Outlier Detection

Eric Cai

2023-09-21

Detecting Outliers

Using crime data from the file uscrime.txt (http://www.statsci.org/data/general/uscrime.txt, description at http://www.statsci.org/data/general/uscrime.html), test to see whether there are any outliers in the last column (number of crimes per 100,000 people). Use the grubbs.test function in the outliers package in R.

As always, let's read in our file and then inspect our data:

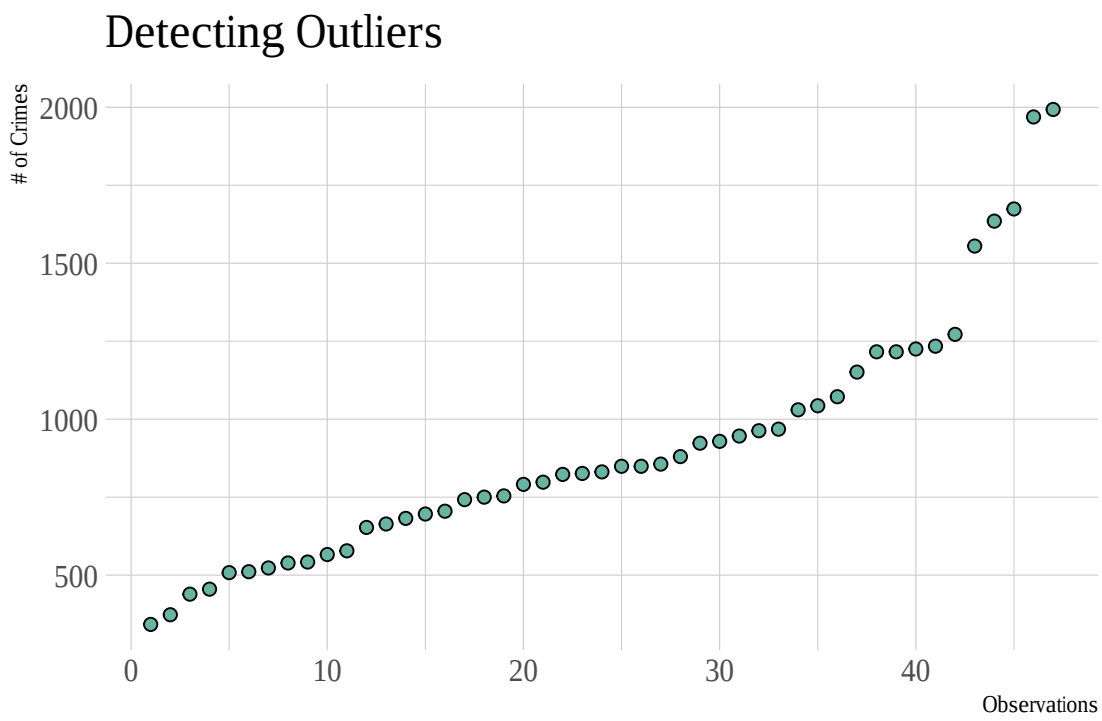
```
file <- 'uscrime.txt'
crime_data <- read.table(file, header=TRUE)
kable(head(crime_data),
      format='markdown',
      align='c',
      caption='Head of US Crime')
```

Table 1: Head of US Crime

M	So	Ed	Po1	Po2	LF	M.F	Pop	NW	U1	U2	Wealth	Ineq	Prob	Time	Crime
15.1	1	9.1	5.8	5.6	0.510	95.0	33	30.1	0.108	4.1	3940	26.1	0.084602	26.2011	791
14.3	0	11.3	10.3	9.5	0.583	101.2	13	10.2	0.096	3.6	5570	19.4	0.029599	25.2999	1635
14.2	1	8.9	4.5	4.4	0.533	96.9	18	21.9	0.094	3.3	3180	25.0	0.083401	24.3006	578
13.6	0	12.1	14.9	14.1	0.577	99.4	157	8.0	0.102	3.9	6730	16.7	0.015801	29.9012	1969
14.1	0	12.1	10.9	10.1	0.591	98.5	18	3.0	0.091	2.0	5780	17.4	0.041399	21.2998	1234
12.1	0	11.0	11.8	11.5	0.547	96.4	25	4.4	0.084	2.9	6890	12.6	0.034201	20.9995	682

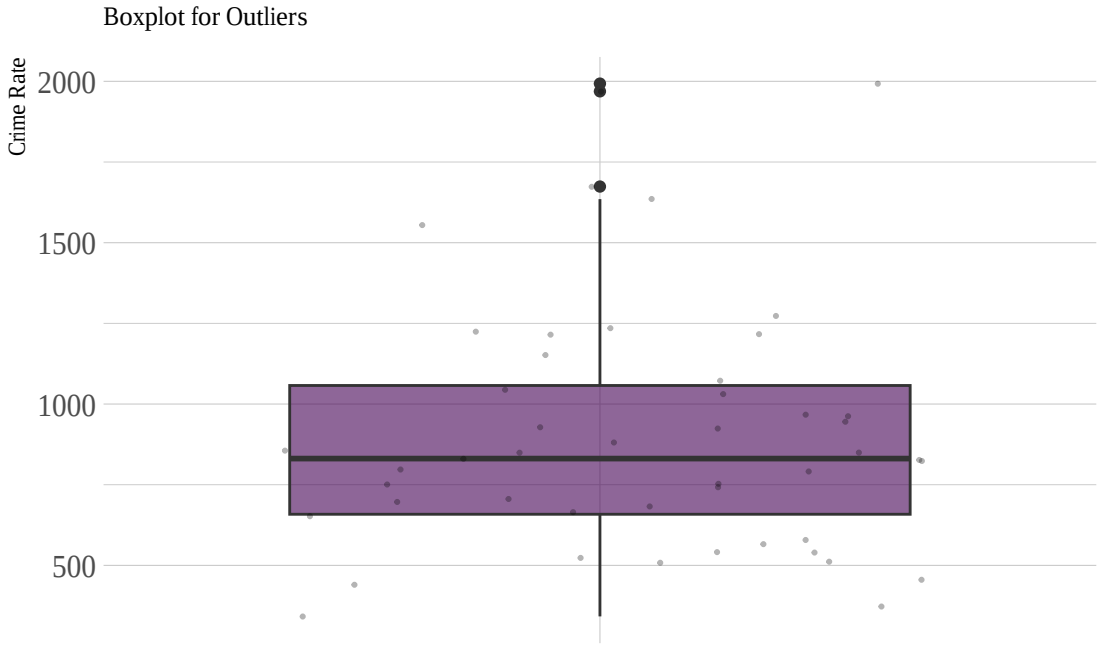
Since we're asked to test whether there are outliers in our crimes column, let's just plot our data to graphically see any anomolies or outliers. There's no guarantee that we'll see anything meaningful, but we'll have a rough idea of how the data graphically looks at least. If the time column was properly formatted, I would have plotted a time-series graph, but I have no idea what the time column means haha so so we'll just use each observation as our x-values.

```
## pipe in our data and transform it into a plot using ggplot
crime_data %>%
        ggplot( aes(x=1:nrow(crime_data), y=sort(Crime))) +
        labs(title='Detecting Outliers',
             x='Observations',
             y='# of Crimes'
             ) +
        theme_minimal() +
        geom_point(shape=21, color="black", fill="#69b3a2", size=2) +
        theme_ipsum()
```



The plot reveals possible outliers at the higher crime rates, but we need a more robust and precise way to detect them. One way is the use of a box-and-whisker plot. So let's create one.

```
crime_data %>%
  ggplot( aes(x="", y=Crime, fill=factor(1))) +
    geom_boxplot() +
    scale_fill_viridis(discrete = TRUE, alpha=0.6) +
    geom_jitter(color="black", size=0.4, alpha=0.3) +
    theme_ipsum() +
    theme(
      legend.position="",
      plot.title = element_text(size=10)
    ) +
    ggtitle("Boxplot for Outliers") +
    labs(x='', y='Crime Rate')
```



The horizontal line through the middle of the box shown above is the median, the 50th percentile, while the top and the bottom of the box are the 25th and the 75th percentiles of the values, and the ends of the vertical lines up and down from the box reveal the local minimum and maximum of our data. Any dots that lie beyond those extreme lines show potential outliers.

The boxplot confirms the summary of the column:

```
summary(crime_data$Crime)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   342.0   658.5   831.0   905.1  1057.5  1993.0
```

As expected, our potential outliers lie outside the extreme lines near the 2000s, which is what we saw in our scatterplot. But it's hard to tell how many there are. While boxplots provide great summaries of data, they are not great at capturing the granularity and distribution of our data, which is why I artificially incorporated jitter. Boxplots require complementary tools to identify outliers as they are difficult to identify on their own.

Luckily, we have a more robust way to identify outliers: the Grubbs' test. But before we utilize this tool, we need to understand what it is.

The Grubbs' Test

The Grubbs' test is used to detect a single outlier in a univariate data set that follows, and this is the key, *an approximately normal distribution*. In other words, the Grubbs' test assumes that our data set is normally distributed. More on that in just a moment. The Grubbs' test, as its name infers, is a hypothesis test by which it tests the two following hypotheses:

1. The **Null Hypothesis** (H_0), which states that there are no outliers in the dataset
2. The **Alternative Hypothesis** (H_1), which states that there is exactly one outlier in the dataset

In a hypothesis test, we assume the null hypothesis is true unless we have reasonable evidence to accept H_1 and reject H_0. So how do we quantify reasonable evidence?

The Test Statistic

The Grubbs' test uses a test statistic defined as:

$$G_i = \frac{max|Y_i - \bar{Y}|}{s}$$

where $\bar{Y}$ and s denote the sample mean and standard deviation, respectively. This test statistic is a random variable that we use to test whether or not H_0 might be true. The test statistic measures the largest deviation a point Y_i may have from the sample mean $\bar{Y}$ in relation to the variability of our crime data (s). This is pretty much how you calculate a Z score, except for the fact that G_i calculates the maximum absolute deviation from the sample mean. In English, the t-statistic summarizes the extent to which our data is consistent with H_0.

Ways to Detect Outliers

1. Assessing Critical Regions If G_i falls out of a critical region of our normally distributed data, then Y_i is statistically farther than the rest of the data. And if G_i falls *within* an accepted region, then Y_i is not an outlier. For a two-sided test, i.e. the highest and the lowest possible outliers, H_0 is rejected if this mess is true:

$$G > \frac{(N-1)}{\sqrt{N}} \sqrt{\frac{(t_{\alpha/(2N),N-2})^2}{N-2+(t_{\alpha/(2N),N-2})^2}}$$

where $t_{\alpha/(2N),N-2}$ denotes the critical value of the t-distribution with $(N-2)$ degrees of freedom and a significance level of $\frac{\alpha}{2N}$ (in a two-sided test). For a one-sided test, we drop the 2, leaving $\frac{\alpha}{N}$. More on α later on.

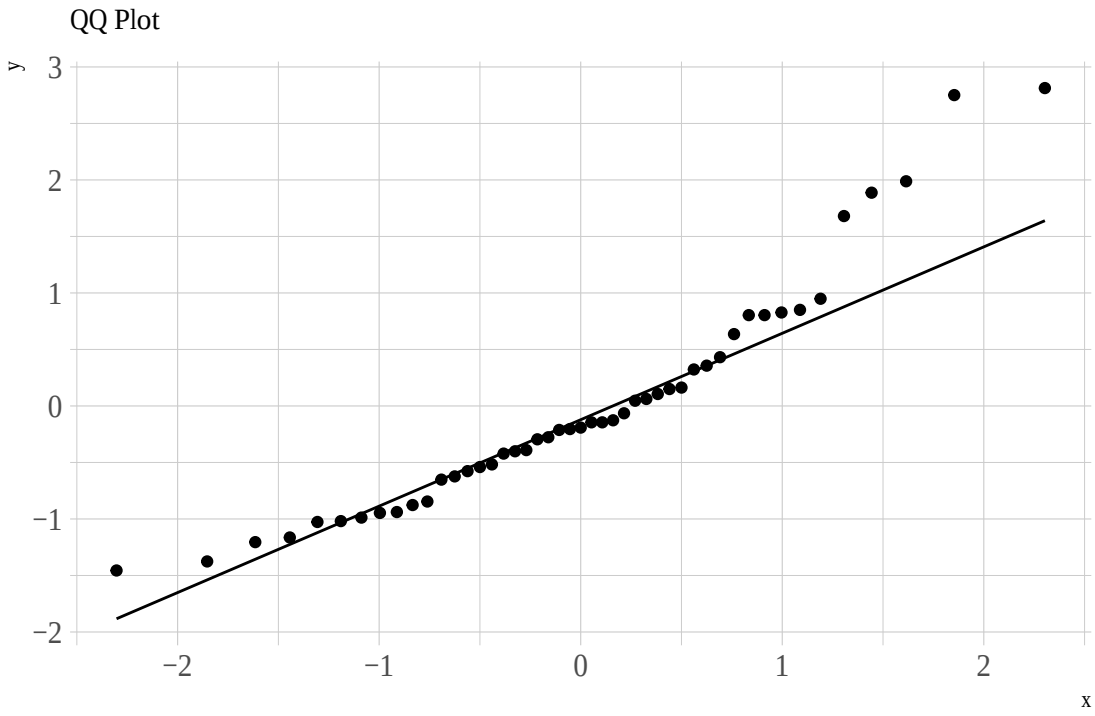
Standardizing Our Data This is why the Grubb' test requires that our data be normally distributed. So we'll need to scale our crime rate column (remember, univariate) and determine if it follows an approximately normal distribution before we perform the Grubbs' test. Let's do that now before we move on to an alternative way to assess outliers.

```
## scale() will standardize our crime column
crime_scaled <- scale(crime_data$Crime)
```

Tests for Normality After we've scaled our data, we'll want to test to see if our scaled data follows a normal distribution. There are a variety of ways to test for normality, many of which involve hypothesis testing. I will opt for a graphical method, since using a hypothesis test within a hypothesis test is too meta. (What is this? Inception?)

Using a quantile-quantile (QQ) plot, we'll see how well the data fits a straight line. We already saw a similar plot of this at the start of this report, but now we're fitting a line to the data. If the data were normally distributed, most of the points would be on the line.

```r
ggplot(data.frame(crime_scaled), aes(sample=crime_scaled)) +
        geom_qq() +
        geom_qq_line() +
        theme_minimal() +
        labs(title='QQ Plot') +
        theme_ipsum() +
        theme(
                plot.title = element_text(size=11)
)
```



Most of the data seems to be on the line-ish, which suggests that our data approximately follows a normal distribution.

2. Comparing P-Values and α Returning back to our discussion of detecting outliers, an alternative way to assess outliers is to calculate the p-value of our Grubbs' test and compare it to our sensitivity for error, α. When we conduct a hypothesis test, we typically end up either rejecting or failing to reject the null hypothesis H_0. Sometimes we make the correct decision, sometimes not. α, also known as the level of significance, is the error rate at which we wrongly reject H_0. It is our threshold for error.

How we determine this threshold for error depends on how important the experiment we're conducting is. For example, if we're testing the efficacy of a drug, our tolerance for error might be very, very small, like 0.00001–that is, for every 100,000 experiments, we would encounter 1 false-positive. If, however, our experiment isn't as important, like the alternative hypothesis (H_1) that my local Trader Joe's is cursed because they always run out of snacks, then we might use a larger tolerance for error, like 0.2. Using a threshold of 0.2 means that I am willing to be right 2 times out of 10 (one can hope).

The most common threshold is 0.05 because trying to reduce the number of False-Positives below 5% often costs more than it's worth. If α is very small, it makes it harder to reject H_0 even in the case that it happens to be false. The p-value, then, is the probability of seeing a test statistic (G_j) as extreme as, or more extreme, than the one that we calculated (G_i), assuming that there are no outliers present.

If the p-value is small, G_i is unlikely to occur by random chance alone when outliers are absent, suggesting that the result is statistically significant and not likely to be due to random chance. The larger the p the more probable it is for an event to occur by random chance thus supporting the null hypothesis. So if $p < \alpha$, then we reject H_0, and if $p \geq \alpha$, then we fail to reject H_0.

Using the example of my local Trader Joe's being cursed, a p-value less than α would suggest that the snack shortages are statistically significant and not likely due to random chance, supporting the hypothesis that my Trader Joe's is cursed. On the other hand, a p-value greater than or equal to α would suggest that the shortages are statistically insignificant and is likely due to random chance, suggesting that we don't have enough evidence to reject the null hypothesis (which is not as fun as the alternative hypothesis that Trader Joe's is cursed).

We formulate this problem as a piece-wise function:

$$G_i = \begin{cases} H_0 & \text{if } p < \alpha \\ H_1 & \text{if } p \geq \alpha \end{cases} \tag{1}$$

Finding Outliers

So that was all words and no code. Let's finally use grubbs.test to determine our outliers. As always, for visual purposes, I will be storing the potential outliers that we find in a table.

```
## creating our table
outliers_tbbl <- data.frame(Outlier = numeric(0),
                            PValue = numeric(0),
                            IsOutlier = logical(0))
## we'll sort our data from lowest to highest, not that it really matters
data <- sort(crime_data$Crime)
## the while-loop loops until we've iterated through every data point
## the loop breaks when we've met our criteria defined below
while(length(data) > 0){
        ## grubbs.test starts with the highest outlier
        upper <- grubbs.test(data, type=10)
        ## if we want the lowest, we set opposite to TRUE
        lower <- grubbs.test(data, opposite=TRUE, type=10)
        ## we check whether p < alpha
        ## if the p-value of our upper outlier is less than our alpha
        ## meaning that it's unlikely for an outlier to be there
        if(upper$p.value < 0.1){
                ## then we grab our outlier
                outlier_value <- as.numeric(gsub("[^0-9.]",
                                                 "",
                                                 upper$alternative[[1]]))
                ## put it into an upper outlier dataframe
                upper_df <- data.frame(Outlier=outlier_value,
                                       PValue=upper$p.value,
                                       IsOutlier=TRUE
                )
                ## append it to our total outliers table
```

```
                outliers_tbbl <- rbind(outliers_tbbl, upper_df)
                ## then we remove the outlier from our dataset and check again
                ## until there are no more outliers that meet the criteria
                data <- data[data != outlier_value]
        ## then we move on to the lower outliers and repeat the same thing
        } else if(lower$p.value < 0.1){
                outlier_value <- as.numeric(gsub("[^0-9.]",
                                                 "",
                                                 upper$alternative[[1]]))
                lower_df <- data.frame(Outlier=outlier_value,
                                       PValue=lower$p.value,
                                       IsOutlier=TRUE)
                outliers_tbbl <- rbind(outliers_tbbl, lower_df)
                data <- data[data != outlier_value]
        } else{
        ## if there are no p-values that meet our threshold for tolerance
        ## then it means we have no outliers and we end the loop
                break
        }
}
kable(outliers_tbbl,
      row.names=T,
      col.names=c('# of Crimes', 'P-Value', 'Outlier'),
      align='c',
      caption=paste('Grubbs\' Test with $\\alpha$ Set to 0.1')
)
```

Table 2: Grubbs' Test with α Set to 0.1

	# of Crimes	P-Value	Outlier
1	1993	0.0788749	TRUE
2	1969	0.0284782	TRUE

Interpreting Our Results

Our loop returns only 2 outliers, which is mainly consistent with the graphs that we saw above. If you remember the boxplot, there was at least one more datapoint outside the extreme line in the ~1600s. This didn't meet our threshold, so our loop stopped and didn't record it in our table.

One important detail to point out is that I set our α to 0.1. I had previously set it to 0.05 and it wasn't flagging any of our data points as an outlier. So, again, how we choose our α is important.

If we really don't want to wrongly flag our crime rates, then we set a lower threshold for error. In effect, this means that we incorrectly flag a high crime rate as an outlier 5 out of 100 times. And if we set a higher threshold for error, say 0.1, then we flag a high crime rate as an outlier 1 out of 10 times. As one can imagine, setting α can be politically or socially-motivated and has far-reaching implications.

If an urban planning organization wants to give off the impression that a city is safer than it actually is, they may want to increase their threshold for error, in effect saying that high crime rates are just anomalies rather than being a part of the data, thereby downplaying the significance of high crime rates. On the other hand, if public policy wants to secure more funding to address high crime rates, they may want to minimize the number of times a high crime rate is flagged as an outlier, essentially saying that high crime is actually more

consistent, rather than inconsistent, with the data. These examples illustrate how adjusting the threshold for error in outlier detection can have significant implications for various public and private sectors.

As such, the choice of α should be carefully considered based on the specific context. As we've seen so far, many of these decisions are trade-offs between being conservative (minimizing false positives) and being sensitive (detecting genuine outliers effectively).

This is why researchers often report the p-values of any test that they conduct. Doing so prevents parties with hidden agendas from picking and choosing certain α values after the fact in order to manipulate a desired reject/accept conclusion.