

# Gabriel Tapia Z.

Bachiller en Economía

Economía | Finanzas | Data Science

LinkedIn: <https://www.linkedin.com/in/-gabrieltapiaz/>

## Sobre mí:

Soy Bachiller en Economía con experiencia en Análisis de Datos, Estudios Económicos y Análisis Financiero. Logré la clasificación académica de Quinto Superior y cuento con formación técnica en Econometría, Finanzas, Gestión de Riesgos e Inversiones, y en lenguajes de programación y software como Python, R, Excel, SQL, Power BI y Tableau. Poseo habilidades destacadas en comunicación, resolución de problemas y liderazgo. Estoy orientado a la optimización del trabajo en equipo y comprometido con el aprendizaje continuo.

## Proyectos Destacados

### 1) Aproximación econométrica sobre el uso de los grupos de Incoterms en las importaciones peruanas de 2021

**Fecha:** 2022 | **Indicadores Analizados:** Incoterms del Grupo C y D y del Grupo E y F de las importaciones peruanas del 2021.

**1.1. Objetivo:** Realizar un análisis exploratorio de los factores que influyen en el uso de diferentes grupos de Incoterms en el comercio exterior peruano.

**1.2. Metodología:**

- Implementación de un modelo de regresión Probit binaria para examinar la adopción de Incoterms en las importaciones peruanas, diferenciando entre el grupo C y D, y el grupo E y F, lo que permite identificar de manera precisa la influencia de diversas variables en la elección de estos conjuntos de términos comerciales.
- Revisión de la literatura científica para realizar el uso de la metodología adecuada.
- Formulación de la ecuación econométrica.
- Recopilación de variables relevantes y procesamiento de una base de datos con 706,862 registros de importación del Perú correspondientes al año 2021.
- Estimación econométrica mediante software econométrico (EViews).
- Análisis de los resultados y obtención de conclusiones.

**1.3. Herramientas:** Información pública de las Bases de Datos DBF de los Regímenes Definitivos de Comercio Internacional de la Sunat, Python, Microsoft Excel, EViews.

**1.4. Resultados:**

Los resultados evidencian el empleo de los grupos de Incoterms en las importaciones peruanas de 2021 mediante la estimación de un modelo de regresión Probit Binaria, en el cual se define la variable dependiente (CE) para diferenciar entre el uso de Incoterms del grupo C y D (CE = 1) y del grupo E y F (CE = 0). La metodología se sustenta en una muestra considerable de 706,862 registros y utiliza variables explicativas que abarcan aspectos geográficos (mediante indicadores dicotómicos que señalan la procedencia de Sudamérica o Norteamérica), la distancia promedio

en kilómetros desde el país de origen y el ratio del peso sobre el valor total de la mercadería, permitiendo identificar de forma precisa cómo cada factor incide en la probabilidad de emplear un determinado grupo de Incoterms. Los hallazgos revelan que un mayor valor de la distancia se relaciona con una probabilidad reducida de optar por los Incoterms del grupo C y D, lo cual sugiere que en operaciones con mayores distancias, posiblemente se prefiere mitigar riesgos mediante condiciones contractuales que favorecen el grupo E y F; asimismo, el ratio peso/valor emerge como un determinante significativo, ya que un mayor peso relativo frente al valor incrementa los costos logísticos y, en consecuencia, incentiva la selección de términos que repartan de forma distinta los riesgos y responsabilidades en el transporte. Además, la relevancia de las variables dicotómicas geográficas evidencia que el origen de las importaciones juega un papel crucial, orientando la elección del Incoterm a ajustes estratégicos que responden a las particularidades de la región de procedencia.

**Tabla 1**  
*Análisis de regresión Probit Binaria de los grupos de Incoterms para las importaciones peruanas, 2021*

Dependent Variable: CE Method: ML - Binary Probit (Newton-Raphson / Marquardt steps) Date: 07/11/22 Time: 12:39 Sample: 1 706862 Included observations: 706862 Convergence achieved after 4 iterations Coefficient covariance computed using the Huber-White method				
Variable	Coefficient	Std. Error	z-Statistic	Prob.
D_SA	-0.056381	0.015489	-3.640080	0.0003
D_NA	-0.972877	0.009288	-104.7481	0.0000
LOG(DISTANCIA)	0.046775	0.007433	6.293238	0.0000
LOG(PESO_VALOR)	-0.036818	0.000643	-57.29722	0.0000
C	-1.011598	0.071665	-14.11573	0.0000
McFadden R-squared	0.053170	Mean dependent var		0.273980
S.D. dependent var	0.445999	S.E. of regression		0.434042
Akaike info criterion	1.111929	Sum squared resid		133166.3
Schwarz criterion	1.112010	Log likelihood		-392985.1
Hannan-Quinn criter.	1.111951	Deviance		785970.2
Restr. deviance	830106.7	Restr. log likelihood		-415053.3
LR statistic	44136.49	Avg. log likelihood		-0.555957
Prob(LR statistic)	0.000000			
Obs with Dep=0	513196	Total obs		706862
Obs with Dep=1	193666			

*Nota.* Elaboración propia utilizando EViews con información pública las Bases de Datos DBF de los Regímenes Definitivos de Comercio Internacional de la Sunat.

2) **Modelo econométrico PD Logístico Mejorado para Riesgo de Crédito**

**Fecha:** 2024 | **Indicadores Analizados:** AUC (curva ROC), Gini, Kolmogorov-Smirnov, Coeficientes, base de datos pública de préstamos otorgados del 2007 al 2015 por LendingClub Corporation en la plataforma Kaggle.

2.1. **Objetivo:** Desarrollar un modelo econométrico predictivo robusto de Probabilidad de Default (PD) mediante regresión logística, con mejoras sobre el enfoque tradicional, a partir de datos públicos de LendingClub, disponibles en Kaggle, el modelo se construirá mediante una selección, transformación, cálculo, validación, prueba y elaboración de variables y variables dummy. Su desempeño será evaluado con indicadores como AUC (curva ROC), coeficiente de Gini y Kolmogorov-Smirnov, asegurando una diferenciación efectiva entre clientes cumplidores e

incumplidores. Además, se establecerán una metodología y estrategias adecuadas, evitando errores metodológicos como invalidez estadística, permitiendo que su aplicación sea un factor adecuado y válido para la toma de decisiones.

## 2.2. Metodología:

- Recolección de datos a partir de archivos CSV.
- Selección de variables relevantes para representar características del crédito.
- Preprocesamiento de datos para separar la información cuantitativa y cualitativa.
- Elaboración y cálculo de variables (incluyendo dummy).
- Interpretación de coeficientes para identificar variables determinantes.
- Evaluación del desempeño del modelo a través de métricas como AUC (curva ROC), matriz de confusión, Gini y Kolmogorov-Smirnov.
- Validación complementaria con análisis estadístico.

## 2.3. Herramientas: Python, Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, Microsoft Excel.

## 2.4. Resultados:

### Indicadores Cuantitativos:

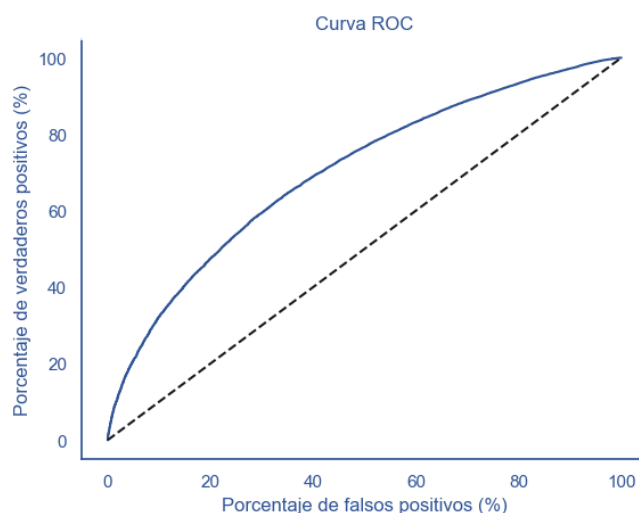
- AUC: 0.70
- Coeficiente de Gini: 40.43%
- Kolmogorov-Smirnov: 29.66%

El modelo mostró que tiene un desempeño útil y consistente con un AUC de 0.70, capaz de ser un indicador clave en la toma de decisiones, lo que indica un buen balance entre sensibilidad y especificidad. Las métricas evidenciaron un desempeño razonablemente sólido en la detección de defaults (incumplimiento), y la interpretación de coeficientes permitió identificar variables clave para la segmentación del riesgo.

### Figura 1

*Curva ROC del Modelo (AUC = 0.7)*

(En porcentaje)



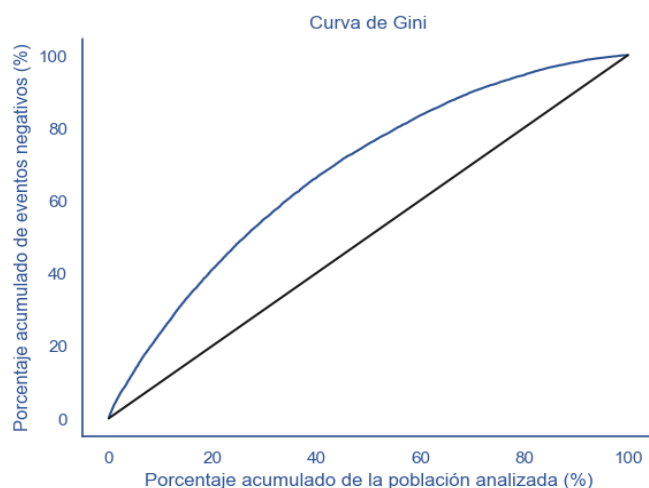
*Nota.* Elaboración propia con información pública de LendingClub Corporation en la plataforma Kaggle.

Asimismo, se calculó el coeficiente de Gini para el presente modelo PD mejorado, el cual es de 40.43%, lo que indica que el modelo tiene una capacidad moderada para diferenciar entre clientes que incumplen y los que no. Este valor implica que, al comparar las probabilidades asignadas a distintos clientes, el modelo logra distinguir correctamente a aquellos con mayor riesgo de incumplimiento en una proporción razonable de los casos.

**Figura 2**

*Curva de Gini*

(En porcentaje)



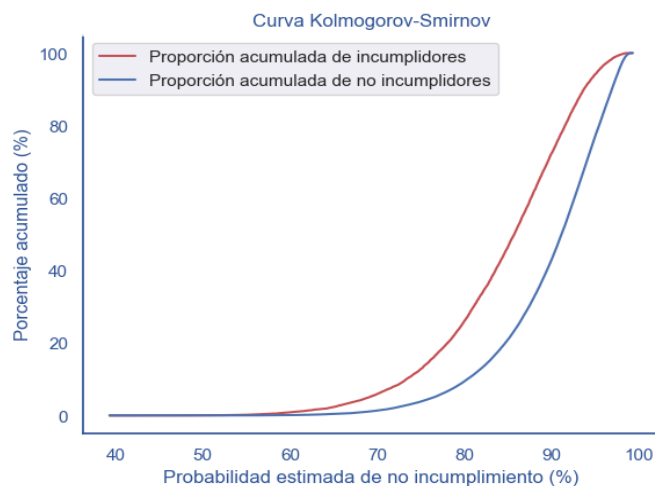
*Nota.* Elaboración propia con información pública de LendingClub Corporation en la plataforma Kaggle.

Un KS del 29.66% indica que el modelo tiene una capacidad de separación moderada entre clientes que incumplen y los que no. Esto podría indicar que existe cierta superposición entre las distribuciones de incumplidores y no incumplidores, lo que implica que, aunque se logra distinguir en gran medida a ambos grupos, aún hay casos en los que los perfiles puedan presentar ambigüedad o incertidumbre.

**Figura 3**

(En porcentaje)

*Curva Kolmogorov-Smirnov*



*Nota.* Elaboración propia con información pública de LendingClub Corporation en la plataforma Kaggle.

**Figura 4***Fragmento del código de Python utilizado para el análisis*

```
[58]: from sklearn.metrics import roc_curve, roc_auc_score

[59]: roc_curve(df_actual_predicted_probs['loan_data_targets_test'], df_actual_predicted_probs['y_hat_test_proba'])

[59]: (array([0.          , 0.          , 0.          , ..., 0.99990186, 0.99990186,
          1.          ]),
      array([0.00000000e+00, 1.20384750e-05, 1.20384750e-04, ...,
          9.99975923e-01, 1.00000000e+00, 1.00000000e+00]),
      array([1.99266957, 0.99266957, 0.99071534, ..., 0.44818091, 0.39392732,
          0.39324259]))

[60]: fpr, tpr, thresholds = roc_curve(df_actual_predicted_probs['loan_data_targets_test'],
                                     df_actual_predicted_probs['y_hat_test_proba'])

[61]: import matplotlib.pyplot as plt
import seaborn as sns
sns.set()

[62]: plt.figure(facecolor='white')

ax = plt.gca()
ax.set_facecolor('white')
ax.spines['top'].set_color('white')
ax.spines['right'].set_color('white')
ax.spines['left'].set_color('#2F5496')
ax.spines['bottom'].set_color('#2F5496')

plt.plot(fpr, tpr, color='#2F5496')

plt.plot(fpr, fpr, linestyle='--', color='k')

plt.xlabel('Porcentaje de falsos positivos', color='#2F5496')
plt.ylabel('Porcentaje de verdaderos positivos', color='#2F5496')
plt.title('Curva ROC', color='#2F5496')

plt.xticks(color='#2F5496')
plt.yticks(color='#2F5496')

plt.grid(False)

plt.show()
```

*Nota.* Elaboración propia utilizando Jupyter Notebook con información pública de LendingClub Corporation en la plataforma Kaggle.

### 3) Análisis Geoespacial (GIS) del PBI per Cápita en Perú

**Fecha:** 2024 | **Indicadores Analizados:** PBI del Perú del 2023 a Precios Constantes del 2007, Población Estimada al 30 de junio, por Años Calendario y Sexo según Departamento del Perú en el 2023, regiones geográficas, distribución económica.

**3.1. Objetivo:** Analizar y visualizar la distribución del PBI per cápita en las distintas regiones de Perú a través de un mapa coroplético, integrando datos económicos y geográficos para facilitar la toma de decisiones en análisis económico y planificación territorial.

#### 3.2. Metodología:

- Lectura del archivo Excel con indicadores económicos (PBI) y carga del shapefile de Perú.
- Aplicación de merge entre el shapefile y el archivo Excel utilizando los campos correspondientes.
- Conversión de los datos al sistema de coordenadas EPSG:3857.
- Visualización y análisis en QGIS.
- Generación de un mapa coroplético para representar el PBI per cápita por región.
- Exportación del resultado en QGIS.

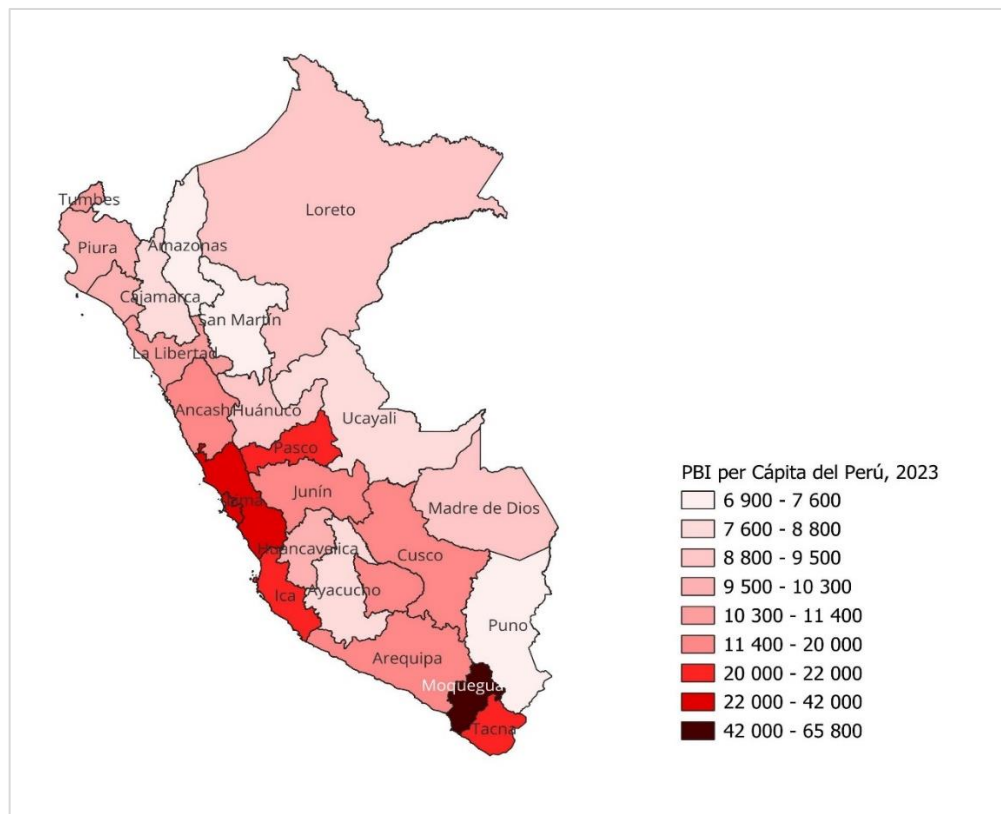
**3.3. Herramientas:** Python, QGIS, Excel, Estadísticas Públicas del INEI.

### 3.4. Resultados:

Se generó un mapa coroplético que muestra la distribución del PBI per cápita por región en Perú, permitiendo identificar de manera visual las disparidades económicas entre las distintas zonas del país. En el mapa, las tonalidades más oscuras representan valores más altos de PBI per cápita, concentrándose principalmente en algunas regiones costeras y del sur, mientras que las tonalidades más claras indican menores niveles de PBI, evidenciando la desigualdad económica entre regiones.

**Figura 5**

*Mapa coroplético del PBI per Cápita del Perú según Departamentos, 2023*  
(En soles del 2007)



*Nota.* Elaboración propia utilizando QGIS con información pública del INEI.

## 4) Automatización del Cálculo de Variación Interanual Real de Tributos Internos y Aduaneros

**Fecha:** 2023 | **Indicadores Analizados:** Estadísticas Públicas de la Nota Tributaria y Aduanera de la Sunat y Estadísticas Públicas del Índice Temático del INEI.

- 4.1. Objetivo:** Automatizar y optimizar el proceso de cálculo de la variación interanual de los tributos internos y aduaneros, ajustando los datos nominales mediante un deflactor derivado del IPC. Esto permitirá obtener resultados precisos y en un tiempo significativamente menor en comparación con el proceso manual en Microsoft Excel.

4.2. Metodología:

- Lectura y procesamiento de datos públicos de tributos internos, aduaneros e IPC desde archivos de Microsoft Excel de la Sunat y el INEI.
- Establecimiento de una fecha inicial para renombrar y organizar las columnas mensuales.
- Identificación y transformación de los datos del IPC, incluyendo la conversión de fechas y el cálculo del deflactor (utilizando un rezago de 12 meses).
- Cálculo de la variación interanual nominal para cada mes y región, seguido del ajuste de estos valores mediante el deflactor.
- Reestructuración y transposición del DataFrame para facilitar el análisis comparativo.
- Exportación de los resultados finales a un nuevo archivo Microsoft Excel.

4.3. Herramientas: Python, Microsoft Excel y Estadísticas Públicas de la Sunat y el INEI.

4.4. Resultados:

Se obtuvo un DataFrame consolidado que muestra la variación interanual real (ajustada mediante el deflactor del IPC) de los tributos internos y aduaneros por región. Este proceso automatizado reemplaza los cálculos manuales realizados previamente en Microsoft Excel, reduciendo en más de un 80% el tiempo invertido y minimizando la posibilidad de errores, lo que representa una mejora sustancial en la eficiencia del análisis.

Figura 6

*Fragmento de código de carga y transformación de Datos de IPC y Tributos Internos y Aduaneros*

```
https://www.inei.gob.pe/estadisticas/indice-tematico/price-indexes/

Índice de Precios al Consumidor de Lima Metropolitana (Bases 2009=100 y Diciembre 2021=100)
Variación Porcentual Mensual, Acumulada y Anual

Paso 1: Leer archivo descargado de las estadísticas del INEI
archivo_ipc = 'IPC.xlsx' # Ruta archivo de La base del INEI

https://www.sunat.gob.pe/estadisticasestudios/ingresos-recaudados.html

Paso 2: Leer archivo de Tributos Internos o Aduaneros
archivo_tributos = 'TRIBUTOS.xlsx'

Paso 3: Fijar fecha inicial de Tributos Internos o Aduaneros y la columna ('Ene.' o 'Ene')

import pandas as pd
from datetime import datetime, timedelta

# Fijar la fecha inicial
fecha_inicial = datetime(year=2005, month=1, day=1)

# Cargar el archivo con los datos de tributos internos (IGV, IR, Aduaneros, etc.)
df = pd.read_excel(archivo_tributos, header=0)

# (IMPORTANTE) Hallar la columna que contiene "Ene." y calcular los meses
meses = df.columns.tolist()[df.columns.tolist().index('Ene.'):]

# Función para calcular los meses desde la fecha inicial
```

*Nota.* Elaboración propia utilizando Jupyter Notebook con información pública de la Sunat y el INEI.

5) Modelo financiero de portafolio óptimo de Markowitz dinámico

Fecha: 2024 | Indicadores Analizados: Rendimientos diarios de las empresas pertenecientes al Índice S&P/BVL Peru Select y el rendimiento del Bono del gobierno peruano a 10 años.

**5.1. Objetivo:** Optimizar matemáticamente un portafolio en ventanas interanuales móviles y analizar su evolución en función del riesgo y retorno. Identificar los períodos de mayor y menor desempeño en la estimación de la frontera eficiente, observando además la dinámica estocástica de los rendimientos del índice S&P/BVL Peru Select.

**5.2. Metodología:**

- Análisis de rendimientos diarios de las acciones que conforman el índice S&P/BVL Peru Select.
- Optimización dinámica de portafolios basada en el modelo de Markowitz.
- Uso de ventanas interanuales móviles para capturar la evolución del portafolio.
- Maximización del ratio de Sharpe con restricciones de no apalancamiento y plena inversión.
- Optimización resuelta mediante SLSQP (Sequential Least Squares Programming).
- Incorporación de la tasa libre de riesgo para ajustar los retornos (rendimiento del Bono del gobierno peruano a 10 años (en S/)).
- Evaluación de la evolución temporal de los pesos óptimos, volatilidad y rendimiento del portafolio.

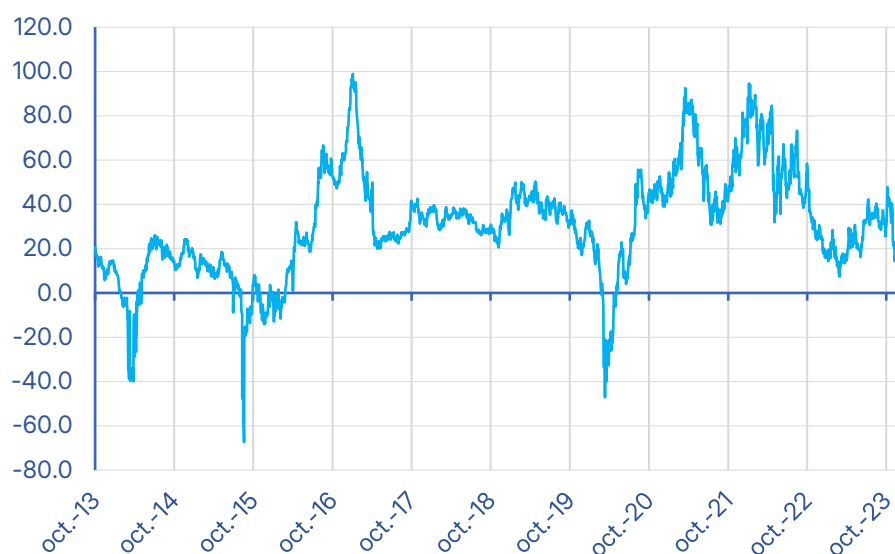
**5.3. Herramientas:** Microsoft Excel, Python, Estadísticas Públicas de la BVL y el BCRP (BCRPData).

**5.4. Resultados:**

El portafolio óptimo con ventana interanual móvil de rendimiento máximo porcentual (positivo) durante el intervalo de tiempo analizado fue de 95.1% el día 11 de enero de 2017 y el de rendimiento mínimo porcentual (negativo) de -65.3% durante el 27 de agosto de 2015, denotando momentos de rendimientos extraordinarios y pérdidas cuantiosas.

**Figura 7**

*Rendimientos de portafolios de Markowitz con ventanas interanuales móviles, 2013-2023*  
(En porcentaje)



*Nota.* Elaboración propia con información pública de la BVL y el BCRP.

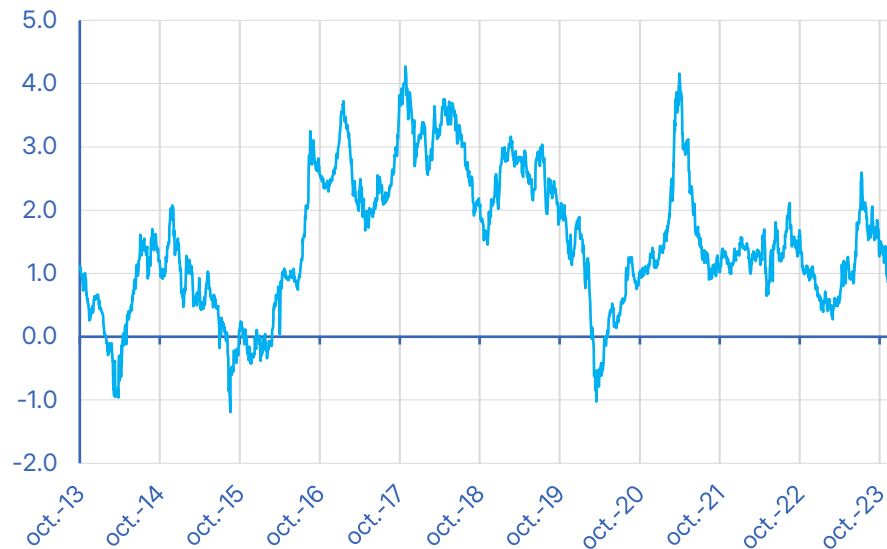


La evolución de la ventana interanual del ratio de Sharpe tuvo períodos de alta rentabilidad ajustada por riesgo seguidos de caídas. Destacan un máximo de 4.3 el 04 de noviembre de 2017 y un mínimo de -1.2 el 27 de agosto de 2015, reflejando momentos de excelente desempeño y de riesgo no compensado, respectivamente.

**Figura 8**

*Ratio de Sharpe con ventanas interanuales móviles, 2013-2023*

(En porcentaje)



*Nota.* Elaboración propia con información de la BVL y el BCRP.

**Figura 9**

*Fragmento del código de Python utilizado para el análisis*

```
import pandas as pd
import numpy as np
import scipy.optimize as sco
from datetime import timedelta

# Cargar el archivo de Excel
archivo = 'REND_SELECTIVO.xlsx'
df = pd.read_excel(archivo)

# Convertir la columna 'Fecha' a formato datetime
df['Fecha'] = pd.to_datetime(df['Fecha'])

# Detectar automáticamente las columnas "Fecha" y "Tasa Free Risk"
fecha_col = 'Fecha'
tasa_free_risk_col = 'Tasa Free Risk'
accion_cols = [col for col in df.columns if col not in [fecha_col, tasa_free_risk_col]]

# Función para calcular métricas del portafolio
def desemp_portafolio(pesos, rendimientos, tasa_risk_free):
    rend_portafolio = np.sum(rendimientos.mean() * pesos) * 252 # Asumiendo 252 días hábiles
    desv_est_port = np.sqrt(np.dot(pesos.T, np.dot(rendimientos.cov() * 252, pesos)))
    ratio_sharpe = (rend_portafolio - tasa_risk_free) / desv_est_port
    return rend_portafolio, desv_est_port, ratio_sharpe

# Función a minimizar (Sharpe negativo porque scipy minimiza)
def ratio_sharpe_negativo(pesos, rendimientos, tasa_risk_free):
    return -desemp_portafolio(pesos, rendimientos, tasa_risk_free)[2]
```

*Nota.* Elaboración propia utilizando Jupyter Notebook con información de la BVL y el BCRP.