Reproducible Research: Peer Assessment 1

Load required library

```
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
```

Loading and preprocessing the data

- Load the concerned dataset into ‘activity’
- Convert ‘date’ into R ‘date’ format (observe the difference between activity$date datatype before & after conversion)

```
activity<-read.csv("./activity.csv", header = TRUE)
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 ...
```

```
activity[,2] <- as.Date(activity$date)
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 ...
```

- dataset size is 17568 x 3

- date:
  - 61 days,
  - start date on 2012-10-01 (Mon),
  - end date on 2012-11-30 (Fri),
  - 8 weekends,
  - 288 observations on each day
- interval:
  - minutes value at the interval of 5 mins
  - but minutes value at n*60 is surprisingly n*100
  - minutes value resets to 0 on each new day

```
# dataset size is 17568 x 3
dim(activity)
```

```
## [1] 17568     3
```

```
# 61 consecutive days
length(unique(activity$date))
```

```
## [1] 61
```

```
# 288 observations on each day
head(table(activity$date))
```

```
##
## 2012-10-01 2012-10-02 2012-10-03 2012-10-04 2012-10-05 2012-10-06
##        288        288        288        288        288        288
```

```
# consecutive difference between interval is mostly 5 except at n*60 & on ne
w day
table(diff(activity$interval, 1))
```

```
##
## -2355     5    45
##    60 16104  1403
```

Imputing missing values (moving this question above as this is crucial in identifying no-activity-days)

- NA is found in 'steps' variable only
  - total count of NA is 2304 (8 days of no activity * 288 observations per day)
  - there was no activity captured on 8 of the 61 days

- 3 options (1. remove those observations, 2. replace NA with 0, 3. replace NA with mean of valid ‘steps’ in all ‘interval’ across valid days)
- Going ahead with 2nd option (reapace NA with 0)

```
# NA count in 'steps' column is 2304
sum(is.na(activity$steps))
```

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

```
sum(is.na(activity))
```

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

```
# NA count on each of the days
statsNA <- activity %>% group_by(date) %>% summarise (daysStepsNA = sum(is.n
a(steps)))
unique(statsNA$daysStepsNA)
```

```
## [1] 288   0
```

```
# dates on which there was no activity
statsNA$date[which(statsNA$daysStepsNA == 288)]
```

```
## [1] "2012-10-01" "2012-10-08" "2012-11-01" "2012-11-04" "2012-11-09"
## [6] "2012-11-10" "2012-11-14" "2012-11-30"
```

```
# Make a new dataset by replacing all NA with 0
activity_NAs_Zero <- activity
activity_NAs_Zero[is.na(activity_NAs_Zero)] <- 0
```

What is mean total number of steps taken per day?

- Calculate the total number of steps taken per day
  - Both calculations below should give same result

```
stepsCount <- with(activity, tapply(steps, date, sum, na.rm=TRUE))
stepsCount_NAs_Zero <- with(activity_NAs_Zero, tapply(steps, date, sum))
```

- Make a histogram of the total number of steps taken each day
  - Min & max steps count are 41 & 21194
  - Histogram of steps count on each days with 100 bins which shows 10 instances of near-zero values which is due to 8 days of no-activity and 2 small values (41 & 126 are small w.r.t. max 21194)

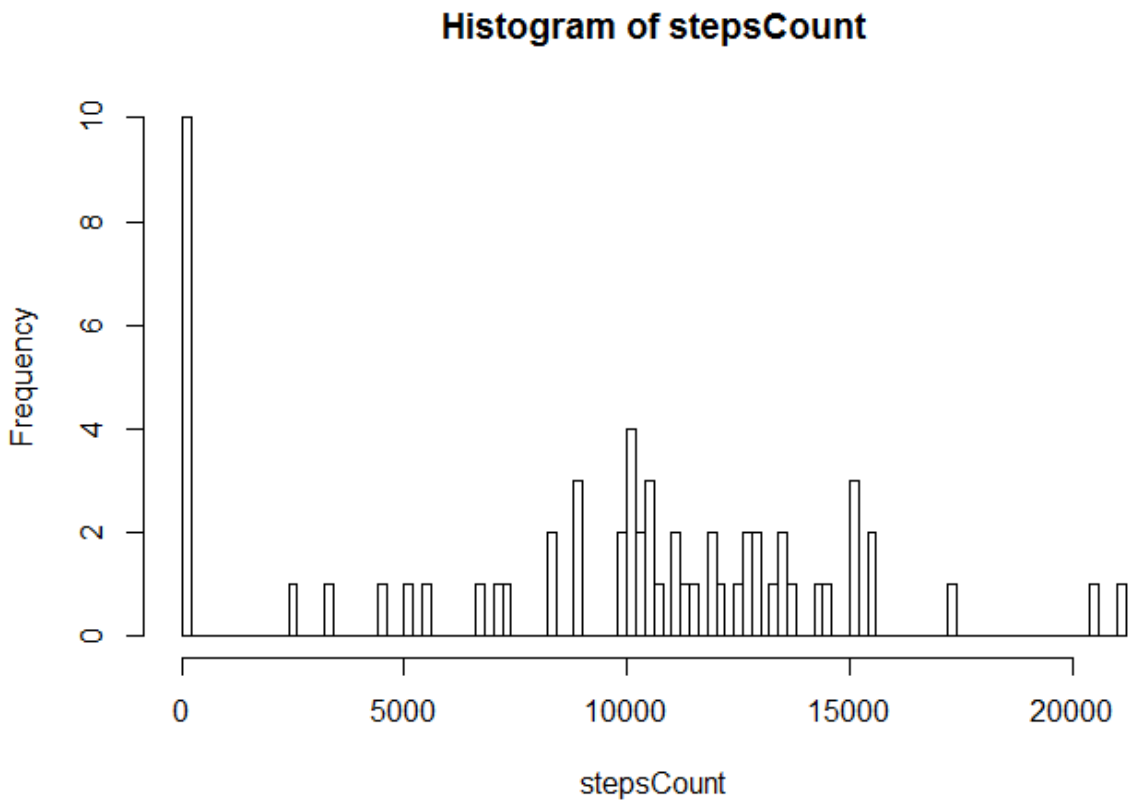
```
# min & max steps count
min(stepsCount)
```

```
## [1] 0
```

```
max(stepsCount)
```

```
## [1] 21194
```

```
# Hist plot with 100 bins
hist(stepsCount, breaks=100)
```



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

```
# mean & median of daily steps count incuding no activity days
mean(stepsCount)
```

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

```
median(stepsCount)
```

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

What is the average daily activity pattern?

- Make a time series plot (type = “l”) of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis), i.e., plot average of steps count in each intervals
- average daily activity in each interval
    - Both steps mean have been plotted together

```
# Ignore those 8 no acitivity days
stepsSplit <- split(activity$steps, activity$date)
stepsSplit = data.frame(stepsSplit)
dim(stepsSplit)
```

```
## [1] 288  61
```

```
stepsMean = apply(stepsSplit, 1, mean, na.rm=TRUE)

# Plot steps mean in each interval across days
plot(activity$interval[1:288], stepsMean, type = "l", col="red", xlab = "Int
ervals", ylab = "Average Steps", main = "Steps mean in intervals over days
(original -vs- imputed)")

# Include those 8 no activity days using imputed data
# "na.rm=TRUE" is redundant in this case
stepsSplit_NAs_Zero <- split(activity_NAs_Zero$steps, activity_NAs_Zero$dat
e)
stepsSplit_NAs_Zero = data.frame(stepsSplit_NAs_Zero)
dim(stepsSplit_NAs_Zero)
```

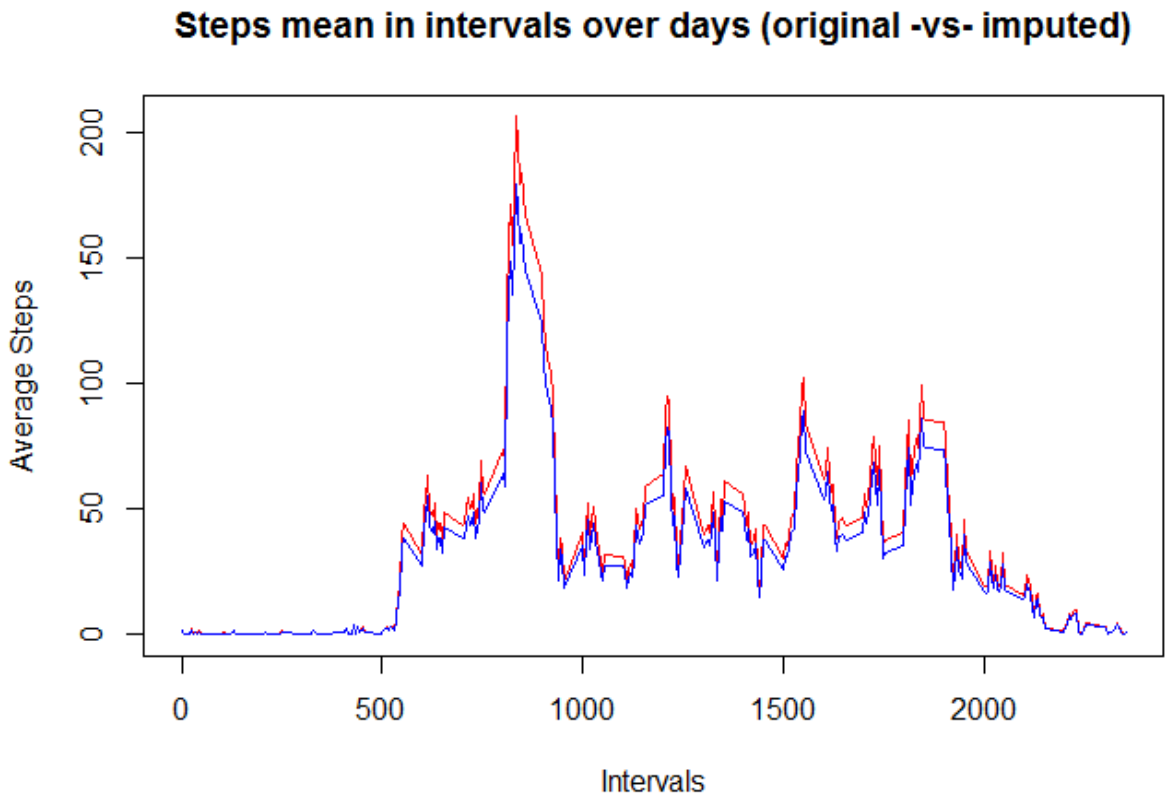
```
## [1] 288  61
```

```
stepsMean_NAs_Zero = apply(stepsSplit_NAs_Zero, 1, mean, na.rm=TRUE)

# Compare these means
head(data.frame(stepsMean, stepsMean_NAs_Zero))
```

```
##   stepsMean stepsMean_NAs_Zero
## 1 1.7169811         1.49180328
## 2 0.3396226         0.29508197
## 3 0.1320755         0.11475410
## 4 0.1509434         0.13114754
## 5 0.0754717         0.06557377
## 6 2.0943396         1.81967213
```

```
# Plot steps mean in each interval across days for imputed dataset
lines(activity$interval[1:288], stepsMean_NAs_Zero, col="blue")
```



Are there differences in activity patterns between weekdays and weekends?

- Split dataset into weekday (53 days) & weekend (8 days)

```
# Mutate dataset with a new column for 'day'
activity_NAs_Zero <- mutate(activity_NAs_Zero, day = ifelse(weekdays(activit
y_NAs_Zero$date) == "Saturday" | weekdays(activity_NAs_Zero$date) == "Sunda
y", "weekend", "weekday"))
activity_NAs_Zero$day <- as.factor(activity_NAs_Zero$day)
str(activity_NAs_Zero)
```

```
## 'data.frame':    17568 obs. of  4 variables:
##  $ steps   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ date    : Date, format: "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
##  $ day     : Factor w/ 2 levels "weekday","weekend": 1 1 1 1 1 1 1 1 1
1 ...
```

```
#  Split dataset into weekday (53 days) & weekend (8 days)
activityWeekend <- subset(activity_NAs_Zero, as.character(activity_NAs_Zero
$day) == "weekend")
activityWeekday <- subset(activity_NAs_Zero, as.character(activity_NAs_Zero
$day) == "weekday")

stepsSplitWeekday <- split(activityWeekday$steps, activityWeekday$date)
stepsSplitWeekday = data.frame(stepsSplitWeekday)
dim(stepsSplitWeekday)
```

```
## [1] 288  45
```

```
stepsMeanWeekday = apply(stepsSplitWeekday, 1, mean, na.rm=TRUE)

stepsSplitWeekend <- split(activityWeekend$steps, activityWeekend$date)
stepsSplitWeekend = data.frame(stepsSplitWeekend)
dim(stepsSplitWeekend)
```

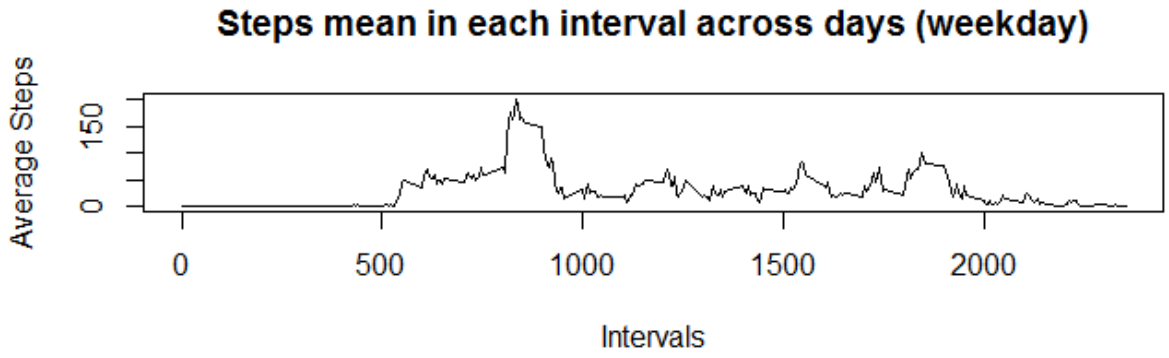
```
## [1] 288  16
```

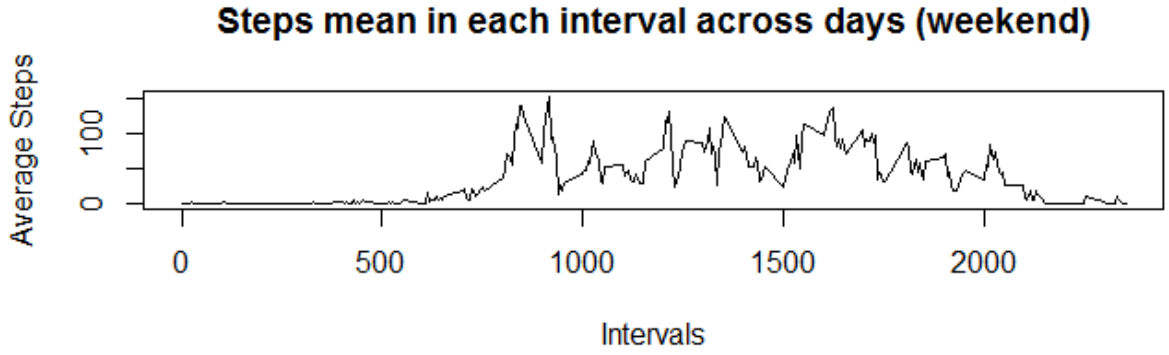
```
stepsMeanWeekend = apply(stepsSplitWeekend, 1, mean, na.rm=TRUE)

par(mfrow=c(2,1))

# Plot steps mean in each interval across days (weekday)
plot(activity$interval[1:288], stepsMeanWeekday, type = "l", xlab = "Interva
ls", ylab = "Average Steps", main = "Steps mean in each interval across day
s (weekday)")

# Plot steps mean in each interval across days (weekend)
plot(activity$interval[1:288], stepsMeanWeekend, type = "l", xlab = "Interva
ls", ylab = "Average Steps", main = "Steps mean in each interval across day
s (weekend)")
```


Steps mean in each interval across days (weekday)
Average Steps
0
150
0
500
1000
1500
2000
Intervals


Steps mean in each interval across days (weekend)
Average Steps
0
100
0
500
1000
1500
2000
Intervals