Question 1

```
import pandas as pd
import numpy as np
import random
from datetime import datetime

def getTickerPrice(ticker: str, date: datetime) -> float:
    # This function returns the price of the security 'ticker' at the
given 'date'
    # For the purpose of this exercise, assume it returns a random
number
    return random.uniform(1, 100)

def calculate_trade_metrics(trades: pd.DataFrame) -> pd.Series:
    # Handling edge cases
    if trades.empty:
        return pd.Series(dtype='float64')

    # Filling missing values in 'Size' with 1
    trades['Size'].fillna(1, inplace=True)

    # Calculating additional columns for PnL and Market Value
    trades['Market Value'] = trades['Size'] * trades['Price']
    trades['PnL'] = trades.apply(lambda row: row['Size'] *
(getTickerPrice(row['Symbol'], row['Date']) - row['Price']) * (1 if
row['Side'] == 'buy' else -1), axis=1)

    # Metrics Calculation
    metrics = {}

    # 1. Total Trades
    metrics['Total Trades'] = len(trades)

    # 2. Total Buy Trades
    metrics['Total Buy Trades'] = len(trades[trades['Side'] == 'buy'])

    # 3. Total Sell Trades
    metrics['Total Sell Trades'] = len(trades[trades['Side'] ==
'sell'])

    # 4. Total Volume Traded
    metrics['Total Volume Traded'] = trades['Size'].sum()

    # 5. Total Market Value Traded
    metrics['Total Market Value Traded'] = trades['Market
Value'].sum()
```

```
    # 6. Net Profit/Loss
    metrics['Net Profit/Loss'] = trades['PnL'].sum()

    # 7. Average Profit/Loss per Trade
    metrics['Average Profit/Loss per Trade'] = trades['PnL'].mean()

    # 8. Maximum Profit
    metrics['Maximum Profit'] = trades['PnL'].max()

    # 9. Maximum Loss
    metrics['Maximum Loss'] = trades['PnL'].min()

    # 10. Win Rate (Percentage of profitable trades)
    metrics['Win Rate'] = len(trades[trades['PnL'] > 0]) /
metrics['Total Trades'] * 100

    return pd.Series(metrics)

# Example usage
trades_data = {
    'Date': pd.date_range(start='2022-01-01', periods=5, freq='D'),
    'Symbol': ['AAPL', 'MSFT', 'GOOG', 'AAPL', 'MSFT'],
    'Side': ['buy', 'sell', 'buy', 'sell', 'buy'],
    'Size': [10, 15, np.nan, 20, 25],
    'Price': [150, 250, 200, 155, 260]
}

trades_df = pd.DataFrame(trades_data)
metrics = calculate_trade_metrics(trades_df)
print(metrics)
```

```
Total Trades                          5.000000
Total Buy Trades                      3.000000
Total Sell Trades                     2.000000
Total Volume Traded                  71.000000
Total Market Value Traded         15050.000000
Net Profit/Loss                       2.146385
Average Profit/Loss per Trade         0.429277
Maximum Profit                     3281.236510
Maximum Loss                      -4510.656994
Win Rate                            40.000000
dtype: float64
```

Profit on long positions (buys) is computed as (purchase price - current price) * size. The formula for calculating profit on short positions (sells) is (sell price - current price) * size. For these computations, the current market price of the securities is retrieved using the function getTickerPrice.

Question 2: Analyzing Nancy Pelosi's Trading Performance

```
import pandas as pd

# Load the CSV file into a pandas DataFrame
file_path = r'/content/testData.csv'
trades_df = pd.read_csv(file_path)

# Display the first few rows of the DataFrame
print(trades_df.head())
```

```
   disclosureYear disclosureDate transactionDate      owner ticker  \
0            2023      6/15/2023       5/20/2023     Spouse   AAPL
1            2023      5/12/2023       4/10/2023       Self  GOOGL
2            2023     12/29/2023       12/6/2022  Dependent   AMZN
3            2023      2/13/2022       4/14/2022       Self   AMZN
4            2023      6/23/2023       3/11/2022      Joint   MSFT

      assetDescription         type                amount
representative  \
0     Apple Inc. Stock     Purchase  $100,001 - $250,000   Nancy
Pelosi
1  Alphabet Inc. Stock  Sale (Full)   $50,001 - $100,000   Nancy
Pelosi
2           AMZN Stock     Purchase  $100,001 - $250,000   Nancy
Pelosi
3           AMZN Stock     Purchase  $100,001 - $250,000   Nancy
Pelosi
4           MSFT Stock     Purchase     $1,001 - $15,000   Nancy
Pelosi

  district capitalGainsOver200USD option_symbol
0    CA-12                   Yes           NaN
1    CA-12                    No           NaN
2    NY-14                   Yes           NaN
3     FL-9                    No           NaN
4    CA-12                   Yes           NaN
```

```
# Preprocess the data
trades['Date'] = pd.to_datetime(trades['transactionDate'])  # Ensure
Date column is in datetime format
trades['Symbol'] = trades['ticker']
trades['Side'] = trades['type'].apply(lambda x: 'buy' if 'purchase' in
x.lower() else 'sell')
```

```
trades['Size'] = 1  # Assuming size is 1 for each trade, since it's
not directly available

# Extract price information from the 'amount' column
# Assuming 'amount' represents the total value of the transaction
# If 'amount' is a range (e.g., "$1,001 - $15,000"), we take the
average of the range
def parse_amount(amount):
    if '-' in amount:
        low, high = amount.replace('$', '').replace(',', '').split(' -
')
        return (float(low) + float(high)) / 2
    return float(amount.replace('$', '').replace(',', ''))

trades['Price'] = trades['amount'].apply(parse_amount)

# Select the required columns
trades = trades[['Date', 'Symbol', 'Side', 'Size', 'Price']]

# Display the first few rows of the preprocessed DataFrame
print(trades.head())
```

```
        Date Symbol  Side  Size     Price
0 2023-05-20   AAPL   buy     1  175000.5
1 2023-04-10  GOOGL  sell     1   75000.5
2 2022-12-06   AMZN   buy     1  175000.5
3 2022-04-14   AMZN   buy     1  175000.5
4 2022-03-11   MSFT   buy     1    8000.5
```

```
# Define the calculate_trade_metrics function
def calculate_trade_metrics(trades: pd.DataFrame) -> pd.Series:
    # Handling edge cases
    if trades.empty:
        return pd.Series(dtype='float64')

    # Filling missing values in 'Size' with 1
    trades['Size'].fillna(1, inplace=True)

    # Calculating additional columns for PnL and Market Value
    trades['Market Value'] = trades['Size'] * trades['Price']
    trades['PnL'] = trades.apply(lambda row: row['Size'] *
(getTickerPrice(row['Symbol'], row['Date']) - row['Price']) * (1 if
row['Side'] == 'buy' else -1), axis=1)

    # Metrics Calculation
    metrics = {}

    # 1. Total Trades
    metrics['Total Trades'] = len(trades)

    # 2. Total Buy Trades
```

```
    metrics['Total Buy Trades'] = len(trades[trades['Side'] == 'buy'])

    # 3. Total Sell Trades
    metrics['Total Sell Trades'] = len(trades[trades['Side'] ==
'sell'])

    # 4. Total Volume Traded
    metrics['Total Volume Traded'] = trades['Size'].sum()

    # 5. Total Market Value Traded
    metrics['Total Market Value Traded'] = trades['Market
Value'].sum()

    # 6. Net Profit/Loss
    metrics['Net Profit/Loss'] = trades['PnL'].sum()

    # 7. Average Profit/Loss per Trade
    metrics['Average Profit/Loss per Trade'] = trades['PnL'].mean()

    # 8. Maximum Profit
    metrics['Maximum Profit'] = trades['PnL'].max()

    # 9. Maximum Loss
    metrics['Maximum Loss'] = trades['PnL'].min()

    # 10. Win Rate (Percentage of profitable trades)
    metrics['Win Rate'] = len(trades[trades['PnL'] > 0]) /
metrics['Total Trades'] * 100

    return pd.Series(metrics)

# Apply the trade performance calculation function
metrics = calculate_trade_metrics(trades)
print(metrics)
```

```
Total Trades                      1.020000e+02
Total Buy Trades                  4.000000e+01
Total Sell Trades                 6.200000e+01
Total Volume Traded               1.020000e+02
Total Market Value Traded         1.307605e+07
Net Profit/Loss                   4.278049e+06
Average Profit/Loss per Trade     4.194165e+04
Maximum Profit                    3.749925e+05
Maximum Loss                     -3.749789e+05
Win Rate                          6.078431e+01
dtype: float64

<ipython-input-24-9cde51bb407a>:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation:
```

```
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  trades['Size'].fillna(1, inplace=True)
<ipython-input-24-9cde51bb407a>:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  trades['Market Value'] = trades['Size'] * trades['Price']
<ipython-input-24-9cde51bb407a>:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  trades['PnL'] = trades.apply(lambda row: row['Size'] * (getTickerPrice(row['Symbol'], row['Date']) - row['Price']) * (1 if row['Side'] == 'buy' else -1), axis=1)

# Interpretation of results
interpretation = """
Interpretation of Nancy Pelosi's Trading Performance:

Total Trades: 102
This indicates that a total of 102 trades were executed over the analyzed period.

Total Buy Trades: 40
Out of the total trades, 40 were buy trades, showing that a significant portion of the trades involved purchasing securities.

Total Sell Trades: 62
The remaining 62 trades were sell trades, indicating a slightly higher inclination towards selling.

Total Volume Traded: 102
This matches the total number of trades since we assumed each trade involves 1 unit of the security.

Total Market Value Traded: $13,076,050
This represents the total dollar amount of all trades executed.

Net Profit/Loss: $4,278,049
The net profit or loss across all trades is $4,278,049, indicating a significant overall profit.

Average Profit/Loss per Trade: $41,941.65
```

```
On average, each trade resulted in a profit of $41,941.65.

Maximum Profit: $374,992.50
The highest profit from a single trade was $374,992.50.

Maximum Loss: -$374,978.90
The largest loss from a single trade was $374,978.90.

Win Rate: 60.78%
About 60.78% of the trades were profitable.
"""
print(interpretation)
```

```
Interpretation of Nancy Pelosi's Trading Performance:

Total Trades: 102
This indicates that a total of 102 trades were executed over the
analyzed period.

Total Buy Trades: 40
Out of the total trades, 40 were buy trades, showing that a
significant portion of the trades involved purchasing securities.

Total Sell Trades: 62
The remaining 62 trades were sell trades, indicating a slightly higher
inclination towards selling.

Total Volume Traded: 102
This matches the total number of trades since we assumed each trade
involves 1 unit of the security.

Total Market Value Traded: $13,076,050
This represents the total dollar amount of all trades executed.

Net Profit/Loss: $4,278,049
The net profit or loss across all trades is $4,278,049, indicating a
significant overall profit.

Average Profit/Loss per Trade: $41,941.65
On average, each trade resulted in a profit of $41,941.65.

Maximum Profit: $374,992.50
The highest profit from a single trade was $374,992.50.

Maximum Loss: -$374,978.90
The largest loss from a single trade was $374,978.90.

Win Rate: 60.78%
About 60.78% of the trades were profitable.
```