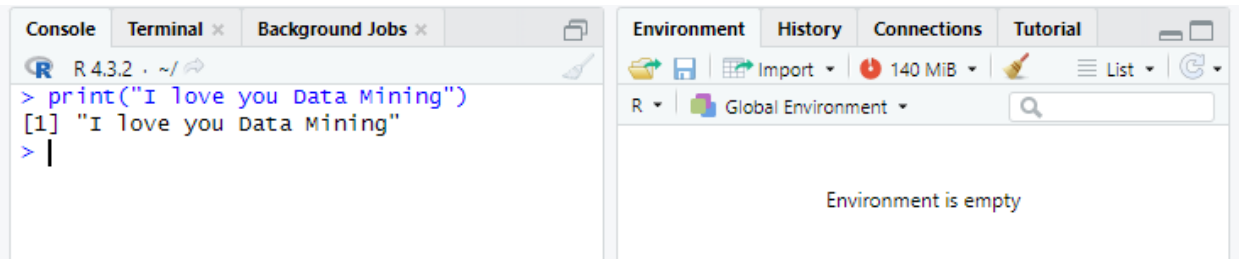


Name: Greendee Roper B. Panogalon      BSIT 3B

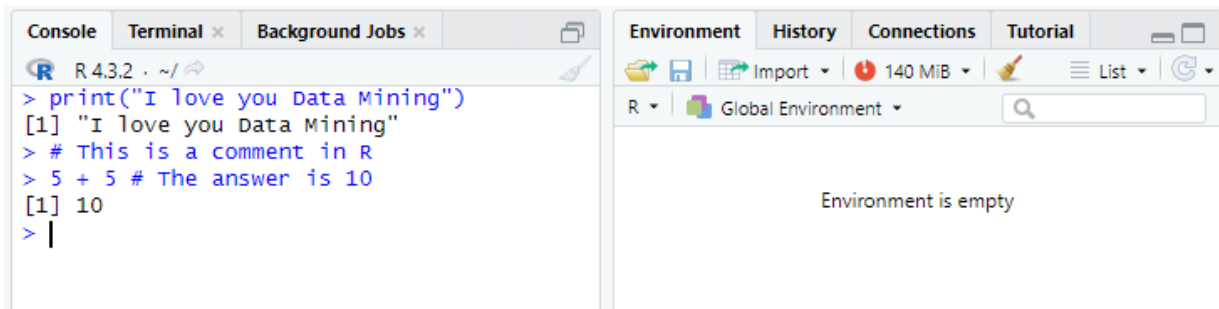
## R Language Exercises

1. In R, display the message "I love Data Mining".



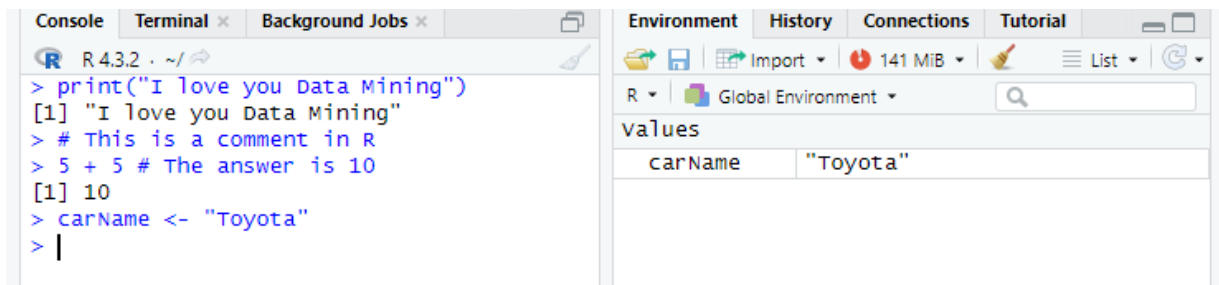
The screenshot shows the R Studio interface. The Console pane on the left contains the following text: `> print("I love you Data Mining")`, `[1] "I love you Data Mining"`, and a prompt `> |`. The Environment pane on the right shows "Global Environment" and the message "Environment is empty".

2. Comments in R are written with a special character. Give an example.



The screenshot shows the R Studio interface. The Console pane on the left contains the following text: `> print("I love you Data Mining")`, `[1] "I love you Data Mining"`, `> # This is a comment in R`, `> 5 + 5 # The answer is 10`, `[1] 10`, and a prompt `> |`. The Environment pane on the right shows "Global Environment" and the message "Environment is empty".

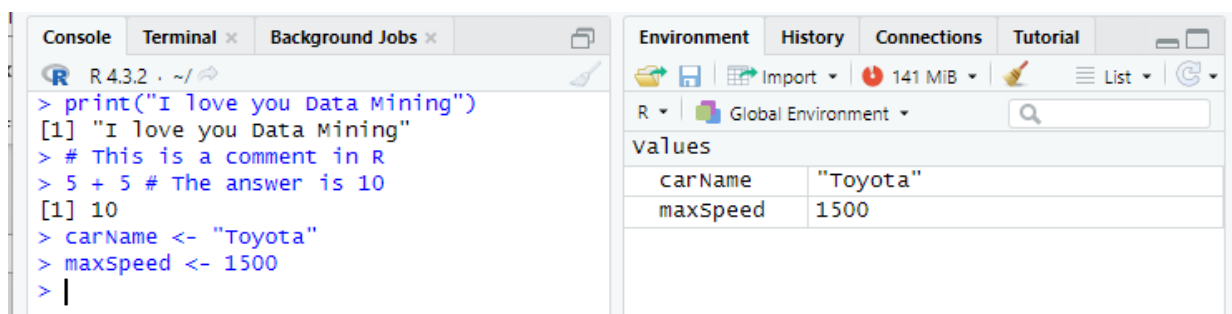
3. Create a variable named carName and assign the value Toyota to it.



The screenshot shows the R Studio interface. The Console pane on the left contains the following text: `> print("I love you Data Mining")`, `[1] "I love you Data Mining"`, `> # This is a comment in R`, `> 5 + 5 # The answer is 10`, `[1] 10`, `> carName <- "Toyota"`, and a prompt `> |`. The Environment pane on the right shows "Global Environment" and a table of values.

values	
carName	"Toyota"

4. Create a variable named maxSpeed and assign the value 1500 to it.



The screenshot shows the R Studio interface. The Console pane on the left contains the following text: `> print("I love you Data Mining")`, `[1] "I love you Data Mining"`, `> # This is a comment in R`, `> 5 + 5 # The answer is 10`, `[1] 10`, `> carName <- "Toyota"`, `> maxSpeed <- 1500`, and a prompt `> |`. The Environment pane on the right shows "Global Environment" and a table of values.

values	
carName	"Toyota"
maxSpeed	1500

5. List all data types used in R and give examples.

Numeric:                numVar <- 5.5

Integer:                intVar <- 97L

Complex:                complexVar <- 5i + 3

Character/String:       charVar <- "This is Character Data Type"

Logical/Boolean:       logVar <- TRUE

Factor:                facVar <- factor(c("Greendee", "Meriam", "Merde" ))

Vector:                vectorVar <- c(1, 2, 3, 4, 5)

List:                    listVar <- list("Asin", "Tuyo", "Suka")

Matrix:                matrixVar <- matrix(c(1,2,3,4,5,6), nrow = 3, ncol = 2)

Array:                arrayVar <- c(1:24)

Data Frame:            dataframeVar <- data.frame (  
                          `Sari-Sari Store` = c("KangKong Chips", "Cornetto", "Pepsi"),  
                          Price = c(130, 20, 35),  
                          Quantity = c(100, 50, 33)  
                          )

Date:                    dateVar <- as.Date("2024-02-06" )

DateTime:              dateTimeVar <- as.POSIXct("2024-02-06 15:50:30")

**Then, to execute that, call the variable or use `print(variable)` to display the contents of the variable in the R console.**

6. Use the correct function to find the lowest and highest number in a set. Give example for each function.

**Lowest Number:**

```
> numbers <- c(8, 6, 10, 12)
> lowestNumber <- min(numbers)
> print(lowestNumber)
[1] 6
```

highestNu...	12
i	10
lowestNum...	6

### Highest Number:

```
[1] 6
> numbers <- c(8, 6, 10, 12)
> highestNumber <- max(numbers)
> print(highestNumber)
[1] 12
```

highestNu...	12
i	10
lowestNum...	6
maxSpeed	1500

7. List 10 built in function in R and give examples.

print():	print("Hello world!!")
sum():	sum(c(2, 4, 6))
mean():	mean(c(3, 2, 3))
sort():	sort(c(2, 1, 4, 3, 5, 7, 6))
str():	str(c(1, 2, 3, 4, 5))
unique():	unique(c(1, 2, 2, 3, 4, 4, 4, 5, 6))
sqrt():	sqrt(25)
round():	round(3.14159, 2)
paste():	paste("Hello World!!")
seq():	seq(1, 10, by = 3)

8. Complete the code. Print "Yes" if a is equal to b, otherwise print "No".

```
a <- 70
b <- 80
if(a == b){
  print("Yes")
} else{
  print("No")
}
```

9. Complete the code. Print i as long as i is less than 10.

```
i <- 1
while(i < 10){
  print(i)
  i <- i + 1
}
```

10. Write a function that will return the product of the two numbers. Give example on how the function is called.

```
calculateProduct <- function(a, b){
  productResult <- a * b
  return(productResult)
}
```

# Example on how the function is called.

```
result <- calculateProduct(10, 10)
```

```
cat("The product is:", result, "\n")
```

```
> print(numbers)
[1] 6
> numbers <- c(8, 6, 10, 12)
> highestNumber <- max(numbers)
> print(highestNumber)
[1] 12
>
> calculateProduct <- function(a, b){
+   productResult <- a * b
+   return(productResult)
+ }
>
> # Example on how the function is called.
> result <- calculateProduct(10, 10)
> cat("The product is:", result, "\n")
The product is: 100
>
```

