MSDS680 Week 1 - R Warm Up

Benjamin Siebold

Aug, 31, 2020

Introduction

In this project, a few R functions will be applied to R datasets with the intention of getting "warmed" up in R prior to building ML models in the following weeks, basic R functions for data exploration will be inspected, and the dataExplorer package will be tested to see what it has to offer. Throughout this R file, basic data investigation and cleaning will be completed that is also necessary when cleaning data for ML models Before applying any functions or solving any data issues, the number of rows of data and a brief summary of the data is provided.

```
nrow(airquality)
```

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

```
summary(airquality)
```

```
##      Ozone           Solar.R           Wind             Temp
##  Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00
##  1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00
##  Median : 31.50   Median :205.0   Median : 9.700   Median :79.00
##  Mean   : 42.13   Mean   :185.9   Mean   : 9.958   Mean   :77.88
##  3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00
##  Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00
##  NA's   :37       NA's   :7
##      Month            Day
##  Min.   :5.000   Min.   : 1.0
##  1st Qu.:6.000   1st Qu.: 8.0
##  Median :7.000   Median :16.0
##  Mean   :6.993   Mean   :15.8
##  3rd Qu.:8.000   3rd Qu.:23.0
##  Max.   :9.000   Max.   :31.0
##
```

From above, it can be seen that both Ozone, and Solar.R have missing values, and will need to be cleaned.

Find and Remove NULLS

One of the first steps in data cleaning is finding missing values, and if the missing data appears to be problematic, remove those rows from the dataset. The R base dataset "airquality" has missing values and thus is a good candidate for this first exercise.

```
nrow(airquality[!complete.cases(airquality), ])
```

```
## [1] 42
```

```
airquality_no_NA = na.omit(airquality)
nrow(airquality_no_NA)
```

```
## [1] 111
```

```
summary(airquality_no_NA)
```

```
##      Ozone           Solar.R           Wind             Temp
##  Min.   :  1.0   Min.   :  7.0   Min.   : 2.30   Min.   :57.00
##  1st Qu.: 18.0   1st Qu.:113.5   1st Qu.: 7.40   1st Qu.:71.00
##  Median : 31.0   Median :207.0   Median : 9.70   Median :79.00
##  Mean   : 42.1   Mean   :184.8   Mean   : 9.94   Mean   :77.79
##  3rd Qu.: 62.0   3rd Qu.:255.5   3rd Qu.:11.50   3rd Qu.:84.50
##  Max.   :168.0   Max.   :334.0   Max.   :20.70   Max.   :97.00
##      Month            Day
##  Min.   :5.000   Min.   : 1.00
##  1st Qu.:6.000   1st Qu.: 9.00
##  Median :7.000   Median :16.00
##  Mean   :7.216   Mean   :15.95
##  3rd Qu.:9.000   3rd Qu.:22.50
##  Max.   :9.000   Max.   :31.00
```

Above multiple functions can be seen that accomplish the goal of finding NULLS and removing them. The nrow function allows for a total count of dataset rows to be taken. By performing the function on all three datasets it can be seen all the rows with NULLS were removed (153-42 = 111). The second function (summary) provides some basic descriptive statistics of the cleaned dataset. The most important function used above is the complete.cases() function, which provides a a boolean matrix of TRUES and FALSES if a row has a NULL value or not. By combining this with an outer dataframe and NOT call, all rows with a NULL value can be returned or counted.

Imput missing values with means or medians

Aside from removing all NULL rows in a dataset, another option is to replace missing cells or NULL values with median or mean values of that column. This type of cleaning is necessary because dropping rows may be more problematic if there are many factors, removing rows that have values for those factors will remove necessary data.

```
air_means = airquality
air_means$Ozone[is.na(air_means$Ozone)] = round(mean(air_means$Ozone, na.rm=TRUE), 1)
air_means$Solar.R[is.na(air_means$Solar.R)] = round(mean(air_means$Solar.R, na.rm=TRUE), 1)
summary(air_means)
```

```
##      Ozone           Solar.R           Wind             Temp
##  Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00
##  1st Qu.: 21.00   1st Qu.:120.0   1st Qu.: 7.400   1st Qu.:72.00
##  Median : 42.10   Median :194.0   Median : 9.700   Median :79.00
##  Mean   : 42.12   Mean   :185.9   Mean   : 9.958   Mean   :77.88
##  3rd Qu.: 46.00   3rd Qu.:256.0   3rd Qu.:11.500   3rd Qu.:85.00
##  Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00
##      Month            Day
##  Min.   :5.000   Min.   : 1.0
##  1st Qu.:6.000   1st Qu.: 8.0
##  Median :7.000   Median :16.0
##  Mean   :6.993   Mean   :15.8
##  3rd Qu.:8.000   3rd Qu.:23.0
```

```
##  Max.   :9.000   Max.   :31.0
```

From the initial dataset summary, compared to the imputed means dataset, it can be seen the median of Ozone column has increased and the median of the Solar.R column has decreased. This is because the mean values that were imputed were higher and lower respectively. This will need to be considered if using models like kmeans/kmedians because it will impact how the clusters are made. The functions used in the above steps are round and mean, which allow a full column to be rounded to a certain placement, and mean which fins the mean of a column in the dataset.

Scale or normalize the data

Another useful preparation to make to data is to perform data scaling. This is useful because if multiple columns or variables are being used with different scales, they can be normalized to give equal weights.

```
summary(mtcars)
```

```
##       mpg             cyl             disp             hp
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0
##       drat             wt             qsec             vs
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000
##        am              gear            carb
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000
##  Median :0.0000   Median :4.000   Median :2.000
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000
```

```
scaled_cars = mtcars
scaled_cars$mpg = scale(scaled_cars$mpg)
scaled_cars$disp = scale(scaled_cars$disp)
scaled_cars$hp = scale(scaled_cars$hp)
summary(scaled_cars)
```

```
##        mpg.V1             cyl             disp.V1               hp.V1
##  Min.   :-1.6078826   Min.   :4.000   Min.   :-1.2879099   Min.   :-1.3810318
##  1st Qu.:-0.7741273   1st Qu.:4.000   1st Qu.:-0.8867035   1st Qu.:-0.7319924
##  Median :-0.1477738   Median :6.000   Median :-0.2777331   Median :-0.3454858
##  Mean   : 0.0000000   Mean   :6.188   Mean   : 0.0000000   Mean   : 0.0000000
##  3rd Qu.: 0.4495434   3rd Qu.:8.000   3rd Qu.: 0.7687521   3rd Qu.: 0.4858679
##  Max.   : 2.2912716   Max.   :8.000   Max.   : 1.9467538   Max.   : 2.7465668
##       drat             wt             qsec             vs
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000
```

```
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000
##        am              gear            carb
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000
##  Median :0.0000   Median :4.000   Median :2.000
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000
```

Above, the scale functino is used in R to scale the mpg, disp, and hp columns of the mtcars dataset. The scale function finds the difference between each cell and the columns mean, and the divides it by the columns standard deviation to normalize the data with a new mean of 0.

Create dummy variables

The last beneficial data cleaning technique in this notebook is to change categorical data into dummy variables. This will affect the dataframe based off how many values are in the columns converting to dummy variables

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
dummy_vars = dummyVars("~.", data=iris, fullRank=T)
dummy_iris = data.frame(predict(dummy_vars, iris))
head(dummy_iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species.versicolor
## 1          5.1         3.5          1.4         0.2                  0
## 2          4.9         3.0          1.4         0.2                  0
## 3          4.7         3.2          1.3         0.2                  0
## 4          4.6         3.1          1.5         0.2                  0
## 5          5.0         3.6          1.4         0.2                  0
## 6          5.4         3.9          1.7         0.4                  0
##   Species.virginica
## 1                 0
## 2                 0
## 3                 0
## 4                 0
## 5                 0
## 6                 0
```

From above the dummyVars() function from the caret library can be seen. This function maps categorical columns into boolean numbers (0,1) to allow for certain ML models to be used. If a categorial column with

more than two values exists, a column for each category is mapped, then, a dataframe can be created based off these dummy variables as shown above.

Data Exploration in R

Now that data cleaning has been completed in R, some basic data exploration functions will be looked at to show how data can be understood efficiently in R. Aside from the head, tail, and summary, additional functions such as length, or count from the dplyr library can be used to get a count occurance of each value in a column. Below these functions can be seen.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

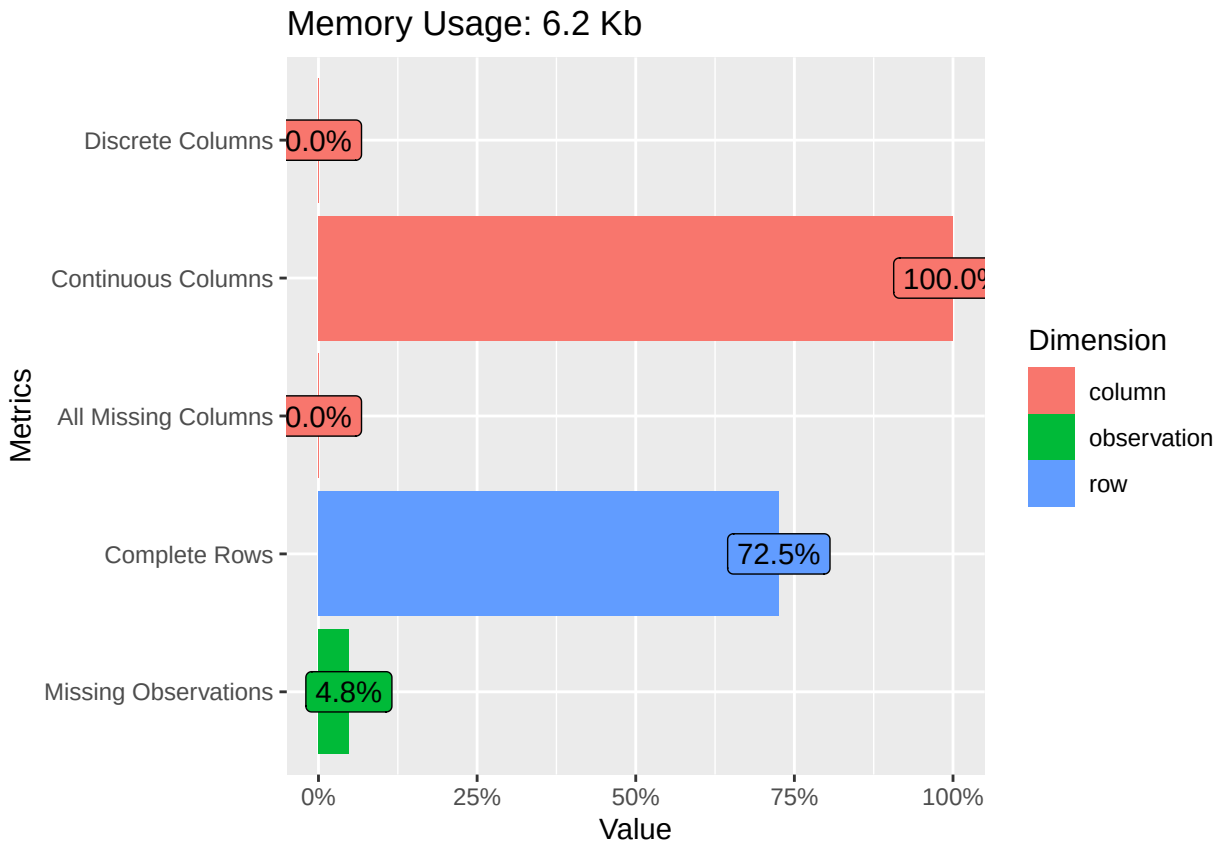
```
library(DataExplorer)
length(iris)
```

```
## [1] 5
```

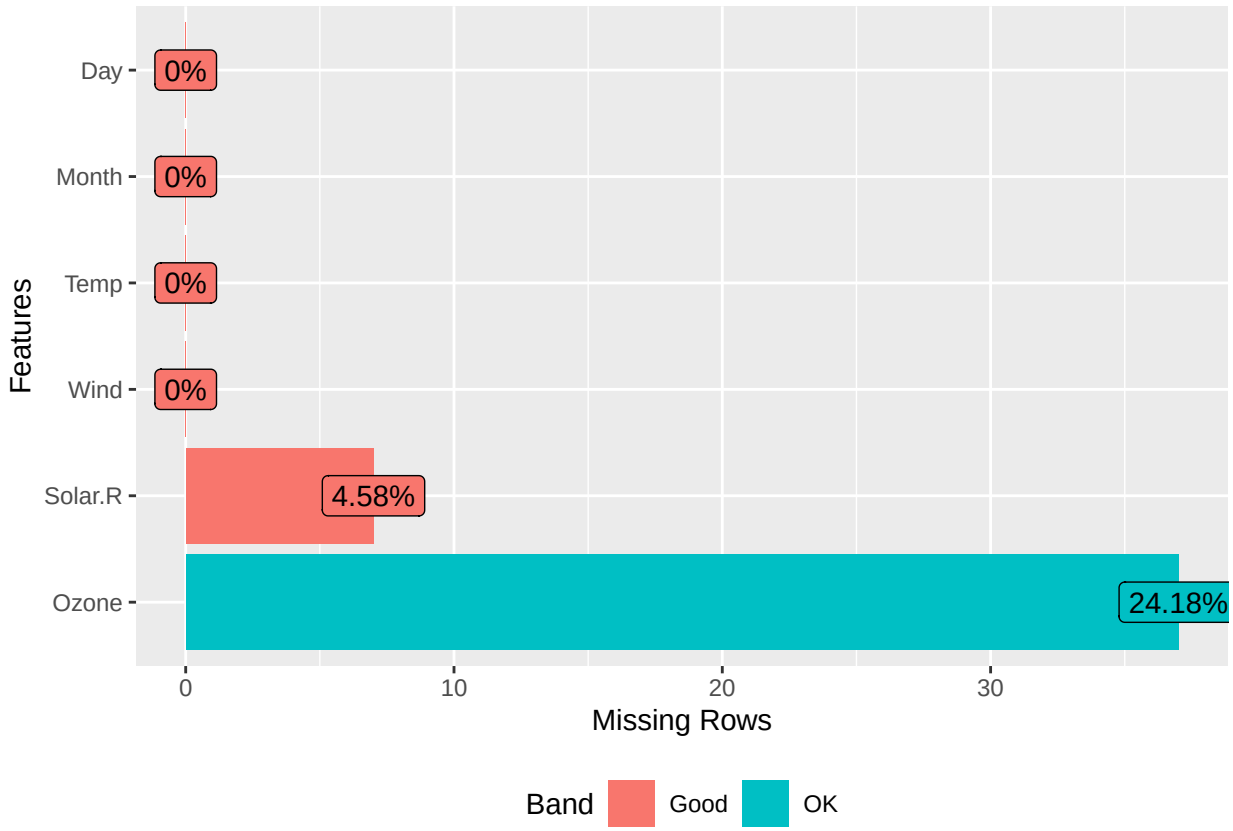
```
iris %>% count(Species)
```

```
##      Species  n
## 1     setosa 50
## 2 versicolor 50
## 3  virginica 50
```

```
plot_intro(airquality)
```




```
plot_missing(airquality)
```




```
plot_correlation(na.omit(airquality))
```

Inspect DataExploer

The last part of the R warm up is to make use of the DataExplorer package in R. After working through some guides for DataExplorer, the quick visualization options for getting an idea of what data is useful seems really convenient. For example, both the plot_intro and plot_missing functions allow for quick inspection of potential columns for removal, removal of rows, or imputing data. In addition to this, the ease of plotting correlation matrices, and qq plots are features that are very helpful. For example, from the missing visual it can be seen much of the data is missing on the Ozone column, and from the correlation matrix it can be seen Ozone and Temp are relatively highly positively correlated, and Ozone and wind negatively.

Current comfortability

Thusfar I have spent a lot more time in python, and have used packages such as seaborn coupled with pandas to accomplish similar plots. In R, my experience is with ggplot2, which seems less simple and efficient to get off the ground moving.

Conclusion

In this exercise, some questions were answered regarding ways to clean data in R, what functions can be used to explore data, and how the DataExplorer package works. Overall, using these methods will be important when dealing with raw data that may not be formatted for ML.

References

http://uc-r.github.io/missing_values#missing

http://topepo.github.io/caret/pre-processing.html

https://cran.r-project.org/web/packages/DataExplorer/vignettes/dataexplorer-intro.html#alternative