

# **Optimización de Retornos Ajustados por Riesgo en Portafolios de Grandes Dimensiones**

**Proyecto Integrador**

**Pregrado – Línea de Énfasis**

**Integrantes:**

**Daniel García García**

**Ricardo José Garzón Arias**

**Santiago Alberto Moreno Quevedo**

**Profesores:**

**Edwin Nelson Montoya Múnера**

**Henry Laniado Rodas**

**José Antonio Solano Atehortúa**

**Universidad EAFIT**

**2021**

# Tabla de contenido

---

- [Tabla de contenido](#)
- [Introducción](#)
- [Marco teórico](#)
- [Desarrollo metodológico](#)
  - [Entendimiento del problema](#)
    - [Supuestos](#)
  - [Análisis exploratorio de datos](#)
    - [Entendimiento de los datos](#)
    - [Preparación de los datos](#)
    - [Análisis descriptivo](#)
      - [Gráficas de precios](#)
      - [Gráficas de retornos](#)
      - [Análisis descriptivo multivariado](#)
        - [Descripción de precios](#)
        - [Descripción de retornos](#)
      - [Análisis de regresiones y correlaciones](#)
        - [Matriz de correlación](#)
        - [Regresiones lineales para cada acción respecto a las demás](#)
      - [Análisis de la matriz de covarianzas](#)
      - [Identificación de outliers](#)
        - [Con YNDX](#)
        - [Sin YNDX](#)
      - [Clustering no supervisado con K-Means](#)
      - [Clasificación supervisada con K-Nearest Neighbors](#)
  - [Selección de modelos](#)
    - [Modelos](#)
      - [Portafolio de pesos iguales](#)
        - [Entrenamiento](#)
        - [Validación](#)
      - [Portafolio de varianza mínima](#)
        - [Entrenamiento](#)
        - [Validación](#)
      - [Portafolio de varianza mínima con shrinkage de Ledoit & Wolf](#)
        - [Entrenamiento](#)
        - [Validación](#)
    - [Comparación de modelos](#)
      - [Entrenamiento](#)
      - [Validación](#)
  - [Análisis y conclusiones](#)
- [Tecnología](#)

- [Desarrollo del proyecto](#)
- [Despliegue del proyecto](#)
- [Otros aspectos tecnológicos](#)
  - [Fuentes de datos](#)
  - [Ingesta de datos](#)
  - [Almacenamiento](#)
- [Ambiente de procesamiento](#)
- [Aplicaciones](#)
- [Conclusiones generales del proyecto](#)
- [Referencias](#)

## Introducción

---

Este proyecto consiste en hacer un análisis de diferentes herramientas para determinar si se pueden mejorar los retornos ajustados por riesgo, donde el riesgo está dado por la desviación estándar de los retornos  $\sigma$ , de un portafolio grande de acciones, por medio de técnicas estadísticas que disminuyan los errores computacionales en las operaciones necesarias para optimizar el retorno ajustado por riesgo. Para lograr esto, se necesita una gran cantidad de datos históricos de diferentes acciones, con los cuales se construirán 50 portafolios de acciones, cada uno con 300 acciones y se hará un análisis de sus precios y rendimientos históricos a lo largo de varios años, para determinar un portafolio de mínima varianza, que posteriormente será mejorado por medio de técnicas como el shrinkage de la matriz de covarianzas de Ledoit & Wolf (2003) y la matriz generalizada de Moore-Penrose (Penrose & Todd, 1955). El objetivo final, con base en estos resultados, será construir un portafolio de mínima varianza, que, dada una lista de activos y sus retornos históricos, tenga un mejor desempeño que el portafolio de distribución de pesos iguales para todos los activos y que el portafolio de mínima varianza calculado con los métodos de covarianza y matriz inversa habituales.

## Marco teórico

---

En 1952, Harry Markowitz propuso una teoría que consiste en que los inversionistas pueden definir un nivel de riesgo ( $\sigma$ ) que están dispuestos a asumir al invertir en un conjunto de activos y, para este nivel de riesgo establecido, existirán diferentes combinaciones (portafolios) de ponderaciones para cada uno de los activos que tendrán distintos retornos esperados con la misma desviación estándar. Con base en esto, Markowitz determinó que existe un conjunto de portafolios eficientes, que son los portafolios que mayor retorno esperado tienen para cada nivel de riesgo dado y, un inversionista racional que tenga un nivel de riesgo definido que está dispuesto a asumir, debería escoger siempre el portafolio eficiente para este nivel de riesgo y no otro, ya que todos los demás portafolios tendrían el mismo riesgo, pero con menor retorno esperado. Esto es lo mismo que decir que, para un retorno esperado determinado, el inversionista debe escoger el portafolio que le provea ese retorno con el menor riesgo posible. Uno de los portafolios que se encuentra en esta frontera eficiente es el portafolio de mínima varianza, que es el portafolio cuya combinación ponderada de activos resulta en la menor varianza posible para ese conjunto determinado de activos (Markowitz, 1952).

A pesar de la teoría clásica propuesta por Markowitz, el modelo continuo del tiempo para el proceso de cotización de acciones ha evolucionado a lo largo de los años, y en 1982 Fernholz & Shay propusieron un modelo ventajoso para analizar eventos a largo plazo o asintóticos, por medio de un modelo logarítmico, debido a que los procesos del precio logarítmico se asemejan a procesos aleatorios lineales ordinarios en lugar de uno exponencial. Todo esto se debe a que la definición tradicional de las carteras de acciones sólo da como resultado el valor de la inversión

en cada acción, y no los pesos. Markowitz en su teoría sólo considera tiempos discretos, mientras que en la teoría logarítmica se considera también el tiempo continuo. Lo que se busca ahora no es sólo el conjunto de portafolios eficientes, que son los que mayor retorno esperado tienen para cada nivel de riesgo dado, sino que también se busca maximizar el valor esperado de  $\text{Log } Z_{\pi}(t)$ . Esto para producir el portafolio con mayor valor asintótico. Es decir, el portafolio donde se tengan en cuenta las ponderaciones del mercado, como la tasa de crecimiento, ya que con esto se puede determinar el comportamiento de alguna cartera a largo plazo (Fernholz, 2002). Para calcular los pesos del portafolio de mínima varianza, se debe usar la siguiente fórmula (Breaking Down Finance, n.d.):

$$w_{MV} = \frac{\Sigma^{-1}\mathbf{1}}{\mathbf{1}^T \Sigma^{-1}\mathbf{1}}$$

donde,

$\Sigma^{-1}$  es la matriz inversa de la matriz de covarianzas de orden  $n \times n$   
 $\mathbf{1}$  es un vector de 1s de orden  $n \times 1$   
 $n$  es el número de activos del portafolio

Bai & Shi (2011) mencionan que la matriz de covarianzas es fundamental en la selección de portafolios, pero que tiene varias falencias, especialmente cuando las dimensiones de ésta son muy grandes o cuando se tienen activos que observaciones en los datos. Esto puede llevar a que la matriz de covarianzas no sea de rango completo y que su inversa no exista. Por otro lado, si la matriz sí es invertible, su inversa no es un estimador insesgado de la inversa teórica. Finalmente, esta matriz puede ser muy volátil, implicando muchos cambios en los pesos óptimos a lo largo del tiempo. Para corregir esto, proponen usar un shrinkage a la matriz de covarianzas, como el de Ledoit & Wolf (2003).

## Desarrollo metodológico

---

### Entendimiento del problema

La diversificación es un concepto muy utilizado en la conformación de portafolios. La idea es conseguir un portafolio de activos (como acciones) al que se le pueda reducir el riesgo sin sacrificar rentabilidad (es decir, aumentar la rentabilidad ajustada por riesgo). Para lograr esto, hay que tener en cuenta que los activos pueden tener correlaciones entre sí (por ejemplo, es probable que las acciones del sector energético presenten movimientos similares) y sacar la varianza del portafolio completo, teniendo en cuenta las covarianzas. Esto hace que sea necesario usar la matriz de covarianza de los activos, que, cuando el número de activos es muy grande, la estimación de esta matriz no es muy precisa y los errores de estimación llevan a que los modelos tengan un desempeño inferior en la etapa de validación con datos fuera de la muestra original (Demiguel et al., 2009). Adicionalmente, las matrices de covarianza para muchos activos son matrices difíciles de invertir, lo cual puede llevar a errores computacionales, haciendo que sea preferible una estimación de esta matriz (Levina et al., 2008).

El problema que se quiere resolver consiste en hallar un modelo de rebalanceo de los pesos de las acciones de un portafolio grande, con el fin de minimizar la varianza del portafolio, para obtener una buena rentabilidad ajustada por riesgo, sin que los errores de estimación causados por el cálculo de matrices de covarianza grandes y su inversión afecten los resultados en el período de validación con datos externos a la muestra.

## Supuestos

Hay varios supuestos que se tienen que asumir para que los resultados sean válidos.

Inicialmente, se deben cumplir los supuestos de la teoría de portafolios de Markowitz, que, según Petters & Dong (2016) son los siguientes:

- *Inversionistas racionales*: esto quiere decir que los inversionistas siempre tomarán las decisiones financieras que les provean la mayor satisfacción.
- *Mercado en equilibrio*: el mercado está en competencia perfecta, lo cual implica que la oferta es igual a la demanda.
- *No es posible el arbitraje*: no existe ninguna oportunidad de arbitraje en el mercado (arbitraje se define como la compra y venta simultáneas de activos equivalentes en diferentes mercados para explotar una asimetría de precios y obtener una utilidad libre de riesgo (Wordnik, s.f.)).
- *Acceso a la información*: los participantes del mercado tienen acceso rápido a información confiable acerca de los activos.
- *Eficiencia del mercado*: los precios de mercado reflejan toda la información disponible del activo, incluyendo información pasada y expectativas a futuro.
- *Liquidex*: se puede comprar/vender cualquier número de unidades de un activo rápidamente.
- *No hay costos de transacción*: la compra/venta de activos tiene ningún costo asociado.
- *No hay impuestos*: todas las transacciones del mercado son libres de impuestos.
- *Préstamos*: todos los préstamos se pueden hacer a la tasa libre de riesgo.

Por otro lado, hay otros supuestos importantes que se tienen en cuenta en este problema:

- *Alto capital disponible*: es importante que el monto disponible para invertir en los activos que conforman el portafolio sea grande. Esto es necesario porque los pesos de los activos serán valores reales, pero el número de unidades de cada activo es un entero, lo cual implica que habrá un error de redondeo en el porcentaje del portafolio que en realidad se invierte en un activo, respecto al que el modelo arroja. Si el monto de dinero para invertir es grande, estos errores producto de tener que comprar unidades enteras de un activo se hacen insignificantes. Si el monto es pequeño, este redondeo puede implicar diferencias significativas entre los porcentajes teóricos y los realmente invertidos.
- *Historia disponible de activos*: el modelo necesita usar datos históricos para calcular un retorno esperado y una desviación estándar, por lo cual se necesita que los activos tengan información histórica disponible.

## Análisis exploratorio de datos

### Entendimiento de los datos

Los datos históricos de acciones tomados de <https://www.kaggle.com/martholi/stockminutedata/version/1> son datos históricos de acciones, aproximadamente entre el 2010 y finales de 2020.

Estos datos, según los metadatos del dataset, son minuto a minuto e incluyen los precios de apertura, cierre, máximo y mínimo, además del volumen transado para cada minuto para cada acción. Estos datos originalmente están en una carpeta con 1477 archivos, de los cuales 1474 son archivos en formato .parquet y cada archivo contiene la información histórica de los precios de las acciones, minuto a minuto.

Con base en esta información, se concluyó que el rebalanceo de portafolios no es algo que requiera una granularidad tan alta como minuto a minuto, especialmente teniendo en cuenta que, a pesar de que se supone que no hay costos de transacción, en la práctica sí los hay y la logística y los costos asociados a rebalancear un portafolio cada minuto serían temas difíciles de tratar. Por este motivo, una de las primeras transformaciones necesarias es pasar los datos de

minuto a minuto a datos diarios.

Por otro lado, para el rebalanceo de portafolios y los cálculos de volatilidades y retornos, no será necesario usar las variables de precio de apertura, precio máximo, precio mínimo y volumen, sino que solamente se usará el precio de cierre, que será suficiente para hacer los cálculos necesarios. Por este motivo, se eliminarán estas columnas para todas las acciones.

Otro aspecto importante de los datos es que no todas las acciones se transan en los mismos períodos de tiempo, por diferentes motivos como los siguientes:

- Hay acciones que son menos líquidas, lo cual implica que no se están comprando/vendiendo tan frecuentemente en los mercados. Esto lleva a que haya acciones que tienen muchos más datos históricos que otras, por el hecho de que las otras no se transan tanto. Esto lleva a que se den valores nulos.
- Hay acciones que fueron emitidas en bolsa después de cierta fecha, lo cual implica que antes de su fecha de emisión, no tendrán valores históricos. Esto llevará a que tengan muchos valores nulos cuando se analicen todas las acciones en el mismo rango de fechas.
- Hay acciones que pudieron haber sido sacadas de bolsa, lo cual implica que no tendrán precios históricos después de cierta fecha. Como en el caso anterior, esto llevará a que estas acciones tengan muchos valores nulos cuando se analicen en conjunto todas las acciones en el mismo rango de fechas.

## Preparación de los datos

Los datos se fueron ajustando de manera iterativa, a medida que se concluía que eran necesarios ciertos cambios. El proceso de preparación fue el siguiente:

1. Se tomaron los datos de Kaggle y se subieron a un bucket de S3 (`s3://proyecto-integrador-20212-pregrado`) ubicándolos en la zona *raw* creada en este bucket.
2. Se creó un código para ejecutar una operación ETL que toma los datos originales de la zona *raw* y los procesa, tomando el precio de cierre del último minuto de cada día, para usarlo como precio de cierre del día, convirtiendo así los datos de minuto a minuto en datos diarios. Estos datos diarios se escribieron en el mismo formato que los originales, una acción por archivo en un archivo `.parquet` y luego se cargaron a la zona *trusted* del bucket de S3 mencionado anteriormente.
3. Una vez se tenían los datos diarios de todas las acciones, se procedió a hacer otra operación de ETL en la que se hizo una limpieza de estos datos en diferentes pasos, de la siguiente manera:
  - 3.1. Se creó un DataFrame de datos, donde cada columna corresponde a los precios de cierre diarios de una acción específica, y el índice son las fechas históricas diarias. Aquí se descartaron las demás columnas de cada acción (precio de apertura, precio máximo, precio mínimo y volumen).
  - 3.2. Después de tener una matriz de precios inicial, se contaron los valores nulos de cada una de las acciones, que se pueden dar por los motivos explicados en [Entendimiento de los datos](#).
  - 3.3. Con base en estos valores y el hecho de que algunas de las acciones podrían no haberse emitido en el 2010, se tomó una fecha inicial diferente, que fue enero 1, 2014, como fecha inicial para los datos históricos de las acciones. Todos los datos anteriores a esta fecha se descartaron y quedaron 1697 días históricos para 1474 acciones.
  - 3.4. Después del cambio de fecha, se encontraron las acciones que más valores nulos tuvieran y se tomó la decisión de eliminar todas las acciones que tuvieran más de 63 días nulos en todos los datos históricos. Esto se hizo con el fin de que los valores nulos no tengan una incidencia importante en los resultados posteriores. Los 63 días se escogieron arbitrariamente, con base en el hecho de que un año equivale a 252 días en los que los mercados están abiertos, y 63 días sería un trimestre. Esto quiere decir que, si en los siete

años (aproximadamente) de datos, a una acción le falta más de un trimestre de datos, se elimina. Este paso resultó en que se eliminaron 1075 acciones de las 1474 originales, quedando 399 acciones elegibles.

3.5. Tras haber eliminado estas acciones, se llenaron los valores nulos restantes de la siguiente manera:

3.5.1. Se llenan todos los valores nulos con el último valor no nulo disponible. Esto tiene sentido cuando, por ejemplo, alguna acción no fue transada un día, lo cual implica que durante ese día, su precio fue igual que el último precio del día anterior.

3.5.2. Después de esto, es posible que todavía queden algunos valores nulos, por días en los que no existe un día anterior no nulo en los datos, sea por una coincidencia en la fecha de corte o porque la fecha de emisión de la acción fue posterior a la fecha inicial. Estos valores nulos se llenan tomando el valor siguiente no nulo en la historia.

3.6. Finalmente, se tomo la matriz de precios resultante, de dimensiones \$1697\text{times}399\$, cuyas columnas son las acciones y sus filas son las fechas históricas, y se cargó a un archivo .parquet en la zona *refined* del bucket de S3 anteriormente mencionado.

4. Para no hacer un análisis en los precios, que pueden tener escalas muy diferentes, y para buscar datos cuya distribución se acerque más a la normal, se creó una matriz de retornos de las acciones, donde cada dato es el cambio porcentual diario de cada acción entre dos días. Esta nueva matriz de retornos, de dimensiones \$1696\text{times}399\$, se cargó a otro archivo .parquet en la zona *refined* en el mismo bucket de S3 mencionado anteriormente.

5. Tras hacer un análisis exploratorio, descrito en detalle en [Análisis descriptivo](#), se hayaron otras propiedades de los datos que hicieron necesario volver a este paso de preparación/limpieza. Con base en los resultados obtenidos, dados datos atípicos y acciones/índices casi equivalentes, se procedió a eliminar los siguientes activos, por razones que se explican mejor en la sección del análisis:

- VTI: es un índice compuesto por muchas acciones
- SPY: es un ETF que sigue al S&P500, entonces contiene muchas de las acciones que ya están por otro lado
- DIA: es un ETF que sigue al índice Dow Jones, entonces contiene muchas de las acciones que ya están por otro lado
- IWN: es un ETF que sigue al índice Russel 2000 Value, entonces contiene muchas de las acciones que ya están por otro lado
- IJH: es un ETF que sigue al índice S&P MidCap 400, entonces contiene muchas de las acciones que ya están por otro lado
- IWF: es un ETF que sigue al índice Russel 1000 Growth, entonces contiene muchas de las acciones que ya están por otro lado
- GOOG: es la acción de Google clase C (la clase A, GOOGL, también está, pero no se eliminará)
- YNDX: tiene un valor muy atípico que debe ser un error

6. Una vez eliminados estos activos, se guardaron los datos filtrados en la zona *refined* del bucket de S3 y se procedió a hacer una separación de los datos en entrenamiento y validación, usando el 80% de los datos para el entrenamiento. Esto quiere decir que las matrices de retornos y de covarianzas que se usarán tomarán en cuenta solamente los datos hasta una fecha específica, para validar los rendimientos después. Esto resultó en una matriz de retornos de entrenamiento de dimensiones \$1356\text{times}391\$

7. Después de separar los datos en entrenamiento y validación, se crearon 50 portafolios, cada uno con 300 activos escogidos de manera aleatoria de entre los 391 activos totales. Cada uno de estos portafolios fue cargado en la zona *refined* del bucket de S3 mencionado anteriormente.

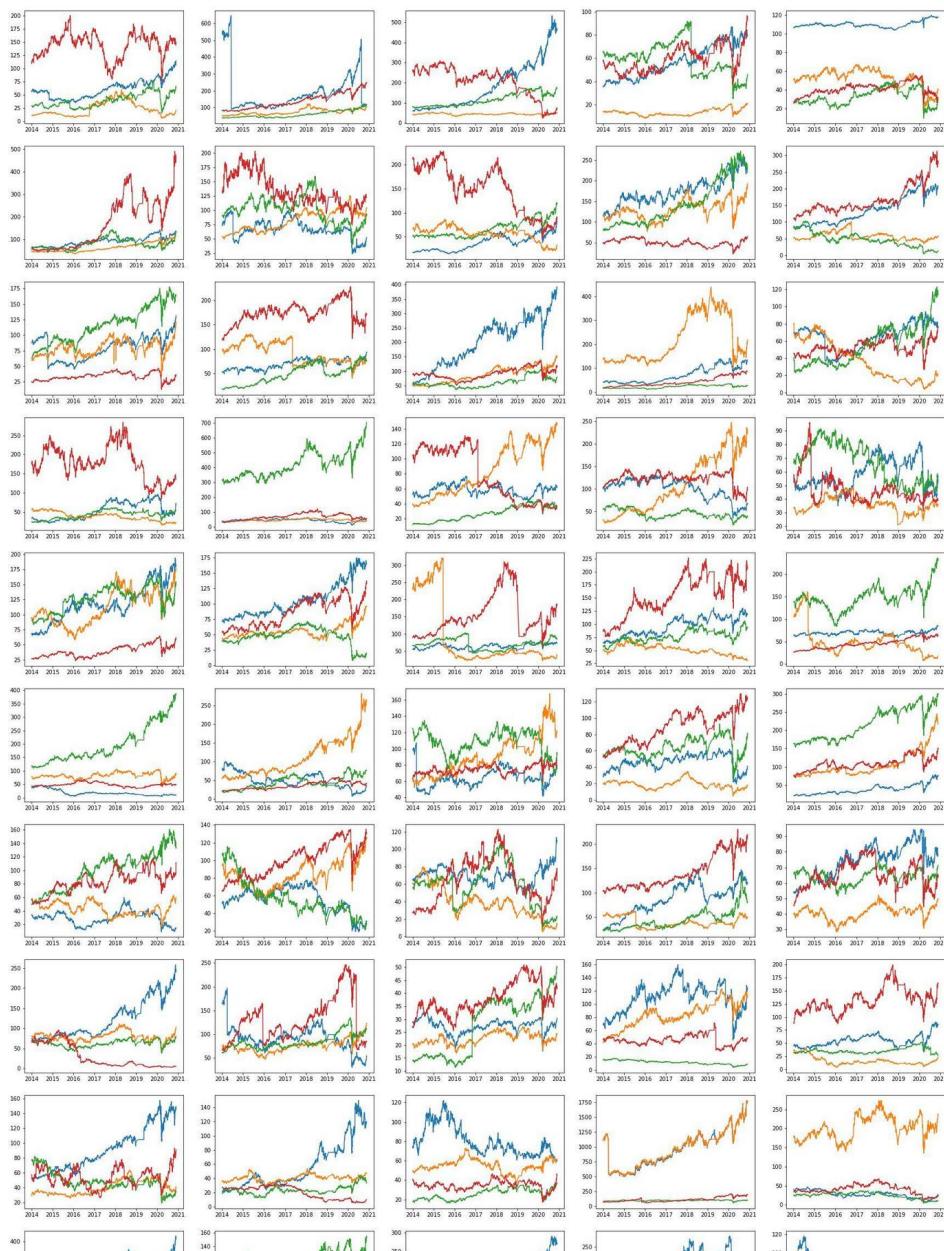
8. A cada uno de los portafolios aleatorios se le calcularon la matriz de covarianzas con el método habitual y la matriz de covarianzas usando el shrinkage de Ledoit & Wolf. Estas matrices también se cargaron en la zona *refined* del bucket de S3.

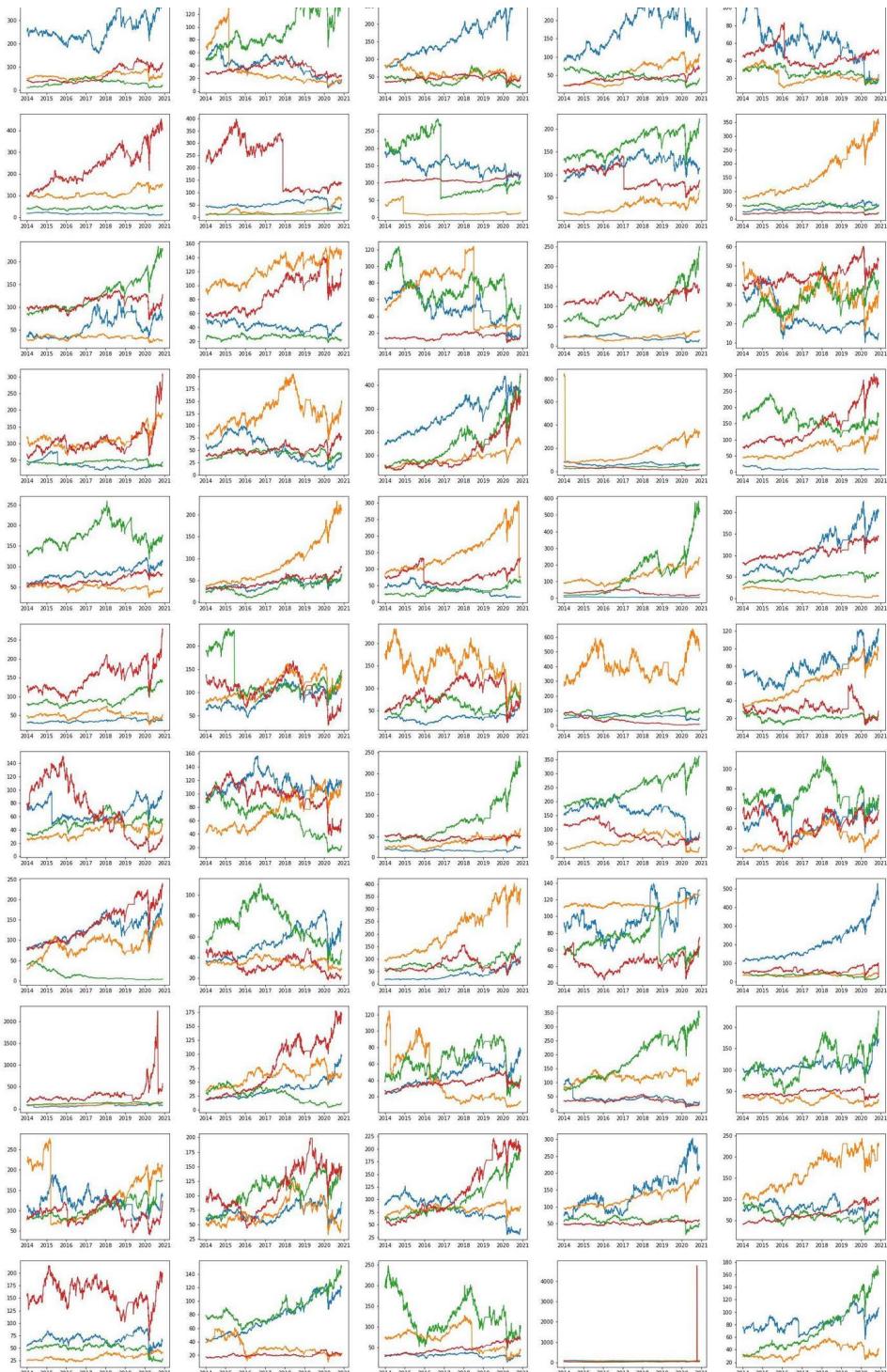
## Análisis descriptivo

Se hicieron diferentes análisis a los datos, los cuales fueron útiles para obtener más información de los mismos y ayudaron a filtrar algunos datos que podían causar resultados inesperados.

### Gráficas de precios

Se graficaron los comportamientos históricos de los precios de la siguiente manera:



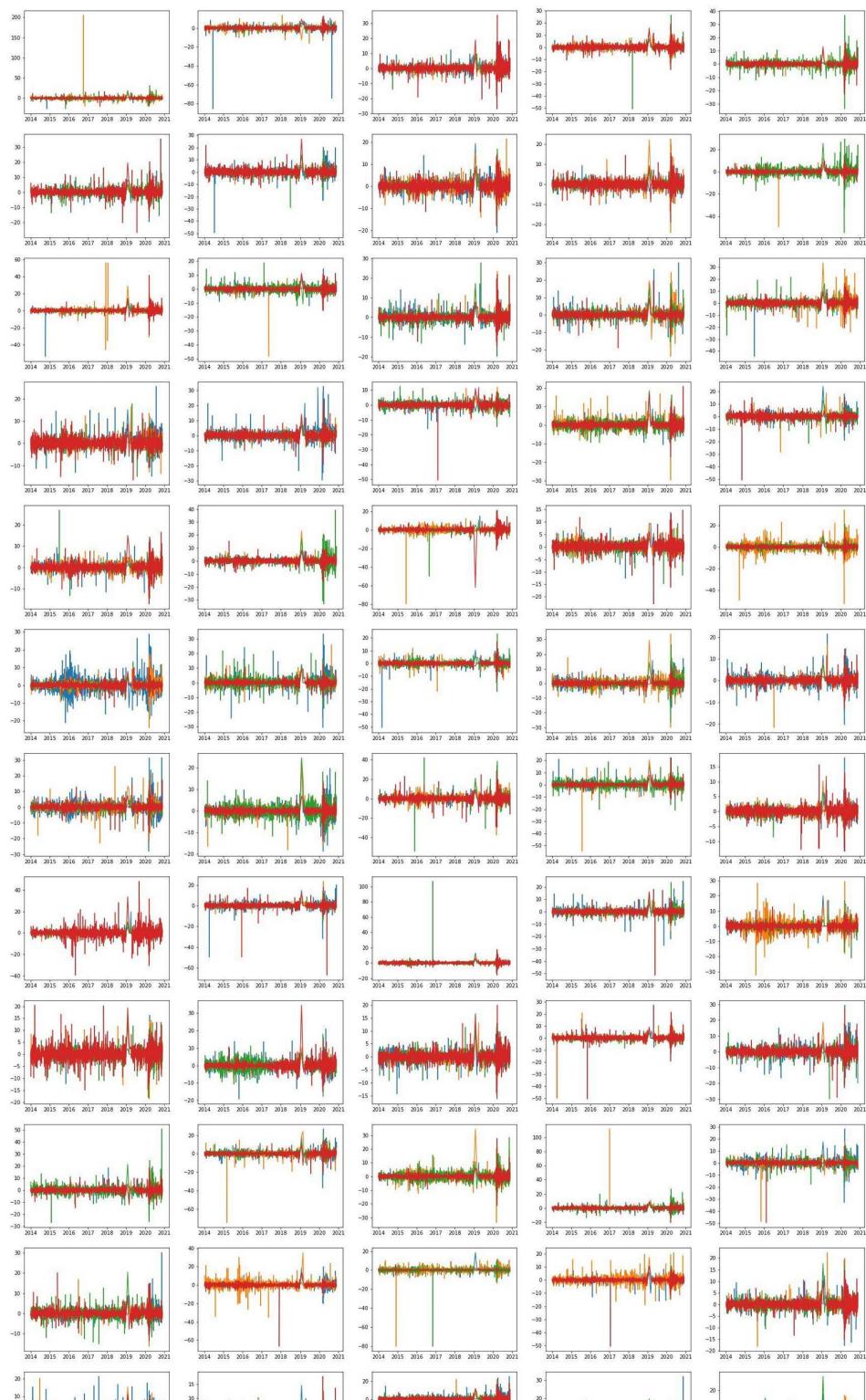


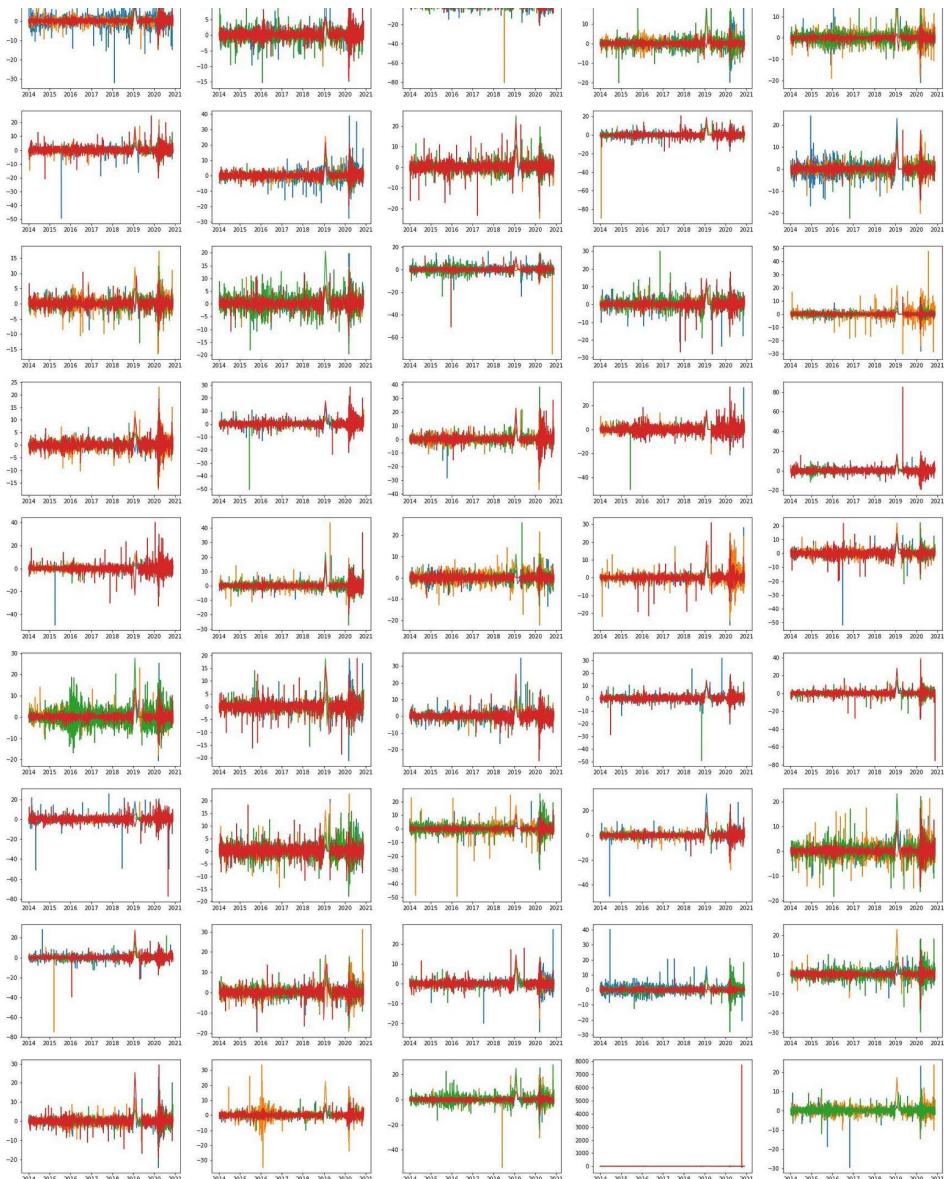
En la imagen se pueden ver 100 gráficas en total, repartidas en 20 filas y 5 columnas, donde cada gráfica tiene los precios históricos de cuatro acciones diferentes de las 399 acciones totales (exceptuando una gráfica que tiene sólo tres).

Aquí se puede ver que los precios son muy diferentes entre todas las acciones y que presentan particularidades de las series de tiempo, como tendencias.

## Gráficas de retornos

De la misma manera que se graficaron los precios, también se graficaron los retornos diarios de todas las acciones, usando el mismo formato de 100 gráficas repartidas en 20 filas y 5 columnas.





En estas gráficas de retornos se puede ver que las acciones ya se parecen mucho más y que su comportamiento se parece más al de una distribución normal. Esto facilita el análisis posterior, ya que los datos quedan en escalas similares y con distribuciones estadísticas parecidas.

## Análisis descriptivo multivariado

### Descripción de precios

Se hizo una descripción de las principales variables estadísticas para la matriz de precios. En la imagen no se muestran los datos completos, pero éstos se pueden observar en el código que se adjunta ([https://github.com/samorenog/proyecto\\_integrador\\_20212/blob/master/Notebooks/Exploratory\\_Analysis.ipynb](https://github.com/samorenog/proyecto_integrador_20212/blob/master/Notebooks/Exploratory_Analysis.ipynb)).



Acción con máxima correlación	Coefficiente de correlación máximo	Acción con mínima correlación	Coefficiente de correlación mínimo
VTI	SPY	0.992008	-0.331665
SPY	DIA	0.970871	-0.342634
IWN	IJH	0.956508	-0.334154
GOOGL	GOOG	0.933488	-0.007715
IJH	DIA	0.921494	-0.330763
...	...	...	...
KDP	IJH	0.261576	-0.067199
AAPL	A	0.244015	0.000000
AA	A	0.188259	0.000000
AGG	AES	0.182168	-0.100269
YNDX	DTE	0.105906	-0.096135

Estos resultados llevaron a que se conociera más información sobre los datos, como el hecho de que no solamente hay acciones, sino también ETFs sobre índices bursátiles (un ETF es un activo que busca seguir un índice bursátil, un commodity, etc. (Investopedia, 2021)). Como los índices bursátiles están compuestos de varias acciones, invertir en estos y además en las acciones individuales que los componen puede tener como resultado que se esté alocando inesperadamente más capital del deseado en el mismo activo. Adicionalmente, como se puede ver en la tabla, estos índices tienen correlaciones muy altas entre sí, probablemente porque están compuestos por muchas acciones, que en conjunto se mueven de manera similar. Por estos motivos, estos índices se eliminaron, como se explica en . Por otro lado, se encontró que existían dos acciones diferentes para Google, una clase A y la otra clase C, que también se mueven de manera muy similar, como se puede ver en la tabla. Por este motivo, también se eliminó la acción clase C, que es GOOG.

En el análisis de mayores correlaciones negativas (en valor absoluto) no se obtuvieron resultados suficientemente significativos como para hacer cambios:

Acción con máxima correlación	Coefficiente de correlación máximo	Acción con mínima correlación	Coefficiente de correlación mínimo
IEF	AGG	0.690663	-0.467576
SCHW	BAC	0.782771	-0.464439
MET	AMP	0.824852	-0.440339
JPM	BAC	0.910547	-0.428117
ZION	USB	0.817203	-0.426048
...	...	...	...
DLX	C	0.617884	0.000000
DLTR	DIA	0.424936	0.000000
DLR	CCI	0.606968	0.000000
DY	DIA	0.523756	0.000000
AA	A	0.188259	0.000000

398 rows × 4 columns

### Rgresiones lineales para cada acción respecto a las demás

Se hizo un análisis en el que se implementó una regresión lineal para cada acción, en donde los retornos de dicha acción fueran la variable dependiente y los retornos de todas las demás acciones fueran las variables independientes. Con esto, se calculó el coeficiente de determinación  $R^2$  para cada regresión, con el fin de determinar si una acción podía ser explicada por todas las demás.

Para esto se separaron los datos en datos de entrenamiento y validación, donde el 80% de los datos se usó para ajustar la regresión. A continuación, se muestran los resultados, ordenados de

mayor a menor por el valor de \$R^2\$ en validación:

	R^2 entrenamiento	R^2 validación
SPY	0.992542	0.988988
IWN	0.962536	0.955054
IJH	0.985331	0.914920
VTI	0.993848	0.886890
EEM	0.932986	0.871587
...	...	...
TJX	0.523571	-119.848669
MA	0.430700	-150.119390
EWT	0.445596	-159.416823
RSX	0.743472	-583.895869
MBT	0.616940	-775.075931

399 rows × 2 columns

Con este análisis, se puede reforzar el resultado de que los índices tienden a ser variables que pueden estar explicadas por las demás variables, ya que un índice está compuesto por acciones individuales. Esto reitera la importancia de la decisión de sacar estos índices de los activos posibles para formar portafolios.

### Análisis de la matriz de covarianzas

Se calculó la matriz de covarianzas para los retornos de todas las acciones, usando el método habitual y el método con shrinkage de Ledoit & Wolf (2003). En los resultados se puede observar que el shrinkage de Ledoit & Wolf aumenta el determinante de la matriz de covarianzas, aunque esto no es necesario en este caso, ya que el determinante con el método habitual no es cercano a 0. Sin embargo, la matriz de covarianzas original sí está muy mal condicionada y el shrinkage de Ledoit & Wolf mejora mucho este número condición disminuyéndolo significativamente. Esto demuestra lo establecido por Bai & Shi (2011), de la matriz de covarianzas puede tender a ser muy volátil con los cambios, ya que, como se puede ver en este caso, está muy mal condicionada. A continuación, los resultados obtenidos:

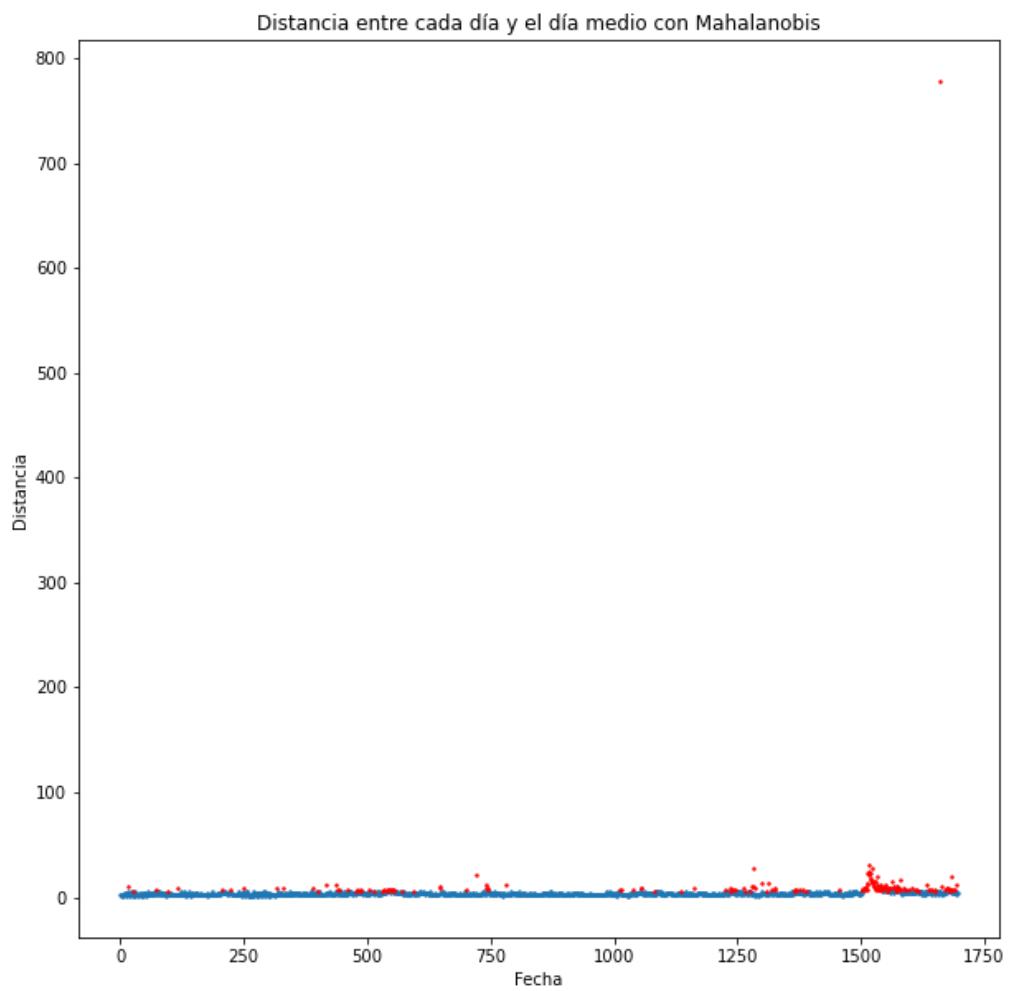
```
El determinante de la matriz de covarianzas habitual es 4.8236161316752686e+64
El determinante de la matriz de covarianzas con shrinkage de Ledoit & Wolf es inf
El número condición de la matriz de covarianzas habitual es 9092168.044344164
El número condición de la matriz de covarianzas con shrinkage de Ledoit & Wolf es 1.0572061506581245
```

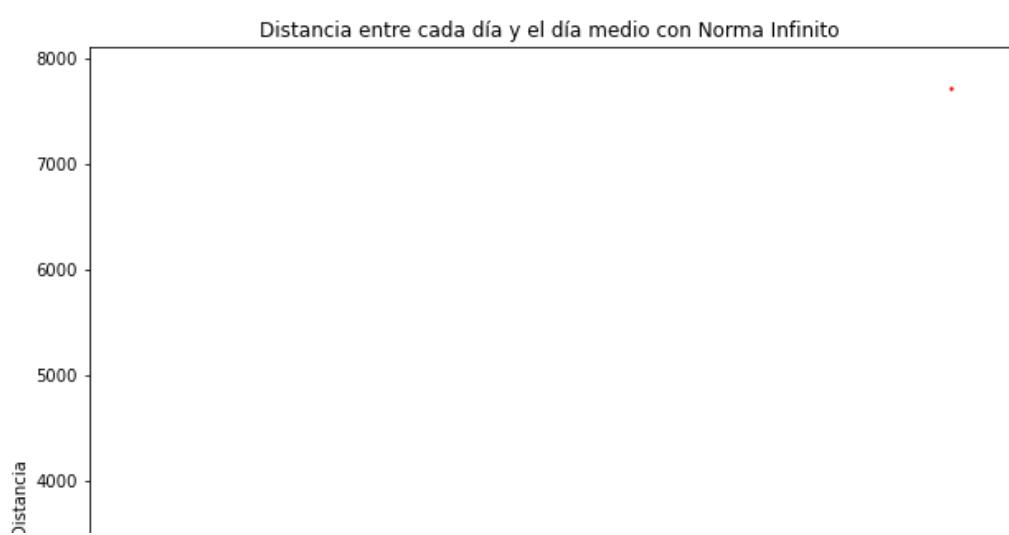
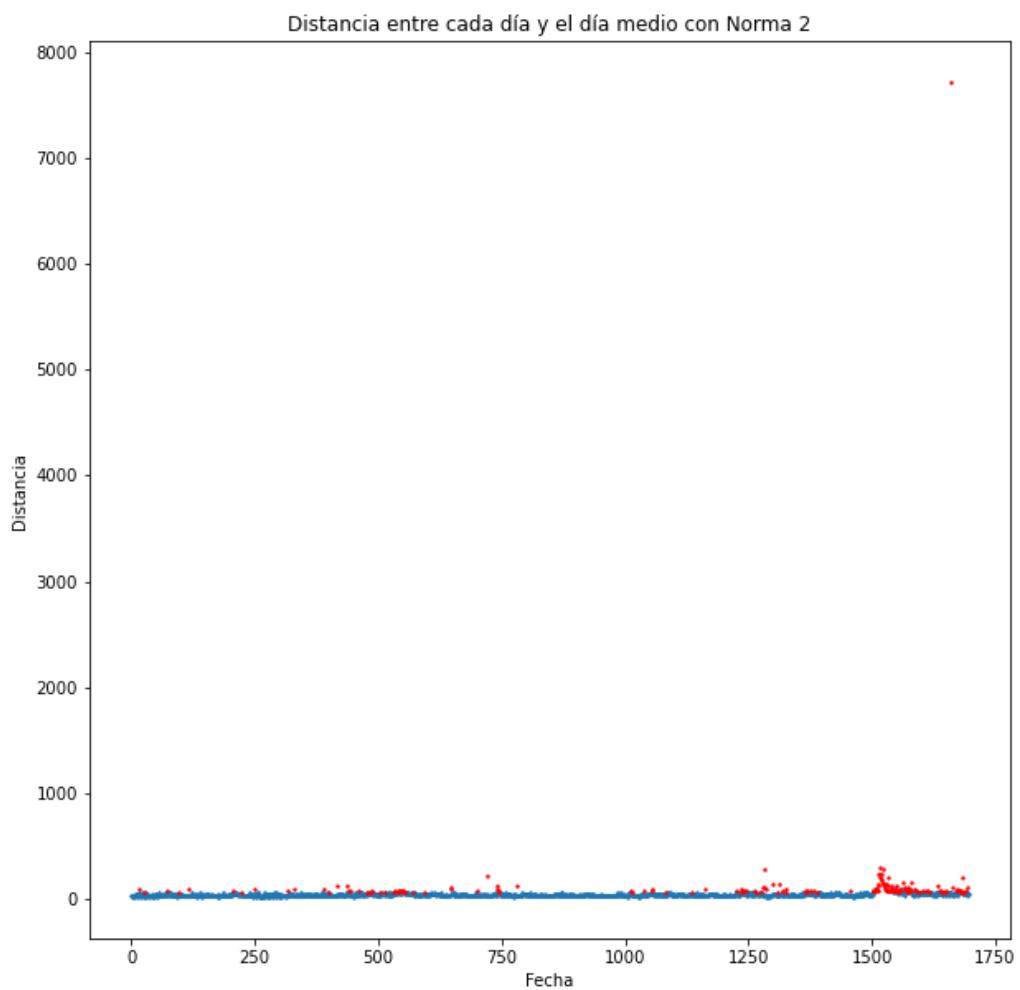
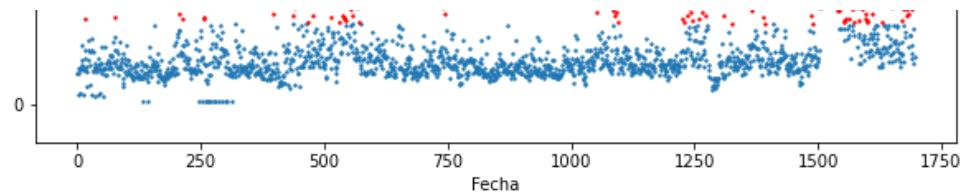
### Identificación de outliers

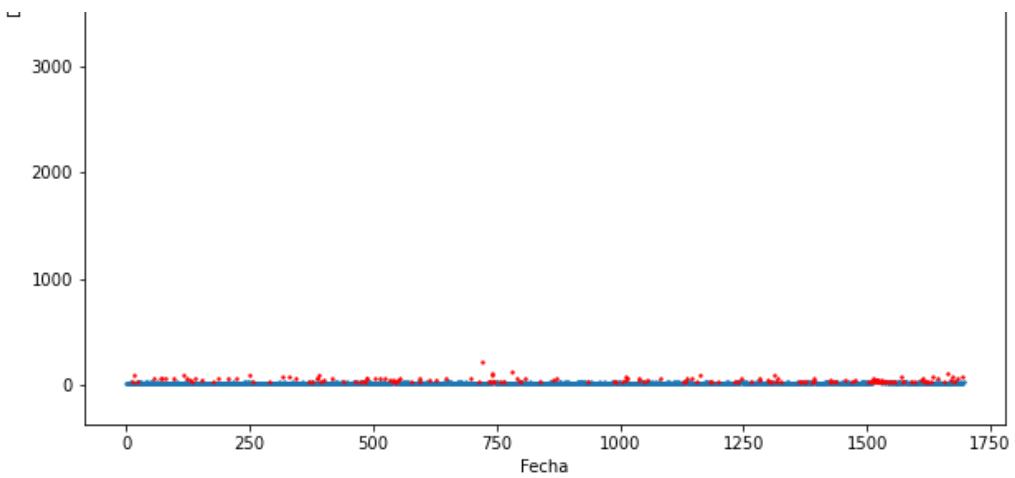
Se hizo un análisis de identificación de outliers usando diferentes métricas. Se buscó encontrar si había días atípicos en los datos. Para esto, se calculó el vector de medias y se calcularon las distancias de cada día de datos (cada registro) a este vector, usando la distancia de Mahalanobis y cada una de las distancias inducidas por la Norma-1, Norma-2 y Norma-\$\infty\$. Con este análisis, se encontró que el activo YNDX tenía un comportamiento muy atípico, independientemente de la métrica que se usara, por lo cual se procedió a eliminarlo, como se explicó en . A continuación, se muestran los resultados de este análisis de outliers:

**Con YNDX**





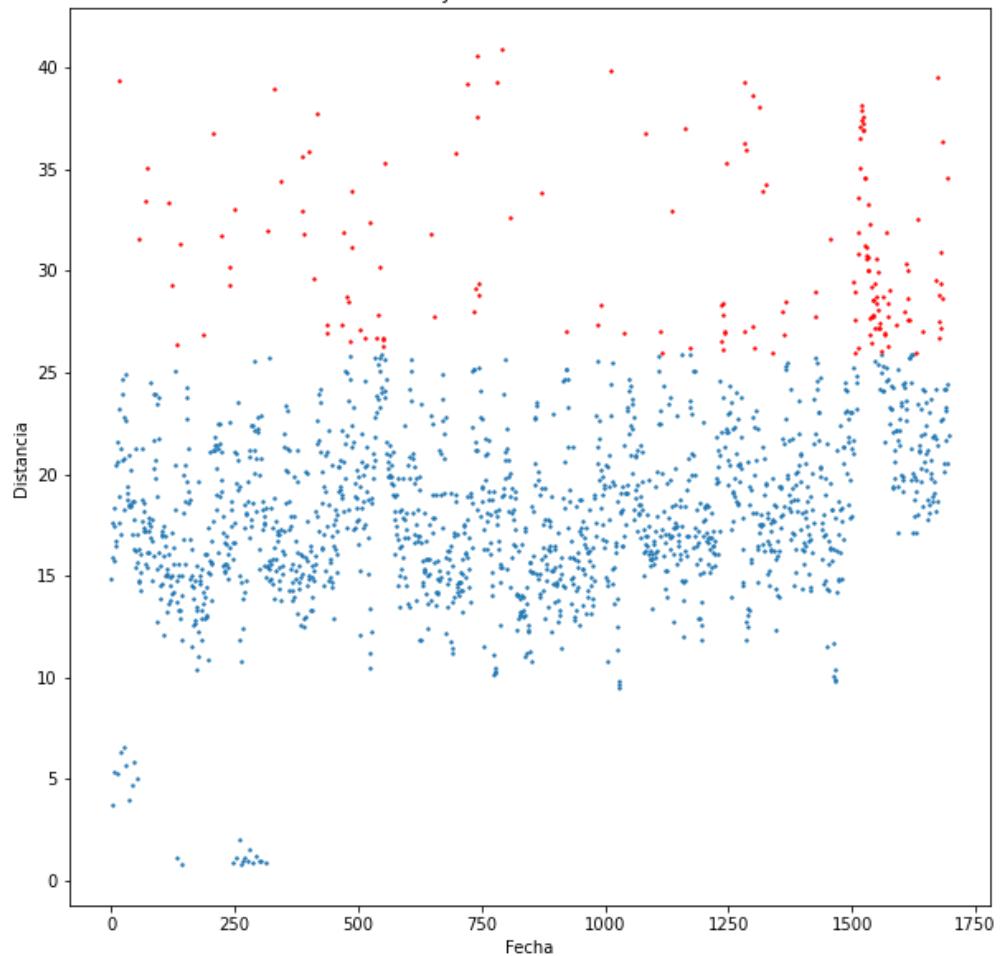




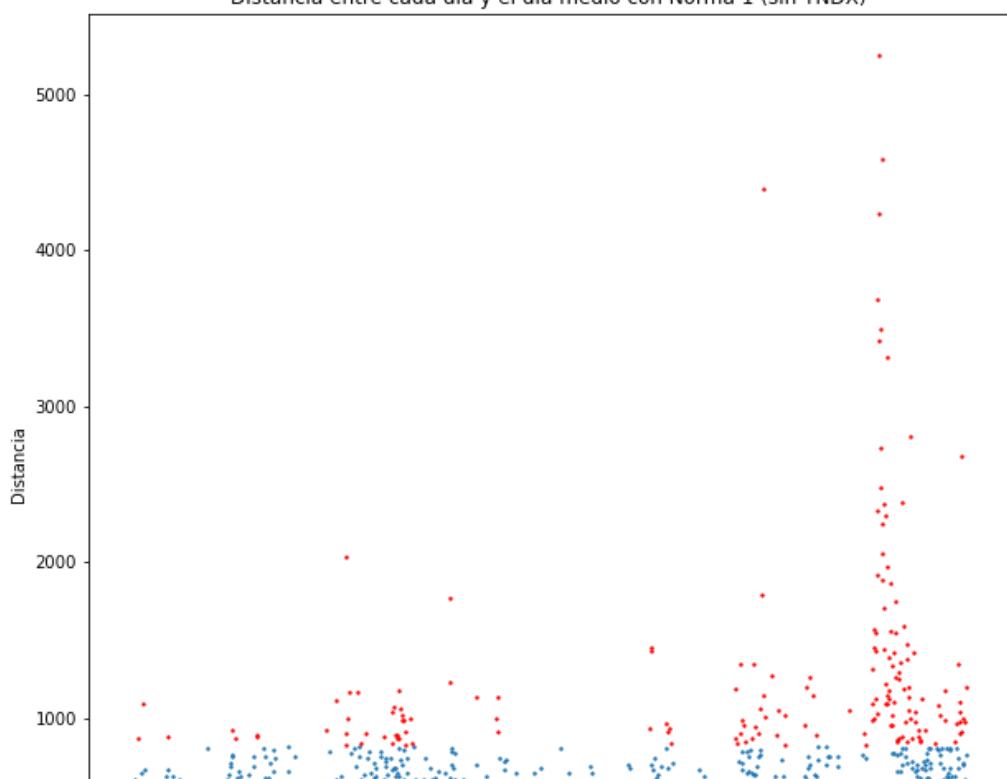
Sin YNDX

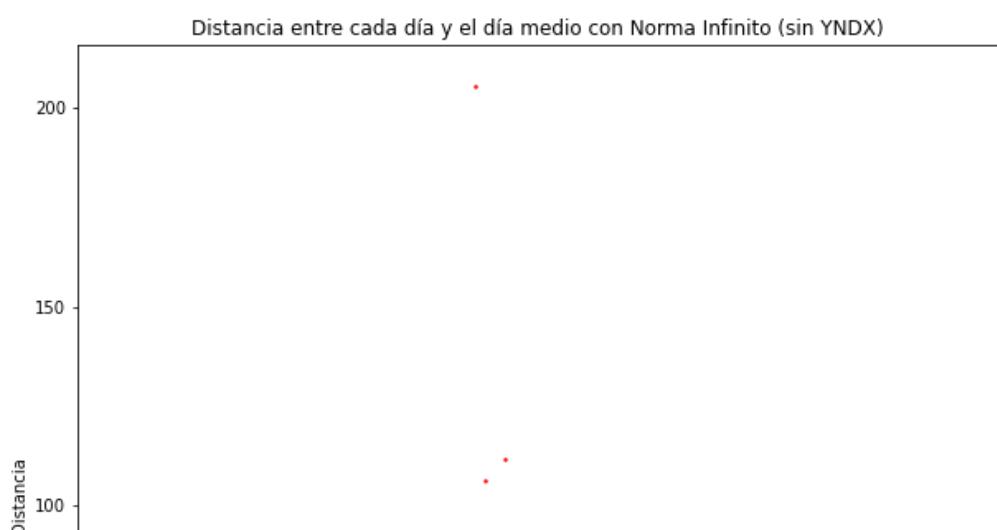
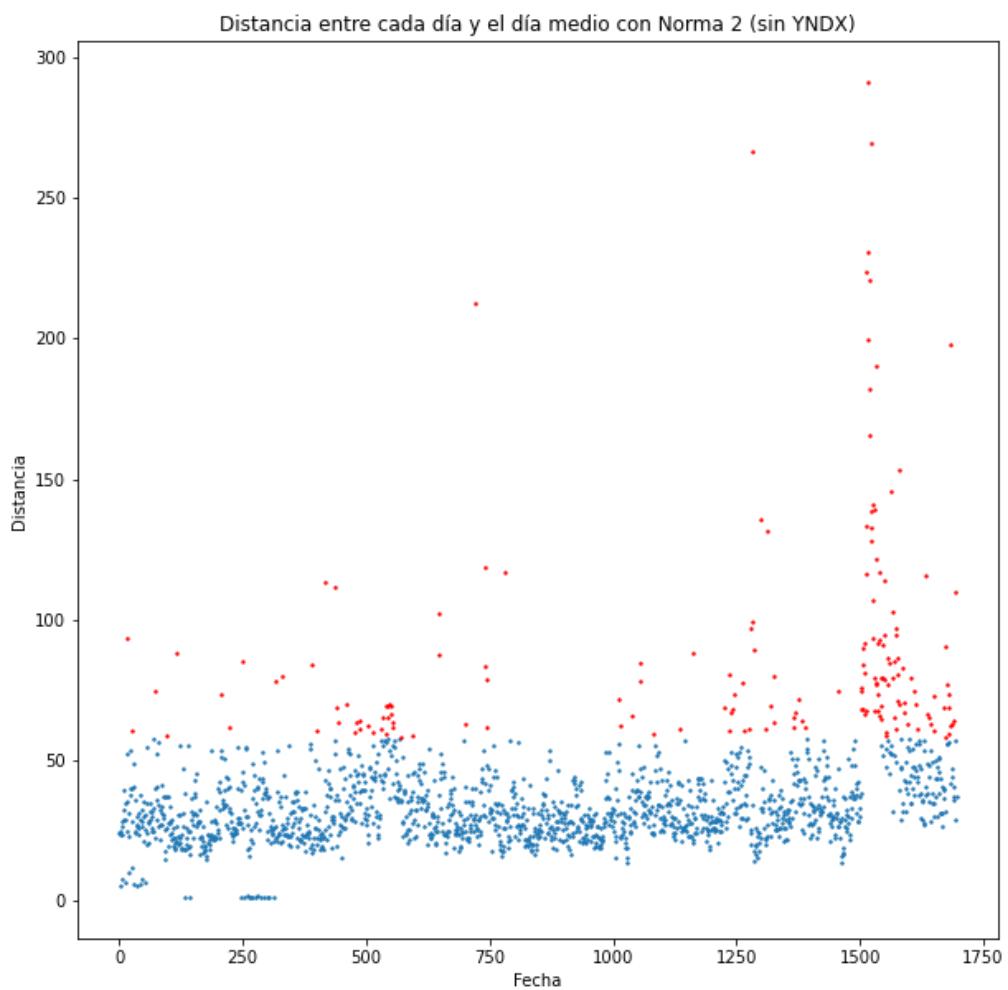
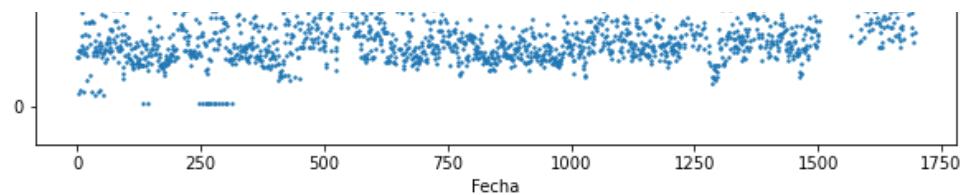


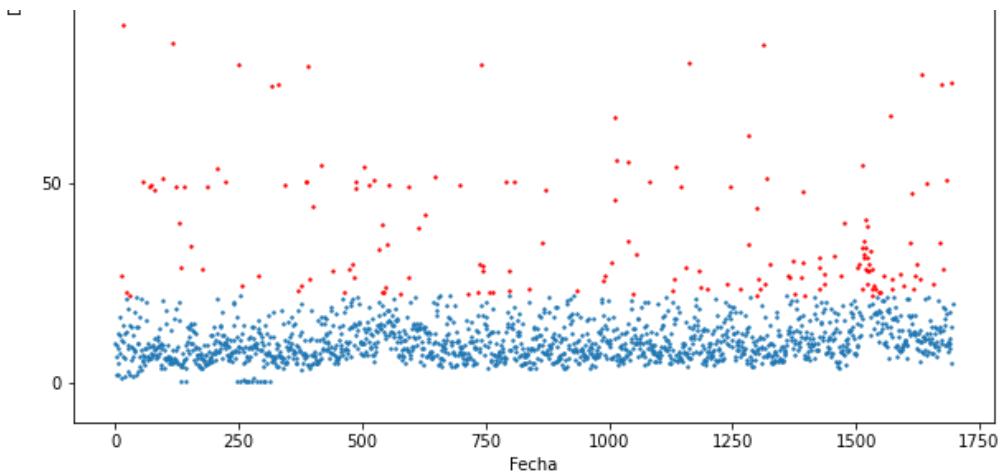
Distancia entre cada día y el día medio con Mahalanobis (sin YNDX)



Distancia entre cada día y el día medio con Norma 1 (sin YNDX)

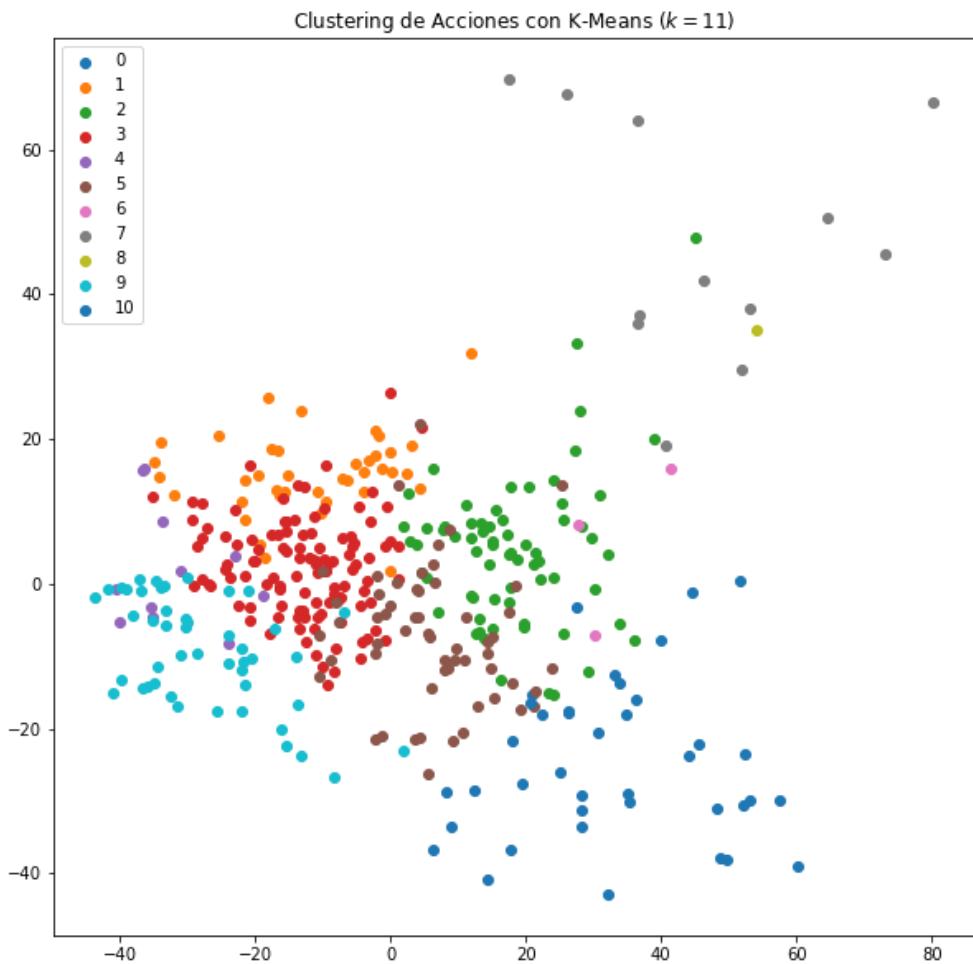






### Clustering no supervisado con K-Means

Se hizo un análisis de clusterización con K-Means, buscando posibles agrupamientos de los retornos de las acciones. Como se buscaba clasificar las acciones, se usó la matriz transpuesta de retornos, en donde cada acción es una observación y cada día es una variable. Se usó un valor de  $k=11$ , debido a que hay 11 sectores principales de la economía, entre los cuales se podrían dividir las acciones. Estos sectores son: Communication Services, Consumer Discretionary, Consumer Staples, Energy, Financials, Health Care, Industrials, Information Technology, Materials, Real Estate y Utilities (Fidelity, 2021). Como este clustering se hace para datos multidimensionales, no es posible visualizarlos. Por este motivo, se hizo un análisis de componentes principales (PCA), para encontrar los dos componentes principales de que mayor variabilidad explican en los datos y se graficaron los clusters con base en estos dos componentes. A continuación, se muestran los resultados de este clustering:

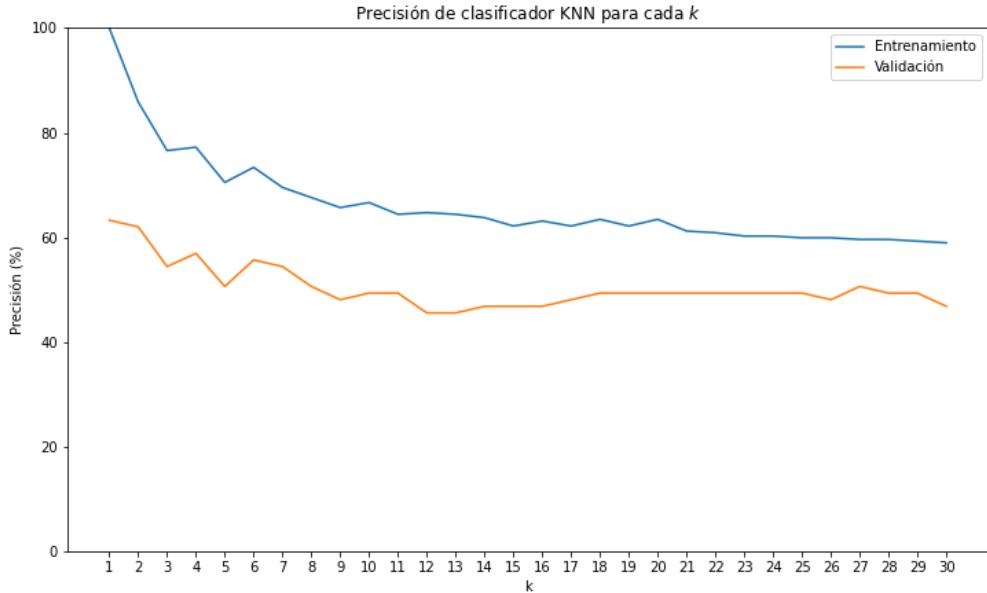


A pesar de que en algunos casos hay clusters claros, en otros no hay una división tan clara de las acciones y los clusters a los que pertenecen. Esto también puede pasar porque la clusterización fue hecha en una dimensión mucho más alta que la que se puede graficar.

Las etiquetas encontradas fueron agregadas a la matriz de retornos transpuesta y se guardó esta nueva matriz con etiquetas en la zona *refined* en S3.

### **Clasificación supervisada con K-Nearest Neighbors**

Con base en los clusters anteriormente encontrados, se hizo una clasificación usando K-Nearest Neighbors para diferentes valores de  $k$ , buscando determinar si se podía, con base en la matriz de retornos transpuesta etiquetada, clasificar correctamente la mayoría de las acciones. El modelo de KNN se entrenó con 80% de los datos y se validó con el 20% restante. Los resultados se muestran a continuación:



## Selección de modelos

### Modelos

Se seleccionaron tres modelos diferentes para ser comparados

- Un modelo de portafolios con pesos iguales para todos los activos del portafolio
- Un modelo de portafolios con los pesos de los activos ajustados para que el portafolio sea el de mínima varianza, usando la fórmula:

$$w_{MV} = \frac{\Sigma^{-1}\mathbf{1}}{\mathbf{1}^T \Sigma^{-1}\mathbf{1}}$$

donde,

$\Sigma^{-1}$  es la matriz inversa de la matriz de covarianzas de orden  $n \times n$

$\mathbf{1}$  es un vector de 1s de orden  $n \times 1$

$n$  es el número de activos del portafolio

- Un modelo de portafolios con los pesos de los activos ajustados para que el portafolio sea el de mínima varianza, usando la fórmula

$$w_{MV} = \frac{\Sigma^{-1}\mathbf{1}}{\mathbf{1}^T \Sigma^{-1}\mathbf{1}}$$

donde,

$\Sigma^{-1}$  es la matriz inversa generalizada de Moore-Penrose de la matriz de covarianzas estimada con shrinkage de Ledoit & Wolf de orden  $n \times n$

$\mathbf{1}$  es un vector de 1s de orden  $n \times 1$

$n$  es el número de activos del portafolio

Para todos los modelos, se crearon 50 portafolios con 300 activos seleccionados de manera aleatoria, entre los 391 activos totales restantes después de la limpieza descrita en . Cada uno de estos portafolios contiene los retornos históricos de los 300 activos que lo componen.

Cada portafolio se separó en un conjunto de entrenamiento y validación, con 80% de los datos para entrenamiento. En este caso, la separación no es aleatoria, porque se está tratando con series de tiempo, entonces los datos de entrenamiento van desde 2014-01-01 hasta 2019-07-17 y los datos de validación van desde 2019-07-18 hasta 2020-11-25.

Se calcularon las matrices de covarianzas para cada portafolio, tanto con el método habitual como con el shrinkage de Ledoit & Wolf, usando solamente los datos de entrenamiento, que son los únicos que se pueden usar para definir los pesos iniciales, para no incurrir en *data leakage* entrenando un modelo con información a la que en un caso real no tendría acceso, como, por ejemplo, datos de días posteriores.

La fórmula para calcular el retorno de un portafolio para cualquier día está dada por:

$$R_p = \sum_{i=1}^n w_i R_i$$

donde,

$R_p$  es el retorno del portafolio

$w_i$  es el peso del activo  $i$  en el portafolio

$R_i$  es el retorno del activo  $i$  para ese día en el portafolio

Vectorialmente, la fórmula también se puede expresar como

$$R_p = \vec{w} \cdot \vec{R}$$

La fórmula para calcular la varianza de un portafolio para cualquier día está dada por:

$$\sigma_p^2 = \vec{w}^T \Sigma \vec{w}$$

donde,

$\sigma_p^2$  es la varianza del portafolio

$\vec{w}$  es el vector de pesos de los activos del portafolio

$\Sigma$  es la matriz de covarianzas del portafolio

A continuación se muestran los resultados de los retornos, las desviaciones estándar y los retornos ajustados por riesgo ( $\frac{R}{\sigma}$ ) para cada modelo, tanto en entrenamiento como en validación. Los datos completos se pueden encontrar en [https://github.com/samorenog/proyecto\\_integrador\\_20212/blob/master/Notebooks/Risk\\_Adjusted\\_Comparisons\\_Generation.ipynb](https://github.com/samorenog/proyecto_integrador_20212/blob/master/Notebooks/Risk_Adjusted_Comparisons_Generation.ipynb)

## Portafolio de pesos iguales

### Entrenamiento









	Minima varianza - Pesos iguales	Minima varianza con LW - Pesos iguales	Minima varianza con LW - Minima Varianza
portfolio_0	-0.351985	0.063533	0.415499
portfolio_1	-0.212818	0.066991	0.279809
portfolio_2	0.085010	0.280012	0.195002
portfolio_3	-0.283216	0.105777	0.388992
portfolio_4	-0.408700	0.164297	0.572997
portfolio_5	0.109010	0.155741	0.046730
portfolio_6	-0.128097	0.242418	0.369515
portfolio_7	-0.392993	0.056926	0.449919
portfolio_8	-0.389185	0.115658	0.504843
portfolio_9	-0.374492	0.149063	0.523555
portfolio_10	-0.299757	0.150528	0.450285
portfolio_11	-0.057667	0.136578	0.194245
portfolio_12	-0.121764	0.346286	0.468050
portfolio_13	0.066533	0.214355	0.147822
portfolio_14	-0.328009	0.097920	0.429930
portfolio_15	-0.162626	0.113713	0.276340
portfolio_16	-0.439468	0.197398	0.636886
portfolio_17	-0.145354	0.165763	0.311117
portfolio_18	-0.418505	0.024432	0.442937
portfolio_19	-0.212064	0.225374	0.437437
portfolio_20	-0.448793	0.049937	0.498730
portfolio_21	-0.295995	0.230657	0.528663
portfolio_22	0.211639	0.218638	0.007000
portfolio_23	-0.034456	0.160344	0.194800
portfolio_24	-0.004023	-0.067305	-0.063282
portfolio_25	-0.095152	0.284572	0.379725
portfolio_26	-0.247861	0.174845	0.422706
portfolio_27	-0.335462	0.184501	0.519983
portfolio_28	-0.438679	0.094381	0.533060
portfolio_29	0.178002	0.2223261	0.047290
portfolio_30	-0.047970	0.253739	0.301709
portfolio_31	-0.254695	0.104153	0.358848
portfolio_32	-0.293236	0.083271	0.376507
portfolio_33	-0.410803	0.060664	0.471458
portfolio_34	0.087300	0.193624	0.106325
portfolio_35	0.313401	0.512952	0.199551
portfolio_36	-0.148000	0.211494	0.359494
portfolio_37	-0.242908	0.221521	0.464429
portfolio_38	-0.138546	0.349380	0.487926
portfolio_39	-0.128135	0.266032	0.394168
portfolio_40	-0.118817	0.108356	0.227173
portfolio_41	-0.526369	-0.010338	0.516031
portfolio_42	-0.167887	0.259879	0.427765
portfolio_43	-0.194041	0.149602	0.343643
portfolio_44	-0.410286	0.079266	0.489551
portfolio_45	-0.029359	0.281501	0.310860
portfolio_46	-0.305775	-0.041061	0.264714
portfolio_47	-0.401303	-0.035371	0.365931
portfolio_48	-0.268452	0.135673	0.404124
portfolio_49	-0.038359	0.090724	0.129083
Promedio	-0.194023	0.158053	0.352076

Como se puede ver en los datos de entrenamiento, en promedio, los portafolios con el modelo de mínima varianza tienen mejores rendimientos ajustados por riesgo que los portafolios con el modelo de pesos iguales. Los portafolios con el modelo de mínima varianza usando el shrinkage de Ledoit & Wolf para la matriz de covarianzas y la inversa generalizada de Moore-Penrose tienen, en promedio, mejor retorno ajustado por riesgo que los otros dos modelos.

## Validación

	Minima varianza - Pesos Iguales	Minima varianza con LVV - Pesos Iguales	Minima varianza con LVV - Minima Varianza
portfolio_0	-22.180264	3.914041	26.094305
portfolio_1	13.383966	7.083953	-6.300013
portfolio_2	14.466868	3.791890	-10.674977
portfolio_3	10.121478	5.719740	-4.401738
portfolio_4	-15.058641	1.713057	16.771697
portfolio_5	10.367741	8.185807	-2.181934
portfolio_6	8.437757	3.376634	-5.061123
portfolio_7	56.044448	8.058750	-47.985698
portfolio_8	27.056353	5.016012	-22.040341
portfolio_9	2.274240	3.528681	1.254441
portfolio_10	10.843325	6.447725	-4.395601
portfolio_11	-30.824918	-3.835044	26.989874
portfolio_12	1.465872	4.631572	3.165700
portfolio_13	9.084389	5.923204	-3.161185
portfolio_14	-9.599311	-3.407296	6.192016
portfolio_15	55.257308	15.914869	-39.342439
portfolio_16	-6.742484	2.612947	9.355431
portfolio_17	-4.083183	3.794687	7.877871
portfolio_18	15.448478	8.479885	-6.968593
portfolio_19	18.252871	-0.281500	-18.534371
portfolio_20	18.301019	3.162522	-15.138498
portfolio_21	7.685604	3.304902	-4.381702
portfolio_22	0.458744	1.654069	1.195324
portfolio_23	18.411073	13.155468	-5.255604
portfolio_24	-1.445538	-1.308643	0.136896
portfolio_25	9.360567	5.465878	-3.894689
portfolio_26	5.233744	4.318342	-0.915402
portfolio_27	-40.154691	1.475677	41.630368
portfolio_28	11.929828	8.071750	-3.858079
portfolio_29	12.801273	6.659044	-6.142229
portfolio_30	24.675293	6.085268	-18.590025
portfolio_31	15.415883	0.622950	-14.792933
portfolio_32	33.348522	4.007443	-29.341079
portfolio_33	29.956411	6.715409	-23.241002
portfolio_34	13.193060	2.409646	-10.783414
portfolio_35	4.780526	1.632218	-3.148309
portfolio_36	-24.531075	1.951207	26.482282
portfolio_37	-6.449258	9.417151	15.866409
portfolio_38	-12.129591	3.571619	15.701210
portfolio_39	32.428550	3.919382	-28.509169
portfolio_40	11.629069	8.252037	-3.377032
portfolio_41	-16.588930	5.715730	22.304660
portfolio_42	-17.857605	-3.453112	14.404494
portfolio_43	8.023186	3.930487	-4.092698
portfolio_44	-12.580281	-3.733478	8.846802
portfolio_45	2.202139	5.381000	3.178861
portfolio_46	-6.551411	-0.055657	6.495755
portfolio_47	16.023580	2.142306	-13.881274
portfolio_48	30.426578	8.058070	-22.368508
portfolio_49	22.295876	6.912773	-15.383103
Promedio	7.086189	4.202221	-2.883967

## Análisis y conclusiones

- Como se puede ver en los datos de entrenamiento, en promedio, los portafolios con el modelo de mínima varianza tienen mejores rendimientos ajustados por riesgo que los portafolios con el modelo de pesos iguales. Los portafolios con el modelo de mínima varianza usando el shrinkage de Ledoit & Wolf para la matriz de covarianzas y la inversa generalizada de Moore-Penrose tienen, en promedio, mejor retorno ajustado por riesgo que los otros dos modelos.
- Como se puede ver en la tabla de comparaciones, en el período de validación, el mejor modelo es el de mínima varianza usando los métodos tradicionales, en promedio. Sin embargo las diferencias son mucho más volátiles. Esto puede deberse a que la matriz de covarianzas está muy desactualizada, ya que se está usando la misma que para el entrenamiento. Lo ideal sería calcular la matriz de covarianzas de manera móvil, para que cada vez se tenga la versión más actualizada. Debido a este problema, es probable que estos resultados de validación no sean muy significativos, aunque está claro que, tanto en entrenamiento como en validación, el retorno ajustado por riesgo es superior en los modelos de mínima varianza, respecto al modelo de pesos iguales.

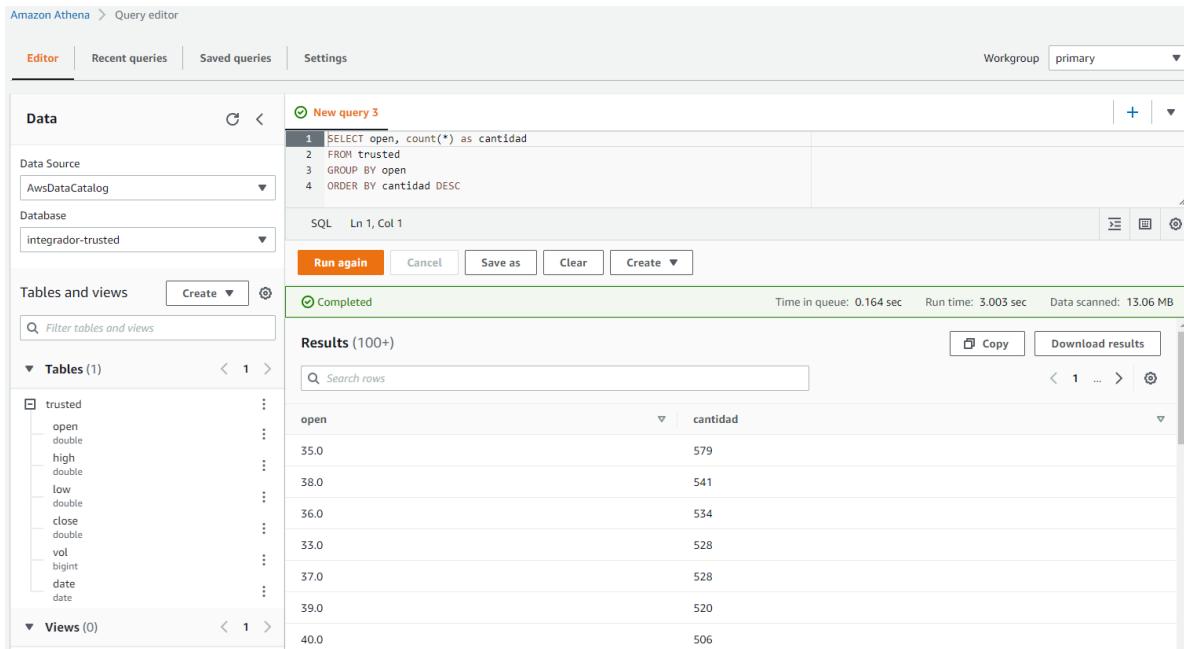
## Tecnología

---

Ingeniería de datos y uso de tecnología:

1. La fuente de nuestros datos fue la plataforma **Kaggle**. Una de las comunidades de científicos de datos más grandes del mundo, que permite publicar conjuntos de datos. Allí nos encontramos con un dataset grande de acciones ("StockMinuteData"), que contiene precios históricos minuto a minuto de 399 acciones. Este conjunto de datos, está en formato ".parquet", pesa 13 GB y contiene 4 features de las acciones las cuales son: Precio de apertura, Precio más alto durante de ese minuto, precio más bajo, y por último el precio de cierre en ese minuto.
2. Como podemos darnos cuenta, los datos que estamos utilizando son precios de acciones que están dados únicamente por valores cuantitativos, por lo que vale la pena aclarar que la naturaleza de nuestros datos es **estructurada** y los podemos representar en matrices, al igual que podemos aplicar cualquier operación de ordenamiento o de tipo matricial.
3. Para nuestro caso, no estamos buscando procesar datos inmediatamente, sino que queremos re balancear portafolios basados en datos históricos, por lo que la ingestión de nuestros datos es del tipo **batch**, esto significa que primero almacenamos los datos y luego los pasamos por unos niveles de procesamiento. Para el almacenamiento de los datos decidimos implementar un **Datalake** con ayuda de la plataforma en la nube **AWS**. Más específicamente, con **AWS S3** que nos permite el almacenamiento de nuestros datos como objetos. En nuestro Datalake tenemos 3 zonas para organización para los datos: Raw, Trusted y Refined. En la zona Raw tenemos los datos tal cual los recuperamos de la plataforma, en la zona Trusted almacenamos los datos que ya hemos pasado por un proceso **ETL** que consiste en transformar los datos de cada minuto para que se encuentren expresados por días, esto nos va permitir reducir el gran tamaño de los datos y posteriormente hacer un mejor procesamiento. En la última zona, almacenamos los datos que hemos pasado por otro proceso ETL, donde hacemos una limpieza de los datos y nos quedamos con los datos de cierre de cada acción, de manera que tenemos un dataset donde cada registro es un día diferente y cada feature corresponde a una acción diferente. Tenemos otro dataset con la misma organización pero que no contiene los precios de cierre sino los retornos (precio de cierre de un día menos el precio de cierre del día anterior). Allí almacenamos también las matrices de covarianzas, las matrices de retornos ajustados según diversas medidas y los 50 portafolios que creamos.

4. Para el procesamiento de datos y los procesos ETL utilizamos en una gran medida las estructuras Dataframes que nos permiten un muy buen manejo de los datos estructurados. Además, nos apoyamos de herramientas como numpy para operaciones básicas y para algunas operaciones más avanzadas utilizamos sklearn que es una biblioteca de aprendizaje automático y nos ayuda a hacer regresiones lineales, calcular métricas y muchas funciones más. Scipy es otra biblioteca que cabe señalar ya que tiene una gran variedad de herramientas y algoritmos matemáticos.
5. Como parte de la visualización de los datos nos apoyamos en el ambiente de Matplotlib que nos permite generar gráficos a partir de listas o arrays, que son las estructuras de datos que más utilizamos para representar nuestros datos, y nos da varias facilidades para configurar lo que queremos ver y cómo lo queremos ver.
6. Utilizamos **SQL** como motor de consulta para hacer una exploración de nuestros datos trusted. Algunas de las consultas que hicimos, fueron las siguientes:



The screenshot shows the Amazon Athena Query Editor interface. The top navigation bar includes 'Amazon Athena > Query editor' and tabs for 'Editor', 'Recent queries', 'Saved queries', and 'Settings'. A 'Workgroup' dropdown is set to 'primary'. The main area has three sections: 'Data' (with 'Data Source' set to 'AwsDataCatalog' and 'Database' set to 'integrador-trusted'), 'Tables and views' (listing 'Tables (1)' with 'trusted' selected, showing columns 'open', 'double', 'high', 'low', 'close', 'vol', 'bigint', 'date', and 'date'), and 'Views (0)'. The central panel displays a 'New query 3' with the following SQL code:

```

1 SELECT open, count(*) as cantidad
2 FROM trusted
3 GROUP BY open
4 ORDER BY cantidad DESC
    
```

The status bar indicates the query was completed with a run time of 3.003 sec and 13.06 MB scanned data. Below this, the 'Results (100+)' section shows a table with two columns: 'open' and 'cantidad', containing the following data:

open	cantidad
35.0	579
38.0	541
36.0	534
33.0	528
37.0	528
39.0	520
40.0	506

Esta consulta nos devuelve los valores de apertura más comunes en el mercado. Siendo los primeros, los que más veces se repiten en las acciones y los últimos, los valores de apertura más extraños ya que suelen aparecer pocas veces.

Amazon Athena > Query editor

Editor | Recent queries | Saved queries | Settings Workgroup primary

Data < New query 3

Data Source: AwsDataCatalog Database: integrador-trusted

Tables and views < 1 >

Tables (1) < 1 >

trusted

- open
- double
- high
- double
- low
- double
- close
- double
- vol
- bigrnt
- date
- date

Views (0) < 1 >

New query 3

```
1 SELECT *
2 FROM trusted
3 WHERE high-low >= 500
4 ORDER BY high - low DESC
```

SQL Ln 1, Col 1

Run again Cancel Save as Clear Create

Completed Time in queue: 0.177 sec Run time: 1.75 sec Data scanned: 25.57 MB

Results (9)

open	high	low	close	vol	date
92.69	100000.0	92.48	92.79	40002	2010-05-06
1.82E7	1.82E7	1.818E7	1.819E7	520231	2011-01-05
1.814E7	1.815E7	1.813E7	1.81449E7	43596	2011-01-07
1.755E7	1.756E7	1.755E7	1.7556E7	24396	2010-12-31
1.751E7	1.752E7	1.751E7	1.751E7	2999241	2011-01-01
1.7685E7	1.769E7	1.768E7	1.768E7	1800	2011-01-03
1 R10SF7	1 R11F7	1 R1F7	1 R11F7	4R9200	2011-01-06

Copy Download results

Acá podemos ver los días en que alguna acción tuvo mayor volatilidad. Esto lo calculamos con el precio más alto que tuvo y el menor que tuvo. Por lo que nos mostrará entonces, los cambios más bruscos que hubo en el precio de una acción.

Amazon Athena > Query editor

Editor | Recent queries | Saved queries | Settings Workgroup primary

Data < New query 3

Data Source: AwsDataCatalog Database: integrador-trusted

Tables and views < 1 >

Tables (1) < 1 >

trusted

- open
- double
- high
- double
- low
- double
- close
- double
- vol
- bigrnt
- date
- date

New query 3

```
1 SELECT *
2 FROM trusted
3 WHERE open = close
4
```

SQL Ln 4, Col 1

Run again Cancel Save as Clear Create

Completed Time in queue: 0.152 sec Run time: 2.772 sec Data scanned: 77.74 MB

Results (100+)

open	high	low	close	vol	date
16.06	16.06	16.05	16.06	7376	2011-08-25
16.52	16.52	16.52	16.52	33378	2011-08-26
16.74	16.74	16.74	16.74	12646	2011-08-30
16.29	16.29	16.29	16.29	26561	2011-08-31
15.9	15.91	15.88	15.9	4200	2011-09-06
17.48	17.48	17.48	17.48	80316	2011-09-07

Copy Download results

En esta consulta podemos observar los días en que el valor de la acción no se movió, puesto que el mismo precio con el que empezó el día fue con el que terminó.

## Desarrollo del proyecto

### Despliegue del proyecto

En una implementación real, el proyecto partiría de una herramienta que esté constantemente leyendo datos reales y, a pesar de que en este proyecto se use procesamiento en batch, lo ideal sería una aplicación con un modelo de streaming usando AWS Kinesis, por ejemplo. Además, la fuente de datos también tendría que ser dinámica para que esté constantemente enviando variaciones en los datos, a diferencia de los datos usados aquí, que son estáticos. Para esto se podría usar una plataforma como Bloomberg (es costosa). Para el procesamiento de datos se podría usar un cluster de EMR, usando Spark para procesar en paralelo los datos que llegan de diferentes acciones, para darles el formato adecuado e ir almacenándolos en un sistema

distribuido. Con los archivos distribuidos, se podrían ir generando tablas (estos datos son estructurados) en aplicaciones que luego se puedan consultar con SQL, por medio de Hive o Athena, por ejemplo. Para el análisis exploratorio podría usarse una aplicación como Zeppelin, que pueda usar los datos directamente del cluster de EMR. Finalmente, el acceso a los datos y la visualización por parte del usuario final podría ser por medio de una interfaz web, llamando APIs que entreguen los datos.

## Otros aspectos tecnológicos

### Fuentes de datos

La fuente de datos usada fue <https://www.kaggle.com/martholi/stockminutedata>, que contiene los datos de 1474 activos en archivos en formato .parquet.

### Ingesta de datos

La ingestá de datos fue tomando los datos de Kaggle, descargándolos manualmente y llevándolos a S3. Una vez en S3, los datos fueron procesados por medio de un crawler de Glue, para llevarlos a tablas y, posteriormente, consultarlos con Athena.

### Almacenamiento

Todo el almacenamiento fue usando un datalake en S3. Se usaron tres zonas: *raw*, *trusted* y *refined*. En la zona *raw* se almacenaron los datos tal cual como venían de la fuente. Luego se crearon algunos procesos ETL para pasar estos datos de minutos a días, que es el formato deseado, y se cargaron en la zona *trusted*. Posteriormente, se crearon otros procesos ETL, para transformar los datos y limpiarlos, creando las matrices de precios y retornos, que luego serían usadas para el análisis, procesamiento y generación de los modelos. Estas matrices y los demás datos ya generados y procesados completamente fueron cargados en la zona *refined* en el bucket de S3. La dirección del bucket de S3 es *s3://proyecto-integrador-20212-pregrado*.

## Ambiente de procesamiento

- El lenguaje de programación usado fue Python.
- El ambiente de procesamiento usado en general fue Jupyter.
- Se usaron librerías como:
  - *pandas*: para la gestión de DataFrames, la lectura de archivos .parquet desde S3 y la generación de archivos .parquet que se cargaban a S3.
  - *numpy*: se usó para diferentes cálculos algebráicos, como los cálculos de determinantes, matrices inversas, matrices pseudo-inversas, matrices de covarianzas habituales, normas, entre otros.
  - *scikit-learn*: para modelos de clasificación, regresión, clusterización y para el pre-procesamiento de datos para estos modelos, además del módulo de *covariance* que se usó para estimar la matriz con shrinkage de Ledoit & Wolf.
  - *matplotlib* se usó para todas las gráficas y toda la parte de visualización del proyecto.
  - *scipy* se usó para obtener la distancia de Mahalanobis entre cada vector y el vector de medias y para algunos análisis de estadística descriptiva de los datos.

## Aplicaciones

Toda la visualización se exportó a imágenes por medio de *matplotlib* y todos los resultados y matrices que se generaron se guardaron en el formato .parquet en la zona *refined* en el datalake creado en S3 *s3://proyecto-integrador-20212-pregrado*. También se hicieron algunas consultas por medio de Athena, con base en datos de tablas generadas por un crawler de Glue.

## Conclusiones generales del proyecto

---

### Referencias

---

- Bai, J., & Shi, S. (2011). Estimating High Dimensional Covariance Matrices and its Applications. ANNALS OF ECONOMICS AND FINANCE, 12(2), 199–215.
- Breaking Down Finance. (n.d.). Minimum Variance Portfolio - Breaking Down Finance. Retrieved November 25, 2021, from <https://breakingdownfinance.com/finance-topics/modern-portfolio-theory/minimum-variance-portfolio/>
- Demiguel, V., Garlappi, L., Nogales, F. J., & Uppal, R. (2009). A Generalized Approach to Portfolio Optimization: Improving Performance by Constraining Portfolio Norms. MANAGEMENT SCIENCE, 55(5), 798–812. <https://doi.org/10.1287/mnsc.1080.0986>
- Fernholz, E. R. (2002). Stochastic Portfolio Theory. Stochastic Portfolio Theory, 1–24. [https://doi.org/10.1007/978-1-4757-3699-1\\_1](https://doi.org/10.1007/978-1-4757-3699-1_1)
- Fidelity. (2021). Sectors & Industries Overview - U.S. Sectors- Fidelity. [https://eresearch.fidelity.com/eresearch/markets\\_sectors/sectors/sectors\\_in\\_market.jhtml](https://eresearch.fidelity.com/eresearch/markets_sectors/sectors/sectors_in_market.jhtml)
- Investopedia. (2021). Exchange Traded Fund (ETF) Definition and Overview. <https://www.investopedia.com/terms/e/etf.asp>
- Ledoit, O., & Wolf, M. (2003). Honey, I Shrunk the Sample Covariance Matrix.
- Levina, E., Rothman, A., & Zhu, J. I. (2008). SPARSE ESTIMATION OF LARGE COVARIANCE MATRICES VIA A NESTED LASSO PENALTY. The Annals of Applied Statistics, 2(1), 245–263. <https://doi.org/10.1214/07-AOAS139>
- Markowitz, H. (1952). Portfolio Selection. The Journal of Finance, 7(1), 77–91.
- Penrose Communicated, R., & Todd, J. A. (1955). A generalized inverse for matrices. Mathematical Proceedings of the Cambridge Philosophical Society, 51(3), 406–413. <https://doi.org/10.1017/S0305004100030401>
- Petters, A. O., & Dong, X. (2016). An Introduction to Mathematical Finance with Applications. <http://doi.org/10.1007/978-1-4939-3783-7>
- Washington University. (2013). Chapter 1 Portfolio Theory with Matrix Algebra.
- Wordnik. (n.d.). arbitrage - definition and meaning. Retrieved November 25, 2021, from <https://www.wordnik.com/words/arbitrage>