StatR 510 HW 1 Solutions

4.

(a)

This is exactly what the `seq` function is meant to do.

```
seq(from = 2, to = 100, by = 2)
```

```
##  [1]   2   4   6   8  10  12  14  16  18  20  22  24  26  28  30  32  34  36  38
## [20]  40  42  44  46  48  50  52  54  56  58  60  62  64  66  68  70  72  74  76
## [39]  78  80  82  84  86  88  90  92  94  96  98 100
```

(b)

The `:` operator is convenient shorthand when your sequence increases by 1. Multiplying by `2` gives us all even numbers.

```
(1:50) * 2
```

```
##  [1]   2   4   6   8  10  12  14  16  18  20  22  24  26  28  30  32  34  36  38
## [20]  40  42  44  46  48  50  52  54  56  58  60  62  64  66  68  70  72  74  76
## [39]  78  80  82  84  86  88  90  92  94  96  98 100
```

(c)

The `rep` function *repeats* the first argument the specified number of times. The `cumsum` function takes the cumulative sum of the resulting vector.

```
cumsum(rep(2, 50))
```

```
##  [1]   2   4   6   8  10  12  14  16  18  20  22  24  26  28  30  32  34  36  38
## [20]  40  42  44  46  48  50  52  54  56  58  60  62  64  66  68  70  72  74  76
## [39]  78  80  82  84  86  88  90  92  94  96  98 100
```

5.

Let's create a vector on which to operate. Yours probably looked different, but the ideas are the same.

```
x <- 1:50
```

(a)

$\sum_{i=1}^{n} X_i$

```
sum(x)
```

```
## [1] 1275
```

(b)

$\text{SS} = \sum_{i=1}^{n} X_i^2$

```
sum(x^2)
```

```
## [1] 42925
```

(c)

$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$

```
mean(x)
```

```
## [1] 25.5
```

(d)

$\sum_{i=1}^{n}(X_i - \overline{X})^2$

```
sum((x - mean(x))^2)
```

```
## [1] 10412.5
```

(e)

$\text{SS} - n\overline{X}^2$

```
sum(x^2) - length(x) * mean(x)^2
```

```
## [1] 10412.5
```

Optional proof

$$\begin{aligned}
\sum_{i=1}^{n}(X_i - \overline{X})^2 &= \sum_{i=1}^{n} \left(X_i - \overline{X}\right)\left(X_i - \overline{X}\right) \\
&= \sum_{i=1}^{n} \left(X_i^2 - 2X_i\overline{X} + \overline{X}^2\right) \\
&= \sum_{i=1}^{n} X_i^2 - 2\overline{X}\sum_{i=1}^{n} X_i + \sum_{i=1}^{n} \overline{X}^2 \\
&= \sum_{i=1}^{n} X_i^2 - 2\overline{X}n\overline{X} + n\overline{X}^2 \qquad \left(\text{since } \sum_{i=1}^{n} 1 = n\right) \\
&= \sum_{i=1}^{n} X_i^2 - n\overline{X}^2 \\
&= SS - n\overline{X}^2.
\end{aligned}$$

6.

In setting this problem up, I use the `set.seed()` command below. This is a way to make random number generation reproducible. This way, you can get the same results I did below by using the same seed.

```r
set.seed(1)

Names <- c("Alana", "Bettie", "Consuela", "Dona", "Elaine", "Frances", "Gerri", "Helene",
    "Ichabod", "Jin", "Kenyatta", "Larry", "Mikhailo", "Nick", "Odin")
Sex <- c(rep("F", 8), rep("M", 7))
Grades <- round(runif(15, 50, 100))
Grades
```

```
##  [1] 63 69 79 95 60 95 97 83 81 53 60 59 84 69 88
```

You can see the 15 different grades generated from a uniform distribution between 50 and 100.

(a) Which grades were greater than or equal to 90?

We use the square brackes `[` and `]` to *subset* the `Grades` vector.

```r
gt90 <- Grades[Grades >= 90]
gt90
```

```
## [1] 95 95 97
```

(b) Who earned the highest grades?

To extract just the names:

```r
Names[gt90]
```

```
## [1] NA NA NA
```

(c) Who earned the lowest grades?

We can use the expression `Grades < 60` inside the square brackets *without* first assigning it to a variable name as we did with `gt90`.

```r
Names[Grades < 60]
```

```
## [1] "Jin"   "Larry"
```

(d) Extracting the grades of the male and female students

```r
Grades.M <- Grades[Sex == "M"]
Grades.F <- Grades[Sex == "F"]
Grades.F
```

```
## [1] 63 69 79 95 60 95 97 83
```

(e) Average grades of the male and female students

```r
sum(Grades.F) / length(Grades.F)
```

```
## [1] 80.125
```

```r
sum(Grades.M) / length(Grades.M)
```

```
## [1] 70.57143
```

Clearly the guys need to step up their game!

Problem 8.

(a)

The ls function returns a character vector of the names of all objects in our global environment. To remove them all, we use that as list argument to the rm function:

```
rm(list = ls())
```

(b)

When adding two vectors of different length, the shorter of the two will be repeated as many times as necessary to complete the addition. Adding a vector of length 8 to a vector of length 2, for example, will repeat the vector of length 2 four times. In the example given, what R is really doing is c(0,2,0,2) + c(3,4,5,6). I R this is called *recycling*.

(c)

A vector can only contain objects of the same class, e.g., character or numeric. A list, by contrast, can be a collection of just about anything: you can construct a list where its first object is a vector, the second is a data frame, and the third is another list.

(d)

The length and mode functions return information *about* an object. Other useful descriptive functions are names, str, and class.

(e)

In the help file for a function, the "Value" section describes the output of the function.