Reproducible Research Project using UCI Activity Recognition Data (Johns Hopkins Data Science Certification)

MD Alamgir Ph.D.

August 29, 2016

Project Description

Impute missing values and update the dataset

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

Load the data

```
# Set working directory
setwd("C:/Users/MD/Documents")
# Data Source: UCI Activity Recognition database
# https://archive.ics.uci.edu/ml/datasets/Heterogeneity+Activity+Recognition
# Download the zipfile and extract it to the working directory.
# Load data to a data frame dfrr
dfrr <- read.csv("activity.csv")
```

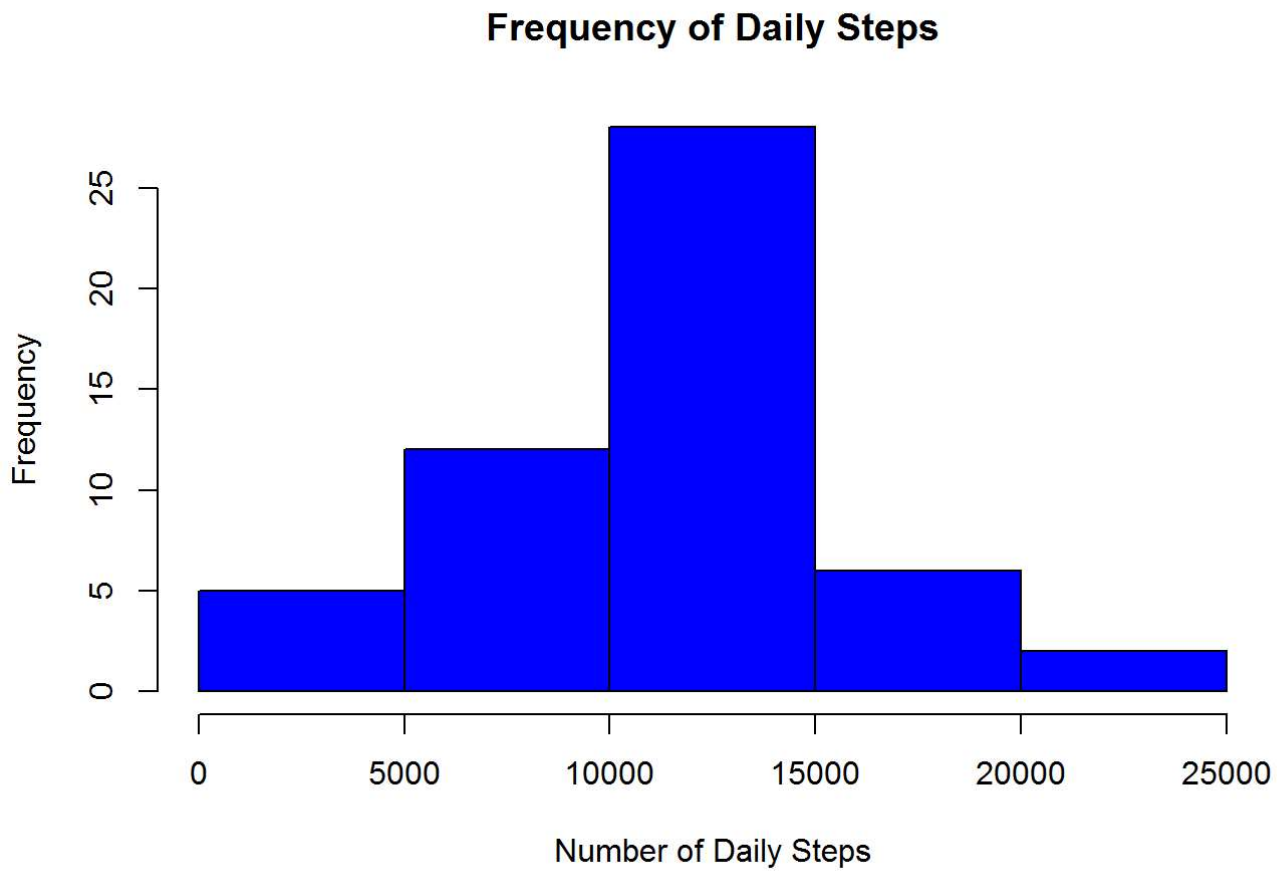
Total Number of Daily Steps

- 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
```

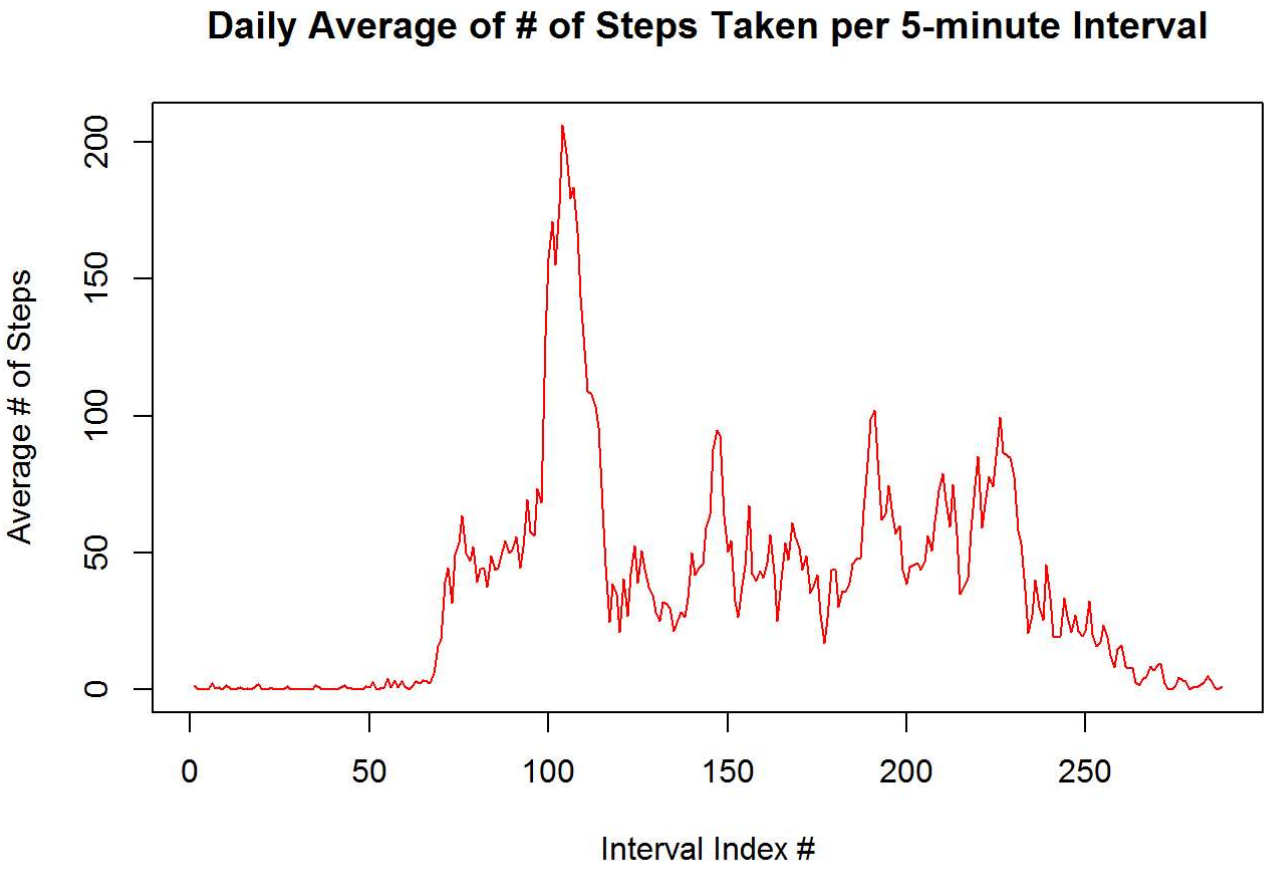
Average Daily Activity Pattern

- Create a 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 perinterval
 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
```