BSDS-201 - R for Data Science

Course Instructor - Ms. Manpreet Bhatia

Gauri Sharan - BSc Data Science (Sem 3)

January 6, 2024

Outline

R Programming Overview

- R is a powerful programming language and environment for statistical computing and data analysis.
- Developed by statisticians and data scientists for various data-related tasks.
- Open-source and has a vast and active user community.

Basic R Syntax

```
# Assigning variables
x <- 5
y <- c(1, 2, 3, 4, 5)

# Printing
print(x)
print(y)

# Basic arithmetic
result <- x + sum(y)
```

- R uses the assignment operator <- or = for variable assignment.
- Comments are preceded by the # symbol.

Data Types in R

```
# Numeric
num_var <- 42

# Character
char_var <- "Hello, R!"

# Logical
logical_var <- TRUE

# Vector
vector_var <- c(1, 2, 3, 4, 5)
```

- R supports various data types, including numeric, character, logical, and vectors.

Data Manipulation in R

```
# Create a data frame
data <- data.frame(
  Name = c("John", "Mary", "Alice"),
  Age = c(25, 30, 22),
  Height = c(180, 165, 175)
)

# Filter data
filtered_data <- subset(data, Age > 22)
```

- R is known for its powerful data manipulation capabilities.
- The `data.frame` is a common structure for handling tabular data.

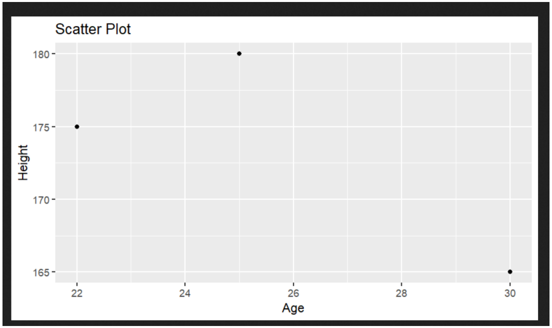
Data Visualization - ggplot2

```
# Install and load ggplot2 package
install.packages("ggplot2")
library(ggplot2)

# Create a scatter plot
ggplot(data, aes(x = Age, y = Height)) +
  geom_point() +
  labs(title = "Scatter Plot", x = "Age",
  y = "Height")
```

- The `ggplot2` package is widely used for creating elegant and versatile visualizations in R.
- A basic example of a scatter plot using `ggplot2`.

ggplot Output



Functions in R

```
# Create a custom function
multiply_by_two <- function(x) {
  result <- x * 2
  return(result)
}

# Use the function
output <- multiply_by_two(7)
```

- Functions in R are defined using the `function` keyword.
- Functions can take arguments and return values.

Conditional Statements - if-else

```
# Example of if-else statement
age <- 30

if (age >= 18) {
  print("Adult")
} else {
  print("Minor")
}
```

- Conditional statements are used for decision-making in R.
- Here, an example of an if-else statement is provided.

Loops - for and while

```r
# Example of a for loop
for (i in 1:5) {
  print(paste("Iteration:", i))
}

# Example of a while loop
counter <- 1
while (counter <= 5) {
  print(paste("Iteration:", counter))
  counter <- counter + 1
}
```

- R supports both `for` and `while` loops for iterative tasks.
- Here, examples of both types of loops are provided.

Data Import and Export in R

```r
# Import data from a CSV file
csv_data <- read.csv("data.csv")

# Export data to a CSV file
write.csv(csv_data, "exported_data.csv")
```

- R provides functions like `read.csv` and `write.csv` for importing and exporting data.
- CSV (Comma-Separated Values) is a common format for data exchange.

Data Manipulation with tidyr

```
# Install and load tidyr package
install.packages("tidyr")
library(tidyr)

# Example of gathering data
wide_data <- data.frame(
  Name = c("John", "Mary", "Alice"),
  Exam1 = c(85, 92, 78),
  Exam2 = c(90, 88, 95)
)

long_data <- gather(wide_data, key = "Exam",
value = "Score", -Name)
```

Data Manipulation with tidyr

- The ‘tidyr‘ package is commonly used for data reshaping and tidying.
- The ‘gather‘ function is used here to convert wide-format data to long-format.

Data Cleaning with dplyr

```
# Example of data cleaning using dplyr
cleaned_data <- example %>%
  filter(!is.na(Height)) %>%
  mutate(Name = tolower(Name)) %>%
  select(Name, Age, Height)
```

- The ‘dplyr‘ package provides functions for data cleaning and manipulation.
- Here, the ‘filter‘, ‘mutate‘, and ‘select‘ functions are used for various data cleaning tasks.

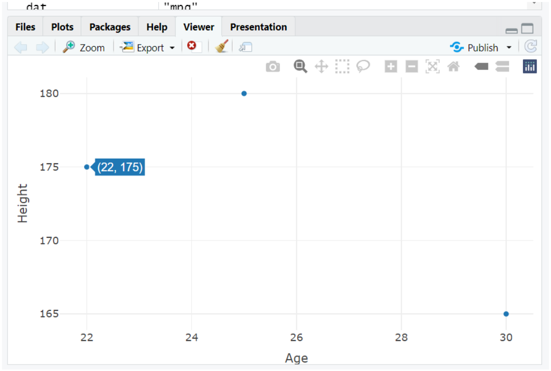
Interactive Visualizations with plotly

```r
# Install and load plotly package
install.packages("plotly")
library(plotly)

# Example of interactive plot
interactive_plot <- plot_ly(data, x = ~Age,
y = ~Height, type = "scatter", mode =
"markers")
```

- The ‘plotly‘ package enables the creation of interactive visualizations.
- Here, an example of an interactive scatter plot is provided.

plotly Output


dat
"mpg"
Files
Plots
Packages
Help
Viewer
Presentation
Zoom
Export
Publish
180
175
170
165
(22, 175)
Height
22
24
26
28
30
Age