

UNIVERSIDAD SAN FRANCISCO DE QUITO

FUNDAMENTOS DE CIENCIA DE DATOS

PROYECTO 1

TEMA: IMPACTO DE INSIDER TRADING FRENTE A LAS VARIACIONES DEL PRECIO DE LA ACCIÓN DE APPLE QUE COTIZAN EN LA BOLSA DE NUEVA YORK DESDE EL AÑO 1 DE ENERO DEL 2015 HASTA EL 27 DE MARZO 2025.

ALUMNO: SANTIAGO RODRIGUEZ

MARZO 2025

Contenido

| | |
|---|----------|
| REPORTE DE EDA | 4 |
| 1. DATOS EXTRAÍDOS DE YAHOO FINANCE | 4 |
| 1.1 Librerías | 4 |
| 1.2 Valores vacíos | 4 |
| 1.3 Estadísticas básicas | 6 |
| 1.4 Gráficos de datos extraídos de Yahoo Finance..... | 6 |
| 2. DATOS EXTRAÍDOS DE OPENINSIDER.COM | 8 |
| 2.1 Consulta de nombres de directivos y validar que no estén registrados más de una vez o que tengan errores..... | 10 |
| 2.2 Validación y ajuste de cargos duplicados | 10 |
| 2.3 Revisar cantidades mínima y máximas de cantidad de acciones negociadas | 12 |
| 2.4 Gráficos de datos extraídos de OpenInsider.com | 14 |

INDICE DE FIGURAS Y GRÁFICOS

| | |
|--|----|
| ILUSTRACIÓN 1: IMPRESIÓN INICIAL DE TABLA DE DATOS DE YAHOO FINANCE | 4 |
| ILUSTRACIÓN 2: INFORMACIÓN DE TABLA DE YAHOO FINANCE | 5 |
| ILUSTRACIÓN 3: CAMBIO DE TIPO DE DATOS Y REDONDEO A 2 DECIMALES..... | 5 |
| ILUSTRACIÓN 4: VERIFICACIÓN DE VALORES DUPLICADOS | 5 |
| ILUSTRACIÓN 5: ESTADÍSTICAS BÁSICAS DE LA TABLA YAHOO FINANCE | 6 |
| GRÁFICO 1: EVOLUCIÓN DE OPEN Y CLOSE..... | 6 |
| GRÁFICO 2: EVOLUCIÓN DE HIGH Y LOW | 7 |
| GRÁFICO 3: VERIFICAR OUTLIERS..... | 7 |
| ILUSTRACIÓN 6: VISTA INICIAL DE DATOS EXTRAÍDOS DE OPENINSIDER.COM..... | 8 |
| ILUSTRACIÓN 7: INFORMACIÓN DE TABLA DE OPENINSIDER.COM | 8 |
| ILUSTRACIÓN 8: VERIFICACIÓN Y CONTEO DE VALORES NAN DE OPENINSIDERS.COM | 9 |
| ILUSTRACIÓN 9: COMPROBACIÓN DE CAMBIO DE DATOS DE OPENINSIDERS.COM | 9 |
| ILUSTRACIÓN 10: VERIFICACIÓN DE NOMBRES DE DIRECTIVOS DE AAPL..... | 10 |
| ILUSTRACIÓN 11: VALIDACIÓN DE CARGOS DUPLICADOS..... | 10 |
| ILUSTRACIÓN 12: EVIDENCIA DE MÚLTIPLES CARGOS ASOCIADOS A UNA PERSONA | 11 |
| ILUSTRACIÓN 13: TIPOS DE TRANSACCIÓN | 11 |
| ILUSTRACIÓN 14: REVISAR MÍNIMOS Y MÁXIMOS DE LA COLUMNA 'QUANTITY OF SHARES' | 12 |
| ILUSTRACIÓN 15: LISTA DE CANTIDAD DE TRANSACCIONES REALIZADAS ORDENADAS DE MAYOR A MENOR | 13 |
| GRÁFICO 4: OUTLIERS EN OPENINSIDERS | 14 |
| GRÁFICO 5: SCATTERPLOT DE OPENINSIDERS | 14 |
| GRÁFICO 6: CONTEO DE TIPO DE OPERACIONES | 15 |

| | |
|---|----|
| GRÁFICO 7: RANKING DE DIRECTIVOS 1..... | 15 |
| GRÁFICO 8: EVOLUCIÓN DE ACCIONES NEGOCIADAS | 16 |

REPORTE DE EDA

A continuación, se presenta la documentación de los hallazgos encontrados con el EDA aplicado a los bases de datos.

1. DATOS EXTRAÍDOS DE YAHOO FINANCE

1.1 Librerías

Se utilizó las librerías:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Ilustración 1: librerías utilizadas en EDA

1.2 Valores vacíos

- Luego de leer el archivo AAPL precios_acciones.csv que es la información extraída con el API de Yahoo Finance se observa que la tabla incluye valores nan en las 2 primeras filas, da a entender que son consideradas como cabeceras.

| | Price | Close | High | Low | Open | Volume |
|---|------------|--------------------|--------------------|--------------------|--------------------|-----------|
| 0 | Ticker | AAPL | AAPL | AAPL | AAPL | AAPL |
| 1 | Date | NaN | NaN | NaN | NaN | NaN |
| 2 | 2015-01-02 | 24.320430755615234 | 24.789799848077692 | 23.87997951870223 | 24.7786766820896 | 212818400 |
| 3 | 2015-01-05 | 23.635284423828125 | 24.169164129068307 | 23.4484274604303 | 24.08908208842444 | 257142000 |
| 4 | 2015-01-06 | 23.637516021728516 | 23.897781878192376 | 23.274921764638442 | 23.699801693652454 | 263188400 |

Ilustración 1: impresión inicial de tabla de datos de Yahoo Finance

- Aplico un `.drop()` a la primera fila junto con los valores vacíos.
- Cambio de nombre a las columnas y a partir de ahí aplico un `.info()`, arrojando los siguientes resultados:

```
<class 'pandas.core.frame.DataFrame'>
Index: 2574 entries, 2 to 2575
Data columns (total 6 columns):
#   Column   Non-Null Count  Dtype
---  -
0   Date     2574 non-null   object
1   Close    2574 non-null   object
2   High     2574 non-null   object
3   Low      2574 non-null   object
4   Open     2574 non-null   object
5   Volume   2574 non-null   object
dtypes: object(6)
memory usage: 140.8+ KB
```

Ilustración 2: información de tabla de yahoo finance

- Estoy consciente de que el cambio de tipo de datos corresponde a la fase de data_wrangling, sin embargo, es necesario para poder realizar gráficos iniciales y sobretodo las fechas se encuentren bien formateadas para poder avanzar con el EDA.

```
# cambiar tipo de datos y redondear a 2 decimales en las columnas que aplique

raw_precios_AAPL['Date'] = pd.to_datetime(raw_precios_AAPL['Date'])
raw_precios_AAPL['Close'] = pd.to_numeric(raw_precios_AAPL['Close']).round(2)
raw_precios_AAPL['High'] = pd.to_numeric(raw_precios_AAPL['High']).round(2)
raw_precios_AAPL['Low'] = pd.to_numeric(raw_precios_AAPL['Low']).round(2)
raw_precios_AAPL['Open'] = pd.to_numeric(raw_precios_AAPL['Open']).round(2)
raw_precios_AAPL['Volume'] = pd.to_numeric(raw_precios_AAPL['Volume']).round(2)
```

Ilustración 3: cambio de tipo de datos y redondeo a 2 decimales

- Verifico si hay valores nulos y duplicados en específico por la columna 'Date'.

```
# verificar si hay valores nulos
raw_precios_AAPL.isna().sum()

Date      0
Open      0
High      0
Low       0
Close     0
Volume    0
dtype: int64

raw_precios_AAPL.duplicated(['Date']).sum()

np.int64(0)
```

Ilustración 4: verificación de valores duplicados

Hasta aquí podemos observar que la tabla mayormente no tiene mucho por limpiar, no posee datos nulos y no tendría sentido aplicar una eliminación de datos duplicados ya que la

información se la extrajo por día, sin embargo, se aplicó un. `duplicate` a 'Date' únicamente para confirmar.

1.3 Estadísticas básicas

Se aplico un `.describe()` para sacar estadísticas básicas:

```
raw_precios_AAPL.describe()
```

| | Date | Open | High | Low | Close | Volume |
|-------|-------------------------------|-------------|-------------|-------------|-------------|--------------|
| count | 2574 | 2574.000000 | 2574.000000 | 2574.000000 | 2574.000000 | 2.574000e+03 |
| mean | 2020-02-11 23:11:19.720279808 | 97.198216 | 98.239848 | 96.223754 | 97.284540 | 1.156422e+08 |
| min | 2015-01-02 00:00:00 | 20.596722 | 20.978906 | 20.475435 | 20.674536 | 2.323470e+07 |
| 25% | 2017-07-24 06:00:00 | 35.829304 | 36.023978 | 35.591215 | 35.830737 | 6.963792e+07 |
| 50% | 2020-02-12 12:00:00 | 68.411218 | 69.446233 | 67.555614 | 68.790337 | 9.841010e+07 |
| 75% | 2022-08-31 18:00:00 | 154.880092 | 157.213646 | 152.578035 | 155.051308 | 1.411344e+08 |
| max | 2025-03-27 00:00:00 | 257.906429 | 259.814335 | 257.347047 | 258.735504 | 6.488252e+08 |
| std | NaN | 68.009634 | 68.725053 | 67.347645 | 68.083339 | 6.833532e+07 |

Ilustración 5: estadísticas básicas de la tabla Yahoo Finance

1.4 Gráficos de datos extraídos de Yahoo Finance

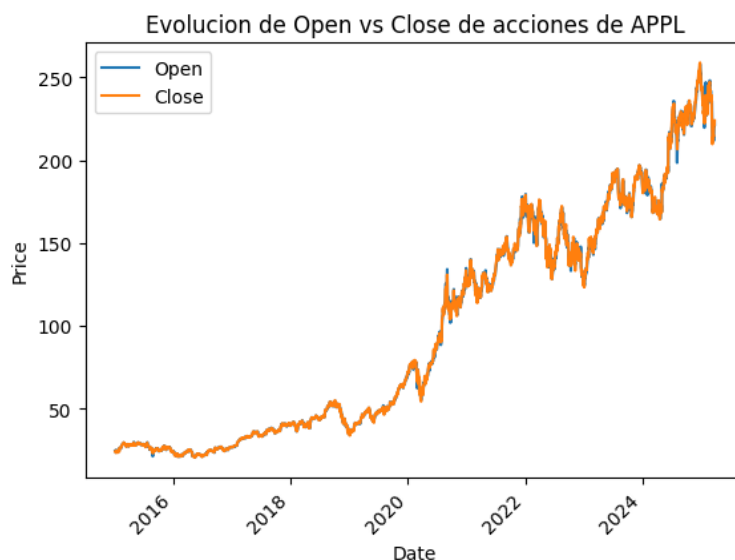


Gráfico 1: Evolución de open y close

En el gráfico se observa variaciones mínimas entre el precio open - close y aquí quiero explicar una parte importante.

Lo ideal sería que el precio con el que cierra el día anterior, abra el siguiente día, sin embargo, esto no ocurre por algo que se llama Horas extendidas (extended hours). Esto significa que al cierre del mercado que es las 16H00 hora EEUU, los inversionistas a través de sus brokers pueden colocar órdenes de compra o venta de acciones luego de esas horas y por efecto de la oferta y demanda, el precio fluctúa. Por este motivo difiere el precio al día siguiente.

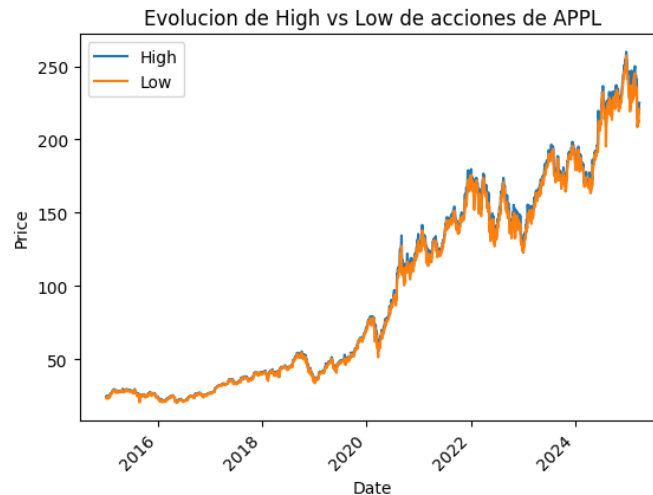


Gráfico 2: Evolución de high y low

Aquí se diferencia de forma más clara la evolución del precio de la acción, como el nombre de las columnas lo indica, se compara el precio más alto vs el más bajo.

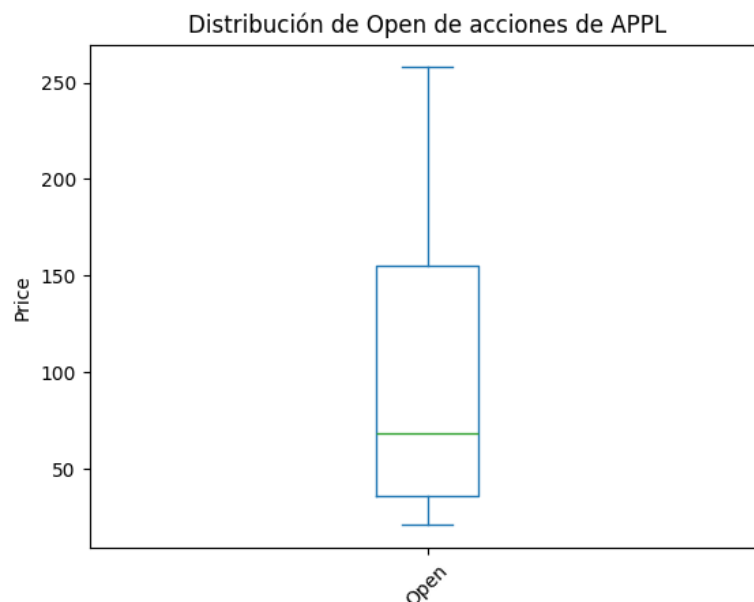


Gráfico 3: verificar outliers

Se observa un comportamiento normal del precio de la acción durante los años que se están considerando para este estudio.

2. DATOS EXTRAÍDOS DE OPENINSIDER.COM

```
raw_insiders_AAPL = pd.read_csv('../Dataset/raw/AAPL_transaccionesOPENINSIDER.csv'), sep = ',',
raw_insiders_AAPL
```

| | X | Filing Date | Trade Date | Ticker | Insider Name | Title | Trade Type | Price | Qty | Owned | ΔOwn | Value | 1d | 1w | 1m | 6m |
|-----|-----|---------------------|------------|--------|---------------------|------------------------------|-------------|----------|----------|-----------|------|---------------|-----|------|------|------|
| 0 | D | 2004-04-21 19:07:40 | 2004-04-19 | AAPL | Heinen Nancy R | SVP | S - Sale+OE | \$28.00 | -200,000 | 1,315 | -99% | -\$5,600,000 | 0.0 | -3.0 | -5.0 | 71.0 |
| 1 | D | 2004-04-21 19:08:35 | 2004-04-19 | AAPL | Tamaddon Sina | SVP | S - Sale+OE | \$28.08 | -678,400 | 6,452 | -99% | -\$19,046,284 | 0.0 | -3.0 | -5.0 | 71.0 |
| 2 | DM | 2004-04-21 19:09:19 | 2004-04-19 | AAPL | Rubinstein Jonathan | SVP | S - Sale+OE | \$28.35 | -250,000 | 9,906 | -96% | -\$7,087,500 | 0.0 | -3.0 | -5.0 | 71.0 |
| 3 | D | 2004-04-21 19:11:31 | 2004-04-19 | AAPL | Cook Timothy D | EVP | S - Sale+OE | \$27.99 | -84,000 | 4,722 | -95% | -\$2,351,280 | 0.0 | -3.0 | -5.0 | 71.0 |
| 4 | D | 2004-04-21 19:12:19 | 2004-04-19 | AAPL | Cook Timothy D | EVP | S - Sale+OE | \$28.21 | -94,000 | 4,722 | -95% | -\$2,651,880 | 0.0 | -3.0 | -5.0 | 71.0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 599 | NaN | 2024-10-08 18:30:13 | 2024-10-04 | AAPL | Maestri Luca | SVP, CFO | S - Sale | \$226.52 | -59,305 | 107,788 | -35% | -\$13,433,769 | NaN | NaN | NaN | NaN |
| 600 | NaN | 2024-11-19 18:30:49 | 2024-11-18 | AAPL | Kondo Chris | Principal Accounting Officer | S - Sale | \$228.87 | -4,130 | 15,419 | -21% | -\$945,233 | NaN | NaN | NaN | NaN |
| 601 | M | 2024-11-19 18:31:42 | 2024-11-15 | AAPL | Levinson Arthur D | Dir | S - Sale | \$227.32 | -200,000 | 4,215,576 | -5% | -\$45,464,500 | NaN | NaN | NaN | NaN |
| 602 | NaN | 2024-12-18 18:30:20 | 2024-12-16 | AAPL | Williams Jeffrey E | COO | S - Sale | \$249.97 | -100,000 | 389,944 | -20% | -\$24,997,395 | NaN | NaN | NaN | NaN |
| 603 | D | 2025-02-04 18:33:25 | 2025-02-03 | AAPL | Levinson Arthur D | Dir | S - Sale+OE | \$226.35 | -1,516 | 4,215,576 | 0% | -\$343,147 | NaN | NaN | NaN | NaN |

604 rows × 16 columns

Ilustración 6: Vista inicial de datos extraídos de OpenInsider.com

- Se aplica un .info() y arroja lo siguiente:

```
raw_insiders_AAPL.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 604 entries, 0 to 603
Data columns (total 16 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   X                409 non-null   object
1   Filing Date      604 non-null   object
2   Trade Date       604 non-null   object
3   Ticker           604 non-null   object
4   Insider Name     604 non-null   object
5   Title            604 non-null   object
6   Trade Type       604 non-null   object
7   Price            604 non-null   object
8   Qty              604 non-null   object
9   Owned             604 non-null   object
10  ΔOwn             604 non-null   object
11  Value            604 non-null   object
12  1d               469 non-null   float64
13  1w               467 non-null   float64
14  1m               464 non-null   float64
15  6m               440 non-null   float64
dtypes: float64(4), object(12)
memory usage: 75.6+ KB
```

Ilustración 7: Información de tabla de OpenInsider.com

- La mayoría de las columnas son de tipo objeto y solo las ultimas 4 son de tipo float.
- Se aplica una validación de valores nan y se obtuvo lo siguiente:


```
raw_insiders_AAPL.isna().sum()

X          195
Filing Date    0
Trade Date     0
Ticker         0
Insider Name   0
Title          0
Trade Type     0
Price          0
Qty           0
Owned          0
DeltaOwned     0
Value          0
1d            135
1w            137
1m            140
6m            164
dtype: int64
```

Ilustración 8: verificación y conteo de valores nan de openinsiders.com

Se encuentran valores nulos en 5 de 16 columnas, de momento no haré nada ya que más adelante estas columnas no representan un impacto significativo dentro del desarrollo del proyecto y tengo planeado eliminarlas.

Apliqué de nuevo un cambio de tipo de datos a la base de datos, esto fue necesario para realizar gráficos iniciales del EDA. Luego del cambio de tipo de datos, las columnas quedaron de la siguiente forma:

```
raw_insiders_AAPL.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 604 entries, 0 to 603
Data columns (total 16 columns):
 #   Column              Non-Null Count  Dtype
---  ---
 0   x                   409 non-null   object
 1   filing_date         604 non-null   datetime64[ns]
 2   trade_date          604 non-null   datetime64[ns]
 3   ticker              604 non-null   object
 4   insider_name        604 non-null   object
 5   title               604 non-null   object
 6   trade_type          604 non-null   object
 7   price               604 non-null   float64
 8   quantity_of_shares  604 non-null   int64
 9   owned               604 non-null   int64
10   delta_owned         604 non-null   object
11   value               604 non-null   int64
12   1d                   469 non-null   float64
13   1w                   467 non-null   float64
14   1m                   464 non-null   float64
15   6m                   440 non-null   float64
dtypes: datetime64[ns](2), float64(5), int64(3), object(6)
memory usage: 75.6+ KB
```

Ilustración 9: comprobación de cambio de datos de openinsiders.com

2.1 Consulta de nombres de directivos y validar que no estén registrados más de una vez o que tengan errores.

```
# consultar los nombres de directivos a lo largo del tiempo y validar que no existan duplicados en los nombres por errores de tipeo o de extracción de datos.
insider_name = raw_insiders_AAPL['insider_name'].unique()
insider_name

array(['Heinen Nancy R', 'Tamaddon Sina', 'Rubinstein Jonathan',
      'Cook Timothy D', 'Anderson Fred D', 'Schiller Philip W',
      'Tevanian Avadis', 'Serlet Bertrand', 'York Jerome B',
      'Johnson Ronald B', 'Levinson Arthur D', 'Oppenheimer Peter',
      'Fadell Anthony', 'Schmidt Eric E', 'Campbell William V',
      'Drexler Millard S', 'Ive Jonathan P', 'Forstall Scott J',
      'Mansfield Robert J', 'Rafael Betsy', 'Papermaster Mark D',
      'Sewell D Bruce', 'Williams Jeffrey E', 'Jung Andrea',
      'Iger Robert A', 'Riccio Daniel J.', 'Mansfield Robert L',
      'Cue Eduardo H', 'Maestri Luca', 'Federighi Craig',
      'Ahrendts Angela J', 'Wagner Susan', 'Srouji Johny', 'Kondo Chris',
      'Gore Albert Jr', 'O'Brien Deirdre', 'Adams Katherine L.'],
      dtype=object)

pd.Series(insider_name).duplicated().sum()

np.int64(0)
```

Ilustración 10: verificación de nombres de directivos de AAPL

Con esta validación quería asegurarme que por ejemplo una persona no tenga registrado su nombre de diferente forma, por ejemplo "TIM COOK", "COOK TIMOTHY D", "TIMOTHY COOK"; las 3 formas antes escritas se refieren a la misma persona. En este caso, no existen diferentes formas de escribir el nombre al referirse a una misma persona, puedo continuar con el análisis.

2.2 Validación y ajuste de cargos duplicados

```
cargo_duplicado = raw_insiders_AAPL['title'].unique()
cargo_duplicado

array(['SVP', 'EVP', 'Dir', 'COO', 'CFO', 'SVP, CFO',
      'VP, Controller, PAO', 'SVP, Gen'l Counsel, Secretary',
      'VP, Corporate Controller', 'CEO', 'Principal Accounting Officer',
      'SVP, GC, Secretary'], dtype=object)

cargo_duplicado_series = pd.Series(cargo_duplicado)
cargo_duplicado_series.duplicated().sum()

np.int64(0)
```

Ilustración 11: validación de cargos duplicados

Realicé el mismo procedimiento anterior, ahora con la columna 'title'. En este caso podemos observar que el registro "SVP", "VP", "Controller" se repiten al momento de imprimir por primera vez "cargo duplicado", sin embargo, al momento de aplicar el .duplicated no lo reconoce, por lo tanto no salta como registro duplicado.

Esto sucede a que pueden llegar a existir varias líneas que contienen SVP más otro tipo de texto, esto lo voy a comprobar a continuación

```
cargo_duplicado_filtrado_svp = cargo_duplicado_series[cargo_duplicado_series.str.contains('SVP')]
print(cargo_duplicado_filtrado_svp)
cargo_duplicado_filtrado_vp = cargo_duplicado_series[cargo_duplicado_series.str.contains('VP')]
print(cargo_duplicado_filtrado_vp)
cargo_duplicado_filtrado_controller = cargo_duplicado_series[cargo_duplicado_series.str.contains('Controller')]
print(cargo_duplicado_filtrado_controller)
```

```
0          SVP
5          SVP, CFO
7  SVP, Gen'l Counsel, Secretary
11         SVP, GC, Secretary
dtype: object
0          SVP
1          EVP
5          SVP, CFO
6  VP, Controller, PAO
7  SVP, Gen'l Counsel, Secretary
8  VP, Corporate Controller
11         SVP, GC, Secretary
dtype: object
6  VP, Controller, PAO
8  VP, Corporate Controller
dtype: object
```

Ilustración 12: evidencia de múltiples cargos asociados a una persona

En efecto, se observa que hay líneas que tienen más de un cargo, esto representaría un problema más adelante ya que se debe escoger solo un cargo. Lo que se me ocurre es reemplazar estos casos con el primer título que tengan, excepto en el caso de SVP, CFO; ahí voy a colocar CFO. Esto lo haré más adelante en la limpieza de datos.

Revisar tipos de transacción que existen

```
# revisar cuantos tipos de transacción existen

raw_insiders_AAPL['trade_type'].unique()
```

```
array(['S - Sale+OE', 'S - Sale', 'P - Purchase'], dtype=object)
```

Ilustración 13: tipos de transacción

Existen 3 tipos de transacciones:

- Sale+OE = significa que una acción fue vendida luego de haberse ejecutado un contrato de opciones. (put option) Es un contrato financiero que otorga al comprador el derecho, pero no la obligación, de vender un activo subyacente a un precio específico (precio de ejercicio o strike price) en o antes de una fecha de vencimiento determinada.

Una opción de venta es una herramienta para protegerse contra la caída del precio de un activo o para especular sobre su posible disminución.

- Sale = la venta normal de las acciones.
- Purchase = compra de acciones

2.3 Revisar cantidades mínima y máximas de cantidad de acciones negociadas

```
# revisar mínimos y máximos en quantity_of_shares correspondiente a ventas

raw_insiders_AAPL['quantity_of_shares'].describe()
```

```
count    6.040000e+02
mean      6.537933e+04
std       1.314147e+05
min       2.500000e+01
25%       1.015475e+04
50%       2.815000e+04
75%       6.647550e+04
max       2.386440e+06
Name: quantity_of_shares, dtype: float64
```

Ilustración 14: revisar mínimos y máximos de la columna 'quantity of shares'

El valor mínimo de acciones negociadas fue de 25 acciones y el máximo fue de 2'386.440. Dado el formato con el que se presentó el anterior output, decido utilizar otro enfoque para estar seguro del valor máximo de acciones negociadas.

```
# revisar cuantas cantidades de acciones se han negociado

sorted_shares = np.sort((raw_insiders_AAPL['quantity_of_shares'] / 1e6).unique()[::-1])

formatted_shares = [f"{int(val * 1e6):,}" for val in sorted_shares]

# Mostrar solo los primeros valores con un formato más legible
print("\n".join(formatted_shares[:50]))
```

✓ 0.0s

2,386,440
700,000
678,400
500,000
470,000
450,000
350,000
348,846
348,425
335,000
334,000
300,000
275,000
269,883
268,644
268,623
265,160
261,934
257,343
257,000
250,000
243,431
240,569
235,000
223,986
...
125,000
123,448
122,000
121,072

Ilustración 15: lista de cantidad de transacciones realizadas ordenadas de mayor a menor

Se observa que la transacción más alta fue de mas de 2'386.400 de acciones y le sigue una de 700.000.

2.4 Gráficos de datos extraídos de OpenInsider.com

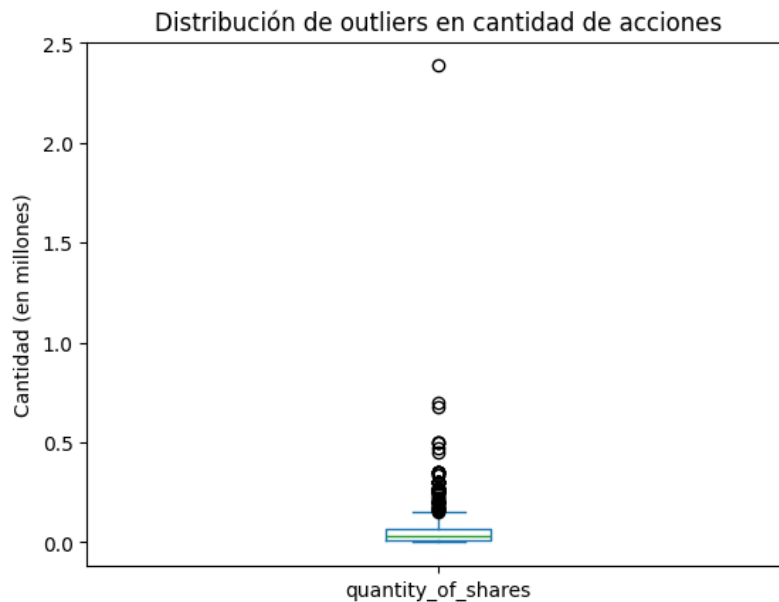


Gráfico 4: Outliers en Openinsiders

La mayoría de las transacciones se concentra entre los 10K a 300K de acciones negociadas, como se evidenció anteriormente, hay una transacción de más de 2 millones de acciones. Esto se puede evidenciar con el scatterplot a continuación:

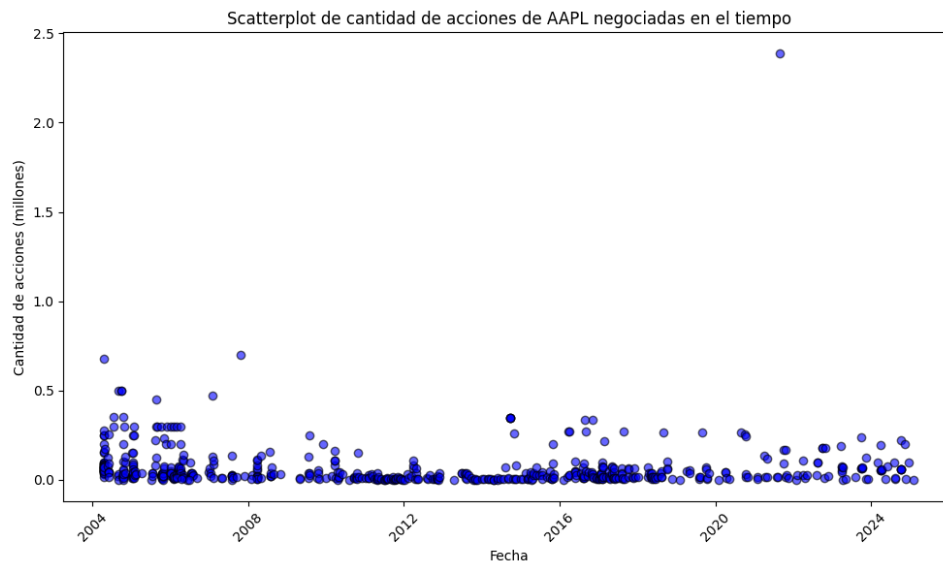


Gráfico 5: Scatterplot de Openinsiders

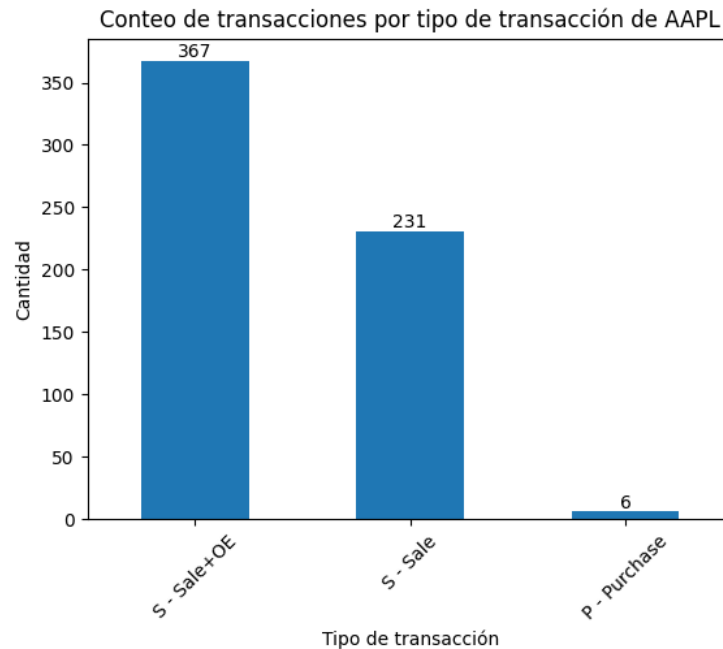


Gráfico 6: conteo de tipo de operaciones

Existen más transacciones por Sale+OE que por Sale y apenas 6 compras de acciones. Las caídas de precio de la acción generalmente están asociadas a las ventas de acciones debido a la especulación o que el mercado ha entrado en pánico.

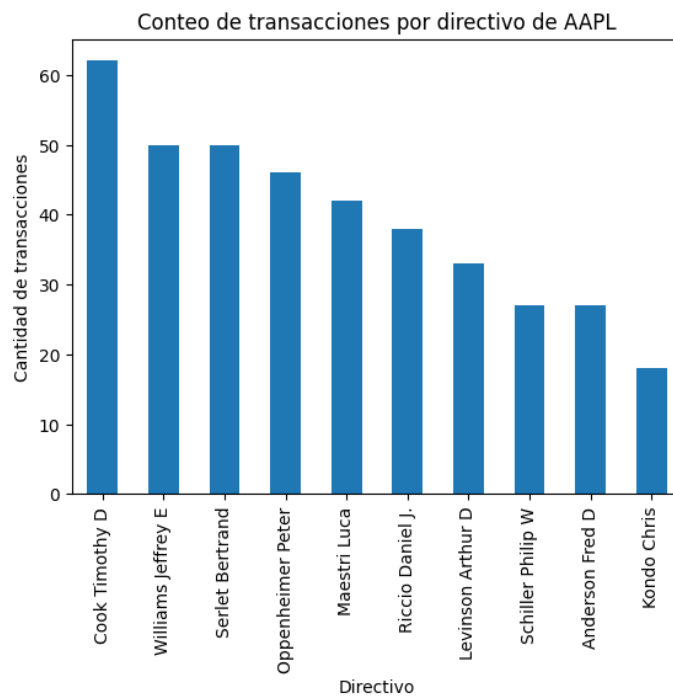


Gráfico 7: Ranking de directivos 1

En un gráfico que se ha extraído los 10 directivos que más han realizado transacciones de acciones se puede observar a Tim Cook, actual CEO.

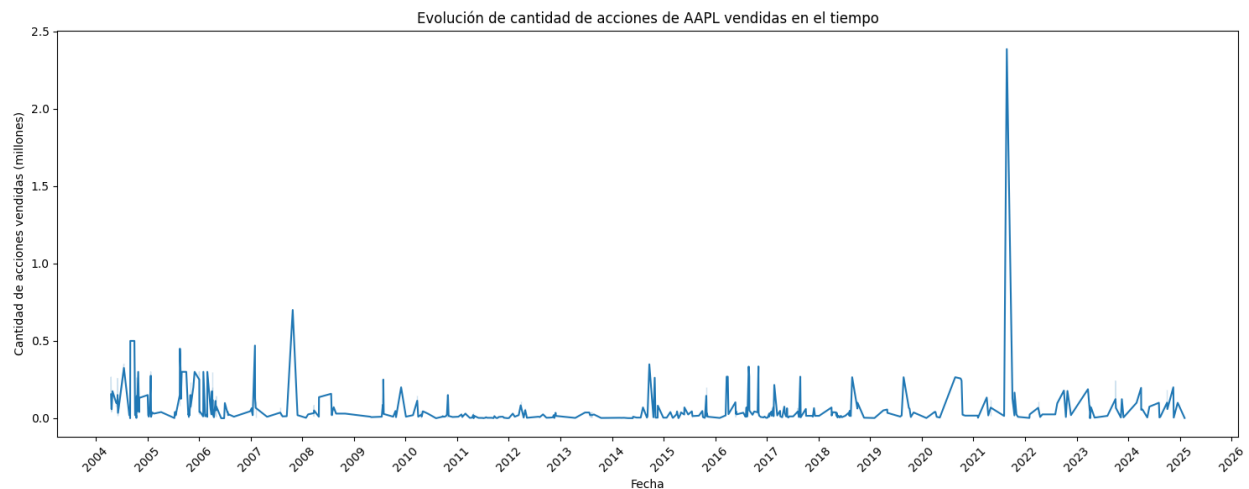


Gráfico 8: Evolución de acciones negociadas

En el 2021 hubo un pico de negociaciones de acciones, es justamente el valor máximo negociado. Como se observa en el gráfico, a lo largo de los años las transacciones se han mantenido entre un rango de 1K a 300K de acciones.

Con esto se concluye el EDA, exportamos los datos a una nueva carpeta y renombramos al archivo.csv.