

Generalised Linear Models: Logistic regression

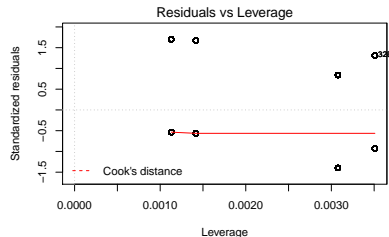
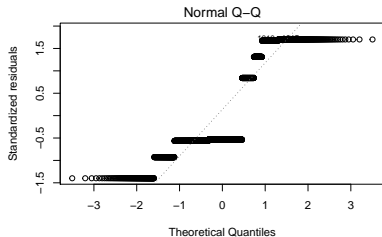
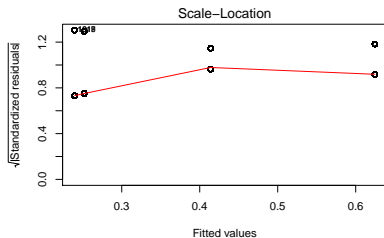
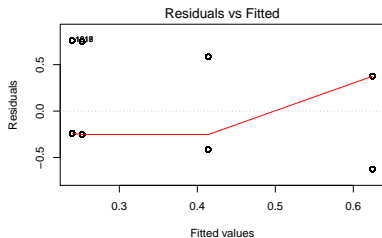
Q: Survival of passengers on the Titanic ~ Class

Read titanic_long.csv dataset.

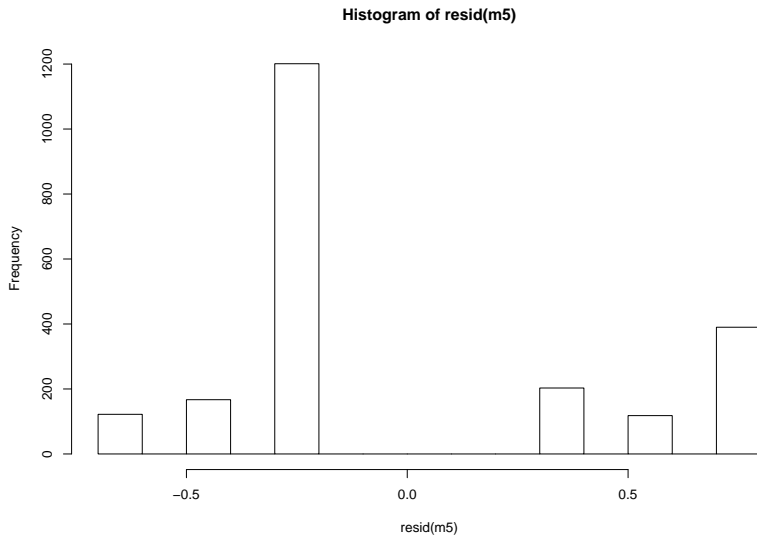
	class	age	sex	survived
1	first	adult	male	1
2	first	adult	male	1
3	first	adult	male	1
4	first	adult	male	1
5	first	adult	male	1
6	first	adult	male	1

Let's fit linear model:

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



Weird residuals!



What if your residuals are clearly non-normal? | 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

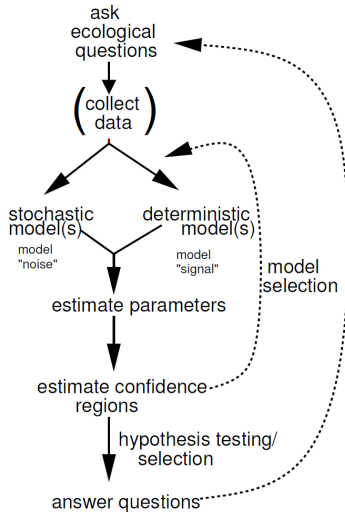


Figure 1.5 Flow of the modeling process.

Figure 1:

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

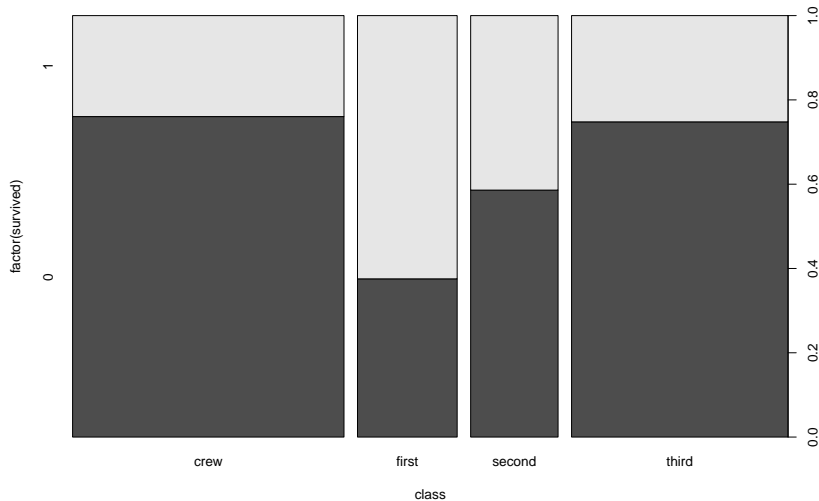
```
library(dplyr)
titanic %>%
  group_by(class, survived) %>%
  summarise(count = n())
```

```
# A tibble: 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 =
n())
```

Or graphically...

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



Fitting GLMs in R: glm

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

Call:

```
glm(formula = survived ~ class, family = binomial(link = "logit"),  
    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 ***
classthir	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
```

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
```

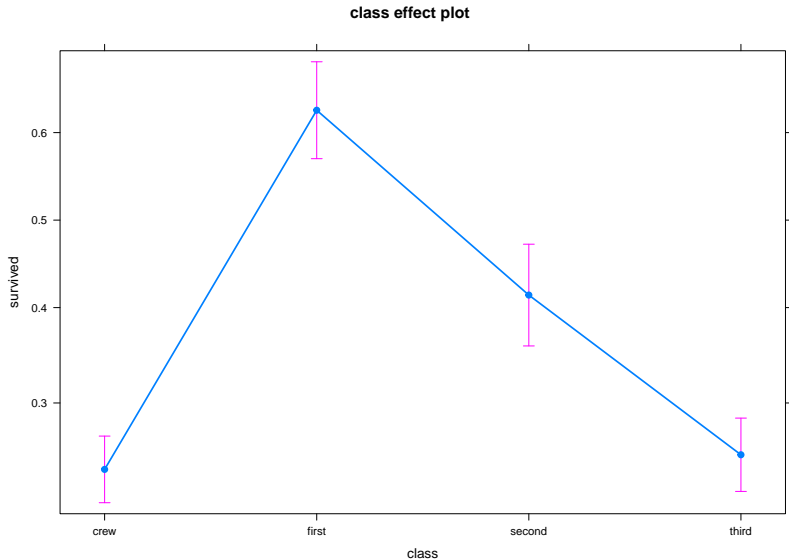
```
class effect
```

```
class
```

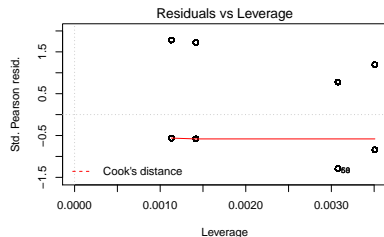
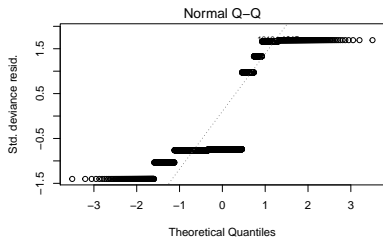
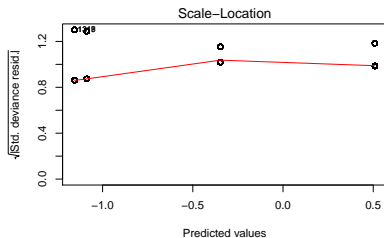
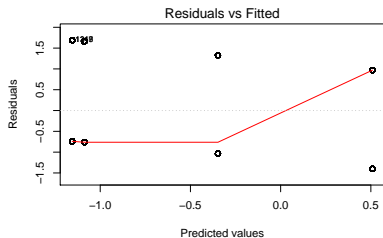
	crew	first	second	third
	0.2395480	0.6246154	0.4140351	0.2521246

Effects plot

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



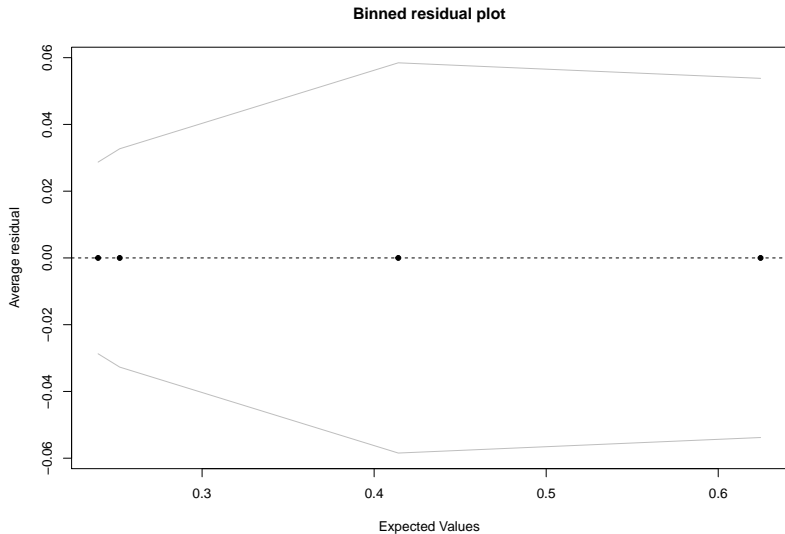
Logistic regression: model checking



Not very useful.

Binned residual plots for logistic regression

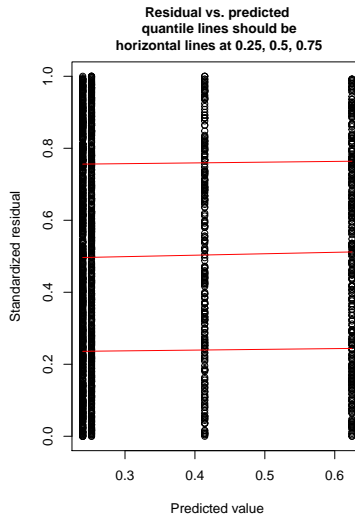
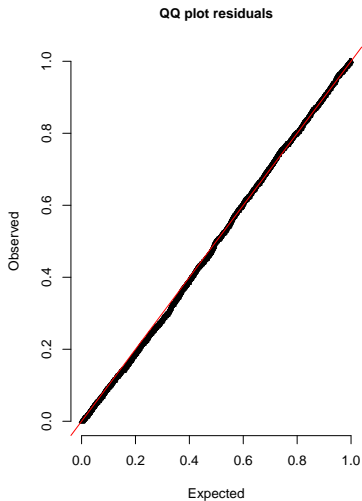
```
predvals <- predict(tit.glm, type="response")  
arm::binnedplot(predvals, titanic$survived - predvals)
```



Residual diagnostics with DHARMA

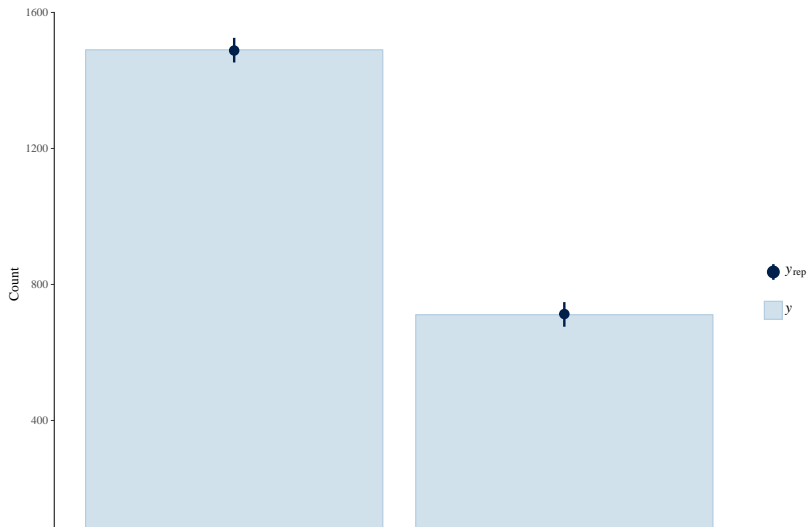
```
library(DHARMA)
simulateResiduals(tit.glm, plot = TRUE)
```

DHARMA scaled residual plots



Model checking with simulated data

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



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`.

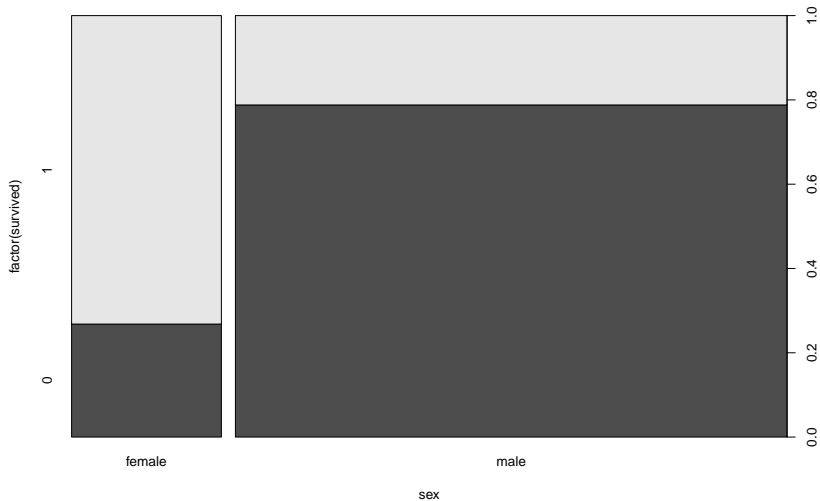
Recapitulating

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

Call:

```
glm(formula = survived ~ sex, family = binomial(link = "logit"),  
     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

Effects

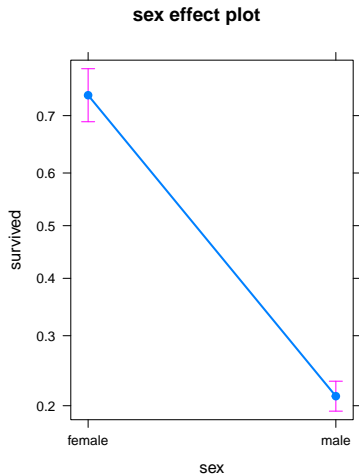
```
model: survived ~ sex
```

```
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 additive model with both factors

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

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

```
      coef.est coef.se
```

```
(Intercept)  1.19      0.16
```

```
classfirst    0.88      0.16
```

```
classecond -0.07      0.17
```

```
classthird  -0.78      0.14
```

```
sexmale      -2.42      0.14
```

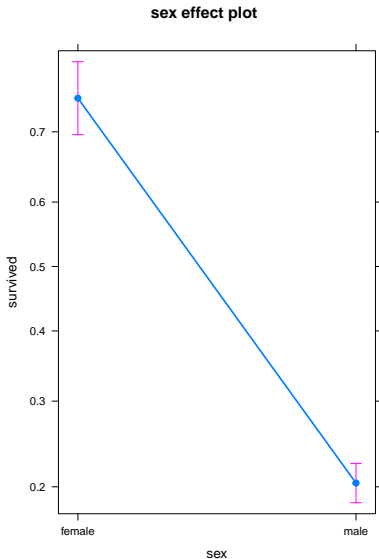
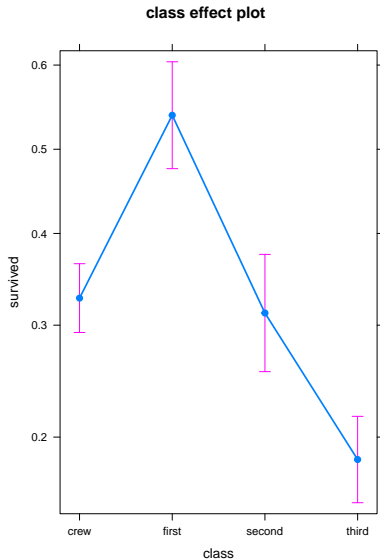
```
---
```

```
  n = 2201, k = 5
```

```
residual deviance = 2228.9, null deviance = 2769.5 (difference
```

Plot additive model

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



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

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

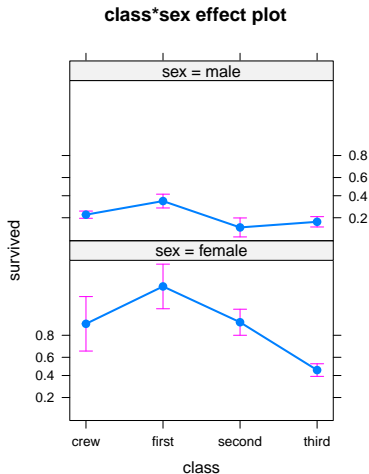
residual deviance = 2163.7, null deviance = 2769.5 (difference

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.

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

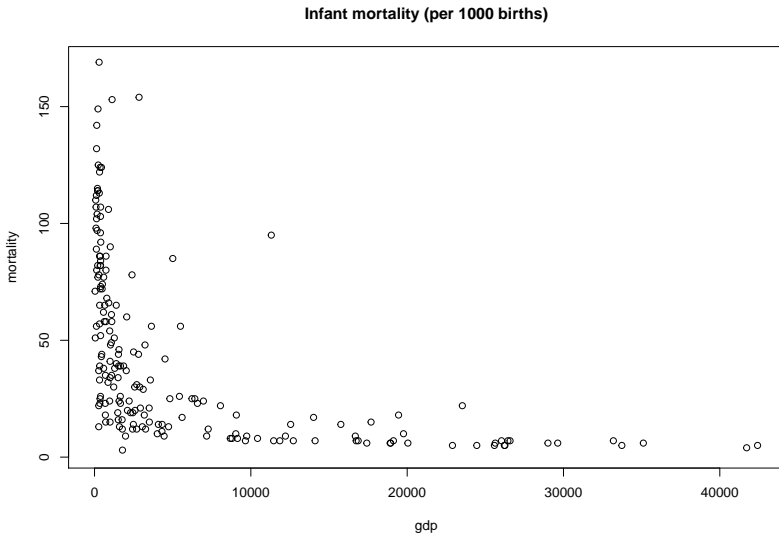
Example dataset: GDP and infant mortality

Read UN_GDP_infantmortality.csv.

	country	mortality	gdp
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(link = "logit"))
```

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
```

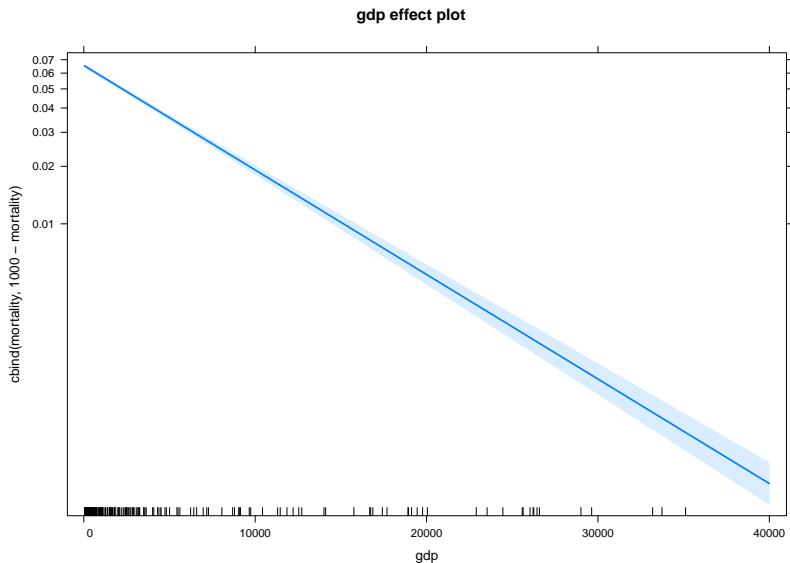
```
gdp effect
```

```
gdp
```

	40	10000	20000	30000	40000
	0.0652177296	0.0191438829	0.0054028095	0.0015096074	0.0004206154

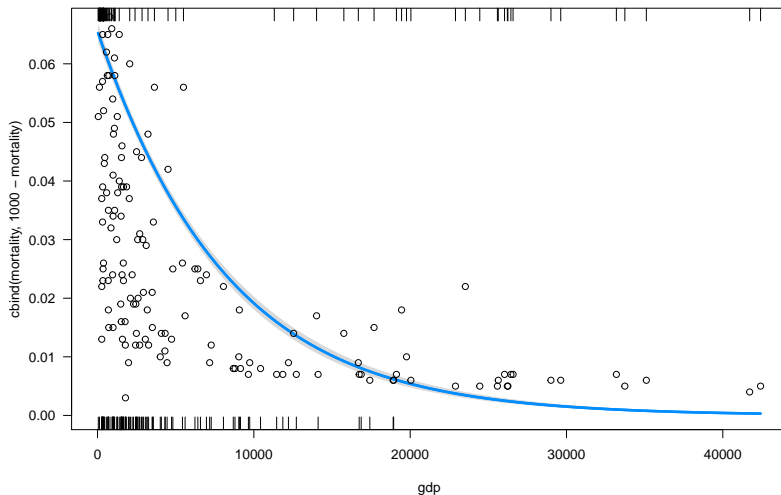
Effects plot

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



Plot model using visreg:

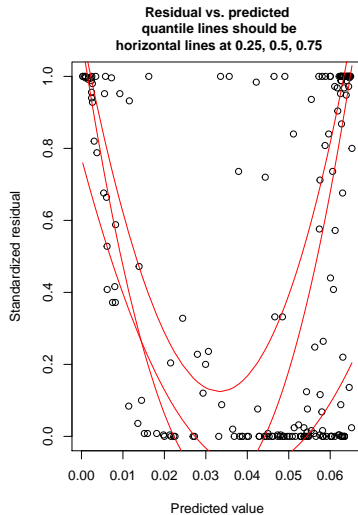
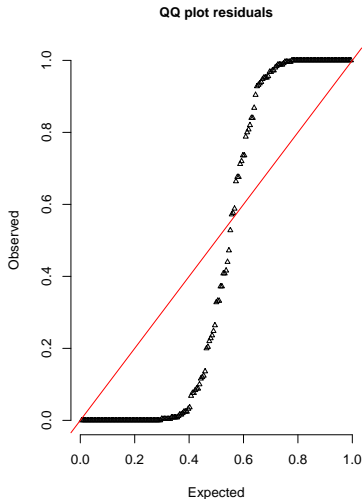
```
visreg(gdp.glm, scale = "response")  
points(mortality/1000 ~ gdp, data = gdp)
```



Residuals diagnostics with DHARMa

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

DHARMa scaled residual plots



Overdispersion

Testing for overdispersion (DHARMa)

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

DHARMa nonparametric overdispersion test via comparison to simulation under H_0 = fitted model

```
data:  simres  
dispersion = 20.761, 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
```

	40	10000	20000	30000	40000
	0.0652177296	0.0191438829	0.0054028095	0.0015096074	0.0004206154

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

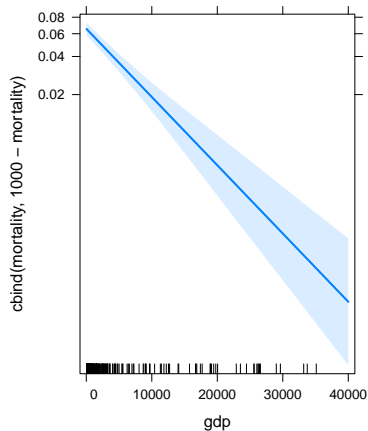
```
gdp effect
```

```
gdp
```

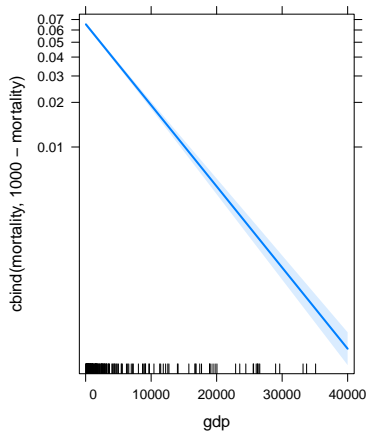
	40	10000	20000	30000	40000
	0.0652177296	0.0191438829	0.0054028095	0.0015096074	0.0004206154

But standard errors (uncertainty) do!

gdp effect plot

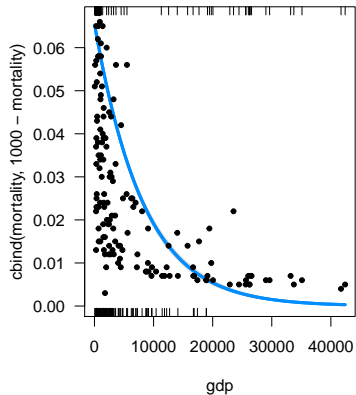


gdp effect plot

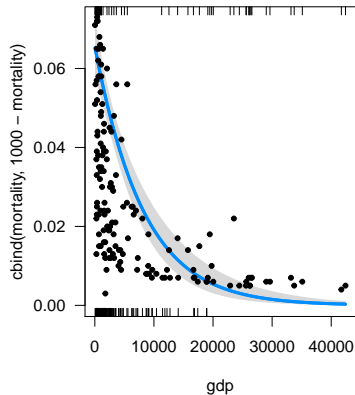


Plot model and data

Binomial



Quasibinomial



Overdispersion

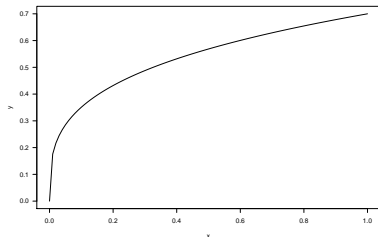
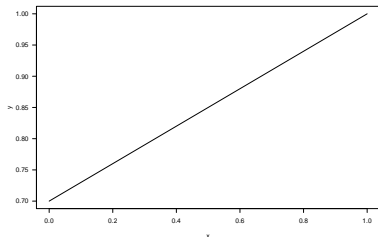
Whenever you fit logistic regression to **proportion** data, check family quasibinomial.

Think about the shape of relationships

$$y \sim x + z$$

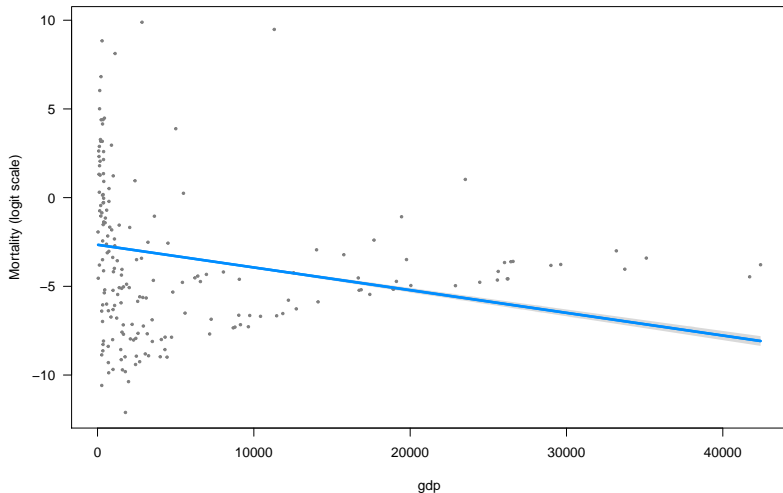
Really? Not everything has to be linear! Actually, it often is not.

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



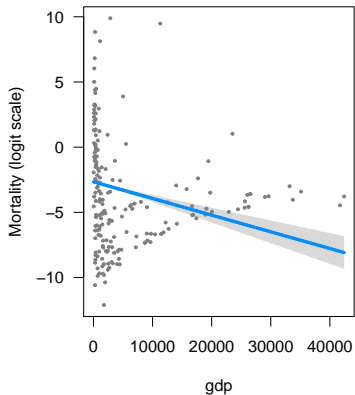
Think about the shape of relationships

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

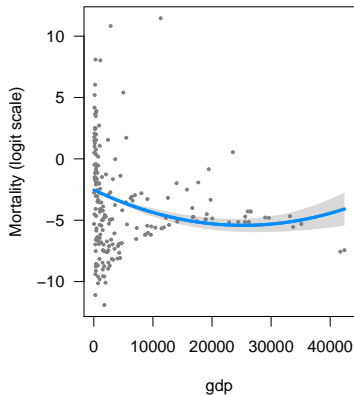


Think about the shape of relationships

Mortality ~ GDP

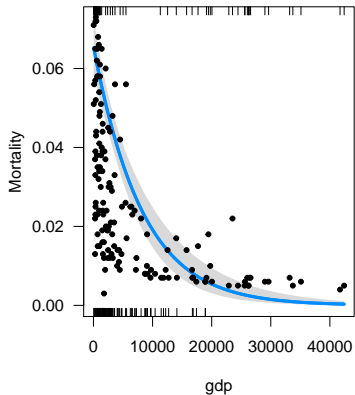


Mortality ~ GDP + GDP^2

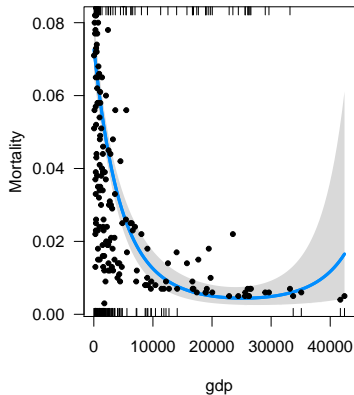


Think about the shape of relationships

Mortality ~ GDP

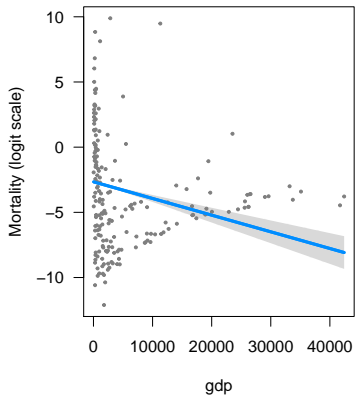


Mortality ~ GDP + GDP^2

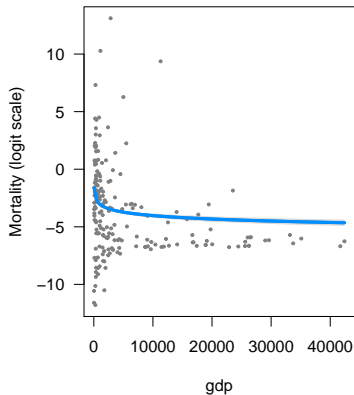


Think about the shape of relationships

Mortality ~ GDP

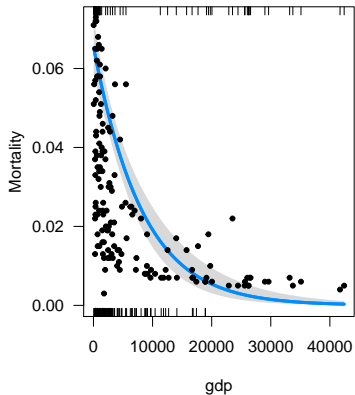


Mortality ~ log(GDP)



Think about the shape of relationships

Mortality ~ GDP



Mortality ~ log(GDP)

