Bellabeat_Case_Study

2023-12-11

Setting up environment

Install packages.

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## 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 error
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
##
## here() starts at /cloud/project
##
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
##
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
##
##
## Attaching package: 'janitor'
##
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

Leaf tracks *activity*, *sleep* and *stress*. We will upload relevant data available. In this case, that's data on activity and sleep.

Cleaning

```
dailyActivity_merged <- read_csv("Fitabase Data 4.12.16-5.12.16/dailyActivity_merged.csv")
```

Start with daily activity data

```
## Rows: 940 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): ActivityDate
## dbl (14): Id, TotalSteps, TotalDistance, TrackerDistance, LoggedActivitiesDi...
##
## 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.
```

```
skim_without_charts(dailyActivity_merged)
```

Table 1: Data summary

Name	dailyActivity_merged
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
ActivityDate	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
TotalSteps	0	1	7.637910e+03	5.087150e+03	0	3.789750e+03	7.405500e+03	1.072700e+04	3.601900e+04
TotalDistance	0	1	5.490000e+00	3.920000e+00	0	2.620000e+00	5.240000e+00	7.710000e+00	2.803000e+01
TrackerDistance	0	1	5.480000e+00	3.910000e+00	0	2.620000e+00	5.240000e+00	7.710000e+00	2.803000e+01
LoggedActivitiesDistance	0	1	1.100000e-01	6.200000e-01	0	0.000000e+00	0.000000e+00	0.000000e+00	4.940000e+00
VeryActiveDistance	0	1	1.500000e+00	2.660000e+00	0	0.000000e+00	2.100000e-01	2.050000e+00	2.192000e+01
ModeratelyActiveDistance	0	1	5.700000e-01	8.800000e-01	0	0.000000e+00	2.400000e-01	8.000000e-01	6.480000e+00
LightActiveDistance	0	1	3.340000e+00	2.040000e+00	0	1.950000e+00	3.360000e+00	4.780000e+00	1.071000e+01
SedentaryActiveDistance	0	1	0.000000e+00	1.000000e-02	0	0.000000e+00	0.000000e+00	0.000000e+00	1.100000e-01
VeryActiveMinutes	0	1	2.116000e+01	3.284000e+01	0	0.000000e+00	4.000000e+00	3.200000e+01	2.100000e+02
FairlyActiveMinutes	0	1	1.356000e+01	1.999000e+01	0	0.000000e+00	6.000000e+00	1.900000e+01	1.430000e+02
LightlyActiveMinutes	0	1	1.928100e+02	1.091700e+02	0	1.270000e+02	1.990000e+02	2.640000e+02	5.180000e+02
SedentaryMinutes	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
glimpse(dailyActivity_merged)
```

```
## Rows: 940
## Columns: 15
## $ Id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036~
## $ ActivityDate             <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4/15/~
## $ TotalSteps               <dbl> 13162, 10735, 10460, 9762, 12669, 9705, 13019~
## $ TotalDistance            <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8~
## $ TrackerDistance          <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8~
## $ LoggedActivitiesDistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveDistance       <dbl> 1.88, 1.57, 2.44, 2.14, 2.71, 3.19, 3.25, 3.5~
## $ ModeratelyActiveDistance <dbl> 0.55, 0.69, 0.40, 1.26, 0.41, 0.78, 0.64, 1.3~
## $ LightActiveDistance      <dbl> 6.06, 4.71, 3.91, 2.83, 5.04, 2.51, 4.71, 5.0~
## $ SedentaryActiveDistance  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveMinutes        <dbl> 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66, 4~
## $ FairlyActiveMinutes      <dbl> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21~
## $ LightlyActiveMinutes     <dbl> 328, 217, 181, 209, 221, 164, 233, 264, 205, ~
## $ SedentaryMinutes         <dbl> 728, 776, 1218, 726, 773, 539, 1149, 775, 818~
## $ Calories                 <dbl> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 203~
```

Check for session IDs

```r
unique(dailyActivity_merged$Id)
```

```
##  [1] 1503960366 1624580081 1644430081 1844505072 1927972279 2022484408
##  [7] 2026352035 2320127002 2347167796 2873212765 3372868164 3977333714
## [13] 4020332650 4057192912 4319703577 4388161847 4445114986 4558609924
## [19] 4702921684 5553957443 5577150313 6117666160 6290855005 6775888955
## [25] 6962181067 7007744171 7086361926 8053475328 8253242879 8378563200
## [31] 8583815059 8792009665 8877689391
```

Check for missing values

```r
dailyActivity_merged %>% filter(!complete.cases(.))
```

```
## # A tibble: 0 x 15
## # i 15 variables: Id <dbl>, ActivityDate <chr>, TotalSteps <dbl>,
## #   TotalDistance <dbl>, TrackerDistance <dbl>, LoggedActivitiesDistance <dbl>,
## #   VeryActiveDistance <dbl>, ModeratelyActiveDistance <dbl>,
## #   LightActiveDistance <dbl>, SedentaryActiveDistance <dbl>,
## #   VeryActiveMinutes <dbl>, FairlyActiveMinutes <dbl>,
## #   LightlyActiveMinutes <dbl>, SedentaryMinutes <dbl>, Calories <dbl>
```

Check for N/A & confirm

```r
mean(dailyActivity_merged$TotalSteps)
```

```
## [1] 7637.911
```

```r
mean(dailyActivity_merged$TotalDistance)
```

```
## [1] 5.489702
```

```r
mean(dailyActivity_merged$TrackerDistance)
```

```
## [1] 5.475351
```

```r
mean(dailyActivity_merged$LoggedActivitiesDistance)
```

```
## [1] 0.1081709
```

```r
mean(dailyActivity_merged$VeryActiveDistance)
```

```
## [1] 1.502681
```

```r
mean(dailyActivity_merged$ModeratelyActiveDistance)
```

```
## [1] 0.5675426
```

```r
mean(dailyActivity_merged$LightActiveDistance)
```

```
## [1] 3.340819
```

```r
mean(dailyActivity_merged$SedentaryActiveDistance)
```

```
## [1] 0.001606383
```

```r
mean(dailyActivity_merged$VeryActiveMinutes)
```

```
## [1] 21.16489
```

```r
mean(dailyActivity_merged$FairlyActiveMinutes)
```

```
## [1] 13.56489
```

```r
mean(dailyActivity_merged$LightlyActiveMinutes)
```

```
## [1] 192.8128
```

```r
mean(dailyActivity_merged$SedentaryMinutes)
```

```
## [1] 991.2106
```

```r
mean(dailyActivity_merged$Calories)
```

```
## [1] 2303.61
```

Convert character to date

```r
dailyActivity_merged$ActivityDate <- mdy(dailyActivity_merged$ActivityDate)
```

Repeat same cleaning process for **hourly** data

```r
hourlyIntensities_merged <- read_csv("Fitabase Data 4.12.16-5.12.16/hourlyIntensities_merged.csv")
```

```
## Rows: 22099 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityHour
## dbl (3): Id, TotalIntensity, AverageIntensity
##
## 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
skim_without_charts(hourlyIntensities_merged)
```

Table 4: Data summary

Name	hourlyIntensities_merged
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
ActivityHour	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	2.4225e+09	1503960366	2320127002	4.445115e+09	6.962181e+09	8877689391
TotalIntensity	0	1	1.204000e+01	2.11130e+01	0	0	3.000000e+00	1.600000e+01	180
AverageIntensity	0	1	2.000000e-01	3.5000e-01	0	0	5.000000e-02	2.700000e-01	3

```
glimpse(hourlyIntensities_merged)
```

```
## Rows: 22,099
## Columns: 4
## $ Id               <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 15039~
## $ ActivityHour     <chr> "4/12/2016 12:00:00 AM", "4/12/2016 1:00:00 AM", "4/1~
## $ TotalIntensity   <dbl> 20, 8, 7, 0, 0, 0, 0, 0, 13, 30, 29, 12, 11, 6, 36, 5~
## $ AverageIntensity <dbl> 0.333333, 0.133333, 0.116667, 0.000000, 0.000000, 0.0~
```

```
unique(hourlyIntensities_merged$Id)
```

```
##  [1] 1503960366 1624580081 1644430081 1844505072 1927972279 2022484408
##  [7] 2026352035 2320127002 2347167796 2873212765 3372868164 3977333714
## [13] 4020332650 4057192912 4319703577 4388161847 4445114986 4558609924
## [19] 4702921684 5553957443 5577150313 6117666160 6290855005 6775888955
## [25] 6962181067 7007744171 7086361926 8053475328 8253242879 8378563200
## [31] 8583815059 8792009665 8877689391
```

```
hourlyIntensities_merged %>% filter(!complete.cases(.))
```

```
## # A tibble: 0 x 4
## # i 4 variables: Id <dbl>, ActivityHour <chr>, TotalIntensity <dbl>,
## #   AverageIntensity <dbl>
```

```
mean(hourlyIntensities_merged$TotalIntensity)
```

```
## [1] 12.03534
```

```
mean(hourlyIntensities_merged$AverageIntensity)
```

```
## [1] 0.200589
```

```
hourlyIntensities_merged$ActivityHour <- mdy_hms(hourlyIntensities_merged$ActivityHour)
```

And for **sleep** data

```
sleepDay_merged <- read_csv("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.
```

```
skim_without_charts(sleepDay_merged)
```

Table 7: Data summary

Name	sleepDay_merged
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
SleepDay	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
TotalSleepRecords	0	1	1.120000e+00	3.450000e-01	1	1	1	1	3
TotalMinutesAsleep	0	1	4.194700e+02	1.183400e+02	58	361	433	490	796
TotalTimeInBed	0	1	4.586400e+02	1.271000e+02	61	403	463	526	961

```
glimpse(sleepDay_merged)
```

```
## Rows: 413
## Columns: 5
## $ Id                 <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 150~
## $ SleepDay           <chr> "4/12/2016 12:00:00 AM", "4/13/2016 12:00:00 AM", "~
## $ TotalSleepRecords  <dbl> 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ TotalMinutesAsleep <dbl> 327, 384, 412, 340, 700, 304, 360, 325, 361, 430, 2~
## $ TotalTimeInBed     <dbl> 346, 407, 442, 367, 712, 320, 377, 364, 384, 449, 3~
```

```
unique(sleepDay_merged$Id)
```

```
##  [1] 1503960366 1644430081 1844505072 1927972279 2026352035 2320127002
##  [7] 2347167796 3977333714 4020332650 4319703577 4388161847 4445114986
## [13] 4558609924 4702921684 5553957443 5577150313 6117666160 6775888955
## [19] 6962181067 7007744171 7086361926 8053475328 8378563200 8792009665
```

```
sleepDay_merged %>% filter(!complete.cases(.))
```

```
## # A tibble: 0 x 5
## # i 5 variables: Id <dbl>, SleepDay <chr>, TotalSleepRecords <dbl>,
## #   TotalMinutesAsleep <dbl>, TotalTimeInBed <dbl>
```

We have less session ID participation in sleep dataset.

Hence, we will focus on **daily** and **hourly** *activity* data. More comprehensive.

Analysis : Very Active Days

Select relevant data columns that help answer business task

```
dailyActivity_merged <- dailyActivity_merged %>% select(Id, ActivityDate,
                                                        VeryActiveMinutes,
                                                        FairlyActiveMinutes,
                                                        LightlyActiveMinutes)
```

Focus on **very active** users and their day preferences

Identify these *very active* users. [CDC] (https://health.gov/sites/default/files/2019-09/Physical_Activity_Guidelines_2nd_edition.pdf).

```
VeryActiveUsers <- dailyActivity_merged %>%
  group_by(Id) %>%
  summarize(VeryMeanActiveMinutes = mean(VeryActiveMinutes))
VeryActiveUsers <- VeryActiveUsers %>% filter(VeryMeanActiveMinutes > 10)
```

Must also establish relationship between *intensity* & *very active* metrics

Identify these *very intense* users.

```
VeryIntenseUsers <- hourlyIntensities_merged %>%
  group_by(Id) %>%
  summarize(TotalMeanIntensity = mean(TotalIntensity))
VeryIntenseUsers <- VeryIntenseUsers %>% filter(TotalMeanIntensity > 12)
```

Confirm. Do they mostly match? Yes

```
FullVeryActiveUserdata <- inner_join(VeryActiveUsers, VeryIntenseUsers)
```

```
## Joining with `by = join_by(Id)`
```

About half of fitbit users identify as **very active**

Now, what are their day preferences?

Extract Ids and join with daily data table

```
FullVeryActiveUserdata <- FullVeryActiveUserdata %>% select(Id)
dailyActivity_merged <- inner_join(FullVeryActiveUserdata, dailyActivity_merged)
```

```
## Joining with `by = join_by(Id)`
```

```
unique(dailyActivity_merged$Id)
```

```
##  [1] 1503960366 2022484408 2347167796 2873212765 3977333714 4388161847
##  [7] 4558609924 5553957443 5577150313 6962181067 7007744171 7086361926
## [13] 8053475328 8378563200 8877689391
```

Change date to weekdays

```
dailyActivity_merged$ActivityDate <- weekdays(dailyActivity_merged$ActivityDate)
colnames(dailyActivity_merged)[2] = "Day"
```

Group data by day

```
DayPreferences <- dailyActivity_merged %>%
  group_by(Day) %>%
  summarize(VeryActiveMeanMinutes = mean(VeryActiveMinutes))
```

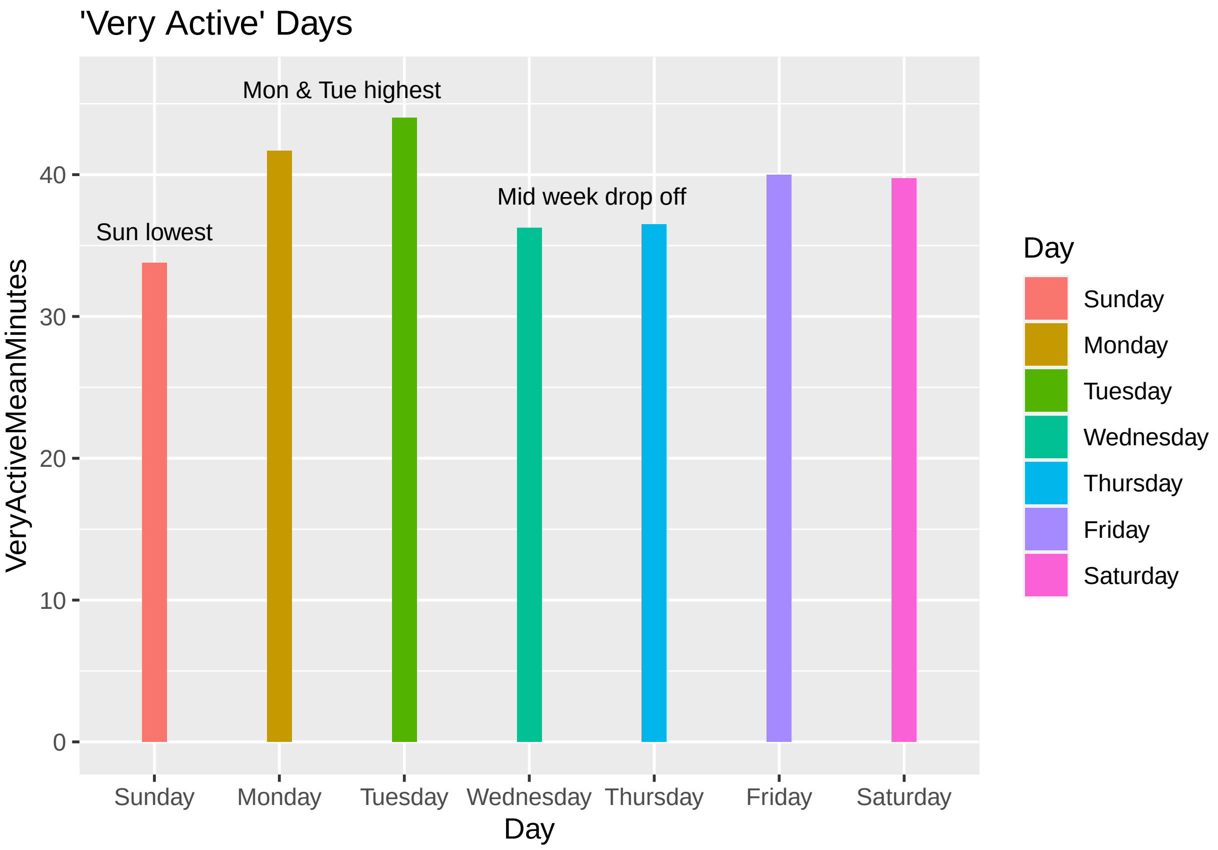
Let's order the data

```
DayPreferences$Day <- factor(DayPreferences$Day, levels = c("Sunday", "Monday",
                                                            "Tuesday",
                                                            "Wednesday",
                                                            "Thursday",
                                                            "Friday",
                                                            "Saturday"))
DayPreferences[order(DayPreferences$Day), ]
```

```
## # A tibble: 7 x 2
##   Day       VeryActiveMeanMinutes
##   <fct>                     <dbl>
## 1 Sunday                     33.8
## 2 Monday                     41.7
## 3 Tuesday                    44.0
## 4 Wednesday                  36.2
## 5 Thursday                   36.5
## 6 Friday                     40
## 7 Saturday                   39.7
```

Data Visualization: “Very Active” day preferences

```
ggplot(data = DayPreferences, aes(x=Day,y=VeryActiveMeanMinutes, fill = Day)) +
  geom_bar(stat = 'identity', width = 0.2) +
  labs(title = "'Very Active' Days") +
  annotate("text", x = 2.5, y = 46, label = "Mon & Tue highest", size = 3) +
  annotate("text", x = 1, y = 36, label = "Sun lowest", size = 3) +
  annotate("text", x = 4.5, y = 38.5, label = "Mid week drop off", size = 3)
```


'Very Active' Days

Analysis : Hour

What are very active users' hourly preferences?

Join extracted Ids with hourly data

```
hourlyIntensities_merged <- inner_join(FullVeryActiveUserdata, hourlyIntensities_merged)
```

```
## Joining with `by = join_by(Id)`
```

Convert datetime into day and time columns

```
hourlyIntensities_merged$Date <- as.Date(hourlyIntensities_merged$ActivityHour)
hourlyIntensities_merged$Time <- format(hourlyIntensities_merged$ActivityHour,"%H:%M:%S")
hourlyIntensities_merged$Date <- weekdays(hourlyIntensities_merged$Date)
colnames(hourlyIntensities_merged)[5] = "Day"
```

Group data by hour

```
Hourlypreferences <- hourlyIntensities_merged %>%
  group_by(Time) %>%
  summarize(TotalMeanIntensity = mean(TotalIntensity))
```

Convert military time to am / pm

```
Hourlypreferences$Time <- format(strptime(Hourlypreferences$Time, format = '%H:%M:%S'),'%I %p')
```

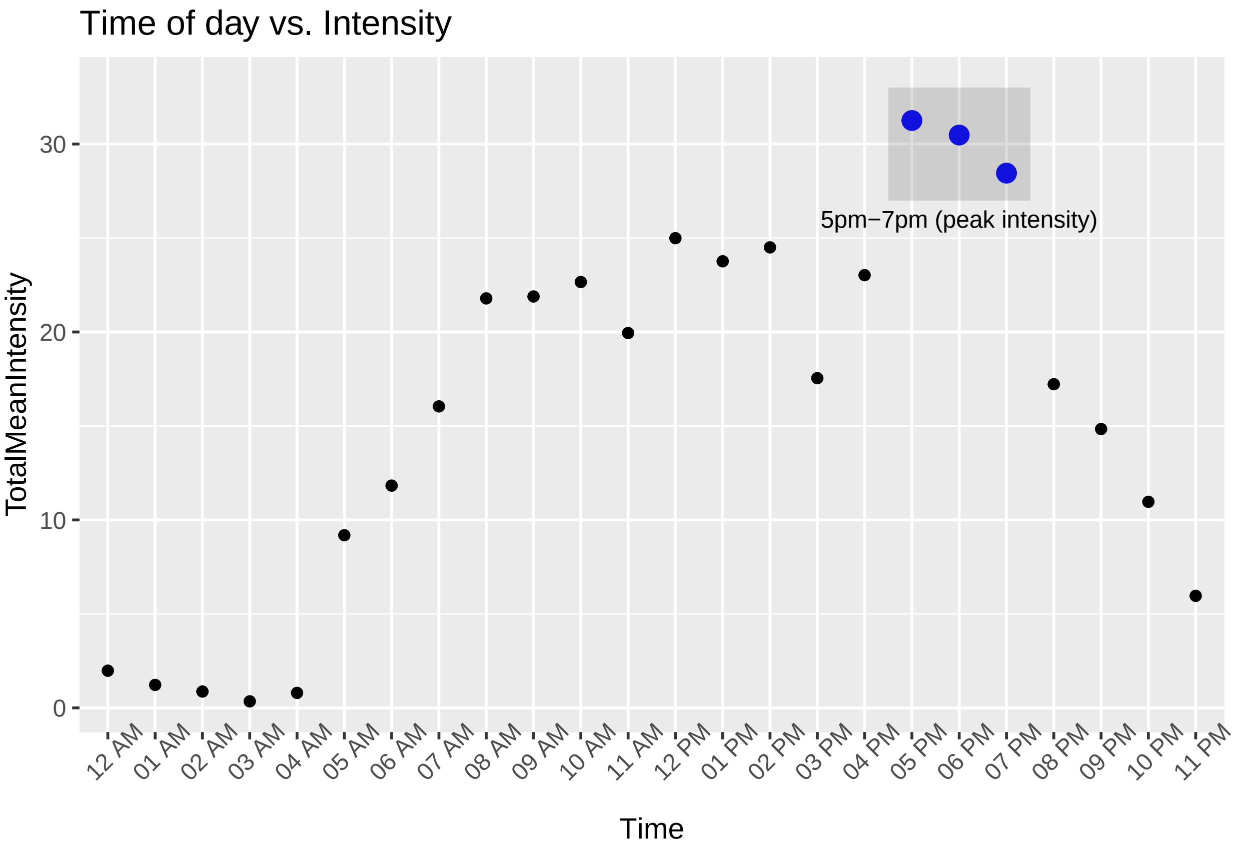
Order

```
Hourlypreferences$Time <- factor(Hourlypreferences$Time, levels = c("12 AM", "01 AM", "02 AM","03 AM","
Hourlypreferences[order(Hourlypreferences$Time), ]
```

```
## # A tibble: 24 x 2
##    Time  TotalMeanIntensity
##    <fct>              <dbl>
##  1 12 AM              1.98
##  2 01 AM              1.22
##  3 02 AM              0.871
##  4 03 AM              0.348
##  5 04 AM              0.799
##  6 05 AM              9.19
##  7 06 AM             11.8
##  8 07 AM             16.0
##  9 08 AM             21.8
## 10 09 AM             21.9
## # i 14 more rows
```

Plot: Time of day vs. Intensity

```
Hourlypreferences_most <- Hourlypreferences %>% filter(TotalMeanIntensity > 25)
p <- ggplot(data = Hourlypreferences) +
  geom_point(mapping = aes(x=Time,y=TotalMeanIntensity)) +
  geom_point(data = Hourlypreferences_most,
             aes(x=Time, y=TotalMeanIntensity),
             color="blue",
             size=3) +
  labs(title = "Time of day vs. Intensity")
p + theme(axis.text.x = element_text(angle = 45)) +
annotate("text", x = 19, y = 26, label = "5pm-7pm (peak intensity)", size = 3) +
annotate("rect", xmin = 17.5, xmax = 20.5, ymin = 27, ymax = 33,
  alpha = .2)
```



Does Day Intensity match up with DayPreferences data? Let's see

Analysis: Day Intensity

```
DayIntensity <- hourlyIntensities_merged %>%
  group_by(Day) %>%
  summarize(TotalMeanIntensity = mean(TotalIntensity))
```

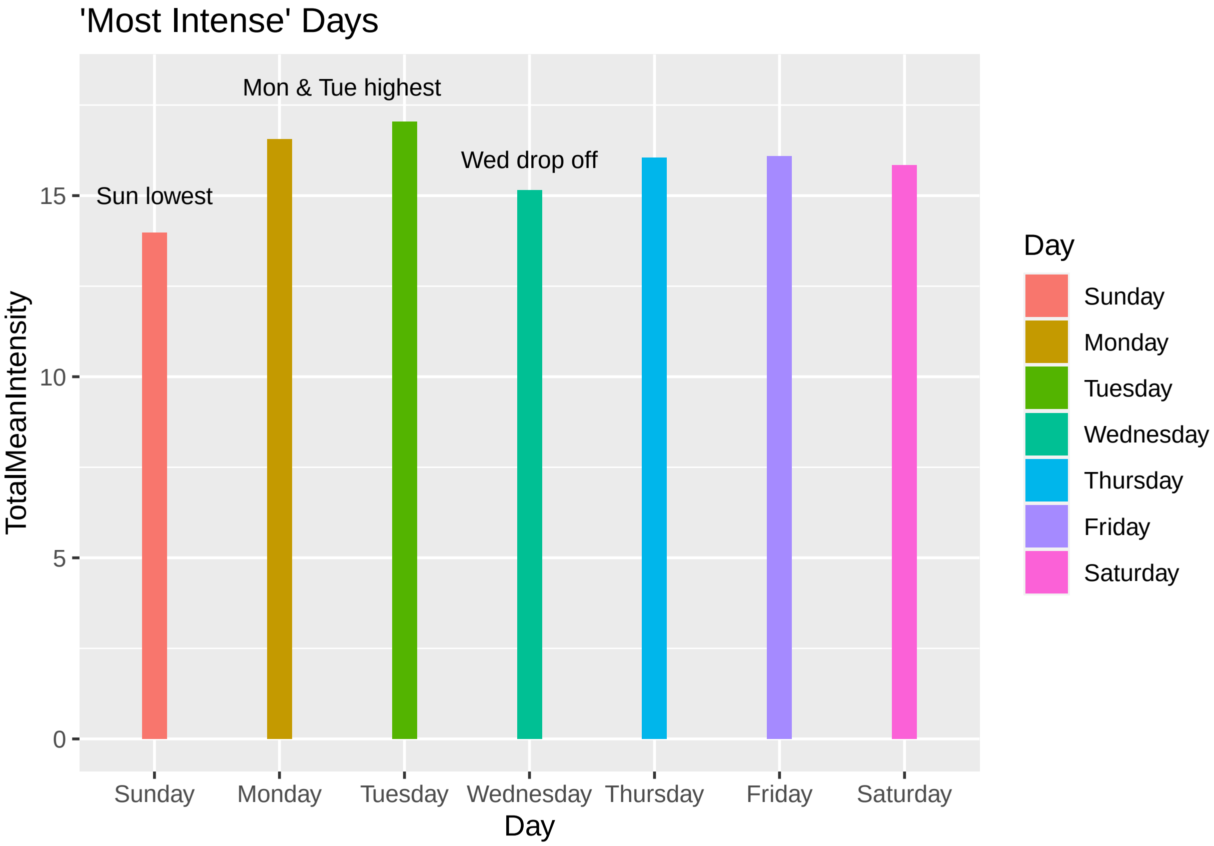
Order

```
DayIntensity$Day <- factor(DayIntensity$Day, levels = c("Sunday", "Monday",
                                                        "Tuesday", "Wednesday",
                                                        "Thursday", "Friday",
                                                        "Saturday"))
DayIntensity[order(DayIntensity$Day), ]
```

```
## # A tibble: 7 x 2
##   Day       TotalMeanIntensity
##   <fct>                  <dbl>
## 1 Sunday                  14.0
## 2 Monday                  16.6
## 3 Tuesday                 17.0
## 4 Wednesday               15.2
## 5 Thursday                16.0
## 6 Friday                  16.1
## 7 Saturday                15.8
```

Data Visualization: Day Intensity

```r
ggplot(data = DayIntensity, aes(x=Day,y=TotalMeanIntensity, fill = Day)) +
  geom_bar(stat = 'identity', width = 0.2) +
  labs(title = "'Most Intense' Days") +
  annotate("text", x = 1, y = 15, label = "Sun lowest", size = 3) +
  annotate("text", x = 2.5, y = 18, label = "Mon & Tue highest", size = 3) +
  annotate("text", x = 4, y = 16, label = "Wed drop off", size = 3)
```



Once again, relationship between *intensity* & *very active* metrics established.

KEY TAKEAWAYS

- **Monday** and **Tuesday** are consistently strongest days for "very active" users
- **Sunday** and **Wednesday** are weakest
- **5pm to 7pm** are the most active times for these users