CIND 123 - Data Analytics: Basic Methods

Stephanie Boissonneault

Assignment 1 (10%)

Stephanie Boissonneault

CIND123 D30, Student ID 500616408

# Instructions

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. Review this website for more details on using R Markdown <http://rmarkdown.rstudio.com>.

Use RStudio for this assignment. Complete the assignment by inserting your code wherever you see the string “#INSERT YOUR ANSWER HERE”.

When you click the **Knit** button, a document (PDF, Word, or HTML format) will be generated that includes both the assignment content as well as the output of any embedded R code chunks.

**NOTE**: YOU SHOULD NEVER HAVE install.packages IN YOUR CODE; OTHERWISE, THE Knit OPTION WILL GIVE AN ERROR. COMMENT OUT ALL PACKAGE INSTALLATIONS.

Submit **both** the rmd and generated output files. Failing to submit both files will be subject to mark deduction. PDF or HTML is preferred.

## Sample Question and Solution

Use seq() to create the vector .

seq(3, 30, 2)

## [1] 3 5 7 9 11 13 15 17 19 21 23 25 27 29

seq(3, 29, 2)

## [1] 3 5 7 9 11 13 15 17 19 21 23 25 27 29

## Question 1 (32 points)

## Q1a (8 points)

Create and print a vector x with all integers from 4 to 115 and a vector y containing multiples of 4 in the same range. Hint: use seq()function. Calculate the difference in lengths of the vectors x and y. Hint: use length()

x <- 4:115  
x

## [1] 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21  
## [19] 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39  
## [37] 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57  
## [55] 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75  
## [73] 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93  
## [91] 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111  
## [109] 112 113 114 115

y <- seq(from = 4, to = 115, by = 4)  
y

## [1] 4 8 12 16 20 24 28 32 36 40 44 48 52 56 60 64 68 72 76  
## [20] 80 84 88 92 96 100 104 108 112

#Difference in length  
length(x)-length(y)

## [1] 84

## Q1b (8 points)

Create a new vector, y\_square, with the square of elements at indices 1, 3, 7, 12, 17, 20, 22, and 24 from the variable y. Hint: Use indexing rather than a for loop. Calculate the mean and median of the FIRST five values from y\_square.

y\_square <- y[c(1, 3, 7, 12, 17, 20, 22, 24)]\*\*2   
y\_square

## [1] 16 144 784 2304 4624 6400 7744 9216

mean(y\_square[1:5])

## [1] 1574.4

median(y\_square[1:5])

## [1] 784

## Q1c (8 points)

For a given factor variable of factorVar <- factor(c(1, 6, 5.4, 3.2)), would it be correct to use the following commands to convert factor to number?

as.numeric(factorVar)

If not, explain your answer and provide the correct one.

# When assigning a numeric vector into a factor variable, the factor will first recognize the different levels of the variables. In this case, there are four levels and they are recognized as 1, 3.2, 5.4, and 6.  
factorVar <- factor(c(1, 6, 5.4, 3.2))  
factorVar

## [1] 1 6 5.4 3.2  
## Levels: 1 3.2 5.4 6

#The factor internally assigns an integer number starting from 1 to each corresponding category in the level. In this case, 1 is the internal level of 1, 3.2 is the internal level 2, 5.4 is the internal level 3, and 6 is the internal level 4.  
  
#Because these values are now internally assigned a number corresponding to their level, the as.numeric() function will convert the factor to a numeric class, but it will display the numbers corresponding to each value's internal level:  
  
numericVar <- as.numeric(factorVar)  
numericVar

## [1] 1 4 3 2

class(numericVar)

## [1] "numeric"

#To display the factor variable as numeric variables, we would need to first convert the factor to a character vector, followed by a numeric vector to ensure that the original numeric values are displayed rather than their associated internal level.   
  
numericVar2 <- as.numeric(as.character(factorVar))  
numericVar2

## [1] 1.0 6.0 5.4 3.2

class(numericVar2)

## [1] "numeric"

## Q1d (8 points)

A comma-separated values file dataset.csv consists of missing values represented by Not A Number (null) and question mark (?). How can you read this type of files in R? NOTE: Please make sure you have saved the dataset.csv file at your current working directory.

#I would set the header to false because the first row of the dataset has no header, and if "null" and "?" both represent missing values (rather than NULL or nothing), I would read these as NA strings using the following command:   
  
dataset <- read.csv("dataset.csv", sep = ",", header = FALSE, stringsAsFactor = FALSE, na.strings= c("null", "?"))  
  
dataset

## V1 V2 V3 V4 V5 V6 V7 V8 V9 V10  
## 1 1 2 3 4 5 6 7 8 9 10  
## 2 11 12 13 14 15 16 17 18 19 20  
## 3 21 22 23 24 25 26 27 28 29 30  
## 4 31 32 33 34 35 36 37 38 39 40  
## 5 41 42 43 44 45 NA 47 48 49 50  
## 6 51 52 53 NA 55 56 57 NA 59 60  
## 7 61 62 63 64 65 66 67 68 69 70  
## 8 71 72 NA 74 75 76 77 78 79 80  
## 9 81 82 83 84 85 86 87 88 89 NA  
## 10 91 92 93 94 95 96 97 98 99 100  
## 11 NA 102 103 104 105 106 107 108 109 110  
## 12 111 112 113 114 115 116 117 118 119 120  
## 13 121 122 123 124 125 126 127 128 129 130  
## 14 131 132 133 134 135 136 137 138 139 NA  
## 15 141 142 143 144 145 146 147 148 149 150  
## 16 151 152 153 154 155 156 157 158 159 160  
## 17 161 162 163 164 NA 166 167 168 169 170

# Question 2 (32 points)

## Q2a (8 points)

Compute:

Hint: Use factorial(n) to compute .

sum\_function <- function(n){  
 return(sum((-1)\*\*n / factorial(n)\*\*2))  
 }   
sum\_function(5:20)

## [1] -6.755419e-05

## Q2b (8 points)

Compute:

NOTE: The symbol represents multiplication.

prod\_function <- function(n) {  
 return(prod(4\*n + (1/(2\*\*n))))  
 }  
prod\_function(1:5)

## [1] 144833.6

## Q2c (8 points)

Describe what the following R command does: c(0:5)[NA]

c(0:5)[NA]

## [1] NA NA NA NA NA NA

#This command is asking R to conduct indexing on a vector made of a sequence of numbers between 0 to 5 (0, 1, 2, 3, 4, 5). The square brackets is an indexing tool and is used to extract a value from the vector. Normally, we indicate a number inside of the square brackets that corresponds with the position of the element we would like to extract from the vector.In this case, we are asking R to extract NA. NA represents a missing value. If we try to extract a missing value from the vector, the output for each value of the vector is also displayed as a missing value... This is because the indexing variable, NA, is missing from the vector.

## Q2d (8 points)

Describe the purpose of is.vector(), is.character(), is.numeric(), and is.na() functions? Please use x <- c("a", "b", NA, 2) to explain your description.

x <- c("a", "b", NA, 2)  
  
#The is.vector() function evaluates the object to let us know whether it is a vector and will returns a boolean value of (either True or False). In this example, the value returned is True because x is a single object containing the vector c("a", "b", NA, 2).  
is.vector(x)

## [1] TRUE

#The is.character function evaluates whether an object contains the character data type as either True or False. In this example, the function returned boolean value T because the object x contains the characters "a" and "b" because they are in quotation marks.   
is.character(x)

## [1] TRUE

#The is.numeric function returns the logical value of True when the object in question contains values of the numeric datatype (integers and float numbers). Otherwise the function will return False. In this example, is.numeric(x) returned False because vectors in R must contain values of the same data type. Thus, combining the numeric value of 2 with the character types "a" and "b" in a vector caused R to convert all items in that vector to the character type. Now, when I print x, the number 2 appears in quotation marks indicating that it is now a string. This can also be verified using the class() function as follows:  
is.numeric(x)

## [1] FALSE

x

## [1] "a" "b" NA "2"

class(x)

## [1] "character"

#The is.na() function returns True or False based on the respective position of present or missing values. In this example, the object x contains NA (a missing value). The function returns F, F, T, F because 1st, 2nd, and 4th values of the vector are present, but the 3rd is missing.   
is.na(x)

## [1] FALSE FALSE TRUE FALSE

# Question 3 (36 points)

The airquality dataset contains daily air quality measurements in New York from May to September 1973. The variables include Ozone level, Solar radiation, wind speed, temperature in Fahrenheit, month, and day. Please see the detailed description using help("airquality").

Install the airquality data set on your computer using the command install.packages("datasets"). Then load the datasets package into your session.

library(datasets)

## Q3a (4 points)

Display the first 10 rows of the airquality data set.

data("airquality")  
head(airquality, 10)

## 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  
## 7 23 299 8.6 65 5 7  
## 8 19 99 13.8 59 5 8  
## 9 8 19 20.1 61 5 9  
## 10 NA 194 8.6 69 5 10

## Q3b (8 points)

Compute the average of the first four variables (Ozone, Solar.R, Wind and Temp) for the fifth month using the sapply() function. Hint: You might need to consider removing the NA values; otherwise, the average will not be computed.

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  
## 7 23 299 8.6 65 5 7  
## 8 19 99 13.8 59 5 8  
## 9 8 19 20.1 61 5 9  
## 10 NA 194 8.6 69 5 10  
## 11 7 NA 6.9 74 5 11  
## 12 16 256 9.7 69 5 12  
## 13 11 290 9.2 66 5 13  
## 14 14 274 10.9 68 5 14  
## 15 18 65 13.2 58 5 15  
## 16 14 334 11.5 64 5 16  
## 17 34 307 12.0 66 5 17  
## 18 6 78 18.4 57 5 18  
## 19 30 322 11.5 68 5 19  
## 20 11 44 9.7 62 5 20  
## 21 1 8 9.7 59 5 21  
## 22 11 320 16.6 73 5 22  
## 23 4 25 9.7 61 5 23  
## 24 32 92 12.0 61 5 24  
## 25 NA 66 16.6 57 5 25  
## 26 NA 266 14.9 58 5 26  
## 27 NA NA 8.0 57 5 27  
## 28 23 13 12.0 67 5 28  
## 29 45 252 14.9 81 5 29  
## 30 115 223 5.7 79 5 30  
## 31 37 279 7.4 76 5 31  
## 32 NA 286 8.6 78 6 1  
## 33 NA 287 9.7 74 6 2  
## 34 NA 242 16.1 67 6 3  
## 35 NA 186 9.2 84 6 4  
## 36 NA 220 8.6 85 6 5  
## 37 NA 264 14.3 79 6 6  
## 38 29 127 9.7 82 6 7  
## 39 NA 273 6.9 87 6 8  
## 40 71 291 13.8 90 6 9  
## 41 39 323 11.5 87 6 10  
## 42 NA 259 10.9 93 6 11  
## 43 NA 250 9.2 92 6 12  
## 44 23 148 8.0 82 6 13  
## 45 NA 332 13.8 80 6 14  
## 46 NA 322 11.5 79 6 15  
## 47 21 191 14.9 77 6 16  
## 48 37 284 20.7 72 6 17  
## 49 20 37 9.2 65 6 18  
## 50 12 120 11.5 73 6 19  
## 51 13 137 10.3 76 6 20  
## 52 NA 150 6.3 77 6 21  
## 53 NA 59 1.7 76 6 22  
## 54 NA 91 4.6 76 6 23  
## 55 NA 250 6.3 76 6 24  
## 56 NA 135 8.0 75 6 25  
## 57 NA 127 8.0 78 6 26  
## 58 NA 47 10.3 73 6 27  
## 59 NA 98 11.5 80 6 28  
## 60 NA 31 14.9 77 6 29  
## 61 NA 138 8.0 83 6 30  
## 62 135 269 4.1 84 7 1  
## 63 49 248 9.2 85 7 2  
## 64 32 236 9.2 81 7 3  
## 65 NA 101 10.9 84 7 4  
## 66 64 175 4.6 83 7 5  
## 67 40 314 10.9 83 7 6  
## 68 77 276 5.1 88 7 7  
## 69 97 267 6.3 92 7 8  
## 70 97 272 5.7 92 7 9  
## 71 85 175 7.4 89 7 10  
## 72 NA 139 8.6 82 7 11  
## 73 10 264 14.3 73 7 12  
## 74 27 175 14.9 81 7 13  
## 75 NA 291 14.9 91 7 14  
## 76 7 48 14.3 80 7 15  
## 77 48 260 6.9 81 7 16  
## 78 35 274 10.3 82 7 17  
## 79 61 285 6.3 84 7 18  
## 80 79 187 5.1 87 7 19  
## 81 63 220 11.5 85 7 20  
## 82 16 7 6.9 74 7 21  
## 83 NA 258 9.7 81 7 22  
## 84 NA 295 11.5 82 7 23  
## 85 80 294 8.6 86 7 24  
## 86 108 223 8.0 85 7 25  
## 87 20 81 8.6 82 7 26  
## 88 52 82 12.0 86 7 27  
## 89 82 213 7.4 88 7 28  
## 90 50 275 7.4 86 7 29  
## 91 64 253 7.4 83 7 30  
## 92 59 254 9.2 81 7 31  
## 93 39 83 6.9 81 8 1  
## 94 9 24 13.8 81 8 2  
## 95 16 77 7.4 82 8 3  
## 96 78 NA 6.9 86 8 4  
## 97 35 NA 7.4 85 8 5  
## 98 66 NA 4.6 87 8 6  
## 99 122 255 4.0 89 8 7  
## 100 89 229 10.3 90 8 8  
## 101 110 207 8.0 90 8 9  
## 102 NA 222 8.6 92 8 10  
## 103 NA 137 11.5 86 8 11  
## 104 44 192 11.5 86 8 12  
## 105 28 273 11.5 82 8 13  
## 106 65 157 9.7 80 8 14  
## 107 NA 64 11.5 79 8 15  
## 108 22 71 10.3 77 8 16  
## 109 59 51 6.3 79 8 17  
## 110 23 115 7.4 76 8 18  
## 111 31 244 10.9 78 8 19  
## 112 44 190 10.3 78 8 20  
## 113 21 259 15.5 77 8 21  
## 114 9 36 14.3 72 8 22  
## 115 NA 255 12.6 75 8 23  
## 116 45 212 9.7 79 8 24  
## 117 168 238 3.4 81 8 25  
## 118 73 215 8.0 86 8 26  
## 119 NA 153 5.7 88 8 27  
## 120 76 203 9.7 97 8 28  
## 121 118 225 2.3 94 8 29  
## 122 84 237 6.3 96 8 30  
## 123 85 188 6.3 94 8 31  
## 124 96 167 6.9 91 9 1  
## 125 78 197 5.1 92 9 2  
## 126 73 183 2.8 93 9 3  
## 127 91 189 4.6 93 9 4  
## 128 47 95 7.4 87 9 5  
## 129 32 92 15.5 84 9 6  
## 130 20 252 10.9 80 9 7  
## 131 23 220 10.3 78 9 8  
## 132 21 230 10.9 75 9 9  
## 133 24 259 9.7 73 9 10  
## 134 44 236 14.9 81 9 11  
## 135 21 259 15.5 76 9 12  
## 136 28 238 6.3 77 9 13  
## 137 9 24 10.9 71 9 14  
## 138 13 112 11.5 71 9 15  
## 139 46 237 6.9 78 9 16  
## 140 18 224 13.8 67 9 17  
## 141 13 27 10.3 76 9 18  
## 142 24 238 10.3 68 9 19  
## 143 16 201 8.0 82 9 20  
## 144 13 238 12.6 64 9 21  
## 145 23 14 9.2 71 9 22  
## 146 36 139 10.3 81 9 23  
## 147 7 49 10.3 69 9 24  
## 148 14 20 16.6 63 9 25  
## 149 30 193 6.9 70 9 26  
## 150 NA 145 13.2 77 9 27  
## 151 14 191 14.3 75 9 28  
## 152 18 131 8.0 76 9 29  
## 153 20 223 11.5 68 9 30

airquality\_m5<-subset((airquality), Month==5)  
airquality\_m5

## 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  
## 7 23 299 8.6 65 5 7  
## 8 19 99 13.8 59 5 8  
## 9 8 19 20.1 61 5 9  
## 10 NA 194 8.6 69 5 10  
## 11 7 NA 6.9 74 5 11  
## 12 16 256 9.7 69 5 12  
## 13 11 290 9.2 66 5 13  
## 14 14 274 10.9 68 5 14  
## 15 18 65 13.2 58 5 15  
## 16 14 334 11.5 64 5 16  
## 17 34 307 12.0 66 5 17  
## 18 6 78 18.4 57 5 18  
## 19 30 322 11.5 68 5 19  
## 20 11 44 9.7 62 5 20  
## 21 1 8 9.7 59 5 21  
## 22 11 320 16.6 73 5 22  
## 23 4 25 9.7 61 5 23  
## 24 32 92 12.0 61 5 24  
## 25 NA 66 16.6 57 5 25  
## 26 NA 266 14.9 58 5 26  
## 27 NA NA 8.0 57 5 27  
## 28 23 13 12.0 67 5 28  
## 29 45 252 14.9 81 5 29  
## 30 115 223 5.7 79 5 30  
## 31 37 279 7.4 76 5 31

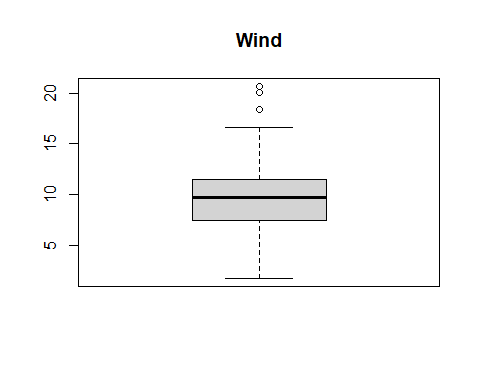
sapply(airquality\_m5[,-5:-6], mean, na.rm = TRUE)

## Ozone Solar.R Wind Temp   
## 23.61538 181.29630 11.62258 65.54839

## Q3c (8 points)

Construct a boxplot for the all Wind and Temp variables, then display the values of all the outliers which lie beyond the whiskers.

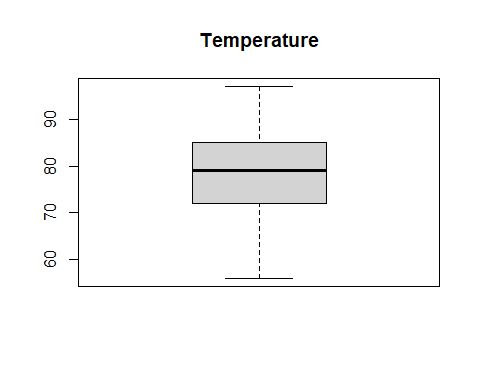
#Wind boxplot followed by outliers  
  
wind\_plot<-boxplot(airquality$Wind, main = "Wind")



wind\_plot$out

## [1] 20.1 18.4 20.7

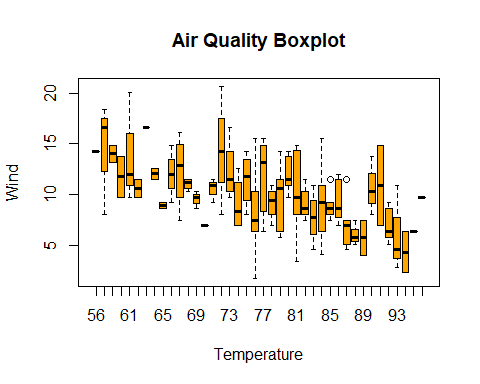
#Temp boxplot followed by outliers   
temp\_plot<-boxplot(airquality$Temp, main = "Temperature")



temp\_plot$out

## numeric(0)

#Wind / Temp boxplot followed by outliers  
wind\_temp\_plot<-boxplot(formula = Wind ~ Temp, data = airquality, main = "Air Quality Boxplot", xlab= "Temperature", ylab= "Wind", col="orange")



wind\_temp\_plot$out

## [1] 11.5 11.5

## Q3d (8 points)

Compute the upper quartile of the Wind variable with two different methods. HINT: Only show the upper quartile using indexing. For the type of quartile, please see <https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/quantile>.

#Method 1  
quantile(airquality$Wind, probs = c(0.75))

## 75%   
## 11.5

#Method 2  
wind\_2<-cbind(summary(airquality$Wind))  
wind\_2[5,]

## 3rd Qu.   
## 11.5

#Method 3  
wind\_1<-data.frame(quantile(airquality$Wind))  
wind\_1[4,]

## [1] 11.5

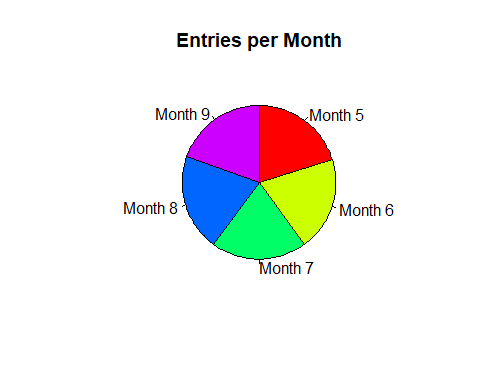
## Q3e (8 points)

Construct a pie chart to describe the number of entries by Month. HINT: use the table() function to count and tabulate the number of entries within a Month.

#Tabulating number of entries in a month  
month\_entries<-table(airquality$Month, useNA="no")  
month\_entries

##   
## 5 6 7 8 9   
## 31 30 31 31 30

#Create labels for pie chart  
lbls <- c("Month 5", "Month 6", "Month 7", "Month 8", "Month 9")  
  
#Create pie chart  
pie(month\_entries, labels = lbls, edges = 200, radius = 0.8, clockwise = TRUE, col=rainbow(length(month\_entries)), main = "Entries per Month")



END of Assignment #1.