ANHELINA BABII

RBC Royal Bank

https://ca.finance.yahoo.com/
https://www.investing.com/
https://www.dbrsmorningstar.com/

MERTON MODEL

$$DD(t) = \frac{\log(\frac{V_A}{D}) + (r - \frac{1}{2}\sigma_A^2)(T-t)}{\sigma_A\sqrt{T-t}}$$

mathematical model

Needed DATA

Total assets: *1917.219B*
Outstanding debt: *1405.265B*
Divident yield: *0.037*
Risk-free rate: *0.03157*
std of daily returns: *0.025* *(code is shown below)*

```
# data definers
symbol = "RBC"
start_date = "2002-01-01"
end_date = "2022-12-31"
# historical price data from Yahoo Finance
df = yf.download(symbol, start=start_date, end=end_date)
# daily returns and standard deviation
returns = df["Adj Close"].pct_change()
sstd = returns.std()
print(f"Standard Deviation of Daily Returns: {sstd}")
```

```
[**********100%**********]  1 of 1 completed
Standard Deviation of Daily Returns: 0.0253071224012347
```

```
def equations(vars):
    V, sigma = vars
    d1 = (np.log(V / D) + (r + 0.5 * sigma**2) * T) / (sigma * np.sqrt(T))
    d2 = d1 - sigma * np.sqrt(T)
    eq1 = E - V * norm.cdf(d1) + D * np.exp(-r * T) * norm.cdf(d2)
    eq2 = sigma_E - sigma * norm.cdf(d1)
    return [eq1, eq2]
solution, info, _, _ = opt.fsolve(equations, [V0, sigma0], full_output=True)
V, sigma = solution
FVD = 1000
dividend_yield = 0.037
mu = r - dividend_yield
DD = (np.log(V / FVD) + (mu - 0.5 * sigma**2) * T) / (sigma * np.sqrt(T))
PD = 1 - norm.cdf(DD)
print(f"Probability of default for RBC: {format(PD, '.10f')}")
```

```
Probability of default for RBC: 0.0000000512
```

CREDITMETRICS MODEL

uses credit ratings (DBRS in this case)

	Date	Credit_Rating
0	2004-12-31	AA
1	2005-12-31	AA
2	2006-12-31	AA
3	2007-12-31	AA
4	2008-12-31	AA
5	2009-12-31	AA
6	2010-12-31	AA
7	2011-12-31	AA
8	2012-12-31	AA
9	2013-12-31	AA
10	2014-12-31	AA
11	2015-12-31	AA
12	2016-12-31	AA
13	2017-12-31	AA
14	2018-12-31	AA
15	2019-12-31	AA
16	2020-12-31	AA
17	2021-12-31	AA
18	2022-12-31	AA

imported data

	AAA	AA	A	BBB	BB	B	CCC	Default
AAA	0.90788	0.08291	0.00716	0.00102	0.00102	0.00000	0.00000	0.00000
AA	0.00103	0.91219	0.07851	0.00620	0.00103	0.00103	0.00000	0.00000
A	0.00924	0.02361	0.90041	0.05441	0.00719	0.00308	0.00103	0.00103
BBB	0.00000	0.00318	0.05938	0.86947	0.05302	0.01166	0.00117	0.00212
BB	0.00000	0.00110	0.00659	0.07692	0.80549	0.08791	0.00989	0.01209
B	0.00000	0.00114	0.00227	0.00454	0.06470	0.82747	0.04086	0.05902
CCC	0.00228	0.00000	0.00228	0.01251	0.02275	0.12856	0.60637	0.22526
Default	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	1.00000

transition matrix

```
transition_matrix = pd.DataFrame(trans_matrix,
                                 columns=ratings, index=ratings)
transition_matrix = transition_matrix / 100

current_credit_rating = credit_ratings['DBRS Rating'].iloc[-1]
one_year_default_pb = transition_matrix.loc[current_credit_rating, 'D']

print(f"Pobability of default for RBC: {one_year_default_pb}")
credit_ratings
```

```
Pobability of default for RBC: 0.0007000000000000001
```