GLMs: binomial family

Mandy Vogel

University Leipzig

July 25, 2016

Overview

Table of Contents I

Beyond Linear Models

- linear models are central to the practice of statistics
- the standard linear model cannot handle non-normal responses, such as counts or proportions. This motivates the development of generalized linear models that can represent categorical, binary and other response types.

Beyond Linear Models

- Some data has a grouped, nested or hierarchical structure. Repeated measures, longitudinal and multilevel data consist of several observations taken on the same individual or group. This induces a correlation structure in the error. mixed effect models allow the modeling of such data.
- non-parametric regression models: Methods such as additive models, trees and neural networks allow a more flexible regression modeling of the response that combine the predictors in a nonparametric manner.

Generalized Linear Models

Linear modeling assumes constant variance and normally distributed errors. Certain kinds of respond variables lack these constraints. GLMs are excellent at dealing with it.

Input/Output

```
> m1 <- lm(bweight ~ hyp, data=births)
> m2 <- glm(bweight ~ hyp, family=gaussian, data=births)
```

give the same answer. The model formula is the same for both, but for `glm()` it is necessary to specify the family of likelihoods which will be used to fit the model.
The `glm()` function allows us to fit other models including logistic regression and Poisson regression.

Beyond Linear Models

- We begin with a binary response variable:

Bernoulli model

- $f(y;p) = p^y(1-p)^{1-y}$
- it is modelled with a logit as canonical link

$$\eta = \log(\frac{p}{1-p})$$

- i.e. our linear model looks like

$$\eta = \log(\frac{p}{1-p}) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n + \epsilon$$

with a binomial error structure

Data Structure

Load the data birthsweights.rdata . The structure of the data should be of the following form:

Input/Output

```
> str(births)
'data.frame': 500 obs. of  8 variables:
$ id      : num  100 101 102 103 104 105 106 107 108 109 ..
$ preterm: chr  "normal" "normal" "normal" "normal" ...
$ gestwks: num  39.8 39 38.1 39.5 39.5 ...
$ hyp     : chr  "normal" "normal" "normal" "normal" ...
$ matage : num  33 32 33 38 40 29 32 40 41 39 ...
$ bweight: num  3576 3784 2796 3226 3138 ...
$ lowbw   : chr  "normal" "normal" "normal" "normal" ...
$ sex     : chr  "F" "F" "F" "F" ...
```

Data from: Michæl Hills and Bianca De Stavola (2002). A Short Introduction to Stata 8 for Biostatistics, Timberlake Consultants Ltd URL: http://www.timberlake.co.uk

Binary Response Variable

Many statistical problems involve binary response variables. For example, we often classify individuals as:

- dead or alive,
- occupied or empty,
- healthy or diseased,
- wilted or turgid,
- male or female,
- literate or illiterate,
- mature or immature,
- solvent or insolvent, or
- employed or unemployed.

Binary Response Variable

Question

Which variable in the births data set is (most) suitable to use as binary response given this data set? Why?

Predicting Low Birth Weight

- Now we are more interested in predicting birth weight under 2500g (`lowbw`).
- This requires a model where the outcome is not metric, but binary.
- For a binary response we use a `glm()` with a binomial family.
- the binomial family uses a logit link as default

Predicting Low Birth Weight

How it looks in R:

Input/Output

```
> m <- glm(lowbw ~ hyp, family=binomial, data=births)
> summary(m)
Call:
glm(formula = lowbw ~ hyp, family = binomial, data = births)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.8067  -0.4430  -0.4430  -0.4430   2.1773

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.2721     0.1661 -13.682  < 2e-16 ***
hyphyper      1.3166     0.3111   4.232 2.32e-05 ***
---

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 366.92  on 499  degrees of freedom
Residual deviance: 350.84  on 498  degrees of freedom
AIC: 354.84
```

Predicting Low Birth Weight

What it looks like as a math formula:

$$\log\left(\frac{\text{Pr(lowbw)}}{1-\text{Pr(lowbw)}}\right) = \beta_0 + \beta_1 \cdot \text{hyp} + \epsilon$$

Interpreting the Cœfficients

- While using a binomial family R uses a logit as link function.
- Therefore the returned estimates are log odds (Intercept) or log odds ratios (for the parameters).
- The `arm` package contains a function `invlogit()` which dœs invert the logit function.
- Alternatively you can use the formula

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

Interpreting the Cœfficients

- Our example is a simple analysis of variance.
- Our model here is

$$\Pr(\text{lowbw}) = \text{logit}^{-1}(-2.2721 + 1.3166 \cdot \text{hyp})$$

- We have two levels of our predictor variable `hyp`: normal and hyp.
- For the reference level normal hyp $= 0$
- in this case we get

$$\Pr(\text{lowbw}) = \text{logit}^{-1}(-2.2721 + 1.3166 \cdot 0) = \text{logit}^{-1}(-2.2721)$$

which is a log odds as mentioned before, so

Input/Output

```
> invlogit(coef(m)[1])
(Intercept)
 0.09345794
```

Interpreting the Cœfficients

- The result is the probability of low birth weight within the group of moms with normal blood pressure. We can check this by using `table`:

Input/Output

```
> table(births$lowbw,births$hyp)

        normal hyper
  normal    388    52
  low        40    20
> 40/(388+40)
[1] 0.09345794
```

Interpreting the Cœfficients

- for the level hyp (i.e. hyp $= 1$) we get a difference of 1.3166 on the logit scale

$$\Pr(\text{lowbw}) = \text{logit}^{-1}(-2.2721 + 1.3166 \cdot 1)$$

- which turns out to be

Input/Output

```
> invlogit(coef(m)[1]+coef(m)[2])
(Intercept)
  0.2777778
```

- so the probability for low birth weight is 27.8% in for moms with high blood pressure

Understanding the Cœfficients

- in this simple case, the response variable gives the probability for low birth weight for each of the two groups of moms (with and without high blood pressure)
- you can get the result also using (a) a proportion test:

```
> prop.test(c(20,40),c(72,428))

2-sample test for equality of proportions with continuity

data:  c(20, 40) out of c(72, 428)
X-squared = 18.121, df = 1, p-value = 2.073e-05
alternative hypothesis: two.sided
95 percent confidence interval:
 0.06913673 0.29950294
sample estimates:
    prop 1     prop 2
0.27777778 0.09345794
```

Understanding the Cœfficients

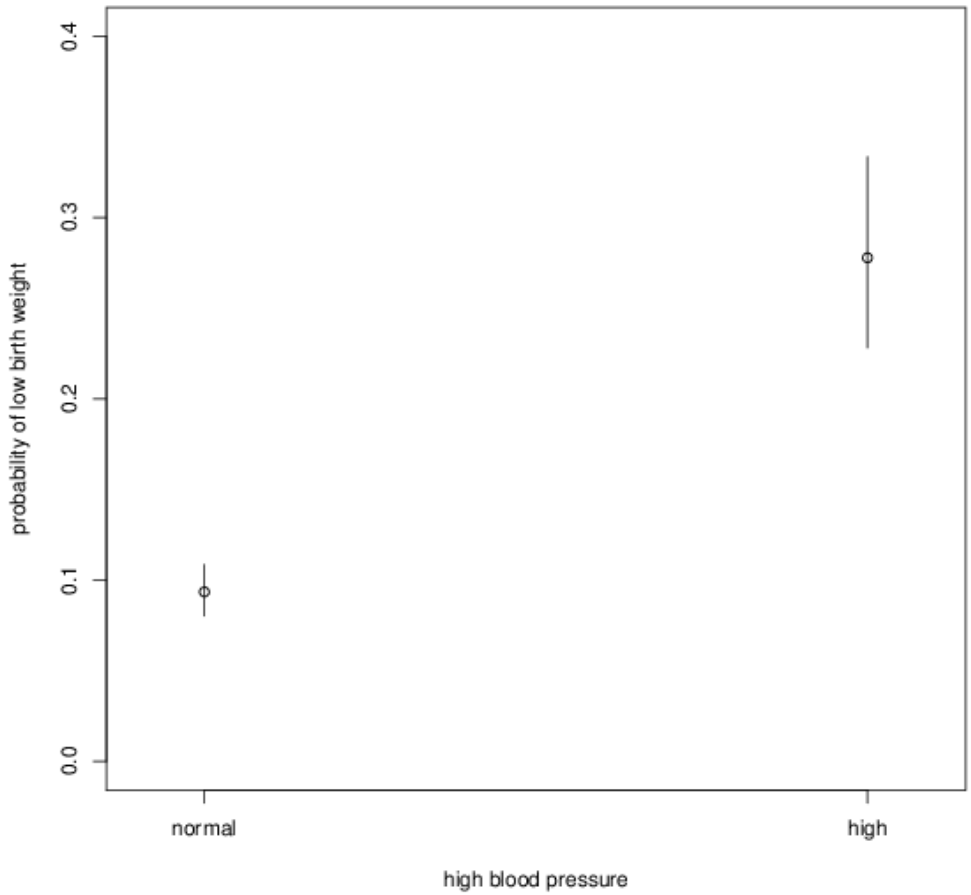
- or (b) a χ^2-test:

Input/Output

```
> chisq.test(table(births$lowbw,births$hyp))

Pearson's Chi-squared test with Yates' continuity correction

data:  table(births$lowbw, births$hyp)
X-squared = 18.121, df = 1, p-value = 2.073e-05
```

Understanding the Cœfficients

- a hand made plot

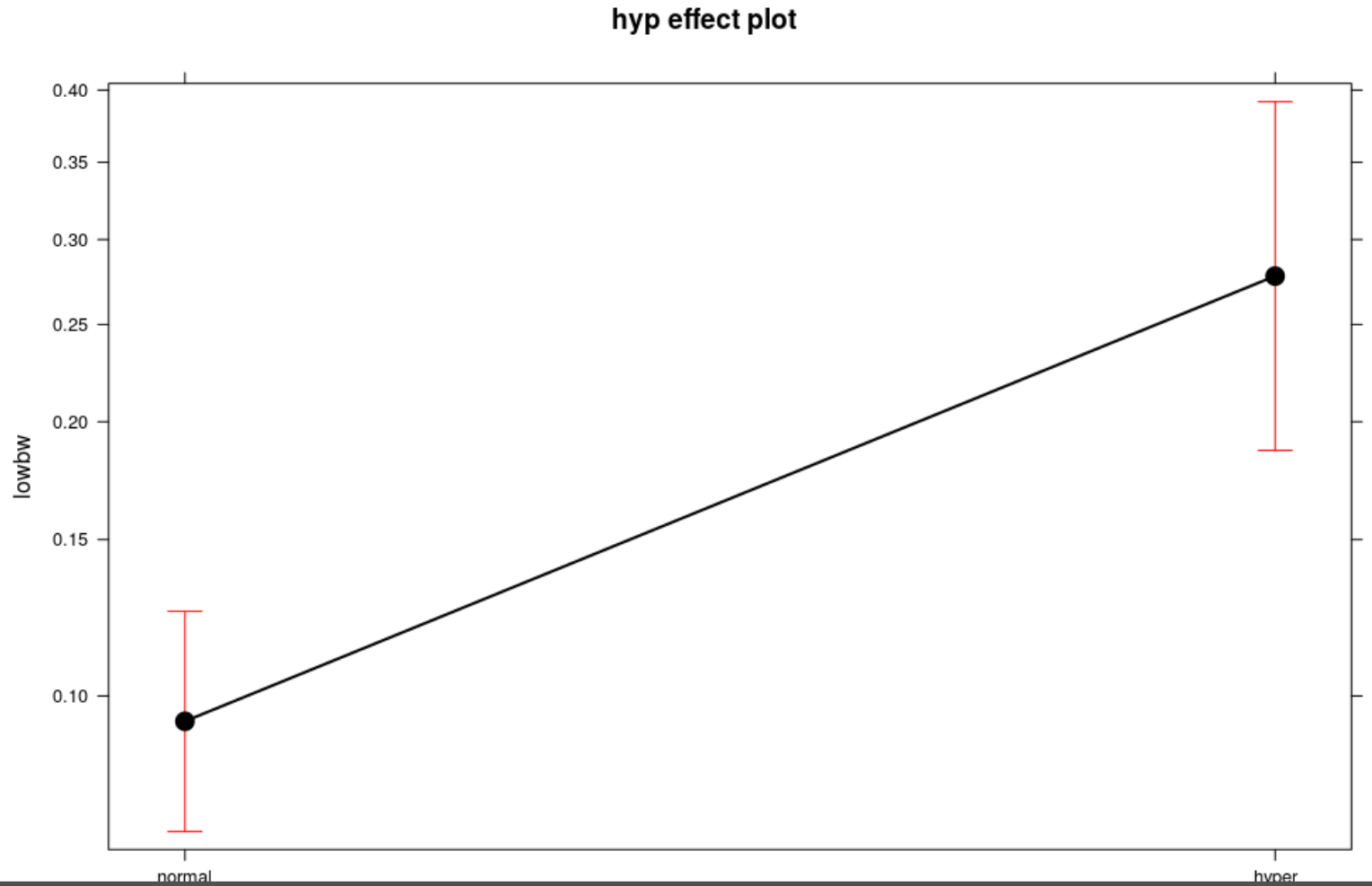


Understanding the Cœfficients

- and one effect plot (`effects` package)

Input

```
> plot(Effect("hyp",m))
```



Understanding the Cœfficients

- btw: `Effect()` gives you the probabilities without using a explicit transformation

Input/Output

```
> Effect("hyp",m)

 hyp effect
hyp
    normal      hyper
0.09345794 0.27777778
```

Controlling

Controlling the effect of `hyp` on `lowbw` for `sex`

Input/Output

```
> m2 <- glm(lowbw ~ hyp+sex, family=binomial, data=births)

             Estimate   StdErr  Pr(>|z|)
(Intercept)  -2.5088   0.2331  < 2e-16 ***
hyphyper      1.3625   0.3144 1.47e-05 *** hyp controlled for
sexF          0.4473   0.2843    0.116     sex controlled for
```

When you control for a variable you are assuming that any interaction can be ignored.

Interaction (effect modification)

- We add an interaction term to the model

Input/Output

```
> m3 <- glm(lowbw ~ hyp + sex + hyp:sex,
+           family=binomial, data=births) # or shorter
> m3 <- glm(lowbw ~ hyp*sex, family=binomial,
            data=births)
```

Interaction (effect modification)

- we have four estimates now, and to get the effects in terms of probabilites we need to type

Input/Output

```
> m3 <- glm(lowbw ~ hyp*sex, family=binomial, data=births)
> summary(m3)

Call:
glm(formula = lowbw ~ hyp * sex, family = binomial, data = births)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.8090  -0.5074  -0.3749  -0.3749   2.3195

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -2.6198     0.2674  -9.796  < 2e-16 ***
hyphyper       1.6707     0.4326   3.862 0.000112 ***
sexF           0.6347     0.3421   1.855 0.063535 .
hyphyper:sexF -0.6507     0.6366  -1.022 0.306694
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)
```

Interaction Cœfficients

Input/Output

```
> invlogit(coef(m3)[1])
  (Intercept)
   0.0678733
> invlogit(coef(m3)[1] + coef(m3)[2])
  (Intercept)
   0.2790698
> invlogit(coef(m3)[1] + coef(m3)[3])
  (Intercept)
   0.1207729
> invlogit(coef(m3)[1] + coef(m3)[2] + coef(m3)[3] + coef(m3)[4])
  (Intercept)
   0.2758621
```

Exercises

You can calculate the effects by hand and using the `invlogit()` function, but this becomes a little annoying, the `allEffects()` function provides a nicer way to do the same.

- now you have three models, use the `Effects()`, `allEffects()` and the `plot()` function to get the following information:
  1. the estimated probability for moms with hypertension to get a baby with low birth weight for all three models
  2. is their a difference in effects between boys and girls? Which model can answer this question?

Testing for Interaction

- Do we need to keep the interaction term?

Input/Output

```
> m2 <- glm(lowbw ~ hyp+sex, family=binomial,
+                            data=births)
> m3 <- glm(lowbw ~ hyp*sex, family=binomial,
+                            data=births)
> anova(m2,m3,test="Chisq")

  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       497     348.34
2       496     347.29  1   1.0561   0.3041
```

- The anova function conducts an analysis of variance – a test of significance between two nested models.
- The interaction term dœs not improve the fit - so we leave it out and keep the simpler model.

Stratified Effects

- When there is a strong interaction it may be best to report stratified effects.
- Omitting the main effect of `hyp` in an interaction model gives us the effect of `hyp` within strata of `sex`.

Stratified Effects

Input/Output

```
> m4 <- glm(lowbw ~ sex + sex:hyp, family=binomial, data=birth
> summary(m4)

Call:
glm(formula = lowbw ~ sex + sex:hyp, family = binomial, data =

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.8090  -0.5074  -0.3749  -0.3749   2.3195

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -2.6198     0.2674  -9.796  < 2e-16 ***
sexF           0.6347     0.3421   1.855 0.063535 .
sexM:hyphyper  1.6707     0.4326   3.862 0.000112 ***
sexF:hyphyper  1.0200     0.4670   2.184 0.028952 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’
```

Stratified Effects

A slightly shorter way to define the same model:

Input/Output

```
> m4 <- glm(lowbw ~ sex/hyp, family=binomial, data=births)
> m4

Call:  glm(formula = lowbw ~ sex/hyp, family = binomial, data

Coefficients:
  (Intercept)           sexF  sexM:hyphyper  sexF:hyphyper
      -2.6198         0.6347         1.6707         1.0200

Degrees of Freedom: 499 Total (i.e. Null);  496 Residual
Null Deviance:      366.9
Residual Deviance: 347.3  AIC: 355.3
```

Exercise

- compare the effects in `m3` and `m4`

Understanding the Cœfficients

```
> ftable(births$hyp,
+        births$sex,
+        births$lowbw)
           normal low

normal M      206  15
       F      182  25
hyper  M       31  12
       F       21   8
```

```
## male/normal bp
> 15/(206+15)
[1] 0.0678733
## female/normal bp
> 25/(25+182)
[1] 0.1207729
## male/high bp
> 12/(12+31)
[1] 0.2790698
## female/high bp
> 8/(8+21)
[1] 0.2758621
```

Simple Logistic Regression

- now we model the probability of low birth weight dependent on gestational age
- so the model in R is

Input

```
> m5 <- glm(lowbw ~ gestwks, family=binomial, data=births)
```

- and as math formula

$$\log\left(\frac{\text{Pr(lowbw)}}{1-\text{Pr(lowbw)}}\right) = \beta_0 + \beta_1 \cdot \text{gestwks} + \epsilon$$

Simple Logistic Regression

- where the output look similar to the output above

Input/Output

```
> summary(m5)

Call:
glm(formula = lowbw ~ gestwks, family = binomial, data = births)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0873  -0.3623  -0.2223  -0.1369   2.9753

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  31.8477     4.0574   7.849 4.18e-15 ***
gestwks      -0.8965     0.1084  -8.272  < 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: 360.38  on 489  degrees of freedom
Residual deviance: 205.75  on 488  degrees of freedom
  (10 observations deleted due to missingness)
```

Understanding the Cœfficients

- this relationship is described by

$$\Pr(\text{lowbw}) = \text{logit}^{-1}(31.8477 + -0.8965 \cdot \text{gestwks})$$

- the intercept

Input/Output

```
> invlogit(coef(m)[1])
(Intercept)
  1
```

is interpretable as the probability for a low birth weight at a hypothetical gestational age of 0 (which makes no sense because it lies outside the range of gestational ages in our data)

- the parameter for `gestwks` describes how fast the probability decreases with increasing gestation age

Understanding the Cœfficients

$$\Pr(\text{lowbw}) = \text{logit}^{-1}(31.8477 + -0.8965 \cdot \text{gestwks})$$

- the cœfficient for `gestwks` is best interpretable if we use it as argument to the exponential function

Input/Output

```
> exp(coef(m5)[2])
   gestwks
0.4080114
```

this way it is interpretable as odds ratio for low birth weight for a difference of 1 week of gestational age

Exercise

1. here is a example for the `Effects()` command for regression

Input/Output

```
> Effect("gestwks",m5)

 gestwks effect
gestwks
       25         30         35         40
0.99992022 0.99299324 0.61574996 0.01779725
> Effect("gestwks",m5,xlevels = list(gestwks = c(20,30,40)))

 gestwks effect
gestwks
       20         30         40
0.99999910 0.99299324 0.01779725
```

2. use the command to gain the estimated probability of low birth weight for a gestational age of 27 and 36 weeks

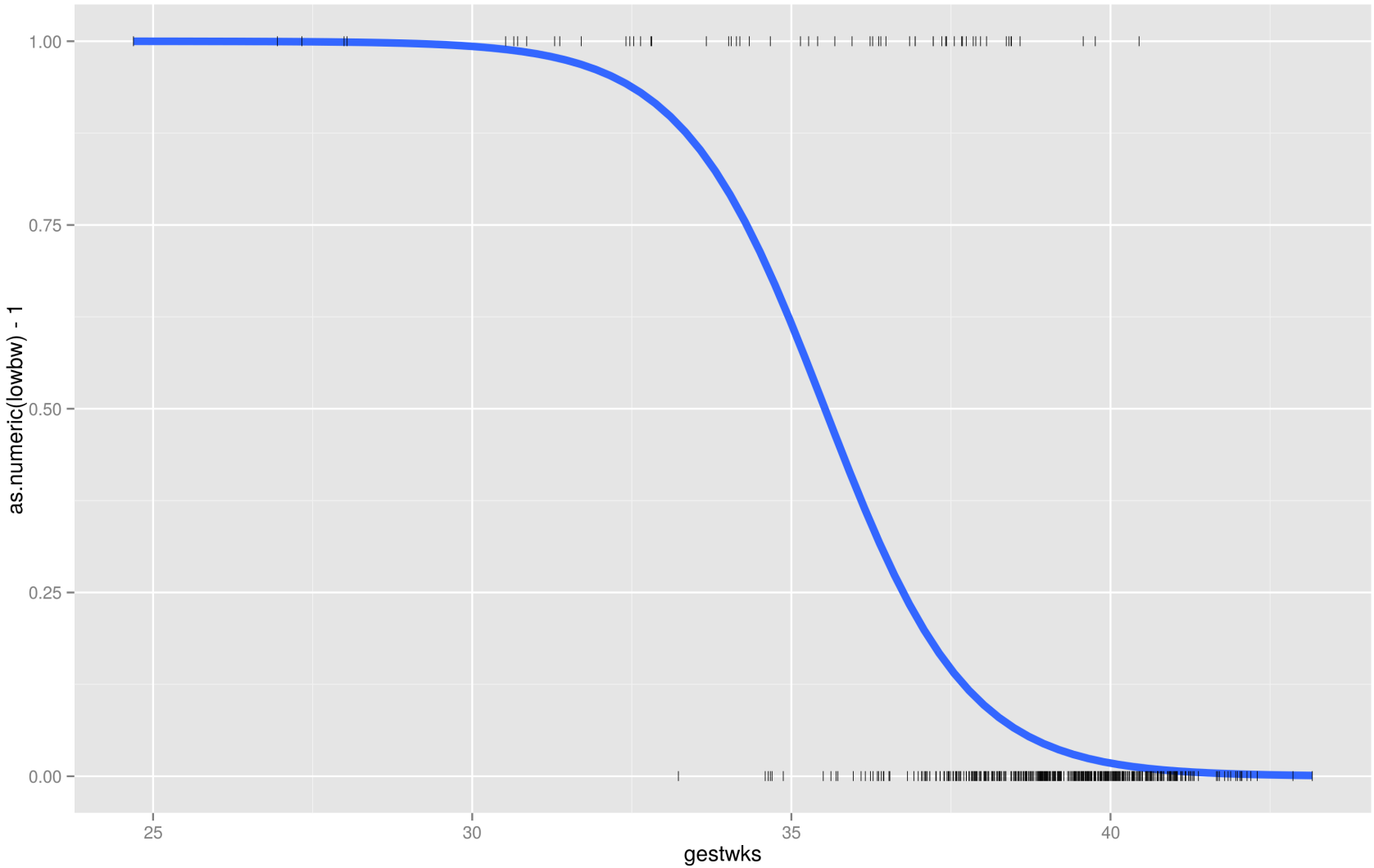
ggplot() and glm()

- ggplot2 knows also glms
- unfortunately the y-variable needs to be coded in 0s and 1s, but we can do this on the fly with `as.numeric()`

Input

```
> require(ggplot2)
> ggplot(births,aes(x = gestwks, y = as.numeric(lowbw)-1)) +
+     geom_smooth(method = "glm", family = "binomial",se = F,size = 2) +
+     geom_point(shape="|")  ## adds actual values
```

ggplot() and glm()



Exercise

Take the code producing the graph

1. try to change the axis titles (`xlab()` and `ylab()`)
2. add a title (`ggtitle()`)
3. change the colour of the function to black, set se = T
4. change the colour of the points to red for the low birth weight and green for the one with normal birth weight
5. change the position of the legend; place it somewhere near the upper right corner inside the plotting area (`legend.position`)

The Challenger Disaster Example

In January 1986, the space shuttle Challenger exploded shortly after launch. An investigation was launched into the cause of the crash and attention focused on the rubber O-ring seals in the rocket boosters. At lower temperatures, rubber becomes more brittle and is a less effective sealant. At the time of the launch, the temperature was 31°F. Could the failure of the O-rings have been predicted? In the 23 previous shuttle missions for which data exists, some evidence of damage due to blow by and erosion was recorded on some O-rings. Each shuttle had two boosters, each with three O-rings. For each mission, we know the number of O-rings out of six showing some damage and the launch temperature.(faraway)

The Challenger Disaster Example

- the data are given in the data frame `orings` in the `faraway` package
- after loading we have a look at the first six lines

```
> library(faraway)
> data(orings)
> head(orings)
  temp damage
1   53      5
2   57      1
3   58      1
4   63      1
5   66      0
6   67      0
```

- we see that every shuttle mission has its own row (but not every O-ring)

The Challenger Disaster Example

- that is not a problem: one way of defining a binary response variable in a glm is to form a two-column matrix with the first column representing the number of "successes" y and the second column the number of "failures" n-y.

```
> m <- glm(cbind(damage,6-damage) ~ temp,
+                          family=binomial, orings)
```

- we see that every shuttle mission has its own row (but not every O-ring)

The Challenger Disaster Example

- the output looks familiar:

```
> summary(m)
Call:
glm(formula = cbind(damage, 6 - damage) ~ temp,
     family = binomial, data = orings)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.9529  -0.7345  -0.4393  -0.2079   1.9565
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 11.66299    3.29626   3.538 0.000403 ***
temp        -0.21623    0.05318  -4.066 4.78e-05 ***
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 38.898  on 22  degrees of freedom
Residual deviance: 16.912  on 21  degrees of freedom
AIC: 33.675
```

- remember, the response is a probability. Therefore our model describes the probability of a damaged O-ring depending on the temperature

Understanding the Cœfficients

- this relationship is described by

$$\Pr(\text{damage}) = \text{logit}^{-1}(11.66299 + -0.21623 \cdot \text{temp})$$

- the intercept

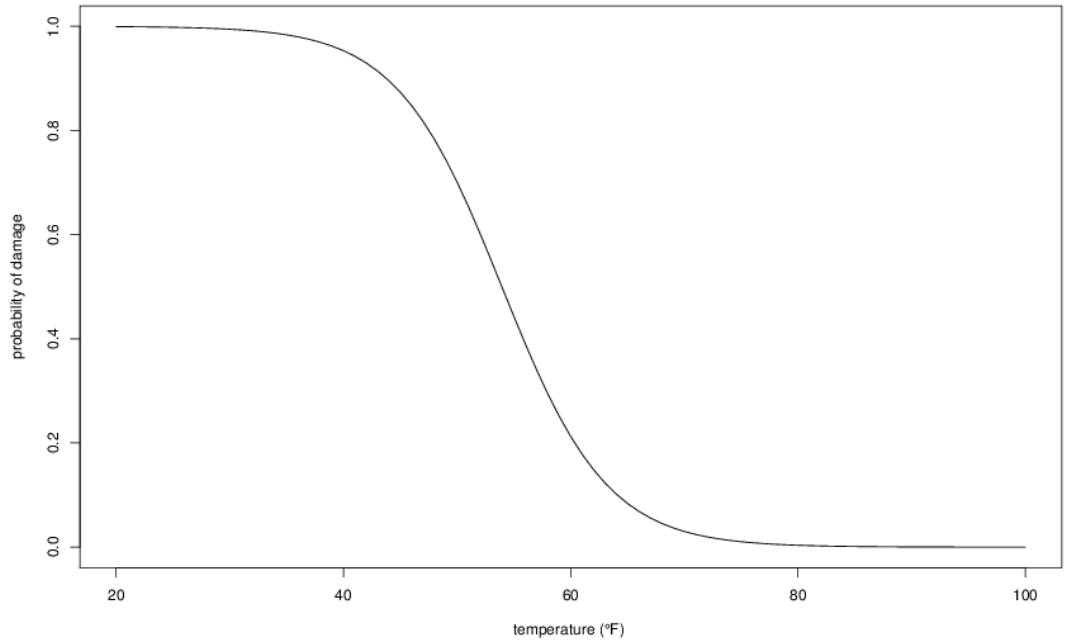
```
> invlogit(coef(m)[1])
(Intercept)
  0.9999914
```

  is interpretable as the probability for a damaged O-ring at a temperature of 0°F

- the parameter for temperature describes how fast the probability decreases with increasing temperature

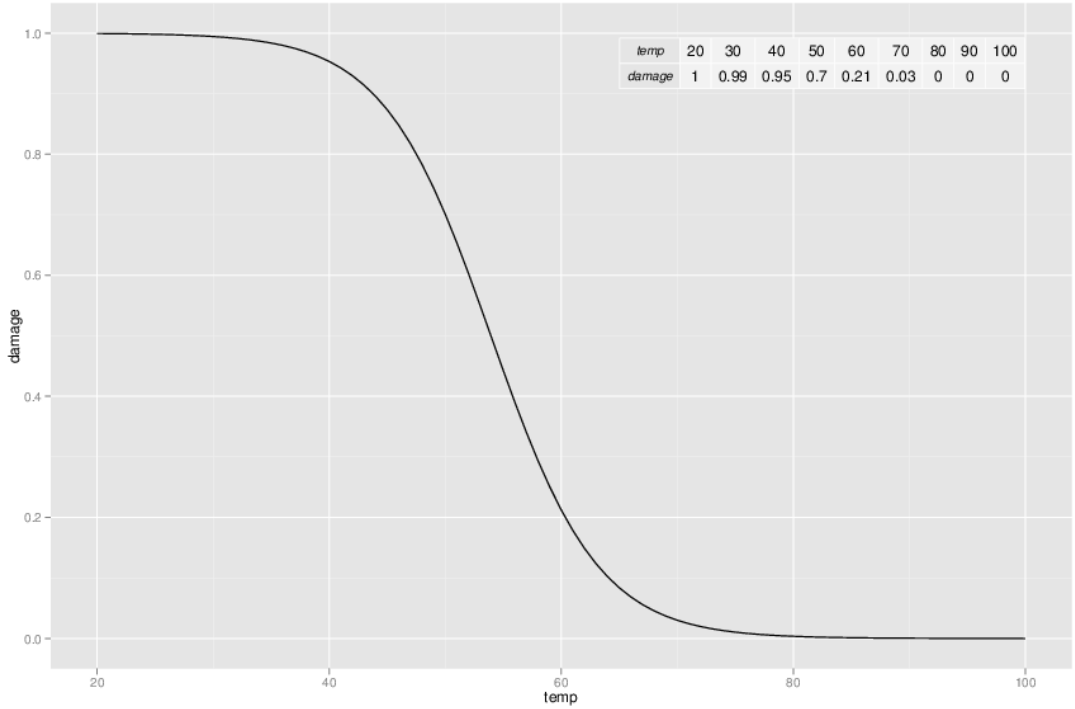
Understanding the Cœfficients

```
> tf <- 20:100
> pd <- predict(m,newdata=list(temp=tf), type="response")
> plot(tf,pd,type="l",
+  xlab=expression(paste("temperature (",degree,"F)",sep=" ")),
+  ylab="probability of damage")
```



Understanding the Cœfficients

and the same plot made with ggplot (incl. adding a table)



Parasite Infection Example

- the binary response variable is parasite infection (infected or not)
- the explanatory variables are weight and age (continuous)
- and sex (categorical)
- we want to investigate if there is a different effect of age for each of the sexes on the outcome variable

```
> infection <- read.table("infection.txt",header=T)
> summary(infection)
    infected            age              sex
 Min.   :0.000   Min.   :  2.00   Min.   :0.000
 1st Qu.:0.000   1st Qu.: 46.00   1st Qu.:0.000
 Median :0.000   Median : 84.50   Median :1.000
 Mean   :0.324   Mean   : 93.69   Mean   :0.514
 3rd Qu.:1.000   3rd Qu.:139.25   3rd Qu.:1.000
 Max.   :1.000   Max.   :200.00   Max.   :1.000
```

Parasite Infection Example

```
> m <- glm(infected~age*sex,family=binomial,
+                                  data=infection)
> summary(m)
Call:
glm(formula = infected ~ age * sex, family = binomial,
                                       data = infection)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0411  -0.7307  -0.4363   0.6632   2.3215
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.000513   0.413639  -7.254 4.05e-13 ***
age          0.015657   0.003176   4.929 8.25e-07 ***
sex          0.116664   0.553956   0.211   0.8332
age:sex      0.011050   0.004612   2.396   0.0166 *

(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 629.85  on 499  degrees of freedom
Residual deviance: 477.61  on 496  degrees of freedom
AIC: 485.61
```

Parasite Infection Example

- so for male at a age of 0 there is a probability of

```
> invlogit(coef(m)[1])
(Intercept)
 0.04740269
```

- for females is the probability at age 0

```
> invlogit(coef(m)[1]+coef(m)[3])
(Intercept)
 0.05295775
```

Parasite Infection Example

- so what about the slope?
- for males the underlying model is the following

$$\Pr(\text{infection}) = \text{logit}^{-1}(-3.000513 + 0.015657 \cdot \text{age})$$

- for females the slope is almost twice as high

$$\Pr(\text{infection}) = \text{logit}^{-1}(-2.883849 + 0.02670685 \cdot \text{age})$$

- we can compare them by looking at the age where the probability to be infected is 50%

Parasite Infection Example

- this is the case when $-3.000513 + 0.015657 \cdot \text{age} = 0$ respectively $-2.883849 + 0.02670685 \cdot \text{age} = 0$; you can do it by hand or use R

```
> ## male
> solve(0.015657,3.000513)
[1] 191.6404
> ## female
> solve(0.02670685,2.883849,)
[1] 107.9816
```

- `solve()` solves systems of linear equations in the form A*x=b, where A is the matrix of cœfficients and b are the (negative) intercepts, here we have the special case with just one equation

Parasite Infection Example

- you can also use the `allEffects()` function (part of the `effects` package), which give you the probabilities for being infected on several ages for both sexes

```
> allEffects(m)
 model: infected ~ age * sex


 age*sex effect
     sex
age             0          1
  2   0.04883687 0.05570148
  24  0.06756215 0.09596497
  46  0.09276694 0.16038932
  68  0.12610300 0.25582483
  90  0.16918450 0.38219715
  112 0.22322468 0.52680374
  134 0.28853152 0.66704908
  156 0.36399154 0.78286130
  178 0.44679328 0.86645480
  200 0.53265591 0.92110968
```

Parasite Infection Example

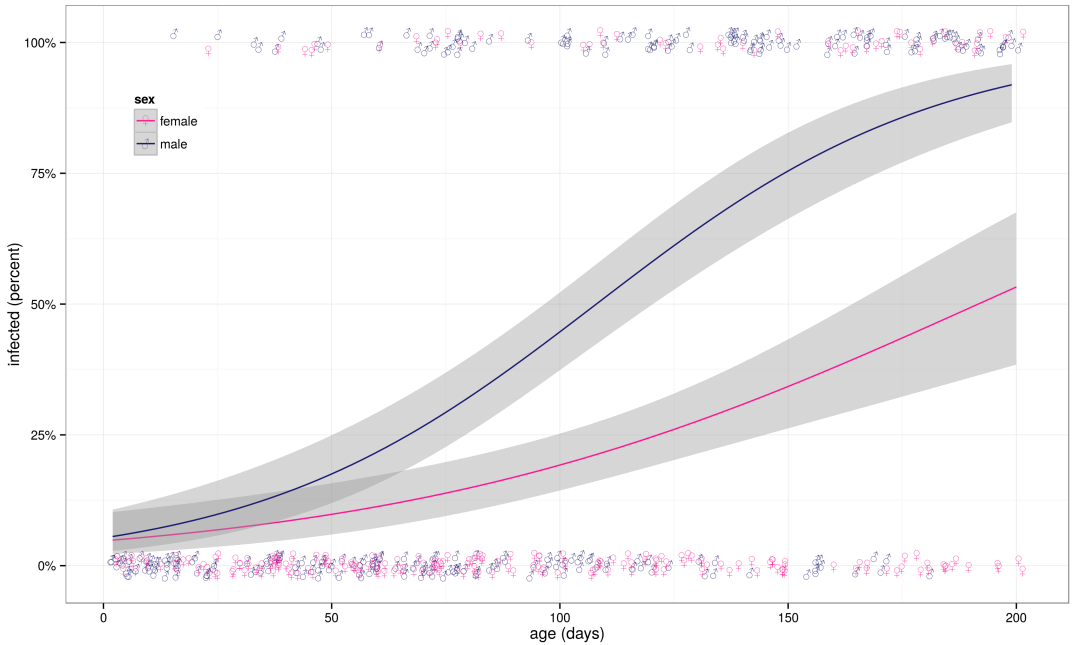


Table of Contents I

Count Data

- a great deal of the data collected is in the form of counts
- for example:
  - number of individuals that died
  - number of firms going
  - bankrupt, the number of days of frost,
  - the number of red blood cells on a microscope slide, and the
  - number of craters in a sector of lunar landscape
- with count data, the number 0 often appears as a value of the response (zero inflated data)

Count Data

- we must consider a different cases in dealing with data on frequencies: cases
  - where we count how many times something happened, but we have no way of knowing how often it did not happen (e.g. lightning strikes, bankruptcies, deaths, births).
  - count data on proportions, where we know the number doing a particular thing, but also the number not doing that thing (e.g. the proportion dying, sex ratios at birth, proportions of different groups responding to a questionnaire)

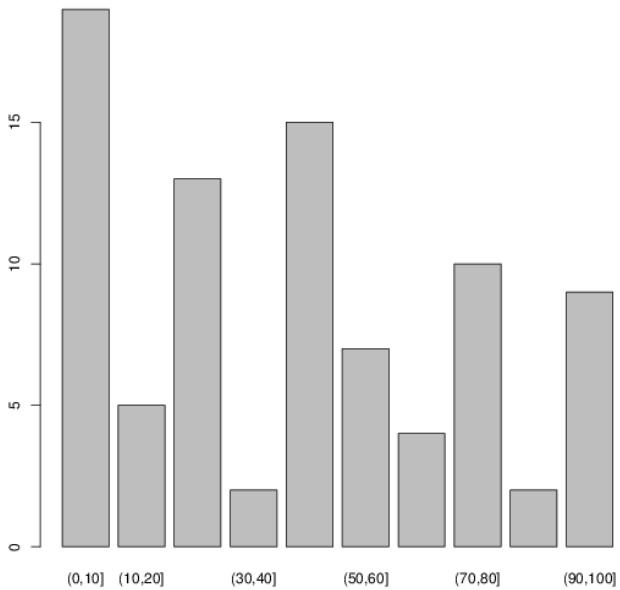
A Poisson Regression

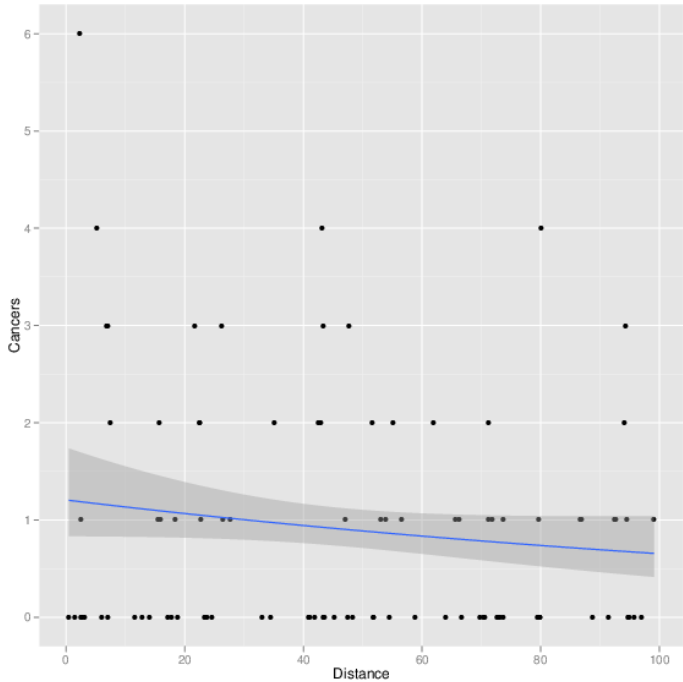
- The following example has a count (the number of reported cancer cases per year per clinic) as the response variable
- and a single continuous explanatory variable (the distance from a nuclear plant to the clinic in km).
- The question is whether or not proximity to the reactor affects the number of cancer cases.

```
> cancer <- read.table("clusters.txt",header=T)
> head(cancer)
  Cancers Distance
1       0 11.46952
2       0 66.55395
3       0 47.46230
4       0 48.38129
5       0 73.76534
6       0 70.57555
```

Count Data

- look at a barplot (cut the `Distance` variable in ten classes) and a scatter plot





Count Data

- There seems to be a downward trend in cancer cases with distance. But is the trend significant?

```
> m <- glm(Cancers~Distance,family=poisson,data=cancer)
> summary(m)
Call:
glm(formula = Cancers ~ Distance, family = poisson,
                                        data = cancer)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5504  -1.3491  -1.1553   0.3877   3.1304
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.186865   0.188728   0.990   0.3221
Distance    -0.006138   0.003667  -1.674   0.0941 .
(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 149.48  on 93  degrees of freedom
Residual deviance: 146.64  on 92  degrees of freedom
AIC: 262.41
```

Count Data

- The trend does not look to be significant, but look at the residual deviance:
- It is assumed that this is the same as the residual degrees of freedom (because the errors are supposed to be Poisson distributed)
- this indicates that we have overdispersion (extra, unexplained variation in the response).
- we compensate for the overdispersion by refitting the model using quasi-Poisson rather than Poisson errors

Count Data

- the refitted model

```
> m <- glm(Cancers~Distance,family=quasipoisson,data=cancer)
> summary(m)
Call:
glm(formula = Cancers ~ Distance,
                family = quasipoisson, data = cancer)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5504  -1.3491  -1.1553   0.3877   3.1304

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.186865   0.235364   0.794    0.429
Distance    -0.006138   0.004573  -1.342    0.183

(Dispersion parameter for quasipoisson family
                                   taken to be 1.555271)
    Null deviance: 149.48  on 93  degrees of freedom
Residual deviance: 146.64  on 92  degrees of freedom
```

Interpreting the Cœfficients

- the estimates remained the same, but the p-vals changed
- so there is no compelling evidence to support the existence of a trend in cancer incidence with distance from the nuclear plant (this is a completely made up example, neither considering varying population nor clinic density)

Interpreting the Cœfficients

- if you use glms with Poisson errors, the default link function is `log`
- so the parameter estimates and the predictions from the model (the 'linear predictor') are in logs, and need to be antilogged
- so we have the following following formula for our model

$$\text{count} = \exp\left(0.186865 - 0.006138 \cdot \text{Distance}\right)$$

- antilog the intercept:

```
> exp(coef(m)[1])
(Intercept)
   1.205464
```

- get 1.2 expected cases at a distance of zero

Interpreting the Cœfficients

- the slope for `Distance` is a bit easier to interpret than with a logit link

```
> exp(coef(m)[2])
 Distance
0.9938805
```

means that for every additional km distance you get 0.006 less cancer cases (it is nicer to say for every 10 km the expected count of cancer cases decreases by 6%)

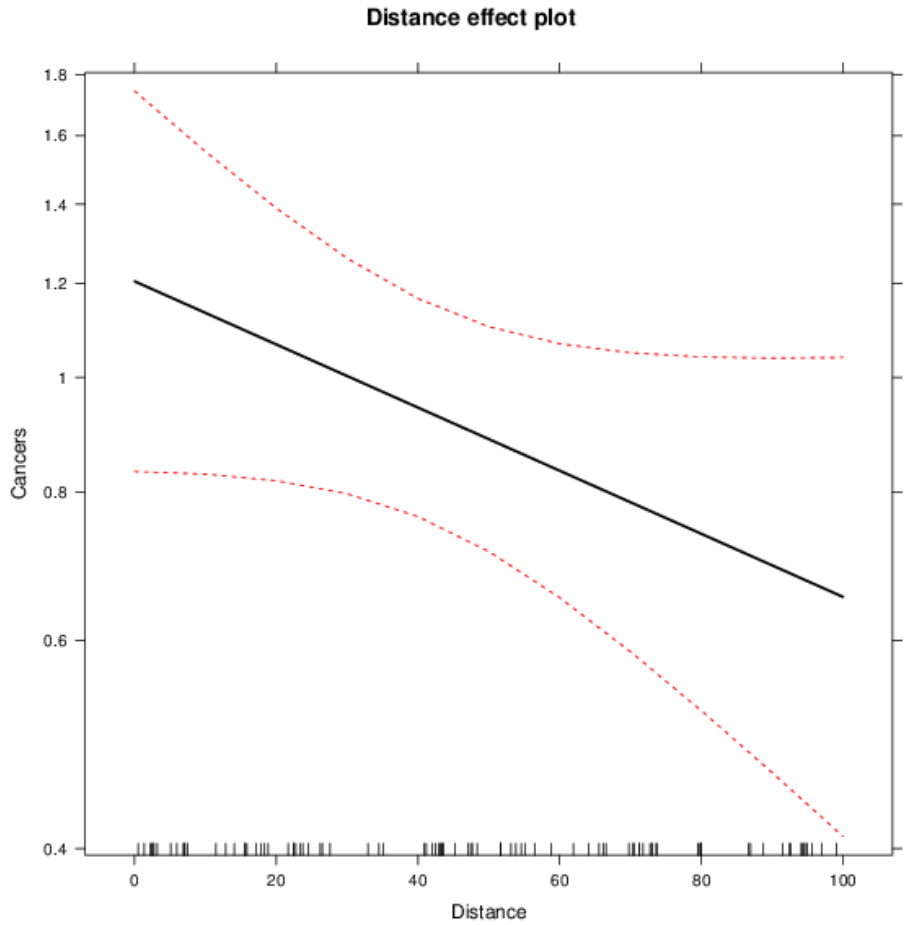
Interpreting the Cœfficients

- again, the `effects` package is very helpful to give an overview

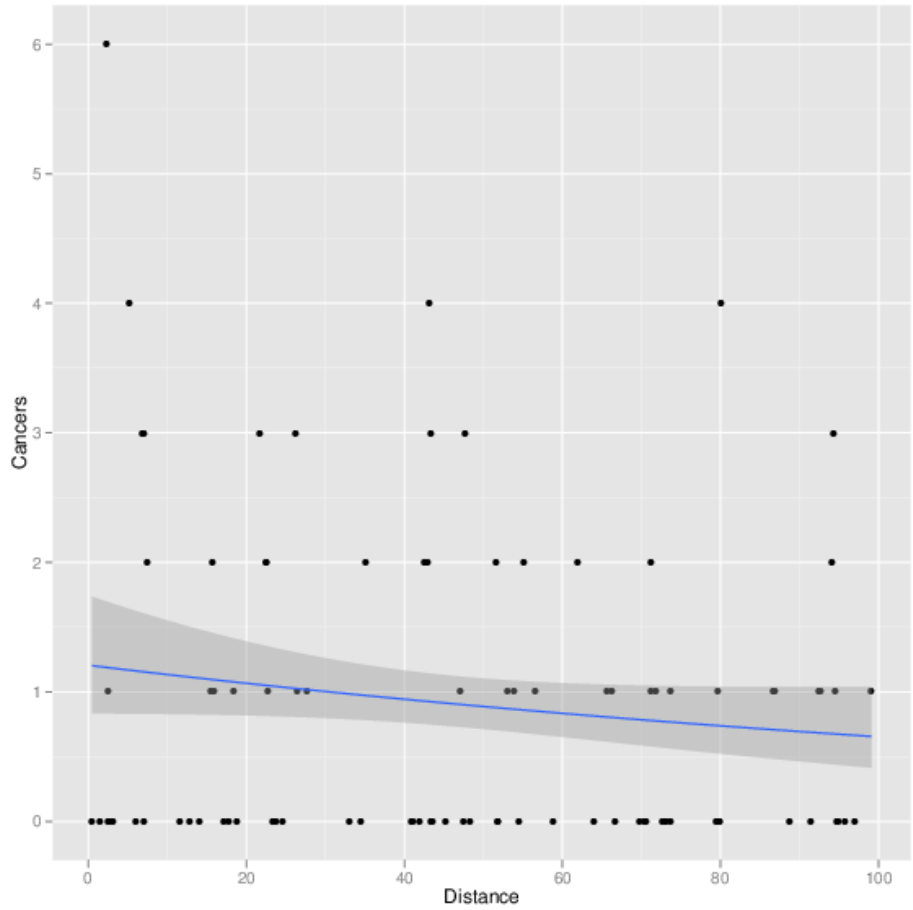
```
> allEffects(m,xlevels=list(Distance=seq(0,100,by=10))
+ )
 model: Cancers ~ Distance

 Distance effect
Distance
        0        10        20        30
1.2054642 1.1336940 1.0661968 1.0027182
       40        50        60        70
0.9430189 0.8868740 0.8340718 0.7844133
       80        90       100
0.7377114 0.6937900 0.6524835
```

Interpreting the Cœfficients

- now the effect plot and the (non-significant) fitted line can be drawn



Interpreting the Cœfficients



Anova with Count Data

- next example the response variable is a count of infected blood cells per mm^2 on microscope slides prepared from randomly selected individuals
- explanatory variables are smoker (logical, yes or no)
- and body mass score (three levels, normal, overweight, obese)
- so we fit the following model (including the interaction term)

Anova with Count Data

```
> m <- glm(cells~smoker*weight,family=poisson,data=cells)
> summary(m)
Call:
glm(formula = cells ~ smoker * weight, family = poisson, data = c
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6511  -1.1742  -0.9148   0.5533   3.6436
Coefficients:                Estimate Std. Error z value Pr(>|z|)
(Intercept)                  -0.8712     0.1302  -6.692 2.20e-11 ***
smokerTRUE                    0.8224     0.1833   4.486 7.27e-06 ***
weightobese                   0.4993     0.1671   2.987 0.002817 **
weightover                    0.2618     0.1866   1.404 0.160465
smokerTRUE:weightobese        0.8063     0.2296   3.511 0.000446 ***
smokerTRUE:weightover         0.4935     0.2546   1.939 0.052548 .

(Dispersion parameter for poisson family taken to be 1)
    Null deviance: 1052.95  on 510  degrees of freedom
Residual deviance:  792.85  on 505  degrees of freedom
AIC: 1318.5
```

Anova with Count Data

- again we see overdispersion (residual deviance $>$ degrees of freedom)
- we compensate by refitting the model using quasi-Poisson errors

Anova with Count Data

```
> m <- glm(cells~smoker*weight,family=quasipoisson,data=cells)
> summary(m)
Call:
glm(formula = cells ~ smoker * weight, family = quasipoisson,
    data = cells)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6511  -1.1742  -0.9148   0.5533   3.6436
Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)             -0.8712     0.1760  -4.950 1.01e-06 ***
smokerTRUE               0.8224     0.2479   3.318 0.000973 ***
weightobese              0.4993     0.2260   2.209 0.027598 *
weightover               0.2618     0.2522   1.038 0.299723
smokerTRUE:weightobese   0.8063     0.3105   2.597 0.009675 **
smokerTRUE:weightover    0.4935     0.3442   1.434 0.152226

(Dispersion parameter for quasipoisson family taken to be 1.82792
```

Interpreting the Cœfficients

- remember poisson has log as link so

```
> exp(coef(m)[1])
(Intercept)
  0.4184397
```

  is the expected count of infected blood cells for a normal weighted non-smoker

- all the other estimates are interpretable as factors (because of the log link!)
- so a smoker has

```
> exp(coef(m)[2])
smokerTRUE
  2.276029
```

  more than twice as many infected cells which is

```
> exp(coef(m)[1])*exp(coef(m)[2])
(Intercept)
   0.952381
```

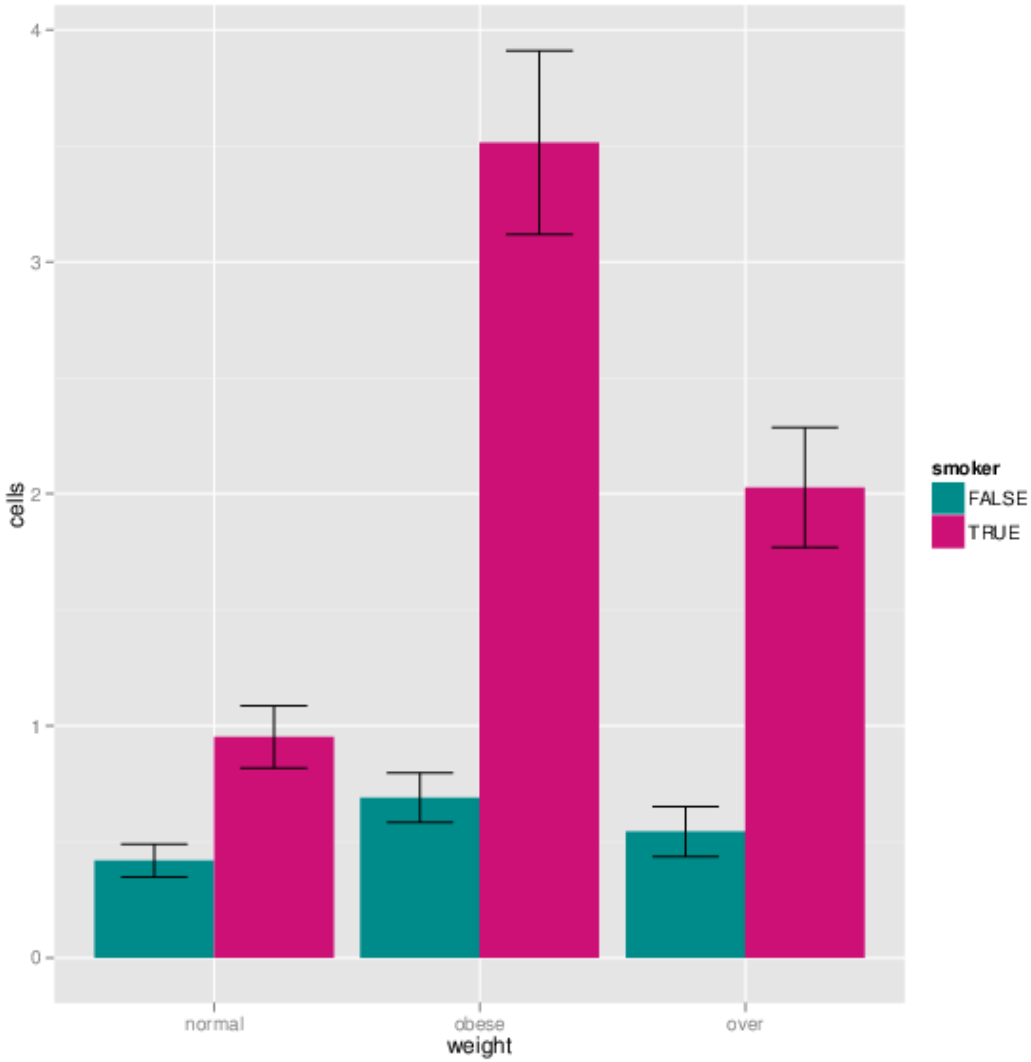
Interpreting the Cœfficients

- unfortunately `effect()` dœs not work on our model object, so we use `tapply()` (for simple models a good alternative, as soon as I remove an interaction term, or nested effects this dœs not work anymore)

```
> with(cells,tapply(cells,list(smoker,weight),mean))
         normal     obese      over
FALSE 0.4184397 0.6893939 0.5436893
TRUE  0.9523810 3.5142857 2.0270270
```

Interpreting the Cœfficients

- for visualization we use barplot with errorbars indicating the standard error



Ancova with Count Data

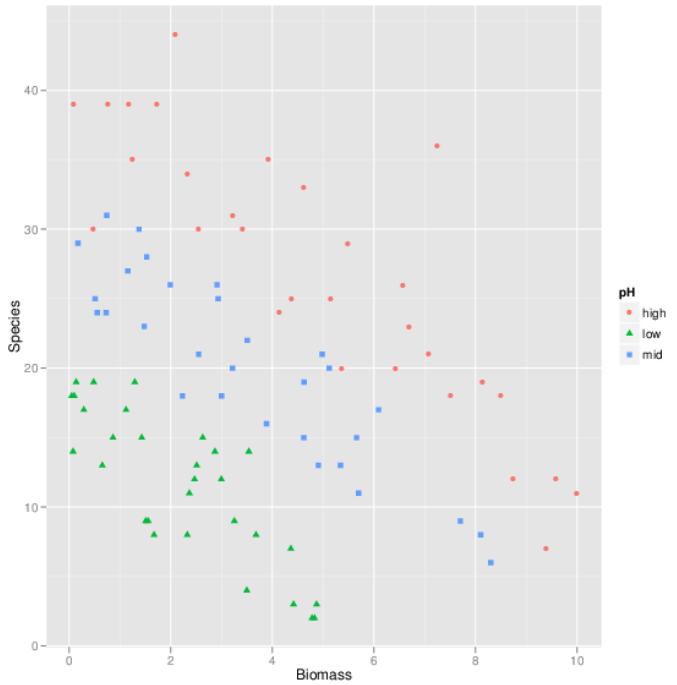
- last example: analysis of covariance
- response is a count of the number of plant species on plots
- that have different biomass (a continuous explanatory variable) and
- different soil pH (a categorical variable with three levels: high, mid and low)

```
> species<-read.table("species.txt",header=T)
> head(species)
    pH   Biomass Species
1 high 0.4692972      30
2 high 1.7308704      39
3 high 2.0897785      44
4 high 3.9257871      35
5 high 4.3667927      25
6 high 5.4819747      29
```

Ancova with Count Data

- this time we begin with a scatter plot

```
p <- ggplot(species,aes(x=Biomass,y=Species,
+                                  shape=pH,colour=pH)) +
     geom_point()
```



Ancova with Count Data

- we see: number of species declines with Biomass
- soil pH has a big effect on Species
- Dœs the slope of the relationship between Species and Biomass depend on pH?

Ancova with Count Data

- define the model and look at the summary

```
> m <- glm(Species~Biomass*pH,family=poisson,data=species)
> summary(m)
...
Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)      3.76812    0.06153  61.240  < 2e-16 ***
Biomass         -0.10713    0.01249  -8.577  < 2e-16 ***
pHlow           -0.81557    0.10284  -7.931 2.18e-15 ***
pHmid           -0.33146    0.09217  -3.596 0.000323 ***
Biomass:pHlow   -0.15503    0.04003  -3.873 0.000108 ***
Biomass:pHmid   -0.03189    0.02308  -1.382 0.166954
...
```

Ancova with Count Data

- test for the need for different slopes by comparing this maximal model (with six parameters) with a simpler model with different intercepts but the same slope

```
> m2 <- glm(Species~Biomass+pH,
+                   family=poisson,data=species)
> anova(m,m2,test="Chi")
Analysis of Deviance Table

Model 1: Species ~ Biomass * pH
Model 2: Species ~ Biomass + pH
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1        84     83.201
2        86     99.242 -2   -16.04 0.0003288 ***
```

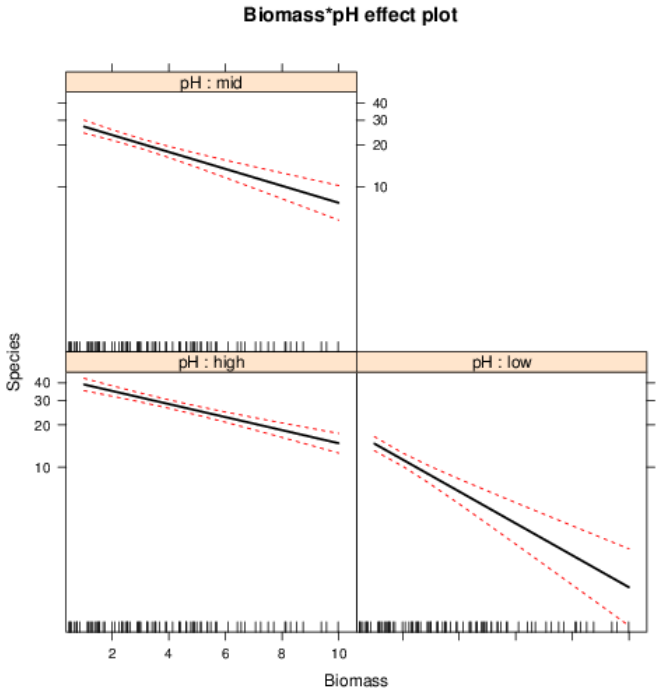
- AIC: m: 514.4; m2: 526.4

Ancova with Count Data

- slopes are very significantly different $p = 0.00033$, so it is justified to retain the more complicated model
- finally, we have a look on the effects and then draw the fitted lines through the scatterplot using the plot object p from above

```
> allEffects(m,xlevels=list(Biomass=1:10))
 model: Species ~ Biomass * pH
 Biomass*pH effect
       pH
Biomass     high       low       mid
      1  38.89998 14.737487 27.048707
      2  34.94810 11.338867 23.538030
      3  31.39769  8.724005 20.483007
      4  28.20797  6.712158 17.824498
      5  25.34229  5.164264 15.511039
      6  22.76775  3.973330 13.497847
      ....
```

Ancova with Count Data



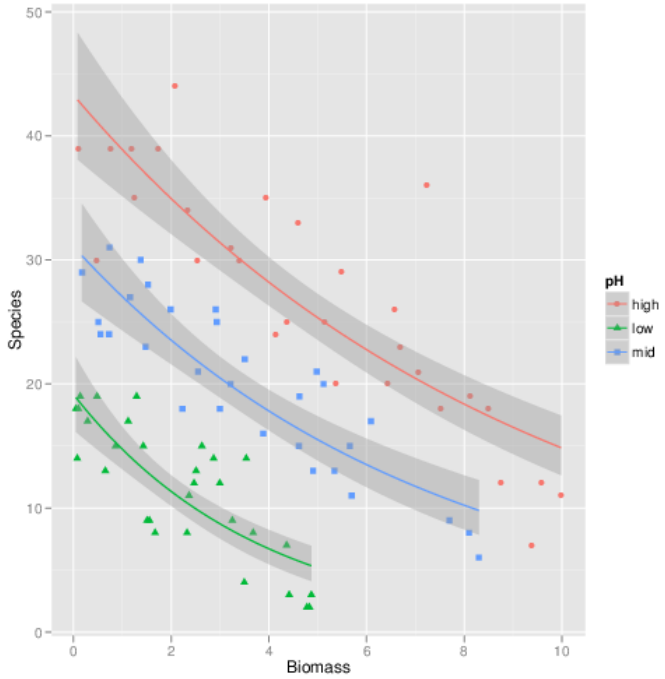


Table of Contents I

Proportion Data

- For comparisons of one binomial proportion with a constant, use `binom.test()`
- For comparison of two samples of proportion data, use `prop.test()`
- The use of GLMs on proportion data is for complex models

GLMs & Proportion Data

- uses also logit as link function and binomial error distribution
- if there is overdispersion use quasibinomial to compensate
- fitted values are counts
- we have seen one example so far: in the challenger example we have already used the responds variable in form of a proportion

GLMs & Proportion Data

- we use an example concerning sex ratios in insects as response and
- population density as explanatory variable
- so load the data and fit the model

```
> numbers <-read.table("sexratio.txt",header=T)
> head(numbers)
  density females males
1       1       1     0
2       4       3     1
3      10       7     3
4      22      18     4
5      55      22    33
6     121      41    80
> m <- glm(cbind(males,females)~density,
+                    family=binomial,data=numbers)
```

GLMs & Proportion Data

```
> summary(m)
Call:
glm(formula = cbind(males, females) ~ density, family = binomial,
    data = numbers)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.4619  -1.2760  -0.9911   0.5742   1.8795

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.0807368  0.1550376   0.521    0.603
density     0.0035101  0.0005116   6.862 6.81e-12 ***

(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 71.159  on 7  degrees of freedom
Residual deviance: 22.091  on 6  degrees of freedom
AIC: 54.618
```

GLMs & Proportion Data

- the residual deviance is larger than the residual degrees of freedom
- because it is something like a growth process we try a log transformation (before using quasibinomial family)

```
> m <- glm(cbind(males,females)~log(density),
+                            family=binomial,data=numbers)
> summary(m)
Call:
glm(formula = cbind(males, females) ~ log(density),
                   family = binomial, data = numbers)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9697  -0.3411   0.1499   0.4019   1.0372

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.65927    0.48758  -5.454 4.92e-08 ***
log(density)  0.69410    0.09056   7.665 1.80e-14 ***

    Null deviance: 71.1593  on 7  degrees of freedom
Residual deviance:  5.6739  on 6  degrees of freedom
AIC: 38.201
```

GLMs & Proportion Data

- the transformation caused a welcome decrease in the residual deviance
- we conclude that the proportion of animals that are males increases significantly with increasing density, and
- that the logistic model is linearized by logarithmic transformation of the explanatory variable

GLMs & Proportion Data

```
ggplot(numbers, aes(x=log(density),y=males/(males+female
  geom_point() +
  geom_smooth(method=glm,family=binomial)
```

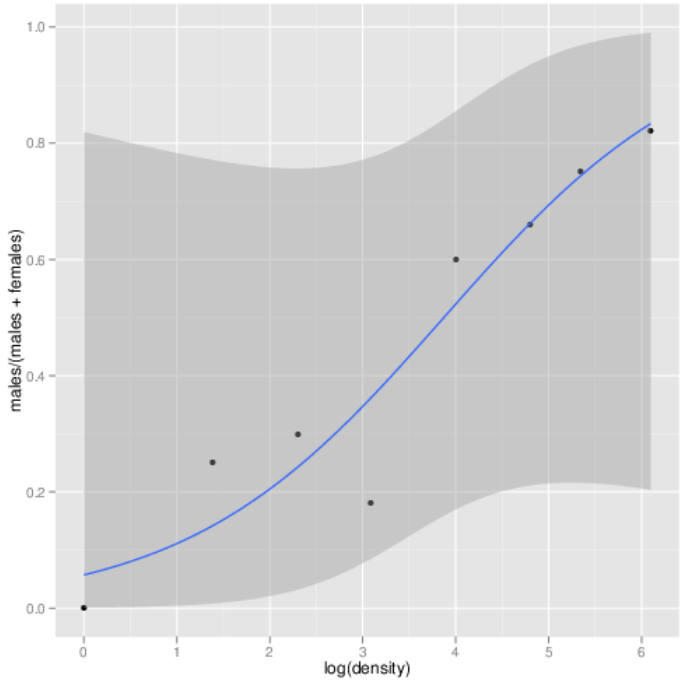


Table of Contents I

R syntax - glm() vs lm()

- the glm() function needs
  - a model formula (like lm)
  - the specification of error distribution (family=)

Input/Output

```
> m1 <- lm(bweight ~ hyp, data=births)
> m2 <- glm(bweight ~ hyp, family=gaussian, data=births)
```

`glm()` for logistic regression

- every error family has a canonical link
- we have seen binomial error family with its canonical logit link
- common choices for link functions used with binomial errors
  - logit: $\eta = \log(p/(1-p))$
  - probit: $\eta = \Phi^{-1}(p)$
  - log-log: $\log(-\log(1-p))$

Odds

- logistic regression is more understandable if you look at the cœfficients in terms of odds where
- $\Omega(A) = \frac{P(A)}{1 - P(A)}$
- so what are the corresponding odds for a probability of
  - $p = 1$
  - $p = 0.99$
  - $p = 0.5$
  - $p = 0.1$
  - $p = 0.01$
  - $p = 0$

Odds

$$p = 1 \qquad \omega = \infty$$

$$p = 0.99 \qquad \omega = 99$$

$$p = 0.5 \qquad \omega = 1$$

$$p = 0.1 \qquad \omega = 0.\overline{1}$$

$$p = 0.01 \qquad \omega = 0.\overline{01}$$

$$p = 0 \qquad \omega = 0$$

Remember the Data

Input/Output

```
> str(births)
'data.frame': 500 obs. of  8 variables:
$ id      : num  100 101 102 103 104 105 106 107 108 109 ...
$ preterm: Factor w/ 2 levels "preterm","normal": 2 2 2 2 2 2
$ gestwks: num  39.8 39 38.1 39.5 39.5 ...
$ hyp     : Factor w/ 2 levels "normal","hyper": 1 1 1 1 2 1 2
$ matage : num  33 32 33 38 40 29 32 40 41 39 ...
$ bweight: num  3576 3784 2796 3226 3138 ...
$ lowbw   : Factor w/ 2 levels "normal","low": 1 1 1 1 1 1 1 1
$ sex     : Factor w/ 2 levels "M","F": 2 2 2 2 2 2 1 1 2 2 ..
```

Data from: Michæl Hills and Bianca De Stavola (2002). A Short Introduction to Stata 8 for Biostatistics, Timberlake Consultants Ltd URL: http://www.timberlake.co.uk

Exercises

Remember: We used hypertension of the mom to explain variation in the birth weight (in terms of low birth weight or not of course) of the kid. Without looking in the material of the last session, try to redo the model. Here are some hints:

- of course you need the `glm()` function
- you need to specify the formula which has to have the general form $y \sim x$
- additional you need to specify the data and the error family (in the case `binomial`)
- use the `summary()` function on the model
- use `Effect()` or `allEffects()` commands on the model
- how to interpret the results? Is the effect of hypertension statistically significant?

Exercise - Solution

Input/Output

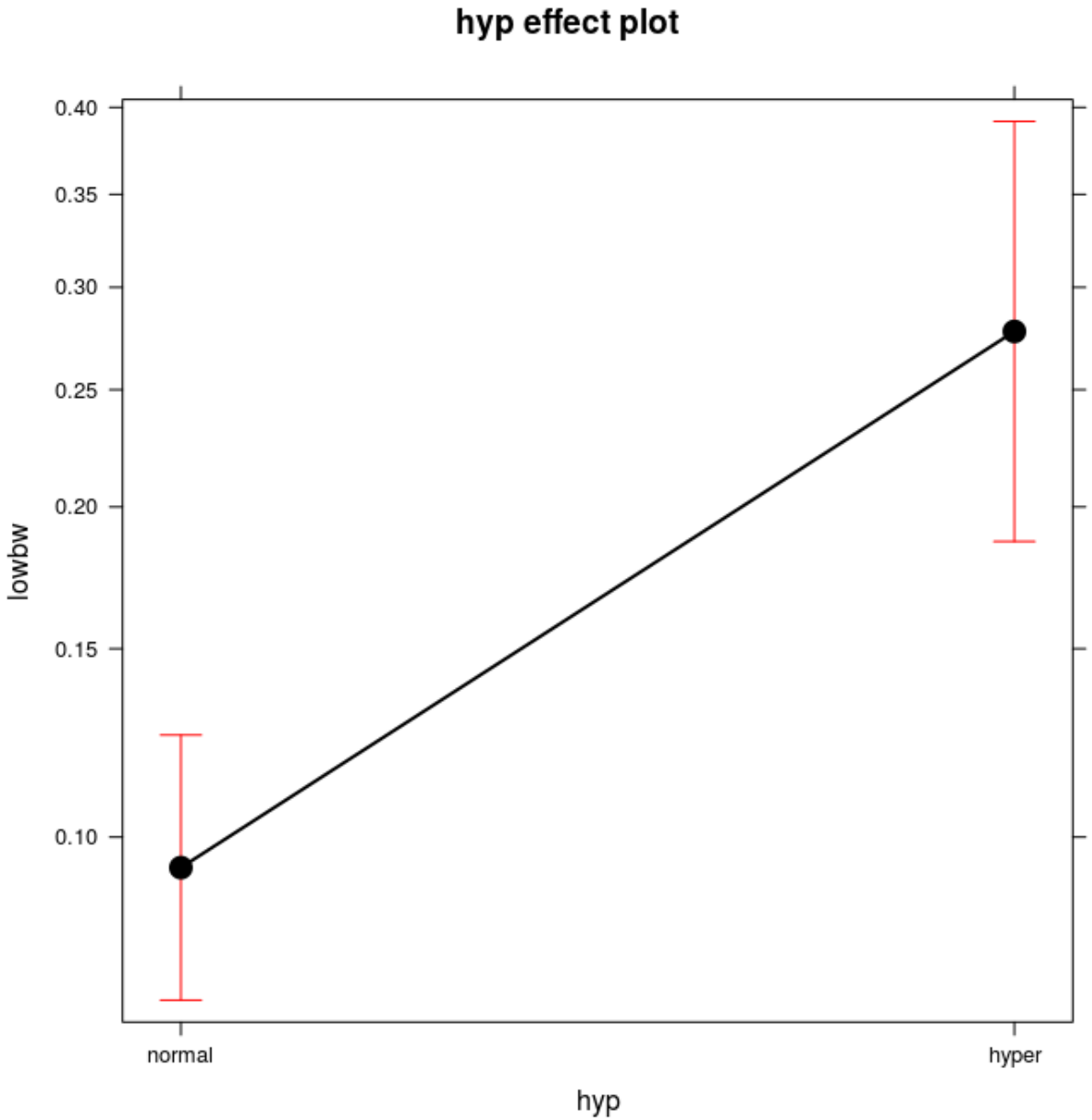
```
> m <- glm(lowbw ~ hyp, family=binomial, data=births)
> summary(m)
Call:
glm(formula = lowbw ~ hyp, family = binomial, data = births)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.8067  -0.4430  -0.4430  -0.4430   2.1773

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.2721     0.1661 -13.682  < 2e-16 ***
hyphyper      1.3166     0.3111   4.232 2.32e-05 ***
---

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 366.92  on 499  degrees of freedom
Residual deviance: 350.84  on 498  degrees of freedom
AIC: 354.84
```

Exercise - Solution

Input/Output

```
> require(effects)
Lade nötiges Paket: effects
> allEffects(m.hyp)
 model: lowbw ~ hyp

 hyp effect
hyp
    normal      hyper
0.09345794 0.27777778
> plot(allEffects(m.hyp))
```

Exercise - Solution



Exercise - Solution

Input/Output

```
> res <- allEffects(m.hyp, se = T)
> summary(res)
 model: lowbw ~ hyp

 hyp effect
hyp
    normal      hyper
0.09345794 0.27777778

 Lower 95 Percent Confidence Limits
hyp
    normal      hyper
0.06929267 0.18675845

 Upper 95 Percent Confidence Limits
hyp
   normal     hyper
0.1249195 0.3917861
```

Exercises

What is the relationship between the cœfficients of the model (from the model summary) and the effects?

Exercises - Solutions

What is the relationship between the cœfficients of the model (from the model summary) and the effects?

- we have to use the inverse link function on the cœfficients to transform the cœfficients on the logit scale to more interpretable probabilities

Input/Output

```
> invlogit(coef(m.hyp)[1])
(Intercept)
 0.09345794
> invlogit(coef(m.hyp)[1] + coef(m.hyp)[2])
(Intercept)
  0.2777778
```

Table of Contents I

Simple Logistic Regression

- now we model the probability of low birth weight dependent on gestational age (numeric variable)
- so the model in R is

Input

```
> m.wks <- glm(lowbw ~ gestwks, family=binomial, data=births)
```

- and as math formula

$$\log\left(\frac{\text{Pr(lowbw)}}{1-\text{Pr(lowbw)}}\right) = \beta_0 + \beta_1 \cdot \text{gestwks} + \epsilon$$

Simple Logistic Regression

- where the output look similar to the output above

Input/Output

```
> summary(m.wks)

Call:
glm(formula = lowbw ~ gestwks, family = binomial, data = births)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0873  -0.3623  -0.2223  -0.1369   2.9753

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  31.8477     4.0574   7.849 4.18e-15 ***
gestwks      -0.8965     0.1084  -8.272  < 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: 360.38  on 489  degrees of freedom
Residual deviance: 205.75  on 488  degrees of freedom
  (10 observations deleted due to missingness)
```

Understanding the Cœfficients

- this relationship is described by

$$\Pr(\text{lowbw}) = \text{logit}^{-1}(31.8477 + -0.8965 \cdot \text{gestwks})$$

- the intercept

Input/Output

```
> invlogit(coef(m.wks)[1])
(Intercept)
  1
```

  is interpretable as the probability for a low birth weight at a hypothetical gestational age of 0 (which makes no sense because it lies outside the range of gestational ages in our data and is nonsense anyway)

- the parameter for `gestwks` describes how fast the probability decreases with increasing gestational age

Understanding the Cœfficients

$$\Pr(\text{lowbw}) = \text{logit}^{-1}(31.8477 + -0.8965 \cdot \text{gestwks})$$

- the cœfficient for `gestwks` is best interpretable if we use it as argument to the exponential function

Input/Output

```
> exp(coef(m.wks)[2])
  gestwks
0.4080114
```

this way it is interpretable as odds ratio for low birth weight for a difference of 1 week of gestational age (because we are measuring gestational in weeks as unit)

Exercise

1. here is a example for the `Effects()` command for regression

Input/Output

```
> Effect("gestwks",m.wks)

 gestwks effect
gestwks
       25         30         35         40
0.99992022 0.99299324 0.61574996 0.01779725
> Effect("gestwks",m.wks,xlevels = list(gestwks = c(20,30,40)))

 gestwks effect
gestwks
       20         30         40
0.99999910 0.99299324 0.01779725
```

2. use the command to gain the estimated probability of low birth weight for a gestational age of 27 and 36 weeks

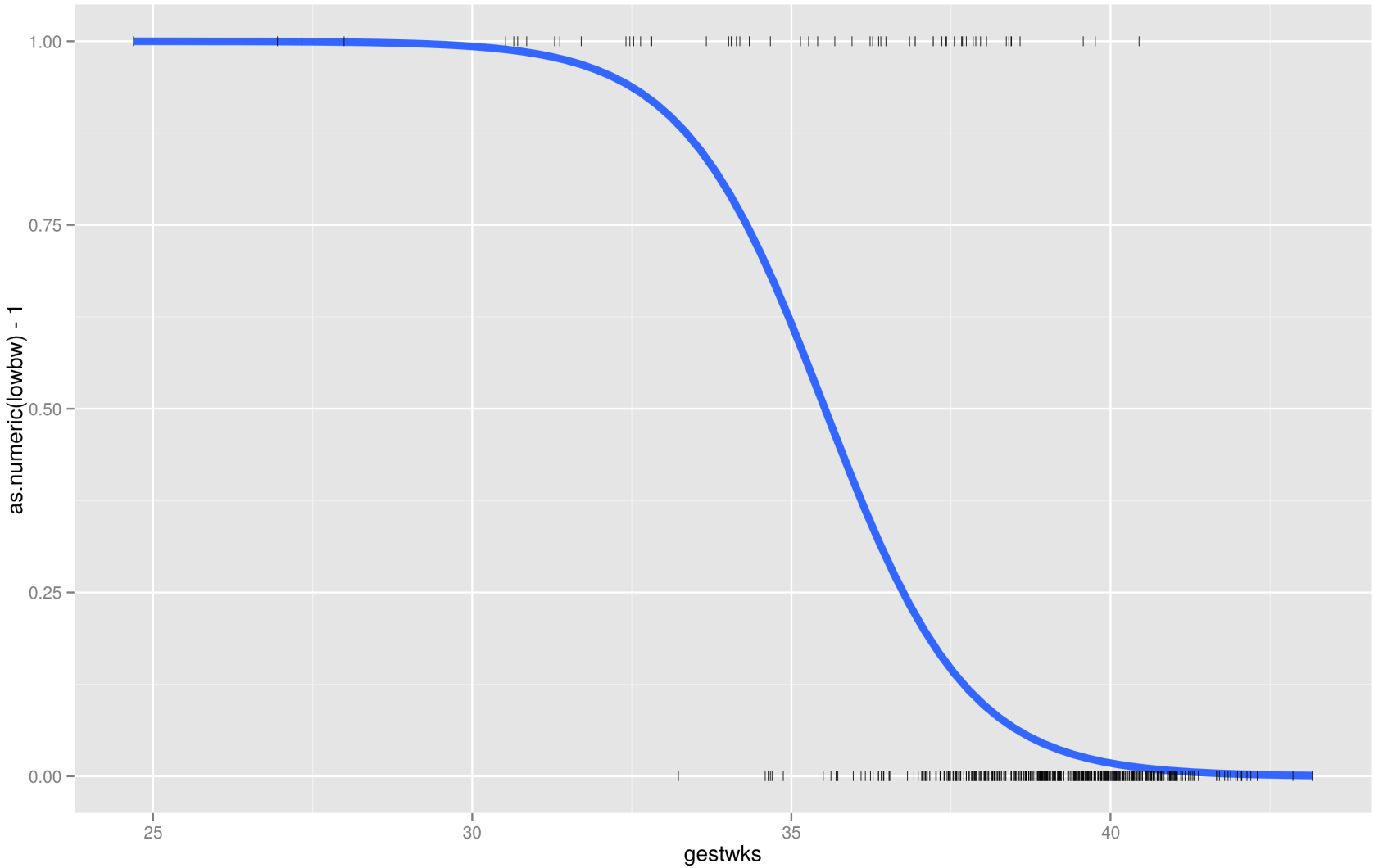
ggplot() and glm()

- ggplot2 knows also glms
- unfortunately the y-variable needs to be coded in 0s and 1s, but we can do this on the fly with `as.numeric()`

Input

```
> require(ggplot2)
> ggplot(births,aes(x = gestwks, y = as.numeric(lowbw)-1)) +
+     geom_smooth(method = "glm", family = "binomial",se = F,size = 2) +
+     geom_point(shape="|")  ## adds actual values
```

ggplot() and glm()



Exercise

Take the code producing the graph

1. try to change the axis titles (`xlab()` and `ylab()`)
2. add a title (`ggtitle()`)
3. change the colour of the function to black, set se = T
4. change the colour of the points to red for the low birth weight and green for the one with normal birth weight
5. change the position of the legend; place it somewhere near the upper right corner inside the plotting area (`legend.position`)

Table of Contents I

The Challenger Disaster Example

In January 1986, the space shuttle Challenger exploded shortly after launch. An investigation was launched into the cause of the crash and attention focused on the rubber O-ring seals in the rocket boosters. At lower temperatures, rubber becomes more brittle and is a less effective sealant. At the time of the launch, the temperature was 31°F. Could the failure of the O-rings have been predicted? In the 23 previous shuttle missions for which data exists, some evidence of damage due to blow by and erosion was recorded on some O-rings. Each shuttle had two boosters, each with three O-rings. For each mission, we know the number of O-rings out of six showing some damage and the launch temperature.(faraway)
`http://www.history.com/topics/challenger-disaster/videos/engineering-disasters---challenger`

The Challenger Disaster Example

- the data are given in the data frame `orings` in the `faraway` package
- after loading we have a look at the first six lines

Input/Output

```
> library(faraway)
> data(orings)
> head(orings)
  temp damage
1   53      5
2   57      1
3   58      1
4   63      1
5   66      0
6   67      0
```

- we see that every shuttle mission has its own row (but not every O-ring)

The Challenger Disaster Example

- that is not a problem: one way of defining a binary response variable in a glm is to form a two-column matrix with the first column representing the number of "successes" y and the second column the number of "failures" n-y.

Input/Output

```
> m.oring <- glm(cbind(damage,6-damage) ~ temp,
+                          family=binomial, orings)
```

The Challenger Disaster Example

- the output looks familiar:

Input/Output

```
> summary(m.oring)
Call:
glm(formula = cbind(damage, 6 - damage) ~ temp,
     family = binomial, data = orings)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.9529  -0.7345  -0.4393  -0.2079   1.9565
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 11.66299    3.29626   3.538 0.000403 ***
temp        -0.21623    0.05318  -4.066 4.78e-05 ***
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 38.898  on 22  degrees of freedom
Residual deviance: 16.912  on 21  degrees of freedom
AIC: 33.675
```

- remember, the response is a probability. Therefore our model describes the probability of a damaged O-ring depending on the temperature

Understanding the Cœfficients

- this relationship is described by

$$\Pr(\text{damage}) = \text{logit}^{-1}(11.66299 + -0.21623 \cdot \text{temp})$$

Input/Output

```
> invlogit(coef(m.oring)[1])
(Intercept)
  0.9999914
```

- the intercept is interpretable as the probability for a damaged O-ring at a temperature of 0°F

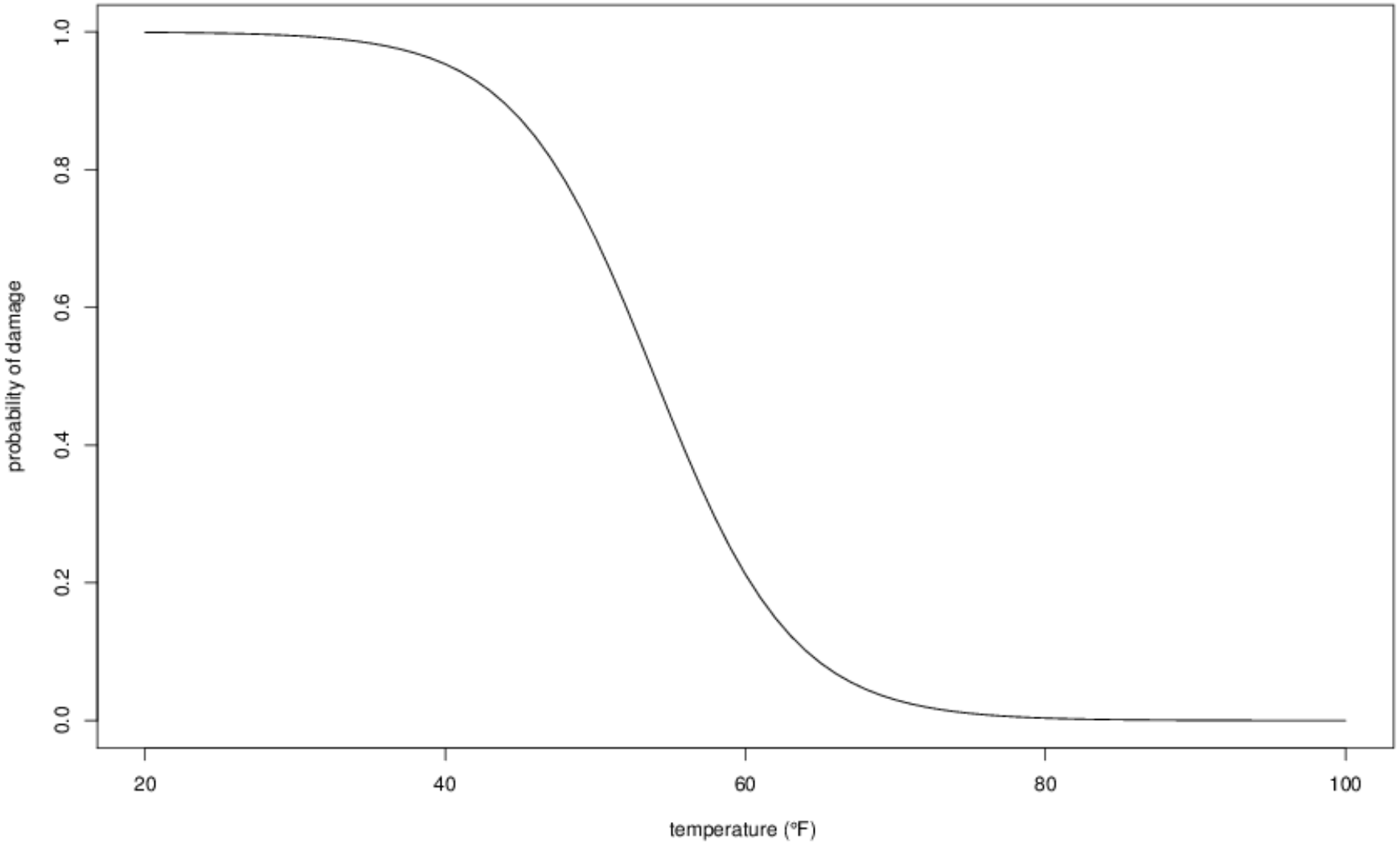
Understanding the Cœfficients

- the parameter for temperature describes how fast the probability decreases with increasing temperature and it is again best interpretable as odds ratio

Input/Output

```
> exp(coef(m.oring)[2])
     temp
0.8055471
```

Understanding the Cœfficients



Understanding the Cœfficients

and the same plot made with ggplot (incl. adding a table)

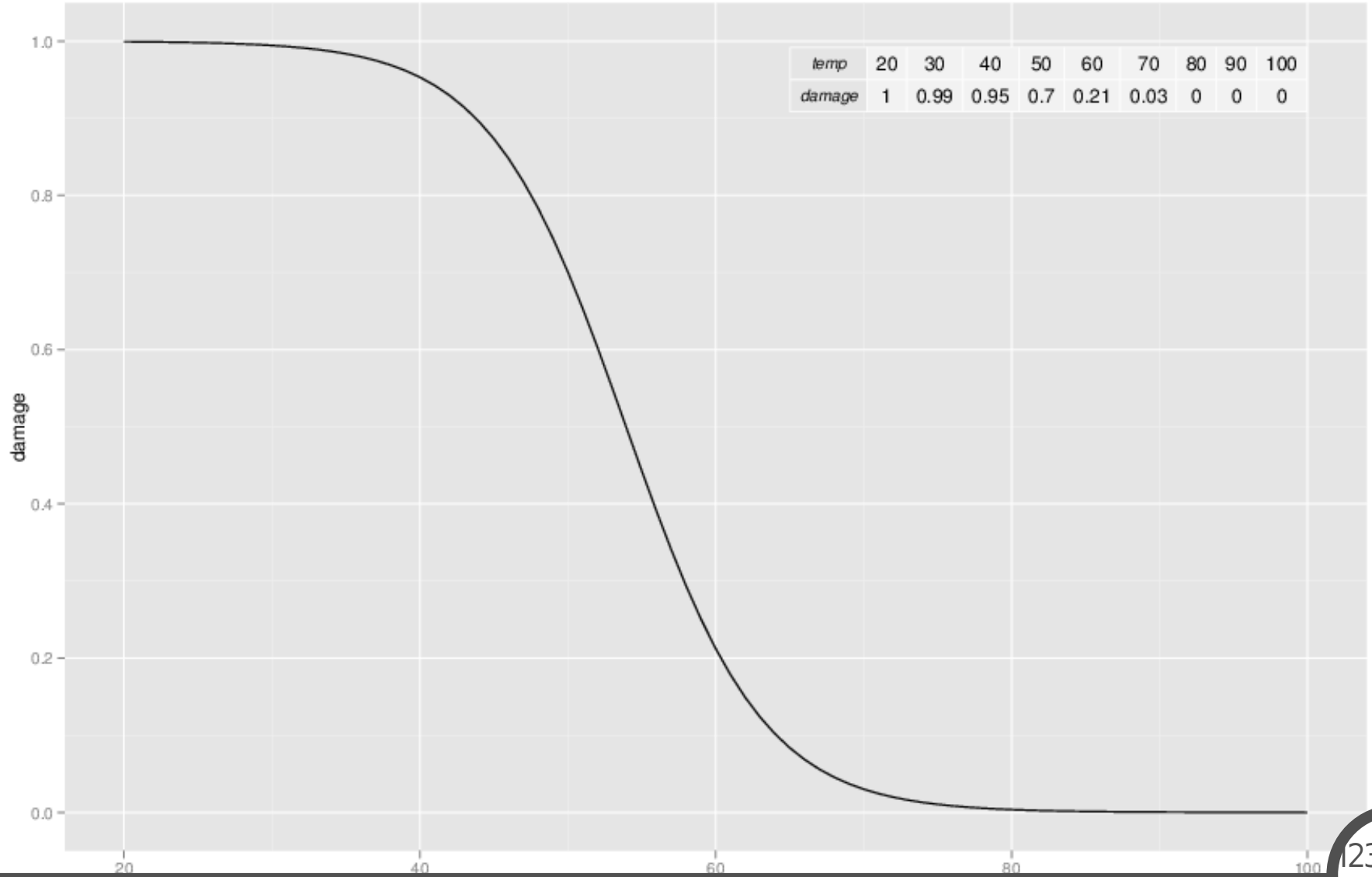


Table of Contents I

Parasite Infection Example

- the binary response variable is parasite infection (infected or not)
- the explanatory variables are weight and age (continuous)
- and sex (categorical)
- we want to investigate if there is a different effect of age for each of the sexes on the outcome variable

Input/Output

```
> load("infection.rdata")
> summary(infection)
         infected          age                sex
 infected    :338   Min.   :  2.00   female:243
 not infected:162   1st Qu.: 46.00   male  :257
                    Median : 84.50
                    Mean   : 93.69
                    3rd Qu.:139.25
                    Max.   :200.00
```

Parasite Infection Example

Input/Output

```
> m.inf <- glm(infected~age*sex,family=binomial,
+                                 data=infection)
> summary(m.inf)
Call:
glm(formula = infected ~ age * sex, family = binomial,
                                       data = infection)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0411  -0.7307  -0.4363   0.6632   2.3215
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.000513   0.413639  -7.254 4.05e-13 ***
age          0.015657   0.003176   4.929 8.25e-07 ***
sex          0.116664   0.553956   0.211   0.8332
age:sex      0.011050   0.004612   2.396   0.0166 *

(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 629.85  on 499  degrees of freedom
Residual deviance: 477.61  on 496  degrees of freedom
AIC: 485.61
```

Parasite Infection Example

- so for male at a age of 0 there is a probability of

Input/Output

```
> invlogit(coef(m.inf)[1])
(Intercept)
 0.04740269
```

- for females the probability at age 0 is

Input/Output

```
> invlogit(coef(m.inf)[1]+coef(m.inf)[3])
(Intercept)
 0.05295775
```

Compare Slopes

- so what about the slope?
- for males the underlying model is the following

$$\text{Pr(infection)} = \text{logit}^{-1}(-3.000513 + 0.015657 \cdot \text{age})$$

- for females the slope is almost twice as high

$$\text{Pr(infection)} = \text{logit}^{-1}(-2.883849 + 0.02670685 \cdot \text{age})$$

Compare Slopes

- looking at the odds ratios (which seem to be rather small)
- for males and females:

Input/Output

```
> exp(coef(m.inf)[2]) ## males
    age
1.01578
> exp(coef(m.inf)[2] + coef(m.inf)[4]) ## females
     age
1.027067
```

- these are the odds ratios for +1 time unit

Compare Slopes

- if time unit is days you get the odds ratio for +1 month by

Input/Output

```
> exp(30 * coef(m.inf)[2])
     age
1.599512
> exp(30 * (coef(m.inf)[2] + coef(m.inf)[4]))
     age
2.228225
```

- so keep in mind the scale you are measuring on

Compare Slopes

- we can also compare them by looking at the age where the probability to be infected is 50%
- this is the case when

$$-3.000513 + 0.015657 \cdot \text{age} = 0$$

respectively

$$-2.883849 + 0.02670685 \cdot \text{age} = 0$$

you can do it by hand or use R

Compare Slopes

- `solve()` solves systems of linear equations in the form A*x=b, where A is the matrix of cœfficients and b are the (negative) intercepts, here we have the special case with just one equation

Input/Output

```
> ## male
> solve(0.015657,3.000513)
[1] 191.6404
> ## female
> solve(0.02670685,2.883849)
[1] 107.9816
```

Compare Effects

- you can also use the `allEffects()` function (part of the `effects` package), which give you the probabilities for being infected on several ages for both sexes

Input/Output

```
> allEffects(m.inf)
 model: infected ~ age * sex

 age*sex effect
     sex
age             0          1
  2   0.04883687 0.05570148
  24  0.06756215 0.09596497
  46  0.09276694 0.16038932
  68  0.12610300 0.25582483
  90  0.16918450 0.38219715
  112 0.22322468 0.52680374
  134 0.28853152 0.66704908
  156 0.36399154 0.78286130
  178 0.44679328 0.86645480
  200 0.53265591 0.92110968
```

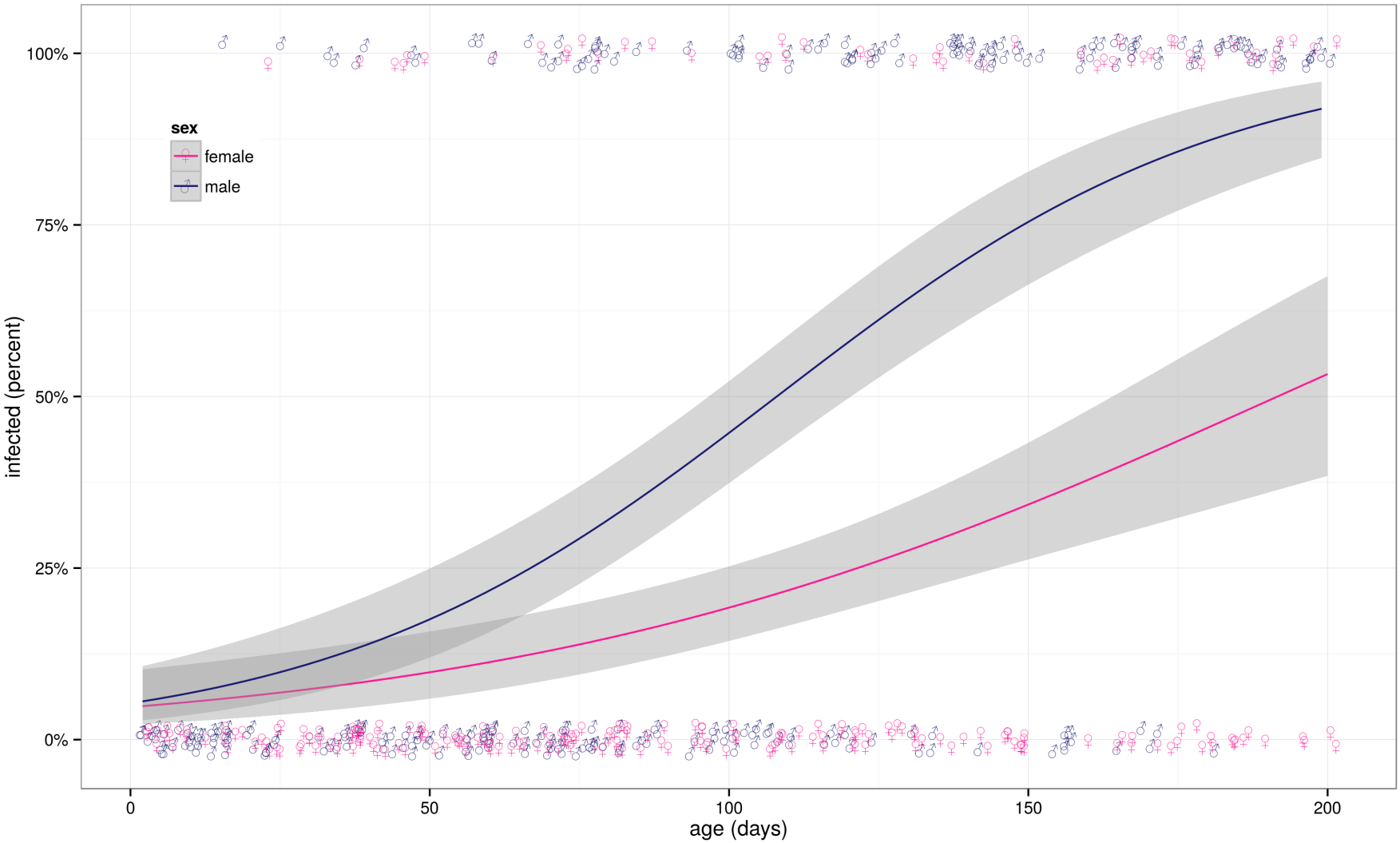
Compare Effects

- choose values of age

Input/Output

```
> allEffects(m.inf,
+            xlevels = list(age = seq(0,200,by = 50)))
 model: infected ~ age * sex

 age*sex effect
     sex
age         female       male
  0    0.04740269 0.05295775
  50   0.09817379 0.17530204
  100  0.19234385 0.44690980
  150  0.34253427 0.75439251
  200  0.53265591 0.92110968
```

Parasite Infection graph



Exercise

Try to reproduce the plot! Hints:

1. set up a ggplot object, think about the æsthetics (`aes()`). Which quality of the graph you wanna set to which variable?
2. begin with the lines (`geom_smooth()`)
3. add the points (`geom_jitter()`; do not think about the symbols in the first place; try to adjust the width and height appropriately)
4. change the colour of the lines and points (`scale_colour_manual()`); I used midnightblue for male and deeppink for female
5. change the symbols (`scale_shape_manual()`); use

```
values = c("male" = "\u2642","female" = "\u2640")
```

   as values
6. set the axes titles
7. change to text of the y axis to percentage
8. etc