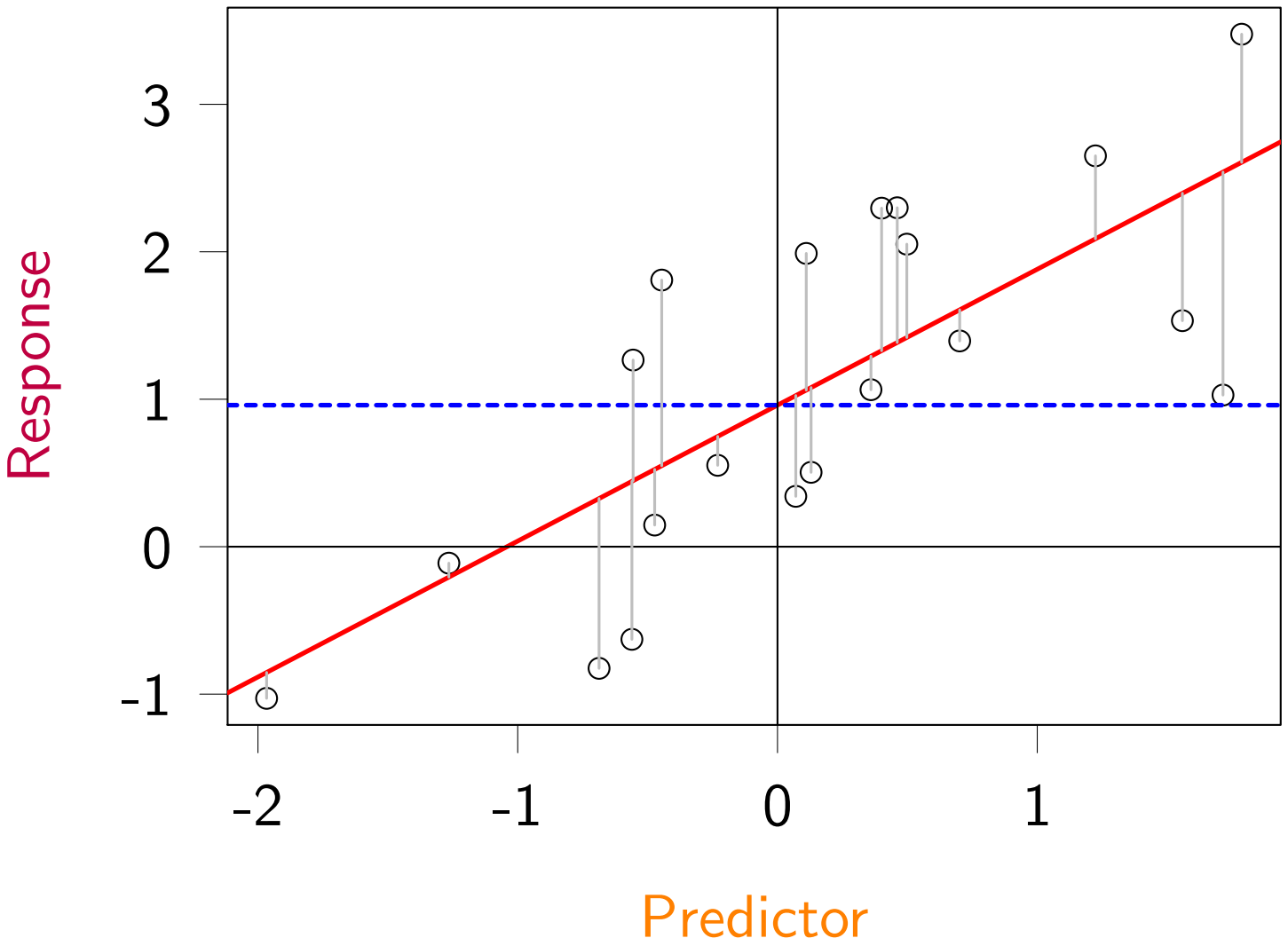
Generalized Linear Models (GLMs)

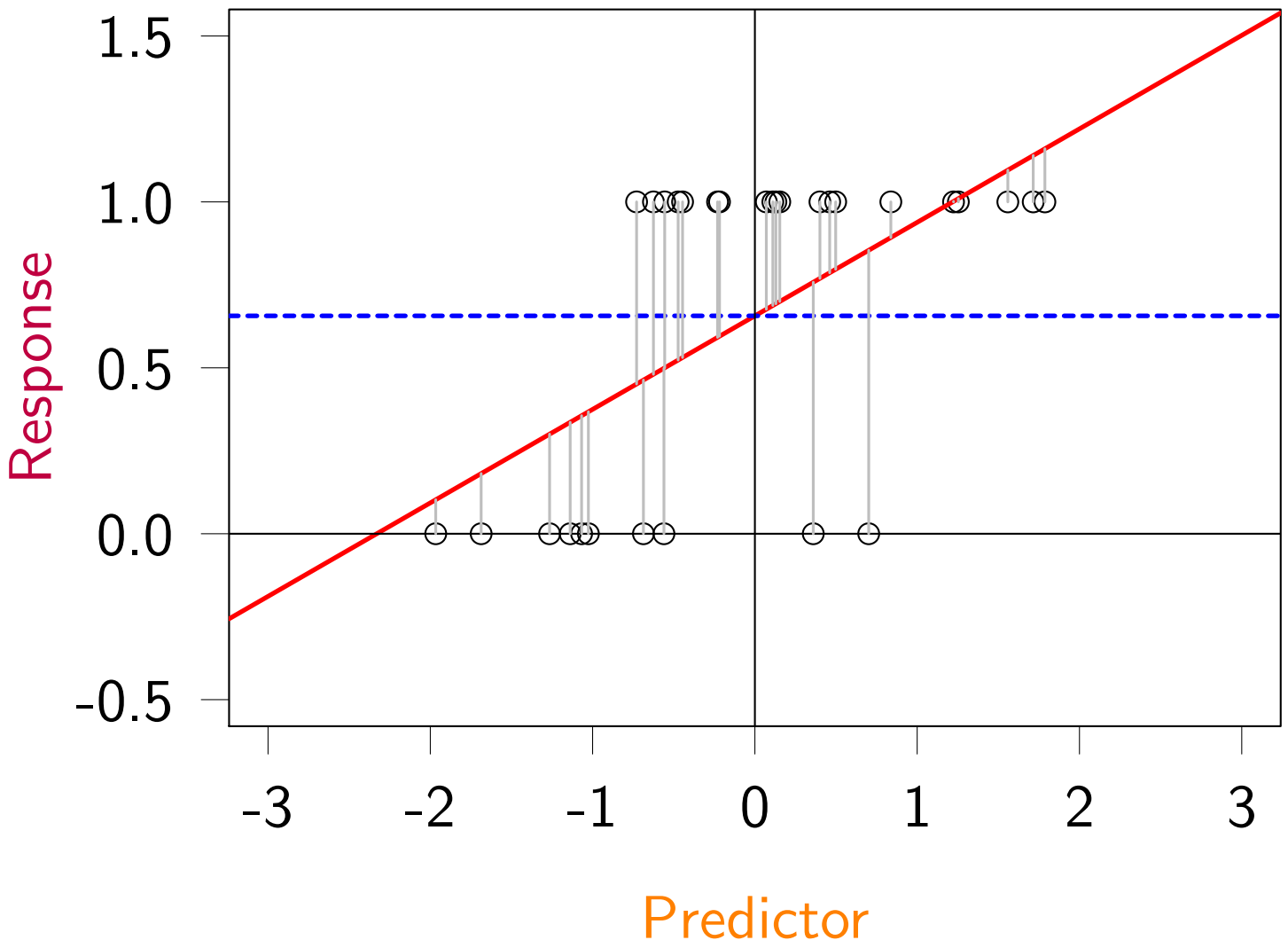
May 17, 2018

A simple linear model

Response = **Intercept** + **Slope** × **Predictor** + **Error**



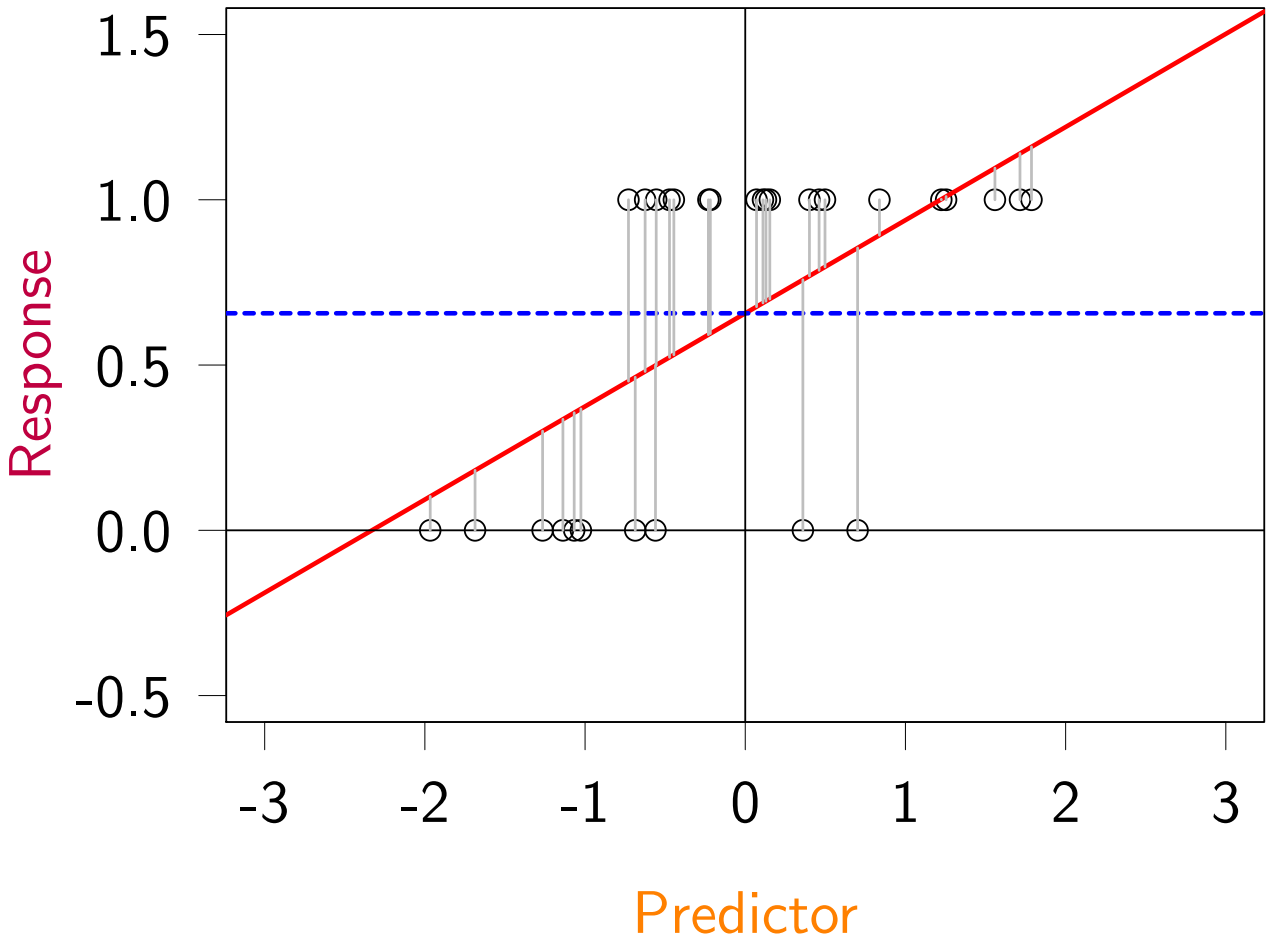
A simple linear model failure: binary data



Linear model basic assumptions

- Linear combination of parameters (including transformation, polynoms, interactions. . .)
  Risk: biologically meaningless
- Predictor not perfectly correlated
  Risk: Model won't run, unstable convergence, or huge SE
- Little error in predictors
  Risk: bias estimates (underestimate with Gaussian error)
- Gaussian error distribution
  Risk: Poor predictions
- Homoscedasticity (constant error variance)
  Risk: Over-optimistic uncertainty, unreliable predictions
- Independence of error
  Risk: Bias and over-optimistic uncertainty

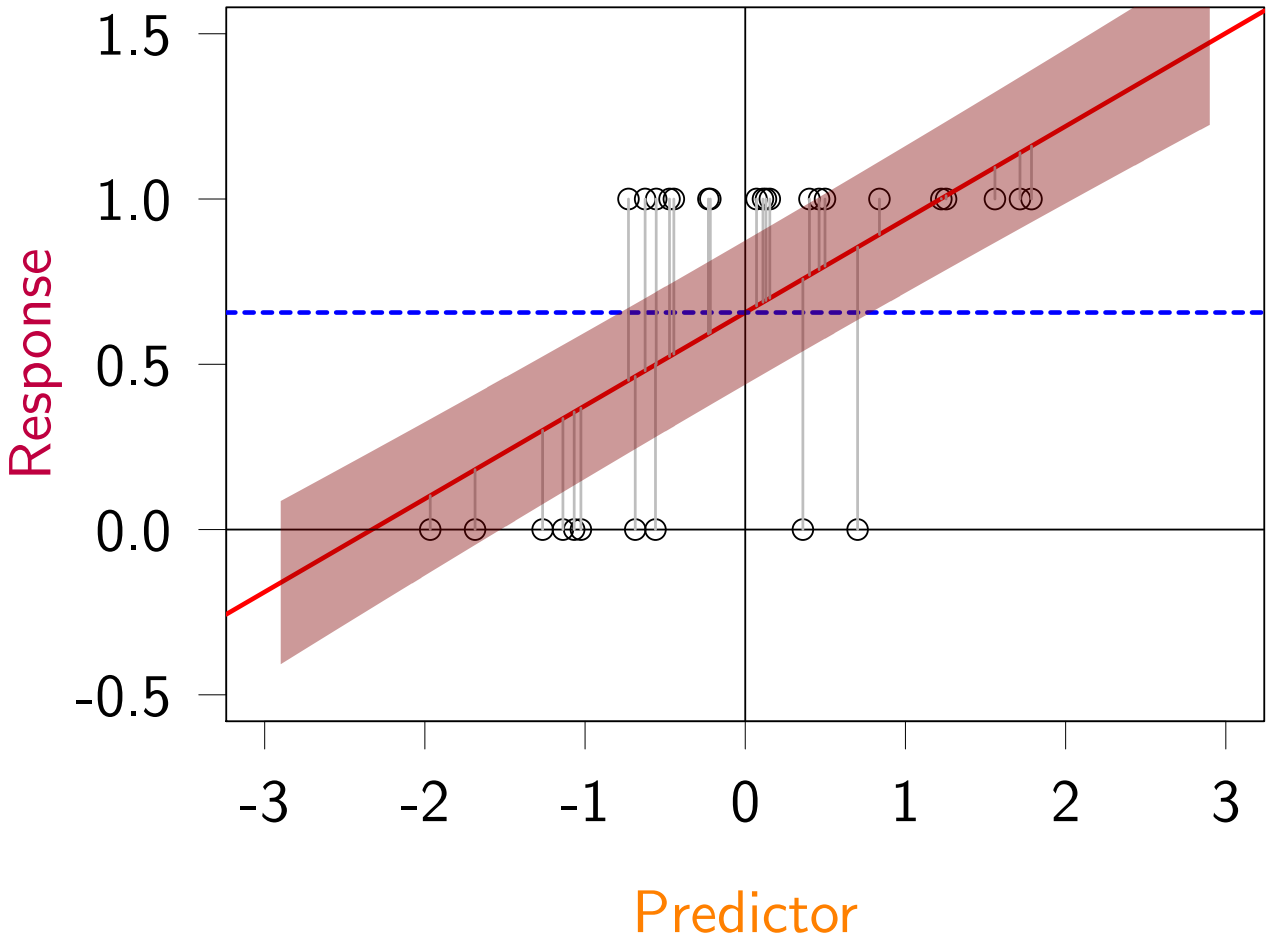
A simple linear model failure: binary data



Assumptions violated:

Non-Gaussian errors, non-constant error variance, correlated errors

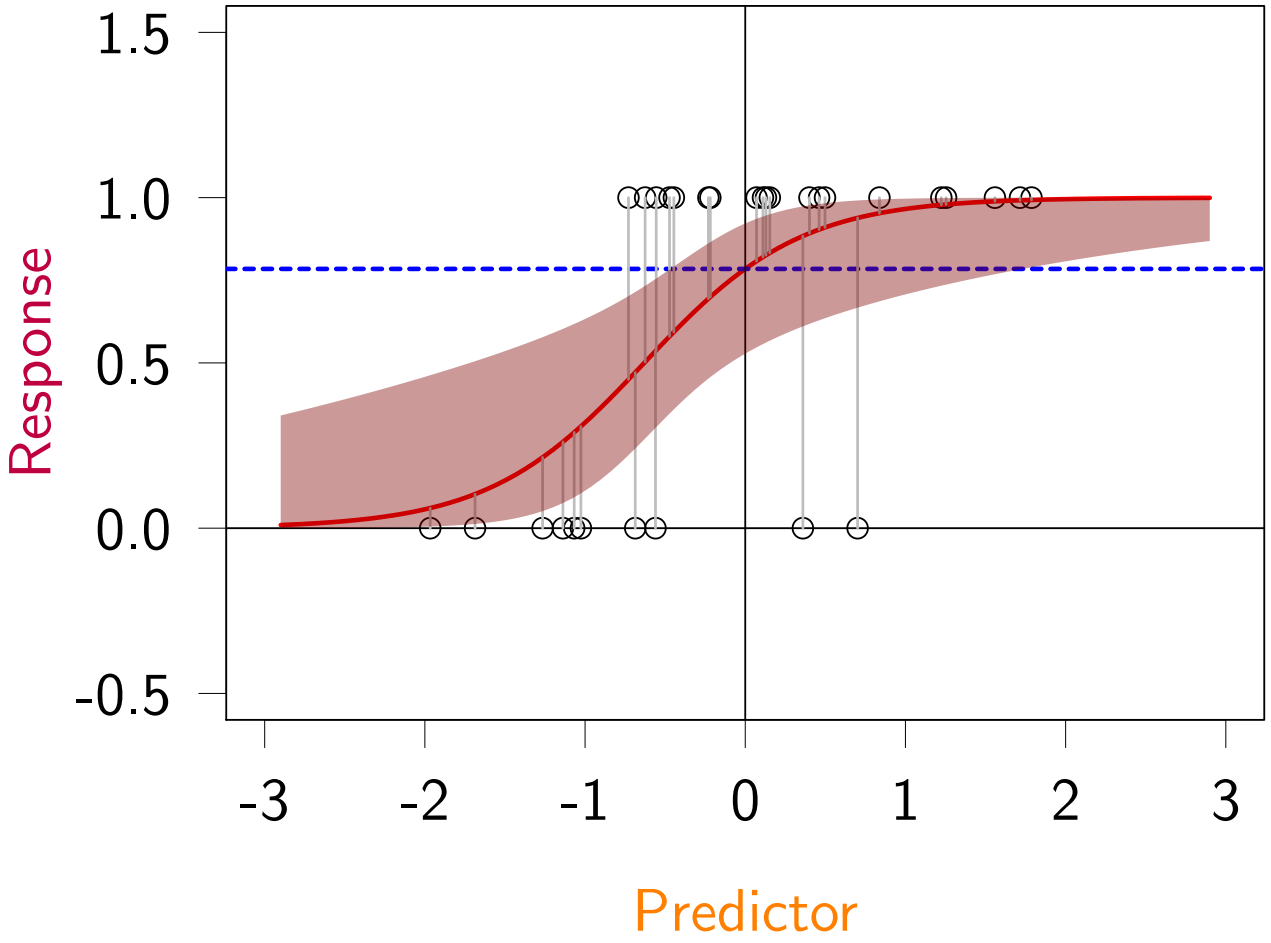
A simple linear model failure: binary data



Practical consequences:

Non-sensical predictions, wrong confidence-interval and p-value, extrapolation ALWAYS fails

What we want our model to do



Good features:

Never out of [0,1], variable uncertainty, non-linear trend, close fit

That is what a Generalized Linear Model does

Vocabulary warning

- General Linear Model (=linear model with several responses, multivariate)
- **Generalized Linear Model (=non-normal errors, and uncertainty dependent on the mean)**

That is what a Generalized Linear Model does

Vocabulary warning

- General Linear Model (=linear model with several responses, multivariate)
- **Generalized Linear Model (=non-normal errors, and uncertainty dependent on the mean)**

What a GLM is:

1. A linear function ($y = \mu + \beta x \ldots$)
2. A probability distribution (Bernouilli, Binomial, Poisson...)
3. A "link function" to convert between the scale of the linear function ($-\infty$ to $+\infty$) and the scale of the data and the probability distribution (often positive integer: 0, 1, 2, 3...)

A GLM fits a continuous expected response; we observe discrete realizations

Logistic regression

- Binary or proportion data (survival, presence/absence. . .)
- Binomial probability distribution (= Bernouilly if binary data)
- Link function often logit: $y = \log(\frac{probability}{1-probability})$
- Back-transformation inverse-logit: $probability = \frac{1}{1+exp(-y)}$
- Linear function $y = intercept + slope_1 predictor_1 + slope_2 predictor_2 + \ldots$

Logistic regression

- Binary or proportion data (survival, presence/absence. . .)
- Binomial probability distribution (= Bernouilly if binary data)
- Link function often logit: $y = \log(\frac{probability}{1-probability})$
- Back-transformation inverse-logit: $probability = \frac{1}{1+exp(-y)}$
- Linear function $y = intercept + slope_1 predictor_1 + slope_2 predictor_2 + \ldots$

Binomial (and Bernouilli distribution in R):

```
bernouilli_random_sample <- rbinom(n = 10000, size = 1, prob = 0.3)
hist(bernouilli_random_sample)
mean(bernouilli_random_sample); 0.3
var(bernouilli_random_sample); 0.3*(1-0.3)
```

Logistic regression

- Binary or proportion data (survival, presence/absence. . .)
- Binomial probability distribution (= Bernouilly if binary data)
- Link function often logit: $y = \log(\frac{probability}{1-probability})$
- Back-transformation inverse-logit: $probability = \frac{1}{1+exp(-y)}$
- Linear function $y = intercept + slope_1 predictor_1 + slope_2 predictor_2 + \ldots$

Binomial (and Bernouilli distribution in R):

```
bernouilli_random_sample <- rbinom(n = 10000, size = 1, prob = 0.3)
hist(bernouilli_random_sample)
mean(bernouilli_random_sample); 0.3
var(bernouilli_random_sample); 0.3*(1-0.3)
```

Logistic regression in R:

```
glm(formula = obs ~ 1 + x, family = "binomial", data=data)
```

Logistic regression

Exercise

1. Load `survivalsize.csv`
2. Plot survival data. What kind of distribution is it?
3. Fit a linear model and a logistic model with intercept only. How to interpret the estimate?
4. Fit a linear regression and a logistic regression of survival on relative size, compare the output
5. Check the diagnostic plots for both models. Should you be worried?
6. Extract and visualize a model prediction from both models (use the function predict, and/or do it by hand to practice link-function back-transformation)

Logistic regression: Jensen's inequality

Logistic regression: trivia

- Many other link-functions, e.g. probit has nice properties to measure selection

Logistic regression: trivia

- Many other link-functions, e.g. probit has nice properties to measure selection
- logit and inverse-logit functions in boot and mixtools packages

Logistic regression: trivia

- Many other link-functions, e.g. probit has nice properties to measure selection
- logit and inverse-logit functions in boot and mixtools packages
- exp(slope) is an odd-ratio

Logistic regression: trivia

- Many other link-functions, e.g. probit has nice properties to measure selection
- logit and inverse-logit functions in boot and mixtools packages
- exp(slope) is an odd-ratio
- GLMs on binary data are more (or equally) powerful than proportion data

Poisson regression

- Count data
- Poisson distribution
- Link function: logarithm
- Inverse link function: exponential
- Linear function $y = intercept + slope_1 predictor_1 + slope_2 predictor_2 + \ldots$

Poisson regression

- Count data
- Poisson distribution
- Link function: logarithm
- Inverse link function: exponential
- Linear function $y = intercept + slope_1 predictor_1 + slope_2 predictor_2 + \ldots$

Poisson distribution in R:

```
poisson_random_sample <- rpois(n = 10000, lambda = 4)
hist(poisson_random_sample)
mean(poisson_random_sample)
var(poisson_random_sample)
```

Poisson regression

- Count data
- Poisson distribution
- Link function: logarithm
- Inverse link function: exponential
- Linear function $y = intercept + slope_1 predictor_1 + slope_2 predictor_2 + \ldots$

Poisson distribution in R:

```
poisson_random_sample <- rpois(n = 10000, lambda = 4)
hist(poisson_random_sample)
mean(poisson_random_sample)
var(poisson_random_sample)
```

Poisson regression in R:

```
glm(formula = obs ~ 1 + x, family = "poisson", data = data)
```

Poisson regression

Exercise

1. Load the data reproduction.csv
2. Plot reproduction data, calculate the mean and variance.
3. Overlay a Gaussian distribution of same mean and variance, does it fit?
4. Fit an compare a lm and a Poisson glm of reproduction on size
5. Check the diagnostic plots for both models. Should you be worried?
6. Extract and visualize a model prediction from both models (use the function predict, and/or do it by hand to practice link-function back-transformation)
7. Before GLMs, researchers used to log-transform the data and fit linear models. What are the problems with this approach?

Poisson regression: over-dispersion

Sadly, simple Poisson models are almost always wrong in natura

- Poisson assumes: expected value = values variance

Poisson regression: over-dispersion

Sadly, simple Poisson models are almost always wrong in natura

- Poisson assumes: expected value = values variance
- True if all the sources of variation are in the model (almost impossible)

Poisson regression: over-dispersion

Sadly, simple Poisson models are almost always wrong in natura

- Poisson assumes: expected value = values variance
- True if all the sources of variation are in the model (almost impossible)
- Main risk:

Poisson regression: over-dispersion

Sadly, simple Poisson models are almost always wrong in natura

- Poisson assumes: expected value = values variance
- True if all the sources of variation are in the model (almost impossible)
- Main risk:
  - underestimate uncertainty

Poisson regression: over-dispersion

Sadly, simple Poisson models are almost always wrong in natura

- Poisson assumes: expected value = values variance
- True if all the sources of variation are in the model (almost impossible)
- Main risk:
  - underestimate uncertainty
  - anti-conservative p-values

Poisson regression: over-dispersion

Sadly, simple Poisson models are almost always wrong in natura

- Poisson assumes: expected value = values variance
- True if all the sources of variation are in the model (almost impossible)
- Main risk:
  - underestimate uncertainty
  - anti-conservative p-values
  - find effects that are not real

Poisson regression: over-dispersion

Sadly, simple Poisson models are almost always wrong in natura

- Poisson assumes: expected value = values variance
- True if all the sources of variation are in the model (almost impossible)
- Main risk:
  - underestimate uncertainty
  - anti-conservative p-values
  - find effects that are not real
- Opposite risk rare but possible (e.g., bird clutch size)

Poisson regression: over-dispersion

Sadly, simple Poisson models are almost always wrong in natura

- Poisson assumes: expected value = values variance
- True if all the sources of variation are in the model (almost impossible)
- Main risk:
  - underestimate uncertainty
  - anti-conservative p-values
  - find effects that are not real
- Opposite risk rare but possible (e.g., bird clutch size)
- Solutions: relax mean/variance relationship

Poisson regression: over-dispersion

Sadly, simple Poisson models are almost always wrong in natura

- Poisson assumes: expected value = values variance
- True if all the sources of variation are in the model (almost impossible)
- Main risk:
  - underestimate uncertainty
  - anti-conservative p-values
  - find effects that are not real
- Opposite risk rare but possible (e.g., bird clutch size)
- Solutions: relax mean/variance relationship
  - Negative-binomial distribution

Poisson regression: over-dispersion

Sadly, simple Poisson models are almost always wrong in natura

- Poisson assumes: expected value = values variance
- True if all the sources of variation are in the model (almost impossible)
- Main risk:
  - underestimate uncertainty
  - anti-conservative p-values
  - find effects that are not real
- Opposite risk rare but possible (e.g., bird clutch size)
- Solutions: relax mean/variance relationship
  - Negative-binomial distribution
  - Quasi-Poisson (multiplicative over-dispersion)

Poisson regression: over-dispersion

Sadly, simple Poisson models are almost always wrong in natura

- Poisson assumes: expected value = values variance
- True if all the sources of variation are in the model (almost impossible)
- Main risk:
  - underestimate uncertainty
  - anti-conservative p-values
  - find effects that are not real
- Opposite risk rare but possible (e.g., bird clutch size)
- Solutions: relax mean/variance relationship
  - Negative-binomial distribution
  - Quasi-Poisson (multiplicative over-dispersion)
  - Observation-level random effect (additive over-dispersion)

Poisson regression: over-dispersion

Sadly, simple Poisson models are almost always wrong in natura

- Poisson assumes: expected value = values variance
- True if all the sources of variation are in the model (almost impossible)
- Main risk:
  - underestimate uncertainty
  - anti-conservative p-values
  - find effects that are not real
- Opposite risk rare but possible (e.g., bird clutch size)
- Solutions: relax mean/variance relationship
  - Negative-binomial distribution
  - Quasi-Poisson (multiplicative over-dispersion)
  - Observation-level random effect (additive over-dispersion)
- **You should NEVER use glm(family = poisson) again!**

Poisson regression: over-dispersion demonstrated

```
set.seed(123) # random seed
x <- rnorm(100) # predictor with no effect
y <- exp(-1 + rnorm(100, 0, 2)) # variation of unknown origin
obs <- sapply(y, FUN = function(y){
  rpois(n = 1, lambda = y)}) # generate Poisson data

#plot(x, obs) # visualize data
glm2 <- glm(obs ~ x, family = "poisson") #fit Poisson regression
sglm2 <- summary(glm2) #look at the summary
sglm2$coefficients[2,4] # the p-value
```

```
[1] 1.25526e-49
```

```
glm2q <- glm(obs ~ x, family = "quasipoisson") #quasiPoisson regression
sglm2q <- summary(glm2q) #summary
sglm2q$coefficients[2,4] # p-value
```

```
[1] 0.09476039
```

Poisson regression: over-dispersion demonstrated

Maybe just bad luck?

Exercise

Write a for loop to look at the distribution of p-values

```
set.seed( )
poisson_pvalues <- vector(length = 1000)
quasipoisson_pvalues <-
for(i in  )
{
  simulation + poisson and quasipoisson GLMs
}
hist( )
mean( <0.05)
```