PA1_template.Rmd

MDA

August 29, 2016

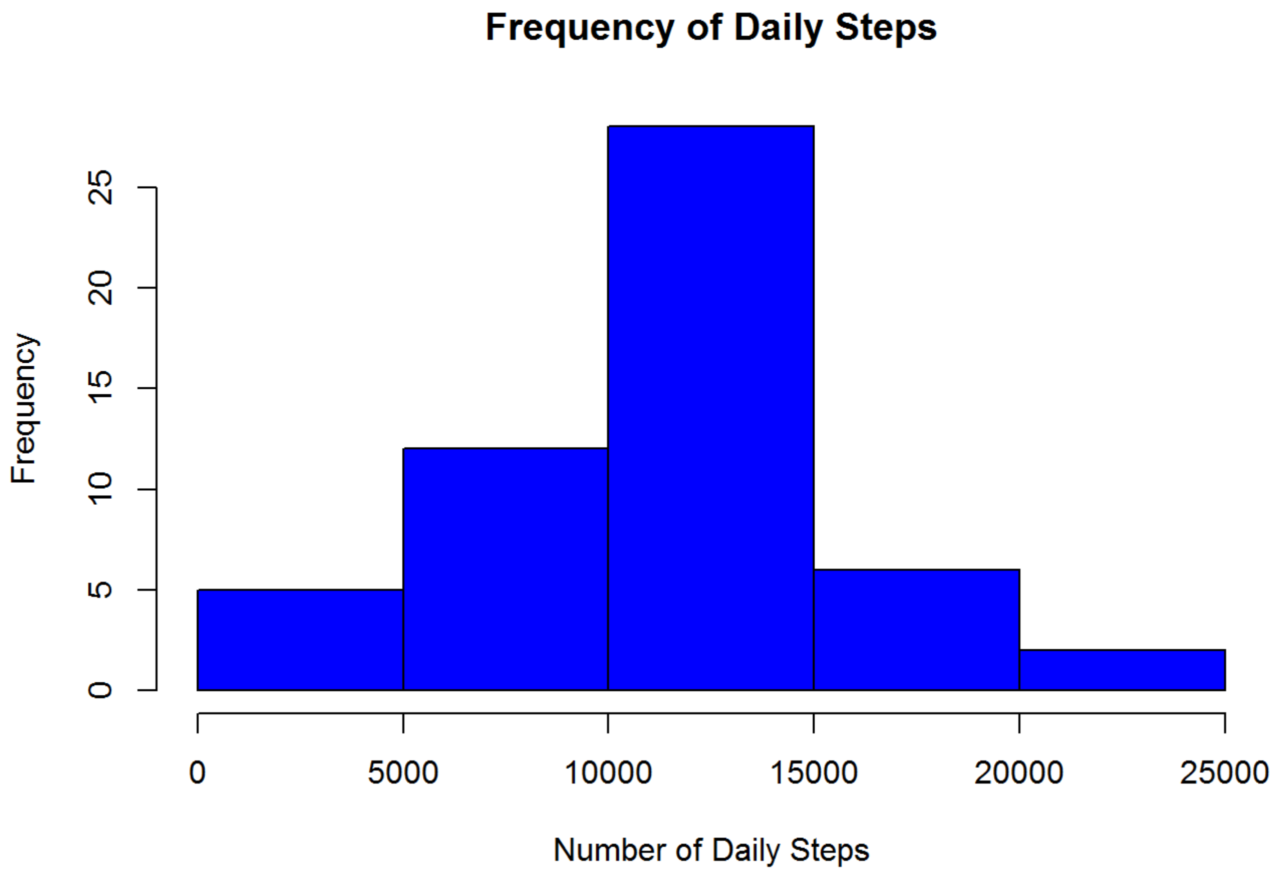
Load the data

```
# Set working directory
setwd("C:/Users/MD/Documents")
# Load data to a data frame dfrr
dfrr <- read.csv("activity.csv")
```

Total Number of Daily Steps

- Calculate the total number of steps taken per day
- Make a histogram of total number of steps taken each day
- Calculate and report the mean and the median of the total number of steps taken per day

```
# tds is total daily steps

tds <- tapply (dfrr$steps, dfrr$date,sum)

# create a histogram from tds slicing thedata into 5 bins, so breaks=5+1=6.
hist(tds, breaks=6, col="blue", xlab="Number of Daily Steps", ylab = "Frequency", main ="Frequen
cy of Daily Steps")
```



```
# tdsmean = mean of total daily steps
# tdsmed  = median of total daily steps

tdsmean <- mean(tds, na.rm=TRUE)
tdsmean
```

```
## [1] 10766.19
```

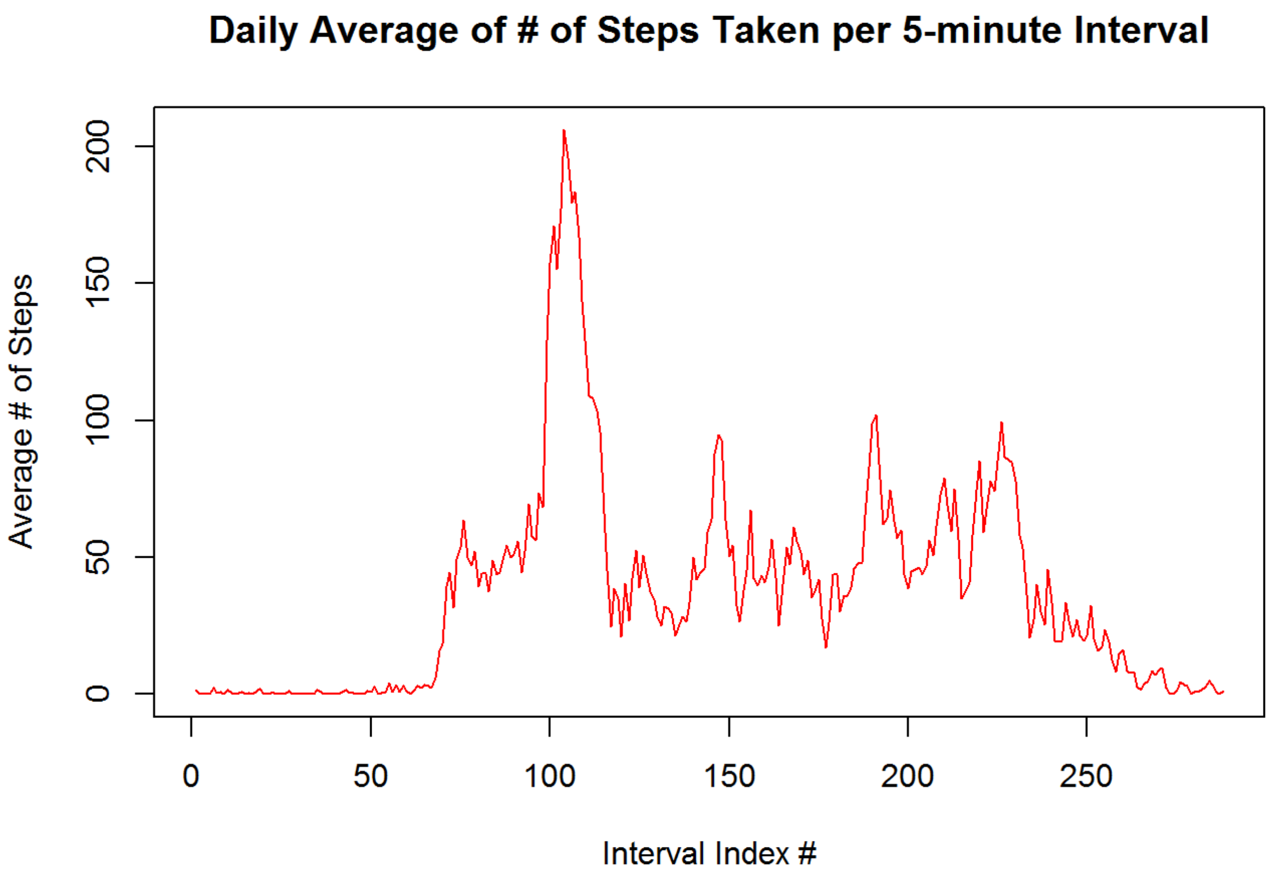
```
tdsmed <- median(tds, na.rm=TRUE)
tdsmed
```

```
## [1] 10765
```

Average Daily Activity Pattern

- Time series plot (i.e. type = “l”) of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)
- Determine the 5-minute interval, on average across all the days in the dataset, which contains the maximum number of steps.

```
# mspi = mean steps per interval
 mspi <- tapply(dfrr$steps, dfrr$interval, mean, na.rm=TRUE)

# plot mspi as time-series (type="l")


plot(mspi, type = "l", main= "Daily Average of # of Steps Taken per 5-minute Interval", xlab =
"Interval Index #", ylab ="Average # of Steps", col="red")
```



```
# interval index having the maximum number of steps
which(mspi %in% max(mspi))
```

```
## [1] 104
```

Imputing Missing Values

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

```
dfna <-is.na(dfrr$steps)
tdfna <-sum(as.numeric(dfna))
tdfna
```

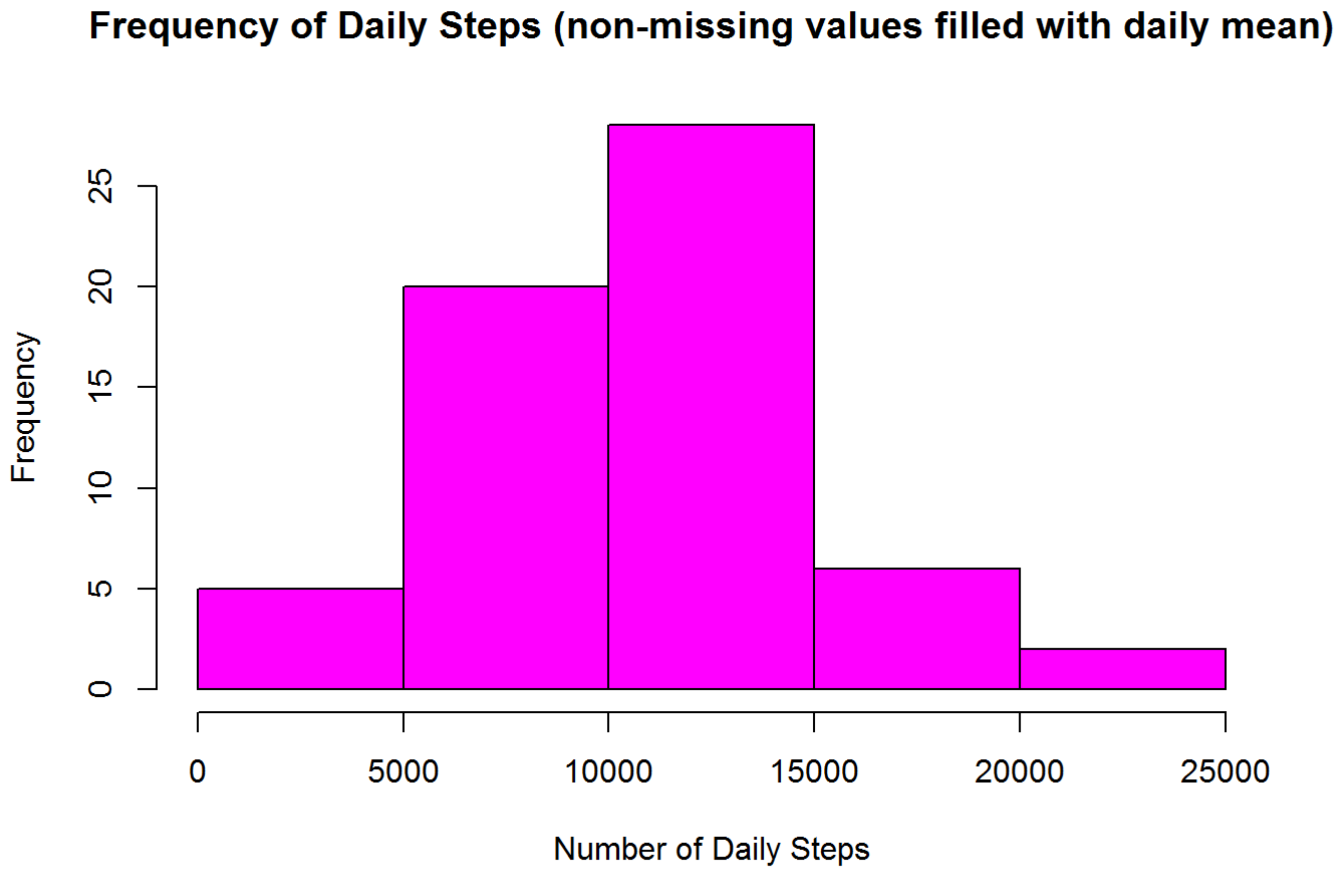
```
## [1] 2304
```

- Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to be sophisticated. For example, you could use the mean/median for that day, or the mean for that 5-minute interval, etc.
- Create a new dataset that is equal to the original dataset but with the missing data filled in.
- 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?

Including Plots

You can also embed plots, for example:

```
dfn<-transform(dfrr, steps=ifelse(is.na(dfrr$steps), tapply(dfrr$steps, dfrr$date, mean, na.rm=T
RUE), dfrr$steps))

dfns<-tapply(dfn$steps, dfn$date, sum, na.rm=TRUE)

hist(dfns, breaks=6, col="magenta", main="Frequency of Daily Steps (non-missing values filled wi
th daily mean)", xlab="Number of Daily Steps", ylab="Frequency")
```



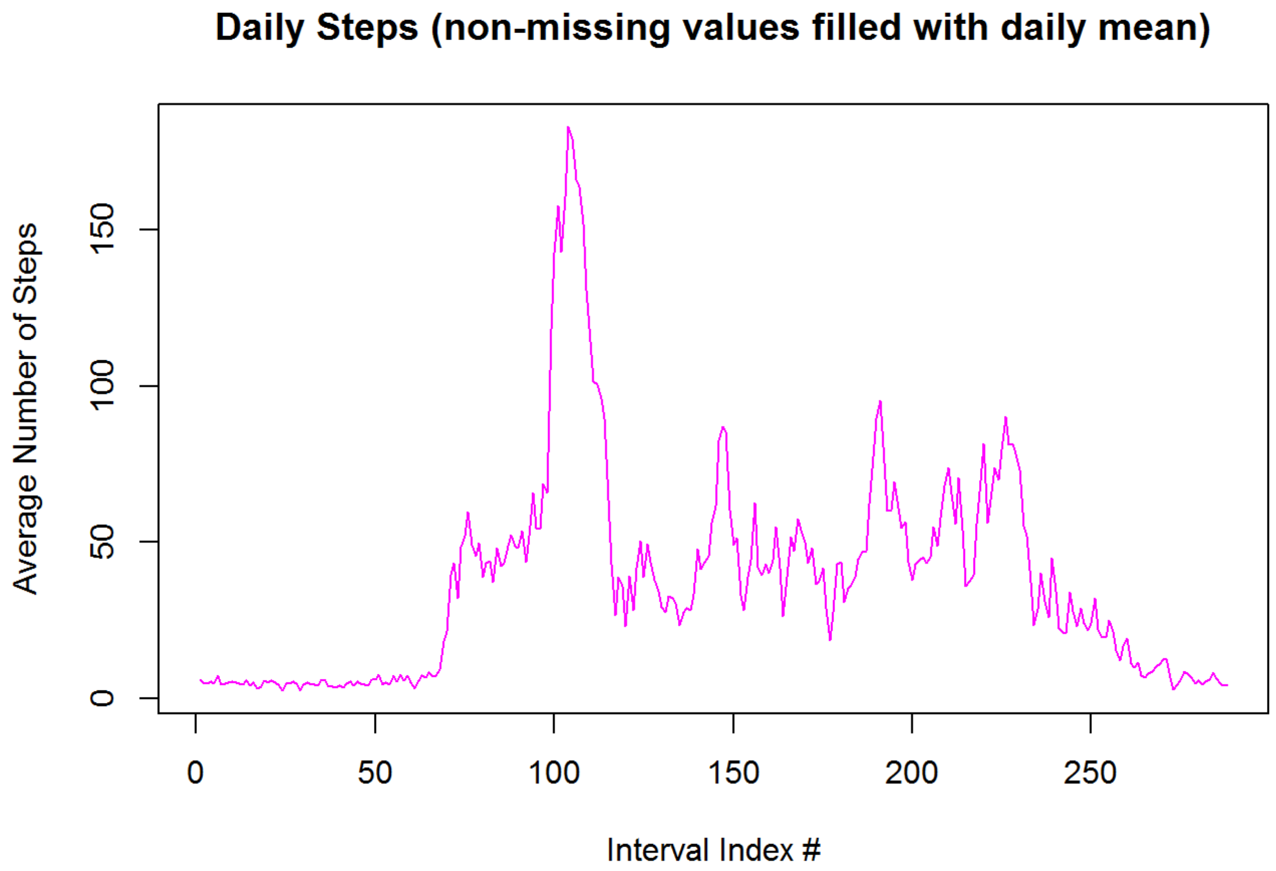
```
dfns_mean<-mean(dfns, na.rm=TRUE)
dfns_mean
```

```
## [1] 10579.21
```

```
dfns_median<-median(dfns, na.rm=TRUE)
dfns_median
```

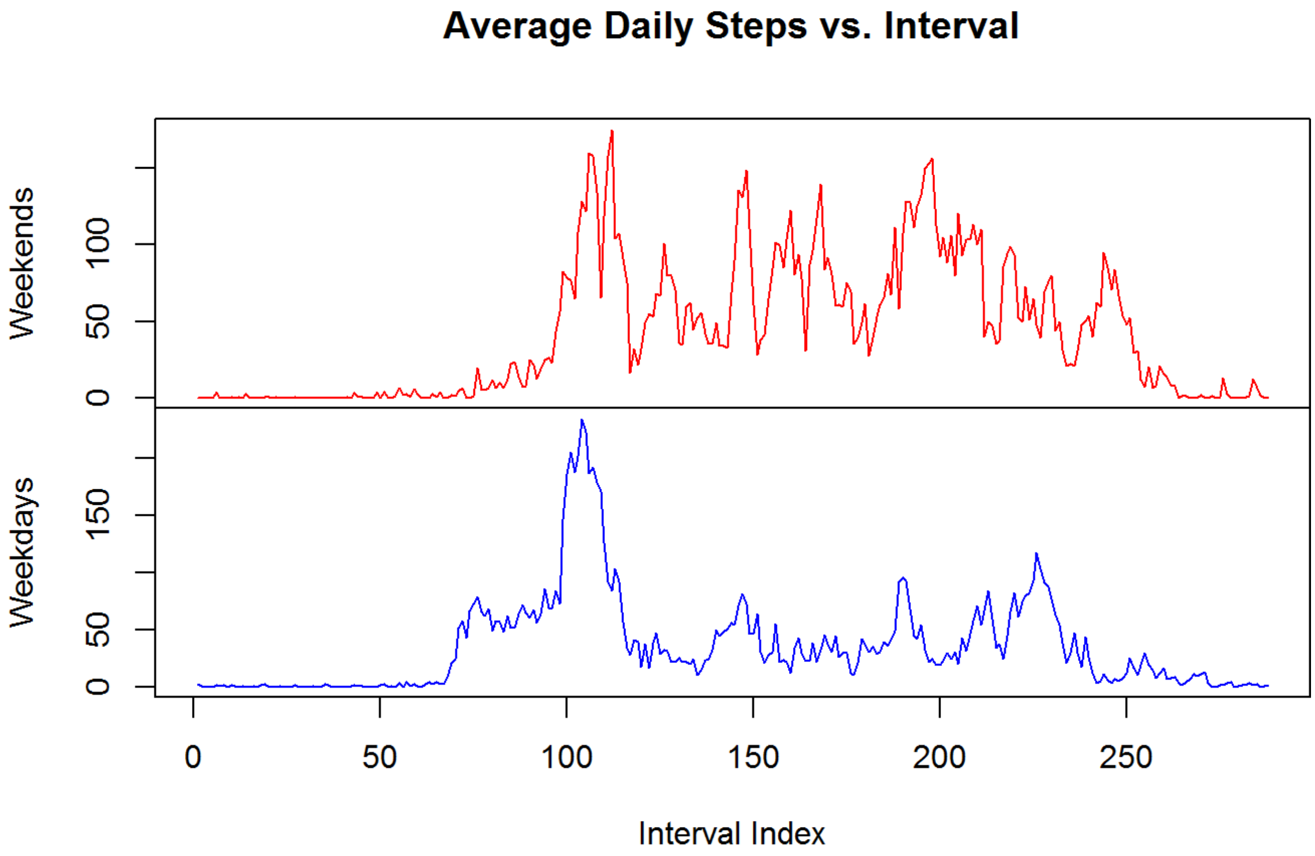
```
## [1] 10395
```

```
dfnn<-tapply(dfn$steps, dfn$interval, mean, na.rm=TRUE)

plot(dfnn, type="l", xlab="Interval Index #", ylab="Average Number of Steps", main="Daily Steps (non-missing values filled with daily mean)", col="magenta")
```



##Panel Plot for Average Steps for Weekdays and Weekends

```
xd<-as.POSIXlt(dfrr$date, format="%Y-%m-%d")
xwd<-xd$wday

# Weekend days set to factor level "0"
xwd[xwd==0]=0
xwd[xwd==6]=0

# Weekdays set to factor level "1"
xwd[xwd!=0]=1

# Define factor levels
xwdf<-factor(xwd, levels=c(0,1))

# Next, add the factor variable as column "wd", to a copy of the dataset(dfrrf).
dfrrf<-dfrr
dfrrf$wd<-xwdf

# Calculate average number of steps per day
mspiwd<- tapply(dfrrf$steps, list(dfrrf$interval, dfrrf$wd), mean, na.rm=TRUE)

# Make a panel plot to display weekday and weekend data in separate panels.
    par(mfrow=c(2,1))
with (dfrrf, {
    par(mai=c(0,1,1,0))
    plot (mspiwd[,1], type="l", main="Average Daily Steps vs. Interval", xaxt="n", ylab="Weekends", col="red")
    title=("Average Daily Steps vs. Interval Index #")
    par(mai=c(1,1,0,0))
    plot (mspiwd[,2], type="l", xlab="Interval Index", ylab="Weekdays", col="blue")
    })
```



End of Report