
PRONOSTICO DE CAUDALES EN EL SITIO DE PRESA DEL PROYECTO HIDROELÉCTRICO ITUANGO PARA ALERTAS TEMPRANAS DE LOS FRENTES DE TRABAJO Y LAS COMUNIDADES AGUAS ABAJO

Juliana Vélez Ochoa

INTRODUCCIÓN

En este documento se describe paso a paso parte del análisis de la serie de caudales medios diarios del río Cauca a la altura del puente Pescadero cerca del corregimiento del Valle, municipio de Toledo.

La estación allí ubicada cuenta con 32 años de registro desde 1984 a 2016.

El objetivo de este informe es mostrar los resultados utilizando el lenguaje de programación interpretado Python.

Para iniciar el trabajo se definió trabajar con PyCharm, que es un IDE o entorno de desarrollo integrado multiplataforma utilizado para desarrollar en el lenguaje de programación Python.

Objetivos principales para el desarrollo de parte del análisis de la serie son:

- Importar la serie a Pycharm.
- Graficar la serie de caudales.
- Obtener el análisis de frecuencia (percentil de la serie) y graficar.
- Obtener Histogramas.
- A partir de la definición de una ventana tiempo de la serie, obtener:
 - ✓ Media de la serie.
 - ✓ Mediana de la serie.
 - ✓ Desviación estándar de la serie.
 - ✓ Intercuartil de la serie.
 - ✓ Percentil 90 de la serie.
 - ✓ Percentil 10 de la serie.
 - ✓ Variabilidad de la serie
 - ✓ Índice de Jay - Kendall
- Se utilizará Test Mann – Kendall, el cual se utiliza para analizar los datos recopilados a lo largo del tiempo para aumentar o disminuir constantemente las tendencias en los valores. Con este test se pretende definir la tendencia de la serie objeto de este trabajo

IMPORTACIÓN DE SERIE Y GRÁFICA OBTENIDA

La serie de caudales medios diarios de la estación Pescadero se importó a Pycharm, directamente de un archivo Excel.

Una vez importada se estableció el código correspondiente para obtener gráficamente la serie.

```
PyCharm
Is VCS Window Help

Tarea1.py x
61
62 import ...
65
66 Archivo = pd.read_excel("C://Users//32295553//Desktop//Ejemplo.xlsx", "Hoja1")
67
68 plt.plot(Archivo["Fecha"], Archivo["Pescadero"])
69 plt.ylabel(u"Caudal (m3/s)")
70 plt.xlabel(u"Tiempo (años)")
71 plt.show()
72
```

Figura 1 Código para graficar la serie

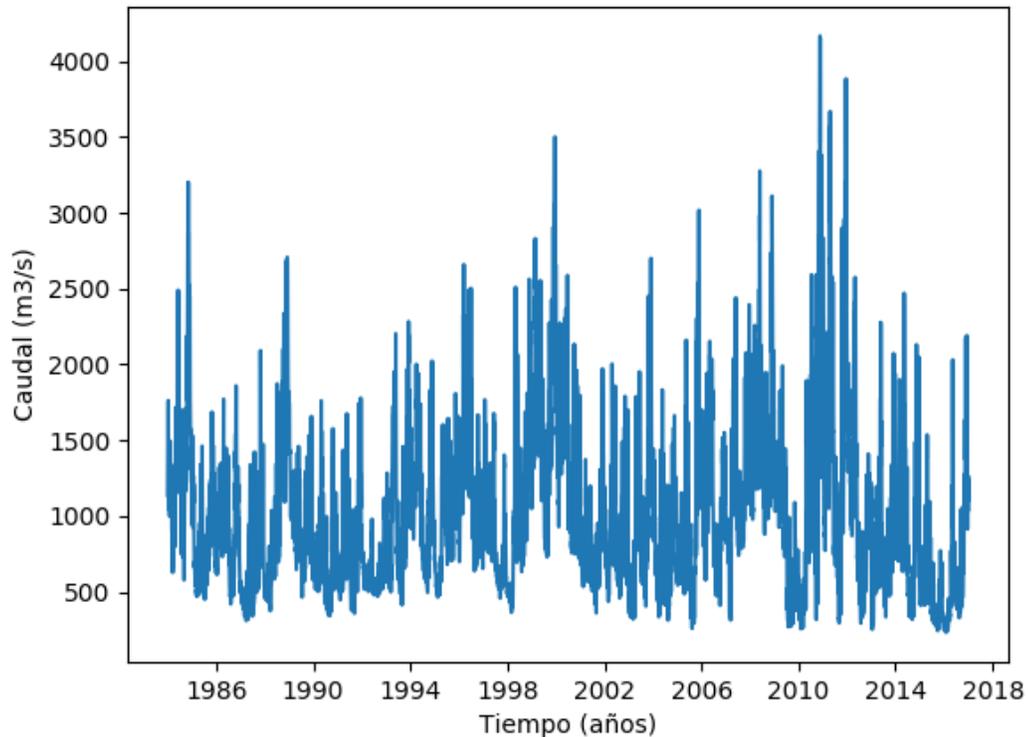


Figura 2 Gráfica se la serie de la estación Pescadero (1984 – 2016)

PERCENTILES (ANÁLISIS DE FRECUENCIA)

Con el fin de analizar la frecuencia de los caudales de la serie, se cálculo el percentil de esta.

Esta medida se utiliza para que una vez los datos se ordenen de mayor a menos, se halle el valor de la variable por debajo del cual se encuentra un porcentaje dado de datos en un grupo de datos.

Para hallar el percentil se utilizó el siguiente código.

```
Tarea1.py x  
72  
73 #01_Percentiles  
74 P10=np.percentile(Archivo["Pescadero"],0.5)  
75 P=np.percentile(Archivo["Pescadero"],range(100))  
76 plt.plot(P,range(100),'blue')  
77 plt.show()
```

Figura 3 Código percentil

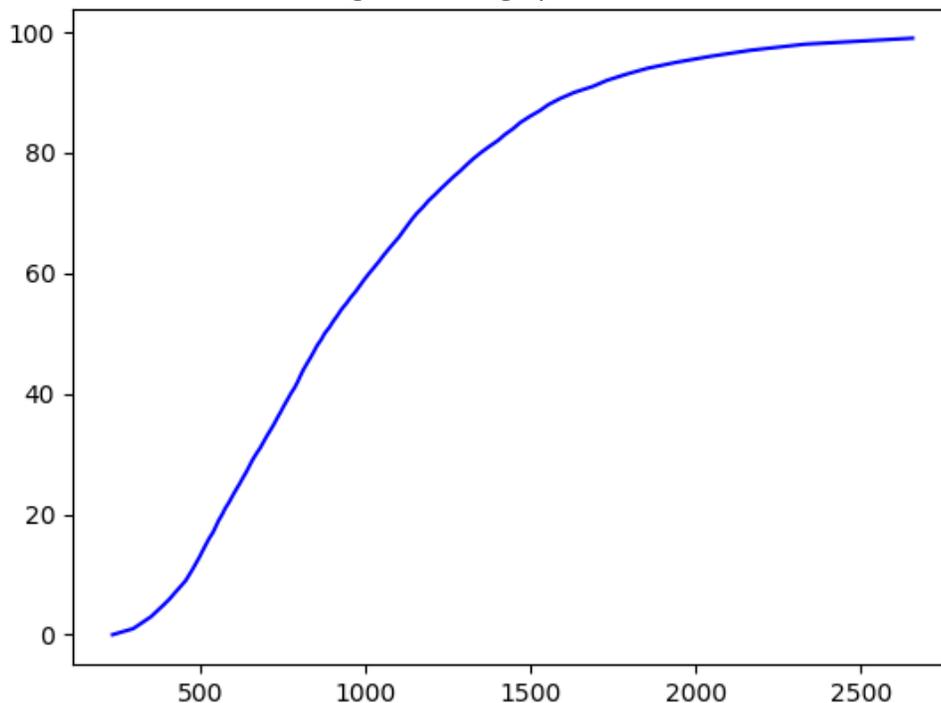


Figura 4 Análisis de frecuencia

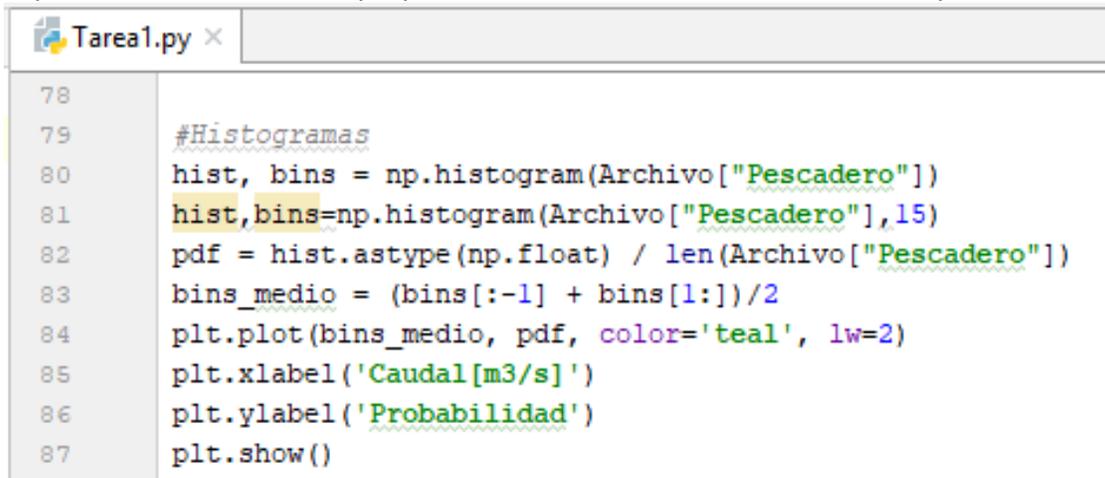
De la gráfica anterior se observa que la probabilidad de excedencia de caudales del orden de 500 m³/s es de un 90% aproximadamente, es decir que la mayor parte del tiempo el Cauca en la zona de Pescadero se sobrepasa este caudal.

Adicionalmente, se observa que los caudales medios del río están por los lados de 900 – 1000 m³/s y que solo un 10% en los 32 años de registro los caudales alcanzados han sido del orden 1600 m³/s.

HISTOGRAMA

El siguiente paso consistió en obtener un histograma de la serie.

Este es una representación gráfica de una variable en forma de barras, donde la superficie de cada barra es proporcional a la frecuencia de los valores representados.



```
78
79 #Histogramas
80 hist, bins = np.histogram(Archivo["Pescadero"])
81 hist, bins=np.histogram(Archivo["Pescadero"], 15)
82 pdf = hist.astype(np.float) / len(Archivo["Pescadero"])
83 bins_medio = (bins[:-1] + bins[1:])/2
84 plt.plot(bins_medio, pdf, color='teal', lw=2)
85 plt.xlabel('Caudal [m3/s]')
86 plt.ylabel('Probabilidad')
87 plt.show()
```

Figura 5 Código para hallar histograma

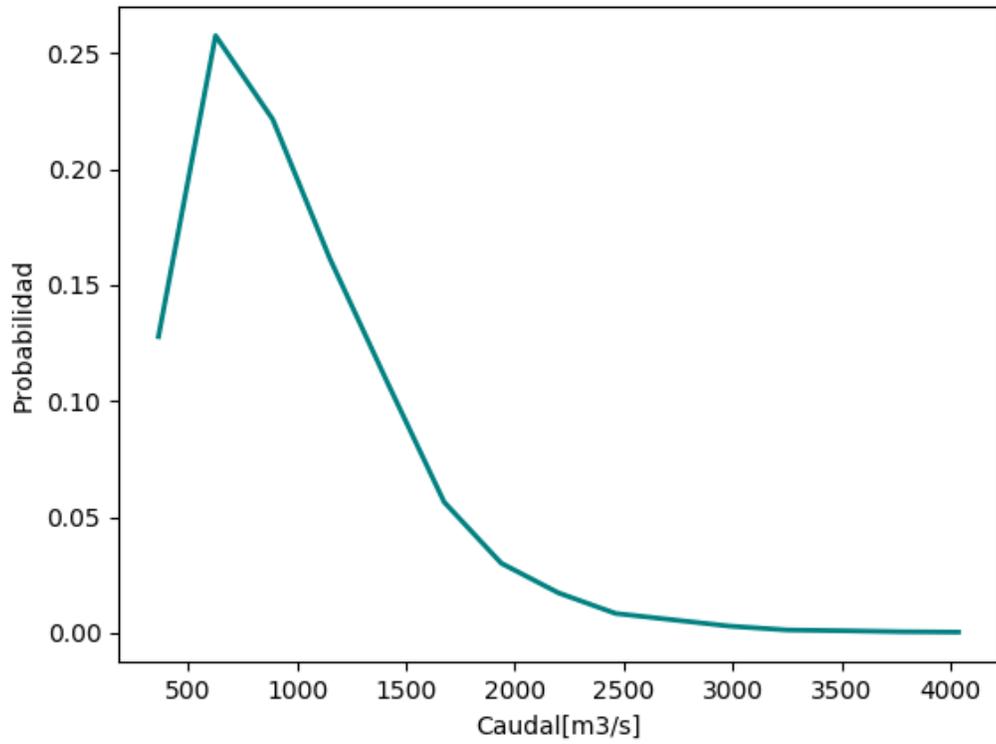


Figura 6 Histograma de la serie de la estación Pescadero

MEDIA, MEDIANA, INTERCUANTIL, PERCENTIL 90 Y 10, VARIABILIDAD DE UNA VENTANA DE LA SERIE



```
Tarea1.py x
90 Matriz_pdf=np.zeros((len(Archivo["Pescadero"]) - Ventana, len(hist)))
91 Media_Serie = np.zeros((len(Archivo["Pescadero"]) - Ventana))
92 Mediana_Serie = np.zeros((len(Archivo["Pescadero"]) - Ventana))
93 std_Serie = np.zeros((len(Archivo["Pescadero"]) - Ventana))
94 IQR_Serie = np.zeros((len(Archivo["Pescadero"]) - Ventana))
95 Asim_Serie = np.zeros((len(Archivo["Pescadero"]) - Ventana))
96 YK_Serie = np.zeros((len(Archivo["Pescadero"]) - Ventana))
97 P90_Serie = np.zeros((len(Archivo["Pescadero"]) - Ventana))
98 P10_Serie = np.zeros((len(Archivo["Pescadero"]) - Ventana))
99 Matriz_pdf_Est = np.zeros((len(Archivo["Pescadero"])-Ventana, len(hist)))
100 #Intervalos
101 for i in range(0, len(Archivo["Pescadero"])-Ventana):
102     Serie = Archivo["Pescadero"][i: i + Ventana]
103     hist_Serie, bins_Serie = np.histogram(Serie, bins)
104     pdf_Serie = hist_Serie.astype(np.float) / len(Serie)
105     Matriz_pdf[i, :] = pdf_Serie
106     Matriz_pdf_Est[i, :] = pdf
107     Media_Serie[i] = Serie.mean()
108     Mediana_Serie[i] = np.median(Serie)
109     P10_Serie[i] = np.percentile(Serie, 10)
110     P90_Serie[i] = np.percentile(Serie, 90)
111     std_Serie[i] = np.std(Serie)
112     IQR_Serie[i] = np.percentile(Serie, 75) - np.percentile(Serie, 25)
113     YK_Serie[i] = (np.percentile(Serie, 75)-2*np.percentile(Serie, 50)+np.percentile(Serie, 25))/(np.percentile(Serie, 75)
114
115     plt.imshow(Matriz_pdf.T, origin='lower', aspect='auto', vmin=0, vmax=0.5)
116     plt.colorbar()
117     plt.show()
```

Figura 7 Código utilizado (Media, Mediana, P10, P90, Yule – Kendall, DesvSta)

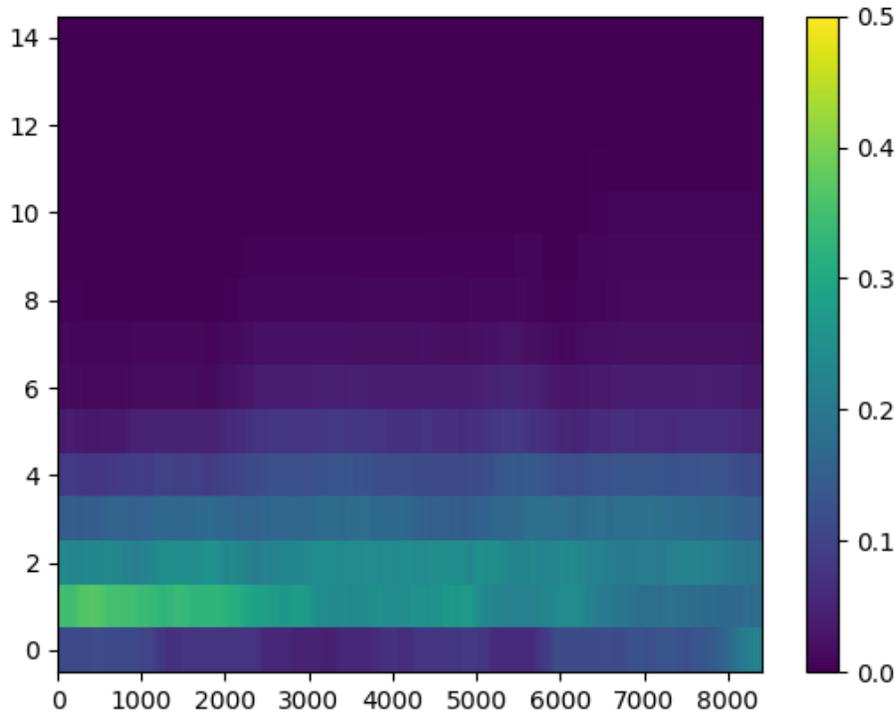


Figura 8 Variabilidad de la ventana de la serie definida

**Conclusiones al
respecto?**

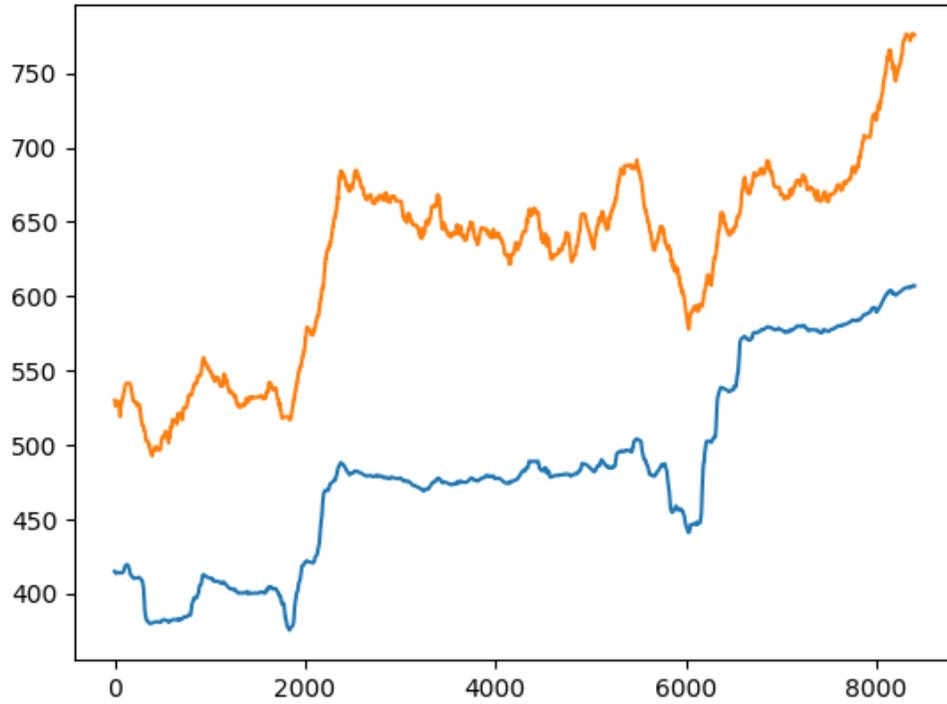


Figura 9 Desviación estándar e intercuantil de la ventana definida de la serie

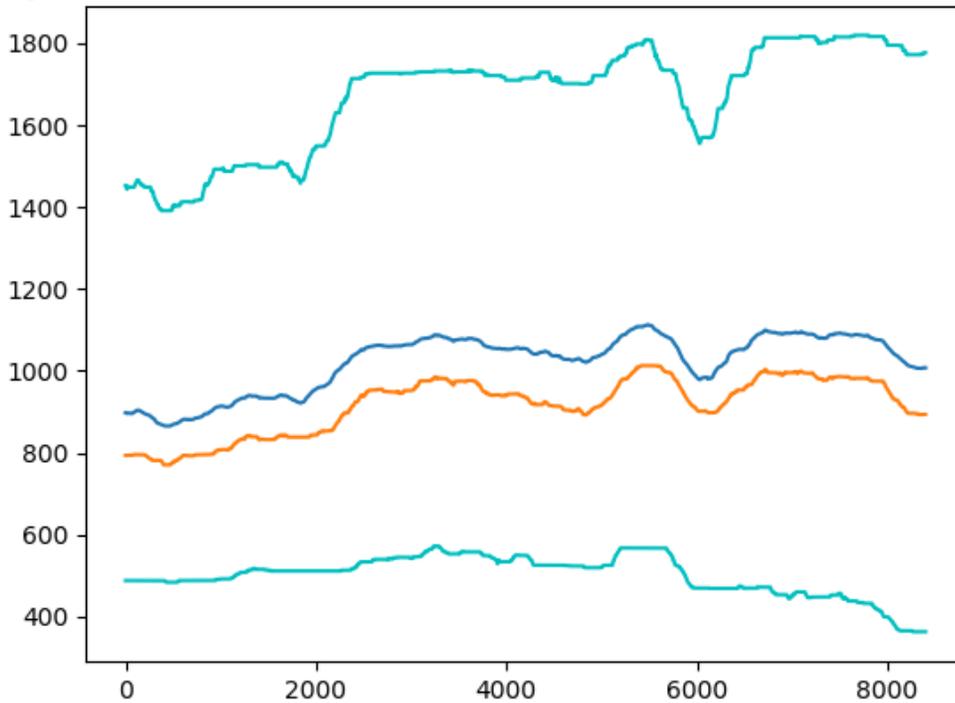


Figura 10 Media, Mediana, Percentil 90 y 10 de la ventana definida de la serie

**Algo que decir de la
física del problema?**

De la gráfica anterior se observa que, en el intervalo estudiado la media corresponde a caudales del orden de $1000 \text{ m}^3/\text{s}$, la mediana del orden de $800 - 900 \text{ m}^3/\text{s}$, el percentil 90 de caudales entre 1400 y $1800 \text{ m}^3/\text{s}$ y el percentil 10 de caudales entre 500 y $300 \text{ m}^3/\text{s}$.

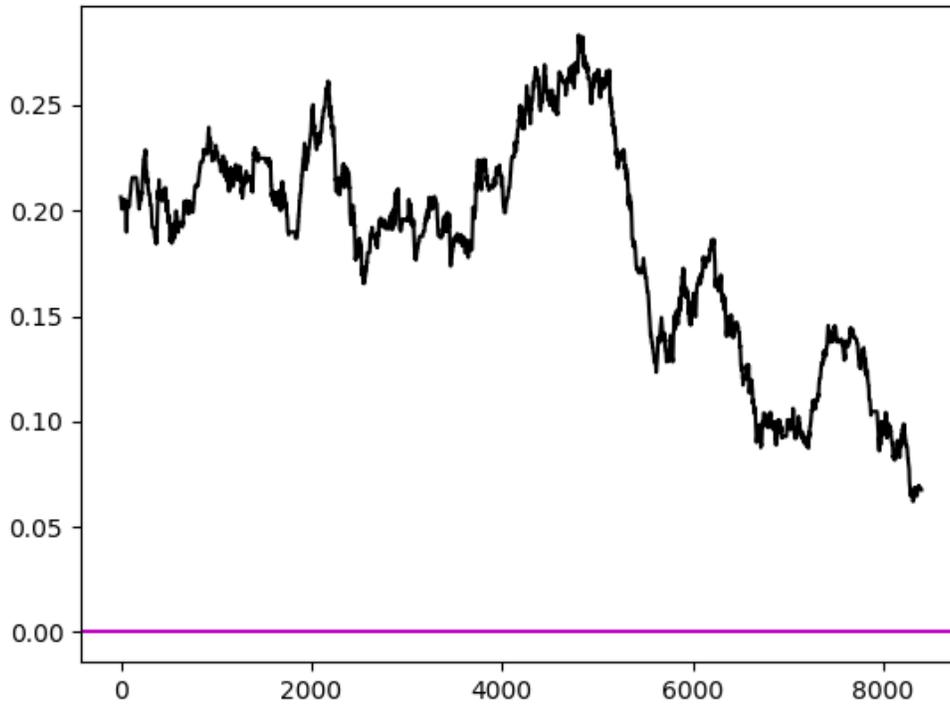


Figura 11 Índice de Yule – Kendall

El Índice de Yule – Kendall, es una estandarización de la muestra.

Es una medida simétrica de asociación que se aplica a variables nominales y ordinales dicotómicas. El coeficiente no puede valer > 1 o < -1 . En este caso se presentan valores negativos.

TEST MANN – KENDALL

Este Test se puede usar para encontrar tendencias de los datos. Es importante contar con buena información para que el Test encuentre con mayor probabilidad una tendencia. Cuantos más puntos de datos tenga, es más probable que el Test encuentre una tendencia verdadera.



```
} #TEST DE TENDENCIA DE MANN-RENDALL
def mk_test(x, alpha = 0.05): #by MICHAEL SCHRAMM
}
    """
    Input:
    | x: a vector of data
    | alpha: significance level (0.05 default)
    Output:
    | trend: tells the trend (increasing, decreasing or no trend)
    | h: True (if trend is present) or False (if trend is absence)
    | p: p value of the significance test
    | z: normalized test statistics

    Examples
    -----
    | x = np.random.rand(100)
    | trend,h,p,z = mk_test(x,0.05)
}
    """
n = len(x)

# calculate S
s = 0
} for k in range(n-1):
    | for j in range(k+1,n):
}     | s += np.sign(x[j] - x[k])

# calculate the unique data
unique_x = np.unique(x)
g = len(unique_x)
```

????



```
# calculate the var(s)
if n == g: # there is no tie
    var_s = (n*(n-1)*(2*n+5))/18
else: # there are some ties in data
    tp = np.zeros(unique_x.shape)
    for i in range(len(unique_x)):
        tp[i] = sum(unique_x[i] == x)
    var_s = (n*(n-1)*(2*n+5) + np.sum(tp*(tp-1)*(2*tp+5)))/18

if s>0:
    z = (s - 1)/np.sqrt(var_s)
elif s == 0:
    z = 0
elif s<0:
    z = (s + 1)/np.sqrt(var_s)

# calculate the p_value
p = 2*(1-norm.cdf(abs(z))) # two tail test
h = abs(z) > norm.ppf(1-alpha/2)

if (z<0) and h:
    trend = 'decreasing'
elif (z>0) and h:
    trend = 'increasing'
else:
    trend = 'no trend'

return trend, h, p, z
```

????