

R Wrapper for Python



\$ whoami

Valery Calderón Briz

@valerybriz

Ingeniera en Telecomunicaciones

Software Engineer en Prescripto

Co-organizadora de PyLadies CDMX

Breakout Detection

Este paquete desarrollado en el **Lenguaje R por Twitter** sirve para detectar cambios o anomalías en una serie de datos respecto al tiempo.

Pueden encontrar más información del paquete en:

<http://bit.ly/bkoutinthewild>

¿Cómo funciona el Breakout Detection?

Usualmente las anomalías o breakouts están caracterizadas por dos estados que se mantienen y un período de transición intermedia.

Para detectar esto el paquete utiliza una técnica de estadísticas llamada **E-Divisive with Medians (EDM)** que es utilizada ampliamente para detectar cambios de nivel o saltos.

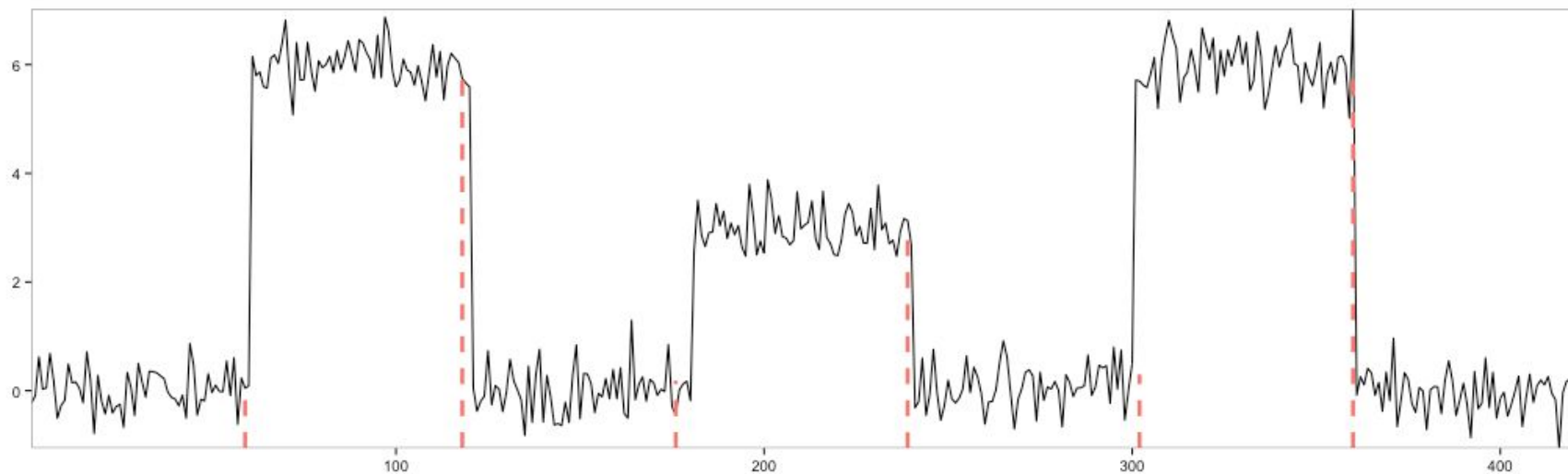
Documentación en:

<http://bit.ly/edmBreakingBad>

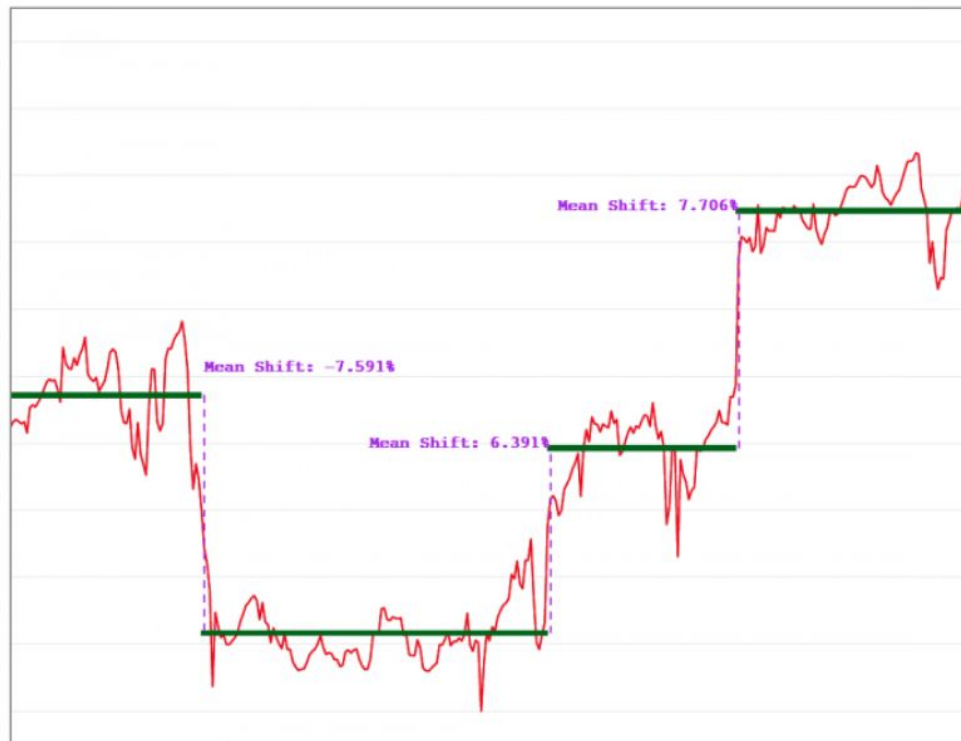
¿Cómo funciona el EDM?

- Detección de Divergencia
 - Mean Shift
 - Ramp Up
- Detección de cambios en la distribución

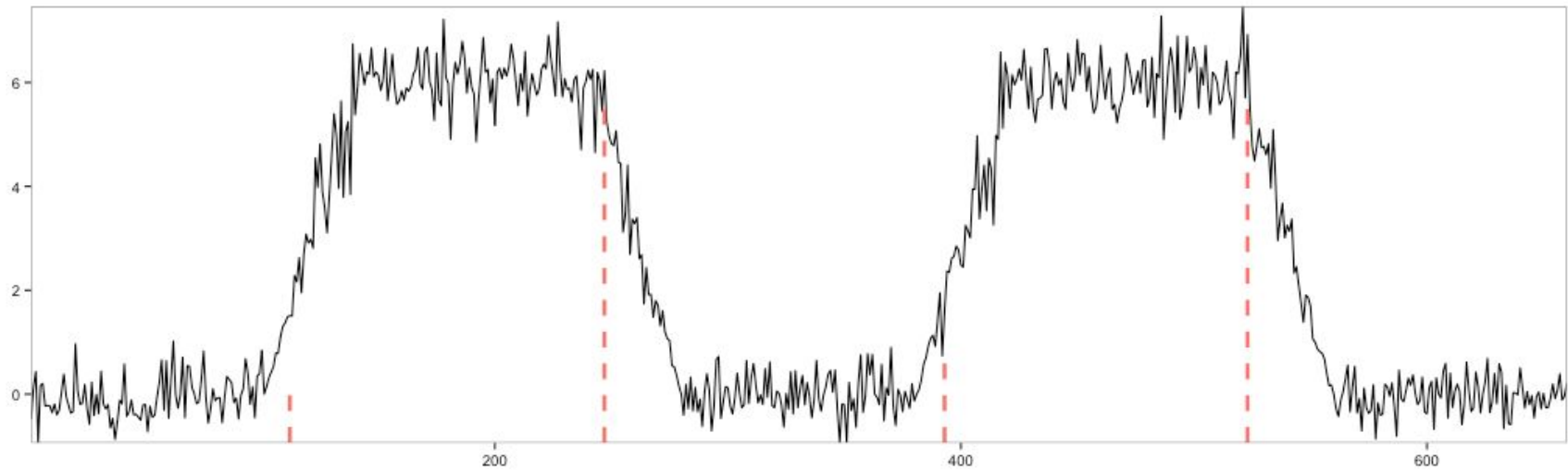
Mean Shift



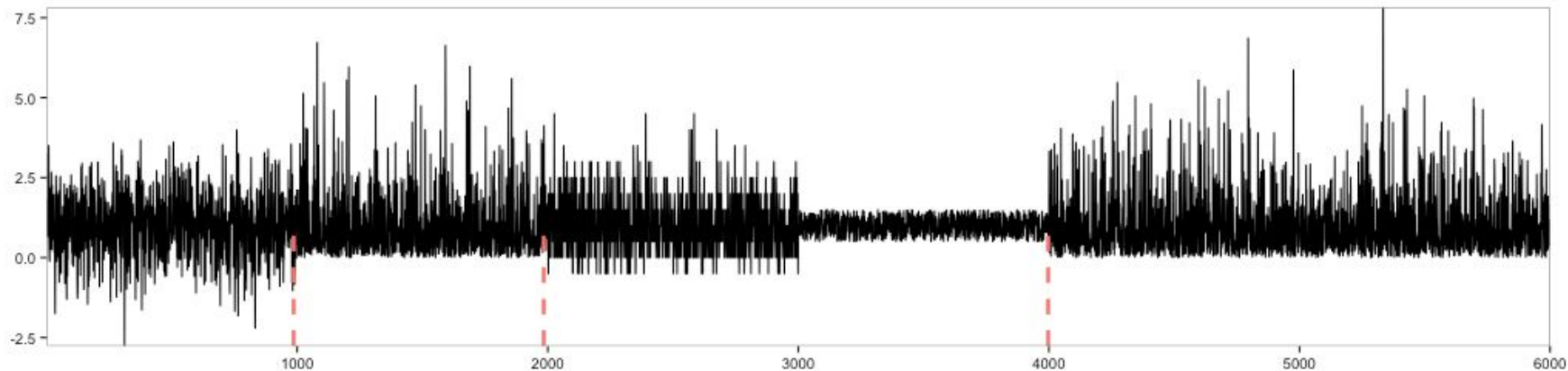
Mean Shift



Ramp Up

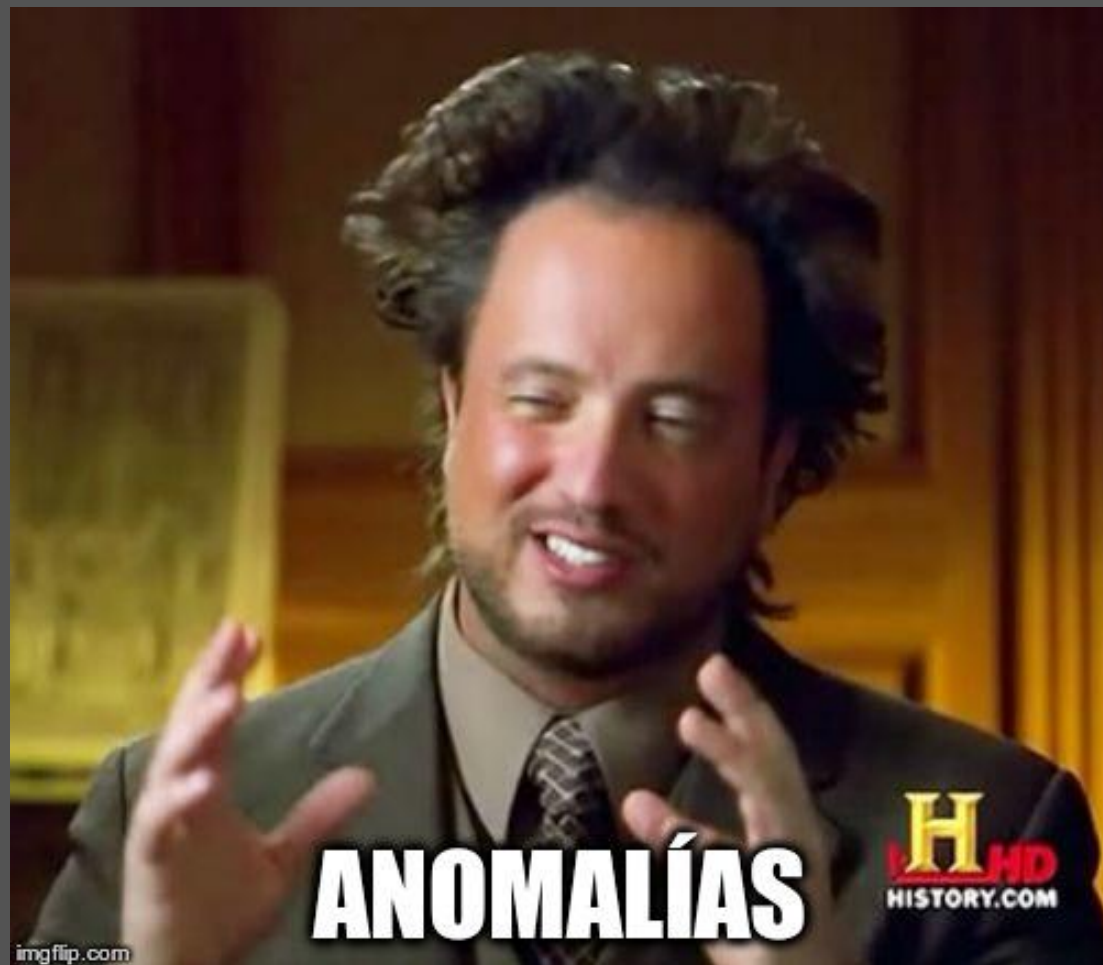


Cambios en la distribución



Ventajas

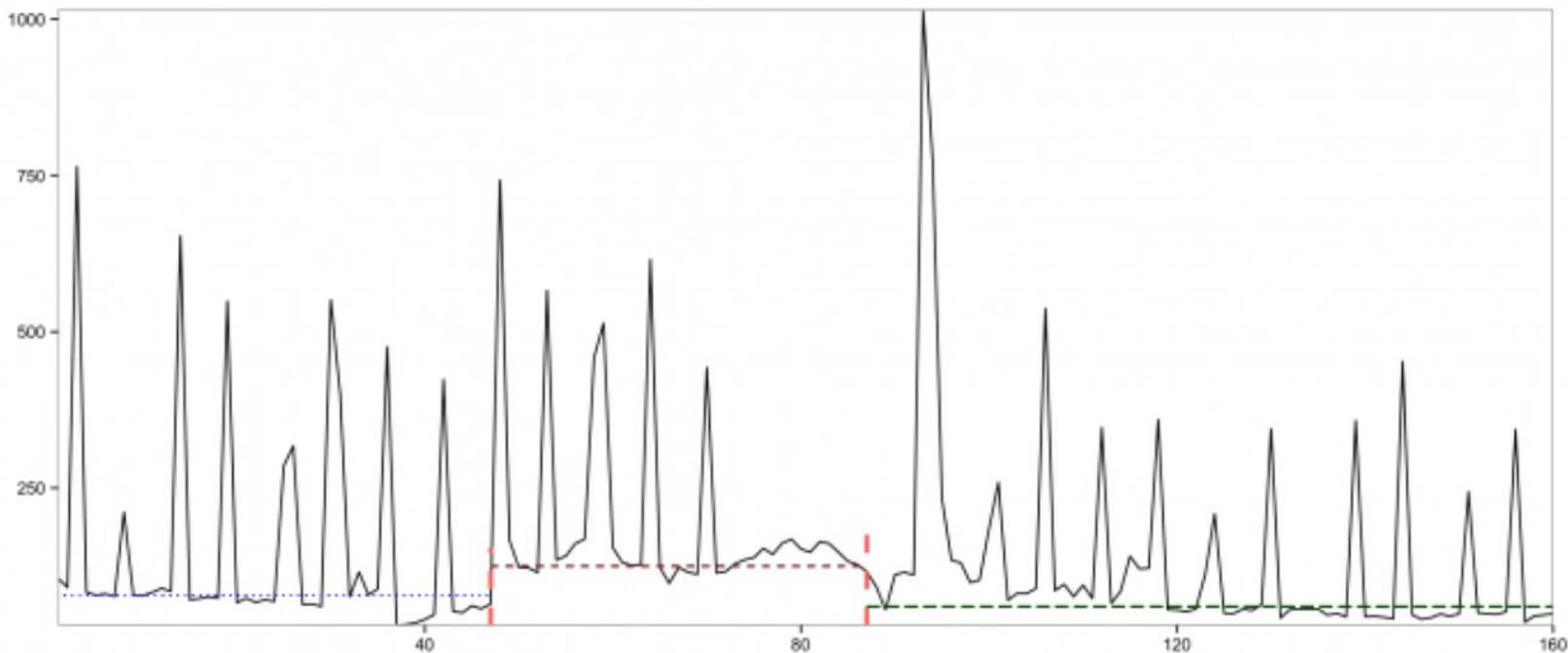
- Es robusto en presencia de anomalías.
- EDM no es paramétrico.
- Es más rápido que otros algoritmos.



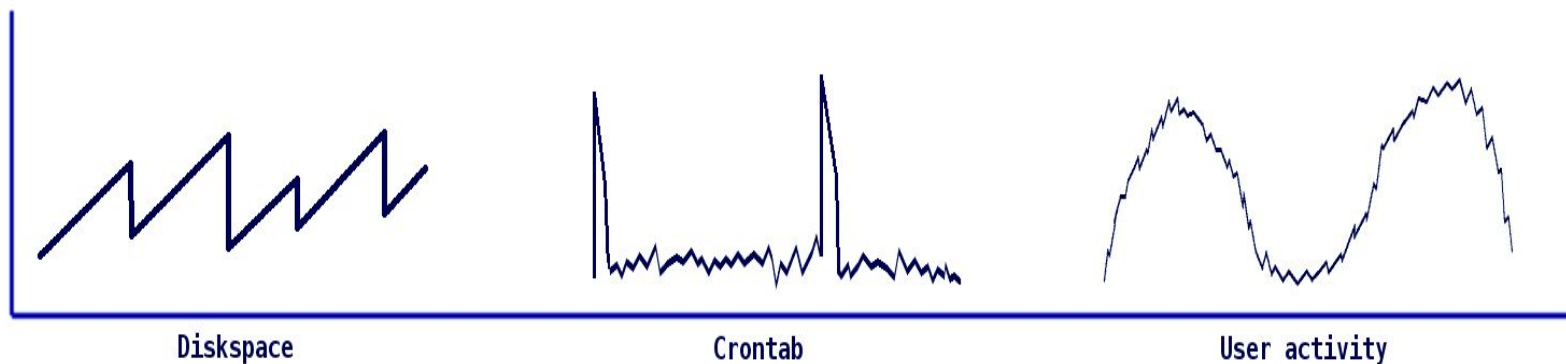
imgflip.com

Robusto en presencia de anomalías

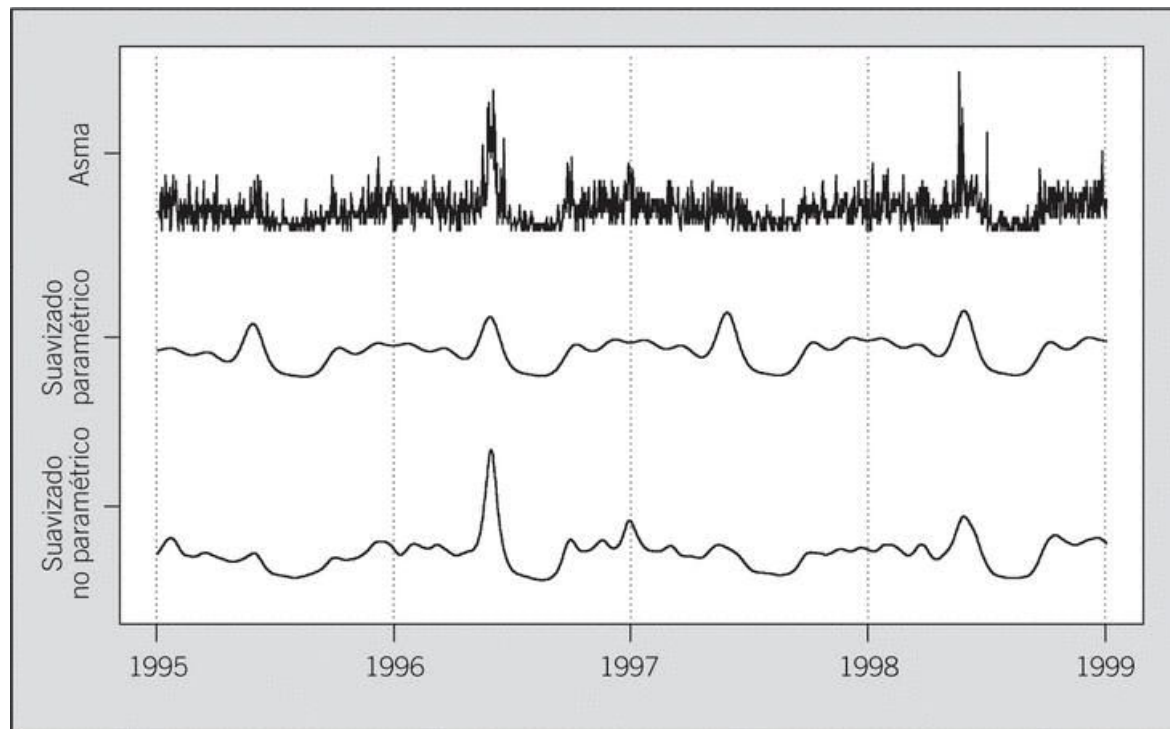
Las anomalías no son lo mismo que un Breakout



Robusto en presencia de anomalías



EDM no es paramétrico



3.5x más rápido que otros algoritmos

De detección de Breakouts

Data Points	E-Divisive	EDM
6,000	29 Min.	24 Seg.
600	7 Seg.	1 Seg.

Fuente: <https://anomaly.io/anomaly-detection-using-twitter-breakout/>

Desventajas

- Se debe combinar con otros algoritmos si se necesita descartar breakouts específicos.
- Es necesario encontrar la combinación correcta de parámetros dependiendo de la serie de datos.

Las variables del algoritmo

- 1 Data: Serie de datos que serán analizados.
- 2 Min.size: El valor mínimo de observaciones entre puntos de cambio.
- 3 Method: "amoc" (al menos un cambio) o "multi" (múltiples cambios).
- 4 Degree: El grado de regresión polinomial.
- 5 Beta: Constante para un futuro control en la penalización.
- 6 Plot: Este indica si se generará un Ploteo luego de la ejecución.

Instalación del paquete



```
> install.packages("devtools")  
> devtools::install_github("twitter/BreakoutDetection")  
> library(BreakoutDetection)
```

Código en R



```
library(BreakoutDetection)

Detect <- function(tsdata, min, met, deg )
{
  res = breakout(tsdata, min.size=min, method=met,
    beta=.001, degree=deg, plot=FALSE)
  return(res)
}
```

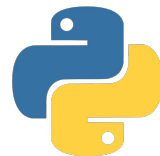
Requerimientos en Python



rpy2

<https://rpy2.readthedocs.io>

Importando desde R



```
minsize = 30
method = 'multi'
degree = 0
with open('Codigo.R') as code:
    rcode = os.linesep.join(code.readlines())
    wrapper = SignatureTranslatedAnonymousPackage(rcode, "Codigo")
```

Ejecutando la función



```
result = wrapper.Detect( FloatVector(mydata), minsize, method, degree)
```

```
print (result)
```

```
>>>>>
```

```
$loc
```

```
[1] 74 116 147 177 209
```

```
$time
```

```
[1] 0.013
```

```
$pval
```

```
[1] NA
```

Obteniendo los datos relevantes



```
if len(result) > 0 :  
    for val in result[0]:  
        print (val)
```

```
>>>>>
```

```
74
```

```
116
```

```
147
```

```
177
```

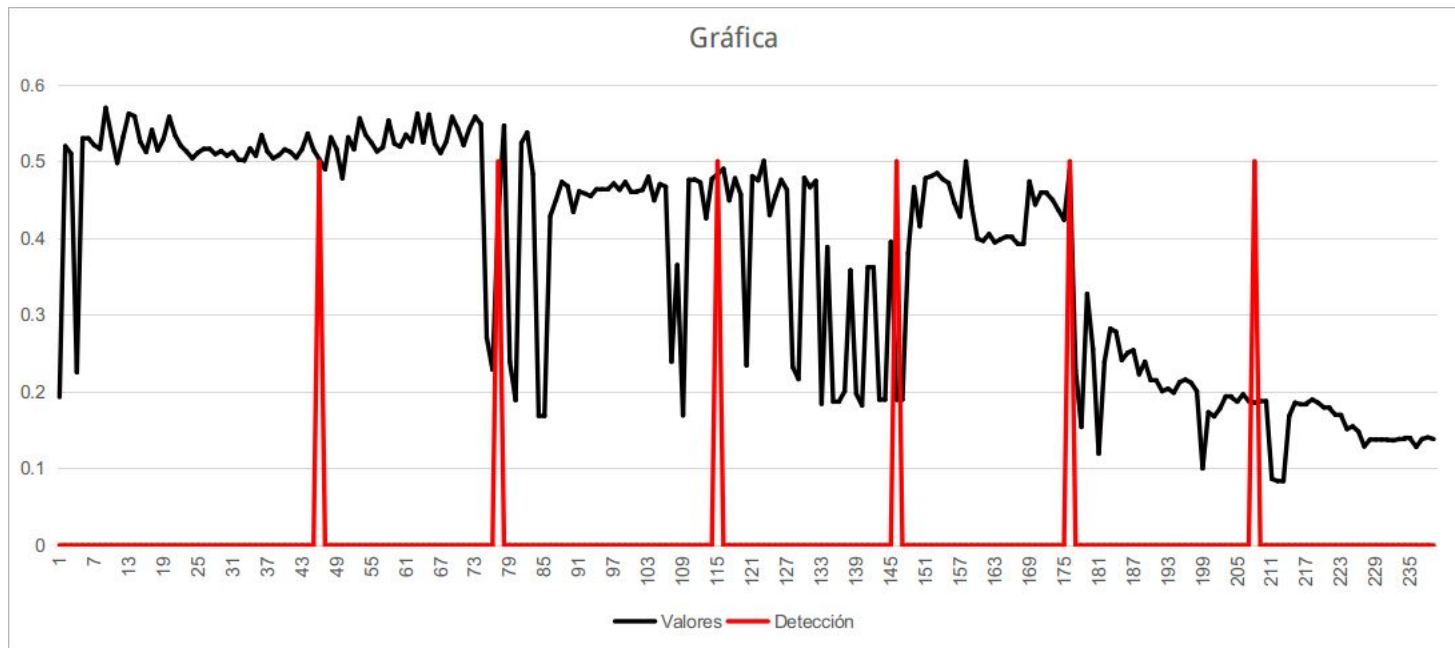
```
209
```

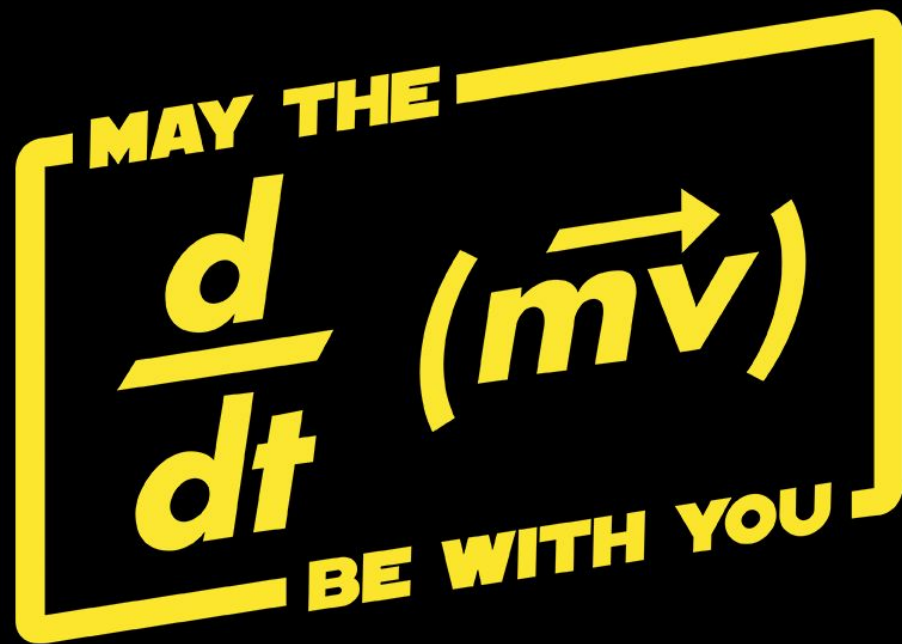
Caso Práctico: Sensor de combustible



Resultados

min.size=30, method='multi', beta=.001, degree=0





Preguntas / comentarios

Repositorio:

https://github.com/pyladies-nwuk/Python_meets_R

@valerybriz



pyladies