Assignment 9: Distance to Default

Analysis

- Plot the percentiles of DD, do you see any trends?
  - The analysis of DD and PD metrics underscores their importance in reflecting changes in economic conditions and the financial health of corporations. Understanding how these indicators interact with the BAA spread during economic downturns offers valuable insights into market dynamics and investor sentiments. The observed trends suggest that while companies may have some degree of control over their default risk, larger economic forces play a significant role in shaping both the actual likelihood of default and how the market perceives these risks. This interplay highlights the complex nature of financial markets and the myriad factors influencing investor decisions and corporate stability.
- What do you notice about the NBER recession data?
  - During economic downturns, the behavior of the metrics DD and PD is notably different. Typically, DD experiences a significant increase, while the response of PD tends to be more variable and less consistent.
  - During economic downturns, the variations in DD and PD metrics are largely a result of how markets and businesses react to increased uncertainty and the looming risk of economic decline. DD's unexpected increase during recessions can be linked to factors like governmental fiscal interventions, more lenient monetary policies, and corporate measures aimed at enhancing liquidity and financial resilience. On the other hand, the rise in PD during these periods aligns with typical forecasts, as the real and perceived risks of bankruptcy escalate in response to the prevailing economic difficulties.
- Refelect and comment on BAA spread
  - Elevated levels of DD are typically seen in conjunction with economic downturns, as evidenced in the early 1980s, 1990s, 2000s, and the 2007-2009 financial crisis. Often, DD begins to rise before a downturn and peaks during or just after it. Meanwhile, the BAA spread usually widens during these periods, indicating an increase in credit risk premiums. This uptick in DD might be due to companies strengthening their financial positions in anticipation of or in response to worsening economic conditions. Alternatively, it could reflect changes in market perceptions of risk. The growing BAA spread during these times suggests a higher premium demanded by investors for taking on corporate debt risks compared to

nearly risk-free Federal funds, likely a result of heightened investor caution in economically challenging times.

- PD's trend is relatively consistent over time, but it shows a slight uptick during periods of economic strain, particularly notable during the 2007-2009 financial crisis. While PD doesn't fluctuate as sharply as DD, it does reveal a slow but steady increase during economic difficulties.

- The overall stability of PD in a stable economic environment suggests that the default risk inherent in firms doesn't vary significantly under normal circumstances. However, the increase in PD during recessions may indicate either an actual rise in defaults or an adjustment in risk perception as companies face reduced profits and liquidity challenges. The credit market-driven crisis of 2007-2009 distinctly highlights the significant impact such economic downturns can have on PD, emphasizing the extreme nature of the challenges posed by that period.

In [ ]:
```
import pandas as pd
import numpy as np
import random
from scipy.stats import norm
from scipy.optimize import fsolve
from scipy.optimize import newton
import matplotlib.pyplot as plt
```

In [ ]:
```
file_path = 'DTB3.csv'
fed_df = pd.read_csv(file_path)

fed_df['DTB3'] = pd.to_numeric(fed_df['DTB3'], errors='coerce')
fed_df['r'] = np.log(1 + fed_df['DTB3'] / 100)
fed_df['YEAR'] =pd.to_datetime(fed_df['DATE']).dt.year
fed_df.groupby(by = ['YEAR'])['r'].first()
fed_df = fed_df.dropna()
fed_df = fed_df.drop('YEAR', axis = 1)

display(fed_df)
```

	DATE	DTB3	r
0	1970-01-02	7.92	0.076220
1	1970-01-05	7.91	0.076127
2	1970-01-06	7.93	0.076313
3	1970-01-07	7.90	0.076035
4	1970-01-08	7.91	0.076127
...	...	...	...
13820	2022-12-23	4.23	0.041430
13822	2022-12-27	4.35	0.042580
13823	2022-12-28	4.35	0.042580
13824	2022-12-29	4.34	0.042485
13825	2022-12-30	4.30	0.042101

13241 rows × 3 columns

In [ ]:
```
# Replace 'your_file.csv' with the path to your CSV file
file_path = 'funda_2022.csv'

# Read the CSV file
funda_df = pd.read_csv(file_path, usecols = ['indfmt', 'datafmt', 'popsrc', 'fic',
display(funda_df)
```

```
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\3450330936.py:5: DtypeWarni
ng: Columns (8) have mixed types. Specify dtype option on import or set low_memory=F
alse.
  funda_df = pd.read_csv(file_path, usecols = ['indfmt', 'datafmt', 'popsrc', 'fic',
'consol', 'cusip', 'datadate', 'dlc', 'dltt', 'fyear'])
```

	datadate	fyear	indfmt	consol	popsrc	datafmt	cusip	dlc	dltt
0	1961-12-31	1961	INDL	C	D	STD	000032102	NaN	0.100
1	1962-12-31	1962	INDL	C	D	STD	000032102	NaN	0.000
2	1963-12-31	1963	INDL	C	D	STD	000032102	NaN	0.015
3	1964-12-31	1964	INDL	C	D	STD	000032102	0.088	0.522
4	1965-12-31	1965	INDL	C	D	STD	000032102	0.300	1.154
...	...	...	...	...	...	...	...	...	...
539313	2022-12-31	2022	INDL	C	D	STD	G2415A113	180.576	958.747
539314	2021-12-31	2021	INDL	C	D	STD	405552100	1224.107	117.807
539315	2022-12-31	2022	INDL	C	D	STD	405552100	526.542	12052.615
539316	2021-12-31	2021	INDL	C	D	STD	00449R109	102.676	20.505
539317	2022-12-31	2022	INDL	C	D	STD	00449R109	7.130	343.767

539318 rows × 10 columns

In [ ]:

```
funda_df['datadate'] = pd.to_datetime(funda_df['datadate'], errors='coerce')
funda_df['cusip'] = funda_df['cusip'].str[0:6]
funda_df['YEAR'] = funda_df['datadate'].dt.year
funda_df['dlc'] = funda_df['dlc']*1000000
funda_df['dltt'] = funda_df['dltt']*1000000
funda_df['F'] = funda_df['dlc'] + 0.5 * funda_df['dltt']

#apply the filters
funda_df = funda_df[
    (funda_df['indfmt'] == 'INDL') &
    (funda_df['datafmt'] == 'STD') &
    (funda_df['popsrc'] == 'D') &
    (funda_df['fic'] == 'USA') &
    (funda_df['YEAR'] >= 1970) &
    (funda_df['YEAR'] <= 2020)
]



#Lag by 1 year
```

```
funda_df = funda_df.sort_values(by ='datadate')
funda_df['YEAR'] = funda_df['YEAR'] + 1

#get rid of nans
funda_df = funda_df.dropna()


# Display the filtered dataframe
print(funda_df)
```

```
          datadate  fyear indfmt consol popsrc datafmt   cusip         dlc  \
64428   1970-01-31   1969   INDL      C      D     STD  268226   3305000.0
82907   1970-01-31   1969   INDL      C      D     STD  368226   2014000.0
218033  1970-01-31   1969   INDL      C      D     STD  194831  23759000.0
180006  1970-01-31   1969   INDL      C      D     STD  821328   9751000.0
87275   1970-01-31   1969   INDL      C      D     STD  379892   3094000.0
...            ...    ...    ...    ...    ...     ...     ...         ...
299179  2020-12-31   2020   INDL      C      D     STD  64049K    708000.0
299404  2020-12-31   2020   INDL      C      D     STD  55955D   9813000.0
299197  2020-12-31   2020   INDL      C      D     STD  76118L    729000.0
396489  2020-12-31   2020   INDL      C      D     STD  867892   7290000.0
299367  2020-12-31   2020   INDL      C      D     STD  749119   3650000.0

               dltt  fic  YEAR            F
64428     3010000.0  USA  1971    4810000.0
82907     1933000.0  USA  1971    2980500.0
218033   63963000.0  USA  1971   55740500.0
180006    3803000.0  USA  1971   11652500.0
87275     5985000.0  USA  1971    6086500.0
...             ...  ...   ...          ...
299179   11414000.0  USA  2021    6415000.0
299404   32278000.0  USA  2021   25952000.0
299197    1764000.0  USA  2021    1611000.0
396489  788051000.0  USA  2021  401315500.0
299367  193124000.0  USA  2021  100212000.0

[336645 rows x 12 columns]
```

In [ ]:

```
file_path = 'dsf.csv'

chunk_size=10000

percentage = 0.1  # Example: 10% of each chunk
dataframes = []

for chunk in pd.read_csv(file_path, chunksize=chunk_size, usecols=['CUSIP', 'DATE',
    sample = chunk.sample(frac=percentage)  # Sample each chunk
    dataframes.append(sample)

# Concatenate all sampled chunks
dsf_df = pd.concat(dataframes, ignore_index=True)

dsf_df['DATE'] = pd.to_datetime(dsf_df['DATE'], errors='coerce', format='%Y%m%d')
dsf_df['YEAR'] = dsf_df['DATE'].dt.year
dsf_df['CUSIP'] = dsf_df['CUSIP'].str[0:6]
```

```
#converting columns to numeric
dsf_df['RET'] = pd.to_numeric(dsf_df['RET'], errors='coerce')
dsf_df['SHROUT'] = pd.to_numeric(dsf_df['SHROUT'], errors='coerce')
dsf_df['PRC'] = pd.to_numeric(dsf_df['PRC'], errors='coerce')

dsf_df['E'] = (dsf_df['PRC']).abs() * dsf_df['SHROUT']

dsf_df = dsf_df[dsf_df['YEAR'] >= 1970]
#lag
dsf_df['YEAR'] = dsf_df['YEAR'] + 1
dsf_df = dsf_df.sort_values(by ='DATE')


#get rid of nans
dsf_df = dsf_df.dropna()
print(dsf_df)
```

```
          CUSIP        DATE        PRC       RET     SHROUT  YEAR             E
3955909  95751D 1970-01-02   21.62500  0.017647     6699.0  1971  1.448659e+05
2240104  296470 1970-01-02   31.37500  0.004000    12137.0  1971  3.807984e+05
244675   067149 1970-01-02   46.37500  0.024862     2343.0  1971  1.086566e+05
3088241  340632 1970-01-02   34.00000  0.038168     1453.0  1971  4.940200e+04
4303307  610304 1970-01-02   42.00000  0.012048     6313.0  1971  2.651460e+05
...         ...        ...        ...       ...        ...   ...           ...
7725531  G3223R 2021-06-30  252.00999 -0.003913    40084.0  2022  1.010157e+07
9435682  922042 2021-06-30   54.31000 -0.006221  1550390.0  2022  8.420168e+07
1256309  140755 2021-06-30   14.27000 -0.011088    50052.0  2022  7.142420e+05
1864631  46434V 2021-06-30   50.52000  0.000000   111750.0  2022  5.645610e+06
9688530  696389 2021-06-30    4.07000 -0.016908     7302.0  2022  2.971914e+04

[4936020 rows x 7 columns]
```

In [ ]:

```
funda_df = funda_df.rename(columns=str.upper)
merged_df = pd.merge(dsf_df, funda_df, on=['CUSIP', 'YEAR'], how='inner')
display(merged_df)
```

	CUSIP	DATE	PRC	RET	SHROUT	YEAR	E	DATADATE	FYEAR
0	296470	1970-01-02	31.375	0.004000	12137.0	1971	380798.375	1970-10-31	197(
1	296470	1970-01-08	29.750	-0.024590	12137.0	1971	361075.750	1970-10-31	197(
2	296470	1970-01-29	29.625	-0.020661	12137.0	1971	359558.625	1970-10-31	197(
3	296470	1970-02-17	29.875	-0.004167	12137.0	1971	362592.875	1970-10-31	197(
4	296470	1970-02-24	30.125	-0.008230	12137.0	1971	365627.125	1970-10-31	197(
...	...	...	...	...	...	...	...	...	..
2859059	809171	2020-12-21	16.200	-0.248260	14502.0	2021	234932.400	2020-12-31	202(
2859060	809171	2020-12-22	13.100	-0.191358	14502.0	2021	189976.200	2020-12-31	202(
2859061	91680M	2020-12-21	41.100	-0.067816	72460.0	2021	2978106.000	2020-12-31	202(
2859062	09077B	2020-12-23	32.300	-0.032934	30596.0	2021	988250.800	2020-12-31	202(
2859063	212873	2020-12-24	10.200	-0.004878	75000.0	2021	765000.000	2020-12-31	202(

2859064 rows × 17 columns

In [ ]:
```
#Annualizing the data
annret = merged_df.groupby(by=['CUSIP', 'YEAR']).apply(lambda x: np.exp(np.sum(np.l
sigmae = merged_df.groupby(by=['CUSIP','YEAR'])['RET'].std()*np.sqrt(250)
E = merged_df.groupby(by=['CUSIP','YEAR'])['E'].first()

lagged_df = pd.DataFrame()

# Lag the 'annret' column by 1 year
lagged_df['annret'] = annret

# Lag the 'sigmae' column by 1 year
lagged_df['sigmae'] = sigmae

# Lag the 'mrkt_cap' column by 1 year
lagged_df['E'] = E
```

In [ ]:
```
merged_df = pd.merge(lagged_df, merged_df, on = ['CUSIP', 'YEAR'], how = 'inner')
display(merged_df)
```

	CUSIP	YEAR	annret	sigmae	E_x	DATE	PRC	RET	SHROUT	
0	000307	2015	1.115584	0.502147	394440.84	2014-10-17	19.08	-0.016495	20673.0	3
1	000307	2015	1.115584	0.502147	394440.84	2014-10-31	21.78	0.027358	20673.0	4.
2	000307	2015	1.115584	0.502147	394440.84	2014-12-02	31.61	0.068266	21081.0	6
3	000307	2015	1.115584	0.502147	394440.84	2014-12-09	32.41	0.045484	21081.0	6
4	000307	2015	1.115584	0.502147	394440.84	2014-12-19	29.50	-0.009735	21081.0	6
...	...	...	...	...	...	...	...	...	...	
2859059	U72603	2017	1.188844	0.539431	143033.92	2016-10-24	3.26	-0.009119	48855.0	1.
2859060	U72603	2017	1.188844	0.539431	143033.92	2016-11-01	3.51	0.005731	48902.0	1
2859061	U72603	2017	1.188844	0.539431	143033.92	2016-11-08	3.49	-0.002857	48902.0	1
2859062	U72603	2017	1.188844	0.539431	143033.92	2016-12-23	4.20	0.007194	48902.0	2
2859063	U72603	2017	1.188844	0.539431	143033.92	2016-12-27	4.20	0.000000	48902.0	2

2859064 rows × 20 columns

In [ ]:
```
merged_df = merged_df.drop('E_y', axis =1)
merged_df['E'] = merged_df['E_x']
merged_df = merged_df.drop('E_x', axis =1)
```

Method 1: Naive Computation using the KMV model

In [ ]:
```
merged_df = merged_df.groupby('YEAR').apply(lambda x: x.sample(n=200, random_state=

display(merged_df)
```

	CUSIP	YEAR	annret	sigmae	DATE	PRC	RET	SHROUT	DATADATE
0	30231G	1971	0.943953	0.221137	1970-08-04	60.6250	-0.002058	223663.0	1970-12-31
1	691449	1971	0.962330	0.439959	1970-04-15	14.0000	0.018182	935.0	1970-12-31
2	109017	1971	1.118476	0.496185	1970-10-22	8.1250	0.000000	1300.0	1970-10-31
3	963303	1971	0.932060	0.252582	1970-12-24	7.8750	0.000000	2427.0	1970-06-30
4	445582	1971	1.163828	0.408111	1970-10-23	17.0000	-0.014493	5212.0	1970-12-31
...	...	...	...	...	...	...	...	...	...
10195	410120	2021	0.852969	0.726875	2020-03-11	21.3200	-0.136842	88652.0	2020-12-31
10196	423452	2021	0.945013	0.728479	2020-09-17	16.5600	-0.004808	107478.0	2020-09-30
10197	70450Y	2021	1.052045	0.520348	2020-12-24	238.6400	-0.003341	1171692.0	2020-12-31
10198	44183U	2021	3.305110	4.908658	2020-04-07	0.1081	-0.019056	87007.0	2020-12-31
10199	31154R	2021	1.013286	0.419698	2020-02-27	6.2000	-0.050536	30079.0	2020-12-31

10200 rows × 19 columns

In [ ]:

```
T = 1
# Calculating Naive σD (three versions)
sigmaD_naive_1 = 0.05 + 0.25 * merged_df['sigmae']
sigmaD_naive_2 = 0.05 + 0.5 * merged_df['sigmae']
sigmaD_naive_3 = 0.25 * merged_df['sigmae']

# Calculating Naive σV for each version
Naive_sigma_1 = (merged_df['E'] / (merged_df['E'] + merged_df['F'])) * merged_df['s
Naive_sigma_2 = (merged_df['E'] / (merged_df['E'] + merged_df['F'])) * merged_df['s
Naive_sigma_3 = (merged_df['E'] / (merged_df['E'] + merged_df['F'])) * merged_df['s


merged_df['Naive_sigma_3'] = Naive_sigma_3
merged_df['Naive_sigma_2'] = Naive_sigma_2
merged_df['Naive_sigma_1'] = Naive_sigma_1


# Computing DD_naive for each version
merged_df['DD_naive_1'] = np.log(merged_df['E'] / merged_df['F']) + (merged_df['ann
```

```
merged_df['DD_naive_2'] = np.log(merged_df['E'] / merged_df['F']) + (merged_df['ann
merged_df['DD_naive_3'] = np.log(merged_df['E'] / merged_df['F']) + (merged_df['ann
```

In [ ]:

```
# Calculate DD and PD for the filtered DataFrame
for i in range(1, 4):
    naive_sigma = merged_df[f'Naive_sigma_{i}']
    merged_df[f'DD_naive_{i}'] = np.log(merged_df['E'] / merged_df['F']) + (merged_
    merged_df[f'PD_naive_{i}'] = norm.cdf(-merged_df[f'DD_naive_{i}'])

# Display the results
print('DD and PD for Scenario 1:')
print(merged_df[['DD_naive_1', 'PD_naive_1']])
print('DD and PD for Scenario 2:')
print(merged_df[['DD_naive_2', 'PD_naive_2']])
print('DD and PD for Scenario 3:')
print(merged_df[['DD_naive_3', 'PD_naive_3']])
```

```
DD and PD for Scenario 1:
       DD_naive_1  PD_naive_1
0        3.600261    0.000159
1       -1.719682    0.957255
2        2.058263    0.019782
3        2.997802    0.001360
4        4.020695    0.000029
...           ...         ...
10195   -2.687702    0.996403
10196   -0.187328    0.574298
10197    1.847397    0.032345
10198   -1.240055    0.892522
10199   -0.682742    0.752615

[10200 rows x 2 columns]
DD and PD for Scenario 2:
       DD_naive_2  PD_naive_2
0        0.525011    0.299788
1       -4.221202    0.999988
2       -0.527227    0.700982
3        0.061921    0.475313
4        3.418057    0.000315
...           ...         ...
10195   -4.384083    0.999994
10196   -1.944859    0.974104
10197   -0.452055    0.674385
10198   -2.756419    0.997078
10199   -3.369191    0.999623

[10200 rows x 2 columns]
DD and PD for Scenario 3:
       DD_naive_3    PD_naive_3
0       11.524314  4.975096e-31
1        1.032152  1.510005e-01
2        4.438361  4.532334e-06
3        9.347519  4.486392e-21
4        4.392815  5.594625e-06
...           ...           ...
10195   -1.661942  9.517378e-01
10196    0.838499  2.008751e-01
10197    3.806663  7.042724e-05
10198   -1.146981  8.743053e-01
10199    2.444907  7.244476e-03

[10200 rows x 2 columns]
```

In [ ]:

```python
def compute_column_statistics(df, column_name):
    if column_name not in df.columns:
        return f"Column '{column_name}' not found in the DataFrame."

    # Select the column
    column = df[column_name]

    # Calculating the required statistics
    stats = {
        'Statistic': ["Scenario",'Number of Observations', 'Mean', '25th Percentile
```

```
                          '75th Percentile', 'Standard Deviation', 'Minimum', 'Maximum'
            'Value': [
                column_name,
                column.count(),
                column.mean(),
                column.quantile(0.25),
                column.quantile(0.5),
                column.quantile(0.75),
                column.std(),
                column.min(),
                column.max()
            ]
        }

        return pd.DataFrame(stats)
```

In [ ]:

```
stats = compute_column_statistics(merged_df, 'DD_naive_1')
print(stats)
stats = compute_column_statistics(merged_df, 'DD_naive_2')
print(stats)
stats = compute_column_statistics(merged_df, 'DD_naive_3')
print(stats)


stats = compute_column_statistics(merged_df, 'PD_naive_1')
print(stats)
stats = compute_column_statistics(merged_df, 'PD_naive_2')
print(stats)
stats = compute_column_statistics(merged_df, 'PD_naive_3')
print(stats)
```

```
                 Statistic      Value
0                 Scenario  DD_naive_1
1   Number of Observations      10191
2                     Mean        inf
3          25th Percentile  -0.647363
4  50th Percentile (Median)   1.738526
5          75th Percentile   4.335258
6       Standard Deviation        NaN
7                  Minimum -10.715617
8                  Maximum        inf
                 Statistic      Value
0                 Scenario  DD_naive_2
1   Number of Observations      10191
2                     Mean        inf
3          25th Percentile   -2.88275
4  50th Percentile (Median)  -0.856364
5          75th Percentile   1.433874
6       Standard Deviation        NaN
7                  Minimum -11.448668
8                  Maximum        inf
                 Statistic      Value
0                 Scenario  DD_naive_3
1   Number of Observations      10191
2                     Mean        inf
3          25th Percentile   1.104089
4  50th Percentile (Median)     5.0906
5          75th Percentile  12.606675
6       Standard Deviation        NaN
7                  Minimum -10.579371
8                  Maximum        inf
                 Statistic      Value
0                 Scenario  PD_naive_1
1   Number of Observations      10191
2                     Mean   0.316063
3          25th Percentile   0.000007
4  50th Percentile (Median)   0.041059
5          75th Percentile   0.741301
6       Standard Deviation   0.400798
7                  Minimum        0.0
8                  Maximum        1.0
                 Statistic      Value
0                 Scenario  PD_naive_2
1   Number of Observations      10191
2                     Mean   0.599286
3          25th Percentile   0.075804
4  50th Percentile (Median)   0.804102
5          75th Percentile   0.998029
6       Standard Deviation    0.42024
7                  Minimum        0.0
8                  Maximum        1.0
                 Statistic      Value
0                 Scenario  PD_naive_3
1   Number of Observations      10191
2                     Mean   0.187142
3          25th Percentile        0.0
4  50th Percentile (Median)        0.0
```

```
5           75th Percentile    0.134777
6        Standard Deviation    0.346144
7                   Minimum         0.0
8                   Maximum         1.0
```

```
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\panda
s\core\nanops.py:1010: RuntimeWarning: invalid value encountered in subtract
  sqr = _ensure_numeric((avg - values) ** 2)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\panda
s\core\nanops.py:1010: RuntimeWarning: invalid value encountered in subtract
  sqr = _ensure_numeric((avg - values) ** 2)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\panda
s\core\nanops.py:1010: RuntimeWarning: invalid value encountered in subtract
  sqr = _ensure_numeric((avg - values) ** 2)
```

In [ ]:

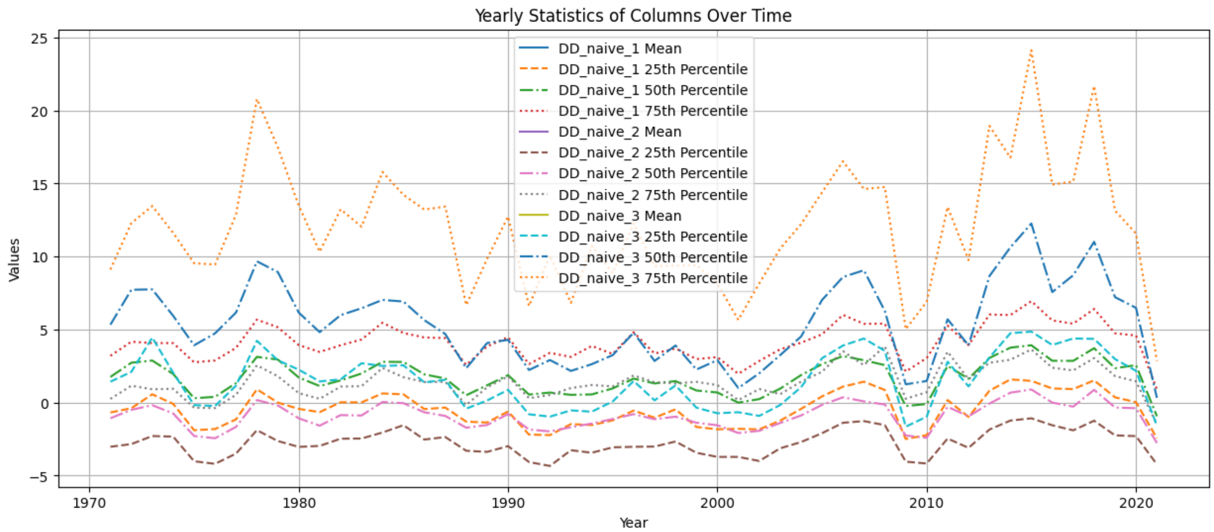
```python
correlation_df = pd.DataFrame()

correlation_df['Date_method1'] = merged_df['DATE']
correlation_df['DD_method1_n1'] = merged_df['DD_naive_1']
correlation_df['DD_method1_n2'] = merged_df['DD_naive_2']
correlation_df['DD_method1_n3'] = merged_df['DD_naive_3']

correlation_df['PD_method1_n1'] = merged_df['PD_naive_1']
correlation_df['PD_method1_n2'] = merged_df['PD_naive_2']
correlation_df['PD_method1_n3'] = merged_df['PD_naive_3']
```

In [ ]:

```python
columns_to_plot = ['DD_naive_1', 'DD_naive_2', 'DD_naive_3']

# Plotting
plt.figure(figsize=(15, 5 * len(columns_to_plot)))
# Prepare the plot
plt.figure(figsize=(15, 6))

# Plotting each column's statistics
for column in columns_to_plot:
    # Group by 'YEAR' and calculate the required statistics for each column
    stats = merged_df.groupby('YEAR')[column].agg(['mean', lambda x: x.quantile(0.2
                                                  lambda x: x.quantile(0.50), lambda x:

    # Renaming the columns for clarity
    stats.columns = ['Mean', '25th Percentile', '50th Percentile', '75th Percentile

    # Plot each statistic
    plt.plot(stats['Mean'], label=f'{column} Mean')
    plt.plot(stats['25th Percentile'], label=f'{column} 25th Percentile', linestyle
    plt.plot(stats['50th Percentile'], label=f'{column} 50th Percentile', linestyle
    plt.plot(stats['75th Percentile'], label=f'{column} 75th Percentile', linestyle

# Adding labels and title
plt.title('Yearly Statistics of Columns Over Time')
plt.xlabel('Year')
plt.ylabel('Values')
plt.legend()
plt.grid(True)
```

```
# Show the plot
plt.show()
```

<Figure size 1500x1500 with 0 Axes>



Method 2: Directly Solving for the Unknowns

In [ ]:
```
subset_df = merged_df[['F', 'E', 'sigmae', 'annret', 'CUSIP', 'YEAR', "DATE"]]
display(subset_df)
```

	F	E	sigmae	annret	CUSIP	YEAR	DATE
0	2.571517e+09	1.333378e+07	0.221137	0.943953	30231G	1971	1970-08-04
1	4.466350e+07	2.127125e+04	0.439959	0.962330	691449	1971	1970-04-15
2	1.056000e+06	1.771250e+04	0.496185	1.118476	109017	1971	1970-10-22
3	4.049000e+06	2.411831e+04	0.252582	0.932060	963303	1971	1970-12-24
4	9.550000e+04	1.074975e+05	0.408111	1.163828	445582	1971	1970-10-23
...	...	...	...	...	...	...	...
10195	1.927108e+09	3.768396e+06	0.726875	0.852969	410120	2021	2020-03-11
10196	2.686705e+08	4.995069e+06	0.728479	0.945013	423452	2021	2020-09-17
10197	4.934500e+09	1.311649e+08	0.520348	1.052045	70450Y	2021	2020-12-24
10198	1.645000e+05	1.140667e+04	4.908658	3.305110	44183U	2021	2020-04-07
10199	2.534055e+08	2.015293e+05	0.419698	1.013286	31154R	2021	2020-02-27

10200 rows × 7 columns

In [ ]:
```
fed_df['DATE'] = pd.to_datetime(fed_df['DATE'])
fed_subset_df = pd.merge(subset_df, fed_df, on='DATE', how='inner')
display(fed_subset_df)
```

	F	E	sigmae	annret	CUSIP	YEAR	DATE	DTB3	
0	2.571517e+09	1.333378e+07	0.221137	0.943953	30231G	1971	1970-08-04	6.46	0.0625
1	7.836400e+07	2.535520e+05	0.539241	1.267152	27637F	1971	1970-08-04	6.46	0.0625
2	2.331100e+07	7.609250e+04	0.279248	1.150950	054595	1971	1970-08-04	6.46	0.0625
3	4.466350e+07	2.127125e+04	0.439959	0.962330	691449	1971	1970-04-15	6.38	0.0618
4	1.056000e+06	1.771250e+04	0.496185	1.118476	109017	1971	1970-10-22	5.78	0.0561
...	...	...	...	...	...	...	...	...	
10101	1.635150e+07	1.119087e+05	0.686622	0.719641	829399	2021	2020-07-22	0.13	0.0012
10102	1.927108e+09	3.768396e+06	0.726875	0.852969	410120	2021	2020-03-11	0.41	0.0040
10103	2.686705e+08	4.995069e+06	0.728479	0.945013	423452	2021	2020-09-17	0.09	0.0009
10104	4.934500e+09	1.311649e+08	0.520348	1.052045	70450Y	2021	2020-12-24	0.09	0.0009
10105	1.645000e+05	1.140667e+04	4.908658	3.305110	44183U	2021	2020-04-07	0.14	0.0013

10106 rows × 9 columns

In [ ]:

```
fed_subset_df['V2'] = fed_subset_df['E'] + fed_subset_df['F']
fed_subset_df['SIGMAV2'] = (fed_subset_df['E'] / (fed_subset_df['E'] + fed_subset_d
                (fed_subset_df['F'] / (fed_subset_df['E'] + fed_subset_df['F'])) *

# Drop rows with missing E_x, F, or sigmae
fed_subset_df.dropna(subset=['E', 'F', 'sigmae'], inplace=True)

#Sorting Data:
fed_subset_df.sort_values(by=['CUSIP', 'YEAR'], inplace=True)

#Financial Modeling:
# Define the equations as functions
def equations(vars, row):
    V2, SIGMAV2 = vars
    RF = row['r']  # Make sure RF is a column in your DataFrame
    F = row['F']
    E = row['E']

    eq1 = V2 * norm.cdf((np.log(V2 / F) + RF + SIGMAV2**2 * 0.5) / SIGMAV2) - \
          np.exp(-1 * RF) * F * norm.cdf((np.log(V2 / F) + RF + SIGMAV2**2 * 0.5) /
```

```
    eq2 = (V2 / E) * norm.cdf((np.log(V2 / F) + RF + SIGMAV2**2 * 0.5) / SIGMAV2) *
    return [eq1, eq2]

# Apply the solver for each row
def solve_row(row):
    initial_guesses = [row['V2'], row['SIGMAV2']]
    solution = fsolve(equations, initial_guesses, args=(row,))
    return pd.Series(solution, index=['V2_solved', 'SIGMAV2_solved'])

solve_results = fed_subset_df.apply(solve_row, axis=1)
fed_subset_df = pd.concat([fed_subset_df, solve_results], axis=1)

# Calculating DD and PD
fed_subset_df['DD_direct'] = (np.log(fed_subset_df['V2_solved'] / fed_subset_df['F'
                              + (fed_subset_df['annret'] - fed_subset_df['SIGMAV2_s
fed_subset_df['PD_direct'] = norm.cdf(-fed_subset_df['DD_direct'])

# Drop rows with missing DD_direct
fed_subset_df.dropna(subset=['DD_direct'], inplace=True)
```

```
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The number of calls to function has re
ached maxfev = 600.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\922477194.py:19: RuntimeWar
ning: divide by zero encountered in scalar divide
  eq1 = V2 * norm.cdf((np.log(V2 / F) + RF + SIGMAV2**2 * 0.5) / SIGMAV2) - \
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\922477194.py:20: RuntimeWar
ning: divide by zero encountered in scalar divide
  np.exp(-1 * RF) * F * norm.cdf((np.log(V2 / F) + RF + SIGMAV2**2 * 0.5) / SIGMAV2
- SIGMAV2) - E
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\922477194.py:21: RuntimeWar
ning: divide by zero encountered in scalar divide
  eq2 = (V2 / E) * norm.cdf((np.log(V2 / F) + RF + SIGMAV2**2 * 0.5) / SIGMAV2) * SI
GMAV2 - row['sigmae']
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last five Jacobian evaluations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\922477194.py:19: RuntimeWar
ning: invalid value encountered in log
  eq1 = V2 * norm.cdf((np.log(V2 / F) + RF + SIGMAV2**2 * 0.5) / SIGMAV2) - \
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\922477194.py:20: RuntimeWar
ning: invalid value encountered in log
  np.exp(-1 * RF) * F * norm.cdf((np.log(V2 / F) + RF + SIGMAV2**2 * 0.5) / SIGMAV2
- SIGMAV2) - E
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\922477194.py:21: RuntimeWar
ning: invalid value encountered in log
  eq2 = (V2 / E) * norm.cdf((np.log(V2 / F) + RF + SIGMAV2**2 * 0.5) / SIGMAV2) * SI
GMAV2 - row['sigmae']
```

In [ ]:
```
display(fed_subset_df)
```

	F	E	sigmae	annret	CUSIP	YEAR	DATE	DTB3	r
4336	5731500.0	7.268079e+03	0.691222	0.788149	000360	1992	1991-07-10	5.58	0.054299
3261	2019000.0	1.160600e+04	1.127327	0.933612	000771	1987	1986-07-11	5.75	0.055908
3369	2272000.0	9.080500e+03	0.688497	1.370134	000771	1988	1987-05-26	5.61	0.054583
2117	0.0	3.127500e+03	0.000000	1.000000	000774	1981	1980-03-20	15.02	0.139936
6492	6037000.0	2.268112e+04	3.523572	0.978110	000788	2003	2002-02-15	1.71	0.016955
...	...	...	...	...	...	...	...	...	...
7090	0.0	1.795494e+05	0.575076	1.040731	989855	2006	2005-01-28	2.41	0.023814
8667	0.0	1.407949e+06	0.745561	1.498249	98986T	2014	2013-03-12	0.10	0.001000
9971	724450000.0	5.671244e+06	0.411921	1.335924	98986T	2021	2020-10-29	0.09	0.000900
2228	78500.0	5.156250e+03	0.392840	0.996914	989905	1982	1981-09-23	14.20	0.132781
8567	708000.0	1.464891e+05	0.730746	0.789225	U72603	2014	2013-04-03	0.06	0.000600

10097 rows × 15 columns

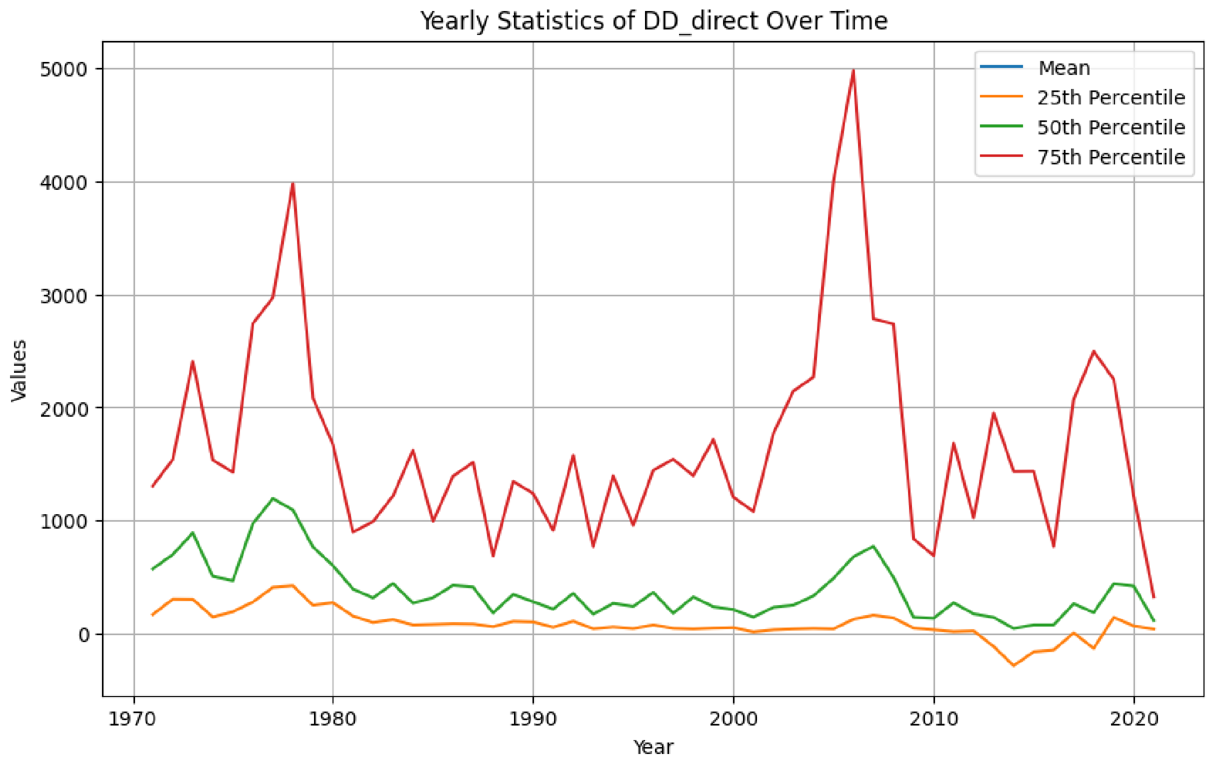
In [ ]:
```
stats = compute_column_statistics(fed_subset_df, 'DD_direct')
print(stats)
stats = compute_column_statistics(fed_subset_df, 'PD_direct')
print(stats)
```

```
                   Statistic         Value
0                   Scenario     DD_direct
1     Number of Observations         10097
2                       Mean           inf
3            25th Percentile     69.114918
4  50th Percentile (Median)     349.275254
5            75th Percentile   1565.904421
6         Standard Deviation           NaN
7                    Minimum -33664.620319
8                    Maximum           inf
                   Statistic      Value
0                   Scenario  PD_direct
1     Number of Observations      10097
2                       Mean   0.055512
3            25th Percentile        0.0
4  50th Percentile (Median)        0.0
5            75th Percentile        0.0
6         Standard Deviation   0.225954
7                    Minimum        0.0
8                    Maximum        1.0
```

```
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\panda
s\core\nanops.py:1010: RuntimeWarning: invalid value encountered in subtract
  sqr = _ensure_numeric((avg - values) ** 2)
```

In [ ]:

```
# Group by 'YEAR' and calculate the required statistics
stats = fed_subset_df.groupby('YEAR')['DD_direct'].agg(['mean', 'std', lambda x: x.
                                                       lambda x: x.quantile(0.50), l

# Renaming the columns for clarity
stats.columns = ['Mean', 'Standard Deviation', '25th Percentile', '50th Percentile'

# Plotting
plt.figure(figsize=(10, 6))
stats[['Mean', '25th Percentile', '50th Percentile', '75th Percentile']].plot(ax=pl
plt.title('Yearly Statistics of DD_direct Over Time')
plt.xlabel('Year')
plt.ylabel('Values')
plt.legend()
plt.grid(True)
plt.show()
```



In [ ]:
```
correlation_df['Date_method2'] = fed_subset_df['DATE']

correlation_df['DD_method2'] = fed_subset_df['DD_direct']
correlation_df['DD_method2'] = fed_subset_df['PD_direct']
```

Method 3: Computing Distance Default Using the KMV Model (iterative Method)

In [ ]:
```
method3_df = merged_df[['CUSIP', 'YEAR', 'sigmae', 'annret', 'E', 'F', 'DATE', 'RET
method3_df = pd.merge(method3_df, fed_df, on='DATE', how='inner')
method3_df = method3_df.groupby('YEAR').sample(n=100, random_state=2)
display(method3_df)
```

	CUSIP	YEAR	sigmae	annret	E	F	DATE	RET	DTB
112	163267	1971	0.627077	0.805713	37004.625	1.047800e+07	1970-05-11	-0.028571	6.7
29	29382L	1971	0.575252	0.777764	17933.500	5.201000e+06	1970-01-28	0.015625	7.9
181	92220P	1971	0.643227	0.747981	185584.000	2.372150e+07	1970-02-03	0.004902	7.7
179	276191	1971	0.520578	1.012928	188784.000	4.084270e+08	1970-01-23	-0.065421	7.8
192	806500	1971	0.327023	0.845576	182595.000	1.527500e+08	1970-03-23	0.005618	6.1
...	...	...	...	...	...	...	...	...	
9958	03071H	2021	0.416512	0.917635	1280947.080	3.305000e+05	2020-11-25	-0.014102	0.0
9989	23311P	2021	1.667434	1.000433	5314498.300	3.126500e+09	2020-10-21	0.021245	0.1
10004	83413U	2021	0.283100	1.039370	873534.870	3.357255e+08	2020-06-26	-0.009317	0.1
10103	423452	2021	0.728479	0.945013	4995068.820	2.686705e+08	2020-09-17	-0.004808	0.0
9920	19200A	2021	0.864964	0.636959	31514.097	5.760000e+05	2020-08-14	-0.086363	0.1

5100 rows × 10 columns

In [ ]:
```
import pandas as pd
import numpy as np
from scipy.stats import norm
from scipy.optimize import fsolve

# Assuming 'crsp' is a pre-loaded DataFrame containing daily stock data
daily_stock_data = dsf_df
daily_stock_data['RET'] = pd.to_numeric(daily_stock_data['RET'], errors='coerce')

def calculate_initial_sigmaE(stock_data):
    """ Calculate the initial sigmaE (volatility) based on stock data returns. """
    return stock_data['RET'].std()

def iterative_black_scholes(E, F, r, sigmaE_initial, T=1, max_iterations=1000, tole
    """ Iterative Black-Scholes model implementation. """
    sigmaV = max(sigmaE_initial, min_sigmaV)  # Initial estimate of sigmaV
    V = E + F  # Improved initial guess for firm value

    for _ in range(max_iterations):
        sigmaV_old = sigmaV
```

```
        def equation(V_guess):
            """ Equation to solve in the Black-Scholes model. """
            sigmaV_safe = max(sigmaV, min_sigmaV)  # Ensure sigmaV is not too small
            V_safe = max(V_guess, F + 1e-3)  # Avoid V being too close to F
            d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe
            d2 = d1 - sigmaV_safe * np.sqrt(T)
            return V_safe * norm.cdf(d1) - np.exp(-r * T) * F * norm.cdf(d2) - E

        V = fsolve(equation, V)[0]  # Use V as the initial guess
        d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
        sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update

        if np.abs(sigmaV - sigmaV_old) < tolerance:
            break

    return V, sigmaV

# Assuming 'final_data' is a pre-loaded DataFrame
# Sample 100 firms for each year
selected_data = method3_df
all_years_data = []  # List to store data for each year

for year in range(1970, 2021):
    yearly_data = selected_data[selected_data['YEAR'] == year]

    if not yearly_data.empty:
        sampled_firms = yearly_data.sample(n=100, random_state=1)

        for index, row in sampled_firms.iterrows():
            prev_year_stock_data = daily_stock_data[(daily_stock_data['CUSIP'] == r
                                                     (daily_stock_data['DATE'].dt.ye

            if not prev_year_stock_data.empty and row['E'] > 0 and row['F'] > 0 and
                initial_sigmaE = calculate_initial_sigmaE(prev_year_stock_data)
                V, sigmaV = iterative_black_scholes(row['E'], row['F'], row['r'], i

                d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sig
                DD = d1
                PD = norm.cdf(-DD)

                sampled_firms.at[index, 'V_method3'] = V
                sampled_firms.at[index, 'sigmaV_method3'] = sigmaV
                sampled_firms.at[index, 'DD_method3'] = DD
                sampled_firms.at[index, 'PD_method3'] = PD
            else:
                sampled_firms.at[index, 'V_method3'] = np.nan
                sampled_firms.at[index, 'sigmaV_method3'] = np.nan
                sampled_firms.at[index, 'DD_method3'] = np.nan
                sampled_firms.at[index, 'PD_method3'] = np.nan
        all_years_data.append(sampled_firms)
# Concatenate all yearly data into a single DataFrame
final_results_df = pd.concat(all_years_data)
```

```
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
```

```
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
```

```
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
```

```
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
```

```
(1))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
```

```
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
```

```
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
```

```
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
```

```
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
```

```
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
```

```
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
```

```
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
```

```
(1))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
```

```
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
```

```
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
```

```
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
```

```
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
```

```
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
```

```
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
```

```
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
```

```
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
```

```
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
```

```
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\scipy
\optimize\_minpack_py.py:177: RuntimeWarning: The iteration is not making good progr
ess, as measured by the
  improvement from the last ten iterations.
  warnings.warn(msg, RuntimeWarning)
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:32: RuntimeWa
rning: divide by zero encountered in scalar divide
  sigmaV = max(E / (V * norm.cdf(d1)) * sigmaE_initial, min_sigmaV)  # Update with a
safe value
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: invalid value encountered in scalar divide
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:26: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V_safe / F) + (r + sigmaV_safe**2 / 2) * T) / (sigmaV_safe * np.sqrt
(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:31: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / F) + (r + sigmaV**2 / 2) * T) / (sigmaV * np.sqrt(T))
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1134509438.py:58: RuntimeWa
rning: overflow encountered in scalar power
  d1 = (np.log(V / row['F']) + (row['r'] + (sigmaV ** 2) / 2)) / (sigmaV * np.sqrt
(1))
```

In [ ]:

```
display(final_results_df)
```

	CUSIP	YEAR	sigmae	annret	E	F	DATE	RET	DTB
146	879369	1971	0.393465	1.033604	6.144875e+03	4.133000e+06	1970-04-03	0.010417	6.3
62	218867	1971	0.511753	0.784215	1.099800e+05	9.750000e+06	1970-06-16	-0.010050	6.7
57	53630P	1971	0.390423	1.010584	1.425000e+04	1.295000e+05	1970-03-30	0.008000	6.3
36	934436	1971	0.321068	1.164633	2.340745e+05	1.064595e+08	1970-02-16	0.004310	6.8
141	677194	1971	0.229905	0.948825	3.021750e+04	1.199500e+07	1970-09-11	-0.008197	6.3
...	...	...	...	...	...	...	...	...	
9844	23204G	2020	0.366594	1.131864	6.061638e+05	1.376052e+09	2019-07-03	0.013453	2.1
9878	16359R	2020	0.164891	1.077933	4.434203e+06	1.276080e+08	2019-04-25	-0.004001	2.3
9714	482738	2020	0.288535	1.102049	1.871992e+05	5.827500e+06	2019-06-17	0.012146	2.1
9715	36269P	2020	0.633109	0.689923	1.423750e+05	9.216500e+06	2019-06-17	-0.038409	2.1
9834	87265H	2020	0.261109	1.157823	1.615415e+06	6.700550e+08	2019-07-01	0.005848	2.1

5000 rows × 14 columns

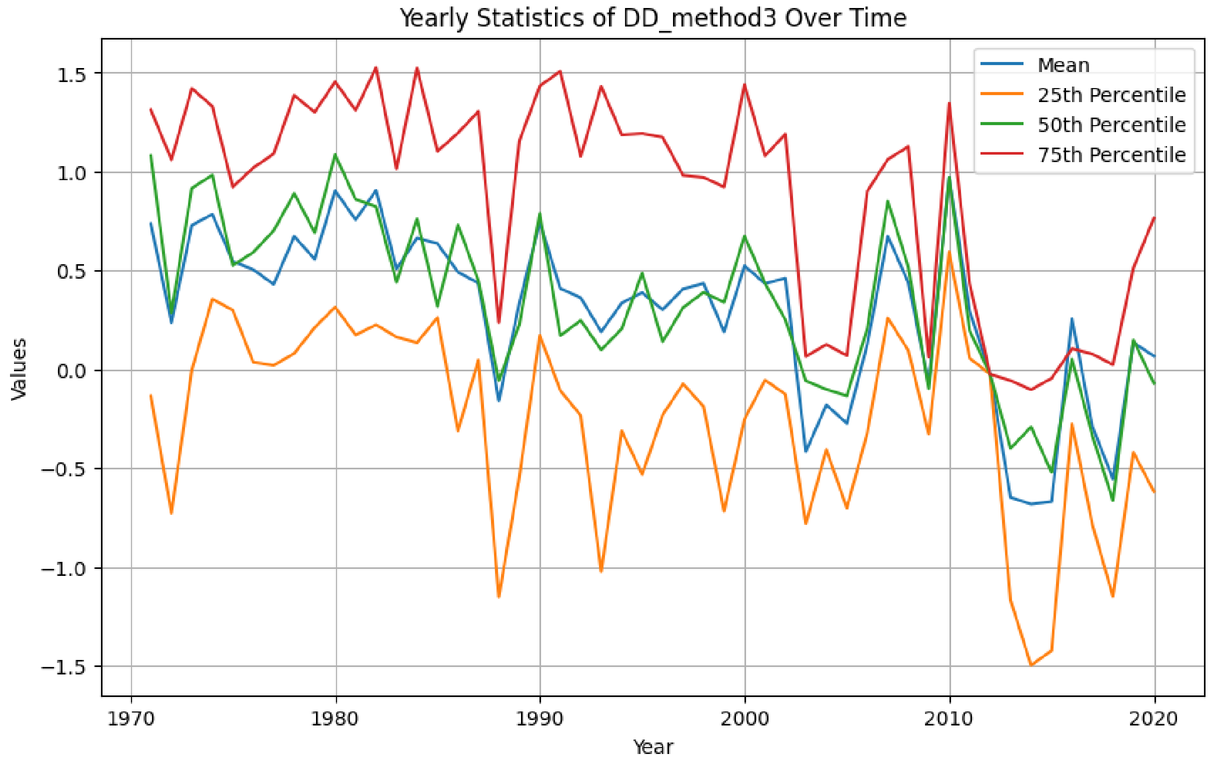
In [ ]:

```
stats = compute_column_statistics(final_results_df, 'PD_method3')
print(stats)
stats = compute_column_statistics(final_results_df, 'DD_method3')
print(stats)
```

```
                     Statistic       Value
0                     Scenario  PD_method3
1       Number of Observations        1156
2                         Mean    0.374004
3              25th Percentile    0.121692
4  50th Percentile (Median)      0.340201
5              75th Percentile    0.564856
6           Standard Deviation    0.276408
7                      Minimum    0.022959
8                      Maximum    0.998301
                     Statistic       Value
0                     Scenario  DD_method3
1       Number of Observations        1156
2                         Mean    0.413745
3              25th Percentile   -0.163293
4  50th Percentile (Median)      0.411914
5              75th Percentile    1.166572
6           Standard Deviation     0.94239
7                      Minimum   -2.929226
8                      Maximum    1.996154
```

In [ ]:

```python
# Group by 'YEAR' and calculate the required statistics
stats = final_results_df.groupby('YEAR')['DD_method3'].agg(['mean', 'std', lambda x
                                                       lambda x: x.quantile(0.50), l

# Renaming the columns for clarity
stats.columns = ['Mean', 'Standard Deviation', '25th Percentile', '50th Percentile'

# Plotting
plt.figure(figsize=(10, 6))
stats[['Mean', '25th Percentile', '50th Percentile', '75th Percentile']].plot(ax=pl
plt.title('Yearly Statistics of DD_method3 Over Time')
plt.xlabel('Year')
plt.ylabel('Values')
plt.legend()
plt.grid(True)
plt.show()
```



In [ ]:
```
display(correlation_df)
```

	DD_method1_n1	DD_method1_n2	DD_method1_n3	PD_method1_n1	PD_method1_n2
0	3.600261	0.525011	11.524314	0.000159	0.299788
1	-1.719682	-4.221202	1.032152	0.957255	0.999988
2	2.058263	-0.527227	4.438361	0.019782	0.700982
3	2.997802	0.061921	9.347519	0.001360	0.475313
4	4.020695	3.418057	4.392815	0.000029	0.000315
...	...	...	...	...	...
10195	-2.687702	-4.384083	-1.661942	0.996403	0.999994
10196	-0.187328	-1.944859	0.838499	0.574298	0.974104
10197	1.847397	-0.452055	3.806663	0.032345	0.674385
10198	-1.240055	-2.756419	-1.146981	0.892522	0.997078
10199	-0.682742	-3.369191	2.444907	0.752615	0.999623

10200 rows × 9 columns

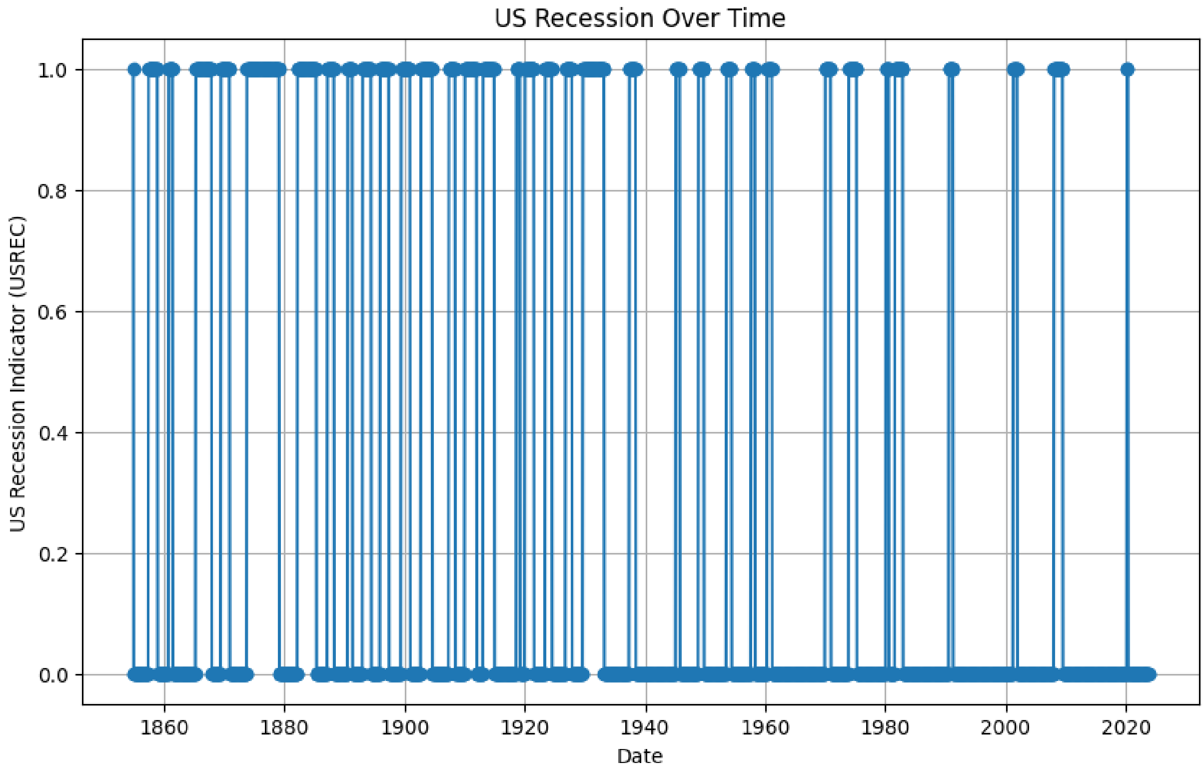
In [ ]:
```
correlation_df['Date_method3'] = final_results_df['DATE']

correlation_df['DD_method3'] = final_results_df['DD_method3']
correlation_df['PD_method3'] = final_results_df['PD_method3']
```

In [ ]:
```
matrix = correlation_df.corr()
print(matrix)
```

```
               DD_method1_n1  DD_method1_n2  DD_method1_n3  PD_method1_n1  \
DD_method1_n1       1.000000       0.964433       0.836443      -0.859529
DD_method1_n2       0.964433       1.000000       0.759461      -0.798706
DD_method1_n3       0.836443       0.759461       1.000000      -0.665331
PD_method1_n1      -0.859529      -0.798706      -0.665331       1.000000
PD_method1_n2      -0.828257      -0.848931      -0.679147       0.723560
PD_method1_n3      -0.752549      -0.665909      -0.567695       0.846716
DD_method2          0.044653       0.036393       0.038790      -0.049466
DD_method3         -0.034205      -0.044744      -0.008650       0.039778
PD_method3          0.028894       0.041945       0.003306      -0.039364

               PD_method1_n2  PD_method1_n3  DD_method2  DD_method3  \
DD_method1_n1      -0.828257      -0.752549    0.044653   -0.034205
DD_method1_n2      -0.848931      -0.665909    0.036393   -0.044744
DD_method1_n3      -0.679147      -0.567695    0.038790   -0.008650
PD_method1_n1       0.723560       0.846716   -0.049466    0.039778
PD_method1_n2       1.000000       0.501905   -0.053453    0.063311
PD_method1_n3       0.501905       1.000000   -0.032809    0.002410
DD_method2         -0.053453      -0.032809    1.000000   -0.159725
DD_method3          0.063311       0.002410   -0.159725    1.000000
PD_method3         -0.061984      -0.000506    0.161904   -0.987061

               PD_method3
DD_method1_n1    0.028894
DD_method1_n2    0.041945
DD_method1_n3    0.003306
PD_method1_n1   -0.039364
PD_method1_n2   -0.061984
PD_method1_n3   -0.000506
DD_method2       0.161904
DD_method3      -0.987061
PD_method3       1.000000
```

In [ ]:
```
file_path = 'USREC.csv'
NBER_df = pd.read_csv(file_path)

NBER_df['USREC'] = pd.to_numeric(NBER_df['USREC'], errors='coerce')
NBER_df['DATE'] = pd.to_datetime(NBER_df['DATE'], errors='coerce')
NBER_df['YEAR'] = (NBER_df['DATE']).dt.year
```

In [ ]:
```
# Plotting
plt.figure(figsize=(10, 6))
plt.plot(NBER_df['DATE'], NBER_df['USREC'], marker='o')
plt.title('US Recession Over Time')
plt.xlabel('Date')
plt.ylabel('US Recession Indicator (USREC)')
plt.grid(True)
plt.show()
```



In [ ]:
```
def create_year_column(df, date_column_name):
    if date_column_name not in df.columns:
        return f"Column '{date_column_name}' not found in the DataFrame."

    # Ensure the date column is in datetime format
    df[date_column_name] = pd.to_datetime(df[date_column_name], errors='coerce')

    # Create and return the 'Month_Year' column
    month_year_column = df[date_column_name].dt.to_period('Y')
    return month_year_column

correlation_df['Year1'] = create_year_column(correlation_df, 'Date_method1')
correlation_df['Year2'] = create_year_column(correlation_df, 'Date_method2')
correlation_df['Year3'] = create_year_column(correlation_df, 'Date_method3')
```

In [ ]:
```
def subset_on_matching_month_years(df, col1, col2, col3):
    # Check if columns exist in the DataFrame
    if not all(column in df.columns for column in [col1, col2, col3]):
        return "One or more specified columns do not exist in the DataFrame."

    # Subset the DataFrame based on the condition
    subset_df = df[(df[col1] == df[col2]) & (df[col1] == df[col3])]

    return subset_df

# Example usage
# Assuming 'data' is your DataFrame
# data = pd.read_csv('path_to_your_data.csv')  # Uncomment and modify this line as
subset_data = subset_on_matching_month_years(correlation_df, 'Year1', 'Year2', 'Yea
subset_data['YEAR'] = subset_data['Year1']
```

```
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\4122822034.py:15: SettingWi
thCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  subset_data['YEAR'] = subset_data['Year1']
```

In [ ]:
```
display(subset_data)
```

	Date_method1	DD_method1_n1	DD_method1_n2	DD_method1_n3	PD_method1_n1	P
1	1970-04-15	-1.719682	-4.221202	1.032152	9.572549e-01	
3	1970-12-24	2.997802	0.061921	9.347519	1.359673e-03	
11	1970-10-06	4.745885	1.253440	16.326992	1.037983e-06	
12	1970-03-17	2.482707	0.336141	4.315786	6.519421e-03	
13	1970-10-20	-1.465141	-3.036053	-0.387335	9.285588e-01	
...	...	...	...	...	...	
9870	2019-04-08	1.151851	-0.383707	2.118429	1.246912e-01	
9876	2019-10-07	3.239176	-0.596924	11.646125	5.993785e-04	
9879	2019-01-22	2.163308	1.254036	2.749774	1.525876e-02	
9881	2019-03-25	3.188615	0.448632	8.004445	7.147803e-04	
9882	2019-05-22	4.855164	1.564169	14.440377	6.014362e-07	

882 rows × 16 columns

In [ ]:
```
# Now perform the merge
NEBR_PD_df = pd.merge(NBER_df, subset_data, on='YEAR', how='inner')
#NEBR_PD_df = pd.merge(NBER_df, subset_data, on='YEAR', how='inner')
```

In [ ]:
```
# Filter the DataFrame for rows where 'USREC' equals 1
NBER1_df = NEBR_PD_df[NEBR_PD_df['USREC'] == 1]
NBER2_df = NEBR_PD_df[NEBR_PD_df['USREC'] == 0]
```

In [ ]:
```
# Compute descriptive statistics for the numeric columns
descriptive_stats = NBER1_df.describe()
print(descriptive_stats)
```

```
                                DATE   USREC         YEAR  \
count                           2124  2124.0  2124.000000   
mean   1979-09-20 10:55:35.593220288     1.0  1979.279190   
min              1970-01-01 00:00:00     1.0  1970.000000   
25%              1974-01-01 00:00:00     1.0  1974.000000   
50%              1975-01-01 00:00:00     1.0  1975.000000   
75%              1982-04-01 00:00:00     1.0  1982.000000   
max              2009-06-01 00:00:00     1.0  2009.000000   
std                              NaN     0.0    10.068195   

                        Date_method1  DD_method1_n1  DD_method1_n2  \
count                           2124    2124.000000    2124.000000   
mean   1979-10-04 01:26:06.101694912            inf            inf   
min              1970-01-06 00:00:00      -8.242555      -9.302220   
25%              1974-01-02 00:00:00      -0.884845      -3.218829   
50%              1975-04-25 00:00:00       1.163360      -1.319328   
75%              1982-06-30 00:00:00       3.273333       0.346769   
max              2009-07-28 00:00:00            inf            inf   
std                              NaN            NaN            NaN   

       DD_method1_n3  PD_method1_n1  PD_method1_n2  PD_method1_n3  \
count    2124.000000    2124.000000    2124.000000   2.124000e+03   
mean             inf       0.360359       0.686335   1.914273e-01   
min        -7.965216       0.000000       0.000000   0.000000e+00   
25%         1.010288       0.000531       0.364383   1.170158e-21   
50%         4.667230       0.122342       0.906470   1.532010e-06   
75%         9.488656       0.811800       0.999356   1.561787e-01   
max              inf       1.000000       1.000000   1.000000e+00   
std              NaN       0.406773       0.388491   3.521459e-01   

                        Date_method2   DD_method2  \
count                           2124  2124.000000   
mean   1979-09-23 21:41:01.016949120     0.008361   
min              1970-01-06 00:00:00     0.000000   
25%              1974-01-09 00:00:00     0.000000   
50%              1975-03-24 00:00:00     0.000000   
75%              1982-03-31 00:00:00     0.000000   
max              2009-08-20 00:00:00     1.000000   
std                              NaN     0.086805   

                        Date_method3   DD_method3   PD_method3  
count                           2124  2124.000000  2124.000000  
mean   1979-09-23 21:41:01.016949120     0.592746     0.318879  
min              1970-01-06 00:00:00    -2.104637     0.022959  
25%              1974-01-09 00:00:00     0.060619     0.111571  
50%              1975-03-24 00:00:00     0.584840     0.279329  
75%              1982-03-31 00:00:00     1.218217     0.475831  
max              2009-08-20 00:00:00     1.996154     0.982339  
std                              NaN     0.823806     0.241686  
```

```
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\panda
s\core\nanops.py:1010: RuntimeWarning: invalid value encountered in subtract
  sqr = _ensure_numeric((avg - values) ** 2)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\panda
s\core\nanops.py:1010: RuntimeWarning: invalid value encountered in subtract
  sqr = _ensure_numeric((avg - values) ** 2)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\panda
s\core\nanops.py:1010: RuntimeWarning: invalid value encountered in subtract
  sqr = _ensure_numeric((avg - values) ** 2)
```

In [ ]:

```
# Compute descriptive statistics for the numeric columns
descriptive_stats = NBER2_df.describe()
print(descriptive_stats)
```

```
                                DATE   USREC         YEAR  \
count                           8460  8460.0  8460.000000   
mean   1988-08-19 03:11:49.787234048     0.0  1988.171040   
min              1970-12-01 00:00:00     0.0  1970.000000   
25%              1977-03-01 00:00:00     0.0  1977.000000   
50%              1985-09-01 00:00:00     0.0  1985.000000   
75%              1998-07-01 00:00:00     0.0  1998.000000   
max              2019-12-01 00:00:00     0.0  2019.000000   
std                              NaN     0.0    13.474453   

                        Date_method1  DD_method1_n1  DD_method1_n2  \
count                           8460    8460.000000    8460.000000   
mean   1988-09-04 18:36:05.106382976            inf            inf   
min              1970-01-06 00:00:00      -8.242555      -9.302220   
25%              1977-04-19 00:00:00      -0.460349      -2.794069   
50%              1985-11-07 00:00:00       2.059778      -0.596924   
75%              1998-08-03 00:00:00       4.365100       1.316732   
max              2019-12-30 00:00:00            inf            inf   
std                              NaN            NaN            NaN   

       DD_method1_n3  PD_method1_n1  PD_method1_n2  PD_method1_n3  \
count    8460.000000    8460.000000    8460.000000   8.460000e+03   
mean             inf       0.298666       0.587672   1.666692e-01   
min        -7.965216       0.000000       0.000000   0.000000e+00   
25%         1.569374       0.000006       0.093964   1.723771e-41   
50%         6.285398       0.019710       0.724721   1.635070e-10   
75%        13.441631       0.677367       0.997398   5.828044e-02   
max              inf       1.000000       1.000000   1.000000e+00   
std              NaN       0.392842       0.414768   3.313944e-01   

                        Date_method2   DD_method2  \
count                           8460  8460.000000   
mean   1988-08-29 02:45:37.021276416     0.041475   
min              1970-01-06 00:00:00     0.000000   
25%              1977-04-19 00:00:00     0.000000   
50%              1985-08-01 00:00:00     0.000000   
75%              1998-06-29 00:00:00     0.000000   
max              2019-12-31 00:00:00     1.000000   
std                              NaN     0.199159   

                        Date_method3   DD_method3   PD_method3  
count                           8460  8460.000000  8460.000000  
mean   1988-08-29 02:45:37.021276416     0.432210     0.368924  
min              1970-01-06 00:00:00    -2.167164     0.022959  
25%              1977-04-19 00:00:00    -0.151182     0.120857  
50%              1985-08-01 00:00:00     0.434836     0.331841  
75%              1998-06-29 00:00:00     1.170713     0.560084  
max              2019-12-31 00:00:00     1.996154     0.984889  
std                              NaN     0.930855     0.274313  
```

```
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\panda
s\core\nanops.py:1010: RuntimeWarning: invalid value encountered in subtract
  sqr = _ensure_numeric((avg - values) ** 2)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\panda
s\core\nanops.py:1010: RuntimeWarning: invalid value encountered in subtract
  sqr = _ensure_numeric((avg - values) ** 2)
c:\Users\morganhales\AppData\Local\Programs\Python\Python311\Lib\site-packages\panda
s\core\nanops.py:1010: RuntimeWarning: invalid value encountered in subtract
  sqr = _ensure_numeric((avg - values) ** 2)
```

In [ ]:

```
#Get Moody's BAA_Fed Fund Spread
file_path = 'BAAFFM.csv'
Moodys_df = pd.read_csv(file_path)


Moodys_df['BAAFFM'] = pd.to_numeric(Moodys_df['BAAFFM'], errors='coerce')
Moodys_df['DATE'] = pd.to_datetime(Moodys_df['DATE'], errors='coerce')

Moodys_df['YEAR'] = (Moodys_df['DATE']).dt.year
display(Moodys_df)
```

	DATE	BAAFFM	YEAR
0	1954-07-01	2.70	1954
1	1954-08-01	2.27	1954
2	1954-09-01	2.41	1954
3	1954-10-01	2.61	1954
4	1954-11-01	2.62	1954
...	...	...	...
827	2023-06-01	0.67	2023
828	2023-07-01	0.62	2023
829	2023-08-01	0.69	2023
830	2023-09-01	0.83	2023
831	2023-10-01	1.30	2023

832 rows × 3 columns

In [ ]:

```
# Convert 'YEAR' in subset_data from Period to integer
subset_data['YEAR'] = subset_data['YEAR'].dt.year

# Now perform the merge
MOOD_PD_df = pd.merge(Moodys_df, subset_data, on='YEAR', how='inner')
```

```
C:\Users\morganhales\AppData\Local\Temp\2\ipykernel_7436\1221831940.py:2: SettingWit
hCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  subset_data['YEAR'] = subset_data['YEAR'].dt.year
```

In [ ]:
```
display(MOOD_PD_df)
```

	DATE	BAAFFM	YEAR	Date_method1	DD_method1_n1	DD_method1_n2	DD_metho
0	1970-01-01	-0.12	1970	1970-04-15	-1.719682	-4.221202	1.0
1	1970-01-01	-0.12	1970	1970-12-24	2.997802	0.061921	9.3
2	1970-01-01	-0.12	1970	1970-10-06	4.745885	1.253440	16.3
3	1970-01-01	-0.12	1970	1970-03-17	2.482707	0.336141	4.3
4	1970-01-01	-0.12	1970	1970-10-20	-1.465141	-3.036053	-0.3
...	...	...	...	...	...	...	
10579	2019-12-01	2.33	2019	2019-04-08	1.151851	-0.383707	2.1
10580	2019-12-01	2.33	2019	2019-10-07	3.239176	-0.596924	11.6
10581	2019-12-01	2.33	2019	2019-01-22	2.163308	1.254036	2.7
10582	2019-12-01	2.33	2019	2019-03-25	3.188615	0.448632	8.0
10583	2019-12-01	2.33	2019	2019-05-22	4.855164	1.564169	14.4

10584 rows × 18 columns

In [ ]:
```
# Plotting
plt.figure(figsize=(12, 6))

# Plot each variable over time
plt.plot(MOOD_PD_df['YEAR'], MOOD_PD_df['BAAFFM'], label='BAAFFM', marker='o')
plt.plot(MOOD_PD_df['YEAR'], MOOD_PD_df['DD_method3'], label='DD_method3', marker='
plt.plot(MOOD_PD_df['YEAR'], MOOD_PD_df['PD_method3'], label='PD_method3', marker='

plt.title('Variables Over Time')
```

```
plt.xlabel('Year')
plt.ylabel('Values')
plt.legend()
plt.grid(True)
plt.show()
```