##

## 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

##

## Attaching package: 'lubridate'

## The following object is masked from 'package:base':

##

## date

1. Code for reading in the dataset and processing the data

activity<-read.csv("E:/R/Reproducible Research/2/Assignment/activity.csv", header=TRUE, sep=",")

summary(activity)

## steps date interval

## Min. : 0.00 2012-10-01: 288 Min. : 0.0

## 1st Qu.: 0.00 2012-10-02: 288 1st Qu.: 588.8

## Median : 0.00 2012-10-03: 288 Median :1177.5

## Mean : 37.38 2012-10-04: 288 Mean :1177.5

## 3rd Qu.: 12.00 2012-10-05: 288 3rd Qu.:1766.2

## Max. :806.00 2012-10-06: 288 Max. :2355.0

## NA's :2304 (Other) :15840

activity$date<-ymd(activity$date)

groupActivity <- group\_by(activity, date)

newActivity<-summarize(groupActivity, steps= sum(steps, na.rm = TRUE ))

newActivity

## # A tibble: 61 x 2

## date steps

## <date> <int>

## 1 2012-10-01 0

## 2 2012-10-02 126

## 3 2012-10-03 11352

## 4 2012-10-04 12116

## 5 2012-10-05 13294

## 6 2012-10-06 15420

## 7 2012-10-07 11015

## 8 2012-10-08 0

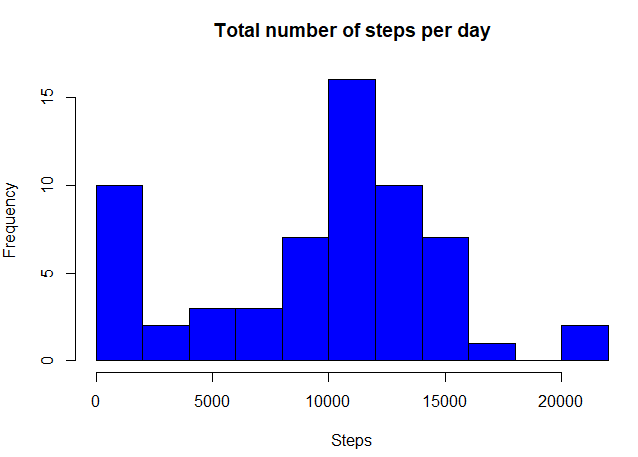
## 9 2012-10-09 12811

## 10 2012-10-10 9900

## # ... with 51 more rows

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

hist(newActivity$steps, breaks=8,main = "Total number of steps per day", xlab = "Steps", col = "blue")



### 3. Mean and median number of steps taken each day

mean(newActivity$steps)

## [1] 9354.23

median(newActivity$steps)

## [1] 10395

### 4. Time series plot of the average number of steps taken

First we fix our dataset and found the average number of steps taken

intervalGroupActivity <- group\_by(activity, interval)

averageActivity<-summarize(intervalGroupActivity, steps= mean(steps, na.rm = TRUE ))

averageActivity

## # A tibble: 288 x 2

## interval steps

## <int> <dbl>

## 1 0 1.7169811

## 2 5 0.3396226

## 3 10 0.1320755

## 4 15 0.1509434

## 5 20 0.0754717

## 6 25 2.0943396

## 7 30 0.5283019

## 8 35 0.8679245

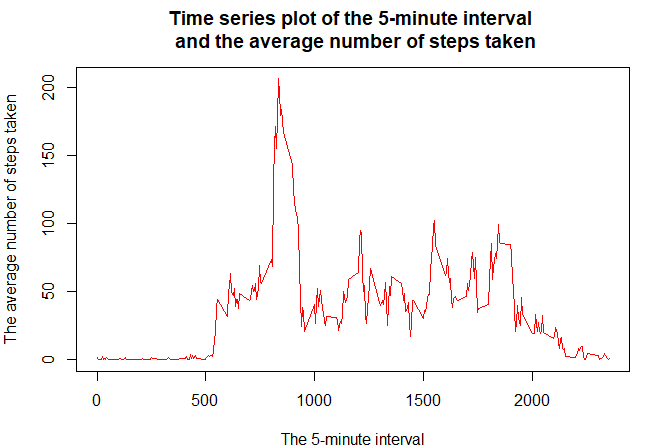
## 9 40 0.0000000

## 10 45 1.4716981

## # ... with 278 more rows

And then we make the time series plot

plot(averageActivity$interval,averageActivity$steps, type="l", col="red",main="Time series plot of the 5-minute interval \n and the average number of steps taken", xlab = "The 5-minute interval", ylab = "The average number of steps taken")



### 5. Which is the 5-minute interval that, on average, contains the maximum number of steps

averageActivity[which.max(averageActivity$steps), ]$interval

## [1] 835

6. Code to describe and show a strategy for imputing missing data

1. Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

summary(activity)

## steps date interval

## Min. : 0.00 Min. :2012-10-01 Min. : 0.0

## 1st Qu.: 0.00 1st Qu.:2012-10-16 1st Qu.: 588.8

## Median : 0.00 Median :2012-10-31 Median :1177.5

## Mean : 37.38 Mean :2012-10-31 Mean :1177.5

## 3rd Qu.: 12.00 3rd Qu.:2012-11-15 3rd Qu.:1766.2

## Max. :806.00 Max. :2012-11-30 Max. :2355.0

## NA's :2304

So we have 2304 NA’s

1. Devise a strategy for filling in all of the missing values in the dataset. Create a new dataset that is equal to the original dataset but with the missing data filled in. I will use the mean for that 5-minute interval.

newData<-activity

**for** (i **in** 1:nrow(newData))

{

**if** (is.na(newData$steps[i])){

newData$steps[i]<-averageActivity[which(newData$interval[i] == averageActivity$interval),]$steps

}

}

summary(newData)

## steps date interval

## Min. : 0.00 Min. :2012-10-01 Min. : 0.0

## 1st Qu.: 0.00 1st Qu.:2012-10-16 1st Qu.: 588.8

## Median : 0.00 Median :2012-10-31 Median :1177.5

## Mean : 37.38 Mean :2012-10-31 Mean :1177.5

## 3rd Qu.: 27.00 3rd Qu.:2012-11-15 3rd Qu.:1766.2

## Max. :806.00 Max. :2012-11-30 Max. :2355.0

sum(is.na(newData))

## [1] 0

## [1] 0

So as we can see we don’t have any NA’s in the new dataset.

1. Make a histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day. Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

groupNewData <- group\_by(newData, date)

newFillData<-summarize(groupNewData, steps= sum(steps, na.rm = TRUE ))

newFillData

## # A tibble: 61 x 2

## date steps

## <date> <dbl>

## 1 2012-10-01 10766.19

## 2 2012-10-02 126.00

## 3 2012-10-03 11352.00

## 4 2012-10-04 12116.00

## 5 2012-10-05 13294.00

## 6 2012-10-06 15420.00

## 7 2012-10-07 11015.00

## 8 2012-10-08 10766.19

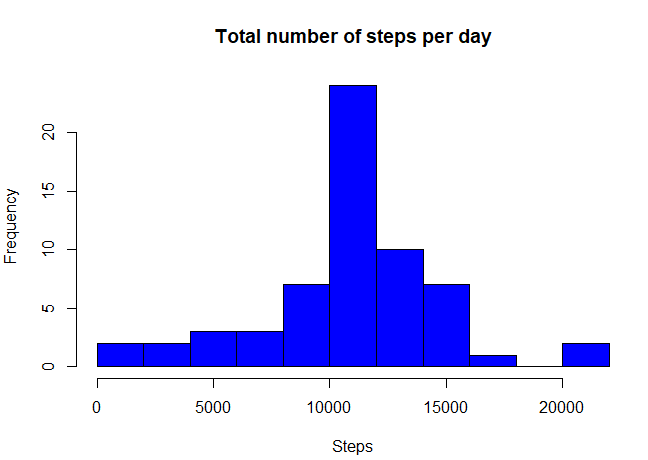
## 9 2012-10-09 12811.00

## 10 2012-10-10 9900.00

## # ... with 51 more rows

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

hist(newFillData$steps, breaks=8,main = "Total number of steps per day", xlab = "Steps", col = "blue")



Mean and median number of steps taken each day

mean(newFillData$steps)

## [1] 10766.19

median(newFillData$steps)

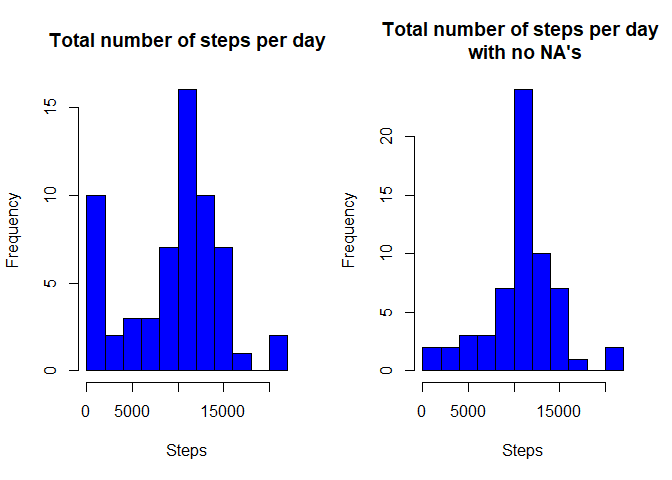
## [1] 10766.19

So, after imputing the missing data, the new mean of total steps taken per day is the same as that of the old mean; the new median of total steps taken per day is greater than that of the old median. We can see from the Histograms the big change is that the NA’s have move from the first class.

par(mfrow=c(1,2))

hist(newActivity$steps, breaks=8,main = "Total number of steps per day", xlab = "Steps", col = "blue")

hist(newFillData$steps, breaks=8,main = "Total number of steps per day \n with no NA's", xlab = "Steps", col = "blue")



**for** (i **in** 1:nrow(activity))

{

**if**(weekdays(activity$date[i])=="Σάββατο" |weekdays(activity$date[i])=="Κυριακή"){

activity$weekdays[i]<-"weekend"

}**else**{activity$weekdays[i]<-"weekday"}

}

intervalGroupActivityNA <- group\_by(activity, interval,weekdays)

dataNA<-summarize(intervalGroupActivityNA, steps= mean(steps, na.rm = TRUE ))

### 8. Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

qplot(interval,steps, data = dataNA,facets = weekdays~., geom="line")

