

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

Francisco Rodriguez-Sanchez

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

Modern statistics are easier than this

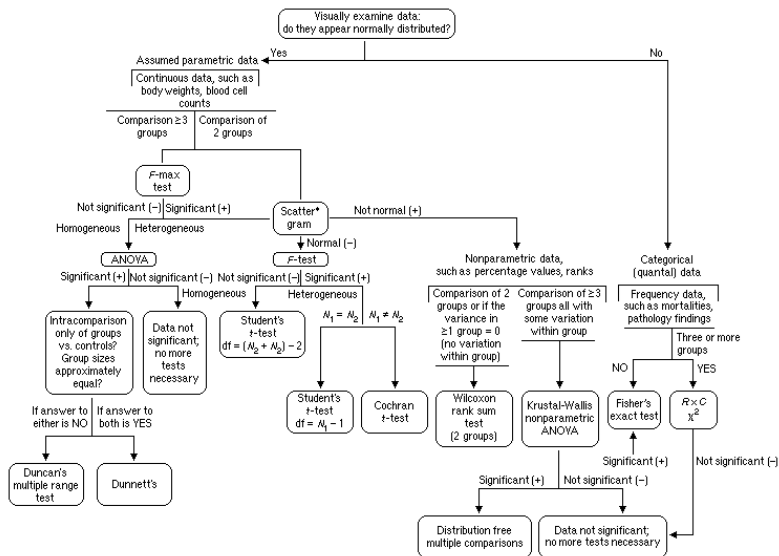
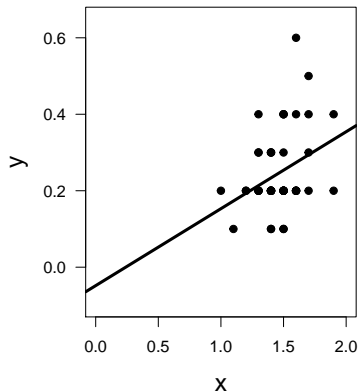


Figure 1:

# 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

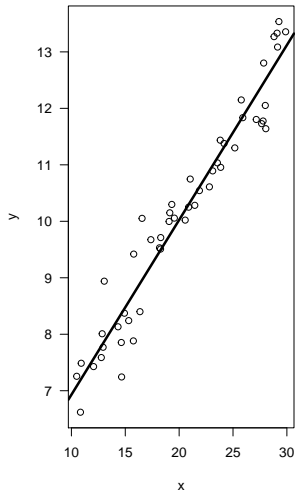
$b$  = slope

$\sigma$  = residual variation

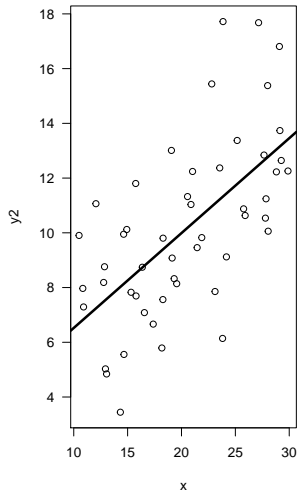
$\varepsilon$  = residuals

# Residual variation (error)

**small**



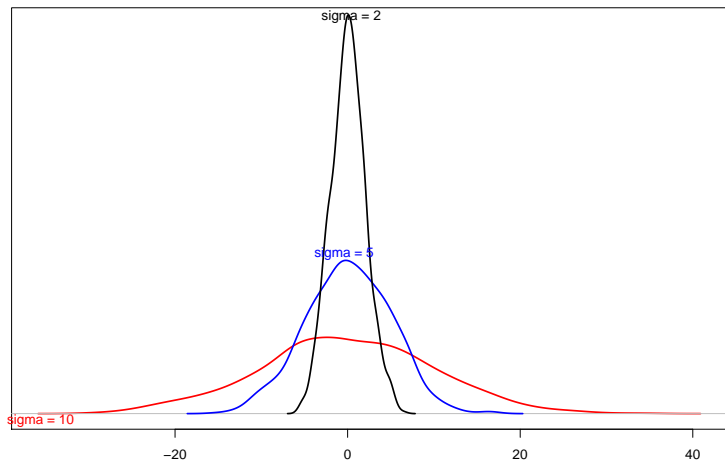
**large**



# Residual variation

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

Distribution of residuals



## In a Normal distribution

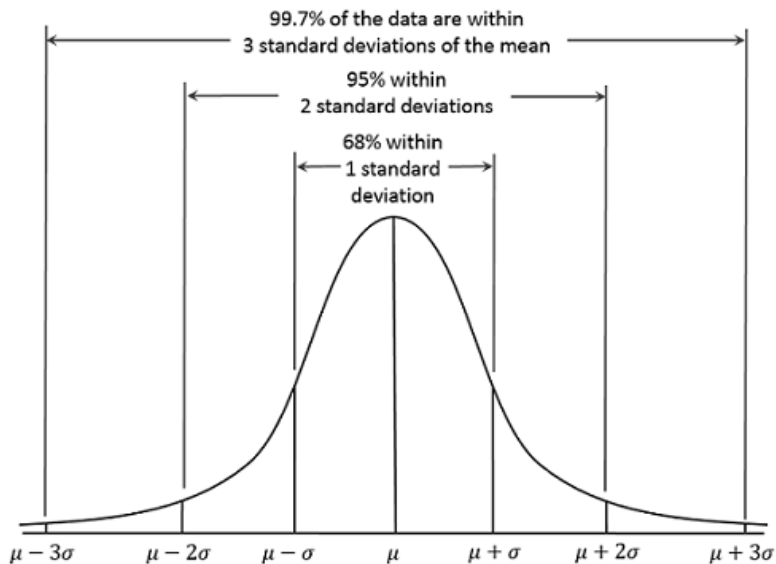


Figure 2:

## Quick refresher of linear models

- ▶ Download datasets from [http://bit.ly/DEAD\\_datasets](http://bit.ly/DEAD_datasets)



- ▶ Download datasets from `http://bit.ly/DEAD_datasets`
- ▶ Load iris data into R

- ▶ Download datasets from [http://bit.ly/DEAD\\_datasets](http://bit.ly/DEAD_datasets)
- ▶ Load iris data into R
- ▶ Q: What is the relationship between petal width and length in *Iris setosa*?

# Iris dataset

```
str(setosa)
```

```
'data.frame':  50 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ..
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ..
 $ Species      : Factor w/ 3 levels "setosa","versicolor",...: 1
```

Always plot your data first!

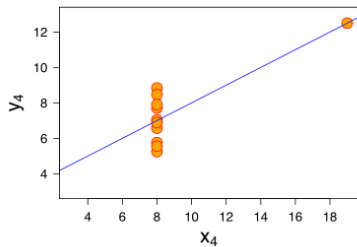
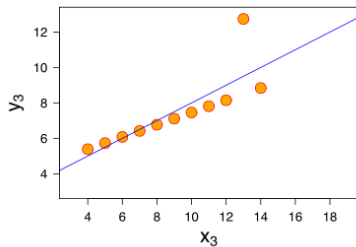
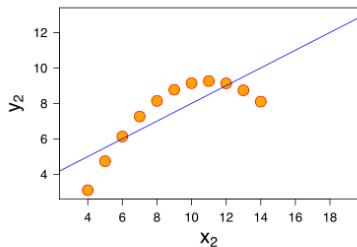
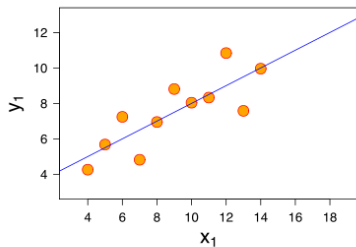
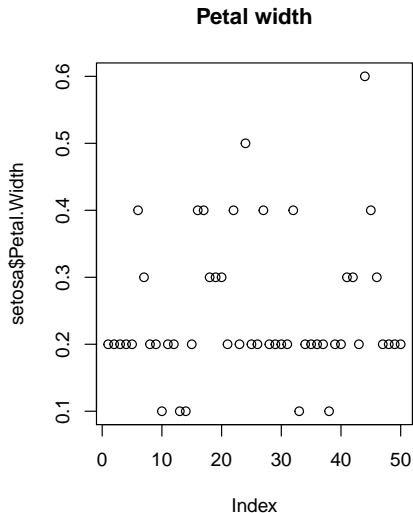


Figure 3:

# Exploratory Data Analysis (EDA)

## Outliers

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



## Outliers impact on regression

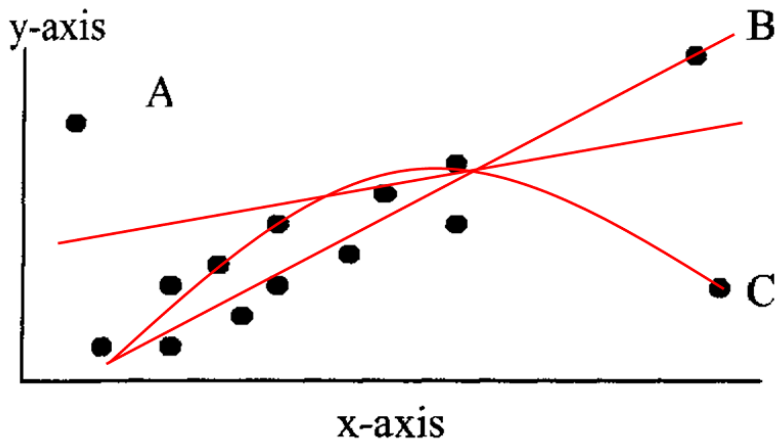
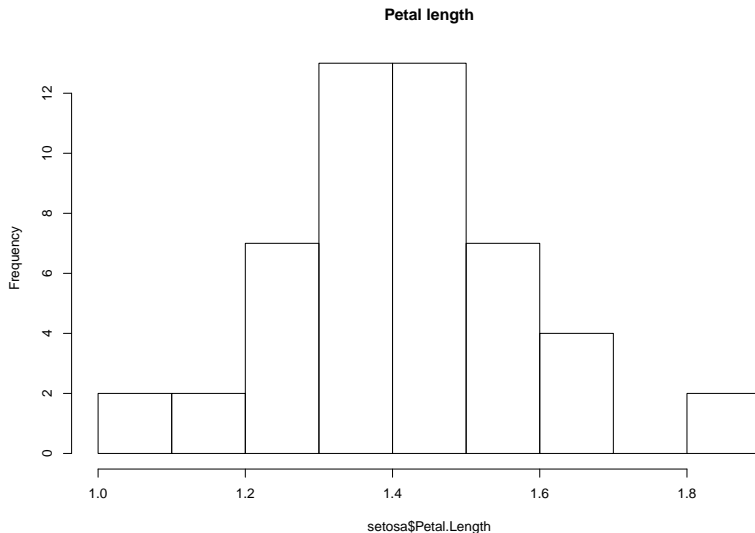


Figure 4:

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

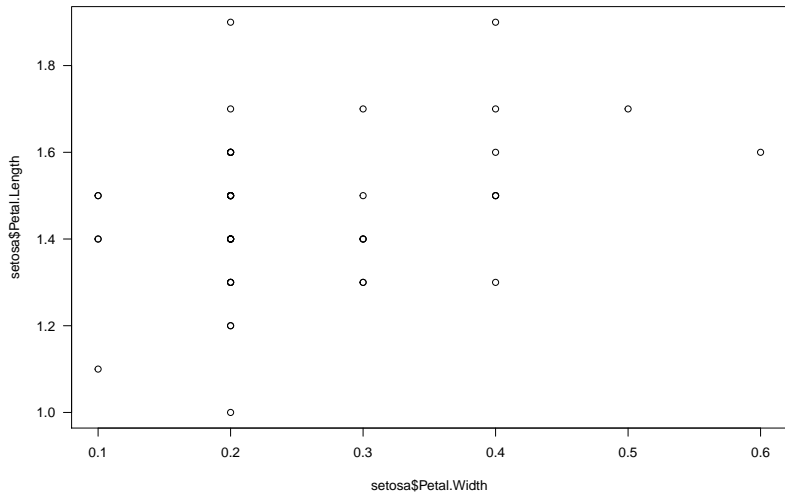
# Histogram

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



# Scatterplot

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





Now fit model

Hint: `lm`

## Now fit model

Hint: `lm`

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

## What does this mean?

Call:

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

Residuals:

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

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.32756	0.05996	22.141	<2e-16 ***
Petal.Width	0.54649	0.22439	2.435	0.0186 *

---

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

Residual standard error: 0.1655 on 48 degrees of freedom

Multiple R-squared: 0.11, Adjusted R-squared: 0.09144

F-statistic: 5.931 on 1 and 48 DF, p-value: 0.01864

## Retrieving model coefficients

```
coef(m1)
```

```
(Intercept) Petal.Width  
  1.3275634    0.5464903
```

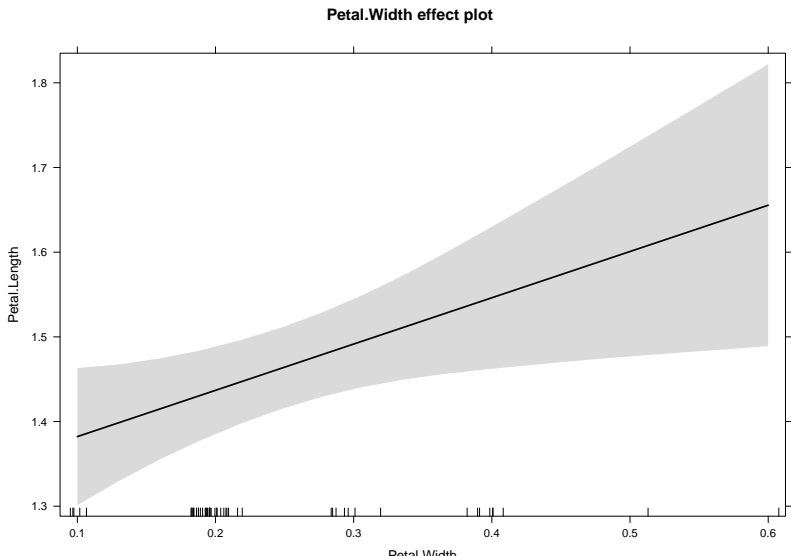
## Confidence intervals

```
confint(m1)
```

	2.5 %	97.5 %
(Intercept)	1.20700694	1.4481199
Petal.Width	0.09531905	0.9976615

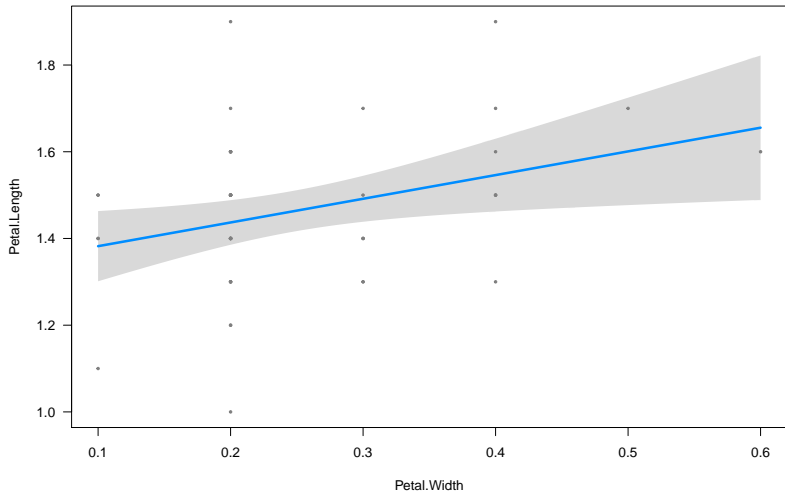
# Plot effects

```
library(effects)  
plot(allEffects(m1))
```



## Plot model (visreg)

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



# Linear model assumptions

- ▶ Linearity (transformations, GAM...)



# Linear model assumptions

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

# Linear model assumptions

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

# Linear model assumptions

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

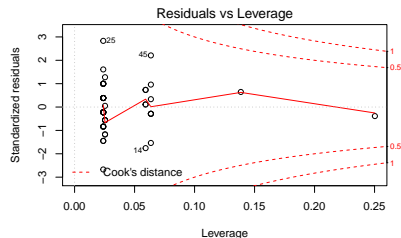
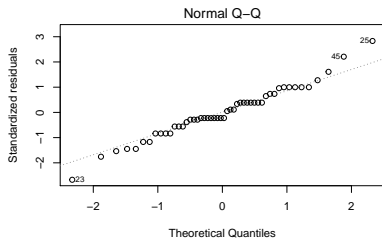
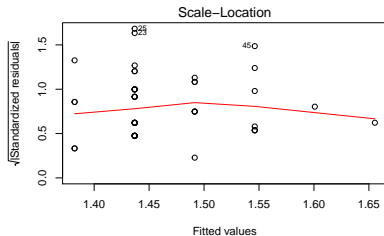
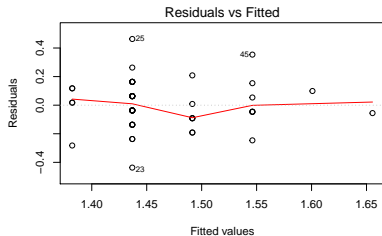
# Linear model assumptions

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

# Linear model assumptions

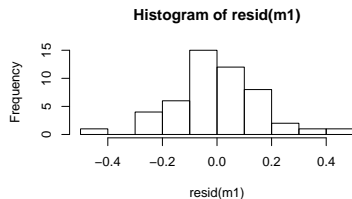
- ▶ Linearity (transformations, GAM...)
- ▶ Residuals:
  - ▶ Independent
  - ▶ Equal variance
  - ▶ Normal
- ▶ No measurement error in predictors

# Model checking: residuals



# Are residuals normal?

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



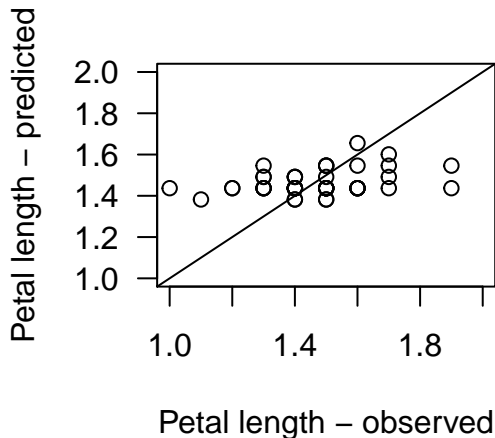
```
lm(formula = Petal.Length ~ Petal.Width)
      coef.est coef.se
(Intercept)  1.33    0.06
Petal.Width  0.55    0.22
---
n = 50, k = 2
residual sd = 0.17, R-Squared = 0.75
```

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

## How good is the model in predicting petal length?

Observed vs Predicted values: use fitted.

```
plot(setosa$Petal.Length, fitted(m1), xlab = "Petal length - obs
```





## Using fitted model for prediction

Q: Expected petal length if width = 0.39?

## Using fitted model for prediction

Q: Expected petal length if width = 0.39?

```
predict(m1, data.frame(Petal.Width = c(0.39)), se.fit = TRUE)
```

```
$fit
```

```
1
```

```
1.540695
```

```
$se.fit
```

```
[1] 0.03990149
```

```
$df
```

```
[1] 48
```

```
$residual.scale
```

```
[1] 0.1655341
```

# Important functions

- ▶ `plot`

# Important functions

- ▶ `plot`
- ▶ `summary`

# Important functions

- ▶ `plot`
- ▶ `summary`
- ▶ `coef`

# Important functions

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

# Important functions

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

# Important functions

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



# Important functions

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

# Important functions

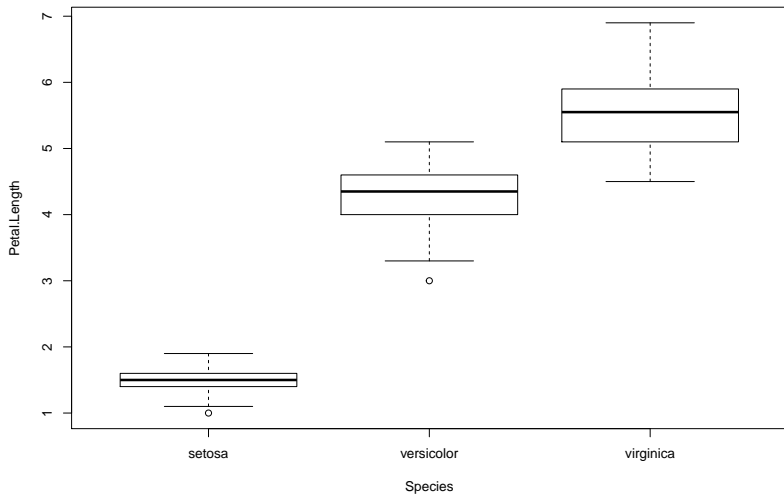
- ▶ `plot`
- ▶ `summary`
- ▶ `coef`
- ▶ `confint`
- ▶ `fitted`
- ▶ `resid`
- ▶ `allEffects`
- ▶ `predict`

Categorical predictors (factors)

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

First, a plot:

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



## Linear model with categorical predictors

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

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

## Model

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

Call:

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

Residuals:

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

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.46200	0.06086	24.02	<2e-16 ***
Speciesversicolor	2.79800	0.08607	32.51	<2e-16 ***
Speciesvirginica	4.09000	0.08607	47.52	<2e-16 ***

---

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

Residual standard error: 0.4303 on 147 degrees of freedom

Multiple R-squared: 0.9414, Adjusted R-squared: 0.9406

## Alternatively, no intercept

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

Call:

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

Residuals:

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

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
Speciessetosa	1.46200	0.06086	24.02	<2e-16 ***
Speciesversicolor	4.26000	0.06086	70.00	<2e-16 ***
Speciesvirginica	5.55200	0.06086	91.23	<2e-16 ***

---

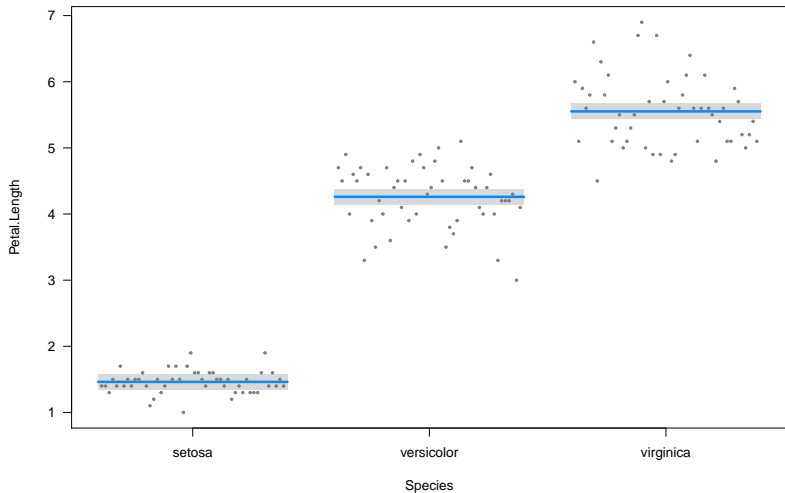
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4303 on 147 degrees of freedom

Multiple R-squared: 0.9895, Adjusted R-squared: 0.9892

# Petal length differences across 3 *Iris* species

```
visreg(m3)
```





# Are differences statistically significant?

Compare CIs

```
summary(allEffects(m3))
```

```
model: Petal.Length ~ Species - 1
```

Species effect

Species

setosa	versicolor	virginica
1.462	4.260	5.552

Lower 95 Percent Confidence Limits

Species

setosa	versicolor	virginica
1.341729	4.139729	5.431729

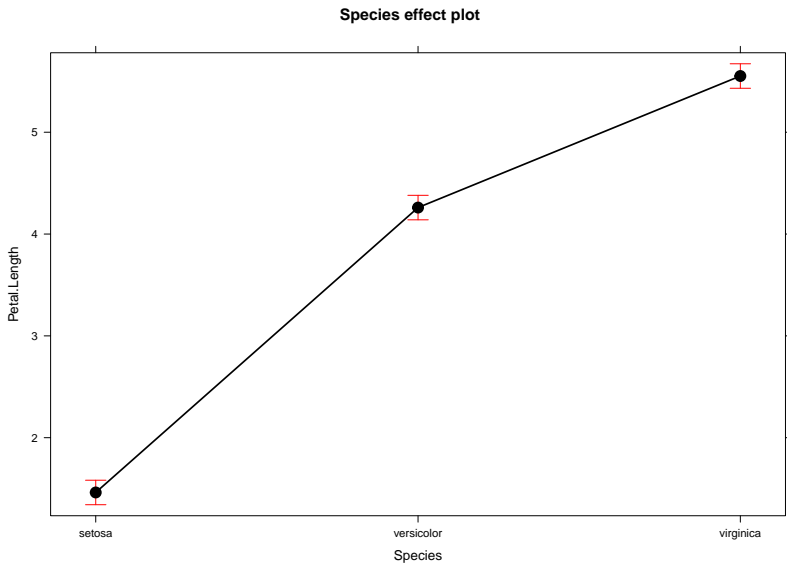
Upper 95 Percent Confidence Limits

Species

setosa	versicolor	virginica
1.582271	4.380271	5.672271

# Plotting effects

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



Combining continuous and categorical predictors

## Predicting *Iris* petal length according to species and petal width

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

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

$$y_i = a + b_{\text{versicolor}} + c_{\text{virginica}} + d \cdot \text{PetalWidth}_i + \varepsilon_i$$

## Predicting *Iris* petal length according to species and petal width

Call:

```
lm(formula = Petal.Length ~ Species + Petal.Width, data = iris)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-1.02977	-0.22241	-0.01514	0.18180	1.17449

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.21140	0.06524	18.568	< 2e-16 ***
Speciesversicolor	1.69779	0.18095	9.383	< 2e-16 ***
Speciesvirginica	2.27669	0.28132	8.093	2.08e-13 ***
Petal.Width	1.01871	0.15224	6.691	4.41e-10 ***

---

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

Residual standard error: 0.3777 on 146 degrees of freedom

Multiple R-squared: 0.9551. Adjusted R-squared: 0.9542

## Generalised Linear Models (GLMs)

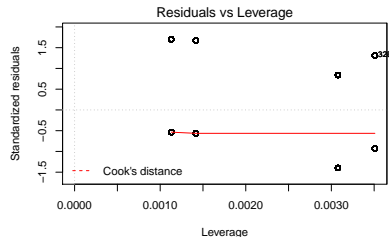
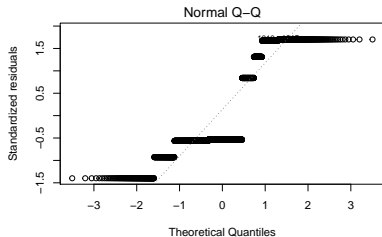
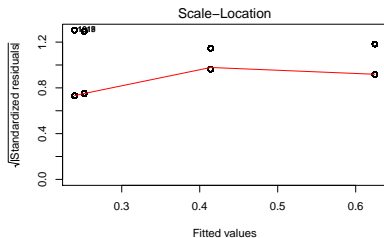
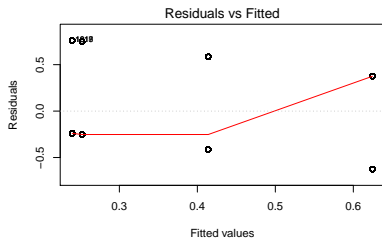
## Q: Survival of passengers on the Titanic ~ Class

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

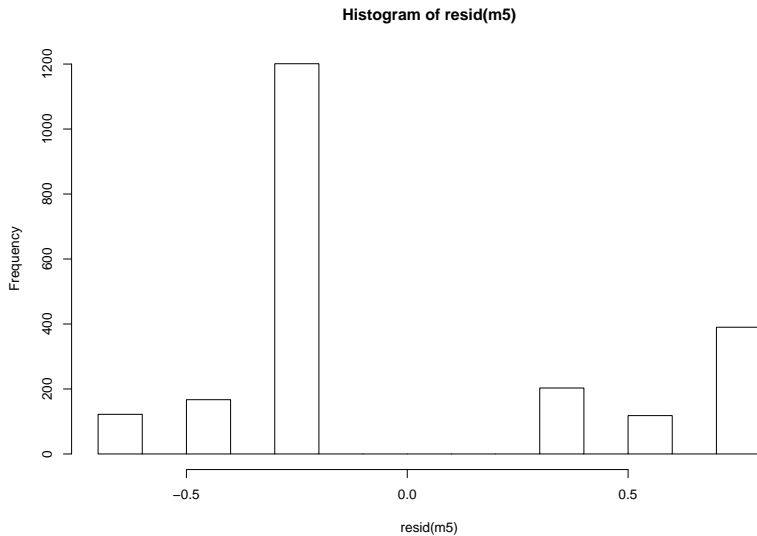
# Let's fit linear model:

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





# Weird residuals!



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

- ▶ Binary variables (0/1)

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

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

# Generalised Linear Models

1. **Response variable** - distribution family

# Generalised Linear Models

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

# Generalised Linear Models

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

- ▶ Bernoulli - Binomial
- ▶ Poisson

# Generalised Linear Models

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

- ▶ Bernoulli - Binomial
- ▶ Poisson
- ▶ Gamma

# Generalised Linear Models

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

- ▶ Bernoulli - Binomial
- ▶ Poisson
- ▶ Gamma
- ▶ etc



# Generalised Linear Models

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

# Generalised Linear Models

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

# Generalised Linear Models

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

# Generalised Linear Models

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

- ▶ Bernoulli - 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

- ▶ Bernoulli - 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

- ▶ Bernoulli - Binomial
- ▶ Poisson
- ▶ Gamma
- ▶ etc

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

## 3. **Link function**

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

# The modelling process

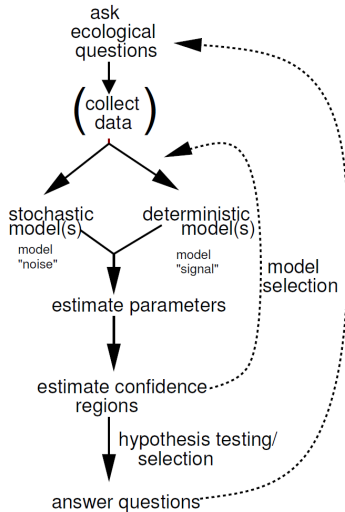


Figure 1.5 Flow of the modeling process.

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

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

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

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



## 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

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

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

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

## Back to survival of Titanic passengers

How many passengers travelled in each class?

## Back to survival of Titanic passengers

How many passengers travelled in each class?

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

crew	first	second	third
885	325	285	706

## Back to survival of Titanic passengers

How many passengers travelled in each class?

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

crew	first	second	third
885	325	285	706

How many survived?

## Back to survival of Titanic passengers

How many passengers travelled in each class?

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

crew	first	second	third
885	325	285	706

How many survived?

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

crew	first	second	third
212	203	118	178

## Back to survival of Titanic passengers

How many passengers travelled in each class?

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

crew	first	second	third
885	325	285	706

How many survived?

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

crew	first	second	third
212	203	118	178

What proportion survived in each class?

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

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

## Back to survival of Titanic passengers (dplyr)

Passenger survival according to class

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

Source: local data frame [8 x 3]

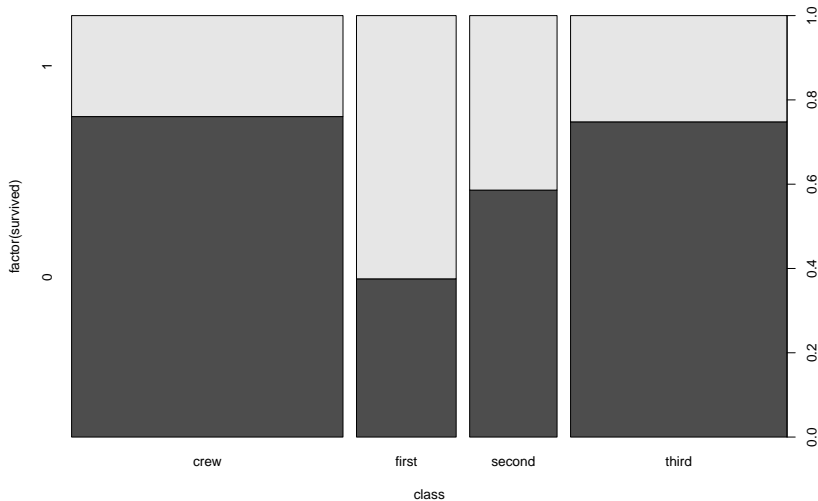
Groups: class [?]

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

Or summarise(group by(titanic, class, survived), count =

Or graphically...

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





## 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 ***
classecond	0.80785	0.14375	5.620	1.91e-08 ***
classtthird	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)

# 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
```

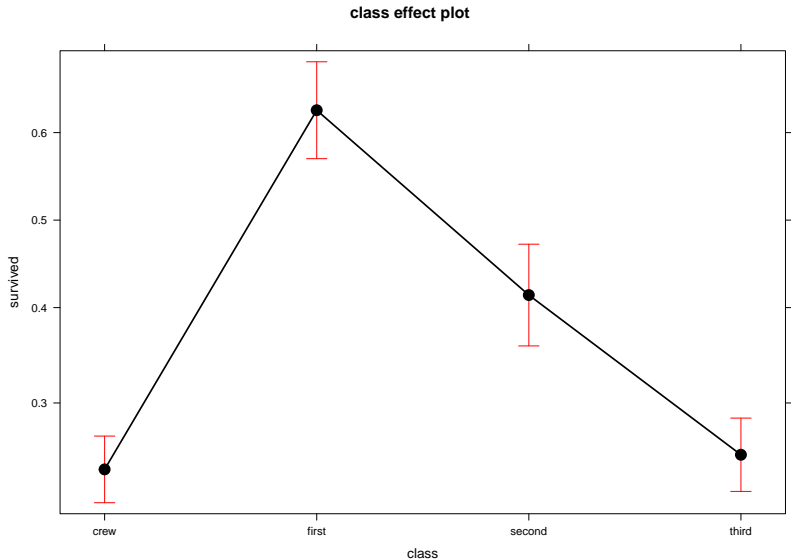
```
class effect
```

```
class
```

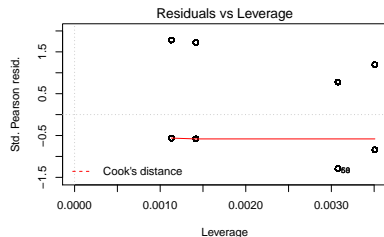
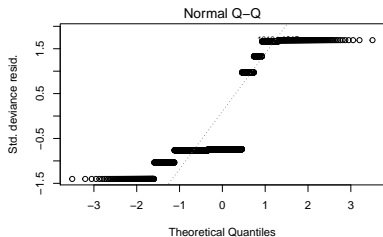
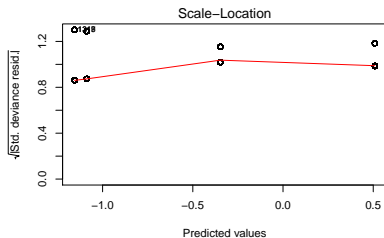
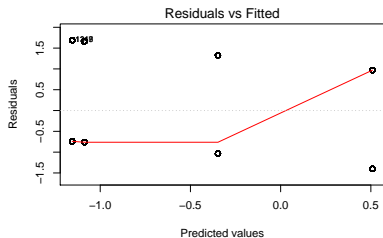
	crew	first	second	third
	0.2395480	0.6246154	0.4140351	0.2521246

# Effects plot

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



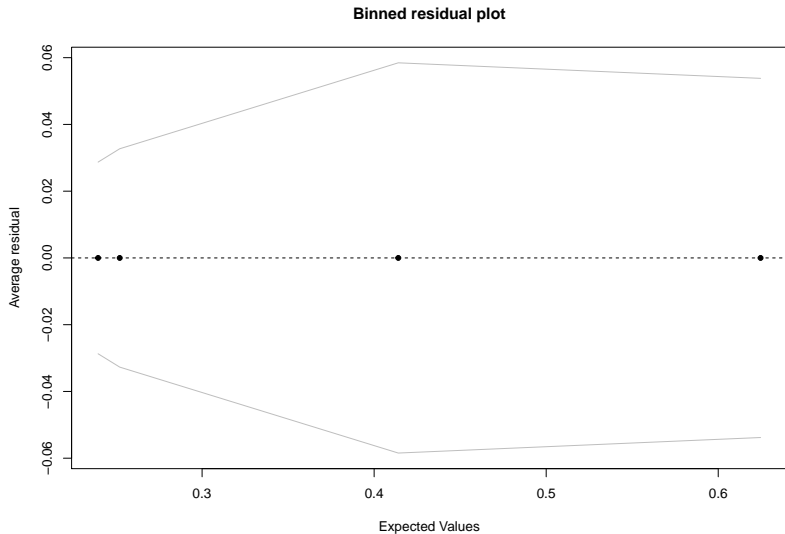
# Logistic regression: model checking



Not very useful.

## Binned residual plots for logistic regression

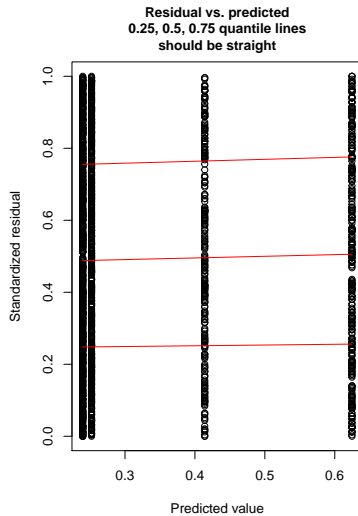
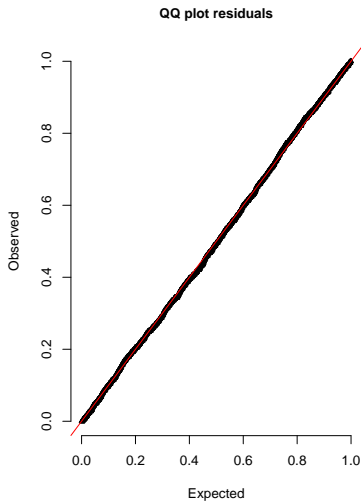
```
predvals <- predict(tit.glm, type="response")  
arm::binnedplot(predvals, titanic$survived - predvals)
```



# Residual diagnostics with DHARMA

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

DHARMA scaled residual plots





# Recapitulating

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

# Recapitulating

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

# Recapitulating

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

# Recapitulating

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

# Recapitulating

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

# Recapitulating

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

# Recapitulating

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

# Recapitulating

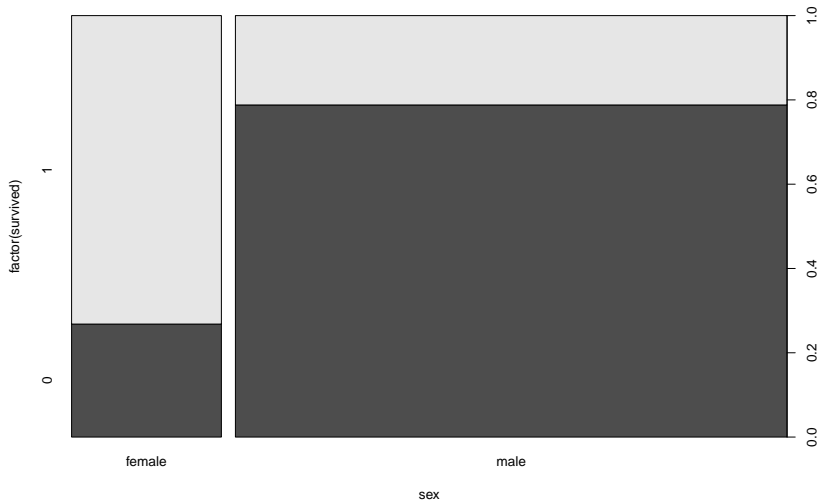
1. Import data: `read.table` or `read.csv`
2. Check data: `summary`
3. Plot data: `plot`
4. Fit model: `glm`. Don't forget to specify family!
5. Examine models: `summary`
6. Use `plogis` to apply back-transformation (*invlogit*) to parameter estimates (`coef`). Alternatively, use `allEffects` from `effects`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



Q: Did men have higher survival than women?

## Plot first

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



## Fit model

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

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

# Effects

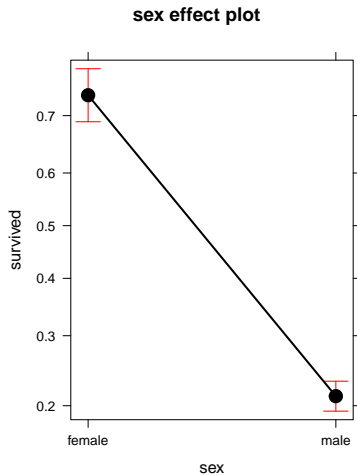
```
model: survived ~ sex
```

```
sex effect
```

```
sex
```

```
female    male
```

```
0.7319149 0.2120162
```



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

## Let's look at the data

tapply

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

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

Mmmm...

## Fit model with both factors (interactions)

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

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

	coef.est	coef.se
--	----------	---------

(Intercept)	1.90	0.62
-------------	------	------

classfirst	1.67	0.80
------------	------	------

classecond	0.07	0.69
------------	------	------

classthird	-2.06	0.64
------------	-------	------

sexmale	-3.15	0.62
---------	-------	------

classfirst:sexmale	-1.06	0.82
--------------------	-------	------

classecond:sexmale	-0.64	0.72
--------------------	-------	------

classthird:sexmale	1.74	0.65
--------------------	------	------

---

n = 2201, k = 8

residual deviance = 2163.7, null deviance = 2769.5 (difference

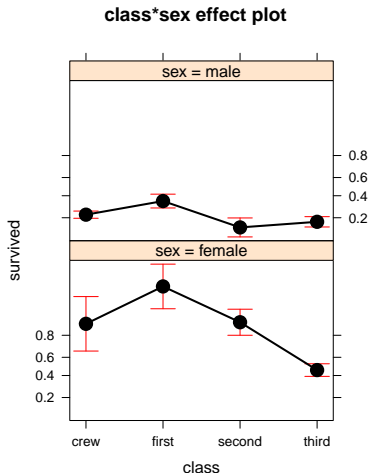
# Effects

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

```
class*sex effect
```

```
sex
```

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



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



Logistic regression for proportion data

## Read Titanic data in different format

Read Titanic\_prop.csv data.

X	Class	Sex	Age	No
Min. : 1.00	1st :4	Female:8	Adult:8	Min. : 0.00
1st Qu.: 4.75	2nd :4	Male :8	Child:8	1st Qu.: 0.00
Median : 8.50	3rd :4			Median : 8.50
Mean : 8.50	Crew:4			Mean : 93.12
3rd Qu.:12.25				3rd Qu.: 96.25
Max. :16.00				Max. :670.00

Yes
Min. : 0.00
1st Qu.: 9.50
Median : 14.00
Mean : 44.44
3rd Qu.: 75.25
Max. :192.00

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

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

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

Call:

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

Deviance Residuals:

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

Coefficients:

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

---

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

(Dispersion parameter for binomial family taken to be 1)

# Effects

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

```
Class effect
```

```
Class
```

	1st	2nd	3rd	Crew
	0.6246154	0.4140351	0.2521246	0.2395480

Compare with former model based on raw data:

```
model: survived ~ class
```

```
class effect
```

```
class
```

	crew	first	second	third
	0.2395480	0.6246154	0.4140351	0.2521246

Same results!

Logistic regression with continuous predictors

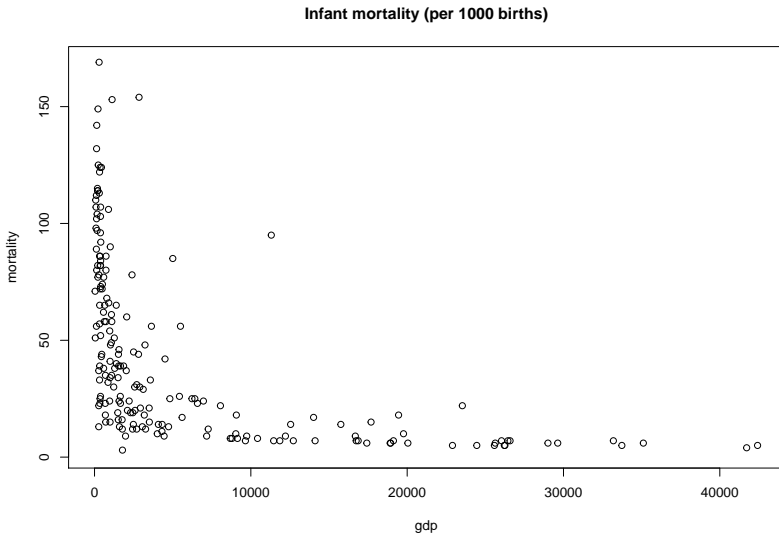
Example dataset: GDP and infant mortality

Read UN\_GDP\_infantmortality.csv.

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

# EDA

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



## 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
```

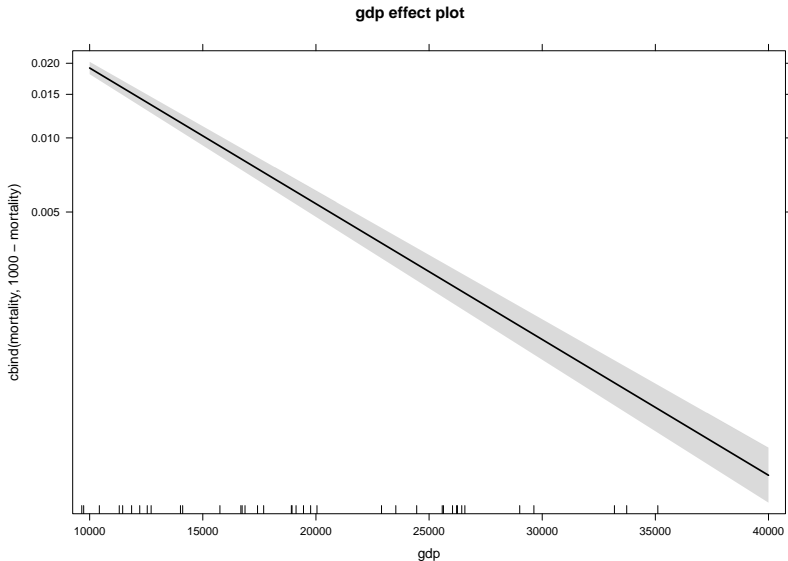
```
gdp effect
```

```
gdp
```

	10000	20000	30000	40000
	0.0191438829	0.0054028095	0.0015096074	0.0004206154

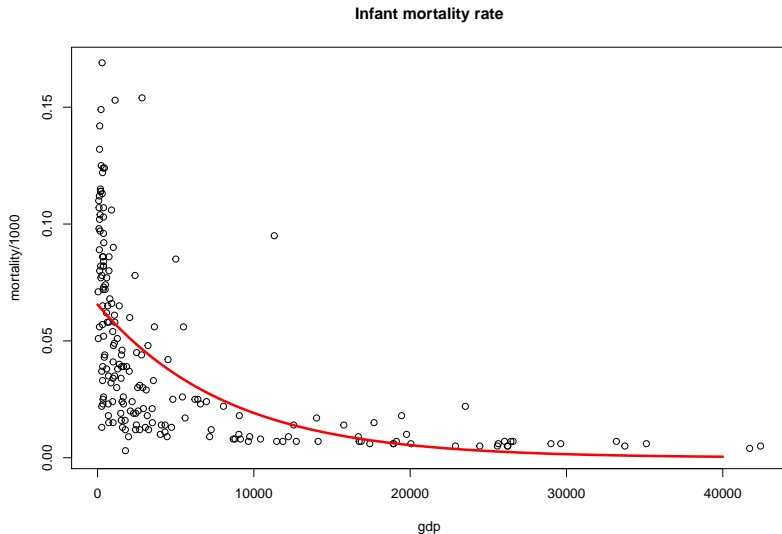
# Effects plot

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



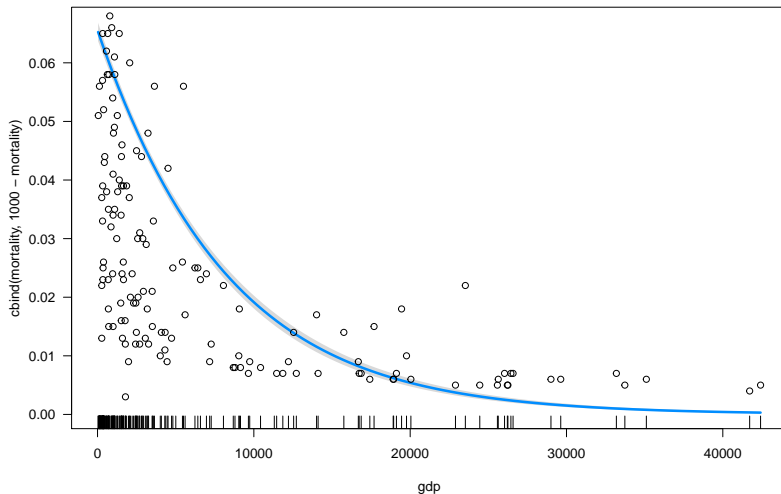
## Plot model and data

```
plot(mortality/1000 ~ gdp, data = gdp, main = "Infant mortality  
curve(plogis(coef(gdp.glm)[1] + coef(gdp.glm)[2]*x), from = 0, t
```



## Plot model using visreg:

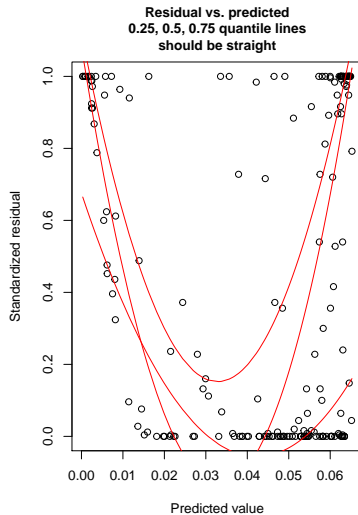
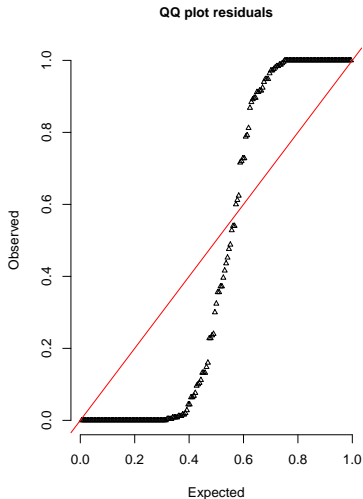
```
visreg(gdp.glm, scale = "response")  
points(mortality/1000 ~ gdp, data = gdp)
```



# Residuals diagnostics with DHARMa

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

DHARMa scaled residual plots



## Overdispersion

# Testing for overdispersion (DHARMa)

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

Overdispersion test via comparison to simulation under  $H_0$

```
data:  simres  
dispersion = 20.571, p-value < 2.2e-16  
alternative hypothesis: overdispersion
```

## Overdispersion in logistic regression with proportion data

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

Call:

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

Deviance Residuals:

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

Coefficients:

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

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

(Dispersion parameter for quasibinomial family taken to be 20.79)



## Mean estimates do not change after accounting for overdispersion

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

```
gdp effect
```

```
gdp
```

	10000	20000	30000	40000
	0.0191438829	0.0054028095	0.0015096074	0.0004206154

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

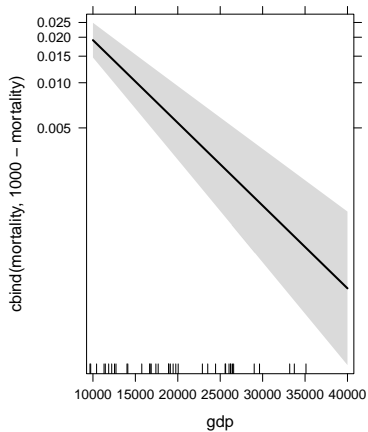
```
gdp effect
```

```
gdp
```

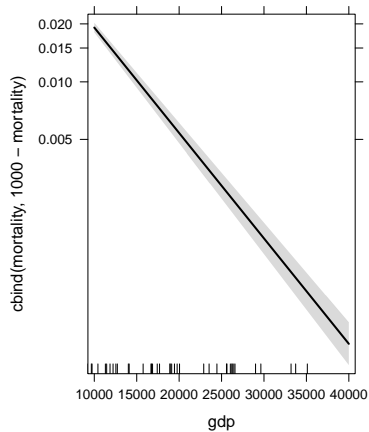
	10000	20000	30000	40000
	0.0191438829	0.0054028095	0.0015096074	0.0004206154

But standard errors (uncertainty) do!

**gdp effect plot**

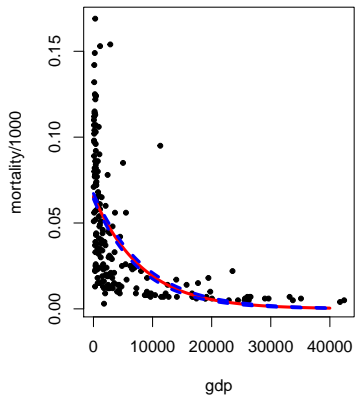


**gdp effect plot**

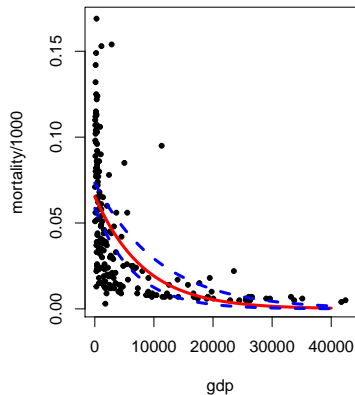


# Plot model and data

**Binomial**



**Quasibinomial**



# Overdispersion

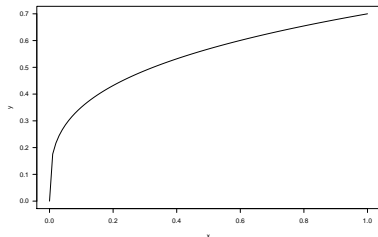
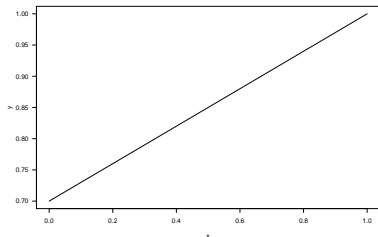
Whenever you fit logistic regression to **proportion** data, check family quasibinomial.

# 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.



GLMs for count data: Poisson regression

# Types of response variable

- ▶ Gaussian: 1m

# Types of response variable

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



# Types of response variable

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

# Poisson regression

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

Then

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

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

# Poisson regression

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

Then

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

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

## Example dataset: Seedling counts in 0.5 m2 quadrats

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

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

light

Min. : 2.571
1st Qu.:26.879
Median :47.493
Mean :47.959
3rd Qu.:67.522
Max. :99.135

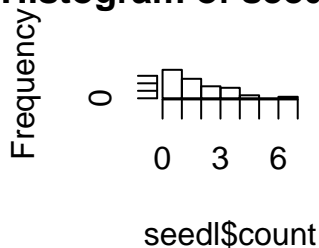
# EDA

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

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

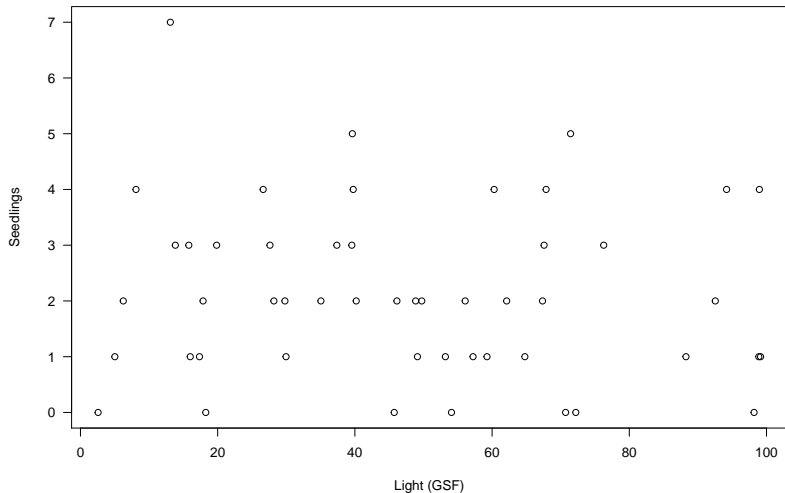
```
hist(seed1$count)
```

## Histogram of seed1\$count



## Q: Relationship between Nseedlings and light?

```
plot(seed1$light, seed1$count, las = 1, xlab = "Light (GSF)", ylab = "Seedlings")
```



## Let's fit model (Poisson regression)

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

Call:

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

Deviance Residuals:

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

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	0.881805	0.188892	4.668	3.04e-06 ***
light	-0.002576	0.003528	-0.730	0.465

---

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

(Dispersion parameter for poisson family taken to be 1)

# Interpreting Poisson regression output

Parameter estimates (log scale):

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

(Intercept)	light
0.881805022	-0.002575656

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

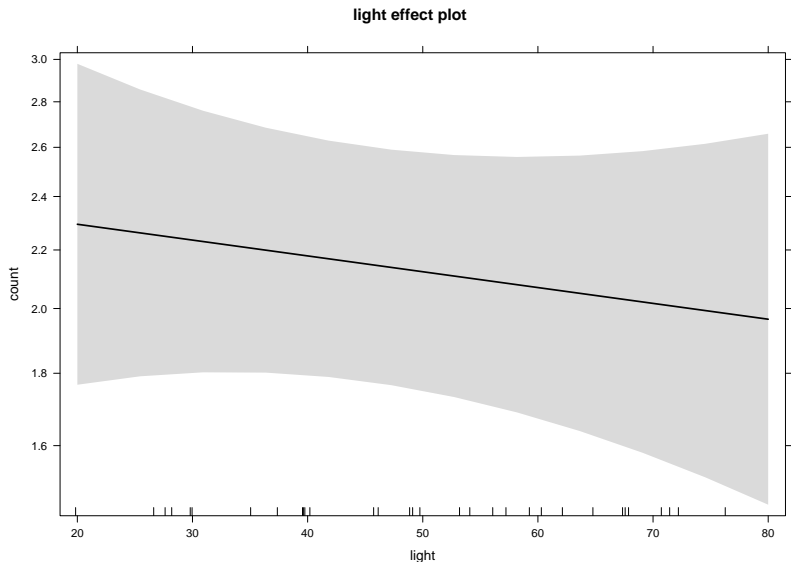
```
exp(coef(seedl.glm))
```

(Intercept)	light
2.4152554	0.9974277



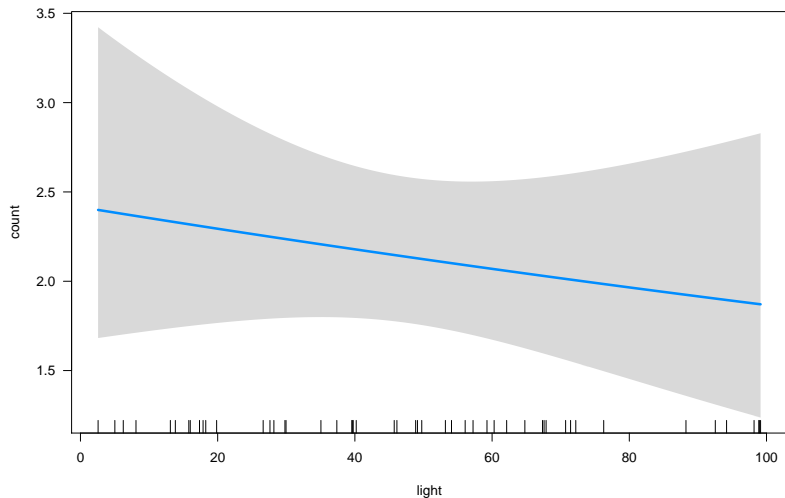
## So what's the relationship between Nseedlings and light?

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

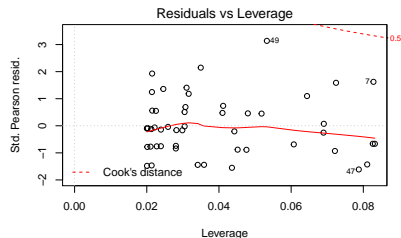
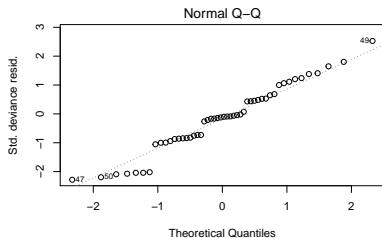
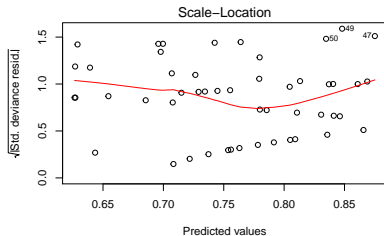
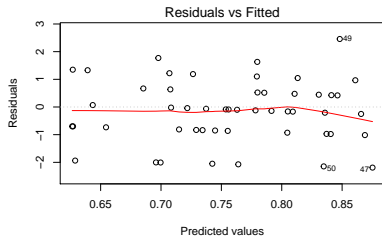


## Using visreg

```
visreg(seed1.glm, scale = "response")
```

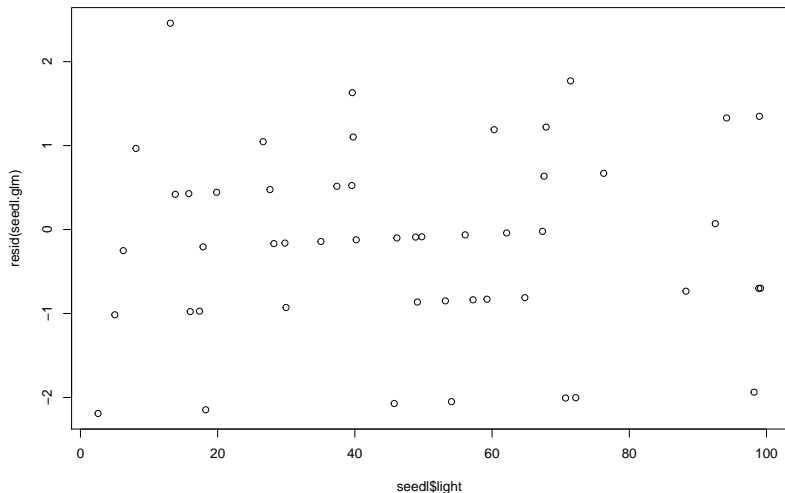


# Poisson regression: model checking



## Is there pattern of residuals along predictor?

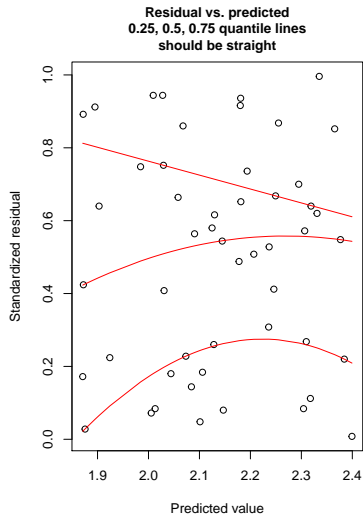
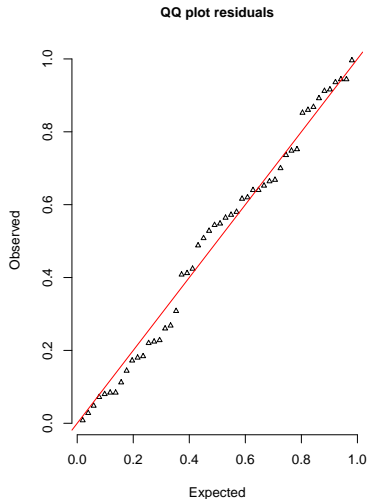
```
plot(seed1$light, resid(seed1.glm))
```



# Residuals diagnostics with DHARMA

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

DHARMA scaled residual plots



Poisson regression: Overdispersion

## Always check overdispersion with count data

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

Overdispersion test via comparison to simulation under  $H_0$

```
data:  simres  
dispersion = 1.1271, p-value = 0.256  
alternative hypothesis: overdispersion
```

# Accounting for overdispersion in count data

Use family quasipoisson

Call:

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

Deviance Residuals:

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

Coefficients:

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

---

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

(Dispersion parameter for quasipoisson family taken to be 1.1349

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



## Mean estimates do not change after accounting for overdispersion

```
model: count ~ light
```

```
light effect
```

```
light
```

	20	40	60	80
	2.293988	2.178810	2.069414	1.965512

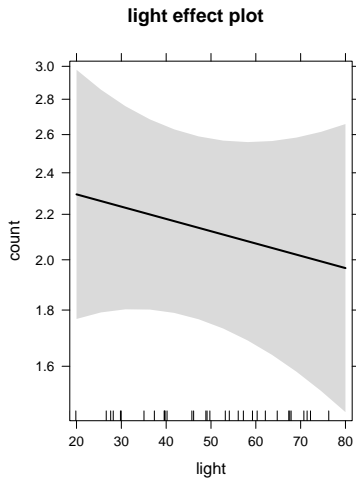
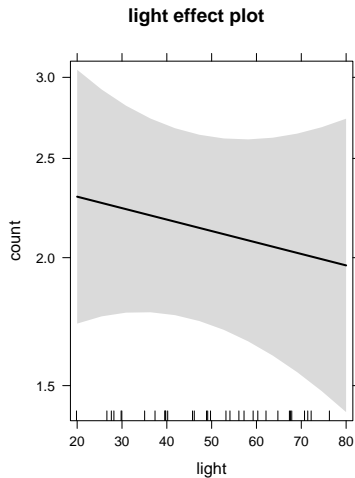
```
model: count ~ light
```

```
light effect
```

```
light
```

	20	40	60	80
	2.293988	2.178810	2.069414	1.965512

But standard errors may change



## Mixed / Multilevel Models

## 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 ***

---

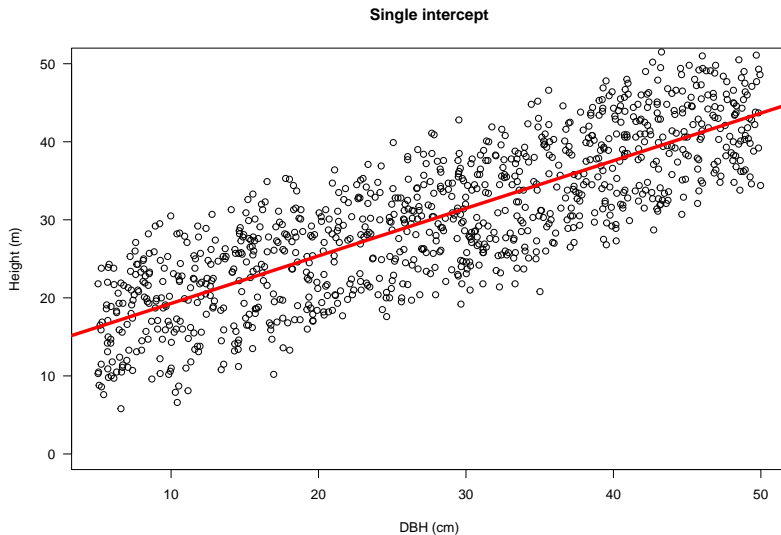
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.549 on 998 degrees of freedom

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

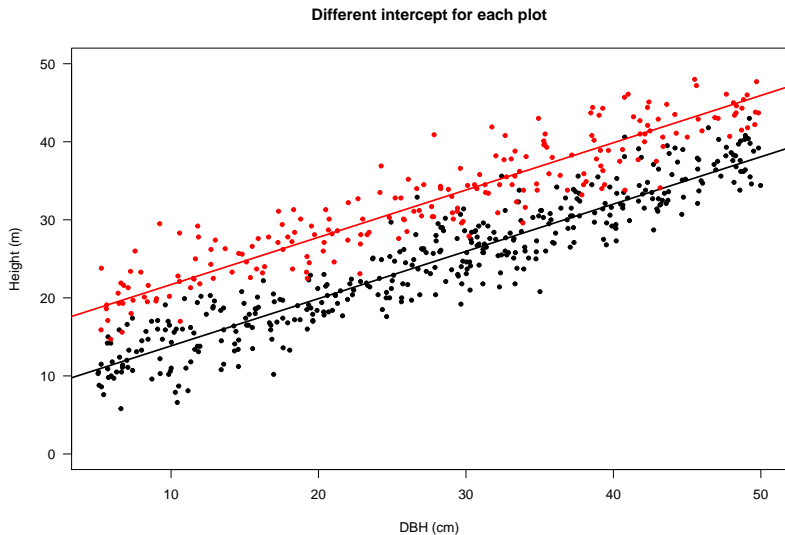
F-statistic: 2038 on 1 and 998 DF, p-value: < 2.2e-16

# There is only one intercept





# What if allometry varies among plots?



## 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

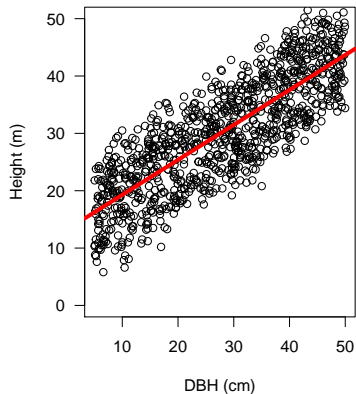
---

n = 1000, k = 11

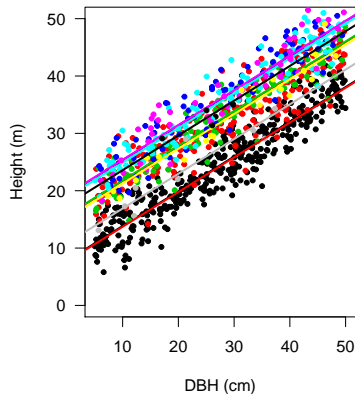
residual sd = 2.89, R-Squared = 0.91

# Single vs varying intercept

**Pooling all plots**



**Different intercept for each plot**



# Mixed models enable us to account for variability

## ► Varying intercepts

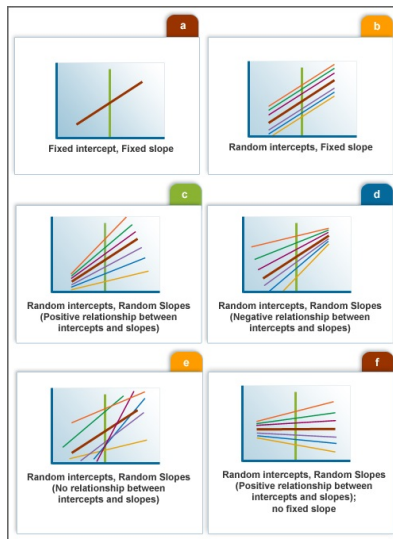


Figure 6:

# Mixed models enable us to account for variability

- ▶ Varying intercepts
- ▶ Varying slopes

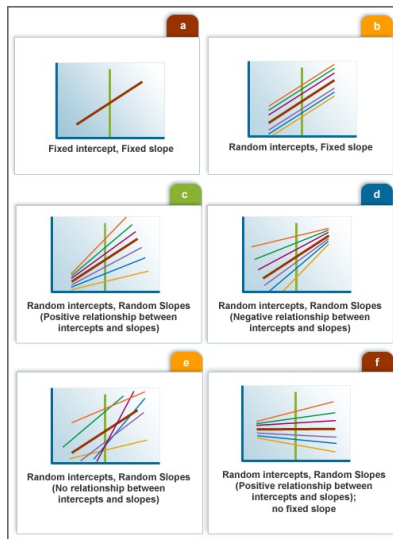


Figure 6:

## Mixed model with varying intercepts

$$y_i = a_j + bx_i + \varepsilon_i$$

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

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

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))`

# Fitting mixed/multilevel models

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

Linear mixed model fit by REML ['lmerMod']

Formula: height ~ dbh + (1 | plot)

Data: trees

REML criterion at convergence: 5007.6

Scaled residuals:

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

Random effects:

Groups	Name	Variance	Std.Dev.
plot	(Intercept)	19.834	4.454
Residual		8.325	2.885

Number of obs: 1000, groups: plot, 10

Fixed effects:

## Retrieve model coefficients

```
coef(mixed)
```

```
$plot
```

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

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

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

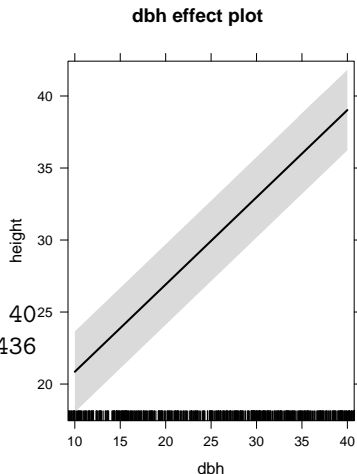
# Visualising model: allEffects

```
model: height ~ dbh
```

```
dbh effect
```

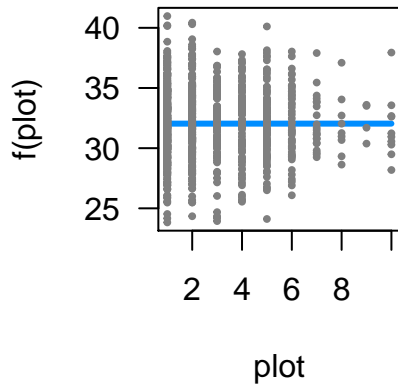
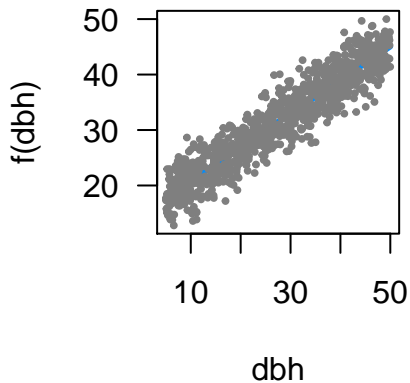
```
dbh
```

	10	20	30	40
	20.85471	26.91126	32.96781	39.02436



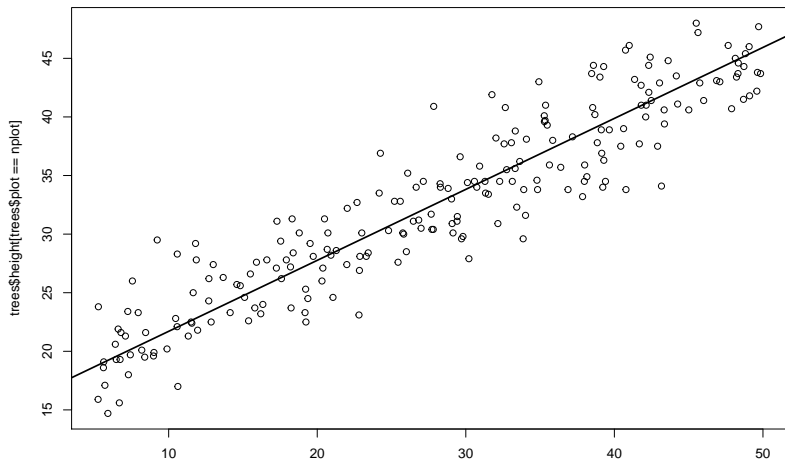


## Visualising model: visreg



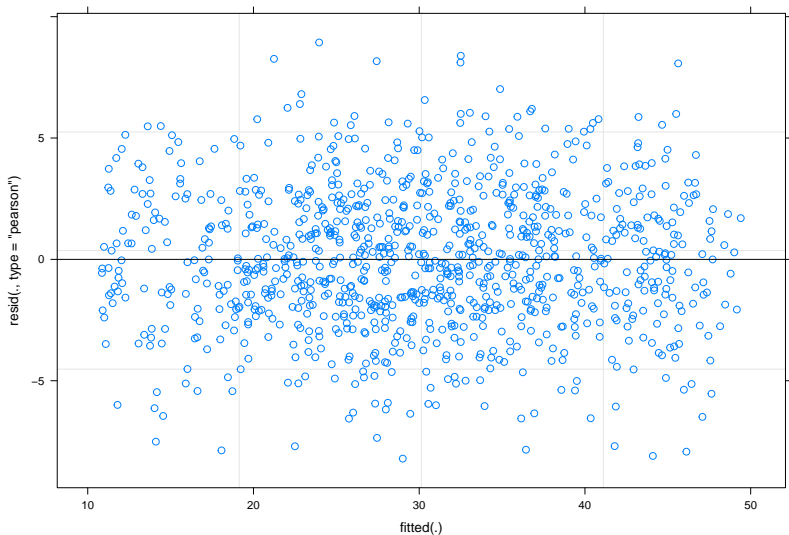
## Plotting regression for individual forest plots

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



# Checking residuals

```
plot(mixed)
```



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

Linear mixed model fit by REML ['lmerMod']

Formula: height ~ dbh + (1 + dbh | plot)

Data: trees

REML criterion at convergence: 5006.6

Scaled residuals:

Min	1Q	Median	3Q	Max
-2.87075	-0.65452	-0.02314	0.69251	3.10445

Random effects:

Groups	Name	Variance	Std.Dev.	Corr
plot	(Intercept)	2.092e+01	4.57422	
	dbh	1.287e-04	0.01135	-0.41
Residual		8.304e+00	2.88163	

Number of obs: 1000, groups: plot, 10

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	14.817566	1.478311	10.02



## Varying intercepts and slopes

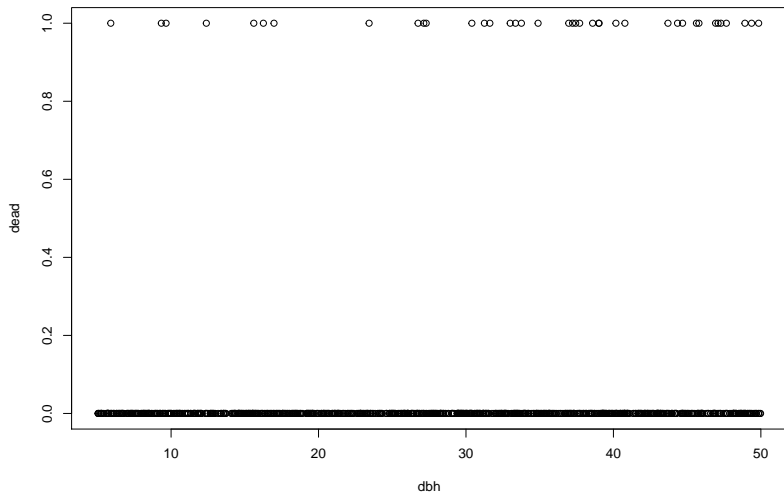
```
$plot
      (Intercept)      dbh
1      7.554578 0.6144452
2     15.966916 0.5942835
3     15.868970 0.6008673
4     19.321160 0.6031855
5     18.866368 0.6039353
6     19.355007 0.6038333
7     15.159257 0.6067449
8     10.965431 0.6080747
9     17.348836 0.6024600
10     7.769139 0.6109349
```

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

## Multilevel logistic regression

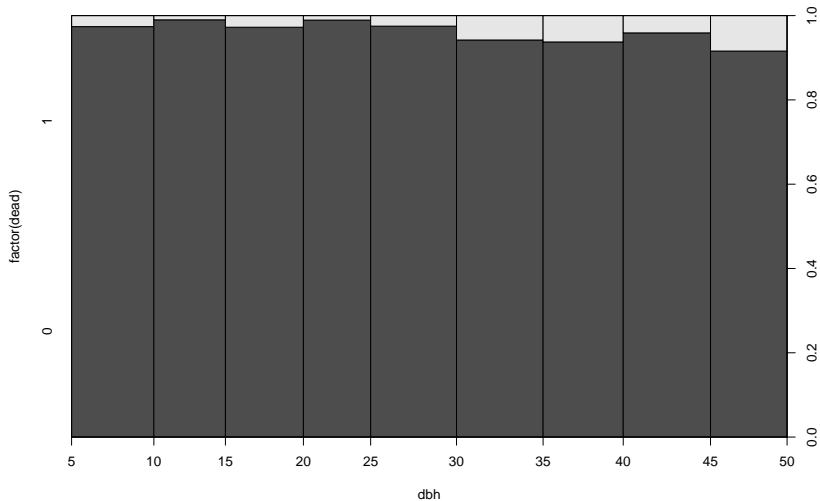
## Q: Relationship between tree size and mortality

```
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 =
```

Generalized linear mixed model fit by maximum likelihood (Laplace

Approximation) [glmerMod]

Family: binomial ( logit )

Formula: dead ~ dbh + (1 | plot)

Data: trees

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

Scaled residuals:

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

Random effects:

Groups	Name	Variance	Std.Dev.
plot	(Intercept)	0	0

Number of obs: 1000, groups: plot, 10

## Retrieve model coefficients

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

```
$plot
```

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

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

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



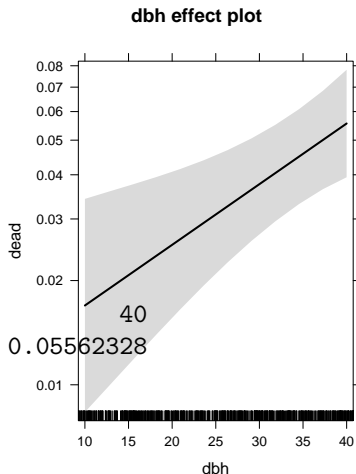
# Visualising model: allEffects

```
model: dead ~ dbh
```

```
dbh effect
```

```
dbh
```

	10	20	30	40
	0.01695545	0.02531581	0.03764063	0.05562328



END

:)

Source code and materials:

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



Figure 7: