HPS
https://hps.vi4io.org/

Christopher L. Lübbers

Using R for High-Performance Data Analytics

Seminar Newest Trends in High-Performance Data Analytics

Outline

High-Performance Data Analytics

HPDA: Transforming Data into Insights

- Defining High-Performance Data Analytics
- Essential for Big Data Challenges
- Increasing Relevance in Various Industries

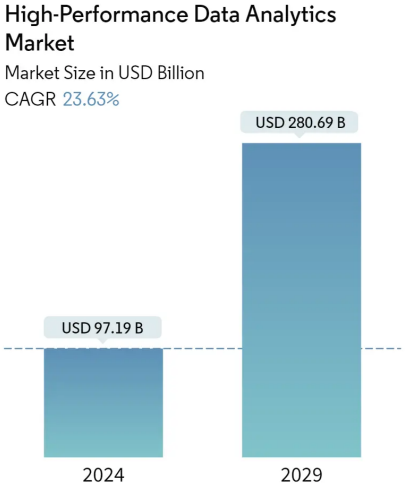
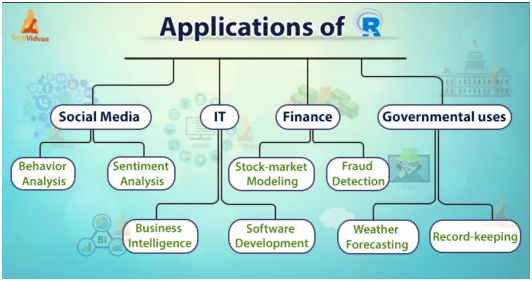


Image source: Mordor Intelligence Research and Advisory, *High Performance Data Analytics Market Size*

R's Role in Data Analytics Landscape

R is a Versatile Tool in Data Analytics

- Comprehensive Statistical Toolkit
- Community and Package Ecosystem
- Integration with Other Technologies



G. The R Foundation, *CRAN-GWDG*
Wendt and Anderson, "Ten simple rules for finding and selecting R packages"
Image source: TechVidvan, *R Applications – 9 Real-world Use Cases of R programming*

Outline

Introduction to Basic Applications of R

R's Foundational Role in Data Analytics

- R's Versatility in Data Analysis
- Fundamental Statistical Methods
- Ease of Data Visualization

The R Foundation, *R-Project*

Data Manipulation and Exploration in R

R's Power in Data Handling and EDA

- Data Cleaning Techniques
- EDA Tools in R
- Streamlining Data Analysis

1111Y1	63133.34	1881	185	10011	NNNNNN11111
1111N1	71130.04	2151	273	15851	NNYNNN11111
1111N1	64123.72	1381	138	10011	NNNNNN11111
1111N1	66135.55	2201	219	10011	NNNNNN11111
1111N1	67131.34	2001	247	14751	NNNNNN11111
1111N1	70123.02	1601	170	11011	NNNNNN11111
1111N1	67128.23	1801	209	12931	NNNNNN11111
1111N1	62124.72	1351	155	12021	NYNNNN11111
1111N1	69119.82	1341	164	13031	NNNNNN11111
1111N1	63125.73	1451	201	15651	NNNNNN11111
1111N1	67124.32	1551	205	15051	NNNNNN11111
1111N1	60126.43	1351	161	12631	NNNNNN11111
1111N1	58131.34	1501	180	13031	NNNNNN11111
1111N1	66127.43	1701	198	12831	NYNNNN11111
1111N1	66123.92	1481	182	13441	NNNNNN11111
1111N1	61134.04	1801	230	15051	NNNNNY11111
1111N1	62125.23	1381	161	12331	NNNNNN11111
1111N1	63125.73	1451	193	14851	NNNNNN11111
1111N1	66128.23	1751	201	12631	NNNNNN11111
1111N1	65125.53	1531	186	13341	NNNNNN11111
1111N1	60132.24	1651	183	11821	NNNNNN11111
1111N1	66129.03	1801	220	14041	NYNNYN11111
1111Y1	64121.52	1251	154	12931	NNNNNY11111
1111N1	62123.82	1301	175	14551	NNNNNN11111
1111N1	65124.12	1451	189	14451	NNNNNN11111

Data Source: Driscoll, "User Guide to the 2018 Natality Public Use File"

Dat

R

```r
library(readr) # import data
library(dplyr) # transform data
data <- read_fwf("Nat2018PublicUS.c20190509.r20190717.txt",
                 col_positions = fwf_cols(Month = c(13,14),
                    Ane = c(537,537), Men = c(538,538),
                    Cya = c(539,539), Her = c(540,540),
                    Omp = c(541,541), Gas = c(542,542),
                    Lim = c(549,549), Cle = c(550,550),
                    Pal = c(551,551), Dow = c(552,552),
                    Chr = c(553,553), Hyp = c(554,554)),
                    col_types = "iffffffffffff") %>%
  group_by(Month) %>%
  summarise(Ane = sum(Ane == "Y"), Men = sum(Men == "Y"),
            Cya = sum(Cya == "Y"), Her = sum(Her == "Y"),
            Omp = sum(Omp == "Y"), Gas = sum(Gas == "Y"),
            Lim = sum(Lim == "Y"), Cle = sum(Cle == "Y"),
            Pal = sum(Pal == "Y"), Dow = sum(Dow == "P"),
            Chr = sum(Chr == "P"), Hyp = sum(Hyp == "Y"))
```

Data

Table: Monthly counts of birth anomalies.

Month	Ane	Men	Cya	Her	Omp	Gas	Lim	Cle	Pal	Dow	Chr	Hyp
1	29	55	172	46	39	73	48	183	77	103	102	174
2	25	45	175	35	31	55	34	142	81	115	100	180
3	31	48	182	41	47	72	40	200	86	90	96	180
4	34	45	186	36	32	75	42	173	56	87	90	193
5	33	40	187	46	24	80	35	180	75	91	100	197
6	34	48	189	35	33	75	45	154	74	102	100	182
7	26	43	198	34	21	74	36	179	79	86	92	193
8	24	41	189	44	43	62	48	183	88	109	94	194
9	34	44	147	40	37	66	36	158	73	112	103	196
10	25	43	207	45	31	65	49	181	77	108	115	220
11	36	55	188	39	39	62	43	144	68	98	79	173
12	23	48	196	31	31	71	31	177	86	86	73	156

Driscoll, “User Guide to the 2018 Natality Public Use File”

Statistical Analysis and Hypothesis Testing

R as a Statistical Powerhouse

- Basic Tests Implementation
- Insightful Statistical Analysis
- Example: Linear Models

```
> lm_example <- lm(Month ~ Ane + Men + Cya, data=data)
> summary(lm_example)

Call:
lm(formula = Month ~ Ane + Men + Cya, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-4.3204 -2.8382 -0.6839  2.4580  5.1843

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -6.42556   23.22043  -0.277    0.789
Ane          0.01341    0.29111   0.046    0.964
Men         -0.02629    0.26175  -0.100    0.922
Cya          0.07444    0.08687   0.857    0.416

Residual standard error: 4.014 on 8 degrees of freedom
Multiple R-squared:  0.09846,   Adjusted R-squared:  -0.2
F-statistic: 0.2913 on 3 and 8 DF,  p-value: 0.8307
```

Data Visualization with R

R's Visualization Ecosystem: Converting Data into Insights

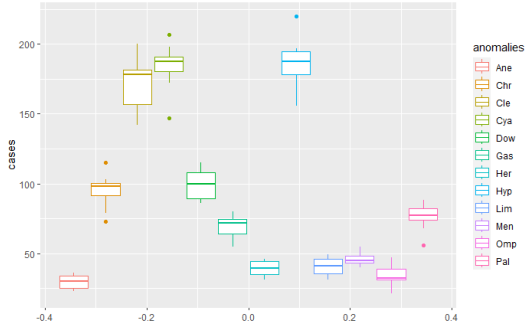
- ggplot2 and Beyond
- Visual Storytelling
- Transformative Visualization

Hadley Wickham, *ggplot2: Elegant Graphics for Data Analysis*

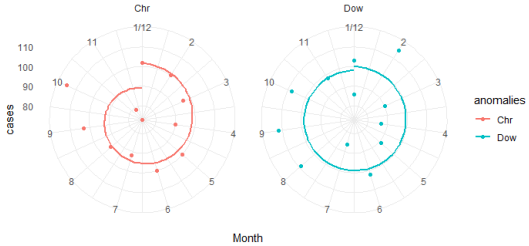
Visualization Example

```r
library(tidyr)
data_vis <- data %>%
  # make the table longer to visualize every birth anomaly
  pivot_longer(cols = 2:13, names_to="anomalies", values_to = "cases")

# plotting
library(ggplot2)
ggplot(data_vis, aes(y = cases, color = anomalies)) +
geom_boxplot()
```




```r
data_vis %>%
  filter(anomalies %in% c("Dow", "Chr")) %>%
ggplot(aes(x = Month, y = cases, color = anomalies)) +
  geom_point() +
  geom_smooth(method = "lm", se=FALSE) +
  coord_polar() +
  facet_grid(.~anomalies) +
  scale_x_continuous("Month", breaks = c(1,2,3,4,5,6,7,8,9,10,11,12)) +
  theme_minimal()
```



Interfacing R with Other Data Sources

R's Flexibility in Data Integration

- Database Interaction ("odbc", "DBI")
- Data Import/Export
- Format Compatibility

odbc: Jim Hester, *odbc: Connect to ODBC Compatible Databases (using the DBI Interface)*

Summary: Basic Applications of R

- Essential Data Manipulation
- Statistical Analysis Mastery
- Visual Storytelling
- Versatile Data Integration

Outline

Memory Management

Optimizing Performance through Effective Memory Management

- Inherent Memory Storage
- Efficient Data Structures with "data.table"
- Advanced Techniques with "bigMemory"
- Regular Cleanup and Efficiency Tools

Michael J. Kane, “Scalable Strategies for Computing with Massive Data”

Big Data

Harnessing Apache Spark with Sparklyr for Big Data

- R for Big Data: Leveraging Sparklyr
- Scalable Data Processing with Spark
- Integrating R and Apache Spark

Kalinowski, *Posit AI Blog: Deep Learning with R, 2nd Edition*
Pilli, "Performance Improvement and Reporting Techniques Using SparklyR and H2O.Ai"
Image source: Kalinowski, *SparklyR Documentation*

CPU vs. GPU in R: A Comparative Analysis

Advancements in R through GPU Computing

- GPU Computing in R Transforms Computational Speed
- integrating with TensorFlow
- Case Study: clrng Package Performance Boost

Data

Table: Monthly counts of birth anomalies.

	Ane	Men	Cya	Her	Omp	Gas	Lim	Cle	Pal	Dow	Chr	Hyp
Jan	29	55	172	46	39	73	48	183	77	103	102	174
Feb	25	45	175	35	31	55	34	142	81	115	100	180
Mar	31	48	182	41	47	72	40	200	86	90	96	180
Apr	34	45	186	36	32	75	42	173	56	87	90	193
May	33	40	187	46	24	80	35	180	75	91	100	197
Jun	34	48	189	35	33	75	45	154	74	102	100	182
Jul	26	43	198	34	21	74	36	179	79	86	92	193
Aug	24	41	189	44	43	62	48	183	88	109	94	194
Sep	34	44	147	40	37	66	36	158	73	112	103	196
Oct	25	43	207	45	31	65	49	181	77	108	115	220
Nov	36	55	188	39	39	62	43	144	68	98	79	173
Dec	23	48	196	31	31	71	31	177	86	86	73	156

Driscoll, “User Guide to the 2018 Natality Public Use File”

CPU Computing

Fisher's test simulation on CPU in R

R

```r
time_cpu <- system.time(result_cpu <- fisher.test(month,
                                                  simulate.p.value = TRUE,
                                                  B = 1015808))
```

- runtime = 49.3
- p-value = 0.403804

Xu, "Statistical Computing With Graphics Processing Units"

GPU Computing

Fisher's test simulation with clrng on GPU in R

R

```r
library(clrng)
streams <- createStreamsGpu(n = 256*64)
month_gpu <- vclMatrix(month, type = "integer")
time_gpu <- system.time(result_gpu <- clrng::fisher.sim(month_gpu, 1e6,
                                                        streams=streams, type="double",
                                                        returnStatistics=TRUE,
                                                        Nglobal = c(256,64)))
```

- runtime = 2.2
- p-value = 0.403507

Xu, "Statistical Computing With Graphics Processing Units"

Comparison

Table: Comparison of Fisher's test simulation on different Devices.

Device	runtime	p-value
Intel 2.5 ghz	49.3	0.403804
AMD Radeon VIII	2.2	0.403507

Advanced Use of R - Key Takeaways

Advanced R Techniques

- Effective Memory Management with BigMemory
- Big Data handling with SparklyR
- GPU Computing for Speed

Outline

Introduction to Parallel Computing in R

Harnessing the Power of Parallelism in R

- Overview of Parallel Computing
- Importance in High-Performance Data Analytics
- R's capabilities for Parallelization

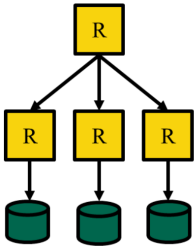
McCallum and Weston, *Parallel R*

Parallelization Techniques in R

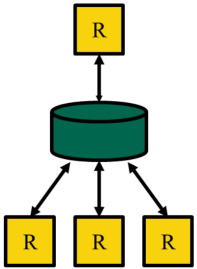
Diverse Techniques for Efficient Parallel Computing in R

- Embarrassingly Parallel Tasks
- Worker Queues and Task Distribution
- Shared Memory Parallelization
- Message Passing in Distributed Computing

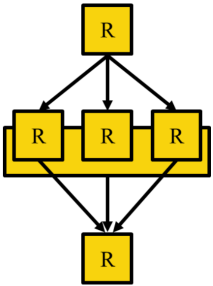
Parallelization schema



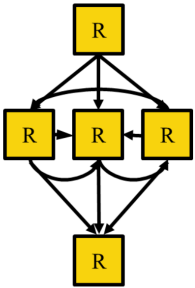
Embarrassingly/Pleasingly Parallel



Shared file system or database



Shared memory



Message Passing

Image source: *Using R at LRZ*

R Packages for Parallel Computing

Leveraging Specialized R Packages for Efficient Parallel Computing

- Utilizing "foreach" and "doParallel"
- the "parallel" Package in R
- Advanced Parallelization with "future" and "promises"

Folashade Daniel, *foreach: Provides Foreach Looping Construct*
Bengtsson, “A Unifying Framework for Parallel and Distributed Processing in R using Futures”

Use Case

Mostly biological, pharmaceutical, and gene Data

- > 20.000 annotated genes of fruit fly
- calculation of correlation coefficients
- parallelization with "doParallel" and "foreach"

Metah, “A Parallel Computing Approach for Identifying Retinitis Pigmentosa Modifiers in Drosophila Using Eye Size and Gene Expression Data”

Benchmark

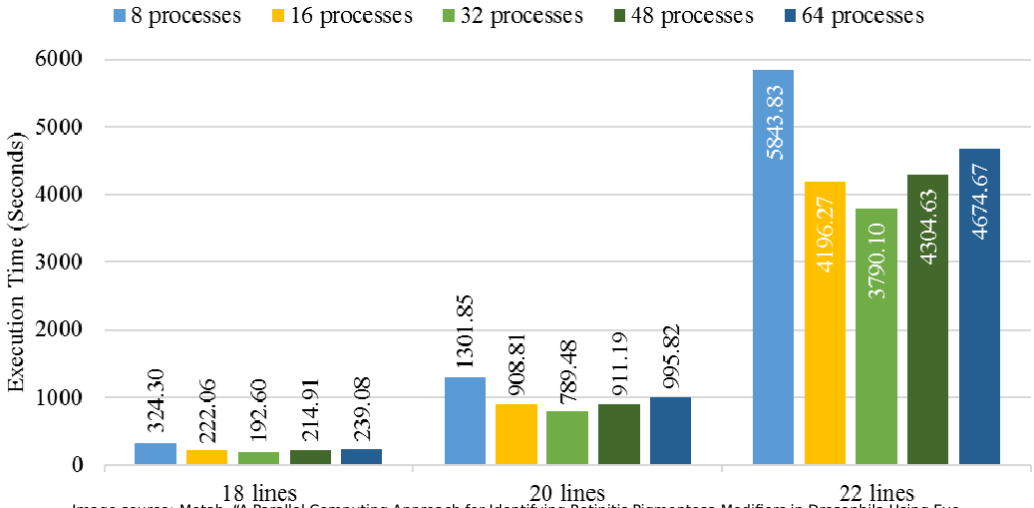


Image source: Metah, "A Parallel Computing Approach for Identifying Retinitis Pigmentosa Modifiers in Drosophila Using Eye Size and Gene Expression Data"

Key Takeaways from "R and Parallel Computing

- Parallel Computing: Enhancing R's Data Processing
- Techniques: Embarrassingly Parallel, Worker Queues, Message Passing
- R Packages: 'foreach', 'doParallel', 'parallel', 'future', 'promises'

Outline

Summary

The Power of R in High-Performance Data Analytics

- Comprehensive Capabilities
- Parallel Computing Power
- Continuous Advancements
- for further information: CRAN Task View

CRAN Task View: Eddelbuettel, *CRAN Task View: High-Performance and Parallel Computing with R*

References I

AI Training Series: High Performance Data Analytics. Leibniz Supercomputing Centre, May 2023. (Visited on 01/11/2024).

Bengtsson, Henrik. "A Unifying Framework for Parallel and Distributed Processing in R using Futures". In: *The R Journal* 13.2 (2021), pp. 208–227. DOI: 10.32614/RJ-2021-048. URL: https://doi.org/10.32614/RJ-2021-048.

CC BY SA Posit Software. *TensorFlow for R*. 2023. URL: https://github.com/rstudio/tensorflow (visited on 01/15/2024).

Driscoll, Anne (CDC/OPHSS/NCHS). "User Guide to the 2018 Natality Public Use File". In: (). URL: http://www.cdc.gov/nchs/data_access/VitalStatsOnline.htm.

Eddelbuettel, Dirk. *CRAN Task View: High-Performance and Parallel Computing with R*. 2023. URL: https://cran.r-project.org/web/views/HighPerformanceComputing.html (visited on 12/17/2023).

— .“Rcpp: Seamless R and C++ Integration”. In: *Journal of Statistical Software* 40.8 (2011), pp. 1–18. DOI: 10.18637/jss.v040.i08.

Folashade Daniel, Steve Weston. *foreach: Provides Foreach Looping Construct*. 2023. URL: https://github.com/RevolutionAnalytics/foreach (visited on 01/15/2024).

Jim Hester, Hadley Wickham. *odbc: Connect to ODBC Compatible Databases (using the DBI Interface)*. 2023. URL: https://github.com/r-dbi/odbc (visited on 01/15/2024).

References II

Kalinowski, Tomasz. *Posit AI Blog: Deep Learning with R, 2nd Edition*. 2022. URL: https://blogs.rstudio.com/tensorflow/posts/2022-05-31-deep-learning-with-R-2e/.

— .*SparklyR Documentation*. 2023. URL: https://spark.rstudio.com/deployment/data-lakes.html.

McCallum, Ethan and Stephen Weston. *Parallel R*. " O'Reilly Media, Inc.", 2011.

Metah, Chawin. "A Parallel Computing Approach for Identifying Retinitis Pigmentosa Modifiers in Drosophila Using Eye Size and Gene Expression Data". PhD thesis. Purdue University, 2023.

Michael J. Kane John W. Emerson, Stephen Weston. "Scalable Strategies for Computing with Massive Data". In: *Journal of Statistical Software* 55.14 (2013), pp. 1–19. URL: https://www.jstatsoft.org/article/view/v055i14.

Mordor Intelligence Research and Advisory. *High Performance Data Analytics Market Size*. 2024. URL: https://www.mordorintelligence.com/industry-reports/high-performance-data-analytics-market (visited on 01/15/2024).

Pilli, Happy Justin. "Performance Improvement and Reporting Techniques Using SparklyR and H2O.Ai". In: ().

TechVidvan. *R Applications – 9 Real-world Use Cases of R programming*. URL: https://techvidvan.com/tutorials/r-applications/ (visited on 01/15/2024).

The R Foundation. *R-Project*. 2023. URL: https://www.r-project.org/about.html (visited on 01/15/2024).

The R Foundation, GWDG. *CRAN-GWDG*. 2023. URL: https://ftp.gwdg.de/pub/misc/cran/ (visited on 01/15/2024).

References III

Wendt, Caroline J and G. Brooke Anderson. “Ten simple rules for finding and selecting R packages”. In: *PLoS Computational Biology* 18 (2022). URL: https://api.semanticscholar.org/CorpusID:247675731.

Wickham, H. *Advanced R, Second Edition*. Chapman & Hall/CRC the R Series. CRC Press, 2019. ISBN: 978-1-351-20130-8.

Wickham, Hadley. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN: 978-3-319-24277-4. URL: https://ggplot2.tidyverse.org.

Xu, Ruoyong. “Statistical Computing With Graphics Processing Units”. In: ().

Appendix

Benchmarks

- profiling with "Rprof"
- visualization with "profvis"
- microbenchmarks with "bench"

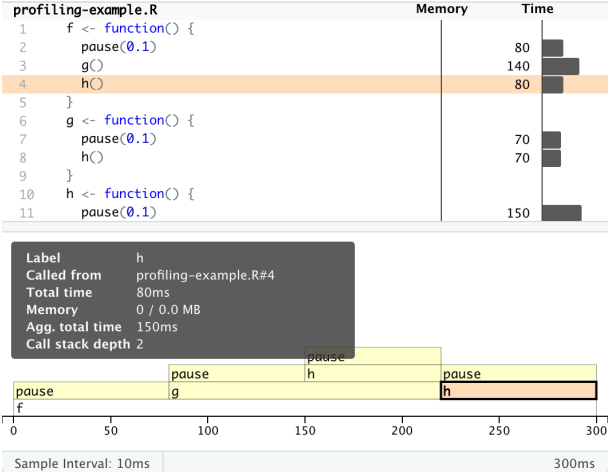


Image source: H. Wickham, *Advanced R, Second Edition*

Rewriting R code in C++

When everything else is not fast enough: Rcpp

- typical bottlenecks in R:
  - loops
  - recursive functions
  - advanced data structures
- write your code in C++!

Eddelbuettel, “Rcpp: Seamless R and C++ Integration”

Rcpp

cppFunction() allows you to write C++ functions in R

R

```r
cppFunction('int add(int x, int y, int z) {
  int sum = x + y + z;
  return sum;
}')
# add works like a regular R function
add
#> function (x, y, z)
#> .Call(<pointer: 0x107536a00>, x, y, z)
add(1, 2, 3)
#> [1] 6
```

H. Wickham, *Advanced R, Second Edition*