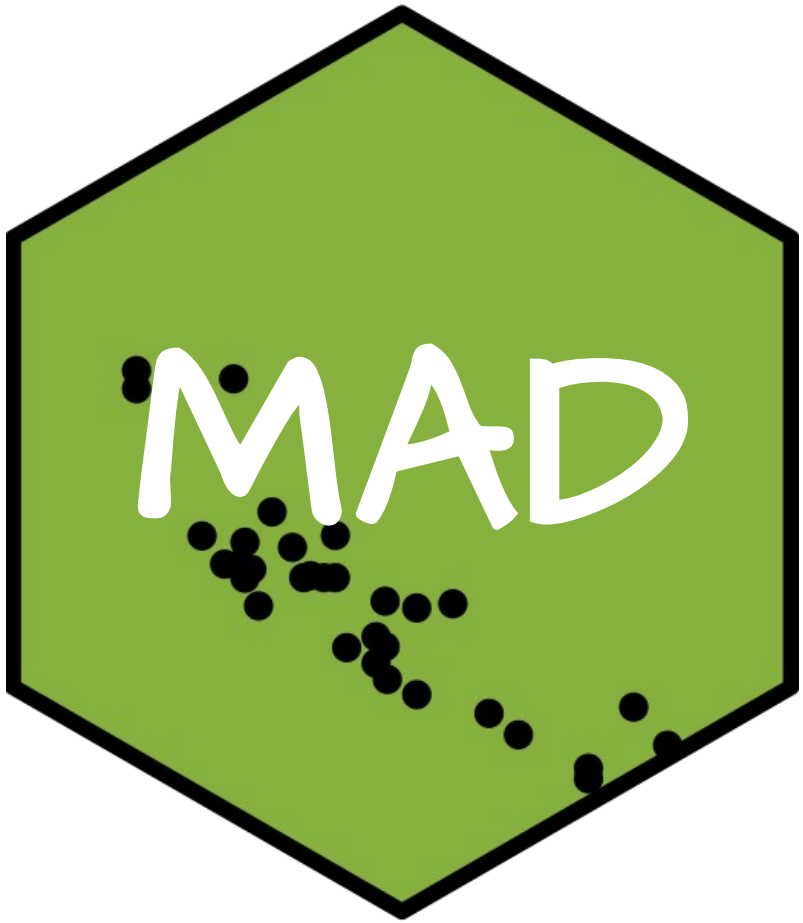
MAD – Data Analysis & Biostatistics in R

Getting to Work - 2

James R. Hunter, Ph.D.

DIPA, EPM, UNIFESP

11 September 2020


MAD

Section 1

Tidy Data

Summary of the Data

- Overall Structure of Data
  - How many variables
  - What types
- Use either `str()` or `glimpse()` -str()\- Base R -glimpse()\-tibble‘

Load soro as Example

```
soro <- readRDS("C:/Users/james/OneDrive/Documents/MAD/MAD-Infecto-2020/einstein_so
str(soro)
```

```
## 'data.frame':    200 obs. of  10 variables:
##  $ pacid     : chr  "b6d668e4f818f7b3643ed593b8fb902bf9d2501e" "a090625661c06e9c
##  $ dt_collect: chr  "28/05/2020" "11/05/2020" "16/06/2020" "10/06/2020" ...
##  $ analysis  : chr  "IgM, COVID19" "IgG, COVID19" "IgG, COVID19" "COVID IgM Inte
##  $ result    : chr  "0.74" "0.03" "0.02" "Não reagente" ...
##  $ unit      : chr  "AU/ml" "AU/ml" "AU/ml" "NULL" ...
##  $ reference : chr  "<=0.90" "<=0.90" "<=0.90" "" ...
##  $ sex       : Factor w/ 2 levels "female","male": 2 1 1 2 1 1 1 2 1 2 ...
##  $ birth_yr  : num  1989 1975 1997 2006 1983 ...
##  $ uf        : Factor w/ 25 levels "AC","AL","AM",..: 24 9 24 24 24 24 24 24 24
##  $ city      : Factor w/ 21 levels "BARUERI","CAMPINAS",..: 19 NA 19 19 19 19 19
```

glimpse() Alternative to str()

```
tibble::glimpse(soro)
```

```
## Rows: 200
## Columns: 10
## $ pacid       <chr> "b6d668e4f818f7b3643ed593b8fb902bf9d2501e", "a090625661c...
## $ dt_collect  <chr> "28/05/2020", "11/05/2020", "16/06/2020", "10/06/2020", ...
## $ analysis    <chr> "IgM, COVID19", "IgG, COVID19", "IgG, COVID19", "COVID I...
## $ result      <chr> "0.74", "0.03", "0.02", "Não reagente", "0.47", "0.90", ...
## $ unit        <chr> "AU/ml", "AU/ml", "AU/ml", "NULL", "AU/ml", "AU/ml", "NU...
## $ reference   <chr> "<=0.90", "<=0.90", "<=0.90", "", "<=0.90", "<=0.90", ""...
## $ sex         <fct> male, female, female, male, female, female, female, male...
## $ birth_yr    <dbl> 1989, 1975, 1997, 2006, 1983, 1963, 1988, 1971, 1968, 19...
## $ uf          <fct> SP, GO, SP, SP, SP, SP, SP, SP, SP, SP, SP, SP, SP, SP, ...
## $ city        <fct> SAO PAULO, NA, SAO PAULO, SAO PAULO, SAO PAULO, SAO PAUL...
```

Overall Look in More Detail

- `summarytools::dfSummary()`
  - Short summary of each variable in set
  - Presentation based on variable type
  - Many options
  - I leave out "graph" column
    - ★ Set `graph.col = FALSE` to omit

```
library(summarytools)
dfSummary(einstein_soro, graph.col = FALSE)
```

```
## Data Frame Summary
## soro
## Dimensions: 200 x 1
## Duplicates: 2
##
## ------------------------------------------------------------------------------------------
## No   Variable      Stats / Values                    Freqs (% of Valid)   Valid    Missing
## ---- ------------ --------------------------------- -------------------- -------- ---------
## 1    pacid        1. 373a2ae841153ee5f4d86c245        2 ( 1.0%)          200      0
##      [character]  2. 95ecc1410a0f8abfde332e73d        2 ( 1.0%)          (100%)   (0%)
##                   3. 0185739f5a8229250be56af87        1 ( 0.5%)
##                   4. 018c762d69595658644fc0236        1 ( 0.5%)
##                   5. 0201860541a4da84f23b4f1b9        1 ( 0.5%)
##                   6. 02a4efe034724d2631bb563eb        1 ( 0.5%)
##                   7. 043ef43cd291fb45c23a4e8df        1 ( 0.5%)
##                   8. 06d164f01d1f385e4e2f5a341        1 ( 0.5%)
##                   9. 08f4e21de519f8d521fca7df6        1 ( 0.5%)
##                   10. 0a67cd063da4bfade9be8e0e4       1 ( 0.5%)
##                   [ 188 others ]                    188 (94.0%)
## ------------------------------------------------------------------------------------------
```

```
## Data Frame Summary
## soro
## Dimensions: 200 x 1
## Duplicates: 139
##
## ------------------------------------------------------------------------------
## No   Variable      Stats / Values     Freqs (% of Valid)   Valid    Missing
## ---- ------------ ------------------ -------------------- -------- ---------
## 1    dt_collect    1. 08/06/2020       13 ( 6.5%)          200      0
##      [character]   2. 05/06/2020       11 ( 5.5%)          (100%)   (0%)
##                    3. 04/06/2020        8 ( 4.0%)
##                    4. 06/06/2020        8 ( 4.0%)
##                    5. 14/05/2020        8 ( 4.0%)
##                    6. 22/05/2020        8 ( 4.0%)
##                    7. 12/05/2020        7 ( 3.5%)
##                    8. 19/05/2020        7 ( 3.5%)
##                    9. 21/05/2020        7 ( 3.5%)
##                    10. 10/06/2020       6 ( 3.0%)
##                    [ 51 others ]      117 (58.5%)
## ------------------------------------------------------------------------------
```

```
## Data Frame Summary
## soro
## Dimensions: 200 x 1
## Duplicates: 196
##
## ------------------------------------------------------------------------------
## No   Variable      Stats / Values          Freqs (% of Valid)   Valid    Missing
## ---- ------------ ----------------------- -------------------- -------- ---------
## 1    analysis     1. COVID IgG Interp     36 (18.0%)           200      0
##      [character]  2. COVID IgM Interp     45 (22.5%)           (100%)   (0%)
##                   3. IgG, COVID19         60 (30.0%)
##                   4. IgM, COVID19         59 (29.5%)
## ------------------------------------------------------------------------------
```

```
## Data Frame Summary
## soro
## Dimensions: 200 x 1
## Duplicates: 130
##
## ------------------------------------------------------------------------------
## No   Variable      Stats / Values      Freqs (% of Valid)   Valid    Missing
## ---- ------------ ------------------- -------------------- -------- ---------
## 1    result        1. Não reagente     68 (34.0%)           200      0
##      [character]   2. 0.02             11 ( 5.5%)           (100%)   (0%)
##                    3. Reagente         10 ( 5.0%)
##                    4. 0.03              7 ( 3.5%)
##                    5. 0.04              7 ( 3.5%)
##                    6. 0.06              7 ( 3.5%)
##                    7. 0.54              5 ( 2.5%)
##                    8. 0.05              3 ( 1.5%)
##                    9. 0.10              3 ( 1.5%)
##                    10. 0.33             3 ( 1.5%)
##                    [ 60 others ]       76 (38.0%)
## ------------------------------------------------------------------------------
```

```
## Data Frame Summary
## soro
## Dimensions: 200 x 3
## Duplicates: 194
##
## ------------------------------------------------------------------------------
## No   Variable      Stats / Values        Freqs (% of Valid)   Valid    Missing
## ---- ------------ --------------------- -------------------- -------- ---------
## 1    unit         1. AU/ml              114 (57.0%)          200      0
##      [character]  2. NULL                86 (43.0%)          (100%)   (0%)
##
## 2    reference    1. (Empty string)      75 (37.5%)          200      0
##      [character]  2. <=0.90             114 (57.0%)          (100%)   (0%)
##                   3. Não Reagente        11 ( 5.5%)
##
## 3    sex          1. female              98 (49.0%)          200      0
##      [factor]     2. male               102 (51.0%)          (100%)   (0%)
## ------------------------------------------------------------------------------
```

```
## Data Frame Summary
## soro
## Dimensions: 200 x 1
## Duplicates: 138
##
## ------------------------------------------------------------------------------------
## No   Variable    Stats / Values               Freqs (% of Valid)   Valid     Missing
## ---- ----------- ---------------------------- -------------------- --------- ---------
## 1    birth_yr    Mean (sd) : 1978.2 (15.8)    61 distinct values   197       3
##      [numeric]   min < med < max:                                  (98.5%)   (1.5%)
##                  1933 < 1980 < 2020
##                  IQR (CV) : 21 (0)
## ------------------------------------------------------------------------------------
```

```
## Data Frame Summary
## soro
## Dimensions: 200 x 2
## Duplicates: 186
##
## ------------------------------------------------------------------------------
## No   Variable   Stats / Values              Freqs (% of Valid)   Valid    Missing
## ---- ---------- -------------------------- -------------------- -------- ---------
## 1    uf         1. AC                        0 ( 0.0%)          200      0
##      [factor]   2. AL                        0 ( 0.0%)          (100%)   (0%)
##                 3. AM                        0 ( 0.0%)
##                 4. AP                        0 ( 0.0%)
##                 5. BA                        0 ( 0.0%)
##                 6. CE                        0 ( 0.0%)
##                 7. DF                        0 ( 0.0%)
##                 8. ES                        0 ( 0.0%)
##                 9. GO                        1 ( 0.5%)
##                 10. MA                       0 ( 0.0%)
##                 [ 15 others ]              199 (99.5%)
##
## 2    city       1. BARUERI                   4 ( 2.2%)          180      20
##      [factor]   2. CAMPINAS                  0 ( 0.0%)          (90%)    (10%)
##                 3. CARAPICUIBA               2 ( 1.1%)
##                 4. COTIA                     0 ( 0.0%)
##                 5. DIADEMA                   0 ( 0.0%)
##                 6. EMBU                      1 ( 0.6%)
##                 7. EMBU-GUACU                0 ( 0.0%)
##                 8. GUARULHOS                 0 ( 0.0%)
##                 9. ITAPECERICA DA SERRA      0 ( 0.0%)
##                 10. ITAPEVI                  0 ( 0.0%)
##                 [ 11 others ]              173 (96.1%)
## ------------------------------------------------------------------------------
```

Munging This Data Set

- `dt_collect`: non-standard format, character
  - Transform to Date with functions from `lubridate`
- `analysis`: different ways of reporting same test
  - Can isolate the antibody name with `stringr` functions
- `result`: problem of `Não reagente` as 0
  - Other string values
  - Deal with string values and transform to numeric
- `unit`: only one value
  - Eliminate: not useful to analysis
  - Use `janitor::remove_constant()`
- `reference`: three values; what use is it?
  - Can assign useful values for 3 values or remove

Section 2

Clean Variable Names

- Universal first munging step
- Our variables already have been cleaned
- `janitor::clean_names()`

Clean Names Example

```
test_df <- as.data.frame(matrix(ncol = 6))
names(test_df) <- c("firstName", "$abc@!*", "% successful (2009)",
                    "REPEAT VALUE", "REPEAT VALUE", "")
test_df
```

```
##   firstName $abc@!* % successful (2009) REPEAT VALUE REPEAT VALUE
## 1        NA      NA                   NA           NA            NA NA
```

```
# apply clean_names()

test_df <- janitor::clean_names(test_df)

test_df
```

```
##   first_name abc percent_successful_2009 repeat_value repeat_value_2  x
## 1         NA  NA                      NA           NA             NA NA
```

Assigning Names to Variables

- Can use `names()` to create names for your variables
- Names should be in a vector the same length as the number of columns
- `names(test_df) <-` to receive the vector

```
names(test_df) <- c("first_name", "last_name",
                    "percent_successful_2009",
                    "value_1", "value_2", "standard")
```

Section 3

Munging the Variables

Section 4

Convert Dates from Text to Date Format

Current format of `dt_collect`

- `dt_collect` format
  - Currently string in "dd/mm/yyyy"
  - Brazilian standard format

Parsing the Format

- `lubridate` package
  - Need to put in memory
    - ★ Not automatically loaded with `tidyverse`
  - `library(lubridate)`
- Functions combinations of 1st letters of day, month, year
  - In order that data is recorded
  - In our case, function would be `dmy()`
- If it were American standard date ("mm/dd/yyyy")
  - Function would be `mdy()`
- `lubridate` has all the possibilities
- All formats can work with any separator
  - Ignores them

Example of Date Conversion with `lubridate`

```r
library(lubridate)  # need to load as it is not loaded as part of tidyverse

br_text <- "28/05/2020"
(br_date <- dmy(br_text)) # remember: parentheses outside a function
```

```
## [1] "2020-05-28"
```

```r
#                             print it to screen

us_text <- "05-28-2020"
(us_date <- mdy(us_text))
```

```
## [1] "2020-05-28"
```

Section 5

Connecting Functions – The Pipe

Need to Join Functions Together

- In a way we can later understand and remember
- Hypothetical example (from Ismay and Kim, **ModernDive**)
  - Data frame `x`
  - Functions `f()`, `g()`, and `h()`
- Sequence of actions:
  - Take `x` then
  - Use `x` as an input to a function `f()` then
  - Use the output of `f(x)` as an input to a function `g()` then
  - Use the output of `g(f(x))` as an input to a function `h()`
- Nesting parentheses solution
  - `h(g(f(x)))`
  - Easy to understand – **NOT**

"Pipe" (%>%) Operator

- Takes what is on the left side of operator
- Makes that the first argument of function on right side
- Sort of means "and then"

Example in Pipe Form

```
x %>%
  f() %>%
  g() %>%
  h()
```

1. Take `x` then
2. Use this output as the input to the next function `f()` then
3. Use this output as the input to the next function `g()` then
4. Use this output as the input to the next function `h()`

Section 6

`mutate()` Function - How We Modify (and Add) Variables

Basics of `mutate()`

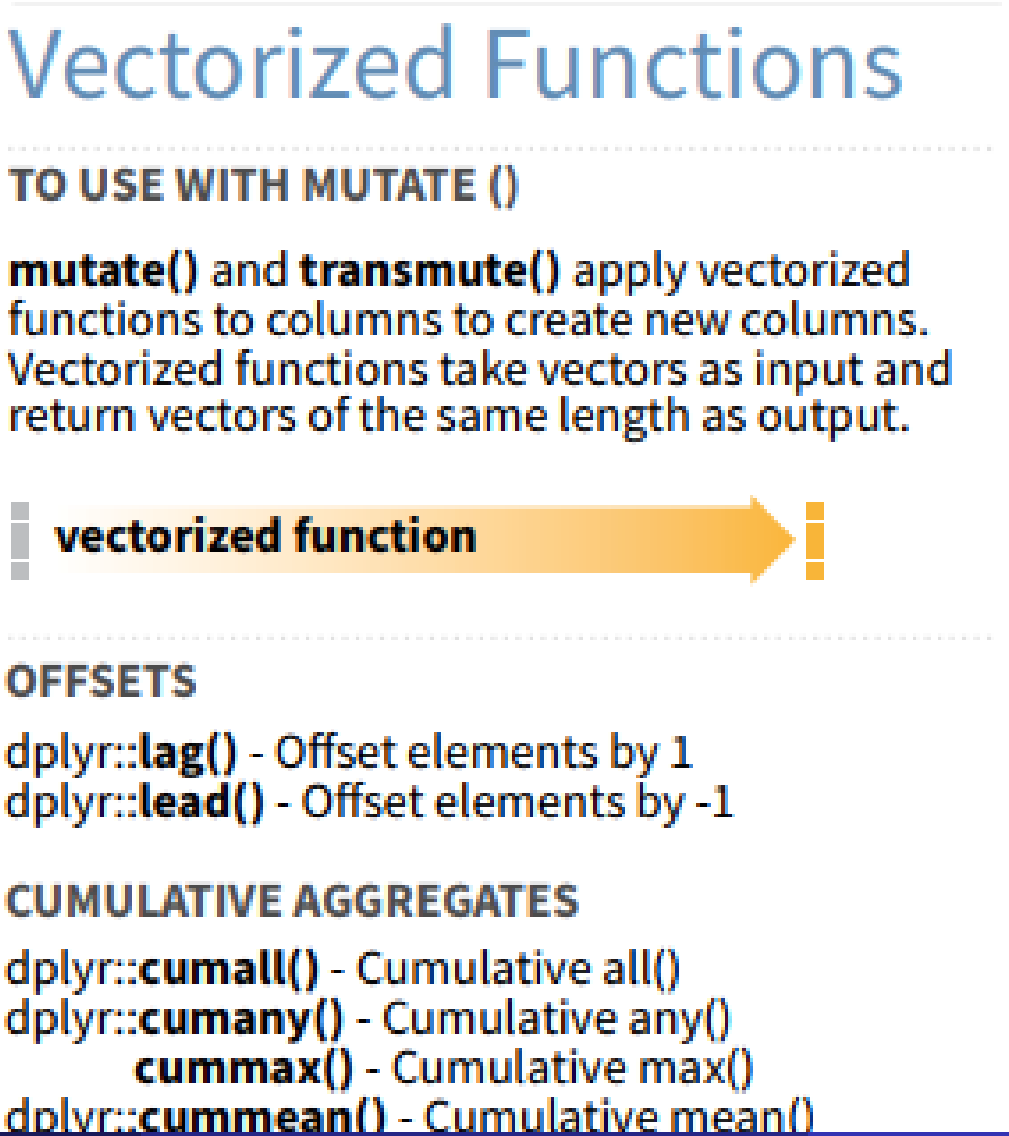
- `dplyr::mutate()`
  - 1st argument: data frame or tibble to be modified
  - 2nd argument: modification in form of assignment
    - ★ **Here** assignment uses "=" not "<-"

`mutate()` Assignment

- Variable name on left side
- If variable name does not exist in tibble, it will be added
- If existing variable, overwrite current value
  - Do this in a new tibble

Functions for Assignment

- Wide variety



Steps to Accomplish Mutation (`dt_collect`)

1. Establish the name of the revised data frame then
2. Assign to it the data from the old version then
3. Transform the date to a Date class

Code to Accomplish This

```
library(tidyverse)
soro_b <- soro %>%  # steps 1 and 2; note use of Pipe
  mutate(dt_collect = dmy(dt_collect)) # step 3

glimpse(soro_b$dt_collect)
```

```
##  Date[1:200], format: "2020-05-28" "2020-05-11" "2020-06-16" "2020-06-10" "2020-
```

Section 7

Clean up `analysis` Categories

Remember

- `analysis` had 2 ways of referring to each of 2 antibodies
- We want to reduce variable to values "IgG" and "IgM" only

```
table(soro$analysis)
```

```
##
## COVID IgG Interp COVID IgM Interp     IgG, COVID19     IgM, COVID19
##              36               45               60               59
```

`mutate()` with `ifelse()`

- All the values include the antibody name we want
  - "IgG" or "IgM"
- We can search for "IgG"
  - If case has it, put that value in `analysis`
    - If not, put other
- Use `ifelse()` to make the selection
- Because only two values, transform `analysis` into factor
- Do the search with `stringr::str_detect(var, pattern)`
  - `var`: variable to be searched
  - `pattern`: pattern to detect
  - `str_detect(analysis, "IgG")`

Code for Mutation

```
soro_b <- soro %>%
  mutate(analysis = ifelse(str_detect(analysis, "IgG"), "IgG", "IgM")) %>%
  mutate(analysis = factor(analysis))

glimpse(soro_b$analysis)
```

```
##  Factor w/ 2 levels "IgG","IgM": 2 1 1 2 2 2 2 1 2 2 ...
```

2nd Approach for `analysis` with `forcats`

- Use functions from `forcats` to manipulate `analysis`
- `forcats`: functions to manipulate factors
- Start by transforming `analysis` to a factor data type
- Call `factor()`

```
x <- c("a", "b", "c")
glimpse(x)
```

```
##  chr [1:3] "a" "b" "c"
```

```
fct_x <- factor(x)
glimpse(fct_x)
```

```
##  Factor w/ 3 levels "a","b","c": 1 2 3
```

- Values now: 1, 2, 3
- Levels: a, b, c

Apply This to analysis

- What we will do with analysis is manipulate levels

```
soro_b <- soro %>%
  mutate(analysis_f = factor(analysis))
glimpse(soro_b$analysis_f)
```

```
##  Factor w/ 4 levels "COVID IgG Interp",..: 4 3 3 2 4 4 2 3 4 4 ...
```

```
levels(soro_b$analysis_f)
```

```
## [1] "COVID IgG Interp" "COVID IgM Interp" "IgG, COVID19"     "IgM, COVID19"
```

```
table(soro_b$analysis_f)
```

```
##
## COVID IgG Interp COVID IgM Interp     IgG, COVID19     IgM, COVID19
##               36               45               60               59
```

`mutate()` Applied to `fct_collapse()`

- `forcats::fct_collapse()`: reduce number of levels based on existing values
- **Don't forget the Cheat Sheet: "Factors with forcats::"**
- Since we will have 2 final levels ("IgG" or "IgM")
  - Need to define each separately

Code for This

```
soro_b <- soro %>%
  mutate(analysis_f = factor(analysis)) %>%
  mutate(analysis_f = fct_collapse(analysis_f,
                                   IgG = c("COVID IgG Interp", "IgG, COVID19"),
                                   IgM = c("COVID IgM Interp", "IgM, COVID19")))
glimpse(soro_b$analysis_f)
```

```
##  Factor w/ 2 levels "IgG","IgM": 2 1 1 2 2 2 2 1 2 2 ...
```

```
levels(soro_b$analysis_f)
```

```
## [1] "IgG" "IgM"
```

```
fct_count(soro_b$analysis_f)
```

```
## # A tibble: 2 x 2
##   f         n
##   <fct> <int>
## 1 IgG      96
## 2 IgM     104
```

Even More Compact Form to Get Same Result

```
soro_b <- soro %>%
mutate(analysis_f = fct_collapse(factor(analysis),
                                 IgG = c("COVID IgG Interp", "IgG, COVID19"),
                                 IgM = c("COVID IgM Interp", "IgM, COVID19")))
```

Section 8

Non-Numeric Values in `result`

Problem - String Values in `result`

- "Não reagente" and "Reagente"

```
dfSummary(soro$result, graph.col = FALSE)
```

```
## Data Frame Summary
## soro
## Dimensions: 200 x 1
## Duplicates: 130
##
## ------------------------------------------------------------------------------
## No   Variable      Stats / Values      Freqs (% of Valid)   Valid    Missing
## ---- ------------ ------------------- -------------------- -------- ---------
## 1    result        1. Não reagente     68 (34.0%)           200      0
##      [character]   2. 0.02             11 ( 5.5%)           (100%)   (0%)
##                    3. Reagente         10 ( 5.0%)
##                    4. 0.03              7 ( 3.5%)
##                    5. 0.04              7 ( 3.5%)
##                    6. 0.06              7 ( 3.5%)
##                    7. 0.54              5 ( 2.5%)
##                    8. 0.05              3 ( 1.5%)
##                    9. 0.10              3 ( 1.5%)
##                    10. 0.33             3 ( 1.5%)
##                    [ 60 others ]       76 (38.0%)
## ------------------------------------------------------------------------------
```

Strategy in Base R

- Treat "Não reagente" as 0
- Treat "Reagente" and blank strings as NA
- Use `for` loop to test all the cases
- Use `if...then...else` to test values and make changes

Code

```
soro_b <- soro

for(i in 1:nrow(soro_b)){

if(soro_b$result[i] == "Nao reagente") {
  soro_b$result[i] <- 0
  } else {
    if(soro_b$result[i] %in% c("Reagente", "")){
      soro_b$result[i] <- NA
    } # end second if
}  # end else
}  # end of if
}  # end of loop
soro_b$result <- as.numeric(soro_b$result)
# above line is what made else test optional
```

Same Logic with tidyverse: mutate() & ifelse()

```r
soro_b <- soro %>%
  mutate(result = as.numeric(ifelse(result == "Não reagente", 0, result)))
summarytools::dfSummary(soro_b$result, graph.col = FALSE)
```

```
## Data Frame Summary
## soro_b
## Dimensions: 200 x 1
## Duplicates: 133
##
## ------------------------------------------------------------------------------
## No   Variable    Stats / Values         Freqs (% of Valid)   Valid   Missing
## ---- ----------- ---------------------- -------------------- ------- ---------
## 1    result      Mean (sd) : 1 (4)      66 distinct values   186     14
##      [numeric]   min < med < max:                           (93%)   (7%)
##                  0 < 0 < 30.8
##                  IQR (CV) : 0.5 (3.9)
## ------------------------------------------------------------------------------
```

New Problem with `result()`

- What is that 30.8 Value?
- Mean = 1.7
- Value is 5.29 standard deviations outside mean
- Reference value from `reference` is "<=0.90"
  - This value 30 times higher than reference
- **Outlier**
- Important Issue in statistics
- Lesson: Take careful note of range of numerical values
  - Problem to be solved during analysis phase

Section 9

Removing Unnecessary Variables (Columns)

Remove `unit` with `janitor::remove_constant()`

- `unit` only has 1 value: "AU/ml"
- No variance to measure
- `remove_constant()`: removes columns that only have 1 value (plus NA)

```
table(soro$unit, useNA = "ifany")
```

```
##
## AU/ml  <NA>
##   114    86
```

Remove unit

```
soro_b <-  soro %>%
  janitor::remove_constant(na.rm = TRUE)
glimpse(soro_b)
```

```
## Rows: 200
## Columns: 9
## $ pacid      <chr> "b6d668e4f818f7b3643ed593b8fb902bf9d2501e", "a090625661c...
## $ dt_collect <chr> "28/05/2020", "11/05/2020", "16/06/2020", "10/06/2020", ...
## $ analysis   <chr> "IgM, COVID19", "IgG, COVID19", "IgG, COVID19", "COVID I...
## $ result     <chr> "0.74", "0.03", "0.02", "Não reagente", "0.47", "0.90", ...
## $ reference  <chr> "<=0.90", "<=0.90", "<=0.90", "", "<=0.90", "<=0.90", ""...
## $ sex        <fct> male, female, female, male, female, female, female, male...
## $ birth_yr   <dbl> 1989, 1975, 1997, 2006, 1983, 1963, 1988, 1971, 1968, 19...
## $ uf         <fct> SP, GO, SP, SP, SP, SP, SP, SP, SP, SP, SP, SP, SP, SP, ...
## $ city       <fct> SAO PAULO, NA, SAO PAULO, SAO PAULO, SAO PAULO, SAO PAUL...
```

Remove `reference` with `dplyr::select()`

- `reference` has really one value: "<=0.90"

```
table(soro$reference, useNA = "ifany")
```

```
##
##                   <=0.90 Não Reagente
##           75          114           11
```

`select()`: 2nd Major `dplyr` Verb

- Works on columns (variables)
- If we want to include columns in an operation
  - Positively `select()` them in arguments

Simple select() Example

```
a <- tibble(x = c("a", "b", "c"),
            y = 1:3,
            z = c("d", "e", "f"))
a #show the tibble on the screen
```

```
## # A tibble: 3 x 3
##   x         y z
##   <chr> <int> <chr>
## 1 a         1 d
## 2 b         2 e
## 3 c         3 f
```

```
a %>% select(y) #just show the selected variable
```

```
## # A tibble: 3 x 1
##       y
##   <int>
## 1     1
## 2     2
## 3     3
```

Remove a Variables with `select(-var)`

```
a
```

```
## # A tibble: 3 x 3
##   x         y z
##   <chr> <int> <chr>
## 1 a         1 d
## 2 b         2 e
## 3 c         3 f
```

```
a %>% select(-x)
```

```
## # A tibble: 3 x 2
##       y z
##   <int> <chr>
## 1     1 d
## 2     2 e
## 3     3 f
```

Remove reference with dplyr::select()

```
soro_b <- soro %>%
  select(-reference)
glimpse(soro_b)
```

```
## Rows: 200
## Columns: 9
## $ pacid      <chr> "b6d668e4f818f7b3643ed593b8fb902bf9d2501e", "a090625661c...
## $ dt_collect <chr> "28/05/2020", "11/05/2020", "16/06/2020", "10/06/2020", ...
## $ analysis   <chr> "IgM, COVID19", "IgG, COVID19", "IgG, COVID19", "COVID I...
## $ result     <chr> "0.74", "0.03", "0.02", "Não reagente", "0.47", "0.90", ...
## $ unit       <chr> "AU/ml", "AU/ml", "AU/ml", NA, "AU/ml", "AU/ml", NA, "AU...
## $ sex        <fct> male, female, female, male, female, female, female, male...
## $ birth_yr   <dbl> 1989, 1975, 1997, 2006, 1983, 1963, 1988, 1971, 1968, 19...
## $ uf         <fct> SP, GO, SP, SP, SP, SP, SP, SP, SP, SP, SP, SP, SP, SP, ...
## $ city       <fct> SAO PAULO, NA, SAO PAULO, SAO PAULO, SAO PAULO, SAO PAUL...
```

Make state a Factor

```
soro_b <- soro %>%
  mutate(uf = factor(uf))
```

Combine All Munging Ops with Pipe

- Using pipe, we can combine all these operations in 1 big command

```
soro_b <- soro %>%
  mutate(dt_collect = dmy(dt_collect)) %>%
  mutate(analysis = factor(analysis)) %>%
  mutate(analysis = fct_collapse(analysis,
                                 IgG = c("COVID IgG Interp", "IgG, COVID19"),
                                 IgM = c("COVID IgM Interp", "IgM, COVID19"))) %>%
  mutate(result = as.numeric(ifelse(result == "NÃ£o reagente", 0, result))) %>%
  janitor::remove_constant(na.rm = TRUE) %>%  # unit variable
  select(-reference) %>%
  mutate(uf = factor(uf))

glimpse(soro_b)
```

```
## Rows: 200
## Columns: 8
## $ pacid      <chr> "b6d668e4f818f7b3643ed593b8fb902bf9d2501e", "a090625661c...
## $ dt_collect <date> 2020-05-28, 2020-05-11, 2020-06-16, 2020-06-10, 2020-04...
## $ analysis   <fct> IgM, IgG, IgG, IgM, IgM, IgM, IgM, IgG, IgM, IgM, IgM, I...
## $ result     <dbl> 0.74, 0.03, 0.02, NA, 0.47, 0.90, NA, 30.77, 0.41, 0.54,...
## $ sex        <fct> male, female, female, male, female, female, female, male...
## $ birth_yr   <dbl> 1989, 1975, 1997, 2006, 1983, 1963, 1988, 1971, 1968, 19...
## $ uf         <fct> SP, GO, SP, SP, SP, SP, SP, SP, SP, SP, SP, SP, SP, SP, ...
## $ city       <fct> SAO PAULO, NA, SAO PAULO, SAO PAULO, SAO PAULO, SAO PAUL...
```

Section 10

Is `soro_b` Tidyverse "Tidy"?