UNIVERSIDAD IBEROAMERICANA

Métodos de Monte Carlo

Aplicación Práctica

Marco Villacañas

Andrés Moguel

Inferencia Estadística I

Dr. Nelson Muriel

Licenciatura en Actuaría

Semestre Otoño 2021

6 de diciembre de 2021

Introducción
En esta última entrega se desarrolla una aplicación práctica de los métodos de Monte Carlo. Conocer el comportamiento histórico actual y futuro de acciones y portafolios de opciones, es de suma importancia para los distintos participantes del mercado de estos activos financieros. Al conocer y analizar su comportamiento se pueden tomar decisiones de compra, venta, estrategias de inversión y optimización. Este trabajo presenta la simulación de acciones y retornos de portafolios de acciones a futuro utilizando los métodos de Monte Carlo. Se desarrollan los temas de caminatas aleatorias y Movimiento Browniano Geométrico; modelos utilizados en la proyección de los portafolios y acciones respectivamente.

Marco Teórico
Caminatas Aleatorias (RW1)
Quizás la versión más simple de la hipótesis de la caminata aleatoria es el caso de incrementos distribuidos de forma independiente e idéntica en el que el
La dinámica de $\{P_t\}$ viene dada por la siguiente ecuación:

$$P_t = \mu + P_{t-1} + \epsilon_t \qquad \epsilon_t \sim i.i.d(0, \sigma^2)$$

donde es el cambio de precio esperado, y $i.i.d(0, \sigma^2)$ denota que es ϵ_t es independiente e idénticamente distribuida con media 0 y varianza σ^2. La independencia del incremento $\{\epsilon_t\}$ implica que la caminata aleatoria es un juego limpio, pero en un sentido mucho mas fuerte que martingala: Independencia implica que no solo los incrementos estén correlacionados, sino que también las funciones de los incrementos tampoco están correlacionadas. Llamaremos a esto el Random Walk o RW1.

Para desarrollar algo de intuición para RW1, considere su media condicional y varianza en la fecha t, condicionada a algún valor inicial Po en la fecha 0:

$$E[P_t|P_0] = P_0 + \mu t$$
$$Var[P_t|P_0] = a^2 t$$

Seguimos con la sustitución recursiva P_t la caminata aleatoria no es estacionaria y que su media y varianza condicional ambas son lineales en el tiempo. Estas implicaciones también son validas para las otras formas de la hipótesis del paseo aleatorio (RW2 y RW3) que se describen más adelante.

Quizás el supuesto distributivo más común para las innovaciones o incrementos ϵ_t es la normalidad. Si los ϵ_t son i.i.d $N(0, \sigma^2)$ entonces el ejemplo anterior es equivalente a un movimiento browniano aritmético, muestreado a intervalos regulares o unitarios. Esta disposición de la distribución simplifica muchos cálculos que rodean a la caminata aleatoria, pero sufre el mismo problema que afecta a los retornos distribuidos normalmente: la violación del principio de *limited* liability. Entonces siempre habrá una probabilidad positiva de que $P_t < 0$ para evitar eso, podemos usarlo realizado anteriormente y para reafirmarlo log P_t sigue una caminata aleatoria con incrementos que se distribuyen normalmente:

$$P_t = \mu + p_{t-1} + \epsilon_i \quad \epsilon_i \; i.i.d \; N(0, a^2)$$

Esto implica que los rendimientos compuestos continuamente son variables normales de i.i.d con media μ y varianza σ^2.

Incrementos Independientes (RW2)

A pesar de la elegancia y simplicidad de RW1, la suposición de idéntica incrementos distribuidos no es aprobado e para los precios de los activos financieros durante mucho tiempo lapsos de tiempo. Por ejemplo, durante los doscientos años de historia de la Bolsa de Valores de Nueva York, ha habido innumerables cambios en el entorno económico, social, tecnológico, institucional y regulatorio en el que las acciones se determinan los precios. La afirmación de que la ley de probabilidad de stoc diario.

los rendimientos se han mantenido iguales durante este período de doscientos años es simplemente increíble. Por lo tanto, relajamos los supuestos de RW1 para incluir procesos con incrementos independientes, pero no distribuidos de manera idéntica, y lo llamaremos modelo Random Walk 2 o RW2. RW2 contiene claramente RW1 como un caso especial, pero también contiene procesos de precios considerablemente más generales. Por ejemplo, RW2 permite heterocedasticidad incondicional en el ϵ_i una característica particularmente útil dada la variación en el tiempo de la volatilidad de muchas series de rendimiento de activos financieros, Aunque RW2 es más débil que RW1, aún conserva la propiedad económica más interesante de la caminata aleatoria i.i.d. La transformación de los incrementos de precios futuros no es previsible mediante cualquier transformación arbitraria de los incrementos de precios pasados.

Incrementos no Correlacionados (RW3)

Una versión aún más general de la hipótesis del paseo aleatorio, la más a menudo probado en la literatura empírica reciente, puede obtenerse relajando el supuesto de independencia de RW2 para incluir procesos con dependientes, pero incrementos no correlacionados. Esta es la forma más débil del paseo aleatorio. hipótesis, a la que nos referiremos como modelo Random Walk 3 o RW3, y contiene RW1 y RW2 como casos especiales. Un simple ejemplo de un proceso que satisface los supuestos de RW3, pero no de RW1 o RW2 es cualquier proceso para cual $Cov[\epsilon_t, \epsilon_{t-1}] = 0 \; para \; todo \; k \neq 0$ donde Cov $\left[\epsilon_t^2, \epsilon_{t-h}^2\right] \neq 0$ para cualquier $k \neq 0$ Tal proceso tiene incrementos no correlacionados, pero claramente no es independiente ya que sus incrementos al cuadrado están correlacionados

Movimiento Browniano Geométrico

Sea $p(t)$ un movimiento Browniano aritmético y dejemos que represente el precio de un activo. Dado que para $p(t)$ y para cualquier $t_1, t_2 \in [0, T]$ un intervalo de tiempo con *T* fijo:

$$p(t_2) - p(t_1) \sim N(\mu(t_2 - t_1), \sigma^2(t_2 - t_1)$$

Con μ el coeficiente de deriva y σ^2 parámetro de varianza o volatilidad.

Esto muestra que los cambios de precio sobre cualquier intervalo serán normalmente distribuidos. Dado que el soporte de la distribución normal es la línea de los reales, cambios de precios distribuidos normalmente implican que los precios pueden ser negativos con una probabilidad positiva. Dado que la mayoría de los activos financieros disfrutan de responsabilidad limitada (*limited liability)*, es decir que la pérdida máxima tiene un tope de -100% de la inversión total, los precios negativos son empíricamente implausibles.

Este problema de precios negativos se puede eliminar definiendo $p(t)$como el logaritmo natural del precio $P(t)$. Bajo esta redefinición, $p(t)$ puede ser un movimiento Browniano aritmético sin violar el principio de responsabilidad limitada. Esto es porque el precio $P(t) = e^{p(t)}$ siempre es mayor o igual a cero. El proceso de precios $P(t) = e^{p(t)}$ es un proceso Browniano geométrico.

Lema de Itô

La ecuación diferencial estocástica que gobierna las dinámicas de *p y t:*

$$df(p,t) = \frac{\partial f}{\partial p}dp + \frac{\partial f}{\partial t}dt + \frac{1}{2}\frac{\partial^2 f}{\partial p^2}(dp)^2$$

Ahora dejemos que *p* denote el proceso de log-precios de un activo y supongamos que satisface:

$$dp(t) = pdt + adB(t)$$

con $B(t)$ un movimiento Browniano estándar con coeficiente de deriva 0 y volatilidad 1.

Entonces las dinámicas del proceso de precios $P(t) = e^{p(t)}$dadas por el Lema de Itô son:

$$dP = \frac{\partial P}{\partial p}dp + \frac{1}{2}\frac{\partial^2 P}{\partial p^2}(dp)^2$$

$$= e^p dp + \frac{1}{2}e^p(dp)^2$$

$$= P(\mu dt + \sigma dB) + \frac{1}{2}P(\sigma^2 dt)$$

$$dP = \left(\mu + \frac{1}{2}\sigma^2\right)Pdt + \sigma PdB$$

De esta última expresión podemos observar que tanto la media instantánea como la desviación estándar instantánea de el movimiento Browniano geométrico son proporcionales a *P.* Además, también implica que el cambio porcentual instantáneo de precios dP/P se comporta como un movimiento Browniano Geométrico o una caminata aleatoria.

Con esto podemos definir la ecuación diferencial estocástica para un movimiento Browniano Geométrico para simular el precio de una acción:

$$\frac{dS}{S} = \mu dt + \sigma dW_t$$

con W_t: un proceso de Wiener $\sim N(0, \sigma^2 = t)$ que para una simulación de Monte Carlo también se puede definir como:

$$\frac{\Delta S}{S} = \mu\Delta t + \sigma\epsilon\sqrt{\Delta t}$$

Dónde:

S: es el precio de la acción.

ΔS: es el cambio en el precio de la acción.

μ: es el retorno esperado (puede ser en base a la información pasada de la acción o respecto a una tasa libre de riesgo como la LIBOR)

σ: la desviación estándar de los retornos o la volatilidad anualizada de la acción.

ϵ: la variable aleatoria generada

Δt: el periodo de tiempo transcurrido.

Esta ecuación será utilizada para las simulaciones de Monte Carlo realizadas en esta entrega.

Resultados

Acciones

Para este ejercicio de simulación se escogió de manera voluntaria la acción de Apple, se analizo su conducta en un periodo de tiempo de 365 días (29-12-2019 al 30-12-2020) obteniendo los datos históricos de Yahoo! Finance, haciendo 10,000 iteraciones para analizar las posibles direcciones que tomara la acción en dicho periodo, utilizando una volatilidad anualizada del 0.2379 así como una tasa libre de riesgo del 0.02, también se obtuvieron diferentes datos estadísticos para observar y valorar dicha acción como lo son la media, mediana, rango intercuantil, desviación estándar, mínimo de la acción y máximo de la acción.

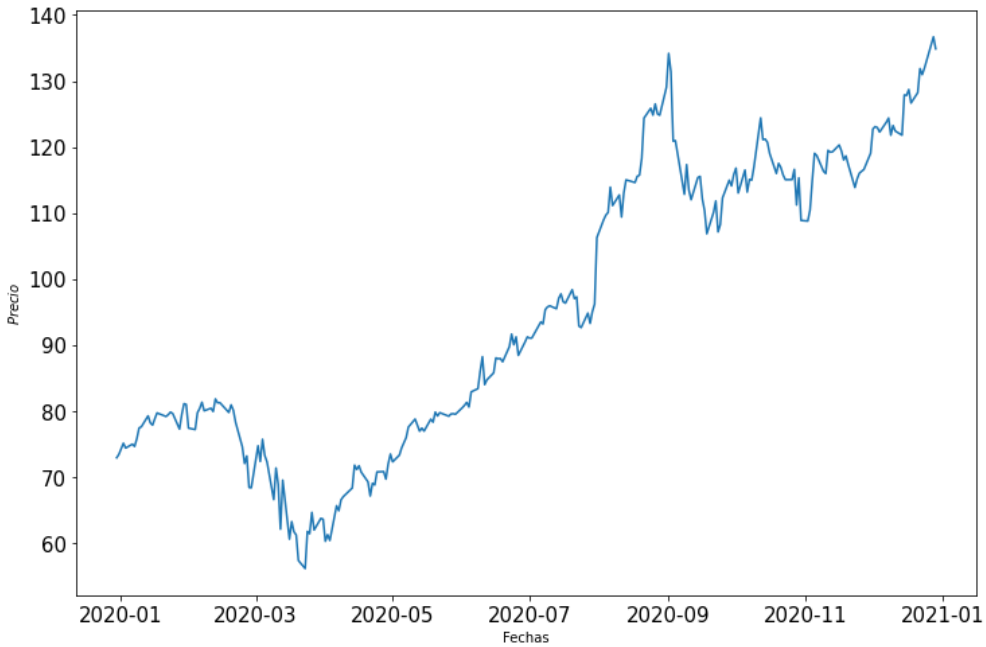

Figura 1: Historial de 1 año del precio de la acción de Apple (29-12-2019 a 30-12-2020).

Esta imagen representa el comportamiento que tuvo la acción de Apple en un periodo de un año, esto se hace con la función “yf.Ticker” la cual es una herramienta que utiliza las API de Yahoo! finance, la acción durante un año obteniendo en el periodo de 30-12-2019 un precio $72.3649 de apertura y teniendo un máximo de $73.1725, de igual manera el 29-12-2020 abrió con $138.0500 y llegando a un máximo de $138.7899.

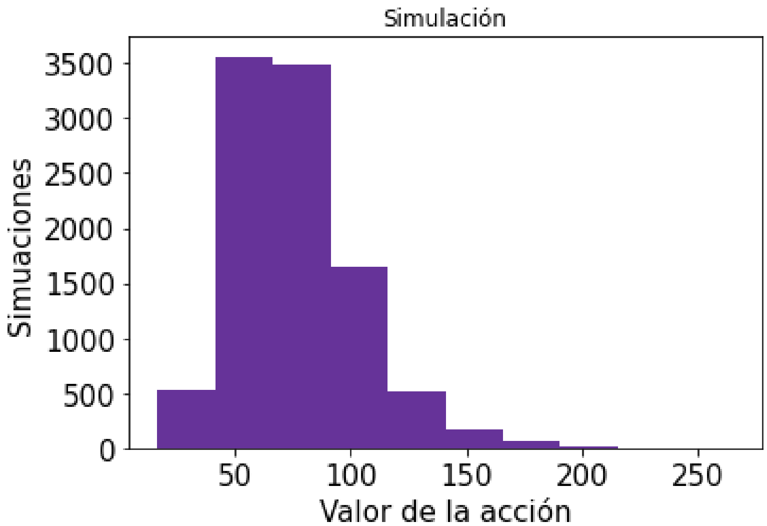

Figura 2: Distribución de la valuación de la acción final.

Una volatilidad anualizada de la acción de Apple del 0.2379 y una tasa libre de riesgo de EEUU del 0.02 se obtiene una volatilidad diaria del 0.0211, Usando las simulaciones de Monte Carlo con números aleatorios de la librería numpy se obtuvieron los siguientes resultados del último día (252), un máximo de 265.2956, mínimo de 16.8519, Rango intercuartil (IQR) 33.3071, una media 76.2771, Desviación estándar 26.6993 y mediana 71.8409.

En el histograma notamos que los valores de la acción se concentran entre los 50 y 75, se obtuvo de la manera en la que se multiplico la tasa libre de riesgo de EEUU multiplicado por el tamaño del día 1 términos de años sumando la volatilidad de la acción y multiplicado la raíz del tamaño del día 1 términos de años multiplicado por un np.random.

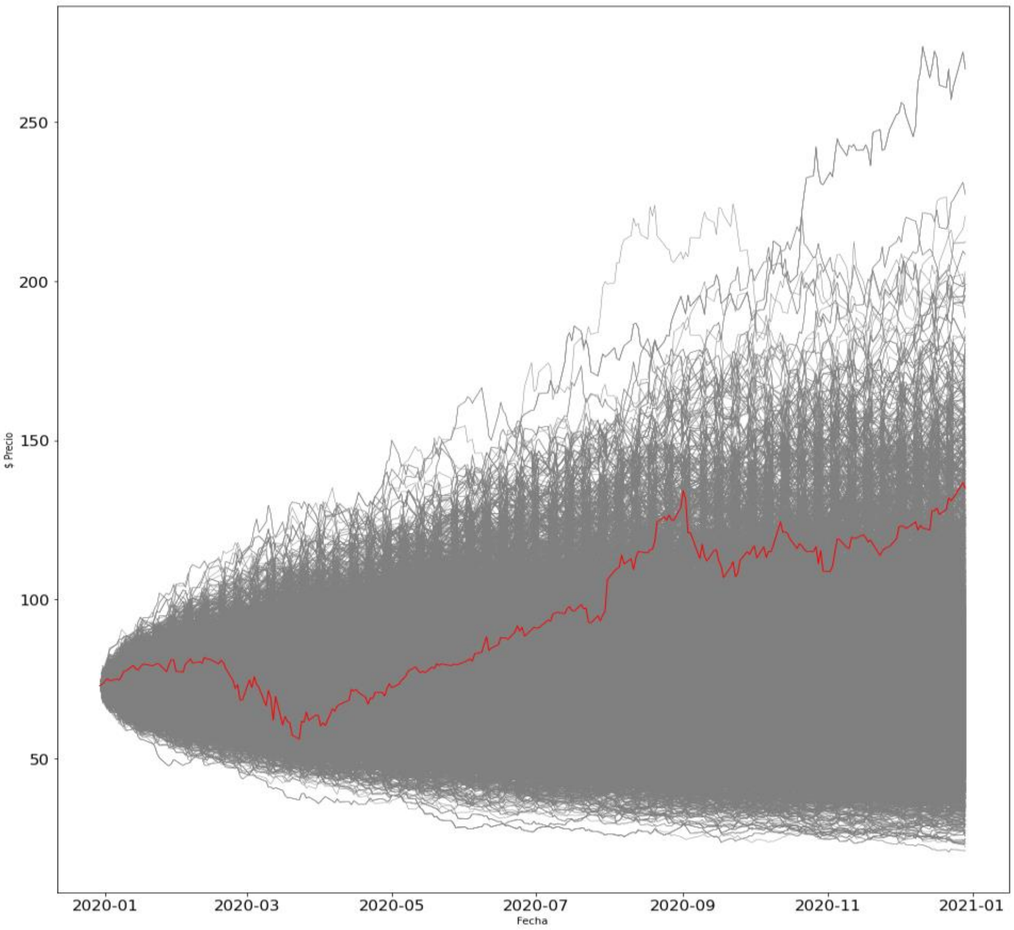

Figura 3: Simulaciones de la acción de Apple en rojo los precios históricos.
En esta imagen nos muestra la simulación de 10,000 iteraciones utilizando un np.random.choice para generar muestras aleatorias y vemos el comportamiento que puede llegar a tener en cada una de ellas, en el eje X evaluando la acción durante un año y mostrando en el eje Y el precio que puede llegar a tener la acción notando una concentración entre los precios de 50 y 150. La línea roja nos representa el comportamiento real que tuvo la acción.

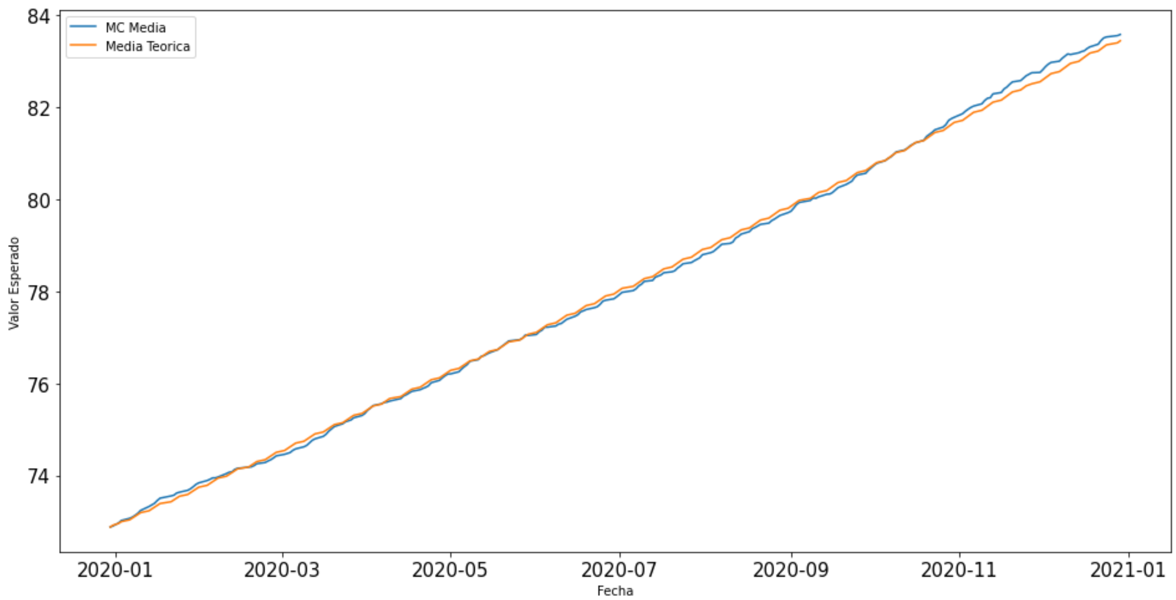

Figura 4: Grafica de la media Monte Carlo vs. Teórica.

Esta grafica muestra la media con función al tiempo con un intervalo de un año comparándola con la aproximación de Monte Carlo. Se puede observar que tanto la media de Monte Carlo y la Media Teórica se comportan de manera muy similar teniendo un ligero despunte en los últimos meses del año, pero siempre en constante crecimiento en términos de valor esperado.

Portafolios

Los Métodos de Monte Carlo nos permiten simular a futuro. En este caso se tomará la idea de simular una sola acción, pero ahora se hará para in portafolio compuesto de distintas acciones. Se toma un portafolio de tecnología con las acciones de Apple, Amazon y Tesla. Se le asignaron pesos aleatorios a cada activo dentro del portafolio. Los datos históricos de precios de las acciones se obtienen de *Yahoo! Finance* a Través de *pandas_datareader*. Se toma la información de precios de cierre en un periodo de 300 días que termina el 3 de diciembre de 2021. De esta información se calcula la media de los retornos históricos que se calcula con la función *pct_change*. También se calcula la matriz de covarianzas de los retornos. Esta simulación se hace de la forma de una caminata aleatoria. Para la aplicación de los Métodos de Monte Carlo se decide hacer 10,000 simulaciones del valor portafolio durante 365 días; 365 días a futuro. Es decir, simular los posibles valores para conocer retornos y valuación si se invierte en un año con este portafolio de acciones. Para esta simulación se asume que los retornos diarios se distribuyen de forma normal multivariada:

$$R_t \sim MVN(\mu, \Sigma)$$

Tal que:

$$R_t = \mu + LZ_t$$

Donde µ es la matriz de retornos medios, *L* es la matriz triangular inferior de la matriz de covarianzas, que se obtiene a través de la descomposición de Cholesky, y Z_t son variables aleatorias, no correlacionadas, generadas de la distribución Normal tal que:

$$Z_t = N(0, I)$$

Dónde I representa la matriz identidad.
Asumiendo un valor inicial de $10,000 USD (los precios descargados de Yahoo! Finance son en USD) para el portafolio, realizando las simulación obtenemos los siguientes resultados (las simulaciones se hicieron en Python ver Anexo 2 para consultar el código utilizado):

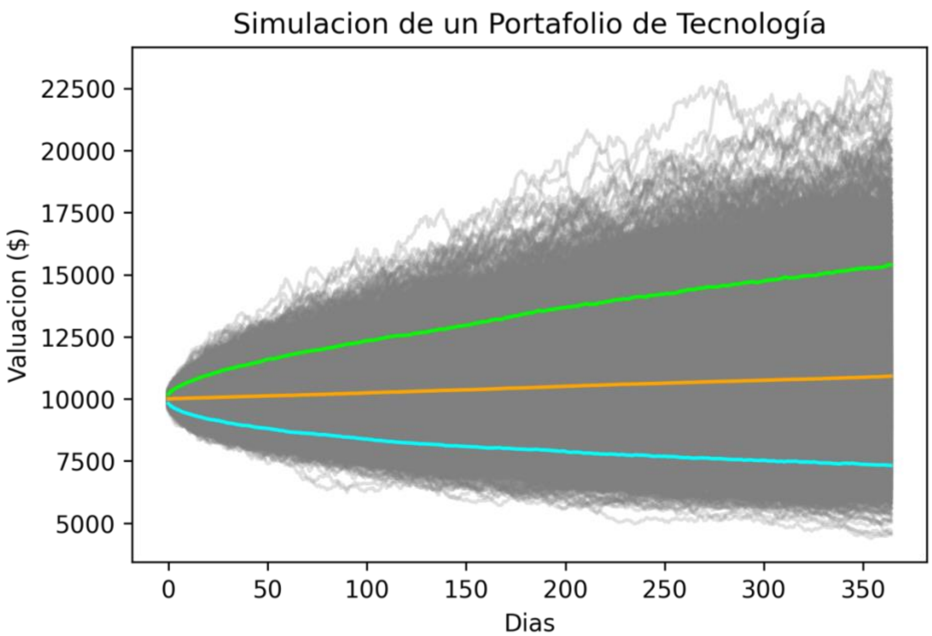

Figura 5: Simulaciones de la valuación de un portafolio de acciones de tecnología durante 1 año. En naranja, los valores promedios, en verde el cuantil del 95% y en azul el cuantil del 5%.

La figura anterior muestra el comportamiento de la valuación del portafolio a lo largo de los 365 días. Se puede observar que conforme pasa el tiempo, el cuantil del 85% y el cuantil del 5% se alejan más de la media. Esta simulaciones muestran que conforme pasa el tiempo las ganancias o pérdidas pueden ser mucho mayores. Sin embargo, es importante notar que la media se mantiene constantemente alrededor de los 10,000 iniciales. Es decir que la ganancia o pérdida promedio no es de tan gran magnitud como los cuantiles previamente mencionados. De todas las simulaciones el valor máximo que alcanza el portafolio es de $23.214,98 lo que representa un retorno del 132.15%. El valor mínimo alcanzado por el portafolio es de $4.377.10 que representa un retorno del -56.23%. Con esto podemos concluir que en estas simulaciones el portafolio da retornos más positivos. Para poder analizar esto más a fondo, resultaría más práctico analizar los valores finales de las simulaciones.

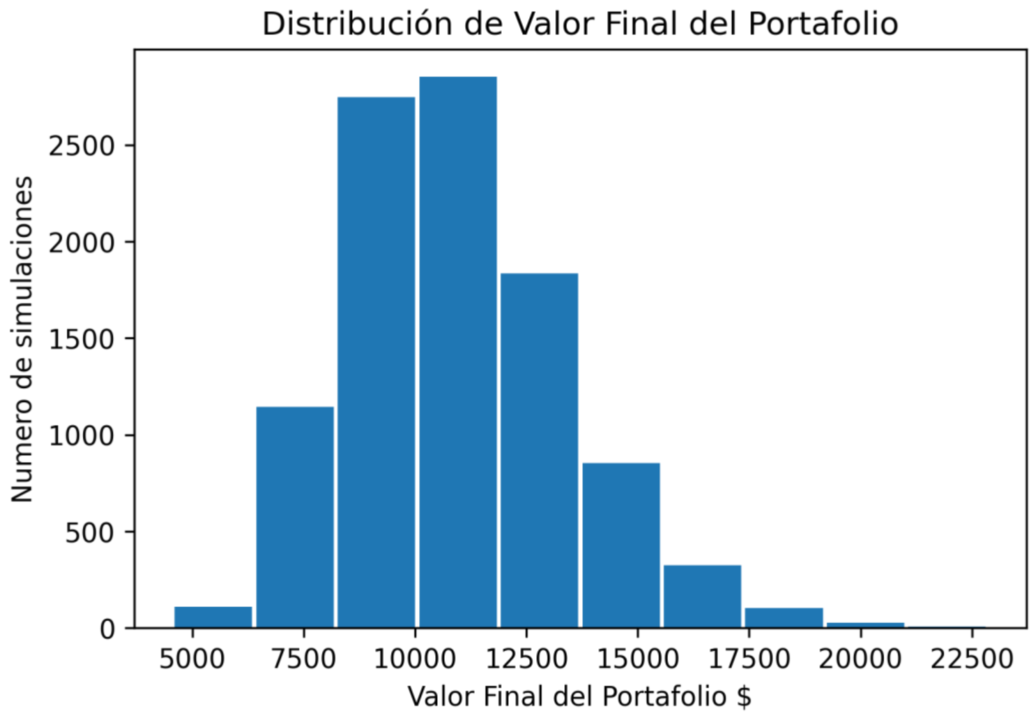

Figura 6: Distribución de las valuaciones finales del portafolio.

La figura muestra que más de 5000 (más del 50%) simulaciones dejan la valuación final del portafolio entre $8,000 y $12,000. En este caso 6,078 (aproximadamente 61%) simulaciones dieron un valor final mayor a $10,000; un retorno positivo. Este histograma tiene una cola derecha muy pesada lo que quiere decir que algunas simulaciones dieron un valor final muy alto para el portafolio. Analizando más a detalle los valores finales su valor esperado es de $10,914.76 (ganancia esperada de $914.76) lo que representa un retorno del 9.15%. La varianza es 6132828.84 y la desviación estándar es de $2,476.45. Este valor no es tan grande comparado con algunas ganancias y refleja el dato ya mencionado que alrededor del 50% de las valuaciones finales están entre $8,000 y $12,000. La valuación final máxima alcanzada es de $22,857.99 lo que representa una ganancia de $12,857.99 (retorno del 128.58%). La valuación más baja que fue proyectada alcanzó un valor de $4,557.72. Esto representa una pérdida de $5,442.28; un retorno de -54.42%. De nuevo, los retornos de ganancia son de mayor magnitud a esos de pérdidas. En la figura 5 se resaltaron los cuantiles del 5% y del 95%. La valuación final del 5% es de $7,324.83 que representa un retorno de -26.75% (una pérdida de $2,675.17). Es decir que del 5% de los “peores” escenarios para este portafolio esperan recibir un retorno negativo de esa magnitud o mayor; esperan que su valuación final sea menor o igual a $7,324.83. Por su parte, la valuación final del 95% fue de 15395.78 lo que representa una ganancia de $5395.78 (retorno de %). Es decir que en el 5% de los “mejores” escenarios se espera una ganancia/retorno mayor o igual a las últimas cifras. Se puede generar un box plot para entender de mejor manera la distribución de las valuaciones finales.

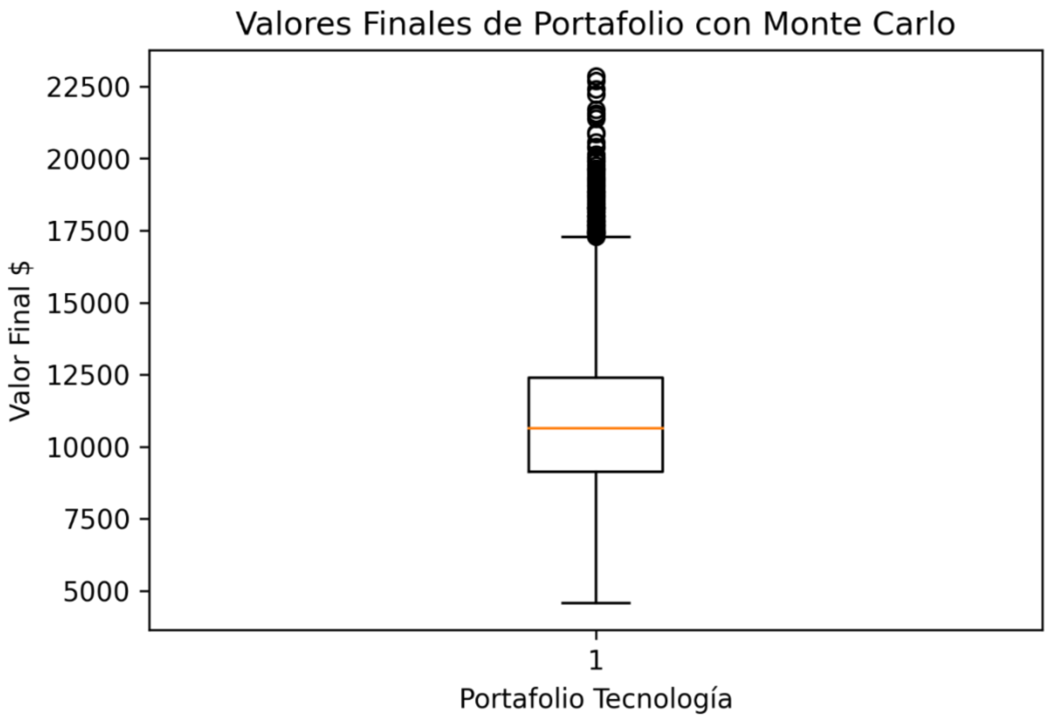

Figura 7: Box Plot de la distribución de las valuaciones finales del portafolio.

De este gráfico se tiene que el valor máximo alcanzado es de $22,857.99 y el valor mínimo es de $4,557.72. El primer cuantil es de $9,134.11 (el 25% de las simulaciones se acumulan aquí). Esto representa un retorno de -8.66%. La media es de $10,914.56 y la mediana es de $10,631.50, representa un retorno del 6.32%. Al ser mayor a $10,000 indica que más del 50% de las simulaciones tienen retornos positivos. Ya que la media es mayor a la mediana estos datos tienen una distribución asimetría positiva o a la derecha. El tercer cuantil es de $12,394.59; (el 75% de las simulaciones se acumulan aquí) que representa un retorno del 23.95%. Esto nos deja con un *IQR* de $3,260.98. Esto indica que el 50% de las simulaciones se encuentran entre una valuación de $9,134.11 y $12,394.59; Hay una diferencia de $3,260.98 entre los “peores” 25% de escenarios y los “mejores”. Los círculos en la figura representan los valores atípicos. (a una distancia de 1.5*IQR de los Q1 y Q3). Para las valuaciones de mayor tamaño se consideran valores atípicos a partir de $17,286.06. Hay 125 valores atípicos. Esto coincide más o menos con la cola pesada a la derecha que se puede apreciar en la figura 6. Una forma más sencilla y práctica de ver los valores finales es en términos del retorno final de la inversión inicial al portafolio.

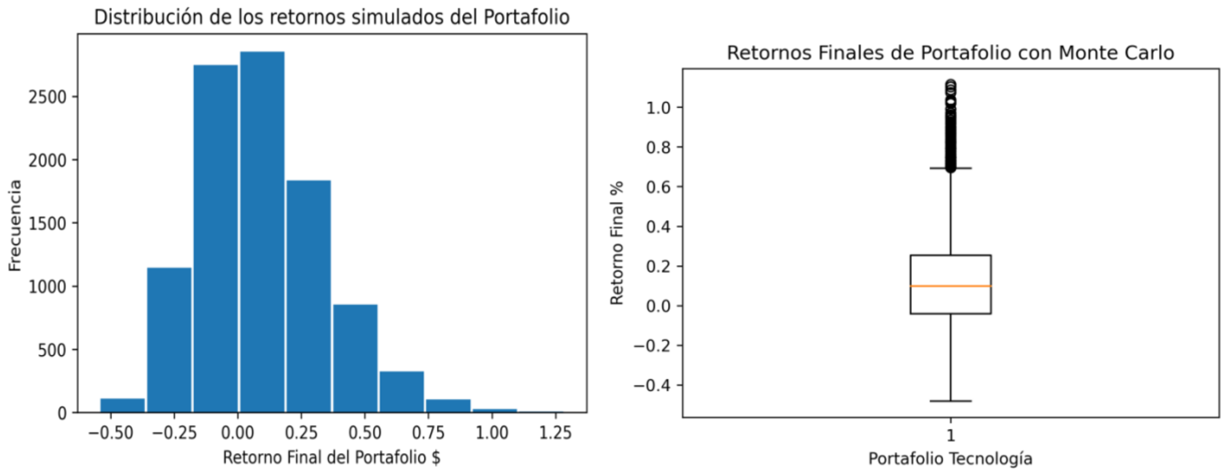

Figura 8: izq.: Distribución de las retornos finales del portafolio. Der: Box Plot de los retornos del portafolio.

Nota: el box plot no corresponde a estos valores. Esto se debe a que al guardar la imagen se generó otra simulación en Python borrando los datos con los que se trabajaba. Los datos correctos son: (Ver Anexo 3.) Estos datos se usan en el análisis.

Se observa la misma distribución de los datos que en la figura 6, pero ahora esto se muestra en términos de retornos. El retorno medio es de 9.15%, si se considera que la tasa libre de riesgo en estados unidos a 1 año es de 0.26% (tomado el 3 de dic de 2021) este portafolio sería una muy buena inversión (hay que tomar en cuenta que esta es una inversión con riesgo y pueden existir retornos negativos.) La desviación estándar de los retornos es de 0.247658 (x100%). El retorno mínimo es de -54.42% y el máximo de 128.58%. El 25% de las simulaciones se acumulan en un retorno de -8.66% (primer cuantil), la mediana se encuentra en un retorno de 6.32% y el tercer cuantil se alcanza en un retorno de 23.96%. El rango inter cuartil es de 32.60%, (tomando el retorno negativo). Si se toman ambos en valor absoluto el IQR es de 15.29%. Existen valores atípicos a partir de un retorno de 72.86%. Se reitera que, en esta simulación, este portafolio presenta muy buenos retornos y una mayor probabilidad de un retorno positivo. En este caso se recomendaría invertir buen capital en estos activos en vez de generar intereses en el banco o con bonos de gobierno. Este portafolio también es efectivo en contra de la inflación en Estados Unidos donde la tasa de inflación anual medida por el Índice de Precios al Consumidor para octubre de 2021 alcanzó el 6.2%.

Conclusiones

Los Métodos de Monte Carlo, permiten simular series de tiempo a futuro. Esto es particularmente útil para hacer predicciones sobre precios y valuaciones de activos en el mercado financiero. El poder considerar todos o la mayoría de los escenarios que pueden ocurrir, dadas condiciones históricas e iniciales, puede se útil para optimizar las estrategias de inversión de distintos jugadores en los mercados. Se reconoce que el modelo está limitado a los métodos de simulación seleccionados (movimiento Browniano geométrico y Caminata Aleatoria) así como las condiciones iniciales dadas y el plazo de tiempo (mejor para mediano y corto plazo). El mercado financiero es volátil y realmente impredecible pero nunca está de más estar consciente de las posibilidades que pueden existir a futuro.

Referencias

Campbell, J. Y., Lo, A. W., & MacKinlay, A. C. (2011). *The econometrics of Financial Markets*. New Age International (P) Ltd., Publ.

DeSilver, D. (2021, November 24). *Inflation has risen around the world, but the U.S. has seen one of the biggest increases*. Pew Research Center. Retrieved December 5, 2021, from https://www.pewresearch.org/fact-tank/2021/11/24/inflation-has-risen-around-the-world-but-the-u-s-has-seen-one-of-the-biggest-increases/.

Robert, C., & Casella, G. (2010). *Introducing Monte Carlo methods with R*. Springer New York.

U.S. Department of the Treasury. Daily Treasury Yield Curve Rates. (2021, December 3). Retrieved December 5, 2021, from https://www.treasury.gov/resource-center/data-chart-center/interest-rates/pages/textview.aspx?data=yield.

Anexos

Anexo 1: Código Simulación de Acción en Python

Código Python.

Obtención de datos

```python
#Se hará con Movimiento Browniano Geometrico
#Es la ecuacion diferencial estocastica
# dS/S = µdt + sigmadWt
#con Wt un proceso de Wiener N(0, sigma^2 = t)

#Librerias
#Importamos paquetes
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy as sp

from scipy import stats
%matplotlib inline

#se usará yfinance para descargar los datos de las acciones de yahoo finance
import yfinance as yf

#otros paquetes de ser necesario
import statistics

#Descarguemos el historial de la accion
aapl = yf.Ticker("AAPL")
actual_hist = aapl.history(start="2019-12-29", end="2020-12-30", auto_adjust=False)

#Vemos si los imports funcionan revisando algunos valores
print(actual_hist.iloc[[0,-1]])

#Revisemos el plot del historial de la acción
fig = plt.figure(figsize=(12,8))
plt.plot(actual_hist['Close'])
plt.xlabel('Fechas')
plt.ylabel('$ Precio $')
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
```

Print que costa en ver los datos de la acción el primer día y el ulyimo.

```
                  Open        High  ...  Dividends  Stock Splits
Date                                ...
2019-12-30   72.364998   73.172501  ...        0.0           0.0
2020-12-29  138.050003  138.789993  ...        0.0           0.0
```

Movimiento Browniano

```
#Movimiento Browniano y Monte Carlo SIUUUU

#Cuantos "pasos" hay en la simulacion, para cuantos dias (1 paso = 1 dia)
n_t = len(actual_hist)
print("Numero de dias: ", n_t)

#Numero de simulaciones a realizar
#Posibles direcciones que tomará la accion en ese periood de tiempo
n_mc = 10000

#inicializa el data frame donde se guardan las simulaciones de monte carlo
St = pd.DataFrame(0., index=actual_hist.index, columns=list(range(1,n_mc+1)))
#primer valor para todas las simulaciones es el valor de cierre del ultimo dia
St.iloc[0] = actual_hist['Close'].iloc[0] #dia 0

#Ahora veamos la volatilidad (Anualizada)
# sigma dos volatilidad anualizada de APPLE historico
#sigma 1 es asumida
sigma1 = 0.25
sigma2 = 0.2379
print(sigma2)

#Usar tasa libre de riesgo (black schultz)
#mu1 se asume
mu1 = 0.08
mu2 = 0.02 #risk free rate de EEUU
mu3 = 0.068 #risk free en Mex

#Convertir tamaño de paso de 1 dia a terminos de años
#Son dos años de info entonces se hace:
dt = 2./(n_t - 1)
#imprimimos la volatilidad diaria
print("Volatilidad diaria: ", sigma2*np.sqrt(dt))
```

Print de número de días.

```
Numero de dias:  253
0.2379
```

Volatilidad

```
In [5]: print("Volatilidad diaria: ", sigma2*np.sqrt(dt))
Volatilidad diaria:  0.021193816483655252
```

Simulación Montecarlo

```python
#Simulaciones de Monte Carlo utilizando valores aleatorios de numpy

for i in range(1, n_t):
    dS_2_S = mu2*dt + sigma2*np.sqrt(dt)*np.random.randn(n_mc)
    St.iloc[i] = St.iloc[i-1] + St.iloc[i-1]*dS_2_S

#Summary Statistics del último dia
print("Count:", len(St.iloc[252]))
print("Maximo: ",np.max(St.iloc[252]))
print("Minimo: ", np.min(St.iloc[252]))
print("IQR: ", sp.stats.iqr(St.iloc[252]))
print("Media:", np.mean(St.iloc[252]))
print("Desv Est: ", np.std(St.iloc[252]))
print("Mediana:", np.median(St.iloc[252]))

#graficas
plt.hist(St.iloc[252], color='rebeccapurple')
plt.title("Simulación")
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.xlabel('Valor de la acción',fontsize=15)
plt.ylabel('Simuaciones',fontsize=15)

#visualizemos lo creado
fig = plt.figure(figsize=(16,18))
axl = fig.add_subplot(111)
```

Print de estadisticas

```
Count: 10000
Maximo:  296.3676215235504
Minimo:  19.724883434428573
IQR:  32.833221858077295
Media: 76.10827779505155
Desv Est:  26.29977428601136
Mediana: 72.05845072887894
```

10,000 iteraciones con su respectiva graficas

```
#cambiar tamaño para ver cuantas sims deseamos mostrar
for i in np.random.choice(np.array(range(1, n_mc+1)), size = 10000):
    axl.plot(St[i], 'grey', lw=0.5)
    #print(St[i])

#Graficas
axl.plot(actual_hist['Close'], 'r', lw=1)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.xlabel('Fecha')
plt.ylabel('$ Precio')
plt.show()
```

Comparación de valores teóricos con valores obtenidos Monte Carlo

```
#Comparemos Aproximaciones de MC con Valores teoricos
#Mean from monte carlo
St_mc_mean = St.mean(axis=1)
#Theoretical Mean
St_th_mean = actual_hist['Close'].iloc[0]*np.exp(mu3*np.arange(n_t)/n_t*2.)
#convertir el teorico en un data frame
St_th_mean = pd.DataFrame(St_th_mean, index = St_mc_mean.index)

#Imprimiendo utlimos means de MC y Teoria
print("Expected Value from MC: ", St_mc_mean.iloc[-1])
print("Expected Value Theoretical: ", St_th_mean.iloc[-1])


#duda de error absoluto
for i in range(1,n_t):
    (St_mc_mean.iloc[i]-St_th_mean.iloc[i])


#Grafiquemos la media como una funcion del tiempo
fig = plt.figure(figsize=(16,8))
axl = fig.add_subplot(111)
plt.plot(St_mc_mean)
plt.plot(St_th_mean)
plt.xlabel('Fecha')
plt.ylabel('Valor Esperado')
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.legend(['MC Media', 'Media Teorica'])
```

```
Expected Value from MC:  76.10827779505155
Expected Value Theoretical:  0    83.452421
```

Anexo 2: Código Simulación de Portafolio en Python

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import datetime as dt
from pandas_datareader import data as pdr

# Obtener datos del portafolio
def get_data(stocks, start, end):
    stockData = pdr.get_data_yahoo(stocks, start, end)
    stockData = stockData['Close']
    returns = stockData.pct_change()
    meanReturns = returns.mean()
    covMatrix = returns.cov()
    return meanReturns, covMatrix

#Definimos nuestro portafolio y el numero de datos historicos que queremos
stocks = ['AAPL', 'T', 'AMZN']
endDate = dt.datetime.now()
startDate = endDate - dt.timedelta(days=300)

meanReturns, covMatrix = get_data(stocks, startDate, endDate)

#Generamos pesos aleatorios
weights = np.random.random(len(meanReturns))
weights /= np.sum(weights)

# Simulacion de MC
# numero de simulaciones
sims = 10000
#dias de la simulacion a futuro
T = 365

meanM = np.full(shape=(T, len(weights)), fill_value=meanReturns)
#determinamos el "drift" de la simulacion
meanM = meanM.T

portfolio_sims = np.full(shape=(T, sims), fill_value=0.0)

#Valor inicial de portafolio
initialPortfolio = 10000

for m in range(0, sims):
    #generacion de variables aleatorias normales
    Z = np.random.normal(size=(T, len(weights)))
    #Cholesky
    L = np.linalg.cholesky(covMatrix)
    #Retoronos diarios correlacionados de las acciones
    dailyReturns = meanM + np.inner(L, Z)
    #vamos acumulando retornos a través deltiempo
    portfolio_sims[:,m] = np.cumprod(np.inner(weights, dailyReturns.T)+1)*initialPortfolio

    #Graficamos
fig,ax=plt.subplots()
ax.plot(portfolio_sims,color="grey", alpha=0.25)
plt.ylabel('Valuacion ($)')
plt.xlabel('Dias')
plt.title('Simulacion de un Portafolio de Tecnología')
ax.plot(np.arange(0,365),portfolio_sims.mean(axis=1), color="orange")
ax.plot(np.arange(0,365),np.quantile(portfolio_sims,axis=1, q=0.05), color="cyan")
ax.plot(np.arange(0,365),np.quantile(portfolio_sims,axis=1, q=0.95), color="lime")
#plt.savefig('sims.png', dpi=300, bbox_inches='tight')
```

```python
#Histograma de últimos precios
plt.hist(portfolio_sims[T-1,:], rwidth=0.95)
plt.xlabel('Valor Final del Portafolio $')
plt.ylabel('Numero de simulaciones')
plt.title('Distribución de Valor Final del Portafolio')
plt.savefig('sims.png', dpi=300, bbox_inches='tight')

#Todos los ultimos precios simulados
print("Count:", len(portfolio_sims[T-1,:]))
print("Media:", np.mean(portfolio_sims[T-1,:]))
print("Varianza:", np.var(portfolio_sims[T-1,:]))
print("Stdev:", np.std(portfolio_sims[T-1,:]))
print("Max:", np.max(portfolio_sims[T-1,:]))
print("Min:", np.min(portfolio_sims[T-1,:]))

minimo=np.min(portfolio_sims[T-1,:])
maximo=np.max(portfolio_sims[T-1,:])

maxgain= maximo-initialPortfolio
maxloss= minimo-initialPortfolio

mean=np.mean(portfolio_sims[T-1,:])
expected = mean-initialPortfolio

print("Ganancia Maxima:", maxgain)
print("Perdida Maxima:", maxloss)
print("perdida/ganancia esperada: ", expected)

#Pandas decribe
#Ponemos los ultimos precios en un DF
valores=pd.DataFrame(portfolio_sims[T-1,:], columns=['valor_final']);
valores.describe()

#Box Plot de Los Valores Finales
plt.boxplot(portfolio_sims[T-1,:])
plt.title('Valores Finales de Portafolio con Monte Carlo')
plt.xlabel('Portafolio Tecnología')
plt.ylabel('Valor Final $')
plt.savefig('sims.png', dpi=300, bbox_inches='tight')

#Veamos cuantiles de los ultimos valores
cuantil_025=np.quantile(portfolio_sims[T-1,:], 0.975)
cuantil_10=np.quantile(portfolio_sims[T-1,:], 0.9)
cuantil_05=np.quantile(portfolio_sims[T-1,:], 0.95)
cuantil_01=np.quantile(portfolio_sims[T-1,:], 0.99)
print(cuantil_025)
print(cuantil_10)
print(cuantil_05)
print(cuantil_01)

#percentiles
per5=np.percentile(valores["valor_final"],5)
perdida = per5-initialPortfolio
print(per5)
print(perdida)

per95=np.percentile(valores["valor_final"],95)
ganancia = per95-initialPortfolio
print(per95)
print(ganancia)

#grafiquemos retornos
returns.describe()
#returns como %
returns=(valores["valor_final"]-initialPortfolio)/initialPortfolio
plt.hist(returns,rwidth=0.95)
plt.xlabel('Retorno Final del Portafolio $')
plt.ylabel('Frecuencia')
plt.title('Distribución de los retornos simulados del Portafolio')
plt.savefig('sims.png', dpi=300, bbox_inches='tight')

#Box Plot Retornos
plt.boxplot(returns)
plt.title('Retornos Finales de Portafolio con Monte Carlo')
plt.xlabel('Portafolio Tecnología')
plt.ylabel('Retorno Final %')
plt.savefig('sims.png', dpi=300, bbox_inches='tight')
```

Anexo 3: Nota Datos Retornos

count	10000.000000
mean	0.091476
std	0.247658
min	−0.544228
25%	−0.086589
50%	0.063150
75%	0.239459
max	1.285799