Modifying Data Structures

```
library(mosaic)
library(tidyverse)

# Questions: Given a data structure, such a vector or a data
# frame:
# (1) How do we modify an element?
# (2) How do we add elements to the structure?
# (3) How do we delete elements from the structure?
#
# Let's begin with a vector.
primes <- c(11,13,15,19)
primes

## [1] 11 13 15 19

# Let's change the third element from 15 to 17.
primes[3] <- 17
primes

## [1] 11 13 17 19

# Let's add a fifth element.
primes[5] <- 23
primes

## [1] 11 13 17 19 23

# Let's delete the fourth element.
some_primes <- primes[-4]
some_primes

## [1] 11 13 17 23

# Let's get fancy!
two_digit_numbers <- 10:99
two_digit_numbers

##  [1] 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
33 34
## [26] 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57
58 59
## [51] 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82
83 84
## [76] 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99

# Delete 31 through 51.
some_two_digit_numbers <- two_digit_numbers[-(22:42)]
some_two_digit_numbers
```

```
##  [1] 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 52 53
54 55
## [26] 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78
79 80
## [51] 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99

# Delete the even numbers.
odd_two_digit_numbers <- two_digit_numbers[-(seq(1,89,2))]
odd_two_digit_numbers

##  [1] 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55
57 59
## [26] 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95 97 99

# From the preceding result, delete the non-primes.
odds <- odd_two_digit_numbers # Give the vector a shorter name.
two_digit_primes <- odds[c(1,2,4,5,7,10,11,14,16,17,19,22,25,
                           26,29,31,32,35,36,39,44)]
two_digit_primes

##  [1] 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 81 87 97

# Exercise: What happens if we try this:
# primes[7] <- 31

# Matrix Example: Solve the following system of linear equations
# using Gaussian elimination:
#  3x +  y + 2z = 1
#   x +      z = 0
# -2x + 2y - 3z = 5
#
# Solution: Let's begin with the first two rows of the coefficient
# matrix C
c <- matrix(c(3,1,2,1,0,1),2,3,byrow = TRUE)
c

##      [,1] [,2] [,3]
## [1,]    3    1    2
## [2,]    1    0    1

# Next. we add the third row to C.  To add a row to a matrix, we
# use the rbind() function
c <- rbind(c,c(-2,2,-3))
c

##      [,1] [,2] [,3]
## [1,]    3    1    2
## [2,]    1    0    1
## [3,]   -2    2   -3

# Next, we add the column of constant terms to C to obtain the
# augmented matrix A.  To add a column to a matrix, we use the
# cbind() function.
a <- cbind(c,c(1,0,5))
a

##      [,1] [,2] [,3] [,4]
## [1,]    3    1    2    1
## [2,]    1    0    1    0
## [3,]   -2    2   -3    5

# For the sake of illustration, let's change to element in row 2
# and column 4 of A to 3, and then change it back to 0.
a[2,4] <- 3
a

##      [,1] [,2] [,3] [,4]
## [1,]    3    1    2    1
## [2,]    1    0    1    3
## [3,]   -2    2   -3    5

a[2,4] <- 0
a

##      [,1] [,2] [,3] [,4]
## [1,]    3    1    2    1
## [2,]    1    0    1    0
## [3,]   -2    2   -3    5

# A type 3 elementary row operation: Replace row 3 of A with
# 2(row 2) + (row 3).
a[3,] <- 2*a[2,] + a[3,]
a

##      [,1] [,2] [,3] [,4]
## [1,]    3    1    2    1
## [2,]    1    0    1    0
## [3,]    0    2   -1    5

# A type 2 elementary row operation: Interchange (swap) row 1 and
# row 2 of A.
temp <- a[1,]
a[1,] <- a[2,]
a[2,] <- temp
a

##      [,1] [,2] [,3] [,4]
## [1,]    1    0    1    0
## [2,]    3    1    2    1
## [3,]    0    2   -1    5

# Exercise: A type 1 elementary row operation is to multiply a row
# by a number.  Multiply row 2 of A by 4, and then multiply row 2 by
# 1/4 to restore A to the matrix shown above.
```

```
#
# Finishing our example ...
# Add -3 times row 1 to row 2.
a[2,] <- -3*a[1,] + a[2,]
a
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    1    0
## [2,]    0    1   -1    1
## [3,]    0    2   -1    5
```

```
# Add -2 times row 2 to row 3.
a[3,] <- -2*a[2,] + a[3,]
a
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    1    0
## [2,]    0    1   -1    1
## [3,]    0    0    1    3
```

```
# The above matrix is the augmented matrix for the following system
# of linear equations:
# x +     z = 0
#     y - z = 1
#         z = 3
# This system is easily solved to yield the solution x = -3, y = 4,
# and z = 3.
#
# Data Frames: Working with data frames is similar to working with
# vectors.  First, let's create a small data frame.
name = c("Curly","Joe","Moe","Shemp")
siblings = c(6L,4L,2L,3L)
height = c(74.5,70,59.8,62)
gender = c("M","M","F","M")
my_data1 <- tibble(name,siblings,height,gender)
my_data1
```

```
## # A tibble: 4 × 4
##   name  siblings height gender
##   <chr>    <int>  <dbl> <chr>
## 1 Curly        6   74.5 M
## 2 Joe          4   70   M
## 3 Moe          2   59.8 F
## 4 Shemp        3   62   M
```

```
# Let's access some elements.
# Accessing the element in row 2 and column 3:
my_data1[2,3]
```

```
## # A tibble: 1 × 1
##   height
##    <dbl>
## 1     70
```

```
# Another way:
my_data1[2,"height"]
```

```
## # A tibble: 1 × 1
##   height
##    <dbl>
## 1     70
```

```
# Accessing the third row:
my_data1[3,]
```

```
## # A tibble: 1 × 4
##   name  siblings height gender
##   <chr>    <int>  <dbl> <chr>
## 1 Moe          2   59.8 F
```

```
# Accessing the third column:
my_data1[,3]
```

```
## # A tibble: 4 × 1
##   height
##    <dbl>
## 1   74.5
## 2   70
## 3   59.8
## 4   62
```

```
# Accessing the name of the second column:
names(my_data1)[2]
```

```
## [1] "siblings"
```

```
# Modifying the value in row 3 and column 2:
my_data1[3,2] <- 1
my_data1
```

```
## # A tibble: 4 × 4
##   name  siblings height gender
##   <chr>    <int>  <dbl> <chr>
## 1 Curly        6   74.5 M
## 2 Joe          4   70   M
## 3 Moe          1   59.8 F
## 4 Shemp        3   62   M
```

```
# Changing the name of column 4:
names(my_data1)[4] <- "sex"
my_data1
```

```
## # A tibble: 4 × 4
##   name  siblings height sex
##   <chr>    <int>  <dbl> <chr>
## 1 Curly        6   74.5 M
## 2 Joe          4   70   M
## 3 Moe          1   59.8 F
## 4 Shemp        3   62   M
```

```
# Next, let's add a fifth column to my_data1.  To do this, we use
# cbind().
eyes <- c("green","blue","hazel","blue")
my_data1 <- cbind(my_data1,eyes)
my_data1
```

```
##    name siblings height sex  eyes
## 1 Curly        6   74.5   M green
## 2   Joe        4   70.0   M  blue
## 3   Moe        1   59.8   F hazel
## 4 Shemp        3   62.0   M  blue
```

```
# Lastly, let's create a second data frame with the same column
# names, and then use rbind() to combine them.
name <- c("Tom","Zoe")
siblings <- c(5L,2L)
height <- c(68.4,51)
sex <- c("M","F")
eyes <- c("blue","brown")
my_data2 <- tibble(name,siblings,height,sex,eyes)
my_data2
```

```
## # A tibble: 2 × 5
##   name  siblings height sex   eyes
##   <chr>    <int>  <dbl> <chr> <chr>
## 1 Tom          5   68.4 M     blue
## 2 Zoe          2   51   F     brown
```

```
my_data <- rbind(my_data1,my_data2)
my_data
```

```
##    name siblings height sex  eyes
## 1 Curly        6   74.5   M green
## 2   Joe        4   70.0   M  blue
## 3   Moe        1   59.8   F hazel
## 4 Shemp        3   62.0   M  blue
## 5   Tom        5   68.4   M  blue
## 6   Zoe        2   51.0   F brown
```

```
# An alternate solution:
mydata <- rbind(my_data1,
                list("Tom",5L,68.4,"M","blue"),
                list("Zoe",2L,51,"F","brown"))
mydata

##    name siblings height sex  eyes
## 1 Curly        6   74.5   M green
## 2   Joe        4   70.0   M  blue
## 3   Moe        1   59.8   F hazel
## 4 Shemp        3   62.0   M  blue
## 5   Tom        5   68.4   M  blue
## 6   Zoe        2   51.0   F brown

# Delete row 5 from mydata (without saving the result).
mydata[-5,]

##    name siblings height sex  eyes
## 1 Curly        6   74.5   M green
## 2   Joe        4   70.0   M  blue
## 3   Moe        1   59.8   F hazel
## 4 Shemp        3   62.0   M  blue
## 6   Zoe        2   51.0   F brown

# Finally, let's delete column 2.
mydata[,-2]

##    name height sex  eyes
## 1 Curly   74.5   M green
## 2   Joe   70.0   M  blue
## 3   Moe   59.8   F hazel
## 4 Shemp   62.0   M  blue
## 5   Tom   68.4   M  blue
## 6   Zoe   51.0   F brown
```