UNIT II
Read Data from Various Sources

By

C.Sivamurugan (AP/CSE)

Read Data from Various Sources

- CSV Files
- Excel Files
- JSON Files
- SQL Databases
- Web APIs
- Web Scraping
- Big Data Sources
- Streaming Data

1. CSV Files

- **CSV Files :** CSV (Comma-Separated Values) files store tabular data in a plain-text format, where each line represents a row of data, and values within each row are separated by commas (or other specified delimiters).
- CSV files are widely used for data exchange between different software applications and can be easily opened and edited in text editors or spreadsheet software.
- **Library:** readr
- **Function :** read.csv()
- **Code:**

```
install.packages("readr")          # Install library file
library(readr)                     # Load Library file
data <- read_csv('data.csv')       # Read CSV Files
```

2. EXCEL Files

- **Excel Files :** Excel files store data in a structured manner using worksheets, rows, columns, and cells.
- The data within each worksheet is organized into rows and columns, with individual values stored in cells at the intersections of rows and columns.
- **Library:** readx1
- **Function :** read_excel()
- **Code:**

```
install.packages("readxl")                    # Install library file
library(readxl)                               # Load Library file
data <- read_excel('path/to/your/file.xlsx', sheet = 'Sheet1')
                                              # Read Excel Files
```

3. JSON Files

- **JSON Files :** JSON (JavaScript Object Notation)files store data in a text-based format using key-value pairs, arrays, and nested structures.
- **Library:** jsonlite
- **Function :** fromJSON()
- **Code:**

```
install.packages("jsonlite"")  # Install library file
library(jsonlite)                    # Load Library file
data <- fromJSON(file = 'path/to/your/file.json')
                                          # Read CSV Files
```

4. SQL Databases

- **SQL Databases:** SQL databases use tables to store data, where each table represents a collection of related records
- **Library:** DBI
- **Function :** dbGetQuery()
- **Code:**

```
install.packages("DBI")          # Install library file
library(DBI)                     # Load Library file
query <- "SELECT * FROM your_table"
data <- dbGetQuery(con, query)
                                 # Read SQL Database Files
```

5. Web APIs

- **Web APIs:** Data in a web API (Application Programming Interface) is typically stored on a server and exposed to clients over the internet for communication and interaction.
- The data in a web API is stored and managed on the server, and clients can make requests to access or manipulate that data.
- **Library:** httr
- **Function : GET() & content()**

  GET() function to send a GET request to the API's URL and retrieve the response.

  content() function to extract the content of the response

- **Code:**

```
install.packages(“httr")      # Install library file
library(httr)                 # Load Library file
url <- "https://google.com/data"
response <- GET(url)
api_data <- content(response, as = "parsed")
                              # Read from Web API
```

6. Web Scraping

- **Web Scraping:** Web scraping involves extracting information from web pages, and the data you scrape can be saved in various ways based on your needs.
- CSV, JSON, Excel are some common ways data is stored after web scraping
- **Library:** rvest
- **Function :read_html() , html_text(), etc**

  read_html(): Read HTML content from a web page.

  html_text(): Extract text content from HTML elements

Code:

```
install.packages("rvest")      # Install library file
library(rvest)                 # Load Library file
url <- "https://example.com"  # Load a web page
page <- read_html(url)
# Extract specific elements using CSS selectors
titles <- page %>% html_nodes("h2") %>% html_text()          # Read from Web Scraping
```

7. Streaming Data

- **Streaming Data:** Big data sources store and manage massive volumes of data that exceed the capacity of traditional database systems
- Here's how data is typically stored and managed in streaming data scenarios:
  - **Data Streams**
  - **Real-Time Analytics and Alerts**
- **Library: streamR**
- **Function : filterStream()**
- **Code:**

```
install.packages("streamR")
library(streamR)
setup_twitter_oauth("API Key", "API Secret", "Access Token",      "Access Token Secret")
# Collect streaming data
tweets <- filterStream(file.name = "tweets.json", track = c("keyword1",          "keyword2"))
                # Read from Streaming Data
```

8. Big Data Sources

- **Big Data Sources:** Streaming data is a continuous flow of real-time data that is generated, collected, and processed as it becomes available
- Here's how data is typically stored in various big data sources:
  - **Hadoop Distributed File System (HDFS)**
  - **NoSQL Databases (e.g., MongoDB, Cassandra)**
  - **Columnar Databases (e.g., Apache Parquet, Apache ORC)**
- **Library:**
  - **sparklyr for Apache Spark**
  - **rhipe for Hadoop**
  - **rmongodb for MongoDB**
- **Function : spark_read_csv()** Read CSV files**, etc**
- **Code:**

```
install.packages(“sparklyr")      # Install library file
library(sparklyr)                 # Load Library file
sc <- spark_connect(master = "local")
df <- spark_read_csv(sc, "path/to/your/csv/file.csv")
                                  # Read from Big Data Sources
```