Activity Data Review

Russell Husfeld

May 24, 2016

Loading and preprocessing the data

Show any code that is needed to

Load the data (i.e. read.csv())

Process/transform the data (if necessary) into a format suitable for your analysis

```
activity <- read.csv("~/Documents/activity.csv")
str(activity)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1 1 1 1 1
...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

```
#convert the date into Date formate "Y-m-d"
activity$date <- as.Date(activity$date,"%Y-%m-%d")
#check to make sure integer to date class has changed.
str(activity)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Date, format: "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

```
activity$steps <- as.numeric(activity$steps)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.1.3
```

```
##
## 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
```

For this part of the assignment, you can ignore the missing values in the dataset.

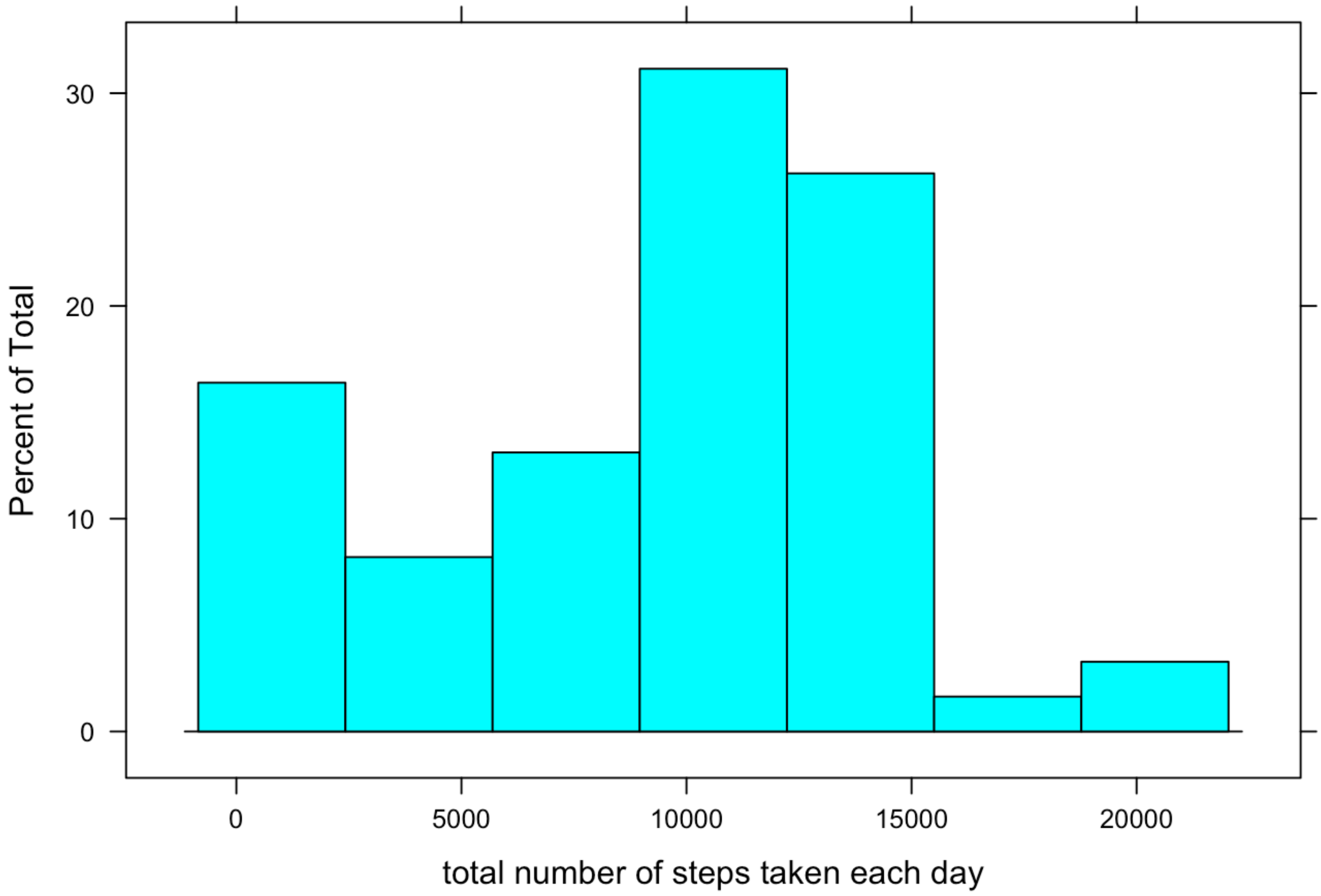
Make a histogram of the total number of steps taken each day

Calculate and report the mean and median total number of steps taken per day

```
activity.date<- activity %>%
  group_by(date) %>%
  summarize(tsteps=sum(steps, na.rm=TRUE))
library(lattice)
```

```
## Warning: package 'lattice' was built under R version 3.1.3
```

```
histogram(~activity.date$tstep, xlab="total number of steps taken each day")
```



```
meansteps<- mean(activity.date$tsteps)
mediansteps<- median(activity.date$tsteps)
print(meansteps)
```

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

```
print(mediansteps)
```

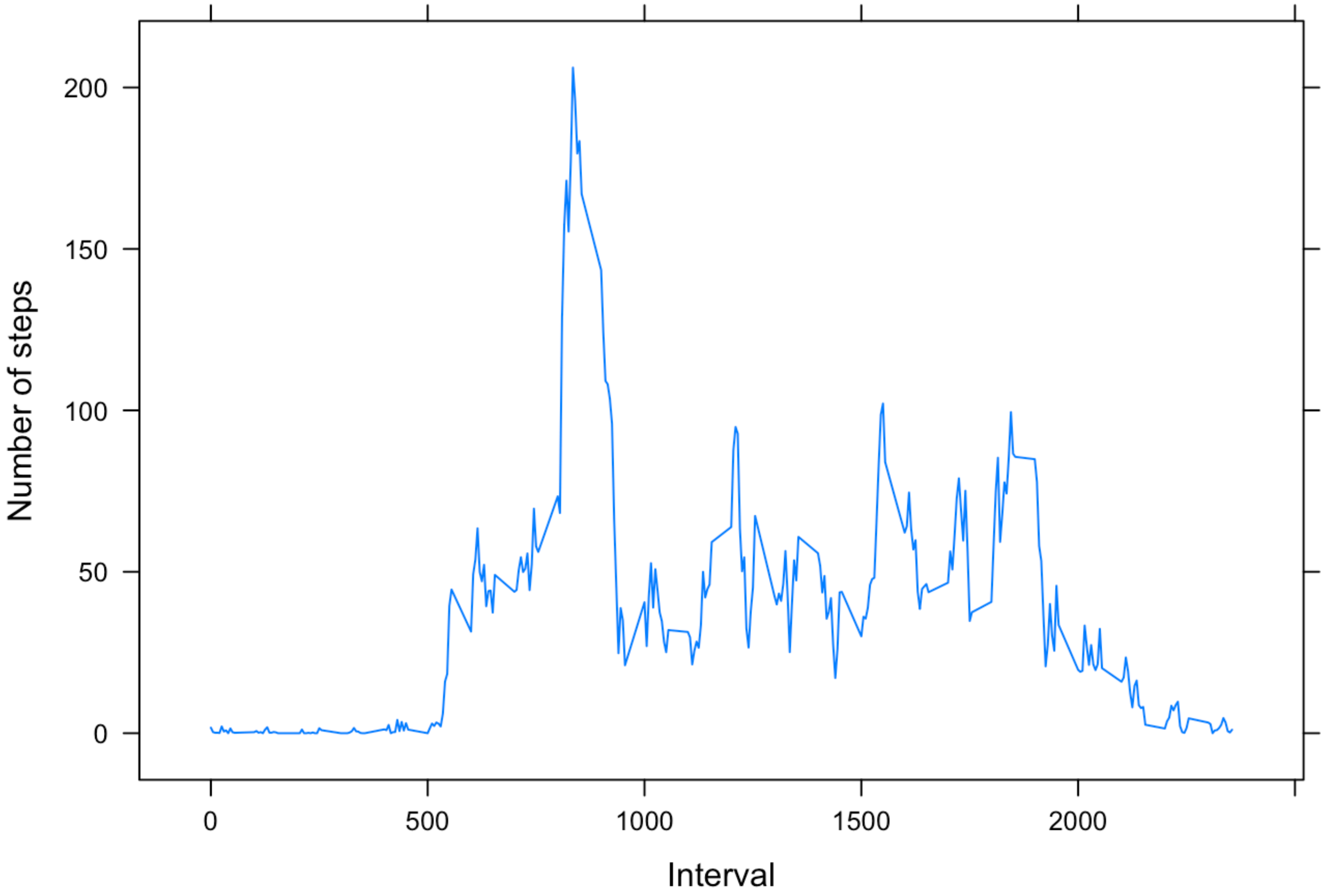
```
## [1] 10395
```

What is the average daily activity pattern?

Make 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 days (y-axis)

```
activityint<- activity %>%
  group_by(interval) %>%
  summarize(tsteps=sum(steps, na.rm=TRUE), avgsteps=mean(steps, na.rm=TRUE))

with(activityint, xyplot(avgsteps~interval, type="l", xlab="Interval", ylab="Number o
f steps"))
```



Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

```
maxint<- which.max(activityint$tsteps)
activityint[maxint, ]
```

```
## Source: local data frame [1 x 3]
## 
##   interval tsteps avgsteps
##      (int)  (dbl)    (dbl)
## 1      835  10927 206.1698
```

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

```
missing.values<- sum(is.na(activity$steps))
missing.values
```

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

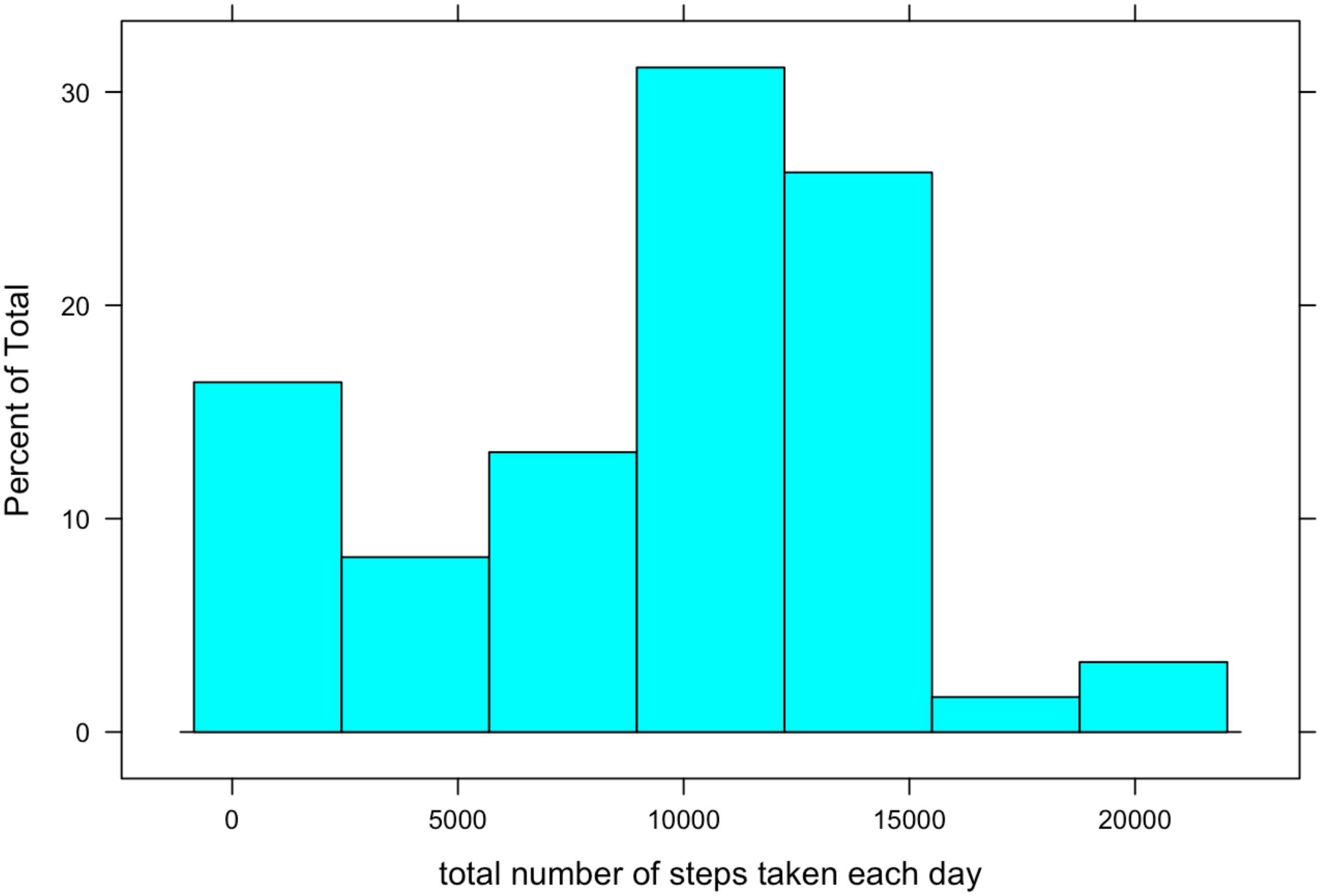
Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to be sophisticated. For example, you could use the mean/median for that day, or the mean for that 5-minute interval, etc.

Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
#use the mean for the day on any missiong values for the dataset.
activity.replaceNA<- activity %>%
  group_by(interval)  %>%
  mutate(steps= ifelse(is.na(steps), mean(steps, na.rm=TRUE), steps))
```

Make a histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day.

```
histogram(activity.date$tsteps, xlab="total number of steps taken each day")
```



Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps? Not a significant impact to the dataset.

Are there differences in activity patterns between weekdays and weekends?

For this part the weekdays() function may be of some help here. Use the dataset with the filled-in missing values for this part.

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.

```
library(lubridate)
```

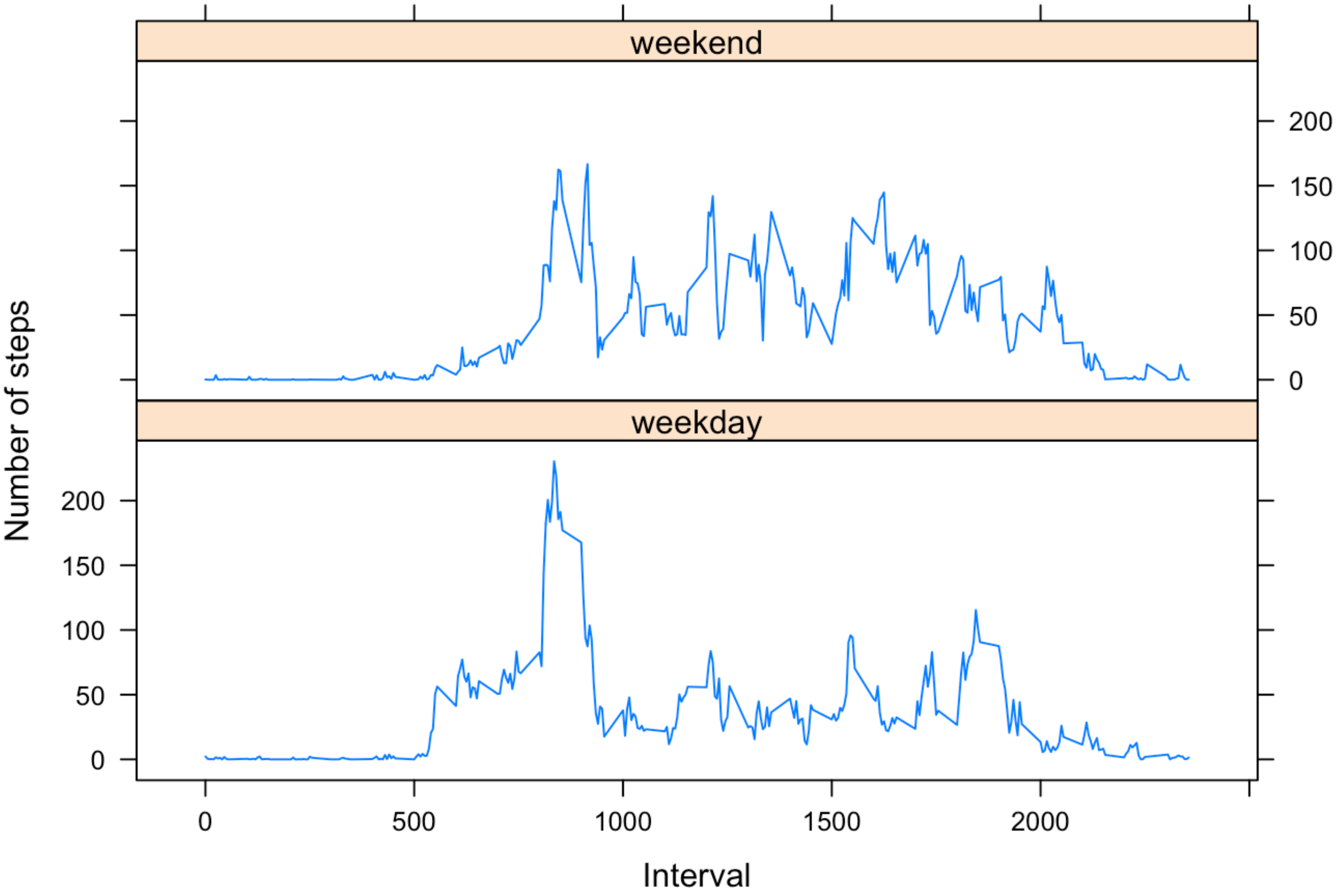
```
## Warning: package 'lubridate' was built under R version 3.1.3
```

```
activity.replaceNA$day<- wday(activity.replaceNA$date, label=TRUE)
activity.replaceNA$daytype<- activity.replaceNA$day
levels(activity.replaceNA$daytype) <- list(
    weekday = c("Mon", "Tues", "Wed", "Thurs", "Fri"),
    weekend = c("Sun", "Sat"))

activity.typeday<- activity.replaceNA %>%
  group_by(daytype, interval)  %>%
  summarize(total.steps=sum(steps, na.rm=TRUE), average.steps=mean(steps, na.rm=TRUE)
)
```

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).

```
xyplot(average.steps~interval|daytype, data=activity.typeday, type='l', layout=(c(1,2
)),
      ylab="Number of steps", xlab="Interval")
```


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