

# daniel\_jackson\_module01\_R\_markdown

Daniel Jackson

September 3, 2023

Module 01 Question 2 example R code

```
# Using simple assignment
```

```
x = 3
```

```
y = 4
```

```
z = 5
```

```
# Use arithmetic with assignments
```

```
var_add = (x + y)
```

```
print(var_add)
```

```
## [1] 7
```

```
var_sub = (z - x)
```

```
print(var_sub)
```

```
## [1] 2
```

```
var_mult = (x * y)
```

```
print(var_mult)
```

```
## [1] 12
```

```
var_div = (z / x)
```

```
print(var_div)
```

```
## [1] 1.666667
```

```
# Create vector of assignments. Print new vector and call the first element
```

```
arith_vec = c(var_add, var_sub, var_mult, var_div)
```

```
print(arith_vec)
```

```
## [1] 7.000000 2.000000 12.000000 1.666667
```

```
arith_vec[1]
```

```
## [1] 7
```

```
# Mutiply the vector by 2 and assign that vector to new vector. Call first and second element
double_arith_vec = ((arith_vec) * 2)
print(double_arith_vec)
```

```
## [1] 14.000000 4.000000 24.000000 3.333333
```

```
double_arith_vec[1:2]
```

```
## [1] 14 4
```

```
# Square arith_vec and assign it to new vector. Print vector with second element not showing
square_arith_vec = ((arith_vec) ^ 2)
print(square_arith_vec)
```

```
## [1] 49.000000 4.000000 144.000000 2.777778
```

```
print(square_arith_vec[-2])
```

```
## [1] 49.000000 144.000000 2.777778
```

```
# Add doubleb_arith_vec and square_artih_vec and assign it to new vector
add_vec = (double_arith_vec) + (square_arith_vec)
print(add_vec)
```

```
## [1] 63.000000 8.000000 168.000000 6.111111
```

```
# Check to see if elements in double_arith_vec are equal to arith_vec
double_arith_vec == arith_vec
```

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

```
# This returns FALSE FALSE FALSE FALSE
```

```
# Divide double_arith_vec by 2 and check to see if elements in new vector are equal to arith_vec
half_double_vec = ((double_arith_vec) / 2)
half_double_vec == arith_vec
```

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

```
# This returned TRUE TRUE TRUE TRUE
```

```
# Create data frame of MLB teams, city they are from, what league they are in,
## when they teams were created, and how many World Series they won. Name the column
### names as well.
mlb_df = data.frame(team_name = c('Red Sox', 'Yankees', 'Dodgers', 'Mariners' ),
                    team_city = c('Boston', 'New York', 'Los Angeles', 'Seattle'),
                    team_league = c('American', 'American', 'National', 'American'),
                    team_origination_year = c(1901, 1903, 1883, 1977),
                    world_series = c(9, 27, 7, 0)
                    )
print(mlb_df)
```

```
##   team_name   team_city team_league team_origination_year world_series
## 1   Red Sox      Boston   American             1901             9
## 2   Yankees    New York   American             1903            27
## 3   Dodgers    Los Angeles National             1883             7
## 4   Mariners    Seattle   American             1977             0
```

```
# Data frame subsetting
```

```
mlb_df[1] # First column with column name
```

```
##   team_name
## 1   Red Sox
## 2   Yankees
## 3   Dodgers
## 4   Mariners
```

```
mlb_df[,1] # Elements of first column listed out
```

```
## [1] "Red Sox" "Yankees" "Dodgers" "Mariners"
```

```
class(mlb_df[1]) # Class is data.frame
```

```
## [1] "data.frame"
```

```
class(mlb_df[,1]) # Class is character
```

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

```
mlb_df[4] # Last column with column name
```

```
##   team_origination_year
## 1             1901
## 2             1903
## 3             1883
## 4             1977
```

```
mlb_df[,4] # Elements of last column listed out
```

```
## [1] 1901 1903 1883 1977
```

```
class(mlb_df[4]) # Class is data.frame
```

```
## [1] "data.frame"
```

```
class(mlb_df[,4]) # Class is numeric
```

```
## [1] "numeric"
```

```
# Call to team_league using $ syntax and print results
print(mlb_df$team_league)
```

```
## [1] "American" "American" "National" "American"
```

```
# Create matrix
matrix_ex_1 = matrix(c(4, 5, 6, 4, 8, 19, 40, 99), nrow = 2, ncol = 4)
print(matrix_ex_1)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    4    6    8   40
## [2,]    5    4   19   99
```

```
# Create same matrix by binding vectors as rows and check to see if elements in
## matrices equal each other
matrix_ex_2 = rbind(c(4, 6, 8, 40), c(5, 4, 19, 99))
print(matrix_ex_2)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    4    6    8   40
## [2,]    5    4   19   99
```

```
(matrix_ex_1) == (matrix_ex_2) # This returned all true
```

```
##      [,1] [,2] [,3] [,4]
## [1,] TRUE TRUE TRUE TRUE
## [2,] TRUE TRUE TRUE TRUE
```

```
# Create same matrix by stacking columns side by side and check to see if elements
## in matrix equal matrix_ex_1
matrix_ex_3 = cbind(c(4,5), c(6,4), c(8, 19), c(40,99))
print(matrix_ex_3)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    4    6    8   40
## [2,]    5    4   19   99
```

```
(matrix_ex_1) == (matrix_ex_3) # This also returned all true
```

```
##      [,1] [,2] [,3] [,4]
## [1,] TRUE TRUE TRUE TRUE
## [2,] TRUE TRUE TRUE TRUE
```

```
# Subset matrix_ex_1
matrix_ex_1[2,4] # Returns element in second row, fourth column
```

```
## [1] 99
```

```
matrix_ex_1[1,] # Returns all elements in first row
```

```
## [1] 4 6 8 40
```

```
matrix_ex_1[,2] # Returns all elements in second column
```

```
## [1] 6 4
```

```
# Functions
```

```
even_num = c(2, 4, 6, 8, 10)
```

```
odd_num = c(1, 3, 5, 7, 9)
```

```
# Length, sum, mean, median, standard deviation on each
```

```
# Length
```

```
length(even_num)
```

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

```
length(odd_num)
```

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

```
length(even_num) == length(odd_num) # This returns true
```

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

```
# Sum
```

```
sum(even_num)
```

```
## [1] 30
```

```
sum(odd_num)
```

```
## [1] 25
```

```
sum(even_num) == sum(odd_num) # This returns false
```

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

```
# Mean
```

```
mean(even_num)
```

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

```

mean(odd_num)

## [1] 5

mean(even_num) == mean(odd_num) # This returns false

## [1] FALSE

# Median
median(even_num)

## [1] 6

median(odd_num)

## [1] 5

median(even_num) == median(odd_num) # This returns false

## [1] FALSE

# Standard deviation
sd(even_num)

## [1] 3.162278

sd(odd_num)

## [1] 3.162278

sd(even_num) == sd(odd_num) # This returns true

## [1] TRUE

# Check working directory then read in weight.csv saved in DSE5001 folder
getwd()

## [1] "/Users/doojerthekid/Documents/Merrimack Grad School Documents/DSE5001/week_1"

# Returned: "/Users/doojerthekid/Documents/Merrimack Grad School Documents/DSE5002 Git Clone"
# Let's set new directory using setwd()
setwd("/Users/doojerthekid/Documents/Merrimack Grad School Documents/DSE5001")
# Check directory now
getwd()

## [1] "/Users/doojerthekid/Documents/Merrimack Grad School Documents/DSE5001"

```

```
# Returned: "/Users/doojerthekid/Documents/Merrimack Grad School Documents/DSE5001"  
# Read in weight.csv data  
weight_df = read.csv("week_1/weight.csv")
```

```
## Warning in read.table(file = file, header = header, sep = sep, quote = quote, :  
## incomplete final line found by readTableHeader on 'week_1/weight.csv'
```