Reproduicble Research_WK2_Proj1

LH

2022-12-16

Loading and preprocessing the data

1. Load the data (i.e. read.csv())

```
library(plyr)
library(ggplot2)

act <- read.csv("activity.csv")
head(act)
```

```
##   steps       date interval
## 1    NA 2012-10-01        0
## 2    NA 2012-10-01        5
## 3    NA 2012-10-01       10
## 4    NA 2012-10-01       15
## 5    NA 2012-10-01       20
## 6    NA 2012-10-01       25
```

```
str(act)
```

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

2. Process/transform the data (if necessary) into a format suitable for your analysis

What is mean total number of steps taken per day?

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

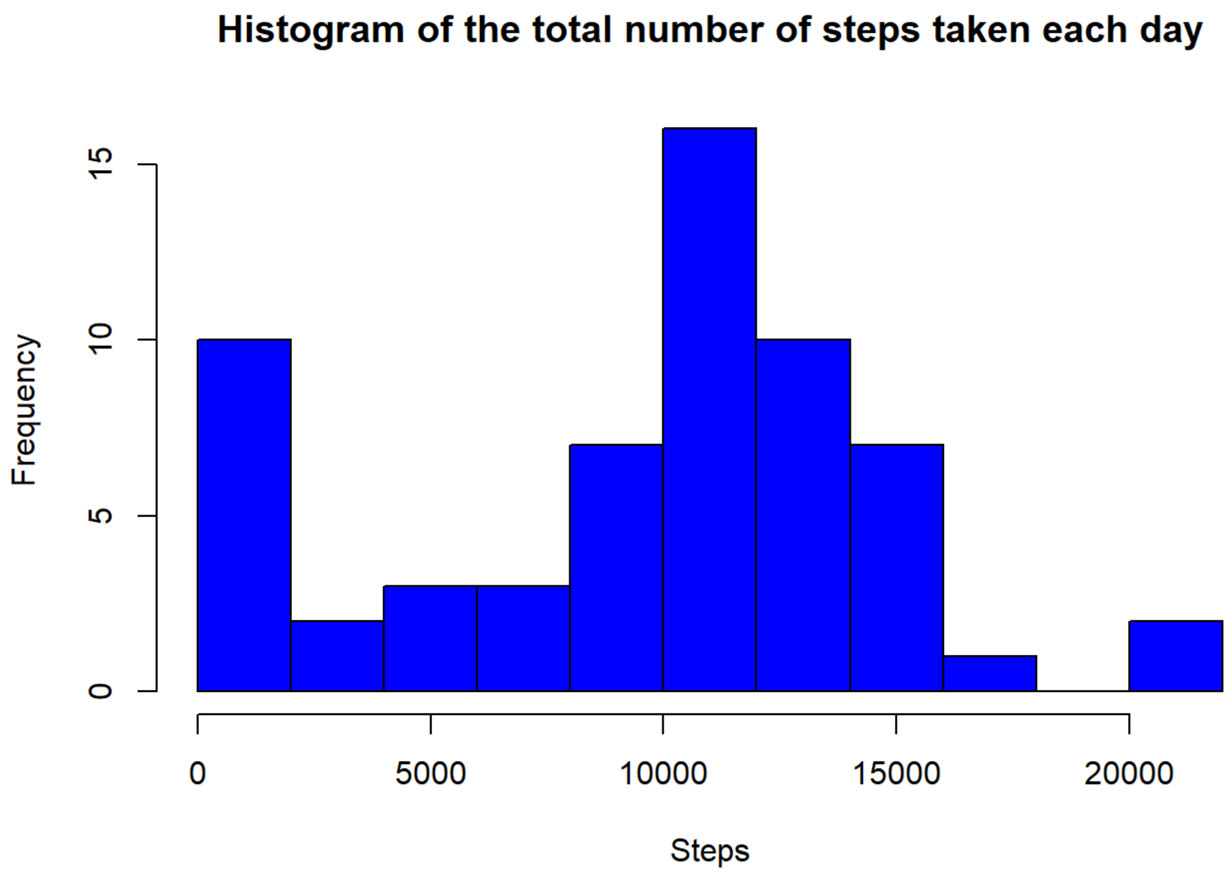
1. Calculate the total number of steps taken per day

```
tnstpd <-aggregate(act$steps, by=list(act$date), FUN = sum, na.rm=TRUE)
colnames(tnstpd) <- c("date", "tn_steps")
head(tnstpd)
```

```
##         date tn_steps
## 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
```

2. If you do not understand the difference between a histogram and a barplot, research the difference between them. Make a histogram of the total number of steps taken each day

```
hist(tnstpd$tn_steps, breaks =8, main = 'Histogram of the total number of steps taken e
ach day', xlab = 'Steps', col = 'blue')
```



3. Calculate and report the mean and median of the total number of steps taken per day difference between them. Make a histogram of the total number of steps taken each day

```
meanstpd <- round(mean(tnstpd$tn_steps))
print(paste('Mean of the total number of steps taken per day =', meanstpd))
```

```
## [1] "Mean of the total number of steps taken per day = 9354"
```

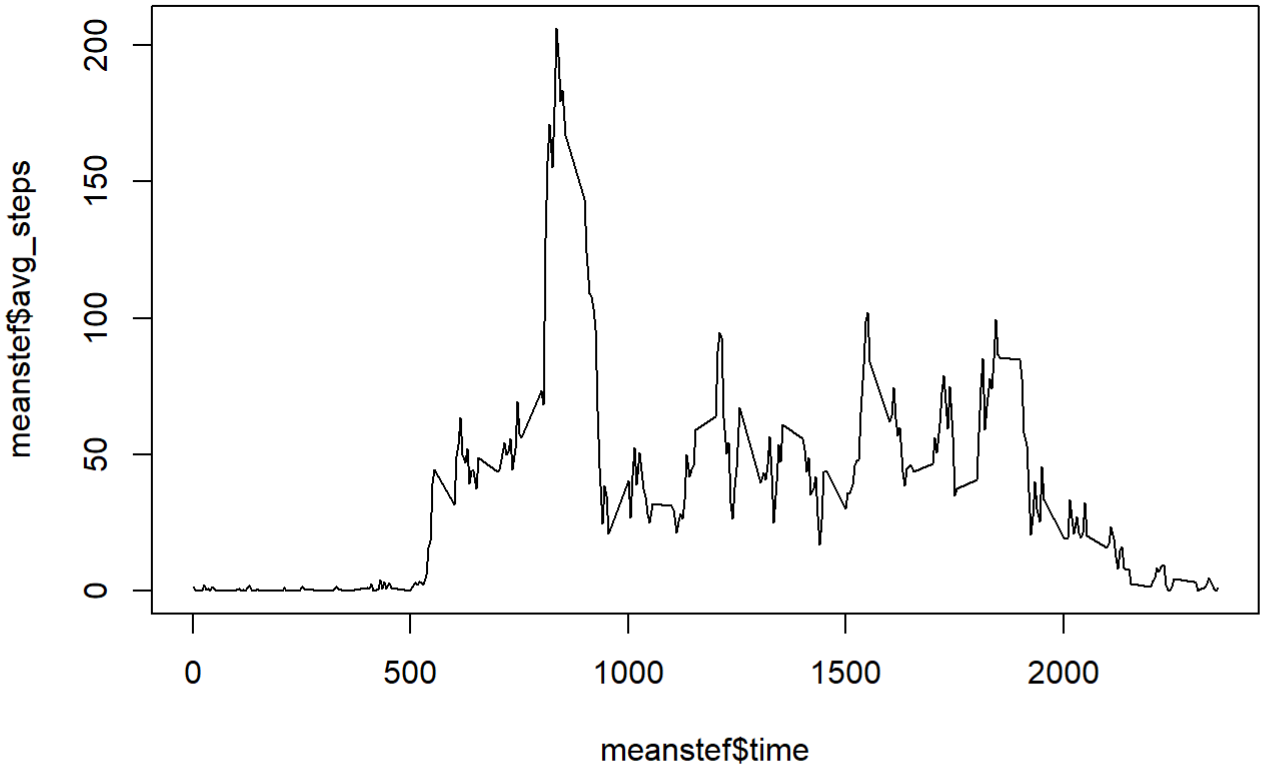
```
medistpd<- round(median(tnstpd$tn_steps))
print(paste('Median of the total number of steps taken per day =', medistpd))
```

```
## [1] "Median of the total number of steps taken per day = 10395"
```

What is the average daily activity pattern?

1. Make 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)

```
    clean_act <- na.omit(act)
    meanstef<- -aggregate(clean_act$steps, by=list(clean_act$interval), FUN = mean, na.rm=T
RUE)
    colnames(meanstef) <- c("time", "avg_steps")
    meanstef$time <-meanstef$time * (-1)
    meanstef$avg_steps <-meanstef$avg_steps * (-1)
    plot(meanstef$time, meanstef$avg_steps, type = "l")
```



2. Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps? difference between them. Make a histogram of the total number of steps taken each day

```
    meanstef[which.max(meanstef$avg_steps),]
```

```
##     time avg_steps
## 104  835  206.1698
```

Imputing missing values

Note that there are a number of days/intervals where there are missing values (coded as NA). The presence of missing days may introduce bias into some calculations or summaries of the data.

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

```
    sum(is.na(act$steps))
```

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

```
      colMeans(is.na.data.frame(act))
```

```
##     steps      date  interval
## 0.1311475 0.0000000 0.0000000
```

2. 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.

```
       act_fill_NA <- act
```

3. Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
     #We will use mean of steps from cleaning data to replace all NA in the original data
     #There're 8 days of missing data
     act_fill_NA$steps[is.na(act_fill_NA$steps)] <- round(mean(clean_act$steps))
     head(act_fill_NA)
```

```
##   steps       date interval
## 1    37 2012-10-01        0
## 2    37 2012-10-01        5
## 3    37 2012-10-01       10
## 4    37 2012-10-01       15
## 5    37 2012-10-01       20
## 6    37 2012-10-01       25
```

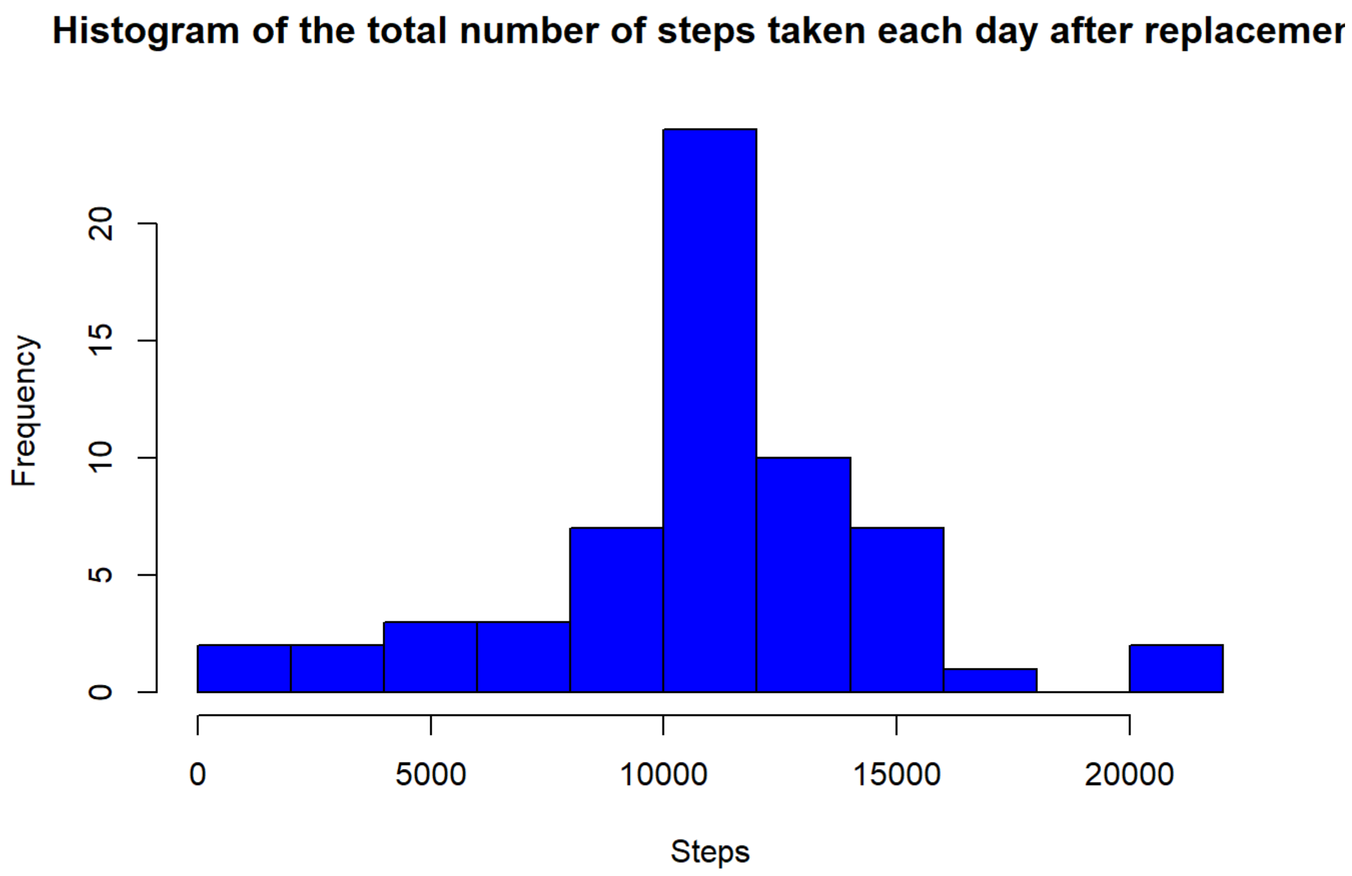
```
     str(act_fill_NA)
```

```
## 'data.frame':   17568 obs. of  3 variables:
##  $ steps   : num  37 37 37 37 37 37 37 37 37 37 ...
##  $ date    : chr  "2012-10-01" "2012-10-01" "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

4. 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?

```
     tnstpd_NA <-aggregate(act_fill_NA$steps, by=list(act_fill_NA$date), FUN = sum, na.rm=TRU
E)
     colnames(tnstpd_NA) <- c("date", "tn_steps")

     hist(tnstpd_NA$tn_steps, breaks =8, main = 'Histogram of the total number of steps taken
each day after replacement', xlab = 'Steps', col = 'blue')
```



```
     #Calculate and report the mean and median of the total number of steps taken per day
     meanstpd_NA <- round(mean(tnstpd_NA$tn_steps))
     print(paste('Mean of the total number of steps taken per day after replacement =', means
tpd_NA))
```

```
## [1] "Mean of the total number of steps taken per day after replacement = 10752"
```

```
     medistpd_NA<- round(median(tnstpd_NA$tn_steps))
     print(paste('Median of the total number of steps taken per day after replacement =', med
istpd_NA))
```

```
## [1] "Median of the total number of steps taken per day after replacement = 10656"
```

Are there differences in activity patterns between weekdays and weekends?

For this part the weekdays() function may be of some help here. Use the dataset with the filled-in missing values for this part.

1. Create a new factor variable in the dataset with two levels – “weekday” and “weekend” indicating whether a given date is a weekday or weekend day.

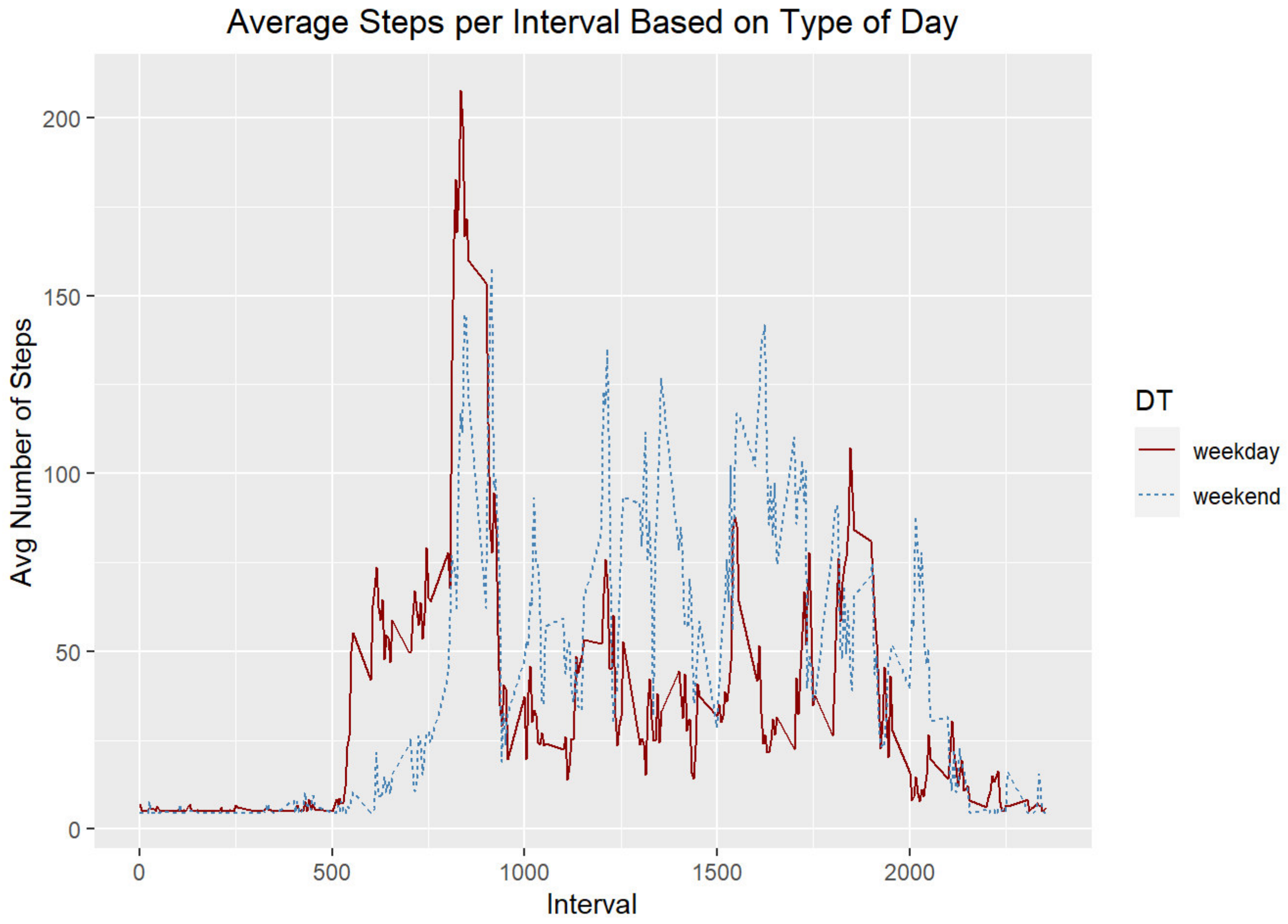
```
     act_fill_NA$DT <- ifelse(weekdays(as.Date(act_fill_NA$date)) %in% c("Saturday", "Sunda
y"), "weekend", "weekday")
     act_fill_NA$DT <- as.factor(act_fill_NA$DT)
     str(act_fill_NA)
```

```
## 'data.frame':   17568 obs. of  4 variables:
##  $ steps   : num  37 37 37 37 37 37 37 37 37 37 ...
##  $ date    : chr  "2012-10-01" "2012-10-01" "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
##  $ DT      : Factor w/ 2 levels "weekday","weekend": 1 1 1 1 1 1 1 1 1 1 ...
```

2. Make a panel plot containing 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 weekday days or weekend days (y-axis). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

```
interval_comp <- ddply(act_fill_NA, .(interval, DT), summarize, Avg = mean(steps))

ggplot(interval_comp, aes(x = interval, y = Avg)) +
  geom_line(aes(color = DT, linetype = DT)) +
  scale_color_manual(values = c("darkred", "steelblue")) +
  labs(x="Interval", y="Avg Number of Steps") +
  labs(title = "Average Steps per Interval Based on Type of Day")+
  theme(plot.title = element_text(hjust = 0.5))
```


Average Steps per Interval Based on Type of Day
Avg Number of Steps
Interval
DT
weekday
weekend