R\_\_Anomaly Detection

## Purpose

In statistics, a outlier is defined as a observation which stands far away from the most of other observations. Outliers exist because of: 1) measurement error or 2) analytical relevance (the outlier value is large or unusual in some way that makes it noteworthy for analysis).

This R analysis’ purpose is to create a framework for assessing outliers in any appropriate dataset.

A fully-completed R outlier analysis will have the following components:

1. Ability to load any appropriate dataset (Excel, CSV, etc.)
2. A step to normalize or scale the data (which is necessary when performing outlier analysis)

## Delivery Format

R markdown document (our group’s standard for a delivery format)

# Step 1

Load data and initialize packages

## Packages Required

library(dplyr)

library(tidyr)

library(ggplot2)

library(outliers)

library()

etc.

# Step 2

Normalize the data

**Normalize**

Normalization the data is a typical step when comparing attributes of different natures.

We will use a normalization method (for example, Z-transformation) to ensure that typical deviations are equal, so that what is an outlier has a clear meaning in all the dimensions of the problem.

Notes

This should be its own section of our R markdown document. The code in this section will take as input the data loaded in Step 1.

This script deals with two approaches for identifying outliers: 1) a multivariate approach (local outlier factor) and 2) univariate approaches.

# Step 3a – Univariate Outlier Detection

***Notes***

*This section on univariate outlier detection will have its own section in our R markdown document.*

## Method 1 – Z-Score

Z-score, also called a standard score, of an observation is [broadly speaking] a distance from the population center measured in number of normalization units. The default choice for center is sample mean and for normalization unit is standard deviation.

Observation is not an outlier based on z-score if its absolute value of default z-score is lower then some threshold (popular choice is 3, but we will develop a script that allows the user to specify this).

**This part of the script will include the following sections:**

A first step that allows us to assign a z-score for the “outlier” threshold.

A section in which we will specify a package that we will use to write code to calculate the z-scores. The script will then use that package to calc the z-scores and retain the results.

## Method 2 – MAD

Median Absolute Deviation is a robust normalization unit based on median as a population center. In order to use MAD “as a consistent estimator for the estimation of the standard deviation” one takes its value multiplied by a factor. This way base R function mad is implemented.

Observation is not an outlier based on MAD if its absolute value of z-score with median as center and MAD as normalization unit is lower then some threshold (popular choice is 3).

## Method 3 – IQR

Tukey’s fences is a technique used in box plots. The non-outlier range is defined with [Q1−k(Q3−Q1), Q3+k(Q3−Q1)], where Q1 and Q3 are the lower and upper quartiles respectively, k - some nonnegative constant (popular choice is 1.5).

Observation is not an outlier based on Tukey’s fences if its value lies in non-outlier range.

Here is the function for a Tukey fence outlier (TRUE = non-outlier, FALSE = outlier)

# Step 3b – Multivariate Outlier Detection

## Local Outlier Factor (LOF)

This section deals with a multivariate approach for finding outliers. The local outlier factor (lof) approach identifies multivariate outliers. LOF compares the density of a data point to densities of neighboring data points.

We will then find the “outlier scoring” using the LOF (local outlier factor) test. Find the outlier scoring using the LOF (local outlier factor) mechanism.

## < See R script: R\_\_LOF >

## Outlier Detection by Clustering

# Step 4

De-normalize the data by applying the reverse normalization model, thereby obtaining the original data. It would also be fine to just cbind the outlier\_scores measure calculated in the last step to another column in the original data. The goal is to end up with a denormalized dataset that has a column added containing the outlier\_scores.

Then we filter the dataset to get one dataset with the outliers and another with the rest, using "outlier=1.5" as a threshold.

The output of this step will consist of two files.

The first file will have the same columns as the input dataset and will also have the following two columns:

1. cluster
   1. This column will contain the cluster to which that observation was assigned. For example, cluster\_1, cluster\_2, etc.
2. outlier
   1. This column will contain the outlier scoring. In the first file, all of the outlier scores should be < 1.5 (since this file represents the non-outliers, and we set out outlier threshold at 1.5).

The second file will have the same columns as the input dataset and will also have the following columns:

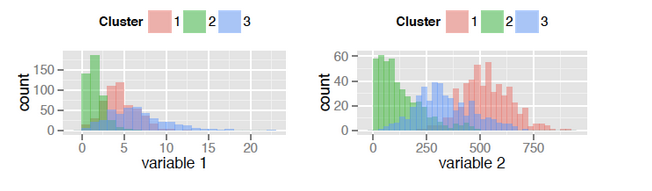
1. cluster
   1. Exactly like the first data frame, the second data frame will have a column called “cluster” that contains the cluster to which that row was assigned.
2. outlier
   1. Again, this column will contain the outlier scoring. In the second data frame, all of the outlier scores should be > 1.5 (since this file represents the outliers, and we set out outlier threshold at 1.5).

**Output**

1. A .csv or Excel file of the second file from Step 4 (the file with only the outliers). Call this file “results\_\_Outliers”
2. A .csv or Excel file of the first file from Step 4. Call this file “results\_Non-outliers”
3. Visuals.

Produce visuals similar to the output shown below for each variable in the analysis.

* 1. Cluster histogram plot that shows how the clusters overlap.
     1. This should be shown for each variable in the data frame.



* 1. Histograms of the variables both with and without outliers.
     1. This will be made from the ***results\_\_Non-Outliers*** and the ***results\_\_Outliers*** files mentioned above.

