

CCSS_atipico_final

May 8, 2021

```
[ ]: %%capture
from google.colab import drive
drive.mount('/content/drive')

[ ]: path="/content/drive/MyDrive/alejoherrera.com/análisis/"

[ ]: import pandas as pd
import plotly.express as px
import io
import requests
from io import StringIO
import os
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import datasets
from scipy.stats import t, zscore
import random
from sklearn.neighbors import LocalOutlierFactor
from sklearn.metrics import confusion_matrix
from plotnine import ggplot, geom_point, aes, geom_abline, theme, element_text, \
↳ labs
from sklearn import preprocessing
from sklearn.ensemble import IsolationForest
import warnings
warnings.filterwarnings("ignore")
```

1 Detección de datos estadísticamente atípicos en compras.

Elaborado por Alejandro Herrera López (alejandroherreracr@gmail.com)

Este documento está cubierto en su prosa por la licencia , siendo un documento de cultura libre.

En su código fuente está cubierto por GPLv3

2 Contexto del presente trabajo:

Es un trabajo completa y exclusivamente a título personal, se debe tener el cuidado en el momento de su lectura, que un dato estadísticamente atípico no implica por sí mismo, una actuación irregular o extraña de la Administración Pública. El trabajo se realiza a un alto nivel, sin ingresar a información del expediente de cada contratación y para la realización del mismo se utiliza como fuente el Sistema Integrado de la Actividad Contractual, de la Contraloría General de la República, el cual a su vez se alimenta de información ingresada por la misma.

```
[ ]: data_CCSS=pd.read_excel(path+"compras_CCSS_2020.xlsx")
```

3 Datos analizados:

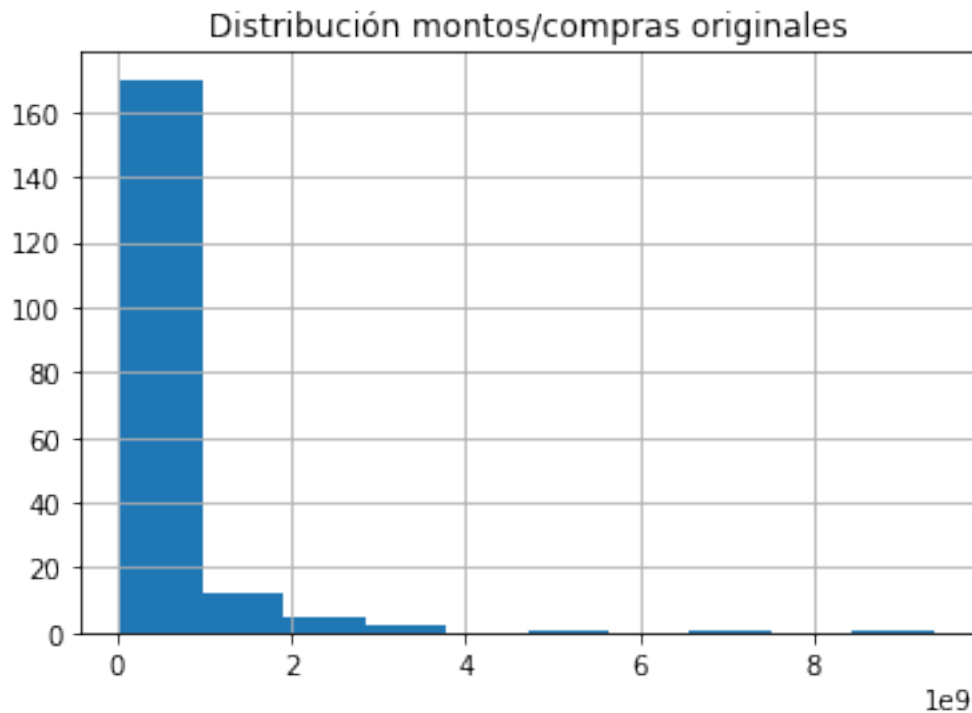
Para el mismo se utilizó como dataset las contrataciones directas registradas por la **CCSS, por un monto por encima de los 50 millones de colones**. A las variables extraídas y disponibles en el SIAC, se le incorporó un cálculo de los días entre las fechas de invitación a apertura de cada procedimiento, y los días entre las fechas de apertura y adjudicación. A continuación se hace un detalle de las variables disponibles:

```
[ ]: data_CCSS.dtypes
```

```
[ ]: INSTITUCION LICITANTE      object
      ANNIO                     int64
      NUMERO_PROCEDIMIENTO      object
      OBJETO_CONTRACTUAL        object
      FECHA_APERTURA           datetime64[ns]
      FECHA_INVITACION          datetime64[ns]
      DIAS_INVITACION_APER      int64
      FECHA_ADJUDICACION        datetime64[ns]
      DIAS_APERTURA_ADJU       int64
      FECHA_COMUNICO            datetime64[ns]
      LIMITE_LEGAL              object
      MODALIDAD_CONTRATACION    object
      PRESUPUESTO_ESTIMADO      float64
      EXCEPCION_CONTRATACION_DIRECTA object
      LINEA                     int64
      ESTADO_LINEA              object
      BIEN_SERVICIO              object
      CEDULA_ADJUDICADO         int64
      ADJUDICADO                object
      CANTIDAD_REQUERIDA        int64
      CANTIDAD_ADJUDICADA       int64
      INESTIMABLE               object
      MONTO_ADJUDICADO          float64
      SUBPARTIDA_GASTO          object
      dtype: object
```

```
[ ]: #se transforma en numerica la variable de monto
data_CCSS.MONTO_ADJUDICADO=pd.to_numeric(data_CCSS.MONTO_ADJUDICADO)
```

```
[ ]: data_CCSS.MONTO_ADJUDICADO.hist()
plt.title('Distribución montos/compras originales')
plt.show()
```

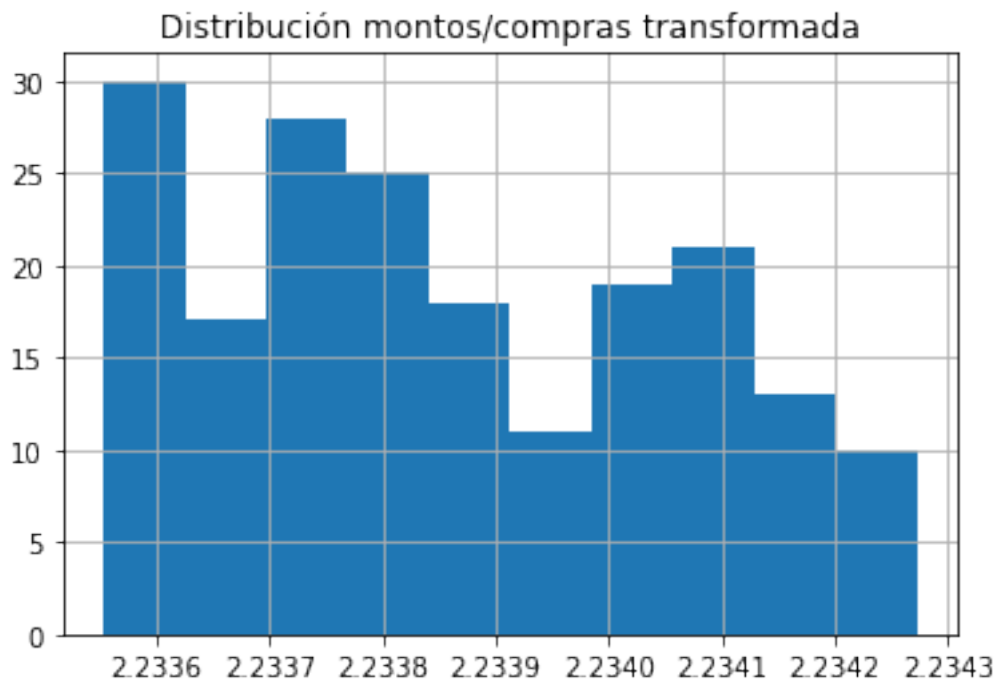


Se evidencia que la distribución del ingreso es muy asimétrica a la derecha, por lo tanto se aplica la transformación de Box-Cox.

Se dispone el histograma con los datos transformados.

```
[ ]: y_MONTO_ADJUDICADO_bc, fitted_lambda_MONTO_ADJUDICADO = stats.boxcox(data_CCSS.
    ↳MONTO_ADJUDICADO)
data_CCSS['MONTO_ADJUDICADO_bc'] = y_MONTO_ADJUDICADO_bc
```

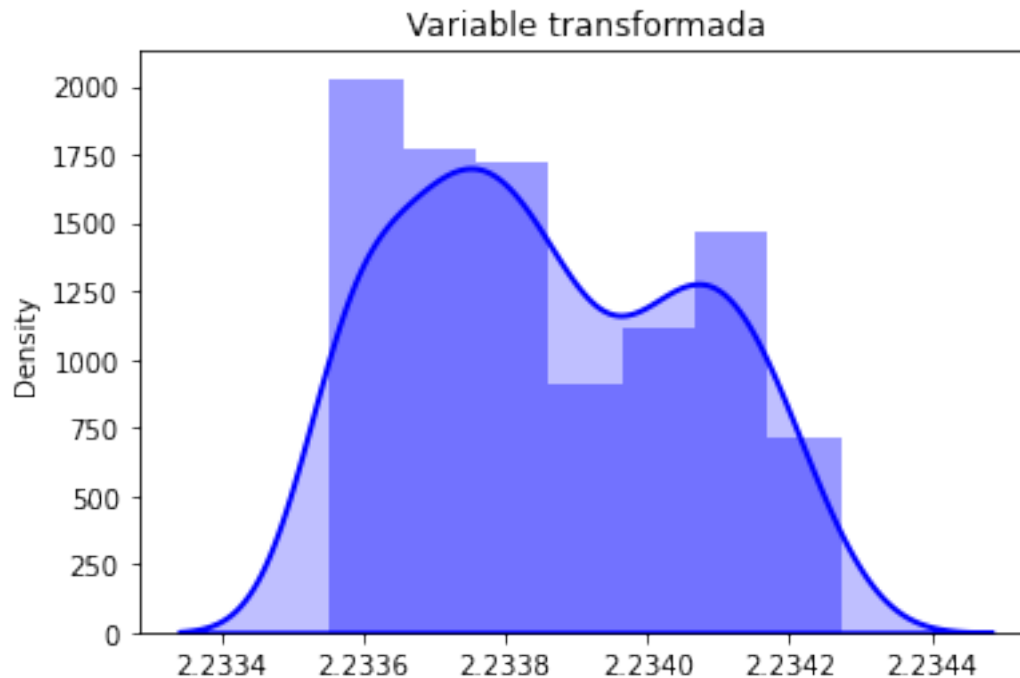
```
[ ]: data_CCSS.MONTO_ADJUDICADO_bc.hist()
plt.title('Distribución montos/compras transformada')
plt.show()
```



```
[ ]: # crear ejes para el gráfico: 1 fila 2 columnas
```

```
sns.distplot(y_MONTO_ADJUDICADO_bc, hist = True, kde = True,
              kde_kws= {'shade':True, 'linewidth':2},
              label = 'Variable transformada',
              color = 'blue').set_title('Variable transformada')
```

```
[ ]: Text(0.5, 1.0, 'Variable transformada')
```



[]:

4 Aplicación de la regla de Tukey para ubicar outliers

```
[ ]: def fivenum(x, range = 1.5, nan_remove = True):
    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 - range * RI
        upper_whisker = q3 + range * 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 - range * RI
upper_whisker = q3 + range * 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

```

```

[ ]: tuk = fivenum(y_MONTO_ADJUDICADO_bc)
print(tuk)

```

```

[2.23355339 2.23370148 2.23383383 2.23403521 2.23427244]

```

```

[ ]: data_CCSS['outlier_tukey'] = pd.Series(y_MONTO_ADJUDICADO_bc).apply(lambda x:
    ↳ 'outlier' if (x < tuk[0]) | (x > tuk[4]) else 'no_outlier')
data_CCSS.outlier_tukey.value_counts()

```

```

[ ]: no_outlier    192
     Name: outlier_tukey, dtype: int64

```

```

[ ]:

```

5 Atipicidad por algoritmo Local Outlier Factor

Se realiza el análisis con las variables numéricas disponibles, a saber :

['DIAS_INVITACION_APER', 'DIAS_APERTURA_ADJU', 'CANTIDAD_ADJUDICADA', 'MONTO_ADJUDICADO']

Para detectar atipicidades se aplica el algoritmo de Local Outlier Factor, implementado en la librería de sklearn. Más información del algoritmo en https://en.wikipedia.org/wiki/Local_outlier_factor

```

[ ]: clf = LocalOutlierFactor(n_neighbors = 5, novelty = False, contamination = 0.1)
     # use fit_predict para calcular el número de errores al comparar
     # con las etiquetas asignadas a la muestra de entrenamiento.
X = data_CCSS[['DIAS_INVITACION_APER',
    ↳ 'DIAS_APERTURA_ADJU', 'CANTIDAD_ADJUDICADA', 'MONTO_ADJUDICADO']]

```

Se escalan los datos por las diferencias de magnitud entre datos de días y montos.

```

[ ]: X=preprocessing.scale(X)

```

Se estima los puntajes basados en la distancia con 5 vecinos y un valor de densidad mayor a 2 para clasificar un procedimiento como atípico, además se estima un 10% de valores atípicos

```

[ ]: y_pred = clf.fit_predict(X)
     lof_scores = -1 * clf.negative_outlier_factor_
     outlier_5 = lof_scores > 2
     unique, counts = np.unique(outlier_5, return_counts=True)

```

```
#unique, counts
```

```
[ ]: #registra en el DF si es considerado anómalo y el LOF
```

```
data_CCSS['outlier_5'] = outlier_5
```

```
data_CCSS['score'] =lof_scores
```

```
[ ]: #Cantidad de proyecto anómalos con estos valores
```

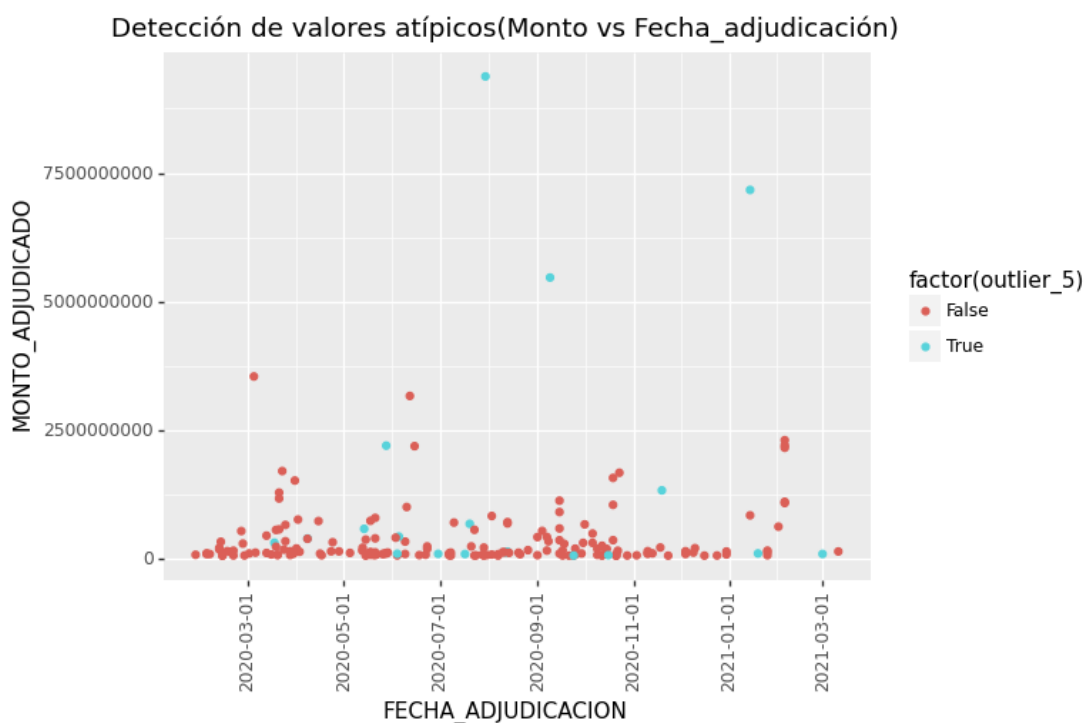
```
outliers= data_CCSS[data_CCSS['outlier_5'].astype(str).str.contains("True")]
```

```
len(outliers)
```

```
[ ]: 19
```

Con el algoritmo LOF se ubican 19 procedimientos con un score mayor a 2 (mayor atípidad)

```
[ ]: (ggplot(data_CCSS, aes('FECHA_ADJUDICACION', 'MONTO_ADJUDICADO', color =  
  ↳'factor(outlier_5)'))  
+ geom_point()+theme(axis_text_x = element_text(angle = 90, hjust = 0.1)))+  
  ↳labs(title = "Detección de valores atípicos(Monto vs Fecha_adjudicación)"))
```



```
[ ]: <ggplot: (8748381097385)>
```

```
[ ]: fig = px.scatter_3d(data_CCSS, x='NUMERO_PROCEDIMIENTO', y='outlier_5',  
  ↳z='MONTO_ADJUDICADO',
```

```
color="score",size="MONTO_ADJUDICADO",title="Grado de Anómalo VS_
↪(MONTO,NUMERO_PROCEDIMIENTO)")
```

```
[ ]: %%capture
#si desea ejecutar y ver el gráfico elimine el comando %%capture
fig.update_layout(
    scene = dict(
        zaxis = dict(nticks=5)))

fig.update_layout(scene = dict(
    xaxis_title='NUMERO_PROCEDIMIENTO',
    yaxis_title='Anómalo',
    zaxis_title='MONTO_ADJUDICADO'),
    width=700,
    margin=dict(r=20, b=10, l=10, t=10))

fig.show()
```

```
[ ]: fig.write_html("ccss.file.expo.html")
```

Un gráfico interactivo y más detallado lo puede visualizar en el siguiente enlace. Puede ver el detalle dándole click a cada punto, valores atípico en el extremo derecho. [Aquí](#)

Los procedimientos de contratación que conforme a este algoritmos se consideran atípicos, con un valor de densidad superior a 2 son los siguientes:

```
[ ]: outliers
```

```
[ ]:
```

	INSTITUCION LICITANTE	ANNIO	...	outlier_5	score
5	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	True	2.798088
11	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	True	2.864522
16	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	True	3.728772
18	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	True	2.008377
21	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	True	2.543753
38	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	True	4.140410
40	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	True	2.722570
43	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	True	2.678074
48	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	True	3.388903
88	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	True	3.317409
121	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	True	2.012796
133	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	True	2.175247
138	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	True	4.751360
142	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	True	6.140561
162	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	True	2.068111
164	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	True	5.879304
168	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	True	4.221059
188	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	True	2.321640

```
190 CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS) 2020 ... True 4.736304
```

```
[19 rows x 28 columns]
```

```
[ ]:
```

6 Atipicidad por algoritmo de Isolation Forest

Se aplica a la variable monto adjudicado el algoritmo de Isolation Forest para determinar valores atípico, con un total de 50 estimadores. Más información en https://en.wikipedia.org/wiki/Isolation_forest

```
[ ]: model=IsolationForest(n_estimators = 50, max_samples = 'auto', \
    contamination = float(0.1), max_features = 1.0)
    model.fit(X)
```

```
[ ]: IsolationForest(behaviour='deprecated', bootstrap=False, contamination=0.1,
    max_features=1.0, max_samples='auto', n_estimators=50,
    n_jobs=None, random_state=None, verbose=0, warm_start=False)
```

Se asigna un valor de 1 a los datos típicos y un -1 a los atípicos conforme al puntaje otorgado por el algoritmo, conforme el ejemplo de seguido.

```
[ ]: data_CCSS['PUNTAJE'] = model.decision_function(X)
    data_CCSS['ATIPICO'] = model.predict(X)
    data_CCSS[["MONTO_ADJUDICADO", "PUNTAJE", "ATIPICO"]].head(6)
```

```
[ ]: MONTO_ADJUDICADO  PUNTAJE  ATIPICO
0      5.425363e+07  0.114899      1
1      6.860413e+07  0.156472      1
2      7.768400e+07  0.148557      1
3      1.150780e+08  0.163765      1
4      3.250000e+08  0.132697      1
5      3.847753e+08  0.022809      1
```

Los siguientes procedimientos son clasificados atípicos por el algoritmo. Son un mayor número que con el algoritmo LOF y algunos son clasificados atípicos por ambos algoritmos.

```
[ ]: atipico= data_CCSS.loc[data_CCSS['ATIPICO'] == -1]
    atipico_index = list(atipico.index)
    atipico
```

```
[ ]: INSTITUCION LICITANTE  ANNIO  ...  PUNTAJE  ATIPICO
28  CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)  2020  ... -0.096166      -1
38  CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)  2020  ... -0.167054      -1
40  CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)  2020  ... -0.244271      -1
49  CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)  2020  ... -0.080094      -1
61  CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)  2020  ... -0.024722      -1
```

122	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	-0.000174	-1
132	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	-0.041838	-1
138	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	-0.107006	-1
144	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	-0.028322	-1
146	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	-0.017130	-1
162	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	-0.129628	-1
164	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	-0.257411	-1
168	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	-0.081008	-1
177	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	-0.018689	-1
178	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	-0.021522	-1
179	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	-0.068425	-1
180	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	-0.054307	-1
181	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	-0.078139	-1
188	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	-0.179330	-1
190	CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)	2020	...	-0.123416	-1

[20 rows x 30 columns]

```
[ ]: fig = px.scatter_3d(atipico, x='MONTO_ADJUDICADO', y='ATIPICO',
    ↪z='NUMERO_PROCEDIMIENTO',
    color="PUNTAJE",size="MONTO_ADJUDICADO",title="Procedimientos
    ↪atípicos conforme a Isolation Forest")
```

```
[ ]: %%capture
#si desea ejecutar y ver el gráfico elimine el comando %%capture
fig.update_layout(
    scene = dict(
        zaxis = dict(nticks=5)))

fig.update_layout(scene = dict(
    xaxis_title='MONTO_ADJUDICADO',
    yaxis_title='ATIPICO',
    zaxis_title='NUMERO_PROCEDIMIENTO'),
    width=700,
    margin=dict(r=20, b=10, l=10, t=10))

fig.show()
```

```
[ ]: fig.write_html("ccss2.file.isolatio.html")
```

Un gráfico interactivo y más detallado lo puede visualizar en el siguiente enlace. Puede ver el detalle dándole click a cada punto, valores atípico en el extremo derecho. [Aquí](#)

7 Procedimientos que resultaron atípicos en ambos algoritmos de LOF y de Isolation Forest

La siguiente lista de procedimientos son de aquellos que resultaron atípicos usando ambos algoritmos.

```
[ ]: Coincidentes_ISO_LOF = data_CCSS.query("ATIPICO == -1 & score >= 2")
Coincidentes_ISO_LOF
```

```
[ ]:
      INSTITUCION LICITANTE  ANNIO  ...  PUNTAJE ATIPICO
38  CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)  2020  ... -0.167054      -1
40  CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)  2020  ... -0.244271      -1
138 CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)  2020  ... -0.107006      -1
162 CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)  2020  ... -0.129628      -1
164 CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)  2020  ... -0.257411      -1
168 CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)  2020  ... -0.081008      -1
188 CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)  2020  ... -0.179330      -1
190 CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)  2020  ... -0.123416      -1
```

[8 rows x 30 columns]

7.1 Detalle en la adquisición de las pruebas PCR

Haciendo un detalle del costo de adquisición de las pruebas PCR tenemos los siguientes costos unitarios.

```
[ ]: data_PCR=pd.read_excel(path+"PCR.xlsx")
```

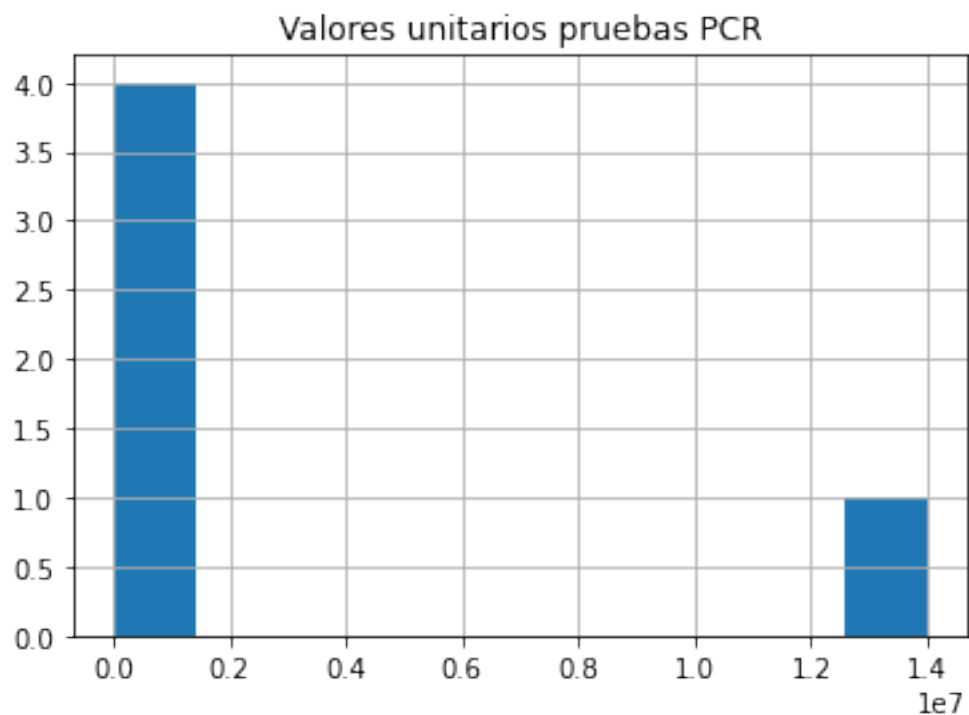
```
[ ]: data_PCR
```

```
[ ]:
      INSTITUCION LICITANTE  ...
SUBPARTIDA DE GASTO
0  CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)  ...  UTILES Y MATERIALES MEDICO,
HOSPITALARIO Y DE ...
1  CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)  ...  UTILES Y MATERIALES MEDICO,
HOSPITALARIO Y DE ...
2  CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)  ...  UTILES Y MATERIALES MEDICO,
HOSPITALARIO Y DE ...
3  CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)  ...  UTILES Y MATERIALES MEDICO,
HOSPITALARIO Y DE ...
4  CAJA COSTARRICENSE DE SEGURO SOCIAL (CCSS)  ...  UTILES Y MATERIALES MEDICO,
HOSPITALARIO Y DE ...
```

[5 rows x 24 columns]

```
[ ]: #se transforma en numerica la variable de monto
data_PCR.COSTO_UNITARIO=pd.to_numeric(data_PCR.COSTO_UNITARIO)
```

```
[ ]: data_PCR.COSTO_UNITARO.hist()  
plt.title('Valores unitarios pruebas PCR')  
plt.show()
```



Se ubica un valor absolutamente atípico en el costo unitario de las pruebas PCR en una de las contrataciones. Corresponde al procedimiento 2020CD-000238-5101 y podría derivarse a errores de registro de la información.

Para acceder a los datos usados en el presente análisis puede descargarlos de [aquí](#)