Reproducible Research: Peer Assessment 1

Stanislav Gerasymenko

Loading and preprocessing the data

Load required packages:

```
require(dplyr)
require(ggplot2)
require(gridExtra)
```

Set the working directory (**should be customised by the user**) and read the data into R:

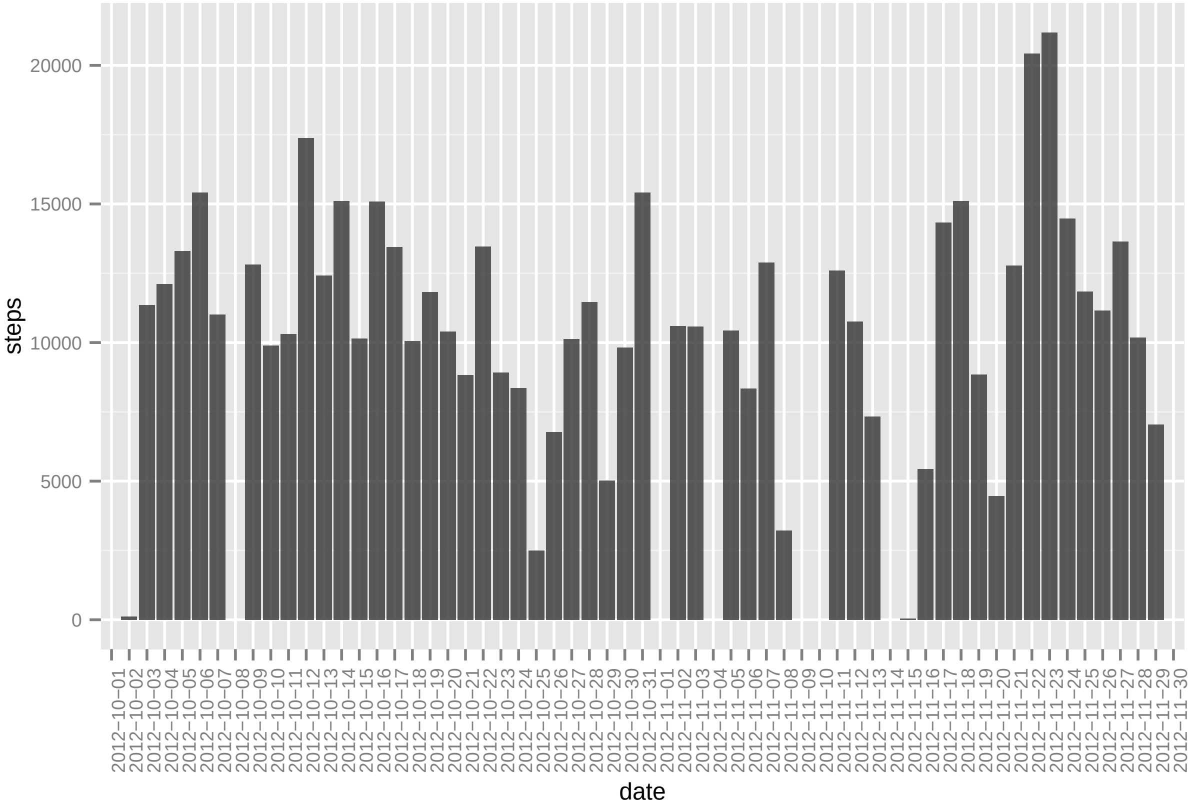
```
wd <- paste("C:/!SG/OneDrive/Projects/SCIENCE/Coursera",
            "Coursera_05_Reproducible_Research/Course_Project1",
            "RepData_PeerAssessment1", sep = "/")
setwd(paste(wd,"INPUT",sep="/"))
inputdf <-
        read.csv("activity.csv")%>%
        mutate(interval = as.factor(interval))
# head(inputdf)
# str(inputdf)
# tail(inputdf)
```

What is mean total number of steps taken per day?

Calculate and graph the total steps walked during each day:

```
totalStepsByDay <-
        inputdf %>%
        group_by(date) %>%
        summarise(steps = sum(steps))
# totalStepsByDay
```

```
## Warning: Removed 8 rows containing missing values (position_stack).
```



Median value of steps walked during a day is:

```
totalStepsByDay %>%
        summarise(steps = median(steps, na.rm = TRUE))
```

```
## Source: local data frame [1 x 1]
##
##   steps
## 1 10765
```

Mean value of steps walked during a day is:

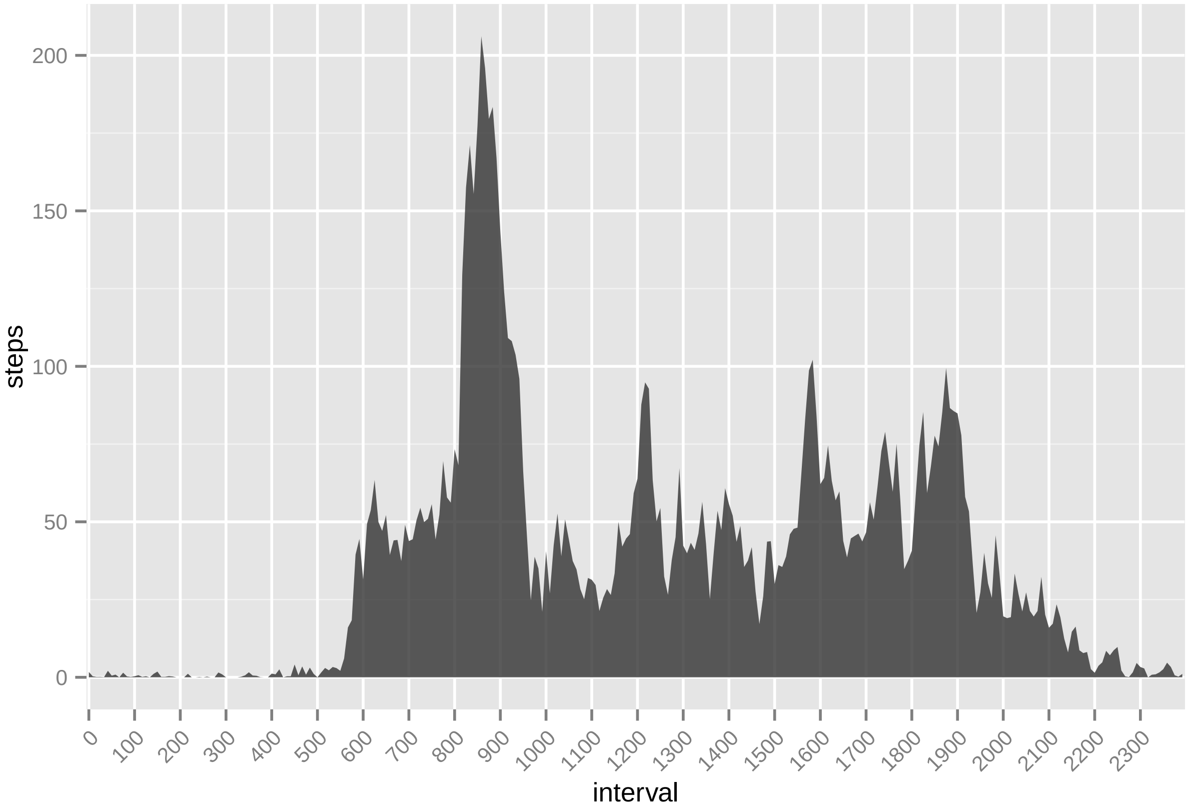
```
totalStepsByDay %>%
        summarise(steps = mean(steps, na.rm = TRUE))
```

```
## Source: local data frame [1 x 1]
##
##      steps
## 1 10766.19
```

What is the average daily activity pattern?

Calculate and plot the mean steps walked during each interval over the whole period:

```
totalStepsByInterval <-
        inputdf %>%
        mutate(interval = as.factor(interval)) %>%
        group_by(interval) %>%
        summarise(steps = mean(steps, na.rm = TRUE))
# str(totalStepsByInterval)
# tail(totalStepsByInterval)
```



Find the interval with maximum steps walked on average:

```
totalStepsByInterval %>%
        filter(steps == max(steps))
```

```
## Source: local data frame [1 x 2]
##
##   interval    steps
## 1      835 206.1698
```

Imputing missing values

Calculate the number of missing values before imputing the missing values:

```
inputdf %>%
        filter(is.na(steps)) %>%
        summarise(count = n())
```

```
##   count
## 1  2304
```

Impute the missing values with average values for the corresponding interval:

```
temp <- left_join(inputdf, totalStepsByInterval, by = "interval")
inputdfCorrected <- inputdf
inputdfCorrected[is.na(inputdf$steps), "steps"] <-
        temp[is.na(inputdf$steps), "steps.y"]
rm(temp)
```

Calculate the number of missing values after imputing missing values:

```
inputdfCorrected %>%
        filter(is.na(steps)) %>%
        summarise(count = n())
```

```
##   count
## 1     0
```

Calculate and graph the total steps walked during each day after imputing the missing values:

```
totalStepsByDayCorrected <-
        inputdfCorrected %>%
        group_by(date) %>%
        summarise(steps = sum(steps))
# totalStepsByDay
```

Median value of steps walked during a day after imputing the missing values is:

```
totalStepsByDayCorrected %>%
        summarise(steps = median(steps, na.rm = TRUE))
```

```
## Source: local data frame [1 x 1]
##
##      steps
## 1 10766.19
```

Mean value of steps walked during a day after imputing the missing values is:

```
totalStepsByDayCorrected %>%
        summarise(steps = mean(steps, na.rm = TRUE))
```

```
## Source: local data frame [1 x 1]
##
##      steps
## 1 10766.19
```

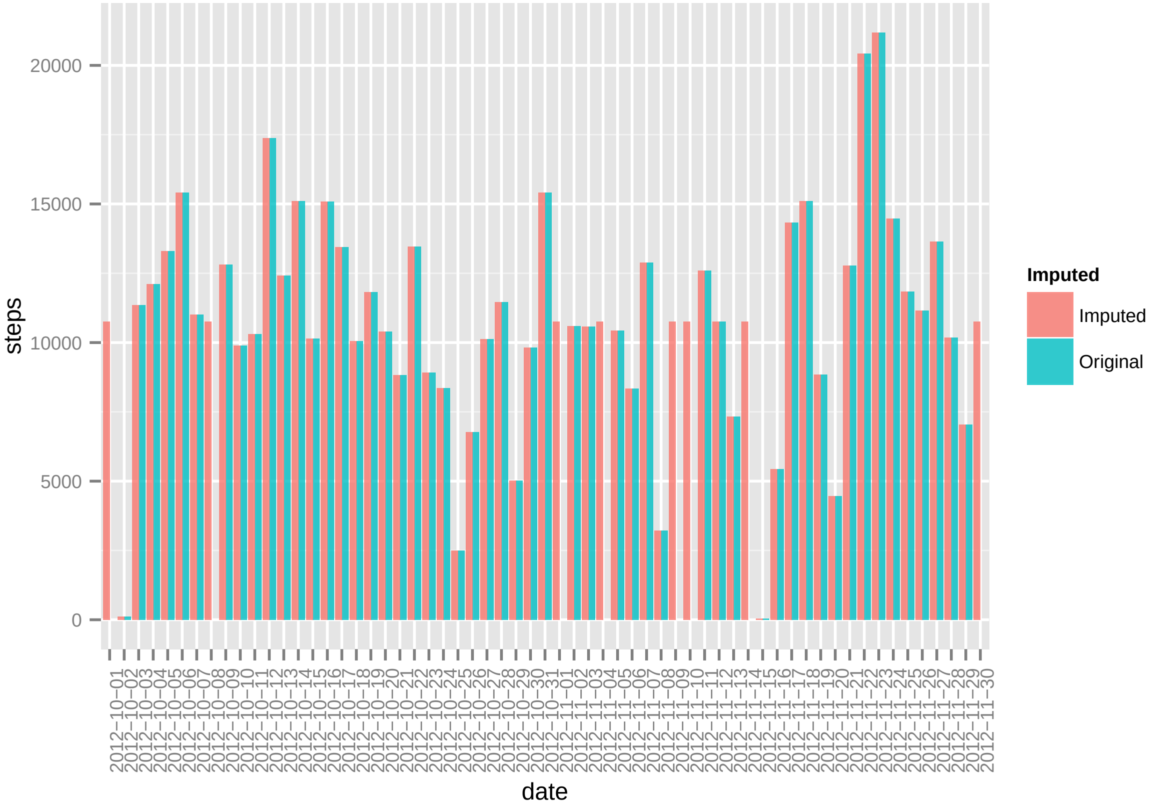
Calculate and plot the mean steps walked during each interval over the whole period after imputing missing values:

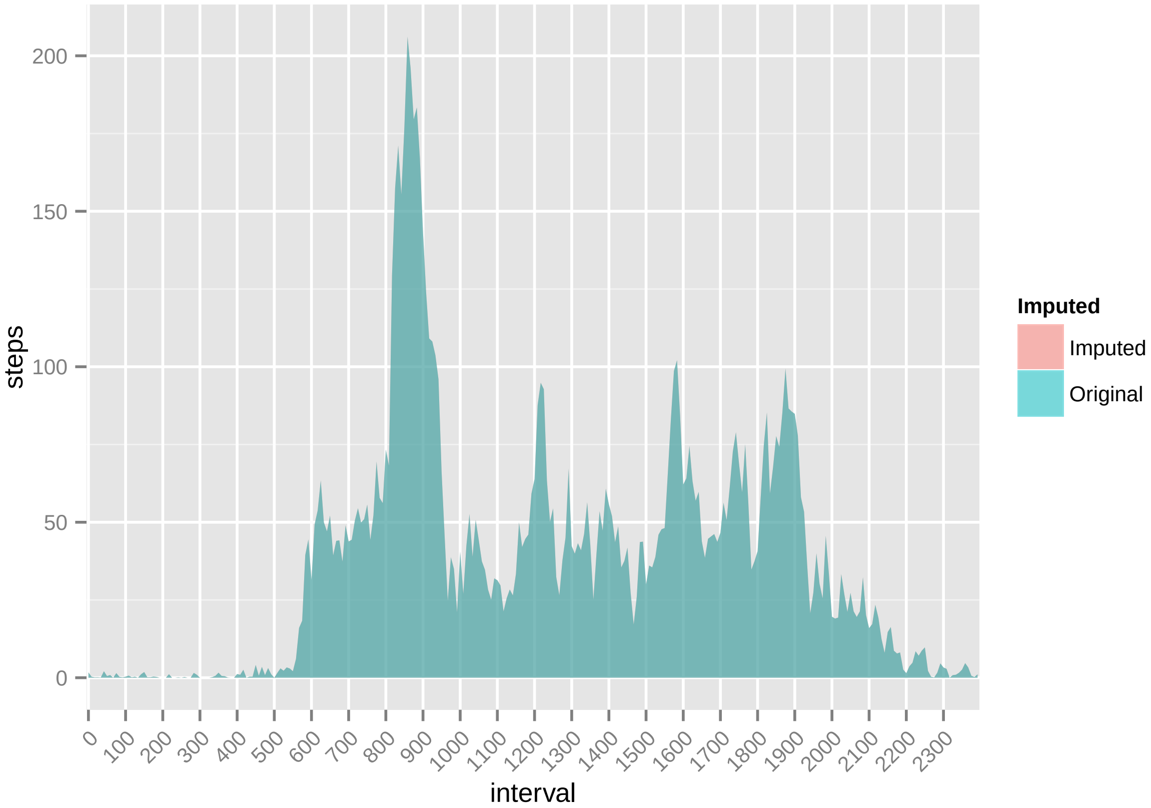
```
totalStepsByIntervalCorrected <-
        inputdf %>%
        mutate(interval = as.factor(interval)) %>%
        group_by(interval) %>%
        summarise(steps = mean(steps, na.rm = TRUE))
# str(totalStepsByIntervalCorrected)
# tail(totalStepsByIntervalCorrected)
```

Join original and imputed tables for convenient graphing in one plot and see the comparison on the plot:

```
totalStepsByDayComparison <- rbind(totalStepsByDay,totalStepsByDayCorrected)
totalStepsByDayComparison$Imputed <-
        rep(c("Original", "Imputed"),
            times = c(nrow(totalStepsByDay), nrow(totalStepsByDayCorrected)))
totalStepsByDayComparison$Imputed <-
        as.factor(totalStepsByDayComparison$Imputed)

totalStepsByIntervalComparison <-
        rbind(totalStepsByInterval, totalStepsByIntervalCorrected)
totalStepsByIntervalComparison$Imputed <- rep(c("Original", "Imputed"),
                           times = c(nrow(totalStepsByInterval),
                                     nrow(totalStepsByIntervalCorrected)))
totalStepsByIntervalComparison$Imputed <-
        as.factor(totalStepsByIntervalComparison$Imputed)
```


steps
20000
15000
10000
5000
0
2012-10-01
2012-10-02
2012-10-03
2012-10-04
2012-10-05
2012-10-06
2012-10-07
2012-10-08
2012-10-09
2012-10-10
2012-10-11
2012-10-12
2012-10-13
2012-10-14
2012-10-15
2012-10-16
2012-10-17
2012-10-18
2012-10-19
2012-10-20
2012-10-21
2012-10-22
2012-10-23
2012-10-24
2012-10-25
2012-10-26
2012-10-27
2012-10-28
2012-10-29
2012-10-30
2012-10-31
2012-11-01
2012-11-02
2012-11-03
2012-11-04
2012-11-05
2012-11-06
2012-11-07
2012-11-08
2012-11-09
2012-11-10
2012-11-11
2012-11-12
2012-11-13
2012-11-14
2012-11-15
2012-11-16
2012-11-17
2012-11-18
2012-11-19
2012-11-20
2012-11-21
2012-11-22
2012-11-23
2012-11-24
2012-11-25
2012-11-26
2012-11-27
2012-11-28
2012-11-29
2012-11-30
date
Imputed
Imputed
Original


steps
200
150
100
50
0
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
interval
Imputed
Imputed
Original

Are there differences in activity patterns between weekdays and weekends?

Create a factor with values WorkingDay/Weekend:

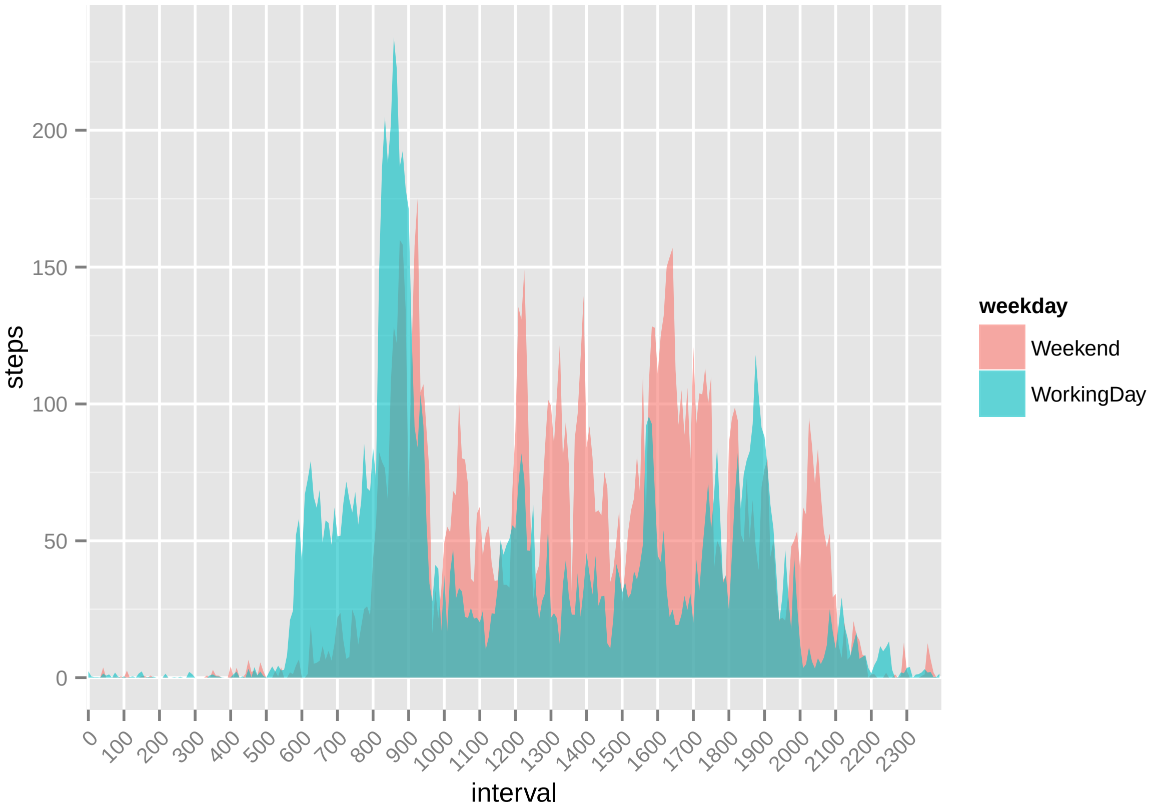
```
Sys.setlocale("LC_TIME", "C")
```

```
## [1] "C"
```

```
inputdfCorrected <-
        inputdf %>%
        mutate(weekday = weekdays(as.Date(date))) %>%
        mutate(weekday = ifelse(weekday == "Monday" |
                                        weekday ==  "Tuesday" |
                                        weekday ==  "Wednesday" |
                                        weekday ==  "Thursday" |
                                        weekday ==  "Friday",
                                "WorkingDay", "Weekend")) %>%
        mutate(weekday = as.factor(weekday))
# str(inputdfCorrected)
```

Calculate and plot the mean steps walked during each interval over WorkingDay/ Weekend periods after imputing missing values:

```
totalStepsByIntervalWeekdayCorrected <-
        inputdfCorrected %>%
        mutate(interval = as.factor(interval)) %>%
        group_by(interval, weekday) %>%
        summarise(steps = mean(steps, na.rm = TRUE))
# str(totalStepsByInterval)
# tail(totalStepsByInterval)
```


steps
200
150
100
50
0
interval
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
weekday
Weekend
WorkingDay