Impact of Oil Price Volatility on Taiwanese Stock Return

Patrick Wang

2023-09-07

Table of contents

1. Load Data **1**
1.1 Packages and default settings . . . 1
1.2 Download and clean price data . . . 2
1.2.1 Clean OVX data . . . 2
1.2.2 Clean ^TWII ,USDTWD=X and ^TNX data . . . 3
1.2.3 Find top 105 market cap symbols . . . 3
1.2.4 Get top 105 stock returns . . . 3
1.3 Data Check . . . 4

2. Data Visualization **5**
2.1 Plot CBOE Crude Oil Volatility Index . . . 5
2.2 Plot TSEC weighted index . . . 5
2.3 Plot Interest Rate . . . 5

3. Model OVX movement on aggregate stock return **5**
3.1 Base Model . . . 5
3.2 Symmetric Impact . . . 8
3.3 Asymmetric Impact . . . 9

4. Model OVX movement on individual stock return **10**
4.1 Post-ranking quartly Jensen's alpha . . . 10
4.2 Impact on individual stock return . . . 10
4.3 . . . 10

1. Load 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.

1.1 Packages and default settings

```python
import pandas as pd
import yfinance as yf
import os
import requests
import csv
from bs4 import BeautifulSoup as bs
import matplotlib.pyplot as plt
import statsmodels.api as sm
```

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

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

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

1.2 Download and clean price data

```python
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)
```

1.2.1 Clean OVX data

```python
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
```

1.2.2 Clean ^TWII ,USDTWD=X and ^TNX data

```
# Calculate market premium if have time !!!
twii_df = clean_price_data('^TWII', start_date, end_date)
fx_df = clean_price_data('USDTWD=X', start_date, end_date)
int_df = clean_price_data('^TNX', start_date, end_date)
```

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

1.2.3 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])
```

1.2.4 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']
```

1.3 Data Check

```
print(ovx_df.head())
print(twii_df.head())
print(fx_df.head())
print(int_df.head())
print(individual_df.head())
```

```
                Close      diff  pct_return  diff_pos   diff_neg
Date
2010-01-04  35.439999       NaN         NaN  0.000000   0.000000
2010-01-05  34.270000 -1.169998   -3.301349  0.000000 -1.169998
2010-01-06  34.400002  0.130001    0.379344  0.130001   0.000000
2010-01-07  33.610001 -0.790001   -2.296514  0.000000 -0.790001
2010-01-08  31.340000 -2.270000   -6.753944  0.000000 -2.270000
                  Close        diff  pct_return
Date
2010-01-04  8207.849609         NaN         NaN
2010-01-05  8211.400391    3.550781    0.043261
2010-01-06  8327.620117  116.219727    1.415346
2010-01-07  8237.419922  -90.200195   -1.083145
2010-01-08  8280.900391   43.480469    0.527841
                Close      diff  pct_return
Date
2010-01-04  31.660000       NaN         NaN
2010-01-05  31.860001  0.200001    0.631714
2010-01-06  31.780001 -0.080000   -0.251098
2010-01-07  31.809999  0.029999    0.094395
2010-01-08  31.790001 -0.019999   -0.062869
            Close   diff  pct_return
Date
2010-01-04  3.841    NaN         NaN
2010-01-05  3.755 -0.086   -2.238999
2010-01-06  3.808  0.053    1.411451
2010-01-07  3.822  0.014    0.367645
2010-01-08  3.808 -0.014   -0.366299
                2330      2317      2454      2382      2412      2308  \
Date
2010-01-04       NaN       NaN       NaN       NaN       NaN       NaN
2010-01-05 -0.616335  0.660067  0.176370 -0.725689  0.168630 -0.696519
2010-01-06  0.620157 -0.655739  1.232392  1.461988  0.336702 -0.400801
2010-01-07 -1.078589 -0.990101 -2.434778 -0.864551 -2.013425 -1.810868
2010-01-08 -0.311522  0.333334 -1.960782 -0.436051 -0.513697 -0.102457

                2881      6505      2882      2303  ...      2352  8464  \
Date                                                ...
2010-01-04       NaN       NaN       NaN       NaN  ...       NaN   NaN
2010-01-05  0.637756  0.728153 -0.168075  1.176475  ... -3.605770   0.0
2010-01-06  3.168571  2.168678  1.683503  6.976737  ...  0.498755   0.0
2010-01-07 -1.597052  0.825468 -0.993378  0.000000  ... -2.481390   0.0
2010-01-08  0.499372  0.233915  1.003345 -1.086950  ...  1.781173   0.0

                2059      3035      9904      1477      2354      2385  2633  \
```

```
Date
2010-01-04       NaN       NaN       NaN       NaN       NaN       NaN   NaN
2010-01-05 -1.977398 -2.463769  3.162061  0.166110  0.000000 -1.948055   0.0
2010-01-06  0.864555  2.080240 -0.574719  0.829189  0.000000  0.000000   0.0
2010-01-07 -0.857144 -3.202328 -0.770709 -1.644741 -1.639348  0.264900   0.0
2010-01-08 -0.288188  1.203004  1.165046  0.000000  0.000000  2.245709   0.0

                1519
Date
2010-01-04       NaN
2010-01-05 -1.826483
2010-01-06  0.155038
2010-01-07  0.000000
2010-01-08 -0.773991

[5 rows x 105 columns]
```

2. Data Visualization

```
def plot_and_save(df, label, filename):
    plt.figure().gca().tick_params(axis='both', which='both', length=0.02)
    plt.plot(df.index, df['Close'], 'k', lw=1.25, label=label)
    plt.xticks(rotation=45, fontsize=8)  # Changed fontsize to 8
    plt.yticks(fontsize=8)  # Changed fontsize to 8
    plt.legend(loc='upper right', prop={'size': 10})
    plt.tight_layout()
    plt.savefig(filename)
    plt.close()
```

2.1 Plot CBOE Crude Oil Volatility Index

```
plot_and_save(ovx_df, 'CBOE Crude Oil Volatility Index', 'plots/ovx.png')
```

2.2 Plot TSEC weighted index

```
plot_and_save(twii_df, 'TSEC weighted index', 'plots/twii.png')
```

2.3 Plot Interest Rate

```
plot_and_save(int_df, 'US 10-Year Treasury Bond Yield', 'plots/int_rate.png')
```

3. Model OVX movement on aggregate stock return

3.1 Base Model

Set up the Base model without incorporating OVX:

$R^m_{prem,t} = \beta_0 + \beta_1 FX_t + \beta_2 R^m_{prem,t-1} + \varepsilon_t$

The dependent variable $R^m_{prem,t} = R_{m,t} - R_{f,t}$ represents the daily Taiwan aggregate stock market excess return (pending), FX_t represents the foreign exchange rate between the New Taiwan Dollar and the U.S. Dollar, and ΔOVX_t represents the OVX index movement between time t-1 and time t. To reflect earlier market condition and momentum, the lagged market premium is added. Moreover, because international investments have a big influence

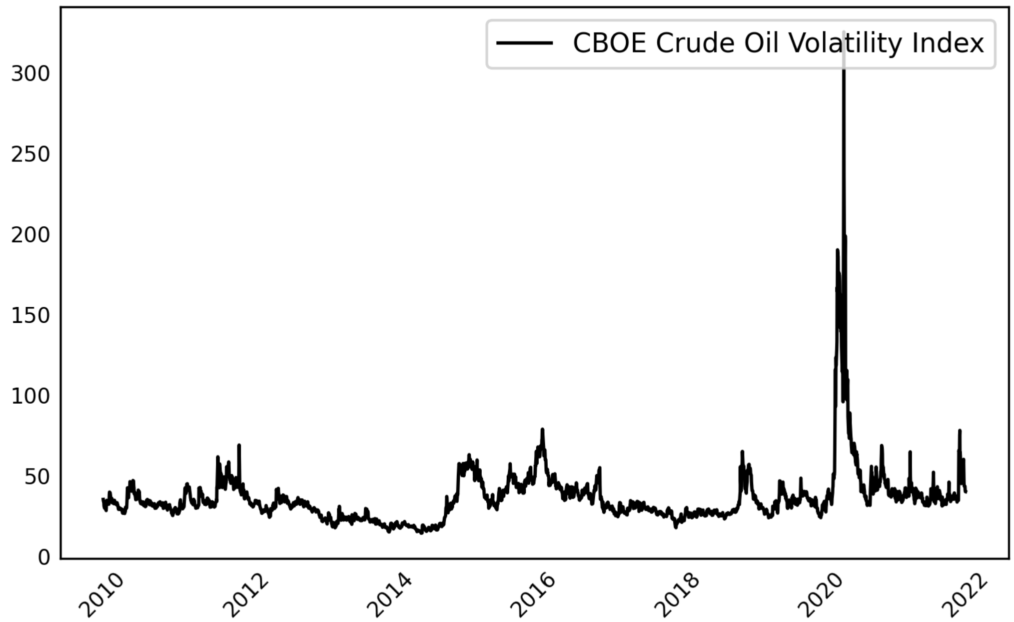

Figure 1: “CBOE Crude Oil Volatility Index”

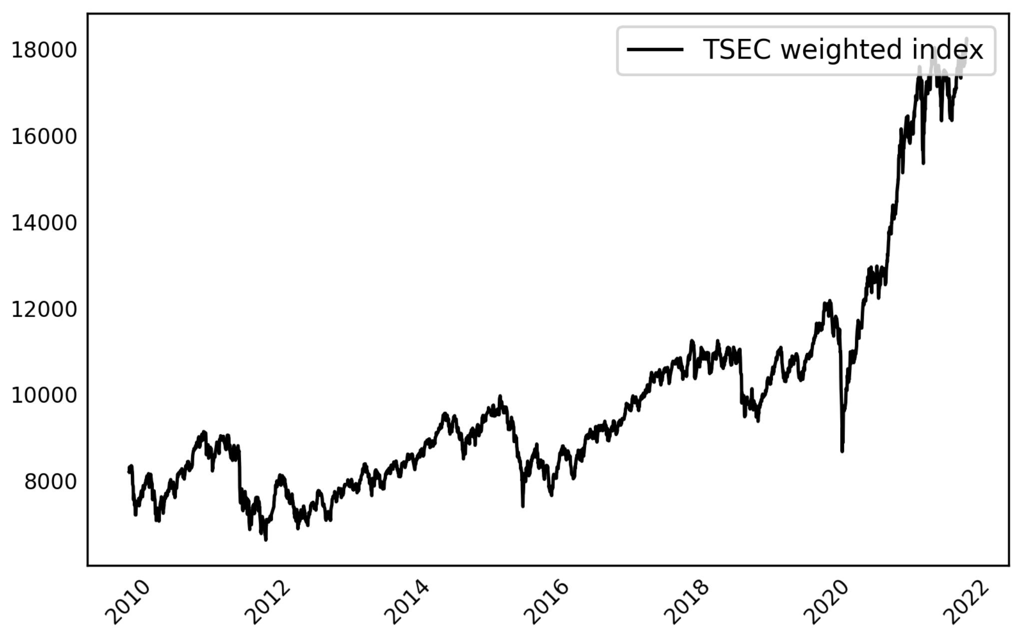

Figure 2: “TSEC weighted index”

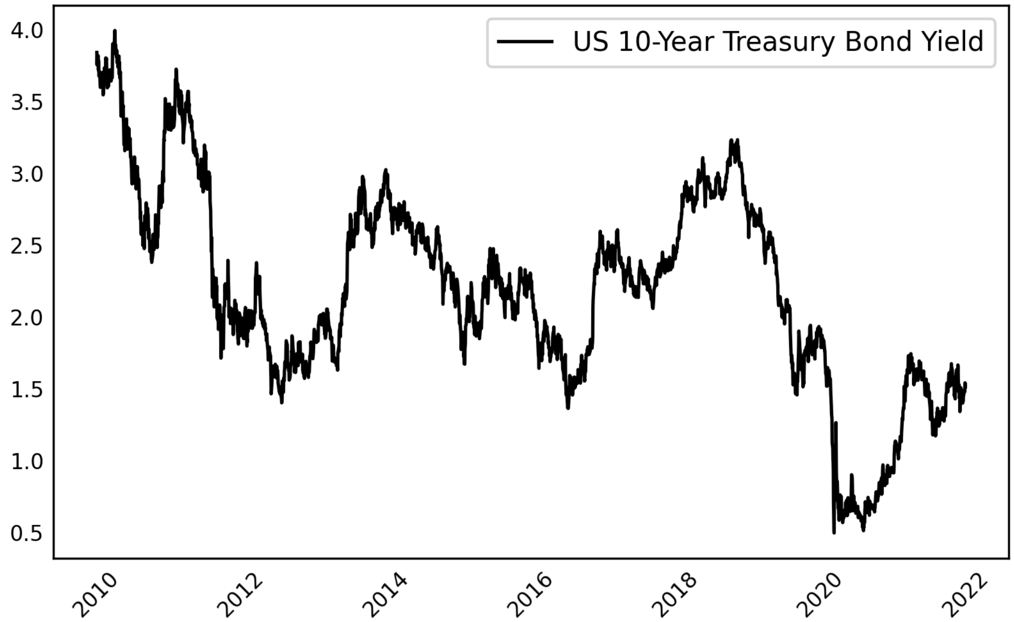

Figure 3: "US 10-Year Treasury Bond Yield"

on Taiwan's stock market, the foreign exchange rate between the New Taiwan Dollar and the US Dollar is also included.

```
# Align the dataframes by date index and drop rows with missing data
merged_df = pd.concat([twii_df['pct_return'], fx_df['Close']], axis=1, keys=['mkt_rt', 'fx'])
merged_df.dropna(inplace=True)

# Create lagged variable for twii_df['pct_return']
merged_df['lag_mkt_rt'] = merged_df['mkt_rt'].shift(1)

# Drop any rows with NaN values (due to lagged variables or otherwise)
merged_df.dropna(inplace=True)

# Separate independent and dependent variables
X = merged_df[['fx', 'lag_mkt_rt']]
y = merged_df['mkt_rt']

# Add a constant (intercept) to the independent variables
X = sm.add_constant(X)

# Run the regression
model = sm.OLS(y, X)
result0 = model.fit()
print(result0.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 mkt_rt   R-squared:                       0.001
Model:                            OLS   Adj. R-squared:                  0.001
Method:                 Least Squares   F-statistic:                     2.161
Date:                Thu, 07 Sep 2023   Prob (F-statistic):              0.115
Time:                        01:53:49   Log-Likelihood:                -4073.0
```

```
No. Observations:                2931   AIC:                             8152.
Df Residuals:                    2928   BIC:                             8170.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.2318      0.370      0.627      0.531      -0.493       0.957
fx            -0.0067      0.012     -0.545      0.586      -0.031       0.017
lag_mkt_rt     0.0366      0.018      1.983      0.048       0.000       0.073
==============================================================================
Omnibus:                      404.361   Durbin-Watson:                   2.000
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2782.548
Skew:                          -0.453   Prob(JB):                         0.00
Kurtosis:                       7.687   Cond. No.                         623.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

When we use the exchange rate between TWD and USD as well as the lagged market premium as independent variables, we can see that the coefficient for the exchange rate is significant at a confidence interval of ??%, while the coefficient for the lagged market premium is significant at a confidence interval of ??%. Because the foreign exchange rate is computed by converting 1 USD to TWD, the result shows that the value of the New Taiwan Dollar has a positive impact on the aggregate stock return. Furthermore, the lagged dependent variable has a positive effect on market excess return, indicating that the Taiwan stock market retains momentum beyond trading days. These findings are consistent across three models. The introduction of the OVX index into the model simply changes the magnitude of the effect.

3.2 Symmetric Impact

We first assume that the impact of OVX on Taiwan stock returns is symmetric:

$R^m_{prem,t} = \beta_0 + \beta_1 FX_t + \beta_2 R^m_{prem,t-1} + \beta_3 \Delta OVX_t + \varepsilon_t$

```
merged_df = pd.concat([twii_df['pct_return'], fx_df['Close'], ovx_df['diff']], axis=1, keys=['mkt_rt', '
merged_df.dropna(inplace=True)

merged_df['lag_mkt_rt'] = merged_df['mkt_rt'].shift(1)
merged_df.dropna(inplace=True)

X = merged_df[['fx', 'lag_mkt_rt', 'ovx_diff']]
X = sm.add_constant(X)
y = merged_df['mkt_rt']

model = sm.OLS(y, X)
result1 = model.fit()
print(result1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 mkt_rt   R-squared:                       0.017
Model:                            OLS   Adj. R-squared:                  0.016
Method:                 Least Squares   F-statistic:                     15.92
Date:                Thu, 07 Sep 2023   Prob (F-statistic):           2.89e-10
Time:                        01:53:49   Log-Likelihood:                -3946.7
No. Observations:                2839   AIC:                             7901.
Df Residuals:                    2835   BIC:                             7925.
```

```
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.2420      0.374      0.646      0.518      -0.492       0.976
fx            -0.0072      0.012     -0.578      0.563      -0.031       0.017
lag_mkt_rt     0.0304      0.019      1.629      0.104      -0.006       0.067
ovx_diff      -0.0247      0.004     -6.687      0.000      -0.032      -0.017
==============================================================================
Omnibus:                      383.678   Durbin-Watson:                   2.032
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2915.044
Skew:                          -0.398   Prob(JB):                         0.00
Kurtosis:                       7.900   Cond. No.                         620.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

According to the empirical result, increased levels of uncertainty in the oil market are associated with decreased levels of average stock returns in Taiwan. One of the possible explanations for this phenomenon is that, given that oil is a crucial component in the manufacture of a wide variety of goods, an increase in the volatility of oil prices has a deleterious effect on the investments that corporations make in the real economy, which, in turn, has a negative impact on the stock returns.

3.3 Asymmetric Impact

After assessing the symmetrical impact of OVX on Taiwan stock market, it is sufficient to execute an analysis of the asymmetric influence of OVX movement on the Taiwan stock market by splitting the relevant data regarding OVX movement into two distinct groups.

$R^m_{prem,t} = \beta_0 + \beta_1 FX_t + \beta_2 R^m_{prem,t-1} + \beta_{31}\Delta OVX^+_t + \beta_{32}\Delta OVX^-_t + \varepsilon_t$

$\Delta OVX^+_t = max(\Delta OVX_t, 0)$ and $\Delta OVX^-_t = min(\Delta OVX_t, 0)$ denote positive and negative OVX shock, respectively. The setting in the regression model (2) is similar to Xiao et. al. (2018), who compare the asymmetric impacts of oil price uncertainty on Chinese stock returns.

```python
merged_df = pd.concat([twii_df['pct_return'], fx_df['Close'], ovx_df['diff_pos'], ovx_df['diff_neg'], ],
                        axis=1, keys=['mkt_rt', 'fx', 'ovx_pos_diff', 'ovx_neg_diff'])
merged_df.dropna(inplace=True)

merged_df['lag_mkt_rt'] = merged_df['mkt_rt'].shift(1)
merged_df.dropna(inplace=True)

X = merged_df[['fx', 'lag_mkt_rt', 'ovx_pos_diff', 'ovx_neg_diff']]
X = sm.add_constant(X)
y = merged_df['mkt_rt']

model = sm.OLS(y, X)
result2 = model.fit()
print(result2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 mkt_rt   R-squared:                       0.017
Model:                            OLS   Adj. R-squared:                  0.015
Method:                 Least Squares   F-statistic:                     11.94
Date:                Thu, 07 Sep 2023   Prob (F-statistic):           1.27e-09
```

```
Time:                        01:53:49   Log-Likelihood:                -3946.7
No. Observations:                2839   AIC:                              7903.
Df Residuals:                    2834   BIC:                              7933.
Df Model:                           4
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const            0.2436      0.375      0.650      0.516      -0.491       0.978
fx              -0.0072      0.012     -0.580      0.562      -0.032       0.017
lag_mkt_rt       0.0303      0.019      1.625      0.104      -0.006       0.067
ovx_pos_diff    -0.0250      0.005     -5.133      0.000      -0.035      -0.015
ovx_neg_diff    -0.0241      0.006     -3.957      0.000      -0.036      -0.012
==============================================================================
Omnibus:                      383.410   Durbin-Watson:                   2.032
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2921.583
Skew:                          -0.396   Prob(JB):                         0.00
Kurtosis:                       7.906   Cond. No.                         621.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

We can observe that both positive and negative movement of the OVX have a negative impact on aggregate stock return. This means that any change in the degree of oil price uncertainty will always have a negative impact on the stock market. In other words, whether the price of oil rises or falls in a short amount of time, the stock market will suffer. We can also observe that a positive change in oil price uncertainty has a greater negative impact on aggregate stock return than a negative change. This conclusion is consistent with the proposed explanation in the first model, in which we say that the volatility of oil prices has a negative influence on corporate investments in the real economy.

4. Model OVX movement on individual stock return

4.1 Post-ranking quartly Jensen's alpha

4.2 Impact on individual stock return

4.3