

Assignment #1
2016-19516, Sangjun Son

Example 1

1부터 100 사이에서 40개의 정수를 임의로 선택하여 저장하자.

```
1 v = sample(1:100, 40, replace=F)
```

Explanation: 임의의 정수를 선택하기 위해 `sample()` 함수를 통해 정수배열을 생성하고 이를 `v`에 대입하였다.

(1-1) 예제 1에서 만들어진 벡터를 사용하여 5행 8열의 행렬을 생성하시오.

```
1 m = matrix(v, nrow=5, ncol=8)
2 m
```

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,]    6   74   82   34   17   40   88   36
[2,]   20    2   47    9   55   29   37   16
[3,]   59   81   28   77   85   80   71   89
[4,]   67   24   97    4   32   83   15    8
[5,]   86   10   93   63   13   53   76   14
```

Explanation: 40개의 정수 배열을 함수로 재생성하기 위해 `matrix()` 생성자로 5×8 행렬을 생성 후 `m`에 저장하고 출력하였다.

(1-2) 예제 1에서 생성된 행렬에서 2행과 3행만을 추출하여 저장하시오.

```
1 m1 = m[2:3,]
2 m1
```

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,]   20    2   47    9   55   29   37   16
[2,]   59   81   28   77   85   80   71   89
```

Explanation: 위 예제에서 저장한 행렬 `m`의 2 ~ 3행을 가져오기 위해 `indexing`을 하여 `m1`에 저장 후 출력하였다.

(1-3) 예제 1에서 생성된 행렬에서 (1, 4, 7, 8)열을 추출하여 저장하시오.

```
1 m2 = m[,c(1,4,7,8)]
2 m2
```

```
      [,1] [,2] [,3] [,4]
[1,]    6   34   88   36
[2,]   20    9   37   16
[3,]   59   77   71   89
[4,]   67    4   15    8
[5,]   86   63   76   14
```

Explanation: 행렬 `m`의 1, 4, 7, 8 열을 가져오기 위해 두 번째 인덱스에 대한 `indexing`을 하여 `m2`에 저장 후 출력하였다.

(1-4) 예제 1에서 생성된 행렬의 7번째 열의 평균과 분산을 구하시오.

```
1 mean(m[,7]); var(m[,7])
```

```
[1] 57.4
[1] 920.3
```

Explanation: 행렬 `m`의 7열을 가져오기 위해 두 번째 인덱스에 대한 `indexing`을 하고 이렇게 생성된 벡터에 대한 평균과 분산을 구하기 위해 각각 `mean()`, `var()` 함수를 사용하여 구하였다.

Assignment #1
2016-19516, Sangjun Son

Example 2

(2-1) 1부터 20 사이에서 1개의 숫자를 임의로 선택하여 저장하자. 이 숫자가 10 이상이면 "P"를 출력하고 10 미만이면 "NP"를 출력하는 코드를 작성해보시오.

```
1 n = sample(1:20, 1, replace=F)
2 if (n >= 10) print("P") else print("NP")
```

```
[1] "P"
```

Explanation: 1 ~ 20 사이의 숫자 1개를 선택하여 정수 n 에 저장하고 if-else 문을 사용하여 n 의 값이 10 이상일 경우 "P" 아니면 "NP"를 출력하였다.

Example 3

(3-1) 1부터 20 사이에서 8개의 숫자를 임의로 선택하여 저장하자. 8개의 숫자들 각각에 대해 숫자가 10 이상이면 "P"를 출력하고 10 미만이면 "NP"를 출력하는 코드를 작성해보시오.

```
1 v = sample(1:20, 8, replace=F)
2 for (n in v) { if (n >= 10) print("P") else print("NP") }
```

```
[1] "NP"
[1] "NP"
[1] "P"
[1] "P"
[1] "NP"
[1] "P"
[1] "P"
[1] "P"
```

Explanation: 위의 코드를 활용하면 조금 더 수월하게 8개 숫자에 대한 구현한 논리를 재사용할 수 있다. 한 숫자가 아닌 벡터에 대해 for 반복문을 통해 8개에 대한 "P" 또는 "NP"를 출력한다.

Example 4

(4-1) 1부터 100까지의 자연수를 n 으로 저장하고, 그 합을 구하시오.

```
1 n = 1:100
2 sum(n)
```

```
[1] 5050
```

Explanation: 1 ~ 100에 대한 모든 숫자를 벡터 n 에 저장하고 벡터의 합을 $sum()$ 로 구하여 출력하였다.

(4-2) 1부터 200까지의 자연수 중 3의 배수를 $three$ 로 저장하고, 그 합을 구하시오.

```
1 three = seq(from=3, to=200, by=3)
2 sum(three)
```

```
[1] 6633
```

Explanation: 1 ~ 200 사이의 모든 3의 배수를 저장하기 위해 $seq()$ 를 통해 초기값 3, 경계값 200, 간격 3을 주어 구간에 해당하는 모든 숫자 벡터를 $three$ 에 저장한다. $three$ 벡터 원소의 합을 $sum()$ 로 구하여 출력하였다.

Assignment #1
2016-19516, Sangjun Son

(4-3) 위 예제 4-2에서 저장한 `three` 벡터의 표준편차를 구하시오. 단, 소수점 둘째자리까지 반올림한 값을 출력한다.

```
1 round(sd(three), 2)

[1] 57.59
```

Explanation: 벡터의 표준편차를 구하기 위해 `sd()` 함수를 사용하고 소수 둘째자리까지 반올림하여 표현한 실수값을 구하여 출력하였다.

Example 5

다음과 같은 행렬을 변수명 `m`으로 저장한다.

$$m = \begin{pmatrix} 7 & 3 \\ 6 & 1 \\ 9 & 10 \end{pmatrix}$$

(5-1) 행렬 `m`의 각 성분을 4로 나눈 나머지를 성분으로 갖는 행렬을 구하시오.

```
1 m = cbind(c(7, 6, 9), c(3, 1, 10))
2 m %% 4

      [,1] [,2]
[1,]    3    3
[2,]    2    1
[3,]    1    2
```

Explanation: 열 벡터를 만들어 열 방향으로 계속 쌓아 행렬을 만들기 위해 `cbind()`를 사용해 행렬 `m`을 생성하고 각 원소 별로 나머지 연산 `%%`을 수행하여 출력하였다.

(5-2) 행렬 `m`의 1행에는 10씩을, 2행에는 20씩을, 3행에는 30씩을 더한 값을 성분으로 갖는 행렬을 구하시오.

```
1 rbind(m[,1]+10, m[,2]+20, m[,3]+30)

      [,1] [,2]
[1,]   17   13
[2,]   26   21
[3,]   39   40
```

Explanation: 각각의 행 방향 인덱싱을 통해 i 행에 해당하는 벡터를 추출하고 원소별 덧셈 $(+10 \cdot i)$ 을 하여 행 방향으로 쌓아 행렬을 만들고 출력하였다.