

SAS and R

-The Odd Couple-

PhUSE - Single Day Event Frankfurt - 04. Mai 2011



Company

- HMS Analytical Software is a specialist for Information Technology in the field of Data Analysis and Business Intelligence Systems
- Profile
 - 40 employees in Heidelberg, Germany
 - SAS Institute Silver Consulting Partner for 14 years
 - Doing data oriented software projects for more than 20 years
- Technologies
 - Analytics and Data Management:
SAS, JMP, R, Microsoft SQL Server
 - Application Development: Microsoft .NET, Java

Agenda

- Concepts behind SAS and R
- Installation and support
- How to develop programs?
- SAS-Macro vs. R functions
- Performance comparison using large data sets
- Interfaces from SAS to R and vice versa
- Software quality
- Conclusion

Concepts Behind SAS

- 1970 presentation of the **Statistical Analysis Software** at the NC State University
- 1976 foundation of the SAS Institute, Cary, North Carolina
- commercial software product
- largest privately-held software company in the world^[1]
 - approx. 11.800 employees
 - 2010 Revenue: \$2.43 billion
 - customers at more than 50,000 sites in 126 countries
- Focus:
 - data integration / data warehousing
 - business intelligence / data mining
 - research and development
 - various and compatible software components
 - license model→ you can not „buy“ SAS

Concepts Behind R

- 1993 first occurrence in a mailing list^[1]
- Open Source Software under the General Public License (GPL)
 - freely available
 - around 2 million users^[2]
- Focus:
 - statistical computing
 - graphics
- The R Foundation for Statistical Computing
 - official voice of the R project
 - non-profit organization seated in Vienna, Austria
 - consists of ordinary and supporting members, mainly the R Development Core-Team
 - holds and administers the copyrights of the R software and documentation.
- huge amount of freely available additional packages
 - not developed by the R Development Core-Team but by programmers from throughout the world

Motivation of the Comparison SAS vs. R

- Focus
 - SAS language and the SAS-windowing environment (Display Manager)
 - R environment
- Not included
 - e.g. SAS Enterprise Guide
 - other frontends of the SAS-Plattform

Installation and Support

SAS

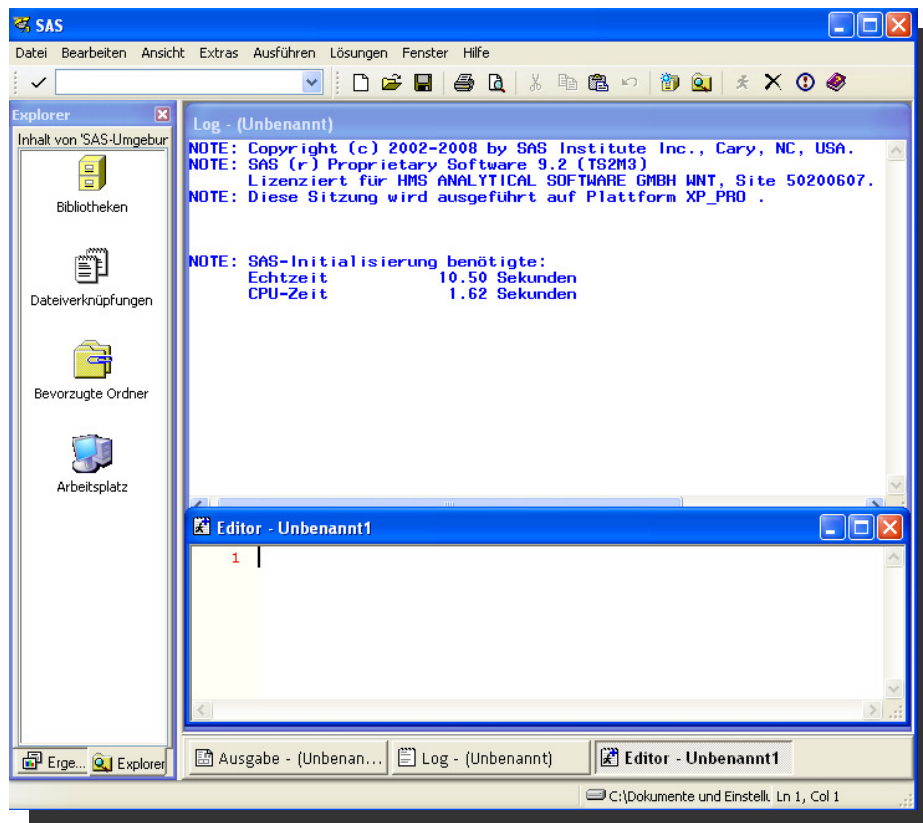
- download-center / DVD
- long installation process
- SAS Foundation
 - PROC CORR, PLOT, ...
 - MACRO
- other modules
 - SAS STAT, GRAPH, ...
- support:
 - online
 - hotline (+49 6221 415 - 200)
 - training
 - newsletter

R

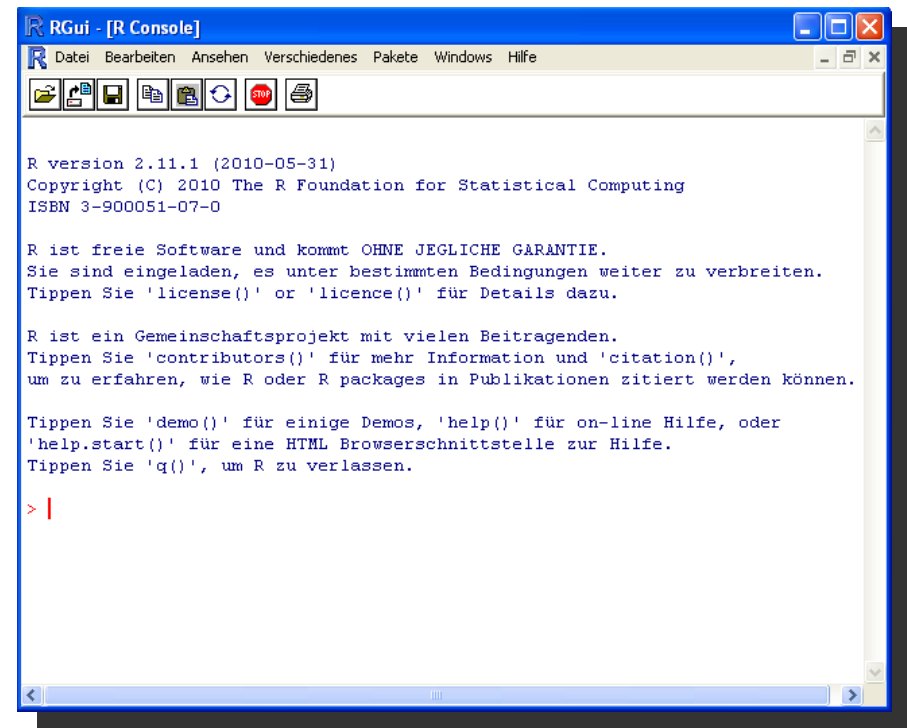
- download from [CRAN](#)
- short installation process
- R „Base“
 - recommended / base packages
- other packages
 - BioConductor, ...
- „support“ / help:
 - [mailing-list](#)
 - [R-journal](#)
 - [bug-tracking](#)
 - [R-search](#)

How to Develop Programs?

SAS windowing environment

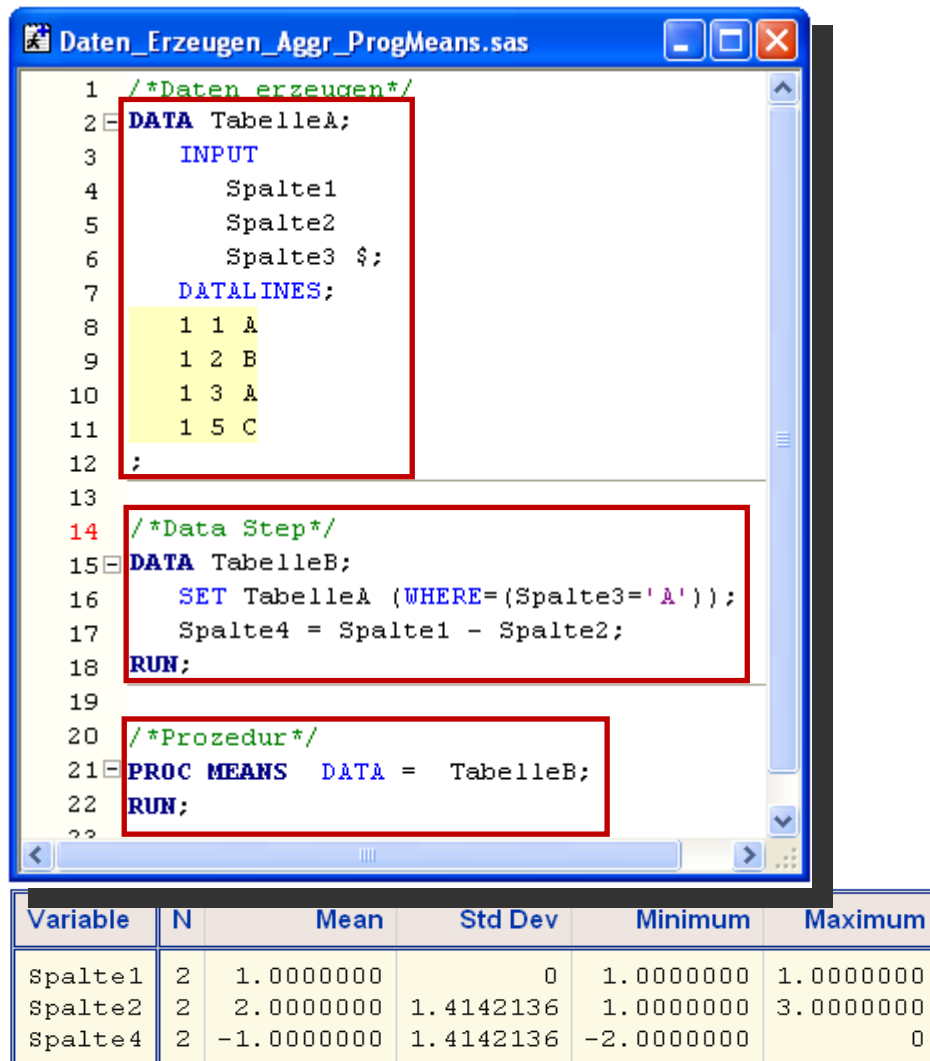


R environment



How to Develop Programs?

SAS Code



```

1  /*Daten erzeugen*/
2  DATA TabelleA;
3      INPUT
4      Spalte1
5      Spalte2
6      Spalte3 $;
7      DATALINES;
8      1 1 A
9      1 2 B
10     1 3 A
11     1 5 C
12 ;
13
14 /*Data Step*/
15 DATA TabelleB;
16     SET TabelleA (WHERE=(Spalte3='A'));
17     Spalte4 = Spalte1 - Spalte2;
18 RUN;
19
20 /*Prozedur*/
21 PROC MEANS DATA = TabelleB;
22 RUN;
23

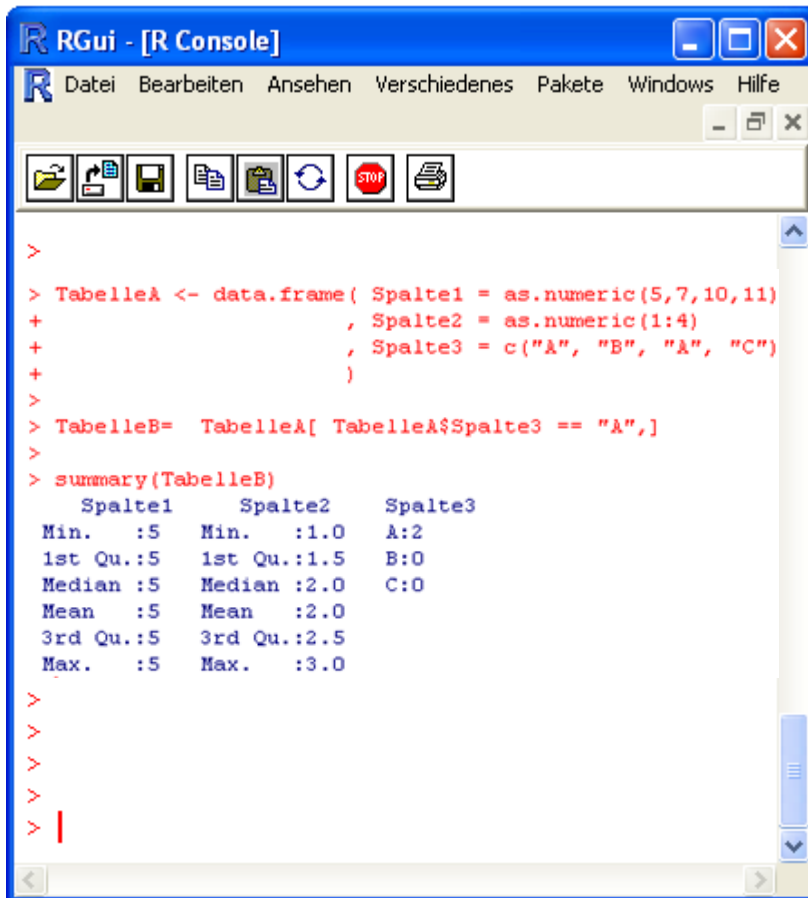
```

Variable	N	Mean	Std Dev	Minimum	Maximum
Spalte1	2	1.00000000	0	1.00000000	1.00000000
Spalte2	2	2.00000000	1.4142136	1.00000000	3.00000000
Spalte4	2	-1.00000000	1.4142136	-2.00000000	0

- code blocks
- data types
 - numerical values
 - alphanumerical values
 - tables
 - matrices (only in SAS IML)
 - macro variables
- DATA STEP
 - key element
 - e.g. generation of data
 - concatenation of data
- PROC MEANS
 - key element
 - special functionalities

How to Develop Programs?

R Code



```
>
> TabelleA <- data.frame( Spalte1 = as.numeric(5,7,10,11)
+                        , Spalte2 = as.numeric(1:4)
+                        , Spalte3 = c("A", "B", "A", "C")
+                        )
>
> TabelleB= TabelleA[ TabelleA$Spalte3 == "A",]
>
> summary(TabelleB)
  Spalte1    Spalte2    Spalte3
Min.   :5    Min.   :1.0    A:2
1st Qu.:5    1st Qu.:1.5    B:0
Median :5    Median :2.0    C:0
Mean   :5     Mean   :2.0
3rd Qu.:5    3rd Qu.:2.5
Max.    :5     Max.   :3.0
>
>
>
>
> |
```

- „line based“ programming
- object oriented data structures
 - vector
 - matrix
 - dataframe
 -
- optimized for matrix manipulation
- R-functions
 - special functionalities

SAS-Macros vs. R-Functions

SAS-macro language

```
%MACRO BerechneMean (p_s_LibRef =
                        , p_sTableName =
                        , p_sOutTableName =
                        )
%LET l_sTableRef = &p_s_LibRef..&p_sTableName.;
PROC MEANS DATA = &l_sTableRef. MEAN ;
  VAR Height;
  CLASS Sex;
  OUTPUT OUT=&p_sOutTableName.;
RUN;
%MEND BerechneMean;
```

```
/*Aufruf des Makros*/
%BerechneMean( p_s_LibRef      = SasHelp
               , p_sTableName  = Class
               , p_sOutTableName = OutData
               );
```

- storing recurring program sequences
- macros are executed by the macro-processor
- macros base on text replacement
- use of macro-libraries
- macro variables store strings (local or global scope)

SAS-Macros vs. R-Functions

R-functions

```
> berechneMean <- function(p_vector){  
+   l_result <- mean(p_vector)  
+   return(l_result)  
+ }
```

```
> berechneMean( c(4,6,7,8,3,6,10))  
[1] 6.285714
```

- storing recurring program sequences
- several functions are usually stored as packages (help pages, manuals etc.)
- internal variables have local scope

Performance Comparison Using Large Data Sets

Pseudocode:

```
start = system time;
```

```
Create data by concatenation of  
data1 and data2; /*512 byte/record*/
```

```
Sort by row1 & row2;
```

```
Select * where row1 = „ABCD“;
```

```
end = system time;
```

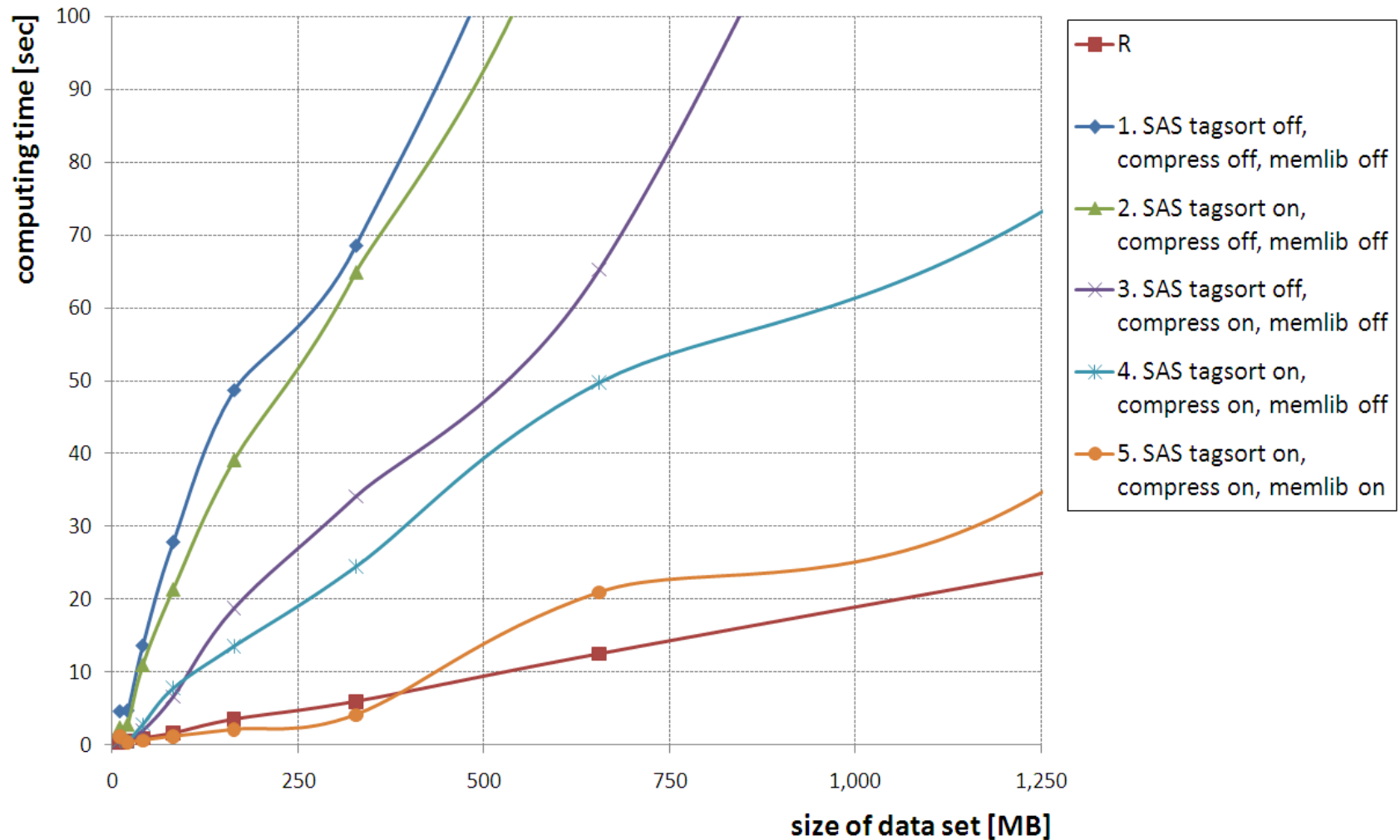
```
duration = end - start;
```

SAS Options

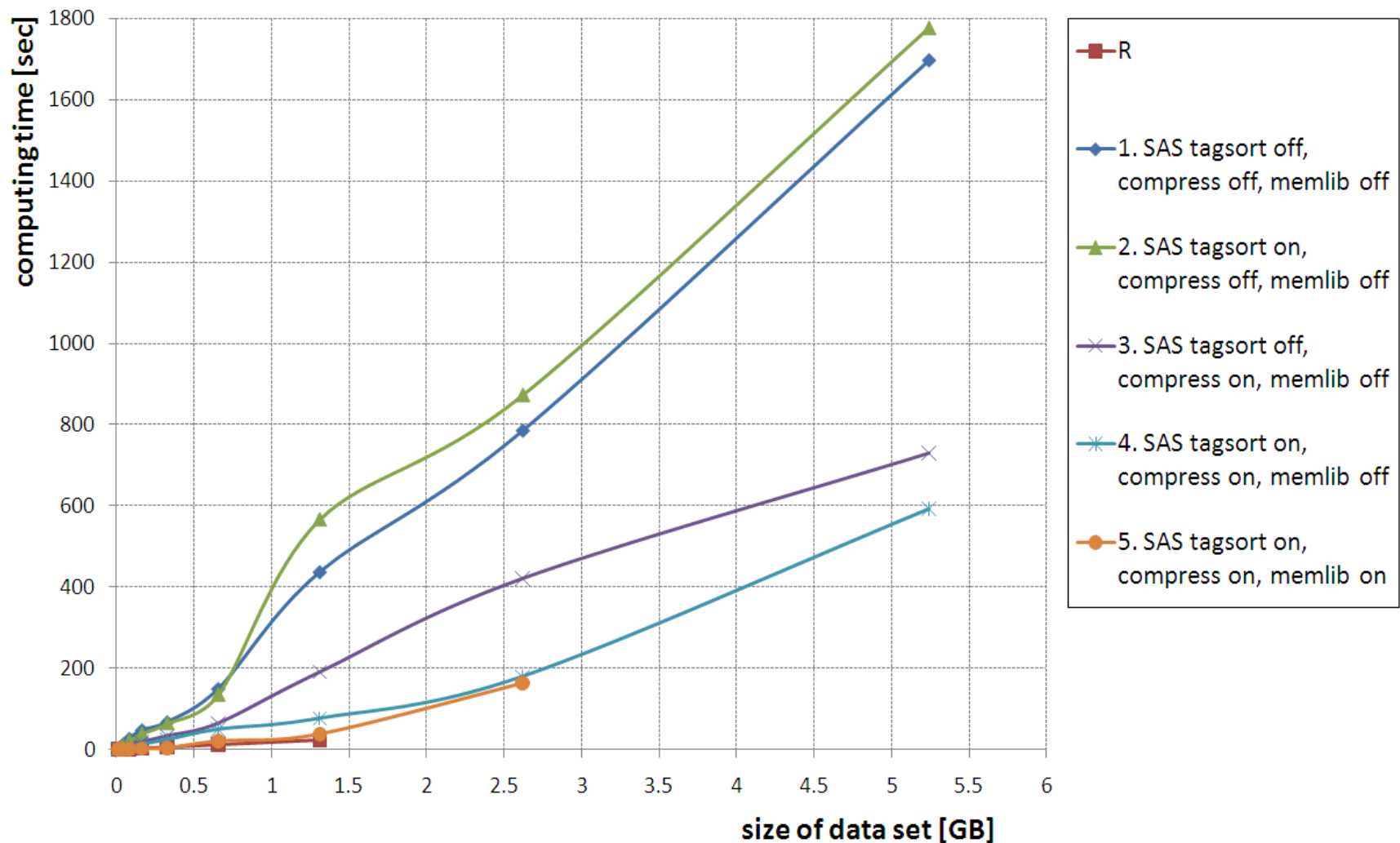
- **tagsort:**
 - stores only the BY variables and the observation numbers in temporary files
 - requires less temporary storage space and memory
 - processing time is usually higher
- **compress:**
 - observations in a newly created data set are compressed (variable-length records) by using “Run Length Encoding” (RLE) (compresses observations by reducing repeated consecutive characters (including blanks) to two-byte or three-byte representations)
- **memlib**
 - library is processed in memory
 - files are kept in memory until SAS is terminated or the files are deleted.

1. SAS tagsort off, compress off, memlib off
2. SAS tagsort on, compress off, memlib off
3. SAS tagsort off, compress on, memlib off
4. SAS tagsort on, compress on, memlib off
5. SAS tagsort on, compress on, memlib on

Performance Comparison Using Large Data Sets

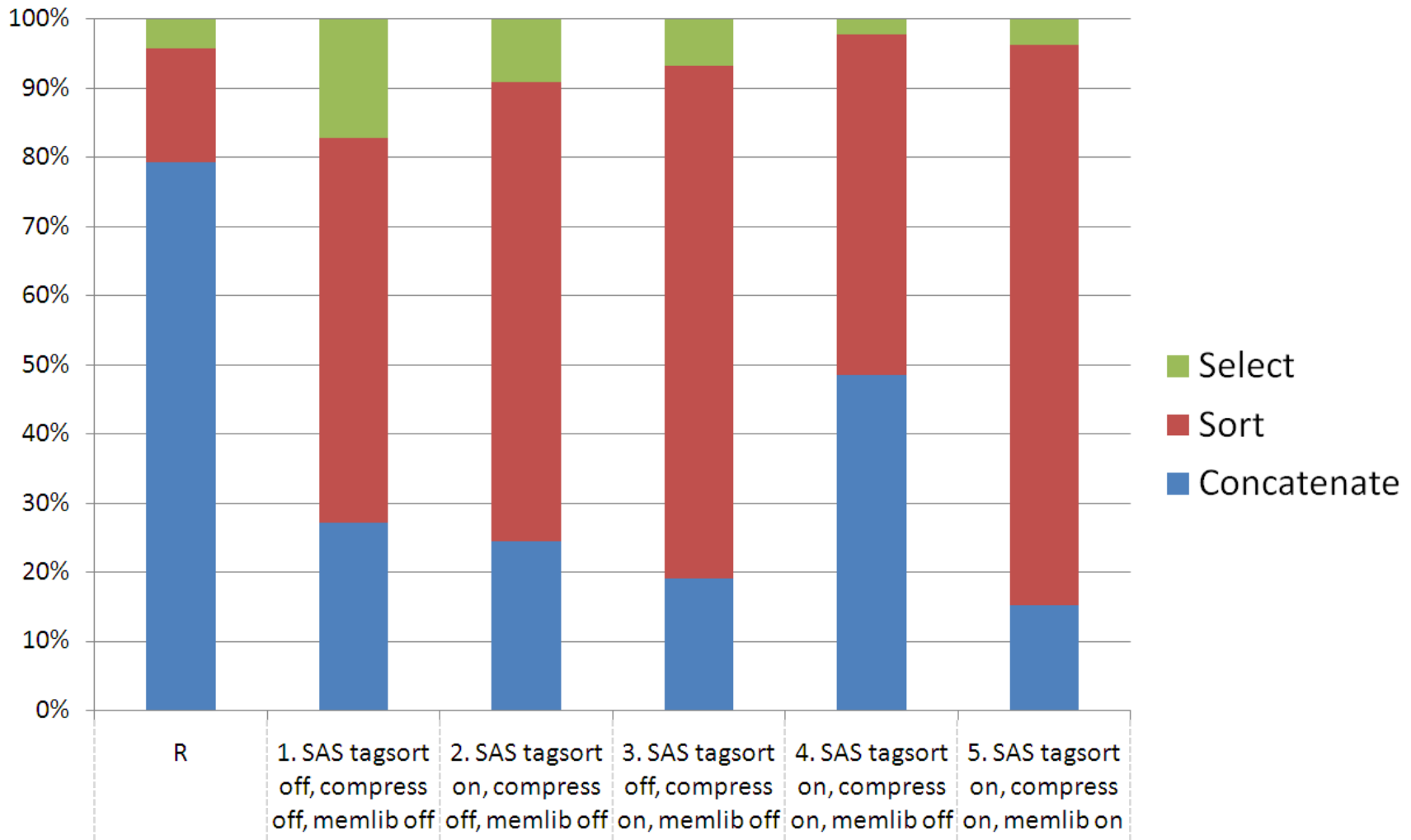


Performance Comparison Using Large Data Sets



Performance Comparison Using Large Data Sets– 1.3 GB

Computing Time in Percentage



Performance Comparison Using Large Data Sets

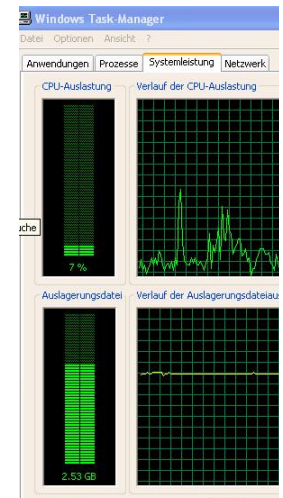
- Intel Core Duo CPU, 2.53 GHz, 2GB RAM
- SAS 9.2
- R 2.11.1

→ R is slightly faster

- data are loaded into the main memory
- limitation given by the size of the main memory

→ SAS processes large data sets

- reads and writes data to the hard disk for each step
- accordingly computation time is higher
- almost as fast as R by the use of memlib (data are loaded to the main memory - also limitation given by the size)



Interface SAS IML 9.22 to R

- IML must be installed and comes with an extra license
- SAS must be started with Option RLANG
- at least compatible with R Version R-2.10.1

```
PROC IML;
```

```
SUBMIT / R;
```

```
#R-Code
```

```
rx <- matrix( 1:3, nrow=1)
```

```
rm <- matrix( 1:9, nrow=3  
            , byrow=TRUE)
```

```
rq <- rm %*% t(rx)
```

```
print(rq)
```

```
ENDSUBMIT;
```

```
QUIT;
```

start-statement R-code

end-statement R-code

Interface SAS IML 9.22 to R

Data export from SAS to R:

Subroutine	SAS source	R target object
ExportDataSetToR	SAS data set	R data frame
ExportMatrixToR	SAS/IML matrix	R matrix

```

PROC IML;
  RUN ExportDataSetToR("Sashelp.Class", "df" );
  submit / R;
    names(df)
  endsubmit;
QUIT;

```

```

The SAS System      16:21 Thursday, February 24, 2011  17
[1] "Name"  "Sex"   "Age"   "Height" "Weight"

```

Data import from R to SAS:

Subroutine	R source	SAS target object
ImportDataSetFromR	R expression	SAS data set
ImportMatrixFromR	R expression	SAS/IML matrix

Interface from R to SAS

- no direct interface to SAS implemented
- calling SAS with system commands:

```
> system(
+       "C:\\Programme\\SAS\\SASFoundation\\9.2\\sas.exe
+       C:\\HMS\\Projekte\\KSFE2011\\ProcMeansExport.sas"
+     )|
```

- calling SAS-program:

```
PROC MEANS DATA = SAShelp.CLASS MEAN SUM;
  VAR Height;|
  CLASS Sex;
  OUTPUT OUT=Results;
RUN;
PROC EXPORT DATA= Results
  OUTFILE= "C:\\tmp\\Results.csv"
  DBMS=CSV REPLACE;
  PUTNAMES=YES;
RUN;
```

- Data import to R:

```
> sasData <- read.table(
+       file = "C:\\tmp\\Results.csv"
+       , sep = ",", header = TRUE )
```

```
> sasData
  Sex X_TYPE_ X_FREQ_ X_STAT_ Height
1    0      19      N 19.000000
2    0      19     MIN 51.300000
3    0      19     MAX 72.000000
4    0      19     MEAN 62.336842
5    0      19     STD  5.127075
```

Software Quality

SAS

- customer orientated
- installation / operational – qualification tool
- developer work together locally
- well defined test procedures

R

- developer orientated
- installation qualification-process could be improved (no operational qualification tool)
- developers are spread world wide
- tests of additional packages could be improved

Conclusion

- Focus was on the SAS language and R environment
 - SAS platform has many frontends, servers, meta-data administration, middleware-components
- Differences
 - history of origin
 - user requirements
 - performance using large data sets
 - installation and support
- Similarities
 - development environment
 - structure of recurring code
- Common
 - interface
- Personal impression
 - SAS comes along with a higher proof of quality
 - R packages implement the state of the art algorithms
 - R is well integrated into SAS IML 9.22
 - there is no general better or worse



Thank you for your attention

Dr. Peter Bowerunge

Software Engineer

HMS Analytical Software GmbH

Rohrbacher Str. 26 • 69115 Heidelberg

Telefon +49 6221 6051-0

Peter.Bowerunge@analytical-software.de

www.analytical-software.de