CleaningDataInR

Exercise 1: Cleaning flat files

1. Open up Rstudio program

2. Open up ex1.r so that you can highlight and run the codes :)
3. Read in the csv file and assign the it to the df variable
    - ***df <- read.csv("C:\\Users\\benjit\\Google Drive\\CleaningDataWithR\\ex1\\amkresalehousingdata.csv",stringsAsFactors = FALSE)***
    - We will not be exploring the factors data type

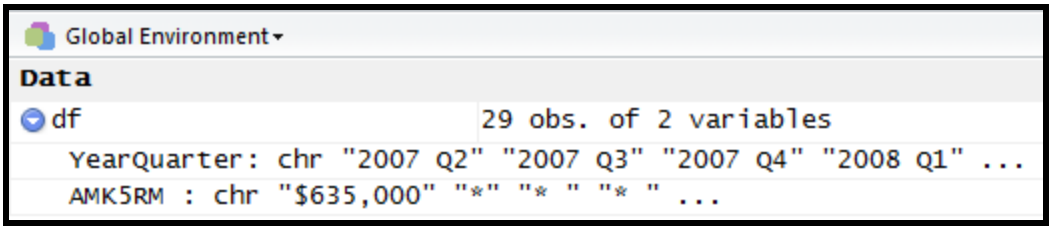
4. Lets use the class function to see what object df is
    - ***class(df)***
    - The variable df is a data frame of consisting of the data from the csv file
    - To view the data you can enter ***df*** or use ***head(df)*** for large datasets in the console

```
> head(df)
  YearQuarter   AMK5RM
1     2007 Q2 $635,000
2     2007 Q3        *
3     2007 Q4        *
4     2008 Q1        *
5     2008 Q2 $667,500
6     2008 Q3 $649,000
```

5. Lets call the summary function of df
    - ***summary(df)***

```
> summary(df)
 YearQuarter           AMK5RM
 Length:29          Length:29
 Class :character   Class :character
 Mode  :character   Mode  :character
```

    - Calling the summary function of the data frame allows us to see descriptive statistic of the variable involved
6. Looks like our prices are being read as characters in the data frame
    - If prices are read as numeric, we usually will see some description statistic of the numbers
        - e.g mean, median, quartiles
    - Let us confirm that that data type is actually a character
        - ***str(df)***
        - The Environment tab of Rstudio also shows the structure of the variable df



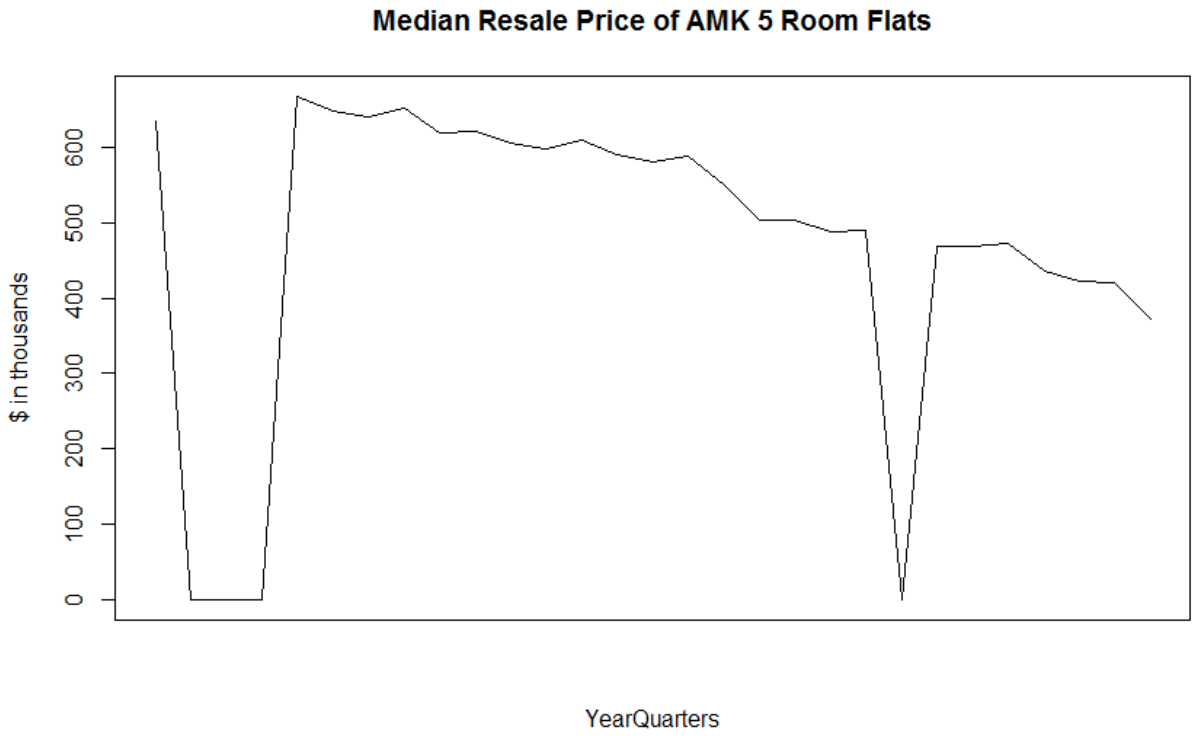
To convert this column to numeric we need to

- Remove $ and ,
- Replace * with 0
  - based on the data dictionary * refers to cases where there are less than 20 resale transactions in the quarter for the particular town and flat type. The median prices of these cases are not shown as they may not be representative.
  - We could do some variable imputation but for now we will just replace it with 0

7. Lets us take a look at gsub() to perform character manipulation
   - ***?gsub*** if you want to read the description and more examples
   - ***df$AMK5RM <- gsub("\\$","",df$AMK5RM)***
     - You need to escape the $ with \\
       - $ is actually used in regular expression
     - The above code will replace $ with blank and assign the modified column back to the AMK5RM data frame column
   - Lets replace the rest intineratively
     - ***df$AMK5RM <- gsub(" ","",df$AMK5RM)***
     - ***df$AMK5RM <- gsub(",","",df$AMK5RM)***
     - ***df$AMK5RM <- gsub("*","0",df$AMK5RM)***
       - * is also a regular expression
8. Now use the summary function, you will see that the dataset is still a character, we will now convert this row to numeric
   - ***df$AMK5RM <- as.numeric(df$AMK5RM)***
9. Now use the summary function and you will see the difference
   - Convert to numeric
     - ***df$AMK5RM <- as.numeric(df$AMK5RM)***

```
> summary(df)
 YearQuarter           AMK5RM
 Length:29          Length:29
 Class :character   Class :character
 Mode  :character   Mode  :character
```

10. Now you can display a chart or run your statistic algorithm on your dataset
    - ***plot(df$AMK5RM/1000, main="Median Resale Price of AMK 5 Room Flats",***
      ,xlab="YearQuarters",ylab="$ in thousands",type="lines",xaxt='n')


Median Resale Price of AMK 5 Room Flats
$ in thousands
0
100
200
300
400
500
600
YearQuarters