# CIND 123 - Data Analytics: Basic Methods

Qian (Jessie) Ma

# Assignment 1 (10%)

## [Qian (Jessie) Ma]

## [CIND 123 Section D40 & student number: 501274167]

---

# 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 (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 $(3, 5 \ldots, 29)$.

```
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()

```
# Insert your answer here.
x<-c(4:115)
y<-c(seq(from=4,to=115,by=4))
diff<-length(x)-length(y)
```

```
# Print x
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
```

```
# Print y
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
```

```
# Print the difference in lengths between x and y
diff
```

```
## [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` .

```
# Insert your answer here.
indices_vector<-c(1,3,7,12,17,20,22,24)
y_square<-c(y[indices_vector]^2)
y_square
```

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

```
# Print mean
mean(y_square[1:5])
```

```
## [1] 1574.4
```

```
# Print median
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.

```
# Insert your answer here.
# It would not be correct to use the command `as.numeric(factorVar)` to convert factor to number
because if the factor is numeric, we need to first convert it to a character vector then to a nu
meric. If the factor is a character then we just need to convert it to numeric. The correct comm
and would be `as.numeric(as.character(factorVar))`. Please see example below.
factorVar<-factor(c(1,6,5.4,3.2))
numeric_factorVar<-as.numeric(as.character(factorVar))
numeric_factorVar
```

```
## [1] 1.0 6.0 5.4 3.2
```

```
# View class (the class is numeric)
class(numeric_factorVar)
```

```
## [1] "numeric"
```

```
# If we convert it directly to numeric from factor the following result is produced. As you can
see the vector elements have changed, R assigned the numbers 1 4 3 2 to the categories of the fa
ctor.
failed_factorVar<-as.numeric(factorVar)
failed_factorVar
```

```
## [1] 1 4 3 2
```

```
# View class (although the class is numeric, the contents of the vector were not retained)
class(failed_factorVar)
```

```
## [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.

```
# Insert your answer here.
# In order to read a csv file that contains null values and question marks, I must set the na.st
rings option to c("null","?") to replace any nulls and ?s with the logical value NA as shown bel
ow.
dataset<-read.csv("dataset.csv",
                  header=T,
                  na.strings=c("null","?"))
dataset
```

```
##     X1  X2  X3  X4  X5  X6  X7  X8  X9 X10
## 1   11  12  13  14  15  16  17  18  19  20
## 2   21  22  23  24  25  26  27  28  29  30
## 3   31  32  33  34  35  36  37  38  39  40
## 4   41  42  43  44  45  NA  47  48  49  50
## 5   51  52  53  NA  55  56  57  NA  59  60
## 6   61  62  63  64  65  66  67  68  69  70
## 7   71  72  NA  74  75  76  77  78  79  80
## 8   81  82  83  84  85  86  87  88  89  NA
## 9   91  92  93  94  95  96  97  98  99 100
## 10  NA 102 103 104 105 106 107 108 109 110
## 11 111 112 113 114 115 116 117 118 119 120
## 12 121 122 123 124 125 126 127 128 129 130
## 13 131 132 133 134 135 136 137 138 139  NA
## 14 141 142 143 144 145 146 147 148 149 150
## 15 151 152 153 154 155 156 157 158 159 160
## 16 161 162 163 164  NA 166 167 168 169 170
```

# Question 2 (32 points)

# Q2a (8 points)

Compute:

$$\sum_{n=5}^{20} \frac{(-1)^n}{(n!)^2}$$

Hint: Use `factorial(n)` to compute $n!$.

```
# Insert your answer here.
n=5:20
sum((-1)^n/(factorial(n))^2)
```

```
## [1] -6.755419e-05
```

# Q2b (8 points)

Compute:

$$\prod_{n=1}^{5} \left(4n + \frac{1}{2^n}\right)$$

NOTE: The symbol $\Pi$ represents multiplication.

```
# Insert your answer here.
n=1:5
prod((4*n)+(1/2^n))
```

```
## [1] 144833.6
```

# Q2c (8 points)

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

```
# Insert your answer here.
# NA is of type logical. The logical type has a property whereby it recycles vector elements. Th
is R command can be interpreted as c(0:5)[c(NA,NA,NA,NA,NA,NA)] due to the logical indices being
recycled. c(0:5) creates the vector 0 1 2 3 4 5. This subset operation returns the elements of t
he vector with the value of NA, and in this case there are none which is why NA appears for each
of the 6 indices. Each NA returned corresponds to each element from the vector 0 to 5 during sub
setting.
c(0:5)[NA]
```

```
## [1] NA NA NA NA NA NA
```

# 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.

```
# Insert your answer here.
# The purpose of the `is.vector()`, `is.character()`, `is.numeric()`, and `is.na()` functions is to serve as a test that returns a logical value, either TRUE or FALSE, indicating whether the argument passed to it satisfies what the function is testing for.

# For instance `is.vector()` tests whether the object is a vector or not, returning TRUE if it is, FALSE if it is not. In the example below we see is.vector(x) returns TRUE indicating x is a vector.
x<-c("a", "b", NA, 2)
is.vector(x)
```

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

```
# `is.character()` tests whether the object passed to it as an argument is of character type. x is a character as shown in the example below.
is.character(x)
```

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

```
# `is.numeric()` tests whether the object is of numeric type. x is not numeric as shown in the example below.
is.numeric(x)
```

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

```
# `is.na()` tests whether an object is an NA value. x is NA at the 3rd index only as shown below.
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.

```
# Insert your answer here.
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.

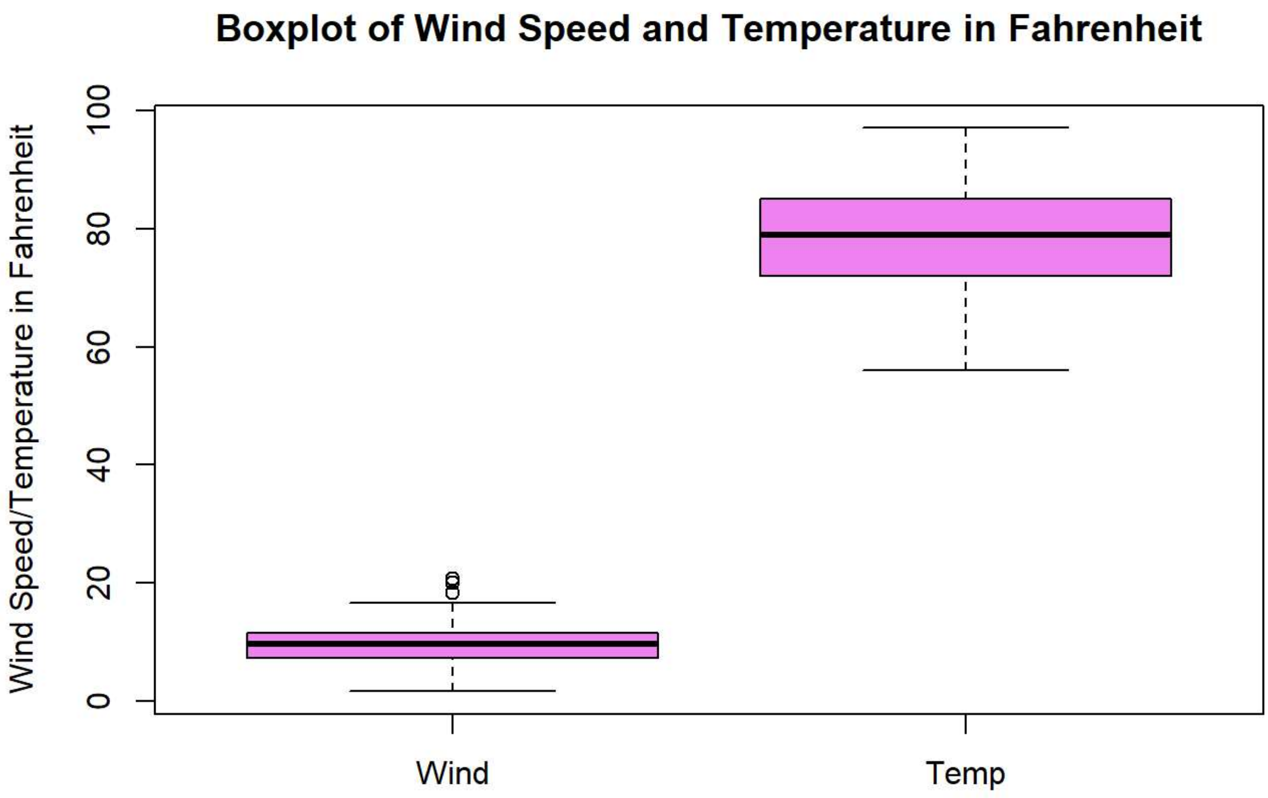
```
# Insert your answer here.
ds<-airquality
sapply((subset(ds,Month==5,select=1:4)),mean,na.rm=TRUE)
```

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

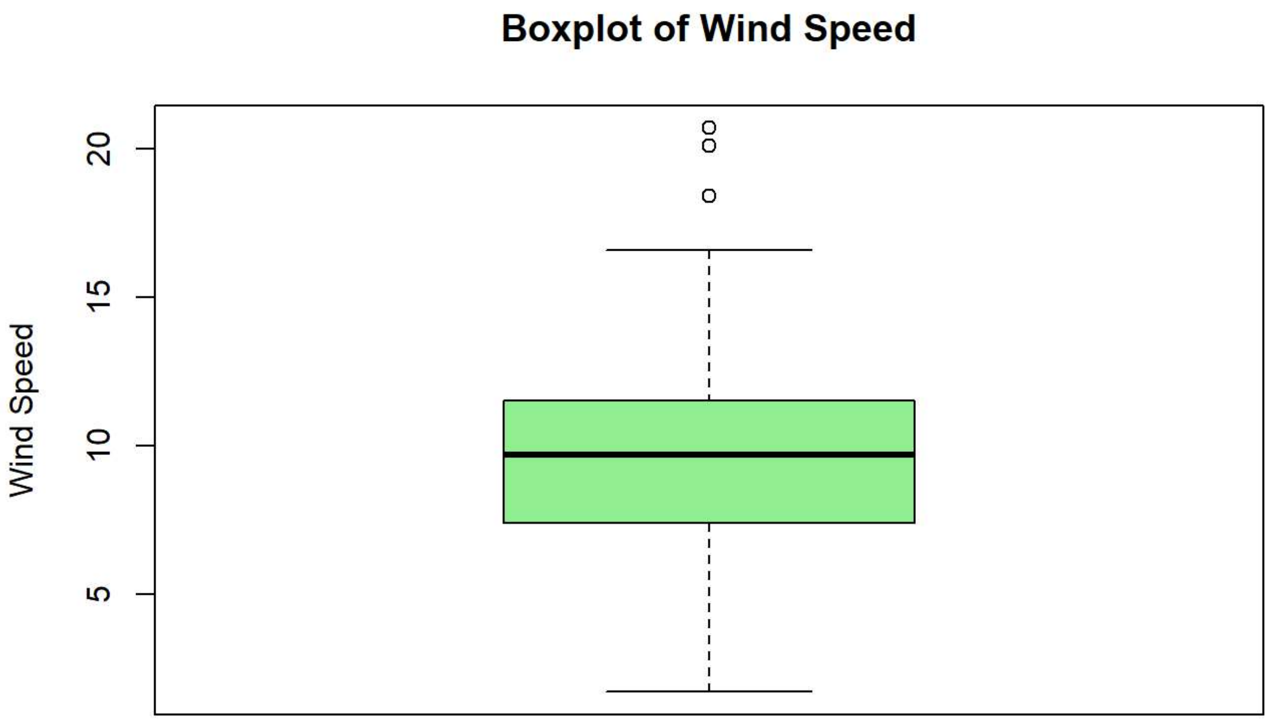
# Q3c (8 points)

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

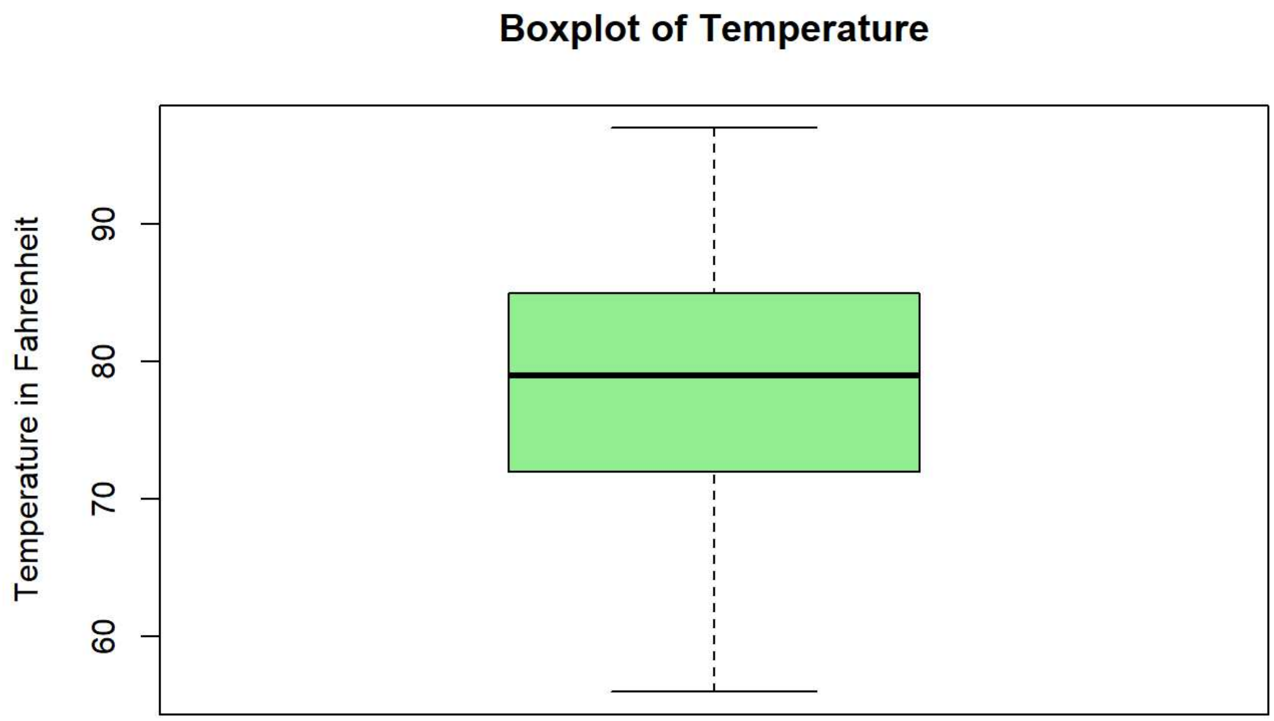
```
# Insert your answer here.
# Combined boxplot
boxplot(ds[3:4],na.rm=TRUE,main="Boxplot of Wind Speed and Temperature in Fahrenheit",ylab="Wind
Speed/Temperature in Fahrenheit",col="violet")
```



```
# Wind boxplot
boxplot(ds$Wind,na.rm=TRUE,main="Boxplot of Wind Speed",ylab="Wind Speed",col="lightgreen")
```



```
# Temperature boxplot
boxplot(ds$Temp,na.rm=TRUE,main="Boxplot of Temperature",ylab="Temperature in Fahrenheit",col="l
ightgreen")
```



```
# Before computing outliers, check summary first
summary(ds[3:4])
```

```
##       Wind             Temp      
##  Min.   : 1.700   Min.   :56.00  
##  1st Qu.: 7.400   1st Qu.:72.00  
##  Median : 9.700   Median :79.00  
##  Mean   : 9.958   Mean   :77.88  
##  3rd Qu.:11.500   3rd Qu.:85.00  
##  Max.   :20.700   Max.   :97.00  
```

```
# Get IQR and lower and upper fences for Wind and Temp variables
IQR(ds$Wind)
```

```
## [1] 4.1
```

```
IQR(ds$Temp)
```

```
## [1] 13
```

```
wind_lowerfence=7.4-(1.5*4.1)
wind_lowerfence
```

```
## [1] 1.25
```

```
wind_upperfence=11.5+(1.5*4.1)
wind_upperfence
```

```
## [1] 17.65
```

```
temp_lowerfence=72-(1.5*13)
temp_lowerfence
```

```
## [1] 52.5
```

```
temp_upperfence=85+(1.5*13)
temp_upperfence
```

```
## [1] 104.5
```

```
# Identify and display outliers for Wind variable
ds$Wind[which(ds$Wind<wind_lowerfence|ds$Wind>wind_upperfence)]
```

```
## [1] 20.1 18.4 20.7
```

```
# Identify and display outliers for Temp variable
ds$Temp[which(ds$Temp<temp_lowerfence|ds$Temp>temp_upperfence)]
```

```
## integer(0)
```

```
# The output shows the Wind variable has the outliers 20.1, 18.4 and 20.7 while the Temp variabl
e does not have any outliers.
```

# 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 (https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/quantile).

```
# Insert your answer here.
# Upper quartile computation using method 1 (result is 11.75):
wind_upperquartile1<-quantile(ds[,3],probs=0.75,type=6)
wind_upperquartile1
```

```
##   75%
## 11.75
```

```
# Upper quartile computation using method 2 (result is also 11.75):
wind_sorted<-sort(ds$Wind)
wind_sorted
```

```
##   [1]  1.7  2.3  2.8  3.4  4.0  4.1  4.6  4.6  4.6  4.6  5.1  5.1  5.1  5.7  5.7
##  [16]  5.7  6.3  6.3  6.3  6.3  6.3  6.3  6.3  6.3  6.9  6.9  6.9  6.9  6.9  6.9
##  [31]  6.9  6.9  6.9  7.4  7.4  7.4  7.4  7.4  7.4  7.4  7.4  7.4  7.4  8.0  8.0
##  [46]  8.0  8.0  8.0  8.0  8.0  8.0  8.0  8.0  8.0  8.6  8.6  8.6  8.6  8.6  8.6
##  [61]  8.6  8.6  9.2  9.2  9.2  9.2  9.2  9.2  9.2  9.2  9.7  9.7  9.7  9.7  9.7
##  [76]  9.7  9.7  9.7  9.7  9.7  9.7 10.3 10.3 10.3 10.3 10.3 10.3 10.3 10.3 10.3
##  [91] 10.3 10.3 10.9 10.9 10.9 10.9 10.9 10.9 10.9 10.9 11.5 11.5 11.5 11.5 11.5
## [106] 11.5 11.5 11.5 11.5 11.5 11.5 11.5 11.5 11.5 11.5 12.0 12.0 12.0 12.0 12.6
## [121] 12.6 12.6 13.2 13.2 13.8 13.8 13.8 13.8 13.8 14.3 14.3 14.3 14.3 14.3 14.3
## [136] 14.9 14.9 14.9 14.9 14.9 14.9 14.9 14.9 15.5 15.5 15.5 16.1 16.6 16.6 16.6
## [151] 18.4 20.1 20.7
```

```
length(wind_sorted)
```

```
## [1] 153
```

```
wind_upperquartile2_position<-0.75*(153+1)
wind_upperquartile2_position
```

```
## [1] 115.5
```

```
wind_upperquartile2_a<-wind_sorted[115]
wind_upperquartile2_a
```

```
## [1] 11.5
```

```
wind_upperquartile2_b<-wind_sorted[116]
wind_upperquartile2_b
```

```
## [1] 12
```

```
wind_upperquartile2<-11.5+0.50*(12-11.5)
wind_upperquartile2
```

```
## [1] 11.75
```

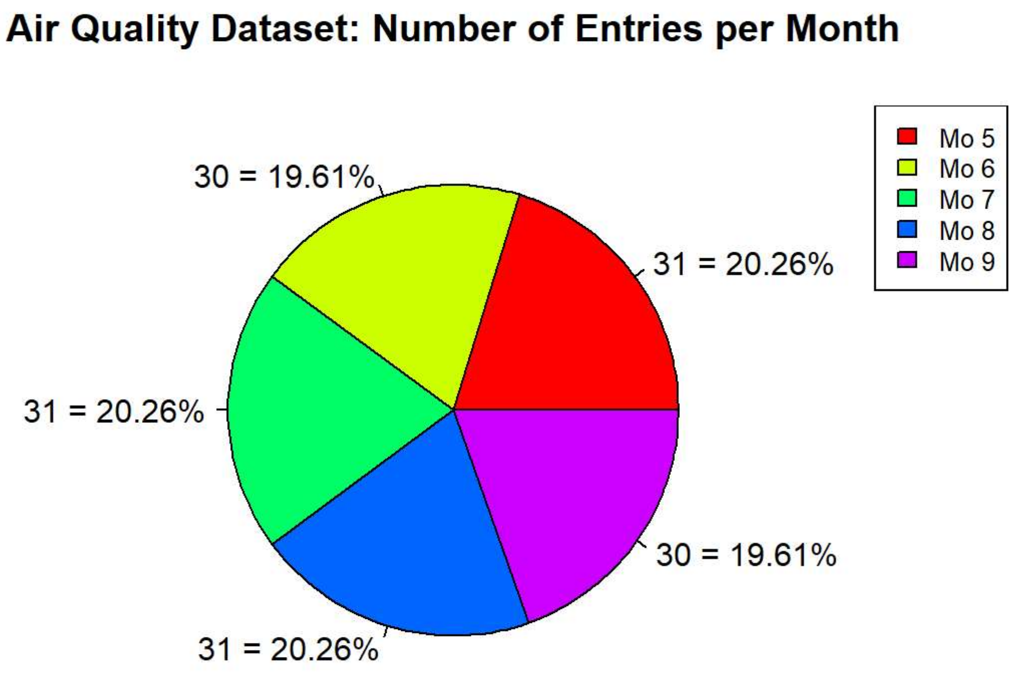
# Q3e (8 points)

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` .

```
# Insert your answer here.
# First, tabulate the entries
table(airquality$Month)
```

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

```
# Next, construct the pie chart
slices<-c(31,30,31,31,30)
pie_labels<-paste0(slices," = ",round(100*slices/sum(slices),2),"%")
pie(slices,labels=pie_labels,main="Air Quality Dataset: Number of Entries per Month",col=rainbow(length(slices)))
legend("topright",legend=c("Mo 5", "Mo 6","Mo 7","Mo 8","Mo 9"),cex=0.8,fill=rainbow(length(slices)))
```



END of Assignment #1.