Cumulative Ride Miles Plots

DH

2025-07-23

#{r setup, include=FALSE} #knitr::opts_chunk$set(echo = TRUE) #

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
# Set working directory to "C:/Data/Projects"
setwd("C:/Users/hanse_1gnov55/OneDrive/Documents/R/Strava")  #on Surface 3 Laptop
getwd()
```

```
## [1] "C:/Users/hanse_1gnov55/OneDrive/Documents/R/Strava"
```

```
# Package Installation

#install.packages("readxl","dplyr", "lubridate", "ggplot2", "readr" , "janitor")
```

```
# START HERE AS OF JULY 21
# Master code block.  By hand!

###### Cumulative Miles, Duration and Calories by Normalized Date over three years

#1. load packages

library(readxl)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
library(ggplot2)
library (readr)
library (janitor)
```

```
##
## Attaching package: 'janitor'

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

```r
#2. read and prepare the data

Rides <- read_xlsx ("Rides Miles Comp.xlsx")
```

```
## Warning: Expecting logical in L1213 / R1213C12: got 'Diverge Comp Gray with
## Yellow'

## Warning: Expecting logical in L1215 / R1215C12: got 'Diverge Comp Gray with
## Yellow'

## Warning: Expecting logical in L1216 / R1216C12: got 'Diverge Comp Gray with
## Yellow'

## Warning: Expecting logical in L1217 / R1217C12: got 'Diverge Comp Gray with
## Yellow'

## Warning: Expecting logical in L1218 / R1218C12: got 'Diverge Comp Gray with
## Yellow'

## Warning: Expecting logical in L1219 / R1219C12: got 'Diverge Comp Gray with
## Yellow'

## Warning: Expecting logical in L1220 / R1220C12: got 'Diverge Comp Gray with
## Yellow'

## Warning: Expecting logical in L1221 / R1221C12: got 'Diverge Comp Gray with
## Yellow'

## Warning: Expecting logical in L1222 / R1222C12: got 'Diverge Comp Gray with
## Yellow'
```

```
## Warning: Expecting logical in L1929 / R1929C12: got 'Diverge 2024'

## Warning: Expecting logical in L1930 / R1930C12: got 'Diverge 2024'

## Warning: Expecting logical in L1932 / R1932C12: got 'Diverge 2024'

## Warning: Expecting logical in L1933 / R1933C12: got 'Diverge 2024'

## Warning: Expecting logical in L1936 / R1936C12: got 'Diverge 2024'

## Warning: Expecting logical in L1937 / R1937C12: got 'Diverge 2024'

## Warning: Expecting logical in L1938 / R1938C12: got 'Diverge 2024'

## Warning: Expecting logical in L1939 / R1939C12: got 'Diverge 2024'

## Warning: Expecting logical in L1942 / R1942C12: got 'Diverge 2024'

## Warning: Expecting logical in L1944 / R1944C12: got 'Diverge 2024'

## Warning: Expecting logical in L1945 / R1945C12: got 'Diverge 2024'

## Warning: Expecting logical in L1946 / R1946C12: got 'Diverge 2024'

## New names:
## * `Elapsed Time` -> `Elapsed Time...6`
## * `Distance` -> `Distance...7`
## * `Max Heart Rate` -> `Max Heart Rate...8`
## * `Relative Effort` -> `Relative Effort...9`
## * `Commute` -> `Commute...10`
## * `Elapsed Time` -> `Elapsed Time...16`
## * `Distance` -> `Distance...18`
## * `Max Heart Rate` -> `Max Heart Rate...31`
## * `Relative Effort` -> `Relative Effort...38`
## * `Commute` -> `Commute...51`
```

```
#glimpse (Rides)

StravaRides <- Rides %>%
  select(`Activity Date`, `Activity Type`, Distance...7, `Moving Time` , Calories) |>
    rename (
    datetime = 'Activity Date',
    activity = `Activity Type`,
    distance = Distance...7 ,
    calories = `Calories`,
    duration = `Moving Time`
  )

 #glimpse (StravaRides)
```

```r
#PARSING EXCEL TIMESTAMP.

StravaRides_clean <- StravaRides %>%
  mutate(
    # Parse the existing character column `datetime`
    datetime_parsed = mdy_hms(datetime, tz = "UTC"),

    # Now derive date/time pieces off the parsed datetime
    date          = as_date(datetime_parsed),
    short_date    = format(date, "%m/%d/%Y"),
    year          = year(date),
    day_of_year = yday(date))
#glimpse (StravaRides_clean)
```

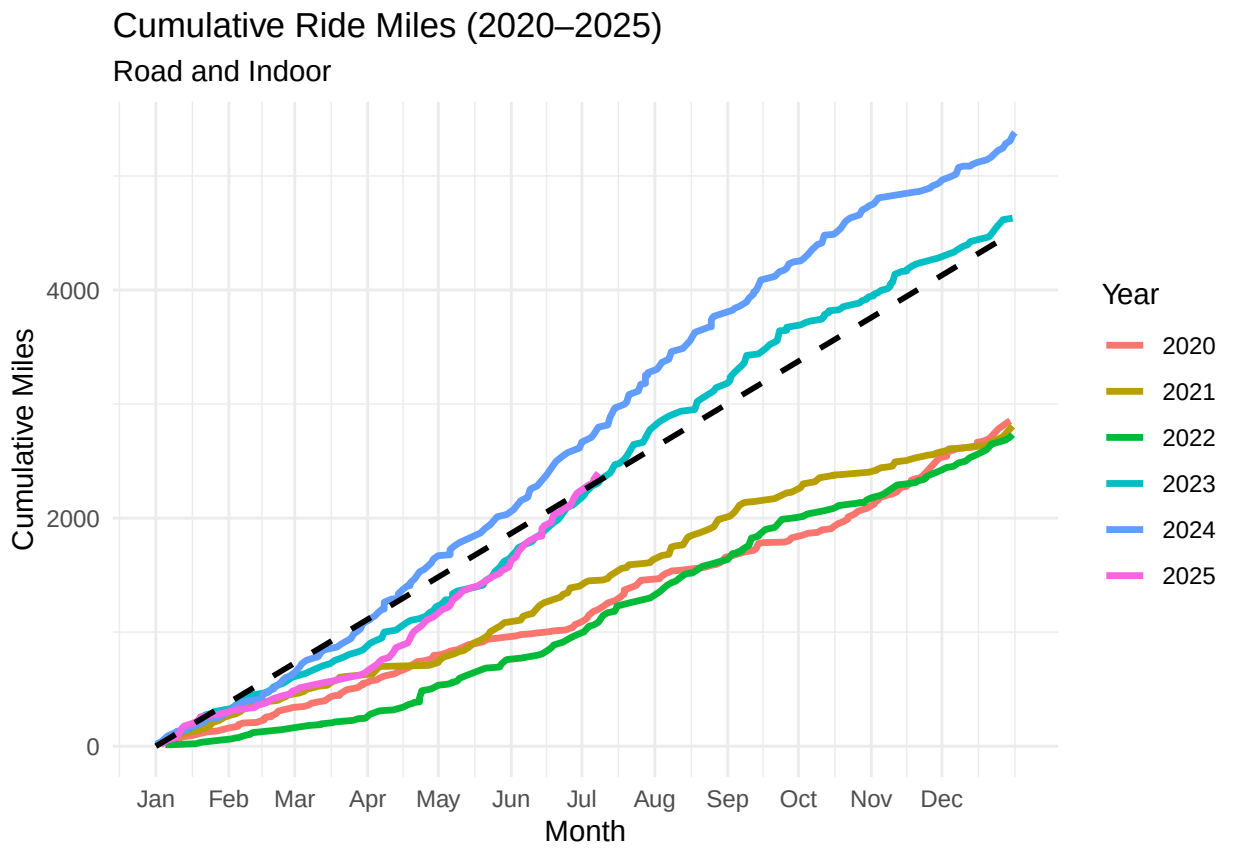
```r
#creating the annual cumulative values

StravaRides_clean <- StravaRides_clean |>
  arrange(year, day_of_year) |>
  mutate(
    # Miles conversions
    distance_miles = distance * 0.621371,

    # Duration conversions
    duration_min = duration / 60,

    # Fill calories if NA
    calories = coalesce(calories, 300 * (duration_min / 60))
  ) |>
  group_by(year) |>
  mutate(
    # Running totals reset each year
    cumulative_miles    = cumsum(distance_miles),
    cumulative_duration = cumsum(duration_min),
    cumulative_calories = cumsum(calories)
  ) |>
  ungroup()

#glimpse (StravaRides_clean)
```

```r
##### GRAPH ONE

# Filtering and mutation

StravaGraph1 <- StravaRides_clean |>
  filter(
    activity %in% c("Ride", "Virtual Ride"),
    year     %in% c(2020:2025))

#Special reference variable

target_line <- data.frame(
  day_of_year   = 1:365,
  target_miles  = seq(0, 4500, length.out = 365)
```

```
)

# A plot of overlaying years of cumulative miles

my_plotMiles <- ggplot(StravaGraph1,
  aes(x = day_of_year, y = cumulative_miles,
      color = as.factor(year))) +
  geom_line(size = 1.2) +
  geom_line(data = target_line,
            aes (x=day_of_year, y= target_miles),
            color = "black",
            linetype = "dashed",
            size = 1,
            inherit.aes = FALSE)+
  scale_x_continuous(
  breaks = c(1, 32, 60, 91, 121, 152, 182, 213, 244, 274, 305, 335),
  labels = month.abb
  ) +
  labs(
    title = "Cumulative Ride Miles (2020-2025)",
    subtitle = "Road and Indoor",
    x = "Month",
    y = "Cumulative Miles",
    color = "Year"
  ) +
  theme_minimal()
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

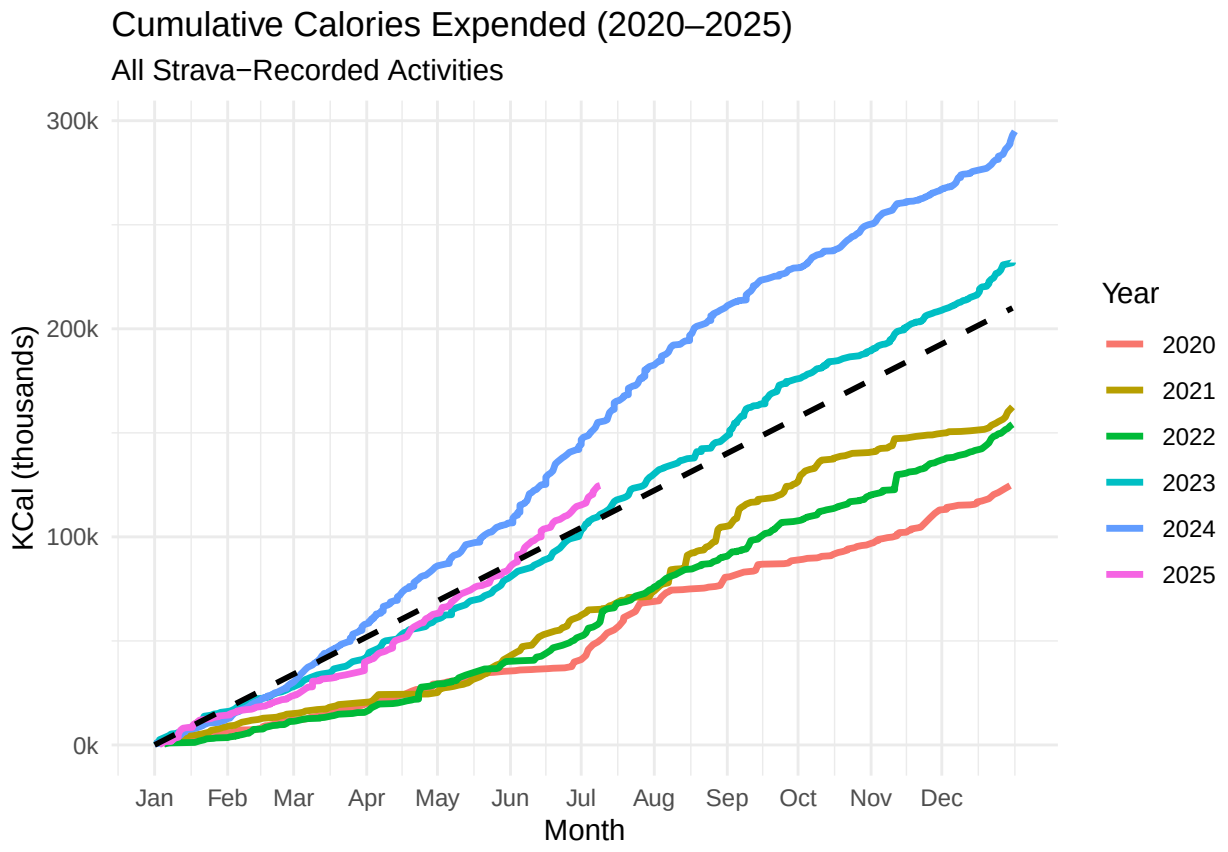
```
my_plotMiles
```




```
ggsave("cumulative_miles.jpg", plot = my_plotMiles, width = 10, height = 6, dpi = 300)
```

```
##### GRAPH TWO

# Filtering and mutation

StravaGraph1 <- StravaRides_clean |>
  filter (year     %in% c(2020:2025))

#Special reference variable
ktarget_line <- data.frame(
 day_of_year    = 1:365,
 ktarget_miles  = seq(0, 210, length.out = 365))

# A plot of overlaying years of cumulative calories

my_plotMiles <- ggplot(StravaGraph1,
  aes(x = day_of_year, y = (cumulative_calories/1000) ,
      color = as.factor(year))) +
  geom_line(size = 1.2) +
  geom_line(data = ktarget_line,
              aes (x=day_of_year, y= ktarget_miles),
              color = "black",
              linetype = "dashed",
              size = 1,
              inherit.aes = FALSE)+
```

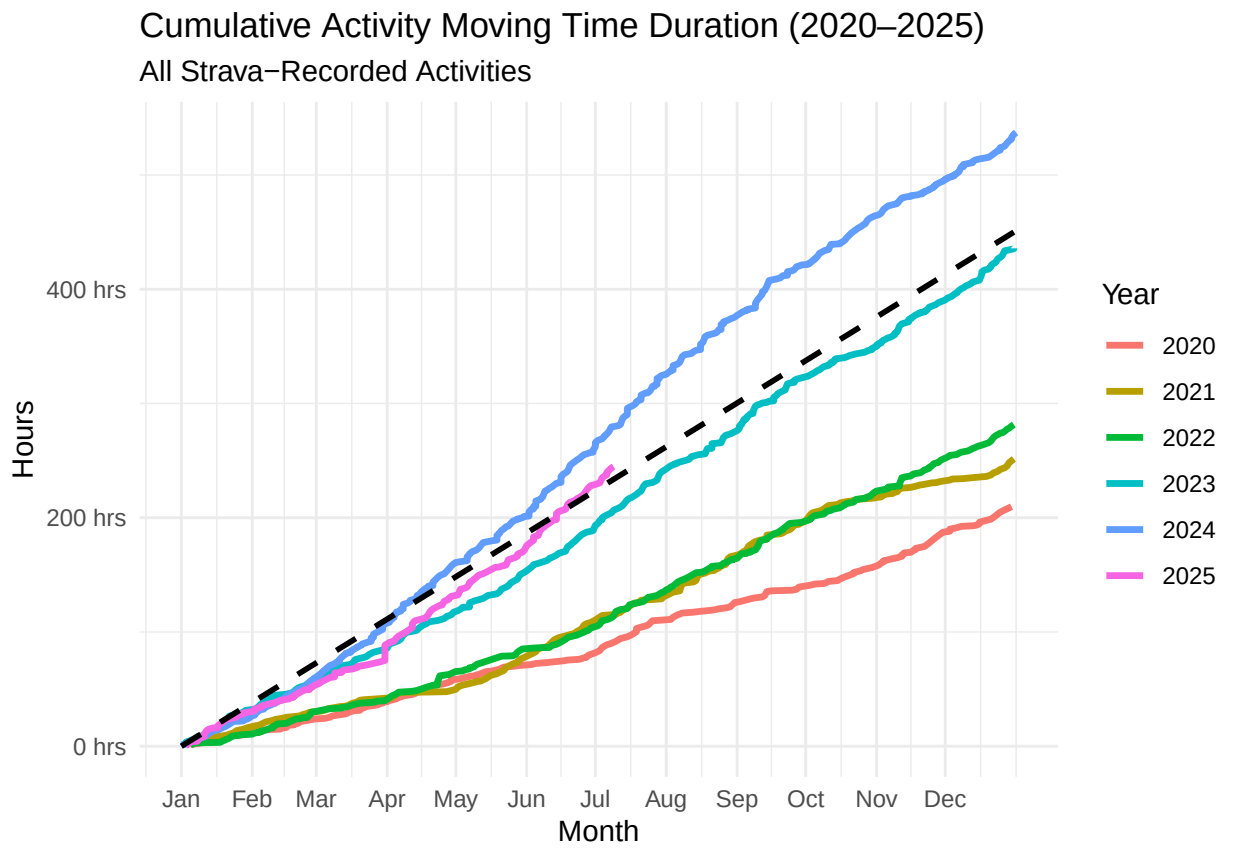
```
scale_x_continuous(
  breaks = c(1, 32, 60, 91, 121, 152, 182, 213, 244, 274, 305, 335),
  labels = month.abb
  ) +
scale_y_continuous(labels=scales::label_number(suffix = "k")) +
  labs(
    title = "Cumulative Calories Expended (2020-2025)",
    subtitle = "All Strava-Recorded Activities",
    x = "Month",
    y = "KCal (thousands)",
    color = "Year"
  ) +
  theme_minimal()
my_plotMiles
```




```
ggsave("cumulative_KCals.jpg", plot = my_plotMiles, width = 10, height = 6, dpi = 300)
```

```
##### GRAPH THREE

# Filtering and mutation

StravaGraph1 <- StravaRides_clean |>
  filter (year     %in% c(2020:2025))

#Special reference variable
```

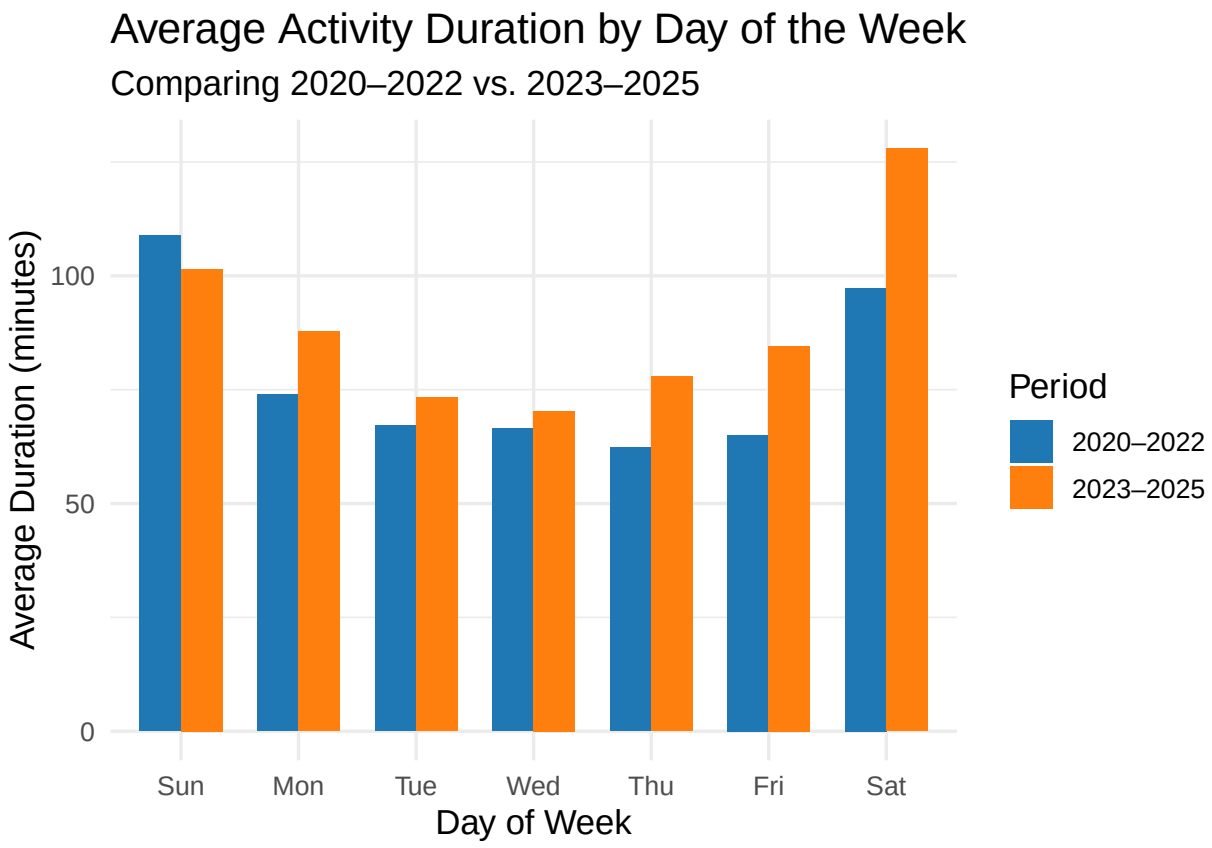
```r
ktarget_line <- data.frame(
 day_of_year   = 1:365,
 ktarget_duration  = seq(0, 450, length.out = 365))

# A plot of overlaying years of cumulative calories

my_plotMiles <- ggplot(StravaGraph1,
  aes(x = day_of_year, y = (cumulative_duration/60) ,
      color = as.factor(year))) +
  geom_line(size = 1.2) +
  geom_line(data = ktarget_line,
              aes (x=day_of_year, y= ktarget_duration),
              color = "black",
              linetype = "dashed",
              size = 1,
              inherit.aes = FALSE)+
scale_x_continuous(
  breaks = c(1, 32, 60, 91, 121, 152, 182, 213, 244, 274, 305, 335),
  labels = month.abb
  ) +
scale_y_continuous(labels=scales::label_number(suffix = " hrs")) +
  labs(
    title = "Cumulative Activity Moving Time Duration (2020-2025)",
    subtitle = "All Strava-Recorded Activities",
    x = "Month",
    y = "Hours",
    color = "Year"
  ) +
  theme_minimal()
my_plotMiles
```




```
ggsave("cumulative_duration.jpg", plot = my_plotMiles, width = 10, height = 6, dpi = 300)
```

```
##### GRAPH FOUR PAIRED BARS

# Prepare weekday comparison data
weekday_summary <- StravaRides_clean |>
  filter(year %in% 2020:2025) |>
  mutate(
    weekday = wday(date, label = TRUE, abbr = TRUE),
    period  = case_when(
      year %in% 2020:2022 ~ "2020-2022",
      year %in% 2023:2025 ~ "2023-2025"
    )
  ) |>
  group_by(period, weekday) |>
  summarise(
    avg_duration = mean(duration_min, na.rm = TRUE),
    .groups = "drop"
  )

# Create paired bar plot
my_plotWeekdays <- ggplot(weekday_summary, aes(x = weekday, y = avg_duration, fill = period)) +
  geom_bar(stat = "identity", position = "dodge", width = 0.7) +
  labs(
    title = "Average Activity Duration by Day of the Week",
    subtitle = "Comparing 2020-2022 vs. 2023-2025",
```

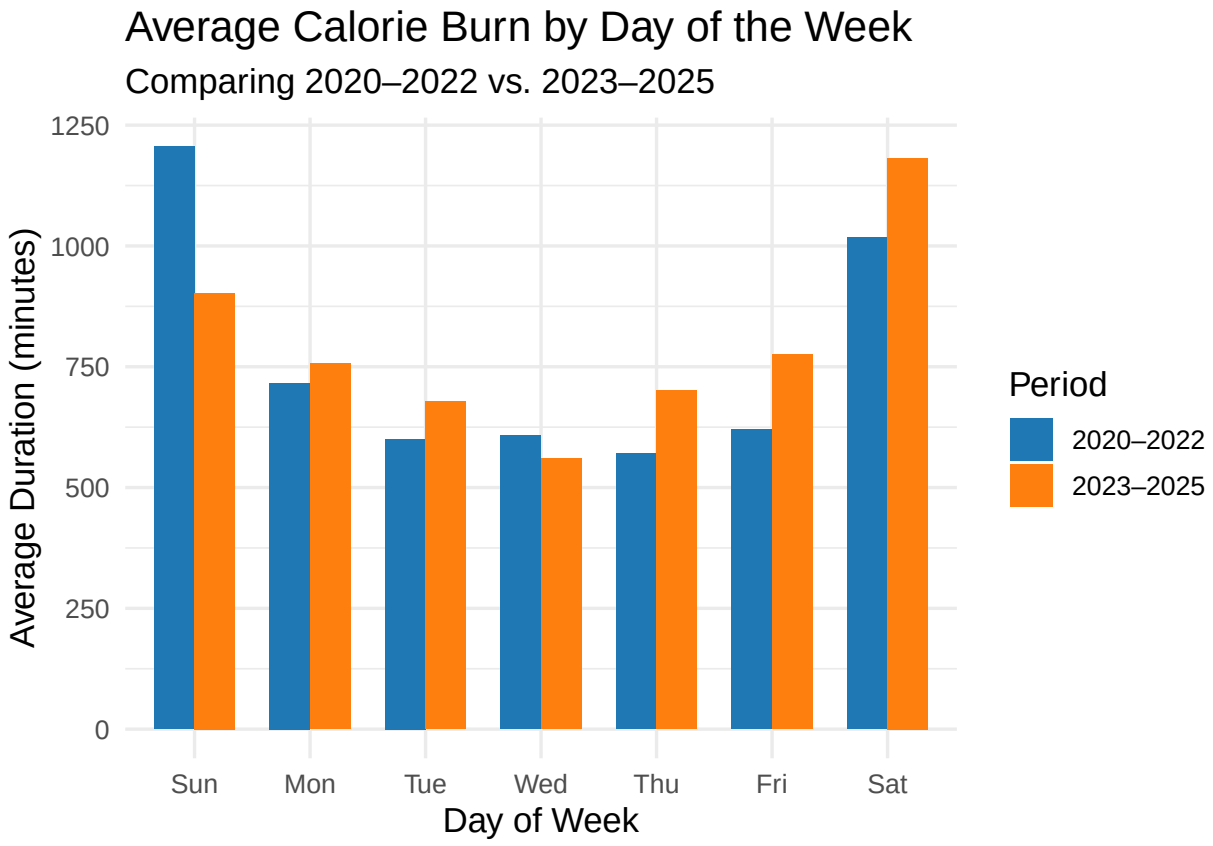
```
    x = "Day of Week",
    y = "Average Duration (minutes)",
    fill = "Period"
  ) +
  scale_fill_manual(values = c("2020-2022" = "#1f77b4", "2023-2025" = "#ff7f0e")) +
  theme_minimal(base_size = 13)

my_plotWeekdays
```




```
ggsave("avg_duration_by_weekday.jpg", plot = my_plotWeekdays, width = 9, height = 6, dpi = 300)
```

```
##### GRAPH FIVE PAIRED BARS

# Prepare weekday comparison data
weekday_summary <- StravaRides_clean |>
  filter(year %in% 2020:2025) |>
  mutate(
    weekday = wday(date, label = TRUE, abbr = TRUE),
    period  = case_when(
      year %in% 2020:2022 ~ "2020-2022",
      year %in% 2023:2025 ~ "2023-2025"
    )
  ) |>
  group_by(period, weekday) |>
  summarise(
```

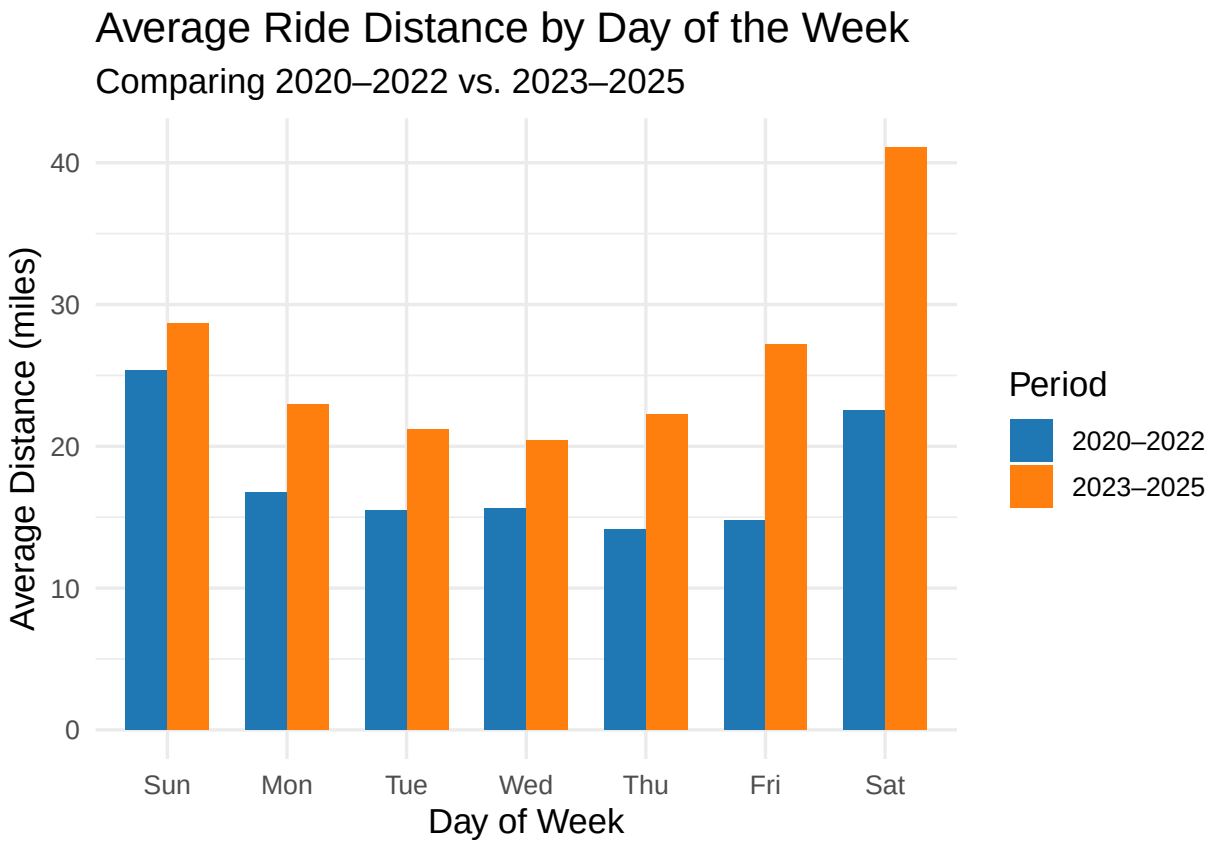
```
    avg_calories = mean(calories, na.rm = TRUE),
    .groups = "drop"
  )

# Create paired bar plot
my_plotWeekdays <- ggplot(weekday_summary, aes(x = weekday, y = avg_calories, fill = period)) +
  geom_bar(stat = "identity", position = "dodge", width = 0.7) +
  labs(
    title = "Average Calorie Burn by Day of the Week",
    subtitle = "Comparing 2020-2022 vs. 2023-2025",
    x = "Day of Week",
    y = "Average Duration (minutes)",
    fill = "Period"
  ) +
  scale_fill_manual(values = c("2020-2022" = "#1f77b4", "2023-2025" = "#ff7f0e")) +
  theme_minimal(base_size = 13)

my_plotWeekdays
```




```
ggsave("avg_calories_by_weekday.jpg", plot = my_plotWeekdays, width = 9, height = 6, dpi = 300)
```

```
##### GRAPH SIX PAIRED BARS

# Prepare weekday comparison data
weekday_summary <- StravaRides_clean |>
```

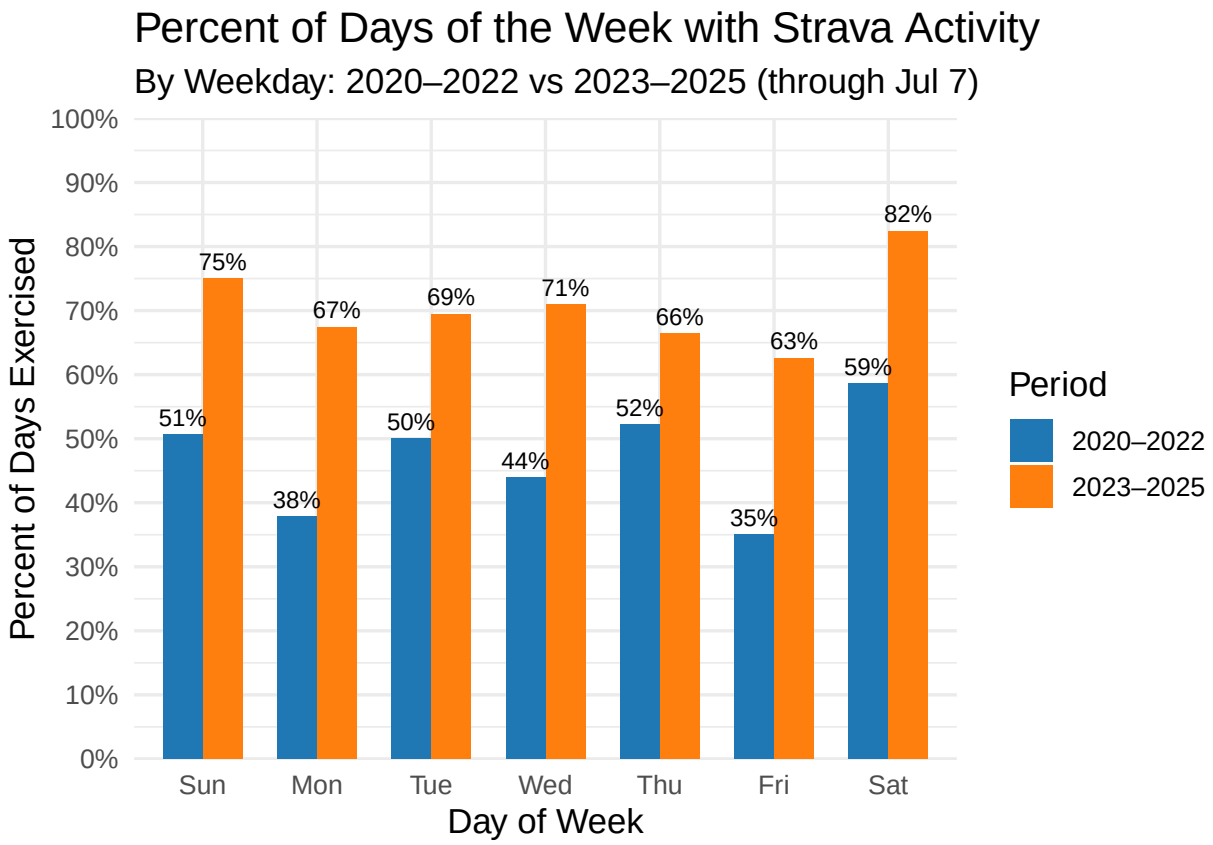
```r
  filter(activity %in% c("Ride", "Virtual Ride"), year %in% 2020:2025)|>
  mutate(
    weekday = wday(date, label = TRUE, abbr = TRUE),
    period  = case_when(
      year %in% 2020:2022 ~ "2020-2022",
      year %in% 2023:2025 ~ "2023-2025"
    )
  ) |>
  group_by(period, weekday) |>
  summarise(
    avg_distance = mean(distance_miles, na.rm = TRUE),
    .groups = "drop"
  )

# Create paired bar plot
my_plotWeekdays <- ggplot(weekday_summary, aes(x = weekday, y = avg_distance, fill = period)) +
  geom_bar(stat = "identity", position = "dodge", width = 0.7) +
  labs(
    title = "Average Ride Distance by Day of the Week",
    subtitle = "Comparing 2020-2022 vs. 2023-2025",
    x = "Day of Week",
    y = "Average Distance (miles)",
    fill = "Period"
  ) +
  scale_fill_manual(values = c("2020-2022" = "#1f77b4", "2023-2025" = "#ff7f0e")) +
  theme_minimal(base_size = 13)

my_plotWeekdays
```




```
ggsave("avg_distance_by_weekday.jpg", plot = my_plotWeekdays, width = 9, height = 6, dpi = 300)
```

```
##### GRAPH SEVEN UPDATED: PERCENT WEEKDAY FREQUENCY OF >20 MIN RIDE

# 1. Denominator: total calendar days by period & weekday
calendar_days <- tibble::tibble(
  date = c(
    seq(as.Date("2020-01-01"), as.Date("2022-12-31"), by = "day"),
    seq(as.Date("2023-01-01"), as.Date("2025-07-07"), by = "day")
  )
) %>%
  mutate(
    period = case_when(
      date <= as.Date("2022-12-31")                  ~ "2020-2022",
      date >= as.Date("2023-01-01")                  ~ "2023-2025"
    ),
    weekday = wday(date, label = TRUE, abbr = TRUE)
  ) %>%
  group_by(period, weekday) %>%
  summarise(total_days = n(), .groups = "drop")

# 2. Numerator: days with any ride >20 min
long_ride_days <- StravaRides_clean %>%
  mutate(
    date    = as_date(datetime_parsed),
    period  = case_when(
```

```
      year %in% 2020:2022 ~ "2020-2022",
      year %in% 2023:2025 ~ "2023-2025"
    ),
    weekday = wday(date, label = TRUE, abbr = TRUE),
    over20  = duration_min > 20
  ) %>%
  filter(period %in% c("2020-2022", "2023-2025")) %>%
  group_by(period, date, weekday) %>%
  summarise(has_long = any(over20), .groups = "drop") %>%
  group_by(period, weekday) %>%
  summarise(days_with_long = sum(has_long), .groups = "drop")

# 3. Combine & compute %
weekday_freq <- calendar_days %>%
  left_join(long_ride_days, by = c("period", "weekday")) %>%
  mutate(
    days_with_long     = coalesce(days_with_long, 0),
    pct_days_with_long = days_with_long / total_days * 100
  )

# 4. Plot with 0-100% y-axis and labels
my_plotFreq <- ggplot(weekday_freq, aes(x = weekday, y = pct_days_with_long, fill = period)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.7), width = 0.7) +
  geom_text(aes(label = paste0(round(pct_days_with_long), "%")),
            position = position_dodge(width = 0.7),
            vjust = -0.5, size = 3) +
  scale_y_continuous(
    limits = c(0, 100),
    breaks = seq(0, 100, by = 10),
    labels = function(x) paste0(x, "%"),
    expand = c(0, 0.02)
  ) +
  labs(
    title    = "Percent of Days of the Week with Strava Activity",
    subtitle = "By Weekday: 2020-2022 vs 2023-2025 (through Jul 7)",
    x        = "Day of Week",
    y        = "Percent of Days Exercised",
    fill     = "Period"
  ) +
  scale_fill_manual(values = c("2020-2022" = "#1f77b4", "2023-2025" = "#ff7f0e")) +
  theme_minimal(base_size = 13)

my_plotFreq
```




```
ggsave("pct_freq_by_weekday_labeled.jpg", plot = my_plotFreq, width = 9, height = 6, dpi = 300)
```