Introduction To R Part II

Adrian Rohit Dass

January 12th, 2024

Recap: Outline from Part I

- Why use R?
- R Basics
- R for Database Management
    - Reading-in data, merging datasets, reshaping, recoding variables, sub-setting data, etc.
- R for Statistical Analysis
    - Descriptive and Regression Analysis
- Applied Example
- Other topics in R
    - Tidyverse
    - Parallel Processing
    - R Studio
    - R Markdown
- Applied Example 2
- R Resources

R for Health Economics

- A survey conducted to IHPME health economics students in late 2021 suggested the following research interests
  - Working with data
    - Common tasks: reading in data, creating new variables, data subsets, etc.
    - Example packages: base, tidyverse, data.table, etc.
  - Applied econometrics
    - Common tasks: descriptive analysis, regression analysis, etc.
    - Example packages: stats, plm, lmtest, sandwich, etc.
  - Economic Evaluation
    - Common tasks: model building (Markov, Microsim, etc.), sensitivity analysis, etc.
    - Example packages: base, stats, ggplot2, etc.

Outline for Part II

- data.table package
  - Comparison to base R and tidyverse
    - Reading in data, syntax, etc.
- R as a programming language
  - functions, for loops, flow control, matrix algebra, etc.
- R Markdown for beamer presentations
- R Resources

data.table Package in R

data.table Package in R (Continued)

Package: data.table

Description from documentation: Fast aggregation of large data (e.g. 100GB in RAM), fast ordered joins, fast add/modify/delete of columns by group using no copies at all, list columns, friendly and fast character-separated-value read/write. Offers a natural and flexible syntax, for faster development.

General syntax

DT[i, j, by]

Source: https://cran.r-project.org/web/packages/data.table/data.table.pdf

data.table Package in R (Continued)

Why data.table? Factors to consider:

- Speed
- Memory Usage
- Syntax
- Features

See full discussion: https://stackoverflow.com/questions/21435339/data-table-vs-dplyr-can-one-do-something-well-the-other-cant-or-does-poorly

A note on computational efficiency

- Quote from Xu et al. (2016): “The authors have worked on several cases, in which analysis can be significantly improved by just replacing the usage of data frame with data table package.” Empowering R with High Performance Computing Resources for Big Data Analytics

data.table Package in R (Continued)

Syntax comparisons

- Base R

read.csv(file, ...)

- Tidyverse (readr package)

read_csv(file, …)

- data.table

fread(file,…)

How do the load times compare? Test on Canadian Community Health Survey 2013/14, 285.4 MB

Full Code for Applied Example

```
# Base R ----
t1 = Sys.time()
d1 = read.csv("cchs201314.csv")
comp.time1 = Sys.time() - t1

# Tidyverse (readr package) ----
library(readr)
t2 = Sys.time()
d2 = read_csv("cchs201314.csv")
comp.time2 = Sys.time() - t2

# data.table ----
library(data.table)
t3 = Sys.time()
d3 = fread("cchs201314.csv")
comp.time3 = Sys.time() - t3
```

data.table Package in R (Continued)

Subset Data: Age group >=3 (18 and over)

- Base R

```
cchs.sub = cchsdata[cchsdata$DHHGAGE>=3,]
```

- Tidyverse

```
cchs.sub = filter(cchsdata, DHHGAGE>=3)
```

- data.table

```
cchs.sub = cchsdata[DHHGAGE>=3]
```

data.table Package in R (Continued)

Create new variable: flag (=1) for age group >=3 (18 and over), 0 otherwise

- Base R

```
cchs$age_flag = 0
cchs$age_flag[cchs$DHHGAGE>=3] = 1
```

- Tidyverse

```
cchs = mutate(cchs, age_flag = ifelse(DHHGAGE>=3, 1, 0))
```

- data.table

```
cchs[,age_flag:=ifelse(DHHGAGE>=3, 1, 0)]
```

data.table Package in R (Continued)

Frequency of Age Group variable (CCHS)

- Base R

```
ftable = table(cchsdata$DHHGAGE)
```

- Tidyverse (dplyr package)

```
ftable = cchsdata %>%
 group_by(DHHGAGE) %>%
 summarize(freq = n())
```

- data.table

```
ftable = cchsdata[,.(.N),by=DHHGAGE]
```

Summary for this section

- When working with data in R, many options are available
- data.table may be a favourable alternative to base and tidyverse
- As noted in discussion post, consider: speed, memory usage, syntax, and features

R as a programming language

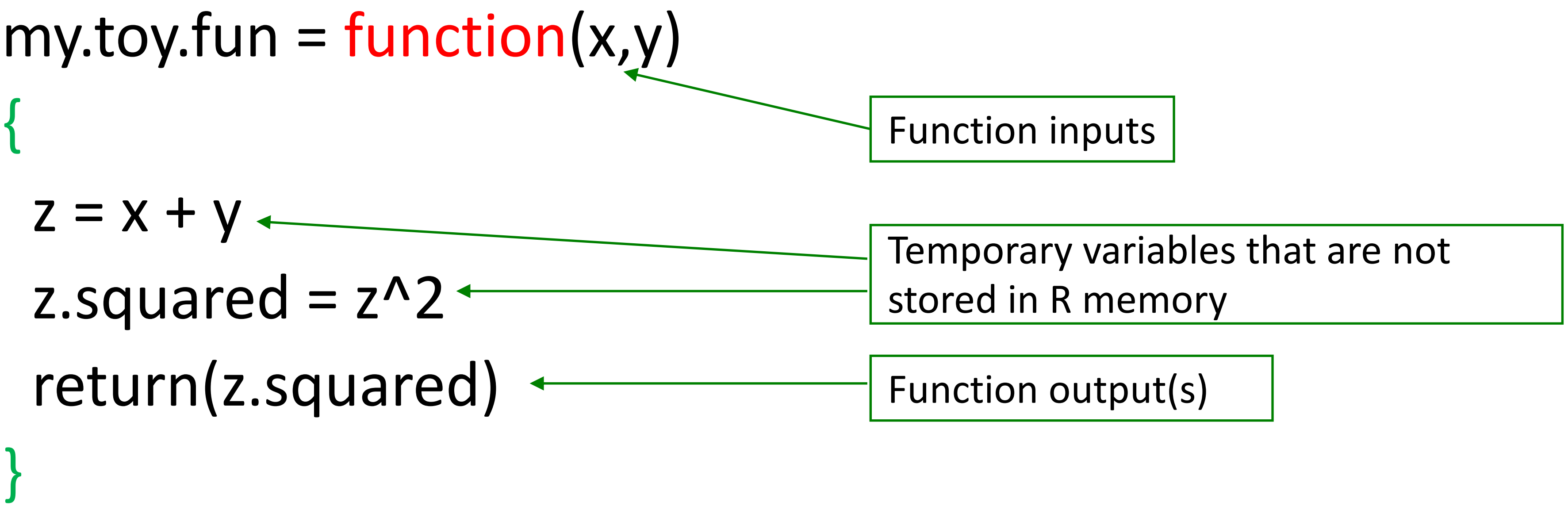
R as a programming language

- The programming language in R is object oriented
  - Roughly speaking, this means that data, variables, vectors, matrices, characters, arrays, etc. are treated as “objects” of a certain “class” that are created throughout the analysis and stored by name.
  - We then apply “methods” for certain “generic functions” to these objects
- It can be used for tasks outside of data analysis, similar to other programming languages
- The language itself was designed for programming with data
- See Kleiber & Zeileis (2008) for more

Writing Functions in R

- Using R for data analysis typically involves the utilization a sequence of commands for inputs to produce outputs
- These sequences can be wrapped into a function, which can be called to avoid repeating these sequences by hand
- Functions can be used for many different purposes, including data formatting, Markov and Microsim models, applied econometrics, etc.

Writing Functions in R (Continued)



```
my.toy.fun = function(x,y)
{
  z = x + y
  z.squared = z^2
  return(z.squared)
}
```

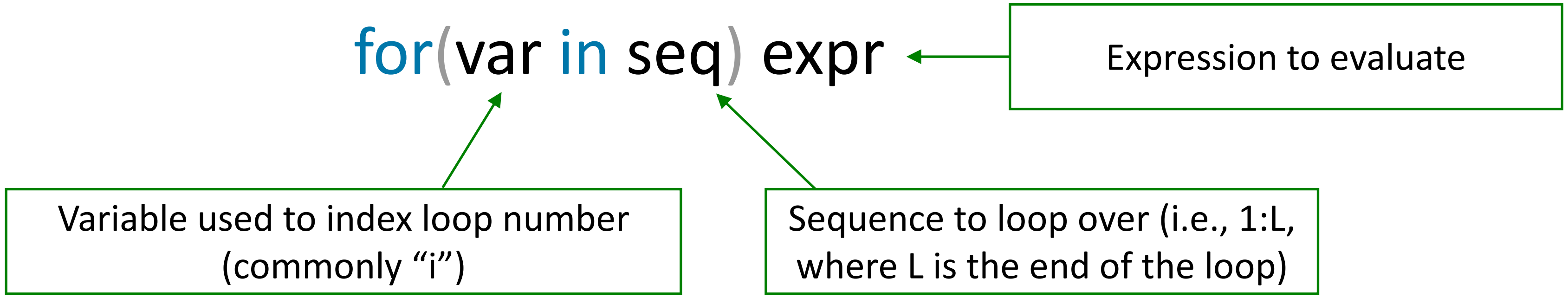
Function inputs

Temporary variables that are not stored in R memory

Function output(s)

```
my.toy.fun(x = 4, y = 5)
[1] 81
```

For Statement in R



for(var in seq) expr

Expression to evaluate

Variable used to index loop number (commonly “i”)

Sequence to loop over (i.e., 1:L, where L is the end of the loop)

Example

```
for (i in 1:10)
{
print(i)
}
```

Foreach function

- Package: foreach
- Provides looping structure that returns a value
- General syntax: foreach(..., .combine, .init, .final = NULL, .inorder = TRUE, .multicombine = FALSE, .maxcombine = if (.multicombine) 100 else 2, .errorhandling = c("stop", "remove", "pass"), .packages = NULL, .export = NULL, .noexport = NULL, .verbose = FALSE)

Online help file:

https://www.rdocumentation.org/packages/foreach/versions/1.5.2/topics/foreach

Foreach function (continued)

Applied Example

- Read in 10 waves of a dataset
  - Could be survey waves, administrative dataset, etc.
  - Contain the same variables in each wave
  - Goal is combine (stack) the datasets

Foreach function (continued)

```
# Load libraries ----
library(foreach)
library(data.table)
# Load data ----
files = paste0("Yearly Files/wdata", paste0(seq(1, 10, 1), ".csv"))
data.all = foreach(i=1:length(files), .combine = "rbind") %do%
{
ydata = fread(files[i])
return(ydata)
}
```

List function in R

- Generic vectors where each element can be virtually any type of object (Kleiber & Zeileis, 2008)
- This allows us to combine scalars, numeric vectors, data frames, etc. into one object
- Objects can be extracted from list by name through ‘$’ or [[ ]] (element-number wise extraction)

Ex:

```
my.list = list("id" = seq(1, 10, 1), "Explanation" = "Patient ID")
my.list$id # Extract ID
my.list[[1]] # Same as above, but different method
```

Applied Example

Common task: Exporting regression results in R

- In R, some functions are available by default (OLS, GLM, etc.) whereas others are contained in packages written by other users
- This implies that user-written packages designed to work for default R packages may not work for other models
- In addition, some packages are designed to work with LaTeX, which is not necessarily helpful for users who work with Microsoft Word
- Finally, it may be difficult to achieve the desired formatting of results with existing packages

Is it possible to write our own function to create a regression results table?

Bonus challenge: Must be compatible with MS Word and LaTeX

A note on R and LaTeX

- You need to have a LaTeX distribution installed to typeset using this approach
- If you are an active LaTeX user, you likely have MiKTeX, MacTeX, or similar installed already, so no further action is needed
- If you're interested in LaTeX but don't have a LaTeX distribution, you may consider installing *TinyTeX* from the tinytex R package. To install through R:

```
tinytex::install_tinytex()
```

For more details, please see https://bookdown.org/yihui/rmarkdown-cookbook/install-latex.html

Applied Example (Continued)

- Analysis of Health Expenditure Data in Jones et al. (2013) *Chapter Three*
- The data covers the medical expenditures of US citizens aged 65 years and older who qualify for health care under Medicare.
  - Outcome of interest is total annual health care expenditures (measured in US dollars).
  - Other key variables are age, gender, household income, supplementary insurance status (insurance beyond Medicare), physical and activity limitations and the total number of chronic conditions.
- Data can be downloaded from here (mus03data.dta): https://www.stata-press.com/data/musr.html

Regression Results Function Code

```
# Load Libraries ----
library(lmtest)
library(sandwich)
library(foreign)
# Regression Results Function ----
reg.results.fun = function(model, digits)
{
  reg.results = coeftest(model)
  beta = reg.results[,1]
  se = reg.results[,2]
  sig.stars = symnum(reg.results[,4], corr = FALSE, na = FALSE,
                     cutpoints = c(0, 0.001, 0.01, 0.05, 0.1, 1),
                     symbols = c("***", "**", "*", ".", ""))

  results.table = data.frame(cbind("Variable" = rownames(reg.results),
                                   "Beta" = paste(round(beta, digits), sig.stars, sep = ""),
                                   "SE" = round(se, digits)), row.names = NULL)

  return(list("coefficients" = beta,
              "results.table" = results.table))
}
```

Regression Results Function Code (Continued)

```
# Load Data ----
cost.data.all = read.dta("Expenditure Data/mus03data.dta")
cost.data = cost.data.all[cost.data.all$totexp>0,]
# Regression ----
ols.cost.data = lm(totexp ~ age + female + income + suppins + phylim + actlim + totchr, data = cost.data)
summary(ols.cost.data)
# Export Regression Results ----
ols.cost.data.results.all = reg.results.fun(ols.cost.data, 2)
ols.cost.data.results = ols.cost.data.results.all$results.table
save(ols.cost.data.results, file = "ols.cost.data.results.RData")
```

R Markdown code

~~~
---
title: "Untitled"
output:
  pdf_document: default
  html_document: default
  word_document: default
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = FALSE)
```

## Regression

```{r regression}
load("ols.cost.data.results.RData")
knitr::kable(ols.cost.data.results)
```
~~~

Flow Control in R: If Statements

An if/else statement in R takes the general form:

```
if (cond) {
R code if true
} else {
R code if not true
}
```

Can be omitted if empty

Applied Example (Continued)

Common task: Creating output using heteroskedasticity robust standard errors

- Typical summary() function in R only gives output using the standard OLS variance matrix (i.e., assuming homoskedasticity)
- Can we modify our regression function to give results using heteroskedasticity robust variance matrix?
- Can we also add a test for heteroskedasticity in the regression results function?

Updated Regression Results Function Code

```
reg.results.fun = function(model, digits, robust = FALSE)
{
  if (robust==TRUE)
  {
    reg.results = coeftest(model, vcovHC(model, type = "HC1"))
    beta = reg.results[,1]
    se = reg.results[,2]
    sig.stars = symnum(reg.results[,4], corr = FALSE, na = FALSE,
                    cutpoints = c(0, 0.001, 0.01, 0.05, 0.1, 1),
                    symbols = c("***", "**", "*", ".", ""))
    results.table = data.frame(cbind("Variable" = rownames(reg.results),
                                 "Beta.Robust" = paste(round(beta, digits), sig.stars, sep = ""),
                                 "SE.Robust" = round(se, digits)), row.names = NULL)
  } else { same as before (see previous slide)}
  test.hetero = bptest(model)
  return(list("coefficients" = beta,
            "results.table" = results.table,
            "Het.Test" = test.hetero))
}
```

Updated Regression Results Function Code (Continued)

```
# Load Data ----
cost.data.all = read.dta("Expenditure Data/mus03data.dta")
cost.data = cost.data.all[cost.data.all$totexp>0,]
# Regression ----
ols.cost.data = lm(totexp ~ age + female + income + suppins + phylim + actlim + totchr, data = cost.data)
# Export Regression Results ----
ols.cost.data.results.all = reg.results.fun(ols.cost.data, 2)
ols.cost.data.results = ols.cost.data.results.all$results.table
ols.cost.data.results.robust.all = reg.results.fun(ols.cost.data, 2, robust = TRUE)
ols.cost.data.results.robust = ols.cost.data.results.robust.all$results.table
ols.cost.data.results.combine = merge(ols.cost.data.results, ols.cost.data.results.robust, by = "Variable")
save(ols.cost.data.results.combine, file = "ols.cost.data.results.combine.RData")
```

Matrix Algebra

Matrix algebra is often used to perform calculations in decision-making models. It can also be helpful to implement a statistical method that is not available in R.

- Matrix multiplication

```
%*%
```

- Element wise-multiplication

```
*
```

- Matrix inverse

```
solve()
```

Applied Example

Markov model with the following transitional probability matrix

H	**S**	**D**
0.9	0.08	0.02
0	0.8	0.2
0	0	1

Everybody in the model starts in H

Can we use matrix algebra to solve for the second period health states?

Matrix Algebra Code

```
rm(list = ls()) # Clear memory
A = matrix(0, nrow = 2, ncol = 3) #2X3 matrix of 0s (Two time periods, 3 health states)
P = rbind(c(0.9, 0.08, 0.02),
          c(0,0.8, 0.2),
           c(0,0,1)) #3x3 transitional probability matrix
A[1,] = c(1, 0, 0) #Initial health states
A[2,] = A[1,] %*% P #Health states in period 2
print(A) #Display matrix
```

Summary for this section

- Using the R programming language allows for the creation of custom made functions and operations
- This allows us to go beyond the pre-canned routines available in R to create custom made solutions that suit particular needs
- Combining custom functions with pre-canned routines give us a large number of tools to use for data analysis

R Programming in Economic Evaluation

- Markov

Alarid-Escudero, F., Krijkamp, E. M., Enns, E. A., Yang, A., Hunink, M. G., Pechlivanoglou, P., & Jalal, H. (2021). A Tutorial on time-dependent cohort state-transition models in R using a cost-effectiveness analysis example. *arXiv preprint arXiv:2108.13552*.

- Microsimulation

Krijkamp, E. M., Alarid-Escudero, F., Enns, E. A., Jalal, H. J., Hunink, M. M., & Pechlivanoglou, P. (2018). Microsimulation modeling for health decision sciences using R: a tutorial. *Medical Decision Making*, *38*(3), 400-422.

Making Beamer Presentations with R Markdown

- Similar to typsetting word documents, R Markdown can also be used to create presentations
- This includes the LaTeX based beamer presentations
- Outside of R Markdown (i.e. through a LaTeX editor), creating presentations in beamer can be tedious (i.e. \begin{frame} \end{frame} to create slides, \begin{itemize} \end{itemize} to create bulleted lists, etc.) and tables need to be in a certain format to be included
- R Markdown can simplify the process

Making Beamer Presentations with R Markdown (Continued)

- Go to File --> New File --> R Markdown --> Presentation --> PDF (Beamer)
- New slides can be created with one line of code: ##
- Similar to documents, R chunks can be used to work with R and include R objects in output
- List of possible themes and colours can be found here:

https://hartwork.org/beamer-theme-matrix/

Applied Example: Beamer Presentation

- Can we create a beamer presentation in R Markdown that includes the regression results we generated previously?

R Markdown Code

~~~
---
title: "Untitled"
header-includes:
  - \usepackage{booktabs}
output:
  beamer_presentation:
    theme: "Madrid"
---
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = FALSE)
```
## Regression
```{r regression}
load("ols.cost.data.results.combine.RData")
knitr::kable(ols.cost.data.results.combine, format = "latex", booktabs = T)
```
~~~

Conclusions

- data.table offers another set of tools for working with data in R
- Using the programming capabilities within R allows us to write our own functions that expand the functionality of R
    - This is particularly helpful when no R package/function is available
- R markdown for the creation of Microsoft Word and LaTeX documents
- R and LaTeX can work together to produce presentations (in addition to documents) with R Markdown

Additional Resources

Introduction to data.table

- https://cran.r-project.org/web/packages/data.table/vignettes/datatable-intro.html

Applied Econometrics with R

- Kleiber, C., & Zeileis, A. (2008). *Applied econometrics with R*. Springer Science & Business Media.

R for Medical Decision Making

- Jalal, H., Pechlivanoglou, P., Krijkamp, E., Alarid-Escudero, F., Enns, E., & Hunink, M. M. (2017). An overview of R in health decision sciences. *Medical decision making*, *37*(7), 735-746.
- Krijkamp, E. M., Alarid-Escudero, F., Enns, E. A., Jalal, H. J., Hunink, M. M., & Pechlivanoglou, P. (2018). Microsimulation modeling for health decision sciences using R: a tutorial.
- Alarid-Escudero, F., Krijkamp, E. M., Enns, E. A., Yang, A., Hunink, M. G., Pechlivanoglou, P., & Jalal, H. (2021). A Tutorial on time-dependent cohort state-transition models in R using a cost-effectiveness analysis example. *arXiv preprint arXiv:2108.13552*.

Posit Cheatsheets

- https://posit.co/resources/cheatsheets/

Thank you for listening

Good luck with R!