Merging

1/31/2022

Here we have three data datasets and we merge the second and third into the first. Note that the first dataset has *two* sets of IDs (name and code). One of these is used to help merge with `df_2` and the other with `df_3`.

Basic merging

```
df_1 <- data.frame(name = LETTERS[1:4], code = 1:4, X = runif(4))

df_2 <- data.frame(name = c("A", "B", "D"), var_2 = runif(3))

df_3 <- data.frame(code = 1:3, var_3= c("a", "v", "w"))

df <- df_1 %>% left_join(df_2) %>% left_join(df_3)
```

```
## Joining, by = "name"
## Joining, by = "code"
```

```
df %>% kable()
```

name	code	X	var_2	var_3
A	1	0.8815352	0.5811779	a
B	2	0.8545843	0.1655006	v
C	3	0.7624539	NA	w
D	4	0.9734170	0.7080061	NA

WDI

Here is an example in which we extract a variable from WDI and then merge in region information.

```
library(WDI)


# GDP data
df <- WDI::WDI(start = 1992, end = 1993)

head(df)
```

```
##   iso2c                    country NY.GDP.PCAP.KD year
## 1    ZH Africa Eastern and Southern       1105.286 1993
## 2    ZH Africa Eastern and Southern       1151.733 1992
## 3    ZI  Africa Western and Central       1190.951 1993
## 4    ZI  Africa Western and Central       1237.023 1992
```

```
## 5    1A                  Arab World       4191.887 1993
## 6    1A                  Arab World       4176.685 1992
```

```
# Lets add in region and iso3c

# We make a codes dataset
country_codes <- WDI::WDI_data$country %>% data.frame() %>%
  dplyr::select(iso2c,  iso3c, region)

head(country_codes)
```

```
##   iso2c iso3c                    region
## 1    AW   ABW Latin America & Caribbean
## 2    AF   AFG                South Asia
## 3    A9   AFR                Aggregates
## 4    AO   AGO        Sub-Saharan Africa
## 5    AL   ALB     Europe & Central Asia
## 6    AD   AND     Europe & Central Asia
```

```
# Merge in
df <- df %>% left_join(country_codes)
```

```
## Joining, by = "iso2c"
```

```
head(df)
```

```
##   iso2c                     country NY.GDP.PCAP.KD year iso3c     region
## 1    ZH Africa Eastern and Southern       1105.286 1993  <NA>       <NA>
## 2    ZH Africa Eastern and Southern       1151.733 1992  <NA>       <NA>
## 3    ZI  Africa Western and Central       1190.951 1993  <NA>       <NA>
## 4    ZI  Africa Western and Central       1237.023 1992  <NA>       <NA>
## 5    1A                  Arab World       4191.887 1993   ARB Aggregates
## 6    1A                  Arab World       4176.685 1992   ARB Aggregates
```

```
# Lets get rid of the aggregates because Africa is not a country
df <- df %>% filter(region != "Aggregates")
head(df)
```

```
##   iso2c     country NY.GDP.PCAP.KD year iso3c                    region
## 1    AF Afghanistan             NA 1993   AFG                South Asia
## 2    AF Afghanistan             NA 1992   AFG                South Asia
## 3    AL     Albania       1197.647 1993   ALB      Europe & Central Asia
## 4    AL     Albania       1086.499 1992   ALB      Europe & Central Asia
## 5    DZ     Algeria       2867.194 1993   DZA Middle East & North Africa
## 6    DZ     Algeria       2994.489 1992   DZA Middle East & North Africa
```

Issues

Merging is in principle easy but in practice pretty hard.

Key challenges:

Different naming

Merging will not work if you have different codes in teh datasets you are merging:

```
df_1 <- data.frame(id = c("Germany", "U.K.", "Zaire"), X1 = 1:3)
df_2 <- data.frame(id = c("Germany", "UK", "DRC"), X2 = 5:7)

left_join(df_1, df_2)
```

```
## Joining, by = "id"
```

```
##        id X1 X2
## 1 Germany  1  5
## 2    U.K.  2 NA
## 3   Zaire  3 NA
```

We end up with missing data in X2 because the right ids were not found in the two datasets.

Better:

```
df_1 %>% left_join(df_2)
```

```
## Joining, by = "id"
```

```
##        id X1 X2
## 1 Germany  1  5
## 2    U.K.  2 NA
## 3   Zaire  3 NA
```

```
df_1 %>% mutate(id = recode(id, "U.K." = "UK", "Zaire" = "DRC")) %>%
  left_join(df_2)
```

```
## Joining, by = "id"
```

```
##        id X1 X2
## 1 Germany  1  5
## 2      UK  2  6
## 3     DRC  3  7
```

Best: have one dataset that has a complete list of authorative codes — hopefully a very standard set of codes. Prep all other data so that they have the same coding system.

Incomplete data produces many holes

```
df_1 <- data.frame(id = c("Germany"), X1 = 1)
df_2 <- data.frame(id = c("DRC"), X2 = 2)

left_join(df_1, df_2)
```

```
## Joining, by = "id"
```

```
##        id X1 X2
## 1 Germany  1 NA
```

Better:

```
df_0 <- data.frame(id = c("DRC", "Germany"))

df_0 %>% left_join(df_1) %>% left_join(df_2)
```

```
## Joining, by = "id"
## Joining, by = "id"
```

```
##        id X1 X2
## 1     DRC NA  2
## 2 Germany  1 NA
```

again: start off with a complete frame, with all countries, and add into this.

Data is the wrong shape

Very often you want to combine two datasets but they have different shapes. Viz:

```
df_1 <- data.frame(id = c("France", "Germany"), GDP_1990 = runif(2), GDP_1995 = runif(2))
df_1
```

```
##        id  GDP_1990  GDP_1995
## 1  France 0.1603886 0.6201746
## 2 Germany 0.2538812 0.2253731
```

```
df_2 <- data.frame(id = c("France", "France", "Germany", "Germany"),
                   Year = c(1990, 1995, 1990, 1995),
                   inflation = runif(4))
df_2
```

```
##        id Year inflation
## 1  France 1990 0.4024712
## 2  France 1995 0.5127072
## 3 Germany 1990 0.1117149
## 4 Germany 1995 0.6854723
```

we need to get these two into the same shape in order to merge. I use `gather` here but there are many other reshape functions that might be better for your data.

```
df_1 <-
  df_1 %>% gather("Year", "GDP",  -id) %>%
  mutate(Year = gsub("GDP_", "", Year),
         Year = as.numeric(Year))

df_1 %>% left_join(df_2)
```

```
## Joining, by = c("id", "Year")
```

```
##        id Year       GDP inflation
## 1  France 1990 0.1603886 0.4024712
## 2 Germany 1990 0.2538812 0.1117149
## 3  France 1995 0.6201746 0.5127072
## 4 Germany 1995 0.2253731 0.6854723
```

Notice that the joining was done on both id and Year since both of these were in the two datasets.