Using the across function from dplyr package

Alierwai Reng

Table of contents

1 Getting Started

1.1 Load the required libraries

```
library(tidyverse)
```

1.2 Load data

```
# Set the seed for reproducibility; create a same student-grade dataset
set.seed(254)
student_grades <-
    tibble(
        name = c(
            "Ayen", "Deng", "Akuien",
            "Atong", "Tut", "Garang",
            "Wichar", "Nyikuoth"
            ),
        english = rnorm(n = 8, mean = 85, sd = 15),
        mathematics = rnorm(n = 8, mean = 82, sd = 12.5),
        statistics = rnorm(n = 8, mean = 89, sd = 10.5),
        data_science = rnorm(n = 8, mean = 78, sd = 14)
    )

# Display output
student_grades |>
    knitr::kable()
```

name	english	mathematics	statistics	data_science
Ayen	81.12773	57.56785	81.18913	46.89237
Deng	108.97947	86.99167	102.00057	63.40586
Akuien	87.02066	76.42730	93.02909	77.34297
Atong	77.77471	88.91984	99.17388	68.38323
Tut	100.30413	102.50166	105.14887	55.17427
Garang	89.45313	78.36310	80.02635	58.06276
Wichar	83.07781	80.40586	114.14167	103.17680
Nyikuoth	109.16066	89.24462	97.66874	71.84307

2 Transform the data and select two highest and two lowest grades for each student

```
# Trim grades above 100 and round to 2 decimal places
final_grades <-
    student_grades |>
    mutate(
        across(where(is.numeric), \(x) if_else(x > 100, 100, round(x, 1)))
    )

final_grades|>
    knitr::kable()
```

name	english	mathematics	statistics	data_science
Ayen	81.1	57.6	81.2	46.9
Deng	100.0	87.0	100.0	63.4
Akuien	87.0	76.4	93.0	77.3
Atong	77.8	88.9	99.2	68.4
Tut	100.0	100.0	100.0	55.2

name	english	mathematics	statistics	data_science
Garang	89.5	78.4	80.0	58.1
Wichar	83.1	80.4	100.0	100.0
Nyikuoth	100.0	89.2	97.7	71.8

```r
# Compute the top 2 best grades
top_2_scores <-
    final_grades |>
    pivot_longer(
        where(is.numeric),
        names_to = "subject",
        values_to = "grade"
    ) |>
    # Select the two highest grades for each student; ties are retained by default
    slice_max(order_by = grade, n = 2, by = name, with_ties = TRUE)

top_2_scores
```

```
# A tibble: 17 x 3
   name     subject      grade
   <chr>    <chr>        <dbl>
 1 Ayen     statistics    81.2
 2 Ayen     english       81.1
 3 Deng     english      100  
 4 Deng     statistics   100  
 5 Akuien   statistics    93  
 6 Akuien   english       87  
 7 Atong    statistics    99.2
 8 Atong    mathematics   88.9
 9 Tut      english      100  
10 Tut      mathematics  100  
11 Tut      statistics   100  
12 Garang   english       89.5
13 Garang   statistics    80  
14 Wichar   statistics   100  
15 Wichar   data_science 100  
16 Nyikuoth english      100  
17 Nyikuoth statistics    97.7
```

```r
# Compute the bottom 2 worst grades
bottom_2_scores <-
    final_grades |>
    pivot_longer(
        where(is.numeric),
        names_to = "subject",
        values_to = "grade"
    ) |>
    # Select the two lowest grades for each student; ties are retained by default
    slice_min(order_by = grade, n = 2, by = name, with_ties = TRUE)

bottom_2_scores
```

```
# A tibble: 18 x 3
   name     subject      grade
   <chr>    <chr>        <dbl>
 1 Ayen     data_science  46.9
 2 Ayen     mathematics   57.6
 3 Deng     data_science  63.4
 4 Deng     mathematics   87
 5 Akuien   mathematics   76.4
 6 Akuien   data_science  77.3
 7 Atong    data_science  68.4
 8 Atong    english       77.8
 9 Tut      data_science  55.2
10 Tut      english      100
11 Tut      mathematics  100
12 Tut      statistics   100
13 Garang   data_science  58.1
14 Garang   mathematics   78.4
15 Wichar   mathematics   80.4
16 Wichar   english       83.1
17 Nyikuoth data_science  71.8
18 Nyikuoth mathematics   89.2
```

```r
# Import multiple Excel files into R
library(readxl)

paths <- list.files("../00-data/multiple_excel_files", pattern = "[.]xlsx$", full.names = TRU

census <-
    paths |>
    set_names(basename) |>
    map(\(path) read_excel(path)) |>
    list_rbind(names_to = 'state') |>
    # mutate(state = str_remove_all(state, '.xlsx')) |>
    separate_wider_delim(
        state,
        delim = '.',
        names = c('state', NA)
    ) |>
    mutate(state = str_replace_all(state, '_', ' ') |> str_to_title()) |>
    janitor::clean_names() |>
    select(
        former_region ,
        state,
        state2 = region_name,
        gender = variable_name,
        age_category = age_name,
        population = x2008
        ) |>
    separate_wider_delim(
        gender,
        delim = ' ',
        names = c(NA, 'gender', NA)
    ) |>
    filter(gender != 'Total', age_category != 'Total') |>
    mutate(
        age_category = case_when(
            age_category %in% c("0 to 4", "5 to 9", "10 to 14") ~ "0-14",
            age_category %in% c("15 to 19", "20 to 24")         ~ "15-24",
            age_category %in% c("25 to 29", "30 to 34")         ~ "25-34",
            age_category %in% c("35 to 39", "40 to 44")         ~ "35-44",
            age_category %in% c("45 to 49", "50 to 54")         ~ "45-54",
            age_category %in% c("55 to 59", "60 to 64")         ~ "55-64",
            .default = "65+"
        )
    ) |>
    summarize(
        total = sum(population, na.rm = TRUE),
        .by = c(former_region, state2, gender, age_category)
    )

# Inspect output
```

```
census
```

```
# A tibble: 140 x 5
   former_region state2            gender age_category  total
   <chr>         <chr>             <chr>  <chr>         <dbl>
 1 <NA>          Central Equatoria Male   0-14         242247
 2 <NA>          Central Equatoria Male   15-24        124513
 3 <NA>          Central Equatoria Male   25-34         95507
 4 <NA>          Central Equatoria Male   35-44         59775
 5 <NA>          Central Equatoria Male   45-54         32567
 6 <NA>          Central Equatoria Male   55-64         15704
 7 <NA>          Central Equatoria Male   65+           11409
 8 <NA>          Central Equatoria Female 0-14         221216
 9 <NA>          Central Equatoria Female 15-24        115726
10 <NA>          Central Equatoria Female 25-34         86092
# i 130 more rows
```