Análisis de Anomalías

Jose Fernando Zea y Fernando López-Torrijos

6 de abril de 2021

Prólogo

En español se define anomalía como "Desviación o discrepancia de una regla o de un uso". En inglés lo definen como "something that is unusual enough to be noticeable or seem strange". Según Aggarwal (2017) un outlier es un punto que difiere significativamente de los demás puntos o si difiere del mecanismo generador de los datos.

Podría hablarse de muchos tipos de anomalías dentro de un conjunto de datos:

- Datos faltantes
- Valores no probables (¿error de digitación?)
- Valores atípicos
- No respuesta
- Formato incorrecto

Algunos definen los datos atípicos como datos muy *grandes* o muy *pequeños* comparados con el grueso del conjunto de datos. Son observaciones con un comportamiento extraño porque toman valores que no se esperan. Pero esos datos es mejor denominarlos *valores extremos*.

Este documento se centra en las *anomalías*, distinguiendo éstas del ruido aleatorio, que también puede generar algunos **valores atípicos**. La separación entre ruido aleatorio y dato anómalo no es siempre clara.

El dato en color rojo de la Figura 1 está dentro de la tendencia de los demás datos, pero es significativamente grande respecto a los restantes mirado desde el punto de vista de la variable x. Desde el punto de vista de la variable y, podría tratarse de un valor indistingible. Podría tratarse de un dato aislado alto pero válido o podría tratarse de un dato anómalo.

El dato en color rojo de la Figura 2 está fuera de la tendencia. Dentro de una mirada a sólo la variable x, no resalta. Dentro de solo la variable y tampoco. Sólo en la dimensión conjunta x, y se observa su anomalía.

El dato en color rojo de la Figura 3 se presentaría tanto en x como un dato anómalo en medio de dos conglomerados de datos. Respecto a la variable y, tal vez sería indistinguible. Es un dato anómalo desde la dimensión conjunta x, y.

Aplicaciones

- Sistema de detección de intrusiones: transacciones bancarías anómalas (demasiados movimientos en una cuenta), actividad inusual en una red de telefonía móvil.

- Fraude de tarjeta de crédito: localizaciones, movimientos inusuales. Eventos obtenidos a partir de sensores: comportamientos extraños que pueden estar asociados al mal funcionamiento de un dispositivo.

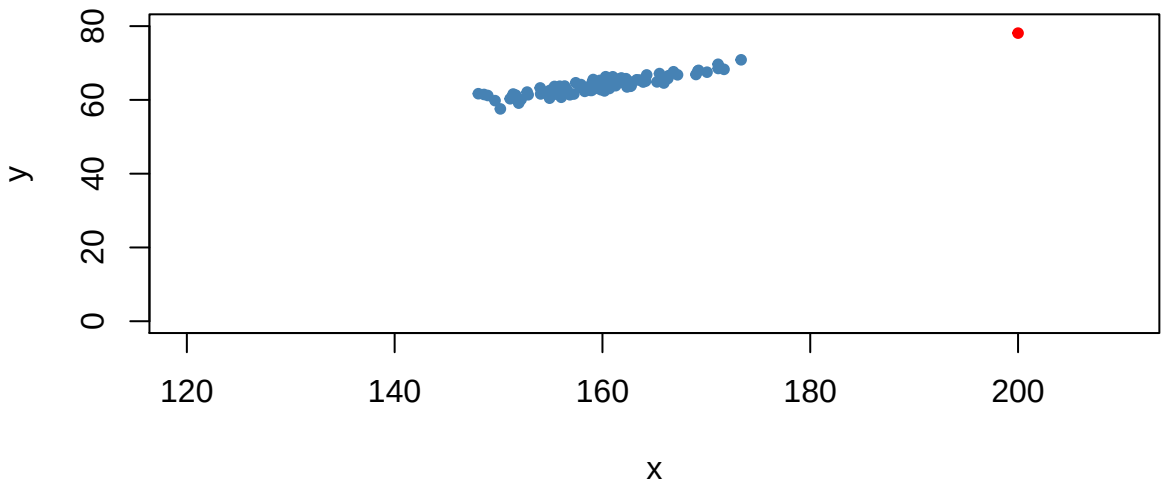


Figure 1: Dato extremo

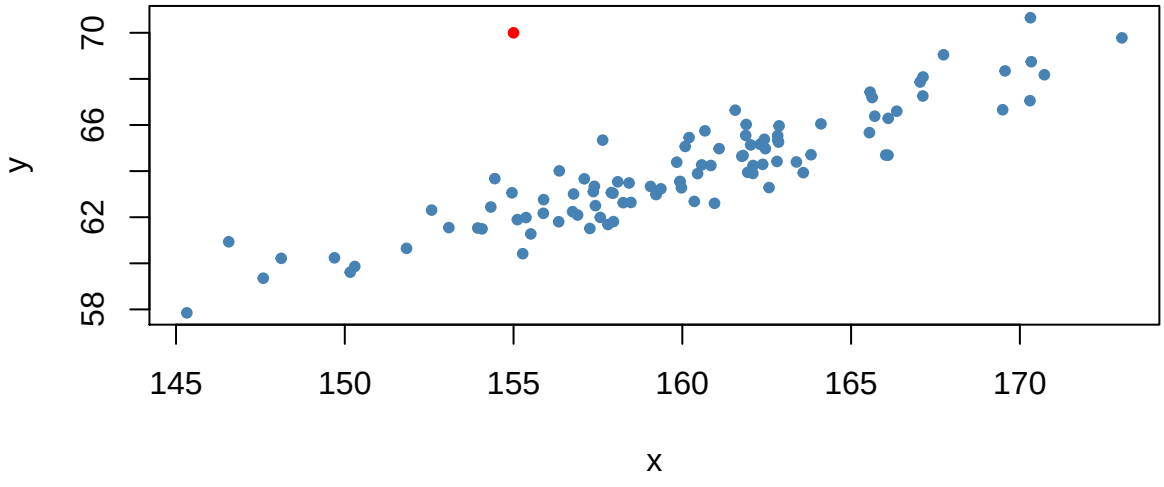


Figure 2: Dato fuera de la tendencia

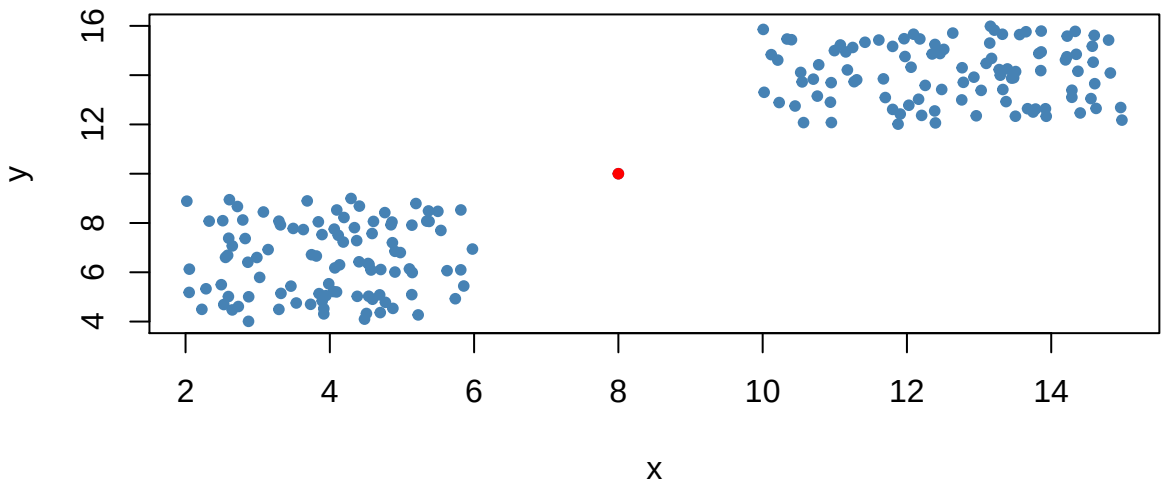


Figure 3: Dato intermedio

- Diagnóstico médico: imágenes de resonancia magnética, tomografías, electrocardiogramas pueden estar asociadas a detección de enfermedades.
- Incumplimiento de la ley: reclamaciones de seguros, actividad de trading, movimientos financieros.
- Ciencias de la tierra: anomalías ambientales, uso del suelo, condiciones ambientales anómalas.
- Procesamiento de imágenes: Eliminación de la pixelación que impide una mejor interpretación de una imagen.

Presentación de las técnicas tradicionales o más usuales.

Análisis variable a variable.

El primer caso que se trabaja es la definición de un dato anómalo cuando se trabaja con una única variable a la vez.

El primer contexto en el que se encuentran es en el diagrama de caja o boxplot.

El Diagrama de caja se construye con base en los cuartiles (ver Figura 4):

Cuartil 1 (q1): valor a partir del cual el 25% de los datos tienen un valor menor a éste. O leído al revés, el valor a partir del cual el 75% de los datos tienen un valor mayor a éste.

Ojo. El 25% hace referencia a la cantidad de datos, no a sus valores.

Cuartil 2 (q2): valor a partir del cual el 50% de los datos tienen un valor menor. Se denomina también mediana. Es un valor robusto frente a datos extremos, es decir, no se afecta por la presencia de datos extremos.

Cuartil 3 (q3): valor a partir del cual el 75% de los datos tienen un valor menor a éste. O leído al revés, el valor a partir del cual el 25% de los datos tienen un valor mayor a éste.

Rango intercuartil (RI): La diferencia entre el cuartil 3 y el 1. $RI = q3 - q1$

Bajo las reglas de Tukey, un estadístico activo en las décadas del 40 al 80 del siglo XX, y quien popularizó los diagramas de caja, todo dato que está alejado más de 1.5 veces el RI del cuartil más cercano se dice que es un dato *atípico*. Un dato atípico lo denominan *extremo* si está ubicado a una distancia mayor de 3 veces el RI del cuartil más cercano y se llama *moderado* en caso contrario.

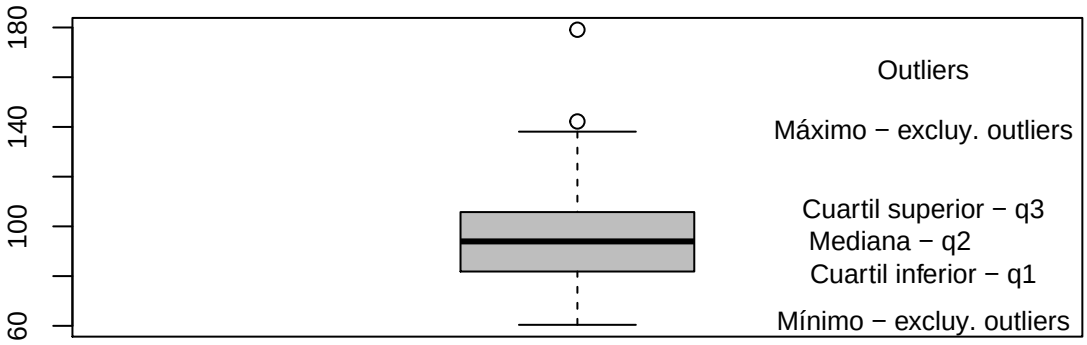


Figure 4: Elementos de un diagrama de caja

Sobre el conjunto de datos que generó el diagrama del ejemplo (ver Figura 4), hay dos círculos que reflejan los datos atípicos.

Atípico moderado: 142.2

Atípico extremo: 179.1

Los cuartiles de la Figura 4 son: 81.9, 93.9, 105.5

Regla empírica de Tukey

Tukey en los años 60 construyó una regla para identificar valores extremos en datos con distribuciones gausianas o normales, para esto se basó en el uso de dos cantidades conocidas como bisagras o bigotes (hinges/whiskers en la lieratura anglosajona).

El bigote inferior se calcula como $LW = q_1 - k \times RI$ Es decir, el bigote inferior (lower whisker) se calcula como el percentil 25 (o primer cuartil) menos k veces el rango intercuartílico (RI). Tukey propuso como valor de k = 1,5.

SI LW es menor al mínimo de los datos el valor que finalmente se deja como bigote inferior es el mínimo. Es decir,

$$LW = max(min(x), q_1 - k \times RI)$$

El bigote superior se calcula como:

$$UW = min(max(x), q_3 + k \times RI)$$

Bajo normalidad, con un valor de 1,5 la probabilidad de que un dato sea considerado como un valor extremo es de 0.006977´. En otras palabras aproximadamente 7 de cada mil valores serán detectados como valores extremos.

Filzmoser, Gussenbauer, and Templ (2016) propusieron realizar previamente una transformación de las distribuciones asimétricas con el objeto de aproximarlas a una distribución normal y posteriormente aplicar la regla de Tukey para detección de valores extremos.

$$y(\lambda) = \begin{cases} \frac{x^{\lambda}-1}{\lambda}, \text{Sí } \lambda \neq 0 \\ log(\lambda), \text{Sí} \lambda = 0 \end{cases}$$

Ilustraremos la transformación de Box - Cox con un ejemplo en python:

```
# importar modulos
import os
import pandas as pd
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns
```

```
## C:\Users\Fernando\Documents\R\win-library\4.0\reticulate\python\rpytools\loader.py:24: VisibleDeprec
##     Install tornado itself to use zmq with the tornado IOLoop.
##
##   level=level
```

```
from sklearn import datasets
from scipy.stats import t, zscore
import random
```

Simularemos una distribución asimétrica a la izquierda:

```
# Gnerar una distribución asimétrica (una exponencial en particular)
np.random.seed(0)
x = np.random.exponential(size = 1000)
np.mean(x), np.var(x)
```

```
## (1.003540208760709, 1.0590341339276639)
```

```
# Transformación de Box-Cox (Dupla: arreglo de valores transformados y lambda)
y, fitted_lambda = stats.boxcox(x)
```

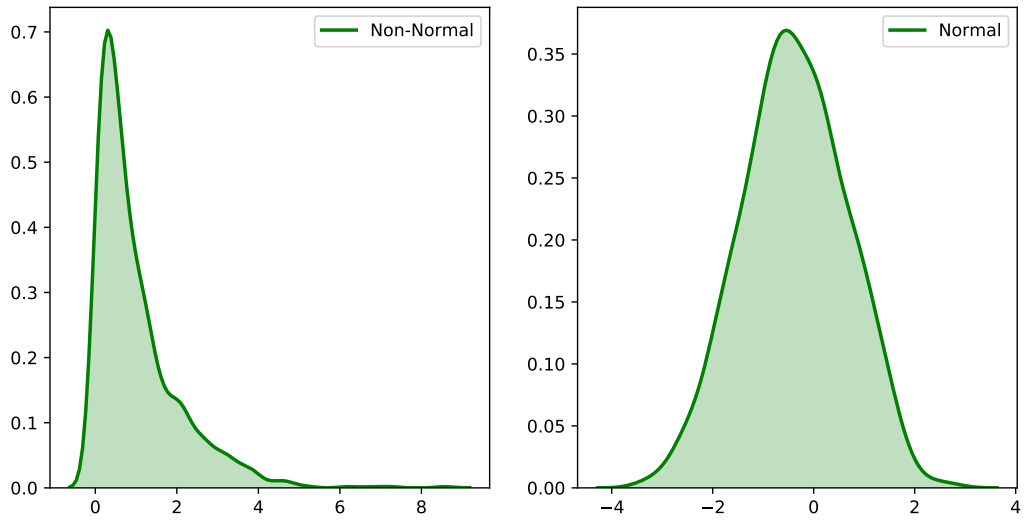
Se compara la distribución original con la transformada mediante Box - Cox

```
# crear ejes para el gráfico: 1 fila 2 columnas
fig, ax = plt.subplots(1, 2)
# graficando la variable en su manera original y luego de transformada
sns.distplot(x, hist = False, kde = True,
             kde_kws = {'shade': True, 'linewidth': 2},
             label = "Non-Normal", color ="green", ax = ax[0])

sns.distplot(y, hist = False, kde = True,
             kde_kws = {'shade': True, 'linewidth': 2},
             label = "Normal", color ="green", ax = ax[1])

# Anadiendo leyenda a los gráficos
plt.legend(loc = "upper right")
```

```
# re-escalando los subplots
fig.set_figheight(5)
fig.set_figwidth(10)
plt.show()
# Se restituyen los valores a 1 fila 1 columna para los plots
```




```
fig, ax = plt.subplots(1, 1)
```

```
print(f"Lambda utilizado para la transformación: {fitted_lambda}")
```

```
## Lambda utilizado para la transformación: 0.242013194225711
```

Ejercicio: se ilustra la identificación de atípicos con la base de datos iris. Se utiliza para la demostración la variable Sepal.Length. Dada la aparentemente normalidad de los datos se detectarán directamente los *outliers* mediante la regla de Tukey.

Python no tiene una función para su cálculo, así que se crea la función *fivenum* para que genere calcule y entregue los cinco números de Tukey: bigote inferior, primer cuartil, mediana, tercer cuartil y bigote superior, para una lista, arreglo univariado de numpy o serie de pandas.

```
def fivenum(x, range = 1.5, nan_remove = True):
    """Devuelve los cinco números de Tukey (mínimo, bigote inferior, mediana,
    bigote superior, maximo) para una lista, arreglo univariado de numpy o serie de pandas"""

    if(isinstance(x, list)):
        x = np.array(x)
    try:
        np.sum(x)
    except TypeError:
        print('Error: debe proveer una lista o un arreglo de sólo números')
    if(nan_remove == True):
        y = x[~np.isnan(x)]
        q1 = np.percentile(y, 25)
        q3 = np.percentile(y,75)
        md = np.median(y)
```

```
        RI = q3-q1
        lower_whisker = q1 - 1.5 * RI
        upper_whisker = q3 + 1.5 * RI
        lower_whisker = np.max([lower_whisker, np.min(y)])
        upper_whisker = np.min([upper_whisker, np.max(y)])

    else:
        q1 = np.percentile(x, 25)
        q3 = np.percentile(x,75)
        md = np.median(x)
        RI = q3-q1
        lower_whisker = q1 - 1.5 * RI
        upper_whisker = q3 + 1.5 * RI
        lower_whisker = np.max([lower_whisker, np.min(x)])
        upper_whisker = np.min([upper_whisker, np.max(x)])

    salida = np.array([lower_whisker, q1, md, q3, upper_whisker])
    return salida
```

Se cargan los datos para el ejemplo:

```
iris = datasets.load_iris()
iris_df = pd.DataFrame(iris.data, columns=iris.feature_names)
iris_df['target'] = iris['target']
iris_df.columns = ['sepal_length', 'sepal_width', 'petal_length',
       'petal_width', 'species']
```

Se calculan los 5 números de Tukey y se guardan en la variable *fnums*:

```
fnums = fivenum(iris_df.sepal_width)
fnums
```

```
## array([2.05, 2.8 , 3.  , 3.3 , 4.05])
```

Se crea una nueva variable denominada *outlier_sw* donde se almacena si es o no es un *outlier*. Se imprime el conteo.

```
iris_df['outlier_sw'] = iris_df['sepal_width'].apply(lambda x: 'outlier' if (x < fnums[0]) | (x > fnums
iris_df.outlier_sw.value_counts()
```

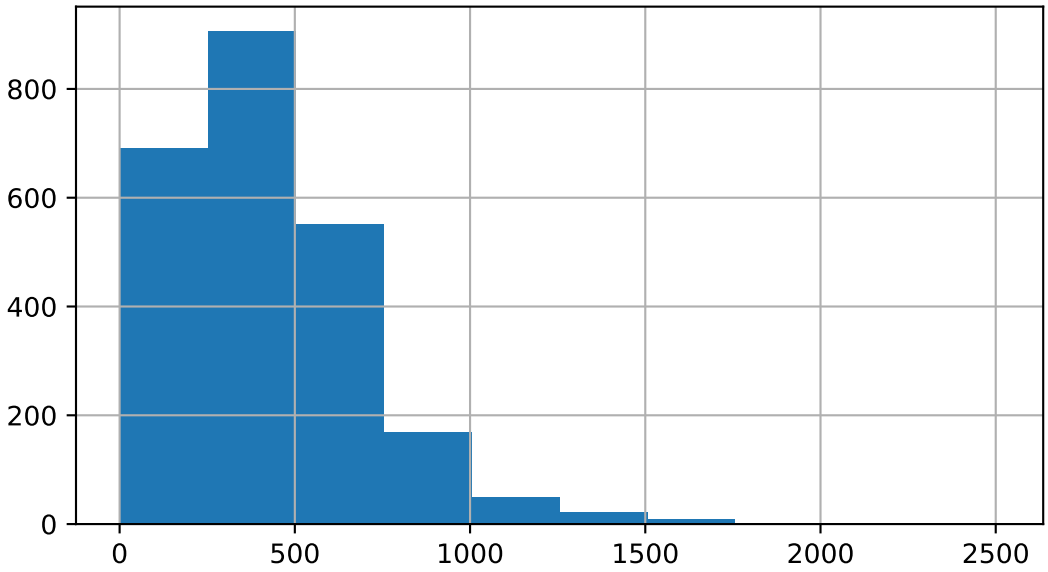
```
## no_outlier    146
## outlier         4
## Name: outlier_sw, dtype: int64
```

Ejemplo de una variable sesgada

Detectar los *outliers* de la variable *Income* de base de datos de empresas Lucy.

```
Lucy = pd.read_csv("Lucy.csv")
```

```
Lucy.Income.hist()
plt.show()
```



Se evidencia que la distribución del ingreso es muy asimétrica a la derecha, por lo tanto se aplica la transformación de Box-Cox y se almacena en la variable *Income_bc*:

```
#  (Dupla: arreglo de valores transformados y lambda)
y_income_bc, fitted_lambda_income = stats.boxcox(Lucy.Income)
Lucy['Income_bc'] = y_income_bc
Lucy.head()
```

```
##       ID Ubication  Level Zone  Income  Employees  Taxes SPAM  Income_bc
## 0  AB001      c1k1  Small    A     281         41    3.0   no  18.293608
## 1  AB002      c1k2  Small    A     329         19    4.0  yes  19.521137
## 2  AB003      c1k3  Small    A     405         68    7.0   no  21.248678
## 3  AB004      c1k4  Small    A     360         89    5.0   no  20.253837
## 4  AB005      c1k5  Small    A     391         91    7.0  yes  20.947122
```

El valor de lambda considerado por parte de la función *stats.boxcox* es:

```
fitted_lambda_income
```

```
## 0.35890092929069184
```

Se guardan los cinco números de Tukey en la variable *fn_income_bc*:

```
fn_income_bc = fivenum(Lucy.Income_bc)
fn_income_bc
```

```
## array([ 5.34786936, 16.83158034, 20.92531863, 24.48738767, 35.97109866])
```

Se crea una nueva variable denominada *outlier_Income* donde se almacena si es un outlier. Se imprime el conteo.

```
Lucy['outlier_Income'] = Lucy['Income_bc'].apply(lambda x: 'outlier' if (x < fn_income_bc[0]) | (x > fn
Lucy.outlier_Income.value_counts()
```

```
## no_outlier    2379
## outlier         17
## Name: outlier_Income, dtype: int64
```

Ejercicio:

Detectar los valores extremos de la variable *Employees*.