Python

January 17, 2024

1

___
###

#
Python
2
#####
“ ”
-
.
###

@
###

###

###
-
Bili-
bili

___ #

[ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
from datetime import datetime
```

[ ]:
```
df_mkt = pd.read_excel('Monthly_Market_Value_and_Return.xlsx', index_col=0)
df_mkt.reset_index(inplace=True)
```

```
df_mkt['stock'] = df_mkt['stock'].map(lambda x: str(x).zfill(6))
unique_dates = df_mkt['date'].unique()
unique_dates.sort()
unique_dates_nochange = unique_dates.copy()
unique_dates = [datetime.strptime(date, "%Y-%m") for date in unique_dates]
df_mkt
```

[ ]:

	stock	date	value_in_thousand	monthly_return	next_month_return
0	000001	2005-12	11947347.99	0.051370	0.034202
1	000001	2006-01	12355970.65	0.034202	0.077165
2	000001	2006-02	13309423.50	0.077165	-0.068713
3	000001	2006-03	12394887.09	-0.068713	0.237049
4	000001	2006-04	15333078.53	0.237049	0.114213
…	…	…	…	…	…
419256	605599	2021-11	9636666.94	0.061697	0.086360
419257	605599	2021-12	10468889.19	0.086360	-0.116642
419258	605599	2022-01	9247778.04	-0.116642	0.105971
419259	605599	2022-02	10227778.07	0.105971	-0.170342
419260	605599	2022-03	8485555.80	-0.170342	NaN

[419261 rows x 5 columns]

[ ]:
```
#
df = df_mkt
portfolio_returns = {f'portfolio_{i}': [] for i in range(1, 6)}

for date in unique_dates_nochange:

    df_date = df[df['date'] == date]

    #  df_date
    if df_date.empty:
        #
        continue

    #
    thresholds = np.percentile(df_date['value_in_thousand'], [20, 40, 60, 80,␣
↪100])

    #
    for i in range(5):
        if i == 0:
            #
            portfolio = df_date[df_date['value_in_thousand'] <= thresholds[i]]
        else:
            #
```

```
            portfolio = df_date[(df_date['value_in_thousand'] >␣
 ↪thresholds[i-1]) & (df_date['value_in_thousand'] <= thresholds[i])]

        #
        if portfolio.empty:
            portfolio_return = 0  #
        else:
            #
            if not np.isnan(portfolio['next_month_return'].mean()):
                portfolio_return = portfolio['next_month_return'].mean()

        portfolio_returns[f'portfolio_{i+1}'].append(portfolio_return)
```

[ ]:
```
#
hedge_returns = []
for i in range(len(portfolio_returns['portfolio_1'])):
    hedge_return = portfolio_returns['portfolio_1'][i] -␣
 ↪portfolio_returns['portfolio_5'][i]
    hedge_returns.append(hedge_return)

#
average_returns = {portfolio: np.mean(returns) for portfolio, returns in␣
 ↪portfolio_returns.items()}
average_hedge_return = np.mean(hedge_returns)

#
print("Average Returns by Portfolio:")
for portfolio, average_return in average_returns.items():
    print(f"{portfolio}: {average_return}")

print(f"Average Hedge Return: {average_hedge_return}")
```

```
Average Returns by Portfolio:
portfolio_1: 0.03632338665770842
portfolio_2: 0.0209551217564462
portfolio_3: 0.01726534987882018
portfolio_4: 0.014766130366628655
portfolio_5: 0.012561872521888598
Average Hedge Return: 0.023761514135819815
```

[ ]:
```
# Calculate the cumulative returns for each portfolio
cumulative_portfolio_returns = {portfolio: np.cumsum(returns) for portfolio,␣
 ↪returns in portfolio_returns.items()}
# Calculate the cumulative hedge returns
cumulative_hedge_returns = np.cumsum(hedge_returns)
# Get the final cumulative return for each portfolio and the hedge strategy
```

```
final_cumulative_returns = {portfolio: returns[-1] if len(returns) > 0 else 0␣
 ↪for portfolio, returns in cumulative_portfolio_returns.items()}
final_cumulative_hedge_return = cumulative_hedge_returns[-1] if␣
 ↪len(cumulative_hedge_returns) > 0 else 0
# Print the final cumulative returns by portfolio
print("Final Cumulative Returns by Portfolio:")
for portfolio, final_return in final_cumulative_returns.items():
    print(f"{portfolio}: {final_return}")
# Print the final cumulative hedge return
print(f"Final Cumulative Hedge Return: {final_cumulative_hedge_return}")
```

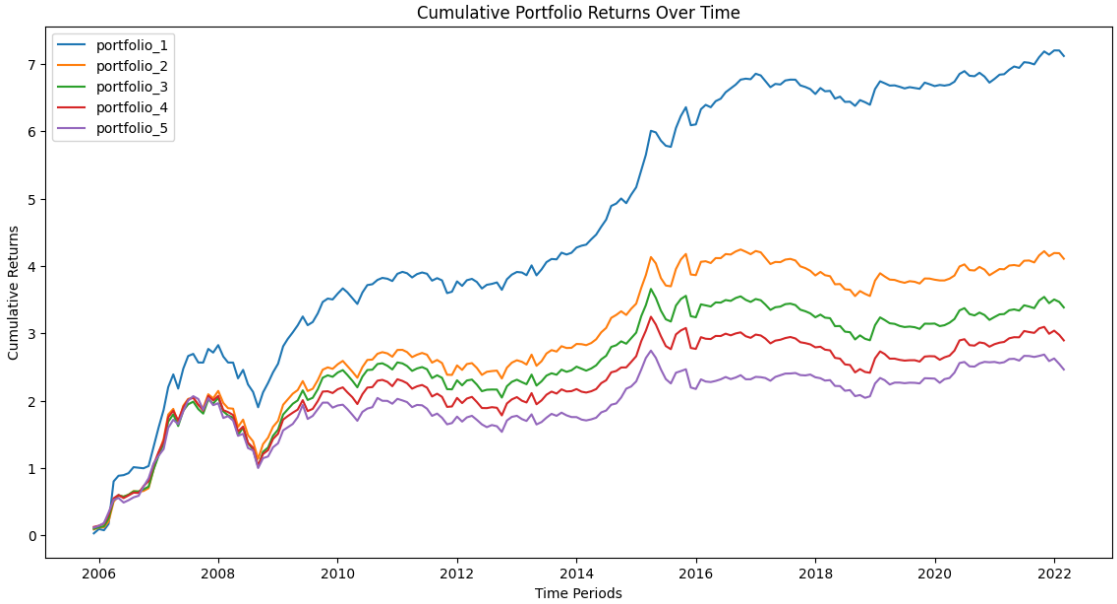
```
Final Cumulative Returns by Portfolio:
portfolio_1: 7.119383784910844
portfolio_2: 4.107203864263453
portfolio_3: 3.3840085762487546
portfolio_4: 2.894161551859217
portfolio_5: 2.4621270142901657
Final Cumulative Hedge Return: 4.657256770620686
```

[ ]:
```
# Calculate the cumulative returns for each portfolio
cumulative_portfolio_returns = {portfolio: np.cumsum(returns) for portfolio,␣
 ↪returns in portfolio_returns.items()}

# Plotting the cumulative returns for each portfolio
plt.figure(figsize=(14, 7))  # Set the figure size

for portfolio, returns in cumulative_portfolio_returns.items():
    plt.plot(unique_dates, returns, label=portfolio)

plt.title('Cumulative Portfolio Returns Over Time')
plt.xlabel('Time Periods')
plt.ylabel('Cumulative Returns')
plt.legend()
plt.show()
```


Cumulative Portfolio Returns Over Time
portfolio_1
portfolio_2
portfolio_3
portfolio_4
portfolio_5
Cumulative Returns
Time Periods
0
1
2
3
4
5
6
7
2006
2008
2010
2012
2014
2016
2018
2020
2022

---

2 ...?
