

INSTITUTION: KCA University

NAME: Raphael Kang'eri

UNIT NAME: Programming for Data Science

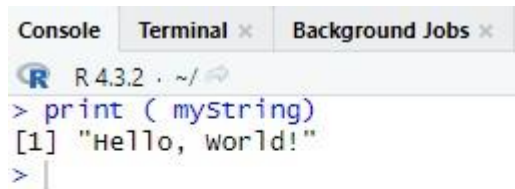
UNIT CODE: BSD 320

INTRODUCTION TO R FOR DATA SCIENCE

1. # My first program in R Programming

```
myString <- "Hello, World!"
```

```
print ( myString)
```

A screenshot of the R console interface. At the top, there are three tabs: 'Console', 'Terminal', and 'Background Jobs'. The 'Console' tab is active. Below the tabs, the R logo and version 'R 4.3.2' are visible. The console shows the following commands and output: a prompt '>' followed by 'print ( myString)', then the output '[1] "Hello, world!"', and finally another prompt '>' with a cursor.

```
R 4.3.2 . ~/
> print ( myString)
[1] "Hello, world!"
> |
```

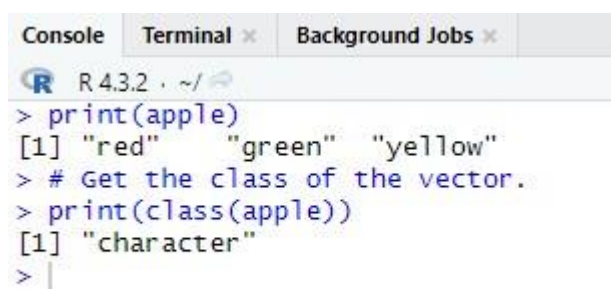
2. # Create a vector.

```
apple <- c('red','green',"yellow")
```

```
print(apple)
```

```
# Get the class of the vector.
```

```
print(class(apple))
```

A screenshot of the R console interface. At the top, there are three tabs: 'Console', 'Terminal', and 'Background Jobs'. The 'Console' tab is active. Below the tabs, the R logo and version 'R 4.3.2' are visible. The console shows the following commands and output: a prompt '>' followed by 'print(apple)', then the output '[1] "red" "green" "yellow"', then a comment '# Get the class of the vector.', then a prompt '>' followed by 'print(class(apple))', then the output '[1] "character"', and finally another prompt '>' with a cursor.

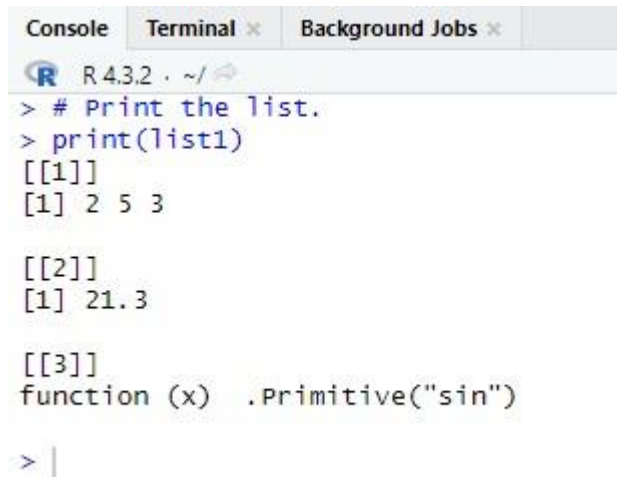
```
R 4.3.2 . ~/
> print(apple)
[1] "red" "green" "yellow"
> # Get the class of the vector.
> print(class(apple))
[1] "character"
> |
```

3. # Create a list.

```
list1 <- list(c(2,5,3),21.3,sin)
```

# Print the list.

```
print(list1)
```



The screenshot shows the R console interface with three tabs: 'Console', 'Terminal', and 'Background Jobs'. The 'Console' tab is active, displaying the R prompt and the execution of the code from the previous block. The output shows the list 'list1' with three elements: a vector of integers [2, 5, 3], a numeric value 21.3, and a function object for 'sin'.

```
R 4.3.2 . ~/
> # Print the list.
> print(list1)
[[1]]
[1] 2 5 3

[[2]]
[1] 21.3

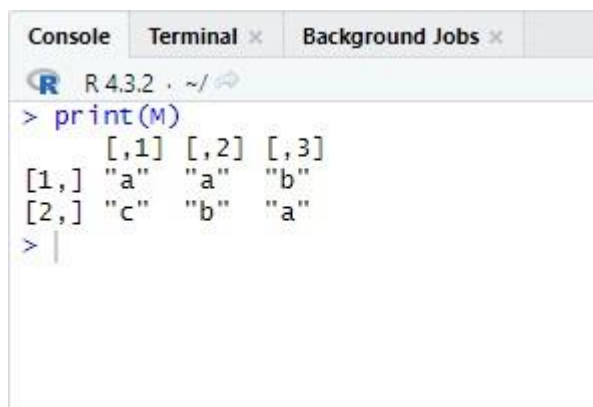
[[3]]
function (x) .Primitive("sin")

> |
```

4. # Create a matrix.

```
M = matrix( c('a','a','b','c','b','a'), nrow = 2, ncol = 3, byrow = TRUE)
```

```
print(M)
```



The screenshot shows the R console interface with the same three tabs. The 'Console' tab is active, displaying the R prompt and the execution of the code from the previous block. The output shows the matrix 'M' with 2 rows and 3 columns, filled by row with the characters 'a', 'a', 'b', 'c', 'b', 'a'.

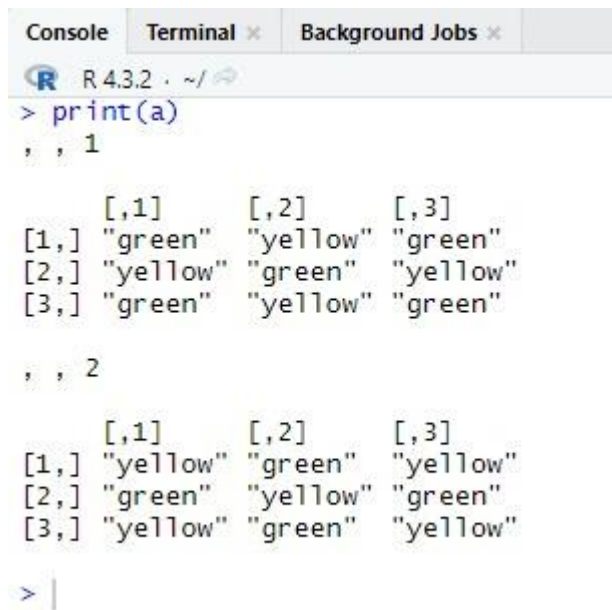
```
R 4.3.2 . ~/
> print(M)
      [,1] [,2] [,3]
[1,] "a"  "a"  "b"
[2,] "c"  "b"  "a"

> |
```

5. # Create an array.

```
a <- array(c('green','yellow'),dim = c(3,3,2))
```

```
print(a)
```



```
R 4.3.2 . ~/   
> print(a)  
, , 1  
  
      [,1]      [,2]      [,3]  
[1,] "green" "yellow" "green"  
[2,] "yellow" "green"  "yellow"  
[3,] "green"  "yellow" "green"  
  
, , 2  
  
      [,1]      [,2]      [,3]  
[1,] "yellow" "green"  "yellow"  
[2,] "green"  "yellow" "green"  
[3,] "yellow" "green"  "yellow"  
  
> |
```

6. # Create a vector.

```
apple_colors <- c('green','green','yellow','red','red','red','green')
```

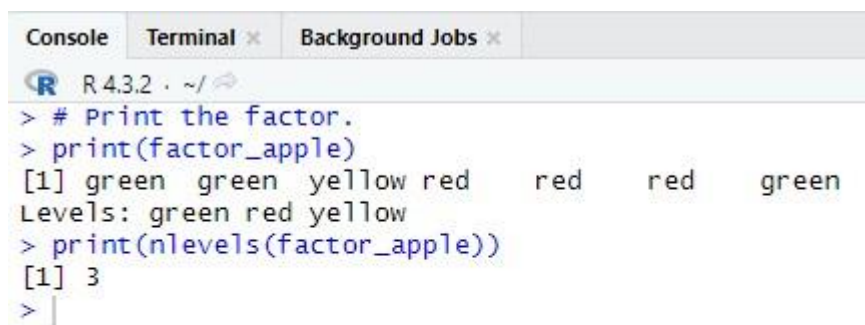
# Create a factor object.

```
factor_apple <- factor(apple_colors)
```


# Print the factor.

```
print(factor_apple)
```

```
print(nlevels(factor_apple))
```

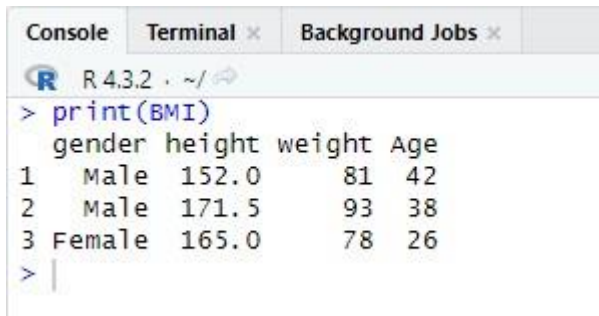


The screenshot shows an R console window with the following content:

```
R 4.3.2 . ~/ 
> # Print the factor.
> print(factor_apple)
[1] green green yellow red red red green
Levels: green red yellow
> print(nlevels(factor_apple))
[1] 3
> |
```

7. # Create the data frame.

```
BMI <- data.frame(  
  gender = c("Male", "Male", "Female"),  
  height = c(152, 171.5, 165),  
  weight = c(81, 93, 78),  
  Age = c(42, 38, 26)  
)  
print(BMI)
```



The screenshot shows an R console window with the following content:

```
R 4.3.2 ~/  
> print(BMI)  
  gender height weight Age  
1  Male  152.0     81  42  
2  Male  171.5     93  38  
3 Female  165.0     78  26  
> |
```

	gender	height	weight	Age
1	Male	152.0	81	42
2	Male	171.5	93	38
3	Female	165.0	78	26

8. # Assignment using equal operator.

```
var.1 = c(0,1,2,3)
```

# Assignment using leftward operator.

```
var.2 <- c("learn","R")
```

# Assignment using rightward operator.

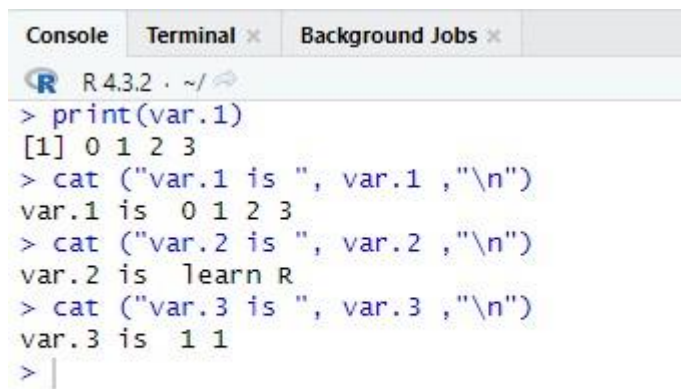
```
c(TRUE,1) -> var.3
```

```
print(var.1)
```

```
cat ("var.1 is ", var.1 ,"\n")
```

```
cat ("var.2 is ", var.2 ,"\n")
```

```
cat ("var.3 is ", var.3 ,"\n")
```



The screenshot shows an R console window with the following content:

```
R 4.3.2 . ~/
> print(var.1)
[1] 0 1 2 3
> cat ("var.1 is ", var.1 ,"\n")
var.1 is  0 1 2 3
> cat ("var.2 is ", var.2 ,"\n")
var.2 is  learn R
> cat ("var.3 is ", var.3 ,"\n")
var.3 is  1 1
> |
```

9. `var_x <- "Hello"`

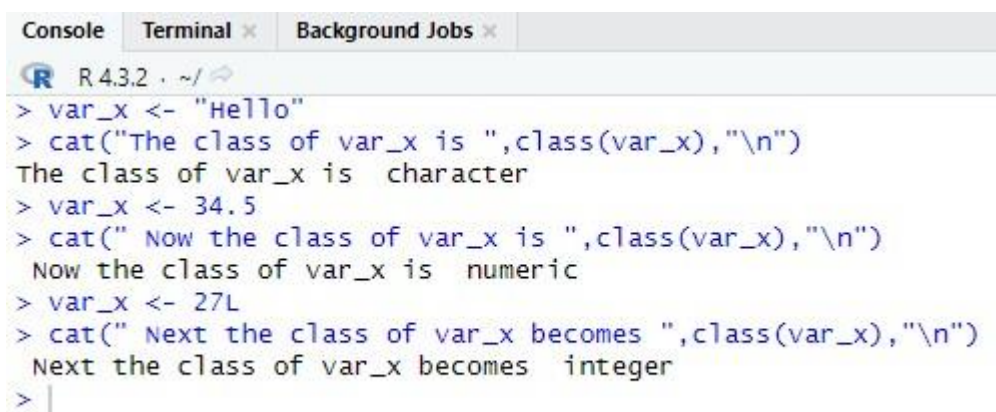
`cat("The class of var_x is ",class(var_x),"\n")`

`var_x <- 34.5`

`cat(" Now the class of var_x is ",class(var_x),"\n")`

`var_x <- 27L`

`cat(" Next the class of var_x becomes ",class(var_x),"\n")`

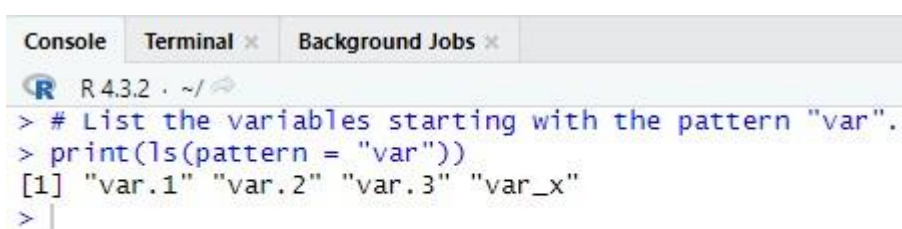


The screenshot shows an R console window with the following text:

```
R 4.3.2 . ~/
> var_x <- "Hello"
> cat("The class of var_x is ",class(var_x),"\n")
The class of var_x is  character
> var_x <- 34.5
> cat(" Now the class of var_x is ",class(var_x),"\n")
Now the class of var_x is  numeric
> var_x <- 27L
> cat(" Next the class of var_x becomes ",class(var_x),"\n")
Next the class of var_x becomes  integer
> |
```

10. # List the variables starting with the pattern "var".

`print(ls(pattern = "var"))`



The screenshot shows an R console window with the following text:

```
R 4.3.2 . ~/
> # List the variables starting with the pattern "var".
> print(ls(pattern = "var"))
[1] "var.1" "var.2" "var.3" "var_x"
> |
```



11. `print(ls(all.name = TRUE))`

```
Console Terminal x Background Jobs x
R 4.3.2 . ~/
> print(ls(all.name = TRUE))
[1] ".Random.seed" "a" "apple" "apple_colors"
[5] "BMI" "data" "factor_apple" "input"
[9] "list1" "M" "mystring" "v"
[13] "var.1" "var.2" "var.3" "var_x"
> |
```

1.

INSTITUTION: KCA University

NAME: Raphael Kang'eri

UNIT NAME: Programming for Data Science

UNIT CODE: BSD 320

R-CSV

1. # Get and print current working directory.

```
print(getwd())
```

```
# Set current working directory.
```

```
setwd("/web/com")
```

```
# Get and print current working directory.
```

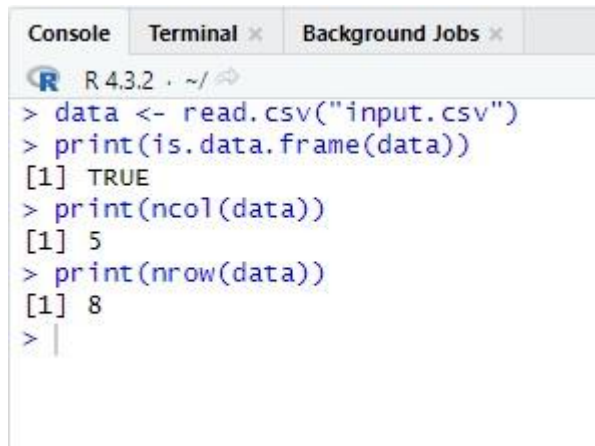
```
print(getwd())
```

```
Console Terminal x Background Jobs x
R 4.3.2 . ~/
> # Get and print current working directory.
> print(getwd())
[1] "C:/Users/hp/Documents"
> # Set current working directory.
> setwd("/web/com")
Error in setwd("/web/com") : cannot change working directory
> |
```

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

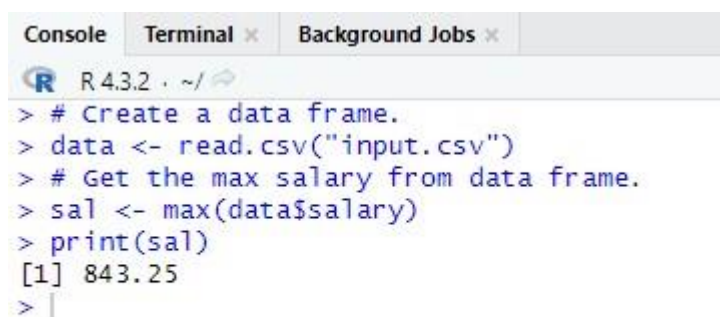
```
Console Terminal x Background Jobs x
R 4.3.2 . ~/
> print(data)
  id  name salary start_date dept
1  1  Rick  623.30 2012-01-01   IT
2  2   Dan  515.20 2013-09-23 Operations
3  3 Michelle 611.00 2014-11-15   IT
4  4   Ryan  729.00 2014-05-11   HR
5  5   Gary  843.25 2015-03-27 Finance
6  6   Nina  578.00 2013-05-21   IT
7  7  Simon  632.80 2013-07-30 Operations
8  8   Guru  722.50 2014-06-17 Finance
> |
```

```
3. data <- read.csv("input.csv")
   print(is.data.frame(data))
   print(ncol(data))
   print(nrow(data))
```

A screenshot of the R console interface showing the execution of R code. The console has tabs for 'Console', 'Terminal', and 'Background Jobs'. The R version is 4.3.2. The code entered is: > data <- read.csv("input.csv"), > print(is.data.frame(data)), > print(ncol(data)), > print(nrow(data)). The output is: [1] TRUE, [1] 5, [1] 8. The prompt > is followed by a vertical bar |.

```
R 4.3.2 . ~/
> data <- read.csv("input.csv")
> print(is.data.frame(data))
[1] TRUE
> print(ncol(data))
[1] 5
> print(nrow(data))
[1] 8
> |
```

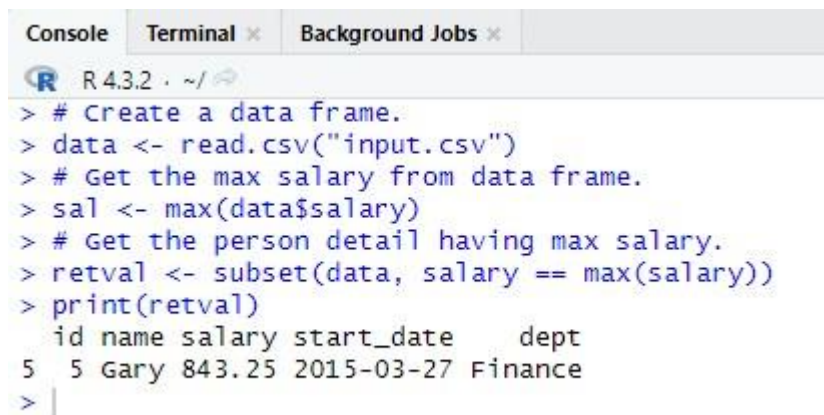
```
4. # Create a data frame.
   data <- read.csv("input.csv")
   # Get the max salary from data frame.
   sal <- max(data$salary)
   print(sal)
```

A screenshot of the R console interface showing the execution of R code. The console has tabs for 'Console', 'Terminal', and 'Background Jobs'. The R version is 4.3.2. The code entered is: > # Create a data frame., > data <- read.csv("input.csv"), > # Get the max salary from data frame., > sal <- max(data\$salary), > print(sal). The output is: [1] 843.25. The prompt > is followed by a vertical bar |.

```
R 4.3.2 . ~/
> # Create a data frame.
> data <- read.csv("input.csv")
> # Get the max salary from data frame.
> sal <- max(data$salary)
> print(sal)
[1] 843.25
> |
```

5. # Create a data frame.

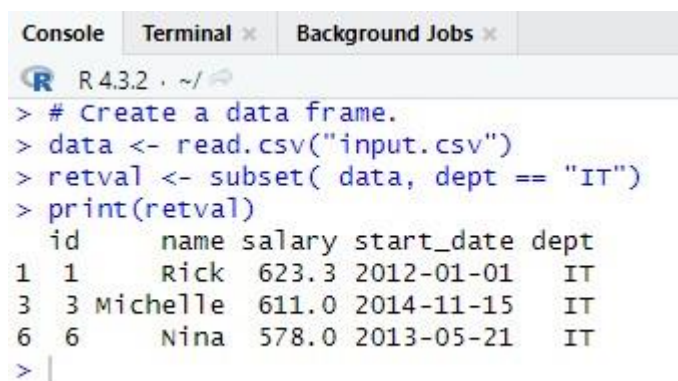
```
data <- read.csv("input.csv")  
# Get the max salary from data frame.  
sal <- max(data$salary)  
# Get the person detail having max salary.  
retval <- subset(data, salary == max(salary))  
print(retval)
```

A screenshot of the R console showing the execution of R code for step 5. The console has tabs for 'Console', 'Terminal', and 'Background Jobs'. The code executed is: > # Create a data frame.  
> data <- read.csv("input.csv")  
> # Get the max salary from data frame.  
> sal <- max(data\$salary)  
> # Get the person detail having max salary.  
> retval <- subset(data, salary == max(salary))  
> print(retval). The output is a data frame with 5 rows and 5 columns: id, name, salary, start\_date, and dept. The first row is: 5, 5, Gary, 843.25, 2015-03-27, Finance.  

	id	name	salary	start_date	dept
5	5	Gary	843.25	2015-03-27	Finance

6. # Create a data frame.

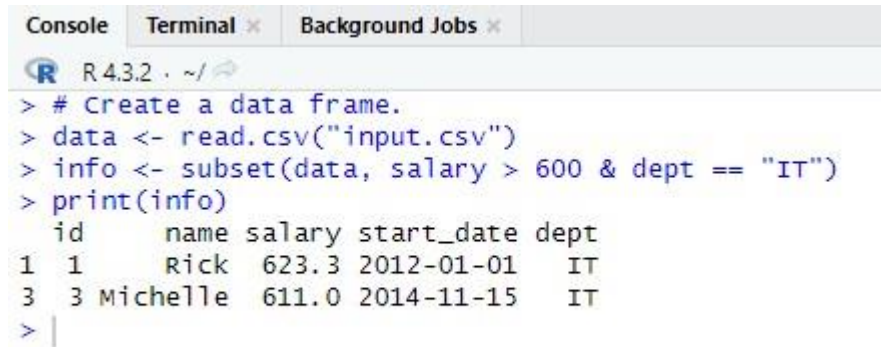
```
data <- read.csv("input.csv")  
retval <- subset( data, dept == "IT")  
print(retval)
```

A screenshot of the R console showing the execution of R code for step 6. The console has tabs for 'Console', 'Terminal', and 'Background Jobs'. The code executed is: > # Create a data frame.  
> data <- read.csv("input.csv")  
> retval <- subset( data, dept == "IT")  
> print(retval). The output is a data frame with 3 rows and 5 columns: id, name, salary, start\_date, and dept. The rows are: 1, 1, Rick, 623.3, 2012-01-01, IT; 3, 3, Michelle, 611.0, 2014-11-15, IT; 6, 6, Nina, 578.0, 2013-05-21, IT.  

	id	name	salary	start_date	dept
1	1	Rick	623.3	2012-01-01	IT
3	3	Michelle	611.0	2014-11-15	IT
6	6	Nina	578.0	2013-05-21	IT

7. # Create a data frame.

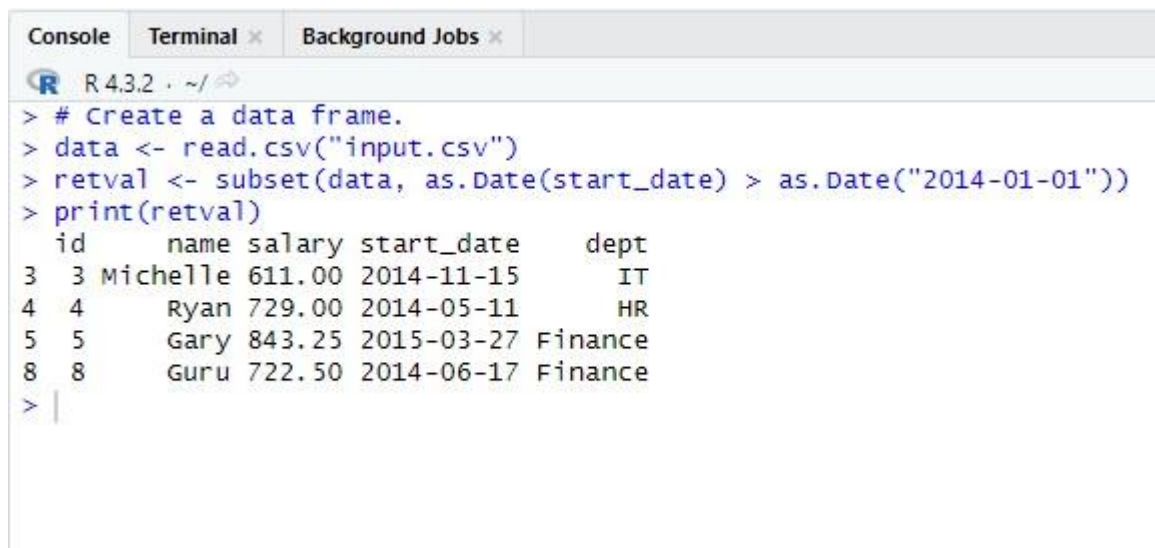
```
data <- read.csv("input.csv")  
info <- subset(data, salary > 600 & dept == "IT")  
print(info)
```

A screenshot of an R console window with tabs for 'Console', 'Terminal', and 'Background Jobs'. The console shows the execution of R code to create a data frame and filter it. The output is a data frame with 3 rows and 5 columns: id, name, salary, start\_date, and dept.

```
R 4.3.2 . ~/ >  
> # Create a data frame.  
> data <- read.csv("input.csv")  
> info <- subset(data, salary > 600 & dept == "IT")  
> print(info)  
  id    name salary start_date dept  
1  1    Rick  623.3 2012-01-01   IT  
3  3 Michelle  611.0 2014-11-15   IT  
> |
```

8. # Create a data frame.

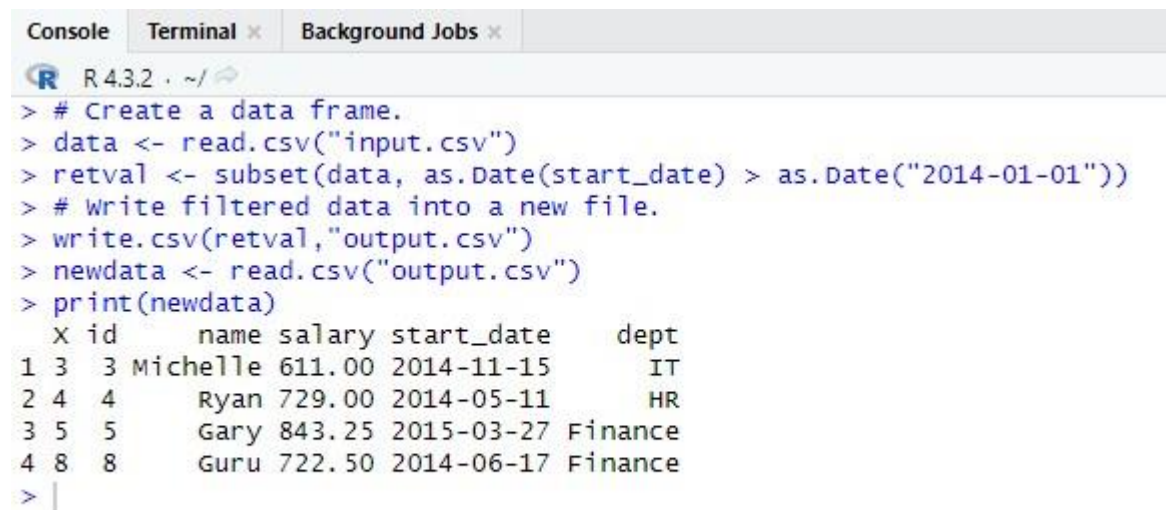
```
data <- read.csv("input.csv")  
retval <- subset(data, as.Date(start_date) > as.Date("2014-01-01"))  
print(retval)
```

A screenshot of an R console window with tabs for 'Console', 'Terminal', and 'Background Jobs'. The console shows the execution of R code to create a data frame and filter it by start\_date. The output is a data frame with 4 rows and 5 columns: id, name, salary, start\_date, and dept.

```
R 4.3.2 . ~/ >  
> # Create a data frame.  
> data <- read.csv("input.csv")  
> retval <- subset(data, as.Date(start_date) > as.Date("2014-01-01"))  
> print(retval)  
  id    name salary start_date    dept  
3  3 Michelle  611.00 2014-11-15     IT  
4  4    Ryan  729.00 2014-05-11     HR  
5  5    Gary  843.25 2015-03-27 Finance  
8  8    Guru  722.50 2014-06-17 Finance  
> |
```

9. # 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")
print(newdata)
```



The screenshot shows an R console window with the following content:

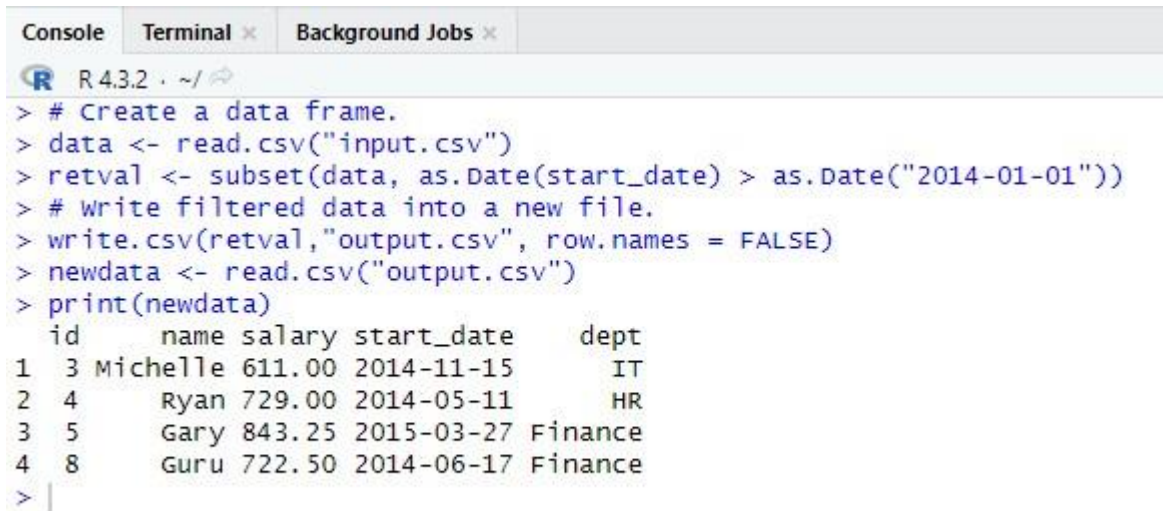
```
R 4.3.2 . ~/
> # Create a data frame.
> data <- read.csv("input.csv")
> retval <- subset(data, as.Date(start_date) > as.Date("2014-01-01"))
> # Write filtered data into a new file.
> write.csv(retval,"output.csv")
> newdata <- read.csv("output.csv")
> print(newdata)
```

	x	id	name	salary	start_date	dept
1	3	3	Michelle	611.00	2014-11-15	IT
2	4	4	Ryan	729.00	2014-05-11	HR
3	5	5	Gary	843.25	2015-03-27	Finance
4	8	8	Guru	722.50	2014-06-17	Finance

```
> |
```

10.# 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")
print(newdata)
```



The screenshot shows an R console window with the following content:

```
R 4.3.2 . ~/
> # 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")
> print(newdata)
  id name salary start_date dept
1  3 Michelle 611.00 2014-11-15  IT
2  4   Ryan 729.00 2014-05-11  HR
3  5   Gary 843.25 2015-03-27 Finance
4  8   Guru 722.50 2014-06-17 Finance
> |
```



INSTITUTION: KCA University

NAME: Raphael Kang'eri

UNIT NAME: Programming for Data Science

UNIT CODE: BSD 320

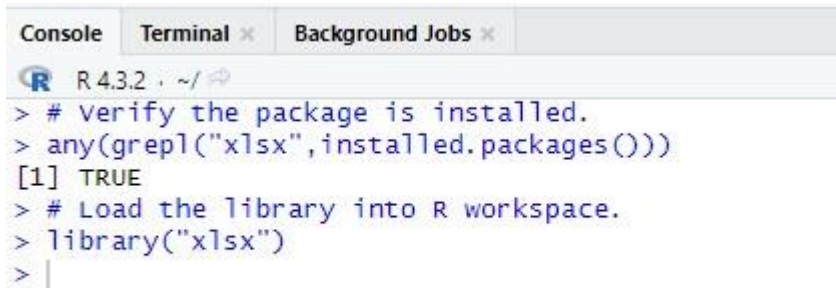
R-EXCEL

1. # Verify the package is installed.

```
any(grepl("xlsx",installed.packages()))
```

```
# Load the library into R workspace.
```

```
library("xlsx")
```

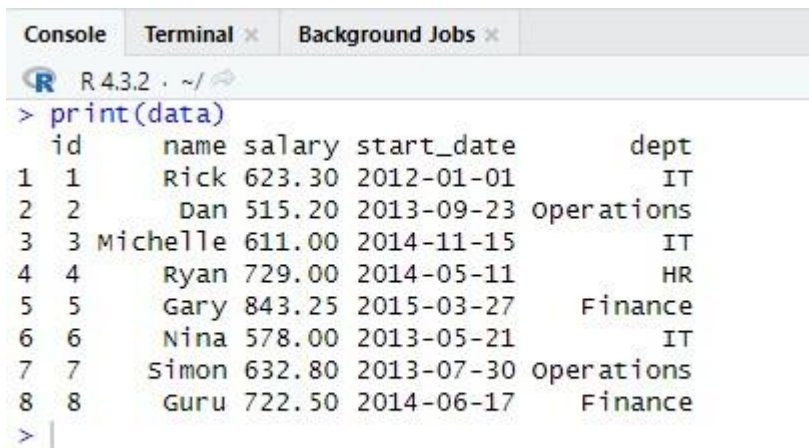


```
Console Terminal x Background Jobs x
R 4.3.2 . ~/
> # verify the package is installed.
> any(grepl("xlsx",installed.packages()))
[1] TRUE
> # Load the library into R workspace.
> library("xlsx")
> |
```

2. # Read the first worksheet in the file input.xlsx.

```
data <- read.xlsx("input.xlsx", sheetIndex = 1)
```

```
print(data)
```



```
Console Terminal x Background Jobs x
R 4.3.2 . ~/
> print(data)
  id  name salary start_date dept
1  1  Rick  623.30 2012-01-01   IT
2  2   Dan  515.20 2013-09-23 operations
3  3 Michelle 611.00 2014-11-15   IT
4  4   Ryan  729.00 2014-05-11   HR
5  5   Gary  843.25 2015-03-27 Finance
6  6   Nina  578.00 2013-05-21   IT
7  7  Simon  632.80 2013-07-30 operations
8  8   Guru  722.50 2014-06-17 Finance
> |
```