VII. Data analysis with R

1. Programming and data analytics

- The exciting world of programming
  - The R-versus-Python debate
- Programming as a data analyst
  - Programming languages
  - From spreadsheets to SQL to R
- Learn programming using RStudio
  - Cloud access to RStudio
    - load library
      - **install.packages("tidyverse")** and press **Enter** (Windows)
      - **library(tidyverse)** and press **Enter**
      - **library(lubridate)** and press **Enter**
  - Get started in RStudio Desktop
    - To install: **install.packages("tidyverse")** and click **Run**
    - Load library
      - **library(tidyverse)** and click **Run**/ Enter
      - **library(lubridate)** and click **Run**.
  - Connecting with other analysts in the R community

2. Programming using RStudio

- Understand basic programming concepts
  - # comment in your code
  - Vectors and lists in R
    - c(x, y, z, ...)
  - Dates and times in R
    - today()
    - now()
    - strings to a date/time data type
  - Other common data structures
    - Data frames -->data.frame()
    - Files
      - dir.create ("destination_folder")
      - file.copy() — file.copy ("new_text_file.txt" , "destination_folder")
      - delete R files using the unlink() — unlink ("some_file.csv")
    - Matrices
      - matrix(c(3:8), nrow = 2)
      - matrix(c(3:8), **ncol** = 2)
- Explore coding in R
  - Operators and calculations
    - Decimal values like 4.5 are called **numerics**.
      - Whole numbers like 4 are called **integers**. **Integers** are also numerics.
      - Boolean values (TRUE or FALSE) are called logical
        - my_logical <- TRUE
      - **Text** (or **string**) values are called **characters**:
        - my_character <- "some text"
    - x <- 10 --> gán x = 10
    - types of operators: assignment
      - my_apples <- 5
      - my_fruit <-my_oranges + my_apples
    - 28 %%6 -->mod
    - 2^5 --> lũy thừa
  - Logical operators and conditional statements
    - AND(& or && in R)
    - OR (| or || in R)
    - NOT (!)
    - (non-zero numbers are taken as TRUE)
    - Conditional statements
      - if()
        - if (condition) {
        - expr
        - }
      - else if()
        - } else if (condition2) {
        - expr2
      - else()
        - } else {
        - expr3
        - }
  - R sandbox Activity
    - ## Step 1: Using `R packages`
      - install.packages("tidyverse")
      - library(tidyverse)
    - # Step 2: Viewing data
      - head(diamonds)
      - return summaries of each column
        - str(diamonds)
        - glimpse(diamonds)
      - returns a list of column names — colnames(diamonds)
    - ## Step 3: Cleaning data
      - rename the columns
        - rename(diamonds, carat_new = carat)
        - rename(diamonds, carat_new = carat, cut_new = cut)
      - summarizing your data — summarize(diamonds, mean_carat = mean (carat))
    - ## Step 4: Visualizing data
      - ggplot(data = diamonds, aes(x = carat, y = price)) + geom_point()
      - geom_point() --> a scatterplot
      - ggplot(data = diamonds, aes(x = carat, y = price , color = cut)) + geom_point() — change the color as the cut of the diamond
      - ggplot(data = diamonds, aes(x = carat, y = price, color = cut)) + geom_point() + facet_wrap(~cut) — create a different plot for each type of cut
    - ## Step 5: Documentation
      - ***R markdown***
      - **##** to the beginning of the line
      - **knit** button to export your work
- Learning about R packages
  - Available R packages
  - updates by running: **tidyverse_update()**
  - Hands-On Activity: Installing and loading tidyverse
    - Read tidyverse vignettes: browseVignettes()
    - browseVignettes("ggplot2")
- Explore the **tidyverse**
  - ggplot2 (), --> data visualization -->plot
  - tidyr() -->clean data
  - readr() -->importing data — read_csv()
  - dplyr() -->data manipulation tasks
    - `arrange(filter($storage_1, inventory >= 40), count)`
  - tibble() -->data frames
  - purrr() -->functions and vectors -
  - stringr()
  - forcats()
  - tidyverse flow chart
  - **Working with pipes**
  - R resources for more help

3. Working with data in R

- R data frames = spreadsheet
- Working with data frames
- Hands-On Activity: Create your own data frame
  - ## Step 1: Load packages
  - ## Step 2: Create data frame
    - data.frame()
    - names <- c("", "", "", "", "")
    - age <- c("", "", "", "", "")
    - people <- data.frame (names, age)
  - **## Step 3: inspect the data frame**
    - head(people)
    - str(people) --> get a rapid overview of your data
    - glimpse(people)
    - colnames(people)
    - to create a new variable — **mutate** (people, age_in_20 = age + 20)
  - ## Step 4: Try it yourself (DataCamp)
    - Selection of data frame elements
    - my_df[1,2] selects the value at the first row and second column in my_df
    - my_df[1,] selects all elements of the first row ,
    - planets_df[1:3,"type"] --> lấy liên column, thay thế cho từ tự column
    - to select an entire column
      - planets_df[,3]
      - planets_df[,"diameter"]
      - planets_df$diameter
- More about tibbles
  - library(tidyverse)
  - data(diamonds)
  - view(diamonds)
- Data-import basics
  - The readr package
  - The readr package
    - read_csv(): comma-separated values (.csv) files
    - read_tsv(): tab-separated values files
    - read_delim(): general delimited files
    - read_fwf(): fixed-width files
    - read_table(): tabular files where columns are separated by white-space
    - read_log(): web log files
  - To import spreadsheet (excel) data into R — library(readxl)
- Hands-On Activity: Importing and working with data
  - ## Step 1: Load packages
    - install.packages("tidyverse")
    - library(tidyverse)
  - ## Step 2: Import data — bookings_df <- read_csv("hotel_bookings.csv")
  - ## Step 3: Inspect & clean data
    - head(bookings_df)
    - str(bookings_df)
    - colnames(bookings_df)
    - new_df <- select(bookings_df, `adr`, adults)
    - mutate(new_df, total = `adr` / adults)
  - # Step 4: Import your own data
- **Cleaning data**
  - Cleaning up with the basics
  - File-naming conventions
  - More on R operators
    - Not equal to: !=
    - Equal to: ==
    - **AND (&) and OR (|)**
    - AND (&&) and OR (||)
    - Logical NOT (!)
  - Organize your data
    - arrange().
    - group by()
    - filter()
  - Hands-On Activity: Cleaning data in R
    - ## Step 1: Load packages
      - install.packages("tidyverse")
      - install.packages("skimr")
      - install.packages("janitor")
      - load library
        - library(tidyverse)
        - library(skimr)
        - library(janitor)
    - ## Step 2: Import data — bookings_df <- read_csv("hotel_bookings.csv")
    - ## Step 3: Getting to know your data
      - head(bookings_df)
      - str(bookings_df)
      - glimpse(bookings_df)
      - colnames(bookings_df)
      - skim_without_charts (bookings_df) -->to look overview about the data
      - skim_without_charts(TootHGrowth)
      - **unique**(iris$Species) --> xem có bao nhiêu Species khác nhau trong dữ liệu iris
    - ## Step 4: Cleaning your data
      - Create a new data frame with just those columns
      - trimmed_df <- bookings_df %>%
        - select (hotel, is_canceled, lead_time)
      - rename (hotel_type = hotel) -->rename hotel
      - `rename(quarter1_sales = q1_sales)`
      - unite(arrival_month_year, c("arrival_date_month", "arrival_date_year"), sep = " ")
    - ## Step 5: Another way of doing things
      - make changes to your columns
      - example_df <- bookings_df %>%
        - mutate(guests = adults + children + babies)
      - Calculate the total number
        - example_df <- bookings_df %>%
        - summarize (number_canceled = sum(is_canceled), average_lead_time = mean(lead_time))
  - Transforming data (tạo ra cột mới = biến mới)
    - separate()
    - unite() --> to merge columns together (to create a new column) (tạo cột mới = biến mới)
      - unite(retail, "price", price_dollars, price_cents , sep=".")
    - mutate() --> to add columns to calculation (mutate: sử dụng để tạo ra biến mới)
      - ***mutate***(salary_data, total_wages = standard_wages + overtime_wages)
      - ```{r}
      - example_df <- bookings_df %>%
        - **mutate**(guests = adults + children + babies)
      - head(example_df)
      - ```
  - Wide to long with tidyr (chức năng như Pivot trong Excel)
    - pivot_longer() -->more rows
    - pivot_wider() -->more columns
  - Clean, organize, and transform data with R
    - Clean
      - **glimpse()**
      - **rename()**
      - **rename_with()**
      - **skim_without_charts()**
      - **clean_names()**
      - **select()**
    - Organize
      - max()
      - arrange() — penguins %>% arrange(bill_length_mm)
      - drop_na()
      - mean()
      - summarize()
      - filter()
      - group_by()
    - Transform
      - **separate()**
      - **unite()**
      - **mutate()**
- **Take a closer look at the data**
  - Same data, different outcome
  - The bias function: bias()
    - A score of 0.8/0.7 indicates that the model is **biased.**
    - The closer the score is to zero (-->0.01), the less likely it is that the model is biased .
  - Working with biased data — sample()
  - Hands-On Activity: Changing your data
    - ## Step 1: Load packages
      - install packages
        - install.packages("tidyverse")
        - install.packages("skimr")
        - install.packages("janitor")
      - load packages
        - library(tidyverse)
        - library(skimr)
        - library(janitor)
    - ## Step 2: Import data — hotel_bookings <-read_csv("hotel_bookings.csv")
    - ## Step 3: Getting to know your data
      - head(hotel_bookings)
      - str(hotel_bookings)
      - glimpse(hotel_bookings)
      - colnames(hotel_bookings)
    - ## Step 4: Manipulating your data
      - to sort data
        - arrange(hotel_bookings,)
        - arrange (hotel_bookings, desc (lead_time)) (lead_time từ cao xuống thấp)
      - create new data frame — hotel_bookings_v2 <-arrange(hotel_bookings, desc(lead_time))
      - find out the max/min
        - max(hotel_bookings$lead_time)
        - min(hotel_bookings$lead_time)
      - how early you should run promotions — mean (hotel_bookings$lead_time)
      - creating a new dataset + only contains City Hotel
        - hotel_bookings_city <- filter(hotel_bookings, hotel_bookings $hotel=="City Hotel")
        - mean(hotel_bookings_city$lead_time)
        - hotel_summary <-
          - hotel_bookings %>%
          - group_by (hotel) %>%
          - **summarise** (average_lead_time=mean(lead_time), min_lead_time=min(lead_time), max_lead_time=max(lead_time))
          - summarize : đóng vai trò như lọc (sort) ra 1 bảng

4. More about visualizations, aesthetics, and annotations

- Create data visualizations in R
  - Hands-On Activity: Visualizing data with ggplot2
    - Prepare your data
      - install.packages("ggplot2")
      - install.packages("palmerpenguins")
      - library(ggplot2)
      - library(palmerpenguins)
      - **examine the data frame**
        - data(penguins)
        - View(penguins)
    - Create a plot in ggplot2
      - ggplot(data = penguins) +
        - geom_point() → scatter plots (*vẽ phân tán*)
        - geom_bar → bar charts
      - ggplot(data = penguins) + geom_point (mapping = aes (x = flipper_length_mm, y = body_mass_g))
  - Getting started with ggplot()
  - Common problems when visualizing in R
  - Hands-On Activity: Using ggplot
    - ## Step 1: Import your data — hotel_bookings <- read.csv("hotel_bookings.csv")
    - ## Step 2: Look at a sample of your data
      - head(hotel_bookings)
      - colnames(hotel_bookings)
    - ## Step 3: Install and load the 'ggplot2' package
      - install.packages("ggplot2")
      - library(ggplot2)
    - ## Step 4: Begin creating a plot
      - ggplot(data = hotel_bookings) +
      - geom_point(mapping = aes(x = lead_time, y = children))
    - ## Step 5: Try it on your own
      - ggplot(data = hotel_bookings) +
      - geom_point (mapping = aes (x = stays_in_weekend_nights, y =children))
- Explore aesthetics in analysis
  - Enhancing visualizations in R
  - Doing more with ggplot
    - geom_bar()
    - geom_smooth()
  - Aesthetics and facets
  - **Hands-on: Aesthetis and visualizations**
    - hotel_bookings <- read.csv("hotel_bookings.csv")
    - head(hotel_bookings)
    - colnames(hotel_bookings)
    - install.packages("ggplot2")
    - library(ggplot2)
    - ggplot(data = hotel_bookings) +
    - geom_bar(mapping = aes(x = distribution_channel))
  - Filtering and plots
    - data %>%
    - filter(variable1 == "DS") %>%
    - ggplot(aes(x = weight, y = variable2, colour = variable1)) +
    - geom_point(alpha = 0.3, position = position_jitter()) + stat_smooth(method = "lm")
  - Hands-on: **Filtering** and Plots
    - install.packages("tidyverse")
    - library(tidyverse)
    - onlineta_city_hotels <- filter(hotel_bookings, (hotel=="City Hotel" & hotel_bookings$market_segment=="Online TA"))
    - ggplot(data = onlineta_city_hotels) + geom_point (mapping = aes (x = lead_time, y = children))
- **Annotate and save visualizations**
  - Annotation layer
  - Saving your visualization — ggsave("children_vs_lead_time.png")
  - **Hands-On** Activity: Annotating and saving visualizations
    - ## Step 1: Import your data
    - ## Step 2: Refresh Your Memory
      - look at data()
      - look at column names()
    - ## Step 3: Install and load the 'ggplot2' and 'tidyverse' packages
    - **## Step 4: Annotating your chart**
      - faceting a plot with a title
      - ggplot(data = hotel_bookings) +
        - geom_bar(mapping = aes(x = market_segment)) +
        - facet_wrap(~hotel) +
        - labs (title=" Comparison of market segments by hotel type for hotel bookings ")
      - (r **earliest** year)
        - min (hotel_bookings$arrival_date_year)
      - (r latest year)
        - max(hotel_bookings$arrival_date_year)
      - save them as variables
        - mindate <- min(hotel_bookings$arrival_date_year)
        - maxdate <- max(hotel_bookings$arrival_date_year)
      - ```{r city bar chart with timeframe- subtitle}
      - ggplot(data = hotel_bookings) +
        - geom_bar(mapping = aes(x = market_segment)) +
        - facet_wrap(~hotel) +
        - labs(title="Comparison of market segments by hotel type for hotel bookings", subtitle =paste0("Data from: ", mindate, " to ", maxdate))
      - ```{r city bar chart with timeframe as caption}
      - ggplot (data = hotel_bookings) +
        - geom_bar (mapping = aes(x = market_segment)) +
        - facet_wrap(~hotel) +
        - labs (title="Comparison of market segments by hotel type for hotel bookings", caption =paste0 ("Data from: ", mindate, " to ", maxdate))
    - to add a caption to the plot — labs (title="Comparison of market segments by hotel type for hotel bookings", caption =paste0 ("Data from: ", mindate, " to ", maxdate))
    - to put a text label inside the grid of your plot — annotate()

5. Documentation and reports

- **Develop documentation and reports in RStudio**
  - Documentation and reports
  - Overview of R Markdown
  - Using R Markdown in RStudio — install.packages("rmarkdown")
- Create R Markdown documents
  - Structure of markdown documents
  - Even more document elements
    - * HTML
    - [link](http://linkedin.com/in/.../)
    - ![example_1](r-project.org/logo/Rlogo.png) --> link to image
- Understand code chunks and exports
  - Code chunks
  - Exporting documentation
  - Hands-On Activity: Exporting your R Markdown **notebook** — Export your R Markdown notebook
  - Hands-On Activity: Using R Markdown **templates**