HW1

Steven Black

2/3/2019

## Homework 1 for IS677 Intro to Data Science

### 1. You keep track of your mileage each time you fill up your automobile. At your last 6 fillups the mileage numbers were: 65311, 65624, 65908, 66219, 66499, 66821, 67145, and 67447. Enter these numbers into a vector using R. (10 points)

mileage\_at\_last\_six\_fills <- c(65311, 65624, 65908, 66219, 66499, 66821, 67145, 67447)  
  
print(mileage\_at\_last\_six\_fills)

## [1] 65311 65624 65908 66219 66499 66821 67145 67447

### 2. Use the function diff() on the data. What does it indicate? (10 points)

diff(mileage\_at\_last\_six\_fills)

## [1] 313 284 311 280 322 324 302

### 3. Use the max() function to find the maximum, the mean() function to find the average, sd() function to find the standard deviation, and the min() function to get the minimum number of miles between fill-ups. (10 points)

maximum <- max(mileage\_at\_last\_six\_fills)  
average <- mean(mileage\_at\_last\_six\_fills)  
sdev <- sd(mileage\_at\_last\_six\_fills)  
minimum <- min(mileage\_at\_last\_six\_fills)  
paste("max: ", maximum, " avg: ", average, " stdev: ", sdev, " min: ", minimum)

## [1] "max: 67447 avg: 66371.75 stdev: 745.923540700673 min: 65311"

### 4. Enter your commute times for two weeks (10 days) in minutes: 17 16 20 24 22 15 21 15 17 22. Change the entry for 24 to 18. (10 points)

commute\_times <- c(17, 16, 20, 24, 22, 15, 21, 15, 17, 22)  
replace(commute\_times, commute\_times==24, 18)

## [1] 17 16 20 18 22 15 21 15 17 22

### 5. How many times were your commute 20 minutes or more? Hint: use the sum() function.

sum(commute\_times>=20)

## [1] 5

### 6. Suppose we assume: x = c(1,3,5,7,9), y = c(2,3,5,7,11,13). For the following commands, guess the value first, than try it out in R. Explain the results of each in one sentence or less. (50 points)

x = c(1,3,5,7,9)  
y = c(2,3,5,7,11,13)

1. x+1

x+1

## [1] 2 4 6 8 10

Adding a scalar to a vector increases every element in the vector by the scalar.

1. y\*2

y\*2

## [1] 4 6 10 14 22 26

Multiplying a scalar by a vector multiplies every element in the vector by thet scalar.

1. length(x) and length(y)

len\_x <- length(x)  
len\_y <- length(y)  
paste(len\_x, " ", len\_y)

## [1] "5 6"

The length function returns the length of the vector passed to it.

1. x + y

x + y

## Warning in x + y: longer object length is not a multiple of shorter object  
## length

## [1] 3 6 10 14 20 14

Adding two vectors returns a vector returns a vector with the sum of every ith element.

1. sum(x>5) and sum(x[x>5])

x\_gt\_five <- sum(x>5)  
x\_w\_gt\_five <- sum(x[x>5])  
paste(x\_gt\_five, x\_w\_gt\_five)

## [1] "2 16"

The first command returns the number of elements in x greater than five and the second returns the sum of those elements.

1. sum(x>5 | x< 3) # read | as ‘or’, & as ‘and’

sum(x>5 | x< 3)

## [1] 3

This returns the sum of all the numbers in x that are greater than five or less than three.

1. y[3]

y[3]

## [1] 5

This returns the 3rd element in y.

1. y[-3]

y[-3]

## [1] 2 3 7 11 13

This returns the vector y without the 3rd element.

1. y[x] (What is NA?)

y[x]

## [1] 2 5 11 NA NA

This returns the elements in y corresponding to the index values of the elements in x. In R NA stands for a missing value.

1. y[y>=7]

y[y>=7]

## [1] 7 11 13

This returns the values of y greater than or equal to 7.