HW 2, STAT 650
Due: Friday, September 13

Directions: The assignment should be completed using Quarto and submitted to Canvas as a self-contained HTML or PDF file.

Exercise 1

Write a function called `sum_squares` that for any given positive integer n computes the sum $1^2 + 2^2 + \cdots + n^2$. Use your function to find the value of the sum when $n = 5$, $n = 10$ and $n = 50$.

Exercise 2

This exercise uses the `airquality` data frame:

```
head(airquality)
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

(a) Use a `for` loop to count the number of `NA` values in each column of the `airquality` data frame (Hint: make use of the `is.na()` function in your code). The output of your code should look like this:

```
##   Ozone Solar.R    Wind    Temp   Month     Day
##      37       7       0       0       0       0
```

(b) Repeat part (a), but this time use the `apply()` function instead of a `for` loop.

Exercise 3

The following R code randomly simulates rolling two dice, and then takes the sum.

```
dice <- sample(1:6, size = 2, replace = TRUE)
dice
```

```
## [1] 3 6
```

```
sum(dice)
```

```
## [1] 9
```

Use a `for` loop to repeatedly simulate rolling two dice 10,000 times, and compute the proportion of rolls where the sum is 7. (Hint: the proportion should be close to 1/6)

Exercise 4

Recall that a confidence interval for a population mean is given by the formula $\bar{x} \pm t \cdot s/\sqrt{n}$, where $\bar{x}$ is the sample mean, s is the sample standard deviation, n is the sample size, and t is the critical value. In R, the command $t =$ `abs(qt((1-cl)/2, df = n-1))` computes the critical value for a specified confidence level, `cl`, and degrees of freedom, $n - 1$. For example:

```
# cl = 0.95, n = 100
abs(qt((1-0.95)/2, df = 99))
```

```
## [1] 1.984217
```

```
# cl = 0.9, n = 100
abs(qt((1-0.9)/2, df = 99))
```

```
## [1] 1.660391
```

Write a function called `compute_ci()` that computes a confidence interval for a population mean. The function should have three arguments:

- `x`, a numeric vector.
- `cl`, the confidence level. Set the default to `cl = 0.95`.
- `na_rm`, a logical value (`TRUE` / `FALSE`) indicating whether `NA` values should be removed. Set the default to `na_rm = FALSE`.

This is what the output of your function should look like:

```
library(mosaicData)
compute_ci(CPS85$wage) # uses default cl = 0.95
```

```
## [1] 8.587194 9.460933
```

```
compute_ci(CPS85$wage, cl = 0.8)
```

```
## [1] 8.738705 9.309422
```

```
# Ozone column has NA values
compute_ci(airquality$Ozone, cl = 0.9, na_rm = TRUE)
```

```
## [1] 37.05046 47.20816
```