Introduction to R

PD Dr. Daniel Wollschläger
wollschlaeger@uni-mainz.de

MSE CM 2.2

2021-01-21

Universitätsmedizin Mainz, IMBEI

Topics

- What is R?
- R pros and cons
- What working with R looks like
- Use R with the RStudio Development Environment
  - Get help / documentation
  - Work with contributed packages
  - Read and store data
  - Explore / transform / aggregate data
  - Diagrams
  - Statistical tests & regression models

Showcase

- Radiation dose to the heart in 3D conformal tangential breast cancer radiotherapy
- 3D Raytracing Diagrams
- Rhineland-Palatinate Mortality Monitoring Dashboard

What is R?

What is R? www.r-project.org

- ~~Statistics program~~
- Free statistical environment for interactive use
- (Scripting programming language)
- Base R = GNU R interactive console + base packages
  - vs. Integrated development environment (IDE)
  - vs. Microsoft R, ...
  - vs. Contributed extension packages

Pros: Why use R?

- **It works** (Windows, MacOS, Linux)
- Widely used: biometry, bioinformatics, data science
  - Active development & growing community
- Free & open source
  - No cost / no licensing issues → use anywhere
- Powerful extension mechanism: Packages
  - Fast moving innovation
- Very good support for reproducible data analysis
- Platform beyond traditional statistical analysis

Cons: What's not that great about R

- First steps easy but hard to master
  - Use it or lose it `r4stats.com/articles/why-r-is-hard-to-learn/`
- Organic growth → Inconsistent naming scheme
  - `read.table()`, `seq_int()`, `TukeyHSD()`, `trimws()`
- Built-in documentation very terse, technical
- There's more than one way to do it
  - Base R solutions vs. contributed packages
  - Inconsistent approaches & confusion
- Non-academic sector: maybe less mainstream than SAS
- Performance issues with very large datasets

Switching from SAS – key differences

- Real programming language vs. MACROs
- More low-level syntax (LEGO vs. FisherPrice)
- No restriction to working with just 1 data set
- Different philosophy – default output: little vs. much
- Run R from SAS/IML: `submit/r; … endsubmit;`

Should you switch to R?

- Results matter
  - Tools matter only insofar as they enable results
- Do you have to? (SAS not available ...)
- Can you afford to invest time?
  - ...and keep doing so?
- Do you have a concrete project / task?
- Do you have an R guru next door?
- Will there be interference with learning SAS?

Current developments

- Base R: stable, improvements under the hood
- Dynamically growing ecosystem around base R
  - Books, conferences, user groups, online courses
  - Contributed packages – long-term stability issues
- Hot topics
  - Machine learning
  - Interactive web applications
  - Reproducible analysis & reports

Documentation: Systematic

- Integrated help system with executable examples
- Official introduction and FAQs `www.r-project.org`
- Books
  - `link.springer.com/book/10.1007/978-3-662-61736-6`
  - `link.springer.com/book/10.1007/978-3-662-49102-7`
  - `www.routledge.com/Chapman--HallCRC-The-R-Series/book-series/CRCTHERSER`
- Online courses
  - `https://www.edx.org/course/statistics-and-r`
- Cheat sheets `www.rstudio.com/resources/cheatsheets/`

Learn R: Unsystematic documentation

- Google `"R"`
- ~~Email-lists~~ (low signal-to-noise ratio) : `r-project.org`
- Q&A: `stackoverflow.com/tags/R`
- Q&A: `stats.stackexchange.com/tags/R`
- Twitter: `#rstats`

Beware

Quality issues – wrong / outdated info?

Using R in RStudio

Run R

- Get R from `cloud.r-project.org/`
- Get a free integrated development environment (IDE)
  - **RStudio** ←
    `www.rstudio.com/products/rstudio/download/`
    Supported: Windows, MacOS, Linux
- ~~Graphical user interfaces~~ (limited functionality)
  - Rcmdr (R Commander)
  - Jamovi `www.jamovi.org`

Start

- R session: Working directory
- Use the integrated help system
- Objects
  - Create, show & remove objects
  - Classes: vector, matrix & data frame
  - Types: character, numerical, logical, factor, date / time
- Arithmetic
- Logic: Operators → associativity, ( )
- Numerical precision – FAQ 7.31
- Functions: Arguments & return values

Start – Vocabulary

- `getwd()`, `setwd()`
- `?⟨function⟩`, `??⟨word⟩`, `help()`, `help.start()`, `example()`
- `?Reserved`, `<-`, `=`
- `print()`
- `ls()`, `rm()`
- `?Arithmetic`
- `round(⟨number⟩, digits=⟨digits⟩)`
- `exp()`, `factorial()`
- `TRUE`, `FALSE`, `!`, `==`, `!=`, `<`, `>`, `<=`, `>=`, `|`, `&`, `xor()`

Hands on – run R within RStudio

- Do some arithmetic
- Does $1 - 49\frac{1}{49} = 0$ hold?
- Store an integer each in objects a, b, c
- Is 1 the same as `"1"`? Why?
- What is function `cv.glm()` about? (no Google!)
- What arguments can be supplied to function `mean`?
- Run the examples for `var()`
  - What is going on – how is this all related to `var()`?

Frequent sources of confusion & frustration

- R is case sensitive – objects, functions, arguments
- Mis-spelled objects, functions, arguments
- ( ) vs. [ ] vs. { } – always close when open
- Whitespace is often meaningless – but not always
- Handling of missing values – NA (not available)
- Handling of categorical variables (factors)
- Execution order (associativity) → use ( ) liberally
- Undocumented code → use comments # liberally

Using R packages

R PACKAGES

- Extend base R functionality for a specific purpose
  - Specialized statistical analysis
  - Tools for specialized data
  - More convenient solutions than base R
- Bundle new functions, data sets and documentation
- Contributed by independent developers
- Have dependency management

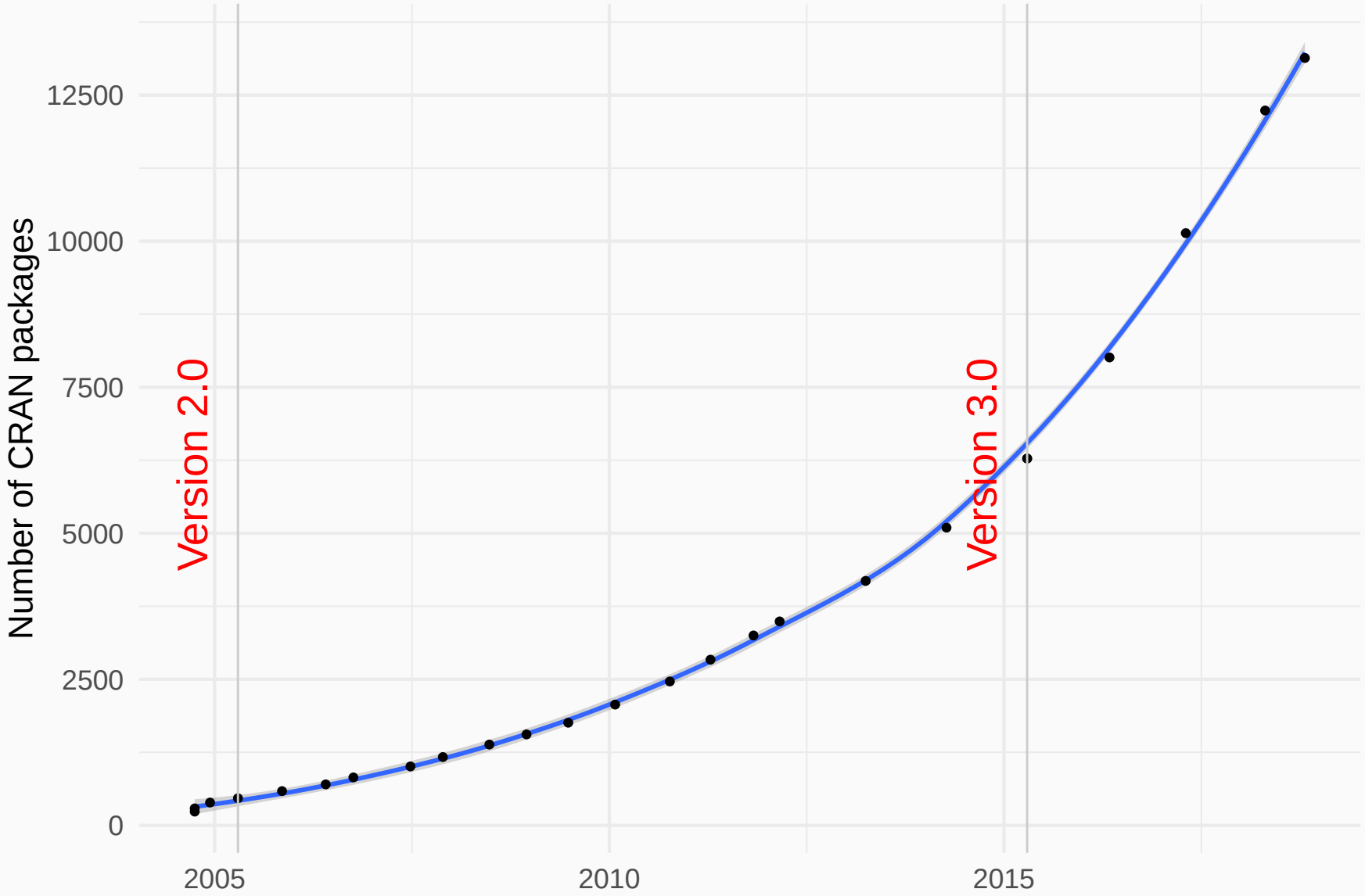
Using packages

1. Install – once for every major R version
2. Load – each session
3. Use like any base R functionality

R package repositories

- CRAN
  - Curated official package repository network
  - Some quality assurance for submitted packages
    - Can be installed, run, are documented, not malicious
    - **But:** No guarantee for working correctly
  - Task Views `cran.r-project.org/web/views/`
- GitHub
  - Not curated, no QA, bleeding edge

Growth of CRAN package repository



http://blog.revolutionanalytics.com/2016/04/cran-package-growth.html

Using packages – Vocabulary

- `install.packages("⟨package name⟩")`
- `installed.packages()`, `.libPaths()`
- `update.packages()`
- `library(⟨package name⟩)`
- `sessionInfo()`
- `help(package="⟨package name⟩")`
- `data()`
- `vignette(package="⟨package name⟩")`,
- `vignette("⟨topic⟩")`,
- `citation("⟨package name⟩")`

Hands on

- Load package `dpylr`
- What help topics are documented in package `ggplot2`?
- Where are your packages stored on disk?
- What packages are currently active in your session?
- Look at the `dpylr` vignette Introduction to dplyr
- Run the examples for `filter()` from package `dplyr`
- Check if there are updates for your installed packages
    - But don't install them

Explore data – Vocabulary

- `dim()`, `nrow()`, `ncol()`
- `head()`, `tail()`, `names()`, `View()`
- `str()`, `summary()`
- `[ , ]`, `drop=FALSE`, `$`

Read & store data

- Import → clean → transform ↔ explore
  - $\sim$ 80% of the work, then → model
- Import data
  - Files, Clipboard, URL
  - Plain text file: Comma-separated, tab-delimited, ...
  - R format file
  - SAS / Stata / SPSS file: package `haven`
  - Spreadsheet (Excel): package `readxl`
  - Database: `RSQLite`, `RPostgreSQL`, `RMySQL`, ...
- Packages `readxl`, `openxlsx`

Read & store data – Vocabulary

- `read.table()`, `write.table()`
- `file="clipboard"`, `file=pipe("pbpaste")`, `file=url()`
- `stringsAsFactors=FALSE`
- `load()`, `save()`
- Package `haven`
  - `read_sas()`, `read_spss()`
  - `write_sas()`, `write_sav()`

Clean & correct data

- Variable names (coerced to legal names)
- Date & time formatting
- Names
  - Umlauts, hyphens, whitespace, capitalization
  - Given names, family names – which is which
- Measurement units
- Missing data coding
- Plausibility: Valid dates / categories / values, date logic,

  ...

Transform data

- Select subsets
  - Cases
  - Variables
  - Remove duplicates
- Change & create new variables
  - Recode variables
  - Cut continuous variables into categories
  - Calculate new based on old variables
  - Generate sequences
  - Simulate using random numbers
- Sort cases & variables

Transform data – Vocabulary

- Package `dplyr`
  - `filter()`
  - `select()`, `everything()`
  - `mutate()`, `rename()`, `if_else()`
  - `arrange()`, `desc()`
- `unique()`, `na.omit()`
- `scale()`, `cut()`, `as.Date()`, `strptime()`,
- `:`, `seq()`, `rep()`
- `sample()`, `runif()`, `rnorm()`

Read & store data – Hands on

- Read text file
  - `http://dwoll.de/dat_passos.csv`
- How large is the data set?
- What variables of what kind are there?
- Take a look at the first / last 10 observations
- Save the data set to an R file
- Save the data set as a tab-separated ASCII file

Transform data – Hands on

- Check and transform the PASSOS data set
  - Sort according to tumor side and age
  - Subset: Only women age ≥ 60 with left-sided tumor
  - Re-order variables: observed comes 1st, metric 2nd
  - Build age-groups (10 years)
  - Do a median split for BMI
  - Check that age is consistent with DOB and RT start
  - Center `observed`
  - Add normal random variable

Summarise data

- Descriptive statistics
- Group wise operations
- Frequency tables

Summarise data – Vocabulary

- Package `dplyr`
  - `group_by()`
  - `summarise()`, `n()`
- `sum()`, `min()`, `max()`, `range()`, `diff()`, `quantile()`
- `mean()`, `median()`, `sd()`, `var()`, `IQR()`, `modeest::mlv()`
- `DescTools::Skew()`, `DescTools::Kurt()`
- `cov()`, `cor()`
- `xtabs()`, `table()`
- `prop.table()`, `addmargins()`

Diagrams

- Books
  - R Graphics Cookbook `www.r-graphics.org`
  - Graphical data analysis with R `www.gradaanwr.net`
  - Datenvisualisierung mit R `www.datendesign-r.de`
- Base graphics
- Package `ggplot2` ←
- 2D interactive graphics
  - `www.htmlwidgets.org/showcase_plotly.html`
  - `gallery.htmlwidgets.org/`
- 3D interactive graphics: Package `rgl`

Diagrams – Vocabulary

- ggplot2
  - ggplot(), aes()
  - group=, linetype=, shape=, color=, fill=
  - geom_point(), geom_line(), geom_bar()
  - geom_histogram(), geom_boxplot()
  - geom_smooth(), geom_abline(), geom_text()
  - position=position_jitter(), position_dodge()
  - ggtitle(), xlab(), ylab()
  - facet_grid(), facet_wrap()
- ggsave()
- pdf(), jpeg(), dev.off()

Data containers

- Vectors
    - Recycling
- Factors
- Data frames
- Lists

Data containers – Vocabulary

- `c()`, `numeric()`, `character()`, `logical()`
- `LETTERS`, `letters`
- `[]`, `length()`
- `which()`, `%in%`
- `factor()`, `ordered()`
- `nlevels()`, `levels()`, `droplevels()`, `interaction()`
- `data.frame()`, `list()`
- `cbind()`, `rbind()`, `order()`
- `lapply()`, `sapply()`

Specific data types

- Missing values
- Character strings
- Date & time

Specific data types – Vocabulary

- Missing values
  - `NA`, `is.na()`, `anyNA()`, `na.omit()`
  - `na.rm=TRUE`, `use="pairwise"`, `use="complete"`
  - Multiple imputation: Package `mice`
- Character strings
  - `nchar()`, `trimws()`, `tolower()`, `toupper()`
  - `paste()`, `paste0()`, `sprintf()`
  - `grepl()`, `gsub()`, `glob2rx()`
  - Package `stringr`
- Date & time
  - `Sys.date()`, `as.Date()`, `strptime()`
  - Package `lubridate`

Advanced data transformations

- Split
- Combine
- Merge
- Transform between wide $\leftrightarrow$ long format

Advanced data transformations – Vocabulary

- `split()`, `lapply()`
- `dplyr`
  - `bind_rows()`, `bind_cols()`
  - `left_join()`
- `tidyr`
  - `gather()`, `spread()`
  - `separate()`, `unite()`

Statistical models

- Linear regression (OLS)
  - Goodness of fit
  - Diagnostics
  - Variable selection
- *t*-test, ANOVA, ANCOVA
- GLM: Logistic, Poisson regression
- Goodness-of-fit tests
- Independence tests

Statistical models – Vocabulary

- `lm()`, `summary()`, `anova()`, `step()`, `add1()`, `drop1()`
- `coef()`, `confint()`, `fitted()`, `residuals()`, `rstandard()`
- Package `car`
  - `vif()`, `residualPlots()`, `qqPlot()`
  - `spreadLevelPlot()`, `influenceIndexPlot()`
- `t.test()`, `alternative=`, `aov()`
- `glm()`, `MASS::polr()`
- `binom.test()`, `chisq.test()`, `fisher.test()`

Create documents & share results

- Books
  - Reproducible research with R & RStudio
    `christophergandrud.github.io/RepResR-RStudio/`
  - Dynamic Documents with R and knitr (Yihui Xie)
    `yihui.name/knitr/`
- Create reproducible reports – docx, pdf, html
  - `knitr`
  - `rmarkdown` `rmarkdown.rstudio.com`
- Tables `https://davidgohel.github.io/flextable/`
- Publication-ready diagrams

Create documents & share resuls – Vocabulary

- `knitr::kable()`, `flextable`, `huxtable`
- `scale_x_continuous()`, `scale_x_discrete(labels)`
- `coord_cartesian(xlim, ylim)`, `coord_fixed()`
- `theme()`, `guides()`, `annotate()`, `ggthemes`
- `ggtitle()`, `xlab()`, `ylab()`
- `scale_fill_grey()`, `scale_color_grey()`
- `scale_fill_brewer()`, `scale_fill_viridis()`
- `scale_color_discrete(name, labels)`
- `scale_shape_discrete(name, labels)`
- `cowplot::plot_grid()`
- `ggsave()`, `pdf()`, `tiff()`, `svg()`

- Books
  - Programmieren mit R (Uwe Ligges)
  - Advanced R: `https://adv-r.hadley.nz/` (Hadley Wickham)
  - R packages: `r-pkgs.had.co.nz` (Hadley Wickham)
- Control structures
  - Conditions
    - `if() { … } else { … }`
    - `switch() { … }`
  - Loops
    - `for() { … }`
    - `while() { … }`

- Function signature
  - Formal arguments and defaults
- Function body
  - Check actual arguments
  - Handle errors
  - Return value
  - Scope
- Generic functions
- Analyze functions
- Debugging