Activity Monitoring Summary

B. Tyson Dube

February 21, 2016

Introduction

It is now possible to collect a large amount of data about personal movement using activity monitoring devices such as a Fitbit, Nike Fuelband, or Jawbone Up.

This analysis makes use of data from a personal activity monitoring device that 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.

Using this data we seek to answer the following questions:

1. What is mean total number of steps taken per day?
2. What is the average daily activity pattern?
3. Are there differences in activity patterns between weekdays and weekends?

Getting Started: Exploratory Analysis

```
require(ggplot2)
require(lubridate)
require(RColorBrewer)
require(dplyr)
require(ggthemes)
require(scales)
require(lattice)
```

Now we load the data in a fully reproducible way.

```
loadData <- function(dataURL="", destF="default.csv", method = NULL){
  if(!file.exists(destF)){
            temp <- tempfile()
            download.file(dataURL, temp, method = method)
            unzip(temp, destF)
            unlink(temp)
        }else{
            message("Data already downloaded.")
        }
}

dataURL <-"https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"

loadData(dataURL, "activity.csv", method = "curl")

active <- read.csv("activity.csv")

active$date<-as.Date(active$date)
```

We can also create a new column for the days for the week. This will be helpful in answering the questions about activity level throughout the week.

```
active$Weekday<-wday(active$date, label = TRUE, abbr = FALSE)
head(active)
```

```
##   steps       date interval Weekday
## 1    NA 2012-10-01        0  Monday
## 2    NA 2012-10-01        5  Monday
## 3    NA 2012-10-01       10  Monday
## 4    NA 2012-10-01       15  Monday
## 5    NA 2012-10-01       20  Monday
## 6    NA 2012-10-01       25  Monday
```

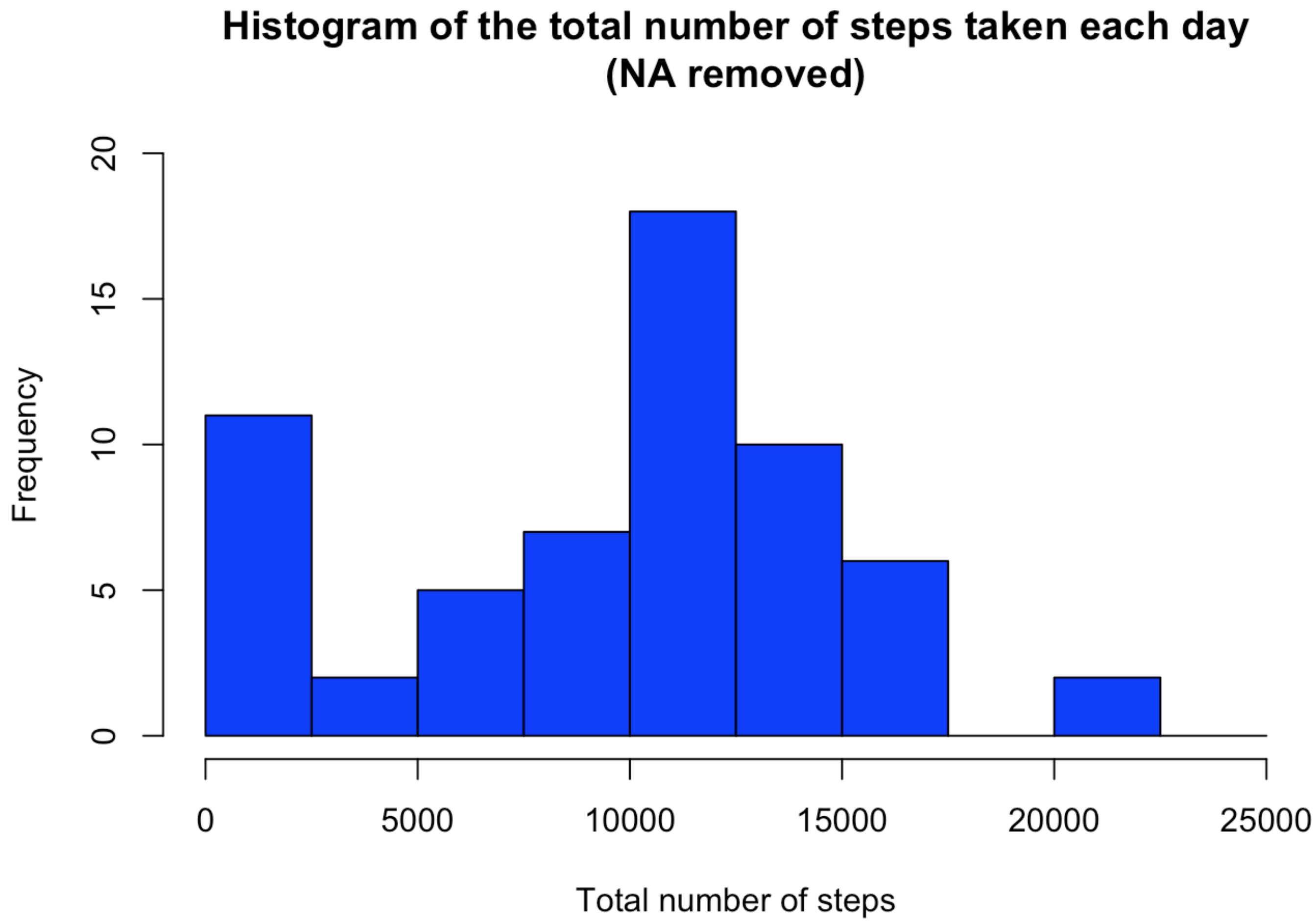
What is mean total number of steps taken per day?

We decide not to use imputation to answer this question and therefore ignore the NA values in the data set. This requires that we compute the total number of steps taken per day.

```
sum_data <- aggregate(active$steps, by=list(active$date), FUN=sum, na.rm=TRUE)
names(sum_data) <- c("date", "total")
head(sum_data)
```

```
##         date total
## 1 2012-10-01     0
## 2 2012-10-02   126
## 3 2012-10-03 11352
## 4 2012-10-04 12116
## 5 2012-10-05 13294
## 6 2012-10-06 15420
```

```
hist(sum_data$total,
     breaks=seq(from=0, to=25000, by=2500),
     col="blue",
     xlab="Total number of steps",
     ylim=c(0, 20),
     main="Histogram of the total number of steps taken each day\n(NA removed)")
```

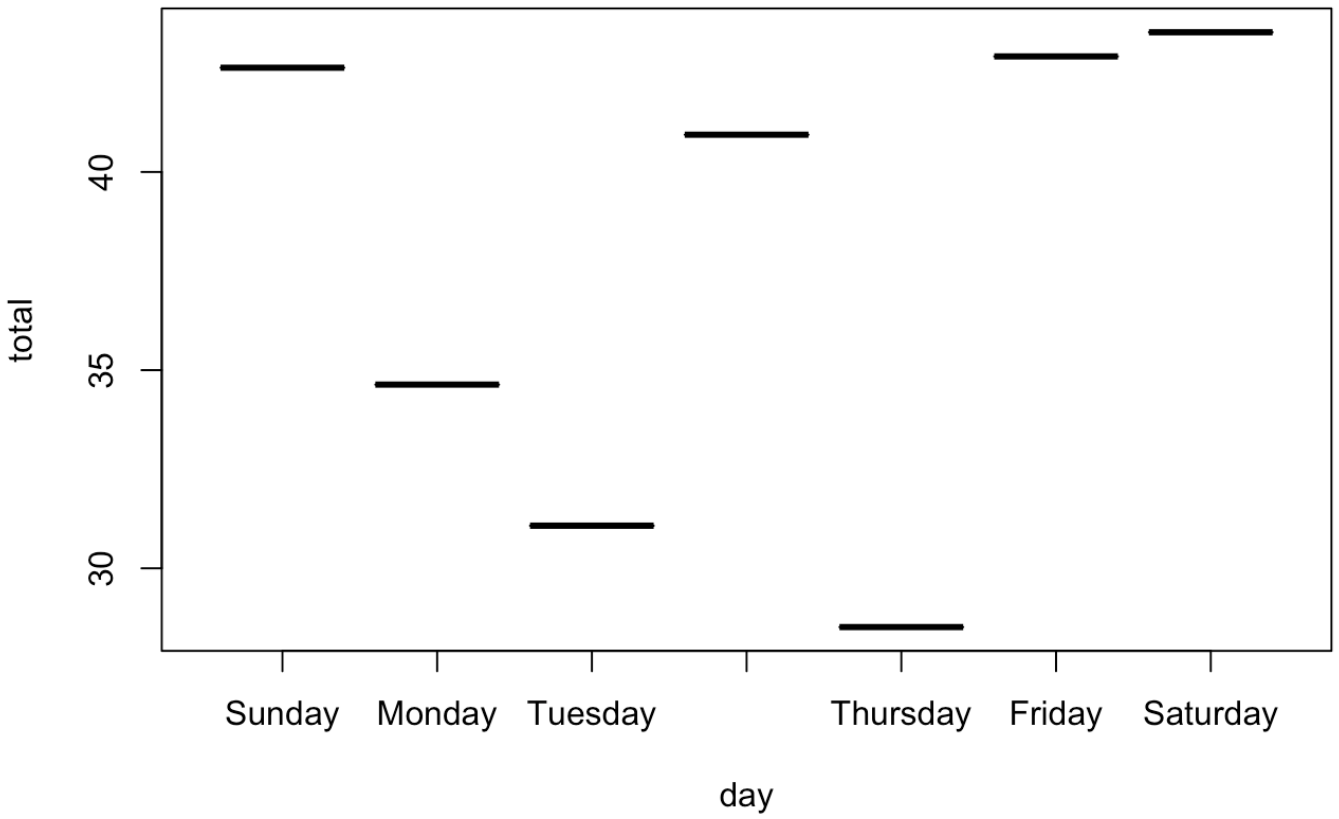


Let’s also look at the average by weekday:

```
avg_data <- aggregate(active$steps, by=list(active$Weekday), FUN=mean, na.rm=TRUE)
names(avg_data) <- c("day", "total")
head(avg_data)
```

```
##         day    total
## 1    Sunday 42.63095
## 2    Monday 34.63492
## 3   Tuesday 31.07485
## 4 Wednesday 40.94010
## 5  Thursday 28.51649
## 6    Friday 42.91567
```

```
plot(avg_data)
```



The mean and median steps per day are 9,354 and 10,395 respectively.

```
mean(sum_data$total)
```

```
## [1] 9354.23
```

```
median(sum_data$total)
```

```
## [1] 10395
```

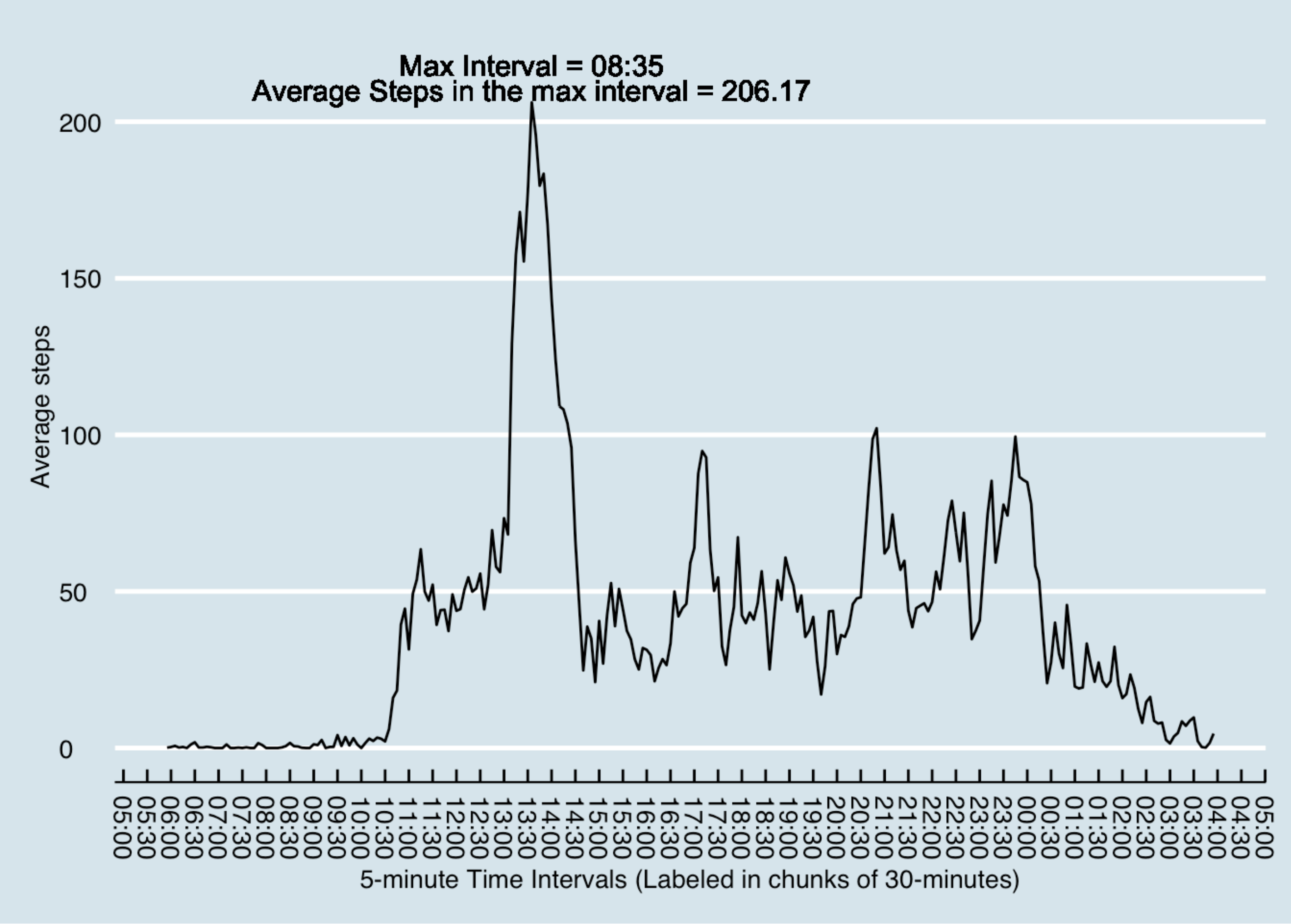
What is the average daily activity pattern?

Here we look at minute intervals as a time series plot showing the average number of steps accross all days in the dataset. This plot also shows the max interval value.

```
active$Interval <- as.POSIXct(strptime(sprintf("%04d", active$interval), "%H%M"))
make.max.interval.ggplot<- function(active.dataframe){
  active.intervals <- active.dataframe %>%
    group_by(Interval) %>%
    summarise(Average = mean(steps, na.rm = TRUE)) %>%
    arrange(Interval)
  max.active <- active.intervals[which.max(active.intervals$Average),]
  max.interval <<- max.active$Interval[1]
  max.average <<- round(max.active$Average[1], 2)
  ggplot(active.intervals, aes(x = Interval, y = Average)) +
    geom_line() +
    theme_economist() +
    geom_text(aes(label = paste("Max Interval =", format(max.interval, "%H:%M")),
                  x = max.interval,
                  y = max.average + 12),
              color = "black",
              size = 4) +
    geom_text(aes(label = paste("Average Steps in the max interval =", max.average
),
                  x = max.interval,
                  y = max.average + 4),
              color = "black",
              size = 4) +
    theme(axis.text.x=element_text(angle=270,
                                   hjust=1,
                                   vjust=0.5,
                                   size = 10)) +
    scale_x_datetime(breaks = date_breaks("30 mins"),
                     labels = date_format("%H:%M"),
                     limits = c(active.intervals$Interval[12], active.intervals$In
terval[286-10])) +
    ylab("Average steps") +
    xlab("5-minute Time Intervals (Labeled in chunks of 30-minutes)")
}

make.max.interval.ggplot(active)
```

```
## Warning: Removed 23 rows containing missing values (geom_path).
```



Imputing missing values

First we calculate the number of missing values in the data set. We see that there are 2,304 missing values in this dataset.

```
sum(is.na(active$steps))
```

```
## [1] 2304
```

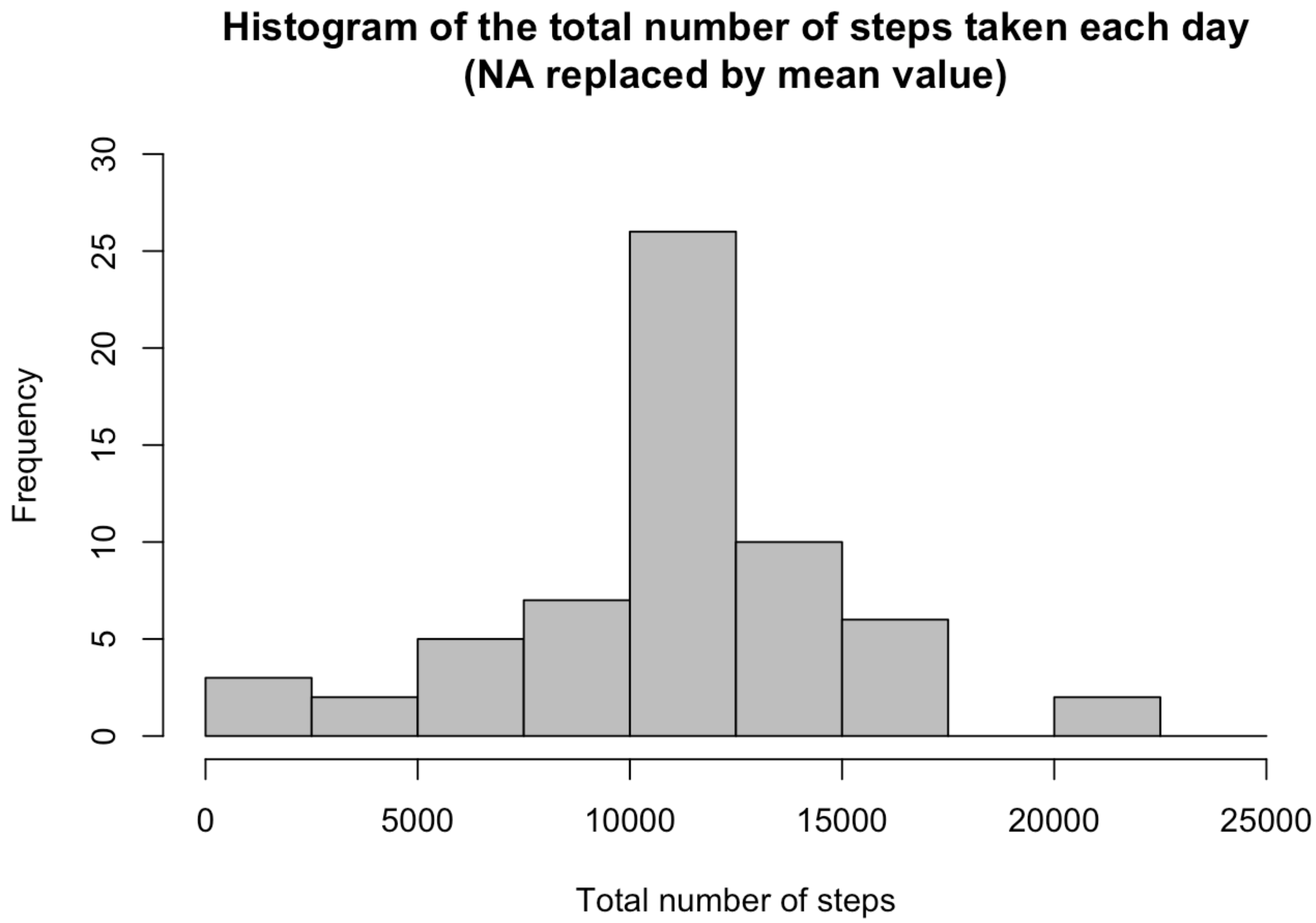
We use mean imputation to correct for missing values and create a new dataset that is equal to the original dataset but with the missing data filled in.

```
p <- which(is.na(active$steps))
m <- rep(mean(active$steps, na.rm=TRUE), times=length(p))
active[p,"steps"] <- m
head(active)
```

```
##     steps       date interval Weekday            Interval
## 1 37.3826 2012-10-01        0  Monday 2016-02-21 00:00:00
## 2 37.3826 2012-10-01        5  Monday 2016-02-21 00:05:00
## 3 37.3826 2012-10-01       10  Monday 2016-02-21 00:10:00
## 4 37.3826 2012-10-01       15  Monday 2016-02-21 00:15:00
## 5 37.3826 2012-10-01       20  Monday 2016-02-21 00:20:00
## 6 37.3826 2012-10-01       25  Monday 2016-02-21 00:25:00
```

Now we view the updated histogram.

```
sum_data <- aggregate(active$steps, by=list(active$date), FUN=sum)
names(sum_data) <- c("date", "total")
hist(sum_data$total,
     breaks=seq(from=0, to=25000, by=2500),
     col="grey",
     xlab="Total number of steps",
     ylim=c(0, 30),
     main="Histogram of the total number of steps taken each day\n(NA replaced by
mean value)")
```



We see that the mean and median for the NA imputed data are 10766 and 10766 respectively.

This is significantly different than the non-imputed data and likely has a lot to do with our simple imputaion method.

Are there differences in activity patterns between weekdays and weekends?

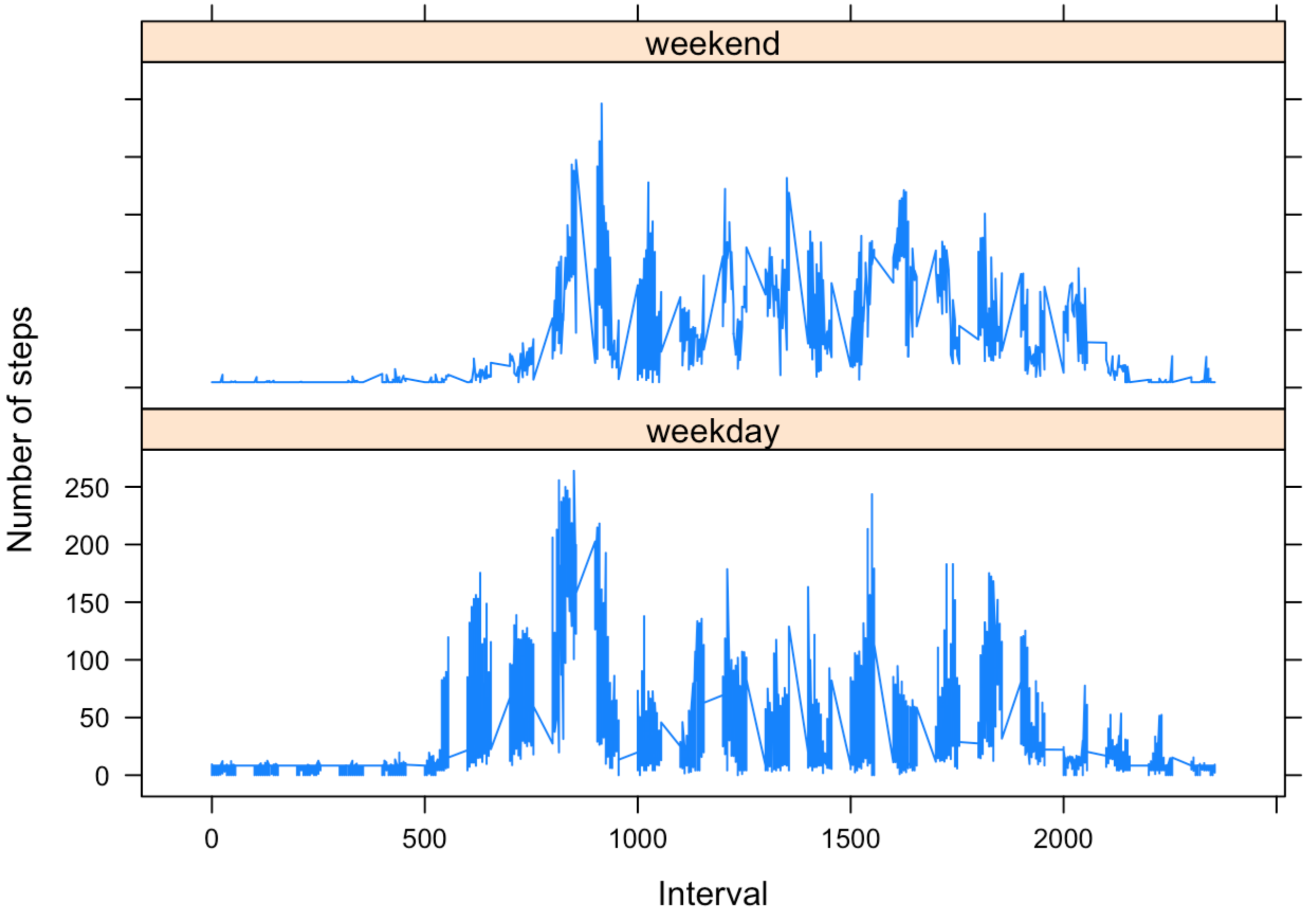
Here we create a new factor variable in the dataset with two levels - “weekdays” and “weekend” indicating whether a given date is a weekday or weekend day.

```
active$daytype <- ifelse(
  active$Weekday == "Saturday"|
    active$Weekday =="Sunday", "weekend", "weekday")

mean_data <- aggregate(active$steps,
                       by=list(active$daytype,
                               active$Weekday, active$interval), mean)
names(mean_data) <- c("daytype", "weekday", "interval", "mean")
head(mean_data)
```

```
##   daytype   weekday interval     mean
## 1 weekend    Sunday        0 4.672825
## 2 weekday    Monday        0 9.418355
## 3 weekday   Tuesday        0 0.000000
## 4 weekday Wednesday        0 7.931400
## 5 weekday  Thursday        0 9.375844
## 6 weekday    Friday        0 8.307244
```

```
xyplot(mean ~ interval | daytype, mean_data,
       type="l",
       lwd=1,
       xlab="Interval",
       ylab="Number of steps",
       layout=c(1,2))
```


weekend
weekday
Number of steps
0
50
100
150
200
250
Interval
0
500
1000
1500
2000