using negative binomial

```
library('MASS')  
seedl.nb <- glm.nb(count ~ light, data = seedl)
```

Call:

```
glm.nb(formula = count ~ light, data = seedl, init.theta = 22.23419419,  
       link = log)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	0.881996	0.198213	4.450	8.6e-06 ***
light	-0.002580	0.003691	-0.699	0.485

---

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

(Dispersion parameter for Negative Binomial(22.2342) family taken to be 1)

Null deviance: 58.247 on 49 degrees of freedom  
Residual deviance: 57.756 on 48 degrees of freedom  
AIC: 183.83

Number of Fisher Scoring iterations: 1

# Comparing Poisson and Negative Binomial

```
compare_models(seedl.glm, seedl.nb)
```

Parameter	seedl.glm	seedl.nb
(Intercept)	0.88 ( 0.51, 1.25)	0.88 ( 0.49, 1.27)
light	-2.58e-03 (-0.01, 0.00)	-2.58e-03 (-0.01, 0.00)
Observations	50	50

```
compare_performance(seedl.glm, seedl.nb)
```

```
# Comparison of Model Performance Indices
```

Name	Model	AIC (weights)	AICc (weights)	BIC (weights)
seedl.glm	glm	182.0 (0.710)	182.3 (0.737)	185.9 (0.864)
seedl.nb	negbin	183.8 (0.290)	184.3 (0.263)	189.6 (0.136)

Name	Nagelkerke's R2	RMSE	Sigma	Score_log	Score_spherical
seedl.glm	0.015	1.529	1	-1.780	0.131
seedl.nb	0.014	1.529	1	-1.821	0.133

What if survey plots have  
different area?

---

## Shall we *standardise* counts dividing by sampling plot area?

Model would be: count/area ~ light

	sample	count	light	area
1	1	0	70.71854	0.50
2	2	1	88.26021	0.25
3	3	2	67.35133	0.50
4	4	3	67.57850	1.00
5	5	4	26.63098	0.25
6	6	3	15.79433	1.00

*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†

<https://doi.org/10.2307/2983064>

## Use offset to account for variable sampling effort

```
seedl.offset <- glm(count ~ light,  
                    offset = log(area),  
                    data = seedl,  
                    family = poisson)
```

## Note estimates now referred to area units!

Call:

```
glm(formula = count ~ light, family = poisson, data = seedl,  
     offset = log(area))
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	1.513185	0.183245	8.258	<2e-16 ***
light	-0.005674	0.003384	-1.677	0.0936 .

---

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

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 95.199 on 49 degrees of freedom  
Residual deviance: 92.354 on 48 degrees of freedom  
AIC: 211.9

```
exp(coef(seedl.offset)[1])
```

(Intercept)

4.541173

## Prediction

---

## Predicting number of seedlings given light

```
new.lights <- data.frame(light = c(10, 90))  
predict(seedl.glm, newdata = new.lights, type = 'response', se.fit
```

```
$fit
```

```
      1      2  
2.353841 1.915533
```

```
$se.fit
```

```
      1      2  
0.3756992 0.3502446
```

```
$residual.scale
```

```
[1] 1
```

## Prediction (easystats)

```
new.lights <- data.frame(light = c(10, 90))  
estimate_expectation(seedl.glm, data = new.lights)
```

### Model-based Predictions

light	Predicted	SE	95% CI
10	2.35	0.38	[1.72, 3.22]
90	1.92	0.35	[1.34, 2.74]

Variable predicted: count

Predictions are on the response-scale.

```
estimate_prediction(seedl.glm, data = new.lights)
```

### Model-based Predictions

light	Predicted	95% CI
10	2.35	[0.00, 6.00]
90	1.92	[0.00, 5.00]

- Infant mortality ~ GDP
- Number of cones consumed by squirrels ([data](#))
- Elephant matings ([Poole 1989](#))

## Mixed / Multilevel models

---

## 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

$$y_i \sim N(\mu_i, \sigma^2)$$

$$\mu_i = \alpha + \beta x_i$$

In this case:

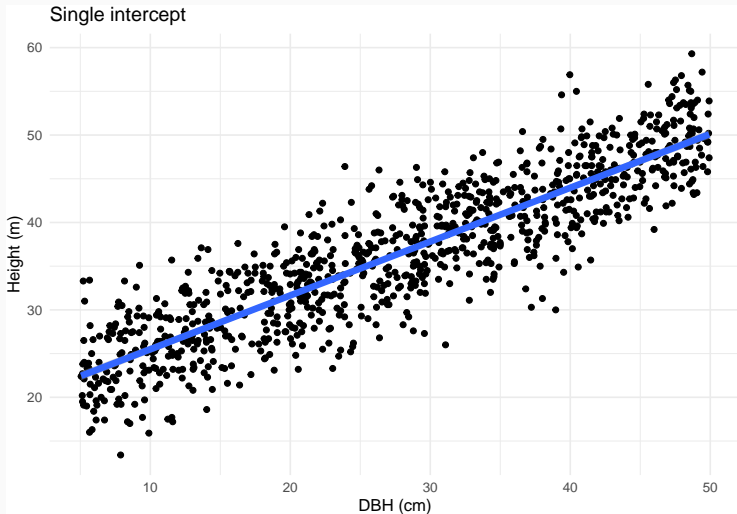
$$\text{Height}_i \sim N(\mu_i, \sigma^2)$$

$$\mu_i = \alpha + \beta \text{DBH}_i$$

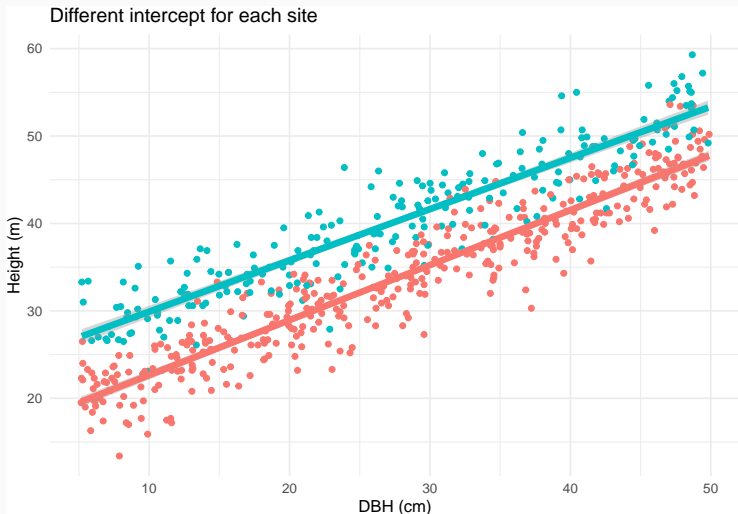
$\alpha$ : expected height when DBH = 0

$\beta$ : how much height increases with every unit increase of DBH

# There is only one intercept



## What if allometry varies among sites?



## Fitting a varying intercepts model with `lm`

Call:

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

Residuals:

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

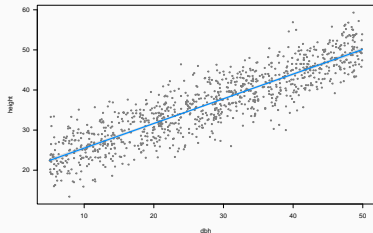
Coefficients:

	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	***

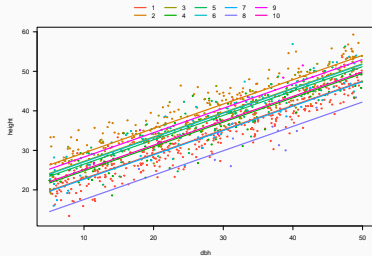
---

# Single vs varying intercept

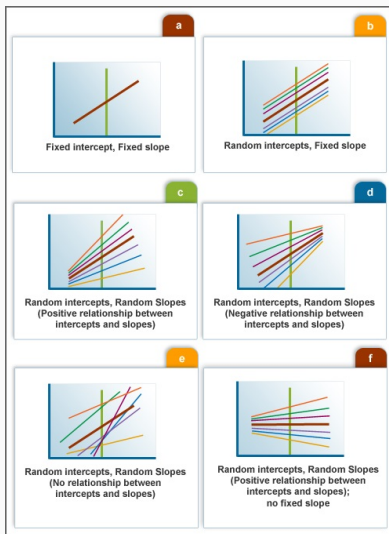
Single intercept



Different intercept for each site



# Mixed models enable us to account for variability



[www.esourceresearch.org/](http://www.esourceresearch.org/)

$$y_i = a + \alpha_j + b \cdot x_i + \varepsilon_i$$

$$\alpha_j \sim N(0, \tau^2)$$

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

In our example:

$$\text{Height}_i = a + \text{site}_j + b \cdot \text{DBH}_i + \varepsilon_i$$

$$\text{site}_j \sim N(0, \tau^2)$$

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

Mixed models estimate **varying parameters**

(intercepts and/or slopes)

**with pooling among levels**

(rather than considering them fully independent)

- **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))`

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/](http://andrewgelman.com/2005/01/25/why_i_dont_use/)

## What is a random effect, really?

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

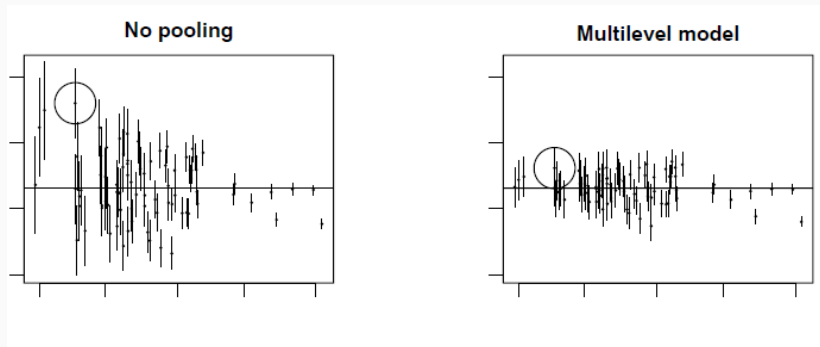
## What is a random effect, really?

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

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

# Shrinkage improves parameter estimation

Especially for groups with low sample size



*From Gelman & Hill p. 253*

# Fitting mixed/multilevel models

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

```
Family: gaussian ( identity )  
Formula:      height ~ dbh + (1 | site)  
Data: trees
```

	AIC	BIC	logLik	-2*log(L)	df.resid
	5110.3	5129.9	-2551.1	5102.3	996

Random effects:

Conditional model:

Groups	Name	Variance	Std.Dev.
site	(Intercept)	10.007	3.163
Residual		9.252	3.042

Number of obs: 1000, groups: site, 10

Dispersion estimate for gaussian family ( $\sigma^2$ ): 9.25

Conditional model:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	19.014671	1.045018	18.20	<2e-16 ***
dbh	0.616911	0.007569	81.51	<2e-16 ***

---

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

# 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 -0.197
```

```
library(equatiomatic)
equatiomatic::extract_eq(mixed)
```

$$\begin{aligned} \text{height}_i &\sim N(\alpha_{j[i]} + \beta_1(\text{dbh}), \sigma^2) \\ \alpha_j &\sim N(\mu_{\alpha_j}, \sigma_{\alpha_j}^2), \text{ for site } j = 1, \dots, J \end{aligned} \tag{6}$$

```
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"
```

## Compare site effects between mixed and lm

	lm	mixed
site1	16.7	16.7
site2	23.2	23.2
site3	21.1	21.0
site4	18.6	18.6
site5	20.3	20.3
site6	20.9	20.9
site7	16.5	16.6
site8	11.4	11.9
site9	22.1	21.8
site10	19.0	19.0

## 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 [broom.mixed](#)

## Results table

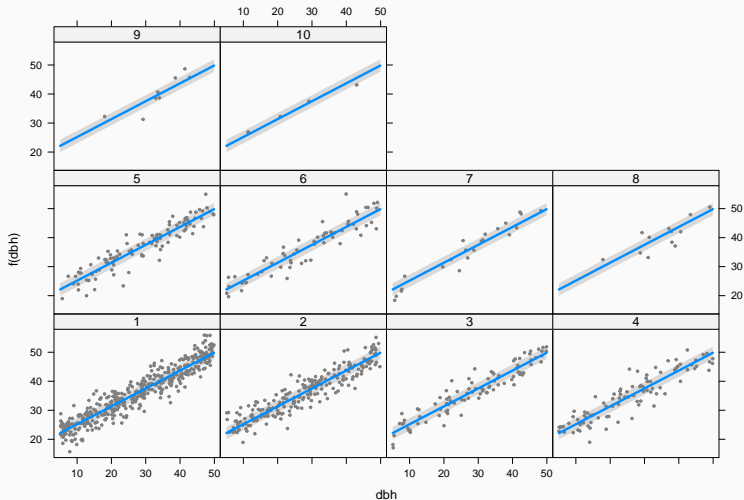
```
library(modelsummary)
```

```
modelsummary(mixed)
```

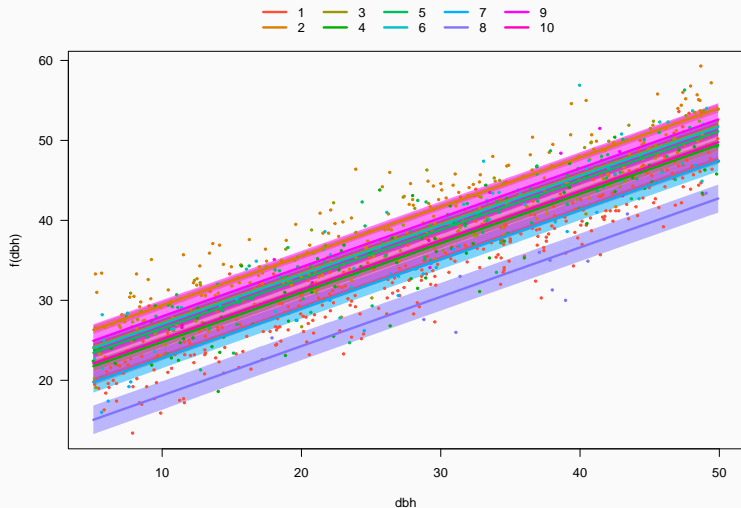
	(1)
(Intercept)	19.011
	(1.100)
dbh	0.617
	(0.008)
SD (Intercept site)	3.346
SD (Observations)	3.043
Num.Obs.	1000
R2 Marg.	0.753
R2 Cond.	0.888
AIC	5116.3
BIC	5135.9
ICC	0.5
RMSE	3.03

# Visualising model: visreg

```
visreg(mixed, xvar = 'dbh', by = 'site')
```

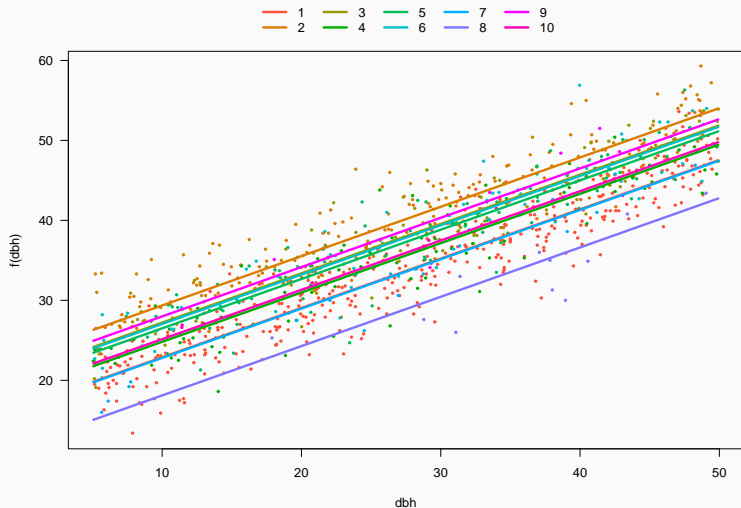


```
visreg(mixed, xvar = 'dbh', by = 'site', overlay = TRUE)
```

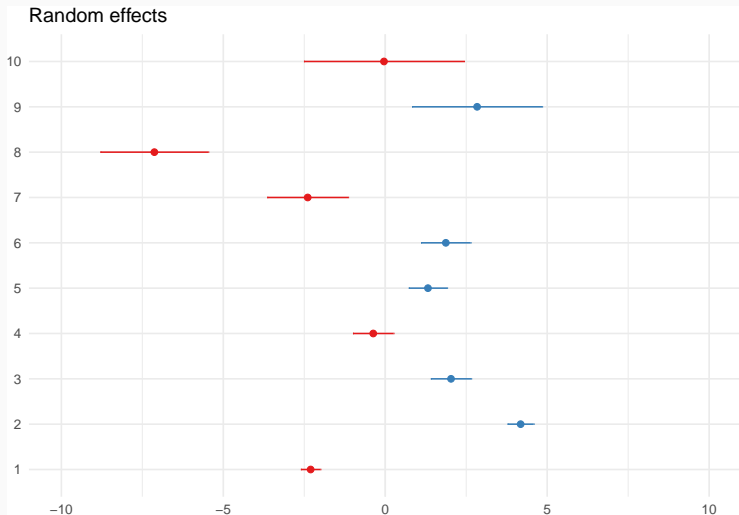


# Visualising model

```
visreg(mixed, xvar = 'dbh', by = 'site', overlay = TRUE, band = FALSE)
```

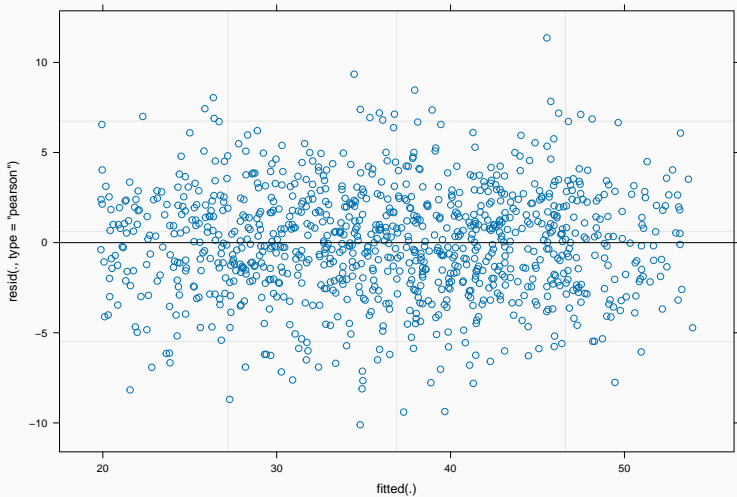


```
sjPlot::plot_model(mixed, type = 're')
```



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

```
plot(mixed)
```

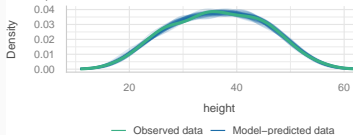


# Checking residuals

```
library('performance')  
check_model(mixed)
```

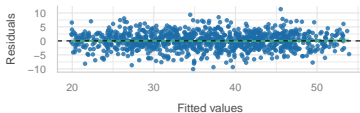
## Posterior Predictive Check

Model-predicted lines should resemble observed data line



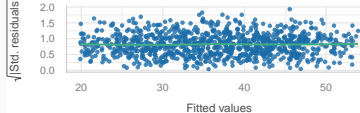
## Linearity

Reference line should be flat and horizontal



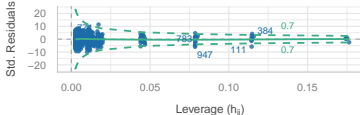
## Homogeneity of Variance

Reference line should be flat and horizontal



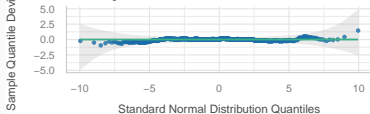
## Influential Observations

Points should be inside the contour lines



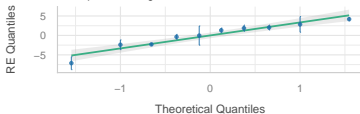
## Normality of Residuals

Dots should fall along the line



## Normality of Random Effects (site)

Dots should be plotted along the line

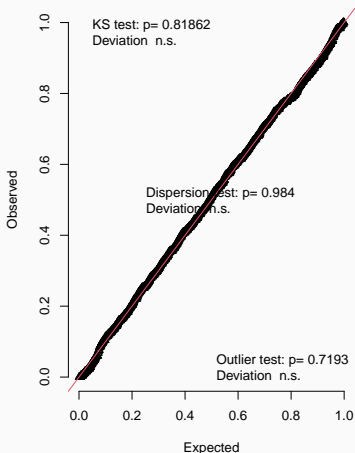


# Checking residuals (DHARMA)

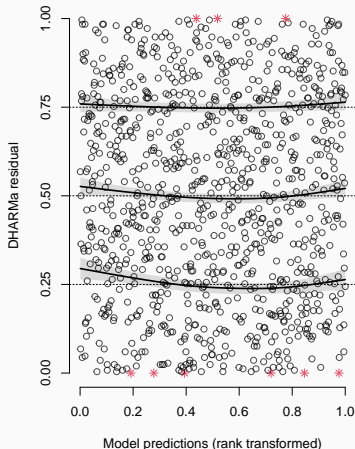
```
DHARMA::simulateResiduals(mixed, plot = TRUE, re.form = NULL)
```

DHARMA residual

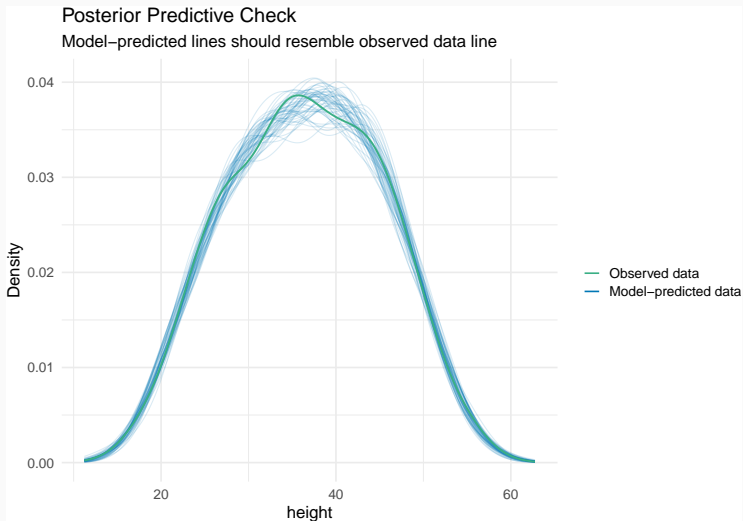
QQ plot residuals



DHARMA residual vs. predicted  
No significant problems detected



```
check_predictions(mixed)
```



Many approaches! Somewhat polemic (e.g. see [this](#)).

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

```
r2(mixed)
```

```
# R2 for Mixed Models
```

```
Conditional R2: 0.888
```

```
  Marginal R2: 0.753
```

## Growing the hierarchy: adding site-level predictors

---

We had:

$$y_i = a + \alpha_j + b \cdot x_i + \varepsilon_i$$

$$\alpha_j \sim N(0, \tau^2)$$

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

We had:

$$y_i = a + \alpha_j + b \cdot x_i + \varepsilon_i$$

$$\alpha_j \sim N(0, \tau^2)$$

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

Now

$$y_i = a + \alpha_j + b \cdot x_i + \varepsilon_i$$

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

$$\mu_j = \delta \cdot \text{Predictor}_j$$

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

## Are height differences among sites related to temperature?

$$\text{Height}_i = \text{site}_j + b \cdot \text{DBH}_i + \varepsilon_i$$

$$\text{site}_j \sim N(\mu_j, \tau^2)$$

$$\mu_j = a + \delta \cdot \text{Temperature}_j$$

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

## 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

```
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:

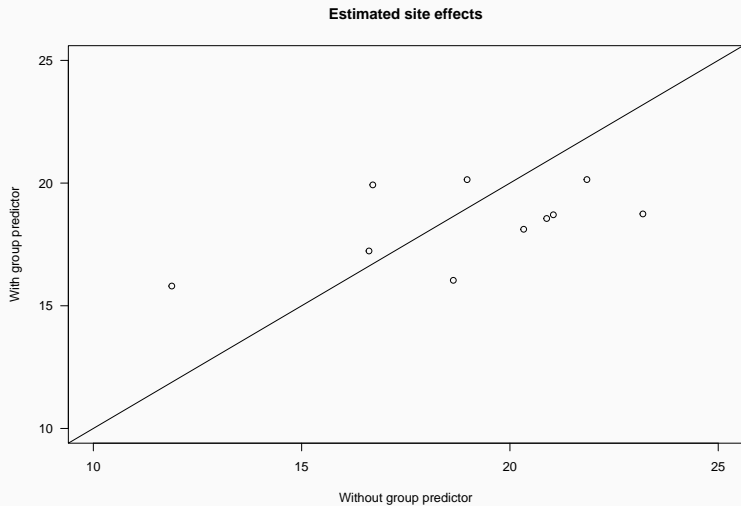
	Estimate	Std. Error	t value
(Intercept)	18.340954	0.655054	27.999
dbh	0.616894	0.007571	81.484
temp.c	1.115104	0.248000	4.496

Correlation of Fixed Effects:

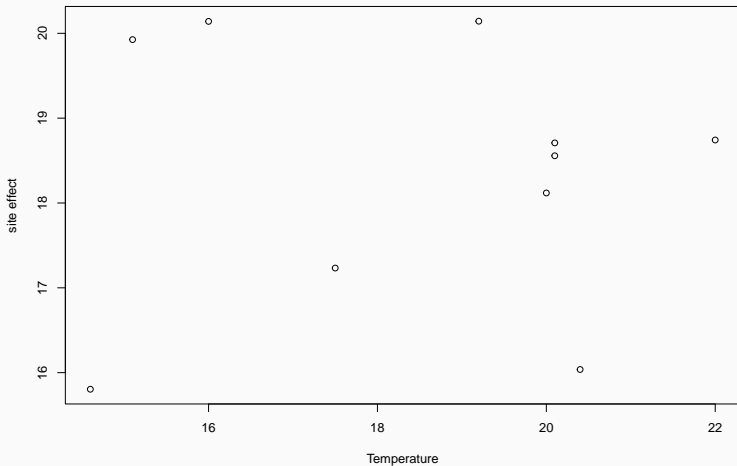
(Intr) dbh

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

## Comparing site effects with and without group predictor



# Are site effects related to temperature?



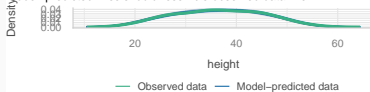
```
modelsummary(group.pred)
```

	(1)
(Intercept)	18.341 (0.655)
dbh	0.617 (0.008)
temp.c	1.115 (0.248)
SD (Intercept site)	1.777
SD (Observations)	3.044
Num.Obs.	1000
R2 Marg.	0.853
R2 Cond.	0.890
AIC	5108.2
BIC	5132.7

## check\_model(group.pred)

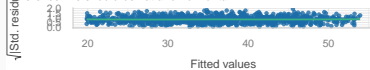
### Posterior Predictive Check

Model-predicted lines should resemble observed data line



### Homogeneity of Variance

Reference line should be flat and horizontal



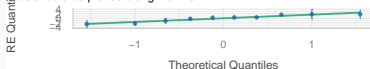
### Collinearity

High collinearity (VIF) may inflate parameter uncertainty



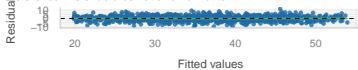
### Normality of Random Effects (site)

Dots should be plotted along the line



### Linearity

Reference line should be flat and horizontal



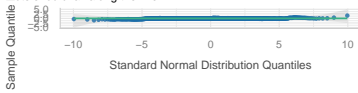
### Influential Observations

Points should be inside the contour lines



### Normality of Residuals

Dots should fall along the line

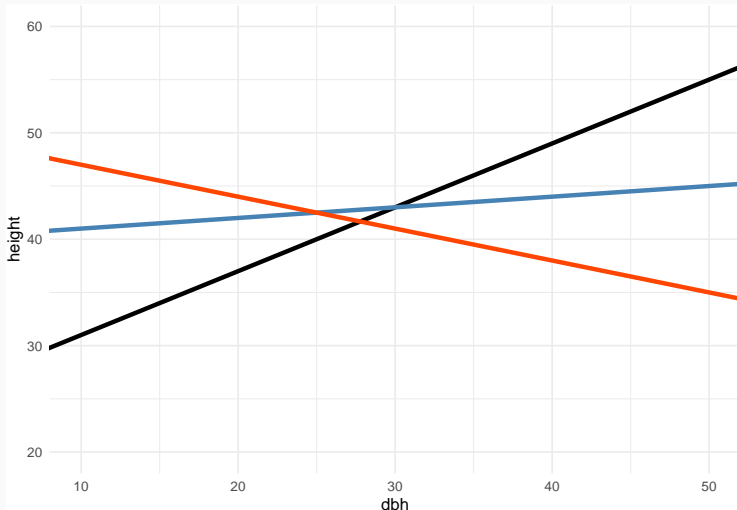


## Varying intercepts and slopes

---

# Varying intercepts and slopes

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



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

$$\begin{aligned} \text{height}_i &\sim N(\alpha_{j[i]} + \beta_{1j[i]}(\text{dbh}), \sigma^2) \\ \begin{pmatrix} \alpha_j \\ \beta_{1j} \end{pmatrix} &\sim N\left(\begin{pmatrix} \mu_{\alpha_j} \\ \mu_{\beta_{1j}} \end{pmatrix}, \begin{pmatrix} \sigma_{\alpha_j}^2 & \rho_{\alpha_j\beta_{1j}} \\ \rho_{\beta_{1j}\alpha_j} & \sigma_{\beta_{1j}}^2 \end{pmatrix}\right), \text{ for site } j = 1, \dots, J \end{aligned} \quad (7)$$

# 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 -0.722

optimizer (nloptwrap) convergence code: 0 (OK)

## 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"
```

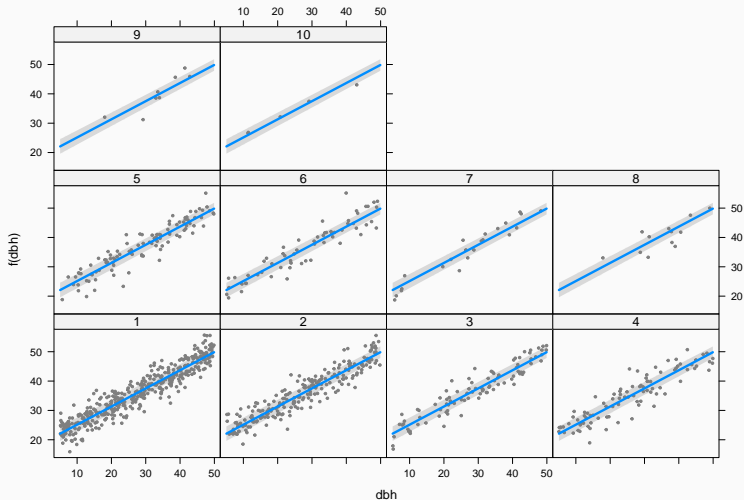
# Results table

```
modelsummary(mixed.slopes)
```

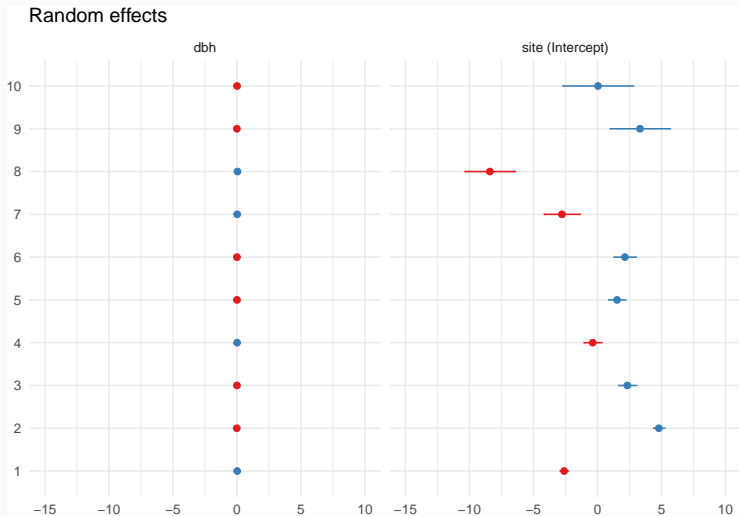
	(1)
(Intercept)	18.953
	(1.292)
dbh	0.618
	(0.009)
SD (Intercept site)	3.957
SD (dbh site)	0.018
Cor (Intercept dbh site)	-
	1.000
SD (Observations)	3.037
Num.Obs.	1000
R2 Marg.	0.871
AIC	5117.1
BIC	5146.5
RMSE	3.02

## Visualising model: visreg

```
visreg(mixed.slopes, xvar = "dbh", by = "site")
```



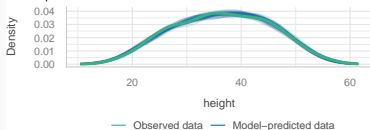
```
plot_model(mixed.slopes, type = 're')
```



## check\_model(mixed.slopes)

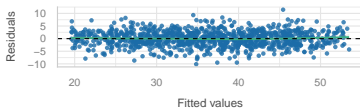
### Posterior Predictive Check

Model-predicted lines should resemble observed data line



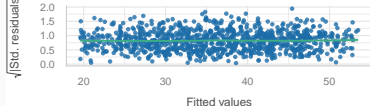
### Linearity

Reference line should be flat and horizontal



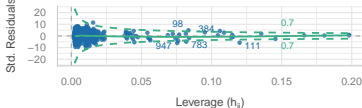
### Homogeneity of Variance

Reference line should be flat and horizontal



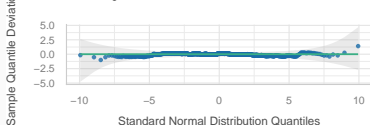
### Influential Observations

Points should be inside the contour lines



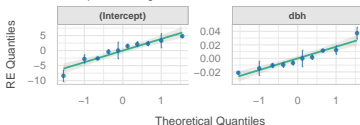
### Normality of Residuals

Dots should fall along the line



### Normality of Random Effects (site)

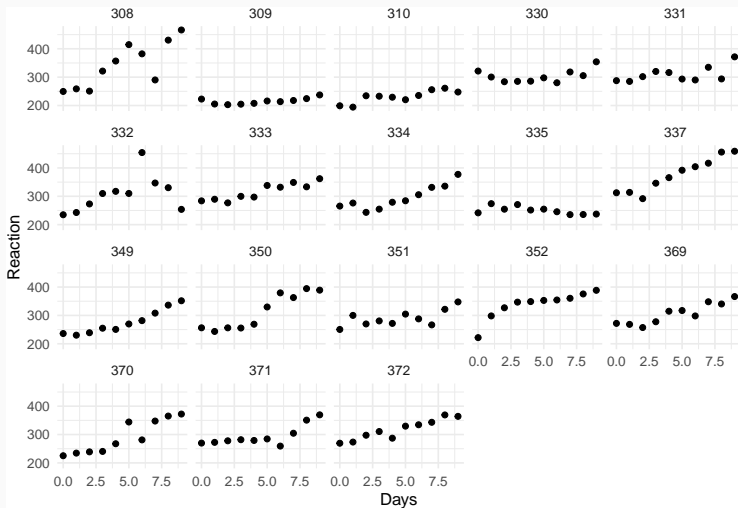
Dots should be plotted along the line



## 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:
```

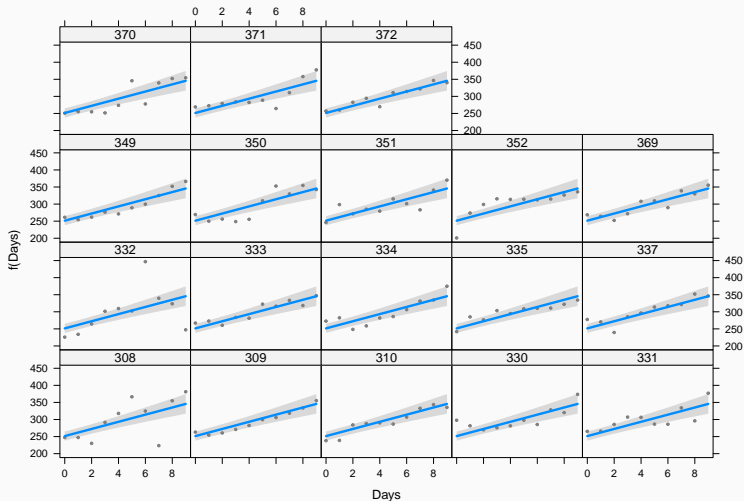
	Estimate	Std. Error	t value
(Intercept)	251.405	6.825	36.838
Days	10.467	1.546	6.771

```
Correlation of Fixed Effects:
```

```
(Intr)
```

## Varying intercepts and slopes (lme4)

```
visreg(sleep, xvar = 'Days', by = 'Subject')
```



## 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)	298.51	9.05	32.98	<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)	45.67	53	17.11	<2e-16 ***

---

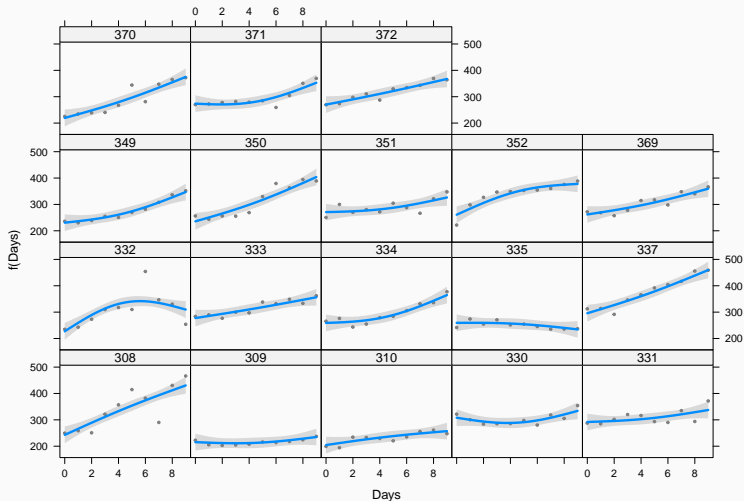
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) = 0.835 Deviance explained = 87.7%

-REML = 883.64 Scale est. = 523.2 n = 180

## Fitting multilevel models (GAMM) with mgcv

```
visreg(sgamma, xvar = 'Days', by = 'Subject')
```



## **Hierarchical generalized additive models: an introduction with mgcv**

Eric J Pedersen <sup>Corresp., 1,2</sup>, David L. Miller <sup>3,4</sup>, Gavin L. Simpson <sup>5</sup>, Noam Ross <sup>6</sup>

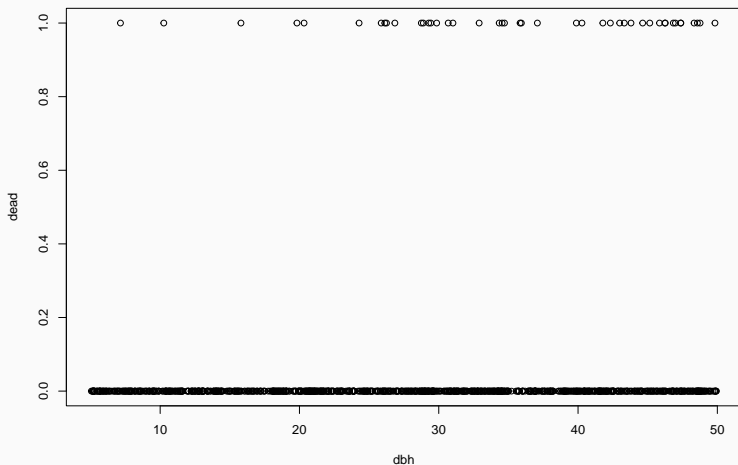
<https://doi.org/10.7287/peerj.preprints.27320v1>

## Multilevel logistic regression

---

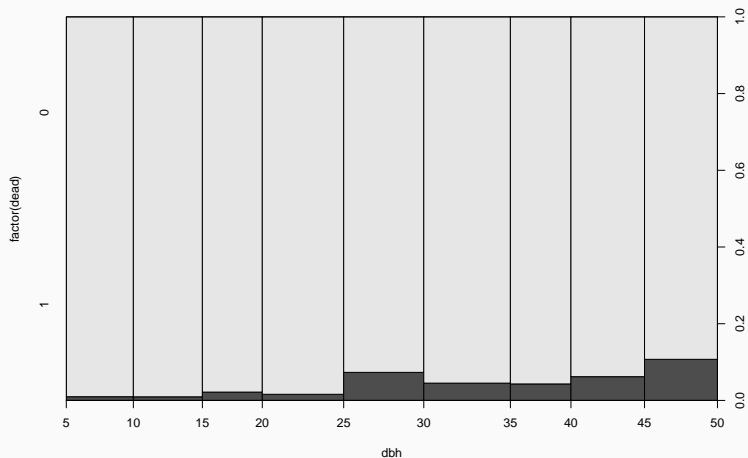
## Q: Relationship between tree size and mortality

```
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)
```

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  
AIC: 347.69

Number of Fisher Scoring iterations: 6

## Logistic regression with *independent* site effects

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

Call:

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

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	***
site2	-0.29692	0.46073	-0.644	0.5193	
site3	0.21275	0.52799	0.403	0.6870	
site4	0.39841	0.53025	0.751	0.4524	
site5	-0.42557	0.64018	-0.665	0.5062	
site6	0.66861	0.53656	1.246	0.2127	
site7	0.11862	1.06211	0.112	0.9111	
site8	0.43899	1.08058	0.406	0.6846	
site9	-13.63389	840.90382	-0.016	0.9871	
site10	-13.17148	1042.21823	-0.013	0.9899	

---

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: 338.58 on 989 degrees of freedom

# 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	-2*log(L)	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

Correlation of Fixed Effects:

(Intr)

dbh -0.952

optimizer (Nelder\_Mead) convergence code: 0 (OK)

boundary (singular) fit: see help('isSingular')

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

```
$site
```

```
  (Intercept)      dbh
```

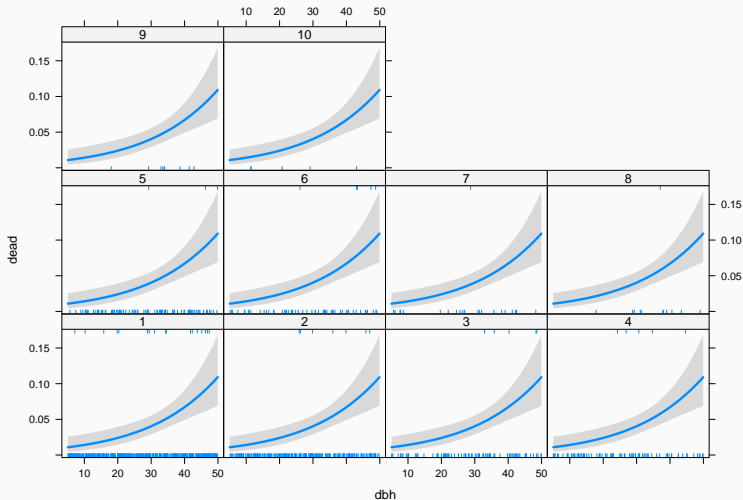
```
1   -4.778744  0.05364989
2   -4.778744  0.05364989
3   -4.778744  0.05364989
4   -4.778744  0.05364989
5   -4.778744  0.05364989
6   -4.778744  0.05364989
7   -4.778744  0.05364989
8   -4.778744  0.05364989
9   -4.778744  0.05364989
10  -4.778744  0.05364989
```

```
attr(,"class")
```

```
[1] "coef.mer"
```

## Visualising model: visreg

```
visreg(mixed.logis, xvar = 'dbh', by = 'site', scale = 'response')
```



## Advantages of multilevel models

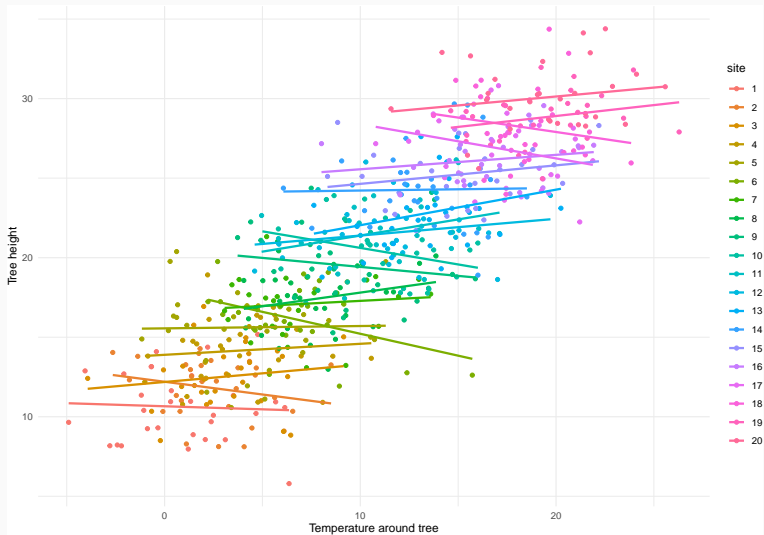
- Perfect for **structured data** (space-time)
- Predictors enter at the appropriate level
- Accommodate **variation** in treatment effects
- More **efficient inference** of regression parameters
- Using all the data to perform inferences for groups with **small sample size**
- Allow predictions for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

- 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)
  - $y \sim x + (1 + x \mid \text{group1}) + (1 + x \mid \text{group2})$

## Heterogeneity bias

---

# Predictor (temperature) correlates with groups (random site effects)



## Fitting a model with varying intercepts

```
Family: gaussian ( identity )  
Formula:      tree.height ~ tree.temp + (1 | site)  
Data: df
```

AIC	BIC	logLik	-2*log(L)	df.resid
2681.6	2699.2	-1336.8	2673.6	596

Random effects:

Conditional model:

Groups	Name	Variance	Std.Dev.
site	(Intercept)	31.119	5.578
Residual		4.212	2.052

Number of obs: 600, groups: site, 20

Dispersion estimate for gaussian family ( $\sigma^2$ ): 4.21

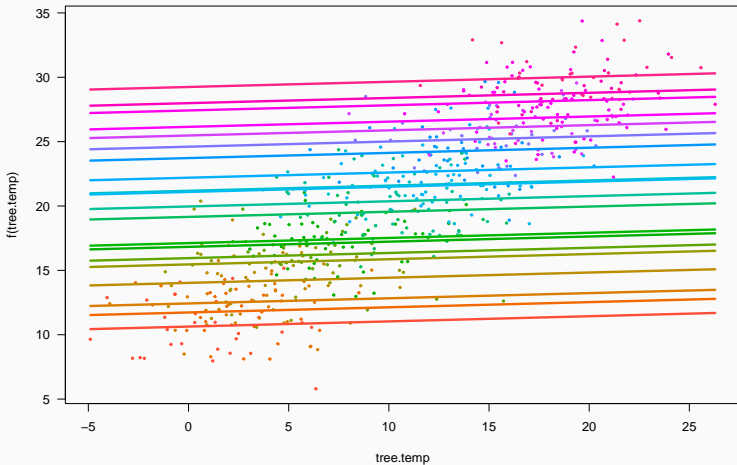
Conditional model:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	20.11896	1.28855	15.614	<2e-16 ***
tree.temp	0.04021	0.02954	1.361	0.173

---

# Fitting a model with varying intercepts

```
visreg(m, xvar = "tree.temp", by = "site", overlay = T, band = F, legend = F)
```



# Fitting a model with varying intercepts and slopes

```
Family: gaussian ( identity )  
Formula:      tree.height ~ tree.temp + (1 + tree.temp | site)  
Data: df
```

AIC	BIC	logLik	-2*log(L)	df.resid
2684.7	2711.1	-1336.4	2672.7	594

Random effects:

Conditional model:

Groups	Name	Variance	Std.Dev.	Corr
site	(Intercept)	2.813e+01	5.30422	
	tree.temp	7.845e-04	0.02801	0.99
Residual		4.205e+00	2.05054	

Number of obs: 600, groups: site, 20

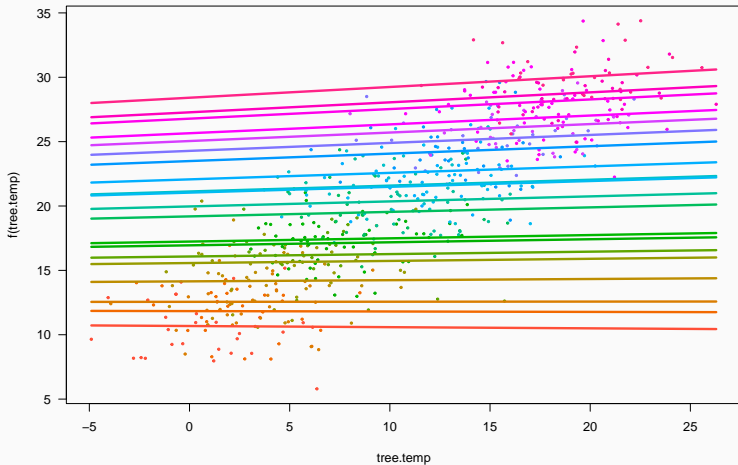
Dispersion estimate for gaussian family ( $\sigma^2$ ): 4.2

Conditional model:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	19.96903	1.24001	16.104	<2e-16 ***
tree.temp	0.03936	0.03021	1.303	0.193

## Fitting a model with varying intercepts and slopes

```
visreg(m, xvar = "tree.temp", by = "site", overlay = T, band = F, legend = F)
```



## Do predictors vary within- and between groups?

```
performance::check_group_variation(m)
```

Check site variation

```
Variable | Variation | Design
```

```
-----
```

```
tree.temp |      both |
```

## Group-mean centering

```
df <- datawizard::demmean(df, select = "tree.temp", by = "site")  
head(df)
```

	site	tree	tree.temp	tree.height	tree.temp_between	tree.temp_within
1	1	1	-0.6814269	13.148025	0.8586887	-1.54011
2	1	2	0.3094675	10.945306	0.8586887	-0.54922
3	1	3	5.6761249	10.933339	0.8586887	4.81743
4	1	4	1.2115252	7.967865	0.8586887	0.35283
5	1	5	1.3878632	12.580771	0.8586887	0.52917
6	1	6	6.1451950	10.578532	0.8586887	5.28650

# Model with site effects and temperature (between & within)

```
m <- glmmTMB(tree.height ~ tree.temp_between + tree.temp_within + (1|site), data = df)
```

Family: gaussian ( identity )

Formula:

```
tree.height ~ tree.temp_between + tree.temp_within + (1 | site)
```

Data: df

AIC	BIC	logLik	-2*log(L)	df.resid
2589.1	2611.1	-1289.6	2579.1	595

Random effects:

Conditional model:

Groups	Name	Variance	Std.Dev.
site	(Intercept)	0.1429	0.3781
Residual		4.2093	2.0516

Number of obs: 600, groups: site, 20

Dispersion estimate for gaussian family (sigma<sup>2</sup>): 4.21

Conditional model:

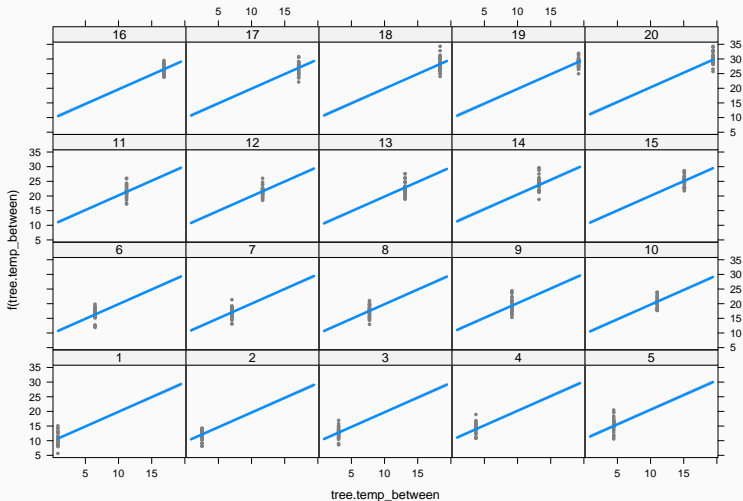
	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	9.96527	0.24758	40.25	<2e-16 ***
tree.temp_between	1.00125	0.02055	48.73	<2e-16 ***
tree.temp_within	0.02256	0.02925	0.77	0.44

---

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

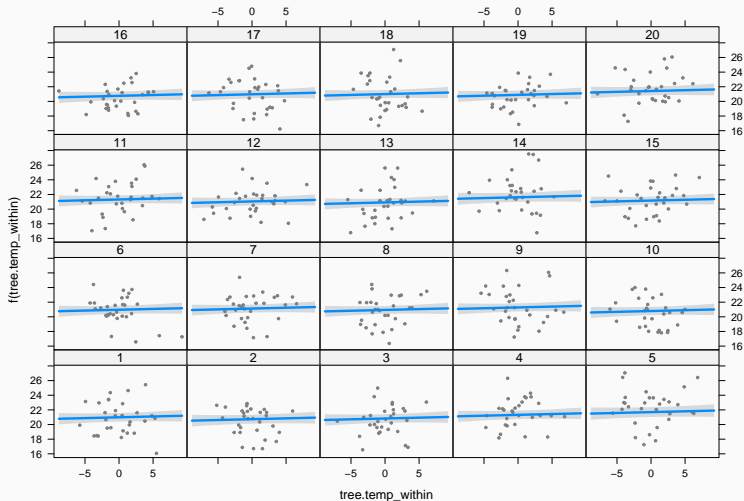
# Model with site effects and temperature (between & within)

```
visreg(m, xvar = "tree.temp_between", by = "site")
```



# Model with site effects and temperature (between & within)

```
visreg(m, xvar = "tree.temp_within", by = "site")
```



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

<http://mfviz.com/hierarchical-models/>

<https://www.jstor.org/stable/23736938>

<https://m-clark.github.io/mixed-models-with-R/>

[https://www.middleprofessor.com/files/applied-biostatistics\\_bookdown/\\_book/lmm](https://www.middleprofessor.com/files/applied-biostatistics_bookdown/_book/lmm)

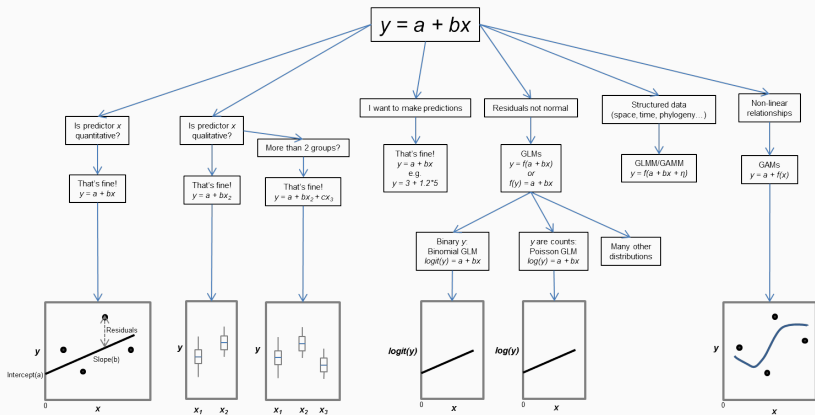
<https://bookdown.org/roback/bookdown-BeyondMLR/>

<https://www.learn-mlms.com/>

<https://doi.org/10.7717/peerj.4794>

<https://doi.org/10.7717/peerj.9522>

- **Starlings**: body mass growth depending on nest type
- **Mixed\_binom**: Species presence/absence ~ environment
- **Mixed\_count**: Species counts ~ environment





Source code and materials:

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

```
knitr::knit_exit()
```