# Reproducible Research: Peer Assessment 1

## Load and process the data

##### 1. Load the data

{r, echo=FALSE, results='hide', warning=FALSE, message=FALSE} library(ggplot2) library(scales) library(Hmisc)

{r, results='markup', warning=TRUE, message=TRUE} if(!file.exists('activity.csv')){ unzip('activity.zip') } activityData <- read.csv('activity.csv') ##### 2. transform interval data

#activityData$interval <- strptime(gsub("([0-9]{1,2})([0-9]{2})", "\\1:\\2", activityData$interval), format='%H:%M')

## mean total number of steps taken per day.

stepsByDay <- tapply(activityData$steps, activityData$date, sum, na.rm=TRUE)

##### 1. Histogram of the total number of steps taken each day

qplot(stepsByDay, xlab='Total steps per day', ylab='Frequency using binwith 500', binwidth=500)

##### 2. Mean and median total number of steps taken per day

stepsByDayMean <- mean(stepsByDay)  
stepsByDayMedian <- median(stepsByDay)

* Mean: r stepsByDayMean
* Median: r stepsByDayMedian

## Average daily activity pattern?

averageStepsPerTimeBlock <- aggregate(x=list(meanSteps=activityData$steps), by=list(interval=activityData$interval), FUN=mean, na.rm=TRUE)

##### 1. Time series plot

ggplot(data=averageStepsPerTimeBlock, aes(x=interval, y=meanSteps)) +  
 geom\_line() +  
 xlab("5-minute interval") +  
 ylab("average number of steps taken")

##### 2. The 5-minute interval on average across all the days in the dataset that contains the maximum number of steps?

mostSteps <- which.max(averageStepsPerTimeBlock$meanSteps)  
timeMostSteps <- gsub("([0-9]{1,2})([0-9]{2})", "\\1:\\2", averageStepsPerTimeBlock[mostSteps,'interval'])

* Most Steps at: r timeMostSteps

## Missing values

##### 1. The total number of missing values in the dataset

numMissingValues <- length(which(is.na(activityData$steps)))

* Number of missing values: r numMissingValues

##### 2. Devise a strategy for filling in all of the missing values in the dataset.

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

activityDataImputed <- activityData  
activityDataImputed$steps <- impute(activityData$steps, fun=mean)

##### 4. Histogram of the total number of steps taken each day

stepsByDayImputed <- tapply(activityDataImputed$steps, activityDataImputed$date, sum)  
qplot(stepsByDayImputed, xlab='Total steps per day (Imputed)', ylab='Frequency using binwith 500', binwidth=500)

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

stepsByDayMeanImputed <- mean(stepsByDayImputed)  
stepsByDayMedianImputed <- median(stepsByDayImputed)

* Mean (Imputed): r stepsByDayMeanImputed
* Median (Imputed): r stepsByDayMedianImputed

## Are there differences in activity patterns between weekdays and weekends?

##### 1. 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.

activityDataImputed$dateType <- ifelse(as.POSIXlt(activityDataImputed$date)$wday %in% c(0,6), 'weekend', 'weekday')

##### 2. Panel plot containing a time series plot

averagedActivityDataImputed <- aggregate(steps ~ interval + dateType, data=activityDataImputed, mean)  
ggplot(averagedActivityDataImputed, aes(interval, steps)) +   
 geom\_line() +   
 facet\_grid(dateType ~ .) +  
 xlab("5-minute interval") +   
 ylab("avarage number of steps")