Assignment II:

Adopting an Open Science Workflow

Jul 11, 2024

Abstract

This project uses the TRR 266 Template for Reproducible Empirical Accounting Research (TREAT) and provides an infrastructure for open science oriented empirical projects. It has been adapted to analyze residual income based on grouped P/B ratios of U.S. public firms using external Worldscope data sets. It showcases a reproducible workflow integrating Python scripts and data analysis. The analysis is performed using a Python script that processes input data files from Wolrdscope and generates output documentation files with the results. This code base, adapted from TREAT, should give you an overview on how the template is supposed to be used for my specific project and how to structure a reproducible empirical project.

1 List of Abbreviations

AMEX: American Stock Exchange

CAPM: Capital Asset Pricing Model

BVE: Book Value of Equity

IDE: Integrated Development Environment

MVE: Market Value of Equity

NYSE: New York Stock Exchange

P/B: Price-to-Book

UK: United Kingdom

US: United States

2 Introduction

The primary objective is to demonstrate a reproducible and collaborative research workflow by calculating and analyzing residual incomes and Price-to-Book (P/B) ratios. The analysis includes filtering relevant data, identifying key financial years, grouping firms based on their P/B ratios, and calculating delfated residual incomes for each firm/year.

Assignment I aims to replicate an empirical table that involves calculating the residual income of US firms over a defined period and examining the relationship between residual income and the Price-to-Book (P/B) ratio. The replication process includes data preparation, cleaning, normalization, and the application of statistical methods to compute and interpret financial metrics using Python. By following a step-by-step approach, I provide insights into the trends of residual earnings across different P/B groups, identical to trends presented by Penman (2013). This document presents the analysis performed in Assignment I, adapted to the open science workflow. Note that intermediate check steps have been omitted for brevity. The full code, including intermediate print steps to verify outputs, is presented in `code/python/Assignment1_analyze_residual_income.py`.

3 Assumptions and Notes

The aim of Assignment I is to replicate a specific empirical table that involves calculating the residual income of US firms over a defined period and examining the relationship between residual income and the Price-to-Book (P/B) ratio. The replication process involves data preparation, cleaning, and normalization, followed by the application of statistical methods to compute and interpret financial metrics. For Assignment I, I used the Python programming language to carry out the empirical analysis. Visual Studio Code was used as the Integrated Development Environment (IDE) for writing, debugging, and optimizing the Python code.

3.1 Assumptions

1. A constant cost of equity is assumed to simplify the calculation of residual income across multiple years, making the analysis more straightforward and easier to interpret. This approach allows for a direct comparison of performance across firms and time periods without the increased complexity due to fluctuating rates. In contrary, estimating a dynamic cost of equity

typically involves financial models such as the Capital Asset Pricing Model (CAPM), which would require additional data like risk-free rates and firm beta values that are not available.

2. The level of cost of equity capital is assumed to be 8% (re = 0.08), a reasonable average estimate for the period 1996-2015 in the US (Goedhart, Koller, and Williams (2002); Damodaran (2024)). This is assumption reflects the return equity investors typically expect given the level of risk during that period.
3. The entire Worldscope panel data provides the Book Value of Equity (BVE), Market Value of Equity (MVE), and net income as values for the end of the respective fiscal year.
4. The beginning of Year 0 data is equivalent to the end of the previous year because the financial data recorded at the end of one year represents the starting values for the following year. Therefore, residual income is deflated by the BVE at the end of the year before Year 0 to ensure comparability across different firms and time periods.
5. It is assumed that Penman (2013) restricted the possible P/B values to a range of 0 to 7 to exclude outliers. Hence, extreme P/B values that are negative and very high (greater than 7) are excluded to focus the analysis on firms with more stable and reasonable valuations, reducing the impact of outliers.

By following the steps provided in Section 4 and adhering to the assumptions made, I successfully replicated the analysis and produced the required table. A thorough step-by-step approach helped to understand and verify the outputs.

4 Replication Steps

4.1 Step 1: Data Loading and Merging

Begin by loading the data from the provided Worldscope files. The static data contains information on unique ISIN IDs for each firm, as well as the corresponding firm name and country. The panel data provides the financial performance numbers of each ISIN. Merge the data on the 'isin' column to include country information in the panel data. This merging step is crucial for filtering the US firms specifically in further steps.

```
import pandas as pd
import numpy as np
import os
import yaml

# Function to get the path of the current file
def get_current_path():
    try:
        return os.path.dirname(os.path.abspath(__file__))
    except NameError:
        return os.getcwd()

# Define the relative path to the configuration file
current_path = get_current_path()
project_root = os.path.abspath(os.path.join(current_path, '..'))
config_path = os.path.join(project_root, 'config', 'config.yaml')

# Load the configuration from YAML file
with open(config_path, 'r') as file:
    config = yaml.safe_load(file)

# Extract paths from the config file
STATIC_DATA_PATH = os.path.join(project_root, 'data', 'external',
                                'wscp_static.txt')
PANEL_DATA_PATH = os.path.join(project_root, 'data', 'external',
                               'wscp_panel.xlsx')

# Load the data from Worldscope files into pandas DataFrames
static_data = pd.read_csv(STATIC_DATA_PATH, delimiter='\t')
```

```
panel_data = pd.read_excel(PANEL_DATA_PATH)


# Normalize column names to avoid issues with column naming conventions
static_data.columns = (
    static_data.columns.str.strip()
    .str.lower()
    .str.replace(' ', '_')
)


panel_data.columns = (
    panel_data.columns.str.strip()
    .str.lower()
    .str.replace(' ', '_')
)


# Merge data on 'isin' to include country information in panel_data
merged_data = panel_data.merge(
    static_data[['isin', 'country']],
    on='isin',
    how='left'
)
```

4.2 Step 2: Filter Data for US Firms and Relevant Years

Filter the merged data to include only US firms (excluding UK and German firms) and the year range 1996-2015. This ensures that further calculations are based on the specific subset of data that matches the assignment's criteria.

```
us_firms = merged_data[merged_data['country'] == 'UNITED STATES']
filtered_data = us_firms[
    (us_firms['year_'] >= 1996) &
```

```
    (us_firms['year_'] <= 2015)
].copy()
```

4.3 Step 3: Identify Year 0 to Year 6 for Each Firm

To calculate the P/B values and residual incomes for each firm/year, it is necessary to identify the base year (Year 0) and the subsequent relative years within the range 0 – 6 in the filtered data, as presented by Penman (2013). Define Year 0 as the earliest year in the dataset for each firm and display it in a new column 'year_0' that holds this value, making 'year_0' the same for each ISIN. Additionally, create the column 'relative_year', represented as the difference between the current year and Year 0. Since a firm may have more than 7 years of observations in the filtered data, limit the data to include only the first seven years (Year 0 to Year 6) for each ISIN to match the assignment's requirements. Notably, the number of observations after merging Step 1 (298,120), filtering Step 2 (142,837), and identifying years in Step 3 (89,584) nearly halves with each step.

```
filtered_data['year_0'] = filtered_data.groupby('isin')['year_'].transform('min')
filtered_data['relative_year'] = filtered_data['year_'] - filtered_data['year_0']
filtered_data = filtered_data[filtered_data['relative_year'] <= 6]
```

4.4 Step 4: Calculate P/B Ratio for Each Firm in Year 0

According to Penman (2013), the residual earnings are calculated after P/B groups are formed in Year 0. Therefore, consider the firms' financial performance data only for Year 0 to calculate the P/B ratio. Replace zero and infinite BVE values with NaN to drop invalid calculations (affecting 28 observations in total). Use the following formula to calculate the P/B ratio:

$$P/B_t = \frac{Market\ value\ of\ equity_t}{Book\ value\ of\ equity_t}$$

```
year_0_data = filtered_data[filtered_data['relative_year'] == 0].copy()


# replace zero and very small BVE values with NaN
```

```
year_0_data['bve'] = year_0_data['bve'].replace([0, np.inf, -np.inf], np.nan)


# Use P/B Ratio formula to calculate it for each firm in Year 0
year_0_data['p/b'] = year_0_data['mve'] / year_0_data['bve']


# Remove rows with NaN P/B ratios
year_0_data = year_0_data.dropna(subset=['p/b'])
```

4.5 Step 5: Form P/B Groups

Sort the firms based on their P/B ratios in descending order and divide them into 20 P/B groups, with the highest P/B ratios in group 1 and the lowest in group 20, as presented by Penman (2013). As a result, each P/B group contains approximately 900 firms, ensuring an equal division of the data set. Calculate the median P/B value for each group and merge the P/B group assignment information back into the filtered data, providing information on which P/B group each firm belongs to in each year.

```
# Sort firms based on their P/B ratios for Year 0 in descending order
num_groups = 20  # define the number of groups
year_0_data = year_0_data.sort_values(
    'p/b', ascending=False
).reset_index(drop=True)


# Assign the groups so that the highest P/B ratios are in group 1
year_0_data['p/b_group'] = pd.qcut(
    year_0_data.index,
    num_groups,
    labels=range(1, num_groups + 1)
)


# Calculate median P/B value for each group
```

```
median_pb_values = year_0_data.groupby('p/b_group', observed=True)['p/b'].median()


# Merge the P/B groups back to the filtered data for Year 0
filtered_data = filtered_data.merge(
    year_0_data[['isin', 'p/b_group']],
    on='isin',
    how='left'
)


# Fill missing P/B groups for years other than Year 0 with same group as Year 0
filtered_data['p/b_group'] = filtered_data.groupby('isin')['p/b_group'].ffill()
```

4.6 Step 6: Calculate Residual Income

This step involves calculating the residual income for each P/B group for Years 0 to 6, followed by deflation by the BVE at the beginning of Year 0. Assume a cost of equity capital of 8%, a reasonable assumption for US firms in the given time range. Based on Penman (2013), the residual income for each company/year should be deflated by the book value of equity at the beginning of Year 0.

To identify whether the financial data from Worldscope contains the numbers as of the beginning or end of the fiscal year, a random sample check was conducted. For example, for ISIN AN8068571086 in its *2020 Annual Report* (2021), the values for total assets, stockholders' equity, net income, and other financial metrics are measured at the end of each fiscal year (Schlumberger Limited 2021, 20). Hence, assume that the entire Worldscope panel data provides the BVE, MVE, and net income as values for the end of the respective fiscal year.

In case a company has observations before 1996, the filtered data set does not enable calculation of residual income in Year 0 (1996) due to missing value of BVE in Year -1, resulting into NaN. To avoid that, create a copy of the merged data from Step 1 to preserve all years of observations. Filter it to US companies and identify the year_minus_1 for each firm. Extract BVE for the year before Year 0 (relative_year = -1) for the deflation step and extract the BVE of the

previous year for each year for the residual income calculation step.

Calculate the non-deflated residual income for each firm/year using the following formula:

$$\text{Residual income}_t = \text{Net income}_t - r_e \times \text{Book value of equity}_{t-1}$$

Assume that the beginning of Year 0 is equivalent to the end of the previous year because the financial data recorded at the end of one year represents the starting values for the following year. According to Penman (2013), residual income is deflated by book value at the beginning of year 0. Hence, for deflating the residual income, use the book value of equity from the year before Year 0 (bve_deflation) for all years (Years 1 to 6). This means that the BVE used for deflation remains constant across all years, normalizing the residual income across different firms and time periods. For example, for Year 0 (1996), use the BVE at the end of 1995 to deflate the residual income.

Calculate the deflated residual income for each firm/year using the following general formula:

$$\text{Deflated residual income}_t = \frac{\text{Residual income}_t}{\text{Book value of equity}_{\text{Year 0 start}}}$$

```
cost_of_equity = 0.08  # assumption for cost of equity capital


# Create a copy of the merged data from Step 1 to keep all years of observations
us_full_data = merged_data[merged_data['country'] == 'UNITED STATES'].copy()


# Identify Year 0 for each firm, ensuring Year 0 is between 1996 and 2015
us_full_data['year_0'] = us_full_data.groupby('isin')['year_'].transform(
    lambda x: x[(x >= 1996) & (x <= 2015)].min()
)
us_full_data = us_full_data[us_full_data['year_0'].notna()]
# Calculate the relative year and year_minus_1
us_full_data['relative_year'] = us_full_data['year_'] - us_full_data['year_0']
```

```
us_full_data['year_minus_1'] = us_full_data['year_0'] - 1


# Convert years to integer
us_full_data['year_0'] = us_full_data['year_0'].astype(int)
us_full_data['relative_year'] = us_full_data['relative_year'].astype(int)
us_full_data['year_minus_1'] = us_full_data['year_minus_1'].astype(int)


# Extract BVE for the year before Year 0 (relative_year = -1) for deflation
bve_deflation = (
    us_full_data[us_full_data['relative_year'] == -1][['isin', 'bve']]
    .rename(columns={'bve': 'bve_deflation'})
)


# Merge the bve_deflation values back to the filtered data
filtered_data = filtered_data.merge(bve_deflation, on='isin', how='left')


# Calculate the BVE of the previous year for each year
us_full_data['bve_prev'] = us_full_data.groupby('isin')['bve'].shift(1)
# Merge bve_prev back to filtered_data
filtered_data = filtered_data.merge(
    us_full_data[['isin', 'year_', 'bve_prev']],
    on=['isin', 'year_'],
    how='left'
)


# Function to calculate non-deflated residual income
def calculate_residual_income(row):
    return row['ninc'] - (cost_of_equity * row['bve_prev'])
# Apply the function to calculate non-deflated residual income for each row
```

```
filtered_data['residual_income'] = filtered_data.apply(
    calculate_residual_income, axis=1
)


# Function to deflate residual income
def deflate_residual_income(row):
    if pd.notna(row['bve_deflation']) and row['bve_deflation'] != 0:
        return row['residual_income'] / row['bve_deflation']
    else:
        return np.nan
# Apply the function to deflate residual income for each row
filtered_data['deflated_residual_income'] = filtered_data.apply(
    deflate_residual_income, axis=1
)
```

4.7 Step 7: Output the Replicated Table

Calculate the median deflated residual income for each P/B group and relative year. Include the median P/B values in the output table. Ensure consistency with Penman (2013) by formatting the results to three decimal places for residual income and two decimal places for P/B values. The resulting table will be saved into an Excel file in the `data/generated` folder of the repository.

The output table certainly indicates a wide range of P/B ratios among the firms, with some extreme values in high and low P/B groups. There is a noticeable trend where firms with very high or very low P/B ratios have less stable and more extreme residual earnings over time, while middle P/B groups seem to show more stable trends as presented by Penman (2013), suggesting a balanced reflection of expectations between market valuation and expected financial performance.

The variation in trends can be explained by the difference in data sets and possible additional assumptions made by Penman (2013) that are not available. For example, the assignment data set includes all USA firms from the Worldscope database, providing a broader sample than

Penman's original study, which included only NYSE and AMEX firms. Furthermore, the assignment analysis focuses on the years 1996 to 2015, a more recent period with different economic conditions compared to the time frame used in Penman's (2013) study.

```
median_residual_income = filtered_data.groupby(
    ['p/b_group', 'relative_year'], observed=True
)['deflated_residual_income'].median().unstack()

# Formatting the results (decimal places)
median_residual_income = median_residual_income.apply(
    lambda col: col.map(
        lambda x: f"{x:.3f}" if pd.notna(x) else "NaN"
    )
)
median_pb_values = median_pb_values.apply(lambda x: f"{x:.2f}")

# Add the median P/B values column to the output table
median_residual_income.insert(0, 'P/B', median_pb_values)

# Display the median residual income table
print(
    "Median Residual Income by P/B Group and Relative Year:\n",
    median_residual_income
)
```

```
Median Residual Income by P/B Group and Relative Year:
 relative_year      P/B       0       1       2       3       4       5       6
p/b_group
1                117.42  -0.792  -0.367   0.000  -0.397  -0.662  -1.357  -0.975
2                 19.23  -0.094  -0.380  -0.179  -0.327   0.084  -0.346  -0.126
3                  9.56   0.063   0.159   0.017   0.108  -0.025  -0.322  -0.202
```

4	6.42	0.109	0.149	0.109	0.106	0.118	-0.040	-0.100
5	4.82	0.134	0.144	0.110	0.109	0.127	0.023	0.003
6	3.81	0.105	0.114	0.064	0.052	0.050	-0.026	-0.025
7	3.15	0.099	0.115	0.077	0.092	0.103	0.032	0.013
8	2.62	0.076	0.075	0.055	0.067	0.084	0.024	0.012
9	2.26	0.074	0.080	0.068	0.072	0.103	0.030	0.031
10	1.96	0.065	0.066	0.055	0.064	0.064	0.017	0.044
11	1.71	0.046	0.048	0.048	0.054	0.053	0.032	0.022
12	1.51	0.035	0.041	0.043	0.039	0.045	0.031	0.026
13	1.31	0.022	0.028	0.033	0.029	0.028	-0.000	0.016
14	1.12	-0.003	0.002	-0.000	-0.005	0.003	-0.033	-0.016
15	0.94	-0.031	-0.021	-0.009	-0.005	-0.002	-0.018	-0.006
16	0.73	-0.052	-0.050	-0.037	-0.039	-0.027	-0.041	-0.052
17	0.17	-0.091	-0.138	-0.054	-0.042	-0.038	-0.030	-0.085
18	-1.92	-0.239	-0.141	-0.148	-0.161	-0.095	-0.059	-0.001
19	-15.57	-1.324	-0.517	-0.354	-0.476	-0.377	-0.611	-0.315
20	-190.52	-1.036	-0.421	-0.218	-0.166	0.304	0.370	-0.076

4.8 Step 8: Modification - Exclude Extreme P/B Values

As indicated in Step 7, the output table certainly includes outliers. To replicate the more stable P/B values observed by Penman (2013), exclude extreme P/B values that are negative or higher than 7. Recalculate the P/B groups based on this filtered data. Merge the updated P/B group information back into the filtered data and repeat the calculation of median deflated residual income for each P/B group and relative year. Format the updated results with the required decimal places and save to an Excel file. The modified output of the replicated table, closely resembling Penman (2013), is presented in Table 2.

The modified table resembles more the one of Penman (2013) by presenting similar trends. In particular, the exclusion of extreme P/B values results in more stable and consistent residual earnings across P/B groups and years. High P/B groups show higher residual earnings initially, which slowly decline over time, and low P/B groups have negative residual earnings that tend to

fluctuate but still remain negative.

```
# Filter out extreme P/B values that are negative and higher than 7
filtered_year_0_data = year_0_data[
    (year_0_data['p/b'] > 0) & (year_0_data['p/b'] <= 7)
]

# Recalculate the P/B groups based on the filtered data
filtered_year_0_data = filtered_year_0_data.sort_values(
    'p/b', ascending=False
).reset_index(drop=True)

filtered_year_0_data['p/b_group'] = pd.qcut(
    filtered_year_0_data.index, num_groups, labels=range(1, num_groups + 1)
)

# Merge the P/B groups back to the filtered data for Year 0
filtered_data = filtered_data.drop(columns=['p/b_group'])
filtered_data = filtered_data.merge(
    filtered_year_0_data[['isin', 'p/b_group']],
    on='isin',
    how='left'
)
filtered_data['p/b_group'] = filtered_data.groupby('isin')['p/b_group'].ffill()

# Calculate median P/B values again
median_pb_values = filtered_year_0_data.groupby(
    'p/b_group',
    observed=True
)['p/b'].median()
```

```
# Calculate the median deflated residual income for each group and year
median_residual_income_filtered = filtered_data.groupby(
    ['p/b_group', 'relative_year'],
    observed=True
)['deflated_residual_income'].median().unstack()

# Formatting the results (decimal places)
median_residual_income_filtered = median_residual_income_filtered.apply(
    lambda col: col.map(lambda x: f"{x:.3f}" if pd.notna(x) else "NaN")
)
median_pb_values = median_pb_values.apply(lambda x: f"{x:.2f}")

# Add the median P/B values column to the output table
median_residual_income_filtered.insert(0, 'P/B', median_pb_values)

# Display the median residual income table
print(
    "Median Residual Income by P/B Group and Relative Year:\n",
    median_residual_income_filtered
)
```

```
Median Residual Income by P/B Group and Relative Year:
 relative_year   P/B       0       1       2       3       4       5       6
p/b_group
1               6.27   0.102   0.147   0.070   0.106   0.073  -0.142  -0.150
2               5.13   0.150   0.172   0.148   0.117   0.070   0.022  -0.011
3               4.31   0.113   0.112   0.074   0.086   0.127  -0.019  -0.043
4               3.73   0.095   0.112   0.070   0.053   0.061  -0.015  -0.001
5               3.29   0.106   0.116   0.070   0.099   0.102  -0.012   0.011
6               2.92   0.097   0.109   0.078   0.084   0.100   0.067   0.031
```

7	2.58	0.075	0.078	0.053	0.053	0.066	0.020	0.009
8	2.33	0.080	0.081	0.066	0.086	0.107	0.055	0.025
9	2.12	0.067	0.071	0.064	0.071	0.061	0.014	0.042
10	1.93	0.064	0.066	0.060	0.061	0.064	0.016	0.053
11	1.77	0.045	0.052	0.041	0.048	0.040	0.016	0.007
12	1.61	0.047	0.048	0.052	0.058	0.074	0.053	0.042
13	1.48	0.029	0.034	0.033	0.033	0.041	0.018	0.016
14	1.35	0.022	0.036	0.042	0.032	0.034	-0.000	0.018
15	1.22	0.019	0.005	0.009	0.012	0.019	-0.012	0.008
16	1.10	-0.007	0.002	0.004	-0.007	-0.003	-0.034	-0.024
17	0.98	-0.025	-0.015	-0.003	0.001	0.008	-0.010	-0.011
18	0.85	-0.038	-0.033	-0.023	-0.021	-0.017	-0.044	-0.016
19	0.68	-0.056	-0.050	-0.044	-0.045	-0.030	-0.035	-0.069
20	0.36	-0.107	-0.146	-0.059	-0.050	-0.053	-0.047	-0.085

5 Conclusion

This project demonstrates the effectiveness of using an open science and collaborative workflow for analyzing residual incomes and Price-to-Book ratios of U.S. public firms. By following a step-by-step approach and using the TRR 266 Template for Reproducible Empirical Accounting Research, I was able to replicate an empirical table from Penman (2013) and provide insights into the trends of residual earnings across different P/B groups, similar to Penman (2013).

The analysis revealed that high P/B groups initially exhibit higher residual earnings, which slowly decline over time, while low P/B groups consistently show negative residual earnings. Excluding extreme P/B values resulted in more stable and consistent residual earnings across groups and years.

In the future, this repository can be cloned or forked (if made public) to kickstart further projects on residual income analysis. Thanks for reading and enjoy!

References

Damodaran, Aswath. 2024. "Cost of Equity and Capital (US)." https://pages.stern.nyu.edu/~adamodar/New_Home_Page/datafile/wacc.html.

Goedhart, Marc H., Tim M. Koller, and Zane D. Williams. 2002. "The Real Cost of Equity." https://www.mckinsey.com/capabilities/strategy-and-corporate-finance/our-insights/the-real-cost-of-equity#.

Penman, Stephen H. 2013. *Financial Statement Analysis and Security Valuation*. 5th ed. New York: McGraw-Hill Irwin.

Schlumberger Limited. 2021. "2020 Annual Report." https://investorcenter.slb.com/static-files/c6adfd3f-5bf4-43bf-9794-04c52433c474.