Creating a dataframe from SAS datasets

Table of Contents

## Objectives

After completing this unit, users should be able to:

* 1. know to to list, access and load R Studio’s built-in datasets
  2. access an internal location on their machine containing data and import it into an R data frame
  3. access an external location containing data and import it into an R data frame
  4. identify the key characteristics of a r dataframe , and r tibble and ways in which these differ from SAS datasets

#### R Studio is designed to read data from the local file system or from an accessible external file system.

#### tidyverse, developed by R Studio, includes the HAVEN package to read SAS/SPSS/Stata files. This incorporates earlier work from the readr package.

#### see also: <https://tutorials.methodsconsultants.com/posts/reading-sas-spss-or-stata-files-into-r-using-haven/>

## Internal data locations

* Built-ins: There are several built-in internal data frames made available as part of the R Studio implementation. These are used for training purposes in the R Studio documentation and represent the universe of data that R users have contributed over the years. These built-ins provide a useful introduction to various data ranging from format categories to data frames. depending on the version of R.

By default, these are written to a default location on the users’ C: drive, during initial setup of RStudio, e.g. “C:/Program Files/R/R-4.0.4/library” .

# To see the list of pre-loaded data, type the function data() at the R Studio console:  
  
data()  
  
## Or do it this way  
data(package = .packages(all.available = TRUE))  
# Loading a built-in R data.  
  
# Loading  
data(mtcars)  
  
# Print the first 6 rows  
head(mtcars, 6)

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 21.0 6 160 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160 110 3.90 2.875 17.02 0 1 4 4  
## Datsun 710 22.8 4 108 93 3.85 2.320 18.61 1 1 4 1  
## Hornet 4 Drive 21.4 6 258 110 3.08 3.215 19.44 1 0 3 1  
## Hornet Sportabout 18.7 8 360 175 3.15 3.440 17.02 0 0 3 2  
## Valiant 18.1 6 225 105 2.76 3.460 20.22 1 0 3 1

* Local project data: R Studio can be set up to work with ‘projects.’ These projects are typically stored somewhere on the user’s C: drive , but can be configured to an accessible location external to the user’s machine. For example, I have copied several \*.csv files into ‘data’ contained in a project I set up with R studio. This is the path on my machine: ‘C:.’ A utility named “here” can point to this default location for the project and saves some typing.

library(tidyverse)  
#the here package to tell R where your data is,  
#install.packages("here")  
library(here)  
  
## read in a \*.csv file into a tibble with large/medium/small columns for each of 7 items  
frames\_wide<-read\_csv(here("data/csv","frames\_wide.csv"))  
head(frames\_wide)

## # A tibble: 6 x 25  
## id gender age condition large\_item1 large\_item2 large\_item3 large\_item4  
## <dbl> <chr> <dbl> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 1 male 36 category 8 7 6 6  
## 2 2 male 46 category 9 9 8 8  
## 3 3 female 33 property 8 9 9 2  
## 4 4 female 71 property 9 8 3 2  
## 5 5 female 23 property 9 9 9 9  
## 6 6 female 31 category 9 9 8 8  
## # ... with 17 more variables: large\_item5 <dbl>, large\_item6 <dbl>,  
## # large\_item7 <dbl>, medium\_item1 <dbl>, medium\_item2 <dbl>,  
## # medium\_item3 <dbl>, medium\_item4 <dbl>, medium\_item5 <dbl>,  
## # medium\_item6 <dbl>, medium\_item7 <dbl>, small\_item1 <dbl>,  
## # small\_item2 <dbl>, small\_item3 <dbl>, small\_item4 <dbl>, small\_item5 <dbl>,  
## # small\_item6 <dbl>, small\_item7 <dbl>

* In a similar fashion, the sas subfolder I created contains \*.xpt (sas Transport) files. The haven package contained in the ‘tidyverse’ can read these into a tibble (a.k.a ‘data frame’).

# Read in a sas \*.xpt dataset.   
## Currently accessing \*.xpt can be done only by a script:  
  
library(haven)  
 ae\_xpt<-read\_xpt(here("data/sas","ae.xpt"))  
head(ae\_xpt)

## # A tibble: 6 x 35  
## STUDYID DOMAIN USUBJID AESEQ AESPID AETERM AELLT AELLTCD AEDECOD AEPTCD AEHLT  
## <chr> <chr> <chr> <dbl> <chr> <chr> <chr> <dbl> <chr> <dbl> <chr>  
## 1 CDISCPI~ AE 01-701~ 1 E07 APPLI~ APPL~ NA APPLIC~ NA HLT\_~  
## 2 CDISCPI~ AE 01-701~ 2 E08 APPLI~ APPL~ NA APPLIC~ NA HLT\_~  
## 3 CDISCPI~ AE 01-701~ 3 E06 DIARR~ DIAR~ NA DIARRH~ NA HLT\_~  
## 4 CDISCPI~ AE 01-701~ 3 E10 ATRIO~ AV B~ NA ATRIOV~ NA HLT\_~  
## 5 CDISCPI~ AE 01-701~ 1 E08 ERYTH~ ERYT~ NA ERYTHE~ NA HLT\_~  
## 6 CDISCPI~ AE 01-701~ 2 E09 ERYTH~ LOCA~ NA ERYTHE~ NA HLT\_~  
## # ... with 24 more variables: AEHLTCD <dbl>, AEHLGT <chr>, AEHLGTCD <dbl>,  
## # AEBODSYS <chr>, AEBDSYCD <dbl>, AESOC <chr>, AESOCCD <dbl>, AESEV <chr>,  
## # AESER <chr>, AEACN <chr>, AEREL <chr>, AEOUT <chr>, AESCAN <chr>,  
## # AESCONG <chr>, AESDISAB <chr>, AESDTH <chr>, AESHOSP <chr>, AESLIFE <chr>,  
## # AESOD <chr>, AEDTC <chr>, AESTDTC <chr>, AEENDTC <chr>, AESTDY <dbl>,  
## # AEENDY <dbl>

## from Navitas' SharePoint location:   
## OneDrive setup C:\Users\thomasb\Navitas Life Sciences\NDS R Training - Documents\General\Data  
## note escaped '\' characters.  
cm\_xpt<-read\_xpt(here("data/sas","cm.xpt"))  
head(cm\_xpt)

## # A tibble: 6 x 21  
## STUDYID DOMAIN USUBJID CMSEQ CMSPID CMTRT CMDECOD CMINDC CMCLAS CMDOSE CMDOSU  
## <chr> <chr> <chr> <dbl> <chr> <chr> <chr> <chr> <chr> <dbl> <chr>   
## 1 CDISCPI~ CM 01-701~ 1 1 ASPI~ ACETYL~ "" NERVO~ 1 TABLET  
## 2 CDISCPI~ CM 01-701~ 5 1 ASPI~ ACETYL~ "" NERVO~ 1 TABLET  
## 3 CDISCPI~ CM 01-701~ 9 1 ASPI~ ACETYL~ "" NERVO~ 1 TABLET  
## 4 CDISCPI~ CM 01-701~ 13 1 ASPI~ ACETYL~ "" NERVO~ 1 TABLET  
## 5 CDISCPI~ CM 01-701~ 18 1 ASPI~ ACETYL~ "" NERVO~ 1 TABLET  
## 6 CDISCPI~ CM 01-701~ 23 1 ASPI~ ACETYL~ "" NERVO~ 1 TABLET  
## # ... with 10 more variables: CMDOSFRQ <chr>, CMROUTE <chr>, VISITNUM <dbl>,  
## # VISIT <chr>, VISITDY <dbl>, CMDTC <chr>, CMSTDTC <chr>, CMENDTC <chr>,  
## # CMSTDY <dbl>, CMENDY <dbl>

To load SAS *.sas7bdat datasets into R: -* .sas7bdat files can be imported from the file Menu in R Studio - or you can use datasetName<-haven::read\_sas(data, path) in a script

## reading in a \*.sas7bdat is possible either way:  
library(haven)  
skinproduct\_vfdemo <- read\_sas("data/sas/skinproduct\_vfdemo.sas7bdat", NULL)  
head(skinproduct\_vfdemo)

## # A tibble: 6 x 5  
## ProductKey DistributionCenter DATE\_CHAR Discount Revenue  
## <chr> <chr> <chr> <dbl> <dbl>  
## 1 Alex's Anti Dandruff Shampoo 17~ Atlanta 02/10/20~ 0 35966  
## 2 Alex's Anti Dandruff Shampoo 17~ Atlanta 02/17/20~ 0 25270  
## 3 Alex's Anti Dandruff Shampoo 17~ Atlanta 02/24/20~ 0 20980  
## 4 Alex's Anti Dandruff Shampoo 17~ Atlanta 03/03/20~ 0 86823  
## 5 Alex's Anti Dandruff Shampoo 17~ Atlanta 03/10/20~ 0 21888  
## 6 Alex's Anti Dandruff Shampoo 17~ Atlanta 03/17/20~ 0 21517

## Attaching, finding, summarizing

* + attach() utility reduces the need to use a$b referencing

## Some useful utilities for looking at SAS data in R

require(utils)  
  
attach (cm\_xpt)  
  
## Janitor package useful for character data  
## NOT run install.packages("janitor")  
library(janitor)  
tabyl( CMDECOD,show\_na=TRUE)

## CMDECOD n percent  
## ACETYLSALICYLIC ACID 380 0.0505992011  
## ALGELDRATE 36 0.0047936085  
## ALPRAZOLAM 2 0.0002663116  
## AMLODIPINE 112 0.0149134487  
## BUDESONIDE 3 0.0003994674  
## CALCIUM 180 0.0239680426  
## CALCIUM CARBONATE 14 0.0018641811  
## CIMETIDINE 1 0.0001331558  
## CLOBETASOL PROPIONATE 1 0.0001331558  
## DIGOXIN 39 0.0051930759  
## DILTIAZEM HYDROCHLORIDE 1 0.0001331558  
## DONEPEZIL HYDROCHLORIDE 10 0.0013315579  
## DOXAZOSIN MESILATE 32 0.0042609854  
## ESTROGENS CONJUGATED 195 0.0259653795  
## FELODIPINE 1 0.0001331558  
## FERROUS SULFATE 13 0.0017310253  
## FLUVASTATIN 21 0.0027962716  
## FUROSEMIDE 48 0.0063914780  
## GUAIFENESIN 2 0.0002663116  
## HALOPERIDOL 1 0.0001331558  
## HYDROCORTISONE 105 0.0139813582  
## IPRATROPIUM BROMIDE 12 0.0015978695  
## LEUPRORELIN ACETATE 17 0.0022636485  
## LOPERAMIDE HYDROCHLORIDE 14 0.0018641811  
## LOSARTAN POTASSIUM 6 0.0007989348  
## METFORMIN HYDROCHLORIDE 23 0.0030625832  
## NAPROXEN SODIUM 46 0.0061251664  
## NIFEDIPINE 9 0.0011984021  
## NIZATIDINE 53 0.0070572570  
## PAROXETINE HYDROCHLORIDE 1 0.0001331558  
## SALBUTAMOL SULFATE 21 0.0027962716  
## SIMETICONE 13 0.0017310253  
## SUMATRIPTAN 13 0.0017310253  
## UNCODED 6085 0.8102529960

## turn off the attachment  
detach(cm\_xpt)

require(utils)  
  
summary(women$height) # refers to variable 'height' in the data frame

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 58.0 61.5 65.0 65.0 68.5 72.0

attach(women)  
summary(height) # The same variable now available by name

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 58.0 61.5 65.0 65.0 68.5 72.0

height <- height\*2.54 # Don't do this. It creates a new variable  
 # in the user's workspace  
find("height")

## [1] ".GlobalEnv" "women"

summary(height) # The new variable in the workspace

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 147.3 156.2 165.1 165.1 174.0 182.9

rm(height)  
summary(height) # The original variable.

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 58.0 61.5 65.0 65.0 68.5 72.0

height <<- height\*25.4 # Change the copy in the attached environment  
find("height")

## [1] "women"

summary(height) # The changed copy

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1473 1562 1651 1651 1740 1829

detach("women")  
summary(women$height) # unchanged

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 58.0 61.5 65.0 65.0 68.5 72.0