Tidy Data

Required packages

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.4
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Tidy versus Untidy Data

Health data (or any data in general) may come in multiple formats. Most of the graphical and numerical analysis in R and in other computing environments require data to be in standard form with each observation in a unique row, each variable in a unique column and each value in a unique cell.

Example 1 - Tidy Data

The following data table is an example of tidy data that shows the values of four variables country, year, population, and cases.

```
table1 <- tribble(
  ~country, ~year, ~cases, ~population,
  "Afghanistan",  1999,    745,   19987071,
  "Afghanistan",  2000,   2666,   20595360,
  "Brazil",       1999,  37737,  172006362,
  "Brazil",       2000,  80488,  174504898,
  "China",        1999, 212258, 1272915272,
  "China",        2000, 213766, 1280428583
)
table1
```

```
## # A tibble: 6 x 4
##   country      year  cases population
##   <chr>       <dbl>  <dbl>      <dbl>
## 1 Afghanistan  1999    745   19987071
## 2 Afghanistan  2000   2666   20595360
## 3 Brazil       1999  37737  172006362
## 4 Brazil       2000  80488  174504898
## 5 China        1999 212258 1272915272
```

```
## 6 China        2000 213766 1280428583
```

Tidy datasets more or less have a standard format as in `table1`. Untidy datasets may come in various formats.

Example 2 - Untidy Data -1

```
table2 <- tribble(
  ~country, ~year, ~type, ~count,
  "Afghanistan",  1999, "cases",      745,
  "Afghanistan",  1999, "population", 19987071,
  "Afghanistan",  2000, "cases",      2666,
  "Afghanistan",  2000, "population", 20595360,
  "Brazil",       1999, "cases",      37737,
  "Brazil",       1999, "population", 172006362,
  "Brazil",       2000, "cases",      80488,
  "Brazil",       2000, "population", 174504898,
  "China",        1999, "cases",      212258,
  "China",        1999, "population", 1272915272,
  "China",        2000, "cases",      213766,
  "China",        2000, "population", 1280428583,
)
  table2
```

```
## # A tibble: 12 x 4
##    country      year type            count
##    <chr>       <dbl> <chr>           <dbl>
##  1 Afghanistan  1999 cases             745
##  2 Afghanistan  1999 population   19987071
##  3 Afghanistan  2000 cases            2666
##  4 Afghanistan  2000 population   20595360
##  5 Brazil       1999 cases           37737
##  6 Brazil       1999 population  172006362
##  7 Brazil       2000 cases           80488
##  8 Brazil       2000 population  174504898
##  9 China        1999 cases          212258
## 10 China        1999 population 1272915272
## 11 China        2000 cases          213766
## 12 China        2000 population 1280428583
```

The Dataset in `table2` is said to have a `long` format where where the `type` column combines two variables `cases` and `population`.

Example 3 - Untidy Data -2

```
table3 <- tribble(
  ~country, ~year, ~rate,
  "Afghanistan",  1999,    "745/19987071",
  "Afghanistan",  2000,   "2666/20595360",
```

```
  "Brazil",       1999,  "37737/172006362",
  "Brazil",       2000,  "80488/174504898",
  "China",        1999, "212258/1272915272",
  "China",        2000, "213766/1280428583"
)
table3
```

```
## # A tibble: 6 x 3
##   country      year rate
##   <chr>       <dbl> <chr>
## 1 Afghanistan  1999 745/19987071
## 2 Afghanistan  2000 2666/20595360
## 3 Brazil       1999 37737/172006362
## 4 Brazil       2000 80488/174504898
## 5 China        1999 212258/1272915272
## 6 China        2000 213766/1280428583
```

`table3` presents data in table `table1` in slightly different format where column `rate` aggregrates cases and population as one measure.

Example 4 - Untidy Data -3

```
# cases
table4a <- tribble(
  ~country, ~`1999`, ~ `2000`,
  "Afghanistan", 745, 2666,
   "Brazil", 37737, 80488,
   "China", 212258, 213766
)

table4a
```

```
## # A tibble: 3 x 3
##   country     `1999` `2000`
##   <chr>        <dbl>  <dbl>
## 1 Afghanistan    745   2666
## 2 Brazil       37737  80488
## 3 China       212258 213766
```

```
# population
table4b <- tribble(
  ~country, ~`1999`, ~ `2000`,
  "Afghanistan", 19987071, 20595360,
   "Brazil", 172006362, 174504898,
   "China", 1272915272, 1280428583
)

table4b
```

```
## # A tibble: 3 x 3
##   country         `1999`     `2000`
##   <chr>            <dbl>      <dbl>
## 1 Afghanistan   19987071   20595360
```

```
## 2 Brazil       172006362  174504898
## 3 China       1272915272 1280428583
```

`table4a` and `table4b` are the examples of `wide` or longitudinal (repeated) data. Note the columns `1999` and `2000` can be treated as values of a single variable `year`.

Pivoting columns into variables

From wider to longer dataset

`table4a` and `table4b` can be turned into the long format as follows.

```
tb4a <- table4a %>%
  pivot_longer(c(`1999`, `2000`), names_to = "year", values_to = "cases")

tb4b<- table4b %>%
  pivot_longer(c(`1999`, `2000`), names_to = "year", values_to = "population")

left_join(tb4a,tb4b)
```

```
## Joining, by = c("country", "year")
```

```
## # A tibble: 6 x 4
##   country     year   cases population
##   <chr>       <chr>  <dbl>      <dbl>
## 1 Afghanistan 1999     745   19987071
## 2 Afghanistan 2000    2666   20595360
## 3 Brazil      1999   37737  172006362
## 4 Brazil      2000   80488  174504898
## 5 China       1999  212258 1272915272
## 6 China       2000  213766 1280428583
```

From two untidy datasets `table4a` and `table4b` we have created the tidy dataset as in `table1`.

From longer to wider dataset

A `long` data format can be turned into a `wide` format as follows:

```
table2 %>%
    pivot_wider(names_from = type, values_from = count)
```

```
## # A tibble: 6 x 4
##   country      year  cases population
##   <chr>       <dbl>  <dbl>      <dbl>
## 1 Afghanistan  1999    745   19987071
## 2 Afghanistan  2000   2666   20595360
## 3 Brazil       1999  37737  172006362
## 4 Brazil       2000  80488  174504898
## 5 China        1999 212258 1272915272
## 6 China        2000 213766 1280428583
```

Separating one column into two

```
table3 %>%
  separate(rate, into = c("cases", "population"))
```

```
## # A tibble: 6 x 4
##   country      year cases  population
##   <chr>       <dbl> <chr>  <chr>
## 1 Afghanistan  1999 745    19987071
## 2 Afghanistan  2000 2666   20595360
## 3 Brazil       1999 37737  172006362
## 4 Brazil       2000 80488  174504898
## 5 China        1999 212258 1272915272
## 6 China        2000 213766 1280428583
```

```
table3 %>%
  separate(rate, into = c("cases", "population"), convert = TRUE)
```

```
## # A tibble: 6 x 4
##   country      year  cases population
##   <chr>       <dbl>  <int>      <int>
## 1 Afghanistan  1999    745   19987071
## 2 Afghanistan  2000   2666   20595360
## 3 Brazil       1999  37737  172006362
## 4 Brazil       2000  80488  174504898
## 5 China        1999 212258 1272915272
## 6 China        2000 213766 1280428583
```

Uniting two columns into one

```
table5<- table3 %>%
  separate(year, into = c("century", "year"), sep = 2)
table5
```

```
## # A tibble: 6 x 4
##   country     century year  rate
##   <chr>       <chr>   <chr> <chr>
## 1 Afghanistan 19      99    745/19987071
## 2 Afghanistan 20      00    2666/20595360
## 3 Brazil      19      99    37737/172006362
## 4 Brazil      20      00    80488/174504898
## 5 China       19      99    212258/1272915272
## 6 China       20      00    213766/1280428583
```

```
table5 %>%
  unite(new, century, year)
```

```
## # A tibble: 6 x 3
##   country     new   rate
##   <chr>       <chr> <chr>
## 1 Afghanistan 19_99 745/19987071
## 2 Afghanistan 20_00 2666/20595360
## 3 Brazil      19_99 37737/172006362
```

```
## 4 Brazil      20_00 80488/174504898
## 5 China       19_99 212258/1272915272
## 6 China       20_00 213766/1280428583
```

```
table5 %>%
  unite(new, century, year, sep="")
```

```
## # A tibble: 6 x 3
##   country     new   rate
##   <chr>       <chr> <chr>
## 1 Afghanistan 1999  745/19987071
## 2 Afghanistan 2000  2666/20595360
## 3 Brazil      1999  37737/172006362
## 4 Brazil      2000  80488/174504898
## 5 China       1999  212258/1272915272
## 6 China       2000  213766/1280428583
```

WHO tuberculosis data

The dataset `who` is part of `tidyr` package and contains tuberculosis(TB) cases by year, country, age, gender, and diagnosis method. You can also download the data from `http://www.who.int/tb/country/data/download/en/`.

```
who
```

```
## # A tibble: 7,240 x 60
##    country iso2  iso3   year new_sp_m014 new_sp_m1524 new_sp_m2534 new_sp_m3544
##    <chr>   <chr> <chr> <int>       <int>        <int>        <int>        <int>
##  1 Afghan~ AF    AFG    1980          NA           NA           NA           NA
##  2 Afghan~ AF    AFG    1981          NA           NA           NA           NA
##  3 Afghan~ AF    AFG    1982          NA           NA           NA           NA
##  4 Afghan~ AF    AFG    1983          NA           NA           NA           NA
##  5 Afghan~ AF    AFG    1984          NA           NA           NA           NA
##  6 Afghan~ AF    AFG    1985          NA           NA           NA           NA
##  7 Afghan~ AF    AFG    1986          NA           NA           NA           NA
##  8 Afghan~ AF    AFG    1987          NA           NA           NA           NA
##  9 Afghan~ AF    AFG    1988          NA           NA           NA           NA
## 10 Afghan~ AF    AFG    1989          NA           NA           NA           NA
## # ... with 7,230 more rows, and 52 more variables: new_sp_m4554 <int>,
## #   new_sp_m5564 <int>, new_sp_m65 <int>, new_sp_f014 <int>,
## #   new_sp_f1524 <int>, new_sp_f2534 <int>, new_sp_f3544 <int>,
## #   new_sp_f4554 <int>, new_sp_f5564 <int>, new_sp_f65 <int>,
## #   new_sn_m014 <int>, new_sn_m1524 <int>, new_sn_m2534 <int>,
## #   new_sn_m3544 <int>, new_sn_m4554 <int>, new_sn_m5564 <int>,
## #   new_sn_m65 <int>, new_sn_f014 <int>, new_sn_f1524 <int>,
## #   new_sn_f2534 <int>, new_sn_f3544 <int>, new_sn_f4554 <int>,
## #   new_sn_f5564 <int>, new_sn_f65 <int>, new_ep_m014 <int>,
## #   new_ep_m1524 <int>, new_ep_m2534 <int>, new_ep_m3544 <int>,
## #   new_ep_m4554 <int>, new_ep_m5564 <int>, new_ep_m65 <int>,
## #   new_ep_f014 <int>, new_ep_f1524 <int>, new_ep_f2534 <int>,
## #   new_ep_f3544 <int>, new_ep_f4554 <int>, new_ep_f5564 <int>,
## #   new_ep_f65 <int>, newrel_m014 <int>, newrel_m1524 <int>,
## #   newrel_m2534 <int>, newrel_m3544 <int>, newrel_m4554 <int>,
## #   newrel_m5564 <int>, newrel_m65 <int>, newrel_f014 <int>,
```

```
## #   newrel_f1524 <int>, newrel_f2534 <int>, newrel_f3544 <int>,
## #   newrel_f4554 <int>, newrel_f5564 <int>, newrel_f65 <int>
```

Note the dataset is in wide format, as the columns after the column labeled `year` are actually values of multiple variables. We will use `pivot_longer` to create two new columns `key` having the names of each column and `cases` having the counts.

```
who1 <- who %>%
  pivot_longer(
    cols = new_sp_m014:newrel_f65,
    names_to = "key",
    values_to = "cases",
    values_drop_na = TRUE
  )
who1
```

```
## # A tibble: 76,046 x 6
##    country     iso2  iso3   year key          cases
##    <chr>       <chr> <chr> <int> <chr>        <int>
##  1 Afghanistan AF    AFG    1997 new_sp_m014      0
##  2 Afghanistan AF    AFG    1997 new_sp_m1524    10
##  3 Afghanistan AF    AFG    1997 new_sp_m2534     6
##  4 Afghanistan AF    AFG    1997 new_sp_m3544     3
##  5 Afghanistan AF    AFG    1997 new_sp_m4554     5
##  6 Afghanistan AF    AFG    1997 new_sp_m5564     2
##  7 Afghanistan AF    AFG    1997 new_sp_m65       0
##  8 Afghanistan AF    AFG    1997 new_sp_f014      5
##  9 Afghanistan AF    AFG    1997 new_sp_f1524    38
## 10 Afghanistan AF    AFG    1997 new_sp_f2534    36
## # ... with 76,036 more rows
```

Let's look at the structure of the values of new column `key` by counting them.

```
who1 %>%
  count(key)
```

```
## # A tibble: 56 x 2
##    key              n
##    <chr>        <int>
##  1 new_ep_f014   1032
##  2 new_ep_f1524  1021
##  3 new_ep_f2534  1021
##  4 new_ep_f3544  1021
##  5 new_ep_f4554  1017
##  6 new_ep_f5564  1017
##  7 new_ep_f65    1014
##  8 new_ep_m014   1038
##  9 new_ep_m1524  1026
## 10 new_ep_m2534  1020
## # ... with 46 more rows
```

Data dictionary is a good resource to get more information about these values.

- The first three letters of each column denote whether the column contains new or old cases of TB. In this dataset, each column contains new cases.

- The next two letters describe the type of TB, e.g. ‘rel‘ stands for relapse; ‘ep‘ stands for extrapulmonary TB etc.
- The sixth letter gives the sex of TB patients. The dataset groups cases by males (m) and females (f).
- The remaining numbers gives the age group, e.g. 014 : 0-14 years old, 1524 : 15-24 years old and so on.

```
who2 <- who1 %>%
  mutate(key = str_replace(key, "newrel", "new_rel"))
who2
```

```
## # A tibble: 76,046 x 6
##    country     iso2  iso3   year key          cases
##    <chr>       <chr> <chr> <int> <chr>        <int>
##  1 Afghanistan AF    AFG    1997 new_sp_m014      0
##  2 Afghanistan AF    AFG    1997 new_sp_m1524    10
##  3 Afghanistan AF    AFG    1997 new_sp_m2534     6
##  4 Afghanistan AF    AFG    1997 new_sp_m3544     3
##  5 Afghanistan AF    AFG    1997 new_sp_m4554     5
##  6 Afghanistan AF    AFG    1997 new_sp_m5564     2
##  7 Afghanistan AF    AFG    1997 new_sp_m65       0
##  8 Afghanistan AF    AFG    1997 new_sp_f014      5
##  9 Afghanistan AF    AFG    1997 new_sp_f1524    38
## 10 Afghanistan AF    AFG    1997 new_sp_f2534    36
## # ... with 76,036 more rows
```

```
who3 <- who2 %>%
  separate(key, c("new", "type", "sexage"), sep = "_")
who3
```

```
## # A tibble: 76,046 x 8
##    country     iso2  iso3   year new   type  sexage cases
##    <chr>       <chr> <chr> <int> <chr> <chr> <chr>  <int>
##  1 Afghanistan AF    AFG    1997 new   sp    m014       0
##  2 Afghanistan AF    AFG    1997 new   sp    m1524     10
##  3 Afghanistan AF    AFG    1997 new   sp    m2534      6
##  4 Afghanistan AF    AFG    1997 new   sp    m3544      3
##  5 Afghanistan AF    AFG    1997 new   sp    m4554      5
##  6 Afghanistan AF    AFG    1997 new   sp    m5564      2
##  7 Afghanistan AF    AFG    1997 new   sp    m65        0
##  8 Afghanistan AF    AFG    1997 new   sp    f014       5
##  9 Afghanistan AF    AFG    1997 new   sp    f1524     38
## 10 Afghanistan AF    AFG    1997 new   sp    f2534     36
## # ... with 76,036 more rows
```

```
who4 <- who3 %>%
  select(-new, -iso2, -iso3) %>%
  separate(sexage, c("sex", "age"), sep = 1)
who4
```

```
## # A tibble: 76,046 x 6
##    country      year type  sex   age   cases
##    <chr>       <int> <chr> <chr> <chr> <int>
##  1 Afghanistan  1997 sp    m     014       0
##  2 Afghanistan  1997 sp    m     1524     10
##  3 Afghanistan  1997 sp    m     2534      6
##  4 Afghanistan  1997 sp    m     3544      3
```

```
##  5 Afghanistan  1997 sp    m     4554      5
##  6 Afghanistan  1997 sp    m     5564      2
##  7 Afghanistan  1997 sp    m     65        0
##  8 Afghanistan  1997 sp    f     014       5
##  9 Afghanistan  1997 sp    f     1524     38
## 10 Afghanistan  1997 sp    f     2534     36
## # ... with 76,036 more rows
```

Note `who4` represents a tidy data that can used for further analsyis.

The above codes can be put together pipe operator `%>%` as follows.

```
who %>%
  pivot_longer(
    cols = new_sp_m014:newrel_f65,
    names_to = "key",
    values_to = "cases",
    values_drop_na = TRUE
  ) %>%
  mutate(
    key = stringr::str_replace(key, "newrel", "new_rel")
  ) %>%
  separate(key, c("new", "var", "sexage")) %>%
  select(-new, -iso2, -iso3) %>%
  separate(sexage, c("sex", "age"), sep = 1)
```

```
## # A tibble: 76,046 x 6
##    country      year var   sex   age   cases
##    <chr>       <int> <chr> <chr> <chr> <int>
##  1 Afghanistan  1997 sp    m     014       0
##  2 Afghanistan  1997 sp    m     1524     10
##  3 Afghanistan  1997 sp    m     2534      6
##  4 Afghanistan  1997 sp    m     3544      3
##  5 Afghanistan  1997 sp    m     4554      5
##  6 Afghanistan  1997 sp    m     5564      2
##  7 Afghanistan  1997 sp    m     65        0
##  8 Afghanistan  1997 sp    f     014       5
##  9 Afghanistan  1997 sp    f     1524     38
## 10 Afghanistan  1997 sp    f     2534     36
## # ... with 76,036 more rows
```

Save a permanent copy of the tidy data `who` as an R data set in `.rds` format.

```
setwd("~/Box/MyDocs/Teaching/Spring/2021/DSCI 610/LectureMaterials/Week 7/Lecture")
saveRDS(who4, file="who.rds")
```