

```
date <- format(Sys.time(), "%b %d %Y")
```

Project 1: Daily Step Movement

Submitted by Andrew Chang on Jan 10 2021 for the Johns Hopkins Reproducible Research course

Introduction

Using two months of data collecting personal step activity at 5 minute intervals a day from a monitoring device, the following study details central tendencies in total daily steps, step activity trends across intervals throughout the day, and differences between step activity within different parts of the week.

Loading and Preprocessing the Data

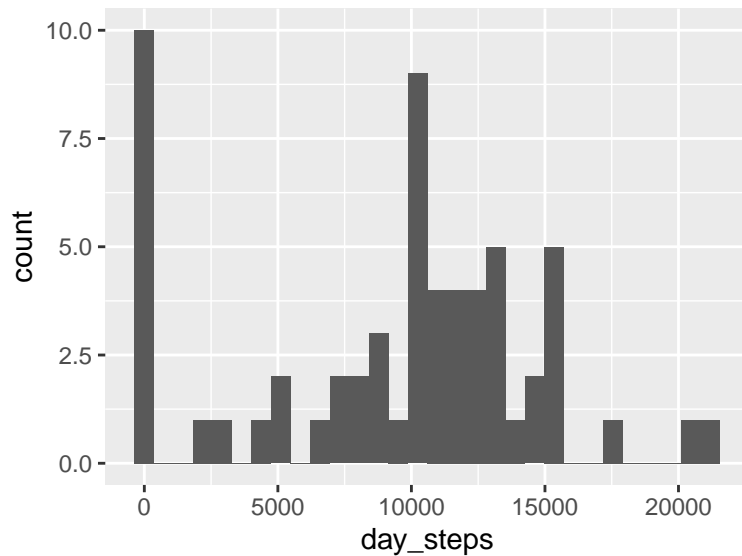
The following code was used to read in the data and specify vector types:

```
readAct <- read.csv("C:/Users/arkai/Documents/R/projects/DSS05_RepRes/RepData_PeerAssessment1/repdata_d  
                  colClasses = c("numeric", "POSIXct", "integer"))
```

Central Tendency

To find the central tendencies of the number of steps per day, steps were aggregated by day and the metrics before the mean and median were taken.

```
actDaily <- readAct %>%  
  select(date, steps) %>%  
  group_by(date) %>%  
  summarize(  
    day_steps = sum(steps, na.rm = TRUE)  
  )  
  
actDailyMean <- round(mean(actDaily$day_steps, na.rm = TRUE), 1)  
actDailyMed <- round(median(actDaily$day_steps, na.rm = TRUE), 1)  
  
ggplot(data = actDaily, aes(day_steps)) +  
  geom_histogram()
```



The mean and median number of steps taken each day are 9354.2 and 10395, respectively. Through this process, the data revealed a number of missing values as well as zeros. The latter were retained, since the intervals in which they occurred seemed to be adjacent to active ones. The former were removed in the calculation of central tendencies.

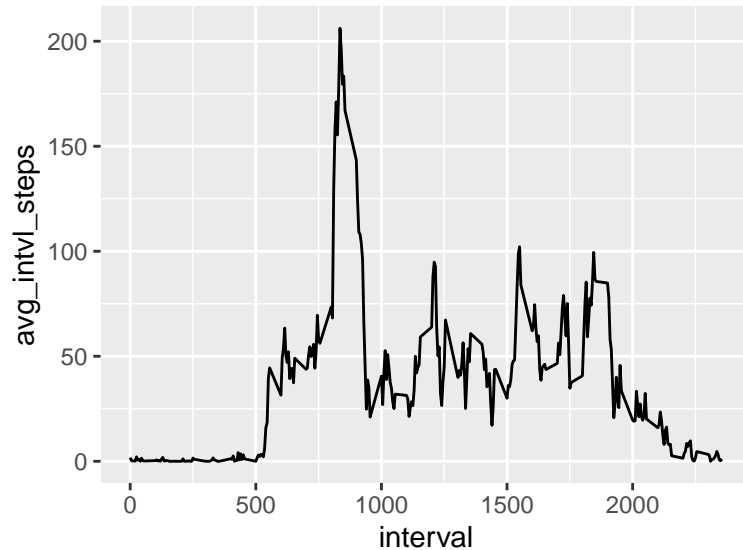
Daily Step Trends

The steps were then averaged across days for every interval, again removing missing values:

```
actIntvl <- readAct %>%
  select(
    date, steps, interval
  ) %>%
  group_by(interval) %>%
  summarize(
    avg_intvl_steps = mean(steps, na.rm = TRUE)
  )

maxIntvlSteps <- round(max(actIntvl$avg_intvl_steps, na.rm = TRUE), 1)
maxIntvl <- actIntvl$interval[which.max(actIntvl$avg_intvl_steps)]

ggplot(data = actIntvl, aes(interval, avg_intvl_steps)) +
  geom_line()
```



The interval with the highest average daily step count is 835 with 206.2 steps.

```
avgSteps <- readAct %>%
  select(
    steps, date, interval
  ) %>%
  filter(is.na(steps)) %>%
  inner_join(actIntvl, by = "interval") %>%
  select(
    date,
    interval,
    avg_intvl_steps
  )

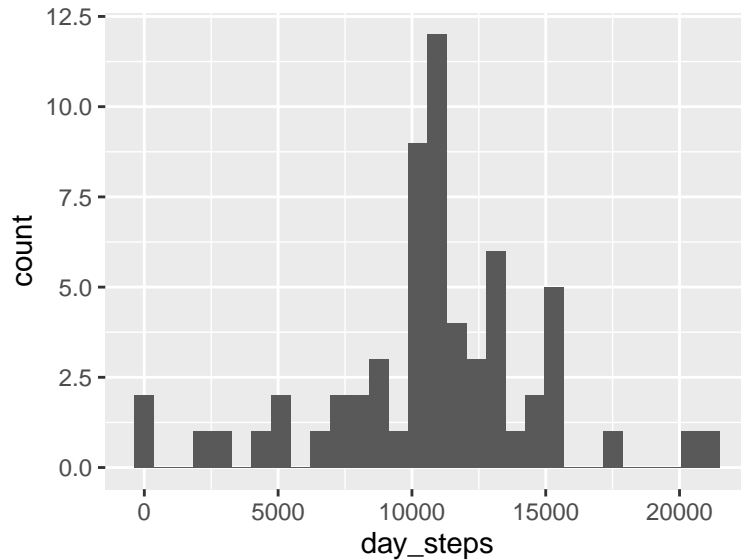
actDaily_imp <- readAct %>%
  left_join(avgSteps, by = c("date", "interval")) %>%
  mutate(
    steps = ifelse(is.na(steps), avg_intvl_steps, steps)
  )

actDaily_impAvg <- actDaily_imp %>%
  select(
    date, steps
  ) %>%
  group_by(date) %>%
  summarize(
    day_steps = sum(steps, na.rm = TRUE)
  )

nullStepRows <- avgSteps %>%
  count()
nullStepCount <- nullStepRows$n

actDailyMean_imp <- round(mean(actDaily_impAvg$day_steps, na.rm = TRUE), 1)
actDailyMed_imp <- round(median(actDaily_impAvg$day_steps, na.rm = TRUE), 1)
```

```
ggplot(data = actDaily_impAvg, aes(day_steps)) +  
  geom_histogram()
```



To address the 2304 missing values, a new dataframe was created which collected the average daily steps from the non-null step counts. After the imputation of missing values, the new mean and median are 10766.2 and 10766.2 respectively. The operation removed a number of zero-step averages previously skewing the data and caused the two measures of central tendency to converge.

Weekly Patterns

Finally, splitting the data by weekday and weekend yielded a panel showing a significantly narrower range of activity during weekends:

```
readAct_wkpt <- actDaily_imp %>%  
  mutate(  
    weekpart = factor(ifelse(weekdays(date) %in% c("Saturday", "Sunday"), "weekend", "weekd  
  )  
  
actDaily_wkptAvg <- readAct_wkpt %>%  
  select(  
    date, weekpart, steps  
  ) %>%  
  group_by(date, weekpart) %>%  
  summarize(  
    day_steps = sum(steps, na.rm = TRUE)  
  )  
  
ggplot(data = actDaily_wkptAvg, aes(day_steps)) +  
  geom_histogram() +  
  facet_grid(rows = vars(weekpart))
```

