

## A1

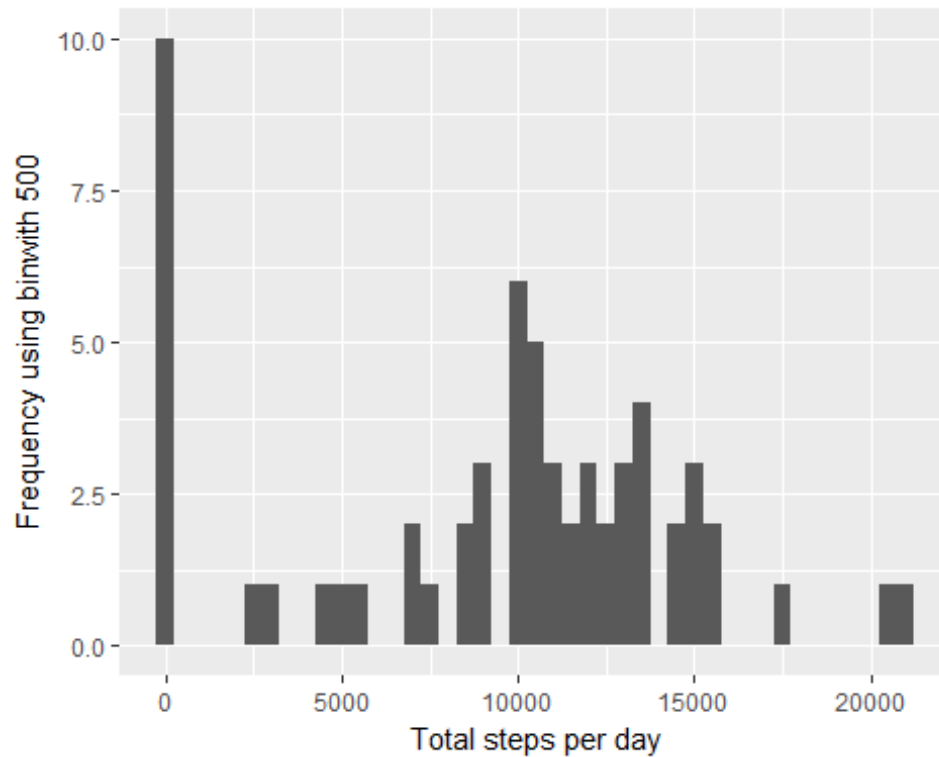
### What is mean total number of steps taken per day?

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

```
## 2012-10-01 2012-10-02 2012-10-03 2012-10-04 2012-10-05 2012-10-06
##          0         126        11352        12116        13294        15420
## 2012-10-07 2012-10-08 2012-10-09 2012-10-10 2012-10-11 2012-10-12
##        11015          0        12811         9900        10304        17382
## 2012-10-13 2012-10-14 2012-10-15 2012-10-16 2012-10-17 2012-10-18
##        12426        15098        10139        15084        13452        10056
## 2012-10-19 2012-10-20 2012-10-21 2012-10-22 2012-10-23 2012-10-24
##        11829        10395         8821        13460         8918         8355
## 2012-10-25 2012-10-26 2012-10-27 2012-10-28 2012-10-29 2012-10-30
##         2492         6778        10119        11458         5018         9819
## 2012-10-31 2012-11-01 2012-11-02 2012-11-03 2012-11-04 2012-11-05
##        15414          0        10600        10571          0        10439
## 2012-11-06 2012-11-07 2012-11-08 2012-11-09 2012-11-10 2012-11-11
##         8334        12883         3219          0          0        12608
## 2012-11-12 2012-11-13 2012-11-14 2012-11-15 2012-11-16 2012-11-17
##        10765         7336          0         41        5441        14339
## 2012-11-18 2012-11-19 2012-11-20 2012-11-21 2012-11-22 2012-11-23
##        15110         8841         4472        12787        20427        21194
## 2012-11-24 2012-11-25 2012-11-26 2012-11-27 2012-11-28 2012-11-29
##        14478        11834        11162        13646        10183         7047
## 2012-11-30
##          0
```

### Make a histogram of the total number of steps taken each day

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



#2. Calculate and

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

```
stepsByDayMean <- mean(stepsByDay)
print(stepsByDayMean)

## [1] 9354.23

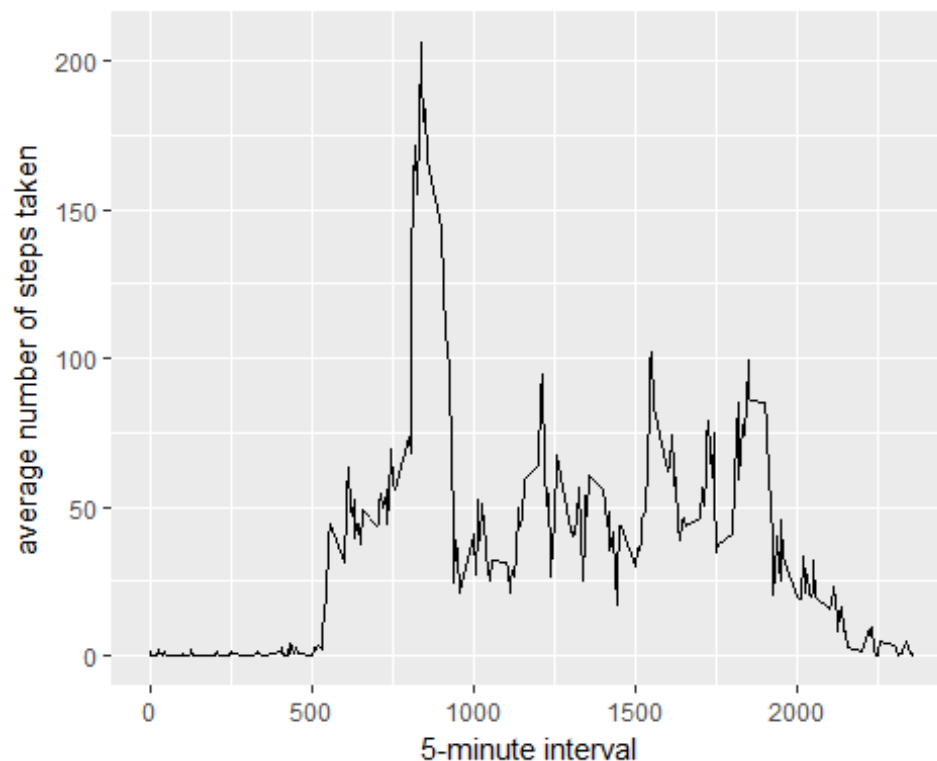
stepsByDayMedian <- median(stepsByDay)
print(stepsByDayMedian)

## [1] 10395
```

## What is the average daily activity pattern?

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

## 1. Make a time series plot



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

```
mostSteps <- which.max(averageStepsPerTimeBlock$meanSteps)
mostSteps

## [1] 104

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

## [1] "8:35"
```

## Imputing missing values

### 1. Calculate and report the total number of missing values in the dataset

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

## [1] 2304
```

**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.**

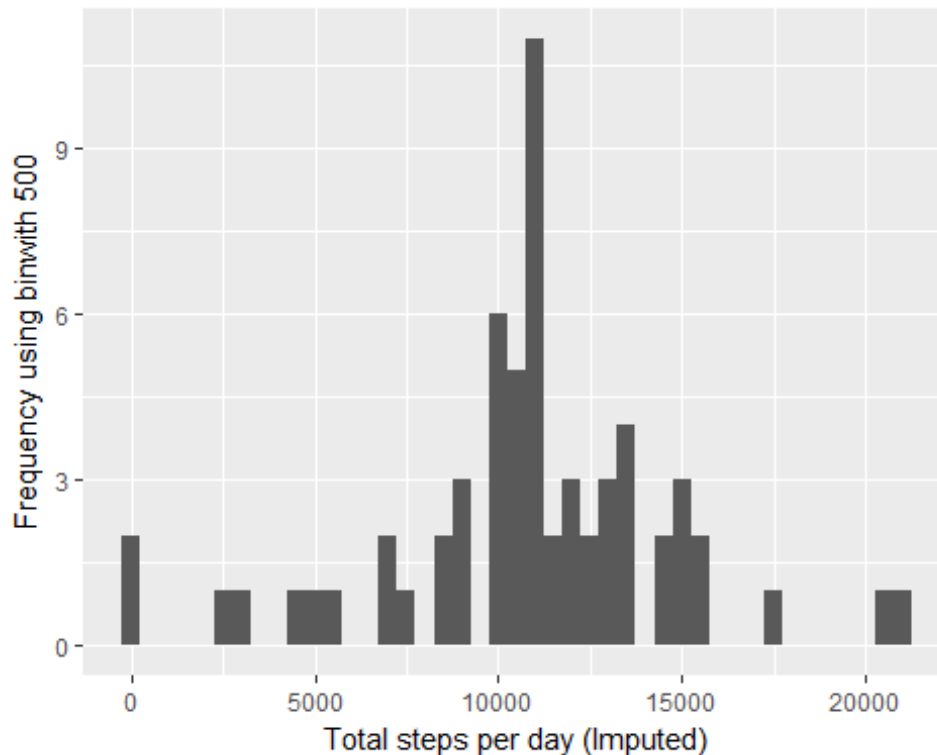
```
library(Hmisc)

## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
##
## Attaching package: 'Hmisc'
##
## The following objects are masked from 'package:base':
##
##      format.pval, round.POSIXt, trunc.POSIXt, units

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

**4. Make a 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 binwidth 500', binwidth=500)
```



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

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

**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. Make a 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")
```

