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.

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

```
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
config_path = os.path.join(get_current_path(), '..', 'config', 'config.yaml')
with open(config_path, 'r') as file:
    config = yaml.safe_load(file)

# Extract paths from the config file
STATIC_DATA_PATH = os.path.join(get_current_path(), '..', config['data_paths']['static_data'])
PANEL_DATA_PATH = os.path.join(get_current_path(), '..', config['data_paths']['panel_data'])

# Check if files exist
if not os.path.exists(STATIC_DATA_PATH):
    raise FileNotFoundError(f"{STATIC_DATA_PATH} does not exist.")
if not os.path.exists(PANEL_DATA_PATH):
    raise FileNotFoundError(f"{PANEL_DATA_PATH} does not exist.")

# Load the data from Worldscope files into pandas DataFrames
static_data = pd.read_csv(STATIC_DATA_PATH, delimiter='\t')  # use '\t' due to tab-separation
panel_data = pd.read_excel(PANEL_DATA_PATH)

# Normalize column names to avoid issues caused by variation in 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')
```

```
# Print the merged data to check. Now we have added a column on countries in panel data
print("Step 1: Merged data sample on 'isin' from static and panel data:\n", merged_data.head(1
print()  # add this to get a blank line for better readability
# To understand how many rows there are in the merged data
num_rows1 = len(merged_data)
print("Number of rows in merged data sample on 'isin' from static and panel data:", num_rows1)
print()
print()
```

```
Step 1: Merged data sample on 'isin' from static and panel data:
             isin  year_           mve           bve          ninc  \
0   AN8068571086   1990  1.378149e+10  3.254816e+09  5.702810e+08
1   AN8068571086   1991  1.496761e+10  3.852886e+09  8.156520e+08
2   AN8068571086   1992  1.384502e+10  4.230806e+09  6.616030e+08
3   AN8068571086   1993  1.439984e+10  4.406340e+09  3.347630e+08
4   AN8068571086   1994  1.220244e+10  4.582954e+09  5.360770e+08
5   AN8068571086   1995  1.683731e+10  4.964017e+09  6.491570e+08
6   AN8068571086   1996  2.462290e+10  5.626380e+09  8.514830e+08
7   AN8068571086   1997  4.009191e+10  6.694924e+09  1.295697e+09
8   AN8068571086   1998  2.532696e+10  8.119075e+09  1.014199e+09
9   AN8068571086   1999  3.176288e+10  7.721028e+09  3.666940e+08
10  AN8068571086   2000  4.578247e+10  8.295216e+09  7.345960e+08

          country
0   UNITED STATES
1   UNITED STATES
2   UNITED STATES
3   UNITED STATES
4   UNITED STATES
5   UNITED STATES
```

```
6   UNITED STATES
7   UNITED STATES
8   UNITED STATES
9   UNITED STATES
10  UNITED STATES
```

```
Number of rows in merged data sample on 'isin' from static and panel data: 298120
```

```
# Filter the merged data for US firms and years 1996-2015
us_firms = merged_data[merged_data['country'] == 'UNITED STATES']  # create a new DataFrame tha
filtered_data = us_firms[(us_firms['year_'] >= 1996) & (us_firms['year_'] <= 2015)].copy()  # 

# Print the filtered data to check
print("Step 2: Filtered data sample on US firms and years 1996-2015:\n", filtered_data.head(22)
print()

# To understand how many rows were dropped out
num_rows2 = len(filtered_data)
print("Number of rows in filtered data sample on US firms and years 1996-2015:", num_rows2)
print()
print()
```

```
Step 2: Filtered data sample on US firms and years 1996-2015:
             isin  year_           mve           bve          ninc  \
6   AN8068571086   1996  2.462290e+10  5.626380e+09  8.514830e+08
7   AN8068571086   1997  4.009191e+10  6.694924e+09  1.295697e+09
8   AN8068571086   1998  2.532696e+10  8.119075e+09  1.014199e+09
9   AN8068571086   1999  3.176288e+10  7.721028e+09  3.666940e+08
10  AN8068571086   2000  4.578247e+10  8.295216e+09  7.345960e+08
11  AN8068571086   2001  3.164518e+10  8.378000e+09  5.220000e+08
```

12	AN8068571086	2002	2.450367e+10	5.606138e+09	-2.320000e+09
13	AN8068571086	2003	3.206309e+10	5.881000e+09	3.830020e+08
14	AN8068571086	2004	3.940019e+10	6.116737e+09	1.224000e+09
15	AN8068571086	2005	5.720212e+10	7.592000e+09	2.207000e+09
16	AN8068571086	2006	7.439575e+10	1.042000e+10	3.709851e+09
17	AN8068571086	2007	1.176128e+11	1.487589e+10	5.177000e+09
18	AN8068571086	2008	5.054629e+10	1.686237e+10	5.434801e+09
19	AN8068571086	2009	7.777037e+10	1.912000e+10	3.134000e+09
20	AN8068571086	2010	1.136578e+11	3.122600e+10	4.267000e+09
21	AN8068571086	2011	9.111020e+10	3.126300e+10	4.997000e+09
22	AN8068571086	2012	9.204627e+10	3.475100e+10	5.490000e+09
23	AN8068571086	2013	1.178035e+11	3.946900e+10	6.732000e+09
24	AN8068571086	2014	1.089244e+11	3.785000e+10	5.438000e+09
25	AN8068571086	2015	8.760600e+10	3.563300e+10	2.072000e+09
34	AU000000AVH4	1999	2.862579e+07	3.040225e+06	-8.107832e+05
35	AU000000AVH4	2000	4.715262e+06	5.227754e+06	-1.539973e+06

	country
6	UNITED STATES
7	UNITED STATES
8	UNITED STATES
9	UNITED STATES
10	UNITED STATES
11	UNITED STATES
12	UNITED STATES
13	UNITED STATES
14	UNITED STATES
15	UNITED STATES
16	UNITED STATES
17	UNITED STATES

```
18  UNITED STATES
19  UNITED STATES
20  UNITED STATES
21  UNITED STATES
22  UNITED STATES
23  UNITED STATES
24  UNITED STATES
25  UNITED STATES
34  UNITED STATES
35  UNITED STATES
```

```
Number of rows in filtered data sample on US firms and years 1996-2015: 142837
```

```
# Identify Year 0 to Year 6 for each firm within the filtered data
filtered_data['year_0'] = filtered_data.groupby('isin')['year_'].transform('min')  # this colu
filtered_data['relative_year'] = filtered_data['year_'] - filtered_data['year_0']  # this colu
filtered_data = filtered_data[filtered_data['relative_year'] <= 6]  # filter the data to inclu

# Print the filtered data to check if it's empty or has data
print("Step 3: Filtered data sample after identifying the Year 0 and relative years:\n", filte
print()

# To understand how many rows were dropped out
num_rows3 = len(filtered_data)
print("Number of rows in filtered data sample after identifying the Year 0 and relative years:
print()
print()
```

```
Step 3: Filtered data sample after identifying the Year 0 and relative years:
            isin  year_          mve          bve         ninc  \
```

6	AN8068571086	1996	2.462290e+10	5.626380e+09	8.514830e+08
7	AN8068571086	1997	4.009191e+10	6.694924e+09	1.295697e+09
8	AN8068571086	1998	2.532696e+10	8.119075e+09	1.014199e+09
9	AN8068571086	1999	3.176288e+10	7.721028e+09	3.666940e+08
10	AN8068571086	2000	4.578247e+10	8.295216e+09	7.345960e+08
11	AN8068571086	2001	3.164518e+10	8.378000e+09	5.220000e+08
12	AN8068571086	2002	2.450367e+10	5.606138e+09	-2.320000e+09
34	AU000000AVH4	1999	2.862579e+07	3.040225e+06	-8.107832e+05
35	AU000000AVH4	2000	4.715262e+06	5.227754e+06	-1.539973e+06
36	AU000000AVH4	2001	1.203144e+07	4.548271e+06	-1.121748e+06
37	AU000000AVH4	2002	7.939532e+06	8.251444e+06	-1.544901e+06
38	AU000000AVH4	2003	1.133298e+07	8.553178e+06	-2.672489e+06
39	AU000000AVH4	2004	5.571049e+07	8.659432e+06	-4.985310e+06
40	AU000000AVH4	2005	5.255392e+07	1.307238e+07	-7.182573e+06
55	AU000000AXP3	2015	5.356347e+07	1.704517e+07	-1.088878e+07
68	AU000000BIQ0	2015	6.256718e+07	1.745620e+07	-3.838337e+06
74	AU000000BLT8	1998	1.292723e+06	1.033445e+06	-5.250816e+05
75	AU000000BLT8	1999	3.426664e+06	1.034165e+06	-1.383048e+06
76	AU000000BLT8	2000	4.161143e+06	2.744636e+05	-6.646792e+05
77	AU000000BLT8	2001	1.653097e+06	5.073972e+05	-7.855979e+05
78	AU000000BLT8	2002	7.447966e+06	8.525630e+06	-6.332391e+05
79	AU000000BLT8	2003	2.993219e+07	4.651587e+06	-5.545786e+06

	country	year_0	relative_year
6	UNITED STATES	1996	0
7	UNITED STATES	1996	1
8	UNITED STATES	1996	2
9	UNITED STATES	1996	3
10	UNITED STATES	1996	4
11	UNITED STATES	1996	5

```
12  UNITED STATES    1996             6
34  UNITED STATES    1999             0
35  UNITED STATES    1999             1
36  UNITED STATES    1999             2
37  UNITED STATES    1999             3
38  UNITED STATES    1999             4
39  UNITED STATES    1999             5
40  UNITED STATES    1999             6
55  UNITED STATES    2015             0
68  UNITED STATES    2015             0
74  UNITED STATES    1998             0
75  UNITED STATES    1998             1
76  UNITED STATES    1998             2
77  UNITED STATES    1998             3
78  UNITED STATES    1998             4
79  UNITED STATES    1998             5

Number of rows in filtered data sample after identifying the Year 0 and relative years: 89584
```

```
# Calculate P/B Ratio for each firm in Year 0

year_0_data = filtered_data[filtered_data['relative_year'] == 0].copy()  # create a copy of fil
print("Step 4: Year 0 data:\n", year_0_data)
print()

year_0_data['bve'] = year_0_data['bve'].replace([0, np.inf, -np.inf], np.nan)  # replace zero a
# To verify: display rows where 'bve' is NaN
nan_bve_rows = year_0_data[year_0_data['bve'].isna()]
print("Rows where 'bve' has been replaced with NaN:\n", nan_bve_rows)
```

```
num_nan_bve_rows = len(nan_bve_rows)
print("Number of rows where 'bve' has been replaced with NaN:", num_nan_bve_rows)
print()

# 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']

# Before removing NaN P/B ratios
num_rows_before = len(year_0_data)
# Remove rows with NaN P/B ratios
year_0_data = year_0_data.dropna(subset=['p/b'])
# After removing NaN P/B ratios
num_rows_after = len(year_0_data)

# Calculate the number of dropped observations
num_dropped = num_rows_before - num_rows_after
print("Number of observations dropped due to NaN P/B ratios:", num_dropped)  # checked: the nu
print("Number of rows in Year 0 data after removing NaN P/B ratios:", num_rows_after)
print("Year 0 data after calculating P/B ratio and dropping NaN values:\n", year_0_data)  # th
print()
print()
```

```
Step 4: Year 0 data:
                 isin  year_           mve           bve          ninc  \
6       AN8068571086   1996  2.462290e+10  5.626380e+09  8.514830e+08
34      AU000000AVH4   1999  2.862579e+07  3.040225e+06 -8.107832e+05
55      AU000000AXP3   2015  5.356347e+07  1.704517e+07 -1.088878e+07
68      AU000000BIQ0   2015  6.256718e+07  1.745620e+07 -3.838337e+06
74      AU000000BLT8   1998  1.292723e+06  1.033445e+06 -5.250816e+05
...              ...    ...           ...           ...           ...
```

```
298067  VGG7982P1045   2008  5.940000e+07  3.418515e+07 -5.978900e+04
298069  VGG7996N1298   2001  6.453097e+06  2.711511e+06 -7.451337e+05
298085  VGG872101032   2000  5.169297e+08  1.874880e+08  1.845300e+07
298103  VGG9220E1079   2008  9.737100e+05  2.103214e+05 -7.026973e+04
298107  VGG934461051   2010  2.709000e+07  1.715800e+07 -5.290000e+05

              country  year_0  relative_year
6       UNITED STATES    1996              0
34      UNITED STATES    1999              0
55      UNITED STATES    2015              0
68      UNITED STATES    2015              0
74      UNITED STATES    1998              0
...               ...     ...            ...
298067  UNITED STATES    2008              0
298069  UNITED STATES    2001              0
298085  UNITED STATES    2000              0
298103  UNITED STATES    2008              0
298107  UNITED STATES    2010              0

[18017 rows x 8 columns]

Rows where 'bve' has been replaced with NaN:
                isin  year_           mve  bve       ninc        country  \
99340   US03170T1043   2008  1.350000e+07  NaN   -44942.0  UNITED STATES
100216  US03462P1093   2001  2.643200e+06  NaN    -5494.0  UNITED STATES
110965  US0746812066   2008  7.638206e+05  NaN   -28262.0  UNITED STATES
113574  US0918442098   2014  6.120000e+07  NaN   -46408.0  UNITED STATES
122820  US1398931014   2007  1.992000e+06  NaN     7242.0  UNITED STATES
129126  US1713342046   2008  3.531343e+06  NaN -1361848.0  UNITED STATES
137049  US2202611016   2000  8.056976e+07  NaN  -110923.0  UNITED STATES
```

138103	US22527R1086	2001	2.642400e+06	NaN	-37354.0	UNITED STATES
148675	US2737711059	2009	7.229743e+05	NaN	-40025.0	UNITED STATES
152476	US29271J1097	1996	4.303131e+05	NaN	0.0	UNITED STATES
154134	US2942621002	1998	2.310000e+06	NaN	0.0	UNITED STATES
172893	US3913141012	2010	4.124010e+06	NaN	-68.0	UNITED STATES
182252	US4492362078	2003	4.603654e+06	NaN	-6414103.0	UNITED STATES
183261	US45104N1090	2000	5.625000e+07	NaN	0.0	UNITED STATES
183442	US45166V2051	2005	1.606482e+06	NaN	-221000.0	UNITED STATES
216130	US6284511069	1998	3.874910e+05	NaN	-35552.0	UNITED STATES
220856	US64949W1099	1999	3.850000e+07	NaN	-185.0	UNITED STATES
241413	US74338Q1004	1999	5.999500e+06	NaN	-136900.0	UNITED STATES
246464	US75584Q1085	2001	3.831104e+06	NaN	0.0	UNITED STATES
246885	US7574632037	2003	1.560000e+06	NaN	107151.0	UNITED STATES
247256	US75886Y1064	2000	1.743153e+05	NaN	-87342.0	UNITED STATES
253359	US78646R1068	2000	4.746662e+05	NaN	-5001.0	UNITED STATES
256163	US81213Y1082	2001	1.454694e+05	NaN	0.0	UNITED STATES
267656	US86681U2078	2000	2.151544e+07	NaN	-13470.0	UNITED STATES
268591	US86866P1066	1999	3.033540e+05	NaN	1201414.0	UNITED STATES
272706	US87944M1071	1998	1.066176e+05	NaN	-10416.0	UNITED STATES
274233	US88337U1025	2001	5.046598e+05	NaN	0.0	UNITED STATES
277598	US89421L1070	2001	8.858950e+06	NaN	-16715.0	UNITED STATES

	year_0	relative_year
99340	2008	0
100216	2001	0
110965	2008	0
113574	2014	0
122820	2007	0
129126	2008	0
137049	2000	0

```
138103    2001             0
148675    2009             0
152476    1996             0
154134    1998             0
172893    2010             0
182252    2003             0
183261    2000             0
183442    2005             0
216130    1998             0
220856    1999             0
241413    1999             0
246464    2001             0
246885    2003             0
247256    2000             0
253359    2000             0
256163    2001             0
267656    2000             0
268591    1999             0
272706    1998             0
274233    2001             0
277598    2001             0
Number of rows where 'bve' has been replaced with NaN: 28


Number of observations dropped due to NaN P/B ratios: 28
Number of rows in Year 0 data after removing NaN P/B ratios: 17989
Year 0 data after calculating P/B ratio and dropping NaN values:
                 isin  year_           mve           bve          ninc  \
6       AN8068571086   1996  2.462290e+10  5.626380e+09  8.514830e+08
34      AU000000AVH4   1999  2.862579e+07  3.040225e+06 -8.107832e+05
55      AU000000AXP3   2015  5.356347e+07  1.704517e+07 -1.088878e+07
```

```
68      AU000000BIQ0   2015  6.256718e+07  1.745620e+07 -3.838337e+06
74      AU000000BLT8   1998  1.292723e+06  1.033445e+06 -5.250816e+05
...              ...    ...           ...           ...           ...
298067  VGG7982P1045   2008  5.940000e+07  3.418515e+07 -5.978900e+04
298069  VGG7996N1298   2001  6.453097e+06  2.711511e+06 -7.451337e+05
298085  VGG872101032   2000  5.169297e+08  1.874880e+08  1.845300e+07
298103  VGG9220E1079   2008  9.737100e+05  2.103214e+05 -7.026973e+04
298107  VGG934461051   2010  2.709000e+07  1.715800e+07 -5.290000e+05

               country  year_0  relative_year       p/b
6       UNITED STATES    1996              0  4.376331
34      UNITED STATES    1999              0  9.415682
55      UNITED STATES    2015              0  3.142443
68      UNITED STATES    2015              0  3.584238
74      UNITED STATES    1998              0  1.250887
...               ...     ...            ...       ...
298067  UNITED STATES    2008              0  1.737597
298069  UNITED STATES    2001              0  2.379889
298085  UNITED STATES    2000              0  2.757135
298103  UNITED STATES    2008              0  4.629630
298107  UNITED STATES    2010              0  1.578855

[17989 rows x 9 columns]
```

```python
# Form P/B groups


# 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)  # sort tl
```

```
print("Step 5: Year 0 data sorted in descending order based on the P/B ratios:\n", year_0_data)
print()

# Assign the groups so that the highest P/B ratios are in group 1 and lowest in group 20
year_0_data['p/b_group'] = pd.qcut(year_0_data.index, num_groups, labels=range(1, num_groups +
print("Year 0 data updated with P/B groups:\n", year_0_data)  # checked: p/b ratios grouping al

# Count the number of firms in each P/B group before merging
pb_group_counts_pre_merge = year_0_data.groupby('p/b_group', observed=True)['isin'].nunique()
print("Number of firms assigned to each P/B group before merging:\n", pb_group_counts_pre_merge
print()

# Calculate median P/B value for each group
median_pb_values = year_0_data.groupby('p/b_group', observed=True)['p/b'].median()
# Display the median P/B values
print("Median P/B values for each group:\n", median_pb_values)
print()

# 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 the same group as Year 0
filtered_data['p/b_group'] = filtered_data.groupby('isin')['p/b_group'].ffill()

# Display the combined filtered data after merging P/B groups
print("Filtered data after merging P/B groups:\n", filtered_data.head(22))  # the DataFrame now
# Count the number of firms in each P/B group after merging
pb_group_counts_post_merge = filtered_data.groupby('p/b_group', observed=True)['isin'].nunique
print("Number of firms assigned to each P/B group after merging:\n", pb_group_counts_post_merge
# Calculate the number of rows in the merged filtered data sample
```

```
num_rows_merged_filtered = len(filtered_data)
print("Number of rows in the merged filtered data sample:", num_rows_merged_filtered)  # checke
print()
```

```
Step 5: Year 0 data sorted in descending order based on the P/B ratios:
                isin  year_           mve      bve      ninc        country  \
0      US1566611000   2006  1.674397e+08     67.0  -51368.0  UNITED STATES
1      US26850R1068   1999  2.559492e+08    122.0   -3578.0  UNITED STATES
2      US9180992011   2009  4.546560e+07    141.0  -20828.0  UNITED STATES
3      US05501N2018   2013  1.601475e+08   1728.0  -29925.0  UNITED STATES
4      US23246Y1010   1997  5.304422e+06     58.0 -345688.0  UNITED STATES
...             ...    ...           ...      ...       ...            ...
17984  US75509Q1004   2011  1.853700e+08   -468.0  -17502.0  UNITED STATES
17985  US21872T1079   2014  5.346250e+07   -108.0  -64522.0  UNITED STATES
17986  US63254D1028   2015  1.038770e+07    -10.0  -31831.0  UNITED STATES
17987  US6625463087   2011  2.803760e+10 -16070.0  -47071.0  UNITED STATES
17988  US75081R1041   2013  1.468056e+09   -768.0 -243872.0  UNITED STATES

       year_0  relative_year           p/b
0        2006              0  2.499100e+06
1        1999              0  2.097945e+06
2        2009              0  3.224511e+05
3        2013              0  9.267795e+04
4        1997              0  9.145554e+04
...       ...            ...           ...
17984    2011              0 -3.960897e+05
17985    2014              0 -4.950231e+05
17986    2015              0 -1.038770e+06
17987    2011              0 -1.744717e+06
17988    2013              0 -1.911532e+06
```

```
[17989 rows x 9 columns]


Year 0 data updated with P/B groups:
                isin  year_           mve      bve      ninc        country  \
0      US1566611000   2006  1.674397e+08     67.0  -51368.0  UNITED STATES
1      US26850R1068   1999  2.559492e+08    122.0   -3578.0  UNITED STATES
2      US9180992011   2009  4.546560e+07    141.0  -20828.0  UNITED STATES
3      US05501N2018   2013  1.601475e+08   1728.0  -29925.0  UNITED STATES
4      US23246Y1010   1997  5.304422e+06     58.0 -345688.0  UNITED STATES
...             ...    ...           ...      ...       ...            ...
17984  US75509Q1004   2011  1.853700e+08   -468.0  -17502.0  UNITED STATES
17985  US21872T1079   2014  5.346250e+07   -108.0  -64522.0  UNITED STATES
17986  US63254D1028   2015  1.038770e+07    -10.0  -31831.0  UNITED STATES
17987  US6625463087   2011  2.803760e+10 -16070.0  -47071.0  UNITED STATES
17988  US75081R1041   2013  1.468056e+09   -768.0 -243872.0  UNITED STATES

       year_0  relative_year           p/b p/b_group
0        2006              0  2.499100e+06         1
1        1999              0  2.097945e+06         1
2        2009              0  3.224511e+05         1
3        2013              0  9.267795e+04         1
4        1997              0  9.145554e+04         1
...       ...            ...           ...       ...
17984    2011              0 -3.960897e+05        20
17985    2014              0 -4.950231e+05        20
17986    2015              0 -1.038770e+06        20
17987    2011              0 -1.744717e+06        20
17988    2013              0 -1.911532e+06        20
```

```
[17989 rows x 10 columns]
Number of firms assigned to each P/B group before merging:
 p/b_group
1     900
2     899
3     900
4     899
5     900
6     899
7     899
8     900
9     899
10    900
11    899
12    899
13    900
14    899
15    900
16    899
17    899
18    900
19    899
20    900
Name: isin, dtype: int64


Median P/B values for each group:
 p/b_group
1     117.418412
2      19.233211
3       9.562627
```

```
4       6.423647
5       4.817770
6       3.813771
7       3.146504
8       2.622431
9       2.255640
10      1.955162
11      1.714173
12      1.505500
13      1.312785
14      1.121658
15      0.941750
16      0.725272
17      0.172427
18     -1.918757
19    -15.565863
20   -190.516733
Name: p/b, dtype: float64


Filtered data after merging P/B groups:
             isin  year_           mve           bve          ninc  \
0   AN8068571086   1996  2.462290e+10  5.626380e+09  8.514830e+08
1   AN8068571086   1997  4.009191e+10  6.694924e+09  1.295697e+09
2   AN8068571086   1998  2.532696e+10  8.119075e+09  1.014199e+09
3   AN8068571086   1999  3.176288e+10  7.721028e+09  3.666940e+08
4   AN8068571086   2000  4.578247e+10  8.295216e+09  7.345960e+08
5   AN8068571086   2001  3.164518e+10  8.378000e+09  5.220000e+08
6   AN8068571086   2002  2.450367e+10  5.606138e+09 -2.320000e+09
7   AU000000AVH4   1999  2.862579e+07  3.040225e+06 -8.107832e+05
8   AU000000AVH4   2000  4.715262e+06  5.227754e+06 -1.539973e+06
```

9	AU000000AVH4	2001	1.203144e+07	4.548271e+06	-1.121748e+06
10	AU000000AVH4	2002	7.939532e+06	8.251444e+06	-1.544901e+06
11	AU000000AVH4	2003	1.133298e+07	8.553178e+06	-2.672489e+06
12	AU000000AVH4	2004	5.571049e+07	8.659432e+06	-4.985310e+06
13	AU000000AVH4	2005	5.255392e+07	1.307238e+07	-7.182573e+06
14	AU000000AXP3	2015	5.356347e+07	1.704517e+07	-1.088878e+07
15	AU000000BIQ0	2015	6.256718e+07	1.745620e+07	-3.838337e+06
16	AU000000BLT8	1998	1.292723e+06	1.033445e+06	-5.250816e+05
17	AU000000BLT8	1999	3.426664e+06	1.034165e+06	-1.383048e+06
18	AU000000BLT8	2000	4.161143e+06	2.744636e+05	-6.646792e+05
19	AU000000BLT8	2001	1.653097e+06	5.073972e+05	-7.855979e+05
20	AU000000BLT8	2002	7.447966e+06	8.525630e+06	-6.332391e+05
21	AU000000BLT8	2003	2.993219e+07	4.651587e+06	-5.545786e+06

	country	year_0	relative_year	p/b_group
0	UNITED STATES	1996	0	5
1	UNITED STATES	1996	1	5
2	UNITED STATES	1996	2	5
3	UNITED STATES	1996	3	5
4	UNITED STATES	1996	4	5
5	UNITED STATES	1996	5	5
6	UNITED STATES	1996	6	5
7	UNITED STATES	1999	0	3
8	UNITED STATES	1999	1	3
9	UNITED STATES	1999	2	3
10	UNITED STATES	1999	3	3
11	UNITED STATES	1999	4	3
12	UNITED STATES	1999	5	3
13	UNITED STATES	1999	6	3
14	UNITED STATES	2015	0	7

```
15  UNITED STATES    2015              0         6
16  UNITED STATES    1998              0        13
17  UNITED STATES    1998              1        13
18  UNITED STATES    1998              2        13
19  UNITED STATES    1998              3        13
20  UNITED STATES    1998              4        13
21  UNITED STATES    1998              5        13
Number of firms assigned to each P/B group after merging:
 p/b_group
1     900
2     899
3     900
4     899
5     900
6     899
7     899
8     900
9     899
10    900
11    899
12    899
13    900
14    899
15    900
16    899
17    899
18    900
19    899
20    900
Name: isin, dtype: int64
```

```
Number of rows in the merged filtered data sample: 89584
```

```
# Calculate residual income for each P/B group for Years 0 to 6, deflated by the book value of

cost_of_equity = 0.08  # assumption for cost of equity capital

# Create a copy of the merged data from Step 1 to preserve all years of observations
us_full_data = merged_data[merged_data['country'] == 'UNITED STATES'].copy()
print("Step 6: Merged data (Step 1) filtered to US only, all years preserved:\n", us_full_data
print()

# 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 >= 199
us_full_data = us_full_data[us_full_data['year_0'].notna()]  # remove rows where year_0 is NaN
# 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)
# Print the data to verify the steps
print("Data after identifying Year 0, relative year, and Year -1:\n", us_full_data.head(32))
print()

# 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(colu
print(bve_deflation)
# Merge the bve_deflation values back to the filtered data
```

```
filtered_data = filtered_data.merge(bve_deflation, on='isin', how='left')
print("Filtered data with bve_deflation:\n",filtered_data)  # new column on bve_deflation
print()

# Calculate the BVE of the previous year for each year (for residual income calculation)
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', 'y
print("Filtered data with bve_prev:\n",filtered_data)  # new column on bve_prev
print()

# 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)
# Display the 'residual_income' values
print("Residual income values:\n", filtered_data)  # new column on residual_income
print()

# 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=
print("Deflated residual income values:\n", filtered_data)  # new column on deflated_residual_
```

```
print()
```

```
Step 6: Merged data (Step 1) filtered to US only, all years preserved:
```

	isin	year_	mve	bve	ninc	\
0	AN8068571086	1990	1.378149e+10	3.254816e+09	5.702810e+08	
1	AN8068571086	1991	1.496761e+10	3.852886e+09	8.156520e+08	
2	AN8068571086	1992	1.384502e+10	4.230806e+09	6.616030e+08	
3	AN8068571086	1993	1.439984e+10	4.406340e+09	3.347630e+08	
4	AN8068571086	1994	1.220244e+10	4.582954e+09	5.360770e+08	
5	AN8068571086	1995	1.683731e+10	4.964017e+09	6.491570e+08	
6	AN8068571086	1996	2.462290e+10	5.626380e+09	8.514830e+08	
7	AN8068571086	1997	4.009191e+10	6.694924e+09	1.295697e+09	
8	AN8068571086	1998	2.532696e+10	8.119075e+09	1.014199e+09	
9	AN8068571086	1999	3.176288e+10	7.721028e+09	3.666940e+08	
10	AN8068571086	2000	4.578247e+10	8.295216e+09	7.345960e+08	
11	AN8068571086	2001	3.164518e+10	8.378000e+09	5.220000e+08	
12	AN8068571086	2002	2.450367e+10	5.606138e+09	-2.320000e+09	
13	AN8068571086	2003	3.206309e+10	5.881000e+09	3.830020e+08	
14	AN8068571086	2004	3.940019e+10	6.116737e+09	1.224000e+09	
15	AN8068571086	2005	5.720212e+10	7.592000e+09	2.207000e+09	
16	AN8068571086	2006	7.439575e+10	1.042000e+10	3.709851e+09	
17	AN8068571086	2007	1.176128e+11	1.487589e+10	5.177000e+09	
18	AN8068571086	2008	5.054629e+10	1.686237e+10	5.434801e+09	
19	AN8068571086	2009	7.777037e+10	1.912000e+10	3.134000e+09	
20	AN8068571086	2010	1.136578e+11	3.122600e+10	4.267000e+09	
21	AN8068571086	2011	9.111020e+10	3.126300e+10	4.997000e+09	

	country
0	UNITED STATES
1	UNITED STATES

```
2   UNITED STATES
3   UNITED STATES
4   UNITED STATES
5   UNITED STATES
6   UNITED STATES
7   UNITED STATES
8   UNITED STATES
9   UNITED STATES
10  UNITED STATES
11  UNITED STATES
12  UNITED STATES
13  UNITED STATES
14  UNITED STATES
15  UNITED STATES
16  UNITED STATES
17  UNITED STATES
18  UNITED STATES
19  UNITED STATES
20  UNITED STATES
21  UNITED STATES


Data after identifying Year 0, relative year, and Year -1:
             isin  year_           mve           bve          ninc  \
0   AN8068571086   1990  1.378149e+10  3.254816e+09  5.702810e+08
1   AN8068571086   1991  1.496761e+10  3.852886e+09  8.156520e+08
2   AN8068571086   1992  1.384502e+10  4.230806e+09  6.616030e+08
3   AN8068571086   1993  1.439984e+10  4.406340e+09  3.347630e+08
4   AN8068571086   1994  1.220244e+10  4.582954e+09  5.360770e+08
5   AN8068571086   1995  1.683731e+10  4.964017e+09  6.491570e+08
6   AN8068571086   1996  2.462290e+10  5.626380e+09  8.514830e+08
```

7	AN8068571086	1997	4.009191e+10	6.694924e+09	1.295697e+09
8	AN8068571086	1998	2.532696e+10	8.119075e+09	1.014199e+09
9	AN8068571086	1999	3.176288e+10	7.721028e+09	3.666940e+08
10	AN8068571086	2000	4.578247e+10	8.295216e+09	7.345960e+08
11	AN8068571086	2001	3.164518e+10	8.378000e+09	5.220000e+08
12	AN8068571086	2002	2.450367e+10	5.606138e+09	-2.320000e+09
13	AN8068571086	2003	3.206309e+10	5.881000e+09	3.830020e+08
14	AN8068571086	2004	3.940019e+10	6.116737e+09	1.224000e+09
15	AN8068571086	2005	5.720212e+10	7.592000e+09	2.207000e+09
16	AN8068571086	2006	7.439575e+10	1.042000e+10	3.709851e+09
17	AN8068571086	2007	1.176128e+11	1.487589e+10	5.177000e+09
18	AN8068571086	2008	5.054629e+10	1.686237e+10	5.434801e+09
19	AN8068571086	2009	7.777037e+10	1.912000e+10	3.134000e+09
20	AN8068571086	2010	1.136578e+11	3.122600e+10	4.267000e+09
21	AN8068571086	2011	9.111020e+10	3.126300e+10	4.997000e+09
22	AN8068571086	2012	9.204627e+10	3.475100e+10	5.490000e+09
23	AN8068571086	2013	1.178035e+11	3.946900e+10	6.732000e+09
24	AN8068571086	2014	1.089244e+11	3.785000e+10	5.438000e+09
25	AN8068571086	2015	8.760600e+10	3.563300e+10	2.072000e+09
26	AN8068571086	2016	1.167744e+11	4.107800e+10	-1.687000e+09
27	AN8068571086	2017	9.326776e+10	3.684200e+10	-1.505000e+09
28	AN8068571086	2018	4.989864e+10	3.616200e+10	2.138000e+09
29	AN8068571086	2019	5.567700e+10	2.376000e+10	-1.013700e+10
30	AN8068571086	2020	3.038802e+10	1.207100e+10	-1.051800e+10
31	AN8068571086	2021	4.203128e+10	1.500400e+10	1.881000e+09

	country	year_0	relative_year	year_minus_1
0	UNITED STATES	1996	-6	1995
1	UNITED STATES	1996	-5	1995
2	UNITED STATES	1996	-4	1995

3	UNITED STATES	1996	-3	1995
4	UNITED STATES	1996	-2	1995
5	UNITED STATES	1996	-1	1995
6	UNITED STATES	1996	0	1995
7	UNITED STATES	1996	1	1995
8	UNITED STATES	1996	2	1995
9	UNITED STATES	1996	3	1995
10	UNITED STATES	1996	4	1995
11	UNITED STATES	1996	5	1995
12	UNITED STATES	1996	6	1995
13	UNITED STATES	1996	7	1995
14	UNITED STATES	1996	8	1995
15	UNITED STATES	1996	9	1995
16	UNITED STATES	1996	10	1995
17	UNITED STATES	1996	11	1995
18	UNITED STATES	1996	12	1995
19	UNITED STATES	1996	13	1995
20	UNITED STATES	1996	14	1995
21	UNITED STATES	1996	15	1995
22	UNITED STATES	1996	16	1995
23	UNITED STATES	1996	17	1995
24	UNITED STATES	1996	18	1995
25	UNITED STATES	1996	19	1995
26	UNITED STATES	1996	20	1995
27	UNITED STATES	1996	21	1995
28	UNITED STATES	1996	22	1995
29	UNITED STATES	1996	23	1995
30	UNITED STATES	1996	24	1995
31	UNITED STATES	1996	25	1995

```
                isin  bve_deflation
5       AN8068571086   4.964017e+09
417     BM6283511085   1.620457e+08
428     BM8113717072   2.319710e+08
604     BMG4776G1015   1.795500e+09
1176    CA10382K1021   4.637073e+07
...              ...            ...
297497  US9898241073   2.189300e+08
297531  US9898551018   2.233300e+07
297572  US9898671063   2.136700e+07
297620  US9899171097   7.264900e+07
297624  US9899221090   2.959000e+07

[5602 rows x 2 columns]
Filtered data with bve_deflation:
                isin  year_           mve           bve          ninc  \
0       AN8068571086   1996  2.462290e+10  5.626380e+09  8.514830e+08
1       AN8068571086   1997  4.009191e+10  6.694924e+09  1.295697e+09
2       AN8068571086   1998  2.532696e+10  8.119075e+09  1.014199e+09
3       AN8068571086   1999  3.176288e+10  7.721028e+09  3.666940e+08
4       AN8068571086   2000  4.578247e+10  8.295216e+09  7.345960e+08
...              ...    ...           ...           ...           ...
89579   VGG9220E1079   2011  3.369199e+07  2.382768e+07 -9.825526e+06
89580   VGG934461051   2010  2.709000e+07  1.715800e+07 -5.290000e+05
89581   VGG934461051   2011  1.935000e+07  9.799000e+06 -7.932000e+06
89582   VGG934461051   2012  1.572060e+07  4.964000e+06 -4.539000e+06
89583   VGG934461051   2013  1.152844e+07  4.722000e+06 -3.570000e+05

              country  year_0  relative_year p/b_group  bve_deflation
0       UNITED STATES    1996              0         5   4.964017e+09
```

```
1      UNITED STATES    1996              1         5   4.964017e+09
2      UNITED STATES    1996              2         5   4.964017e+09
3      UNITED STATES    1996              3         5   4.964017e+09
4      UNITED STATES    1996              4         5   4.964017e+09
...              ...     ...            ...       ...            ...
89579  UNITED STATES    2008              3         5            NaN
89580  UNITED STATES    2010              0        12            NaN
89581  UNITED STATES    2010              1        12            NaN
89582  UNITED STATES    2010              2        12            NaN
89583  UNITED STATES    2010              3        12            NaN

[89584 rows x 10 columns]

Filtered data with bve_prev:
                isin  year_           mve           bve          ninc  \
0      AN8068571086   1996  2.462290e+10  5.626380e+09  8.514830e+08
1      AN8068571086   1997  4.009191e+10  6.694924e+09  1.295697e+09
2      AN8068571086   1998  2.532696e+10  8.119075e+09  1.014199e+09
3      AN8068571086   1999  3.176288e+10  7.721028e+09  3.666940e+08
4      AN8068571086   2000  4.578247e+10  8.295216e+09  7.345960e+08
...             ...    ...           ...           ...           ...
89579  VGG9220E1079   2011  3.369199e+07  2.382768e+07 -9.825526e+06
89580  VGG934461051   2010  2.709000e+07  1.715800e+07 -5.290000e+05
89581  VGG934461051   2011  1.935000e+07  9.799000e+06 -7.932000e+06
89582  VGG934461051   2012  1.572060e+07  4.964000e+06 -4.539000e+06
89583  VGG934461051   2013  1.152844e+07  4.722000e+06 -3.570000e+05

              country  year_0  relative_year p/b_group  bve_deflation  \
0      UNITED STATES    1996              0         5   4.964017e+09
1      UNITED STATES    1996              1         5   4.964017e+09
```

```
2      UNITED STATES    1996              2         5   4.964017e+09
3      UNITED STATES    1996              3         5   4.964017e+09
4      UNITED STATES    1996              4         5   4.964017e+09
...              ...     ...            ...       ...            ...
89579  UNITED STATES    2008              3         5            NaN
89580  UNITED STATES    2010              0        12            NaN
89581  UNITED STATES    2010              1        12            NaN
89582  UNITED STATES    2010              2        12            NaN
89583  UNITED STATES    2010              3        12            NaN

           bve_prev
0      4.964017e+09
1      5.626380e+09
2      6.694924e+09
3      8.119075e+09
4      7.721028e+09
...             ...
89579  1.376815e+05
89580           NaN
89581  1.715800e+07
89582  9.799000e+06
89583  4.964000e+06

[89584 rows x 11 columns]

Residual income values:
                isin  year_           mve           bve          ninc  \
0      AN8068571086   1996  2.462290e+10  5.626380e+09  8.514830e+08
1      AN8068571086   1997  4.009191e+10  6.694924e+09  1.295697e+09
2      AN8068571086   1998  2.532696e+10  8.119075e+09  1.014199e+09
```

```
3      AN8068571086   1999  3.176288e+10  7.721028e+09  3.666940e+08
4      AN8068571086   2000  4.578247e+10  8.295216e+09  7.345960e+08
...             ...    ...           ...           ...           ...
89579  VGG9220E1079   2011  3.369199e+07  2.382768e+07 -9.825526e+06
89580  VGG934461051   2010  2.709000e+07  1.715800e+07 -5.290000e+05
89581  VGG934461051   2011  1.935000e+07  9.799000e+06 -7.932000e+06
89582  VGG934461051   2012  1.572060e+07  4.964000e+06 -4.539000e+06
89583  VGG934461051   2013  1.152844e+07  4.722000e+06 -3.570000e+05

              country  year_0  relative_year p/b_group  bve_deflation  \
0      UNITED STATES    1996              0         5   4.964017e+09
1      UNITED STATES    1996              1         5   4.964017e+09
2      UNITED STATES    1996              2         5   4.964017e+09
3      UNITED STATES    1996              3         5   4.964017e+09
4      UNITED STATES    1996              4         5   4.964017e+09
...              ...     ...            ...       ...           ...
89579  UNITED STATES    2008              3         5           NaN
89580  UNITED STATES    2010              0        12           NaN
89581  UNITED STATES    2010              1        12           NaN
89582  UNITED STATES    2010              2        12           NaN
89583  UNITED STATES    2010              3        12           NaN

           bve_prev  residual_income
0      4.964017e+09     4.543616e+08
1      5.626380e+09     8.455866e+08
2      6.694924e+09     4.786051e+08
3      8.119075e+09    -2.828320e+08
4      7.721028e+09     1.169138e+08
...             ...              ...
89579  1.376815e+05    -9.836540e+06
```

```
89580           NaN              NaN
89581  1.715800e+07    -9.304640e+06
89582  9.799000e+06    -5.322920e+06
89583  4.964000e+06    -7.541200e+05

[89584 rows x 12 columns]
```

Deflated residual income values:

```
                isin  year_           mve           bve          ninc  \
0       AN8068571086   1996  2.462290e+10  5.626380e+09  8.514830e+08
1       AN8068571086   1997  4.009191e+10  6.694924e+09  1.295697e+09
2       AN8068571086   1998  2.532696e+10  8.119075e+09  1.014199e+09
3       AN8068571086   1999  3.176288e+10  7.721028e+09  3.666940e+08
4       AN8068571086   2000  4.578247e+10  8.295216e+09  7.345960e+08
...              ...    ...           ...           ...           ...
89579   VGG9220E1079   2011  3.369199e+07  2.382768e+07 -9.825526e+06
89580   VGG934461051   2010  2.709000e+07  1.715800e+07 -5.290000e+05
89581   VGG934461051   2011  1.935000e+07  9.799000e+06 -7.932000e+06
89582   VGG934461051   2012  1.572060e+07  4.964000e+06 -4.539000e+06
89583   VGG934461051   2013  1.152844e+07  4.722000e+06 -3.570000e+05

              country  year_0  relative_year p/b_group  bve_deflation  \
0       UNITED STATES    1996              0         5   4.964017e+09
1       UNITED STATES    1996              1         5   4.964017e+09
2       UNITED STATES    1996              2         5   4.964017e+09
3       UNITED STATES    1996              3         5   4.964017e+09
4       UNITED STATES    1996              4         5   4.964017e+09
...               ...     ...            ...       ...            ...
89579   UNITED STATES    2008              3         5            NaN
89580   UNITED STATES    2010              0        12            NaN
```

```
89581  UNITED STATES    2010              1        12            NaN
89582  UNITED STATES    2010              2        12            NaN
89583  UNITED STATES    2010              3        12            NaN

           bve_prev  residual_income  deflated_residual_income
0      4.964017e+09     4.543616e+08                  0.091531
1      5.626380e+09     8.455866e+08                  0.170343
2      6.694924e+09     4.786051e+08                  0.096415
3      8.119075e+09    -2.828320e+08                 -0.056976
4      7.721028e+09     1.169138e+08                  0.023552
...             ...              ...                       ...
89579  1.376815e+05    -9.836540e+06                       NaN
89580           NaN              NaN                       NaN
89581  1.715800e+07    -9.304640e+06                       NaN
89582  9.799000e+06    -5.322920e+06                       NaN
89583  4.964000e+06    -7.541200e+05                       NaN

[89584 rows x 13 columns]
```

```python
# Output the replicated table


# Group by P/B group and relative year, then calculate the median deflated residual income for
median_residual_income = filtered_data.groupby(['p/b_group', 'relative_year'], observed=True)[

# Formatting the results (decimal places)
median_residual_income = median_residual_income.apply(lambda col: col.map(lambda x: f"{x:.3f}"
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)
print()
```

```
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
```

```
# Step 8: Modification - exclude negative and very high P/B values
```

```
# 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)]
print("Filtered Year 0 data after excluding negative and very high P/B values:\n", filtered_yea
print()

# 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(dr
filtered_year_0_data['p/b_group'] = pd.qcut(filtered_year_0_data.index, num_groups, labels=rang
print("Filtered Year 0 data updated with P/B groups:\n", filtered_year_0_data)
print()

# 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
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()

# Group by P/B group and relative year, then calculate the median deflated residual income for
median_residual_income_filtered = filtered_data.groupby(['p/b_group', 'relative_year'], observe

# Formatting the results (decimal places)
median_residual_income_filtered = median_residual_income_filtered.apply(lambda col: col.map(la
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_filte
```

```
Filtered Year 0 data after excluding negative and very high P/B values:
                isin  year_           mve           bve         ninc  \
2920    US14888B1035   1999  3.312000e+08  4.736410e+07  -11520099.0
2921    US40449J1034   2014  2.903267e+08  4.154500e+07     -32000.0
2922    US49720P5061   2003  2.944822e+08  4.215300e+07   -9872000.0
2923    US6718041022   2008  2.583850e+05  3.700000e+04     -16000.0
2924    US45769F1021   1996  1.368682e+08  1.962700e+07    2676000.0
...              ...    ...           ...           ...          ...
14971   US60447H1077   2006  1.500000e+02  7.016943e+06     847204.0
14972   US26878J2015   2005  2.538000e+03  2.020490e+08   12459000.0
14973   US2198801019   2015  3.073082e+04  2.594022e+09  -38207000.0
14974   US67021C3051   2005  8.450000e+03  1.201292e+09  131248000.0
14975   US48282H3084   2004  1.792613e+02  9.801800e+07    4887000.0

              country  year_0  relative_year       p/b p/b_group
2920    UNITED STATES    1999              0  6.992638         4
2921    UNITED STATES    2014              0  6.988246         4
2922    UNITED STATES    2003              0  6.986032         4
2923    UNITED STATES    2008              0  6.983378         4
2924    UNITED STATES    1996              0  6.973468         4
...               ...     ...            ...       ...       ...
14971   UNITED STATES    2006              0  0.000021        17
14972   UNITED STATES    2005              0  0.000013        17
14973   UNITED STATES    2015              0  0.000012        17
14974   UNITED STATES    2005              0  0.000007        17
14975   UNITED STATES    2004              0  0.000002        17
```

[12056 rows x 10 columns]

Filtered Year 0 data updated with P/B groups:

	isin	year_	mve	bve	ninc	\
0	US14888B1035	1999	3.312000e+08	4.736410e+07	-11520099.0	
1	US40449J1034	2014	2.903267e+08	4.154500e+07	-32000.0	
2	US49720P5061	2003	2.944822e+08	4.215300e+07	-9872000.0	
3	US6718041022	2008	2.583850e+05	3.700000e+04	-16000.0	
4	US45769F1021	1996	1.368682e+08	1.962700e+07	2676000.0	
...	...	...	...	...	...	
12051	US60447H1077	2006	1.500000e+02	7.016943e+06	847204.0	
12052	US26878J2015	2005	2.538000e+03	2.020490e+08	12459000.0	
12053	US2198801019	2015	3.073082e+04	2.594022e+09	-38207000.0	
12054	US67021C3051	2005	8.450000e+03	1.201292e+09	131248000.0	
12055	US48282H3084	2004	1.792613e+02	9.801800e+07	4887000.0	

	country	year_0	relative_year	p/b	p/b_group
0	UNITED STATES	1999	0	6.992638	1
1	UNITED STATES	2014	0	6.988246	1
2	UNITED STATES	2003	0	6.986032	1
3	UNITED STATES	2008	0	6.983378	1
4	UNITED STATES	1996	0	6.973468	1
...	...	...	...	...	...
12051	UNITED STATES	2006	0	0.000021	20
12052	UNITED STATES	2005	0	0.000013	20
12053	UNITED STATES	2015	0	0.000012	20
12054	UNITED STATES	2005	0	0.000007	20
12055	UNITED STATES	2004	0	0.000002	20

[12056 rows x 10 columns]

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.