Draft Quarto document

AUTHOR
Mikkel Alnor

PUBLISHED
May 7, 2025

Setup chunk

Importing DIME data

Importing CGM data manually

```r
cgm_101 <- here("data-raw/dime/cgm/101.csv") |>
  read_csv(show_col_types = FALSE,
           name_repair = to_snake_case,
           n_max = 100)

sleep_101 <- here("data-raw/dime/sleep/101.csv") |>
  read_csv(show_col_types = FALSE,
           name_repair = to_snake_case,
           n_max = 50)



View(sleep_101)
```

Functions

Adding function

```r
#' Adds two numbers
#'
#' @param num1 is a number
#' @param num2 is a number
#'
#' @returns gives the sum of the two numbers
#' @export
add_numbers <- function(num1, num2) {
  added <- num1 + num2
  return(added)
}
```

Importing functions individually(not general)

```r
#' Import function CGM
#'
#' @param file_path File path for cgm data
#'
```

```r
#' @returns Imported file as data.frame
import_cgm <- function(file_path) {
  cgm <- file_path |>
    read_csv(
      show_col_types = FALSE,
      name_repair = to_snake_case,
      n_max = 100
    )
  return(cgm)
}


#' Import function Sleep
#'
#' @param file_path File path for Sleep data
#'
#' @returns Imported file data.frame
import_sleep <- function(file_path) {
  sleep <- file_path |>
    read_csv(
      show_col_types = FALSE,
      name_repair = to_snake_case,
      n_max = 100
    )
  return(sleep)
}

here("data-raw/dime/cgm/104.csv") |>
  import_cgm()


here("data-raw/dime/sleep/101.csv") |>
  import_sleep()
```

General import function use example

```r
#import_data_snake("sleep","105",50)

#here::here("data-raw/dime/cgm/101.csv") |>
#import_dime()
```

```r
#cgm_files <- here::here("data-raw/dime/cgm/") |>
#  dir_ls(glob = "*.csv")
#sleep_files <- here::here("data-raw/dime/sleep/") |>
#  dir_ls(glob = "*.csv")

# With pipes
#cgm_data <- cgm_files |>
#  map(import_dime) |>
#  list_rbind(names_to = "file_path_id")


#sleep_data <- sleep_files |>
```

```
#   map(import_dime) |>
#   list_rbind(names_to = "file_path_id")
```

Exercise: Convert map to function and use on sleep

```
#test <- here("data-raw/dime/sleep/") |>
#import_csv()
```

Cleaning characters and dates

```
#library(stringr)
#library(lubridate)

text <- "data-raw/dime/sleep/101.csv"

str_extract(text,"\\d{3}")
str_extract(text,"[:digit:]+\\.csv$")
```

Using regex for ID extraction

```
cgm_data |>
  dplyr::mutate(
    id = stringr::str_extract(file_path_id,"[:digit:]+\\.csv$",) |>
      stringr::str_remove("\\.csv$") |>
      as.integer(),
    .before = file_path_id
  ) |>
  select(-file_path_id)
```

```
cgm_data
sleep_data
```

Working with dates extraction

```
time_extraction <- function(data){
  data_output <- data |>
    dplyr::mutate(
      date = lubridate::as_date(device_timestamp),
      hour = lubridate::hour(device_timestamp),
      .before = device_timestamp
    ) |>
    dplyr::select(-device_timestamp)

  return(data_output)
}
```

```
test2 <- here::here("data-raw/dime/cgm") |>
  import_csv() |>
  get_participant_id() |>
  time_extraction()
test2
```

```
prepare_dates <- function(data,column){
  data_output <- data |>
    dplyr::mutate(
      date = lubridate::as_date({{column}}),
      hour = lubridate::hour({{column}}),
      .before = {{column}}
    ) |>
    dplyr::select(-{{column}})

    return(data_output)
}
```

Section 10

```
summarise_column <- function(data, column, functions) {
  data |>
    dplyr::select(
      -tidyselect::contains("device_timestamp"),
      -tidyselect::contains("datetime")
    ) |>
    dplyr::group_by(dplyr::pick(-{{ column }})) |>
    dplyr::summarise(
      dplyr::across(
        {{ column }},
        functions
      ),
      .groups = "drop"
    )
}


cgm_data |>
  summarise_column(glucose,list(sum=sum,median=median,sd=sd))
```

```
# A tibble: 506 × 6
     id date        hour glucose_sum glucose_median glucose_sd
  <int> <date>     <int>       <dbl>          <dbl>      <dbl>
1   101 2021-03-18     8        16.3           5.4       0.351
2   101 2021-03-18     9        20.2           5.1       0.300
3   101 2021-03-18    10        21.2           5.35      0.392
4   101 2021-03-18    11        16.1           3.95      0.189
5   101 2021-03-18    12        16.1           4.05      0.0957
6   101 2021-03-18    13        16.4           4.05      0.141
7   101 2021-03-18    14        22.5           5.55      0.946
8   101 2021-03-18    15        29.2           7.25      0.141
```

```
 9   101 2021-03-18   16       26.2        6.5      0.7
10   101 2021-03-18   17       19.2        4.7      0.356
# i 496 more rows
```

```
sleep_data |>
  summarise_column(seconds,sum)
```

```
# A tibble: 1,258 × 5
      id date        hour sleep_type seconds
   <int> <date>     <int> <chr>        <dbl>
 1   101 2021-05-21    23 deep           390
 2   101 2021-05-21    23 light         2730
 3   101 2021-05-21    23 wake           450
 4   101 2021-05-22     0 deep          1200
 5   101 2021-05-22     0 light         1230
 6   101 2021-05-22     0 rem           1620
 7   101 2021-05-22     1 deep           420
 8   101 2021-05-22     1 light         2970
 9   101 2021-05-22     2 light         3870
10   101 2021-05-22     2 rem           1050
# i 1,248 more rows
```