

title: Bellabeat

Analyse de données - Bellabeat

Projet Certification Google

Ce rapport présente les analyses effectuées à partir de données issues des activités quotidiennes, du sommeil, de la fréquence cardiaque, des calories brûlées et des mesures de poids d'utilisateurs de montres connectées, notamment via le suivi des pas, des activités intenses, et des cycles de sommeil.

Vu d'ensemble

Bellabeat est une modeste entreprise prospère, mais elle a le potentiel de devenir un acteur plus important sur le marché mondial des appareils intelligents. Urška Sršen, cofondatrice et directrice de la création de Bellabeat, estime que l'analyse des données de fitness des appareils intelligents pourrait aider à favoriser de nouvelles opportunités de croissance pour l'entreprise.

Objectif

Analyser les données d'utilisation des appareils intelligents afin d'obtenir un aperçu de la façon dont les consommateurs utilisent les appareils intelligents non Bellabeat afin d'adapter la gamme de Bellabeat à leurs besoins et à leur consommation

Données

Le jeu de données utilisé est Fit Bit Fitness Tracker Data (CC0: Domaine public) Ce jeu de données contient un tracker de fitness personnel de trente utilisateurs de fit bit qui ont consenti à la soumission de données de suivi personnel, y compris la sortie au niveau minute pour l'activité physique, la fréquence cardiaque et la surveillance du sommeil. Il comprend des informations sur l'activité quotidienne, les pas et la fréquence cardiaque qui peuvent être utilisés pour explorer les habitudes des utilisateurs. Inconvénient du jeu de données : L'utilisation d'un si petit jeu de données pour une analyse peut rendre l'analyse moins fiable.

Import des données

```
daily_activity <- read_csv("~/Downloads/Coursera/Projets_R/Bellabeat/sources/dailyActivity_merged.csv")
heartrates_seconds <- read_csv("~/Downloads/Coursera/Projets_R/Bellabeat/sources/heartrate_seconds_merged.csv")
hourly_calories <- read_csv("~/Downloads/Coursera/Projets_R/Bellabeat/sources/hourlyCalories_merged.csv")
hourly_intensities <- read_csv("~/Downloads/Coursera/Projets_R/Bellabeat/sources/hourlyIntensities_merged.csv")
hourly_steps <- read_csv("~/Downloads/Coursera/Projets_R/Bellabeat/sources/hourlySteps_merged.csv")
minute_calories <- read_csv("~/Downloads/Coursera/Projets_R/Bellabeat/sources/minuteCaloriesNarrow_merged.csv")
minute_intensities <- read_csv("~/Downloads/Coursera/Projets_R/Bellabeat/sources/minuteIntensitiesNarrow_merged.csv")
minute_met <- read_csv("~/Downloads/Coursera/Projets_R/Bellabeat/sources/minuteMETsNarrow_merged.csv")
minute_sleep <- read_csv("~/Downloads/Coursera/Projets_R/Bellabeat/sources/minuteSleep_merged.csv")
minute_steps <- read_csv("~/Downloads/Coursera/Projets_R/Bellabeat/sources/minuteStepsNarrow_merged.csv")
sleep_day <- read_csv("~/Downloads/Coursera/Projets_R/Bellabeat/sources/sleepDay_merged.csv")
weight_log_info <- read_csv("~/Downloads/Coursera/Projets_R/Bellabeat/sources/weightLogInfo_merged.csv")
```

Nettoyage et Préparation des Données

Les différentes sources de données ont été importées et nettoyées :

- Suppression des doublons
- Normalisation des noms de colonnes
- Filtrage des valeurs manquantes et invalides
- Agrégation par utilisateur sur plusieurs dimensions (pas moyens, minutes de sommeil, etc.)
- Vérification des types et formats de données

Suppression des doublons

```
daily_activity <- distinct(daily_activity)
heartrates_seconds <- distinct(heartrates_seconds)
hourly_calories <- distinct(hourly_calories)
hourly_intensities <- distinct(hourly_intensities)
hourly_steps <- distinct(hourly_steps)
minute_calories <- distinct(minute_calories)
minute_intensities <- distinct(minute_intensities)
minute_met <- distinct(minute_met)
minute_sleep <- distinct(minute_sleep)
minute_steps <- distinct(minute_steps)
sleep_day <- distinct(sleep_day)
weight_log_info <- distinct(weight_log_info)
```

Normalisation des noms de colonnes

```
colnames(daily_activity) <- c("id", "activity_date", "total_steps", "total_distance",
"tracker_distance", "logged_activities_distance", "very_active_distance", "moderately_active_distance", "light_active_distance", "sedentary_active_distance", "very_active_minutes", "fairly_active_minutes", "lightly_active_minutes", "sedentary_minutes", "calories")
colnames(heartrates_seconds) <- c("id", "time", "value")
colnames(hourly_calories) <- c("id", "activity_hour", "calories")
colnames(hourly_intensities) <- c("id", "activity_hour", "total_intensity", "average_intensity")
colnames(hourly_steps) <- c("id", "activity_hour", "step_total")
colnames(minute_calories) <- c("id", "activity_minute", "calories")
colnames(minute_intensities) <- c("id", "activity_minute", "intensity")
colnames(minute_met) <- c("id", "activity_minute", "mets")
colnames(minute_sleep) <- c("id", "date", "value", "log_id")
colnames(minute_steps) <- c("id", "activity_minute", "steps")
colnames(sleep_day) <- c("id", "sleep_day", "total_sleep_records", "total_minutes_asleep", "total_time_in_bed")
colnames(weight_log_info) <- c("id", "date", "weight_kg", "weight_pounds", "fat", "bmi", "is_manual_report", "log_id")
```

Agrégation par utilisateur sur plusieurs dimensions (pas moyens, minutes de sommeil, etc.)

```
hourly_data <- merge(hourly_calories, hourly_intensities, by = c("id", "activity_hour"), all.x = TRUE)
hourly_data <- merge(hourly_data, hourly_steps, by = c("id", "activity_hour"), all.x = TRUE)
rm(hourly_calories, hourly_intensities, hourly_steps)
minute_data <- merge(minute_calories, minute_intensities, by = c("id", "activity_minute"), all.x = TRUE)
minute_data <- merge(minute_data, minute_met, by = c("id", "activity_minute"), all.x = TRUE)
minute_data <- merge(minute_data, minute_steps, by = c("id", "activity_minute"), all.x = TRUE)
rm(minute_calories, minute_intensities, minute_met, minute_steps)
```

Filtrage des valeurs manquantes et invalides

```
daily_activity <- filter(daily_activity, total_distance != 0)
hourly_data <- filter(hourly_data, step_total != 0)
minute_data <- filter(minute_data, steps != 0)
```

Vérification des types et formats de données

```

heartrates_seconds$time <- strptime(heartrates_seconds$time, "%m/%d/%Y %I:%M:%S %p")
hourly_data$datetime <- strptime(hourly_data$activity_hour, "%m/%d/%Y %I:%M:%S %p")
hourly_data$date <- format(hourly_data$datetime, "%Y-%m-%d")
hourly_data$time <- format(hourly_data$datetime, "%H:%M:%S")
minute_data$datetime <- strptime(minute_data$activity_minute, "%m/%d/%Y %I:%M:%S %p")
minute_data$date <- format(minute_data$datetime, "%Y-%m-%d")
minute_data$time <- format(minute_data$datetime, "%H:%M:%S")
daily_activity$activity_date <- strptime(daily_activity$activity_date, "%m/%d/%Y")
minute_sleep$datetime <- strptime(minute_sleep$date, "%m/%d/%Y %I:%M:%S %p")
minute_sleep$date <- format(minute_sleep$datetime, "%Y-%m-%d")
minute_sleep$time <- format(minute_sleep$datetime, "%H:%M:%S")
sleep_day$datetime <- strptime(sleep_day$sleep_day, "%m/%d/%Y %I:%M:%S %p")
sleep_day$date <- format(sleep_day$datetime, "%Y-%m-%d")
sleep_day$time <- format(sleep_day$datetime, "%H:%M:%S")
weight_log_info$datetime <- strptime(weight_log_info$date, "%m/%d/%Y %I:%M:%S %p")
weight_log_info$date <- format(weight_log_info$datetime, "%Y-%m-%d")
weight_log_info$time <- format(weight_log_info$datetime, "%H:%M:%S")
hourly_data$activity_hour = NULL
minute_data$activity_minute = NULL
sleep_day$sleep_day = NULL

```

```

heartrates_seconds$hour <- lubridate::floor_date(heartrates_seconds$time, unit = "hour")

heartrates_hours <- heartrates_seconds %>%
  group_by(id, hour) %>%
  summarise_at(vars(value), list(heartrate = mean))

hourly_data <- merge(hourly_data, heartrates_hours, by.x=c("id", "datetime"), by.y=c(
  "id", "hour"), all.y= TRUE)
rm(heartrates_seconds, heartrates_hours)

```

Analyse

Synthèse des Utilisateurs

```

unique_data <- list()
unique_data[["nb_users_hourly"]] <- nrow(hourly_data %>% distinct(id))
unique_data[["nb_users_minute"]] <- nrow(minute_data %>% distinct(id))
unique_data[["nb_users_daily_activity"]] <- nrow(daily_activity %>% distinct(id))
unique_data[["nb_users_minute_sleep"]] <- nrow(minute_sleep %>% distinct(id))
unique_data[["nb_users_sleep_day"]] <- nrow(sleep_day %>% distinct(id))
unique_data[["nb_users_weight"]] <- nrow(weight_log_info %>% distinct(id))
unique_data <- data.frame(unique_data)
user_data <- daily_activity %>% distinct(id)

```

Grace à cette manipulation nous savons désormais que le jeu de données présente - 33 utilisateurs dont - 24 dorment avec la montre et - 8 seulement reportent leur poids dans leur application ou se pèsent grâce à une balance connectée.

Régularité d'utilisation

Les utilisateurs portent leur montre combien de jours en moyenne sur 30 jours ?

```
library(plyr)
daily_activity_nb <- daily_activity %>% count('id')
user_data <- merge(user_data, daily_activity_nb, by = "id", all.x = TRUE)
colnames(user_data)[colnames(user_data) == 'freq'] <- 'daily_activity_nb'
rm(daily_activity_nb)
```

Dépenses calorifique moyen par utilisateurs

```
daily_calories_m <- daily_activity %>%
  group_by(id) %>%
  summarise_at(vars(calories), list(daily_calories_m = mean))
daily_calories_m$daily_calories_m <- round(daily_calories_m$daily_calories_m, 0)
user_data <- merge(user_data, daily_calories_m, by = "id", all.x = TRUE)
rm(daily_calories_m)
```

Gestion du poids

Comment les utilisateurs traquent ils leurs poids ? A quelle fréquence ? De façon automatique ou non ?

```
weight_log_nb <- weight_log_info %>% count('id')
user_data <- merge(user_data, weight_log_nb, by = "id", all.x = TRUE)
colnames(user_data)[colnames(user_data) == 'freq'] <- 'weight_log_nb'
rm(weight_log_nb)

is_manual_report <- distinct(weight_log_info, id, is_manual_report)
user_data <- merge(user_data, is_manual_report, by = "id", all.x = TRUE)
rm(is_manual_report)
```

Habitudes de sommeil

Combien de minutes de sommeil moyennes par jour et par utilisateur

```
daily_sleep <- sleep_day %>%
  group_by(id) %>%
  summarise_at(vars(total_minutes_asleep), list(minute_sleep_m = ~ mean(.)))
user_data <- merge(user_data, daily_sleep, by = "id", all.x = TRUE)
rm(daily_sleep)
```

Ici est apparu une difficulté, en effet une nuit est à cheval sur deux jours. Quand commence le sommeil, est-ce une sieste. J'ai simplifié cela en prenant seulement les données de "minutes endormies".

A quelle heure se couchent les utilisateurs en moyenne chaque jour ?

```

user_bedtime <- minute_sleep %>%
  group_by(date, id) %>%
  filter(row_number()==1) %>%
  filter(time>"17:00:00")
user_bedtime$seconds <- period_to_seconds(lubridate::hms(user_bedtime$time))
user_bedtime_m <- user_bedtime %>%
  group_by(id) %>%
  summarise_at(vars(seconds), list(mean = mean))
user_bedtime_m$user_bedtime_m <- round(seconds_to_period(user_bedtime_m$mean),0)
user_bedtime_m$user_bedtime_m <- format(user_bedtime_m$user_bedtime_m, "%H:%M:%S")
user_bedtime_m$mean = NULL
user_data <- merge(user_data, user_bedtime_m, by = "id", all.x = TRUE)
rm(user_bedtime, user_bedtime_m)

```

Est ce que les utilisateurs ont l'habitude de faire des siestes (endormissement entre 8:00 et 17:00) ?

```

daily_sleep <- minute_sleep %>%
  group_by(log_id) %>%
  filter(row_number()==1) %>%
  filter(time > '08:00:00' & time < '17:00:00')
nap_nb <- daily_sleep %>%
  group_by(id) %>%
  tally()
user_data <- merge(user_data, nap_nb, by = "id", all.x = TRUE)
colnames(user_data)[colnames(user_data) == "n"] <- "nap_nb"
rm(daily_sleep, nap_nb)

```

Pouvoir afficher les heures de sommeil moyen.

```

detach("package:plyr", unload = TRUE)

hour_sleep <- minute_sleep %>%
  group_by(hour = lubridate::floor_date(datetime, "1 hour")) %>%
  summarise(count = n())

hour_sleep$date <- format(hour_sleep$hour, "%Y-%m-%d")
hour_sleep$time <- format(hour_sleep$hour, "%H:%M:%S")
hour_sleep$hour = NULL

```

Pas moyens

Combien les utilisateurs font de pas par jour en moyenne ?

```

daily_steps_m <- daily_activity %>%
  group_by(id) %>%
  summarise_at(vars(total_steps), list(daily_steps_m = mean))
daily_steps_m$daily_steps_m <- round(daily_steps_m$daily_steps_m, 0)
user_data <- merge(user_data, daily_steps_m, by = "id", all.x = TRUE)
rm(daily_steps_m)

```

Activité moyenne

```
daily_activity_m <- daily_activity %>%
  group_by(id) %>%
  summarise_at(c("very_active_minutes", "fairly_active_minutes", "lightly_active_minu
tes", "sedentary_minutes"), mean, na.rm = TRUE)
daily_activity_m <- mutate_if(daily_activity_m, is.numeric, round)
colnames(daily_activity_m)[colnames(daily_activity_m) == "very_active_minutes"] <- "v
ery_active_minutes_m"
colnames(daily_activity_m)[colnames(daily_activity_m) == "fairly_active_minutes"] <-
"fairly_active_minutes_m"
colnames(daily_activity_m)[colnames(daily_activity_m) == "lightly_active_minutes"] <-
"lightly_active_minutes_m"
colnames(daily_activity_m)[colnames(daily_activity_m) == "sedentary_minutes"] <- "sed
entary_minutes_m"
user_data <- merge(user_data, daily_activity_m, by = "id", all.x = TRUE)
rm(daily_activity_m)

user_data_m <- user_data %>%
  summarise_at(
    vars(minute_sleep_m, daily_steps_m, daily_calories_m, very_active_minutes_m, fair
ly_active_minutes_m, lightly_active_minutes_m, sedentary_minutes_m),
    list(g = ~ mean(., na.rm = TRUE))
  )
```

Activité en semaine et le week end

```
daily_activity$day <- weekdays(daily_activity$activity_date)
weekend_activity <- daily_activity %>%
  filter(day == "Saturday" | day == "Sunday")
weekend_activity_m <- weekend_activity %>%
  group_by(id) %>%
  summarise_at(vars(very_active_minutes, fairly_active_minutes, lightly_active_minut
es, sedentary_minutes), list(we_m = ~ round(mean(.))))
user_data <- merge(user_data, weekend_activity_m, by = "id", all.x = TRUE)
rm(weekend_activity, weekend_activity_m)

week_activity <- daily_activity %>%
  filter(day != "Sat" & day != "Sun")
week_activity_m <- week_activity %>%
  group_by(id) %>%
  summarise_at(vars(very_active_minutes, fairly_active_minutes, lightly_active_minut
es, sedentary_minutes), list(w_m = ~ round(mean(.))))
user_data <- merge(user_data, week_activity_m, by = "id", all.x = TRUE)
rm(week_activity, week_activity_m)
```

```
sleep_day$day <- weekdays(sleep_day$datetime)
```

```

weekend_activity <- daily_activity %>%
  filter(day == "Saturday" | day == "Sunday")
weekend_sleep <- sleep_day %>%
  filter(day == "Saturday" | day == "Sunday")
weekend_activity_m <- weekend_activity %>%
  summarise_at(vars(total_steps, calories, very_active_minutes, fairly_active_minut
s, lightly_active_minutes, sedentary_minutes), list(we_m = ~ round(mean(.))))
weekend_sleep_m <- weekend_sleep %>%
  summarise_at(vars(total_minutes_asleep), list(we_m = ~ round(mean(.))))
weekend_activity_m$sleep_we_m <- weekend_sleep_m$we_m

week_activity <- daily_activity %>%
  filter(day != "Saturday" | day != "Sunday")
week_sleep <- sleep_day %>%
  filter(day != "Saturday" | day != "Sunday")
week_activity_m <- week_activity %>%
  summarise_at(vars(total_steps, calories, very_active_minutes, fairly_active_minut
s, lightly_active_minutes, sedentary_minutes), list(w_m = ~ round(mean(.))))
week_sleep_m <- week_sleep %>%
  summarise_at(vars(total_minutes_asleep), list(we_m = ~ round(mean(.))))
week_activity_m$sleep_w_m <- week_sleep_m$we_m

max_min_u_d <- data.frame(
  minute_sleep_m_g = c(1440, 0), # Valeur max et min pour le sommeil
  daily_steps_m_g = c(15000, 0), # Valeur max et min pour les pas quotidiens
  daily_calories_m_g = c(4000, 0), # Valeur max et min pour les calories
  very_active_minutes_m_g = c(1440, 0), # Minutes très actives
  fairly_active_minutes_m_g = c(1440, 0), # Minutes moyennement actives
  lightly_active_minutes_m_g = c(1440, 0), # Minutes légèrement actives
  sedentary_minutes_m_g = c(1440, 0) # Minutes sédentaires (max = 24h)
)
max_min_w_a_m <- data.frame(
  sleep_w_m = c(1440, 0), # Valeur max et min pour le sommeil
  total_steps_w_m = c(15000, 0), # Valeur max et min pour les pas quotidiens
  calories_w_m = c(4000, 0), # Valeur max et min pour les calories
  very_active_minutes_w_m = c(1440, 0), # Minutes très actives
  fairly_active_minutes_w_m = c(1440, 0), # Minutes moyennement actives
  lightly_active_minutes_w_m = c(1440, 0), # Minutes légèrement actives
  sedentary_minutes_w_m = c(1440, 0) # Minutes sédentaires (max = 24h)
)
max_min_we_a_m <- data.frame(
  sleep_we_m = c(1440, 0), # Valeur max et min pour le sommeil
  total_steps_we_m = c(15000, 0), # Valeur max et min pour les pas quotidiens
  calories_we_m = c(4000, 0), # Valeur max et min pour les calories
  very_active_minutes_we_m = c(1440, 0), # Minutes très actives
  fairly_active_minutes_we_m = c(1440, 0), # Minutes moyennement actives
  lightly_active_minutes_we_m = c(1440, 0), # Minutes légèrement actives
  sedentary_minutes_we_m = c(1440, 0) # Minutes sédentaires (max = 24h)
)

user_data_m <- rbind(max_min_u_d, user_data_m)
week_activity_m <- rbind(max_min_w_a_m, week_activity_m)
weekend_activity_m <- rbind(max_min_we_a_m, weekend_activity_m)

rm(max_min_u_d, max_min_w_a_m, max_min_we_a_m)

```


Détermination d'un score d'activité

#####en multipliant le nombre de pas, les minutes intensives et les calories.

```
user_data <- mutate(user_data, activity_score = daily_steps_m * very_active_minutes_m  
* daily_calories_m)
```

Création des moyennes pour les 5 utilisateurs les plus actifs

et les 5 les moins actifs.

```

most_active <- user_data %>%
  arrange(activity_score, decreasing=TRUE) %>%
  filter(row_number()<=5)

most_active_m <- most_active %>%
  summarise_at(
    vars(minute_sleep_m, daily_steps_m, daily_calories_m, very_active_minutes_m, fair
ly_active_minutes_m, lightly_active_minutes_m, sedentary_minutes_m),
    list(g = ~ round(mean(., na.rm = TRUE))) # Exclure les NA lors du calcul de la m
oyenne
  )

less_active <- user_data %>%
  arrange(activity_score) %>%
  filter(activity_score>0) %>%
  filter(row_number()<=5)

less_active_m <- less_active %>%
  summarise_at(
    vars(minute_sleep_m, daily_steps_m, daily_calories_m, very_active_minutes_m, fair
ly_active_minutes_m, lightly_active_minutes_m, sedentary_minutes_m),
    list(g = ~ round(mean(., na.rm = TRUE))) # Exclure les NA lors du calcul de la m
oyenne
  )

max_min_m_a_m <- data.frame(
  minute_sleep_m_g = c(1440, 0), # Valeur max et min pour le sommeil
  daily_steps_m_g = c(15000, 0), # Valeur max et min pour les pas quotidiens
  daily_calories_m_g = c(4000, 0), # Valeur max et min pour les calories
  very_active_minutes_m_g = c(1440, 0), # Minutes très actives
  fairly_active_minutes_m_g = c(1440, 0), # Minutes moyennement actives
  lightly_active_minutes_m_g = c(1440, 0), # Minutes légèrement actives
  sedentary_minutes_m_g = c(1440, 0) # Minutes sédentaires (max = 24h)
)

max_min_l_a_m <- data.frame(
  minute_sleep_m_g = c(1440, 0), # Valeur max et min pour le sommeil
  daily_steps_m_g = c(15000, 0), # Valeur max et min pour les pas quotidiens
  daily_calories_m_g = c(4000, 0), # Valeur max et min pour les calories
  very_active_minutes_m_g = c(1440, 0), # Minutes très actives
  fairly_active_minutes_m_g = c(1440, 0), # Minutes moyennement actives
  lightly_active_minutes_m_g = c(1440, 0), # Minutes légèrement actives
  sedentary_minutes_m_g = c(1440, 0) # Minutes sédentaires (max = 24h)
)

most_active_m <- rbind(max_min_m_a_m, most_active_m)
less_active_m <- rbind(max_min_l_a_m, less_active_m)

rm(max_min_m_a_m, max_min_l_a_m)

```

Mise en longueur du tableau d'activité

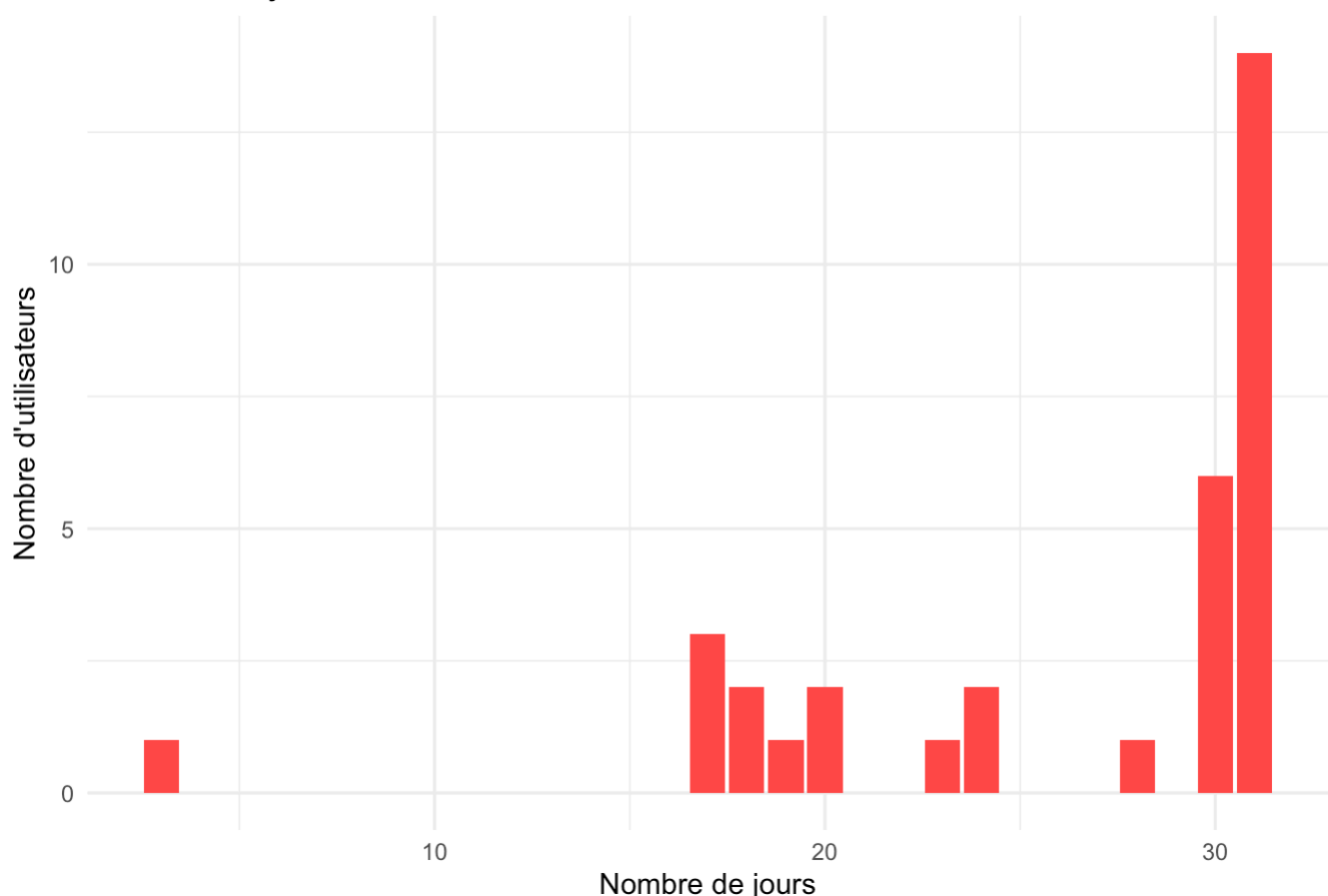
```
daily_activity_long <- daily_activity %>%
  select(very_active_minutes, fairly_active_minutes, lightly_active_minutes, sedentary_minutes, day) %>%
  pivot_longer(cols = c(very_active_minutes, fairly_active_minutes, lightly_active_minutes, sedentary_minutes),
               names_to = "activity_type", values_to = "minutes") %>%
  group_by(activity_type, day) %>% # Group by after pivoting
  summarise(m = round(mean(minutes, na.rm = TRUE)), .groups = 'drop')
```

Visualisation

Régularité d'utilisation

```
ggplot(user_data, aes(daily_activity_nb)) +
  geom_bar(fill="#ff4949") +
  theme_minimal() +
  labs(title = "Nombre de jours d'utilisation de la montre",
       x = "Nombre de jours",
       y = "Nombre d'utilisateurs")
```

Nombre de jours d'utilisation de la montre



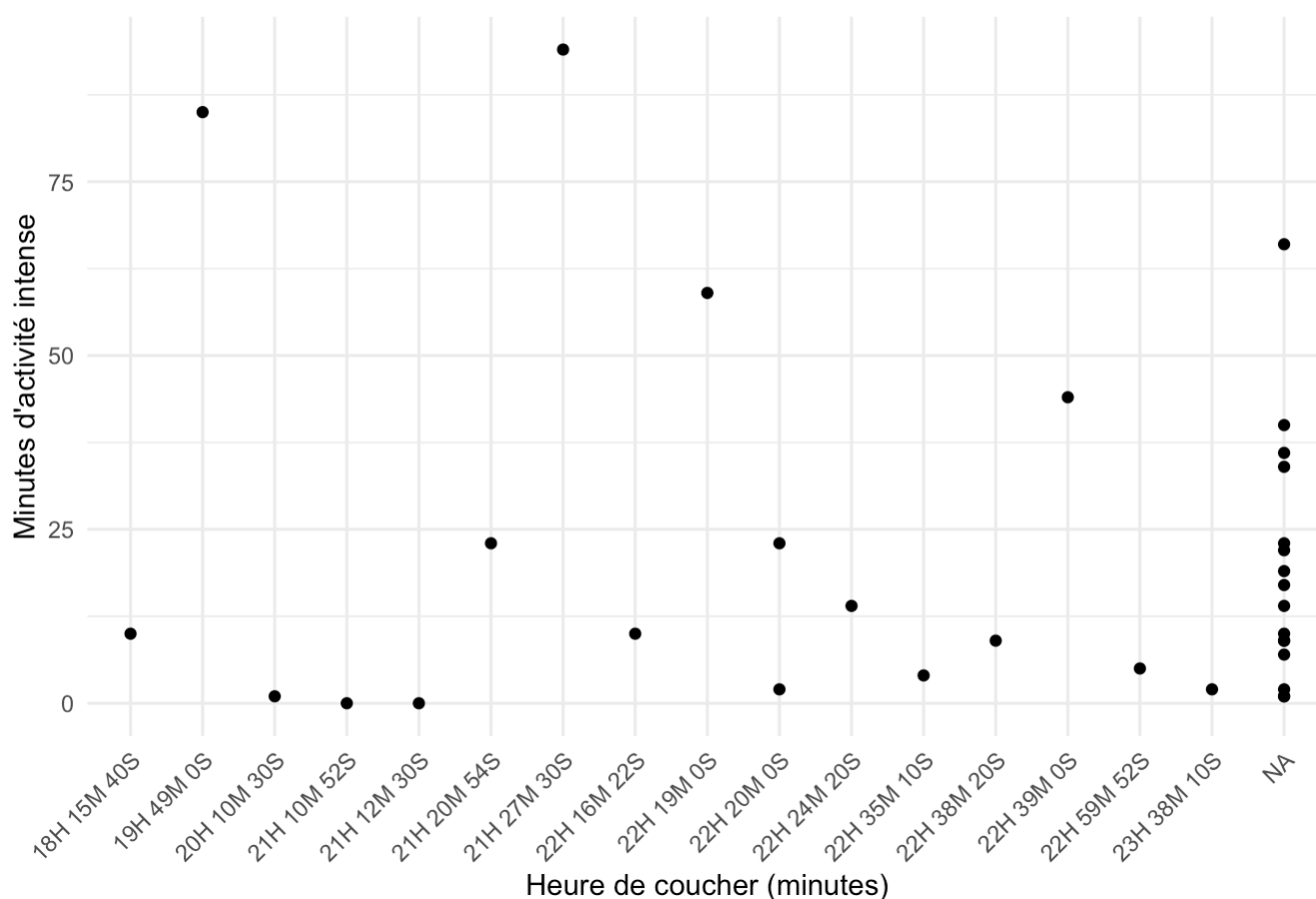
La moitié des utilisateurs portent quotidiennement leur montre. La quasi totalité la porte au moins un jour sur deux.

Heure de coucher en fonction du temps d'activité intense ou

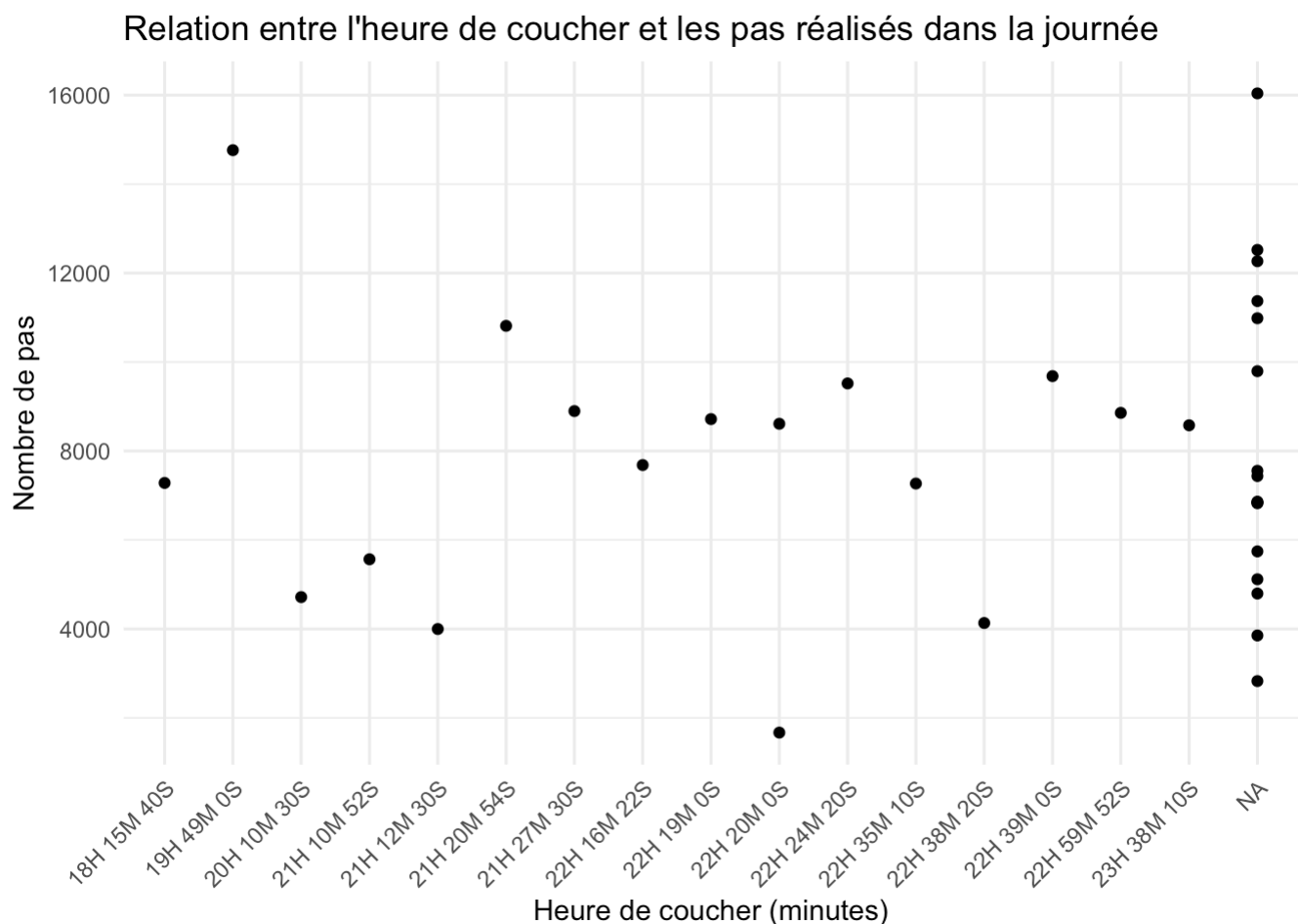
des pas

```
ggplot(user_data) +
  geom_point(aes(x = user_bedtime_m, y = very_active_minutes_m)) +
  theme_minimal() +
  labs(
    title = "Relation entre l'heure de coucher et les minutes d'activité intense",
    x = "Heure de coucher (minutes)",
    y = "Minutes d'activité intense"
  ) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1) # Tourner les étiquettes de
l'axe des x
  )
```

Relation entre l'heure de coucher et les minutes d'activité intense



```
ggplot(user_data) +
  geom_point(aes(x = user_bedtime_m, y = daily_steps_m)) +
  theme_minimal() +
  labs(
    title = "Relation entre l'heure de coucher et les pas réalisés dans la journée",
    x = "Heure de coucher (minutes)",
    y = "Nombre de pas"
  ) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1) # Tourner les étiquettes de
l'axe des x
  )
```



Ces schémas ne montrent pas de corrélation entre les activités de la journée et l'heure du coucher.

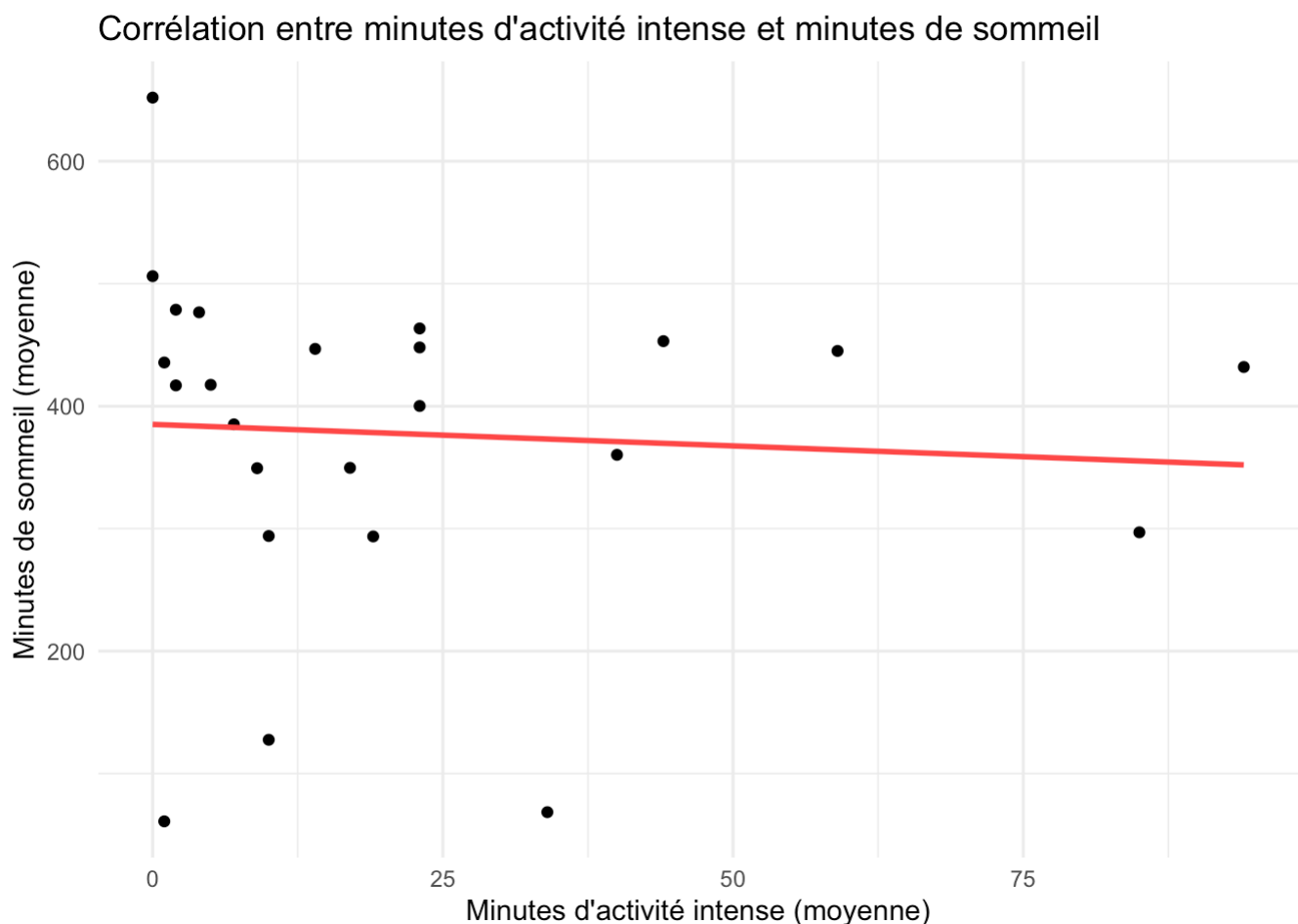
Corrélation entre minutes d'activité intense et minutes de sommeil

```
ggplot(user_data, aes(x = very_active_minutes_m, y = minute_sleep_m)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "#ff4949") +
  labs(title = "Corrélation entre minutes d'activité intense et minutes de sommeil",
       x = "Minutes d'activité intense (moyenne)",
       y = "Minutes de sommeil (moyenne)") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 9 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 9 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



Il n'y a pas ou très peu de corrélation entre le sommeil et la durée d'activité intense.

Corrélation entre minutes de sédentarité et minutes de sommeil

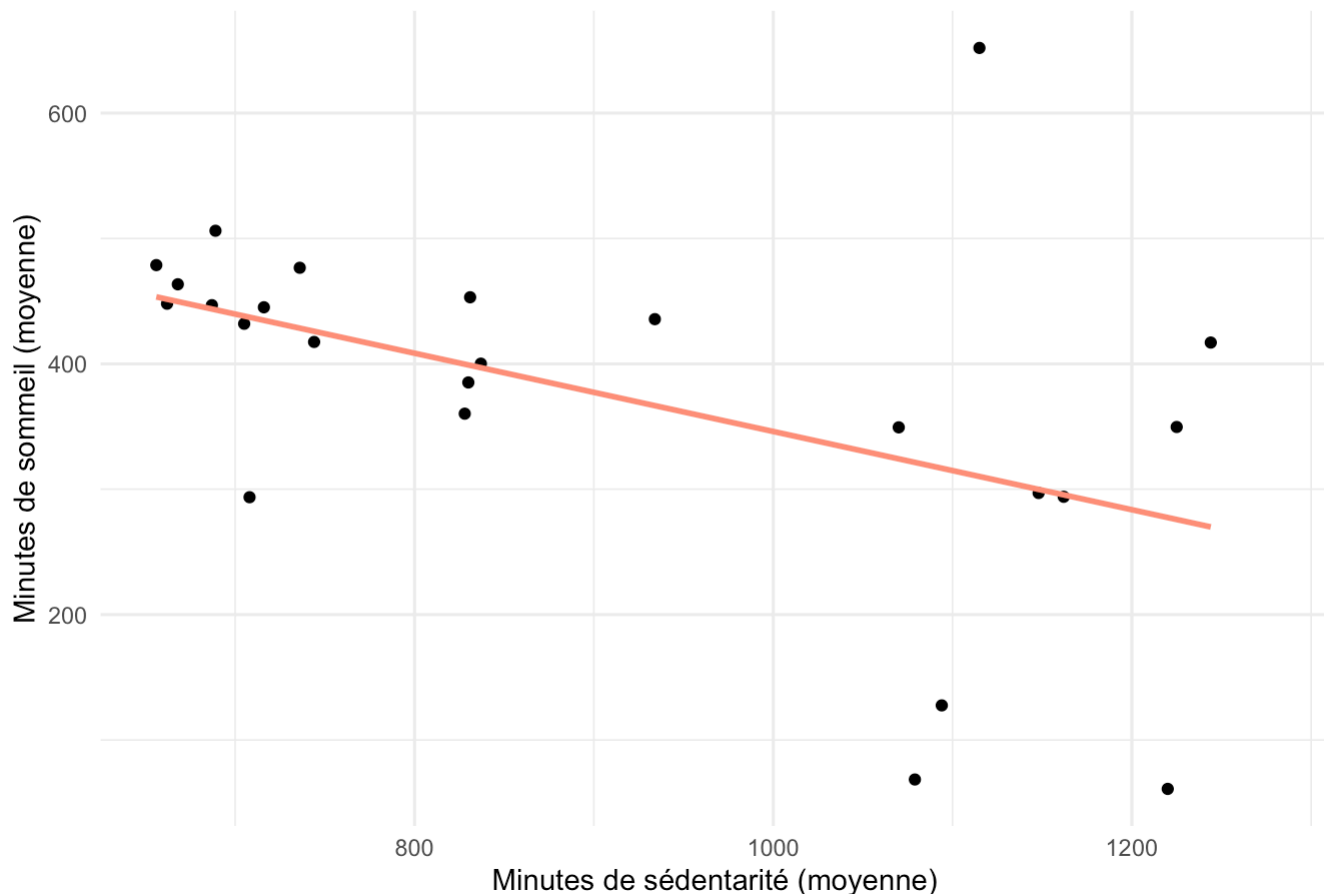
```
ggplot(user_data, aes(x = sedentary_minutes_m, y = minute_sleep_m)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "#fe8f77") +
  labs(title = "Corrélation entre minutes de sédentarité et minutes de sommeil",
        x = "Minutes de sédentarité (moyenne)",
        y = "Minutes de sommeil (moyenne)") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 9 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 9 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

Corrélation entre minutes de sédentarité et minutes de sommeil



Ce schéma montre le lien entre la sédentarité et le sommeil : Moins l'on bouge dans la journée moins l'on dort.

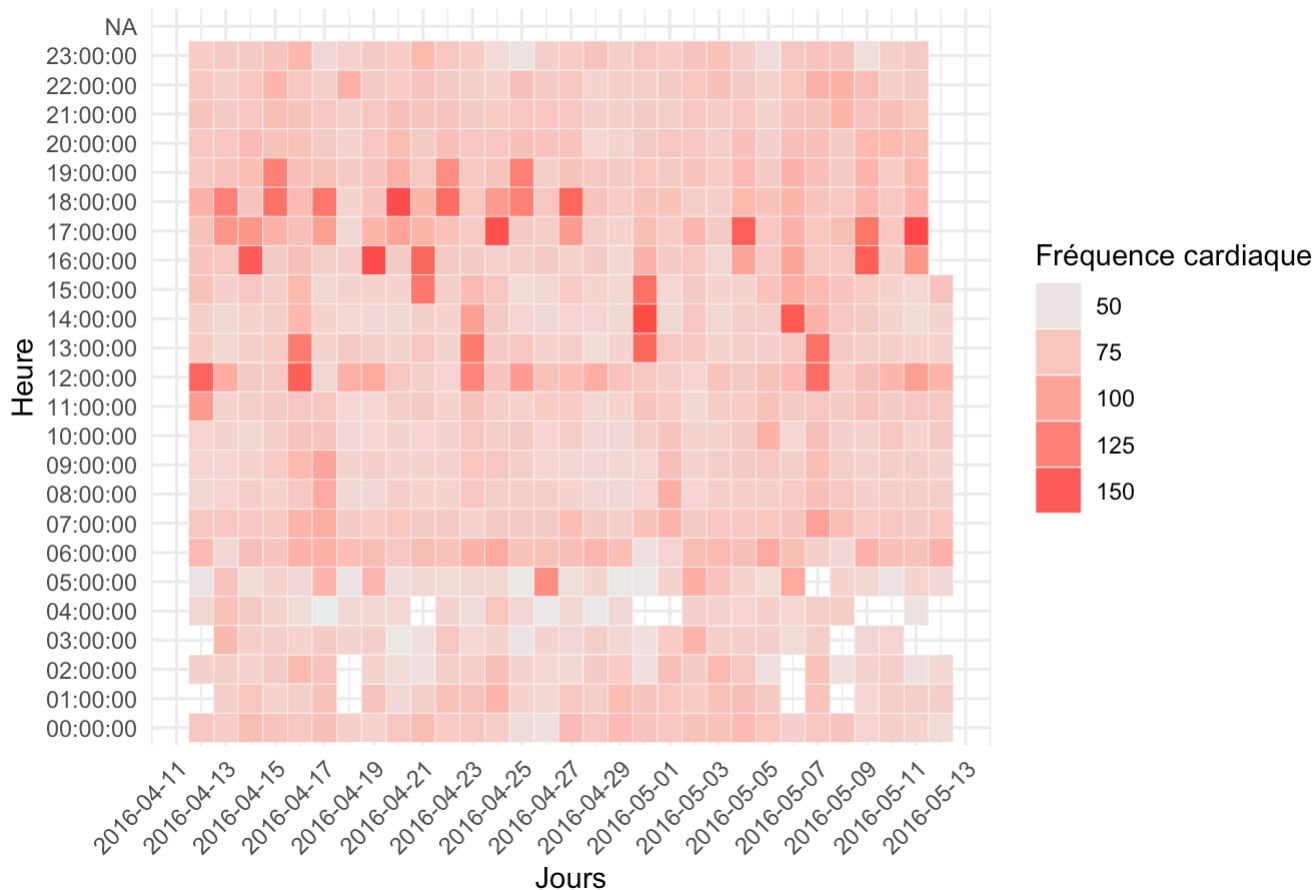
Visualisation de l'activité selon les heures de la journée

```
hourly_data <- hourly_data %>%
  mutate(date = as.Date(date))

ggplot(hourly_data, aes(date, time, fill= heartrate)) +
  geom_tile(color="#ffffff") +
  scale_fill_gradient(low="#e9edee", high="#ff4949") +
  guides(fill = guide_legend(title = "Fréquence cardiaque")) +
  labs(title = "Heatmap de la fréquence cardiaque selon l'heure",
       x = "Jours",
       y = "Heure") +
  scale_x_date(
    date_breaks = "2 days", # Mettre une étiquette toutes les 2 dates
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Removed 1066 rows containing missing values or values outside the scale range
## (`geom_tile()`).
```

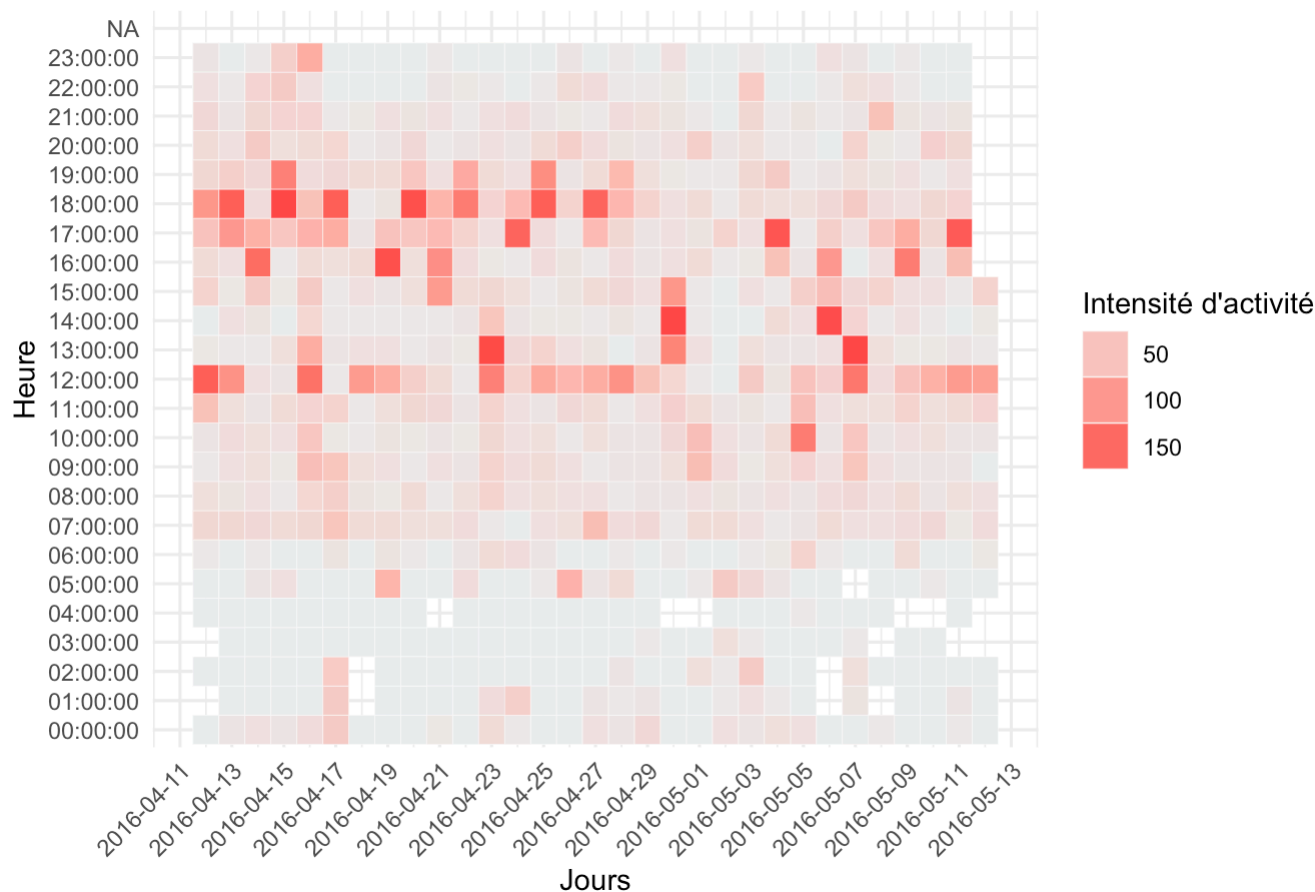
Heatmap de la fréquence cardiaque selon l'heure



```
# =
ggplot(hourly_data, aes(date, time, fill= total_intensity)) +
  geom_tile(color="#ffffff") +
  scale_fill_gradient(low="#e9edee", high="#ff4949") +
  guides(fill = guide_legend(title = "Intensité d'activité")) +
  labs(title = "Heatmap de l'intensité d'activité selon l'heure",
        x = "Jours",
        y = "Heure") +
  scale_x_date(
    date_breaks = "2 days", # Mettre une étiquette toutes les 2 dates
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Removed 1066 rows containing missing values or values outside the scale range
## (`geom_tile()`).
```

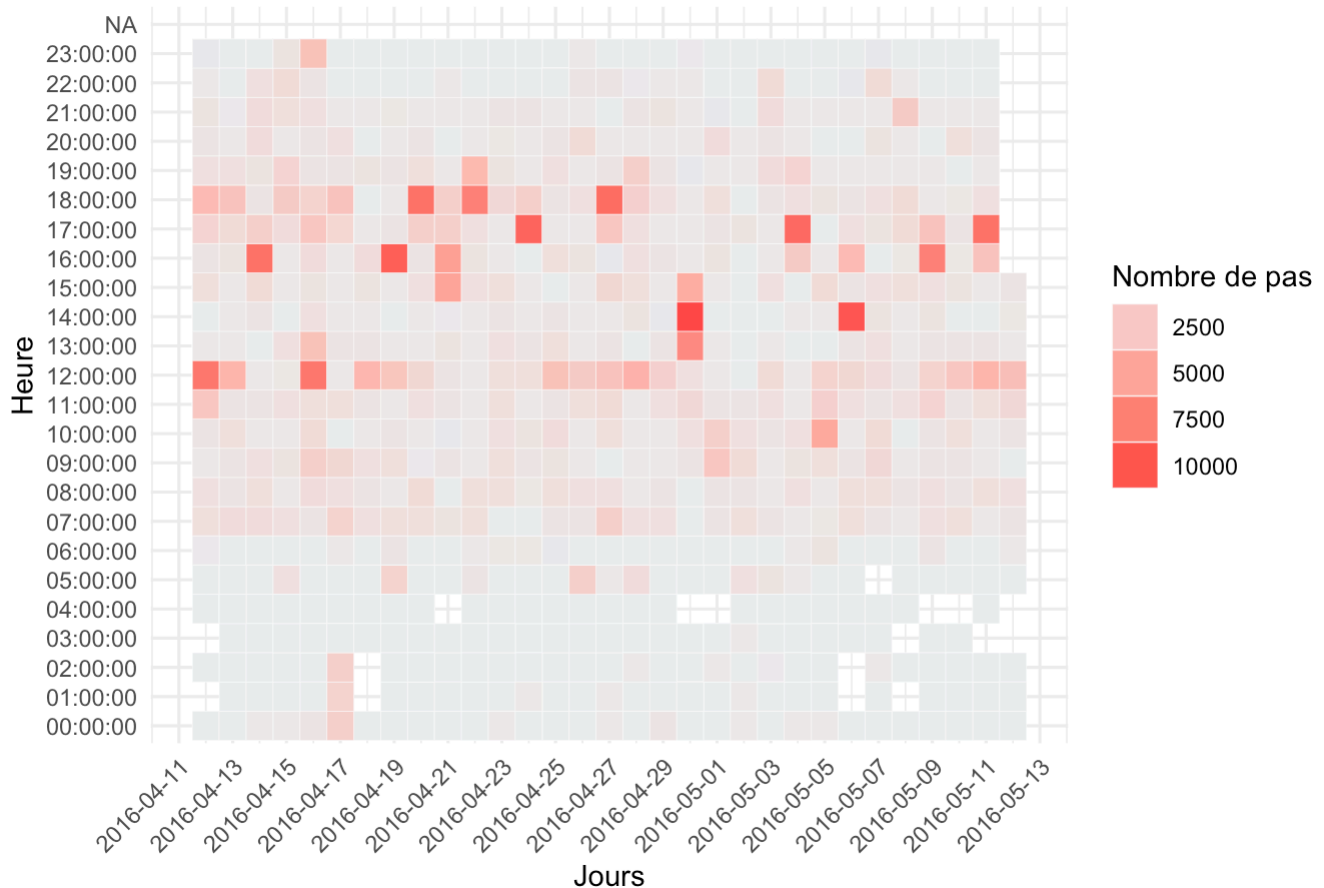

Heatmap de l'intensité d'activité selon l'heure



```
# =
ggplot(hourly_data, aes(date, time, fill= step_total)) +
  geom_tile(color="#ffffff") +
  scale_fill_gradient(low="#e9edee", high="#ff4949") +
  guides(fill = guide_legend(title = "Nombre de pas")) +
  labs(title = "Heatmap du nombre de pas selon l'heure",
        x = "Jours",
        y = "Heure") +
  scale_x_date(
    date_breaks = "2 days", # Mettre une étiquette toutes les 2 dates
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Removed 1066 rows containing missing values or values outside the scale range
## (`geom_tile()`).
```

Heatmap du nombre de pas selon l'heure



Ces trois schémas nous montrent les heures les plus actives de la journée. Les heures étant les mêmes on peut en conclure qu'il y a un lien entre la fréquence cardiaque, les pas et / ou l'activité physique.

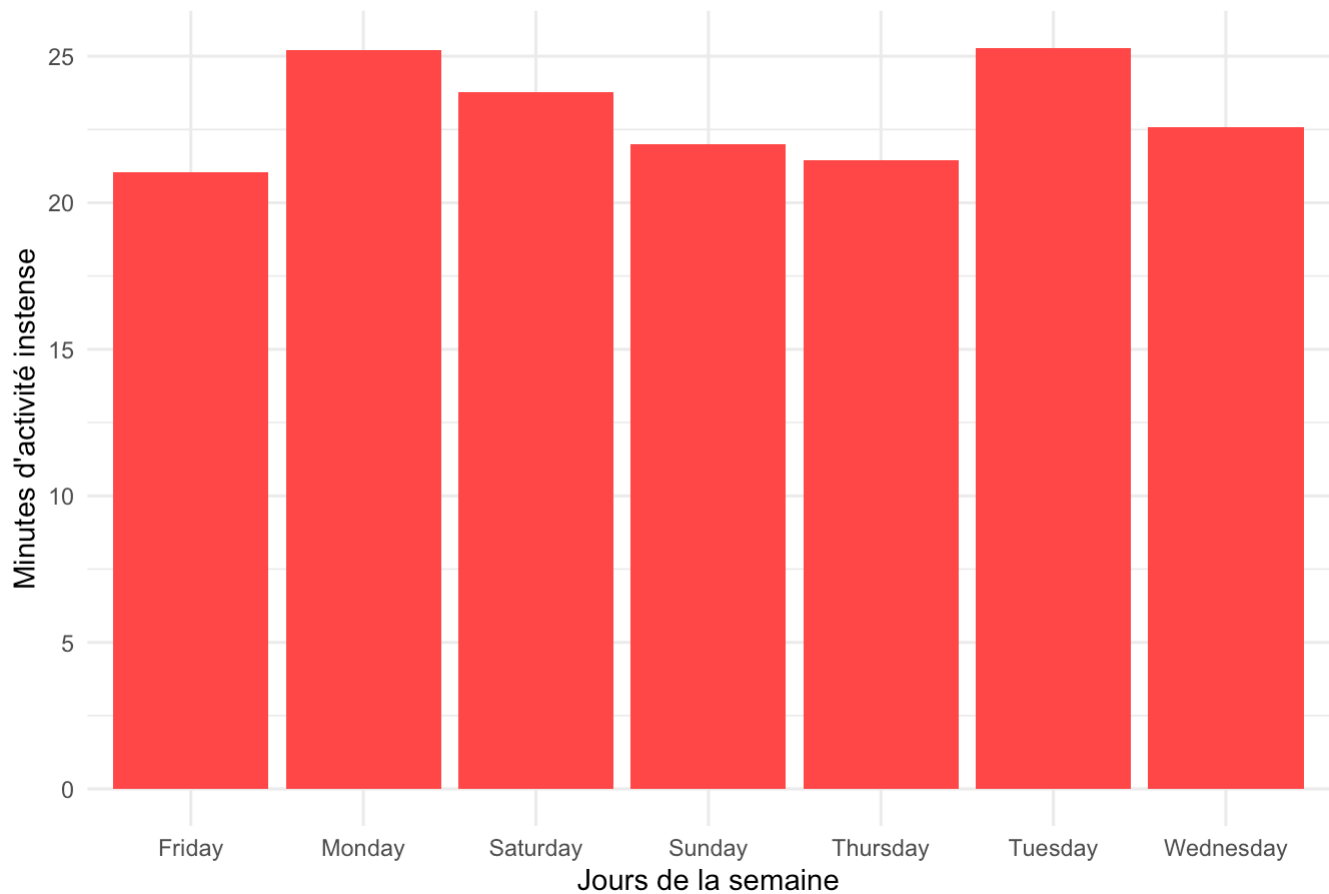
Activité selon le jour de la semaine

```
daily_activity_summary <- daily_activity %>%
  group_by(day) %>%
  summarise(
    avg_very_active_minutes = mean(very_active_minutes),
    avg_sedentary_minutes = mean(sedentary_minutes)
  )

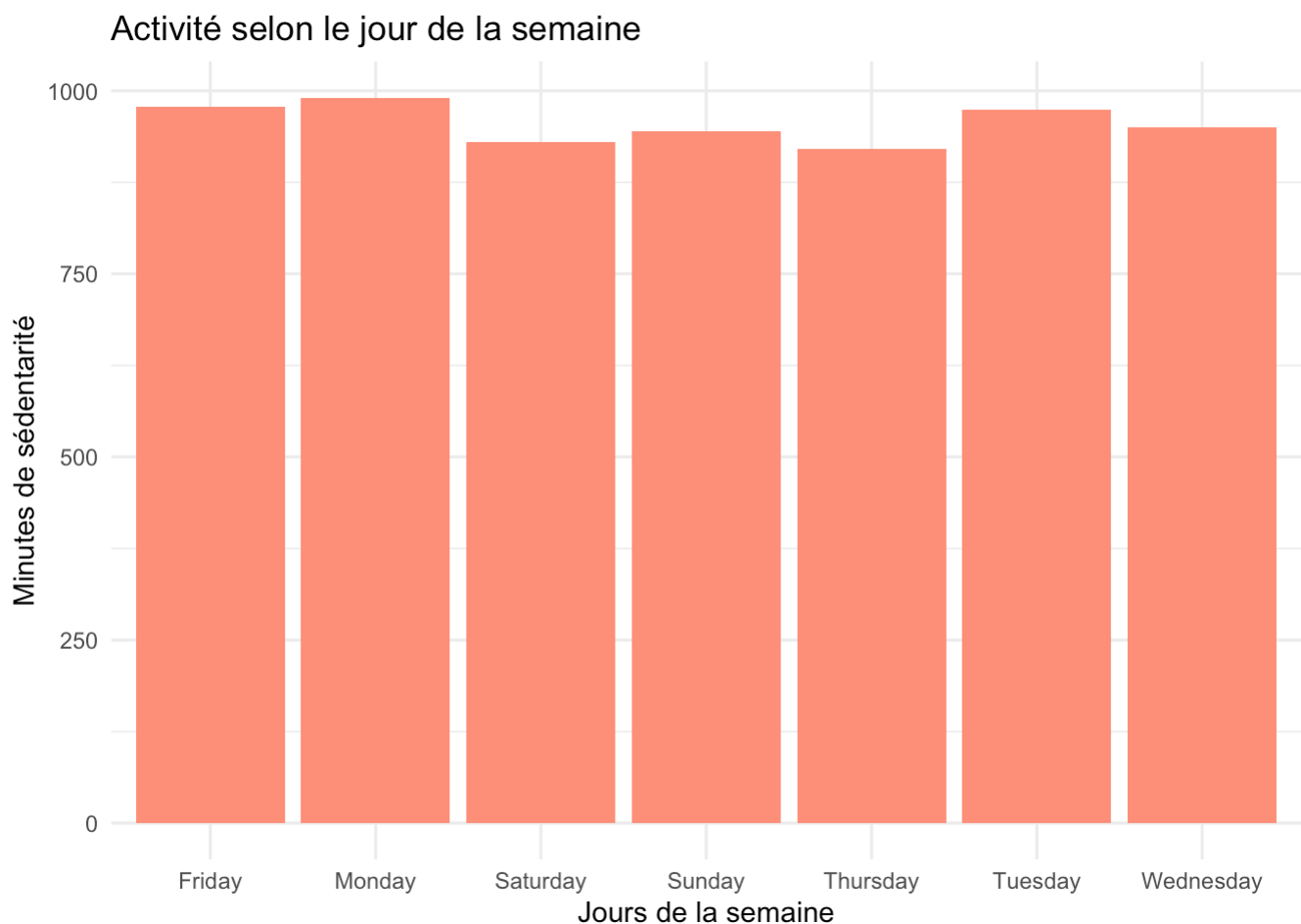
daily_activity$day <- factor(daily_activity$day, c("Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday", "Sunday")) # Ordonne les données en fonction des jours de la semaine

ggplot(daily_activity_summary) +
  geom_bar(aes(x = day, y = avg_very_active_minutes), fill="#ff4949", position = "stack", stat = "identity") +
  theme_minimal() +
  labs(title = "Activité selon le jour de la semaine",
    x = "Jours de la semaine",
    y = "Minutes d'activité instense")
```

Activité selon le jour de la semaine



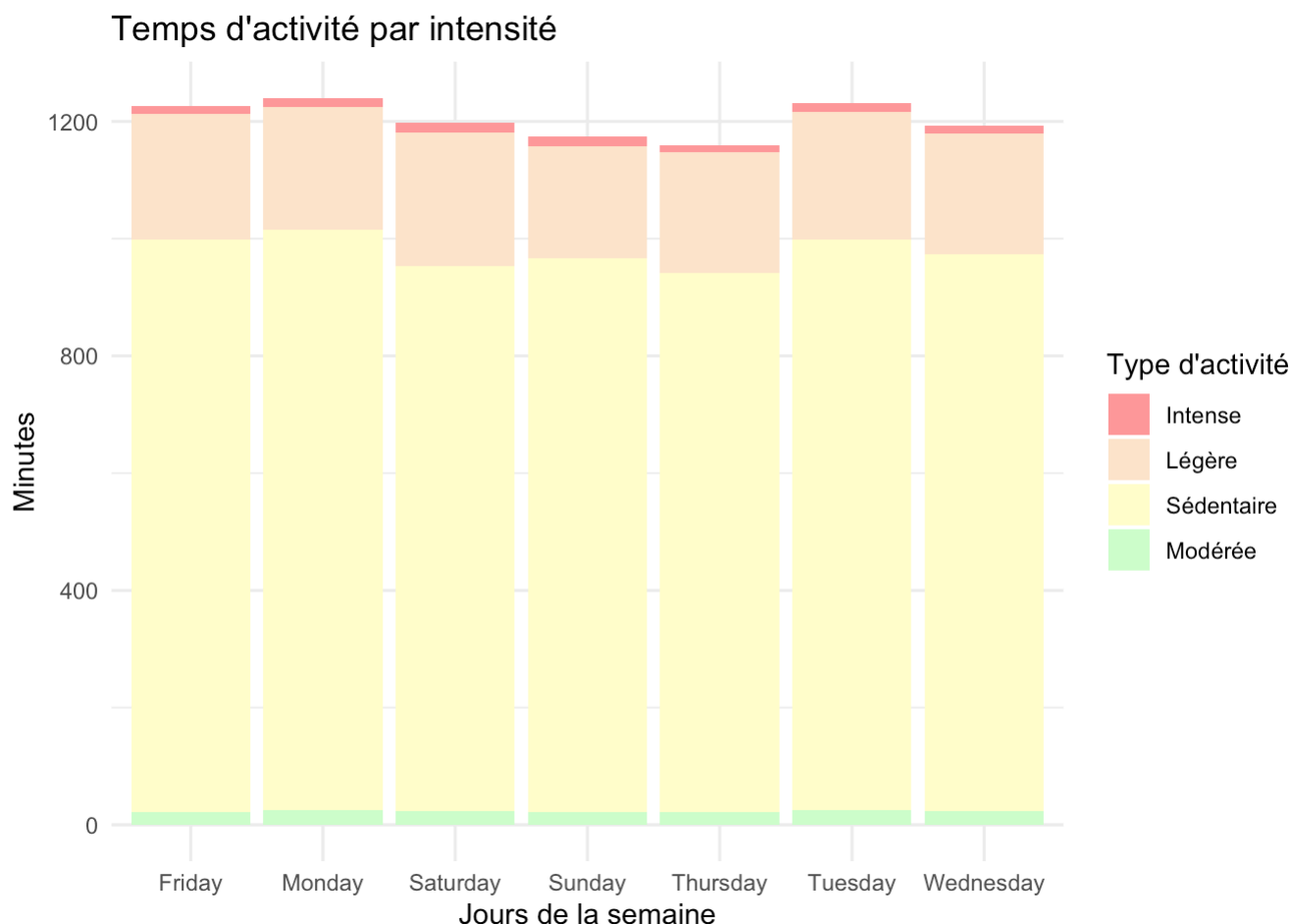
```
ggplot(daily_activity_summary) +  
  geom_bar(aes(x = day, y = avg_sedentary_minutes), fill="#fe8f77", position = "stack", stat = "identity") +  
  
  theme_minimal()+  
  labs(title = "Activité selon le jour de la semaine",  
        x = "Jours de la semaine",  
        y = "Minutes de sédentarité")
```



Ces schémas ne montrent pas de variation impactante.

Temps d'activité par intensité divisé par jour sur le mois

```
ggplot(daily_activity_long) +
  geom_bar(aes(x = day, y = m, fill=activity_type), stat = "identity") +
  theme_minimal() +
  guides(fill = guide_legend(title = "Type d'activité")) +
  scale_fill_manual(values = c("#ff9999", "#ffe6cc", "#ffffcc", "#ccffcc"),
                    labels = c("Intense", "Légère", "Sédentaire", "Modérée")) +
  labs(title = "Temps d'activité par intensité",
       x = "Jours de la semaine",
       y = "Minutes")
```

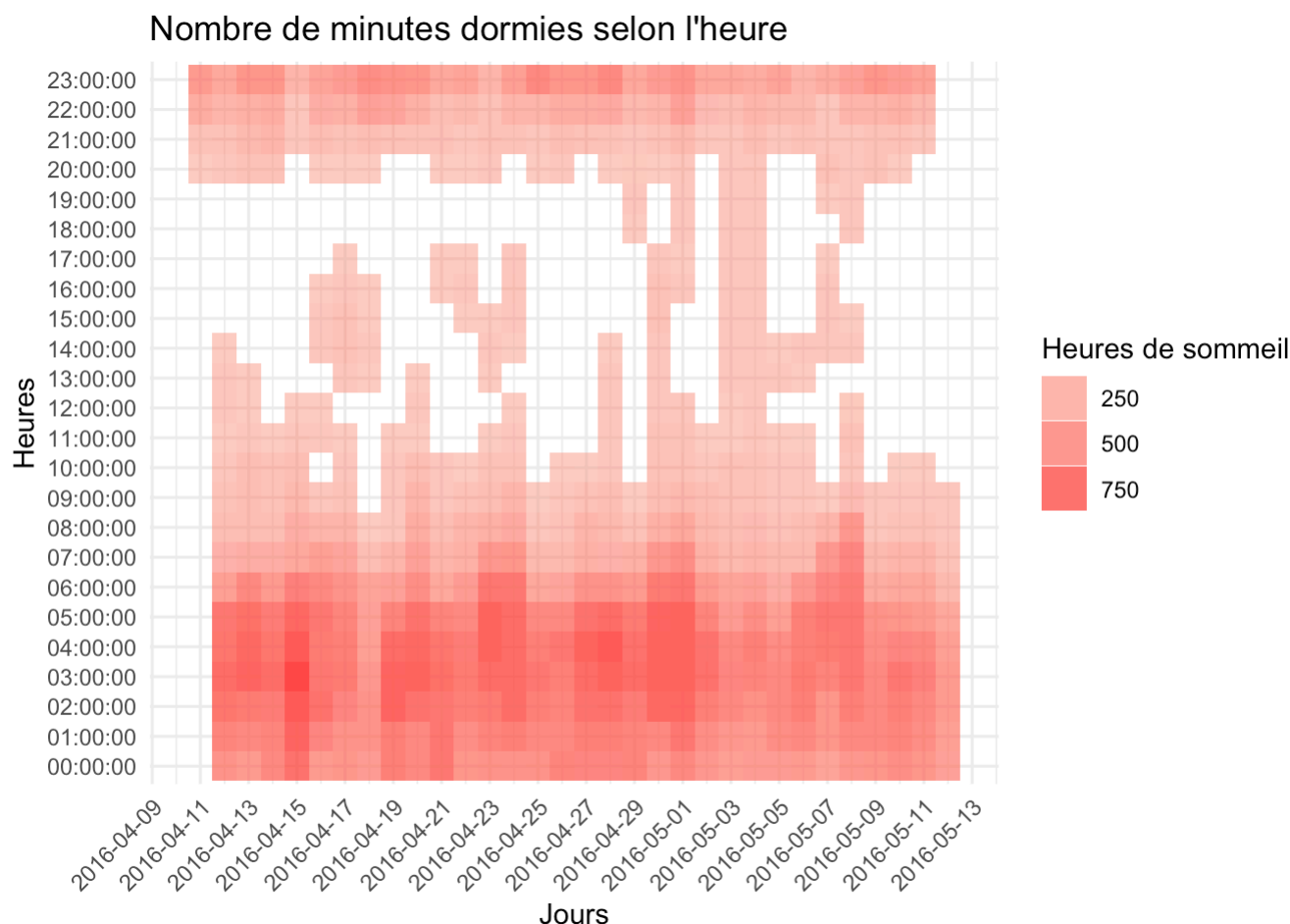


Ce schéma ne montre pas de variation impactante.

Rendu du sommeil sur le mois selon les heures de la journée

```
hour_sleep <- hour_sleep %>%
  mutate(date = as.Date(date))

ggplot(hour_sleep, aes(date, time, fill = count)) +
  geom_tile() +
  scale_fill_gradient(low = "#fe8f7775", high = "#ff4949") +
  guides(fill = guide_legend(title = "Heures de sommeil")) +
  labs(title = "Nombre de minutes dormies selon l'heure",
       x = "Jours",
       y = "Heures") +
  scale_x_date(
    date_breaks = "2 days",
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

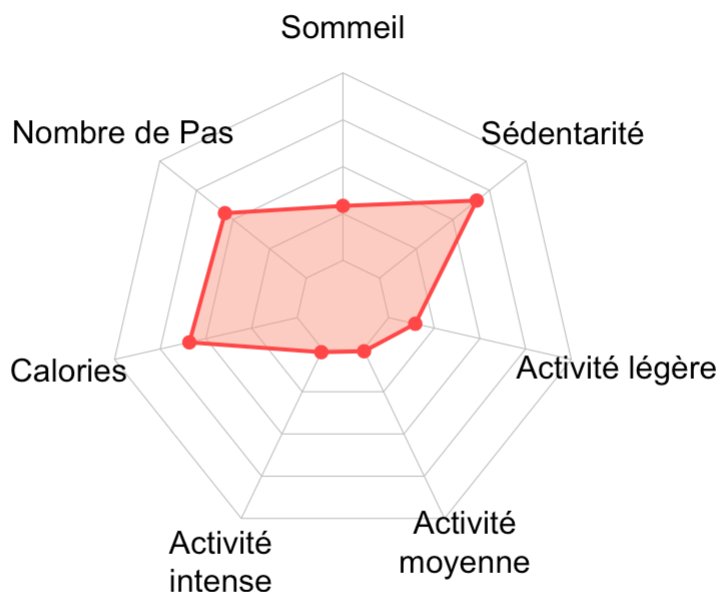


Cette visualisation montrent clairement les périodes d'endormissement des utilisateurs. On peut y voir quelques sieste dans la journée et un maximum de dormeurs de 23h à 6h.

Comparaison des moyennes (sommeil, pas; calories, activité et sédentarité) générale, en semaine et en weekend

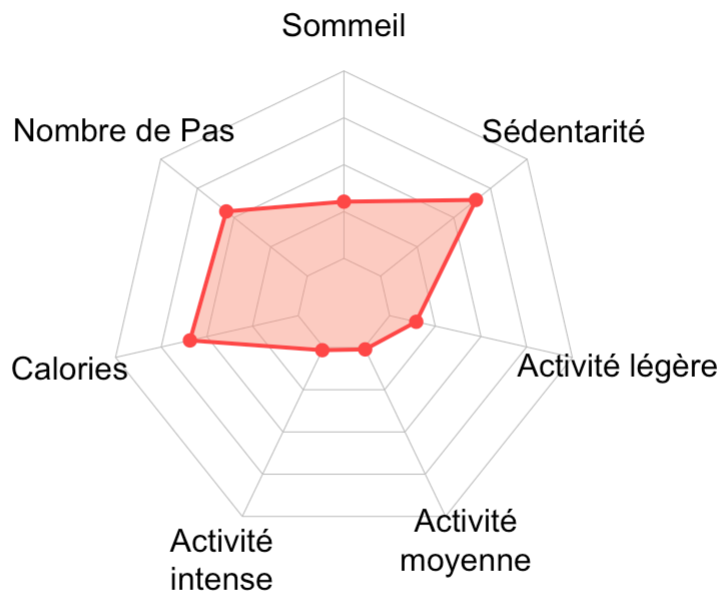
```
radarchart(
  week_activity_m,
  axistype = 5,
  pcol = "#ff4949",
  pfcoll = "#fe8f7775",
  plwd = 2,
  cglcol = "grey",
  cglty = 1,
  axislabcol = "transparent",
  caxislabels = "",
  cglwd = 0.5,
  title = "Moyennes d'Activité des Utilisateurs en Semaine",
  vlabels = c("Sommeil", "Nombre de Pas", "Calories", "Activité\nintense", "Activité\nmoyenne\n", "Activité légère", "Sédentarité"),
  palcex = 1
)
```

Moyennes d'Activité des Utilisateurs en Semaine



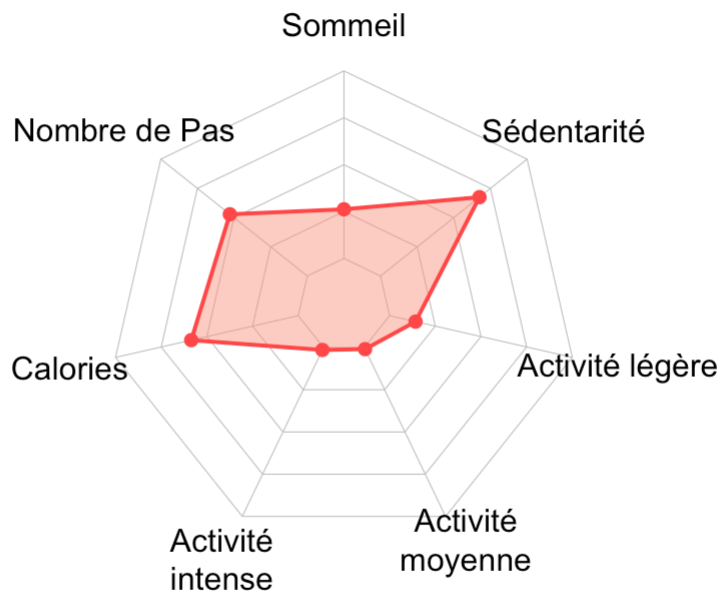
```
radarchart(
  weekend_activity_m,
  axistype = 5,
  pcol = "#ff4949",
  pfcoll = "#fe8f7775",
  plwd = 2,
  cglcol = "grey",
  cglty = 1,
  axislabcol = "transparent",
  caxislabels = "",
  cglwd = 0.5,
  title = "Moyennes d'Activité des Utilisateurs en Weekend",
  vlabels = c("Sommeil", "Nombre de Pas", "Calories", "Activité\nintense", "Activité\nmoyenne\n", "Activité légère", "Sédentarité"),
)
```

Moyennes d'Activité des Utilisateurs en Weekend



```
radarchart(
  user_data_m,
  axistype = 5,
  pcol = "#ff4949",
  pfcoll = "#fe8f7775",
  plwd = 2,
  cglcol = "grey",
  cglty = 1,
  axislabcol = "transparent",
  caxislabels = "",
  cglwd = 0.5,
  title = "Moyennes d'Activité des Utilisateurs",
  vlabels = c("Sommeil", "Nombre de Pas", "Calories", "Activité\nintense", "Activité\nmoyenne\n", "Activité légère", "Sédentarité"),
  palcex = 1
)
```


Moyennes d'Activité des Utilisateurs

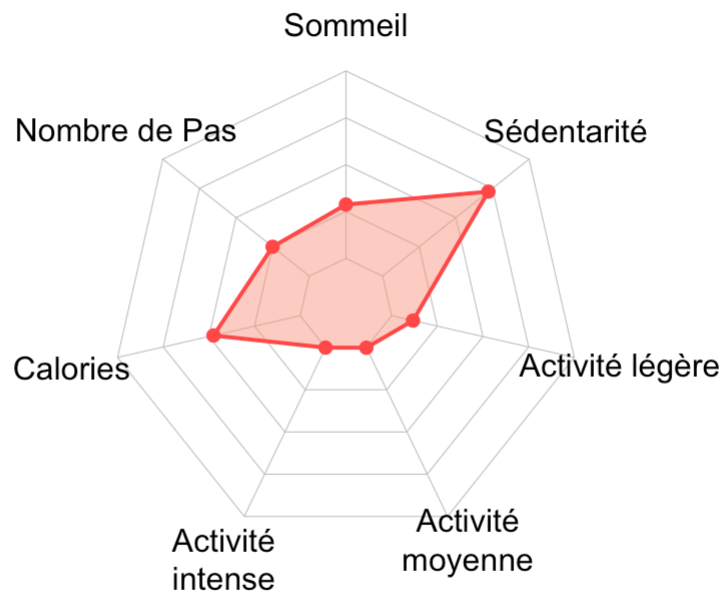


Les utilisateurs ne semblent pas avoir d'habitudes différentes la semaine et le week end.

Ayant créé un score d'activité nous reprenons ces schémas pour les 5 utilisateurs les plus actives, ainsi que les 5 les moins actifs.

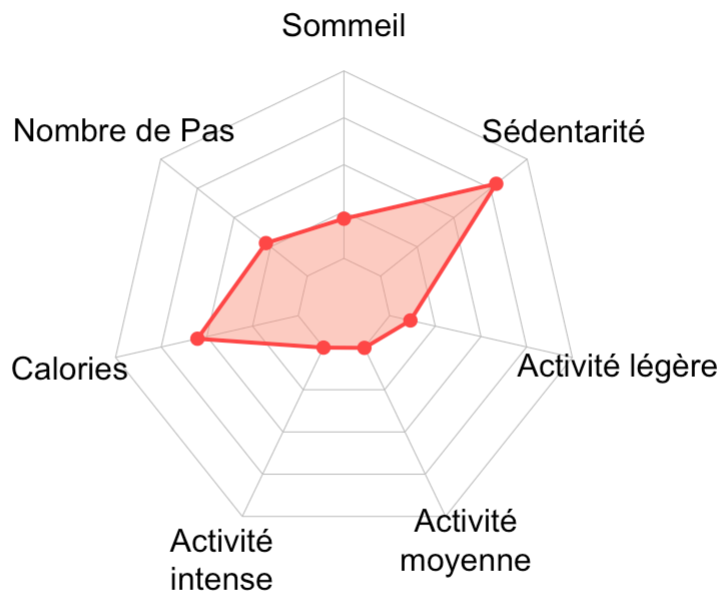
```
# Moyennes des 5 utilisateurs les moins actifs
radarchart(
  most_active_m,
  axistype = 5,
  pcol = "#ff4949",
  pfcoll = "#fe8f7775",
  plwd = 2,
  cglcol = "grey",
  cglty = 1,
  axislabcol = "transparent",
  caxislabels = "",
  cglwd = 0.5,
  title = "Moyennes des 5 utilisateurs les plus actifs",
  vlabels = c("Sommeil", "Nombre de Pas", "Calories", "Activité\nintense", "Activité\nmoyenne\n", "Activité légère", "Sédentarité"),
  palcex = 1
)
```

Moyennes des 5 utilisateurs les plus actifs



```
# Moyennes des 5 utilisateurs les moins actifs
radarchart(
  less_active_m,
  axistype = 5,
  pcol = "#ff4949",
  pfcoll = "#fe8f7775",
  plwd = 2,
  cglcol = "grey",
  cglty = 1,
  axislabcol = "transparent",
  caxislabels = "",
  cglwd = 0.5,
  title = "Moyennes des 5 utilisateurs les moins actifs",
  vlabels = c("Sommeil", "Nombre de Pas", "Calories", "Activité\nintense", "Activité\nmoyenne\n", "Activité légère", "Sédentarité"),
  palcex = 1
)
```

Moyennes des 5 utilisateurs les moins actifs



La différence la plus notable est le nombre de pas et non l'intensité d'activité.

Constations sur les données

Les valeurs qui pour moi ressortent le plus de cette analyse sont le manque de régularité dans le port de la montre. Cela peut être dû à l'autonomie faible des Fit Bit. Les produits Bellabeat ayant une plus grande autonomie nous pouvons espérer que les utilisatrices Bellabeat soient plus rigoureuses sur ce point.

Nous avons pu voir aussi que les utilisateurs Fit Bit ne portent pas tous leurs montres la nuit, encore une fois cela peut être dû au fait que certains utilisateurs mettent leurs montres à charger la nuit, ce qui sera moins le cas pour les utilisatrices Bellabeat.

Nous avons pu établir différents planning journaliers cela permet de connaître les heures d'activité et les heures de sommeil habituels des utilisateurs.

Nous avons aussi constaté un faible taux d'utilisateurs qui se pèsent ou qui, en tout cas, rajoutent leurs poids à leurs applications.

Conclusion et Axes d'actions

De ces constations nous pouvons tirer des axes d'amélioration pour Bellabeat :

- L'Application Bellabeat peut s'appuyer sur les plannings fournis pour personnaliser l'expérience des utilisatrices : des notifications pour bouger lors des phases de sédentarité, des notifications de motivation pour les heures où les utilisatrices vont d'habitude faire leur séance de sport, notifications quelques heures avant les repas pour motiver à manger sainement, notification bien-être avant le coucher, ...

- Les utilisateurs de fit bit ayant le “score” le plus élevé n’étant pas ceux qui font les séances de sport les plus intensives mais ceux marchant le plus, Bellabeat peut axer sa communication sur les bienfaits de la marche. En effet la marche est l’activité qui est adapté à tous, elle est préconisée par L’Organisation Mondiale de la Santé pour ses bienfaits sur le corps et l’esprit.
- Pour aussi avoir acquérir plus de données, de clients et donc de chiffres d’affaires, il peut être une bonne idée de créer une balance connectée. Vendue séparément ou dans un pack, tracker + balance, les utilisateurs n’en seront que plus motivées à personnaliser encore plus leurs expériences Bellabeat et à prendre soin d’elle.