Homework 2

Zooey Nguyen

10/14/2019

Question 1

The following output is a vector of the values of `sin(k*pi)` for values of k ranging from 1 to 10, making use of the fact that the `sin()` function is vectorised, meaning that when it is applied to a vector, it is applied to all elements of the vector individually. The result should theoretically be a vector of zeroes, but because of R's issues with rounding, it instead ends up as a vector of extremely small values that are essentially zero.

```
k_vector <- (1:10)*pi
sin(k_vector)
```

```
##  [1]  1.224606e-16 -2.449213e-16  3.673819e-16 -4.898425e-16  6.123032e-16
##  [6] -7.347638e-16  8.572244e-16 -9.796851e-16  1.102146e-15 -1.224606e-15
```

Question 2

a. Vector of commute times.

```
commutes <- c(17, 16, 20, 24, 22, 15, 21, 15, 17, 22)
```

b. His average commute time can be found by typing `mean(commute_times)`, which outputs 18.9. His maximum commute time can be found with `max(commute_times)`, which outputs 24. His minimum commute time can be found with `min(commute_times)`, which outputs 15.

c. Updating vector and mean.

```
commutes[which(commutes == 24)] <- 18
mean(commutes) # corrected mean commute time
```

```
## [1] 18.3
```

d. Updating the RData file.

```
load("commute_times.RData")
commute_times[which(commute_times == 24)] <- 18
mean(commute_times) # corrected mean commute time
```

```
## [1] 18.79452
```

Question 3

a. Writing the skewness function.

```
skewness <- function(x) {
  mean <- sum(x)/length(x)
  numer <- sqrt(length(x)) * sum((x-mean)^3)
  denom <- (sum((x-mean)^2))^(3/2)
  numer/denom
}
```

b. Skewness of Andy's commute times.

```
skewness(commute_times)
```

```
## [1] 0.40136
```

c. Writing the one-pass skewness function.

```
skewness_one <- function(x) {
  n <- length(x)
  mean <- sum(x)/n
  numer <- sqrt(n) * (sum(x^3) - 3*mean*sum(x^2) + 2*n*mean^3)
  denom <- (sum(x^2) - n*mean^2)^(3/2)
  numer/denom
}
skewness_one(commute_times) # should match 0.40136
```

```
## [1] 0.40136
```

d. Comparing skewness formulas. The answers do not differ from the results in (b) and (c) because when the commute times are multiplied by 1e10, the relations between the variables do not change much, except that 1e10 is factored into the individual multiplications. As such, the differences between each term in the skewness functions are significant enough compared to the size of the initial inputs to not lose a drastic amount of precision in our final answer.

```
commute_times_large <- commute_times*1e10
skewness(commute_times_large)
```

```
## [1] 0.40136
```

```
skewness_one(commute_times_large)
```

```
## [1] 0.40136
```

Question 4

```
two_exp <- 2^(1:15)
three_power <- (1:15)^3
two_exp[two_exp > three_power]
```

```
## [1]     2  1024  2048  4096  8192 16384 32768
```

Question 5

```
na_only <- function(x) {
  is.na(x) & !is.nan(x)
}

test <- c(7,NA,4,NA,3,Inf,NaN)
na_only(test) # should output FALSE TRUE FALSE TRUE FALSE FALSE FALSE
```

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

Question 6

NA is treated as unknown in R, meaning it represents some boolean that could be either true or false. So when a logical operation has an ambiguous result, NA is outputted. For expressions where the logical operation is absolutely true or absolutely false regardless of whether NA would be true or false, then you get the absolute answer. This is seen when `4 > 3 | NA` and `4 < 3 & NA` are evaluated to TRUE and FALSE, respectively. In the first case, the left hand side is true and the right hand side is ambiguous, but because you're comparing them to see whether the left hand side OR right hand side is true, it can evaluate to TRUE anyway. For the other two cases, they evaluate to NA because the output also depends on whether the right-hand side is TRUE or FALSE.

Question 7

Pseudocode:

```
function is_TRUE(vector x) {
    • each element is compared to TRUE by checking exact match.
    • this vector then changes NA values to false: check it with an AND statement of a vector that identifies which values are NOT NA, then all matches of NA turn to FALSE.
    • output vector of comparisons.
}
function is_FALSE(vector x) {
    • each element is compared to FALSE by checking exact match.
    • this vector then changes NA values to false: check it with an AND statement of a vector that identifies which values are NOT NA, then all matches of NA turn to FALSE.
    • output vector of comparisons for element AND FALSE.
}
```

```
is_TRUE <- function(x) {
  (x == TRUE) & !is.na(x)
}

is_FALSE <- function(x) {
  (x == FALSE) & !is.na(x)
}
test1 <- c(TRUE, FALSE, NA)
test2 <- c(1, 0, 1)
is_TRUE(test1)
```

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

```
is_FALSE(test1)
```

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

```
is_TRUE(test2)
```

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

```
is_FALSE(test2)
```

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

Question 8

```
jerry <- 2 : (8 * (5 %% 3) ^ -2) : (7 > 2)
jerry
```

```
## [1] 2 1
```