Reading a CSV File

Following is a simple example of **read.csv()** function to read a CSV file available in your current working directory −

data <- read.csv("input.csv")

data

the CSV File

By default the **read.csv()** function gives the output as a data frame. This can be easily checked as follows. Also we can check the number of columns and rows.

data <- read.csv("input.csv")

ncol(data)

nrow(data)

Get the maximum salary

# Get the max salary from data frame.

sal <- max(data$salary)

sal

When we execute the above code, it produces the following result −

[1] 843.25

Get the details of the person with max salary

We can fetch rows meeting specific filter criteria similar to a SQL where clause.

# Get the max salary from data frame.

sal <- max(data$salary)

# Get the person detail having max salary.

retval <- subset(data, salary == max(salary))

retval

Get all the people working in IT department

retval <- subset( data, dept == "IT")

retval

Get the persons in IT department whose salary is greater than 600

info <- subset(data, salary > 600 & dept == "IT")

info

Get the people who joined on or after 2014

retval <- subset(data, as.Date(start\_date) > as.Date("2014-01-01"))

retval

Writing into a CSV File

R can create csv file form existing data frame. The **write.csv()** function is used to create the csv file. This file gets created in the working directory.

# Create a data frame.

data <- read.csv("input.csv")

retval <- subset(data, as.Date(start\_date) > as.Date("2014-01-01"))

# Write filtered data into a new file.

write.csv(retval,"output.csv")

newdata <- read.csv("output.csv")

newdata

Here the column X comes from the data set newper. This can be dropped using additional parameters while writing the file.

# Create a data frame.

data <- read.csv("input.csv")

retval <- subset(data, as.Date(start\_date) > as.Date("2014-01-01"))

# Write filtered data into a new file.

write.csv(retval,"output.csv", row.names = FALSE)

newdata <- read.csv("output.csv")

newdata