R-Guru Resource Hub for Rapid R Learning

Sunil Gupta
GuptaProgramming@gmail.com

R-Guru.com
R Mentoring and Training

What does learning R mean to You?
https://r-guru.com/what-is-r

Keywords, Syntax, File & Variable References
One Line R Commands are Powerful
Show and Tell: Showcase R Flavors

In CSV file, Unique and then Sort by Column # 56

In ADSL, assign numeric values to SEXN variable based on SEX values

Create DM_EX by Left Joining DM and EX by USUBJID

R Programming is Not for Everyone
Programmer's Toolbox

R-Guru Webinars
- Unique Content
- Answers to be Productive

Very Technical

- Short-cut language since one-line R commands are concise

- Similar to SAS's advanced macros

- Syntax is not intuitive since need to remember keywords and syntax

Why Should You Learn R?

- Large Pharma are developing R packages
- Pharma and R Conferences
- CROs are providing R Training to better prepare their teams
- SAS Tools Integration with R
- CDISC recognizes R
- PhUSE SDE R Webinars
- FDA installed R
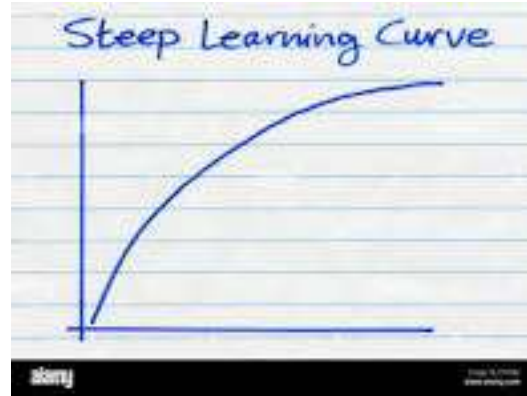- New R Programming Position Requirements

R-Guru Resource Hub for Rapid R Learning

Agenda

- Avoid the Steep Learning Curve

- Apply R Best Practices

- Leverage R Cheat Sheets

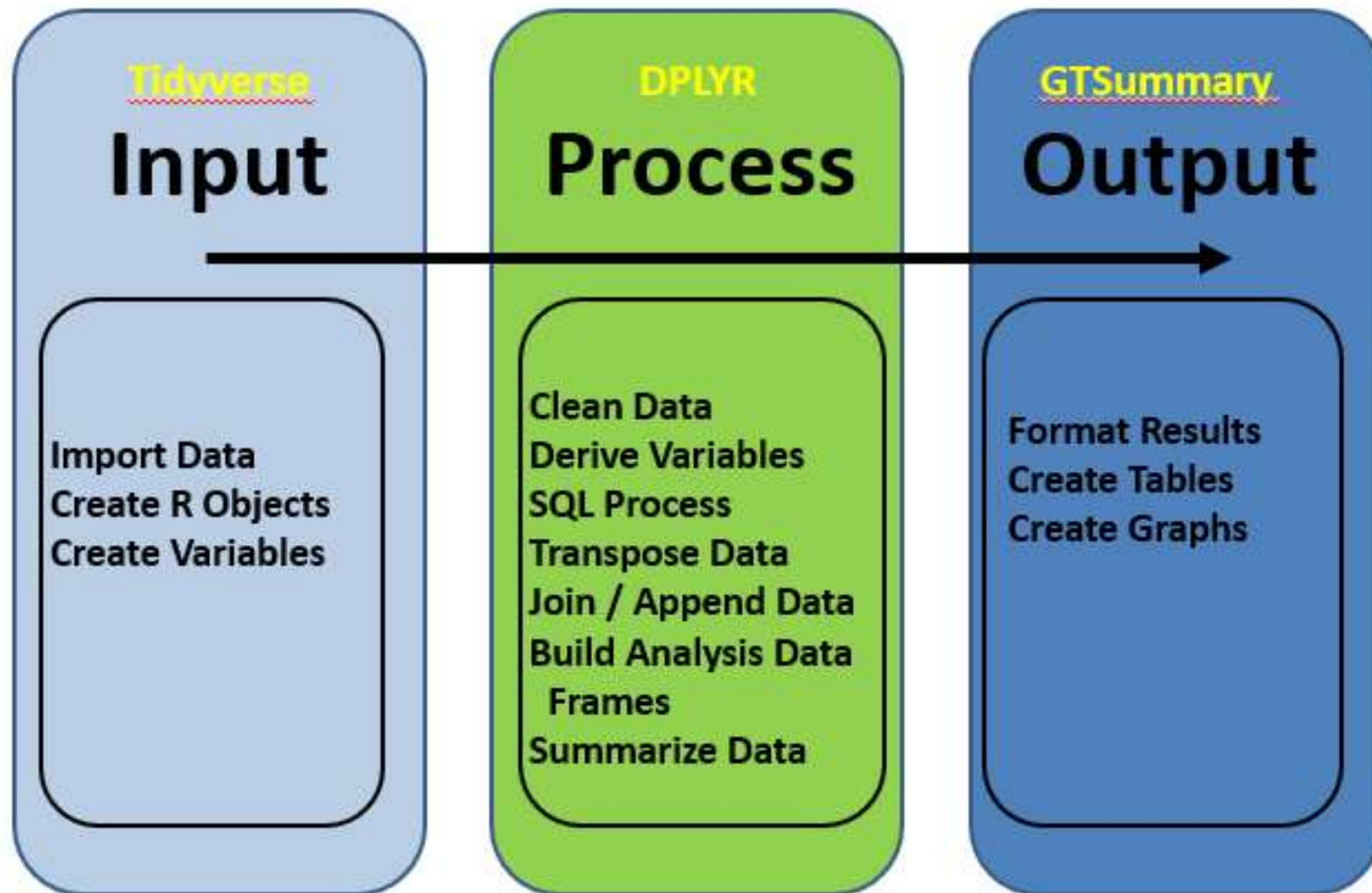- Learn Pharmaverse R packages

- Read on-line R books and blogs

How can You be Ready for R?
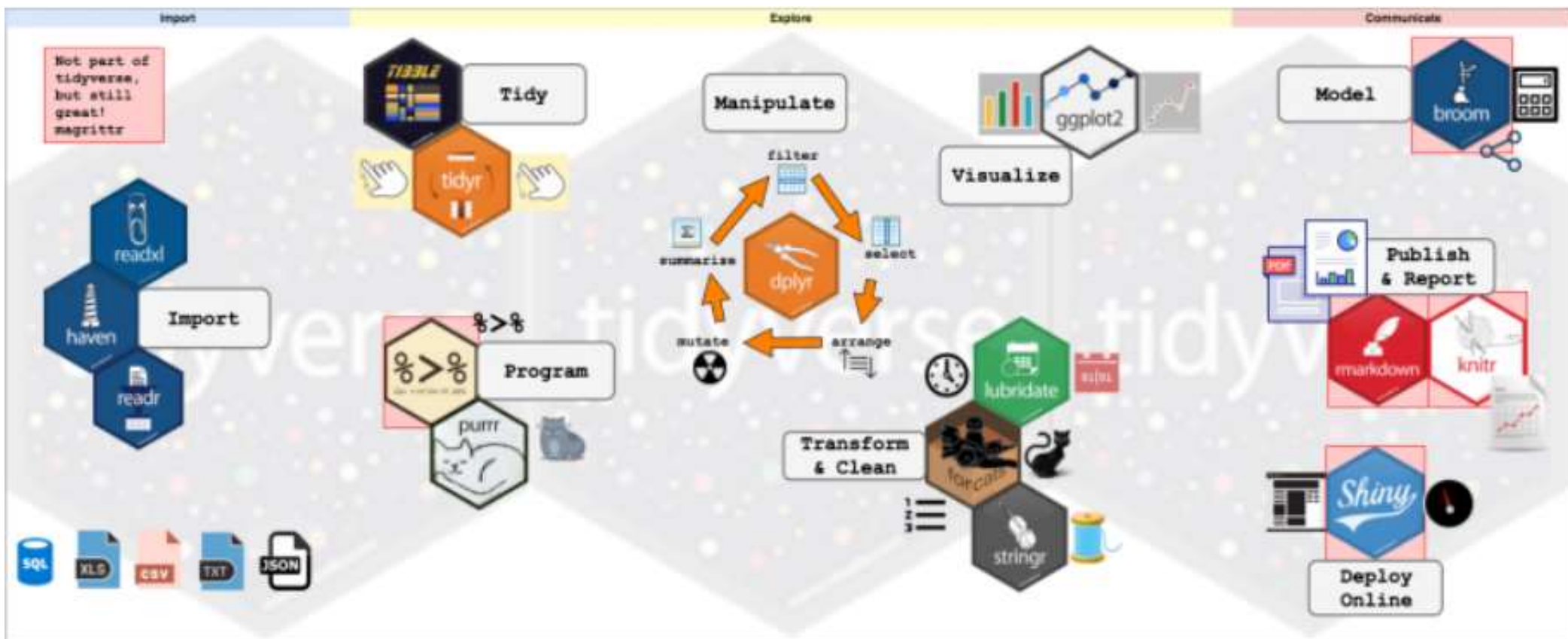

Steep Learning Curve

- Steep learning curve since R is a very technical language and requires remembering R syntax

- Have a plan to apply hands-on simple R examples

- Use a mentor to address challenges and questions since debugging is very difficult

- Learn common R packages & function from Tidyverse & DPLYR

- Stay focused and not get distracted with matrices or statistical modeling

Avoid the Steep Learning Curve - https://r-guru.com/learn-r

R Process: Data Input to Statistical Analysis

Tidyverse

Input

Import Data
Create R Objects
Create Variables

DPLYR

Process

Clean Data
Derive Variables
SQL Process
Transpose Data
Join / Append Data
Build Analysis Data
   Frames
Summarize Data

GTSummary

Output

Format Results
Create Tables
Create Graphs

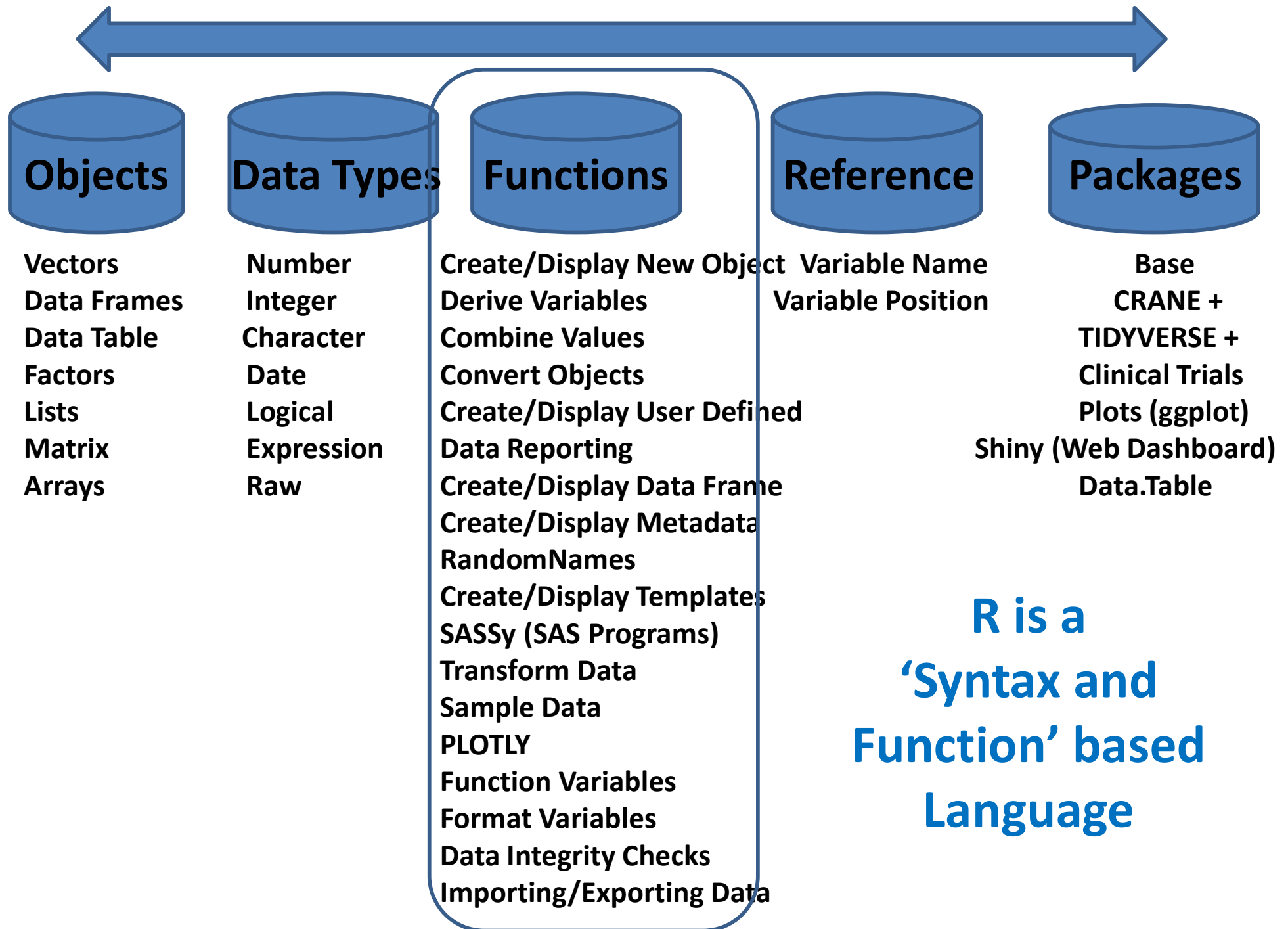R has functions to import data, process data and output results!

Objects in Memory
The Final Frontier

Tidyverse is a Validated and Popular R Package!
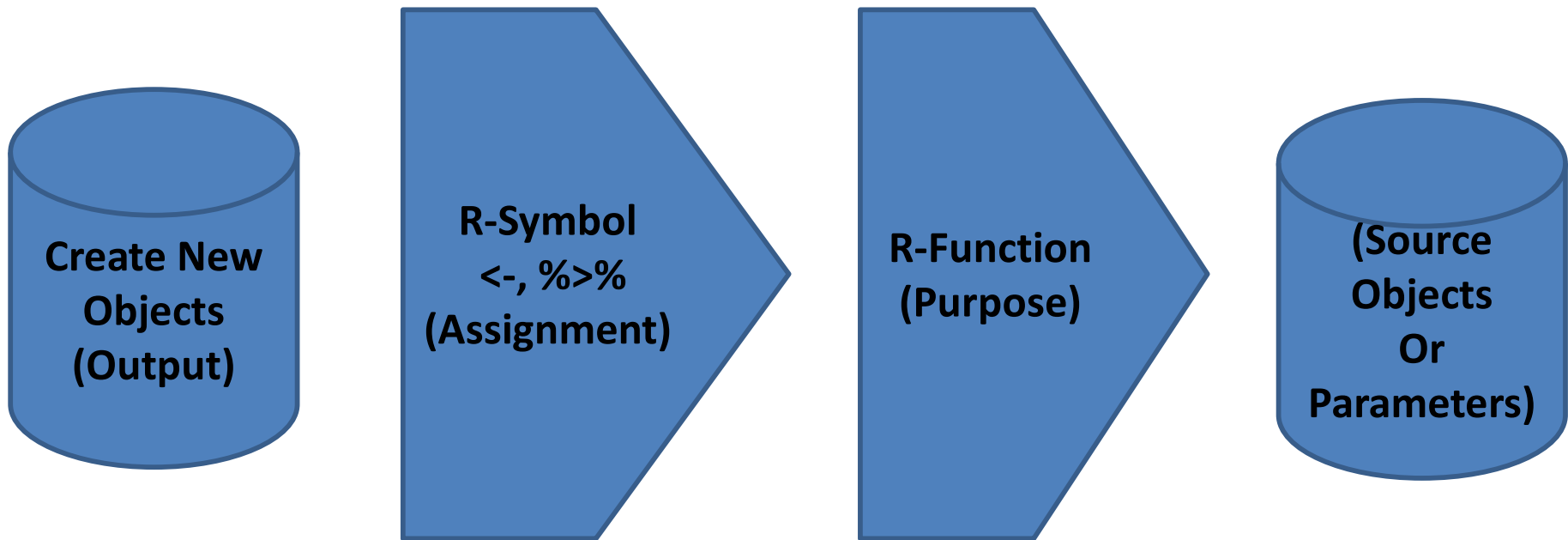
Most all Features

- Import Data
- Data Manipulation
- Program Language
- Visualize, Statistical Models
- Publish, Web Applications

R Structure, Rules and Scope - https://r-guru.com/r-process-flow

Objects	Data Types	Functions	Reference	Packages
Vectors	Number	Create/Display New Object	Variable Name	Base
Data Frames	Integer	Derive Variables	Variable Position	CRANE +
Data Table	Character	Combine Values		TIDYVERSE +
Factors	Date	Convert Objects		Clinical Trials
Lists	Logical	Create/Display User Defined		Plots (ggplot)
Matrix	Expression	Data Reporting		Shiny (Web Dashboard)
Arrays	Raw	Create/Display Data Frame		Data.Table
		Create/Display Metadata		
		RandomNames		
		Create/Display Templates		
		SASSy (SAS Programs)		
		Transform Data		
		Sample Data		
		PLOTLY		
		Function Variables		
		Format Variables		
		Data Integrity Checks		
		Importing/Exporting Data		

R is a 'Syntax and Function' based Language

R Data Object Process Flow: Validated R Packages

Assure open and closed brackets: [], (), ''. Close bracket defines end of R-command. %>% saves time from creating intermediate objects.

Create New Objects (Output)

R-Symbol <-, %>% (Assignment)

R-Function (Purpose)

(Source Objects Or Parameters)

Requires Valid Object Names, Symbols, Functions, Parameters and Objects

One 'Function' Away
my_data <- cbind(usubjid, age, date, in_study)
New_R_Object <- R_Function(R_Objects)

R_Function(Parameter 1, Parameter 2, etc.)
R Keyword(Existing R_Objects)

Understanding R Programming
%>% combines R-commands, Variables are Index Referenced

- Import Data (CSV, Excel, Text, Datasets)
- Metadata Properties (name, label, length)
- Load R Dataframes

- Create $ Variables
- Update $ Variables
- Numeric Derivations
- String Operations
- Conditional Processing

- [Drop / Keep / Rename] $ Variables

- Add / [Subset Condition] / Sort Records

R has most all SAS operations

Data Management Example: Learn and apply from tasks of simple R functions with default parameters - https://r-guru.com/Common-R-Tasks

Basic R Examples with comments

Data Management Operations:

- **# Create Data Frame**
- mydata <- data.frame(
- class = c("1st", "2nd", "3rd", "Crew"),
- n = c(325, 285, 706, 885),
- prop = c(14.8, 12.9, 32.1, 40.2)
-)
- mydata

- **# Keeping Variables**
- test_df2=mydata[c('class', 'n')]
- test_df2

- **# Dropping Variables**
- test_df3= subset(mydata, select = -c(class))
- test_df3

- test_df4= mydata[-c(3)]
- test_df4

- mydata$myvar <- NULL
- **# subset(x, subset, select, drop = FALSE, ...)**
- x – data frame
- subset – Subset expression
- select – Keep variables

R Exercises for each type of Task

Data Management Operations Exercises (*Next Section*)

1. Create mydata1 data frame from dropping gender1 variable in mydataframe.
2. Create mydata2 data frame from keeping gender and age variables in mydataframe.
3. Create mydata2b data frame by creating new variable newvar as if age > 50 then 'Above 50' else '50 or Below'. (cut(), case_when(), mutate() with case_when(), mutate())
4. In mydataframe, rename variable gender to sex.
5. Replace NA values with Zeros.

You can program SDTMs and ADaMs in R

SDTM/ADaM Datasets using R

R- Dplyr Syntax

Subject level derivation -Sample code:

```
adsl <-    dm %>%        # read %>% as "and then"
           select(studyid, subjid, age, sex, height, weight, race, scrfl) %>%
           mutate(bmi = (weight*703)/height^2 ) %>%
           filter(scrfl == "Y") %>%
           select(-scrfl) %>%
           arrange(studyid, subjid)
```

R is a Programming Language: Process, Order, Logic & Comments

Avoid the Steep Learning Curve

- Logic and Complex Variables

- Piping, %>% to concatenate R functions
  (Select, Mutate, Filter, Arrange)

```
Order
  6  adsl <- dm %>%    # separate lines per R command help for reading
  1  select(studyid, subjid, age, sex, height, weight, race, scrfl) %>%
  2  mutate(bmi = (weight*703)/height^2 ) %>%
  3  filter(scrfl == "Y") %>%
  4  select(-scrfl) %>%
  5  arrange(studyid, subjid)
```

With %>%, several R commands execute together which is similar to SAS Procedures.

- SAS dataset options are direct variable and record references in R.

df2 <- df1[*row* conditions , *column* conditions]

```
print(df1[df1$vr1 == 'male', c('vr1', 'vr2')])   # print vr1 & vr2 for males
```
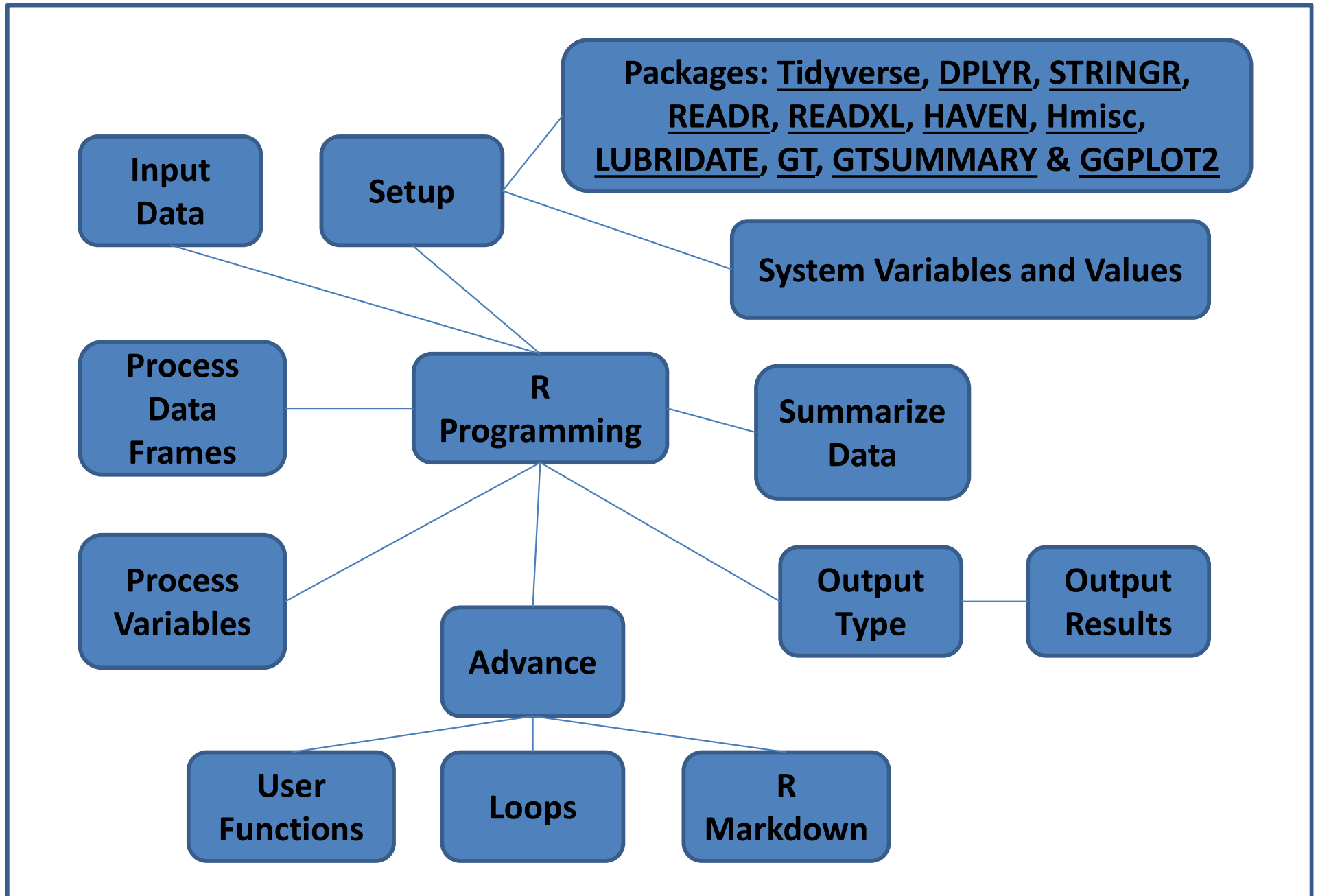
R-Guru.com is a Resource Hub for SAS Programmers
Site Map and Best Practices Checklist
https://r-guru.com/best-practices-checklist
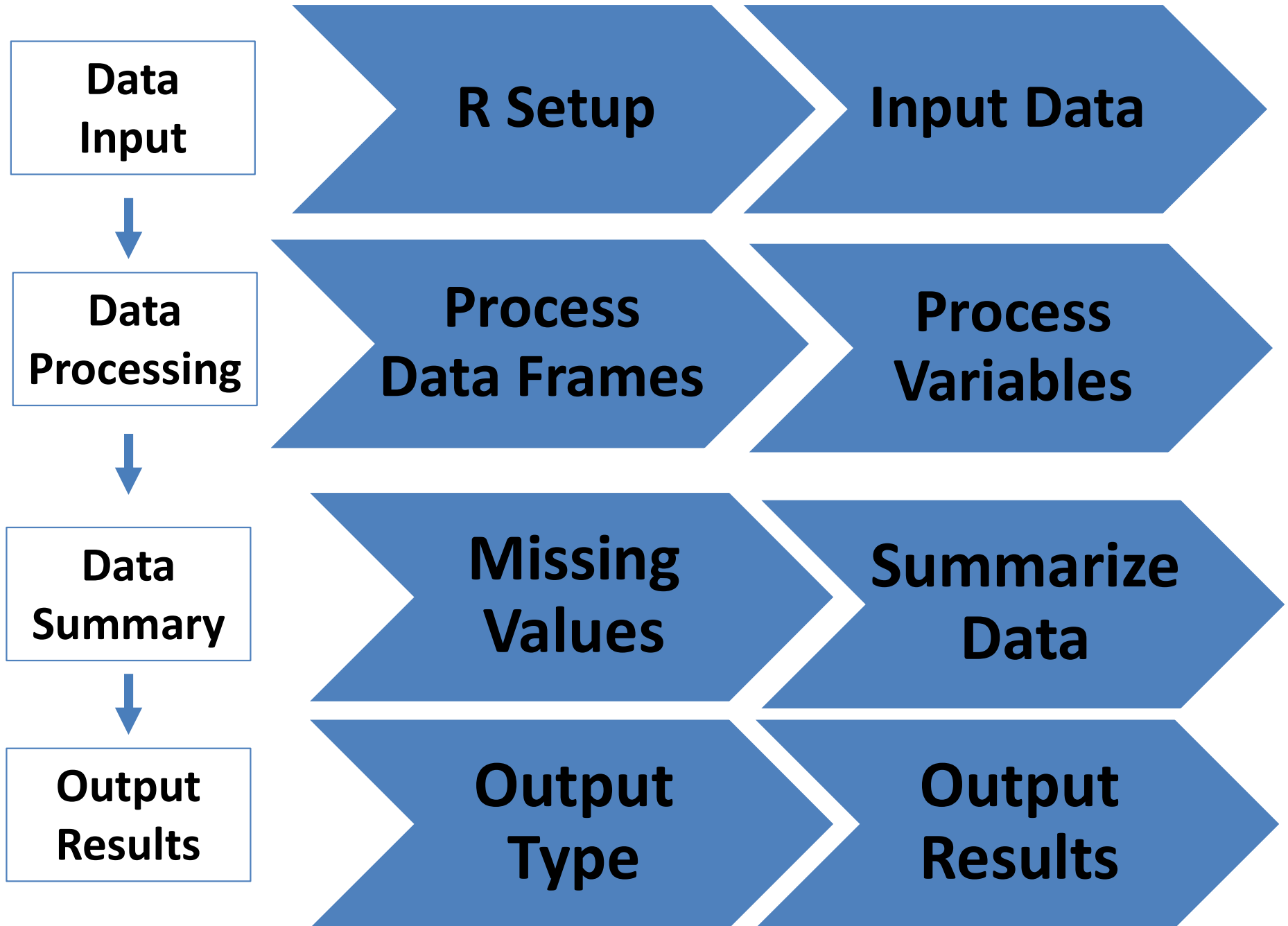
- Home
  - R Programming Best Practices
    - R Online Guides
    - R Cheat Sheets
    - R Debugging
    - Videos and White Papers
  - Join us
    - Testimonials
    - Site Map
    - Contact
  - In The News
- Install & Learn R
  - Install R Software
    - Install R Packages and Load Libraries
  - What 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

R_Guru <- R(*for SAS Programmers*)

Search R-Guru

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

R Best Practices Checklist

Data Input	R Setup	Input Data
Data Processing	Process Data Frames	Process Variables
Data Summary	Missing Values	Summarize Data
Output Results	Output Type	Output Results

R Best Practices Checklist for Data Input & Processing

R Setup	Common Packages, df$var[r,c], %>%	Directories & Path Names
Input Data	Excel & SAS Datasets	Descriptive Statistics
Process Data Frames	names(df) <- tolower(names(df))	Copy, Join, Query, Append & Transpose
Process Variables	Filter, Select, Mutate & Order	Functions: Type, Strings, Num & Dates

15

R-Guru Best Practices Mind Map

Input Data

Setup

Packages: Tidyverse, DPLYR, STRINGR, READR, READXL, HAVEN, Hmisc, LUBRIDATE, GT, GTSUMMARY & GGPLOT2

System Variables and Values

Process Data Frames

R Programming

Summarize Data

Process Variables

Advance

Output Type

Output Results

User Functions

Loops

R Markdown

R-Guru Best Practices Checklist

Data Input	R Setup	Input Data
↓		
Data Processing	Process Data Frames	Process Variables
↓		
Data Summary	Missing Values	Summarize Data
↓		
Output Results	Output Type	Output Results

R-Guru Best Practices Checklist for Data Input & Processing

R Setup	Common Packages, df$var, df[r,c], %>%	Directories & Path Names
Input Data	Excel & SAS Datasets	Descriptive Statistics
Process Data Frames	names(df) <- tolower(names(df))	Copy, Join, Query, Append & Transpose
Process Variables	Filter, Select, Mutate & Order	Functions: Type, Strings, Num & Dates

R-Guru Best Practices Checklist for Data Summary & Output

System Variables and Values	row_number(), nrow(), ncol()	Missing Values NA, '.', ' '
Summarize Data	group_by(), rowwise() & Overall	N, Min, Max & Sum
Output Type	Data Frame, Excel & SAS Dataset	RTF, PDF & HTML
Output Results	Lists, Tables & Graphs	Statistical Modeling

R-Guru Cheat Sheet is for SAS Programmers
Download at R-Guru.com

R-Guru.com Cheat Sheet for Statistical Programmers
R Process: Data Input to Statistical Analysis

Tidyverse

Input
- Import Data
- Create R Objects
- Create Variables

DPLYR

Process
- Clean Data
- Derive Variables
- SQL Process
- Transpose Data
- Join / Append Data
- Build Analysis Data Frames
- Summarize Data

GTSummary

Output
- Format Results
- Create Tables
- Create Graphs

Create New Objects (Output)

R-Symbol <-, %>% (Assignment)

R-Function (Purpose)

(Source Objects Or Parameters)
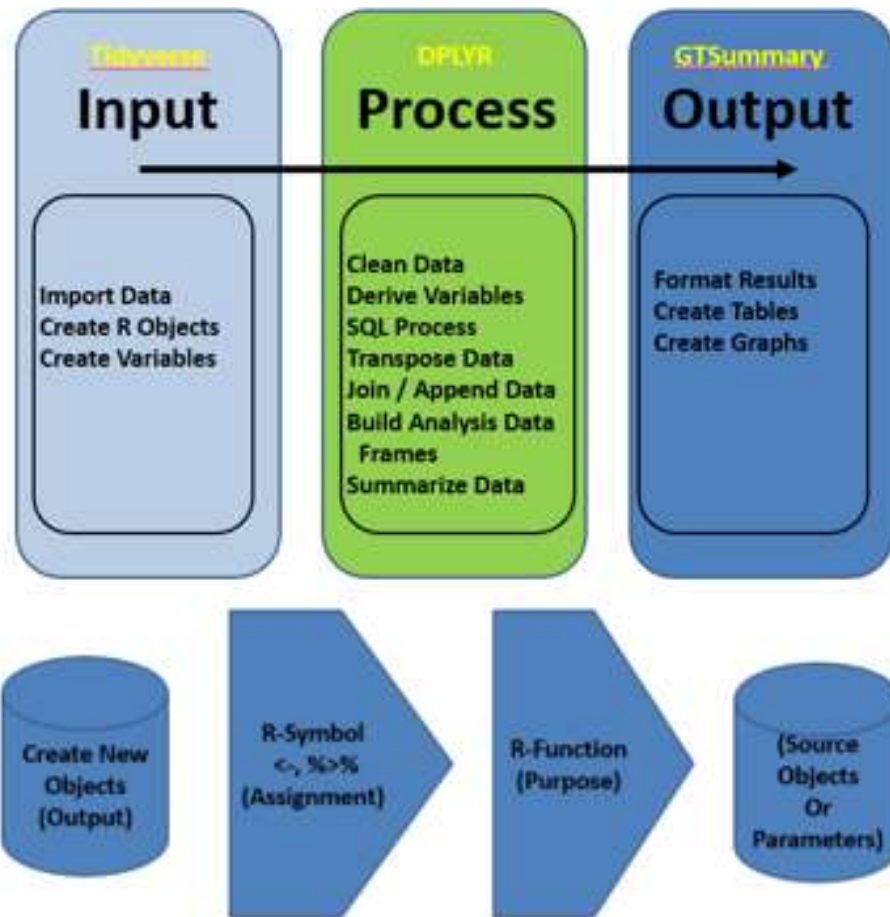
Requires Valid Object Names, Symbols, Functions, Parameters and Objects

This guide contains common and best practice examples for creating, updating and reporting data frames in the pharma and medical device industries. This guide has sections for workspace setup, compare and contrast common R function and R and SAS and debugging which are ideal for SAS programmers making the transition to R. When possible, base R sample data frames are used in examples.

Tidyverse, DPLYR, DATA.TYPE, STRINGR, READR, READXL, HAVEN, Hmisc, arsenal, LUBRIDATE, PARSEDATE, GT, GTSUMMARY & GGPLOT2 are common and validated R packages by RStudio and the Pharma Industry.

Mutate() function has five key features: case_when(), simple expression, summary functions, rowwise(), and group_by()/ungroup() with summary functions.

df# are data frame names & vr# are variable names. Character or numeric variables depend on the function and values. R functions may be nested for multiple tasks.

R-Guru Best Practices Mind Map

Input Data

Setup

Packages: Tidyverse, DPLYR, STRINGR, READR, READXL, HAVEN, Hmisc, LUBRIDATE, GT, GTSUMMARY & GGPLOT2

System Variables and Values

Process Data Frames

R Programming

Summarize Data

Process Variables

Advance

Output Type

Output Results

User Functions

Loops

R Markdown
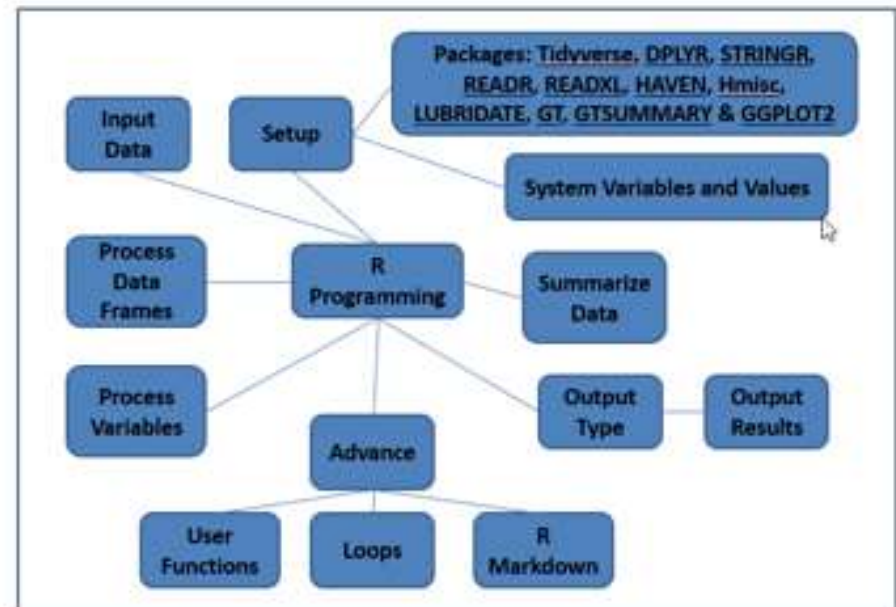
R-Guru Cheat Sheet: Compare and Contrast R and SAS

TASK	R	SAS
Language	Interpreter	Compiler and Interpreter
Character Var Length	N/A	length
Rounding 2.5	2 (even number)	3 (up)
Sorting Missing Values	'NA' is last obs unless converted to missing	Missing is first obs
Common Features **Data:** Input (Excel, CSV), Management, Analysis, & Reporting (RTF, PDF) **Var Type:** Character, Numeric and Date Variables **Other:** SQL, Do-Loops	R Studio Data Frames Direct Variable and Record References as.character() , as.numeric() vfmt[df$vr1] R Shiny App	Display Manager Datasets Dataset Options (Keep, Drop, Where) put(), input() proc format

R-Guru Cheat Sheet is for SAS Programmers
Compare and Contrast Common R Functions

TASK	METHOD 1	METHOD 2
Query, Add Variables	mutate(dose2 = (dose*2)) cbind(df1, vr1=1, vr2='Drug A')	df1[df1$vr1 == 'male', c('vr1', 'vr2')] # df options
Add Variables by Conditions	case_when(grep("Yes", vr1) ~ 'Yes')	ifelse(data$vr1 >= 4, 1, 0) , if_else()
Add Summary Variables (Overall)	summarize(mean_mpg = mean(mpg, na.rm = TRUE)) mutate(vr3 = mean(vr2, .1))	summarise_at(vars(mpg, wt), list(m=mean, sd=sd), na.rm=TRUE) apply(mtcars, 2, mean)
Group By Vars	group_by(vr)	ungroup() # best practices to prevent subsequent group processing, best used with mutate() to keep all variables
Variable Type Conversion	as.character(vr1)	as.numeric(vr1) as.Date("2021-01-25")
Recode Values	vfmt <- c("M"="Male", "MALE"="Male", "F"="Female", "FEMALE"= "Female") df$vr2 <- vfmt[df$vr1]	recode(vr1, 'val1'='val1a', 'val2'='val2a') recode(vr1, !!!vfmt$vr1))

R-Guru Cheat Sheet is for SAS Programmers
Debugging R: Syntax, Logic, Data

ERROR TYPE	SOLUTIONS
Invalid or Missing Packages, Path names, Libraries not Loaded	Load and confirm packages, path names and libraries
Invalid or Missing Data Frames, Objects or Variables	Confirm correct and existing data frames (instead of matrix), objects and vars, lower case all names since case-sensitive, correct order of tasks (select, filter, etc.) within DPLYR (SQL) functions, apply group_by() before summary functions to prevent overall summaries
Invalid or Missing Functions or Operations	Confirm functions exist and correctly applied, confirm variable and function types are consistent
Invalid or Missing Parameters and Options	Confirm correct function usage, case-sensitive, cut/paste working example
Invalid or Missing Data or Format	Confirm data import is correct, lower case data since case-sensitive, remove extra spaces before and after data values, confirm correct date format, apply factors to assign invalid data as NA, data by descriptive stats, freq counts, min, max, etc.
Invalid Logic	Confirm process logic flow, test and view inputs and outputs of each function

Pharmaverse, *the New Frontier* - https://r-guru.com/pharma

For the first time in pharma history, there is collaboration between pharma companies and industry to build Pharmaverse R packages

Pharmaverse, *the New Frontier* - https://r-guru.com/pharma

Pharmaverse: Regulatory Submission Process Flow

R Package	Metadata	Raw to SDTMs	To ADaMs	To Tables, Lists and Graphs
R Scripts	N/A	R Scripts	R Scripts	R Scripts: Tables & Lists, Graphs
Pharmaverse	Metacore	SDTMChecks	Admiral	TLGs, Tpylr

admiral

Our Charter

End-to-End Clinical Reporting Packages

SDTM
ADaM
TLGs
eSub
• Metadata
Utility
Validation

Metadata

metacore | OS Activity 57 | riskmetric 0.47 | CRAN 0.1.1 | Contributors 9
Provides an interface to read in various metadata sources and store in a standardized object

metatools | OS Activity 57 | riskmetric 0.6 | CRAN 0.1.3 | Contributors 2
Enables the use of metacore objects to build, enhance or check datasets using metadata - including removing/adding supplemental qualifiers from/to the parent SDTM domain

‹ eSub   Utility ›

Pharmaverse R packages are developed and validated by top pharma companies. These R packages help 'jump start' the process!

R for Clinical Study Reports and Submission

https://r4csr.org
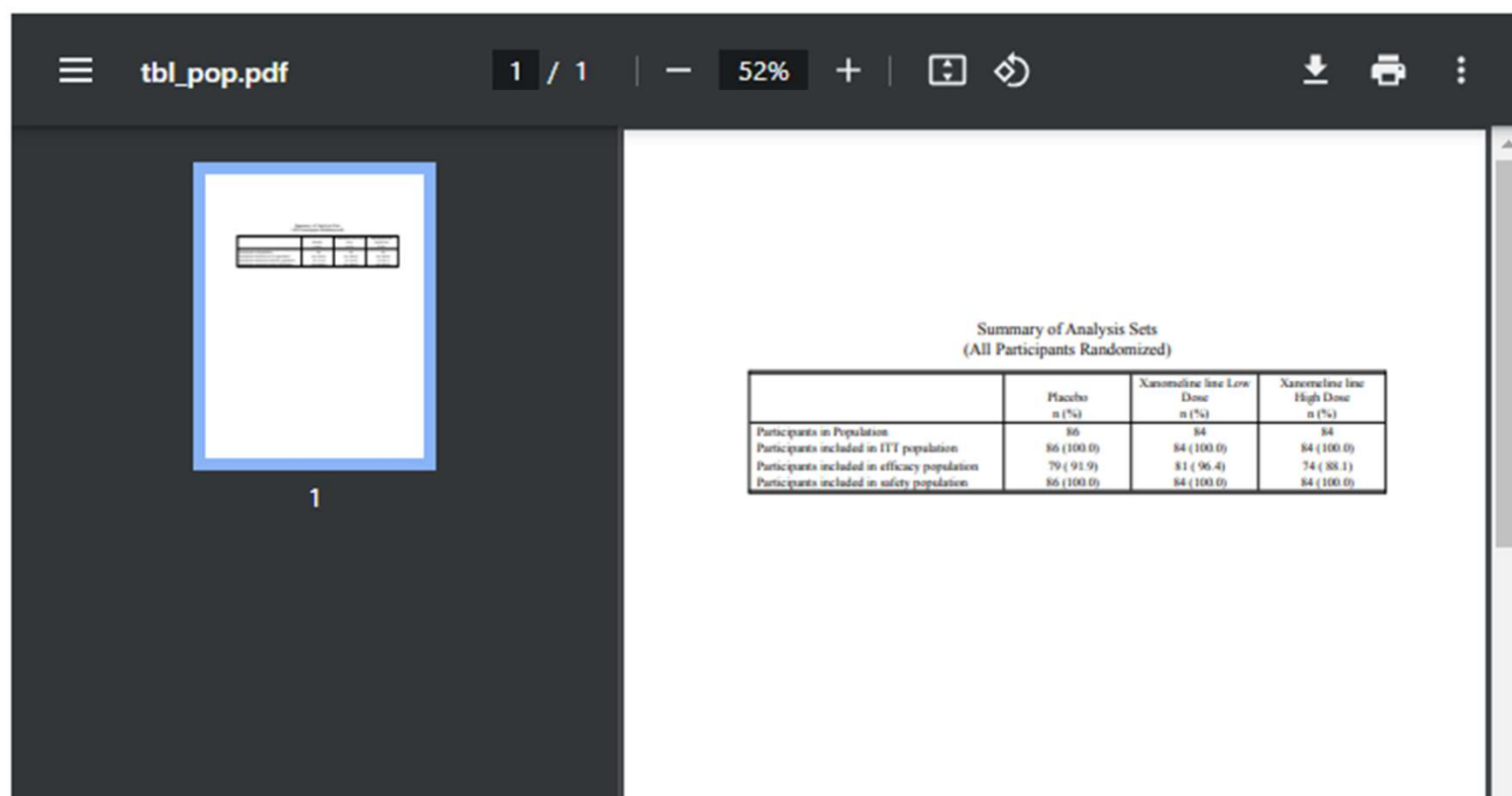
3 Analysis population

Following ICH E3 guidance, we need to summarize the number of participants included in each efficacy analysis in Section 11.1, Data Sets Analysed.

```r
library(haven) # Read SAS data
library(dplyr) # Manipulate data
library(tidyr) # Manipulate data
library(r2rtf) # Reporting in RTF format
```

In this chapter, we illustrate how to create a summary table for the analysis population for a study.

tbl_pop.pdf   1 / 1   52%

Summary of Analysis Sets
(All Participants Randomized)

	Placebo n (%)	Xanomeline line Low Dose n (%)	Xanomeline line High Dose n (%)
Participants in Population	86	84	84
Participants included in ITT population	86 (100.0)	84 (100.0)	84 (100.0)
Participants included in efficacy population	79 (91.9)	81 (96.4)	74 (88.1)
Participants included in safety population	86 (100.0)	84 (100.0)	84 (100.0)

Shiny

- *Benefits*: Impact Analysis, Data Transparency

- Create Shiny App in *Days* instead of weeks or months with SAS

- Enable *Rapid* Data Visualization
  - Data Queries / Lists / Detail / Summary Graphs / Stats / Tables

- *Interactive* Tables, Lists and Graphs

- Fosters collaboration and communication among clinical team

- R Shiny Submission Packages for interactive reviews

R-Guru.com is a Resource Hub for SAS Programmers
R Cheat Sheets, Books and Blogs, Pharmaverse

https://r-guru.com/r-cheat-sheets
**https://r-guru.com/books-and-blogs**

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
- GGPLOT2
- 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
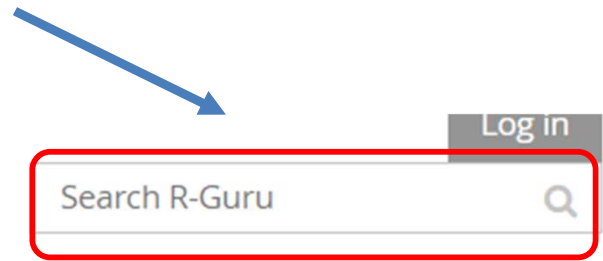- Mastering Shiny

Pharmaverse

- Metadata
- R Shiny
- SDTMs
- ADaMs
- Statistical Analysis
- Tables, Lists and Graphs
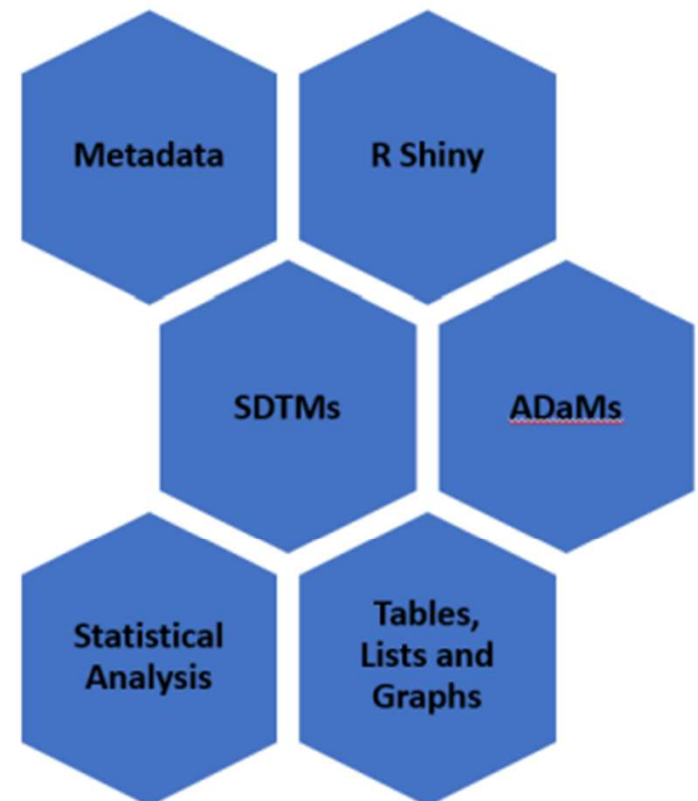
R-Guru is for SAS Programmers Looking to make smooth transition to R
https://r-guru.com/install_r

Keyword Search and Header Links: Every Page

Log in

Search R-Guru

R-Guru ⌣ Resource Hub

Click: Site-Map, How-To, Tasks I, Tasks II, Functions, Book, Tidyverse, Debug, Test, Exercises, Data, Pharma, Shiny, Cheat Sheets, Best Practices, Videos, FAQs

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

Home > Ins
Install R Packa

Install R Software

What Is R and Why Learn R?

Learn R Programming

Learn R Programming

Below are s   Run R Programs to Create Objects   tudio provides a great programming environment to access
and manag
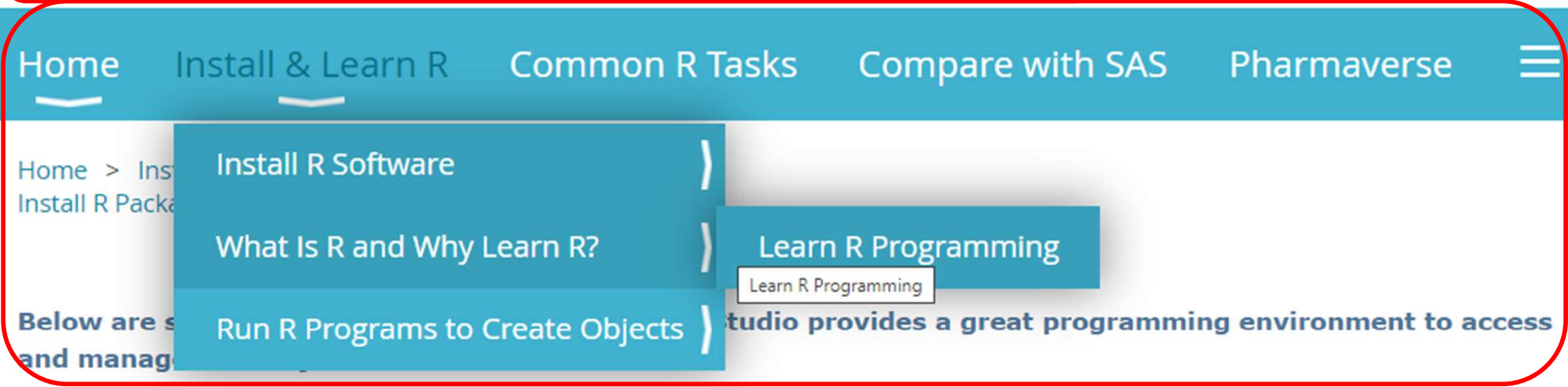
1. **Read How to Download R and RStudio article**

2. **View Install R Packages Video**

3. **Check Hardware Configuration to confirm minimum hardware and memory (Workbench, Connect, Package Manager)**

Navigation for Learning: Left to Right and Top to Bottom

R-Guru.com:
Over 100 Common R FAQs
https://r-guru.com/Common-R-FAQs

R-Guru ⌣ Resource Hub

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

Home > Common R FAQs

Common R FAQs UCLA R FAQs (*Submit your Common R FAQ*)

Below are common R technical questions with R solutions to solve real-world tasks.

1. What are common syntax for libname, filename and reading datasets?   See R paper.

```
sdtm <- "c:/product/study/analysis/data/sdtm" # assign libname to object named sdtm

out <- "c:/product/study/analysis/data/adam"  # assign out filename to path

library(haven) # required to read SAS datasets

dm <- read_sas(file.path(sdtm,"dm.sas7bdat")) # read sas file as a data frame

#'read_sas' function from the haven package (part of the tidyverse)

taadmin <- read_sas("H:/rproject/project_y_r2/taadmin.sas7bdt")
```

R-Guru.com: Introduction to R Webinars
https://r-guru.com/videos-and-white-papers

GuptaProgramming@gmail.com

What is R and Why Should You Learn R?

Sunil Gupta

SAS® Savvy
The One Stop SAS® & CDISC Solutions

UC San Diego EXTENDED STUDIES
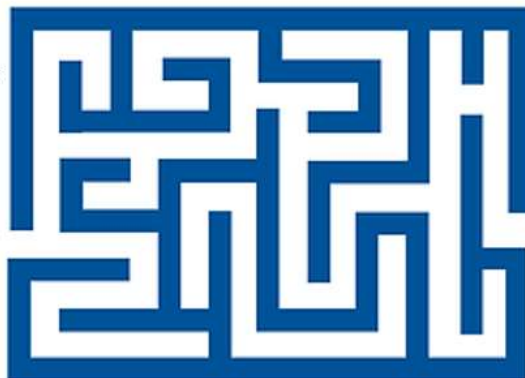
R-Guru ⌣ Resource Hub

Metadata (Pharmaverse) > SDTMs > SDTMChecks (Pharmaverse) > ADaMs

Learning R _can be confusing_:
- For SAS programmers
- For SDTMs and ADaMs
- For Tables, Lists & Graphs
- For Pharmaverse R Packages

Start

Finish

Learning R _can be fun_:
- Less Technical
- Easy to understand R concepts
- Easy to Search and Navigate for R Solutions within four clicks

Writing Your First R Program, Just the Basics

Sunil Gupta
Sunil@SASSavvy.com