

## Generalised Linear Models: Logistic regression

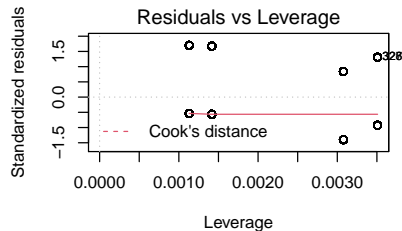
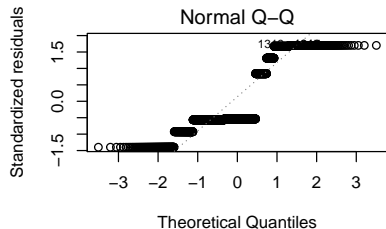
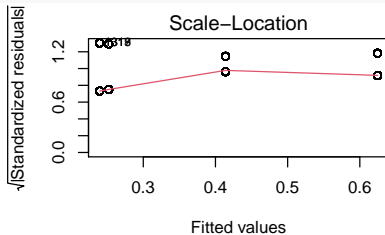
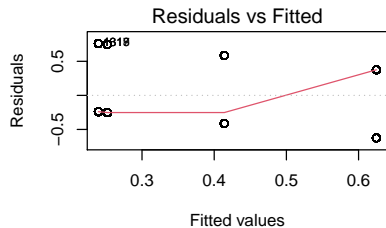
## Q: Survival of passengers on the Titanic ~ Class

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

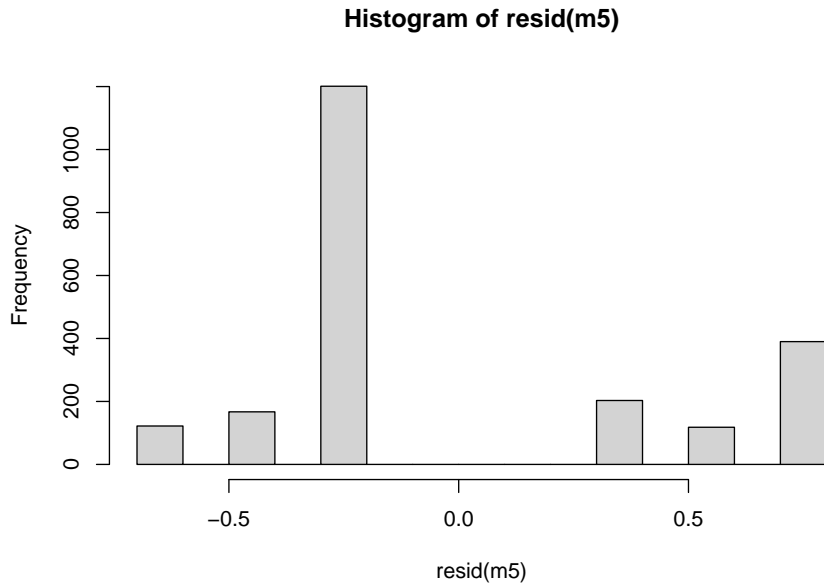
# Let's fit linear model:

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



null device

## 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

- ▶ 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

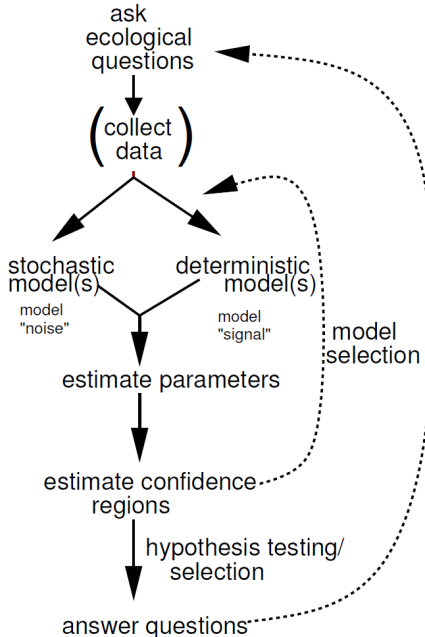
- ▶ Bernoulli - Binomial
- ▶ Poisson
- ▶ Gamma
- ▶ etc

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

## 3. **Link function**

- ▶ Gaussian: identity
- ▶ Binomial: logit, probit
- ▶ Poisson: log...
- ▶ See [family](#).

# The modelling process



# 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 survived in each class?

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

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

## Back to survival of Titanic passengers (dplyr)

Passenger survival according to class

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

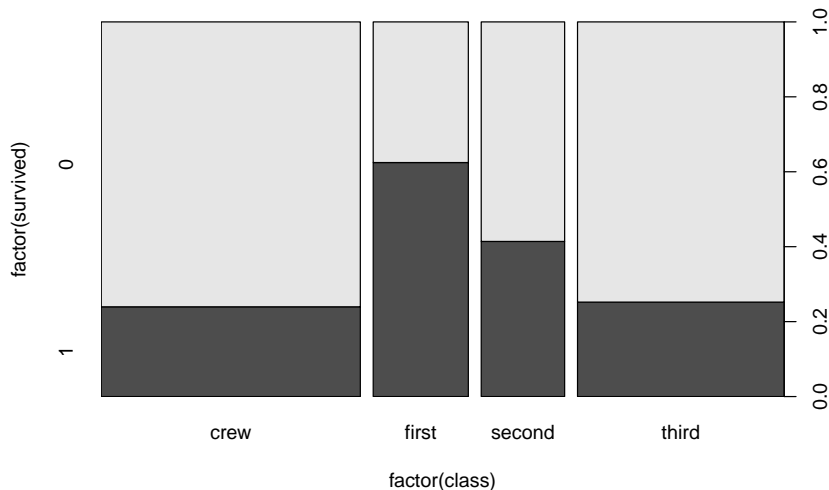
```
# A tibble: 8 x 3
```

```
# Groups:   class [4]
```

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

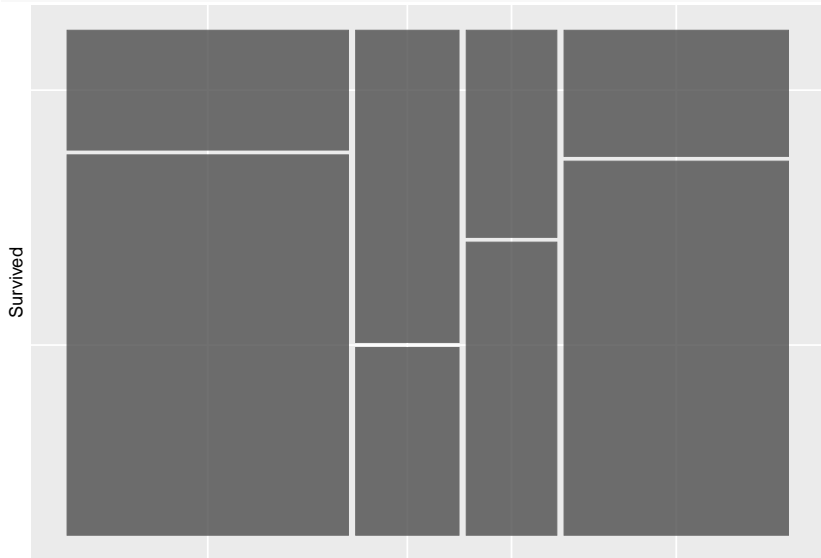
Or graphically...

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



# Mosaic plots (ggplot2)

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





## Fitting GLMs in R: glm

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

which corresponds to

$$\begin{aligned} \text{logit}(\text{Pr}(\text{survival})_i) &= a + b \cdot \text{class}_i \\ \text{logit}(\text{Pr}(\text{survival})_i) &= a + b_{\text{first}} + c_{\text{second}} + d_{\text{third}} \end{aligned}$$

# 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)

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	classecond	classtthird
-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
```

Must 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

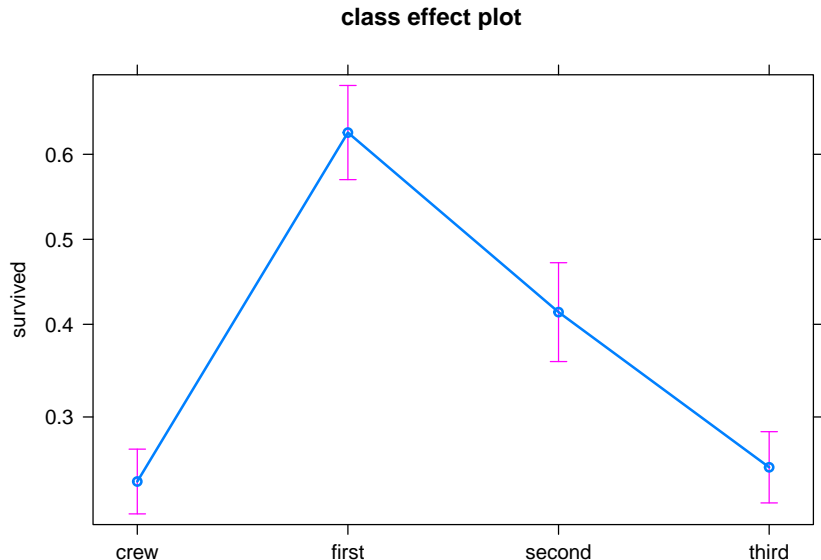
# Presenting model results

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

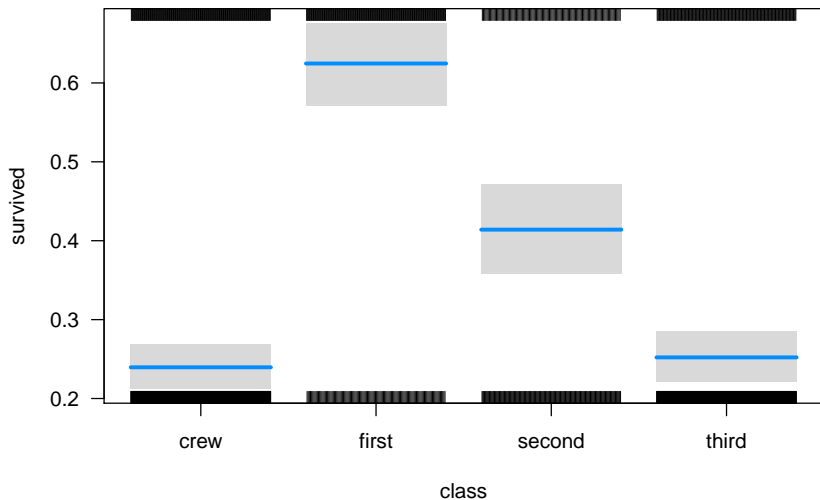
# Visualising model: effects package

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



## Visualising model: visreg package

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

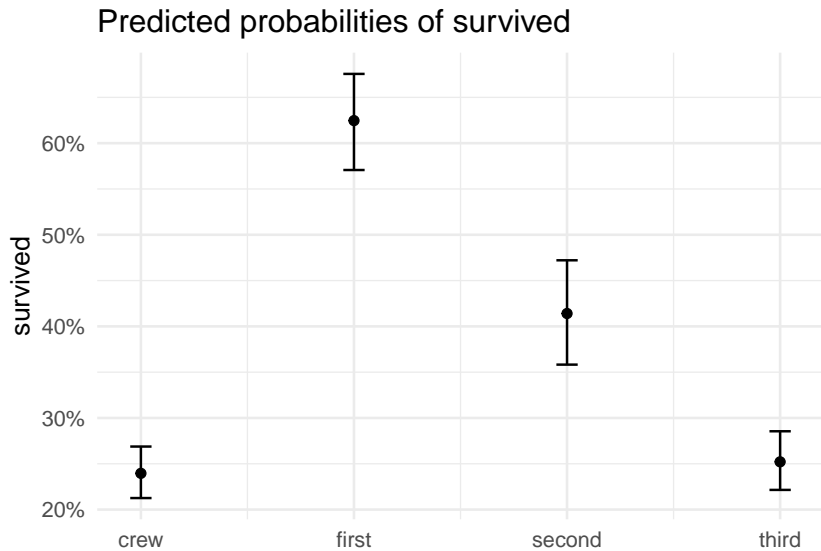




## Visualising model: sjPlot package

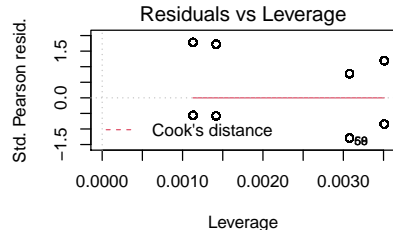
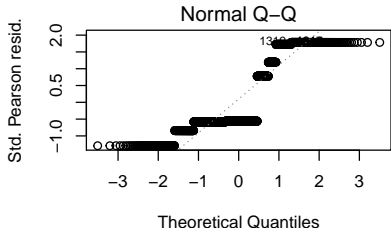
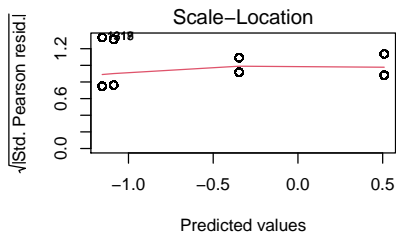
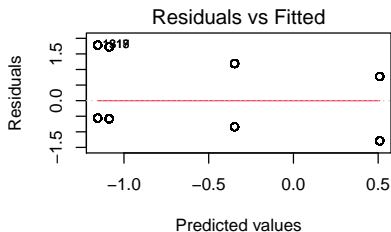
```
sjPlot::plot_model(tit.glm, type = "eff")
```

```
$class
```



# Logistic regression: model checking

Not very useful

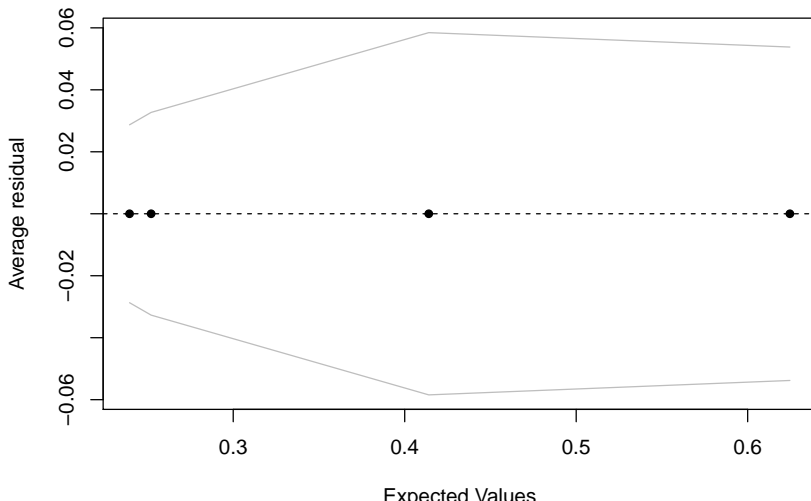


null device

## Binned residual plots for logistic regression

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

**Binned residual plot**

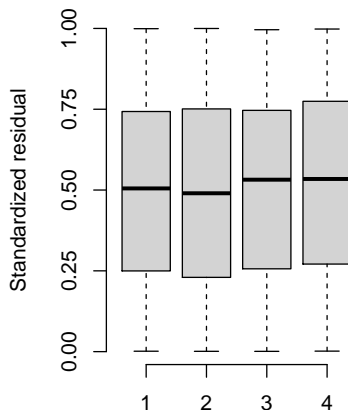
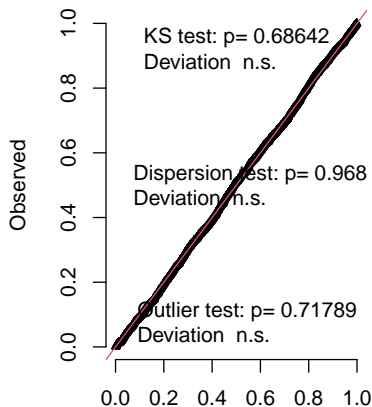


# Residual diagnostics with DHARMA

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

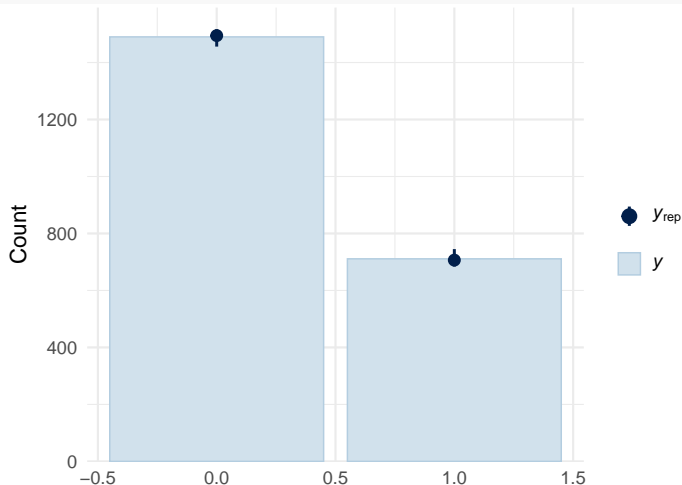
## DHARMA residual diagnostics

### QQ plot residuals



# Model checking with simulated data

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



# Pseudo R-squared for GLMs

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

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

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

# Recapitulating

1. **Visualise data**

# Recapitulating

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



# Recapitulating

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

# Recapitulating

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

# Recapitulating

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

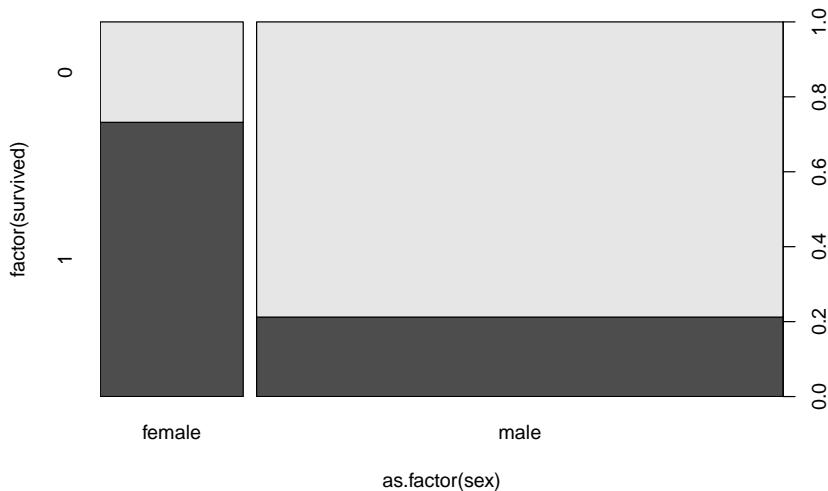
# Recapitulating

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

Q: Did men have higher survival than women?

## Plot first

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



## Fit model

Call:

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

Deviance Residuals:

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

Coefficients:

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

---

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

(Dispersion parameter for binomial family taken to be 1)

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

# Effects

```
model: survived ~ sex
```

```
sex effect
```

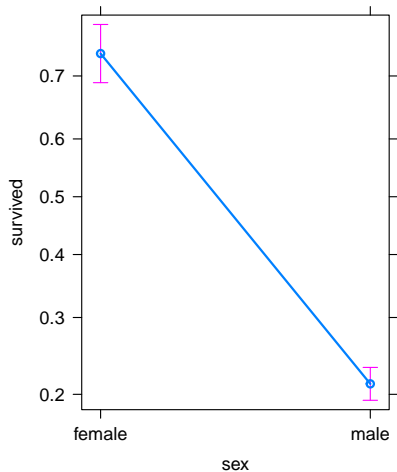
```
sex
```

```
female
```

```
male
```

```
0.7319149 0.2120162
```

**sex effect plot**





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

## Let's look at the data

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

```
, , = female
```

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

```
, , = male
```

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

Mmmm...

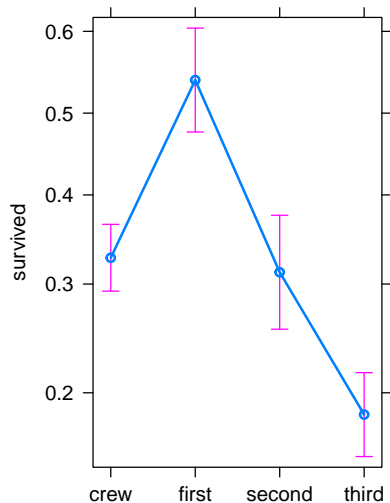
## Fit additive model with both factors

```
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
```

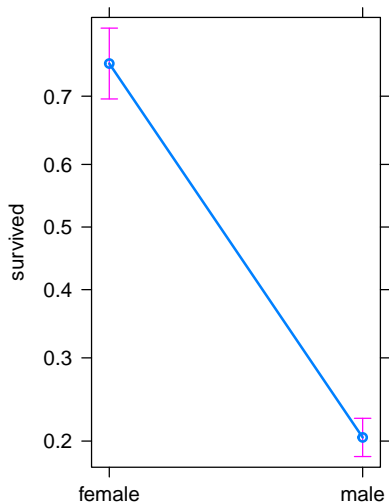
## Plot additive model

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

**class effect plot**



**sex effect plot**



## Fit model with both factors (interactions)

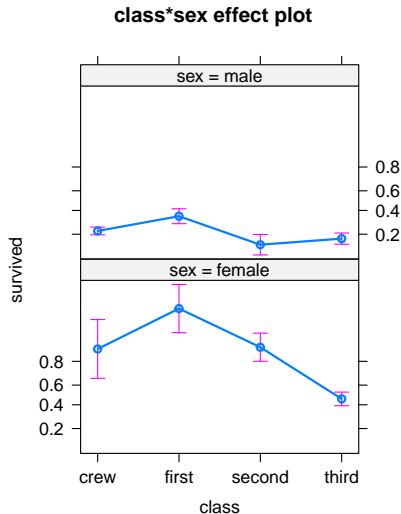
```
tit.sex.class <- glm(survived ~ class * sex, family = binomial,  
glm(formula = survived ~ class * sex, family = binomial, data =  
                                coef.est coef.se  
(Intercept)          1.90      0.62  
classfirst            1.67      0.80  
classecond            0.07      0.69  
classtthird          -2.06      0.64  
sexmale              -3.15      0.62  
classfirst:sexmale   -1.06      0.82  
classecond:sexmale  -0.64      0.72  
classtthird: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

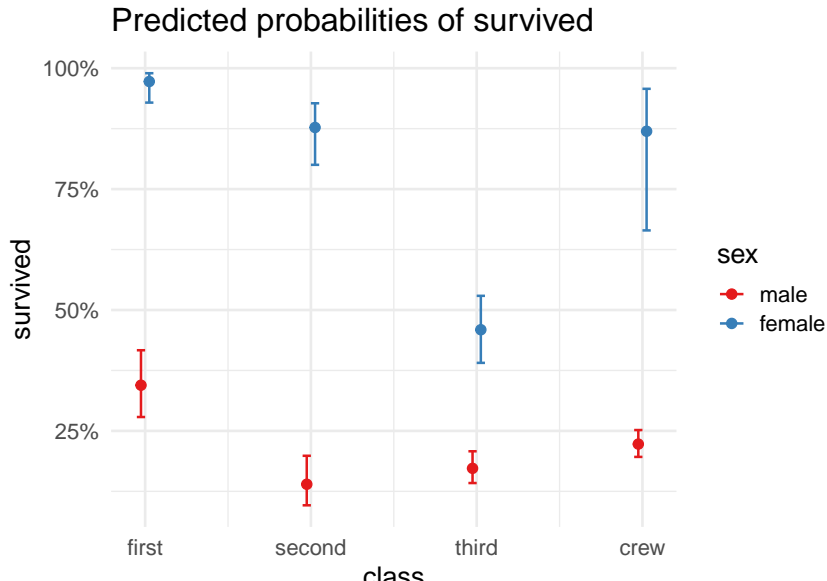
sex		
class	female	male
crew	0.8695652	0.2227378
first	0.9724138	0.3444444
second	0.8773585	0.1396648
third	0.4591837	0.1725490



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

## 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
```

Call:

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

Deviance Residuals:

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

Coefficients:

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

---

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

(Dispersion parameter for binomial family taken to be 1)

# Effects

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

Class effect

Class

	1st	2nd	3rd	Crew
	0.6246154	0.4140351	0.2521246	0.2395480

Compare with former model based on raw data:

```
model: survived ~ class
```

class effect

class

	crew	first	second	third
	0.2395480	0.6246154	0.4140351	0.2521246

Same results!

Logistic regression with continuous predictors

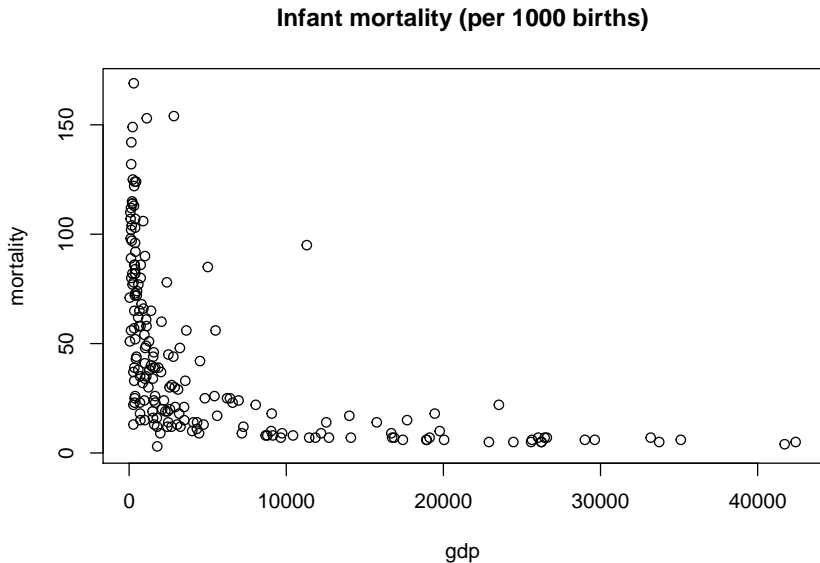
Example dataset: GDP and infant mortality

Read UN\_GDP\_infantmortality.csv.

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

# EDA

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



## Fit model

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

Call:

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

Deviance Residuals:

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

Coefficients:

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

---

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

(Dispersion parameter for binomial family taken to be 1)

# Effects

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

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

```
gdp effect
```

```
gdp
```

40

10000

20000

30000

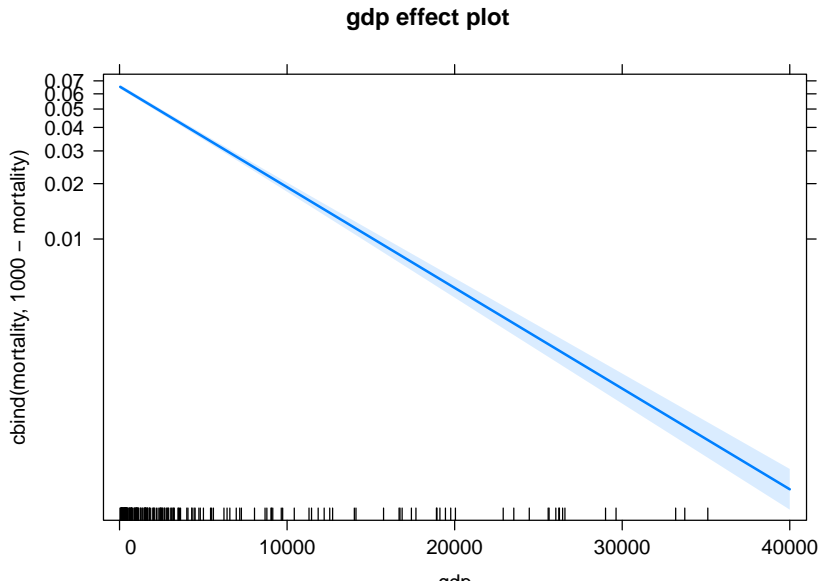
40000

0.0652177296 0.0191438829 0.0054028095 0.0015096074 0.0004206154



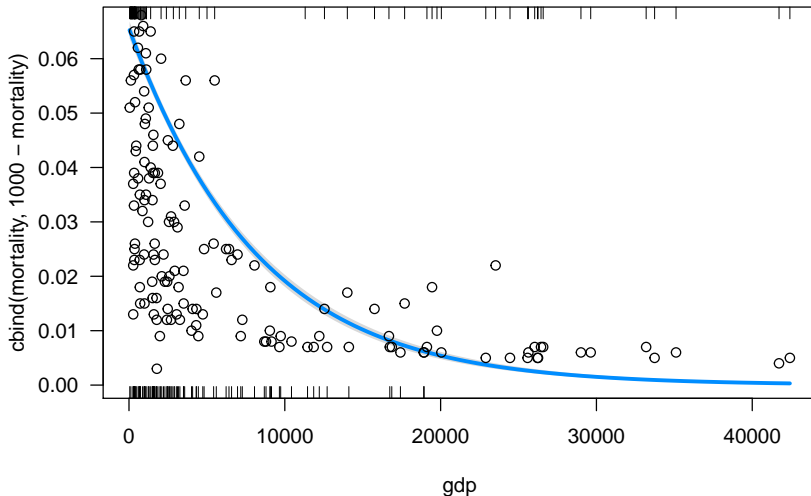
## Effects plot

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



## Plot model using visreg:

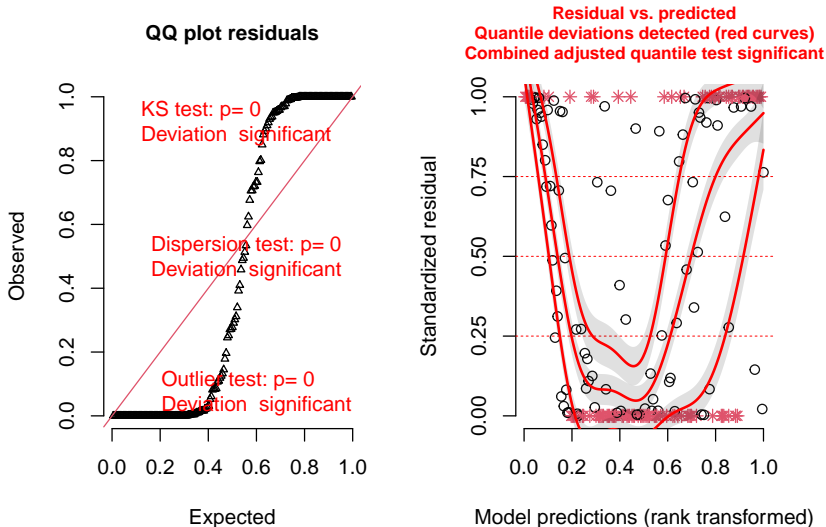
```
visreg(gdp.glm, scale = "response")  
points(mortality/1000 ~ gdp, data = gdp)
```



# Residuals diagnostics with DHARMA

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

## DHARMA residual diagnostics



## Overdispersion

# Testing for overdispersion (DHARMA)

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

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

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

## Overdispersion in logistic regression with proportion data

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

Call:

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

Deviance Residuals:

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

Coefficients:

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

---

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

(Dispersion parameter for quasibinomial family taken to be 20.79)

## Mean estimates do not change after accounting for overdispersion

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

```
gdp effect
```

```
gdp
```

	40	10000	20000	30000	40000
	0.0652177296	0.0191438829	0.0054028095	0.0015096074	0.0004206154

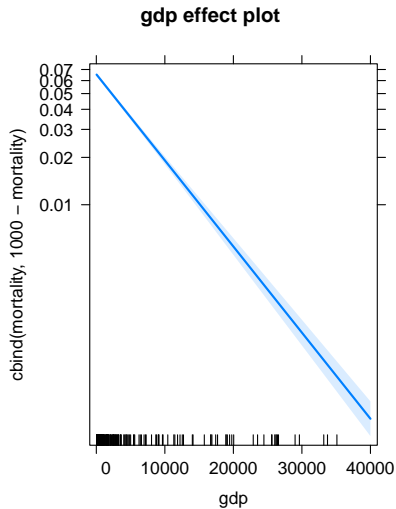
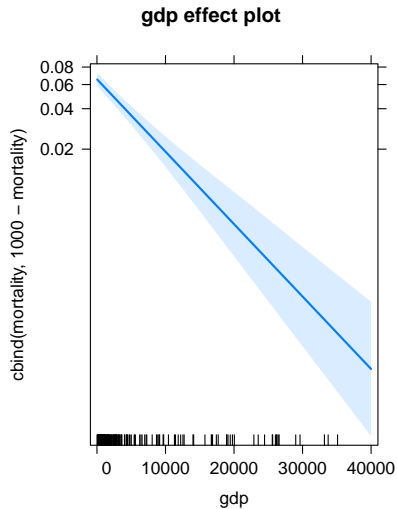
```
model: cbind(mortality, 1000 - mortality) ~ gdp
```

```
gdp effect
```

```
gdp
```

	40	10000	20000	30000	40000
	0.0652177296	0.0191438829	0.0054028095	0.0015096074	0.0004206154

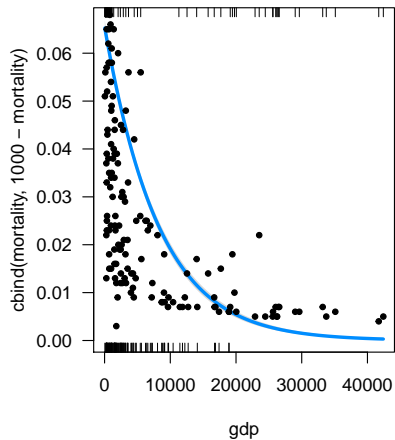
But standard errors (uncertainty) do!



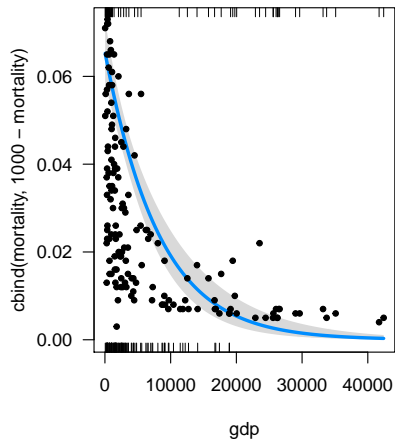


# Plot model and data

**Binomial**



**Quasibinomial**



# Overdispersion

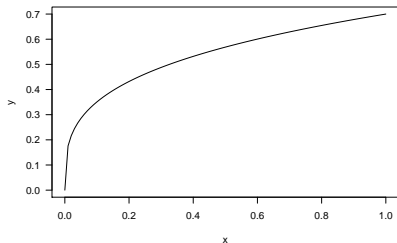
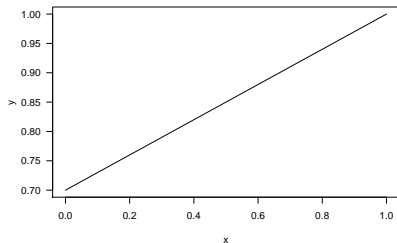
Whenever you fit logistic regression to **proportion** data, check family quasibinomial.

# Think about the shape of relationships

$$y \sim x + z$$

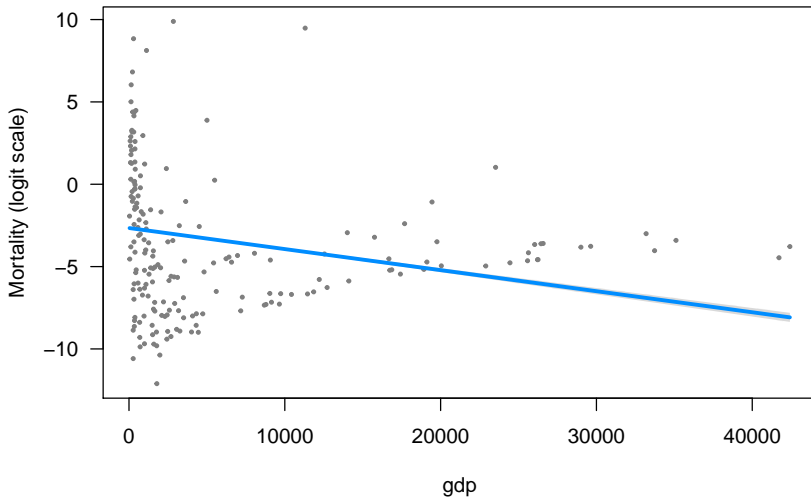
Really? Not everything has to be linear! Actually, it often is not.

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



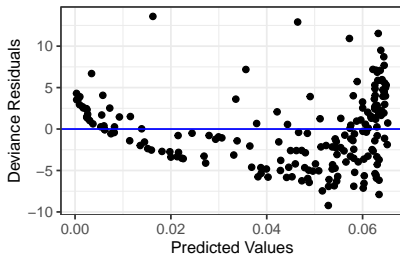
# Think about the shape of relationships

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

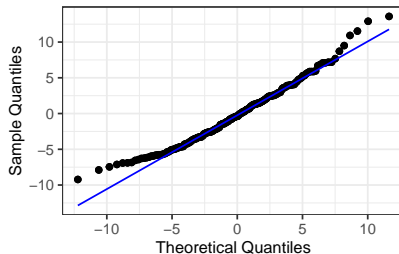


# Think about the shape of relationships

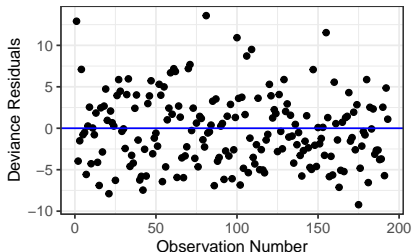
**Residual Plot**



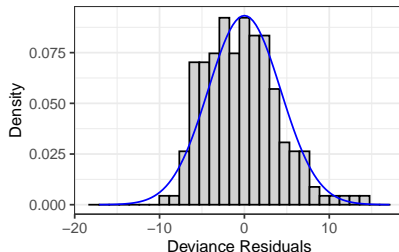
**Q-Q Plot**



**Index Plot**

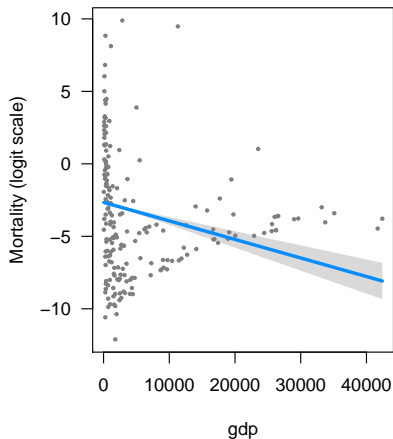


**Histogram**

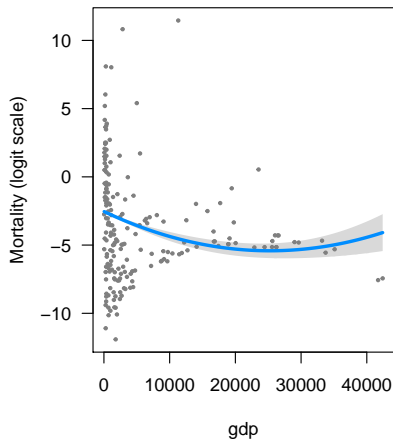


# Think about the shape of relationships

**Mortality ~ GDP**

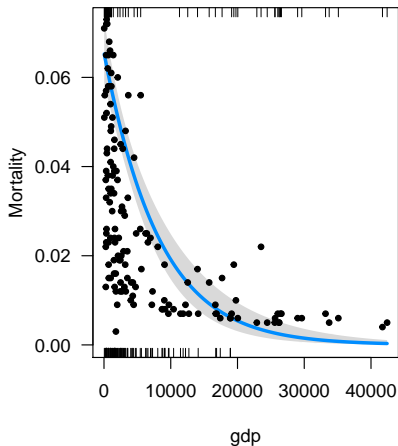


**Mortality ~ GDP + GDP^2**

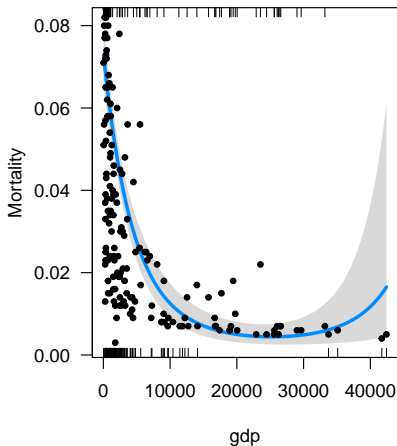


# Think about the shape of relationships

Mortality ~ GDP

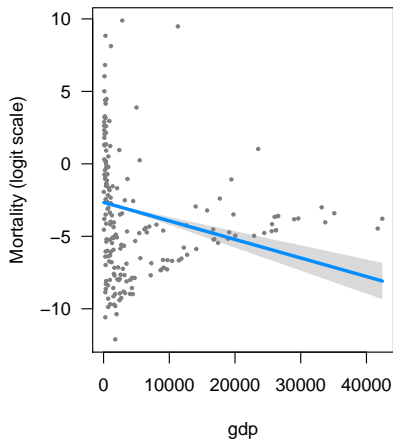


Mortality ~ GDP + GDP^2

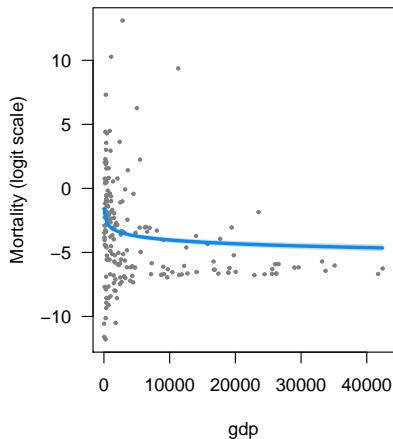


# Think about the shape of relationships

**Mortality ~ GDP**



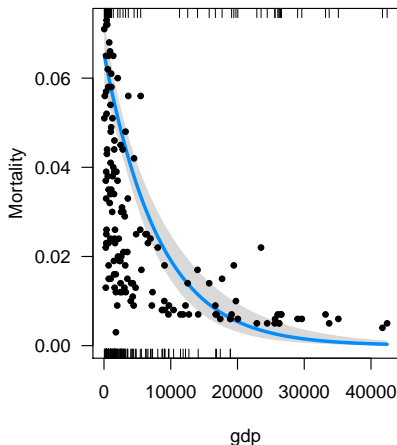
**Mortality ~ log(GDP)**



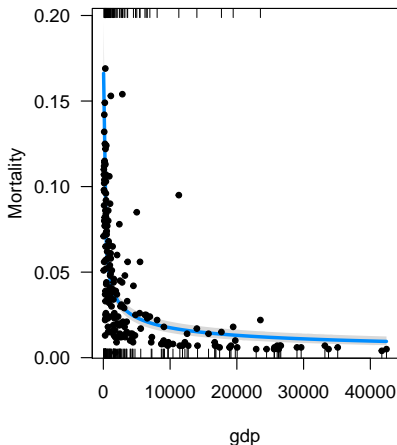


# Think about the shape of relationships

Mortality ~ GDP



Mortality ~ log(GDP)



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

## More examples

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

## Seed set among plants

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

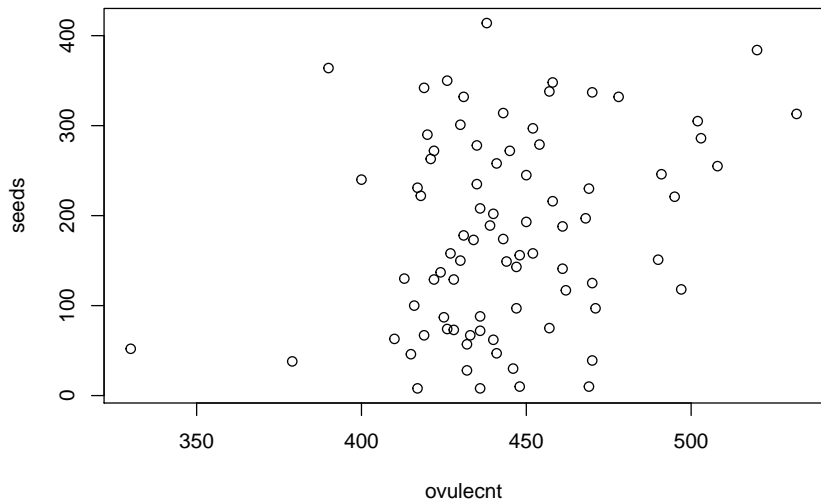
```
# A tibble: 6 x 6
```

	species <chr>	plant <dbl>	pcmass <dbl>	fertilized <dbl>	seeds <dbl>	ovulecnt <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)
```

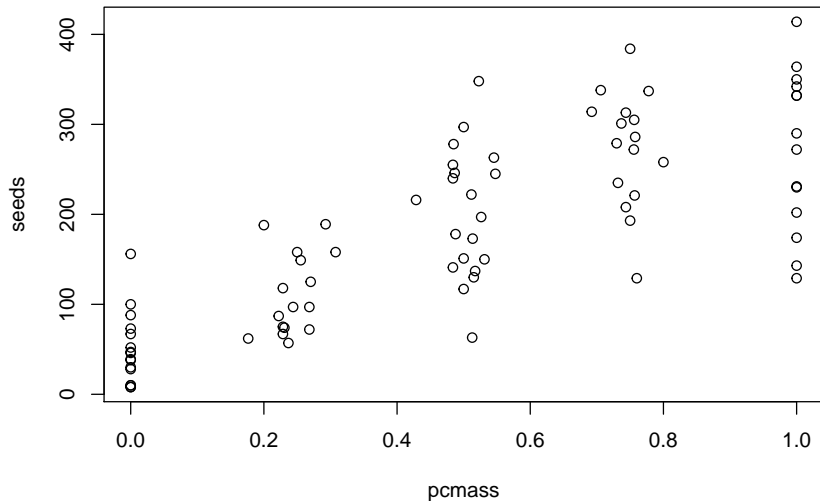
## Number of seeds vs Number of ovules

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

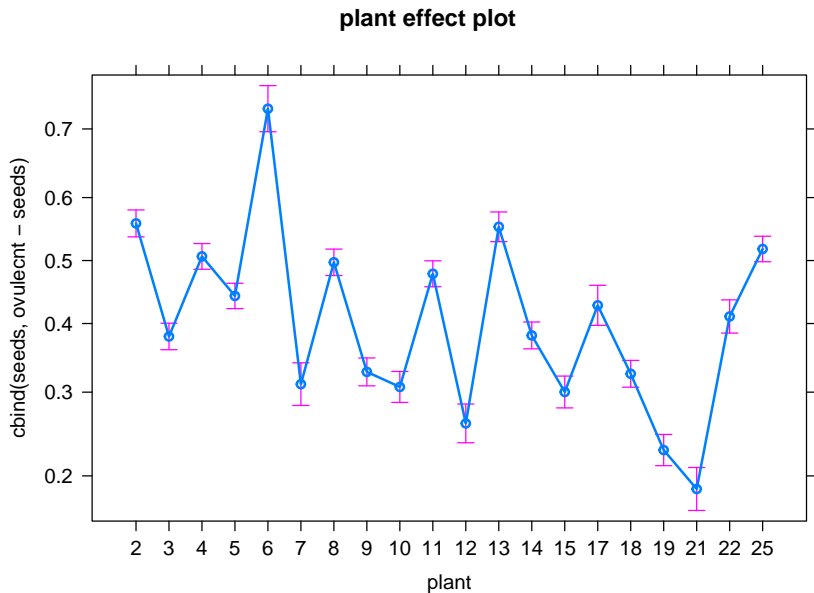


# Number of seeds vs Proportion outcross pollen

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

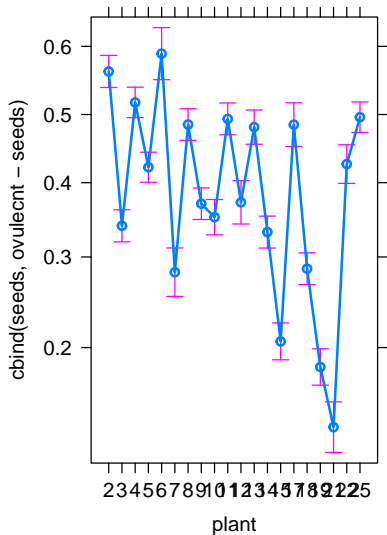


## Seed set across plants



Seed set  $\sim$  outcross pollen

plant effect plot



pcmass effect plot

