



SKILLING
CENTER

TECMILENIO



Aplicaciones Financieras

Asignación de activos en Python





No cabe duda de que la decisión más difícil para inversiones en el mercado de valores es la selección de activos, considerando por un lado el rendimiento y por el otro lado, el riesgo.

El departamento de banca de inversión de un prestigioso banco ofrece a sus clientes una variedad de fondos de inversión. La empresa conforma los portafolios de cada fondo de inversión de acuerdo con las preferencias y conveniencias de segmentos de clientes. Para tal efecto, está buscando la herramienta para hacer pruebas de la composición de carteras antes de tomar la decisión definitiva de los activos y la cantidad de recursos asignados para cada uno.



- ¿Cuáles son los factores que la institución debe considerar antes de tomar la decisión de la composición del portafolio de cada fondo?
- ¿Cuáles activos deberá seleccionar para diversificar lo más posible el portafolio para mitigar el riesgo?
- ¿Cuáles activos deberá seleccionar para maximizar el rendimiento esperado del portafolio?



- La selección de activos financieros en un portafolio de inversión es una decisión que debe estar basada en objetivos de rendimiento y riesgo previamente planteados antes de invertir.
- Por ejemplo, un inversionista puede optar por incluir activos que maximicen el rendimiento, mientras que otro inversionista puede optar por activos con menor exposición al riesgo.
- Hay una gran variedad de formas, reglas o criterios para seleccionar activos. Cada uno aplica la que mejor se ajusta a sus necesidades.





- Por asignación de activos, se refiere a distribuir el monto total de la inversión entre ciertos activos (ya sean acciones, opciones, bonos o cualquier otro instrumento financiero) (Lewinson, 2020).
- La teoría moderna de portafolio (*modern portfolio theory*) se refiere a un método para seleccionar inversiones que maximice el rendimiento esperado para un nivel de riesgo dado.
- La idea principal de esta teoría se centra en reconocer el impacto del desempeño individual de un activo financiero sobre el desempeño del portafolio en su conjunto.
- En este punto, entra el concepto de la diversificación de activos para reducir el riesgo total.

Evaluación del desempeño de una cartera básica

- Imagina que consideras el supuesto de igualar los pesos de cada activo como proporción del valor total del portafolio, lo cual no garantiza el mejor desempeño del portafolio, pero es un criterio básico para diversificar.
- A continuación, se analiza un portafolio en Python, calculando los rendimientos por medio de la librería pyfolio y obtener su métricas de evaluación de desempeño (Lewinson, 2020).

- # 1 importar las librerías

```
- import yfinance as yf  
- import numpy as np  
- import pandas as pd  
- import pyfolio as pf
```

- # 2 establecer los activos y periodo de analisis

```
- RISKY_ASSETS = ['AAPL', 'IBM', 'AMZN', 'DIS',  
- 'WMT']  
- START_DATE = '2020-01-01'  
- END_DATE = '2023-02-28'  
- n_assets = len(RISKY_ASSETS)
```

- # 3 descargar los precios historicos de Yahoo Finance

```
- prices_df = yf.download(RISKY_ASSETS,  
- start=START_DATE,  
- end=END_DATE)
```

- # 4 Calcular los rendimientos diarios individuales de los activos

```
- returns = prices_df['Adj  
- Close'].pct_change().dropna()
```

Date	AAPL	AMZN	DIS	IBM	WMT
2023-02-27 00:00:00	0.0082475	0.00278077	0.00149545	-0.000612712	-0.00722964
2023-02-24 00:00:00	-0.0180053	-0.0242121	-0.0140568	-0.00168198	0.00267447
2023-02-23 00:00:00	0.00329051	0.000313172	0.000491769	-0.00137442	-0.0149058
2023-02-22 00:00:00	0.00289607	0.0127934	-0.00401603	-0.00561845	-0.0209733
2023-02-21 00:00:00	-0.0266798	-0.0269547	-0.0297472	-0.0245149	0.00607747
2023-02-17 00:00:00	-0.0075467	-0.00967911	-0.00576397	0.00014818	0.0150413
2023-02-16 00:00:00	-0.0104294	-0.0297549	-0.0312156	-0.0102639	-0.0156922
2023-02-15 00:00:00	0.0139034	0.014644	0.0146758	0.00286743	0.00742319
2023-02-14 00:00:00	-0.00422495	0.00160735	0	-0.00975618	-0.00287841
2023-02-13 00:00:00	0.0188068	0.0197726	-0.00370159	0.0129056	0.015238

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.



Continuación del código Python para análisis del desempeño de portafolio:

- **# 5 definir ponderaciones para cada activo (en este caso, iguales)**
- **portfolio_weights = n_assets * [1 / n_assets]**

- **# 6 Calcular los rendimientos diarios del portafolio**
- **portfolio_returns = pd.Series(np.dot(portfolio_weights, returns.T),**
- **index=returns.index)**

La librería pyfolio (pf) genera una serie de gráficas para analizar el desempeño del portafolio en su conjunto:

- **# 7 Crear las graficas de indicadores**
 - **pf.create_simple_tear_sheet(portfolio_returns)**
- Gráfica de rendimientos acumulados del portafolio. Muestra la evolución del rendimiento en el tiempo de análisis, como porcentaje de cambio sobre el valor inicial.

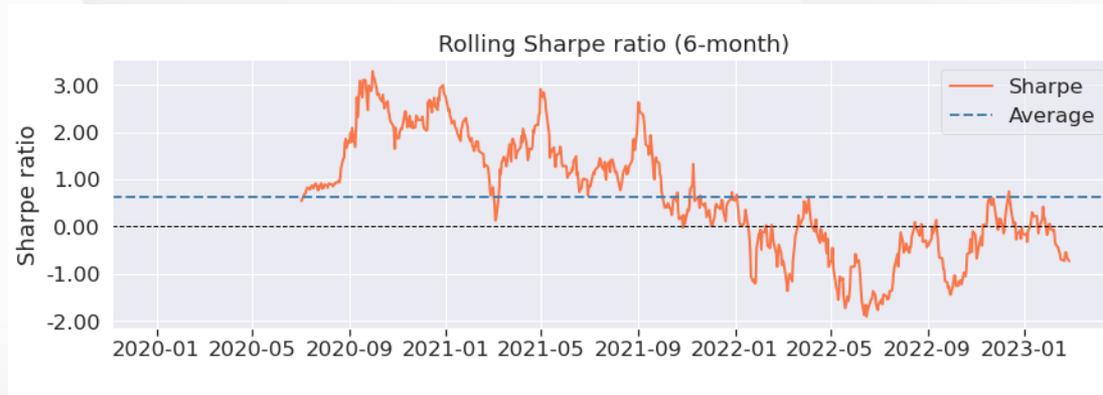


Date	0
2023-02-27 00:00:00	0.000936275
2023-02-24 00:00:00	-0.0110563
2023-02-23 00:00:00	-0.00243695
2023-02-22 00:00:00	-0.00298365
2023-02-21 00:00:00	-0.0203638
2023-02-17 00:00:00	-0.00156006
2023-02-16 00:00:00	-0.0194712
2023-02-15 00:00:00	0.0107028
2023-02-14 00:00:00	-0.00305044
2023-02-13 00:00:00	0.0126043

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

- Gráfica del métrico Sharpe ratio.



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

El Sharpe ratio mide el retorno en exceso sobre la tasa libre de riesgo por unidad de la desviación estándar (riesgo). A mayor valor del Sharpe ratio, mejor es el desempeño del portafolio.

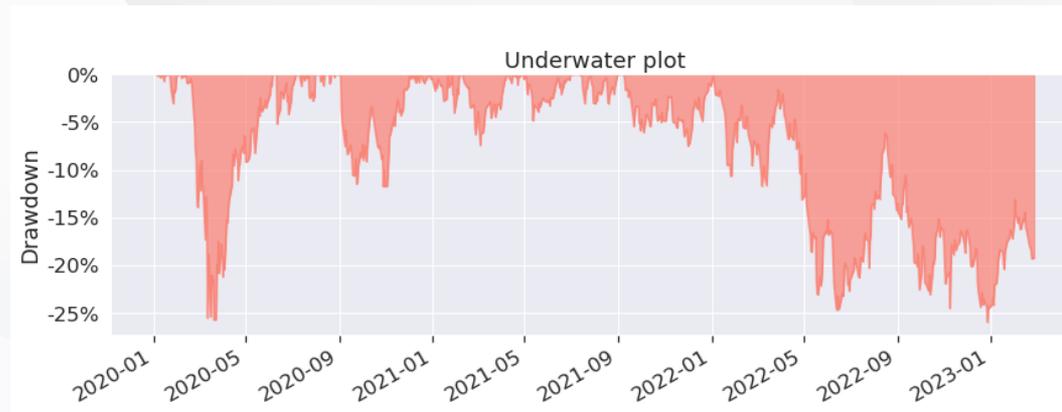
$$\text{Sharpe ratio} = \frac{r_p - r_f}{\sigma_p}$$

En donde r_p es el rendimiento del portafolio, r_f es la tasa libre de riesgo y σ_p es la desviación estándar de los rendimientos del portafolio. Cuando no se considera la tasa libre de riesgo o de referencia, por default se asume un valor de 0 %.

- El Sharpe ratio se calcula como un métrico móvil, regularmente en un periodo de 6 meses.
- Un valor negativo significa que la tasa de referencia es mayor que el retorno histórico del portafolio. De acuerdo con Fernando (2022), este métrico fue desarrollado por William F. Sharpe, que también desarrolló el modelo de valuación de activos (CAPM).

- Gráfica “debajo del agua” (underwater plot).

Muestra el portafolio desde un punto de vista pesimista. Se enfoca en pérdidas o reducciones en rendimientos. Se observan los periodos de reducción y el tiempo que duraron.



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

- El modelo pyfolio de evaluación del desempeño de una cartera proporciona un marco de referencia para seleccionar activos financieros y determinar objetivos de tolerancia a los métricos especificados.
- En el ejercicio desarrollado se seleccionaron acciones de diferentes industrias como una estrategia de diversificación, sin embargo, el inversionista podrá hacer pruebas con diferente mezcla en su cartera de inversión.



Cálculo de la frontera eficiente mediante simulaciones de Monte Carlo

- Continuando con la teoría moderna de portafolios, el modelo de la frontera eficiente (*Efficient Frontier*) es un conjunto de portafolios óptimos bajo el enfoque de riesgo-retorno (Lewinson, 2020). Ofrece la mezcla óptima de activos en un portafolio, con el mayor retorno esperado para un nivel dado de riesgo y el menor nivel de riesgo para un nivel de retorno esperado. Todas las carteras ubicadas debajo de la curva de la frontera eficiente se consideran subóptimas.
- Se desarrollará la frontera eficiente usando simulaciones Monte Carlo, generando miles de portafolios, asignando pesos aleatorios a cada activo y visualizando los resultados.
- Como ejemplo, se continúa utilizando el mismo conjunto de acciones del ejercicio anterior.
 - **# 1 importar las librerías**
 - **import yfinance as yf**
 - **import numpy as np**
 - **import pandas as pd**
 - **import matplotlib.pyplot as plt**

 - **# 2 establecer los parametros**
 - **N_PORTFOLIOS = 10 ** 5 # se generan 100,000 portafolios**
 - **N_DAYS = 252 # días de operacion promedio en un año**
 - **RISKY_ASSETS = ['AAPL', 'IBM', 'AMZN', 'DIS', 'WMT']**
 - **RISKY_ASSETS.sort()**
 - **START_DATE = '2020-01-01'**
 - **END_DATE = '2023-02-28'**
 - **n_assets = len(RISKY_ASSETS)**





Continuación del código Python para el modelo de la frontera eficiente:

```
- # 3 descargar los precios historicos de Yahoo Finance
- prices_df = yf.download(RISKY_ASSETS, start=START_DATE,
- end=END_DATE)

- # 4 calculo de los retornos promedio anualizados y su desviacion estandar
- returns_df = prices_df['Adj Close'].pct_change().dropna()
- avg_returns = returns_df.mean() * N_DAYS
- cov_mat = returns_df.cov() * N_DAYS

- # 5 simulaciones de ponderacion aleatoria del portafolio
- np.random.seed(42)
- weights = np.random.random(size=(N_PORTFOLIOS, n_assets))
- weights /= np.sum(weights, axis=1)[:, np.newaxis]

- # 6 calculo de los metricos del portafolio
- portf_rtns = np.dot(weights, avg_returns)

- portf_vol = []
- for i in range(0, len(weights)):
-     portf_vol.append(np.sqrt(np.dot(weights[i].T,
-                                     np.dot(cov_mat, weights[i])))
- portf_vol = np.array(portf_vol)
- portf_sharpe_ratio = portf_rtns / portf_vol

- # 7 generar un dataframe integrando toda la informacion
- portf_results_df = pd.DataFrame({'returns': portf_rtns,
- 'volatility': portf_vol,
- 'sharpe_ratio':
-     portf_sharpe_ratio})
```

- Vista parcial del dataframe portf_results_df con las series de tiempo de la simulación de retornos, volatilidad y el Sharpe ratio:

Inde ▾	returns	volatility	sharpe ratio ▲
99990	0.106881	0.283157	0.377463
99991	0.128571	0.243257	0.528538
99992	0.0847666	0.250397	0.338528
99993	0.144335	0.266683	0.541224
99994	0.0951953	0.26228	0.362954
99995	0.0962723	0.257767	0.373486
99996	0.0496869	0.290379	0.17111
99997	0.0984585	0.250035	0.393779
99998	0.0748464	0.238704	0.313553
99999	0.107577	0.25689	0.418765

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.



Continuación del código Python para el modelo de la frontera eficiente:

```
- # 8 ubicar los puntos creando la frontera eficiente
- N_POINTS = 100
- portf_vol_ef = []
- indices_to_skip = []

- portf_rtms_ef = np.linspace(portf_results_df.returns.min(),
-                             portf_results_df.returns.max(),
-                             N_POINTS)
- portf_rtms_ef = np.round(portf_rtms_ef, 2)
- portf_rtms = np.round(portf_rtms, 2)

- for point_index in range(N_POINTS):
-     if portf_rtms_ef[point_index] not in portf_rtms:
-         indices_to_skip.append(point_index)
-         continue
-     matched_ind = np.where(portf_rtms ==
-                             portf_rtms_ef[point_index])
-     portf_vol_ef.append(np.min(portf_vol[matched_ind]))
- portf_rtms_ef = np.delete(portf_rtms_ef, indices_to_skip)
```

Continuación del código Python para el modelo de la frontera eficiente:

```
- # 9 grafica de la frontera eficiente
- MARKS = ['o', '^', '1', '*', 'v']

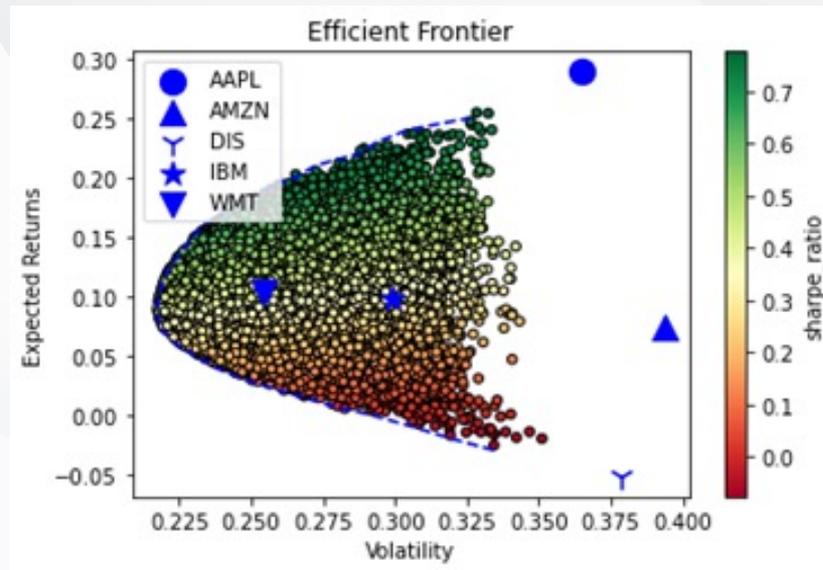
- fig, ax = plt.subplots()
- portf_results_df.plot(kind='scatter', x='volatility',
-                       y='returns', c='sharpe_ratio',
-                       cmap='RdYlGn', edgecolors='black',
-                       ax=ax)
- ax.set(xlabel='Volatility',
-        ylabel='Expected Returns',
-        title='Efficient Frontier')
- ax.plot(portf_vol_ef, portf_rtms_ef, 'b--')
- for asset_index in range(n_assets):
-     ax.scatter(x=np.sqrt(cov_mat.iloc[asset_index, asset_index]),
-               y=avg_returns[asset_index],
-               marker=MARKS[asset_index],
-               s=150,
-               color='blue',
-               label=RISKY_ASSETS[asset_index])
- ax.legend(loc="upper left")
```



Finalmente se obtiene la gráfica del modelo de la *frontera eficiente*, con todos los portafolios creados aleatoriamente, mostrando los cinco puntos de las acciones individuales y la frontera eficiente (línea azul). La forma de la frontera es un poco irregular, lo que es de esperar cuando se usan solo valores simulados que no son tan frecuentes en algunos casos extremos. Además, se colorean los puntos que representan cada una de las carteras simuladas con base en el valor del índice de Sharpe (Lewinson, 2020).



En la siguiente figura, los marcadores muestran la ubicación de cada una de las acciones en el portafolio con base en el retorno esperado y volatilidad (riesgo). En 2024, la acción de Apple tiene un Sharpe ratio alto, lo cual significa mejor desempeño ajustado por el riesgo, al contrario de las acciones de Amazon y Disney. Por lo tanto, el inversionista decidirá hacer ajustes en las acciones incluidas en su portafolio.



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Una vez realizadas las 100,000 ponderaciones del portafolio, es posible determinar cuál simulación obtuvo el mayor rendimiento, el mayor Sharpe ratio y la menor volatilidad, con sus respectivas ponderaciones.

- # 10 identificar la simulación con mayor retorno y la de menor volatilidad
- `max_sharpe_ind = np.argmax(portf_results_df.sharpe_ratio)`
- `max_sharpe_portf = portf_results_df.loc[max_sharpe_ind]`



Continuación del código Python:

- **# 11 impresión de los resultados para maximizar el Sharpe ratio**
- **print('Maximum Sharpe ratio portfolio ----')**
- **print('Performance')**
- **for index, value in max_sharpe_portf.items():**
- **print(f'{index}: {100 * value:.2f}% ', end="", flush=True)**
- **print('\nWeights')**
- **for x, y in zip(RISKY_ASSETS, weights[np.argmax(portf_results_df.sharpe_ratio)]):**
- **print(f'{x}: {100*y:.2f}% ', end="", flush=True)**

Y la impresión del retorno, volatilidad, el Sharpe ratio, así como las ponderaciones del portafolio es la siguiente:

Maximum Sharpe ratio portfolio ----

Performance: returns: 23.64 %, volatility: 30.45 % sharpe_ratio: 77.65 %

Weights

AAPL: 73.65 % AMZN: 0.23 % DIS: 2.18 % IBM: 0.04 % WMT: 23.90 %

El valor máximo del Sharpe ratio (77.65 %) le corresponde un rendimiento del 23.64 % y volatilidad del 30.45 %. Además, las ponderaciones de este portafolio sugieren incluir mayores recursos para la acción de Apple (73.65 %), seguida por Walmart (23.90 %) y Disney (2.1 %), y prácticamente nada a Amazon (0.23 %) e IBM (0.04 %).



Por otro lado, se determina la ponderación óptima, ahora para minimizar la volatilidad.

- # 12 identificar la simulacion con menor volatilidad y la ponderacion optima
- `min_vol_ind = np.argmin(portf_results_df.volatility)`
- `min_vol_portf = portf_results_df.loc[min_vol_ind]`

- # impresión de los resultados para minimizar la volatilidad
- `print('Minimum Volatility portfolio ----')`
- `print('Performance')`
- `for index, value in min_vol_portf.items():`
- `print(f'{index}: {100 * value:.2f}% ', end="", flush=True)`
- `print('\nWeights')`
- `for x, y in zip(RISKY_ASSETS, weights[np.argmin(portf_results_df.volatility)]):`
- `print(f'{x}: {100*y:.2f}% ', end="", flush=True)`



La impresión de los resultados bajo el enfoque de minimizar volatilidad es la siguiente:

Minimum Volatility portfolio ----

Performance: returns: 9.06 % volatility: 21.76 % sharpe_ratio: 41.64 %

Weights

AAPL: 0.38 % AMZN: 11.17 % DIS: 5.40 % IBM: 26.66 % WMT: 56.39 %

Bajo este criterio, el modelo tiene menor rendimiento esperado 9.06 %, volatilidad 21.76 % y el Sharpe ratio del 41.64 %. Las ponderaciones sugieren asignar más recursos a la acción de Walmart (56.39 %), seguida por IBM (26.66 %) y Amazon (11.17 %), un poco a Disney y prácticamente nada a Apple.



Para incrementar tu comprensión del tema, realiza lo siguiente:

1. Desarrolla el código Python para contestar los siguientes incisos:
 - a) Selecciona un portafolio de cinco acciones y/o instrumentos financieros que se cotizan en la bolsa de valores.
 - b) Comenta el criterio para tu selección.
 - c) Descarga los precios históricos en un periodo de 3 años a la fecha.
 - d) Elabora la gráfica de rendimientos acumulados del portafolio.
 - e) Elabora la gráfica del Sharpe ratio.
 - f) Elabora la gráfica “debajo del agua” (underwater plot).
 - g) Desarrolla el modelo de la frontera eficiente, identificando la mezcla óptima para maximizar el rendimiento y la mezcla óptima para maximizar el Sharpe ratio.

Presenta un reporte con los resultados y elabora tus comentarios y observaciones sobre el análisis que desarrollaste.



La selección de activos financieros para incluirlos en una cartera o portafolio de inversión es una tarea que todo inversionista debe practicar. Para esto, hay diversos métodos estadísticos especializados para orientar la decisión de la selección, considerando principalmente dos parámetros: los retornos esperados y la volatilidad del portafolio.

En realidad, no hay una fórmula mágica que determine lo mejor para ambos parámetros, dado que cada inversionista tiene diferente criterio para balancear su portafolio entre rendimiento y riesgo. En este tema se utilizó la ponderación o pesos de cada activo en el portafolio, tanto iguales como diferentes, para determinar la mejor mezcla de una posición de acciones.

Mediante herramientas para la asignación de activos a un portafolio de inversión cubiertas en este tema, tanto el departamento de banca de inversión del banco como tú mismo, tienen la confianza de llevar a cabo las estrategias adecuadas para invertir o desinvertir en determinados activos financieros, en beneficio de sus clientes y de la misma institución.



- Fernando, J. (2022). *Sharpe Ratio: Definition, Formula and Examples*. Recuperado de <https://www.investopedia.com/terms/s/sharperatio.asp>
- Lewinson, E. (2020). *Python for Finance Cookbook*. Packt Publishing.

Aplicaciones Financieras

Diseño de cuadros de mando
interactivos (dashboards)



El Sr. Monet, director de editorial de la publicación mensual de una prestigiada empresa administradora de inversiones, está diseñando el formato de sus publicaciones como un servicio informativo a sus clientes. Por lo que ha investigado en publicaciones similares en el mercado, sabe que hay una gran variedad de estilos y maneras de mostrar los análisis que interesan a sus clientes. Incluso ha visto en la sección financiera de algunos periódicos de circulación nacional, que publican breves cápsulas financieras muy informativas y de fácil interpretación.

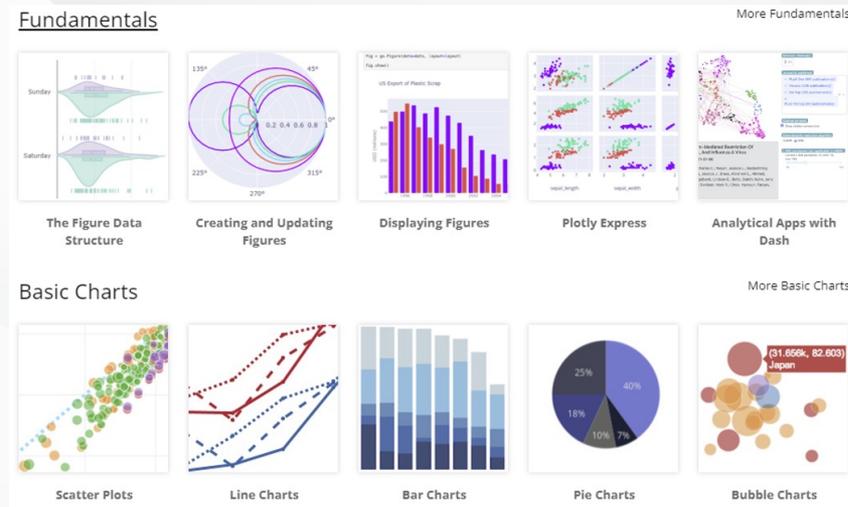
El Sr. Monet ha tomado la decisión de solicitar los servicios de expertos en programación Python y en finanzas para que le desarrollen el modelo más apropiado para sus requerimientos.

- ¿Cuáles serán las especificaciones del servicio que está solicitando?
- ¿Cuál será el formato ideal para que sus clientes aprecien sus publicaciones periódicas?



Empleo de la librería Plotly en Python

- Plotly es una librería para graficar una amplia gama de tipos especializados en el diseño de visualizaciones básicas e interactivas con el usuario, enfocado en proporcionar mucha más flexibilidad en el diseño, que incluye títulos, tipos de letras, tamaños, colores, columnas, etc., todo lo relacionado con la configuración de un dashboard (Plotly, s.f.).



Fuente: Plotly. (s.f.). *Plotly Open Source Graphing Library for Python*. Recuperado de <https://plotly.com/python/>

- Como todas las librerías de Python, Plotly también tiene módulos para diferentes circunstancias, como `plotly.graph_objects`, `plotly.express` y `plotly.subplots`, entre otros.
- Dabbas (2021) afirma que el `GraphFrame` es el espacio para desarrollar y configurar elementos como gráficas, tablas, texto, animaciones, etc. Es el complemento del `DataFrame` utilizado en Python.



	Open	High	Low	Close	Adj Close	Volume
Date						
2023-01-02	82.500000	85.400002	82.500000	84.650002	83.964363	161367
2023-01-03	84.650002	85.040001	83.760002	84.510002	83.825500	697403
2023-01-04	84.980003	87.150002	84.970001	85.889999	85.194321	1970927
2023-01-05	85.599998	89.239998	85.599998	87.680000	86.969818	2645337
2023-01-06	88.860001	89.500000	85.800003	88.120003	87.406258	1858177
...
2023-03-24	87.339996	88.000000	86.540001	87.489998	86.781357	1146129
2023-03-27	88.870003	88.870003	87.610001	88.400002	87.683990	1291015
2023-03-28	88.300003	90.000000	88.220001	88.400002	87.683990	2333676
2023-03-29	88.500000	90.000000	88.029999	89.980003	89.251198	1640833
2023-03-30	90.029999	92.900002	89.889999	91.730003	90.987022	1635816
[62 rows x 6 columns]						

Fuente: Yahoo Finance. (s.f.). *Grupo Bimbo, S.A.B. de C.V. (BIMBOA.MX)*. Recuperado de <https://finance.yahoo.com/quote/BIMBOA.MX?.tsrc=fin-srch>

- # configurar la grafica y mostrarla
- fig = px.line(prices_df, prices_df.index, y='Close', title="Precio de la accion de Bimbo - Simbolo BIMBOA.MX")
- fig.show()

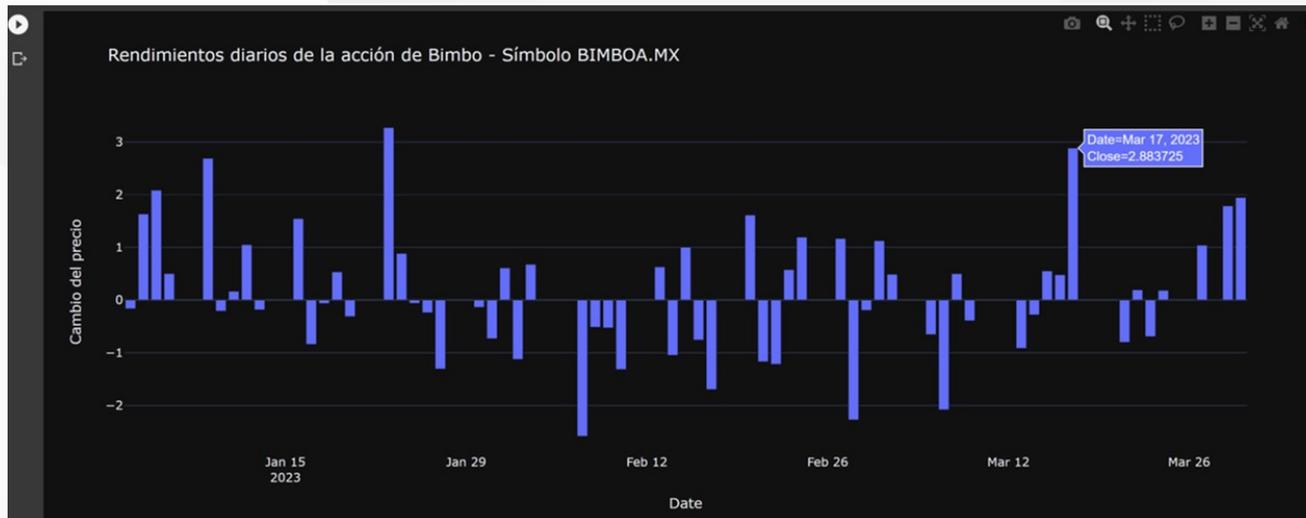
La gráfica incluye opciones de interacción con el usuario en la parte superior derecha. Además, muestra los valores de la variable al pasar el cursor por la gráfica, como se muestra en el cuadro azul.



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

El siguiente código genera una gráfica de tipo barras con los rendimientos diarios de la misma acción de Bimbo en el periodo de análisis.

- **returns_df = prices_df['Close'].pct_change().dropna() * 100**
- **fig = px.bar(returns_df, returns_df.index, y='Close', title='Rendimientos diarios de la acción de Bimbo - Símbolo BIMBOA.MX',**
- **template="plotly_dark")**
- **fig.layout.yaxis.title = "Cambio del precio"**
- **fig.show()**



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Este tipo de gráficas son muy utilizadas para mostrar la dimensión de la variación de los precios, así como los periodos de tiempo en mantener alza o baja.

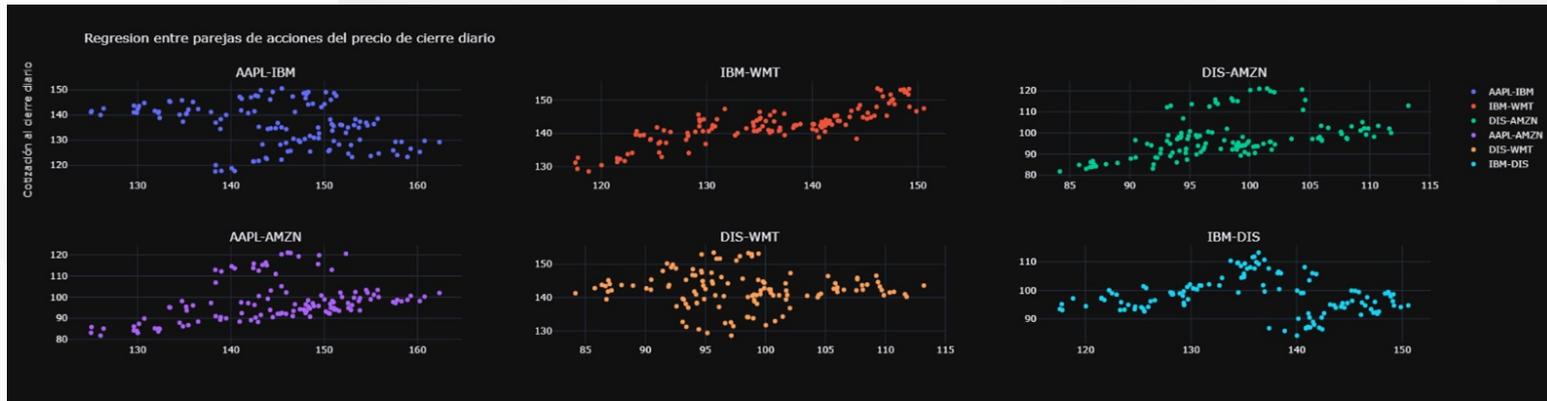


Continuación del código Python para gráficas integradas de regresión:

- **# desarrollar la grafica integrada**
- **from plotly.subplots import make_subplots**
- **fig = make_subplots(rows=2, cols=3, subplot_titles=['AAPL-IBM', 'IBM-WMT', 'DIS-AMZN', 'AAPL-AMZN', 'DIS-WMT', 'IBM-DIS'])**
- **fig.add_scatter(x=prices_close['AAPL'], y=prices_close['IBM'], row=1, col=1, mode='markers', name='AAPL-IBM')**
- **fig.add_scatter(x=prices_close['IBM'], y=prices_close['WMT'], row=1, col=2, mode='markers', name='IBM-WMT')**
- **fig.add_scatter(x=prices_close['DIS'], y=prices_close['AMZN'], row=1, col=3, mode='markers', name='DIS-AMZN')**
- **fig.add_scatter(x=prices_close['AAPL'], y=prices_close['AMZN'], row=2, col=1, mode='markers', name='AAPL-AMZN')**
- **fig.add_scatter(x=prices_close['DIS'], y=prices_close['WMT'], row=2, col=2, mode='markers', name='DIS-WMT')**
- **fig.add_scatter(x=prices_close['IBM'], y=prices_close['DIS'], row=2, col=3, mode='markers', name='IBM-DIS')**
- **fig.update_layout(title='Regresion entre parejas de acciones del precio de cierre diario', template="plotly_dark", yaxis_title="Cotización al cierre diario")**
- **fig.show()**



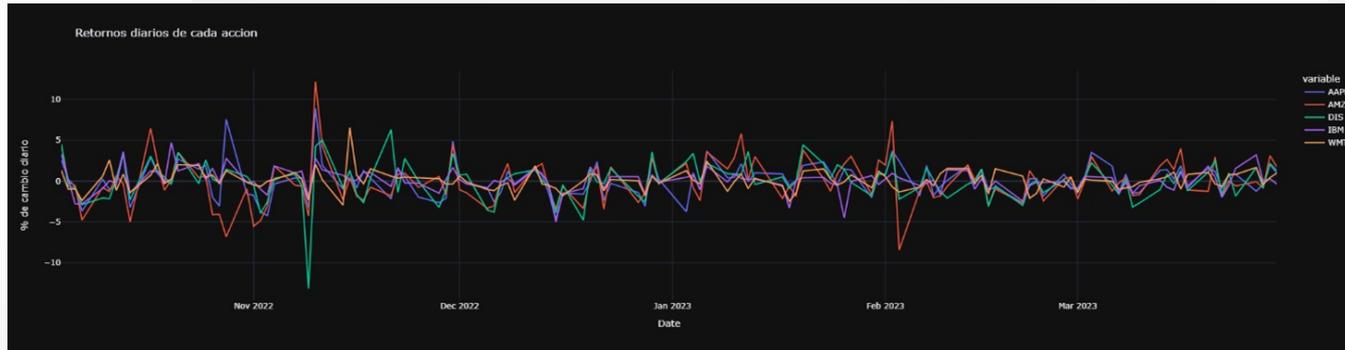
En la gráfica se puede observar la relación de desempeño entre los activos de un portafolio. Lo recomendable es que los activos tengan diferente comportamiento para asegurar la diversificación del riesgo del portafolio.



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.



La gráfica de línea de los retornos diario es la siguiente:



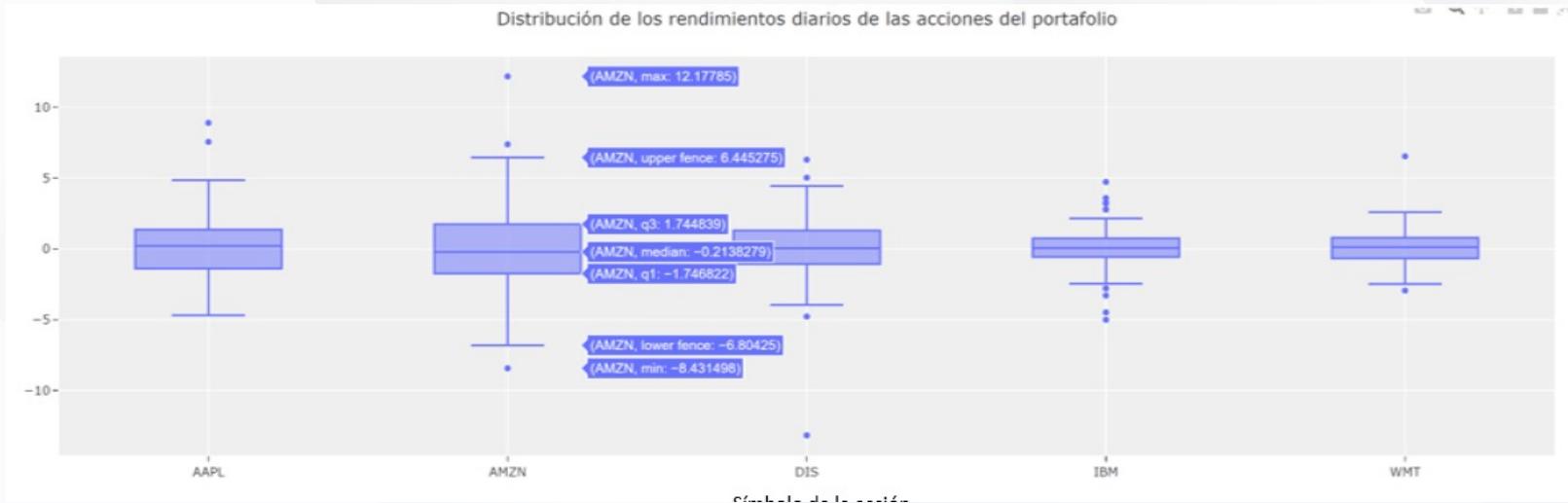
Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Adicionalmente, con las gráficas de caja (*box plots*) se puede comparar visualmente la distribución de los rendimientos diarios de cada acción en el portafolio. Al pasar el cursor por la gráfica de una acción se muestran los valores máximos, mínimo, mediana y los cuartiles, lo cual muestra la concentración de las fluctuaciones de los precios de las acciones en el periodo de análisis. A continuación, se desarrolla el código Python.

- **fig = px.box(returns_df * 100, y=returns_df.columns)**
- **fig.update_layout(title="Distribución de los rendimientos diarios de las acciones del portafolio", template='ggplot2')**
- **fig.update_xaxes(title="Simbolo de la accion")**
- **fig.update_yaxes(title="% de cambio diario")**
- **fig.show()**



El resultado es la siguiente gráfica:



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.



Dashboards interactivos

- Los dashboards interactivos ofrecen mayor funcionalidad y flexibilidad al usuario para navegar entre las variables y obtener mayor detalle del análisis en particular.
- Hay una gran variedad de software para elaborar y presentar dashboards, lo más populares son Tableau y Power BI, los cuales tienen una extensa variedad de opciones para el diseño y la visualización. Sin embargo, en Python se tiene mayor flexibilidad en el diseño.
- La selección de componentes a mostrar en un dashboard no depende del analista, sino del usuario. Por lo tanto, es importante tener mucha claridad en lo que el usuario quiere obtener y la forma en que lo va a usar.





Continuando con las acciones de Apple (AAPL), IBM (IBM), Amazon (AMZN), Disney (DIS) y Walmart (WMT) para integrar el portafolio, se va a desarrollar un dashboard básico interactivo para mostrar los precios históricos de la selección del activo.

```
- # importar utilerías
- import yfinance as yf
- from ipywidgets import interact
- import seaborn as sns
- import matplotlib.pyplot as plt

- # descargar los precios históricos y calcular los retornos diarios
- RISKY_ASSETS = ['AAPL', 'IBM', 'AMZN', 'DIS', 'WMT']
- START_DATE = '2022-10-01'
- END_DATE = '2023-03-31'
- prices_df = yf.download(RISKY_ASSETS, start=START_DATE,
-                         end=END_DATE)
- prices_close = prices_df["Close"]
- returns_df = prices_df["Close"].pct_change().dropna() * 100

- @interact
- def avg_prices(accion = prices_close): #, axis=prices_close.index):
-     sns.lineplot(data = prices_close, x = prices_close.index, y = accion)
-     plt.title(f'Comportamiento del precio de {accion}')
```



El resultado es la gráfica interactiva que permite visualizar los precios históricos de la acción seleccionada.

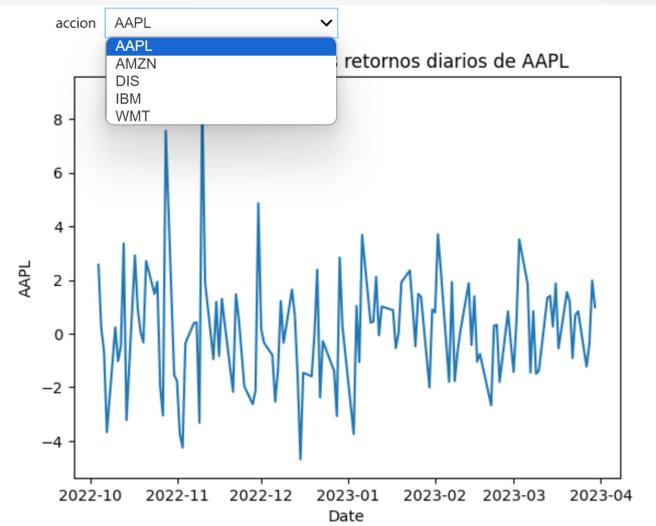
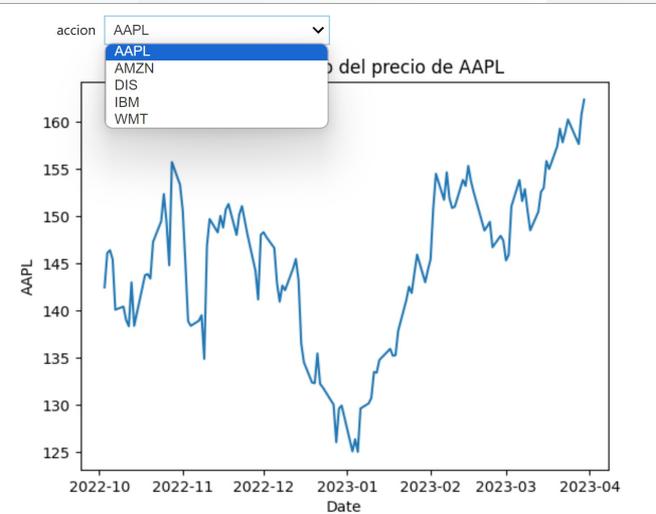
Con el mismo procedimiento, ahora se cambia la variable de los precios para ahora mostrar los retornos diarios (returns_df).

@interact

- **def avg_prices(accion = returns_df):**
- **sns.lineplot(data = returns_df, x = returns_df.index, y = accion)**
- **plt.title(f'Comportamiento de los retornos diarios de {accion}')**

Las gráficas del lado derecho muestran el comportamiento de los precios y retornos diarios históricos para la acción seleccionada de Apple.

Estas pantallas se obtuvieron directamente del software que se está explicando en la computadora, para fines educativos.





A continuación, se desarrolla la gráfica interactiva con la librería Dash de Python en Visual Studio, mediante el cual se genera una liga en internet, para luego abrirla en el explorador.

- **from dash import Dash, dcc, html, Input, Output**
- **import plotly.express as px**
- **import yfinance as yf**

- **app = Dash(__name__)**

- **app.layout = html.Div([**
- **html.H4('Análisis de precios del portafolio'),**
- **dcc.Graph(id="time-series-chart"),**
- **html.P("Seleccionar la accion:"),**
- **dcc.Dropdown(**
- **id="ticker",**
- **options=["AAPL","IBM", "AMZN", "DIS",**
- **"WMT","^VIX"],**
- **value="AMZN",**
- **clearable=False,**
- **),**
- **])**



Continuación del código Python para la gráfica interactiva en internet:

- **RISKY_ASSETS = ['AAPL', 'IBM', 'AMZN', 'DIS', 'WMT']**
- **START_DATE = '2022-10-01'**
- **END_DATE = '2023-03-31'**
- **prices_df = yf.download(RISKY_ASSETS, start=START_DATE,**
end=END_DATE)
- **prices_close = prices_df["Close"]**

- **@app.callback(**
Output("time-series-chart", "figure"),
Input("ticker", "value"))
- **def display_time_series(ticker):**
fig = px.line(prices_close, x=prices_close.index, y=ticker)

- **return fig**
- **app.run_server(debug=True)**

El resultado se muestra en la terminal de la computadora, en el cual nos indica que la dirección en internet <http://127.0.0.1:8050/>, y se copia en el explorador para obtener la gráfica interactiva.

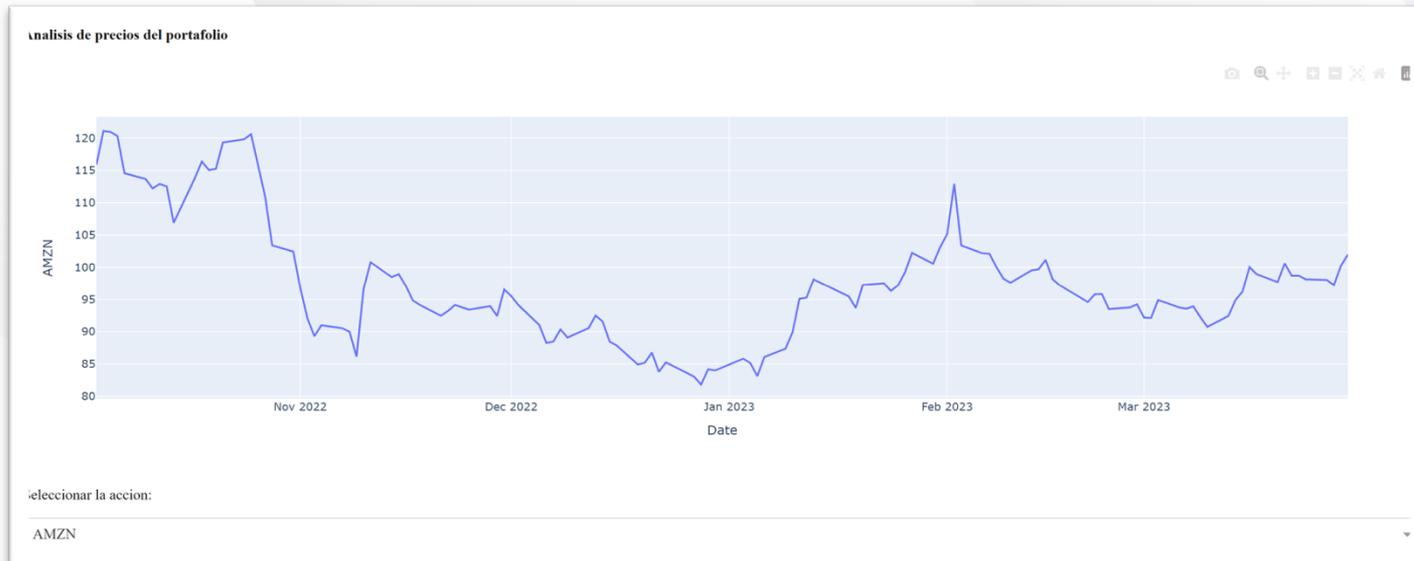
```
C:\ProgramData\Anaconda3\ x + v
[*****100%*****] 6 of 6 completed
Dash is running on http://127.0.0.1:8050/

* Serving Flask app "Dash_from_Plotly" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
[*****100%*****] 6 of 6 completed
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.



En la gráfica, se tiene el cuadro inferior de opciones en donde se selecciona la acción que se quiere mostrar.



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

La ventaja de este método es que se puede compartir el dashboard a los interesados para que fácilmente puedan acceder desde cualquier dispositivo.



Para reforzar tu comprensión del tema, realiza lo siguiente:

1. Investiga en internet cuatro librerías disponibles para elaborar dashboards en Python, y describe sus características generales, ventajas y desventajas.
2. Desarrolla el código Python utilizando la librería de Plotly para responder los siguientes incisos:
 - a) Descarga los precios históricos de la empresa Grupo Aeroportuario Del Sureste S.A. (ASR) de un periodo de 3 años a la fecha.
 - b) Elabora en gráficas separadas, los precios de cierre y los retornos diarios.
 - c) Elabora la gráfica que muestre los precios de cierre diarios y su respectiva línea de tendencia. Muestra la ecuación de la línea de tendencia y el coeficiente de regresión.
3. Selecciona cuatro acciones y/o activos financieros, descarga los precios históricos de 3 años a la fecha y elabora las gráficas de correlación de los precios históricos entre parejas de activos.
4. Con los mismos activos que seleccionaste en el punto anterior, utilizando el módulo @interact de Python, elabora la gráfica interactiva de los precios históricos, con la opción de seleccionar el activo a visualizar.



Como has observado en este tema, la visualización de los resultados es tan importante como el propio análisis. Es parte del modelo completo, ya sea para análisis técnico, comportamientos, patrones, desempeños, relaciones de precios y rendimientos de los activos financieros de forma individual o del portafolio en su conjunto, como un robusto apoyo para la toma de decisiones.

Como sabes, no todos los inversionistas prefieren el mismo formato y contenido del dashboard, aunque se trate del mismo propósito. En primer lugar, debes considerar los requerimientos del usuario final y luego seleccionar el diseño que mejor se adapte a sus requerimientos.

Ahora el Sr. Monet está listo con las especificaciones y poner manos a la obra para configurar en Python sus publicaciones. Reconoce que le va a consumir algo de tiempo, pero está convencido de que es parte importante para el éxito de sus publicaciones.



- Belorkar, A., Guntuku, S., Hora, S., y Kumar, A. (2020). *Interactive Data Visualization with Python* (2ª ed.). Packt Publishing.
- Dabbas, E. (2021). *Interactive Dashboards and Data Apps with Plotly and Dash*. Packt Publishing.
- Plotly. (s.f.). *Plotly Open Source Graphing Library for Python*. Recuperado de <https://plotly.com/python/>
- Yahoo Finance. (s.f.). *Grupo Bimbo, S.A.B. de C.V. (BIMBOA.MX)*. Recuperado de <https://finance.yahoo.com/quote/BIMBOA.MX?.tsrc=fin-srch>

Tecmilenio no guarda relación alguna con las marcas mencionadas como ejemplo. Las marcas son propiedad de sus titulares conforme a la legislación aplicable, estas se utilizan con fines académicos y didácticos, por lo que no existen fines de lucro, relación publicitaria o de patrocinio.

Todos los derechos reservados @ Universidad Tecmilenio

La obra presentada es propiedad de ENSEÑANZA E INVESTIGACIÓN SUPERIOR A.C. (UNIVERSIDAD TECNILENIO), protegida por la Ley Federal de Derecho de Autor; la alteración o deformación de una obra, así como su reproducción, exhibición o ejecución pública sin el consentimiento de su autor y titular de los derechos correspondientes es constitutivo de un delito tipificado en la Ley Federal de Derechos de Autor, así como en las Leyes Internacionales de Derecho de Autor. El uso de imágenes, fragmentos de videos, fragmentos de eventos culturales, programas y demás material que sea objeto de protección de los derechos de autor, es exclusivamente para fines educativos e informativos, y cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por UNIVERSIDAD TECNILENIO. Queda prohibido copiar, reproducir, distribuir, publicar, transmitir, difundir, o en cualquier modo explotar cualquier parte de esta obra sin la autorización previa por escrito de UNIVERSIDAD TECNILENIO. Sin embargo, usted podrá bajar material a su computadora personal para uso exclusivamente personal o educacional y no comercial limitado a una copia por página. No se podrá remover o alterar de la copia ninguna leyenda de Derechos de Autor o la que manifieste la autoría del material.