Bellabeat Capstone Project with R

Introduction

This project is the investigation of the data collected by Bellabeat, a high-tech manufacturer of health-focused products for women. The dataset collected from thirty-three eligible Fitbit users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring. The main goal is to gain insight into how consumers are using the company smart devices.

Loading and Cleaning Datasets

Loading Libraries

First, load all the libraries for data analysis and visualization.

```{r library, message=FALSE, warning=FALSE} library("tidyverse") library("lubridate") library("viridis")

library("ggrepel") library("GGally")

<br />
<br />

#### Loading Datasets

```{r load}
dailyActivity_merged <- read.csv("/kaggle/input/fitbit/Fitabase Data 4.12.16-5.12.16/dailyActivity_merged.csv")
dailyCalories_merged <- read.csv("/kaggle/input/fitbit/Fitabase Data 4.12.16-5.12.16/dailyCalories_merged.csv")
dailyIntensities_merged <- read.csv("/kaggle/input/fitbit/Fitabase Data 4.12.16-5.12.16/dailyIntensities_merged.csv")
dailySteps_merged <- read.csv("/kaggle/input/fitbit/Fitabase Data 4.12.16-5.12.16/dailySteps_merged.csv")
hourlyCalories_merged <- read.csv("/kaggle/input/fitbit/Fitabase Data 4.12.16-5.12.16/hourlyCalories_merged.csv")
hourlyIntensities_merged <- read.csv("/kaggle/input/fitbit/Fitabase Data 4.12.16-5.12.16/hourlyIntensities_merged.csv")
```
```

```
hourlySteps_merged <- read.csv("/kaggle/input/fitbit/Fitabase Data
4.12.16-5.12.16/hourlySteps_merged.csv")
minuteCaloriesNarrow_merged <- read.csv("/kaggle/input/fitbit/Fitabase
Data 4.12.16-5.12.16/minuteCaloriesNarrow_merged.csv")
minuteCaloriesWide_merged <- read.csv("/kaggle/input/fitbit/Fitabase
Data 4.12.16-5.12.16/minuteCaloriesWide_merged.csv", header=FALSE)
minuteIntensitiesNarrow_merged <-
read.csv("/kaggle/input/fitbit/Fitabase Data
4.12.16-5.12.16/minuteIntensitiesNarrow_merged.csv")
minuteIntensitiesWide_merged <-
read.csv("/kaggle/input/fitbit/Fitabase Data
4.12.16-5.12.16/minuteIntensitiesWide_merged.csv", header=FALSE)
minuteMETsNarrow_merged <- read.csv("/kaggle/input/fitbit/Fitabase
Data 4.12.16-5.12.16/minuteMETsNarrow_merged.csv")
minuteSleep_merged <- read.csv("/kaggle/input/fitbit/Fitabase Data
4.12.16-5.12.16/minuteSleep_merged.csv")
minuteStepsNarrow_merged <- read.csv("/kaggle/input/fitbit/Fitabase
Data 4.12.16-5.12.16/minuteStepsNarrow_merged.csv")
minuteStepsWide_merged <- read.csv("/kaggle/input/fitbit/Fitabase Data
4.12.16-5.12.16/minuteStepsWide_merged.csv", header=FALSE)
sleepDay_merged <- read.csv("/kaggle/input/fitbit/Fitabase Data
4.12.16-5.12.16/sleepDay_merged.csv")
weightLogInfo_merged <- read.csv("/kaggle/input/fitbit/Fitabase Data
4.12.16-5.12.16/weightLogInfo_merged.csv")
```

Check for Missing Values

```{r , results = “hold”} sum(is.na(dailyActivity_merged)) sum(is.na(dailyCalories_merged))
sum(is.na(dailyIntensities_merged)) sum(is.na(dailySteps_merged))

sum(is.na(hourlyCalories_merged)) sum(is.na(hourlyIntensities_merged))
sum(is.na(hourlySteps_merged)) sum(is.na(minuteCaloriesNarrow_merged))
sum(is.na(minuteCaloriesWide_merged)) sum(is.na(minuteIntensitiesNarrow_merged))
sum(is.na(minuteIntensitiesWide_merged)) sum(is.na(minuteMETsNarrow_merged))
sum(is.na(minuteSleep_merged)) sum(is.na(minuteStepsNarrow_merged))
sum(is.na(minuteStepsWide_merged)) sum(is.na(sleepDay_merged))
sum(is.na(weightLogInfo_merged))

<br />
<br />
Only weightLogInfo_merged data frame has 65 missing values while the
other data frames have none of it.
<br />
<br />
```

Check and Remove Duplication


```{r , results = "hold"}
sum(duplicated(dailyActivity_merged))
sum(duplicated(dailyCalories_merged))
sum(duplicated(dailyIntensities_merged))
sum(duplicated(dailySteps_merged))

sum(duplicated(hourlyCalories_merged))
sum(duplicated(hourlyIntensities_merged))
sum(duplicated(hourlySteps_merged))
sum(duplicated(minuteCaloriesNarrow_merged))
sum(duplicated(minuteCaloriesWide_merged))
sum(duplicated(minuteIntensitiesNarrow_merged))
sum(duplicated(minuteIntensitiesWide_merged))
sum(duplicated(minuteMETsNarrow_merged))
sum(duplicated(minuteSleep_merged))
sum(duplicated(minuteStepsNarrow_merged))
sum(duplicated(minuteStepsWide_merged))
sum(duplicated(sleepDay_merged))
sum(duplicated(weightLogInfo_merged))
```

There're some duplication on sleepDay_merged and minuteSleep_merged data frames. So let's remove them all. ```{r } sleepDay_merged <- sleepDay_merged[! duplicated(sleepDay_merged),]

minuteSleep_merged <- minuteSleep_merged[!duplicated(minuteSleep_merged),]

Exploring Datasets

First, let's see the number of users (Id) in each data frame.


```{r , results = "hold"}
n_distinct(dailyActivity_merged$Id)
n_distinct(dailyCalories_merged$Id)
n_distinct(dailyIntensities_merged$Id)
n_distinct(dailySteps_merged$Id)

n_distinct(hourlyCalories_merged$Id)
n_distinct(hourlyIntensities_merged$Id)
```

```
n_distinct(hourlySteps_merged$Id)
n_distinct(minuteCaloriesNarrow_merged$Id)
n_distinct(minuteCaloriesWide_merged$Id)
n_distinct(minuteIntensitiesNarrow_merged$Id)
n_distinct(minuteIntensitiesWide_merged$Id)
n_distinct(minuteMETsNarrow_merged$Id)
n_distinct(minuteSleep_merged$Id)
n_distinct(minuteStepsNarrow_merged$Id)
n_distinct(minuteStepsWide_merged$Id)
n_distinct(sleepDay_merged$Id)
n_distinct(weightLogInfo_merged$Id)
```

Wide_merged data frames have 0 number of Id probably because they are modified version of Narrow_merged data frames with additional column. All the columns are renamed to capital V with tag number, Id column are rename to V1. In that column ,the name “Id” itself are underneath the new header V1

```
 {r , results = "hold"} n_distinct(minuteCaloriesWide_merged$V1)
n_distinct(minuteIntensitiesWide_merged$V1)
n_distinct(minuteStepsWide_merged$V1)
```

So each Wide merged table indicate 33 participants (not included second column header).

- There’re only 8 users(Id) in weightLogInfo_merged
- sleepDay_merged and minuteSleep_merged both have 24 users participated.
- 14 users participate in heartrate_seconds_merged.
- All the other data frames have 33 users.

Exploring Daily-Based Data Frames

Now let’s check the consistency of all daily-based data frame to see whether or not each users have the same number of days participated in every daily-based data frame. If the number are the same, we can merge all daily-base data frame into one.

```{r } dailyActivity_merged %>%
group_by(Id) %>% summarise(count = n_distinct(ActivityDate))

dailyCalories_merged %>%
group_by(Id) %>% summarise(count = n_distinct(ActivityDay))

dailyIntensities_merged %>% group_by(Id) %>% summarise(count =
n_distinct(ActivityDay))

dailySteps_merged %>% group_by(Id) %>% summarise(count = n_distinct(ActivityDay))

<br />
<br />
This turns out to be number of day recorded for each user are all the
same in all these four data frame. So we can combine them into one but
it seem to be that only the data frame dailyActivity_merged already
have all the data the other three data frame have. So using only
```

```
dailyActivity_merged would be enough.
<br />
<br />


#### Exploring Hourly-Based Data Frames


<br />
Applying the same process with all hourly-based table. Check the
number of time recorded for each users.
<br />

```{r }
hourlyCalories_merged %>%
  group_by(Id) %>%
  summarise(count = n_distinct(ActivityHour))

hourlyIntensities_merged %>%
  group_by(Id) %>%
  summarise(count = n_distinct(ActivityHour))

hourlySteps_merged %>%
  group_by(Id) %>%
  summarise(count = n_distinct(ActivityHour))
```

All these three tables have the same set of Id and each Id also has the same number of recorded hours.

Preparing Data Frames for further Analysis

Organize Recorded Time in each Data Frame

Before processing daily-based and hourly-based data frames. Let's first prepare some other data frames that will be used individually.
We start by organizing time recorded in heartrate_seconds_merged table.

Then extract date as "ActivityDay" using date() ` Check days of participation for each users.

Only 14 participants with days of participation vary from 4 to 31

Applying the same process with minuteSleep_merged table. `{r }`
`minuteSleep_merged$date <- parse_date_time( minuteSleep_merged$date, "%m/%d/%y %I:%M:%S %p" )` Organize time recorded in minuteMETsNarrow_merged table from character class into date time class.
```

{r } minuteMETsNarrow_merged$ActivityMinute <- parse_date_time(minuteMETsNarrow_merged$ActivityMinute, "%m/%d/%y %I:%M:%S %p") And also for dailyActivity_merged

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

<br />
<br />
METs value need to be divided by 10, and then add more columns with extracted date and hour from ActivityMinute. The data from this minuteMETsNarrow_merged table will later be incorporated into daily and hourly based tables.
<br />
```{r }
minuteMETsNarrow_merged <-
  mutate(minuteMETsNarrow_merged, METs = METs/10,
         date = date(ActivityMinute),
         ActivityHour = floor_date(ActivityMinute, "hour"))
```

Creating METs Daily Data Frame

For daily METs value, we use average average METs value calculated by mean function. {r } METs_daily <- minuteMETsNarrow_merged %>% group_by(Id,date) %>% summarize(METs = mean(METs), .groups='drop') This table will be combined with dailyActivity_merged, but the number of day in this table contain 6 row less than dailyActivity_merged. A little bit of data will be lost, but the combination will give bigger and more complex picture of how all these data related.

{r } Activity_METs_daily <- inner_join(dailyActivity_merged, METs_daily, by = c("Id","ActivityDate" = "date")) Besides using this table for creating box plot, We would also use it for creating bar chart.

Add weekday colomn into this table.

```
Activity_METs_daily <- Activity_METs_daily %>%
  mutate(Weekday = wday(ActivityDate, label = T,
                        week_start = 1))
```

Prepare Sleep Data Frame Before Merging

We incorporate sleepDay_merged table into daily-based tables because we want to see how sleeping time can have some effect on other values such as calories burned, METs, etc. {r } sleepDay_merged %>% group_by(Id) %>% summarise(count = n_distinct(SleepDay)) There're only 24 users with only 410 rows in SleepDay_merged table compared to 33 users with 940 rows on other daily-based tables. To combine them together we would loose significant amount of data. So we only use this combination only to see the effect of sleeping on other values.
```

{r } sleepDay_merged <- sleepDay_merged %>% mutate(DateTime = mdy_hms(SleepDay))

Merging and Creating Data Frame

{r } Activity_Sleep_daily <- inner_join(dailyActivity_merged, sleepDay_merged, by = c("Id","ActivityDate" = "DateTime")) Adding sleep efficiency column, the value is the percentage of total minutes asleep divided by total time in bed. Normal sleep efficiency is considered to be 80% or greater. {r } Activity_Sleep_daily <- Activity_Sleep_daily %>% mutate(Sleep_Efficiency = ((TotalMinutesAsleep/TotalTimeInBed)*100)) Then combine Activity_Sleep_daily table with METs_daily to be SMAd (Sleep_METs_Activity_daily) {r } SMAd <- inner_join(Activity_Sleep_daily, METs_daily, by = c("Id","ActivityDate" = "date")) Add weekday column to SMAd

{r } SMAd <- SMAd %>% mutate(Weekday = wday(ActivityDate, label = T, week_start = 1)) SMAd table will be use for almost every daily based plot

Create data frame Activity_METs_daily_long for a chart that display each activity minutes into box plot ```{r } Activity_METs_daily_long <- Activity_METs_daily %>% pivot_longer(cols = c('VeryActiveMinutes', 'FairlyActiveMinutes', 'LightlyActiveMinutes', 'SedentaryMinutes'), names_to='ActivityMinutes', values_to='Minutes')

It's done for daily-based tables. Now, let's try to join all hourly-based tables together.


```{r }
list_df <- list(hourlyCalories_merged,
                hourlyIntensities_merged,
                hourlySteps_merged)

hourly_outer_join <- list_df %>%
  reduce(full_join, by= c('Id','ActivityHour'))
```

Change time format from character to datetime {r } hourly_outer_join$ActivityHour <- parse_date_time(hourly_outer_join$ActivityHour, "%m/%d/%y %I:%M:%S %p") Add more columns from extracting ActivityHour into time, date, day, and weekday from datetime

{r } hourly_outer_join <- hourly_outer_join %>% mutate(Weekday = wday(ActivityHour, label = T, week_start = 1), date = date(ActivityHour), Day = day(ActivityHour),

Time = format(ActivityHour, "%k")) Create hourly_METs data frame for merging with hourly-based tables. {r } hourly_METs <- minuteMETsNarrow_merged %>% group_by(Id,ActivityHour) %>% summarize(METs = mean(METs), .groups='drop') Combining hourly_METs table with other hourly-based table called hourly_outer_joint_1 {r } hourly_outer_join_1 <- inner_join(hourly_outer_join, hourly_METs, by = c("Id","ActivityHour")) Create data frame for labeling pie chart from the table minuteIntensitiesNarrow_merged. This label indicated the percentage of each intensity value based on time spending for each users. We call this data frame df_1

```{r } df_1 <- minuteIntensitiesNarrow_merged %>% group_by(Id) %>% mutate(countId= n()) %>% ungroup %>% group_by(Id,countId,Intensity) %>% summarise(count=n(), .groups='drop' ) %>% group_by(Id) %>% mutate(cumcount=cumsum(count), pos=cumsum(count)-count/2, per=paste0(round(100*count/countId,2),'%')) %>% ungroup

<br />
<br />
And for the data frame minuteSleep_merged will be used in plotting pie chart, we also need another data frame called df_2 for labeling each segment as well.
<br />
```{r }
df_2 <- minuteSleep_merged %>%
  group_by(Id) %>%
  mutate(countId= n()) %>%
  ungroup %>%
  group_by(Id,countId,value) %>%
  summarise(count=n(),
            .groups='drop' ) %>%
  group_by(Id) %>%
  mutate(cumcount=cumsum(count),
         pos=cumsum(count)-count/2,
         per=paste0(round(100*count/countId,2),'%')) %>%
  ungroup
```

In this project, all 33 participants will be categorized based on their participation and time spending on intense activities. To make it easier each Id will be replaced by user number from 01 to 33 by creating another column that tags each Id with simplified number.

```{r } thirtythree <- c("01", "02", "03", "04", "05", "06", "07", "08", "09", 10:33)

Users <- paste("User", thirtythree, sep = " ")

Id <- unlist(distinct(dailyActivity_merged, Id))

users_id <- tibble(Users, Id)

<br />
<br />
Then attach this newly created column to dailyActivity_merged table.

This modified data frame will be used to create two more columns that categorized users based on days of participation and level of intensity.

<br />

Participation level of each user will be based on number of days recorded in this table. Activity level of each user will be based on the median value of "very active minutes" recorded in dailyActivity_merged table. which means Users with high activity level, or spend more minutes each day with high intensity activity are likely to do exercise regularly.
<br />

```{r }

dailyActivity_merged <- dailyActivity_merged %>%
  left_join(users_id, by = "Id")
```

Create a data frame with 2 column called ALPL, which stands for Activity Level and Participation Level. This data frame can be attached to any data frames in this study to put tag on each user to see how they spend their time on intense activity, and how often they participated in the program.

```{r } ALPL <- dailyActivity_merged %>% group_by(Users, Id) %>% summarize( MedVeryActive = median(VeryActiveMinutes), UsageRecords = n_distinct(ActivityDate), .groups='drop' ) %>% mutate( ActivityLevel = case_when( MedVeryActive < 4 ~ "Low", MedVeryActive <= 32 ~ "Med", MedVeryActive < 211 ~ "High" ), ParticipationLevel = case_when( UsageRecords < 14 ~ "Low Usage", UsageRecords < 21 ~ "Moderate Usage", UsageRecords < 31 ~ "High Usage", UsageRecords == 31 ~ "Daily Usage", ), ActivityLevel = factor( ActivityLevel, levels=c('Low', 'Med', 'High') ), ParticipationLevel = factor( ParticipationLevel, levels=c('Low Usage','Moderate Usage', 'High Usage','Daily Usage') ) ) %>% subset(select = -c(MedVeryActive,UsageRecords))
```


Our activity level is based the median value of very active minutes each user spent in each day, which can tell how much exercise they do on regular basis. We categorize activity level into "High" for users who have median value of time over 32 minutes each day spending on intense activity, "Med" for users with median value from over 4 minutes to 32 spending on intense activity, and "Low" for users who spent 4 minutes or less.

For participation level, we categorize users based on number of days, from dailActivity_merged table, each user has their data recorded in this table. We have "Daily Usage" for users who have their data recorded everyday from beginning to the end to program, "High Usage" for users who participate from 21 to 30 days, "Moderate Usage" for 14 to 20 days, and "Low Usage" for users who participate less than 14 days.

Attach this ALPL data frame to all the data frames we need to analyze on participation and intensity. So we can see how intensity level and participation level of user can tell about the different in other outcome.


```{r }
hourly_outer_join_1_ALPL <- hourly_outer_join_1 %>%
  left_join(ALPL, by = "Id")

Activity_METs_daily_ALPL <- Activity_METs_daily %>%
  left_join(ALPL, by = "Id")

SMAd_ALPL <- SMAd %>%
  left_join(ALPL, by = "Id")

minuteIntensitiesNarrow_merged_ALPL <-
  minuteIntensitiesNarrow_merged %>%
  left_join(ALPL, by = "Id")

minuteSleep_merged_ALPL <- minuteSleep_merged %>%
  left_join(ALPL, by = "Id")

df_1_ALPL <- df_1 %>%
  left_join(ALPL, by = "Id")

df_2_ALPL <- df_2 %>%
  left_join(ALPL, by = "Id")

Activity_METs_daily_long_ALPL <- Activity_METs_daily_long %>%
  left_join(ALPL, by = "Id")



weightLogInfo_merged_ALPL <- weightLogInfo_merged %>%
  left_join(ALPL, by = "Id")
```

Visualization and Analysis

Intensity Distribution

By User, Activity Level, Participation Level

```{r } cp <- coord_polar(theta = “y”) cp$is_free <- function() TRUE

<br />
<br />

```{r intensity_pie, fig.height = 16, fig.width = 12, fig.align = "center"}
options(ggrepel.max.overlaps = Inf)

minuteIntensitiesNarrow_merged_ALPL %>%
  ggplot(aes(x=1, fill=factor(Intensity))) +
  geom_bar(position = "fill", width = 1) +
  cp +
  scale_fill_manual(values=c("dodgerblue",
                             "olivedrab1",
                             "orangered",
                             "yellow1")) +

  facet_wrap(~Users~ActivityLevel~ParticipationLevel
             , scales = "free") +
  geom_label_repel(data = df_1_ALPL,
                   aes(label = per, y=pos),
                   position = position_fill(vjust = 0.5),
                   size = 3.0,
                   show.legend = FALSE) +
  theme(aspect.ratio = 1) +
  guides(fill = guide_legend(title = "Intensity")) +
  theme_classic() +
  theme(axis.text.y = element_blank(),
        strip.background = element_rect(fill = "thistle1"),
        legend.position = "bottom") +
  labs(title="Intensity Distribution",
       subtitle="By User, Activity Level, Participation Level",
       y="Intensity Percentage")
```

- Intensity value.(0 = Sedentary, 1 = Light, 2 = Moderate, 3 = Very Active)
```

- Above each pie labeled with user number, activity level and participation level. This pie chart have all the users which can be use to located level of activity and participation for every single one of them.
- Pie charts of users with high activity level always have proportion of intensity value as very active (3) more than 2%.
- Most users with medium activity level have proportion of intensity value of 3 between 1% to 2%, except user 03, user 09, user 11 and user 18. User 11 still have proportion of intensity value of 3 and 2 combined less than 2%, less than some users labeled as low activity level.

Hourly Box Plot of Calories, Intensity, Step and METS Distribution over Day

```{r boxplot1, fig.height = 3, fig.width = 10, fig.align = "center"} w <- hourly_outer_join_1_ALPL %>% ggplot() + theme(plot.title=element_text(size=11), axis.title=element_text(size=9), axis.text=element_text(size=8), axis.text.x = element_text(angle = 90,vjust=0.5))

w + geom_boxplot(aes(Time, Calories), fill="burlywood4", width=0.7) + labs(x='Time', y='Calories', title = "Calories vs Time of Day")

w + geom_boxplot(aes(Time, TotalIntensity), fill="royalblue4", width=0.7) + labs(x='Time', y='Total Intensity', title = "Total Intensity vs Time of Day")

w + geom_boxplot(aes(Time, StepTotal), fill="red1", width=0.7) + labs(x='Time', y='Total Steps', title = "Total Steps vs Time of Day")

w + geom_boxplot(aes(Time, METs), fill="olivedrab1", width=0.7) + labs(x='Time', y='METs', title = "METs vs Time of Day")

```


* As we can see, there are too many outliers. For any hourly-based chart, median value would be more appropriate than mean value for analysis.
* In this hourly-based table we use. TotalIntensity means value calculated by adding all the minute-level intensity values that occurred within the hour. But in the data frame dailyIntensities_merge, there is no column of intensity value. Instead there are only columns of time spending and distance based on intensity level.

Heat Map for Hourly Median Calories, Intensity, Steps and METs Distribution per Weekday


```{r heatmap1, fig.height = 3, fig.width = 10, fig.align = "center"}
options(repr.plot.width=20, repr.plot.height=80)

z1 <- hourly_outer_join_1_ALPL %>%
  group_by(Weekday, Time) %>%
  summarize(medCal = round(median(Calories)),
            medInt = round(median(TotalIntensity)),
            medStep = round(median(StepTotal)),
            medMETs = round(median(METs), digits = 2),
            .groups='drop') %>%
  ggplot(aes(Time, Weekday)) +
  scale_y_discrete(limits = rev) +
  theme(axis.text.x = element_text(angle = 90)) +
  guides(fill = guide_colourbar(barwidth = 0.45,
                                barheight = 6))


  z1 + geom_tile(aes(fill=medCal),colour = "white")  +
    geom_text(aes(label = medCal),
              size = 2.5, color = "black") +
    scale_fill_viridis(option = "H",
                       name = "Median Calories",
                       guide = guide_colorbar(
                         title.position = "top",
                         direction = "vertical")) +
    labs(x='Time',
         y='Weekday',
         title = "Hourly Median Calories Distribution per Weekday")


  z1 + geom_tile(aes(fill=medInt),colour = "white")  +
    geom_text(aes(label = medInt),
              size = 2.5, color = "black") +
    scale_fill_viridis(option = "H",
                       name = "Median Intensity",
                       guide = guide_colorbar(
                         title.position = "top",
                         direction = "vertical")) +
    labs(x='Time',
         y='Weekday',
         title = "Hourly Median Intensity Distribution per Weekday")
```

```
z1 + geom_tile(aes(fill=medStep),colour = "white")  +
  geom_text(aes(label = medStep),
            size = 2.5, color = "black") +
  scale_fill_viridis(option = "H",
                     name = "Median Steps",
                     guide = guide_colorbar(
                       title.position = "top",
                       direction = "vertical")) +
  labs(x='Time',
       y='Weekday',
       title = "Hourly Median Steps Distribution per Weekday")

z1 + geom_tile(aes(fill=medMETs),colour = "white")  +
  geom_text(aes(label = medMETs),
            size = 2.5, color = "black") +
  scale_fill_viridis(option = "H",
                     name = "Median METs",
                     guide = guide_colorbar(
                       title.position = "top",
                       direction = "vertical")) +
  labs(x='Time',
       y='Weekday',
       title = "Hourly Median METs Distribution per Weekday")
```

- Calories, intensity, steps and METs are all at their peak as list below
    1. Tuesday 18:00
    2. Wednesday 18:00
    3. Thursday 17:00
    4. Saturday 13:00
- In the late afternoon 16:00 to 19:00 in the evening, all the value on this heat map are in the orange zone, except on Saturday the orange zone starts from 11:00.
- Only in calories heat map we can find the orange zone cluster from 10:00 to 13:00 during the weekdays, especially on Tuesday.

Hourly Median Calories Heat Map per Weekday

By Activity Level

```{r heatmap2, fig.height = 8, fig.width = 8, fig.align = “center”} hourly_outer_join_1_ALPL %>% group_by(ActivityLevel,Weekday, Time) %>% summarize(medCal =
```

round(median(Calories)), .groups='drop') %>% ggplot(aes(Time, Weekday)) + scale_y_discrete(limits = rev) + theme(axis.text.x = element_text(angle = 90)) + geom_tile(aes(fill=medCal),colour = "white") + geom_text(aes(label = medCal), size = 2.5, color = "black") + scale_fill_viridis(option = "H", name = "Median Calories", guide = guide_colorbar(title.position = "top", direction = "vertical")) + labs(x='Time', y='Weekday', title = "Hourly Median Calories Distribution per Weekday") + facet_wrap(~ActivityLevel, ncol = 1)

```
<br />
<br />

* Surprisingly, people with medium activity level burn less calories during sleeping time than both people with high and low activity level.

<br />
<br />
#### Hourly Median Intensity Heat Map per Weekday
##### By Activity Level

<br />



```{r heatmap3, fig.height = 8, fig.width = 8, fig.align = "center"}
hourly_outer_join_1_ALPL %>%
  group_by(ActivityLevel,Weekday, Time) %>%
  summarize(medInt = round(median(TotalIntensity)),
            .groups='drop') %>%
  ggplot(aes(Time, Weekday)) +
  scale_y_discrete(limits = rev) +
  theme(axis.text.x = element_text(angle = 90)) +
  geom_tile(aes(fill=medInt),colour = "white")  +
  geom_text(aes(label = medInt),
            size = 2.5, color = "black") +
  scale_fill_viridis(option = "H",
                     name = "Median Intensity",
                     guide = guide_colorbar(
                       title.position = "top",
                       direction = "vertical")) +
  labs(x='Time',
       y='Weekday',
       title = "Hourly Median Intensity Distribution per Weekday") +
  facet_wrap(~ActivityLevel, ncol = 1)
```

- Again, we can see more orange zone on the heat map of medium activity level, especially on Saturday.
```

Hourly Median Steps Heat Map per Weekday

By Activity Level

```{r heatmap4, fig.height = 8, fig.width = 8, fig.align = “center”} hourly_outer_join_1_ALPL %>% group_by(ActivityLevel,Weekday, Time) %>% summarize(medStep = round(median(StepTotal)), .groups=‘drop’) %>% ggplot(aes(Time, Weekday)) + scale_y_discrete(limits = rev) + theme(axis.text.x = element_text(angle = 90)) + geom_tile(aes(fill=medStep),colour = “white”) + geom_text(aes(label = medStep), size = 2.5, color = “black”) + scale_fill_viridis(option = “H”, name = “Median Steps”, guide = guide_colorbar( title.position = “top”, direction = “vertical”)) + labs(x=‘Time’, y=‘Weekday’, title = “Hourly Median Steps Distribution per Weekday”) + facet_wrap(~ActivityLevel, ncol = 1)

```


* For high activity level, Saturday 13:00 and Wednesday 18:00 are the obviously peak points.
* For the medium activity level, the peak is on Saturday 11:00

Hourly Median METs Heat Map per Weekday
By Activity Level


```{r heatmap5, fig.height = 8, fig.width = 8, fig.align = "center"}
hourly_outer_join_1_ALPL %>%
  group_by(ActivityLevel,Weekday, Time) %>%
  summarize(medMETs = round(median(METs), digits = 2),
            .groups='drop') %>%
  ggplot(aes(Time, Weekday)) +
  scale_y_discrete(limits = rev) +
  theme(axis.text.x = element_text(angle = 90)) +
  geom_tile(aes(fill=medMETs),colour = "white")  +
  geom_text(aes(label = medMETs),
            size = 2.5, color = "black") +
  scale_fill_viridis(option = "H",
                     name = "Median METs",
                     guide = guide_colorbar(
```

```
                    title.position = "top",
                    direction = "vertical")) +
  labs(x='Time',
       y='Weekday',
       title = "Hourly Median METs Distribution per Weekday") +
  facet_wrap(~ActivityLevel, ncol = 1)
```

- Similar pattern as see on the previous heat map. For high activity level, the peak are at Saturday 13:00, Thursday 14:00 and Wednesday 18:00. Monday seems to have the least overall METs value. And for medium activity level, the peak is at Saturday 11:00 and overall METs value is lowest on Thursday.

Bar Chart for Hourly Average Calories, Intensity, Steps and METs Distribution per Weekday

```{r, fig.height = 4, fig.width = 5, fig.align = "center"} s <- hourly_outer_join_1_ALPL %>% group_by(Weekday) %>% summarize(avgCal = mean(Calories), avgInt = mean(TotalIntensity), avgStep = mean(StepTotal), avgMETs = mean(METs), .groups='drop') %>% ggplot() + theme(plot.title=element_text(size=11), axis.title=element_text(size=9), axis.text=element_text(size=8))

s + geom_col(aes(Weekday, avgCal), fill="darkolivegreen1", width=0.6) + labs(x='Weekday', y='Average Calories', title = "Hourly Average Calories Distribution per Weekday")

s + geom_col(aes(Weekday, avgInt), fill="darkorange", width=0.6) + labs(x='Weekday', y='Average Intensity', title = "Hourly Average Intensity Distribution per Weekday")

s + geom_col(aes(Weekday, avgStep), fill="cadetblue1", width=0.6) + labs(x='Weekday', y='Average Steps', title = "Hourly Average Steps Distribution per Weekday")

s + geom_col(aes(Weekday, avgMETs), fill="mediumpurple1", width=0.6) + labs(x='Weekday', y='Average METs', title = "Hourly Average METss Distribution per Weekday")

<br />
<br />

* Intensity value in this chart is calculated by adding all the minute-level intensity values that occurred within the hour. Not the same value on dailyIntensities_merged table.
* The data from all these bar charts are all hourly-based. The values are in hourly scale. Later in the project there will be another bar chart from daily-based table for comparison on pattern.
* We use average value calculated by mean function instead of median
```

value because this bar chart is weekday-based, not on hour. And also because all the value on another bar chart from daily-based table are average value, not median, so we'd better use value calculate from the same method.
* All the values from these chart are related, we can see similar pattern. The bar is highest on Saturday, followed by Tuesday. Sunday is the lowest for every value.

The next bar chart will display the value of each activity level side by side.

Bar Chart for Hourly Average Calories, Intensity, Steps and METs Distribution per Weekday
By Activity Level


```{r, fig.height = 4, fig.width = 5, fig.align = "center" }
G <- hourly_outer_join_1_ALPL %>%
  group_by(ActivityLevel,Weekday) %>%
  summarize(avgCal = mean(Calories),
            avgInt = mean(TotalIntensity),
            avgStep = mean(StepTotal),
            avgMETs = mean(METs),
            .groups='drop') %>%
  ggplot() +
  scale_fill_manual(values=
                      c("gold1",
                        "red",
                        "grey0")) +
  theme(plot.title=element_text(size=11),
        axis.title=element_text(size=9),
        axis.text=element_text(size=8),
        axis.text.x = element_text(angle = 90,vjust=0.5))



  G + geom_col(aes(Weekday, avgCal,
                fill=ActivityLevel),
                position = "dodge") +
```

```
  labs(x='Weekday',
       y='Average Calories',
       title = "Hourly Average Calories Distribution per Weekday",
       subtitle = "By Activity Level")

G + geom_col(aes(Weekday, avgInt,
             fill=ActivityLevel),
             position = "dodge") +
  labs(x='Weekday',
       y='Average Intensity',
       title = "Hourly Average Intensity Distribution per Weekday",
       subtitle = "By Activity Level")

G + geom_col(aes(Weekday, avgStep,
             fill=ActivityLevel),
             position = "dodge") +
  labs(x='Weekday',
       y='Average Steps',
       title = "Hourly Average Steps Distribution per Weekday",
       subtitle = "By Activity Level")

G + geom_col(aes(Weekday, avgMETs,
             fill=ActivityLevel),
             position = "dodge") +
  labs(x='Weekday',
       y='Average METs',
       title = "Hourly Average METs Distribution per Weekday",
       subtitle = "By Activity Level")
```

- The pattern for each activity level will be different from the previous bar chart.
- High activity level bar is peak at Tuesday, not on Saturday as previous chart, followed by Thursday. The lowest bar is on Sunday.
- Medium activity level is peak on Saturday, followed by Tuesday and Monday. The lowest is on Wednesday. We can see the range from minimum to maximum height is widest in this group.
- Chart of low activity level is peak on Saturday and Sunday is the lowest.

The chart below is weekday chart with the daily-based values of calories, distance, steps and METs. All the data are taken from Activity_METs_daily_ALPL table. We can compare patterns of the same value from both daily-base data and hourly-base data

Bar Chart for daily Average Calories, Distance, Steps and METs Distribution per Weekday

```{r daily-based bar chart, fig.height = 4, fig.width = 5, fig.align = "center"} J <- Activity_METs_daily_ALPL %>% group_by(Weekday) %>% summarize(avgCal = mean(Calories), avgDis = mean(TotalDistance), avgStep = mean(TotalSteps), avgMETs = mean(METs), .groups='drop') %>% ggplot() + theme(plot.title=element_text(size=11), axis.title=element_text(size=9), axis.text=element_text(size=8))

J + geom_col(aes(Weekday, avgCal), fill="darkolivegreen1", width=0.6) + labs(x='Weekday', y='Average Calories', title = "Daily Average Calories Distribution per Weekday")

J + geom_col(aes(Weekday, avgDis), fill="red", width=0.6) + labs(x='Weekday', y='Average Distance', title = "Daily Average Distance Distribution per Weekday")

J + geom_col(aes(Weekday, avgStep), fill="cadetblue1", width=0.6) + labs(x='Weekday', y='Average Steps', title = "Daily Average Steps Distribution per Weekday")

J + geom_col(aes(Weekday, avgMETs), fill="mediumpurple1", width=0.6) + labs(x='Weekday', y='Average METs', title = "Daily Average METss Distribution per Weekday")

<br />
<br />

* This bar chart display the same values on daily scale except the distance chart. Activity_METs_daily_ALPL table doesn't have the same intensity value like in the hourly-based table.
* The pattern of calories, steps and METs values are similar to hourly based chart, only daily scale give much higher values for calories and steps

<br />
<br />

The next bar chart will display the value of each activity level side by side.

<br />
<br />

#### Bar Chart for daily Average Calories, Distance, Steps and METs Distribution per Weekday
##### By Activity Level

<br />
```

```{r, fig.height = 4, fig.width = 5, fig.align = "center"}
J1 <- Activity_METs_daily_ALPL %>%
  group_by(ActivityLevel,Weekday) %>%
  summarize(avgCal = mean(Calories),
            avgDis = mean(TotalDistance),
            avgStep = mean(TotalSteps),
            avgMETs = mean(METs),
            .groups='drop') %>%
  ggplot() +
  scale_fill_manual(values=
                      c("sandybrown",
                        "peru",
                        "darkkhaki")) +

  theme(plot.title=element_text(size=11),
        axis.title=element_text(size=9),
        axis.text=element_text(size=8),
        axis.text.x = element_text(angle = 90,vjust=0.5))



  J1 + geom_col(aes(Weekday, avgCal,
                fill=ActivityLevel),
                position = "dodge") +
    labs(x='Weekday',
         y='Average Calories',
         title = "Daily Average Calories Distribution per Weekday",
         subtitle = "By Activity Level")

  J1 + geom_col(aes(Weekday, avgDis,
                fill=ActivityLevel),
                position = "dodge") +
    labs(x='Weekday',
         y='Average Distance',
         title = "Daily Average Distance Distribution per Weekday",
         subtitle = "By Activity Level")

  J1 + geom_col(aes(Weekday, avgStep,
                fill=ActivityLevel),
                position = "dodge") +
    labs(x='Weekday',
         y='Average Steps',
         title = "Daily Average Steps Distribution per Weekday",
         subtitle = "By Activity Level")

  J1 + geom_col(aes(Weekday, avgMETs,
                fill=ActivityLevel),
                position = "dodge") +
    labs(x='Weekday',
```

```
        y='Average METs',
        title = "Daily Average METss Distribution per Weekday",
        subtitle = "By Activity Level")
```

- From step chart, only people with high activity level have average daily steps over recommended value of 10,000 steps.
- As expected, we can see very similar pattern between step chart and distance chart.
- Only on Saturday, people with medium activity level have average steps slightly higher than people with high activity level, but have average distance slightly lower.
- People with low activity level have average steps equal or lower than 5,000.

Bar Chart for Hourly Median Calories, Intensity, Steps, METs Distribution over Day

```{r hourly_barplot, fig.height = 3, fig.width = 10, fig.align = "center"}

v <- hourly_outer_join_1_ALPL %>% group_by(Time) %>% summarize(medCal = median(Calories), medInt = median(TotalIntensity), medStep = median(StepTotal), medMETs = median(METs), .groups='drop') %>% ggplot() + theme(plot.title=element_text(size=11), axis.title=element_text(size=9), axis.text=element_text(size=8), axis.text.x = element_text(angle = 90,vjust=0.5))

v + geom_col(aes(Time, medCal), fill="burlywood4", width=0.6) + labs(x='Time', y='Median Calories', title = "Median Calories vs Time of Day")

v + geom_col(aes(Time, medInt), fill="royalblue4", width=0.6) + labs(x='Time', y='Median Intensity', title = "Median Intensity vs Time of Day")

v + geom_col(aes(Time, medStep), fill="red1", width=0.6) + labs(x='Time', y='Median Steps', title = "Median Steps vs Time of Day")

v + geom_col(aes(Time, medMETs), fill="olivedrab1", width=0.6) + labs(x='Time', y='Median METs', title = "Median METs vs Time of Day")

<br />
<br />

* We can also see similar pattern of all the values in these four chart. There are two peak point from 17:00 to 19:00, and also from 11:00 to 13:00
* Sleeping hours are from 23:00 to 11:00 of the next day, there is no activity and step, only there is still calorie burning during this time.
* Peak of all these chart is at 18:00, followed by 12:00 (only Calorie chart, the second peak is at 13:00).
```


Bar Chart for Hourly Median Calories, Intensity, Steps, METs Distribution over Day
By Activity Level


```{r hourly_barplot_activity_level, fig.height = 3, fig.width = 10, fig.align = "center"}
E <- hourly_outer_join_1_ALPL %>%
  group_by(ActivityLevel,Time) %>%
  summarize(medCal = median(Calories),
            medInt = median(TotalIntensity),
            medStep = median(StepTotal),
            medMETs = median(METs),
            .groups='drop') %>%
  ggplot() +
  scale_fill_manual(values=
                      c("red",
                        "green",
                        "blue4")) +
  theme(plot.title=element_text(size=11),
        axis.title=element_text(size=9),
        axis.text=element_text(size=8),
        axis.text.x = element_text(angle = 90,vjust=0.5))



  E + geom_col(aes(Time,
                   medCal,
                   fill=ActivityLevel),
                   position = "dodge") +
    labs(x="Time",
         y="Median Calories",
         title = "Median Calories vs Time of Day",
         subtitle = "By Activity Level")

  E + geom_col(aes(Time,
                   medInt,
                   fill=ActivityLevel),
                   position = "dodge") +
    labs(x="Time",
```

```
       y="Median Intensity",
       title = "Median Intensity vs Time of Day",
       subtitle = "By Activity Level")

E + geom_col(aes(Time,
                 medStep,
                 fill=ActivityLevel),
             position = "dodge") +
  labs(x="Time",
       y="Median Steps",
       title = "Median Steps vs Time of Day",
       subtitle = "By Activity Level")

E + geom_col(aes(Time,
                 medMETs,
                 fill=ActivityLevel),
             position = "dodge") +
  labs(x="Time",
       y="Median METs",
       title = "Median METs vs Time of Day",
       subtitle = "By Activity Level")
```

- The overall pattern is similar to the previous chart, but surprisingly on the intensity chart from 16:00 to 23:00, people with medium activity level have intensity value higher than people with high activity level.
- The contradiction could probably come from different metrics used to define intensity. Users are categorized into groups based on activity level from the data on daily-based chart, which define intensity as time spending on intense activity. But in this chart from hourly-based values, intensity is not measured as time.
- Another interesting point is calories burning during sleep time. From 22:00 to 6:00 in the morning of the next day, the red bars of people with low activity level are higher than the green bars of people with medium activity level. This indicates low activity people have higher calories burned during sleep time higher than medium activity people.
- From 19:00 in the evening to 23:00, people with medium activity level have number of steps higher than people with high activity level.
- From 20:00 to 22:00, people with medium activity level have highest METs value.
- For the chart that indicate activity like step chart and intensity chart, people in medium activity level seem to have less sleep as their activity start at 6:00 and end at 23:00 while people in other group start at 7:00 (high activity level) or 9:00 (low activity level) and finish at 22:00 for both high and low activity level.

Step Total and Total Intensity

As Related to Calorie Burn

```{r , warning=FALSE, fig.height = 8, fig.width = 8, fig.align = "center"}
hourly_outer_join_1_ALPL %>% ggplot(aes(StepTotal, TotalIntensity,
color=Calories )) + geom_point() + scale_color_viridis(option = "H") +
geom_smooth(method='loess', formula='y~x',color='deeppink') + labs( x='Step Total',
y='Total Intensity', title='Step Total vs. Total Intensity as Related to Calorie Burn' )
```


* The line is slightly curved in clockwise direction, which means intensity value will increase according to steps, but the rate will slightly decrease as number of steps increase over 10,000.
* It's interesting that on y axis where step value equal to zero, there two orange spot on the top with high intensity values. The color of that 2 spot are in orange zone which indicates high calories burns, at least for zero step. May be that could tell about possibility of high intensity activity or some exercise without taking any distance.

Total Intensity and Calories
As Related to Step Total


```{r , warning=FALSE, fig.height = 8, fig.width = 8, fig.align = "center"}
hourly_outer_join_1_ALPL %>%
  ggplot(aes(TotalIntensity, Calories,
             color=StepTotal
  )) +
  geom_point() +
  scale_color_viridis(option = "H") +
  geom_smooth(method='loess', formula='y~x',color='deeppink') +
  labs(
    x='Total Intensity',
    y='Calories',
    title='Total Intensity vs. Calories as Related to Step Total'
```

)

- Now the same set of value, we change all the axis. On the far right there are two dark blue dots that could be the same two orange dot of the previous scatter plot. With the intensity at the far right end and calories burned over 650, and taken zero step, this is one of very few oddity.
- The overall curve is slightly counter-clockwise, which mean the increasing rate of calories burned will slightly increase as intensity get higher.

Step Total and METs

As Related to Total Intensity

```{r , warning=FALSE, fig.height = 8, fig.width = 8, fig.align = "center"}
hourly_outer_join_1_ALPL %>% ggplot(aes(StepTotal, METs, color=TotalIntensity )) + geom_point() + scale_color_viridis(option = "H") + geom_smooth(method='loess', formula='y~x',color='deeppink') + labs( x='Step Total', y='METs', title='Step Total Vs. METs as Related to Total Intensity' )
```


* I change y axis from calories to METs, these two dot from previous two scatter plots show up again at zero step. The METs value of that 2 dot have METs values approximately around 7, the indication of vigorous-intensity activity.
* With only 33 users, we may be able to locale these 2 dots by using facet_grid function regarding activity level and participation level, then see check the Id that belong to the same groups from the previous pie chart (Intensity Distribution for Each Participant)
* The curve slightly change from clockwise to counter-clockwise in the middle of the curve, almost linear relationship. We probably can develop linear regression model to predict METs value as related to step and intensity activity for users.

METs and Calories
As Related to Total Intensity

```
<br />

```{r , warning=FALSE, fig.height = 8, fig.width = 8, fig.align =
"center"}
hourly_outer_join_1_ALPL %>%
  ggplot(aes(METs, Calories,
             color=TotalIntensity
  )) +
  geom_point() +
  scale_color_viridis(option = "H") +
  geom_smooth(method='loess', formula='y~x',color='deeppink') +
  labs(
    x='METs',
    y='Calories',
    title='METs Vs. Calories as Related to Total Intensity'
  )
```

- As expected, the relationship between colories burned and METs are nearly linear, including colors of intensity value are compatible to this relationship as well.

Step Total and Calories

As Related to METs

```
{r , warning=FALSE, fig.height = 8, fig.width = 8, fig.align =
"center"} hourly_outer_join_1_ALPL %>%    ggplot(aes(StepTotal,
Calories,              color=METs   )) +   geom_point() +
scale_color_viridis(option = "H") +   geom_smooth(method='loess',
formula='y~x',color='deeppink') +   labs(     x='Step Total',
y='Calories',      title='Step Total Vs. Calories as Related to METs'
)
```

- Positive correlation as expected, curve slightly change from clockwise to counter-clockwise in the middle with maximum steps a little bit above recommended daily steps. Very impressive for hourly scale.

Multiple Linear Regression Model for Hourly-Based Data Frame
```

Regression Model describes the correlation between a dependent variable, y, and one or more independent variables, X. From all the values in hourly-based data frame we have, we can see from our scatter plots that many values have almost linear relationship to one another. Linear regression model created from all these data can forecast the outcome from the data based on behavioral factors.

The functional form of Multiple Linear Regression is :

$$y=\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_n x_n$$

We use lm() to find all the coefficients $\beta_0, \beta_1, \beta_2, \ldots, \beta_n$ for all the independent variables $x_1, x_2, \ldots, x_n$ we have.

Linear Regression Model between METs and Intensity

```{r } options(scipen = 999)

lmMETs_Intensity <- lm(METs ~ TotalIntensity, data=hourly_outer_join_1_ALPL)

summary(lmMETs_Intensity)

<br />
<br />

From the regression analysis we get from summary function, the significance level of intensity(TotalIntensity) indicates this variable contributes to the explanation of the dependent variable(METs) significantly (three asterisks). R-squared is 0.9522 and very low p-value which is quite good.

<br />
<br />

##### Linear Regression Model between METs and Steps

<br />
<br />

```{r }

```
options(scipen = 999)

lmMETs_Step <-
  lm(METs ~
       StepTotal,
       data=hourly_outer_join_1_ALPL)

summary(lmMETs_Step)
```

In this model, Step also contribute significantly to explanation of METs, with very low p-value and R-squared is 0.8187.

Multiple Linear Regression Model between METs, Steps and Intensity

```{r } options(scipen = 999)

lmMETs_StepandIntensity <- lm(METs ~ StepTotal + TotalIntensity, data=hourly_outer_join_1_ALPL)

summary(lmMETs_StepandIntensity)
```


In this model, we incorporate both step and intensity to explain the correlation of these two independent variables and METs. The result is even better than the previous two models. Very low p-value and better R-squared of 0.9569

* The model from this result is the following equation.

METs = 1.0114400 + 0.0001869(StepTotal) + 0.0330450(TotalIntensity)

Create regression diagnostics plot.

```
<br />
<br />

```{r}
plot(lmMETs_StepandIntensity)
```

- The line in Residuals vs Fitted chart is almost horizontal line without distinct patterns, an indication for a linear relationship.
- Most of the points in Normal Q-Q chart fall approximately along the reference line in the middle, but as it comes closer to toward the ends on both sides, these points gradually deviate from the dashed line.
- Most of the dots are cluster at the bottom-left of Scale-Location chart. Horizontal line with equally spread points is a good indication of homoscedasticity. But our case have a heteroscedasticity problem.
- From Residuals vs Leverage chart, we can see all the outliers that might influence the regression results when included or excluded from the analysis. As we can see from our previous box plot, data from hourly-based table contain so many extreme values.

Hourly Heart Rate Box Plot for Each User

- We can see clearer in heat map that people with low activity level tend to have higher hear rate than people with medium and high activity level.
- People with high activity level have lower heart rate during sleep time than people with both medium and low activity level. This could indicate the difference in quality of sleep.
- For people with high activity level, the peak of heart rate in this heat map is correlate to the peak time of the previous hourly calorie heat map, at 13:00 on Saturday and 18:00 on Wednesday. Only the peak at 6:00 on Thursday is not corresponding to the calories burned at that time.
- On Tuesday 18:00, heart rate is not significantly high in all activity level, in contrast to the high values of calories burned, total steps, intensity, and METs on hourly heat map.
- For people with high activity level, we can see the orange zone stand out at 6:00 from Monday to Friday in contrast to the color of surrounding hours. This could suggest the routine of morning exercise before going to work.
```

Sleeping Value Distribution

By User, Activity Level, Participation Level

```{r pie_chart, fig.height = 15, fig.width = 10, fig.align = "center"}
minuteSleep_merged_ALPL %>% ggplot(aes(x=1, fill=factor(value))) + geom_bar(position = "fill", width = 1) + cp + scale_fill_manual(values=c("cadetblue1", "goldenrod1", "firebrick1")) +

facet_wrap(~Users~ActivityLevel~ParticipationLevel, scales = "free") + geom_label_repel(data = df_2_ALPL, aes(label = per, y=pos), position = position_fill(vjust = 0.5), size = 3.0, show.legend = FALSE) + theme(aspect.ratio = 1) + guides(fill = guide_legend(title = "Value")) + theme_classic() + theme(axis.text.y = element_blank(), strip.background = element_rect(fill = "thistle1"), legend.position = "bottom") + labs(title="Sleeping Value Distribution", subtitle = "By User, Activity Level, Participation Level", y="Sleeping Value Percentage")


<br />
<br />

* Value indicating the sleep state.(1 = asleep, 2 = restless, 3 = awake)
* People with low activity level tend to have more of red area in this pie chart, which indicate the higher percentage of time they stayed awake while sleeping. This could suggest about the quality of sleep they have.
* There is some exception like user number 3, this person have very high percentage of awaking time while sleeping.

<br />
<br />

#### Box Plot for Activity Minutes per Weekday


<br />
<br />

Start from this chart, we will use data from daily-based data frames, like Activity_METs_daily_long_ALPL from this chart or Activity_METs_daily_ALPL for the next charts with additional data on sleeping. For the chart involved sleeping, we use data from SMAd_ALPL (which stand for Sleep, METs, Activity daily) instead.

<br />
```

```
The reason is because we have lost huge chunk of data (from 934 rows to 410 rows )  while incorporating sleep data into daily activity data, due to less number of participant (only 24 users) on SleepDay_merged table. We want all the charts created to be as inclusive as possible, So we would use data from SMAd table only for the charts involved sleeping.

<br />
<br />

```{r activity_METs_daily_long_boxplot, fig.height = 8, fig.width = 8, fig.align = "center"}
Activity_METs_daily_long_ALPL %>%
  mutate(Weekday = wday(ActivityDate,
                        label = T,
                        week_start = 1)) %>%
  ggplot() +
  geom_boxplot(aes(Weekday,Minutes,fill=ActivityMinutes)) +
  theme_classic()
```

- This chart we can see the comparison of each activity minutes stand side by side in each weekday.
- The ranges of very active minutes are wider than the ranges of fairly active minutes.
- Sedentary minutes has the widest range and has the lowest median value on Thursday

The next chart we can see distribution of each activity minutes on weekday in closer look.

Box Plot for Very Active Minutes, Fairy Active Minutes, Lightly Active Minutes and Sedetary Minutes per Weekday

```{r active_minutes_weekday_boxplot, fig.height = 6, fig.width = 6, fig.align = “center”}

MM <- Activity_METs_daily_ALPL %>% mutate(Weekday = wday(ActivityDate, label = T, week_start = 1)) %>% ggplot() + scale_fill_manual(values= c(“gold1”, “lightpink”, “olivedrab2”, “darkorange”, “cadetblue1”, “mediumorchid2”, “firebrick2”)) + theme_classic()+ theme(axis.text.x = element_text(angle = 90))

MM + geom_boxplot(aes(Weekday, VeryActiveMinutes, fill=Weekday)) + labs(x=‘Weekday’, y=‘Very Active Minutes’, title = “Very Active Minutes Distribution per Weekday”)
```
```

MM + geom_boxplot(aes(Weekday, FairlyActiveMinutes, fill=Weekday)) + labs(x='Weekday', y='Fairy Active Minutes', title = "Fairy Active Minutes Distribution per Weekday")

MM + geom_boxplot(aes(Weekday, LightlyActiveMinutes, fill=Weekday)) + labs(x='Weekday', y='Lightly Active Minutes', title = "Lightly Active Minutes Distribution per Weekday")

MM + geom_boxplot(aes(Weekday, SedentaryMinutes, fill=Weekday)) + labs(x='Weekday', y='Sedentary Minutes', title = "Sedentary Minutes Distribution per Weekday")

* Sunday is the day that all the median values, except median value of sedentary minutes, are at the lowest. Sunday is like a rest day of the week according to the previous heat maps. All the value like calories burned, intensity, steps and METs are at the lowest.
* Median of very active minutes and fairly active minutes are highest on Tuesday. But as you can see, both charts have so many outliers.

We also can create box plot for each activity level to see the differences in activity minutes distribution per weekday.

Box Plot for Activity Minutes per Weekday
By Activity Level


```{r activity_METs_daily_long_boxplot_activity, fig.height = 12, fig.width = 8, fig.align = "center"}
Activity_METs_daily_long_ALPL %>%
  mutate(Weekday = wday(ActivityDate,
                        label = T,
                        week_start = 1)) %>%
  ggplot() +
  geom_boxplot(aes(Weekday,Minutes,fill=ActivityMinutes)) +
  theme_classic() +
  facet_wrap(~ActivityLevel, ncol = 1) +
  theme(strip.background = element_rect(fill = "aliceblue")) +
  labs(x='Minutes',
       y='Weekday',
```

```
  title = "Box Plot for Activity Minutes per Weekday",
  subtitle = "By Activity Level")
```

- People with high activity have higher value of very active minutes than fairy active minute, but the chart of people with medium activity level goes on the other way around.
- Sedentary minutes box plot of people with medium activity level mostly has median value lower than 1,000 minute. lower than the values from both people with high activity and low activity level.

Next chart is the activity minutes stacked bar for each individual user.

Weekday Average Activity Minutes Stacked Bar

By User, Activity Level

```{r bar_chart_weekday_activity_minutes, fig.height = 10, fig.width = 10, fig.align = "center"}
Activity_METs_daily_long_ALPL %>% mutate(Weekday = wday(ActivityDate, label = T, week_start = 1)) %>% ggplot(aes(Weekday,Minutes,fill=ActivityMinutes)) + geom_bar(width = 0.7,stat = "summary", fun = "mean") + theme_classic() + facet_wrap(~Users~ActivityLevel) + theme(axis.text.x = element_text(angle = 90,vjust=0.5), strip.background = element_rect(fill = "thistle1")) + labs(title="Weekday Average Activity Minutes Stacked Bar", subtitle="By User, Activity Level, Participation Level")
```

```
<br />
<br />

* For many users, the height of their stacked bars reach approximately near, or even reach, full 1,440 minutes in a day. Many of them have over 1,000 sedentary minutes in a day. Could that mean sedentary minutes include sleep time?

<br />

Next chart is weekday average activity minutes stacked bars categorized by activity level.

<br />
<br />

#### Weekday Average Activity Minutes Stacked Bar
```

````
##### Activity Level


<br />
<br />

```{r stackedbar_2, fig.height = 8, fig.width = 8, fig.align =
"center"}
Activity_METs_daily_long_ALPL %>%
  mutate(Weekday = wday(ActivityDate,
                        label = T,
                        week_start = 1)) %>%
  ggplot(aes(Weekday,Minutes,fill=ActivityMinutes)) +
  geom_bar(width = 0.7,stat = "summary", fun = "mean") +
  theme_classic()  +
  facet_wrap(~ActivityLevel) +
  theme(axis.text.x = element_text(angle = 90,vjust=0.5),
        strip.background = element_rect(fill = "thistle1")) +
  labs(title="Weekday Average Activity Minutes Stacked Bar",
       subtitle="By User, Activity Level, Participation Level")
````

- High activity level users have highest average value of very active minutes, with the peak on Tuesday.
- Users with low activity level have the shortest of both very active minutes and fairly active minutes. On the other hand, this group have the value of sedentary minutes the most in every weekday.
- Users with medium activity level have the most of fairly active minutes proportion.

Box Plot for Total Steps, Calories, METs and Total Distance per Weekday

```{r daily_boxplot, fig.height = 6, fig.width = 6, fig.align = “center”} AM <- Activity_METs_daily %>% mutate(Weekday = wday(ActivityDate, label = T, week_start = 1)) %>% ggplot() + scale_fill_manual(values= c(“gold1”, “lightpink”, “olivedrab2”, “darkorange”, “cadetblue1”, “mediumorchid2”, “firebrick2”)) + theme_classic()+ theme(axis.text.x = element_text(angle = 90))

AM + geom_boxplot(aes(Weekday, TotalSteps, fill=Weekday)) + labs(x=‘Weekday’, y=‘Total Steps’, title = “Total Steps Distribution per Weekday”)

AM + geom_boxplot(aes(Weekday, Calories, fill=Weekday)) + labs(x=‘Weekday’, y=‘Calories’, title = “Calories Distribution per Weekday”)

AM + geom_boxplot(aes(Weekday, METs, fill=Weekday)) + labs(x=‘Weekday’, y=‘METs’, title = “METs Distribution per Weekday”)
````
````

AM + geom_boxplot(aes(Weekday, TotalDistance, fill=Weekday)) + labs(x='Weekday', y='Total Distance', title = "Total Distance Distribution per Weekday")

* The charts above, we have calories, steps and METs box plots we can compare to the previous heat map we created from hour-based table. We can see the median line in the box plots of three chart have the highest values on Tuesday, instead of Saturday on hourly-based heat maps that we can see more orange zone. Keep in mind that there are always differences on daily and hourly scale, and also we have to take hours from non-orange zone on heat map into account, like sleeping hours that color in heat maps are in deep purple.
* The median value of distance is also highest on Tuesday.
* Sunday is the day that all median values are at the lowest.
* There are some outliers, but not as many as the data from hourly-based table.

Correlation Matrix for Daily-Based Data Fram with Sleeping Data

Now we use SMAd_ALPL table which stands for Sleep, METs and Activity daily in order to see how sleeping pattern can have some effect on other activities during waking hours, or some biological processes that happen throughout the day like calories burned and METs.

But before we creating any chart, let's use correlation matrix to see how all the data related to each other


```{r corr_chart, , message=FALSE, warning=FALSE, fig.height = 12, fig.width = 12, fig.align = "center"}
ggcorr(SMAd_ALPL[, 2:21],
       label = TRUE,
       hjust = 1.0,
       angle = -45,
       layout.exp = 1)
```

- The matrix doesn't indicate correlation between total time in bed, total minutes asleep and total sleep records with METs and calories burned, but sleep efficiency has some correlation.

Let's create multiple linear regression for this data frame and see how we can develop simple equation for all these relationship

Multiple Linear Regression Model for Sleep and METs Incorporated Daily-Based Data Frame

```
options(scipen = 999)

modSMAd_ALPL <- lm(Calories ~
              TotalSteps +
              VeryActiveMinutes +
              FairlyActiveMinutes +
              LightlyActiveMinutes +
              SedentaryMinutes+
              VeryActiveDistance +
              ModeratelyActiveDistance +
              LightActiveDistance +
              SedentaryActiveDistance +
              TotalSleepRecords +
              TotalMinutesAsleep +
              TotalTimeInBed +
              Sleep_Efficiency,
            data=SMAd_ALPL)

summary(modSMAd_ALPL)
```

- Adjusted R-Squared and p-value is good but we need to eliminate some independent variables that don't significantly contribute to explaining the variance in the dependent variable.
- In this case, these variables would be moderately active distance, sedentary active distance, total sleep records, total minutes asleep and total time in bed.

So the adjusted model would be as following.

```
options(scipen = 999)

modSMAd_ALPL_redux <- lm(Calories ~
                      TotalSteps +
                      VeryActiveMinutes +
                      FairlyActiveMinutes +
                      LightlyActiveMinutes +
                      SedentaryMinutes+
                      VeryActiveDistance +
                      LightActiveDistance +
                      Sleep_Efficiency,
                    data=SMAd_ALPL)


summary(modSMAd_ALPL_redux)
```

- With this model, we have every independent variable that contributes significantly to explain the variance in the dependent variable.
- Adjusted R-squared may be a little bit less than the previous model, but p-value is still good.

- The model from this result is the following equation.

Calories = -600.69244 - 0.22911(TotalSteps) + 13.96435(VeryActiveMinutes) + 16.58093(FairlyActiveMinutes) - 1.31264(LightlyActiveMinutes) + 0.81123(SedentaryMinutes) + 227.76369(VeryActiveDistance) + 544.11503(LightActiveDistance) + 17.54828(Sleep_Efficiency)

Let's check the diagnostic plot to see if this model has a good fit for normal distribution or not.

```
plot(modSMAd_ALPL_redux)
```

- Compare to the previous model created from hourly-based table, this model may not have value of Adjusted R-squared as high as previous model, but the diagnosis plot gives better result.
- The line in Residuals vs Fitted chart is well horizontal with the points scatter evenly above and below that line.

- Scale-Location plot of this model is even better than the previous model, with almost horizontal line at the height close to value of 1.
- All the point in Normal Q-Q plot follow along the straight line nicely, with very few extreme observation.
- There is an observation 289 on Residual vs Leverage plot that locates outside “Cook’s distance” dashed.

If sleep sleep efficiency is the only sleep related data that significantly contributes to calories burned. Let’s take a look at overall sleep distribution per weekday.

Box Plots for Total Minutes Asleep, Total Time in Bed and Sleep Efficiency Distribution per Weekday

```{r overall_sleep_boxplot, fig.height = 6, fig.width = 6, fig.align = “center”, warning=FALSE}
SL <- SMAd_ALPL %>% ggplot() + scale_fill_manual(values= c(“gold1”, “lightpink”, “olivedrab2”, “darkorange”, “cadetblue1”, “mediumorchid2”, “firebrick2”)) + theme_classic()+ theme(axis.text.x = element_text(angle = 90))

SL + geom_boxplot(aes(Weekday, TotalMinutesAsleep, fill=Weekday)) + labs(x=‘Weekday’, y=‘Total Minutes Asleep’, title = “Total Minutes Asleep Distribution per Weekday”)

SL + geom_boxplot(aes(Weekday, TotalTimeInBed, fill=Weekday)) + labs(x=‘Weekday’, y=‘Total Time In Bed’, title = “Total Time In Bed Distribution per Weekday”)

SL + geom_boxplot(aes(Weekday, Sleep_Efficiency, fill=Weekday)) + labs(x=‘Weekday’, y=‘Sleep Efficiency’, title = “Sleep Efficiency Distribution per Weekday”)


<br />
<br />

* Median value of total minutes asleep is at the highest on Sunday, and lowest on Friday.
* The median value of sleep efficiency also at the lowest on Sunday as well.
* During the weekday, Wednesday has the highest median value of both total minutes asleep and total time in bed. But for sleep efficiency, Monday has the highest median value during weekday.

<br />
<br />

#### Sleep Efficiency Distribution per Weekday
##### By Activity Level
```

```
<br />
<br />

```{r sleep_efficiency_boxplot, fig.height = 10, fig.width = 8,
fig.align = "center"}
SMAd_ALPL %>%
  ggplot() +
  scale_fill_manual(values=
                      c("gold1",
                        "lightpink",
                        "olivedrab2",
                        "darkorange",
                        "cadetblue1",
                        "mediumorchid2",
                        "firebrick2")) +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 90)) +
  geom_boxplot(aes(Weekday,
                        Sleep_Efficiency,
                        fill=Weekday)) +
  facet_wrap(~ActivityLevel, ncol = 1) +
  theme(strip.background = element_rect(fill = "aliceblue")) +
  labs(x='Weekday',
         y='Sleep Efficiency',
         title = "Sleep Efficiency Distribution per Weekday",
         subtitle = "By Activity Level")
```

- We can not see much difference in median value of sleep efficiency in each group, only box plot of people with medium activity level has widest range.
- There are some outliers indicate sleep deprivation in every group. Most of them are in medium activity level, follow by low activity level.

From previous activity minutes stacked bar charts, we notice the part of average sedentary minutes in some bars are over 1,200 minutes as compared to total 1,440 minutes in one day. The question is, does sedentary minutes include sleep time? Let's see the plot between total minutes asleep and sedentary minutes as related to sleep efficiency and sleep fragmentation (total sleep records).

Total Minutes Asleep Vs. Sedentary Minutes

As Related to Sleep Efficiency and Total Sleep Records
```

```{r, warning=FALSE, fig.height = 8, fig.width = 8, fig.align = "center"} SMAd_ALPL %>% ggplot(aes(TotalMinutesAsleep, SedentaryMinutes, color=Sleep_Efficiency )) + geom_point(aes(shape=factor(TotalSleepRecords))) + scale_color_viridis(option = "H") + geom_smooth(method='loess', formula='y~x',color='deeppink') + labs( x='Total Minutes Asleep', y='Sedentary Minutes', title='Total Minutes Asleep Vs. Sedentary Minutes Related to Sleep Efficiency and Total Sleep Records')

<br />
<br />

* As total minutes asleep increase, sedentary minutes decline.
* At the top-left of the chart, we can see the highest sedentary minutes over 1,200 minutes, but total minute asleep less than an hour.

<br />

Let's find out more about that observation. This dot has the maximum value of sedentary minutes.

<br />

```{r}
SMAd_ALPL %>%
  filter(SedentaryMinutes ==
           max(SMAd_ALPL[["SedentaryMinutes"]]))
```

This is user number 15, with daily usage and low activity level. This person spent only 65 minutes in bed and has only 59 minute asleep. This could suggest total time in bed may not be the same as, or not included with, sedentary minutes.

Check if there is any overlap between total minutes asleep and sedentary minutes

```{r}
SMAd_ALPL %>%
  filter(SedentaryMinutes + TotalMinutesAsleep > 1440)
```

- There are only three case that the sum of total minutes asleep and sedentary minutes has mount over 1,440 minutes of the whole day.

Let's take a look at each activity level.
```

Total Minutes Asleep Vs. Sedentary Minutes

As Related to Sleep Efficiency and Total Sleep Records

By Activity Level

```
{r SMAd_sedentary_sleep_facet, fig.height = 8, fig.width = 8,
fig.align = "center", warning=FALSE} SMAd_ALPL %>%
ggplot(aes(TotalMinutesAsleep, SedentaryMinutes,
color=Sleep_Efficiency   )) +
geom_point(aes(shape=factor(TotalSleepRecords))) +
scale_color_viridis(option = "H") +   geom_smooth(method='loess',
formula='y~x',color='deeppink') +   labs(     x='Total Minutes
Asleep',      y='Sedentary Minutes',      title='Total Minutes Asleep
Vs. Sedentary Minutes \nAs Related to Sleep Efficiency and Total Sleep
Records',     subtitle = 'By Activity Level') +
facet_wrap(~ActivityLevel)
```

- People with low activity level have the wide range of sedentary minutes, from 0 to 1,265 minutes, follow by people with medium activity level. People with high activity level have shortest range of sedentary minutes as well as total minutes asleep.

Now, let's take a look at the relationship between very active minutes and total minutes asleep to see how different on the other end.

Total Minutes Asleep Vs. Very Active Minutes

As Related to Sleep Efficiency and Total Sleep Records

```{r, warning=FALSE, fig.height = 8, fig.width = 8, fig.align = "center"} SMAd_ALPL %>% ggplot(aes(TotalMinutesAsleep, VeryActiveMinutes, color=Sleep_Efficiency )) + geom_point(aes(shape=factor(TotalSleepRecords))) + scale_color_viridis(option = "H") + geom_smooth(method='loess', formula='y~x',color='deeppink') + labs( x='Total Minutes Asleep', y='Very Active Minutes', title='Total Minutes Asleep Vs. Very Active Minutes Related to Sleep Efficiency and Total Sleep Records')

```



```
```

```
* The curve indicate no significant relationship between very active
minute and total minutes asleep.
* There are so many observation with very active minutes equal to
zero.
* Most observation with very active minute over an hour tend to
cluster around total minutes asleep from 300 to 600 minutes.

<br />

Now, let's break it down into activity level.

<br />
<br />

#### Total Minutes Asleep Vs. Very Active Minutes
#### As Related to Sleep Efficiency and Total Sleep Records
##### By Activity Level

<br />
<br />

```{r SMAd_veryactive_sleep_facet, fig.height = 8, fig.width = 8,
fig.align = "center", warning=FALSE}
SMAd_ALPL %>%
  ggplot(aes(TotalMinutesAsleep, VeryActiveMinutes,
             color=Sleep_Efficiency
  )) +
  geom_point(aes(shape=factor(TotalSleepRecords))) +
  scale_color_viridis(option = "H") +
  geom_smooth(method='loess', formula='y~x',color='deeppink') +
  labs(
    x='Total Minutes Asleep',
    y='Very Active Minutes',
    title='Total Minutes Asleep Vs. Very Active Minutes \nAs Related
to Sleep Efficiency and Total Sleep Records',
    subtitle = 'By Activity Level') +
  facet_wrap(~ActivityLevel)
```

- The curve of people with high activity level obviously display the peak in the middle of the curve, and it becomes less and less clear as we go down to medium and low activity level.

Next chart, we will see the relationship between calories burned and total minutes asleep as related to sleep effiency and total sleep records.
```

From our previous correlation matrix, total minutes asleep may not significantly contribute to the calories burned, specially in term of linear relationship, but let's see how it could give us some insight if we plot it as related to sleep efficiency and sleep fragmentation (TotalSleepRecords).

Total Minutes Asleep Vs. Calories

As Related to Sleep Efficiency and Total Sleep Records

```{r , warning=FALSE, fig.height = 8, fig.width = 8, fig.align = "center"}

SMAd_ALPL %>% ggplot(aes(TotalMinutesAsleep, Calories, color=Sleep_Efficiency )) + geom_point(aes(shape=factor(TotalSleepRecords))) +

scale_color_viridis(option = "H") + geom_smooth(method='loess', formula='y~x',color='green') + labs(x='Total Minutes Asleep', y='Calories', color = 'Sleep Efficiency', shape = 'Total Sleep Records', title= 'Total Minutes Asleep Vs. Calories', subtitle = 'As Related to Sleep Efficiency and Total Sleep Records' )


<br />
<br />

* The line is non-linear, but the blue or green dots which indicate sleep deprivation (sleep efficiency less than 80%) all cluster underneat the green line.
* All these blue and green dots have the values of daily calories burned less than 2,000.
* From this chart, sleep fragmentation doesn't seem to have any effect on calories burned. Many research on sleep fragmentation unanimously agreed it have effects on overall well being, but there's still no substantial model can explain how it should be incorporate into sleep efficiency.

<br />

The following chart is scatter plot between calories burned and total minute asleep, categorized by sleep fragmentation (TotalSleepRecords).

<br />
<br />

#### Total Minutes Asleep Vs. Calories
#### As Related to Sleep Efficiency and Total Sleep Records
##### By Total Sleep Records
```

```
<br />
<br />

```{r SMAd_by_sleep_records, fig.height = 8, fig.width = 8, fig.align
= "center", warning=FALSE}
SMAd_ALPL %>%
  ggplot(aes(TotalMinutesAsleep, Calories,
             color=Sleep_Efficiency
  )) +
  geom_point(aes(shape=factor(TotalSleepRecords))) +

  scale_color_viridis(option = "H") +
  geom_smooth(method='loess', formula='y~x',color='green') +
  labs(x='Total Minutes Asleep',
       y='Calories',
       color = 'Sleep Efficiency',
       shape = 'Total Sleep Records',
       title='Total Minutes Asleep Vs. Calories \nAs Related to Sleep
Efficiency and Total Sleep Records',
       subtitle = 'By Total Sleep Records') +
  facet_wrap(~TotalSleepRecords)
```

- From the distribution of dot in these three chart, sleep fragmentation effect on calories burned doesn't seem as clear as sleep efficiency. The rate of calories burned may depends more on other factors during non-sleeping hours. Further study about sleep effect on calories and other bio-chemical process still need to be done.

Next chart will be between total minutes asleep and calories burned, categorize by activity level.

Total Minutes Asleep Vs. Calories

As Related to Sleep Efficiency and Total Sleep Records

By Activity Level

```{r SMAd_by_ActivityLevel, fig.height = 8, fig.width = 8, fig.align = "center"} SMAd_ALPL %>% ggplot(aes(TotalMinutesAsleep, Calories, color=Sleep_Efficiency )) + geom_point(aes(shape=factor(TotalSleepRecords))) +

scale_color_viridis(option = "H") + geom_smooth(method='loess', formula='y~x',color='green') + labs(x='Total Minutes Asleep', y='Calories', color = 'Sleep Efficiency', shape = 'Total Sleep Records', title='Total Minutes Asleep Vs. Calories Related to

Sleep Efficiency and Total Sleep Records', subtitle = 'By Activity Level') + facet_wrap(~ActivityLevel)

<br />
<br />

* There seems to be not much different in calories burned related to activity level. The distribution of those dots seem to spread evenly from top to bottom, but the overall cluster of high activity level chart seem to be higher than those on med and low activity charts.
* It is interesting to see the dot that indicates the highest calories burned in the low activity level chart. But don't forget all the number are collected in daily scale.
* We can identify the user with highest calories burned by using filter function

<br />
<br />

```{r filter}
SMAd_ALPL %>%
  filter(Calories > 4800) %>%
  distinct(Users, ActivityLevel, ParticipationLevel)
```

- This user is number 22, in the group of low activity level and high usage

Next let's try to identify user with sleep deprivation using facet grid function.

Total Minutes Asleep Vs. Calories

As Related to Sleep Efficiency and Total Sleep Records

By Activity Level and Participation Level

```{r sleep_calories_scatter_grid, fig.height = 8, fig.width = 8, fig.align = "center"} SMAd_ALPL %>% ggplot(aes(TotalMinutesAsleep, Calories, color=Sleep_Efficiency )) + geom_point(aes(shape=factor(TotalSleepRecords))) +

scale_color_viridis(option = "H") + geom_smooth(method='loess', formula='y~x',color='green') + labs(x='Total Minutes Asleep', y='Calories', color = 'Sleep Efficiency', shape = 'Total Sleep Records', title='Total Minutes Asleep Vs. Calories Related to Sleep Efficiency and Total Sleep Records', subtitle = 'By Activity Level and Participation Level') + facet_grid(ActivityLevel~ParticipationLevel)
```

```
<br />
<br />

* People with high activity, sleep time tend to cluster around 250 to 600 minutes, the range is not as scattered as other groups.
* As we can see, all the blue dots that indicate sleep deprivation are in the group of medium activity level and high participation level (high usage).
* we can use the very first intensity distribution pie chart to identify the users with inadequate sleep. There are only two users who fall within both categories, user number 3 and user number 12. But there's so many users in the group of low activity level and high usage.
* To identify all users with sleep deprivation, it would be easier to use filter function.

<br />
<br />

```{r another filter}
SMAd_ALPL %>%
  filter(Sleep_Efficiency < 80) %>%
  distinct(Users, ActivityLevel, ParticipationLevel)
```

- Now we know they are user number 4 (low activity level, daily usage) and 12 (medium activity level, high usage).

The previous chart display the total calories burned on daily scale, it would be more complete to consider the rate of calories burned, like METs values, to get the whole picture.

Total Minutes Asleep Vs. METs

As Related to Sleep Efficiency and Total Sleep Records

```{r , warning=FALSE, fig.height = 8, fig.width = 8, fig.align = "center"} SMAd_ALPL %>% ggplot(aes(TotalMinutesAsleep, METs, color=Sleep_Efficiency )) + geom_point(aes(shape=factor(TotalSleepRecords))) +

scale_color_viridis(option = "H") + geom_smooth(method='loess', formula='y~x',color='green') + labs(x='Total Minutes Asleep', y='METs', color = 'Sleep Efficiency', shape = 'Total Sleep Records', title='Total Minutes Asleep Vs. METs Related to Sleep Efficiency and Total Sleep Records')
```
```

```
<br />
<br />

* In this chart, we can see some blue and green dots above the line,
with the highest METs value around 1.75.
* Both total minutes asleep and sleep efficiency do not have much
effect on METs value.

<br />

Let's break down this chart by activity and participation level.

<br />
<br />

#### Total Minutes Asleep Vs. METs
#### As Related to Sleep Efficiency and Total Sleep Records
##### By Activity Level, Participation Level

<br />
<br />

```{r sleep_METs_scatter_grid, fig.height = 8, fig.width = 8,
fig.align = "center"}
SMAd_ALPL %>%
  ggplot(aes(TotalMinutesAsleep, METs,
             color=Sleep_Efficiency
  )) +
  geom_point(aes(shape=factor(TotalSleepRecords))) +

  scale_color_viridis(option = "H") +
  geom_smooth(method='loess', formula='y~x',color='green') +
  labs(x='Total Minutes Asleep',
       y='METs',
       color = 'Sleep Efficiency',
       shape = 'Total Sleep Records',
       title='Total Minutes Asleep Vs. METs \nAs Related to Sleep
Efficiency and Total Sleep Records',
       subtitle ='By Activity Level, Participation Level') +
  facet_grid(ActivityLevel~ParticipationLevel)
```

- Although there are some similarity between this METs and previous calories plot, but calories value on the previous chart are total calories summed up for the whole day, not the rate of energy used as METs. So we can see both similarity and difference in pattern of pink lines from both chart.
```

- On the previous calories chart (by activity and participation level), if we take a look at the plot of low activity level and high usage, there is a dot on the top that indicates high calories burned of almost 5,000. But in this METs chart (by activity and participation level), at the chart of low activity level and high usage, all the dot cluster close together and there is no dot stand out with high METs value apart from other dots.

Let's see what's it like if we plot sleep efficiency on x-axis, and calories on y-axis with related to total steps

Sleep Efficiency Vs. Calories

As Related to Total Steps

```{r , warning=FALSE, fig.height = 8, fig.width = 8, fig.align = "center"} SMAd_ALPL %>% ggplot(aes(Sleep_Efficiency, Calories, color=TotalSteps )) + geom_point() + scale_color_viridis(option = "H") + geom_smooth(method='loess', formula='y~x',color='deeppink') + labs( x='Sleep Efficiency', y='Calories', title='Sleep Efficiency Vs. Calories Related to Total Steps')

<br />
<br />

* The curve slightly goes up as the value of sleep efficiency increase.

<br />

Next, let's make another chart between METs and sleep efficiency as related to total steps.

<br />
<br />

#### Sleep Efficiency Vs. METs
#### As Related to Total Steps

<br />
<br />

```{r , warning=FALSE, fig.height = 8, fig.width = 8, fig.align = "center"}

```
SMAd_ALPL %>%
  ggplot(aes(Sleep_Efficiency, METs,
             color=TotalSteps
  )) +
  geom_point() +
  scale_color_viridis(option = "H") +
  geom_smooth(method='loess', formula='y~x',color='deeppink') +
  labs(
    x='Sleep Efficiency',
    y='METs',
    title='Sleep Efficiency Vs. METs \nAs Related to Total Steps')
```

- The curve does not indicate significant relationship between sleep efficiency and METs values as related to total steps. But let's take a closer look at each activity level anyway.

Sleep Efficiency Vs. METs

As Related to Total Steps

By Activity Level

```
{r sleep_efficient_METs_wrap, fig.height = 8, fig.width = 8, fig.align
= "center"} SMAd_ALPL %>%    ggplot(aes(Sleep_Efficiency, METs,
color=TotalSteps   )) +   geom_point() +   scale_color_viridis(option
= "H") +   geom_smooth(method='loess', formula='y~x',color='deeppink')
+   labs(     x='Sleep Efficiency',      y='METs',      title='Sleep
Efficiency Vs. METs \nAs Related to Total Steps',     subtitle = 'By
Activity Level') +   facet_wrap(~ActivityLevel)
```

- All the people with high activity level have sleep efficiency over 80%, and also have the widest range of METs value.
- People with medium activity level have the widest range of sleep efficiency from about 58% to almost 100%.

Total Steps Vs. METs

As Related to Total Minutes Asleep

```{r , warning=FALSE, fig.height = 8, fig.width = 8, fig.align = "center"} SMAd_ALPL %>% ggplot(aes(TotalSteps, METs, color=TotalMinutesAsleep )) + geom_point() + scale_color_viridis(option = "H") + geom_smooth(method='loess', formula='y~x',color='deeppink') + labs( x='Total Steps', y='METs', title='Total Steps Vs. METs Related to Total Minutes Asleep')

<br />
<br />

* This plot indicates positive correlation between METs and total steps taken each day.
* The increasing rate of average METs value seems to decrease as number of steps increase.
* Total minutes asleep, unlike sleep efficiency, doesn't have any indication to have any effects on daily average METs values, as the color range from low to high number of minutes asleep on these dots just disperse evenly throughout the chart.

<br />

Let's break it down into each activity level and participation level.

<br />
<br />

#### Total Steps Vs. METs
#### As Related to Total Minutes Asleep
##### By Activity Level, Participation Level

<br />
<br />

```{r activity_METs_scatter_grid_2, warning=FALSE, fig.height = 8, fig.width = 8, fig.align = "center"}
SMAd_ALPL %>%
  ggplot(aes(TotalSteps, METs,
             color=TotalMinutesAsleep
  )) +
  geom_point() +
  scale_color_viridis(option = "H") +
  geom_smooth(method='loess', formula='y~x',color='deeppink') +
  labs(
    x='Total Steps',
    y='METs',
    title='Total Steps Vs. METs \nAs Related to Total Minutes Asleep',

    subtitle ='By Activity Level, Participation Level') +
    facet_grid(ActivityLevel~ParticipationLevel)
```
```

- The number of steps in low activity users range from 0 to 12,500 steps, with an exception of a dot on high usage chart.
- Users with high activity level have number of steps range from about 2,500 to 20,000 steps.
- We can see average steps taken daily in each activity level as following.

```
SMAd_ALPL %>%
  group_by(ActivityLevel) %>%
  summarize(avgSteps = mean(TotalSteps))
```

- Only users with high activity level have number of daily steps over 10,000 steps, recommended by CDC (Centers for Disease Control and Prevention).
- In the chart of high activity level and high usage, some observations taken less steps but have higher METs value than users in the group of high activity and daily usage. That may suggest about some other activities or exercises that can burned calories, instead of just walking or jogging.

Next chart, let's take a look at some data from weightLogInfo_merged_ALPL table.

Weight (Kg.) Vs. BMI

By Activity Level

```{r , warning=FALSE, fig.height = 8, fig.width = 8, fig.align = "center"}
weightLogInfo_merged_ALPL %>% ggplot(aes(WeightKg, BMI, color=ActivityLevel)) + geom_point() + facet_wrap(~ActivityLevel) + theme(axis.text.x = element_text(angle = 90,vjust=0.5)) + labs( x='Weight (Kg.)', y='BMI', title='Weight (Kg.) Vs. BMI', subtitle ='By Activity Level')
```

```
<br />
<br />

* Look like we found extreme case, low activity level, high weight and high BMI to the point of obesity. This is user number 5.
* User number 5 is one of the example of inconsistency we can find throughout this data set. Based on the dailyActivity_merged table, User 05 is daily user because we can find his data recorded from the beginning to the last day. But in some other table, for example
```

sleepDay_merged table, User 05 participated only 5 days.

We can only find out some more information about his daily activity on non-sleep related table to get some idea on the life style related to data on this weightLogInfo_merged_ALPL table.

The chart below using data from the table Activity_METs_daily_ALPL, which provide substantial amount of data for this user to get some idea about daily life style.

Total Steps Vs. Calories
As Related to Total Distance
User 05


```{r , warning=FALSE, fig.height = 8, fig.width = 8, fig.align = "center"}
Activity_METs_daily_ALPL %>%
  filter(Users =="User 05") %>%
  ggplot(aes(TotalSteps, Calories,
             color=TotalDistance
  )) +
  geom_point() +
  scale_color_viridis(option = "H") +
  geom_smooth(method='loess', formula='y~x',color='deeppink') +
  labs(
    x='Total Steps',
    y='Calories',
    title='Total Steps Vs. Calories \nAs Related to Total Distance',
    subtitle = 'User 05')
```

- Daily steps is less than 4,000 and calories burned range from about 2,000 to 2,600. To find out the healthy amount of calories burned, we still need more information such as sex, age, height, etc.

Data Limitations

1. The data set has some inconsistency and incomplete in many level. There are different number of user participation in many data frames. If we want to join two data frame together, let's take hourly based tables and heart rate table for instance, the differences in both user participation and also number of days or hours each user participated, make it inevitably to exclude significant amount of valuable data.
2. There are irregularity in time recording. Let's take heartrate_second_merged table for instance. The data in this table is supposed to recorded every 5 seconds, but beside the fact that not every user equally participated in collecting data, the data frame also skip on the period of time when there's no data collected. Instead of having timeline with constant recorded interval every 5 seconds with NA value on the absent data, the absent periods are just skipped with the beginning of next recorded period on the next row. This cause some problem in case we want to plot some line chart to see how heart rate swing in each day, without consistent interval with NA value during absent period, Line chart will automatically connect the line across the absent period.
3. The data set does not include very important data, such as gender, age, residential location, which is crucial factors that have effects on rate of metabolism, weather condition that suitable of outdoor activity, etc.

Recommendations

1. From correlation matrix and linear regression model we created, Bellabeat should develop app that can predict trend calories burned, weight change, etc., as related to recorded data from users that depicted in simple dashboard. This app display all the recorded and development trend to users, and encourage users to improve by sending notification about daily steps, sleep hours and the most important, caution or warning for some unusual data recorded to the app., like unusual heart rate.
2. Bellabeat app. should also develop custom made programs for each group of users according to their previous collected data. These programs encourage users to take their own steps to achieve their goal of improvement, in term of calories burned, weight loss, better sleeping quality, etc. All these program can also introduce users to more Bellabeat products that can help users achive their goal along the path.

Special Thanks

The author in the list below are the source of good ideas, and inspiration for this project. Please do yourself a favor, take a look at their beautiful works.

1.) https://www.kaggle.com/code/alixmcgettrick/bellabeat-case-study-in-r

2.) https://www.kaggle.com/code/zulkhaireesulaiman/bellabeat-capstone-project-in-r#-7.-Recommendations–

3.) https://www.kaggle.com/code/aymangouda/bellabeat-fitbit-capstone-project

Thank you for taking your time to have a look at my project!

Any comments and recommendations for improvement would be highly appreciated!