In-class Assignment 12

Andrew Shao (NetID: as13381)

Question 1 (1 pt): Write your own function to compute the variance of a numeric vector. Then use your function to compute the variance of `1:100`.

Answer:

```
my_variance <- function(vec) {
  return(
    sum((vec - mean(vec, na.rm = T)) ** 2) / (sum(!is.na(vec)) - 1)
  )
}

my_variance(1:100)
```

```
## [1] 841.6667
```

Question 2 (1 pt): Write a `fizzbuzz` function. It takes a single number as input. If the number is divisible by three, it returns `"fizz"`. If it's divisible by five it returns `"buzz"`. If it's divisible by three and five, it returns `"fizzbuzz"`. Otherwise, it returns the number. Output the results of `fizzbuzz(6)`, `fizzbuzz(10)`, `fizzbuzz(15)` and `fizzbuzz(2)`.

Answer:

```
fizzbuzz <- function(x) {
  if (x %% 3 == 0) {
    if (x %% 5 == 0) {
      return('fizzbuzz')
    }
    return('fizz')
  }
  if (x %% 5 == 0) {
    return('buzz')
  }
  return(x)
}

sapply(c(6, 10, 15, 2), fizzbuzz)
```

```
## [1] "fizz"     "buzz"     "fizzbuzz" "2"
```

Question 3 (1 pt): Carefully read the documentation of the base R function `cut()`. Modify the following `cold_hot` function by using `cut()` to simplify the set of nested if-else statements. After the modification, output the results of `cold_hot(-5)`, `cold_hot(0)`, `cold_hot(10)`, `cold_hot(20)` and `cold_hot(30)`.

```
cold_hot <- function(temp){
  if (temp < 0) {
    "freezing"
  } else if (temp < 10) {
    "cold"
  } else if (temp < 20) {
    "cool"
  } else if (temp < 30) {
    "warm"
  } else {
    "hot"
  }
}
```

Answer:

```
cold_hot <- function(temp) {
  cut(temp,
      breaks = c(-Inf, 0:3*10, +Inf),
      labels = c('freezing', 'cold', 'cool', 'warm', 'hot'),
      right = F)
}
sapply(c(-5, 0, 10, 20, 30), cold_hot)
```

```
## [1] freezing cold     cool     warm     hot
## Levels: freezing cold cool warm hot
```

Question 4 (1 pt): A function called `commas()` is defined below. What does `commas(letters, collapse = "-")` do? Why? (If there is an error when knitting, use the code chunk option `error = TRUE`.)

```
commas <- function(...) {
  str_c(..., collapse = ", ")
}
```

Answer: `commas(letters, collapse = "-")` produces an error, because witin the `commas` function the `str_c()` function call already has a set value for the `collapse` argument so by including `collapse = "-"`, `commas` will pass 2 `collapse` argument values which will encounter an error.

```
commas(letters, collapse = "-")
```

```
## Error in str_c(..., collapse = ", "): formal argument "collapse" matched by multiple actual arguments
```

Question 5 (1 pt): Carefully read the documentation of the function `cor()` of package `stats`. The default value for the `method` argument to `cor()` is `c("pearson", "kendall", "spearman")`. What does that mean? What value is used by default?

Answer: It means the `method` argument must be passed one of `"pearson"`, `"kendall"`, or `"spearman"` or an appropriate abbreviation else an error will occur. `"pearson"` is used by default.