Bachelor Thesist Data

Patrick Wang

Data

This paper investigates the influence of oil price volatility on Taiwan aggregate and individual stock returns from January 1, 2010, to December 31, 2021, because the Taiwan stock market's market structure has become more stable since the 2008 financial crisis. In the empirical analysis, the daily data are used because the OVX index is a way to measure how volatile oil prices are in the short term. All the return data are denoted as daily return percentage.

Update working directroy and import packages

```
import pandas as pd
import yfinance as yf
import os
import requests
import csv
from bs4 import BeautifulSoup as bs

# Get the current working directory
print(os.getcwd())
os.chdir('/Users/patrickwang/Documents/ovx-py-repo/')
print(os.getcwd())
```

```
/Users/patrickwang/Documents/ovx-py-repo/data
/Users/patrickwang/Documents/ovx-py-repo
```

Default settings

```
start_date = "2010-01-01"
end_date = "2021-12-31"
num_picks = 105
```

This funciton download and clean yfinance price data

```
def clean_price_data(ticker, start_date, end_date, download=False):
    """
    Clean and process price data for a given ticker and date range.

    Args:
        ticker (str): Ticker symbol.
        start_date (str): Start date in "YYYY-MM-DD" format.
        end_date (str): End date in "YYYY-MM-DD" format.
        download (bool, optional): Whether to download and save the data to CSV.

    Returns:
        pd.DataFrame: Processed price data.
    """
    data_df = yf.download(ticker, start=start_date, end=end_date)

    # save the data to csv
    if download == True:
        cleaned_ticker = ''.join([char for char in ticker if char.isalnum()])
        csv_filename = f"data/{cleaned_ticker.lower()}_data.csv"
        data_df.to_csv(csv_filename)
    else: None

    # The difference betwen consecutive 'Close' data
    data_df['diff']  = data_df['Close'].diff()
    data_df['pct_return'] = data_df['Close'].pct_change() * 100

    # Drop Open, High, Low, Adj Close, Volume
    columns_to_drop = ['Open', 'High', 'Low', 'Adj Close', 'Volume']
    data_df = data_df.drop(columns_to_drop, axis=1)

    return(data_df)
```

Clean OVX data

```
ovx_df = clean_price_data('^OVX', start_date, end_date, download=True)

# Create two columns contain positive and negative ovx change
ovx_df['diff_pos'] = ovx_df['diff'][ovx_df['diff'] > 0]
ovx_df['diff_neg'] = ovx_df['diff'][ovx_df['diff'] <= 0]

# Fill positive and negative ovx change NaN with 0 for regression
ovx_df['diff_pos'] = ovx_df['diff_pos'].fillna(0)
ovx_df['diff_neg'] = ovx_df['diff_neg'].fillna(0)
```

```
[**********************100%%**********************]  1 of 1 completed
```

Clean ^TWII and USDTWD=X data

```
twii_df = clean_price_data('^TWII', start_date, end_date)
fx_df = clean_price_data('USDTWD=X', start_date, end_date)
```

```
[**********************100%%**********************]  1 of 1 completed
[**********************100%%**********************]  1 of 1 completed
```

Find top 105 market cap symbols

```
# Extract ranked stock symbols from taifex website
URL = "https://www.taifex.com.tw/cht/9/futuresQADetail"
response = requests.get(URL)
html = response.content

# Create a BeautifulSoup object to parse the HTML
soup = bs(html, 'html.parser')
target_td_elements = soup.find_all('td', {'align': 'right', 'headers': 'name_a'})

tickers, companies = [], []
for idx, rows in enumerate(target_td_elements):
    # alternate between ticker and company name
    clean_text = ''.join([char for char in rows.text if char.isalnum()])

    # ticker start with 0
```

```
    tickers.append(clean_text) if idx % 2 == 0 else companies.append(clean_text)

# Save the top 105 ticker symbols into csv file
top_tickers = tickers[:num_picks]
with open("data/top_ticker_symbols.csv", "w", newline="") as csvfile:
    writer = csv.writer(csvfile)
    writer.writerow(["Ticker Symbol"])
    writer.writerows([[ticker] for ticker in top_tickers])
```

Get top 105 stock returns

```
# Create dataframe contain top 105 tickers (not optimized)
filepath = 'data/top_ticker_symbols.csv'
top_ticker_df = pd.read_csv(filepath)

## Check: print(top_ticker_df.shape)

individual_df = pd.DataFrame()
for symbol in top_ticker_df['Ticker Symbol']:
    symbol_txt = str(symbol) + '.TW'
    individual_df[symbol] = clean_price_data(symbol_txt, start_date, end_date)['pct_return
```

```
[**********************100%%**********************]  1 of 1 completed
[**********************100%%**********************]  1 of 1 completed
[**********************100%%**********************]  1 of 1 completed
[**********************100%%**********************]  1 of 1 completed
[**********************100%%**********************]  1 of 1 completed
[**********************100%%**********************]  1 of 1 completed
[**********************100%%**********************]  1 of 1 completed
[**********************100%%**********************]  1 of 1 completed
[**********************100%%**********************]  1 of 1 completed
[**********************100%%**********************]  1 of 1 completed
[**********************100%%**********************]  1 of 1 completed
[**********************100%%**********************]  1 of 1 completed
[**********************100%%**********************]  1 of 1 completed
[**********************100%%**********************]  1 of 1 completed
[**********************100%%**********************]  1 of 1 completed
[**********************100%%**********************]  1 of 1 completed
[**********************100%%**********************]  1 of 1 completed
```

```
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
```

```
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
```

```
[*********************100%%**********************]  1 of 1 completed

/var/folders/_k/zqfq3vl52tjdjgflf3v567240000gn/T/ipykernel_86238/3078830461.py:10: Performanc
  individual_df[symbol] = clean_price_data(symbol_txt, start_date, end_date)['pct_return']
/var/folders/_k/zqfq3vl52tjdjgflf3v567240000gn/T/ipykernel_86238/3078830461.py:10: Performanc
  individual_df[symbol] = clean_price_data(symbol_txt, start_date, end_date)['pct_return']
/var/folders/_k/zqfq3vl52tjdjgflf3v567240000gn/T/ipykernel_86238/3078830461.py:10: Performanc
  individual_df[symbol] = clean_price_data(symbol_txt, start_date, end_date)['pct_return']
/var/folders/_k/zqfq3vl52tjdjgflf3v567240000gn/T/ipykernel_86238/3078830461.py:10: Performanc
  individual_df[symbol] = clean_price_data(symbol_txt, start_date, end_date)['pct_return']
/var/folders/_k/zqfq3vl52tjdjgflf3v567240000gn/T/ipykernel_86238/3078830461.py:10: Performanc
  individual_df[symbol] = clean_price_data(symbol_txt, start_date, end_date)['pct_return']
```