Introduction to WRDS and Bankruptcy data / Practical 1 - Data set construction

Louis OLIVE

Table of contents


```
library(tidyverse)
library(scales)
# install.packages("RSQLite")
library(RSQLite)
#install.packages("dbplyr")
library(dbplyr)
# install.packages("RPostgres")
library(RPostgres)
```

1 Introduction to WRDS / Compustat / CRSP with R

WRDS (Wharton Research Data Services) is an online platform that provides access to a wide range of financial, economic, and corporate data for academic and professional research. It serves as a central hub for databases like Compustat and CRSP.

Compustat offers financial and accounting data on publicly traded companies, including balance sheets, income statements, and other fundamental metrics. It's widely used for company performance analysis and financial modeling.

CRSP (Center for Research in Security Prices) provides stock market data, such as prices, returns, and trading volumes, for U.S. companies.

A good reference about WRDS / Compustat / CRSP usage with R or Python is the online book Tidy Finance with R. In particular the chapter WRDS, CRSP, and Compustat gives solid bases and tricks to allow a fast kickstart with WRDS data.

A second very good reference also using R is Empirical Research in Accounting: Tools and Methods, especially chapter 6 / 7 / 8.

1.1 Connecting to WRDS remote Database (NO MORE VALID, KEEP FOR ARCHIVE)

We assume here that you have registered to WRDS using your ut-capitole.fr email.

First it is a good practice to store your id/password in a dotfile for R, so that you can easily connect to WRDS from your code/notebook.

Tidy Finance authors propose to create two environment variables inside .Renviron.

See below:

```
# First create two environment variables to connect wrds
# in a terminal: touch $HOME/.Renviron
# inside the .Renviron file
# wrds_user = your_user
# wrds_password = your_password
# if needed re-read the .Renviron file
# readRenviron("~/.Renviron")

# see here for more details
# https://www.tidy-finance.org/r/setting-up-your-environment.html#creating-environment-var
```

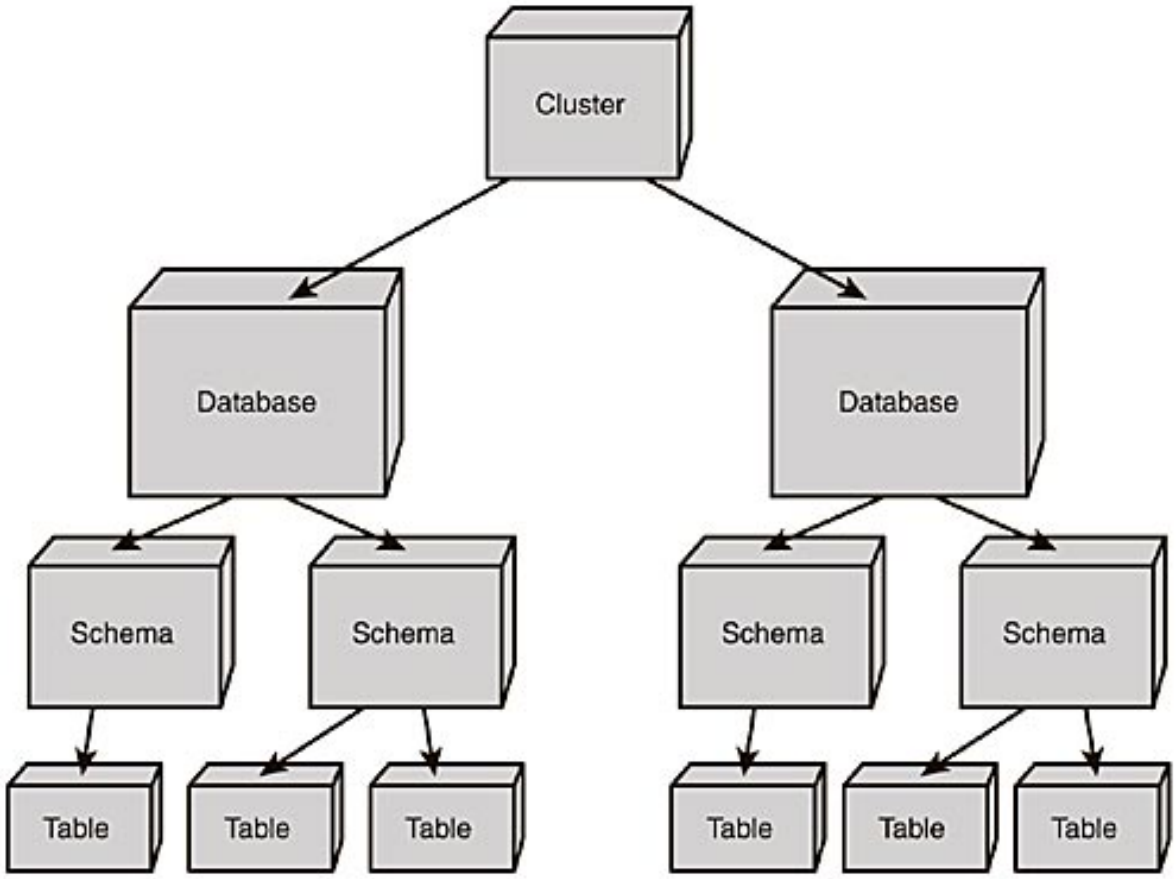
Using the environment variables defined before (wrds_user and wrds_password), we first set up a connection to WRDS remote PostgreSQL Database:

```
# wrds <- dbConnect(
#     Postgres(),
#     host = "wrds-pgdata.wharton.upenn.edu",
#     dbname = "wrds",
#     port = 9737,
#     sslmode = "require",
#     user = Sys.getenv("wrds_user"),
#     password = Sys.getenv("wrds_password")
# )
```

Alternatively, but it is a bad practice, you can hardcode you identifiers in your `R` code:

```
# Hardocded
# wrds <- dbConnect(
#   Postgres(),
#   host = "wrds-pgdata.wharton.upenn.edu",
#   dbname = "wrds",
#   port = 9737,
#   sslmode = "require",
#   user = "YOUR_WRDS_USER",
#   password = "YOUR_WRDS_PWD"
# )
```

WRDS hosts Compustat/CRSP on a remote `PostgreSQL` Database. Roughly speaking a `PostgreSQL` Database contains Schema(s) which contain data Table(s):



We give some basic usage for `PostgreSQL` with `R`, more details available here.

We will use extensively `dbplyr` which is a database back-end / interface for `dplyr`. In conjunction with `RPostgres` it allows to use remote Database Tables as if they were `R` data frames or `tibble` and automatically convert `dplyr` verbs or commands into `SQL`.

1.2 Compustat

Starting with `Compustat`, the table `funda` from schema `comp` contains annual firm-level information (Fundamentals Annual) on North American companies and correspond to data available on WRDS website here. It will be our source of financial/accounting data, an overview of `Compustat` North America data is available on WRDS website.

Using the database connection set up before (`wrds`), we request Fundamentals for APPLE INC, selecting items needed to compute Altman's Z-Score Ratios:

```r
# Use dplyr verbs with a remote database table
# https://dbplyr.tidyverse.org/reference/tbl.src_dbi.html


# WON'T WORK AFTER 31 DEC 2024
# funda_db <- tbl(wrds, I("comp.funda"))

# Replaced by local files
funda_db <- readRDS("./wrds_data/compustat_all_light.rds")

# funda_db <- tbl(wrds, in_schema("comp", "funda")) # equivalently

# Perform a SQL request on remote database "as if" we are using a local data frame / tibbl
# For example here we request data from Fundamentals Annual / funda
# We filter on Company names / conm containing 'APPLE INC'
# We select some financial items and compute Altman's ratios
(apple_altman <- funda_db %>%
  filter(grepl('^APPLE INC', conm)) %>%
  select(gvkey, fyear, conm, at, wcap, re, ebit, lt, sale) %>%
  mutate(WCTA = wcap / at,
         RETA = re / at,
         EBTA = ebit / at,
         TLTA  = lt / at, # as a proxy for ME/TL
         SLTA = sale / at) %>%
    collect())
```

```
# A tibble: 44 x 14
   gvkey  fyear conm       at   wcap     re  ebit     lt  sale  WCTA  RETA  EBTA
   <chr>  <int> <chr>   <dbl>  <dbl>  <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
 1 001690  1980 APPLE~   65.4   16.3   14.5  23.6   39.4  117. 0.250 0.222 0.361
 2 001690  1981 APPLE~  255.   157.    54.1  66.1   77.5  335. 0.615 0.212 0.260
 3 001690  1982 APPLE~  358.   231.   116.  102.   101.   583. 0.647 0.324 0.286
 4 001690  1983 APPLE~  557.   340.   194.  130.   179.   983. 0.611 0.349 0.233
 5 001690  1984 APPLE~  789.   432.   256.   91.4  324.  1516. 0.548 0.324 0.116
 6 001690  1985 APPLE~  936.   527.   316.  147.   386.  1918. 0.563 0.337 0.157
 7 001690  1986 APPLE~ 1160.   712.   467.  274.   466.  1902. 0.614 0.403 0.236
 8 001690  1987 APPLE~ 1478.   829.   573.  371.   641.  2661. 0.561 0.387 0.251
 9 001690  1988 APPLE~ 2082.   956.   777.  620.  1079.  4071. 0.459 0.373 0.298
10 001690  1989 APPLE~ 2744.  1399.  1170.  634.  1258.  5284. 0.510 0.427 0.231
# i 34 more rows
# i 2 more variables: TLTA <dbl>, SLTA <dbl>
```

The following command allows to list all data Tables living in the `comp` (for Compustat) Schema and explore further Compustat Database if needed:

```
# compustat_list <- dbListObjects(wrds, Id(schema = "comp"))
# compustat_list
# cleaner
#https://stackoverflow.com/questions/43720911/list-tables-within-a-postgres-schema-using-r
# (compustat_tables <- wrds %>%
#   DBI::dbListObjects(DBI::Id(schema = 'comp')) %>%
#   dplyr::pull(table) %>%
#   purrr::map(~slot(.x, 'name')) %>%
#   dplyr::bind_rows())
```

Other tables of interest than `comp.funda`: `comp.fundq` (Fundamentals Quarterly, see here) and `comp.company` (see Security here).

`comp.fundq` being similar to `comp.funda`, we focus below on `comp.company` looking at APPLE INC (never defaulted) and ENRON (defaulted):

```
# WON'T WORK AFTER 31 DEC 2024
# company_db <- tbl(wrds, I("comp.company"))

# Replaced by local files
company_db <- readRDS("./wrds_data/company_all.rds")

(companies_master_data <- company_db %>%
  filter(grepl('^APPLE INC|ENRON CORP', conm))  %>%
    collect())
```

```
# A tibble: 3 x 39
  conm     gvkey add1  add2  add3  add4  addzip busdesc cik   city  conml costat
  <chr>    <chr> <chr> <chr> <chr> <chr> <chr>  <chr>   <chr> <chr> <chr> <chr>
1 APPLE I~ 0016~ One ~ <NA>  <NA>  <NA>  95014  Apple ~ 0000~ Cupe~ Appl~ A
2 ENRON C~ 0061~ Enro~ <NA>  <NA>  <NA>  77002  On Nov~ 0001~ Hous~ Enro~ I
3 ENRON C~ 0608~ 1400~ <NA>  <NA>  <NA>  77002~ <NA>    0000~ HOUS~ Enro~ I
# i 27 more variables: county <chr>, dlrsn <chr>, ein <chr>, fax <chr>,
#   fic <chr>, fyrc <int>, ggroup <chr>, gind <chr>, gsector <chr>,
#   gsubind <chr>, idbflag <chr>, incorp <chr>, loc <chr>, naics <chr>,
#   phone <chr>, prican <chr>, prirow <chr>, priusa <chr>, sic <chr>,
#   spcindcd <int>, spcseccd <int>, spcsrc <chr>, state <chr>, stko <int>,
#   weburl <chr>, dldte <date>, ipodate <date>
```

In particular we will see later that fields `dlrsn`, `dldte` contains information that will be usefull to detect default/bankruptcy:

```
companies_master_data %>% select(conm, gvkey, dlrsn, dldte, fyrc)
```

```
# A tibble: 3 x 5
  conm              gvkey  dlrsn dldte       fyrc
  <chr>             <chr>  <chr> <date>     <int>
1 APPLE INC         001690 <NA>  NA             9
2 ENRON CORP        006127 02    2005-04-30    12
3 ENRON CORP  -OLD 060874 05    1995-06-30    12
```

1.3 Data files on Google Drive / Moodle (only $<$ 100Mb)

Corresponding data files (`.rds`) containing all Compustat/CRSP data up to 30 September 2024 are available in the Google Drive / Moodle (only $<$ 100Mb) project folder `wrds_data`:

- Fundamentals Annual (`comp.funda`):

```
compustat_all <- readRDS("./wrds_data/compustat_all_light.rds")
# compustat_all <- readRDS("./wrds_data/compustat_all.rds") # uncomment for full data (~26


compustat_all %>%
  filter(grepl('^APPLE INC|ENRON CORP', conm)) %>%
  select(gvkey, fyear, conm, at, wcap, re, ebit, lt, sale) %>%
  mutate(WCTA = wcap / at,
         RETA = re / at,
         EBTA = ebit / at,
         TLTA  = lt / at, # as a proxy for ME/TL
         SLTA = sale / at)
```

```
# A tibble: 83 x 14
   gvkey  fyear conm       at   wcap     re  ebit     lt  sale  WCTA  RETA  EBTA
   <chr>  <int> <chr>   <dbl>  <dbl>  <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
 1 001690  1980 APPLE~   65.4   16.3   14.5  23.6   39.4  117. 0.250 0.222 0.361
 2 001690  1981 APPLE~  255.   157.    54.1  66.1   77.5  335. 0.615 0.212 0.260
 3 001690  1982 APPLE~  358.   231.   116.  102.   101.   583. 0.647 0.324 0.286
 4 001690  1983 APPLE~  557.   340.   194.  130.   179.   983. 0.611 0.349 0.233
 5 001690  1984 APPLE~  789.   432.   256.   91.4  324.  1516. 0.548 0.324 0.116
 6 001690  1985 APPLE~  936.   527.   316.  147.   386.  1918. 0.563 0.337 0.157
```

```
 7 001690  1986 APPLE~ 1160.   712.   467.  274.   466.  1902. 0.614 0.403 0.236
 8 001690  1987 APPLE~ 1478.   829.   573.  371.   641.  2661. 0.561 0.387 0.251
 9 001690  1988 APPLE~ 2082.   956.   777.  620.  1079.  4071. 0.459 0.373 0.298
10 001690  1989 APPLE~ 2744.  1399.  1170.  634.  1258.  5284. 0.510 0.427 0.231
# i 73 more rows
# i 2 more variables: TLTA <dbl>, SLTA <dbl>
```

```
# # The data has been previously collected this way:
# funda_db <- tbl(wrds, in_schema("comp", "funda"))

# # Setting the analysis horizon
# start_date <- lubridate::ymd("1960-01-01")
# end_date <- lubridate::ymd("2023-06-30")

# compustat_all <- funda_db %>%
#     filter(
#         indfmt == "INDL" &
#             datafmt == "STD" &
#             consol == "C" &
#             datadate >= start_date & datadate <= end_date) %>%
#     mutate(year = year(datadate)) %>%
#     group_by(gvkey, year) %>%
#     # removing possible duplicates (ie multiple data update for same company/fiscal year
#         filter(datadate == max(datadate)) %>%
#     ungroup() %>%
#     select(-year) %>%
#     collect()

# saveRDS(compustat_all, "./wrds_data/compustat_all.rds")
# toy data set for quarto
# usual_suspects <- 'APPLE INC|ENRON CORP$|WORLDCOM|GENERAL MOTORS|CHRYSLER|TEXACO|CALPINE
# compustat_all_light <- compustat_all %>%
#    filter(grepl(usual_suspects, conm))
# saveRDS(compustat_all_light, "./wrds_data/compustat_all_light.rds")
```

We provide a variable dictionary (also available here with definitions):

```
(compustat_variables <- readRDS("./wrds_data/compustat_funda_variables.rds"))
```

```
# A tibble: 940 x 2
   variable_postgres Description
```

```
   <chr>             <chr>
 1 gvkey             GVKEY -- Global Company Key (GVKEY)
 2 conm              Company Name (CONM)
 3 tic               Ticker Symbol (TIC)
 4 cusip             CUSIP (CUSIP)
 5 cik               CIK Number (CIK)
 6 exchg             Stock Exchange Code (EXCHG)
 7 fyr               Fiscal Year-End (FYR)
 8 fic               Foreign Incorporation Code (FIC)
 9 acctchg           ACCTCHG -- Adoption of Accounting Changes (ACCTCHG)
10 acctstd           ACCTSTD -- Accounting Standard (ACCTSTD)
# i 930 more rows
```

- Company (`comp.company`):

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

company_all %>%
  filter(grepl('^APPLE INC|ENRON CORP', conm)) %>%
  select(conm, gvkey, dlrsn, dldte, fyrc)
```

```
# A tibble: 3 x 5
  conm              gvkey  dlrsn dldte       fyrc
  <chr>             <chr>  <chr> <date>     <int>
1 APPLE INC         001690 <NA>  NA             9
2 ENRON CORP        006127 02    2005-04-30    12
3 ENRON CORP  -OLD 060874 05    1995-06-30    12
```

```
# The data has been previously collected this way:
# company_db <- tbl(wrds, in_schema("comp", "company"))
# company_all <- company_db %>%
#     collect()

# saveRDS(company_all, "./wrds_data/company_all.rds")
```

We provide a variable dictionary (also available here with definitions):

```
(company_variables <- readRDS("./wrds_data/compustat_company_variables.rds"))
```

```
# A tibble: 39 x 2
   variable_postgres Description
```

```
   <chr>             <chr>
 1 add1              Address Line 1
 2 add2              Address Line 2
 3 add3              Address Line 3
 4 add4              Address Line 4
 5 addzip            Postal Code
 6 busdesc           S&P Business Description
 7 cik               CIK Number
 8 city              City
 9 conm              Company Name
10 conml             Company Legal Name
# i 29 more rows
```

In particular two fields relate to companies deletion:

```
company_variables %>% filter(variable_postgres %in% c('dldte','dlrsn'))
```

```
# A tibble: 2 x 2
  variable_postgres Description
  <chr>             <chr>
1 dldte             Research Company Deletion Date
2 dlrsn             Research Co Reason for Deletion
```

We see below that the company ENRON CORP was removed from Compustat for reasons 02 and 05, while APPLE INC is still alive:

```
companies_master_data %>% select(conm, gvkey, dlrsn, dldte, fyrc)
```

```
# A tibble: 3 x 5
  conm              gvkey  dlrsn dldte       fyrc
  <chr>             <chr>  <chr> <date>     <int>
1 APPLE INC         001690 <NA>  NA             9
2 ENRON CORP        006127 02    2005-04-30    12
3 ENRON CORP  -OLD 060874 05    1995-06-30    12
```

The following table gives definitions for deletion codes, in particular 02 refers to bankruptcy (ie Chapter 11) and 03 to liquidation (ie Chapter 7):

```
inact_all <- readRDS("./wrds_data/inact_all.rds")
inact_all %>% filter(as.integer(dlrsncd) <= 10 )
```

```
# A tibble: 9 x 2
  dlrsncd dlrsndesc
  <chr>   <chr>
1 01      Acquisition or merger
2 02      Bankruptcy
3 03      Liquidation
4 04      Reverse Acquisition
5 05      No longer fits file format
6 06      Leveraged buyout
7 09      Now a private company
8 10      Other (no SEC filings, etc)
9 07      Other (no longer files with SEC among other possible reasons) but pri~
```

So a way to detect the event of default/bankruptcy is to screen companies in `comp.company` table for which deletion codes are either `02` (ie Chapter 11) or `03` (ie Chapter 7).

Below we give a brief overview of (not)deleted companies, ca. 18k are alive, 15k. have been acquired (but the may obfuscate a bankruptcy), 2k were liquidated, 1k gone bankrupt:

```
dlrsn_companies <- company_all %>%
    select(gvkey, conm, ipodate, dldte, dlrsn) %>%
    left_join(inact_all, by = c("dlrsn"="dlrsncd"))

dlrsn_companies %>%
    group_by(dlrsndesc) %>%
    summarize(n = n_distinct(conm)) %>%
    arrange(desc(n))
```

```
# A tibble: 11 x 2
   dlrsndesc                                                              n
   <chr>                                                              <int>
 1 <NA>                                                               18513
 2 Acquisition or merger                                              16155
 3 Other (no SEC filings, etc)                                        11877
 4 Liquidation                                                         2549
 5 Other (no longer files with SEC among other possible reasons) but pric~  2095
 6 Bankruptcy                                                          1044
 7 Reverse Acquisition                                                  990
 8 Now a private company                                                871
 9 No longer fits file format                                            99
10 Other (issue-level activity; company remains active on the file)      92
11 Leveraged buyout                                                      91
```

1.4 CRSP

The CRSP Daily Security Data is a financial database that tracks stock prices and related information for U.S. companies. It provides daily updates on stock prices, returns, trading volume, and other market data. Researchers and investors use this data to analyze stock market trends, evaluate company performance, and study investment strategies. Its usage through WRDS with R is described in depth in the Tidy finance book here, not that the Database has beeen recently updated to a new format (CRSP-V2) that we will use.

Data is available on Google Drive / Moodle in the folder `wrds_data`:

```
crsp_daily <- readRDS("./wrds_data/crsp_daily_light.rds")
# crsp_daily <- readRDS("./wrds_data/crsp_daily.rds") # uncomment for full

# # The data has been previously collected this way:
# # Setting the analysis horizon
# start_date <- lubridate::ymd("1960-01-01")
# end_date <- lubridate::ymd("2024-09-30")
#
# # CRSP daily
#
# # CRSP monthly daily file (v2 format)
# dsf_db <- tbl(wrds, I("crsp.dsf_v2"))
# # identifying information
# stksecurityinfohist_db <- tbl(wrds, I("crsp.stksecurityinfohist"))
#
# permnos <- stksecurityinfohist_db %>%
#   distinct(permno) %>%
#   pull(permno)
#
# batch_size <- 500
# 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))
#     ]
#
#     crsp_daily_sub <- dsf_db %>%
#     filter(permno %in% permno_batch) %>%
```

```
#     filter(dlycaldt >= start_date & dlycaldt <= end_date) %>%
#     inner_join(
#       stksecurityinfohist_db %>%
#         filter(sharetype == "NS" &
#                securitytype == "EQTY" &
#                securitysubtype == "COM" &
#                usincflg == "Y" &
#                issuertype %in% c("ACOR", "CORP") &
#                primaryexch %in% c("N", "A", "Q") &
#                conditionaltype %in% c("RW", "NW") &
#                tradingstatusflg == "A") %>%
#         select(permno, secinfostartdt, secinfoenddt),
#       join_by(permno)
#     ) %>%
#     filter(dlycaldt >= secinfostartdt & dlycaldt <= secinfoenddt)  %>%
#     select(permno,
#            cusip = hdrcusip,
#            date = dlycaldt,
#            vol = dlyvol,
#            shrout,
#            prc = dlyprc,
#            cap = dlycap,
#            close = dlyclose,
#            low = dlylow,
#            high = dlyhigh,
#            bid = dlybid,
#            ask = dlyask,
#            open = dlyopen,
#            ret = dlyret) %>%
#     collect() # %>%
#     # drop_na() # this causes common stocks like IBM with NA bid/ask to be removed from
#
#     if (nrow(crsp_daily_sub) > 0) {
#         dbWriteTable(tidy_finance,
#                      "crsp_daily",
#                      value = crsp_daily_sub,
#                      overwrite = ifelse(j == 1, TRUE, FALSE),
#                      append = ifelse(j != 1, TRUE, FALSE)
#         )
#     }
#
```

```
#     message("Batch ", j, " out of ", batches, " done (", percent(j / batches), ")\n")
# }
#
# crsp_daily_db <- tbl(tidy_finance, "crsp_daily")
#
# crsp_daily <- crsp_daily_db %>% collect()
# saveRDS(crsp_daily, "./wrds_data/crsp_daily_full.rds")


# usual_suspects <- 'APPLE INC|ENRON CORP$|WORLDCOM|GENERAL MOTORS|CHRYSLER|TEXACO|CALPINE
# comp_sub <- company_all %>% filter(grepl(usual_suspects, conm))
# subset_mapping <- ccmxpf_linktable %>%
#   filter(gvkey %in% unique(comp_sub$gvkey)) %>%
#   left_join(company_all %>% select(gvkey, conm), by = c("gvkey"))
#
# crsp_daily_light <- crsp_daily %>%
#   left_join(subset_mapping, by = c("permno"), relationship = "many-to-many") %>%
#     filter(!is.na(gvkey) &
#            (date >= linkdt & date <= linkenddt))
#
# saveRDS(crsp_daily_light, "./wrds_data/crsp_daily_light.rds")
```

1.5 Linking Compustat and CRSP

CRSP and Compustat use different keys to identify stocks and firms. CRSP uses `permno` for stocks, while Compustat uses `gvkey` to identify firms (see here). WRDS provides a mapping table allowing us to merge CRSP and Compustat, the CRSP/Compustat Merged (CCM) Database (see here).

We use here instructions from the `Tidy Finance` book available here.

We first create a mapping table from CRSP (`permno` identifier) to Compustat (`gvkey` identifier):

```
# WON'T WORK AFTER 31 DEC 2024
# ccmxpf_linktable_db <- tbl(
#   wrds,
#   in_schema("crsp", "ccmxpf_linktable")
# )

# ccmxpf_linktable <- ccmxpf_linktable_db %>%
```

```
#   filter(linktype %in% c("LU", "LC") &
#     linkprim %in% c("P", "C") &
#     usedflag == 1) %>%
#   select(permno = lpermno, gvkey, linkdt, linkenddt) %>%
#   collect() %>%
#   mutate(linkenddt = replace_na(linkenddt, today()))

# Replaced by local files
ccmxpf_linktable <- readRDS("./wrds_data/ccmxpf_linktable.rds")
```

We then select three particular companies from Compustat, for illustrative purposes:

```
acc <- funda_db %>%
  filter(grepl('^APPLE INC|ENRON CORP$|EASTMAN KODAK', conm)) %>%
  select(gvkey, fyear, conm, at, wcap, re, ebit, lt, sale) %>%
  mutate(WCTA = wcap / at,
         RETA = re / at,
         EBTA = ebit / at,
         TLTA  = lt / at, # as a proxy for ME/TL
         SLTA = sale / at) %>%
  collect()
acc
```

```
# A tibble: 147 x 14
   gvkey  fyear conm       at   wcap     re  ebit     lt  sale  WCTA  RETA  EBTA
   <chr>  <int> <chr>   <dbl>  <dbl>  <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
 1 001690  1980 APPLE~   65.4   16.3   14.5  23.6   39.4  117. 0.250 0.222 0.361
 2 001690  1981 APPLE~  255.   157.    54.1  66.1   77.5  335. 0.615 0.212 0.260
 3 001690  1982 APPLE~  358.   231.   116.  102.   101.   583. 0.647 0.324 0.286
 4 001690  1983 APPLE~  557.   340.   194.  130.   179.   983. 0.611 0.349 0.233
 5 001690  1984 APPLE~  789.   432.   256.   91.4  324.  1516. 0.548 0.324 0.116
 6 001690  1985 APPLE~  936.   527.   316.  147.   386.  1918. 0.563 0.337 0.157
 7 001690  1986 APPLE~ 1160.   712.   467.  274.   466.  1902. 0.614 0.403 0.236
 8 001690  1987 APPLE~ 1478.   829.   573.  371.   641.  2661. 0.561 0.387 0.251
 9 001690  1988 APPLE~ 2082.   956.   777.  620.  1079.  4071. 0.459 0.373 0.298
10 001690  1989 APPLE~ 2744.  1399.  1170.  634.  1258.  5284. 0.510 0.427 0.231
# i 137 more rows
# i 2 more variables: TLTA <dbl>, SLTA <dbl>
```

We filter the mapping table for these companies, we notice that mappings/links between `permno` (CRSP) and `gvkey` (Compustat) are valid between two dates, so that one `gvkey`

can be linked to many `permnos` depending on the value date (see EASTMAN KODAK), and conversely:

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

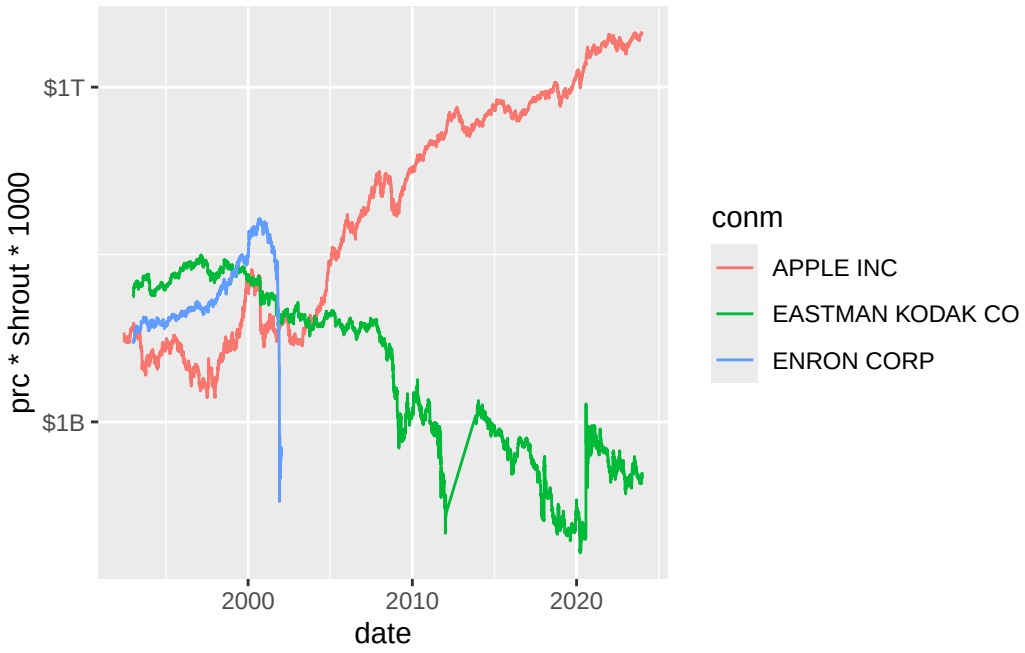
```
# A tibble: 6 x 5
  permno gvkey  linkdt     linkenddt  conm
   <dbl> <chr>  <date>     <date>     <chr>
1  14593 001690 1980-12-12 2024-10-12 APPLE INC
2  11754 004194 1950-01-01 1962-01-30 EASTMAN KODAK CO
3  11754 004194 1962-01-31 2012-01-18 EASTMAN KODAK CO
4  14276 004194 2013-11-01 2024-10-12 EASTMAN KODAK CO
5  23317 006127 1962-01-01 1962-01-30 ENRON CORP
6  23317 006127 1962-01-31 2002-01-11 ENRON CORP
```

We then merge the daily CRSP data set with the preceding mapping (for the selected 3 companies) to add columns refering to Compustat identifiers:

```
merged_crsp_compustat_sub <- crsp_daily %>%
  select(cusip, permno, date, prc, vol, shrout, bid, ask) %>%
  left_join(subset_mapping, by = c("permno"), relationship = "many-to-many") %>%
  filter(!is.na(gvkey) &
           (date >= linkdt & date <= linkenddt))
```

As a dumb example of linking CRSP to Compustat, we can plot market capitalization of APPLE, ENRON and EASTMAN KODAK: using company names `conm` from Compustat) and using daily prices (`prc`) and outstanding shares (`shrout`) from CRSP; we use a log scale:

```
ggplot(merged_crsp_compustat_sub,
       aes(x=date, y=prc*shrout*1000, color=conm)) +
  geom_line() +
  scale_y_log10(breaks = log_breaks(4, 1000),
                labels = scales::label_dollar(scale_cut = cut_short_scale()))
```



2 Bankruptcy / Default data

We describe below two additional sources of bankruptcy/default data for US corporations.

The first LoPucki is freely available online and focuses on bankruptcy cases for US public companies.

The second is proprietary and deals with bankruptcy and default. Data has been extracted from Annual reports published by the rating agency Moody's.

2.1 LoPucki

The LoPucki Bankruptcy Research Database tracks large U.S. companies that file for Chapter 11 bankruptcy and covers 1980-2022.

It provides detailed information on each case, including financial data, court decisions, and outcomes.

The database helps researchers and professionals understand how companies handle bankruptcy and what factors affect their recovery or failure. We won't need necessarily all the information, but at least the date of event (Chapter 7 or 11), and fortunately the `gvkey` (as `GvkeyBefore`) of the company, allowing to match Compustat data. It has been discontinued in January 2023.

Data is available online, or in the `default_data` folder of Google Drive / Moodle:

```
lopucki <- readxl::read_xlsx(
    "./default_data/Bankruptcy - LoPucki/Florida-UCLA-LoPucki Bankruptcy Research Database

# Some companies fill multiple time (eg American Apparel)
lopucki %>%
  group_by(GvkeyBefore) %>%
  mutate(n=n()) %>%
  ungroup() %>%
  filter(n>1)
```

```
# A tibble: 224 x 218
   NameCorp AfterEmerging Assets1Before Assets2Before Assets3Before AssetsBefore
   <chr>    <chr>                 <dbl>         <dbl>         <dbl>        <dbl>
 1 Advance~ Acquired by ~          412.           NA            NA          412.
 2 Alleghe~ <NA>                   850.         1283.          1511          850.
 3 AM Inte~ <NA>                   546.          686            544          546.
 4 AM Inte~ Change name ~          441.          600.           NA          441.
 5 Amcast ~ refiled; liq~           NA           230            NA          230
 6 Amcast ~ <NA>                    NA            NA           230          230
 7 America~ <NA>                   294.           NA            NA          294.
 8 America~ <NA>                    NA           294.           NA          294.
 9 Amerise~ <NA>                    NA          1887           1462         1887
10 AmeriTr~ <NA>                   264.          193            141          264.
# i 214 more rows
# i 212 more variables: AssetsCurrDollar <dbl>, AssetsEmerging <dbl>,
#   AssetsPetCurrDollar <dbl>, AssetsPetition <dbl>, BondISIN <chr>,
#   BondPriceDisp <dbl>, BondPriceFile <dbl>, BondPriceMoveDuring <dbl>,
#   CaseNum <chr>, CaseNumRefile <chr>, CaseNumTransfer <chr>, CeoFiling <chr>,
#   CeoNotes <chr>, CeoReplaced <chr>, CikBefore <chr>, CikEmerging <chr>,
#   CityChange <chr>, CityDisposed <chr>, CityFiledCategory <chr>, ...
```

We retain only key information, namely company name, date the bankruptcy case has been filed (which materializes the event we are interested in), Compustat identifier and type of bankruptcy:

```
# Extract Chapter 7/11
# When multiple filings for a company (ie gvkey) you can select the min date (bankruptcy f
lopucki_clean <- lopucki %>%
    select(NameCorp, Chapter, GvkeyBefore, DateFiled) %>%
```

```
    filter(Chapter %in% c('7', '11')) %>%
    group_by(GvkeyBefore) %>%
    summarize(DateFiled = min(DateFiled),
              NameCorp = NameCorp[which.min(DateFiled)],
              Chapter = Chapter[which.min(DateFiled)]) %>%
    mutate(DateFiled = lubridate::as_date(DateFiled)) %>%
    ungroup()
```

Using the field `GvkeyBefore` we are able to match Compustat, for example regarding ENRON CORP one of the largest corporate bankruptcy in U.S. history:

```
gvkey_enron <- company_all %>% filter(conm == 'ENRON CORP') %>% pull(gvkey)

lopucki_clean %>% filter(GvkeyBefore == gvkey_enron)
```

```
# A tibble: 1 x 4
  GvkeyBefore DateFiled  NameCorp    Chapter
  <chr>       <date>     <chr>       <chr>
1 006127      2001-12-02 Enron Corp. 11
```

Compustat deleted ENRON CORP from its database in 2005:

```
company_all %>% filter(gvkey == gvkey_enron) %>% select(conm, dldte, dlrsn)
```

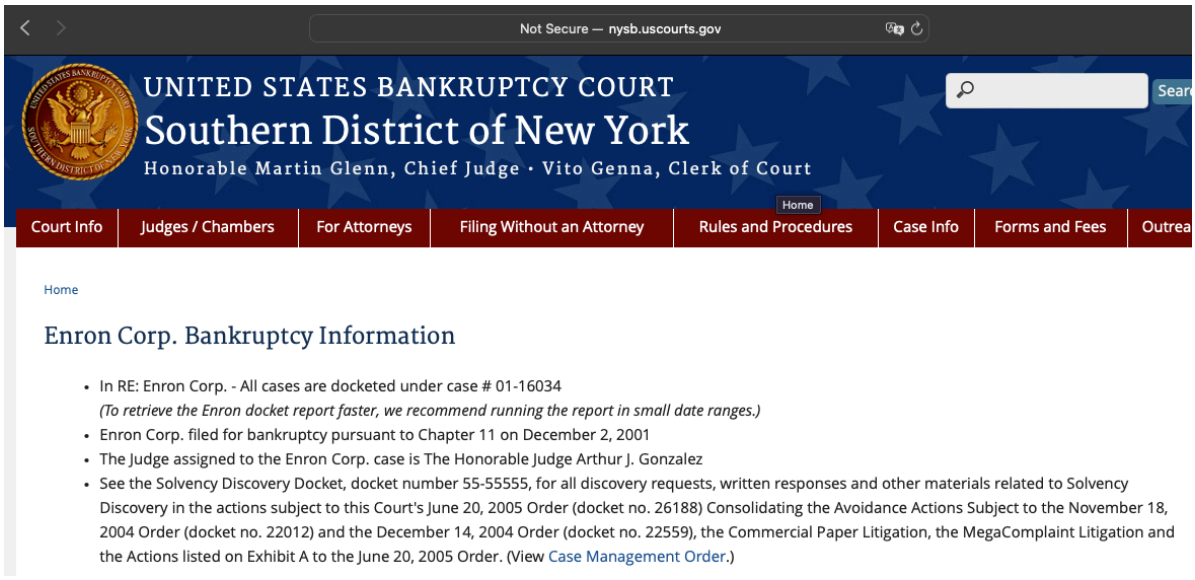
```
# A tibble: 1 x 3
  conm       dldte      dlrsn
  <chr>      <date>     <chr>
1 ENRON CORP 2005-04-30 02
```

and last available financial statements pertain to fiscal year 2000:

```
compustat_all %>%
  filter(gvkey == gvkey_enron, fyear > 1998) %>%
  select(gvkey, conm, datadate, fyear, fdate)
```

```
# A tibble: 2 x 5
  gvkey  conm       datadate   fyear fdate
  <chr>  <chr>      <date>     <int> <date>
1 006127 ENRON CORP 1999-12-31  1999 NA
2 006127 ENRON CORP 2000-12-31  2000 NA
```

LoPucki is more current in that case and matches what happened in real life:



LoPucki can be used to improve Compustat coverage and currency regarding bankruptcy data.

2.2 Moody's Annual Default Reports

Moody's Annual Default Reports for Corporates provide detailed analysis of corporate bond defaults and credit rating changes. These reports track default rates, recovery rates, and rating transitions for companies around the world. They help investors, analysts, and researchers assess credit risk and understand trends in corporate defaults and the performance of credit ratings over time.

Data was extracted from Moody's Annual reports covering Corporate Default starting in 2002, it is more heterogeneous than LoPucki and not directly linkable to Compustat since only the name of the company is reported, also depending on the year, either full date, the month or nothing (in that case we use the end of year for the event) is reported regarding the default/bankruptcy event.

Data is available in the `default_data` folder of Google Drive / Moodle:

```
dat_default_moodys_annual <- readRDS("./default_data/dat_default_moodys_annual.rds")
dat_default_moodys_annual %>%
  filter(country %in% c('United States', 'Canada')) %>%
  select(company_name, default_date, default_type)
```

```
# A tibble: 1,350 x 3
   company_name                          default_date default_type
   <chr>                                 <date>       <chr>
 1 AAi.FosterGrant, Inc.                 2002-01-31   Missed interest payment
 2 Archibald Candy Corporation           2002-01-31   Missed interest payment
 3 BTI Telecom Corporation               2002-01-31   Distressed exchange
 4 Duty Free International               2002-01-31   Grace period default
 5 Frontier Corporation                  2002-01-31   Chapter 11
 6 Glasstech, Inc.                       2002-01-31   Chapter 11
 7 Global Crossing Holdings Limited      2002-01-31   Chapter 11
 8 Hartmarx Corporation                  2002-01-31   Distressed exchange
 9 IT Group, Inc.                        2002-01-31   Chapter 11
10 Kaiser Aluminum & Chemical Corporation 2002-01-31   Missed interest payment
# i 1,340 more rows
```

Nonetheless, playing a little bit with the company names, 459 companies out of 1400 directly match Compustat:

```
trim_pattern <- ",|\\.|INC|LLC|CORPORATION|CORP|COMPANY|\\*| "

dat_default_moodys_annual <- dat_default_moodys_annual %>%
mutate(
  compact_conm = stringr::str_to_upper(company_name),
  compact_conm = stringr::str_trim(gsub(trim_pattern, "", compact_conm))
      )

company_all <- company_all %>%
  mutate(compact_conm = stringr::str_trim(gsub(trim_pattern, "", conm)))

moodys_default <- dat_default_moodys_annual %>%
                    left_join(company_all %>%
                                select(gvkey, conm, compact_conm, dldte, dlrsn),
                                  by = c("compact_conm")) %>%
                    filter(!is.na(gvkey)) %>%
                    select(company_name, gvkey, default_date, dldte, default_type, dlrsn)
moodys_default
```

```
# A tibble: 459 x 6
   company_name                gvkey default_date dldte      default_type dlrsn
   <chr>                       <chr> <date>       <date>     <chr>        <chr>
 1 AAi.FosterGrant, Inc.       1281~ 2002-01-31   2002-03-19 Missed inte~ 10
 2 Archibald Candy Corporation 1285~ 2002-01-31   2003-01-23 Missed inte~ 10
```

```
 3 BTI Telecom Corporation      1234~ 2002-01-31   2004-09-20 Distressed ~ 10
 4 Duty Free International      0154~ 2002-01-31   1997-10-31 Grace perio~ 01
 5 Frontier Corporation         0091~ 2002-01-31   1999-09-29 Chapter 11   01
 6 Glasstech, Inc.              1463~ 2002-01-31   2015-04-20 Chapter 11   10
 7 Hartmarx Corporation         0055~ 2002-01-31   2013-06-24 Distressed ~ 10
 8 IT Group, Inc.               0061~ 2002-01-31   2003-12-15 Chapter 11   07
 9 Kaiser Aluminum & Chemical ~ 0145~ 2002-01-31   2006-07-06 Missed inte~ 10
10 Evenflo Company, Inc.        1462~ 2002-02-28   2004-08-16 Missed inte~ 10
# i 449 more rows
```

A lot of deleted companies from Compustat Database for reason different than credit (Acquisitions, Other) are in fact defaulting companies:

```
moodys_default %>% filter(dldte > default_date | !dlrsn %in% c('02', '03'))
```

```
# A tibble: 449 x 6
   company_name                gvkey default_date dldte      default_type dlrsn
   <chr>                       <chr> <date>       <date>     <chr>        <chr>
 1 AAi.FosterGrant, Inc.       1281~ 2002-01-31   2002-03-19 Missed inte~ 10
 2 Archibald Candy Corporation 1285~ 2002-01-31   2003-01-23 Missed inte~ 10
 3 BTI Telecom Corporation     1234~ 2002-01-31   2004-09-20 Distressed ~ 10
 4 Duty Free International     0154~ 2002-01-31   1997-10-31 Grace perio~ 01
 5 Frontier Corporation        0091~ 2002-01-31   1999-09-29 Chapter 11   01
 6 Glasstech, Inc.             1463~ 2002-01-31   2015-04-20 Chapter 11   10
 7 Hartmarx Corporation        0055~ 2002-01-31   2013-06-24 Distressed ~ 10
 8 IT Group, Inc.              0061~ 2002-01-31   2003-12-15 Chapter 11   07
 9 Kaiser Aluminum & Chemical ~ 0145~ 2002-01-31   2006-07-06 Missed inte~ 10
10 Evenflo Company, Inc.       1462~ 2002-02-28   2004-08-16 Missed inte~ 10
# i 439 more rows
```

Using `fuzzy matching` techniques it is possible to match more corporations.

Looking a the academic literature, for example Duffie (2007), the bankruptcy event is usually defined combining different sources, below a more complete Moody's database than ours and Compustat (using deletion codes `02` (ie Chapter 11) or `03` (ie Chapter 7)):

- Bankruptcy. An exit is treated for our purposes as a bankruptcy if coded in Moodys database under any of the following categories of events: Bankruptcy, Bankruptcy Section 77, Chapter 10,[12] Chapter 11, Chapter 7, and Prepackaged Chapter 11. A bankrutcy is also recorded if data item AFTNT35 of Compustat is 2 or 3 (for Chapter 11 and Chapter 7, respectively). In some cases, our data reflect bankruptcy exits based on information from Bloomberg and other data sources. Our dataset has 175 bankruptcy exits, although many defaults that

3 Practical 1 - Data set construction

- **Task 1**: Your aim is to build a data set that you will use to predict default/bankruptcy for US public corporations. You need basically to build as set of target variables Y (event of default) and predictors X made of financial data for US companies observed at various fiscal years end (so-called "firm year" observations in Academic literature).
- **Task 2**: Apply a first baseline Logistic Regression model for example using Altman's ratios. Altman's Z-Score components (X1-X5) (as in Altman (1968)) are shown below:

ent ratio-profiles. The final discriminant function is as follows:

$$\text{(I)} \quad Z = .012X_1 + .014X_2 + .033X_3 + .006X_4 + .999X_5$$

where X_1 = Working capital/Total assets
X_2 = Retained Earnings/Total assets
X_3 = Earnings before interest and taxes/Total assets
X_4 = Market value equity/Book value of total debt
X_5 = Sales/Total assets
Z = Overall Index

In a first step (it allows you to work only with Compustat data), or as "bis-baseline" model you can proxy $X_4 = \frac{\text{Market Value}}{\text{Total liabilities}}$ with $\tilde{X}_4 = \frac{\text{Total Assets}}{\text{Total liabilities}}$. Market value of firms is available joining Compustat with CRSP.

For this model you will use Shumway's discrete hazard or dynamic approach (article Shumway (2001) available here).

I recommend that you gradually read and try to understand it as it describes well the process of building a bankruptcy prediction model). Roughly speaking the discrete hazard model is

equivalent to a (multi-period) Logistic Regression model[1] in which the observations are "firm year".

Shumway's approach allows to add macroeconomic variables (shared by all firms at a given point of time) and time-varying market variables (eg. stock volatility, distance-to-default, returns, relating to a specific company) as predictors.

Below a sketch of proof (to be improved) giving more details than in Shumway (2001):

Sketch of proof

TEMPORARY, should be better developed/explained.
We re-use the presentation in Suresh et al. (2022) (available here), see also Allison (1982) or Singer & Willett (1993)
We first briefly introduce the framework of discrete-time survival models.
In the survival setting, the observed data is given by $\mathcal{D}_n = \{\tilde{T}_i, \delta_i, X_i; i = 1, ..., n\}$ where $\tilde{T}_i = \min(T_i, C_i)$ is the observed event time for the $i-$th individual $(i = 1, ..., n)$, with T_i denoting the true event time, C_i the censoring time, $\delta_i = \mathbf{1}_{T_i \leq C_i}$ the event indicator, and $X_i = (Xi_1, ..., Xi_J)$ the observed baseline covariate vector.
We consider the common situation of a right-censored survival outcome, where if an individual's observed survival time is censored we know only that they experienced the event beyond that time. We also assume independent or noninformative censoring conditional on covariates, such that T_i and C_i are independent random variables given X_i.
We record a single "nonrepeatable" event occurrence (here the bankruptcy/failure) in discrete intervals dividing continuous time into a sequence of J contiguous time periods $]t_0, t_1],]t_1, t_2], ...,]t_{J-1}, t_J]$. In our case we effectively observe the event on yearly intervals, on a specific horizon $]t_0, t_J]$.
Quoting Singer & Willett (1993), discrete-time survival models focus on "whether and if so when (in which time period) the single"nonrepeatable" event occurs". Because each individual can experience the target event only once, event occurrence is inherently conditional. An individual can experience the event in a time period j only if he or she did not already experience it any of the earlier time periods prior to j. Similarly, once an individual experiences the event, he or she cannot experience it again in any later time period.
Within this framework the hazard in a particular time interval is the probability of an individual (here a firm) experiencing the event during that interval given that they have survived up to the start of that interval.
For an individual with predictor X_i, the hazard (or hazard function) in interval $A_j =]t_{j-1}, t_j]$ is the conditional probability:

[1] "The hazard model can be thought of as a binary logit model that includes each firm year as a separate observation." Shumway (2001)

$$\begin{aligned}\lambda_{ij}(X_i) &= \mathbb{P}(T_i \in A_j | T_i > t_{j-1}, X_i) \\ &= \mathbb{P}(t_{j-1} < T_i \leq t_j | T_i > t_{j-1}, X_i)\end{aligned}$$

Additionally the Survival function or survival probability is usually introduced:

$$S_i(t|X_i) = \mathbb{P}(T_i > t|X_i) = \prod_{j|t_j \leq t} (1 - \lambda_{ij}(X_i))$$

We now express the likelihood function for the discrete-time hazard model.
To construct the likelihood, we first focus on individual/subject i.
Individual i contributes the product of the conditional survival probabilities for the time intervals in which they are observed but do not experience the event. Individuals that are observed to have a failure ($\delta_i = 1$) additionally contribute the conditional failure probability in the interval in which they experience the event of interest. We use j_i to denote the last interval during which we have information about individual i, such that $T_i \in A_{j_i}$. Individual i does not contribute any information to the likelihood for intervals beyond A_{j_i}.
For an uncensored individual:

$$\begin{aligned}\mathbb{P}(T_i \in A_{j_i}|X_i) &= \mathbb{P}(T_i = t_{j_i}|X_i) \\ &= \mathbb{P}(T_i = t_{j_i}|T_i > t_{j-1}, X_i)\mathbb{P}(T_i \neq t_{j_i-1}|T_i > t_{j-2}, X_i)...\mathbb{P}(T_i \neq t_1|T_i > t_0, X_i) \\ &= \lambda_{ij_i}(X_i)(1 - \lambda_{ij_{i-1}}(X_i))...(1 - \lambda_{i1}(X_i)) \\ &= \lambda_{ij_i} \prod_{j=1}^{j_i-1} (1 - \lambda_{ij}(X_i))\end{aligned}$$

where we have used the definition of hazard function λ_{ij}.
Similarly, for a censored individual:

$$\begin{aligned}\mathbb{P}(T_i > t_{j_i}|X_i) &= \mathbb{P}(T_i \neq t_{j_i}|T_i > t_{j-1}, X_i)\mathbb{P}(T_i \neq t_{j_i-1}|T_i > t_{j-2}, X_i)...\mathbb{P}(T_i \neq t_1|T_i > t_0, X_i) \\ &= (1 - \lambda_{ij_i}(X_i))(1 - \lambda_{ij_{i-1}}(X_i))...(1 - \lambda_{i1}(X_i)) \\ &= \prod_{j=1}^{j_i} (1 - \lambda_{ij}(X_i)) \\ &= S_i(t_{j_i}|X_i)\end{aligned}$$

Finally, the likelihood writes as the product for each i of the two preceding:

$$
\begin{aligned}
L =& \prod_{i=1}^{n} \left[\mathbb{P}(T_i = t_{j_i}|X_i)\right]^{\delta_i} \left[\mathbb{P}(T_i > t_{j_i}|X_i)\right]^{1-\delta_i} \\
=& \prod_{i=1}^{n} \left[\lambda_{ij_i}(X_i) \prod_{j=1}^{j_i-1}(1-\lambda_{ij}(X_i))\right]^{\delta_i} \left[\prod_{j=1}^{j_i}(1-\lambda_{ij}(X_i))\right]^{1-\delta_i}
\end{aligned}
$$

assuming individuals i are independent conditionally on X_i.
Introducing an event history indicator $y_{ij} = I(T_i \in A_{j_i}) = I(t_{j-1} < T_i \leq t_j)$, which for censored subjects is given by $(y_{i1}, \ldots, y_{ij_i}) = (0, \ldots, 0)$ and for subjects that experience the event is $(y_{i1}, \ldots, y_{ij_i}) = (0, \ldots, 1)$. The likelihood can then be written as:

$$
L = \prod_{i=1}^{n} \prod_{j=1}^{j_i} \lambda_{ij}(X_i)^{y_{ij}} (1-\lambda_{ij}(X_i))^{1-y_{ij}}
$$

which is equivalent to the likelihood of a binomial model with independent observations $y_i j$, subject-specific probabilities $\lambda_{ij}(X_i)$ for subject i experiencing the event in interval $]t_{j-1}, t_j]$, and with predictor X_i. Note that we do not make the assumption that the event indicators within a subject i are independent and have a binomial distribution. Instead, we observe that the likelihood function for the discrete-time survival model under non-informative censoring can be represented using a binomial likelihood that assumes independent event indicators.
Better developed in Singer & Willett (1993):

Taking logarithms gives the log-likelihood function:

$$l = \sum_{i=1}^{n} \left[(1 - c_i) \log_e h_{ij_i} + (1 - c_i) \sum_{j=1}^{j_i - 1} \log_e (1 - h_{ij}) + c_i \sum_{j=1}^{j_i} \log_e (1 - h_{ij}) \right].$$

Or, more simply:

$$l = \sum_{i=1}^{n} \left[(1 - c_i) \log_e \left(\frac{h_{ij_i}}{1 - h_{ij_i}} \right) + \sum_{j=1}^{j_i} \log_e (1 - h_{ij}) \right]. \tag{8}$$

Equation 8 can be modified to introduce the event-history indicators y_{ij}. If individual i is not censored ($c_i = 0$), the target event occurs in time period j_i; thus all y_{ij} equal zero except for the very last (when $j = j_i$), which equals one. If individual i is censored ($c_i = 1$), in contrast, the target event does not occur in any time period including the last (when $j = j_i$); so all the y_{ij}, including that for time period j_i, equal zero. Therefore, we can write:

$$\sum_{j=1}^{j_i} y_{ij} \log_e \left(\frac{h_{ij}}{1 - h_{ij}} \right) = \begin{cases} \log_e \left(\dfrac{h_{ij_i}}{1 - h_{ij_i}} \right) & \text{when } c_i = 0 \\ 0 & \text{when } c_i = 1 \end{cases}$$
$$= (1 - c_i) \log_e \left(\frac{h_{ij_i}}{1 - h_{ij_i}} \right). \tag{9}$$

Substituting from (9) for the first term inside the bracket in Equation 8 eliminates the censoring indicator from the log-likelihood, replacing it by the dichotomous realizations of the event-history process, the y_{ij}:

$$l = \sum_{i=1}^{n} \left[\sum_{j=1}^{j_i} y_{ij} \log_e \left(\frac{h_{ij_i}}{1 - h_{ij_i}} \right) + \sum_{j=1}^{j_i} \log_e (1 - h_{ij}) \right].$$

This can be rewritten as:

$$l = \sum_{i=1}^{n} \sum_{j=1}^{j_i} \left[\log_e \left(\frac{h_{ij_i}}{1 - h_{ij_i}} \right)^{y_{ij}} + \log_e (1 - h_{ij}) \right].$$

170

Collecting like terms and antilogging yields:

$$L = \prod_{i=1}^{n} \prod_{j=1}^{j_i} h_{ij}^{y_{ij}} (1 - h_{ij})^{(1 - y_{ij})}. \tag{10}$$

Equation 10 is the likelihood function for the discrete-time hazard process in terms of the data (the y_{ij}) and the hazard probability parameters (the h_{ij}). However, in Equation 3, we have reparameterized the h_{ij} as a logistic function of a smaller number of secondary parameters: $\alpha_1, \alpha_2, \ldots, \alpha_J, \beta_1, \beta_2, \ldots, \beta_P$. Maximizing the likelihood in (10), under the logistic reparameterization in (3), provides maximum likelihood estimates of $\alpha_1, \alpha_2, \ldots, \alpha_J, \beta_1, \beta_2, \ldots, \beta_P$ (and hence the h_{ij}). Notice that the likelihood function for the discrete-time hazard model in (10) is identical to the likelihood function for a sequence of $N = (j_1 + j_2 + \cdots + j_n)$ independent Bernoulli trials with parameters h_{ij}.

As demonstrated by Allison (1982), Brown (1975), and Laird and Oliver (1981), the equivalence of the likelihood functions of the discrete-time hazard model in (10) and the independent Bernoulli trials model allows us to treat the N dichotomous observed values y_{ij} as a collection of independent dichotomous variables with a hypothesized logistic dependence on predictors. We can regard them as the values of the outcome variable in a logistic regression analysis of the time-period indicators D and covariates Z. This provides a simple method of obtaining maximum likelihood estimates of $\alpha_1, \alpha_2, \ldots, \alpha_J, \beta_1, \beta_2, \ldots, \beta_P$ (and hence the h_{ij}) using nothing more than standard logistic regression analysis software. Because computer software for conducting logistic regression analysis is so widely available, we now illustrate the fitting of hazard models via this modified logistic regression approach, rather than via direct maximization of the likelihood in (10).

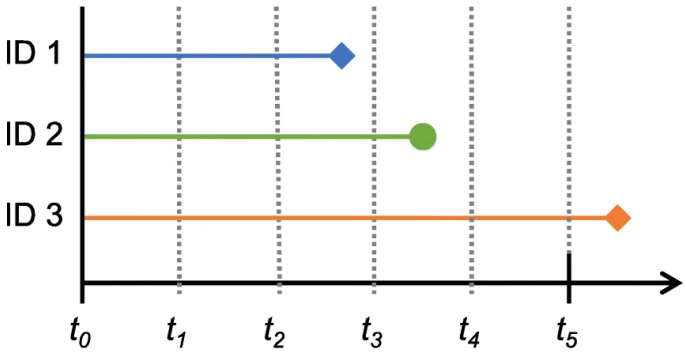
Cox (1972) proposed that since in discrete time the hazards, λ_{ij}, are probabilities, they can be parameterized to have a logistic dependence with the predictors and time intervals (the latter being sometimes dropped, or relaxed using time dependent predictors).
We assume the predictors are linearly associated with the logistic transformation of the hazard (logit-hazard) instead of with the hazard probabilities themselves. Specifically, the conditional odds of experiencing failure in each time interval $]t_{j-1}, t_j]$ (given that it has not yet occurred) is assumed to be linear function of the predictor and interval effects.

This model is sometimes referred to as the continuation ratio model, and is specified as:

$$\log\left(\frac{\lambda_{ij}}{1-\lambda_{ij}}|X_i\right) = \alpha_j + \beta X_i$$

This is roughly the result (equivalence of likelihood) and model (logit-hazard are linear) used in Shumway (2001). Having transformed the data set into "firm-year" or "person-period" data set, the likelihood for a discrete-time hazard model can be estimated using a logistic regression routine.
Subjects i contribute a row for each time interval at which they are still at risk at the start, i.e., all j such that $t_{j-1} < T$. Each record contains the subject's failure indicator for experiencing the event during that interval (i.e., their event history indicator y_{ij}), a copy of their predictor vector X_i (can also be time-dependent), and a factor variable identifying the interval A_j to which the record corresponds, see an example below:

ID	Time	Event
1	2.7	1
2	3.4	0
3	5.6	1



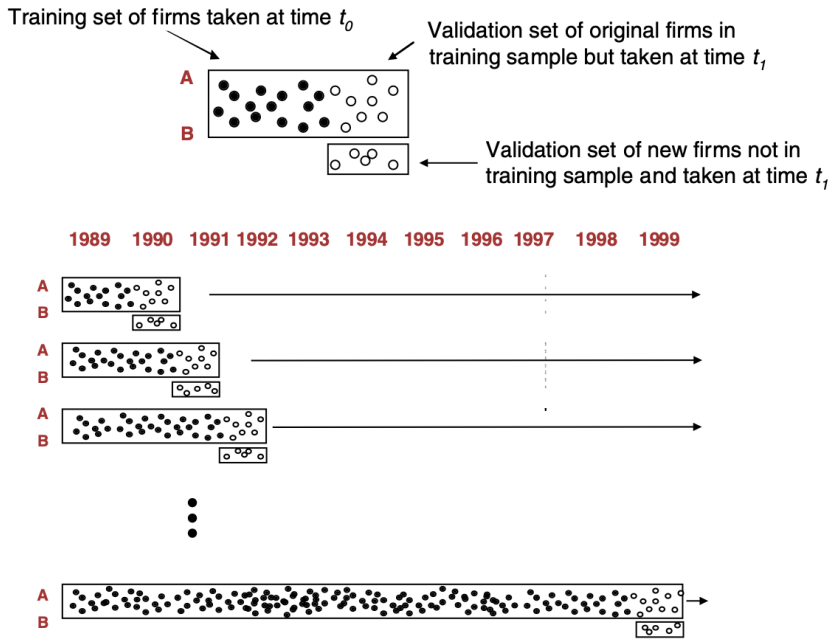
ID	Interval	Event
1	1	0
1	2	0
1	3	1
2	1	0
2	2	0
2	3	0
2	4	0
3	1	0
3	2	0
3	3	0
3	4	0
3	5	0

You will compare this dynamic approach to a simplified approach, the so-called static model in Shumway's article. In that approach, the observations are only "firm", meaning that each company is only observed once, either just before default ($Y = 1$), or at one point in time where it is in good health ($Y = 0$). This is the approach followed in Altman (1968) (using LDA) and Desbois (2008) (a firm or farm is represented once, either healthy or non healthy).

To compare between static and dynamic approaches you will use code developed in the next Task:

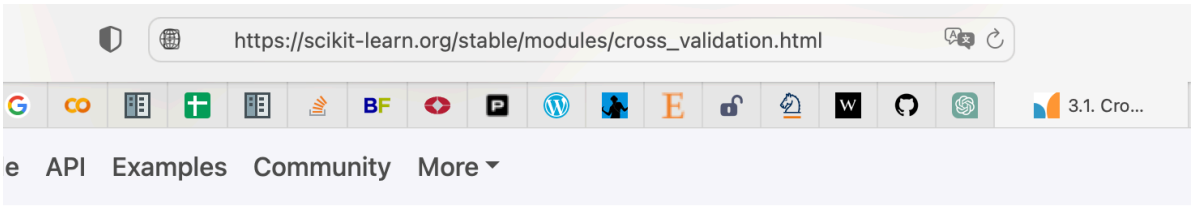
- **Task 3**: Write `R` code to assess this model and future improvements/iterations: estimation of testing error on unseen data / time series cross-validation, ROC curve analysis etc. As a first step, explain why classic K-fold Cross-Validation might not be adapted to the data set. A proposed approach is to implement something similar to the following scheme (also known as walk forward or time series split):

Exhibit 14. A Resampling Validation Approach

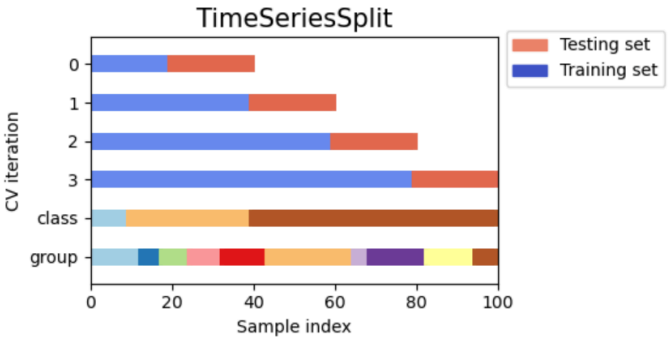


Moody's fits a model using a sample of historical data on firms and tests the model using both data on those firms one year later, and using data on new firms one year later (upper portion of exhibit). Dark circles represent in-sample data, used to estimate the model, and white circles represent testing data. We do "walk-forward testing" (bottom left) by fitting the parameters of a model using data through a particular year, and testing on data from the following year, and then inching the whole process forward one year.

which is also described here in a more generic way in the scikit-learn documentation, see below:



Here is a visualization of the cross-validation behavior.



You may adapt a little bit the length of the splits to your needs, and also your training set window may be either "fixed" (i.e. you keep all past data in training set) or "rolling" (i.e. with a fixed length or time period).

3.1 Some guidance

Rather than giving you a very detailed to do list for each of these task, I give below some guidelines in order to help you build a default/bankruptcy prediction data set which corresponds to Task 1. Once this is finished the other Tasks 2 and 3 are relatively straightforward and have been already done in the exam for Task 2 or shown in the first part of the course for Task 3 (you'll need to adapt the Cross-Validation code to perform Time-Series split instead of random K-Fold split).

For building your data set, I propose that you to use primarily (i) `Compustat/Lopucki/Moody's Annual Reports` to detect companies default/bankruptcies and (ii) `Compustat/CRSP` to build financial predictors for your model.

- Step 1: Explore `Compustat` Data(base), in particular reading the first section of this document you should be able to: get data from `comp.funda` (annual financial items) and `comp.company` (master company data) tables using either Company name (`conm`) or Identifier (`gvkey`); to merge `comp.funda`/`comp.company` if needed; to search/filter for relevant information. Examples have been given in the first section of this document.
- Step 2: Using (i) information in `comp.company` Table, especially fields `dldte` and `dlsrn` and (ii) information in `comp.funda`, especially fields `fyear` (Fiscal Year), `fyr` (Fiscal

Year-End, meaning month), `datadate` (Annual Close of Fiscal Period, a date), `fdate` (Final Date, ie date of publication of financial data, available in Compustat since 2004), start building for each observations in `comp.funda` Database a variable Y indicating if company has defaulted in the next [you have to decide] months/years. Usually the horizon retained in bankruptcy prediction models is 1 year.

Regarding fiscal year and reporting dates: APPLE INC fiscal year ends on September 30, its annual financial data is usually published in the beginning of November which is a rather short delay:

```
compustat_all %>%
  filter(grepl('^APPLE INC', conm), fyear>2017) %>%
  select(conm, gvkey, fyear, fyr, datadate, fdate)
```

```
# A tibble: 6 x 6
  conm      gvkey  fyear   fyr datadate   fdate
  <chr>     <chr>  <int> <int> <date>     <date>
1 APPLE INC 001690  2018     9 2018-09-30 2018-11-05
2 APPLE INC 001690  2019     9 2019-09-30 2019-11-04
3 APPLE INC 001690  2020     9 2020-09-30 2020-11-02
4 APPLE INC 001690  2021     9 2021-09-30 2021-11-01
5 APPLE INC 001690  2022     9 2022-09-30 2022-10-31
6 APPLE INC 001690  2023     9 2023-09-30 2023-11-04
```

If APPLE INC were to go bankrupt in October 2024, the last available financial data would be pertaining to fiscal year 2023, ending on 30 October 2023, reported on 4 November 2023.

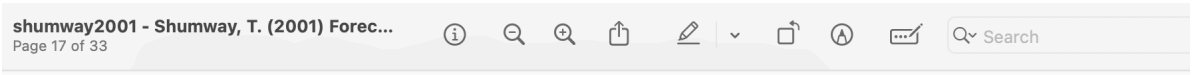
Looking, when the `fdate` field is available, at the average delay between `datadate` (Annual Close of Fiscal Period) and `fdate` (date of publication of financial data), you can devise a reasonable rule of thumb, say 180 days / 6 months to be on the safe side:

```
reporting <- compustat_all %>%
  select(gvkey, fyear, fyr, datadate, fdate, final) %>%
  filter(!is.na(fdate)) %>%
  mutate(diff = fdate-datadate) %>%
  group_by(fyear) %>%
  summarize(n = n_distinct(gvkey),
            m=round(mean(diff),0))

reporting %>% print(n=30)
```

```
# A tibble: 18 x 3
   fyear     n m
   <int> <int> <drtn>
 1  2006     6 102 days
 2  2007     7 285 days
 3  2008     7 240 days
 4  2009     7 154 days
 5  2010     7  91 days
 6  2011     7  83 days
 7  2012     7 137 days
 8  2013     7  81 days
 9  2014     7  56 days
10  2015     7 125 days
11  2016     7 351 days
12  2017     7 306 days
13  2018     7 257 days
14  2019     7 196 days
15  2020     6 172 days
16  2021     5 110 days
17  2022     5  68 days
18  2023     4  48 days
```

For example in Shumway (2001), the author proposes retaining a more conservative 6-month lag for Financial data, starting from the beginning of calendar year in which the bankruptcy occurs, you can use this approach to label your "firm year" observations with 0 or 1:

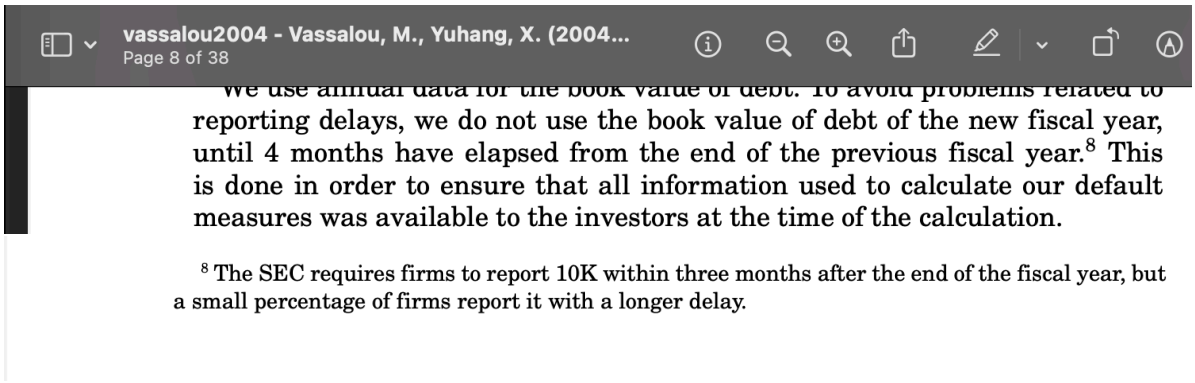


> In order to make my forecasting exercise realistic, I lag all data to ensure that the data are observable in the beginning of the year in which bankruptcy is observed. To construct Altman's (and Zmijewski's) variables, I lag COMPUSTAT data to ensure that each firm's fiscal year ends at least six months before the beginning of the year of interest. I lag the market-driven variables described below in a similar fashion.

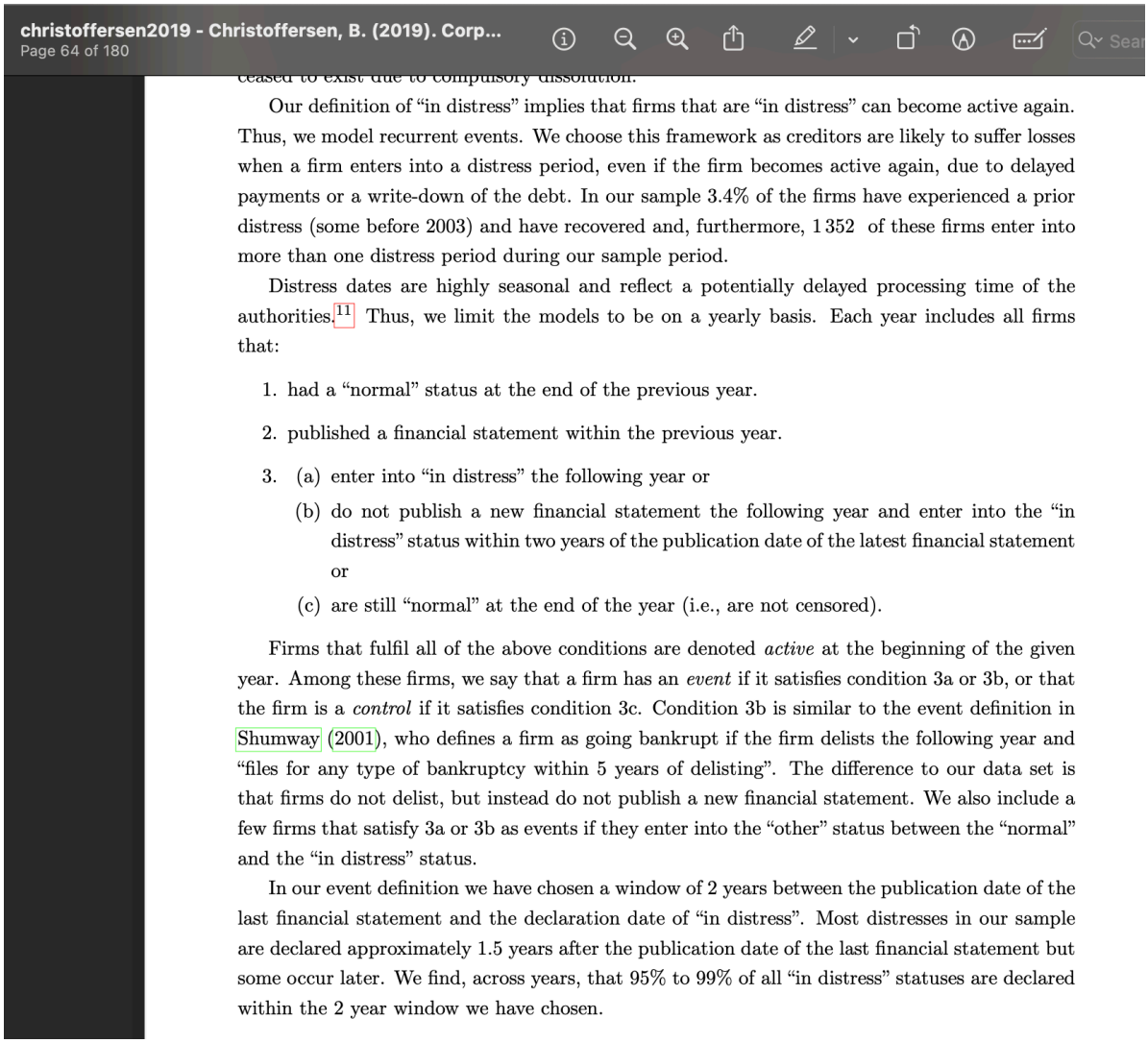
The approach is similar in Christoffersen (2019):

> We use CRSP and Compustat for market data and financial statements, respectively. We lag data from Compustat by 3 months to reflect the typical delay on financial statements, use quarterly data with annualized flow variables when available and otherwise we use yearly data. Data from CRSP is lagged by 1 month to reflect that we only know past market data. Summary

In Vassalou (2004), authors use a 4-month lag, referring to a SEC constraint on firms report publication:



In Christoffersen (2019), the event of interest definition is detailed as followed:



Then you have the choice to censor the firm year observations following an event of default. It can also be interesting to keep the following firm year information. For example looking at Eastman Kodak which filed bankruptcy in January 2012, then emerged from bankruptcy in March 2013, Compustat never stopped reporting Financial results:

```
compustat_all %>%
    filter(grepl('KODAK', conm), fyear>=2009, fyear<=2014) %>%
    select(conm, gvkey, fyear, fyr, datadate, fdate)
```

```
# A tibble: 6 x 6
  conm             gvkey  fyear   fyr datadate   fdate
```

```
  <chr>            <chr>  <int> <int> <date>     <date>
1 EASTMAN KODAK CO 004194  2009    12 2009-12-31 2010-02-25
2 EASTMAN KODAK CO 004194  2010    12 2010-12-31 2011-03-15
3 EASTMAN KODAK CO 004194  2011    12 2011-12-31 2012-03-27
4 EASTMAN KODAK CO 004194  2012    12 2012-12-31 2013-04-08
5 EASTMAN KODAK CO 004194  2013    12 2013-12-31 2014-04-04
6 EASTMAN KODAK CO 004194  2014    12 2014-12-31 2015-03-22
```

and the bankruptcy information is not available in Table `comp.company`:

```
company_all %>%
    filter(gvkey=='004194') %>%
    select(conm, gvkey, dlrsn, dldte)
```

```
# A tibble: 1 x 4
  conm             gvkey  dlrsn dldte
  <chr>            <chr>  <chr> <date>
1 EASTMAN KODAK CO 004194 <NA>  NA
```

Bankruptcy is reported both in Moody's (the report for 2012 didn't indicate precise dates for default events so end of year was attributed as a fallback):

```
dat_default_moodys_annual %>% filter(grepl('Eastman', company_name))
```

```
# A tibble: 1 x 5
  company_name          country       default_type default_date compact_conm
  <chr>                 <chr>         <chr>        <date>       <chr>
1 Eastman Kodak Company United States Chapter 11   2012-12-31   EASTMANKODAK
```

and LoPucki, also showing the date of emerging from bankruptcy:

```
lopucki %>%
  filter(GvkeyBefore == '004194') %>%
  select(NameCorp, GvkeyEmerging, DateFiled, DateEmerging)
```

```
# A tibble: 1 x 4
  NameCorp              GvkeyEmerging DateFiled           DateEmerging
  <chr>                 <chr>         <dttm>              <dttm>
1 Eastman Kodak Company 004194        2012-01-19 00:00:00 2013-08-23 00:00:00
```

Regarding the bankruptcy event, on 19 January 2012, only Financials pertaining to fiscal year 2010 ending in December were available for the analyst published on 15 March 2011, so the firm year observation `gvkey=004194` / `fyear=2010` could be labeled as 1.

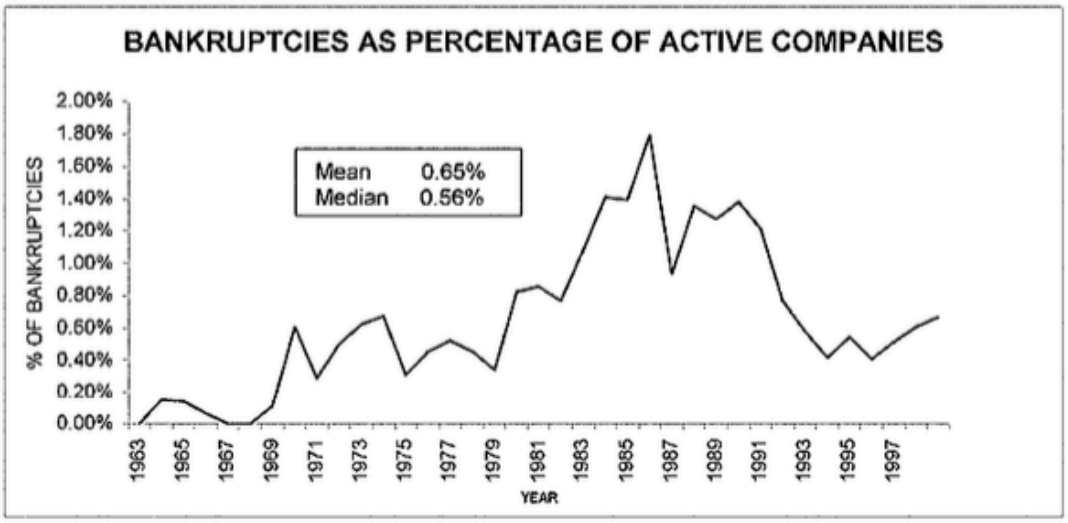
To fully use data at hand, one could furthermore consider that EASTMAN KODAK after 2014 is a fresh company and use its firm year observations after that date labeled with 0 until any new event of default.

- Step 3: Performing Data quality check for both Y and X is always a good practice

Regarding Y:

- checking default for usual suspects, for example largest bankruptcies in the US (excluding banks)
- comparing the number of obtained bankruptcies by year with another source, for example Chava & Jarrow (2004):

PANEL A: BANKRUPTCIES AS PERCENTAGE OF ACTIVE FIRMS DURING 1962-1999



PANEL B: BANKRUPTCIES AS PERCENTAGE OF ACTIVE FIRMS

The following table gives the number of bankruptcies, total number of active firms and the percentage of bankruptcies each year during the sample period of 1962-1999.

YEAR	# of bankrupt firms	# of active firms	% of bankrupt to active firms	YEAR	# of bankrupt firms	# of active firms	% of bankrupt to active firms
1963	0	1294	0.00%	1981	35	4090	0.86%
1964	2	1362	0.15%	1982	36	4699	0.77%
1965	2	1444	0.14%	1983	50	4645	1.08%
1966	1	1536	0.07%	1984	72	5112	1.41%
1967	0	1612	0.00%	1985	73	5240	1.39%
1968	0	1721	0.00%	1986	94	5250	1.79%
1969	2	1842	0.11%	1987	52	5584	0.93%
1970	12	1988	0.60%	1988	80	5911	1.35%
1971	6	2089	0.29%	1989	74	5814	1.27%
1972	11	2208	0.50%	1990	79	5726	1.38%
1973	27	4355	0.62%	1991	69	5689	1.21%
1974	29	4310	0.67%	1992	45	5874	0.77%
1975	13	4215	0.31%	1993	36	6246	0.58%
1976	19	4197	0.45%	1994	29	6971	0.42%
1977	22	4229	0.52%	1995	40	7387	0.54%
1978	19	4182	0.45%	1996	31	7620	0.41%
1979	14	4110	0.34%	1997	41	8023	0.51%
1980	33	4017	0.82%	1998	49	8079	0.61%

Figure 4. Bankruptcies as % of firms listed on NYSE–AMEX–NASDAQ. The above figure

Regarding X:

- checking the consistency of Compustat Financials, at least at the macro level, they should be balanced and articulate. For example Casey (2016) authors build rules to recover some aggregated Financial items from Compustat which are sometimes reported as NA, while they can directly be reconstituted from other items. It allows to increase sample size and improve data quality.

For example we give below some rather easily implementable rules (variable names match database/`R` data files) taken from the Appendix of Casey (2016) to check Income Statement (retrieving Net Income form other items) and Balance Sheet (Total Assets = Total Liabilites + Equity):

```
# **Table 1: Top-Level COMPUSTAT Derivation of the Income Statement**
#
#   **Income statement**
#
#
#   Sales = sale
#
#   Costs of goods sold = cogs
#
#   Sales, general & administrative expenses (SG&A)= xsga
#
#   EBITDA = sale - cogs - xsga (  ebitda)
#
#   Depreciation & amortization expense = dp
#
#   EBIT = EBITDA - dp (  oiadp)
#
#   Non-operating income = nopi
#
#   Special items = spi
#
#   Interest expense = xint
#
#   Pretax income = EBIT + nopi + spi - xint (  pi)
#
#   Tax expense = txt
#
#   Operating income before extraordinary items
#   before minority interest = Pretax income - txt (  ibmii )
#
```

```
#   Minority interest = mii
#
#   Operating income before extraordinary items
#   after minority interest = Pretax income - txt - mii (  ib)
#
#   Extraordinary items & Discontinued operations = xido
#
#   Net income = Operating income before extra. items and after min. interest +
#     extraordinary items & discontinued operations (  ni)
#
#   check: Net income = ni

# **Table 4: Top-Level COMPUSTAT Derivation of the Balance Sheet**
#
#   **Assets**
#
#   Cash & Equivalents = che
#
# Receivables = rect
#
# Inventories = invt
#
# Other current assets = aco
#
# Current assets total = cash & cash equivalents + receivables + inventories +
#   other current assets (  act)
#
# Net plant property & equipment (PP&E) = ppent
#
# Investments at equity = ivaeq
#
# Investments other = ivao
#
# Intangibles = intan
#
# Assets other = ao
#
# Total assets = current assets total + plant property & equipment +
#   investments at equity + investments other + intangibles + assets other (  at)
#
# **Liabilities**
#
```

```
#   Accounts payable = ap
#
# Taxes payable = txp
#
# Debt due in one year = dlc
#
# Other current liabilities = lco
#
# Total current liabilities = accounts payable + taxes payable +
#   debt due in one year + other current liabilities (  lct)
#
# Long term debt = dltt
#
# Deferred taxes = txditc
#
# Liabilities other = lo
#
# Non-controlling interest = mib + mibn
#
# Total liabilities = total current liabilities + long term debt + deferred taxes +
#   liabilities other (  lt) + Non-controlling interest
#
# **Shareholders' Equity**
#
#   Equity = seq
#
# Total Assets = Total Liabilities + Equity
```

- Step 4: Preparing Practical 2 which uses Distance-to-Default (market based) as a predictor by linking Compustat to CRSP as described in the first section of this document.
- Step 5: Feature Engineering, computing financial ratios from Fundamentals

Your task is to create additional predictors by computing financial ratios from Compustat Fundamentals data. This step prepares Practical 3.

Besides Altman or Desbois (Desbois (2008)) financial ratios presented in the first lesson ($r_1, ..., r_{37}$), Compustat provides 70 pre-calculated financial ratios for all U.S. companies using a SAS routine, the routine has been translated to R and is available here for inspiration. In particular some ratios look at the evolution of Fundamentals between two or more reporting periods or fiscal years.

Allison, P. D. (1982). Discrete-time methods for the analysis of event histories. *Sociological Methodology*, *13*, 61–98. https://doi.org/10.2307/270718

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance*, *23*(4), 589–609. http://www.jstor.org/stable/2978933

Casey, al, R. J. (2016). Do compustat financial statement data articulate? *Journal of Financial Reporting*, *1*(1), 37–59. https://doi.org/10.2308/jfir-51329

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]*.

Desbois, D. (2008). Introduction to scoring methods: financial problems of farm holdings. *Case Studies in Business, Industry and Government Statistics*, *2*(1), 56–76. https://hal.science/hal-01172847

Duffie, S., D. (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

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

Singer, J. D., & Willett, J. B. (1993). It's about time: Using discrete-time survival analysis to study duration and the timing of events. *Journal of Educational Statistics*, *18*(2), 155–195.

Suresh, K., Severn, C., & Ghosh, D. (2022). Survival prediction models: An introduction to discrete-time modeling. *BMC Medical Research Methodology*, *22*(1). https://doi.org/10.1186/s12874-022-01679-6

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