Multilevel Modeling with R

Spirin Nikita

Dorodnicyn Computing Center of the Russian Academy of Sciences

03.24.2010, Moscow

Packages covered

- SAS
- MySQL
- Python
- Mathematica
- R

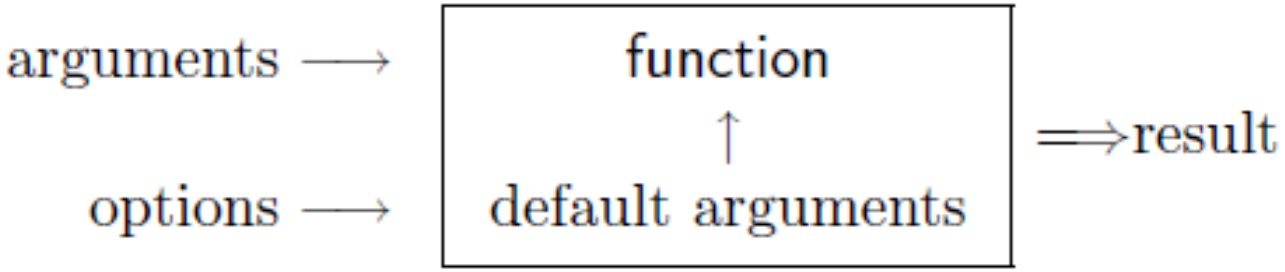
Agenda

- R programming language and R Paradigm
- Basic operations in R
- Graphics with R
- Statistics with R
- Multilevel Models and ML with R

8 min. times 5 equals 40 min.

Overview

- Free and commercialized
- GNU GPL
- R core team
- http://cran.r-project.org
- Interpreter



Concepts

- Actions with in-memory objects
- function()
- function
- library

Basic Notation

object	modes	several modes possible in the same object?
vector	numeric, character, complex *or* logical	No
factor	numeric *or* character	No
array	numeric, character, complex *or* logical	No
matrix	numeric, character, complex *or* logical	No
data frame	numeric, character, complex *or* logical	Yes
ts	numeric, character, complex *or* logical	No
list	numeric, character, complex, logical, function, expression, ...	Yes

Basic Notation

- A-Z and a-z
- _
- .
- 0-9
- Case Sensitive

Basic operations in R

- "assign" operator

```
> n <- 10 + 2
> n
[1] 12
> n <- 3 + rnorm(1)
> n
[1] 2.208807
```

Basic operations in R

- ls() function

```
> name <- "Carmen"; n1 <- 10; n2 <- 100; m <- 0.5
> ls()
[1] "m"     "n1"    "n2"    "name"


> ls(pat = "m")
[1] "m"     "name"


> ls(pat = "^m")
[1] "m"
```

Basic operations in R

- HELP

```
> ?lm

help(lm)

help("lm")

help.search("tree")

> apropos(help)
[1] "help"          ".helpForCall" "help.search"
[4] "help.start"
```

Reading Data

- getwd()
- setwd()

- readtable()
- scan()
- read.fwf()

- ASCII
- Excel, SAS, SPSS, SQL-type databases

Reading Data

```
read.csv(file, header = TRUE, sep = ",", quote="\"", dec=".",
         fill = TRUE, ...)
```

```
> mydata <- scan("data.dat", what = list("", 0, 0))
```

```
> mydata <- read.fwf("data.txt", widths=c(1, 4, 3))
> mydata
  V1   V2  V3
1  A 1.50 1.2
2  A 1.55 1.3
3  B 1.60 1.4
4  B 1.65 1.5
5  C 1.70 1.6
6  C 1.75 1.7
```

```
A1.501.2
A1.551.3
B1.601.4
B1.651.5
C1.701.6
C1.751.7
```

Saving Data

save.image()

```
write.table(x, file = "", append = FALSE, quote = TRUE, sep = " ",
            eol = "\n", na = "NA", dec = ".", row.names = TRUE,
            col.names = TRUE, qmethod = c("escape", "double"))
```

Generating Data

```
> 1:10-1
 [1] 0 1 2 3 4 5 6 7 8 9
> 1:(10-1)
[1] 1 2 3 4 5 6 7 8 9
```

```
> seq(1, 5, 0.5)
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
```

```
> seq(length=9, from=1, to=5)
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
```

Generating Data

```
> rep(1, 30)
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

> sequence(4:5)
[1] 1 2 3 4 1 2 3 4 5
> sequence(c(10,5))
 [1]  1  2  3  4  5  6  7  8  9 10  1  2  3  4  5

> gl(3, 5)
 [1] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3
Levels: 1 2 3
> gl(3, 5, length=30)
 [1] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3
Levels: 1 2 3
> gl(2, 6, label=c("Male", "Female"))
 [1] Male   Male   Male   Male   Male   Male
```

Generating Data

- Cartesian product

```
> expand.grid(h=c(60,80), w=c(100, 300), sex=c("Male", "Female"))
   h   w    sex
1 60 100   Male
2 80 100   Male
3 60 300   Male
4 80 300   Male
5 60 100 Female
6 80 100 Female
7 60 300 Female
8 80 300 Female
```

Generating Data

rfunc(n, p1, p2, ...)

law	function
Gaussian (normal)	`rnorm(n, mean=0, sd=1)`
exponential	`rexp(n, rate=1)`
gamma	`rgamma(n, shape, scale=1)`
Poisson	`rpois(n, lambda)`
Weibull	`rweibull(n, shape, scale=1)`
Cauchy	`rcauchy(n, location=0, scale=1)`
beta	`rbeta(n, shape1, shape2)`
'Student' (t)	`rt(n, df)`
Fisher–Snedecor (F)	`rf(n, df1, df2)`
Pearson (χ^2)	`rchisq(n, df)`
binomial	`rbinom(n, size, prob)`
multinomial	`rmultinom(n, size, prob)`
geometric	`rgeom(n, prob)`
hypergeometric	`rhyper(nn, m, n, k)`
logistic	`rlogis(n, location=0, scale=1)`
lognormal	`rlnorm(n, meanlog=0, sdlog=1)`
negative binomial	`rnbinom(n, size, prob)`
uniform	`runif(n, min=0, max=1)`
Wilcoxon's statistics	`rwilcox(nn, m, n)`, `rsignrank(nn, n)`

Of course Matrices

```
> matrix(1:6, 2, 3, byrow=TRUE)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
> x <- 1:15
> x
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
> dim(x)
NULL
> dim(x) <- c(5, 3)
> x
     [,1] [,2] [,3]
[1,]    1    6   11
[2,]    2    7   12
[3,]    3    8   13
[4,]    4    9   14
[5,]    5   10   15
```

Of course Matrices

```
> rbind(m1, m2) %*% cbind(m1, m2)
     [,1] [,2] [,3] [,4]
[1,]    2    2    4    4
[2,]    2    2    4    4
[3,]    4    4    8    8
[4,]    4    4    8    8
> cbind(m1, m2) %*% rbind(m1, m2)
     [,1] [,2]
[1,]   10   10
[2,]   10   10
```

```
> diag(m1)
[1] 1 1
> diag(rbind(m1, m2) %*% cbind(m1, m2))
[1] 2 2 8 8
```

Syntactic sugar

```
> x <- 3; y <- 2.5; z <- 1
> exp1 <- expression(x / (y + exp(z)))
> exp1
expression(x/(y + exp(z)))
> eval(exp1)
[1] 0.5749019


> D(exp1, "x")
1/(y + exp(z))
> D(exp1, "y")
-x/(y + exp(z))^2
> D(exp1, "z")
-x * exp(z)/(y + exp(z))^2
```

Graphics with R

- Device paradigm

- Window()
- Pdf()
- X11()

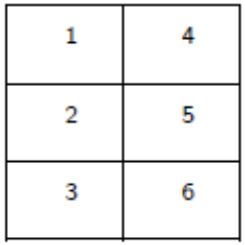
```
> dev.cur()
pdf
  4
```

and to change the active device:

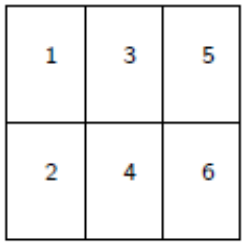
```
> dev.set(3)
X11
  3
```

Graphics with R

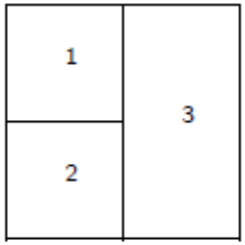
```
> layout(matrix(1:6, 3, 2))
> layout.show(6)
```




```
> layout(matrix(1:6, 2, 3))
> layout.show(6)
```




```
> m <- matrix(c(1:3, 3), 2, 2)
> layout(m)
> layout.show(3)
```



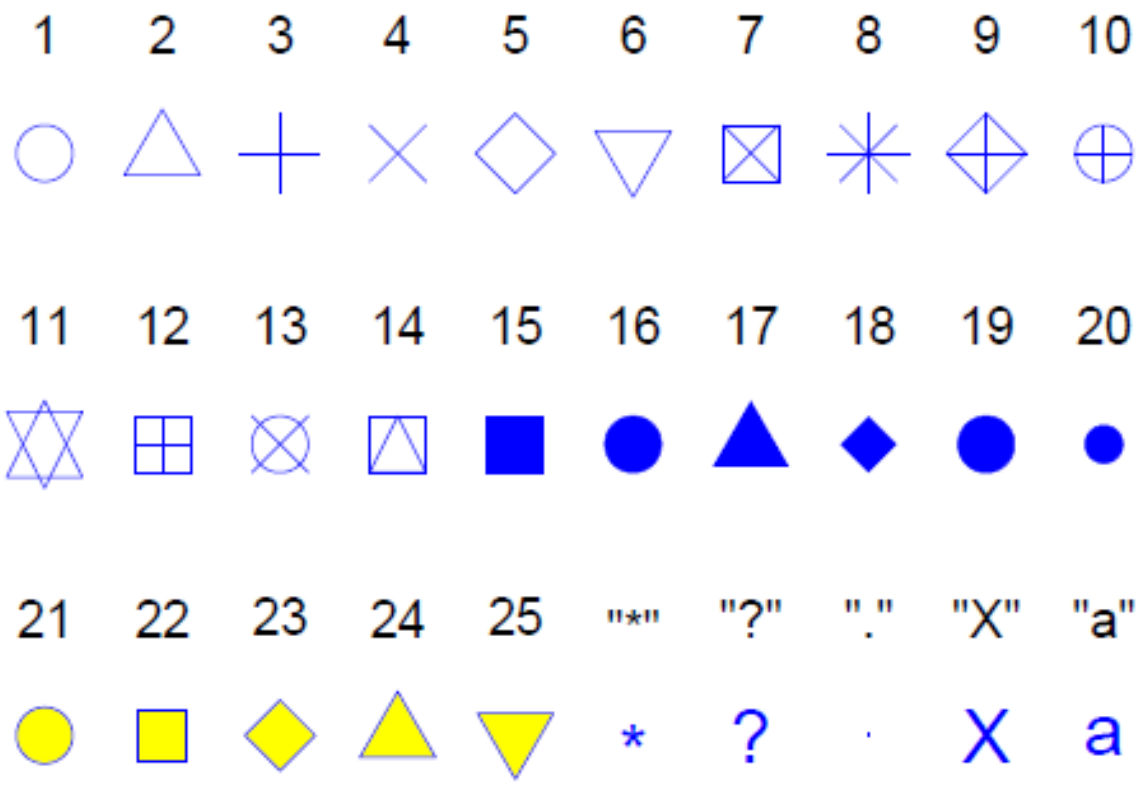
Graphics with R

- Legend for a graph

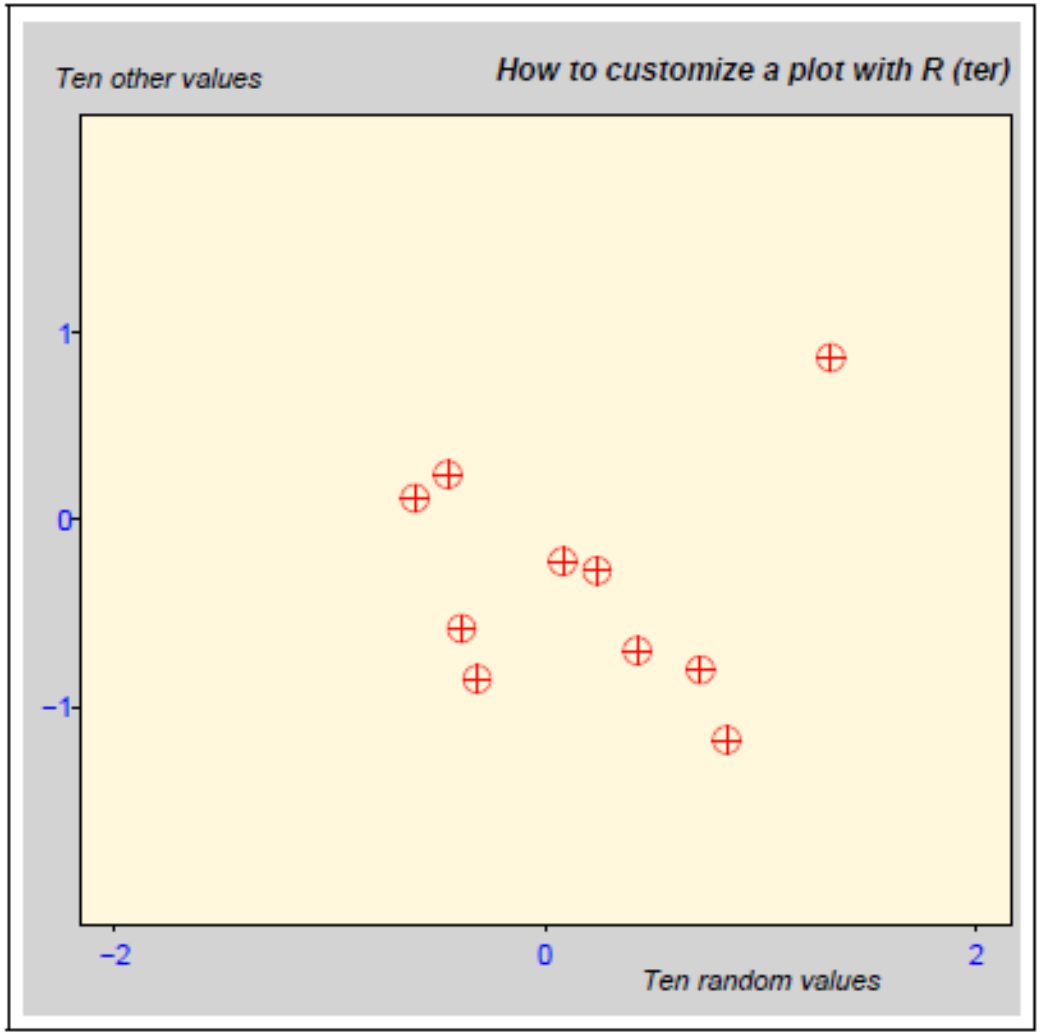
```
> text(x, y, expression(p == over(1, 1+e^-(beta*x+alpha))))
```

$$p = \frac{1}{1 + e^{-(\beta x + \alpha)}}$$

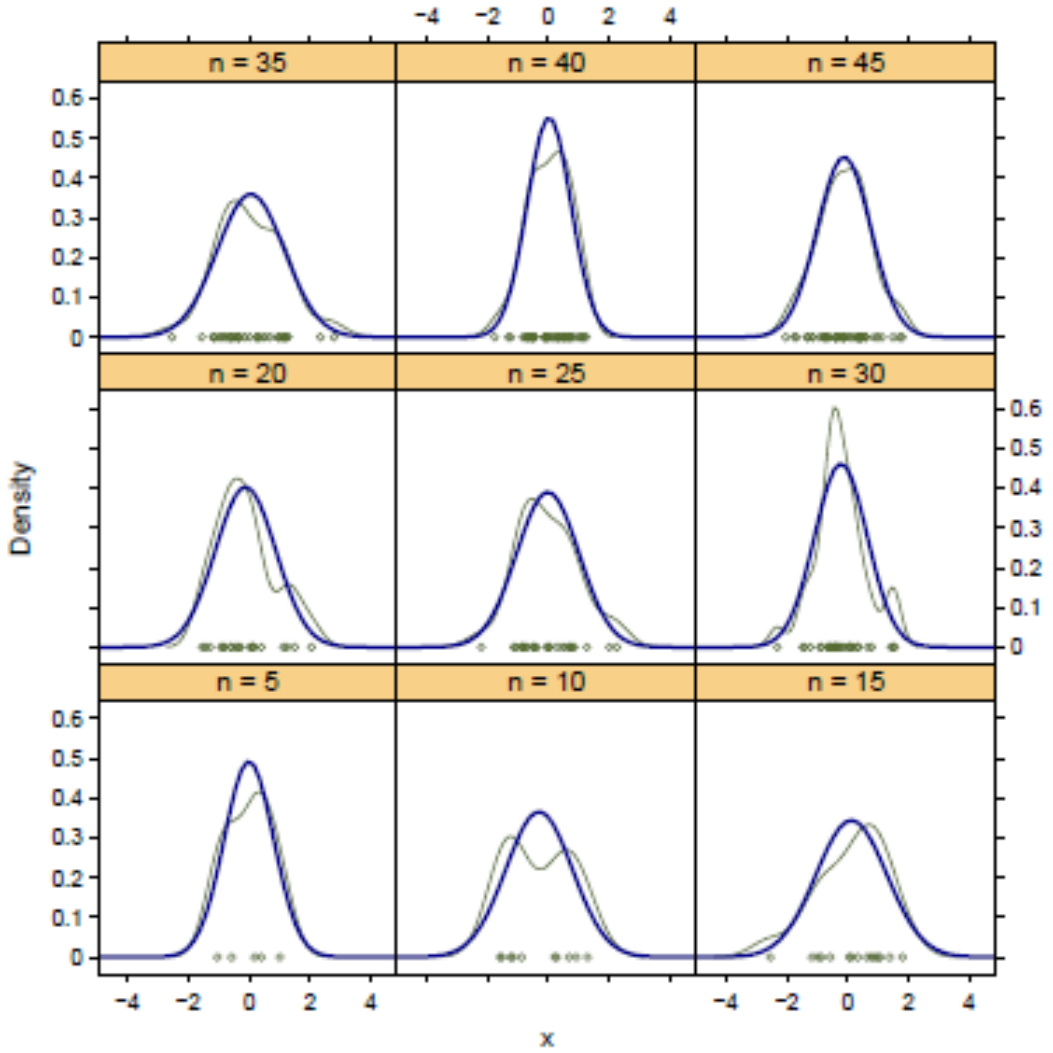
Graphics with R



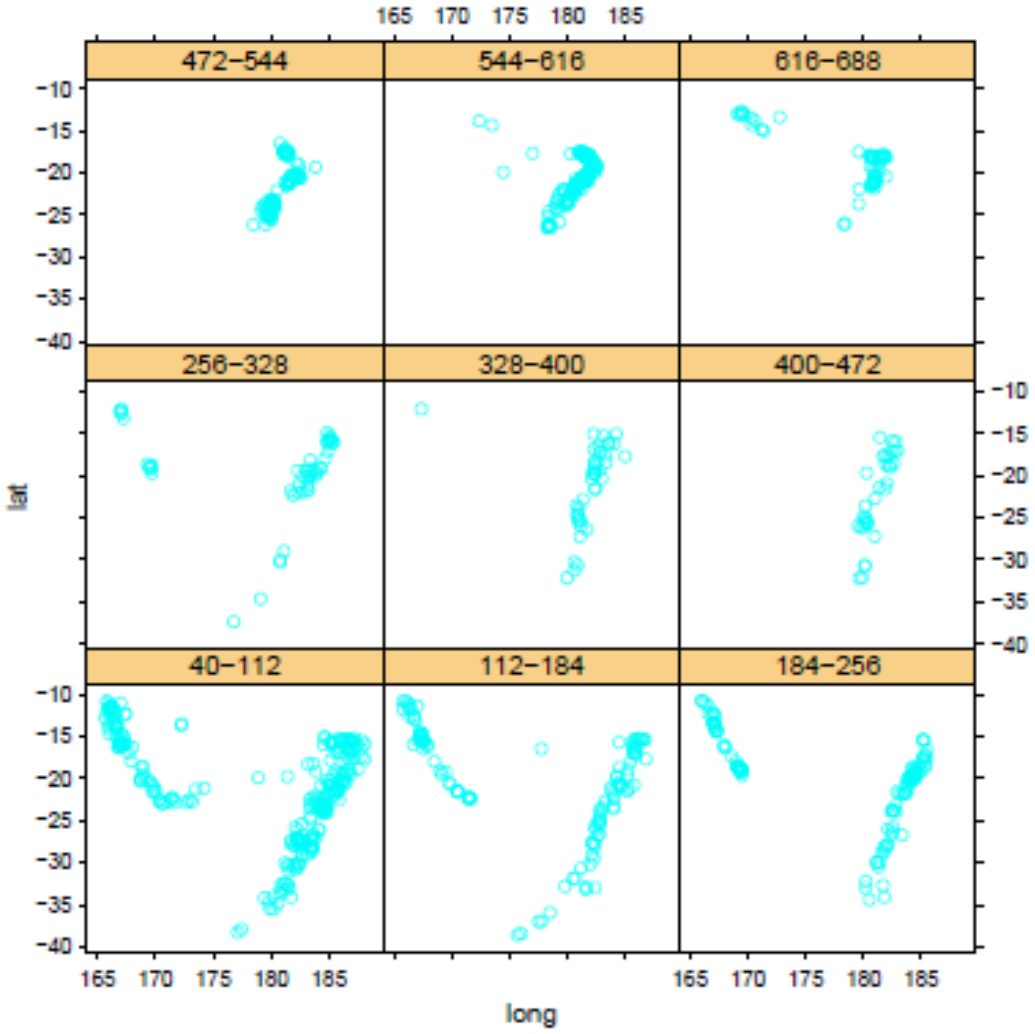
Graphics with R



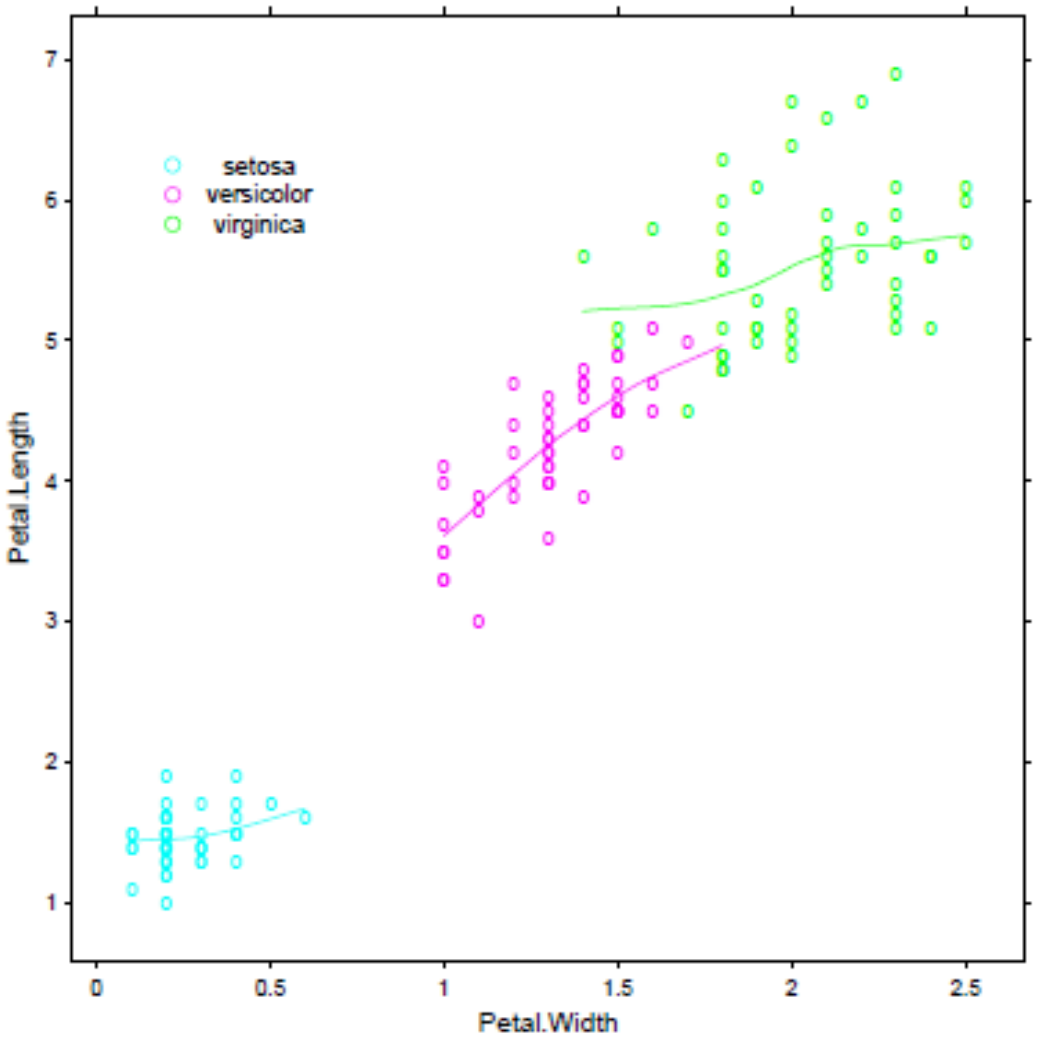
Graphics with R



Graphics with R



Graphics with R



Statistics with R

- > library(stats)

- Key operator "~"
- @model description operator

- y ~ model

Statistics with R

`a+b`	additive effects of `a` and of `b`
`X`	if `X` is a matrix, this specifies an additive effect of each of its columns, i.e. `X[,1]+X[,2]+...+X[,ncol(X)]`; some of the columns may be selected with numeric indices (e.g., `X[,2:4]`)
`a:b`	interactive effect between `a` and `b`
`a*b`	additive and interactive effects (identical to `a+b+a:b`)
`poly(a, n)`	polynomials of `a` up to degree n
`^n`	includes all interactions up to level n, i.e. `(a+b+c)^2` is identical to `a+b+c+a:b+a:c+b:c`
`b %in% a`	the effects of `b` are nested in `a` (identical to `a+a:b`, or `a/b`)
`-b`	removes the effect of `b`, for example: `(a+b+c)^2-a:b` is identical to `a+b+c+a:c+b:c`
`-1`	`y~x-1` is a regression through the origin (id. for `y~x+0` or `0+y~x`)
`1`	`y~1` fits a model with no effects (only the intercept)
`offset(...)`	adds an effect to the model without estimating any parameter (e.g., `offset(3*x)`)

Statistics with R

- Quiz
- y~x1+x2

$$y = \beta_1 x_1 + \beta_2 x_2 + \alpha$$

- y~I(x1+x2)

$$y = \beta(x_1 + x_2) + \alpha$$

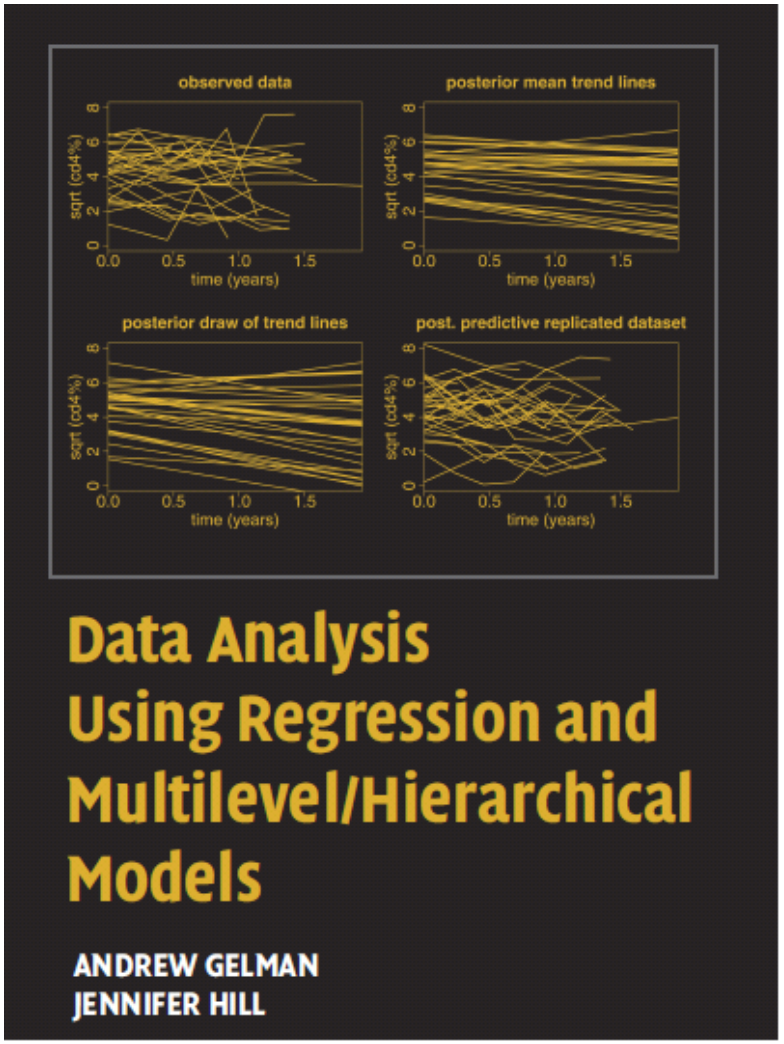
- y ~ poly(x, 2)

$$y = \beta_1 x + \beta_2 x^2 + \alpha$$

Statistics with R

Package	Description
base	base R functions
datasets	base R datasets
grDevices	graphics devices for base and grid graphics
graphics	base graphics
grid	grid graphics
methods	definition of methods and classes for R objects and programming tools
splines	regression spline functions and classes
stats	statistical functions
stats4	statistical functions using S4 classes
tcltk	functions to interface R with Tcl/Tk graphical user interface elements
tools	tools for package development and administration
utils	R utility functions

Multilevel Modeling with R



Multilevel Modeling with R

- Why multilevel modeling?

- Using all the data to perform inferences for groups with small sample size
- Predict an output for a new group
- Hierarchical models avoid overfitting effect of least squares regression
- Yields accurate measure of predictive uncertainty

Multilevel Modeling with R

```
fss = c(0,8,15,33,42,45,49,54,98,143,165,175,179,200)
# include the library
library(caTools)
# read training and scoring data
train <- read.csv("C:/Users/Spirinus/Desktop/Final
   Package/R/S_AUC_Train_1_7500.csv")
score <- read.csv("C:/Users/Spirinus/Desktop/Final
   Package/R/S_AUC_Train_Test_7501_15000.csv")
# data preparation
train[train$Target == - 1, "Target"] <- 0
train$RowID = NULL
```

Multilevel Modeling with R

```
# build the model
AUClogistic <- glm(Target ~ ., data=train[1:1000,fss+1],
   family=binomial(link="logit"))
# get predictions on a scoring dataset
test_scores <- predict(AUClogistic, type="response",
   score[1:1000,])
testY = score[1:1000,]$Target
# calculate AUC
colAUC(test_scores,testY)
```

Multilevel Modeling with R

lmer()

library(matrix)

Examples:

lmer(y ~ 1 + (1 | county))

lmer(y ~ x + (1 | county))

lmer(y ~ x + (1 + x | county))

Summary

- How R works
- Basic objects in R
- R graphical capabilities
- R for statistical analysis
- Multilevel modeling in R

More Information

- R for Beginners, Emmanuel Paradis, Institut des Sciences de l' Evolution Universite Montpellier II,

  F-34095 Montpellier cedex 05, France
- http://cran.r-project.org
- Data Analysis Using Regression and Multilevel Hierarchical Models, A. Gelman J.Hill

Acknowledgements

Thank you!