

Reproducible Research: Peer Assessment 1

William L. Kennedy

Loading and preprocessing the data

```
# Package dependencies (uncomment and install if necessary)

# install.packages("data.table")
# install.packages("lubridate")
# install.packages("lattice")
library(data.table)
library(lubridate)

##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:data.table':
##
##      hour, mday, month, quarter, wday, week, yday, year

library(lattice)

# set the working directory (replace path accordingly)
setwd("/Users/adakemia/Documents/Academic/Coursera/DataScienceSpecialization/05ReproducibleResearch/Pro

# List and check for necessary files
url <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
zfile <- "activity.zip"
file <- "activity.csv"

if (!file.exists(file)) {
  if (!file.exists(zfile)) {
    method <- switch(Sys.info()[['sysname']],
                     "Windows" = "internal",
                     "Darwin" = "curl",
                     "Linux" = "wget",
                     "auto")
    download.file(url, destfile=zfile, method)
  }
  unzip(zfile, exdir=".")
}

# Read files into data.table object
data <- fread(file, sep=",", stringsAsFactors=F, header=T,
              na.strings="NA", colClasses=c("numeric","character","numeric"))
# Check structure of data
str(data)

## Classes 'data.table' and 'data.frame':  17568 obs. of  3 variables:
```

```
## $ steps : num NA NA NA NA NA NA NA NA NA NA ...
## $ date : chr "2012-10-01" "2012-10-01" "2012-10-01" "2012-10-01" ...
## $ interval: num 0 5 10 15 20 25 30 35 40 45 ...
## - attr(*, ".internal.selfref")=<externalptr>
```

```
# Process data
```

```
# Set date formatting (lubridate package)
data[,date := ymd(date)]
```

```
##      steps      date interval
## 1:      NA 2012-10-01         0
## 2:      NA 2012-10-01         5
## 3:      NA 2012-10-01        10
## 4:      NA 2012-10-01        15
## 5:      NA 2012-10-01        20
## ---
## 17564:    NA 2012-11-30       2335
## 17565:    NA 2012-11-30       2340
## 17566:    NA 2012-11-30       2345
## 17567:    NA 2012-11-30       2350
## 17568:    NA 2012-11-30       2355
```

```
# Data quality checks
```

```
# Check for missingness
colSums(is.na(data))
```

```
##      steps      date interval
##      2304         0         0
```

```
# Percent missing
sum(is.na(data$steps)) / nrow(data) * 100
```

```
## [1] 13.11475
```

```
# after trying several fixes to the skip in the interval time entry
# I decided to leave it. I don't see a noticeable difference in output
# and several methods introduced anomalies I couldn't fix in the time
# allotted. Given more time I would convert to a timeseries object (e.g.,
# ts, zoo, TimeSeries, etc. for more complete analyses)
```

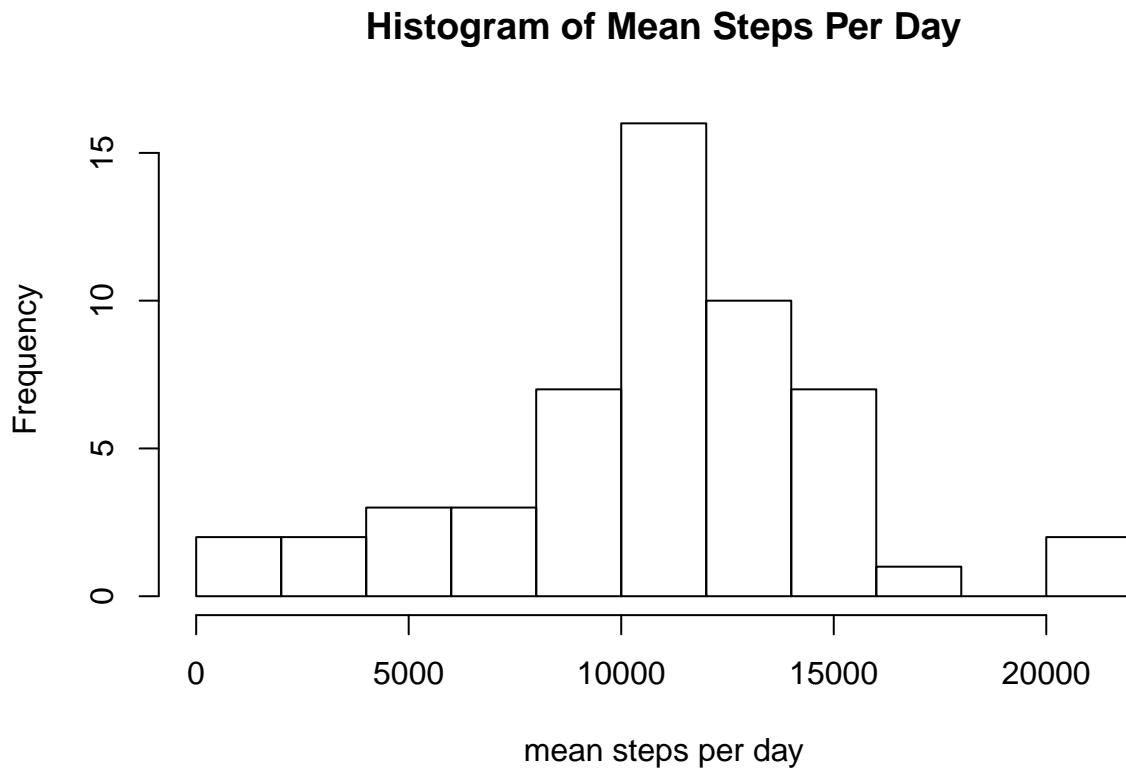
What is mean total number of steps taken per day?

First, we can take a quick view of the data via a histogram. We can see several things from the histogram:

1. The shape is relatively normal
2. Due to normality, we would expect mean and median to be fairly close

3. We would expect the mean and median to fall roughly just over 10,000 steps

```
hist(data[,sum(steps), by=date]$V1, breaks=8,  
      main="Histogram of Mean Steps Per Day",  
      xlab="mean steps per day")
```



Next, we can compare to the actual mean and median for the entire period. As expected, the mean and median are very close and just over 10,000 steps.

```
data[, .(sum = sum(steps)), by=date][,.(median = median(sum, na.rm=T),  
                                          mean = mean(sum, na.rm=T))]
```

```
##      median      mean  
## 1:  10765 10766.19
```

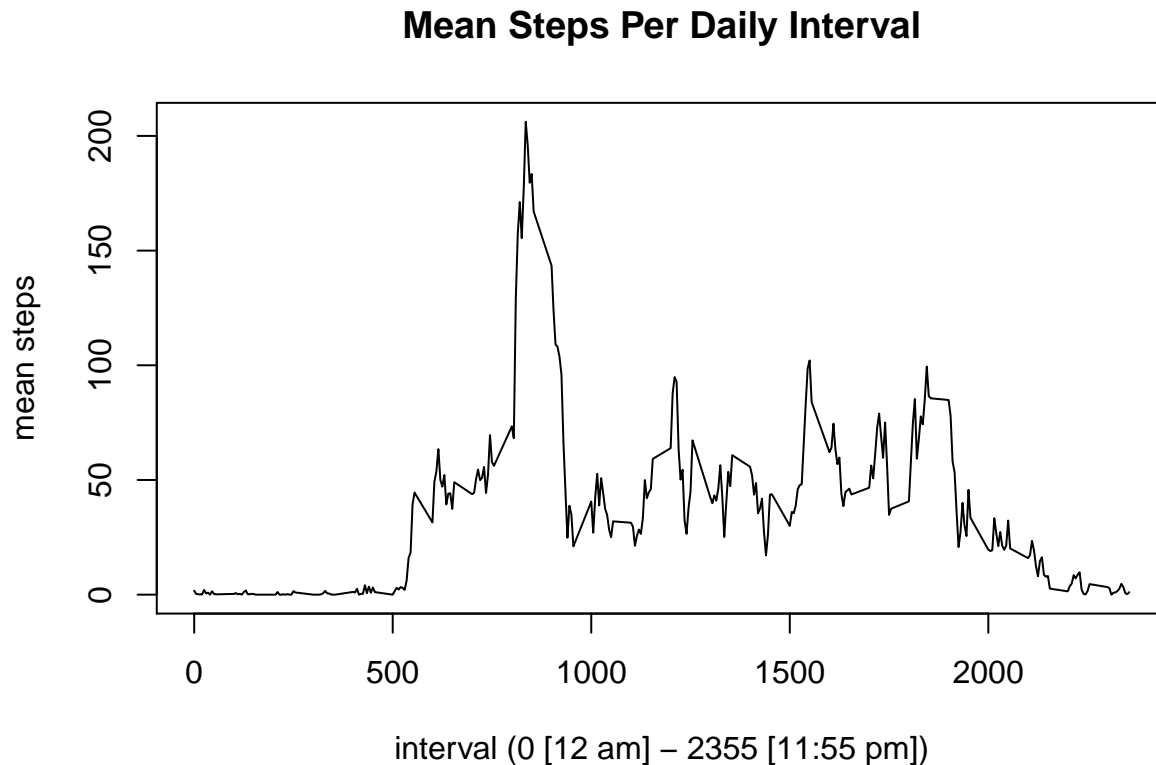
What is the average daily activity pattern?

So what does a typical day look like for this individual? First, we can create a time series based on the mean for each time interval across all days. By plotting this timeseries, we can see qualitatively what a typical day looks like.

In this case, we see that activity begins around 5 am, there is, on average, a large spike in activity in the morning around 8:30 am followed by spikes around noon, mid-afternoon, and then early evening perhaps corresponding to morning exercise, lunch, afternoon break, and returning home. It might be interesting to compare weekend and weekday patterns. And that is what we show below in the last section.

```
ts <- data[, .(mean = round( mean(steps, na.rm=T), 2)), by=interval]

plot(ts$mean ~ ts$interval, type="l", main="Mean Steps Per Daily Interval",
      xlab="interval (0 [12 am] - 2355 [11:55 pm])", ylab="mean steps")
```



Now we can look at specific descriptives like the interval with the average maximum number of steps. As expected from the time series plot, it is 8:35 am.

```
ts[which.max(mean)]
```

```
##      interval    mean
## 1:         835 206.17
```

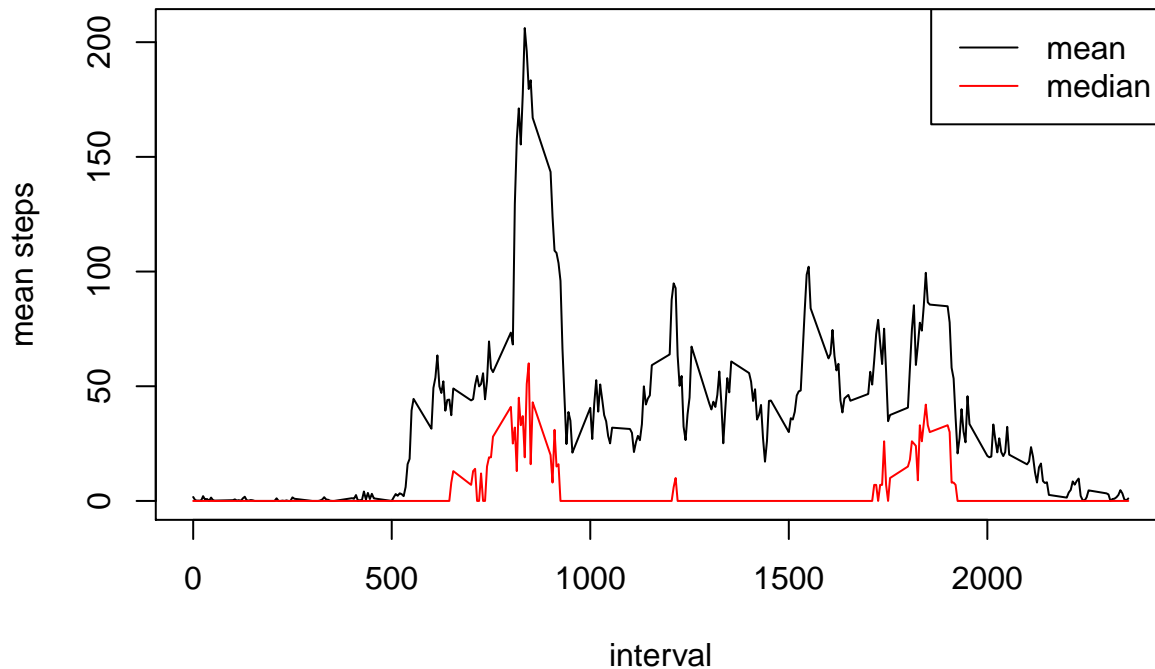
Imputing missing values

Imputing missing values is a tricky but important task. Using listwise deletion is known to be more biased than other methods (add citation). On the other hand, more complex methods will be less biased than replacing with means or medians. For this exercise, the task seems to be an exercise in looking at and thinking about the data and less about using tools for missing data. Based on this thinking, and based on the output below, it seemed using the median might be best. After trying each though, the mean may be the less biased choice. I need to look into this further. So in the end, I am using the mean to impute the missing values. Both are shown below.

```
# Compare mean and median by interval
ts2 <- data[, .(median = median(steps, na.rm=T),
                  mean = mean(steps, na.rm=T)),
              by=interval]
```

```
plot(ts2$mean ~ ts2$interval, type="l", main="Mean Steps Per Daily Interval",
     xlab="interval", ylab="mean steps")
lines(ts2$median ~ ts2$interval, col="red")
legend("topright", legend=c("mean", "median"), lty=1, col=c("black", "red"))
```

Mean Steps Per Daily Interval



```
# Where median shows zero, "random" sample of percentage of zero days
nrow(data[interval == 1000][steps == 0])/nrow(data[!is.na(steps) & interval == 1000])
```

```
## [1] 0.7735849
```

```
nrow(data[interval == 1500][steps == 0])/nrow(data[!is.na(steps) & interval == 1500])
```

```
## [1] 0.7735849
```

```
nrow(data[interval == 2000][steps == 0])/nrow(data[!is.na(steps) & interval == 2000])
```

```
## [1] 0.7169811
```

```
# Compare interval by weekday
data[, .(median = median(steps, na.rm=T),
                        mean = mean(steps, na.rm=T)),
      by=list(interval, wday(date))][interval==900]
```

```
##   interval wday median    mean
## 1:     900    2     71 249.85714
```

```
## 2:      900    3      36 134.00000
## 3:      900    4      14 137.50000
## 4:      900    5       0 137.75000
## 5:      900    6      16 218.14286
## 6:      900    7      20 112.14286
## 7:      900    1      15  19.14286
```

```
ts2[interval == 900]
```

```
##      interval median      mean
## 1:         900      20 143.4528
```

```
nrow(data[interval == 900][steps == 0])/nrow(data[!is.na(steps) & interval == 900])
```

```
## [1] 0.3773585
```

```
# Based on numbers, median might be safer/less biased than mean (but ideally
# would check)
# (of course MI or FIML would be best)
# Could consider making exceptions for days where percentage of zeroes
# is significantly lower than 50%, etc. but not this time....
```

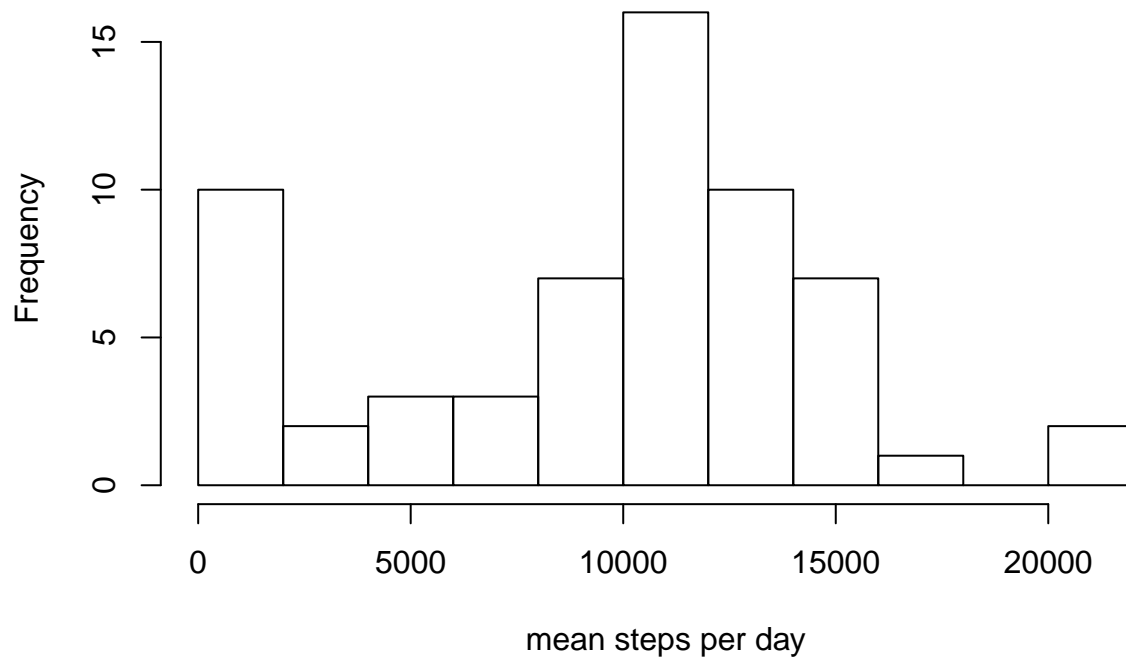
```
#setkey(ts2, interval)
#setkey(data, interval)
```

```
data_imp <- merge(data, ts2, by="interval", all=T)
data_imp[is.na(steps), steps := median]
```

```
##      interval steps      date median      mean
## 1:         0      0 2012-10-01      0 1.716981
## 2:         0      0 2012-10-02      0 1.716981
## 3:         0      0 2012-10-03      0 1.716981
## 4:         0     47 2012-10-04      0 1.716981
## 5:         0      0 2012-10-05      0 1.716981
## ---
## 17564:    2355      0 2012-11-26      0 1.075472
## 17565:    2355      0 2012-11-27      0 1.075472
## 17566:    2355      0 2012-11-28      0 1.075472
## 17567:    2355      0 2012-11-29      0 1.075472
## 17568:    2355      0 2012-11-30      0 1.075472
```

```
hist(data_imp[,sum(steps), by=date]$V1, breaks=8,
      main="Histogram of Mean Steps Per Day",
      xlab="mean steps per day")
```

Histogram of Mean Steps Per Day



```
data_imp[, .(sum = sum(steps)), by=date][,.(median = median(sum, na.rm=T),
  mean = mean(sum, na.rm=T))]
```

```
##      median      mean
## 1: 10395 9503.869
```

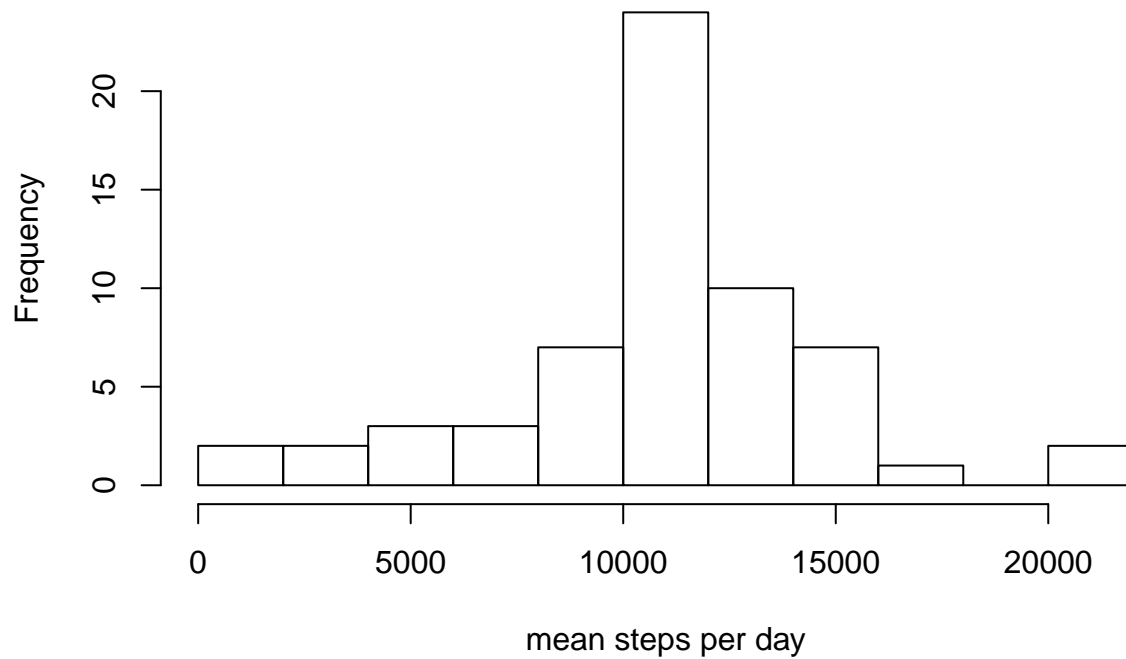
Looks a little "funny", not sure this is OK. Let's try mean.

```
data_imp_mean <- merge(data, ts2, by="interval", all=T)
data_imp_mean[is.na(steps), steps := mean]
```

```
##      interval      steps      date median      mean
## 1:         0 1.716981 2012-10-01      0 1.716981
## 2:         0 0.000000 2012-10-02      0 1.716981
## 3:         0 0.000000 2012-10-03      0 1.716981
## 4:         0 47.000000 2012-10-04      0 1.716981
## 5:         0 0.000000 2012-10-05      0 1.716981
## ---
## 17564: 2355 0.000000 2012-11-26      0 1.075472
## 17565: 2355 0.000000 2012-11-27      0 1.075472
## 17566: 2355 0.000000 2012-11-28      0 1.075472
## 17567: 2355 0.000000 2012-11-29      0 1.075472
## 17568: 2355 1.075472 2012-11-30      0 1.075472
```

```
hist(data_imp_mean[,sum(steps), by=date]$V1, breaks=8,
  main="Histogram of Mean Steps Per Day",
  xlab="mean steps per day")
```

Histogram of Mean Steps Per Day



```
data_imp_mean[, .(sum = sum(steps)), by=date][,.(median = median(sum, na.rm=T),
    mean = mean(sum, na.rm=T))]
```

```
##      median      mean
## 1: 10766.19 10766.19
```

```
data_imp_mean[which.max(interval)]
```

```
##   interval  steps      date median      mean
## 1:      2355 1.075472 2012-10-01      0 1.075472
```

Are there differences in activity patterns between weekdays and weekends?

Differences between weekday and weekend activity as shown in the plot below include a slower rise in activity on weekend mornings, as well as higher activity variability throughout the day and evening.

```
# add weekday factor
data_imp_mean[, weekday := ifelse(wday(date) %in% c(2:6), "weekday", "weekend")]
```

```
##      interval  steps      date median      mean weekday
## 1:          0 1.716981 2012-10-01      0 1.716981 weekday
## 2:          0 0.000000 2012-10-02      0 1.716981 weekday
## 3:          0 0.000000 2012-10-03      0 1.716981 weekday
## 4:          0 47.000000 2012-10-04      0 1.716981 weekday
## 5:          0 0.000000 2012-10-05      0 1.716981 weekday
## ---
```



```
## 17564:      2355  0.000000 2012-11-26      0 1.075472 weekday
## 17565:      2355  0.000000 2012-11-27      0 1.075472 weekday
## 17566:      2355  0.000000 2012-11-28      0 1.075472 weekday
## 17567:      2355  0.000000 2012-11-29      0 1.075472 weekday
## 17568:      2355  1.075472 2012-11-30      0 1.075472 weekday
```

```
nrow(data_imp_mean[weekday == "weekday"])
```

```
## [1] 12960
```

```
ts_imp_mean <- data_imp_mean[, .(median = median(steps),
                                mean = mean(steps),
                                weekday),
                               by=list(weekday,interval)]
```

```
xyplot(data=ts_imp_mean,
       mean ~ interval | factor(weekday),
       type="l",
       layout=c(1,2))
```

