Exploratory Data Analysis II

PART I

Course Contents

1. 2022-12-07: Introduction: About the course [lead by TK] - An introduction to open and public data, and data science
2. 2022-12-14: Exploratory Data Analysis (EDA) 1 [lead by hs]
   - R Basics with RStudio and/or RStudio.cloud; Toy Data
3. **2022-12-21: Exploratory Data Analysis (EDA) 2 [lead by hs]**
   - R Markdown, `tidyverse` I: `dplyr`; WDI
4. 2023-01-11: Exploratory Data Analysis (EDA) 3 [lead by hs]
   - `tidyverse`II: `readr`, `ggplot2`; Public Data, WDI, WIR, etc
5. 2023-01-18: Exploratory Data Analysis (EDA) 4 [lead by hs]
   - `tidyverse` III: `tidyr`, etc.; WDI, WIR, etc
6. 2023-01-25: Exploratory Data Analysis (EDA) 5 [lead by hs]
   - `tidyverse` IV; WDI, WIR, etc
7. 2023-02-01: Introduction to PPDAC
   - Problem-Plan-Data-Analysis-Conclusion Cycle: [lead by TK]
8. 2023-02-08: Model building I [lead by TK] -Collecting and visualizing data and Introduction to WDI (World Development Indicators by World Bank)
9. 2023-02-15: Model building II [lead by TK] -Analyzing data and communications
10. 2023-02-22: Project Presentation

Review

- R on R Studio/Posit Cloud (RStudio Cloud)
- Three ways to run codes
  1. Console
  2. R Script
  3. Code Chunk in R Notebook
- Packages
  1. `tidyverse`
  2. `rmarkdown`
  3. `gapminder`

EDA (A diagram from R4DS by H.W. and G.G.) **Today**: R Markdown and `dplyr`

R Markdown

What is R Markdown Notebook

R Markdown provides an authoring framework for data science. You can use a single R Markdown file to both

- save and execute code

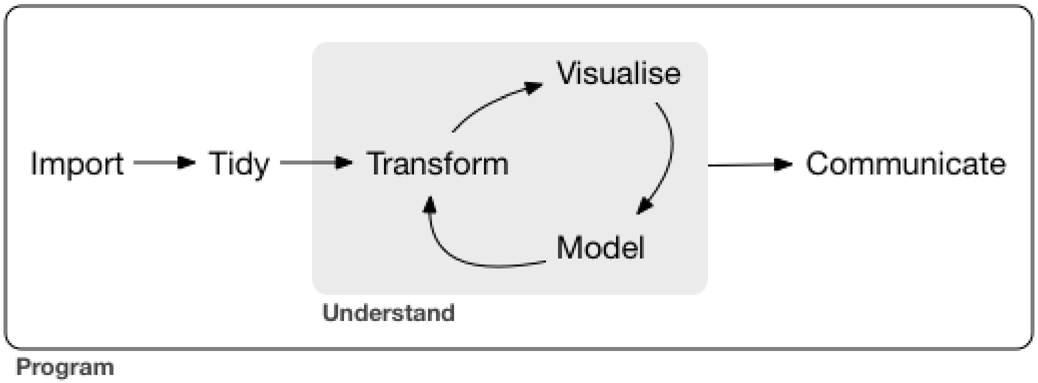

Figure 1: EDA from r4ds

- generate high quality reports that can be shared with an audience

R Notebooks are an implementation of Literate Programming that allows for direct interaction with R while producing a reproducible document with publication-quality output.

An R Notebook is an R Markdown document with chunks that can be executed independently and interactively, with output visible immediately beneath the input.

(Reference: R Markdown: The Definitive Guide, 3.2 Notebook)

Two Goodies

- **Important and Essential**: Reproducible Research with Literate Programming
- **Useful to Render Vairous Formats**: R Notebook (HTML), R Markdown (HTML), PDF, MS Word, MS Powerpoint, Ioslides Presentation (HTML), Slidy Presentatoin (HTML), Beamer Presentation (PDF)

Records of EDA and Communication

1. Memo on scratch paper: R Scripts
2. Record on a notebook: R Notebook (a type of an R Markdown format)
3. Short paper or a digital communication: R Notebook
4. Paper or a report: R Markdown (html, pdf, or MS Word)
5. Presentation

- R Markdown with a presentation format (html, pdf, or PowerPoint)

6. Publication of a Book

- BOOKDOWN: Write HTML, PDF, ePub, and Kindle books with R Markdown. Free online document is provided in pdf as well
- Arxive Page

R Studio Setup

1. Start R Studio
2. Create a Project
3. Tool > Install Packages `rmarkdown`
    - Or on Console: `install.packages("rmarkdown")`
4. Tool > Install Packages `tinytex` (for pdf generation)

5. Let’s try!
    a. File > New File > R Notebook
    b. Save with a file name, say, test-notebook
    c. Preview by [Preview] button
    d. Run Code Chunk `plot(cars)` and then Preview again.
    e. Knit PDF, Word (and HTML)

Note: R Notebooks are relatively new feature of RStudio and are only available in version 1.0 or higher of RStudio.

Default YAML: R Notebook, HTML, PDF, WORD

```
---
title: "The Title of This R Notebook"
author: "Your Name"
date: "2021-12-22"
output:
  html_notebook: default
  html_document: default
  word_document: default
  pdf_document: default
---
```

- Original format is `output: html_document`
- Indention matters in YAML. So it is safer to copy and paste `output:` to `pdf_document: default`
- R Notebook is also an HTML format, so html_notebook part may disappear after knitting in HTML.

An Example of R Notebooks

1. Moodle: QALL401 2021W 2021-12-22 Examples of R Notebook

2. Open the file

- R Codes with Outputs
- Headings, Links, Explanations, etc.
- [Hide] button, and [Code] button with a menu
- Choose: Download Rmd

3. Save as “*file_name*.nb.html” in your R project directory

4. Download and Save “jhu_covid.Rmd”. (Or, open the file in editor)

5. Preview by [Preview] button.

6. Knit to other formats, e.g. Word under [Preview] button

N.B. If Step 4 does not work, create a new R Notebook and copy and paste [R Markdown Source File] in Moodle

Knit, Notebook Mode and Preview of Default.Rmd

1. Knit to HTML
2. Knit to PDF (require TeX system, install `tinytex` package)
3. Knit to Word
4. Controlling a code chunk and its output

- Highlight and run
- Run all chunks above
- Expand, collapse and clear output
- Show output in other window

- Modify chunk options

5. Output Options:

- Notebook, HTML, PDF, Word
- General, Figures and Advanced

`yaml` - YAML Ain't a Markup Language - Example

```
---
title: "File Name --Subtitle--"
author: "My Name"
date: "2021/12/22"
output:
  html_notebook:
    number_sections: yes
    toc: yes
    toc_float: yes
  word_document:
    fig_caption: yes
    fig_height: 5
    fig_width: 6
#    reference_docx: word-styles-reference-01.docx
---
```

R Markdown: Quick References (See Moodle)

- R Studio Help (menu bar) > Markdown Quick Reference
- R Studio Help (menu bar) > Cheat Sheet
  - R Markdown Cheat Sheet
  - R Markdown Reference Guide
- R Markdown: The Definitive Guide by Yihui Xie, J. J. Allaire, Garrett Grolemund
- In Textbook: R for Data Science: Communicate
- Markdown: R Markdown is based on the Markdown language of Pandoc
  - Pandoc's Markdown: Detailed Information
  - Markdown Tutorials: Interactive Practicum
  - DARING FIREBALL: Markdown (detailed explanation and editor as Dingus)

Markdown Language – or use WYSIWYG editor

- Headers: #, ##, ###, ####
- Lists: 1. 2. . . . , *
- Links: linked phrase
- Images: `![alt text](figures/filename.jpg)`
- Block quotes" > (block)
- LaTeX equations: e.g. `$\frac{a}{b}$` for $\frac{a}{b}$
- Horizontal rules: Three or more asterisks or dashes (*** or - - -)
- Tables
- Footnotes
- Bibliographies and Citations
- Slide breaks
- *Italicized text* by `_italic_`, **Bold text** by `**bold**`
- Superscripts, Subscripts, Strikethrough text

MS Word: Happy collaboration with Rmd to docx

- Use R Markdown to create a Word document
  - Save as: “word-styles-reference-01.docx’’
- Edit the Word styles
  - Edit the styles of the file “word-styles-reference-01.docx’’ .
- Save this document as your style reference docx file
- Format an Rmd report using the styles reference docx file

```
---
title: "Test Report"
author: "Your Name"
date: "January 6, 2021"
output:
  word_document:
    reference_docx: word-styles-reference-01.docx
---
```

Why R Markdown?

R Markdown Cheat Sheet

- .Rmd files: An R Markdown (.Rmd) file is a record of your research. It contains
  1. the code that a scientist needs to reproduce your work along with
  2. the narration that a reader needs to understand your work.
- Reproducible Research: At the click of a button, or the type of a command, you can rerun the code in an R Markdown file to Rmd reproduce your work and export the results as a finished report.
- Dynamic Documents: You can choose to export the finished report as a html, nb.html, pdf, MS Word, ODT, RTF, or markdown document; or as a html or pdf based slide show.

Literate Programming by D. Knuth

Literate programming is an approach to programming introduced by Donald Knuth in which a program is given as an explanation of the program logic in a natural language, such as English, interspersed with snippets of macros and traditional source code, from which a compilable source code can be generated

D. Knuth Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do.

Reproducible Research - Quote from a Coursera Course

Reproducible Research Reproducible research is the idea that data analyses, and more generally, scientific claims, are published with their data and software code so that others may verify the findings and build upon them. The need for reproducibility is increasing dramatically as data analyses become more complex, involving larger datasets and more sophisticated computations. Reproducibility allows for people to focus on the actual content of a data analysis, rather than on superficial details reported in a written summary. In addition, reproducibility makes an analysis more useful to others because the data and code that actually conducted the analysis are available.

R Markdown workflow, R for Data Science

R Markdown is also important because it so tightly integrates prose and code. This makes it a great analysis notebook because it lets you develop code and record your thoughts. It:

- Records what you did and why you did it. Regardless of how great your memory is, if you don't record what you do, there will come a time when you have forgotten important details. Write them down so you don't forget!
- Supports rigorous thinking. You are more likely to come up with a strong analysis if you record your thoughts as you go, and continue to reflect on them. This also saves you time when you eventually write up your analysis to share with others.
- Helps others understand your work. It is rare to do data analysis by yourself, and you'll often be working as part of a team. A lab notebook helps you share why you did it with your colleagues or lab mates.

Examples of yaml

```
---
title: "R Notebook"
output: html_notebook
---
```

Default + author + date The format of date can be changed.

```
---
title: "Title of the Notebook"
author: "Your Name"
date: "2021-12-22"
output: html_notebook
---
```

Examples

Notebook of Coronavirus

```
---
title: "A Study of Cases of Coronavirus Pandemic"
author: "Hiroshi Suzuki"
date: "2021/12/22"
output:
  html_notebook:
    number_sections: yes
    toc: yes
    toc_float: yes
---
```

Notebook of Coronavirus + pdf + word

```
---
title: "A Study of Cases of Coronavirus Pandemic"
author: "Hiroshi Suzuki"
date: "2021/12/22"
output:
  html_notebook:
    number_sections: yes
    toc: yes
    toc_float: yes
  pdf_document:
    toc: true
    number_sections: true
  word_document: default
---
```

Edit the word file file_show.docx to create my-styles.docx, e.g., a4-my-styles.docx changed the paper size to A4 from US letter.

```
---
title: "A Study of Cases of Coronavirus Pandemic"
author: "Hiroshi Suzuki"
date: "2021/12/22"
output:
  html_notebook:
    number_sections: yes
    toc: yes
    toc_float: yes
  pdf_document:
    toc: true
    number_sections: true
  word_document:
    reference_docx: a4-my-styles.docx
---
```

Part II: Data Transforamtion with `dplyr`

`dplyr` Overview

dplyr is a grammar of data manipulation, providing a consistent set of verbs that help you solve the most common data manipulation challenges:

- `select()` picks variables based on their names.
- `filter()` picks cases based on their values.
- `mutate()` adds new variables that are functions of existing variables
- `summarise()` reduces multiple values down to a single summary.
- `arrange()` changes the ordering of the rows.
- `group_by()` takes an existing tbl and converts it into a grouped tbl.

You can learn more about them in vignette("dplyr"). As well as these single-table verbs, dplyr also provides a variety of two-table verbs, which you can learn about in vignette("two-table").

If you are new to dplyr, the best place to start is the data transformation chapter in R for data science.

`select`: Subset columns using their names and types

Helper Function	Use	Example
-	Columns except	select(babynames, -prop)
:	Columns between (inclusive)	select(babynames, year:n)
contains()	Columns that contains a string	select(babynames, contains("n"))
ends_with()	Columns that ends with a string	select(babynames, ends_with("n"))
matches()	Columns that matches a regex	select(babynames, matches("n"))
num_range()	Columns with a numerical suffix in the range	Not applicable with babynames

Helper Function	Use	Example
one_of()	Columns whose name appear in the given set	select(babynames, one_of(c("sex", "gender")))
starts_with()	Columns that starts with a string	select(babynames, starts_with("n"))

`filter`: Subset rows using column values

Logical operator	tests	Example
>	Is x greater than y?	x > y
>=	Is x greater than or equal to y?	x >= y
<	Is x less than y?	x < y
<=	Is x less than or equal to y?	x <= y
==	Is x equal to y?	x == y
!=	Is x not equal to y?	x != y
is.na()	Is x an NA?	is.na(x)
!is.na()	Is x not an NA?	!is.na(x)

`arrange and Pipe %>%`

- `arrange()` orders the rows of a data frame by the values of selected columns.

Unlike other `dplyr` verbs, `arrange()` largely ignores grouping; you need to explicitly mention grouping variables (‘or use .by_group = TRUE) in order to group by them, and functions of variables are evaluated once per data frame, not once per group.

- `pipes` in R for Data Science.

`mutate`

- Create, modify, and delete columns
- Useful mutate functions
  - +, -, log(), etc., for their usual mathematical meanings
  - lead(), lag()
  - dense_rank(), min_rank(), percent_rank(), row_number(), cume_dist(), ntile()
  - cumsum(), cummean(), cummin(), cummax(), cumany(), cumall()
  - na_if(), coalesce()### `group_by()` and `summarise()`

- `group_by`

- `summarise` or `summarize`

```
iris %>%
  group_by(Species) %>%
  summarize(sl = mean(Sepal.Length), sw = mean(Sepal.Width),
  pl = mean(Petal.Length), pw = mean(Petal.Width))
```

```
## # A tibble: 3 x 5
##   Species       sl    sw    pl    pw
##   <fct>      <dbl> <dbl> <dbl> <dbl>
## 1 setosa      5.01  3.43  1.46 0.246
## 2 versicolor  5.94  2.77  4.26 1.33
## 3 virginica   6.59  2.97  5.55 2.03
```

Summary functions So far our summarise() examples have relied on sum(), max(), and mean(). But you can use any function in summarise() so long as it meets one criteria: the function must take a vector of values as input and return a single value as output. Functions that do this are known as summary functions and they are common in the field of descriptive statistics. Some of the most useful summary functions include:

1. Measures of location - mean(x), median(x), quantile(x, 0.25), min(x), and max(x)
2. Measures of spread - sd(x), var(x), IQR(x), and mad(x)
3. Measures of position - first(x), nth(x, 2), and last(x)
4. Counts - n_distinct(x) and n(), which takes no arguments, and returns the size of the current group or data frame.
5. Counts and proportions of logical values - sum(!is.na(x)), which counts the number of TRUEs returned by a logical test; mean(y == 0), which returns the proportion of TRUEs returned by a logical test.

- if_else(), recode(), case_when()

Examples

Introduction to WDI

4. More Examples

Learning Resources, III

- Textbook: R for Data Science, Part II Explore

RStudio Primers: See References in Moodle at the bottom

1. The Basics – r4ds: Explore, I

- Visualization Basics
- Programming Basics

2. **Work with Data** – r4ds: Wrangle, I

- **Working with Tibbles**
- **Isolating Data with dplyr**
- **Deriving Information with dplyr**

3. Visualize Data – r4ds: Explore, II
4. Tidy Your Data – r4ds: Wrangle, II
5. Iterate – r4ds: Program
6. Write Functions – r4ds: Program

Learning Resources, EDA2-1

- Textbook: R for Data Science, Part V Communicate
- R Markdown: The Definitive Guide by Yihui Xie, J. J. Allaire, Garrett Grolemund [Last Revised: 2020-12-14]
- BOOKDOWN: Write HTML, PDF, ePub, and Kindle books with R Markdown. Free online document is provided in pdf as well
  - Arxive Page
- RMarkdown for Scientists by Nicholas Tierney [Last Revised: 2020-09-09]
- Report Writing for Data Science in R by Roger Peng
- Social Science Computing Cooperative at the University of Wisconsin
  - R for Researchers: R Markdown [Last Revised: 2015-04-16]

Practicum: R Markdown and R Notebook

1. Install `rmarkdown`, `tinytex`, `tidyverse` (and `timetk`)
2. Try RMarkdown: HTML, PDF, Word
3. Try R Notebook:

- Code Chunk
- RStudio Help
- Visual Editor
- Download from Moodle
- Export and import files
- *file name*.nb.html and *file name*.Rmd

4. YAML
5. Shared link to RStudio.cloud: https://moodle3.icu.ac.jp/mod/url/view.php?id=185785

NB. `Sys.setenv(LANG = "en")`

`ggplot2` Overview

`ggplot2` is a system for declaratively creating graphics, based on The Grammar of Graphics. You provide the data, tell ggplot2 how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

Examples

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```

```
ggplot(data = mpg) +
  geom_boxplot(mapping = aes(x = class, y = hwy))
```

Template

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

Basics of Fundamentals of Statistics

R Commands Related to R Basics

- Fundamentals of Statistics: statistical measurements such as
  - mean: `mean()` or `mean(x, na.rm = TRUE)`

    - median: `median()` or `median(x, na.rm = TRUE)`
    - quantile: `quantile()` or `quantile(x, na.rm = TRUE)`
    - variance: `var()` or `var(x, na.rm = TRUE)`
    - standard deviation: `sd()`
    - covariance: `cov()`
    - correlation: `cor()`
- summary()

The Week Two Assignment (in Moodle)

- Pick two data from the built-in dataset. (`library(help = "datasets")` or go to the site The R Datasets Package)
    - One of them can be `iris` but do not choose `cars` or `AirPassengers`.
    - `ggplot2` examples of `cars`, `iris` and `AirPassengers` are given below.
- Create an R Notebook of a Data Analysis containing the following and submit the rendered HTML file (*file name*.nb.html):
    1. title, date, and author, i.e., Your Name
    2. an explanation of the data and the variables
    3. at least one code chunk containing the following:

    - `head()`, `str()`

    4. for each dataset, at least one code chunk containing graphs using ggplot2. Please try at least two `geoms`:

    - `geom_hist()`, `geom_boxplot()`, `goem_col()`, etc.
    - `geom_line()`, `geom_point()`, etc.

    5. your findings and/or questions
    6. file name: ID.nb.html, e.g. 123456.nb.html
    7. option: `median()`, `mean()`, `sd()` of a quantitative (numeric) variable, `cor()` of two quantitative (numeric) variables (or a correlation table)
- Submit your R Notebook file to Moodle (The Third Assignment) by 2021-01-11 23:59:00

Note on R Notebook

Please note the following.

- There are essentially three modes: R Scripts, R Notebook and R Markdown.
- R Notebook is a special type of R Markdown but please use R Notebook at least for Suzuki's assignments.
- To start, choose R Notebook from New File in the File Menu. If you started with R Markdown, please switch it with the Preview button hidden under the triangle on the right of the knit button.
- The file we preview has the name *file name*.nb.html. For example if the original file name is a3_12345.Rmd, then a3_12345.nb.html is created. You can check it using Files tab.
- When you preview R Notebook, on the top right, you can find Code button. If you press it you also can find download Rmd, which is the source of R Notebook you edited. In this way we can share both the outputs and the source.
- One difficulty is that you cannot include the outputs of the code chunk in the preview.
- Check Preview on Save and/or save the file before preview, i.e., pressing the preview button.
- Select Run all under Run button. Then all outputs apear on your editor and all outputs will appear in your preview.
- If some of your code chunks have problems, run each code chunk from top to bottom so that all outputs appear in your editor or viewer.
- When you share your R Notebook, do not forget to share *file name*.nb.html.

- If you have a *file name*.nb.html, then find it from Files in R Studio, you can automatically create *file name*.Rmd to edit the source.
- To create a fancy document with R Notebook, see the Markdown Quick Reference under Help on top menu for the editor. If you are using Visual Editor using A bottun on top left pane, see https://rstudio.github.io/visual-markdown-editing/.
- R Studio introduced Visual Editor last year. It seems to be stable but it is not perfect to go back and forth from the original editor using tags. I always use the original editor and I am confident on all the functions of it but I do not have much experience on Visual Editor.

Responses to the Week Three Assignment

Setup

We load two packages; `datasets` and `ggplot2`. The `datasets` are loaded automatically and you do not need the first line of the followiong code chunk. But it is safer to include it because some data names are used previously for different purposes.

```
library(datasets)
library(ggplot2)
```

For explanation, we use the following population data of WDI.

```
library(WDI)
pop <- WDI(
  country = c("NG", "BD", "RU", "MX", "JP"),
  indicator = c(population = "SP.POP.TOTL"),
  start = 1960, end = 2020)
head(pop)
```

```
##      country iso2c iso3c year population
## 1 Bangladesh    BD   BGD 2020  164689383
## 2 Bangladesh    BD   BGD 2019  163046173
## 3 Bangladesh    BD   BGD 2018  161376713
## 4 Bangladesh    BD   BGD 2017  159685421
## 5 Bangladesh    BD   BGD 2016  157977151
## 6 Bangladesh    BD   BGD 2015  156256287
```

Assignments - See Moodle

1. Assignment Week 2-1: Introduction Plus Forum

- Due: Tuesday, 20 December 2022, 11:59 PM

2. Assignment Week 2-2: Quiz 1 on R Basics

- Due: Tuesday, 20 December 2022, 11:59 PM

:::

Questions

- List questions based on this data.
- What do you want to see?
- What kind of chart do you want to construct?