Class Activity 8

Your name here

2024-04-10

Your turn 1

In the provided R code, we start with two datasets, `DBP_wide` and `BP_wide`, representing blood pressure measurements in a wide format. We then demonstrate how to transform `BP_wide` into a long format using `pivot_longer()`.

```
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 <-


```

```
Error: <text>:4:0: unexpected end of input
2:
3:
  ^
```

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 <-


```

```
Error: <text>:4:0: unexpected end of input
2:
3:
  ^
```

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 <-


```

```
Error: <text>:4:0: unexpected end of input
2:
3:
  ^
```

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 <-


```

```
Error: <text>:4:0: unexpected end of input
2:
3:
  ^
```

e. Calculate the mean SBP and DBP for each visit and save the result as `mean_BP_by_visit`.

```
mean_BP_by_visit <-
```

```
Error: <text>:3:0: unexpected end of input
1: mean_BP_by_visit <-
2:
  ^
```

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 <-
```

```
Error: <text>:3:0: unexpected end of input
1: parsed_date <-
2:
  ^
```

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

```
dob <-


```

```
Error: <text>:4:0: unexpected end of input
2:
3:
  ^
```

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

```
new_date_time <-
```

```
Error: <text>:3:0: unexpected end of input
1: new_date_time <-
2:
  ^
```

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 = ,
  month =  ,
  day =  ,
  hour = ,
  minute =
)
```

```
Error: object '' not found
```

```
extracted_components
```

```
Error in eval(expr, envir, enclos): object 'extracted_components' not found
```

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 <-
```

```
Error: <text>:4:0: unexpected end of input
2: new_date <-
3:
  ^
```

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 <-
duration_weeks <-
duration_months <-
duration_years <-

duration_results <- tibble(
  days = duration_days,
  weeks = duration_weeks,
  months = duration_months,
  years = duration_years
)
```

```
Error: object 'duration_days' not found
```

```
duration_results
```

```
Error in eval(expr, envir, enclos): object 'duration_results' not found
```