Homework 2, My Name.

Please submit the solution on Canvas into the corresponding assignment (e.g. "Homework #1") in the form of R Markdown report, knitted into either of the available formats (HTML, pdf or Word). Provide only code and output. NO NEED TO COPY THE PROBLEM FORMULATION (!)

Problem #1 (make sure to include code with outputs)

For the *fl_student_survey.csv* file, proceed to pick a quantitative variable (can be the same as in previous homework):

1. Describe its center via two main corresponding measures introduced in class. Which do you think is more appropriate? Why?
2. Describe its variability via at least two methods (that *do not* use measures of position) from the class. Advantages/Downsides of each method?
3. Provide the plot outlining the five-number summary for measures of position. Are there any outliers? Report the main variability measure. In general, what are the advantages of that measure as opposed to standard deviation?

Problem #2

2.21,

2.34,

2.37 (no cheating with *median*() function; *mean*() is fine),

2.46 (DON'T use *range*() or *sd*()),

2.54,

2.62,

2.66,

2.76 (only parts a-b-c; no need to plot)

Problem #3

Define your own function in *R* and demonstrate its functionality:

1. Function needs to contain:
   a. at least one assignment operation ('<-')
   b. at least one *required* argument (without a default value)
   c. at least one argument with a default value
   d. at least one conditional *if*-expression
2. Demonstrate:
   a. 3 calls to show that function works properly on various *required* inputs}
   b. a call in which you specify a non-default value

Function idea examples: check if a string/sequence has a certain property, or make a cyclic algebraic calculation, or provide various plots for inputted sequence/data…

Example: Function *is.increasing*() allows to check if a sequence of numbers is monotonically increasing (returns TRUE) or not (returns FALSE). Upon user's request, it can also print out the consecutive differences.

```
is.increasing <- function(x,printout=FALSE){
  n <- length(x)
  dif.vec <- x[2:n]-x[1:(n-1)]
  if (printout == TRUE) print(dif.vec)
  if (sum(dif.vec<=0) != 0) return(F)  # Important note: function stops after executing the
  return(T)                            # first 'return' statement it gets to
}

x <- c(1:10)
is.increasing(x)
```

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

```
y <- c(5,5,5)
is.increasing(y)
```

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

```
z <- c(1,5,3,7,9)
is.increasing(z)
```

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

```
is.increasing(z,printout=T)   # The call where you specify the non-default value
```

```
## [1]  4 -2  4  2
```

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