# R Notebook - Emily Liang

Step 1: Load the packages.

```
#install.packages('tidyverse')
#install.packages('skimr')
#install.packages('cowplot')
#install.packages("plotly")
library(plotly)
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(tidyverse) #wrangle data
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks plotly::filter(), stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr) #clean data
library(lubridate)  #wrangle date attributes
```

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

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

```
library(skimr) #get summary data
library(ggplot2) #visualize data
library(cowplot) #grid the plot
```

```
##
## Attaching package: 'cowplot'
```

```
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
library(readr) #save csv
library(plotly) #pie chart
```

Step 2: Prepare the data and if needed, combine them in one data frame.

```
setwd("C:/Users/Emily/Downloads/data")

daily_activity <- read.csv("dailyActivity_merged.csv")
sleep_day <- read.csv("sleepDay_merged.csv")
weight <- read.csv("weightLogInfo_merged.csv")
hourly_step <- read.csv("hourlySteps_merged.csv")
head(daily_activity)
```

| | Id | ActivityDate | TotalSteps | TotalDistance | TrackerDistance | LoggedActivities |
|---|---:|---|---:|---:|---:|---:|
| | <dbl> | <chr> | <int> | <dbl> | <dbl> | |
| 1 | 1503960366 | 4/12/2016 | 13162 | 8.50 | 8.50 | |
| 2 | 1503960366 | 4/13/2016 | 10735 | 6.97 | 6.97 | |
| 3 | 1503960366 | 4/14/2016 | 10460 | 6.74 | 6.74 | |
| 4 | 1503960366 | 4/15/2016 | 9762 | 6.28 | 6.28 | |
| 5 | 1503960366 | 4/16/2016 | 12669 | 8.16 | 8.16 | |
| 6 | 1503960366 | 4/17/2016 | 9705 | 6.48 | 6.48 | |

6 rows | 1-7 of 16 columns

```
head(sleep_day)
```

| | Id | SleepDay | TotalSleepRecords | TotalMinutesAsleep | TotalTimeIn |
|---|---|---|---|---|---|
| | <dbl> | <chr> | <int> | <int> | < |
| 1 | 1503960366 | 4/12/2016 12:00:00 AM | 1 | 327 | |
| 2 | 1503960366 | 4/13/2016 12:00:00 AM | 2 | 384 | |
| 3 | 1503960366 | 4/15/2016 12:00:00 AM | 1 | 412 | |
| 4 | 1503960366 | 4/16/2016 12:00:00 AM | 2 | 340 | |
| 5 | 1503960366 | 4/17/2016 12:00:00 AM | 1 | 700 | |
| 6 | 1503960366 | 4/19/2016 12:00:00 AM | 1 | 304 | |

6 rows

```
head(weight)
```

| | Id | Date | Weight… | WeightPounds | … | BMI | IsManualReport | |
|---|---|---|---|---|---|---|---|---|
| | <dbl> | <chr> | <dbl> | <dbl> | <int> | <dbl> | <chr> | |
| 1 | 1503960366 | 5/2/2016 11:59:59 PM | 52.6 | 115.9631 | 22 | 22.65 | True | 1.4 |
| 2 | 1503960366 | 5/3/2016 11:59:59 PM | 52.6 | 115.9631 | NA | 22.65 | True | 1.4 |
| 3 | 1927972279 | 4/13/2016 1:08:52 AM | 133.5 | 294.3171 | NA | 47.54 | False | 1.4 |
| 4 | 2873212765 | 4/21/2016 11:59:59 PM | 56.7 | 125.0021 | NA | 21.45 | True | 1.4 |
| 5 | 2873212765 | 5/12/2016 11:59:59 PM | 57.3 | 126.3249 | NA | 21.69 | True | 1.4 |
| 6 | 4319703577 | 4/17/2016 11:59:59 PM | 72.4 | 159.6147 | 25 | 27.45 | True | 1.4 |

6 rows

```
#Check for NA and duplicates
sum(is.na(daily_activity))
```

```
## [1] 0
```

```
sum(is.na(sleep_day))
```

```
## [1] 0
```

```
sum(is.na(weight))
```

```
## [1] 65
```

```
sum(duplicated(daily_activity))
```

```
## [1] 0
```

```
sum(duplicated(sleep_day))
```

```
## [1] 3
```

```
sum(duplicated(weight))
```

```
## [1] 0
```

```
#We will leave the NA. The NA belong to "Fat" data of different dates.
#Remove duplicates.
sleep_day <- sleep_day[!duplicated(sleep_day), ]
sum(duplicated(sleep_day))
```

```
## [1] 0
```

```
#Add a new column for the weekdays
daily_activity <- daily_activity %>% mutate( Weekday = weekdays(as.Date(ActivityDate, "%m/%d/%Y"
)))

merged1 <- merge(daily_activity,sleep_day,by = c("Id"), all=TRUE)
merged_data <- merge(merged1, weight, by = c("Id"), all=TRUE)

#Order from Monday to Sunday for plot later
merged_data$Weekday <- factor(merged_data$Weekday, levels= c("Monday",
    "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"))

merged_data[order(merged_data$Weekday), ]
```

| | Id | ActivityDate | TotalSteps | TotalDistance | TrackerDistance | LoggedActivit |
|---|---|---|---|---|---|---|
| | <dbl> | <chr> | <int> | <dbl> | <dbl> | |
| 69 | 1503960366 | 4/18/2016 | 13019 | 8.59 | 8.59 | |
| 70 | 1503960366 | 4/18/2016 | 13019 | 8.59 | 8.59 | |
| 71 | 1503960366 | 4/18/2016 | 13019 | 8.59 | 8.59 | |
| 72 | 1503960366 | 4/18/2016 | 13019 | 8.59 | 8.59 | |
| 73 | 1503960366 | 4/18/2016 | 13019 | 8.59 | 8.59 | |
| 74 | 1503960366 | 4/18/2016 | 13019 | 8.59 | 8.59 | |
| 81 | 1503960366 | 4/18/2016 | 13019 | 8.59 | 8.59 | |

| | Id | ActivityDate | TotalSteps | TotalDistance | TrackerDistance | LoggedActivit |
|---|---|---|---|---|---|---|
| | <dbl> | <chr> | <int> | <dbl> | <dbl> | |
| 82 | 1503960366 | 4/18/2016 | 13019 | 8.59 | 8.59 | |
| 89 | 1503960366 | 4/18/2016 | 13019 | 8.59 | 8.59 | |
| 90 | 1503960366 | 4/18/2016 | 13019 | 8.59 | 8.59 | |

1-10 of 10,000 rows | 1-7 of 28 columns Previous **1** 2 3 4 5 6 ... 1000 Next

```
#Save CSV for Tableau presentation
write_csv(merged_data, "merged_data.csv")

#Check for NA and duplicates in merged data.
sum(is.na(merged_data))
```

```
## [1] 98978
```

```
sum(duplicated(merged_data))
```

```
## [1] 0
```

```
n_distinct(merged_data$Id)
```

```
## [1] 33
```

Step 3: Examine the dataset and check if all 30 users are unique.

```
#Check to see if all users are unique.We supposed to have 30 users or 30 IDs. So We have 3 extra
from daily activity, 6 less from the sleep day table, and 22 less from the weight table.
n_distinct(daily_activity$Id)
```

```
## [1] 33
```

```
n_distinct(sleep_day$Id)
```

```
## [1] 24
```

```
n_distinct(weight$Id)
```

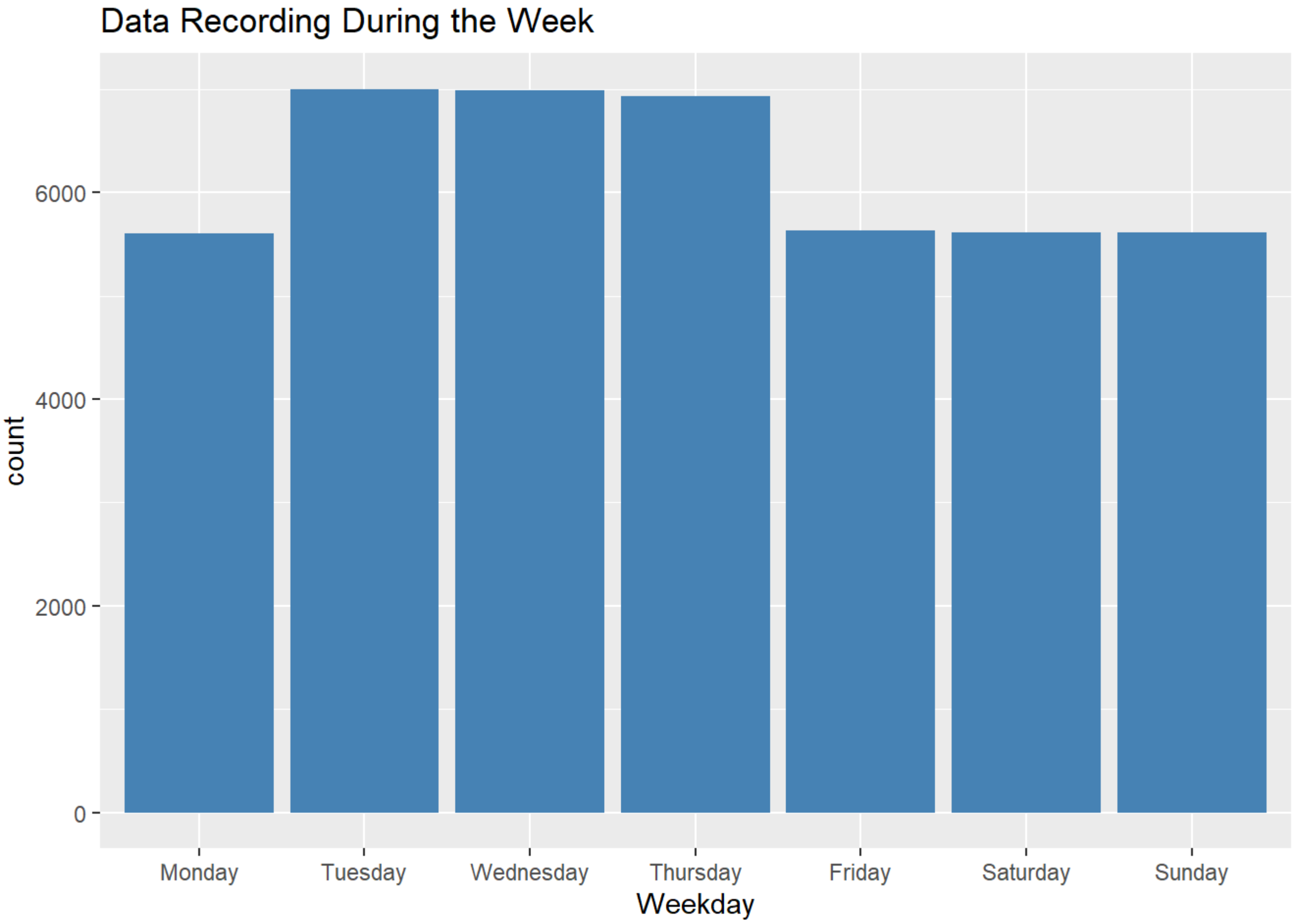
```
## [1] 8
```

```
#Since weight table only has 8 users enter their information. Let's take a look at how they enter the information. 5 users are manually reporting the weight and 3 uers are reporting it with a connected device - wifi connected scale.
weight %>%
  filter(IsManualReport == "True") %>%
  group_by(Id) %>%
  summarise("Manual Weight Report"=n()) %>%
  distinct()
```

| Id | Manual Weight Report |
|---:|---:|
| <dbl> | <int> |
| 1503960366 | 2 |
| 2873212765 | 2 |
| 4319703577 | 2 |
| 4558609924 | 5 |
| 6962181067 | 30 |

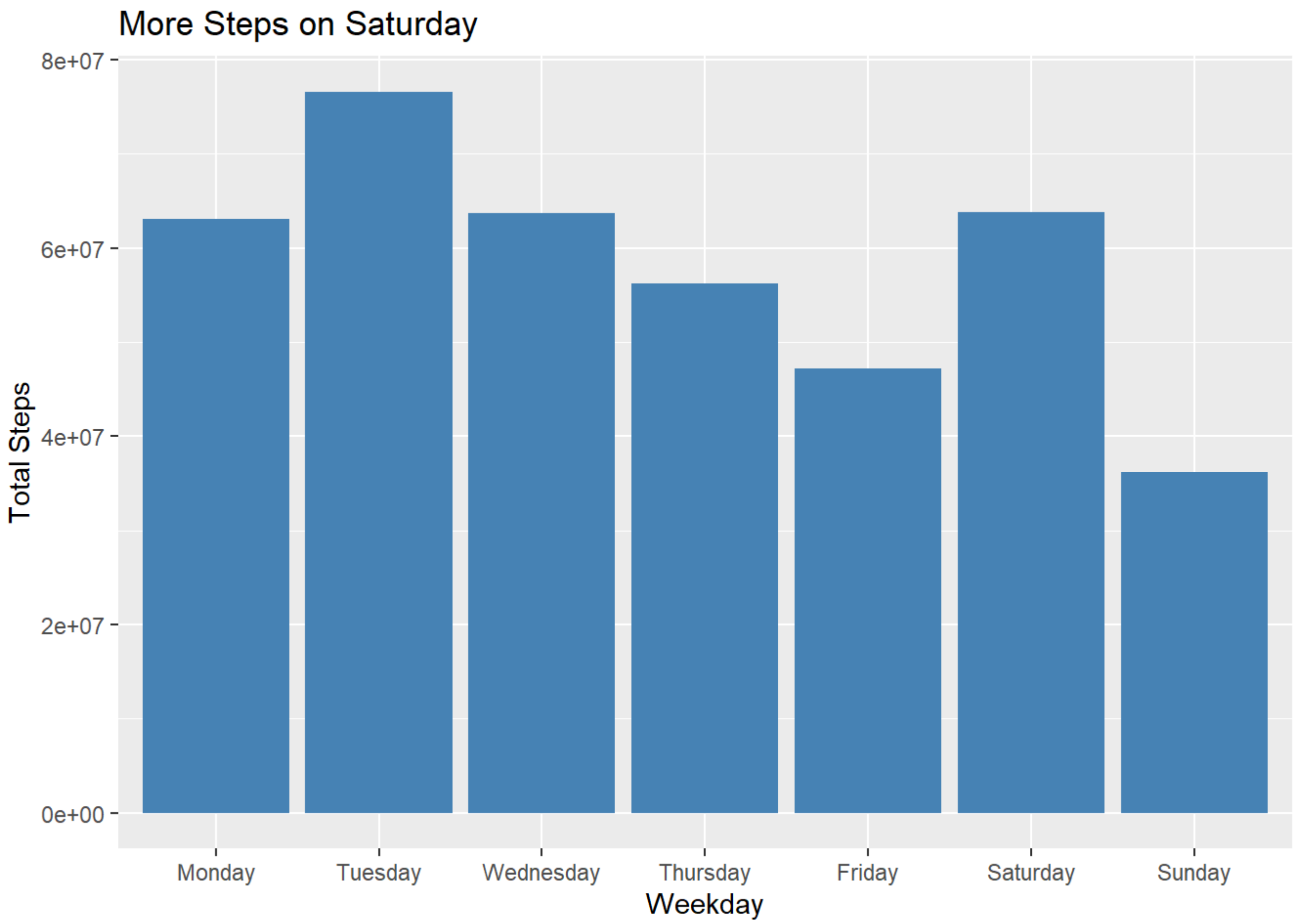
5 rows

```
#When are users most active in recording their data. We noticed users track their data more from Tuesday to Thursday and we have more of those days' data than other days.
ggplot(data=merged_data, aes(x=Weekday))+
  geom_bar(fill="steelblue")+
  labs(title="Data Recording During the Week")
```

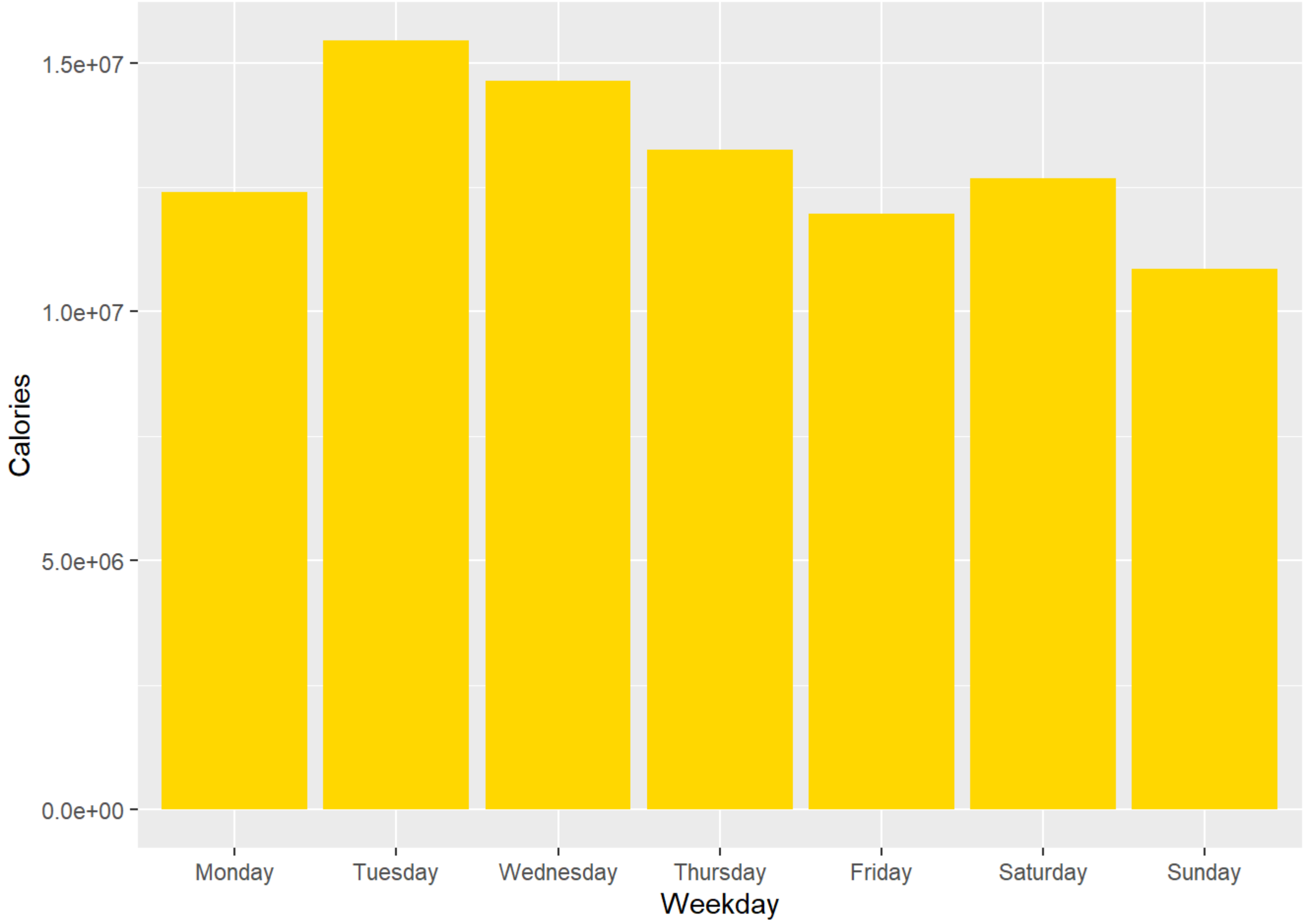


Step 4: Weekly and hourly summary

```
#Weekly
ggplot(data=merged_data, aes(x=Weekday, y=TotalSteps, fill=Weekday))+
  geom_bar(stat="identity", fill="steelblue")+
  labs(title="More Steps on Saturday", y="Total Steps")
```

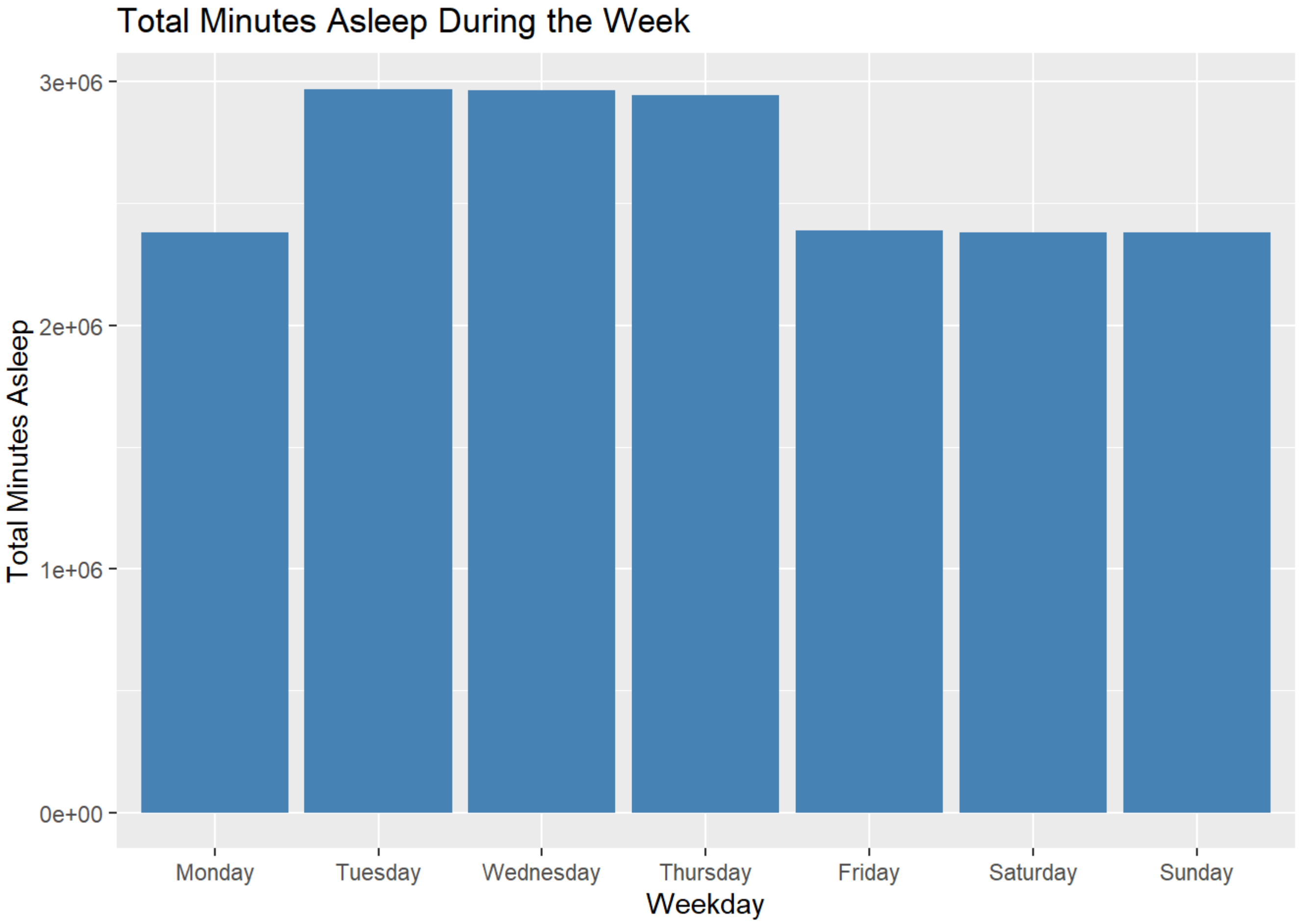


```
ggplot(data=merged_data, aes(x=Weekday, y=Calories, fill=Weekday))+
  geom_bar(stat="identity", fill="gold")
```



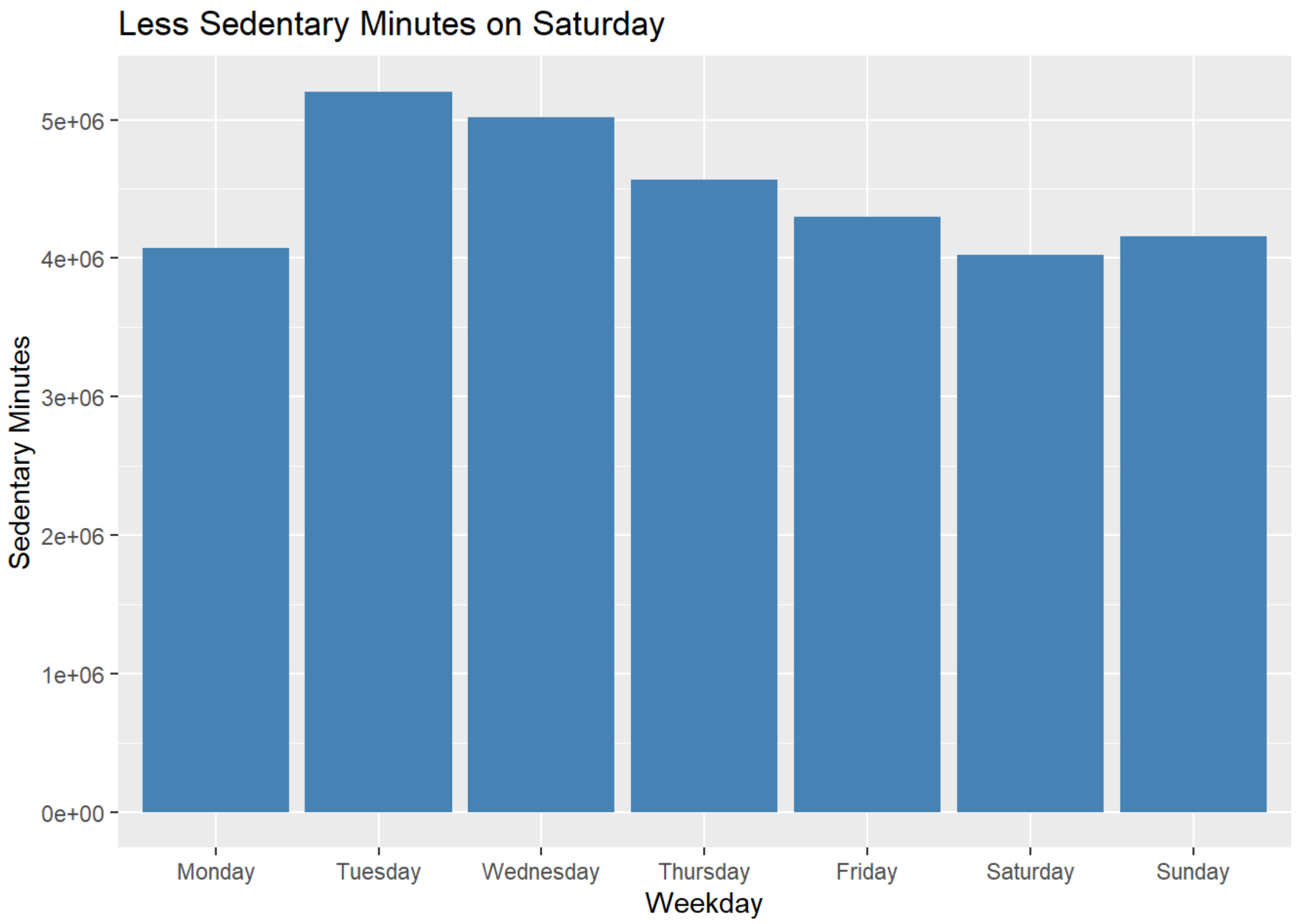
```
ggplot(data=merged_data, aes(x=Weekday, y=TotalMinutesAsleep, fill=Weekday))+
  geom_bar(stat="identity", fill="steelblue")+
  labs(title="Total Minutes Asleep During the Week", y="Total Minutes Asleep")
```

```
## Warning: Removed 971 rows containing missing values (position_stack).
```

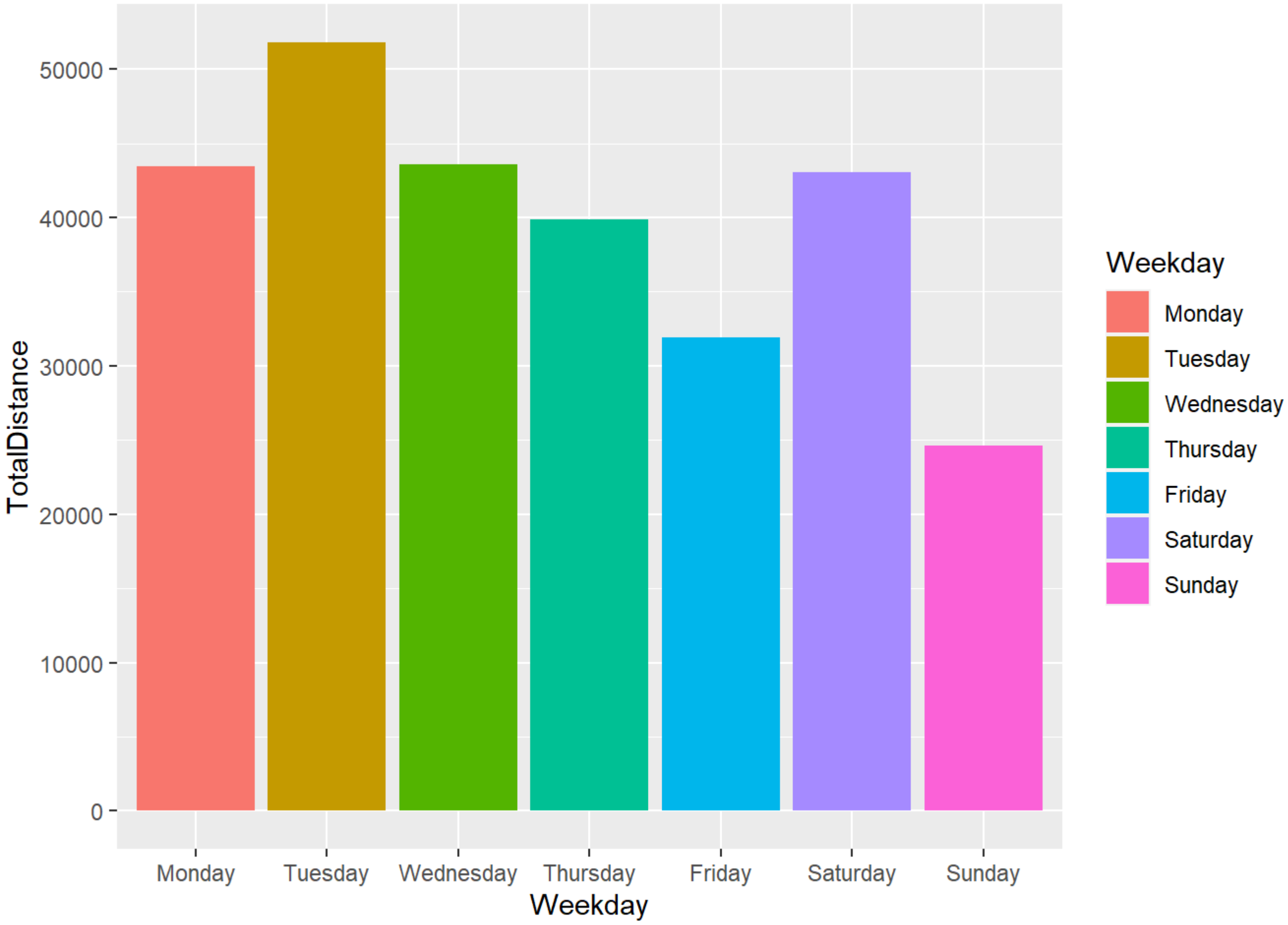


Total Minutes Asleep During the Week

```
ggplot(data=merged_data, aes(x=Weekday, y=SedentaryMinutes, fill=Weekday))+
  geom_bar(stat="identity", fill="steelblue")+
  labs(title="Less Sedentary Minutes on Saturday", y="Sedentary Minutes")
```



```
ggplot(data=merged_data, aes(x=Weekday, y=TotalDistance, fill=Weekday))+
  geom_bar(stat="identity")
```



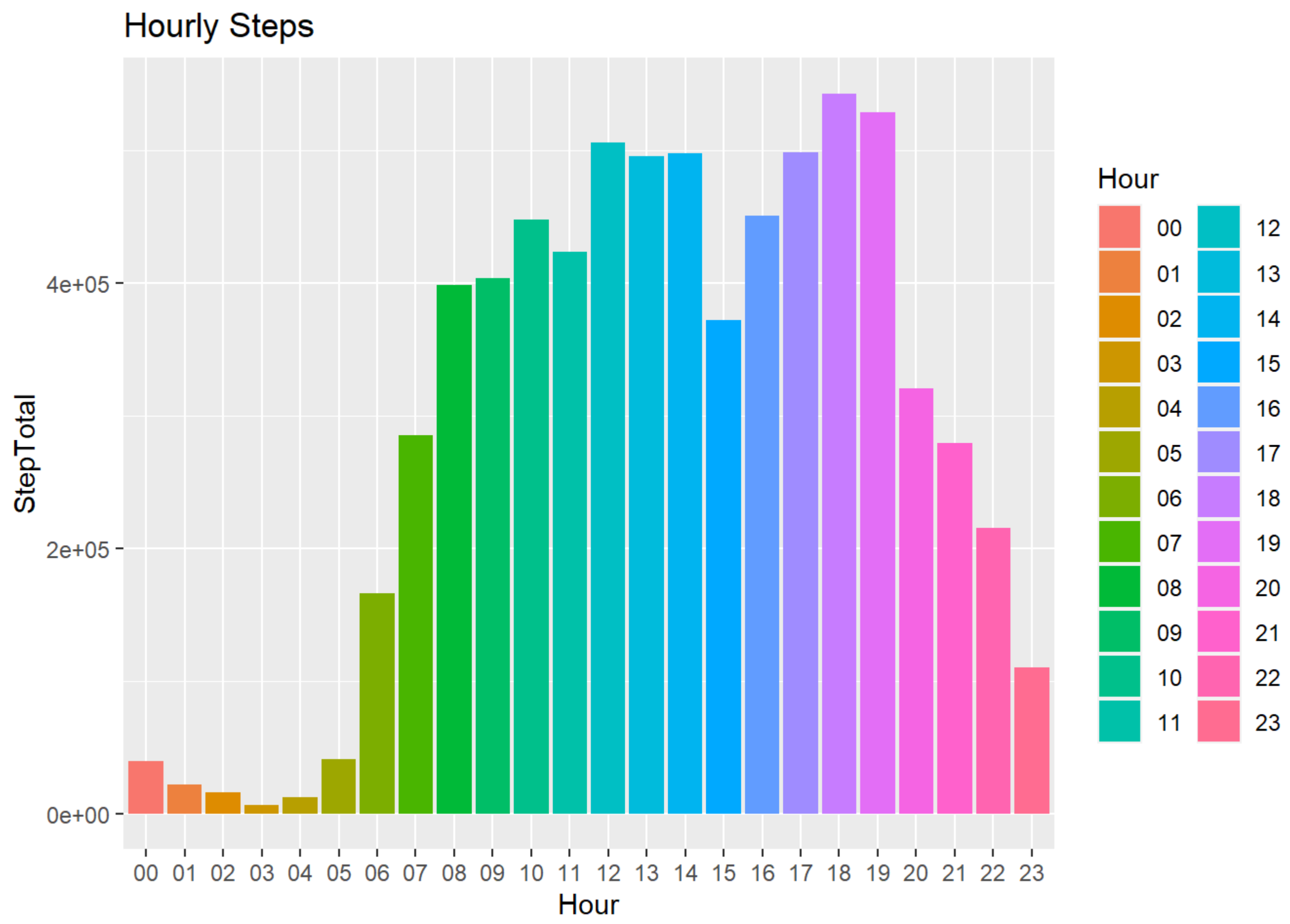
```
#Hourly
head(hourly_step)
```

```
                Id  ActivityHour                     StepTotal
             <dbl>  <chr>                                <int>
1       1503960366  4/12/2016 12:00:00 AM                  373
2       1503960366  4/12/2016 1:00:00 AM                   160
3       1503960366  4/12/2016 2:00:00 AM                   151
4       1503960366  4/12/2016 3:00:00 AM                     0
5       1503960366  4/12/2016 4:00:00 AM                     0
6       1503960366  4/12/2016 5:00:00 AM                     0
6 rows
```

```
n_distinct(hourly_step$Id) #33 users
```

```
## [1] 33
```

```
hourly_step$ActivityHour=as.POSIXct(hourly_step$ActivityHour,format="%m/%d/%Y %I:%M:%S %p")
hourly_step$Hour <-  format(hourly_step$ActivityHour,format= "%H")

ggplot(data=hourly_step, aes(x=Hour, y=StepTotal, fill=Hour))+
  geom_bar(stat="identity")+
  labs(title="Hourly Steps")
```



Step 5: Statistics summary mean, median, min, max for all 3 tables + merged data

```
daily_activity %>%
 dplyr::select(TotalSteps,
         TotalDistance,
         VeryActiveMinutes,
         FairlyActiveMinutes,
         LightlyActiveMinutes,
         SedentaryMinutes,
         Calories) %>%
  summary()
```

```
##    TotalSteps    TotalDistance    VeryActiveMinutes FairlyActiveMinutes
##  Min.   :    0   Min.   : 0.000   Min.   :  0.00    Min.   :  0.00
##  1st Qu.: 3790   1st Qu.: 2.620   1st Qu.:  0.00    1st Qu.:  0.00
##  Median : 7406   Median : 5.245   Median :  4.00    Median :  6.00
##  Mean   : 7638   Mean   : 5.490   Mean   : 21.16    Mean   : 13.56
##  3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.: 32.00    3rd Qu.: 19.00
##  Max.   :36019   Max.   :28.030   Max.   :210.00    Max.   :143.00
##  LightlyActiveMinutes SedentaryMinutes    Calories
##  Min.   :  0.0        Min.   :   0.0   Min.   :   0
##  1st Qu.:127.0        1st Qu.: 729.8   1st Qu.:1828
##  Median :199.0        Median :1057.5   Median :2134
##  Mean   :192.8        Mean   : 991.2   Mean   :2304
##  3rd Qu.:264.0        3rd Qu.:1229.5   3rd Qu.:2793
##  Max.   :518.0        Max.   :1440.0   Max.   :4900
```

```
sleep_day %>%
  dplyr::select(TotalSleepRecords,
  TotalMinutesAsleep,
  TotalTimeInBed) %>%
  summary()
```

```
##  TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##  Min.   :1.00      Min.   : 58.0      Min.   : 61.0
##  1st Qu.:1.00      1st Qu.:361.0      1st Qu.:403.8
##  Median :1.00      Median :432.5      Median :463.0
##  Mean   :1.12      Mean   :419.2      Mean   :458.5
##  3rd Qu.:1.00      3rd Qu.:490.0      3rd Qu.:526.0
##  Max.   :3.00      Max.   :796.0      Max.   :961.0
```

```
weight %>%
  dplyr::select(WeightPounds, BMI) %>%
  summary()
```

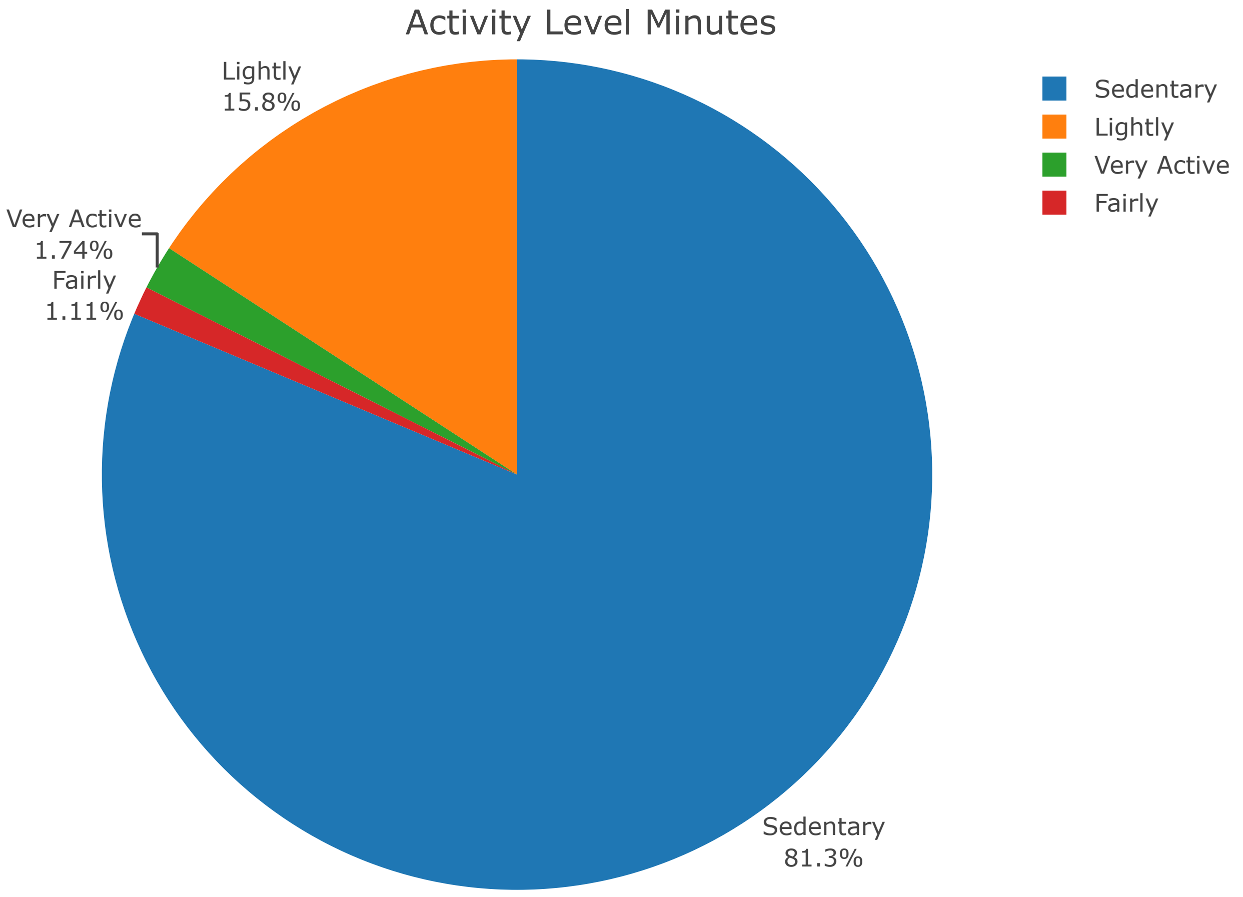
```
##   WeightPounds        BMI
##  Min.   :116.0   Min.   :21.45
##  1st Qu.:135.4   1st Qu.:23.96
##  Median :137.8   Median :24.39
##  Mean   :158.8   Mean   :25.19
##  3rd Qu.:187.5   3rd Qu.:25.56
##  Max.   :294.3   Max.   :47.54
```

```
#Optional for merged data
merged_data %>%
  dplyr::select(Weekday,
         TotalSteps,
         TotalDistance,
         VeryActiveMinutes,
         FairlyActiveMinutes,
         LightlyActiveMinutes,
         SedentaryMinutes,
         Calories,
         TotalMinutesAsleep,
         TotalTimeInBed,
         WeightPounds,
         BMI
         ) %>%
  summary()
```

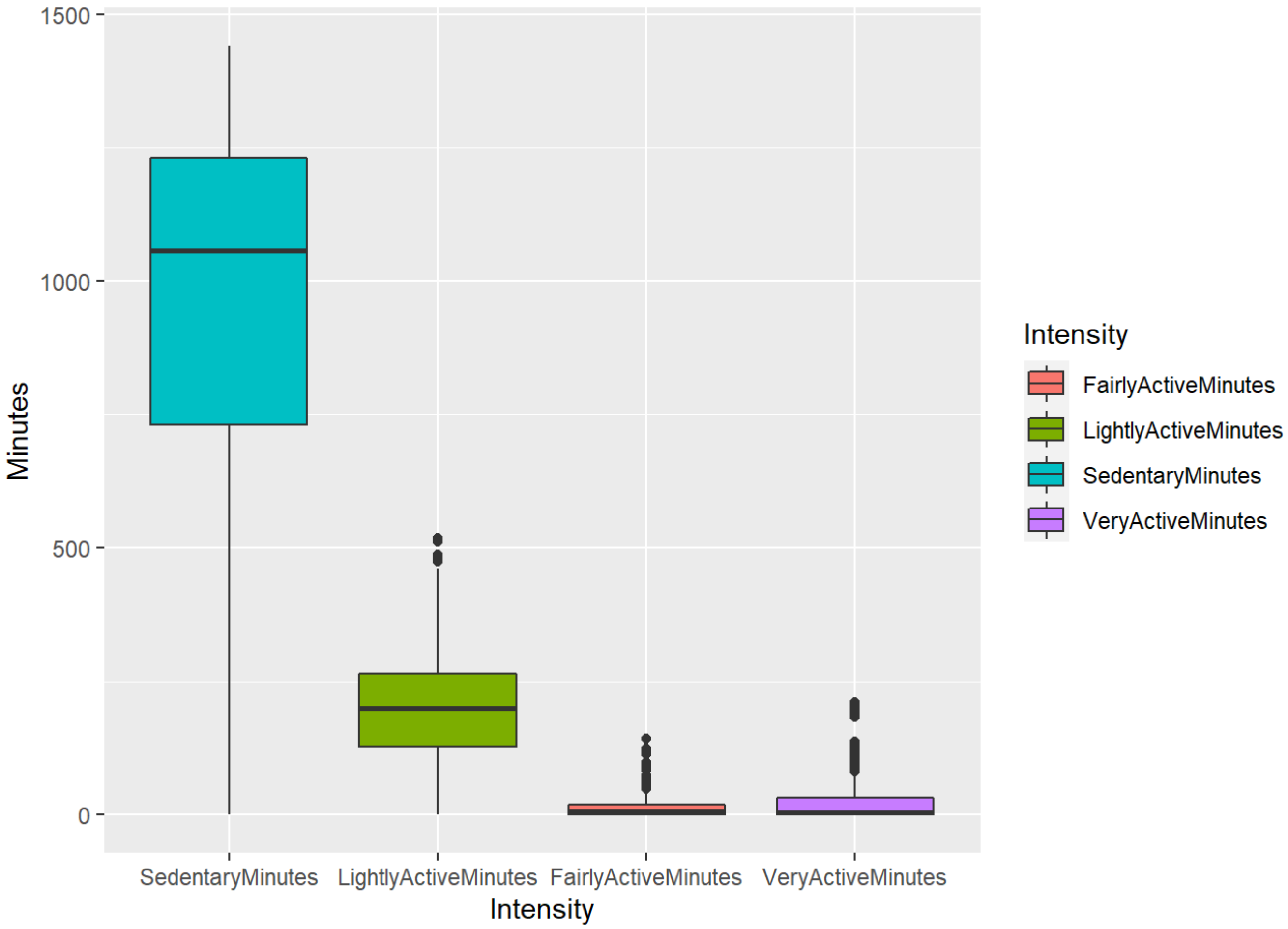
```
##       Weekday       TotalSteps    TotalDistance    VeryActiveMinutes
##  Monday   :5609   Min.   :    0   Min.   : 0.000   Min.   :  0.00
##  Tuesday  :7004   1st Qu.: 5832   1st Qu.: 3.910   1st Qu.:  0.00
##  Wednesday:6988   Median :10199   Median : 6.820   Median : 15.00
##  Thursday :6930   Mean   : 9373   Mean   : 6.415   Mean   : 23.57
##  Friday   :5632   3rd Qu.:12109   3rd Qu.: 8.350   3rd Qu.: 38.00
##  Saturday :5616   Max.   :36019   Max.   :28.030   Max.   :210.00
##  Sunday   :5610
##  FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes    Calories
##  Min.   :  0.00      Min.   :  0.0        Min.   :   0.0   Min.   :   0
##  1st Qu.:  3.00      1st Qu.:194.0        1st Qu.: 637.0   1st Qu.:1850
##  Median : 14.00      Median :238.0        Median : 697.0   Median :2046
##  Mean   : 17.82      Mean   :232.2        Mean   : 722.6   Mean   :2103
##  3rd Qu.: 31.00      3rd Qu.:288.0        3rd Qu.: 745.0   3rd Qu.:2182
##  Max.   :143.00      Max.   :518.0        Max.   :1440.0   Max.   :4900
##
##  TotalMinutesAsleep TotalTimeInBed   WeightPounds        BMI
##  Min.   : 58.0      Min.   : 61.0   Min.   :116.0   Min.   :21.45
##  1st Qu.:400.0      1st Qu.:421.0   1st Qu.:134.9   1st Qu.:23.89
##  Median :442.0      Median :457.0   Median :135.6   Median :24.00
##  Mean   :433.8      Mean   :458.2   Mean   :139.6   Mean   :24.42
##  3rd Qu.:477.0      3rd Qu.:510.0   3rd Qu.:136.7   3rd Qu.:24.21
##  Max.   :796.0      Max.   :961.0   Max.   :294.3   Max.   :47.54
##  NA's   :971        NA's   :971     NA's   :8881    NA's   :8881
```

Step 6: analysis on active minutes, calorie, total steps. The American Heart Association and World Health Organization recommend at least 150 minutes of moderate-intensity activity or 75 minutes of vigorous activity, or a combination of both, each week. That means it needs an daily goal of 21.4 minutes of FairlyActiveMinutes or 10.7 minutes of VeryActiveMinutes

```
#Active users
active_users <- daily_activity %>%
  filter(FairlyActiveMinutes >= 21.4 | VeryActiveMinutes>=10.7) %>%
  group_by(Id) %>%
  count(Id)

total_minutes <- sum(daily_activity$SedentaryMinutes, daily_activity$VeryActiveMinutes, daily_ac
tivity$FairlyActiveMinutes, daily_activity$LightlyActiveMinutes)
sedentary_percentage <- sum(daily_activity$SedentaryMinutes)/total_minutes*100
lightly_percentage <- sum(daily_activity$LightlyActiveMinutes)/total_minutes*100
fairly_percentage <- sum(daily_activity$FairlyActiveMinutes)/total_minutes*100
active_percentage <- sum(daily_activity$VeryActiveMinutes)/total_minutes*100

#Pie charts
percentage <- data.frame(
  level=c("Sedentary", "Lightly", "Fairly", "Very Active"),
  minutes=c(sedentary_percentage,lightly_percentage,fairly_percentage,active_percentage)
)


plot_ly(percentage, labels = ~level, values = ~minutes, type = 'pie',textposition = 'outside',te
xtinfo = 'label+percent') %>%
  layout(title = 'Activity Level Minutes',
         xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
         yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))
```

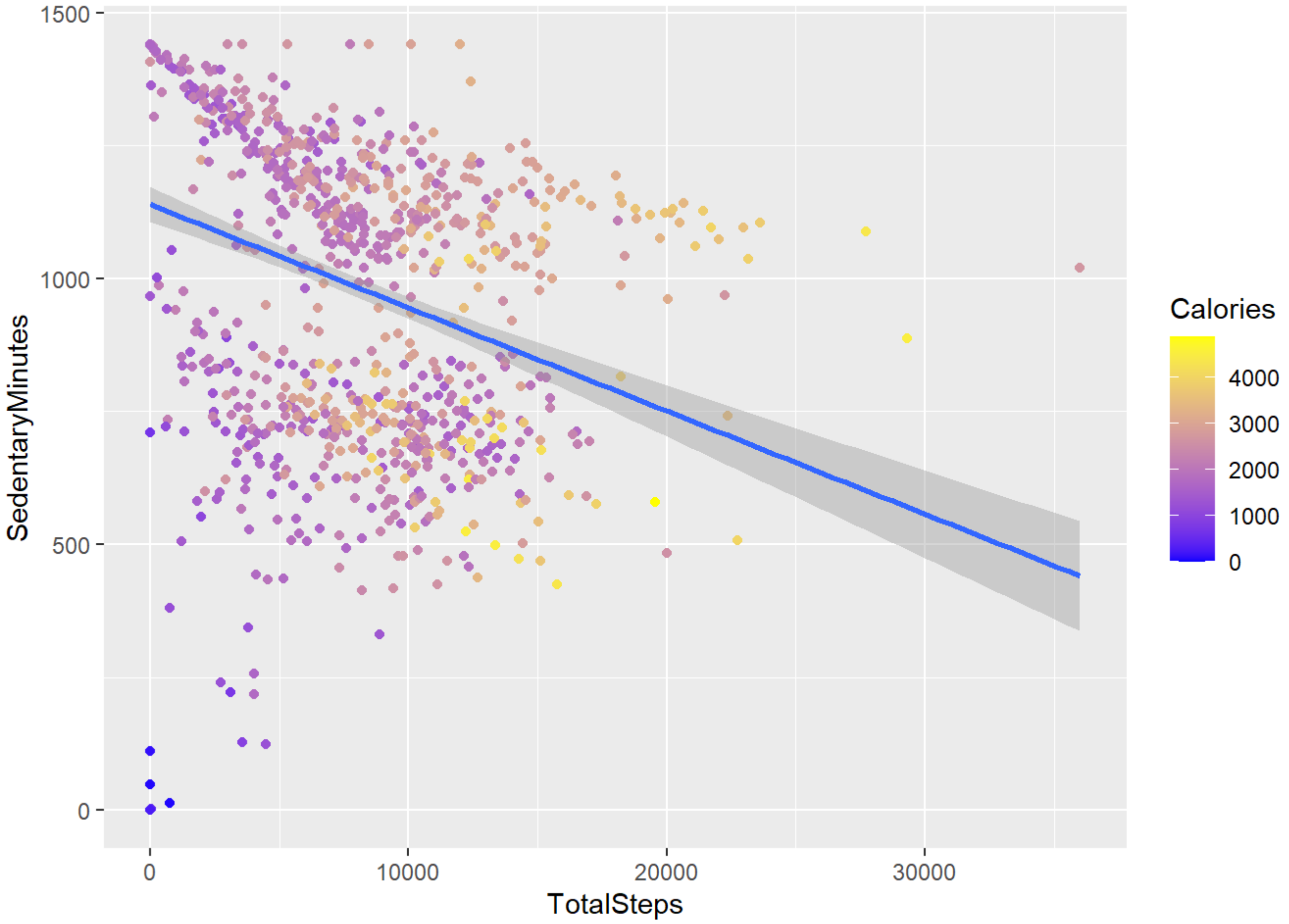


```
#How active are the users
active_minute <- daily_activity %>%
  gather(key=Intensity, value=active_minutes, ends_with("minutes")) %>%
  select(Intensity, active_minutes)


ggplot(data=active_minute, aes(x=Intensity, y=active_minutes))+
  geom_boxplot(aes(fill=Intensity))+
  scale_x_discrete(limits=c("SedentaryMinutes","LightlyActiveMinutes","FairlyActiveMinutes","VeryActiveMinutes"))+
  ylab("Minutes")
```



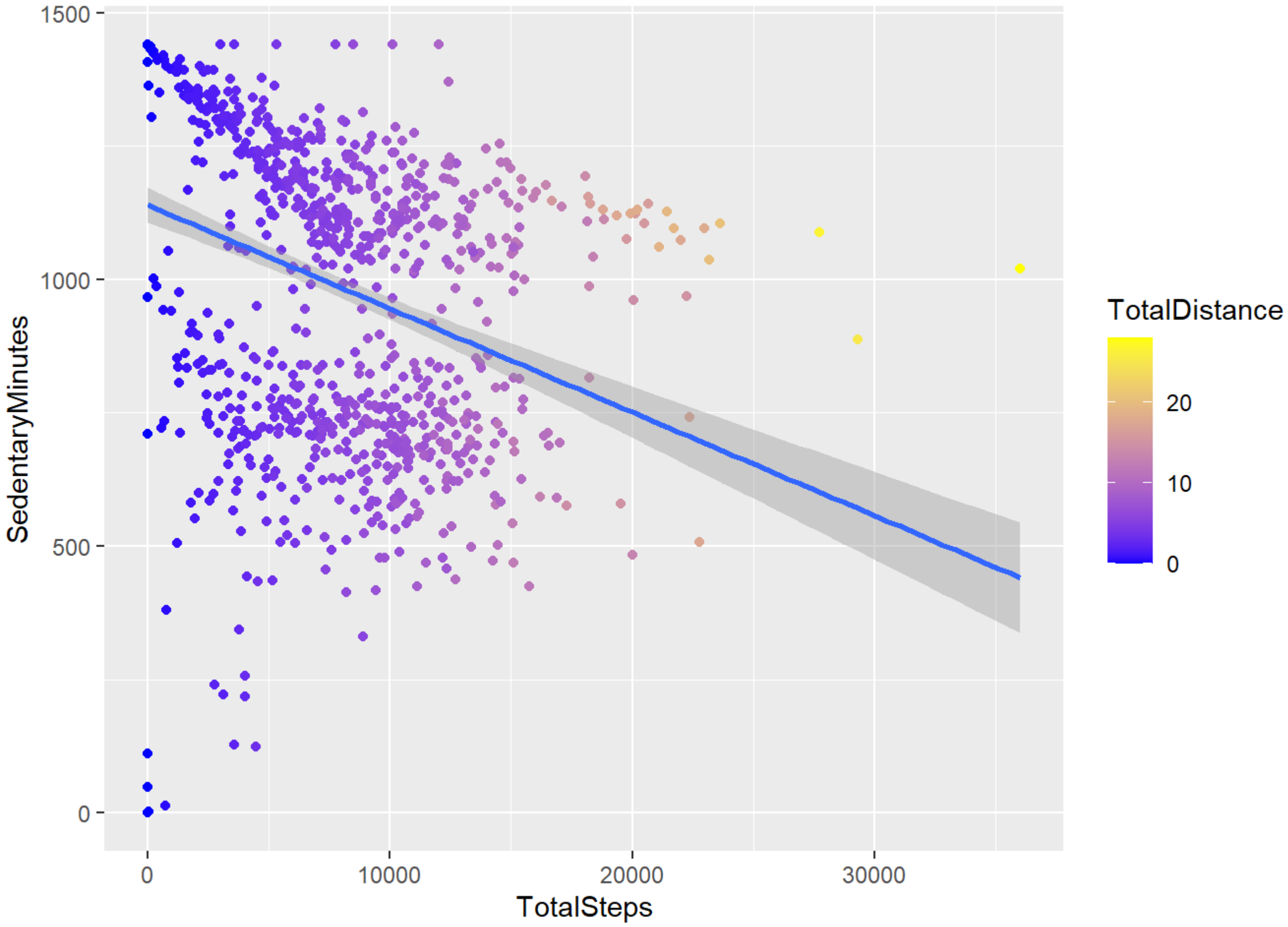
```
# Total steps vs Sedentary Minutes with Calories and Total Distance. The two plots are very similar.
# Users who are more active burn more calories. Users who are sedentary take the less steps and burn less calories.
par(mfrow = c(2, 2))
ggplot(data=daily_activity, aes(x=TotalSteps, y=SedentaryMinutes, color=Calories))+
  geom_point()+
  stat_smooth(method=lm)+
  scale_color_gradient(low="blue", high="yellow")
```

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



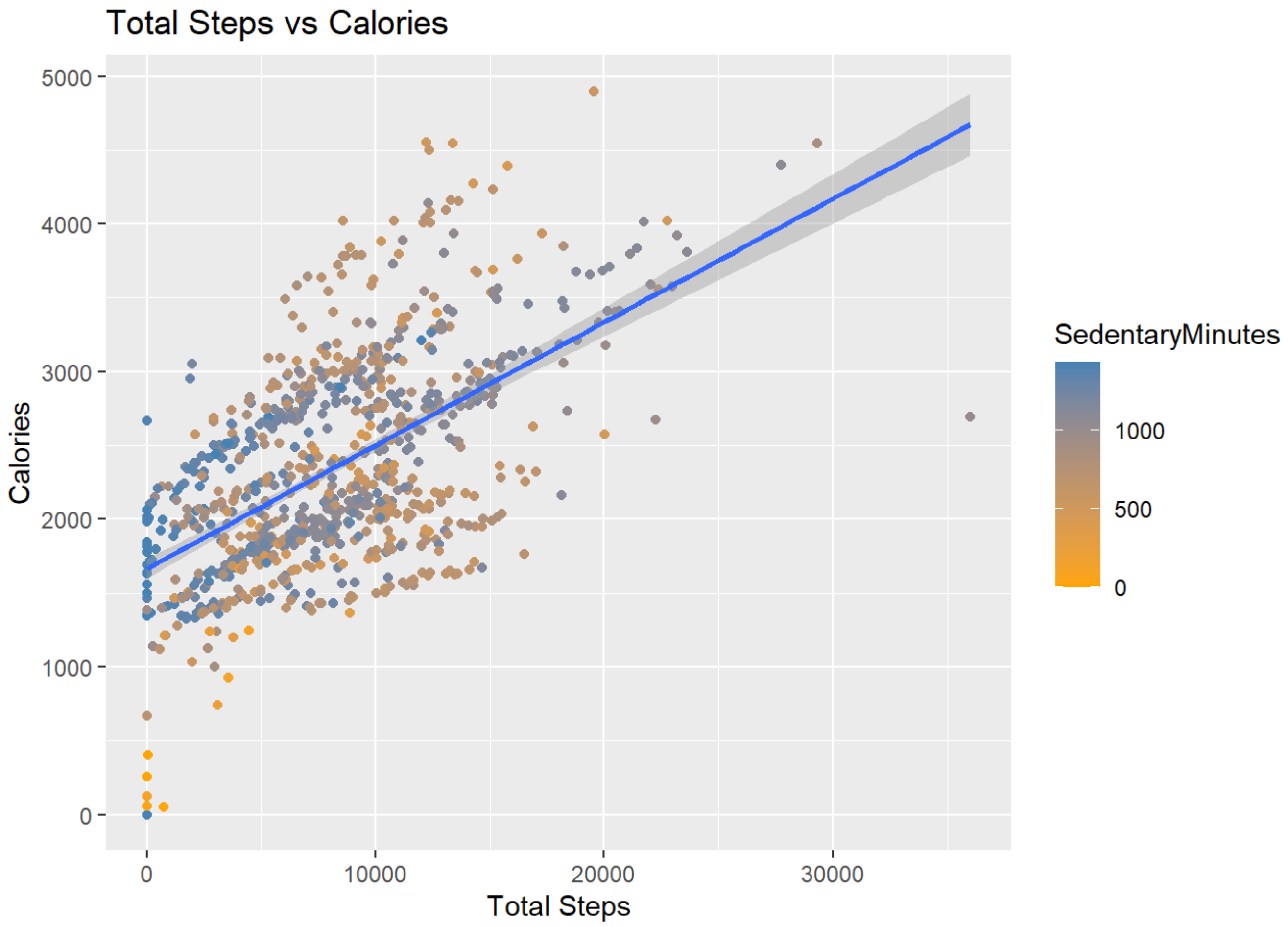
```
ggplot(data=daily_activity, aes(x=TotalSteps, y=SedentaryMinutes, color=TotalDistance))+
  geom_point()+
  stat_smooth(method=lm)+
  scale_color_gradient(low="blue", high="yellow")
```

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



```
# Interesting find here that some user who are sedentary, takes minimal step, but still able to
 burn over 1500 to 2500 calories
ggplot(data=daily_activity, aes(x=TotalSteps, y = Calories, color=SedentaryMinutes))+
  geom_point()+
  labs(title="Total Steps vs Calories")+
  xlab("Total Steps")+
  stat_smooth(method=lm)+
  scale_color_gradient(low="orange", high="steelblue")
```

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

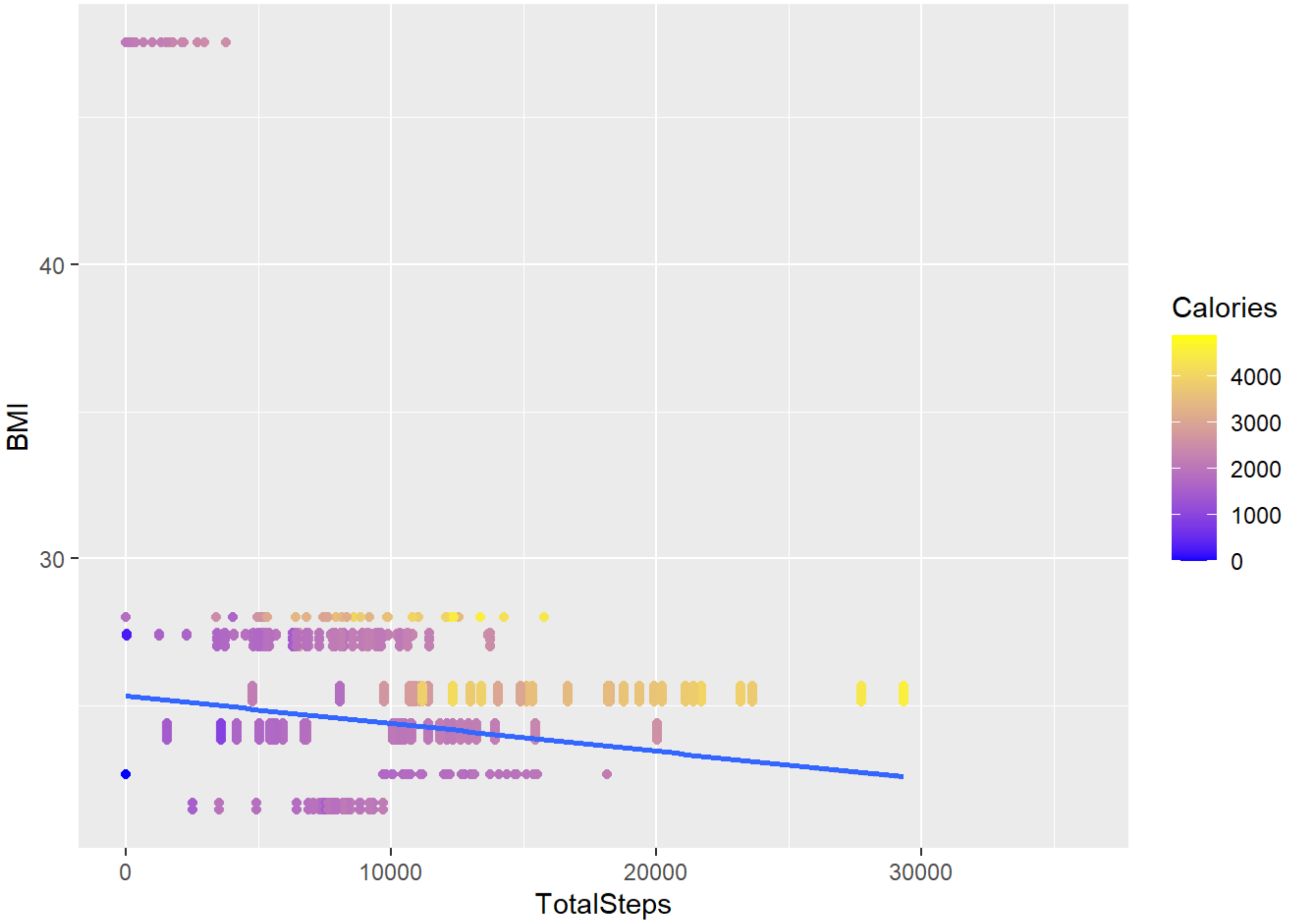


```
# Users who take more steps, burn more calories and has lower BMI. We also see some outliers in
 the top left corner.
ggplot(data=merged_data, aes(x=TotalSteps, y = BMI, color=Calories))+
  geom_point()+
  stat_smooth(method=lm)+
   scale_color_gradient(low="blue", high="yellow")
```

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

```
## Warning: Removed 8881 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 8881 rows containing missing values (geom_point).
```



Step 7: Regression analysis and R value, leverage points (lm.influence)

```
#With lm() analysis, we want to look at the R-squared. 0% indicates that the model explains none of the variability of the response data around its mean. 100% indicates that the model explains all the variability of the response data around its mean.


step_vs_sedentary.mod <- lm(SedentaryMinutes ~ TotalSteps, data = merged_data)
summary(step_vs_sedentary.mod)
```

```
##
## Call:
## lm(formula = SedentaryMinutes ~ TotalSteps, data = merged_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -811.33  -63.62  -37.76   41.37  742.49
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.115e+02  2.354e+00  344.79   <2e-16 ***
## TotalSteps  -9.486e-03  2.287e-04  -41.48   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 202.5 on 43387 degrees of freedom
## Multiple R-squared:  0.03815,    Adjusted R-squared:  0.03813
## F-statistic:  1721 on 1 and 43387 DF,  p-value: < 2.2e-16
```

```
bmi_vs_steps.mod <- lm(BMI ~ TotalSteps, data = merged_data)
summary(bmi_vs_steps.mod)
```

```
##
## Call:
## lm(formula = BMI ~ TotalSteps, data = merged_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6517 -0.7069 -0.3289 -0.0292 22.5574
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.534e+01  2.611e-02  970.45   <2e-16 ***
## TotalSteps  -9.404e-05  2.463e-06  -38.19   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.862 on 34506 degrees of freedom
##   (8881 observations deleted due to missingness)
## Multiple R-squared:  0.04055,    Adjusted R-squared:  0.04052
## F-statistic:  1458 on 1 and 34506 DF,  p-value: < 2.2e-16
```

```
calories_vs_steps.mod <- lm(Calories ~ TotalSteps, data = merged_data)
summary(calories_vs_steps.mod)
```

```
##
## Call:
## lm(formula = Calories ~ TotalSteps, data = merged_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1478.95  -176.96  -116.26    14.13  2258.40
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.479e+03  5.293e+00   279.4   <2e-16 ***
## TotalSteps  6.661e-02  5.143e-04   129.5   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 455.5 on 43387 degrees of freedom
## Multiple R-squared:  0.2788, Adjusted R-squared:  0.2788
## F-statistic: 1.677e+04 on 1 and 43387 DF,  p-value: < 2.2e-16
```

```
sedentary_vs_sleep.mod <- lm(SedentaryMinutes ~ TotalMinutesAsleep, data = merged_data)
summary(sedentary_vs_sleep.mod)
```

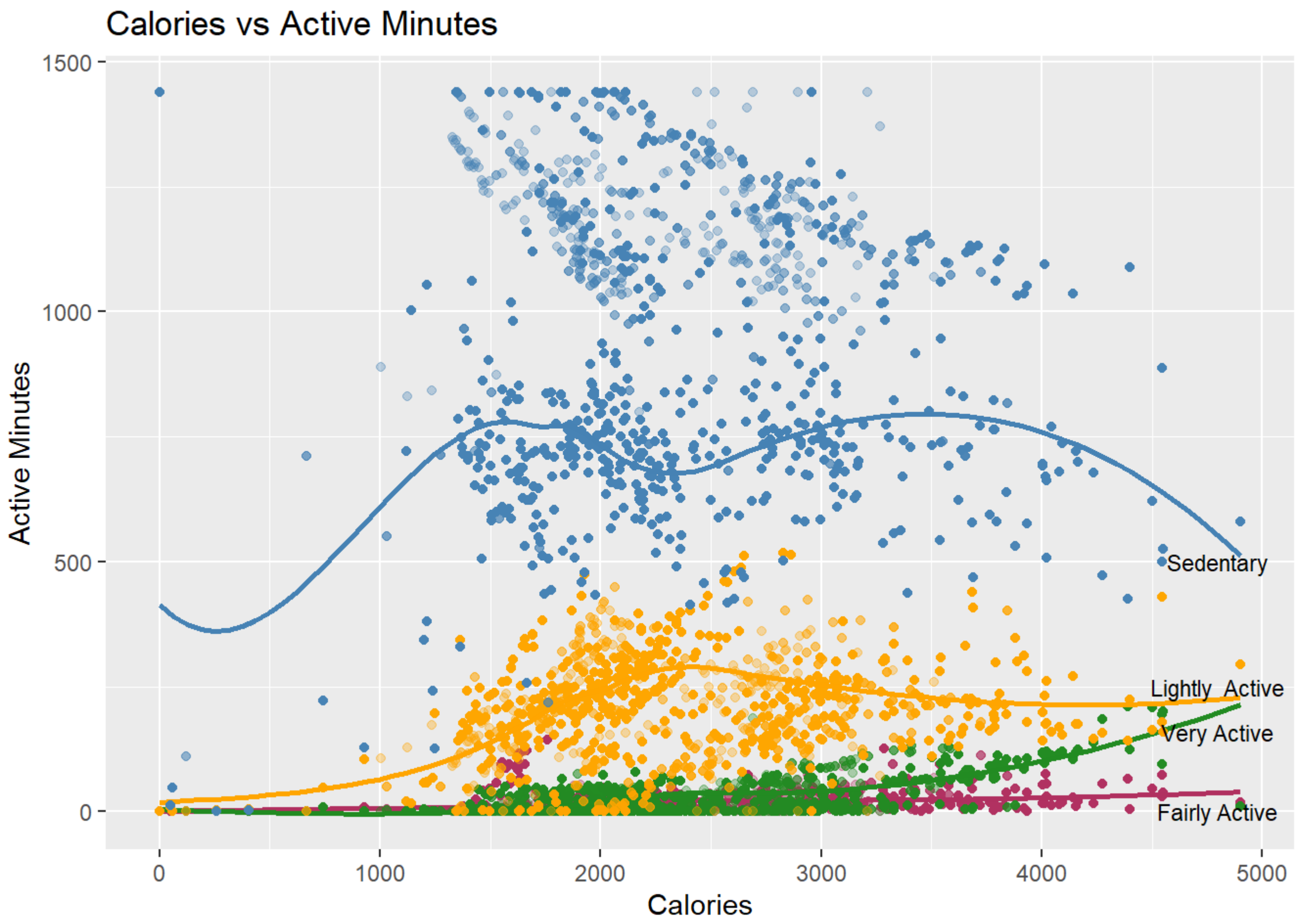
```
##
## Call:
## lm(formula = SedentaryMinutes ~ TotalMinutesAsleep, data = merged_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -878.84  -76.54  -17.80   42.03  866.28
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        904.88714    4.48547  201.74   <2e-16 ***
## TotalMinutesAsleep  -0.44156    0.01011  -43.69   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 194.4 on 42416 degrees of freedom
##   (971 observations deleted due to missingness)
## Multiple R-squared:  0.04306,    Adjusted R-squared:  0.04304
## F-statistic:  1909 on 1 and 42416 DF,  p-value: < 2.2e-16
```

```
veryactive_vs_sleep.mod <- lm(VeryActiveMinutes ~ TotalMinutesAsleep, data = merged_data)
summary(veryactive_vs_sleep.mod)
```

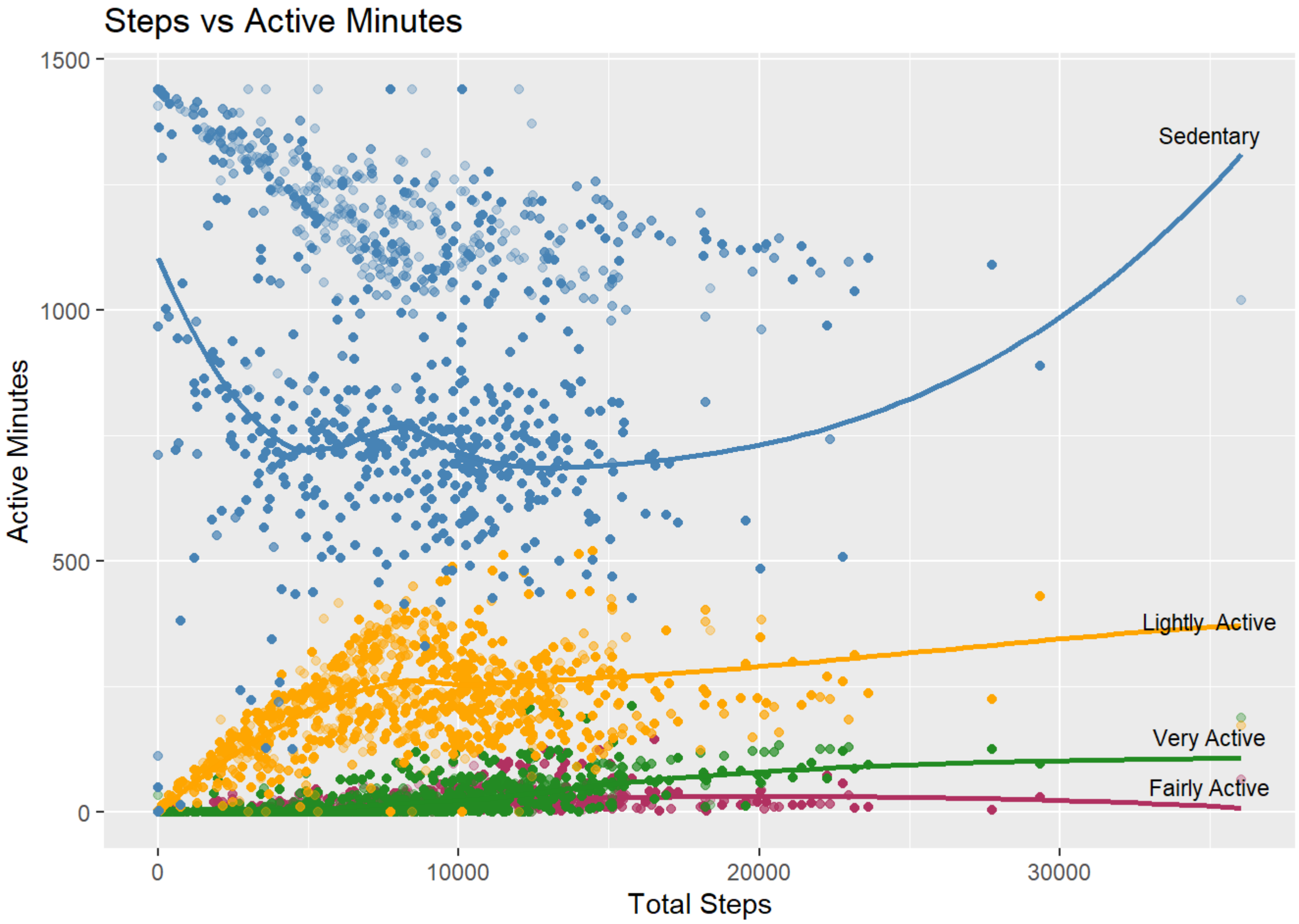
```
## 
## Call:
## lm(formula = VeryActiveMinutes ~ TotalMinutesAsleep, data = merged_data)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -23.500 -22.737  -7.984  14.862 187.401 
## 
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)    
## (Intercept)        23.595768   0.582829  40.485   <2e-16 ***
## TotalMinutesAsleep -0.001652   0.001313  -1.258    0.208    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 25.26 on 42416 degrees of freedom
##   (971 observations deleted due to missingness)
## Multiple R-squared:  3.732e-05,  Adjusted R-squared:  1.374e-05 
## F-statistic: 1.583 on 1 and 42416 DF,  p-value: 0.2084
```

Step 8: This high volume of moderate-to-vigorous physical activity is achieved by a very small proportion of the population. Let’s take a look at this.

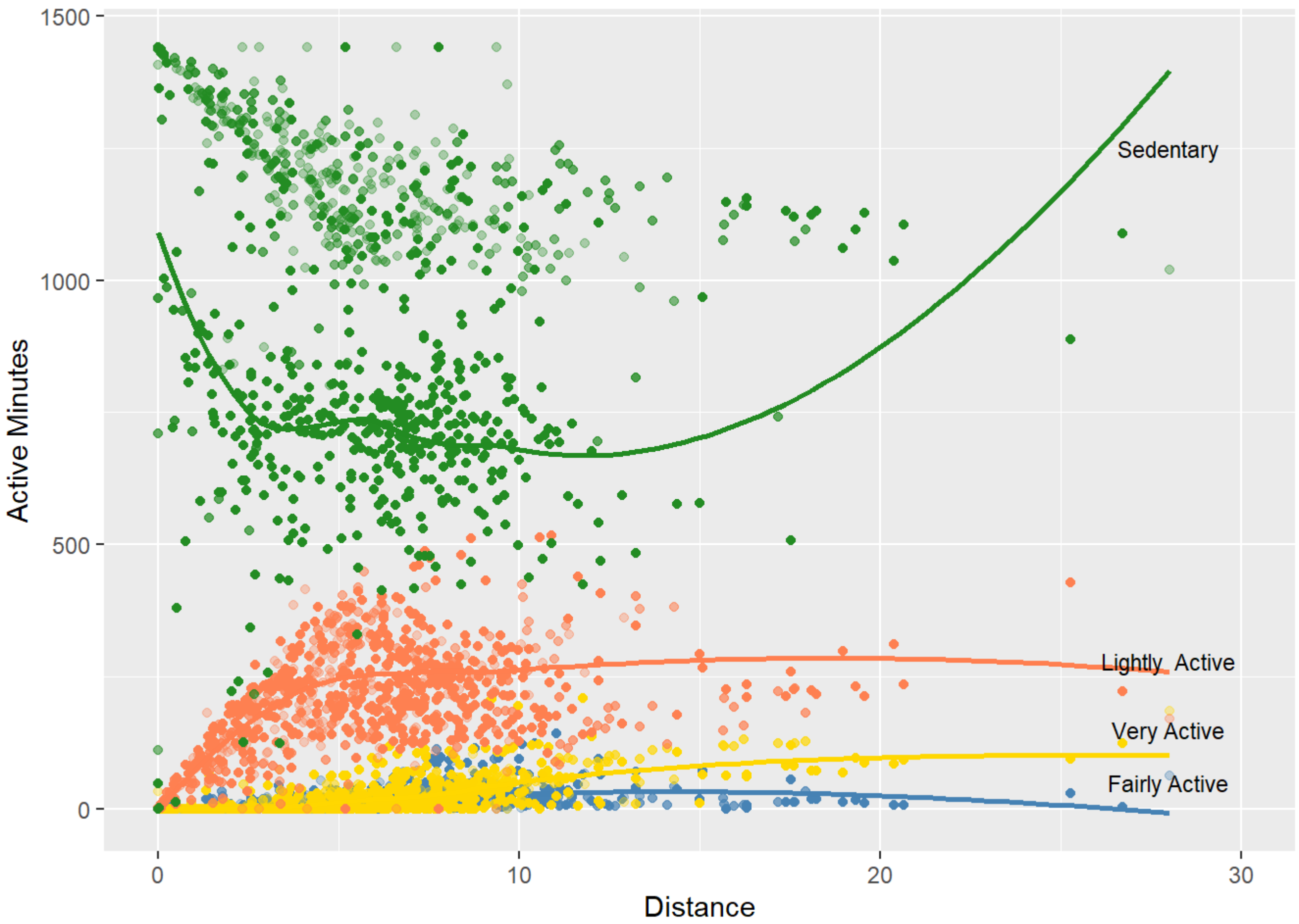
```
active_minutes_vs_calories <- ggplot(data = merged_data) +
  geom_point(mapping=aes(x=Calories, y=FairlyActiveMinutes), color = "maroon", alpha = 1/3) +
  geom_smooth(method = loess,formula =y ~ x, mapping=aes(x=Calories, y=FairlyActiveMinutes, colo
r=FairlyActiveMinutes), color = "maroon", se = FALSE) +
  
  geom_point(mapping=aes(x=Calories, y=VeryActiveMinutes), color = "forestgreen", alpha = 1/3) +
  geom_smooth(method = loess,formula =y ~ x,mapping=aes(x=Calories, y=VeryActiveMinutes, color=V
eryActiveMinutes), color = "forestgreen", se = FALSE) +
  
  geom_point(mapping=aes(x=Calories, y=LightlyActiveMinutes), color = "orange", alpha = 1/3) +
  geom_smooth(method = loess,formula =y ~ x,mapping=aes(x=Calories, y=LightlyActiveMinutes, colo
r=LightlyActiveMinutes), color = "orange", se = FALSE) +
  
  geom_point(mapping=aes(x=Calories, y=SedentaryMinutes), color = "steelblue", alpha = 1/3) +
  geom_smooth(method = loess,formula =y ~ x,mapping=aes(x=Calories, y=SedentaryMinutes, color=Se
dentaryeMinutes), color = "steelblue", se = FALSE) +
  
  annotate("text", x=4800, y=160, label="Very Active", color="black", size=3)+
  annotate("text", x=4800, y=0, label="Fairly Active", color="black", size=3)+
  annotate("text", x=4800, y=500, label="Sedentary", color="black", size=3)+
  annotate("text", x=4800, y=250, label="Lightly  Active", color="black", size=3)+
  labs(x = "Calories", y = "Active Minutes", title="Calories vs Active Minutes")

active_minutes_vs_calories
```



```
active_minutes_vs_steps <- ggplot(data = merged_data) +
  geom_point(mapping=aes(x=TotalSteps, y=FairlyActiveMinutes), color = "maroon", alpha = 1/3) +
  geom_smooth(method = loess,formula =y ~ x, mapping=aes(x=TotalSteps, y=FairlyActiveMinutes, co
lor=FairlyActiveMinutes), color = "maroon", se = FALSE) +

  geom_point(mapping=aes(x=TotalSteps, y=VeryActiveMinutes), color = "forestgreen", alpha = 1/3)
+
  geom_smooth(method = loess,formula =y ~ x,mapping=aes(x=TotalSteps, y=VeryActiveMinutes, color
=VeryActiveMinutes), color = "forestgreen", se = FALSE) +

  geom_point(mapping=aes(x=TotalSteps, y=LightlyActiveMinutes), color = "orange", alpha = 1/3) +
  geom_smooth(method = loess,formula =y ~ x,mapping=aes(x=TotalSteps, y=LightlyActiveMinutes, co
lor=LightlyActiveMinutes), color = "orange", se = FALSE) +

   geom_point(mapping=aes(x=TotalSteps, y=SedentaryMinutes), color = "steelblue", alpha = 1/3) +
  geom_smooth(method = loess,formula =y ~ x,mapping=aes(x=TotalSteps, y=SedentaryMinutes, color=
SedentaryMinutes), color = "steelblue", se = FALSE) +

  annotate("text", x=35000, y=150, label="Very Active", color="black", size=3)+
  annotate("text", x=35000, y=50, label="Fairly Active", color="black", size=3)+
  annotate("text", x=35000, y=1350, label="Sedentary", color="black", size=3)+
  annotate("text", x=35000, y=380, label="Lightly  Active", color="black", size=3)+
  labs(x = "Total Steps", y = "Active Minutes", title="Steps vs Active Minutes")

active_minutes_vs_steps
```


Steps vs Active Minutes
Active Minutes
1500
1000
500
0
Total Steps
0
10000
20000
30000
Sedentary
Lightly Active
Very Active
Fairly Active

```
active_minutes_vs_distance <- ggplot(data = merged_data) +
  geom_point(mapping=aes(x=TotalDistance, y=FairlyActiveMinutes), color = "steelblue", alpha = 1
/3) +
  geom_smooth(method = loess,formula =y ~ x, mapping=aes(x=TotalDistance, y=FairlyActiveMinutes,
color=FairlyActiveMinutes), color = "steelblue", se = FALSE) +

  geom_point(mapping=aes(x=TotalDistance, y=VeryActiveMinutes), color = "gold", alpha = 1/3) +
  geom_smooth(method = loess,formula =y ~ x,mapping=aes(x=TotalDistance, y=VeryActiveMinutes, co
lor=VeryActiveMinutes), color = "gold", se = FALSE) +

  geom_point(mapping=aes(x=TotalDistance, y=LightlyActiveMinutes), color = "coral", alpha = 1/3)
+
  geom_smooth(method = loess,formula =y ~ x,mapping=aes(x=TotalDistance, y=LightlyActiveMinutes,
color=LightlyActiveMinutes), color = "coral", se = FALSE) +

   geom_point(mapping=aes(x=TotalDistance, y=SedentaryMinutes), color = "forestgreen", alpha = 1
/3) +
  geom_smooth(method = loess,formula =y ~ x,mapping=aes(x=TotalDistance, y=SedentaryMinutes, col
or=SedentaryMinutes), color = "forestgreen", se = FALSE) +

  scale_x_continuous(limits = c(0, 30))+

  annotate("text", x=28, y=150, label="Very Active", color="black", size=3)+
  annotate("text", x=28, y=50, label="Fairly Active", color="black", size=3)+
  annotate("text", x=28, y=1250, label="Sedentary", color="black", size=3)+
  annotate("text", x=28, y=280, label="Lightly  Active", color="black", size=3)+
  labs(x = "Distance", y = "Active Minutes")

active_minutes_vs_distance
```



Step 9: Analysis on sleep

```
#Sleep time in hours instead of minutes
sleep_day_in_hour <-sleep_day
sleep_day_in_hour$TotalMinutesAsleep <- sleep_day_in_hour$TotalMinutesAsleep/60
sleep_day_in_hour$TotalTimeInBed <- sleep_day_in_hour$TotalTimeInBed/60
head(sleep_day_in_hour)
```

| | Id | SleepDay | TotalSleepRecords | TotalMinutesAsleep | TotalTimeIn |
|---|---|---|---|---|---|
| | <dbl> | <chr> | <int> | <dbl> | < |
| 1 | 1503960366 | 4/12/2016 12:00:00 AM | 1 | 5.450000 | 5.766 |
| 2 | 1503960366 | 4/13/2016 12:00:00 AM | 2 | 6.400000 | 6.783 |
| 3 | 1503960366 | 4/15/2016 12:00:00 AM | 1 | 6.866667 | 7.366 |
| 4 | 1503960366 | 4/16/2016 12:00:00 AM | 2 | 5.666667 | 6.116 |
| 5 | 1503960366 | 4/17/2016 12:00:00 AM | 1 | 11.666667 | 11.866 |
| 6 | 1503960366 | 4/19/2016 12:00:00 AM | 1 | 5.066667 | 5.333 |

6 rows

```
#Check for any sleep outliers. # of times user sleep more than 10 hours or less than 1
sum(sleep_day_in_hour$TotalMinutesAsleep > 9)
```

```
## [1] 39
```

```
sum(sleep_day_in_hour$TotalTimeInBed > 9)
```

```
## [1] 87
```

```
sum(sleep_day_in_hour$TotalMinutesAsleep < 2)
```

```
## [1] 15
```

```
sum(sleep_day_in_hour$TotalTimeInBed < 2)
```

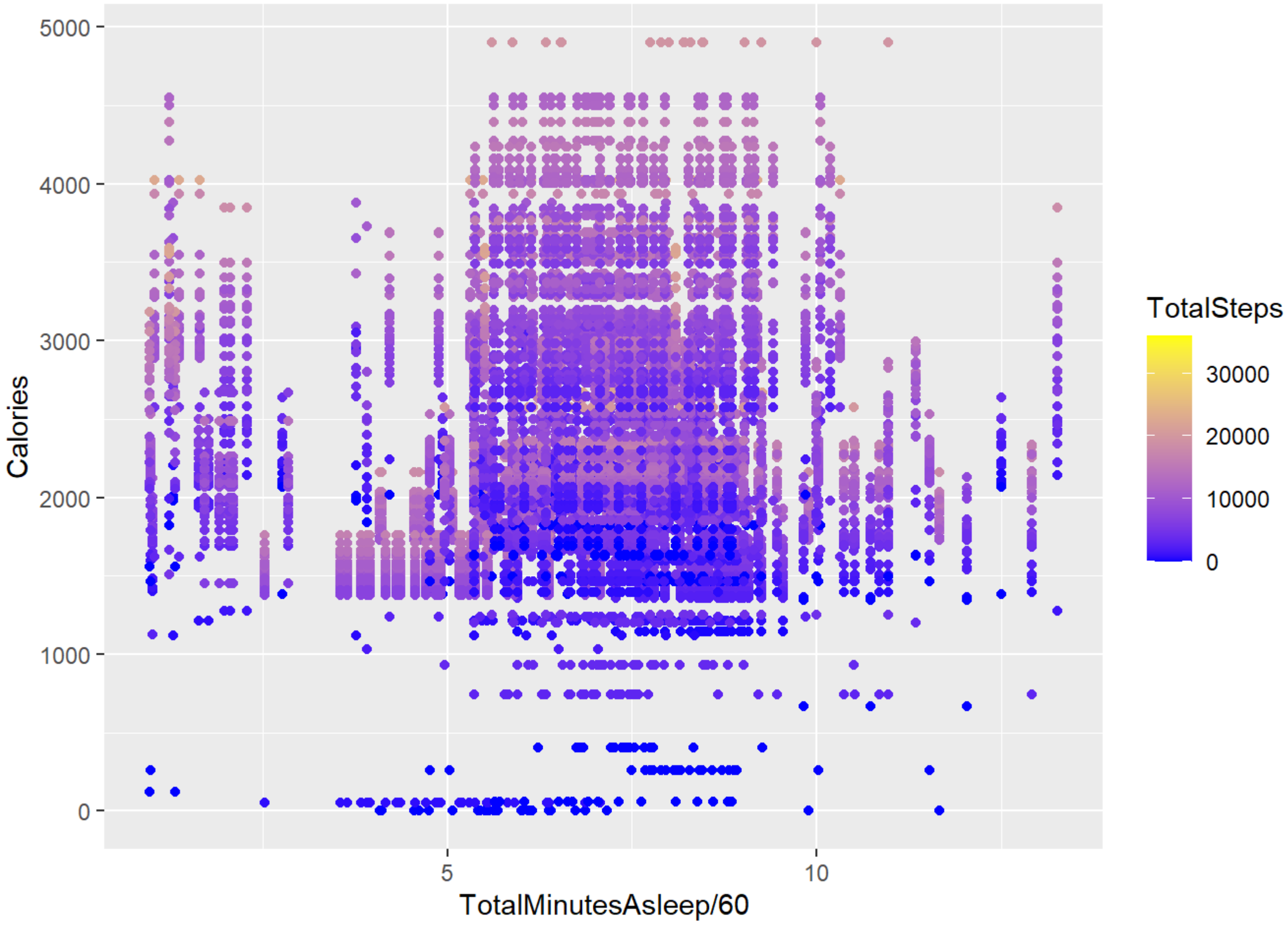
```
## [1] 12
```

```
#According to article: https://blog.fitbit.com/sleep-study/#:~:text=The%20average%20Fitbit%20use
r%20is,is%20spent%20restless%20or%20awake.&text=People%20who%20sleep%205%20hours,the%20beginnin
g%20of%20the%20night. 55 minutes are spend awake in bed before going to sleep. Let see how many
 users in our study is according to the FitBit data

awake_in_bed <- mutate(sleep_day, AwakeTime = TotalTimeInBed - TotalMinutesAsleep)
awake_in_bed <- awake_in_bed %>%
  filter(AwakeTime >= 55) %>%
  group_by(Id) %>%
  arrange(AwakeTime, desc=TRUE)

n_distinct(awake_in_bed$Id) #13 users spend more than 55 minutes in bed before falling alseep
```

```
## [1] 13
```

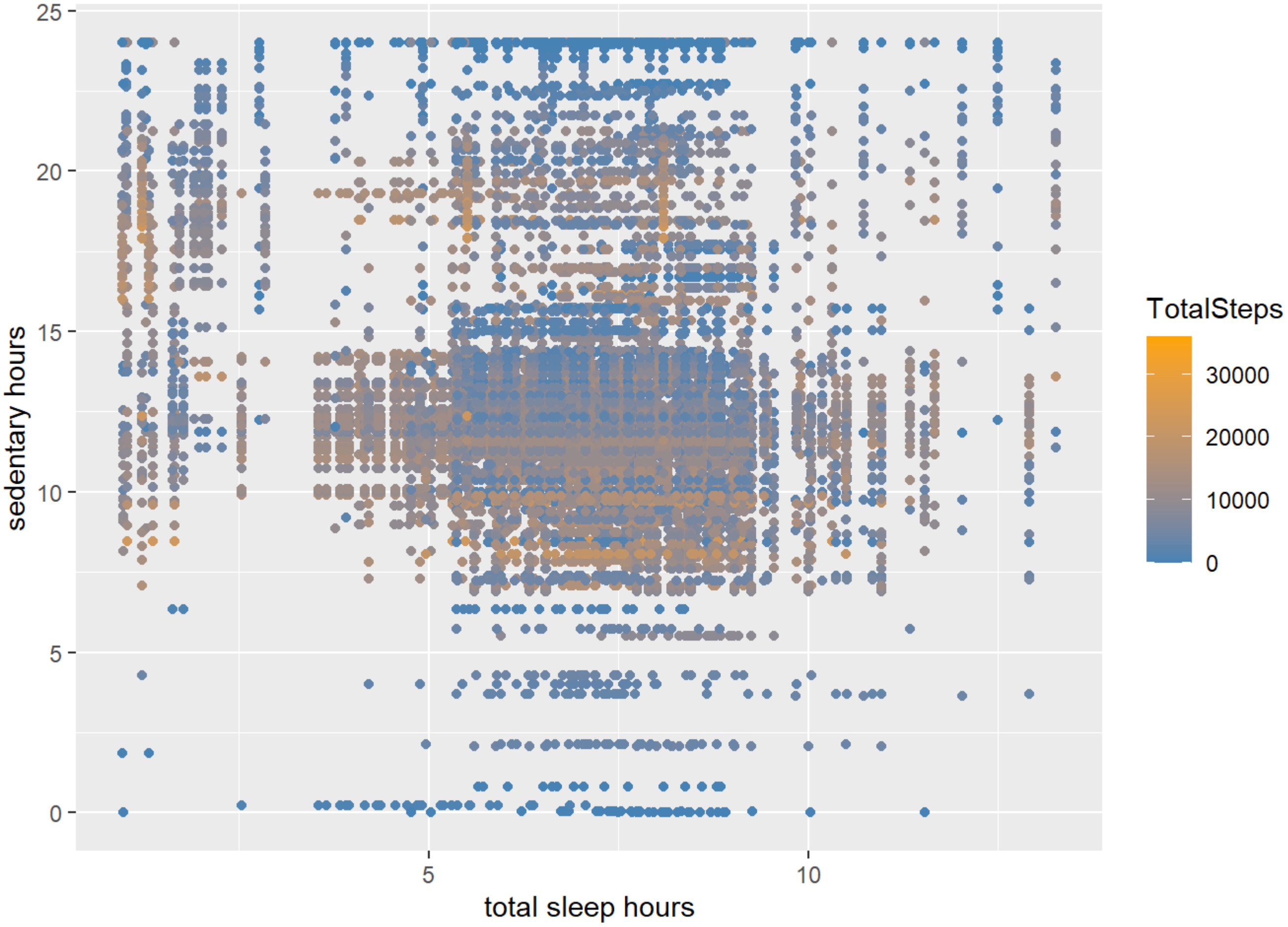
```
#How many minutes an user sleep may not correlate well with how actively they are, but sedentary
time account for about 80% of during the day

# Majority of the users sleep between 5 to 10 hours burns around 1500 to 4500 calories a day.
ggplot(data=merged_data, aes(x=TotalMinutesAsleep/60, y=Calories, color=TotalSteps))+
  geom_point()+
  scale_color_gradient(low="blue", high="yellow")
```

```
## Warning: Removed 971 rows containing missing values (geom_point).
```



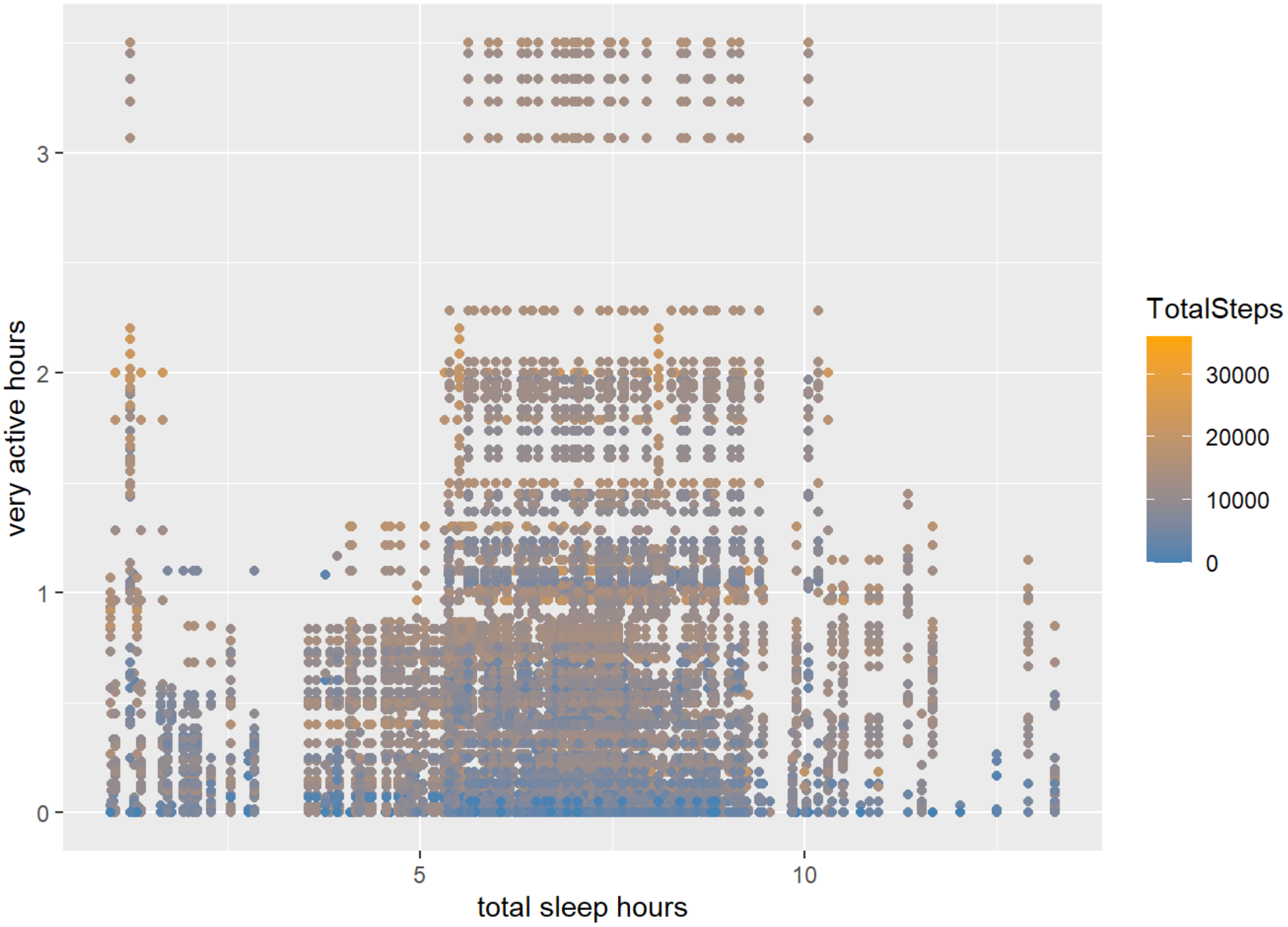
```
# Majority of the users sleep between 5 to 10 hours spend 7 to 24 hours in sedentary and only 0
 to 2 hours in very active mode.
ggplot(data=merged_data, aes(x=TotalMinutesAsleep/60 ,y=SedentaryMinutes/60, color=TotalSteps))+
  geom_point()+
  scale_color_gradient(low="steelblue", high="orange") +
  ylab("sedentary hours")+
  xlab("total sleep hours")
```

```
## Warning: Removed 971 rows containing missing values (geom_point).
```



```
ggplot(data=merged_data, aes(x=TotalMinutesAsleep/60 ,y=VeryActiveMinutes/60, color=TotalSteps))
+
  geom_point()+
  scale_color_gradient(low="steelblue", high="orange")+
  ylab("very active hours")+
  xlab("total sleep hours")
```

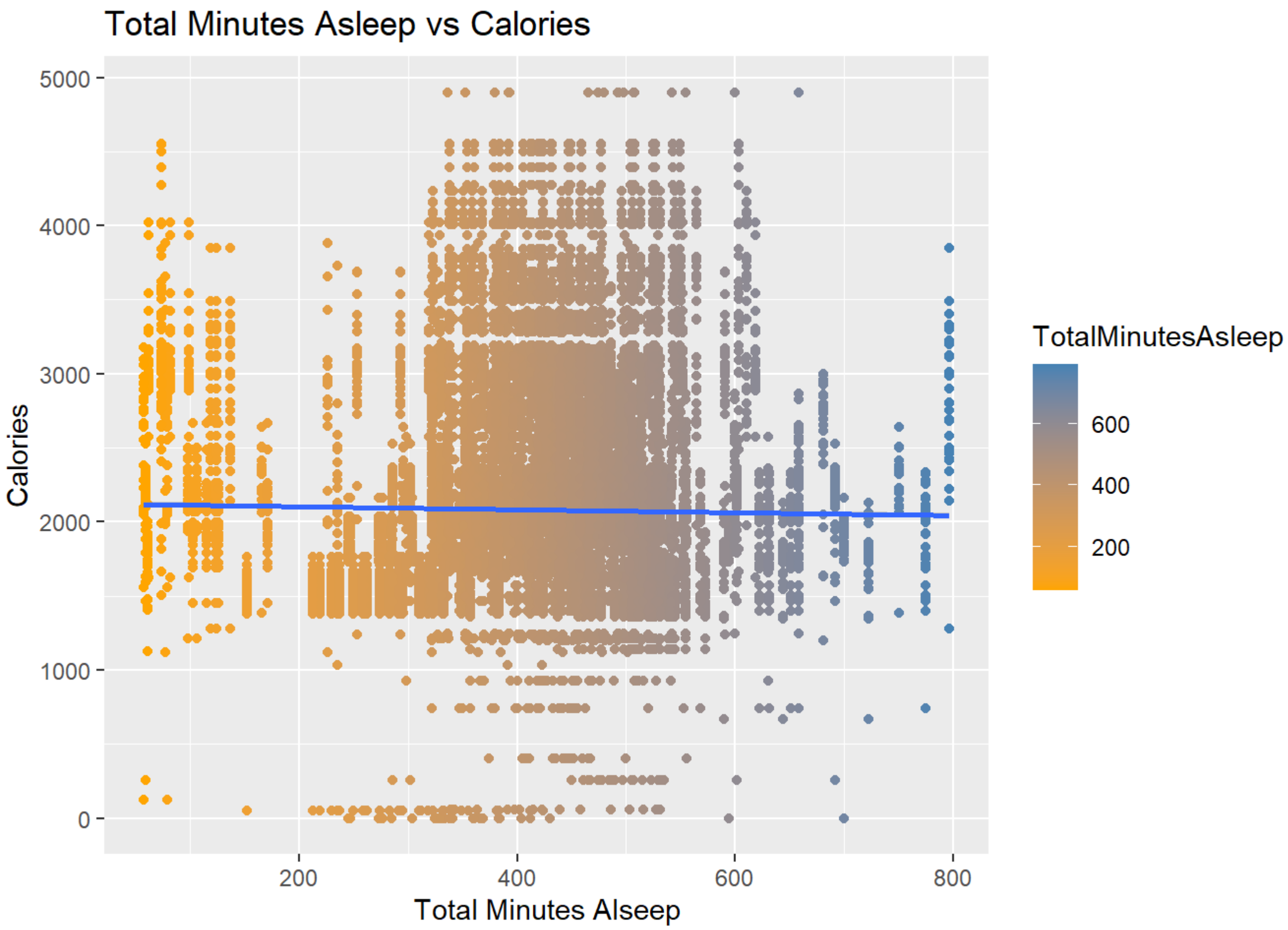
```
## Warning: Removed 971 rows containing missing values (geom_point).
```



```
ggplot(data=merged_data, aes(x=TotalMinutesAsleep, y = Calories, color=TotalMinutesAsleep))+
  geom_point()+
  labs(title="Total Minutes Asleep vs Calories")+
  xlab("Total Minutes Alseep")+
  stat_smooth(method=lm)+
  scale_color_gradient(low="orange", high="steelblue")
```

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

```
## Warning: Removed 971 rows containing non-finite values (stat_smooth).

## Warning: Removed 971 rows containing missing values (geom_point).
```


Total Minutes Asleep vs Calories
Calories
5000
4000
3000
2000
1000
0
200
400
600
800
Total Minutes Alseep
TotalMinutesAsleep
600
400
200