Week 2 - Assignment2

Reenie Christudass

2022-06-18

```
## 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)
print(paste(c("Numeric vector are: ", num_vector), collapse=" "))
```

```
## [1] "Numeric vector are:  3 2 1"
```

```
## 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")
print(paste(c("Character vector are: ", char_vector), collapse=" "))
```

```
## [1] "Character vector are:  three two one"
```

```
## 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)
print(paste(c("Hours slept each night of the week: ", week1_sleep), collapse=" "))
```

```
## [1] "Hours slept each night of the week:  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
weekdays1 <- c( "Sunday" ,"Monday", "Tuesday", "Wednesday", "Thrusday", "Friday", "Saturday")
print(paste("Display the amount of sleep on week 1:",weekdays1[3], week1_sleep[3]))
```

```
## [1] "Display the amount of sleep on week 1: Tuesday 7.7"
```

```
## Create a vector called `week1_sleep_weekdays`
## Assign the weekday values using indice slicing
identify_weekdays <- weekdays1[1:5]
week1_sleep_weekdays <- week1_sleep[1:5]
print(paste("Display the amount of sleep for week1:",identify_weekdays, week1_sleep_weekdays))
```

```
## [1] "Display the amount of sleep for week1: Sunday 6.1"
## [2] "Display the amount of sleep for week1: Monday 8.8"
## [3] "Display the amount of sleep for week1: Tuesday 7.7"
## [4] "Display the amount of sleep for week1: Wednesday 6.4"
## [5] "Display the amount of sleep for week1: Thrusday 6.2"
```

```r
## 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[1:7])
print(paste("Display the total amount of sleep for week1:",total_sleep_week1))
```

```
## [1] "Display the total amount of sleep for week1: 48.7"
```

```r
## 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
identify_weekdays2 <- c( "Sunday" ,"Monday", "Tuesday", "Wednesday", "Thrusday", "Friday", "Saturday")
week2_sleep <- c(7.1, 7.4, 7.9, 6.5, 8.1, 8.2, 8.9)
print(paste("Display the amount of sleep for week2:",identify_weekdays2, week2_sleep))
```

```
## [1] "Display the amount of sleep for week2: Sunday 7.1"
## [2] "Display the amount of sleep for week2: Monday 7.4"
## [3] "Display the amount of sleep for week2: Tuesday 7.9"
## [4] "Display the amount of sleep for week2: Wednesday 6.5"
## [5] "Display the amount of sleep for week2: Thrusday 8.1"
## [6] "Display the amount of sleep for week2: Friday 8.2"
## [7] "Display the amount of sleep for week2: Saturday 8.9"
```

```r
## 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[1:7])
print(paste("Display the Total amount of sleep for week2:",total_sleep_week2))
```

```
## [1] "Display the Total amount of sleep for week2: 54.1"
```

```r
## Determine if the total sleep in week 1 is less than week 2 by using the < operator
#(total_sleep_week2 < total_sleep_week1)
print(paste("Determine if the total sleep in week 1 is less than week 2:",(total_sleep_week2 < total_sl
```

```
## [1] "Determine if the total sleep in week 1 is less than week 2: FALSE"
```

```r
## Calculate the mean hours slept in week 1 using the `mean()` function
print(paste("Calculate the mean hours slept in week 1:",mean(total_sleep_week1)))
```

```
## [1] "Calculate the mean hours slept in week 1: 48.7"
```

```r
## Create a vector called `days` containing the days of the week.
## Start with Sunday and end with Saturday
days <- c( "Sunday" ,"Monday", "Tuesday", "Wednesday", "Thrusday", "Friday", "Saturday")
print(paste("Days of week 1:",days))
```

```
## [1] "Days of week 1: Sunday"    "Days of week 1: Monday"
## [3] "Days of week 1: Tuesday"   "Days of week 1: Wednesday"
## [5] "Days of week 1: Thrusday"  "Days of week 1: Friday"
## [7] "Days of week 1: Saturday"
```

```r
## Assign the names of each day to `week1_sleep` and `week2_sleep` using the `names` function and `days
names(week1_sleep) <- c( "Sunday" ,"Monday", "Tuesday", "Wednesday", "Thrusday", "Friday", "Saturday")
names(week2_sleep) <- c( "Sunday" ,"Monday", "Tuesday", "Wednesday", "Thrusday", "Friday", "Saturday")
print(paste("Days of week 1:",names(week1_sleep)))
```

```
## [1] "Days of week 1: Sunday"    "Days of week 1: Monday"
## [3] "Days of week 1: Tuesday"   "Days of week 1: Wednesday"
## [5] "Days of week 1: Thrusday"  "Days of week 1: Friday"
## [7] "Days of week 1: Saturday"
```

```r
print(paste("Days of week 2:",names(week2_sleep)))
```

```
## [1] "Days of week 2: Sunday"    "Days of week 2: Monday"
## [3] "Days of week 2: Tuesday"   "Days of week 2: Wednesday"
## [5] "Days of week 2: Thrusday"  "Days of week 2: Friday"
## [7] "Days of week 2: Saturday"
```

```r
## Display the amount of sleep on Tuesday of week 1 by selecting the variable name
week1_sleep[3]
```

```
## Tuesday
##     7.7
```

```r
## Create vector called weekdays from the days vector
weekdays <- days[2:6]
print(paste("Weekdays:",weekdays))
```

```
## [1] "Weekdays: Monday"    "Weekdays: Tuesday"   "Weekdays: Wednesday"
## [4] "Weekdays: Thrusday"  "Weekdays: Friday"
```

```r
## Create vector called weekends containing Sunday and Saturday
weekends <- days[7:1]
print(paste("Days vector:",weekends))
```

```
## [1] "Days vector: Saturday"  "Days vector: Friday"    "Days vector: Thrusday"
## [4] "Days vector: Wednesday" "Days vector: Tuesday"   "Days vector: Monday"
## [7] "Days vector: Sunday"
```

```r
## 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])
print(weekdays1_mean)
```

```
## [1] 7.2
```

```r
print(weekdays2_mean)
```

```
## [1] 7.62
```

```
## Using the weekdays1_mean and weekdays2_mean variables,
## see if weekdays1_mean is greater than weekdays2_mean using the `>` operator
#weekdays1_mean > weekdays2_mean
print(paste("Is weekdays1_mean is greater than weekdays2_mean:",weekdays1_mean > weekdays2_mean))
```

```
## [1] "Is weekdays1_mean is greater than weekdays2_mean: FALSE"
```

```
## Determine how many days in week 1 had over 8 hours of sleep using the `>` operator
# <- c(6.1, 8.8, 7.7, 6.4, 6.2, 6.9, 6.6)
#length(week1_sleep[week1_sleep > 8])
print(paste("Count the days in week 1 had over 8 hours of sleep:",length(week1_sleep[week1_sleep > 8])))
```

```
## [1] "Count the days in week 1 had over 8 hours of sleep: 1"
```

```
## 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 <- c(student01, student02, student03)
grades <- matrix(students_combined,  nrow = 3, byrow = TRUE)

print(grades)
```

```
##       [,1] [,2]
## [1,] 100.0 87.1
## [2,]  77.2 88.9
## [3,]  66.3 87.9
```

```
## Add a new student row with `rbind()`
student04 <- c(95.2, 94.1)
students_combined <- rbind(student01, student02, student03, student04)
grades <- matrix(students_combined, nrow = 4, byrow = TRUE)

rownames(grades) <- c("student01", "student02", "student03", "student04")
colnames(grades) <- c("assignment01", "assignment02")
print(grades)
```

```
##           assignment01 assignment02
## student01        100.0         77.2
## student02         66.3         95.2
## student03         87.1         88.9
## student04         87.9         94.1
```

```
## Add a new assignment column with `cbind()`
assignment04 <- c(92.1, 84.3, 75.1, 97.8)
grades <- cbind(grades, assignment04)
print(grades)
```

```
##           assignment01 assignment02 assignment04
## student01        100.0         77.2         92.1
## student02         66.3         95.2         84.3
## student03         87.1         88.9         75.1
## student04         87.9         94.1         97.8
```

```
## 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")

colnames(grades) <- c("Assignment 1", "Assignment 2", "Assignment 3")
rownames(grades) <- c("Florinda Baird", "Jinny Foss", "Lou Purvis", "Nola Maloney")
print(grades)
```

```
##                Assignment 1 Assignment 2 Assignment 3
## Florinda Baird        100.0         77.2         92.1
## Jinny Foss             66.3         95.2         84.3
## Lou Purvis             87.1         88.9         75.1
## Nola Maloney           87.9         94.1         97.8
```

```
## Total points for each assignment using `colSums()`
colSums(grades)
```

```
## Assignment 1 Assignment 2 Assignment 3
##        341.3        355.4        349.3
```

```
## Total points for each student using `rowSums()`
rowSums(grades)
```

```
## Florinda Baird     Jinny Foss     Lou Purvis   Nola Maloney
##          269.3          245.8          251.1          279.8
```

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

```
##                Assignment 1 Assignment 2 Assignment 3
## Florinda Baird       110.00        84.92       101.31
## Jinny Foss            72.93       104.72        92.73
## Lou Purvis            95.81        97.79        82.61
## Nola Maloney          96.69       103.51       107.58
```

```
## 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 <- factor(genres_vector)
print(factor_genre_vector)
```

```
## [1] Fantasy Sci-Fi  Sci-Fi  Mystery Sci-Fi  Fantasy
## Levels: Fantasy Mystery Sci-Fi
```

```r
## Use the `summary()` function to print a summary of `factor_genre_vector`
summary(factor_genre_vector)
```

```
## Fantasy Mystery  Sci-Fi
##       2       1       3
```

```r
## 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("no", "neutral", "yes")
)
factor_recommendations_vector
```

```
## [1] neutral no      no      neutral yes
## Levels: no < neutral < yes
```

```r
## Use the `summary()` function to print a summary of `factor_recommendations_vector`
summary(factor_recommendations_vector)
```

```
##      no neutral     yes
##       2       2       1
```

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

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

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

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 145          6.7         3.3          5.7         2.5 virginica
## 146          6.7         3.0          5.2         2.3 virginica
## 147          6.3         2.5          5.0         1.9 virginica
## 148          6.5         3.0          5.2         2.0 virginica
## 149          6.2         3.4          5.4         2.3 virginica
## 150          5.9         3.0          5.1         1.8 virginica
```

```r
## 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)
print(characters_df)
```

```
##        name   race in_fellowship ring_bearer  age
## 1    Aragon    Men          TRUE       FALSE   88
## 2     Bilbo Hobbit         FALSE        TRUE  129
## 3     Frodo Hobbit          TRUE        TRUE   51
## 4 Galadriel    Elf         FALSE       FALSE 7000
## 5       Sam Hobbit          TRUE        TRUE   36
## 6   Gandalf   Maia          TRUE        TRUE 2019
## 7   Legolas    Elf          TRUE       FALSE 2931
## 8    Sauron   Maia         FALSE        TRUE 7052
## 9    Gollum Hobbit         FALSE        TRUE  589
```

```r
## Sorting the characters_df by age using the order function and assign the result to the sorted_charac
sorted_characters_df <- characters_df[order(age),]
## Use `head()` to output the first few rows of `sorted_characters_df`
head(sorted_characters_df)
```

```
##      name   race in_fellowship ring_bearer  age
## 5     Sam Hobbit          TRUE        TRUE   36
## 3   Frodo Hobbit          TRUE        TRUE   51
## 1  Aragon    Men          TRUE       FALSE   88
## 2   Bilbo Hobbit         FALSE        TRUE  129
## 9  Gollum Hobbit         FALSE        TRUE  589
## 6 Gandalf   Maia          TRUE        TRUE 2019
```

```r
## 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)
```

```
##      name   race in_fellowship ring_bearer  age
## 2   Bilbo Hobbit         FALSE        TRUE  129
## 3   Frodo Hobbit          TRUE        TRUE   51
## 5     Sam Hobbit          TRUE        TRUE   36
## 6 Gandalf   Maia          TRUE        TRUE 2019
## 8  Sauron   Maia         FALSE        TRUE 7052
## 9  Gollum Hobbit         FALSE        TRUE  589
```