Smart Beta, Fundamental Indexing, Factor Investing and Stock Returns

November 5, 2023

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

[2]:
```
compustat_data = pd.read_csv('funda_2022.csv', low_memory=False)
```

[3]:
```
compustat_data.head()
```

[3]:
```
   gvkey    datadate  fyear indfmt consol popsrc datafmt   tic      cusip  \
0   1000  1961-12-31   1961   INDL      C      D     STD  AE.2  000032102
1   1000  1962-12-31   1962   INDL      C      D     STD  AE.2  000032102
2   1000  1963-12-31   1963   INDL      C      D     STD  AE.2  000032102
3   1000  1964-12-31   1964   INDL      C      D     STD  AE.2  000032102
4   1000  1965-12-31   1965   INDL      C      D     STD  AE.2  000032102

                   conm  … priusa   sic spcindcd  spcseccd  spcsrc  state  \
0  A & E PLASTIK PAK INC  …    1.0  3089    325.0     978.0     NaN    NaN
1  A & E PLASTIK PAK INC  …    1.0  3089    325.0     978.0     NaN    NaN
2  A & E PLASTIK PAK INC  …    1.0  3089    325.0     978.0     NaN    NaN
3  A & E PLASTIK PAK INC  …    1.0  3089    325.0     978.0     NaN    NaN
4  A & E PLASTIK PAK INC  …    1.0  3089    325.0     978.0     NaN    NaN

  stko weburl       dldte ipodate
0  0.0    NaN  1978-06-30      NaN
1  0.0    NaN  1978-06-30      NaN
2  0.0    NaN  1978-06-30      NaN
3  0.0    NaN  1978-06-30      NaN
4  0.0    NaN  1978-06-30      NaN

[5 rows x 981 columns]
```

[4]:
```
compustat_data.shape
```

[4]:
```
(539318, 981)
```

[5]:
```
compustat_data.dtypes
```

[5]:
```
gvkey          int64
datadate      object
fyear          int64
indfmt        object
consol        object
               …
state         object
stko         float64
weburl        object
dldte         object
ipodate       object
Length: 981, dtype: object
```

[6]:
```
compustat_data.isna().sum()
```

[6]:
```
gvkey            0
datadate         0
fyear            0
indfmt           0
consol           0
              …
state        44784
stko         44376
weburl      240298
dldte       195763
ipodate     368348
Length: 981, dtype: int64
```

[7]:
```
# Convert 'date' column to datetime format
compustat_data['datadate'] = pd.to_datetime(compustat_data['datadate'])

# Create a copy of the dataframe to work with
compustat_copy = compustat_data.copy(deep=True)

# Check the data type of the 'date' column and the first few rows of the copied␣
 ↪dataframe
compustat_copy.dtypes['datadate']
```

[7]:
```
dtype('<M8[ns]')
```

[8]:
```
compustat_copy.shape
```

[8]:
```
(539318, 981)
```

[9]:
```
compustat_copy.head()
```

[9]:
```
   gvkey   datadate  fyear indfmt consol popsrc datafmt   tic      cusip  \
0   1000 1961-12-31   1961   INDL      C      D     STD  AE.2  000032102
1   1000 1962-12-31   1962   INDL      C      D     STD  AE.2  000032102
2   1000 1963-12-31   1963   INDL      C      D     STD  AE.2  000032102
3   1000 1964-12-31   1964   INDL      C      D     STD  AE.2  000032102
4   1000 1965-12-31   1965   INDL      C      D     STD  AE.2  000032102

                    conm  … priusa   sic spcindcd  spcseccd  spcsrc  state  \
0  A & E PLASTIK PAK INC  …    1.0  3089    325.0     978.0     NaN    NaN
1  A & E PLASTIK PAK INC  …    1.0  3089    325.0     978.0     NaN    NaN
2  A & E PLASTIK PAK INC  …    1.0  3089    325.0     978.0     NaN    NaN
3  A & E PLASTIK PAK INC  …    1.0  3089    325.0     978.0     NaN    NaN
4  A & E PLASTIK PAK INC  …    1.0  3089    325.0     978.0     NaN    NaN

  stko weburl       dldte ipodate
0  0.0    NaN  1978-06-30      NaN
1  0.0    NaN  1978-06-30      NaN
2  0.0    NaN  1978-06-30      NaN
3  0.0    NaN  1978-06-30      NaN
4  0.0    NaN  1978-06-30      NaN

[5 rows x 981 columns]
```

0.1 Function to compute the descriptive stats of the Financial Ratios Per Year

[10]:
```
def describe_yearly_stats(df, year):
    # Filter the dataframe for the given year
    df_year = df[df['datadate'].dt.year == year].copy(deep=True)

    # Sort by 'datadate' immediately after copying
    df_year.sort_values('datadate', inplace=True)

    # Columns for which to describe statistics
    columns_to_describe = [
        'ch', 'ivst', 'rect', 'invt', 'aco', 'act', 'ppent', 'ivaeq', 'ivao',␣
 ↪'intan', 'ao', 'at',
        'dlc', 'ap', 'txp', 'lco', 'lct', 'dltt', 'lo', 'txditc', 'mib', 'lt',␣
 ↪'pstk', 'ceq', 'teq',
        'sale', 'cogs', 'xsga', 'oibdp', 'dp', 'oiadp', 'xint', 'nopi', 'spi',␣
 ↪'pi', 'txt', 'mii', 'ib', 'dvp',
        'cstke', 'xido', 'ni', 'ibc', 'dpc', 'xidoc', 'txdc', 'esubc', 'sppiv',␣
 ↪'fopo', 'fopt', 'recch',
        'invch', 'apalch', 'txach', 'aoloch', 'oancf', 'ivch', 'siv', 'capx',␣
 ↪'sppe', 'aqc', 'ivstch', 'ivaco',
```

```
        'ivncf', 'sstk', 'prstkc', 'dv', 'dltis', 'dltr', 'dlcch', 'fiao',␣
↪'fincf', 'exre', 'chech', 'fsrco',
        'fuseo', 'wcapc', 'net_debt_issued_ratio', 'book_leverage', 'wc_ta',␣
↪'re_ta', 'ebit_ta', 'mv_tl',
        'sales_ta', 'z_score', 'sale_to_at_avg'
    ]

    # Compute additional ratios for describe
    df_year['dltis_at'] = df_year['dltis'] / df_year['at']
    df_year['dlc_at'] = df_year['dlc'] / df_year['at']
    df_year['dltt_at'] = df_year['dltt'] / df_year['at']
    df_year['ppent_at'] = df_year['ppent'] / df_year['at']
    df_year['mkvalt_at'] = df_year.get('mkvalt', pd.Series(index=df_year.
↪index)) / df_year['at']  # 'mkvalt' might not be present
    df_year['ni_at'] = df_year['ni'] / df_year['at']
    df_year['net_debt_issued'] = df_year['dltis'] - df_year['dltr'] +␣
↪df_year['dlcch']
    df_year['net_debt_issued_ratio'] = df_year['net_debt_issued'] /␣
↪df_year['at']
    df_year['book_leverage'] = (df_year['dltt'] + df_year['dlc']) /␣
↪(df_year['dltt'] + df_year['dlc'] + df_year['seq'])
    df_year['wc_ta'] = (df_year['act'] - df_year['lct']) / df_year['at']
    df_year['re_ta'] = df_year['re'] / df_year['at']
    df_year['ebit_ta'] = df_year['oiadp'] / df_year['at']
    df_year['mv_tl'] = (df_year['prcc_f'] * df_year['csho']) / df_year['lt']
    df_year['sales_ta'] = df_year['sale'] / df_year['at']

    # Calculate the Altman Z-Score
    df_year['z_score'] = (1.2 * df_year['wc_ta']) + \
                         (1.4 * df_year['re_ta']) + \
                         (3.3 * df_year['ebit_ta']) + \
                         (0.6 * df_year['mv_tl']) + \
                         (0.99 * df_year['sales_ta'])

    # Calculate rolling average for 'at' and 'sale_to_at_avg'
    df_year['at_rolling_avg'] = df_year['at'].rolling(window=2).mean()
    df_year['sale_to_at_avg'] = df_year['sale'] / df_year['at_rolling_avg']

    # Add all computed ratio columns to describe list
    computed_ratios = ['dltis_at', 'dlc_at', 'dltt_at', 'ppent_at',␣
↪'mkvalt_at', 'ni_at',
                       'net_debt_issued_ratio', 'book_leverage', 'wc_ta',␣
↪'re_ta',
                       'ebit_ta', 'mv_tl', 'sales_ta', 'z_score',␣
↪'sale_to_at_avg']
```

```
    columns_to_describe.extend(computed_ratios)

    # Print descriptive statistics for each column
    for col in columns_to_describe:
        if col in df_year:
            print(f"Descriptive statistics for {col} in the year {year}:")
            print(df_year[col].describe())
            print("\n")
        else:
            print(f"Column {col} not found in dataframe for the year {year}.\n")

    return df_year  # You can return the filtered dataframe with additional␣
↪columns if needed
```

0.2 Descriptive Stats of the Finaincal Ratios for the Year 1970

[11]:
```
compustat_copy_1970 = compustat_copy[compustat_copy['datadate'].dt.year == 1970]
```

[12]:
```
compustat_copy_1970['ch'].describe()
```

[12]:
```
count    2816.000000
mean       56.715982
std       341.745658
min         0.000000
25%         0.598750
50%         1.893500
75%         7.789000
max      6220.254000
Name: ch, dtype: float64
```

[13]:
```
compustat_copy_1970['ivst'].describe()
```

[13]:
```
count    2783.000000
mean       12.409155
std        57.738601
min        -0.001000
25%         0.000000
50%         0.200000
75%         3.397000
max      1173.203000
Name: ivst, dtype: float64
```

[14]:
```
compustat_copy_1970['rect'].describe()
```

[14]:
```
count     3378.000000
mean        94.053072
std        576.884046
```

```
min          0.000000
25%          2.284500
50%          6.822000
75%         24.602250
max      15951.402000
Name: rect, dtype: float64
```

[15]:
```
compustat_copy_1970['invt'].describe()
```

[15]:
```
count    3285.000000
mean       40.582404
std       141.497855
min         0.000000
25%         1.675000
50%         6.627000
75%        23.017000
max      4115.059000
Name: invt, dtype: float64
```

[16]:
```
compustat_copy_1970['aco'].describe()
```

[16]:
```
count    3263.000000
mean        2.563479
std        13.633103
min        -5.101000
25%         0.079500
50%         0.299000
75%         1.178500
max       390.000000
Name: aco, dtype: float64
```

[17]:
```
compustat_copy_1970['act'].describe()
```

[17]:
```
count    3474.000000
mean       93.495580
std       316.934658
min         0.020000
25%         6.948000
50%        18.689500
75%        54.736250
max      6527.746000
Name: act, dtype: float64
```

[18]:
```
compustat_copy_1970['ppent'].describe()
```

[18]:
```
count     3776.000000
mean       140.036473
std        834.682773
```

```
min          0.000000
25%          3.766250
50%         12.801000
75%         53.191250
max      42550.199000
Name: ppent, dtype: float64
```

[19]:
```
compustat_copy_1970['ivaeq'].describe()
```

[19]:
```
count    3296.000000
mean        9.187081
std        72.796807
min         0.000000
25%         0.000000
50%         0.000000
75%         0.295750
max      2631.908000
Name: ivaeq, dtype: float64
```

[20]:
```
compustat_copy_1970['ivao'].describe()
```

[20]:
```
count     3287.000000
mean        34.761339
std        338.156953
min          0.000000
25%          0.000000
50%          0.049000
75%          1.861500
max      15523.199000
Name: ivao, dtype: float64
```

[21]:
```
compustat_copy_1970['intan'].describe()
```

[21]:
```
count    3249.000000
mean        4.249424
std        17.780756
min         0.000000
25%         0.000000
50%         0.049000
75%         1.654000
max       435.248000
Name: intan, dtype: float64
```

[22]:
```
compustat_copy_1970['ao'].describe()
```

[22]:
```
count    3755.000000
mean        9.574822
std        60.544986
```

```
min        -0.025000
25%         0.142000
50%         0.664000
75%         3.266500
max      1767.878000
Name: ao, dtype: float64
```

[23]:
```
compustat_copy_1970['at'].describe()
```

[23]:
```
count     3782.000000
mean       443.840716
std       1755.187977
min          0.334000
25%         15.243750
50%         47.404500
75%        233.743500
max      49641.500000
Name: at, dtype: float64
```

[24]:
```
compustat_copy_1970['dlc'].describe()
```

[24]:
```
count    3715.000000
mean       28.348970
std       162.256449
min         0.000000
25%         0.486500
50%         2.538000
75%        12.168500
max      7646.275000
Name: dlc, dtype: float64
```

[25]:
```
compustat_copy_1970['ap'].describe()
```

[25]:
```
count     3384.000000
mean       102.399753
std        854.866108
min          0.000000
25%          1.009750
50%          3.141000
75%         11.234750
max      25643.207000
Name: ap, dtype: float64
```

[26]:
```
compustat_copy_1970['txp'].describe()
```

[26]:
```
count    3251.000000
mean        5.312647
std        31.613917
```

```
min        -0.117000
25%         0.068500
50%         0.430000
75%         1.677000
max       762.845000
Name: txp, dtype: float64
```

[27]:
```
compustat_copy_1970['lco'].describe()
```

[27]:
```
count    3254.000000
mean       13.520867
std        70.727329
min         0.000000
25%         0.506000
50%         1.574500
75%         5.751250
max      1865.442000
Name: lco, dtype: float64
```

[28]:
```
compustat_copy_1970['lct'].describe()
```

[28]:
```
count    3490.000000
mean       56.765866
std       254.182225
min         0.032000
25%         3.252250
50%         9.575500
75%        30.497250
max      7861.090000
Name: lct, dtype: float64
```

[29]:
```
compustat_copy_1970['dltt'].describe()
```

[29]:
```
count     3741.000000
mean        71.352833
std        374.361493
min          0.000000
25%          1.000000
50%          6.613000
75%         32.401000
max      18248.297000
Name: dltt, dtype: float64
```

[30]:
```
compustat_copy_1970['lo'].describe()
```

[30]:
```
count    3526.000000
mean       13.154559
std       118.343319
```

```
min        -0.018000
25%         0.000000
50%         0.000000
75%         1.087750
max      5630.297000
Name: lo, dtype: float64
```

[31]:
```
compustat_copy_1970['txditc'].describe()
```

[31]:
```
count    3462.000000
mean        6.944296
std        31.959672
min        -0.040000
25%         0.000000
50%         0.209000
75%         1.994500
max       998.353000
Name: txditc, dtype: float64
```

[32]:
```
compustat_copy_1970['mib'].describe()
```

[32]:
```
count    3750.000000
mean        2.344391
std        23.752359
min        -0.085000
25%         0.000000
50%         0.000000
75%         0.000000
max       798.276000
Name: mib, dtype: float64
```

[33]:
```
compustat_copy_1970['lt'].describe()
```

[33]:
```
count     3751.000000
mean       307.947243
std       1399.573805
min          0.032000
25%          6.337500
50%         22.698000
75%        121.887000
max      28229.309000
Name: lt, dtype: float64
```

[34]:
```
compustat_copy_1970['pstk'].describe()
```

[34]:
```
count    3778.000000
mean        5.943685
std        31.468075
```

```
min         0.000000
25%         0.000000
50%         0.000000
75%         0.250000
max       926.429000
Name: pstk, dtype: float64
```

[35]:
```
compustat_copy_1970['ceq'].describe()
```

[35]:
```
count     3768.000000
mean       122.166394
std        573.617310
min        -32.678000
25%          7.013750
50%         21.640000
75%         74.831250
max      24294.699000
Name: ceq, dtype: float64
```

[36]:
```
compustat_copy_1970['teq'].describe()
```

[36]:
```
count      299.000000
mean       191.503779
std        742.274621
min          0.475000
25%         10.283500
50%         37.268000
75%        141.541500
max      10950.642000
Name: teq, dtype: float64
```

[37]:
```
compustat_copy_1970['sale'].describe()
```

[37]:
```
count     3632.000000
mean       233.135155
std        801.758935
min          0.000000
25%         18.141500
50%         50.656000
75%        149.590500
max      18752.402000
Name: sale, dtype: float64
```

[38]:
```
compustat_copy_1970['cogs'].describe()
```

[38]:
```
count     3629.000000
mean       165.912341
std        578.336378
```

```
min          0.000000
25%         11.901000
50%         33.544000
75%        105.111000
max      15589.203000
Name: cogs, dtype: float64
```

[39]:
```
compustat_copy_1970['xsga'].describe()
```

[39]:
```
count    3154.000000
mean       36.918033
std       128.148356
min         0.051000
25%         2.764250
50%         7.662500
75%        22.866000
max      2985.069000
Name: xsga, dtype: float64
```

[40]:
```
compustat_copy_1970['oibdp'].describe()
```

[40]:
```
count    3624.000000
mean       35.202506
std       168.530230
min       -83.763000
25%         1.586750
50%         5.419000
75%        21.577250
max      6776.137000
Name: oibdp, dtype: float64
```

[41]:
```
compustat_copy_1970['dp'].describe()
```

[41]:
```
count    3487.000000
mean       11.232145
std        66.114212
min         0.000000
25%         0.366500
50%         1.218000
75%         5.043500
max      2531.971000
Name: dp, dtype: float64
```

[42]:
```
compustat_copy_1970['oiadp'].describe()
```

[42]:
```
count    3633.000000
mean       24.388670
std       108.392265
```

```
min      -140.118000
25%         1.022000
50%         3.945000
75%        16.388000
max      4244.167000
Name: oiadp, dtype: float64
```

[43]:
```
compustat_copy_1970['xint'].describe()
```

[43]:
```
count    3491.000000
mean        8.206649
std        38.976463
min        -0.092000
25%         0.188000
50%         0.827000
75%         4.272000
max      1003.301000
Name: xint, dtype: float64
```

[44]:
```
compustat_copy_1970['nopi'].describe()
```

[44]:
```
count    3625.000000
mean        2.512034
std        21.365369
min      -141.337000
25%         0.000000
50%         0.179000
75%         1.041000
max       713.842000
Name: nopi, dtype: float64
```

[45]:
```
compustat_copy_1970['spi'].describe()
```

[45]:
```
count    3271.000000
mean       -0.062276
std         2.533084
min      -143.058000
25%         0.000000
50%         0.000000
75%         0.000000
max         8.034000
Name: spi, dtype: float64
```

[46]:
```
(compustat_copy_1970['nopi'] + compustat_copy_1970['spi']).describe()
```

[46]:
```
count    3253.000000
mean        2.959300
std        21.966945
```

```
min      -133.951000
25%         0.003000
50%         0.170000
75%         0.899000
max       713.842000
dtype: float64
```

[47]:
```
compustat_copy_1970['pi'].describe()
```

[47]:
```
count    3766.000000
mean       21.647012
std       105.767522
min      -161.267000
25%         0.831250
50%         3.649500
75%        13.812250
max      3954.681000
Name: pi, dtype: float64
```

[48]:
```
compustat_copy_1970['txt'].describe()
```

[48]:
```
count    3784.000000
mean        9.033869
std        47.924850
min       -73.720000
25%         0.339750
50%         1.517000
75%         5.341000
max      1700.033000
Name: txt, dtype: float64
```

[49]:
```
compustat_copy_1970['mii'].describe()
```

[49]:
```
count    3585.000000
mean        0.222033
std         2.237573
min        -4.640000
25%         0.000000
50%         0.000000
75%         0.000000
max        65.249000
Name: mii, dtype: float64
```

[50]:
```
compustat_copy_1970['ib'].describe()
```

[50]:
```
count    3785.000000
mean       12.337072
std        57.746965
```

```
min      -143.094000
25%         0.441000
50%         1.959000
75%         8.051000
max      2189.400000
Name: ib, dtype: float64
```

[51]:
```
compustat_copy_1970['dvp'].describe()
```

[51]:
```
count    3785.000000
mean        0.491305
std         2.461121
min         0.000000
25%         0.000000
50%         0.000000
75%         0.026000
max        66.913000
Name: dvp, dtype: float64
```

[52]:
```
compustat_copy_1970['cstke'].describe()
```

[52]:
```
count    3786.000000
mean        0.079671
std         1.458710
min        -1.319000
25%         0.000000
50%         0.000000
75%         0.000000
max        65.663000
Name: cstke, dtype: float64
```

[53]:
```
compustat_copy_1970['xido'].describe()
```

[53]:
```
count    3784.000000
mean       -0.336147
std         5.169269
min      -264.271000
25%         0.000000
50%         0.000000
75%         0.000000
max        56.000000
Name: xido, dtype: float64
```

[54]:
```
compustat_copy_1970['ni'].describe()
```

[54]:
```
count    3643.000000
mean       11.798416
std        58.886175
```

```
min      -227.350000
25%         0.385000
50%         1.740000
75%         7.253500
max      2189.400000
Name: ni, dtype: float64
```

[55]:
```
compustat_copy_1970['ibc'].describe()
```

[55]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: ibc, dtype: float64
```

[56]:
```
compustat_copy_1970['dpc'].describe()
```

[56]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: dpc, dtype: float64
```

[57]:
```
(compustat_copy_1970['xidoc'] + compustat_copy_1970['txdc'] +␣
 ↪compustat_copy_1970['esubc'] +  compustat_copy_1970['sppiv'] +␣
 ↪compustat_copy_1970['fopo']).describe()
```

[57]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
dtype: float64
```

[58]:
```
compustat_copy_1970['fopt'].describe()
```

[58]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: fopt, dtype: float64
```

[59]:
```
compustat_copy_1970['recch'].describe()
```

[59]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: recch, dtype: float64
```

[60]:
```
compustat_copy_1970['invch'].describe()
```

[60]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: invch, dtype: float64
```

[61]:
```
compustat_copy_1970['apalch'].describe()
```

[61]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: apalch, dtype: float64
```

[62]:
```
compustat_copy_1970['txach'].describe()
```

[62]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: txach, dtype: float64
```

[63]:
```
compustat_copy_1970['aoloch'].describe()
```

[63]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: aoloch, dtype: float64
```

[64]:
```
compustat_copy_1970['oancf'].describe()
```

[64]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: oancf, dtype: float64
```

[65]:
```
compustat_copy_1970['ivch'].describe()
```

[65]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: ivch, dtype: float64
```

[66]:
```
compustat_copy_1970['siv'].describe()
```

[66]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: siv, dtype: float64
```

[67]:
```
compustat_copy_1970['capx'].describe()
```

[67]:
```
count    3101.000000
mean       22.737469
std       158.538837
min         0.000000
25%         0.713000
50%         2.205000
75%         8.658000
max      7159.180000
Name: capx, dtype: float64
```

[68]:
```
compustat_copy_1970['sppe'].describe()
```

[68]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: sppe, dtype: float64
```

[69]:
```
compustat_copy_1970['aqc'].describe()
```

[69]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: aqc, dtype: float64
```

[70]:
```
(compustat_copy_1970['ivstch']+compustat_copy_1970['ivaco']).describe()
```

[70]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
dtype: float64
```

[71]:
```
compustat_copy_1970['ivncf'].describe()
```

[71]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: ivncf, dtype: float64
```

[72]:
```
compustat_copy_1970['sstk'].describe()
```

[72]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: sstk, dtype: float64
```

[73]:
```
compustat_copy_1970['prstkc'].describe()
```

[73]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: prstkc, dtype: float64
```

[74]:
```
compustat_copy_1970['dv'].describe()
```

[74]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: dv, dtype: float64
```

[75]:
```
compustat_copy_1970['dltis'].describe()
```

[75]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: dltis, dtype: float64
```

[76]:
```
compustat_copy_1970['dltr'].describe()
```

[76]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: dltr, dtype: float64
```

[77]:
```
compustat_copy_1970['dlcch'].describe()
```

[77]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: dlcch, dtype: float64
```

[78]:
```
compustat_copy_1970['fiao'].describe()
```

[78]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: fiao, dtype: float64
```

[79]:
```
compustat_copy_1970['fincf'].describe()
```

[79]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: fincf, dtype: float64
```

[80]:
```
compustat_copy_1970['exre'].describe()
```

[80]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: exre, dtype: float64
```

[81]:
```
compustat_copy_1970['chech'].describe()
```

[81]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: chech, dtype: float64
```

[82]:
```
compustat_copy_1970['fsrco'].describe()
```

[82]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: fsrco, dtype: float64
```

[83]:
```
compustat_copy_1970['fuseo'].describe()
```

[83]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: fuseo, dtype: float64
```

[84]:
```
compustat_copy_1970['wcapc'].describe()
```

[84]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: wcapc, dtype: float64
```

[85]:
```
(compustat_copy_1970['dlc']+compustat_copy_1970['dltt']).describe()
```

[85]:
```
count     3714.000000
mean        99.893023
std        483.839336
min          0.000000
25%          2.499250
50%         11.703000
75%         57.115500
max      20453.632000
dtype: float64
```

[86]:
```
compustat_copy_1970['dv'].describe()
```

[86]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: dv, dtype: float64
```

[87]:
```
(compustat_copy_1970['dltis']/compustat_copy_1970['at']).describe()
```

[87]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
dtype: float64
```

[88]:
```
(compustat_copy_1970['dlc']/compustat_copy_1970['at']).describe()
```

[88]:
```
count    3715.000000
mean        0.089445
std         0.278293
min         0.000000
25%         0.017563
50%         0.053793
75%         0.115790
max        15.772455
dtype: float64
```

[89]:
```
compustat_copy_1970['net_debt_issued'] = compustat_copy_1970['dltis'] -␣
 ↪compustat_copy_1970['dltr'] + compustat_copy_1970['dlcch']

# Calculate the net debt issued ratio for each row
compustat_copy_1970['net_debt_issued_ratio'] =␣
 ↪compustat_copy_1970['net_debt_issued'] / compustat_copy_1970['at']

compustat_copy_1970['net_debt_issued_ratio'].describe()
```

[89]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: net_debt_issued_ratio, dtype: float64
```

[90]:
```
(compustat_copy_1970['dltt']/compustat_copy_1970['at']).describe()
```

[90]:
```
count    3740.000000
mean        0.196815
std         0.169121
min         0.000000
25%         0.048808
50%         0.166143
75%         0.301717
max         1.042676
dtype: float64
```

[91]:
```
compustat_copy_1970['book_leverage'] = (compustat_copy_1970['dltt'] +␣
 ↪compustat_copy_1970['dlc']) / (compustat_copy_1970['dltt'] +␣
 ↪compustat_copy_1970['dlc'] + compustat_copy_1970['seq'])
compustat_copy_1970['book_leverage'].describe()
```

[91]:
```
count    3217.000000
mean        0.368689
std         0.223648
min        -1.933921
25%         0.204989
50%         0.369464
75%         0.535259
max         1.579893
Name: book_leverage, dtype: float64
```

[92]:
```
(compustat_copy_1970['ppent']/compustat_copy_1970['at']).describe()
```

[92]:
```
count    3775.000000
mean        0.356182
std         0.250731
min         0.000000
25%         0.174207
50%         0.306077
75%         0.501145
max         0.963801
dtype: float64
```

[93]:
```
(compustat_copy_1970['mkvalt']/compustat_copy_1970['at']).describe()
```

[93]:
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
dtype: float64
```

[94]:
```
(compustat_copy_1970['ni']/compustat_copy_1970['at']).describe()
```

[94]:
```
count    3636.000000
mean        0.026027
std         0.223768
min       -10.988024
25%         0.013719
50%         0.041745
75%         0.070658
max         0.627324
dtype: float64
```

[95]:
```
((compustat_copy_1970['act']-compustat_copy_1970['lct'])/
 ↪compustat_copy_1970['at']).describe()
```

[95]:
```
count    3469.000000
mean        0.252852
std         0.477989
min       -24.868263
25%         0.110517
50%         0.268833
75%         0.410853
max         0.914160
dtype: float64
```

[96]:
```
(compustat_copy_1970['re']/compustat_copy_1970['at']).describe()
```

[96]:
```
count    3765.000000
mean        0.203784
std         1.019436
min       -59.083832
25%         0.090272
50%         0.225021
75%         0.383378
max         0.875967
dtype: float64
```

[97]:
```
(compustat_copy_1970['oiadp']/compustat_copy_1970['at']).describe()
```

[97]:
```
count    3626.000000
mean        0.084302
std         0.148275
min        -4.500000
25%         0.042407
50%         0.087794
75%         0.137177
max         0.712997
dtype: float64
```

[98]:
```
(compustat_copy_1970['prcc_f']*compustat_copy_1970['csho']).describe()
```

[98]:
```
count     3139.000000
mean       223.638669
std       1123.182872
min          0.296500
25%         11.869937
50%         34.502500
75%        126.491256
max      36409.973415
dtype: float64
```

[99]:
```
(compustat_copy_1970['sale']/compustat_copy_1970['at']).describe()
```

[99]:
```
count    3625.000000
mean        1.331206
std         1.066564
min         0.000000
25%         0.679422
50%         1.212104
75%         1.643656
max        10.232324
dtype: float64
```

[100]:
```
# Calculate the individual components of the Z-Score
compustat_copy_1970['wc_ta'] = (compustat_copy_1970['act'] -␣
 ↪compustat_copy_1970['lct']) / compustat_copy_1970['at']
compustat_copy_1970['re_ta'] = compustat_copy_1970['re'] /␣
 ↪compustat_copy_1970['at']
compustat_copy_1970['ebit_ta'] = compustat_copy_1970['oiadp'] /␣
 ↪compustat_copy_1970['at']
compustat_copy_1970['mv_tl'] = (compustat_copy_1970['prcc_f'] *␣
 ↪compustat_copy_1970['csho']) / compustat_copy_1970['lt']
compustat_copy_1970['sales_ta'] = compustat_copy_1970['sale'] /␣
 ↪compustat_copy_1970['at']

# Calculate the Altman Z-Score
```

```
compustat_copy_1970['z_score'] = (1.2 * compustat_copy_1970['wc_ta']) + \
                                 (1.4 * compustat_copy_1970['re_ta']) + \
                                 (3.3 * compustat_copy_1970['ebit_ta']) + \
                                 (0.6 * compustat_copy_1970['mv_tl']) + \
                                 (0.99 * compustat_copy_1970['sales_ta'])
compustat_copy_1970['z_score'].describe()
```

[100]:
```
count    2936.000000
mean        4.206318
std         4.875305
min       -20.407377
25%         2.050065
50%         3.219521
75%         4.912407
max       118.609690
Name: z_score, dtype: float64
```

[101]:
```
# First, sort the DataFrame by date if it's not already sorted
compustat_copy_1970 = compustat_copy_1970.sort_values('datadate')

# Calculate the rolling average of 'at' using a window of 2 periods (current␣
 ↪and previous)
compustat_copy_1970['at_rolling_avg'] = compustat_copy_1970['at'].
 ↪rolling(window=2).mean()

# Calculate the SALE/(AT(t) + AT(t-1))/2 ratio
# We'll use shift() to ensure we're not using future data
compustat_copy_1970['sale_to_at_avg'] = compustat_copy_1970['sale'] /␣
 ↪compustat_copy_1970['at_rolling_avg']
compustat_copy_1970['sale_to_at_avg'].describe()
```

[101]:
```
count    3567.000000
mean        1.240566
std         1.486658
min         0.000000
25%         0.253121
50%         0.805249
75%         1.776972
max        17.491663
Name: sale_to_at_avg, dtype: float64
```

[102]:
```
describe_yearly_stats(compustat_copy, 1970)
```

```
Descriptive statistics for ch in the year 1970:
count    2816.000000
mean       56.715982
std       341.745658
min         0.000000
```

```
25%         0.598750
50%         1.893500
75%         7.789000
max      6220.254000
Name: ch, dtype: float64


Descriptive statistics for ivst in the year 1970:
count    2783.000000
mean       12.409155
std        57.738601
min        -0.001000
25%         0.000000
50%         0.200000
75%         3.397000
max      1173.203000
Name: ivst, dtype: float64


Descriptive statistics for rect in the year 1970:
count     3378.000000
mean        94.053072
std        576.884046
min          0.000000
25%          2.284500
50%          6.822000
75%         24.602250
max      15951.402000
Name: rect, dtype: float64


Descriptive statistics for invt in the year 1970:
count    3285.000000
mean       40.582404
std       141.497855
min         0.000000
25%         1.675000
50%         6.627000
75%        23.017000
max      4115.059000
Name: invt, dtype: float64


Descriptive statistics for aco in the year 1970:
count    3263.000000
mean        2.563479
std        13.633103
min        -5.101000
```

```
25%         0.079500
50%         0.299000
75%         1.178500
max       390.000000
Name: aco, dtype: float64


Descriptive statistics for act in the year 1970:
count    3474.000000
mean       93.495580
std       316.934658
min         0.020000
25%         6.948000
50%        18.689500
75%        54.736250
max      6527.746000
Name: act, dtype: float64


Descriptive statistics for ppent in the year 1970:
count     3776.000000
mean       140.036473
std        834.682773
min          0.000000
25%          3.766250
50%         12.801000
75%         53.191250
max      42550.199000
Name: ppent, dtype: float64


Descriptive statistics for ivaeq in the year 1970:
count    3296.000000
mean        9.187081
std        72.796807
min         0.000000
25%         0.000000
50%         0.000000
75%         0.295750
max      2631.908000
Name: ivaeq, dtype: float64


Descriptive statistics for ivao in the year 1970:
count     3287.000000
mean        34.761339
std        338.156953
min          0.000000
```

```
25%          0.000000
50%          0.049000
75%          1.861500
max      15523.199000
Name: ivao, dtype: float64


Descriptive statistics for intan in the year 1970:
count    3249.000000
mean        4.249424
std        17.780756
min         0.000000
25%         0.000000
50%         0.049000
75%         1.654000
max       435.248000
Name: intan, dtype: float64


Descriptive statistics for ao in the year 1970:
count    3755.000000
mean        9.574822
std        60.544986
min        -0.025000
25%         0.142000
50%         0.664000
75%         3.266500
max      1767.878000
Name: ao, dtype: float64


Descriptive statistics for at in the year 1970:
count     3782.000000
mean       443.840716
std       1755.187977
min          0.334000
25%         15.243750
50%         47.404500
75%        233.743500
max      49641.500000
Name: at, dtype: float64


Descriptive statistics for dlc in the year 1970:
count    3715.000000
mean       28.348970
std       162.256449
min         0.000000
```

```
25%         0.486500
50%         2.538000
75%        12.168500
max      7646.275000
Name: dlc, dtype: float64


Descriptive statistics for ap in the year 1970:
count     3384.000000
mean       102.399753
std        854.866108
min          0.000000
25%          1.009750
50%          3.141000
75%         11.234750
max      25643.207000
Name: ap, dtype: float64


Descriptive statistics for txp in the year 1970:
count    3251.000000
mean        5.312647
std        31.613917
min        -0.117000
25%         0.068500
50%         0.430000
75%         1.677000
max       762.845000
Name: txp, dtype: float64


Descriptive statistics for lco in the year 1970:
count    3254.000000
mean       13.520867
std        70.727329
min         0.000000
25%         0.506000
50%         1.574500
75%         5.751250
max      1865.442000
Name: lco, dtype: float64


Descriptive statistics for lct in the year 1970:
count    3490.000000
mean       56.765866
std       254.182225
min         0.032000
```

```
25%         3.252250
50%         9.575500
75%        30.497250
max      7861.090000
Name: lct, dtype: float64


Descriptive statistics for dltt in the year 1970:
count     3741.000000
mean        71.352833
std        374.361493
min          0.000000
25%          1.000000
50%          6.613000
75%         32.401000
max      18248.297000
Name: dltt, dtype: float64


Descriptive statistics for lo in the year 1970:
count    3526.000000
mean       13.154559
std       118.343319
min        -0.018000
25%         0.000000
50%         0.000000
75%         1.087750
max      5630.297000
Name: lo, dtype: float64


Descriptive statistics for txditc in the year 1970:
count    3462.000000
mean        6.944296
std        31.959672
min        -0.040000
25%         0.000000
50%         0.209000
75%         1.994500
max       998.353000
Name: txditc, dtype: float64


Descriptive statistics for mib in the year 1970:
count    3750.000000
mean        2.344391
std        23.752359
min        -0.085000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max       798.276000
Name: mib, dtype: float64


Descriptive statistics for lt in the year 1970:
count     3751.000000
mean       307.947243
std       1399.573805
min          0.032000
25%          6.337500
50%         22.698000
75%        121.887000
max      28229.309000
Name: lt, dtype: float64


Descriptive statistics for pstk in the year 1970:
count    3778.000000
mean        5.943685
std        31.468075
min         0.000000
25%         0.000000
50%         0.000000
75%         0.250000
max       926.429000
Name: pstk, dtype: float64


Descriptive statistics for ceq in the year 1970:
count     3768.000000
mean       122.166394
std        573.617310
min        -32.678000
25%          7.013750
50%         21.640000
75%         74.831250
max      24294.699000
Name: ceq, dtype: float64


Descriptive statistics for teq in the year 1970:
count      299.000000
mean       191.503779
std        742.274621
min          0.475000
```

```
25%         10.283500
50%         37.268000
75%        141.541500
max      10950.642000
Name: teq, dtype: float64


Descriptive statistics for sale in the year 1970:
count     3632.000000
mean       233.135155
std        801.758935
min          0.000000
25%         18.141500
50%         50.656000
75%        149.590500
max      18752.402000
Name: sale, dtype: float64


Descriptive statistics for cogs in the year 1970:
count     3629.000000
mean       165.912341
std        578.336378
min          0.000000
25%         11.901000
50%         33.544000
75%        105.111000
max      15589.203000
Name: cogs, dtype: float64


Descriptive statistics for xsga in the year 1970:
count    3154.000000
mean       36.918033
std       128.148356
min         0.051000
25%         2.764250
50%         7.662500
75%        22.866000
max      2985.069000
Name: xsga, dtype: float64


Descriptive statistics for oibdp in the year 1970:
count    3624.000000
mean       35.202506
std       168.530230
min       -83.763000
```

```
25%         1.586750
50%         5.419000
75%        21.577250
max      6776.137000
Name: oibdp, dtype: float64


Descriptive statistics for dp in the year 1970:
count    3487.000000
mean       11.232145
std        66.114212
min         0.000000
25%         0.366500
50%         1.218000
75%         5.043500
max      2531.971000
Name: dp, dtype: float64


Descriptive statistics for oiadp in the year 1970:
count    3633.000000
mean       24.388670
std       108.392265
min      -140.118000
25%         1.022000
50%         3.945000
75%        16.388000
max      4244.167000
Name: oiadp, dtype: float64


Descriptive statistics for xint in the year 1970:
count    3491.000000
mean        8.206649
std        38.976463
min        -0.092000
25%         0.188000
50%         0.827000
75%         4.272000
max      1003.301000
Name: xint, dtype: float64


Descriptive statistics for nopi in the year 1970:
count    3625.000000
mean        2.512034
std        21.365369
min      -141.337000
```

```
25%         0.000000
50%         0.179000
75%         1.041000
max       713.842000
Name: nopi, dtype: float64


Descriptive statistics for spi in the year 1970:
count    3271.000000
mean       -0.062276
std         2.533084
min      -143.058000
25%         0.000000
50%         0.000000
75%         0.000000
max         8.034000
Name: spi, dtype: float64


Descriptive statistics for pi in the year 1970:
count    3766.000000
mean       21.647012
std       105.767522
min      -161.267000
25%         0.831250
50%         3.649500
75%        13.812250
max      3954.681000
Name: pi, dtype: float64


Descriptive statistics for txt in the year 1970:
count    3784.000000
mean        9.033869
std        47.924850
min       -73.720000
25%         0.339750
50%         1.517000
75%         5.341000
max      1700.033000
Name: txt, dtype: float64


Descriptive statistics for mii in the year 1970:
count    3585.000000
mean        0.222033
std         2.237573
min        -4.640000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max        65.249000
Name: mii, dtype: float64


Descriptive statistics for ib in the year 1970:
count    3785.000000
mean       12.337072
std        57.746965
min      -143.094000
25%         0.441000
50%         1.959000
75%         8.051000
max      2189.400000
Name: ib, dtype: float64


Descriptive statistics for dvp in the year 1970:
count    3785.000000
mean        0.491305
std         2.461121
min         0.000000
25%         0.000000
50%         0.000000
75%         0.026000
max        66.913000
Name: dvp, dtype: float64


Descriptive statistics for cstke in the year 1970:
count    3786.000000
mean        0.079671
std         1.458710
min        -1.319000
25%         0.000000
50%         0.000000
75%         0.000000
max        65.663000
Name: cstke, dtype: float64


Descriptive statistics for xido in the year 1970:
count    3784.000000
mean       -0.336147
std         5.169269
min      -264.271000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max        56.000000
Name: xido, dtype: float64


Descriptive statistics for ni in the year 1970:
count    3643.000000
mean       11.798416
std        58.886175
min      -227.350000
25%         0.385000
50%         1.740000
75%         7.253500
max      2189.400000
Name: ni, dtype: float64


Descriptive statistics for ibc in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: ibc, dtype: float64


Descriptive statistics for dpc in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: dpc, dtype: float64


Descriptive statistics for xidoc in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: xidoc, dtype: float64


Descriptive statistics for txdc in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: txdc, dtype: float64


Descriptive statistics for esubc in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: esubc, dtype: float64


Descriptive statistics for sppiv in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: sppiv, dtype: float64


Descriptive statistics for fopo in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: fopo, dtype: float64


Descriptive statistics for fopt in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: fopt, dtype: float64


Descriptive statistics for recch in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: recch, dtype: float64


Descriptive statistics for invch in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: invch, dtype: float64


Descriptive statistics for apalch in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: apalch, dtype: float64


Descriptive statistics for txach in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: txach, dtype: float64


Descriptive statistics for aoloch in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: aoloch, dtype: float64


Descriptive statistics for oancf in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: oancf, dtype: float64


Descriptive statistics for ivch in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: ivch, dtype: float64


Descriptive statistics for siv in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: siv, dtype: float64


Descriptive statistics for capx in the year 1970:
count    3101.000000
mean       22.737469
std       158.538837
min         0.000000
25%         0.713000
50%         2.205000
75%         8.658000
max      7159.180000
Name: capx, dtype: float64


Descriptive statistics for sppe in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: sppe, dtype: float64


Descriptive statistics for aqc in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: aqc, dtype: float64


Descriptive statistics for ivstch in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: ivstch, dtype: float64


Descriptive statistics for ivaco in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: ivaco, dtype: float64


Descriptive statistics for ivncf in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: ivncf, dtype: float64


Descriptive statistics for sstk in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: sstk, dtype: float64


Descriptive statistics for prstkc in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: prstkc, dtype: float64


Descriptive statistics for dv in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: dv, dtype: float64


Descriptive statistics for dltis in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: dltis, dtype: float64


Descriptive statistics for dltr in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: dltr, dtype: float64


Descriptive statistics for dlcch in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: dlcch, dtype: float64


Descriptive statistics for fiao in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: fiao, dtype: float64


Descriptive statistics for fincf in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: fincf, dtype: float64


Descriptive statistics for exre in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: exre, dtype: float64


Descriptive statistics for chech in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: chech, dtype: float64


Descriptive statistics for fsrco in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: fsrco, dtype: float64


Descriptive statistics for fuseo in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: fuseo, dtype: float64


Descriptive statistics for wcapc in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: wcapc, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 1970:
count    3217.000000
mean        0.368689
std         0.223648
min        -1.933921
25%         0.204989
50%         0.369464
75%         0.535259
max         1.579893
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 1970:
count    3469.000000
mean        0.252852
std         0.477989
min       -24.868263
25%         0.110517
50%         0.268833
75%         0.410853
max         0.914160
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 1970:
count    3765.000000
mean        0.203784
std         1.019436
min       -59.083832
```

```
25%         0.090272
50%         0.225021
75%         0.383378
max         0.875967
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 1970:
count    3626.000000
mean        0.084302
std         0.148275
min        -4.500000
25%         0.042407
50%         0.087794
75%         0.137177
max         0.712997
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 1970:
count    3087.000000
mean        2.927391
std         7.318612
min         0.020838
25%         0.596583
50%         1.163334
75%         2.748271
max       196.798544
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 1970:
count    3625.000000
mean        1.331206
std         1.066564
min         0.000000
25%         0.679422
50%         1.212104
75%         1.643656
max        10.232324
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 1970:
count    2936.000000
mean        4.206318
std         4.875305
min       -20.407377
```

```
25%         2.050065
50%         3.219521
75%         4.912407
max       118.609690
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 1970:
count    3567.000000
mean        1.240566
std         1.486658
min         0.000000
25%         0.253121
50%         0.805249
75%         1.776972
max        17.491663
Name: sale_to_at_avg, dtype: float64


Descriptive statistics for dltis_at in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: dltis_at, dtype: float64


Descriptive statistics for dlc_at in the year 1970:
count    3715.000000
mean        0.089445
std         0.278293
min         0.000000
25%         0.017563
50%         0.053793
75%         0.115790
max        15.772455
Name: dlc_at, dtype: float64


Descriptive statistics for dltt_at in the year 1970:
count    3740.000000
mean        0.196815
std         0.169121
min         0.000000
```

```
25%         0.048808
50%         0.166143
75%         0.301717
max         1.042676
Name: dltt_at, dtype: float64


Descriptive statistics for ppent_at in the year 1970:
count    3775.000000
mean        0.356182
std         0.250731
min         0.000000
25%         0.174207
50%         0.306077
75%         0.501145
max         0.963801
Name: ppent_at, dtype: float64


Descriptive statistics for mkvalt_at in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: mkvalt_at, dtype: float64


Descriptive statistics for ni_at in the year 1970:
count    3636.000000
mean        0.026027
std         0.223768
min       -10.988024
25%         0.013719
50%         0.041745
75%         0.070658
max         0.627324
Name: ni_at, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 1970:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 1970:
count    3217.000000
mean        0.368689
std         0.223648
min        -1.933921
25%         0.204989
50%         0.369464
75%         0.535259
max         1.579893
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 1970:
count    3469.000000
mean        0.252852
std         0.477989
min       -24.868263
25%         0.110517
50%         0.268833
75%         0.410853
max         0.914160
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 1970:
count    3765.000000
mean        0.203784
std         1.019436
min       -59.083832
25%         0.090272
50%         0.225021
75%         0.383378
max         0.875967
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 1970:
count    3626.000000
mean        0.084302
std         0.148275
min        -4.500000
```

```
25%         0.042407
50%         0.087794
75%         0.137177
max         0.712997
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 1970:
count    3087.000000
mean        2.927391
std         7.318612
min         0.020838
25%         0.596583
50%         1.163334
75%         2.748271
max       196.798544
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 1970:
count    3625.000000
mean        1.331206
std         1.066564
min         0.000000
25%         0.679422
50%         1.212104
75%         1.643656
max        10.232324
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 1970:
count    2936.000000
mean        4.206318
std         4.875305
min       -20.407377
25%         2.050065
50%         3.219521
75%         4.912407
max       118.609690
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 1970:
count    3567.000000
mean        1.240566
std         1.486658
min         0.000000
```

```
25%         0.253121
50%         0.805249
75%         1.776972
max        17.491663
Name: sale_to_at_avg, dtype: float64
```

[102]:

	gvkey	datadate	fyear	indfmt	consol	popsrc	datafmt	tic	\
88717	5270	1970-01-31	1969	INDL	C	D	STD	GTY.1	
122821	6939	1970-01-31	1969	INDL	C	D	STD	MAGM	
60812	3964	1970-01-31	1969	INDL	C	D	STD	DDS	
214500	11313	1970-01-31	1969	INDL	C	D	STD	WSO	
61777	4018	1970-01-31	1969	INDL	C	D	STD	DOMN.	
…	…	…	…	…	…	…	…	…	
96056	5629	1970-12-31	1970	INDL	C	D	STD	HVE	
96275	5643	1970-12-31	1970	INDL	C	D	STD	HLT	
96347	5646	1970-12-31	1970	INDL	C	D	STD	HINE	
93470	5518	1970-12-31	1970	INDL	C	D	STD	HAS	
471338	145348	1970-12-31	1970	INDL	C	D	STD	PPL2	

	cusip	conm	…	net_debt_issued_ratio	\
88717	387604101	GRANT (W.T.) CO	…	NaN	
122821	559142104	MAGIC MARKER CORP	…	NaN	
60812	254067101	DILLARDS INC -CL A	…	NaN	
214500	942622200	WATSCO INC	…	NaN	
61777	257028100	DOMAIN INDUSTRIES INC	…	NaN	
…	…	…	…	…	
96056	429812100	HIGH VOLTAGE ENGINEERING	…	NaN	
96275	43300A203	HILTON WORLDWIDE HOLDINGS	…	NaN	
96347	433236106	HINES (EDWARD) LUMBER CO	…	NaN	
93470	418056107	HASBRO INC	…	NaN	
471338	69399Y000	PPL ELECTRIC UTILITIES CORP	…	NaN	

	book_leverage	wc_ta	re_ta	ebit_ta	mv_tl	sales_ta	\
88717	NaN	0.359116	0.298919	0.126812	1.578530	1.704555	
122821	0.495194	0.266398	-0.172863	0.034336	0.739623	0.830926	
60812	0.357648	0.527356	0.184804	0.043400	0.500788	1.773427	
214500	0.247563	0.432993	0.495533	0.191449	3.877698	1.148692	
61777	0.609135	0.165404	0.101796	0.101227	0.410231	1.787840	
…	…	…	…	…	…	…	
96056	0.448090	0.382096	0.221653	-0.091226	1.714945	0.681846	
96275	0.560486	0.069927	0.309820	0.078673	1.326032	0.618748	
96347	0.000000	0.375394	0.547325	-0.045674	2.466434	1.862656	
93470	0.363097	0.245906	0.251699	0.052657	0.605859	1.455305	
471338	0.571266	-0.046235	0.095226	0.053404	0.535484	0.223858	

```
         z_score at_rolling_avg sale_to_at_avg
88717   3.902533            NaN            NaN
122821  1.457368       358.4115       0.009790
60812   3.090940        20.4985       3.181501
214500  5.308942        19.9540       0.180415
61777   2.691147        15.6295       3.217185
…          …              …              …
96056   2.171777        21.4455       0.837845
96275   2.185461       203.8255       1.157500
96347   4.389895       216.9885       0.452194
93470   2.625501        41.7880       1.076051
471338  0.796977       585.7050       0.435907

[3843 rows x 998 columns]
```

0.3 Descriptive Stats of the Finaincal Ratios for the Year 1975

[103]:
```
describe_yearly_stats(compustat_copy, 1975)
```

```
Descriptive statistics for ch in the year 1975:
count     5935.000000
mean        40.038351
std        454.300663
min        -26.222000
25%          0.224000
50%          1.030000
75%          4.413000
max      18258.805000
Name: ch, dtype: float64


Descriptive statistics for ivst in the year 1975:
count    5930.000000
mean       30.192074
std       217.762740
min        -0.001000
25%         0.000000
50%         0.000000
75%         2.475500
max      4584.441000
Name: ivst, dtype: float64


Descriptive statistics for rect in the year 1975:
count     5808.000000
mean       121.362790
std        979.482221
min          0.000000
```

```
25%          1.059000
50%          5.091000
75%         23.059000
max      36770.087000
Name: rect, dtype: float64


Descriptive statistics for invt in the year 1975:
count    5938.000000
mean       43.074459
std       191.206975
min         0.000000
25%         0.499250
50%         4.002500
75%        19.499750
max      5690.891000
Name: invt, dtype: float64


Descriptive statistics for aco in the year 1975:
count    5899.000000
mean        3.414496
std        18.599762
min        -0.088000
25%         0.034000
50%         0.218000
75%         1.076000
max       564.400000
Name: aco, dtype: float64


Descriptive statistics for act in the year 1975:
count     5547.000000
mean       107.259930
std        490.749265
min          0.000000
25%          3.213000
50%         13.360000
75%         50.285500
max      14004.801000
Name: act, dtype: float64


Descriptive statistics for ppent in the year 1975:
count     6256.000000
mean       149.117134
std       1061.155428
min          0.000000
```

```
25%          1.448500
50%          7.825500
75%         41.598000
max      70441.875000
Name: ppent, dtype: float64


Descriptive statistics for ivaeq in the year 1975:
count    5833.000000
mean        9.458247
std        83.976880
min        -0.932000
25%         0.000000
50%         0.000000
75%         0.047000
max      3480.517000
Name: ivaeq, dtype: float64


Descriptive statistics for ivao in the year 1975:
count     5833.000000
mean        46.483086
std        504.369854
min          0.000000
25%          0.000000
50%          0.018000
75%          1.578000
max      31201.406000
Name: ivao, dtype: float64


Descriptive statistics for intan in the year 1975:
count    5693.000000
mean        3.636551
std        19.303082
min         0.000000
25%         0.000000
50%         0.000000
75%         0.775000
max       571.063000
Name: intan, dtype: float64


Descriptive statistics for ao in the year 1975:
count    6253.000000
mean       14.903816
std        98.314933
min         0.000000
```

```
25%         0.054000
50%         0.365000
75%         2.390000
max      3294.525000
Name: ao, dtype: float64


Descriptive statistics for at in the year 1975:
count     6279.000000
mean       527.900332
std       2648.267584
min          0.000000
25%          7.758500
50%         33.251000
75%        177.371500
max      80156.188000
Name: at, dtype: float64


Descriptive statistics for dlc in the year 1975:
count    6252.000000
mean       41.932826
std       253.802873
min         0.000000
25%         0.230000
50%         1.347500
75%         9.382250
max      7812.962000
Name: dlc, dtype: float64


Descriptive statistics for ap in the year 1975:
count     6048.000000
mean       115.081967
std       1277.121478
min          0.000000
25%          0.492000
50%          2.173500
75%          9.949500
max      56544.816000
Name: ap, dtype: float64


Descriptive statistics for txp in the year 1975:
count    5874.000000
mean        6.380139
std        40.557747
min        -0.708000
```

```
25%         0.000000
50%         0.181500
75%         1.748500
max      1260.640000
Name: txp, dtype: float64


Descriptive statistics for lco in the year 1975:
count    5895.000000
mean       18.037907
std       108.781414
min         0.000000
25%         0.279500
50%         1.327000
75%         6.545000
max      4405.887000
Name: lco, dtype: float64


Descriptive statistics for lct in the year 1975:
count    5789.000000
mean       69.622106
std       363.407849
min         0.000000
25%         1.730000
50%         7.031000
75%        28.348000
max      9785.672000
Name: lct, dtype: float64


Descriptive statistics for dltt in the year 1975:
count     6254.000000
mean        84.051016
std        559.148045
min          0.000000
25%          0.572000
50%          5.023000
75%         31.576000
max      31793.309000
Name: dltt, dtype: float64


Descriptive statistics for lo in the year 1975:
count     6254.000000
mean        27.468600
std        290.259623
min        -19.228000
```

```
25%          0.000000
50%          0.000000
75%          1.188000
max      13398.871000
Name: lo, dtype: float64


Descriptive statistics for txditc in the year 1975:
count    5906.000000
mean       12.746174
std       102.569306
min         0.000000
25%         0.000000
50%         0.130500
75%         2.200750
max      6746.391000
Name: txditc, dtype: float64


Descriptive statistics for mib in the year 1975:
count    6134.000000
mean        2.461089
std        26.040393
min        -0.049000
25%         0.000000
50%         0.000000
75%         0.000000
max       863.784000
Name: mib, dtype: float64


Descriptive statistics for lt in the year 1975:
count     6256.000000
mean       389.801030
std       2282.535836
min          0.000000
25%          3.592500
50%         16.499500
75%         97.030750
max      64742.962000
Name: lt, dtype: float64


Descriptive statistics for pstk in the year 1975:
count    6279.000000
mean        6.848618
std        54.464118
min         0.000000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max      2995.784000
Name: pstk, dtype: float64


Descriptive statistics for ceq in the year 1975:
count     6278.000000
mean       125.449238
std        631.241973
min        -55.966000
25%          2.909000
50%         13.336000
75%         67.435000
max      30772.325000
Name: ceq, dtype: float64


Descriptive statistics for teq in the year 1975:
count      517.000000
mean       176.987354
std        871.828336
min        -11.318000
25%          3.178000
50%         15.343000
75%         94.943000
max      17024.382000
Name: teq, dtype: float64


Descriptive statistics for sale in the year 1975:
count     6097.000000
mean       292.178715
std       1318.432760
min          0.000000
25%          8.039000
50%         36.440000
75%        147.859000
max      44865.012000
Name: sale, dtype: float64


Descriptive statistics for cogs in the year 1975:
count     6096.000000
mean       213.850307
std        980.368430
min          0.000000
```

```
25%          5.057500
50%         24.852000
75%        102.705750
max      30982.008000
Name: cogs, dtype: float64


Descriptive statistics for xsga in the year 1975:
count    5034.000000
mean       42.790529
std       190.664076
min         0.000000
25%         1.431500
50%         5.524500
75%        21.908250
max      5935.102000
Name: xsga, dtype: float64


Descriptive statistics for oibdp in the year 1975:
count     6072.000000
mean        42.967093
std        269.002454
min       -271.979000
25%          0.519000
50%          3.589500
75%         18.810750
max      11608.305000
Name: oibdp, dtype: float64


Descriptive statistics for dp in the year 1975:
count    5863.000000
mean       11.692943
std        79.759458
min         0.000000
25%         0.163000
50%         0.783000
75%         3.852500
max      4089.501000
Name: dp, dtype: float64


Descriptive statistics for oiadp in the year 1975:
count    6097.000000
mean       31.661787
std       199.879178
min      -286.566000
```

```
25%         0.264000
50%         2.517000
75%        14.246000
max      9349.004000
Name: oiadp, dtype: float64


Descriptive statistics for xint in the year 1975:
count    5966.000000
mean       13.219119
std        85.797268
min         0.000000
25%         0.132000
50%         0.730500
75%         4.105000
max      2878.315000
Name: xint, dtype: float64


Descriptive statistics for nopi in the year 1975:
count    6094.000000
mean        2.865564
std        28.197259
min      -365.627000
25%         0.000000
50%         0.094000
75%         0.837000
max      1006.000000
Name: nopi, dtype: float64


Descriptive statistics for spi in the year 1975:
count    5604.000000
mean       -0.150684
std         3.876597
min      -200.000000
25%         0.000000
50%         0.000000
75%         0.000000
max        63.700000
Name: spi, dtype: float64


Descriptive statistics for pi in the year 1975:
count    6247.000000
mean       26.288295
std       190.669378
min      -294.806000
```

```
25%         0.105000
50%         1.870000
75%        11.228000
max      9905.000000
Name: pi, dtype: float64


Descriptive statistics for txt in the year 1975:
count    6268.000000
mean       12.205010
std       120.896413
min      -117.466000
25%         0.014750
50%         0.677500
75%         4.276250
max      7279.000000
Name: txt, dtype: float64


Descriptive statistics for mii in the year 1975:
count    5910.000000
mean        0.283829
std         3.341676
min        -2.617000
25%         0.000000
50%         0.000000
75%         0.000000
max       123.000000
Name: mii, dtype: float64


Descriptive statistics for ib in the year 1975:
count    6267.000000
mean       13.780208
std        75.761919
min      -207.214000
25%         0.065000
50%         1.049000
75%         7.060000
max      3147.701000
Name: ib, dtype: float64


Descriptive statistics for dvp in the year 1975:
count    6265.000000
mean        0.560960
std         4.190232
min         0.000000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max       231.900000
Name: dvp, dtype: float64


Descriptive statistics for cstke in the year 1975:
count    6268.000000
mean        0.050221
std         1.073108
min        -1.457000
25%         0.000000
50%         0.000000
75%         0.000000
max        64.482000
Name: cstke, dtype: float64


Descriptive statistics for xido in the year 1975:
count    6269.000000
mean       -0.113334
std         6.199290
min      -410.600000
25%         0.000000
50%         0.000000
75%         0.000000
max       119.620000
Name: xido, dtype: float64


Descriptive statistics for ni in the year 1975:
count    6106.000000
mean       13.402591
std        76.737757
min      -451.900000
25%         0.066000
50%         1.019000
75%         6.300500
max      3147.701000
Name: ni, dtype: float64


Descriptive statistics for ibc in the year 1975:
count    5806.000000
mean       13.103505
std        77.679661
min      -207.214000
```

```
25%         0.053000
50%         0.871000
75%         5.236000
max      3147.721000
Name: ibc, dtype: float64


Descriptive statistics for dpc in the year 1975:
count    5774.000000
mean       11.696381
std        79.291489
min        -0.156000
25%         0.191000
50%         0.814500
75%         3.959750
max      4088.088000
Name: dpc, dtype: float64


Descriptive statistics for xidoc in the year 1975:
count    5490.000000
mean        0.027410
std         2.618194
min       -91.769000
25%         0.000000
50%         0.000000
75%         0.000000
max       119.700000
Name: xidoc, dtype: float64


Descriptive statistics for txdc in the year 1975:
count    5614.000000
mean        2.309725
std        29.817741
min       -54.301000
25%         0.000000
50%         0.000000
75%         0.261000
max      2071.161000
Name: txdc, dtype: float64


Descriptive statistics for esubc in the year 1975:
count    4915.000000
mean       -0.642746
std        15.862486
min      -730.462000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max        49.000000
Name: esubc, dtype: float64


Descriptive statistics for sppiv in the year 1975:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: sppiv, dtype: float64


Descriptive statistics for fopo in the year 1975:
count    5799.000000
mean        0.712908
std        10.961684
min      -239.957000
25%         0.000000
50%         0.000000
75%         0.090000
max       379.988000
Name: fopo, dtype: float64


Descriptive statistics for fopt in the year 1975:
count    5836.000000
mean       27.050364
std       171.038337
min       -67.212000
25%         0.301000
50%         1.974500
75%        10.183750
max      9067.016000
Name: fopt, dtype: float64


Descriptive statistics for recch in the year 1975:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: recch, dtype: float64


Descriptive statistics for invch in the year 1975:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: invch, dtype: float64


Descriptive statistics for apalch in the year 1975:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: apalch, dtype: float64


Descriptive statistics for txach in the year 1975:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: txach, dtype: float64


Descriptive statistics for aoloch in the year 1975:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: aoloch, dtype: float64


Descriptive statistics for oancf in the year 1975:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: oancf, dtype: float64


Descriptive statistics for ivch in the year 1975:
count    5271.000000
mean        4.231421
std        68.960069
min       -27.883000
25%         0.000000
50%         0.000000
75%         0.019000
max      4319.973000
Name: ivch, dtype: float64


Descriptive statistics for siv in the year 1975:
count    4523.000000
mean        2.830986
std        39.175640
min       -23.320000
25%         0.000000
50%         0.000000
75%         0.000000
max      2128.805000
Name: siv, dtype: float64


Descriptive statistics for capx in the year 1975:
count    5829.000000
mean       25.882260
std       166.789329
min        -0.335000
```

```
25%         0.165000
50%         1.183000
75%         7.006000
max      9354.660000
Name: capx, dtype: float64


Descriptive statistics for sppe in the year 1975:
count    4485.000000
mean        1.805850
std        11.854664
min        -1.800000
25%         0.000000
50%         0.023000
75%         0.400000
max       411.755000
Name: sppe, dtype: float64


Descriptive statistics for aqc in the year 1975:
count    5373.000000
mean        0.520197
std         6.334484
min        -2.107000
25%         0.000000
50%         0.000000
75%         0.000000
max       242.014000
Name: aqc, dtype: float64


Descriptive statistics for ivstch in the year 1975:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: ivstch, dtype: float64


Descriptive statistics for ivaco in the year 1975:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: ivaco, dtype: float64


Descriptive statistics for ivncf in the year 1975:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: ivncf, dtype: float64


Descriptive statistics for sstk in the year 1975:
count    5785.000000
mean        3.104256
std        21.527824
min        -0.043000
25%         0.000000
50%         0.000000
75%         0.029000
max      1022.651000
Name: sstk, dtype: float64


Descriptive statistics for prstkc in the year 1975:
count    5788.000000
mean        0.265574
std         2.304605
min        -4.243000
25%         0.000000
50%         0.000000
75%         0.005000
max       104.114000
Name: prstkc, dtype: float64


Descriptive statistics for dv in the year 1975:
count    5784.000000
mean        6.382238
std        42.501757
min         0.000000
```

```
25%         0.000000
50%         0.065000
75%         1.438000
max      2166.360000
Name: dv, dtype: float64


Descriptive statistics for dltis in the year 1975:
count    5780.000000
mean       15.705315
std       110.470486
min       -40.000000
25%         0.000000
50%         0.213500
75%         3.996250
max      6736.680000
Name: dltis, dtype: float64


Descriptive statistics for dltr in the year 1975:
count    5725.000000
mean        7.734594
std        57.256475
min       -14.337000
25%         0.048000
50%         0.503000
75%         3.029000
max      3652.520000
Name: dltr, dtype: float64


Descriptive statistics for dlcch in the year 1975:
count    295.000000
mean      15.532736
std       52.899238
min     -151.500000
25%       -1.050000
50%        4.000000
75%       20.007500
max      299.537000
Name: dlcch, dtype: float64


Descriptive statistics for fiao in the year 1975:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: fiao, dtype: float64


Descriptive statistics for fincf in the year 1975:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: fincf, dtype: float64


Descriptive statistics for exre in the year 1975:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: exre, dtype: float64


Descriptive statistics for chech in the year 1975:
count    628.000000
mean       3.077753
std       38.698502
min     -200.984000
25%       -0.654750
50%        0.002500
75%        0.753000
max      582.557000
Name: chech, dtype: float64


Descriptive statistics for fsrco in the year 1975:
count    5535.000000
mean        3.635816
std        34.526515
min      -723.000000
```

```
25%         0.000000
50%         0.010000
75%         0.351500
max      1473.265000
Name: fsrco, dtype: float64


Descriptive statistics for fuseo in the year 1975:
count    5534.000000
mean        4.304038
std        34.779657
min      -298.000000
25%         0.000000
50%         0.029000
75%         0.472000
max      1302.000000
Name: fuseo, dtype: float64


Descriptive statistics for wcapc in the year 1975:
count    5127.000000
mean        4.033289
std        37.262729
min      -701.781000
25%        -0.293500
50%         0.225000
75%         2.052000
max       994.801000
Name: wcapc, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 1975:
count    293.000000
mean       0.052646
std        0.083273
min       -0.274088
25%        0.000436
50%        0.041199
75%        0.095329
max        0.409754
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 1975:
count    6221.000000
mean        0.490339
std         5.918094
min       -60.339623
```

```
25%         0.207809
50%         0.399071
75%         0.585992
max       448.428571
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 1975:
count    5537.000000
mean            -inf
std              NaN
min             -inf
25%         0.077567
50%         0.266016
75%         0.423020
max         0.992443
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 1975:
count    6271.000000
mean            -inf
std              NaN
min             -inf
25%         0.024723
50%         0.178463
75%         0.356859
max         0.986877
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 1975:
count    6097.000000
mean            -inf
std              NaN
min             -inf
25%         0.028011
50%         0.085869
75%         0.141706
max         2.808399
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 1975:
count    3894.000000
mean        1.560920
std         5.503953
min         0.001998
```

```
25%         0.270311
50%         0.589355
75%         1.358056
max       213.821078
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 1975:
count    6096.000000
mean        1.398648
std         1.148490
min         0.000000
25%         0.636743
50%         1.269404
75%         1.799214
max        12.490414
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 1975:
count    3566.000000
mean        3.580790
std         3.801300
min       -14.416281
25%         2.057367
50%         3.141745
75%         4.268687
max       127.924633
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 1975:
count    5866.000000
mean        1.320248
std         1.674175
min         0.000000
25%         0.180182
50%         0.746086
75%         2.019985
max        20.866232
Name: sale_to_at_avg, dtype: float64


Descriptive statistics for dltis_at in the year 1975:
count    5779.000000
mean        0.056807
std         0.112387
min        -0.409882
```

```
25%         0.000000
50%         0.011345
75%         0.073625
max         3.034335
Name: dltis_at, dtype: float64


Descriptive statistics for dlc_at in the year 1975:
count    6252.000000
mean             inf
std              NaN
min         0.000000
25%         0.013752
50%         0.048629
75%         0.119813
max              inf
Name: dlc_at, dtype: float64


Descriptive statistics for dltt_at in the year 1975:
count    6253.000000
mean        0.230304
std         0.931123
min         0.000000
25%         0.045459
50%         0.178579
75%         0.320024
max        68.573506
Name: dltt_at, dtype: float64


Descriptive statistics for ppent_at in the year 1975:
count    6255.000000
mean        0.342234
std         0.257849
min         0.000000
25%         0.142185
50%         0.289753
75%         0.499128
max         1.000000
Name: ppent_at, dtype: float64


Descriptive statistics for mkvalt_at in the year 1975:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: mkvalt_at, dtype: float64


Descriptive statistics for ni_at in the year 1975:
count    6106.000000
mean            -inf
std              NaN
min             -inf
25%         0.006056
50%         0.039355
75%         0.074876
max         4.072961
Name: ni_at, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 1975:
count    293.000000
mean       0.052646
std        0.083273
min       -0.274088
25%        0.000436
50%        0.041199
75%        0.095329
max        0.409754
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 1975:
count    6221.000000
mean        0.490339
std         5.918094
min       -60.339623
25%         0.207809
50%         0.399071
75%         0.585992
max       448.428571
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 1975:
count    5537.000000
mean            -inf
std              NaN
min             -inf
```

```
25%         0.077567
50%         0.266016
75%         0.423020
max         0.992443
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 1975:
count    6271.000000
mean            -inf
std              NaN
min             -inf
25%         0.024723
50%         0.178463
75%         0.356859
max         0.986877
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 1975:
count    6097.000000
mean            -inf
std              NaN
min             -inf
25%         0.028011
50%         0.085869
75%         0.141706
max         2.808399
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 1975:
count    3894.000000
mean        1.560920
std         5.503953
min         0.001998
25%         0.270311
50%         0.589355
75%         1.358056
max       213.821078
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 1975:
count    6096.000000
mean        1.398648
std         1.148490
min         0.000000
```

```
25%         0.636743
50%         1.269404
75%         1.799214
max        12.490414
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 1975:
count    3566.000000
mean        3.580790
std         3.801300
min       -14.416281
25%         2.057367
50%         3.141745
75%         4.268687
max       127.924633
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 1975:
count    5866.000000
mean        1.320248
std         1.674175
min         0.000000
25%         0.180182
50%         0.746086
75%         2.019985
max        20.866232
Name: sale_to_at_avg, dtype: float64
```

[103]:

	gvkey	datadate	fyear	indfmt	consol	popsrc	datafmt	tic	\
212288	11220	1975-01-31	1974	INDL	C	D	STD	VNO	
190306	10151	1975-01-31	1974	INDL	C	D	STD	SNID	
166719	8987	1975-01-31	1974	INDL	C	D	STD	IHT	
155198	8445	1975-01-31	1974	INDL	C	D	STD	JCP1	
155253	8446	1975-01-31	1974	INDL	C	D	STD	CPPRQ	
…	…	…	…	…	…	…	…	…	
98057	5732	1975-12-31	1975	INDL	C	D	STD	HOV.	
97988	5727	1975-12-31	1975	INDL	C	D	STD	1595B	
97956	5726	1975-12-31	1975	INDL	C	D	STD	0780B	
98276	5742	1975-12-31	1975	INDL	C	D	STD	CNP	
471343	145348	1975-12-31	1975	INDL	C	D	STD	PPL2	

	cusip	conm	…	net_debt_issued_ratio	\
212288	929042109	VORNADO REALTY TRUST	…	NaN	

```
190306  866665102          SUN CITY INDUSTRIES  …                  NaN
166719  457919108     INNSUITES HOSPITALITY TR  …                  NaN
155198  708152004    PENNEY (J C) FUNDING CORP  …                  NaN
155253  679535104            OLD COPPER CO INC  …                  NaN
…          …                            …  …                  …
98057   441776101          HOUSE OF VISION INC  …                  NaN
97988   441722105           HOUSE OF ADLER INC  …                  NaN
97956   44199Z937          HOUGHTON MIFFLIN CO  …                  NaN
98276   15189T107       CENTERPOINT ENERGY INC  …             0.062795
471343  69399Y000  PPL ELECTRIC UTILITIES CORP  …             0.061065

       book_leverage     wc_ta     re_ta   ebit_ta     mv_tl  sales_ta  \
212288       0.514263  0.219783  0.250043  0.027954  0.155054  2.373157
190306       0.207486  0.397855  0.348206  0.085722  0.751760  4.571388
166719       0.868015       NaN  0.002882  0.076744  0.099516  0.080871
155198       0.805020       NaN  0.084846  0.118942       NaN  0.119264
155253       0.237795  0.246271  0.382381  0.127919  2.195739  2.526507
…               …       …       …       …       …       …
98057        0.281315  0.462315  0.434307  0.123573  0.965097  2.101274
97988        0.270270  0.379332  0.102079  0.105230       NaN  1.272842
97956        0.092244  0.515302  0.452413  0.156381  1.444010  1.118178
98276        0.569488 -0.052759  0.185248  0.090666  0.453094  0.318573
471343       0.532179  0.003335  0.088094  0.072784  0.394448  0.235362

         z_score at_rolling_avg sale_to_at_avg
212288  3.148504            NaN            NaN
190306  6.224528       193.4185       0.248968
166719       NaN        79.1625       0.150981
155198       NaN       806.3560       0.216669
155253  5.071677      2105.0485       3.294797
…          …             …             …
98057   4.229917        11.8095       2.960413
97988        NaN         9.1125       0.221674
97956   3.741201        44.2195       2.196226
98276   1.082479      1038.7300       0.610508
471343  0.837196      2151.2455       0.252937

[6518 rows x 998 columns]
```

0.4 Descriptive Stats of the Finaincal Ratios for the Year 1980

[104]:
```
describe_yearly_stats(compustat_copy, 1980)
```

```
Descriptive statistics for ch in the year 1980:
count     5607.000000
mean        56.821866
std        435.121621
min        -49.850000
```

```
25%          0.181000
50%          0.869000
75%          4.597000
max      10535.402000
Name: ch, dtype: float64


Descriptive statistics for ivst in the year 1980:
count     5594.000000
mean        77.444861
std        632.378519
min          0.000000
25%          0.000000
50%          0.100000
75%          5.409000
max      19576.656000
Name: ivst, dtype: float64


Descriptive statistics for rect in the year 1980:
count     5700.000000
mean       220.909980
std       1801.991568
min          0.000000
25%          1.263750
50%          7.521500
75%         45.160500
max      71763.000000
Name: rect, dtype: float64


Descriptive statistics for invt in the year 1980:
count    5960.000000
mean       74.345338
std       349.529047
min         0.000000
25%         0.379000
50%         4.578000
75%        28.085750
max      9087.000000
Name: invt, dtype: float64


Descriptive statistics for aco in the year 1980:
count    5859.000000
mean        7.772503
std        47.760249
min        -7.100000
```

```
25%         0.038000
50%         0.311000
75%         2.076000
max      2133.778000
Name: aco, dtype: float64


Descriptive statistics for act in the year 1980:
count     5406.000000
mean       198.378401
std        914.975335
min          0.000000
25%          3.755500
50%         18.340000
75%         85.745500
max      23458.500000
Name: act, dtype: float64


Descriptive statistics for ppent in the year 1980:
count      6302.000000
mean        277.787086
std        1880.866915
min           0.000000
25%           1.555750
50%          11.696500
75%          68.438250
max      110023.000000
Name: ppent, dtype: float64


Descriptive statistics for ivaeq in the year 1980:
count    5667.000000
mean       16.934636
std       138.092539
min        -4.536000
25%         0.000000
50%         0.000000
75%         0.000500
max      5017.699000
Name: ivaeq, dtype: float64


Descriptive statistics for ivao in the year 1980:
count     5722.000000
mean        89.314783
std        727.860457
min          0.000000
```

```
25%          0.000000
50%          0.005000
75%          2.580000
max      31325.797000
Name: ivao, dtype: float64


Descriptive statistics for intan in the year 1980:
count    5407.000000
mean        5.280635
std        29.821870
min         0.000000
25%         0.000000
50%         0.000000
75%         0.458000
max       860.768000
Name: intan, dtype: float64


Descriptive statistics for ao in the year 1980:
count     6291.000000
mean        43.980546
std        756.405042
min         -3.135000
25%          0.058000
50%          0.563000
75%          5.268000
max      56322.946000
Name: ao, dtype: float64


Descriptive statistics for at in the year 1980:
count      6326.000000
mean        976.960267
std        4825.329446
min           0.000000
25%           8.692750
50%          47.478000
75%         331.939500
max      125451.000000
Name: at, dtype: float64


Descriptive statistics for dlc in the year 1980:
count     6288.000000
mean        98.953494
std        710.994688
min          0.000000
```

```
25%          0.243000
50%          1.763000
75%         13.613000
max      23377.008000
Name: dlc, dtype: float64


Descriptive statistics for ap in the year 1980:
count     6040.000000
mean       207.893868
std       2082.119154
min          0.000000
25%          0.621750
50%          3.267500
75%         20.640750
max      88426.188000
Name: ap, dtype: float64


Descriptive statistics for txp in the year 1980:
count    5797.000000
mean       13.957180
std       117.152393
min        -1.917000
25%         0.000000
50%         0.238000
75%         2.618000
max      4555.000000
Name: txp, dtype: float64


Descriptive statistics for lco in the year 1980:
count    5855.000000
mean       38.790715
std       210.013617
min         0.000000
25%         0.360000
50%         2.025000
75%        11.832500
max      6628.793000
Name: lco, dtype: float64


Descriptive statistics for lct in the year 1980:
count     5661.000000
mean       142.633203
std        751.124477
min          0.000000
```

```
25%          2.055000
50%          9.393000
75%         47.274000
max      17365.402000
Name: lct, dtype: float64


Descriptive statistics for dltt in the year 1980:
count     6295.000000
mean       144.597504
std        923.655840
min          0.000000
25%          0.658500
50%          7.367000
75%         47.802500
max      41255.012000
Name: dltt, dtype: float64


Descriptive statistics for lo in the year 1980:
count     6299.000000
mean        61.901399
std        636.509945
min         -2.510000
25%          0.000000
50%          0.011000
75%          2.821000
max      31892.406000
Name: lo, dtype: float64


Descriptive statistics for txditc in the year 1980:
count     5833.000000
mean        32.401810
std        278.959341
min         -3.192000
25%          0.000000
50%          0.192000
75%          4.300000
max      17641.301000
Name: txditc, dtype: float64


Descriptive statistics for mib in the year 1980:
count    6027.000000
mean        4.621262
std        68.543046
min        -0.290000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max      3489.000000
Name: mib, dtype: float64


Descriptive statistics for lt in the year 1980:
count      6298.000000
mean        739.510644
std        4214.360718
min           0.000000
25%           4.329500
50%          24.315000
75%         179.419000
max      110989.000000
Name: lt, dtype: float64


Descriptive statistics for pstk in the year 1980:
count    6325.000000
mean       11.369041
std        68.183903
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max      1959.700000
Name: pstk, dtype: float64


Descriptive statistics for ceq in the year 1980:
count     6325.000000
mean       215.816689
std       1058.280870
min       -339.800000
25%          3.019000
50%         18.145000
75%        108.082000
max      49447.613000
Name: ceq, dtype: float64


Descriptive statistics for teq in the year 1980:
count      614.000000
mean       287.331884
std       1351.241647
min        -55.603000
```

```
25%          2.650000
50%         20.706500
75%        125.536750
max      25412.654000
Name: teq, dtype: float64


Descriptive statistics for sale in the year 1980:
count      6101.00000
mean        589.00119
std        2900.13737
min           0.00000
25%           8.60900
50%          49.78000
75%         267.63700
max      103142.00000
Name: sale, dtype: float64


Descriptive statistics for cogs in the year 1980:
count     6101.000000
mean       440.473702
std       2311.365572
min         -1.374000
25%          5.300000
50%         32.555000
75%        181.689000
max      84260.750000
Name: cogs, dtype: float64


Descriptive statistics for xsga in the year 1980:
count     4877.000000
mean        78.894045
std        332.471564
min         -0.536000
25%          1.692000
50%          7.563000
75%         37.476000
max      10324.000000
Name: xsga, dtype: float64


Descriptive statistics for oibdp in the year 1980:
count     6076.000000
mean        85.158877
std        481.176518
min       -770.902000
```

```
25%          0.708000
50%          5.255500
75%         34.350500
max      19787.605000
Name: oibdp, dtype: float64


Descriptive statistics for dp in the year 1980:
count    5830.000000
mean       22.760553
std       156.827721
min         0.000000
25%         0.170250
50%         1.055000
75%         6.544000
max      7112.102000
Name: dp, dtype: float64


Descriptive statistics for oiadp in the year 1980:
count     6101.000000
mean        63.712396
std        356.579948
min      -2278.301000
25%          0.409000
50%          3.839000
75%         26.940000
max      12675.504000
Name: oiadp, dtype: float64


Descriptive statistics for xint in the year 1980:
count     5997.000000
mean        34.334550
std        255.911918
min          0.000000
25%          0.188000
50%          1.195000
75%          7.686000
max      10495.000000
Name: xint, dtype: float64


Descriptive statistics for nopi in the year 1980:
count    6101.000000
mean        8.123296
std        66.678540
min      -871.000000
```

```
25%         0.006000
50%         0.252000
75%         1.941000
max      2127.063000
Name: nopi, dtype: float64


Descriptive statistics for spi in the year 1980:
count    5456.000000
mean        0.189431
std         8.607290
min      -187.500000
25%         0.000000
50%         0.000000
75%         0.000000
max       294.417000
Name: spi, dtype: float64


Descriptive statistics for pi in the year 1980:
count     6290.000000
mean        53.449707
std        339.740022
min      -1980.700000
25%          0.224000
50%          3.169500
75%         22.189000
max      11272.598000
Name: pi, dtype: float64


Descriptive statistics for txt in the year 1980:
count    6316.000000
mean       22.140919
std       164.753193
min      -435.400000
25%         0.025750
50%         1.021000
75%         7.260750
max      5428.000000
Name: txt, dtype: float64


Descriptive statistics for mii in the year 1980:
count    5830.000000
mean        0.730326
std        16.721953
min       -14.000000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max      1116.000000
Name: mii, dtype: float64


Descriptive statistics for ib in the year 1980:
count    6316.000000
mean       30.468719
std       175.102790
min     -1709.700000
25%         0.158750
50%         2.003500
75%        14.700250
max      6079.699000
Name: ib, dtype: float64


Descriptive statistics for dvp in the year 1980:
count    6314.000000
mean        0.934449
std         5.542657
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max       150.700000
Name: dvp, dtype: float64


Descriptive statistics for cstke in the year 1980:
count    6316.000000
mean        0.047672
std         0.917203
min        -1.014000
25%         0.000000
50%         0.000000
75%         0.000000
max        64.011000
Name: cstke, dtype: float64


Descriptive statistics for xido in the year 1980:
count    6316.000000
mean        0.259028
std        14.513250
min      -411.884000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max       459.000000
Name: xido, dtype: float64


Descriptive statistics for ni in the year 1980:
count    6114.000000
mean       30.464484
std       181.292506
min     -1709.700000
25%         0.162250
50%         1.861500
75%        13.096000
max      6079.699000
Name: ni, dtype: float64


Descriptive statistics for ibc in the year 1980:
count    5815.000000
mean       29.400666
std       180.914351
min     -1709.700000
25%         0.120000
50%         1.497000
75%        10.428000
max      6079.699000
Name: ibc, dtype: float64


Descriptive statistics for dpc in the year 1980:
count    5771.000000
mean       23.332352
std       157.654941
min         0.000000
25%         0.197000
50%         1.150000
75%         6.751000
max      7039.199000
Name: dpc, dtype: float64


Descriptive statistics for xidoc in the year 1980:
count    5465.000000
mean        0.704257
std        14.397752
min       -83.000000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max       568.286000
Name: xidoc, dtype: float64


Descriptive statistics for txdc in the year 1980:
count    5639.000000
mean        5.390247
std        52.981646
min       -79.900000
25%         0.000000
50%         0.000000
75%         0.550000
max      2750.701000
Name: txdc, dtype: float64


Descriptive statistics for esubc in the year 1980:
count    4791.000000
mean       -0.633294
std        13.713725
min      -324.341000
25%         0.000000
50%         0.000000
75%         0.000000
max       464.001000
Name: esubc, dtype: float64


Descriptive statistics for sppiv in the year 1980:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: sppiv, dtype: float64


Descriptive statistics for fopo in the year 1980:
count    5812.000000
mean        2.822231
std       111.375721
min      -294.983000
```

```
25%         0.000000
50%         0.000000
75%         0.071000
max      8188.000000
Name: fopo, dtype: float64


Descriptive statistics for fopt in the year 1980:
count     5841.000000
mean        60.583716
std        403.662661
min      -1102.700000
25%          0.368000
50%          2.992000
75%         19.014000
max      15367.102000
Name: fopt, dtype: float64


Descriptive statistics for recch in the year 1980:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: recch, dtype: float64


Descriptive statistics for invch in the year 1980:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: invch, dtype: float64


Descriptive statistics for apalch in the year 1980:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: apalch, dtype: float64


Descriptive statistics for txach in the year 1980:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: txach, dtype: float64


Descriptive statistics for aoloch in the year 1980:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: aoloch, dtype: float64


Descriptive statistics for oancf in the year 1980:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: oancf, dtype: float64


Descriptive statistics for ivch in the year 1980:
count    5234.000000
mean       11.266934
std       183.730904
min       -66.500000
```

```
25%         0.000000
50%         0.000000
75%         0.099000
max      9164.457000
Name: ivch, dtype: float64


Descriptive statistics for siv in the year 1980:
count    5225.000000
mean        5.271646
std       135.951293
min      -324.226000
25%         0.000000
50%         0.000000
75%         0.000000
max      9030.785000
Name: siv, dtype: float64


Descriptive statistics for capx in the year 1980:
count     5781.000000
mean        57.751885
std        366.788833
min         -0.007000
25%          0.267000
50%          2.074000
75%         14.814000
max      17029.805000
Name: capx, dtype: float64


Descriptive statistics for sppe in the year 1980:
count    4379.000000
mean        3.055357
std        16.048819
min        -0.731000
25%         0.000000
50%         0.032000
75%         0.699500
max       420.506000
Name: sppe, dtype: float64


Descriptive statistics for aqc in the year 1980:
count    5245.000000
mean        3.605758
std        46.898276
min        -9.609000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max      2507.001000
Name: aqc, dtype: float64


Descriptive statistics for ivstch in the year 1980:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: ivstch, dtype: float64


Descriptive statistics for ivaco in the year 1980:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: ivaco, dtype: float64


Descriptive statistics for ivncf in the year 1980:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: ivncf, dtype: float64


Descriptive statistics for sstk in the year 1980:
count    5755.000000
mean        6.743768
std        46.741367
min        -4.000000
```

```
25%         0.000000
50%         0.004000
75%         0.536500
max      2591.701000
Name: sstk, dtype: float64


Descriptive statistics for prstkc in the year 1980:
count    5775.000000
mean        1.387423
std        16.693244
min        -0.001000
25%         0.000000
50%         0.000000
75%         0.003000
max       715.100000
Name: prstkc, dtype: float64


Descriptive statistics for dv in the year 1980:
count    5818.000000
mean       12.670638
std        79.325181
min         0.000000
25%         0.000000
50%         0.134500
75%         2.956000
max      3769.901000
Name: dv, dtype: float64


Descriptive statistics for dltis in the year 1980:
count     5728.000000
mean        28.785925
std        209.565392
min       -174.000000
25%          0.000000
50%          0.465500
75%          7.160000
max      11480.203000
Name: dltis, dtype: float64


Descriptive statistics for dltr in the year 1980:
count    5665.000000
mean       13.891352
std       107.915824
min       -13.224000
```

```
25%         0.067000
50%         0.727000
75%         4.640000
max      6727.000000
Name: dltr, dtype: float64


Descriptive statistics for dlcch in the year 1980:
count    338.000000
mean      -1.075284
std       44.705108
min     -169.436000
25%      -11.540000
50%        0.000000
75%        5.095000
max      290.000000
Name: dlcch, dtype: float64


Descriptive statistics for fiao in the year 1980:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: fiao, dtype: float64


Descriptive statistics for fincf in the year 1980:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: fincf, dtype: float64


Descriptive statistics for exre in the year 1980:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: exre, dtype: float64


Descriptive statistics for chech in the year 1980:
count    694.000000
mean       2.844235
std       50.449974
min     -375.494000
25%       -0.620000
50%        0.017000
75%        1.222750
max      794.402000
Name: chech, dtype: float64


Descriptive statistics for fsrco in the year 1980:
count    5516.000000
mean        9.222979
std        97.155820
min     -1367.100000
25%         0.000000
50%         0.015000
75%         0.604000
max      3970.894000
Name: fsrco, dtype: float64


Descriptive statistics for fuseo in the year 1980:
count    5517.000000
mean       10.901255
std       113.246868
min      -530.000000
25%         0.000000
50%         0.055000
75%         0.849000
max      5173.000000
Name: fuseo, dtype: float64


Descriptive statistics for wcapc in the year 1980:
count    5096.000000
mean        3.894176
std        94.729391
min     -3539.901000
```

```
25%        -0.389500
50%         0.334500
75%         3.307250
max      3953.000000
Name: wcapc, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 1980:
count    335.000000
mean       0.019276
std        0.084654
min       -0.495654
25%       -0.020927
50%        0.007689
75%        0.059650
max        0.808301
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 1980:
count    6279.000000
mean        0.380991
std         3.795273
min      -239.666667
25%         0.200718
50%         0.408342
75%         0.598211
max        67.541284
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 1980:
count    5392.000000
mean            -inf
std              NaN
min             -inf
25%         0.061310
50%         0.242592
75%         0.409210
max         0.983269
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 1980:
count    6303.000000
mean            -inf
std              NaN
min             -inf
```

```
25%         0.027478
50%         0.168887
75%         0.348189
max         0.955771
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 1980:
count    6100.000000
mean        0.059850
std         0.784485
min       -56.000000
25%         0.033321
50%         0.092140
75%         0.148166
max         3.758635
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 1980:
count    4510.000000
mean             inf
std              NaN
min         0.000000
25%         0.363748
50%         0.893823
75%         2.361555
max              inf
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 1980:
count    6100.000000
mean        1.352999
std         1.097766
min         0.000000
25%         0.536128
50%         1.257119
75%         1.810175
max        14.012832
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 1980:
count    3965.000000
mean             inf
std              NaN
min       -18.920079
```

```
25%         2.379986
50%         3.431358
75%         4.818075
max              inf
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 1980:
count    5711.000000
mean        1.303649
std         1.634661
min         0.000000
25%         0.138418
50%         0.696974
75%         2.039132
max        19.825945
Name: sale_to_at_avg, dtype: float64


Descriptive statistics for dltis_at in the year 1980:
count    5727.000000
mean        0.073627
std         0.593097
min        -2.022866
25%         0.000000
50%         0.017126
75%         0.082563
max        43.500000
Name: dltis_at, dtype: float64


Descriptive statistics for dlc_at in the year 1980:
count    6288.000000
mean             inf
std              NaN
min         0.000000
25%         0.013053
50%         0.043162
75%         0.115782
max              inf
Name: dlc_at, dtype: float64


Descriptive statistics for dltt_at in the year 1980:
count    6294.000000
mean        0.225279
std         0.685057
min         0.000000
```

```
25%         0.037518
50%         0.173029
75%         0.317253
max        43.500000
Name: dltt_at, dtype: float64


Descriptive statistics for ppent_at in the year 1980:
count    6301.000000
mean        0.339859
std         0.261594
min         0.000000
25%         0.128927
50%         0.288712
75%         0.512868
max         1.000000
Name: ppent_at, dtype: float64


Descriptive statistics for mkvalt_at in the year 1980:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: mkvalt_at, dtype: float64


Descriptive statistics for ni_at in the year 1980:
count    6114.000000
mean            -inf
std              NaN
min             -inf
25%         0.011085
50%         0.046383
75%         0.085254
max         4.871571
Name: ni_at, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 1980:
count    335.000000
mean       0.019276
std        0.084654
min       -0.495654
```

```
25%       -0.020927
50%        0.007689
75%        0.059650
max        0.808301
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 1980:
count    6279.000000
mean        0.380991
std         3.795273
min      -239.666667
25%         0.200718
50%         0.408342
75%         0.598211
max        67.541284
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 1980:
count    5392.000000
mean            -inf
std              NaN
min             -inf
25%         0.061310
50%         0.242592
75%         0.409210
max         0.983269
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 1980:
count    6303.000000
mean            -inf
std              NaN
min             -inf
25%         0.027478
50%         0.168887
75%         0.348189
max         0.955771
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 1980:
count    6100.000000
mean        0.059850
std         0.784485
min       -56.000000
```

```
25%         0.033321
50%         0.092140
75%         0.148166
max         3.758635
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 1980:
count    4510.000000
mean             inf
std              NaN
min         0.000000
25%         0.363748
50%         0.893823
75%         2.361555
max              inf
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 1980:
count    6100.000000
mean        1.352999
std         1.097766
min         0.000000
25%         0.536128
50%         1.257119
75%         1.810175
max        14.012832
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 1980:
count    3965.000000
mean             inf
std              NaN
min       -18.920079
25%         2.379986
50%         3.431358
75%         4.818075
max              inf
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 1980:
count    5711.000000
mean        1.303649
std         1.634661
min         0.000000
```

```
25%          0.138418
50%          0.696974
75%          2.039132
max         19.825945
Name: sale_to_at_avg, dtype: float64
```

[104]:

	gvkey	datadate	fyear	indfmt	consol	popsrc	datafmt	tic	\
62990	4075	1980-01-31	1979	INDL	C	D	STD	1198B	
137243	7618	1980-01-31	1979	INDL	C	D	STD	MPH.2	
11772	1573	1980-01-31	1979	INDL	C	D	STD	ASC.1	
7024	1348	1980-01-31	1979	INDL	C	D	STD	3044B	
214418	11310	1980-01-31	1979	INDL	C	D	STD	ZRBA	
…	…	…	…	…	…	…	…	…	
98120	5735	1980-12-31	1980	INDL	C	D	STD	HBC1	
98062	5732	1980-12-31	1980	INDL	C	D	STD	HOV.	
97993	5727	1980-12-31	1980	INDL	C	D	STD	1595B	
99419	5792	1980-12-31	1980	INDL	C	D	STD	HBP	
471348	145348	1980-12-31	1980	INDL	C	D	STD	PPL2	

	cusip	conm	…	net_debt_issued_ratio	\
62990	261794101	DREXEL INDUSTRIES INC-PA	…	NaN	
137243	626643100	MURPHY (G.C.) CO	…	NaN	
11772	030096101	AMERICAN STORES CO	…	NaN	
7024	021492103	ALTIUS CORP	…	NaN	
214418	989131107	ZAREBA SYSTEMS INC	…	NaN	
…	…	…	…	…	
98120	404280000	HSBC FINANCE CORP	…	NaN	
98062	441776101	HOUSE OF VISION INC	…	NaN	
97993	441722105	HOUSE OF ADLER INC	…	NaN	
99419	448451104	HUTTIG BUILDING PRODUCTS INC	…	NaN	
471348	69399Y000	PPL ELECTRIC UTILITIES CORP	…	0.052975	

	book_leverage	wc_ta	re_ta	ebit_ta	mv_tl	sales_ta	\
62990	0.752989	-0.041935	0.009160	0.137541	NaN	1.581365	
137243	0.385212	0.360258	0.418009	0.094935	0.342679	2.620976	
11772	0.573757	0.134230	0.094692	0.081096	0.319970	3.150198	
7024	0.169540	0.393548	0.301075	0.356989	5.778409	3.410753	
214418	0.098039	0.672659	0.496893	0.118953	2.600248	1.839547	
…	…	…	…	…	…	…	
98120	0.744942	NaN	0.189187	0.085118	0.177378	0.172538	
98062	0.458985	0.379237	0.200151	-0.263118	0.470461	1.893318	
97993	0.193492	0.407009	0.287907	0.083142	NaN	1.262254	
99419	0.192459	0.489504	0.588149	0.176543	NaN	2.557591	
471348	0.483828	0.008484	0.088666	0.052183	0.342862	0.205915	

```
         z_score at_rolling_avg sale_to_at_avg
62990        NaN            NaN            NaN
137243  4.131183       147.8815       5.118118
11772   3.871941       745.3550       5.079902
7024    8.915519       601.1995       0.002638
214418  5.276684         2.4855       3.334943
…            …              …              …
98120        NaN      4997.3130       0.192023
98062   2.023668      2787.4665       0.008996
97993        NaN         8.8055       0.625859
99419        NaN        32.8130       4.774876
471348  0.716090      2180.6670       0.406046

[6752 rows x 998 columns]
```

0.5 Descriptive Stats of the Finaincal Ratios for the Year 1985

[105]:
```
describe_yearly_stats(compustat_copy, 1985)
```

```
Descriptive statistics for ch in the year 1985:
count    4990.000000
mean       69.655629
std       378.696841
min       -45.622000
25%         0.126000
50%         0.742000
75%         4.960000
max      7636.746000
Name: ch, dtype: float64


Descriptive statistics for ivst in the year 1985:
count     4956.000000
mean       143.976442
std       1129.136993
min         -4.903000
25%          0.000000
50%          0.250000
75%         10.536750
max      27879.804000
Name: ivst, dtype: float64


Descriptive statistics for rect in the year 1985:
count      6528.000000
mean        311.804030
std        2751.571681
min           0.000000
```

```
25%           0.982000
50%           6.625000
75%          43.307000
max      115264.000000
Name: rect, dtype: float64


Descriptive statistics for invt in the year 1985:
count     6893.000000
mean        90.069460
std        783.609474
min          0.000000
25%          0.138000
50%          3.300000
75%         25.435000
max      49780.000000
Name: invt, dtype: float64


Descriptive statistics for aco in the year 1985:
count    6766.000000
mean       13.933380
std        81.045437
min       -30.356000
25%         0.041000
50%         0.345000
75%         2.772500
max      2558.000000
Name: aco, dtype: float64


Descriptive statistics for act in the year 1985:
count     6254.000000
mean       223.075360
std       1062.181498
min          0.000000
25%          3.143500
50%         16.753000
75%         90.493750
max      26350.000000
Name: act, dtype: float64


Descriptive statistics for ppent in the year 1985:
count     7337.000000
mean       352.939443
std       1852.095092
min          0.000000
```

```
25%          1.497000
50%         10.949000
75%         75.721000
max      48262.000000
Name: ppent, dtype: float64


Descriptive statistics for ivaeq in the year 1985:
count    6491.000000
mean       22.383554
std       178.358684
min       -34.300000
25%         0.000000
50%         0.000000
75%         0.000000
max      5718.500000
Name: ivaeq, dtype: float64


Descriptive statistics for ivao in the year 1985:
count     6614.000000
mean       192.630574
std       2055.945304
min          0.000000
25%          0.000000
50%          0.000000
75%          3.210250
max      96140.000000
Name: ivao, dtype: float64


Descriptive statistics for intan in the year 1985:
count    6286.000000
mean       18.459447
std       157.273858
min        -4.382000
25%         0.000000
50%         0.000000
75%         0.813500
max      5945.250000
Name: intan, dtype: float64


Descriptive statistics for ao in the year 1985:
count     7302.000000
mean        64.129665
std        421.767698
min        -11.360000
```

```
25%          0.070000
50%          0.724000
75%          8.100250
max      15230.000000
Name: ao, dtype: float64


Descriptive statistics for at in the year 1985:
count      7383.000000
mean       1524.160559
std        7040.738349
min           0.000000
25%           8.296000
50%          47.687000
75%         378.880500
max      173597.000000
Name: at, dtype: float64


Descriptive statistics for dlc in the year 1985:
count     7305.000000
mean       200.025346
std       1782.687994
min          0.000000
25%          0.200000
50%          1.696000
75%         13.075000
max      70780.000000
Name: dlc, dtype: float64


Descriptive statistics for ap in the year 1985:
count      7034.000000
mean        305.570994
std        2855.202502
min          -0.001000
25%           0.489250
50%           3.000000
75%          20.597000
max      104959.000000
Name: ap, dtype: float64


Descriptive statistics for txp in the year 1985:
count    6758.000000
mean       12.938730
std       106.123283
min       -26.012000
```

```
25%         0.000000
50%         0.003000
75%         1.183000
max      3218.000000
Name: txp, dtype: float64


Descriptive statistics for lco in the year 1985:
count     6766.000000
mean        55.613322
std        314.660911
min          0.000000
25%          0.278250
50%          2.001500
75%         13.604250
max      12077.947000
Name: lco, dtype: float64


Descriptive statistics for lct in the year 1985:
count     6470.000000
mean       177.228355
std       1147.195651
min          0.000000
25%          1.656250
50%          8.551500
75%         47.936000
max      52541.016000
Name: lct, dtype: float64


Descriptive statistics for dltt in the year 1985:
count     7321.000000
mean       215.598625
std       1368.475288
min          0.000000
25%          0.325000
50%          5.644000
75%         52.124000
max      61957.016000
Name: dltt, dtype: float64


Descriptive statistics for lo in the year 1985:
count     7326.000000
mean       105.645369
std       1074.109076
min          0.000000
```

```
25%          0.000000
50%          0.036000
75%          5.444750
max      52475.508000
Name: lo, dtype: float64


Descriptive statistics for txditc in the year 1985:
count     6714.000000
mean        54.969489
std        345.449533
min          0.000000
25%          0.000000
50%          0.090500
75%          4.201000
max      11042.000000
Name: txditc, dtype: float64


Descriptive statistics for mib in the year 1985:
count    6861.000000
mean        5.327536
std        77.295463
min        -2.399000
25%         0.000000
50%         0.000000
75%         0.000000
max      4895.000000
Name: mib, dtype: float64


Descriptive statistics for lt in the year 1985:
count      7324.000000
mean       1200.568330
std        6375.650526
min           0.000000
25%           3.543000
50%          22.900000
75%         210.041000
max      165792.000000
Name: lt, dtype: float64


Descriptive statistics for pstk in the year 1985:
count    7376.000000
mean       15.612943
std        91.549357
min         0.000000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max      2355.000000
Name: pstk, dtype: float64


Descriptive statistics for ceq in the year 1985:
count     7374.000000
mean       276.622573
std       1151.019768
min      -1185.130000
25%          2.816000
50%         18.704000
75%        116.319000
max      31990.008000
Name: ceq, dtype: float64


Descriptive statistics for teq in the year 1985:
count      319.000000
mean       750.983702
std       2435.688806
min       -183.132000
25%         19.312500
50%         87.305000
75%        515.197000
max      29096.000000
Name: teq, dtype: float64


Descriptive statistics for sale in the year 1985:
count     7025.000000
mean       677.949153
std       3335.995573
min          0.000000
25%          6.359000
50%         42.049000
75%        268.160000
max      96371.563000
Name: sale, dtype: float64


Descriptive statistics for cogs in the year 1985:
count     7022.000000
mean       486.117342
std       2565.069004
min          0.000000
```

```
25%          3.713000
50%         26.758500
75%        179.915750
max      81654.500000
Name: cogs, dtype: float64


Descriptive statistics for xsga in the year 1985:
count     5579.000000
mean       105.800767
std        501.200959
min          0.000000
25%          1.925500
50%          8.161000
75%         38.617500
max      17723.004000
Name: xsga, dtype: float64


Descriptive statistics for oibdp in the year 1985:
count     6974.000000
mean       107.096537
std        555.827219
min       -854.900000
25%          0.073250
50%          3.958000
75%         30.787250
max      14397.000000
Name: oibdp, dtype: float64


Descriptive statistics for dp in the year 1985:
count    6769.000000
mean       33.564884
std       198.809825
min         0.000000
25%         0.224000
50%         1.247000
75%         7.615000
max      6208.500000
Name: dp, dtype: float64


Descriptive statistics for oiadp in the year 1985:
count     7022.000000
mean        75.508469
std        394.211620
min       -854.903000
```

```
25%         -0.213500
50%          2.261500
75%         21.510500
max      11230.004000
Name: oiadp, dtype: float64


Descriptive statistics for xint in the year 1985:
count     6912.000000
mean        51.566717
std        366.118665
min          0.000000
25%          0.132000
50%          1.002000
75%          8.217000
max      14028.004000
Name: xint, dtype: float64


Descriptive statistics for nopi in the year 1985:
count    7021.000000
mean        8.703500
std        96.203481
min     -2016.000000
25%         0.007000
50%         0.235000
75%         1.953000
max      3574.000000
Name: nopi, dtype: float64


Descriptive statistics for spi in the year 1985:
count    6567.000000
mean       -4.482086
std        61.114327
min     -2303.000000
25%         0.000000
50%         0.000000
75%         0.000000
max       414.000000
Name: spi, dtype: float64


Descriptive statistics for pi in the year 1985:
count     7304.000000
mean        54.894415
std        337.115303
min      -1013.000000
```

```
25%         -0.445250
50%          1.730000
75%         18.595750
max      11619.000000
Name: pi, dtype: float64


Descriptive statistics for txt in the year 1985:
count    7359.000000
mean       23.079053
std       170.592353
min      -441.000000
25%         0.000000
50%         0.474000
75%         5.962000
max      6620.000000
Name: txt, dtype: float64


Descriptive statistics for mii in the year 1985:
count    6624.000000
mean        0.563124
std        11.911272
min       -36.577000
25%         0.000000
50%         0.000000
75%         0.000000
max       823.000000
Name: mii, dtype: float64


Descriptive statistics for ib in the year 1985:
count    7361.000000
mean       31.356767
std       174.674118
min      -854.600000
25%        -0.373000
50%         1.148000
75%        12.083000
max      6555.000000
Name: ib, dtype: float64


Descriptive statistics for dvp in the year 1985:
count    7356.000000
mean        1.445213
std         8.254974
min         0.000000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max       225.085000
Name: dvp, dtype: float64


Descriptive statistics for cstke in the year 1985:
count    7360.000000
mean        0.069238
std         1.211551
min        -2.108000
25%         0.000000
50%         0.000000
75%         0.000000
max        79.000000
Name: cstke, dtype: float64


Descriptive statistics for xido in the year 1985:
count    7360.000000
mean       -0.753574
std        31.605537
min     -1337.300000
25%         0.000000
50%         0.000000
75%         0.000000
max       342.867000
Name: xido, dtype: float64


Descriptive statistics for ni in the year 1985:
count    7053.000000
mean       28.667225
std       175.944362
min     -1219.139000
25%        -0.451000
50%         1.034000
75%         9.940000
max      6555.000000
Name: ni, dtype: float64


Descriptive statistics for ibc in the year 1985:
count    6743.000000
mean       28.057475
std       177.302077
min      -854.599000
```

```
25%        -0.468000
50%         0.816000
75%         8.190500
max      6555.000000
Name: ibc, dtype: float64


Descriptive statistics for dpc in the year 1985:
count    6666.000000
mean       34.605715
std       201.777664
min         0.000000
25%         0.251000
50%         1.321000
75%         7.950750
max      6208.500000
Name: dpc, dtype: float64


Descriptive statistics for xidoc in the year 1985:
count    6387.000000
mean        0.904342
std        26.970955
min     -1207.000000
25%         0.000000
50%         0.000000
75%         0.000000
max       852.000000
Name: xidoc, dtype: float64


Descriptive statistics for txdc in the year 1985:
count    6541.000000
mean        4.701141
std        41.725169
min      -945.000000
25%         0.000000
50%         0.000000
75%         0.447000
max      1174.000000
Name: txdc, dtype: float64


Descriptive statistics for esubc in the year 1985:
count    5595.000000
mean       -0.566699
std        13.517787
min      -551.400000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max       239.000000
Name: esubc, dtype: float64


Descriptive statistics for sppiv in the year 1985:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: sppiv, dtype: float64


Descriptive statistics for fopo in the year 1985:
count    6741.000000
mean       12.643209
std       133.340508
min      -471.700000
25%         0.000000
50%         0.000000
75%         0.267000
max      4267.000000
Name: fopo, dtype: float64


Descriptive statistics for fopt in the year 1985:
count     6776.000000
mean        79.934830
std        431.349823
min       -251.280000
25%          0.014000
50%          2.628500
75%         19.325750
max      12778.000000
Name: fopt, dtype: float64


Descriptive statistics for recch in the year 1985:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: recch, dtype: float64


Descriptive statistics for invch in the year 1985:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: invch, dtype: float64


Descriptive statistics for apalch in the year 1985:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: apalch, dtype: float64


Descriptive statistics for txach in the year 1985:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: txach, dtype: float64


Descriptive statistics for aoloch in the year 1985:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: aoloch, dtype: float64


Descriptive statistics for oancf in the year 1985:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: oancf, dtype: float64


Descriptive statistics for ivch in the year 1985:
count     6109.000000
mean        46.598632
std        628.690803
min       -261.000000
25%          0.000000
50%          0.000000
75%          0.111000
max      21072.004000
Name: ivch, dtype: float64


Descriptive statistics for siv in the year 1985:
count     6057.000000
mean        16.914219
std        296.801232
min       -261.315000
25%          0.000000
50%          0.000000
75%          0.000000
max      14932.102000
Name: siv, dtype: float64


Descriptive statistics for capx in the year 1985:
count    6652.000000
mean       59.805074
std       331.935397
min         0.000000
```

```
25%         0.269000
50%         2.028500
75%        14.085750
max      9174.199000
Name: capx, dtype: float64


Descriptive statistics for sppe in the year 1985:
count    5003.000000
mean        5.492676
std        46.385335
min       -68.162000
25%         0.000000
50%         0.017000
75%         0.747000
max      1683.000000
Name: sppe, dtype: float64


Descriptive statistics for aqc in the year 1985:
count    6188.000000
mean       13.733191
std       157.439772
min      -270.104000
25%         0.000000
50%         0.000000
75%         0.000000
max      6193.398000
Name: aqc, dtype: float64


Descriptive statistics for ivstch in the year 1985:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: ivstch, dtype: float64


Descriptive statistics for ivaco in the year 1985:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: ivaco, dtype: float64


Descriptive statistics for ivncf in the year 1985:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: ivncf, dtype: float64


Descriptive statistics for sstk in the year 1985:
count    6590.000000
mean        8.260715
std        56.889158
min      -145.000000
25%         0.000000
50%         0.030000
75%         1.278500
max      2755.500000
Name: sstk, dtype: float64


Descriptive statistics for prstkc in the year 1985:
count    6621.000000
mean        8.010239
std       109.713915
min        -1.759000
25%         0.000000
50%         0.000000
75%         0.025000
max      4972.000000
Name: prstkc, dtype: float64


Descriptive statistics for dv in the year 1985:
count    6730.000000
mean       16.603846
std        92.304593
min         0.000000
```

```
25%         0.000000
50%         0.000000
75%         2.291750
max      2703.000000
Name: dv, dtype: float64


Descriptive statistics for dltis in the year 1985:
count     6543.000000
mean        46.798487
std        442.100253
min       -770.000000
25%          0.000000
50%          0.409000
75%          7.492000
max      29558.008000
Name: dltis, dtype: float64


Descriptive statistics for dltr in the year 1985:
count     6473.000000
mean        30.946749
std        297.659232
min       -593.569000
25%          0.027000
50%          0.641000
75%          5.868000
max      19449.004000
Name: dltr, dtype: float64


Descriptive statistics for dlcch in the year 1985:
count     5720.000000
mean        16.286593
std        436.807135
min      -2016.010000
25%         -0.756250
50%          0.000000
75%          0.335000
max      22353.000000
Name: dlcch, dtype: float64


Descriptive statistics for fiao in the year 1985:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: fiao, dtype: float64


Descriptive statistics for fincf in the year 1985:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: fincf, dtype: float64


Descriptive statistics for exre in the year 1985:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: exre, dtype: float64


Descriptive statistics for chech in the year 1985:
count    6378.000000
mean        3.079245
std       102.203574
min     -3453.000000
25%        -0.818000
50%         0.007000
75%         1.348000
max      3375.000000
Name: chech, dtype: float64


Descriptive statistics for fsrco in the year 1985:
count     6458.000000
mean        23.836237
std        265.610041
min      -1945.000000
```

```
25%          0.000000
50%          0.051000
75%          1.377250
max      10730.000000
Name: fsrco, dtype: float64


Descriptive statistics for fuseo in the year 1985:
count     6458.000000
mean        20.466110
std        268.719213
min      -1152.000000
25%          0.000000
50%          0.118000
75%          1.542250
max      15862.000000
Name: fuseo, dtype: float64


Descriptive statistics for wcapc in the year 1985:
count    5846.000000
mean       -0.789993
std        76.852899
min     -3101.001000
25%        -1.223250
50%         0.080000
75%         1.972250
max      1297.000000
Name: wcapc, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 1985:
count    5416.000000
mean             NaN
std              NaN
min             -inf
25%        -0.039361
50%         0.000000
75%         0.054206
max              inf
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 1985:
count    7293.000000
mean        0.429621
std         6.752404
min       -84.736318
```

```
25%         0.147273
50%         0.379068
75%         0.600003
max       542.258065
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 1985:
count    6245.000000
mean            -inf
std              NaN
min             -inf
25%         0.036069
50%         0.220023
75%         0.418607
max         0.992908
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 1985:
count    7276.000000
mean            -inf
std              NaN
min             -inf
25%        -0.104260
50%         0.095981
75%         0.282079
max         3.539222
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 1985:
count    7021.000000
mean            -inf
std              NaN
min             -inf
25%        -0.031989
50%         0.060744
75%         0.119574
max         9.000000
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 1985:
count    5628.000000
mean             inf
std              NaN
min         0.000000
```

```
25%         0.586179
50%         1.373719
75%         3.775998
max              inf
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 1985:
count    7020.000000
mean             inf
std              NaN
min         0.000000
25%         0.420733
50%         0.985886
75%         1.586198
max              inf
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 1985:
count    4966.000000
mean             inf
std              NaN
min      -528.512375
25%         1.654002
50%         3.067904
75%         4.981790
max              inf
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 1985:
count    6560.000000
mean        1.131629
std         1.495677
min         0.000000
25%         0.104555
50%         0.561137
75%         1.721407
max        18.368191
Name: sale_to_at_avg, dtype: float64


Descriptive statistics for dltis_at in the year 1985:
count    6539.000000
mean        0.090518
std         0.260422
min        -0.472177
```

```
25%         0.000000
50%         0.015912
75%         0.096412
max         7.849315
Name: dltis_at, dtype: float64


Descriptive statistics for dlc_at in the year 1985:
count    7303.000000
mean             inf
std              NaN
min         0.000000
25%         0.008990
50%         0.040186
75%         0.125669
max              inf
Name: dlc_at, dtype: float64


Descriptive statistics for dltt_at in the year 1985:
count    7317.000000
mean        0.205917
std         0.909133
min         0.000000
25%         0.020450
50%         0.137552
75%         0.293565
max        74.000000
Name: dltt_at, dtype: float64


Descriptive statistics for ppent_at in the year 1985:
count    7333.000000
mean        0.328549
std         0.265622
min         0.000000
25%         0.110149
50%         0.267995
75%         0.507117
max         1.000000
Name: ppent_at, dtype: float64


Descriptive statistics for mkvalt_at in the year 1985:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: mkvalt_at, dtype: float64


Descriptive statistics for ni_at in the year 1985:
count    7051.000000
mean             NaN
std              NaN
min             -inf
25%        -0.050052
50%         0.028158
75%         0.067318
max              inf
Name: ni_at, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 1985:
count    5416.000000
mean             NaN
std              NaN
min             -inf
25%        -0.039361
50%         0.000000
75%         0.054206
max              inf
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 1985:
count    7293.000000
mean        0.429621
std         6.752404
min       -84.736318
25%         0.147273
50%         0.379068
75%         0.600003
max       542.258065
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 1985:
count    6245.000000
mean            -inf
std              NaN
min             -inf
```

```
25%         0.036069
50%         0.220023
75%         0.418607
max         0.992908
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 1985:
count    7276.000000
mean            -inf
std              NaN
min             -inf
25%        -0.104260
50%         0.095981
75%         0.282079
max         3.539222
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 1985:
count    7021.000000
mean            -inf
std              NaN
min             -inf
25%        -0.031989
50%         0.060744
75%         0.119574
max         9.000000
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 1985:
count    5628.000000
mean             inf
std              NaN
min         0.000000
25%         0.586179
50%         1.373719
75%         3.775998
max              inf
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 1985:
count    7020.000000
mean             inf
std              NaN
min         0.000000
```

```
25%         0.420733
50%         0.985886
75%         1.586198
max              inf
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 1985:
count    4966.000000
mean             inf
std              NaN
min      -528.512375
25%         1.654002
50%         3.067904
75%         4.981790
max              inf
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 1985:
count    6560.000000
mean        1.131629
std         1.495677
min         0.000000
25%         0.104555
50%         0.561137
75%         1.721407
max        18.368191
Name: sale_to_at_avg, dtype: float64
```

[105]:

	gvkey	datadate	fyear	indfmt	consol	popsrc	datafmt	tic	\
39943	2951	1985-01-31	1984	INDL	C	D	STD	DLPX	
137017	7606	1985-01-31	1984	INDL	C	D	STD	MULT.1	
247461	13647	1985-01-31	1984	INDL	C	D	STD	6354C	
156076	8475	1985-01-31	1984	INDL	C	D	STD	PBY.1	
227622	12167	1985-01-31	1984	INDL	C	D	STD	PADCE	
…	…	…	…	…	…	…	…	…	
127578	7169	1985-12-31	1985	INDL	C	D	STD	2282B	
127461	7163	1985-12-31	1985	INDL	C	D	STD	SPGI	
127353	7161	1985-12-31	1985	INDL	C	D	STD	MGRC	
128992	7241	1985-12-31	1985	INDL	C	D	STD	CVS	
471353	145348	1985-12-31	1985	INDL	C	D	STD	PPL2	

	cusip	conm	…	net_debt_issued_ratio	\
39943	24712X106	DELPHAX TECHNOLOGIES INC	…	0.319046	

```
137017  625430103          MULTI SOLUTIONS INC  …              0.000000
247461  196902100          COLORADO PRIME CORP  …                   NaN
156076  713278109    PEP BOYS-MANNY MOE & JACK  …              0.055872
227622  848906103         SPORTS MARKETING INC  …                   NaN
…          …                              …  …                    …
127578  581331105    MCKEE INCOME REALTY TRUST  …                   NaN
127461  78409V104               S&P GLOBAL INC  …              0.001119
127353  580589109             MCGRATH RENTCORP  …                   NaN
128992  126650100              CVS HEALTH CORP  …                   NaN
471353  69399Y000  PPL ELECTRIC UTILITIES CORP  …              0.000603

       book_leverage     wc_ta     re_ta   ebit_ta       mv_tl  sales_ta  \
39943       0.532232  0.187804 -1.785020 -0.189350    7.392073  1.404993
137017      0.000000  0.674449 -1.044747 -0.809339  150.118924  0.612192
247461           NaN       NaN       NaN       NaN         NaN       NaN
156076      0.198865  0.089521  0.437259  0.171810    3.448949  1.808652
227622           NaN       NaN       NaN       NaN         NaN       NaN
…              …       …       …       …          …       …
127578      0.940872       NaN -0.316106 -0.010607         NaN  0.239474
127461      0.007589  0.217662  0.538642  0.205413    4.866830  1.170575
127353      0.522657       NaN  0.213389  0.173847    2.394934  0.478493
128992      0.094753  0.393890  0.561619  0.246570    4.413815  2.642902
471353      0.522691 -0.029072  0.085912  0.110492    0.517757  0.283750

          z_score at_rolling_avg sale_to_at_avg
39943    2.927668            NaN            NaN
137017  87.353300         3.0340       0.311140
247461        NaN            NaN            NaN
156076   5.146497            NaN            NaN
227622        NaN            NaN            NaN
…          …             …             …
127578        NaN       915.1135       0.004835
127461   5.772123       646.1945       2.307677
127353        NaN       651.4970       0.021360
128992   7.337376       917.8780       5.202063
471353   1.041581      4386.1550       0.450623

[8460 rows x 998 columns]
```

0.6 Descriptive Stats of the Finaincal Ratios for the Year 1990

[106]:
```
describe_yearly_stats(compustat_copy, 1990)
```

```
Descriptive statistics for ch in the year 1990:
count     7244.000000
mean       112.766768
std        693.429672
min        -12.510000
```

```
25%          0.375000
50%          2.532000
75%         18.488500
max      19143.812000
Name: ch, dtype: float64


Descriptive statistics for ivst in the year 1990:
count     7164.000000
mean       178.922428
std       2279.736186
min          0.000000
25%          0.000000
50%          0.000000
75%          0.708000
max      93980.985000
Name: ivst, dtype: float64


Descriptive statistics for rect in the year 1990:
count      7157.000000
mean        662.065856
std        5623.197468
min           0.000000
25%           1.212000
50%           9.496000
75%          62.523000
max      188637.000000
Name: rect, dtype: float64


Descriptive statistics for invt in the year 1990:
count     7047.000000
mean       125.398295
std        952.727781
min          0.000000
25%          0.100000
50%          3.571000
75%         33.677000
max      54156.000000
Name: invt, dtype: float64


Descriptive statistics for aco in the year 1990:
count     6881.000000
mean        25.106519
std        202.521297
min         -2.557000
```

```
25%          0.054000
50%          0.583000
75%          4.397000
max      10228.714000
Name: aco, dtype: float64


Descriptive statistics for act in the year 1990:
count     6219.000000
mean       323.857407
std       1704.677888
min          0.000000
25%          3.705000
50%         22.177000
75%        119.827000
max      44788.813000
Name: act, dtype: float64


Descriptive statistics for ppent in the year 1990:
count     7499.000000
mean       475.817091
std       2355.119923
min          0.000000
25%          1.458500
50%         14.272000
75%        112.315000
max      62688.000000
Name: ppent, dtype: float64


Descriptive statistics for ivaeq in the year 1990:
count    6666.000000
mean       26.314128
std       211.697991
min       -11.300000
25%         0.000000
50%         0.000000
75%         0.000000
max      7113.000000
Name: ivaeq, dtype: float64


Descriptive statistics for ivao in the year 1990:
count     6796.000000
mean       227.469200
std       1820.529823
min          0.000000
```

```
25%          0.000000
50%          0.000000
75%          2.249750
max      47101.000000
Name: ivao, dtype: float64


Descriptive statistics for intan in the year 1990:
count     6284.000000
mean        68.683052
std        561.958876
min          0.000000
25%          0.000000
50%          0.000000
75%          5.408500
max      23138.000000
Name: intan, dtype: float64


Descriptive statistics for ao in the year 1990:
count     7537.000000
mean       159.631143
std       1020.770185
min         -0.579000
25%          0.135000
50%          1.543000
75%         20.360000
max      28084.699000
Name: ao, dtype: float64


Descriptive statistics for at in the year 1990:
count      7707.000000
mean       2523.659707
std       12769.881333
min           0.000000
25%          10.795000
50%          76.424000
75%         641.593000
max      352330.000000
Name: at, dtype: float64


Descriptive statistics for dlc in the year 1990:
count     7511.000000
mean       356.953579
std       2956.246329
min         -0.069000
```

```
25%          0.190000
50%          2.061000
75%         22.688000
max      87191.000000
Name: dlc, dtype: float64


Descriptive statistics for ap in the year 1990:
count      7234.000000
mean        538.327473
std        5849.651070
min           0.000000
25%           0.619250
50%           4.090000
75%          31.200000
max      280064.000000
Name: ap, dtype: float64


Descriptive statistics for txp in the year 1990:
count    6876.000000
mean       17.967358
std       128.833046
min      -154.600000
25%         0.000000
50%         0.000000
75%         1.298250
max      3959.600000
Name: txp, dtype: float64


Descriptive statistics for lco in the year 1990:
count     6880.000000
mean        90.737882
std        547.365241
min          0.000000
25%          0.401000
50%          2.879000
75%         21.398250
max      23471.100000
Name: lco, dtype: float64


Descriptive statistics for lct in the year 1990:
count     6395.000000
mean       277.564679
std       1905.087521
min          0.000000
```

```
25%          2.155000
50%         11.917000
75%         69.891500
max      93022.000000
Name: lct, dtype: float64


Descriptive statistics for dltt in the year 1990:
count     7548.000000
mean       371.500616
std       2080.580193
min          0.000000
25%          0.258000
50%          7.346000
75%         97.647500
max      84950.000000
Name: dltt, dtype: float64


Descriptive statistics for lo in the year 1990:
count     7650.000000
mean       223.839758
std       1922.884093
min       -750.000000
25%          0.000000
50%          0.310500
75%         14.408500
max      81058.375000
Name: lo, dtype: float64


Descriptive statistics for txditc in the year 1990:
count     6873.000000
mean        67.195494
std        398.698964
min        -14.000000
25%          0.000000
50%          0.000000
75%          4.067000
max      12568.000000
Name: txditc, dtype: float64


Descriptive statistics for mib in the year 1990:
count    6961.000000
mean       12.854602
std       123.079803
min        -1.928000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max      3658.227000
Name: mib, dtype: float64


Descriptive statistics for lt in the year 1990:
count      7665.000000
mean       2092.672252
std       11763.309627
min           0.000000
25%           4.519000
50%          37.564000
75%         421.268000
max      342051.000000
Name: lt, dtype: float64


Descriptive statistics for pstk in the year 1990:
count    7684.000000
mean       16.055236
std        92.744441
min        -1.002000
25%         0.000000
50%         0.000000
75%         0.000000
max      2341.700000
Name: pstk, dtype: float64


Descriptive statistics for ceq in the year 1990:
count     7684.000000
mean       388.072005
std       1635.696637
min      -4854.100000
25%          2.477750
50%         23.074500
75%        147.982250
max      42832.000000
Name: ceq, dtype: float64


Descriptive statistics for teq in the year 1990:
count      278.000000
mean      1324.071626
std       3543.846701
min       -485.424000
```

```
25%         43.804250
50%        176.210500
75%        946.520000
max      33055.000000
Name: teq, dtype: float64


Descriptive statistics for sale in the year 1990:
count      7203.000000
mean        994.389767
std        4710.543750
min          -4.839000
25%           8.150000
50%          60.086000
75%         377.417000
max      123537.000000
Name: sale, dtype: float64


Descriptive statistics for cogs in the year 1990:
count      7203.000000
mean        694.755520
std        3564.978433
min          -0.768000
25%           4.754000
50%          37.650000
75%         244.675000
max      119980.000000
Name: cogs, dtype: float64


Descriptive statistics for xsga in the year 1990:
count     5653.000000
mean       176.660091
std        888.870991
min        -97.000000
25%          2.551000
50%         11.483000
75%         58.793000
max      27263.000000
Name: xsga, dtype: float64


Descriptive statistics for oibdp in the year 1990:
count     7070.000000
mean       160.716141
std        779.717401
min       -467.000000
```

```
25%          0.085000
50%          5.400500
75%         45.263250
max      18703.000000
Name: oibdp, dtype: float64


Descriptive statistics for dp in the year 1990:
count    6927.000000
mean       50.753099
std       279.700896
min         0.000000
25%         0.283000
50%         1.898000
75%        12.697500
max      9304.000000
Name: dp, dtype: float64


Descriptive statistics for oiadp in the year 1990:
count     7203.000000
mean       112.675472
std        554.683105
min       -811.000000
25%         -0.203000
50%          3.042000
75%         31.708000
max      12870.000000
Name: oiadp, dtype: float64


Descriptive statistics for xint in the year 1990:
count     7094.000000
mean        96.273719
std        709.066679
min          0.000000
25%          0.157000
50%          1.645500
75%         16.576250
max      23798.000000
Name: xint, dtype: float64


Descriptive statistics for nopi in the year 1990:
count    7202.000000
mean        9.030757
std       148.997310
min     -5939.000000
```

```
25%         0.000000
50%         0.203000
75%         1.971250
max      3744.140000
Name: nopi, dtype: float64


Descriptive statistics for spi in the year 1990:
count    6592.000000
mean       -5.626805
std        80.507884
min     -3314.000000
25%        -0.026000
50%         0.000000
75%         0.000000
max       606.170000
Name: spi, dtype: float64


Descriptive statistics for pi in the year 1990:
count     7525.000000
mean        70.472618
std        404.250698
min      -2379.650000
25%         -0.822000
50%          1.401000
75%         21.430000
max      14286.000000
Name: pi, dtype: float64


Descriptive statistics for txt in the year 1990:
count    7562.00000
mean       27.56867
std       171.14885
min      -349.00000
25%         0.00000
50%         0.40400
75%         6.98575
max      7597.00000
Name: txt, dtype: float64


Descriptive statistics for mii in the year 1990:
count    6657.000000
mean        1.435677
std        15.078014
min      -119.000000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max       464.000000
Name: mii, dtype: float64


Descriptive statistics for ib in the year 1990:
count    7677.000000
mean       40.463713
std       238.497060
min     -2679.845000
25%        -0.804000
50%         0.956000
75%        14.099000
max      6533.000000
Name: ib, dtype: float64


Descriptive statistics for dvp in the year 1990:
count    7560.000000
mean        1.759219
std        11.305620
min        -0.526000
25%         0.000000
50%         0.000000
75%         0.000000
max       559.000000
Name: dvp, dtype: float64


Descriptive statistics for cstke in the year 1990:
count    7563.000000
mean        0.065076
std         2.734272
min      -128.752000
25%         0.000000
50%         0.000000
75%         0.000000
max       178.303000
Name: cstke, dtype: float64


Descriptive statistics for xido in the year 1990:
count    7677.000000
mean        1.081624
std        30.785406
min      -651.000000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max       921.823000
Name: xido, dtype: float64


Descriptive statistics for ni in the year 1990:
count    7280.000000
mean       39.846102
std       243.994505
min     -2510.504000
25%        -0.828250
50%         0.900500
75%        12.437000
max      6533.000000
Name: ni, dtype: float64


Descriptive statistics for ibc in the year 1990:
count    6921.000000
mean       38.570464
std       242.988556
min     -2679.845000
25%        -0.855000
50%         0.662000
75%        10.426000
max      6533.000000
Name: ibc, dtype: float64


Descriptive statistics for dpc in the year 1990:
count    6816.000000
mean       54.028218
std       286.636673
min         0.000000
25%         0.343750
50%         2.182500
75%        14.323750
max      9304.000000
Name: dpc, dtype: float64


Descriptive statistics for xidoc in the year 1990:
count    6886.000000
mean        0.197678
std        21.938562
min      -814.973000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max       921.600000
Name: xidoc, dtype: float64


Descriptive statistics for txdc in the year 1990:
count    6726.000000
mean        0.509959
std        27.870298
min      -871.000000
25%         0.000000
50%         0.000000
75%         0.005000
max       691.000000
Name: txdc, dtype: float64


Descriptive statistics for esubc in the year 1990:
count    6044.000000
mean       -0.491986
std        16.547833
min      -433.380000
25%         0.000000
50%         0.000000
75%         0.000000
max       660.346000
Name: esubc, dtype: float64


Descriptive statistics for sppiv in the year 1990:
count    6832.000000
mean       -2.027039
std        31.691820
min     -1172.896000
25%         0.000000
50%         0.000000
75%         0.000000
max       914.000000
Name: sppiv, dtype: float64


Descriptive statistics for fopo in the year 1990:
count    6921.000000
mean       14.264847
std       133.511011
min     -1582.600000
```

```
25%         0.000000
50%         0.034000
75%         1.308000
max      4027.400000
Name: fopo, dtype: float64


Descriptive statistics for fopt in the year 1990:
count      101.000000
mean       773.473010
std       1503.548424
min       -107.000000
25%          0.015000
50%        129.683000
75%        839.320000
max      10330.090000
Name: fopt, dtype: float64


Descriptive statistics for recch in the year 1990:
count    6231.000000
mean       -9.078383
std       150.941996
min     -8076.000000
25%        -2.241500
50%        -0.125000
75%         0.407000
max      2172.000000
Name: recch, dtype: float64


Descriptive statistics for invch in the year 1990:
count     6164.000000
mean        -6.671698
std        282.774916
min      -7000.860000
25%         -1.084250
50%          0.000000
75%          0.112000
max      18137.000000
Name: invch, dtype: float64


Descriptive statistics for apalch in the year 1990:
count    5394.000000
mean        8.484096
std        91.233138
min      -321.216000
```

```
25%        -0.344750
50%         0.128500
75%         1.671750
max      2824.671000
Name: apalch, dtype: float64


Descriptive statistics for txach in the year 1990:
count    5349.000000
mean        0.131880
std        37.973799
min     -1711.800000
25%         0.000000
50%         0.000000
75%         0.000000
max      1192.000000
Name: txach, dtype: float64


Descriptive statistics for aoloch in the year 1990:
count     6820.000000
mean         7.111921
std        236.955018
min     -12652.000000
25%         -0.695000
50%         -0.006000
75%          0.460250
max       5649.000000
Name: aoloch, dtype: float64


Descriptive statistics for oancf in the year 1990:
count     6829.000000
mean        93.909042
std        498.009508
min      -4679.380000
25%         -0.169000
50%          2.370000
75%         24.028000
max      13634.000000
Name: oancf, dtype: float64


Descriptive statistics for ivch in the year 1990:
count      6577.000000
mean        186.652019
std        2867.709556
min         -65.631000
```

```
25%           0.000000
50%           0.000000
75%           0.000000
max      117854.000000
Name: ivch, dtype: float64


Descriptive statistics for siv in the year 1990:
count      6578.000000
mean        165.789756
std        2763.170154
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max      111234.000000
Name: siv, dtype: float64


Descriptive statistics for capx in the year 1990:
count    6817.000000
mean       79.284433
std       410.698082
min        -0.245000
25%         0.207000
50%         2.073000
75%        17.161000
max      9743.800000
Name: capx, dtype: float64


Descriptive statistics for sppe in the year 1990:
count    5704.000000
mean        6.324098
std        59.108231
min       -91.627000
25%         0.000000
50%         0.000000
75%         0.246250
max      1737.562000
Name: sppe, dtype: float64


Descriptive statistics for aqc in the year 1990:
count    6667.000000
mean       15.408582
std       164.051577
min     -1161.891000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max      4595.000000
Name: aqc, dtype: float64


Descriptive statistics for ivstch in the year 1990:
count    6058.000000
mean       -4.449598
std       117.742718
min     -6181.000000
25%         0.000000
50%         0.000000
75%         0.000000
max      1710.188000
Name: ivstch, dtype: float64


Descriptive statistics for ivaco in the year 1990:
count     6830.000000
mean        -0.238979
std        216.527314
min     -13271.100000
25%         -0.100000
50%          0.000000
75%          0.003000
max       2295.748000
Name: ivaco, dtype: float64


Descriptive statistics for ivncf in the year 1990:
count     6831.000000
mean      -101.594103
std        619.488344
min     -15909.000000
25%        -21.212500
50%         -2.137000
75%         -0.100000
max       5144.000000
Name: ivncf, dtype: float64


Descriptive statistics for sstk in the year 1990:
count    6738.000000
mean        8.273634
std        62.227714
min        -0.069000
```

```
25%         0.000000
50%         0.002000
75%         0.728750
max      2446.070000
Name: sstk, dtype: float64


Descriptive statistics for prstkc in the year 1990:
count    6714.000000
mean        8.300382
std        60.153787
min         0.000000
25%         0.000000
50%         0.000000
75%         0.050000
max      2485.000000
Name: prstkc, dtype: float64


Descriptive statistics for dv in the year 1990:
count    6884.000000
mean       23.607575
std       124.132354
min        -5.383000
25%         0.000000
50%         0.000000
75%         2.811750
max      3393.000000
Name: dv, dtype: float64


Descriptive statistics for dltis in the year 1990:
count     6692.000000
mean        97.132380
std        873.085295
min         -3.600000
25%          0.000000
50%          0.237000
75%         12.469250
max      53832.300000
Name: dltis, dtype: float64


Descriptive statistics for dltr in the year 1990:
count     6741.000000
mean        75.277669
std        711.855184
min          0.000000
```

```
25%          0.041000
50%          0.964000
75%         11.980000
max      46203.000000
Name: dltr, dtype: float64


Descriptive statistics for dlcch in the year 1990:
count    3813.000000
mean       20.846873
std       277.652267
min     -2903.601000
25%        -0.425000
50%         0.000000
75%         1.138000
max      8744.895000
Name: dlcch, dtype: float64


Descriptive statistics for fiao in the year 1990:
count    6830.000000
mean        2.850525
std       121.220879
min     -3338.053000
25%         0.000000
50%         0.000000
75%         0.000000
max      4927.189000
Name: fiao, dtype: float64


Descriptive statistics for fincf in the year 1990:
count     6831.000000
mean         9.851264
std        406.777154
min      -6033.000000
25%         -4.111500
50%         -0.009000
75%          2.610000
max      13864.004000
Name: fincf, dtype: float64


Descriptive statistics for exre in the year 1990:
count    6811.000000
mean        0.462430
std        14.494117
min      -220.196000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max       851.000000
Name: exre, dtype: float64


Descriptive statistics for chech in the year 1990:
count    6922.000000
mean        4.076251
std       137.794286
min     -1936.900000
25%        -1.026000
50%         0.000000
75%         1.268500
max      5298.262000
Name: chech, dtype: float64


Descriptive statistics for fsrco in the year 1990:
count     106.000000
mean      215.764349
std       582.618181
min        -5.961000
25%         0.000000
50%         0.574000
75%        68.582000
max      2999.320000
Name: fsrco, dtype: float64


Descriptive statistics for fuseo in the year 1990:
count     106.000000
mean      204.008792
std       480.877173
min        -0.493000
25%         0.000000
50%        12.966000
75%       153.307500
max      3411.904000
Name: fuseo, dtype: float64


Descriptive statistics for wcapc in the year 1990:
count      90.000000
mean       -5.291722
std       380.774587
min     -2148.000000
```

```
25%       -21.008250
50%         0.091000
75%         9.188750
max      1916.591000
Name: wcapc, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 1990:
count    3650.000000
mean            -inf
std              NaN
min             -inf
25%        -0.022532
50%         0.000000
75%         0.043476
max        50.500000
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 1990:
count    7504.000000
mean        0.412893
std         4.133537
min      -237.750000
25%         0.135905
50%         0.410596
75%         0.651805
max       149.334443
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 1990:
count    6206.000000
mean            -inf
std              NaN
min             -inf
25%         0.003868
50%         0.184544
75%         0.390782
max         1.000000
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 1990:
count    7403.000000
mean            -inf
std              NaN
min             -inf
```

```
25%        -0.288629
50%         0.041976
75%         0.230905
max         1.154107
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 1990:
count    7199.000000
mean            -inf
std              NaN
min             -inf
25%        -0.024713
50%         0.056812
75%         0.106188
max        16.000000
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 1990:
count    5686.000000
mean             inf
std              NaN
min         0.000000
25%         0.366690
50%         1.030237
75%         3.091468
max              inf
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 1990:
count    7196.000000
mean             inf
std              NaN
min        -0.754561
25%         0.409970
50%         0.981859
75%         1.580400
max              inf
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 1990:
count    4731.000000
mean        3.431537
std        66.822196
min     -4366.248804
```

```
25%         1.352170
50%         2.705100
75%         4.507037
max       670.856286
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 1990:
count    6430.000000
mean        1.078165
std         1.424060
min        -0.037242
25%         0.098794
50%         0.535888
75%         1.616943
max        21.283204
Name: sale_to_at_avg, dtype: float64


Descriptive statistics for dltis_at in the year 1990:
count    6685.000000
mean        0.094628
std         0.321414
min        -0.001290
25%         0.000000
50%         0.007611
75%         0.092116
max        15.777778
Name: dltis_at, dtype: float64


Descriptive statistics for dlc_at in the year 1990:
count    7508.000000
mean             inf
std              NaN
min        -0.019703
25%         0.006632
50%         0.039648
75%         0.127513
max              inf
Name: dlc_at, dtype: float64


Descriptive statistics for dltt_at in the year 1990:
count    7542.000000
mean        0.226594
std         0.922664
min         0.000000
```

```
25%         0.012870
50%         0.132924
75%         0.313955
max        62.222222
Name: dltt_at, dtype: float64


Descriptive statistics for ppent_at in the year 1990:
count    7492.000000
mean        0.306266
std         0.269109
min         0.000000
25%         0.077710
50%         0.234613
75%         0.475737
max         1.000000
Name: ppent_at, dtype: float64


Descriptive statistics for mkvalt_at in the year 1990:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: mkvalt_at, dtype: float64


Descriptive statistics for ni_at in the year 1990:
count    7277.000000
mean             NaN
std              NaN
min             -inf
25%        -0.055004
50%         0.018282
75%         0.062532
max              inf
Name: ni_at, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 1990:
count    3650.000000
mean            -inf
std              NaN
min             -inf
```

```
25%        -0.022532
50%         0.000000
75%         0.043476
max        50.500000
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 1990:
count    7504.000000
mean        0.412893
std         4.133537
min      -237.750000
25%         0.135905
50%         0.410596
75%         0.651805
max       149.334443
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 1990:
count    6206.000000
mean            -inf
std              NaN
min             -inf
25%         0.003868
50%         0.184544
75%         0.390782
max         1.000000
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 1990:
count    7403.000000
mean            -inf
std              NaN
min             -inf
25%        -0.288629
50%         0.041976
75%         0.230905
max         1.154107
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 1990:
count    7199.000000
mean            -inf
std              NaN
min             -inf
```

```
25%        -0.024713
50%         0.056812
75%         0.106188
max        16.000000
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 1990:
count    5686.000000
mean             inf
std              NaN
min         0.000000
25%         0.366690
50%         1.030237
75%         3.091468
max              inf
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 1990:
count    7196.000000
mean             inf
std              NaN
min        -0.754561
25%         0.409970
50%         0.981859
75%         1.580400
max              inf
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 1990:
count    4731.000000
mean        3.431537
std        66.822196
min     -4366.248804
25%         1.352170
50%         2.705100
75%         4.507037
max       670.856286
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 1990:
count    6430.000000
mean        1.078165
std         1.424060
min        -0.037242
```

```
25%         0.098794
50%         0.535888
75%         1.616943
max        21.283204
Name: sale_to_at_avg, dtype: float64
```

[106]:

```
          gvkey   datadate  fyear indfmt consol popsrc datafmt     tic  \
47         1003 1990-01-31   1989   INDL      C      D     STD    ANTQ
234840    12681 1990-01-31   1989   INDL      C      D     STD   FMY.1
110817     6346 1990-01-31   1989   INDL      C      D     STD    APO3
314797    23218 1990-01-31   1989   INDL      C      D     STD    VENS
234202    12631 1990-01-31   1989   INDL      C      D     STD    LE.2
…           …        …    …     …     …    …     …      …      …
245815    13534 1990-12-31   1990   INDL      C      D     STD   ABR.2
245808    13533 1990-12-31   1990   INDL      C      D     STD   8576B
245787    13532 1990-12-31   1990   INDL      C      D     STD   MGP.2
245923    13546 1990-12-31   1990   INDL      C      D     STD  3XXQXA
471358   145348 1990-12-31   1990   INDL      C      D     STD    PPL2

             cusip                          conm  … net_debt_issued_ratio  \
47       000354100         A.A. IMPORTING CO INC  …                   NaN
234840   592907109              MEYER (FRED) INC  …                   NaN
110817   037612009           CKE RESTAURANTS INC  …              0.027838
314797   923275101            VENTURE STORES INC  …                   NaN
234202   515086106            LANDS END INC -OLD  …                   NaN
…            …                              …  …                   …
245815   038917100          ARBOR PROPERTY TRUST  …                   NaN
245808   12525L104       CF INCOME PARTNERS  -LP  …                   NaN
245787   588539106           MERCHANTS GROUP INC  …              0.000000
245923   363170101       GALAXY CABLEVISION  -LP  …                   NaN
471358   69399Y000  PPL ELECTRIC UTILITIES CORP  …             -0.013740

        book_leverage     wc_ta     re_ta   ebit_ta     mv_tl  sales_ta  \
47           1.101241 -0.124542 -0.403403 -0.229696  0.031865  1.905925
234840       0.452085  0.213917  0.163648  0.066720  0.629129  2.866799
110817       0.723341 -0.091422  0.129327  0.100977  0.727215  1.660646
314797       0.059672  0.129562  0.000000  0.131447       NaN  2.468900
234202       0.056058  0.334262  0.524261  0.279862  6.268852  3.266007
…               …        …         …         …         …         …
245815       0.573461       NaN       NaN  0.090532  1.213243  0.114453
245808       0.773103       NaN       NaN  0.045043  0.056015  0.204481
245787       0.012676       NaN  0.136764  0.046788  0.157134  0.600929
245923       0.708833       NaN       NaN -0.086899  0.806471  0.392872
471358       0.517228 -0.033991  0.112562  0.101697  0.675628  0.308802
```

```
         z_score at_rolling_avg sale_to_at_avg
47      0.433772            NaN            NaN
234840  3.921593       403.5015       5.661776
110817  2.484944       551.8725       0.923349
314797       NaN       428.0045       3.167761
234202  9.053284       358.0450       1.522716
…            …             …              …
245815       NaN       136.6540       0.123421
245808       NaN       206.7985       0.263251
245787       NaN       215.9060       0.460849
245923       NaN       102.9210       0.153700
471358  1.163487      3887.8515       0.614406

[9596 rows x 998 columns]
```

0.7 Descriptive Stats of the Finaincal Ratios for the Year 1995

[107]:
```
describe_yearly_stats(compustat_copy, 1995)
```

```
Descriptive statistics for ch in the year 1995:
count    10991.000000
mean       105.481474
std        738.566205
min         -1.874000
25%          0.782000
50%          4.434000
75%         20.754500
max      18212.764000
Name: ch, dtype: float64


Descriptive statistics for ivst in the year 1995:
count    10938.000000
mean       196.689261
std       2364.296404
min          0.000000
25%          0.000000
50%          0.000000
75%          5.144250
max      74416.496000
Name: ivst, dtype: float64


Descriptive statistics for rect in the year 1995:
count     10038.000000
mean        780.348583
std        8242.809704
min           0.000000
```

```
25%           1.918750
50%          13.185000
75%          90.325000
max      362638.000000
Name: rect, dtype: float64


Descriptive statistics for invt in the year 1995:
count     10036.000000
mean        140.061493
std        1695.414998
min           0.000000
25%           0.000000
50%           2.557500
75%          26.094750
max      112544.000000
Name: invt, dtype: float64


Descriptive statistics for aco in the year 1995:
count     9264.000000
mean        33.592337
std        286.212836
min        -10.700000
25%          0.107000
50%          0.940000
75%          6.460750
max      18223.191000
Name: aco, dtype: float64


Descriptive statistics for act in the year 1995:
count     8232.00000
mean       376.30309
std       2202.87157
min          0.00000
25%          6.25575
50%         28.63300
75%        128.07550
max      67145.30300
Name: act, dtype: float64


Descriptive statistics for ppent in the year 1995:
count     10839.000000
mean        458.863808
std        2734.733126
min           0.000000
```

```
25%           1.849000
50%           9.780000
75%          76.677500
max      121993.000000
Name: ppent, dtype: float64


Descriptive statistics for ivaeq in the year 1995:
count    10131.000000
mean        32.501201
std        292.877307
min        -41.066000
25%          0.000000
50%          0.000000
75%          0.000000
max       9677.311000
Name: ivaeq, dtype: float64


Descriptive statistics for ivao in the year 1995:
count      9736.000000
mean        351.397372
std        3291.913858
min          -0.007000
25%           0.000000
50%           0.000000
75%           8.314500
max      109865.819000
Name: ivao, dtype: float64


Descriptive statistics for intan in the year 1995:
count     9701.000000
mean        74.783567
std        562.965683
min          0.000000
25%          0.000000
50%          0.000000
75%          5.779000
max      20801.000000
Name: intan, dtype: float64


Descriptive statistics for ao in the year 1995:
count     11104.000000
mean        244.046508
std        2604.630120
min       -4990.101000
```

```
25%           0.317750
50%           2.996000
75%          20.438750
max      103406.000000
Name: ao, dtype: float64


Descriptive statistics for at in the year 1995:
count     11099.000000
mean       3002.477081
std       19437.497400
min           0.000000
25%          19.185500
50%         116.117000
75%         596.245000
max      606186.000000
Name: at, dtype: float64


Descriptive statistics for dlc in the year 1995:
count     11064.000000
mean        422.613273
std        4523.125820
min       -1737.650000
25%           0.065000
50%           1.699500
75%          16.000000
max      161728.000000
Name: dlc, dtype: float64


Descriptive statistics for ap in the year 1995:
count     10147.000000
mean        620.817899
std        8194.627576
min           0.000000
25%           1.028500
50%           6.023000
75%          49.004000
max      431730.000000
Name: ap, dtype: float64


Descriptive statistics for txp in the year 1995:
count    9264.000000
mean       14.853283
std       108.207658
min       -53.000000
```

```
25%         0.000000
50%         0.000000
75%         0.786500
max      3426.000000
Name: txp, dtype: float64


Descriptive statistics for lco in the year 1995:
count     9266.000000
mean       112.265969
std        753.893885
min         -4.900000
25%          0.673250
50%          4.010500
75%         23.735250
max      30484.210000
Name: lco, dtype: float64


Descriptive statistics for lct in the year 1995:
count      8351.000000
mean        308.839004
std        2209.792195
min           0.000000
25%           3.248500
50%          13.461000
75%          70.326000
max      101601.000000
Name: lct, dtype: float64


Descriptive statistics for dltt in the year 1995:
count     11075.000000
mean        440.575959
std        3483.154649
min           0.000000
25%           0.103000
50%           5.793000
75%          81.662000
max      153021.000000
Name: dltt, dtype: float64


Descriptive statistics for lo in the year 1995:
count     11094.000000
mean        395.339012
std        3708.958957
min        -599.636000
```

```
25%           0.000000
50%           0.984500
75%          13.816250
max      131940.767000
Name: lo, dtype: float64


Descriptive statistics for txditc in the year 1995:
count     9036.000000
mean        56.928950
std        373.964123
min       -149.878000
25%          0.000000
50%          0.000000
75%          2.204000
max      12431.000000
Name: txditc, dtype: float64


Descriptive statistics for mib in the year 1995:
count    10333.000000
mean        19.197115
std        191.249496
min         -8.000000
25%          0.000000
50%          0.000000
75%          0.000000
max       7586.820000
Name: mib, dtype: float64


Descriptive statistics for lt in the year 1995:
count     11090.000000
mean       2530.528689
std       18136.626566
min           0.000000
25%           7.625000
50%          61.131500
75%         405.957000
max      581932.000000
Name: lt, dtype: float64


Descriptive statistics for pstk in the year 1995:
count    11085.000000
mean        13.187854
std         97.446143
min          0.000000
```

```
25%          0.000000
50%          0.000000
75%          0.000000
max       3071.000000
Name: pstk, dtype: float64


Descriptive statistics for ceq in the year 1995:
count    11085.000000
mean       437.054505
std       1933.830287
min      -4191.000000
25%          6.058000
50%         32.897000
75%        153.794000
max      56413.957000
Name: ceq, dtype: float64


Descriptive statistics for teq in the year 1995:
count      421.000000
mean      1664.654470
std       5097.107613
min       -775.526000
25%         53.213000
50%        211.500000
75%       1083.322000
max      56413.957000
Name: teq, dtype: float64


Descriptive statistics for sale in the year 1995:
count     10091.000000
mean       1020.693956
std        5376.221913
min          -5.819000
25%          12.616000
50%          59.805000
75%         335.478000
max      195805.000000
Name: sale, dtype: float64


Descriptive statistics for cogs in the year 1995:
count     10090.000000
mean        693.760111
std        4060.348982
min          -1.277000
```

```
25%           6.998000
50%          34.679500
75%         212.835250
max      189081.000000
Name: cogs, dtype: float64


Descriptive statistics for xsga in the year 1995:
count     8140.000000
mean       182.910725
std       1000.662386
min       -134.000000
25%          4.118000
50%         13.323500
75%         51.303750
max      24325.000000
Name: xsga, dtype: float64


Descriptive statistics for oibdp in the year 1995:
count     9854.000000
mean       181.072933
std        948.863964
min       -779.000000
25%          0.311500
50%          7.478000
75%         45.853500
max      27824.000000
Name: oibdp, dtype: float64


Descriptive statistics for dp in the year 1995:
count    10762.000000
mean        55.782495
std        391.247435
min         -7.909000
25%          0.407000
50%          1.789000
75%         11.664250
max      20585.000000
Name: dp, dtype: float64


Descriptive statistics for oiadp in the year 1995:
count    10090.000000
mean       122.434931
std        643.993458
min      -3977.000000
```

```
25%         -0.118000
50%          5.096500
75%         32.374000
max      21369.000000
Name: oiadp, dtype: float64


Descriptive statistics for xint in the year 1995:
count     9734.000000
mean        79.184088
std        657.814749
min          0.000000
25%          0.158000
50%          1.546500
75%         13.713000
max      19824.498000
Name: xint, dtype: float64


Descriptive statistics for nopi in the year 1995:
count    10091.000000
mean         6.940913
std        144.762526
min      -3613.183000
25%          0.000000
50%          0.125000
75%          1.311000
max       5330.890000
Name: nopi, dtype: float64


Descriptive statistics for spi in the year 1995:
count    10766.000000
mean        -7.522476
std        110.593727
min      -7845.000000
25%          0.000000
50%          0.000000
75%          0.000000
max       1065.300000
Name: spi, dtype: float64


Descriptive statistics for pi in the year 1995:
count    11075.000000
mean        88.374392
std        445.114332
min      -5283.000000
```

```
25%         -0.522000
50%          3.399000
75%         23.653000
max      12592.000000
Name: pi, dtype: float64


Descriptive statistics for txt in the year 1995:
count    11072.000000
mean        31.913198
std        163.425345
min       -334.000000
25%          0.000000
50%          0.804500
75%          7.341000
max       5502.000000
Name: txt, dtype: float64


Descriptive statistics for mii in the year 1995:
count    10091.000000
mean         1.830939
std         18.000307
min       -288.000000
25%          0.000000
50%          0.000000
75%          0.000000
max        627.000000
Name: mii, dtype: float64


Descriptive statistics for ib in the year 1995:
count    11074.000000
mean        54.830895
std        285.489171
min      -5223.000000
25%         -0.521750
50%          2.397500
75%         15.800250
max       6932.500000
Name: ib, dtype: float64


Descriptive statistics for dvp in the year 1995:
count    11074.000000
mean         1.361315
std         10.262047
min         -0.854000
```

```
25%          0.000000
50%          0.000000
75%          0.000000
max        338.000000
Name: dvp, dtype: float64


Descriptive statistics for cstke in the year 1995:
count    11074.000000
mean         0.012831
std          3.025865
min       -195.200000
25%          0.000000
50%          0.000000
75%          0.000000
max         62.000000
Name: cstke, dtype: float64


Descriptive statistics for xido in the year 1995:
count    11074.000000
mean        -3.226051
std        101.995690
min      -4682.000000
25%          0.000000
50%          0.000000
75%          0.000000
max       1565.734000
Name: xido, dtype: float64


Descriptive statistics for ni in the year 1995:
count    10096.000000
mean        47.672081
std        288.663824
min      -5223.000000
25%         -0.898000
50%          1.954000
75%         14.700500
max       6907.000000
Name: ni, dtype: float64


Descriptive statistics for ibc in the year 1995:
count    9187.000000
mean       49.855218
std       284.417285
min     -5223.000000
```

```
25%        -1.162000
50%         1.606000
75%        14.723000
max      6932.500000
Name: ibc, dtype: float64


Descriptive statistics for dpc in the year 1995:
count     9042.000000
mean        64.579143
std        428.837638
min         -0.498000
25%          0.515000
50%          2.518000
75%         15.583000
max      20585.000000
Name: dpc, dtype: float64


Descriptive statistics for xidoc in the year 1995:
count    9177.000000
mean        0.685669
std        32.029251
min      -831.000000
25%         0.000000
50%         0.000000
75%         0.000000
max      1780.000000
Name: xidoc, dtype: float64


Descriptive statistics for txdc in the year 1995:
count    8964.000000
mean        0.740174
std        46.464872
min     -1973.000000
25%        -0.033000
50%         0.000000
75%         0.028000
max      1392.000000
Name: txdc, dtype: float64


Descriptive statistics for esubc in the year 1995:
count    8104.000000
mean       -0.244428
std        18.361451
min      -468.316000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max       791.000000
Name: esubc, dtype: float64


Descriptive statistics for sppiv in the year 1995:
count    9069.000000
mean       -1.882344
std        82.884039
min     -1580.000000
25%         0.000000
50%         0.000000
75%         0.000000
max      6163.455000
Name: sppiv, dtype: float64


Descriptive statistics for fopo in the year 1995:
count     9187.000000
mean        17.019607
std        236.888319
min      -7243.212000
25%          0.000000
50%          0.096000
75%          1.807500
max      13508.000000
Name: fopo, dtype: float64


Descriptive statistics for fopt in the year 1995:
count       53.000000
mean       470.344094
std       2067.430705
min        -27.000000
25%         -0.002000
50%         34.201000
75%        157.992000
max      14919.000000
Name: fopt, dtype: float64


Descriptive statistics for recch in the year 1995:
count     8324.000000
mean       -13.295653
std        170.981491
min      -4681.000000
```

```
25%         -4.948500
50%         -0.643500
75%          0.021000
max      11514.000000
Name: recch, dtype: float64


Descriptive statistics for invch in the year 1995:
count     8457.000000
mean       -13.893658
std        242.078519
min     -17757.000000
25%         -1.946000
50%          0.000000
75%          0.000000
max       1814.202000
Name: invch, dtype: float64


Descriptive statistics for apalch in the year 1995:
count    7025.000000
mean        8.967966
std       119.040660
min     -6851.000000
25%        -0.142000
50%         0.429000
75%         3.242000
max      3173.000000
Name: apalch, dtype: float64


Descriptive statistics for txach in the year 1995:
count    7219.000000
mean        0.851210
std        34.504993
min     -1049.833000
25%         0.000000
50%         0.000000
75%         0.000000
max      1945.200000
Name: txach, dtype: float64


Descriptive statistics for aoloch in the year 1995:
count     9134.000000
mean         7.758787
std        257.759082
min      -4269.000000
```

```
25%         -0.936000
50%         -0.008000
75%          0.738000
max      17223.000000
Name: aoloch, dtype: float64


Descriptive statistics for oancf in the year 1995:
count     9139.000000
mean       118.236078
std        711.421030
min      -7869.000000
25%         -0.587500
50%          2.707000
75%         26.672500
max      22652.000000
Name: oancf, dtype: float64


Descriptive statistics for ivch in the year 1995:
count      8766.000000
mean        239.212937
std        3784.992967
min           0.000000
25%           0.000000
50%           0.000000
75%           0.175000
max      181456.400000
Name: ivch, dtype: float64


Descriptive statistics for siv in the year 1995:
count      8802.000000
mean        199.614122
std        3412.796850
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max      177940.400000
Name: siv, dtype: float64


Descriptive statistics for capx in the year 1995:
count     9028.000000
mean        86.031633
std        570.444846
min         -0.013000
```

```
25%          0.335750
50%          2.455500
75%         19.094250
max      23485.000000
Name: capx, dtype: float64


Descriptive statistics for sppe in the year 1995:
count    7711.000000
mean        6.164066
std        85.177971
min       -40.000000
25%         0.000000
50%         0.000000
75%         0.082500
max      5642.500000
Name: sppe, dtype: float64


Descriptive statistics for aqc in the year 1995:
count    8813.000000
mean       20.630758
std       200.056555
min     -1501.891000
25%         0.000000
50%         0.000000
75%         0.000000
max      9756.746000
Name: aqc, dtype: float64


Descriptive statistics for ivstch in the year 1995:
count     7987.000000
mean         9.504617
std        336.412636
min      -3304.000000
25%          0.000000
50%          0.000000
75%          0.000000
max      26873.000000
Name: ivstch, dtype: float64


Descriptive statistics for ivaco in the year 1995:
count     9143.000000
mean        -2.665150
std        627.787562
min     -30759.000000
```

```
25%         -0.190000
50%          0.000000
75%          0.000000
max      34325.200000
Name: ivaco, dtype: float64


Descriptive statistics for ivncf in the year 1995:
count     9143.000000
mean      -129.107766
std       1032.280758
min     -42237.000000
25%        -34.324000
50%         -4.024000
75%         -0.281000
max      12208.000000
Name: ivncf, dtype: float64


Descriptive statistics for sstk in the year 1995:
count    9071.000000
mean       12.415320
std        56.064861
min        -0.478000
25%         0.000000
50%         0.166000
75%         3.528000
max      1805.000000
Name: sstk, dtype: float64


Descriptive statistics for prstkc in the year 1995:
count    8808.000000
mean       10.425702
std       130.878865
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max      8350.000000
Name: prstkc, dtype: float64


Descriptive statistics for dv in the year 1995:
count    9123.000000
mean       23.492311
std       133.528153
min        -5.308000
```

```
25%         0.000000
50%         0.000000
75%         1.935500
max      4562.000000
Name: dv, dtype: float64


Descriptive statistics for dltis in the year 1995:
count     8920.000000
mean       133.222692
std       1253.324625
min          0.000000
25%          0.000000
50%          0.370500
75%         18.726500
max      50039.000000
Name: dltis, dtype: float64


Descriptive statistics for dltr in the year 1995:
count     8962.000000
mean        96.830085
std        824.525260
min         -0.158000
25%          0.018000
50%          0.926000
75%         13.760250
max      44138.700000
Name: dltr, dtype: float64


Descriptive statistics for dlcch in the year 1995:
count     4856.000000
mean         0.892155
std        523.531409
min     -19077.000000
25%         -0.224250
50%          0.000000
75%          1.073250
max      15208.000000
Name: dlcch, dtype: float64


Descriptive statistics for fiao in the year 1995:
count     9143.000000
mean         1.867850
std        164.990618
min     -11544.000000
```

```
25%          0.000000
50%          0.000000
75%          0.000000
max       5155.000000
Name: fiao, dtype: float64


Descriptive statistics for fincf in the year 1995:
count     9143.000000
mean        16.465107
std        669.974550
min     -12667.000000
25%         -2.254000
50%          0.521000
75%          8.699000
max      35875.000000
Name: fincf, dtype: float64


Descriptive statistics for exre in the year 1995:
count    9111.000000
mean       -0.001898
std         7.718065
min      -155.090000
25%         0.000000
50%         0.000000
75%         0.000000
max       366.000000
Name: exre, dtype: float64


Descriptive statistics for chech in the year 1995:
count    9186.000000
mean        5.426627
std       161.650918
min     -3959.000000
25%        -0.968500
50%         0.069000
75%         2.791750
max      6797.000000
Name: chech, dtype: float64


Descriptive statistics for fsrco in the year 1995:
count      54.000000
mean       60.806407
std       229.618283
min        -2.404000
```

```
25%         0.000000
50%         1.785000
75%        16.700000
max      1604.000000
Name: fsrco, dtype: float64


Descriptive statistics for fuseo in the year 1995:
count      53.000000
mean      138.532170
std       763.381027
min         0.000000
25%         0.001000
50%         6.264000
75%        32.000000
max      5554.000000
Name: fuseo, dtype: float64


Descriptive statistics for wcapc in the year 1995:
count     51.000000
mean     -19.464588
std       92.883064
min     -546.000000
25%      -11.892000
50%       -0.077000
75%        2.343000
max      110.000000
Name: wcapc, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 1995:
count    4648.000000
mean        0.038824
std         0.292745
min        -3.139013
25%        -0.011627
50%         0.000000
75%         0.045759
max         7.132075
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 1995:
count    11037.000000
mean         0.090332
std         21.940410
min      -1771.333333
```

```
25%          0.066369
50%          0.335997
75%          0.579137
max        301.000000
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 1995:
count    8215.000000
mean            -inf
std              NaN
min             -inf
25%         0.018289
50%         0.202671
75%         0.418524
max         1.000000
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 1995:
count    1.082000e+04
mean             -inf
std               NaN
min              -inf
25%     -2.258473e-01
50%      4.351247e-02
75%      1.698954e-01
max      1.361217e+00
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 1995:
count    1.007600e+04
mean              NaN
std               NaN
min              -inf
25%     -9.988630e-03
50%      5.299271e-02
75%      1.104768e-01
max               inf
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 1995:
count    8037.000000
mean             inf
std              NaN
min         0.000417
```

```
25%         0.534411
50%         1.612729
75%         5.124921
max              inf
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 1995:
count    1.006800e+04
mean              inf
std               NaN
min     -7.796610e-01
25%      2.757985e-01
50%      8.828250e-01
75%      1.500610e+00
max               inf
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 1995:
count    6291.000000
mean             inf
std              NaN
min     -9151.318664
25%         1.731018
50%         3.319311
75%         5.949203
max              inf
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 1995:
count    9347.000000
mean        0.959502
std         1.328425
min        -0.728628
25%         0.083118
50%         0.410017
75%         1.441086
max        27.089912
Name: sale_to_at_avg, dtype: float64


Descriptive statistics for dltis_at in the year 1995:
count    8906.000000
mean        0.139802
std         1.640614
min         0.000000
```

```
25%         0.000000
50%         0.009854
75%         0.113608
max       151.000000
Name: dltis_at, dtype: float64


Descriptive statistics for dlc_at in the year 1995:
count    1.104900e+04
mean              inf
std               NaN
min     -2.173197e-02
25%      1.604702e-03
50%      2.593208e-02
75%      8.800042e-02
max               inf
Name: dlc_at, dtype: float64


Descriptive statistics for dltt_at in the year 1995:
count    1.106400e+04
mean              inf
std               NaN
min      0.000000e+00
25%      3.738611e-03
50%      7.921995e-02
75%      2.705364e-01
max               inf
Name: dltt_at, dtype: float64


Descriptive statistics for ppent_at in the year 1995:
count    10827.000000
mean         0.248765
std          0.255285
min          0.000000
25%          0.030879
50%          0.159476
75%          0.389870
max          0.999050
Name: ppent_at, dtype: float64


Descriptive statistics for mkvalt_at in the year 1995:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: mkvalt_at, dtype: float64


Descriptive statistics for ni_at in the year 1995:
count    1.008200e+04
mean              NaN
std               NaN
min              -inf
25%     -4.109228e-02
50%      2.016029e-02
75%      6.713030e-02
max               inf
Name: ni_at, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 1995:
count    4648.000000
mean        0.038824
std         0.292745
min        -3.139013
25%        -0.011627
50%         0.000000
75%         0.045759
max         7.132075
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 1995:
count    11037.000000
mean         0.090332
std         21.940410
min      -1771.333333
25%          0.066369
50%          0.335997
75%          0.579137
max        301.000000
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 1995:
count    8215.000000
mean            -inf
std              NaN
min             -inf
```

```
25%         0.018289
50%         0.202671
75%         0.418524
max         1.000000
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 1995:
count    1.082000e+04
mean             -inf
std               NaN
min              -inf
25%     -2.258473e-01
50%      4.351247e-02
75%      1.698954e-01
max      1.361217e+00
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 1995:
count    1.007600e+04
mean              NaN
std               NaN
min              -inf
25%     -9.988630e-03
50%      5.299271e-02
75%      1.104768e-01
max               inf
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 1995:
count    8037.000000
mean             inf
std              NaN
min         0.000417
25%         0.534411
50%         1.612729
75%         5.124921
max              inf
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 1995:
count    1.006800e+04
mean              inf
std               NaN
min     -7.796610e-01
```

```
25%      2.757985e-01
50%      8.828250e-01
75%      1.500610e+00
max               inf
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 1995:
count    6291.000000
mean             inf
std              NaN
min     -9151.318664
25%         1.731018
50%         3.319311
75%         5.949203
max              inf
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 1995:
count    9347.000000
mean        0.959502
std         1.328425
min        -0.728628
25%         0.083118
50%         0.410017
75%         1.441086
max        27.089912
Name: sale_to_at_avg, dtype: float64
```

[107]:

	gvkey	datadate	fyear	indfmt	consol	popsrc	datafmt	tic	\
155273	8446	1995-01-31	1994	INDL	C	D	STD	CPPRQ	
172964	9305	1995-01-31	1994	INDL	C	D	STD	SKFBQ	
398611	61445	1995-01-31	1994	INDL	C	D	STD	IBI.1	
220185	11584	1995-01-31	1994	INDL	C	D	STD	FL	
133965	7466	1995-01-31	1994	INDL	C	D	STD	MND.2	
…	…	…	…	…	…	…	…	…	
109689	6306	1995-12-31	1995	INDL	C	D	STD	3KMSIE	
109834	6310	1995-12-31	1995	INDL	C	D	STD	KMI	
310416	22216	1995-12-31	1995	FS	C	D	STD	NBOH	
310729	22263	1995-12-31	1995	INDL	C	D	STD	NSCC	
306519	21381	1995-12-31	1995	INDL	C	D	STD	CPA.1	

	cusip	conm	…	net_debt_issued_ratio	\
155273	679535104	OLD COPPER CO INC	…	0.059129	

```
172964  783774102       S & K FAMOUS BRANDS INC  …                 NaN
398611  461156101    INTIMATE BRANDS INC  -CL A  …            0.000000
220185  344849104               FOOT LOCKER INC  …            0.067817
133965  606592202    MITCHELL ENERGY & DEV CORP  …                 NaN
…          …                              …  …                  …
109689  482580206            KMS INDUSTRIES INC  …            0.000000
109834  49456B101             KINDER MORGAN INC  …            0.005311
310416  632592101   NATIONAL BANCSHARES CORP/OH  …                 NaN
310729  62938T103                       NSC CORP  …                 NaN
306519  142522101  CARLISLE PLASTICS INC  -CL A  …                 NaN

       book_leverage     wc_ta      re_ta   ebit_ta     mv_tl  sales_ta  \
155273       0.491487  0.307802   0.263054  0.121528  0.891546  1.339711
172964       0.258659  0.573647   0.475322  0.076512  1.556559  1.883331
398611       0.000000  0.255106        NaN  0.439773       NaN  2.743227
220185       0.466614  0.086029   0.257848  0.055835  0.741369  1.987299
133965       0.656176  0.007326   0.187283  0.037343  0.629463  0.455716
…               …        …         …        …         …        …
109689       0.000000  0.187793 -45.508607 -1.020344  0.796898  0.474178
109834       0.498516 -0.021322   0.087218  0.090440  1.011357  0.877477
310416       0.134397       NaN   0.055583       NaN       NaN       NaN
310729       0.114695  0.198931   0.012643  0.023267  0.667403  1.148484
306519       0.811138  0.152259  -0.082937  0.062183  0.321922  1.447743

          z_score at_rolling_avg sale_to_at_avg
155273   2.999922            NaN            NaN
172964   4.404750      8130.8450       0.013826
398611        NaN       414.1205       5.091054
220185   3.060726      2470.7755       3.356436
133965   1.223054      3014.4355       0.280567
…           …             …              …
109689 -65.906259       188.5240       0.001607
109834   1.870486       629.0480       1.754062
310416        NaN       716.3005            NaN
310729   1.870640       131.1525       0.763257
306519   1.898220       190.8000       2.234130

[12411 rows x 998 columns]
```

0.8 Descriptive Stats of the Finaincal Ratios for the Year 2000

[108]:
```
describe_yearly_stats(compustat_copy, 2000)
```

```
Descriptive statistics for ch in the year 2000:
count    11241.000000
mean       169.994670
std       1143.459189
min         -4.647000
```

```
25%          1.425000
50%          8.836000
75%         40.420000
max      35368.000000
Name: ch, dtype: float64


Descriptive statistics for ivst in the year 2000:
count     11204.000000
mean        338.266812
std        4546.659685
min           0.000000
25%           0.000000
50%           0.000000
75%           8.244250
max      160631.404000
Name: ivst, dtype: float64


Descriptive statistics for rect in the year 2000:
count     10101.000000
mean       1485.726520
std       16260.075167
min           0.000000
25%           2.240000
50%          20.480000
75%         142.063000
max      611928.000000
Name: rect, dtype: float64


Descriptive statistics for invt in the year 2000:
count     10253.000000
mean        248.656444
std        3798.443812
min           0.000000
25%           0.000000
50%           1.925000
75%          30.938000
max      156336.255000
Name: invt, dtype: float64


Descriptive statistics for aco in the year 2000:
count     9268.000000
mean        77.900873
std        570.932508
min          0.000000
```

```
25%          0.173750
50%          1.926000
75%         12.790750
max      20705.000000
Name: aco, dtype: float64


Descriptive statistics for act in the year 2000:
count      8432.000000
mean        599.440374
std        2816.352172
min           0.000000
25%           9.311250
50%          55.298500
75%         228.957750
max      100931.000000
Name: act, dtype: float64


Descriptive statistics for ppent in the year 2000:
count     11121.000000
mean        635.081850
std        3382.676575
min           0.000000
25%           2.361000
50%          13.784000
75%         115.404000
max      111921.000000
Name: ppent, dtype: float64


Descriptive statistics for ivaeq in the year 2000:
count    10431.000000
mean        76.411111
std        874.355043
min         -7.104000
25%          0.000000
50%          0.000000
75%          0.000000
max      47914.000000
Name: ivaeq, dtype: float64


Descriptive statistics for ivao in the year 2000:
count      9786.000000
mean        788.618727
std        9255.758855
min        -167.655000
```

```
25%           0.000000
50%           0.000000
75%          12.772750
max      347019.000000
Name: ivao, dtype: float64


Descriptive statistics for intan in the year 2000:
count    10205.000000
mean       303.920475
std       2142.122209
min          0.000000
25%          0.000000
50%          0.930000
75%         33.963000
max      93322.000000
Name: intan, dtype: float64


Descriptive statistics for ao in the year 2000:
count     11396.000000
mean        572.499251
std        5619.917230
min         -26.919000
25%           0.559750
50%           5.857500
75%          40.436000
max      241662.000000
Name: ao, dtype: float64


Descriptive statistics for at in the year 2000:
count     11399.000000
mean       5829.385109
std       40110.381734
min           0.000000
25%          35.091000
50%         221.848000
75%        1040.893500
max      902210.000000
Name: at, dtype: float64


Descriptive statistics for dlc in the year 2000:
count     11293.000000
mean        876.069426
std        9949.803664
min           0.000000
```

```
25%           0.055000
50%           2.678000
75%          31.842000
max      399575.933000
Name: dlc, dtype: float64


Descriptive statistics for ap in the year 2000:
count     10394.000000
mean       1106.937856
std       13643.067768
min           0.000000
25%           1.420000
50%           9.491500
75%          80.331250
max      487122.000000
Name: ap, dtype: float64


Descriptive statistics for txp in the year 2000:
count    9450.000000
mean       23.336785
std       179.164601
min       -66.000000
25%         0.000000
50%         0.000000
75%         0.684000
max      6116.000000
Name: txp, dtype: float64


Descriptive statistics for lco in the year 2000:
count     9269.000000
mean       202.433052
std       1338.358090
min          0.000000
25%          1.004000
50%          7.882000
75%         44.953000
max      66931.243000
Name: lco, dtype: float64


Descriptive statistics for lct in the year 2000:
count      8490.000000
mean        531.032683
std        3354.570859
min           0.000000
```

```
25%           4.682250
50%          24.076500
75%         124.668000
max      187828.000000
Name: lct, dtype: float64


Descriptive statistics for dltt in the year 2000:
count     11383.000000
mean        897.812493
std        7537.178120
min           0.000000
25%           0.062500
50%          12.000000
75%         196.044000
max      362360.000000
Name: dltt, dtype: float64


Descriptive statistics for lo in the year 2000:
count     11394.000000
mean        963.725480
std       10781.785217
min           0.000000
25%           0.000000
50%           1.754000
75%          21.074250
max      398251.164000
Name: lo, dtype: float64


Descriptive statistics for txditc in the year 2000:
count     9018.000000
mean        89.205325
std        667.933540
min       -168.371000
25%          0.000000
50%          0.000000
75%          2.530000
max      36713.000000
Name: txditc, dtype: float64


Descriptive statistics for mib in the year 2000:
count    10770.000000
mean        47.871936
std        439.407324
min        -25.096000
```

```
25%          0.000000
50%          0.000000
75%          0.000000
max      21830.000000
Name: mib, dtype: float64


Descriptive statistics for lt in the year 2000:
count     11382.000000
mean       4937.664734
std       37423.805401
min           0.000000
25%          11.874000
50%         119.487500
75%         685.009000
max      856123.000000
Name: lt, dtype: float64


Descriptive statistics for pstk in the year 2000:
count    11371.000000
mean        18.764348
std        154.259597
min         -1.596000
25%          0.000000
50%          0.000000
75%          0.000000
max       5822.245000
Name: pstk, dtype: float64


Descriptive statistics for ceq in the year 2000:
count     11372.000000
mean        829.419947
std        4252.879972
min       -6979.138000
25%           9.272000
50%          55.367000
75%         270.203750
max      224234.303000
Name: ceq, dtype: float64


Descriptive statistics for teq in the year 2000:
count      694.000000
mean      2657.948480
std       8047.252282
min       -967.251000
```

```
25%         53.149500
50%        247.652000
75%       1354.000000
max      73416.000000
Name: teq, dtype: float64


Descriptive statistics for sale in the year 2000:
count     10128.000000
mean       1611.978162
std        7645.622052
min          -0.656000
25%          14.291250
50%          89.904500
75%         510.473500
max      206083.000000
Name: sale, dtype: float64


Descriptive statistics for cogs in the year 2000:
count     10128.000000
mean       1094.224060
std        5612.242323
min        -366.645000
25%           8.689250
50%          50.588000
75%         312.996000
max      159794.000000
Name: cogs, dtype: float64


Descriptive statistics for xsga in the year 2000:
count     8134.000000
mean       265.039439
std       1266.292834
min       -223.000000
25%          6.481750
50%         24.996500
75%         92.143750
max      27040.000000
Name: xsga, dtype: float64


Descriptive statistics for oibdp in the year 2000:
count     9891.000000
mean       303.185696
std       1767.406631
min      -2439.943000
```

```
25%         -0.965500
50%          7.825000
75%         73.184000
max      47803.000000
Name: oibdp, dtype: float64


Descriptive statistics for dp in the year 2000:
count    10997.000000
mean        92.767114
std        541.144820
min         -0.197000
25%          0.644000
50%          3.641000
75%         25.313000
max      22814.000000
Name: dp, dtype: float64


Descriptive statistics for oiadp in the year 2000:
count    10128.000000
mean       209.208066
std       1426.939349
min      -5201.000000
25%         -2.357750
50%          4.049500
75%         50.636250
max      45155.000000
Name: oiadp, dtype: float64


Descriptive statistics for xint in the year 2000:
count     9657.000000
mean       151.286104
std       1373.968330
min          0.000000
25%          0.243000
50%          4.035000
75%         29.635000
max      37107.000000
Name: xint, dtype: float64


Descriptive statistics for nopi in the year 2000:
count    10122.000000
mean         8.594948
std        310.199030
min     -13353.320000
```

```
25%          0.000000
50%          0.245000
75%          3.028000
max       7287.903000
Name: nopi, dtype: float64


Descriptive statistics for spi in the year 2000:
count    11227.000000
mean        -8.387067
std        234.496837
min      -6760.000000
25%         -0.707000
50%          0.000000
75%          0.000000
max       8570.000000
Name: spi, dtype: float64


Descriptive statistics for pi in the year 2000:
count    11362.000000
mean       149.255705
std        966.670558
min      -7439.796000
25%         -3.980750
50%          2.539000
75%         35.525500
max      27493.000000
Name: pi, dtype: float64


Descriptive statistics for txt in the year 2000:
count    11362.000000
mean        55.834097
std        340.914195
min      -2154.000000
25%          0.000000
50%          0.605500
75%         10.244250
max      11273.000000
Name: txt, dtype: float64


Descriptive statistics for mii in the year 2000:
count    10478.000000
mean         3.177057
std         59.448180
min      -4120.000000
```

```
25%          0.000000
50%          0.000000
75%          0.000000
max       1994.011000
Name: mii, dtype: float64


Descriptive statistics for ib in the year 2000:
count    11362.000000
mean        90.813197
std        638.469017
min      -7439.796000
25%         -3.971750
50%          1.753000
75%         23.861750
max      15990.000000
Name: ib, dtype: float64


Descriptive statistics for dvp in the year 2000:
count    11361.000000
mean         1.965578
std         15.336169
min        -29.181000
25%          0.000000
50%          0.000000
75%          0.000000
max        559.862000
Name: dvp, dtype: float64


Descriptive statistics for cstke in the year 2000:
count    11362.000000
mean        -0.372895
std         17.823212
min      -1564.000000
25%          0.000000
50%          0.000000
75%          0.000000
max        276.000000
Name: cstke, dtype: float64


Descriptive statistics for xido in the year 2000:
count    11362.000000
mean         0.899205
std         88.175080
min      -1943.000000
```

```
25%          0.000000
50%          0.000000
75%          0.000000
max       5097.000000
Name: xido, dtype: float64


Descriptive statistics for ni in the year 2000:
count    10128.000000
mean        82.380928
std        638.684057
min      -7439.796000
25%         -5.714250
50%          0.960500
75%         21.300500
max      17720.000000
Name: ni, dtype: float64


Descriptive statistics for ibc in the year 2000:
count     9179.000000
mean        77.083788
std        623.893886
min      -7439.796000
25%         -6.895500
50%          0.255000
75%         21.380500
max      15990.000000
Name: ibc, dtype: float64


Descriptive statistics for dpc in the year 2000:
count     9063.000000
mean       108.067927
std        597.399157
min         -9.178000
25%          0.840500
50%          5.570000
75%         34.308000
max      22814.000000
Name: dpc, dtype: float64


Descriptive statistics for xidoc in the year 2000:
count    9171.000000
mean        0.148236
std        33.555112
min     -1182.000000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max       965.000000
Name: xidoc, dtype: float64


Descriptive statistics for txdc in the year 2000:
count    9035.000000
mean        3.863943
std        82.664157
min     -1505.000000
25%         0.000000
50%         0.000000
75%         0.000000
max      3434.000000
Name: txdc, dtype: float64


Descriptive statistics for esubc in the year 2000:
count    8165.000000
mean       -0.669231
std        75.640430
min     -3792.000000
25%         0.000000
50%         0.000000
75%         0.000000
max      2161.000000
Name: esubc, dtype: float64


Descriptive statistics for sppiv in the year 2000:
count    9066.000000
mean      -16.173695
std       244.560581
min     -9628.333000
25%        -0.010000
50%         0.000000
75%         0.000000
max      6571.600000
Name: sppiv, dtype: float64


Descriptive statistics for fopo in the year 2000:
count     9179.000000
mean        20.548061
std        699.606210
min     -54792.123000
```

```
25%          0.000000
50%          0.603000
75%          6.722000
max      19873.458000
Name: fopo, dtype: float64


Descriptive statistics for fopt in the year 2000:
count      15.000000
mean     1095.627533
std      2269.685866
min      -329.217000
25%        65.526500
50%       360.340000
75%       752.075500
max      8668.245000
Name: fopt, dtype: float64


Descriptive statistics for recch in the year 2000:
count     8339.000000
mean       -39.239985
std        634.144735
min     -52341.630000
25%         -8.171000
50%         -0.720000
75%          0.094000
max       3538.000000
Name: recch, dtype: float64


Descriptive statistics for invch in the year 2000:
count     8482.000000
mean       -22.026287
std        531.586108
min     -30574.222000
25%         -2.005500
50%          0.000000
75%          0.000000
max       7130.601000
Name: invch, dtype: float64


Descriptive statistics for apalch in the year 2000:
count    6953.000000
mean       27.199284
std       258.236880
min     -1227.000000
```

```
25%        -0.310000
50%         0.486000
75%         5.568000
max      9741.000000
Name: apalch, dtype: float64


Descriptive statistics for txach in the year 2000:
count    7115.000000
mean        1.518089
std        77.291569
min     -3626.000000
25%         0.000000
50%         0.000000
75%         0.000000
max      3036.143000
Name: txach, dtype: float64


Descriptive statistics for aoloch in the year 2000:
count     9163.00000
mean        20.70538
std        816.36688
min     -10031.00000
25%         -1.77150
50%         -0.00200
75%          1.48650
max      55905.15700
Name: aoloch, dtype: float64


Descriptive statistics for oancf in the year 2000:
count     9174.000000
mean       177.180403
std       1116.325056
min     -27976.468000
25%         -2.269500
50%          2.384000
75%         44.186750
max      33764.000000
Name: oancf, dtype: float64


Descriptive statistics for ivch in the year 2000:
count      8852.000000
mean        465.793140
std        6620.297857
min           0.000000
```

```
25%           0.000000
50%           0.000000
75%           1.250000
max      243950.000000
Name: ivch, dtype: float64


Descriptive statistics for siv in the year 2000:
count      8873.000000
mean        362.204023
std        5411.926272
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max      228794.000000
Name: siv, dtype: float64


Descriptive statistics for capx in the year 2000:
count     9012.000000
mean       138.816960
std        889.618675
min         -4.000000
25%          0.398000
50%          4.233500
75%         30.576000
max      31605.000000
Name: capx, dtype: float64


Descriptive statistics for sppe in the year 2000:
count     7686.000000
mean        22.350017
std        357.843991
min        -78.000000
25%          0.000000
50%          0.000000
75%          0.104750
max      15993.000000
Name: sppe, dtype: float64


Descriptive statistics for aqc in the year 2000:
count     8718.000000
mean        66.305202
std        643.691918
min      -1519.000000
```

```
25%          0.000000
50%          0.000000
75%          0.723750
max      26295.788000
Name: aqc, dtype: float64


Descriptive statistics for ivstch in the year 2000:
count     7815.000000
mean         5.496186
std        407.204496
min      -4903.976000
25%          0.000000
50%          0.000000
75%          0.000000
max      30876.000000
Name: ivstch, dtype: float64


Descriptive statistics for ivaco in the year 2000:
count     9173.000000
mean        -3.061565
std        990.634962
min     -46834.854000
25%         -0.352000
50%          0.000000
75%          0.000000
max      46190.500000
Name: ivaco, dtype: float64


Descriptive statistics for ivncf in the year 2000:
count     9174.000000
mean      -276.273703
std       2439.315803
min     -98481.000000
25%        -55.635000
50%         -6.115500
75%         -0.209250
max       6893.057000
Name: ivncf, dtype: float64


Descriptive statistics for sstk in the year 2000:
count     9051.000000
mean        43.547697
std        277.618888
min         -0.103000
```

```
25%          0.000000
50%          0.575000
75%          8.895500
max      10413.000000
Name: sstk, dtype: float64


Descriptive statistics for prstkc in the year 2000:
count    8310.000000
mean       28.304506
std       207.491467
min        -3.430000
25%         0.000000
50%         0.000000
75%         0.583750
max      6073.000000
Name: prstkc, dtype: float64


Descriptive statistics for dv in the year 2000:
count    8899.000000
mean       33.980526
std       213.015234
min         0.000000
25%         0.000000
50%         0.000000
75%         0.673000
max      6123.000000
Name: dv, dtype: float64


Descriptive statistics for dltis in the year 2000:
count      8783.000000
mean        291.830018
std        2647.898540
min           0.000000
25%           0.000000
50%           0.270000
75%          31.742500
max      111666.460000
Name: dltis, dtype: float64


Descriptive statistics for dltr in the year 2000:
count     8957.000000
mean       180.329308
std       1350.743194
min         -0.047000
```

```
25%          0.000000
50%          1.232000
75%         26.138000
max      50320.000000
Name: dltr, dtype: float64


Descriptive statistics for dlcch in the year 2000:
count     4762.000000
mean        15.709926
std       1049.165457
min     -25934.000000
25%         -0.150000
50%          0.000000
75%          0.902500
max      26004.000000
Name: dlcch, dtype: float64


Descriptive statistics for fiao in the year 2000:
count     9174.000000
mean        19.049289
std        816.082488
min     -26714.268000
25%         -0.001000
50%          0.000000
75%          0.000000
max      50043.000000
Name: fiao, dtype: float64


Descriptive statistics for fincf in the year 2000:
count     9174.000000
mean       113.744758
std       1973.172619
min     -14165.000000
25%         -3.888750
50%          0.989000
75%         19.412750
max      84490.000000
Name: fincf, dtype: float64


Descriptive statistics for exre in the year 2000:
count    9138.000000
mean       -1.192533
std        27.041503
min     -1311.504000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max       859.002000
Name: exre, dtype: float64


Descriptive statistics for chech in the year 2000:
count     9180.000000
mean        13.797628
std        341.425072
min     -10646.931000
25%         -1.807750
50%          0.080500
75%          6.179250
max      11204.578000
Name: chech, dtype: float64


Descriptive statistics for fsrco in the year 2000:
count      14.000000
mean      710.353143
std      2382.026362
min         0.000000
25%         8.035500
50%        55.208000
75%       171.136500
max      8981.845000
Name: fsrco, dtype: float64


Descriptive statistics for fuseo in the year 2000:
count       14.000000
mean      1291.142714
std       4099.039372
min          0.583000
25%         46.437500
50%        135.748500
75%        347.113250
max      15516.589000
Name: fuseo, dtype: float64


Descriptive statistics for wcapc in the year 2000:
count      11.000000
mean      340.472000
std      1234.746691
min     -1845.568000
```

```
25%         8.795000
50%        70.067000
75%       441.437500
max      3361.153000
Name: wcapc, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 2000:
count    4498.000000
mean             NaN
std              NaN
min             -inf
25%        -0.008341
50%         0.000000
75%         0.047169
max              inf
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 2000:
count    11265.000000
mean         0.300700
std          8.674637
min       -725.000000
25%          0.039925
50%          0.359850
75%          0.639893
max        128.250000
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 2000:
count    8409.000000
mean            -inf
std              NaN
min             -inf
25%        -0.004298
50%         0.170787
75%         0.410979
max         1.000000
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 2000:
count    1.115300e+04
mean             -inf
std               NaN
min              -inf
```

```
25%     -5.331118e-01
50%      1.323934e-02
75%      1.241485e-01
max      1.806813e+00
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 2000:
count    1.011900e+04
mean             -inf
std               NaN
min              -inf
25%     -1.233949e-01
50%      2.481209e-02
75%      8.986904e-02
max      1.326500e+01
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 2000:
count    8972.000000
mean             inf
std              NaN
min         0.000000
25%         0.358887
50%         1.360471
75%         5.855717
max              inf
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 2000:
count    1.009300e+04
mean              inf
std               NaN
min     -2.550544e-01
25%      1.872708e-01
50%      6.692628e-01
75%      1.281094e+00
max               inf
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 2000:
count    7.261000e+03
mean              inf
std               NaN
min     -1.315383e+04
```

```
25%      7.393937e-01
50%      2.579184e+00
75%      5.471469e+00
max               inf
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 2000:
count    9568.000000
mean        0.829410
std         1.334374
min        -0.007749
25%         0.067197
50%         0.295750
75%         1.119043
max        38.771704
Name: sale_to_at_avg, dtype: float64


Descriptive statistics for dltis_at in the year 2000:
count    8745.000000
mean             inf
std              NaN
min         0.000000
25%         0.000000
50%         0.004195
75%         0.105022
max              inf
Name: dltis_at, dtype: float64


Descriptive statistics for dlc_at in the year 2000:
count    1.126000e+04
mean              inf
std               NaN
min      0.000000e+00
25%      9.693411e-04
50%      2.427659e-02
75%      9.503763e-02
max               inf
Name: dlc_at, dtype: float64


Descriptive statistics for dltt_at in the year 2000:
count    1.134600e+04
mean              inf
std               NaN
min      0.000000e+00
```

```
25%      2.033023e-03
50%      8.150712e-02
75%      2.814579e-01
max               inf
Name: dltt_at, dtype: float64


Descriptive statistics for ppent_at in the year 2000:
count    11082.000000
mean         0.216227
std          0.240713
min          0.000000
25%          0.024651
50%          0.118249
75%          0.334142
max          1.000000
Name: ppent_at, dtype: float64


Descriptive statistics for mkvalt_at in the year 2000:
count    9455.000000
mean             inf
std              NaN
min         0.000006
25%         0.167732
50%         0.547724
75%         1.689987
max              inf
Name: mkvalt_at, dtype: float64


Descriptive statistics for ni_at in the year 2000:
count    1.012100e+04
mean              NaN
std               NaN
min              -inf
25%     -1.605978e-01
50%      7.717777e-03
75%      4.906246e-02
max               inf
Name: ni_at, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 2000:
count    4498.000000
mean             NaN
std              NaN
min             -inf
```

```
25%        -0.008341
50%         0.000000
75%         0.047169
max              inf
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 2000:
count    11265.000000
mean         0.300700
std          8.674637
min       -725.000000
25%          0.039925
50%          0.359850
75%          0.639893
max        128.250000
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 2000:
count    8409.000000
mean            -inf
std              NaN
min             -inf
25%        -0.004298
50%         0.170787
75%         0.410979
max         1.000000
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 2000:
count    1.115300e+04
mean             -inf
std               NaN
min              -inf
25%     -5.331118e-01
50%      1.323934e-02
75%      1.241485e-01
max      1.806813e+00
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 2000:
count    1.011900e+04
mean             -inf
std               NaN
min              -inf
```

```
25%     -1.233949e-01
50%      2.481209e-02
75%      8.986904e-02
max      1.326500e+01
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 2000:
count    8972.000000
mean             inf
std              NaN
min         0.000000
25%         0.358887
50%         1.360471
75%         5.855717
max              inf
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 2000:
count    1.009300e+04
mean              inf
std               NaN
min     -2.550544e-01
25%      1.872708e-01
50%      6.692628e-01
75%      1.281094e+00
max               inf
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 2000:
count    7.261000e+03
mean              inf
std               NaN
min     -1.315383e+04
25%      7.393937e-01
50%      2.579184e+00
75%      5.471469e+00
max               inf
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 2000:
count    9568.000000
mean        0.829410
std         1.334374
min        -0.007749
```

```
25%          0.067197
50%          0.295750
75%          1.119043
max         38.771704
Name: sale_to_at_avg, dtype: float64
```

[108]:

	gvkey	datadate	fyear	indfmt	consol	popsrc	datafmt	tic	\
466485	141001	2000-01-31	1999	INDL	C	D	STD	UTIW	
44743	3164	2000-01-31	1999	INDL	C	D	STD	CNJ	
419319	64485	2000-01-31	1999	INDL	C	D	STD	NMGC	
419484	64516	2000-01-31	1999	INDL	C	D	STD	GPM.1	
218567	11511	2000-01-31	1999	INDL	C	D	STD	WSM	
…	…	…	…	…	…	…	…	…	
142024	7846	2000-12-31	2000	INDL	C	D	STD	VZ1	
141899	7841	2000-12-31	2000	INDL	C	D	STD	CRMZ	
360897	29553	2000-12-31	2000	INDL	C	D	STD	FFFL	
360723	29526	2000-12-31	2000	FS	C	D	STD	PTRS.1	
538146	277918	2000-12-31	2000	INDL	C	D	STD	ENVS	

	cusip	conm	…	net_debt_issued_ratio	\
466485	G87210103	UTI WORLDWIDE INC	…	NaN	
44743	193290103	COLE NATIONAL CORP	…	NaN	
419319	640497202	NEOMAGIC CORP	…	-0.003674	
419484	374292100	GETTY PETROLEUM MKTG INC	…	-0.011117	
218567	969904101	WILLIAMS-SONOMA INC	…	NaN	
…	…	…	…	…	
142024	92343V005	VERIZON INC/NJ	…	0.065392	
141899	225426105	CREDITRISKMONITOR.COM INC	…	NaN	
360897	31604Q107	FIDELITY BANKSHARES INC	…	NaN	
360723	738140102	POTTERS FINANCIAL CORP	…	NaN	
538146	29355M200	ENOVA SYSTEMS INC	…	NaN	

	book_leverage	wc_ta	re_ta	ebit_ta	mv_tl	sales_ta	\
466485	0.289693	0.105189	0.063853	0.071915	NaN	2.372969	
44743	0.661325	0.108622	-0.184279	0.049489	0.225831	1.812807	
419319	0.001279	0.683591	0.342687	0.146873	8.112168	1.741350	
419484	0.000000	-0.062101	0.019138	0.024123	0.321182	4.868215	
218567	0.107756	0.262663	0.352425	0.147784	5.023464	1.872939	
…	…	…	…	…	…	…	
142024	0.508104	-0.117580	0.125764	0.196023	NaN	0.607907	
141899	0.448261	-0.038001	-5.977430	-0.179871	2.635975	0.488024	
360897	0.774254	NaN	0.030634	0.010388	0.068287	0.075664	
360723	0.612572	NaN	0.056315	NaN	NaN	NaN	
538146	1.893709	0.697479	-28.075307	-0.567227	8.757904	0.931803	

```
          z_score at_rolling_avg sale_to_at_avg
466485        NaN            NaN            NaN
44743    1.965847       443.0955       2.406754
419319   8.375988       368.7035       0.704355
419484   5.044121       160.0255       5.199490
218567   6.164567       454.9285       3.042221
…             …              …              …
142024        NaN      3612.2465       1.096077
141899  -6.942848      3258.6710       0.000650
360897        NaN       963.6195       0.150988
360723        NaN      1037.1660            NaN
538146 -34.163076        77.2645       0.037313

[12211 rows x 998 columns]
```

0.9 Descriptive Stats of the Finaincal Ratios for the Year 2005

[109]:
```
describe_yearly_stats(compustat_copy, 2005)
```

```
Descriptive statistics for ch in the year 2005:
count     9520.000000
mean       327.543696
std       2009.039289
min         -0.034000
25%          3.204750
50%         17.514000
75%         76.693750
max      52999.000000
Name: ch, dtype: float64


Descriptive statistics for ivst in the year 2005:
count      9485.000000
mean        788.524066
std       11159.120611
min           0.000000
25%           0.000000
50%           0.068000
75%          18.812000
max      536022.969000
Name: ivst, dtype: float64


Descriptive statistics for rect in the year 2005:
count      8287.000000
mean       2809.503532
std       29841.469306
min           0.000000
```

```
25%           2.927500
50%          31.860000
75%         244.717500
max      865967.000000
Name: rect, dtype: float64


Descriptive statistics for invt in the year 2005:
count      8521.000000
mean        470.612351
std        8296.929817
min           0.000000
25%           0.000000
50%           2.221000
75%          44.028000
max      379751.293000
Name: invt, dtype: float64


Descriptive statistics for aco in the year 2005:
count     7529.000000
mean       104.132219
std        712.359042
min          0.000000
25%          0.224000
50%          2.911000
75%         23.026000
max      35154.000000
Name: aco, dtype: float64


Descriptive statistics for act in the year 2005:
count      6837.000000
mean        925.145416
std        4250.119189
min          -0.016000
25%          13.154000
50%          79.035000
75%         361.151000
max      135193.009000
Name: act, dtype: float64


Descriptive statistics for ppent in the year 2005:
count      9344.000000
mean        923.396905
std        4645.840609
min           0.000000
```

```
25%           2.425000
50%          17.961000
75%         172.086250
max      107010.000000
Name: ppent, dtype: float64


Descriptive statistics for ivaeq in the year 2005:
count     8840.000000
mean        92.631896
std        931.417032
min         -2.454000
25%          0.000000
50%          0.000000
75%          0.000000
max      36638.943000
Name: ivaeq, dtype: float64


Descriptive statistics for ivao in the year 2005:
count      8039.000000
mean       1856.992321
std       23461.813289
min           0.000000
25%           0.000000
50%           0.000000
75%          29.415500
max      956318.472000
Name: ivao, dtype: float64


Descriptive statistics for intan in the year 2005:
count      9384.000000
mean        552.284767
std        3656.917807
min          -2.524000
25%           0.000000
50%           5.494500
75%          96.866000
max      157646.803000
Name: intan, dtype: float64


Descriptive statistics for ao in the year 2005:
count      9621.000000
mean       1009.470150
std       11187.731453
min         -70.076000
```

```
25%           0.751000
50%          10.319000
75%          63.346000
max      409292.688000
Name: ao, dtype: float64


Descriptive statistics for at in the year 2005:
count    9.622000e+03
mean     1.138058e+04
std      8.426626e+04
min      0.000000e+00
25%      5.276625e+01
50%      4.006615e+02
75%      1.853794e+03
max      1.588785e+06
Name: at, dtype: float64


Descriptive statistics for dlc in the year 2005:
count      9536.000000
mean       1513.961285
std       16886.611443
min           0.000000
25%           0.005750
50%           2.800000
75%          42.238250
max      495844.235000
Name: dlc, dtype: float64


Descriptive statistics for ap in the year 2005:
count      8700.000000
mean       2218.428926
std       27485.853852
min           0.000000
25%           1.589750
50%          13.390500
75%         127.826500
max      809146.000000
Name: ap, dtype: float64


Descriptive statistics for txp in the year 2005:
count    7845.000000
mean       35.645469
std       259.822972
min      -288.748000
```

```
25%         0.000000
50%         0.000000
75%         2.543000
max      8988.799000
Name: txp, dtype: float64


Descriptive statistics for lco in the year 2005:
count     7528.000000
mean       302.988689
std       1717.931903
min          0.000000
25%          1.696750
50%         13.699500
75%         84.942500
max      69373.000000
Name: lco, dtype: float64


Descriptive statistics for lct in the year 2005:
count      6864.000000
mean        754.394204
std        5275.656084
min           0.000000
25%           6.062000
50%          34.848500
75%         208.833000
max      329795.000000
Name: lct, dtype: float64


Descriptive statistics for dltt in the year 2005:
count      9603.000000
mean       1818.188121
std       14943.178238
min           0.000000
25%           0.031500
50%          25.001000
75%         314.799000
max      461686.000000
Name: dltt, dtype: float64


Descriptive statistics for lo in the year 2005:
count      9622.000000
mean       2254.732909
std       26316.123229
min           0.000000
```

```
25%           0.145000
50%           5.440000
75%          62.287250
max      828017.508000
Name: lo, dtype: float64


Descriptive statistics for txditc in the year 2005:
count     7304.000000
mean       158.938219
std        987.352453
min        -70.200000
25%          0.000000
50%          0.000000
75%         11.498500
max      27370.000000
Name: txditc, dtype: float64


Descriptive statistics for mib in the year 2005:
count     9123.000000
mean        67.784260
std        614.482059
min         -7.906000
25%          0.000000
50%          0.000000
75%          0.000000
max      26754.000000
Name: mib, dtype: float64


Descriptive statistics for lt in the year 2005:
count    9.606000e+03
mean     9.925378e+03
std      7.979193e+04
min      0.000000e+00
25%      1.734925e+01
50%      2.332300e+02
75%      1.222679e+03
max      1.546795e+06
Name: lt, dtype: float64


Descriptive statistics for pstk in the year 2005:
count     9578.000000
mean        20.122255
std        277.085398
min        -69.000000
```

```
25%          0.000000
50%          0.000000
75%          0.000000
max      20000.000000
Name: pstk, dtype: float64


Descriptive statistics for ceq in the year 2005:
count      9596.000000
mean       1378.662404
std        6438.751305
min      -25869.000000
25%          14.697750
50%          97.174500
75%         519.351250
max      187589.950000
Name: ceq, dtype: float64


Descriptive statistics for teq in the year 2005:
count      3212.000000
mean       1992.585508
std        8042.410672
min       -9648.000000
25%          23.472000
50%         156.745000
75%         714.400250
max      112537.000000
Name: teq, dtype: float64


Descriptive statistics for sale in the year 2005:
count      8284.000000
mean       2472.537447
std       11517.321465
min         -26.481000
25%          20.203000
50%         135.069000
75%         893.693000
max      328213.000000
Name: sale, dtype: float64


Descriptive statistics for cogs in the year 2005:
count      8284.000000
mean       1626.270618
std        8391.265180
min         -55.186000
```

```
25%           9.888750
50%          66.474500
75%         527.953250
max      253592.000000
Name: cogs, dtype: float64


Descriptive statistics for xsga in the year 2005:
count     6861.000000
mean       407.689331
std       1898.686154
min          0.000000
25%          7.742000
50%         30.614000
75%        135.498000
max      51105.000000
Name: xsga, dtype: float64


Descriptive statistics for oibdp in the year 2005:
count     8083.000000
mean       511.604404
std       2659.992559
min      -1725.197000
25%         -0.129500
50%         17.438000
75%        140.398000
max      59255.000000
Name: oibdp, dtype: float64


Descriptive statistics for dp in the year 2005:
count     9172.00000
mean       126.00193
std        736.70406
min          0.00000
25%          0.68050
50%          4.30100
75%         33.55350
max      33750.96700
Name: dp, dtype: float64


Descriptive statistics for oiadp in the year 2005:
count     8284.000000
mean       380.191304
std       2202.786977
min      -9423.222000
```

```
25%         -0.750750
50%         12.056500
75%         99.289500
max      53230.000000
Name: oiadp, dtype: float64


Descriptive statistics for xint in the year 2005:
count     8228.000000
mean       184.238695
std       1625.757893
min         -0.145000
25%          0.227000
50%          4.306500
75%         33.702000
max      42284.229000
Name: xint, dtype: float64


Descriptive statistics for nopi in the year 2005:
count     8283.000000
mean        -2.430397
std        648.046401
min     -25843.980000
25%          0.000000
50%          0.296000
75%          3.457000
max      12159.000000
Name: nopi, dtype: float64


Descriptive statistics for spi in the year 2005:
count     9498.000000
mean       -18.446051
std        355.215102
min     -20619.000000
25%         -1.378500
50%          0.000000
75%          0.000000
max       4590.000000
Name: spi, dtype: float64


Descriptive statistics for pi in the year 2005:
count     9566.000000
mean       287.821390
std       1692.914468
min     -21176.000000
```

```
25%         -1.319500
50%          8.536000
75%         76.908500
max      60231.000000
Name: pi, dtype: float64


Descriptive statistics for txt in the year 2005:
count     9566.000000
mean        91.276776
std        563.980345
min      -5878.000000
25%         0.000000
50%          1.644000
75%         20.453750
max      23302.000000
Name: txt, dtype: float64


Descriptive statistics for mii in the year 2005:
count    8959.000000
mean        9.771105
std       101.922507
min      -447.000000
25%         0.000000
50%         0.000000
75%         0.000000
max      4269.000000
Name: mii, dtype: float64


Descriptive statistics for ib in the year 2005:
count     9568.000000
mean       187.484214
std       1151.433916
min     -21176.000000
25%         -1.366000
50%          6.219000
75%         53.112500
max      36130.000000
Name: ib, dtype: float64


Descriptive statistics for dvp in the year 2005:
count    9563.000000
mean        1.597343
std        14.787561
min       -21.197000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max       533.000000
Name: dvp, dtype: float64


Descriptive statistics for cstke in the year 2005:
count    9571.000000
mean       -0.172313
std         7.524336
min      -371.800000
25%         0.000000
50%         0.000000
75%         0.000000
max       257.782000
Name: cstke, dtype: float64


Descriptive statistics for xido in the year 2005:
count    9571.000000
mean        4.478978
std       108.672107
min     -1922.000000
25%         0.000000
50%         0.000000
75%         0.000000
max      4783.000000
Name: xido, dtype: float64


Descriptive statistics for ni in the year 2005:
count     8285.000000
mean       176.196704
std       1145.663330
min     -21176.000000
25%         -2.282000
50%          4.822000
75%         51.478000
max      36130.000000
Name: ni, dtype: float64


Descriptive statistics for ibc in the year 2005:
count     8225.000000
mean       168.766501
std       1120.530297
min     -21176.000000
```

```
25%         -2.347000
50%          4.548000
75%         48.374000
max      36130.000000
Name: ibc, dtype: float64


Descriptive statistics for dpc in the year 2005:
count     8030.000000
mean       135.209222
std        791.952621
min         -0.005000
25%          0.775000
50%          5.428000
75%         39.309750
max      33750.967000
Name: dpc, dtype: float64


Descriptive statistics for xidoc in the year 2005:
count    8224.000000
mean        2.661865
std        94.165887
min     -1066.000000
25%         0.000000
50%         0.000000
75%         0.000000
max      3854.000000
Name: xidoc, dtype: float64


Descriptive statistics for txdc in the year 2005:
count    7991.000000
mean        4.096056
std       120.369249
min     -2486.820000
25%        -0.213000
50%         0.000000
75%         0.276000
max      4494.000000
Name: txdc, dtype: float64


Descriptive statistics for esubc in the year 2005:
count    7385.000000
mean       -2.456122
std        64.034709
min     -1421.000000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max      3815.376000
Name: esubc, dtype: float64


Descriptive statistics for sppiv in the year 2005:
count    8004.000000
mean      -16.990823
std       216.479375
min     -9432.774000
25%        -0.136000
50%         0.000000
75%         0.000000
max      5217.675000
Name: sppiv, dtype: float64


Descriptive statistics for fopo in the year 2005:
count     8225.000000
mean        46.548319
std        719.674969
min     -20431.000000
25%          0.052000
50%          1.325000
75%          9.900000
max      22737.824000
Name: fopo, dtype: float64


Descriptive statistics for fopt in the year 2005:
count      5.000000
mean     334.750200
std      235.305056
min       20.223000
25%      186.632000
50%      416.292000
75%      422.010000
max      628.594000
Name: fopt, dtype: float64


Descriptive statistics for recch in the year 2005:
count    7241.000000
mean      -28.687497
std       246.173618
min     -7279.018000
```

```
25%        -9.653000
50%        -0.860000
75%         0.017000
max      4151.000000
Name: recch, dtype: float64


Descriptive statistics for invch in the year 2005:
count     7592.000000
mean       -67.301816
std       1358.312760
min     -66899.000000
25%         -2.356750
50%          0.000000
75%          0.000000
max       2865.240000
Name: invch, dtype: float64


Descriptive statistics for apalch in the year 2005:
count     5691.000000
mean        22.692302
std        249.940188
min      -2795.000000
25%         -0.152500
50%          0.616000
75%          6.993000
max      11714.106000
Name: apalch, dtype: float64


Descriptive statistics for txach in the year 2005:
count    6062.000000
mean        0.877116
std        99.750211
min     -5348.000000
25%         0.000000
50%         0.000000
75%         0.000000
max      1602.000000
Name: txach, dtype: float64


Descriptive statistics for aoloch in the year 2005:
count     8220.000000
mean         8.543744
std       2188.720306
min     -81258.620000
```

```
25%         -1.527500
50%          0.006000
75%          2.829250
max      86896.000000
Name: aoloch, dtype: float64


Descriptive statistics for oancf in the year 2005:
count     8241.000000
mean       270.278360
std       2411.835917
min     -75741.432000
25%         -0.526000
50%         10.002000
75%         93.933000
max      78470.000000
Name: oancf, dtype: float64


Descriptive statistics for ivch in the year 2005:
count      7969.000000
mean       1227.671060
std       16638.408281
min           0.000000
25%           0.000000
50%           0.000000
75%          19.000000
max      693267.000000
Name: ivch, dtype: float64


Descriptive statistics for siv in the year 2005:
count      7972.000000
mean        936.153886
std       12608.554488
min           0.000000
25%           0.000000
50%           0.000000
75%           9.153500
max      387390.000000
Name: siv, dtype: float64


Descriptive statistics for capx in the year 2005:
count     8218.000000
mean       163.702024
std        941.107774
min       -104.963000
```

```
25%          0.421000
50%          4.175500
75%         39.438000
max      32077.610000
Name: capx, dtype: float64


Descriptive statistics for sppe in the year 2005:
count     6789.000000
mean        20.052017
std        294.863775
min       -161.230000
25%          0.000000
50%          0.000000
75%          0.135000
max      15087.892000
Name: sppe, dtype: float64


Descriptive statistics for aqc in the year 2005:
count     7999.000000
mean        55.102303
std        418.929443
min      -1504.000000
25%          0.000000
50%          0.000000
75%          0.801000
max      12115.000000
Name: aqc, dtype: float64


Descriptive statistics for ivstch in the year 2005:
count      6508.000000
mean        158.730693
std        5844.935622
min      -40105.301000
25%          0.000000
50%          0.000000
75%          0.000000
max      323958.000000
Name: ivstch, dtype: float64


Descriptive statistics for ivaco in the year 2005:
count     8234.000000
mean        23.148707
std        832.145303
min     -21485.000000
```

```
25%         -0.113000
50%          0.000000
75%          0.221750
max      60152.400000
Name: ivaco, dtype: float64


Descriptive statistics for ivncf in the year 2005:
count      8240.000000
mean       -335.970977
std        3241.598988
min     -150445.000000
25%        -102.316500
50%         -11.611500
75%          -0.174000
max      139075.000000
Name: ivncf, dtype: float64


Descriptive statistics for sstk in the year 2005:
count     8138.000000
mean        45.492785
std        600.226692
min         -5.897000
25%          0.012000
50%          1.228000
75%         12.869750
max      49829.952000
Name: sstk, dtype: float64


Descriptive statistics for prstkc in the year 2005:
count     7567.000000
mean        79.609564
std        786.231923
min         -0.905000
25%          0.000000
50%          0.000000
75%          0.735000
max      51871.513000
Name: prstkc, dtype: float64


Descriptive statistics for dv in the year 2005:
count     8079.000000
mean        71.503108
std        568.550755
min          0.000000
```

```
25%          0.000000
50%          0.000000
75%          7.026000
max      36112.000000
Name: dv, dtype: float64


Descriptive statistics for dltis in the year 2005:
count      7945.000000
mean        482.512145
std        4144.247727
min          -0.498000
25%           0.000000
50%           0.369000
75%          67.043000
max      156336.000000
Name: dltis, dtype: float64


Descriptive statistics for dltr in the year 2005:
count      8022.000000
mean        395.662219
std        3722.828001
min           0.000000
25%           0.000000
50%           1.713500
75%          60.692000
max      197914.000000
Name: dltr, dtype: float64


Descriptive statistics for dlcch in the year 2005:
count      4591.000000
mean         16.663923
std        3745.468061
min     -174455.000000
25%          -0.100000
50%           0.000000
75%           0.983000
max      158585.000000
Name: dlcch, dtype: float64


Descriptive statistics for fiao in the year 2005:
count     8236.000000
mean        87.102371
std       1490.459975
min      -7501.907000
```

```
25%         -0.300250
50%          0.000000
75%          0.000000
max      70525.031000
Name: fiao, dtype: float64


Descriptive statistics for fincf in the year 2005:
count      8240.000000
mean         83.879719
std        4015.849221
min     -217380.000000
25%          -8.196250
50%           0.742500
75%          22.094750
max      170817.000000
Name: fincf, dtype: float64


Descriptive statistics for exre in the year 2005:
count    8225.000000
mean        0.153918
std        93.416346
min     -5340.312000
25%         0.000000
50%         0.000000
75%         0.000000
max      3816.550000
Name: exre, dtype: float64


Descriptive statistics for chech in the year 2005:
count     8243.000000
mean        17.726664
std        585.098967
min     -24785.000000
25%         -2.808000
50%          0.318000
75%         11.666000
max      18969.700000
Name: chech, dtype: float64


Descriptive statistics for fsrco in the year 2005:
count      5.000000
mean     142.822000
std      154.409311
min        0.000000
```

```
25%        0.000000
50%      102.946000
75%      304.888000
max      306.276000
Name: fsrco, dtype: float64


Descriptive statistics for fuseo in the year 2005:
count      5.000000
mean     281.249600
std      161.236048
min        0.000000
25%      291.217000
50%      362.421000
75%      368.314000
max      384.296000
Name: fuseo, dtype: float64


Descriptive statistics for wcapc in the year 2005:
count      3.000000
mean     365.026000
std      289.557319
min       90.025000
25%      213.928500
50%      337.832000
75%      502.526500
max      667.221000
Name: wcapc, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 2005:
count    4375.000000
mean             NaN
std              NaN
min             -inf
25%        -0.004614
50%         0.000000
75%         0.031048
max              inf
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 2005:
count    9510.000000
mean        0.143715
std        11.015774
min      -942.000000
```

```
25%         0.019636
50%         0.325763
75%         0.603285
max        46.166667
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 2005:
count    6831.000000
mean            -inf
std              NaN
min             -inf
25%         0.007840
50%         0.176826
75%         0.403260
max         1.000000
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 2005:
count    9371.000000
mean            -inf
std              NaN
min             -inf
25%        -0.577776
50%         0.018528
75%         0.139593
max         1.942625
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 2005:
count    8275.000000
mean            -inf
std              NaN
min             -inf
25%        -0.044694
50%         0.037766
75%         0.095557
max        13.739913
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 2005:
count    7273.000000
mean             inf
std              NaN
min         0.000000
```

```
25%         0.613331
50%         1.958512
75%         5.918055
max              inf
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 2005:
count    8238.000000
mean             inf
std              NaN
min        -0.135052
25%         0.174454
50%         0.627864
75%         1.230337
max              inf
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 2005:
count    5.752000e+03
mean              inf
std               NaN
min     -4.137032e+04
25%      8.750128e-01
50%      2.845987e+00
75%      5.617175e+00
max               inf
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 2005:
count    7621.000000
mean        0.775332
std         1.186316
min        -0.183503
25%         0.054803
50%         0.276750
75%         1.072723
max        17.200986
Name: sale_to_at_avg, dtype: float64


Descriptive statistics for dltis_at in the year 2005:
count    7901.000000
mean             inf
std              NaN
min        -0.029861
```

```
25%         0.000000
50%         0.003977
75%         0.103404
max              inf
Name: dltis_at, dtype: float64


Descriptive statistics for dlc_at in the year 2005:
count    9514.000000
mean             inf
std              NaN
min         0.000000
25%         0.000086
50%         0.017531
75%         0.073970
max              inf
Name: dlc_at, dtype: float64


Descriptive statistics for dltt_at in the year 2005:
count    9562.000000
mean             inf
std              NaN
min         0.000000
25%         0.000969
50%         0.083865
75%         0.257793
max              inf
Name: dltt_at, dtype: float64


Descriptive statistics for ppent_at in the year 2005:
count    9298.000000
mean        0.191425
std         0.237682
min         0.000000
25%         0.018426
50%         0.079007
75%         0.284006
max         1.000000
Name: ppent_at, dtype: float64


Descriptive statistics for mkvalt_at in the year 2005:
count    7596.000000
mean             inf
std              NaN
min         0.000020
```

```
25%         0.288357
50%         0.905100
75%         1.956237
max              inf
Name: mkvalt_at, dtype: float64


Descriptive statistics for ni_at in the year 2005:
count    8276.000000
mean             NaN
std              NaN
min             -inf
25%        -0.076299
50%         0.014180
75%         0.062957
max              inf
Name: ni_at, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 2005:
count    4375.000000
mean             NaN
std              NaN
min             -inf
25%        -0.004614
50%         0.000000
75%         0.031048
max              inf
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 2005:
count    9510.000000
mean        0.143715
std        11.015774
min      -942.000000
25%         0.019636
50%         0.325763
75%         0.603285
max        46.166667
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 2005:
count    6831.000000
mean            -inf
std              NaN
min             -inf
```

```
25%         0.007840
50%         0.176826
75%         0.403260
max         1.000000
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 2005:
count    9371.000000
mean            -inf
std              NaN
min             -inf
25%        -0.577776
50%         0.018528
75%         0.139593
max         1.942625
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 2005:
count    8275.000000
mean            -inf
std              NaN
min             -inf
25%        -0.044694
50%         0.037766
75%         0.095557
max        13.739913
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 2005:
count    7273.000000
mean             inf
std              NaN
min         0.000000
25%         0.613331
50%         1.958512
75%         5.918055
max              inf
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 2005:
count    8238.000000
mean             inf
std              NaN
min        -0.135052
```

```
25%         0.174454
50%         0.627864
75%         1.230337
max              inf
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 2005:
count    5.752000e+03
mean              inf
std               NaN
min     -4.137032e+04
25%      8.750128e-01
50%      2.845987e+00
75%      5.617175e+00
max               inf
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 2005:
count    7621.000000
mean        0.775332
std         1.186316
min        -0.183503
25%         0.054803
50%         0.276750
75%         1.072723
max        17.200986
Name: sale_to_at_avg, dtype: float64
```

[109]:

	gvkey	datadate	fyear	indfmt	consol	popsrc	datafmt	tic	\
85365	5109	2005-01-31	2004	INDL	C	D	STD	GCO	
407896	62829	2005-01-31	2004	INDL	C	D	STD	3BETM	
311437	22495	2005-01-31	2004	INDL	C	D	STD	CYUR	
445739	116104	2005-01-31	2004	INDL	C	D	STD	0190A	
487795	161036	2005-01-31	2004	INDL	C	D	STD	MFRM.1	
…	…	…	…	…	…	…	…	…	
358054	29246	2005-12-31	2005	FS	C	D	STD	AF	
358053	29246	2005-12-31	2005	INDL	C	D	STD	AF	
357778	29211	2005-12-31	2005	FS	C	D	STD	BPFH	
358776	29312	2005-12-31	2005	INDL	C	D	STD	HGSI	
538198	279431	2005-12-31	2005	INDL	C	D	STD	SOL	

	cusip	conm	…	net_debt_issued_ratio	\
85365	371532102	GENESCO INC	…	NaN	

```
407896  030405104         AMERICAN WAGERING INC  …                  NaN
311437  23254T101          CYCLONE URANIUM CORP  …             3.181818
445739  46699D001              J CREW GROUP INC  …                  NaN
487795  57722W106    MATTRESS FIRM HOLDING CORP  …                  NaN
…          …                               …  …                  …
358054  046265104        ASTORIA FINANCIAL CORP  …                  NaN
358053  046265104        ASTORIA FINANCIAL CORP  …            -0.068364
357778  101119105  BOSTON PRIVATE FINL HOLDINGS  …                  NaN
358776  444903108     HUMAN GENOME SCIENCES INC  …                  NaN
538198  75971T301              EMEREN GROUP LTD  …             0.045034

       book_leverage      wc_ta       re_ta   ebit_ta     mv_tl  sales_ta  \
85365       0.372131   0.277302    0.236822  0.140480  1.786289  1.750679
407896      0.162264   0.456278   -1.390911 -0.018953  0.500041  1.196209
311437     -0.218321 -29.018182 -320.200000 -7.418182  2.105079  0.000000
445739      6.554493   0.043739   -2.342117  0.135308       NaN  2.890846
487795           NaN        NaN         NaN       NaN       NaN       NaN
…                …          …           …         …         …         …
358054      0.854623        NaN    0.077094       NaN       NaN       NaN
358053      0.854623        NaN    0.077094  0.017154  0.137706  0.052957
357778      0.568631        NaN    0.028189       NaN       NaN       NaN
358776      0.551843   0.129264   -1.374955 -0.247501  1.933851  0.019170
538198      0.208492  -0.054280    0.119495  0.059847       NaN  0.505816

          z_score at_rolling_avg sale_to_at_avg
85365     3.932842            NaN            NaN
407896    0.021984       322.4285       0.034451
311437 -506.318771         4.6705       0.000000
445739         NaN       139.1245       5.780549
487795         NaN            NaN            NaN
…              …              …              …
358054         NaN     11283.0585            NaN
358053         NaN     22380.2710       0.052957
357778         NaN     13757.1680            NaN
358776   -1.407285      3065.5555       0.006235
538198         NaN       503.5525       0.010104

[10650 rows x 998 columns]
```

0.10 Descriptive Stats of the Finaincal Ratios for the Year 2010

[110]:
```
describe_yearly_stats(compustat_copy, 2010)
```

```
Descriptive statistics for ch in the year 2010:
count      8202.000000
mean        693.145880
std        4936.274067
min          -0.008000
```

```
25%           4.863250
50%          26.824000
75%         139.596500
max      152402.349000
Name: ch, dtype: float64


Descriptive statistics for ivst in the year 2010:
count      8157.000000
mean       1167.825169
std       14529.557431
min           0.000000
25%           0.000000
50%           0.174000
75%          26.289000
max      431810.414000
Name: ivst, dtype: float64


Descriptive statistics for rect in the year 2010:
count    7.093000e+03
mean     4.550004e+03
std      6.058447e+04
min      0.000000e+00
25%      3.865000e+00
50%      4.679600e+01
75%      3.265430e+02
max      3.053530e+06
Name: rect, dtype: float64


Descriptive statistics for invt in the year 2010:
count      7351.000000
mean        552.420654
std        8955.133575
min           0.000000
25%           0.000000
50%           4.037000
75%          70.548000
max      373176.433000
Name: invt, dtype: float64


Descriptive statistics for aco in the year 2010:
count     6456.000000
mean       156.308905
std        918.226729
min          0.000000
```

```
25%          0.345000
50%          5.055500
75%         39.217750
max      23974.429000
Name: aco, dtype: float64


Descriptive statistics for act in the year 2010:
count      5865.000000
mean       1404.859095
std        5854.138816
min           0.000000
25%          17.627000
50%         114.732000
75%         589.459000
max      139974.000000
Name: act, dtype: float64


Descriptive statistics for ppent in the year 2010:
count      8040.000000
mean       1478.301562
std        7806.195865
min           0.000000
25%           2.963750
50%          26.610500
75%         274.248250
max      218567.000000
Name: ppent, dtype: float64


Descriptive statistics for ivaeq in the year 2010:
count     7670.000000
mean       157.651850
std       1392.842143
min         -2.122000
25%          0.000000
50%          0.000000
75%          0.000000
max      55242.112000
Name: ivaeq, dtype: float64


Descriptive statistics for ivao in the year 2010:
count    6.876000e+03
mean     2.695174e+03
std      3.276131e+04
min      0.000000e+00
```

```
25%      0.000000e+00
50%      0.000000e+00
75%      5.150950e+01
max      1.064353e+06
Name: ivao, dtype: float64


Descriptive statistics for intan in the year 2010:
count      8118.000000
mean        968.608604
std        5342.343819
min           0.000000
25%           0.000000
50%           8.685500
75%         157.792750
max      134121.000000
Name: intan, dtype: float64


Descriptive statistics for ao in the year 2010:
count    8.297000e+03
mean     2.014877e+03
std      2.747490e+04
min     -2.091400e+01
25%      1.063000e+00
50%      1.643600e+01
75%      9.984300e+01
max      1.160243e+06
Name: ao, dtype: float64


Descriptive statistics for at in the year 2010:
count    8.297000e+03
mean     1.830786e+04
std      1.460826e+05
min      0.000000e+00
25%      8.001000e+01
50%      5.570530e+02
75%      2.769312e+03
max      3.221972e+06
Name: at, dtype: float64


Descriptive statistics for dlc in the year 2010:
count      8240.000000
mean       1743.304213
std       19901.437293
min         -56.000000
```

```
25%           0.000000
50%           3.538500
75%          52.967500
max      503984.874000
Name: dlc, dtype: float64


Descriptive statistics for ap in the year 2010:
count    7.538000e+03
mean     3.305996e+03
std      4.371992e+04
min      0.000000e+00
25%      1.969000e+00
50%      1.865950e+01
75%      1.876405e+02
max      1.450455e+06
Name: ap, dtype: float64


Descriptive statistics for txp in the year 2010:
count    6757.000000
mean       38.776538
std       297.245956
min      -211.000000
25%         0.000000
50%         0.000000
75%         2.192000
max      9812.000000
Name: txp, dtype: float64


Descriptive statistics for lco in the year 2010:
count     6456.000000
mean       458.695157
std       2177.883339
min          0.000000
25%          2.359000
50%         21.007500
75%        134.734000
max      50041.000000
Name: lco, dtype: float64


Descriptive statistics for lct in the year 2010:
count      5881.000000
mean       1048.457859
std        4851.739184
min           0.000000
```

```
25%           8.023000
50%          53.264000
75%         326.402000
max      114414.000000
Name: lct, dtype: float64


Descriptive statistics for dltt in the year 2010:
count    8.278000e+03
mean     3.829655e+03
std      5.890887e+04
min      0.000000e+00
25%      0.000000e+00
50%      3.442100e+01
75%      4.794150e+02
max      3.039757e+06
Name: dltt, dtype: float64


Descriptive statistics for lo in the year 2010:
count    8.297000e+03
mean     3.755596e+03
std      4.442800e+04
min      0.000000e+00
25%      8.450000e-01
50%      1.103100e+01
75%      1.160210e+02
max      1.388511e+06
Name: lo, dtype: float64


Descriptive statistics for txditc in the year 2010:
count     6197.000000
mean       240.764902
std       1325.030709
min         -1.000000
25%          0.000000
50%          0.000000
75%         25.269000
max      35558.000000
Name: txditc, dtype: float64


Descriptive statistics for mib in the year 2010:
count     8275.000000
mean        14.391704
std        580.727658
min         -1.470000
```

```
25%          0.000000
50%          0.000000
75%          0.000000
max      44466.116000
Name: mib, dtype: float64


Descriptive statistics for lt in the year 2010:
count    8.280000e+03
mean     1.584832e+04
std      1.385952e+05
min      0.000000e+00
25%      2.822725e+01
50%      3.262990e+02
75%      1.810169e+03
max      3.224489e+06
Name: lt, dtype: float64


Descriptive statistics for pstk in the year 2010:
count      8268.000000
mean         97.141964
std        2263.205717
min        -109.000000
25%           0.000000
50%           0.000000
75%           0.000000
max      108804.000000
Name: pstk, dtype: float64


Descriptive statistics for ceq in the year 2010:
count      8278.000000
mean       2235.495786
std       11008.105372
min     -111403.000000
25%          19.798000
50%         137.138500
75%         789.542500
max      211686.000000
Name: ceq, dtype: float64


Descriptive statistics for teq in the year 2010:
count      8295.000000
mean       2474.758436
std       11616.542204
min      -13873.000000
```

```
25%          23.974000
50%         151.805000
75%         830.500000
max      228248.000000
Name: teq, dtype: float64


Descriptive statistics for sale in the year 2010:
count      7110.000000
mean       3517.745182
std       15679.396918
min       -1208.806000
25%          27.404250
50%         191.726000
75%        1220.118250
max      406103.000000
Name: sale, dtype: float64


Descriptive statistics for cogs in the year 2010:
count      7110.000000
mean       2323.208650
std       11600.054436
min        -310.359000
25%          12.894750
50%          93.427000
75%         732.703000
max      307749.000000
Name: cogs, dtype: float64


Descriptive statistics for xsga in the year 2010:
count     6022.000000
mean       564.259769
std       2651.098674
min          0.000000
25%          9.134250
50%         37.947500
75%        182.493750
max      79347.000000
Name: xsga, dtype: float64


Descriptive statistics for oibdp in the year 2010:
count      6917.000000
mean        726.850751
std        3742.914983
min      -14081.063000
```

```
25%          -0.025000
50%          24.676000
75%         210.026000
max      124840.000000
Name: oibdp, dtype: float64


Descriptive statistics for dp in the year 2010:
count     7895.000000
mean       185.123785
std        910.265272
min         -0.244000
25%          0.810000
50%          6.530000
75%         53.827000
max      21542.441000
Name: dp, dtype: float64


Descriptive statistics for oiadp in the year 2010:
count      7110.000000
mean        525.719374
std        3113.830048
min      -14319.905000
25%          -0.689750
50%          15.517500
75%         144.123000
max      124840.000000
Name: oiadp, dtype: float64


Descriptive statistics for xint in the year 2010:
count      7222.000000
mean        231.750418
std        3010.844585
min           0.000000
25%           0.248000
50%           5.531500
75%          46.629000
max      137861.000000
Name: xint, dtype: float64


Descriptive statistics for nopi in the year 2010:
count     7110.000000
mean       -17.840682
std       1188.753985
min     -65729.318000
```

```
25%         -0.293500
50%          0.065000
75%          2.553750
max      13628.000000
Name: nopi, dtype: float64


Descriptive statistics for spi in the year 2010:
count     8199.000000
mean       -22.005132
std        625.511568
min     -41250.000000
25%         -2.915500
50%          0.000000
75%          0.000000
max      15927.000000
Name: spi, dtype: float64


Descriptive statistics for pi in the year 2010:
count     8263.000000
mean       359.701051
std       2006.451470
min     -16017.010000
25%         -3.073500
50%          7.256000
75%         90.538000
max      52959.000000
Name: pi, dtype: float64


Descriptive statistics for txt in the year 2010:
count     8263.000000
mean       101.721788
std        628.620785
min      -2268.999000
25%          0.000000
50%          1.041000
75%         20.546500
max      21561.000000
Name: txt, dtype: float64


Descriptive statistics for mii in the year 2010:
count    7893.000000
mean       15.700436
std       189.926134
min     -1815.657000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max      7668.000000
Name: mii, dtype: float64


Descriptive statistics for ib in the year 2010:
count     8263.000000
mean       243.019202
std       1402.005617
min     -14025.000000
25%         -2.916000
50%          5.239000
75%         65.898500
max      31740.000000
Name: ib, dtype: float64


Descriptive statistics for dvp in the year 2010:
count    8260.000000
mean        7.376872
std       155.659970
min      -112.114000
25%         0.000000
50%         0.000000
75%         0.000000
max      7704.000000
Name: dvp, dtype: float64


Descriptive statistics for cstke in the year 2010:
count    8263.000000
mean       -0.942330
std        26.012301
min     -1117.000000
25%         0.000000
50%         0.000000
75%         0.000000
max       440.509000
Name: cstke, dtype: float64


Descriptive statistics for xido in the year 2010:
count     8264.000000
mean         5.519623
std        319.442816
min      -2118.000000
```

```
25%          0.000000
50%          0.000000
75%          0.000000
max      27170.456000
Name: xido, dtype: float64


Descriptive statistics for ni in the year 2010:
count     7110.000000
mean       252.539778
std       1456.021407
min     -14025.000000
25%         -3.204000
50%          5.050000
75%         68.285000
max      36538.585000
Name: ni, dtype: float64


Descriptive statistics for ibc in the year 2010:
count     7057.000000
mean       260.496309
std       1453.161273
min     -14026.000000
25%         -3.303000
50%          5.092000
75%         68.606000
max      32704.000000
Name: ibc, dtype: float64


Descriptive statistics for dpc in the year 2010:
count     6900.000000
mean       197.233479
std        959.879713
min        -45.500000
25%          0.953750
50%          8.473000
75%         62.908250
max      21542.441000
Name: dpc, dtype: float64


Descriptive statistics for xidoc in the year 2010:
count     7053.000000
mean         7.118885
std        349.265869
min       -405.000000
```

```
25%          0.000000
50%          0.000000
75%          0.000000
max      28109.724000
Name: xidoc, dtype: float64


Descriptive statistics for txdc in the year 2010:
count    6760.000000
mean        9.182948
std       152.029764
min     -3280.000000
25%        -0.371500
50%         0.000000
75%         0.897000
max      4287.000000
Name: txdc, dtype: float64


Descriptive statistics for esubc in the year 2010:
count    6280.000000
mean       -6.242506
std       122.857956
min     -7201.201000
25%         0.000000
50%         0.000000
75%         0.000000
max      1286.000000
Name: esubc, dtype: float64


Descriptive statistics for sppiv in the year 2010:
count     6783.000000
mean       -21.434008
std        455.413930
min     -26151.137000
25%         -0.139000
50%          0.000000
75%          0.001000
max       6325.320000
Name: sppiv, dtype: float64


Descriptive statistics for fopo in the year 2010:
count     7060.000000
mean       105.352052
std       1107.953124
min     -17584.079000
```

```
25%          0.231250
50%          3.599500
75%         21.474500
max      42781.000000
Name: fopo, dtype: float64


Descriptive statistics for fopt in the year 2010:
count     1.00
mean     18.72
std        NaN
min      18.72
25%      18.72
50%      18.72
75%      18.72
max      18.72
Name: fopt, dtype: float64


Descriptive statistics for recch in the year 2010:
count     6164.000000
mean       -46.827212
std        624.641651
min     -32380.257000
25%        -11.988000
50%         -0.511000
75%          0.201000
max       4027.031000
Name: recch, dtype: float64


Descriptive statistics for invch in the year 2010:
count     6524.000000
mean       -33.663203
std       1737.915265
min     -74070.000000
25%         -2.820000
50%          0.000000
75%          0.007000
max      89502.473000
Name: invch, dtype: float64


Descriptive statistics for apalch in the year 2010:
count     4617.000000
mean        27.820246
std        696.541378
min     -36503.886000
```

```
25%         -0.394000
50%          0.501000
75%          8.500000
max       9371.212000
Name: apalch, dtype: float64


Descriptive statistics for txach in the year 2010:
count    5033.000000
mean        1.381589
std        69.362820
min     -2136.000000
25%         0.000000
50%         0.000000
75%         0.000000
max      2348.000000
Name: txach, dtype: float64


Descriptive statistics for aoloch in the year 2010:
count      7056.000000
mean         37.839142
std        2046.874258
min      -51791.000000
25%          -2.427750
50%           0.016000
75%           4.956750
max      103075.000000
Name: aoloch, dtype: float64


Descriptive statistics for oancf in the year 2010:
count      7072.000000
mean        535.294706
std        3433.117440
min      -41890.554000
25%          -0.066250
50%          18.997500
75%         155.819500
max      143819.272000
Name: oancf, dtype: float64


Descriptive statistics for ivch in the year 2010:
count    6.792000e+03
mean     1.401307e+03
std      2.385245e+04
min      0.000000e+00
```

```
25%      0.000000e+00
50%      0.000000e+00
75%      2.577125e+01
max      1.459552e+06
Name: ivch, dtype: float64


Descriptive statistics for siv in the year 2010:
count      6824.000000
mean       1259.247897
std       19408.916869
min           0.000000
25%           0.000000
50%           0.000000
75%          17.373500
max      861875.320000
Name: siv, dtype: float64


Descriptive statistics for capx in the year 2010:
count     7051.00000
mean       239.75001
std       1405.34420
min         -0.56400
25%          0.31850
50%          5.06200
75%         50.11650
max      45078.00000
Name: capx, dtype: float64


Descriptive statistics for sppe in the year 2010:
count    5782.000000
mean       23.985941
std       321.054926
min      -149.000000
25%         0.000000
50%         0.000000
75%         0.071000
max      8627.000000
Name: sppe, dtype: float64


Descriptive statistics for aqc in the year 2010:
count     6855.000000
mean        51.167547
std        441.662264
min     -11384.802000
```

```
25%          0.000000
50%          0.000000
75%          0.216000
max      15924.000000
Name: aqc, dtype: float64


Descriptive statistics for ivstch in the year 2010:
count      5296.000000
mean        206.801720
std        7778.327009
min      -39752.000000
25%          0.000000
50%          0.000000
75%          0.000000
max      482649.593000
Name: ivstch, dtype: float64


Descriptive statistics for ivaco in the year 2010:
count     7070.000000
mean        31.149561
std       1491.802130
min     -28150.000000
25%         -0.163500
50%          0.000000
75%          0.739500
max      94531.000000
Name: ivaco, dtype: float64


Descriptive statistics for ivncf in the year 2010:
count      7072.000000
mean       -214.984353
std        8570.806588
min     -151536.702000
25%        -103.775000
50%         -11.009000
75%          -0.039000
max      540050.000000
Name: ivncf, dtype: float64


Descriptive statistics for sstk in the year 2010:
count     6982.000000
mean        76.076939
std        840.038231
min       -120.611000
```

```
25%          0.000000
50%          0.485500
75%         10.097500
max      31286.975000
Name: sstk, dtype: float64


Descriptive statistics for prstkc in the year 2010:
count     6686.000000
mean        70.034595
std        666.388566
min         -3.715000
25%          0.000000
50%          0.000000
75%          0.539000
max      28654.072000
Name: prstkc, dtype: float64


Descriptive statistics for dv in the year 2010:
count    6968.000000
mean       94.339592
std       511.728636
min        -1.506000
25%         0.000000
50%         0.000000
75%         8.813750
max      9916.000000
Name: dv, dtype: float64


Descriptive statistics for dltis in the year 2010:
count    6.842000e+03
mean     8.672363e+02
std      1.858494e+04
min     -5.040000e+01
25%      0.000000e+00
50%      0.000000e+00
75%      8.182850e+01
max      1.211350e+06
Name: dltis, dtype: float64


Descriptive statistics for dltr in the year 2010:
count    6.954000e+03
mean     1.011810e+03
std      2.531760e+04
min      0.000000e+00
```

```
25%      0.000000e+00
50%      3.102500e+00
75%      1.002747e+02
max      1.642019e+06
Name: dltr, dtype: float64


Descriptive statistics for dlcch in the year 2010:
count     3989.000000
mean      -102.176000
std       2683.548803
min     -93708.000000
25%         -0.400000
50%          0.000000
75%          0.085000
max      13885.000000
Name: dlcch, dtype: float64


Descriptive statistics for fiao in the year 2010:
count     7070.000000
mean        14.174148
std       1347.213793
min     -11436.000000
25%         -3.002250
50%          0.000000
75%          0.000000
max      86952.686000
Name: fiao, dtype: float64


Descriptive statistics for fincf in the year 2010:
count      7072.000000
mean       -286.257127
std        8294.554504
min     -502299.000000
25%         -39.399750
50%          -0.067500
75%           9.084500
max       88814.539000
Name: fincf, dtype: float64


Descriptive statistics for exre in the year 2010:
count     7063.000000
mean         1.205242
std        224.096637
min     -13000.320000
```

```
25%          0.000000
50%          0.000000
75%          0.000000
max       6578.014000
Name: exre, dtype: float64


Descriptive statistics for chech in the year 2010:
count     7073.000000
mean        35.252195
std       1365.090721
min     -38729.557000
25%         -5.601000
50%          0.214000
75%         18.962000
max      56918.523000
Name: chech, dtype: float64


Descriptive statistics for fsrco in the year 2010:
count    1.0
mean     0.0
std      NaN
min      0.0
25%      0.0
50%      0.0
75%      0.0
max      0.0
Name: fsrco, dtype: float64


Descriptive statistics for fuseo in the year 2010:
count    1.0
mean     0.0
std      NaN
min      0.0
25%      0.0
50%      0.0
75%      0.0
max      0.0
Name: fuseo, dtype: float64


Descriptive statistics for wcapc in the year 2010:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: wcapc, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 2010:
count    3842.000000
mean             NaN
std              NaN
min             -inf
25%        -0.011580
50%         0.000000
75%         0.012521
max              inf
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 2010:
count    8210.000000
mean        0.268284
std         4.997790
min      -170.281250
25%         0.016472
50%         0.305095
75%         0.562078
max       270.000000
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 2010:
count    5853.000000
mean            -inf
std              NaN
min             -inf
25%         0.008679
50%         0.168913
75%         0.389815
max         1.000000
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 2010:
count    8034.000000
mean            -inf
std              NaN
min             -inf
```

```
25%        -0.475708
50%         0.016058
75%         0.168760
max         2.359896
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 2010:
count    7104.000000
mean            -inf
std              NaN
min             -inf
25%        -0.027716
50%         0.036973
75%         0.091277
max       182.521479
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 2010:
count    6277.000000
mean             inf
std              NaN
min         0.000009
25%         0.518765
50%         1.696779
75%         5.161789
max              inf
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 2010:
count    7065.000000
mean             inf
std              NaN
min       -11.538462
25%         0.147860
50%         0.573641
75%         1.121052
max              inf
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 2010:
count    4.918000e+03
mean              inf
std               NaN
min     -6.644296e+04
```

```
25%      6.983377e-01
50%      2.611402e+00
75%      4.897707e+00
max               inf
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 2010:
count    6173.000000
mean        0.718027
std         1.231822
min        -0.115585
25%         0.049554
50%         0.245742
75%         0.963509
max        34.535622
Name: sale_to_at_avg, dtype: float64


Descriptive statistics for dltis_at in the year 2010:
count    6796.000000
mean             inf
std              NaN
min        -0.090762
25%         0.000000
50%         0.000000
75%         0.073770
max              inf
Name: dltis_at, dtype: float64


Descriptive statistics for dlc_at in the year 2010:
count    8211.000000
mean             inf
std              NaN
min        -0.022056
25%         0.000000
50%         0.013002
75%         0.058080
max              inf
Name: dlc_at, dtype: float64


Descriptive statistics for dltt_at in the year 2010:
count    8229.000000
mean             inf
std              NaN
min         0.000000
```

```
25%         0.000021
50%         0.075641
75%         0.262634
max              inf
Name: dltt_at, dtype: float64


Descriptive statistics for ppent_at in the year 2010:
count    7989.000000
mean        0.201791
std         0.255414
min         0.000000
25%         0.016438
50%         0.078946
75%         0.300760
max         1.000000
Name: ppent_at, dtype: float64


Descriptive statistics for mkvalt_at in the year 2010:
count    6502.000000
mean             inf
std              NaN
min         0.000043
25%         0.220344
50%         0.740885
75%         1.651010
max              inf
Name: mkvalt_at, dtype: float64


Descriptive statistics for ni_at in the year 2010:
count    7105.000000
mean             NaN
std              NaN
min             -inf
25%        -0.050252
50%         0.012931
75%         0.059420
max              inf
Name: ni_at, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 2010:
count    3842.000000
mean             NaN
std              NaN
min             -inf
```

```
25%        -0.011580
50%         0.000000
75%         0.012521
max              inf
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 2010:
count    8210.000000
mean        0.268284
std         4.997790
min      -170.281250
25%         0.016472
50%         0.305095
75%         0.562078
max       270.000000
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 2010:
count    5853.000000
mean            -inf
std              NaN
min             -inf
25%         0.008679
50%         0.168913
75%         0.389815
max         1.000000
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 2010:
count    8034.000000
mean            -inf
std              NaN
min             -inf
25%        -0.475708
50%         0.016058
75%         0.168760
max         2.359896
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 2010:
count    7104.000000
mean            -inf
std              NaN
min             -inf
```

```
25%        -0.027716
50%         0.036973
75%         0.091277
max       182.521479
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 2010:
count    6277.000000
mean             inf
std              NaN
min         0.000009
25%         0.518765
50%         1.696779
75%         5.161789
max              inf
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 2010:
count    7065.000000
mean             inf
std              NaN
min       -11.538462
25%         0.147860
50%         0.573641
75%         1.121052
max              inf
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 2010:
count    4.918000e+03
mean              inf
std               NaN
min     -6.644296e+04
25%      6.983377e-01
50%      2.611402e+00
75%      4.897707e+00
max               inf
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 2010:
count    6173.000000
mean        0.718027
std         1.231822
min        -0.115585
```

```
25%         0.049554
50%         0.245742
75%         0.963509
max        34.535622
Name: sale_to_at_avg, dtype: float64
```

[110]:

	gvkey	datadate	fyear	indfmt	consol	popsrc	datafmt	tic	\
269709	15520	2010-01-31	2009	INDL	C	D	STD	SIG	
218577	11511	2010-01-31	2009	INDL	C	D	STD	WSM	
220200	11584	2010-01-31	2009	INDL	C	D	STD	FL	
72042	4523	2010-01-31	2009	INDL	C	D	STD	JAS	
221748	11672	2010-01-31	2009	INDL	C	D	STD	TJX	
…	…	…	…	…	…	…	…	…	
402409	61885	2010-12-31	2010	INDL	C	D	STD	CPBK	
402319	61871	2010-12-31	2010	INDL	C	D	STD	PPDI	
402227	61858	2010-12-31	2010	INDL	C	D	STD	BCRA	
402800	61976	2010-12-31	2010	INDL	C	D	STD	HRBR	
538865	311524	2010-12-31	2010	INDL	C	D	STD	TAM	

	cusip	conm	…	net_debt_issued_ratio	\
269709	G81276100	SIGNET JEWELERS LTD	…	-0.083236	
218577	969904101	WILLIAMS-SONOMA INC	…	NaN	
220200	344849104	FOOT LOCKER INC	…	-0.001065	
72042	47758P307	JO-ANN STORES INC	…	NaN	
221748	872540109	TJX COS INC (THE)	…	0.050712	
…	…	…	…	…	
402409	20363C102	COMMUNITY CAPITAL CORP	…	0.000000	
402319	717124101	PHARMACEUTICAL PROD DEV INC	…	0.000000	
402227	090935305	BIOCORAL INC	…	0.206275	
402800	41150R102	HARBOR DIVERSIFIED INC	…	0.000000	
538865	87509U106	TAMINCO CORP	…	NaN	

	book_leverage	wc_ta	re_ta	ebit_ta	mv_tl	sales_ta	\
269709	0.152755	0.620512	0.551741	0.050270	2.076407	1.125333	
218577	0.008396	0.296614	0.366337	0.057495	2.340018	1.492281	
220200	0.066155	0.475497	0.476562	0.030185	2.036115	1.723722	
72042	0.077475	0.358756	0.376849	0.114054	2.111509	1.989904	
221748	0.215254	0.255744	0.332248	0.267293	3.401482	2.718181	
…	…	…	…	…	…	…	
402409	0.690401	NaN	-0.026907	-0.002878	0.044445	0.062200	
402319	0.000000	0.261038	0.333838	0.094100	4.606470	0.738221	
402227	-3.235242	-1.419226	-15.718291	-0.300401	22.032408	0.205607	
402800	0.000000	0.790190	-42.393209	-1.071194	4.304740	0.000000	
538865	NaN	NaN	NaN	NaN	NaN	NaN	

```
          z_score at_rolling_avg sale_to_at_avg
269709   4.042867            NaN            NaN
218577   3.939912      2501.6845       1.240246
220200   4.265547      2447.5845       1.983180
72042    4.571387      1908.2000       1.043234
221748   6.385997      4232.1885       4.793842
…             …             …              …
402409       NaN       655.9340       0.062200
402319   4.585869      1323.9900       1.110711
402227 -11.277004       996.7720       0.000309
402800 -59.354361         3.7970       0.000000
538865       NaN            NaN            NaN

[10177 rows x 998 columns]
```

0.11 Descriptive Stats of the Finaincal Ratios for the Year 2015

[111]:
```
describe_yearly_stats(compustat_copy, 2015)
```

```
Descriptive statistics for ch in the year 2015:
count      8048.000000
mean        953.473142
std        9003.169612
min          -1.346000
25%           5.748750
50%          32.404500
75%         158.804250
max      331351.941000
Name: ch, dtype: float64


Descriptive statistics for ivst in the year 2015:
count      8004.000000
mean       1196.345451
std       14482.813409
min           0.000000
25%           0.000000
50%           0.375000
75%          31.199750
max      552590.000000
Name: ivst, dtype: float64


Descriptive statistics for rect in the year 2015:
count    6.992000e+03
mean     4.643478e+03
std      5.821509e+04
min      0.000000e+00
```

```
25%      2.891000e+00
50%      5.647600e+01
75%      4.076287e+02
max      3.053092e+06
Name: rect, dtype: float64


Descriptive statistics for invt in the year 2015:
count      7254.000000
mean        517.552944
std        6836.725544
min           0.000000
25%           0.000000
50%           2.433500
75%          70.657000
max      313463.000000
Name: invt, dtype: float64


Descriptive statistics for aco in the year 2015:
count     6354.000000
mean       184.162564
std       1292.781527
min        -47.800000
25%          0.296000
50%          4.906000
75%         44.846000
max      51300.000000
Name: aco, dtype: float64


Descriptive statistics for act in the year 2015:
count      5669.000000
mean       1651.868729
std        6859.756114
min          -1.124000
25%          17.947000
50%         147.989000
75%         699.920000
max      157875.743000
Name: act, dtype: float64


Descriptive statistics for ppent in the year 2015:
count      7888.000000
mean       1751.930519
std        9225.369329
min           0.000000
```

```
25%           2.397000
50%          28.806500
75%         365.865250
max      275541.851000
Name: ppent, dtype: float64


Descriptive statistics for ivaeq in the year 2015:
count     7608.000000
mean       178.020948
std       1322.318920
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max      32410.000000
Name: ivaeq, dtype: float64


Descriptive statistics for ivao in the year 2015:
count      6792.000000
mean       2898.961507
std       31021.640394
min           0.000000
25%           0.000000
50%           0.000000
75%          79.073000
max      890282.717000
Name: ivao, dtype: float64


Descriptive statistics for intan in the year 2015:
count      8029.000000
mean       1187.152249
std        6083.955844
min           0.000000
25%           0.000000
50%          13.235000
75%         251.600000
max      225278.000000
Name: intan, dtype: float64


Descriptive statistics for ao in the year 2015:
count      8166.000000
mean       2033.329360
std       22969.210204
min          -3.623000
```

```
25%           1.104500
50%          19.205000
75%         125.546250
max      757767.082000
Name: ao, dtype: float64


Descriptive statistics for at in the year 2015:
count    8.166000e+03
mean     1.954542e+04
std      1.386416e+05
min      0.000000e+00
25%      1.018810e+02
50%      7.712845e+02
75%      3.890376e+03
max      3.221917e+06
Name: at, dtype: float64


Descriptive statistics for dlc in the year 2015:
count      8120.000000
mean       1209.351259
std       11760.944430
min           0.000000
25%           0.000000
50%           4.922000
75%          77.529500
max      282320.547000
Name: dlc, dtype: float64


Descriptive statistics for ap in the year 2015:
count    7.413000e+03
mean     3.928485e+03
std      4.790099e+04
min      0.000000e+00
25%      2.051000e+00
50%      2.325500e+01
75%      2.321000e+02
max      1.433738e+06
Name: ap, dtype: float64


Descriptive statistics for txp in the year 2015:
count    6629.000000
mean       30.152623
std       222.544238
min      -643.000000
```

```
25%         0.000000
50%         0.000000
75%         1.951000
max      8714.000000
Name: txp, dtype: float64


Descriptive statistics for lco in the year 2015:
count     6354.000000
mean       545.933505
std       2622.985670
min          0.000000
25%          2.211000
50%         25.383000
75%        175.387750
max      60142.572000
Name: lco, dtype: float64


Descriptive statistics for lct in the year 2015:
count      5676.000000
mean       1292.771841
std        5832.989548
min           0.000000
25%           7.718250
50%          66.069500
75%         438.794000
max      161244.205000
Name: lct, dtype: float64


Descriptive statistics for dltt in the year 2015:
count    8.138000e+03
mean     3.904872e+03
std      5.818005e+04
min      0.000000e+00
25%      8.250000e-03
50%      4.983300e+01
75%      8.321640e+02
max      3.125721e+06
Name: dltt, dtype: float64


Descriptive statistics for lo in the year 2015:
count      8166.000000
mean       3807.395130
std       38325.159751
min          -0.020000
```

```
25%           1.028000
50%          14.585500
75%         146.215500
max      813702.877000
Name: lo, dtype: float64


Descriptive statistics for txditc in the year 2015:
count     6098.000000
mean       307.874736
std       1873.713982
min          0.000000
25%          0.000000
50%          0.000000
75%         35.983250
max      63199.000000
Name: txditc, dtype: float64


Descriptive statistics for mib in the year 2015:
count     8154.000000
mean        22.177216
std        818.290529
min         -3.507000
25%          0.000000
50%          0.000000
75%          0.000000
max      61621.568000
Name: mib, dtype: float64


Descriptive statistics for lt in the year 2015:
count    8.156000e+03
mean     1.663827e+04
std      1.293373e+05
min      0.000000e+00
25%      3.423300e+01
50%      4.845935e+02
75%      2.599157e+03
max      3.217858e+06
Name: lt, dtype: float64


Descriptive statistics for pstk in the year 2015:
count      8130.000000
mean        112.984647
std        2650.650630
min        -243.000000
```

```
25%           0.000000
50%           0.000000
75%           0.000000
max      136279.000000
Name: pstk, dtype: float64


Descriptive statistics for ceq in the year 2015:
count      8144.000000
mean       2632.370315
std       12824.078059
min     -132249.000000
25%          23.285500
50%         182.329500
75%        1053.712000
max      255550.000000
Name: ceq, dtype: float64


Descriptive statistics for teq in the year 2015:
count      8166.000000
mean       2905.143936
std       13498.528113
min      -50391.000000
25%          28.512500
50%         195.781500
75%        1118.930750
max      258627.000000
Name: teq, dtype: float64


Descriptive statistics for sale in the year 2015:
count      7005.000000
mean       3702.124291
std       15752.738848
min       -1269.295000
25%          18.978000
50%         221.987000
75%        1538.452000
max      483521.000000
Name: sale, dtype: float64


Descriptive statistics for cogs in the year 2015:
count      7006.000000
mean       2459.497824
std       11773.571371
min        -669.218000
```

```
25%           7.790250
50%         100.829500
75%         922.553000
max      355913.000000
Name: cogs, dtype: float64


Descriptive statistics for xsga in the year 2015:
count     5815.000000
mean       643.026417
std       2779.072207
min         -0.157000
25%         11.217500
50%         50.629000
75%        247.346000
max      92996.000000
Name: xsga, dtype: float64


Descriptive statistics for oibdp in the year 2015:
count      6777.000000
mean        721.305162
std        3584.085972
min      -21913.000000
25%          -1.289000
50%          24.939000
75%         272.628000
max      106305.000000
Name: oibdp, dtype: float64


Descriptive statistics for dp in the year 2015:
count     7709.000000
mean       220.799353
std       1111.030872
min          0.000000
25%          0.662000
50%          7.686000
75%         78.793000
max      32237.025000
Name: dp, dtype: float64


Descriptive statistics for oiadp in the year 2015:
count      7005.000000
mean        480.593388
std        2979.243251
min      -25913.000000
```

```
25%          -3.208000
50%          16.324000
75%         178.295000
max      106305.000000
Name: oiadp, dtype: float64


Descriptive statistics for xint in the year 2015:
count     7116.000000
mean       174.933584
std       1891.844585
min         -1.339000
25%          0.377750
50%          5.945000
75%         56.515500
max      88033.000000
Name: xint, dtype: float64


Descriptive statistics for nopi in the year 2015:
count     7004.000000
mean       -33.348639
std       1222.706988
min     -63383.983000
25%         -1.104500
50%          0.017000
75%          2.950500
max       9876.000000
Name: nopi, dtype: float64


Descriptive statistics for spi in the year 2015:
count     8040.000000
mean       -51.104014
std        480.816108
min     -14985.420000
25%         -6.300000
50%          0.000000
75%          0.000000
max       9459.559000
Name: spi, dtype: float64


Descriptive statistics for pi in the year 2015:
count     8127.000000
mean       328.749745
std       2113.906445
min     -28226.000000
```

```
25%         -6.005000
50%          8.269000
75%        117.245000
max      72515.000000
Name: pi, dtype: float64


Descriptive statistics for txt in the year 2015:
count     8126.000000
mean        92.846561
std        602.427625
min      -7076.025000
25%          0.000000
50%          1.399000
75%         26.815750
max      19121.000000
Name: txt, dtype: float64


Descriptive statistics for mii in the year 2015:
count    7895.000000
mean       12.058805
std       150.091127
min     -1653.719000
25%         0.000000
50%         0.000000
75%         0.000000
max      5609.000000
Name: mii, dtype: float64


Descriptive statistics for ib in the year 2015:
count     8127.000000
mean       224.215091
std       1551.912756
min     -22348.000000
25%         -6.207000
50%          6.144000
75%         84.312000
max      53394.000000
Name: ib, dtype: float64


Descriptive statistics for dvp in the year 2015:
count     8119.000000
mean         9.060588
std        212.469566
min        -69.544000
```

```
25%          0.000000
50%          0.000000
75%          0.000000
max      11216.000000
Name: dvp, dtype: float64


Descriptive statistics for cstke in the year 2015:
count    8127.000000
mean       -1.063023
std        28.442005
min      -904.035000
25%         0.000000
50%         0.000000
75%         0.000000
max      1222.723000
Name: cstke, dtype: float64


Descriptive statistics for xido in the year 2015:
count     8127.000000
mean         4.027536
std        230.193564
min      -7808.000000
25%          0.000000
50%          0.000000
75%          0.000000
max      10758.000000
Name: xido, dtype: float64


Descriptive statistics for ni in the year 2015:
count     7006.000000
mean       203.579778
std       1540.125880
min     -23119.000000
25%         -9.507500
50%          3.683500
75%         79.499250
max      53394.000000
Name: ni, dtype: float64


Descriptive statistics for ibc in the year 2015:
count     6949.000000
mean       209.715757
std       1551.165409
min     -22757.000000
```

```
25%         -9.648000
50%          3.634000
75%         82.009000
max      53394.000000
Name: ibc, dtype: float64


Descriptive statistics for dpc in the year 2015:
count     6784.000000
mean       244.734037
std       1260.025406
min         -6.178000
25%          0.781500
50%         10.559000
75%         93.010500
max      31318.503000
Name: dpc, dtype: float64


Descriptive statistics for xidoc in the year 2015:
count     6947.000000
mean         4.873820
std        189.573926
min      -1743.956000
25%          0.000000
50%          0.000000
75%          0.000000
max      10387.966000
Name: xidoc, dtype: float64


Descriptive statistics for txdc in the year 2015:
count    6705.000000
mean      -12.456467
std       295.150336
min     -8127.405000
25%        -0.765000
50%         0.000000
75%         0.779000
max      4117.000000
Name: txdc, dtype: float64


Descriptive statistics for esubc in the year 2015:
count    6204.000000
mean       -3.034719
std        88.938173
min     -2573.000000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max      1779.000000
Name: esubc, dtype: float64


Descriptive statistics for sppiv in the year 2015:
count     6616.000000
mean       -26.475047
std        367.648361
min     -15987.613000
25%         -0.270000
50%          0.000000
75%          0.000000
max       6768.831000
Name: sppiv, dtype: float64


Descriptive statistics for fopo in the year 2015:
count     6955.000000
mean       150.035129
std       1115.088105
min     -14626.872000
25%          0.383000
50%          4.785000
75%         33.304500
max      26932.000000
Name: fopo, dtype: float64


Descriptive statistics for fopt in the year 2015:
count     1.000
mean     11.581
std         NaN
min      11.581
25%      11.581
50%      11.581
75%      11.581
max      11.581
Name: fopt, dtype: float64


Descriptive statistics for recch in the year 2015:
count     5965.000000
mean        -7.515307
std        585.877306
min     -22978.885000
```

```
25%         -5.066000
50%         -0.021000
75%          1.703000
max      22664.000000
Name: recch, dtype: float64


Descriptive statistics for invch in the year 2015:
count     6411.000000
mean       -32.578411
std       1370.815159
min     -45093.000000
25%         -0.936000
50%          0.000000
75%          0.090500
max      40742.000000
Name: invch, dtype: float64


Descriptive statistics for apalch in the year 2015:
count     4367.000000
mean         0.885872
std        592.348587
min     -18372.766000
25%         -2.461500
50%          0.089000
75%          3.270000
max      20858.995000
Name: apalch, dtype: float64


Descriptive statistics for txach in the year 2015:
count    4833.000000
mean       -0.990546
std        75.127061
min     -3282.293000
25%         0.000000
50%         0.000000
75%         0.000000
max      1061.000000
Name: txach, dtype: float64


Descriptive statistics for aoloch in the year 2015:
count      6949.000000
mean         55.094548
std        2432.653846
min      -70894.000000
```

```
25%          -4.081000
50%           0.000000
75%           3.622000
max      112047.441000
Name: aoloch, dtype: float64


Descriptive statistics for oancf in the year 2015:
count     6967.000000
mean       584.871533
std       3029.772725
min     -18750.000000
25%         -0.780500
50%         21.886000
75%        219.139500
max      81266.000000
Name: oancf, dtype: float64


Descriptive statistics for ivch in the year 2015:
count    6.691000e+03
mean     1.366596e+03
std      2.014847e+04
min      0.000000e+00
25%      0.000000e+00
50%      0.000000e+00
75%      4.090950e+01
max      1.169666e+06
Name: ivch, dtype: float64


Descriptive statistics for siv in the year 2015:
count      6681.000000
mean       1206.432652
std       18512.534424
min           0.000000
25%           0.000000
50%           0.000000
75%          17.487000
max      906885.808000
Name: siv, dtype: float64


Descriptive statistics for capx in the year 2015:
count     6942.000000
mean       299.235464
std       1499.225298
min         -2.752000
```

```
25%          0.250000
50%          5.691000
75%         76.625500
max      33614.808000
Name: capx, dtype: float64


Descriptive statistics for sppe in the year 2015:
count    5736.000000
mean       20.300423
std       272.154566
min      -219.000000
25%         0.000000
50%         0.000000
75%         0.051000
max      9738.000000
Name: sppe, dtype: float64


Descriptive statistics for aqc in the year 2015:
count     6764.000000
mean       107.212330
std        960.535554
min     -14166.848000
25%          0.000000
50%          0.000000
75%          0.588000
max      37510.100000
Name: aqc, dtype: float64


Descriptive statistics for ivstch in the year 2015:
count      5143.000000
mean        115.583696
std        4036.997039
min     -113483.000000
25%           0.000000
50%           0.000000
75%           0.000000
max      168664.279000
Name: ivstch, dtype: float64


Descriptive statistics for ivaco in the year 2015:
count     6964.00000
mean        18.37548
std       1204.63088
min     -19582.00000
```

```
25%         -0.18600
50%          0.00000
75%          0.95100
max      74761.00000
Name: ivaco, dtype: float64


Descriptive statistics for ivncf in the year 2015:
count      6965.000000
mean       -437.542268
std        4985.522770
min     -107235.000000
25%        -197.423000
50%         -24.605000
75%          -0.198000
max      248324.000000
Name: ivncf, dtype: float64


Descriptive statistics for sstk in the year 2015:
count     6865.000000
mean        68.070453
std        526.862961
min       -143.200000
25%          0.000000
50%          0.539000
75%         12.000000
max      24744.934000
Name: sstk, dtype: float64


Descriptive statistics for prstkc in the year 2015:
count     6625.000000
mean       121.861374
std        805.154516
min          0.000000
25%          0.000000
50%          0.000000
75%          6.700000
max      36752.000000
Name: prstkc, dtype: float64


Descriptive statistics for dv in the year 2015:
count     6868.000000
mean       137.537546
std        648.948358
min          0.000000
```

```
25%          0.000000
50%          0.000000
75%         27.047750
max      12090.000000
Name: dv, dtype: float64


Descriptive statistics for dltis in the year 2015:
count      6824.000000
mean       1139.146583
std       15276.323956
min          -0.242000
25%           0.000000
50%           2.000000
75%         250.000000
max      790985.000000
Name: dltis, dtype: float64


Descriptive statistics for dltr in the year 2015:
count    6.816000e+03
mean     1.043986e+03
std      1.844540e+04
min      0.000000e+00
25%      0.000000e+00
50%      3.941500e+00
75%      1.792802e+02
max      1.029733e+06
Name: dltr, dtype: float64


Descriptive statistics for dlcch in the year 2015:
count     3925.000000
mean       -64.889069
std       2039.139216
min     -97243.000000
25%         -0.031000
50%          0.000000
75%          0.327000
max      34010.000000
Name: dlcch, dtype: float64


Descriptive statistics for fiao in the year 2015:
count     6963.000000
mean        34.458788
std       1911.887361
min     -95036.000000
```

```
25%         -3.874000
50%          0.000000
75%          0.000000
max      78308.000000
Name: fiao, dtype: float64


Descriptive statistics for fincf in the year 2015:
count      6965.000000
mean        -90.127221
std        4992.122238
min     -249000.000000
25%         -32.977000
50%           0.392000
75%          39.641000
max       92003.000000
Name: fincf, dtype: float64


Descriptive statistics for exre in the year 2015:
count     6960.000000
mean        -6.611783
std        261.532036
min     -13827.000000
25%         -0.363000
50%          0.000000
75%          0.000000
max       8878.318000
Name: exre, dtype: float64


Descriptive statistics for chech in the year 2015:
count     6966.000000
mean        50.655483
std       1752.348958
min     -57438.000000
25%         -9.594750
50%          0.000000
75%         14.076750
max      70092.947000
Name: chech, dtype: float64


Descriptive statistics for fsrco in the year 2015:
count    1.0
mean     0.0
std      NaN
min      0.0
```

```
25%      0.0
50%      0.0
75%      0.0
max      0.0
Name: fsrco, dtype: float64


Descriptive statistics for fuseo in the year 2015:
count    1.0
mean     0.0
std      NaN
min      0.0
25%      0.0
50%      0.0
75%      0.0
max      0.0
Name: fuseo, dtype: float64


Descriptive statistics for wcapc in the year 2015:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: wcapc, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 2015:
count    3791.000000
mean             NaN
std              NaN
min             -inf
25%        -0.000127
50%         0.000000
75%         0.033751
max              inf
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 2015:
count    8088.000000
mean        0.245730
std         5.275869
min      -198.776650
```

```
25%         0.033881
50%         0.347797
75%         0.570754
max       147.040541
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 2015:
count    5664.000000
mean            -inf
std              NaN
min             -inf
25%        -0.007502
50%         0.137816
75%         0.379295
max         1.000000
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 2015:
count    7846.000000
mean            -inf
std              NaN
min             -inf
25%        -0.716159
50%         0.004246
75%         0.130089
max         2.419355
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 2015:
count    7002.000000
mean             NaN
std              NaN
min             -inf
25%        -0.117508
50%         0.022691
75%         0.075866
max              inf
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 2015:
count    6301.000000
mean             inf
std              NaN
min         0.000000
```

```
25%         0.461274
50%         1.577143
75%         4.749759
max              inf
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 2015:
count    6971.000000
mean             inf
std              NaN
min       -31.586933
25%         0.087090
50%         0.450482
75%         0.974346
max              inf
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 2015:
count    4.778000e+03
mean              inf
std               NaN
min     -5.466388e+04
25%      1.146397e-01
50%      2.189191e+00
75%      4.613706e+00
max               inf
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 2015:
count    5872.000000
mean        0.630481
std         1.055895
min        -0.785431
25%         0.028418
50%         0.190013
75%         0.837430
max        25.868154
Name: sale_to_at_avg, dtype: float64


Descriptive statistics for dltis_at in the year 2015:
count    6788.000000
mean             inf
std              NaN
min        -0.001060
```

```
25%         0.000000
50%         0.012303
75%         0.132884
max              inf
Name: dltis_at, dtype: float64


Descriptive statistics for dlc_at in the year 2015:
count    8097.000000
mean             inf
std              NaN
min         0.000000
25%         0.000000
50%         0.013899
75%         0.059711
max              inf
Name: dlc_at, dtype: float64


Descriptive statistics for dltt_at in the year 2015:
count    8099.000000
mean             inf
std              NaN
min         0.000000
25%         0.000309
50%         0.092377
75%         0.315268
max              inf
Name: dltt_at, dtype: float64


Descriptive statistics for ppent_at in the year 2015:
count    7847.000000
mean        0.193899
std         0.261681
min         0.000000
25%         0.012628
50%         0.058534
75%         0.279034
max         1.000000
Name: ppent_at, dtype: float64


Descriptive statistics for mkvalt_at in the year 2015:
count    6462.000000
mean             inf
std              NaN
min         0.000007
```

```
25%         0.199402
50%         0.737148
75%         1.795364
max              inf
Name: mkvalt_at, dtype: float64


Descriptive statistics for ni_at in the year 2015:
count    7004.000000
mean             NaN
std              NaN
min             -inf
25%        -0.160070
50%         0.007641
75%         0.044856
max              inf
Name: ni_at, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 2015:
count    3791.000000
mean             NaN
std              NaN
min             -inf
25%        -0.000127
50%         0.000000
75%         0.033751
max              inf
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 2015:
count    8088.000000
mean        0.245730
std         5.275869
min      -198.776650
25%         0.033881
50%         0.347797
75%         0.570754
max       147.040541
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 2015:
count    5664.000000
mean            -inf
std              NaN
min             -inf
```

```
25%        -0.007502
50%         0.137816
75%         0.379295
max         1.000000
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 2015:
count    7846.000000
mean            -inf
std              NaN
min             -inf
25%        -0.716159
50%         0.004246
75%         0.130089
max         2.419355
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 2015:
count    7002.000000
mean             NaN
std              NaN
min             -inf
25%        -0.117508
50%         0.022691
75%         0.075866
max              inf
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 2015:
count    6301.000000
mean             inf
std              NaN
min         0.000000
25%         0.461274
50%         1.577143
75%         4.749759
max              inf
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 2015:
count    6971.000000
mean             inf
std              NaN
min       -31.586933
```

```
25%         0.087090
50%         0.450482
75%         0.974346
max              inf
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 2015:
count    4.778000e+03
mean              inf
std               NaN
min     -5.466388e+04
25%      1.146397e-01
50%      2.189191e+00
75%      4.613706e+00
max               inf
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 2015:
count    5872.000000
mean        0.630481
std         1.055895
min        -0.785431
25%         0.028418
50%         0.190013
75%         0.837430
max        25.868154
Name: sale_to_at_avg, dtype: float64
```

[111]:

	gvkey	datadate	fyear	indfmt	consol	popsrc	datafmt	tic	\
228550	12234	2015-01-31	2014	INDL	C	D	STD	SIGM	
498554	170616	2015-01-31	2014	INDL	C	D	STD	SPLK	
470938	145049	2015-01-31	2014	INDL	C	D	STD	GME	
487805	161036	2015-01-31	2014	INDL	C	D	STD	MFRM.1	
231008	12421	2015-01-31	2014	INDL	C	D	STD	KSPN	
…	…	…	…	…	…	…	…	…	
335515	25705	2015-12-31	2015	INDL	C	D	STD	NFLT	
335552	25709	2015-12-31	2015	INDL	C	D	STD	EQS	
335623	25714	2015-12-31	2015	INDL	C	D	STD	SHG	
336706	25814	2015-12-31	2015	INDL	C	D	STD	GRMY	
539268	345980	2015-12-31	2015	INDL	C	D	STD	WISH	

	cusip	conm	…	net_debt_issued_ratio	\
228550	826565103	SIGMA DESIGNS INC	…	0.000000	

```
498554  848637104                    SPLUNK INC  …              0.000000
470938  36467W109                 GAMESTOP CORP  …                   NaN
487805  57722W106    MATTRESS FIRM HOLDING CORP  …                   NaN
231008  485837108          KASPIEN HOLDINGS INC  …                   NaN
…          …                                …  …                    …
335515  26923G707  NEWFLEET MULTI-SECT UNC BOND  …                   NaN
335552  294766100        EQUUS TOTAL RETURN INC  …              0.000019
335623  824596100   SHINHAN FINANCIAL GROUP LTD  …                   NaN
336706  233051556  XTRACKERS GERMANY EQUITY ETF  …                   NaN
539268  21077C305              CONTEXTLOGIC INC  …                   NaN

       book_leverage     wc_ta     re_ta   ebit_ta      mv_tl  sales_ta  \
228550       0.000000  0.478861 -1.276278 -0.064675   3.620125  0.917110
498554       0.000000  0.523473 -0.310677 -0.172553  14.686256  0.361339
470938       0.146777  0.099569  0.486918  0.149095   1.742598  2.189200
487805       0.638760  0.001790 -0.000466  0.079434   1.723482  1.125287
231008       0.005432  0.619423  0.293144  0.016339   1.001364  1.280280
…                …         …         …         …          …         …
335515            NaN       NaN       NaN       NaN        NaN       NaN
335552       0.286763       NaN -0.322311 -0.044717   1.490373  0.008490
335623       0.671201       NaN  0.050550  0.012781   0.054983  0.036685
336706            NaN       NaN       NaN       NaN        NaN       NaN
539268            NaN       NaN       NaN       NaN        NaN       NaN

         z_score at_rolling_avg sale_to_at_avg
228550  1.654429            NaN            NaN
498554  8.793274       726.5620       0.620560
470938  4.506047      2747.0455       3.383999
487805  2.411750      2927.6620       0.618450
231008  3.075923       944.5165       0.379549
…          …              …              …
335515       NaN            NaN            NaN
335552       NaN            NaN            NaN
335623       NaN    158476.7515       0.073357
336706       NaN            NaN            NaN
539268       NaN            NaN            NaN

[10544 rows x 998 columns]
```

0.12 Descriptive Stats of the Finaincal Ratios for the Year 2020

[112]:
```
describe_yearly_stats(compustat_copy, 2020)
```

```
Descriptive statistics for ch in the year 2020:
count      7858.000000
mean       1811.494341
std       18007.934239
min          -0.012000
```

```
25%          10.012500
50%          58.333000
75%         283.952250
max      574044.380000
Name: ch, dtype: float64


Descriptive statistics for ivst in the year 2020:
count      7828.000000
mean       1878.518624
std       23868.530689
min           0.000000
25%           0.000000
50%           0.812000
75%          59.955500
max      799019.000000
Name: ivst, dtype: float64


Descriptive statistics for rect in the year 2020:
count    6.891000e+03
mean     5.521452e+03
std      6.975249e+04
min      0.000000e+00
25%      2.436000e+00
50%      4.900900e+01
75%      4.458890e+02
max      3.666405e+06
Name: rect, dtype: float64


Descriptive statistics for invt in the year 2020:
count      7161.000000
mean        656.720528
std        9742.781235
min           0.000000
25%           0.000000
50%           1.179000
75%          63.883000
max      503428.000000
Name: invt, dtype: float64


Descriptive statistics for aco in the year 2020:
count     6375.000000
mean       209.792043
std       1885.318012
min          0.000000
```

```
25%          0.511500
50%          5.584000
75%         40.537000
max      87083.800000
Name: aco, dtype: float64


Descriptive statistics for act in the year 2020:
count      5753.000000
mean       2114.981772
std        9263.184861
min           0.000000
25%          25.171000
50%         177.820000
75%         863.193000
max      234982.768000
Name: act, dtype: float64


Descriptive statistics for ppent in the year 2020:
count      7745.000000
mean       2190.015123
std       11025.831803
min           0.000000
25%           3.872000
50%          40.963000
75%         462.000000
max      252880.411000
Name: ppent, dtype: float64


Descriptive statistics for ivaeq in the year 2020:
count     7511.000000
mean       240.689793
std       1878.204663
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max      41327.000000
Name: ivaeq, dtype: float64


Descriptive statistics for ivao in the year 2020:
count    6.701000e+03
mean     3.377855e+03
std      3.521185e+04
min      0.000000e+00
```

```
25%      0.000000e+00
50%      0.000000e+00
75%      7.243300e+01
max      1.253760e+06
Name: ivao, dtype: float64


Descriptive statistics for intan in the year 2020:
count      7870.000000
mean       1566.890905
std        8222.657499
min           0.000000
25%           0.000000
50%          15.000500
75%         341.885750
max      281575.000000
Name: intan, dtype: float64


Descriptive statistics for ao in the year 2020:
count      7965.000000
mean       2542.017747
std       27234.221618
min           0.000000
25%           1.029000
50%          20.825000
75%         159.256000
max      856637.696000
Name: ao, dtype: float64


Descriptive statistics for at in the year 2020:
count    7.967000e+03
mean     2.498819e+04
std      1.748252e+05
min      0.000000e+00
25%      1.214355e+02
50%      9.639440e+02
75%      4.925901e+03
max      3.985749e+06
Name: at, dtype: float64


Descriptive statistics for dlc in the year 2020:
count      7875.000000
mean       1448.399387
std       17006.626650
min           0.000000
```

```
25%           0.579000
50%           9.000000
75%          91.313000
max      614237.411000
Name: dlc, dtype: float64


Descriptive statistics for ap in the year 2020:
count    7.318000e+03
mean     5.321198e+03
std      6.513775e+04
min      0.000000e+00
25%      1.995500e+00
50%      2.000200e+01
75%      2.150352e+02
max      2.144257e+06
Name: ap, dtype: float64


Descriptive statistics for txp in the year 2020:
count     6590.000000
mean        35.978848
std        276.086454
min       -276.000000
25%          0.000000
50%          0.000000
75%          2.616750
max      12022.000000
Name: txp, dtype: float64


Descriptive statistics for lco in the year 2020:
count     6375.000000
mean       699.699363
std       3615.026046
min          0.000000
25%          3.013000
50%         28.943000
75%        206.698000
max      88076.400000
Name: lco, dtype: float64


Descriptive statistics for lct in the year 2020:
count      5759.000000
mean       1669.362982
std        7809.401860
min           0.000000
```

```
25%           8.953000
50%          73.705000
75%         503.311000
max      199382.898000
Name: lct, dtype: float64


Descriptive statistics for dltt in the year 2020:
count    7.946000e+03
mean     5.058885e+03
std      7.601426e+04
min      0.000000e+00
25%      3.004500e+00
50%      8.270250e+01
75%      1.090260e+03
max      3.923563e+06
Name: dltt, dtype: float64


Descriptive statistics for lo in the year 2020:
count    7.967000e+03
mean     4.481505e+03
std      4.364038e+04
min     -1.094000e+00
25%      8.060000e-01
50%      1.783600e+01
75%      1.726530e+02
max      1.101591e+06
Name: lo, dtype: float64


Descriptive statistics for txditc in the year 2020:
count     6120.000000
mean       264.594651
std       1732.957187
min          0.000000
25%          0.000000
50%          0.000000
75%         26.570750
max      73261.000000
Name: txditc, dtype: float64


Descriptive statistics for mib in the year 2020:
count     7955.000000
mean        25.590497
std        953.016102
min         -3.390000
```

```
25%          0.000000
50%          0.000000
75%          0.000000
max      83429.587000
Name: mib, dtype: float64


Descriptive statistics for lt in the year 2020:
count    7.955000e+03
mean     2.142079e+04
std      1.641320e+05
min      0.000000e+00
25%      4.051100e+01
50%      5.513000e+02
75%      3.317206e+03
max      3.960490e+06
Name: lt, dtype: float64


Descriptive statistics for pstk in the year 2020:
count      7933.000000
mean        149.711735
std        2805.002914
min        -183.000000
25%           0.000000
50%           0.000000
75%           0.000000
max      139966.000000
Name: pstk, dtype: float64


Descriptive statistics for ceq in the year 2020:
count      7947.000000
mean       3227.806497
std       15337.220830
min     -114707.000000
25%          29.148500
50%         236.548000
75%        1271.227500
max      443164.000000
Name: ceq, dtype: float64


Descriptive statistics for teq in the year 2020:
count      7967.000000
mean       3573.527337
std       16310.632875
min      -80708.000000
```

```
25%          39.006000
50%         256.887000
75%        1356.028000
max      451336.000000
Name: teq, dtype: float64


Descriptive statistics for sale in the year 2020:
count      6897.000000
mean       3997.053346
std       18026.558260
min       -1876.885000
25%          15.683000
50%         221.917000
75%        1474.000000
max      521426.000000
Name: sale, dtype: float64


Descriptive statistics for cogs in the year 2020:
count      6896.000000
mean       2627.755409
std       13300.843135
min          -3.836000
25%           9.335500
50%         104.473000
75%         848.504500
max      383618.000000
Name: cogs, dtype: float64


Descriptive statistics for xsga in the year 2020:
count      5756.000000
mean        742.716504
std        3603.883114
min          -5.827000
25%          13.483000
50%          63.453000
75%         301.991750
max      129933.000000
Name: xsga, dtype: float64


Descriptive statistics for oibdp in the year 2020:
count      6685.000000
mean        757.979556
std        3741.862441
min       -9022.000000
```

```
25%          -5.683000
50%          17.244000
75%         256.600000
max      100083.000000
Name: oibdp, dtype: float64


Descriptive statistics for dp in the year 2020:
count     7527.000000
mean       281.259007
std       1471.582215
min         -3.233000
25%          0.767500
50%          7.993000
75%         89.100000
max      52892.000000
Name: dp, dtype: float64


Descriptive statistics for oiadp in the year 2020:
count      6896.000000
mean        466.277428
std        2967.712725
min      -28387.000000
25%         -10.805000
50%           6.194000
75%         157.007750
max      100083.000000
Name: oiadp, dtype: float64


Descriptive statistics for xint in the year 2020:
count     6880.000000
mean       195.242760
std       1805.346357
min          0.000000
25%          0.457750
50%          9.383000
75%         68.025000
max      82703.000000
Name: xint, dtype: float64


Descriptive statistics for nopi in the year 2020:
count     6895.000000
mean       -28.168922
std        854.292772
min     -26713.764000
```

```
25%         -1.999500
50%          0.060000
75%          4.168000
max       7211.000000
Name: nopi, dtype: float64


Descriptive statistics for spi in the year 2020:
count     7756.000000
mean       -88.448583
std        739.948811
min     -28042.100000
25%        -14.624750
50%         -0.058000
75%          0.000000
max      11483.184000
Name: spi, dtype: float64


Descriptive statistics for pi in the year 2020:
count     7889.000000
mean       273.492763
std       2368.168731
min     -28883.000000
25%        -23.465000
50%          1.967000
75%         93.520000
max      67091.000000
Name: pi, dtype: float64


Descriptive statistics for txt in the year 2020:
count     7887.000000
mean        64.877208
std        495.196401
min      -7178.074000
25%          0.000000
50%          0.435000
75%         15.876500
max      12440.000000
Name: txt, dtype: float64


Descriptive statistics for mii in the year 2020:
count    7717.000000
mean       11.359197
std       205.626438
min     -5821.000000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max      9201.325000
Name: mii, dtype: float64


Descriptive statistics for ib in the year 2020:
count     7889.000000
mean       197.635555
std       1935.952966
min     -22440.000000
25%        -23.081000
50%          1.298000
75%         75.387000
max      57411.000000
Name: ib, dtype: float64


Descriptive statistics for dvp in the year 2020:
count     7883.000000
mean         9.859838
std        227.701824
min        -84.140000
25%          0.000000
50%          0.000000
75%          0.000000
max      11790.000000
Name: dvp, dtype: float64


Descriptive statistics for cstke in the year 2020:
count    7888.000000
mean       -1.903030
std        31.402502
min     -1150.780000
25%         0.000000
50%         0.000000
75%         0.000000
max       468.311000
Name: cstke, dtype: float64


Descriptive statistics for xido in the year 2020:
count     7889.000000
mean         4.898330
std        201.396329
min      -2172.500000
```

```
25%          0.000000
50%          0.000000
75%          0.000000
max      11839.000000
Name: xido, dtype: float64


Descriptive statistics for ni in the year 2020:
count     6896.000000
mean       171.521504
std       1904.169255
min     -22440.000000
25%        -29.447750
50%         -0.432000
75%         63.384250
max      57411.000000
Name: ni, dtype: float64


Descriptive statistics for ibc in the year 2020:
count     6859.000000
mean       175.981858
std       1933.910560
min     -23251.000000
25%        -29.375500
50%         -0.436000
75%         66.196000
max      57411.000000
Name: ibc, dtype: float64


Descriptive statistics for dpc in the year 2020:
count     6698.000000
mean       288.202779
std       1452.713752
min        -16.436000
25%          0.981000
50%         10.807500
75%        102.822750
max      52444.000000
Name: dpc, dtype: float64


Descriptive statistics for xidoc in the year 2020:
count    6857.000000
mean        0.167686
std        95.408470
min     -2375.000000
```

```
25%         0.000000
50%         0.000000
75%         0.000000
max      4555.000000
Name: xidoc, dtype: float64


Descriptive statistics for txdc in the year 2020:
count    6608.000000
mean      -20.300825
std       274.511720
min     -8856.000000
25%        -1.706500
50%         0.000000
75%         0.000000
max      1880.000000
Name: txdc, dtype: float64


Descriptive statistics for esubc in the year 2020:
count    6097.000000
mean        3.102849
std       144.067972
min     -3545.000000
25%         0.000000
50%         0.000000
75%         0.000000
max      3072.176000
Name: esubc, dtype: float64


Descriptive statistics for sppiv in the year 2020:
count     6456.000000
mean       -19.309669
std        326.774464
min     -11946.044000
25%         -0.134000
50%          0.000000
75%          0.000000
max      10009.170000
Name: sppiv, dtype: float64


Descriptive statistics for fopo in the year 2020:
count     6861.000000
mean       207.185606
std       1587.139693
min     -29642.000000
```

```
25%          1.011000
50%         10.024000
75%         63.001000
max      38668.783000
Name: fopo, dtype: float64


Descriptive statistics for fopt in the year 2020:
count    1.000
mean     3.286
std        NaN
min      3.286
25%      3.286
50%      3.286
75%      3.286
max      3.286
Name: fopt, dtype: float64


Descriptive statistics for recch in the year 2020:
count     5753.000000
mean         2.138056
std        580.475531
min     -18012.000000
25%         -5.516000
50%          0.000000
75%          3.196000
max      15560.328000
Name: recch, dtype: float64


Descriptive statistics for invch in the year 2020:
count     6383.000000
mean       -67.435540
std       1906.594898
min     -97795.000000
25%         -0.621500
50%          0.000000
75%          0.098500
max      15372.929000
Name: invch, dtype: float64


Descriptive statistics for apalch in the year 2020:
count     3921.000000
mean        44.108892
std       1730.517484
min      -9494.000000
```

```
25%         -2.250000
50%          0.167000
75%          6.177000
max      99786.675000
Name: apalch, dtype: float64


Descriptive statistics for txach in the year 2020:
count    4685.00000
mean       -1.34162
std        65.60361
min     -2305.00000
25%         0.00000
50%         0.00000
75%         0.00000
max      1031.00000
Name: txach, dtype: float64


Descriptive statistics for aoloch in the year 2020:
count      6857.000000
mean        151.554052
std        4545.266185
min     -126892.000000
25%          -7.135000
50%          -0.046000
75%           5.150000
max      204692.000000
Name: aoloch, dtype: float64


Descriptive statistics for oancf in the year 2020:
count      6872.000000
mean        749.996682
std        5383.342421
min      -79910.000000
25%          -3.759250
50%          17.139000
75%         230.033750
max      182220.000000
Name: oancf, dtype: float64


Descriptive statistics for ivch in the year 2020:
count      6597.000000
mean       1790.014161
std       23551.690280
min           0.000000
```

```
25%           0.000000
50%           0.000000
75%          28.944000
max      863123.322000
Name: ivch, dtype: float64


Descriptive statistics for siv in the year 2020:
count    6.599000e+03
mean     1.523825e+03
std      2.316771e+04
min      0.000000e+00
25%      0.000000e+00
50%      0.000000e+00
75%      1.427300e+01
max      1.140771e+06
Name: siv, dtype: float64


Descriptive statistics for capx in the year 2020:
count     6853.000000
mean       272.345824
std       1485.227081
min      -3258.000000
25%          0.184000
50%          4.624000
75%         57.696000
max      40140.000000
Name: capx, dtype: float64


Descriptive statistics for sppe in the year 2020:
count     5753.000000
mean        25.594100
std        421.856969
min       -361.000000
25%          0.000000
50%          0.000000
75%          0.012000
max      13399.000000
Name: sppe, dtype: float64


Descriptive statistics for aqc in the year 2020:
count     6587.000000
mean        70.929520
std        794.322812
min     -14748.000000
```

```
25%          0.000000
50%          0.000000
75%          0.000000
max      38260.000000
Name: aqc, dtype: float64


Descriptive statistics for ivstch in the year 2020:
count      4470.000000
mean         -9.640955
std        7002.561443
min     -353665.000000
25%          0.000000
50%          0.000000
75%          0.000000
max      232742.940000
Name: ivstch, dtype: float64


Descriptive statistics for ivaco in the year 2020:
count      6869.00000
mean        -29.61377
std        2610.94474
min     -152519.00000
25%          -0.21700
50%           0.00000
75%           0.27700
max       24135.43200
Name: ivaco, dtype: float64


Descriptive statistics for ivncf in the year 2020:
count      6871.000000
mean       -609.222488
std        5919.867380
min     -261912.000000
25%        -187.690000
50%         -18.373000
75%          -0.049500
max      122554.000000
Name: ivncf, dtype: float64


Descriptive statistics for sstk in the year 2020:
count     6755.000000
mean        93.681214
std        513.470948
min       -424.897000
```

```
25%          0.000000
50%          0.768000
75%         28.683500
max      19840.000000
Name: sstk, dtype: float64


Descriptive statistics for prstkc in the year 2020:
count     6530.000000
mean       118.935299
std       1290.282869
min        -11.311000
25%          0.000000
50%          0.000000
75%          7.889500
max      75992.000000
Name: prstkc, dtype: float64


Descriptive statistics for dv in the year 2020:
count     6787.000000
mean       151.590062
std        755.387554
min          0.000000
25%          0.000000
50%          0.000000
75%         21.241500
max      15137.000000
Name: dv, dtype: float64


Descriptive statistics for dltis in the year 2020:
count    6.758000e+03
mean     2.260562e+03
std      5.716728e+04
min     -4.100000e+00
25%      0.000000e+00
50%      6.206500e+00
75%      3.226590e+02
max      3.133008e+06
Name: dltis, dtype: float64


Descriptive statistics for dltr in the year 2020:
count    6.787000e+03
mean     2.132563e+03
std      5.685849e+04
min      0.000000e+00
```

```
25%      0.000000e+00
50%      6.100000e+00
75%      2.500125e+02
max      3.169886e+06
Name: dltr, dtype: float64


Descriptive statistics for dlcch in the year 2020:
count      3679.000000
mean        -11.016238
std        3085.335421
min      -45513.000000
25%         -0.119000
50%          0.000000
75%          0.099000
max      167482.474000
Name: dlcch, dtype: float64


Descriptive statistics for fiao in the year 2020:
count      6868.000000
mean        407.078019
std        9580.317309
min      -21528.000000
25%         -4.532750
50%          0.000000
75%          0.000000
max      603185.000000
Name: fiao, dtype: float64


Descriptive statistics for fincf in the year 2020:
count      6871.000000
mean        350.400876
std       10087.061694
min      -86820.000000
25%        -25.157500
50%          3.429000
75%        109.350500
max      596645.000000
Name: fincf, dtype: float64


Descriptive statistics for exre in the year 2020:
count     6869.000000
mean         0.886142
std        334.633228
min      -5614.688000
```

```
25%          0.000000
50%          0.000000
75%          0.011000
max      19434.000000
Name: exre, dtype: float64


Descriptive statistics for chech in the year 2020:
count      6872.000000
mean        492.089280
std        6218.906354
min      -40810.127000
25%          -0.277250
50%          10.522000
75%          98.247250
max      263978.000000
Name: chech, dtype: float64


Descriptive statistics for fsrco in the year 2020:
count    1.0
mean     0.0
std      NaN
min      0.0
25%      0.0
50%      0.0
75%      0.0
max      0.0
Name: fsrco, dtype: float64


Descriptive statistics for fuseo in the year 2020:
count    1.0
mean     0.0
std      NaN
min      0.0
25%      0.0
50%      0.0
75%      0.0
max      0.0
Name: fuseo, dtype: float64


Descriptive statistics for wcapc in the year 2020:
count    0.0
mean     NaN
std      NaN
min      NaN
```

```
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: wcapc, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 2020:
count    3590.000000
mean             inf
std              NaN
min        -7.649024
25%        -0.000960
50%         0.000000
75%         0.035453
max              inf
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 2020:
count    7854.000000
mean             inf
std              NaN
min      -339.000000
25%         0.081129
50%         0.343216
75%         0.581120
max              inf
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 2020:
count    5751.000000
mean            -inf
std              NaN
min             -inf
25%         0.002563
50%         0.161755
75%         0.436328
max         1.000000
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 2020:
count    7728.000000
mean            -inf
std              NaN
min             -inf
```

```
25%        -0.793547
50%        -0.019328
75%         0.107703
max         6.113173
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 2020:
count    6893.000000
mean            -inf
std              NaN
min             -inf
25%        -0.146240
50%         0.012926
75%         0.056387
max      2000.000000
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 2020:
count    6064.000000
mean             inf
std              NaN
min         0.000006
25%         0.486351
50%         1.828045
75%         7.050694
max              inf
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 2020:
count    6867.000000
mean             inf
std              NaN
min        -0.505606
25%         0.060883
50%         0.329764
75%         0.750179
max              inf
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 2020:
count    4.671000e+03
mean              inf
std               NaN
min     -3.071689e+05
```

```
25%      4.002442e-01
50%      2.261999e+00
75%      5.295175e+00
max               inf
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 2020:
count    5577.000000
mean        0.517946
std         0.897456
min        -0.049484
25%         0.019584
50%         0.157745
75%         0.677062
max        15.869585
Name: sale_to_at_avg, dtype: float64


Descriptive statistics for dltis_at in the year 2020:
count    6730.000000
mean             inf
std              NaN
min        -0.000721
25%         0.000000
50%         0.027650
75%         0.128618
max              inf
Name: dltis_at, dtype: float64


Descriptive statistics for dlc_at in the year 2020:
count    7861.000000
mean             inf
std              NaN
min         0.000000
25%         0.003594
50%         0.016751
75%         0.053545
max              inf
Name: dlc_at, dtype: float64


Descriptive statistics for dltt_at in the year 2020:
count    7916.000000
mean             inf
std              NaN
min         0.000000
```

```
25%         0.018861
50%         0.119750
75%         0.347429
max              inf
Name: dltt_at, dtype: float64


Descriptive statistics for ppent_at in the year 2020:
count    7713.000000
mean        0.190454
std         0.251398
min         0.000000
25%         0.013451
50%         0.073288
75%         0.265997
max         1.000000
Name: ppent_at, dtype: float64


Descriptive statistics for mkvalt_at in the year 2020:
count    5965.000000
mean             inf
std              NaN
min         0.000005
25%         0.200568
50%         0.834942
75%         2.327728
max              inf
Name: mkvalt_at, dtype: float64


Descriptive statistics for ni_at in the year 2020:
count    6894.000000
mean             NaN
std              NaN
min             -inf
25%        -0.190221
50%        -0.004694
75%         0.032338
max              inf
Name: ni_at, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 2020:
count    3590.000000
mean             inf
std              NaN
min        -7.649024
```

```
25%        -0.000960
50%         0.000000
75%         0.035453
max              inf
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 2020:
count    7854.000000
mean             inf
std              NaN
min      -339.000000
25%         0.081129
50%         0.343216
75%         0.581120
max              inf
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 2020:
count    5751.000000
mean            -inf
std              NaN
min             -inf
25%         0.002563
50%         0.161755
75%         0.436328
max         1.000000
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 2020:
count    7728.000000
mean            -inf
std              NaN
min             -inf
25%        -0.793547
50%        -0.019328
75%         0.107703
max         6.113173
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 2020:
count    6893.000000
mean            -inf
std              NaN
min             -inf
```

```
25%        -0.146240
50%         0.012926
75%         0.056387
max      2000.000000
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 2020:
count    6064.000000
mean             inf
std              NaN
min         0.000006
25%         0.486351
50%         1.828045
75%         7.050694
max              inf
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 2020:
count    6867.000000
mean             inf
std              NaN
min        -0.505606
25%         0.060883
50%         0.329764
75%         0.750179
max              inf
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 2020:
count    4.671000e+03
mean              inf
std               NaN
min     -3.071689e+05
25%      4.002442e-01
50%      2.261999e+00
75%      5.295175e+00
max               inf
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 2020:
count    5577.000000
mean        0.517946
std         0.897456
min        -0.049484
```

```
25%         0.019584
50%         0.157745
75%         0.677062
max        15.869585
Name: sale_to_at_avg, dtype: float64
```

[112]:

	gvkey	datadate	fyear	indfmt	consol	popsrc	datafmt	tic	\
414540	63763	2020-01-31	2019	INDL	C	D	STD	HIBB	
385274	35374	2020-01-31	2019	INDL	C	D	STD	MDLA	
385150	35287	2020-01-31	2019	INDL	C	D	STD	CHWY	
462831	137310	2020-01-31	2019	INDL	C	D	STD	MRVL	
303998	20823	2020-01-31	2019	INDL	C	D	STD	HQY	
…	…	…	…	…	…	…	…	…	
154928	8431	2020-12-31	2020	INDL	C	D	STD	AFG	
381721	33444	2020-12-31	2020	INDL	C	D	STD	EVLO	
381728	33445	2020-12-31	2020	INDL	C	D	STD	SRRK	
381741	33447	2020-12-31	2020	INDL	C	D	STD	KNSA	
539310	351590	2020-12-31	2020	INDL	C	D	STD	DTRUY	

	cusip	conm	…	net_debt_issued_ratio	\
414540	428567101	HIBBETT INC	…	NaN	
385274	584021109	MEDALLIA INC	…	NaN	
385150	16679L109	CHEWY INC	…	NaN	
462831	573874104	MARVELL TECHNOLOGY INC	…	-0.026946	
303998	42226A107	HEALTHEQUITY INC	…	NaN	
…	…	…	…	…	
154928	025932104	AMERICAN FINANCIAL GROUP INC	…	NaN	
381721	299734202	EVELO BIOSCIENCES INC	…	NaN	
381728	80706P103	SCHOLAR ROCK HOLDNG CRP	…	NaN	
381741	G5269C101	KINIKSA PHRMCETCLS LTD	…	0.000000	
539310	23384L101	DAIMLER TRUCK HOLDING AG	…	-0.010342	

	book_leverage	wc_ta	re_ta	ebit_ta	mv_tl	sales_ta	\
414540	0.435630	0.180382	1.019724	0.070808	0.947853	1.538448	
385274	0.021446	0.294070	-0.657664	-0.163525	11.186972	0.550077	
385150	-1.148295	-0.504922	-1.978366	-0.269575	7.962513	5.198578	
462831	0.154303	0.074312	0.228264	-0.006143	6.497892	0.242442	
303998	0.558077	0.056085	0.082456	0.042538	3.057966	0.207390	
…	…	…	…	…	…	…	
154928	0.323939	NaN	0.073703	0.013267	0.113296	0.106177	
381721	0.577747	0.564294	-3.227297	-1.013758	9.540717	0.000000	
381728	0.179734	0.777327	-0.629230	-0.223736	12.995621	0.039667	
381741	0.021902	0.862472	-1.480859	-0.450298	32.118070	0.000000	
539310	0.714152	0.081138	NaN	0.002561	NaN	0.720418	

```
          z_score at_rolling_avg sale_to_at_avg
414540   3.969514            NaN            NaN
385274   6.149281       750.7040       0.536114
385150   5.658885       831.9850       5.825517
462831   4.527224      6032.7780       0.447416
303998   2.363210      6849.2080       0.077672
…             …             …              …
154928        NaN     36923.5945       0.211545
381721  -1.462035     36828.3195       0.000000
381728   7.150184       239.4720       0.064321
381741  16.746622       368.8845       0.000000
539310        NaN     30302.7525       1.432529

[10886 rows x 998 columns]
```

0.13 Descriptive Stats of the Finaincal Ratios for the Year 2021

[113]:
```
describe_yearly_stats(compustat_copy, 2021)
```

```
Descriptive statistics for ch in the year 2021:
count      7774.000000
mean       2053.760827
std       21464.686362
min           0.000000
25%          14.067500
50%          74.345000
75%         313.987750
max      656125.678000
Name: ch, dtype: float64


Descriptive statistics for ivst in the year 2021:
count      7747.000000
mean       1966.096677
std       24854.782040
min           0.000000
25%           0.000000
50%           1.183000
75%          77.060500
max      976094.000000
Name: ivst, dtype: float64


Descriptive statistics for rect in the year 2021:
count    6.834000e+03
mean     5.833952e+03
std      7.519171e+04
min     -7.000000e-03
```

```
25%      3.326000e+00
50%      5.794450e+01
75%      5.004555e+02
max      3.977869e+06
Name: rect, dtype: float64


Descriptive statistics for invt in the year 2021:
count      7096.000000
mean        687.787350
std        9592.565958
min           0.000000
25%           0.000000
50%           1.468000
75%          75.682250
max      481195.000000
Name: invt, dtype: float64


Descriptive statistics for aco in the year 2021:
count      6331.000000
mean        278.074552
std        3283.732790
min           0.000000
25%           0.920500
50%           7.441000
75%          48.585000
max      158372.600000
Name: aco, dtype: float64


Descriptive statistics for act in the year 2021:
count      5737.000000
mean       2429.746168
std       10702.184141
min           0.000000
25%          40.902000
50%         234.652000
75%        1006.000000
max      227859.796000
Name: act, dtype: float64


Descriptive statistics for ppent in the year 2021:
count      7672.000000
mean       2225.804626
std       11104.468585
min           0.000000
```

```
25%           4.772000
50%          46.494500
75%         457.210750
max      262562.034000
Name: ppent, dtype: float64


Descriptive statistics for ivaeq in the year 2021:
count     7440.000000
mean       253.625373
std       2011.289861
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max      46100.000000
Name: ivaeq, dtype: float64


Descriptive statistics for ivao in the year 2021:
count    6.633000e+03
mean     3.531939e+03
std      3.720526e+04
min      0.000000e+00
25%      0.000000e+00
50%      0.000000e+00
75%      9.212200e+01
max      1.311878e+06
Name: ivao, dtype: float64


Descriptive statistics for intan in the year 2021:
count      7780.000000
mean       1698.809977
std        8621.109357
min           0.000000
25%           0.000000
50%          19.069000
75%         401.175250
max      292716.000000
Name: intan, dtype: float64


Descriptive statistics for ao in the year 2021:
count      7886.000000
mean       2500.279594
std       26388.408292
min           0.000000
```

```
25%           1.148250
50%          22.307000
75%         162.565000
max      920244.360000
Name: ao, dtype: float64


Descriptive statistics for at in the year 2021:
count    7.886000e+03
mean     2.640748e+04
std      1.837091e+05
min      0.000000e+00
25%      1.533367e+02
50%      1.086574e+03
75%      5.581340e+03
max      4.229166e+06
Name: at, dtype: float64


Descriptive statistics for dlc in the year 2021:
count      7801.000000
mean       1411.753269
std       15818.897600
min           0.000000
25%           0.543000
50%           7.938000
75%          87.068000
max      511853.440000
Name: dlc, dtype: float64


Descriptive statistics for ap in the year 2021:
count    7.255000e+03
mean     5.794103e+03
std      7.060685e+04
min      0.000000e+00
25%      2.649000e+00
50%      2.375000e+01
75%      2.591500e+02
max      2.462303e+06
Name: ap, dtype: float64


Descriptive statistics for txp in the year 2021:
count     6533.000000
mean        43.753262
std        306.880163
min       -482.000000
```

```
25%          0.000000
50%          0.000000
75%          3.075000
max      13119.000000
Name: txp, dtype: float64


Descriptive statistics for lco in the year 2021:
count      6331.000000
mean        838.116182
std        4834.597470
min           0.000000
25%           4.022000
50%          35.544000
75%         251.393000
max      159545.000000
Name: lco, dtype: float64


Descriptive statistics for lct in the year 2021:
count      5741.000000
mean       1923.297325
std        9318.053416
min           0.000000
25%          11.595000
50%          85.481000
75%         567.100000
max      194033.000000
Name: lct, dtype: float64


Descriptive statistics for dltt in the year 2021:
count    7.865000e+03
mean     5.336780e+03
std      8.291019e+04
min      0.000000e+00
25%      3.479000e+00
50%      8.560000e+01
75%      1.154429e+03
max      4.155396e+06
Name: dltt, dtype: float64


Descriptive statistics for lo in the year 2021:
count    7.886000e+03
mean     4.294791e+03
std      4.191138e+04
min      0.000000e+00
```

```
25%      8.082500e-01
50%      1.777100e+01
75%      1.738980e+02
max      1.130167e+06
Name: lo, dtype: float64


Descriptive statistics for txditc in the year 2021:
count     6062.000000
mean       289.800543
std       1951.715099
min          0.000000
25%          0.000000
50%          0.000000
75%         32.404250
max      89679.000000
Name: txditc, dtype: float64


Descriptive statistics for mib in the year 2021:
count     7874.000000
mean        25.748072
std        752.810945
min         -3.522000
25%          0.000000
50%          0.000000
75%          0.000000
max      64470.446000
Name: mib, dtype: float64


Descriptive statistics for lt in the year 2021:
count    7.872000e+03
mean     2.253330e+04
std      1.725443e+05
min      0.000000e+00
25%      4.543850e+01
50%      6.071235e+02
75%      3.671871e+03
max      4.181809e+06
Name: lt, dtype: float64


Descriptive statistics for pstk in the year 2021:
count      7852.000000
mean        141.027618
std        2808.104121
min        -136.000000
```

```
25%           0.000000
50%           0.000000
75%           0.000000
max      139966.000000
Name: pstk, dtype: float64


Descriptive statistics for ceq in the year 2021:
count      7864.000000
mean       3547.620783
std       16318.390623
min      -92609.000000
25%         48.851750
50%        291.083500
75%       1487.350000
max      506199.000000
Name: ceq, dtype: float64


Descriptive statistics for teq in the year 2021:
count      7886.000000
mean       3888.010042
std       17357.052922
min      -75680.000000
25%         54.017250
50%        308.355000
75%       1571.529000
max      514930.000000
Name: teq, dtype: float64


Descriptive statistics for sale in the year 2021:
count      6896.000000
mean       4562.356387
std       20839.949179
min         -69.296000
25%          18.268750
50%         244.254500
75%        1698.155000
max      556933.000000
Name: sale, dtype: float64


Descriptive statistics for cogs in the year 2021:
count      6896.000000
mean       2909.670914
std       15132.893364
min       -3703.000000
```

```
25%           6.986000
50%         102.358000
75%         934.834500
max      409163.000000
Name: cogs, dtype: float64


Descriptive statistics for xsga in the year 2021:
count      5772.000000
mean        804.478663
std        4062.568881
min          -1.247000
25%          17.013250
50%          75.978500
75%         343.295250
max      172537.000000
Name: xsga, dtype: float64


Descriptive statistics for oibdp in the year 2021:
count      6690.000000
mean        988.669961
std        4824.681740
min       -7260.687000
25%          -8.087250
50%          22.960000
75%         337.093500
max      125581.000000
Name: oibdp, dtype: float64


Descriptive statistics for dp in the year 2021:
count     7508.000000
mean       275.547374
std       1378.585261
min         -3.967000
25%          0.819250
50%          8.484000
75%         91.974500
max      37145.738000
Name: dp, dtype: float64


Descriptive statistics for oiadp in the year 2021:
count      6896.000000
mean        702.699848
std        3911.916781
min      -11047.982000
```

```
25%         -13.319500
50%          11.311000
75%         217.324750
max      114863.000000
Name: oiadp, dtype: float64


Descriptive statistics for xint in the year 2021:
count     6938.000000
mean       163.527991
std       1517.444907
min      -1042.000000
25%          0.384500
50%          6.918000
75%         60.000000
max      70088.000000
Name: xint, dtype: float64


Descriptive statistics for nopi in the year 2021:
count     6895.000000
mean        28.300219
std        996.417406
min     -27448.219000
25%         -1.080500
50%          0.083000
75%          7.339500
max      19828.135000
Name: nopi, dtype: float64


Descriptive statistics for spi in the year 2021:
count     7766.000000
mean       -31.670196
std        422.738319
min      -8317.000000
25%        -10.000000
50%          0.000000
75%          0.000000
max      13835.000000
Name: spi, dtype: float64


Descriptive statistics for pi in the year 2021:
count      7875.000000
mean        660.530495
std        3732.009808
min       -9522.000000
```

```
25%         -14.823500
50%          12.061000
75%         191.184000
max      111686.000000
Name: pi, dtype: float64


Descriptive statistics for txt in the year 2021:
count     7873.000000
mean       136.367513
std        765.960910
min      -2453.200000
25%          0.000000
50%          1.302000
75%         29.446000
max      23007.000000
Name: txt, dtype: float64


Descriptive statistics for mii in the year 2021:
count    7705.000000
mean       24.569040
std       271.448504
min     -1426.095000
25%         0.000000
50%         0.000000
75%         0.000000
max      8422.000000
Name: mii, dtype: float64


Descriptive statistics for ib in the year 2021:
count     7875.000000
mean       500.389482
std       2995.179312
min      -9501.000000
25%        -15.136000
50%          9.775000
75%        153.945000
max      94680.000000
Name: ib, dtype: float64


Descriptive statistics for dvp in the year 2021:
count     7868.000000
mean        14.226397
std        402.566125
min       -132.947000
```

```
25%          0.000000
50%          0.000000
75%          0.000000
max      22098.000000
Name: dvp, dtype: float64


Descriptive statistics for cstke in the year 2021:
count    7875.000000
mean       -2.408366
std        31.817296
min     -1088.696000
25%         0.000000
50%         0.000000
75%         0.000000
max       129.336000
Name: cstke, dtype: float64


Descriptive statistics for xido in the year 2021:
count     7875.000000
mean         9.699124
std        289.413014
min      -4234.000000
25%          0.000000
50%          0.000000
75%          0.000000
max      14001.617000
Name: xido, dtype: float64


Descriptive statistics for ni in the year 2021:
count     6896.000000
mean       457.923037
std       2839.277426
min      -9501.000000
25%        -22.171750
50%          4.097000
75%        137.004250
max      94680.000000
Name: ni, dtype: float64


Descriptive statistics for ibc in the year 2021:
count     6859.000000
mean       470.507920
std       2874.581158
min      -9501.000000
```

```
25%        -22.973000
50%          3.893000
75%        139.637500
max      94680.000000
Name: ibc, dtype: float64


Descriptive statistics for dpc in the year 2021:
count     6700.000000
mean       284.183569
std       1363.475835
min        -15.945000
25%          1.084750
50%         11.603500
75%        104.831000
max      36406.491000
Name: dpc, dtype: float64


Descriptive statistics for xidoc in the year 2021:
count     6858.000000
mean         4.976865
std        220.422316
min      -4024.000000
25%          0.000000
50%          0.000000
75%          0.000000
max      14744.291000
Name: xidoc, dtype: float64


Descriptive statistics for txdc in the year 2021:
count    6606.000000
mean        1.407574
std       210.583260
min     -4774.000000
25%        -0.758250
50%         0.000000
75%         0.336750
max      5246.000000
Name: txdc, dtype: float64


Descriptive statistics for esubc in the year 2021:
count     6110.000000
mean       -15.621827
std        304.850065
min     -15339.000000
```

```
25%          0.000000
50%          0.000000
75%          0.000000
max       6098.000000
Name: esubc, dtype: float64


Descriptive statistics for sppiv in the year 2021:
count     6427.000000
mean       -33.426547
std        373.502898
min     -14124.335000
25%         -0.343000
50%          0.000000
75%          0.000000
max       8401.000000
Name: sppiv, dtype: float64


Descriptive statistics for fopo in the year 2021:
count     6861.000000
mean        87.177401
std       1560.375205
min     -80973.000000
25%          0.058000
50%          7.333000
75%         46.890000
max      32884.000000
Name: fopo, dtype: float64


Descriptive statistics for fopt in the year 2021:
count    1.000
mean     3.977
std        NaN
min      3.977
25%      3.977
50%      3.977
75%      3.977
max      3.977
Name: fopt, dtype: float64


Descriptive statistics for recch in the year 2021:
count     5746.000000
mean       -98.710779
std        887.388963
min     -31386.778000
```

```
25%        -24.812000
50%         -1.416500
75%          0.033000
max       7088.951000
Name: recch, dtype: float64


Descriptive statistics for invch in the year 2021:
count     6384.000000
mean       -87.255252
std       1387.820652
min     -57630.000000
25%         -5.724750
50%          0.000000
75%          0.000000
max      39250.000000
Name: invch, dtype: float64


Descriptive statistics for apalch in the year 2021:
count     3889.000000
mean       108.890530
std        805.285431
min      -8442.000000
25%         -0.011000
50%          2.183000
75%         26.035000
max      21470.000000
Name: apalch, dtype: float64


Descriptive statistics for txach in the year 2021:
count    4634.000000
mean        0.458339
std        56.008023
min     -1485.000000
25%         0.000000
50%         0.000000
75%         0.000000
max      1223.000000
Name: txach, dtype: float64


Descriptive statistics for aoloch in the year 2021:
count      6858.000000
mean        166.768194
std        3610.115411
min      -39920.000000
```

```
25%          -6.631500
50%          -0.017000
75%           7.218500
max      183746.826000
Name: aoloch, dtype: float64


Descriptive statistics for oancf in the year 2021:
count      6870.000000
mean        862.212228
std        5238.190063
min      -16995.058000
25%          -8.037500
50%          15.500500
75%         249.933500
max      169942.087000
Name: oancf, dtype: float64


Descriptive statistics for ivch in the year 2021:
count    6.601000e+03
mean     1.926995e+03
std      2.583360e+04
min      0.000000e+00
25%      0.000000e+00
50%      0.000000e+00
75%      3.305800e+01
max      1.060805e+06
Name: ivch, dtype: float64


Descriptive statistics for siv in the year 2021:
count    6.609000e+03
mean     1.576644e+03
std      2.453599e+04
min      0.000000e+00
25%      0.000000e+00
50%      0.000000e+00
75%      1.680100e+01
max      1.122452e+06
Name: siv, dtype: float64


Descriptive statistics for capx in the year 2021:
count     6854.000000
mean       290.833091
std       1663.153316
min         -0.540000
```

```
25%          0.269000
50%          5.709000
75%         63.057000
max      61053.000000
Name: capx, dtype: float64


Descriptive statistics for sppe in the year 2021:
count     5740.000000
mean        26.537982
std        439.438336
min       -442.000000
25%          0.000000
50%          0.000000
75%          0.017000
max      14393.000000
Name: sppe, dtype: float64


Descriptive statistics for aqc in the year 2021:
count     6538.000000
mean       107.635147
std        862.004523
min     -25606.000000
25%          0.000000
50%          0.000000
75%          0.797750
max      25453.000000
Name: aqc, dtype: float64


Descriptive statistics for ivstch in the year 2021:
count      4366.000000
mean        146.231264
std        9211.129757
min     -385559.000000
25%           0.000000
50%           0.000000
75%           0.000000
max      396481.321000
Name: ivstch, dtype: float64


Descriptive statistics for ivaco in the year 2021:
count      6869.000000
mean        -19.046020
std        3586.623068
min     -263779.000000
```

```
25%          -0.599000
50%           0.000000
75%           0.191000
max       87739.196000
Name: ivaco, dtype: float64


Descriptive statistics for ivncf in the year 2021:
count      6869.000000
mean       -631.414834
std        5722.172102
min     -313291.000000
25%        -233.463000
50%         -25.071000
75%          -0.150000
max       90850.000000
Name: ivncf, dtype: float64


Descriptive statistics for sstk in the year 2021:
count     6756.000000
mean       100.849026
std        497.744990
min       -501.283000
25%          0.000000
50%          2.634500
75%         42.299250
max      22880.604000
Name: sstk, dtype: float64


Descriptive statistics for prstkc in the year 2021:
count     6541.000000
mean       181.293202
std       1829.197116
min         -1.380000
25%          0.000000
50%          0.000000
75%         10.176000
max      92527.000000
Name: prstkc, dtype: float64


Descriptive statistics for dv in the year 2021:
count     6787.000000
mean       165.382370
std        824.613128
min         -0.603000
```

```
25%          0.000000
50%          0.000000
75%         21.469500
max      16521.000000
Name: dv, dtype: float64


Descriptive statistics for dltis in the year 2021:
count    6.757000e+03
mean     1.552115e+03
std      3.478280e+04
min     -1.940000e+01
25%      0.000000e+00
50%      5.920000e-01
75%      3.000000e+02
max      2.294272e+06
Name: dltis, dtype: float64


Descriptive statistics for dltr in the year 2021:
count    6.795000e+03
mean     1.519529e+03
std      3.578419e+04
min      0.000000e+00
25%      0.000000e+00
50%      5.090000e+00
75%      2.618655e+02
max      2.301335e+06
Name: dltr, dtype: float64


Descriptive statistics for dlcch in the year 2021:
count     3687.000000
mean       -41.485767
std       1761.552298
min     -85553.677000
25%         -0.022000
50%          0.000000
75%          0.053500
max      26438.000000
Name: dlcch, dtype: float64


Descriptive statistics for fiao in the year 2021:
count      6868.000000
mean        262.528367
std        6098.134770
min      -39785.982000
```

```
25%          -6.681500
50%           0.000000
75%           0.000000
max      288033.000000
Name: fiao, dtype: float64


Descriptive statistics for fincf in the year 2021:
count      6869.000000
mean         29.418914
std        6843.960257
min     -145053.000000
25%         -38.622000
50%           3.979000
75%         112.585000
max      291650.000000
Name: fincf, dtype: float64


Descriptive statistics for exre in the year 2021:
count     6867.000000
mean        -6.624261
std        298.278793
min     -15345.000000
25%         -0.032500
50%          0.000000
75%          0.000000
max       5909.544000
Name: exre, dtype: float64


Descriptive statistics for chech in the year 2021:
count      6870.000000
mean        253.681955
std        5136.568810
min      -47582.000000
25%         -13.836500
50%           2.522000
75%          61.627000
max      223984.897000
Name: chech, dtype: float64


Descriptive statistics for fsrco in the year 2021:
count    1.0
mean     0.0
std      NaN
min      0.0
```

```
25%      0.0
50%      0.0
75%      0.0
max      0.0
Name: fsrco, dtype: float64


Descriptive statistics for fuseo in the year 2021:
count    1.0
mean     0.0
std      NaN
min      0.0
25%      0.0
50%      0.0
75%      0.0
max      0.0
Name: fuseo, dtype: float64


Descriptive statistics for wcapc in the year 2021:
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
max      NaN
Name: wcapc, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 2021:
count    3613.000000
mean             NaN
std              NaN
min             -inf
25%       -0.004278
50%        0.000000
75%        0.019448
max              inf
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 2021:
count    7778.000000
mean             inf
std              NaN
min      -916.750000
```

```
25%         0.068141
50%         0.304388
75%         0.548364
max              inf
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 2021:
count    5731.000000
mean            -inf
std              NaN
min             -inf
25%         0.025393
50%         0.191496
75%         0.500832
max         1.000000
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 2021:
count    7674.000000
mean            -inf
std              NaN
min             -inf
25%        -0.755619
50%        -0.016408
75%         0.108141
max         7.025394
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 2021:
count    6895.000000
mean            -inf
std              NaN
min             -inf
25%        -0.149588
50%         0.017685
75%         0.070377
max         4.120338
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 2021:
count    6403.000000
mean             inf
std              NaN
min         0.000000
```

```
25%         0.692348
50%         2.314900
75%         7.927740
max              inf
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 2021:
count    6871.000000
mean             inf
std              NaN
min        -1.192145
25%         0.058855
50%         0.342959
75%         0.749769
max              inf
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 2021:
count    5.087000e+03
mean              inf
std               NaN
min     -1.791099e+05
25%      6.728686e-01
50%      2.546370e+00
75%      5.701925e+00
max               inf
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 2021:
count    5472.000000
mean        0.505744
std         0.832083
min        -0.006726
25%         0.022761
50%         0.159615
75%         0.671625
max        14.671727
Name: sale_to_at_avg, dtype: float64


Descriptive statistics for dltis_at in the year 2021:
count    6731.000000
mean        0.129755
std         1.096160
min        -0.010821
```

```
25%         0.000000
50%         0.002987
75%         0.117236
max        79.000000
Name: dltis_at, dtype: float64


Descriptive statistics for dlc_at in the year 2021:
count    7784.000000
mean             inf
std              NaN
min         0.000000
25%         0.002684
50%         0.012668
75%         0.043262
max              inf
Name: dlc_at, dtype: float64


Descriptive statistics for dltt_at in the year 2021:
count    7840.000000
mean             inf
std              NaN
min         0.000000
25%         0.016247
50%         0.109335
75%         0.329845
max              inf
Name: dltt_at, dtype: float64


Descriptive statistics for ppent_at in the year 2021:
count    7646.000000
mean        0.179060
std         0.240791
min         0.000000
25%         0.012909
50%         0.067401
75%         0.242585
max         0.999485
Name: ppent_at, dtype: float64


Descriptive statistics for mkvalt_at in the year 2021:
count    6321.000000
mean             inf
std              NaN
min         0.000005
```

```
25%         0.309248
50%         0.990448
75%         2.350598
max              inf
Name: mkvalt_at, dtype: float64


Descriptive statistics for ni_at in the year 2021:
count    6896.000000
mean            -inf
std              NaN
min             -inf
25%        -0.167234
50%         0.008389
75%         0.051822
max        20.200000
Name: ni_at, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the year 2021:
count    3613.000000
mean             NaN
std              NaN
min             -inf
25%        -0.004278
50%         0.000000
75%         0.019448
max              inf
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the year 2021:
count    7778.000000
mean             inf
std              NaN
min      -916.750000
25%         0.068141
50%         0.304388
75%         0.548364
max              inf
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the year 2021:
count    5731.000000
mean            -inf
std              NaN
min             -inf
```

```
25%         0.025393
50%         0.191496
75%         0.500832
max         1.000000
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the year 2021:
count    7674.000000
mean            -inf
std              NaN
min             -inf
25%        -0.755619
50%        -0.016408
75%         0.108141
max         7.025394
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the year 2021:
count    6895.000000
mean            -inf
std              NaN
min             -inf
25%        -0.149588
50%         0.017685
75%         0.070377
max         4.120338
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the year 2021:
count    6403.000000
mean             inf
std              NaN
min         0.000000
25%         0.692348
50%         2.314900
75%         7.927740
max              inf
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the year 2021:
count    6871.000000
mean             inf
std              NaN
min        -1.192145
```

```
25%         0.058855
50%         0.342959
75%         0.749769
max              inf
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the year 2021:
count    5.087000e+03
mean              inf
std               NaN
min     -1.791099e+05
25%      6.728686e-01
50%      2.546370e+00
75%      5.701925e+00
max               inf
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the year 2021:
count    5472.000000
mean        0.505744
std         0.832083
min        -0.006726
25%         0.022761
50%         0.159615
75%         0.671625
max        14.671727
Name: sale_to_at_avg, dtype: float64
```

[113]:

	gvkey	datadate	fyear	indfmt	consol	popsrc	datafmt	tic	\
413444	63643	2021-01-31	2020	INDL	C	D	STD	ANF	
357289	29150	2021-01-31	2020	INDL	C	D	STD	URBN	
473831	147242	2021-01-31	2020	INDL	C	D	STD	VRNT	
499763	171141	2021-01-31	2020	INDL	C	D	STD	FIVE	
392251	39942	2021-01-31	2020	INDL	C	D	STD	BRZE	
…	…	…	…	…	…	…	…	…	
380725	32888	2021-12-31	2021	INDL	C	D	STD	LSST	
380735	32890	2021-12-31	2021	INDL	C	D	STD	DTEC	
380740	32891	2021-12-31	2021	INDL	C	D	STD	DWCR	
380689	32881	2021-12-31	2021	INDL	C	D	STD	SIMS	
539316	353945	2021-12-31	2021	INDL	C	D	STD	ACLLY	

	cusip	conm	…	net_debt_issued_ratio	\
413444	002896207	ABERCROMBIE & FITCH -CL A	…	0.03522	

```
357289  917047102          URBAN OUTFITTERS INC  …                0.00000
473831  92343X100            VERINT SYSTEMS INC  …                    NaN
499763  33829M101                FIVE BELOW INC  …                0.00000
392251  10576N102                     BRAZE INC  …                0.00000
…          …                                …  …                      …
380725  63873X208  NTXS LMS SYLS SH DUR INC ETF  …                    NaN
380735  00162Q478   ALPS DISRUPTIVE TECHNLG ETF  …                    NaN
380740  042765685  ARROW DWA COUNTRY ROTATN ETF  …                    NaN
380689  78468R697  SPDR S&P KENSHO INTLGNT STRC  …                    NaN
539316  00449R109      ACCELLERON INDUSTRIES AG  …                    NaN

       book_leverage     wc_ta     re_ta   ebit_ta     mv_tl  sales_ta  \
413444       0.623386  0.211840  0.617564  0.004047  0.608535  0.942828
357289       0.473513  0.089444  0.411124  0.010340  1.296815  0.972762
473831       0.378121  0.035048 -0.076864  0.052368  2.738468  0.390552
499763       0.557334  0.138132  0.242034  0.066876  6.859912  0.847660
392251       0.000000  0.191524 -0.806819 -0.158419       NaN  0.876291
…                 …         …         …         …         …         …
380725            NaN       NaN       NaN       NaN       NaN       NaN
380735            NaN       NaN       NaN       NaN       NaN       NaN
380740            NaN       NaN       NaN       NaN       NaN       NaN
380689            NaN       NaN       NaN       NaN       NaN       NaN
539316       0.298961  0.161409       NaN  0.266705       NaN  1.102583

         z_score at_rolling_avg sale_to_at_avg
413444  2.430673            NaN            NaN
357289  2.458151      3430.6235       1.005575
473831  2.136992      3403.8200       0.374199
499763  5.680426      2788.0325       0.703771
392251       NaN      1243.0820       0.120821
…            …              …              …
380725       NaN            NaN            NaN
380735       NaN            NaN            NaN
380740       NaN            NaN            NaN
380689       NaN            NaN            NaN
539316       NaN            NaN            NaN

[11005 rows x 998 columns]
```

0.14 Descriptive Stats of the Finaincal Ratios for the Year Entire Time Period

[114]:
```
def describe_yearly_stats_full_period(df_year):

    # Columns for which to describe statistics
    columns_to_describe = [
        'ch', 'ivst', 'rect', 'invt', 'aco', 'act', 'ppent', 'ivaeq', 'ivao',␣
↪'intan', 'ao', 'at',
```

```
      'dlc', 'ap', 'txp', 'lco', 'lct', 'dltt', 'lo', 'txditc', 'mib', 'lt',␣
↪'pstk', 'ceq', 'teq',
      'sale', 'cogs', 'xsga', 'oibdp', 'dp', 'oiadp', 'xint', 'nopi', 'spi',␣
↪'pi', 'txt', 'mii', 'ib', 'dvp',
      'cstke', 'xido', 'ni', 'ibc', 'dpc', 'xidoc', 'txdc', 'esubc', 'sppiv',␣
↪'fopo', 'fopt', 'recch',
      'invch', 'apalch', 'txach', 'aoloch', 'oancf', 'ivch', 'siv', 'capx',␣
↪'sppe', 'aqc', 'ivstch', 'ivaco',
      'ivncf', 'sstk', 'prstkc', 'dv', 'dltis', 'dltr', 'dlcch', 'fiao',␣
↪'fincf', 'exre', 'chech', 'fsrco',
      'fuseo', 'wcapc', 'net_debt_issued_ratio', 'book_leverage', 'wc_ta',␣
↪'re_ta', 'ebit_ta', 'mv_tl',
      'sales_ta', 'z_score', 'sale_to_at_avg'
  ]

  # Compute additional ratios for describe
  df_year['dltis_at'] = df_year['dltis'] / df_year['at']
  df_year['dlc_at'] = df_year['dlc'] / df_year['at']
  df_year['dltt_at'] = df_year['dltt'] / df_year['at']
  df_year['ppent_at'] = df_year['ppent'] / df_year['at']
  df_year['mkvalt_at'] = df_year.get('mkvalt', pd.Series(index=df_year.
↪index)) / df_year['at']  # 'mkvalt' might not be present
  df_year['ni_at'] = df_year['ni'] / df_year['at']
  df_year['net_debt_issued'] = df_year['dltis'] - df_year['dltr'] +␣
↪df_year['dlcch']
  df_year['net_debt_issued_ratio'] = df_year['net_debt_issued'] /␣
↪df_year['at']
  df_year['book_leverage'] = (df_year['dltt'] + df_year['dlc']) /␣
↪(df_year['dltt'] + df_year['dlc'] + df_year['seq'])
  df_year['wc_ta'] = (df_year['act'] - df_year['lct']) / df_year['at']
  df_year['re_ta'] = df_year['re'] / df_year['at']
  df_year['ebit_ta'] = df_year['oiadp'] / df_year['at']
  df_year['mv_tl'] = (df_year['prcc_f'] * df_year['csho']) / df_year['lt']
  df_year['sales_ta'] = df_year['sale'] / df_year['at']

  # Calculate the Altman Z-Score
  df_year['z_score'] = (1.2 * df_year['wc_ta']) + \
                       (1.4 * df_year['re_ta']) + \
                       (3.3 * df_year['ebit_ta']) + \
                       (0.6 * df_year['mv_tl']) + \
                       (0.99 * df_year['sales_ta'])

  # Calculate rolling average for 'at' and 'sale_to_at_avg'
  df_year['at_rolling_avg'] = df_year['at'].rolling(window=2).mean()
  df_year['sale_to_at_avg'] = df_year['sale'] / df_year['at_rolling_avg']
```

```
    # Add all computed ratio columns to describe list
    computed_ratios = ['dltis_at', 'dlc_at', 'dltt_at', 'ppent_at',␣
↪'mkvalt_at', 'ni_at',
                       'net_debt_issued_ratio', 'book_leverage', 'wc_ta',␣
↪'re_ta',
                       'ebit_ta', 'mv_tl', 'sales_ta', 'z_score',␣
↪'sale_to_at_avg']

    columns_to_describe.extend(computed_ratios)

    # Print descriptive statistics for each column
    for col in columns_to_describe:
        if col in df_year:
            print(f"Descriptive statistics for {col} in the full dataset:")
            print(df_year[col].describe())
            print("\n")
        else:
            print(f"Column {col} not found in dataframe for the year {year}.\n")

    return df_year  # You can return the filtered dataframe with additional␣
↪columns if needed
```

[115]:
```
describe_yearly_stats_full_period(compustat_copy)
```

```
Descriptive statistics for ch in the full dataset:
count    406899.000000
mean        445.400369
std        7002.577159
min        -279.141000
25%           0.889000
50%           7.523000
75%          52.484000
max      656125.678000
Name: ch, dtype: float64


Descriptive statistics for ivst in the full dataset:
count    403791.000000
mean        613.747949
std       10564.662525
min         -19.538000
25%           0.000000
50%           0.000000
75%          11.611000
max      976094.000000
Name: ivst, dtype: float64
```

```
Descriptive statistics for rect in the full dataset:
count    4.094900e+05
mean     1.941398e+03
std      3.381064e+04
min     -1.670000e-01
25%      1.800000e+00
50%      1.412600e+01
75%      1.202505e+02
max      4.126749e+06
Name: rect, dtype: float64


Descriptive statistics for invt in the full dataset:
count    417232.000000
mean        272.674272
std        5220.752731
min          -3.116000
25%           0.000000
50%           3.591000
75%          34.195000
max      503428.000000
Name: invt, dtype: float64


Descriptive statistics for aco in the full dataset:
count    388386.000000
mean         74.082254
std         880.971446
min        -505.200000
25%           0.091000
50%           0.909000
75%           8.869750
max      158372.600000
Name: aco, dtype: float64


Descriptive statistics for act in the full dataset:
count    359184.000000
mean        704.445985
std        4311.255963
min          -7.760000
25%           6.892000
50%          36.900000
75%         204.110500
max      239728.503000
Name: act, dtype: float64
```

```
Descriptive statistics for ppent in the full dataset:
count    451750.000000
mean        817.341484
std        5471.692199
min           0.000000
25%           2.200000
50%          15.300000
75%         123.260750
max      292684.091000
Name: ppent, dtype: float64


Descriptive statistics for ivaeq in the full dataset:
count    419094.000000
mean         83.984743
std        1006.220139
min       -3858.705000
25%           0.000000
50%           0.000000
75%           0.000000
max      104763.813000
Name: ivaeq, dtype: float64


Descriptive statistics for ivao in the full dataset:
count    3.989120e+05
mean     1.149799e+03
std      1.948064e+04
min     -3.208660e+02
25%      0.000000e+00
50%      0.000000e+00
75%      8.974000e+00
max      2.080428e+06
Name: ivao, dtype: float64


Descriptive statistics for intan in the full dataset:
count    422599.000000
mean        477.086450
std        3940.790622
min         -40.455000
25%           0.000000
50%           0.248000
75%          22.000000
max      310197.000000
Name: intan, dtype: float64
```

```
Descriptive statistics for ao in the full dataset:
count    4.573970e+05
mean     8.278769e+02
std      1.512049e+04
min     -1.113900e+04
25%      2.080000e-01
50%      2.963000e+00
75%      3.300100e+01
max      1.630545e+06
Name: ao, dtype: float64


Descriptive statistics for at in the full dataset:
count    4.630410e+05
mean     8.314195e+03
std      8.676180e+04
min      0.000000e+00
25%      2.139400e+01
50%      1.603270e+02
75%      1.145303e+03
max      4.305288e+06
Name: at, dtype: float64


Descriptive statistics for dlc in the full dataset:
count    452467.000000
mean        782.978830
std       11104.347872
min       -3753.453000
25%           0.103000
50%           2.259000
75%          25.400000
max      614237.411000
Name: dlc, dtype: float64


Descriptive statistics for ap in the full dataset:
count    4.060910e+05
mean     1.742079e+03
std      3.061096e+04
min     -2.457000e+00
25%      1.034000e+00
50%      6.707000e+00
75%      6.686900e+01
max      2.462303e+06
Name: ap, dtype: float64
```

```
Descriptive statistics for txp in the full dataset:
count    374728.000000
mean         22.793583
std         192.978807
min       -1346.000000
25%           0.000000
50%           0.000000
75%           1.590000
max       17656.000000
Name: txp, dtype: float64


Descriptive statistics for lco in the full dataset:
count    368989.000000
mean        241.435656
std        1853.296957
min         -89.000000
25%           0.695000
50%           5.286000
75%          42.473000
max      382408.761000
Name: lco, dtype: float64


Descriptive statistics for lct in the full dataset:
count    364930.000000
mean        554.985023
std        3886.453338
min          -0.002000
25%           3.443000
50%          17.100000
75%         106.853500
max      329795.000000
Name: lct, dtype: float64


Descriptive statistics for dltt in the full dataset:
count    4.610140e+05
mean     1.521038e+03
std      3.208907e+04
min     -2.300000e-02
25%      2.700000e-01
50%      1.104150e+01
75%      1.582927e+02
max      4.211684e+06
Name: dltt, dtype: float64
```

```
Descriptive statistics for lo in the full dataset:
count    4.570080e+05
mean     1.519697e+03
std      2.512706e+04
min     -7.500000e+02
25%      0.000000e+00
50%      1.356000e+00
75%      2.665200e+01
max      2.095003e+06
Name: lo, dtype: float64


Descriptive statistics for txditc in the full dataset:
count    384230.000000
mean        111.706921
std         913.871160
min        -285.769000
25%           0.000000
50%           0.000000
75%           4.943750
max       89679.000000
Name: txditc, dtype: float64


Descriptive statistics for mib in the full dataset:
count    444827.000000
mean         25.158683
std         508.806061
min       -3121.206000
25%           0.000000
50%           0.000000
75%           0.000000
max       83429.587000
Name: mib, dtype: float64


Descriptive statistics for lt in the full dataset:
count    4.573780e+05
mean     7.176778e+03
std      8.201443e+04
min      0.000000e+00
25%      8.577000e+00
50%      7.649550e+01
75%      7.561302e+02
max      4.245011e+06
Name: lt, dtype: float64
```

```
Descriptive statistics for pstk in the full dataset:
count    453361.000000
mean         41.573113
std        1325.931224
min        -252.000000
25%           0.000000
50%           0.000000
75%           0.000000
max      139966.000000
Name: pstk, dtype: float64


Descriptive statistics for ceq in the full dataset:
count    447262.000000
mean       1138.411916
std        7534.489899
min     -139965.000000
25%           6.560000
50%          49.075000
75%         297.761500
max      506199.000000
Name: ceq, dtype: float64


Descriptive statistics for teq in the full dataset:
count    152491.000000
mean       2718.255525
std       12938.462345
min      -86154.000000
25%          24.234000
50%         171.399000
75%         967.867000
max      514930.000000
Name: teq, dtype: float64


Descriptive statistics for sale in the full dataset:
count    422927.000000
mean       1770.227546
std       10537.647421
min      -24954.684000
25%          13.098000
50%          76.132000
75%         479.760000
max      608481.000000
Name: sale, dtype: float64
```

```
Descriptive statistics for cogs in the full dataset:
count    417248.000000
mean       1204.460375
std        7932.089809
min      -23105.164000
25%          7.499000
50%         45.520000
75%        300.930250
max      452776.000000
Name: cogs, dtype: float64


Descriptive statistics for xsga in the full dataset:
count    341994.000000
mean        303.137622
std        1854.424488
min        -283.000000
25%           4.075000
50%          17.177500
75%          83.402750
max      211641.000000
Name: xsga, dtype: float64


Descriptive statistics for oibdp in the full dataset:
count    410698.000000
mean        339.355899
std        2276.027786
min      -76735.000000
25%          0.204000
50%          8.058000
75%         69.853500
max      130622.000000
Name: oibdp, dtype: float64


Descriptive statistics for dp in the full dataset:
count    437396.000000
mean        101.811435
std         687.282416
min        -112.000000
25%           0.412000
50%           2.426000
75%          19.565000
max       52892.000000
Name: dp, dtype: float64
```

```
Descriptive statistics for oiadp in the full dataset:
count    418594.000000
mean        239.020333
std        1850.694369
min      -80053.000000
25%          -0.161000
50%           5.131000
75%          49.000000
max      130622.000000
Name: oiadp, dtype: float64


Descriptive statistics for xint in the full dataset:
count    414313.000000
mean        113.664527
std        1381.102480
min       -3775.000000
25%           0.170000
50%           1.840000
75%          19.242000
max      137861.000000
Name: xint, dtype: float64


Descriptive statistics for nopi in the full dataset:
count    418449.00000
mean         -1.43254
std         622.16742
min      -87714.35800
25%           0.00000
50%           0.16200
75%           2.00000
max       26728.02200
Name: nopi, dtype: float64


Descriptive statistics for spi in the full dataset:
count    433471.000000
mean        -18.837227
std         433.230824
min      -51066.200000
25%          -0.240000
50%           0.000000
75%           0.000000
max      120517.000000
Name: spi, dtype: float64
```

```
Descriptive statistics for pi in the full dataset:
count    462503.000000
mean        175.080063
std        1495.888460
min     -108761.000000
25%          -0.583000
50%           3.728000
75%          35.773000
max      119103.000000
Name: pi, dtype: float64


Descriptive statistics for txt in the full dataset:
count    463379.000000
mean         52.468380
std         464.215586
min      -45415.000000
25%           0.000000
50%           1.012000
75%          10.190000
max       49860.000000
Name: txt, dtype: float64


Descriptive statistics for mii in the full dataset:
count    430733.000000
mean          5.754814
std         109.088535
min      -15835.301000
25%           0.000000
50%           0.000000
75%           0.000000
max       12050.000000
Name: mii, dtype: float64


Descriptive statistics for ib in the full dataset:
count    463625.000000
mean        116.922237
std        1146.018130
min      -99289.000000
25%          -0.571000
50%           2.363000
75%          23.897000
max      104821.000000
Name: ib, dtype: float64
```

```
Descriptive statistics for dvp in the full dataset:
count    463055.000000
mean          3.975039
std         241.710893
min       -1013.000000
25%           0.000000
50%           0.000000
75%           0.000000
max       85419.000000
Name: dvp, dtype: float64


Descriptive statistics for cstke in the full dataset:
count    461827.000000
mean         -0.349960
std          15.979401
min       -2741.588000
25%           0.000000
50%           0.000000
75%           0.000000
max        2631.000000
Name: cstke, dtype: float64


Descriptive statistics for xido in the full dataset:
count    459445.000000
mean          1.215287
std         212.648180
min      -54122.000000
25%           0.000000
50%           0.000000
75%           0.000000
max       80220.090000
Name: xido, dtype: float64


Descriptive statistics for ni in the full dataset:
count    419354.000000
mean        107.634358
std        1112.635482
min      -99289.000000
25%          -0.938750
50%           1.851000
75%          21.125750
max      104821.000000
Name: ni, dtype: float64
```

```
Descriptive statistics for ibc in the full dataset:
count    365210.000000
mean        120.109972
std        1167.990702
min      -99289.000000
25%          -1.582000
50%           1.410000
75%          23.287750
max      105217.000000
Name: ibc, dtype: float64


Descriptive statistics for dpc in the full dataset:
count    359307.000000
mean        117.143465
std         752.937743
min        -155.137000
25%           0.476000
50%           3.177000
75%          27.058000
max       52444.000000
Name: dpc, dtype: float64


Descriptive statistics for xidoc in the full dataset:
count    359765.000000
mean          1.493381
std         165.900289
min      -11489.991000
25%           0.000000
50%           0.000000
75%           0.000000
max       80220.090000
Name: xidoc, dtype: float64


Descriptive statistics for txdc in the full dataset:
count    354334.000000
mean          0.393190
std         188.237958
min      -35561.000000
25%           0.000000
50%           0.000000
75%           0.238000
max       24877.000000
Name: txdc, dtype: float64
```

```
Descriptive statistics for esubc in the full dataset:
count    318407.000000
mean         -1.953680
std         116.003878
min      -31530.688000
25%           0.000000
50%           0.000000
75%           0.000000
max       10600.000000
Name: esubc, dtype: float64


Descriptive statistics for sppiv in the full dataset:
count    267298.000000
mean        -11.701960
std         221.996853
min      -26151.137000
25%          -0.021000
50%           0.000000
75%           0.000000
max       16560.454000
Name: sppiv, dtype: float64


Descriptive statistics for fopo in the full dataset:
count    365212.000000
mean         53.067473
std         885.566438
min     -112954.000000
25%           0.000000
50%           0.284000
75%           6.176000
max      106160.000000
Name: fopo, dtype: float64


Descriptive statistics for fopt in the full dataset:
count    97354.000000
mean        61.121444
std        383.237818
min      -1102.700000
25%          0.310000
50%          2.723000
75%         16.330250
max      20846.305000
Name: fopt, dtype: float64
```

```
Descriptive statistics for recch in the full dataset:
count    237765.000000
mean        -21.962794
std         551.337383
min     -122944.117000
25%          -5.931000
50%          -0.297000
75%           0.329000
max       37779.533000
Name: recch, dtype: float64


Descriptive statistics for invch in the full dataset:
count    247555.000000
mean        -33.765379
std        1322.532817
min     -140808.000000
25%          -1.519000
50%           0.000000
75%           0.028000
max      186295.000000
Name: invch, dtype: float64


Descriptive statistics for apalch in the full dataset:
count    188010.000000
mean         15.576379
std         498.933945
min      -62084.000000
25%          -0.518000
50%           0.293000
75%           4.015000
max       99786.675000
Name: apalch, dtype: float64


Descriptive statistics for txach in the full dataset:
count    198687.000000
mean          0.699928
std          82.085530
min       -5348.000000
25%           0.000000
50%           0.000000
75%           0.000000
max       18183.000000
Name: txach, dtype: float64
```

```
Descriptive statistics for aoloch in the full dataset:
count    268337.000000
mean         27.080485
std        2209.434884
min     -180279.000000
25%          -2.000000
50%           0.000000
75%           1.843000
max      214365.000000
Name: aoloch, dtype: float64


Descriptive statistics for oancf in the full dataset:
count    268755.000000
mean        348.826408
std        2679.365613
min     -140537.000000
25%          -0.720000
50%           6.450000
75%          85.705500
max      182220.000000
Name: oancf, dtype: float64


Descriptive statistics for ivch in the full dataset:
count    3.458270e+05
mean     7.081057e+02
std      1.576654e+04
min     -7.101990e+02
25%      0.000000e+00
50%      0.000000e+00
75%      9.320000e-01
max      3.670475e+06
Name: ivch, dtype: float64


Descriptive statistics for siv in the full dataset:
count    3.400620e+05
mean     6.123588e+02
std      1.448178e+04
min     -7.149000e+02
25%      0.000000e+00
50%      0.000000e+00
75%      1.500000e-01
max      3.551455e+06
Name: siv, dtype: float64
```

```
Descriptive statistics for capx in the full dataset:
count    388739.000000
mean        140.438538
std         967.924944
min       -3258.000000
25%           0.328000
50%           2.982000
75%          25.030000
max       65028.000000
Name: capx, dtype: float64


Descriptive statistics for sppe in the full dataset:
count    295295.000000
mean         14.732395
std         261.674748
min        -649.000000
25%           0.000000
50%           0.000000
75%           0.209000
max       18115.000000
Name: sppe, dtype: float64


Descriptive statistics for aqc in the full dataset:
count    347747.000000
mean         42.181179
std         557.025783
min      -48631.257000
25%           0.000000
50%           0.000000
75%           0.000000
max       66611.000000
Name: aqc, dtype: float64


Descriptive statistics for ivstch in the full dataset:
count    214133.000000
mean         62.917435
std        3893.821861
min     -385559.000000
25%           0.000000
50%           0.000000
75%           0.000000
max      482649.593000
Name: ivstch, dtype: float64
```

```
Descriptive statistics for ivaco in the full dataset:
count    268712.000000
mean          9.036377
std        1169.970324
min     -263779.000000
25%          -0.177000
50%           0.000000
75%           0.082000
max       94531.000000
Name: ivaco, dtype: float64


Descriptive statistics for ivncf in the full dataset:
count    268758.000000
mean       -293.633536
std        4106.698930
min     -313291.000000
25%          -83.184250
50%           -7.496000
75%           -0.154000
max      540050.000000
Name: ivncf, dtype: float64


Descriptive statistics for sstk in the full dataset:
count    360211.000000
mean         34.838193
std         461.828364
min        -516.713000
25%           0.000000
50%           0.145000
75%           4.618000
max       77490.000000
Name: sstk, dtype: float64


Descriptive statistics for prstkc in the full dataset:
count    348228.000000
mean         51.616782
std         668.350853
min        -372.200000
25%           0.000000
50%           0.000000
75%           0.239000
max       95625.000000
Name: prstkc, dtype: float64
```

```
Descriptive statistics for dv in the full dataset:
count    361075.000000
mean         60.029676
std         455.822528
min       -1631.969000
25%           0.000000
50%           0.000000
75%           3.971000
max       82452.000000
Name: dv, dtype: float64


Descriptive statistics for dltis in the full dataset:
count    3.559040e+05
mean     5.283391e+02
std      2.048217e+04
min     -7.700000e+02
25%      0.000000e+00
50%      3.720000e-01
75%      2.950000e+01
max      5.474177e+06
Name: dltis, dtype: float64


Descriptive statistics for dltr in the full dataset:
count    3.573200e+05
mean     4.866380e+02
std      2.124433e+04
min     -5.935690e+02
25%      2.000000e-03
50%      1.124000e+00
75%      2.479525e+01
max      5.451894e+06
Name: dltr, dtype: float64


Descriptive statistics for dlcch in the full dataset:
count    173238.000000
mean         -7.858156
std        1701.748709
min     -189419.000000
25%          -0.366000
50%           0.000000
75%           0.535000
max      167482.474000
Name: dlcch, dtype: float64
```

```
Descriptive statistics for fiao in the full dataset:
count    268712.000000
mean         41.715142
std        2384.603498
min     -180321.866000
25%          -0.513250
50%           0.000000
75%           0.000000
max      603185.000000
Name: fiao, dtype: float64


Descriptive statistics for fincf in the full dataset:
count    268761.000000
mean        -12.339776
std        4260.641864
min     -561116.000000
25%          -9.857000
50%           0.220000
75%          15.326000
max      596645.000000
Name: fincf, dtype: float64


Descriptive statistics for exre in the full dataset:
count    268148.000000
mean         -1.128972
std         197.788316
min      -36894.824000
25%           0.000000
50%           0.000000
75%           0.000000
max       42700.637000
Name: exre, dtype: float64


Descriptive statistics for chech in the full dataset:
count    303771.000000
mean         37.597608
std        1862.942789
min     -173600.000000
25%          -2.420000
50%           0.056000
75%           6.615000
max      263978.000000
Name: chech, dtype: float64
```

```
Descriptive statistics for fsrco in the full dataset:
count    92712.000000
mean        15.471849
std        193.066399
min      -2392.646000
25%          0.000000
50%          0.028000
75%          0.844000
max      19863.004000
Name: fsrco, dtype: float64


Descriptive statistics for fuseo in the full dataset:
count    92722.000000
mean        14.575770
std        204.017407
min      -3406.500000
25%          0.000000
50%          0.077000
75%          1.059000
max      26773.613000
Name: fuseo, dtype: float64


Descriptive statistics for wcapc in the full dataset:
count    83715.000000
mean         2.614494
std         84.954285
min      -9784.000000
25%         -0.463000
50%          0.282000
75%          2.741000
max       4600.185000
Name: wcapc, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the full dataset:
count    1.661380e+05
mean              NaN
std               NaN
min              -inf
25%     -1.167043e-02
50%      0.000000e+00
75%      3.850619e-02
max               inf
Name: net_debt_issued_ratio, dtype: float64
```

```
Descriptive statistics for book_leverage in the full dataset:
count    4.375610e+05
mean              inf
std               NaN
min     -2.966091e+03
25%      9.078025e-02
50%      3.583305e-01
75%      5.869266e-01
max               inf
Name: book_leverage, dtype: float64


Descriptive statistics for wc_ta in the full dataset:
count    3.585970e+05
mean             -inf
std               NaN
min              -inf
25%      2.558635e-02
50%      2.125069e-01
75%      4.167516e-01
max      1.623810e+01
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the full dataset:
count    4.363070e+05
mean              NaN
std               NaN
min              -inf
25%     -2.391467e-01
50%      4.960743e-02
75%      2.321377e-01
max               inf
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the full dataset:
count    4.179830e+05
mean              NaN
std               NaN
min              -inf
25%     -1.480489e-02
50%      5.383142e-02
75%      1.125750e-01
max               inf
Name: ebit_ta, dtype: float64
```

```
Descriptive statistics for mv_tl in the full dataset:
count    3.371610e+05
mean              inf
std               NaN
min      0.000000e+00
25%      4.806794e-01
50%      1.423210e+00
75%      4.325385e+00
max               inf
Name: mv_tl, dtype: float64


Descriptive statistics for sales_ta in the full dataset:
count    4.190630e+05
mean              inf
std               NaN
min     -3.158693e+01
25%      2.736820e-01
50%      8.125681e-01
75%      1.458916e+00
max               inf
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the full dataset:
count    2.755110e+05
mean              inf
std               NaN
min     -3.071689e+05
25%      1.301269e+00
50%      2.927136e+00
75%      5.035409e+00
max               inf
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the full dataset:
count    4.079820e+05
mean              inf
std               NaN
min     -2.263291e+00
25%      2.604414e-01
50%      8.152254e-01
75%      1.489847e+00
max               inf
Name: sale_to_at_avg, dtype: float64
```

```
Descriptive statistics for dltis_at in the full dataset:
count    3.548080e+05
mean              inf
std               NaN
min     -2.416232e+00
25%      0.000000e+00
50%      8.804827e-03
75%      1.015651e-01
max               inf
Name: dltis_at, dtype: float64


Descriptive statistics for dlc_at in the full dataset:
count    4.517670e+05
mean              inf
std               NaN
min     -7.006369e-02
25%      2.307724e-03
50%      2.569249e-02
75%      8.708780e-02
max               inf
Name: dlc_at, dtype: float64


Descriptive statistics for dltt_at in the full dataset:
count    4.576360e+05
mean              inf
std               NaN
min     -3.490137e-02
25%      1.009360e-02
50%      1.152594e-01
75%      2.957725e-01
max               inf
Name: dltt_at, dtype: float64


Descriptive statistics for ppent_at in the full dataset:
count    450560.000000
mean          0.259927
std           0.262723
min           0.000000
25%           0.031290
50%           0.174150
75%           0.408987
max           2.931969
Name: ppent_at, dtype: float64
```

```
Descriptive statistics for mkvalt_at in the full dataset:
count    1.772760e+05
mean              inf
std               NaN
min      0.000000e+00
25%      2.176615e-01
50%      7.216713e-01
75%      1.772633e+00
max               inf
Name: mkvalt_at, dtype: float64


Descriptive statistics for ni_at in the full dataset:
count    4.190340e+05
mean              NaN
std               NaN
min              -inf
25%     -4.100636e-02
50%      2.347948e-02
75%      6.626584e-02
max               inf
Name: ni_at, dtype: float64


Descriptive statistics for net_debt_issued_ratio in the full dataset:
count    1.661380e+05
mean              NaN
std               NaN
min              -inf
25%     -1.167043e-02
50%      0.000000e+00
75%      3.850619e-02
max               inf
Name: net_debt_issued_ratio, dtype: float64


Descriptive statistics for book_leverage in the full dataset:
count    4.375610e+05
mean              inf
std               NaN
min     -2.966091e+03
25%      9.078025e-02
50%      3.583305e-01
75%      5.869266e-01
max               inf
Name: book_leverage, dtype: float64
```

```
Descriptive statistics for wc_ta in the full dataset:
count    3.585970e+05
mean             -inf
std               NaN
min              -inf
25%      2.558635e-02
50%      2.125069e-01
75%      4.167516e-01
max      1.623810e+01
Name: wc_ta, dtype: float64


Descriptive statistics for re_ta in the full dataset:
count    4.363070e+05
mean              NaN
std               NaN
min              -inf
25%     -2.391467e-01
50%      4.960743e-02
75%      2.321377e-01
max               inf
Name: re_ta, dtype: float64


Descriptive statistics for ebit_ta in the full dataset:
count    4.179830e+05
mean              NaN
std               NaN
min              -inf
25%     -1.480489e-02
50%      5.383142e-02
75%      1.125750e-01
max               inf
Name: ebit_ta, dtype: float64


Descriptive statistics for mv_tl in the full dataset:
count    3.371610e+05
mean              inf
std               NaN
min      0.000000e+00
25%      4.806794e-01
50%      1.423210e+00
75%      4.325385e+00
max               inf
Name: mv_tl, dtype: float64
```

```
Descriptive statistics for sales_ta in the full dataset:
count    4.190630e+05
mean              inf
std               NaN
min     -3.158693e+01
25%      2.736820e-01
50%      8.125681e-01
75%      1.458916e+00
max               inf
Name: sales_ta, dtype: float64


Descriptive statistics for z_score in the full dataset:
count    2.755110e+05
mean              inf
std               NaN
min     -3.071689e+05
25%      1.301269e+00
50%      2.927136e+00
75%      5.035409e+00
max               inf
Name: z_score, dtype: float64


Descriptive statistics for sale_to_at_avg in the full dataset:
count    4.079820e+05
mean              inf
std               NaN
min     -2.263291e+00
25%      2.604414e-01
50%      8.152254e-01
75%      1.489847e+00
max               inf
Name: sale_to_at_avg, dtype: float64
```

[115]:

	gvkey	datadate	fyear	indfmt	consol	popsrc	datafmt	tic	\
0	1000	1961-12-31	1961	INDL	C	D	STD	AE.2	
1	1000	1962-12-31	1962	INDL	C	D	STD	AE.2	
2	1000	1963-12-31	1963	INDL	C	D	STD	AE.2	
3	1000	1964-12-31	1964	INDL	C	D	STD	AE.2	
4	1000	1965-12-31	1965	INDL	C	D	STD	AE.2	
…	…	…	…	…	…	…	…	…	
539313	352262	2022-12-31	2022	INDL	C	D	STD	CLCO	
539314	353444	2021-12-31	2021	INDL	C	D	STD	HLN	

```
539315  353444 2022-12-31   2022   INDL      C      D     STD    HLN
539316  353945 2021-12-31   2021   INDL      C      D     STD  ACLLY
539317  353945 2022-12-31   2022   INDL      C      D     STD  ACLLY

            cusip                       conm  … net_debt_issued_ratio  \
0       000032102     A & E PLASTIK PAK INC  …                    NaN
1       000032102     A & E PLASTIK PAK INC  …                    NaN
2       000032102     A & E PLASTIK PAK INC  …                    NaN
3       000032102     A & E PLASTIK PAK INC  …                    NaN
4       000032102     A & E PLASTIK PAK INC  …                    NaN
…           …                          …  …                    …
539313  G2415A113          COOL COMPANY LTD  …                    NaN
539314  405552100                HALEON PLC  …                    NaN
539315  405552100                HALEON PLC  …                    NaN
539316  00449R109  ACCELLERON INDUSTRIES AG  …                    NaN
539317  00449R109  ACCELLERON INDUSTRIES AG  …                    NaN

       book_leverage     wc_ta     re_ta   ebit_ta     mv_tl  sales_ta  \
0                NaN       NaN       NaN       NaN       NaN       NaN
1                NaN       NaN       NaN       NaN       NaN       NaN
2                NaN       NaN       NaN       NaN       NaN       NaN
3           0.501233  0.318503  0.000706  0.052260       NaN  1.435028
4           0.747558  0.044156 -0.084848 -0.104762       NaN  0.730736
…              …       …       …       …       …       …
539313      0.637962 -0.064746  0.041662  0.053903       NaN  0.103485
539314      0.036239  0.029404  0.764971  0.060376       NaN  0.277060
539315      0.389974 -0.008933  0.466437  0.069453  1.669938  0.311877
539316      0.298961  0.161409       NaN  0.266705       NaN  1.102583
539317      0.552871  0.334683  0.189824  0.154994  2.831915  0.795209

         z_score at_rolling_avg sale_to_at_avg
0            NaN            NaN            NaN
1            NaN            NaN            NaN
2            NaN            NaN            NaN
3            NaN            NaN            NaN
4            NaN         1.8630       0.906065
…          …            …             …
539313       NaN     35212.2165       0.006048
539314       NaN     24354.0800       0.530707
539315  2.182208     44299.3465       0.295327
539316       NaN     21317.3395       0.035486
539317  3.665261       833.8180       0.936101

[539318 rows x 998 columns]
```

[116]:
```
compustat_copy['datadate'].min()
```

[116]: Timestamp('1950-06-30 00:00:00')

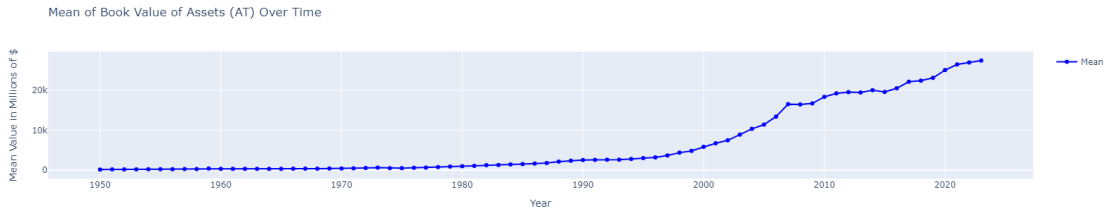
[117]:
```
compustat_copy['datadate'].max()
```

[117]: Timestamp('2023-09-30 00:00:00')

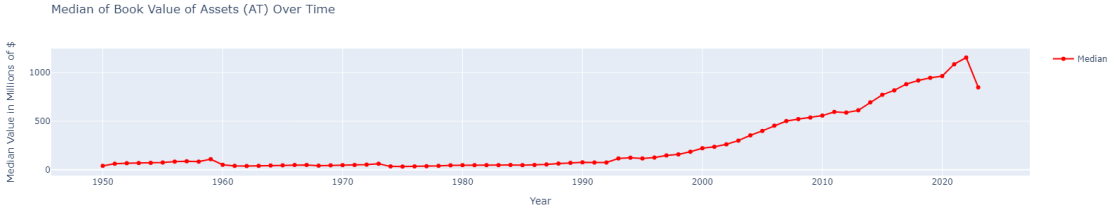
[118]:
```
def mean_median_of_column_by_year(df, year, column_name='at'):
    # Filter the dataframe for the given year
    df_year = df[df['datadate'].dt.year == year]
    # Return the mean and median of the specified column
    return df_year[column_name].mean(), df_year[column_name].median()
```

[119]:
```
from plotly.offline import init_notebook_mode, iplot
import plotly.graph_objs as go

# Activate the notebook mode for Plotly
init_notebook_mode(connected=True)

# Initialize lists to store mean and median values
means = []
medians = []
years = list(range(1950, 2024))

# Compute mean and median for each year
for year in years:
    mean_val, median_val = mean_median_of_column_by_year(compustat_copy, year)
    means.append(mean_val)
    medians.append(median_val)

# Create trace for the mean
mean_trace = go.Scatter(
    x=years,
    y=means,
    mode='lines+markers',
    name='Mean',
    marker=dict(color='blue'),
    line=dict(shape='linear')
)

# Layout for mean
mean_layout = go.Layout(
    title='Mean of Book Value of Assets (AT) Over Time',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Mean Value in Millions of $'),
    showlegend=True
)
```

```
# Figure for mean
mean_fig = go.Figure(data=[mean_trace], layout=mean_layout)

# Plot the mean within the notebook
iplot(mean_fig)
```



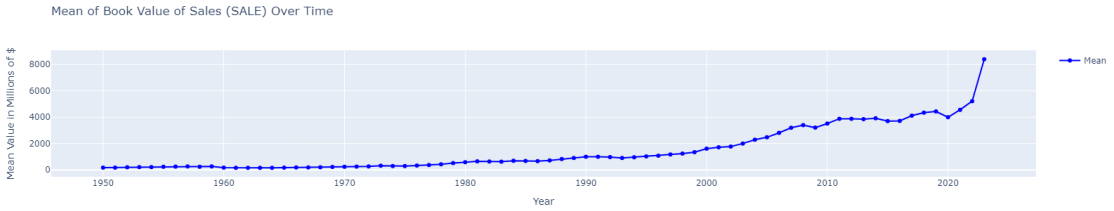
[120]:
```
# Create trace for the median
median_trace = go.Scatter(
    x=years,
    y=medians,
    mode='lines+markers',
    name='Median',
    marker=dict(color='red'),
    line=dict(shape='linear')
)

# Layout for median
median_layout = go.Layout(
    title='Median of Book Value of Assets (AT) Over Time',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Median Value in Millions of $'),
    showlegend=True
)

# Figure for median
median_fig = go.Figure(data=[median_trace], layout=median_layout)

# Plot the median within the notebook
iplot(median_fig)
```



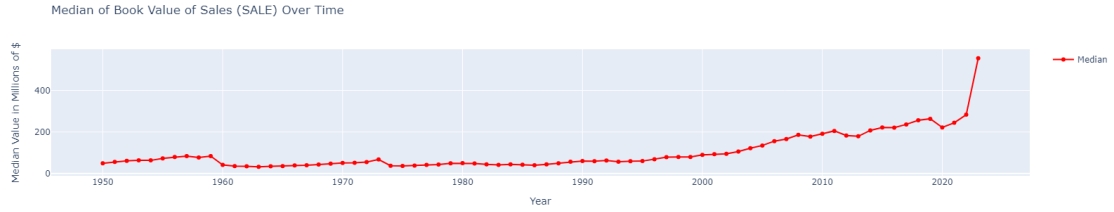
[121]:
```
def mean_median_of_column_by_year(df, year, column_name='sale'):
    # Filter the dataframe for the given year
    df_year = df[df['datadate'].dt.year == year]
    # Return the mean and median of the specified column
    return df_year[column_name].mean(), df_year[column_name].median()
```

[122]:
```
from plotly.offline import init_notebook_mode, iplot
import plotly.graph_objs as go

# Activate the notebook mode for Plotly
init_notebook_mode(connected=True)

# Initialize lists to store mean and median values
means = []
medians = []
years = list(range(1950, 2024))

# Compute mean and median for each year
for year in years:
    mean_val, median_val = mean_median_of_column_by_year(compustat_copy, year)
    means.append(mean_val)
    medians.append(median_val)

# Create trace for the mean
mean_trace = go.Scatter(
    x=years,
    y=means,
    mode='lines+markers',
    name='Mean',
    marker=dict(color='blue'),
    line=dict(shape='linear')
)

# Layout for mean
mean_layout = go.Layout(
    title='Mean of Book Value of Sales (SALE) Over Time',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Mean Value in Millions of $'),
    showlegend=True
)

# Figure for mean
mean_fig = go.Figure(data=[mean_trace], layout=mean_layout)
```

```
# Plot the mean within the notebook
iplot(mean_fig)
```



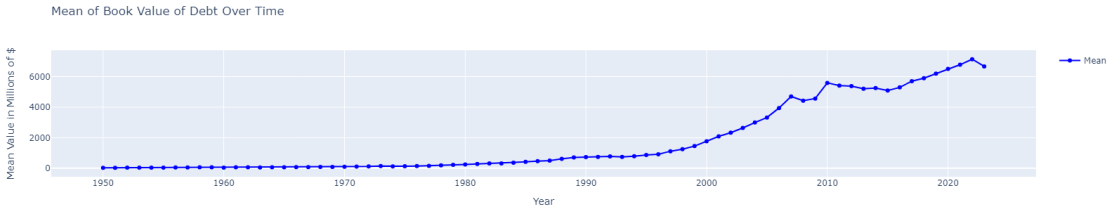
[123]:
```
# Create trace for the median
median_trace = go.Scatter(
    x=years,
    y=medians,
    mode='lines+markers',
    name='Median',
    marker=dict(color='red'),
    line=dict(shape='linear')
)

# Layout for median
median_layout = go.Layout(
    title='Median of Book Value of Sales (SALE) Over Time',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Median Value in Millions of $'),
    showlegend=True
)

# Figure for median
median_fig = go.Figure(data=[median_trace], layout=median_layout)

# Plot the median within the notebook
iplot(median_fig)
```



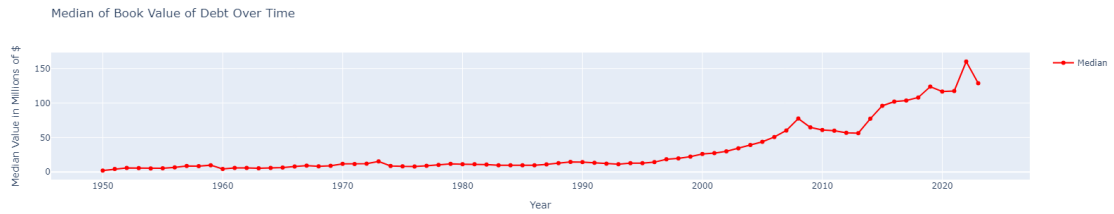
[124]:
```
def mean_median_of_column_by_year(df, year):
    # Filter the dataframe for the given year
    df_year = df[df['datadate'].dt.year == year]
    # Return the mean and median of the specified column
    column_name_1, column_name_2 = 'dlc', 'dltt'
    return (df_year[column_name_1] + df_year[column_name_2]).mean(),␣
 ↪(df_year[column_name_1] + df_year[column_name_2]).median()
```

[125]:
```
from plotly.offline import init_notebook_mode, iplot
import plotly.graph_objs as go

# Activate the notebook mode for Plotly
init_notebook_mode(connected=True)

# Initialize lists to store mean and median values
means = []
medians = []
years = list(range(1950, 2024))

# Compute mean and median for each year
for year in years:
    mean_val, median_val = mean_median_of_column_by_year(compustat_copy, year)
    means.append(mean_val)
    medians.append(median_val)

# Create trace for the mean
mean_trace = go.Scatter(
    x=years,
    y=means,
    mode='lines+markers',
    name='Mean',
    marker=dict(color='blue'),
    line=dict(shape='linear')
)

# Layout for mean
mean_layout = go.Layout(
    title='Mean of Book Value of Debt Over Time',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Mean Value in Millions of $'),
    showlegend=True
)

# Figure for mean
mean_fig = go.Figure(data=[mean_trace], layout=mean_layout)

# Plot the mean within the notebook
```

```
iplot(mean_fig)
```

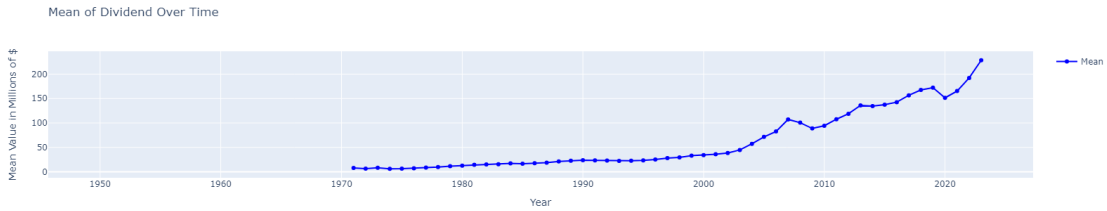


[126]:
```
# Create trace for the median
median_trace = go.Scatter(
    x=years,
    y=medians,
    mode='lines+markers',
    name='Median',
    marker=dict(color='red'),
    line=dict(shape='linear')
)

# Layout for median
median_layout = go.Layout(
    title='Median of Book Value of Debt Over Time',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Median Value in Millions of $'),
    showlegend=True
)

# Figure for median
median_fig = go.Figure(data=[median_trace], layout=median_layout)

# Plot the median within the notebook
iplot(median_fig)
```

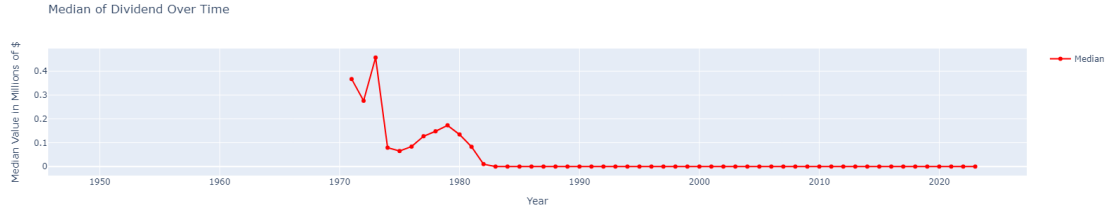


[127]:
```
def mean_median_of_column_by_year(df, year, column_name='dv'):
    # Filter the dataframe for the given year
    df_year = df[df['datadate'].dt.year == year]
    # Return the mean and median of the specified column
    return df_year[column_name].mean(), df_year[column_name].median()
```

[128]:
```
from plotly.offline import init_notebook_mode, iplot
import plotly.graph_objs as go

# Activate the notebook mode for Plotly
init_notebook_mode(connected=True)

# Initialize lists to store mean and median values
means = []
medians = []
years = list(range(1950, 2024))

# Compute mean and median for each year
for year in years:
    mean_val, median_val = mean_median_of_column_by_year(compustat_copy, year)
    means.append(mean_val)
    medians.append(median_val)

# Create trace for the mean
mean_trace = go.Scatter(
    x=years,
    y=means,
    mode='lines+markers',
    name='Mean',
    marker=dict(color='blue'),
    line=dict(shape='linear')
)

# Layout for mean
mean_layout = go.Layout(
    title='Mean of Dividend Over Time',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Mean Value in Millions of $'),
    showlegend=True
)

# Figure for mean
mean_fig = go.Figure(data=[mean_trace], layout=mean_layout)

# Plot the mean within the notebook
iplot(mean_fig)
```



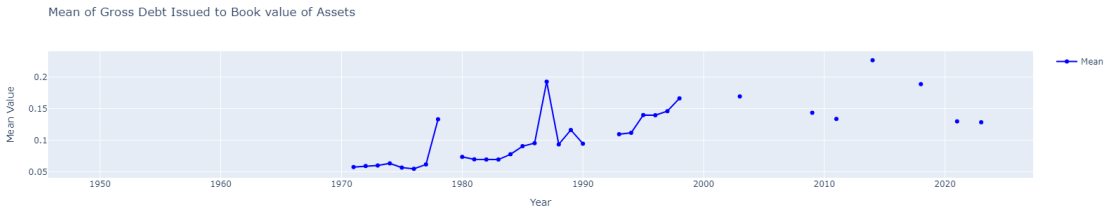
[129]:
```
# Create trace for the median
median_trace = go.Scatter(
    x=years,
    y=medians,
    mode='lines+markers',
    name='Median',
    marker=dict(color='red'),
    line=dict(shape='linear')
)

# Layout for median
median_layout = go.Layout(
    title='Median of Dividend Over Time',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Median Value in Millions of $'),
    showlegend=True
)

# Figure for median
median_fig = go.Figure(data=[median_trace], layout=median_layout)

# Plot the median within the notebook
iplot(median_fig)
```



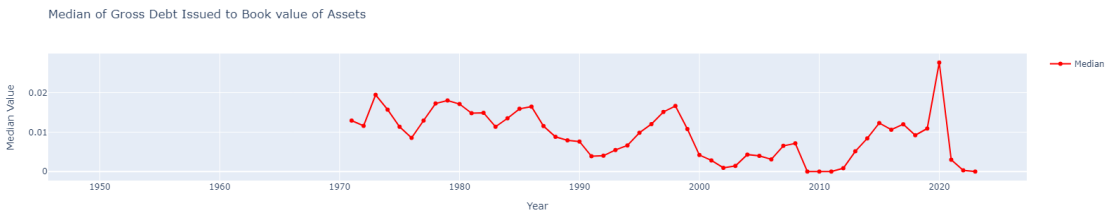
[130]:
```
def mean_median_of_column_by_year(df, year):
    # Filter the dataframe for the given year
```

```
    df_year = df[df['datadate'].dt.year == year]
    # Return the mean and median of the specified column
    column_name_1, column_name_2 = 'dltis', 'at'
    return (df_year[column_name_1] / df_year[column_name_2]).mean(),␣
 ↪(df_year[column_name_1] / df_year[column_name_2]).median()
```

[131]:
```
from plotly.offline import init_notebook_mode, iplot
import plotly.graph_objs as go

# Activate the notebook mode for Plotly
init_notebook_mode(connected=True)

# Initialize lists to store mean and median values
means = []
medians = []
years = list(range(1950, 2024))

# Compute mean and median for each year
for year in years:
    mean_val, median_val = mean_median_of_column_by_year(compustat_copy, year)
    means.append(mean_val)
    medians.append(median_val)

# Create trace for the mean
mean_trace = go.Scatter(
    x=years,
    y=means,
    mode='lines+markers',
    name='Mean',
    marker=dict(color='blue'),
    line=dict(shape='linear')
)

# Layout for mean
mean_layout = go.Layout(
    title='Mean of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Mean Value'),
    showlegend=True
)

# Figure for mean
mean_fig = go.Figure(data=[mean_trace], layout=mean_layout)

# Plot the mean within the notebook
iplot(mean_fig)
```



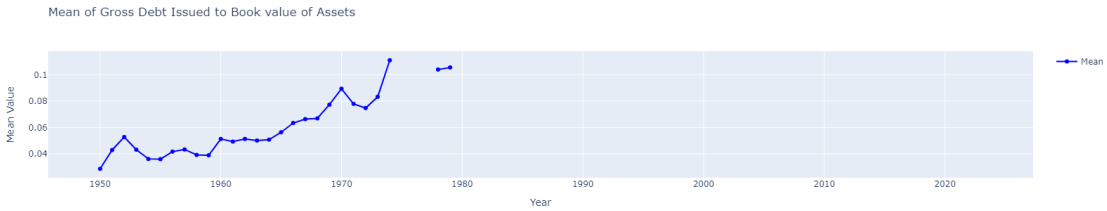
[132]:
```
# Create trace for the median
median_trace = go.Scatter(
    x=years,
    y=medians,
    mode='lines+markers',
    name='Median',
    marker=dict(color='red'),
    line=dict(shape='linear')
)

# Layout for median
median_layout = go.Layout(
    title='Median of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Median Value'),
    showlegend=True
)

# Figure for median
median_fig = go.Figure(data=[median_trace], layout=median_layout)

# Plot the median within the notebook
iplot(median_fig)
```



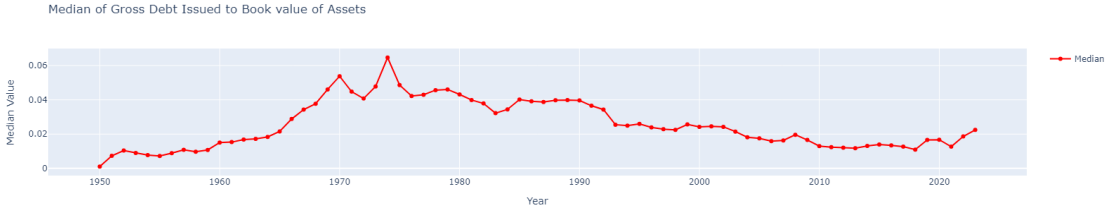
[133]:
```
def mean_median_of_column_by_year(df, year):
    # Filter the dataframe for the given year
```

```
    df_year = df[df['datadate'].dt.year == year]
    # Return the mean and median of the specified column
    column_name_1, column_name_2 = 'dlc', 'at'
    return (df_year[column_name_1] / df_year[column_name_2]).mean(),␣
 ↪(df_year[column_name_1] / df_year[column_name_2]).median()
```

[134]:
```
from plotly.offline import init_notebook_mode, iplot
import plotly.graph_objs as go

# Activate the notebook mode for Plotly
init_notebook_mode(connected=True)

# Initialize lists to store mean and median values
means = []
medians = []
years = list(range(1950, 2024))

# Compute mean and median for each year
for year in years:
    mean_val, median_val = mean_median_of_column_by_year(compustat_copy, year)
    means.append(mean_val)
    medians.append(median_val)

# Create trace for the mean
mean_trace = go.Scatter(
    x=years,
    y=means,
    mode='lines+markers',
    name='Mean',
    marker=dict(color='blue'),
    line=dict(shape='linear')
)

# Layout for mean
mean_layout = go.Layout(
    title='Mean of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Mean Value'),
    showlegend=True
)

# Figure for mean
mean_fig = go.Figure(data=[mean_trace], layout=mean_layout)

# Plot the mean within the notebook
iplot(mean_fig)
```



[135]:
```
# Create trace for the median
median_trace = go.Scatter(
    x=years,
    y=medians,
    mode='lines+markers',
    name='Median',
    marker=dict(color='red'),
    line=dict(shape='linear')
)

# Layout for median
median_layout = go.Layout(
    title='Median of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Median Value'),
    showlegend=True
)

# Figure for median
median_fig = go.Figure(data=[median_trace], layout=median_layout)

# Plot the median within the notebook
iplot(median_fig)
```

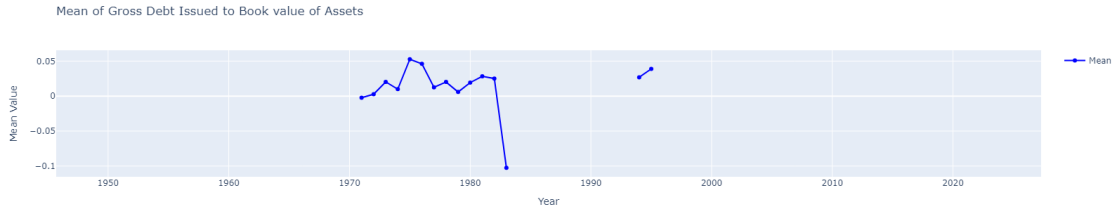


[136]:
```
def mean_median_of_column_by_year(df, year):
    # Filter the dataframe for the given year
```

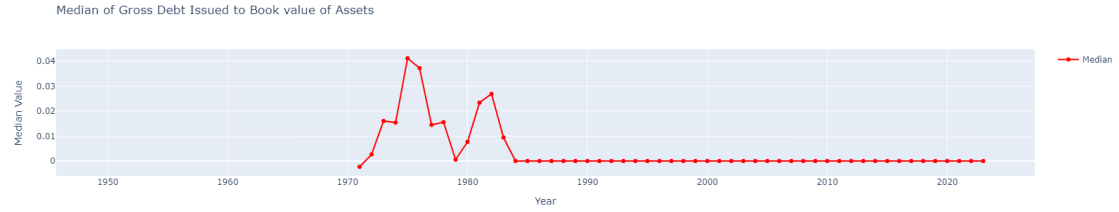
```
    df_year = df[df['datadate'].dt.year == year]
    # Return the mean and median of the specified column
    df_year['net_debt_issued'] = df_year['dltis'] - df_year['dltr'] +␣
 ↪df_year['dlcch']
    df_year['net_debt_issued_ratio'] = df_year['net_debt_issued'] /␣
 ↪df_year['at']
    return df_year['net_debt_issued_ratio'].mean(),␣
 ↪df_year['net_debt_issued_ratio'].median()
```

[137]:
```
from plotly.offline import init_notebook_mode, iplot
import plotly.graph_objs as go

# Activate the notebook mode for Plotly
init_notebook_mode(connected=True)

# Initialize lists to store mean and median values
means = []
medians = []
years = list(range(1950, 2024))

# Compute mean and median for each year
for year in years:
    mean_val, median_val = mean_median_of_column_by_year(compustat_copy, year)
    means.append(mean_val)
    medians.append(median_val)

# Create trace for the mean
mean_trace = go.Scatter(
    x=years,
    y=means,
    mode='lines+markers',
    name='Mean',
    marker=dict(color='blue'),
    line=dict(shape='linear')
)

# Layout for mean
mean_layout = go.Layout(
    title='Mean of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Mean Value'),
    showlegend=True
)

# Figure for mean
mean_fig = go.Figure(data=[mean_trace], layout=mean_layout)
```

```
# Plot the mean within the notebook
iplot(mean_fig)
```



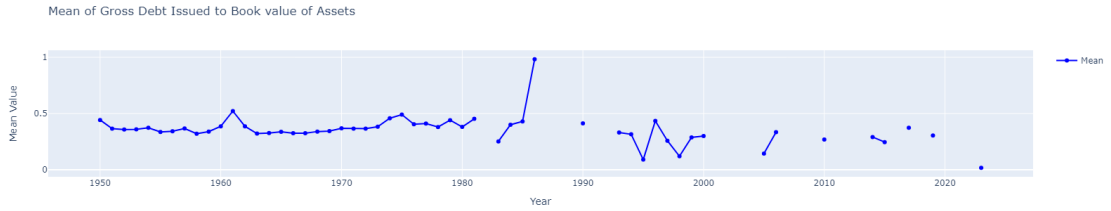
[138]:
```
# Create trace for the median
median_trace = go.Scatter(
    x=years,
    y=medians,
    mode='lines+markers',
    name='Median',
    marker=dict(color='red'),
    line=dict(shape='linear')
)

# Layout for median
median_layout = go.Layout(
    title='Median of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Median Value'),
    showlegend=True
)

# Figure for median
median_fig = go.Figure(data=[median_trace], layout=median_layout)

# Plot the median within the notebook
iplot(median_fig)
```



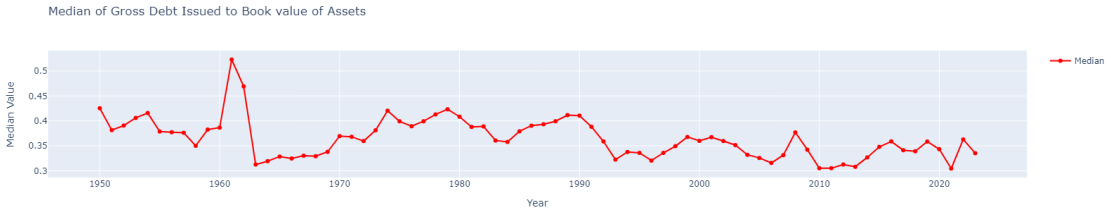
[139]:
```
def mean_median_of_column_by_year(df, year):
    # Filter the dataframe for the given year
    df_year = df[df['datadate'].dt.year == year]
    # Return the mean and median of the specified column
    df_year['book_leverage'] = (df_year['dltt'] + df_year['dlc']) /␣
 ↪(df_year['dltt'] + df_year['dlc'] + df_year['seq'])
    return df_year['book_leverage'].mean(), df_year['book_leverage'].median()
```

[140]:
```
from plotly.offline import init_notebook_mode, iplot
import plotly.graph_objs as go

# Activate the notebook mode for Plotly
init_notebook_mode(connected=True)

# Initialize lists to store mean and median values
means = []
medians = []
years = list(range(1950, 2024))

# Compute mean and median for each year
for year in years:
    mean_val, median_val = mean_median_of_column_by_year(compustat_copy, year)
    means.append(mean_val)
    medians.append(median_val)

# Create trace for the mean
mean_trace = go.Scatter(
    x=years,
    y=means,
    mode='lines+markers',
    name='Mean',
    marker=dict(color='blue'),
    line=dict(shape='linear')
)

# Layout for mean
mean_layout = go.Layout(
    title='Mean of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Mean Value'),
    showlegend=True
)

# Figure for mean
mean_fig = go.Figure(data=[mean_trace], layout=mean_layout)

# Plot the mean within the notebook
```

```
iplot(mean_fig)
```



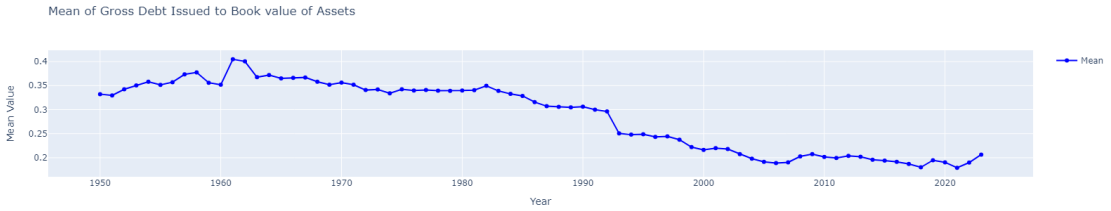
[141]:
```
# Create trace for the median
median_trace = go.Scatter(
    x=years,
    y=medians,
    mode='lines+markers',
    name='Median',
    marker=dict(color='red'),
    line=dict(shape='linear')
)

# Layout for median
median_layout = go.Layout(
    title='Median of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Median Value'),
    showlegend=True
)

# Figure for median
median_fig = go.Figure(data=[median_trace], layout=median_layout)

# Plot the median within the notebook
iplot(median_fig)
```



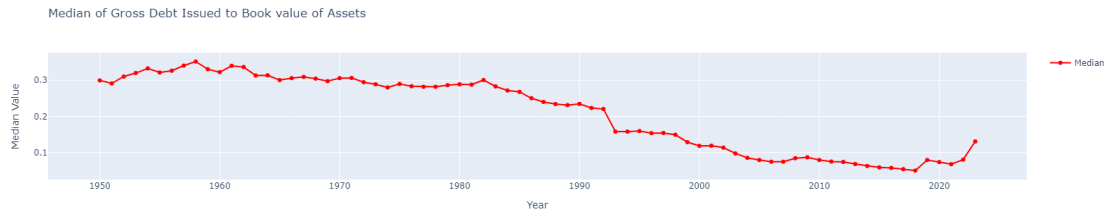
[142]:
```
def mean_median_of_column_by_year(df, year):
    # Filter the dataframe for the given year
    df_year = df[df['datadate'].dt.year == year]
    # Return the mean and median of the specified column
    column_name_1, column_name_2 = 'ppent', 'at'
    return (df_year[column_name_1] / df_year[column_name_2]).mean(),␣
 ↪(df_year[column_name_1] / df_year[column_name_2]).median()
```

[143]:
```
from plotly.offline import init_notebook_mode, iplot
import plotly.graph_objs as go

# Activate the notebook mode for Plotly
init_notebook_mode(connected=True)

# Initialize lists to store mean and median values
means = []
medians = []
years = list(range(1950, 2024))

# Compute mean and median for each year
for year in years:
    mean_val, median_val = mean_median_of_column_by_year(compustat_copy, year)
    means.append(mean_val)
    medians.append(median_val)

# Create trace for the mean
mean_trace = go.Scatter(
    x=years,
    y=means,
    mode='lines+markers',
    name='Mean',
    marker=dict(color='blue'),
    line=dict(shape='linear')
)

# Layout for mean
mean_layout = go.Layout(
    title='Mean of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Mean Value'),
    showlegend=True
)

# Figure for mean
mean_fig = go.Figure(data=[mean_trace], layout=mean_layout)

# Plot the mean within the notebook
```

```
iplot(mean_fig)
```



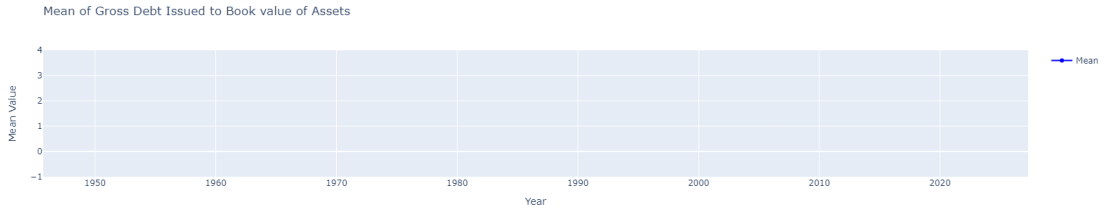
[144]:
```
# Create trace for the median
median_trace = go.Scatter(
    x=years,
    y=medians,
    mode='lines+markers',
    name='Median',
    marker=dict(color='red'),
    line=dict(shape='linear')
)

# Layout for median
median_layout = go.Layout(
    title='Median of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Median Value'),
    showlegend=True
)

# Figure for median
median_fig = go.Figure(data=[median_trace], layout=median_layout)

# Plot the median within the notebook
iplot(median_fig)
```



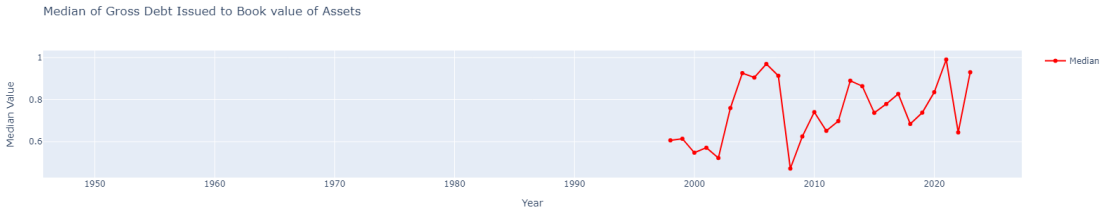
[145]:
```
def mean_median_of_column_by_year(df, year):
    # Filter the dataframe for the given year
    df_year = df[df['datadate'].dt.year == year]
    # Return the mean and median of the specified column
    column_name_1, column_name_2 = 'mkvalt', 'at'
    return (df_year[column_name_1] / df_year[column_name_2]).mean(),␣
 ↪(df_year[column_name_1] / df_year[column_name_2]).median()
```

[146]:
```
from plotly.offline import init_notebook_mode, iplot
import plotly.graph_objs as go

# Activate the notebook mode for Plotly
init_notebook_mode(connected=True)

# Initialize lists to store mean and median values
means = []
medians = []
years = list(range(1950, 2024))

# Compute mean and median for each year
for year in years:
    mean_val, median_val = mean_median_of_column_by_year(compustat_copy, year)
    means.append(mean_val)
    medians.append(median_val)

# Create trace for the mean
mean_trace = go.Scatter(
    x=years,
    y=means,
    mode='lines+markers',
    name='Mean',
    marker=dict(color='blue'),
    line=dict(shape='linear')
)

# Layout for mean
mean_layout = go.Layout(
    title='Mean of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Mean Value'),
    showlegend=True
)

# Figure for mean
mean_fig = go.Figure(data=[mean_trace], layout=mean_layout)

# Plot the mean within the notebook
```

```
iplot(mean_fig)
```



[147]:
```
# Create trace for the median
median_trace = go.Scatter(
    x=years,
    y=medians,
    mode='lines+markers',
    name='Median',
    marker=dict(color='red'),
    line=dict(shape='linear')
)

# Layout for median
median_layout = go.Layout(
    title='Median of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Median Value'),
    showlegend=True
)

# Figure for median
median_fig = go.Figure(data=[median_trace], layout=median_layout)

# Plot the median within the notebook
iplot(median_fig)
```

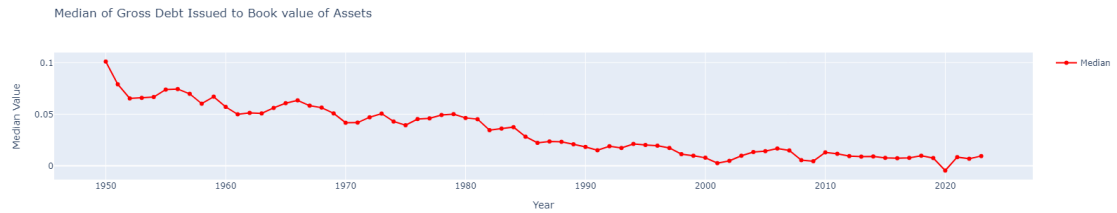


[148]:
```
def mean_median_of_column_by_year(df, year):
    # Filter the dataframe for the given year
    df_year = df[df['datadate'].dt.year == year]
    # Return the mean and median of the specified column
    column_name_1, column_name_2 = 'ni', 'at'
    return (df_year[column_name_1] / df_year[column_name_2]).mean(),␣
 ↪(df_year[column_name_1] / df_year[column_name_2]).median()
```

[149]:
```
from plotly.offline import init_notebook_mode, iplot
import plotly.graph_objs as go

# Activate the notebook mode for Plotly
init_notebook_mode(connected=True)

# Initialize lists to store mean and median values
means = []
medians = []
years = list(range(1950, 2024))

# Compute mean and median for each year
for year in years:
    mean_val, median_val = mean_median_of_column_by_year(compustat_copy, year)
    means.append(mean_val)
    medians.append(median_val)

# Create trace for the mean
mean_trace = go.Scatter(
    x=years,
    y=means,
    mode='lines+markers',
    name='Mean',
    marker=dict(color='blue'),
    line=dict(shape='linear')
)

# Layout for mean
mean_layout = go.Layout(
    title='Mean of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Mean Value'),
    showlegend=True
)

# Figure for mean
mean_fig = go.Figure(data=[mean_trace], layout=mean_layout)

# Plot the mean within the notebook
```

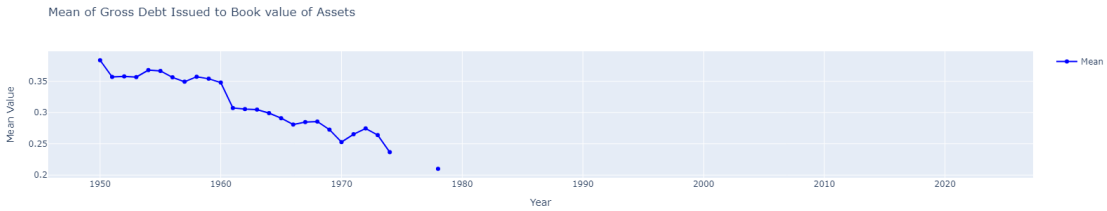
```
iplot(mean_fig)
```

[150]:
```
# Create trace for the median
median_trace = go.Scatter(
    x=years,
    y=medians,
    mode='lines+markers',
    name='Median',
    marker=dict(color='red'),
    line=dict(shape='linear')
)

# Layout for median
median_layout = go.Layout(
    title='Median of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Median Value'),
    showlegend=True
)

# Figure for median
median_fig = go.Figure(data=[median_trace], layout=median_layout)

# Plot the median within the notebook
iplot(median_fig)
```

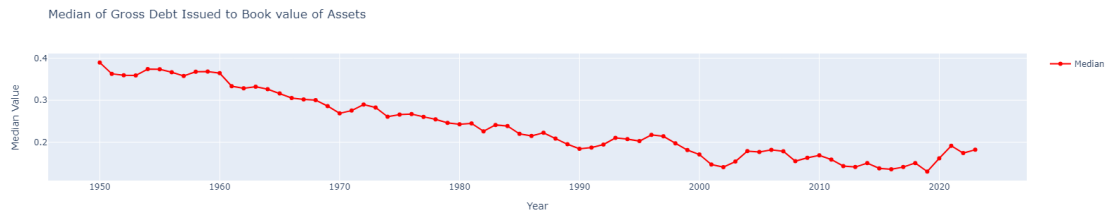


[151]:
```
def mean_median_of_column_by_year(df, year):
    # Filter the dataframe for the given year
    df_year = df[df['datadate'].dt.year == year]
    # Return the mean and median of the specified column
    df_year['wc_ta'] = (df_year['act'] - df_year['lct']) / df_year['at']
    return df_year['wc_ta'].mean(),  df_year['wc_ta'].median()
```

[152]:
```
from plotly.offline import init_notebook_mode, iplot
import plotly.graph_objs as go

# Activate the notebook mode for Plotly
init_notebook_mode(connected=True)

# Initialize lists to store mean and median values
means = []
medians = []
years = list(range(1950, 2024))

# Compute mean and median for each year
for year in years:
    mean_val, median_val = mean_median_of_column_by_year(compustat_copy, year)
    means.append(mean_val)
    medians.append(median_val)

# Create trace for the mean
mean_trace = go.Scatter(
    x=years,
    y=means,
    mode='lines+markers',
    name='Mean',
    marker=dict(color='blue'),
    line=dict(shape='linear')
)

# Layout for mean
mean_layout = go.Layout(
    title='Mean of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Mean Value'),
    showlegend=True
)

# Figure for mean
mean_fig = go.Figure(data=[mean_trace], layout=mean_layout)

# Plot the mean within the notebook
iplot(mean_fig)
```



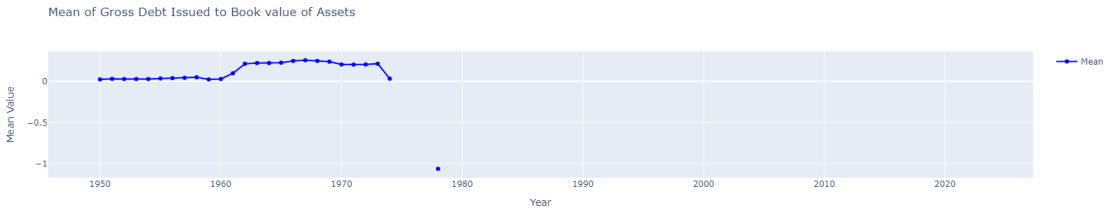
[153]:
```
# Create trace for the median
median_trace = go.Scatter(
    x=years,
    y=medians,
    mode='lines+markers',
    name='Median',
    marker=dict(color='red'),
    line=dict(shape='linear')
)

# Layout for median
median_layout = go.Layout(
    title='Median of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Median Value'),
    showlegend=True
)

# Figure for median
median_fig = go.Figure(data=[median_trace], layout=median_layout)

# Plot the median within the notebook
iplot(median_fig)
```



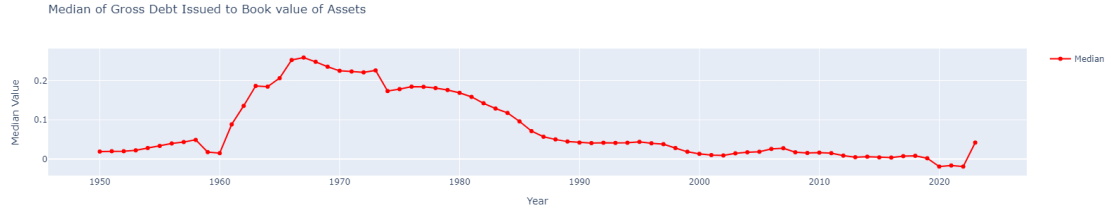
[154]:
```
def mean_median_of_column_by_year(df, year):
    # Filter the dataframe for the given year
```

```
    df_year = df[df['datadate'].dt.year == year]
    # Return the mean and median of the specified column
    column_name_1, column_name_2 = 're', 'at'
    return (df_year[column_name_1] / df_year[column_name_2]).mean(),␣
 ↪(df_year[column_name_1] / df_year[column_name_2]).median()
```

[155]:
```
from plotly.offline import init_notebook_mode, iplot
import plotly.graph_objs as go

# Activate the notebook mode for Plotly
init_notebook_mode(connected=True)

# Initialize lists to store mean and median values
means = []
medians = []
years = list(range(1950, 2024))

# Compute mean and median for each year
for year in years:
    mean_val, median_val = mean_median_of_column_by_year(compustat_copy, year)
    means.append(mean_val)
    medians.append(median_val)

# Create trace for the mean
mean_trace = go.Scatter(
    x=years,
    y=means,
    mode='lines+markers',
    name='Mean',
    marker=dict(color='blue'),
    line=dict(shape='linear')
)

# Layout for mean
mean_layout = go.Layout(
    title='Mean of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Mean Value'),
    showlegend=True
)

# Figure for mean
mean_fig = go.Figure(data=[mean_trace], layout=mean_layout)

# Plot the mean within the notebook
iplot(mean_fig)
```



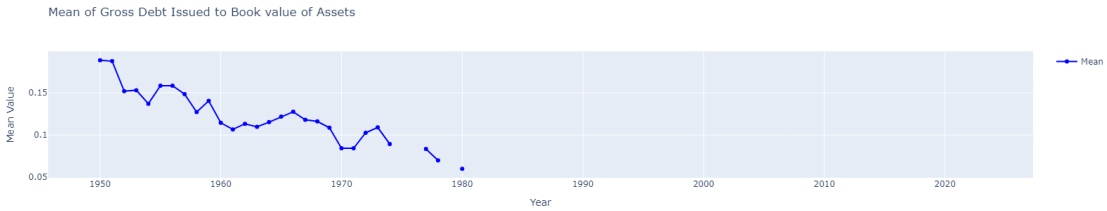
[156]:
```
# Create trace for the median
median_trace = go.Scatter(
    x=years,
    y=medians,
    mode='lines+markers',
    name='Median',
    marker=dict(color='red'),
    line=dict(shape='linear')
)

# Layout for median
median_layout = go.Layout(
    title='Median of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Median Value'),
    showlegend=True
)

# Figure for median
median_fig = go.Figure(data=[median_trace], layout=median_layout)

# Plot the median within the notebook
iplot(median_fig)
```



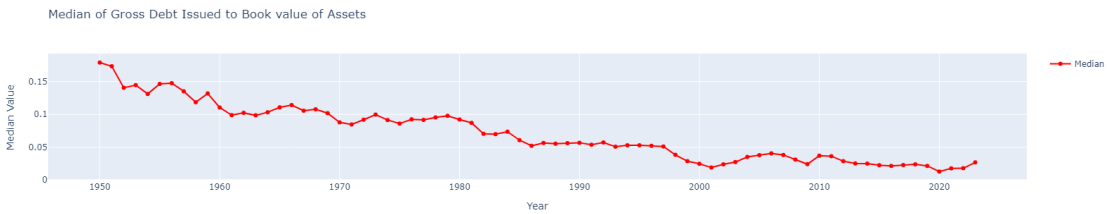
[157]:
```
def mean_median_of_column_by_year(df, year):
    # Filter the dataframe for the given year
```

```
    df_year = df[df['datadate'].dt.year == year]
    # Return the mean and median of the specified column
    column_name_1, column_name_2 = 'oiadp', 'at'
    return (df_year[column_name_1] / df_year[column_name_2]).mean(),␣
 ↪(df_year[column_name_1] / df_year[column_name_2]).median()
```

[158]:
```
from plotly.offline import init_notebook_mode, iplot
import plotly.graph_objs as go

# Activate the notebook mode for Plotly
init_notebook_mode(connected=True)

# Initialize lists to store mean and median values
means = []
medians = []
years = list(range(1950, 2024))

# Compute mean and median for each year
for year in years:
    mean_val, median_val = mean_median_of_column_by_year(compustat_copy, year)
    means.append(mean_val)
    medians.append(median_val)

# Create trace for the mean
mean_trace = go.Scatter(
    x=years,
    y=means,
    mode='lines+markers',
    name='Mean',
    marker=dict(color='blue'),
    line=dict(shape='linear')
)

# Layout for mean
mean_layout = go.Layout(
    title='Mean of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Mean Value'),
    showlegend=True
)

# Figure for mean
mean_fig = go.Figure(data=[mean_trace], layout=mean_layout)

# Plot the mean within the notebook
iplot(mean_fig)
```



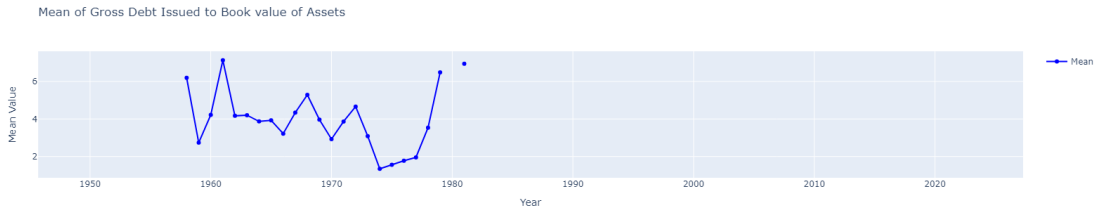
[159]:
```
# Create trace for the median
median_trace = go.Scatter(
    x=years,
    y=medians,
    mode='lines+markers',
    name='Median',
    marker=dict(color='red'),
    line=dict(shape='linear')
)

# Layout for median
median_layout = go.Layout(
    title='Median of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Median Value'),
    showlegend=True
)

# Figure for median
median_fig = go.Figure(data=[median_trace], layout=median_layout)

# Plot the median within the notebook
iplot(median_fig)
```



[160]:
```
def mean_median_of_column_by_year(df, year):
    # Filter the dataframe for the given year
```

```
    df_year = df[df['datadate'].dt.year == year]
    # Return the mean and median of the specified column
    df_year['mv_tl'] = (df_year['prcc_f'] * df_year['csho']) / df_year['lt']
    return df_year['mv_tl'].mean(),  df_year['mv_tl'].median()
```

[161]:
```
from plotly.offline import init_notebook_mode, iplot
import plotly.graph_objs as go

# Activate the notebook mode for Plotly
init_notebook_mode(connected=True)

# Initialize lists to store mean and median values
means = []
medians = []
years = list(range(1950, 2024))

# Compute mean and median for each year
for year in years:
    mean_val, median_val = mean_median_of_column_by_year(compustat_copy, year)
    means.append(mean_val)
    medians.append(median_val)

# Create trace for the mean
mean_trace = go.Scatter(
    x=years,
    y=means,
    mode='lines+markers',
    name='Mean',
    marker=dict(color='blue'),
    line=dict(shape='linear')
)

# Layout for mean
mean_layout = go.Layout(
    title='Mean of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Mean Value'),
    showlegend=True
)

# Figure for mean
mean_fig = go.Figure(data=[mean_trace], layout=mean_layout)

# Plot the mean within the notebook
iplot(mean_fig)
```

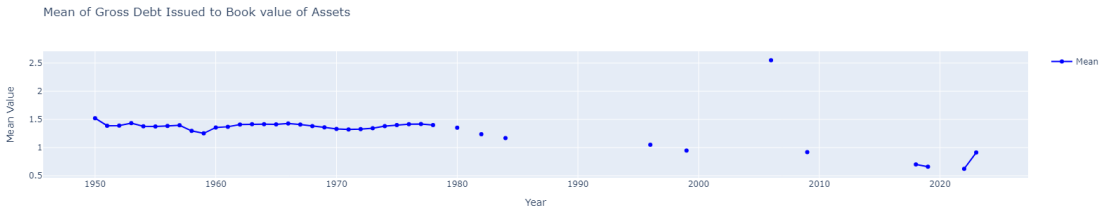


[162]:
```
# Create trace for the median
median_trace = go.Scatter(
    x=years,
    y=medians,
    mode='lines+markers',
    name='Median',
    marker=dict(color='red'),
    line=dict(shape='linear')
)

# Layout for median
median_layout = go.Layout(
    title='Median of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Median Value'),
    showlegend=True
)

# Figure for median
median_fig = go.Figure(data=[median_trace], layout=median_layout)

# Plot the median within the notebook
iplot(median_fig)
```

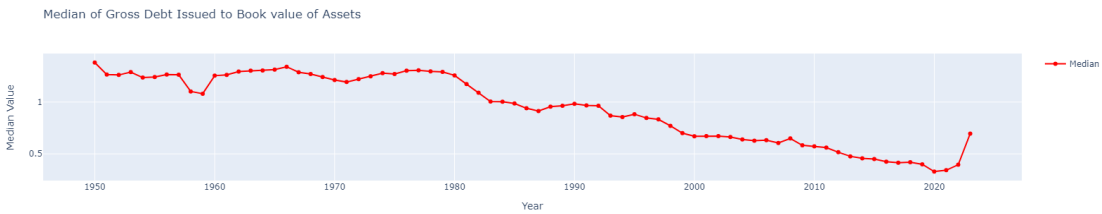
[163]:
```
def mean_median_of_column_by_year(df, year):
    # Filter the dataframe for the given year
```

```
    df_year = df[df['datadate'].dt.year == year]
    # Return the mean and median of the specified column
    column_name_1, column_name_2 = 'sale', 'at'
    return (df_year[column_name_1] / df_year[column_name_2]).mean(),␣
 ↪(df_year[column_name_1] / df_year[column_name_2]).median()
```

[164]:
```
from plotly.offline import init_notebook_mode, iplot
import plotly.graph_objs as go

# Activate the notebook mode for Plotly
init_notebook_mode(connected=True)

# Initialize lists to store mean and median values
means = []
medians = []
years = list(range(1950, 2024))

# Compute mean and median for each year
for year in years:
    mean_val, median_val = mean_median_of_column_by_year(compustat_copy, year)
    means.append(mean_val)
    medians.append(median_val)

# Create trace for the mean
mean_trace = go.Scatter(
    x=years,
    y=means,
    mode='lines+markers',
    name='Mean',
    marker=dict(color='blue'),
    line=dict(shape='linear')
)

# Layout for mean
mean_layout = go.Layout(
    title='Mean of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Mean Value'),
    showlegend=True
)

# Figure for mean
mean_fig = go.Figure(data=[mean_trace], layout=mean_layout)

# Plot the mean within the notebook
iplot(mean_fig)
```



[165]:
```
# Create trace for the median
median_trace = go.Scatter(
    x=years,
    y=medians,
    mode='lines+markers',
    name='Median',
    marker=dict(color='red'),
    line=dict(shape='linear')
)

# Layout for median
median_layout = go.Layout(
    title='Median of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Median Value'),
    showlegend=True
)

# Figure for median
median_fig = go.Figure(data=[median_trace], layout=median_layout)

# Plot the median within the notebook
iplot(median_fig)
```

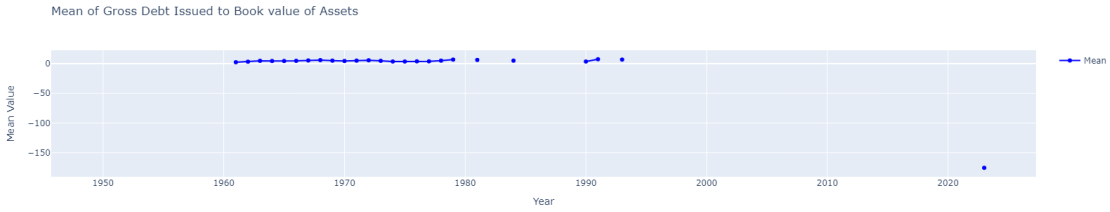


[166]:
```
def mean_median_of_column_by_year(df, year):
    # Filter the dataframe for the given year
```

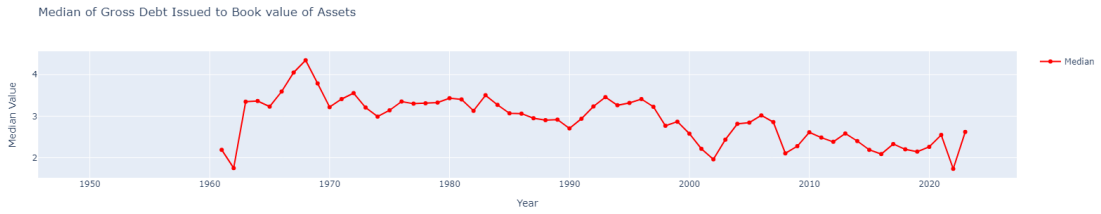
```
    df_year = df[df['datadate'].dt.year == year]
    # Return the mean and median of the specified column
    df_year['wc_ta'] = (df_year['act'] - df_year['lct']) / df_year['at']
    df_year['re_ta'] = df_year['re'] / df_year['at']
    df_year['ebit_ta'] = df_year['oiadp'] / df_year['at']
    df_year['mv_tl'] = (df_year['prcc_f'] * df_year['csho']) / df_year['lt']
    df_year['sales_ta'] = df_year['sale'] / df_year['at']

    # Calculate the Altman Z-Score
    df_year['z_score'] = (1.2 * df_year['wc_ta']) + \
                         (1.4 * df_year['re_ta']) + \
                         (3.3 * df_year['ebit_ta']) + \
                         (0.6 * df_year['mv_tl']) + \
                         (0.99 * df_year['sales_ta'])
    return df_year['z_score'].mean(), df_year['z_score'].median()
```

[167]:
```
from plotly.offline import init_notebook_mode, iplot
import plotly.graph_objs as go

# Activate the notebook mode for Plotly
init_notebook_mode(connected=True)

# Initialize lists to store mean and median values
means = []
medians = []
years = list(range(1950, 2024))

# Compute mean and median for each year
for year in years:
    mean_val, median_val = mean_median_of_column_by_year(compustat_copy, year)
    means.append(mean_val)
    medians.append(median_val)

# Create trace for the mean
mean_trace = go.Scatter(
    x=years,
    y=means,
    mode='lines+markers',
    name='Mean',
    marker=dict(color='blue'),
    line=dict(shape='linear')
)

# Layout for mean
mean_layout = go.Layout(
    title='Mean of Gross Debt Issued to Book value of Assets',
```

```
    xaxis=dict(title='Year'),
    yaxis=dict(title='Mean Value'),
    showlegend=True
)

# Figure for mean
mean_fig = go.Figure(data=[mean_trace], layout=mean_layout)

# Plot the mean within the notebook
iplot(mean_fig)
```



[168]:
```
# Create trace for the median
median_trace = go.Scatter(
    x=years,
    y=medians,
    mode='lines+markers',
    name='Median',
    marker=dict(color='red'),
    line=dict(shape='linear')
)

# Layout for median
median_layout = go.Layout(
    title='Median of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Median Value'),
    showlegend=True
)

# Figure for median
median_fig = go.Figure(data=[median_trace], layout=median_layout)

# Plot the median within the notebook
iplot(median_fig)
```

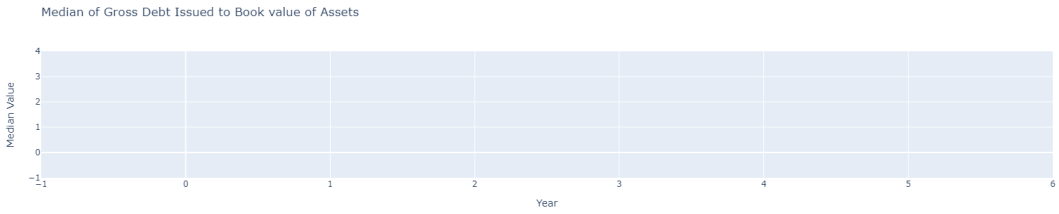


[169]:
```python
def mean_median_of_column_by_year(df, year):
    # Filter the dataframe for the given year
    df_year = df[df['datadate'].dt.year == year]

    # Return the mean and median of the specified column
    # Calculate rolling average for 'at' and 'sale_to_at_avg'
    df_year['at_rolling_avg'] = df_year['at'].rolling(window=2).mean()
    df_year['sale_to_at_avg'] = df_year['sale'] / df_year['at_rolling_avg']
    df_year['sale_to_at_avg'].mean(),df_year['sale_to_at_avg'].median()
```

[170]:
```python
from plotly.offline import init_notebook_mode, iplot
import plotly.graph_objs as go

# Activate the notebook mode for Plotly
init_notebook_mode(connected=True)

# Initialize lists to store mean and median values
means = []
medians = []
years = list(range(1950, 2024))
# for year in years:
#     print(mean_median_of_column_by_year(compustat_copy, year))




# # Compute mean and median for each year
# for year in years:
#     mean_val, median_val = mean_median_of_column_by_year(compustat_copy, year)
#     means.append(mean_val)
#     medians.append(median_val)

# # Create trace for the mean
# mean_trace = go.Scatter(
#     x=years,
#     y=means,
```

```
#     mode='lines+markers',
#     name='Mean',
#     marker=dict(color='blue'),
#     line=dict(shape='linear')
# )

# # Layout for mean
# mean_layout = go.Layout(
#     title='Mean of Gross Debt Issued to Book value of Assets',
#     xaxis=dict(title='Year'),
#     yaxis=dict(title='Mean Value'),
#     showlegend=True
# )

# # Figure for mean
# mean_fig = go.Figure(data=[mean_trace], layout=mean_layout)

# # Plot the mean within the notebook
# iplot(mean_fig)
```

[171]:
```
# Create trace for the median
median_trace = go.Scatter(
    x=years,
    y=medians,
    mode='lines+markers',
    name='Median',
    marker=dict(color='red'),
    line=dict(shape='linear')
)

# Layout for median
median_layout = go.Layout(
    title='Median of Gross Debt Issued to Book value of Assets',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Median Value'),
    showlegend=True
)

# Figure for median
median_fig = go.Figure(data=[median_trace], layout=median_layout)

# Plot the median within the notebook
iplot(median_fig)
```



[172]:
```
compustat_data_copy = compustat_data.copy(deep=True)
```

[173]:
```
compustat_data_copy.columns
```

[173]:
```
Index(['gvkey', 'datadate', 'fyear', 'indfmt', 'consol', 'popsrc', 'datafmt',
       'tic', 'cusip', 'conm',
       …
       'priusa', 'sic', 'spcindcd', 'spcseccd', 'spcsrc', 'state', 'stko',
       'weburl', 'dldte', 'ipodate'],
      dtype='object', length=981)
```

[174]:
```
from pandas.tseries.offsets import MonthEnd
def lag_data(compustat_data):
    # Ensure that 'datadate' is a datetime
    compustat_data['datadate'] = pd.to_datetime(compustat_data['datadate'])

    # Create a new column for the lagged date by adding one month
    compustat_data['lagged_date'] = compustat_data['datadate'] + MonthEnd(1)

    return compustat_data

# Apply the lagging function to your DataFrame
compustat_data_copy = lag_data(compustat_data_copy)
```

[175]:
```
def compute_book_value(compustat_data, start_year=1970, end_year=2022):
    # Ensure that 'datadate' is a datetime
    compustat_data['datadate'] = pd.to_datetime(compustat_data['datadate'])

    # Lag the data by one month
    compustat_data['lagged_date'] = compustat_data['datadate'] + MonthEnd(1)

    # Initialize an empty DataFrame for the results
    book_value_df = pd.DataFrame()

    for year in range(start_year, end_year + 1):
        # Filter for January of the given year
```

```
        january_data = compustat_data[(compustat_data['lagged_date'].dt.year ==␣
↪year) &
                                      (compustat_data['lagged_date'].dt.month␣
↪== 1)]

        # Group by 'gvkey' and calculate the book value per company
        january_data_grouped = january_data.groupby('gvkey').agg({
            'at': 'sum',
            'lt': 'sum'
        }).reset_index()

        january_data_grouped['BookValue'] = january_data_grouped['at'] -␣
↪january_data_grouped['lt']
        january_data_grouped['Year'] = year

        # Append the results to the DataFrame
        book_value_df = pd.concat([book_value_df,␣
↪january_data_grouped[['gvkey', 'Year', 'BookValue']]], ignore_index=True)

    return book_value_df


book_value_results = compute_book_value(compustat_data_copy)
book_value_results.head()
```

[175]:

	gvkey	Year	BookValue
0	1000	1970	10.211
1	1002	1970	6.533
2	1010	1970	192.559
3	1020	1970	24.097
4	1026	1970	19.977

[176]:
```
def compute_avg_cash_flow(compustat_data, start_year=1990, end_year=2022):
    # Convert datadate to datetime and create lagged_date
    compustat_data['datadate'] = pd.to_datetime(compustat_data['datadate'])
    compustat_data['lagged_date'] = compustat_data['datadate'] + MonthEnd(1)

    # Initialize an empty DataFrame for the results
    cash_flow_df = pd.DataFrame()

    for year in range(start_year, end_year + 1):
        # Filter for January of the given year in lagged data
        january_filter = (compustat_data['lagged_date'].dt.year == year) &␣
↪(compustat_data['lagged_date'].dt.month == 1)
        january_data = compustat_data[january_filter]

        # Perform the rolling computation on the filtered data
```

```
        january_data['AvgCashFlow'] = january_data.groupby('gvkey')['oancf'].
↪transform(lambda x: x.rolling(window=60, min_periods=1).mean())

        # Keep only the rows for January of the year
        january_data = january_data[january_filter]

        # Keep only the necessary columns
        january_data = january_data[['gvkey', 'lagged_date', 'AvgCashFlow']]

        # Rename columns and adjust types
        january_data.rename(columns={'lagged_date': 'Year'}, inplace=True)
        january_data['Year'] = january_data['Year'].dt.year

        # Append the results
        cash_flow_df = pd.concat([cash_flow_df, january_data],␣
↪ignore_index=True)

    return cash_flow_df

# Assuming compustat_data_copy is your DataFrame
avg_cash_flow_results = compute_avg_cash_flow(compustat_data_copy)
print(avg_cash_flow_results.head())
```

```
   gvkey  Year  AvgCashFlow
0   1010  1990      100.832
1   1011  1990       -0.525
2   1019  1990        3.623
3   1020  1990       45.182
4   1034  1990        9.456
```

[177]:
```
def compute_avg_revenue(compustat_data, start_year=1970, end_year=2022):
    # Ensure that 'datadate' is a datetime
    compustat_data['datadate'] = pd.to_datetime(compustat_data['datadate'])

    # Lag the data by one month
    compustat_data['lagged_date'] = compustat_data['datadate'] + MonthEnd(1)

    # Initialize an empty DataFrame for the results
    revenue_df = pd.DataFrame(columns=['Year', 'gvkey', 'AvgRevenue'])

    for year in range(start_year, end_year + 1):
        # Filter for January of the given year
        january_data = compustat_data[(compustat_data['lagged_date'].dt.year ==␣
↪year) & (compustat_data['lagged_date'].dt.month == 1)]

        # Group by 'gvkey' and calculate the rolling average of revenue
        # Assuming the revenue column is named 'revt'
```

```
        january_data['AvgRevenue'] = january_data.groupby('gvkey')['sale'].
 ↪transform(lambda x: x.rolling(window=60, min_periods=1).mean())

        # Select only the relevant columns and rename for clarity
        january_results = january_data[['gvkey', 'lagged_date', 'AvgRevenue']].
 ↪rename(columns={'lagged_date': 'Year'})

        # Convert the 'Year' to just the year part
        january_results['Year'] = january_results['Year'].dt.year

        # Append the results to the DataFrame
        revenue_df = pd.concat([revenue_df, january_results], ignore_index=True)

    return revenue_df

avg_revenue_results = compute_avg_revenue(compustat_data_copy)
print(avg_revenue_results.head())
```

```
   Year gvkey  AvgRevenue
0  1970  1000      37.392
1  1970  1002      27.939
2  1970  1010     320.200
3  1970  1020      40.926
4  1970  1026      10.411
```

[178]:
```
# def compute_avg_dividends(compustat_data, start_year=1970, end_year=2022):
#     # Ensure that 'datadate' is a datetime type
#     compustat_data['datadate'] = pd.to_datetime(compustat_data['datadate'])

#     # Lag the data by one month
#     compustat_data['lagged_date'] = compustat_data['datadate'] + MonthEnd(1)

#     # Replace NaN values with zeros in the 'dv' column
#     compustat_data['dv'] = compustat_data['dv'].fillna(0)

#     # Initialize an empty DataFrame for the results
#     dividends_df = pd.DataFrame(columns=['Year', 'gvkey', 'AvgDividends'])

#     # Group by 'gvkey' and calculate the rolling average for each 'gvkey'
#     rolling_dividends = compustat_data.groupby('gvkey')['dv'].
 ↪rolling(window=60, min_periods=1).mean().reset_index()

#     # Merge the rolling average with the original data to align with the␣
 ↪'lagged_date'
#     merged_data = pd.merge(compustat_data, rolling_dividends, on=['gvkey',␣
 ↪'dv'], how='left')
```

```
#     for year in range(start_year, end_year + 1):
#         # Filter for January of the given year
#         january_data = merged_data[(merged_data['lagged_date'].dt.year ==␣
 ↪year) &
#                                    (merged_data['lagged_date'].dt.month == 1)]

#         # Select only the relevant columns and rename for clarity
#         january_data = january_data[['gvkey', 'lagged_date', 'dv_y']].
 ↪rename(columns={'lagged_date': 'Year', 'dv_y': 'AvgDividends'})

#         # Convert the 'Year' to just the year part
#         january_data['Year'] = january_data['Year'].dt.year

#         # Append the results to the DataFrame
#         dividends_df = pd.concat([dividends_df, january_data],␣
 ↪ignore_index=True)

#     return dividends_df

# # Assuming compustat_data_copy is your DataFrame
# avg_dividends_results = compute_avg_dividends(compustat_data_copy)
# print(avg_dividends_results.head())
```

[179]:
```
# def compute_avg_gross_investment(compustat_data, start_year=1970,␣
 ↪end_year=2022):
#     # Ensure that 'datadate' is a datetime type
#     compustat_data['datadate'] = pd.to_datetime(compustat_data['datadate'])

#     # Lag the data by one month
#     compustat_data['lagged_date'] = compustat_data['datadate'] + MonthEnd(1)

#     # Initialize an empty DataFrame for the results
#     investments_df = pd.DataFrame(columns=['Year', 'gvkey',␣
 ↪'AvgGrossInvestment'])

#     # Group by 'gvkey' and calculate the rolling average for each 'gvkey'
#     rolling_investments = compustat_data.groupby('gvkey')['capx'].
 ↪rolling(window=60, min_periods=1).mean().reset_index()

#     # Merge the rolling average with the original data to align with the␣
 ↪'lagged_date'
#     merged_data = pd.merge(compustat_data, rolling_investments, on=['gvkey',␣
 ↪'capx'], how='left')

#     for year in range(start_year, end_year + 1):
#         # Filter for January of the given year
```

```
#         january_data = merged_data[(merged_data['lagged_date'].dt.year ==␣
 ↪year) &
#                                    (merged_data['lagged_date'].dt.month == 1)]

#         # Select only the relevant columns and rename for clarity
#         january_data = january_data[['gvkey', 'lagged_date', 'capx_y']].
 ↪rename(columns={'lagged_date': 'Year', 'capx_y': 'AvgGrossInvestment'})

#         # Convert the 'Year' to just the year part
#         january_data['Year'] = january_data['Year'].dt.year

#         # Append the results to the DataFrame
#         investments_df = pd.concat([investments_df, january_data],␣
 ↪ignore_index=True)

#     # Backfill NaN values
#     investments_df['AvgGrossInvestment'] =␣
 ↪investments_df['AvgGrossInvestment'].fillna(method='bfill')

#     return investments_df

# # Assuming compustat_data_copy is your DataFrame
# avg_gross_investment_results =␣
 ↪compute_avg_gross_investment(compustat_data_copy)
# print(avg_gross_investment_results.head())
```

[180]:
```
def compute_avg_gross_investment(compustat_data, start_year=1970,␣
 ↪end_year=2022):
    # Ensure that 'datadate' is a datetime type
    compustat_data['datadate'] = pd.to_datetime(compustat_data['datadate'])

    # Initialize an empty DataFrame for the results
    investments_df = pd.DataFrame(columns=['Year', 'gvkey',␣
 ↪'AvgGrossInvestment'])

    for year in range(start_year, end_year + 1):
        # Filter data up to the end of the previous year
        yearly_data = compustat_data[compustat_data['datadate'].dt.year < year]

        # Group by 'gvkey' and calculate the rolling average for each 'gvkey'␣
 ↪up to that year
        rolling_investments = yearly_data.groupby('gvkey')['capx'].
 ↪rolling(window=60, min_periods=1).mean().astype('float32').reset_index()

        # Get the last entry for each 'gvkey' which corresponds to the value at␣
 ↪the beginning of the year
```

```
        last_entries = rolling_investments.groupby('gvkey').nth(-1)

        # Add the year and gvkey to the results DataFrame
        last_entries['Year'] = year
        investments_df = pd.concat([investments_df, last_entries[['gvkey',␣
↪'Year', 'capx']]], ignore_index=True)

    # Backfill NaN values
    investments_df['AvgGrossInvestment'] = investments_df['capx'].
↪fillna(method='bfill').astype('float32')
    investments_df.drop(columns='capx', inplace=True)

    return investments_df

avg_gross_investment_results = compute_avg_gross_investment(compustat_data_copy)
print(avg_gross_investment_results.head())
```

```
   Year gvkey  AvgGrossInvestment
0  1970  1000            1.913500
1  1970  1002            0.360000
2  1970  1004            0.313250
3  1970  1010           19.235001
4  1970  1017            1.407000
```

[181]:
```
def compute_avg_roa(compustat_data, start_year=1970, end_year=2022):
    # Ensure that 'datadate' is a datetime type and backfill 'ni' and 'at'
    compustat_data['datadate'] = pd.to_datetime(compustat_data['datadate'])
    compustat_data['ni'] = compustat_data['ni'].fillna(method='bfill').fillna(0)
    compustat_data['at'] = compustat_data['at'].fillna(method='bfill').fillna(0)

    # Lag the data by one month
    compustat_data['lagged_date'] = compustat_data['datadate'] + MonthEnd(1)

    # Calculate ROA, avoiding division by zero
    compustat_data['ROA'] = compustat_data.apply(lambda x: x['ni'] / x['at'] if␣
↪x['at'] != 0 else 0, axis=1)

    # Calculate the 60-month rolling average of ROA for each 'gvkey'
    compustat_data['rolling_avg_ROA'] = compustat_data.groupby('gvkey')['ROA'].
↪transform(
        lambda x: x.rolling(window=60, min_periods=1).mean()
    )

    # Initialize an empty list to store the results
    results = []

    # Loop over each year and compute the average ROA for January
```

```
    for year in range(start_year, end_year + 1):
        january_data = compustat_data[
            (compustat_data['lagged_date'].dt.year == year) &␣
↪(compustat_data['lagged_date'].dt.month == 1)
        ]
        january_data = january_data[['gvkey', 'lagged_date',␣
↪'rolling_avg_ROA']].copy()
        january_data.rename(columns={'lagged_date': 'Year', 'rolling_avg_ROA':␣
↪'AvgROA'}, inplace=True)
        january_data['Year'] = january_data['Year'].dt.year
        results.append(january_data)

    # Concatenate the list of DataFrames into one DataFrame
    roa_df = pd.concat(results, ignore_index=True)

    return roa_df

avg_roa_results = compute_avg_roa(compustat_data_copy)
print(avg_roa_results.head())
```

```
   gvkey  Year    AvgROA
0   1000  1970  0.014259
1   1002  1970  0.015641
2   1010  1970  0.044885
3   1020  1970 -0.006601
4   1026  1970  0.014096
```

[182]:
```
# Function to calculate Asset Turnover for each year on January 1st
def calculate_asset_turnover(compustat_data):
    # Convert 'datadate' to a datetime type and lag it by one month
    compustat_data['datadate'] = pd.to_datetime(compustat_data['datadate'])
    compustat_data['lagged_date'] = compustat_data['datadate'] + MonthEnd(1)

    # Calculate the Asset Turnover Ratio
    # Here we assume that 'sale' for total sales is already present in the data
    # We use 'at' from the previous year as it is lagged to the start of the␣
↪current year
    compustat_data['AssetTurnover'] = compustat_data['sale'] /␣
↪compustat_data['at']

    # Filter the data for January of each year to get the Asset Turnover at the␣
↪beginning of each year
    january_data = compustat_data[compustat_data['lagged_date'].dt.month == 1]
    january_data = january_data[['gvkey', 'lagged_date', 'AssetTurnover']].
↪copy()
    january_data.rename(columns={'lagged_date': 'Year'}, inplace=True)
    january_data['Year'] = january_data['Year'].dt.year
```

```
    return january_data


asset_turnover = calculate_asset_turnover(compustat_data_copy)
asset_turnover.head()
```

[182]:

	gvkey	Year	AssetTurnover
0	1000	1962	0.635593
1	1000	1963	1.129944
2	1000	1964	1.028955
3	1000	1965	1.435028
4	1000	1966	0.730736

[183]:
```
def compute_altman_z_score(compustat_data):
    # Ensure that 'datadate' is a datetime type
    compustat_data['datadate'] = pd.to_datetime(compustat_data['datadate'])

    # Lag the data by one month to use the end-of-year data as the start of␣
 ↪January next year
    compustat_data['lagged_date'] = compustat_data['datadate'] + MonthEnd(1)

    # Calculate the individual components of the Z-Score
    compustat_data['wc_ta'] = (compustat_data['act'] - compustat_data['lct']) /␣
 ↪compustat_data['at']
    compustat_data['re_ta'] = compustat_data['re'] / compustat_data['at']
    compustat_data['ebit_ta'] = compustat_data['oiadp'] / compustat_data['at']
    compustat_data['mv_tl'] = (compustat_data['prcc_f'] *␣
 ↪compustat_data['csho']) / compustat_data['lt']
    compustat_data['sales_ta'] = compustat_data['sale'] / compustat_data['at']

    # Handle potential division by zero or NA values
    for column in ['wc_ta', 're_ta', 'ebit_ta', 'mv_tl', 'sales_ta']:
        compustat_data[column].replace([np.inf, -np.inf], np.nan, inplace=True)
        compustat_data[column].fillna(0, inplace=True)

    # Calculate the Altman Z-Score
    compustat_data['z_score'] = (1.2 * compustat_data['wc_ta']) + \
                                (1.4 * compustat_data['re_ta']) + \
                                (3.3 * compustat_data['ebit_ta']) + \
                                (0.6 * compustat_data['mv_tl']) + \
                                (0.99 * compustat_data['sales_ta'])

    # Filter for January of each year to get the beginning-of-year values
    compustat_data_jan = compustat_data[compustat_data['lagged_date'].dt.month␣
 ↪== 1]
```

```
    # Select only the relevant columns
    compustat_data_jan = compustat_data_jan[['gvkey', 'lagged_date', 'z_score']]

    # Rename columns for clarity
    compustat_data_jan.rename(columns={'lagged_date': 'Year', 'z_score':␣
 ↪'AltmanZ'}, inplace=True)

    # Convert the 'Year' to just the year part
    compustat_data_jan['Year'] = compustat_data_jan['Year'].dt.year

    return compustat_data_jan

# Assuming compustat_data is your DataFrame
z_score_results = compute_altman_z_score(compustat_data_copy)
print(z_score_results.head())
```

```
   gvkey  Year   AltmanZ
0   1000  1962  0.629237
1   1000  1963  1.118644
2   1000  1964  1.142959
3   1000  1965  1.976328
4   1000  1966  0.311913
```

[184]:
```
def compute_ohlson_o_score(compustat_data, start_year=1970, end_year=2022):
    # Ensure 'datadate' is a datetime type and sort the DataFrame
    compustat_data['datadate'] = pd.to_datetime(compustat_data['datadate'])
    compustat_data.sort_values(by=['gvkey', 'datadate'], inplace=True)

    # Backfill or fill NA values for necessary columns
    fill_columns = ['lt', 'at', 'act', 'lct', 'ni', 'dp', 'prcc_f', 'csho']
    for col in fill_columns:
        compustat_data[col] = compustat_data[col].fillna(method='bfill').
 ↪fillna(0)

    # Lag the data by one month to use the end of December data as the start of␣
 ↪January next year
    compustat_data['lagged_date'] = compustat_data['datadate'] + MonthEnd(1)

    # Calculate the necessary components for the Ohlson O-score
    compustat_data['TLTA'] = compustat_data['lt'] / compustat_data['at']
    compustat_data['WCTA'] = (compustat_data['act'] - compustat_data['lct']) /␣
 ↪compustat_data['at']
    compustat_data['CLCA'] = compustat_data['lct'] / compustat_data['act']
    compustat_data['OENEG'] = (compustat_data['lt'] > compustat_data['at']).
 ↪astype(int)
    compustat_data['NITA'] = compustat_data['ni'] / compustat_data['at']
```

```
    compustat_data['FUTL'] = (compustat_data['ni'] + compustat_data['dp']) /␣
↪compustat_data['lt']
    compustat_data['SIZE'] = np.log(compustat_data['at'])
    compustat_data['MVE_BVTD'] = (compustat_data['prcc_f'] *␣
↪compustat_data['csho']) / compustat_data['lt']

    # Calculate the INTWO and CHIN components with a rolling window to check␣
↪for negative net income in the past two years
    compustat_data['INTWO'] = compustat_data.groupby('gvkey')['ni'].
↪rolling(window=2).apply(lambda x: np.any(x < 0), raw=False).
↪reset_index(level=0, drop=True).fillna(method='bfill').fillna(0).astype(int)
    compustat_data['CHIN'] = compustat_data.groupby('gvkey')['ni'].diff() /␣
↪compustat_data.groupby('gvkey')['at'].shift()
    compustat_data['CHIN'] = compustat_data['CHIN'].fillna(method='bfill').
↪fillna(0)

    # Calculate the O-score
    compustat_data['O_score'] = -1.32 - (0.407 * compustat_data['SIZE']) + (6.
↪03 * compustat_data['TLTA']) - \
                                (1.43 * compustat_data['WCTA']) + (0.0757 *␣
↪compustat_data['CLCA']) - \
                                (1.72 * compustat_data['OENEG']) - (2.37 *␣
↪compustat_data['NITA']) - \
                                (1.83 * compustat_data['FUTL']) + (0.285 *␣
↪compustat_data['INTWO']) - \
                                (0.521 * compustat_data['CHIN']) + (0.593 *␣
↪compustat_data['CHIN']) + \
                                (0.717 * compustat_data['MVE_BVTD']) + (0.107␣
↪* compustat_data['MVE_BVTD'])

    # Initialize an empty DataFrame for the results
    ohlson_o_df = pd.DataFrame(columns=['Year', 'gvkey', 'OhlsonO'])

    for year in range(start_year, end_year + 1):
        # Filter for January of the given year
        january_data = compustat_data[(compustat_data['lagged_date'].dt.year ==␣
↪year) &
                                      (compustat_data['lagged_date'].dt.month␣
↪== 1)]

        # Select only the relevant columns and rename for clarity
        january_data = january_data[['gvkey', 'lagged_date', 'O_score']].
↪rename(columns={'lagged_date': 'Year', 'O_score': 'OhlsonO'})

        # Convert the 'Year' to just the year part
        january_data['Year'] = january_data['Year'].dt.year
```

```
        # Append the results to the DataFrame
        ohlson_o_df = pd.concat([ohlson_o_df, january_data], ignore_index=True)

    return ohlson_o_df

# Assuming compustat_data_copy is your DataFrame
ohlson_o_score_results = compute_ohlson_o_score(compustat_data_copy)
print(ohlson_o_score_results.head())
```

```
   Year gvkey   OhlsonO
0  1970  1000  1.837350
1  1970  1002  0.831973
2  1970  1010 -0.161311
3  1970  1020  1.038929
4  1970  1026  1.732989
```

[185]:
```
msf_master = pd.read_csv('msf_master.csv',low_memory=False)
```

[186]:
```
msf_master.head()
```

[186]:

	PERMNO	date	PRC	RET	SHROUT	vwretd
0	10000	1985-12-31	NaN	NaN	NaN	0.043061
1	10000	1986-01-31	-4.3750	C	3680.0	0.009830
2	10000	1986-02-28	-3.2500	-0.257143	3680.0	0.072501
3	10000	1986-03-31	-4.4375	0.365385	3680.0	0.053887
4	10000	1986-04-30	-4.0000	-0.098592	3793.0	-0.007903

[187]:
```
msf_master.dtypes
```

[187]:
```
PERMNO      int64
date       object
PRC       float64
RET        object
SHROUT    float64
vwretd    float64
dtype: object
```

[188]:
```
msf_master.isna().sum()
```

[188]:
```
PERMNO         0
date           0
PRC       142068
RET        78605
SHROUT     36212
vwretd         0
dtype: int64
```

[189]:
```
msf_master['date'] = pd.to_datetime(msf_master['date'],errors='coerce')
msf_master['PRC'] = pd.to_numeric(msf_master['PRC'],errors='coerce')
msf_master['PRC'] = msf_master['PRC'].abs()
msf_master['PRC'] = msf_master['PRC'].fillna(method='bfill')
```

[190]:
```
msf_master.isna().sum()
```

[190]:
```
PERMNO        0
date          0
PRC           0
RET       78605
SHROUT    36212
vwretd        0
dtype: int64
```

[191]:
```
msf_master.dtypes
```

[191]:
```
PERMNO             int64
date      datetime64[ns]
PRC              float64
RET               object
SHROUT           float64
vwretd           float64
dtype: object
```

[192]:
```
msf_master['RET'] = msf_master['RET'].fillna(method='bfill')
msf_master['RET'] = pd.to_numeric(msf_master['RET'],errors='coerce')
msf_master['SHROUT'] = msf_master['SHROUT'].fillna(method='bfill')
```

[193]:
```
msf_master.isna().sum()
```

[193]:
```
PERMNO         0
date           0
PRC            0
RET       170750
SHROUT         0
vwretd         0
dtype: int64
```

[194]:
```
msf_master.dtypes
```

[194]:
```
PERMNO             int64
date      datetime64[ns]
PRC              float64
RET              float64
SHROUT           float64
vwretd           float64
dtype: object
```

[ ]:

[195]:
```
def calculate_market_measures(df):
    # Create a new DataFrame to hold market measures
    market_measures = pd.DataFrame()

    # Define a function to calculate Beta
    def calculate_beta(x):
        if len(x) > 1:
            cov_matrix = np.cov(x['RET'], x['vwretd'], ddof=1)
            cov_ret_vwretd = cov_matrix[0, 1]
            var_vwretd = cov_matrix[1, 1]
            if var_vwretd != 0:
                return cov_ret_vwretd / var_vwretd
        return np.nan

    # Calculate Beta for each stock and year
    market_measures['Beta'] = df.groupby(['PERMNO', df['date'].dt.year]).
 ↪apply(calculate_beta)

    # Calculate Total Volatility for each stock and year
    market_measures['Total_Volatility'] = df.groupby(['PERMNO', df['date'].dt.
 ↪year]) \
                                            .apply(lambda x: np.std(x['RET'],␣
 ↪ddof=1) if len(x) > 1 else np.nan)

    # Calculate Annualized Volatility
    market_measures['Annualized_Volatility'] =␣
 ↪market_measures['Total_Volatility'] * np.sqrt(12) * 100

    # Calculate Annualized Volatility assuming zero average return
    market_measures['Annualized_Volatility_Zero_Avg'] = df.groupby(['PERMNO',␣
 ↪df['date'].dt.year]) \
                                                        .apply(lambda x: np.
 ↪sqrt(np.mean(x['RET']**2)) * np.sqrt(12) * 100 if len(x) > 1 else np.nan)

    # Reset index to bring 'PERMNO' and 'date' back as columns
    market_measures = market_measures.reset_index()

    # Rename 'date' column to 'Year'
    market_measures.rename(columns={'date': 'Year'}, inplace=True)

    return market_measures

# Assuming msf_master is the DataFrame with the cleaned data
market_measures_df = calculate_market_measures(msf_master)
market_measures_df.head()
```

[195]:
```
   PERMNO  Year  Beta  Total_Volatility  Annualized_Volatility  \
0   10000  1985   NaN               NaN                    NaN
1   10000  1986   NaN          0.252801              87.572846
2   10000  1987   NaN          0.154829              53.634240
3   10001  1985   NaN               NaN                    NaN
4   10001  1986   NaN          0.028254               9.787424

   Annualized_Volatility_Zero_Avg
0                             NaN
1                       96.420243
2                       69.502476
3                             NaN
4                       11.299923
```

[196]:
```
msf_data = pd.read_csv("msf.csv", low_memory=False)
```

[197]:
```
msf_data.head()
```

[197]:
```
   PERMNO        date        RET    vwretd
0   10000  1985-12-31        NaN  0.043061
1   10000  1986-01-31          C  0.009830
2   10000  1986-02-28  -0.257143  0.072501
3   10000  1986-03-31   0.365385  0.053887
4   10000  1986-04-30  -0.098592 -0.007903
```

[198]:
```
msf_data.dtypes
```

[198]:
```
PERMNO      int64
date       object
RET        object
vwretd    float64
dtype: object
```

[199]:
```
msf_data['date'] = pd.to_datetime(msf_data['date'],errors='coerce')
msf_data['RET'] = pd.to_numeric(msf_data['RET'],errors='coerce')
msf_data['RET'] = msf_data['RET'].fillna(method='bfill')
```

[200]:
```
msf_data.dtypes
```

[200]:
```
PERMNO             int64
date      datetime64[ns]
RET              float64
vwretd           float64
dtype: object
```

[201]:
```
msf_data.isna().sum()
```

[201]:
```
PERMNO    0
date      0
RET       0
vwretd    0
dtype: int64
```

[202]:
```
msf_data.shape
```

[202]:
```
(4347108, 4)
```

[203]:
```
rf = pd.read_csv("F-F_Research_Data_Factors.CSV", skiprows=3)
```

[204]:
```
rf.head()
```

[204]:

	Unnamed: 0	Mkt-RF	SMB	HML	RF
0	192607	2.96	-2.56	-2.43	0.22
1	192608	2.64	-1.17	3.82	0.25
2	192609	0.36	-1.40	0.13	0.23
3	192610	-3.24	-0.09	0.70	0.32
4	192611	2.53	-0.10	-0.51	0.31

[205]:
```
rf = rf.rename(columns = {'Unnamed: 0':"date"})
```

[206]:
```
rf.head(1165)
```

[206]:

	date	Mkt-RF	SMB	HML	RF
0	192607	2.96	-2.56	-2.43	0.22
1	192608	2.64	-1.17	3.82	0.25
2	192609	0.36	-1.40	0.13	0.23
3	192610	-3.24	-0.09	0.70	0.32
4	192611	2.53	-0.10	-0.51	0.31
…	…	…	…	…	…
1160	202303	2.51	-5.51	-8.85	0.36
1161	202304	0.61	-3.35	-0.04	0.35
1162	202305	0.35	1.61	-7.72	0.36
1163	202306	6.46	1.54	-0.26	0.40
1164	202307	3.21	2.08	4.11	0.45

[1165 rows x 5 columns]

[207]:
```
rf = rf.iloc[:1165]
```

[208]:
```
rf.head()
```

[208]:

	date	Mkt-RF	SMB	HML	RF
0	192607	2.96	-2.56	-2.43	0.22
1	192608	2.64	-1.17	3.82	0.25
2	192609	0.36	-1.40	0.13	0.23

```
3  192610     -3.24     -0.09      0.70      0.32
4  192611      2.53     -0.10     -0.51      0.31
```

[209]:
```
rf['date'] = pd.to_datetime(rf['date'], format='%Y%m')
rf['RF'] = pd.to_numeric(rf['date'], errors='coerce')
```

[210]:
```
rf.head()
```

[210]:
```
        date    Mkt-RF       SMB       HML                    RF
0 1926-07-01      2.96     -2.56     -2.43 -1372896000000000000
1 1926-08-01      2.64     -1.17      3.82 -1370217600000000000
2 1926-09-01      0.36     -1.40      0.13 -1367539200000000000
3 1926-10-01     -3.24     -0.09      0.70 -1364947200000000000
4 1926-11-01      2.53     -0.10     -0.51 -1362268800000000000
```

[211]:
```
msf_data.columns
```

[211]:
```
Index(['PERMNO', 'date', 'RET', 'vwretd'], dtype='object')
```

[212]:
```
rf.dtypes
```

[212]:
```
date      datetime64[ns]
Mkt-RF            object
SMB               object
HML               object
RF                 int64
dtype: object
```

[213]:
```
import pandas as pd

# Copy the msf_data into a new DataFrame
msf_data_lagged = msf_data.copy(deep=True)

# Ensure that the 'date' columns in both DataFrames are in datetime format
msf_data_lagged['date'] = pd.to_datetime(msf_data_lagged['date'],␣
 ↪format='%Y-%m-%d')
rf['date'] = pd.to_datetime(rf['date'], format='%Y-%m-%d')

# Sort the DataFrame by 'PERMNO' and 'date' to ensure correct lagging
msf_data_lagged.sort_values(by=['PERMNO', 'date'], inplace=True)

# Lag the 'RET' and 'vwretd' columns by one month within each 'PERMNO' group
msf_data_lagged['RET_lagged'] = msf_data_lagged.groupby('PERMNO')['RET'].
 ↪shift(1)
msf_data_lagged['vwretd_lagged'] = msf_data_lagged.groupby('PERMNO')['vwretd'].
 ↪shift(1)
```

```
# Now sort msf_data_lagged by 'date' to prepare for the asof merge
msf_data_lagged.sort_values(by='date', inplace=True)

# Also sort the rf DataFrame by 'date'
rf_sorted = rf.sort_values(by='date')

# Perform the asof merge
merged_data = pd.merge_asof(msf_data_lagged, rf_sorted, on='date')
```

[214]:
```
merged_data.dtypes
```

[214]:
```
PERMNO                    int64
date              datetime64[ns]
RET                      float64
vwretd                   float64
RET_lagged               float64
vwretd_lagged            float64
Mkt-RF                    object
SMB                       object
HML                       object
RF                         int64
dtype: object
```

[215]:
```
# Calculate market premium
merged_data['MKT_premium'] = merged_data['vwretd_lagged'] - merged_data['RF']
```

[216]:
```
merged_data.isna().sum()
```

[216]:
```
PERMNO              0
date                0
RET                 0
vwretd              0
RET_lagged      35706
vwretd_lagged   35706
Mkt-RF              0
SMB                 0
HML                 0
RF                  0
MKT_premium     35706
dtype: int64
```

[217]:
```
merged_data.shape
```

[217]:
```
(4347108, 11)
```

[218]:
```
cleaned_data = merged_data.dropna(subset=['RET_lagged', 'vwretd_lagged',␣
↪'MKT_premium'])
```

[219]:
```
cleaned_data.head()
```

[219]:

	PERMNO	date	RET	vwretd	RET_lagged	vwretd_lagged	\
2374	18411	1970-01-30	-0.022549	-0.073254	-0.014493	-0.01968	
2375	31691	1970-01-30	-0.168675	-0.073254	-0.267647	-0.01968	
2376	47458	1970-01-30	0.032374	-0.073254	-0.191860	-0.01968	
2377	37970	1970-01-30	0.006494	-0.073254	-0.174263	-0.01968	
2378	48231	1970-01-30	-0.342105	-0.073254	-0.173913	-0.01968	

	Mkt-RF	SMB	HML	RF	MKT_premium
2374	-8.10	2.93	3.13	0	-0.01968
2375	-8.10	2.93	3.13	0	-0.01968
2376	-8.10	2.93	3.13	0	-0.01968
2377	-8.10	2.93	3.13	0	-0.01968
2378	-8.10	2.93	3.13	0	-0.01968

[220]:
```
from concurrent.futures import ThreadPoolExecutor
from sklearn.linear_model import LinearRegression
import pandas as pd
import numpy as np
import concurrent.futures  # This import is necessary
from tqdm import tqdm

# Define the regression function using sklearn
def run_regression_sklearn(group_data):
    permno = group_data['PERMNO'].iloc[0]  # Get PERMNO
    year = group_data['date'].dt.year.iloc[0]  # Get Year

    y = group_data['RET_lagged'].values.reshape(-1, 1).astype(float)
    X = group_data['MKT_premium'].values.reshape(-1, 1).astype(float)

    # Create and fit the model
    model = LinearRegression()
    model.fit(X, y)

    # Calculate residuals as y - y_pred
    y_pred = model.predict(X)
    residuals = y - y_pred

    # Return results including PERMNO and year
    return {'PERMNO': permno, 'Year': year, 'Alpha': model.intercept_[0],␣
 ↪'Beta': model.coef_[0][0], 'Residuals': residuals.flatten()}

# Prepare the data by grouping by PERMNO and year
grouped_data = cleaned_data.groupby(['PERMNO', cleaned_data['date'].dt.year])

# Initialize a list to store futures
```

```
futures_list = []

# Set up thread pool and run regressions
with ThreadPoolExecutor(max_workers=8) as executor:
    # Submit tasks to the executor
    for name, group in grouped_data:
        futures = executor.submit(run_regression_sklearn, group)
        futures_list.append(futures)

# Use tqdm to show progress
results = [future.result() for future in tqdm(concurrent.futures.
 ↪as_completed(futures_list), total=len(futures_list))]

# Convert the list of results to a DataFrame
capm_beta = pd.DataFrame(results)

# Check the results
print(f"Completed regressions for {len(capm_beta)} PERMNO-year combinations.")
if len(capm_beta) == 0:
    print("No regressions were successful. Please check the data and try again.
 ↪")

# Optionally, print out a few examples to verify
print(capm_beta.head())
```

```
100%|                                          |
388776/388776 [00:02<00:00, 177490.26it/s]

Completed regressions for 388776 PERMNO-year combinations.
   PERMNO  Year      Alpha          Beta  \
0   20693  1982  -4.121253 -1.057376e-17
1   19749  1984   1.198012  2.663877e-18
2   86839  2009   5.509684  4.380882e-18
3   84772  1997  14.161891  1.614596e-17
4   84262  2004  -1.465308 -1.408744e-18

                                           Residuals
0  [0.008956418426773291, 0.10389566834931546, -0…
1  [-0.2017653523118777, 0.07067157600486441, 0.3…
2  [0.31746012981085253, -0.2454441150788554, -0…
3  [-0.06132288345563627, -0.018077555836757825, …
4  [0.10269605926130237, -0.06992812104009415, 0…
```

[221]:
```
cleaned_data['Mkt-RF'] = pd.to_numeric(cleaned_data['Mkt-RF'], errors='coerce')
cleaned_data['SMB'] = pd.to_numeric(cleaned_data['SMB'], errors='coerce')
cleaned_data['HML'] = pd.to_numeric(cleaned_data['HML'], errors='coerce')

from concurrent.futures import ThreadPoolExecutor
```

```
import statsmodels.api as sm
import pandas as pd
import concurrent.futures
from tqdm import tqdm

# Assuming 'cleaned_data' has been properly prepared and includes 'Mkt-RF',␣
 ↪'SMB', 'HML', and 'RET_lagged' columns.

# Define the regression function for the Fama-French 3-factor model
def run_regression_statsmodels(group_data, permno, year):
    try:
        if group_data.shape[0] < 2:
            return None
        y = group_data['RET_lagged'].astype(float)
        X = group_data[['MKT_premium', 'SMB', 'HML']].astype(float)
        X = sm.add_constant(X)
        model = sm.OLS(y, X)
        results = model.fit()
        return {
            'PERMNO': permno,
            'Year': year,
            'Alpha': results.params['const'],
            'Beta_MKT': results.params['MKT_premium'],
            'Beta_SMB': results.params['SMB'],
            'Beta_HML': results.params['HML'],
            'Residuals': results.resid
        }
    except Exception as e:
        print(f"Error in regression for PERMNO {permno}, Year {year}: {e}")
        return None

# Initialize an empty list for storing regression results
three_factor_results_list = []

# Prepare the data for regressions
grouped_data = cleaned_data.groupby(['PERMNO', cleaned_data['date'].dt.year])

# Perform the regressions in a thread pool
with ThreadPoolExecutor(max_workers=4) as executor:
    futures = [executor.submit(run_regression_statsmodels, group, permno, year)
               for (permno, year), group in grouped_data]

    # Process as they complete
    for future in tqdm(concurrent.futures.as_completed(futures),␣
 ↪total=len(futures)):
        result = future.result()
        if result:
```

```
            # Collect the result in the list
            three_factor_results_list.append(result)

# Convert the list of results to a DataFrame
three_factor_beta = pd.DataFrame(three_factor_results_list)

# Print results for the first few companies
print(three_factor_beta.head())
```

```
100%|                                          |
388776/388776 [20:12<00:00, 320.73it/s]

   PERMNO  Year         Alpha      Beta_MKT      Beta_SMB      Beta_HML  \
0   10547  1992 -7.890229e-38  5.592581e-20 -4.645171e-38 -1.352662e-37
1   10637  1994 -1.806004e-37  1.394009e-19  6.371274e-39  1.743533e-38
2   10347  1989 -8.773337e-38  5.364538e-20  6.195657e-38  2.192311e-38
3   10417  1990 -2.492788e-37  1.609506e-19  3.141504e-37  2.542941e-37
4   10622  2004  0.000000e+00 -2.220632e-20  5.277782e-38  5.153599e-38

                                          Residuals
0  1477986   -0.030942
1487447    0.188975
149553…
1  1655042    0.225580
1666364    0.025596
167422…
2  1228823    0.032167
1237902    0.232310
124401…
3  1314269   -0.065083
1319750    0.802015
133056…
4  2670027    0.071250
2674024   -0.019234
268319…
```

[222]:
```
momentum_data = pd.read_csv('F-F_Momentum_Factor.CSV')
```

[223]:
```
momentum_data.head()
```

[223]:
```
     Date   Mom
0  192701  0.36
1  192702 -2.14
2  192703  3.61
3  192704  4.30
4  192705  3.00
```

[224]:
```
momentum_data.dtypes
```

[224]:
```
Date      int64
Mom     float64
dtype: object
```

[225]:
```
momentum_data['Date'] = pd.to_datetime(momentum_data['Date'],errors='coerce',␣
↪format='%Y%m')
```

[226]:
```
momentum_data.dtypes
```

[226]:
```
Date    datetime64[ns]
Mom            float64
dtype: object
```

[227]:
```
momentum_data.isna().sum()
```

[227]:
```
Date    0
Mom     0
dtype: int64
```

[228]:
```
momentum_data.columns
```

[228]:
```
Index(['Date', 'Mom'], dtype='object')
```

[229]:
```
# Extract year and month from the 'date' column in cleaned_data
cleaned_data['Year'] = cleaned_data['date'].dt.year
cleaned_data['Month'] = cleaned_data['date'].dt.month

# Extract year and month from the 'Date' column in momentum_data
momentum_data['Year'] = momentum_data['Date'].dt.year
momentum_data['Month'] = momentum_data['Date'].dt.month

# Merge the two DataFrames on 'Year', 'Month', and 'PERMNO'
merged_data = pd.merge(
    cleaned_data,
    momentum_data[['Year', 'Month', 'Mom']],  # Only select the necessary␣
↪columns from momentum_data
    on=['Year', 'Month'],
    how='left'
)
```

[230]:
```
merged_data.head()
```

[230]:

	PERMNO	date	RET	vwretd	RET_lagged	vwretd_lagged	Mkt-RF	\
0	18411	1970-01-30	-0.022549	-0.073254	-0.014493	-0.01968	-8.1	
1	31691	1970-01-30	-0.168675	-0.073254	-0.267647	-0.01968	-8.1	
2	47458	1970-01-30	0.032374	-0.073254	-0.191860	-0.01968	-8.1	
3	37970	1970-01-30	0.006494	-0.073254	-0.174263	-0.01968	-8.1	
4	48231	1970-01-30	-0.342105	-0.073254	-0.173913	-0.01968	-8.1	

	SMB	HML	RF	MKT_premium	Year	Month	Mom
0	2.93	3.13	0	-0.01968	1970	1	0.6
1	2.93	3.13	0	-0.01968	1970	1	0.6
2	2.93	3.13	0	-0.01968	1970	1	0.6
3	2.93	3.13	0	-0.01968	1970	1	0.6
4	2.93	3.13	0	-0.01968	1970	1	0.6

[231]:
```
import pandas as pd
import statsmodels.api as sm
from concurrent.futures import ThreadPoolExecutor

# Define the regression function for the four-factor model
def run_four_factor_regression(group_data):
    try:
        if group_data.shape[0] < 2:
            return None
        y = group_data['RET_lagged'].astype(float)
        X = group_data[['MKT_premium', 'SMB', 'HML', 'Mom']].astype(float)
        X = sm.add_constant(X)  # Adds a constant term to the predictor
        model = sm.OLS(y, X)
        results = model.fit()
        return {
            'Year': group_data['date'].dt.year.iloc[0],  # Get the year from␣
 ↪the date
            'Alpha': results.params['const'],
            'Beta_MKT': results.params['MKT_premium'],
            'Beta_SMB': results.params['SMB'],
            'Beta_HML': results.params['HML'],
            'Beta_MOM': results.params['Mom'],
            'Residuals': results.resid
        }
    except Exception as e:
        print(f"Error in regression for group: {e}")
        return None

# Perform the regressions in parallel using ThreadPoolExecutor
def execute_regressions(data):
    with ThreadPoolExecutor(max_workers=8) as executor:
        # Group by both PERMNO and Year
        futures = {executor.submit(run_four_factor_regression, group): (permno,␣
 ↪group['date'].dt.year.iloc[0])
                   for (permno, year), group in data.groupby(['PERMNO',␣
 ↪data['date'].dt.year])}
        results = []
        for future in futures:
            result = future.result()
```

```
            if result:
                result['PERMNO'] = futures[future][0]  # Get PERMNO
                results.append(result)
        return results

# Prepare the data by dropping rows with NaN in the required columns
cleaned_data_for_regression = merged_data.dropna(subset=['RET_lagged',␣
 ↪'MKT_premium', 'SMB', 'HML', 'Mom'])

# Execute the regressions
regression_outcomes = execute_regressions(cleaned_data_for_regression)

# Convert the list of results to a DataFrame
four_factor_df = pd.DataFrame(regression_outcomes)

# Check the results
if four_factor_df.empty:
    print("No regressions were successful. Please check the data and try again.
 ↪")
else:
    print(f"Completed regressions for {len(four_factor_df)} PERMNO-year␣
 ↪combinations.")

# Optionally, print out a few examples to verify
print(four_factor_df.head())
```

```
Completed regressions for 380400 PERMNO-year combinations.
   Year          Alpha       Beta_MKT       Beta_SMB       Beta_HML       Beta_MOM  \
0  1986 -5.148684e-37  2.674565e-19  3.873210e-37 -3.405869e-37 -4.286678e-37
1  1987 -6.215247e-37  3.374581e-19  2.258206e-37  6.816054e-37 -1.077309e-37
2  1986  7.091219e-38 -3.683645e-20 -5.334524e-38  4.690861e-38  5.903989e-38
3  1987 -1.156288e-38  6.371060e-21  8.445468e-39  3.471108e-39 -4.611216e-40
4  1988  3.668938e-38 -2.137498e-20  1.487754e-38  3.719674e-38 -1.386851e-38

                                              Residuals  PERMNO
0  965946    -0.122098
972009    -0.121382
979194…   10000
1  1045354   -0.196326
1050753   -0.030185
105918…   10000
2  965017     0.001808
972656     0.001710
979026…   10001
3  1041830    0.018418
1053541   -0.032279
105736…   10001
```

```
4   1127530    -0.045676
1136724     0.051632
114308…    10001
```

[232]:
```
# # Define a function to calculate RSE
# def calculate_rse(residuals, k):
#     return np.sqrt(np.sum(np.square(residuals)) / (len(residuals) - k))

# # Initialize an empty list to store RSE values
# rse_records = []

# # Assuming the `Residuals` are stored as lists or arrays
# for permno in capm_beta['PERMNO'].unique():
#     for year in capm_beta['Year'].unique():
#         # CAPM residuals
#         capm_residuals = capm_beta[(capm_beta['PERMNO'] == permno) &␣
 ↪(capm_beta['Year'] == year)]['Residuals'].values
#         if capm_residuals.size > 0:
#             rse_capm = calculate_rse(capm_residuals[0], k=2)  # CAPM has 2␣
 ↪parameters
#         else:
#             rse_capm = np.nan

#         # Three-factor model residuals
#         three_factor_residuals =␣
 ↪three_factor_beta[(three_factor_beta['PERMNO'] == permno) &␣
 ↪(three_factor_beta['Year'] == year)]['Residuals'].values
#         if three_factor_residuals.size > 0:
#             rse_ff3 = calculate_rse(three_factor_residuals[0], k=4)  # FF3␣
 ↪has 4 parameters
#         else:
#             rse_ff3 = np.nan

#         # Four-factor model residuals
#         four_factor_residuals = four_factor_df[(four_factor_df['PERMNO'] ==␣
 ↪permno) & (four_factor_df['Year'] == year)]['Residuals'].values
#         if four_factor_residuals.size > 0:
#             rse_ff4 = calculate_rse(four_factor_residuals[0], k=5)  # FF4 has␣
 ↪5 parameters
#         else:
#             rse_ff4 = np.nan

#         # Create a dictionary for the RSE of each model and append to the list
#         rse_records.append({
#             'Year': year,
#             'PERMNO': permno,
#             'RSE_CAPM': rse_capm,
```

```
#             'RSE_FF3': rse_ff3,
#             'RSE_FF4': rse_ff4
#         })

# # Convert the list of dictionaries to a DataFrame
# rse_df = pd.DataFrame(rse_records)

# # Display the RSE DataFrame
# rse_df.head()
```

[233]:
```
from concurrent.futures import ThreadPoolExecutor, as_completed
from tqdm import tqdm
import numpy as np
import pandas as pd

# Define a function to calculate RSE
def calculate_rse(residuals, k):
    return np.sqrt(np.sum(np.square(residuals)) / (len(residuals) - k))

# Define a function to compute RSE for a single stock and year
def compute_rse_for_stock_year(permno, year):
    # Find residuals for the current permno and year for each model
    capm_residuals = capm_beta[(capm_beta['PERMNO'] == permno) &␣
 ↪(capm_beta['Year'] == year)]['Residuals'].values
    three_factor_residuals = three_factor_beta[(three_factor_beta['PERMNO'] ==␣
 ↪permno) & (three_factor_beta['Year'] == year)]['Residuals'].values
    four_factor_residuals = four_factor_df[(four_factor_df['PERMNO'] == permno)␣
 ↪& (four_factor_df['Year'] == year)]['Residuals'].values

    # Compute RSE for each model if residuals exist
    rse_capm = calculate_rse(capm_residuals[0], k=2) if capm_residuals.size > 0␣
 ↪else np.nan
    rse_ff3 = calculate_rse(three_factor_residuals[0], k=4) if␣
 ↪three_factor_residuals.size > 0 else np.nan
    rse_ff4 = calculate_rse(four_factor_residuals[0], k=5) if␣
 ↪four_factor_residuals.size > 0 else np.nan

    # Return a dictionary with the results
    return {
        'Year': year,
        'PERMNO': permno,
        'RSE_CAPM': rse_capm,
        'RSE_FF3': rse_ff3,
        'RSE_FF4': rse_ff4
    }

# Get all unique combinations of PERMNO and Year
```

```
unique_permnos = capm_beta['PERMNO'].unique()
unique_years = capm_beta['Year'].unique()
stock_year_combinations = [(permno, year) for permno in unique_permnos for year␣
 ↪in unique_years]

# Use ThreadPoolExecutor to compute RSE in parallel
with ThreadPoolExecutor(max_workers=8) as executor:
    # Map the compute_rse_for_stock_year function across all combinations
    futures = [executor.submit(compute_rse_for_stock_year, permno, year) for␣
 ↪permno, year in stock_year_combinations]

    # Collect results with progress bar
    results = [future.result() for future in tqdm(as_completed(futures),␣
 ↪total=len(futures))]

# Convert the list of results to a DataFrame
rse_df = pd.DataFrame(results)

# Display the RSE DataFrame
print(rse_df.head())
```

```
100%|                                          |
1891358/1891358 [4:52:10<00:00, 107.89it/s]
```

	Year	PERMNO	RSE_CAPM	RSE_FF3	RSE_FF4
0	1986	43028	0.147087	0.164509	0.175867
1	1996	16809	NaN	NaN	NaN
2	1982	86839	NaN	NaN	NaN
3	2019	84772	NaN	NaN	NaN
4	1993	16809	NaN	NaN	NaN

[237]:
```
# Columns to fill NaN values with mean
cols_to_fill = ['Beta', 'Total_Volatility', 'Annualized_Volatility',␣
 ↪'Annualized_Volatility_Zero_Avg']

# Fill NaN with the mean of each PERMNO for the specified columns
for col in cols_to_fill:
    # Compute the mean for each PERMNO
    mean_per_permno = market_measures_df.groupby('PERMNO')[col].
 ↪transform('mean')

    # Fill NaN with the mean
    market_measures_df[col] = market_measures_df[col].fillna(mean_per_permno)
```

[238]:
```
# Fill NaN values with the mean for each PERMNO, if mean is zero fill with zero
for col in ['RSE_CAPM', 'RSE_FF3', 'RSE_FF4']:
    mean_per_permno = rse_df.groupby('PERMNO')[col].transform('mean')
    rse_df[col] = rse_df[col].fillna(mean_per_permno)
```

```
    rse_df[col] = rse_df[col].fillna(0)  # Fill NaNs with zero if the mean is␣
 ↪also NaN

# Assuming m = 12 for monthly data, as there are 12 months in a year
m = 12

# Compute Idiosyncratic Volatility as a percent
rse_df['IdioVol_CAPM'] = 100 * rse_df['RSE_CAPM'] * np.sqrt(m)
rse_df['IdioVol_FF3'] = 100 * rse_df['RSE_FF3'] * np.sqrt(m)
rse_df['IdioVol_FF4'] = 100 * rse_df['RSE_FF4'] * np.sqrt(m)

rse_df.head()
```

[238]:

	Year	PERMNO	RSE_CAPM	RSE_FF3	RSE_FF4	IdioVol_CAPM	IdioVol_FF3	IdioVol_FF4
0	1986	43028	0.147087	0.164509	0.175867	50.952468	56.987487	60.922186
1	1996	16809	0.068902	0.086429	0.099038	23.868269	29.939866	34.307736
2	1982	86839	0.135496	0.168991	0.182274	46.937087	58.540223	63.141589
3	2019	84772	0.157526	0.187301	0.201143	54.568485	64.882898	69.678024
4	1993	16809	0.068902	0.086429	0.099038	23.868269	29.939866	34.307736

[241]:
```
# Function to fill NaNs with PERMNO mean or zero if the mean is NaN
def fill_with_mean_or_zero(df, columns):
    for col in columns:
        mean_per_permno = df.groupby('PERMNO')[col].transform('mean')
        df[col] = df[col].fillna(mean_per_permno)
        df[col] = df[col].fillna(0)  # Fill NaNs with zero if the mean is also␣
 ↪NaN
    return df

# Fill NaNs for CAPM Beta DataFrame
capm_columns_to_fill = ['Alpha', 'Beta']
capm_beta = fill_with_mean_or_zero(capm_beta, capm_columns_to_fill)

# Fill NaNs for Three Factor Beta DataFrame
three_factor_columns_to_fill = ['Alpha', 'Beta_MKT', 'Beta_SMB', 'Beta_HML']
three_factor_beta = fill_with_mean_or_zero(three_factor_beta,␣
 ↪three_factor_columns_to_fill)

# Fill NaNs for Four Factor DataFrame
```

```
four_factor_columns_to_fill = ['Alpha', 'Beta_MKT', 'Beta_SMB', 'Beta_HML',␣
 ↪'Beta_MOM']
four_factor_df = fill_with_mean_or_zero(four_factor_df,␣
 ↪four_factor_columns_to_fill)
```

[243]:
```
monthly_portfolio = pd.read_csv('msf_master.csv', low_memory = False)
monthly_portfolio['date'] = pd.to_datetime(monthly_portfolio['date'], format =␣
 ↪'%Y-%m-%d', errors = 'coerce')
```

[244]:
```
monthly_portfolio.dtypes
```

[244]:
```
PERMNO             int64
date      datetime64[ns]
PRC              float64
RET               object
SHROUT           float64
vwretd           float64
dtype: object
```

[246]:
```
monthly_portfolio['PRC'] = monthly_portfolio['PRC'].abs()
monthly_portfolio['PRC'] = monthly_portfolio['PRC'].bfill()
monthly_portfolio['RET'] = pd.
 ↪to_numeric(monthly_portfolio['RET'],errors='coerce')
monthly_portfolio['RET'] = monthly_portfolio['RET'].bfill()
monthly_portfolio['SHROUT'] = pd.
 ↪to_numeric(monthly_portfolio['SHROUT'],errors='coerce')
monthly_portfolio['SHROUT'] = monthly_portfolio['SHROUT'].bfill()
monthly_portfolio['vwretd'] = pd.
 ↪to_numeric(monthly_portfolio['vwretd'],errors='coerce')
monthly_portfolio['vwretd'] = monthly_portfolio['vwretd'].bfill()
```

[249]:
```
monthly_portfolio.head()
```

[249]:

	PERMNO	date	PRC	RET	SHROUT	vwretd
0	10000	1985-12-31	4.3750	-0.257143	3680.0	0.043061
1	10000	1986-01-31	4.3750	-0.257143	3680.0	0.009830
2	10000	1986-02-28	3.2500	-0.257143	3680.0	0.072501
3	10000	1986-03-31	4.4375	0.365385	3680.0	0.053887
4	10000	1986-04-30	4.0000	-0.098592	3793.0	-0.007903

[267]:
```
rf_rate = pd.read_csv('F-F_Research_Data_Factors.CSV')
```

[271]:
```
rf_rate.head()
```

[271]:

	date	Mkt-RF	SMB	HML	RF
0	1926-07-01	2.96	-2.56	-2.43	0.22
1	1926-08-01	2.64	-1.17	3.82	0.25

```
2 1926-09-01    0.36 -1.40  0.13  0.23
3 1926-10-01   -3.24 -0.09  0.70  0.32
4 1926-11-01    2.53 -0.10 -0.51  0.31
```

[274]:
```
ohlson_o_score_results['Year'] = ohlson_o_score_results['Year'].astype(str).
 ↪astype(int)
ohlson_o_score_results['gvkey'] = ohlson_o_score_results['gvkey'].astype(str).
 ↪astype(int)
```

[276]:
```
z_score_results['Year'] = z_score_results['Year'].astype(str).astype(int)
z_score_results['gvkey'] = z_score_results['gvkey'].astype(str).astype(int)
```

[277]:
```
# Assuming monthly_portfolio is already a DataFrame with the appropriate types
monthly_portfolio['Market_Cap'] = monthly_portfolio['PRC'] *␣
 ↪monthly_portfolio['SHROUT']

# Filter stocks with market cap >= $100 million and stock price > $5
filtered_portfolio = monthly_portfolio[
    (monthly_portfolio['Market_Cap'] >= 100e6) &
    (monthly_portfolio['PRC'] > 5)
]
```

[278]:
```
# Ensure the Year column is present in the monthly_portfolio DataFrame
filtered_portfolio['Year'] = filtered_portfolio['date'].dt.year

# Merge Z-Score and O-Score into the filtered_portfolio DataFrame
portfolio_with_scores = filtered_portfolio.merge(
    z_score_results, left_on=['PERMNO', 'Year'], right_on=['gvkey', 'Year'],␣
 ↪how='left'
).merge(
    ohlson_o_score_results, left_on=['PERMNO', 'Year'], right_on=['gvkey',␣
 ↪'Year'], how='left'
)
```

[279]:
```
# Filter data to get the scores as of January 1st of each year
portfolio_with_scores['Month'] = portfolio_with_scores['date'].dt.month
january_scores = portfolio_with_scores[portfolio_with_scores['Month'] == 1]

# Rank stocks based on AltmanZ and OhlsonO and normalize the ranks to get␣
 ↪weights
january_scores['AltmanZ_rank'] = january_scores.groupby('Year')['AltmanZ'].
 ↪rank(ascending=False, method='min')
january_scores['OhlsonO_rank'] = january_scores.groupby('Year')['OhlsonO'].
 ↪rank(ascending=False, method='min')

# Normalize the ranks to get weights (you might want to adjust the ranking␣
 ↪logic based on the score)
```

```
january_scores['AltmanZ_weight'] = 1 / january_scores['AltmanZ_rank']
january_scores['OhlsonO_weight'] = 1 / january_scores['OhlsonO_rank']

# Sum of weights for each year to normalize
sum_weights_altmanz = january_scores.groupby('Year')['AltmanZ_weight'].
 ↪transform('sum')
sum_weights_ohlsono = january_scores.groupby('Year')['OhlsonO_weight'].
 ↪transform('sum')

january_scores['AltmanZ_weight'] /= sum_weights_altmanz
january_scores['OhlsonO_weight'] /= sum_weights_ohlsono
```

[280]:
```
# Create a DataFrame to hold the weights for each stock for each month
monthly_weights = portfolio_with_scores[['PERMNO', 'Year', 'Month']].copy()
monthly_weights = monthly_weights.merge(
    january_scores[['PERMNO', 'Year', 'AltmanZ_weight', 'OhlsonO_weight']],
    on=['PERMNO', 'Year'],
    how='left'
)

# Forward fill the weights for the rest of the year
monthly_weights.fillna(method='ffill', inplace=True)
```

[281]:
```
# Merge the monthly returns with the weights
monthly_returns_with_weights = portfolio_with_scores.merge(
    monthly_weights,
    on=['PERMNO', 'Year', 'Month'],
    how='left'
)

# Calculate the weighted returns
monthly_returns_with_weights['weighted_ret_AltmanZ'] =␣
 ↪monthly_returns_with_weights['RET'] *␣
 ↪monthly_returns_with_weights['AltmanZ_weight']
monthly_returns_with_weights['weighted_ret_OhlsonO'] =␣
 ↪monthly_returns_with_weights['RET'] *␣
 ↪monthly_returns_with_weights['OhlsonO_weight']
```

[282]:
```
# Group by year and sum the weighted returns for each portfolio to get the␣
 ↪annual returns
annual_returns = monthly_returns_with_weights.groupby('Year').agg({
    'weighted_ret_AltmanZ': 'sum',
    'weighted_ret_OhlsonO': 'sum'
}).reset_index()
```

[283]:
```
# Calculate excess returns over VWRETD, volatility, skewness, kurtosis
```

```
annual_returns['excess_ret_AltmanZ'] = annual_returns['weighted_ret_AltmanZ'] -␣
 ↪monthly_returns_with_weights.groupby('Year')['vwretd'].sum()
annual_returns['excess_ret_OhlsonO'] = annual_returns['weighted_ret_OhlsonO'] -␣
 ↪monthly_returns_with_weights.groupby('Year')['vwretd'].sum()

annual_returns['volatility_AltmanZ'] = monthly_returns_with_weights.
 ↪groupby('Year')['weighted_ret_AltmanZ'].std()
annual_returns['volatility_OhlsonO'] = monthly_returns_with_weights.
 ↪groupby('Year')['weighted_ret_OhlsonO'].std()
```

[284]:
```
annual_returns['volatility_AltmanZ'] = monthly_returns_with_weights.
 ↪groupby('Year')['weighted_ret_AltmanZ'].skew()
annual_returns['volatility_OhlsonO'] = monthly_returns_with_weights.
 ↪groupby('Year')['weighted_ret_OhlsonO'].skew()
```

[286]:
```
# Calculate annual kurtosis for AltmanZ-weighted returns
annual_returns['kurtosis_AltmanZ'] = monthly_returns_with_weights.
 ↪groupby('Year')['weighted_ret_AltmanZ'].apply(lambda x: x.kurt())

# Calculate annual kurtosis for OhlsonO-weighted returns
annual_returns['kurtosis_OhlsonO'] = monthly_returns_with_weights.
 ↪groupby('Year')['weighted_ret_OhlsonO'].apply(lambda x: x.kurt())
```

[287]:
```
# Calculate monthly excess returns over VWRETD for both portfolios
monthly_returns_with_weights['excess_ret_AltmanZ'] =␣
 ↪monthly_returns_with_weights['weighted_ret_AltmanZ'] -␣
 ↪monthly_returns_with_weights['vwretd']
monthly_returns_with_weights['excess_ret_OhlsonO'] =␣
 ↪monthly_returns_with_weights['weighted_ret_OhlsonO'] -␣
 ↪monthly_returns_with_weights['vwretd']

# Aggregate these monthly excess returns annually
annual_excess_returns_AltmanZ = monthly_returns_with_weights.
 ↪groupby('Year')['excess_ret_AltmanZ'].sum()
annual_excess_returns_OhlsonO = monthly_returns_with_weights.
 ↪groupby('Year')['excess_ret_OhlsonO'].sum()

# Add the annual excess returns to the annual_returns DataFrame
annual_returns['excess_ret_AltmanZ'] = annual_excess_returns_AltmanZ
annual_returns['excess_ret_OhlsonO'] = annual_excess_returns_OhlsonO
```

[288]:
```
# Merge the risk-free rate with the monthly returns DataFrame
monthly_returns_with_rf = pd.merge(monthly_returns_with_weights,␣
 ↪rf_rate[['date', 'RF']], how='left', on='date')

# Convert risk-free rate from percentage to decimals if needed
```

```
monthly_returns_with_rf['RF'] = monthly_returns_with_rf['RF'] / 100

# Calculate monthly excess returns over the risk-free rate
monthly_returns_with_rf['excess_ret_AltmanZ_rf'] =␣
 ↪monthly_returns_with_rf['weighted_ret_AltmanZ'] -␣
 ↪monthly_returns_with_rf['RF']
monthly_returns_with_rf['excess_ret_OhlsonO_rf'] =␣
 ↪monthly_returns_with_rf['weighted_ret_OhlsonO'] -␣
 ↪monthly_returns_with_rf['RF']

# Calculate the annualized excess returns and standard deviation of excess␣
 ↪returns
annualized_excess_ret_AltmanZ =␣
 ↪monthly_returns_with_rf['excess_ret_AltmanZ_rf'].mean() * 12
annualized_excess_ret_OhlsonO =␣
 ↪monthly_returns_with_rf['excess_ret_OhlsonO_rf'].mean() * 12
std_dev_excess_ret_AltmanZ = monthly_returns_with_rf['excess_ret_AltmanZ_rf'].
 ↪std() * np.sqrt(12)
std_dev_excess_ret_OhlsonO = monthly_returns_with_rf['excess_ret_OhlsonO_rf'].
 ↪std() * np.sqrt(12)

# Calculate the Sharpe Ratios
sharpe_ratio_AltmanZ = annualized_excess_ret_AltmanZ /␣
 ↪std_dev_excess_ret_AltmanZ
sharpe_ratio_OhlsonO = annualized_excess_ret_OhlsonO /␣
 ↪std_dev_excess_ret_OhlsonO
```

[311]:
```
index_data = pd.read_csv('index_monthly.csv',low_memory=False)
```

[319]:
```
index_data['DATE'] = pd.to_datetime(index_data['DATE'])
```

[321]:
```
index_data.head()
```

[321]:
```
        DATE    vwretd
0 1970-01-30 -0.073254
1 1970-02-27  0.056706
2 1970-03-31 -0.004729
3 1970-04-30 -0.105318
4 1970-05-29 -0.064346
```

[322]:
```
# Calculate the annual return. This assumes that 'vwretd' is the total monthly␣
 ↪return.
# If 'vwretd' is in percentage, you should divide by 100 before adding 1.
index_data['cumulative_return'] = (1 + index_data['vwretd']).cumprod()

# Calculate the year-end cumulative return
index_data['year'] = index_data['DATE'].dt.year
```

```
year_end_cumulative_return = index_data.groupby('year')['cumulative_return'].
 ↪last().reset_index()

# Calculate the year start cumulative return by shifting the year-end return
year_start_cumulative_return = year_end_cumulative_return.shift(1, fill_value=1)

# Calculate the annual return by dividing the year-end by year-start and␣
 ↪subtracting 1
annual_index_return = pd.DataFrame({
    'year': year_end_cumulative_return['year'],
    'vwretd': year_end_cumulative_return['cumulative_return'] /␣
 ↪year_start_cumulative_return['cumulative_return'] - 1
})

annual_index_return.head()
```

[322]:

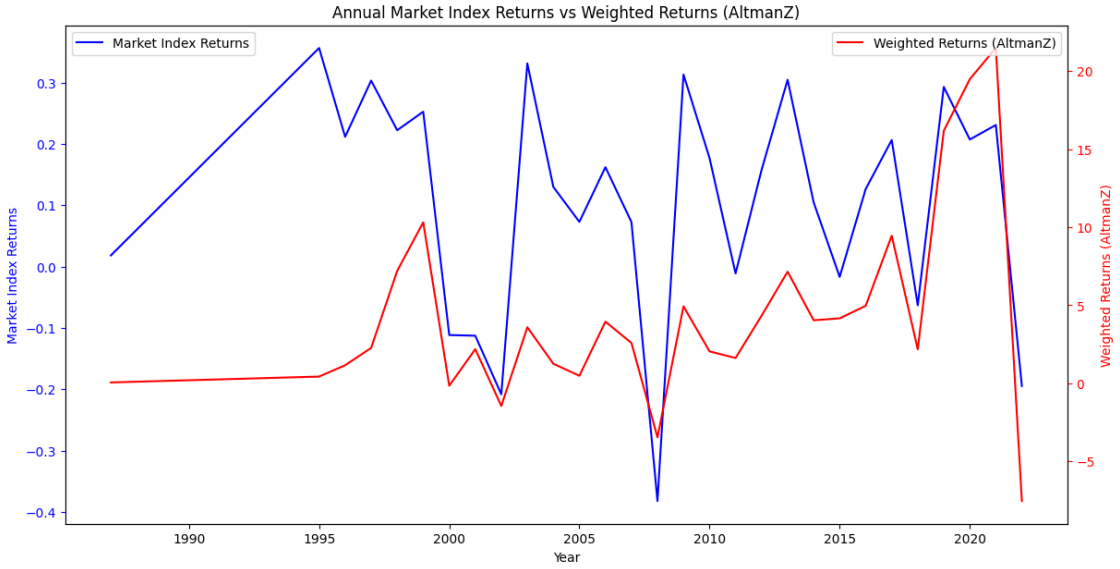
	year	vwretd
0	1970	0.000731
1	1971	0.161988
2	1972	0.173404
3	1973	-0.187496
4	1974	-0.279411

[326]:
```
annual_returns.head()
```

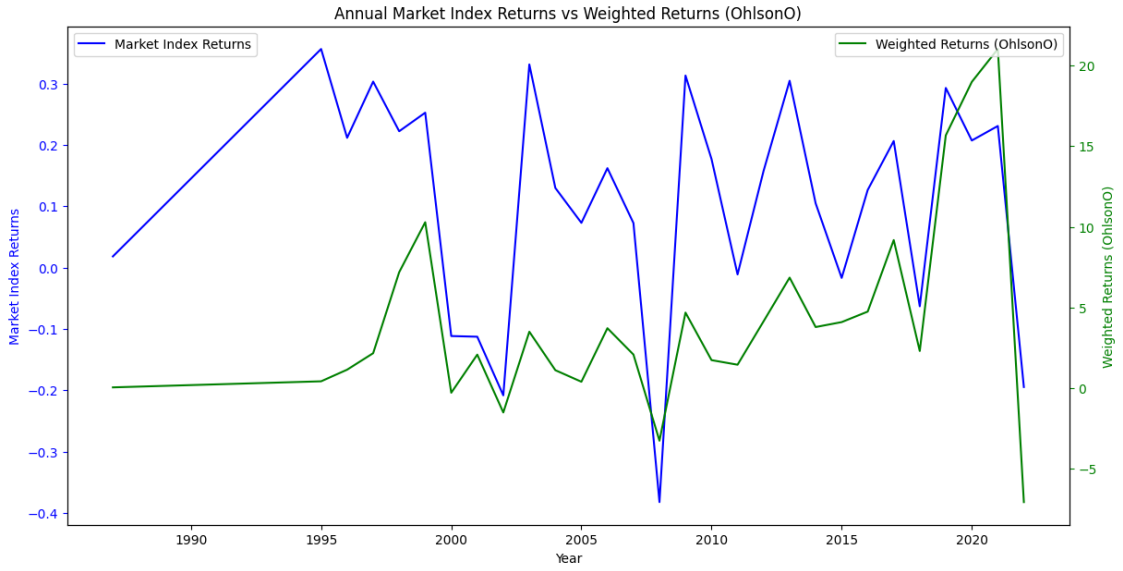
[326]:

	Year	weighted_ret_AltmanZ	weighted_ret_OhlsonO	excess_ret_AltmanZ	excess_ret_OhlsonO	volatility_AltmanZ	volatility_OhlsonO	kurtosis_AltmanZ	kurtosis_OhlsonO
0	1987	0.052640	0.052640	NaN	NaN	NaN	NaN	NaN	NaN
1	1995	0.428721	0.428721	NaN	NaN	NaN	NaN	NaN	NaN
2	1996	1.153885	1.148944	NaN	NaN	NaN	NaN	NaN	NaN
3	1997	2.263348	2.169689	NaN	NaN	NaN	NaN	NaN	NaN
4	1998	7.199123	7.188320	NaN	NaN	NaN	NaN	NaN	NaN

[324]:
```
# Merge both DataFrames on the year
merged_df = pd.merge(annual_index_return, annual_returns, left_on='year',␣
 ↪right_on='Year', how='inner')

# Create a plot with two y-axes
fig, ax1 = plt.subplots(figsize=(14, 7))

# Plot the market index returns
ax1.plot(merged_df['year'], merged_df['vwretd'], color='b', label='Market Index␣
 ↪Returns')
ax1.set_xlabel('Year')
ax1.set_ylabel('Market Index Returns', color='b')
ax1.tick_params('y', colors='b')

# Create another y-axis for the weighted returns based on AltmanZ scores
ax2 = ax1.twinx()
ax2.plot(merged_df['Year'], merged_df['weighted_ret_AltmanZ'], color='r',␣
 ↪label='Weighted Returns (AltmanZ)')
ax2.set_ylabel('Weighted Returns (AltmanZ)', color='r')
ax2.tick_params('y', colors='r')

# Add a title and a legend
plt.title('Annual Market Index Returns vs Weighted Returns (AltmanZ)')
ax1.legend(loc='upper left')
ax2.legend(loc='upper right')

# Show the plot
plt.show()
```



[327]:
```
# Create a plot with two y-axes
fig, ax1 = plt.subplots(figsize=(14, 7))

# Plot the market index returns
ax1.plot(merged_df['year'], merged_df['vwretd'], color='b', label='Market Index␣
 ↪Returns')
ax1.set_xlabel('Year')
ax1.set_ylabel('Market Index Returns', color='b')
ax1.tick_params('y', colors='b')

# Create another y-axis for the weighted returns based on OhlsonO scores
ax2 = ax1.twinx()
ax2.plot(merged_df['Year'], merged_df['weighted_ret_OhlsonO'], color='g',␣
 ↪label='Weighted Returns (OhlsonO)')
ax2.set_ylabel('Weighted Returns (OhlsonO)', color='g')
ax2.tick_params('y', colors='g')

# Add a title and a legend
plt.title('Annual Market Index Returns vs Weighted Returns (OhlsonO)')
ax1.legend(loc='upper left')
ax2.legend(loc='upper right')

# Show the plot
plt.show()
```



1. Performance Through Market Cycles: Fundamental indexing strategies, which rely on firm-specific metrics rather than market prices, have the potential to outperform market-cap weighted indexes during certain market cycles. Especially during periods when market prices

may be overinflated due to speculative trading, fundamental indexes could provide a more stable return by focusing on underlying company health and performance.

2. Impact of Size and Price Filters: By restricting the universe to stocks with a market cap of over $100 million and a stock price over $5, both strategies likely exclude many small-cap and lower-priced stocks. This could reduce exposure to the higher volatility often associated with these stocks, potentially leading to a more stable performance but also possibly missing out on the higher returns these riskier assets can sometimes offer.

3. Diversification and Risk: Fundamental weighted portfolios may offer better diversification as they do not concentrate holdings in the largest companies by market cap, which is a common critique of market-cap weighted indexes. This could potentially lead to lower risk levels as measured by standard deviation or total volatility, and may also affect skewness and kurtosis of returns.

4. Comparison with MKT, SMB, and HML Factors: The performance of fundamental weighted indexes relative to market-cap weighted indexes might differ when compared to the MKT (overall market), SMB (small minus big cap), and HML (high minus low book-to-market ratio) factors. Fundamental indexes may have a different exposure to these factors, which can lead to different performance characteristics, especially in different stages of the business cycle.

5. Sharpe and Information Ratios: The Sharpe Ratio measures excess return per unit of risk (standard deviation), and the Information Ratio assesses the excess return of a portfolio relative to a benchmark, adjusted for the volatility of those excess returns. Fundamental indexes might show different Sharpe and Information Ratios compared to market-cap weighted indexes, which would highlight their risk-adjusted performance differences.

6. Adaptability to Market Changes: Fundamental indexes that are rebalanced annually may be less reactive to immediate market movements, which can be a double-edged sword. On one hand, it can protect against the whims of market sentiment; on the other hand, it might delay the portfolio's adjustment to genuine shifts in economic conditions.

7. Performance in Recessions vs. Expansions: During recessions, market-cap weighted indexes may suffer from a rapid decline as stock prices plummet, whereas fundamental indexes might be more resilient if their underlying companies have strong fundamentals. Conversely, during expansions, market-cap indexes might capture the upside more quickly.

8. Observations on Indexing Mechanisms: Fundamental indexing challenges the notion that the market price is always the best indicator of a company's value. By using fundamentals, this approach attempts to capitalize on potential market inefficiencies. However, it also requires a strong belief in the chosen fundamentals to drive long-term performance.

[328]:
```
monthly_portfolio = pd.read_csv('msf_master.csv', low_memory = False)
monthly_portfolio['date'] = pd.to_datetime(monthly_portfolio['date'], format =␣
 ↪'%Y-%m-%d', errors = 'coerce')
monthly_portfolio['PRC'] = monthly_portfolio['PRC'].abs()
monthly_portfolio['PRC'] = monthly_portfolio['PRC'].bfill()
monthly_portfolio['RET'] = pd.
 ↪to_numeric(monthly_portfolio['RET'],errors='coerce')
monthly_portfolio['RET'] = monthly_portfolio['RET'].bfill()
```

```
monthly_portfolio['SHROUT'] = pd.
 ↪to_numeric(monthly_portfolio['SHROUT'],errors='coerce')
monthly_portfolio['SHROUT'] = monthly_portfolio['SHROUT'].bfill()
monthly_portfolio['vwretd'] = pd.
 ↪to_numeric(monthly_portfolio['vwretd'],errors='coerce')
monthly_portfolio['vwretd'] = monthly_portfolio['vwretd'].bfill()
```

[329]:
```
# Assuming monthly_portfolio is already a DataFrame with the appropriate types
monthly_portfolio['Market_Cap'] = monthly_portfolio['PRC'] *␣
 ↪monthly_portfolio['SHROUT']

# Filter stocks with market cap >= $100 million and stock price > $5
filtered_portfolio = monthly_portfolio[
    (monthly_portfolio['Market_Cap'] >= 100e6) &
    (monthly_portfolio['PRC'] > 5)
]

# Ensure the Year and Month columns are present in the filtered_portfolio␣
 ↪DataFrame
filtered_portfolio['Year'] = filtered_portfolio['date'].dt.year
filtered_portfolio['Month'] = filtered_portfolio['date'].dt.month
```

[330]:
```
portfolio_with_measures = filtered_portfolio.merge(
    market_measures_df,
    on=['PERMNO', 'Year'],
    how='left'
)
```

[331]:
```
# Filter data to get the measures as of January 1st of each year
january_measures = portfolio_with_measures[portfolio_with_measures['Month'] ==␣
 ↪1]

# Rank stocks based on volatility measures and normalize the ranks to get␣
 ↪weights
january_measures['Total_Volatility_rank'] = january_measures.
 ↪groupby('Year')['Total_Volatility'].rank(ascending=True, method='min')
january_measures['Annualized_Volatility_rank'] = january_measures.
 ↪groupby('Year')['Annualized_Volatility'].rank(ascending=True, method='min')
january_measures['Annualized_Volatility_Zero_Avg_rank'] = january_measures.
 ↪groupby('Year')['Annualized_Volatility_Zero_Avg'].rank(ascending=True,␣
 ↪method='min')

# Normalize the ranks to get weights (invert the rank as a weight, lowest␣
 ↪volatility gets the highest weight)
january_measures['Total_Volatility_weight'] = 1 /␣
 ↪january_measures['Total_Volatility_rank']
```

```
january_measures['Annualized_Volatility_weight'] = 1 /␣
 ↪january_measures['Annualized_Volatility_rank']
january_measures['Annualized_Volatility_Zero_Avg_weight'] = 1 /␣
 ↪january_measures['Annualized_Volatility_Zero_Avg_rank']

# Sum of weights for each year to normalize
sum_weights_total_volatility = january_measures.
 ↪groupby('Year')['Total_Volatility_weight'].transform('sum')
sum_weights_annualized_volatility = january_measures.
 ↪groupby('Year')['Annualized_Volatility_weight'].transform('sum')
sum_weights_annualized_volatility_zero_avg = january_measures.
 ↪groupby('Year')['Annualized_Volatility_Zero_Avg_weight'].transform('sum')

january_measures['Total_Volatility_weight'] /= sum_weights_total_volatility
january_measures['Annualized_Volatility_weight'] /=␣
 ↪sum_weights_annualized_volatility
january_measures['Annualized_Volatility_Zero_Avg_weight'] /=␣
 ↪sum_weights_annualized_volatility_zero_avg

# Now we have the weights for each stock for the January portfolio, we will␣
 ↪carry forward these weights for the rest of the year
```

[332]:
```
# Create a DataFrame to hold the weights for each stock for each month
monthly_weights = portfolio_with_measures[['PERMNO', 'Year', 'Month']].copy()
monthly_weights = monthly_weights.merge(
    january_measures[['PERMNO', 'Year', 'Total_Volatility_weight',␣
 ↪'Annualized_Volatility_weight', 'Annualized_Volatility_Zero_Avg_weight']],
    on=['PERMNO', 'Year'],
    how='left'
)

# Forward fill the weights for the rest of the year
monthly_weights.fillna(method='ffill', inplace=True)

# Merge the monthly returns with the weights
monthly_returns_with_weights = portfolio_with_measures.merge(
    monthly_weights,
    on=['PERMNO', 'Year', 'Month'],
    how='left'
)

# Calculate the weighted returns
monthly_returns_with_weights['weighted_ret_Total_Volatility'] =␣
 ↪monthly_returns_with_weights['RET'] *␣
 ↪monthly_returns_with_weights['Total_Volatility_weight']
```

```
monthly_returns_with_weights['weighted_ret_Annualized_Volatility'] =␣
 ↪monthly_returns_with_weights['RET'] *␣
 ↪monthly_returns_with_weights['Annualized_Volatility_weight']
monthly_returns_with_weights['weighted_ret_Annualized_Volatility_Zero_Avg'] =␣
 ↪monthly_returns_with_weights['RET'] *␣
 ↪monthly_returns_with_weights['Annualized_Volatility_Zero_Avg_weight']

# Group by year and sum the weighted returns for each portfolio to get the␣
 ↪annual returns
annual_returns = monthly_returns_with_weights.groupby('Year').agg({
    'weighted_ret_Total_Volatility': 'sum',
    'weighted_ret_Annualized_Volatility': 'sum',
    'weighted_ret_Annualized_Volatility_Zero_Avg': 'sum'
}).reset_index()
```

[333]:
```
# Create a DataFrame to hold the weights for each stock for each month
monthly_weights = portfolio_with_measures[['PERMNO', 'Year', 'Month']].copy()
monthly_weights = monthly_weights.merge(
    january_measures[['PERMNO', 'Year', 'Total_Volatility_weight',␣
 ↪'Annualized_Volatility_weight', 'Annualized_Volatility_Zero_Avg_weight']],
    on=['PERMNO', 'Year'],
    how='left'
)

# Forward fill the weights for the rest of the year
monthly_weights.fillna(method='ffill', inplace=True)

# Merge the monthly returns with the weights
monthly_returns_with_weights = portfolio_with_measures.merge(
    monthly_weights,
    on=['PERMNO', 'Year', 'Month'],
    how='left'
)

# Calculate the weighted returns
monthly_returns_with_weights['weighted_ret_Total_Volatility'] =␣
 ↪monthly_returns_with_weights['RET'] *␣
 ↪monthly_returns_with_weights['Total_Volatility_weight']
monthly_returns_with_weights['weighted_ret_Annualized_Volatility'] =␣
 ↪monthly_returns_with_weights['RET'] *␣
 ↪monthly_returns_with_weights['Annualized_Volatility_weight']
monthly_returns_with_weights['weighted_ret_Annualized_Volatility_Zero_Avg'] =␣
 ↪monthly_returns_with_weights['RET'] *␣
 ↪monthly_returns_with_weights['Annualized_Volatility_Zero_Avg_weight']
```

```
# Group by year and sum the weighted returns for each portfolio to get the annual returns
annual_returns = monthly_returns_with_weights.groupby('Year').agg({
    'weighted_ret_Total_Volatility': 'sum',
    'weighted_ret_Annualized_Volatility': 'sum',
    'weighted_ret_Annualized_Volatility_Zero_Avg': 'sum'
}).reset_index()
```

[334]:
```
annual_returns.head()
```

[334]:

	Year	weighted_ret_Total_Volatility	weighted_ret_Annualized_Volatility	weighted_ret_Annualized_Volatility_Zero_Avg
0	1987	0.000113	0.000113	0.000113
1	1995	0.007057	0.007057	0.007023
2	1996	0.265566	0.265566	0.265153
3	1997	0.349942	0.349942	0.350096
4	1998	0.388388	0.388388	0.382383

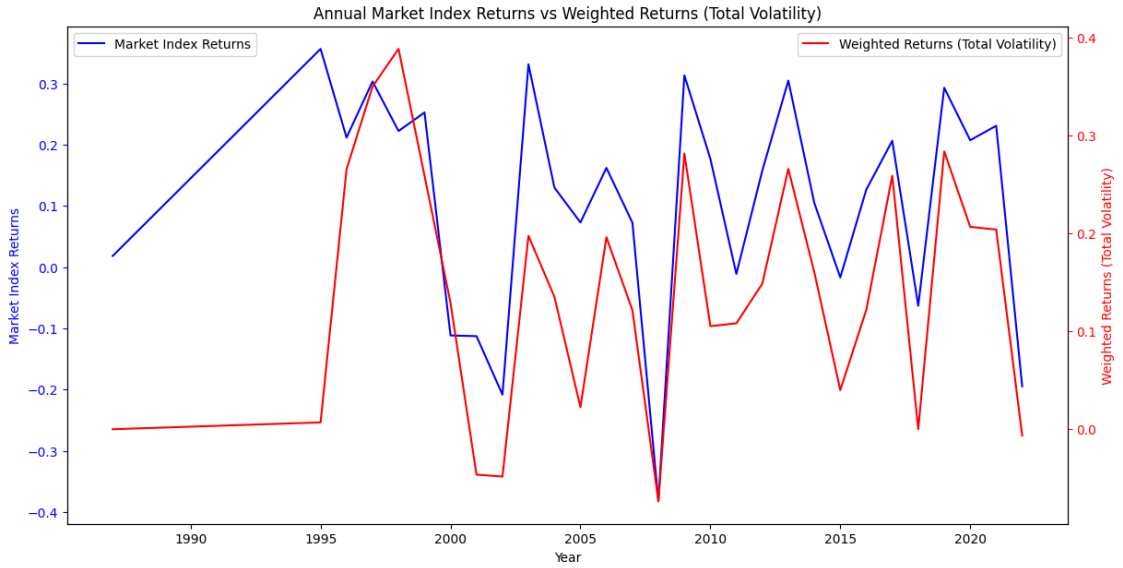
[335]:
```
index_data = pd.read_csv('index_monthly.csv',low_memory=False)
index_data['DATE'] = pd.to_datetime(index_data['DATE'])
# Calculate the annual return. This assumes that 'vwretd' is the total monthly return.
# If 'vwretd' is in percentage, you should divide by 100 before adding 1.
index_data['cumulative_return'] = (1 + index_data['vwretd']).cumprod()

# Calculate the year-end cumulative return
index_data['year'] = index_data['DATE'].dt.year
year_end_cumulative_return = index_data.groupby('year')['cumulative_return'].last().reset_index()

# Calculate the year start cumulative return by shifting the year-end return
year_start_cumulative_return = year_end_cumulative_return.shift(1, fill_value=1)

# Calculate the annual return by dividing the year-end by year-start and subtracting 1
annual_index_return = pd.DataFrame({
    'year': year_end_cumulative_return['year'],
    'vwretd': year_end_cumulative_return['cumulative_return'] / year_start_cumulative_return['cumulative_return'] - 1
})
```

```
annual_index_return.head()
```

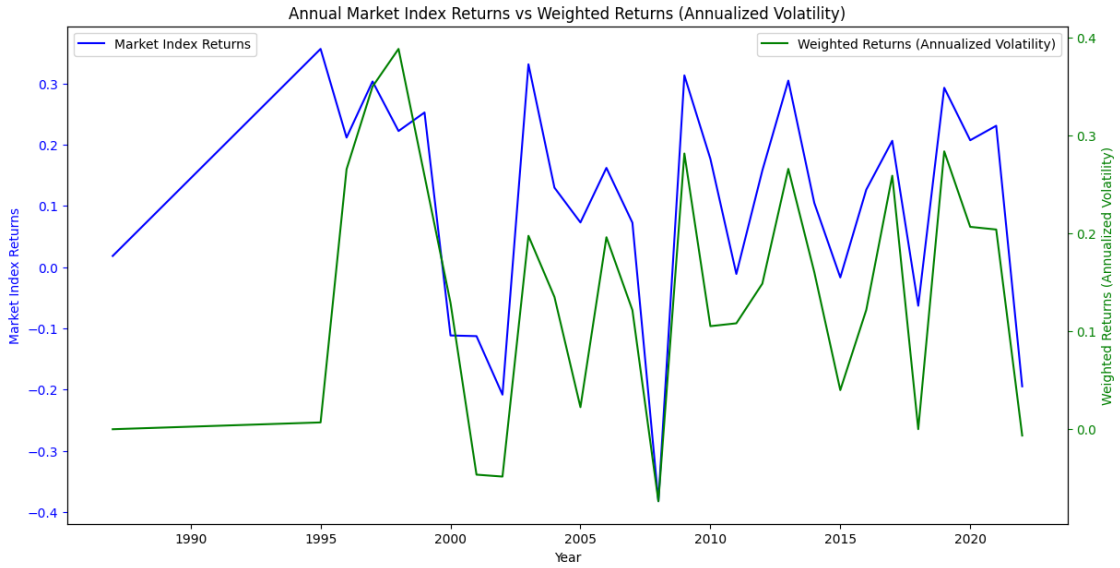
[335]:

	year	vwretd
0	1970	0.000731
1	1971	0.161988
2	1972	0.173404
3	1973	-0.187496
4	1974	-0.279411

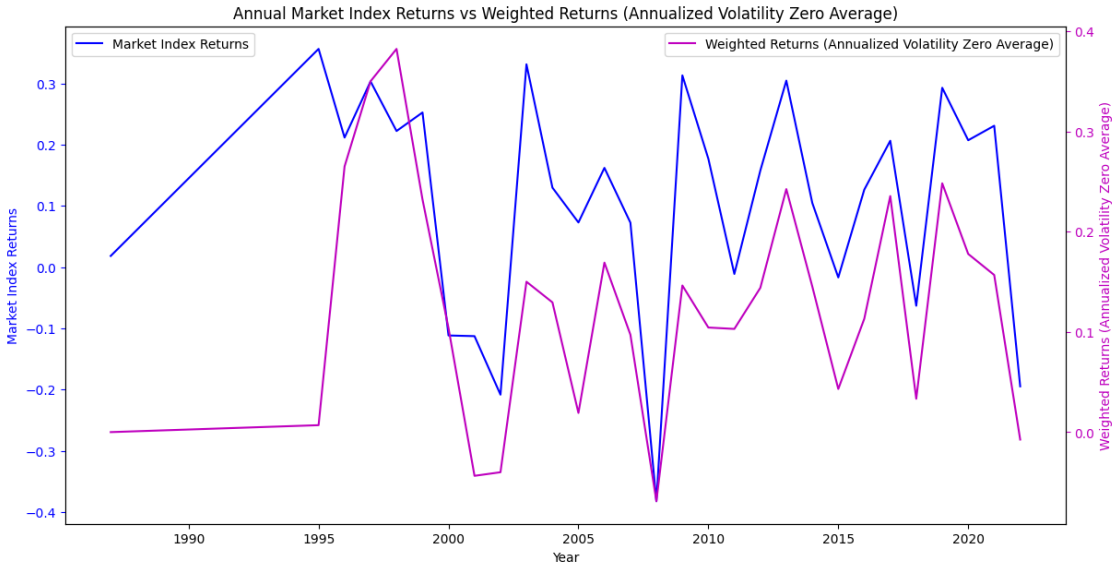
[338]:
```
# Merge both DataFrames on the year
merged_df = pd.merge(annual_index_return, annual_returns, left_on='year',␣
 ↪right_on='Year', how='inner')

# Create a plot with two y-axes
fig, ax1 = plt.subplots(figsize=(14, 7))

# Plot the market index returns
ax1.plot(merged_df['year'], merged_df['vwretd'], color='b', label='Market Index␣
 ↪Returns')
ax1.set_xlabel('Year')
ax1.set_ylabel('Market Index Returns', color='b')
ax1.tick_params('y', colors='b')

# Create another y-axis for the weighted returns based on AltmanZ scores
ax2 = ax1.twinx()
ax2.plot(merged_df['Year'], merged_df['weighted_ret_Total_Volatility'],␣
 ↪color='r', label='Weighted Returns (Total Volatility)')
ax2.set_ylabel('Weighted Returns (Total Volatility)', color='r')
ax2.tick_params('y', colors='r')

# Add a title and a legend
plt.title('Annual Market Index Returns vs Weighted Returns (Total Volatility)')
ax1.legend(loc='upper left')
ax2.legend(loc='upper right')

# Show the plot
plt.show()
```



[339]:
```
# Create a plot with two y-axes
fig, ax1 = plt.subplots(figsize=(14, 7))

# Plot the market index returns
ax1.plot(merged_df['year'], merged_df['vwretd'], color='b', label='Market Index␣
 ↪Returns')
ax1.set_xlabel('Year')
ax1.set_ylabel('Market Index Returns', color='b')
ax1.tick_params('y', colors='b')

# Create another y-axis for the weighted returns based on OhlsonO scores
ax2 = ax1.twinx()
ax2.plot(merged_df['Year'], merged_df['weighted_ret_Annualized_Volatility'],␣
 ↪color='g', label='Weighted Returns (Annualized Volatility)')
ax2.set_ylabel('Weighted Returns (Annualized Volatility)', color='g')
ax2.tick_params('y', colors='g')

# Add a title and a legend
plt.title('Annual Market Index Returns vs Weighted Returns (Annualized␣
 ↪Volatility)')
ax1.legend(loc='upper left')
ax2.legend(loc='upper right')

# Show the plot
plt.show()
```



[350]:
```
# Create a plot with two y-axes
fig, ax1 = plt.subplots(figsize=(14, 7))

# Plot the market index returns
ax1.plot(merged_df['year'], merged_df['vwretd'], color='b', label='Market Index␣
 ↪Returns')
ax1.set_xlabel('Year')
ax1.set_ylabel('Market Index Returns', color='b')
ax1.tick_params('y', colors='b')

# Create another y-axis for the weighted returns based on OhlsonO scores
ax2 = ax1.twinx()
ax2.plot(merged_df['Year'],␣
 ↪merged_df['weighted_ret_Annualized_Volatility_Zero_Avg'], color='m',␣
 ↪label='Weighted Returns (Annualized Volatility Zero Average)')
ax2.set_ylabel('Weighted Returns (Annualized Volatility Zero Average)',␣
 ↪color='m')
ax2.tick_params('y', colors='m')

# Add a title and a legend
plt.title('Annual Market Index Returns vs Weighted Returns (Annualized␣
 ↪Volatility Zero Average)')
ax1.legend(loc='upper left')
ax2.legend(loc='upper right')

# Show the plot
plt.show()
```



1. Risk-Adjusted Return Strategies: By using market variables such as total and annualized volatility, the indexing strategy focuses on risk-adjusted returns. This could attract investors who are more risk-averse and looking for more stable investment opportunities, especially during volatile market periods.

2. Filtering Effect: The restriction on stocks based on market capitalization and stock price aims to create a more stable investment universe by eliminating small-cap stocks, which are often more volatile, and penny stocks, which can be subject to price manipulations and liquidity issues.

3. Variable-Based Weighting: Ranking stocks based on market variables and using these for weighting can lead to a portfolio that might behave differently across various market conditions. For instance, in a bullish market, stocks with lower volatility might underperform the market, while in a bearish market, they could provide downside protection.

4. Rebalancing and Timing: The strategy of holding a portfolio for an entire year and rebalancing on the last trading day could result in a lag in responsiveness to market changes. However, it may also prevent overreacting to short-term market fluctuations, thus potentially reducing transaction costs and overtrading.

5. Comparison with Market Cycles: Visualizing returns over time relative to VWRETD returns, particularly across different business cycles, can illustrate the resilience or sensitivity of this alternate indexing strategy to economic changes. It would be particularly interesting to observe how this strategy performs during market downturns or NBER-defined recessions.

6. Performance Metrics: Computing returns, excess returns, volatility, skewness, and kurtosis offers a comprehensive view of the risk profile and performance characteristics of the return-based indexed portfolio. The Sharpe Ratio will give insight into the risk-adjusted performance, while the Information Ratio will show how much excess return is achieved per unit of risk relative to the benchmark.

7. Comparison with Factor Returns: Analyzing how the return-based index compares with MKT, SMB, and HML returns would show the factor exposures of the portfolio. This analysis could reveal whether the alternate indexing strategy systematically leans towards or away from these known risk factors.

8. Alternative vs. Market Cap Indexing: Alternate indexing mechanisms such as return-based indexing offer a different take on portfolio construction, which can sometimes lead to outperformance of market cap-based indexing during certain periods. It shifts focus from size to the underlying risk-return characteristics of the stocks, which could be advantageous in specific market environments.