R Programming Language

University of Kelaniya
Faculty of Medicine

Dileepa Ediriweera

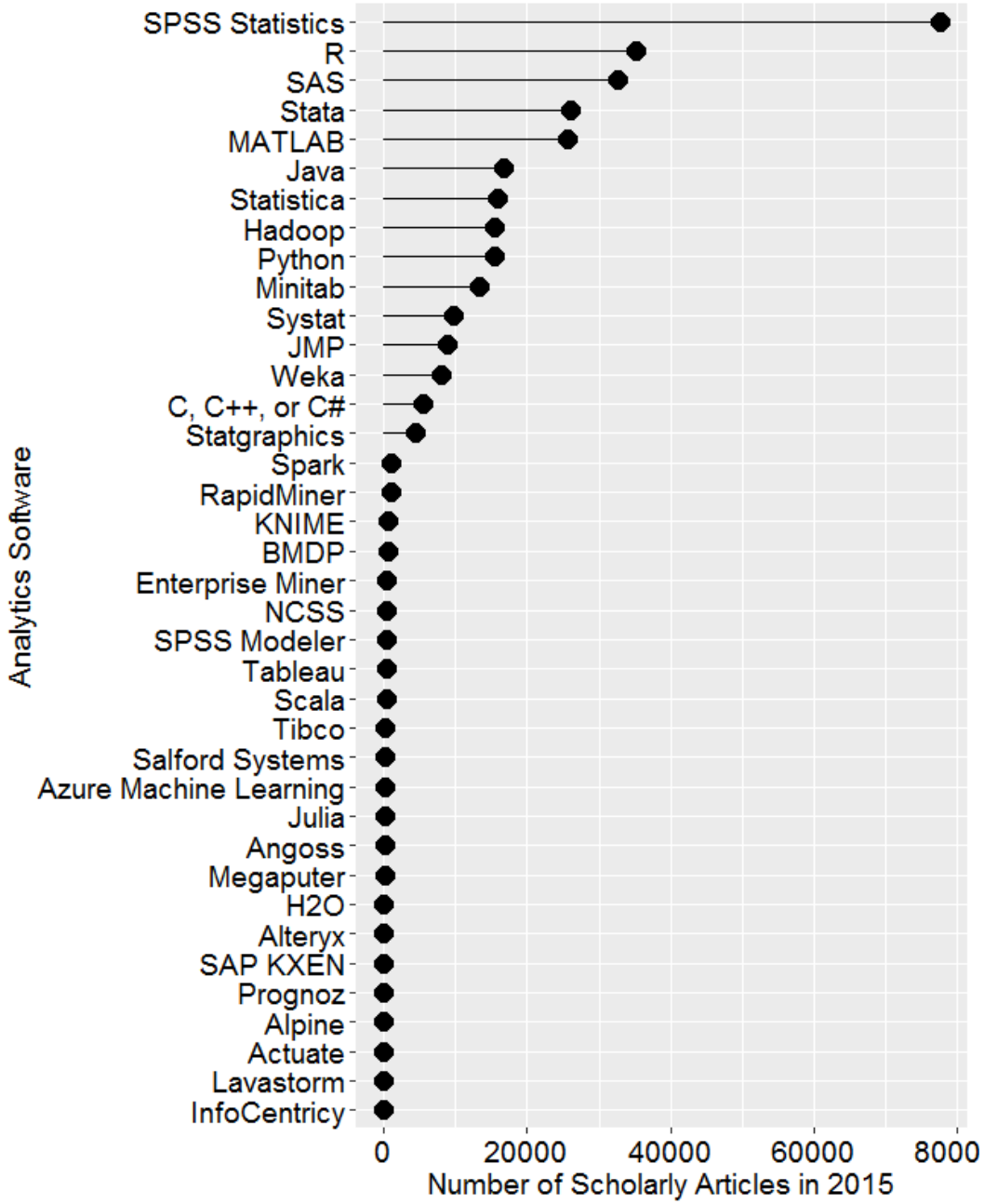
MBBS, DCSD, MSc(Biostat), MSc(BMI), PhD

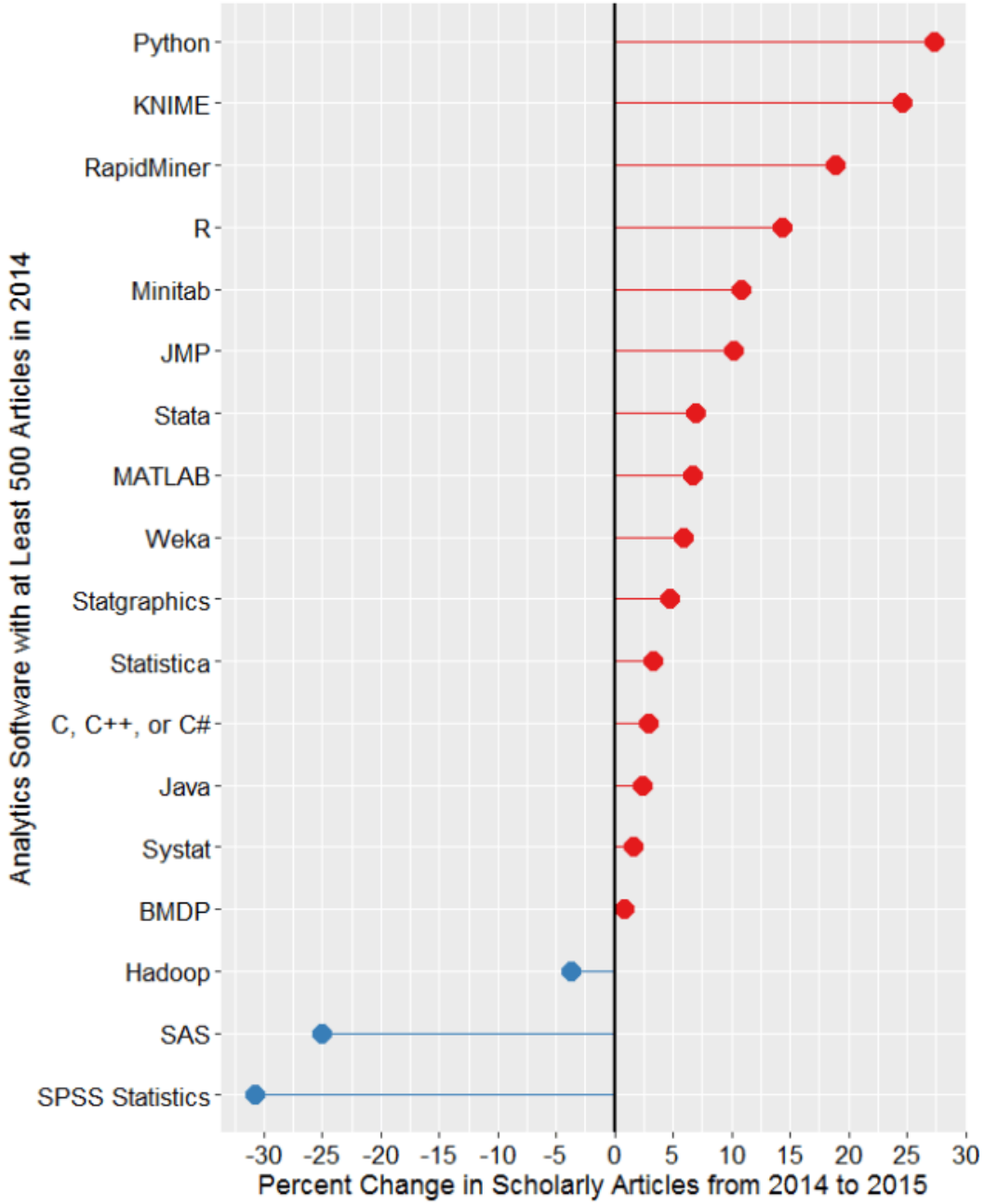
Objectives of the workshop

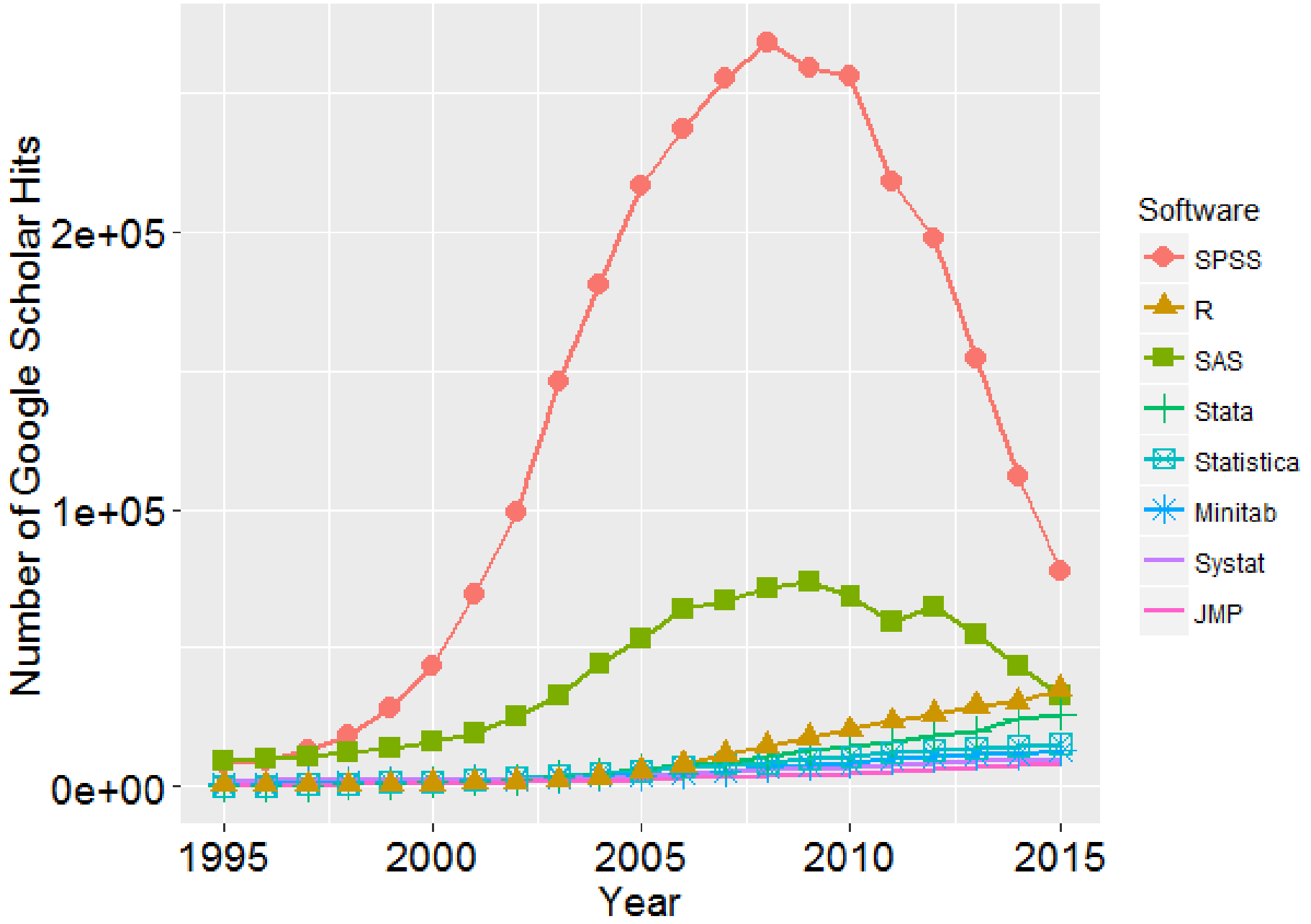
- Introduction to R programming language
- Import data to Rstudio
- Summary measures
  - Mean, median, standard deviation, Centiles, N, min, max
- Visualize data
  - Bar plot, Histogram, Box plot, Scatter plot
- Hypothesis testing
  - T test
  - Pearson’s Chisqaure test

R Programming Language

- R is a **free software** environment for statistical computing and graphics (Ref: https://www.r-project.org/)
- It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS.
- R was created by **Ross Ihaka** and **Robert Gentleman** at the University of Auckland, New Zealand
- R is an implementation of the **S programming language**
  - by John Chambers while at Bell Labs
  - much of the code written for S runs unaltered



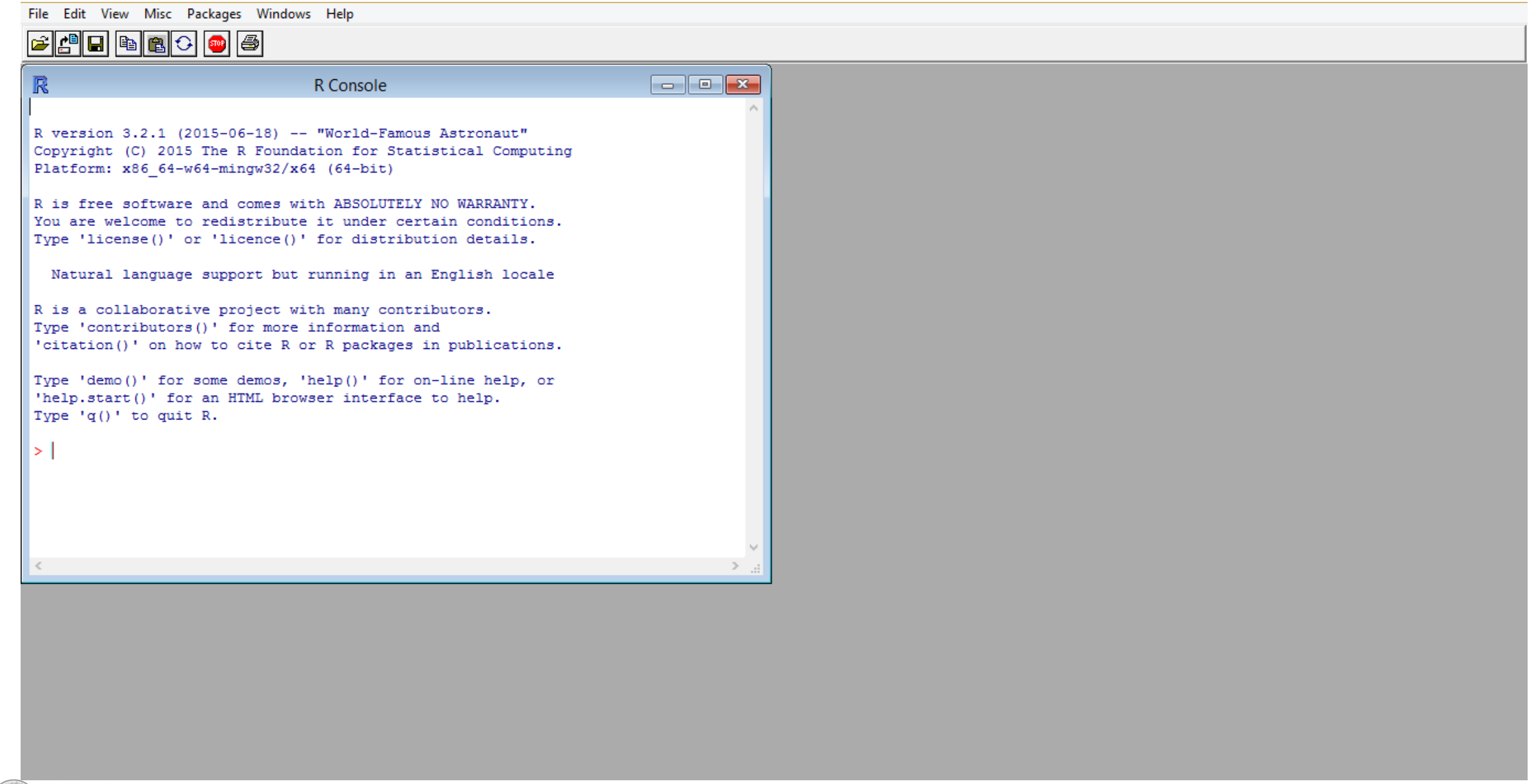




Tutorials

- P. Kuhnert & B. Venables, An Introduction to R: Software for Statistical Modeling & Computing
- J.H. Maindonald, Using R for Data Analysis and Graphics
- B. Muenchen, R for SAS and SPSS Users
- W.J. Owen, The R Guide
- D. Rossiter, Introduction to the R Project for Statistical Computing for Use at the ITC
- W.N. Venebles & D. M. Smith, An Introduction to R

Data analysis with R…


File Edit View Misc Packages Windows Help
R Console
R version 3.2.1 (2015-06-18) -- "World-Famous Astronaut"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
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.
>

RStudio

- RStudio is a **free** and open source integrated development environment (IDE) for R
- RStudio is available in two editions:
  - RStudio Desktop, where the program is run locally as a regular desktop application
  - RStudio Server, which allows accessing RStudio using a web browser while it is running on a remote Linux server.

C:/D-Datos/Göttingen/Papers/LiDAR variables selection Edu/Box-Cox - RStudio

File Edit Code View Plots Session Build Debug Tools Help

Box-Cox — LiDAR variables selection Edu

01-Introduction.Rnw | 02-SRS.Rnw | Data analysis Kalimantan.R

Source on Save | Run | Source

```
# Biomass calculation per tree
kalimantan$w.brown<-brown.moist.d(kalimantan$dbh)
kalimantan$w.yamakura<-yamakura.stem(kalimantan$dbh, kalimantan$h)+yamakura.branch(yamakura.stem(k
kalimantan$w.basuki<-basuki.mixed.d(kalimantan$dbh)
kalimantan$w.samalca<-samalca.d(kalimantan$dbh)
kalimantan$w.hashimoto<-hashimoto.d(kalimantan$dbh)
kalimantan$w.kenzo<-kenzo.d(kalimantan$dbh)
kalimantan$w.forda<-forda.d(kalimantan$dbh)
kalimantan$w.jaya<-jaya.d(kalimantan$dbh)
kalimantan$w.novita<-novita.d(kalimantan$dbh)
kalimantan$w.nugroho.d<-nugroho.d(kalimantan$dbh)
kalimantan$w.nugroho.d.h<
                                                  $h)

plot(kalimantan$dbh, kali
                                                  n with different models", xlab="DBH
points(kalimantan$dbh, ka
points(kalimantan$dbh, ka
points(kalimantan$dbh, ka
points(kalimantan$dbh, kalimantan$w.hashimoto, col=5)
points(kalimantan$dbh, kalimantan$w.kenzo, col=6)
points(kalimantan$dbh, kalimantan$w.forda, col=7)
points(kalimantan$dbh, kalimantan$w.jaya, col=8)
points(kalimantan$dbh, kalimantan$w.novita, col=9)
points(kalimantan$dbh, kalimantan$w.nugroho.d, col=10)
points(kalimantan$dbh, kalimantan$w.nugroho.d.h, col=11)

legend(10,8000, c("Brown", "Yamakura", "Basuki", "Samalca", "Hashimoto", "Kenzo", "Forda", "Jaya",

# Summing all values per plot and nested plot
bio.plot.brown<-as.data.frame(tapply(kalimantan$w.brown, list(kalimantan$plot_id, kalimantan$subpl
```

310:1 (Untitled) R Script

R script

Console | Compile PDF

C:/D-Datos/Göttingen/Indonesia Porjects/Kalimantan Project/Final Data/

```
> kal.plot<-merge(kal.plot, Dmed.Hmed.plot,  by="Plot")
>
>
> # Calculating the                                    AMETER
> kal.plot$dg<-sqrt((4*kal.plot$
>
>
> write.csv(kal.plot, "Kalimanta
>
```

R console

Environment | History

Import Dataset | Clear | List

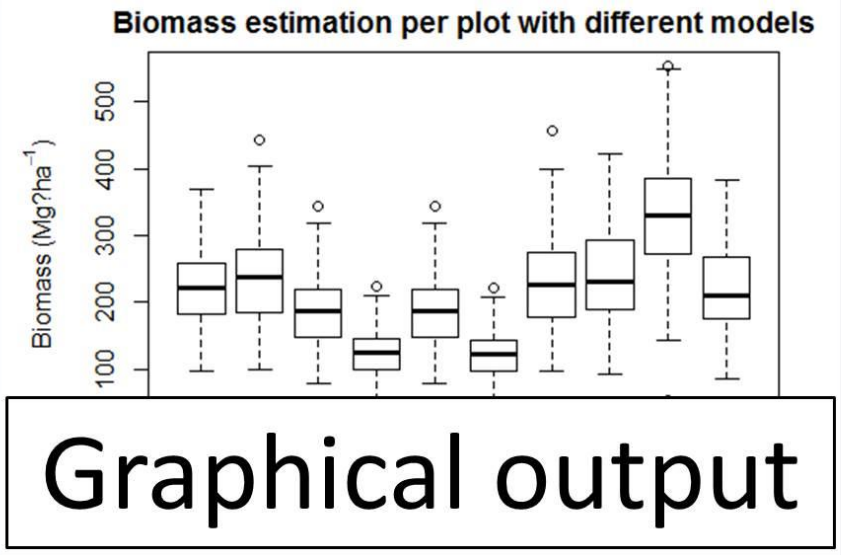
Global Environment

hil.trees	716 obs. of 23 variables
kal.plot	94 obs. of 18 variables
kalimantan	1993 obs. of 44 variables
lsi.plots	59 obs. of 19 variables
lsi	
pub	
wei	
Valu	
EF	12.4339799290545
EFm	49.7359197162173
EFs	198.943678864869
N.tot	2696.5863280181

R environment

Files | Plots | Packages | Help | Viewer

Zoom | Export | Clear All



Graphical output

R basics

- R is a case sensitive language
  - FOO, Foo, and foo are three different objects
- Results of calculations can be stored in objects using the assignment operators:
  - An arrow (<-) formed by a smaller than character and a hyphen without a space!
  - The equal character (=)
- Object names
  - cannot contain `strange' symbols like !, +, -, #
  - dot (.) and an underscore ( ) are allowed
  - can contain a number but cannot start with a number

Types of brackets used in R

- Round brackets / Parenthesis ()
  - For functions
  - mean(), sd(),max()
- Square brackets
  - For manipulating data
  - Data[1,1]
  - Data[1,]
  - Data[,1]

Data frames

- Way of storing data with different types
- Data can be of different types
- There are different ways to create and manipulate data frames
  - Create
  - Import data

Comma-separated values (CSV) file

- Common data exchange format
  - Widely supported by software applications.
  - Move tabular data between programs (SPSS, SAS,ect)
- Use commas as field separators
- Interpreted as a sequence of characters
  - human-readable by text editor
  - Each line - data record

Selecting data

- Data frame
  - Elements
  - Rows
  - Columns

17 observations of 9 variables

	YEAR	CBE	PPO	CPO	PFO	DINC	CFO	RDINC	RFP
1	1925	58.6	60.5	65.8	65.8	51.4	90.9	68.5	877
2	1926	59.4	63.3	63.3	68.0	52.6	92.1	69.6	899
3	1927	53.7	59.9	66.8	65.5	52.1	90.9	70.2	883
4	1928	48.1	56.3	69.9	64.8	52.7	90.9	71.9	884
5	1929	49.0	55.0	68.7	65.6	55.1	91.1	75.2	895
6	1930	48.2	59.6	66.1	62.4	48.8	90.7	68.3	874
7	1931	47.9	57.0	67.4	51.4	41.5	90.0	64.0	791
8	1932	46.0	49.5	69.7	42.8	31.4	87.8	53.9	733
9	1933	50.8	47.3	68.7	41.6	29.4	88.0	53.2	752
10	1934	55.2	56.6	62.2	46.4	33.2	89.1	58.0	811
11	1935	52.2	73.9	47.7	49.7	37.0	87.3	63.2	847
12	1936	57.3	64.4	54.4	50.1	41.8	90.5	70.5	845
13	1937	54.4	62.2	55.0	52.1	44.5	90.4	72.5	849
14	1938	53.6	59.9	57.4	48.4	40.8	90.6	67.8	803
15	1939	53.9	51.0	63.9	47.1	43.5	93.8	73.2	793
16	1940	54.2	41.5	72.4	47.8	46.5	95.5	77.6	798
17	1941	60.0	43.9	67.4	52.2	56.3	97.5	89.5	830

Summary measures

- Mean - mean(variable)
- Standard deviation - sd(variable)
- Quantiles - quantile(variable, c(.32, .57, .98))
- Number of elements - length()
- Minimum - min()
- Maximum - max()
- Tabulate – table()
- Missing values – is.na()

Graphs

- Histogram - hist()
- Box plot - boxplot()
- Scatter plot - plot(x,y)
- Stem & leaf - stem()
- Pie chart - pie()
- Bar plot - barplot()

R Help

- help.start()
- help(mean) or ? Mean
- RSiteSearch("mean")
- apropos("mean")
- example(mean)
- data()

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Project: (None)

Prediction_2011_2013.R | sample_size.R | v1_data.R | Mixed_effect_models.R | Untitled1*

Source on Save | Run | Source

```r
# Load required library
library(lme4)

# Sample data (replace with your actual data)
y <- rnorm(100) # Level 1 outcome variable
group1 <- gl(10, 10) # Grouping factor (10 groups with 10 observations each)
group2 <- gl(4, 25) # Grouping factor (4 groups with 25 observations each)
x1 <- rnorm(100) # Level 1 predictor
x2 <- rnorm(100) # Level 1 predictor

# Define the model formula (specifies diagonal structure)
model <- lmer(y ~ x1 + (1 | group1 ) , REML = FALSE)
summary(model)

model <- lmer(y ~ x1 + x2 + (1 | group1 ) , REML = FALSE)
summary(model)

model <- lmer(y ~ x1 + x2 +  (1 | group1 ) + (1 | group2 ) , REML = FALSE)
summary(model)

?var.test

x <- rnorm(50, mean = 0, sd = 2)
y <- rnorm(30, mean = 1, sd = 1)
var.test(x, y)                # Do x and y have the same variance?
var.test(lm(x ~ 1), lm(y ~ 1))  # The same.
```

Saved to this PC

26:30 (Top Level) R Script

Environment History Connections Tutorial

Console Terminal | Background Jobs

R 4.3.3 · ~/

```
> help.start()
If nothing happens, you should open
'http://127.0.0.1:31820/doc/html/index.html' yourself
>
```

Files Plots Packages Help Viewer Presentation

The R Language | Find in Topic

Statistical Data Analysis

Manuals

An Introduction to R
Writing R Extensions
R Data Import/Export
The R Language Definition
R Installation and Administration
R Internals

Reference

Packages
Search Engine & Keywords

Miscellaneous Material

About R
License
NEWS
Authors
Frequently Asked Questions
User Manuals
Resources
Thanks
Technical papers

Material specific to the Windows port

CHANGES up to R 2.15.0
Windows FAQ

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Project: (None)

Prediction_2011_2013.R | sample_size.R | v1_data.R | Mixed_effect_models.R | Untitled1*

Source on Save | Run | Source

```r
# Load required library
library(lme4)

# Sample data (replace with your actual data)
y <- rnorm(100) # Level 1 outcome variable
group1 <- gl(10, 10) # Grouping factor (10 groups with 10 observations each)
group2 <- gl(4, 25) # Grouping factor (4 groups with 25 observations each)
x1 <- rnorm(100) # Level 1 predictor
x2 <- rnorm(100) # Level 1 predictor

# Define the model formula (specifies diagonal structure)
model <- lmer(y ~ x1 + (1 | group1 ) , REML = FALSE)
summary(model)

model <- lmer(y ~ x1 + x2 + (1 | group1 ) , REML = FALSE)
summary(model)

model <- lmer(y ~ x1 + x2 +  (1 | group1 ) + (1 | group2 ) , REML = FALSE)
summary(model)

?var.test

x <- rnorm(50, mean = 0, sd = 2)
y <- rnorm(30, mean = 1, sd = 1)
var.test(x, y)                  # Do x and y have the same variance?
var.test(lm(x ~ 1), lm(y ~ 1))  # The same.
```

26:30 (Top Level) R Script

Environment History Connections Tutorial

Console

Files Plots Packages Help Viewer Presentation

An Introduction to R | Find in Topic

An Introduction to R

This is an introduction to R (“GNU S”), a language and environment for statistical computing and graphics. R is similar to the award-winning[1] S system, which was developed at Bell Laboratories by John Chambers et al. It provides a wide variety of statistical and graphical techniques (linear and nonlinear modelling, statistical tests, time series analysis, classification, clustering, ...).

This manual provides information on data types, programming elements, statistical modelling and graphics.

This manual is for R, version 4.3.3 (2024-02-29).

Table of Contents

Student's T test

- help(t.test)
- independent 2-group t-test
  - t.test(y~x) # where y is numeric and x is a binary factor
- independent 2-group t-test
  - t.test(y1,y2) # where y1 and y2 are numeric
- paired t-test
  - t.test(y1,y2,paired=TRUE) # where y1 & y2 are numeric
- one sample t-test
  - t.test(y,mu=3) # Ho: mu=3

Compare Two Variances

- var.test(y1,y2)

Chisquared test

- help(chisq.test)
  - chisq.test(table)
  - fisher.test (table)

Simple Linear Regression

- y = mx + c
- plot(x,y)
- Formula
  - model<- lm(y ~ x)
  - summary(model)

Feedback

1. what did you like about this session?
2. what didn't you like about this session?
3. what did you learn from this session?

Thank you