

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

Francisco Rodriguez-Sanchez

[http://bit.ly/frod\\_san](http://bit.ly/frod_san)

## Introduction to linear models

# Modern statistics are easier than this

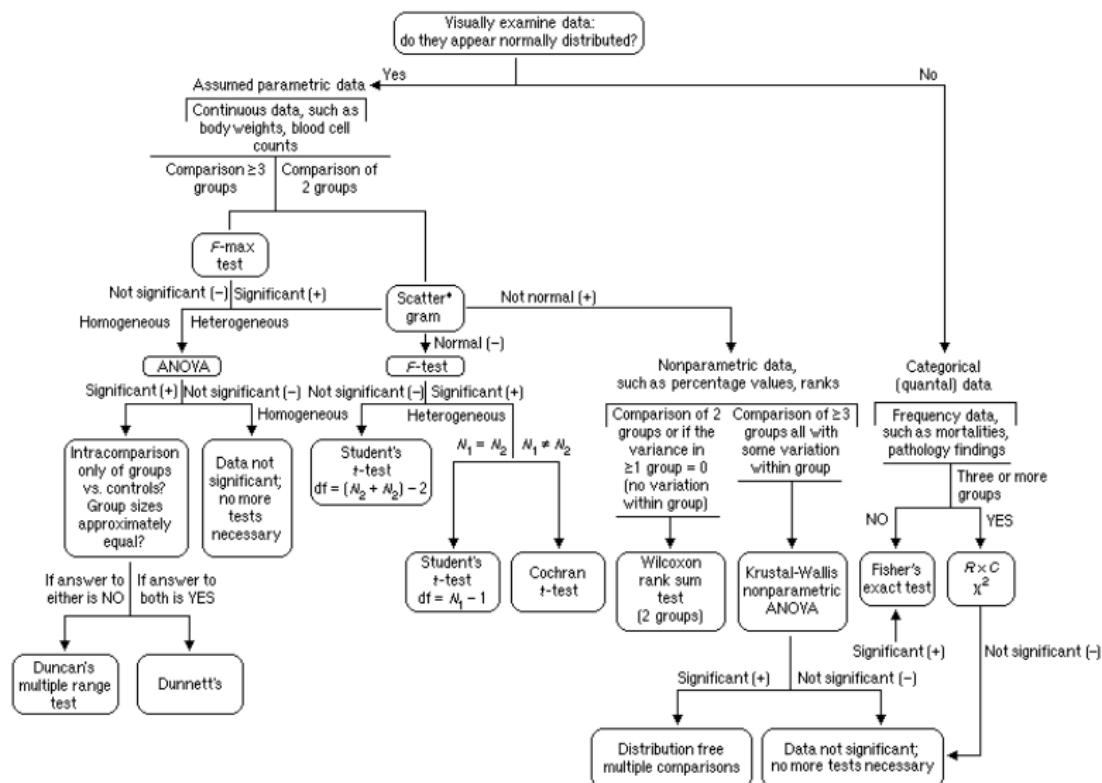
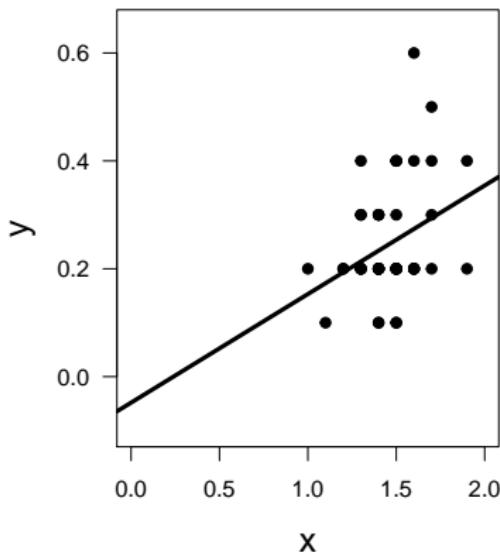


Figure 1

# Our overarching regression framework

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

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



## Data

$y$  = response variable

$x$  = predictor

## Parameters

$a$  = intercept

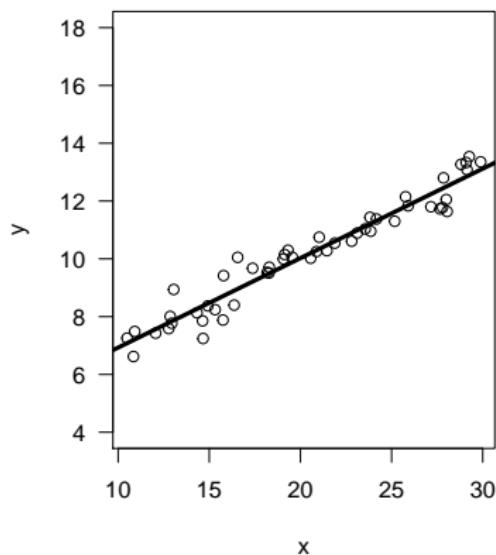
$b$  = slope

$\sigma$  = residual variation

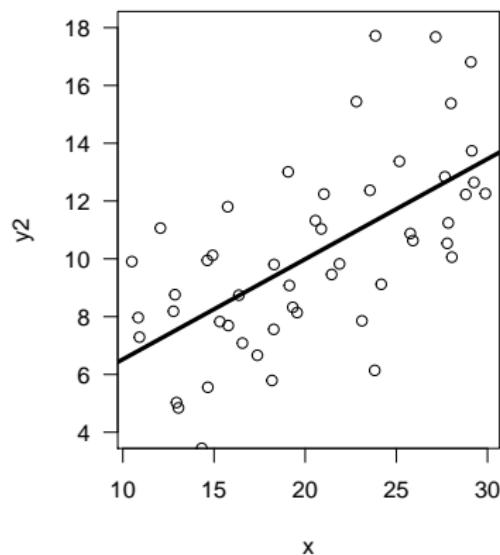
$\varepsilon$  = residuals

## Residual variation (error)

**small**



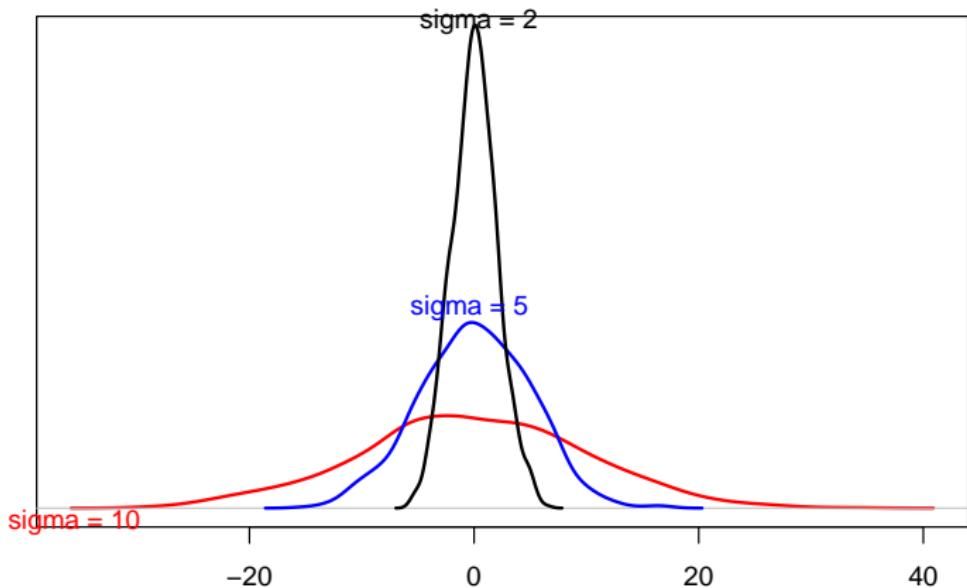
**large**



## Residual variation

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

**Distribution of residuals**



## In a Normal distribution

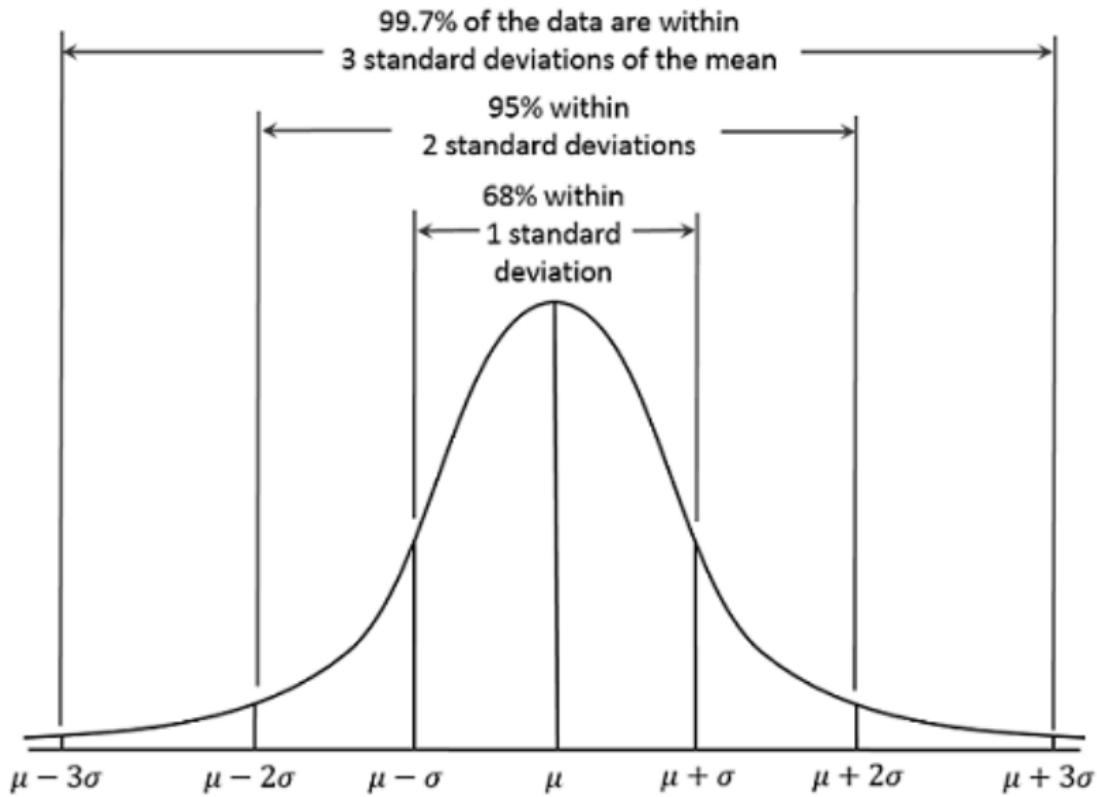


Figure 2

## Different ways to write same model

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

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

## Linear models

## Example dataset: forest trees

- ▶ Go to <https://tinyurl.com/treesdata>

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

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

## Example dataset: forest trees

- ▶ Go to <https://tinyurl.com/treesdata>
- ▶ Download zip file and uncompress (within your project folder!)

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

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

## Questions

- ▶ What is the relationship between DBH and height?

## Questions

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

## Questions

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

Always plot your data first!

# Always plot your data first!

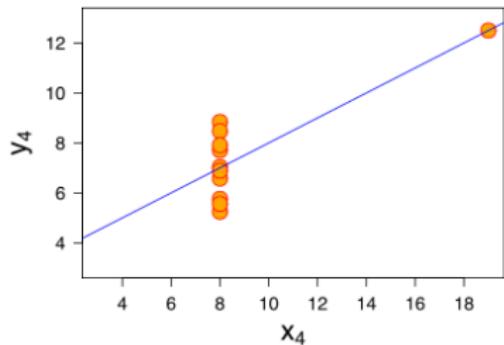
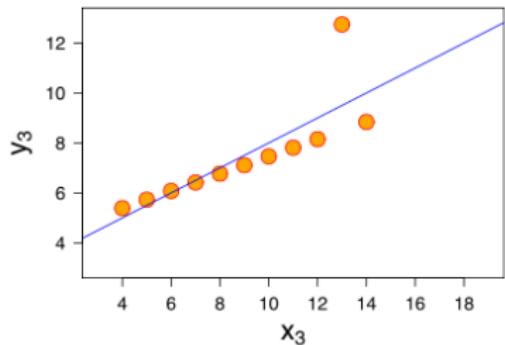
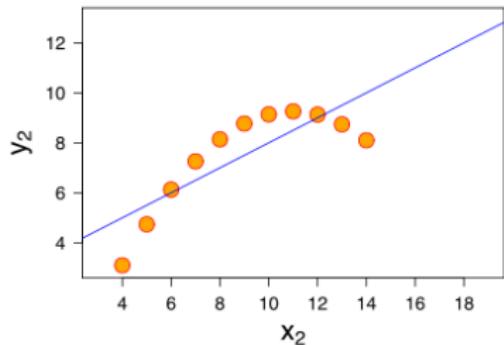
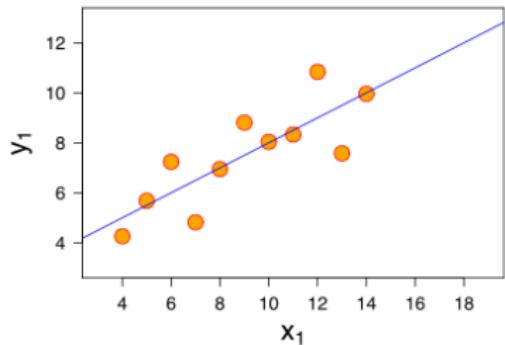
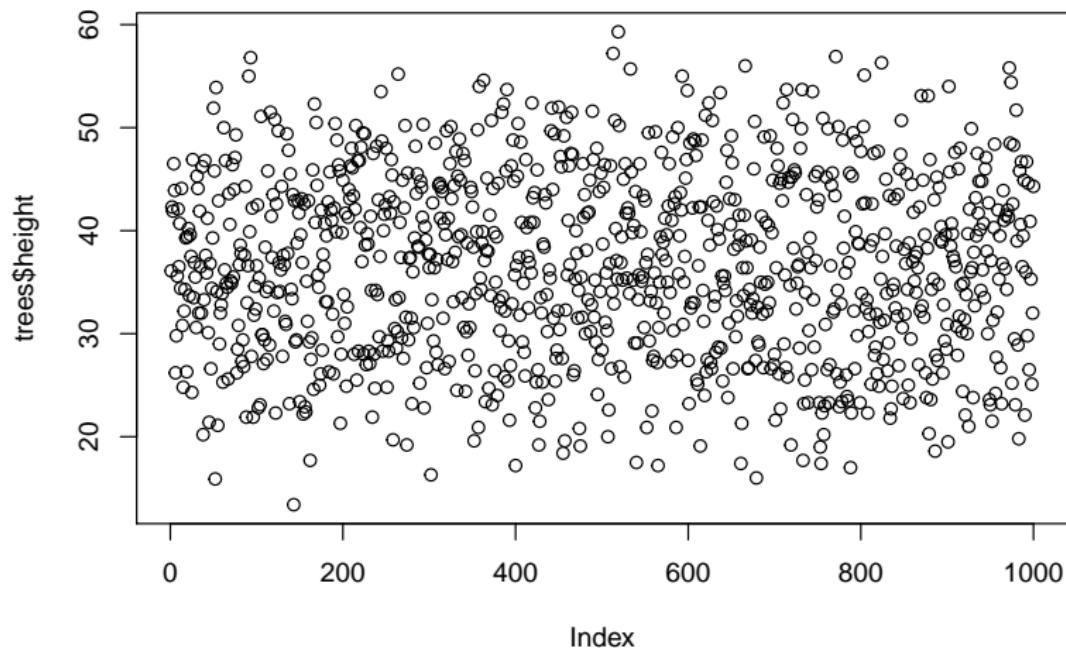


Figure 3

# Exploratory Data Analysis (EDA)

## Outliers

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



## Outliers impact on regression

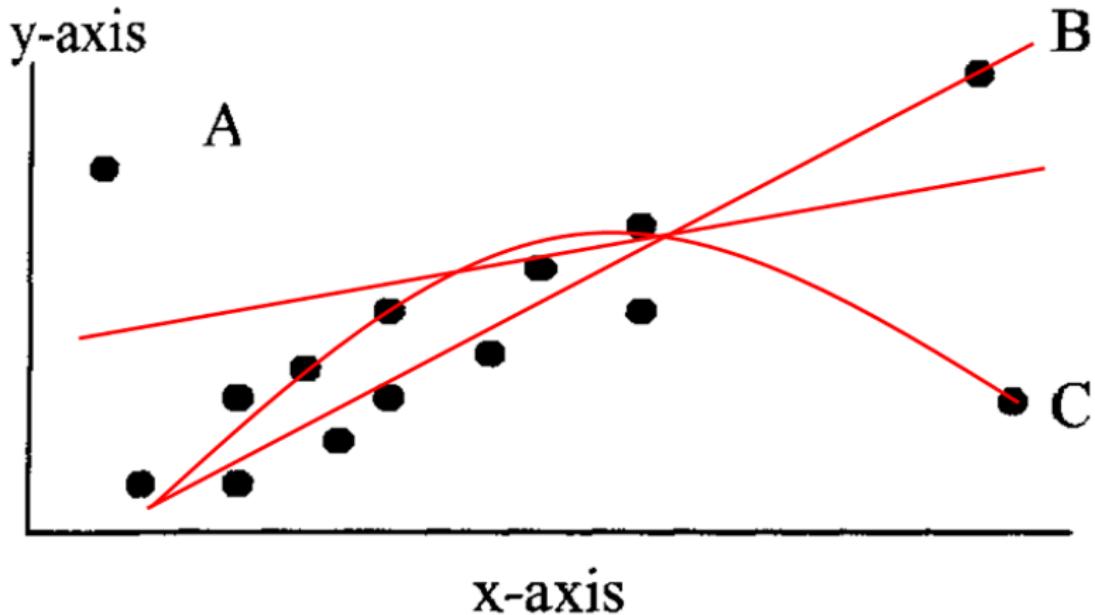
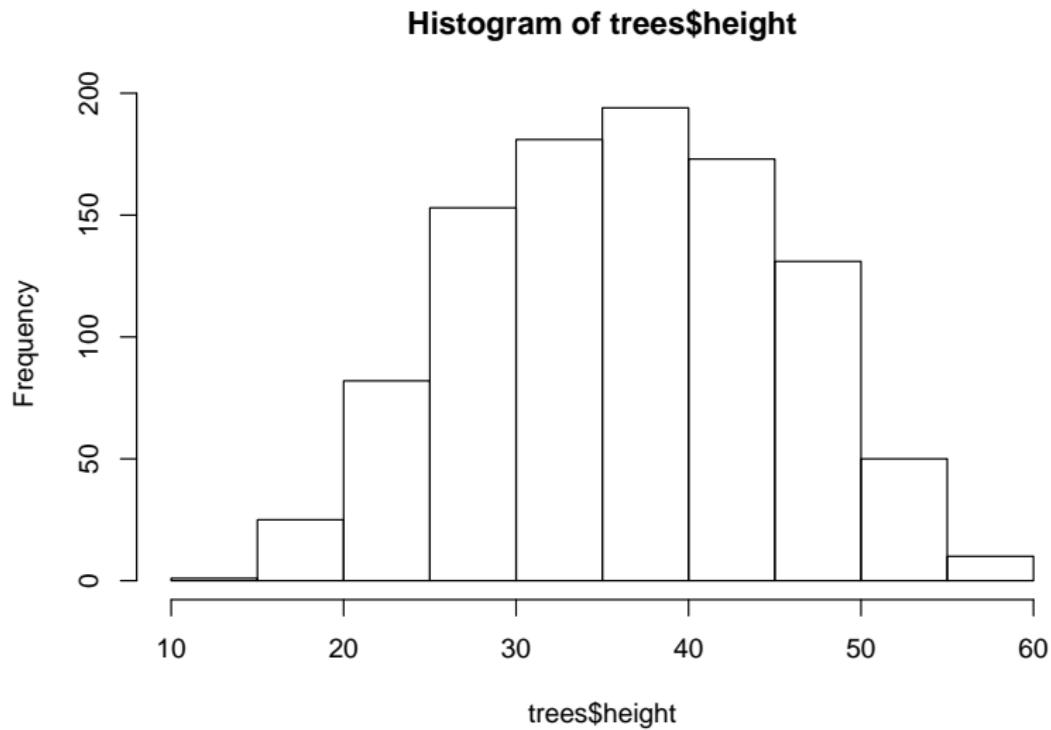


Figure 4

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

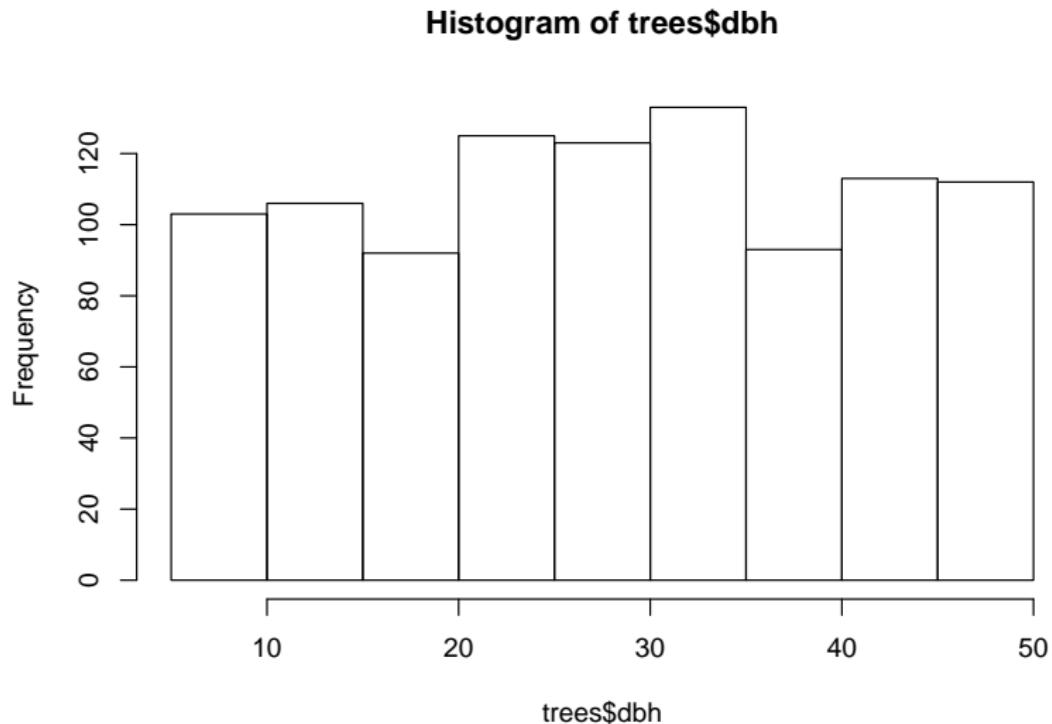
# Histogram of response variable

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



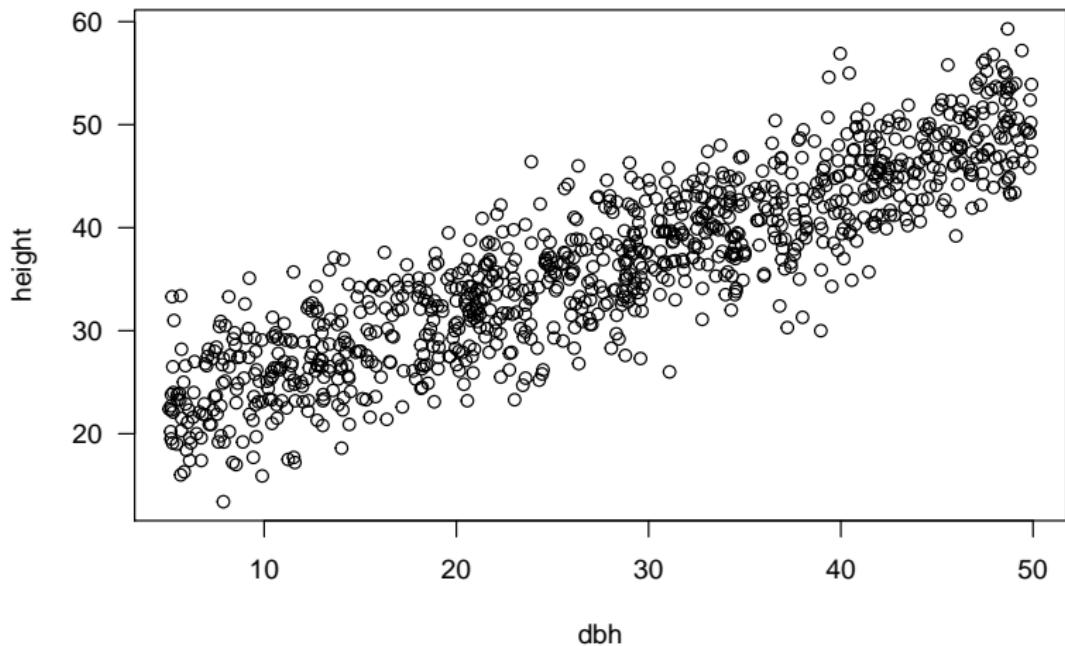
# Histogram of predictor variable

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



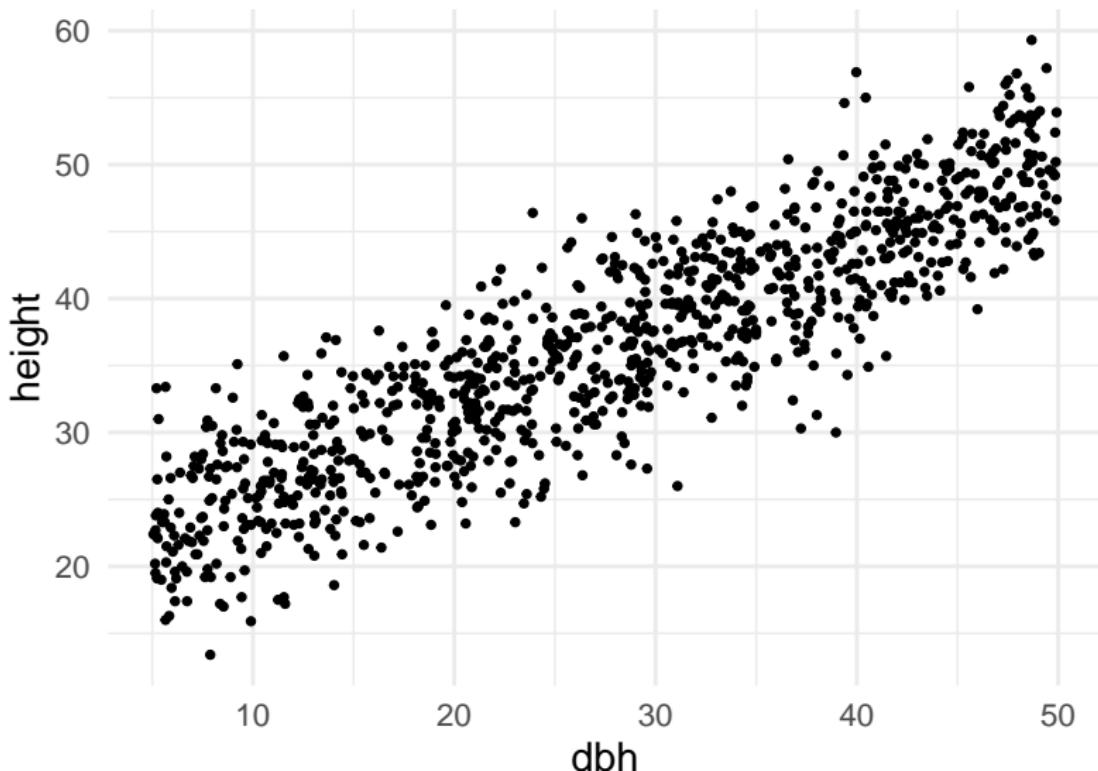
# Scatterplot

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



## Scatterplot

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



Model fitting

Now fit model

Hint: 1m

## Now fit model

Hint: `lm`

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

which corresponds to

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

## Model interpretation

## What does this mean?

Call:

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

Residuals:

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

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	19.33920	0.31064	62.26	<2e-16 ***
dbh	0.61570	0.01013	60.79	<2e-16 ***
---				
Signif. codes:	0 ***	0.001 **	0.01 *	0.05 .
	0.1	'	'	1

Residual standard error: 4.093 on 998 degrees of freedom

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

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

# Avoid dichotomania of statistical significance



EDITORIAL • 20 MARCH 2019

## It's time to talk about ditching statistical significance

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

# Avoid dichotomania of statistical significance



EDITORIAL • 20 MARCH 2019

## It's time to talk about ditching statistical significance

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

# Avoid dichotomania of statistical significance



EDITORIAL • 20 MARCH 2019

## It's time to talk about ditching statistical significance

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

## Communicating results

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

## Presenting model results

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

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

## Presenting model results

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

Model 1	
(Intercept)	19.34 (0.31)***
dbh	0.62 (0.01)***
$R^2$	0.79
Adj. $R^2$	0.79
Num. obs.	1000
RMSE	4.09

\*\*\*  $p < 0.001$ , \*\*  $p < 0.01$ , \*  $p < 0.05$

Table 2: Statistical models

## Retrieving model coefficients

```
coef(m1)
```

	dbh
(Intercept)	19.3391968
	0.6157036

# Tidy up model coefficients with broom

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

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

```
glance(m1)
```

```
# A tibble: 1 x 11
  r.squared adj.r.squared sigma statistic p.value    df logLik    AIC    BIC
  <dbl>        <dbl> <dbl>     <dbl>     <dbl> <int> <dbl> <dbl> <dbl>
1 0.787        0.787  4.09     3695.      0      2 -2827. 5660. 5675.
# ... with 2 more variables: deviance <dbl>, df.residual <int>
```

## Confidence intervals

```
confint(m1)
```

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

## Using effects package

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

model: height ~ dbh

dbh effect

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

Lower 95 Percent Confidence Limits

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

Upper 95 Percent Confidence Limits

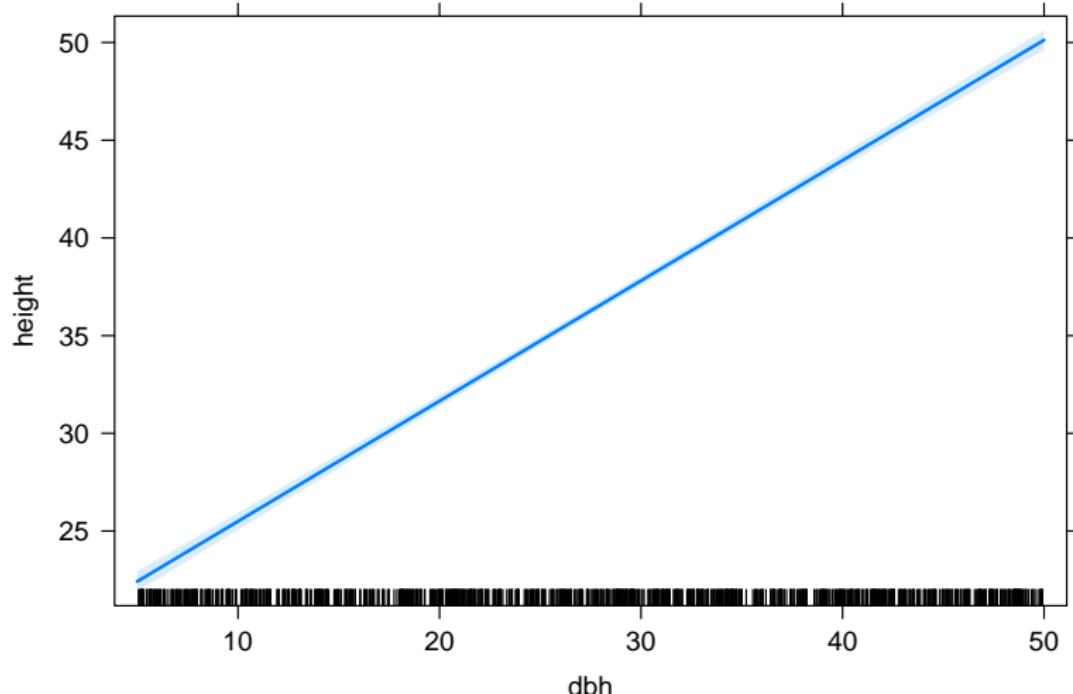
dbh	5	20	30	40	50
dbh	22.93861	31.95167	38.06774	44.31735	50.63207

Visualising fitted model

## Plot effects

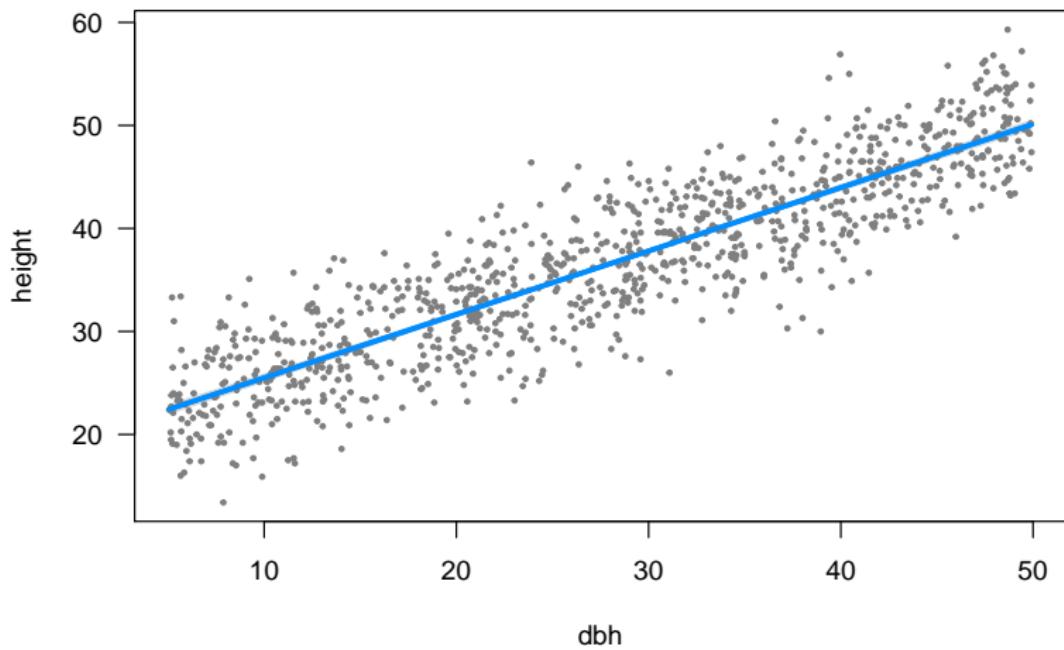
```
plot(allEffects(m1))
```

**dbh effect plot**



## Plot model (visreg)

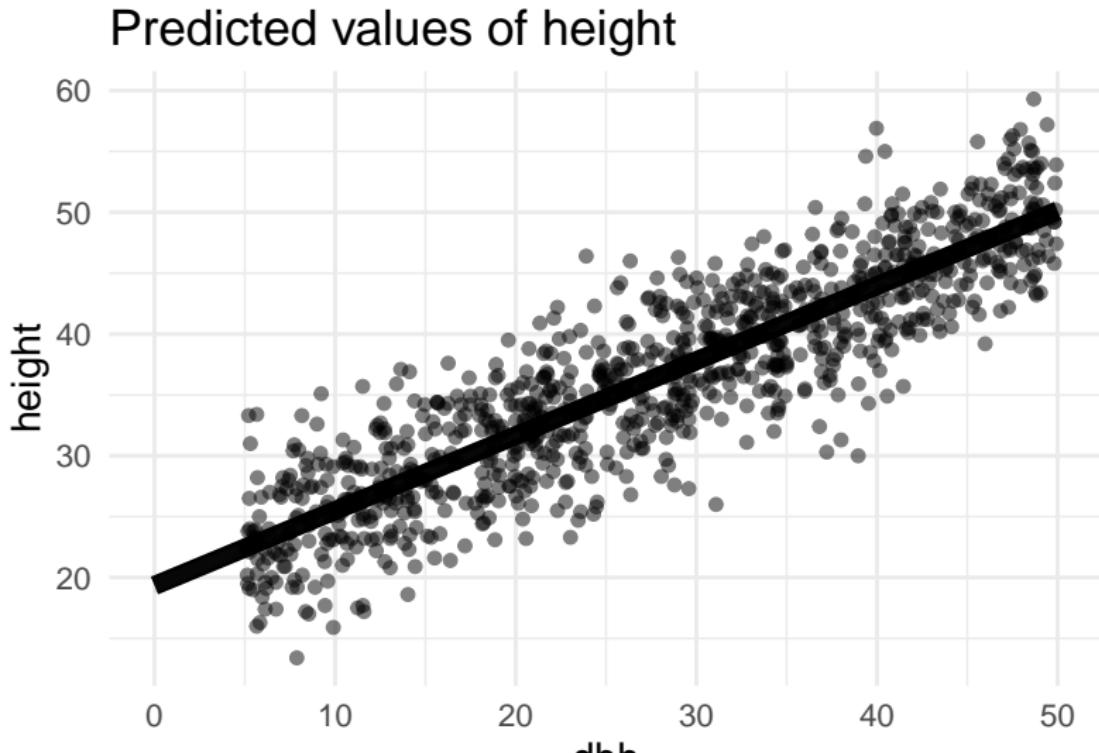
```
library(visreg)  
visreg(m1)
```



## Plot model (sjPlot - ggplot2)

```
sjPlot::plot_model(m1, type = "eff", show.data = TRUE, line.size = 2)
```

\$dbh



Model checking

## Linear model assumptions

- ▶ Linearity (transformations, GAM...)

## Linear model assumptions

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

## Linear model assumptions

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

## Linear model assumptions

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

## Linear model assumptions

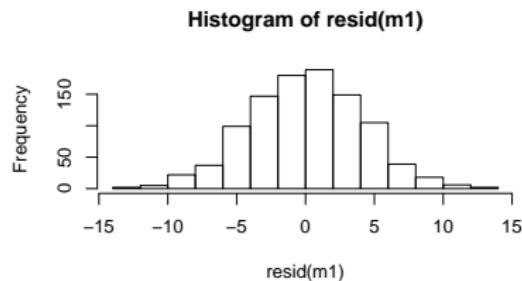
- ▶ Linearity (transformations, GAM...)
- ▶ Residuals:
  - ▶ Independent
  - ▶ Equal variance
  - ▶ Normal

## Linear model assumptions

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

# Are residuals normal?

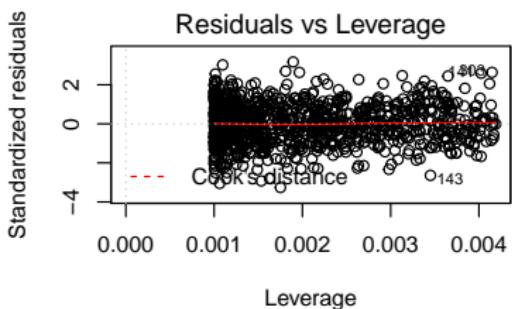
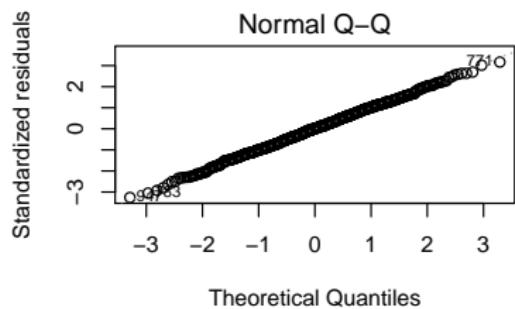
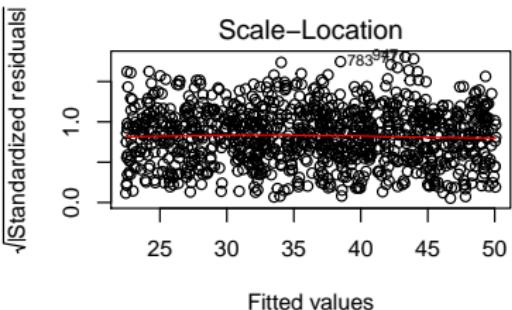
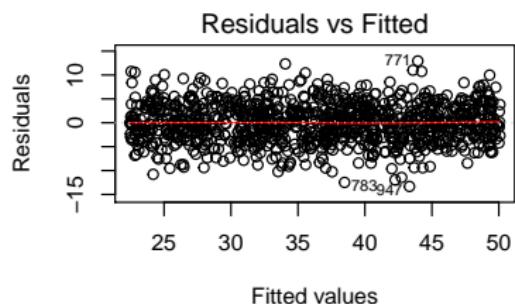
```
hist(resid(m1))
```



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

SD of residuals = 4.09 coincides with estimate of sigma.

# Model checking: residuals

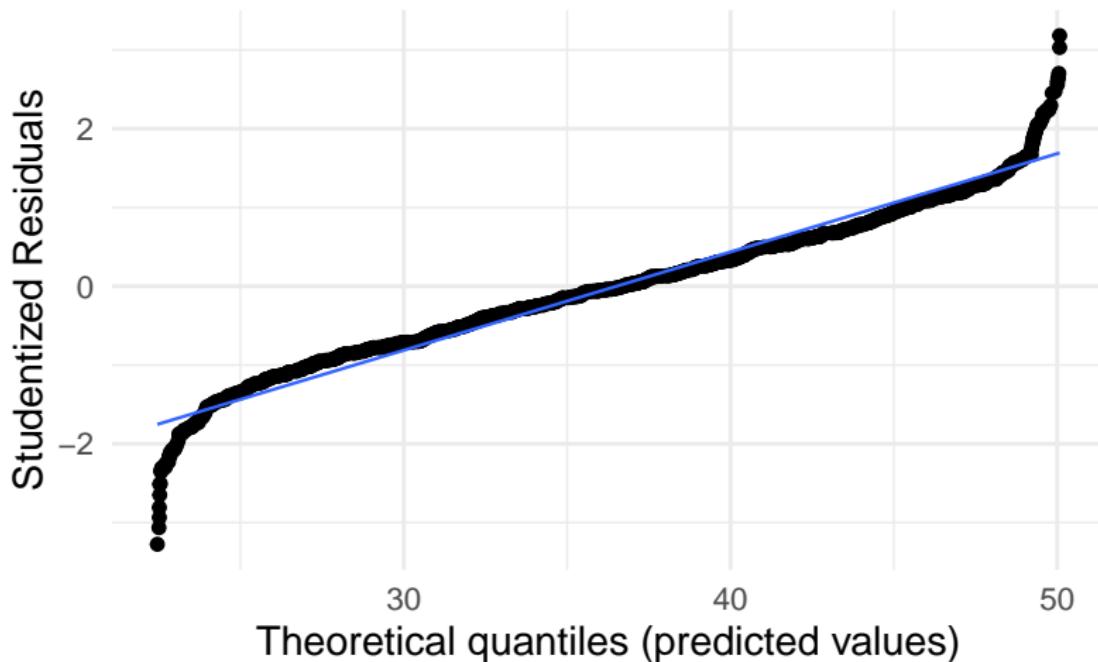


## Model checking (sjPlot)

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

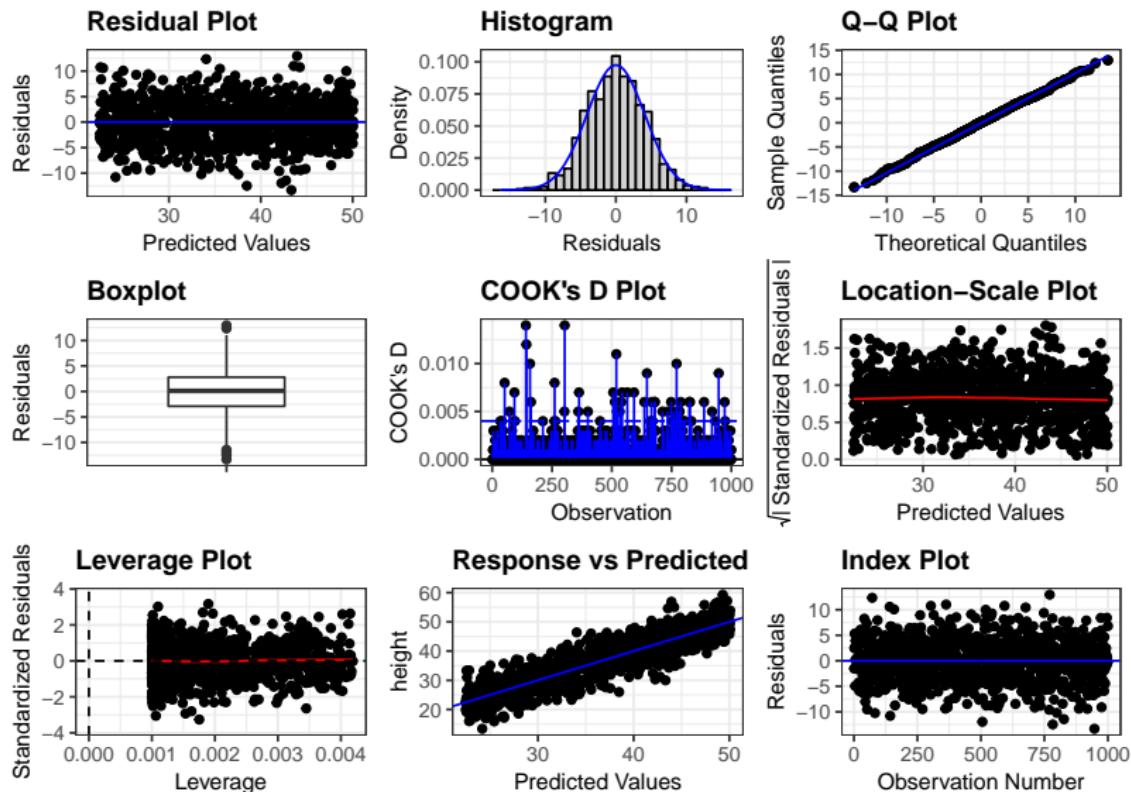
Non-normality of residuals and outliers

Dots should be plotted along the line



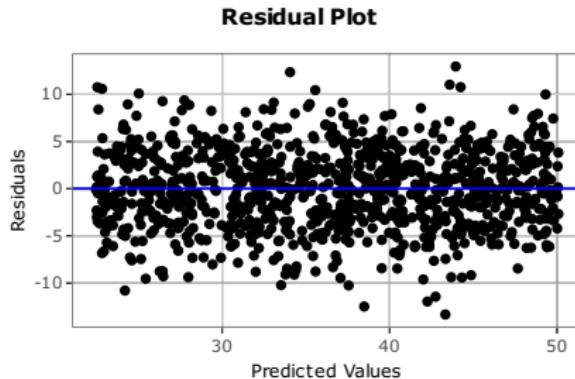
# Model checking (ggResidpanel)

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



# Interactive model checking (ggResidpanel)

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



Using model for prediction

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

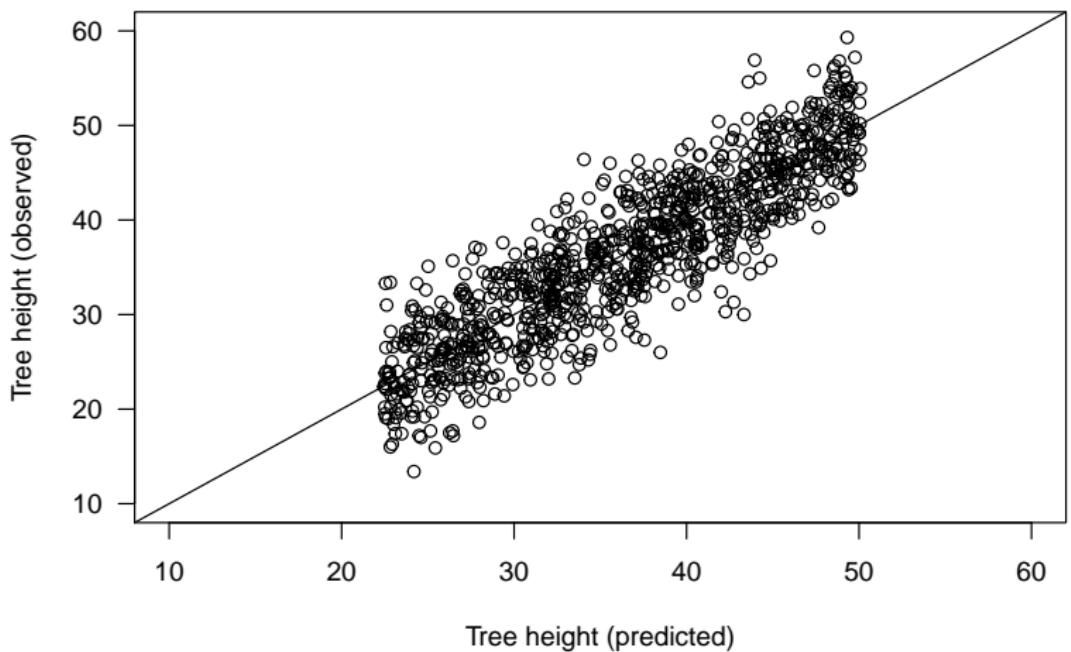
`fitted` gives predictions for each observation

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

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

## Calibration plot: Observed vs Predicted values

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



## Using fitted model for prediction

Q: Expected tree height if DBH = 39 cm?

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

```
$fit
```

```
1
```

```
43.35164
```

```
$se.fit
```

```
[1] 0.1715514
```

```
$df
```

```
[1] 998
```

```
$residual.scale
```

```
[1] 4.092629
```

## Using fitted model for prediction

Q: Expected tree height if DBH = 39 cm?

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

	fit	lwr	upr
1	43.35164	43.01499	43.68828

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

	fit	lwr	upr
1	43.35164	35.31344	51.38983

# Workflow

- ▶ **Visualise data**

# Workflow

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

# Workflow

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

# Workflow

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

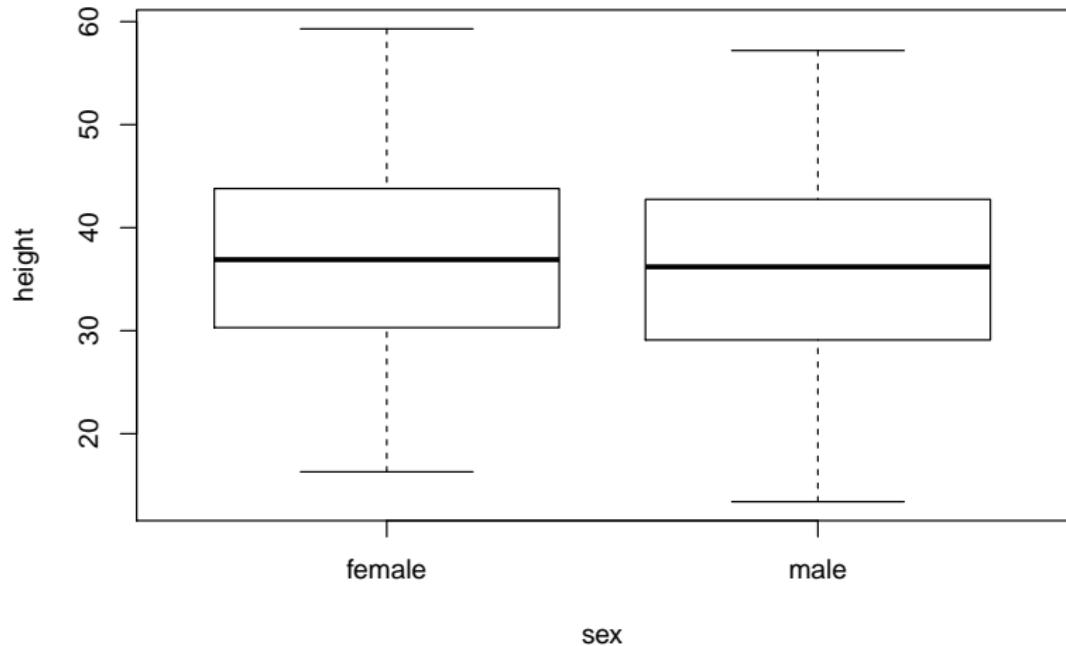
# Workflow

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

Categorical predictors (factors)

## Q: Does tree height vary with sex?

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



## Model height ~ sex

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

Call:

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

Residuals:

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

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )		
(Intercept)	36.9312	0.3981	92.778	<2e-16 ***		
sexmale	-0.8432	0.5607	-1.504	0.133		
---						
Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'	0.1 ' '	1

Residual standard error: 8.865 on 998 degrees of freedom

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

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

## Linear model with categorical predictors

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

corresponds to

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

## Model height ~ sex

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

Call:

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

Residuals:

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

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )		
(Intercept)	36.9312	0.3981	92.778	<2e-16 ***		
sexmale	-0.8432	0.5607	-1.504	0.133		
---						
Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'	0.1 ' '	1

Residual standard error: 8.865 on 998 degrees of freedom

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

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

## Presenting model results

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

## Effects: Height ~ sex

Compare CIs

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

model: height ~ sex

sex effect

sex

	female	male
36.93125	36.08810	

Lower 95 Percent Confidence Limits

sex

	female	male
36.15012	35.31319	

Upper 95 Percent Confidence Limits

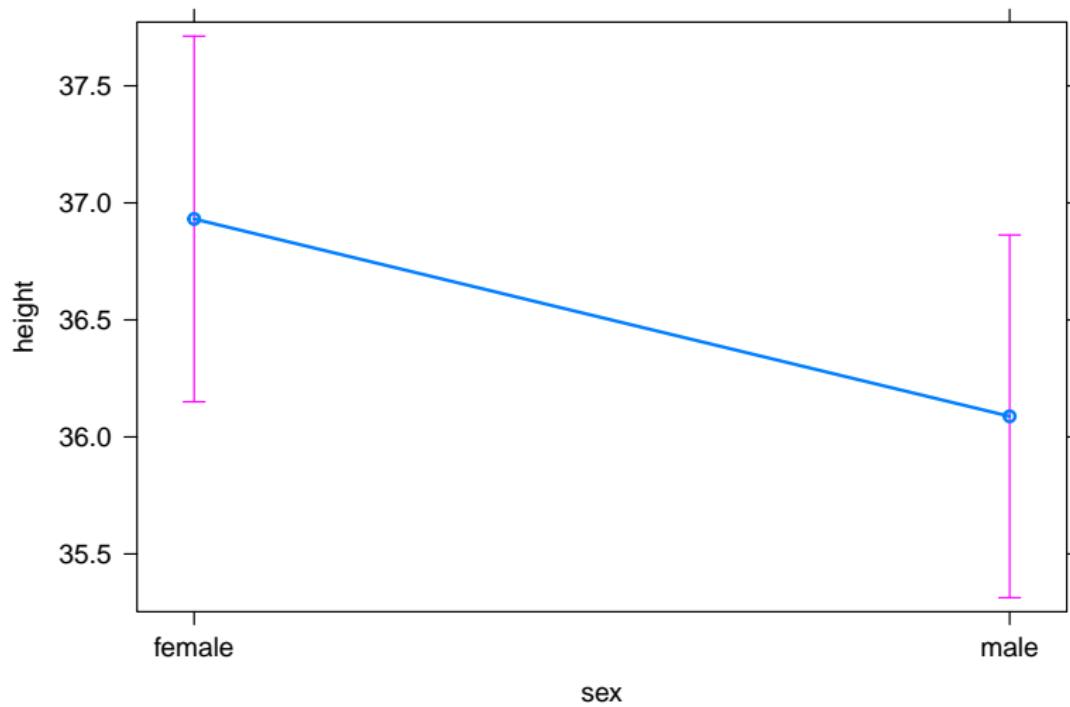
sex

	female	male
37.71238	36.86300	

# Plot

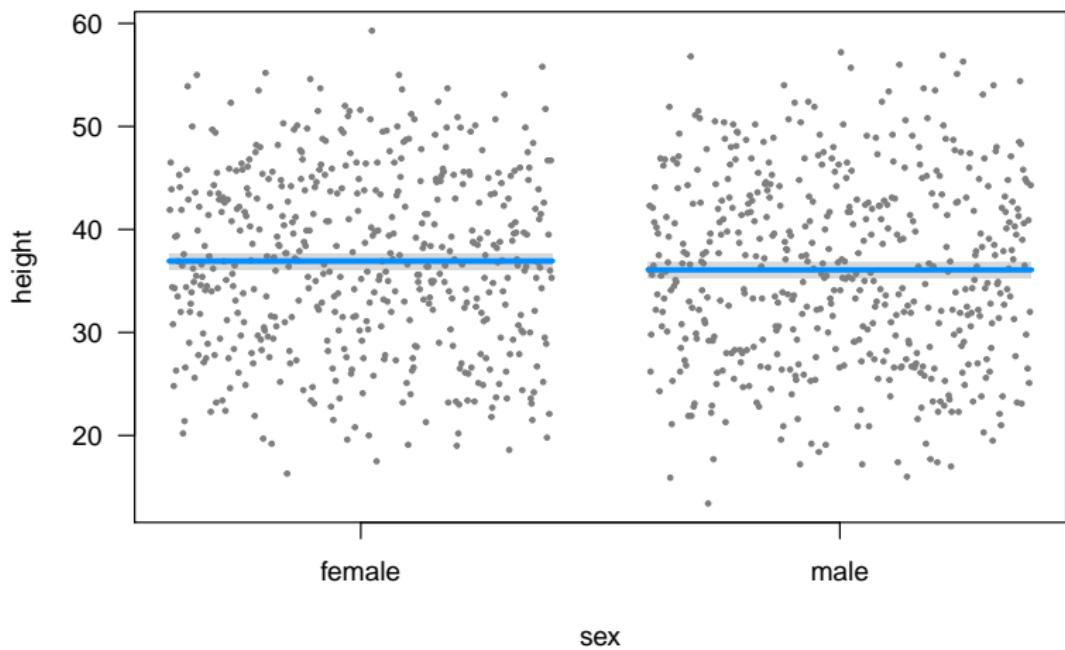
```
plot(allEffects(m2))
```

**sex effect plot**



# Plot (visreg)

```
visreg(m2)
```



## Plot model (sjPlot)

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

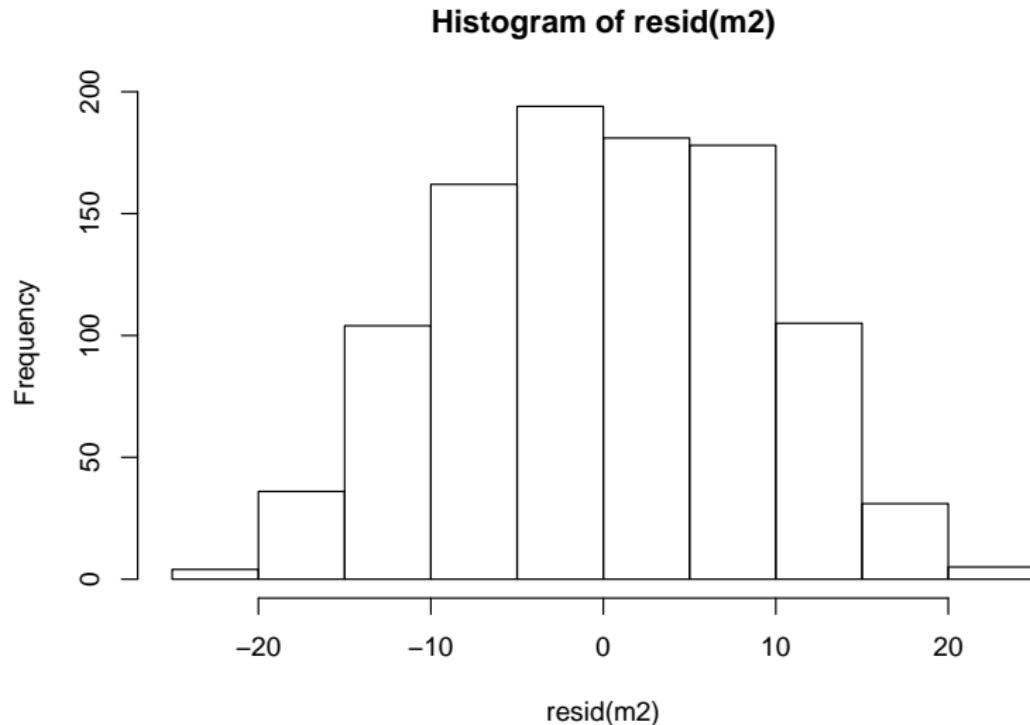
\$sex

Predicted values of height

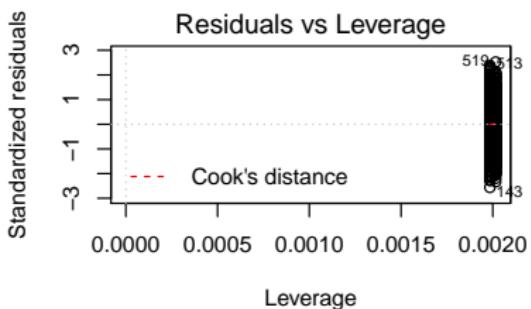
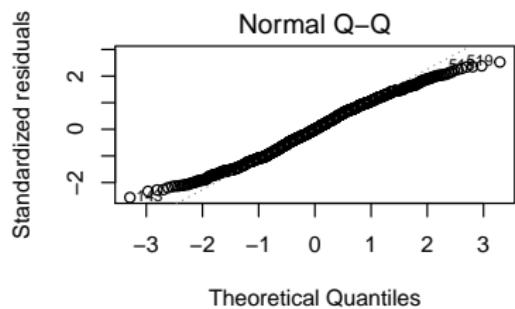
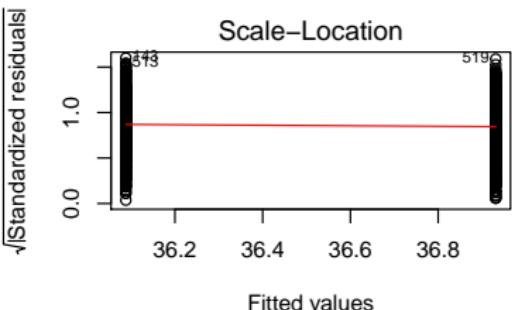
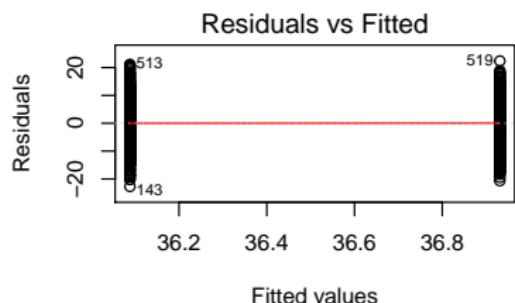


## Model checking: residuals

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

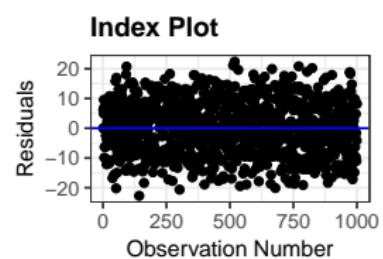
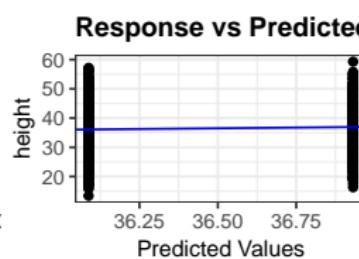
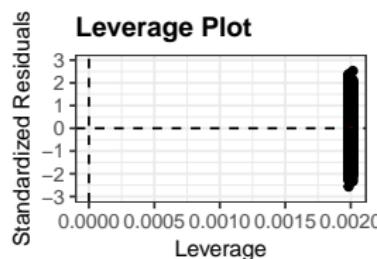
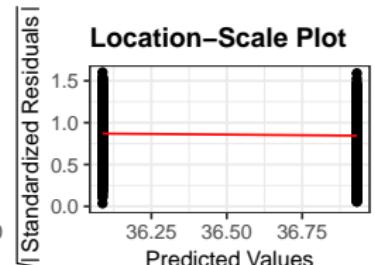
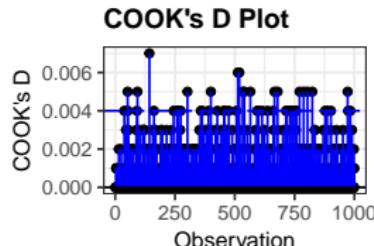
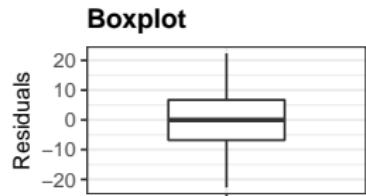
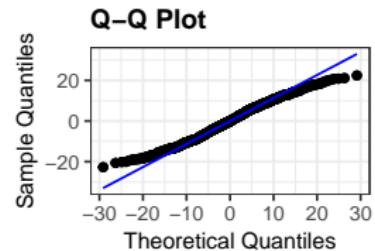
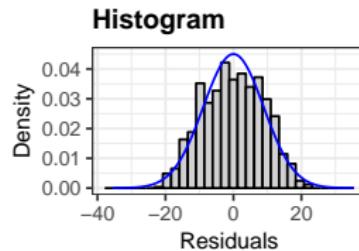
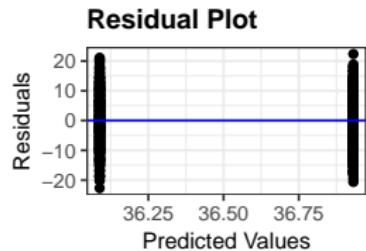


# Model checking: residuals



# Model checking (ggResidpanel)

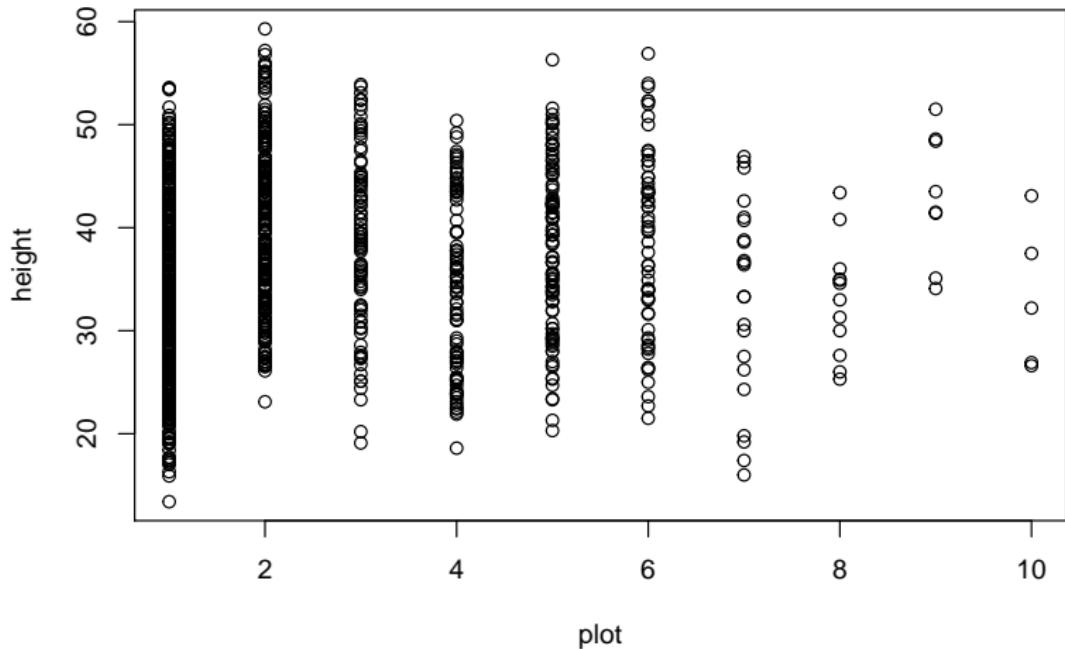
```
ggResidpanel:::resid_panel(m2, plots = "all")
```



Q: Does height differ among field plots?

## Plot data first

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



## Linear model with categorical predictors

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

$$y_i = a + b_{plot2} + c_{plot3} + d_{plot4} + e_{plot5} + \dots + \varepsilon_i$$
$$\varepsilon_i \sim N(0, \sigma^2)$$

## Model Height ~ Plot

All right here?

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

Call:

```
lm(formula = height ~ plot, 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 ***
plot	0.3862	0.1413	2.733	0.00639 **
---				
Signif. codes:	0 ***	0.001 **	0.01 *	0.05 .
	1			

Residual standard error: 8.842 on 998 degrees of freedom

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

## Plot is a factor!

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

# Model Height ~ Plot

Call:

```
lm(formula = height ~ plot, 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 ***
plot2	6.3411	0.7126	8.899	< 2e-16 ***
plot3	4.9991	0.9828	5.086	4.36e-07 ***
plot4	0.5329	0.9872	0.540	0.58949
plot5	4.3723	0.9425	4.639	3.97e-06 ***
plot6	4.7601	1.1709	4.065	5.18e-05 ***
plot7	-0.7416	1.8506	-0.401	0.68871
plot8	-0.6832	2.4753	-0.276	0.78258
plot9	9.1709	3.0165	3.040	0.00243 **
plot10	-0.5816	3.8013	-0.153	0.87843
---				

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

Residual standard error: 8.446 on 990 degrees of freedom

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

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

## Presenting model results

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

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	33.84	0.43	79.33	0.00
plot2	6.34	0.71	8.90	0.00
plot3	5.00	0.98	5.09	0.00
plot4	0.53	0.99	0.54	0.59
plot5	4.37	0.94	4.64	0.00
plot6	4.76	1.17	4.07	0.00
plot7	-0.74	1.85	-0.40	0.69
plot8	-0.68	2.48	-0.28	0.78
plot9	9.17	3.02	3.04	0.00
plot10	-0.58	3.80	-0.15	0.88

## Estimated tree heights for each site

```
summary(allEffects(m3))

model: height ~ plot

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

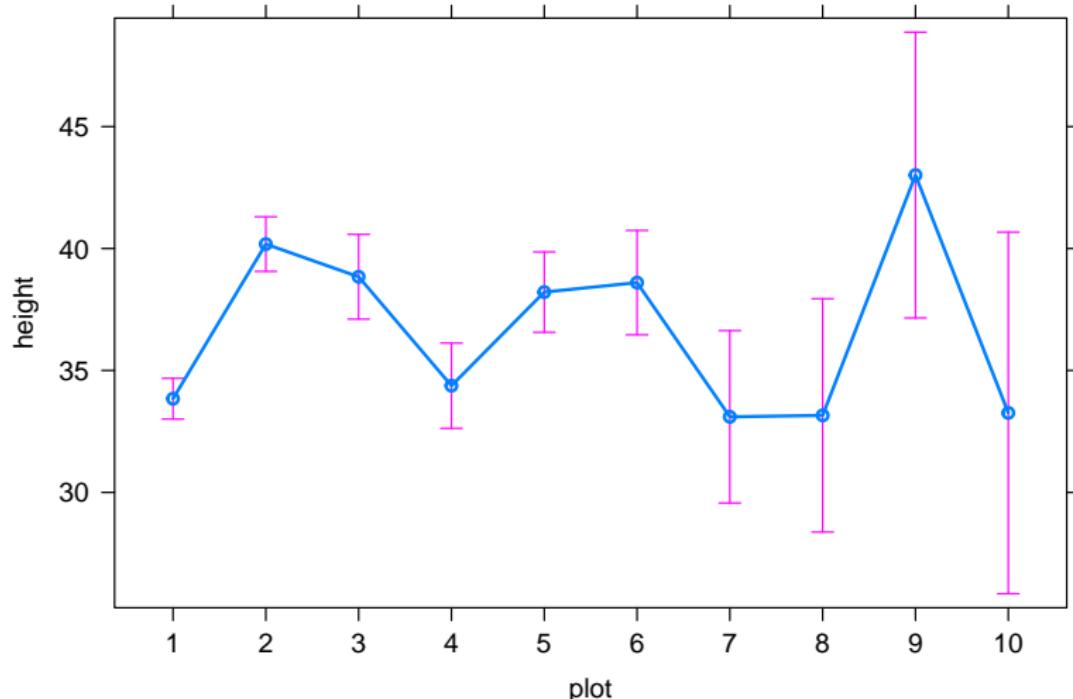
Lower 95 Percent Confidence Limits
plot
  1          2          3          4          5          6          7          8
33.00444 39.06264 37.10317 32.62733 36.56463 36.46190 29.56629 28.37367
  9          10
37.15251 25.84764

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

# Plot

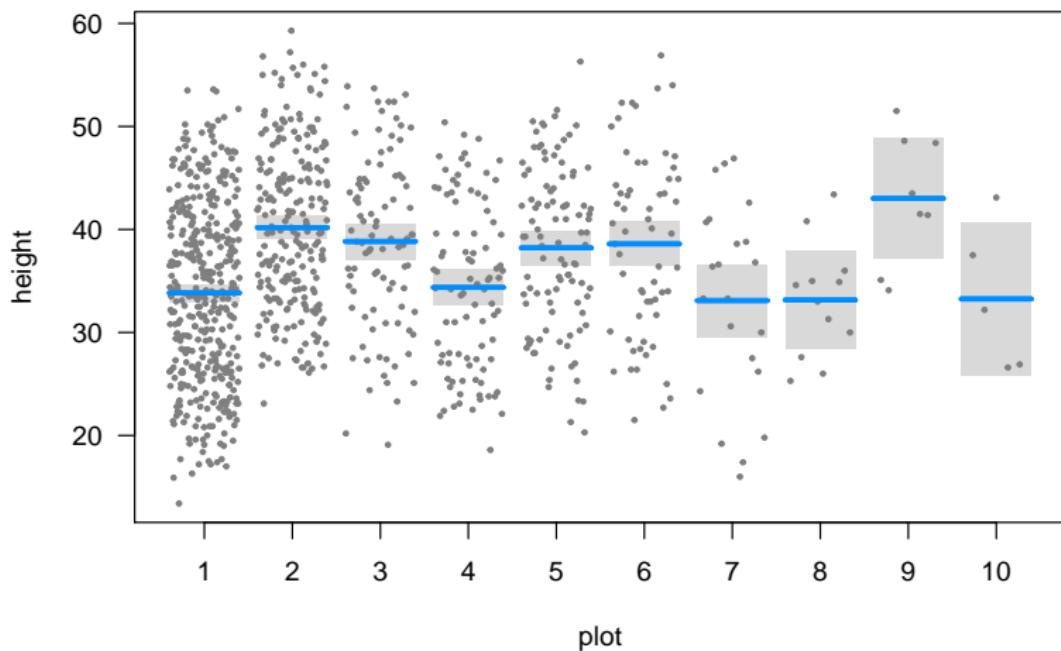
```
plot(allEffects(m3))
```

**plot effect plot**



# Plot (visreg)

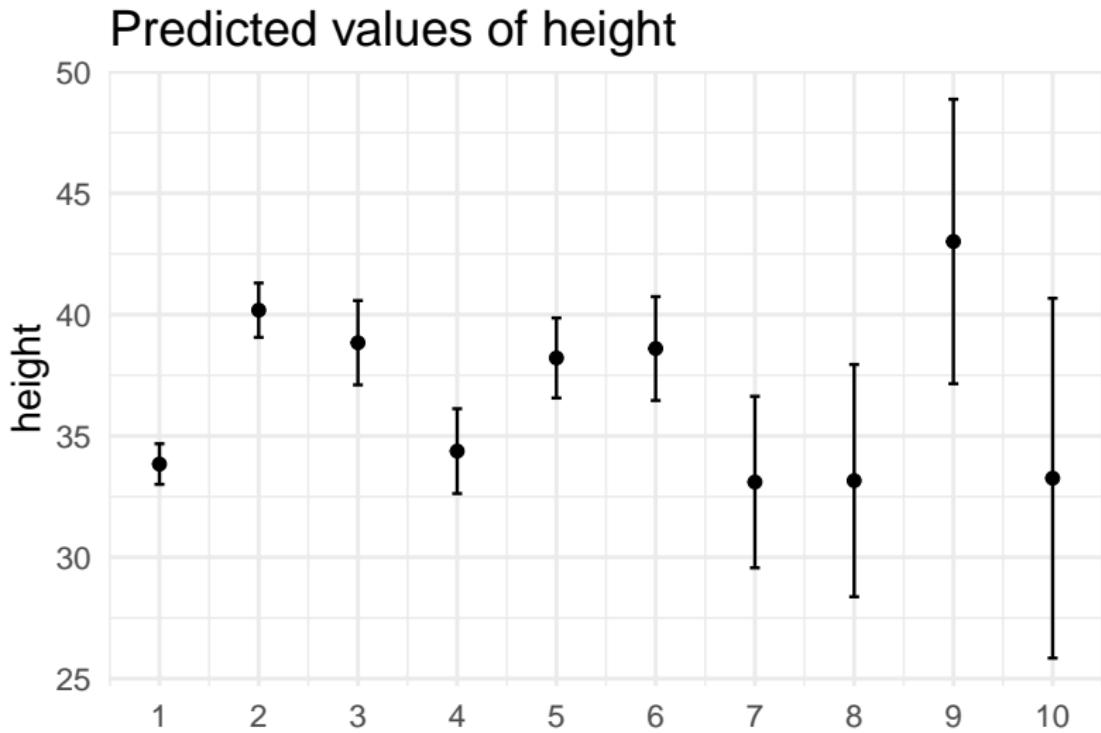
```
visreg(m3)
```



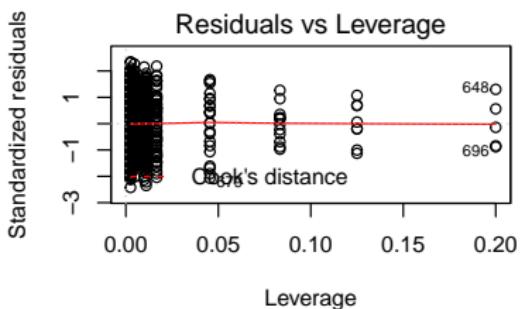
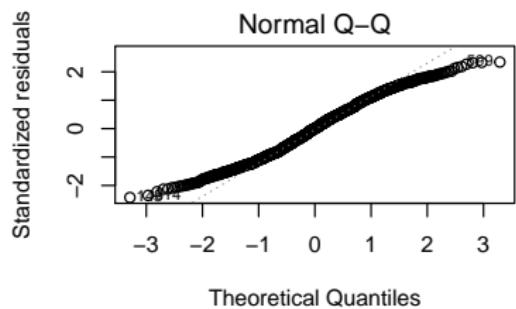
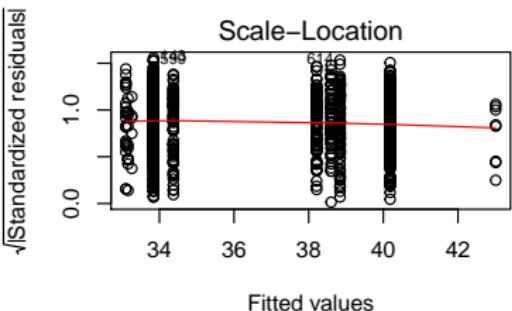
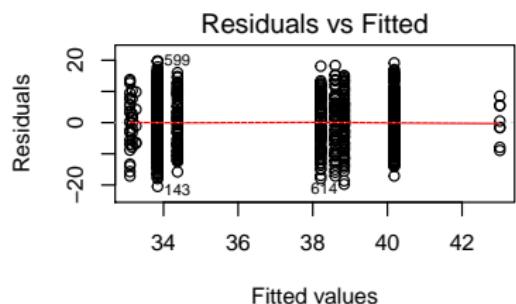
## Plot model (sjPlot)

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

```
$plot
```



# Model checking: residuals



## Combining continuous and categorical predictors

## Predicting tree height based on dbh and site

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

corresponds to

$$y_i = a + b_{plot2} + c_{plot3} + d_{plot4} + e_{plot5} + \dots + k \cdot DBH_i + \varepsilon_i$$
$$\varepsilon_i \sim N(0, \sigma^2)$$

# Predicting tree height based on dbh and site

Call:

```
lm(formula = height ~ plot + 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 ***
plot2	6.504303	0.256730	25.335	< 2e-16 ***
plot3	4.357457	0.354181	12.303	< 2e-16 ***
plot4	1.934650	0.356102	5.433	6.98e-08 ***
plot5	3.637432	0.339688	10.708	< 2e-16 ***
plot6	4.204511	0.421906	9.966	< 2e-16 ***
plot7	-0.176193	0.666772	-0.264	0.7916
plot8	-5.312648	0.893603	-5.945	3.82e-09 ***
plot9	5.437049	1.087766	4.998	6.84e-07 ***
plot10	2.263338	1.369986	1.652	0.0988 .
dbh	0.617075	0.007574	81.473	< 2e-16 ***

---

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

Residual standard error: 3.043 on 989 degrees of freedom

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

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

## Presenting model results

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

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	16.70	0.26	64.09	0.00
plot2	6.50	0.26	25.34	0.00
plot3	4.36	0.35	12.30	0.00
plot4	1.93	0.36	5.43	0.00
plot5	3.64	0.34	10.71	0.00
plot6	4.20	0.42	9.97	0.00
plot7	-0.18	0.67	-0.26	0.79
plot8	-5.31	0.89	-5.95	0.00
plot9	5.44	1.09	5.00	0.00
plot10	2.26	1.37	1.65	0.10
dbh	0.62	0.01	81.47	0.00

## Estimated tree heights for each site

```
summary(allEffects(m4))
```

model: height ~ plot + dbh

plot effect

plot

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

Lower 95 Percent Confidence Limits

plot

1	2	3	4	5	6	7	8
33.60276	40.00512	37.63569	35.20858	36.94739	37.33787	32.45495	26.86438
9	10						
37.22831	33.49623						

Upper 95 Percent Confidence Limits

plot

1	2	3	4	5	6	7	8
34.20599	40.81223	38.88798	36.46947	38.13622	38.87990	35.00141	30.31907
9	10						
41.45454	38.83919						

dbh effect

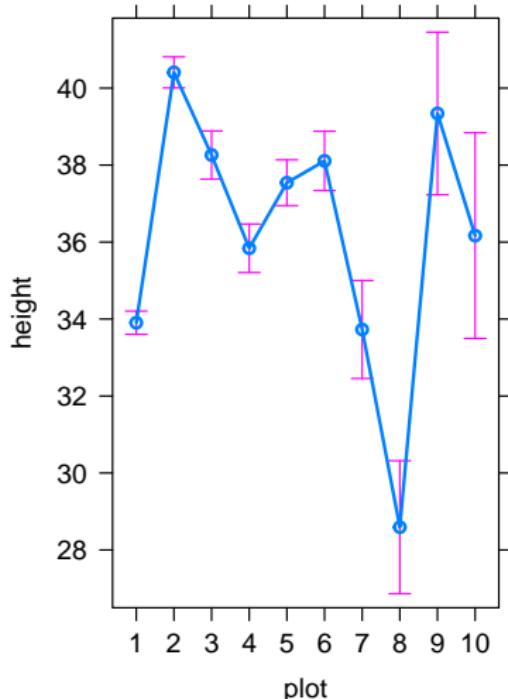
dbh

5	20	30	40	50
22.38634	31.64246	37.81321	43.98396	50.15471

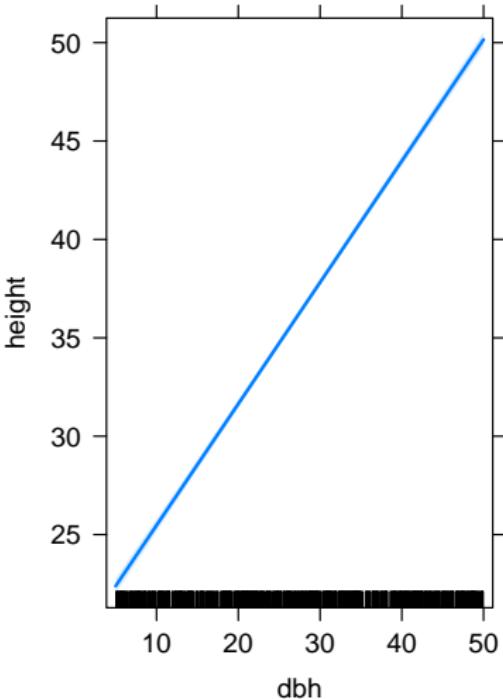
# Plot

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

**plot effect plot**

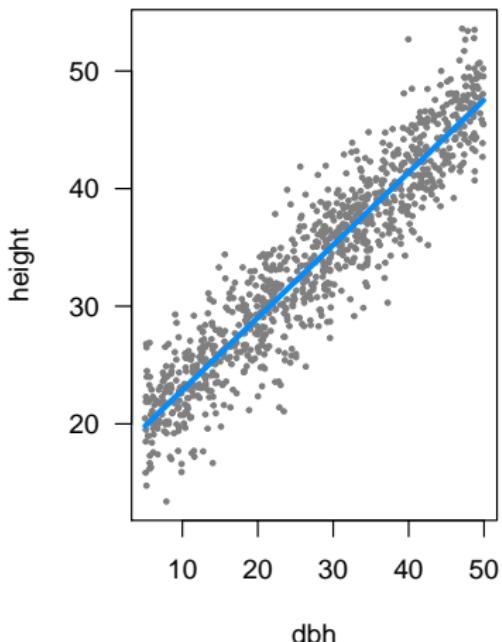
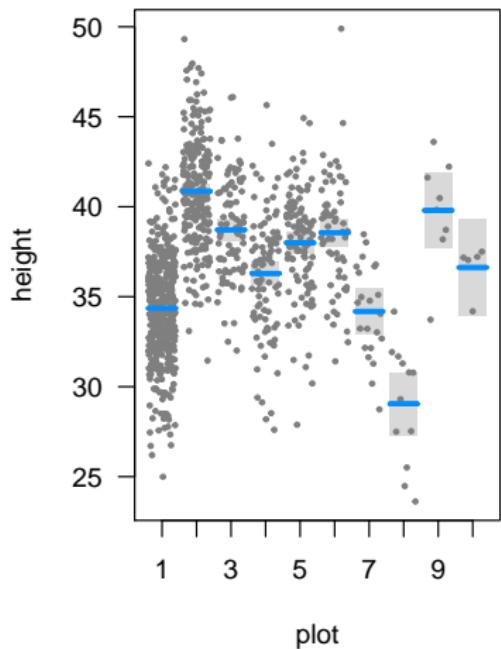


**dbh effect plot**



# Plot (visreg)

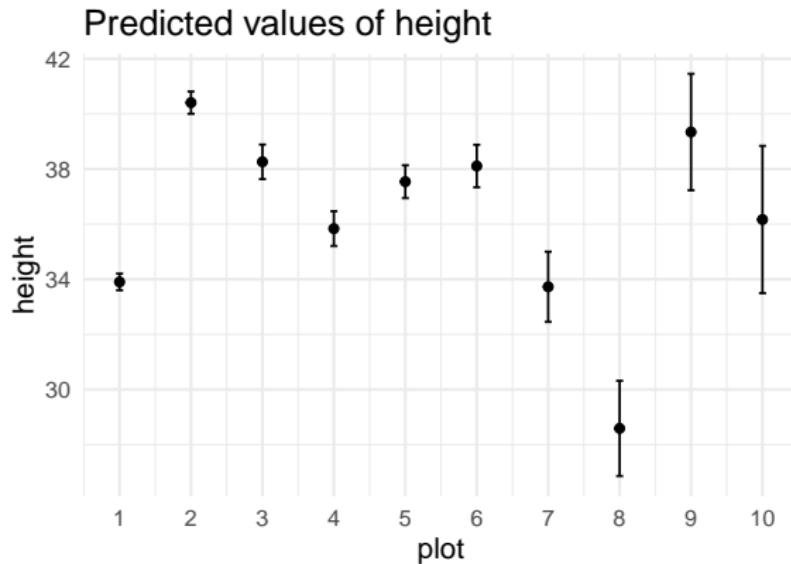
```
visreg(m4)
```



## Plot model (sjPlot)

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

```
$plot
```

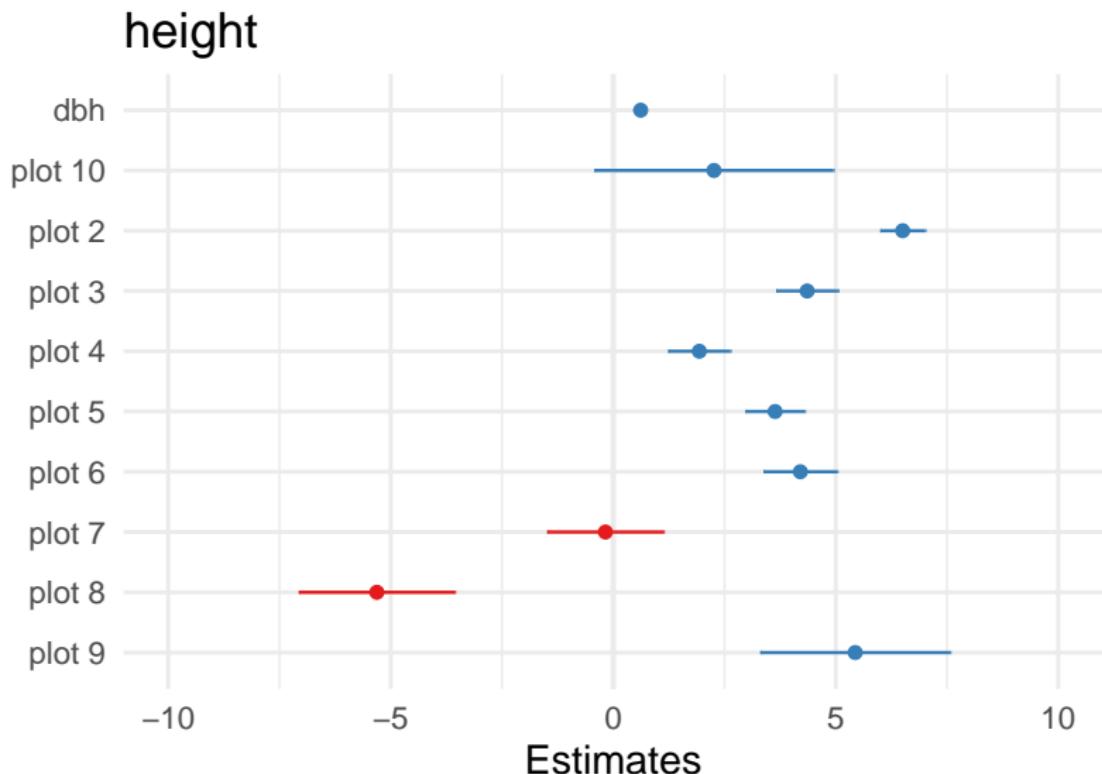


```
$dbh
```

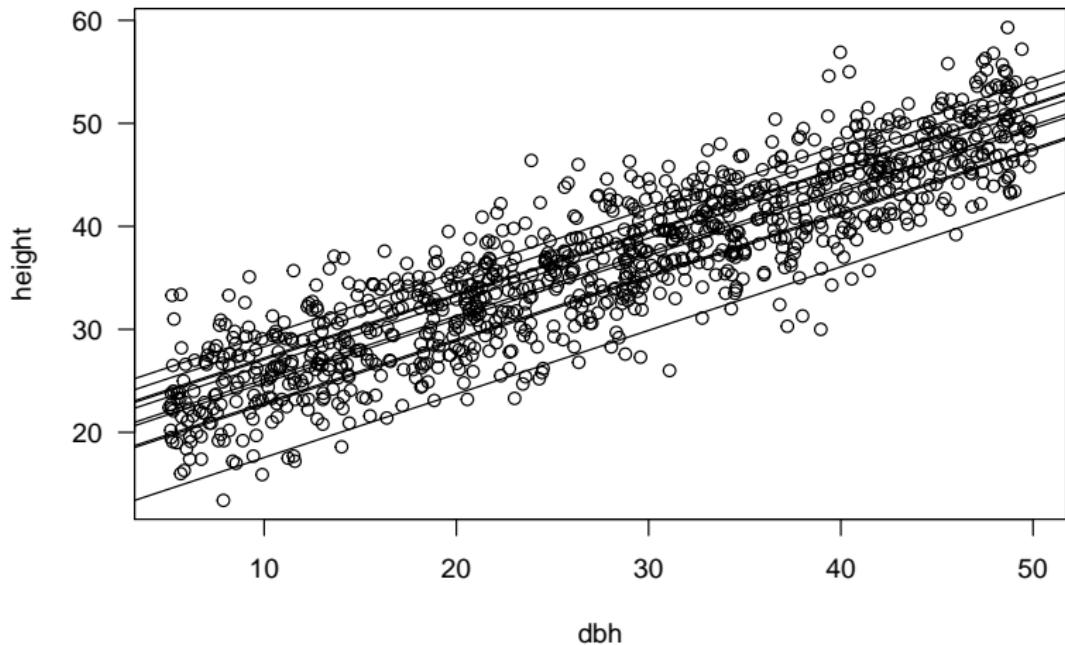
Predicted values of height

# Plot model (sjPlot)

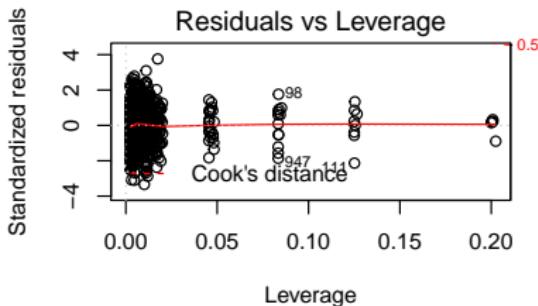
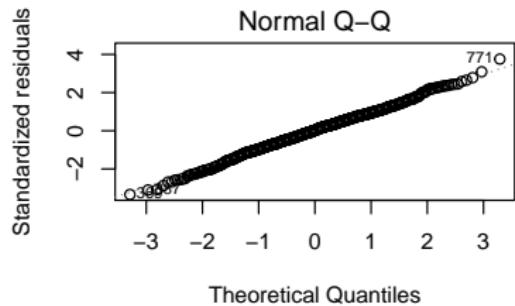
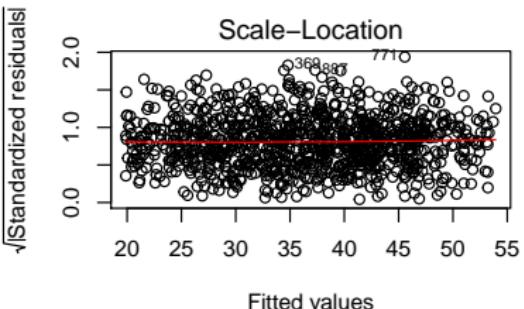
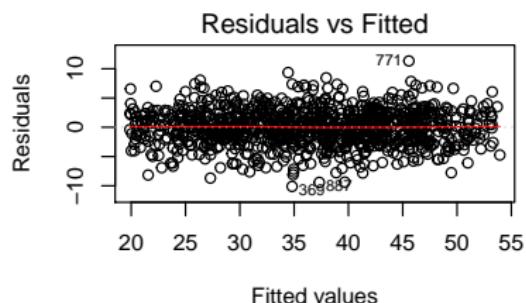
```
plot_model(m4, type = "est")
```



We have fitted model w/ many intercepts and single slope

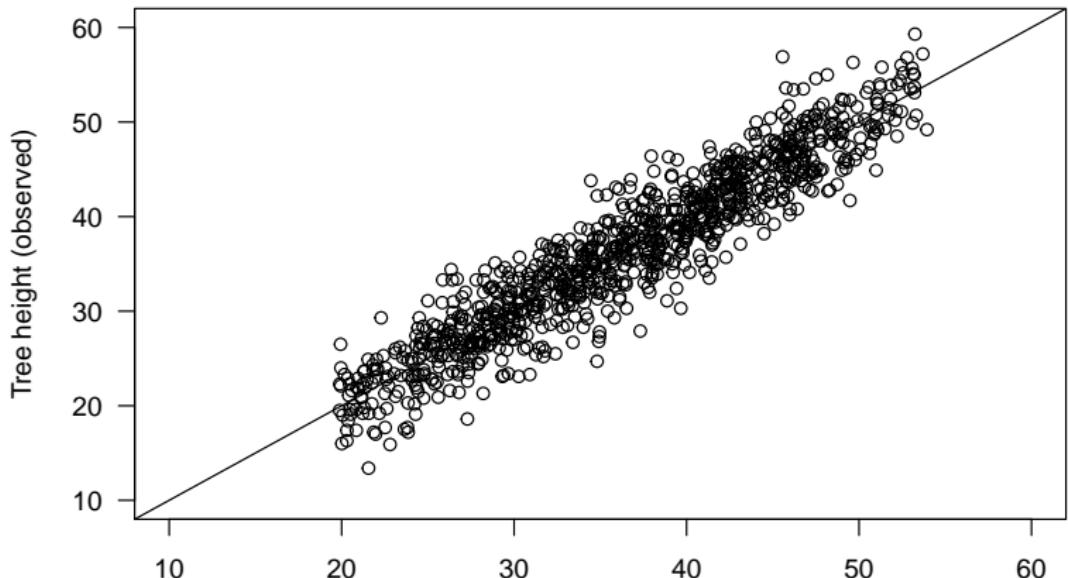


# Model checking: residuals



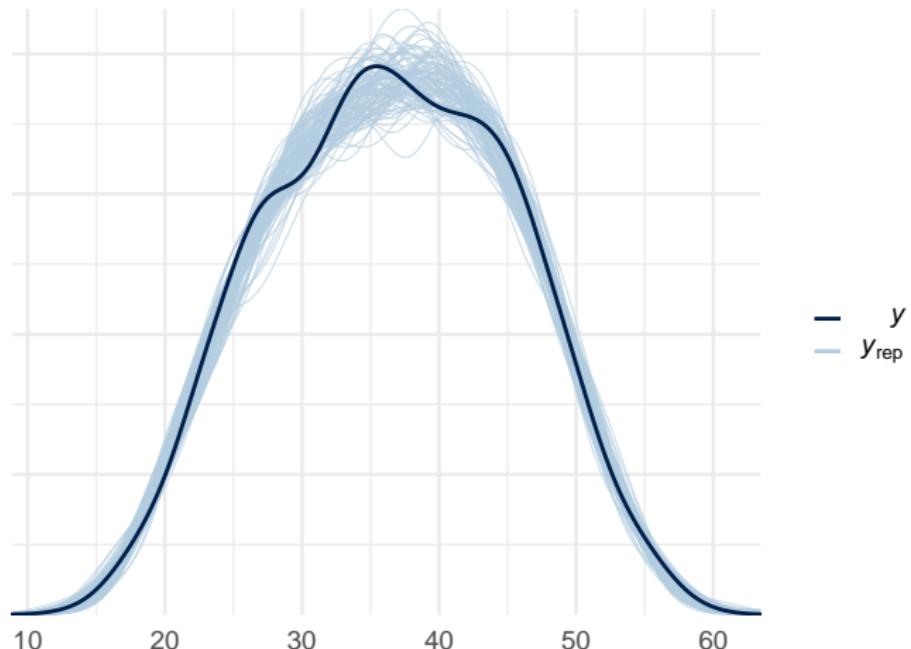
## How good is this model? Calibration plot

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



## Model checking with simulated data

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



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

# Model with interactions

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

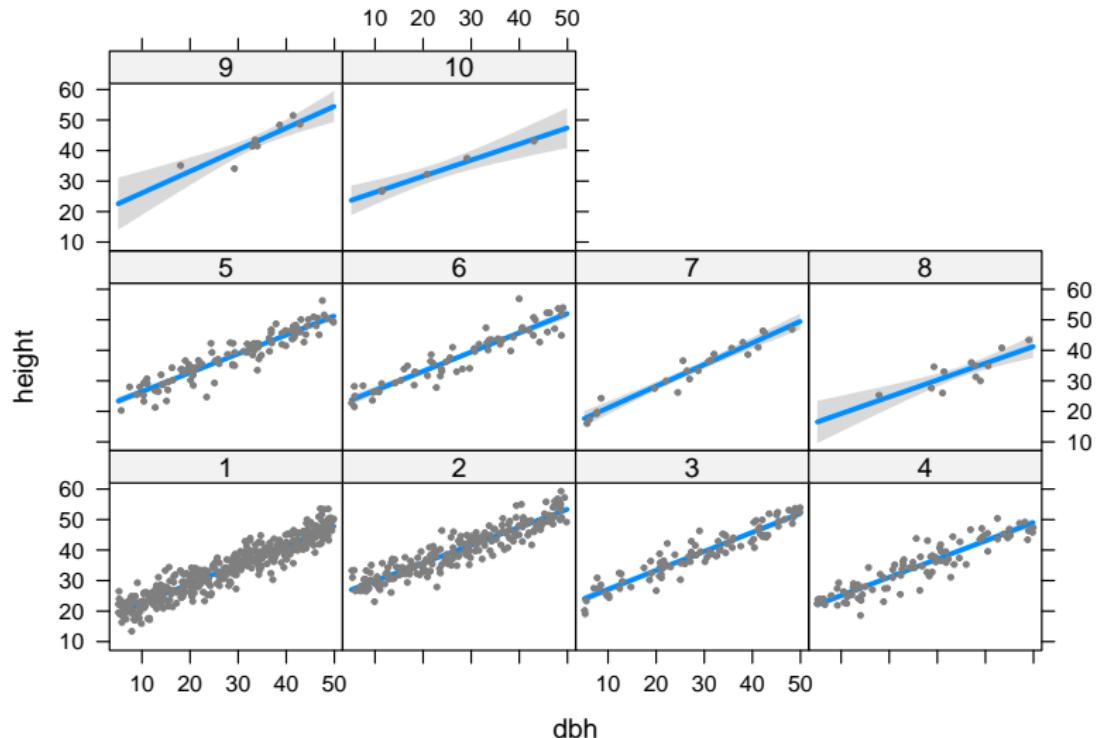
Residuals:
    Min      1Q  Median      3Q     Max 
-10.1017 -1.9839  0.0645  2.0486 11.1789 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 16.359437  0.360054 45.436 < 2e-16 ***
plot2        7.684781  0.609657 12.605 < 2e-16 ***
plot3        4.518568  0.867008  5.212 2.28e-07 ***
plot4        2.769336  0.813259  3.405 0.000688 ***
plot5        3.917607  0.870983  4.498 7.68e-06 ***
plot6        4.155161  1.009379  4.117 4.17e-05 ***
plot7       -2.306799  1.551303 -1.487 0.137334    
plot8       -2.616095  4.090671 -0.640 0.522630    
plot9        2.621560  5.073794  0.517 0.605492    
plot10       4.662340  2.991072  1.559 0.119378    
dbh          0.629299  0.011722 53.685 < 2e-16 ***
plot2:dbh   -0.042784  0.020033 -2.136 0.032950 *  
plot3:dbh   -0.006031  0.027640 -0.218 0.827312    
plot4:dbh   -0.031633  0.028225 -1.121 0.262677    
plot5:dbh   -0.010173  0.027887 -0.365 0.715334    
plot6:dbh   0.001337  0.032109  0.042 0.966797    
plot7:dbh   0.079728  0.052056  1.532 0.125951    
plot8:dbh   -0.079027  0.113386 -0.697 0.485984    
plot9:dbh   0.081035  0.146649  0.553 0.580679    
plot10:dbh  -0.101107  0.114520 -0.883 0.377522  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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

# Does slope vary among forests?

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



## Extra exercises

- ▶ paperplanes: How does flight distance differ with age, gender or paper type?

## Extra exercises

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

## Extra exercises

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

## Extra exercises

- ▶ paperplanes: How does flight distance differ with age, gender or paper type?
- ▶ mammal sleep: Are sleep patterns related to diet?
- ▶ iris: Predict petal length ~ petal width and species
- ▶ racing pigeons: is speed related to sex?

## Generalised Linear Models: Logistic regression

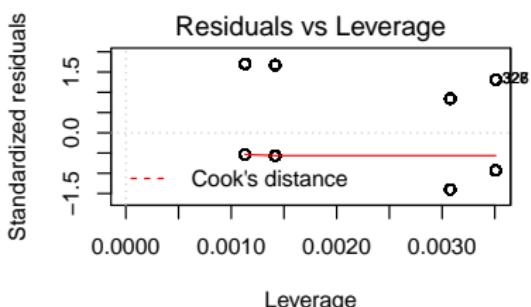
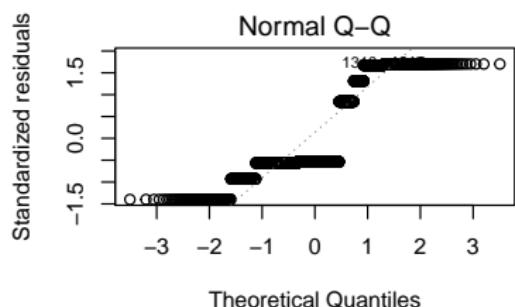
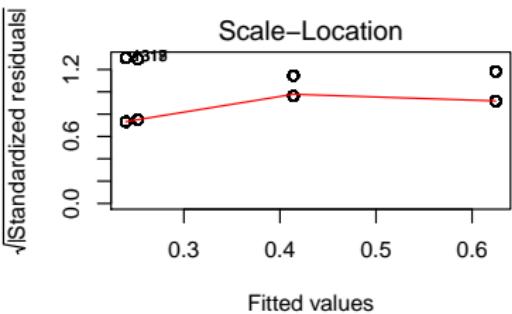
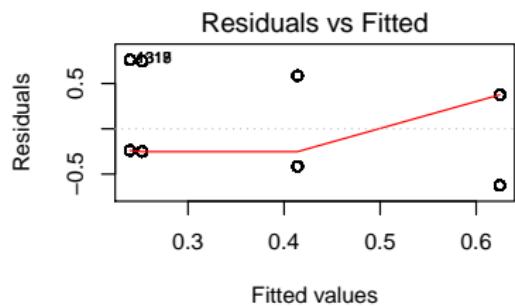
## Q: Survival of passengers on the Titanic ~ Class

Read titanic\_long.csv dataset.

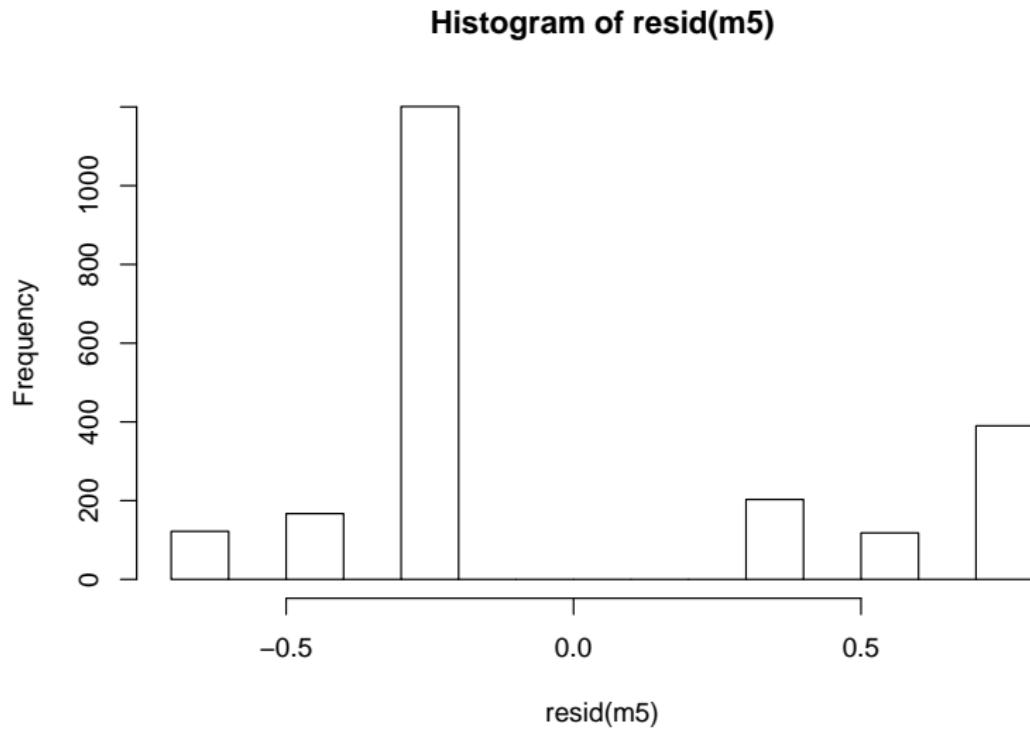
	class	age	sex	survived
1	first	adult	male	1
2	first	adult	male	1
3	first	adult	male	1
4	first	adult	male	1
5	first	adult	male	1
6	first	adult	male	1

## Let's fit linear model:

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



# Weird residuals!



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

- ▶ Binary variables (0/1)

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

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

# Generalised Linear Models

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

# Generalised Linear Models

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

# Generalised Linear Models

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

# Generalised Linear Models

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

- ▶ Bernouilli - Binomial
- ▶ Poisson
- ▶ Gamma

# Generalised Linear Models

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

- ▶ Bernouilli - Binomial
- ▶ Poisson
- ▶ Gamma
- ▶ etc

# Generalised Linear Models

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

# Generalised Linear Models

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

# Generalised Linear Models

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

# Generalised Linear Models

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

# Generalised Linear Models

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

- ▶ Bernouilli - Binomial
- ▶ Poisson
- ▶ Gamma
- ▶ etc

## 2. **Predictors** (continuous or categorical)

## 3. **Link function**

- ▶ Gaussian: identity
- ▶ Binomial: logit, probit
- ▶ Poisson: log...

# Generalised Linear Models

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

- ▶ Bernouilli - Binomial
- ▶ Poisson
- ▶ Gamma
- ▶ etc

## 2. **Predictors** (continuous or categorical)

## 3. **Link function**

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

# The modelling process

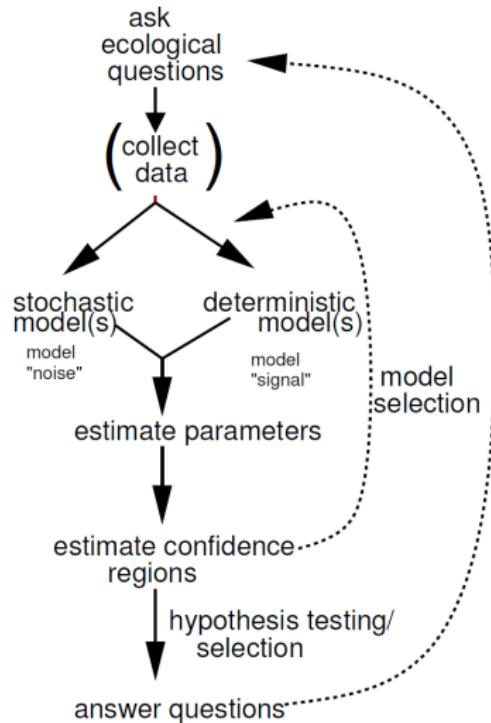


Figure 1.5 Flow of the modeling process.

Figure 5

## Bernoulli - Binomial distribution (Logistic regression)

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

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

Then

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

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

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

## Bernoulli - Binomial distribution (Logistic regression)

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

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

Then

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

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

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

## Back to survival of Titanic passengers

How many survived in each class?

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

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

## Back to survival of Titanic passengers (dplyr)

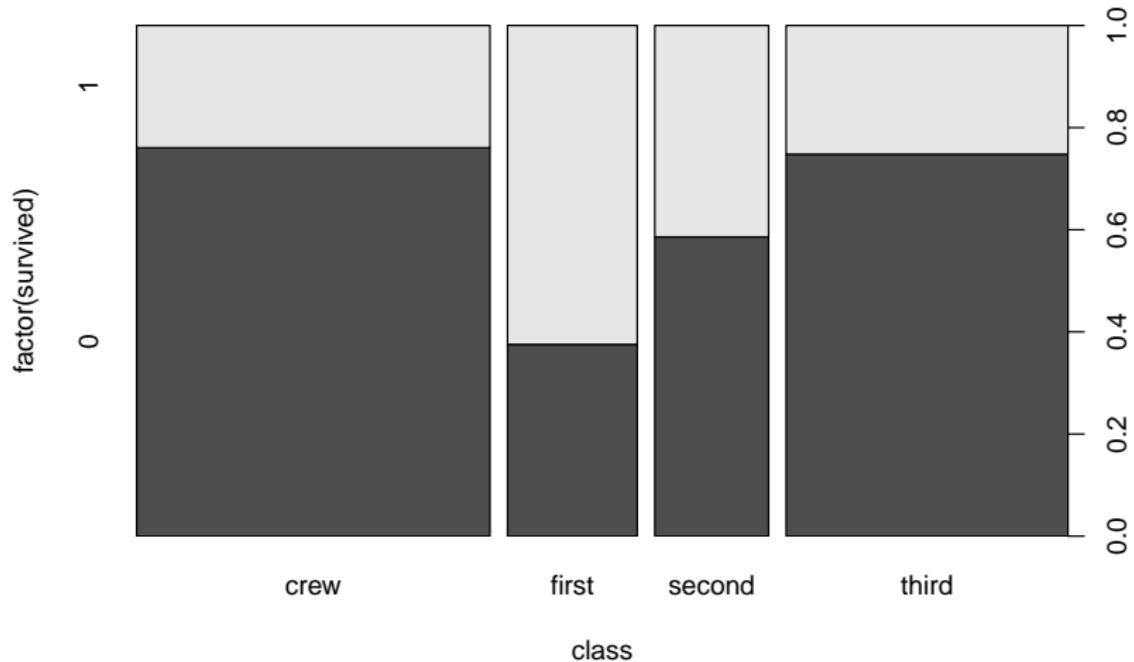
Passenger survival according to class

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

```
# A tibble: 8 x 3
# Groups:   class [4]
  class  survived  count
  <fct>    <int> <int>
1 crew        0    673
2 crew        1    212
3 first       0    122
4 first       1    203
5 second      0    167
6 second      1    118
7 third       0    528
8 third       1    178
```

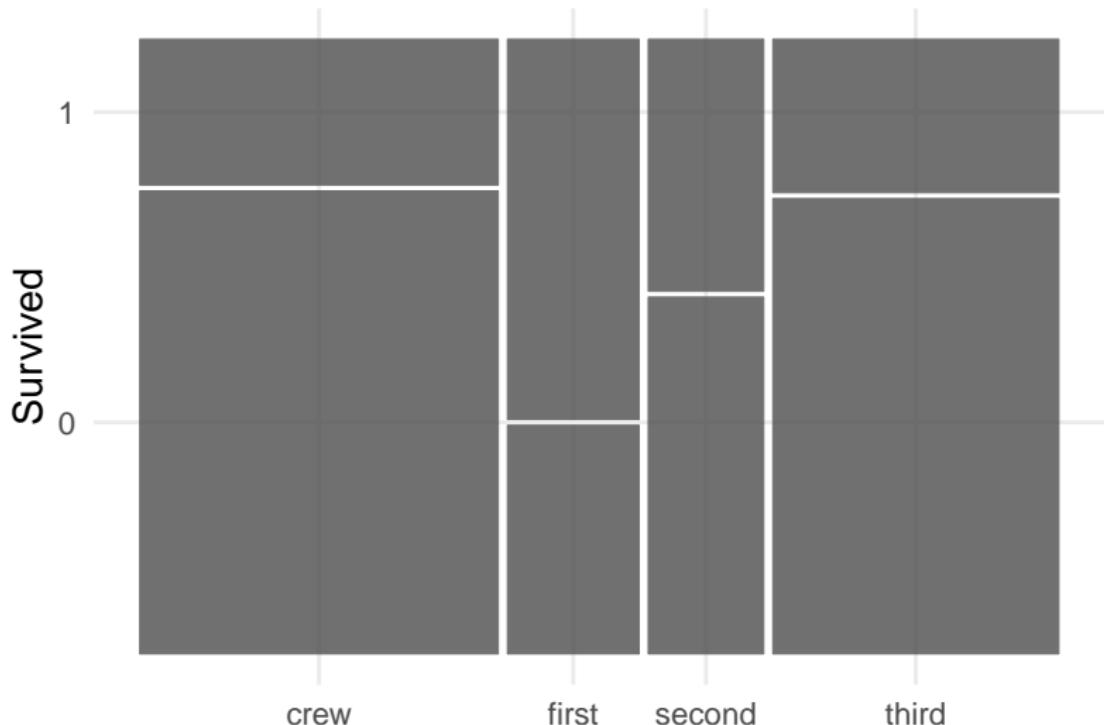
Or graphically...

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



## Mosaic plots (ggplot2)

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



## Fitting GLMs in R: `glm`

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

which corresponds to

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

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

# Fitting GLMs in R: `glm`

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

Call:

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

Deviance Residuals:

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

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-1.15516	0.07876	-14.667	< 2e-16 ***
classfirst	1.66434	0.13902	11.972	< 2e-16 ***
classsecond	0.80785	0.14375	5.620	1.91e-08 ***
classthird	0.06785	0.11711	0.579	0.562

---

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

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2769.5 on 2200 degrees of freedom

Residual deviance: 2588.6 on 2197 degrees of freedom

AIC: 2596.6

Number of Fisher Scoring iterations: 4

**These estimates are in logit scale!**

# Interpreting logistic regression output

Parameter estimates (logit-scale)

	(Intercept)	classfirst	classsecond	classthird
	-1.15515905	1.66434399	0.80784987	0.06784632

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

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

(Intercept)  
0.239548

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

```
[1] 0.239548
```

## Q: Probability of survival for 1st class passengers?

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

(Intercept)

0.6246154

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

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

[1] 0.6246154

## Model interpretation using effects package

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

```
model: survived ~ class
```

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

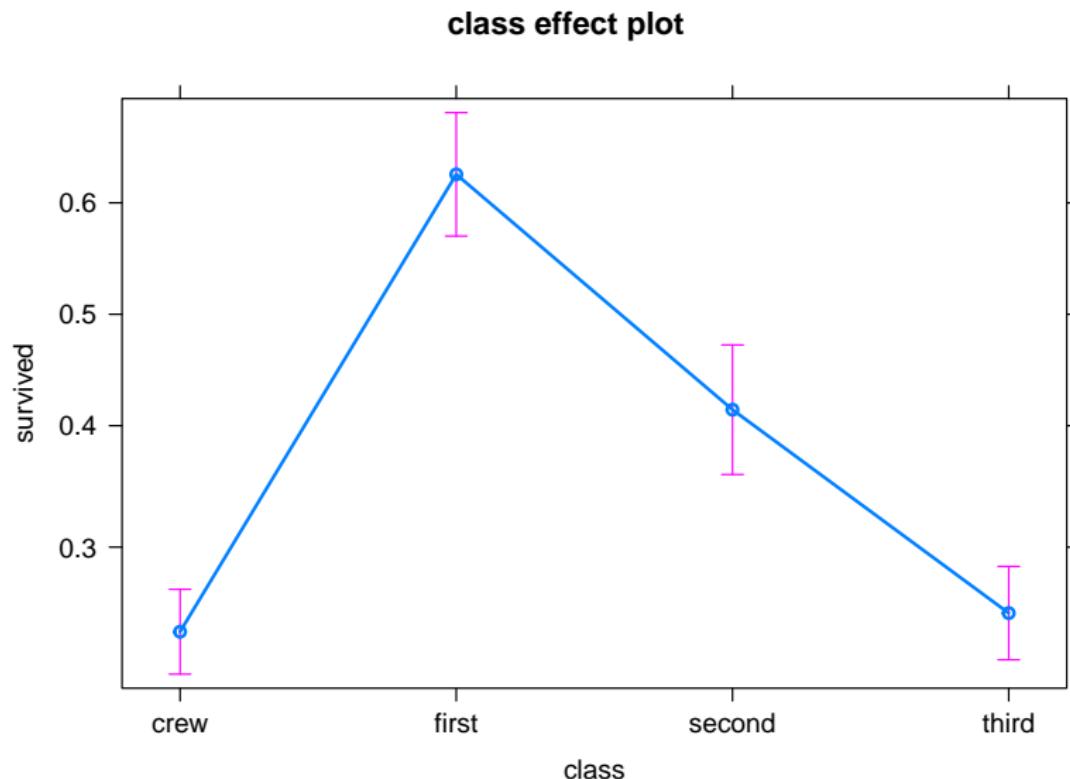
## Presenting model results

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

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

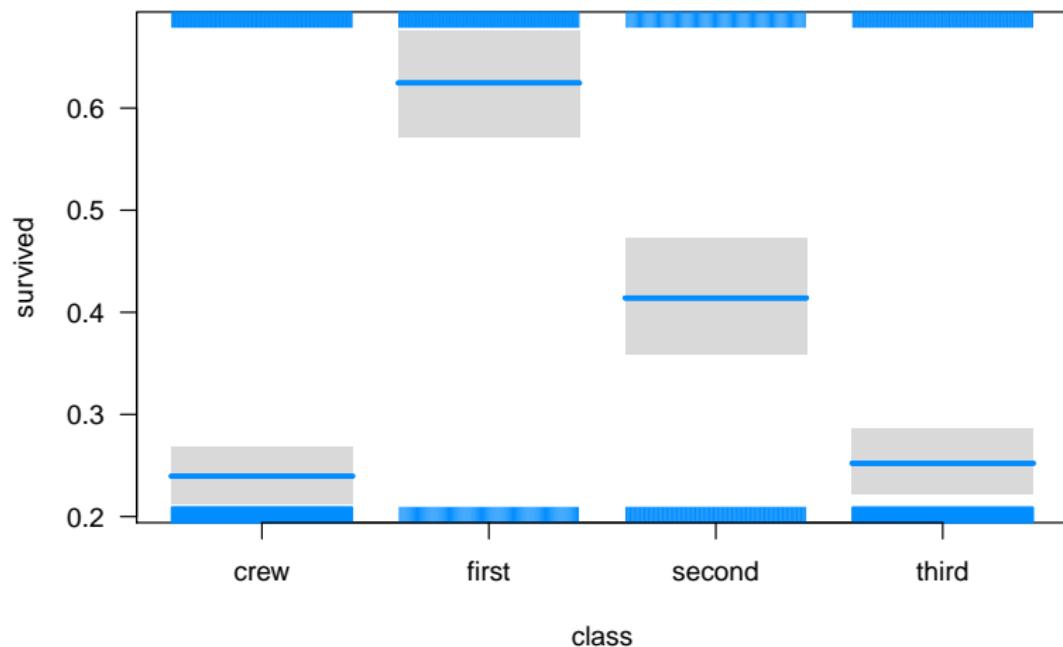
# Visualising model: effects package

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



# Visualising model: visreg package

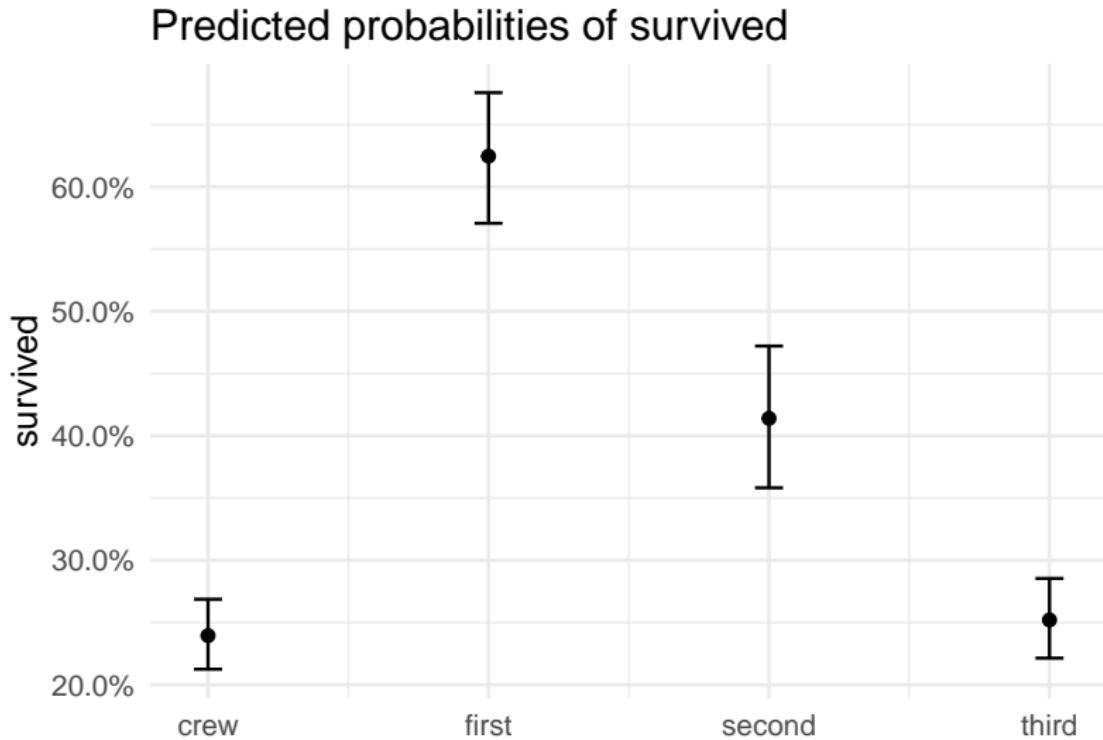
```
visreg(tit.glm, scale = "response")
```



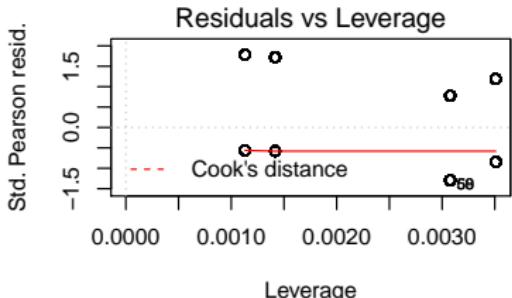
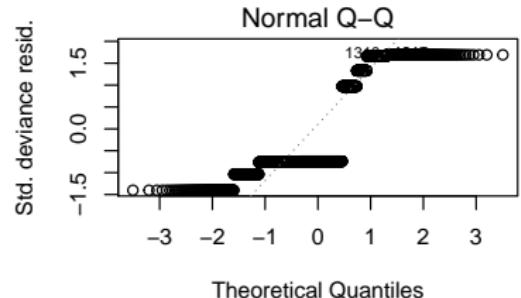
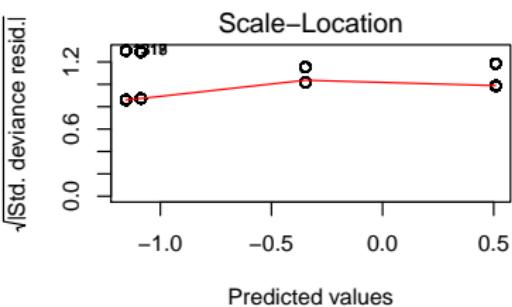
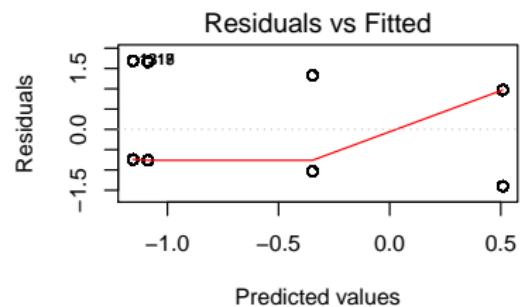
## Visualising model: sjPlot package

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

```
$class
```



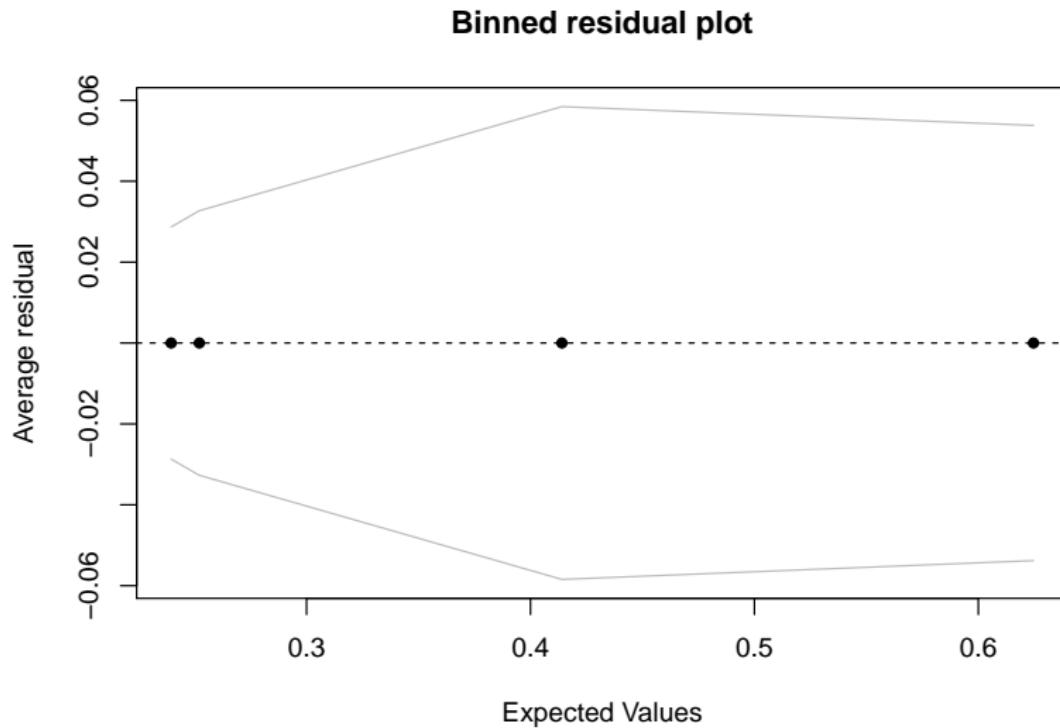
# Logistic regression: model checking



null device

## Binned residual plots for logistic regression

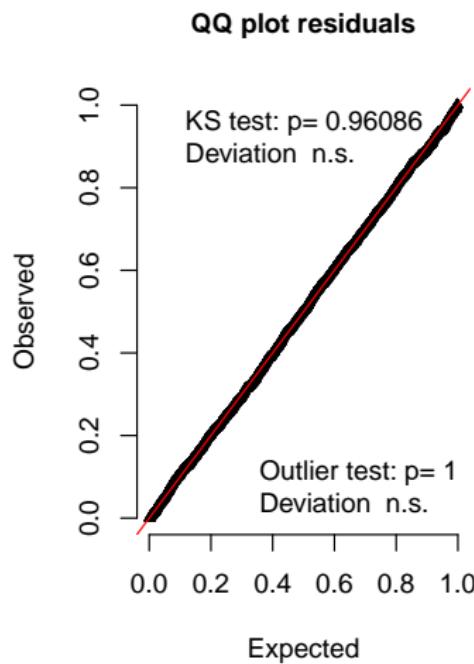
```
predvals <- predict(tit.glm, type="response")
arm::binnedplot(predvals, titanic$survived - predvals)
```



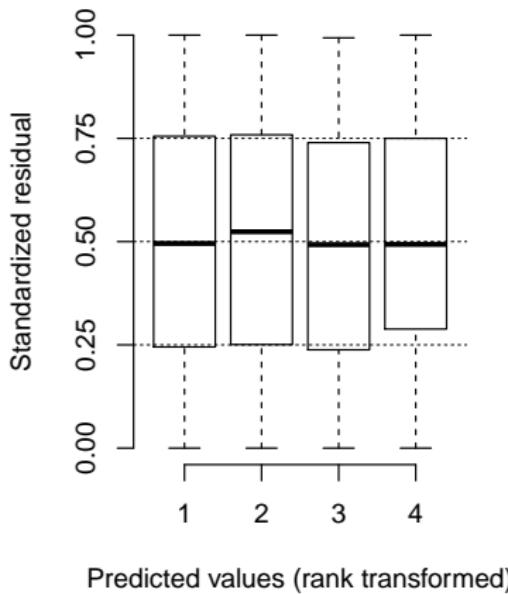
# Residual diagnostics with DHARMA

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

DHARMA scaled residual plots

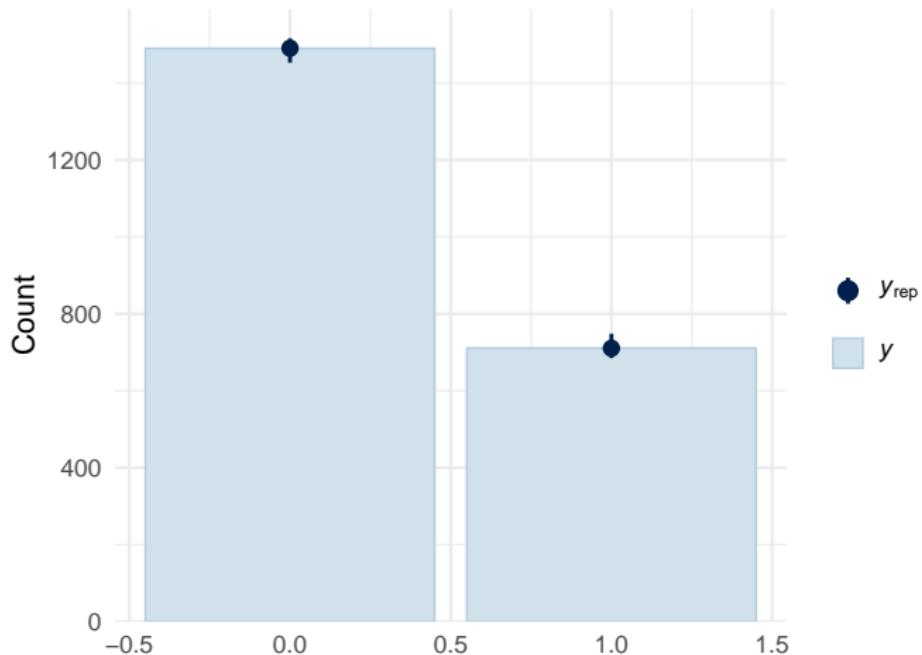


Residual vs. predicted  
lines should match



## Model checking with simulated data

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



## Pseudo R-squared for GLMs

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

R-Squared for (Generalized) Linear (Mixed) Model

Cox & Snell's R-squared: 0.079

Nagelkerke's R-squared: 0.110

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

# Recapitulating

## **1. Visualise data**

## Recapitulating

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

# Recapitulating

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

# Recapitulating

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

# Recapitulating

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

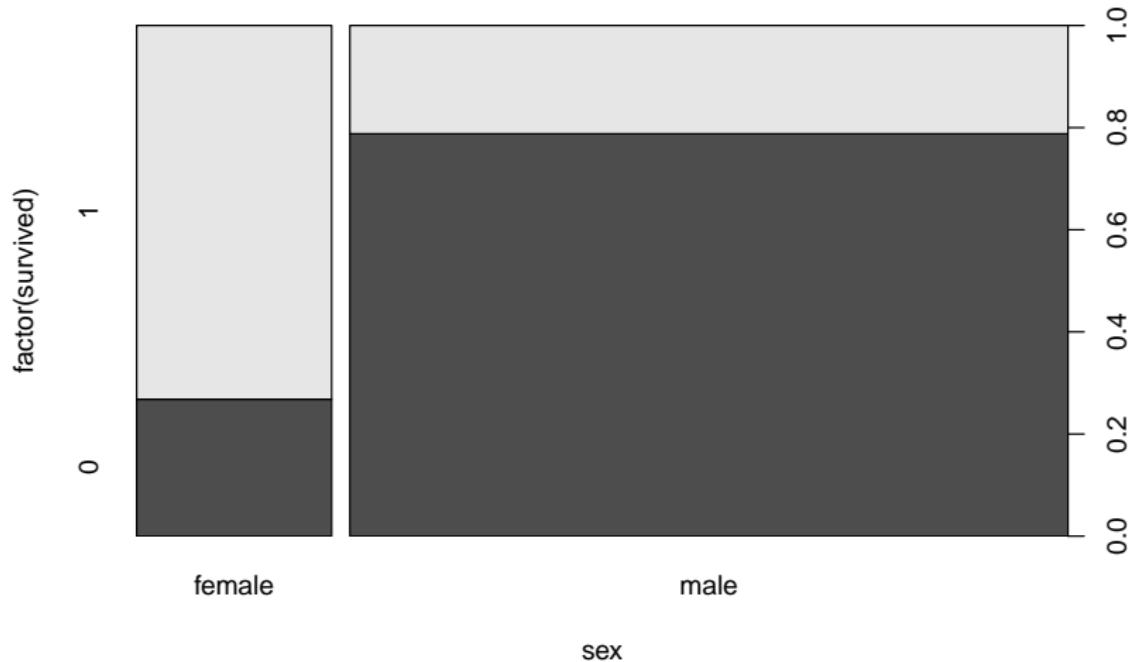
# Recapitulating

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

Q: Did men have higher survival than women?

## Plot first

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



## Fit model

Call:

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

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.6226	-0.6903	-0.6903	0.7901	1.7613

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	1.0044	0.1041	9.645	<2e-16 ***
sexmale	-2.3172	0.1196	-19.376	<2e-16 ***

---

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

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2769.5 on 2200 degrees of freedom  
Residual deviance: 2335.0 on 2199 degrees of freedom  
AIC: 2339

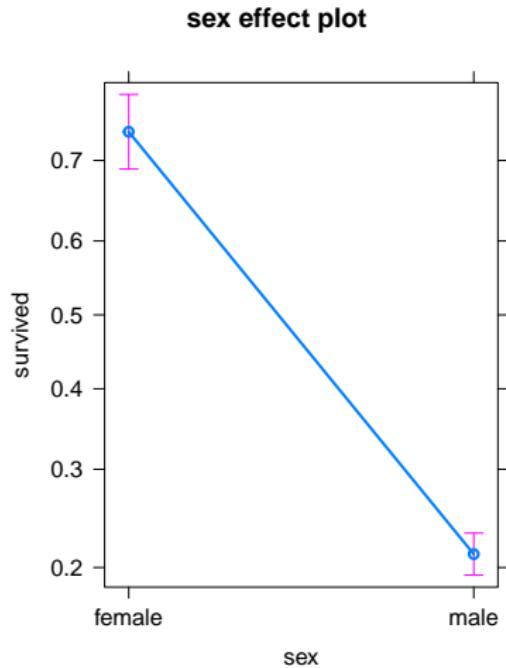
# Effects

```
model: survived ~ sex
```

```
sex effect
```

```
sex
```

female	male
0.7319149	0.2120162



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

## Let's look at the data

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

```
, ,  = female
```

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

```
, ,  = male
```

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

Mmm...

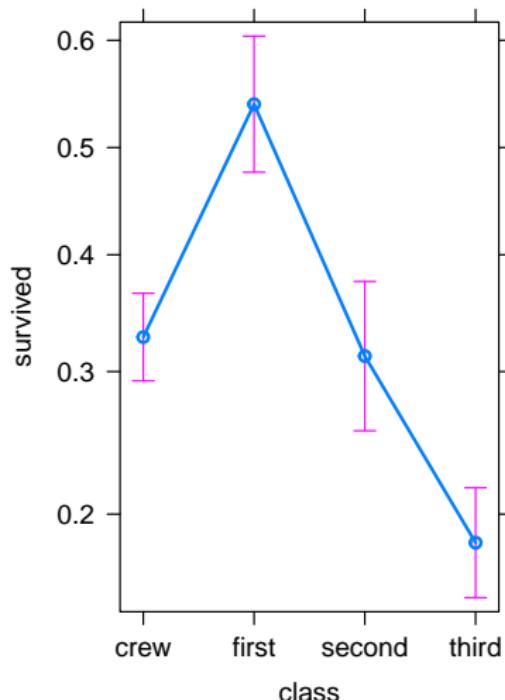
## Fit additive model with both factors

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

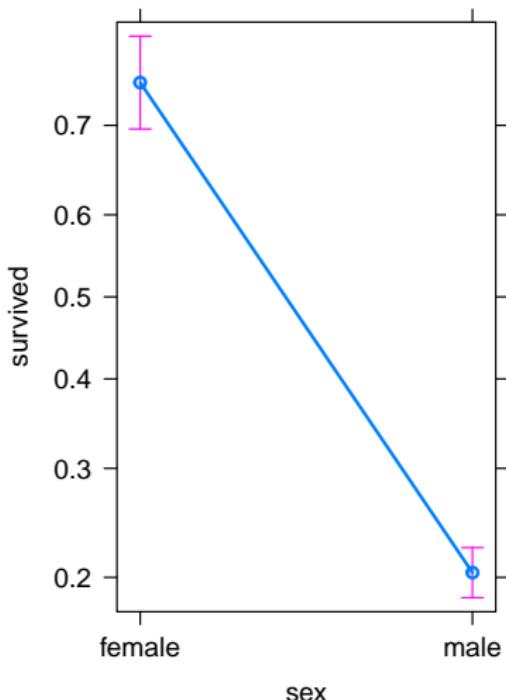
# Plot additive model

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

**class effect plot**



**sex effect plot**



## Fit model with both factors (interactions)

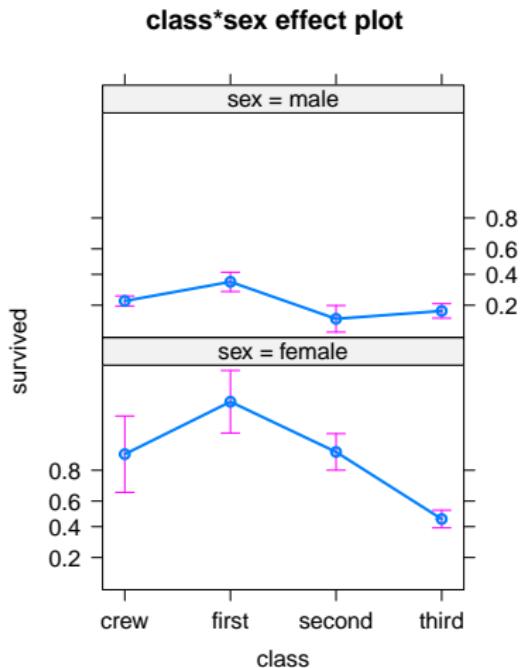
```
tit.sex.class <- glm(survived ~ class * sex, family = binomial,  
  
glm(formula = survived ~ class * sex, family = binomial, data =  
      coef.est coef.se  
(Intercept) 1.90 0.62  
classfirst 1.67 0.80  
classsecond 0.07 0.69  
classthird -2.06 0.64  
sexmale -3.15 0.62  
classfirst:sexmale -1.06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

# Effects

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

class\*sex effect

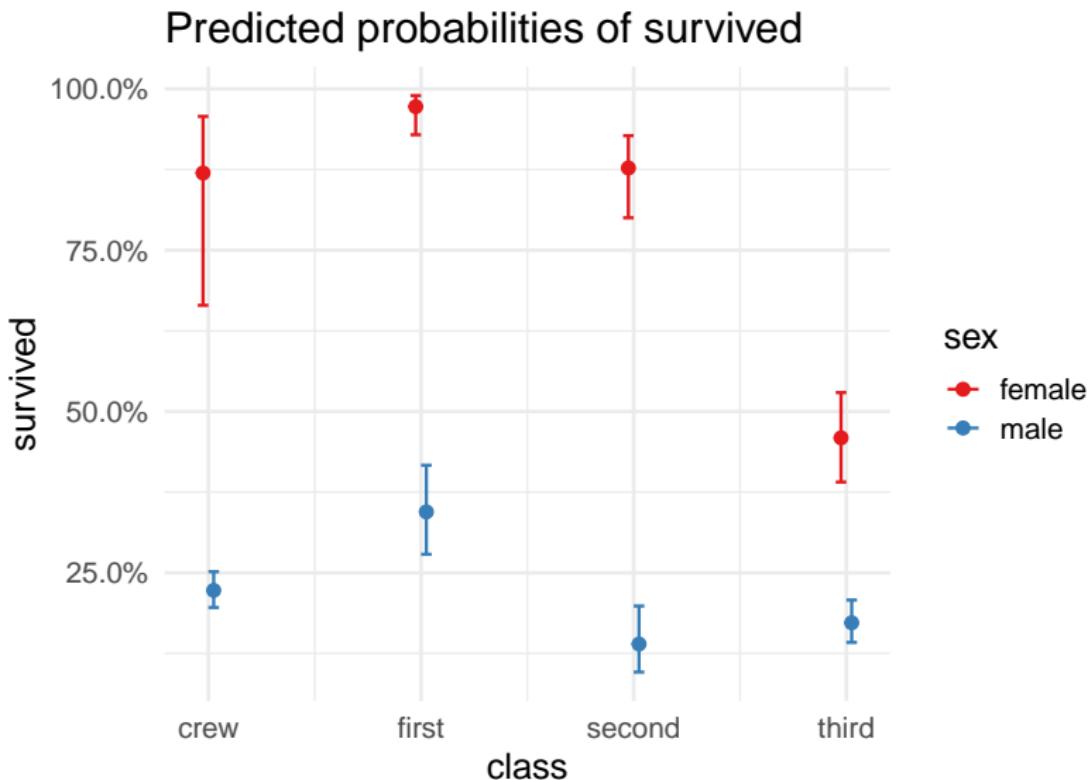
class	sex	
	female	male
crew	0.8695652	0.2227378
first	0.9724138	0.3444444
second	0.8773585	0.1396648
third	0.4591837	0.1725490



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

## Effects (sjPlot)

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



Logistic regression for proportion data

## Read Titanic data in different format

Read Titanic\_prop.csv data.

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

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

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

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

Call:

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

Deviance Residuals:

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

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	0.5092	0.1146	4.445	8.79e-06 ***
Class2nd	-0.8565	0.1661	-5.157	2.51e-07 ***
Class3rd	-1.5965	0.1436	-11.114	< 2e-16 ***
ClassCrew	-1.6643	0.1390	-11.972	< 2e-16 ***
---				

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

(Dispersion parameter for binomial family taken to be 1)

## Effects

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

Class effect

Class

1st	2nd	3rd	Crew
-----	-----	-----	------

0.6246154	0.4140351	0.2521246	0.2395480
-----------	-----------	-----------	-----------

Compare with former model based on raw data:

```
model: survived ~ class
```

class effect

class

crew	first	second	third
------	-------	--------	-------

0.2395480	0.6246154	0.4140351	0.2521246
-----------	-----------	-----------	-----------

Same results!

Logistic regression with continuous predictors

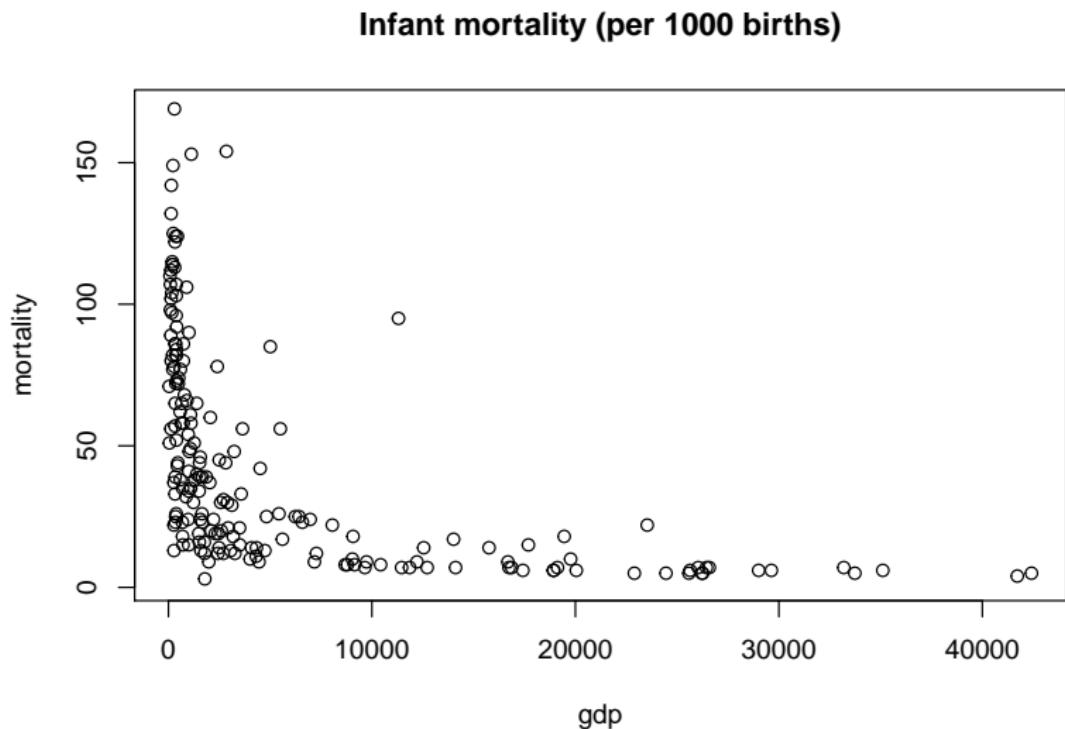
Example dataset: GDP and infant mortality

Read UN\_GDP\_infantmortality.csv.

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

# EDA

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



## Fit model

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

Call:

```
glm(formula = cbind(mortality, 1000 - mortality) ~ gdp, family =  
     data = gdp)
```

Deviance Residuals:

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

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )	
(Intercept)	-2.657e+00	1.311e-02	-202.76	<2e-16	***
gdp	-1.279e-04	3.458e-06	-36.98	<2e-16	***
---					

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

(Dispersion parameter for binomial family taken to be 1)

## Effects

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

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

gdp effect

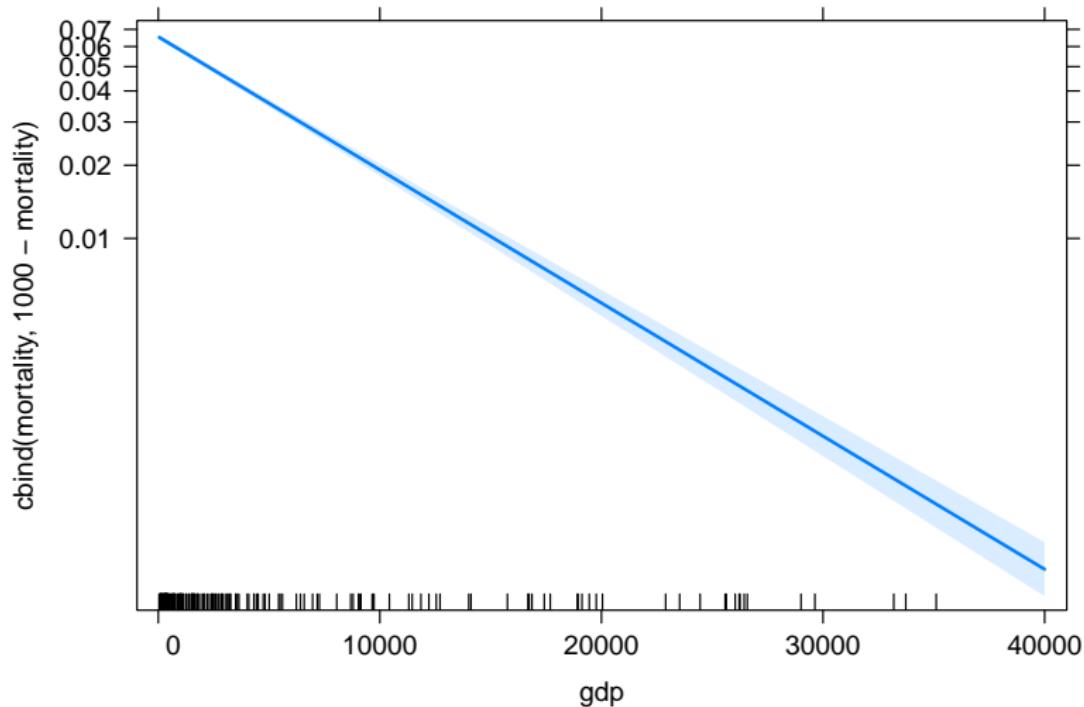
gdp

	40	10000	20000	30000	40000
0.0652177296	0.0191438829	0.0054028095	0.0015096074	0.0004206154	

## Effects plot

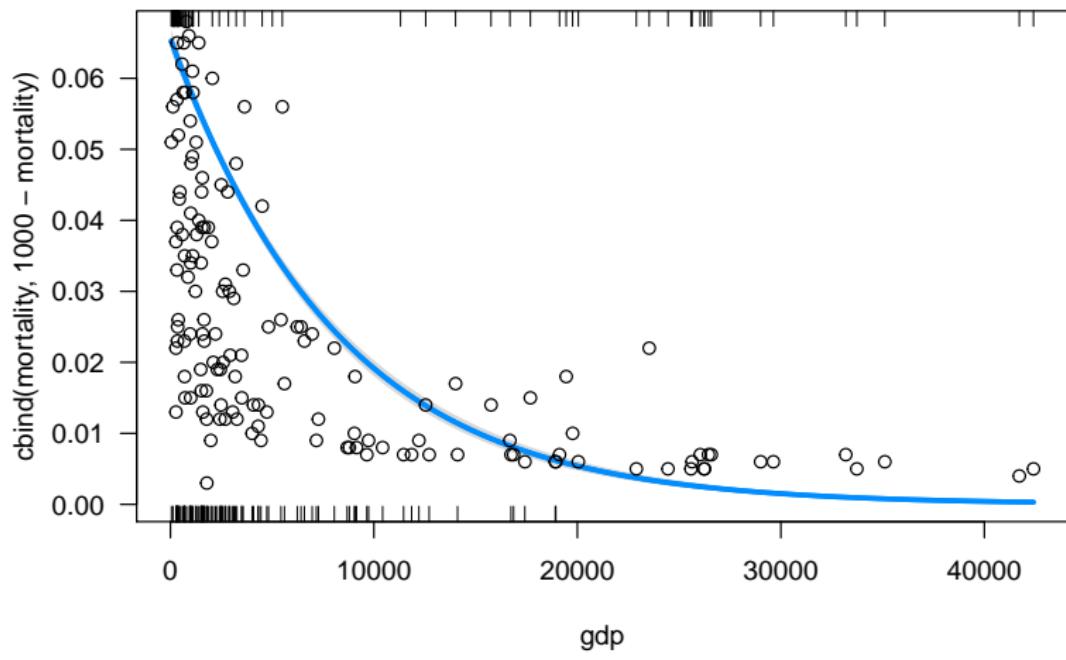
```
plot(allEffects(gdp.glm))
```

**gdp effect plot**



## Plot model using visreg:

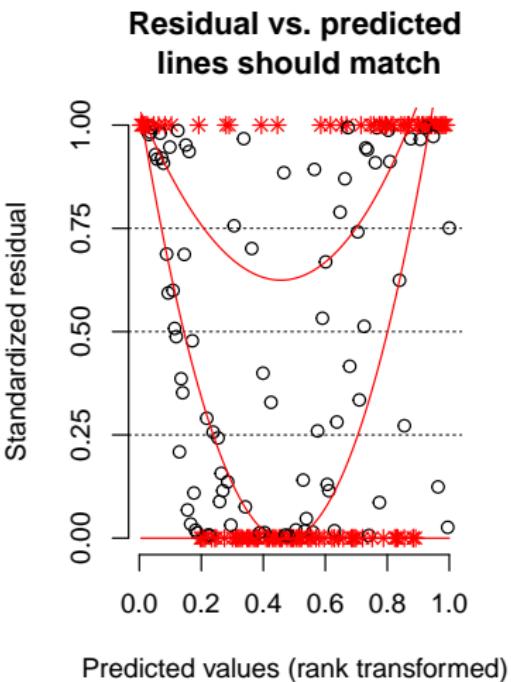
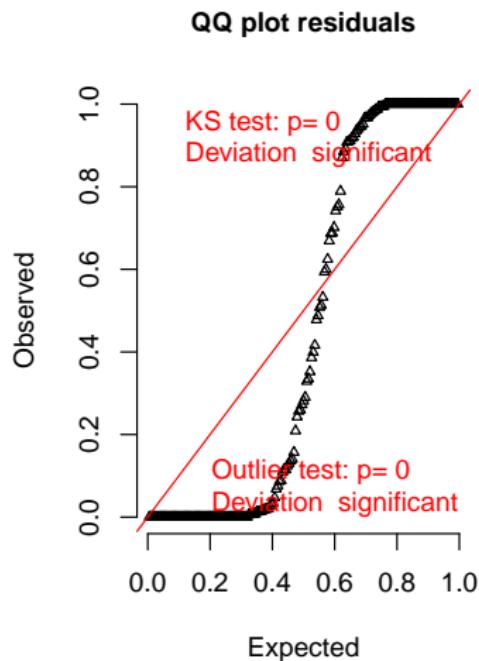
```
visreg(gdp.glm, scale = "response")
points(mortality/1000 ~ gdp, data = gdp)
```



# Residuals diagnostics with DHARMA

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

DHARMA scaled residual plots



## Overdispersion

## Testing for overdispersion (DHARMA)

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

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

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

## Overdispersion in logistic regression with proportion data

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

Call:

```
glm(formula = cbind(mortality, 1000 - mortality) ~ gdp, family =  
     data = gdp)
```

Deviance Residuals:

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

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-2.657e+00	5.977e-02	-44.465	< 2e-16 ***
gdp	-1.279e-04	1.577e-05	-8.111	5.96e-14 ***
---				

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

(Dispersion parameter for quasibinomial family taken to be 20.79)

## Mean estimates do not change after accounting for overdispersion

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

gdp effect

gdp

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

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

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

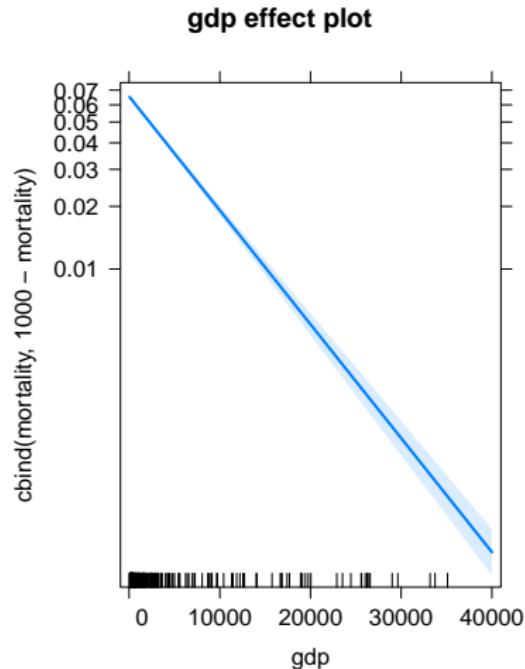
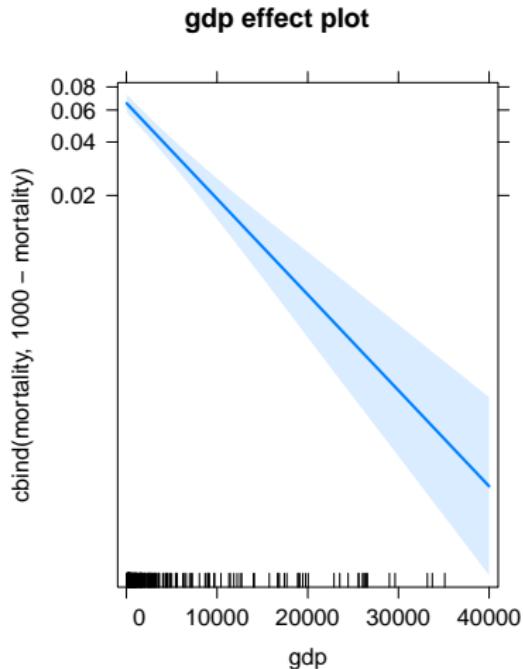
gdp effect

gdp

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

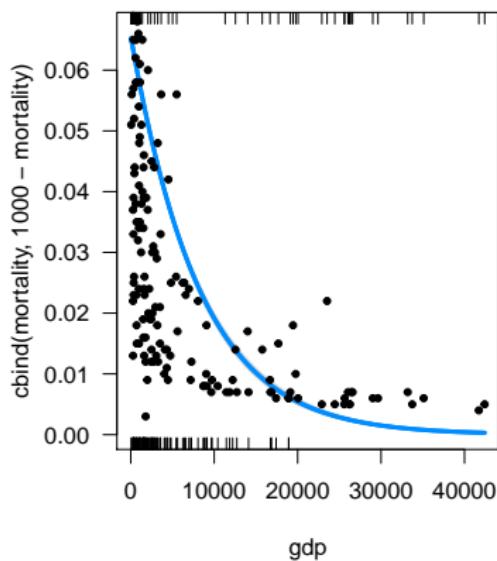
0.0652177296	0.0191438829	0.0054028095	0.0015096074	0.0004206154
--------------	--------------	--------------	--------------	--------------

# But standard errors (uncertainty) do!

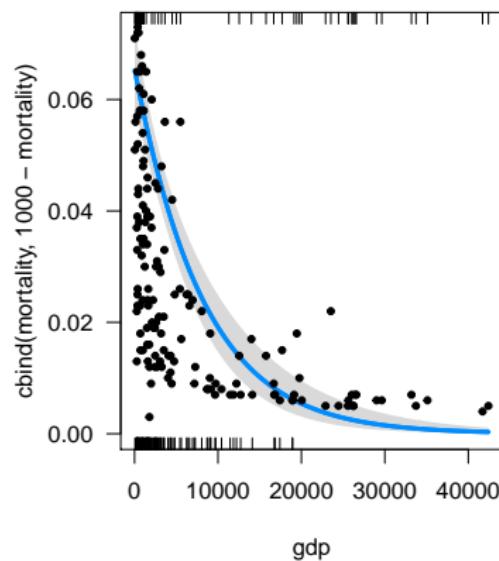


# Plot model and data

**Binomial**



**Quasibinomial**



## Overdispersion

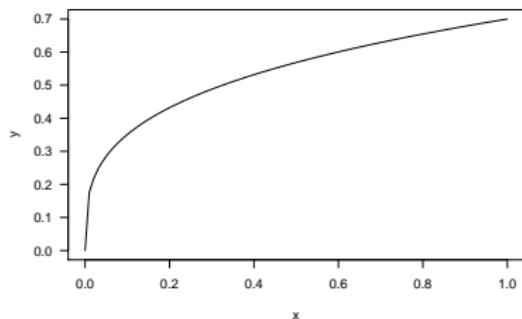
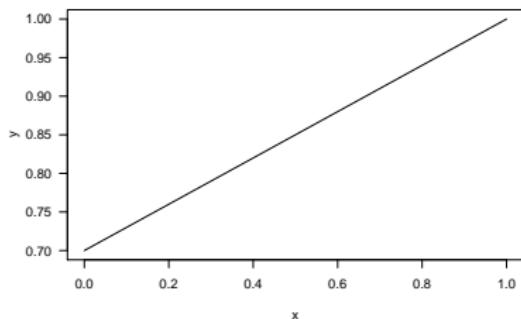
Whenever you fit logistic regression to **proportion** data, check family **quasibinomial**.

# Think about the shape of relationships

$$y \sim x + z$$

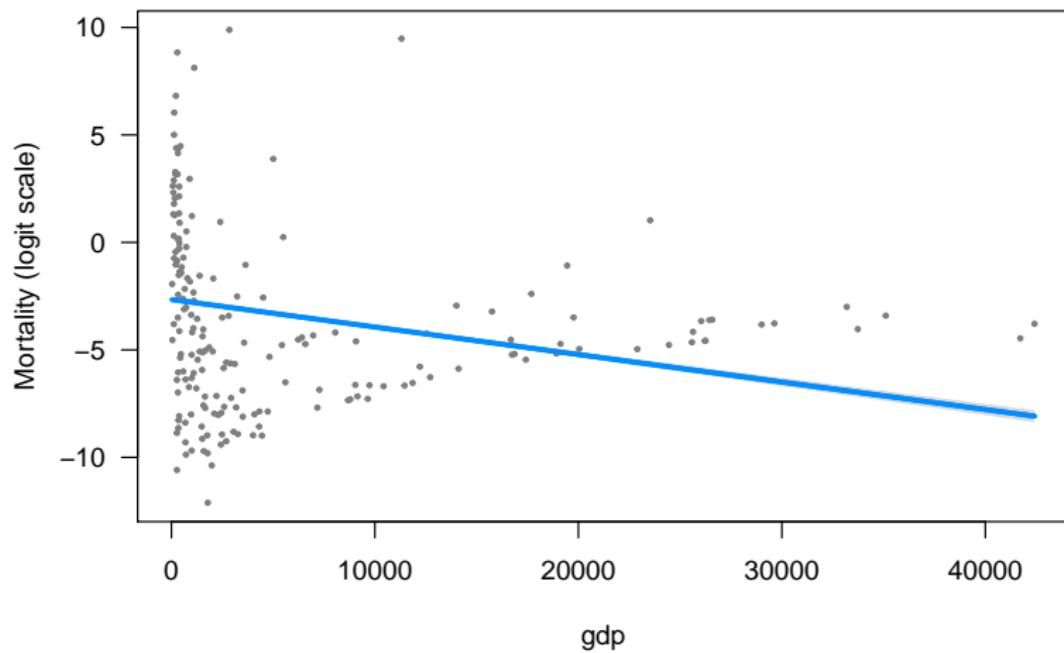
Really? Not everything has to be linear! Actually, it often is not.

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

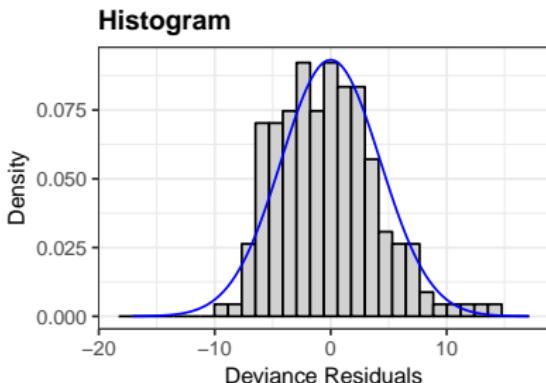
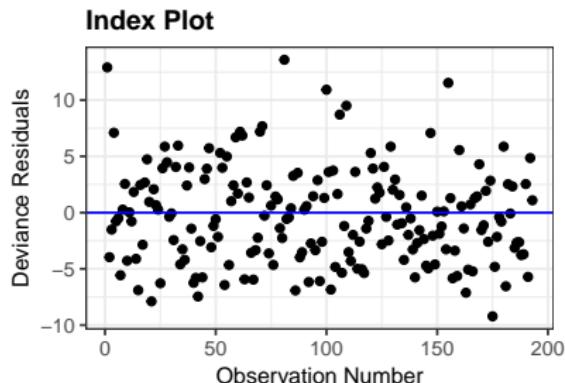
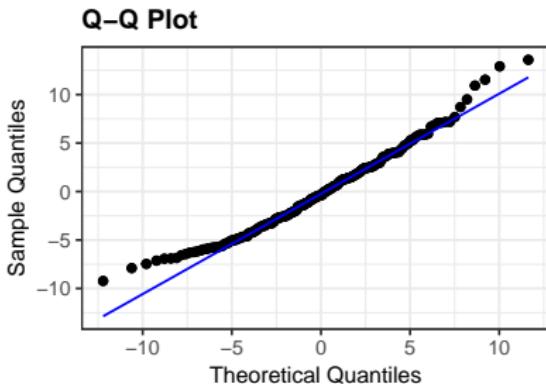
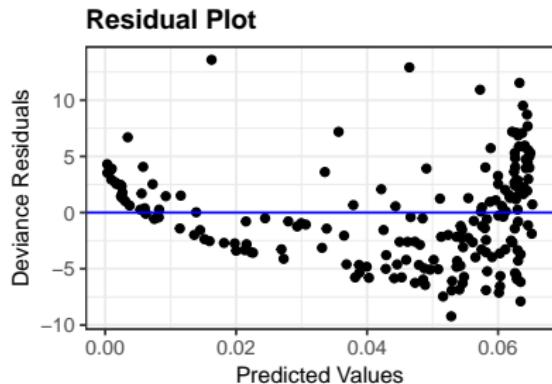


## Think about the shape of relationships

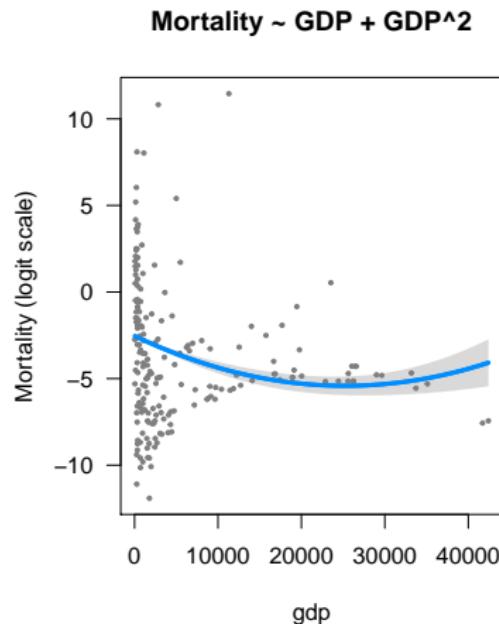
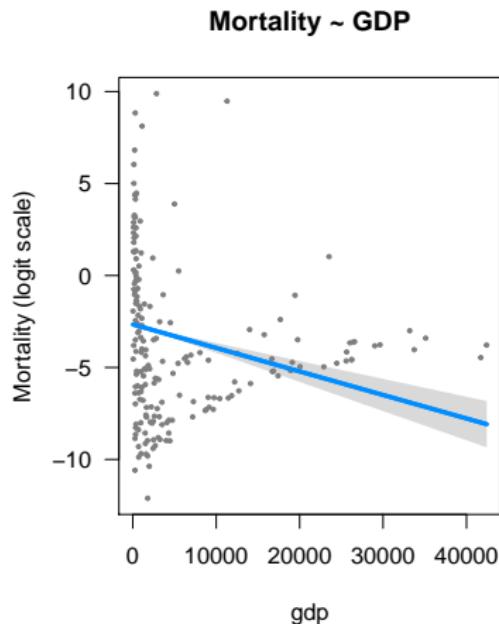
```
visreg(gdp.glm, ylab = "Mortality (logit scale)")
```



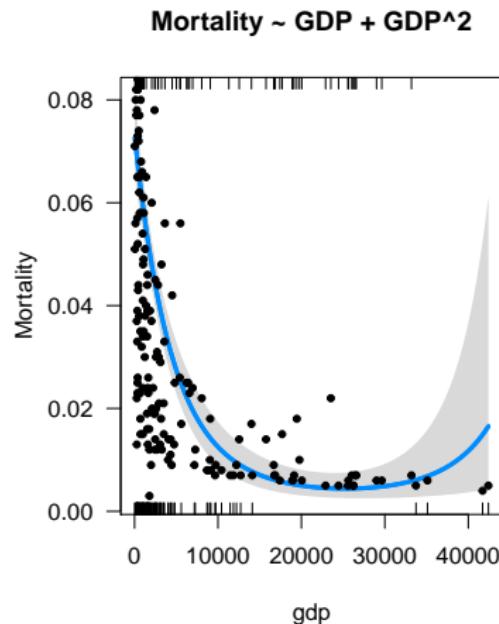
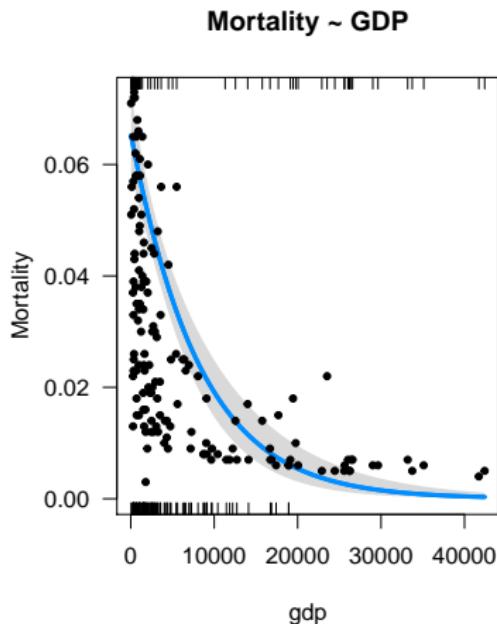
# Think about the shape of relationships



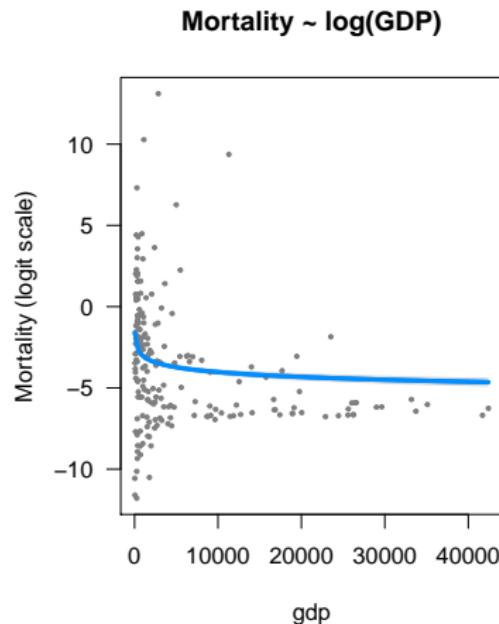
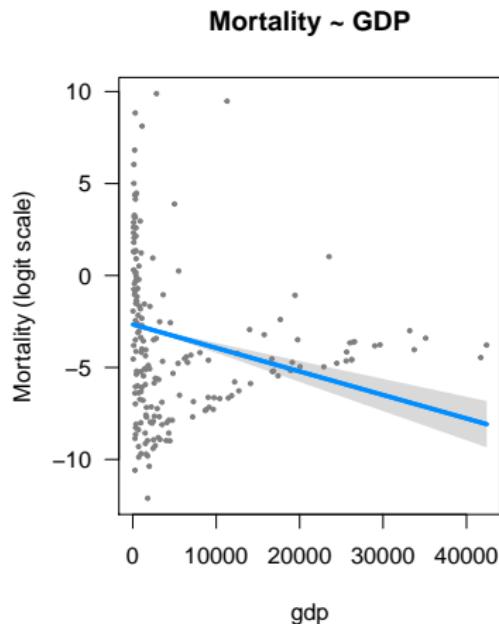
# Think about the shape of relationships



# Think about the shape of relationships

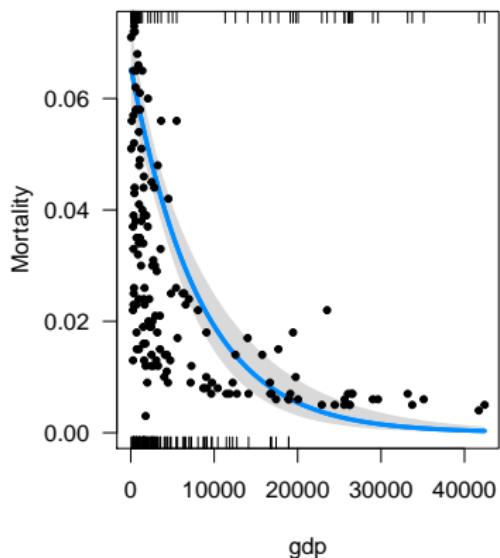


# Think about the shape of relationships

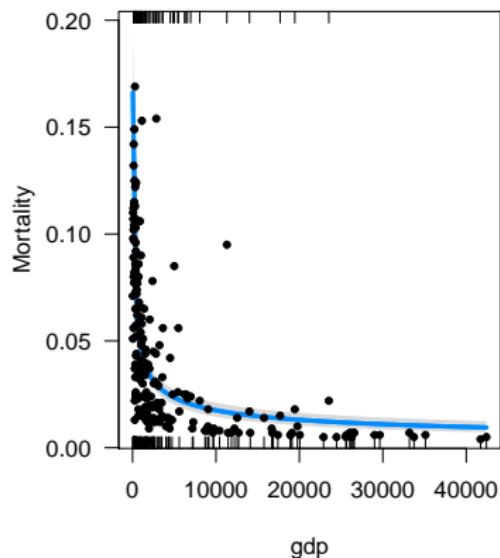


# Think about the shape of relationships

**Mortality ~ GDP**



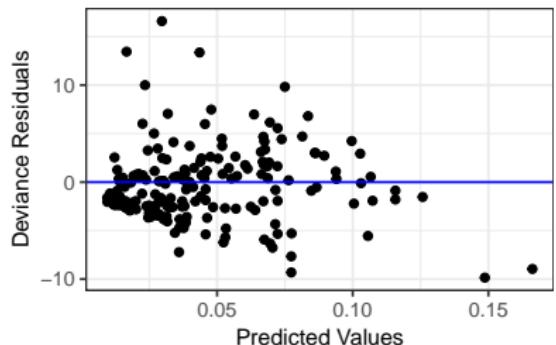
**Mortality ~ log(GDP)**



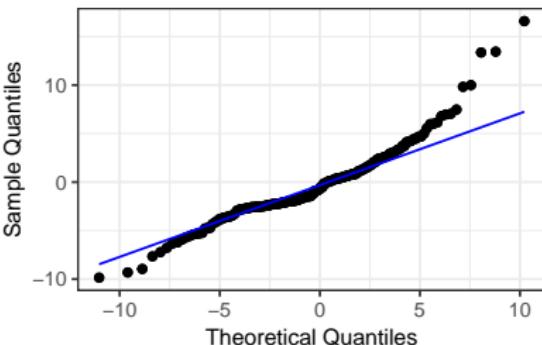
# Think about the shape of relationships

```
resid_panel(gdp.log)
```

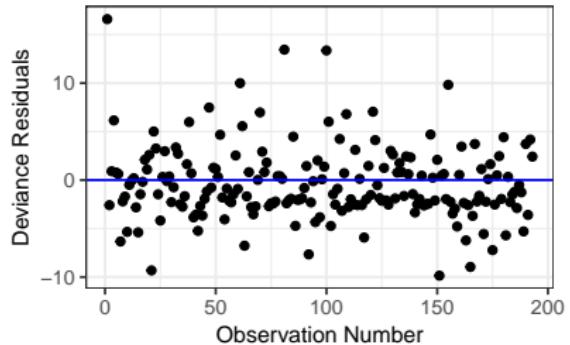
Residual Plot



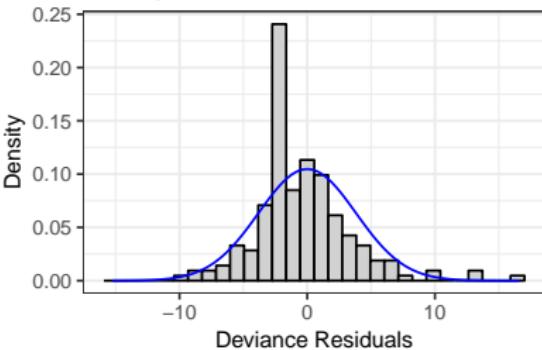
Q-Q Plot



Index Plot



Histogram



## More examples

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

## Seed set among plants

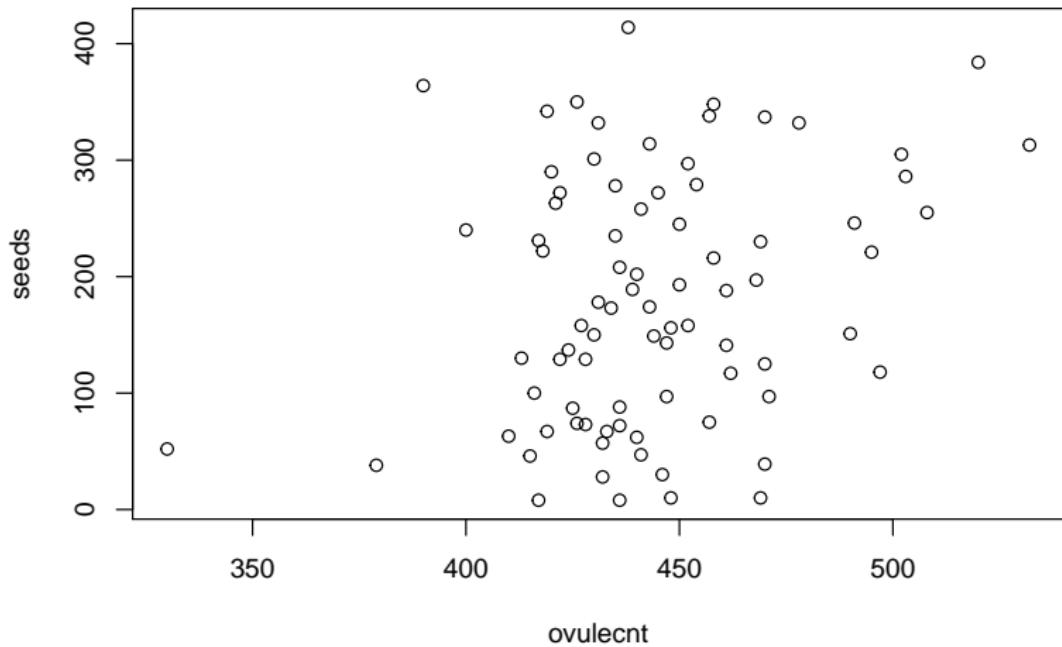
```
seed <- readr::read_csv("data-raw/seedset.csv")
head(seed)
```

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

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

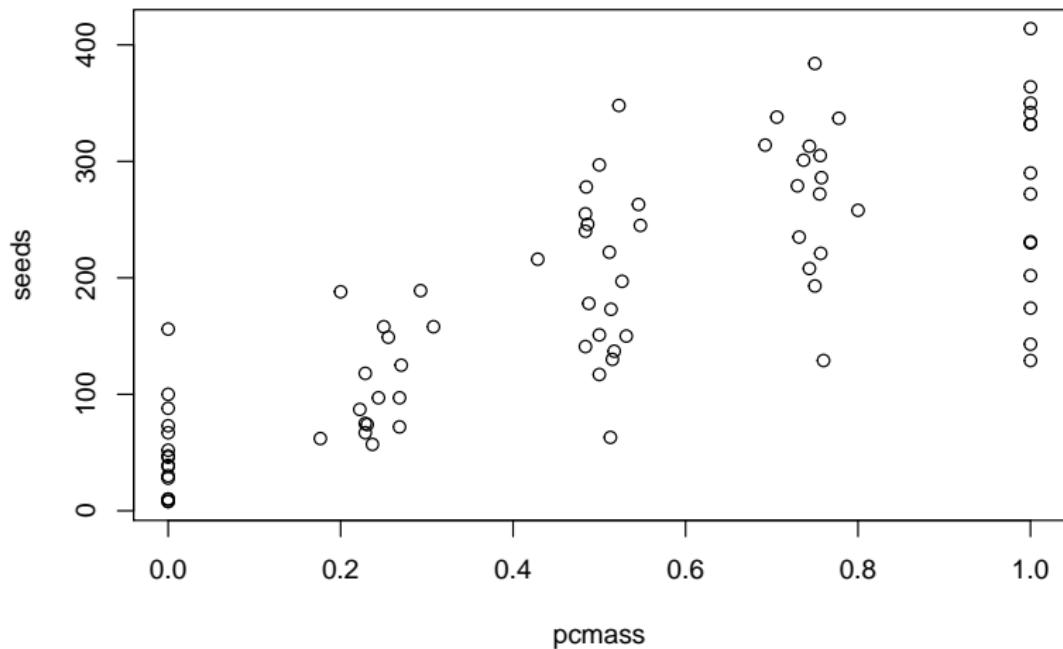
## Number of seeds vs Number of ovules

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

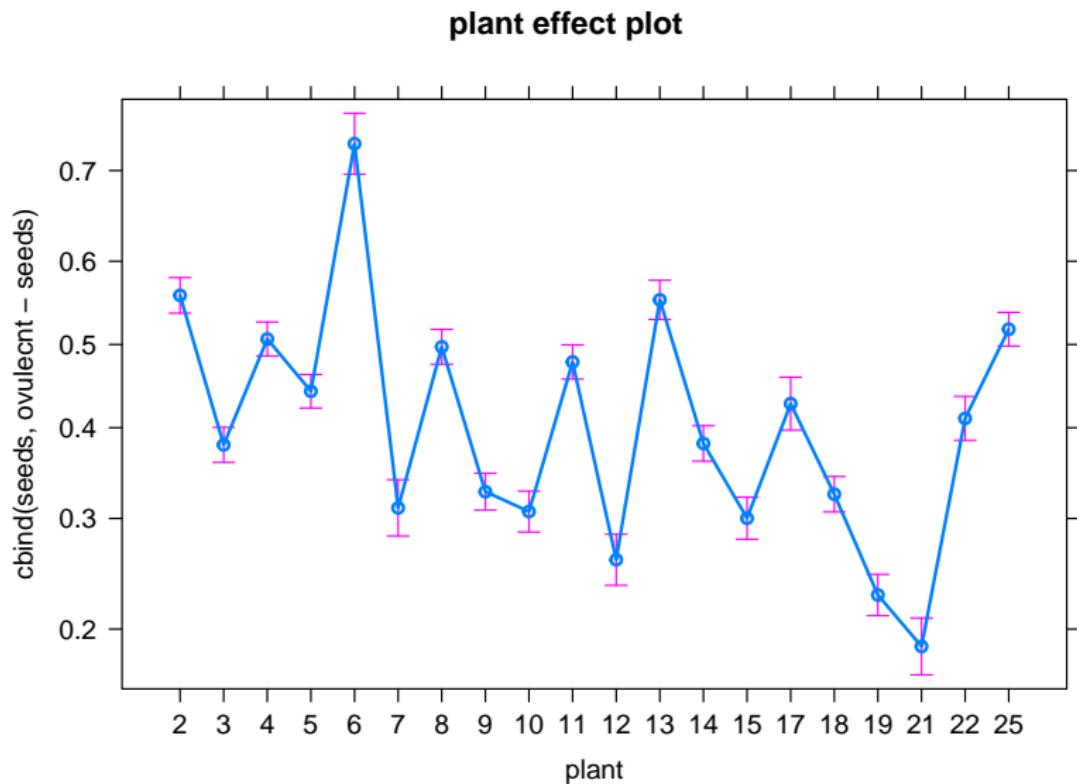


# Number of seeds vs Proportion outcross pollen

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

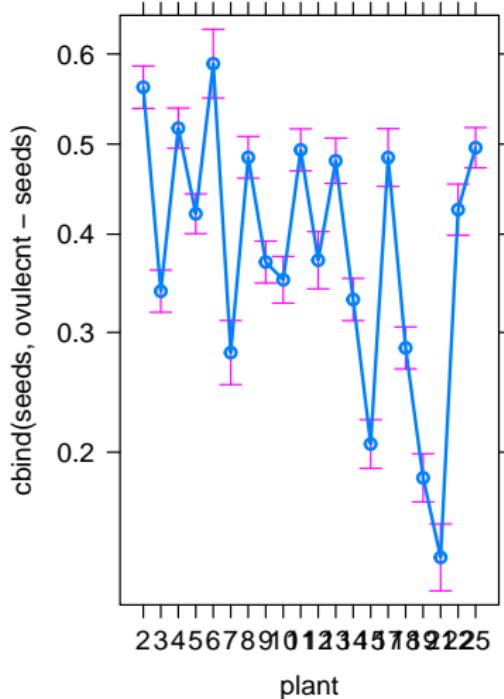


# Seed set across plants

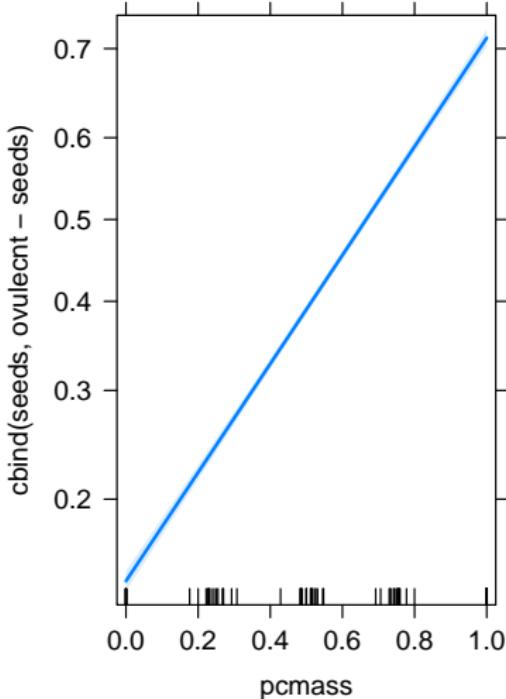


# Seed set ~ outcross pollen

**plant effect plot**



**pcmass effect plot**



GLM for count data: Poisson regression

## Types of response variable

- ▶ Gaussian: `lm`

## Types of response variable

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

## Types of response variable

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

## Poisson regression

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

Then

$$\log(N) = a + bx$$

$$N = e^{a+bx}$$

## Poisson regression

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

Then

$$\log(N) = a + bx$$

$$N = e^{a+bx}$$

## Example dataset: Seedling counts in quadrats

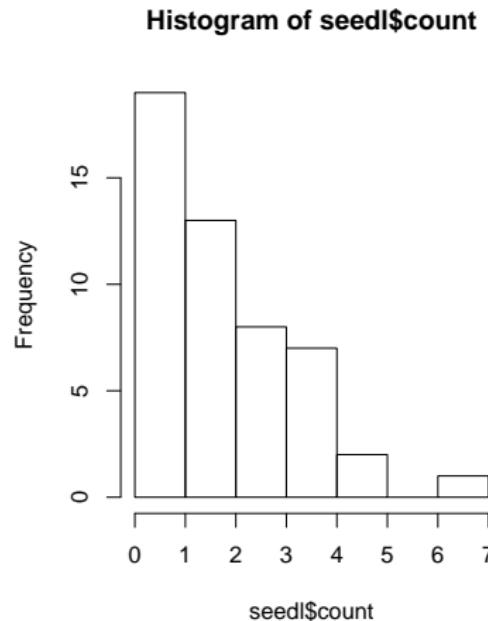
```
seedl <- read.csv("data-raw/seedlings.csv")
```

X	count	row	col
Min. : 1.00	Min. :0.00	Min. :1	Min. : 1.0
1st Qu.:13.25	1st Qu.:1.00	1st Qu.:2	1st Qu.: 3.0
Median :25.50	Median :2.00	Median :3	Median : 5.5
Mean :25.50	Mean :2.14	Mean :3	Mean : 5.5
3rd Qu.:37.75	3rd Qu.:3.00	3rd Qu.:4	3rd Qu.: 8.0
Max. :50.00	Max. :7.00	Max. :5	Max. :10.0
light	area		
Min. : 2.571	Min. :0.25		
1st Qu.:26.879	1st Qu.:0.25		
Median :47.493	Median :0.50		
Mean :47.959	Mean :0.62		
3rd Qu.:67.522	3rd Qu.:1.00		
Max. :99.135	Max. :1.00		

# EDA

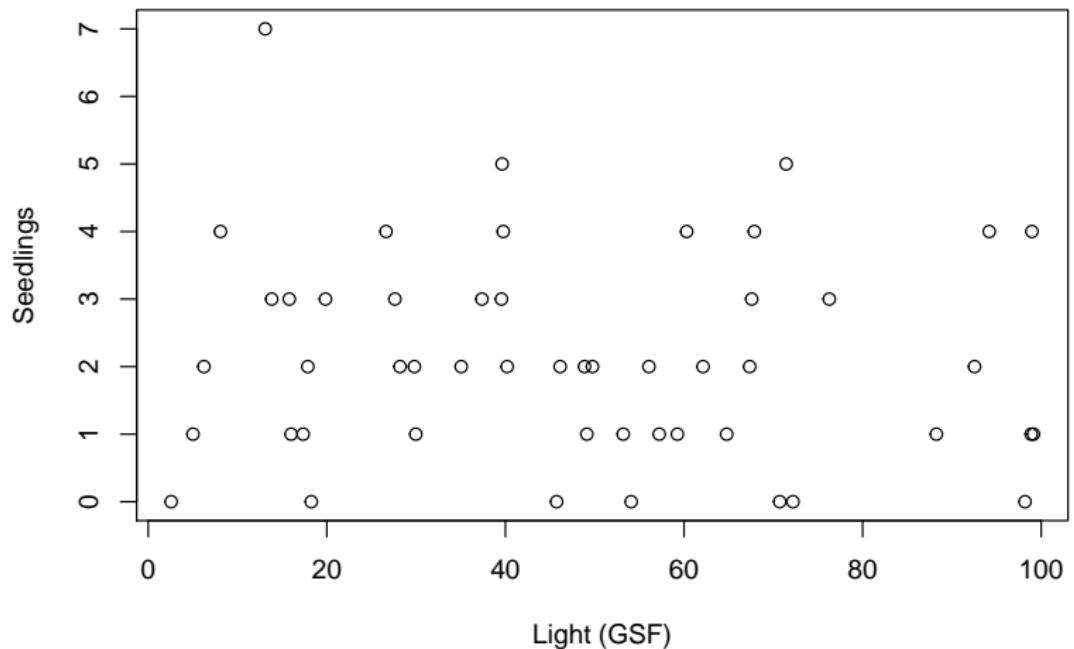
```
table(seed1$count)
```

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



## Q: Relationship between Nseedlings and light?

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



## Let's fit model (Poisson regression)

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

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

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

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.881805  0.188892  4.668 3.04e-06 ***
light       -0.002576  0.003528  -0.730   0.465
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

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

Number of Fisher Scoring iterations: 5
```

# Interpreting Poisson regression output

Parameter estimates (log scale):

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

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

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

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

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

## Using effects package

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

model: count ~ light

light effect

light	3	30	50	70	100
	2.396665	2.235657	2.123408	2.016794	1.866826

Lower 95 Percent Confidence Limits

light	3	30	50	70	100
	1.684579	1.795202	1.753373	1.567785	1.228247

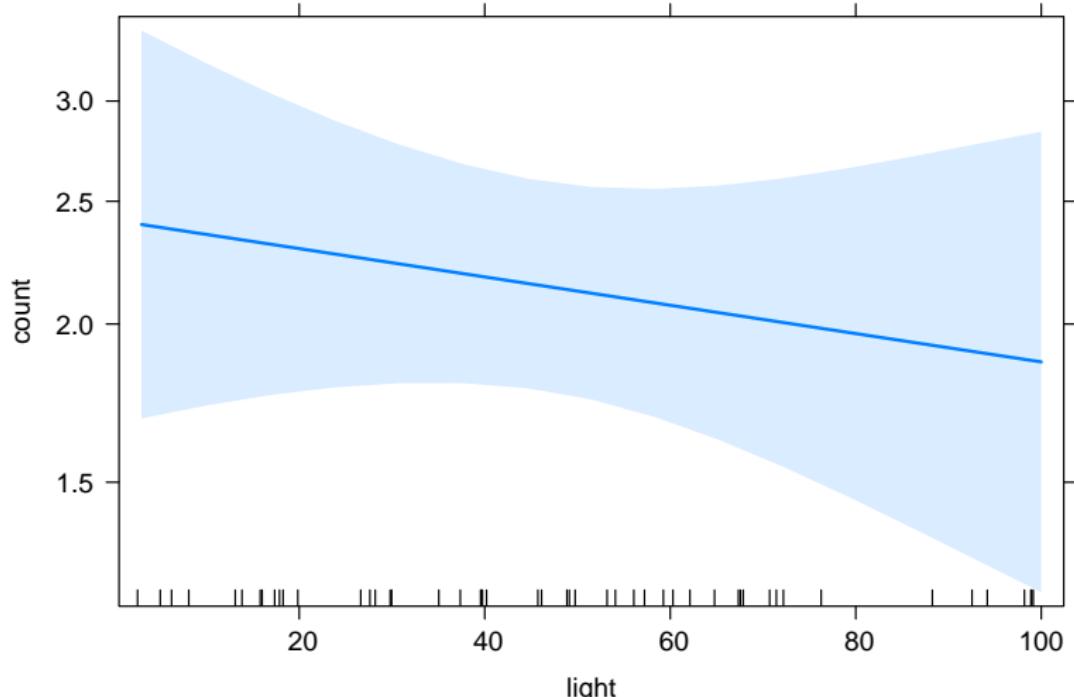
Upper 95 Percent Confidence Limits

light	3	30	50	70	100
	3.409754	2.784179	2.571535	2.594398	2.837408

# So what's the relationship between Nseedlings and light?

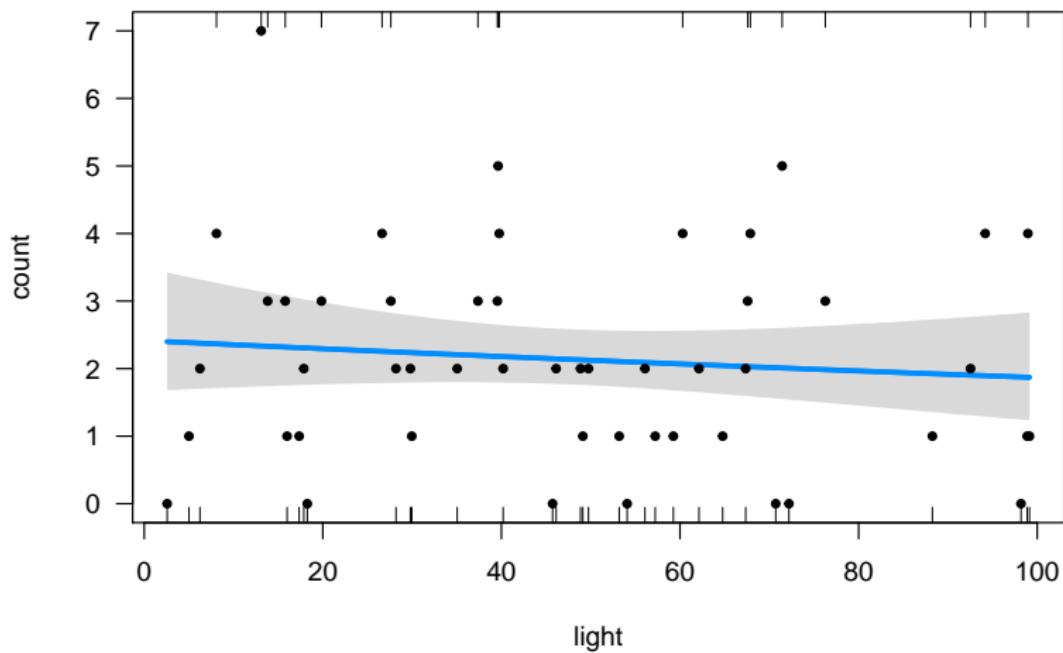
```
plot(allEffects(seed1.glm))
```

**light effect plot**



## Using visreg

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

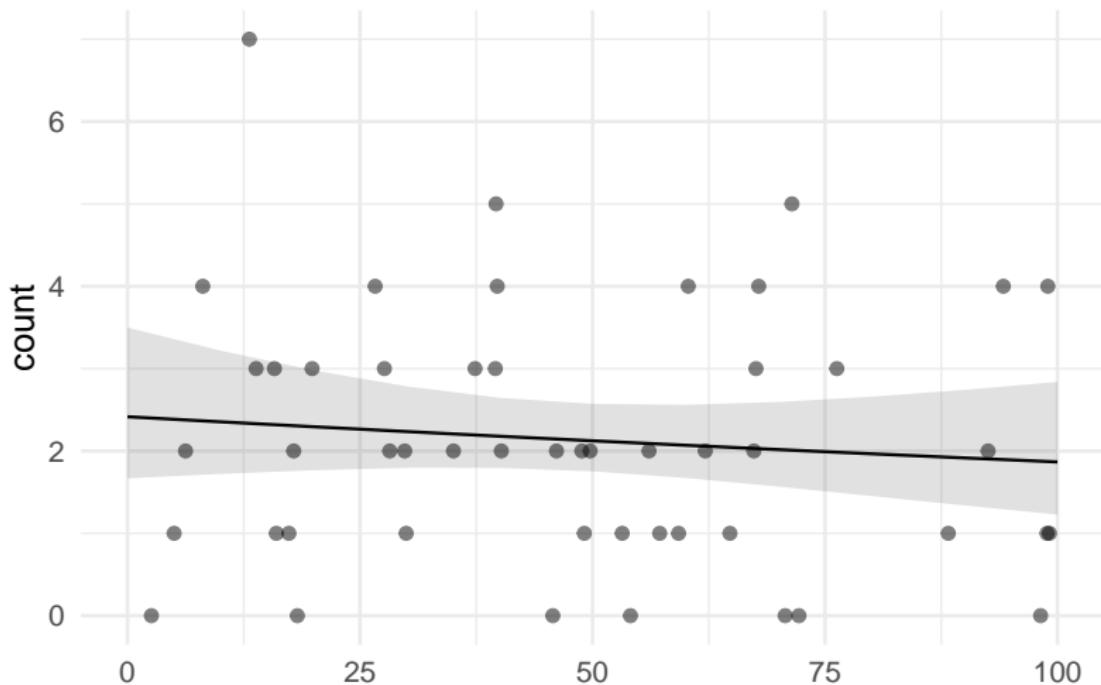


# Using sjPlot

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

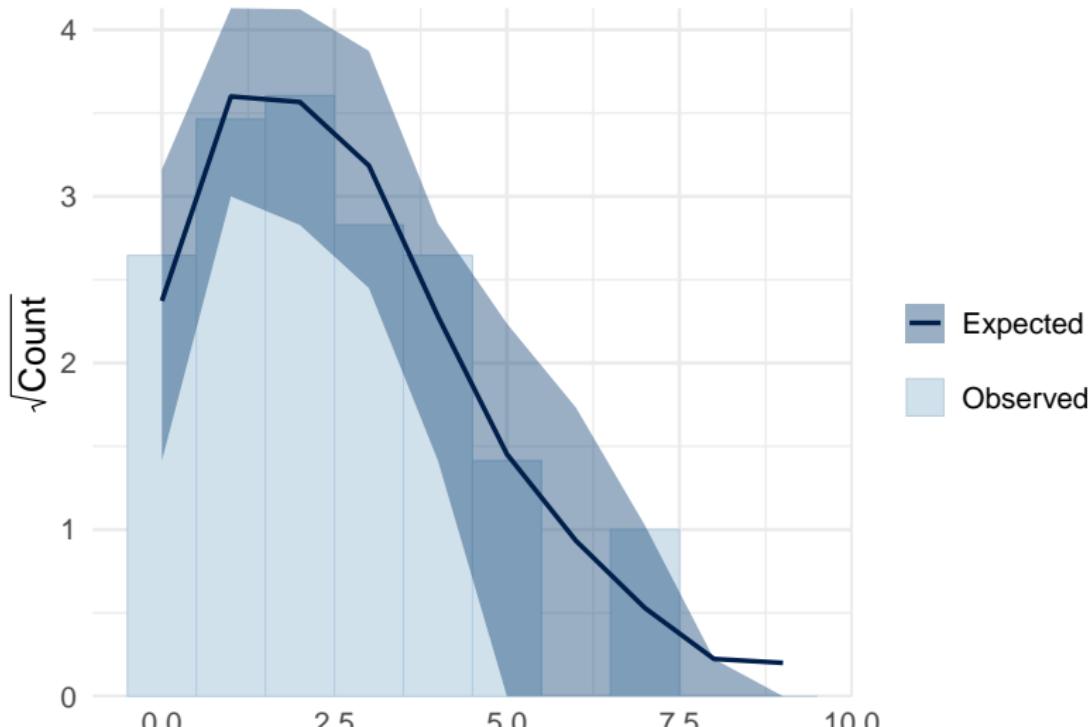
```
$light
```

Predicted counts of count

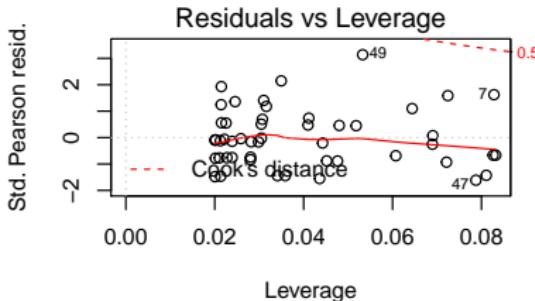
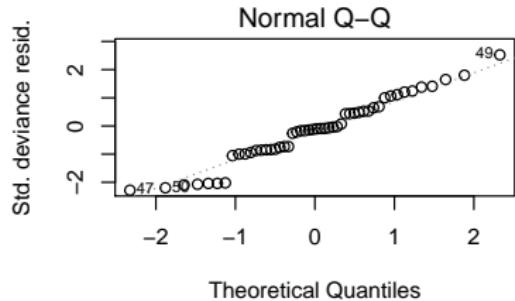
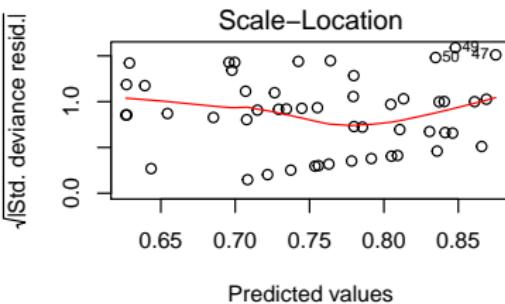
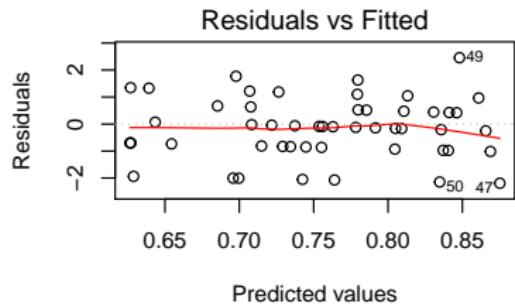


## Calibration plot with count data: rootograms

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



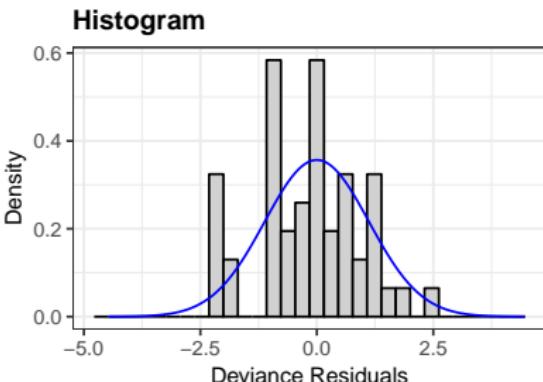
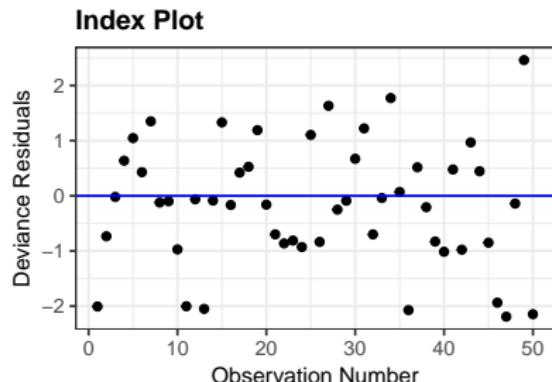
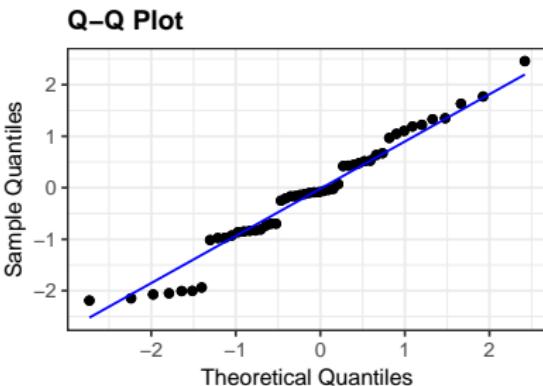
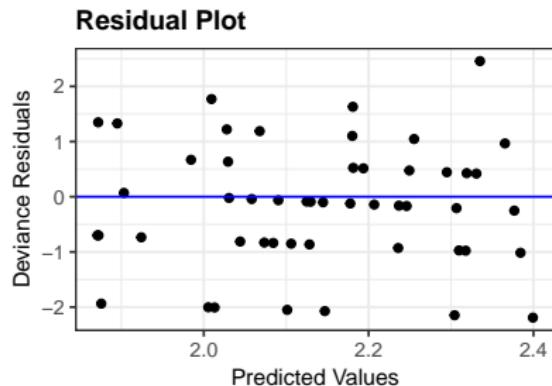
# Poisson regression: model checking



null device

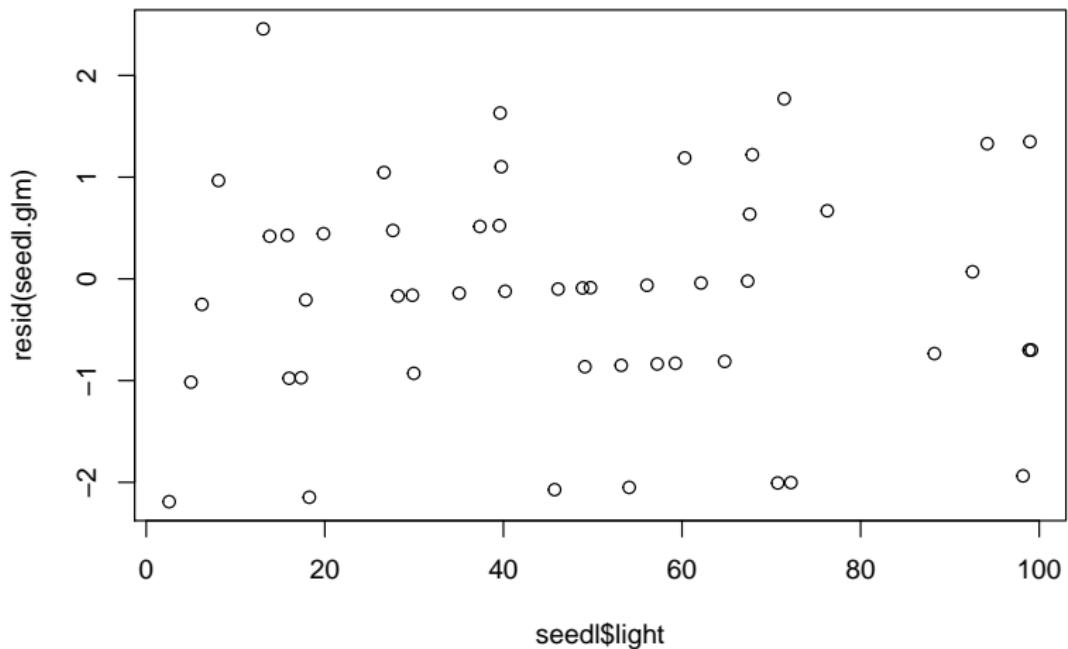
# Poisson regression: model checking

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



## Is there pattern of residuals along predictor?

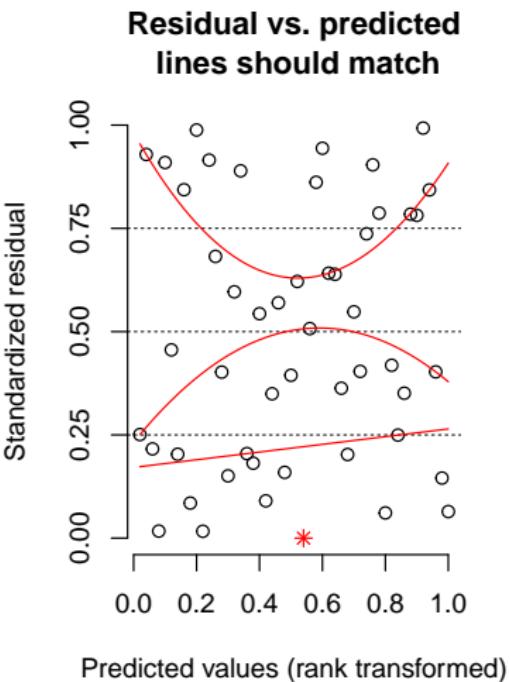
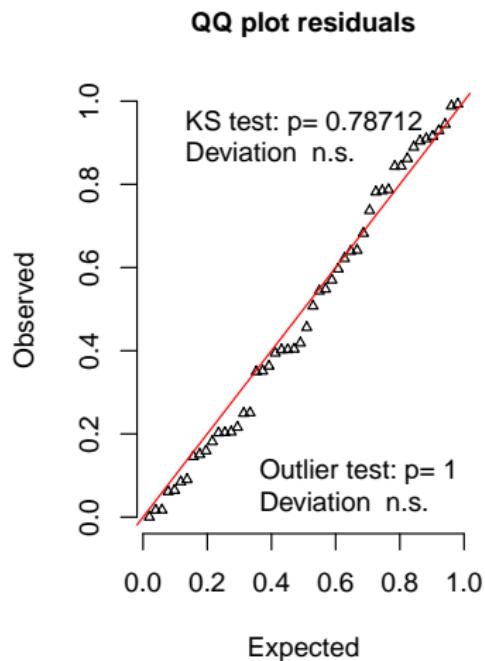
```
plot(seed1$light, resid(seed1.glm))
```



# Residuals diagnostics with DHARMA

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

DHARMA scaled residual plots



## Poisson regression: Overdispersion

## Always check overdispersion with count data

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

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

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

## Accounting for overdispersion in count data

Use family quasipoisson

Call:

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

Deviance Residuals:

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

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.881805	0.201230	4.382	6.37e-05 ***
light	-0.002576	0.003758	-0.685	0.496

---

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

(Dispersion parameter for quasipoisson family taken to be 1.1349)

Null deviance: 63.029 on 49 degrees of freedom

Residual deviance: 62.492 on 48 degrees of freedom

175/311

Mean estimates do not change after accounting for overdispersion

```
model: count ~ light
```

```
light effect  
light
```

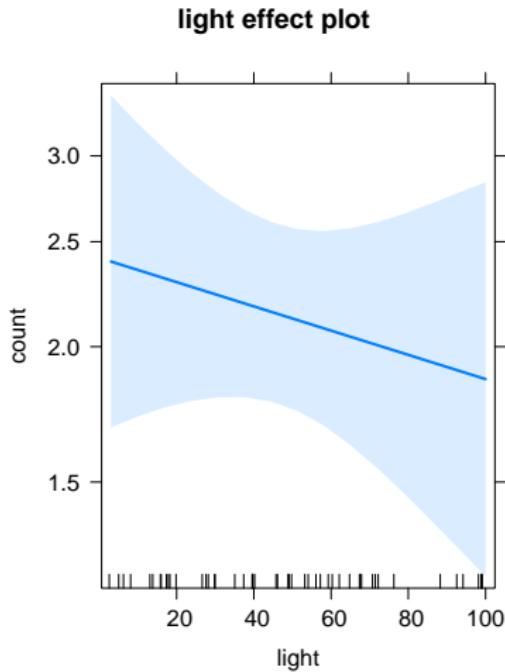
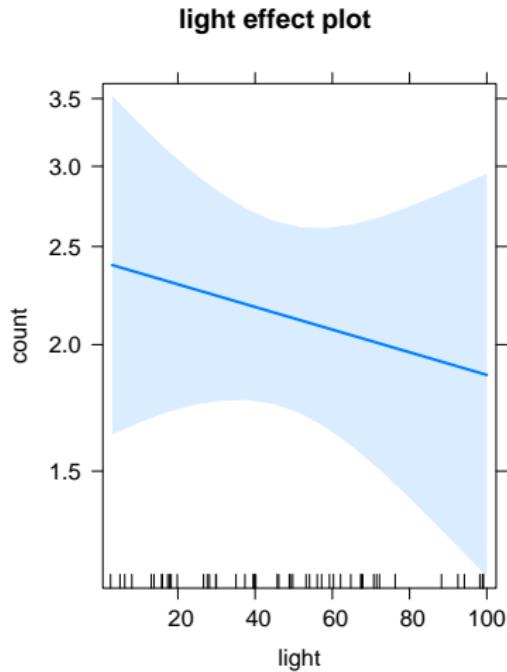
	3	30	50	70	100
2.396665	2.235657	2.123408	2.016794	1.866826	

```
model: count ~ light
```

```
light effect  
light
```

	3	30	50	70	100
2.396665	2.235657	2.123408	2.016794	1.866826	

## But standard errors may change



What if survey plots have different area?

# Avoid regression of ratios

seedlings/area  $\sim$  light

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

## **Spurious Correlation and the Fallacy of the Ratio Standard Revisited**

By RICHARD A. KRONMAL†

Figure 6

## Use offset to standardise response variables in GLMs

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

Call:

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

Deviance Residuals:

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

Coefficients:

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

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 70.263 on 49 degrees of freedom

Note estimates now referred to area units

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

	light
(Intercept)	1.3491422
	0.9955123

## Other examples

- ▶ Infant mortality  $\sim$  GDP

## Other examples

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

Mixed / Multilevel models

## Example dataset: trees

- ▶ Data on 1000 trees from 10 plots.

```
head(trees)
```

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

## Example dataset: trees

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

```
head(trees)
```

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

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

## A simple linear model

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

Call:

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

Residuals:

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

Coefficients:

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

Residual standard error: 5.549 on 998 degrees of freedom

Multiple R-squared: 0.6712, Adjusted R-squared: 0.6709

F-statistic: 2038 on 1 and 998 DF, p-value: < 2.2e-16

## Remember our model structure

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

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

In this case:

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

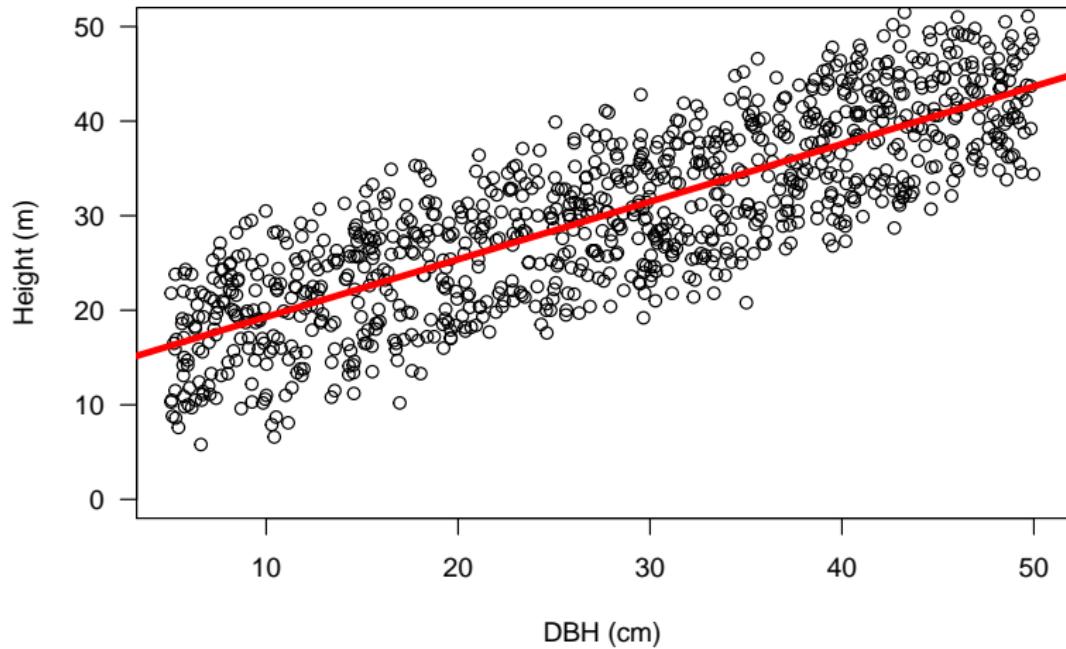
$$\mu_i = \alpha + \beta DBH_i$$

$\alpha$ : expected height when DBH = 0

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

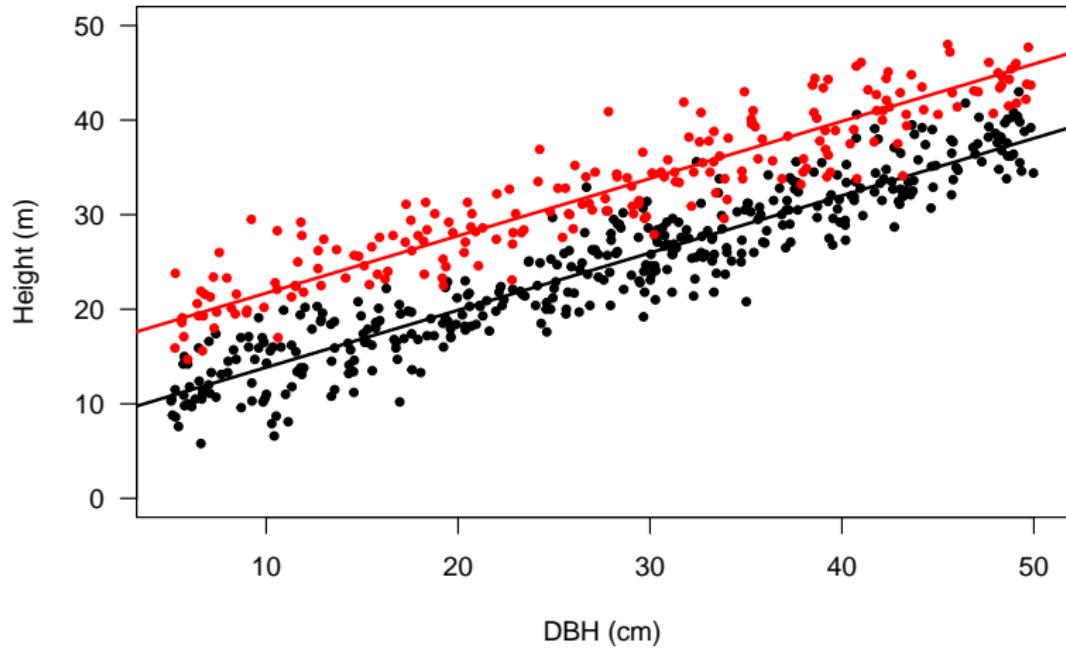
There is only one intercept

Single intercept



## What if allometry varies among plots?

Different intercept for each plot

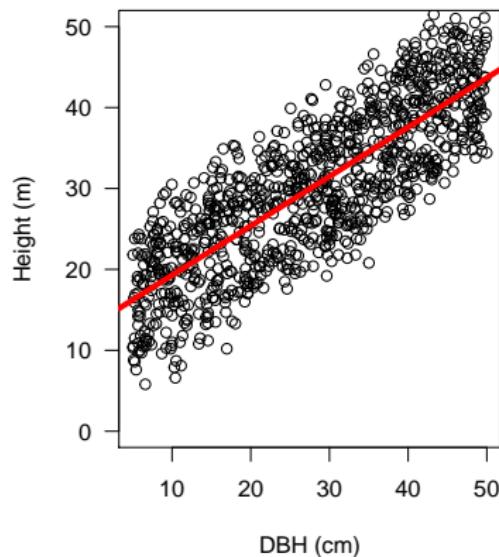


## Fitting a varying intercepts model with lm

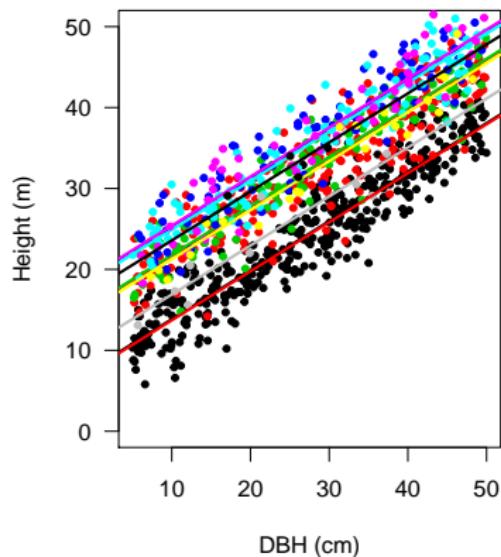
```
lm(formula = height ~ factor(plot) + dbh, data = trees)
            coef.est  coef.se
(Intercept)    7.79    0.24
factor(plot)2  7.86    0.24
factor(plot)3  7.95    0.32
factor(plot)4 11.48    0.33
factor(plot)5 11.05    0.32
factor(plot)6 11.55    0.43
factor(plot)7  7.41    0.63
factor(plot)8 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

# Single vs varying intercept

Pooling all plots



Different intercept for each plot



# Mixed models enable us to account for variability

- ▶ Varying intercepts

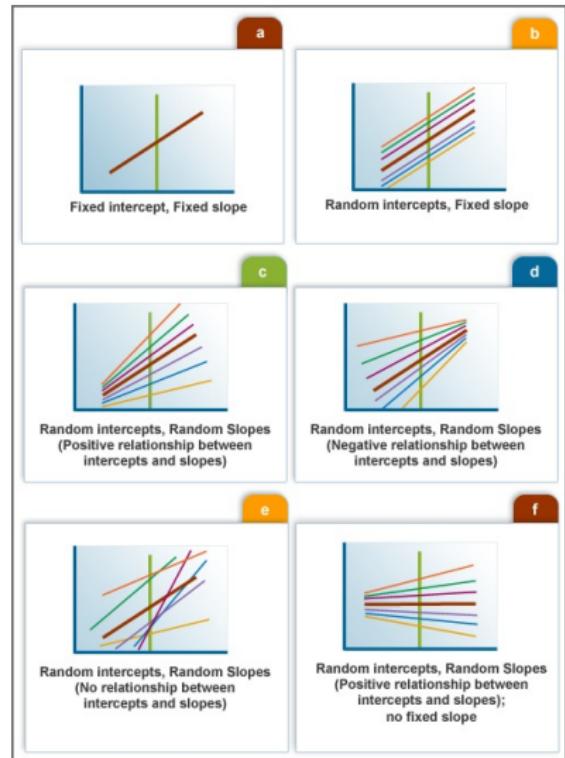


Figure 7

# Mixed models enable us to account for variability

- ▶ Varying intercepts
- ▶ Varying slopes

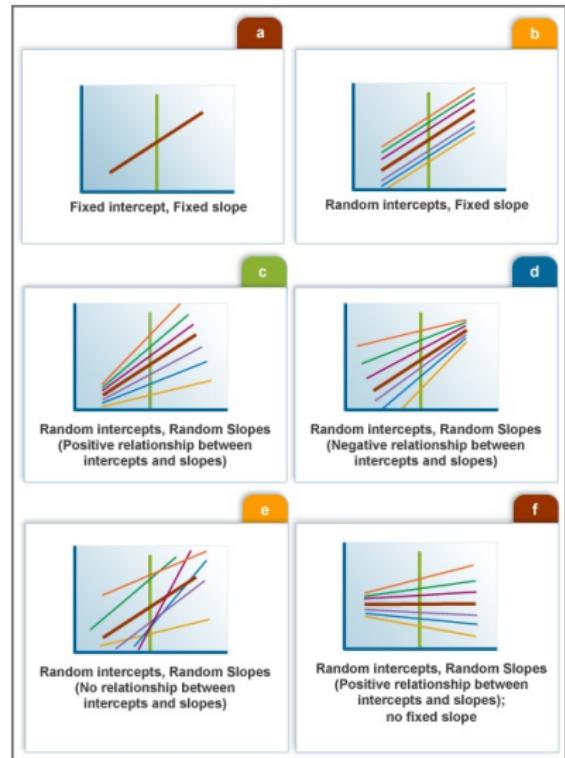


Figure 7

## Mixed model with varying intercepts

$$y_i = a_j + b x_i + \varepsilon_i$$

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

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

En nuestro ejemplo:

$$Height_i = plot_j + bDBH_i + \varepsilon_i$$

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

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

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

Hence there's gradient between

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

Hence there's gradient between

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

Hence there's gradient between

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

Hence there's gradient between

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

Hence there's gradient between

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

Hence there's gradient between

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

## Random vs Fixed effects?

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

[http://andrewgelman.com/2005/01/25/why\\_i\\_dont\\_use/](http://andrewgelman.com/2005/01/25/why_i_dont_use/)

## Random vs Fixed effects?

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

[http://andrewgelman.com/2005/01/25/why\\_i\\_dont\\_use/](http://andrewgelman.com/2005/01/25/why_i_dont_use/)

## Random vs Fixed effects?

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

[http://andrewgelman.com/2005/01/25/why\\_i\\_dont\\_use/](http://andrewgelman.com/2005/01/25/why_i_dont_use/)

## Random vs Fixed effects?

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

[http://andrewgelman.com/2005/01/25/why\\_i\\_dont\\_use/](http://andrewgelman.com/2005/01/25/why_i_dont_use/)

## Random vs Fixed effects?

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

[http://andrewgelman.com/2005/01/25/why\\_i\\_dont\\_use/](http://andrewgelman.com/2005/01/25/why_i_dont_use/)

## What is a random effect, really?

1. Varies by group

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

## What is a random effect, really?

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

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

# Shrinkage improves parameter estimation

Especially for groups with low sample size

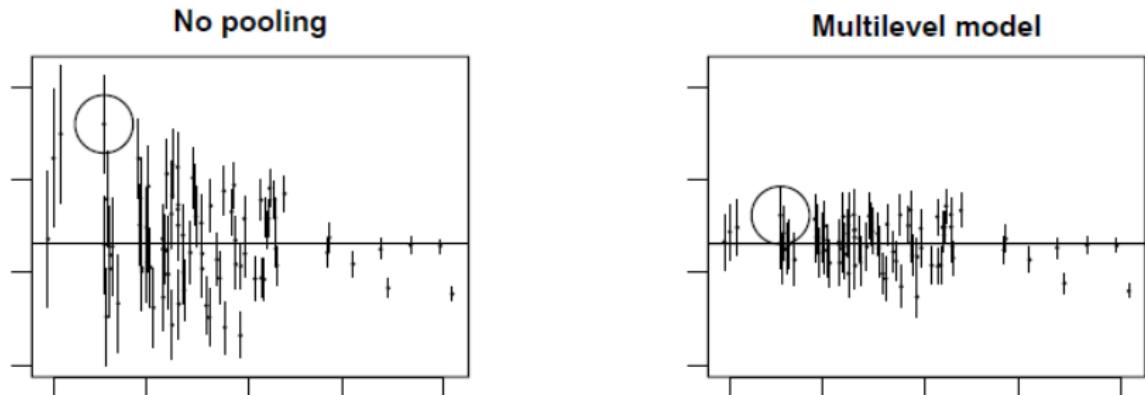


Figure 8

*From Gelman & Hill p. 253*

## Fitting mixed/multilevel models

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

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

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

## Retrieve model coefficients

```
coef(mixed)
```

```
$plot
  (Intercept)      dbh
1      7.798373 0.6056549
2     15.647613 0.6056549
3     15.735397 0.6056549
4     19.253661 0.6056549
5     18.819467 0.6056549
6     19.306574 0.6056549
7     15.197908 0.6056549
8     11.016485 0.6056549
9     17.265448 0.6056549
10    7.940715 0.6056549
```

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

# Broom: model estimates in tidy form

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

```
# A tibble: 4 x 5
  term          estimate std.error statistic group
  <chr>        <dbl>     <dbl>     <dbl> <chr>
1 (Intercept)    14.8      1.44      10.3  fixed
2 dbh            0.606     0.00704    86.0  fixed
3 sd_(Intercept).plot  4.45     NA        NA    plot
4 sd_Observation.Residual 2.89     NA        NA    Residual
```

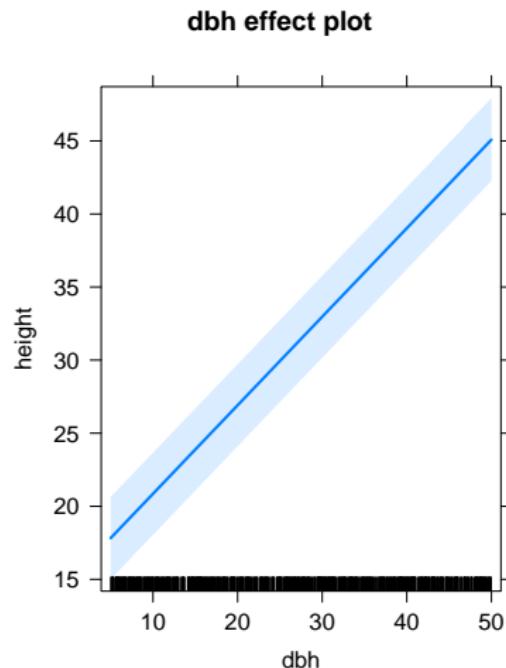
See also `broom.mixed`

# Visualising model: allEffects

```
model: height ~ dbh
```

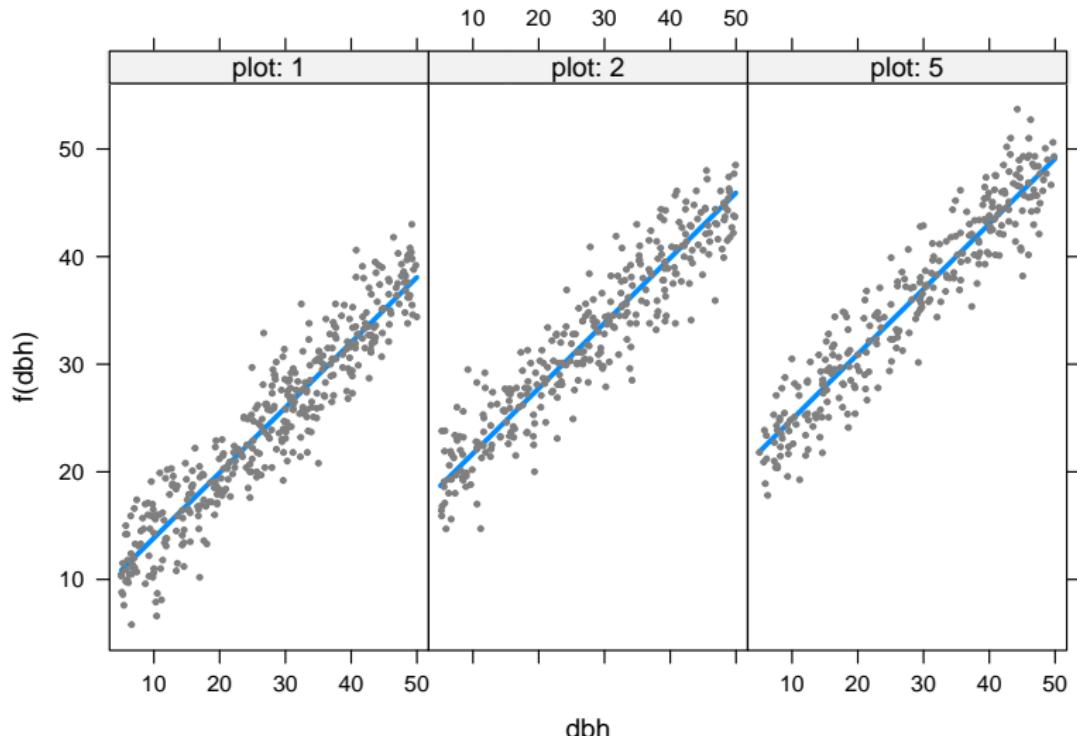
```
dbh effect  
dbh
```

5	20	30	40	50
17.82644	26.91126	32.96781	39.02436	45.08091



## Visualising model: visreg

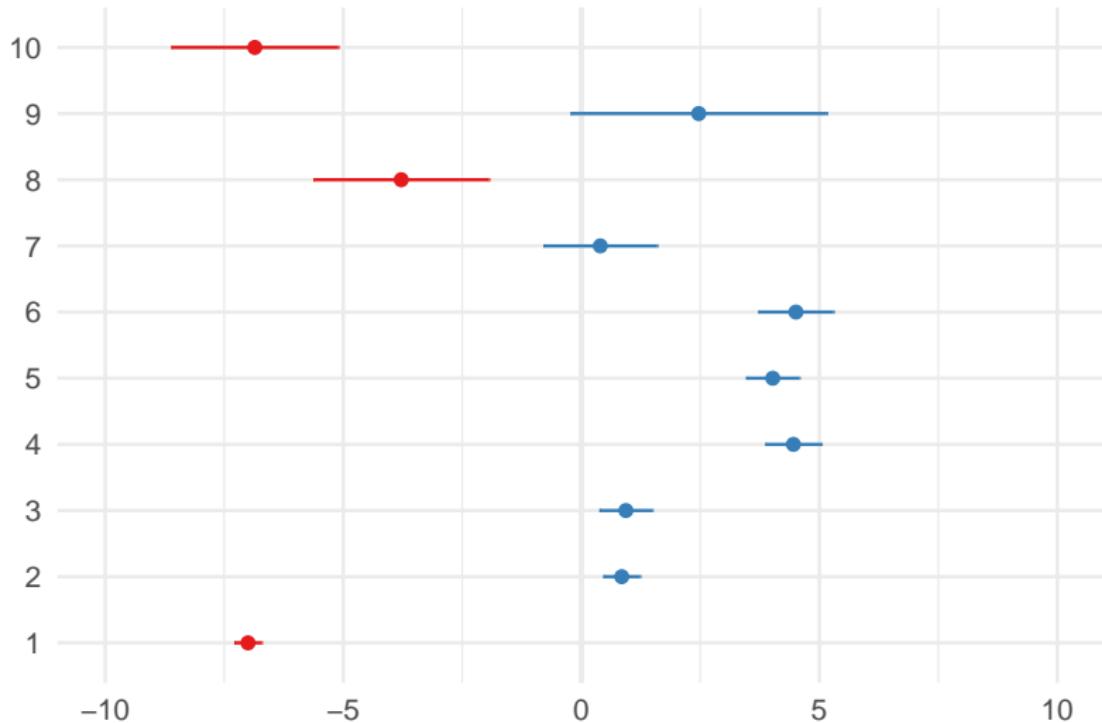
```
visreg(mixed, xvar = "dbh", by = "plot", re.form = ~ (1 | plot))
```



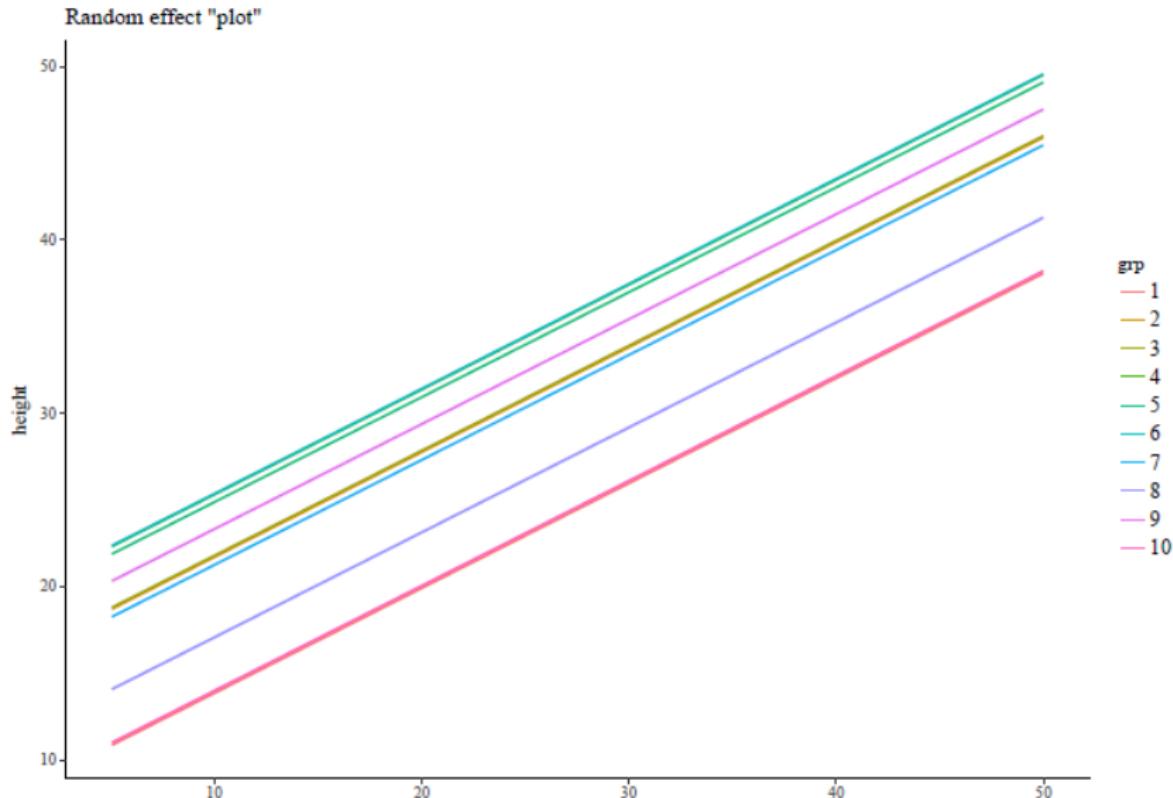
# Visualising model: sjPlot

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

## Random effects



# Visualising model

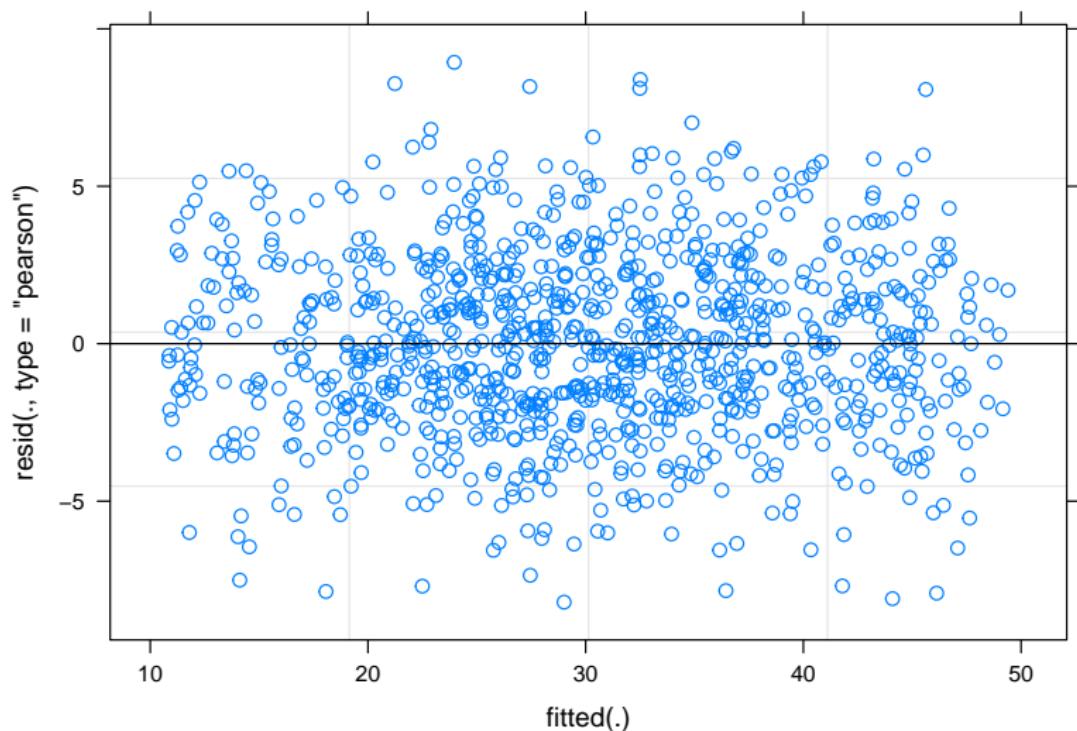


## Using merTools to understand fitted model

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

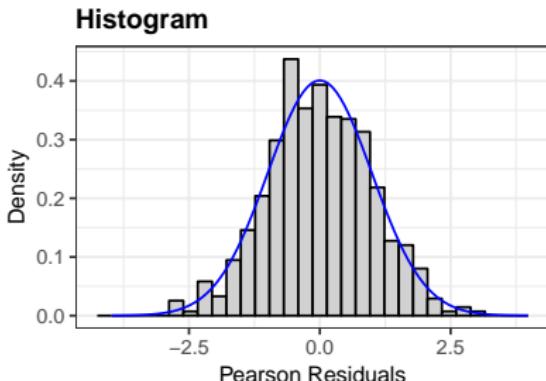
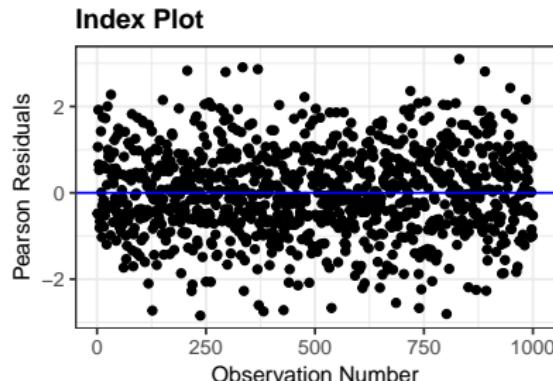
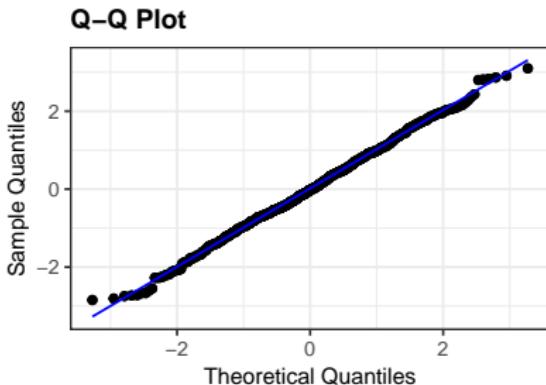
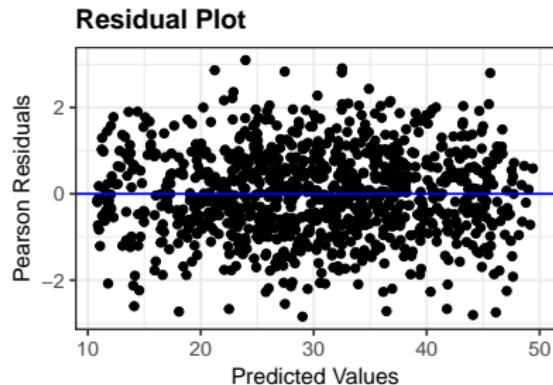
## Checking residuals

```
plot(mixed)
```



# Checking residuals

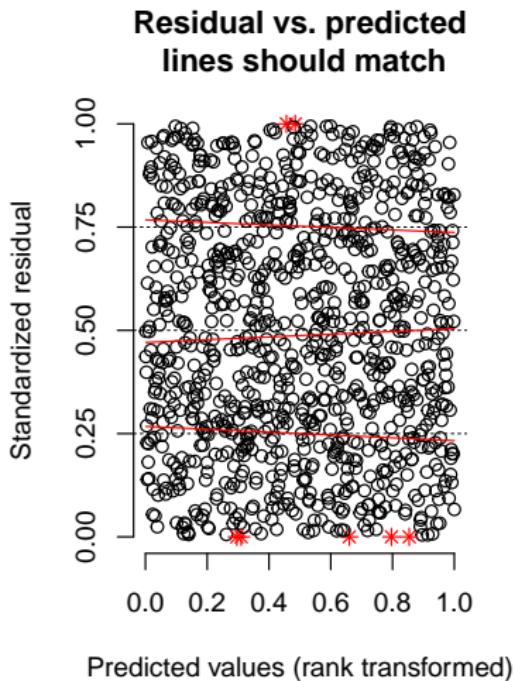
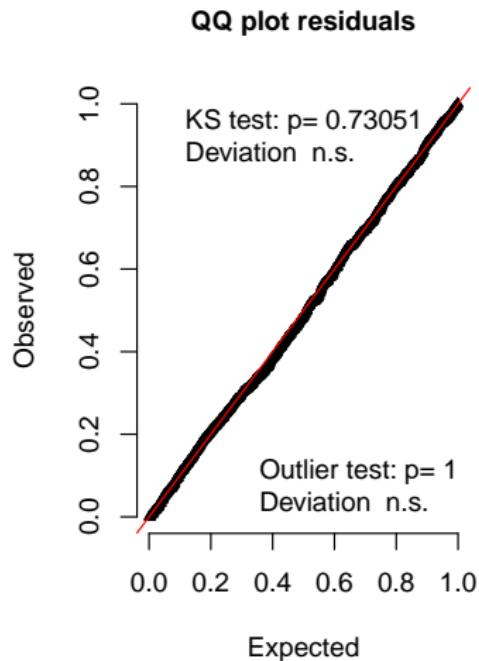
```
ggResidpanel:::resid_panel(mixed)
```



# Checking residuals (DHARMA)

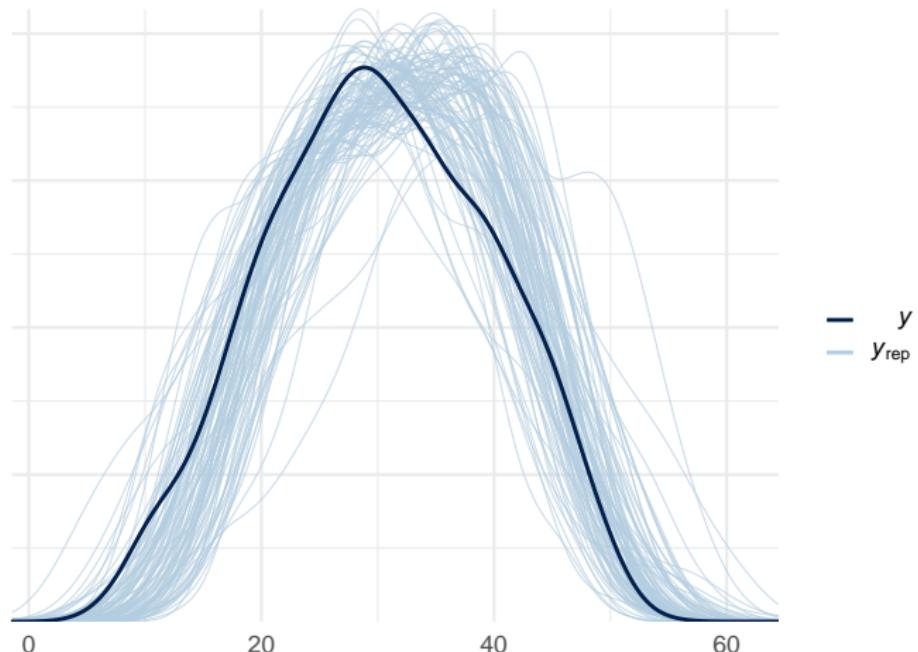
```
DHARMA::simulateResiduals(mixed, plot = TRUE, use.u = TRUE)
```

DHARMA scaled residual plots



## Model checking with simulated data

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



## R-squared for GLMMs

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

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

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

	R2m	R2c
[1,]	0.687565	0.9076325

Growing the hierarchy: adding plot-level  
predictors

## Model with group-level predictors

We had:

$$y_i = a_j + b x_i + \varepsilon_i$$

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

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

Now

$$y_i = a_j + b x_i + \varepsilon_i$$

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

$$\mu_j = \gamma + \delta \cdot \text{predictor}_j$$

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

## Merging trees and plot data

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

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

## Centre continuous variables

Plot temperatures referred as deviations from 15°C

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

## Fit multilevel model

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

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

Error terms:

Groups	Name	Std.Dev.
plot	(Intercept)	3.61
	Residual	2.89

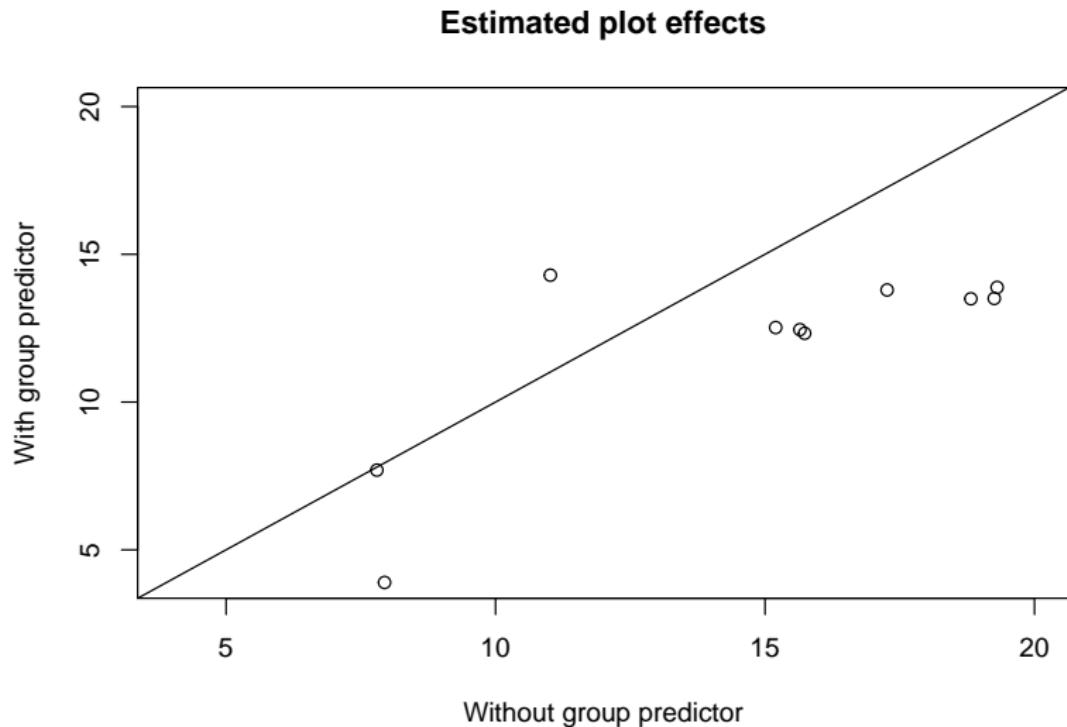
---

number of obs: 1000, groups: plot, 10  
AIC = 5012.8, DIC = 4991  
deviance = 4996.9

## Examine model with merTools

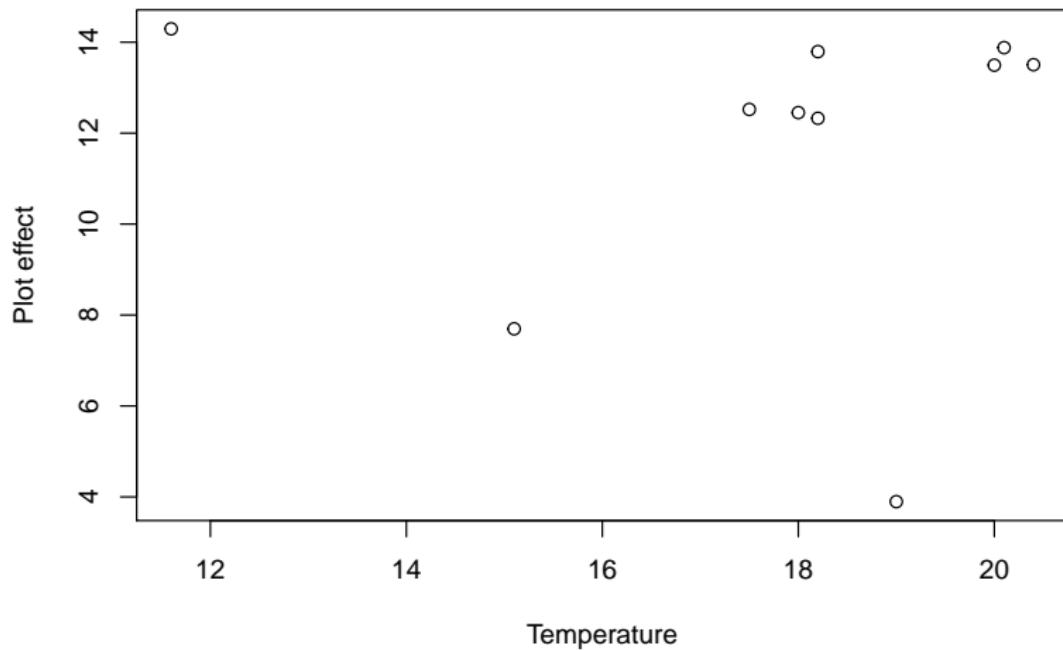
```
shinyMer(group.pred)
```

# Comparing plot effects with and without group predictor



## Are plot effects related to temperature?

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



Varying intercepts and slopes

## Varying intercepts and slopes

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

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

## Varying intercepts and slopes

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

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

## Varying intercepts and slopes

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

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

## Varying intercepts and slopes

```
lmer(formula = height ~ dbh + (1 + dbh | plot), data = trees)
      coef.est coef.se
(Intercept) 14.82     1.30
  dbh         0.61     0.01
```

Error terms:

Groups	Name	Std.Dev.	Corr
plot	(Intercept)	3.98	
	dbh	0.01	-0.29
Residual		2.89	

---

number of obs: 1000, groups: plot, 10  
AIC = 5018.9, DIC = 4995.6  
deviance = 5001.3

## Varying intercepts and slopes

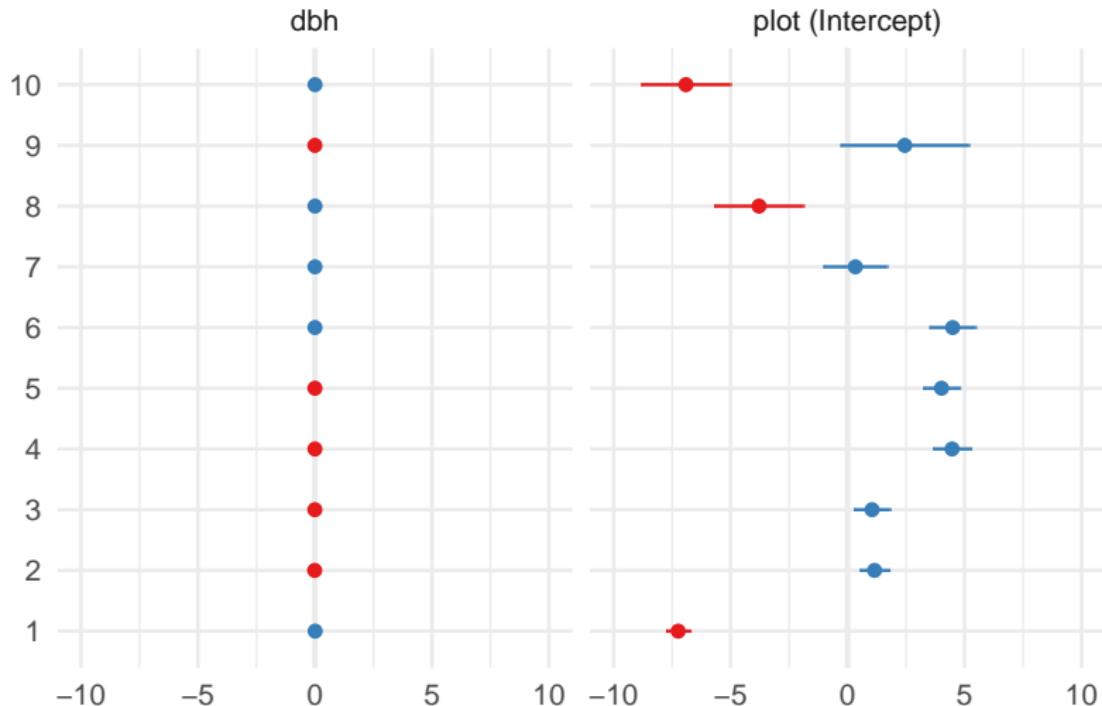
```
$plot
  (Intercept)      dbh
1      7.577721 0.6136944
2     15.969067 0.5941570
3     15.862676 0.6010346
4     19.282902 0.6043557
5     18.832500 0.6049803
6     19.307373 0.6051534
7     15.148412 0.6071041
8     11.032456 0.6072821
9     17.259351 0.6032577
10    7.908138 0.6092241

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

# Visualising model: sjPlot

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

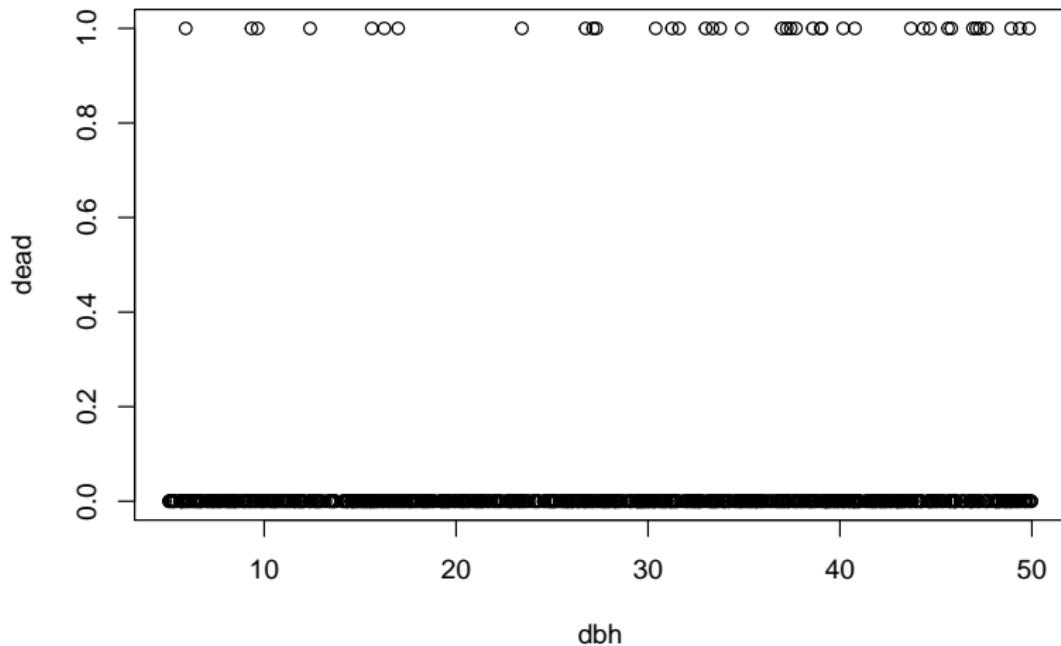
## Random effects



## Multilevel logistic regression

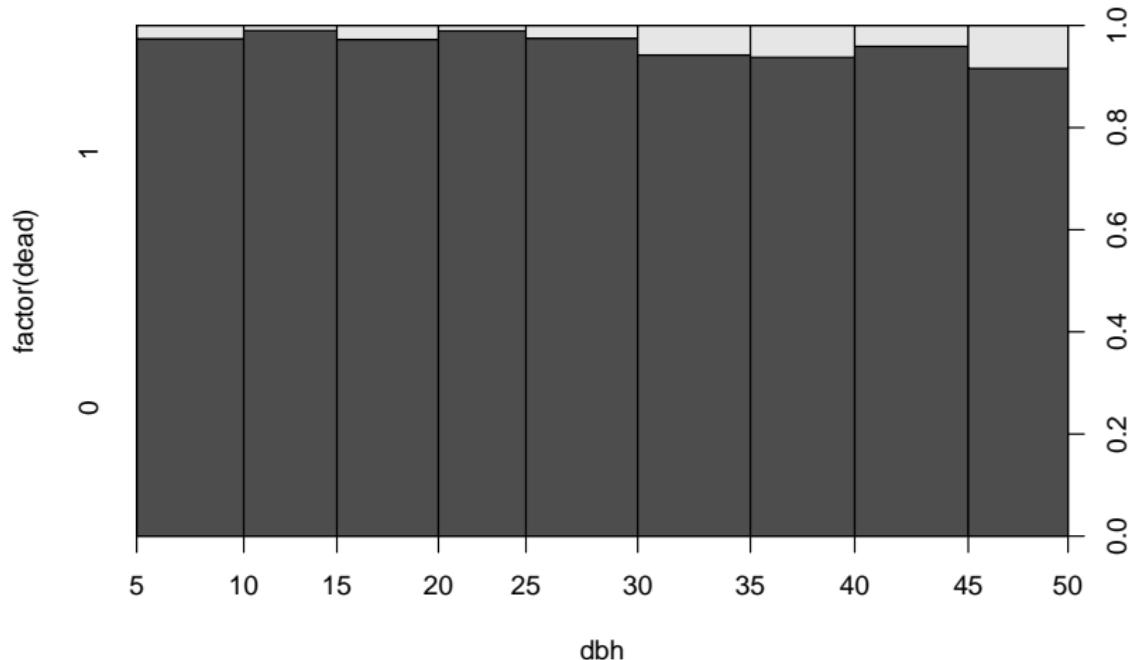
## Q: Relationship between tree size and mortality

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



## Q: Relationship between tree size and mortality

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



## Fit simple logistic regression

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

Call:

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

Deviance Residuals:

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

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-4.46945	0.49445	-9.039	< 2e-16 ***
dbh	0.04094	0.01380	2.967	0.00301 **
---				

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

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 329.51 on 999 degrees of freedom

## Logistic regression with *independent* plot effects

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

Call:

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

Deviance Residuals:

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

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )	
(Intercept)	-4.40106	0.52997	-8.304	<2e-16	***
dbh	0.04060	0.01386	2.929	0.0034	**
factor(plot)2	-0.59168	0.52132	-1.135	0.2564	
factor(plot)3	0.54576	0.47094	1.159	0.2465	
factor(plot)4	0.05507	0.57434	0.096	0.9236	
factor(plot)5	-0.38312	0.64222	-0.597	0.5508	
factor(plot)6	-0.08426	0.76908	-0.110	0.9128	
factor(plot)7	0.03126	1.06064	0.029	0.9765	

## Fit multilevel logistic regression

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

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

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

## Retrieve model coefficients

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

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

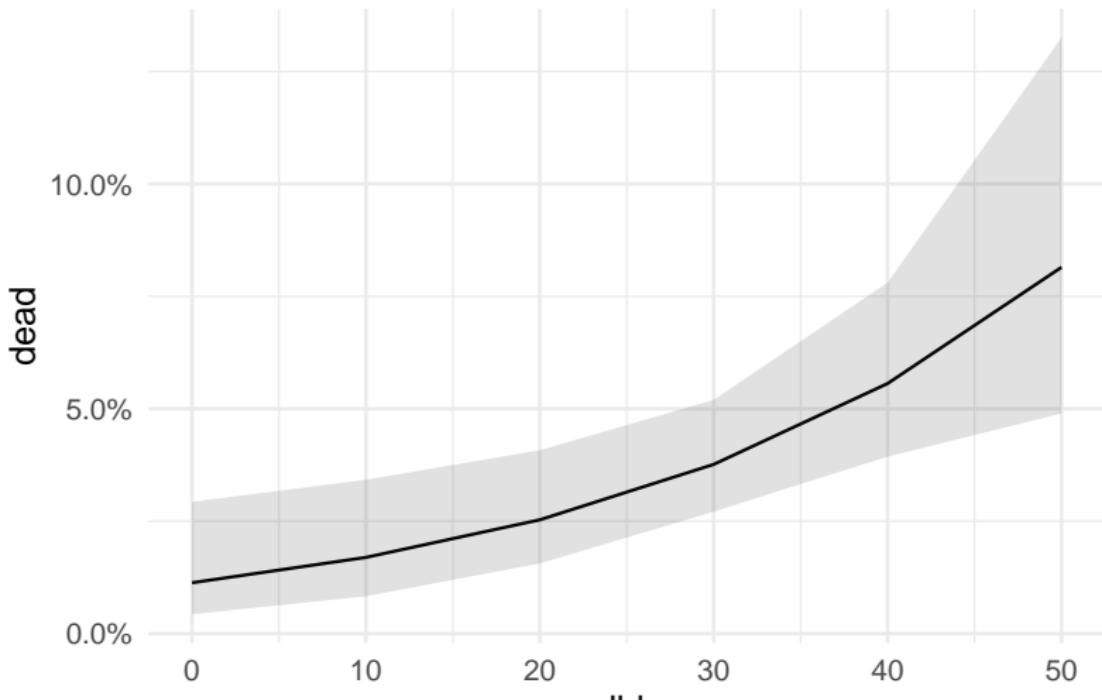
```
attr(,"class")
[1] "coef.mer"
```

## Visualising model: sjPlot

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

\$dbh

Predicted probabilities of dead



## Advantages of multilevel models

- ▶ Perfect for structured data (space-time)

## Advantages of multilevel models

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

## Advantages of multilevel models

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

## Advantages of multilevel models

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

## Advantages of multilevel models

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

## Formula syntax for different models

$y \sim x + (1 | group)$  # varying intercepts

$y \sim x + (1 + x | group)$  # varying intercepts and slopes

$y \sim x + (1 | group/subgroup)$  # nested

$y \sim x + (1 | group1) + (1 | group2)$  # varying intercepts, crossed

$y \sim x + (1 + x | group1) + (1 + x | group2)$  # varying intercepts and slopes, crossed

# GLMM FAQ

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

## More examples

- ▶ sleepstudy

END



Figure 9

Source code and materials:

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