

Homework 1

Week 1

1. Explain the R studio layout. What do the different panes do?
2. What do we use to assign output to an object? Why might it be better to use than an =?

Week 2

3. Explain the difference between a factor and character data type. When would we want to use either?
4. Run the following line of code:

```
hw_factor <- factor(c(2, 1, 5, 3), levels = c(1, 2, 3, 5))
```

Change this into a numeric data type. Explain the steps you took to get there.

```
# Change hw_factor to a character first  
as.numeric(as.character(hw_factor))
```

```
## [1] 2 1 5 3
```

5. I want to create a logical data object called `hw_logical`. I run the following code.

```
hw_logical <- "TRUE"  
  
is.logical(hw_logical)
```

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

Why does `is.logical` return FALSE? How can I fix `hw_logical` so it is a logical data type?

```
# hw_logical is a character data type  
class(hw_logical)
```

```
## [1] "character"
```

```
# Remove quotations to make it logical  
hw_logical <- TRUE  
class(hw_logical)
```

```
## [1] "logical"
```

6. Create a function called `test_function`. It should take two arguments, `x` and `y`. It should do the following:
 - Find the sum of input `x` and the number 6
 - Find the product of input `y` and the number 2
 - Store the output of those first two steps in a list

Run and show the output of `test_function`.

```
test_function <- function(x, y){  
  sum_x <- x + 6  
  
  product_y <- y * 2  
  
  output_list <- list(sum_x, product_y)
```

```

    return(output_list)
}

test_function(x = 1, y = 2)

## [[1]]
## [1] 7
##
## [[2]]
## [1] 4

```

7. Explain the difference between `install.packages` and `library`. When would you use each of these functions?

Week 3

```

# Load up the iris dataset
data("iris")

```

8. Calculate the average `Petal.Length` and `Petal.Width` by `Species`. Save the output to a `tibble` called `avg_petal`.

```

library(tidyverse)

avg_petal <- iris %>%
  group_by(Species) %>%
  summarize(Petal.Length = mean(Petal.Length),
            Petal.Width = mean(Petal.Width))

avg_petal

```

```

## # A tibble: 3 x 3
##   Species    Petal.Length Petal.Width
##   <fct>          <dbl>         <dbl>
## 1 setosa          1.46           0.246
## 2 versicolor      4.26           1.33
## 3 virginica       5.55           2.03

```

9. `avg_petal` is in wide format; change it to long format (you should have three columns: `Species`, a key column, and a value column). Filter the new `tibble` for all values greater than or equal to 2.

```

avg_petal %>%
  gather(Petal.Length, Petal.Width, key = "Metric", value = "Value") %>%
  filter(Value >= 2)

```

```

## # A tibble: 3 x 3
##   Species    Metric      Value
##   <fct>      <chr>     <dbl>
## 1 versicolor Petal.Length 4.26
## 2 virginica  Petal.Length 5.55
## 3 virginica  Petal.Width  2.03

```

10. Explain the difference between a `tibble` and a `data.frame`.