Basic Statistical Analysis

Rodrigo Labouriau

Aarhus University
Department of Mathematics

Module 2, Day 4 - Binomial Models - 2024
(regression and covariance analysis structures)

1

General Remark

This material is only for internal use in the course.

Please, do not circulate and **do not record**.

Outline

Review:

Important general concepts

- Statistical models
- Parameter in statistical model
- Point estimation
- Likelihood function and Maximum likelihood estimate
- Confidence interval and hypothesis test
- Likelihood ratio test
- One-way and two-ways binomial model.
- We will study today binomial regression and binomial covariance analysis models

Binomial one-way classification models

Seed germination example: The data

Watering Level				
1	2	3	4	5
22	41	66	82	79
25	46	72	73	68
27	59	51	73	74
23	38	78	84	70

Number of germinated seeds, out of 100 seeds,
for five different levels of available water (columns)
and four repetitions (rows)

Binomial one-way classification models

Seed germination example: The problem and the steps of the construction of a model

- **Question:**
  Does the amount of available water affect the germination rate ?
- We will develop step by step a mathematical model for this experiment
- **Three steps:**
  1. We model what happens in each single box
  2. Describe separately the results of each of the five watering levels
  3. We consider a model that takes into account **simultaneously** the data with the five watering levels
- One-way classification model
  ("One-way ANOVA-like model")

Review:

Binomial models, one-way classification models

- One classification variable, say T
  Y_{ti} the i^{th} repetition of observations classified as t
- $Y_{11}, Y_{12}, \ldots$ independent
- $Y_{ti} \sim Bi(n_{ti}, p_t)$, for $t = 1, 2, \ldots$
- Equivalently
  $Y_{ti} \sim Bi(n_{ti}, p_{ti})$, for $t = 1, 2, \ldots$
  where $logit(p_{ti}) = T_t$

Binomial one-way classification models - Calculations in R

Seed germination example: preparing for the calculations in R

```
> # We fit a one-way binomial model
>
> # We create the response, a matrix with the numb. of success in the first
> # column and the number of failures in the second column
>
> resp <- cbind(Germ, N-Germ)
> print(resp)
      Germ
 [1,]   22 78
 [2,]   25 75
 [3,]   27 73
```

Binomial one-way classification models - Calculations in R

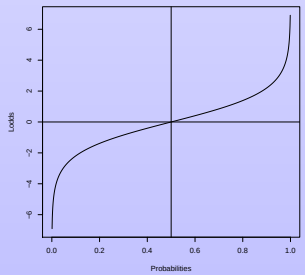
Seed germination example: preparing for the calculations in R

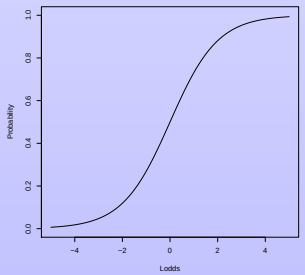
```
> # We fit a one-way binomial model
> OneWayModel <- glm(resp ~ Water - 1, family=binomial(link="identity"))
>
> coefficients(OneWayModel)
Water1 Water2 Water3 Water4 Water5
0.2425 0.4600 0.6675 0.7800 0.7275
>
> # Or equivalently in short
> coef(OneWayModel)
Water1 Water2 Water3 Water4 Water5
0.2425 0.4600 0.6675 0.7800 0.7275
```

Binomial one-way classification models

Another way to parametrize

- Probabilities are numbers between 0 and 1
  During the maximisation process we wight get values out of this range
- Idea: replace the probability by $logit(p) = \log\left(\frac{p}{1-p}\right)$
- $logit^{-1}(L) = \frac{\exp(L)}{1+\exp(L)}$





Binomial one-way classification models - Calculations in R

Parametrising by the logits

```
> # Here we define a function that transforms a probabilities into lodds.
> logit <- function(p){
+   return(log(p/(1-p)))
+ }
> # Next we test some values
> logit(1/2)
[1] 0
```

Binomial one-way classification models - Calculations in R

Parametrising by the logits

```
> # We can also transform lodds in probabilities by the formula
> # exp(lodds)/(1+exp(lodds)) .
> # Here I define a function that transforms probabilities into lodds,
> # called the inverse of the logit.
> ilogit <- function(lodds){
+  return( exp(lodds)/(1+exp(lodds)) )
+ }
> # Next I calculate some values
> ilogit(0)
[1] 0.5
> ilogit(1)
```

Binomial one-way classification models - Calculations in R

Parametrising by the logits

```
> # To fit a model using the lodds instead of probabilities we have to change the
> # parameter "family" of the function glm. So instead of using
> # "family=binomial(link="identity")" we use
> # "family=binomial(link="logit")".
> # Here I fit a model with one different lodd for each watering level
>   fit2 <- glm(resp ~ Water - 1, family=binomial(link="logit"))
>   coefficients(fit2)
    Water1     Water2     Water3     Water4     Water5
-1.1390218 -0.1603427  0.6968995  1.2656664  0.9819754
```

Binomial one-way classification models - Calculations in R

Parametrising by the logits

```
>   fit2 <- glm(resp ~ Water - 1, family=binomial(link="logit"))
> # Please observe the difference between the call and the results for the
> # model defined with the probabilities instead of the lodds
>   fit1 <- glm(resp ~ Water - 1, family=binomial(link="identity"))
>   coefficients(fit1)
Water1 Water2 Water3 Water4 Water5
0.2425 0.4600 0.6675 0.7800 0.7275
```

Binomial one-way classification models - Calculations in R

Parametrising by the logits

```
> # Here I check whether the results of the two models are equivalent
>  coefficients(fit2)
    Water1     Water2     Water3     Water4     Water5
-1.1390218 -0.1603427  0.6968995  1.2656664  0.9819754
>  ilogit(coefficients(fit2))
Water1 Water2 Water3 Water4 Water5
0.2425 0.4600 0.6675 0.7800 0.7275
>  coefficients(fit1)
Water1 Water2 Water3 Water4 Water5
0.2425 0.4600 0.6675 0.7800 0.7275
```

Review:

Binomial models, two-ways classification models

- Two classification variables, say T and S
  Y_{tsi} the i^{th} repetition of observations classified as t and s
- $Y_{111}, Y_{112}, \ldots$ independent
- $Y_{tsi} \sim Bi(n_{tsi}, \rho_{tsi})$, for $t, s = 1, 2, \ldots$
  with several possibilities for the ρ_{tsi}s (yielding different models)
- Some possibilities are:
  - $logit(\rho_{tsi}) = \theta_{tsi}$ (the saturated model)
  - $logit(\rho_{tsi}) = \gamma_{ts}$ (effect modification model)
  - $logit(\rho_{tsi}) = \tau_t + \beta_s$ (additive model)
  - $logit(\rho_{tsi}) = \beta_s$ (no effect of T)
  - $logit(\rho_{tsi}) = \tau_t$ (no effect of S)
  - $logit(\rho_{tsi}) = k$ (null model)

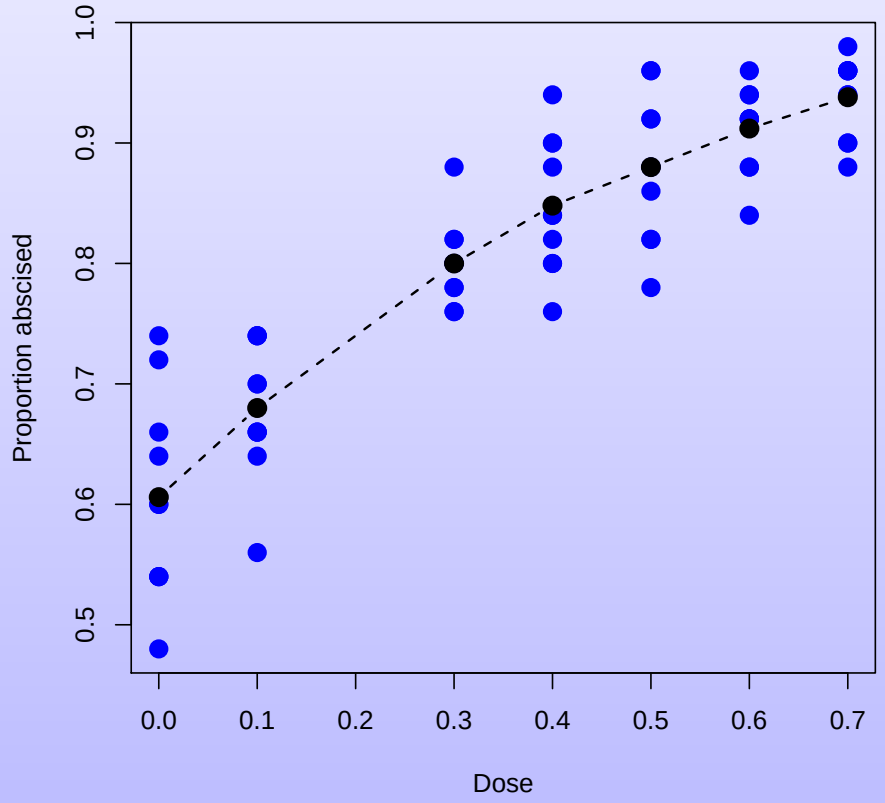
Example: Leave abscission

- Experiment of leave abscission in *Radamachera sp*.
- Several doses of a chemical similar to the abscisic acid were applied to batches of 50 plants.
  Doses (mg/plant): 0, 0.1, 0.3, 0.4. 0.5, 0.6, 0.7
- Counted how many plants presented more than 50 % of abscised leaves after 24h.
- 10 repetitions

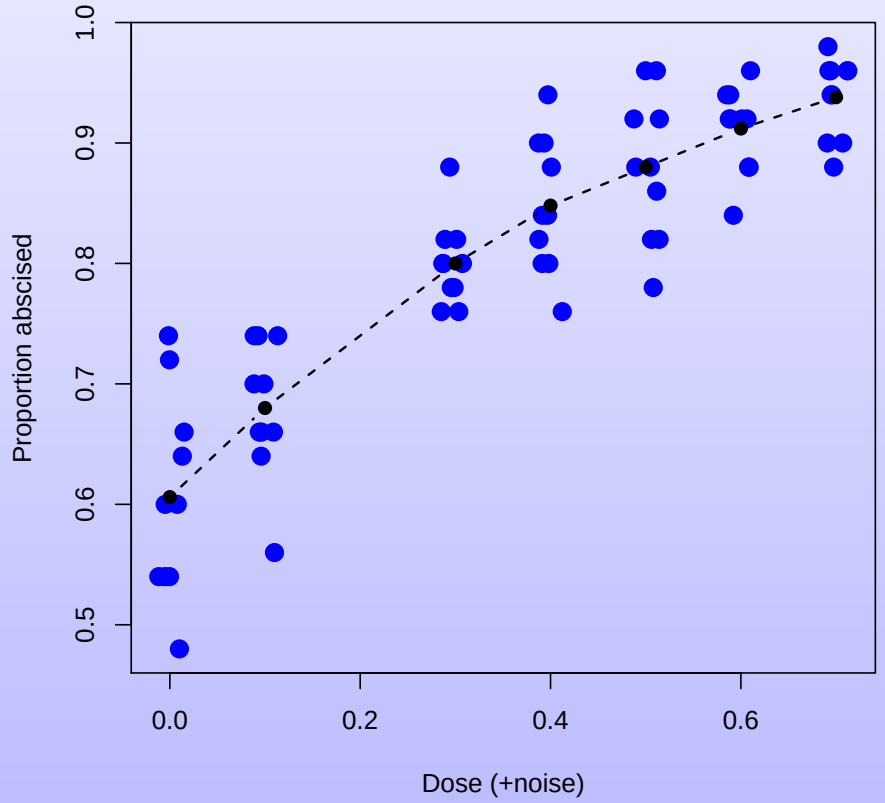
  (all in all 70 observations).

- Data contained in the data-frame "Ch3.Radamachera.simple"

Example: Leave abscission



Example: Leave abscission



Logistic regression

- Regression model: assume that the probabilities of abscission are a (continuous) function of the dose of abscisic acid
- Y is number of plants with more than 50 % of abscised leaves
  out of the 50 plants in each batch
- d is the dose (mg/plant)
- $Y \sim Bi(50, p_d)$
- We assume

$$p_d = \frac{\exp(\alpha + \beta d)}{1 + \exp(\alpha + \beta d)}$$

- Or equivalently,

$$\log\left(\frac{p_d}{1 - p_d}\right) = \alpha + \beta d$$

- α and β are parameters in the model.

Logistic regression

- Probability of an event: p

  Odds of an event: $\frac{p}{1-p}$

  Lodds of an event: $\log\left(\frac{p}{1-p}\right)$

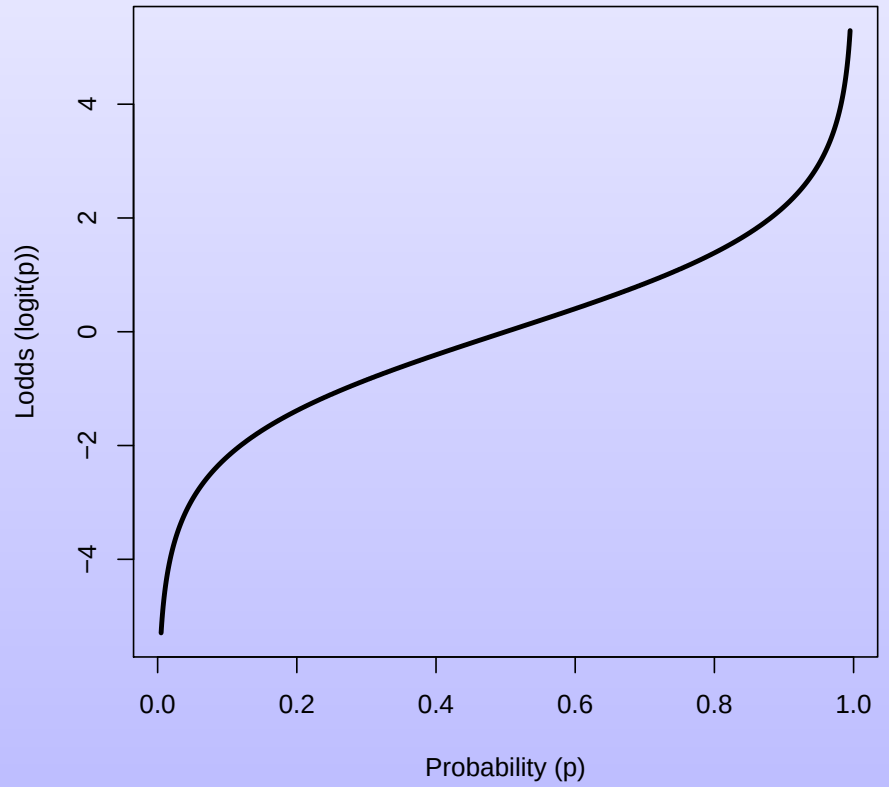
- Logistic function converts probabilities in lodds

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

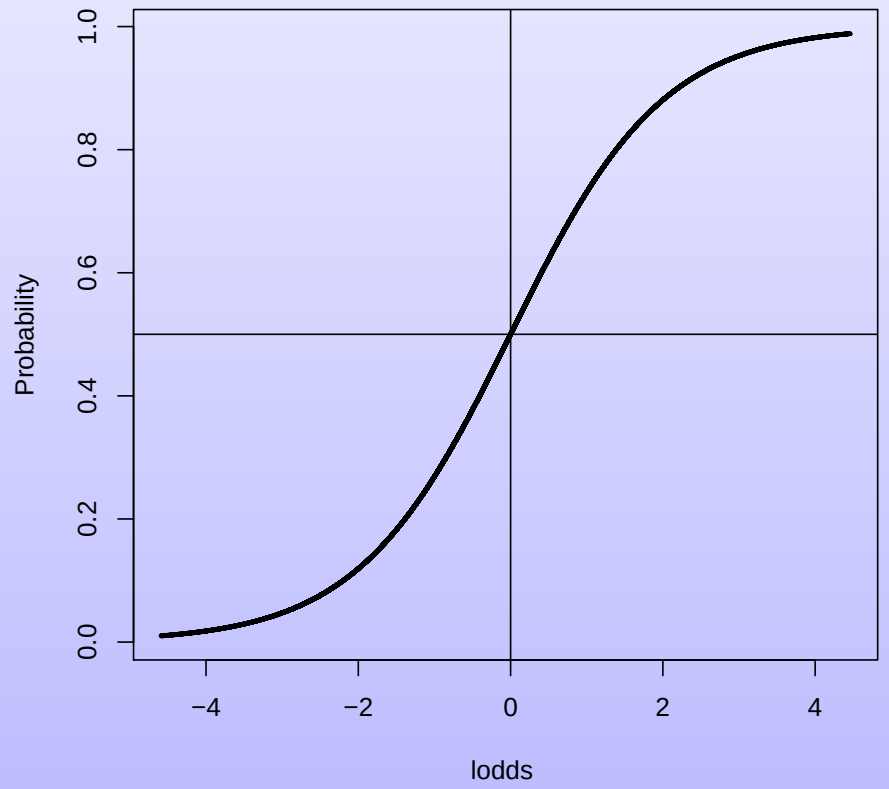
- Inverse logistic converts lodds in probabilities

$$logit^{-1}(lodds) = \frac{\exp(lodds)}{1+\exp(lodds)}$$

Logistic function



Inverse logistic function



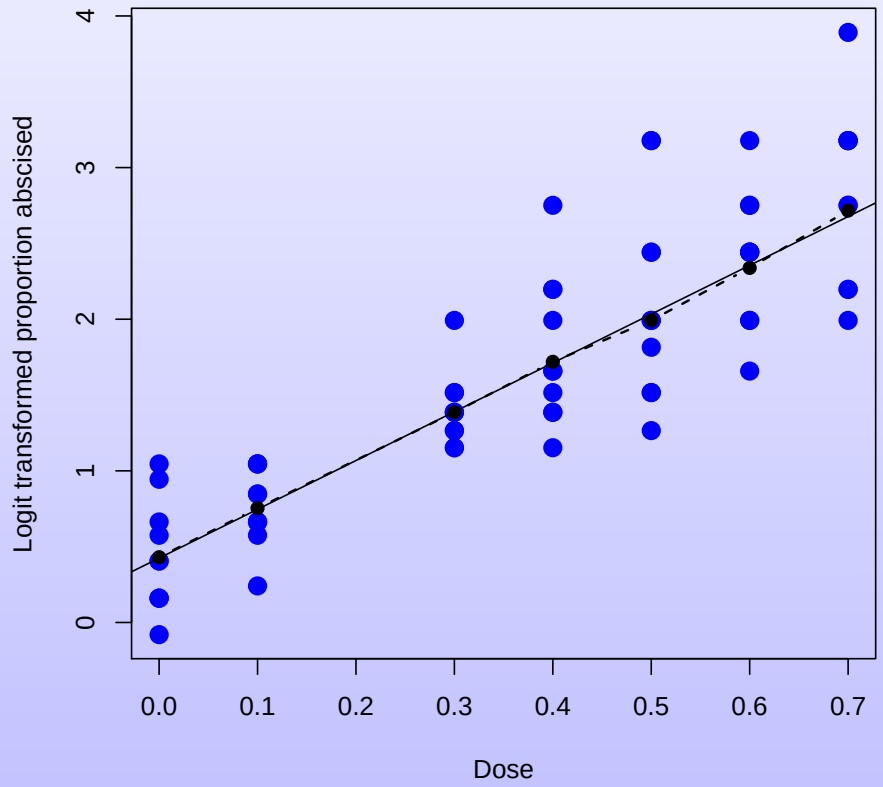
Logistic regression

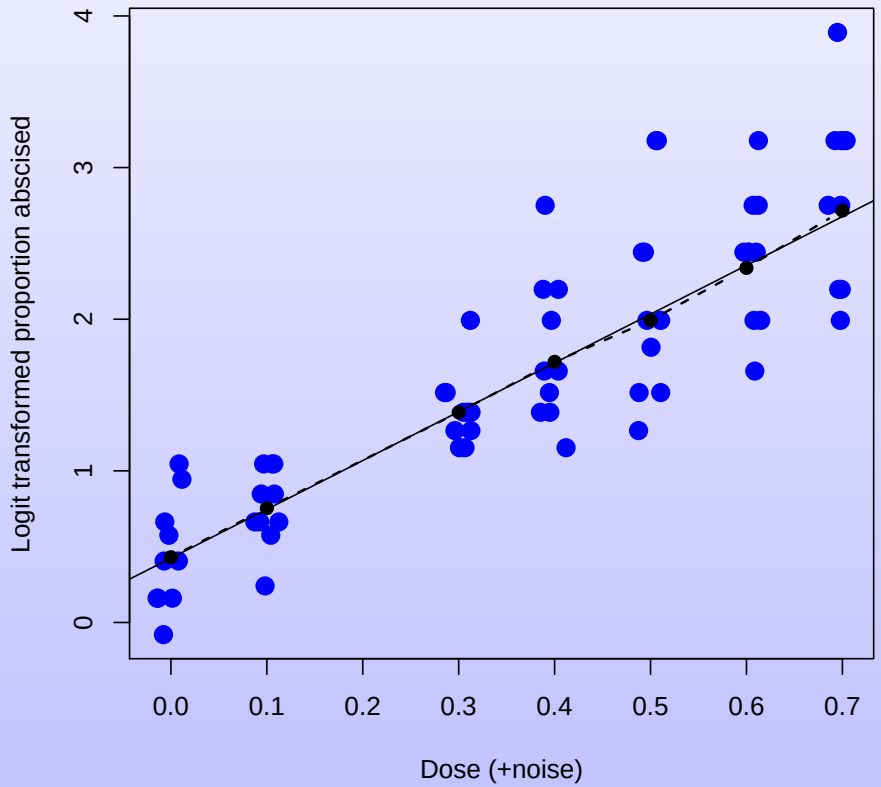
- The inverse logistic function is S-shaped
  taking value 0 at the probability $1/2$
  negative values for $p < 1/2$
  positive values for $p > 1/2$
  asymptotic to 0 and 1.
- It is advantageous to work with in the lodds scale
  since lodds are not bounded (as probabilities are)
- Logistic regression says that the lodds depend linearly on the explanatory variables.

$$logit(p_d) = \log\left(\frac{p_d}{1-p_d}\right) = \alpha + \beta d$$

```
Logit <- function(p){
  return(log(p/(1-p)))
}

ILogit <- function(lodds){
  return(exp(lodds)/(1+exp(lodds)))
}
```


Logit transformed proportion abscised
0
1
2
3
4
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Dose


Logit transformed proportion abscised
0
1
2
3
4
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Dose (+noise)

- The logistic regression says that the lodds depend linearly on the explanatory variables.

$$logit(p_d) = \log\left(\frac{p_d}{1-p_d}\right) = \alpha + \beta d$$

- To fit a logistic regression in R use the function glm with family=binomial(link="logit")
and use a numeric explanatory variable

- R output for the leave abscission example:

```
> resp<-cbind(abscised, nplants-abscised)
```

```
> fit1 <- glm( resp ~ dose, family=binomial(link="logit"))
```

```
> fitted.coef <- coef(fit1); fitted.coef
```

```
(Intercept)        dose
```

```
   0.429748    3.198821
```

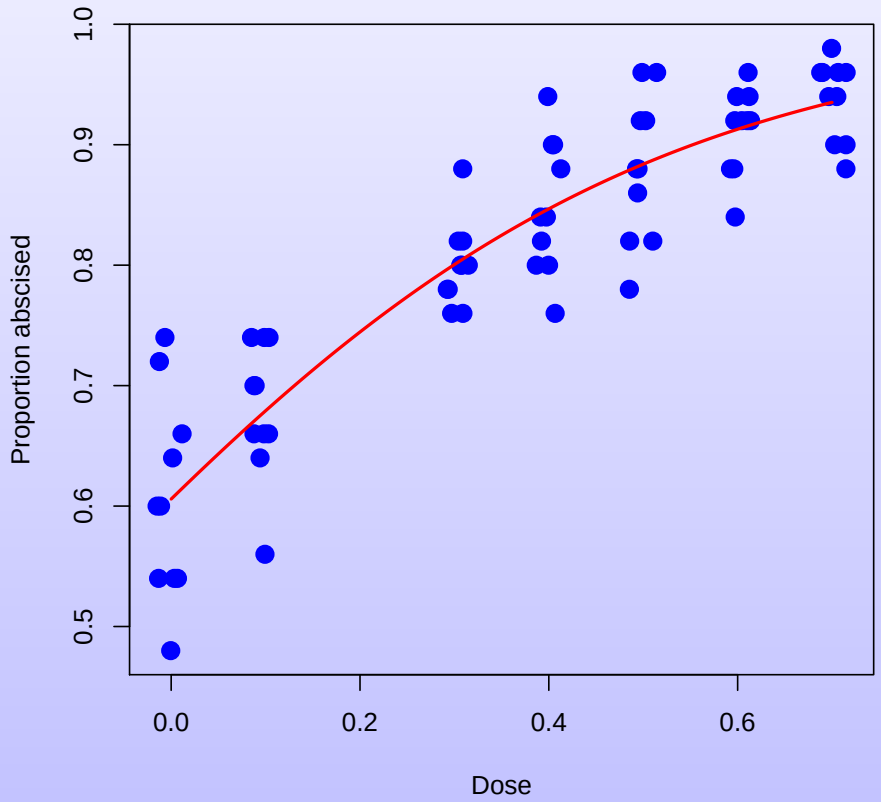
- Here α and β are estimated as 0.4297480 and 3.1988212 respectively

```
> resp<-cbind(abscised, nplants-abscised)
> fit1 <- glm( resp ~ dose, family=binomial(link="logit"))
> summary(fit1)
Call:
glm(formula = resp ~ dose, family = binomial(link = "logit"))
Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.07373  -0.69993  -0.00877   0.76434   2.04812
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.42975    0.06913   6.216  5.1e-10 ***
dose         3.19882    0.19739  16.206  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Dispersion parameter for binomial family taken to be 1) ...
```

Output continued:

```
    Null deviance: 358.360  on 69  degrees of freedom
Residual deviance:  67.451  on 68  degrees of freedom
AIC: 326.69
Number of Fisher Scoring iterations: 4
```

```
# Drawing the logistic adjusted logistic curve
grid.points  <- seq(from=min(dose), to=max(dose), by=0.001)
eta <- fitted.coef[1] + ( fitted.coef[2] * grid.points)
pred.prob <- ILogit(eta); pred.prob
pdf(file="Figure-Ch3-67.pdf")
n<-length(dose)
plot(dose+0.03*(runif(n)-0.5), abscised/nplants, cex=1.5 ,
      xlab="Dose",
      ylab="Proportion abscised",
      pch=19, col='blue')
points(grid.points, pred.prob, type='l', lwd=2, col='red')
dev.off()
```


Proportion abscised
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.2
0.4
0.6
Dose

Fitting a **regression** and a "free curve" model

```
> resp<-cbind(abscised, nplants-abscised)
> fit1 <- glm( resp ~ dose, family=binomial(link="logit"))
> fit2 <- glm( resp ~ 0+factor(dose), family=binomial(link="logit"))
> coef(fit1)
(Intercept)        dose
  0.4297480   3.1988212
> deviance(fit1)
[1] 67.45137
```

Fitting a regression and a "**free curve**" model

```
> fit1 <- glm( resp ~ dose, family=binomial(link="logit"))
> fit2 <- glm( resp ~ 0+factor(dose), family=binomial(link="logit"))
> coef(fit2)
  factor(dose)0 factor(dose)0.1 factor(dose)0.3 factor(dose)0.4 ...
      0.4305291       0.7537718       1.3862944       1.7190001       1.9924302      ...
> deviance(fit2)
[1] 67.3003
```

The free curve model does not assume any form for the response to doses!

Will use the free-curve model to check some assumptions

Testing the homogeneity assumption

Test the homogeneity assumption by comparing, by the likelihood ratio test, the free-curve model and the saturated model.

```
> deviance(fit2)
[1] 67.3003
> length(dose); length(coef(fit2))
[1] 70
[1] 7
> pchisq(deviance(fit2), df=70-7, lower.tail=F)
[1] 0.3322087
>  deviance(fit2)/63
[1] 1.068259
> # Uffa! passed the homogeneity test.
```

Testing the adequacy of the logistic regression curve

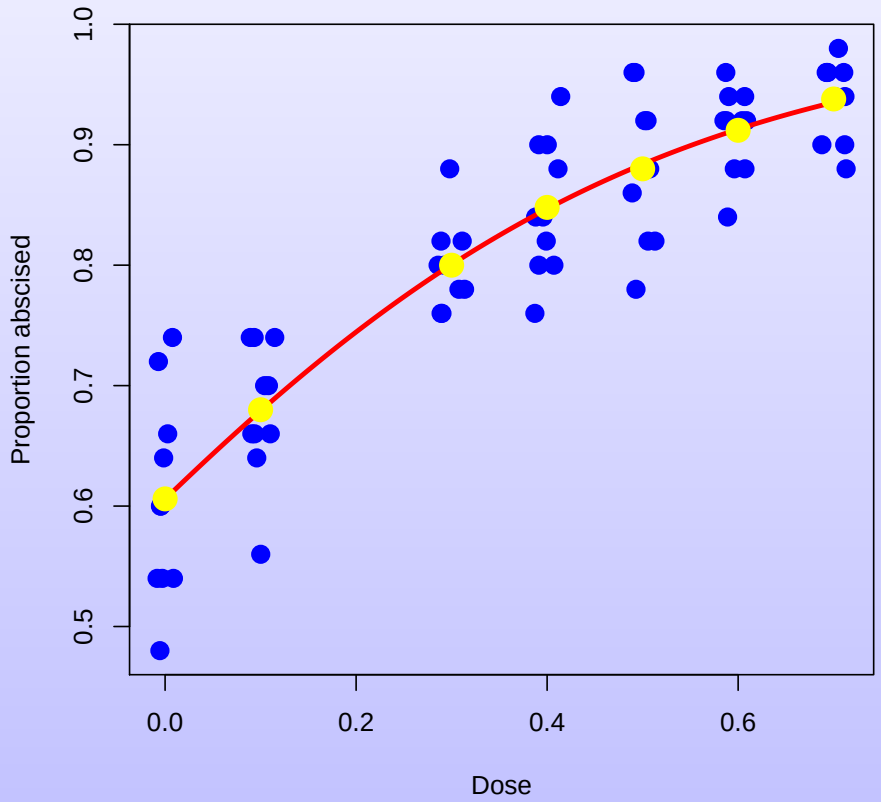
Test the adequacy of the logistic regression curve used by comparing, by the likelihood ratio test, the logistic regression model and the free-curve model.

```
> # Testing the assumed (logistic) regression curve
>   anova(fit1, fit2, test="Chisq")
Analysis of Deviance Table
Model 1: resp ~ dose
Model 2: resp ~ 0 + factor(dose)
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1        68     67.451
2        63     67.300  5    0.151     1.000
> # Again passed the test!
```

Drawing some plots

We make some plots that allow us to visualize what is going on.

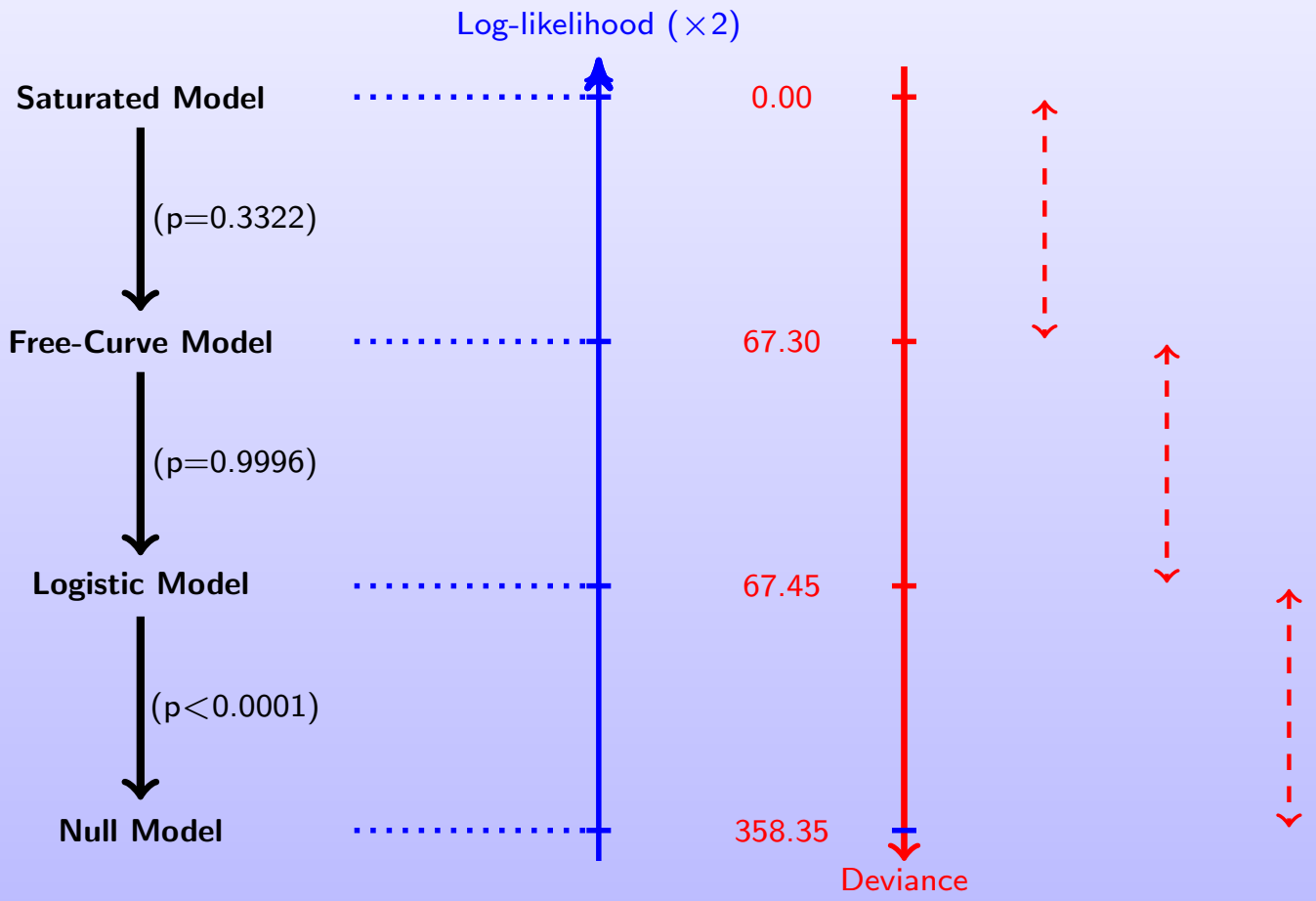
```
> # Ploting the observed proportions and the predicted for the two models
> grid.points  <- seq(from=min(dose), to=max(dose), by=0.0001)
> eta <- fitted.coef[1] + ( fitted.coef[2] * grid.points)
> pred.prob <- ILogit(eta)
> n<-length(dose)
> plot(dose+0.03*(runif(n)-0.5), abscised/nplants, cex=1.5 ,
+       xlab="Dose", ylab="Proportion abscised", pch=19, col='blue')
> points(grid.points, pred.prob, type='l', lwd=3, col='red')
> free.lodds <- ILogit(coef(fit2))
> points(levels(factor(dose)), free.lodds , type='p', cex=2, col='yellow', pch=19)
```


Proportion abscised
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.2
0.4
0.6
Dose

Testing homogeneity and regression curve adequacy simultaneously

```
> # Testing the homogeneity assumption and the appropriatness of the regression
> # curve SIMULTANEOUSLY!
> deviance(fit1)
[1] 67.45137
> length(dose)          # 70 observations
[1] 70
> length(coef(fit1))    #  2 parameters in the free-curve model
[1] 2
> pchisq(deviance(fit1), df=70-2, lower.tail=F)
[1] 0.4959817
>  deviance(fit2)/68
[1] 0.9897103
>  # Again passed the two tests simultaneously!
```

Summarising the analysis performed



Example: Leave abscission

- Experiment of leave abscission in *Radamachera sp*.
- Several doses of a chemical similar to the abscisic acid were applied to batches of 50 plants.
  Doses (mg/plant): 0, 0.1, 0.3, 0.4. 0.5, 0.6, 0.7
- Counted how many plants presented more than 50 % of abscised leaves after 24h.
- 10 repetitions

  (all in all 70 observations).

- Two varieties: 1, 2
  The previous analysis was restricted to variety 1

Example: Leave abscission (continued)

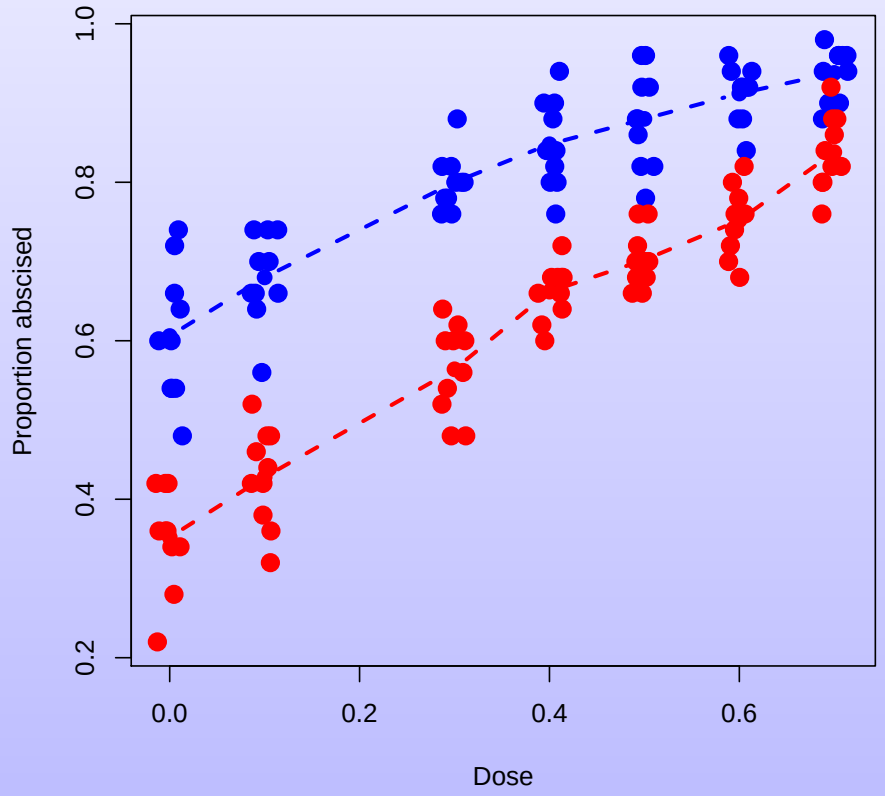
```
> attach(Ch3.Radamachera.simple)
> summary(Ch3.Radamachera.simple)
```

Obs	repetition	dose	variety	abscised	nplants
Min. : 1.00	Min. : 1.0	Min. :0.0000	Min. :1.0	Min. :11.00	Min. :50
1st Qu.: 35.75	1st Qu.: 3.0	1st Qu.:0.1000	1st Qu.:1.0	1st Qu.:30.00	1st Qu.:50
Median : 70.50	Median : 5.5	Median :0.4000	Median :1.5	Median :37.00	Median :50
Mean : 70.50	Mean : 5.5	Mean :0.3714	Mean :1.5	Mean :35.58	Mean :50
3rd Qu.:105.25	3rd Qu.: 8.0	3rd Qu.:0.6000	3rd Qu.:2.0	3rd Qu.:42.25	3rd Qu.:50
Max. :140.00	Max. :10.0	Max. :0.7000	Max. :2.0	Max. :49.00	Max. :50

Example: Leave abscission (continued)

```
> dose1 <- dose[variety==1] ; abscised1 <- abscised[variety==1] ; nplants1 <- nplants[variety==1]
> dose2 <- dose[variety==2] ;abscised2 <- abscised[variety==2];  nplants2 <- nplants[variety==2]
> n <- length(dose1)
> plot(dose1+0.03*(runif(n)-0.5), abscised1/nplants1, cex=1.5 , xlab="Dose",
+       ylab="Proportion abscised",  pch=19, col='blue', ylim=range(abscised/nplants))
> prop <-tapply(abscised1, factor(dose1),sum)/ tapply(nplants1, factor(dose1),sum)
> points(levels(factor(dose1)), prop, type='b', lty=2, lwd=2.5, pch=19, col='blue')
> n <- length(dose2)
> points(dose2+0.03*(runif(n)-0.5), abscised2/nplants2, cex=1.5 , xlab="Dose",
+ ylab="Proportion abscised", pch=19, col='red')
> prop <- tapply(abscised2, factor(dose2),sum)/ tapply(nplants2, factor(dose2),sum)
> points(levels(factor(dose2)), prop, type='b', lty=2, lwd=2.5, pch=19, col='red')
```

Example: Leave abscission

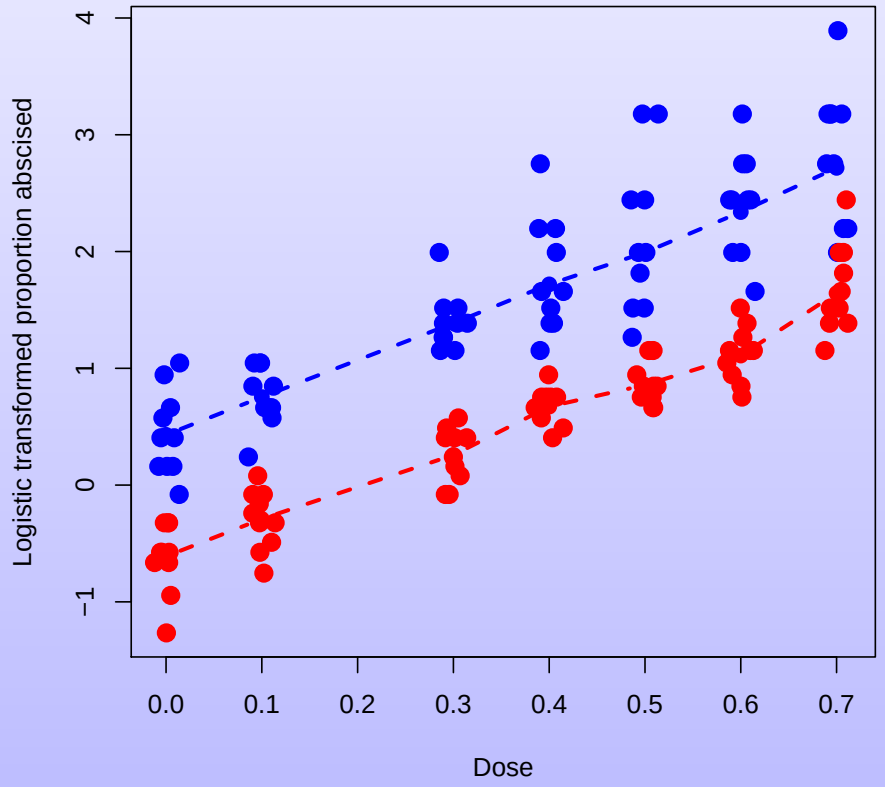


Example: Leave abscission (continued)

Plotting the logit transformed proportions

```
> Logit <- function(p){return(log(p/(1-p)))}
> plot(dose1+0.03*(runif(n)-0.5), Logit(abscised1/nplants1), cex=1.5 ,
+       xlab="Dose", ylab="Logistic transformed proportion abscised",
+       pch=19, col='blue', ylim=range(Logit(abscised/nplants)))
> prop <-tapply(abscised1, factor(dose1),sum)/ tapply(nplants1, factor(dose1),sum)
> points(levels(factor(dose1)), Logit(prop), type='b', lty=2, lwd=2.5, pch=19, col='blue')
> n <- length(dose2)
> points(dose2+0.03*(runif(n)-0.5), Logit(abscised2/nplants2), cex=1.5 , xlab="Dose", ylab="Proportion ab
> prop <- tapply(abscised2, factor(dose2),sum)/ tapply(nplants2, factor(dose2),sum)
> points(levels(factor(dose2)), Logit(prop), type='b', lty=2, lwd=2.5, pch=19, col='red')
```

Example: Leave abscission



Multiple logistic regression

- Regression model: assume that the probabilities of abscission are a (continuous) function of the dose of abscisic acid
- Y is number of plants with more than 50 % of abscised leaves
  out of the 50 plants in each batch
- d is the dose (mg/plant)
- $Y \sim Bi(50, p_d)$
- We assume **for variety** i

$$p_{di} = \frac{\exp(\alpha_i + \beta_i d)}{1 + \exp(\alpha_i + \beta_i d)}$$

- Or equivalently,

$$\log\left(\frac{p_{di}}{1 - p_{di}}\right) = \alpha_i + \beta_i d$$

- α_1, α_2, $\beta_1$and β_2 are the parameters in the model.

Fitting (simultaneously) a separate logistic regression for each variety

```
> response <- cbind(abscised  , nplants-abscised)
```

```
> diff.slopes <- glm(response ~ factor(variety) * dose, family=binomial(link='logit'))
```

Fitting (simultaneously) a separate logistic regression for each variety

```
> summary(diff.slopes)

...

                     Estimate Std. Error z value Pr(>|z|)
(Intercept)           0.42975    0.06913   6.216  5.1e-10 ***
factor(variety)2     -1.04317    0.09539 -10.936  < 2e-16 ***
dose                  3.19882    0.19739  16.206  < 2e-16 ***
factor(variety)2:dose -0.15745    0.25420  -0.619    0.536
(Dispersion parameter for binomial family taken to be 1)

...
```

Fitting (simultaneously) a separate logistic regression for each variety (Different parametrisation)

```
> # Different parametrization
>
> diff.slopes <- glm(response ~ 0 + factor(variety) * dose ,
+                    family=binomial(link='logit')
+                    )
>
```

Fitting (simultaneously) a separate logistic regression for each variety (Different parametrisation)

```
> summary(diff.slopes)

...

Coefficients:

                     Estimate Std. Error z value Pr(>|z|)
factor(variety)1      0.42975    0.06913   6.216  5.1e-10 ***
factor(variety)2     -0.61342    0.06572  -9.333  < 2e-16 ***
dose                  3.19882    0.19739  16.206  < 2e-16 ***
factor(variety)2:dose -0.15745    0.25420  -0.619    0.536
...
```

Fitting (simultaneously) a separate logistic regression for each variety (Different parametrisation)

```
> # Yet another parametrization
>
> diff.slopes <- glm(response ~ 0 + factor(variety) * dose - dose,
+                    family=binomial(link='logit')
+                    )
>
```

Fitting (simultaneously) a separate logistic regression for each variety (Different parametrisation)

```
> summary(diff.slopes)

...

Coefficients:

                     Estimate Std. Error z value Pr(>|z|)
factor(variety)1      0.42975    0.06913   6.216  5.1e-10 ***
factor(variety)2     -0.61342    0.06572  -9.333  < 2e-16 ***
factor(variety)1:dose  3.19882    0.19739  16.206  < 2e-16 ***
factor(variety)2:dose  3.04138    0.16018  18.988  < 2e-16 ***

...
```

Fitting a parallel model

```
> # Fitting a parallel model. That is different intercepts and common slope
> # in the lodds scale
>
> parallel <- glm(response ~ factor(variety) + dose,
+                   family=binomial(link='logit')
+                   )
>
```

Fitting a parallel model

```
> summary(parallel)

Call:

glm(formula = response ~ factor(variety) + dose, family = binomial(link = "logit"))

...

Coefficients:

                 Estimate Std. Error z value Pr(>|z|)

(Intercept)       0.45503    0.05595   8.133 4.17e-16 ***

factor(variety)2 -1.08982    0.05875 -18.549  < 2e-16 ***

dose              3.10427    0.12437  24.960  < 2e-16 ***

---

...
```

Fitting a parallel model (other parametrization)

```
> # Using another parametrization
>
>  parallel <- glm(response ~ 0 + factor(variety) + dose,
+                   family=binomial(link='logit')
+                   )
>
```

Fitting a parallel model (other parametrization)

```
> summary(parallel)

Call:

glm(formula = response ~ 0 + factor(variety) + dose, family = binomial(link = "logit"))

...

Coefficients:

                  Estimate Std. Error z value Pr(>|z|)

factor(variety)1  0.45503    0.05595   8.133 4.17e-16 ***

factor(variety)2 -0.63478    0.05607 -11.321  < 2e-16 ***

dose              3.10427    0.12437  24.960  < 2e-16 ***

---

...
```

Fitting a "free curve model"

```
> # Fitting a free curve model, attributing one lodds (or equiv. probability)
> # for each combination of variety and dose
>
>  free.curve <- glm(response ~ 0 + factor(variety) * factor(dose),
+                   family=binomial(link='logit')
+                   )
>
```

Fitting a "free curve model"

```
> summary(free.curve)

...

Coefficients:

                                  Estimate Std. Error z value Pr(>|z|)
factor(variety)1                  0.430529   0.091523   4.704 2.55e-06 ***
factor(variety)2                 -0.610260   0.093639  -6.517 7.17e-11 ***
factor(dose)0.1                   0.323243   0.132543   2.439   0.0147 *
...
factor(dose)0.7                   2.286086   0.206801  11.055  < 2e-16 ***
factor(variety)2:factor(dose)0.1 -0.002999   0.185756  -0.016   0.9871
...
factor(variety)2:factor(dose)0.7 -0.032405   0.257424  -0.126   0.8998
--- ...
```

Testing homogeneity

```
> # Testing the homogeneity assumption
> deviance(free.curve)
[1] 106.1639
> length(dose)              # 140 observations
[1] 140
> length(coef(free.curve))  # 14 parameters
[1] 14
> pchisq(deviance(free.curve), df=140-4, lower.tail=F)
[1] 0.9725822
> # Uffa! Passed this test!
```

Testing the form of the regression curve

```
> # Testing the apropriatness of the logistic curve used
> anova(diff.slopes, free.curve, test="Chisq")
Analysis of Deviance Table

Model 1: response ~ 0 + factor(variety) * dose - dose
Model 2: response ~ 0 + factor(variety) * factor(dose)
  Resid. Df Resid. Dev  Df Deviance P(>|Chi|)
1       136    109.473
2       126    106.164  10    3.309     0.973
> # Passes this test also!
```

Testing homogeneity AND the form of the regression curve

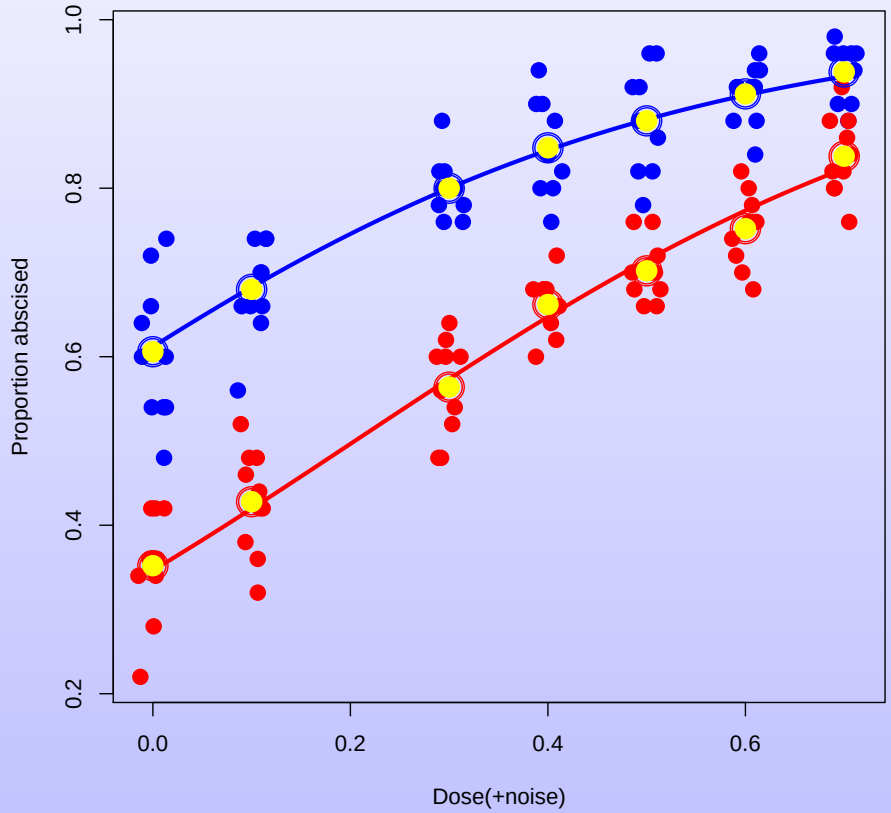
```
> saturated <-  glm(response ~ 0 + factor(1:140),  family=binomial(link='logit') )
> deviance(saturated)
[1] -6.661338e-15
> anova(diff.slopes, saturated, test="Chisq")
Analysis of Deviance Table

Model 1: response ~ 0 + factor(variety) * dose - dose
Model 2: response ~ 0 + factor(1:40)
  Resid. Df Resid. Dev  Df Deviance P(>|Chi|)
1       136    109.473
2         0 -6.661e-15 136  109.473     0.954
```

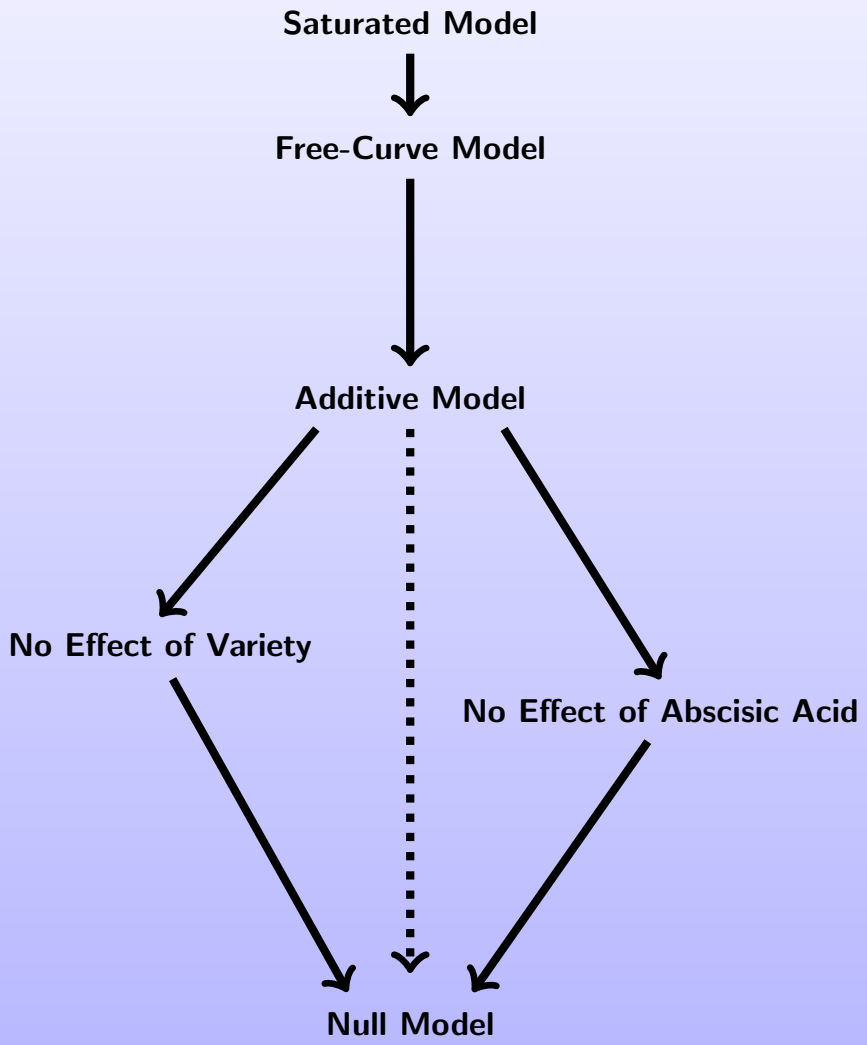
Testing additivity

```
> # Testing additivity, i.e. parallelism of the two curves in the lodds scale
>
> anova(parallel, diff.slopes, test="Chisq")
Analysis of Deviance Table

Model 1: response ~ 0 + factor(variety) + dose
Model 2: response ~ 0 + factor(variety) * dose - dose
  Resid. Df Resid. Dev  Df Deviance P(>|Chi|)
1       137    109.858
2       136    109.473   1    0.384     0.535
```


Proportion abscised
1.0
0.8
0.6
0.4
0.2
0.0
0.2
0.4
0.6
Dose(+noise)

Summarising the analysis performed



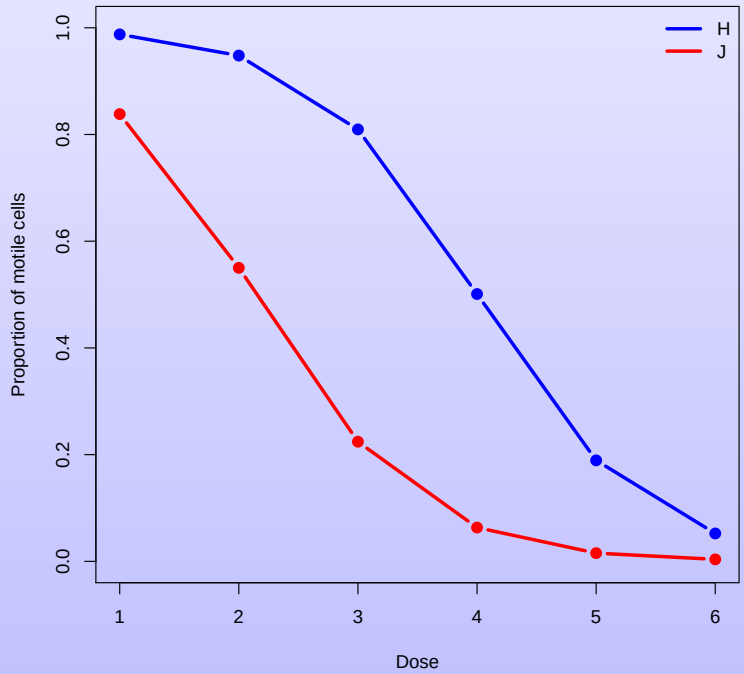
Example: Bulls' Fertility

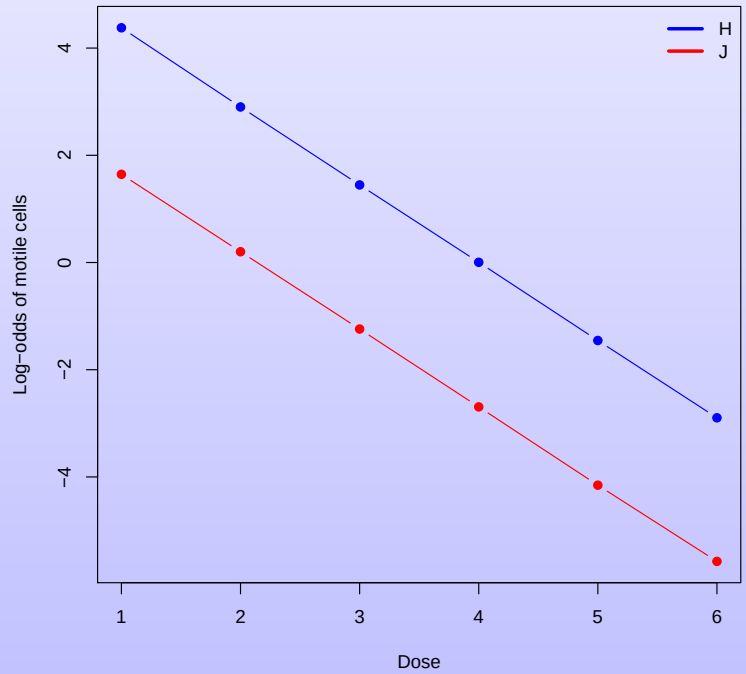
- Large experiment on bulls' seminal viability (492 samples of semen)
- Treatment involving several doses of a drug:
  1,2,3,4,5, and 6 ppm
- The semen was collected from bulls of two breeds (denoted by "J" and "H")
- Analysis using a flow-cytometer, which counted a large number of spermatozoa (in mean 7504 cells were counted per sample) and classified each cell as alive or not using a colouring indicator of cell respiration.
- The total numbers of cells counted for each sample are registered in the variable "N" and the total of alive cells are registered in the variable "Alive".
- Question: Does the viability (*i.e.* the proportion of alive sperm cells) decay with the dose? Is the decay the same for the two breeds?

Example: Bulls' Fertility

```
> str(BullSeminalViabilityALL)
'data.frame':        492 obs. of  4 variables:
 $ Breed: Factor w/ 2 levels "H","J": 1 1 1 1 1 1 1 1 1 1 ...
 $ Dose : num  1 2 3 4 5 6 1 2 3 4 ...
 $ N    : num  7583 7427 7554 7746 7478 ...
 $ Alive: num  7474 7045 6166 3845 1427 ...
> mean(N)
[1] 7504.098
> table(Breed, Dose)
     Dose
Breed  1  2  3  4  5  6
    H 41 41 41 41 41 41
    J 41 41 41 41 41 41
```

Example: Bulls' Fertility





Example: Bulls' Fertility

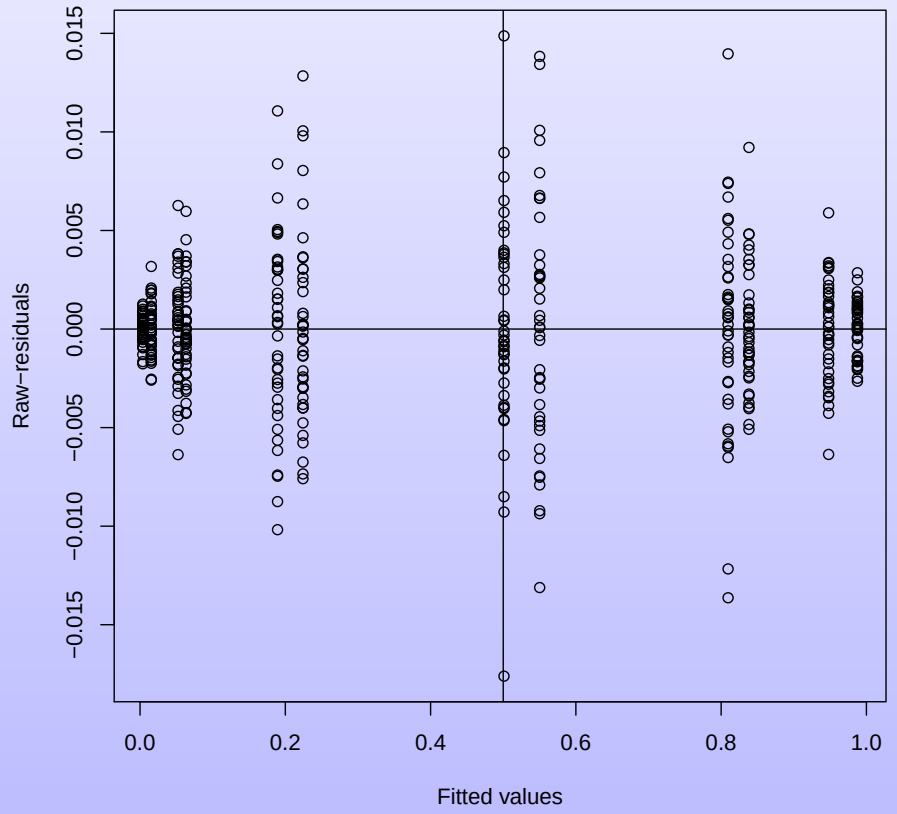
```
> resp <- cbind(Alive, N-Alive)
> resp[1:5, ]
     Alive
[1,]  7474  109
[2,]  7045  382
[3,]  6166 1388
[4,]  3845 3901
[5,]  1427 6051
>
> fit1 <- glm(resp ~ Breed*factor(Dose),
+             family=binomial(link="logit"))
```

Bulls' Fertility: The corner point parametrisation

```
> summary(fit1)

Coefficients:      Estimate Std. Error  z value Pr(>|z|)
(Intercept)         4.37864    0.01631  268.488   <2e-16 ***
BreedJ             -2.73346    0.01703 -160.525   <2e-16 ***
factor(Dose)2      -1.47700    0.01822  -81.082   <2e-16 ***
factor(Dose)3      -2.93148    0.01694 -173.038   <2e-16 ***
factor(Dose)4      -4.37452    0.01670 -261.916   <2e-16 ***
factor(Dose)5      -5.83317    0.01694 -344.244   <2e-16 ***
factor(Dose)6      -7.27526    0.01821 -399.628   <2e-16 ***
BreedJ:factor(Dose)2  0.03308    0.01921    1.722   0.0851 .
BreedJ:factor(Dose)3  0.04544    0.01816    2.503   0.0123 *
BreedJ:factor(Dose)4  0.03615    0.01891    1.911   0.0560 .
BreedJ:factor(Dose)5  0.03421    0.02291    1.493   0.1354
BreedJ:factor(Dose)6  0.05500    0.03494    1.574   0.1155
```

Bulls' Fertility: Raw residuals



Bulls' Fertility: Pearson residuals

- Raw-residuals: For $i = 1, \ldots, 462$,
  $R_i = y_i - n_i \hat{p}_i$,

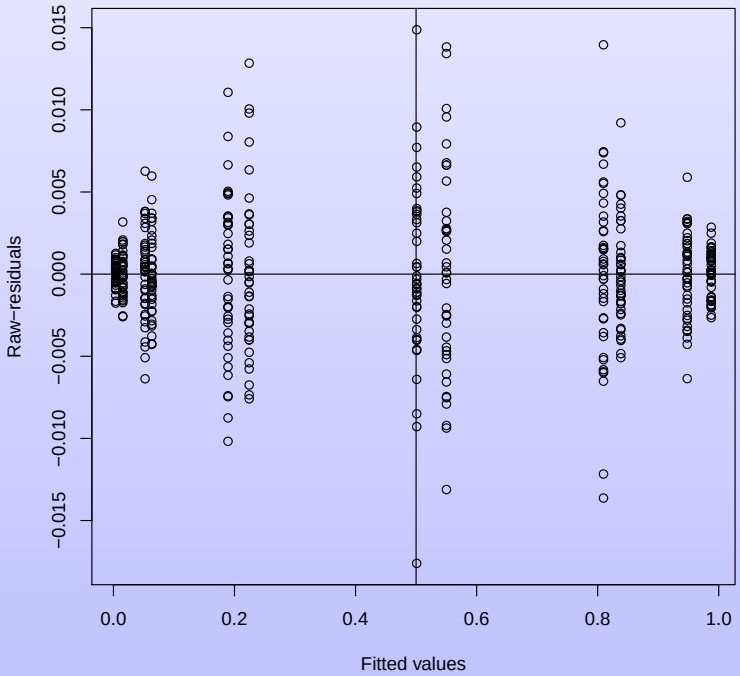
  where R_i is the raw-residual of the ith observation,

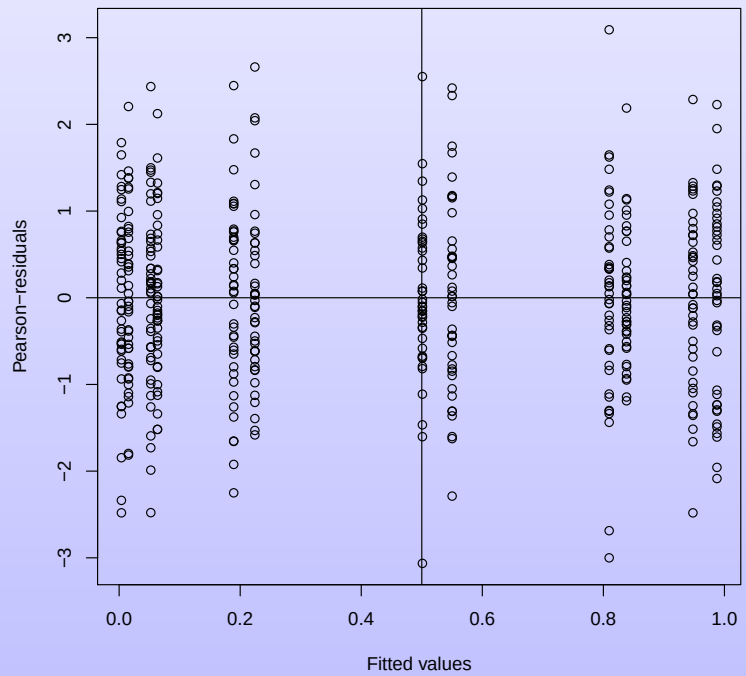
  y_i and n_i are the number of successes and the number of trials and $\hat{p}_i$ is the predicted probability of success (fitted values)

- $Var(R_i) = n_i \hat{p}_i (1 - \hat{p}_i)$

- Pearson residuals:
  $P_i = \frac{y_i - n_i \hat{p}_i}{\sqrt{n_i p_i (1 - \hat{p}_i)}}$

Example: Bulls' Fertility





Example: Bulls' Fertility: Testing homogeneity

```
> deviance(fit1)
[1] 482.0306
> n.observations <- length(N) ; n.observations
[1] 492
> n.parameters.fit1 <- length(coef(fit1)); n.parameters.fit1
[1] 12
> n.df.fit1 <- n.observations - n.parameters.fit1; n.df.fit1
[1] 480
> pchisq(deviance(fit1), df=n.df.fit1, lower.tail=F)
[1] 0.4653527
```

Bulls' Fertility: Fitting two logistic curves

```
> fit2 <- glm(resp ~ 0 + Breed - Dose + Breed:Dose, family=binomial)
> summary(fit2)

glm(formula = resp ~ 0 + Breed - Dose + Breed:Dose, family = binomial)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-3.14096  -0.64440   0.00531   0.66139   2.95600

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
BreedH       5.806178   0.008187   709.2   <2e-16 ***
BreedJ       3.093313   0.005635   548.9   <2e-16 ***
BreedH:Dose -1.451377   0.001999  -726.0   <2e-16 ***
BreedJ:Dose -1.446014   0.002202  -656.6   <2e-16 ***
---
```

Bulls' Fertility: Fitting two logistic curves

```
> summary(fit2)

Call:

glm(formula = resp ~ Breed + Dose + Breed:Dose, family = binomial)

Deviance Residuals:

     Min        1Q    Median        3Q       Max

-3.14096  -0.64440   0.00531   0.66139   2.95600

Coefficients:

             Estimate Std. Error  z value Pr(>|z|)

(Intercept)  5.806178   0.008187  709.166   <2e-16 ***

BreedJ      -2.712865   0.009939 -272.945   <2e-16 ***

Dose        -1.451377   0.001999 -725.990   <2e-16 ***

BreedJ:Dose  0.005363   0.002974    1.803   0.0714 .
```

Bulls' Fertility

```
> fit3 <- glm(resp ~ 0 + Breed + Dose, family=binomial)
> summary(fit3)
Call:
glm(formula = resp ~ 0 + Breed + Dose, family = binomial)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0629  -0.6416   0.0149   0.6890   3.0331
Coefficients:
        Estimate Std. Error z value Pr(>|z|)
BreedH  5.796643   0.006239   929.1   <2e-16 ***
BreedJ  3.100214   0.004141   748.6   <2e-16 ***
Dose   -1.448959   0.001480  -978.9   <2e-16 ***
```

Bulls' Fertility

```
> fit3 <- glm(resp ~ Breed + Dose, family=binomial)
> summary(fit3)
Call:
glm(formula = resp ~ Breed + Dose, family = binomial)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0629  -0.6416   0.0149   0.6890   3.0331
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  5.796643   0.006239   929.1   <2e-16 ***
BreedJ      -2.696428   0.003948  -682.9   <2e-16 ***
Dose        -1.448959   0.001480  -978.9   <2e-16 ***
---
```

Bulls' Fertility

```
> anova(fit3, fit2, test="Chisq")

Analysis of Deviance Table

Model 1: resp ~ Breed + Dose
Model 2: resp ~ 0 + Breed - Dose + Breed:Dose
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       489     493.21
2       488     489.96  1   3.2502  0.07141 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Bulls' Fertility

```
> fit4 <- glm(resp ~ Breed, family=binomial)
> anova(fit4, fit3, test="Chisq")
Analysis of Deviance Table
Model 1: resp ~ Breed
Model 2: resp ~ Breed + Dose
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1       490    2157370
2       489        493  1  2156877 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Bulls' Fertility

```
> fit5 <- glm(resp ~ Dose, family=binomial)
> anova(fit5, fit3, test="Chisq")
Analysis of Deviance Table
Model 1: resp ~ Dose
Model 2: resp ~ Breed + Dose
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1       490     659408
2       489        493  1   658915 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

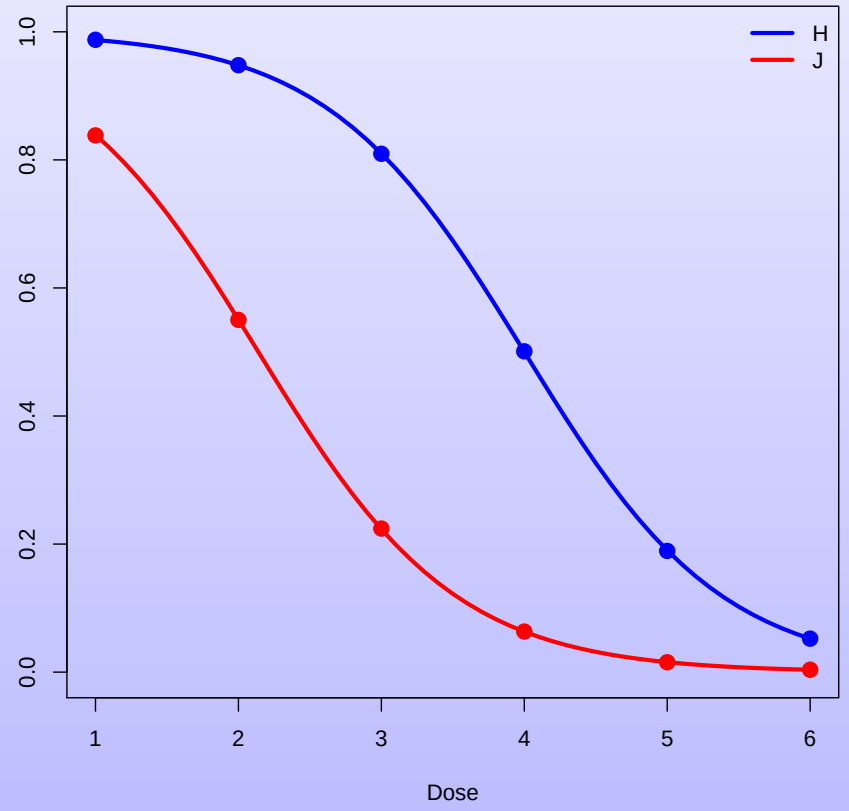
Bulls' Fertility

```
> fit6 <- glm(resp ~ 1, family=binomial)
> anova(fit6, fit3, test="Chisq")
Analysis of Deviance Table
Model 1: resp ~ 1
Model 2: resp ~ Breed + Dose
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1       491    2497923
2       489        493  2  2497430 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Bulls' Fertility

```
> anova(fit6, fit1, test="Chisq")
Analysis of Deviance Table
Model 1: resp ~ 1
Model 2: resp ~ Breed * factor(Dose)
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1       491    2497923
2       480        482 11  2497441 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

Example: Bulls' Fertility



The glm function in R

- The glm function fits **G**eneralized **L**inear **M**odels, a large class of models.

- Examples of Generalized Linear Models:

  Binomial one/two/three ... way models, Binomial regressions, Poisson models, log-linear models (contingency tables), normal linear regression, normal anova, analysis of covariance models, gamma models, inverse gaussian models, some survival models, etc.

- Call: glm (formula= , family= , ...)
  Two important parts: formula and family

Generalized Linear Models

- Generalized Linear Models is a class of statistical models.
- A response variable, Y and
  a collection of explanatory variables, $X_1, \ldots, X_k$.
- The model specifies that Y follows a given probability laws and that the expectation of Y is related to the explanatory variables by

$$g\left(E(Y)\right) = \alpha + \beta_1 X_1 + \cdots + \beta_k X_k,$$

  here g is a given function called the link function.

Specification of the family in glm

- The "family" parameter in glm specifies two characteristics of the generalized linear model:
  The class of probability laws and the link function.
- Common used distributions: Normal, Poisson, Binomial, Gamma, etc.
- Link function.

$$g\left(E(Y)\right) = \alpha + \beta_1 X_1 + \cdots + \beta_k X_k,$$

Common link functions: identity, log, inverse, logit, probit, etc.

identity: $E(Y) = \alpha + \beta_1 X_1 + \cdots + \beta_k X_k$

log: $\log\left(E(Y)\right) = \alpha + \beta_1 X_1 + \cdots + \beta_k X_k$

inverse: $\frac{1}{E(Y)} = \alpha + \beta_1 X_1 + \cdots + \beta_k X_k$

Some common family specifications:

```
binomial(link = "logit")

binomial(link = "probit")

binomial(link = "cloglog")

gaussian(link = "identity")

Gamma(link = "inverse")

inverse.gaussian(link = "1/mu^2")

poisson(link = "log")
```

Can specify your own family

(but we will not do that at this stage)

The formula in glm

- glm (formula= , family= , ...)
- The formula specifies the response variable, the explanatory variables

  and the way the explanatory variables act on the expectation of the response variable.
- The general form is:

$$Y \sim X_1 \sharp \ldots \sharp X_k$$

  Here Y is the response variable (or matrix)
  $X_1, \ldots, X_k$ are the explanatory variables
  $\sharp$ are "operators" connecting the variables
- Possibilities for the operator $\sharp$: "+", "*" and ":"

The formula in glm: defining models with discrete explanatory variables

- Convention: y response variable, A and B factors.

- Single classification analysis of variance model of y, with classes determined by A.

```
y ~ A
```

- Single classification analysis of variance model of the logarithm transformed y, i.e. log(y), with classes determined by A.

```
log(y) ~ A
```

- Single classification analysis of variance model of the square-root transformed variable y, i.e. sqrt(y), with classes determined by A.

```
sqrt(y) ~ A
```

- Two factor additive model of y on A and B.

```
y ~ A + B
```

- Two factor non-additive model of y on A and B.

```
y ~ A*B
y ~ A + B + A:B
```

The formula in glm: defining models with continuous explanatory variables

- Convention: y response variable, A and B factors, x, z, and y numeric variable.
- Simple linear regression model of y on x.

  `y ~ x` or `y ~ 1 + x`
- Simple linear regression of y on x through the origin (that is, without an intercept term).

  `y ~ 0 + x` or `y ~ -1 + x` or `y ~ x - 1`
- Multiple regression of the transformed variable, log(y), on x1 and x2 (with an implicit intercept term).

  `log(y) ~ x1 + x2`
- Single classification analysis of covariance model of y, with classes determined by A, and with covariate x.

  `y ~ A + x`
- Separate simple linear regression models of y on x within the levels of A.

  `y ~ A * x`

Exercises and tutorials

- Please, run (critically) and discuss the tutorial Tutorial-12-SimpleBinomialRegression
- Exercises: 3.4
- Additional exercises: 3.5 and 3.6

  (there I am not guiding step by step, you are free ...)