Gadget User Activities Analysis

Mike Wu

12/27/2018

```
library(dplyr)
library(ggplot2)
library(HotDeckImputation)
```

importing & formattng data:

```
df <- read.csv("activity.csv")
df$date <- as.Date(df$date, "%Y-%m-%d")
```

```
colnames(df)
```

```
## [1] "steps"    "date"     "interval"
```

```
str(df)
```

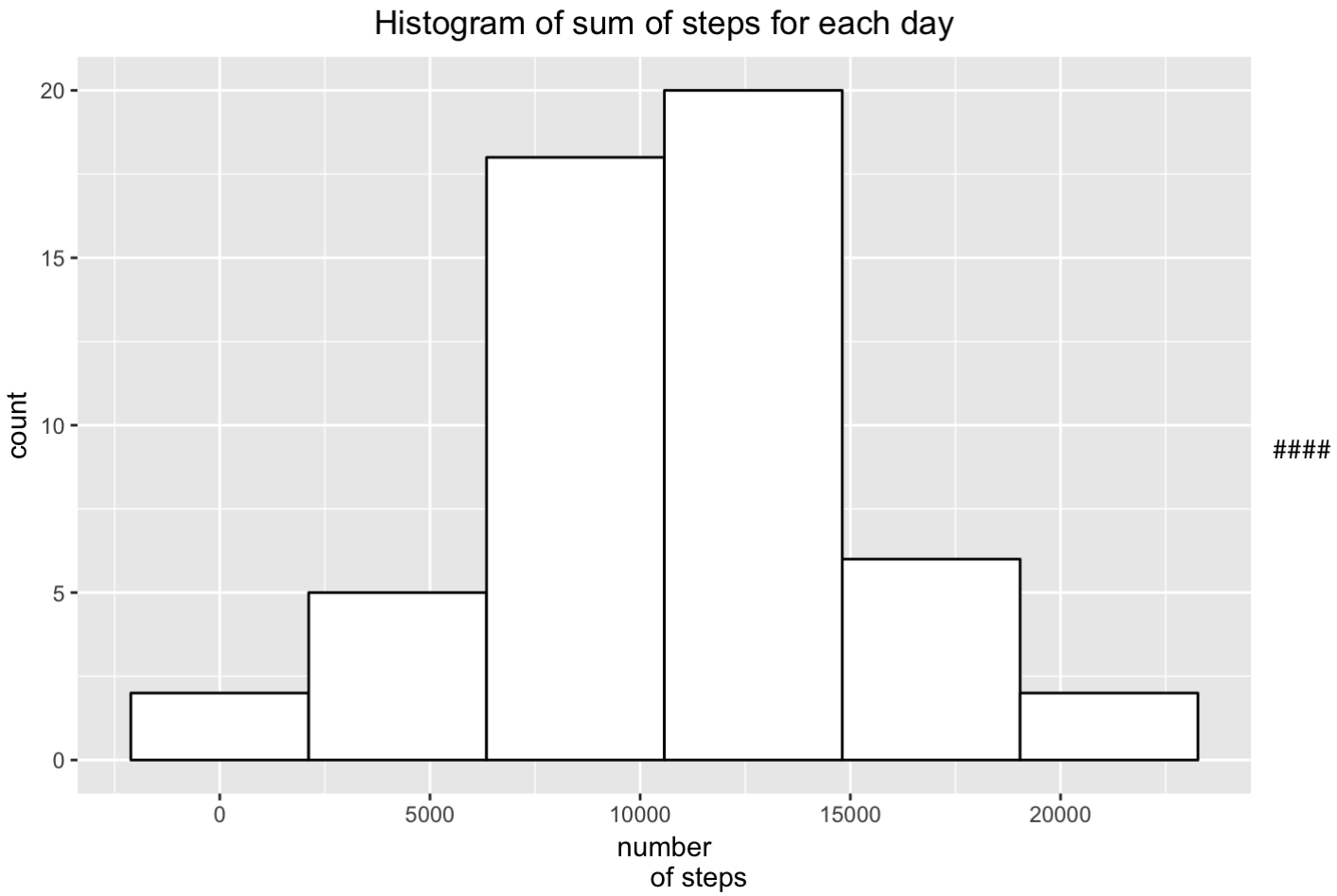
```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Date, format: "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

Histogram of the total number of steps taken each day

(For this part of the assignment, you can ignore the missing values in the dataset.)

```
df1 <- group_by(df, date)
df1 <- df1[!is.na(df1$steps),]
meanr  <- summarise(df1, meansteps = mean(steps, na.rm = T))
```

```
sumr <- summarise(df1, totalsteps = sum(steps, na.rm = T))
ggplot(data=sumr, aes(x =totalsteps )) + geom_histogram(bins = 6, fill="white",color="bl
ack") + xlab("number
         of steps") + ggtitle("Histogram of sum of steps for each day") + theme(plot.tit
le =
                                                                                element_
text(hjust = 0.5))
```



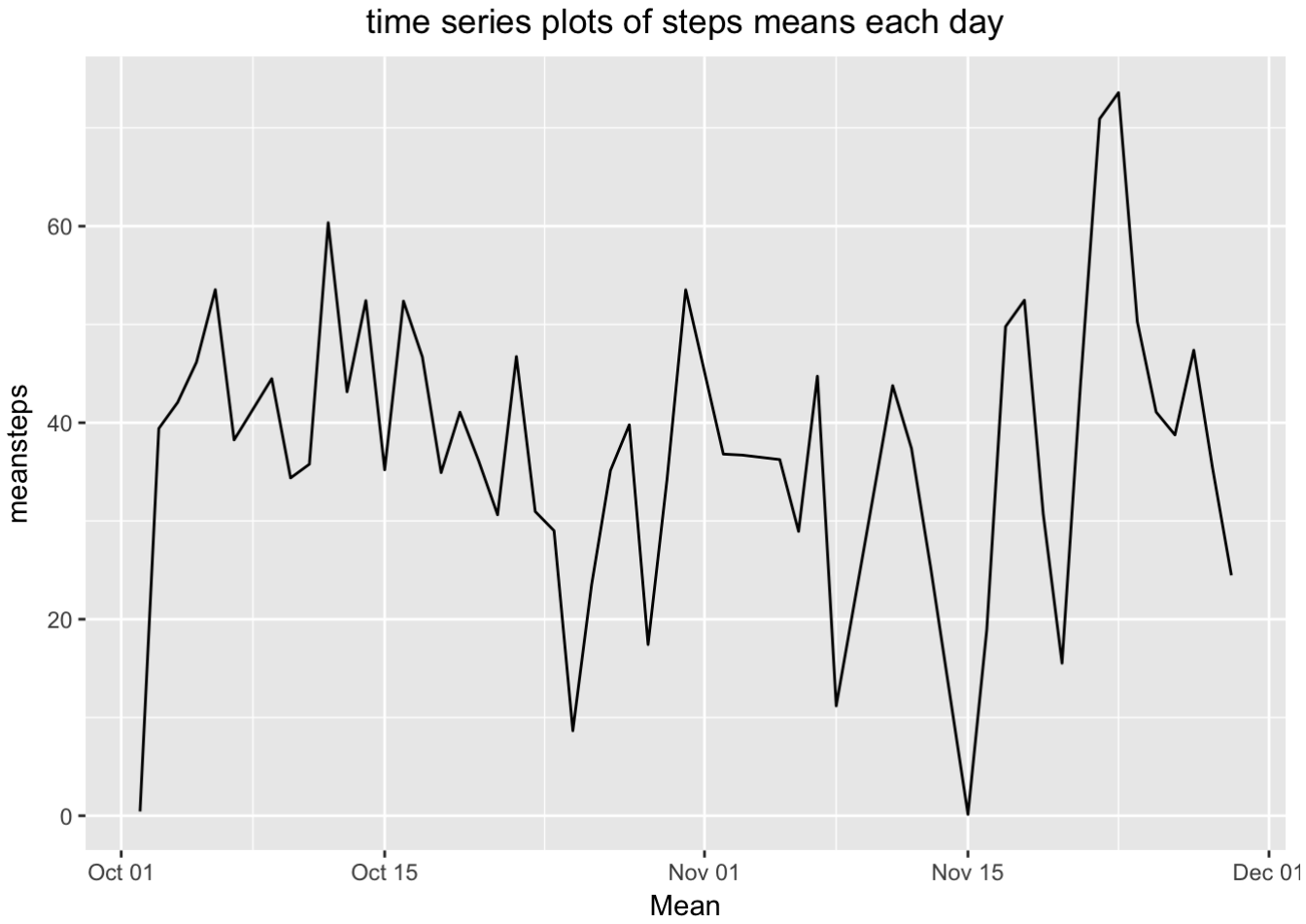
Mean and median number of steps taken each day

```
summary(sumr$totalsteps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      41    8841   10765   10766   13294   21194
```

Time series plot of the average number of steps taken

```
ggplot(data=meanr, aes(x = date, y =meansteps )) + geom_line() + xlab("Mean") + ggtitle(
"time series plots of steps means each day") + theme(plot.title = element_text(hjust =
0.5))
```



The 5-minute interval that, on average, contains the maximum number of steps

```
df[which.max(df$steps),]
```

```
##       steps       date interval
## 16492   806 2012-11-27      615
```

Code to describe and show a strategy for imputing missing data

refer to : Seven way to make up missing data (https://www.theanalysisfactor.com/seven-ways-to-make-up-data-common-methods-to-imputing-missing-data/)

```
length(df[which(is.na(df$steps)),]$steps)/length(df$steps)
```

```
## [1] 0.1311475
```

it shows that missing value comprised of 13% of total values.
I chose Hot deck Imputation,which find all the sample subjects who are similar on other variables, then randomly choose one of their values on the missing variable.
One advantage is you are constrained to only possible values. Another is the random component, which adds in some variability. This is important for accurate standard errors.

```
# convert df into dfm as matrix
dfm <- data.matrix(df, rownames.force = NA)
```

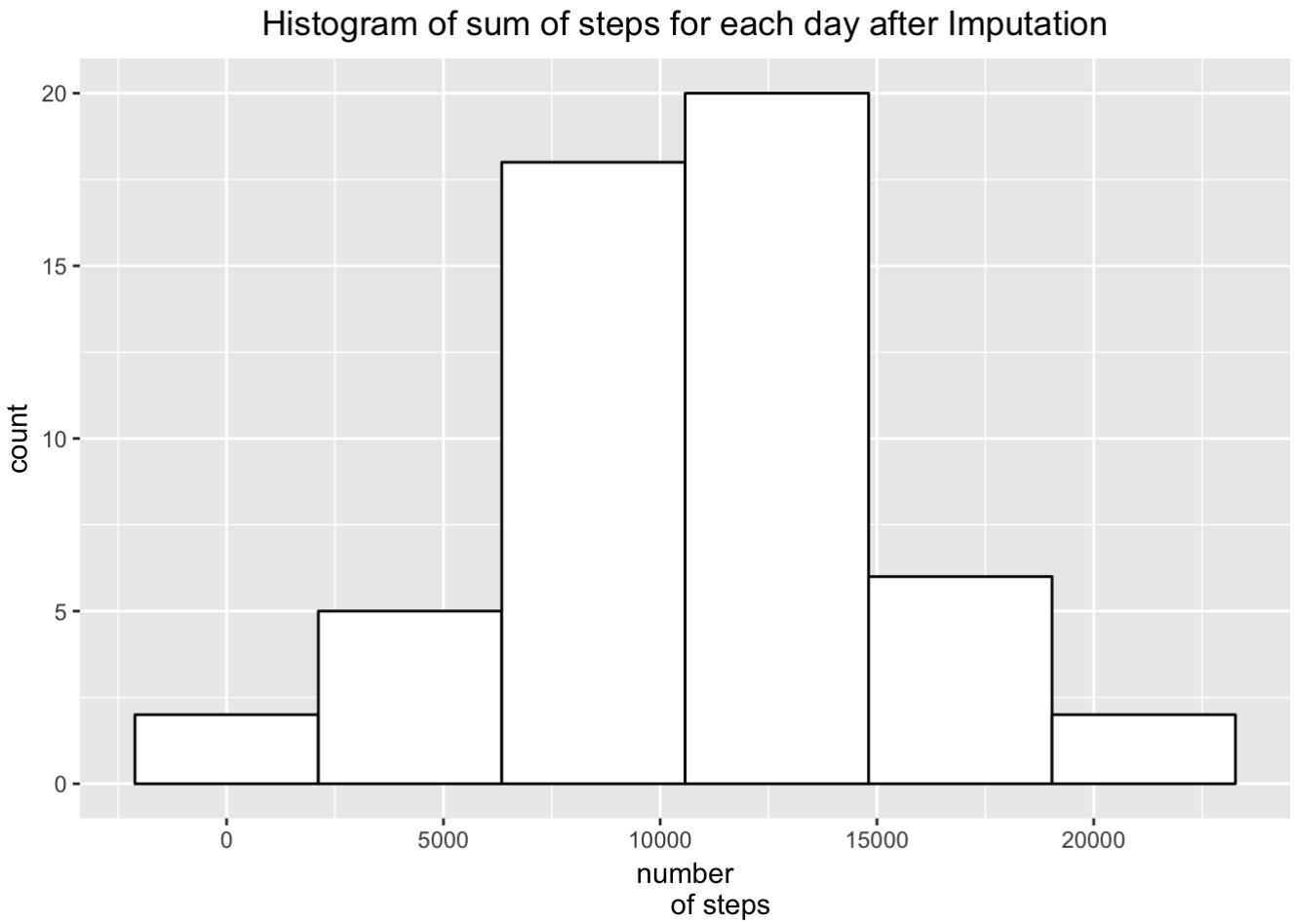
Description:
A comprehensive function that performs nearest neighbor hot deck imputation. Aspects such as variable weighting, distance types, and donor limiting are implemented. New concepts such as the optimal distribution of donors are also available.
refer to : HotDeckImputationPackages (https://cran.r-project.org/web/packages/HotDeckImputation/HotDeckImputation.pdf)

```
dfm1 <- impute.NN_HD(DATA = dfm, distance = "man", weights = "range", attributes = "sim"
,
comp = "rw_dist", donor_limit = Inf, optimal_donor = "no",
list_donors_recipients = NULL, diagnose = NULL)
df3 <- data.frame(dfm1)
df$steps <- df3$X1
```

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

```
df4 <- group_by(df, date)
df4 <- df4[!is.na(df4$steps),]
sumr1 <- summarise(df1, totalsteps = sum(steps, na.rm = T))

ggplot(data=sumr1, aes(x =totalsteps )) + geom_histogram(bins = 6, fill="white",color="b
lack") + xlab("number
         of steps") + ggtitle("Histogram of sum of steps for each day after Imputation")
 + theme(plot.title =
                                                                                element_
text(hjust = 0.5))
```



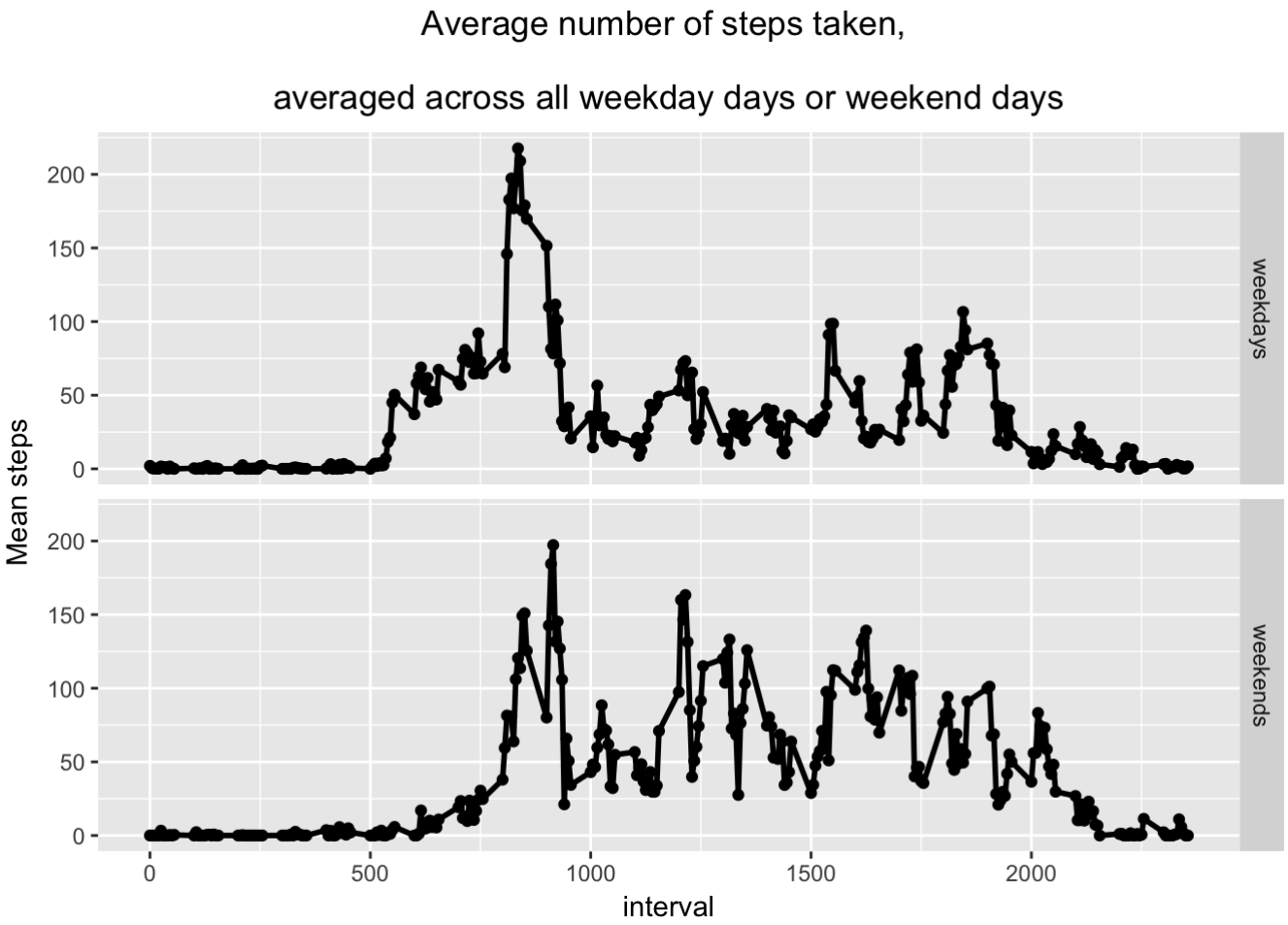
Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

```
dateType <- unlist(df$date)
DateType <- function(dt){

    days <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunda
y")

    if (weekdays(dt) %in% days[1:5]){
        return("weekdays")
    }
    else if (weekdays(dt) %in% days[6:7]) {
        return("weekends")
    }
}
newcol <- sapply(dateType, DateType)
df$DateTpe <- newcol
mean_number_steps <- aggregate(steps ~ interval+DateTpe, df, mean)

g <- qplot(interval, steps, data = mean_number_steps, facets = DateTpe~.)
g + geom_line(size = 1)+ ylab("Mean steps") + ggtitle("Average number of steps taken,
\n averaged across all weekday days or weekend days ")+ theme(plot.title = element_text(
hjust = 0.5))
```



Conclusion:

We do see some subtle differences between the average number of steps between weekdays and weekends. For instance, it appears that the user started a bit later on weekend mornings and tend to do smaller numbers on weekend mornings.