

# Class Activity 8

Your name here

March 19 2024

## Your turn 1

The following code creates the dataset used in this exercise.

```
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)
)
DBP_wide
# A tibble: 4 x 6
  id    sex  v1.DBP v2.DBP v3.DBP  age
<chr> <chr>  <dbl>  <dbl>  <dbl> <dbl>
1 a      F      88     78     94    23
2 b      M      84     78     82    56
3 c      M     102     96     94    41
4 d      F      70     76     74    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_wide
# A tibble: 4 x 5
  id    sex  SBP_v1 SBP_v2 SBP_v3
<chr> <chr>  <dbl>  <dbl>  <dbl>
1 a      F     130    110    112
2 b      M     120    116    122
3 c      M     130    136    138
4 d      F     119    106    118
```

```
BP_long <- BP_wide %>%
  pivot_longer(names_to = "visit", values_to = "SBP", SBP_v1:SBP_v3) %>%
  mutate(visit = parse_number(visit))
BP_long
# A tibble: 12 x 4
  id    sex  visit  SBP
<chr> <chr>  <dbl> <dbl>
1 a      F      1    130
2 a      F      2    110
3 a      F      3    112
```

|    |   |   |   |     |
|----|---|---|---|-----|
| 4  | b | M | 1 | 120 |
| 5  | b | M | 2 | 116 |
| 6  | b | M | 3 | 122 |
| 7  | c | M | 1 | 130 |
| 8  | c | M | 2 | 136 |
| 9  | c | M | 3 | 138 |
| 10 | d | F | 1 | 119 |
| 11 | d | F | 2 | 106 |
| 12 | d | F | 3 | 118 |

a. Create a long dataframe from `DBP_wide` based on the repeated DBP columns and save it as `DBP_long`.

```
DBP_long <- DBP_wide %>%
  pivot_longer(names_to = "visit",
               values_to = "DBP",
               cols = v1.DBP:v3.DBP)
```

```
DBP_long
# A tibble: 12 x 5
```

|    | id    | sex   | age   | visit  | DBP   |
|----|-------|-------|-------|--------|-------|
|    | <chr> | <chr> | <dbl> | <chr>  | <dbl> |
| 1  | a     | F     | 23    | v1.DBP | 88    |
| 2  | a     | F     | 23    | v2.DBP | 78    |
| 3  | a     | F     | 23    | v3.DBP | 94    |
| 4  | b     | M     | 56    | v1.DBP | 84    |
| 5  | b     | M     | 56    | v2.DBP | 78    |
| 6  | b     | M     | 56    | v3.DBP | 82    |
| 7  | c     | M     | 41    | v1.DBP | 102   |
| 8  | c     | M     | 41    | v2.DBP | 96    |
| 9  | c     | M     | 41    | v3.DBP | 94    |
| 10 | d     | F     | 38    | v1.DBP | 70    |
| 11 | d     | F     | 38    | v2.DBP | 76    |
| 12 | d     | F     | 38    | v3.DBP | 74    |

b. Clean up the visit column of `DBP_long` so that the values are 1, 2, 3, and save it as `DBP_long`.

```
DBP_long <- DBP_long %>%
  mutate(visit = parse_number(visit))
```

```
DBP_long
# A tibble: 12 x 5
```

|    | id    | sex   | age   | visit | DBP   |
|----|-------|-------|-------|-------|-------|
|    | <chr> | <chr> | <dbl> | <dbl> | <dbl> |
| 1  | a     | F     | 23    | 1     | 88    |
| 2  | a     | F     | 23    | 2     | 78    |
| 3  | a     | F     | 23    | 3     | 94    |
| 4  | b     | M     | 56    | 1     | 84    |
| 5  | b     | M     | 56    | 2     | 78    |
| 6  | b     | M     | 56    | 3     | 82    |
| 7  | c     | M     | 41    | 1     | 102   |
| 8  | c     | M     | 41    | 2     | 96    |
| 9  | c     | M     | 41    | 3     | 94    |
| 10 | d     | F     | 38    | 1     | 70    |
| 11 | d     | F     | 38    | 2     | 76    |

|    |   |   |    |   |    |
|----|---|---|----|---|----|
| 12 | d | F | 38 | 3 | 74 |
|----|---|---|----|---|----|

c. Make DBP\_long wide with column names visit.1, visit.2, visit.3 for the DBP values, and save it as DBP\_wide2

```
DBP_wide2 <- DBP_long %>%
  pivot_wider(names_from = "visit",
              values_from = "DBP",
              names_prefix = "visit.")
```

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      |

d. Join DBP\_long with BP\_long2 to create a single data frame with columns id, sex, visit, SBP, DBP, and age. Save this as BP\_both\_long.

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

BP\_both\_long

# A tibble: 12 x 6

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

e. Calculate the mean SBP and DBP for each visit and save the result as mean\_BP\_by\_visit.

```
mean_BP_by_visit <- BP_both_long %>%
  group_by(visit) %>%
  summarize(mean_SBP = mean(SBP),
            mean_DBP = mean(DBP))
```

mean\_BP\_by\_visit

# A tibble: 3 x 3

|   | visit | mean_SBP | mean_DBP |
|---|-------|----------|----------|
|   | <dbl> | <dbl>    | <dbl>    |
| 1 | 1     | 125.     | 86       |
| 2 | 2     | 117      | 82       |
| 3 | 3     | 122.     | 86       |

## Your turn 2

a. Parsing Complex Dates: Use `dmy_hms()` to parse the following date-time string: “25-Dec-2020 17:30:00”

```
parsed_date <- dmy_hms("25-Dec-2020 17:30:00")
parsed_date
[1] "2020-12-25 17:30:00 UTC"
```

b. Advanced Date Arithmetic: Calculate the exact age in years for someone born on “1995-05-15 09:30:00”.

```
dob <- ymd_hms("1995-05-15 09:30:00")
exact_age <- as.duration(interval(dob, now())) / dyears(1)
exact_age
[1] 28.84787
```

c. Creating Date-Time Objects: Create a date-time object for March 15, 2020, 13:30:00 using `make_datetime()`.

```
new_date_time <- make_datetime(2020, 3, 15, 13, 30, 0)
new_date_time
[1] "2020-03-15 13:30:00 UTC"
```

d. Extracting Components from Date-Time Objects: Extract the year, month (as a number), day, hour, and minute from “2022-07-01 14:45:00”.

```
example_date_time <- ymd_hms("2022-07-01 14:45:00")
extracted_components <- tibble(
  year = year(example_date_time),
  month = month(example_date_time),
  day = day(example_date_time),
  hour = hour(example_date_time),
  minute = minute(example_date_time)
)
extracted_components
# A tibble: 1 x 5
  year month   day hour minute
<dbl> <dbl> <int> <int> <int>
1  2022     7     1    14     45
```

e. Advanced Date-Time Arithmetic with Periods: Add 2 months and 15 days to “2021-08-01”.

```
initial_date <- ymd("2021-08-01")
new_date <- initial_date + months(2) + days(15)
new_date
[1] "2021-10-16"
```

f. Duration and Time Differences: Calculate the duration in days, weeks, months, and years between “2019-04-01” and “2022-04-01”.

```
start_date <- ymd("2019-04-01")
end_date <- ymd("2022-04-01")
```

```

time_diff <- end_date - start_date
duration_days <- as.duration(time_diff)
duration_weeks <- duration_days / dweeks(1)
duration_months <- duration_days / dmonths(1)
duration_years <- duration_days / dyears(1)

duration_results <- tibble(
  days = duration_days,
  weeks = duration_weeks,
  months = duration_months,
  years = duration_years
)
duration_results
# A tibble: 1 x 4
  days                weeks months years
  <Duration>          <dbl>  <dbl> <dbl>
1 94694400s (~3 years)  157.    36.0  3.00

```