

- Home
 - R Programming Best Practices
 - R Online Guides
 - R Cheat Sheets
 - R Debugging
 - Videos and White Papers
 - Join us
 - Testimonials
 - Site Map
 - Contact
 - In The News
- Install & Learn R
 - Install R Software
 - Install R Packages and Load Libraries
 - What Is R and Why Learn R?
 - Learn R Programming
 - Run R Programs to Create Objects
 - R Process Flow and Scope
- Common R Tasks
- Compare with SAS
- Pharmaverse
- R Books and Blogs
- R Papers
- Common R FAQs



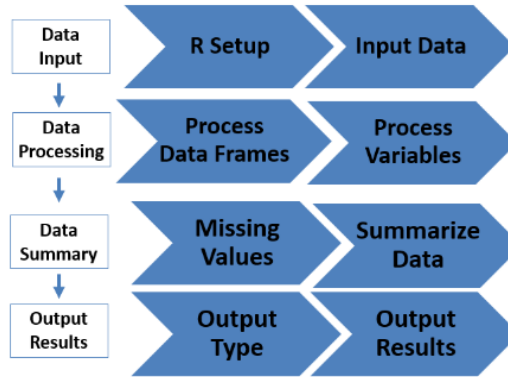
R-Guru Resource Hub

Search R-Guru

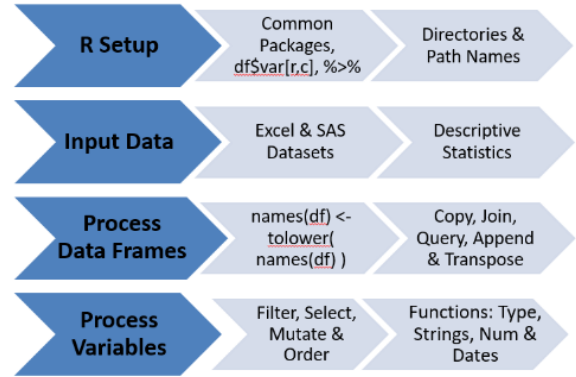
R_Guru <- R(for SAS Programmers)

Home Install & Learn R Common R Tasks Compare with SAS Pharmaverse

R Best Practices Checklist



R Best Practices Checklist for Data Input & Processing



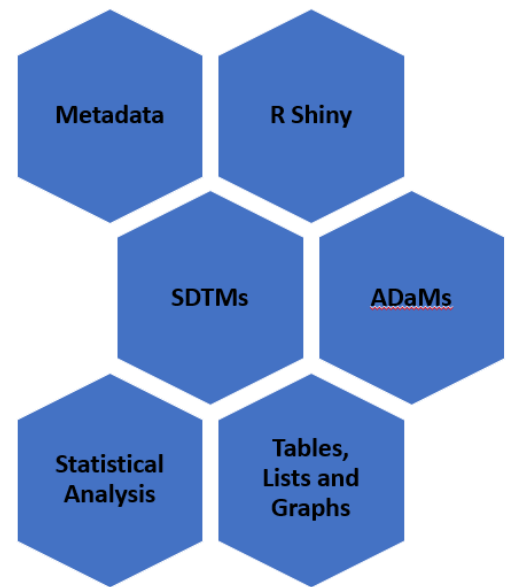
R Cheat Sheets

- R-Guru (All R Cheat Sheets)
- The Essential Functions of R
- Base R
- R Syntax Comparison
- R Packages
- R Reference Card
- R Studio IDE
- READR
- DPLYR
- STRINGR
- LUBRIDATE
- GT Summary
- RMARKDOWN
- GGLOT2
- Advanced R
- Tutorials Point Quick Guide
- The Analysis Factor Tutorials
- SAS 2 R
- Shiny App

R Programming Books and Blogs

- R Fundamentals
- Introduction to R Programming
- R Programming Examples
- R Programming Tasks
- Hands-On Programming with R Programming
- R Programming: Basic Operations
- R-Coder.com
- Advance R book
- The Epidemiologist R Handbook
- Introduction to Data Cleaning with R
- YaRrr! - The Pirates's Guide to R
- R for Clinical Study Reports and Submission
- Educative: R Tutorial for Beginners
- R for Data Science
- Introduction to Tidyverse
- Modern R with Tidyverse
- Tidyverse Blog
- Coding Club
- Mastering Shiny

Pharmaverse



R-Guru.com Cheat Sheet for Statistical Programmers

This guide contains best practice examples for creating and updating tibbles in the pharma industry. Examples show common R tasks for importing data, creating data frames, direct variable referencing, piping, conditional and group processing, sql components, character and date operations, variable type conversions, transposing data frames, joining data frames, appending data frames, deriving summary variables, and creating graphs and output files. When possible, base R sample data frames are used in examples.

`Mutate()` has five features: `case_when()`, simple expression, summary functions, `rowwise()`, and `group_by()/ungroup()` with summary functions. Data utility functions describe and view data frames: `View(df)`, `str(df)`, `summary(df)`, `table(vr)`, `print(df, n=)`, `head(df)`, `tail(df)`, `row_number()`, `nrow()`, `ncol()` and `ls()`. `Tidyverse`, `DPLYR`, `STRINGR`, `READR`, `READXL`, `HAVEN`, `Hmisc`, `LUBRIDATE`, `GT`, `GTSUMMARY` & `GGLOT2` packages are required. `df#` are data frame names and `vr#` are variable names. Character or numeric variables depend on the function and values. Missing and non-missing conditions: `==`, `!=`, `!=""`, `!=""`, `na.rm=TRUE`.

R-Guru Best Practices Mind Map

