Final Project - Fitness Tracker Data

Daniel Terra Gomes

February 2, 2023

{r} rmarkdown::pandoc_available() library(knitr) pandoc("projectFinal/finalProject.Rmd", format = "pdf") {r} install.packages("tinytex") tinytex::install_tinytex() # Library may be used {r} # https://bookdown.org/yihui/rmarkdown/pdf-document.html # Press F1 or Ctrl+Shift+P # Type export and select below install.packages("PACKAGES-NAME") {r} library(janitor) # janitor has simple little tools for examining and cleaning dirty data. library(arrow) library(tidyverse) library(naniar) library(ggsci) library(skimr) # Skim a data frame, getting useful summary statistics library(lubridate) # Lubridate provides tools that make it easier to parse and manipulate dates. # reading files csv {r} # sleep_day_file <- read_csv("projectFinal/fitDatabase/sleepDay_merged.csv") # daily_activity_file <- read_csv("projectFinal/fitDatabase/dailyActivity_merged.csv") # daily_intensities_file <- read_csv("projectFinal/fitDatabase/dailyIntensities_merged.csv") # Converting csv files to parquet to gain performance {r} # write_parquet(sleep_day_file, "projectFinal/projectDataParquet/sleepDay.parquet") # write_parquet(daily_activity_file, "projectFinal/projectDataParquet/dailyActivity.parquet") # write_parquet(daily_intensities_f "projectFinal/projectDataParquet/dailyIntensities.parquet") # Reading Parquet files {r} sleep_day_file <- read_parquet("projectFinal/projectDataParquet/sleepD daily_activity_file <- read_parquet("projectFinal/projectDataParquet/dailyActivity.parquet") daily_intensities_file <- read_parquet("projectFinal/projectDataParquet/dailyIntensities.par # Undestaining data {r} skim_without_charts(sleep_day_file) skim_without_charts(daily_activity_file) skim_without_charts(daily_intensities_file) {r} sleep_day_file daily_activity_file daily_intensities_file {r} glimpse(sleep_day_file) glimpse(daily_activity_file) glimpse(daily_intensities_file) ## Number of unique users {r} count(distinct(sleep_day_file, Id)) count(distinct(daily_activity_file, Id)) count(distinct(daily_intensities_file, Id)) # Data Cleaning {r} anyDuplicated(sleep_day_file) anyDuplicated(daily_activity_file) anyDuplicated(daily_intensities_file) ## Dropping NA and duplicates

“‘{r} sleep_day_file <- sleep_day_file %>% distinct() %>% drop_na()

anyDuplicated(sleep_day_file)

```
## Cleaning names to the format used in the classes
```r
sleep_day_file <- clean_names(sleep_day_file)
daily_activity_file <- clean_names(daily_activity_file)
```
```

```
daily_intensities_file <- clean_names(daily_intensities_file)
```

Tranformating the Dates

{r} glimpse(sleep_day_file) glimpse(daily_activity_file) glimpse(daily_intensities_file) ### {r} sleep_day_file$sleep_day <- mdy_hms(sleep_day_file$sleep_day) daily_activity_file$activity_date <- mdy(daily_activity_file$activity_date) daily_intensities_file$activity_day <- mdy(daily_intensities_file$activity_day) {r} sleep_day_file daily_activity_file daily_intensities_file # Analizing {r} glimpse(sleep_day_file) glimpse(daily_activity_file) glimpse(daily_intensities_file) Devide use throughout the day "'{r} daily_activity_file$total_time = rowSums(daily_activity_file[c("very_active_minutes", "fairly_active_minutes", "lightly_active_minutes", "sedentary_minutes")])

daily_activity_file |> group_by(id) |> summarise(daily_usage_hour = mean(total_time / 60)) |> ggplot(aes(x = daily_usage_hour)) + geom_histogram(# mapping = aes(x = daily_usage_hour), color = "black", fill = "#008b3a") + scale_color_igv() + scale_fill_igv() + theme_grey() + scale_x_continuous(breaks = c(1:24)) + labs(title = "Average Daily Device Usage Time", subtitle = "", x ="Daily Use Time(hours)", y ="Count") "'