Pharma Practice

Create DM dataset

This practice project is an exploration of pharmaceutical data analysis techniques using sample datasets obtained from the SASCRUNCH TRAINING at https://www.sascrunchtraining.com/clinical-project-1.html. The project focuses on applying Clinical Data Interchange Standards Consortium (CDISC) Study Data Tabulation Model (SDTM) standards to real-world-like clinical trial data, with the goal of creating Analysis Data Model (ADaM) datasets and generating statistical reports typical of those used in the pharmaceutical industry. This hands-on exercise serves as a foundation for developing practical skills in statistical programming within the pharma sector.

Download and unzip data.

```
if (length(list.files("data/")) == 0) system("./code/get_data.sh")
```

Load packages.

```
library(tidyverse)
library(metacore)
library(metatools)
library(pharmaversesdtm)
library(admiral)
library(xportr)
library(readxl)
library(xml2)
```

Import data

```
death <- read_excel("data/Project_1_SDTM_DM/DEATH.xlsx")
dm <- read_excel("data/Project_1_SDTM_DM/DM.xlsx")
ds <- read_excel("data/Project_1_SDTM_DM/DS.xlsx")
ex <- read_excel("data/Project_1_SDTM_DM/EX.xlsx")
spcpkb <- read_excel("data/Project_1_SDTM_DM/SPCPKB1.xlsx")

cdm_vars <- read_excel("data/cdm_variables.xlsx") # variable descriptions
dm_only <- read_excel("data/dm_only.xlsx") # this is the SDTM specification
```

Let's build a SDTM DM dataset based on the dm_only specification.

```
DTHDTC <-
  dm %>%
  left_join(death %>% filter(!is.na(DTHCAUSE))) %>%
  pull(DTH_DAT)
```

```
Joining with `by = join_by(STUDY, CENTRE, SUBJECT, PATIENT)`
```

```
sdtm_dm <-
  dm %>%
  transmute(
    STUDYID = "XYZ",
    DOMAIN = "DM",
    USUBJID = paste(STUDYID, SUBJECT, sep = "/"),
    SUBJID = SUBJECT,
    RFSTDTC = if_else(
      spcpkb$PSCHDAY == 1 & spcpkb$PART == "A",
      paste0(spcpkb$IPFD1DAT, spcpkb$IPFD1TIM),
      NA_character_
    ) %>% if_else(. == "NANA", NA, .) %>%
      na.omit() %>%
      pluck(1),
    RFENDTC = if_else(
      spcpkb$PSCHDAY == 1 & spcpkb$PART == "A",
      paste0(spcpkb$IPFD1DAT, spcpkb$IPFD1TIM),
      NA_character_
    ) %>% if_else(. == "NANA", NA, .) %>%
      na.omit() %>%
      pluck(-1),
    RFXSTDTC = "some date",
    RFXENDTC = "some other date",
    RFPENDTC = c(RFXSTDTC[1], ds$DSSTDAT, RFXENDTC[1]),
    DTHDTC = DTHDTC,
    DTHFL = if_else(is.na(DTHDTC), DTHDTC, "Y"),
    SITEID = dm$CENTRE,
    BRTHDTC = dm$BRTHDAT,
    AGE = dm$AGE,
    AGEU = if_else(dm$AGEU == "C29848", "YEARS", dm$AGEU),
    SEX = case_when(
      dm$SEX == "C20197" ~ "M",
      dm$SEX == "C16576" ~ "F",
      TRUE ~ "U"
    ),
    RACE = if_else(dm$RACE == "C41260", "ASIAN", "WHITE"),
    ETHNIC = if_else(dm$ETHNIC == "C41222", "NOT HISPANIC OR LATINO", dm$ETHNIC),
    ARMCD = if_else(!is.na(RFSTDTC), "A01-A02-A03", "NOTASSGN"),
    # I don't get it. I don't think I have access to CDM.IE.IEYN
```

```
    ARM = if_else(ARMCD == "NOTASSIGN", "Not Assigned", ARMCD),
    # Again, I don't know what SDTM.TA is...
    ACTARMCD = ARMCD,
    ACTARM = ARM,
    COUNTRY = "See <DM_Details> tab",
    DMDTC = dm$VIS_DAT,
    CENTRE = dm$CENTRE,
    PART = dm$PART,
    RACEOTH = str_to_upper(dm$RACEOTH),
    VISITDTC = dm$VIS_DAT
  )
```

Geez... I see the benefits of automating this process with the metacore and metatools packages. :) Also, I could not figure out how to create some of the variables because as far as I can tell, I don't have the necessary data but we're just learning.

All 28 variables from the `dm_only` have been created.

```
sdtm_dm
```

```
# A tibble: 21 x 28
   STUDYID DOMAIN USUBJID SUBJID RFSTDTC      RFENDTC RFXSTDTC RFXENDTC RFPENDTC
   <chr>   <chr>  <chr>    <dbl> <chr>        <chr>   <chr>    <chr>    <chr>
 1 XYZ     DM     XYZ/101    101 2017-05-161~ 2018-0~ some da~ some ot~ some da~
 2 XYZ     DM     XYZ/102    102 2017-05-161~ 2018-0~ some da~ some ot~ 2017-08~
 3 XYZ     DM     XYZ/103    103 2017-05-161~ 2018-0~ some da~ some ot~ 2018-06~
 4 XYZ     DM     XYZ/104    104 2017-05-161~ 2018-0~ some da~ some ot~ 2017-10~
 5 XYZ     DM     XYZ/105    105 2017-05-161~ 2018-0~ some da~ some ot~ 2018-06~
 6 XYZ     DM     XYZ/106    106 2017-05-161~ 2018-0~ some da~ some ot~ 2018-06~
 7 XYZ     DM     XYZ/107    107 2017-05-161~ 2018-0~ some da~ some ot~ 2017-10~
 8 XYZ     DM     XYZ/108    108 2017-05-161~ 2018-0~ some da~ some ot~ 2018-02~
 9 XYZ     DM     XYZ/109    109 2017-05-161~ 2018-0~ some da~ some ot~ 2018-03~
10 XYZ     DM     XYZ/110    110 2017-05-161~ 2018-0~ some da~ some ot~ 2017-05~
# i 11 more rows
# i 19 more variables: DTHDTC <chr>, DTHFL <chr>, SITEID <dbl>, BRTHDTC <chr>,
#   AGE <dbl>, AGEU <chr>, SEX <chr>, RACE <chr>, ETHNIC <chr>, ARMCD <chr>,
#   ARM <chr>, ACTARMCD <chr>, ACTARM <chr>, COUNTRY <chr>, DMDTC <chr>,
#   CENTRE <dbl>, PART <chr>, RACEOTH <chr>, VISITDTC <chr>
```

Create ADSL dataset

I'll create an ADaM subject-level dataset using pharmaverse example data.

ADSL stands for Analysis Data Subject-Level Dataset. It's part of the ADaM (Analysis Data Model) standards provided by CDISC for use in statistical analysis related to clinical trials. The ADSL dataset contains

one record per subject and includes key variables necessary for analysis, such as demographic information, treatment information, and other subject-level data. It's the foundational dataset used in many statistical analyses and is often required for regulatory submissions to agencies like the FDA.

Create metacore object.

```
doc <- read_xml(metacore_example("SDTM_define.xml"))
xml_ns_strip(doc)
```

```
ds_spec2 <- xml_to_ds_spec(doc)
ds_vars <- xml_to_ds_vars(doc)
var_spec <- xml_to_var_spec(doc)
value_spec <- xml_to_value_spec(doc)
code_list <- xml_to_codelist(doc)
derivations <- xml_to_derivations(doc)
```

```
meta_obj <- metacore(ds_spec2, ds_vars, var_spec, value_spec, derivations, code_list)
```

```
Warning: core from the ds_vars table only contain missing values.

supp_flag from the ds_vars table only contain missing values.

common from the var_spec table only contain missing values.

The following words in value_spec$origin are not allowed:
    edt


dataset from the supp table only contain missing values.

variable from the supp table only contain missing values.

idvar from the supp table only contain missing values.

qeval from the supp table only contain missing values.

Warning: The following derivations are never used:
 MT.SUPPAE.QVAL: see value level metadata
 MT.SUPPDM.QVAL: see value level metadata

Warning: The following codelist(s) are never used:
 DRUG DICTIONARY
 MEDICAL HISTORY DICTIONARY


 Metadata successfully imported
```

```r
meta_dm <- meta_obj %>%
  select_dataset("DM", simplify = TRUE)
```

References