Statistics: Changes since I was an undergrad

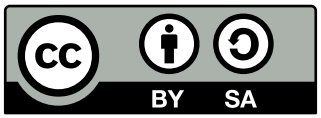
Part 3: Examples of modern methods using R

Pedro J. Aphalo

Department of Biosciences, University of Helsinki

19 November 2014

Department of Biosciences, University of Helsinki, Finland.
http://blogs.helsinki.fi/aphalo/



Lectures in this series

Changes in statistics: 1975–2014

This lecture is the third in a series of four related lectures.

1. Computing capacity has dramatically increased and programming languages have evolved.
2. New statistical methods have been developed.
3. Statistical software is dramatically more powerful and varied.
4. Accountability and reproducibility have become requirements in scientific research.

Outline

The realm of statistics

- Design of experiments.
- Data acquisition protocols.
- Data quality control and exploration.
- Data analysis proper.
  1. Tests of significance.
  2. Model fitting.
- Interpretation of results.

The realm of statistics

- Design of experiments.
- Data acquisition protocols.
- Data quality control and exploration.
- Data analysis proper.
  1. Tests of significance.
  2. Model fitting.
- Interpretation of results.

The realm of statistics

- Design of experiments.
- Data acquisition protocols.
- Data quality control and exploration.
- Data analysis proper.
  1. Tests of significance.
  2. Model fitting.
- Interpretation of results.

The realm of statistics

- Design of experiments.
- Data acquisition protocols.
- Data quality control and exploration.
- Data analysis proper.
  1. Tests of significance.
  2. Model fitting.
- Interpretation of results.

The realm of statistics

- Design of experiments.
- Data acquisition protocols.
- Data quality control and exploration.
- Data analysis proper.
  1. Tests of significance.
  2. Model fitting.
- Interpretation of results.

Today's roadmap

- A very brief introduction to R.
- Practical examples, using R, of the methods presented in Lecture 2.

R

- R is both a programming and scripting language and a computer program.
- It is sometimes called Gnu-S, as it started as a free implementation of the S language, used in S-Plus.
- Why the name 'R'?
  - R comes before S.
  - Both developers' first name is Robert.
  - Robert Ihaka and Robert Gentleman, created R as a teaching tool, in New Zealand.

R advantages

- R is extensible.
- The R language is syntactically complete.
- Routines written in other languages such as C, C++, and FORTRAN can be called from R.
- It is portable and stable.
- It is free.
- User and developer community support is great.

Why scripting?

- It is easy to describe what one has done
- It is easy to explain concisely what needs to be done
- It is easy to exactly repeat the same analysis
  - at a later time
  - on a different data set
  - by a different person

Saying 'Hello' and a bit more in R

```
print("Hello")
```

```
## [1] "Hello"
```

```
2 + 3
```

```
## [1] 5
```

```
rep("ab", 5)
```

```
## [1] "ab" "ab" "ab" "ab" "ab"
```

```
x <- c(3,4,3,2.5,4)
mean(x)
```

```
## [1] 3.3
```

```
var(x)
```

```
## [1] 0.45
```

Multiple tests/comparisons

- Per-test- *vs.* per-experiment risk levels.
- Two approaches:
  1. Use especial methods like Tukey's HSD (honestly significant difference)
  2. Use usual contrast methods and then adjust the P-values with methods like Bonferroni's.
- Bonferroni's great popularity is based on easy of calculation.
- Better methods are currently available.

Adjusting P-values

```
my.p.values <- c(0.013, 0.05, 0.045, 0.001)
p.adjust(my.p.values, "none")

## [1] 0.013 0.050 0.045 0.001

p.adjust(my.p.values, "bonferroni")

## [1] 0.052 0.200 0.180 0.004

p.adjust(my.p.values, "holm")

## [1] 0.039 0.090 0.090 0.004
```

Method "none" is equivalent to using LSD, no correction for multiple tests.

Bonferroni's method

Holm's (1979) method

There are other good methods available.

Adjusting P-values

```
my.p.values <- c(0.013, 0.05, 0.045, 0.001)
p.adjust(my.p.values, "none")

## [1] 0.013 0.050 0.045 0.001

p.adjust(my.p.values, "bonferroni")

## [1] 0.052 0.200 0.180 0.004

p.adjust(my.p.values, "holm")

## [1] 0.039 0.090 0.090 0.004
```

Method "none" is equivalent to using LSD: avoid.

Bonferroni's method

Holm's (1979) method

There are other good methods available.

Adjusting P-values

```
my.p.values <- c(0.013, 0.05, 0.045, 0.001)
p.adjust(my.p.values, "none")

## [1] 0.013 0.050 0.045 0.001

p.adjust(my.p.values, "bonferroni")

## [1] 0.052 0.200 0.180 0.004

p.adjust(my.p.values, "holm")

## [1] 0.039 0.090 0.090 0.004
```

Method "none" is equivalent to using LSD: avoid.
Bonferroni's method is very easy to calculate ($\alpha \cdot 1/n$), this made it popular, but is very conservative.
Holm's (1979) method
There are other good methods available.

Adjusting P-values

```
my.p.values <- c(0.013, 0.05, 0.045, 0.001)
p.adjust(my.p.values, "none")

## [1] 0.013 0.050 0.045 0.001

p.adjust(my.p.values, "bonferroni")

## [1] 0.052 0.200 0.180 0.004

p.adjust(my.p.values, "holm")

## [1] 0.039 0.090 0.090 0.004
```

Method "none" is equivalent to using LSD: avoid.
Bonferroni's method: avoid.
Holm's (1979) method
There are other good methods available.

Adjusting P-values

```
my.p.values <- c(0.013, 0.05, 0.045, 0.001)
p.adjust(my.p.values, "none")

## [1] 0.013 0.050 0.045 0.001

p.adjust(my.p.values, "bonferroni")

## [1] 0.052 0.200 0.180 0.004

p.adjust(my.p.values, "holm")

## [1] 0.039 0.090 0.090 0.004
```

Method "none" is equivalent to using LSD: avoid.
Bonferroni's method: avoid.
Holm's (1979) method is more difficult to calculate, gives strong control of family-wise error rate and is valid under arbitrary assumptions.
There are other good methods available.

Adjusting P-values

```
my.p.values <- c(0.013, 0.05, 0.045, 0.001)
p.adjust(my.p.values, "none")

## [1] 0.013 0.050 0.045 0.001

p.adjust(my.p.values, "bonferroni")

## [1] 0.052 0.200 0.180 0.004

p.adjust(my.p.values, "holm")

## [1] 0.039 0.090 0.090 0.004
```

Method "none" is equivalent to using LSD: avoid.
Bonferroni's method: avoid.
Holm's (1979) method: usually good.
There are other good methods available.

Adjusting P-values

```
my.p.values <- c(0.013, 0.05, 0.045, 0.001)
p.adjust(my.p.values, "none")

## [1] 0.013 0.050 0.045 0.001

p.adjust(my.p.values, "bonferroni")

## [1] 0.052 0.200 0.180 0.004

p.adjust(my.p.values, "holm")

## [1] 0.039 0.090 0.090 0.004
```

Method "none" is equivalent to using LSD: avoid.
Bonferroni's method: avoid.
Holm's (1979) method: usually good.
There are other good methods available.

Transformations

- Applying a transformation to response data alters the functional relationship between dependent and independent variables.
- Transformations may help to fulfil the assumptions of an analysis method, but transformations drastically alter the interpretation of tests of significance and model selection.
- Two approaches:
  1. Old: use a simple and easy to calculate statistical procedure and transform the data to force it to agree with the expectations of the method used.
  2. Modern: use a more complex and difficult to calculate statistical procedure whose assumptions match the properties of the data.
- We start with a simple example of a factorial experiment. . .

Data transformations
In factorial experiments

Using a transformation changes the interpretation of interactions!

```
summary(aov(y ~ x * group, data=my.data))
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## x            1 1398.9  1398.9 304.391  < 2e-16 ***
## group        1  368.4   368.4  80.167 6.71e-11 ***
## x:group      1    0.0     0.0   0.008    0.931
## Residuals   38  174.6     4.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(aov(log(y) ~ x * group, data=my.data))
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## x            1  6.626   6.626  184.43 3.69e-16 ***
## group        1  1.998   1.998   55.60 5.98e-09 ***
## x:group      1  0.450   0.450   12.51  0.00108 **
## Residuals   38  1.365   0.036
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Data transformations
In factorial experiments

```
summary(aov(y ~ x * group, data=my.data))
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## x            1 1398.9  1398.9 304.391  < 2e-16 ***
## group        1  368.4   368.4  80.167 6.71e-11 ***
## x:group      1    0.0     0.0   0.008    0.931
## Residuals   38  174.6     4.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(aov(log(y) ~ x * group, data=my.data))
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## x            1  6.626   6.626  184.43 3.69e-16 ***
## group        1  1.998   1.998   55.60 5.98e-09 ***
## x:group      1  0.450   0.450   12.51  0.00108 **
## Residuals   38  1.365   0.036
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Data transformations
In factorial experiments

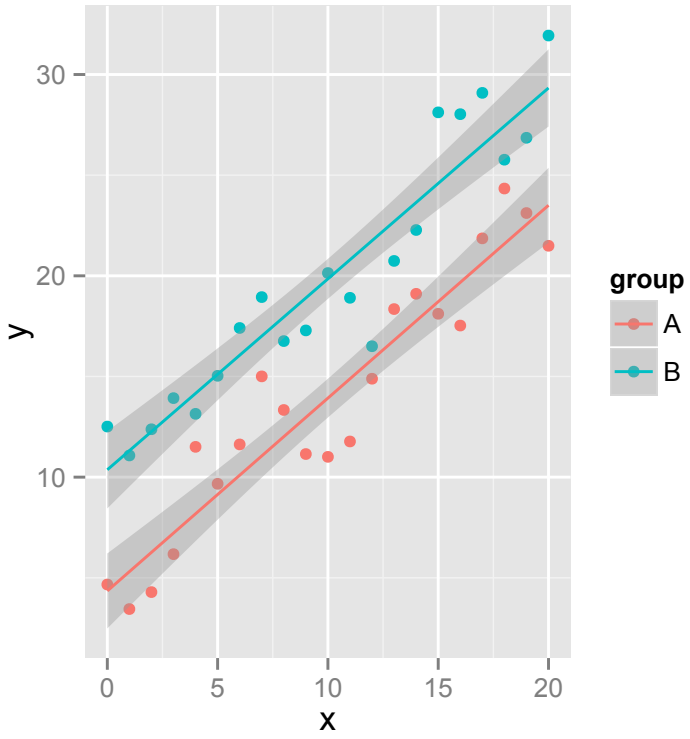
```
summary(aov(y ~ x * group, data=my.data))
```

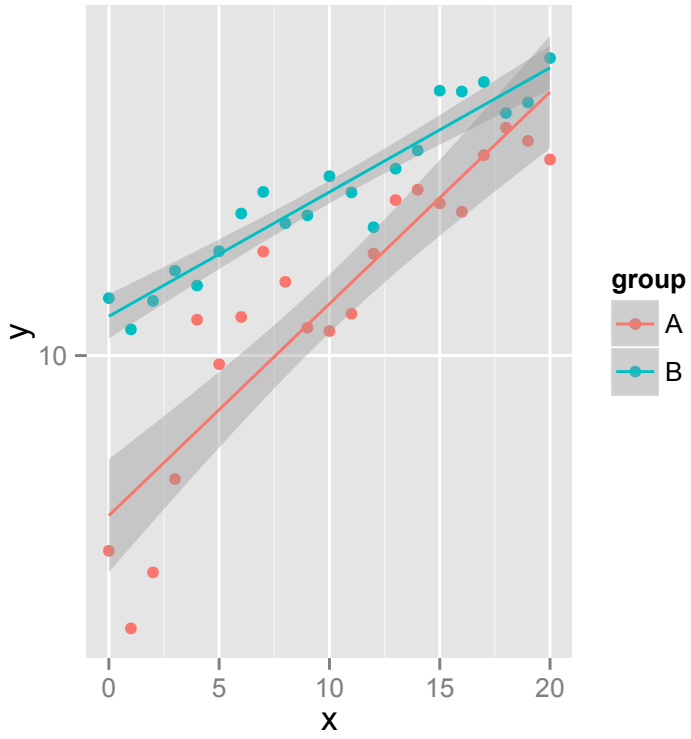
```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## x            1 1398.9  1398.9 304.391  < 2e-16 ***
## group        1  368.4   368.4  80.167 6.71e-11 ***
## x:group      1    0.0     0.0   0.008    0.931
## Residuals   38  174.6     4.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(aov(log(y) ~ x * group, data=my.data))
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## x            1  6.626   6.626  184.43 3.69e-16 ***
## group        1  1.998   1.998   55.60 5.98e-09 ***
## x:group      1  0.450   0.450   12.51  0.00108 **
## Residuals   38  1.365   0.036
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Data transformations
Plots





Dealing with outliers

- The basic question is: are these 'unusual' observations part of the statistical population of interest or are they accidental events that we are not interested in?
- What can be considered an outlier depends on the objectives of the research. Under some circumstances the 'unusual' observations may be the most interesting ones.
- In small data sets it is best to locate and then analyse the reasons for the existence of each individual outlier.
- In large data sets we need automated methods.
- Two approaches:
  1. Distribution-free methods
  2. Methods that assume a given distribution, and use this assumption to discard observations that are unlikely to belong to the assumed distribution.

Dealing with outliers (cont.)

- Two approaches:
  1. Distribution-free methods
  2. Methods that assume a given distribution, and use this assumption to discard observations that are unlikely to belong to the assumed distribution.
- Simple summaries like median and mode are distribution free. Also many re-sampling methods estimate the distribution of random variation from the data.
- There are also methods, which down-weight the influence of the observations unlikely to belong to a population with the assumed distribution (e.g. Normal).

Dealing with outliers (cont.)

- Two approaches:
  1. Distribution-free methods
  2. Methods that assume a given distribution, and use this assumption to discard observations that are unlikely to belong to the assumed distribution.
- Simple summaries like median and mode are distribution free. Also many re-sampling methods estimate the distribution of random variation from the data.
- There are also methods, which down-weight the influence of the observations unlikely to belong to a population with the assumed distribution (e.g. Normal).

Dealing with outliers (cont.)

- Two approaches:
  1. Distribution-free methods
  2. Methods that assume a given distribution, and use this assumption to discard observations that are unlikely to belong to the assumed distribution.
- Simple summaries like median and mode are distribution free. Also many re-sampling methods estimate the distribution of random variation from the data.
- There are also methods, which down-weight the influence of the observations unlikely to belong to a population with the assumed distribution (e.g. Normal).

Data without outliers

Linear model fit, ANOVA

```
summary(lm(y ~ x * group, data=my.data))
```

```
##
## Call:
## lm(formula = y ~ x * group, data = my.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2439 -1.6541  0.0208  1.4860  3.9528
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.347641   0.903149   4.814 2.37e-05 ***
## x            0.957835   0.077255  12.398 6.31e-15 ***
## groupB       6.018602   1.277246   4.712 3.25e-05 ***
## x:groupB    -0.009511   0.109255  -0.087    0.931
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.144 on 38 degrees of freedom
## Multiple R-squared:  0.9101, Adjusted R-squared:  0.903
## F-statistic: 128.2 on 3 and 38 DF,  p-value: < 2.2e-16
```

Data with outliers
Linear model fit, ANOVA

```
summary(lm(y ~ x * group, data=my.dirty.data))
```

```
##
## Call:
## lm(formula = y ~ x * group, data = my.dirty.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.9375  -2.9306  -0.4991   2.4249  27.3806
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.6732     3.0100   2.549   0.0150 *
## x             0.6454     0.2575   2.506   0.0166 *
## groupB        7.5161     4.2568   1.766   0.0855 .
## x:groupB     -0.1594     0.3641  -0.438   0.6641
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.145 on 38 degrees of freedom
## Multiple R-squared:  0.3098, Adjusted R-squared:  0.2554
## F-statistic: 5.687 on 3 and 38 DF,  p-value: 0.00256
```

Data with outliers
Robust linear model fit, ANOVA

```
summary(rlm(y ~ x * group, data=my.dirty.data))
```

```
##
## Call: rlm(formula = y ~ x * group, data = my.dirty.data)
## Residuals:
##      Min       1Q   Median       3Q      Max
## -27.2840  -1.8183  -0.2331   1.6357  29.5062
##
## Coefficients:
##             Value  Std. Error t value
## (Intercept) 5.4953 1.1889     4.6222
## x           0.8630 0.1017     8.4859
## groupB      5.4638 1.6813     3.2496
## x:groupB    0.0440 0.1438     0.3056
##
## Residual standard error: 2.571 on 38 degrees of freedom
```

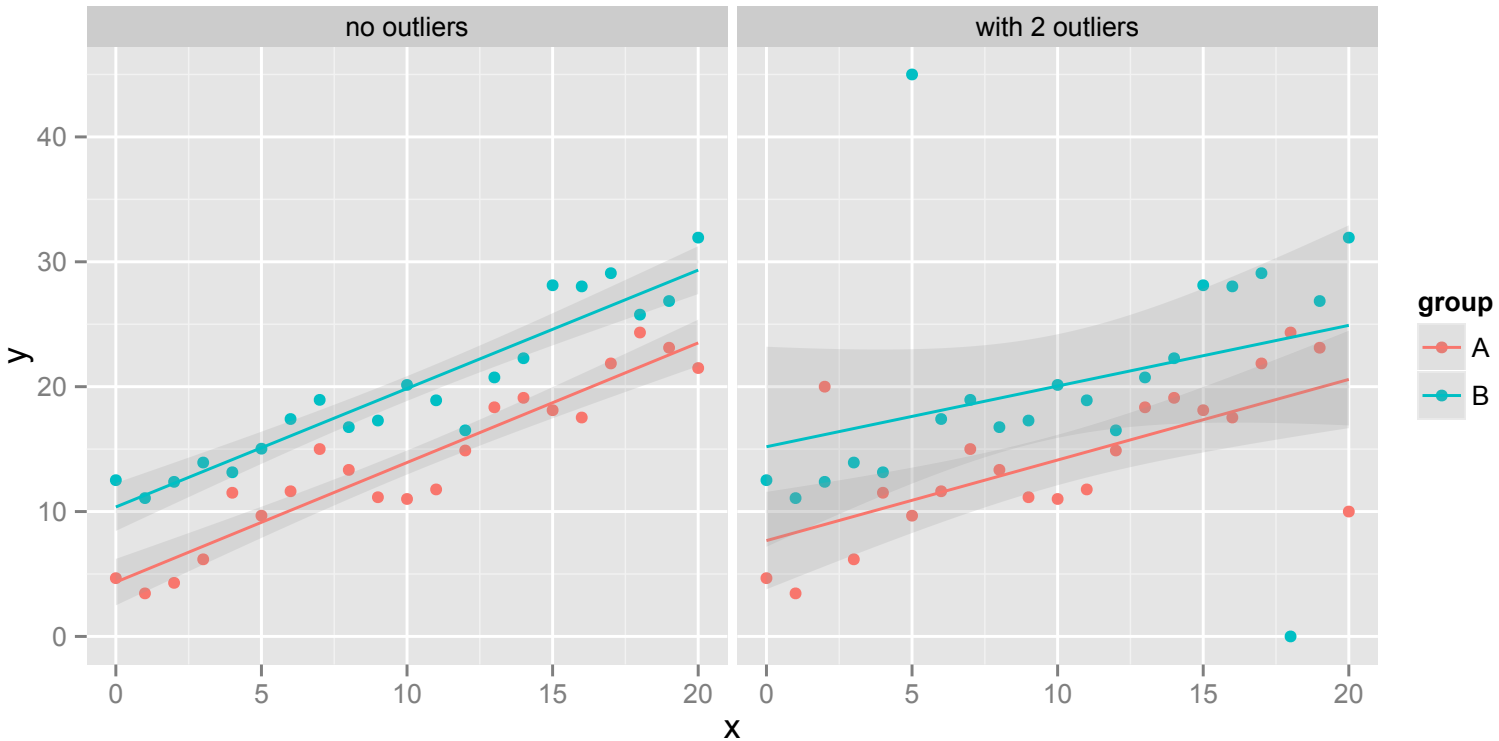
Data with outliers
WLE linear model fit, ANOVA

```
summary(wle.lm(y ~ x * group, data=my.dirty.data))
```

```
##
## Call:
## wle.lm(formula = y ~ x * group, data = my.dirty.data)
##
## Root  1
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9464 -1.5237  0.1414  1.3024  3.7502
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.502350   1.007289   4.470 8.57e-05 ***
## x           0.964654   0.088456  10.905 1.56e-12 ***
## groupB      5.788672   1.396705   4.145  0.00022 ***
## x:groupB    0.006348   0.121998   0.052  0.95881
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.165 on 33.32042 degrees of freedom
## Multiple R-Squared: 0.9055, Adjusted R-squared: 0.897
## F-statistic: 106.5 on 3 and 33.32042 degrees of freedom, p-value:     0
```

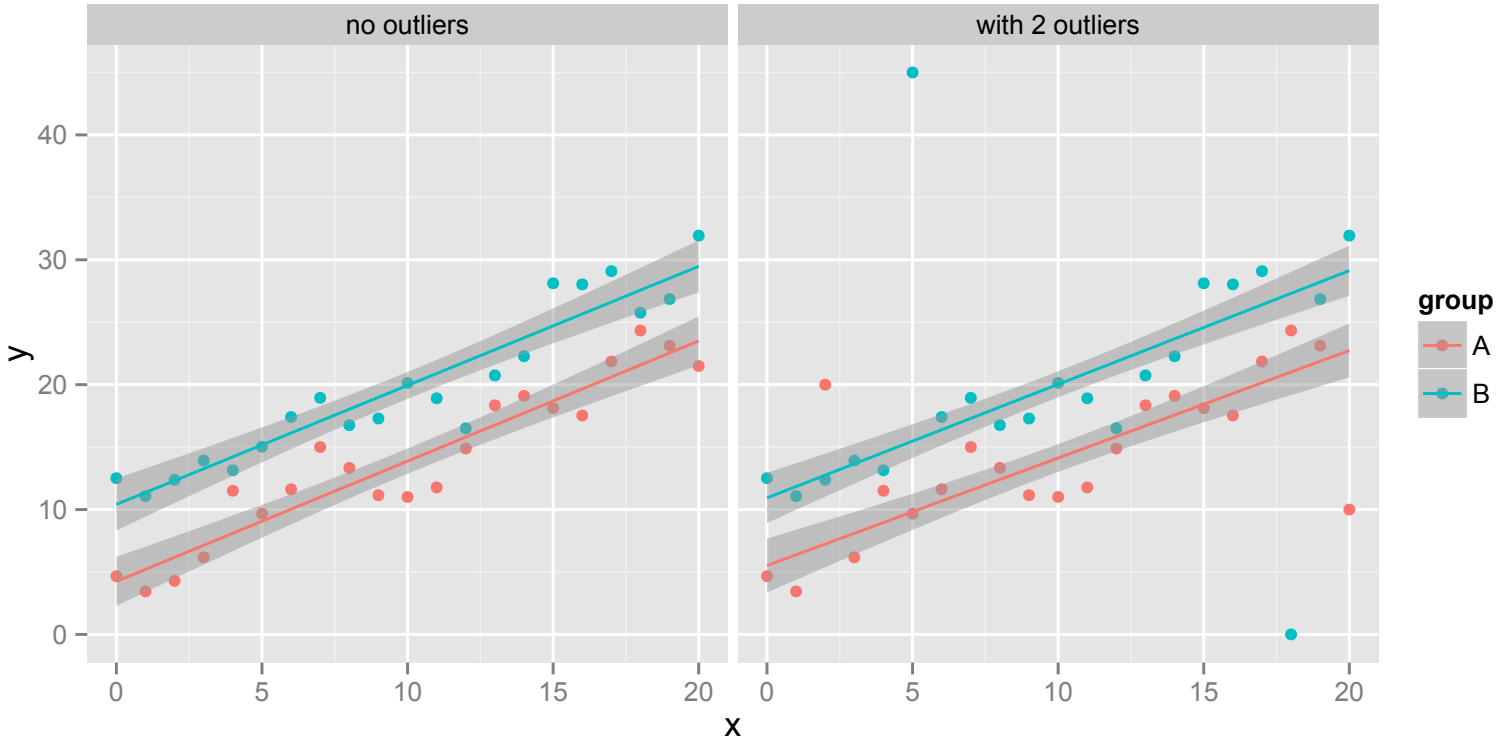
Data with outliers
Plots

Before and after adding outliers: Linear model fit using `lm`



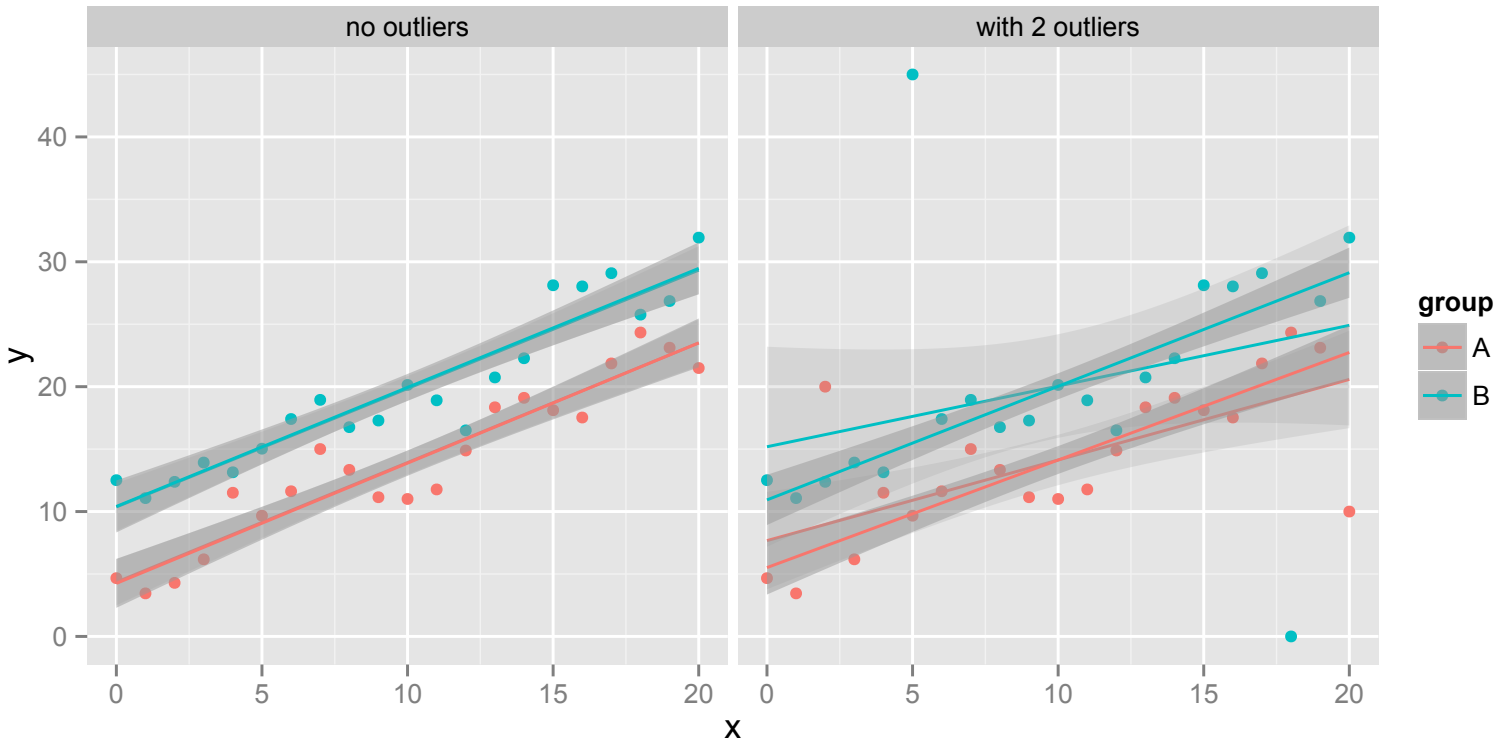
Data with outliers
Plots

Before and after adding outliers: Robust linear model fit using `rlm`



Data with outliers
Plots

Before and after adding outliers: Both fits in the same plot



GLS

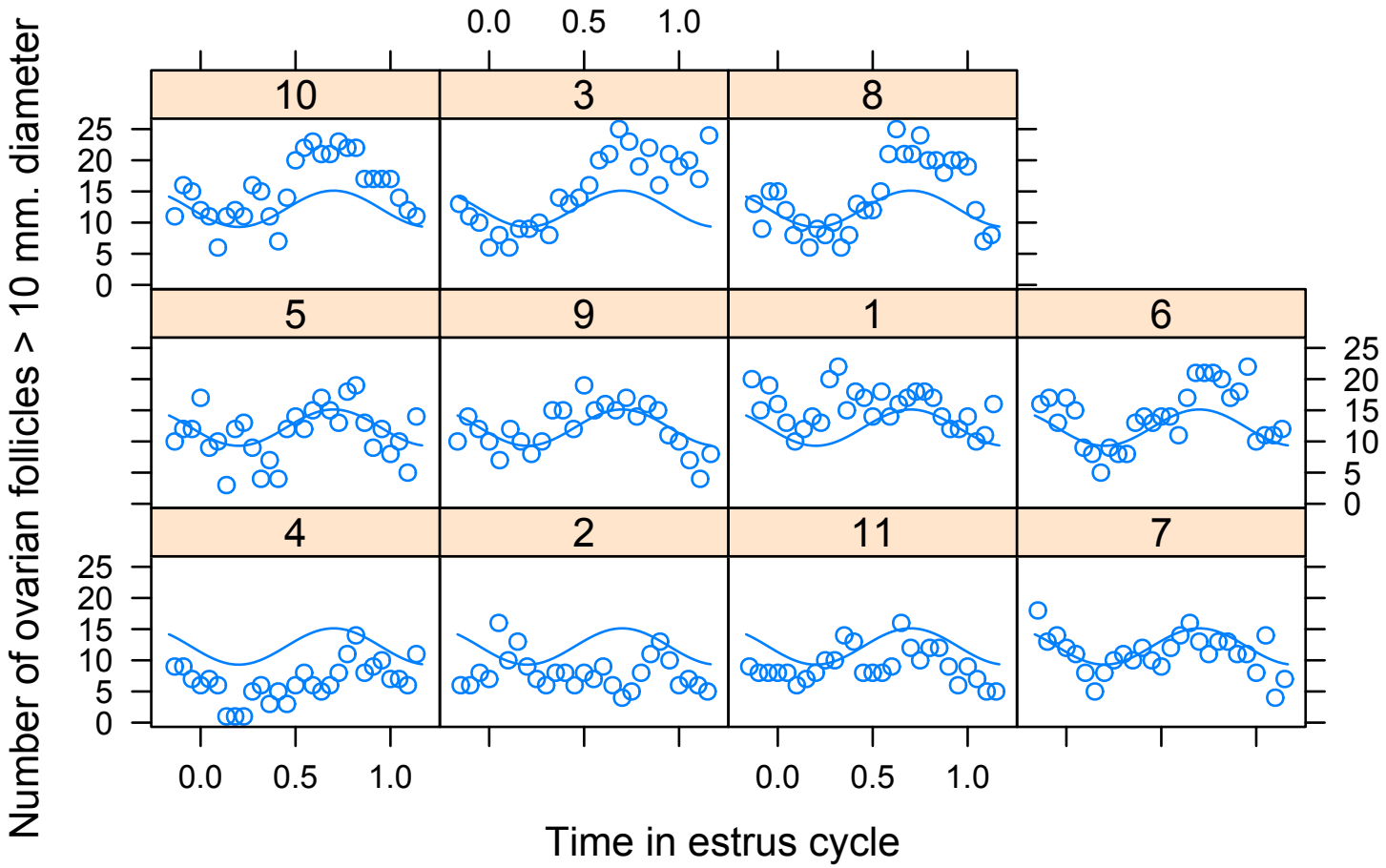
- GLS are basically LM (linear models) in which we can fit parameters describing the random variation in a more flexible way.
- Variance covariate: we can describe changes in error variance as a function of continuous variables (e.g. fitted values, or a covariate like time or age)
- We can define the 'structure' of the error variation. (e.g. separate estimates of error variance for different treatments, or a certain correlation, or auto-correlation structure for the errors).

GLS
Fit

```
# AR(1) errors within each Mare
fm1 <- gls(follicles ~ sin(2*pi*Time) + cos(2*pi*Time), Ovary,
           correlation = corAR1(form = ~ 1 | Mare))
anova(fm1)


## Denom. DF: 305
##                     numDF  F-value p-value
## (Intercept)             1 354.7332  <.0001
## sin(2 * pi * Time)      1  18.5034  <.0001
## cos(2 * pi * Time)      1   1.6633  0.1981
```

Note: This is a model linear in the parameters.

GLS
Plots


Number of ovarian follicles > 10 mm. diameter
Time in estrus cycle
10
3
8
5
9
1
6
4
2
11
7
0.0
0.5
1.0
0
5
10
15
20
25

LME

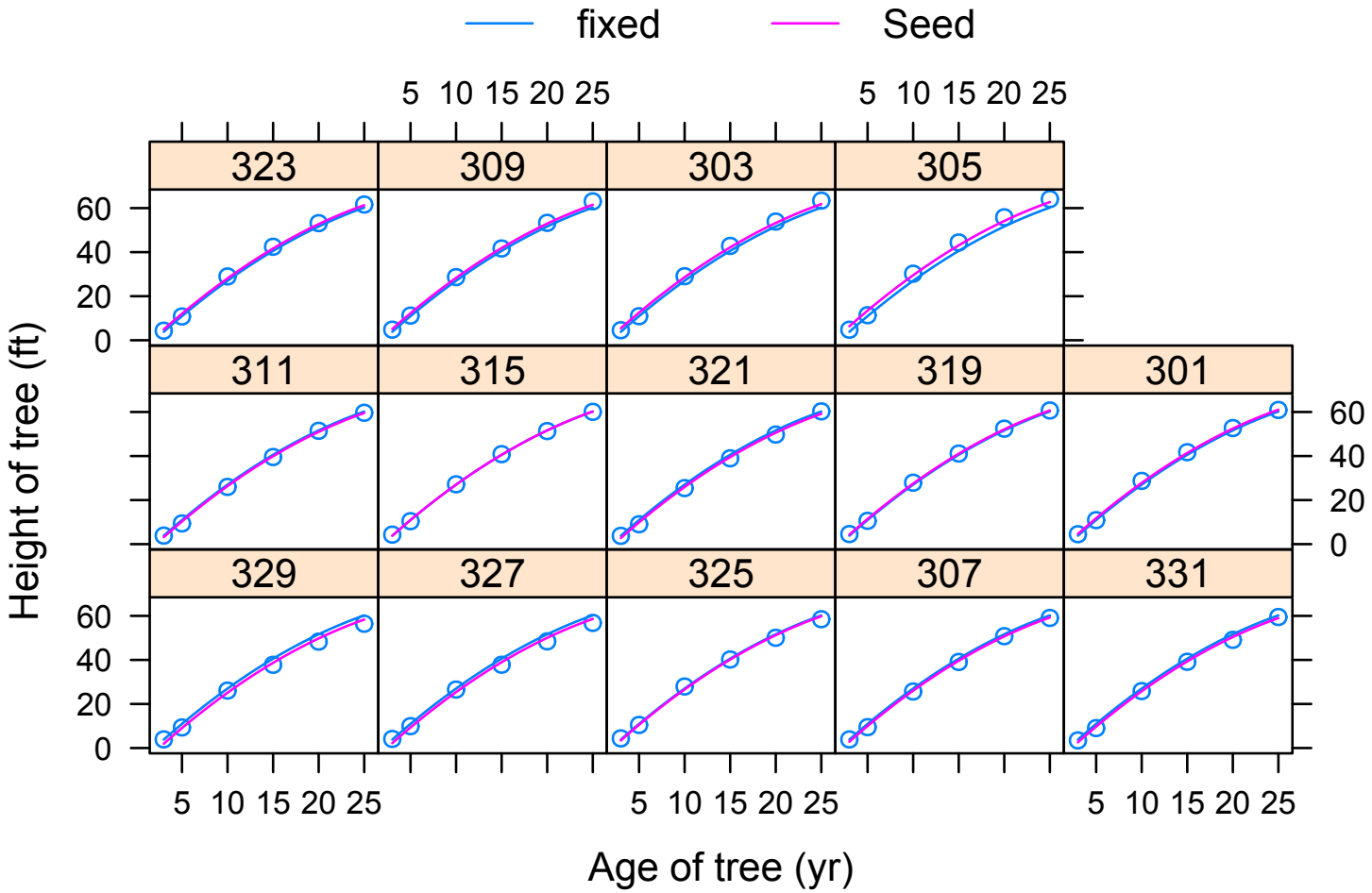
- Linear Mixed Effects models, can be thought as a GLS, in which we include in addition of fixed effects (i.e. the applied treatments), random effects (e.g. years in a field experiment).
- The basic idea is that for fixed effects, we are interested in the individual levels (e.g. control *vs.* treated), while for random effects we are not.
- Mixed models allow also the description of the nesting of treatments in factorial experiments (e.g. split-unit and repeated measures experimental designs).

LME
Fitting a 2nd degree polynomial

```
fm1 <- lme(height ~ age + I(age^2),
           data = Loblolly,
           random = ~ 1 | Seed)
summary(fm1)
```

```
## Linear mixed-effects model fit by REML
##  Data: Loblolly
##        AIC      BIC    logLik
##   300.9795 312.9517 -145.4897
##
## Random effects:
##  Formula: ~1 | Seed
##         (Intercept) Residual
## StdDev:    1.387033 1.022571
##
## Fixed effects: height ~ age + I(age^2)
##                 Value Std.Error DF   t-value p-value
## (Intercept) -7.607232 0.5203803 68 -14.61860       0
## age          3.959044 0.0656130 68  60.33931       0
## I(age^2)    -0.049838 0.0023328 68 -21.36436       0
##  Correlation:
##          (Intr) age
## age      -0.630
## I(age^2)  0.566 -0.976
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -2.04794270 -0.58771042  0.07584935  0.63550743  1.94543736
##
## Number of Observations: 84
## Number of Groups: 14
```

LME
Plots


fixed
Seed
323
309
303
305
311
315
321
319
301
329
327
325
307
331
Height of tree (ft)
Age of tree (yr)
0
20
40
60
5 10 15 20 25

NLME

- The difference with LME, is that instead of fitting a function that is linear in the parameters, we fit a function that is non-linear in the parameters.
- This is useful when theory suggests a certain mathematical relationship, and obtaining estimates of values for biologically meaningful parameters, requires the use of such functions.
- Compared to separate fits to data for each experimental unit it has advantages of making better use of the information in the data set, and of taking into account correlations among parameter estimates in a more 'integrated' manner.

NLME

Fitting $y = \Phi_1 + (\Phi_1 - \Phi_2) \cdot e^{-\Phi_3 x}$

SSasymp is a predefined function for this 'asymptotic' function. The SS in the name means self-starting, which means that in many cases it would be able to find good starting values for the iterative calculations. However, in this example we are explicitly supplying the starting values for the parameters.
In the code listed below Φ_1, Φ_2, Φ_3 are called `Asym`, `R0`, `lrc`, respectively.

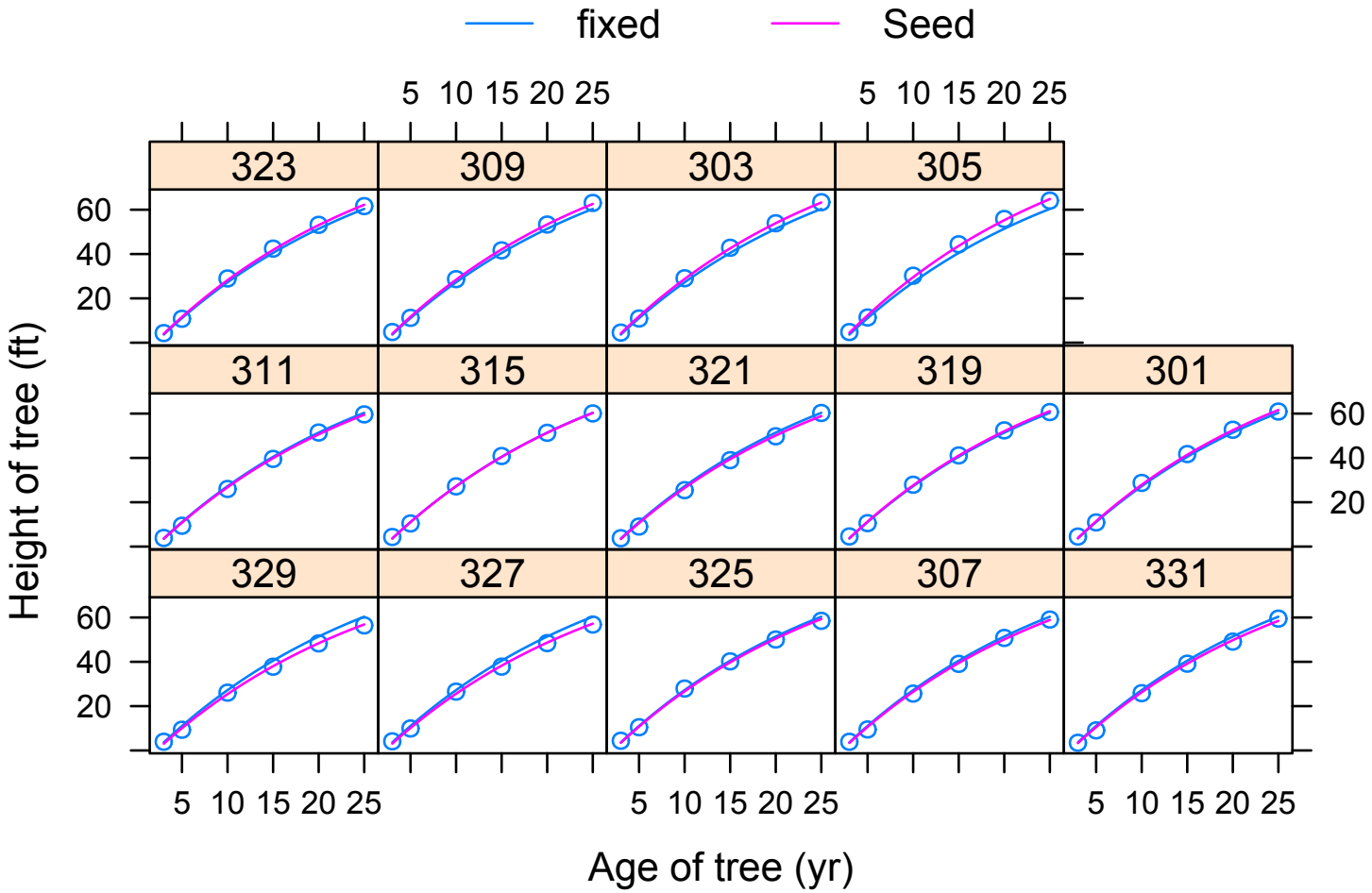
```
fm1 <- nlme(height ~ SSasymp(age, Asym, R0, lrc),
            data = Loblolly,
            fixed = Asym + R0 + lrc ~ 1,
            random = Asym ~ 1,
            start = c(Asym = 103, R0 = -8.5, lrc = -3.3))
```

NLME

Fitting $y = \Phi_1 + (\Phi_1 - \Phi_2) \cdot e^{-\Phi_3 x}$

```
## Nonlinear mixed-effects model fit by maximum likelihood
##   Model: height ~ SSasymp(age, Asym, R0, lrc)
##  Data: Loblolly
##        AIC      BIC    logLik
##   239.4856 251.6397 -114.7428
##
## Random effects:
##  Formula: Asym ~ 1 | Seed
##             Asym  Residual
## StdDev: 3.650642 0.7188625
##
## Fixed effects: Asym + R0 + lrc ~ 1
##          Value Std.Error DF   t-value p-value
## Asym 101.44960 2.4616951 68  41.21128       0
## R0    -8.62733 0.3179505 68 -27.13420       0
## lrc   -3.23375 0.0342702 68 -94.36052       0
##  Correlation:
##     Asym   R0
## R0   0.704
## lrc -0.908 -0.827
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -2.23601930 -0.62380854  0.05917466  0.65727206  1.95794425
##
## Number of Observations: 84
## Number of Groups: 14
```

NLME
Plots


fixed
Seed
323
309
303
305
311
315
321
319
301
329
327
325
307
331
Height of tree (ft)
Age of tree (yr)
5 10 15 20 25
20
40
60

GLM

- GLMs remove the assumption of normally distributed residuals from LMs, and allows the explicit specification of an error distribution function.
- Examples of distributions available are: binomial, Poisson, exponential, etc.
- Examples of link functions are: logarithms, powers, etc.

GLM
Generation of artificial binomial data

```
x <- 1 + rnorm(1000,1)
xbeta <- -1  + (x* 1)
proba <- exp(xbeta)/(1 + exp(xbeta))
y <- ifelse(runif(1000,0,1) < proba,1,0)
table(y)
```

```
## y
##   0   1
## 275 725
```

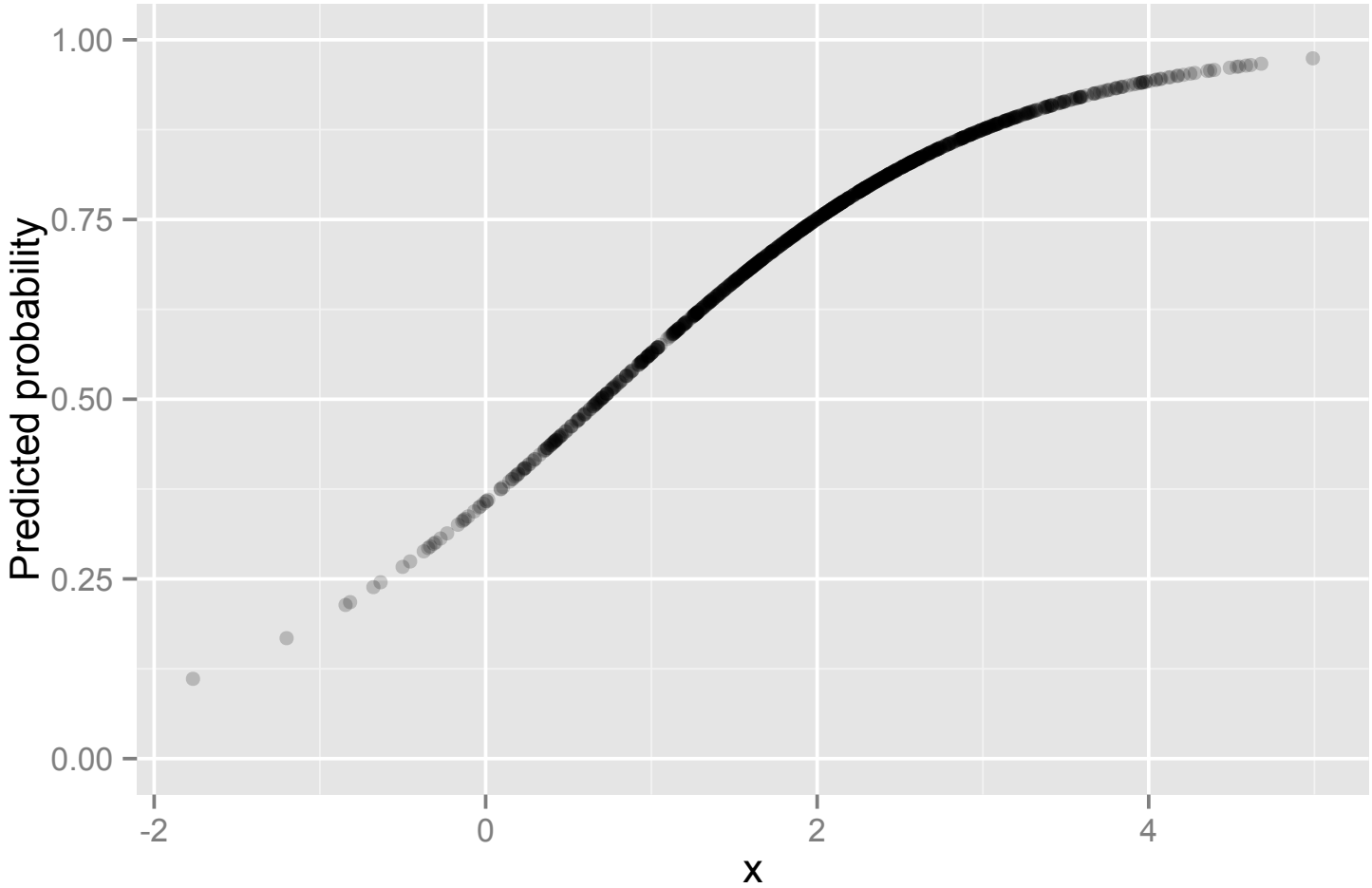
```
df <- data.frame(x,y)
head(df, 8)
```

```
##          x y
## 1 1.163160 1
## 2 2.600422 0
## 3 1.754065 1
## 4 1.815335 1
## 5 2.023157 1
## 6 1.515348 1
## 7 1.262094 0
## 8 3.302508 1
```

GLM
Fitting a model to binomial data

```
res <- glm(y ~ x , family = binomial(link=logit), data = df)
summary(res)


##
## Call:
## glm(formula = y ~ x, family = binomial(link = logit), data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4326  -0.9934   0.5911   0.7965   1.6258
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.58768    0.16347  -3.595 0.000324 ***
## x            0.84543    0.08474   9.977  < 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: 1176.3  on 999  degrees of freedom
## Residual deviance: 1057.6  on 998  degrees of freedom
## AIC: 1061.6
##
## Number of Fisher Scoring iterations: 4
```

GLM
Plots



GLMM

- GLMMs remove the assumption of normally distributed residuals from LME models, and allow the explicit specification of an error distribution function.
- No example will be given for this type of model.

Additive models, AM

- Sometimes we need to fit a continuous response function, but either we do not want to make any assumptions about the underlying functional relationship, or we need a very 'flexible' function.
- If we combine in the same model continuous responses described by splines and a grouping factor we have what is commonly called and additive model.
- This approach can be extended in two ways that can also be combined:
  1. Extend additive models to include mixed effects, variance covariates, and other distributions for random variation in addition to the Normal. AMM, GAM, GAMM.
  2. Extend the use of splines to the description of how the parameters of the error distribution change as function of a covariate. GAMMLSS.

GAM
Fitting a time series

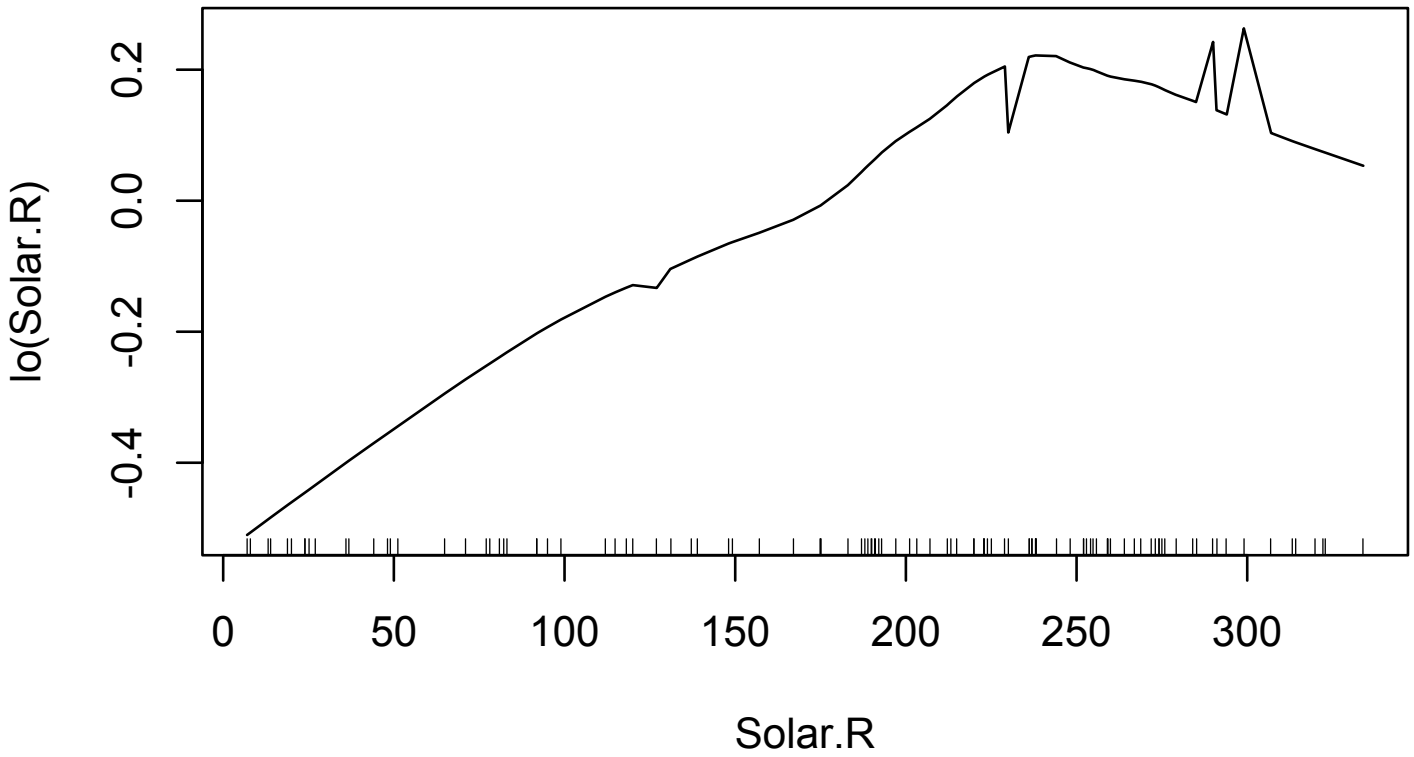
We use loess (lo) smooth terms of solar irradiance, wind speed, and temperature to predict ozone concentration from a time series of daily measurements of air quality. We accept the default of assuming a Normal distribution.

```
fgam1 <- gam(Ozone^(1/3) ~ lo(Solar.R) + lo(Wind, Temp), data=airquality, na=na.gam.replace)

## Warning in (function (frame) :  37 observations omitted due to missing values in the response

anova(fgam1)

## Anova for Nonparametric Effects
##                 Npar Df Npar F     Pr(F)
## (Intercept)
## lo(Solar.R)         2.6 1.7924    0.1614
## lo(Wind, Temp)      6.8 7.1107 8.621e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

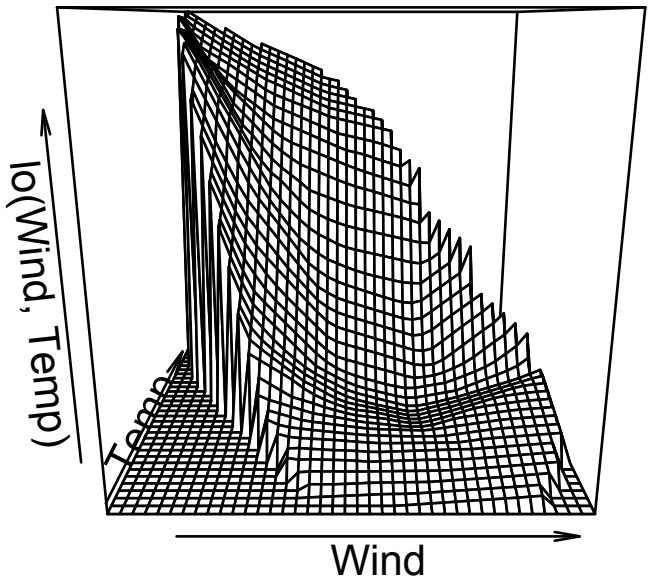
GAM
Summary

```
##
## Call: gam(formula = Ozone^(1/3) ~ lo(Solar.R) + lo(Wind, Temp), data = airquality,
##     na.action = na.gam.replace)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.24948 -0.27358 -0.01743  0.31587  0.96546
##
## (Dispersion Parameter for gaussian family taken to be 0.1997)
##
##     Null Deviance: 90.7149 on 115 degrees of freedom
## Residual Deviance: 20.503 on 102.6761 degrees of freedom
## AIC: 156.8112
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                   Df Sum Sq Mean Sq F value    Pr(>F)
## lo(Solar.R)      1.00 15.562 15.5622  77.933 3.024e-14 ***
## lo(Wind, Temp)   2.00 40.452 20.2261 101.289 < 2.2e-16 ***
## Residuals      102.68 20.503  0.1997
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                 Npar Df Npar F     Pr(F)
## (Intercept)
## lo(Solar.R)         2.6 1.7924    0.1614
## lo(Wind, Temp)      6.8 7.1107 8.621e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

AM
Plots



AM
Plots



Unknown groups

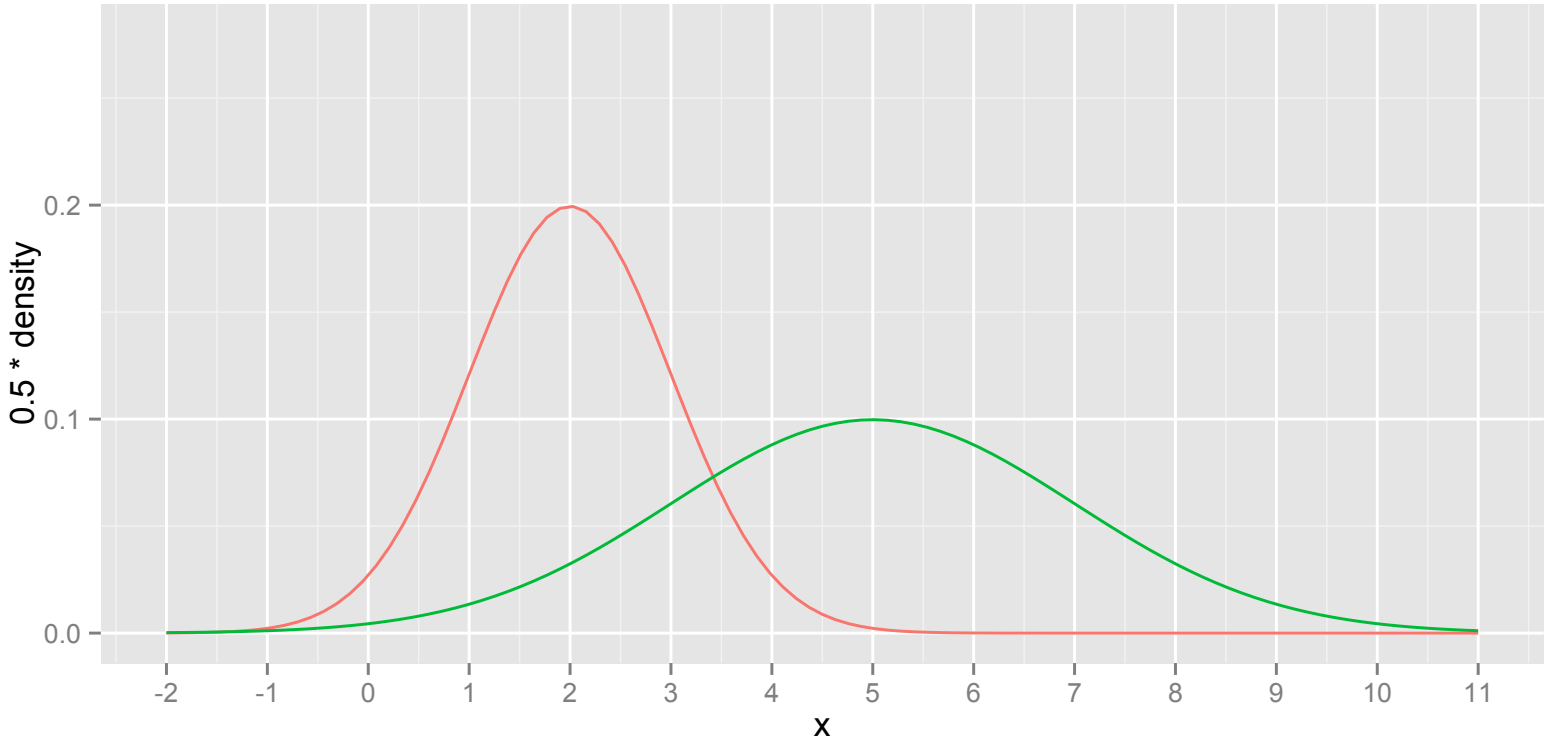
- Sometimes we know that our data set is composed of a mix of observations from two (or a few) populations, say males and females.
- However, we have not recorded the gender of the subjects studied at the time of data collection.
- We may still be interested in estimating how many observations belong to each of these groups, and what are the estimated mean and variance for each of these groups.
- This information can be 'recovered' by fitting a mixture model.
- As most statistical methods, we need to make assumptions about the populations. For example, that each sub-population follows a Normal distribution.

Unknown groups (cont.)

- Some methods do not require an *a priori* assumption on the number of sub-populations, or whether they have the same variance or same mean.
- However, such restrictions can be imposed if relevant to the research problem at hand.
- As opposed to classification methods the fitting of mixture models is best suited to situations where the number of groups of sub-populations is small, and the number of observations in each sub-populations is relatively high.
- Furthermore, a mixture model estimates the parameters of the distributions of the different groups, but does not 'assign' individual observations to the groups.
- The advantage is the flexibility, and that estimates of the values of parameters (and estimates of the confidence of these estimates) can be obtained for each of the sub-populations/groups.

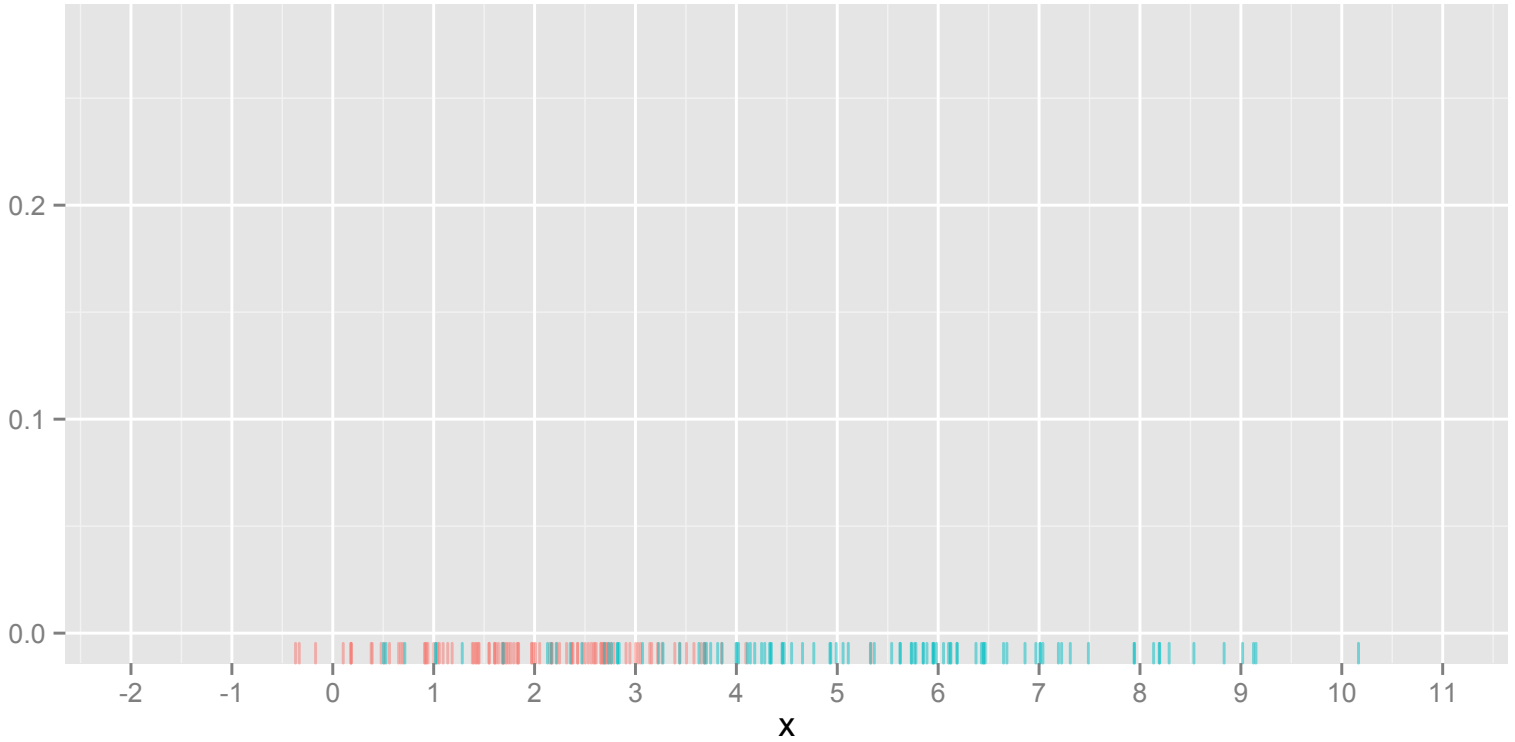
Mixture model example
Plot

We start with two theoretical Normal distributions



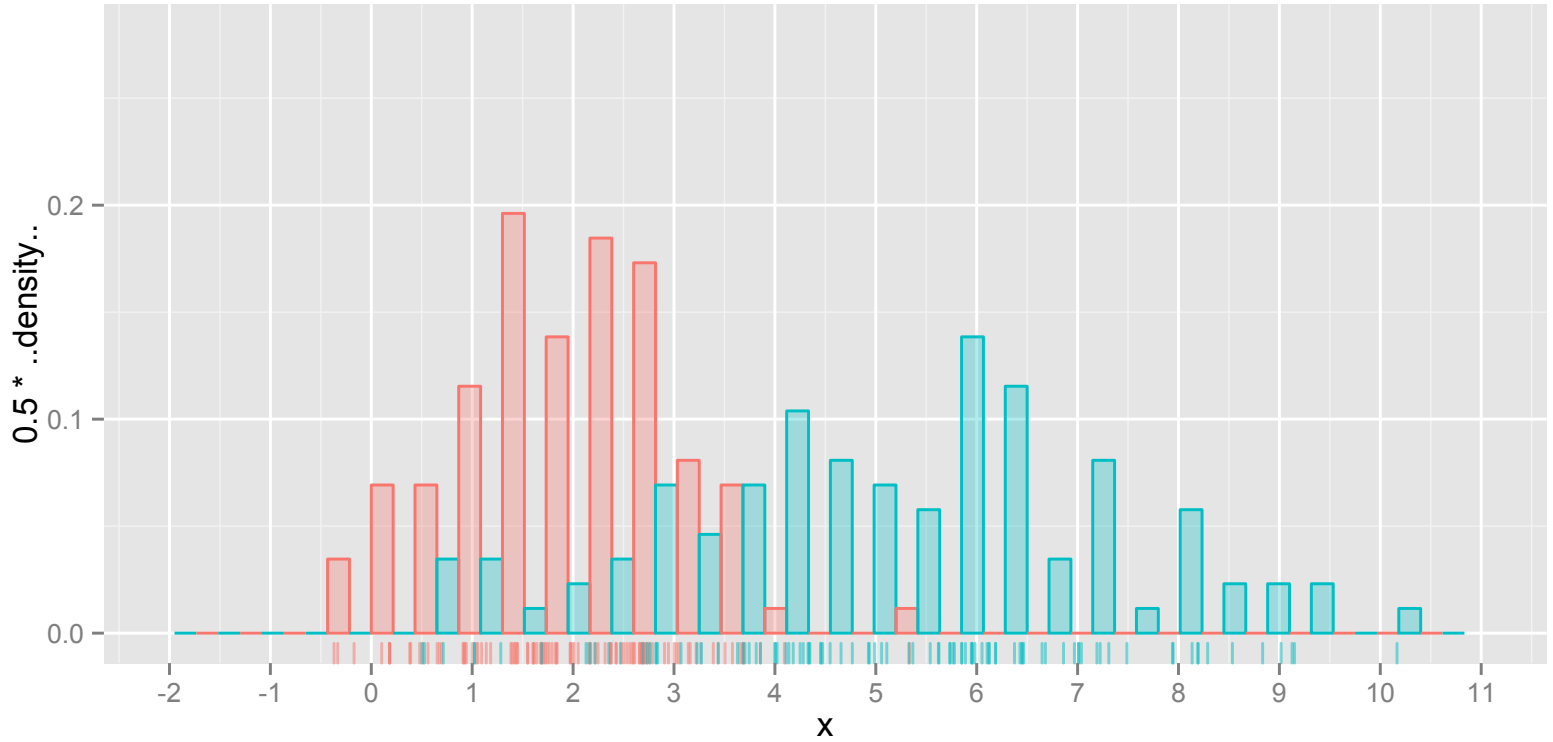
Mixture model example
Plot

We generate 101 artificial observations for each distribution



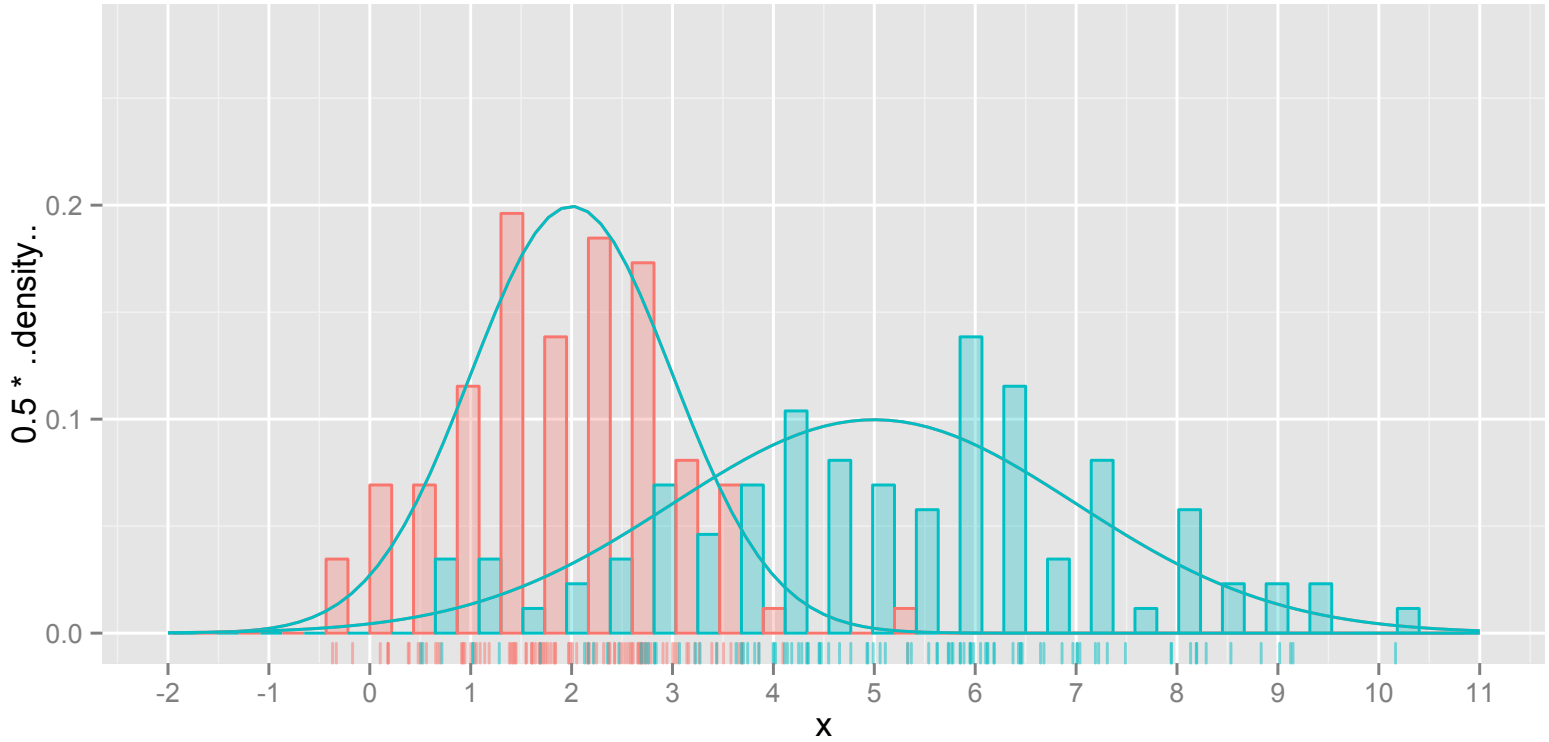
Mixture model example
Plot

We calculate a histogram for each of the groups



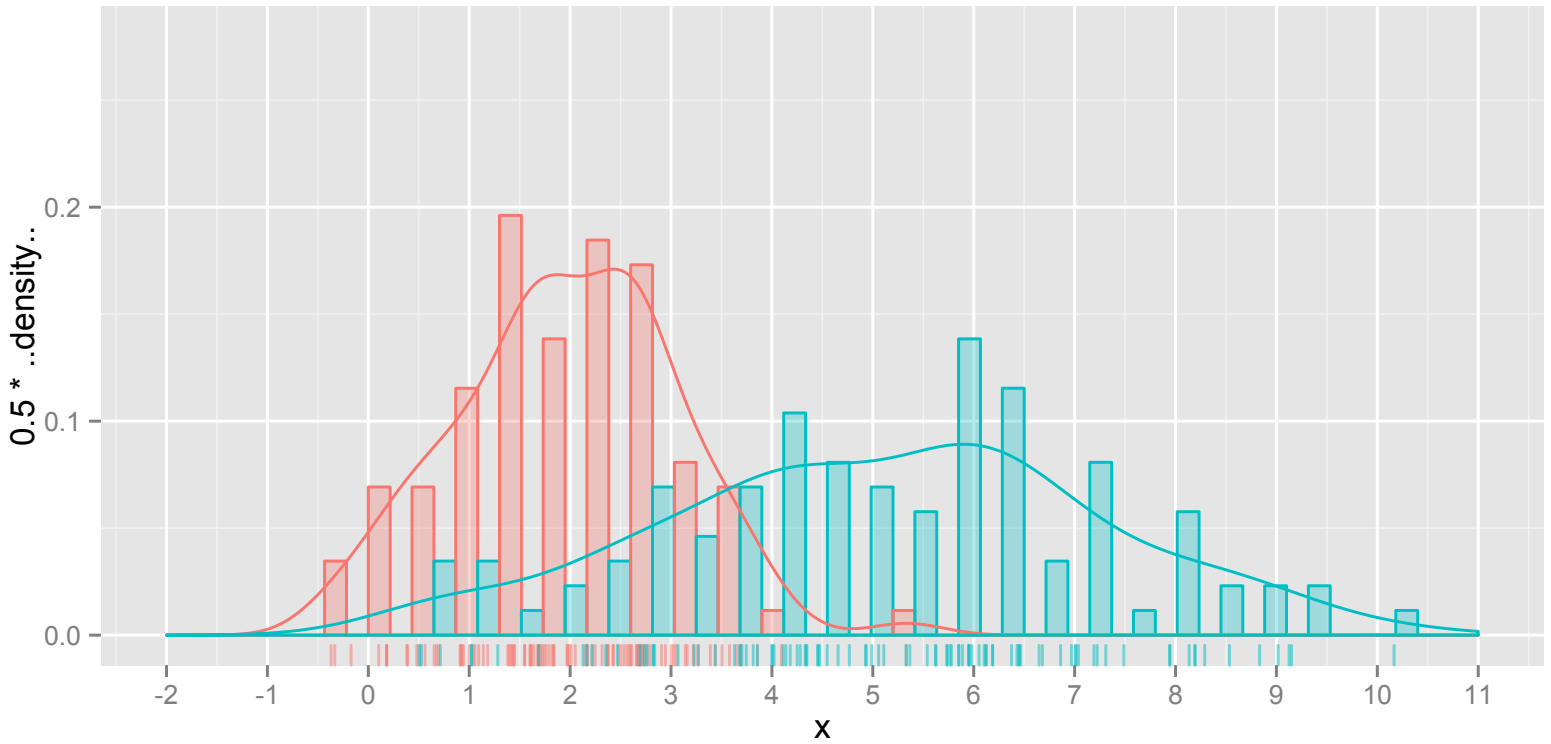
Mixture model example
Plot

We overlay the theoretical distributions on the histograms



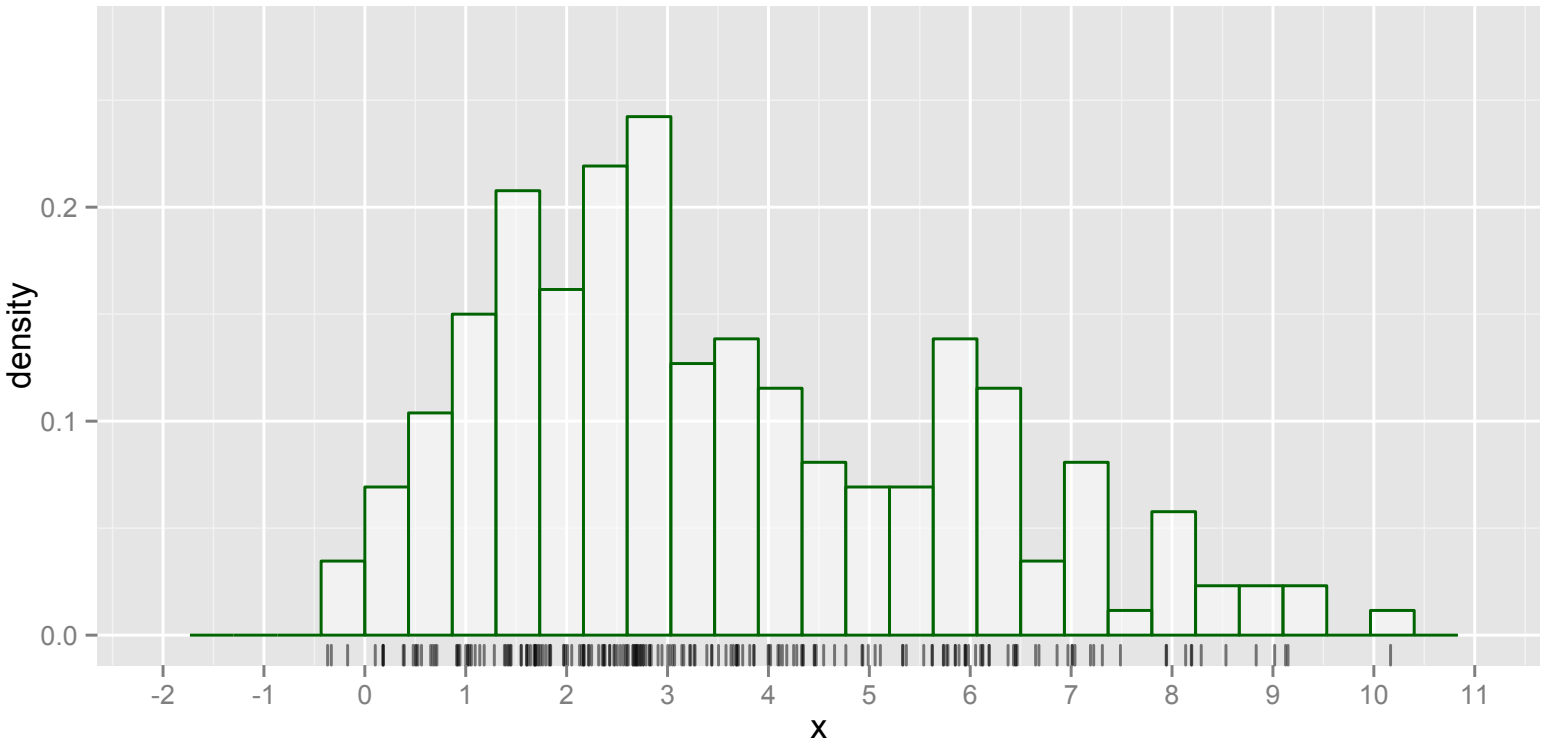
Mixture model example
Plot

We overlay a fitted density curve on top of each histogram



Mixture model example
Plot

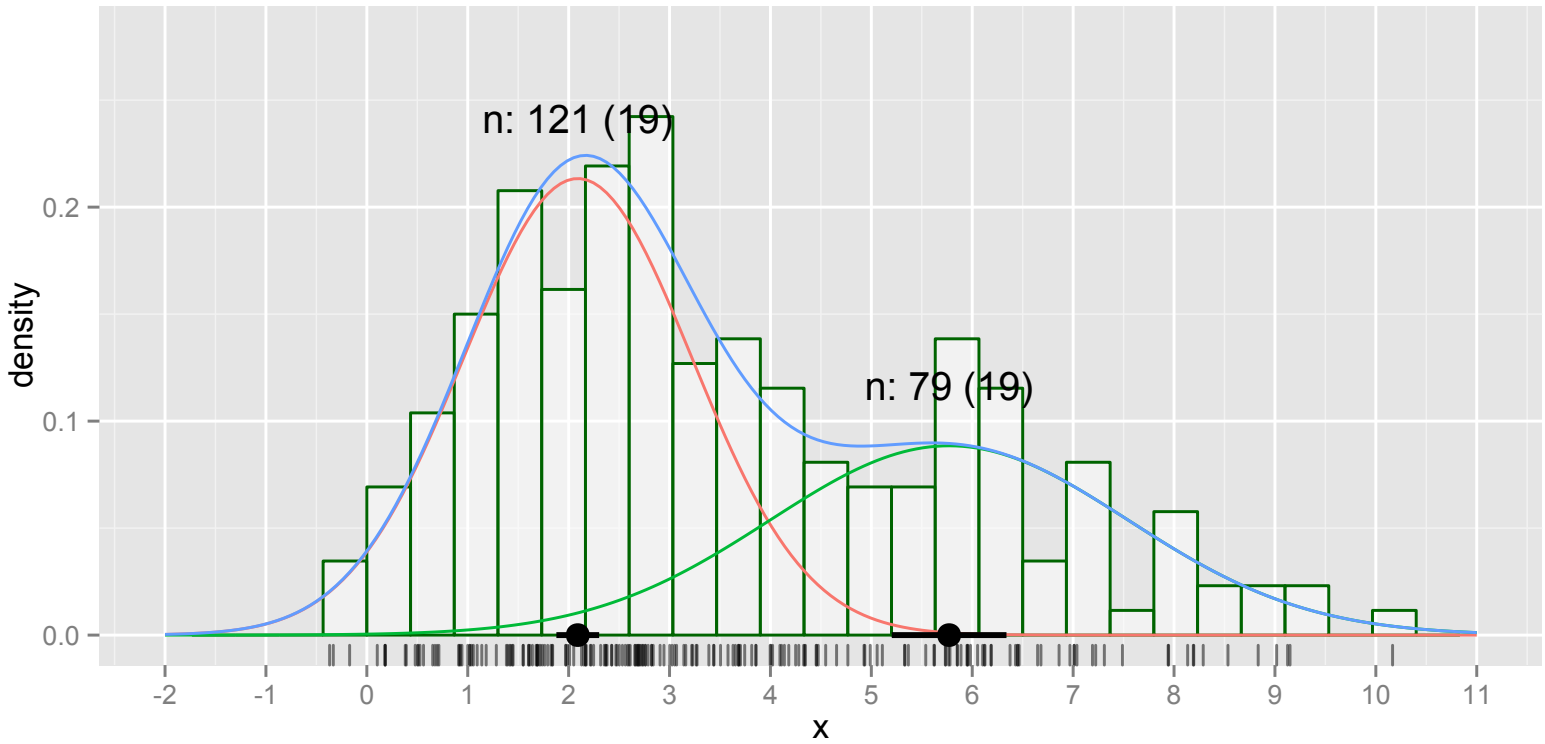
We 'mix' the data into a single group



Mixture model example

Plot

We fit a Normal-mixture model



Conclusions

- Nowadays large data sets can be analysed on a PC.
- Nowadays the range of methods that can be used on a PC is huge.
- Choosing methods to use, and models to fit has become much more complex.
- But in the end, we can extract more reliable information from data.
- We can also use experimental designs that either have fewer restrictions, and/or can better answer the scientific questions of interest.
- We should keep up-to-date, and also teach our students up-to-date ways of doing data analysis and designing experiments.

The End

- I prepared this slide presentation with LaTeX and R, on the RStudio IDE. I used Beamer and knitr, ggplot2 and other packages. This is all free open-source software, available for MS-Windows, OS-X, Linux, and Unix.
- This whole presentation (including examples) is coded in a single text file (except for the logos).
- To be continued. . . Part 4: Reproducible research.
- **Thanks for listening!**