Data analysis using R

-Amresh K. Singh,

Dept of Computer Sc.

MGSU Bikaner

Paper Code:MCS-305(a)

Main topics to be covered

- What is R
- How to obtain and install R
- How to read and export data
- How to do basic statistical analyses
- Econometric packages in R

What is R

- Software for Statistical Data Analysis
- Based on S
- Programming Environment
- Interpreted Language
- Data Storage, Analysis, Graphing
- Free and Open Source Software

Obtaining R

- Current Version: R-2.0.0
- Comprehensive R Archive Network:

  http://cran.r-project.org
- Binary source codes
- Windows executables
- Compiled RPMs for Linux
- Can be obtained on a CD

Installing R

- Binary (Windows/Linux): One step process
  - exe, rpm (Red Hat/Mandrake), apt-get (Debian)
- Linux, from sources:

```
$ tar –zxvf “filename.tar.gz”
$ cd filename
$ ./configure
$ make
$ make check
$ make install
```

Starting R

Windows, Double-click on Desktop Icon

$ R

Linux, type R at command prompt

Strengths and Weaknesses

- Strengths
  - Free and Open Source
  - Strong User Community
  - Highly extensible, flexible
  - Implementation of high end statistical methods
  - Flexible graphics and intelligent defaults
- Weakness
  - Steep learning curve
  - Slow for large datasets

Basics

- Highly Functional
  - Everything done through functions
  - Strict named arguments
  - Abbreviations in arguments OK (e.g. T for TRUE)
- Object Oriented
  - Everything is an object
  - “<-” is an assignment operator
  - “X <- 5”: X GETS the value 5

Getting Help in R

- From Documentation:
  - **?WhatIWantToKnow**
  - **help(“WhatIWantToKnow”)**
  - **help.search(“WhatIWantToKnow”)**
  - **help.start()**
  - **getAnywhere(“WhatIWantToKnow”)**
  - **example(“WhatIWantToKnow”)**
- Documents: “Introduction to R”
- Active Mailing List
  - Archives
  - Directly Asking Questions on the List

Data Structures

- Supports virtually any type of data
- Numbers, characters, logicals (TRUE/ FALSE)
- Arrays of virtually unlimited sizes
- Simplest: Vectors and Matrices
- Lists: Can Contain mixed type variables
- Data Frame: Rectangular Data Set

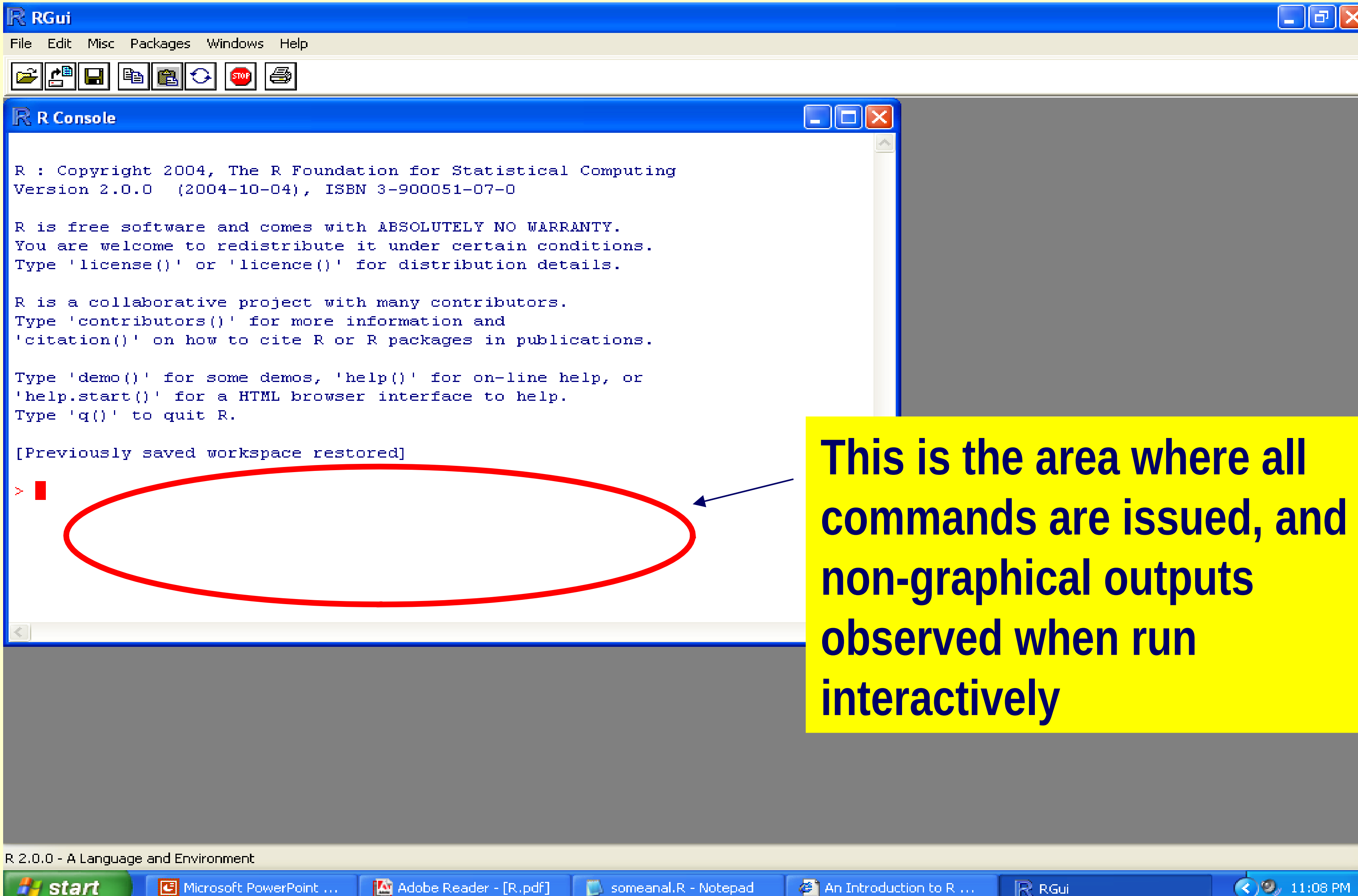
Data Structure in R

	Linear	Rectangular
All Same Type	VECTORS	MATRIX*
Mixed	LIST	DATA FRAME

Running R

- Directly in the Windowing System (Console)
- Using Editors
  - Notepad, WinEdt, Tinn-R: Windows
  - Xemacs, ESS (Emacs speaks Statistics)
- On the Editor:
  - **source(“filename.R”)**
  - Outputs can be diverted by using
    - **sink(“filename.Rout”)**

R Working Area



This is the area where all commands are issued, and non-graphical outputs observed when run interactively

In an R Session…

- First, read data from other sources
- Use packages, libraries, and functions
- Write functions wherever necessary
- Conduct Statistical Data Analysis
- Save outputs to files, write tables
- Save R workspace if necessary (exit prompt)

Specific Tasks

- To see which directories and data are loaded, type: **search()**
- To see which objects are stored, type: **ls()**
- To include a dataset in the searchpath for analysis, type: **attach(NameOfTheDataset, expression)**
- To detach a dataset from the searchpath after analysis, type: **detach(NameOfTheDataset)**

Reading data into R

- R not well suited for data preprocessing
- Preprocess data elsewhere (SPSS, etc…)
- Easiest form of data to input: text file
- Spreadsheet like data:
  - Small/medium size: use `read.table()`
  - Large data: use `scan()`
- Read from other systems:
  - Use the library “foreign”: `library(foreign)`
  - Can import from SAS, SPSS, Epi Info
  - Can export to STATA

Reading Data: summary

- Directly using a vector e.g.: **x <- c(1,2,3…)**
- Using **scan** and **read.table** function
- Using **matrix** function to read data matrices
- Using **data.frame** to read mixed data
- library(foreign) for data from other programs

Accessing Variables

- edit(<mydataobject>)
- Subscripts essential tools
  - x[1] identifies first element in vector x
  - y[1,] identifies first row in matrix y
  - y[,1] identifies first column in matrix y
- $ sign for lists and data frames
  - myframe$age gets age variable of myframe
  - attach(dataframe) -> extract by variable name

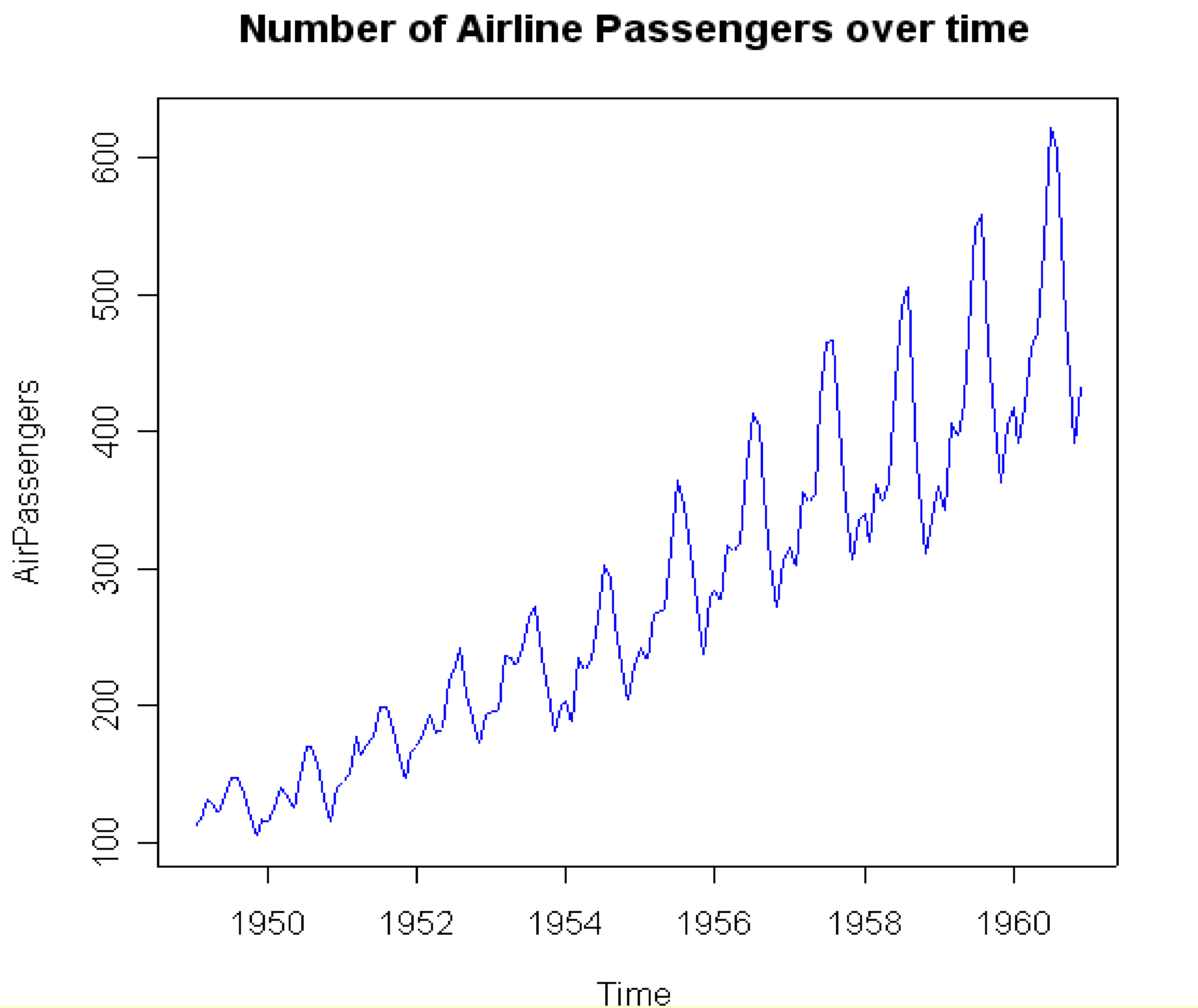
Subset Data

- Using subset function
  - subset() will subset the dataframe
- Subscripting from data frames
  - myframe[,1] gives first column of myframe
- Specifying a vector
  - myframe[1:5] gives first 5 rows of data
- Using logical expressions
  - myframe[myframe[,1], < 5,] gets all rows of the first column that contain values less than 5

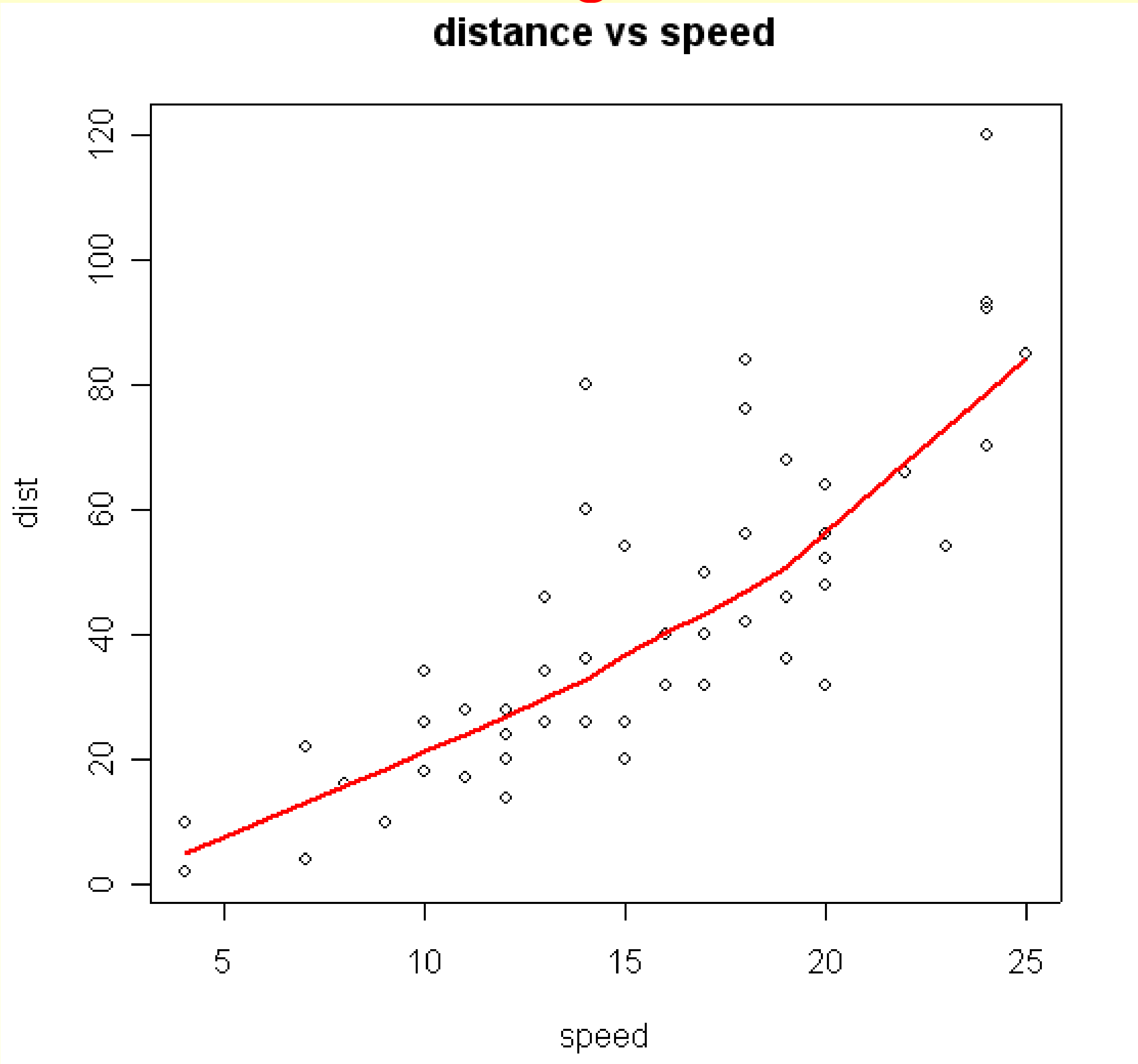
Graphics

- Plot an object, like: plot(num.vec)
  - here plots against index numbers
- Plot sends to graphic devices
  - can specify which graphic device you want
    - postscript, gif, jpeg, etc…
    - you can turn them on and off, like: dev.off()
- Two types of plotting
  - high level: graphs drawn with one call
  - Low Level: add additional information to existing graph

High Level: generated with plot()



Low Level: Scattergram with Lowess



Programming in R

- Functions & Operators typically work on entire vectors
- Expressions surrounded by {}
- Codes separated by newlines, “;” not necessary
- You can write your own functions and use them

Statistical Functions in R

- Descriptive Statistics
- Statistical Modeling
  - Regressions: Linear and Logistic
  - Probit, Tobit Models
  - Time Series
- Multivariate Functions
- Inbuilt Packages, contributed packages

Descriptive Statistics

- Has functions for all common statistics
- summary() gives lowest, mean, median, first, third quartiles, highest for numeric variables
- stem() gives stem-leaf plots
- table() gives tabulation of categorical variables

Statistical Modeling

- Over 400 functions
  - lm, glm, aov, ts
- Numerous libraries & packages
  - survival, coxph, tree (recursive trees), nls, …
- Distinction between factors and regressors
  - factors: categorical, regressors: continuous
  - you must specify factors unless they are obvious to R
  - dummy variables for factors created automatically
- Use of data.frame makes life easy

How to model

- Specify your model like this:
  - $y \sim x_i+c_i$, where
  - y = outcome variable, x_i = main explanatory variables, c_i = covariates, + = add terms
  - Operators have special meanings
    - + = add terms, : = interactions, / = nesting, so on…
- Modeling -- object oriented
  - each modeling procedure produces objects
  - classes and functions for each object

Synopsis of Operators

Operator	Usually means	In Formula means
+ or -	add or subtract	add or remove terms
*	multiplication	main effect and interactions
/	division	main effect and nesting
:	sequence	interaction only
^	exponentiation	limiting interaction depths
%in%	no specific	nesting only

Modeling Example: Regression

carReg <- lm(speed~dist, data=cars)

carReg = becomes an object

to get summary of this regression, we type

summary(carReg)

to get only coefficients, we type

coef(carReg), or carReg$coef

don’t want intercept? add 0, so

carReg <- lm(speed~0+dist, data=cars)

Multivariate Techniques

- Several Libraries available
  - mva, hmisc, glm,
  - MASS: discriminant analysis and multidim scaling
- Econometrics packages
  - dse (multivariate time series, state-space models), ineq: for measuring inequality, poverty estimation, its: for irregular time series, sem: structural equation modeling, and so on…

[http://www.mayin.org/ajayshah/]

Summarizing…

- Effective data handling and storage
- large, coherent set of tools for data analysis
- Good graphical facilities and display
  - on screen
  - on paper
- well-developed, simple, effective programming

References

R home page

http://www.r-project.org

R discussion group

http://www.stat.math.ethz.ch/mailman/listinfo/r-help

Disclaimer

The content may be used by the students (only for educational purpose, not for commercial purpose.)