Practical 2 - Feature Engineering / Distance to Default

Louis OLIVE

Table of contents

```r
library(tidyverse)
library(scales)
```

```r
# Helper function to list memory intensive R objects within environment

# showMemoryUse <- function(sort="size", decreasing=FALSE, limit) {
#   # usage showMemoryUse(decreasing=TRUE, limit = 15)
#   objectList <- ls(parent.frame())
#
#   oneKB <- 1024
#   oneMB <- 1048576
#   oneGB <- 1073741824
```

```
#
#   memoryUse <- sapply(objectList, function(x) as.numeric(object.size(eval(parse(text=x))
#
#   memListing <- sapply(memoryUse, function(size) {
#     if (size >= oneGB) return(paste(round(size/oneGB,2), "GB"))
#     else if (size >= oneMB) return(paste(round(size/oneMB,2), "MB"))
#     else if (size >= oneKB) return(paste(round(size/oneKB,2), "kB"))
#     else return(paste(size, "bytes"))
#   })
#
#   memListing <- data.frame(objectName=names(memListing),memorySize=memListing,row.names=
#
#   if (sort=="alphabetical") memListing <- memListing[order(memListing$objectName,decreas
#   else memListing <- memListing[order(memoryUse,decreasing=decreasing),] #will run if so
#
#   if(!missing(limit)) memListing <- memListing[1:limit,]
#
#   print(memListing, row.names=FALSE)
#   return(invisible(memListing))
# }
```

1 Loading WRDS / Compustat / CRSP data with `R`

We first load Accounting and Market data that we will use in this practical (see `1_wrds_data_wrangling.qmd` for more details on the data sources):

- Compustat Fundamental annual data:

```r
compustat_all <- readRDS("./wrds_data/compustat_all.rds")
```

- Compustat Company data:

```r
company_all <- readRDS("./wrds_data/company_all.rds")
```

- CRSP daily security file

```r
crsp_daily <- readRDS("./wrds_data/crsp_daily_full.rds")
```

Compustat and CRSP linking table:

```
ccmxpf_linktable <- readRDS("./wrds_data/ccmxpf_linktable.rds")
```

2 Practical 2 - Feature Engineering / Distance to Default

The aim of this practical is to enhance the data set from Practical 1.

We are taking advantage of its panel/time series format. Indeed each observation in the data set relates to a specific firm and a fiscal year (so called "firm year" observations).

The tasks ask you to implement and assess ideas from academic literature on corporate bankruptcy prediction.

2.1 Tasks to perform

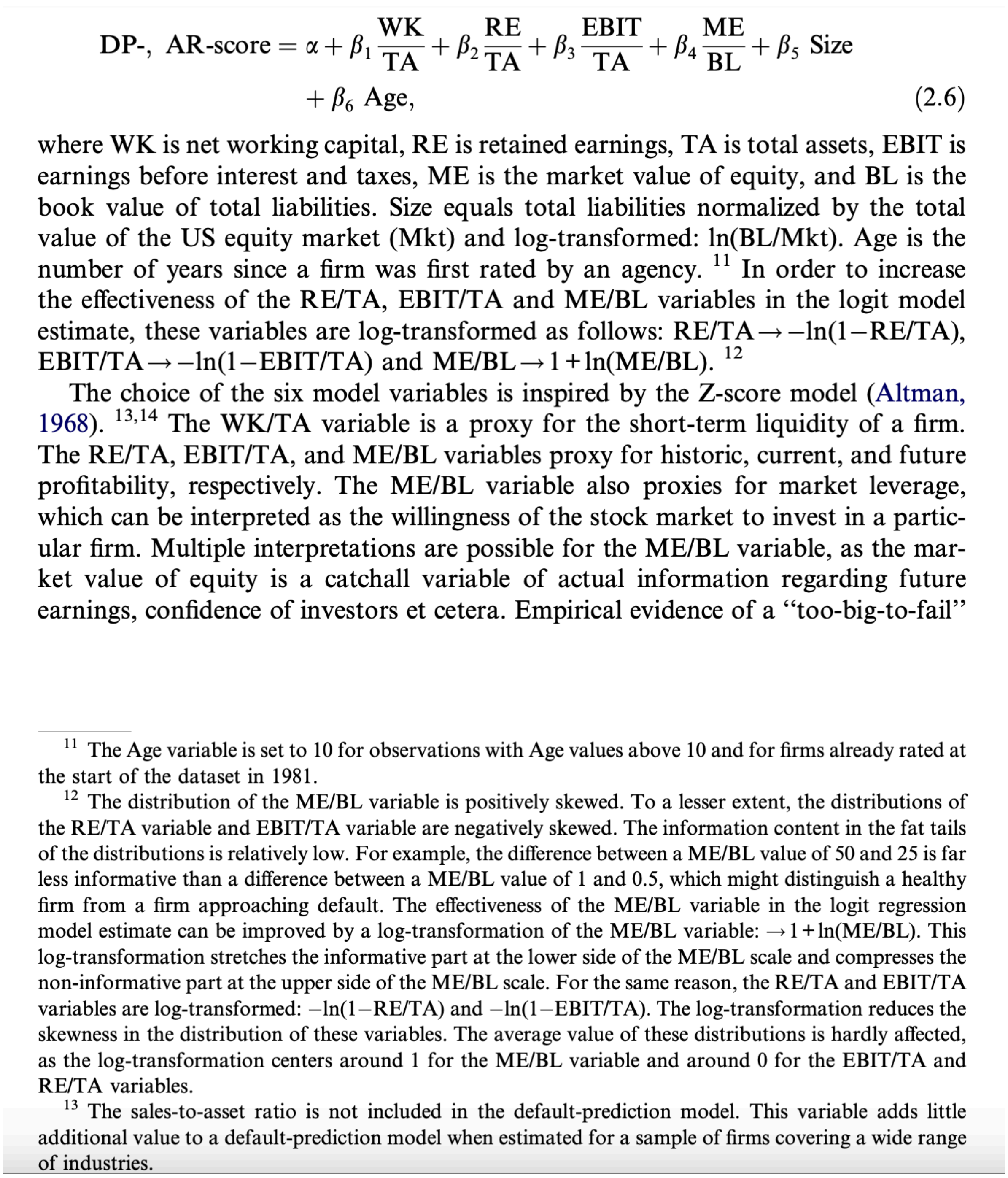
- **Task 1**: Altman model revisited

In Altman & Rijken (2004), authors (among which Altman) look back on Altman's seminal 1968 article and propose a modernized version of the Z-Score model.

The original 1968 article used Linear Discriminant Analysis on a small data set of 33/33 (non)defaulting firms.

In this article, authors re-use the Z-Score financial ratios (removing `SALE/TA`) as predictors in a discrete hazard / multi-period logit (a la Shumway, see Shumway (2001) or Practical 1). They also train their model on a larger data set (ca. 13k firm year observations).

They perform additional transformations of Z-scores financial ratios (mainly log transformation) as described page 2686 of the article:



$$\text{DP-, AR-score} = \alpha + \beta_1 \frac{\text{WK}}{\text{TA}} + \beta_2 \frac{\text{RE}}{\text{TA}} + \beta_3 \frac{\text{EBIT}}{\text{TA}} + \beta_4 \frac{\text{ME}}{\text{BL}} + \beta_5 \text{ Size} + \beta_6 \text{ Age}, \tag{2.6}$$

where WK is net working capital, RE is retained earnings, TA is total assets, EBIT is earnings before interest and taxes, ME is the market value of equity, and BL is the book value of total liabilities. Size equals total liabilities normalized by the total value of the US equity market (Mkt) and log-transformed: ln(BL/Mkt). Age is the number of years since a firm was first rated by an agency. [11] In order to increase the effectiveness of the RE/TA, EBIT/TA and ME/BL variables in the logit model estimate, these variables are log-transformed as follows: RE/TA $\rightarrow -\ln(1-\text{RE/TA})$, EBIT/TA $\rightarrow -\ln(1-\text{EBIT/TA})$ and ME/BL $\rightarrow 1+\ln(\text{ME/BL})$. [12]

The choice of the six model variables is inspired by the Z-score model (Altman, 1968). [13,14] The WK/TA variable is a proxy for the short-term liquidity of a firm. The RE/TA, EBIT/TA, and ME/BL variables proxy for historic, current, and future profitability, respectively. The ME/BL variable also proxies for market leverage, which can be interpreted as the willingness of the stock market to invest in a particular firm. Multiple interpretations are possible for the ME/BL variable, as the market value of equity is a catchall variable of actual information regarding future earnings, confidence of investors et cetera. Empirical evidence of a "too-big-to-fail"

[11] The Age variable is set to 10 for observations with Age values above 10 and for firms already rated at the start of the dataset in 1981.

[12] The distribution of the ME/BL variable is positively skewed. To a lesser extent, the distributions of the RE/TA variable and EBIT/TA variable are negatively skewed. The information content in the fat tails of the distributions is relatively low. For example, the difference between a ME/BL value of 50 and 25 is far less informative than a difference between a ME/BL value of 1 and 0.5, which might distinguish a healthy firm from a firm approaching default. The effectiveness of the ME/BL variable in the logit regression model estimate can be improved by a log-transformation of the ME/BL variable: $\rightarrow 1+\ln(\text{ME/BL})$. This log-transformation stretches the informative part at the lower side of the ME/BL scale and compresses the non-informative part at the upper side of the ME/BL scale. For the same reason, the RE/TA and EBIT/TA variables are log-transformed: $-\ln(1-\text{RE/TA})$ and $-\ln(1-\text{EBIT/TA})$. The log-transformation reduces the skewness in the distribution of these variables. The average value of these distributions is hardly affected, as the log-transformation centers around 1 for the ME/BL variable and around 0 for the EBIT/TA and RE/TA variables.

[13] The sales-to-asset ratio is not included in the default-prediction model. This variable adds little additional value to a default-prediction model when estimated for a sample of firms covering a wide range of industries.

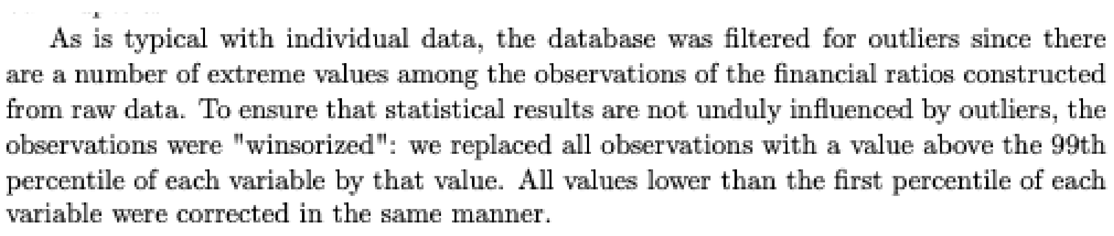
Authors justify taking the log of Altman's ratio by the skewness of some ratios distributions.

At this point you might want to explore more thoroughly some predictors (espicially those used by Altman) in your data set:

- providing descriptive statistics and features distributions,
- showing/plotting individual features "interaction" with the response variable
- identifying "outliers" if any.

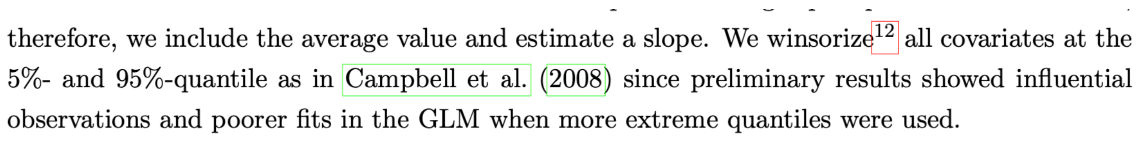
Visualizations help a lot.

Besides "log transformation", an alternative approach is usually performed by academics and practitioners in the literature: winsorizing/winsorization, although it is controversial. See below for examples in literature:



As is typical with individual data, the database was filtered for outliers since there are a number of extreme values among the observations of the financial ratios constructed from raw data. To ensure that statistical results are not unduly influenced by outliers, the observations were "winsorized": we replaced all observations with a value above the 99th percentile of each variable by that value. All values lower than the first percentile of each variable were corrected in the same manner.

or

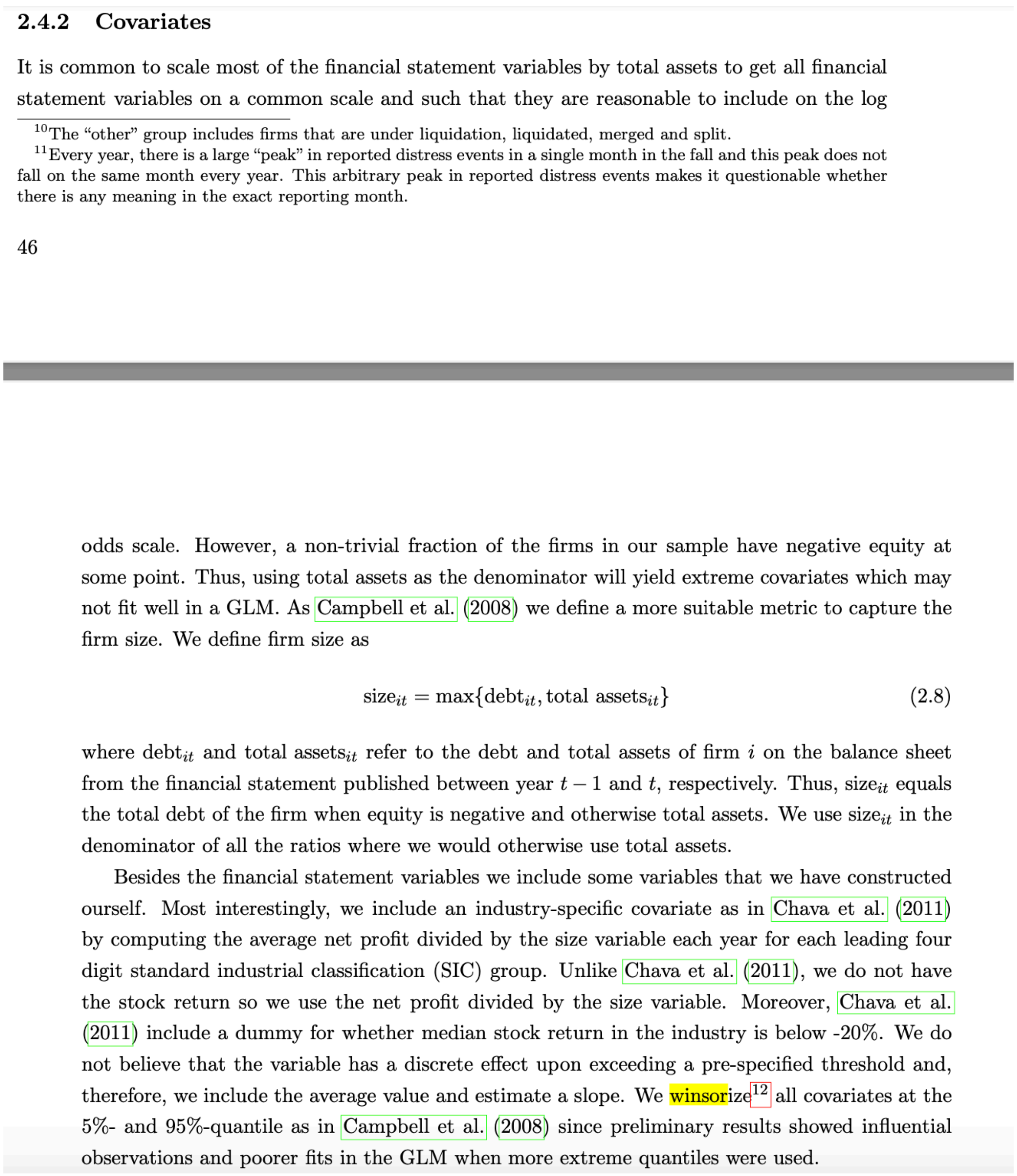


therefore, we include the average value and estimate a slope. We winsorize[12] all covariates at the 5%- and 95%-quantile as in Campbell et al. (2008) since preliminary results showed influential observations and poorer fits in the GLM when more extreme quantiles were used.

TO DO

- Explore the predictors as described before, in particular those needed to build Altman's ratios.
- Build transformed variables as described in Altman & Rijken (2004) article and assess the proposed model versus the previously fitted Altman model from Practical 1.
- Then perform "winsorizing" on Altman variables and assess the model. You can test different levels of "winsorizing".

Later in the project, you can use additional ideas from provided literature, for example in Christoffersen (2019), pages 46-47, the author scale its predictors with an alternative firm size measure, and furthermore uses "winsorizing":



2.4.2 Covariates

It is common to scale most of the financial statement variables by total assets to get all financial statement variables on a common scale and such that they are reasonable to include on the log

[10] The "other" group includes firms that are under liquidation, liquidated, merged and split.

[11] Every year, there is a large "peak" in reported distress events in a single month in the fall and this peak does not fall on the same month every year. This arbitrary peak in reported distress events makes it questionable whether there is any meaning in the exact reporting month.

46

odds scale. However, a non-trivial fraction of the firms in our sample have negative equity at some point. Thus, using total assets as the denominator will yield extreme covariates which may not fit well in a GLM. As Campbell et al. (2008) we define a more suitable metric to capture the firm size. We define firm size as

$$\text{size}_{it} = \max\{\text{debt}_{it}, \text{total assets}_{it}\} \tag{2.8}$$

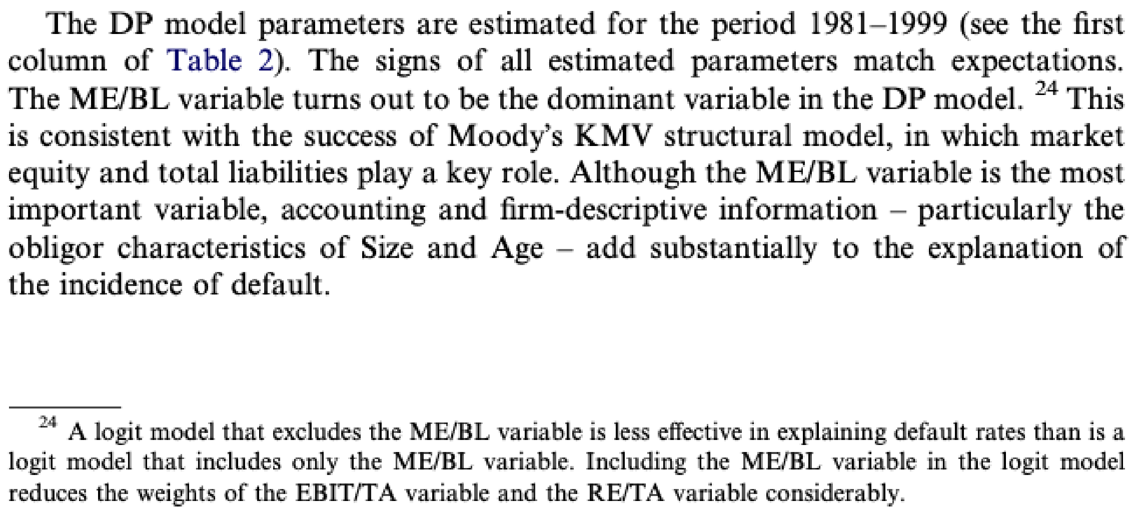
where debt_{it} and total assets_{it} refer to the debt and total assets of firm i on the balance sheet from the financial statement published between year $t-1$ and t, respectively. Thus, size_{it} equals the total debt of the firm when equity is negative and otherwise total assets. We use size_{it} in the denominator of all the ratios where we would otherwise use total assets.

Besides the financial statement variables we include some variables that we have constructed ourself. Most interestingly, we include an industry-specific covariate as in Chava et al. (2011) by computing the average net profit divided by the size variable each year for each leading four digit standard industrial classification (SIC) group. Unlike Chava et al. (2011), we do not have the stock return so we use the net profit divided by the size variable. Moreover, Chava et al. (2011) include a dummy for whether median stock return in the industry is below -20%. We do not believe that the variable has a discrete effect upon exceeding a pre-specified threshold and, therefore, we include the average value and estimate a slope. We winsorize[12] all covariates at the 5%- and 95%-quantile as in Campbell et al. (2008) since preliminary results showed influential observations and poorer fits in the GLM when more extreme quantiles were used.

When using scaling transformations using the data set distribution (for example "winsorizing"), be careful not to learn anything from your testing set when assessing your models. In particular when "winsorizing", take care to avoid using testing set to compute the quantiles used to clip/limit your data.

BONUS Note that similarly to Shumway (2001), authors add firm `Size` and `Age` predictors. It is up to you to add such indicators, using your own assumptions.

Finally, authors remark that the variable `ME/BL` is "dominant" in one of their model:


The DP model parameters are estimated for the period 1981–1999 (see the first column of Table 2). The signs of all estimated parameters match expectations. The ME/BL variable turns out to be the dominant variable in the DP model. [24] This is consistent with the success of Moody's KMV structural model, in which market equity and total liabilities play a key role. Although the ME/BL variable is the most important variable, accounting and firm-descriptive information – particularly the obligor characteristics of Size and Age – add substantially to the explanation of the incidence of default.

[24] A logit model that excludes the ME/BL variable is less effective in explaining default rates than is a logit model that includes only the ME/BL variable. Including the ME/BL variable in the logit model reduces the weights of the EBIT/TA variable and the RE/TA variable considerably.

This finding suggests going further and looking at more complex or less linear predictors based on both "Market Value" and "Book Value" of firms.

The Distance to Default that we describe below is such a predictor.

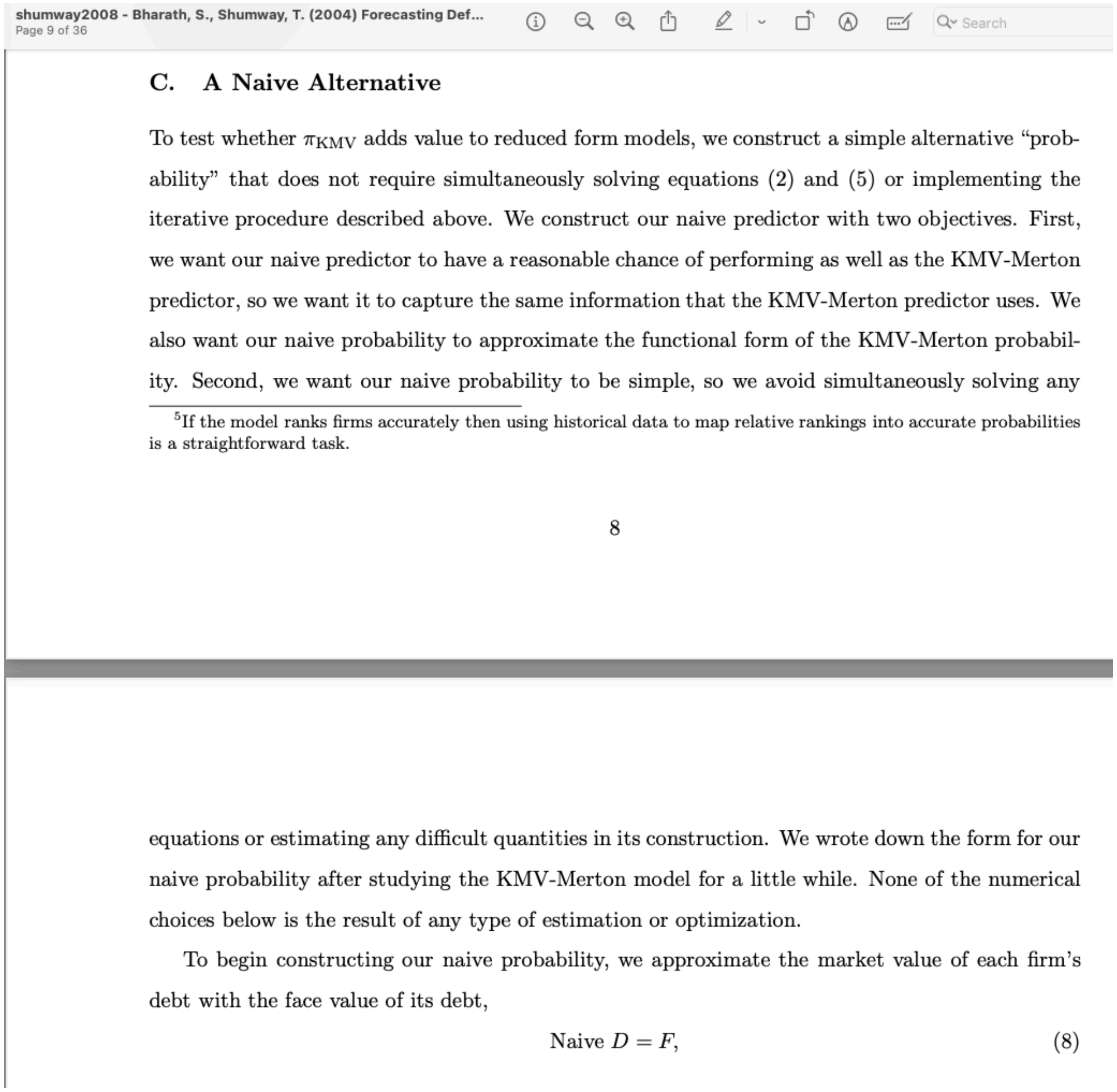
- **Task 2**: Merton model / Distance to Default (Bharath & Shumway (2008), Vassalou & Xing (2004))

In Bharath & Shumway (2008), authors assess predictors based on Merton model / Distance to Default (DtD). DTD is a model based measure for gauging how far a firm is away from default/bankruptcy/failure. We give more theoretical and practical background about the Merton model and DtD in the next section.

More specifically authors assess as predictors two methods of estimating DtD. First a "Naïve" method. Then a more complex "Modeled/Iterative" method.

Stating Bharath & Shumway (2008) abstract: "We compare the model to a"naïve" alternative, which uses the functional form suggested by the Merton model but does not solve the model for an implied probability of default. We find that the "naïve" predictor performs slightly better in hazard models and in out-of-sample forecasts than both the Merton DD model and a reduced-form model that uses the same inputs. […] We conclude that while the Merton DD model does not produce a sufficient statistic for the probability of default, its functional form is useful for forecasting defaults."

The "Naïve" DtD predictor is built the following way (see Section 3.2.2 and following paragraphs for further details):



shumway2008 - Bharath, S., Shumway, T. (2004) Forecasting Def...
Page 9 of 36

C. A Naive Alternative

To test whether π_{KMV} adds value to reduced form models, we construct a simple alternative "probability" that does not require simultaneously solving equations (2) and (5) or implementing the iterative procedure described above. We construct our naive predictor with two objectives. First, we want our naive predictor to have a reasonable chance of performing as well as the KMV-Merton predictor, so we want it to capture the same information that the KMV-Merton predictor uses. We also want our naive probability to approximate the functional form of the KMV-Merton probability. Second, we want our naive probability to be simple, so we avoid simultaneously solving any

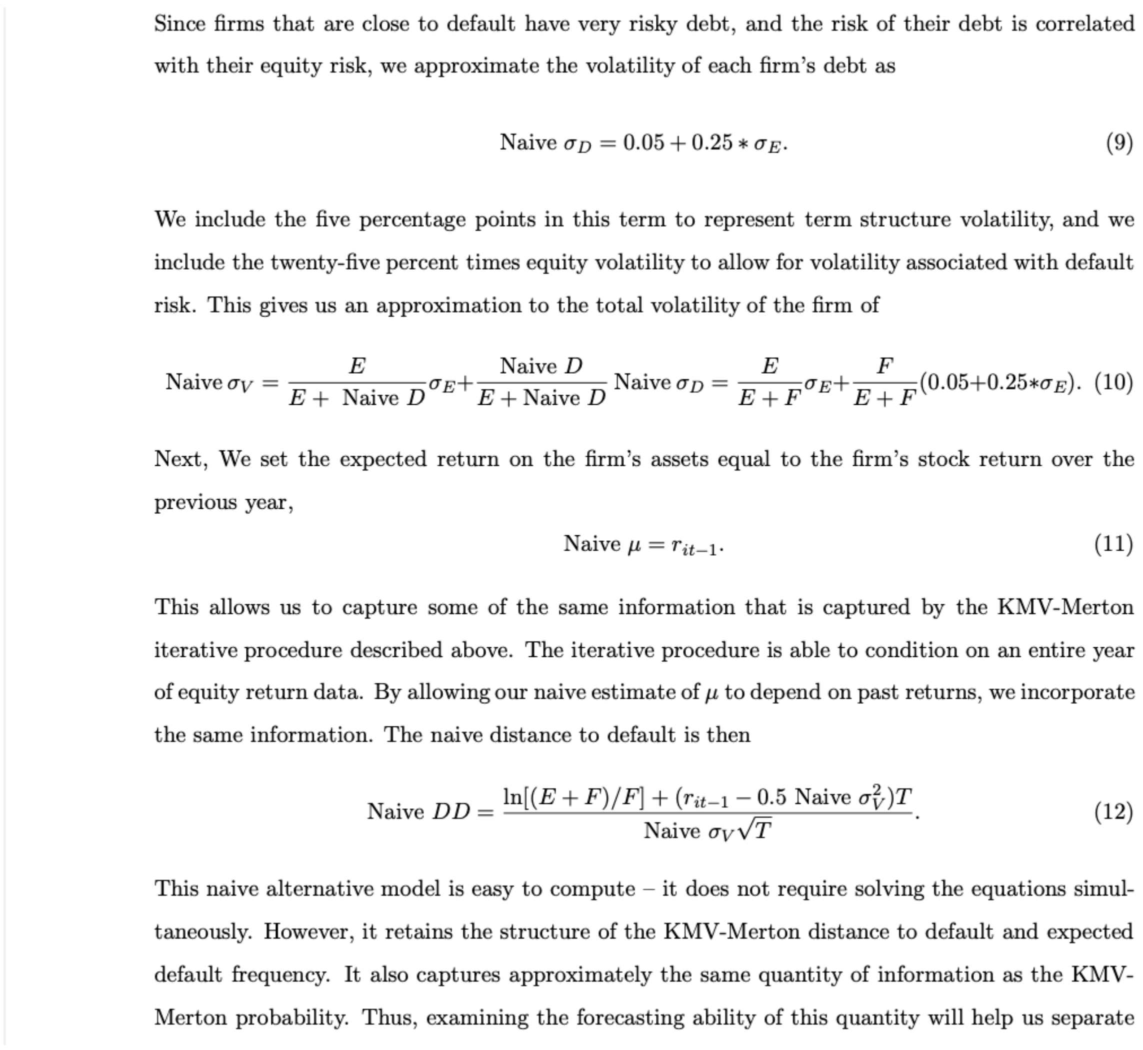
[5]If the model ranks firms accurately then using historical data to map relative rankings into accurate probabilities is a straightforward task.

8

equations or estimating any difficult quantities in its construction. We wrote down the form for our naive probability after studying the KMV-Merton model for a little while. None of the numerical choices below is the result of any type of estimation or optimization.

To begin constructing our naive probability, we approximate the market value of each firm's debt with the face value of its debt,

$$\text{Naive } D = F, \tag{8}$$



Since firms that are close to default have very risky debt, and the risk of their debt is correlated with their equity risk, we approximate the volatility of each firm's debt as

$$\text{Naive } \sigma_D = 0.05 + 0.25 * \sigma_E. \tag{9}$$

We include the five percentage points in this term to represent term structure volatility, and we include the twenty-five percent times equity volatility to allow for volatility associated with default risk. This gives us an approximation to the total volatility of the firm of

$$\text{Naive } \sigma_V = \frac{E}{E + \text{ Naive } D}\sigma_E + \frac{\text{Naive } D}{E + \text{Naive } D}\text{ Naive } \sigma_D = \frac{E}{E+F}\sigma_E + \frac{F}{E+F}(0.05+0.25*\sigma_E). \tag{10}$$

Next, We set the expected return on the firm's assets equal to the firm's stock return over the previous year,

$$\text{Naive } \mu = r_{it-1}. \tag{11}$$

This allows us to capture some of the same information that is captured by the KMV-Merton iterative procedure described above. The iterative procedure is able to condition on an entire year of equity return data. By allowing our naive estimate of μ to depend on past returns, we incorporate the same information. The naive distance to default is then

$$\text{Naive } DD = \frac{\ln[(E+F)/F] + (r_{it-1} - 0.5 \text{ Naive } \sigma_V^2)T}{\text{Naive } \sigma_V \sqrt{T}}. \tag{12}$$

This naive alternative model is easy to compute – it does not require solving the equations simultaneously. However, it retains the structure of the KMV-Merton distance to default and expected default frequency. It also captures approximately the same quantity of information as the KMV-Merton probability. Thus, examining the forecasting ability of this quantity will help us separate

TO DO

- Following the description in the article Bharath & Shumway (2008), first build a "Naïve" DtD predictor. Assess this additional predictor.
- Compare with a "Modeled/Iterative" DtD predictor (data provided).

The idea is to test the article main hypothesis: "Naïve" DtD is better than / as good as a "Modeled" DtD, on your data set.

For the second sub-task, we provide "Modeled/Iterative" DtD values fitted using the so-called KMV method in the literature (data available here `wrds_data/dtd_cleansed.rds`). It has been done on a best effort basis, meaning that it can be improved in many ways.

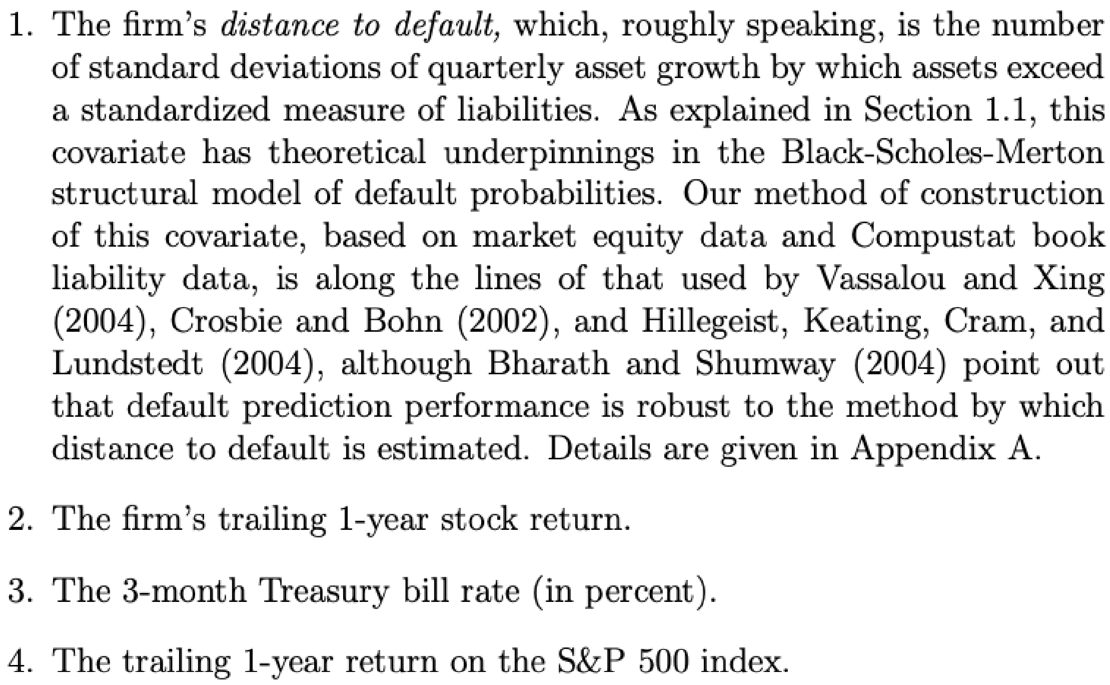
We also provide corresponding `R` code that was used to estimate DtD, to allow reproducibility and for information purposes. We try to document the hypothesis we make at each step and seek to reproduce DtD figures estimated in seminal articles by academics/practitioners.

BONUS The iterative process is rather memory intensive. If really interested, you might want to use the code and adapt it to your own hypothesis. In particular we use Annual Fundamentals data, some academics/practitioners use Quarterly Fundamentals giving a better granularity to the DtD estimates.

- **Task 3**: Complete the data set with additional market data or macroeconomic time series (as in Shumway (2001) / Chava & Jarrow (2004) / Duffie et al. (2007))

TO DO Complete your data set with at least two additional Macro/Market predictors (one for each category Macro/Market). Assess the models.

Some ideas are presented for example in Duffie et al. (2007), the Market indicators are usually built using CRSP, the Macro indicators are usually available on FRED website:



1. The firm's *distance to default,* which, roughly speaking, is the number of standard deviations of quarterly asset growth by which assets exceed a standardized measure of liabilities. As explained in Section 1.1, this covariate has theoretical underpinnings in the Black-Scholes-Merton structural model of default probabilities. Our method of construction of this covariate, based on market equity data and Compustat book liability data, is along the lines of that used by Vassalou and Xing (2004), Crosbie and Bohn (2002), and Hillegeist, Keating, Cram, and Lundstedt (2004), although Bharath and Shumway (2004) point out that default prediction performance is robust to the method by which distance to default is estimated. Details are given in Appendix A.

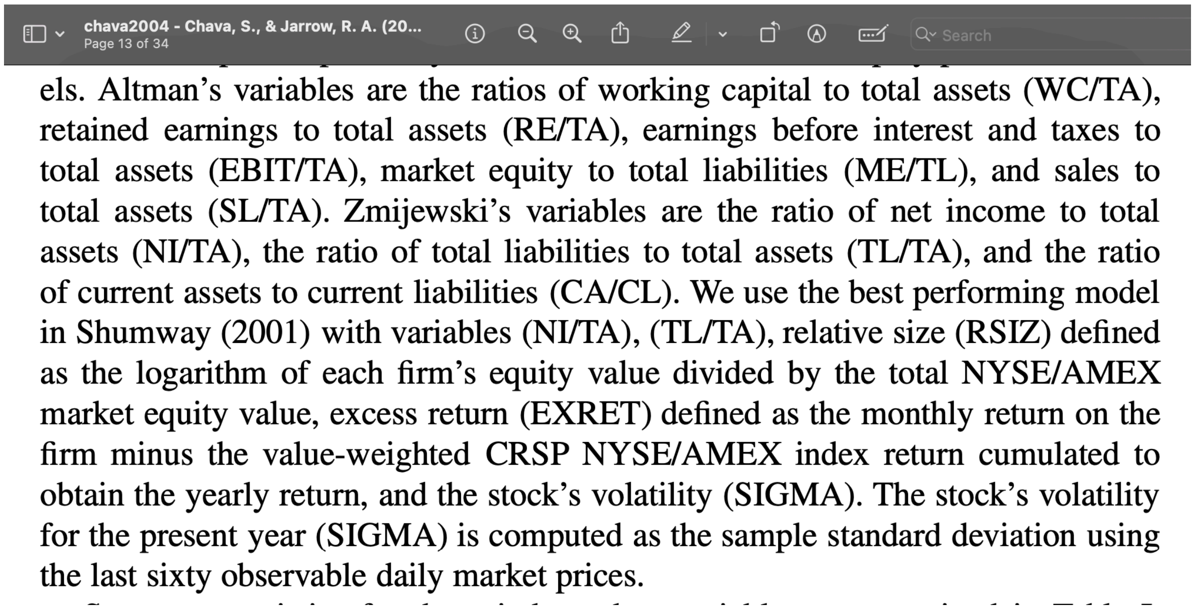
2. The firm's trailing 1-year stock return.

3. The 3-month Treasury bill rate (in percent).

4. The trailing 1-year return on the S&P 500 index.

There is an example usage of `FRED` in the sections below, we use U.S. Treasuries 1Y times series found here, as an input to fit DtD.

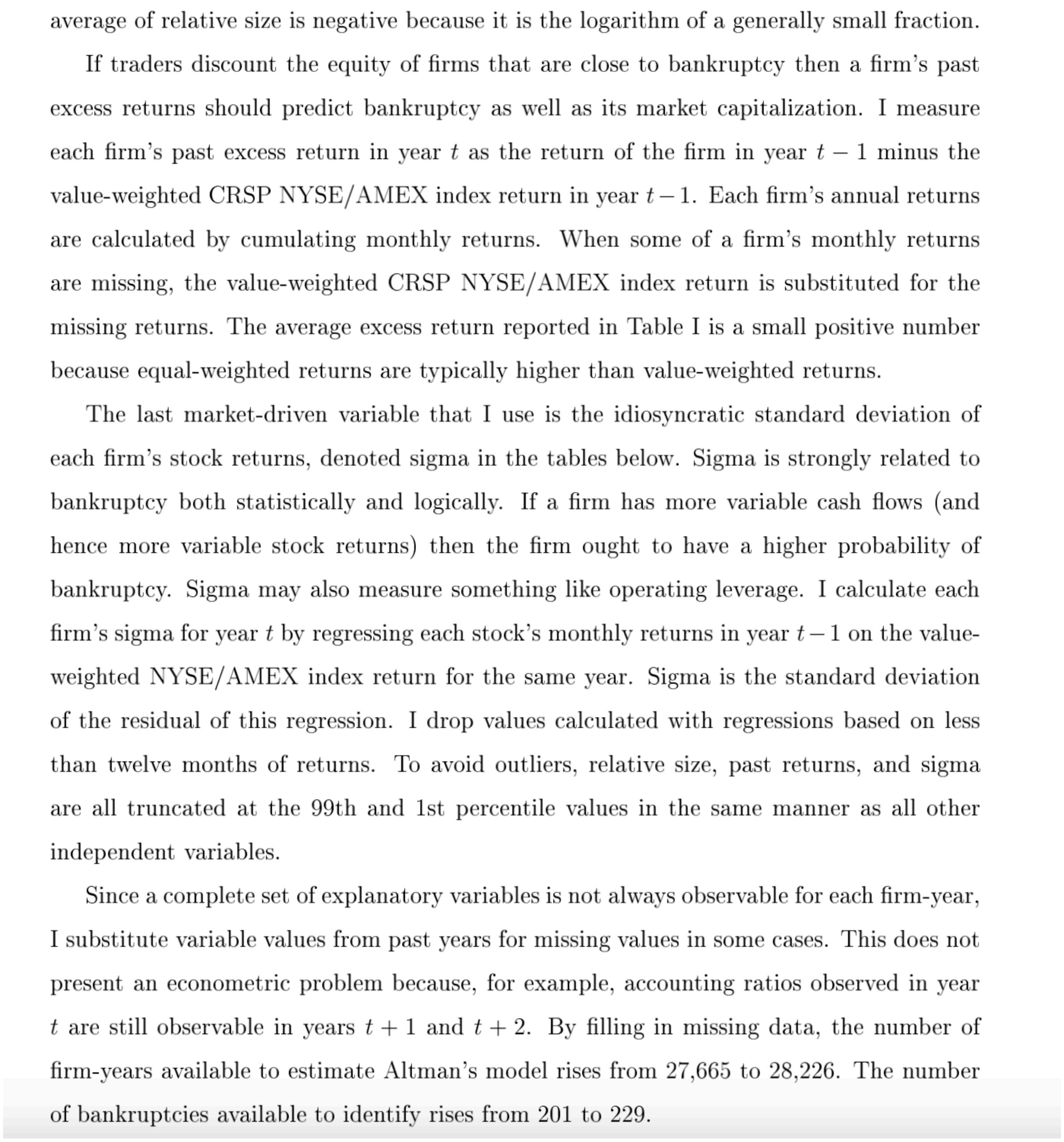
In Chava & Jarrow (2004) authors use firm excess returns against a reference equity index (`EXRET`) and standard deviation of past equity returns (`SIGMA`):



els. Altman's variables are the ratios of working capital to total assets (WC/TA), retained earnings to total assets (RE/TA), earnings before interest and taxes to total assets (EBIT/TA), market equity to total liabilities (ME/TL), and sales to total assets (SL/TA). Zmijewski's variables are the ratio of net income to total assets (NI/TA), the ratio of total liabilities to total assets (TL/TA), and the ratio of current assets to current liabilities (CA/CL). We use the best performing model in Shumway (2001) with variables (NI/TA), (TL/TA), relative size (RSIZ) defined as the logarithm of each firm's equity value divided by the total NYSE/AMEX market equity value, excess return (EXRET) defined as the monthly return on the firm minus the value-weighted CRSP NYSE/AMEX index return cumulated to obtain the yearly return, and the stock's volatility (SIGMA). The stock's volatility for the present year (SIGMA) is computed as the sample standard deviation using the last sixty observable daily market prices.

Note that `SIGMA` (using a 1Y window instead of 60 days in the article) is one of the input of the Distance to Default model estimated in the next sections, so that you can re-use parts of the code below to build the predictor.

BONUS Shumway (2001) also uses excess returns and goes further estimating an idiosyncratic volatility for each firm. He regresses firm returns against a reference equity index returns, then takes the standard deviation of the residuals. You can estimate such idiosyncratic volatility using rolling windows linear regressions.


average of relative size is negative because it is the logarithm of a generally small fraction.

If traders discount the equity of firms that are close to bankruptcy then a firm's past excess returns should predict bankruptcy as well as its market capitalization. I measure each firm's past excess return in year t as the return of the firm in year $t-1$ minus the value-weighted CRSP NYSE/AMEX index return in year $t-1$. Each firm's annual returns are calculated by cumulating monthly returns. When some of a firm's monthly returns are missing, the value-weighted CRSP NYSE/AMEX index return is substituted for the missing returns. The average excess return reported in Table I is a small positive number because equal-weighted returns are typically higher than value-weighted returns.

The last market-driven variable that I use is the idiosyncratic standard deviation of each firm's stock returns, denoted sigma in the tables below. Sigma is strongly related to bankruptcy both statistically and logically. If a firm has more variable cash flows (and hence more variable stock returns) then the firm ought to have a higher probability of bankruptcy. Sigma may also measure something like operating leverage. I calculate each firm's sigma for year t by regressing each stock's monthly returns in year $t-1$ on the value-weighted NYSE/AMEX index return for the same year. Sigma is the standard deviation of the residual of this regression. I drop values calculated with regressions based on less than twelve months of returns. To avoid outliers, relative size, past returns, and sigma are all truncated at the 99th and 1st percentile values in the same manner as all other independent variables.

Since a complete set of explanatory variables is not always observable for each firm-year, I substitute variable values from past years for missing values in some cases. This does not present an econometric problem because, for example, accounting ratios observed in year t are still observable in years $t+1$ and $t+2$. By filling in missing data, the number of firm-years available to estimate Altman's model rises from 27,665 to 28,226. The number of bankruptcies available to identify rises from 201 to 229.

We give below for information purposes some material on Distance to Default, introducing theoretical background and practical implementation aspects.

3 Some theory and practice - From Merton to KMV model

The informal presentation of the Merton model we give below is rather standard (see Duan & Wang (2012) for an extended presentation). For a more formal introduction see also Moreno-Bromberg & Rochet (2018).

3.1 The Merton Model (1974)

The Merton (1974) model assumes that firm's capital structure consists of debt and equity, with equity value at time t denoted by E_t (observable on the market), and debt represented by a single zero coupon bond (denoted by D_t) with maturity date T and principal/face value F. This means that the firm can only default at the maturity of debt T.

The Asset value of the firm V_t (unobservable) follows a geometric Brownian motion:

$$dV_t = \mu V_t dt + \sigma_V V_t dW_t$$

where,

- V_t is the total market value of the firm
- μ is the expected return on V (continuously compounded)
- σ_V is the volatility of firm value
- W_t is a standard Wiener process or Brownian motion

Using (i) Merton's assumptions that the firm has one zero coupon bond maturing in T periods and (ii) the principle of limited liability that protects equity investors, the equity of the firm can be viewed as a call option on the underlying value of the firm with a strike price equal to the face value of the firm's debt and a time-to-maturity of T (payoff $E_T = max(0, V_T - F)$).

Indeed, shareholders would choose not to repay the firm's debt when the value of the firm at maturity T is less than the value of the outstanding debt.

The equity value at time t of the firm is hence a function of the firm's value similar to Black-Scholes-Merton model:

$$E(V_t, \sigma_V) = V_t \mathcal{N}(d_1) - e^{-r(T-t)} F \mathcal{N}(d_2) \tag{1}$$

where,

- E is the market value of the firm's equity
- F is the face value of the firm's debt
- r is the risk-free rate
- $\mathcal{N}(\cdot)$ is the cumulative standard normal distribution function

and,

$$d_1 = \frac{\ln(V_t/F) + (r + \frac{1}{2}\sigma_V^2)(T-t)}{\sigma_V\sqrt{T-t}}$$

with $d_2 = d_1 - \sigma_V\sqrt{T-t}$.

Building on Merton's approach, the KMV Corporation (for its creators Kealhofer, McQuown, Vasicek, now part of Moody's) introduced the Distance to Default (DtD) as:

$$DtD = \frac{\ln(V/DP) + (\mu_V - \frac{1}{2}\sigma_V^2)T}{\sigma_V\sqrt{T}} \tag{2}$$

where,

- μ_V, is an estimate of the expected market annual return of the firm's assets (also called drift);
- DP, is the firm's default point defined as the face value of the firm's short-term debt plus one half of the firm's long-term debt.

The probability of default is then:

$$PD_{Merton} = \mathcal{N}\left(-DtD\right)$$

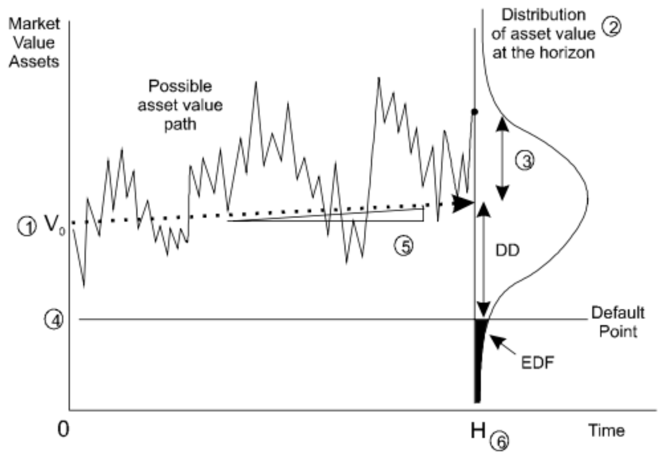
Not that instead of using the normal distribution, KMV uses an empirical distribution fitted on a proprietary database to convert DtD to a probability estimate.

A classical graph gives some intuition on the model, here taken from a white paper from KMV (see Crosbie & Bohn (2003)):

Calculate the Distance-to-default

There are six variables that determine the default probability of a firm over some horizon, from now until time H (see Figure 8):

1. The current asset value.
2. The distribution of the asset value at time H.
3. The volatility of the future assets value at time H.
4. The level of the default point, the book value of the liabilities.
5. The expected rate of growth in the asset value over the horizon.
6. The length of the horizon, H.

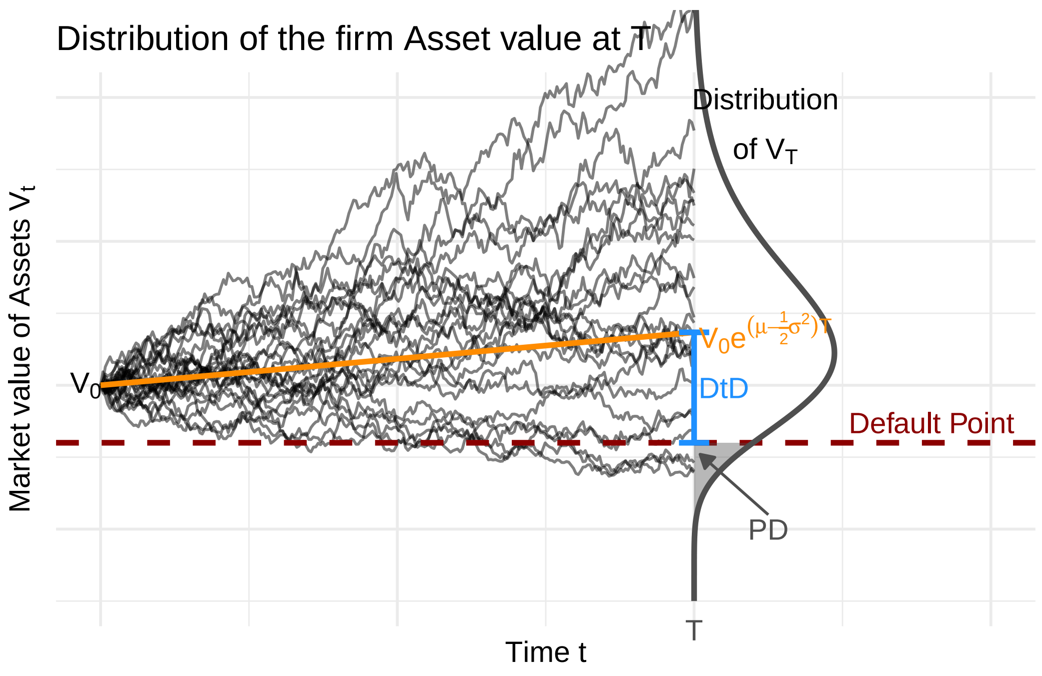

Market Value Assets
Possible asset value path
Distribution of asset value at the horizon ②
③
① V_0
⑤
DD
④
Default Point
EDF
0
H ⑥
Time

KMV LLC 11 Release Date: 15-November-1993
Revised: 14-January-2002

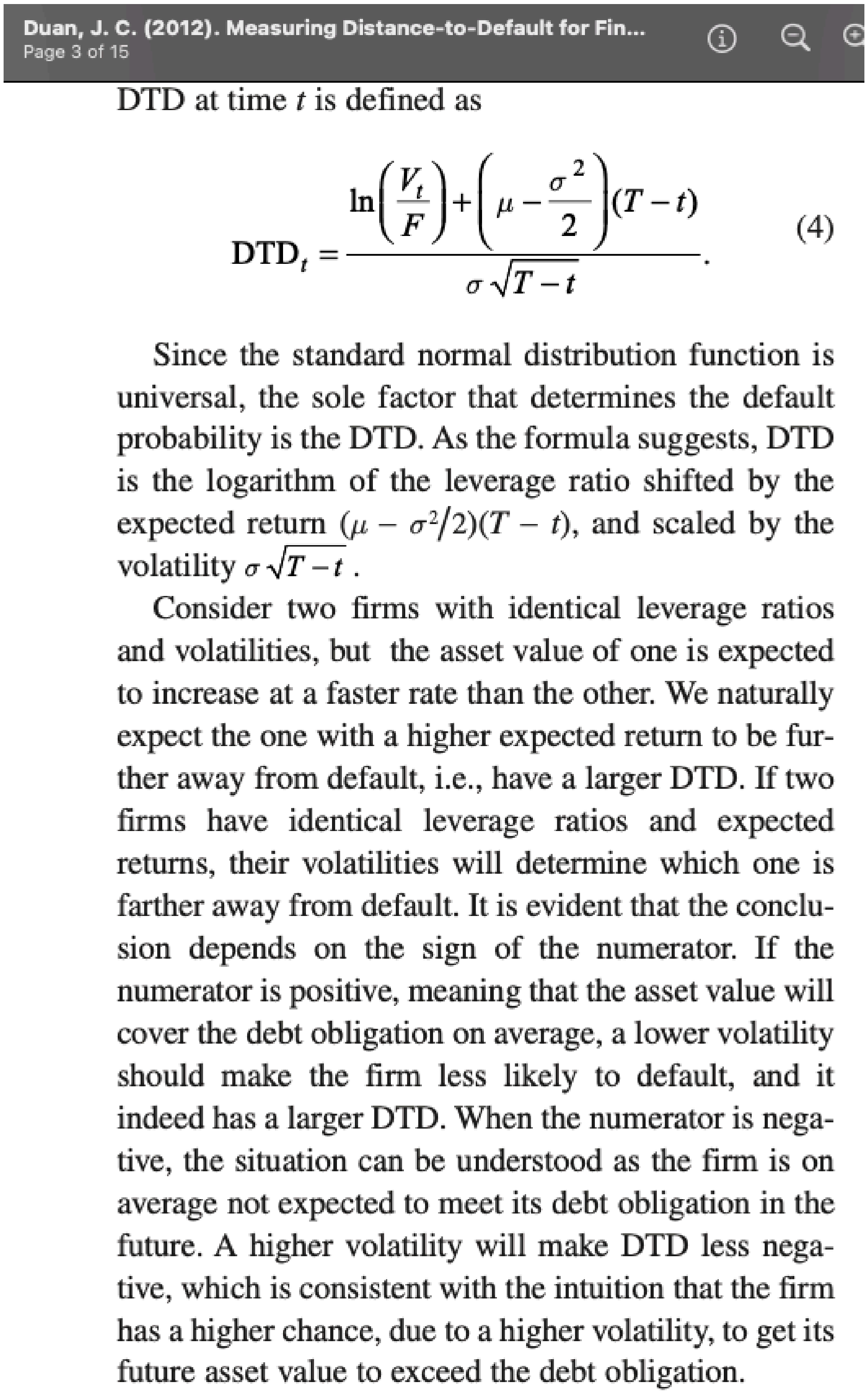
We reproduce below the graph using `R`, allowing you to play with main parameters / drivers of the model; as usually done in the literature, we voluntarily exaggerate some parameters to improve the graph readability:

```r
# We reproduce here the classical Distance to Default plot
# showing multiple paths of Asset value GBM process

# Parameters for the GBM and plot
T. <- 1         # Time to maturity
N <- 252      # # of time steps
```

```r
mu <- 0.2    # Drift (expected return) (excessive for better readability)
sigma <- 0.25  # Volatility
V0 <- 100     # Initial asset value
default_point <- 80  # Default point
dt <- T. / N
time <- seq(0, T., length.out = N)

# Simulate m Geometric Brownian Motion (GBM) paths for asset value V
# better explanations / code here:
# https://gregorygundersen.com/blog/2024/09/28/black-scholes/
# https://gregorygundersen.com/blog/2024/04/13/simulating-gbm/

set.seed(1987)
m <- 25
# one draw for each time step / for each simulation m
w <- matrix(rnorm((N-1)*m), ncol = N-1, nrow = m)
gbm <- exp((mu - sigma * sigma / 2) * dt + sigma * w * sqrt(dt))
gbm <- as_tibble(t(apply(cbind(rep(V0, m), gbm), 1, cumprod))) %>%
         purrr::set_names(time) %>%
         mutate(idx = 1:nrow(gbm)) %>%
         pivot_longer(-idx, names_to = 'time', values_to = 'V') %>%
         mutate(time = as.numeric(time))

# Create a Lognormal distribution function for the returns at time T
# https://gregorygundersen.com/blog/2023/12/17/lognormal/
k <- 4
ymin <- 25
v_dens <- seq(ymin, V0 * exp(k * sigma), length.out = 1000)
lognorm_density <- dlnorm(v_dens, meanlog = log(V0) + mu - sigma ^ 2 / 2, sdlog = sigma)

lognorm_density <- tibble(x = T. + 17 * lognorm_density, # upscale(x17) for plot purposes
                          y = v_dens)



# Distance to Default plot
ggplot(gbm, aes(x = time, y = V)) +
  # GBM paths
  geom_line(aes(group=idx), linewidth = 0.5, alpha = 0.5) +
  # Default point
  geom_hline(yintercept = default_point, linetype = "dashed", color = "darkred", linewidth
  # Rotated Lognormal distribution in T
  geom_path(data = lognorm_density, aes(x = x, y = y), color = "grey31", linewidth = 1) +
  # PD shaded area under curve for the "quantile" Default point
  geom_ribbon(data = lognorm_density %>% filter(y<=default_point),
            aes(x = x, y = y, ymin = y, ymax = default_point), fill = "grey31", alpha = 0.
  # Various annotations
  annotate("text", x = T., y = ymin-10, label = "T", colour = "grey31") +
  annotate("segment", x = T.+0.125,
                      y = default_point-25,
                      xend = T.+0.01,
                      yend = default_point-4,
                      color = "grey31",
                      arrow=arrow(length = unit(.2,"cm"), type="closed")) +
  annotate("text", x = T.+0.125, y = default_point-30, label = "PD", colour = "grey31") +
  annotate("segment", x=0, y=V0, xend=T., yend=V0*exp((mu-1/2*sigma^2)*T.),
           color="darkorange", linewidth = 1) +
  annotate("text", x = -0.025, y = V0, label = "V[0]", parse=TRUE) +
  annotate("segment", x = T., xend = T., y = default_point, yend = V0*exp((mu-1/2*sigma^2)
           arrow = arrow(ends = "both", angle = 90, length = unit(.2,"cm")),
           color="dodgerblue", linewidth = 1) +
  annotate("text", x = T.+0.05,
                   y = 0.5*(default_point+V0*exp((mu-1/2*sigma^2)*T.)),
                   label = "DtD",
                   color="dodgerblue") +
  annotate("text", x = T.+0.12,
                   y = V0*exp((mu-1/2*sigma^2)*T.),
                   color="darkorange",
                   label = "V[0]*e^{(mu-frac(1,2)*sigma^2)*T}",
                   parse=TRUE) +
  annotate("text", x = T. + 0.12, y = 190,
                   label = "atop(Distribution, 'of '*V[T])",
                   parse=TRUE) +
  annotate("text", x = 1.4, y = default_point + 7, label = "Default Point", color = "darkr
  labs(title = "Distribution of the firm Asset value at T",
        x = "Time t", y = expression(Market~value~of~Assets~V[t])) +
  theme_minimal() +
  theme(axis.text.x=element_blank(),
        axis.text.y=element_blank()) +
  coord_cartesian(xlim=c(0, 1.5), ylim=c(ymin, 200), clip='off')
```



In addition, in Duan & Wang (2012), authors give further intuition on the model dynamic:



DTD at time t is defined as

$$\mathrm{DTD}_t = \frac{\ln\left(\frac{V_t}{F}\right) + \left(\mu - \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}}. \qquad (4)$$

Since the standard normal distribution function is universal, the sole factor that determines the default probability is the DTD. As the formula suggests, DTD is the logarithm of the leverage ratio shifted by the expected return $(\mu - \sigma^2/2)(T - t)$, and scaled by the volatility $\sigma\sqrt{T-t}$.

Consider two firms with identical leverage ratios and volatilities, but the asset value of one is expected to increase at a faster rate than the other. We naturally expect the one with a higher expected return to be further away from default, i.e., have a larger DTD. If two firms have identical leverage ratios and expected returns, their volatilities will determine which one is farther away from default. It is evident that the conclusion depends on the sign of the numerator. If the numerator is positive, meaning that the asset value will cover the debt obligation on average, a lower volatility should make the firm less likely to default, and it indeed has a larger DTD. When the numerator is negative, the situation can be understood as the firm is on average not expected to meet its debt obligation in the future. A higher volatility will make DTD less negative, which is consistent with the intuition that the firm has a higher chance, due to a higher volatility, to get its future asset value to exceed the debt obligation.

Implementing the Merton or KMV models presents several challenges.

Notably asset values V are not directly observable or tradable, making them unavailable for direct use in the Distance to Default (DtD) calculation. Parameters μ_V and σ_V which govern the unobserved asset value process V must be estimated.

Various estimation methods have been proposed by both practitioners and academics, Duan & Wang (2012) covers some of these methods.

3.2 Estimation

Usually we observe from the market:

- E, the equity value of the firm; linked to V in Merton model by the call option formula
- σ_E, the volatility of equity; which can be estimated using the standard deviation of log returns

It can be shown that the volatility of the firm's equity σ_E is related to the volatility of the firm's value σ_V the following way:

$$\sigma_E = \left(\frac{V}{E}\right)\frac{\partial E}{\partial V}\sigma_V$$

In the Black-Scholes-Merton model, $\frac{\partial E}{\partial V} = \mathcal{N}(d_1)$, so that

$$\sigma_E = \left(\frac{V}{E}\right)\mathcal{N}(d_1)\sigma_V \tag{3}$$

Using equations (Equation 1) and (Equation 3) we are able to infer / solve for:

- V, the total value of the firm
- σ_V, the volatility of firm value

To estimate DtD, an iterative procedure can be applied instead of solving (Equation 1) and (Equation 3) simultaneously (see for example Crosbie & Bohn (2003), Vassalou & Xing (2004), Bharath & Shumway (2008), Duan & Wang (2012)).

3.2.1 An iterative approach

We mostly repeat below the explanation from an excellent blog post documenting how to practically estimate DtD using Compustat/CRSP data (see here), which mostly repeats the exposition from Bharath & Shumway (2008) article:

Method: Iterative estimation of Merton model (a la KMV)

---

1. Set the initial value of $\sigma_V = \sigma_E[E/(E + DP)]$.
2. Use this value of σ_V and equation (Equation 1) to infer the market value of firm's assets every day for the previous year.
3. Calculate the implied log return on assets each day, based on which generate new estimates of σ_V and μ_V.
4. Repeat steps 2 to 3 until σ_V converges, i.e., the absolute difference in adjacent estimates is less than a tolerance value.
5. Use σ_V and μ_V in equation (Equation 2) to calculate DtD.

3.2.2 A Naïve approach

A Naïve approach is also introduced in Bharath & Shumway (2008) that does not involve iterative procedure to solve (Equation 1). The proxy is constructed as below:

Method: Proxy of Merton/KMV method (Naïve approach, Bharath & Shumway (2008))

---

1. Approximate the market value of debt with the default point DP.
2. Approximate the volatility of debt as $\sigma_D = 0.05 + 0.25\sigma_E$, where 0.05 represents term structure volatility and 25% of equity volatility is included to allow for volatility associated with default risk.
3. Approximate the total volatility as $\sigma_V = \frac{E}{E+DP}\sigma_E + \frac{DP}{E+DP}\sigma_D$

4. Approximate the return on firm's assets with the firm's stock return over the previous year r_{it-1}.

The Naïve Distance to Default is then:

$$DtD_{\text{naïve}} = \frac{\ln[(E + DP)/DP] + (r_{it-1} - 0.5\sigma_V^2)T}{\sigma_V \sqrt{T}}$$

and the Naïve default probability is:

$$PD_{\text{naïve}} = \mathcal{N}(-DtD_{\text{naïve}})$$

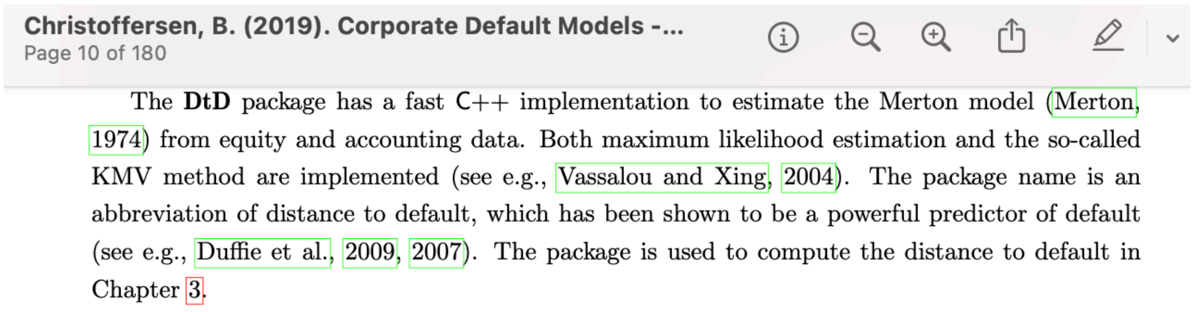
3.3 Fitting DtD in practice using Compustat/CRSP data

The blog post we mention before (see here) also refers to a `SAS` script provided in Bharath & Shumway (2008). In particular it finds some flaws in the provided script and gives an updated version.

We do not try to describe here this script (you can give it a try if you are familiar with `SAS`).

We rather re-implement in the next paragraphs the method in `R` using the algorithm description found in various articles (Crosbie & Bohn (2003), Vassalou & Xing (2004), Bharath & Shumway (2008), Duan & Wang (2012), Forssbæck & Vilhelmsson (2017)).

We then compare our implementation with a `R` package efficiently implementing Distance to Default leveraging `C++` subroutines (package here); it allows us to check if the package `DtD` is usable as is.


Christoffersen, B. (2019). Corporate Default Models -...
Page 10 of 180

The **DtD** package has a fast C++ implementation to estimate the Merton model (Merton, 1974) from equity and accounting data. Both maximum likelihood estimation and the so-called KMV method are implemented (see e.g., Vassalou and Xing, 2004). The package name is an abbreviation of distance to default, which has been shown to be a powerful predictor of default (see e.g., Duffie et al., 2009, 2007). The package is used to compute the distance to default in Chapter 3.

We first start to estimate DtD for a single firm year observation, namely `International Business Machines Corporation` (or `IBM`) for the fiscal year 2011. Figures for this firm year observation are available in Duan & Wang (2012), allowing us to check that our implementation is correct before fitting DtD on the whole data set.

3.3.1 A single firm observation

We first set IBM Compustat identifier for later usage:

```
gvkey_ibm <- '006066'
```

We briefly look at Compustat data for fiscal years 2009-2011, and compute a first estimate of *DP* the default point defined as short-term plus half of long-term debt:

```
(ibm_comp <- compustat_all %>%
    filter(gvkey == gvkey_ibm, fyear >=2009, fyear <= 2011) %>%
    select(gvkey, conm, fyear, datadate, lt, lct, dltt, dlc) %>%
    mutate(DP = lct + 0.5*(lt-lct)))
```

```
# A tibble: 3 x 9
  gvkey  conm                     fyear datadate      lt   lct  dltt   dlc     DP
  <chr>  <chr>                    <int> <date>     <dbl> <dbl> <dbl> <dbl>  <dbl>
1 006066 INTL BUSINESS MACHINES~  2009 2009-12-31 86267 36002 21932  4168 61134.
2 006066 INTL BUSINESS MACHINES~  2010 2010-12-31 90280 40562 21846  6778 65421
3 006066 INTL BUSINESS MACHINES~  2011 2011-12-31 96197 42123 22857  8463 69160
```

```
# # A tibble: 3 × 9
#   gvkey  conm                        fyear datadate      lt   lct  dltt   dlc     DP
#   <chr>  <chr>                       <int> <date>     <dbl> <dbl> <dbl> <dbl>  <dbl>
# 1 006066 INTL BUSINESS MACHINES CORP  2009 2009-12-31 86267 36002 21932  4168 61134.
# 2 006066 INTL BUSINESS MACHINES CORP  2010 2010-12-31 90280 40562 21846  6778 65421
# 3 006066 INTL BUSINESS MACHINES CORP  2011 2011-12-31 96197 42123 22857  8463 69160
```

We filter the Compustat/CRSP mapping table on IBM, we notice that mappings/links are valid between two dates, so that one gvkey can be linked to many permnos depending on the value date (here the link is unique, even if two rows are found):

```
(subset_mapping_ibm <- ccmxpf_linktable %>%
  filter(gvkey %in% gvkey_ibm) %>%
  left_join(company_all %>% select(gvkey, conm), by = c("gvkey")))
```

```
# A tibble: 2 x 5
  permno gvkey  linkdt     linkenddt  conm
   <dbl> <chr>  <date>     <date>     <chr>
1  12490 006066 1950-01-01 1962-01-30 INTL BUSINESS MACHINES CORP
2  12490 006066 1962-01-31 2024-10-12 INTL BUSINESS MACHINES CORP
```

We then merge our daily CRSP data set with the preceding mapping:

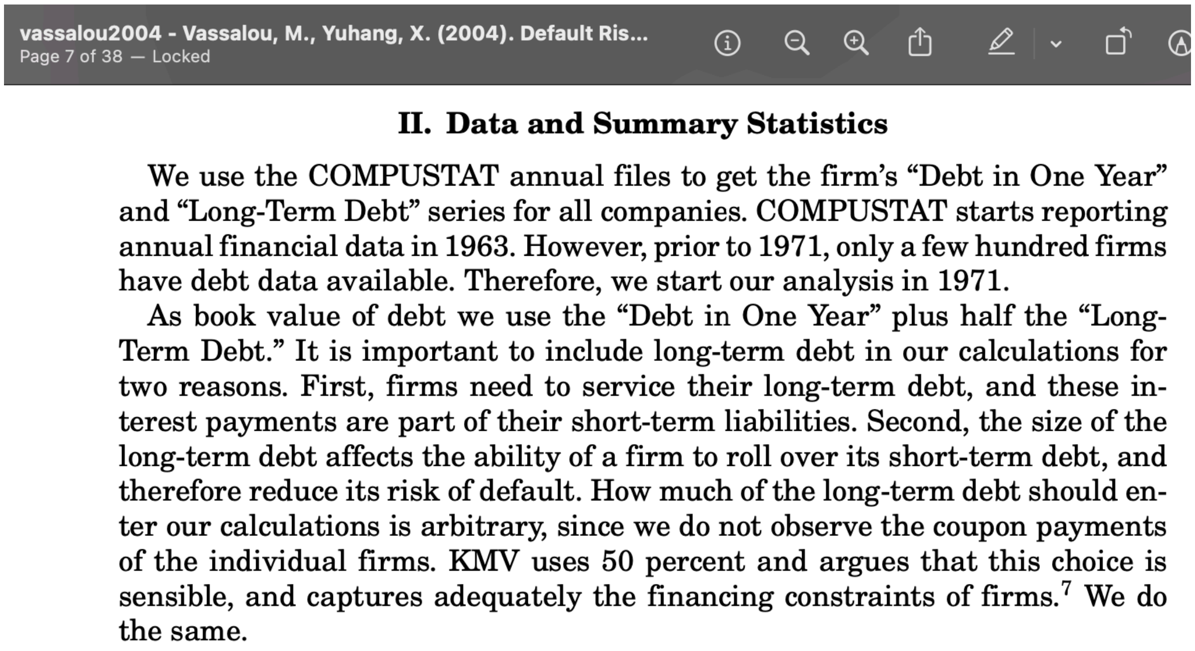
```
(merged_crsp_compustat_ibm <- crsp_daily %>%
  select(permno, date, prc, shrout, ret) %>%
  left_join(subset_mapping_ibm,
            by = join_by(permno,
                         between(date, linkdt, linkenddt))) %>%
    filter(!is.na(gvkey)))
```

```
# A tibble: 16,108 x 9
   permno date         prc shrout      ret gvkey  linkdt     linkenddt  conm
    <dbl> <date>     <dbl>  <int>    <dbl> <chr>  <date>     <date>     <chr>
 1  12490 1960-01-04  437   18268 -0.00285 006066 1950-01-01 1962-01-30 INTL BU~
 2  12490 1960-01-05  440.  18268  0.00801 006066 1950-01-01 1962-01-30 INTL BU~
 3  12490 1960-01-06  442.  18268  0.00284 006066 1950-01-01 1962-01-30 INTL BU~
 4  12490 1960-01-07  441   18268 -0.00170 006066 1950-01-01 1962-01-30 INTL BU~
 5  12490 1960-01-08  437   18268 -0.00907 006066 1950-01-01 1962-01-30 INTL BU~
 6  12490 1960-01-11  428.  18268 -0.0217  006066 1950-01-01 1962-01-30 INTL BU~
 7  12490 1960-01-12  424.  18268 -0.00936 006066 1950-01-01 1962-01-30 INTL BU~
 8  12490 1960-01-13  424   18268  0.00118 006066 1950-01-01 1962-01-30 INTL BU~
 9  12490 1960-01-14  428   18268  0.00943 006066 1950-01-01 1962-01-30 INTL BU~
10  12490 1960-01-15  428   18268  0       006066 1950-01-01 1962-01-30 INTL BU~
# i 16,098 more rows
```

We start preparing Compustat data to be used in conjunction with CRSP.

First we compute Default Point DP using various definitions from less to more conservative:

- article Vassalou & Xing (2004) definition: $DP = dlc + \frac{1}{2}dltt$

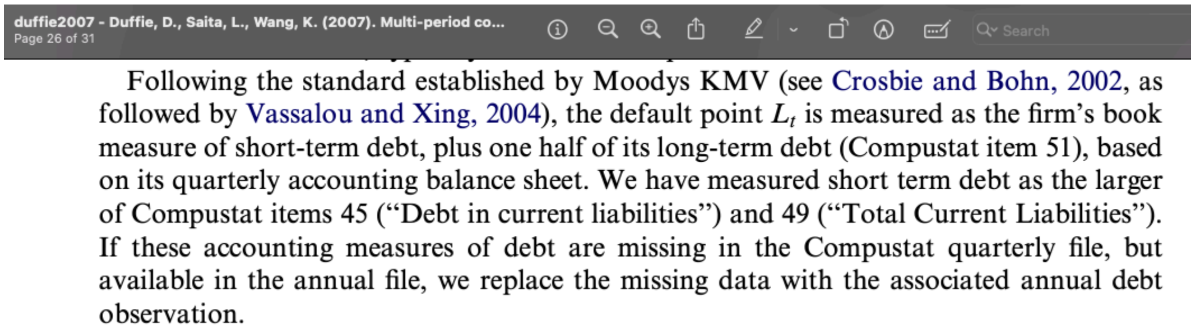

vassalou2004 - Vassalou, M., Yuhang, X. (2004). Default Ris...
Page 7 of 38 — Locked

II. Data and Summary Statistics

We use the COMPUSTAT annual files to get the firm's "Debt in One Year" and "Long-Term Debt" series for all companies. COMPUSTAT starts reporting annual financial data in 1963. However, prior to 1971, only a few hundred firms have debt data available. Therefore, we start our analysis in 1971.

As book value of debt we use the "Debt in One Year" plus half the "Long-Term Debt." It is important to include long-term debt in our calculations for two reasons. First, firms need to service their long-term debt, and these interest payments are part of their short-term liabilities. Second, the size of the long-term debt affects the ability of a firm to roll over its short-term debt, and therefore reduce its risk of default. How much of the long-term debt should enter our calculations is arbitrary, since we do not observe the coupon payments of the individual firms. KMV uses 50 percent and argues that this choice is sensible, and captures adequately the financing constraints of firms.[7] We do the same.

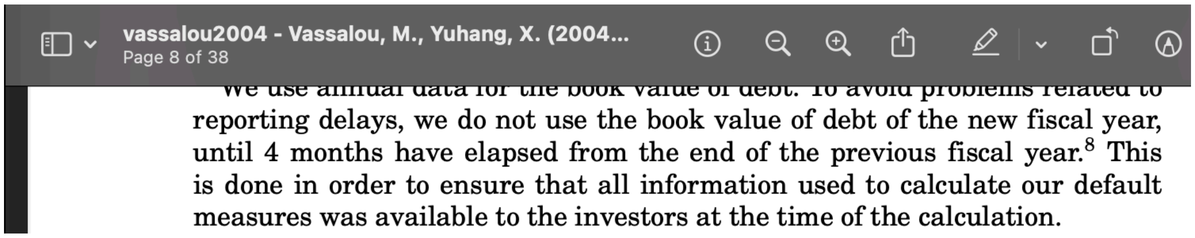
- article Duffie et al. (2007) definition $DP = \max(dlc, lct) + \frac{1}{2} dltt$


duffie2007 - Duffie, D., Saita, L., Wang, K. (2007). Multi-period co...
Page 26 of 31
Search

Following the standard established by Moodys KMV (see Crosbie and Bohn, 2002, as followed by Vassalou and Xing, 2004), the default point L_t is measured as the firm's book measure of short-term debt, plus one half of its long-term debt (Compustat item 51), based on its quarterly accounting balance sheet. We have measured short term debt as the larger of Compustat items 45 ("Debt in current liabilities") and 49 ("Total Current Liabilities"). If these accounting measures of debt are missing in the Compustat quarterly file, but available in the annual file, we replace the missing data with the associated annual debt observation.

- article Crosbie & Bohn (2003) definition $DP = lct + \frac{1}{2}(lt - lct)$

Also following practical implementation in Vassalou & Xing (2004), we ensure that at DtD evaluation date, accounting data is available for at least 3 month since the end of last fiscal year. Then this accounting data is valid for 1 year.



We use annual data for the book value of debt. To avoid problems related to reporting delays, we do not use the book value of debt of the new fiscal year, until 4 months have elapsed from the end of the previous fiscal year.[8] This is done in order to ensure that all information used to calculate our default measures was available to the investors at the time of the calculation.

[8] The SEC requires firms to report 10K within three months after the end of the fiscal year, but a small percentage of firms report it with a longer delay.

```
(compustat_ibm <- compustat_all %>%
  filter(gvkey == gvkey_ibm, fyear >=2009, fyear <= 2011) %>%
  mutate(act = coalesce(act, che + rect + invt + aco),
         at = coalesce(at, act + ppent + ivaeq + ivao + intan + ao),
         lct = coalesce(lct, ap + txp + dlc + lco),
         wcap = coalesce(wcap, act - lct),
         lt = coalesce(lt, lct + dltt + txditc + lo),
         # We test below various versions of Default Point definition
         # From less to more conservative
         DP_vassalou = coalesce(dlc, lct) + 0.5 * pmin(dltt, lt - lct, na.rm=T),
         # Vassalou (2004) p837 As book value of debt we use the "Debt in One Year" [dlc]
         # plus half the "Long- Term Debt." [dltt].
         # careful dlc /dltt = 0 sometimes leading Inf DtD
         # (eg permno 76671 gvkey 021828 fyear 2005)
         DP_duffie = pmax(dlc, lct, na.rm=TRUE) + 0.5 * pmin(dltt, lt - lct, na.rm=T),
         # Duffie (2007) Short-term debt is estimated as the larger of Compustat items DAT
         # and DATA49 [lct].
         # Long-term debt is taken from DATA51 [dltt].
         DP_vassalou = if_else(DP_vassalou<1e-6, DP_duffie, DP_vassalou),
         # See Crosbie (2003)
         # Philip Morris / Altria eoy 2000
         # LT = 64Bn / Default Point = 47.5 = LCT + 0.5*(LT-LCT)
         DP_kmv = pmax(dlc, lct, na.rm=TRUE) + 0.5 * (lt - lct),
         start_valid = datadate %m+% months(3),
         end_valid = start_valid %m+% years(1),
         start_valid = start_valid + 1) %>% # avoid overlapping periods
  select(gvkey, conm, fyear, fyr, start_valid, end_valid, datadate, D=lt, DP_vassalou, DP_
```

```
# A tibble: 3 x 11
  gvkey  conm    fyear   fyr start_valid end_valid  datadate        D DP_vassalou
  <chr>  <chr>   <int> <int> <date>      <date>     <date>      <dbl>       <dbl>
1 006066 INTL B~  2009    12 2010-04-01  2011-03-31 2009-12-31 86267       15134
2 006066 INTL B~  2010    12 2011-04-01  2012-03-31 2010-12-31 90280       17701
3 006066 INTL B~  2011    12 2012-04-01  2013-03-31 2011-12-31 96197       19892.
# i 2 more variables: DP_duffie <dbl>, DP_kmv <dbl>
```

Sometimes accounting periods change for companies (eg for GAS NATURAL INC in 2008 with closing moving from 30 June to 31 Dec):

```
compustat_all %>%
  filter(gvkey=='012994', fyear>=2006, fyear<=2009) %>%
  select(conm, gvkey, datadate, fyear, fyr)
```

```
# A tibble: 4 x 5
  conm             gvkey  datadate   fyear   fyr
  <chr>            <chr>  <date>     <int> <int>
1 GAS NATURAL INC 012994 2006-06-30  2006     6
2 GAS NATURAL INC 012994 2007-06-30  2007     6
3 GAS NATURAL INC 012994 2008-12-31  2008    12
4 GAS NATURAL INC 012994 2009-12-31  2009    12
```

so we extend accounting data validity date to avoid data gap, when merging CRSP and Compustat for *DtD* computations (not useful here for `IBM`, but for a bigger data set) :

```
compustat_ibm <- compustat_ibm %>%
                        mutate(check_end = lead(start_valid),
                               end_valid = if_else(check_end-1!=end_valid & !is.na(check_e
                                                   check_end-1,
                                                   end_valid)) %>%
                        select(-check_end)
```

We then estimate Equity historical volatility, we show estimate at End of Year 2011 that roughly Duan & Wang (2012) figures:

```
ibm_data_dtd <- merged_crsp_compustat_ibm %>%
    left_join(compustat_ibm %>% select(-fyear, -fyr),
              by = join_by(gvkey,
                           between(date, start_valid, end_valid))) %>%
    mutate(E = shrout * prc / 1000,
           year = lubridate::year(date)) %>%
    group_by(gvkey, year) %>%
    arrange(date) %>%
    mutate(log_ret = log(prc/lag(prc)),
           n = n(),
           sig_E_l = sd(log_ret, na.rm = TRUE)*sqrt(252), # assuming 252 business days in
           sig_E = sd(ret, na.rm = TRUE)*sqrt(252)) %>% # assuming 252 business days in 1Y
    filter(n >= 60) %>% # at least 60 days to estimate volatility and DtD
    ungroup() %>%
    select(permno, gvkey, date, start_valid, end_valid, D, DP_vassalou, DP_duffie, DP_kmv,
    filter(!is.na(start_valid))

ibm_data_dtd %>%
  filter(date==lubridate::as_date('2011-12-30')) %>%
  select(date, E, DP_duffie, sig_E_l, sig_E)
```

```
# A tibble: 1 x 5
  date              E DP_duffie sig_E_l sig_E
  <date>        <dbl>     <dbl>   <dbl> <dbl>
1 2011-12-30 216724.     51485   0.225 0.224
```

We have access to E, DP, σ_E, but we lack the risk-free rate r, it is usually defined as the One-Year U.S. Treasury Bill available from FRED:

```
# Vassalou (2004) / Shumway (2008) risk free rate

treasury_1Y <- readRDS("./wrds_data/treasury_1Y.rds")
# # The data has been previously collected this way:
# treasury_1Y_url <- 'https://fred.stlouisfed.org/series/DGS1/downloaddata/DGS1.csv'
# treasury_1Y <- readr::read_csv(treasury_1Y_url, col_types = cols(DATE = "D", VALUE = "d"
#   filter(!is.na(VALUE))
#
# saveRDS(treasury_1Y, "./wrds_data/treasury_1Y.rds")
```

We join we our data set for `IBM`:

```
ibm_data_dtd <- ibm_data_dtd %>%
  left_join(treasury_1Y %>%
              mutate(r = VALUE / 100) %>%
              select(-VALUE), by= c("date"="DATE")) %>%
  fill(r, .direction = "downup")
```

We implement below the algorithm as described in Forssbæck & Vilhelmsson (2017):

has later been employed by, e.g., Bharath and Shumway (2008).[3] The iterative algorithm starts by proposing an initial value for $\sigma_V$, which we set equal to $\sigma_V^{(0)} = \sigma_E \frac{E}{E+D}$, where σ_E is equity return volatility measured as previously, E is the market value of equity at the end of the year, and D is the book value of debt defined as in (5). The superscripted parenthesis shows the iteration number, with (0) denoting initial values. We also need an initial value for the firm's asset value, which we set to $V^{(0)} = E + D$. Finally, we set $r_A^{(0)} = r$. Given these estimates and initial values, we can then solve equation (1) for V to obtain first-iteration asset values for each month in a year as

$$V_t^{(1)} = \frac{E + De^{-rT} N(d_2^{(0)})}{N(d_1^{(0)})}, \tag{6}$$

with $d_1^{(0)} = \frac{\ln(V^{(0)}/D) + (r_A^{(0)} + 0.5\sigma_V^{2(0)})T}{\sigma_V^{(0)}\sqrt{T}}$ and $d_2^{(0)} = d_1^{(0)} - \sigma_V^{(0)}\sqrt{T}$. We then calculate $r_A^{(1)}$ as the growth over each year in the inferred asset values $\{V^{(1)}\}_1^{12}$, and $\sigma_V^{(1)}$ as the annualized standard deviation of the same inferred series. We then reiterate and compute a new series of inferred asset values $\{V^{(2)}\}_1^{12}$ using the parameters estimated in the first iteration. This procedure is repeated until the difference in updated asset volatilities is "sufficiently" small – that is, until $|\sigma_V^{(n)} - \sigma_V^{(n-1)}| < \varepsilon$, where ε is the chosen maximum acceptable difference. After some experimentation we pick $\varepsilon = 0.001$, since a smaller value increases estimation time but has virtually no effect on the estimated distance to default. We refer to distance to default estimated using this iterative method as Merton VX.

```
DtD_estimation <- function(daily_market_accounting_data, DP_var ="DP_duffie", verbose=FALS
# takes a vector of daily equity prices  and accounting data for a firm year.
# e.g. data <- vassalou_data_dtd %>% filter(permno==12490, lubridate::year(date)==2011)


# The iterative algorithm starts by proposing an initial value for $\sigma_V$,
# which we set equal to $\sigma_V^{(0)}=\sigma_E\frac{E}{E+D}$ , where
# $sigma_E$ is equity return volatility measured as previously,
```

```r
# $E$ is the market value of equity at the end of the year,
# and $D$ is the book value of debt defined as $D=ST+\frac{1}{2}LT$


# Checking enough non NA accounting variable (>60 days)
DP_var <- as.name(DP_var)
n_test <- nrow(daily_market_accounting_data %>% filter(!is.na(!!DP_var)))
if(n_test > 60){
  # if yes, forward/fill + filter NA
  daily_market_accounting_data <- daily_market_accounting_data %>%
    fill(!!DP_var, .direction = "down") %>%
    filter(!is.na(!!DP_var))
} else {
  # if no return empty result
  if(verbose){print(glue::glue(
    "failed to fit gvkey:{daily_market_accounting_data[1,] %>% pull(gvkey)}, no data"))}

  return(tibble(sig_V = NA_real_,
              mu_V = NA_real_,
              V = NA_real_,
              DP_eoy = NA_real_,
              E_eoy = NA_real_,
              DtD = NA_real_,
              DtD_kmv = NA_real_,
              PD = NA_real_,
              nb_iter = NA_real_,
              error = NA_real_))

}

# We also need an initial value for the firm's asset value,
# which we set to $V^{(0)}=E+D$.

eoy_data <- daily_market_accounting_data %>% filter(date==max(date))
E_eoy <- eoy_data %>% pull(E)

D_eoy <- eoy_data %>% pull(!!DP_var)
V_eoy <- E_eoy + D_eoy
sig_V <- eoy_data %>% pull(sig_E) * E_eoy / V_eoy

iter_max <- 15
```

```r
T. <- 1

# Initialize V
data <- daily_market_accounting_data %>%
  mutate(V = E + !!DP_var) # could be V = V_eoy

for(i in 1:iter_max){
# Calculate d1, d2, V(V_{i-1}) for all days
  data <- data %>%
    mutate(
           d1 = (log(V / !!DP_var) + (r + (sig_V^2) / 2) * T.) / (sig_V * sqrt(T.)),
           d2 = d1 - sig_V * sqrt(T.),
           # forssbaeck2017 differs slightly from pure kmv (DtD package) method at this st
           # instead of inverting Black Scholes to get V (solving implicit non-linear equa
           # it just "solves" the equation using last evaluated V value for d1 and d2
           # E = V x N(d1) + DPexp(-rT) x N(d2) => V_i+1 =(E + DPexp(-rT)xN(d2(V_i))) / N(
           V = (E + !!DP_var * exp(-r*T.) * pnorm(d2))/pnorm(d1),
           log_ret = log(V/lag(V))
          )

  new_estimate <- data %>% summarise(n = n(),
                                     sig_V = sd(log_ret, na.rm=TRUE) * sqrt(252),
                                     mu_V = sum(log_ret, na.rm=T) / T. + 0.5 * sig_V * sig_V)

  sig_V_new <- new_estimate %>% pull(sig_V)

  if(verbose){
    print(i)
    print(glue::glue('old sig_V: {sig_V}'))
    print(glue::glue('new sig_V: {sig_V_new}'))
  }

  err <- sig_V-sig_V_new

  if(abs(err) < 1e-4){
    # Compute V then DtD for last year date
    mu_V <- new_estimate %>% pull(mu_V)
    data <- data %>%
      filter(date==max(date)) %>%
      mutate(
             d1 = (log(V / !!DP_var) + (r + (sig_V^2) / 2) * T.) / (sig_V * sqrt(T.)),
                d2 = d1 - sig_V * sqrt(T.),
                # From Black Scholes formula Equity value=BS(Asset Value)
                V = (E + !!DP_var * exp(-r*T.) * pnorm(d2)) / pnorm(d1),
                DtD = (log(V / !!DP_var) + (mu_V - sig_V^2 / 2) * T.) / (sig_V * sqrt(T.)),
                # KMV approx without physical drift mu for better "stability"
                DtD_kmv = log(V / !!DP_var) / (sig_V * sqrt(T.)),
                PD = pnorm(-DtD)
               )

    return(tibble(sig_V = sig_V_new,
                  mu_V = new_estimate %>% pull(mu_V),
                  V = data %>% pull(V),
                  DP_eoy = D_eoy,
                  E_eoy = E_eoy,
                  DtD = data %>% pull(DtD),
                  DtD_kmv = data %>% pull(DtD_kmv),
                  PD = data %>% pull(PD),
                  nb_iter = i,
                  error = err))
  }

  sig_V <- sig_V_new
}

data <- data %>%
      filter(date==max(date)) %>%
      mutate(
                d1 = (log(V / !!DP_var) + (r + (sig_V^2) / 2) * T.) / (sig_V * sqrt(T.)),
                d2 = d1 - sig_V * sqrt(T.),
                V = (E + !!DP_var * exp(-r*T.) * pnorm(d2)) / pnorm(d1),
                DtD = (log(V / !!DP_var) + (mu_V - sig_V^2 / 2) * T.) / (sig_V * sqrt(T.)),
                DtD_kmv = log(V / !!DP_var) / (sig_V * sqrt(T.)), # KMV approx without physic
                PD = pnorm(-DtD)
               )

return(tibble(sig_V = sig_V_new,
            mu_V = new_estimate %>% pull(mu_V),
            V = data %>% pull(V),
            DP_eoy = D_eoy,
            E_eoy = E_eoy,
            DtD = data %>% pull(DtD),
                DtD_kmv = data %>% pull(DtD_kmv),
                PD = data %>% pull(PD),
                nb_iter = i,
                error = err))
}
```

We check the implementation against Duan & Wang (2012) which provides figures estimated for IBM at the end of year 2011:

```
data_IBM_2011 <- ibm_data_dtd %>% filter(lubridate::year(date)==2011)
```

We obtain the following figures:

```
(res_2011 <- DtD_estimation(data_IBM_2011))
```

```
# A tibble: 1 x 10
  sig_V  mu_V        V DP_eoy    E_eoy   DtD DtD_kmv        PD nb_iter error
  <dbl> <dbl>    <dbl>  <dbl>    <dbl> <dbl>   <dbl>     <dbl>   <int> <dbl>
1 0.184 0.170 268148.   51485 216724.  9.78    8.95 6.80e-23       2     0
```

```
# # A tibble: 1 × 10
#   sig_V  mu_V        V DP_eoy    E_eoy   DtD DtD_kmv        PD nb_iter error
#   <dbl> <dbl>    <dbl>  <dbl>    <dbl> <dbl>   <dbl>     <dbl>   <int> <dbl>
# 1 0.184 0.170 268148.   51485 216724.  9.78    8.95 6.80e-23       2     0
```

which is comparable with figures found in Duan & Wang (2012) (Table 1, first column, panel "The KMV Method"):

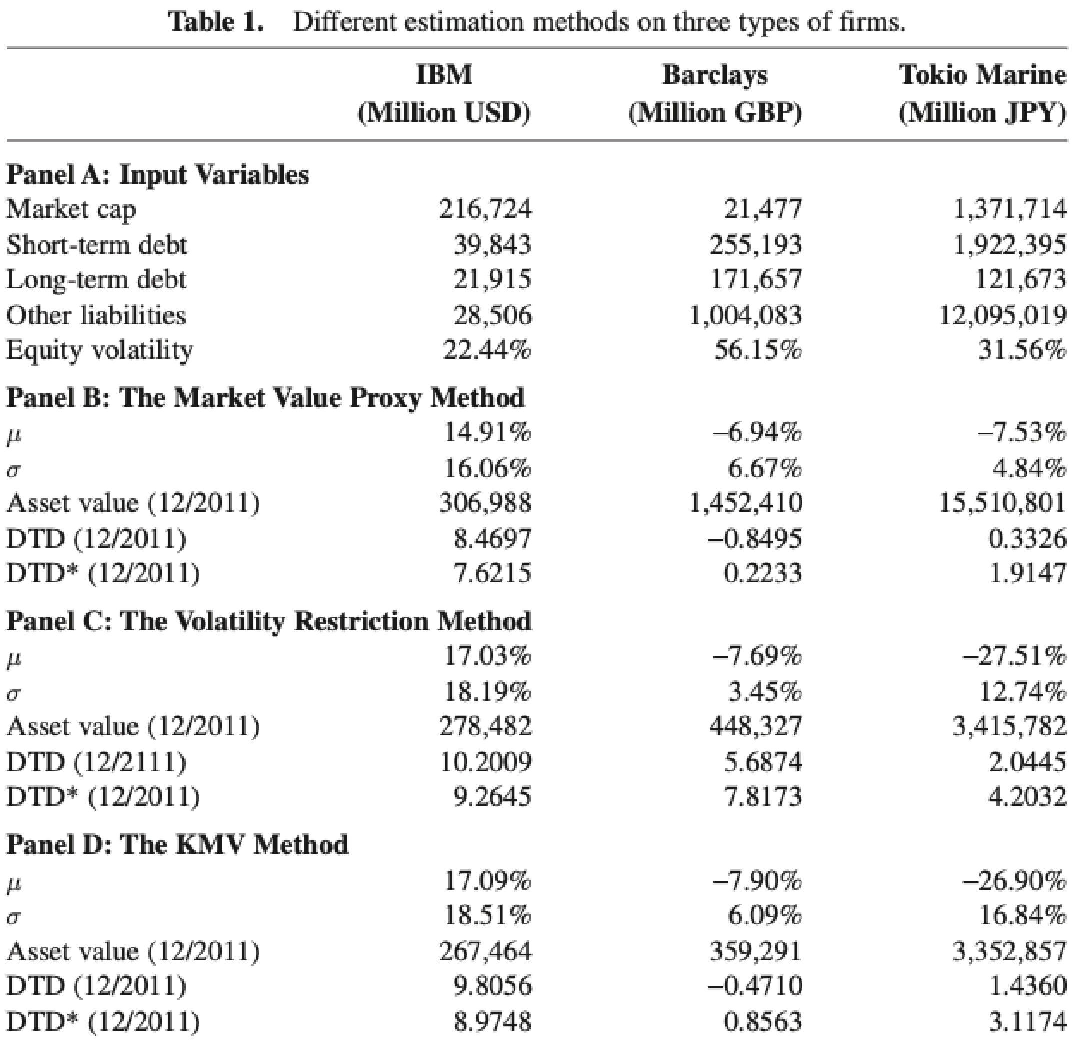


Table 1. Different estimation methods on three types of firms.

	IBM (Million USD)	Barclays (Million GBP)	Tokio Marine (Million JPY)
Panel A: Input Variables			
Market cap	216,724	21,477	1,371,714
Short-term debt	39,843	255,193	1,922,395
Long-term debt	21,915	171,657	121,673
Other liabilities	28,506	1,004,083	12,095,019
Equity volatility	22.44%	56.15%	31.56%
Panel B: The Market Value Proxy Method			
μ	14.91%	–6.94%	–7.53%
σ	16.06%	6.67%	4.84%
Asset value (12/2011)	306,988	1,452,410	15,510,801
DTD (12/2011)	8.4697	–0.8495	0.3326
DTD* (12/2011)	7.6215	0.2233	1.9147
Panel C: The Volatility Restriction Method			
μ	17.03%	–7.69%	–27.51%
σ	18.19%	3.45%	12.74%
Asset value (12/2011)	278,482	448,327	3,415,782
DTD (12/2111)	10.2009	5.6874	2.0445
DTD* (12/2011)	9.2645	7.8173	4.2032
Panel D: The KMV Method			
μ	17.09%	–7.90%	–26.90%
σ	18.51%	6.09%	16.84%
Asset value (12/2011)	267,464	359,291	3,352,857
DTD (12/2011)	9.8056	–0.4710	1.4360
DTD* (12/2011)	8.9748	0.8563	3.1174

We also retrieve similar figures using `DtD` package implemented by the author of Christoffersen (2019). This validates the latter usage of this package which is more efficient than our implementation:

```r
n_IBM_2011 <- nrow(data_IBM_2011)

# Estimating \mu_V and \sigma_V on past year Equity data
(ibm_est <- DtD::BS_fit(S = data_IBM_2011$E,
                        D = data_IBM_2011$DP_duffie,
                        T. = 1,
                        dt = 1 / n_IBM_2011,
                        r = data_IBM_2011$r,
                        vol = data_IBM_2011$sig_E[n_IBM_2011] * data_IBM_2011$E[n_IBM_2011
                             (data_IBM_2011$E[n_IBM_2011] + data_IBM_2011$DP_duffie[n_IBM_
                        method = "iterative"))$ests
```

```
       mu       vol
0.1707592 0.1840194
```

```
# Getting Asset Value from End of Year 2011 equity / Accounting data
VA <- DtD::get_underlying(S = data_IBM_2011$E[n_IBM_2011],
                          D = data_IBM_2011$DP_duffie[n_IBM_2011],
                          T. = 1,
                          r = data_IBM_2011$r[n_IBM_2011],
                          vol = ibm_est$ests['vol'])
(glue::glue('asset value: {round(VA,2)}'))
```

```
asset value: 268147.53
```

```
# Computing Distance to Default at the End of Year 2011
DD <- (log(VA/data_IBM_2011$DP_duffie[n_IBM_2011]) + (ibm_est$ests[['mu']]-0.5*ibm_est$est
(glue::glue('distance to default: {round(DD,3)}'))
```

```
distance to default: 9.804
```

```
#        mu       vol
# 0.1707592 0.1840194
# asset value: 268147.53
# distance to default: 9.804
```

It would be better to use quarterly financial data for Debt/Default Point figures (but also more computationaly intensive plus involving additional data quality checks when fitting the whole data set). We retrieve below the exact figures which were used in Duan & Wang (2012) which relate to Q2-2011/HY-2011 for the End of Year 2011 estimation while we have used Q4-2010 data:

```
compustat_allq_ibm <- readRDS("./wrds_data/compustat_allq_ibm.rds")

# Obtained that way using Compustat Fundamental Quarterly files:
```

```
# compustat_allq_ibm <- readRDS("./wrds_data/compustat_allq.rds") %>%
#                        filter(gvkey=='006066', fyearq %in% c(2010, 2011, 2012)) %>%
#                        select(gvkey, conm, fyearq, fqtr, datadate,
#                               `Short-term debt` = lctq,
#                               dlcq,
#                               ltq,
#                               `Long-term debt` = dlttq,
#                               `Other liabilities` = loq)
#
# saveRDS(compustat_allq_ibm, "./wrds_data/compustat_allq_ibm.rds")

# In @duan2012, authors use Q2-2011 data as End of Year 2011 debt / default point figures,
# while we have used Q4-2010 / FY-2010 data
# Usually, the debt amounts evolve more slowly (with more inertia) than the stock price,
# so it is a good approximation to use annual debt figures:
compustat_allq_ibm %>%
  filter(datadate %in% c(lubridate::as_date('2010-12-31'),
                         lubridate::as_date('2011-06-30'))) %>%
  select(conm, fyearq, fqtr, datadate, `Short-term debt`, `Long-term debt`, `Other liabili
```

```
# A tibble: 2 x 7
  conm                fyearq  fqtr datadate   `Short-term debt` `Long-term debt`
  <chr>                <dbl> <dbl> <date>                 <dbl>            <dbl>
1 INTL BUSINESS MACH~   2010     4 2010-12-31             40562            21846
2 INTL BUSINESS MACH~   2011     2 2011-06-30             39843            21915
# i 1 more variable: `Other liabilities` <dbl>
```

```
# # A tibble: 2 × 7
#   conm                        fyearq  fqtr datadate   `Short-term debt` `Long-term debt`
#   <chr>                        <dbl> <dbl> <date>                 <dbl>            <dbl>
# 1 INTL BUSINESS MACHINES CORP   2010     4 2010-12-31             40562            21846
# 2 INTL BUSINESS MACHINES CORP   2011     2 2011-06-30             39843            21915
```

Duan & Wang (2012) uses Compustat Fundamental Quarterly data:

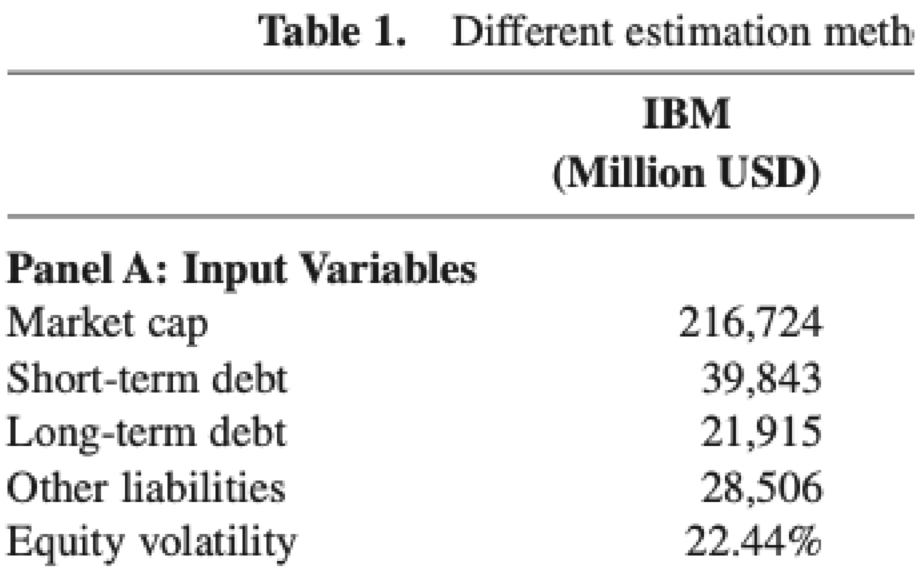


Table 1. Different estimation meth

	IBM (Million USD)
Panel A: Input Variables	
Market cap	216,724
Short-term debt	39,843
Long-term debt	21,915
Other liabilities	28,506
Equity volatility	22.44%

In the next paragraphs, we will implement some modifications relating to Equity volatility estimation (coping with possible outliers in daily returns and uneven spacing of prices time series). For later comparison we show below the effect of that corrected Equity volatility estimation (lowering IBM 2011 Equity volatility from ca. 22.8% to ca. 21%):

```r
library(bizdays)
load_rmetrics_calendars(1960:2030)

vol_equity_estimation <- data_IBM_2011 %>%
  mutate(ret = E/lag(E),
         log_ret = log(ret),
         lagDt = lag(date),
         sig_E_before = sd(ret, na.rm = TRUE)*sqrt(252),
         sig_E_l_before = sd(log_ret, na.rm = TRUE)*sqrt(252)) %>%
  filter(!is.na(lagDt)) %>%
  mutate(log_ret_min = quantile(log_ret, probs=.005, na.rm = TRUE), # winsorizing log_retu
         log_ret_max = quantile(log_ret, probs=.995, na.rm = TRUE), # cf (gvkey 027855 yea
         # dt = bizdays(lagDt, date, "Rmetrics/NYSE"),
         # put a floor at 1 business day to deal with mkt holidays (not flagged by CRSP)
         # cf (permno 12490, year 1980, 30/31 December)
         dt = pmax(1, bizdays(lagDt, date, "Rmetrics/NYSE")),
         time = cumsum(dt/252),
         sq_dt = sqrt(dt)) %>%
  filter(log_ret>log_ret_min, log_ret<log_ret_max) %>%
  mutate(
        # sq_dt = if_else(n>1,sqrt(bizdays(lagDt, date, "Rmetrics/NYSE")), 1),
        sig_E_l = sd(log_ret, na.rm = TRUE)*sqrt(252), # assuming 252 business days in 1Y
        # /!\ not all CRSP data is daily, sometimes uneven, overestimates volatilities
        # https://quant.stackexchange.com/questions/42407/estimating-daily-volatility-of-u
        sig_E_l_adj = sd(log_ret/sq_dt, na.rm = TRUE)*sqrt(252), # correcting for unequal
        sig_E = sd(ret, na.rm = TRUE)*sqrt(252))
vol_equity_estimation %>%
  tail(1) %>%
  select(gvkey, start_valid, end_valid, DP_vassalou, DP_duffie, E, sig_E_before, sig_E, si
```

```
# A tibble: 1 x 11
  gvkey  start_valid end_valid  DP_vassalou DP_duffie       E sig_E_before sig_E
  <chr>  <date>      <date>           <dbl>     <dbl>   <dbl>        <dbl> <dbl>
1 006066 2011-04-01  2012-03-31       17701     51485 216724.        0.228 0.209
# i 3 more variables: sig_E_l_before <dbl>, sig_E_l_adj <dbl>, r <dbl>
```

```
# # A tibble: 1 × 11
#   gvkey  start_valid end_valid  DP_vassalou DP_duffie       E sig_E_before sig_E sig_E_l
#   <chr>  <date>      <date>           <dbl>     <dbl>   <dbl>        <dbl> <dbl>
# 1 006066 2011-04-01  2012-03-31       17701     51485 216724.        0.228 0.209
```

3.3.2 Vassalou (2004) reference implementation (see Vassalou & Xing (2004))

In Vassalou & Xing (2004), authors perform "the first study that uses Merton's (1974) option pricing model to compute default measures for individual firms and assess the effect of default risk on equity returns."

They provide a data set of fitted default probabilities (as $PD_{Merton} = \mathcal{N}(-DtD)$) for "firm year" observations giving CRSP identifiers and covering 1970-1999 for a subset of CRSP/Compustat.

This allows to briefly check the global correctness of our fitted DtD figures on a broader scope as done in Christoffersen (2019):

[2] The probabilities of default in Vassalou and Xing (2004) are available at **www.maria-vassalou.com/data/defaultdataset.zip**. Comparing our distance to default to theirs over the same period after truncating at a 10^{-15} and $1 - 10^{-15}$ probability of default as they do yields a mean and standard deviation of the distance to default of 4.856 and 2.739 , respectively. The corresponding figures in Vassalou and Xing (2004) are 4.391 and 2.608, respectively.

[3] Chava et al. (2011), Duan et al. (2012), Lando et al. (2013), Lando and Nielsen (2010), Qi et al. (2014) show a mean ranging from 1.867 to 16.79 and a standard deviation ranging from 2.653 to 12.83.

We first import Vassalou's data set:

```
# http://www.maria-vassalou.com/data.html
# http://www.maria-vassalou.com/data/defaultdataset.zip

vassalou <- readr::read_fwf("./default_data/Vassalou/defaultdataset.txt")
names(vassalou) <- c("Year", "Month", "Permno", "Default Probability", "Log Value of Asset

vassalou <- vassalou  %>% mutate(`Distance to Default` = -qnorm(`Default Probability`))

# reducing scope a little bit filtering firms still present in 1990
recent_vassalou <- vassalou %>%
  filter(Year>=1990) %>%
  distinct(Permno)

vassalou_perimeter <- ccmxpf_linktable %>%
  filter(permno %in% (recent_vassalou %>% pull(Permno))) %>%
  left_join(company_all %>% select(gvkey, conm), by = c("gvkey"))
```

Using the approach described in Christoffersen (2019) footnote shown just before, we retrieve comparable summary statistics (ie 4.37 mean and 2.6 standard deviation DtD after winsorizing the figures):

```
vassalou %>%
  mutate(`Default Probability` =
           if_else(`Default Probability` < 1e-15,
                   1e-15,
                   if_else(`Default Probability` > 1 - 1e-15,
                           1 - 1e-15,
                           `Default Probability`))) %>%
  mutate(`Distance to Default`=-qnorm(`Default Probability`)) %>%
  summarise(m = mean(`Distance to Default`, na.rm=TRUE),
            s = sd(`Distance to Default`, na.rm=TRUE))
```

```
# A tibble: 1 x 2
      m     s
  <dbl> <dbl>
1  4.37  2.57
```

```
# # A tibble: 1 × 2
#       m     s
#   <dbl> <dbl>
# 1  4.37  2.57
```

We comment below the code to fit DtD on Vassalou's perimeter as it takes some time to run:

```
# merged_crsp_compustat_vassalou <- crsp_daily %>%
#   select(permno, date, prc, shrout, ret) %>%
#   left_join(vassalou_perimeter,
#             by = join_by(permno,
#                          between(date, linkdt, linkenddt))) %>%
#   filter(!is.na(gvkey))
```

```
# compustat_vassalou <- compustat_all %>%
#   filter(gvkey %in% (vassalou_perimeter %>% distinct(gvkey) %>% pull(gvkey)))
#
# compustat_vassalou <- compustat_vassalou %>%
#   mutate(act = coalesce(act, che + rect + invt + aco),
#          at = coalesce(at, act + ppent + ivaeq + ivao + intan + ao),
#          lct = coalesce(lct, ap + txp + dlc + lco),
#          wcap = coalesce(wcap, act - lct),
#          lt = coalesce(lt, lct + dltt + txditc + lo),
#          # We test below various versions of Default Point definition
#          # From less to more conservative
#          DP_vassalou = coalesce(dlc, lct) + 0.5 * pmin(dltt, lt - lct, na.rm=T),
#          # Vassalou (2004) p837 As book value of debt we use the "Debt in One Year" [dlc
#          # plus half the "Long- Term Debt." [dltt].
#          # careful dlc /dltt = 0 sometimes leading Inf DtD
#          # (eg permno 76671 gvkey 021828 fyear 2005)
#          DP_duffie = pmax(dlc, lct, na.rm=TRUE) + 0.5 * pmin(dltt, lt - lct, na.rm=T),
#          # Duffie (2007) Short-term debt is estimated as the larger of Compustat items D
#          # and DATA49 [lct].
#          # Long-term debt is taken from DATA51 [dltt].
#          DP_vassalou = if_else(DP_vassalou<1e-6, DP_duffie, DP_vassalou),
#          # See Crosbie (2003)
#          # Philip Morris / Altria eoy 2000
#          # LT = 64Bn / Default Point = 47.5 = LCT + 0.5*(LT-LCT)
#          DP_kmv = pmax(dlc, lct, na.rm=TRUE) + 0.5 * (lt - lct),
#          start_valid = datadate %m+% months(3),
#          end_valid = start_valid %m+% years(1),
#          start_valid = start_valid + 1) %>% # avoid overlapping periods
#   select(gvkey, conm, fyear, fyr, start_valid, end_valid, datadate, D=lt, DP_vassalou, D
# # Increasing month of fiscal year end:
# compustat_vassalou %>%
#   filter(gvkey=='012994', fyear>2006, fyear<2009) %>%
#   mutate(check_end = lead(start_valid),
#          end_valid = if_else(check_end - 1 > end_valid & !is.na(check_end), check_end -
```

```
# # Decreasing month of fiscal year end:
# compustat_vassalou %>%
#   filter(gvkey=='013468', fyear>=1994) %>%
#   mutate(check_end = lead(start_valid),
#          end_valid = if_else(check_end - 1 > end_valid & !is.na(check_end), check_end -
```

```
# # Ensuring validity dates quasi-overlap:
# compustat_vassalou <- compustat_vassalou %>%
#                         mutate(check_end = lead(start_valid),
#                                end_valid = if_else(check_end-1!=end_valid & !is.na(check
#                                                    check_end-1,
#                                                    end_valid)) %>%
#                         select(-check_end)
#
# # compustat_vassalou %>% filter(gvkey=='008750', fyear>1991)
# # # A tibble: 12 × 13
# # gvkey  conm                   fyear   fyr start_valid end_valid  datadate       D DP_v
# # <chr>  <chr>                  <int> <int> <date>      <date>     <date>     <dbl>
# # 1 008750 PRIME HOSPITALITY CORP  1992     6 1992-10-01  1993-09-30 1992-06-30  783.
# # 2 008750 PRIME HOSPITALITY CORP  1993    12 1994-04-01  1995-03-31 1993-12-31  239.
# # 3 008750 PRIME HOSPITALITY CORP  1994    12 1995-04-01  1996-03-31 1994-12-31  231.
```

```
# # Estimating Equity historical volatility
# vassalou_data_dtd <- merged_crsp_compustat_vassalou %>%
#     left_join(compustat_vassalou %>% select(-fyear, -fyr),
#               by = join_by(gvkey,
#                            between(date, start_valid, end_valid))) %>%
#     mutate(E = shrout * prc / 1000,
#            year = lubridate::year(date)) %>%
#     group_by(gvkey, year) %>%
#     arrange(date) %>%
#     mutate(log_ret = log(prc/lag(prc)),
#            n = n(),
#            sig_E_l = sd(log_ret, na.rm = TRUE)*sqrt(252), # assuming 252 business days i
#             sig_E = sd(ret, na.rm = TRUE)*sqrt(252)) %>% # assuming 252 business days in
#     filter(n >= 60) %>% # at least 60 days to estimate volatility and DtD
#     ungroup() %>%
#     select(permno, gvkey, date, start_valid, end_valid, D, DP_vassalou, DP_duffie, DP_km


# vassalou_data_dtd <- vassalou_data_dtd %>%
#   left_join(treasury_1Y %>%
#                 mutate(r = VALUE / 100) %>%
#                 select(-VALUE), by= c("date"="DATE")) %>%
#   fill(r, .direction = "downup")
```

```
# # Looking at IMB, fyear 1993
# data_IBM_1993 <- vassalou_data_dtd %>% filter(permno==12490, lubridate::year(date)==1993

# (res_1993 <- DtD_estimation(data_IBM_1993, DP = "DP_duffie"))
#
# # # A tibble: 1 × 10
# #   sig_V  mu_V      V DP_eoy  E_eoy   DtD DtD_kmv          PD nb_iter       error
# #   <dbl> <dbl>  <dbl>  <dbl>  <dbl> <dbl>   <dbl>       <dbl>   <int>       <dbl>
# # 1 0.131 0.105 74305. 43164. 32681.  4.89    4.15 0.000000503       2 -0.00000145

# (res_1993 <- DtD_estimation(data_IBM_1993, DP = "DP_vassalou"))

# # # A tibble: 1 × 10
# #  sig_V  mu_V      V DP_eoy  E_eoy   DtD DtD_kmv           PD nb_iter        error
# #  <dbl> <dbl>  <dbl>  <dbl>  <dbl> <dbl>   <dbl>        <dbl>   <int>        <dbl>
# # 1 0.180 0.143 54758. 22894. 32681.  5.56    4.85 0.0000000136       2 -0.000000158
```

```
# # Checking implementation with DtD package for DP defined as in @duffie2007
# n_IBM_1993 <- nrow(data_IBM_1993)
#
# # Estimating \mu_V and \sigmna_V on past year Equity data
# (ibm_est <- DtD::BS_fit(S = data_IBM_1993$E,
#                         D = data_IBM_1993$DP_duffie,
#                         T. = 1,
#                         dt = 1 / n_IBM_1993,
#                         r = data_IBM_1993$r,
#                         vol = data_IBM_1993$sig_E[n_IBM_1993] * data_IBM_1993$E[n_IBM_19
#                              (data_IBM_1993$E[n_IBM_1993] + data_IBM_1993$DP_duffie[n_IB
#                         method = "iterative"))$ests
```

```
#
# # Getting Asset Value from End of Year 2011 equity / Accounting data
# (VA <- DtD::get_underlying(S = data_IBM_1993$E[n_IBM_1993],
#                      D = data_IBM_1993$DP_duffie[n_IBM_1993],
#                      T. = 1,
#                      r = data_IBM_1993$r[n_IBM_1993],
#                      vol = ibm_est$ests['vol']))
#
# # Computing Distance to Default at the End of Year 2011
# (DD <- (log(VA/data_IBM_1993$DP_duffie[n_IBM_1993]) + (ibm_est$ests[['mu']]-0.5*ibm_est$
```

```
# # Checking implementation with DtD package for DP defined as in @vassalou2004
# n_IBM_1993 <- nrow(data_IBM_1993)
#
# # Estimating \mu_V and \sigmna_V on past year Equity data
# (ibm_est <- DtD::BS_fit(S = data_IBM_1993$E,
#                          D = data_IBM_1993$DP_vassalou,
#                          T. = 1,
#                          dt = 1 / n_IBM_1993,
#                          r = data_IBM_1993$r,
#                          vol = data_IBM_1993$sig_E[n_IBM_1993] * data_IBM_1993$E[n_IBM_19
#                               (data_IBM_1993$E[n_IBM_1993] + data_IBM_1993$DP_vassalou[n_
#                          method = "iterative"))
#
# # Getting Asset Value from End of Year 1993 equity / Accounting data
# (VA <- DtD::get_underlying(S = data_IBM_1993$E[n_IBM_1993],
#                      D = data_IBM_1993$DP_vassalou[n_IBM_1993],
#                      T. = 1,
#                      r = data_IBM_1993$r[n_IBM_1993],
#                      vol = ibm_est$ests['vol']))
#
# # Computing Distance to Default at the End of Year 1993
# (DD <- (log(VA/data_IBM_1993$DP_duffie[n_IBM_1993]) + (ibm_est$ests[['mu']]-0.5*ibm_est$
```

```
# data_IBM_1994 <- vassalou_data_dtd %>% filter(permno==12490, lubridate::year(date)==1994
#
# (res_1994 <- DtD_estimation(data_IBM_1994, DP = "DP_vassalou"))
# # # A tibble: 1 × 10
# #   sig_V  mu_V      V DP_eoy  E_eoy   DtD DtD_kmv       PD nb_iter         error
# #   <dbl> <dbl>  <dbl>  <dbl>  <dbl> <dbl>   <dbl>    <dbl>   <int>         <dbl>
# # 1 0.206 0.126 61507. 19720. 43157.  6.04    5.53 7.90e-10       2 0.00000000662
# data_IBM_1995 <- vassalou_data_dtd %>% filter(permno==12490, lubridate::year(date)==1995
#
# (res_1995 <- DtD_estimation(data_IBM_1995, DP = "DP_vassalou"))
# # # A tibble: 1 × 10
# #   sig_V   mu_V      V DP_eoy  E_eoy   DtD DtD_kmv       PD nb_iter     error
# #   <dbl>  <dbl>  <dbl>  <dbl>  <dbl> <dbl>   <dbl>    <dbl>   <int>     <dbl>
# # 1 0.188 0.0868 66060.  15844 51016.  7.95    7.59 9.01e-16       2 -3.08e-13
```

```
# data_IBM_1996 <- vassalou_data_dtd %>% filter(permno==12490, lubridate::year(date)==1996
#
# (res_1996 <- DtD_estimation(data_IBM_1996, DP = "DP_vassalou"))
# # # A tibble: 1 × 10
# #   sig_V   mu_V      V DP_eoy  E_eoy   DtD DtD_kmv       PD nb_iter     error
# #   <dbl>  <dbl>  <dbl>  <dbl>  <dbl> <dbl>   <dbl>    <dbl>   <int>     <dbl>
# # 1 0.274 0.396 94117.  16599 78408.  7.63    6.33 1.16e-14       2 0.00000000203
```

```
# data_IBM_1997 <- vassalou_data_dtd %>% filter(permno==12490, lubridate::year(date)==1997
#
# (res_1997 <- DtD_estimation(data_IBM_1997, DP = "DP_vassalou"))
# # # A tibble: 1 × 10
# #   sig_V  mu_V      V DP_eoy  E_eoy   DtD DtD_kmv       PD nb_iter         error
# #   <dbl> <dbl>  <dbl>  <dbl>  <dbl> <dbl>   <dbl>    <dbl>   <int>         <dbl>
# #   1 0.299 0.267 118647.  17893 101713.  7.08    6.33 7.31e-13       2 5.92e-11
```

```
# # We get figures for Equity volatilities
# (fy_sig_E <- vassalou_data_dtd %>%
#   filter(permno==12490) %>%
#   group_by(year=lubridate::year(date)) %>%
#   filter(year %in% c(1993,1994,1995,1996,1997)) %>%
#   summarise(sig_E = sig_E[which.max(date)]))
# # # # A tibble: 5 × 2
# # #    year sig_E
# # #   <dbl> <dbl>
# # # 1  1993 0.300
# # # 2  1994 0.301
# # # 3  1995 0.233
# # # 4  1996 0.340
# # # 5  1997 0.354
# # We put aside Equity / Asset volatilities (theoretically lower through sigmaE = V / E D
# (bind_cols(fy_sig_E,
#           bind_rows(res_1993, res_1994, res_1995, res_1996, res_1997)) %>%
#   select(-nb_iter, -error))
# ## # A tibble: 5 × 12
# ##    year sig_E sig_V   mu_V       V DP_eoy   E_eoy   DtD DtD_kmv       PD nb_iter
# ##   <dbl> <dbl> <dbl>  <dbl>   <dbl>  <dbl>   <dbl> <dbl>   <dbl>    <dbl>   <int>
# ## 1  1993 0.300 0.180 0.143   54758. 22894.  32681.  5.56    4.85 1.36e- 8       2 -1.5
# ## 2  1994 0.301 0.206 0.126   61507. 19720.  43157.  6.04    5.53 7.90e-10       2  6.6
# ## 3  1995 0.233 0.188 0.0868  66060. 15844   51016.  7.95    7.59 9.01e-16       2 -3.0
# ## 4  1996 0.340 0.274 0.396   94117. 16599   78408.  7.63    6.33 1.16e-14       2  2.0
# ## 5  1997 0.354 0.299 0.267  118647. 17893  101713.  7.08    6.33 7.31e-13       2  5.9
```

```
# # Our figures do not match Vassalou that show Sigma_V constantly higher than our figures
# # and also Sigma_E estimates which is strange
# vassalou %>% filter(Permno == 12490, Year %in% c(1993, 1994, 1995, 1996, 1997), Month==1
## # A tibble: 5 × 7
##    Year Month Permno `Default Probability` `Log Value of Assets` `Volatility of Assets`
##   <dbl> <dbl>  <dbl>                 <dbl>                 <dbl>                  <dbl>
## 1  1993    12  12490              9.94e-12                 0.661                  0.226
## 2  1994    12  12490              1.69e-10                 0.881                  0.288
## 3  1995    12  12490              8.90e-12                 1.03                   0.276
## 4  1996    12  12490              4.07e- 7                 1.37                   0.535
## 5  1997    12  12490              1.44e-11                 1.60                   0.430
```

```
# # We fit here the full Vassalou scope using DtD package, takes time to run on laptop
DtD_estimation_pkg_basic <- function(daily_market_accounting_data, DP_var ="DP_duffie", ve
# takes a vector of daily equity prices  and accounting data for a firm year.
# e.g. data <- vassalou_data_dtd %>% filter(permno==12490, lubridate::year(date)==2011)

# Re-implement using DtD package

# Checking enough non NA accounting variable (>60 days)
DP_var <- as.name(DP_var)
n_test <- nrow(daily_market_accounting_data %>% filter(!is.na(!!DP_var)))
if(n_test > 60){
  # if yes, forward/fill + filter NA
  daily_market_accounting_data <- daily_market_accounting_data %>%
    fill(!!DP_var, .direction = "down") %>%
    filter(!is.na(!!DP_var))
} else {
  # if no return empty result
  if(verbose){print(glue::glue(
    "failed to fit gvkey:{daily_market_accounting_data[1,] %>% pull(gvkey)}, no data"))}

  return(tibble(sig_V = NA_real_,
              mu_V = NA_real_,
              V = NA_real_,
              DP_eoy = NA_real_,
              E_eoy = NA_real_,
              DtD = NA_real_,
              DtD_kmv = NA_real_,
              PD = NA_real_,
              nb_iter = NA_real_,
              error = NA_real_))

}

data <- daily_market_accounting_data
n_r <- nrow(data)

eoy_data <- data %>% filter(date==max(date))
E_eoy <- eoy_data %>% pull(E)
DP_eoy <- eoy_data %>% pull(!!DP_var)
sig_E_eoy <- eoy_data %>% pull(sig_E)
r_eoy <- eoy_data %>% pull(r)

ibm_est <- DtD::BS_fit(S = data$E,
                        D = data %>% pull(!!DP_var),
                        T. = 1,
                        dt = 1 / n_r, # modify this to handle uneven time series
                        # that sometimes occur in CRSP data
                        r = data$r,
                        vol = sig_E_eoy * E_eoy / (E_eoy + DP_eoy),
                       method = "iterative")

VA <- DtD::get_underlying(S = E_eoy,
                           D = DP_eoy,
                           T. = 1,
                           r = r_eoy,
                           vol = ibm_est$ests['vol'])
```

```r
DtD <- (log(VA / DP_eoy) + (ibm_est$ests[['mu']]- 0.5 * ibm_est$ests[['vol']]^2)) / ibm_es
DtD_kmv <- log(VA / DP_eoy) / ibm_est$ests[['vol']]

return(tibble(sig_V = ibm_est$ests[['vol']],
            mu_V = ibm_est$ests[['mu']],
            V = VA,
            DP_eoy = DP_eoy,
            E_eoy = E_eoy,
            DtD = DtD,
            DtD_kmv = DtD_kmv,
            PD = pnorm(-DtD),
            nb_iter = ibm_est$n_iter,
            error = NA_real_))
}
```

```r
poss_DtD_vassalou = purrr::possibly(.f = ~ DtD_estimation_pkg_basic(.x, DP = "DP_vassalou"
                          otherwise = tibble(sig_V = NA_real_,
                                             mu_V = NA_real_,
                                             V = NA_real_,
                                             DtD = NA_real_,
                                             DtD_kmv = NA_real_,
                                             PD = NA_real_,
                                             nb_iter = NA_real_,
                                             error = NA_real_))
```

```r
# t <- Sys.time()
# all_vassalou_pkg <- vassalou_data_dtd %>%
#   group_by(permno, year = lubridate::year(date)) %>%
#   tidyr::nest() %>%
#   mutate(dtd = purrr::map(data, poss_DtD_vassalou))
# print(Sys.time()-t)
# saveRDS(all_vassalou_pkg, "./wrds_data/all_vassalou_pkg.rds")
```

```r
#| code-fold: show
all_vassalou_pkg <- readRDS("./wrds_data/all_vassalou_pkg.rds")
all_vassalou_pkg <- all_vassalou_pkg %>%
                        unnest(dtd) %>%
                        ungroup() %>%
                        mutate(n_obs = map_dbl(data, ~ nrow(.x)))
```

Using Vassalou's definition of Default Point, we find *DtD* summary statistic (mean, sd), comparable with other authors:

```
all_vassalou_pkg %>%
 select(-error) %>%
  mutate(DtD = if_else(DtD < qnorm(1e-15),
         qnorm(1e-15),
         if_else(DtD > qnorm(1-1e-15),
                 qnorm(1-1e-15),
                 DtD))) %>%
 summarize(m = mean(DtD, na.rm = TRUE), s = sd(DtD, na.rm = TRUE))
```

```
# A tibble: 1 x 2
      m     s
  <dbl> <dbl>
1  5.01  2.78
```

```
#  m          s
# 5.014422  2.777746
#The probabilities of default in Vassalou and Xing (2004) are available at www.maria-vassa
# Comparing our distance to default to theirs over the same period after truncating at a 1
# yields a mean and standard deviation of the distance to default of 4.856 and 2.739, resp
# The corresponding figures in Vassalou and Xing (2004) are 4.391 and 2.608, respectively.
```

```
all_vassalou_pkg %>%
 select(-error) %>%
  mutate(DtD_kmv = if_else(DtD_kmv < qnorm(1e-15),
         qnorm(1e-15),
         if_else(DtD_kmv > qnorm(1-1e-15),
                 qnorm(1-1e-15),
                 DtD_kmv))) %>%
 summarize(m = mean(DtD_kmv, na.rm = TRUE), s = sd(DtD_kmv, na.rm = TRUE))
```

```
# A tibble: 1 x 2
      m     s
  <dbl> <dbl>
1  4.99  2.59
```

```
#  m          s
# 4.991748  2.585159
```

```
vassalou %>%
  mutate(`Default Probability` = if_else(`Default Probability` < 1e-15,
                                         1e-15,
                                         if_else(`Default Probability` > 1 - 1e-15,
                                                 1 - 1e-15,
                                                 `Default Probability`))) %>%
  mutate(`Distance to Default`=-qnorm(`Default Probability`)) %>%
  summarise(m = mean(`Distance to Default`, na.rm=TRUE),
            s = sd(`Distance to Default`, na.rm=TRUE))
```

```
# A tibble: 1 x 2
      m     s
  <dbl> <dbl>
1  4.37  2.57
```

```
# # A tibble: 1 × 2
#       m     s
#   <dbl> <dbl>
# 1  4.37  2.57
```

3.3.3 All Compustat Data

We then perform the iterative *DtD* estimation on the whole Compustat/CRSP universe. Again (to render quickly the quarto document) we comment the code which is memory and computational intensive.

Filtering CRSP data:

```
# crsp_daily <- crsp_daily %>%
#   filter(!is.na(prc), !is.na(shrout)) %>%
#   select(permno, date, prc, shrout, ret)
```

Using whole Compustat perimeter to link CRSP:

```
# compustat_perimeter <- ccmxpf_linktable %>%
#   filter(gvkey %in% unique(compustat_all$gvkey)) %>%
#   left_join(company_all %>% select(gvkey, conm), by = c("gvkey"))
```

We then merge our daily CRSP data set with the preceding mapping:

```
# merged_crsp_compustat <- crsp_daily %>%
#   left_join(compustat_perimeter,
#             by = join_by(permno,
#                          between(date, linkdt, linkenddt))) %>%
#   filter(!is.na(gvkey))
```

We prepare Compustat data to be used in conjunction with CRSP, creating in particular Default Points from Debt estimates:

```
# compustat_dtd <- compustat_all %>%
#   mutate(act = coalesce(act, che + rect + invt + aco),
#          at = coalesce(at, act + ppent + ivaeq + ivao + intan + ao),
#          ap = coalesce(ap, 0),
#          txp = coalesce(txp, 0),
#          lco = coalesce(lco, 0),
#          lct = coalesce(lct, ap + txp + dlc + lco),
#          wcap = coalesce(wcap, act - lct),
#          lt = coalesce(lt, lct + dltt + txditc + lo),
#          DP_vassalou = coalesce(dlc, lct) + 0.5 * pmin(dltt, lt - lct, na.rm=T),
#          # Vassalou (2004) p837 As book value of debt we use the "Debt in One Year" [dlc
#          # plus half the "Long- Term Debt." [dltt].
#          # careful dlc /dltt = 0 sometimes leading Inf DtD
#          # (eg permno 76671 gvkey 021828 fyear 2005)
#          DP_duffie = pmax(dlc, lct, na.rm=TRUE) + 0.5 * pmin(dltt, lt - lct, na.rm=T),
#          # Duffie (2007) Short-term debt is estimated as the larger of Compustat items D
#          # and DATA49 [lct].
#          # Long-term debt is taken from DATA51 [dltt].
#          DP_vassalou = if_else(DP_vassalou<1e-6,
#                                if_else(DP_duffie<1e-6, 0.5, DP_duffie), # putting a 0.5M
#                                DP_vassalou),
#          # See Crosbie (2003)
#          # Philip Morris / Altria eoy 2000
#          # LT = 64Bn / Default Point = 47.5 = LCT + 0.5*(LT-LCT)
#          DP_kmv = pmax(dlc, lct, na.rm=TRUE) + 0.5 * (lt - lct),
#          start_valid = datadate %m+% months(3),
#          end_valid = start_valid %m+% years(1),
#          start_valid = start_valid + 1) %>% # avoid overlapping periods
#   select(gvkey, conm, fyear, fyr, start_valid, end_valid, datadate, D=lt, DP_vassalou, D
# #
# # # Extending accounting data validity dates defined before to avoid data gap,
# # when merging CRSP and Compustat
# compustat_dtd <- compustat_dtd %>%
#     mutate(check_end = lead(start_valid),
#            end_valid = if_else(check_end-1!=end_valid & !is.na(check_end),
#                                check_end-1,
#                                end_valid)) %>%
#     select(-check_end)
#
# saveRDS(compustat_dtd, "./wrds_data/compustat_dtd.rds")
```

```
compustat_dtd <- readRDS("./wrds_data/compustat_dtd.rds")
```

Loading risk-free rate data:

```
# Vassalou (2004) / Lu (2008) risk free rate

treasury_1Y <- readRDS("./wrds_data/treasury_1Y.rds")
# # The data has been previously collected this way:
# treasury_1Y_url <- 'https://fred.stlouisfed.org/series/DGS1/downloaddata/DGS1.csv'
# treasury_1Y <- readr::read_csv(treasury_1Y_url, col_types = cols(DATE = "D", VALUE = "d"
#   filter(!is.na(VALUE))
#
# saveRDS(treasury_1Y, "./wrds_data/treasury_1Y.rds")
```

We then estimate Equity historical volatility, it occurs that CRSP provides irregular spacing for share prices observations (usually providing less than 252 observations per year, yielding non daily returns), so that we adjust log returns, we count business days for each increment dt and adjust returns with $\sqrt{dt}$ if necessary:

```
# library(bizdays)
# load_rmetrics_calendars(1960:2030)
```

We split in multiple batches to avoid memory issues on laptops:

```
# t <- Sys.time()
#
# permnos <- merged_crsp_compustat %>%
#    distinct(permno) %>%
#    pull(permno)
#
# batch_size <- 5000
# batches <- ceiling(length(permnos) / batch_size)
```

```
#
# for (j in 1:batches) {
#   print(j)
#   permno_batch <- permnos[
#   ((j - 1) * batch_size + 1):min(j * batch_size, length(permnos))
#   ]
#   compustat_data_dtd_batch <- merged_crsp_compustat %>%
#       filter(permno %in% permno_batch) %>%
#       left_join(compustat_dtd %>% select(-fyear, -fyr),
#                 by = join_by(gvkey,
#                              between(date, start_valid, end_valid))) %>%
#       mutate(E = shrout * prc / 1000,
#              year = lubridate::year(date)) %>%
#       group_by(gvkey) %>%
#       arrange(date) %>%
#       mutate(log_ret = log(prc/lag(prc)),
#              lagDt = lag(date)) %>%
#       filter(!is.na(lagDt)) %>%
#       ungroup() %>%
#       group_by(gvkey, year) %>%
#       mutate(n = n(),
#              log_ret_min = quantile(log_ret, probs=.005, na.rm = TRUE), # winsorizing lo
#              log_ret_max = quantile(log_ret, probs=.995, na.rm = TRUE), # cf (gvkey 0278
#              # dt = bizdays(lagDt, date, "Rmetrics/NYSE"),
#              # put a floor at 1 business day to deal with mkt holidays (not flagged by C
#              # cf (permno 12490, year 1980, 30/31 December)
#              dt = pmax(1, bizdays(lagDt, date, "Rmetrics/NYSE")),
#              time = cumsum(dt/252),
#              sq_dt = sqrt(dt)) %>%
#      filter(log_ret>log_ret_min, log_ret<log_ret_max) %>%
#      mutate(
#              # sq_dt = if_else(n>1,sqrt(bizdays(lagDt, date, "Rmetrics/NYSE")), 1),
#              sig_E_l = sd(log_ret, na.rm = TRUE)*sqrt(252), # assuming 252 business days
#              # /!\ not all CRSP data is daily, sometimes uneven, overestimates volatilit
#              # https://quant.stackexchange.com/questions/42407/estimating-daily-volatili
#              sig_E_l_adj = sd(log_ret/sq_dt, na.rm = TRUE)*sqrt(252), # correcting for u
#              sig_E = sd(ret, na.rm = TRUE)*sqrt(252)) %>% # assuming 252 business days i
#       filter(n >= 60) %>% # at least 60 days to estimate volatility and DtD
#       ungroup() %>%
#       select(permno, gvkey, date, start_valid, end_valid, D, DP_vassalou, DP_duffie, DP_
#
```

```
#   compustat_data_dtd_batch <- compustat_data_dtd_batch %>%
#     left_join(treasury_1Y %>%
#                     mutate(r = VALUE / 100) %>%
#                     select(-VALUE), by= c("date"="DATE")) %>%
#     fill(r, .direction = "downup")
#
#   saveRDS(compustat_data_dtd_batch, glue::glue("./wrds_data/compustat_data_dtd_batch_{j}
#   print(Sys.time()-t)
#
# }
# print(Sys.time()-t)
# #
# # [1] 1
# # Time difference of 4.193479 mins
# # [1] 2
# # Time difference of 11.57161 mins
# # [1] 3
# # Time difference of 17.27337 mins
# # [1] 4
# # Time difference of 21.23308 mins
# # [1] 5
# # Time difference of 23.42284 mins
# # Time difference of 23.42425 mins
#
# compustat_data_dtd_full <- list.files(path = "./wrds_data",
#                                       full.names = TRUE,
#                                       pattern = "compustat_data_dtd_batch_[0-9]{1,2}.rds"
#                            map_dfr(readRDS)
# saveRDS(compustat_data_dtd_full, "./wrds_data/compustat_data_dtd_full.rds")
```

```
compustat_data_dtd_full <- readRDS("./wrds_data/compustat_data_dtd_full.rds")
```

We also have to modify the *DtD* estimation function to account for unequal spacing:

```
DtD_estimation_pkg <- function(daily_market_accounting_data, DP_var ="DP_duffie", verbose=
# takes a vector of daily equity prices  and accounting data for a firm year.
# e.g. data <- vassalou_data_dtd %>% filter(permno==12490, lubridate::year(date)==2011)

# Re-implement using DtD package

# Checking enough non NA accounting variable (>60 days)
```

```r
DP_var <- as.name(DP_var)
n_test <- nrow(daily_market_accounting_data %>% filter(!is.na(!!DP_var)))
if(n_test > 60){
  # if yes, forward/fill default point with last available value + filter NA
  daily_market_accounting_data <- daily_market_accounting_data %>%
    fill(!!DP_var, .direction = "down") %>%
    filter(!is.na(!!DP_var))
} else {
  # if no return empty result
  if(verbose){print(glue::glue(
    "failed to fit gvkey:{daily_market_accounting_data[1,] %>% pull(gvkey)}, no data"))}

  return(tibble(sig_V = NA_real_,
                mu_V = NA_real_,
                V = NA_real_,
                DP_eoy = NA_real_,
                E_eoy = NA_real_,
                DtD = NA_real_,
                DtD_kmv = NA_real_,
                PD = NA_real_,
                nb_iter = NA_real_,
                error = NA_real_))

}

data <- daily_market_accounting_data
n_r <- nrow(data)

eoy_data <- data %>% filter(date==max(date))
E_eoy <- eoy_data %>% pull(E)
DP_eoy <- eoy_data %>% pull(!!DP_var)
sig_E_eoy <- eoy_data %>% pull(sig_E)
sig_E_l_adj_eoy <- eoy_data %>% pull(sig_E_l_adj)
r_eoy <- eoy_data %>% pull(r)

if(unequal){
  firm_est <- DtD::BS_fit(S = data$E,
                          D = data %>% pull(!!DP_var),
                          T. = 1,
                          # dt = 1 / n_r, # modify this to handle uneven time series
                          # that sometimes occur in CRSP data
                          time = data$time,
                          r = data$r,
                          vol = sig_E_l_adj_eoy * E_eoy / (E_eoy + DP_eoy),
                          method = "iterative")
} else {
  firm_est <- DtD::BS_fit(S = data$E,
                          D = data %>% pull(!!DP_var),
                          T. = 1,
                          dt = 1 / n_r, # modify this to handle uneven time series
                          # that sometimes occur in CRSP data
                          r = data$r,
                          vol = sig_E_eoy * E_eoy / (E_eoy + DP_eoy),
                          method = "iterative")
}


VA <- DtD::get_underlying(S = E_eoy,
                           D = DP_eoy,
                           T. = 1,
                           r = r_eoy,
                           vol = firm_est$ests['vol'])

DtD <- (log(VA / DP_eoy) + (firm_est$ests[['mu']]- 0.5 * firm_est$ests[['vol']]^2)) / firm
DtD_kmv <- log(VA / DP_eoy) / firm_est$ests[['vol']]

return(tibble(sig_V = firm_est$ests[['vol']],
              mu_V = firm_est$ests[['mu']],
              V = VA,
              DP_eoy = DP_eoy,
              E_eoy = E_eoy,
              DtD = DtD,
              DtD_kmv = DtD_kmv,
              PD = pnorm(-DtD),
              nb_iter = firm_est$n_iter,
              error = NA_real_))
}
```

```r
poss_DtD = purrr::possibly(.f = ~ DtD_estimation_pkg(.x, DP = "DP_duffie", unequal=TRUE),
                          otherwise = tibble(sig_V = NA_real_,
                                             mu_V = NA_real_,
                                             V = NA_real_,
                                             DP_eoy = NA_real_,
                                             E_eoy = NA_real_,
                                             DtD = NA_real_,
                                             DtD_kmv = NA_real_,
                                             PD = NA_real_,
                                             nb_iter = NA_real_,
                                             error = NA_real_))
```

Again, we split the data set in multiple batches to avoid memory issues on laptops when fitting *DtD*:

```
# t <- Sys.time()
#
# permnos <- merged_crsp_compustat %>%
#    distinct(permno) %>%
#    pull(permno)
#
# batch_size <- 5000
# batches <- ceiling(length(permnos) / batch_size)
#
# for (j in 1:batches) {
#   print(j)
#   permno_batch <- permnos[
#   ((j - 1) * batch_size + 1):min(j * batch_size, length(permnos))
#   ]
#
#   all_compustat_pkg_batch <- compustat_data_dtd_full %>%
#   filter(permno %in% permno_batch) %>%
#   group_by(permno, year = lubridate::year(date)) %>%
#   tidyr::nest() %>%
#   mutate(dtd = purrr::map(data, poss_DtD))
#
#   saveRDS(all_compustat_pkg_batch, glue::glue("./wrds_data/all_compustat_pkg_full_{j}.rd
#   print(Sys.time()-t)
# # [1] 1
# # Time difference of 7.236444 mins
# # [1] 2
# # Time difference of 18.27073 mins
# # [1] 3
# # Time difference of 27.13912 mins
# # [1] 4
# # Time difference of 33.65294 mins
# # [1] 5
# # Time difference of 37.17925 mins
# }
#
# all_compustat_pkg_full <- list.files(path = "./wrds_data",
#                                      full.names = TRUE,
#                                      pattern = "all_compustat_pkg_full_[0-9]{1,2}.rds")
#                            map_dfr(readRDS)
# saveRDS(all_compustat_pkg_full, "./wrds_data/all_compustat_pkg_full.rds")
```

```
all_compustat_pkg_full <- readRDS("./wrds_data/all_compustat_pkg_full.rds")
all_compustat_pkg_full <- all_compustat_pkg_full %>%
                          unnest(dtd) %>%
                          ungroup() %>%
                          mutate(n_obs = map_dbl(data, ~ nrow(.x))) %>%
                          select(-error)
```

```
# # Saving a compact, cleansed version of DtD dataset
# dtd_cleansed <- all_compustat_pkg_full %>%
#   filter(!is.na(DtD)) %>%
#   mutate(gvkey = map_chr(data, ~ min(.x$gvkey))) %>%
#   select(permno, gvkey, year, sig_V, mu_V, DP_eoy, DtD, DtD_kmv)
# saveRDS(dtd_cleansed, "./wrds_data/dtd_cleansed.rds")
```

```
all_compustat_pkg_full %>%
 filter(DtD!=Inf, PD > 1e-15, PD < 1- 1e-15) %>%
 # filter(DtD!=Inf) %>%
 summarize(m = mean(DtD, na.rm = TRUE), s = sd(DtD, na.rm = TRUE), med= median(DtD, na.rm
```

```
# A tibble: 1 x 3
      m     s   med
  <dbl> <dbl> <dbl>
1  3.20  2.44  3.14
```

```
# # A tibble: 1 × 3
#       m     s   med
#   <dbl> <dbl> <dbl>
# 1  3.20  2.44  3.14
```

Comparing the impact of changing volatility estimation and winsorizing daily returns for `IBM` and fiscal year 2011

```
vol_equity_estimation %>%
  tail(1) %>%
  select(gvkey, start_valid, end_valid, DP_vassalou, DP_duffie, E, sig_E_before, sig_E, si
```

```
# A tibble: 1 x 11
  gvkey  start_valid end_valid  DP_vassalou DP_duffie       E sig_E_before sig_E
  <chr>  <date>      <date>           <dbl>     <dbl>   <dbl>        <dbl> <dbl>
1 006066 2011-04-01  2012-03-31       17701     51485 216724.        0.228 0.209
# i 3 more variables: sig_E_l_before <dbl>, sig_E_l_adj <dbl>, r <dbl>
```

```
ibm_data_dtd_adjusted <- compustat_data_dtd_full %>%
   filter(permno == 12490, lubridate::year(date) == 2011)
```

Here assuming 252 business days, and $dt = 1$ increments:

```
DtD_estimation_pkg_basic(ibm_data_dtd %>% filter(lubridate::year(date)==2011), DP = "DP_du
```

```
# A tibble: 1 x 10
  sig_V mu_V       V DP_eoy   E_eoy   DtD DtD_kmv       PD nb_iter error
  <dbl> <dbl>   <dbl>  <dbl>   <dbl> <dbl>   <dbl>    <dbl>   <dbl> <dbl>
1 0.184 0.171 268148.  51485 216724.  9.80    8.97 5.43e-23       2    NA
```

```
# # A tibble: 1 × 10
#   sig_V  mu_V       V DP_eoy   E_eoy   DtD DtD_kmv       PD nb_iter error
#   <dbl> <dbl>   <dbl>  <dbl>   <dbl> <dbl>   <dbl>    <dbl>   <dbl> <dbl>
# 1 0.184 0.171 268148.  51485 216724.  9.80    8.97 5.43e-23       2    NA
```

Here after winsorizing/adjusting:

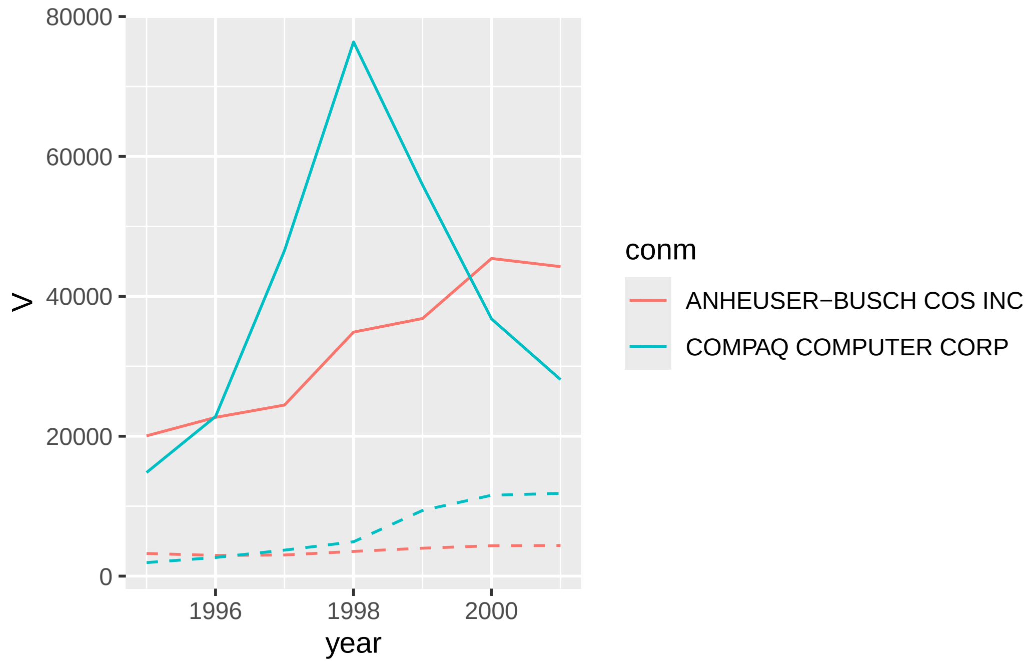
```
DtD_estimation_pkg(ibm_data_dtd_adjusted, DP = "DP_duffie", unequal=TRUE)
```

```
# A tibble: 1 x 10
  sig_V mu_V       V DP_eoy   E_eoy   DtD DtD_kmv       PD nb_iter error
  <dbl> <dbl>   <dbl>  <dbl>   <dbl> <dbl>   <dbl>    <dbl>   <dbl> <dbl>
1 0.176 0.169 268148.  51485 216724.  10.2    9.38 5.93e-25       2    NA
```

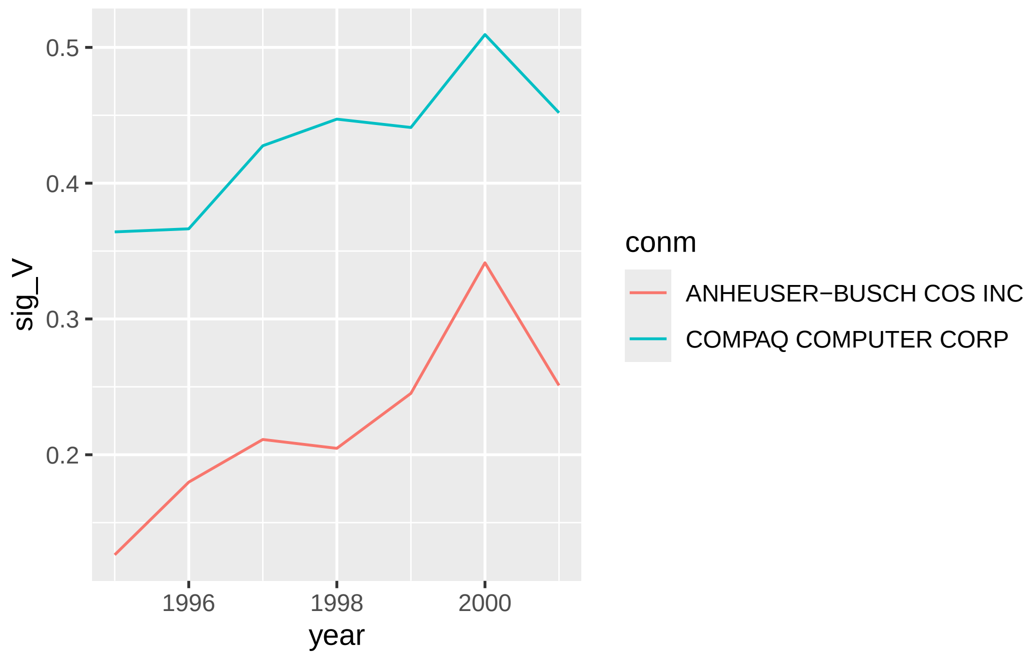
```
# A tibble: 1 × 10
# sig_V  mu_V       V DP_eoy   E_eoy   DtD DtD_kmv       PD nb_iter error
# <dbl> <dbl>   <dbl>  <dbl>   <dbl> <dbl>   <dbl>     <dbl>   <dbl> <dbl>
# 1 0.176 0.169 268148.  51485 216724.  10.2    9.38 5.93e-25       2    NA
```

We compare here two firms Compaq and Anheuser-Busch that were displayed as examples in KMV white paper Crosbie & Bohn (2003) (see p.5 comparison of AnBev vs Compaq as of April 2001):

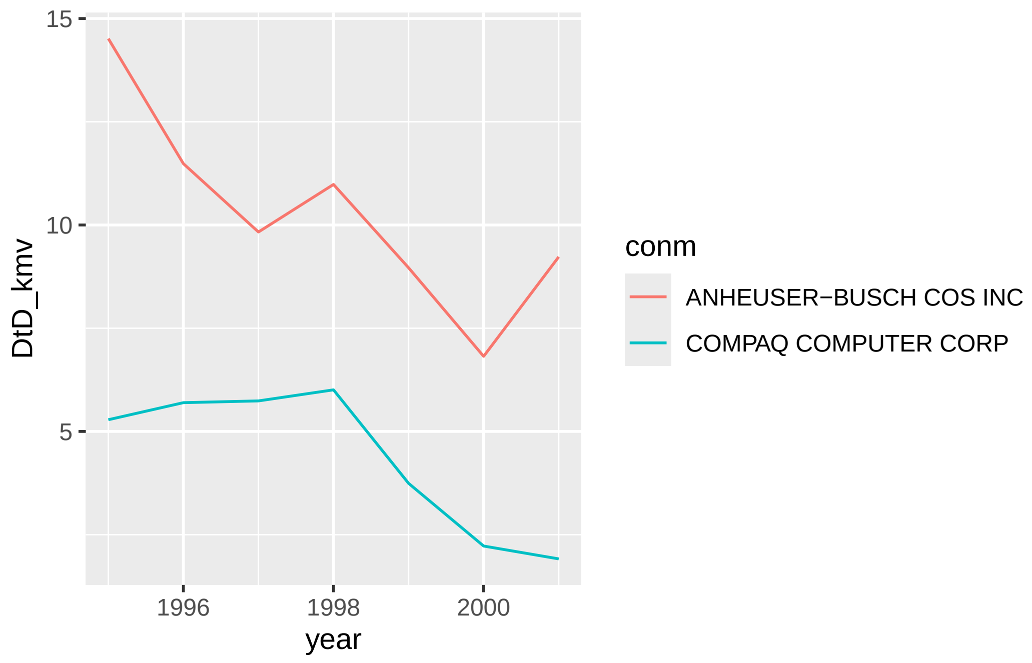
```
compaq_anbev <- all_compustat_pkg_full %>%
  filter(permno %in% c(68347,59184)) %>%
  filter(year>=1995, year<=2001) %>%
  mutate(DP = map_dbl(data, ~mean(.x$DP_duffie, na.rm=TRUE)),
         gvkey = map_chr(data, ~min(.x$gvkey, na.rm=TRUE))) %>%
  left_join(company_all %>% select(gvkey, conm), by= c("gvkey"))


# Market Value of Assets vs Default Point
(ggplot(compaq_anbev, aes(x=year, y=V, col=conm)) +
  geom_line() +
  geom_line(aes(x=year, y = DP, col =conm), linetype = 2))
```



```
# Asset volatilities
(ggplot(compaq_anbev, aes(x=year, y=sig_V, col=conm)) +
  geom_line())
```

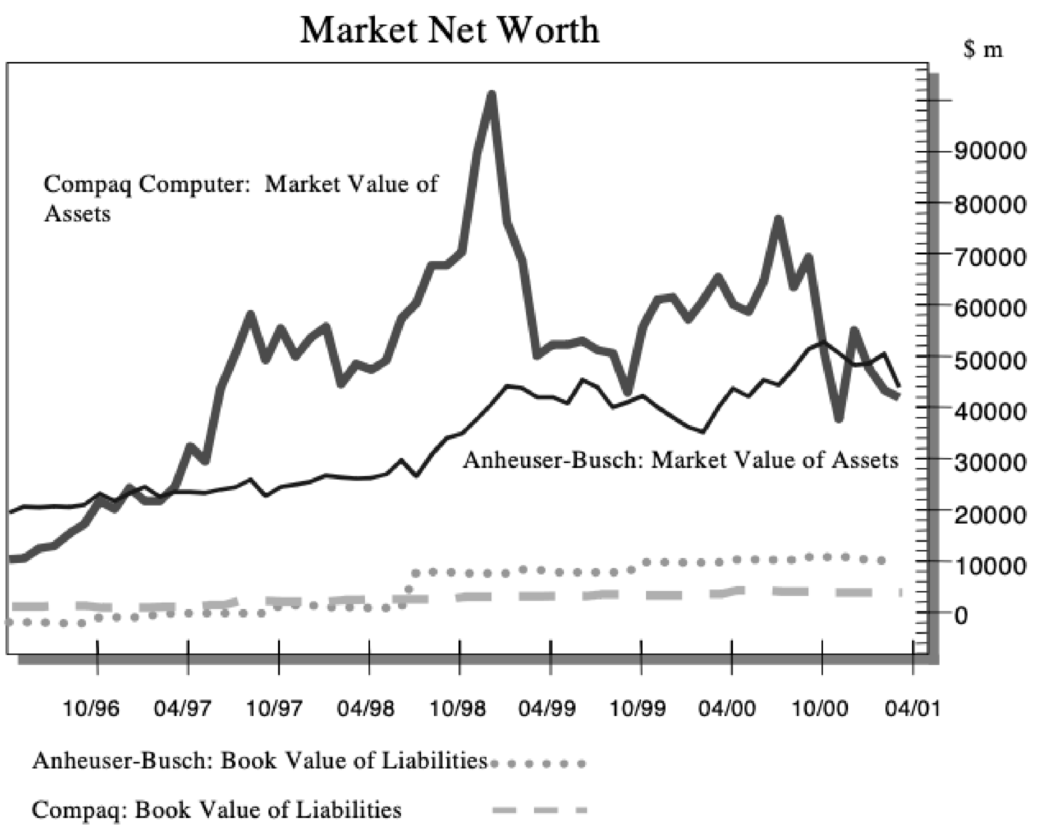


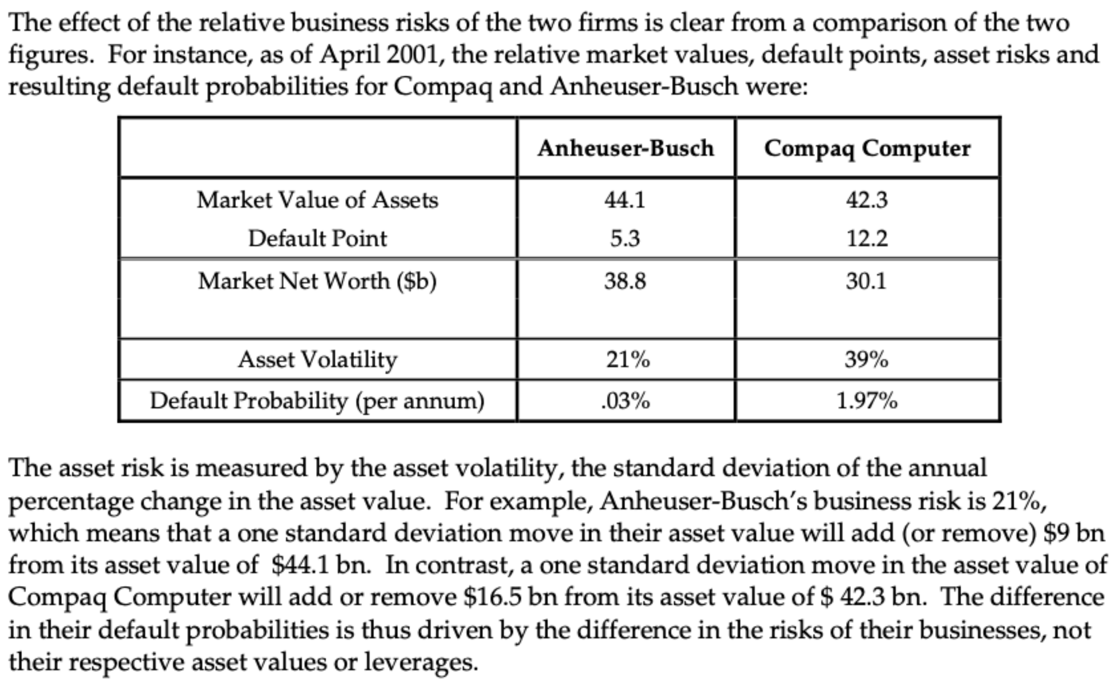
```
# Distance to default
(ggplot(compaq_anbev, aes(x=year, y=DtD_kmv, col=conm)) +
  geom_line())
```



At End of Year 2000, both companies have similar Asset Values around $\$40 Bn$. But Compaq has a higher Default Point and also shows higher asset volatility hence Compaq is materially closer to default than Anheuser-Busch.

KMV models is not directly comparable as it uses a different model as Black Scholes (perpetual down an out barrier) to model Equity and Asset Values, however we somewhat retrieve comparable dynamics for the two companies:


Market Net Worth
$ m
Compaq Computer: Market Value of Assets
Anheuser-Busch: Market Value of Assets
90000
80000
70000
60000
50000
40000
30000
20000
10000
0
10/96
04/97
10/97
04/98
10/98
04/99
10/99
04/00
10/00
04/01
Anheuser-Busch: Book Value of Liabilities
Compaq: Book Value of Liabilities



The effect of the relative business risks of the two firms is clear from a comparison of the two figures. For instance, as of April 2001, the relative market values, default points, asset risks and resulting default probabilities for Compaq and Anheuser-Busch were:

	Anheuser-Busch	**Compaq Computer**
Market Value of Assets	44.1	42.3
Default Point	5.3	12.2
Market Net Worth ($b)	38.8	30.1
Asset Volatility	21%	39%
Default Probability (per annum)	.03%	1.97%

The asset risk is measured by the asset volatility, the standard deviation of the annual percentage change in the asset value. For example, Anheuser-Busch's business risk is 21%, which means that a one standard deviation move in their asset value will add (or remove) $9 bn from its asset value of $44.1 bn. In contrast, a one standard deviation move in the asset value of Compaq Computer will add or remove $16.5 bn from its asset value of $ 42.3 bn. The difference in their default probabilities is thus driven by the difference in the risks of their businesses, not their respective asset values or leverages.

We finish comparing our estimated `IBM` asset volatilities with Vassalou (that do not publish *DtD* figures):

```
vassalou_ibm <- vassalou %>% filter(Permno==12490, Month==12) %>% select(Year, `Volatility

compustat_dtd_ibm <- all_compustat_pkg_full %>% filter(permno %in% c(12490)) %>% select(ye
```

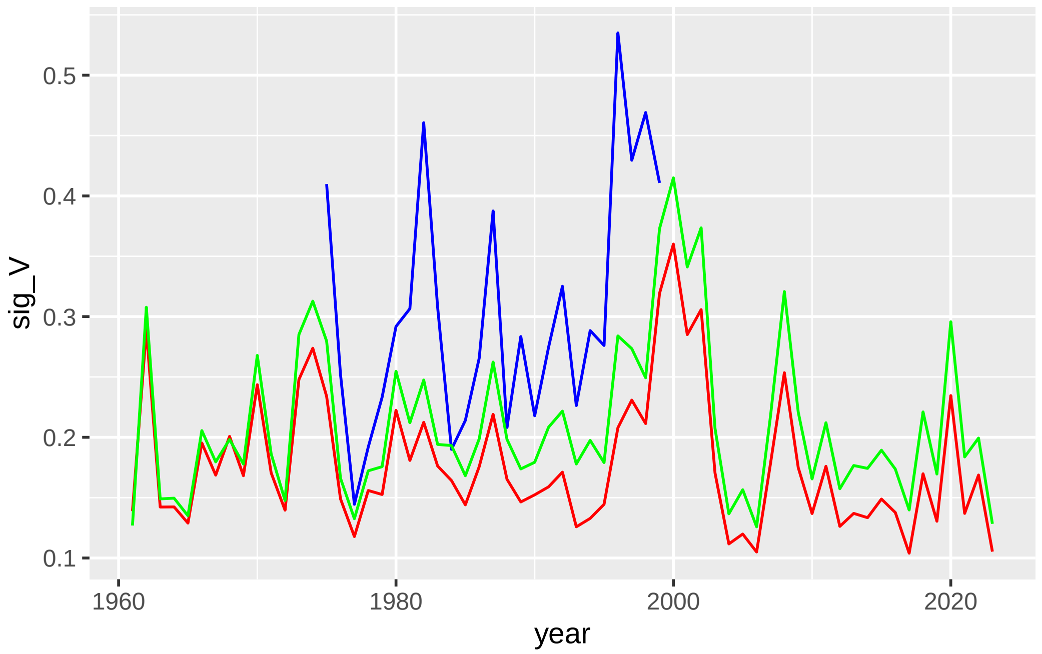
For better comparison we also re-estimate Asset volatilities, using Vassalou's definition of Default Point:

```
poss_DtD_vassalou = purrr::possibly(.f = ~ DtD_estimation_pkg(.x, DP = "DP_vassalou", uneq
                          otherwise = tibble(sig_V = NA_real_,
                                             mu_V = NA_real_,
                                             V = NA_real_,
                                             DP_eoy = NA_real_,
                                             E_eoy = NA_real_,
                                             DtD = NA_real_,
                                             DtD_kmv = NA_real_,
                                             PD = NA_real_,
                                             nb_iter = NA_real_,
                                             error = NA_real_))
```

```r
ibm_compustat_pkg_vassalou <- compustat_data_dtd_full %>%
  filter(permno==12490) %>%
  group_by(permno, year = lubridate::year(date)) %>%
  tidyr::nest() %>%
  mutate(dtd = purrr::map(data, poss_DtD_vassalou))
```

We plot in red/green our Asset volatility estimates using different Default Point definition (Duffie vs Vassalou) together with Vassalou published estimate, again the trend is similar but Vassalou estimates are higher than ours:

```r
ggplot(compustat_dtd_ibm %>% filter(!is.na(sig_V)),
       aes(x=year, y = sig_V)) +
  geom_line(color='red') +
  geom_line(data=vassalou_ibm,
            aes(x=Year, y= `Volatility of Assets` ), color='blue') +
  geom_line(data=ibm_compustat_pkg_vassalou %>%
                   unnest(dtd) %>%
                   ungroup() %>%
                   filter(!is.na(sig_V)),
            aes(x=year, y= sig_V ), color='green')
```



Altman, E. I., & Rijken, H. A. (2004). How rating agencies achieve rating stability. *Journal of*

Banking & Finance, *28*(11), 2679–2714. https://doi.org/10.1016/j.jbankfin.2004.06.006

Bharath, S. T., & Shumway, T. (2008). Forecasting default with the merton distance to default model. *The Review of Financial Studies*, *21*(3), 1339–1369. https://doi.org/10.1093/rfs/hhn044

Chava, S., & Jarrow, R. A. (2004). Bankruptcy prediction with industry effects. *Review of Finance*, *8*(4), 537–569. https://doi.org/10.1093/rof/8.4.537

Christoffersen, B. (2019). Corporate default models: Empirical evidence and methodological contributions. In *Copenhagen Business School [Phd]*.

Crosbie, P., & Bohn, J. R. (2003). Modeling default risk. *KMV LLC*.

Duan, J. -C., & Wang, T. (2012). *Measuring distance-to-default for financial and non-financial firms*.

Duffie, D., Saita, L., & Wang, K. (2007). Multi-period corporate default prediction with stochastic covariates. *Journal of Financial Economics*, *83*(3), 635–665. https://doi.org/https://doi.org/10.1016/j.jfineco.2005.10.011

Forssbæck, J., & Vilhelmsson, A. (2017). Predicting default merton vs. leland. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2914545

Moreno-Bromberg, S., & Rochet, J.-C. (2018). *Continuous-time models in corporate finance, banking, and insurance: A user's guide*. Princeton University Press. https://doi.org/10.2307/j.ctvc774r0

Shumway, T. (2001). Forecasting bankruptcy more accurately: A simple hazard model. *The Journal of Business*, *74*(1), 101–124.

Vassalou, M., & Xing, Y. (2004). Default risk in equity returns. *The Journal of Finance*, *59*(2), 831–868. https://doi.org/10.1111/j.1540-6261.2004.00650.x