Data Transformation in R

Kursus R: Pengenalan dan Praktikal (Sesi 2)

Mohd Azmi Bin Suliman

azmi.suliman@moh.gov.my

Pusat Penyelidikan Penyakit Tak Berjangkit, Institut Kesihatan Umum

Wednesday, 02 October 2024

Data Transformation

- Number
- Factor
- Conditional
- Join and Reshape

Setup: Data Transformation

- Create New Level 1 Header

  → `# Data Transformation`

Number Transformation

Practical: Arithmetic Transformation

- We can perform basic numerical transformation using R
  - → Don't forget to use `mutate` function to create new variable
- For example, convert height from cm to m
  - → Input: Height (`Height`)
  - → Output: Height in meter (`Ht_m`)
  - → Overwrite current dataset
  - → **Hint**: use `.before` or `.after` parameter to arrange the new variable

```
1  asthmads_spss <- asthmads_spss %>%
2    mutate(Ht_m = Height/100, .after = "Height")
```

Practical: Arithmetic Transformation

- For example, convert height in cm (`Height`) cm to height in m (`Ht_m`)

```r
asthmads_spss <- asthmads_spss %>%
  mutate(Ht_m = Height/100, .after = "Height")

asthmads_spss
```

```
# A tibble: 150 × 19
      id idR   Gender   Age WorkStatus Height  Ht_m Weight_Pre WC_Pre PA_HW
   <dbl> <chr> <fct>  <dbl> <fct>       <dbl> <dbl>      <dbl>  <dbl> <dbl>
 1     1 ejGs  Female    34 Unemployed    179  1.79       84.2     77     3
 2     2 A4pG  Male      31 Unemployed    169  1.69       81.8     94     1
 3     3 qkCO  Male      25 Employed      164  1.64       88.5     95     4
 4     4 jcFZ  Female    33 Unemployed    136  1.36       53.2     85     2
 5     5 qVSA  Male      28 Unemployed    172  1.72       71.3     90     3
 6     6 wDAR  Male      33 Unemployed    178  1.78       87.3     92     2
 7     7 FuAU  Female    31 Unemployed    140  1.4        48.8     80     4
 8     8 fnKz  Female    34 Employed      140  1.4        49.1     82     2
 9     9 0YTi  Male      31 Employed      171  1.71       60.1     85     3
10    10 pfMa  Male      28 Employed      163  1.63       93.1    101     5
# ℹ 140 more rows
# ℹ 9 more variables: Weight_Post <dbl>, WC_Post <dbl>, Tx2 <fct>,
#   PEFR_Pre <dbl>, PEFR_Post <dbl>, SxWheeze_Pre <fct>, SxWheeze_Post <fct>,
#   PS_Pre <dbl>, PS_Post <dbl>
```

Practical: Arithmetic Transformation

- Now try yourself!
  - → Calculate BMI_Pre and BMI_Post
  - → **Hint**: Use %>% pipe operator to chain the transformation

Practical: Arithmetic Transformation

- Now try yourself!

  → Calculate BMI_Pre and BMI_Post

```r
asthmads_spss <- asthmads_spss %>%
  mutate(BMI_Pre = Weight_Pre/(Ht_m^2), .after = "Weight_Pre") %>%
  mutate(BMI_Post = Weight_Post/(Ht_m^2), .after = "Weight_Post")

asthmads_spss
```

```
# A tibble: 150 × 21
      id idR   Gender   Age WorkStatus Height  Ht_m Weight_Pre BMI_Pre WC_Pre
   <dbl> <chr> <fct>  <dbl> <fct>       <dbl> <dbl>      <dbl>   <dbl>  <dbl>
 1     1 ejGs  Female    34 Unemployed    179  1.79        84.2    26.3     77
 2     2 A4pG  Male      31 Unemployed    169  1.69        81.8    28.6     94
 3     3 qkCO  Male      25 Employed      164  1.64        88.5    32.9     95
 4     4 jcFZ  Female    33 Unemployed    136  1.36        53.2    28.8     85
 5     5 qVSA  Male      28 Unemployed    172  1.72        71.3    24.1     90
 6     6 wDAR  Male      33 Unemployed    178  1.78        87.3    27.6     92
 7     7 FuAU  Female    31 Unemployed    140  1.4         48.8    24.9     80
 8     8 fnKz  Female    34 Employed      140  1.4         49.1    25.1     82
 9     9 0YTi  Male      31 Employed      171  1.71        60.1    20.6     85
10    10 pfMa  Male      28 Employed      163  1.63        93.1    35.0    101
# ℹ 140 more rows
# ℹ 11 more variables: PA_HW <dbl>, Weight_Post <dbl>, BMI_Post <dbl>,
```

Practical: Rounding

- We can round the numerical variable using `round` function
  - → We can round to nearest decimal point by specifying `digits` parameter

```r
asthmads_spss <- asthmads_spss %>%
  mutate(BMI_Pre = round(BMI_Pre, digits = 2))

asthmads_spss
```

```
# A tibble: 150 × 21
      id idR   Gender   Age WorkStatus Height  Ht_m Weight_Pre BMI_Pre WC_Pre
   <dbl> <chr> <fct>  <dbl> <fct>       <dbl> <dbl>      <dbl>   <dbl>  <dbl>
 1     1 ejGs  Female    34 Unemployed    179  1.79        84.2    26.3     77
 2     2 A4pG  Male      31 Unemployed    169  1.69        81.8    28.6     94
 3     3 qkCO  Male      25 Employed      164  1.64        88.5    32.9     95
 4     4 jcFZ  Female    33 Unemployed    136  1.36        53.2    28.8     85
 5     5 qVSA  Male      28 Unemployed    172  1.72        71.3    24.1     90
 6     6 wDAR  Male      33 Unemployed    178  1.78        87.3    27.6     92
 7     7 FuAU  Female    31 Unemployed    140  1.4         48.8    24.9     80
 8     8 fnKz  Female    34 Employed      140  1.4         49.1    25.0     82
 9     9 0YTi  Male      31 Employed      171  1.71        60.1    20.6     85
10    10 pfMa  Male      28 Employed      163  1.63        93.1    35.0    101
# ℹ 140 more rows
# ℹ 11 more variables: PA_HW <dbl>, Weight_Post <dbl>, BMI_Post <dbl>,
```

Practical: Rounding

- Now try yourself!
  - → Round BMI_Post to 2 decimal points

Practical: Rounding

- Now try yourself!

  → Round BMI_Post to 2 decimal points

```r
asthmads_spss <- asthmads_spss %>%
  mutate(BMI_Post = round(BMI_Post, digits = 2))

asthmads_spss
```

```
# A tibble: 150 × 21
      id idR   Gender   Age WorkStatus Height  Ht_m Weight_Pre BMI_Pre WC_Pre
   <dbl> <chr> <fct>  <dbl> <fct>       <dbl> <dbl>      <dbl>   <dbl>  <dbl>
 1     1 ejGs  Female    34 Unemployed    179  1.79       84.2    26.3     77
 2     2 A4pG  Male      31 Unemployed    169  1.69       81.8    28.6     94
 3     3 qkCO  Male      25 Employed      164  1.64       88.5    32.9     95
 4     4 jcFZ  Female    33 Unemployed    136  1.36       53.2    28.8     85
 5     5 qVSA  Male      28 Unemployed    172  1.72       71.3    24.1     90
 6     6 wDAR  Male      33 Unemployed    178  1.78       87.3    27.6     92
 7     7 FuAU  Female    31 Unemployed    140  1.4        48.8    24.9     80
 8     8 fnKz  Female    34 Employed      140  1.4        49.1    25.0     82
 9     9 0YTi  Male      31 Employed      171  1.71       60.1    20.6     85
10    10 pfMa  Male      28 Employed      163  1.63       93.1    35.0    101
# ℹ 140 more rows
# ℹ 11 more variables: PA_HW <dbl>, Weight_Post <dbl>, BMI_Post <dbl>,
#   WC_Post <dbl>, Tx2 <fct>, PEFR_Pre <dbl>, PEFR_Post <dbl>,
#   SxWheeze_Pre <fct>, SxWheeze_Post <fct>, PS_Pre <dbl>, PS_Post <dbl>
```

Practical: Rounding

- Please note that in R, rounding is done by to the nearest even number
  → aka Banker's rounding

```
1 round(1.5)
```

```
[1] 2
```

```
1 round(2.5)
```

```
[1] 2
```

Practical: Binning (Categorizing Numerical Variable)

- We use `cut` function to categorize numerical variable
  - → For example, we want to categorize BMI_Pre into
    - ⇾ Underweight (< 18.5)
    - ⇾ Normal (18.5 - 22.9)
    - ⇾ Overweight (23 - 24.9)
    - ⇾ Obese (> 25)
  - → We use `breaks` parameter to specify the cut-off points

```
asthmads_spss <- asthmads_spss %>%
  mutate(BMI_PreCat = cut(BMI_Pre,
                          breaks = c(0, 18.49, 22.99, 24.99, 100)),
         .after = "BMI_Pre")

asthmads_spss
```

Practical: Binning (Categorizing Numerical Variable)

- Note: We need to specified small number for the first and large number for the last category

```r
asthmads_spss <- asthmads_spss %>%
  mutate(BMI_PreCat = cut(BMI_Pre,
                          breaks = c(0, 18.49, 22.99, 24.99, 100)),
         .after = "BMI_Pre")

asthmads_spss
```

```
# A tibble: 150 × 22
      id idR   Gender   Age WorkStatus Height  Ht_m Weight_Pre BMI_Pre
   <dbl> <chr> <fct>  <dbl> <fct>       <dbl> <dbl>      <dbl>   <dbl>
 1     1 ejGs  Female    34 Unemployed    179  1.79       84.2    26.3
 2     2 A4pG  Male      31 Unemployed    169  1.69       81.8    28.6
 3     3 qkCO  Male      25 Employed      164  1.64       88.5    32.9
 4     4 jcFZ  Female    33 Unemployed    136  1.36       53.2    28.8
 5     5 qVSA  Male      28 Unemployed    172  1.72       71.3    24.1
 6     6 wDAR  Male      33 Unemployed    178  1.78       87.3    27.6
 7     7 FuAU  Female    31 Unemployed    140  1.4        48.8    24.9
 8     8 fnKz  Female    34 Employed      140  1.4        49.1    25.0
 9     9 0YTi  Male      31 Employed      171  1.71       60.1    20.6
10    10 pfMa  Male      28 Employed      163  1.63       93.1    35.0
# ℹ 140 more rows
# ℹ 13 more variables: BMI_PreCat <fct>, WC_Pre <dbl>, PA_HW <dbl>,
#   Weight_Post <dbl>, BMI_Post <dbl>, WC_Post <dbl>, Tx2 <fct>,
```

```
#    PEFR_Pre <dbl>, PEFR_Post <dbl>, SxWheeze_Pre <fct>, SxWheeze_Post <fct>,
#    PS_Pre <dbl>, PS_Post <dbl>
```

Practical: Binning (Categorizing Numerical Variable)

- We use `cut` function to categorize numerical variable
  - → For example, we want to categorize BMI_Pre into
    - ⇾ Underweight (< 18.5)
    - ⇾ Normal (18.5 - 22.9)
    - ⇾ Overweight (23 - 24.9)
    - ⇾ Obese (> 25)
  - → We use `labels` parameter to label the bin.

```
asthmads_spss <- asthmads_spss %>%
  mutate(BMI_PreCat = cut(BMI_Pre,
                          breaks = c(0, 18.49, 22.99, 24.99, 100),
                          labels = c("Underweight", "Normal", "Overweight", "Ol
         .after = "BMI_Pre")

asthmads_spss
```

Practical: Binning (Categorizing Numerical Variable)

- Note the `labels =` should be one less than `breaks =`

```
asthmads_spss <- asthmads_spss %>%
  mutate(BMI_PreCat = cut(BMI_Pre,
                          breaks = c(0, 18.49, 22.99, 24.99, 100),
                          labels = c("Underweight", "Normal", "Overweight", "Ol
         .after = "BMI_Pre")

asthmads_spss
```

```
# A tibble: 150 × 22
      id idR   Gender   Age WorkStatus Height  Ht_m Weight_Pre BMI_Pre
   <dbl> <chr> <fct>  <dbl> <fct>       <dbl> <dbl>      <dbl>   <dbl>
 1     1 ejGs  Female    34 Unemployed    179  1.79        84.2    26.3
 2     2 A4pG  Male      31 Unemployed    169  1.69        81.8    28.6
 3     3 qkCO  Male      25 Employed      164  1.64        88.5    32.9
 4     4 jcFZ  Female    33 Unemployed    136  1.36        53.2    28.8
 5     5 qVSA  Male      28 Unemployed    172  1.72        71.3    24.1
 6     6 wDAR  Male      33 Unemployed    178  1.78        87.3    27.6
 7     7 FuAU  Female    31 Unemployed    140  1.4         48.8    24.9
 8     8 fnKz  Female    34 Employed      140  1.4         49.1    25.0
 9     9 0YTi  Male      31 Employed      171  1.71        60.1    20.6
10    10 pfMa  Male      28 Employed      163  1.63        93.1    35.0
# ℹ 140 more rows
# ℹ 13 more variables: BMI_PreCat <fct>, WC_Pre <dbl>, PA_HW <dbl>,
#   Weight_Post <dbl>, BMI_Post <dbl>, WC_Post <dbl>, Tx2 <fct>,
```

Practical: Binning

- Now try yourself!
  - → Categorized BMI_Post into the same category as BMI_Pre

Practical: Binning

- Now try yourself!

  → Categorized BMI_Post into the same category as BMI_Pre

```r
asthmads_spss <- asthmads_spss %>%
  mutate(BMI_PostCat = cut(BMI_Post,
                           breaks = c(0, 18.49, 22.99, 24.99, 100),
                           labels = c("Underweight", "Normal", "Overweight", "Ol
         .after = "BMI_Post")

asthmads_spss
```

```
# A tibble: 150 × 23
      id idR   Gender   Age WorkStatus Height  Ht_m Weight_Pre BMI_Pre
   <dbl> <chr> <fct>  <dbl> <fct>       <dbl> <dbl>      <dbl>   <dbl>
 1     1 ejGs  Female    34 Unemployed    179  1.79        84.2    26.3
 2     2 A4pG  Male      31 Unemployed    169  1.69        81.8    28.6
 3     3 qkCO  Male      25 Employed      164  1.64        88.5    32.9
 4     4 jcFZ  Female    33 Unemployed    136  1.36        53.2    28.8
 5     5 qVSA  Male      28 Unemployed    172  1.72        71.3    24.1
 6     6 wDAR  Male      33 Unemployed    178  1.78        87.3    27.6
 7     7 FuAU  Female    31 Unemployed    140  1.4         48.8    24.9
 8     8 fnKz  Female    34 Employed      140  1.4         49.1    25.0
 9     9 0YTi  Male      31 Employed      171  1.71        60.1    20.6
10    10 pfMa  Male      28 Employed      163  1.63        93.1    35.0
# ℹ 140 more rows
```

```
# ℹ 14 more variables: BMI_PreCat <fct>, WC_Pre <dbl>, PA_HW <dbl>,
#   Weight_Post <dbl>, BMI_Post <dbl>, BMI_PostCat <fct>, WC_Post <dbl>,
#   Tx2 <fct>, PEFR_Pre <dbl>, PEFR_Post <dbl>, SxWheeze_Pre <fct>,
#   SxWheeze_Post <fct>, PS_Pre <dbl>, PS_Post <dbl>
```

Factor Transformation

Practical: Modifying Factor Order

- We can modify the order of factor using `fct_relevel` function
  - → For example, we want to change the order of `Gender`
  - → First we check the current order of `Gender` using `levels` function

```
1 levels(asthmads_spss$Gender)
```

```
[1] "Male"   "Female"
```

Practical: Modifying Factor Order

- We can modify the order of factor using `fct_relevel` function
  - → For example, we want to change the order of `Gender`
  - → We can either write the full order, or the first order

```r
asthmads_spss <- asthmads_spss %>%
  mutate(Gender_F = fct_relevel(Gender, "Female", "Male"),
         .after = "Gender")

asthmads_spss
```

```
# A tibble: 150 × 24
      id idR   Gender Gender_F   Age WorkStatus Height  Ht_m Weight_Pre BMI_Pre
   <dbl> <chr> <fct>  <fct>    <dbl> <fct>       <dbl> <dbl>      <dbl>   <dbl>
 1     1 ejGs  Female Female      34 Unemployed    179  1.79       84.2    26.3
 2     2 A4pG  Male   Male        31 Unemployed    169  1.69       81.8    28.6
 3     3 qkCO  Male   Male        25 Employed      164  1.64       88.5    32.9
 4     4 jcFZ  Female Female      33 Unemployed    136  1.36       53.2    28.8
 5     5 qVSA  Male   Male        28 Unemployed    172  1.72       71.3    24.1
 6     6 wDAR  Male   Male        33 Unemployed    178  1.78       87.3    27.6
 7     7 FuAU  Female Female      31 Unemployed    140  1.4        48.8    24.9
 8     8 fnKz  Female Female      34 Employed      140  1.4        49.1    25.0
 9     9 0YTi  Male   Male        31 Employed      171  1.71       60.1    20.6
10    10 pfMa  Male   Male        28 Employed      163  1.63       93.1    35.0
# ℹ 140 more rows
```

```
# ℹ 14 more variables: BMI_PreCat <fct>, WC_Pre <dbl>, PA_HW <dbl>,
#   Weight_Post <dbl>, BMI_Post <dbl>, BMI_PostCat <fct>, WC_Post <dbl>,
#   Tx2 <fct>, PEFR_Pre <dbl>, PEFR_Post <dbl>, SxWheeze_Pre <fct>,
#   SxWheeze_Post <fct>, PS_Pre <dbl>, PS_Post <dbl>
```

Practical: Modifying Factor Order

- We can modify the order of factor using `fct_relevel` function
  - → For example, we want to change the order of `Gender`
  - → We can either write the full order, or the first order

```
asthmads_spss <- asthmads_spss %>%
  mutate(Gender_F = fct_relevel(Gender, "Female"),
         .after = "Gender")

asthmads_spss
```

```
# A tibble: 150 × 24
      id idR   Gender Gender_F   Age WorkStatus Height  Ht_m Weight_Pre BMI_Pre
   <dbl> <chr> <fct>  <fct>    <dbl> <fct>       <dbl> <dbl>      <dbl>   <dbl>
 1     1 ejGs  Female Female      34 Unemployed    179  1.79       84.2    26.3
 2     2 A4pG  Male   Male        31 Unemployed    169  1.69       81.8    28.6
 3     3 qkCO  Male   Male        25 Employed      164  1.64       88.5    32.9
 4     4 jcFZ  Female Female      33 Unemployed    136  1.36       53.2    28.8
 5     5 qVSA  Male   Male        28 Unemployed    172  1.72       71.3    24.1
 6     6 wDAR  Male   Male        33 Unemployed    178  1.78       87.3    27.6
 7     7 FuAU  Female Female      31 Unemployed    140  1.4        48.8    24.9
 8     8 fnKz  Female Female      34 Employed      140  1.4        49.1    25.0
 9     9 0YTi  Male   Male        31 Employed      171  1.71       60.1    20.6
10    10 pfMa  Male   Male        28 Employed      163  1.63       93.1    35.0
# ℹ 140 more rows
```

```
# ℹ 14 more variables: BMI_PreCat <fct>, WC_Pre <dbl>, PA_HW <dbl>,
#   Weight_Post <dbl>, BMI_Post <dbl>, BMI_PostCat <fct>, WC_Post <dbl>,
#   Tx2 <fct>, PEFR_Pre <dbl>, PEFR_Post <dbl>, SxWheeze_Pre <fct>,
#   SxWheeze_Post <fct>, PS_Pre <dbl>, PS_Post <dbl>
```

Practical: Modifying Factor Order

- We can modify the order of factor using `fct_relevel` function
  - → For example, we want to change the order of `Gender`

```
levels(asthmads_spss$Gender)
```

```
[1] "Male"   "Female"
```

```
levels(asthmads_spss$Gender_F)
```

```
[1] "Female" "Male"
```

Practical: Recode Factor

- We can also recode factor using `fct_recode` function
  - → For example, we want to change Treatment Group (`Tx2`) to Intervention Group (`Tx1`)
    - ⇾ `Placebo` to `Control`
    - ⇾ `Drug A` and `Drug B` to `Intervention`
  - → Note: Recode factor does not change the order of the factor

```
asthmads_spss <- asthmads_spss %>%
  mutate(Tx1 = fct_recode(Tx2,
                          "Control" = "Placebo",
                          "Intervention" = "Drug A",
                          "Intervention" = "Drug B"),
         .before = Tx2)

asthmads_spss
```

Practical: Recode Factor

```r
asthmads_spss <- asthmads_spss %>%
  mutate(Tx1 = fct_recode(Tx2,
                          "Control" = "Placebo",
                          "Intervention" = "Drug A",
                          "Intervention" = "Drug B"),
         .before = Tx2)

asthmads_spss
```

```
# A tibble: 150 × 25
      id idR   Gender Gender_F   Age WorkStatus Height  Ht_m Weight_Pre BMI_Pre
   <dbl> <chr> <fct>  <fct>    <dbl> <fct>       <dbl> <dbl>      <dbl>   <dbl>
 1     1 ejGs  Female Female      34 Unemployed    179  1.79       84.2    26.3
 2     2 A4pG  Male   Male        31 Unemployed    169  1.69       81.8    28.6
 3     3 qkCO  Male   Male        25 Employed      164  1.64       88.5    32.9
 4     4 jcFZ  Female Female      33 Unemployed    136  1.36       53.2    28.8
 5     5 qVSA  Male   Male        28 Unemployed    172  1.72       71.3    24.1
 6     6 wDAR  Male   Male        33 Unemployed    178  1.78       87.3    27.6
 7     7 FuAU  Female Female      31 Unemployed    140  1.4        48.8    24.9
 8     8 fnKz  Female Female      34 Employed      140  1.4        49.1    25.0
 9     9 0YTi  Male   Male        31 Employed      171  1.71       60.1    20.6
10    10 pfMa  Male   Male        28 Employed      163  1.63       93.1    35.0
# ℹ 140 more rows
# ℹ 15 more variables: BMI_PreCat <fct>, WC_Pre <dbl>, PA_HW <dbl>,
#   Weight_Post <dbl>, BMI_Post <dbl>, BMI_PostCat <fct>, WC_Post <dbl>,
#   Tx1 <fct>, Tx2 <fct>, PEFR_Pre <dbl>, PEFR_Post <dbl>, SxWheeze_Pre <fct>,
#   SxWheeze_Post <fct>, PS_Pre <dbl>, PS_Post <dbl>
```

Practical: Recode Factor

- We can also recode factor using `fct_recode` function

  → can also be use to collapse the factor

```
asthmads_spss <- asthmads_spss %>%
  mutate(Obese_PreCat = fct_recode(BMI_PreCat,
                                   "Non-Obese" = "Underweight",
                                   "Non-Obese" = "Normal",
                                   "Non-Obese" = "Overweight"),
         .after = "BMI_PreCat")

asthmads_spss
```

Practical: Recode Factor

```r
asthmads_spss <- asthmads_spss %>%
  mutate(Obese_PreCat = fct_recode(BMI_PreCat,
                                   "Non-Obese" = "Underweight",
                                   "Non-Obese" = "Normal",
                                   "Non-Obese" = "Overweight"),
         .after = "BMI_PreCat")

asthmads_spss
```

```
# A tibble: 150 × 26
      id idR   Gender Gender_F   Age WorkStatus Height  Ht_m Weight_Pre BMI_Pre
   <dbl> <chr> <fct>  <fct>    <dbl> <fct>       <dbl> <dbl>      <dbl>   <dbl>
 1     1 ejGs  Female Female      34 Unemployed    179  1.79       84.2    26.3
 2     2 A4pG  Male   Male        31 Unemployed    169  1.69       81.8    28.6
 3     3 qkCO  Male   Male        25 Employed      164  1.64       88.5    32.9
 4     4 jcFZ  Female Female      33 Unemployed    136  1.36       53.2    28.8
 5     5 qVSA  Male   Male        28 Unemployed    172  1.72       71.3    24.1
 6     6 wDAR  Male   Male        33 Unemployed    178  1.78       87.3    27.6
 7     7 FuAU  Female Female      31 Unemployed    140  1.4        48.8    24.9
 8     8 fnKz  Female Female      34 Employed      140  1.4        49.1    25.0
 9     9 0YTi  Male   Male        31 Employed      171  1.71       60.1    20.6
10    10 pfMa  Male   Male        28 Employed      163  1.63       93.1    35.0
# ℹ 140 more rows
# ℹ 16 more variables: BMI_PreCat <fct>, Obese_PreCat <fct>, WC_Pre <dbl>,
#   PA_HW <dbl>, Weight_Post <dbl>, BMI_Post <dbl>, BMI_PostCat <fct>,
#   WC_Post <dbl>, Tx1 <fct>, Tx2 <fct>, PEFR_Pre <dbl>, PEFR_Post <dbl>,
#   SxWheeze_Pre <fct>, SxWheeze_Post <fct>, PS_Pre <dbl>, PS_Post <dbl>
```

Practical: Collapse Factor

- In previous example, we use `fct_recode` to collapse the factor
- Collapsing factor have their own function, i.e., `fct_collapse`

```
asthmads_spss <- asthmads_spss %>%
  mutate(Obese_PostCat = fct_collapse(BMI_PostCat,
                                      "Non-Obese" = c("Underweight",
                                                      "Normal",
                                                      "Overweight")),
         .after = "BMI_PostCat")

asthmads_spss
```

Practical: Collapse Factor

```r
asthmads_spss <- asthmads_spss %>%
  mutate(Obese_PostCat = fct_collapse(BMI_PostCat,
                                      "Non-Obese" = c("Underweight",
                                                      "Normal",
                                                      "Overweight")),
         .after = "BMI_PostCat")

asthmads_spss
```

```
# A tibble: 150 × 27
      id idR   Gender Gender_F   Age WorkStatus Height  Ht_m Weight_Pre BMI_Pre
   <dbl> <chr> <fct>  <fct>    <dbl> <fct>       <dbl> <dbl>      <dbl>   <dbl>
 1     1 ejGs  Female Female      34 Unemployed    179  1.79       84.2    26.3
 2     2 A4pG  Male   Male        31 Unemployed    169  1.69       81.8    28.6
 3     3 qkCO  Male   Male        25 Employed      164  1.64       88.5    32.9
 4     4 jcFZ  Female Female      33 Unemployed    136  1.36       53.2    28.8
 5     5 qVSA  Male   Male        28 Unemployed    172  1.72       71.3    24.1
 6     6 wDAR  Male   Male        33 Unemployed    178  1.78       87.3    27.6
 7     7 FuAU  Female Female      31 Unemployed    140  1.4        48.8    24.9
 8     8 fnKz  Female Female      34 Employed      140  1.4        49.1    25.0
 9     9 0YTi  Male   Male        31 Employed      171  1.71       60.1    20.6
10    10 pfMa  Male   Male        28 Employed      163  1.63       93.1    35.0
# ℹ 140 more rows
# ℹ 17 more variables: BMI_PreCat <fct>, Obese_PreCat <fct>, WC_Pre <dbl>,
#   PA_HW <dbl>, Weight_Post <dbl>, BMI_Post <dbl>, BMI_PostCat <fct>,
#   Obese_PostCat <fct>, WC_Post <dbl>, Tx1 <fct>, Tx2 <fct>, PEFR_Pre <dbl>,
```

Conditional Transformation

Conditional Transformation

- Sometimes, we want to create new variable based on certain condition.
- Commonly, two functions are used for conditional transformation:
  - → `if_else()`
  - → `case_when()`

Practical: `if_else()` Conditional Transformation

- For simple conditional transformation, `if_else()` function is preferable
- `if_else()` function also were common in other software like Excel, SPSS, etc.

```
asthmads_spss <- asthmads_spss %>%
  mutate(Obese_Pre = if_else(BMI_PreCat == "Obese", "Yes", "No"),
         .after = "BMI_PreCat")

asthmads_spss
```

Practical: `if_else()` Conditional Transformation

```r
asthmads_spss <- asthmads_spss %>%
  mutate(Obese_Pre = if_else(BMI_PreCat == "Obese", "Yes", "No"),
         .after = "BMI_PreCat")

asthmads_spss
```

```
# A tibble: 150 × 28
      id idR   Gender Gender_F   Age WorkStatus Height  Ht_m Weight_Pre BMI_Pre
   <dbl> <chr> <fct>  <fct>    <dbl> <fct>       <dbl> <dbl>      <dbl>   <dbl>
 1     1 ejGs  Female Female      34 Unemployed    179  1.79       84.2    26.3
 2     2 A4pG  Male   Male        31 Unemployed    169  1.69       81.8    28.6
 3     3 qkCO  Male   Male        25 Employed      164  1.64       88.5    32.9
 4     4 jcFZ  Female Female      33 Unemployed    136  1.36       53.2    28.8
 5     5 qVSA  Male   Male        28 Unemployed    172  1.72       71.3    24.1
 6     6 wDAR  Male   Male        33 Unemployed    178  1.78       87.3    27.6
 7     7 FuAU  Female Female      31 Unemployed    140  1.4        48.8    24.9
 8     8 fnKz  Female Female      34 Employed      140  1.4        49.1    25.0
 9     9 0YTi  Male   Male        31 Employed      171  1.71       60.1    20.6
10    10 pfMa  Male   Male        28 Employed      163  1.63       93.1    35.0
# ℹ 140 more rows
# ℹ 18 more variables: BMI_PreCat <fct>, Obese_Pre <chr>, Obese_PreCat <fct>,
#   WC_Pre <dbl>, PA_HW <dbl>, Weight_Post <dbl>, BMI_Post <dbl>,
#   BMI_PostCat <fct>, Obese_PostCat <fct>, WC_Post <dbl>, Tx1 <fct>,
#   Tx2 <fct>, PEFR_Pre <dbl>, PEFR_Post <dbl>, SxWheeze_Pre <fct>,
#   SxWheeze_Post <fct>, PS_Pre <dbl>, PS_Post <dbl>
```

Practical: `case_when()` Conditional Transformation

- For more complex conditional transformation, `case_when()` function is preferable
  - → For example, nested `if_else()` in `if_else()` function
- In this example, we want to categorise abdominal obesity (`Abd_Obese`), based on this criteria:
  - → If Male and Waist Circumference >= 90 cm: `Abdominal Obese`
  - → If Male and Waist Circumference < 90 cm: `No Abdominal Obese`
  - → If Female and Waist Circumference >= 80 cm: `Abdominal Obese`
  - → If Female and Waist Circumference < 80 cm: `No Abdominal Obese`

Practical: `case_when()` Conditional Transformation

```r
asthmads_spss <- asthmads_spss %>%
  mutate(AbdObese_Pre = case_when(
    WC_Pre >= 90 & Gender == "Male" ~ "Abdominal Obese",
    WC_Pre < 90 & Gender == "Male" ~ "No Abdominal Obese",
    WC_Pre >= 80 & Gender == "Female" ~ "Abdominal Obese",
    WC_Pre < 80 & Gender == "Female" ~ "No Abdominal Obese"
    ),
    .after = "WC_Pre")

asthmads_spss
```

```
# A tibble: 150 × 29
      id idR   Gender Gender_F   Age WorkStatus Height  Ht_m Weight_Pre BMI_Pre
   <dbl> <chr> <fct>  <fct>    <dbl> <fct>       <dbl> <dbl>      <dbl>   <dbl>
 1     1 ejGs  Female Female      34 Unemployed    179  1.79       84.2    26.3
 2     2 A4pG  Male   Male        31 Unemployed    169  1.69       81.8    28.6
 3     3 qkCO  Male   Male        25 Employed      164  1.64       88.5    32.9
 4     4 jcFZ  Female Female      33 Unemployed    136  1.36       53.2    28.8
 5     5 qVSA  Male   Male        28 Unemployed    172  1.72       71.3    24.1
 6     6 wDAR  Male   Male        33 Unemployed    178  1.78       87.3    27.6
 7     7 FuAU  Female Female      31 Unemployed    140  1.4        48.8    24.9
 8     8 fnKz  Female Female      34 Employed      140  1.4        49.1    25.0
 9     9 0YTi  Male   Male        31 Employed      171  1.71       60.1    20.6
10    10 pfMa  Male   Male        28 Employed      163  1.63       93.1    35.0
# ℹ 140 more rows
# ℹ 19 more variables: BMI_PreCat <fct>, Obese_Pre <chr>, Obese_PreCat <fct>,
```

```
#   WC_Pre <dbl>, AbdObese_Pre <chr>, PA_HW <dbl>, Weight_Post <dbl>,
#   BMI_Post <dbl>, BMI_PostCat <fct>, Obese_PostCat <fct>, WC_Post <dbl>,
#   Tx1 <fct>, Tx2 <fct>, PEFR_Pre <dbl>, PEFR_Post <dbl>, SxWheeze_Pre <fct>,
#   SxWheeze_Post <fct>, PS_Pre <dbl>, PS_Post <dbl>
```

Join and Reshape

- `inner_join()`
- `left_join()`
- `right_join()`
- `full_join()`
- `pivot_longer()`
- `pivot_wider()`

Join Dataset

- Occasionally we have two dataset that correspond to same individual
  - → e.g., pre and post intervention
  - → e.g., demographic and clinical data
- We can join the dataset using `*_join()` function
  - → `inner_join()`, `left_join()`, `right_join()`, `full_join()`
  - → matching key is required, e.g., id
- Add another Level 1 Header
  - → `## Join Dataset`

Practical: Join Dataset

- In this example, we have two SPSS dataset.
- We need to import it.

```
asthmads_pre <- read_sav("Dataset/asthmads_pre.sav") %>%
  as_factor()

asthmads_post <- read_sav("Dataset/asthmads_post.sav") %>%
  as_factor()
```

Practical: Join Dataset

- In the `asthmads_pre` dataset, we have baseline measurement and identifier `idR`

```
1 asthmads_pre
```

```
# A tibble: 150 × 12
      id idR   Gender   Age WorkStatus Height Weight_Pre WC_Pre Tx2     PEFR_Pre
   <dbl> <chr> <fct>  <dbl> <fct>       <dbl>      <dbl>  <dbl> <fct>      <dbl>
 1     1 ejGs  Female    34 Unemployed    179       84.2     77 Placebo      397
 2     2 A4pG  Male      31 Unemployed    169       81.8     94 Placebo      472
 3     3 qkCO  Male      25 Employed      164       88.5     95 Placebo      476
 4     4 jcFZ  Female    33 Unemployed    136       53.2     85 Placebo      416
 5     5 qVSA  Male      28 Unemployed    172       71.3     90 Drug A       452
 6     6 wDAR  Male      33 Unemployed    178       87.3     92 Placebo      484
 7     7 FuAU  Female    31 Unemployed    140       48.8     80 Drug A       366
 8     8 fnKz  Female    34 Employed      140       49.1     82 Placebo      435
 9     9 0YTi  Male      31 Employed      171       60.1     85 Placebo      425
10    10 pfMa  Male      28 Employed      163       93.1    101 Placebo      437
# ℹ 140 more rows
# ℹ 2 more variables: SxWheeze_Pre <fct>, PS_Pre <dbl>
```

Practical: Join Dataset

- In the `asthmads_post` dataset, we have missing baseline measurement
- But we have post measurment and identifier `idR`

```
1 asthmads_post
```

```
# A tibble: 150 × 5
      id idR   PEFR_Post SxWheeze_Post PS_Post
   <dbl> <chr>     <dbl> <fct>            <dbl>
 1     1 ejGs        355 No                   5
 2     2 A4pG        445 Yes                  2
 3     3 qkCO        481 Yes                  3
 4     4 jcFZ        382 No                   5
 5     5 qVSA        475 No                   3
 6     6 wDAR        497 No                   4
 7     7 FuAU        336 No                   3
 8     8 fnKz        413 No                   3
 9     9 0YTi        434 No                   4
10    10 pfMa        413 No                   3
# ℹ 140 more rows
```

Practical: Join Dataset

- We can use `*_join()` function to join the dataset
  - → The function can automatically detect the matching key
  - → However, preferably we specify the matching key using `by` parameter

```r
asthmads_join <- left_join(asthmads_pre, asthmads_post, by = "idR")

asthmads_join
```

```
# A tibble: 150 × 16
    id.x idR   Gender   Age WorkStatus Height Weight_Pre WC_Pre Tx2     PEFR_Pre
   <dbl> <chr> <fct>  <dbl> <fct>       <dbl>      <dbl>  <dbl> <fct>      <dbl>
 1     1 ejGs  Female    34 Unemployed    179       84.2     77 Placebo      397
 2     2 A4pG  Male      31 Unemployed    169       81.8     94 Placebo      472
 3     3 qkCO  Male      25 Employed      164       88.5     95 Placebo      476
 4     4 jcFZ  Female    33 Unemployed    136       53.2     85 Placebo      416
 5     5 qVSA  Male      28 Unemployed    172       71.3     90 Drug A       452
 6     6 wDAR  Male      33 Unemployed    178       87.3     92 Placebo      484
 7     7 FuAU  Female    31 Unemployed    140       48.8     80 Drug A       366
 8     8 fnKz  Female    34 Employed      140       49.1     82 Placebo      435
 9     9 0YTi  Male      31 Employed      171       60.1     85 Placebo      425
10    10 pfMa  Male      28 Employed      163       93.1    101 Placebo      437
# ℹ 140 more rows
```

Reshape Dataset

- Sometimes, we want to reshape the dataset
  - → From wide to long
  - → From long to wide
- For example, certain analysis only accept long format
- Add another Level 1 Header
  - → `## Reshape Dataset`

Practical: Reshape Dataset

- In this example, we have wide dataset with `idR`, `PEFR_Pre`, and `PEFR_Post`

```
asthma_wide <- asthmads_join %>%
  select(idR:Tx2, PEFR_Pre, PEFR_Post)

asthma_wide
```

```
# A tibble: 150 × 10
   idR   Gender   Age WorkStatus Height Weight_Pre WC_Pre Tx2     PEFR_Pre
   <chr> <fct>  <dbl> <fct>       <dbl>      <dbl>  <dbl> <fct>      <dbl>
 1 ejGs  Female    34 Unemployed    179       84.2     77 Placebo      397
 2 A4pG  Male      31 Unemployed    169       81.8     94 Placebo      472
 3 qkCO  Male      25 Employed      164       88.5     95 Placebo      476
 4 jcFZ  Female    33 Unemployed    136       53.2     85 Placebo      416
 5 qVSA  Male      28 Unemployed    172       71.3     90 Drug A       452
 6 wDAR  Male      33 Unemployed    178       87.3     92 Placebo      484
 7 FuAU  Female    31 Unemployed    140       48.8     80 Drug A       366
 8 fnKz  Female    34 Employed      140       49.1     82 Placebo      435
 9 0YTi  Male      31 Employed      171       60.1     85 Placebo      425
10 pfMa  Male      28 Employed      163       93.1    101 Placebo      437
# ℹ 140 more rows
# ℹ 1 more variable: PEFR_Post <dbl>
```

Practical: Reshape Dataset

- We can use `pivot_longer()` function to reshape the dataset
  - → We specify the `cols` parameter to specify the columns to be reshaped
  - → We specify the `names_to` parameter to specify the new column name for the reshaped columns
  - → We specify the `values_to` parameter to specify the new column name for the values of the reshaped columns

```
asthma_long <- asthma_wide %>%
  pivot_longer(cols = c(PEFR_Pre, PEFR_Post),
               names_to = "Time",
               names_prefix = "PEFR_",
               values_to = "PEFR")

asthma_long
```

Practical: Reshape Dataset

```r
asthma_long <- asthma_wide %>%
  pivot_longer(cols = c(PEFR_Pre, PEFR_Post),
               names_to = "Time",
               names_prefix = "PEFR_",
               values_to = "PEFR")

asthma_long
```

```
# A tibble: 300 × 10
   idR   Gender   Age WorkStatus Height Weight_Pre WC_Pre Tx2     Time   PEFR
   <chr> <fct>  <dbl> <fct>       <dbl>      <dbl>  <dbl> <fct>   <chr> <dbl>
 1 ejGs  Female    34 Unemployed    179       84.2     77 Placebo Pre     397
 2 ejGs  Female    34 Unemployed    179       84.2     77 Placebo Post    355
 3 A4pG  Male      31 Unemployed    169       81.8     94 Placebo Pre     472
 4 A4pG  Male      31 Unemployed    169       81.8     94 Placebo Post    445
 5 qkCO  Male      25 Employed      164       88.5     95 Placebo Pre     476
 6 qkCO  Male      25 Employed      164       88.5     95 Placebo Post    481
 7 jcFZ  Female    33 Unemployed    136       53.2     85 Placebo Pre     416
 8 jcFZ  Female    33 Unemployed    136       53.2     85 Placebo Post    382
 9 qVSA  Male      28 Unemployed    172       71.3     90 Drug A  Pre     452
10 qVSA  Male      28 Unemployed    172       71.3     90 Drug A  Post    475
# ℹ 290 more rows
```