

Assignment: ASSIGNMENT 1

Name: Raghuwanshi, Prashant

Date: 2021-06-19

Create a numeric vector with the values of 3, 2, 1 using the `c()` function

Assign the value to a variable named `num_vector`

Print the vector

```
num_vector <- c(3,2,1) num_vector
```

Create a character vector with the values of “three”, “two”, “one” "using the `c()` function

Assign the value to a variable named `char_vector`

Print the vector

```
char_vector <- c("three", "two", "one") char_vector ## Create a vector called week1_sleep representing  
how many hours slept each night of the week ## Use the values 6.1, 8.8, 7.7, 6.4, 6.2, 6.9, 6.6 week1_sleep  
<- c(6.1, 8.8, 7.7, 6.4, 6.2, 6.9, 6.6)
```

Display the amount of sleep on Tuesday of week 1 by selecting the variable index

```
week1_sleep[3]
```

Create a vector called `week1_sleep_weekdays`

Assign the weekday values using indice slicing

```
week1_sleep_weekdays <- week1_sleep[2:6]
```

Add the total hours slept in week one using the `sum` function

Assign the value to variable `total_sleep_week1`

```
total_sleep_week1 <- sum(week1_sleep)
```

Create a vector called `week2_sleep` representing how many hours slept each night of the week

Use the values 7.1, 7.4, 7.9, 6.5, 8.1, 8.2, 8.9

```
week2_sleep <- c(7.1, 7.4, 7.9, 6.5, 8.1, 8.2, 8.9)
```

Add the total hours slept in week two using the `sum` function

Assign the value to variable `total_sleep_week2`

```
total_sleep_week2 <- sum(week2_sleep)
```

Determine if the total sleep in week 1 is less than week 2 by using the `<` operator

```
total_sleep_week1 < total_sleep_week2
```

Calculate the mean hours slept in week 1 using the mean() function

```
mean(week1_sleep)
```

Create a vector called days containing the days of the week.

Start with Sunday and end with Saturday

```
days <- c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")
```

Assign the names of each day to week1_sleep and week2_sleep using the names function and days vector

```
names(week1_sleep) <- days names(week2_sleep) <- days
```

Display the amount of sleep on Tuesday of week 1 by selecting the variable name

```
week1_sleep["Tuesday"]
```

Create vector called weekdays from the days vector

```
weekdays <- days[2:6]
```

Create vector called weekends containing Sunday and Saturday

```
weekends <- days[c(1,7)]
```

Calculate the mean about sleep on weekdays for each week

Assign the values to weekdays1_mean and weekdays2_mean

```
weekdays1_mean <- mean(week1_sleep[weekdays]) weekdays2_mean <- mean(week2_sleep[weekdays])
```

Using the weekdays1_mean and weekdays2_mean variables,

see if weekdays1_mean is greater than weekdays2_mean using the > operator

```
weekdays1_mean > weekdays2_mean
```

Determine how many days in week 1 had over 8 hours of sleep using the > operator

```
week1_sleep_over8hrs <- sum(week1_sleep > 8)
```

Create a matrix from the following three vectors

```
student01 <- c(100.0, 87.1) student02 <- c(77.2, 88.9) student03 <- c(66.3, 87.9)
```

```
students_combined <- matrix(c(student01, student02, student03), ncol=3)
```

```
grades <- matrix(students_combined, byrow = TRUE, nrow = 3) grades ## Add a new student row with  
rbind() student04 <- c(95.2, 94.1) grades <- rbind(grades, student04) grades ## Add a new assignment  
column with cbind() assignment04 <- c(92.1, 84.3, 75.1, 97.8) grades <- cbind(grades, assignment04)  
## Add the following names to columns and rows using rownames() and colnames() assignments <-  
c("Assignment 1", "Assignment 2", "Assignment 3") students <- c("Florinda Baird", "Jinny Foss", "Lou  
Purvis", "Nola Maloney")
```

```
rownames(grades) <- students colnames(grades) <- assignments grades ## Total points for each assignment
using colSums() colSums(grades)
```

Total points for each student using rowSums()

```
rowSums(grades) ## Matrix with 10% and add it to grades weighted_grades <- grades * 0.1 + grades
```

Create a factor of book genres using the genres_vector

Assign the factor vector to factor_genre_vector

```
genres_vector <- c("Fantasy", "Sci-Fi", "Sci-Fi", "Mystery", "Sci-Fi", "Fantasy") factor_genre_vector <-
as.factor(genres_vector)
```

Use the summary() function to print a summary of factor_genre_vector

```
summary(factor_genre_vector)
```

Create ordered factor of book recommendations using the recommendations_vector

no is the lowest and yes is the highest

```
recommendations_vector <- c("neutral", "no", "no", "neutral", "yes") factor_recommendations_vector <-
factor(recommendations_vector, ordered = TRUE, levels = c("yes", "neutral", "no"))
```

Use the summary() function to print a summary of factor_recommendations_vector

```
summary(factor_recommendations_vector)
```

Using the built-in mtcars dataset, view the first few rows using the head() function

```
head(mtcars, 6)
```

Using the built-in mtcars dataset, view the last few rows using the tail() function

```
tail(mtcars, 6)
```

Create a dataframe called characters_df using the following information from LOTR

```
name <- c("Aragon", "Bilbo", "Frodo", "Galadriel", "Sam", "Gandalf", "Legolas", "Sauron", "Gollum")
race <- c("Men", "Hobbit", "Hobbit", "Elf", "Hobbit", "Maia", "Elf", "Maia", "Hobbit") in_fellowship <-
c(TRUE, FALSE, TRUE, FALSE, TRUE, TRUE, TRUE, FALSE, FALSE) ring_bearer <- c(FALSE, TRUE,
TRUE, FALSE, TRUE, TRUE, FALSE, TRUE, TRUE) age <- c(88, 129, 51, 7000, 36, 2019, 2931, 7052, 589)
characters_df <- data.frame(name, race, in_fellowship, ring_bearer, age)
```

Sorting the characters_df by age using the order function and assign the result to the sorted_characters_df

```
sorted_characters_df <- characters_df[order(characters_df$age),] ## Use head() to output the first few
rows of sorted_characters_df head(sorted_characters_df, 6)
```

Select all of the ring bearers from the dataframe and assign it to ringbearers_df

```
ringbearers_df <- characters_df[characters_df$ring_bearer == TRUE,] ## Use head() to output the first  
few rows of ringbearers_df head(ringbearers_df, 3)
```