Automating Data Exploration with R

Outlier Detection

There are different ways of hunting down outliers in a data set but a simple approach is to take the mean or the median of the data and look for any points beyond x standard deviations (68–95–99.7 rule (https://en.wikipedia.org/wiki/68%E2%80%9395%E2%80%9399.7_rule))

```
wt_mean <- mean(mtcars$wt)
print(wt_mean)
```

```
## [1] 3.21725
```

```
wt_sd <- sd(mtcars$wt)
print(wt_sd)
```

```
## [1] 0.9784574
```

How many points in `wt` are outside the 1 sd band (this is fixed from the video - we take a whole standard deviation, not half of the standard deviation - thanks Luis)?

```
sum( (mtcars$wt > (wt_mean + (wt_sd))) | (mtcars$wt < (wt_mean - (wt_sd))))
```

```
## [1] 9
```

```
mtcars$wt[(mtcars$wt > (wt_mean + (wt_sd))) | (mtcars$wt < (wt_mean - (wt_sd)))]
```

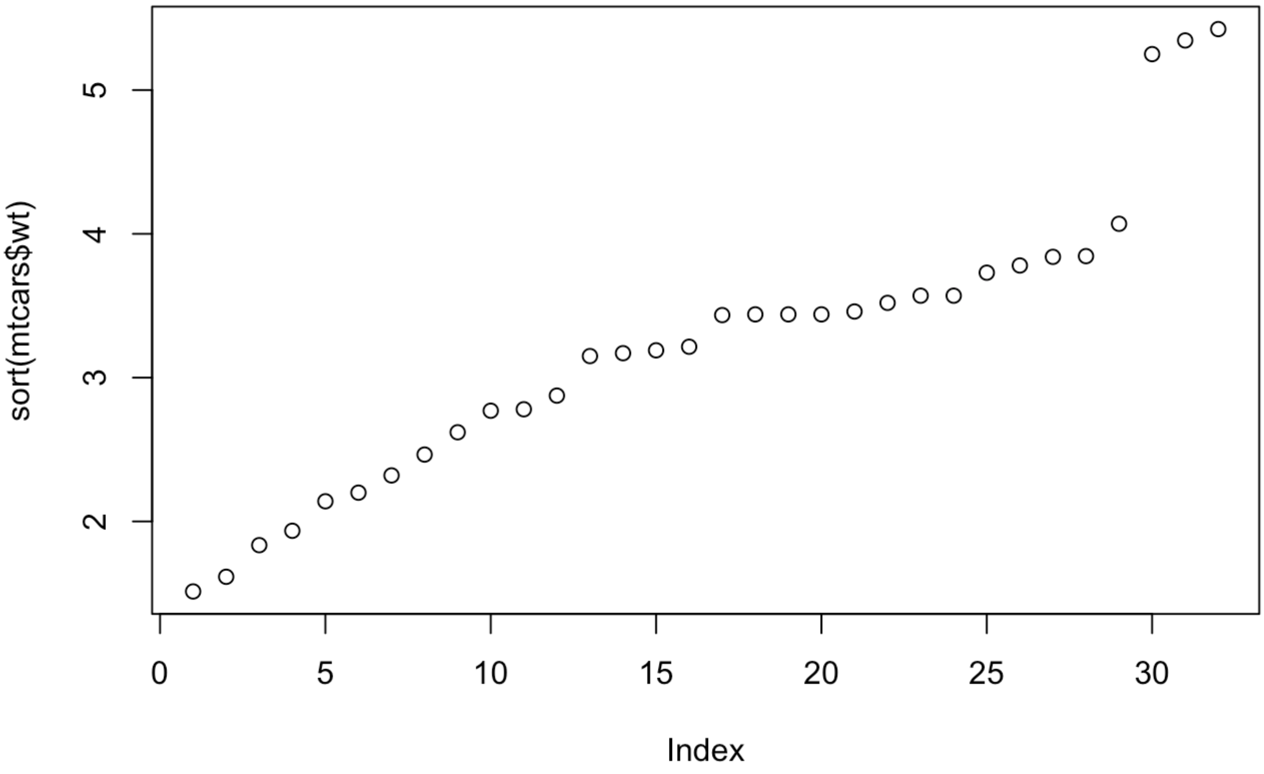
```
## [1] 5.250 5.424 5.345 2.200 1.615 1.835 1.935 2.140 1.513
```

Let’s create a simple but useful function to measure the standard deviation of each feature and detect outliers. This function reports outliers but it can also remove the offending feature with the `remove_outlying_features` function parameter. With just a few extra lines of code it could just as easily impute extreme values down to the mean, 0 or min/max:

```r
Identify_Outliers <- function(data_set, features_to_ignore=c(),
                              outlier_sd_threshold = 2,
                              remove_outlying_features = FALSE) {
     # get standard deviation for each feature
     require(dplyr)
     outliers <- c()
     for (feature_name in setdiff(names(data_set),features_to_ignore)) {
          feature_mean <- mean(data_set[,feature_name], na.rm = TRUE)
          feature_sd <- sd(data_set[,feature_name], na.rm = TRUE)
          outlier_count <- sum(
               data_set[,feature_name] > (feature_mean + (feature_sd * outlier_sd_thresh
old))
               |
               data_set[,feature_name] < (feature_mean - (feature_sd * outlier_sd_thresh
old))
               )
          if (outlier_count > 0) {
               outliers <- rbind(outliers, c(feature_name, outlier_count))
               if (remove_outlying_features)
                    data_set[, feature_name] <- NULL
          }
     }

     outliers <- data.frame(outliers) %>% rename(feature_name=X1, outlier_count=X2) %>%
          mutate(outlier_count=as.numeric(as.character(outlier_count))) %>% arrange(desc(o
tlier_count))

     if (remove_outlying_features) {
          return(data_set)
     } else {
          return(outliers)
     }
}

head(Identify_Outliers(mtcars, remove_outlying_features=FALSE))
```

```
## Loading required package: 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
##   feature_name outlier_count
## 1           wt             3
## 2          mpg             2
## 3           hp             1
## 4         drat             1
## 5         qsec             1
## 6         carb             1
```

```
plot(sort(mtcars$wt))
```


sort(mtcars$wt)
Index