Lab #3 Activities

Thanh Dat Nguyen - THANHDAN

Instructions:

- Fill in your name in line 3.
- Knit to pdf to read the questions in a more readable format.
- Fill in the code chunks below and answer the questions with text responses. It is recommended that you knit to pdf after you fill in each code chunk. Be sure when adding in text responses to never copy-paste symbols from outside of the document.
- Your responses must use code that was covered in class; other methods to solve the problems will not be accepted.
- Submit your knit pdf file to Crowdmark.

A reminder that the R code we have covered in class is available on our STAT 2150 A01 UM Learn page, under Content > Course Material.

Your knit pdf file should show the result answering each question. To do this, after creating an R object, you should also print it in a new line within the code chunk.

Question 1:

The volume of a cylinder is given by $V = \pi r^2 h$, where r is the radius of the circle and h is the height of the cylinder. Write two functions, one of them calling the other, that calculates the volume of a cylinder when the radius and the height are provided. One of your functions should calculate the area of the base of the cylinder (given that the circular base has radius r). Your second function should take the radius and the height as arguments, take the area returned from the first function, and should return the volume.

Once your functions are written, use them to calculate the volume of a cylinder (in cubic centimetres) with a radius of 5 cm and a height of 20 cm.

The R code for the number π is simply `pi`:

```
pi
```

```
## [1] 3.141593
```

```
area <- function(r) {
  return(pi * r^2)
}

volume <- function(r, h) {
  base_area <- area(r)
  return(base_area * h)
}

cylinder_volume <- volume(5, 20)
print(paste("The volume of the cylinder is", cylinder_volume, "cubic centimeters."))
```

```
## [1] "The volume of the cylinder is 1570.7963267949 cubic centimeters."
```

Question 2:

The dataset `state.x77` is built-in to R. It contains data on the 50 states of the United States. (The data is quite old.) The help documentation at `?state.x77` gives details on the dataset (and some other related datasets). The data in `state.x77` is in the format of a matrix. Let's convert the matrix to a data frame and store the data frame as an object called `new_state_data`:

```r
new_state_data = as.data.frame(state.x77)
```

In the data frame `new_state_data`, there is a column labeled `Income` consisting of per capita income (in $). (Remember, this data is quite old!) Write the R code that determines the proportion of all states with per capita income over $5,000.

```r
# Determine proportion of states with per capita income over $5,000
proportion <- sum(new_state_data$Income > 5000) / nrow(new_state_data)
proportion
```

```
## [1] 0.16
```

Create a new variable called `income_cat` that keeps track if the per capita income for each state is below $4,000 (in which case, code it as `1`); between $4,000 and $5,000, inclusive (in which case, code it as `2`); or above $5,000 (in which case, code it as `3`).

Note: if you want to check if a value is between two numbers, we have to use a logical AND in R. For example:

```r
x = 3
x >= 0 & x <= 5
```

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

```r
# Does not work:
# 0 <= x <= 5
```

```r
# Create the new variable income_cat
new_state_data$income_cat <- NA


new_state_data$income_cat[new_state_data$Income < 4000] <- 1  # Below $4,000

new_state_data$income_cat[new_state_data$Income >= 4000 & new_state_data$Income <= 5000] <- 2  # Betwee

new_state_data$income_cat[new_state_data$Income > 5000] <- 3  # Above $5,000

new_state_data$income_cat
```

```
##  [1] 1 3 2 1 3 2 3 2 2 2 2 2 3 2 2 2 1 1 1 3 2 2 2 1 2 2 2 3 2 3 1 2 1 3 2 1 2 2
## [39] 2 1 2 1 2 2 1 2 2 1 2 2
```

After `income_cat` is created, uncomment the below line of code and knit to show the number of states in each of the three income categories.

```r
table(new_state_data$income_cat)
```

```
##
##  1  2  3
## 13 29  8
```

Question 3:

Suppose we have a vector `x` of 100 randomly generated values between 0 and 1 as shown below. (The set.seed() line guarantees that we get a reproducible sequence of random values.)

```r
set.seed(246)
x = runif(100,0,1)
```

Write code using a `while()` loop that will keep adding elements from `x`, starting with the first component, until the sum of the first k components of `x` exceeds 10. Create a list of length 2 that contains: (1) the value k and (2) the sum immediately after it has surpassed 10. Print that list.

```r
set.seed(246)
x <- runif(100, 0, 1)
k <- 1
sum_x <- x[1]

while (sum_x <= 10 && k < length(x)) {
  k <- k + 1
  sum_x <- sum_x + x[k]
}

result <- list(k, sum_x)
print(result)
```

```
## [[1]]
## [1] 22
##
## [[2]]
## [1] 10.47443
```