Course Project 1 - Reproducible Research

Vivek Joshi

March 31, 2017

Introduction

This is the project for peer review in Reproducible Research on Coursera. Data is available from https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip and the file is downloaded into a temporary directory, read and then deleted. The questions are answered based on the available data.

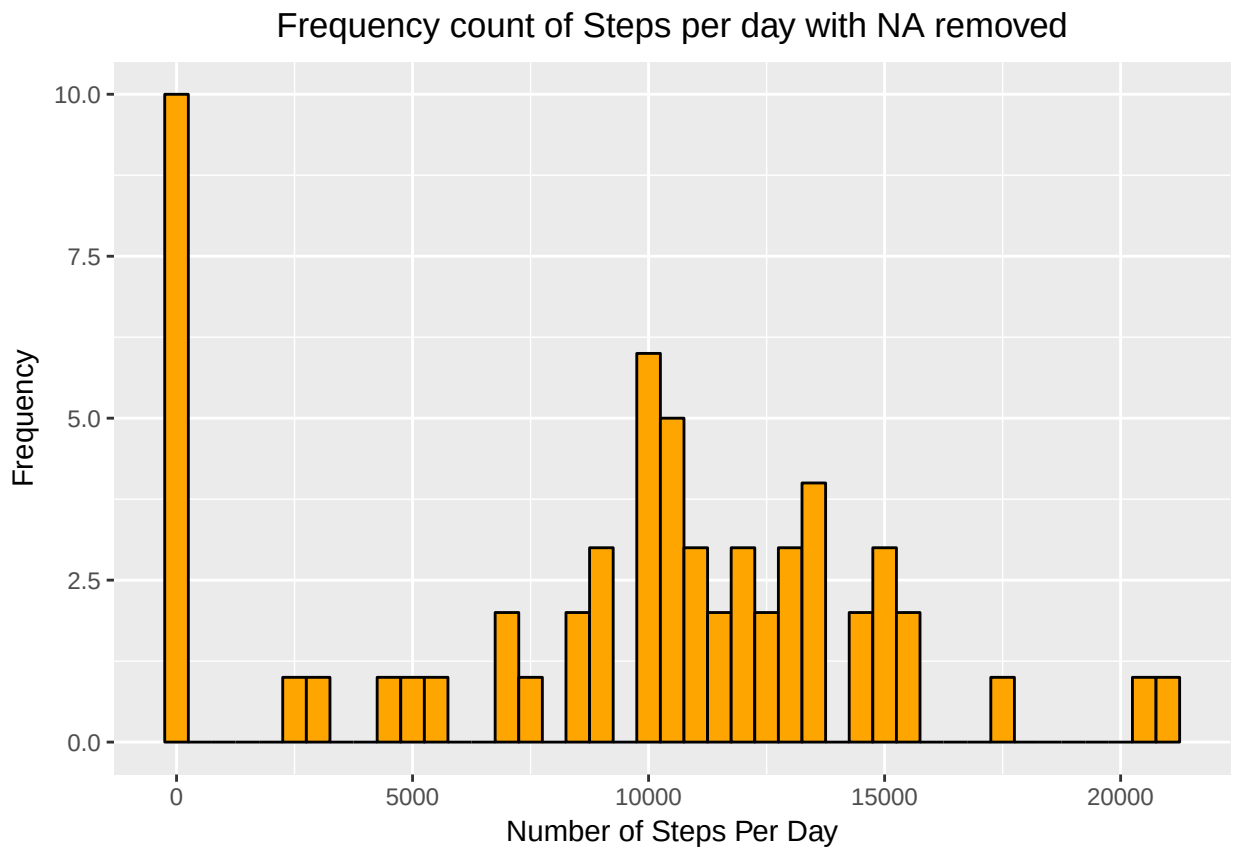
Loading/Reading in the Data

```
setwd("C:\\Users\\vivek\\Documents\\Data Science\\Coursera\\05 - Reproducible Research\\Lecture 02")
download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip", "project.zip")
df <- read.csv(unz("project.zip", "activity.csv"))
file.remove("project.zip")
```

```
## [1] TRUE
```

Histogram of the total number of steps taken per day

```
dplyr_tot <- df %>%
  group_by(date) %>%
  summarise(daily = sum(steps, na.rm = TRUE))
ggplot(dplyr_tot, aes(daily)) +
  geom_histogram(binwidth = 500, fill="orange", colour="black") +
  xlab("Number of Steps Per Day") +
  ylab("Frequency") +
  ggtitle("Frequency count of Steps per day with NA removed") +
  theme(plot.title = element_text(hjust = 0.5))
```



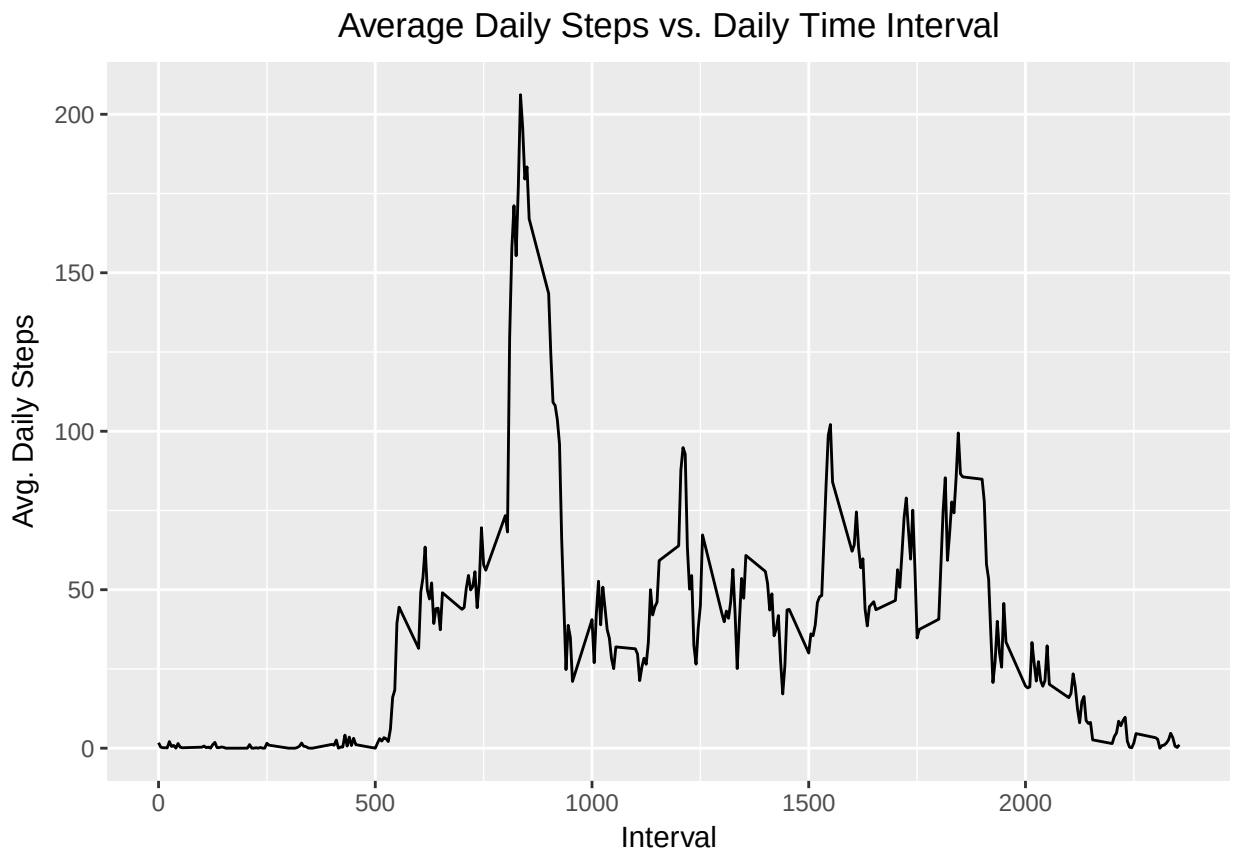
Average number of steps taken per day is : 9354.23

Median number of steps taken per day is : 10395

Time series plot of the average number of steps taken

```
df.avg.steps <- df %>%
  group_by(interval) %>%
  summarise(avg.steps = mean(steps, na.rm = TRUE))

ggplot(df.avg.steps, aes(interval, avg.steps)) + geom_line() +
  xlab("Interval") + ylab("Avg. Daily Steps") +
  ggtitle("Average Daily Steps vs. Daily Time Interval") +
  theme(plot.title = element_text(hjust = 0.5))
```



Five minute interval that on the average has the max number of steps

Five minute interval with the max number of steps is : 835

Code to describe imputation strategy

The data above has a wide distribution, hence I have chosen to use the median value for the same interval for interpolation. Another thing that could be done is to impute on the basis of time interval and whether it is a weekend or weekday, but I have not performed that analysis. To check whether the median for imputation makes sense, we can check with the summary command

```
summary(df$steps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.00    0.00   37.38   12.00  806.00    2304
```

Moving on for the imputation yields

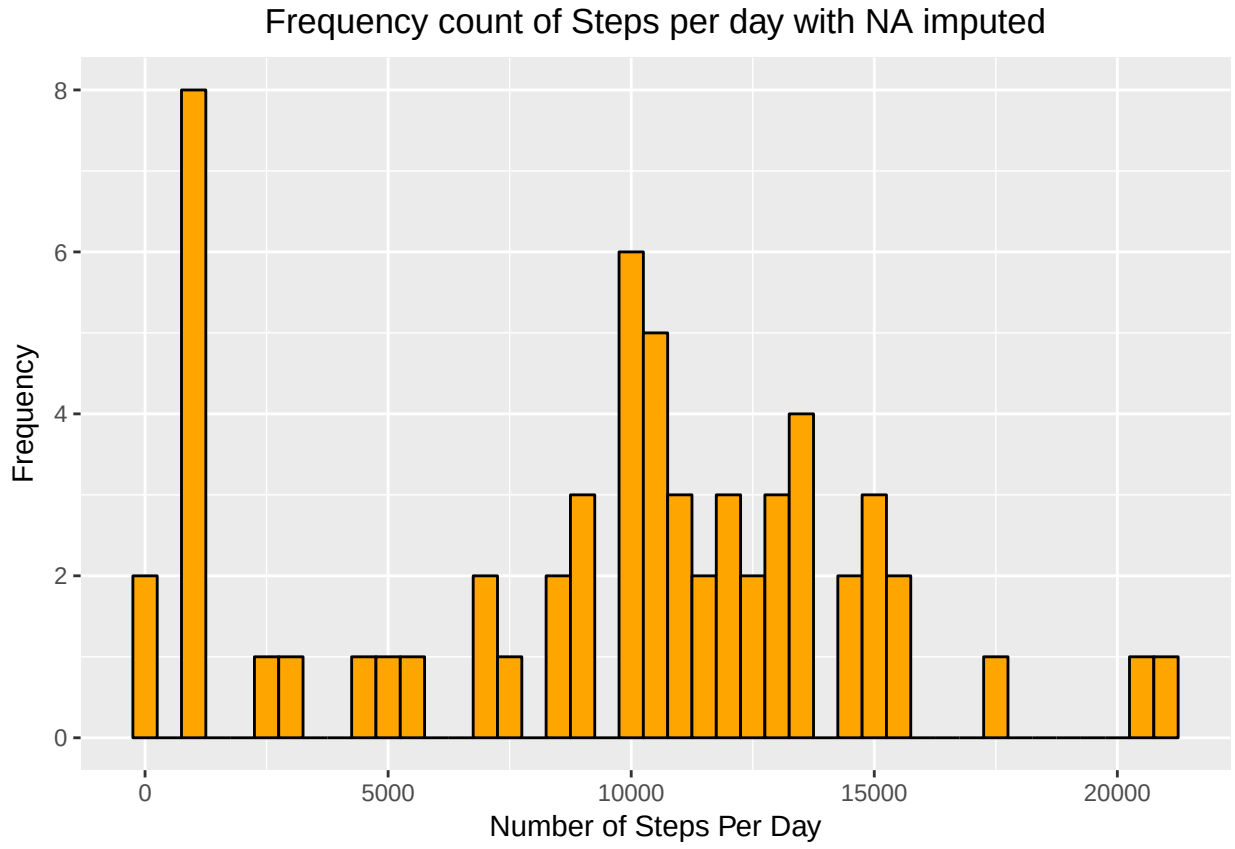
```
df.nona <- df # copy the dataframe, will clean and use...
where.nas <- is.na(df.nona$steps)
# compute median over interval, since data has "wide" distribution.
impute.time <- tapply(df.nona$steps, df.nona$interval, median, na.rm = TRUE, simplify = TRUE)
df.nona$steps[where.nas] <- impute.time[as.character(df.nona$interval[where.nas])]
# check for no NAs
sum(is.na(df.nona))
```

```
## [1] 0
```

Histogram of the total number of steps taken each day after missing values are imputed

```
dplyr.nona.tot <- df.nona %>%
  group_by(date) %>%
  summarise(daily = sum(steps, na.rm = TRUE))

# can use tapply too
# total.steps <- tapply(df$steps, df$date, FUN=sum, na.rm=TRUE, simplify = TRUE)
ggplot(dplyr.nona.tot, aes(daily)) +
  geom_histogram(binwidth = 500, fill="orange", colour="black") +
  xlab("Number of Steps Per Day") +
  ylab("Frequency") +
  ggtitle("Frequency count of Steps per day with NA imputed") +
  theme(plot.title = element_text(hjust = 0.5))
```



Panel plot comparing average number of steps taken per 5-minute interval across weekdays and weekend

```
# convert date to Date type so can use weekday
df.nona$date <- as.Date(df.nona$date)
```

```
df.nona <- df.nona %>%
  mutate(daytype = ifelse(weekdays(df.nona$date) %in% c("Saturday","Sunday"), "Weekend", "Weekday"))


df.nona <- df.nona %>%
  group_by(interval) %>%
  mutate(avg.steps = mean(steps, na.rm = TRUE))

ggplot(df.nona, aes(x =interval , y=steps, color=daytype)) +
  geom_line() +
  labs(title = "Ave Daily Steps (type of day)", x = "Interval", y = "Total Number of Steps") +
  facet_wrap(~ daytype, ncol = 1, nrow=2) +
  theme(plot.title = element_text(hjust = 0.5))
```