

R for SAS Programmers



PhUSE Conference
London 2014

Friedrich Schuster

It's Different, but Friendly



Agenda

- History and main characteristics of R
- Clinical study example
- Recommended Reading

History



70's

History

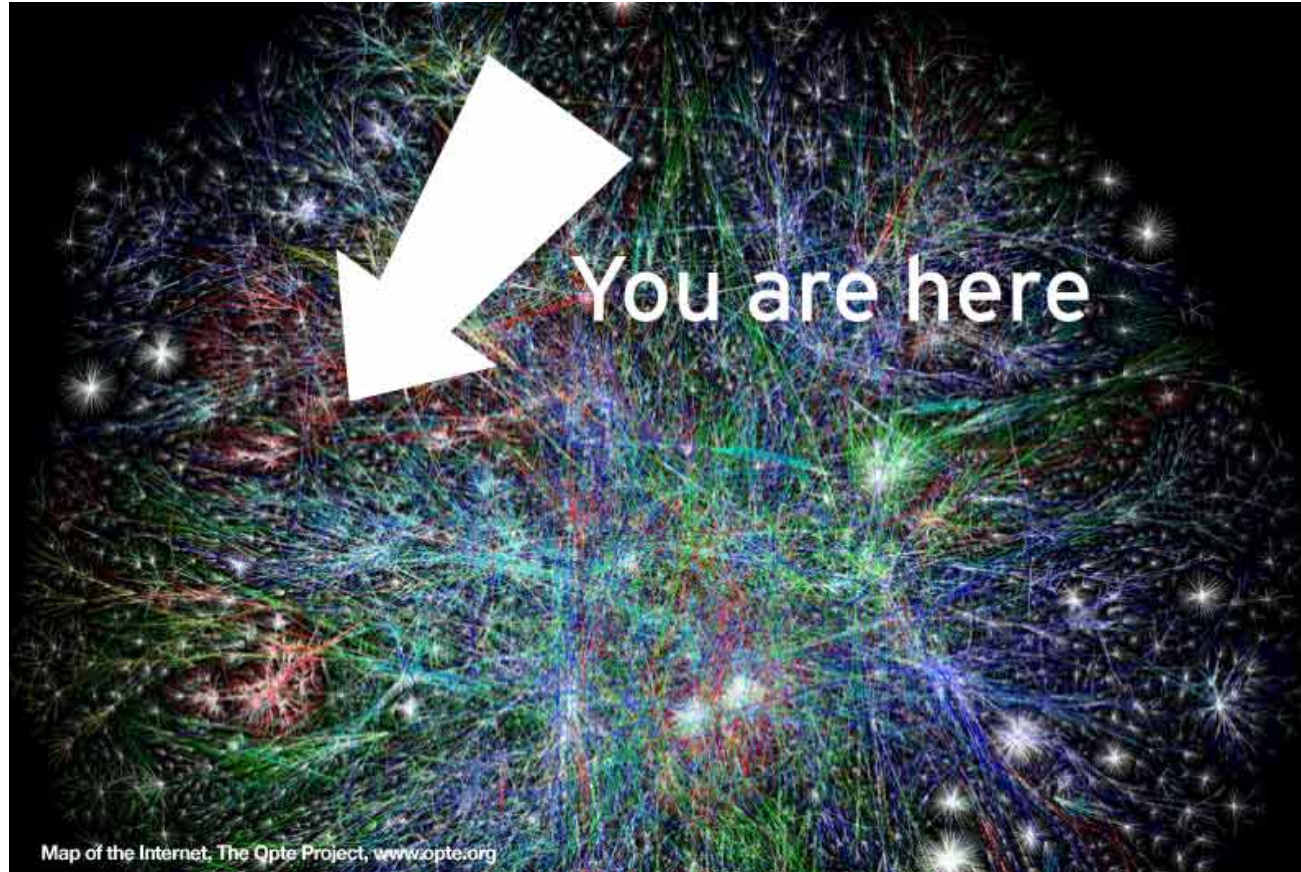


History



90's

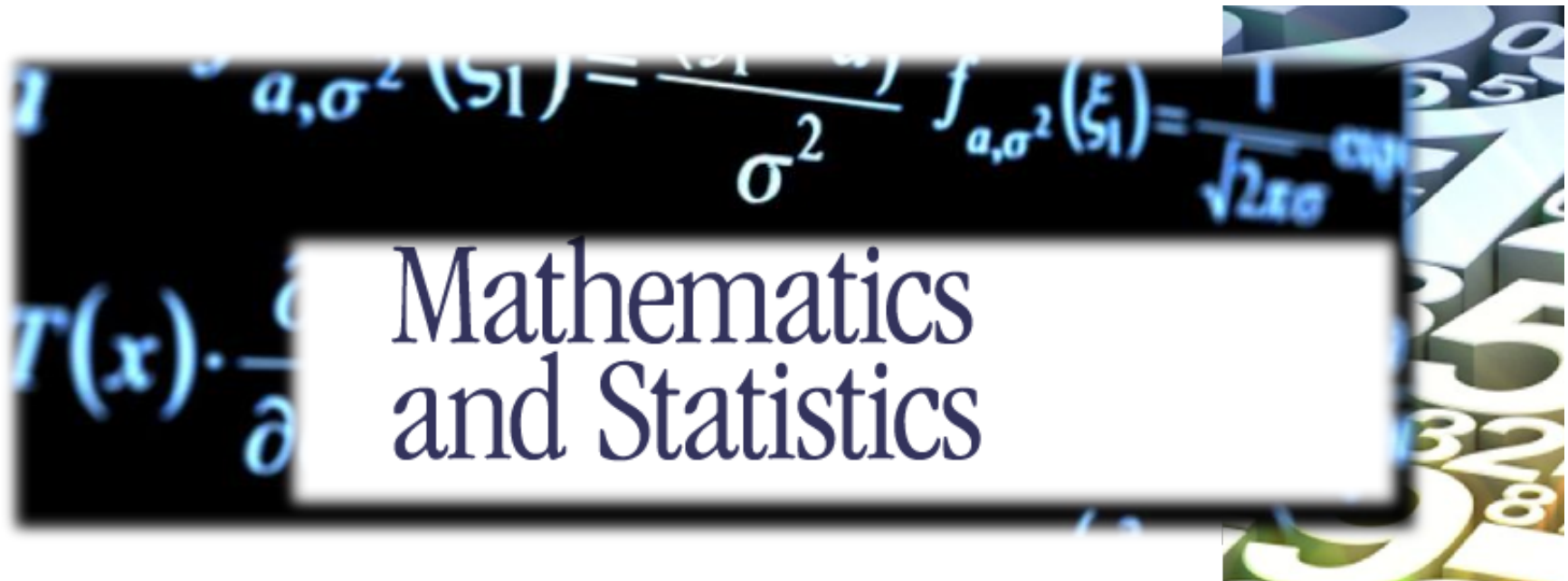
Today: The R UniveRse





R programming language

Domain specific language



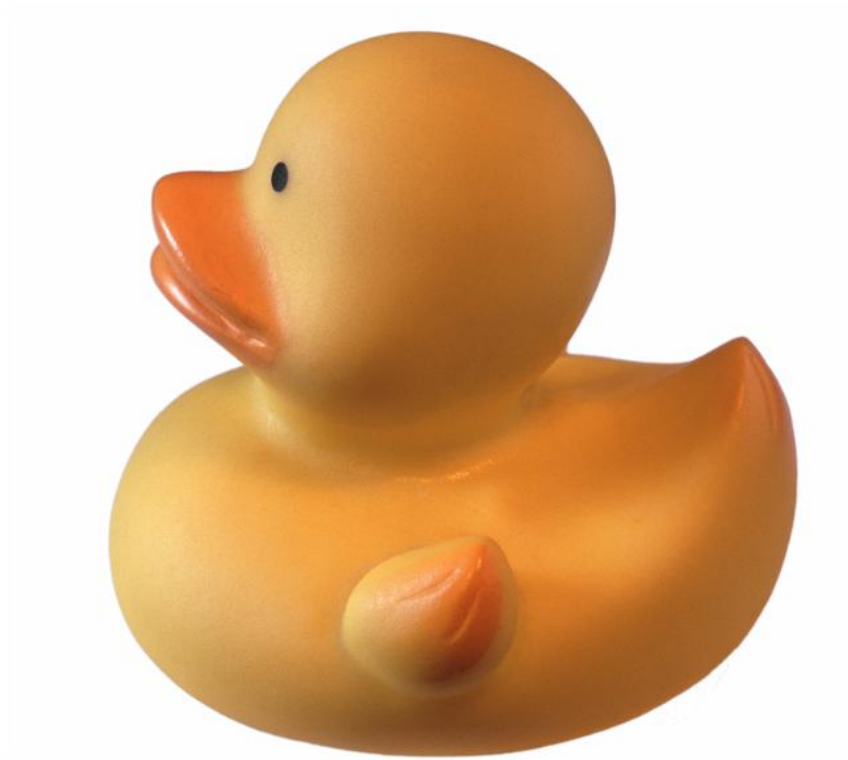
R programming language

Vector language

$$\begin{bmatrix} 1 & 0 & 2 & 0 \\ 0 & 3 & 0 & 4 \\ 0 & 0 & 5 & 0 \\ 6 & 0 & 0 & 7 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 5 \\ 1 \\ 8 \end{bmatrix} = \begin{bmatrix} 4 \\ 47 \\ 5 \\ 68 \end{bmatrix}$$

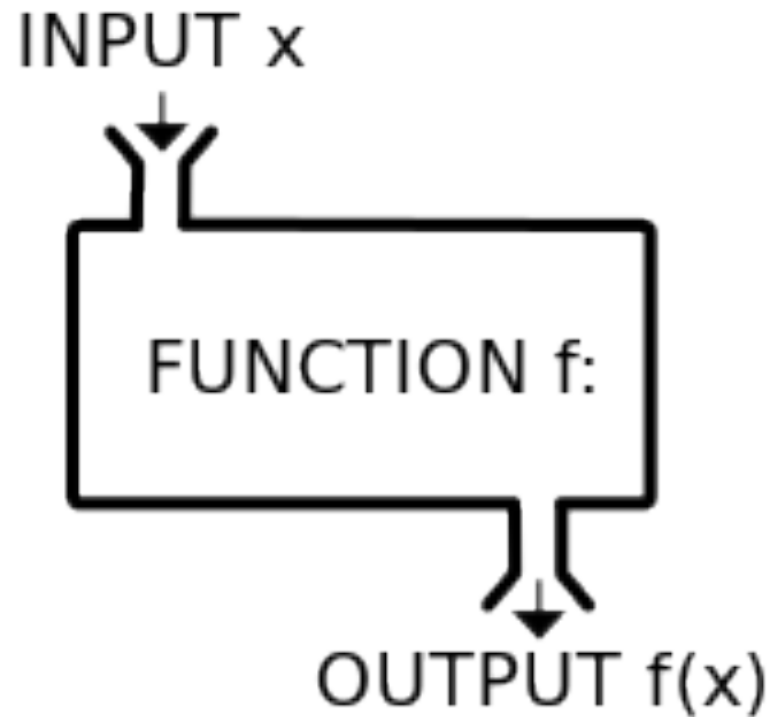
R programming language

Object based



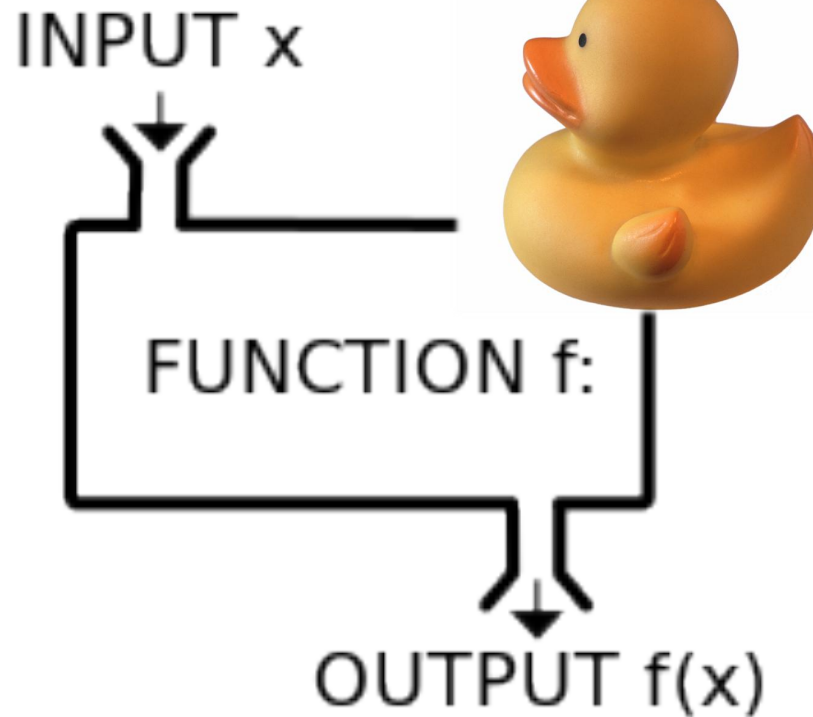
R programming language

Functional language



R programming language

Functions are objects



R programming language

Highly extensible



R as „environment for statistical computing“

- Runs code interactively or in batch mode
- On workstations and/or servers
- Code development: in a console or modern IDE
- Data resides in main memory, not on disk
- Import, export of data
- Powerful graphics subsystem(s)
- Built in help system

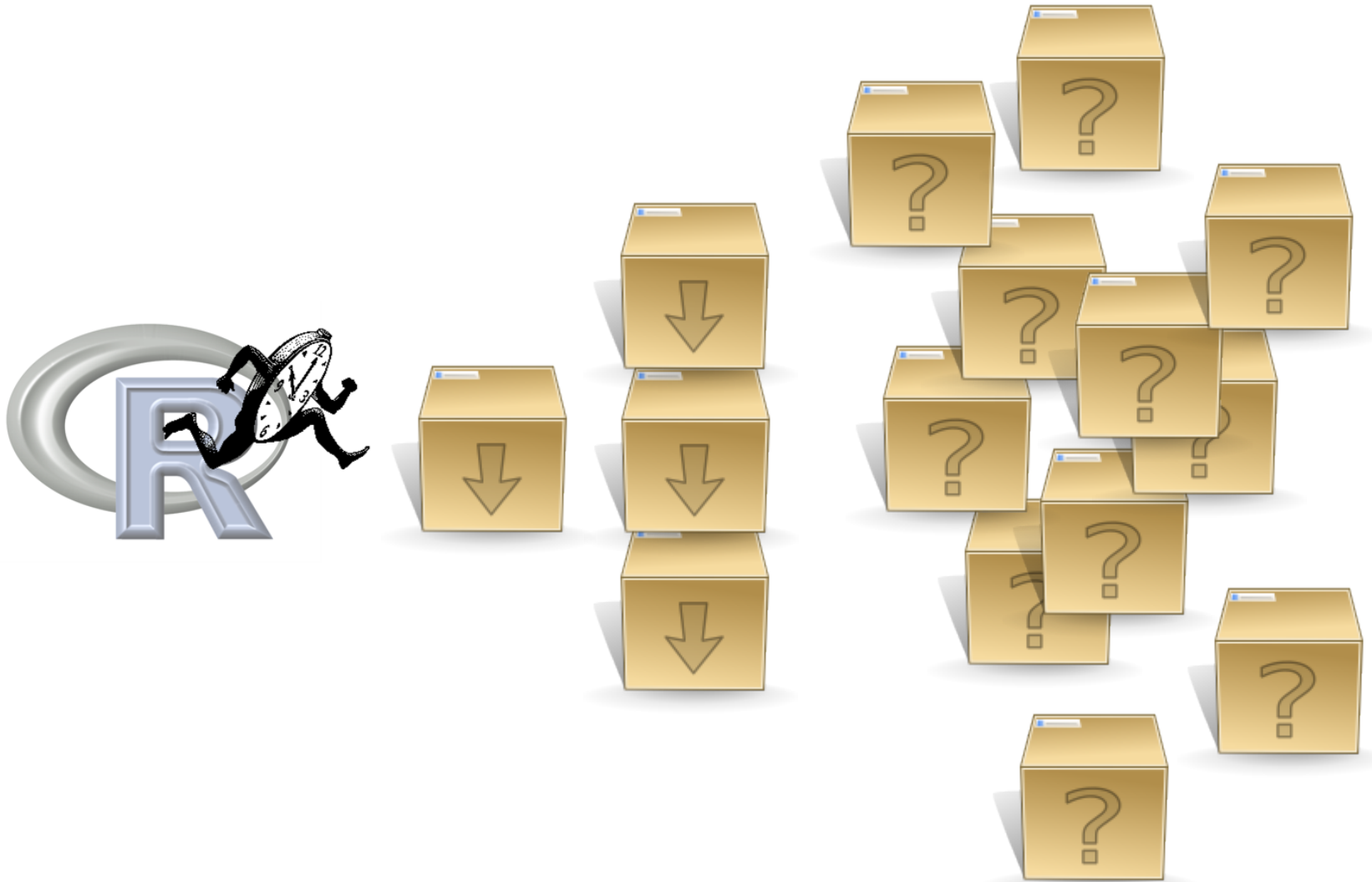
R packages and package infrastructure



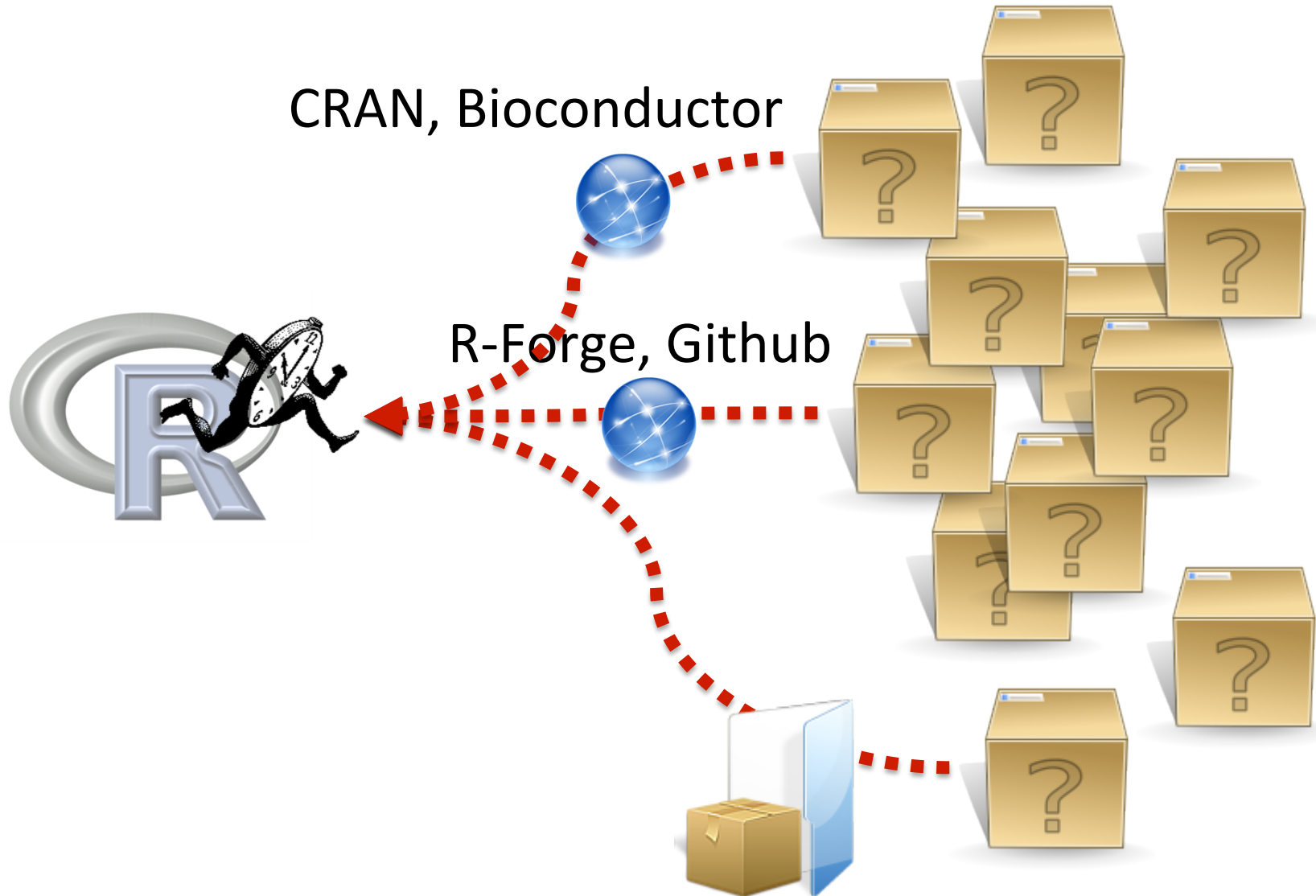
R packages and package infrastructure



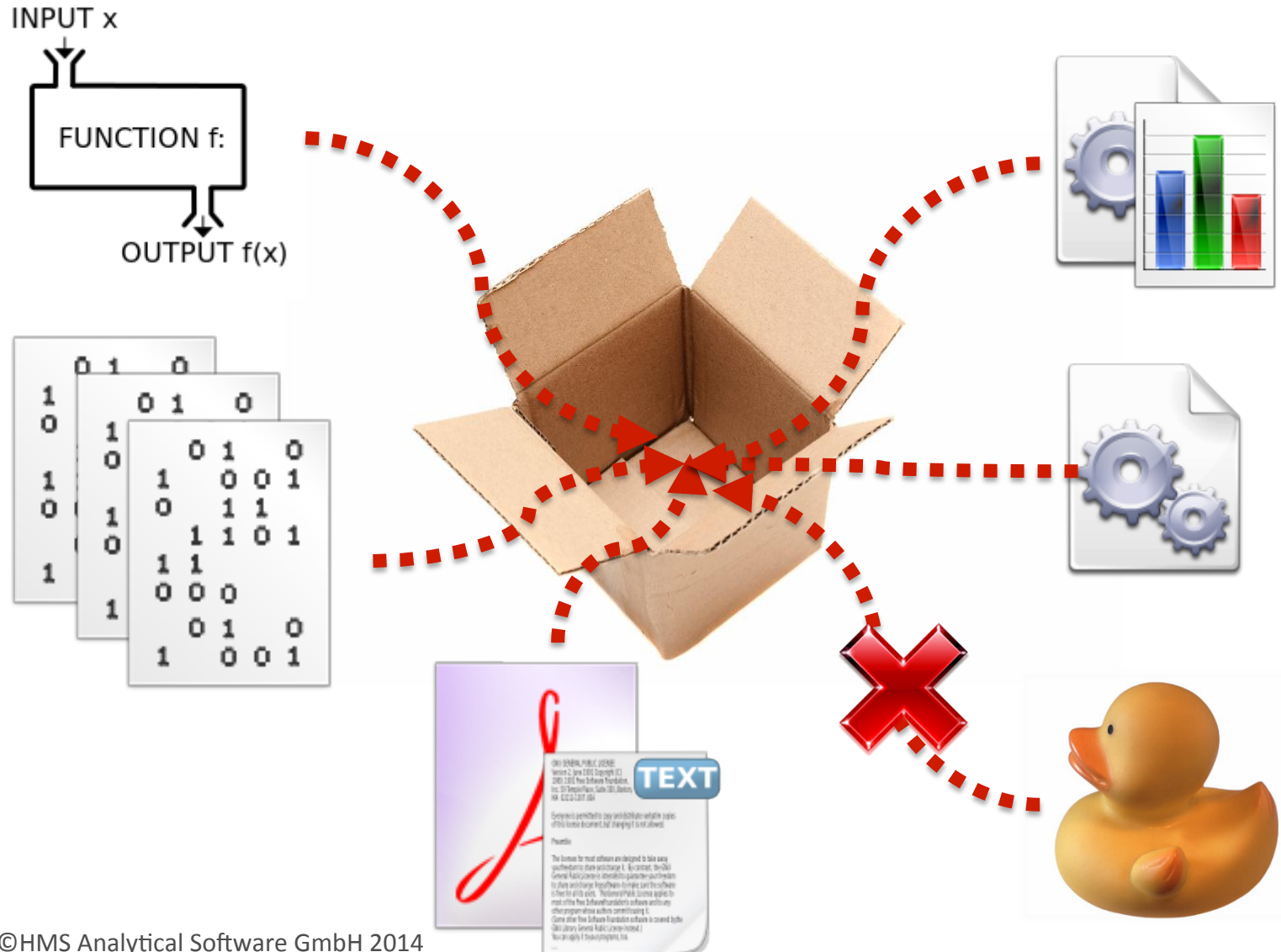
R packages and package infrastructure



R packages and package infrastructure



R packages and package infrastructure



Code examples: data access

Reading a csv file:

- SAS: PROC IMPORT, PROC PRINT

- R:

```
> data1 <- read.csv("../data/DiastolicBloodPressure_initial.csv")
> head(data1)
```

	Subject	Treatment	DBP.Baseline	DBP.Month1	DBP.Month2	DBP.Month3	DBP.Month4
1	1	A	114	115	NA	109	105
2	2	A	116	113	NA	NA	101
3	3		119	115	113	104	98
4	4	A	115	113	112	109	101
5	5	A	116	112	107	104	105
6	6	A	117	112	113	104	102

	Age	Sex
1	43	F
2	51	M
3	48	F
4	42	
5	49	M
6	47	M

Code examples: data structure

Show variable information:

- SAS: PROC CONTENTS, PROC SQL
- R:

```
> str(data1)
'data.frame': 40 obs. of 9 variables:
 $ Subject      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Treatment    : Factor w/ 3 levels "", "A", "B": 2 2 1 2 2 2 2 2 2 2 ...
 $ DBP.Baseline: int  114 116 119 115 116 117 118 120 114 115 ...
 $ DBP.Month1   : int  115 113 115 113 112 112 111 115 112 113 ...
 $ DBP.Month2   : int  NA NA 113 112 107 113 100 113 113 108 ...
 $ DBP.Month3   : int  109 NA 104 109 104 104 109 102 109 106 ...
 $ DBP.Month4   : int  105 101 98 101 105 102 99 102 103 97 ...
 $ Age          : int  43 51 48 42 49 47 50 61 43 51 ...
 $ Sex          : Factor w/ 3 levels "", "F", "M": 2 3 2 1 3 3 2 3 3 3 ...
```

Code examples: identify incomplete observations

Finding incomplete observations:

- SAS: Data step
- R:

```
> data1 <- read.csv("./data/DiastolicBloodPressure_initial.csv")  
> head(data1)
```

	Subject	Treatment	DBP.Baseline	DBP.Month1	DBP.Month2	DBP.Month3	DBP.Month4
1	1	A	114	115	NA	109	105
2	2	A	116	113	NA	NA	101
3	3		119	115	113	104	98

```
> complete.cases(data1)
```

```
[1] FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
[13] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
[25] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
> data1[!complete.cases(data1), ]
```

	Subject	Treatment	DBP.Baseline	DBP.Month1	DBP.Month2	DBP.Month3	DBP.Month4
1	1	A	114	115	NA	109	105
2	2	A	116	113	NA	NA	101
Age Sex							
1	43	F					
2	51	M					

Code examples: identify incomplete observations

Finding blank factor values:

- SAS: Data step
- R:

```
> data1[ , "Treatment"]  
[1] A A A A A A A A A A A A A A A A A A A B B B B B B B B B B B B B B B  
B  
[40] B  
Levels: A B  
> data1$Treatment  
[1] A A A A A A A A A A A A A A A A A A A B B B B B B B B B B B B B B B  
B  
[40] B  
Levels: A B
```

```
> data1[ !data1$Treatment %in% c("A", "B") , ]  
Subject Treatment DBP.Baseline DBP.Month1 DBP.Month2 DBP.Month3 DBP.Month4 Age  
3 3 119 115 113 104 98 48  
Sex  
3 F
```

Code examples: identify incomplete observations

Finding blank factor values:

- SAS: Data step
- R:

```
# Checking for invalid codes
data1[ !data1$Treatment %in% c("A", "B") , ]
data1[ !data1$Sex %in% c("M", "F") , ]
data1[ !data1$Age %in% 35:65 , ]

# Correcting missing data entries
data1[ data1$Subject == 1 , "DBP.Month2"] <- 113
data1[ data1$Subject == 2 , "DBP.Month2"] <- 112
data1[ data1$Subject == 2 , "DBP.Month3"] <- 103
data1[ data1$Subject == 3 , "Treatment"] <- "A"
data1[ data1$Subject == 4 , "Sex"] <- "F"
```

Code examples: create a function

R Functions

- SAS: Macro or PROC FCMP
- R:

```
# This creates the R function 'info'
info <- function(data1) {
  writeLines("display the structure of the data frame")
  str(data1)
  writeLines("Then print the first 6 observations to see the variables and
  print(head(data1))
}

# Call the new function
info(data1)
```

Code examples: date arithmetic

Compute year of birth as a new variable:

```
> # Compute year of birth
> library("lubridate")
> studyDate <- as.Date("2014-06-01")
> YearOfBirth <- year(studyDate - years(data1$Age))
> YearOfBirth
[1] 1971 1963 1966 1972 1965 1967 1964 1953 1971 1963 1967 1969 1960 1962 1972
[16] 1970 1966 1951 1973 1963 1975 1974 1975 1976 1975 1973 1958 1958 1976 1957
[31] 1967 1966 1953 1965 1962 1959 1969 1972 1965 1964
> # Write year of birth as a variable to the data frame: combine columns
> data2 <- cbind(data1, YearOfBirth)
> str(data2)
'data.frame': 40 obs. of 10 variables:
 $ Subject      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Treatment    : Factor w/ 3 levels "", "A", "B": 2 2 2 2 2 2 2 2 2 2 ...
 $ DBP.Baseline: int  114 116 119 115 116 117 118 120 114 115 ...
 $ DBP.Month1   : int  115 113 115 113 112 112 111 115 112 113 ...
 $ DBP.Month2   : num  113 112 113 112 107 113 100 113 113 108 ...
 $ DBP.Month3   : num  109 103 104 109 104 104 109 102 109 106 ...
 $ DBP.Month4   : int  105 101 98 101 105 102 99 102 103 97 ...
 $ Age          : int  43 51 48 42 49 47 50 61 43 51 ...
 $ Sex          : Factor w/ 3 levels "", "F", "M": 2 3 2 2 3 3 2 3 3 3 ...
 $ YearOfBirth  : num  1971 1963 1966 1972 1965 ...
```

Code examples: data exploration

Data summary:

- SAS: PROC MEANS

- R:

```
> summary(data1)
```

Subject	Treatment	DBP.Baseline	DBP.Month1	DBP.Month2
Min. : 1.00	: 1	Min. :114.0	Min. :111.0	Min. :100.0
1st Qu.:10.75	A:19	1st Qu.:115.0	1st Qu.:113.0	1st Qu.:112.0
Median :20.50	B:20	Median :116.5	Median :115.0	Median :113.0
Mean :20.50		Mean :116.7	Mean :114.3	Mean :112.4
3rd Qu.:30.25		3rd Qu.:118.0	3rd Qu.:115.0	3rd Qu.:113.0
Max. :40.00		Max. :121.0	Max. :119.0	Max. :118.0
				NA's :2

DBP.Month3	DBP.Month4	Age	Sex
Min. :102.0	Min. : 97.0	Min. :38.00	: 1
1st Qu.:107.0	1st Qu.:101.8	1st Qu.:42.00	F:17
Median :109.0	Median :106.5	Median :48.00	M:22
Mean :109.5	Mean :106.7	Mean :47.83	
3rd Qu.:113.5	3rd Qu.:112.0	3rd Qu.:51.25	
Max. :117.0	Max. :115.0	Max. :63.00	
NA's :1			

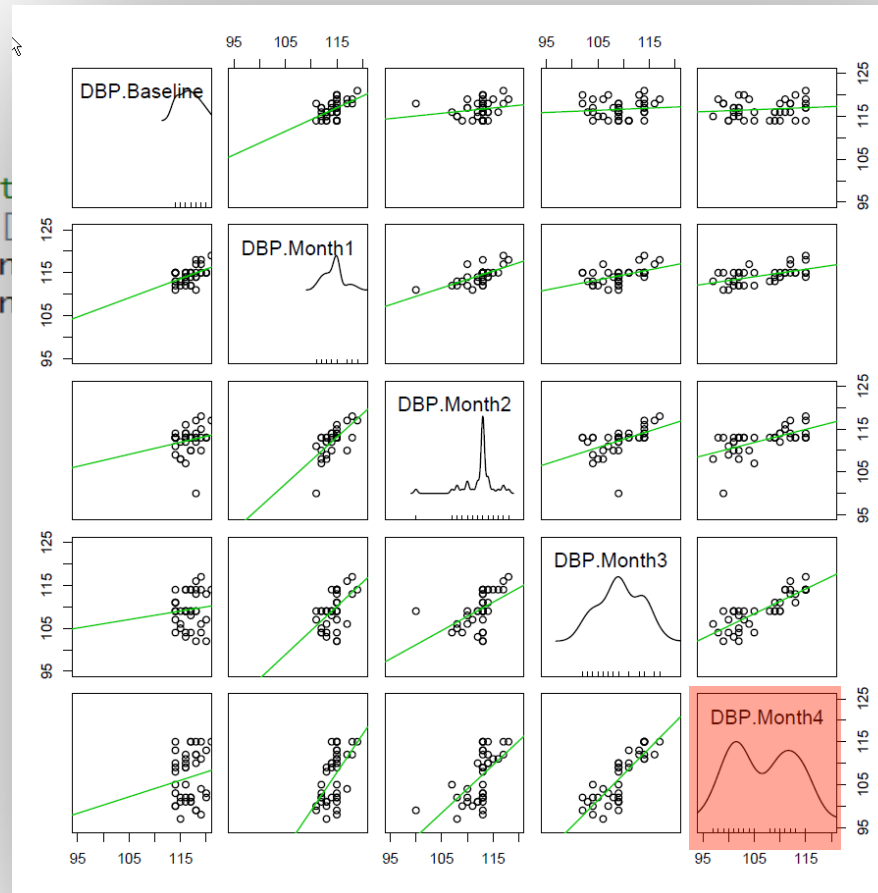
Code examples: data exploration

Scatterplot matrix:

■ SAS: PROC SGSCATTER with ODS Graphics

■ R:

```
# Modified scatterplot
data2 <- data1[,c("Treat", "DBP.Baseline", "DBP.Month1", "DBP.Month2", "DBP.Month3", "DBP.Month4")]
scatterplotMatrix(data2[,c("DBP.Baseline", "DBP.Month1", "DBP.Month2", "DBP.Month3", "DBP.Month4")],
  spm(data2[data2$Treatment == "A", 2:6]),
  spm(data2[data2$Treatment == "B", 2:6]))
```



```
,"DBP.Month3", "DBP.Month4")],
  spm(data2[data2$Treatment == "A", 2:6]),
  spm(data2[data2$Treatment == "B", 2:6]))
```

Code examples: blood pressure by treatment groups

Mean value(s):

- SAS: PROC MEAN, BY processing
- R:

```
> diff <- data1$DBP.Month4 - data1$DBP.Baseline
> by(diff, data2$Treatment, mean)
data2$Treatment: A
[1] -15.2
-----
data2$Treatment: B
[1] -4.8
```

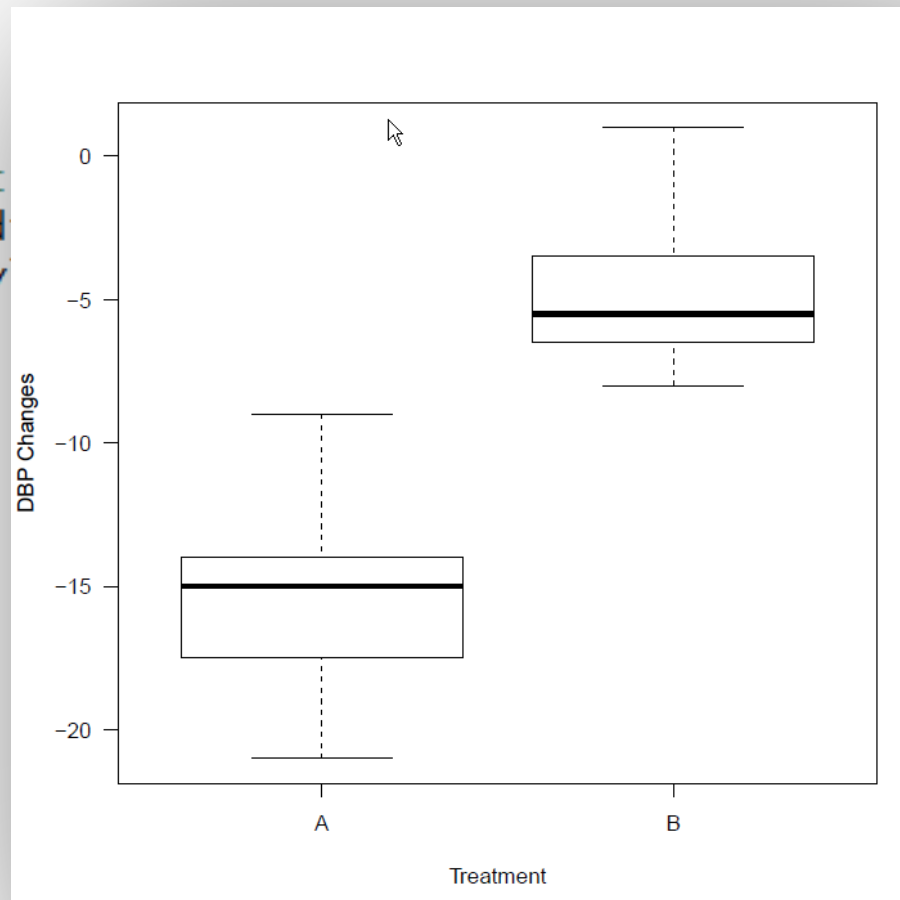
Code examples: blood pressure by treatment groups

Boxplots

- SAS: PROC BOXPLOT

- R:

```
# boxplot  
boxplot(d  
y
```



ent",

Code examples: blood pressure by treatment groups

Student's t-test:

- SAS: PROC TTEST
- R:

```
> # T-test  
> ttResult <- t.test(diff ~ Treatment, data1, var.equal=TRUE)  
> ttResult
```

Two Sample t-test

data: diff by Treatment

t = -12.1504, df = 38, p-value = 1.169e-14

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-12.132758 -8.667242

sample estimates:

mean in group A	mean in group B
-15.2	-4.8

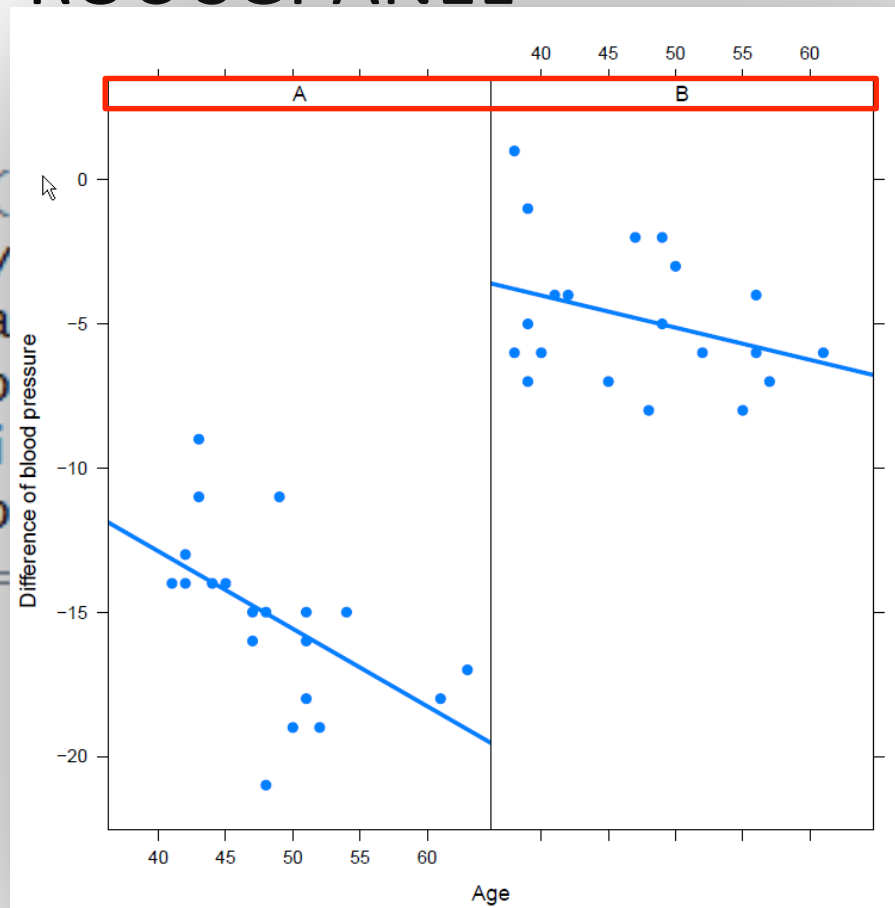
Code examples: blood pressure by treatment groups

Lattice/Trellis plots

- SAS: PROC SGPANEL

- R:

```
library(  
print(xy  
, data  
, xlab  
, stri  
, ylab  
, lwd=  
)
```



```
ure "  
"p", "r"))
```

Code examples: blood pressure by treatment groups

Linear model, ANOVA

- SAS: PROC GLM, PROC REG, PROC ANOVA, PROC MIXEDGPANEL
- R:

```
> # First fit a linear model and create a model object...  
> lmResult = lm(diff ~ Treatment + Age, data=data1)  
> # ... then compute the analysis of variance tables from it.  
> anova(lmResult)
```

Analysis of Variance Table

Response: diff

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Treatment	1	1081.60	1081.60	176.0395	1.228e-15	***
Age	1	51.07	51.07	8.3119	0.006525	**
Residuals	37	227.33	6.14			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

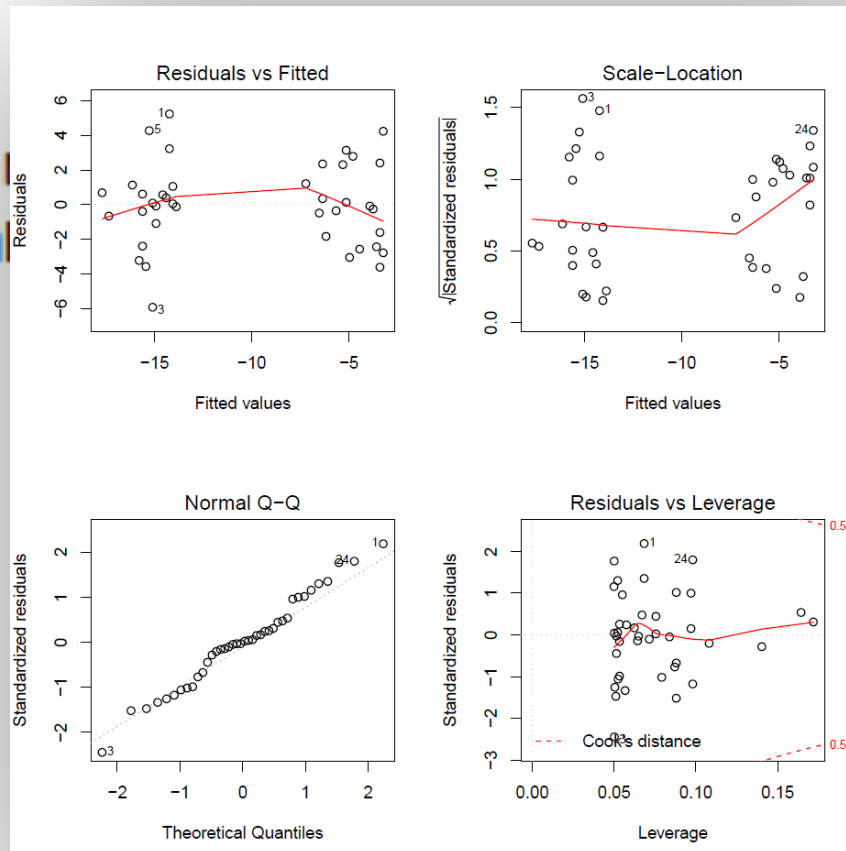
Code examples: blood pressure by treatment groups

Diagnostic plots

- SAS: ODS Graphics with PROC GLM

- R:

```
layout(1,1,1)  
plot(lm)
```



Reporting

PDF output

■ SAS: PR

■ R:

```
> print(xtable(ano
+       , caption=
+       , label =
+       , table.pl
+       , caption.
+       )
% latex table gene
% Fri Sep 12 16:43
\begin{table}[htbp
\centering
\caption{ANOVA tab
\label{tab4RI.coef
\begin{tabular}{lr
\hline
& Df & Sum Sq & M
\hline
Treatment & 1 & 1081.60
Age & 1 & 51.07
Residuals & 37 &
\hline
\end{tabular}
\end{table}
```

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

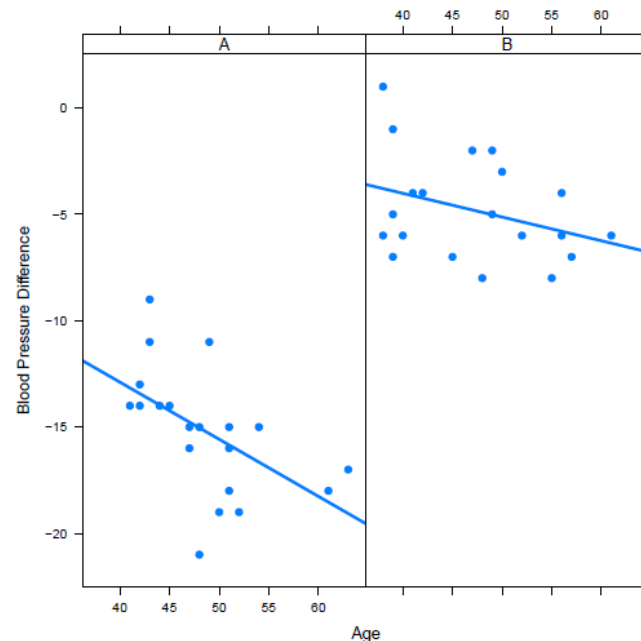


Table 3: ANOVA table for diastolic blood pressure data

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Treatment	1	1081.60	1081.60	176.04	0.0000
Age	1	51.07	51.07	8.31	0.0065
Residuals	37	227.33	6.14		

Recommended reading

- **„R in Action“**, Second edition (Early access edition 2013), by Robert Kabacoff
- **„SAS and R“**, Second edition (2014), by Ken Kleinman and Nicholas J. Horton
- **„Clinical Trial Data Analysis Using R“** (2010), by Din Chen and Karl E. Peace

Thank you!

Friedrich Schuster

Dipl. Soz.

Senior Software Developer

HMS Analytical Software GmbH

Rohrbacher Str. 26 • 69115 Heidelberg

Phone +49 6221 6051-142

Friedrich.Schuster@analytical-software.de

www.analytical-software.de