process_cleaning_merging

Approach:

My analytical approach addresses the distinct characteristics of the data, which is organised into three main temporal categories: daily, hourly and minute-level datasets. I will process each of these categories independently. A critical initial step for each dataset will be to thoroughly examine its integrity and completeness to ensure data quality, before proceeding with further analysis. The same approach will be implemented for the other two data sets representing heart rate and user weight data.

Packages used:

```
library(janitor)
library(skimr)
library(tidyverse)
library(lubridate)
library(here)
library(readr)
```

Daily data sets:

Importing datasets:

```
activity <- read.csv(here("mturkfitbit_export_4.12.16-5.12.16",
                          "Fitabase Data 4.12.16-5.12.16",
                          "dailyActivity_merged.csv"))

steps <- read.csv(here("mturkfitbit_export_4.12.16-5.12.16",
                       "Fitabase Data 4.12.16-5.12.16",
                       "dailySteps_merged.csv"))

intensities <- read_csv(here("mturkfitbit_export_4.12.16-5.12.16",
                             "Fitabase Data 4.12.16-5.12.16",
                             "dailyIntensities_merged.csv"))
```

```
## Rows: 940 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityDay
## dbl (9): Id, SedentaryMinutes, LightlyActiveMinutes, FairlyActiveMinutes, Ve...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
sleep <- read_csv(here("mturkfitbit_export_4.12.16-5.12.16",
                       "Fitabase Data 4.12.16-5.12.16",
                       "sleepDay_merged.csv"))
```

```
## Rows: 413 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): SleepDay
## dbl (4): Id, TotalSleepRecords, TotalMinutesAsleep, TotalTimeInBed
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Upon initial inspection, it is evident that the 'activity' dataset already contains information on calories, intensity and steps, thus eliminating the need to load a separate dataset for these metrics. The 'sleep' dataset will be kept separate due to its distinct nature.

Data cleaning:

It's easier to me to work with snake_case so I 'clear_names()' in all datasets.

```r
daily_all_data_frames <- list(
  activity = activity,
  steps = steps,
  intensities = intensities,
  sleep = sleep
)

daily_cleaned_data_frames <- daily_all_data_frames %>%
  map(~ .x %>% janitor::clean_names())

list2env(daily_cleaned_data_frames, envir = .GlobalEnv)
```

```
## <environment: R_GlobalEnv>
```

Now let's have a qiuck overview of the data using *skim*.

```r
activity %>% skimr::skim_without_charts()
```

Table 1: Data summary

Name	Piped data
Number of rows	940
Number of columns	15
Column type frequency:	
character	1
numeric	14
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
activity_date	0	1	8	9	0	31	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
id	0	1	4.855407e+09	2.424805e+09	1503960366	2.320127e+09	4.445115e+09	6.962181e+09	8.877689e+09
total_steps	0	1	7.637910e+03	5.087150e+03	0	3.789750e+03	7.405500e+03	1.072700e+04	3.601900e+04
total_distance	0	1	5.490000e+00	3.920000e+00	0	2.620000e+00	5.240000e+00	7.710000e+00	2.803000e+01
tracker_distance	0	1	5.480000e+00	3.910000e+00	0	2.620000e+00	5.240000e+00	7.710000e+00	2.803000e+01
logged_activities_distance	0	1	1.100000e-01	6.200000e-01	0	0.000000e+00	0.000000e+00	0.000000e+00	4.940000e+00
very_active_distance	0	1	1.500000e+00	2.660000e+00	0	0.000000e+00	2.100000e-01	2.050000e+00	2.192000e+01
moderately_active_distance	0	1	5.700000e-01	8.800000e-01	0	0.000000e+00	2.400000e-01	8.000000e-01	6.480000e+00
light_active_distance	0	1	3.340000e+00	2.040000e+00	0	1.950000e+00	3.360000e+00	4.780000e+00	1.071000e+01
sedentary_active_distance	0	1	0.000000e+00	1.000000e-02	0	0.000000e+00	0.000000e+00	0.000000e+00	1.100000e-01
very_active_minutes	0	1	2.116000e+01	3.284000e+01	0	0.000000e+00	4.000000e+00	3.200000e+01	2.100000e+02
fairly_active_minutes	0	1	1.356000e+01	1.999000e+01	0	0.000000e+00	6.000000e+00	1.900000e+01	1.430000e+02
lightly_active_minutes	0	1	1.928100e+02	1.091700e+02	0	1.270000e+02	1.990000e+02	2.640000e+02	5.180000e+02
sedentary_minutes	0	1	9.912100e+02	3.012700e+02	0	7.297500e+02	1.057500e+03	1.229500e+03	1.440000e+03
calories	0	1	2.303610e+03	7.181700e+02	0	1.828500e+03	2.134000e+03	2.793250e+03	4.900000e+03

```r
sleep %>% skimr::skim_without_charts()
```

Table 4: Data summary

Name	Piped data
Number of rows	413
Number of columns	5
Column type frequency:	
character	1
numeric	4
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
sleep_day	0	1	20	21	0	31	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
id	0	1	5.000979e+09	2.060360e+09	1503960366	3977333714	4702921684	6962181067	8792009665
total_sleep_records	0	1	1.120000e+00	3.450000e-01	1	1	1	1	3
total_minutes_asleep	0	1	4.194700e+02	1.183400e+02	58	361	433	490	796
total_time_in_bed	0	1	4.586400e+02	1.271000e+02	61	403	463	526	961

There are no missing values in each data set. Let's check if the number of people represented by the 'id' column is the same in every dataset.

```r
n_distinct(activity)
```

```
## [1] 940
```

```r
n_distinct(sleep)
```

```
## [1] 410
```

```r
n_distinct(activity$id)
```

```
## [1] 33
```

```r
n_distinct(sleep$id)
```

```
## [1] 24
```

It's now evident that the sleep dataset contains a smaller number of users. This needs to be taken into consideration during the analysis.

Daily data has been gathered from 33 people; however, only 24 users are represented in the sleep data.

Fixsing the data format:

```r
activity$activity_date=as.POSIXct(activity$activity_date, format="%m/%d/%Y", tz=Sys.timezone())
```

Hourly Data Sets:

Importing datasets:

```r
hourly_calories <- read.csv(here("mturkfitbit_export_4.12.16-5.12.16",
                                 "Fitabase Data 4.12.16-5.12.16",
                                 "hourlyCalories_merged.csv"))
```

```r
hourly_intensities <- read.csv(here("mturkfitbit_export_4.12.16-5.12.16",
                                    "Fitabase Data 4.12.16-5.12.16",
                                    "hourlyIntensities_merged.csv"))

hourly_steps <- read_csv(here("mturkfitbit_export_4.12.16-5.12.16",
                              "Fitabase Data 4.12.16-5.12.16",
                              "hourlySteps_merged.csv"))
```

Upon initial review the datasets should be merge together.

Data cleaning:

```r
hourly_all_data_frames <- list(
  hourly_calories = hourly_calories,
  hourly_intensities = hourly_intensities,
  hourly_steps = hourly_steps
)

hourly_cleaned_data_frames <- hourly_all_data_frames %>%
  map(~ .x %>% janitor::clean_names())

list2env(hourly_cleaned_data_frames, envir = .GlobalEnv)
```

```
## <environment: R_GlobalEnv>
```

Let's again have a quick overview plus check the distinct 'id' value.

```r
hourly_calories %>% skimr::skim_without_charts()
```

Table 7: Data summary

Name	Piped data
Number of rows	22099
Number of columns	3
Column type frequency:	
character	1
numeric	2
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
activity_hour	0	1	19	21	0	736	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
id	0	1	4.848235e+09	2.94225e+09	1503960366	2320127002	4445114986	6962181067	8877689391
calories	0	1	9.739000e+01	6.0700e+01	42	63	83	108	948

```
hourly_intensities %>% skimr::skim_without_charts()
```

Table 10: Data summary

Name	Piped data
Number of rows	22099
Number of columns	4
Column type frequency:	
character	1
numeric	3
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
activity_hour	0	1	19	21	0	736	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
id	0	1	4.848235e+09	2.94225e+09	1503960366	2320127002	4.445115e+09	6.962181e+09	8877689391
total_intensity	0	1	1.204000e+01	2.11130e+01	0	0	3.000000e+00	1.600000e+01	180
average_intensity	0	1	2.000000e-01	3.5000e-01	0	0	5.000000e-02	2.700000e-01	3

```
hourly_steps %>% skimr::skim_without_charts()
```

Table 13: Data summary

Name	Piped data
Number of rows	22099
Number of columns	3
Column type frequency:	
character	1
numeric	2
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
activity_hour	0	1	19	21	0	736	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
id	0	1	4.848235e+09	2.94225e+09	1503960366	2320127002	4445114986	6962181067	8877689391
step_total	0	1	3.201700e+02	6.29038e+02	0	0	40	357	10554

```r
n_distinct(hourly_calories)
```

```
## [1] 22099
```

```r
n_distinct(hourly_intensities)
```

```
## [1] 22099
```

```r
n_distinct(hourly_steps)
```

```
## [1] 22099
```

```r
n_distinct(hourly_calories$id)
```

```
## [1] 33
```

```r
n_distinct(hourly_intensities$id)
```

```
## [1] 33
```

```r
n_distinct(hourly_steps$id)
```

```
## [1] 33
```

Datasets have the same length, let's merge it.

Datasets merging:

```r
hourly_cis_temp <- merge(hourly_calories, hourly_intensities, by = c('id', 'activity_hour'))
hourly_cis <- merge(hourly_cis_temp, hourly_steps, by = c('id', 'activity_hour'))
head(hourly_cis)
```

```
##           id         activity_hour calories total_intensity average_intensity
## 1 1503960366  4/12/2016 1:00:00 AM       61               8          0.133333
## 2 1503960366  4/12/2016 1:00:00 PM       66               6          0.100000
## 3 1503960366 4/12/2016 10:00:00 AM       99              29          0.483333
## 4 1503960366 4/12/2016 10:00:00 PM       65               9          0.150000
## 5 1503960366 4/12/2016 11:00:00 AM       76              12          0.200000
## 6 1503960366 4/12/2016 11:00:00 PM       81              21          0.350000
##   step_total
## 1        160
## 2        221
## 3        676
## 4         89
## 5        360
## 6        338
```

I ended up with one dataset reflecting the hourly data, called 'hourly_cis'.

Fixsing the timestemp.

I would like to separate day and time from 'activity_hour'.

```
hourly_cis$activity_hour=as.POSIXct(hourly_cis$activity_hour, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.tim
hourly_cis$time <- format(hourly_cis$activity_hour, format = "%H:%M:%S")
hourly_cis$date <- format(hourly_cis$activity_hour, format = "%m/%d/%y")
```

Minute Data Sets:

Importing Data:

```
minute_calories <- read.csv(here("mturkfitbit_export_4.12.16-5.12.16",
                                  "Fitabase Data 4.12.16-5.12.16",
                                  "minuteCaloriesNarrow_merged.csv"))

minute_intensities <- read.csv(here("mturkfitbit_export_4.12.16-5.12.16",
                                     "Fitabase Data 4.12.16-5.12.16",
                                     "minuteIntensitiesNarrow_merged.csv"))

minute_METs <- read.csv(here("mturkfitbit_export_4.12.16-5.12.16",
                              "Fitabase Data 4.12.16-5.12.16",
                              "minuteMETsNarrow_merged.csv"))

minute_sleep <- read.csv(here("mturkfitbit_export_4.12.16-5.12.16",
                               "Fitabase Data 4.12.16-5.12.16",
                               "minuteSleep_merged.csv"))

minute_steps <- read.csv(here("mturkfitbit_export_4.12.16-5.12.16",
                               "Fitabase Data 4.12.16-5.12.16",
                               "minuteStepsNarrow_merged.csv"))
```

Data cleaning:

```r
minute_all_data_frames <- list(
  minute_calories = minute_calories,
  minute_intensities = minute_intensities,
  minute_METs = minute_METs,
  minute_sleep = minute_sleep,
  minute_steps = minute_steps
)

minute_cleaned_data_frames <- minute_all_data_frames %>%
  map(~ .x %>% janitor::clean_names())

list2env(minute_cleaned_data_frames, envir = .GlobalEnv)
```

```
## <environment: R_GlobalEnv>
```

Overview of data.

```r
minute_calories %>% skimr::skim_without_charts()
```

Table 16: Data summary

Name	Piped data
Number of rows	1325580
Number of columns	3
Column type frequency:	
character	1
numeric	2
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
activity_minute	0	1	19	21	0	44160	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
id	0	1	4.847898e+09	2.422313e+09	1503960366	2.320127e+09	4.445115e+09	6.962181e+09	8.877689e+09
calories	0	1	1.620000e+00	1.410000e+00	0	9.400000e-01	1.220000e+00	1.430000e+00	1.975000e+01

```r
minute_intensities %>% skimr::skim_without_charts()
```

Table 19: Data summary

Name	Piped data
Number of rows	1325580
Number of columns	3
Column type frequency:	
character	1
numeric	2
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
activity_minute	0	1	19	21	0	44160	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
id	0	1	4847897691	2.422313e+09	1503960366	2320127002	4445114986	6962181067	8877689391
intensity	0	1	0.2	5.200000e-01	0	0	0	0	3

```
minute_METs %>% skimr::skim_without_charts()
```

Table 22: Data summary

Name	Piped data
Number of rows	1325580
Number of columns	3
Column type frequency:	
character	1
numeric	2
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
activity_minute	0	1	19	21	0	44160	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
id	0	1	4.847898e+09	2.422313e+09	1503960366	2320127002	4445114986	6962181067	8877689391
me_ts	0	1	1.469000e+01	1.206000e+01	0	10	10	11	157

```r
minute_sleep %>% skimr::skim_without_charts()
```

Table 25: Data summary

Name	Piped data
Number of rows	188521
Number of columns	4
Column type frequency:	
character	1
numeric	3
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
date	0	1	19	21	0	49773	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
id	0	1	4.996595e+09	2.066950e+09	1503960366	3977333714	4702921684	6962181067	8792009665
value	0	1	1.100000e+00	3.300000e-01	1	1	1	1	3
log_id	0	1	1.149611e+10	6.822863e+07	11372227280	11439308639	11501142214	11552534115	11616251768

```r
minute_steps %>% skimr::skim_without_charts()
```

Table 28: Data summary

Name	Piped data
Number of rows	1325580
Number of columns	3
Column type frequency:	
character	1
numeric	2
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
activity_minute	0	1	19	21	0	44160	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
id	0	1	4.847898e+09	2.422313e+09	1503960366	2320127002	4445114986	6962181067	8877689391
steps	0	1	5.340000e+00	1.813000e+01	0	0	0	0	220

Check unique 'id'.

```
n_distinct(minute_calories)
```

```
## [1] 1325580
```

```
n_distinct(minute_intensities)
```

```
## [1] 1325580
```

```
n_distinct(minute_METs)
```

```
## [1] 1325580
```

```
n_distinct(minute_sleep)
```

```
## [1] 187978
```

```
n_distinct(minute_steps)
```

```
## [1] 1325580
```

```
n_distinct(minute_calories$id)
```

```
## [1] 33
```

```
n_distinct(minute_intensities$id)
```

```
## [1] 33
```

```
n_distinct(minute_METs$id)
```

```
## [1] 33
```

```r
n_distinct(minute_sleep$id)
```

```
## [1] 24
```

```r
n_distinct(minute_steps$id)
```

```
## [1] 33
```

As expected, the 'minute_sleep' data frame contains data from a smaller number of distinct people.

Datasets merging:

```r
minute_ci_temp <- merge(minute_calories, minute_intensities,
                        by = c("id", "activity_minute"))


minute_cim_temp <- merge(minute_ci_temp, minute_METs,
                         by = c("id", "activity_minute"))


minute_ciMs <- merge(minute_cim_temp, minute_steps,
                            by = c("id", "activity_minute"))

head(minute_ciMs)
```

```
##           id      activity_minute calories intensity me_ts steps
## 1 1503960366 4/12/2016 1:00:00 AM   0.9438         0    12     0
## 2 1503960366 4/12/2016 1:00:00 PM   0.9438         0    12     0
## 3 1503960366 4/12/2016 1:01:00 AM   2.6741         1    34    36
## 4 1503960366 4/12/2016 1:01:00 PM   0.9438         0    12     0
## 5 1503960366 4/12/2016 1:02:00 AM   2.0449         1    26     9
## 6 1503960366 4/12/2016 1:02:00 PM   0.9438         0    12     0
```

I ended up with one dataset reflecting the minutes data, called 'minute_ciMs'.

Fixsing the timestemp.

```r
minute_ciMs$activity_minute=as.POSIXct(minute_ciMs$activity_minute, format="%m/%d/%Y %I:%M:%S %p", tz=Sy
minute_ciMs$time <- format(minute_ciMs$activity_minute, format = "%H:%M:%S")
minute_ciMs$date <- format(minute_ciMs$activity_minute, format = "%m/%d/%y")
```

```r
minute_sleep$date=as.POSIXct(minute_sleep$date, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())
minute_sleep$time <- format(minute_sleep$date, format = "%H:%M:%S")
minute_sleep$date <- format(minute_sleep$date, format = "%m/%d/%y")
```

Weight and Heartrate Data Sets:

Importing Data:

```
weight <- read_csv(here("mturkfitbit_export_4.12.16-5.12.16",
                        "Fitabase Data 4.12.16-5.12.16",
                        "weightLogInfo_merged.csv"))

heartrate <- read_csv(here("mturkfitbit_export_4.12.16-5.12.16",
                           "Fitabase Data 4.12.16-5.12.16",
                           "heartrate_seconds_merged.csv"))
```

Data cleaning:

```
weight <- clean_names(weight)
heartrate <- clean_names(heartrate)
```

```
weight %>% skimr::skim_without_charts()
```

Table 31: Data summary

Name	Piped data
Number of rows	67
Number of columns	8
Column type frequency:	
character	1
logical	1
numeric	6
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
date	0	1	19	21	0	56	0

Variable type: logical

skim_variable	n_missing	complete_rate	mean	count
is_manual_report	0	1	0.61	TRU: 41, FAL: 26

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
id	0	1.00	7.009282e+09	1.950322e+09	1.503960e+09	6.962181e+09	6.962181e+09	8.877689e+09	8.877689e+09
weight_kg	0	1.00	7.204000e+01	1.392000e+01	5.260000e+01	6.140000e+01	6.250000e+01	8.505000e+01	1.335000e+02
weight_pounds	0	1.00	1.588100e+02	3.070000e+01	1.159600e+02	1.353600e+02	1.377900e+02	1.875000e+02	2.943200e+02
fat	65	0.03	2.350000e+01	2.120000e+00	2.200000e+01	2.275000e+01	2.350000e+01	2.425000e+01	2.500000e+01
bmi	0	1.00	2.519000e+01	3.070000e+00	2.145000e+01	2.396000e+01	2.439000e+01	2.556000e+01	4.754000e+01
log_id	0	1.00	1.461772e+12	7.829948e+08	1.460444e+12	1.461079e+12	1.461802e+12	1.462375e+12	1.463098e+12

We can see that for 'fat' column most values are not complete (97.015%).

```r
heartrate %>% skimr::skim_without_charts()
```

Table 35: Data summary

Name	Piped data
Number of rows	2483658
Number of columns	3
Column type frequency:	
character	1
numeric	2
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
time	0	1	19	21	0	961274	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
id	0	1	5.513765e+09	1950223761.2	2022484408	4388161847	5553957443	6962181067	8877689391
value	0	1	7.733000e+01	19.4	36	63	73	88	203

```r
n_distinct(weight$id)
```

```
## [1] 8
```

```r
n_distinct(heartrate$id)
```

```
## [1] 14
```

Unfortunately, the group of users from which the 'weight' data is taken is too small to draw any conclusions. Even the 'heart rate' data is based on a small sample.

Fixsing taamesteps:

```
weight$date=as.POSIXct(weight$date, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())
weight$time <- format(weight$date, format = "%H:%M:%S")
weight$date <- format(weight$date, format = "%m/%d/%y")
```

```
heartrate$time=as.POSIXct(heartrate$time, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())
heartrate$date <- format(heartrate$time, format = "%m/%d/%y")
heartrate$time_sec <- format(heartrate$time, format = "%H:%M:%S")
```

Summary:

Following data cleaning and merging operations, I've consolidated user activity information into three primary datasets, encompassing 33 unique users: 'daily_ais', 'hourly_cis' and 'minute_ciMs'. Sleep data requires independent analysis due to its user base, comprising 24 users with both daily and minute-level granularity. Heart rate data shows promise for limited integration into the overall analysis, despite its restricted user base. Contrarily, the weight dataset is currently deemed to have insufficient user representation for meaningful inclusion.

List of dataframs that will be used for futher analysis:

- daily_ais
- hourly_cis
- minute_ciMs
- sleep
- minute_sleep
- heartrate