session 3 code

2024-09-23

SESSION 3 [column calculation/manipulation]

For this training session we will go over the case_when(), rowSum() and best practice when manipulating columns.

The code below is from session 1. We are standarizing the columns with janitor::clean_names() and renaming the columns.

session 1 code

```
# libraries
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
# import data
df_csv <- read.csv(paste0(getwd(), "/example_data.csv"))

# standarize col names
df_csv_clean_names <- df_csv %>% clean_names()

# rename col names
df_csv_new_column_names <- df_csv_clean_names %>%
```

```r
  rename("ethnicity" = "hispanic", "exercise" = "excerise")

# reorder col names
df_csv_select <- df_csv_new_column_names %>%
  select(dob, ethnicity, race, sex, zip_code, insurance, exercise, everything())

# rename object for clarity
df_csv_formating <- df_csv_select
```

session 1 code altnerative w/ pipe operator