Reproducible Research: Peer Assessment 1

Loading and preprocessing the data

First, we load the data and look at the data structure and formats.

```
dane0<-read.csv('activity.csv')
str(dane0)
```

```
## '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 ...
```

We change the format of *date* from factor to date.

```
dane0$date<-as.Date(as.character(dane0$date))
str(dane0)
```

```
## '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 ...
```

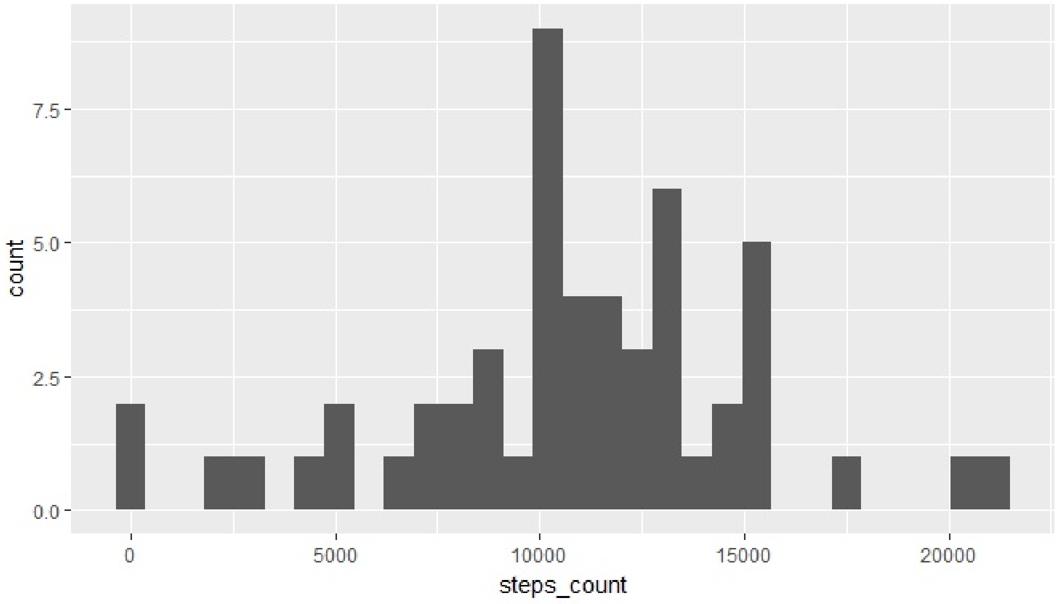
What is mean total number of steps taken per day?

First, we aggregate the data using functions from dplyr package.

```
library(dplyr)
stepsPerDay<-dane0 %>% group_by(date) %>% summarise(steps_count=sum(steps))
```

In order to make histogram, we use ggplot2 package.

```
library(ggplot2)
g<-ggplot(stepsPerDay, aes(x=steps_count))+geom_histogram()
plot(g)
```



To report mean and median, we may use summary function.

```
summary(stepsPerDay$steps_count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##      41    8841   10765   10766   13294   21194       8
```

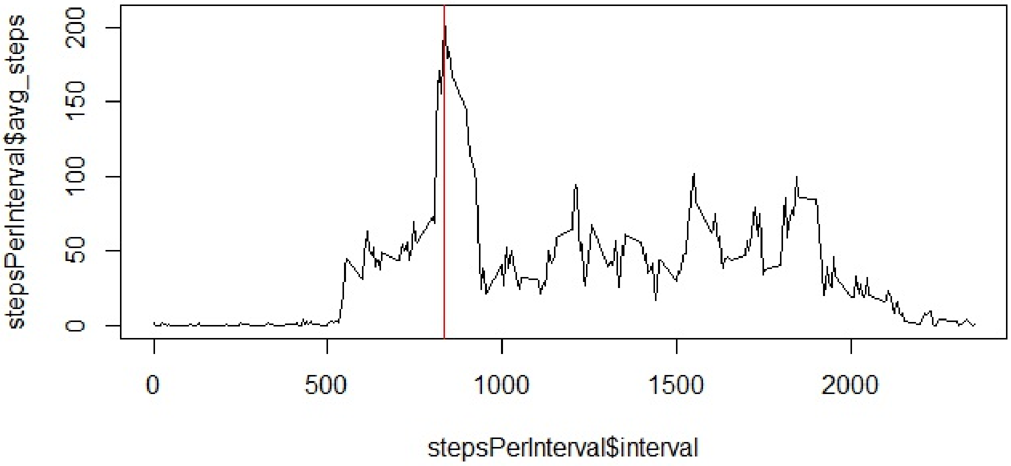
What is the average daily activity pattern?

First, we aggregate the date using dplyr and then make a plot with base graphics.

```
stepsPerInterval<-dane0 %>% group_by(interval) %>% summarise(avg_steps=mean(steps, na.rm=TRUE))
interval_max_steps<-stepsPerInterval[which.max(stepsPerInterval$avg_steps),1]
interval_max_steps
```

```
## # A tibble: 1 x 1
##   interval
##      <int>
## 1      835
```

```
plot(x=stepsPerInterval$interval, y=stepsPerInterval$avg_steps, type="l")
abline(v=stepsPerInterval[which.max(stepsPerInterval$avg_steps),1], type="l", col="red")
```



The interval with highest average number of steps is 835- this is indicated on the plot with red vertical line.

Imputing missing values

First, it is worth checking how many missing values we have in our dataset

```
sapply(dane0, function(x) sum(is.na(x)))
```

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

```
sapply(dane0, function(x) sum(is.na(x))/nrow(dane0))
```

```
##     steps      date  interval 
## 0.1311475 0.0000000 0.0000000
```

There are about 13% of NA values in *steps* variable. As the average number of steps varies significantly between intervals, we will impute missing values with average number of steps per interval.

```
dane1<-merge(dane0, stepsPerInterval, by="interval")
dane1$steps<-ifelse(is.na(dane1$steps), dane1$avg_steps, dane1$steps)
```

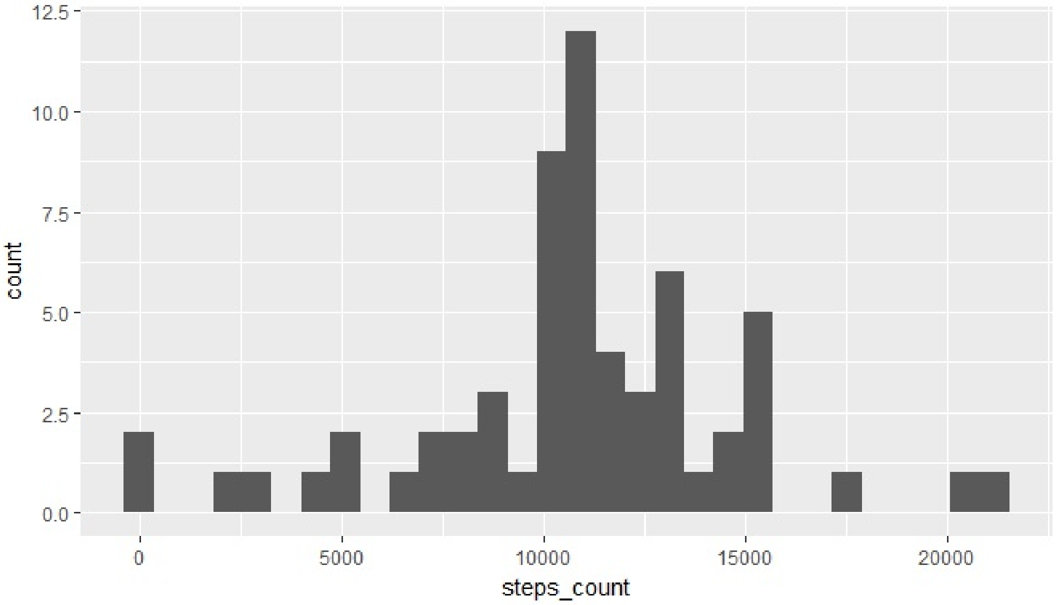
We compare the impact of imputation looking at the statistics for steps count per day.

```
stepsPerDay1<-dane1 %>% group_by(date) %>% summarise(steps_count=sum(steps))
summary(stepsPerDay1$steps_count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##      41    9819   10766   10766   12811   21194
```

The quantiles slightly differ as we have imputed values for 8 days that were initially missing. As we have filled in missing values with average values for each interval, the global mean has not changed- in each of the 8 days that were missing there are average values observed for the remaining days. The histogram of the total number of steps taken each day is presented below.

```
g2<-ggplot(stepsPerDay1, aes(x=steps_count))+geom_histogram()
plot(g2)
```



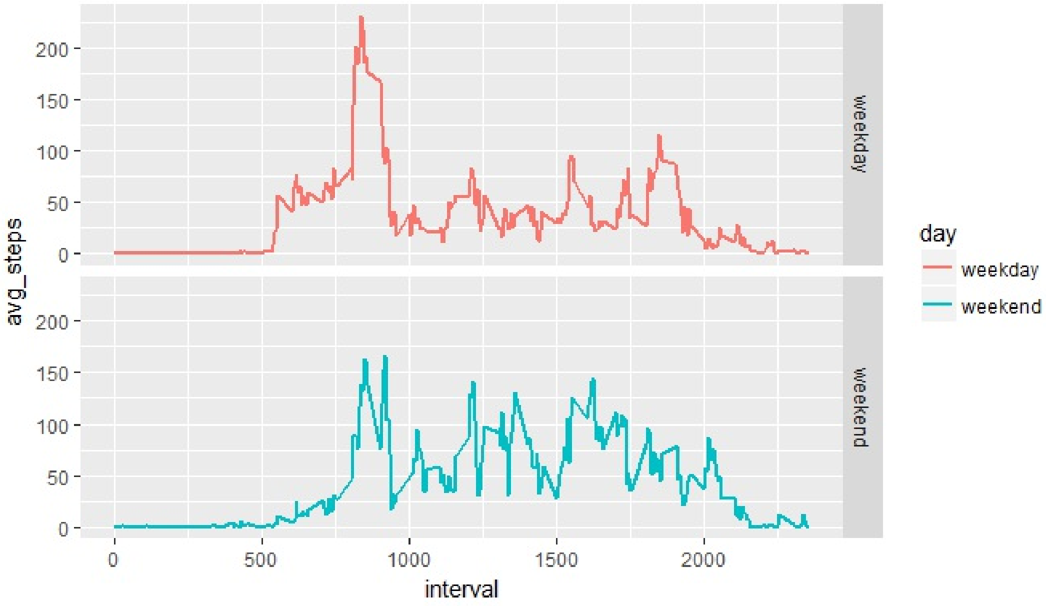
Are there differences in activity patterns between weekdays and weekends?

First, we create new factor variable indicating whether a given date is a weekday or weekend day.

```
dane1$day<-weekdays(dane1$date)
dane1$day<-ifelse(dane1$day %in% c("sobota", "niedziela"), "weekend", "weekday")
dane1$day<-as.factor(dane1$day)
```

In order to make a plot, we first aggregate data using dplyr and then utilise ggplot2.

```
stepsPerWeekday<- dane1 %>% group_by(interval,day) %>% summarise(avg_steps=mean(steps))
g3<-ggplot(stepsPerWeekday, aes(x=interval, y=avg_steps, col=day))+geom_line(lwd=1)+facet_grid(day~.)
plot(g3)
```



It seems that activity begins later on weekend than on weekdays. On weekdays there is a substantial peak around interval 835, while on weekend activity is ditributed more uniformly.