Working with Different Files in R Programming

Student's Name

Institutional Affiliation

Course Code

Instructor

Date

Working with Different Files in R Programming

Introduction:

In data science and statistical computing, basic activities include data manipulation and analysis. R studio is one of the most used programming tools among data scientists and researchers and is popular for its strong support of and ability to work with several file formats such as CSV, XLS (Excel), and JSON(JavaScript). Understanding how to handle various data types and common procedures performed on the data is fundamental and will be described in this article.

R studio offers several packages that make working with different file types easier. For CSV (comma separated values) and XSL (Excel files), these include the readr and readxl packages, and for JSON files, there is the JSONlite package.

1. Working with CSV files

The read.csv() method in R programming can read CSV files.

```
R
# Loading the readr package.
library(readr)

# Read a CSV file
data123 <- read_csv("data.csv")
```

```
# Writing data to a CSV file.
write_csv(data123, "output123.csv")
```

2. Working with Excel Files:

Excel files are frequently used in research and business settings. The readxl package can be used to read Excel files.

```
# Load the readxl package
library(readxl)
# Read an Excel file
data <- read_excel("data.xlsx", sheet = "Sheet1")
```

The write_xlsx() function can be utilized to write data to an Excel file:

```
# Write data to an Excel file
write_xlsx(data, "output.xlsx")
```

3. Working with JSON Files:

JSON (JavaScript Object Notation) files are used for structured data. The jsonlite package in R makes it easier to read and write JSON files.

Reading JSON Files:

The fromJSON() function can read JSON files.

```R
# Load the jsonlite package
library(jsonlite)

# Read a JSON file
data <- fromJSON("data.json")

# Write data to a JSON file
toJSON(data, "output.json")
```

Data Manipulation:

Reading and writing JSON files is made simpler in R with the jsonlite package.

Reading JSON Files:

JSON files can be read using the fromJSON() function.

Using R to Filter Data:

In R, filtering data involves choosing a subset of your data according to predetermined standards. In data analysis, it is a basic process and is followed by other processes such as visualization and aggregation after successfully loading data from those different data formats.

Filtering Criteria: You describe which rows or observations you wish to maintain in your dataset by defining the criteria for filtering data. Usually, these requirements are based on one or more variables or columns.

Functions for Filtering Data: R has several functions and packages for filtering data, including which(), subset(), and filter() from the dplyr package.

```
R
Copy code
# Filtering data
filtered_data <- subset(data, ColumnName > 10)

# Aggregation
library(dplyr)
summarized_data <- data %>%
  group_by(ColumnName) %>%
  summarize(Mean = mean(AnotherColumn))

# Visualization
library(ggplot2)
ggplot(data123, aes(x = Column123, y = Column213)) +
  geom_point()
```

R offers some packages for sophisticated data analysis, making it an adaptable tool for working with a variety of data formats. These skills will give you confidence when handling and analyzing data from various sources in R.

References

Programming fundamentals: Working with .csv Files in base R. (2017a). https://doi.org/10.4135/9781526472854

Programming fundamentals: Working with .csv Files in base R. (2017b). https://doi.org/10.4135/9781526472854