Problem Set 1, Base R: Answer Key

Andie Creel

Problem 1: Basic Operations

Create two numeric variables with any values and perform basic arithmetic operations (addition, subtraction, multiplication, division) on them. Assign the result of a division to a new variable and print it.

```
# variable one
a <- 2

# variable two
b <- 3

a+b
```

```
## [1] 5
```

```
a-b
```

```
## [1] -1
```

```
a*b
```

```
## [1] 6
```

```
c <- a/b
c
```

```
## [1] 0.6666667
```

Problem 2: Working with Vectors

Create a numeric vector with at least 5 elements. Compute the sum (`sum()`) and mean (`mean()`) of the vector. Use a logical comparison (true/false) within a for-loop to identify which elements of the vector are greater than the mean. You may need to use the `print()` command.

```
# vector
myVec <- c(1,2,3,4,5)

# sum
sum(myVec)
```

```
## [1] 15
```

```
# mean
myMean <- mean(myVec)
myMean
```

```
## [1] 3
```

```
# logical comparison
for (i in 1:5) {
  print((myVec[i] > myMean))
}
```

```
## [1] FALSE
## [1] FALSE
## [1] FALSE
## [1] TRUE
## [1] TRUE
```

Problem 3: Basic Data Frame Manipulation

Create a data frame with at least 3 columns and 5 rows. The columns should include a mix of numeric, character, and logical data types. Print the entire data frame. Print only the second column of the data frame.

```
# Make DF
myDF <-     data.frame(
   gender = c("Male", "non-binary","Female", "Male", "non-binary"),
   male = c(T, F, F, T, F),
   height = c(152, 171.5, 165, 1222, 122)
)

# whole df
myDF
```

```
##       gender  male height
## 1       Male  TRUE  152.0
## 2 non-binary FALSE  171.5
## 3     Female FALSE  165.0
## 4       Male  TRUE 1222.0
## 5 non-binary FALSE  122.0
```

```
# second column
myDF$male
```

```
## [1]  TRUE FALSE FALSE  TRUE FALSE
```

```
myDF[,2]
```

```
## [1]  TRUE FALSE FALSE  TRUE FALSE
```

Problem 4: Loop and Function

Write a function that takes a numeric input and returns the square of the number. Use a for loop to apply this function to each element of a numeric vector that you create. Store the results in a new vector and print it using the `print()` command.

```
# function
mySquare <- function(x){
  y <- x^2
  return(y)
}

# test
mySquare(2)
```

```
## [1] 4
```

```
#initializing vector of NAs
newVec <- rep(NA, 5)

for (i in 1:5) {
  # using vector from before
  newVec[i] <- mySquare(myVec[i])

  # print
  print(newVec[i])
}
```

```
## [1] 1
## [1] 4
## [1] 9
## [1] 16
## [1] 25
```

Problem 5: Conditional Logic

Using the data frame created in Problem 3, write an if statement inside a loop to perform a conditional operation (e.g., if a number in a numeric column is even, replace it with half its value). Print the modified data frame.

Practicing Googling to figure out how to code if a number is even or not.

```
# loop
for (i in 1:5) {

  # check if even
  if((myDF$height[i] %% 2) == 0){

     # divide by 2
    myDF$height[i] <- myDF$height[i]/2
```

```
  }else{

    # do nothing
    myDF$height[i] <- myDF$height[i]
  }
}
```