UCLA Statistical Consulting Center
R Bootcamp

Introduction to R

Irina Kukuyeva
ikukuyeva@stat.ucla.edu

September 20, 2010

Outline

1. Introduction
2. Preliminaries
3. Working with Vectors and Matrices
4. Data Sets in R
5. Overview of Plots in R
6. R Environment
7. Common Bugs and Fixes
8. R Resources
9. Appendix: Background Information for the Waves Data

1. Introduction
   - What is R?
   - What can you do with R?
   - What is the catch?
2. Preliminaries
3. Working with Vectors and Matrices
4. Data Sets in R
5. Overview of Plots in R
6. R Environment
7. Common Bugs and Fixes
8. R Resources
9. Appendix: Background Information for the Waves Data

What is R?

> *"R is a computer language that allows the user to program algorithms and use tools that have been programmed by others."*
>
> *Zuur et. al. (2009)*

What can you do with R?

You can ...

- do calculations
- perform statistical analysis (using available code)
- create powerful graphics
- write your own functions

What is the catch?

R has a steep learning curve:

- It requires programming...

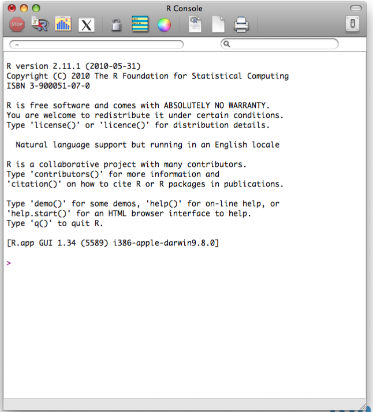
... but

- the programming used in R is very similar across methods
- a lot has already been done in terms of statistical tools

1. Introduction
2. Preliminaries
   - Software Installation
   - Creating Variables
3. Working with Vectors and Matrices
4. Data Sets in R
5. Overview of Plots in R
6. R Environment
7. Common Bugs and Fixes
8. R Resources
9. Appendix: Background Information for the Waves Data

Installing R

1. Go to `http://cran.r-project.org/` and select either:
   - *MacOS X*
   - *Windows* and *base*
2. Select to download the latest version: 2.11.1 (2010-05-31)
3. Install and Open. The R window should look like this:



```
R version 2.11.1 (2010-05-31)
Copyright (C) 2010 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.34 (5589) i386-apple-darwin9.8.0]

>
```

Creating Variables

- To use R as a calculator, type $2 + 5$ and hit ENTER. (Note how R prints the result.) Your output should look like this:

```
[1] 7
```

- To create variables in R, use either $- >$ or $=$:

```
# Approach 1
a=5
a
# Approach 2
a=5; a
# Approach 3
b<-5; b
```

1 Introduction

2 Preliminaries

3 Working with Vectors and Matrices
- Working with Vectors
  - Creating Vectors
  - Some Vector Functions
  - Sub-setting with Vectors
- Working with Matrices
  - Creating Matrices
  - Some Matrix Functions
  - Subsetting with Matrices
- From Vectors to Matrices
  - Creating Matrices from Vectors
- Exercise

4 Data Sets in R

5 Overview of Plots in R

6 R Environment

7 Common Bugs and Fixes

8 R Resources

9 Appendix: Background Information for the Waves Data

Creating Vectors

- Use the concatenation function `c()`:

```
d=c(3,4,7); d
```

```
[1] 3 4 7
```

- For vectors with equal spacing, use `seq()`:

```
e=seq(from=1, to=3, by=0.5); e
```

```
[1] 1.0 1.5 2.0 2.5 3.0
```

- For vectors of a given length, use `rep()`:

```
f=rep(NA, 6); f
```

```
[1] NA NA NA NA NA NA
```

Some Useful Vector Functions I

- To find the length of the vector, use `length()`:

```r
length(d)
```

```
[1] 3
```

- To find the maximum value of the vector, use `max()`:

```r
max(d)
```

```
[1] 7
```

Some Useful Vector Functions II

- To find the mean of the vector, use `mean()`:

```r
mean(d)
```

```
[1] 4.666667
```

Sub-setting with Vectors I

- To find out what is stored in a given element of the vector, use [ ]:

```
d[2]
```

```
[1] 4
```

- To see if any of the elements of a vector equal a certain number, use ==:

```
d==3
```

```
[1] TRUE FALSE FALSE
```

Sub-setting with Vectors II

- To see if any of the elements of a vector do not equal a certain number, use `!=`:

```
d!=3
```

```
[1] FALSE TRUE TRUE
```

- To delete elements of a vector, use `-` and/or `c()`:

```
e[-c(1,3)]; e
```

```
[1] 1.5 2.5 3.0
```

Sub-setting with Vectors III

- To obtain the observation number(s) of the vector when a condition is satisfied, use `which()`:

```
which(d==4)
```

```
[1] 2
```

Note: To store the result, type:

```
a=which(d==4); a
```

Sub-setting with Vectors IV

- To obtain the observation number(s) for the maximum value of the vector, use `which()`:or `which.max()`:

```
a=which(d==max(d)); a
b=which.max(d); b
```

```
[1] 3
```

Creating Matrices

- To create a matrix, use the `matrix()` function:

```
mat<-matrix(10:15, nrow=3, ncol=2,
byrow=F); mat
```

```
     [,1] [,2]
[1,]   10   13
[2,]   11   14
[3,]   12   15
```

Some Useful Matrix Functions I

- To find the transpose of a matrix, use `t()`:

```r
t(mat)
```

```
     [,1] [,2] [,3]
[1,]   10   11   12
[2,]   13   14   15
```

Some Useful Matrix Functions II

- To multiply two matrices, use `%*%`.
  Note: If you use `*` instead, you will be performing matrix multiplication element-wise.

```
mat%*%t(mat)
```

```
     [,1] [,2] [,3]
[1,]  269  292  315
[2,]  292  317  342
[3,]  315  342  369
```

Some Useful Matrix Functions III

- To find the dimensions of a matrix, use `dim()`:

```r
dim(mat)
```

```
[1] 3 2
```

- Alternatively, we can find the rows and columns of the matrix, by `nrow()` and `ncol()`:

```r
nrow(mat); ncol(mat)
```

```
[1] 3
[1] 2
```

Subsetting with Matrices I

- To see what is stored in the first element of the matrix, use `[ ]`:

```
mat[1,1]
```

```
[1] 10
```

- To see what is stored in the first row of the matrix:

```
mat[1,]
```

```
[1] 10 13
```

Subsetting with Matrices II

- To see what is stored in the second column of the matrix:

```r
mat[, 2]
```

```
[1] 13 14 15
```

- To extract elements 1 and 3 from the second column, use c() and [ ]:

```r
mat[c(1,3), 2]
```

```
[1] 13 15
```

Subsetting with Matrices III

- To extract *everything but* elements 1 and 3 from the second column, use `-c()` and `[ ]`:

```
mat[-c(1,3), 2]
```

```
[1] 14
```

- To extract the observation containing the maximum value, use `which.max()` and `[ ]`:

```
mat[which.max(mat)]
```

```
[1] 15
```

Creating Matrices from Vectors

- To stack two vectors, one below the other, use rbind():

```
mat1<-rbind(d,d); mat1
```

```
  [,1] [,2] [,3]
d    3    4    7
d    3    4    7
```

- To stack two vectors, one next to the other, use cbind():

```
mat2<-cbind(d,d); mat2
```

```
     d d
[1,] 3 3
[2,] 4 4
[3,] 7 7
```

Exercise I

Sum all the even rows of column 2 of the 10 × 10 matrix that contains the 1^{st} 100 numbers.

Hint:

Step 1: Create a 10 × 10 matrix (call it `ex1`) containing the elements 1 through 100, input elements by row.

Step 2: Create an index to store the even rows of a matrix.

Hint: Can you use `seq()`?

Step 3: Subset `ex1` appropriately, i.e. sum over the even rows of column 2 of the matrix.

▸ Solution here.

1. Introduction
2. Preliminaries
3. Working with Vectors and Matrices
4. Data Sets in R
   - Importing Data sets into R
     - Data from the Internet
     - Using Data Available in R
     - Importing Data from Your Computer
     - Importing Data from Your Computer
   - Working with Data sets in R
   - Exercise
5. Overview of Plots in R
6. R Environment
7. Common Bugs and Fixes
8. R Resources
9. Appendix: Background Information for the Waves Data

Data sets into R

Approach 1: Using Data Available in R

1. To use a data set available in one of the R packages, install that package (if needed).
2. Load the package into R, using the `library()` function.

```r
library(alr3)
```

3. Extract the data set you want from that package, using the `data()` function. In our case, the data set is called `UN2`.

```r
data(UN2)
```

Data sets into R

Approach 2a: Importing Data from Your Computer

For data sets that are not an R data set object (i.e. do not have a `.RData` extension):

1. Check what folder R is working with now:

```
getwd()
```

2. Tell R in what folder the data set is stored (if different from (1)). Suppose your data set is on your desktop:

```
setwd("~/Desktop")
```

3. Now use the `read.table()` command to read in the data, substituting the name of the file for the website.

Data sets into R

Approach 2b: Importing Data from Your Computer

For data sets that are an R data set object (i.e. have a .RData extension):

- Double click on the file

OR

- Load the data set into R from the console:

```
load("datasetName.RData")
```

Data sets into R

Approach 3a: Data from the Internet

When downloading data from the internet that are not an R data set object, use `read.table()`. In the arguments of the function:

- `header:` if TRUE, tells R to include variables names when importing
- `sep:` tells R how the entires in the data set are separated
  - `sep=",":` when entries are separated by COMMAS
  - `sep="\t":` when entries are separated by TAB
  - `sep=" ":` when entries are separated by SPACE

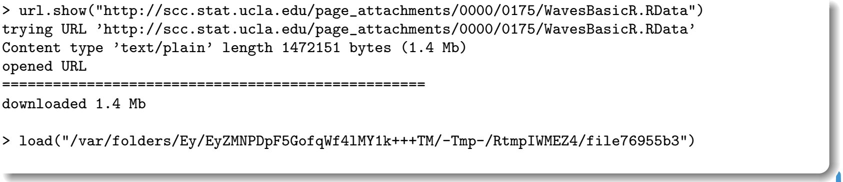
```
data<-read.table("http://www.stat.ucla.edu
   /~vlew/stat130a/datasets/twins.csv",
   header=TRUE, sep=",")
```

Data sets into R

Approach 3b: Data from the Internet

When downloading data from the internet that are an R data set object, use `url.show()`:

```
url.show("http://scc.stat.ucla.edu/page
    _attachments/0000/0175/WavesBasicR.RData")
```



```
> url.show("http://scc.stat.ucla.edu/page_attachments/0000/0175/WavesBasicR.RData")
trying URL 'http://scc.stat.ucla.edu/page_attachments/0000/0175/WavesBasicR.RData'
Content type 'text/plain' length 1472151 bytes (1.4 Mb)
opened URL
==================================================
downloaded 1.4 Mb

> load("/var/folders/Ey/EyZMNPDpF5GofqWf4lMY1k+++TM/-Tmp-/RtmpIWMEZ4/file76955b3")
```

Working with Data sets in R I

- To use the variable names when working with data, use `attach()`:

```r
attach(UN2)
```

- After the variable names have been "attached", to see the variable names, use `names()`:

```r
names(UN2)
```

- To see the descriptions of the variables, use `?`:

```r
?UN2
```

Working with Data sets in R II

- To stop referring to variable names directly, use `detach()` (but not now):

```r
detach(UN2)
```

- To get the mean of all the variables in the data set, use `mean()`:

```r
mean(UN2)
```

```
   logPPgdp  logFertility        Purban      Locality
  10.993094      1.018016     55.538860            NA
Warning message:
In mean.default(X[[4L]], ...) :
  argument is not numeric or logical: returning NA
```

Working with Data sets in R III

- To get the variance-covariance matrix of all the (numerical) variables in the data set, use `var()`:

```r
var(UN2[, -4])
```

```
              logPPgdp logFertility     Purban
logPPgdp     5.6408387   -0.8647205  44.555873
logFertility -0.8647205    0.2887060  -7.630714
Purban       44.5558730   -7.6307145 579.197701
```

Exercise II

Using the data set `WavesBasicR.RData`,[1] find out how many observations are greater than the mean wave height.

Hint:

Step 1: Select the third variable for the analysis.

Step 2: Calculate the mean for the variable.

Step 3: See `which` observations are greater than the mean (save the output as `out`).

Step 4: Calculate the `length` of `out`.

▸ Solution here.

[1] http://scc.stat.ucla.edu/page_attachments/0000/0175/WavesBasicR.RData

1 Introduction

2 Preliminaries

3 Working with Vectors and Matrices

4 Data Sets in R

5 Overview of Plots in R
- Creating Plots
- Saving Plots as a PDF
- Exercise

6 R Environment

7 Common Bugs and Fixes

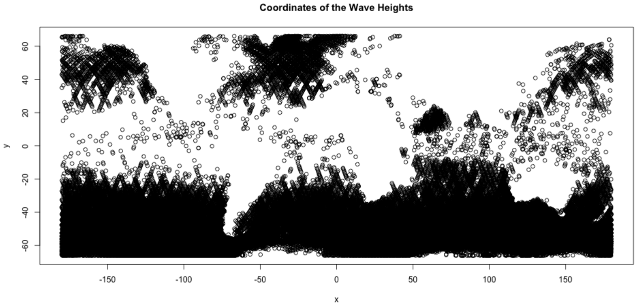
8 R Resources

9 Appendix: Background Information for the Waves Data

Creating Plots in R

- To make a plot in R, you can use `plot()`:

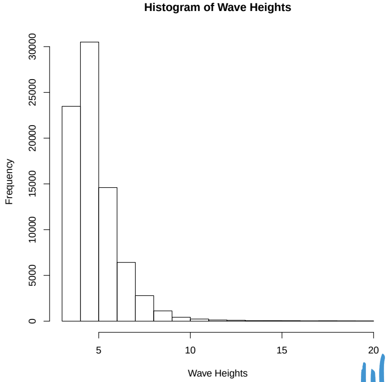
```r
attach(data)
plot(x, y, main="Coordinates of the Wave Heights")
```



Creating Plots in R

- To make a histogram in R, you can use hist():

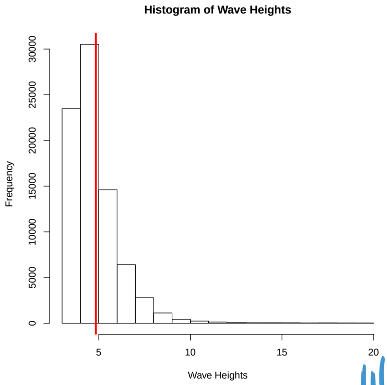
```r
hist(wave_height, xlab="Wave Heights", main="Histogram of Wave Heights")
```



Creating Plots in R

- To add information to the histogram you can use `abline()`:

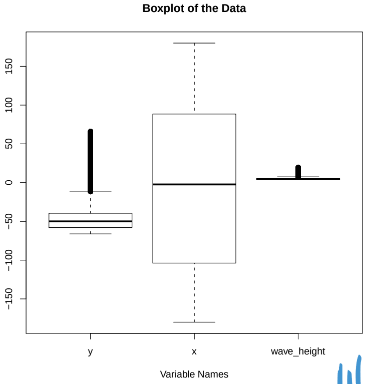
```r
hist(wave_height, xlab="Wave Heights", main="Histogram of Wave Heights")
abline(v=mean(wave_height), col="red", lwd=3)
```



Creating Plots in R

- To make a boxplot in R, you can use `boxplot()`:

```
boxplot(data, xlab="Variable Names", main="Boxplot of the Data")
```

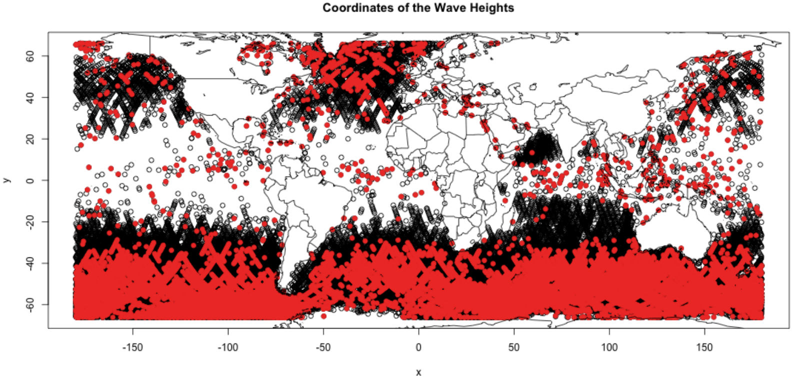


Creating Plots in R I

- To add/highlight points for an existing plot, use `points()`:

```
ind<-which(wave_height>6)
plot(x, y, main="Coordinates of the Wave Heights")
points(y[ind]~x[ind], col="red", pch=19)
library(maps)
map("world", add=TRUE)
```

Creating Plots in R II



Saving Plots as a PDF

Note: The files will be saved in the folder specified with `setwd()`.
To save a plot in R as a PDF, you can use `pdf()`:

```
pdf(HistWaves.pdf", width=6, height=6)
hist(wave_height, xlab="Wave Heights", main="
    Histogram of Wave Heights")
abline(v=mean(wave_height), col="red", lwd=3)
dev.off()
```

Exercise III

Using the data set `WavesBasicR.RData`,[2] find out what hemisphere has the largest waves.

Hint:

Step 1: Set a threshold for "large".

Step 2: Determine `which` observations are greater than the threshold.

Step 3: Highlight these observations in a plot.

▸ Solution here.

[2] http://scc.stat.ucla.edu/page_attachments/0000/0175/WavesBasicR.RData

1. Introduction
2. Preliminaries
3. Working with Vectors and Matrices
4. Data Sets in R
5. Overview of Plots in R
6. R Environment
   - Working with R Objects
   - Saving and Loading R Objects
   - Exporting R Objects to LaTeX
   - Exporting R Objects to Other Formats
   - Saving R Commands
7. Common Bugs and Fixes
8. R Resources
9. Appendix: Background Information for the Waves Data

Working with R Objects I

- To see the names of the objects available to be saved (in your current workspace), use `ls()`.

```r
ls()
```

```
[1] "UN2" "a" "b" "d" "data" "e" "f" "h" "mat1" "mat2"
```

- To remove objects from your workspace, use `rm()`.

```r
rm(d)
ls()
```

```
[1] "UN2" "a" "b" "data" "e" "f" "h" "mat1" "mat2"
```

Working with R Objects II

- To remove *all* the objects from your workspace, type:

```
rm(list=ls())
ls()
```

```
character(0)
```

Saving and Loading R Objects

- To save (to the current directory) all the objects in the workspace, use `save.image()`.

```r
save.image("basicR.RData")
```

- To load (from the current directory), use `load()`.

```r
load("basicR.RData")
```

Exporting R Objects to LaTeX I

- To export certain R objects to be used in LaTeX, use `xtable()`.

```
library(xtable)
xtable(summary(UN2))
```

Exporting R Objects to LaTeX II

```
% latex table generated in R 2.9.0 by xtable 1.5-5 package
% Fri Sep 18 19:58:39 2009
\begin{table}[ht]
\begin{center}
\begin{tabular}{rllll}
  \hline
 &    logPPgdp &  logFertility &     Purban &        Locality \\
  \hline
1 & Min.   : 6.492   & Min.   :0.0000   & Min.   :  6.00   & Afghanistan:  1   \\
  2 & 1st Qu.: 8.867   & 1st Qu.:0.6366   & 1st Qu.: 35.00   & Albania    :  1   \\
  3 & Median :10.920   & Median :0.9783   & Median : 57.00   & Algeria    :  1   \\
  4 & Mean   :10.993   & Mean   :1.0180   & Mean   : 55.54   & Angola     :  1   \\
  5 & 3rd Qu.:12.938   & 3rd Qu.:1.4493   & 3rd Qu.: 75.00   & Argentina  :  1   \\
  6 & Max.   :15.444   & Max.   :2.0794   & Max.   :100.00   & Armenia    :  1   \\
  7 &  &  &  & (Other)    :187   \\
   \hline
\end{tabular}
\end{center}
\end{table}
```

Exporting R Objects to LaTeX III

	logPPgdp	logFertility	Purban	Locality
1	Min. : 6.492	Min. :0.0000	Min. : 6.00	Afghanistan: 1
2	1st Qu.: 8.867	1st Qu.:0.6366	1st Qu.: 35.00	Albania : 1
3	Median :10.920	Median :0.9783	Median : 57.00	Algeria : 1
4	Mean :10.993	Mean :1.0180	Mean : 55.54	Angola : 1
5	3rd Qu.:12.938	3rd Qu.:1.4493	3rd Qu.: 75.00	Argentina : 1
6	Max. :15.444	Max. :2.0794	Max. :100.00	Armenia : 1
7				(Other) :187

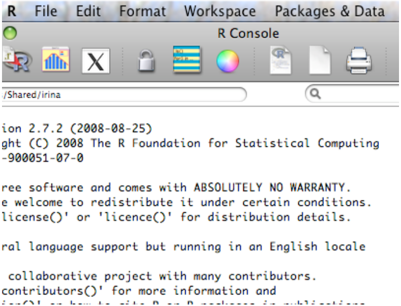
Exporting R Objects to Other Formats

- To save (to the current directory) certain objects in the workspace to be used in Excel, use `write.csv()`.

```r
write.csv(mat, "mat.csv")
```

Saving R Commands I

- To see all of the commands you typed in an R session, click on the Yellow and Green Tablet



```
ion 2.7.2 (2008-08-25)
ght (C) 2008 The R Foundation for Statistical Computing
-900051-07-0

ree software and comes with ABSOLUTELY NO WARRANTY.
e welcome to redistribute it under certain conditions.
license()' or 'licence()' for distribution details.

ral language support but running in an English locale

 collaborative project with many contributors.
contributors()' for more information and
```

Saving R Commands II

- To save all of the commands you typed in an R session, use:

```
savehistory(file="history.log")
```

- Alternatively, use a `.r` file to store your commands.
  1. Go to: File -> New Document
  2. Type your commands
  3. Save the file as "code.r"
  4. Go back to the R Console
  5. To run all the commands, use:

```
source("code.r")
```

1. Introduction
2. Preliminaries
3. Working with Vectors and Matrices
4. Data Sets in R
5. Overview of Plots in R
6. R Environment
7. Common Bugs and Fixes
    - Syntax Error
    - Trailing +
    - Error When Performing Operations
8. R Resources
9. Appendix: Background Information for the Waves Data

Error: syntax error

Possible causes:

- Incorrect spelling (of the function, variable, etc.)
- Including a "+" when copying code from the Console
- Having an extra parenthesis at the end of a function
- Having an extra bracket when subsetting

Trailing +

Possible causes:

- Not closing a function call with a parenthesis
- Not closing brackets when subsetting
- Not closing a function you wrote with a squiggly brace

Error in ... : requires numeric matrix/vector arguments

Possible causes:

1. Objects are data frames, not matrices
2. Elements of the vectors are characters

Possible solutions:

1. Coerce (a copy of) the data set to be a matrix, with the `as.matrix()` command
2. Coerce (a copy of) the vector to have numeric entries, with the `as.numeric()` command

1 Introduction

2 Preliminaries

3 Working with Vectors and Matrices

4 Data Sets in R

5 Overview of Plots in R

6 R Environment

7 Common Bugs and Fixes

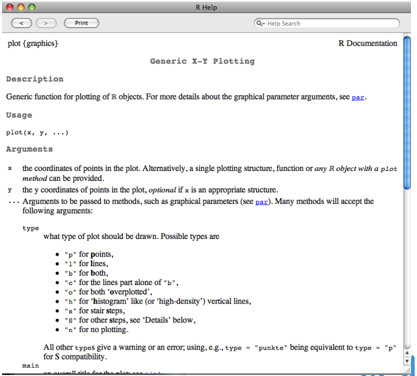
8 R Resources
- Getting Help in R
- Useful Links for R

9 Appendix: Background Information for the Waves Data

R Help

For help with any function in R , put a question mark before the function name to determine what arguments to use, examples and background information.

```
?plot
```



plot {graphics} R Documentation

Generic X-Y Plotting

Description

Generic function for plotting of R objects. For more details about the graphical parameter arguments, see par.

Usage

```
plot(x, y, ...)
```

Arguments

x the coordinates of points in the plot. Alternatively, a single plotting structure, function or *any R object with a* `plot` *method* can be provided.

y the y coordinates of points in the plot, *optional* if `x` is an appropriate structure.

... Arguments to be passed to methods, such as graphical parameters (see par). Many methods will accept the following arguments:

`type`
what type of plot should be drawn. Possible types are

- `"p"` for **p**oints,
- `"l"` for **l**ines,
- `"b"` for **b**oth,
- `"c"` for the lines part alone of `"b"`,
- `"o"` for both '**o**verplotted',
- `"h"` for '**h**istogram' like (or 'high-density') vertical lines,
- `"s"` for stair **s**teps,
- `"S"` for other **s**teps, see 'Details' below,
- `"n"` for no plotting.

All other `type`s give a warning or an error; using, e.g., `type = "punkte"` being equivalent to `type = "p"` for S compatibility.

`main`

Online Resources for R I

- Download R: `http://cran.stat.ucla.edu/`
- Search Engine for R: `http://rseek.org`
- R Reference Card:
  `http://cran.r-project.org/doc/contrib/Short-refcard.pdf`
- R Graph Gallery: `http://addictedtor.free.fr/graphiques/`
- R Graphics Gallery:
  `http://research.stowers-institute.org/efg/R/`
- Statistics with R: `http://zoonek2.free.fr/UNIX/48`
  `R/all.html`
- Springer (useR! series):
  `http://www.springerlink.com/home/main.mpx`

Online Resources for R II

- UCLA Statistics Information Portal: `http://info.stat.ucla.edu/grad/`
- UCLA Statistical Consulting Center: `http://scc.stat.ucla.edu`

References

I. Kukuyeva.
Basic R.
UCLA Statistical Consulting Center, Sept. 28, 2010.
http://scc.stat.ucla.edu/mini-courses/materials-from-past-mini-courses/fall-2009-mini-course-materials/

1 Introduction

2 Preliminaries

3 Working with Vectors and Matrices

4 Data Sets in R

5 Overview of Plots in R

6 R Environment

7 Common Bugs and Fixes

8 R Resources

9 Appendix: Background Information for the Waves Data

Appendix: Motivation for the Waves Data

Motivation

In 1992-2003 ships sunk at sea (due to wind and waves) contributed to 30.9 percent of all losses [a].

[a] Extreme Waves, C. Smith, p. 4

Background

Sea surface topography is measured along a track:

- frequency of 13.6GHz (K_u band) and 5.3 GHz (C band)
- repeat cycle of $\sim$ 10 days
- footprint of 6 km

Appendix: Overview of the Waves Data

Variable of interest: Significant wave heights (in meters) = the average of the top $\frac{1}{3}$ of the waves in the footprint.

Time interval: April 3 - May 3, 2008

Number of observations: 80,072 [3].

[3] 80,072 are a subset of the 5,082,121 ocean observations for the time period. First, every 15th observation was chosen. Then observations with a SWH greater than 3.5 meters were chosen.