# Data Wrangling in R with the Tidyverse (Part 2) - Practice Solutions

Jessica Minnier, PhD & Meike Niederhausen, PhD
OCTRI Biostatistics, Epidemiology, Research & Design (BERD) Workshop

2019/04/25 (Part 2)

solutions: bit.ly/berd_tidy2_solns
slides: bit.ly/berd_tidy2
pdf: bit.ly/berd_tidy2_pdf

# Load the data and packages

```
# install.packages("tidyverse","janitor","glue")
library(tidyverse)
library(lubridate)
library(janitor)
library(glue)
demo_data <- read_csv("data/yrbss_demo.csv")
qn_data <- read_csv("data/yrbss_qn.csv")
```

# Practice 1

1. Add a column of `1`'s to `qn_data` called `qn_yes` and save the resulting data as `qn_data2`.
2. Join `demo_data` and `qn_data2` by column `record`. Keep all rows from `demo_data` and only rows from `qn_data2` that match records in `demo_data`. Call the resulting data `all_data`.
3. Create a `tabyl()` of `qn_yes` for the data `all_data`.
4. Create a 2x2 table of `qn_yes` vs `grade`.

Note about the data:

- q8 = How often wear bicycle helmet
- q12 = Texted while driving
- q31 = Ever smoked
- qn24 = Bullied past 12 months

```
qn_data2 <- qn_data %>% add_column(qn_yes = 1)
all_data <- left_join(demo_data, qn_data2)
all_data %>% tabyl(qn_yes)
```

```
 qn_yes     n percent valid_percent
      1 10000     0.5             1
     NA 10000     0.5            NA
```

```
all_data %>% tabyl(qn_yes,grade)
```

```
 qn_yes 10th 11th 12th  9th NA_
      1 2443 2498 2287 2573 199
     NA 2464 2393 2290 2646 207
```

# Practice 2

1. Make `DBP_wide` into a long dataframe based on the repeated DBP columns and save it as `DBP_long`.
2. Clean up the visit column of `DBP_long` so that the values are 1, 2, 3, and save it as `DBP_long`.
3. Make `DBP_long` wide with column names `visit.1, visit.2, visit.3` for the DBP values, and save it as `DBP_wide2`.
4. Join `DBP_long` with `BP_long2` so that we have one data frame with columns id, sex, visit, SBP, DBP, and age. Save this as `BP_both_long`.

# Practice 2 Initial Data

Copy and paste the code below into R to create the datasets:

```
DBP_wide <- tibble(id = letters[1:4],
                   sex = c("F", "M", "M", "F"),
                   v1.DBP = c(88, 84, 102, 70),
                   v2.DBP = c(78, 78, 96, 76),
                   v3.DBP = c(94, 82, 94, 74),
                   age=c(23, 56, 41, 38)
                   )
BP_wide <- tibble(id = letters[1:4],
                     sex = c("F", "M", "M", "F"),
                     SBP_v1 = c(130, 120, 130, 119),
                     SBP_v2 = c(110, 116, 136, 106),
                     SBP_v3 = c(112, 122, 138, 118))
BP_long <- BP_wide %>%
  gather(key = "visit", value = "SBP", SBP_v1:SBP_v3)
BP_long2 <- BP_long %>%
  mutate(visit = str_replace(visit,"SBP_v",""))
```

# Practice 2 solutions (1/2)

```
DBP_long <- DBP_wide %>%
  gather(key = "visit", value = "DBP",
         v1.DBP, v2.DBP, v3.DBP) %>%
  mutate(visit = str_replace(
    visit, c("v"), "")) %>%
  mutate(visit = str_replace(
    visit, ".DBP", ""))
DBP_long
```

```
# A tibble: 12 x 5
   id    sex     age visit   DBP
   <chr> <chr> <dbl> <chr> <dbl>
 1 a     F        23 1        88
 2 b     M        56 1        84
 3 c     M        41 1       102
 4 d     F        38 1        70
 5 a     F        23 2        78
 6 b     M        56 2        78
 7 c     M        41 2        96
 8 d     F        38 2        76
 9 a     F        23 3        94
10 b     M        56 3        82
```

```
DBP_wide2 <- DBP_long %>%
  spread(
    key = "visit", value = "DBP",
    sep=".") # specify separating character
DBP_wide2
```

```
# A tibble: 4 x 6
  id    sex     age visit.1 visit.2 visit.3
  <chr> <chr> <dbl>   <dbl>   <dbl>   <dbl>
1 a     F        23      88      78      94
2 b     M        56      84      78      82
3 c     M        41     102      96      94
4 d     F        38      70      76      74
```

# Practice 2 solutions (2/2)

```
BP_both_long <- left_join(BP_long2, DBP_long, by = c("id", "sex", "visit"))
BP_both_long
```

```
# A tibble: 12 x 6
   id    sex   visit   SBP   age   DBP
   <chr> <chr> <chr> <dbl> <dbl> <dbl>
 1 a     F     1       130    23    88
 2 b     M     1       120    56    84
 3 c     M     1       130    41   102
 4 d     F     1       119    38    70
 5 a     F     2       110    23    78
 6 b     M     2       116    56    78
 7 c     M     2       136    41    96
 8 d     F     2       106    38    76
 9 a     F     3       112    23    94
10 b     M     3       122    56    82
11 c     M     3       138    41    94
12 d     F     3       118    38    74
```

# Practice 3

```
messy_data <- tibble(NAME = c("J N","A C","D E"),
                     `months follow up` = c("", 10, 11),
                     `Date of visit` = c("July 31, 2003", "Nov 12, 2005", "Aug 3, 2007"))
```

1. Clean column names with `clean_names()`.
2. Replace missing ("") data in `months_follow_up` with NA.
3. Convert `months_follow_up` to a numeric variable.
4. Convert `date_of_visit` to a date.
5. Create a column called `date_last_visit` that is the date of visit *plus* months of follow up.
6. Remove rows (cases) with missing data in `months_follow_up`.
7. Remove the spaces in `name`.

```
messy_data
```

```
# A tibble: 3 x 3
  NAME  `months follow up` `Date of visit`
  <chr> <chr>              <chr>
1 J N   ""                 July 31, 2003
2 A C   10                 Nov 12, 2005
3 D E   11                 Aug 3, 2007
```

# Practice solutions 3 (1/2)

```
clean_data <- messy_data %>%
  clean_names() %>%
  mutate(
    months_follow_up = replace_na(months_follow_up,""),
    months_follow_up = as.numeric(months_follow_up),
    date_of_visit = mdy(date_of_visit),
    date_last_visit = date_of_visit + months(months_follow_up))
clean_data
```

```
# A tibble: 3 x 4
  name  months_follow_up date_of_visit date_last_visit
  <chr>            <dbl> <date>        <date>
1 J N                 NA 2003-07-31    NA
2 A C                 10 2005-11-12    2006-09-12
3 D E                 11 2007-08-03    2008-07-03
```

# Practice solutions 3 (2/2)

```
clean_data <- clean_data %>%
  drop_na(months_follow_up) %>%
  mutate(name = str_replace_all(name," ",""))
clean_data
```

```
# A tibble: 2 x 4
  name  months_follow_up date_of_visit date_last_visit
  <chr>            <dbl> <date>        <date>
1 AC                  10 2005-11-12    2006-09-12
2 DE                  11 2007-08-03    2008-07-03
```