PAI

2023-12-27

Introduction

The R Markdown document is apart of the Reproducible Research online Data Science course offered by John Hopkins University on Coursera. The R Markdown document is for "Peer-graded Assignment: Course Project 1" assignment requiring students to answer questions based on the "activity monitoring data".

The Data

The assignments makes use of data from a personal activity monitoring device. This device collects data at 5 minute intervals through out the day. The data consists of two months of data from an anonymous individual collected during the months of October and November, 2012 and include the number of steps taken in 5 minute intervals each day.

Loading the Data

First the user needs to unzip the csv file and read in the data using the read.csv function.

```r
unzip("activity.zip")
```

```
## Warning in unzip("activity.zip"): error 1 in extracting from zip file
```

```r
activity_data <- read.csv("activity.csv")
```

Loading the Necessary Packages

Following loading the data into the R environment the user needs to download the necessary packages. Depending on the user's package preference will cause the loaded package preferences to vary. For this project ggplot2, dplyr, and lubridate were utilized.

```r
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```r
library (lubridate)
```

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

```
## The following objects are masked from 'package:base':
##
```

```
##     date, intersect, setdiff, union
```

Answering: What is the total mean steps taken per day?

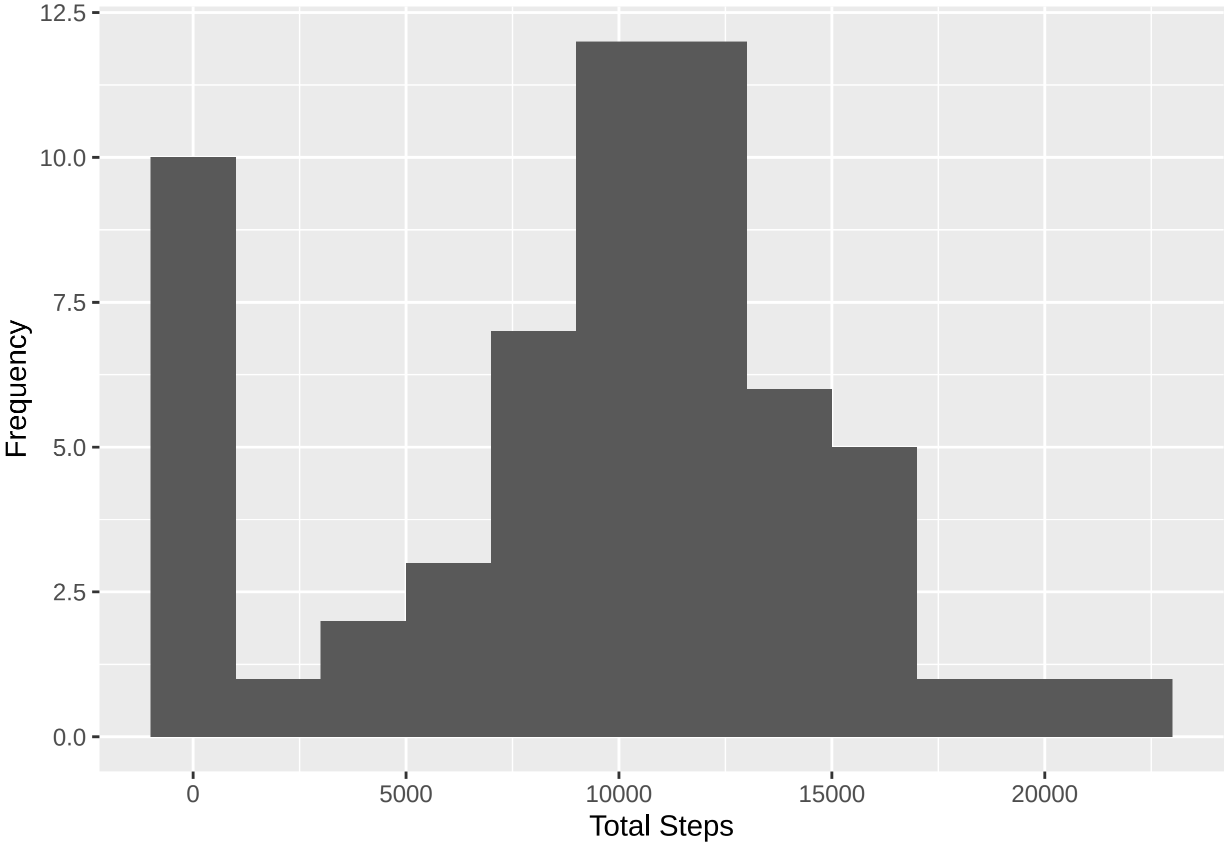
The following code calculates the total, median, and mean of steps taken per day. At this time the missing values can be ignored in the dataset and will be addressed later on.

```
steps_total <- activity_data %>%
  group_by(date) %>%
  summarise(daily_steps = sum(steps, na.rm = TRUE))

mean_steps = mean(steps_total$daily_steps, na.rm=TRUE)
median_steps = median(steps_total$daily_steps, na.rm=TRUE)
```

The following histogram demonstrates the total steps per day and its frequency.

```
ggplot(steps_total, aes(daily_steps)) + geom_histogram(binwidth = 2000) +
  xlab("Total Steps") +
  ylab("Frequency")
```

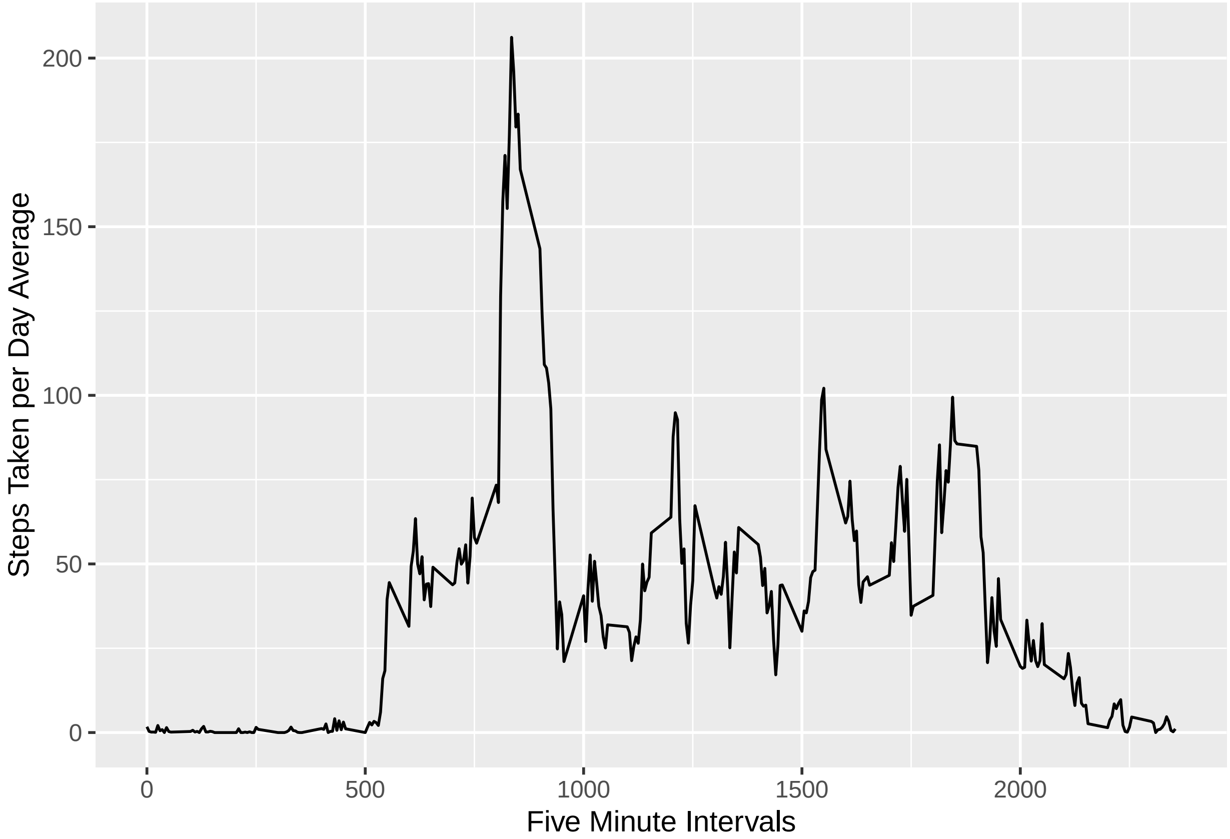


Answering: What is the daily activity pattern?

The following histogram demonstrates the daily activity pattern.

```
interval_steps <- activity_data %>%
  group_by(interval) %>%
  summarise(steps = mean(steps, na.rm =TRUE))

ggplot(data=interval_steps, aes(x=interval, y=steps)) +
  geom_line() +
  xlab("Five Minute Intervals") +
  ylab("Steps Taken per Day Average")
```



Imputing Missing Values

The following code calculates the total missing values in the original dataset used to calculate the total, average, and median steps.

```
missing_values <- !complete.cases(activity_data)
```

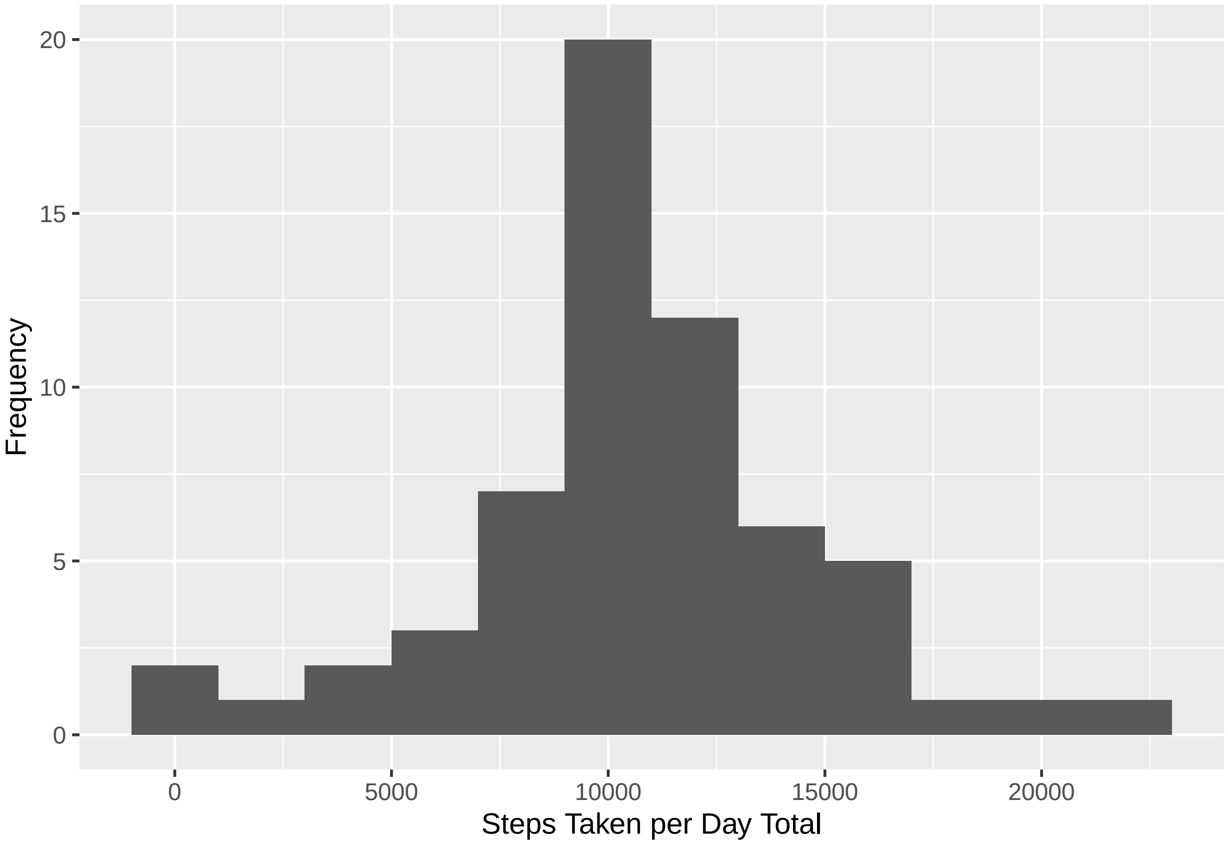
After calculating the total missing values in the dataset, the missing values were filled in.

```
imputed_data <- activity_data %>%
  mutate(
    steps = case_when(
      is.na(steps) ~ interval_steps$steps[match(activity_data$interval, interval_steps$interval)],
      TRUE ~ as.numeric(steps)
    ))
```

Based on the imputed dataset, a new histogram was created representing the daily steps.

```
imputed_total <- imputed_data %>% group_by(date) %>% summarise(daily_steps = sum(steps))

ggplot(imputed_total, aes(daily_steps)) +
  geom_histogram(binwidth = 2000) +
  xlab("Steps Taken per Day Total") +
  ylab("Frequency")
```



##Calculating the New Mean and Median on the Imputed Data The following code calculate the new mean and median based on the Imputed Data.

```
imputed_mean = mean(imputed_total$daily_steps, na.rm=TRUE)
imputed_median = median(imputed_total$daily_steps, na.rm=TRUE)
```

The new imputed mean and median and the mean and median from the original dataset difference were calculated utilizing the following code.

```
mean_difference <- mean_steps - imputed_mean
median_difference <- median_steps - imputed_median
```

Answering: Is There a Difference Between Weekday and Weekend Activity Levels?

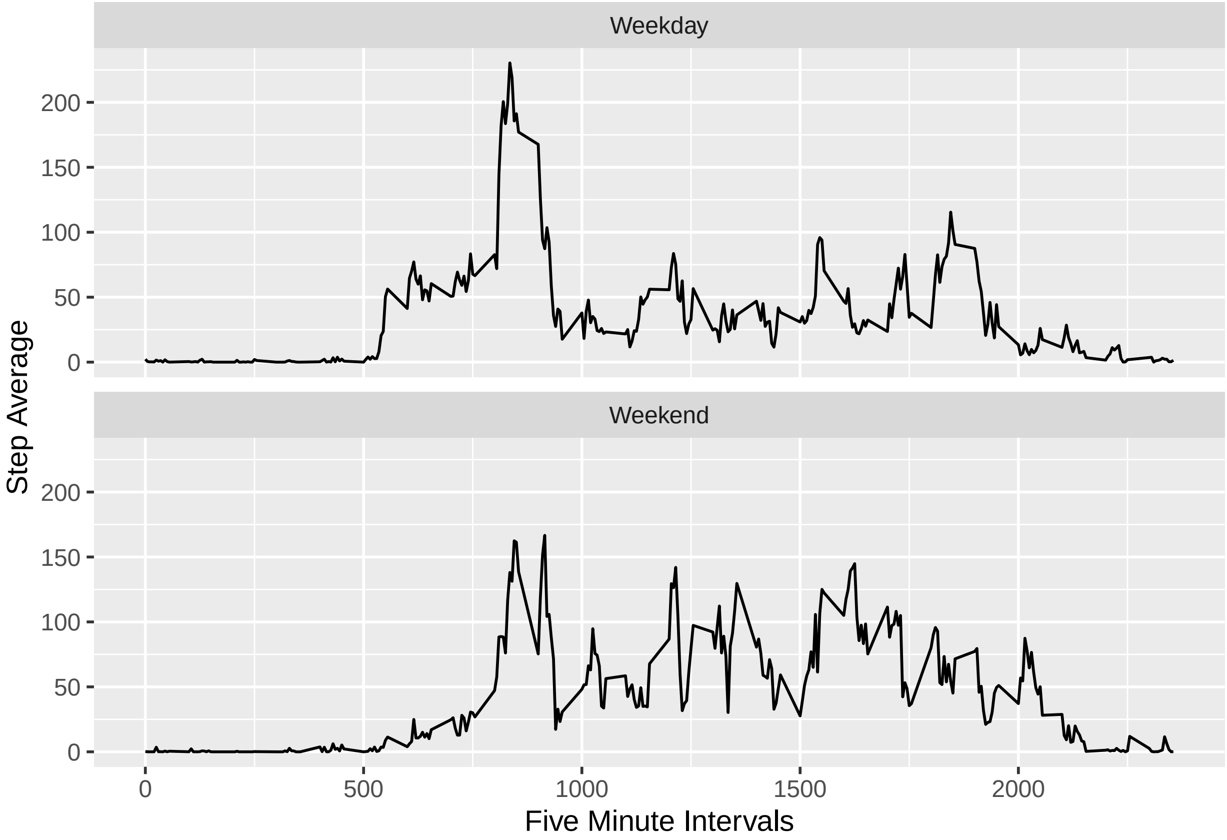
The following code calculates the difference between weekday and weekend activity levels.

```
day <- imputed_data %>%
  mutate(
    date = ymd(date),
    weekday_or_weekend = case_when(wday(date) %in% 2:6 ~ "Weekday",
                                   wday(date) %in% c(1,7) ~ "Weekend")
  ) %>% select(-date) %>%
  group_by(interval, weekday_or_weekend) %>%
  summarise(
    steps = mean(steps)
  )
```

```
## `summarise()` has grouped output by 'interval'. You can override using the
## `.groups` argument.
```

The following histogram demonstrates the activity difference.

```
ggplot(day, aes(interval, steps)) +
  geom_line() +
  facet_wrap(~weekday_or_weekend, nrow = 2) +
  xlab("Five Minute Intervals") +
  ylab("Step Average")
```


Weekday
Weekend
Step Average
Five Minute Intervals