PA1 Activity Data

Jorge DIaz

July 9, 2017

The code below 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.

Part 1: Loading and Preprocessing Data

Step 1A: Load the data from the source file

```
## Read activity.csv file into "activity" variable
activity <- read.csv("activity.csv")

## Convert date field from factor to date
activity$date <- as.Date(activity$date)
```

Step 1B: Process data to make it ready for plotting and analysis

```
## Load reshape2 library to get melt & dcast functions
library(reshape2)

## Melt data frame to prep for casting by date -- by setting the id variable to date and the measure va
actMeltDate <- melt(activity, id.vars="date", measure.vars="steps", na.rm=FALSE)
```

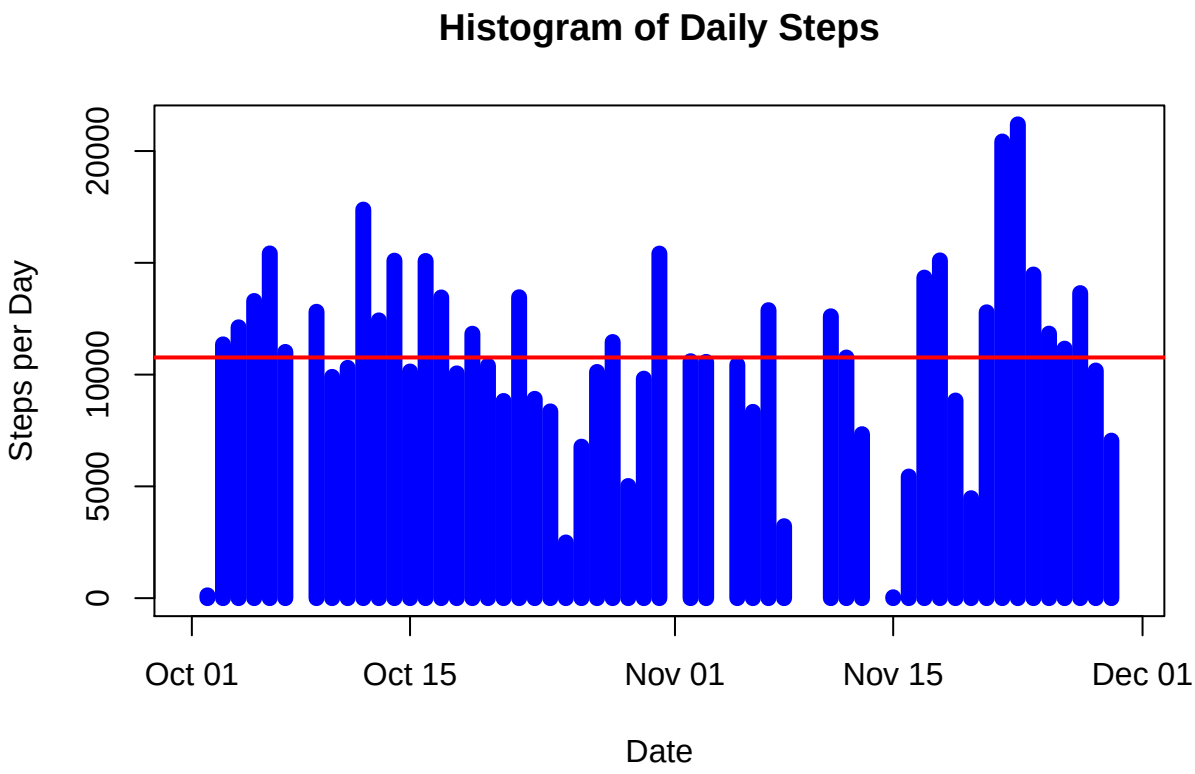
Part 2: What is the mean total number of steps taken per day?

Step 2A: Calculate total number of steps

```
## Cast data frame to see steps per day -- this sums the steps by date to give us a table of 3 columns
actCastDate <- dcast(actMeltDate, date ~ variable, sum)
```

Step 2B: Plot histogram of data

```
## Plot histogram with frequency of steps by day and add a red line showing the mean value
plot(actCastDate$date, actCastDate$steps, type="h", main="Histogram of Daily Steps", xlab="Date", ylab=
abline(h=mean(actCastDate$steps, na.rm=TRUE), col="red", lwd=2)
```




```
echo=TRUE
```

Step 2C: Calculate mean and median of steps by day

```
## Calculate mean and median of daily steps
paste("Mean Steps per Day =", mean(actCastDate$steps, na.rm=TRUE))
```

```
## [1] "Mean Steps per Day = 10766.1886792453"
```

```
paste("Median Steps per Day =", median(actCastDate$steps, na.rm=TRUE))
```

```
## [1] "Median Steps per Day = 10765"
```

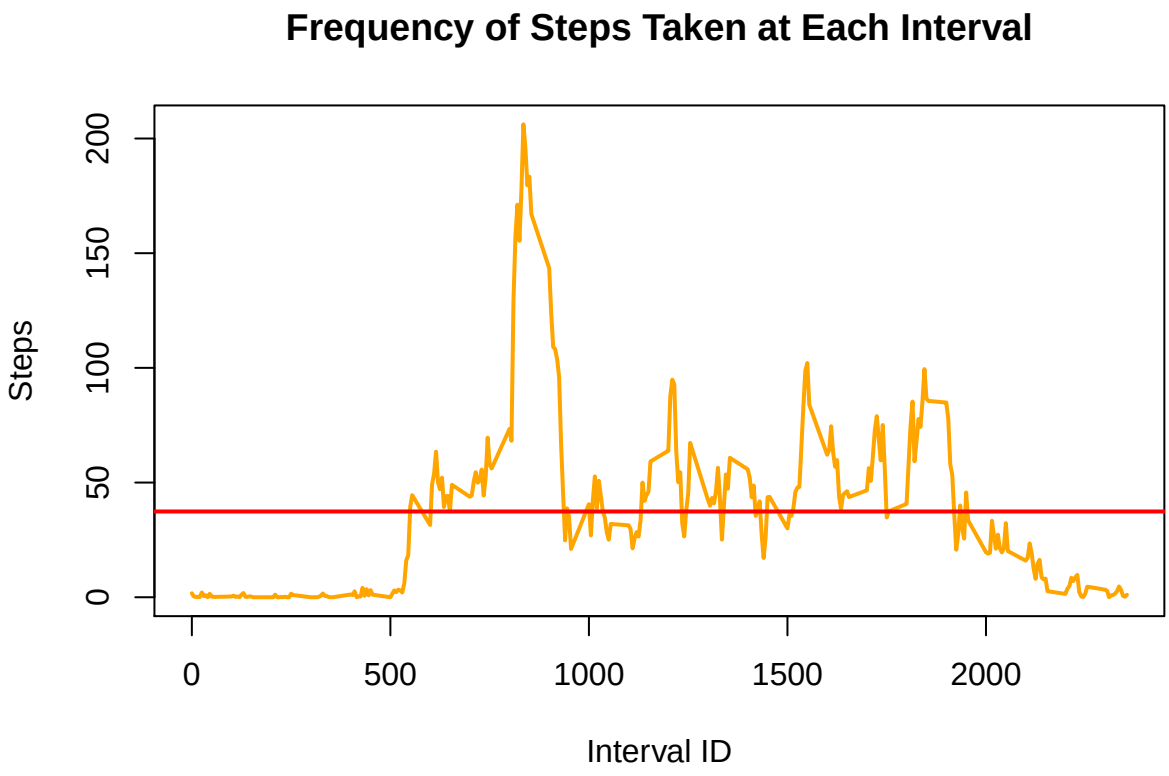
Part3: What is the average daily activity pattern?

Step 3A: Reprocess data to calculate by interval instead of day

```
## Re-melt data frame to prep for casting by interval, including removing NA values to take the mean
actMeltInt <- melt(activity, id.vars="interval", measure.vars="steps", na.rm=TRUE)

## Cast data frame to see mean steps per interval
actCastInt <- dcast(actMeltInt, interval ~ variable, mean)
```

Step 3B: Create a time series plot of average steps by interval

```
## Plot line chart with average frequency of steps by interval and add line with mean
plot(actCastInt$interval, actCastInt$steps, type="l", main="Frequency of Steps Taken at Each Interval",
abline(h=mean(actCastInt$steps, na.rm=TRUE), col="red", lwd=2)
```




```
## Output interval that has max value along with the max value
paste("Interval with max value =", actCastInt$interval[which(actCastInt$steps == max(actCastInt$steps))])
```

```
## [1] "Interval with max value = 835"
```

```
paste("Maximum interval mean steps =", max(actCastInt$steps))
```

```
## [1] "Maximum interval mean steps = 206.169811320755"
```

Part 4: Input missing values

Step 4A: Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

```
## Calculate number of rows in activity data set with NA rows
sum(is.na(activity$steps))
```

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

Step 4B: Document strategy for filling in all of the missing values in the dataset.

Replace NAs with the mean for the particular interval number. For example: if the average number of steps taken during interval x is y, replace each NA with the corresponding y value for that row. Recalculate the steps per day to see how much it differs from the original result (NA's included).

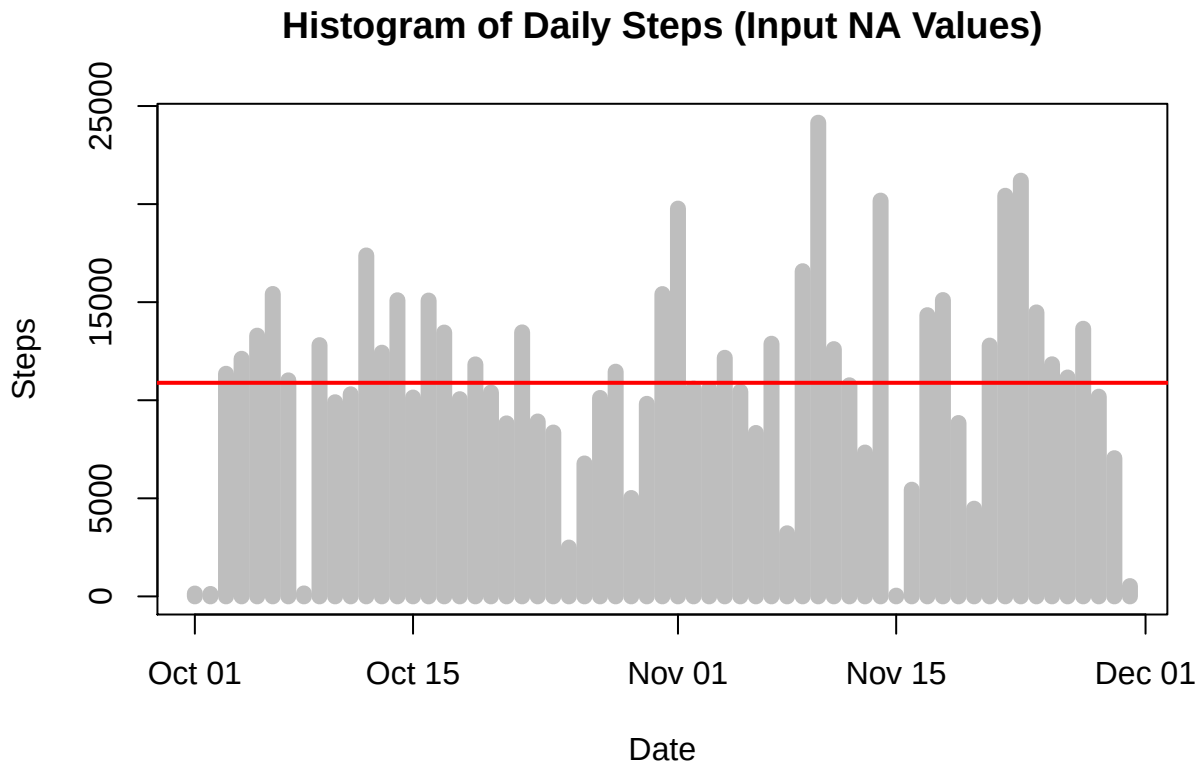
Step 4C: Create new data set with input NA values as stated in strategy

```
## Data frame with mean steps per interval - just renaming to be more descriptive
stepsPerInt <- actCastInt

## Create data frame that will remove NAs
```

```
actNoNA <- activity

## Merge activity data set with stepsPerInt data set
actMerge = merge(actNoNA, stepsPerInt, by="interval", suffixes=c(".act", ".spi"))

## Get list of indexes where steps value = NA
naIndex = which(is.na(actNoNA$steps))

## Replace NA values with value from steps.spi
actNoNA[naIndex,"steps"] = actMerge[naIndex,"steps.spi"]
```

Step 4D: Plot histogram and calculate mean and median of total steps/day with new (no NA) data set and compare with original.

```
## Melt new data frame to prep for casting by date
actMeltDateNoNA <- melt(actNoNA, id.vars="date", measure.vars="steps", na.rm=FALSE)

## Cast data frame to see steps per day
actCastDateNoNA <- dcast(actMeltDateNoNA, date ~ variable, sum)

## Plot histogram with frequency of steps by day
plot(actCastDateNoNA$date, actCastDateNoNA$steps, type="h", main="Histogram of Daily Steps (Input NA Va
abline(h=mean(actCastDateNoNA$steps), col="red", lwd=2)
```




```
## Calculate mean and median of daily steps
paste("Mean daily steps =", mean(actCastDateNoNA$steps, na.rm=TRUE))
```

```
## [1] "Mean daily steps = 10889.7992576554"
```

```
paste("Median daily steps =", median(actCastDateNoNA$steps, na.rm=TRUE))
```

```
## [1] "Median daily steps = 11015"
```

Difference in values: Original Data Set (with NA values) Mean daily steps = 10,766.19 vs. Median daily steps = 10,765 New Data Sets (NA's input with mean value for that interval) Mean daily steps = 10,890 vs. Median daily steps = 11,015

On a percentage basis, the difference in results between the original and new data sets was only 1.2% and 2.3% for the mean and median, respectively. However, the maximum daily value in the set with NA's vs. the set replacing NA's was 21,194 vs. 24,150, which differed more significantly at 13.9%.

Part 5: Are there differences in activity patterns between weekdays and weekends?

Step 5A: Create a new factor variable in the dataset with two levels - "weekday" and "weekend" indicating whether a given date is a weekday or weekend day.

```
## For loop to create new column called "dayOfWeek" and insert whether each date corresponds to a weekda
for (i in 1:nrow(actNoNA)) {
    if (weekdays(actNoNA$date[i]) == "Saturday" | weekdays(actNoNA$date[i]) == "Sunday") {
        actNoNA$dayOfWeek[i] = "weekend"
    } else {
        actNoNA$dayOfWeek[i] = "weekday"
    }
}
```

Step 5B: Make a panel plot containing a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis).

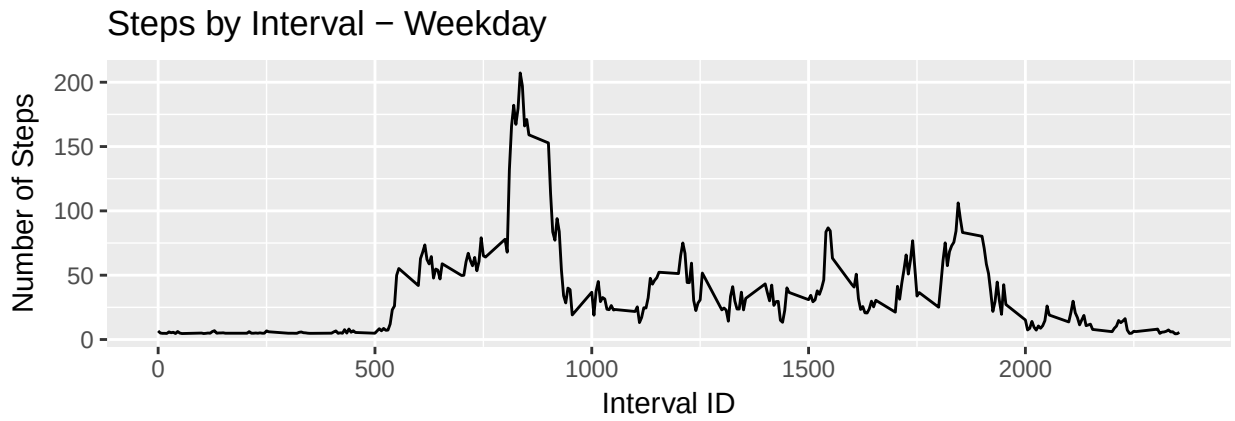
```
## To create a plot, we must first subset the data
actWeekday <- subset(actNoNA, dayOfWeek=="weekday")
actWeekend <- subset(actNoNA, dayOfWeek=="weekend")

## Next, we need to process the data for our needs
actMeltWeekday <- melt(actWeekday, id.vars="interval", measure.vars="steps")
actMeltWeekend <- melt(actWeekend, id.vars="interval", measure.vars="steps")
actCastWeekday <- dcast(actMeltWeekday, interval ~ variable, mean)
actCastWeekend <- dcast(actMeltWeekend, interval ~ variable, mean)

## Load plot packages necessary to continue
library(ggplot2)
library(gridExtra)

## Set plot area to two rows and one column, and then plot charts with mean line in each
plot1 <- qplot(actCastWeekday$interval, actCastWeekday$steps, geom="line", data=actCastWeekday, type="1
```

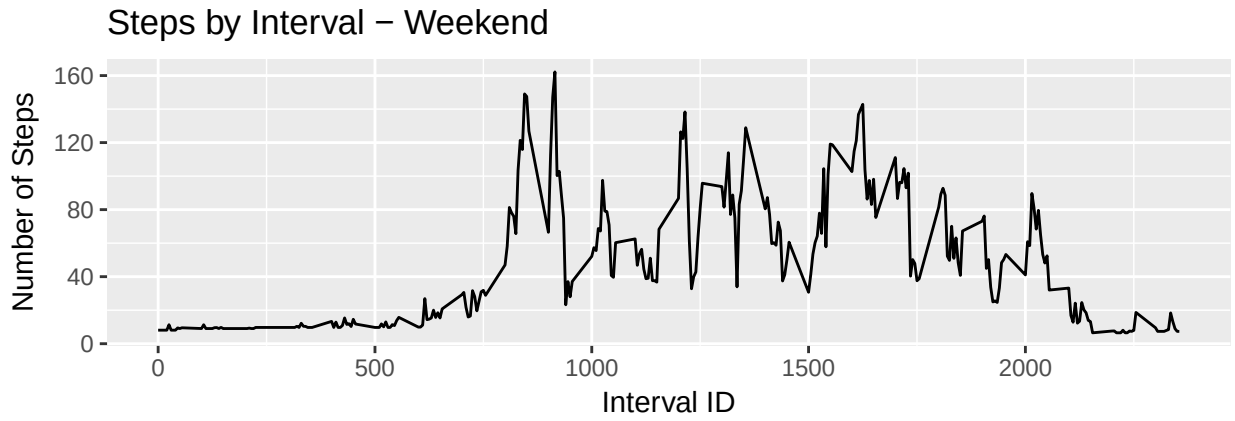
```
## Warning: Ignoring unknown parameters: type
```

```
plot2 <- qplot(actCastWeekend$interval, actCastWeekend$steps, geom="line", data=actCastWeekend, type="1
```

```
## Warning: Ignoring unknown parameters: type
```

```
grid.arrange(plot1, plot2, nrow=2)
```






```
echo=TRUE
```

echo=TRUE