Untitled

Foopaul

2023-05-21

Tidyverse

```
library(tidyverse)
```

0. Loading and preprocessing the data

0.1 Load the data

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

0.2 Process/transform the data (if necessary)

```
activity$date <- as_date(activity$date)

weekday <- weekdays(activity$date)

activity <-cbind(activity, weekday)

summary(activity)
```

```
##      steps             date               interval        weekday         
##  Min.   :  0.00   Min.   :2012-10-01   Min.   :   0.0   Length:17568      
##  1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.: 588.8   Class :character  
##  Median :  0.00   Median :2012-10-31   Median :1177.5   Mode  :character  
##  Mean   : 37.38   Mean   :2012-10-31   Mean   :1177.5                     
##  3rd Qu.: 12.00   3rd Qu.:2012-11-15   3rd Qu.:1766.2                     
##  Max.   :806.00   Max.   :2012-11-30   Max.   :2355.0                     
##  NA's   :2304
```

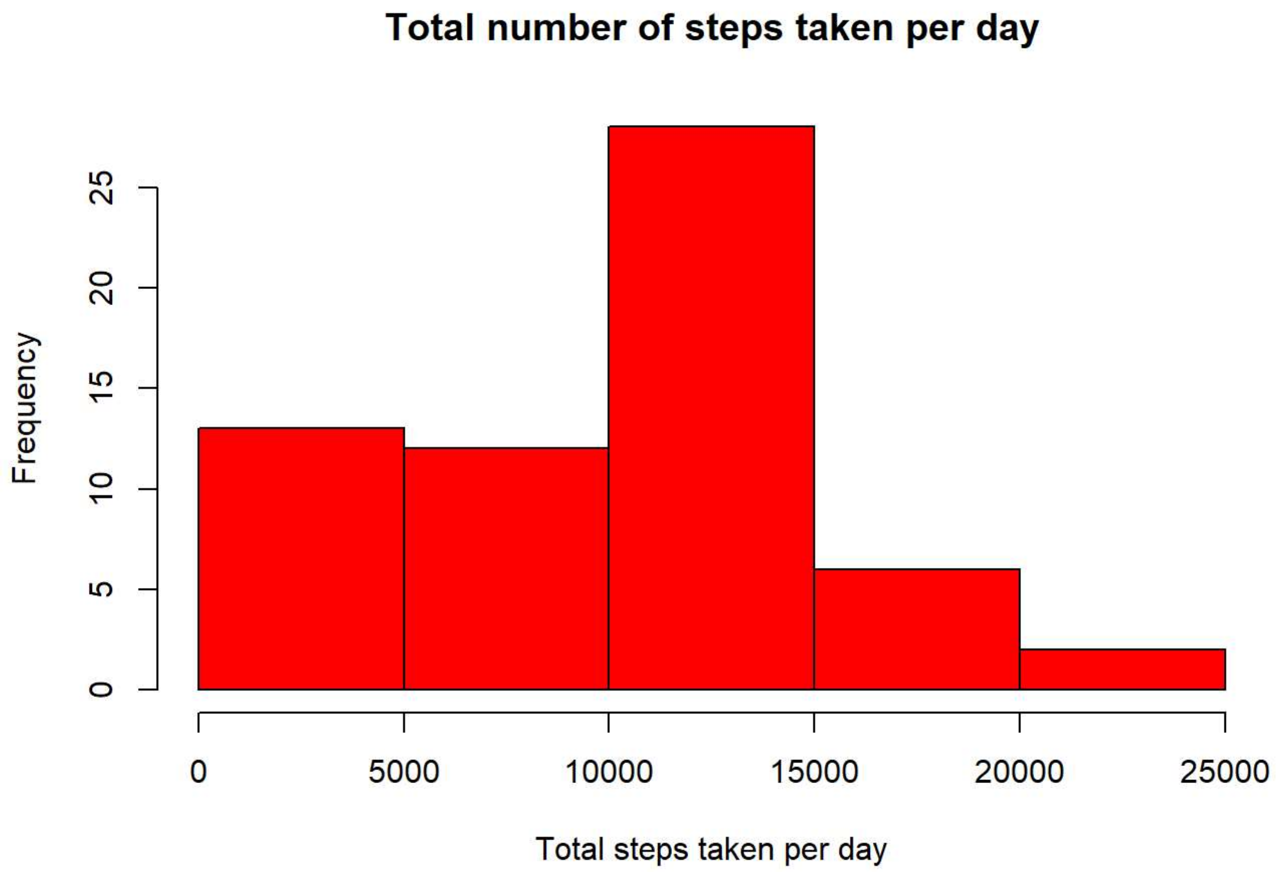
1. What is mean total number of steps taken per day?

1.1 Make a histogram of the total number of steps

taken each day

```
activity.tsteps<- with(activity,
                       aggregate(steps, by = list(date),
                                 FUN = sum, na.rm = TRUE))

names(activity.tsteps)<- c("dates", "steps")
```

```
hist(activity.tsteps$steps,
     main = "Total number of steps taken per day",
     xlab = "Total steps taken per day",
     col = "red")
```



1.2 Calculate and report the mean and median of the total number of steps taken per day

```
mean(activity.tsteps$steps)
```

```
## [1] 9354.23
```

```
median(activity.tsteps$steps)
```

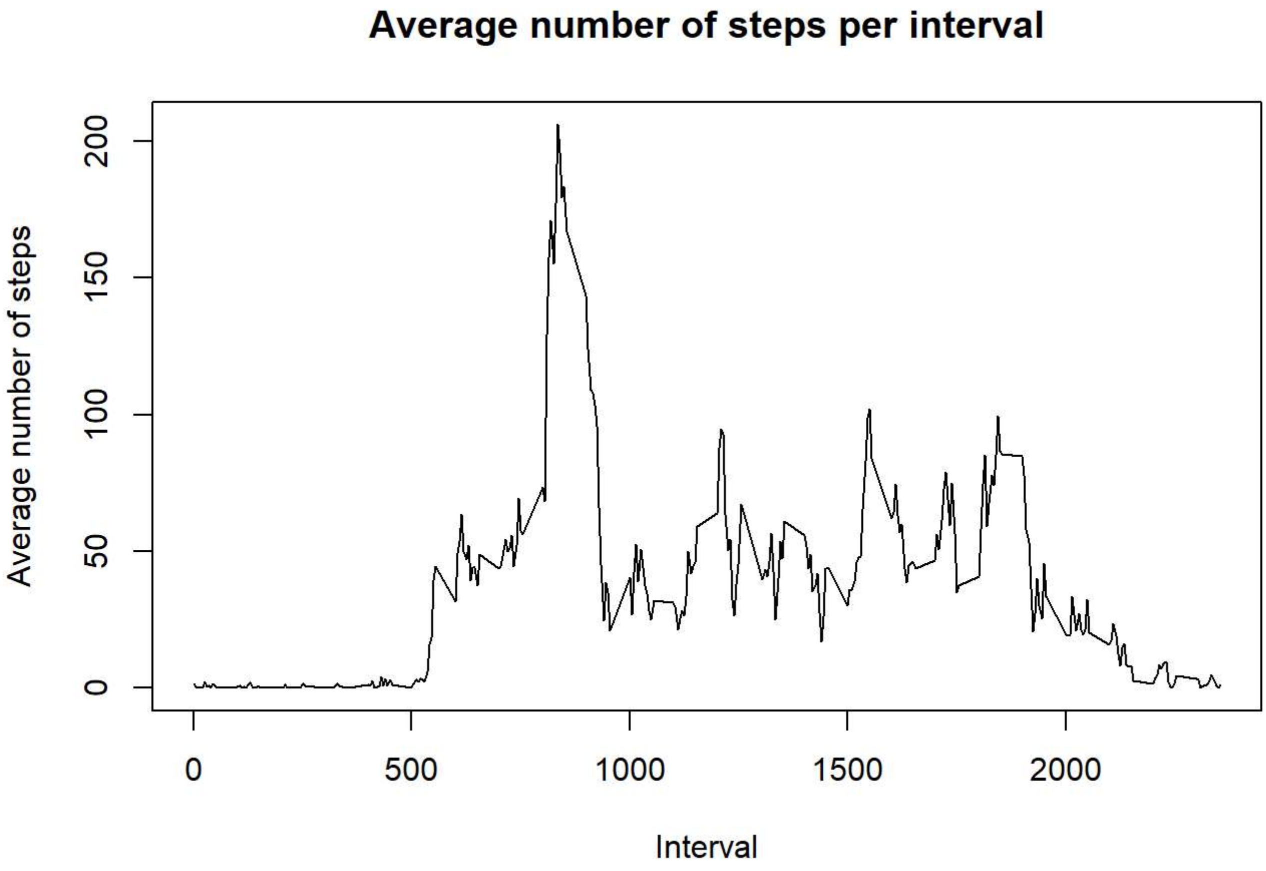
```
## [1] 10395
```

2. What is the average daily activity pattern?

2.1 Make a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)

```
average.daily.activity <- aggregate(activity$steps,
                                    by= list(activity$interval),
                                    FUN = mean , na.rm = TRUE)

names(average.daily.activity) <- c("interval", "mean")
```

```
plot(average.daily.activity$interval, average.daily.activity$mean,
     type = "l", xlab = "Interval",
     ylab = "Average number of steps",
     main = "Average number of steps per interval")
```



2.2 Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

```
average.daily.activity[which.max(average.daily.activity$mean),]$interval
```

```
## [1] 835
```

3. Imputing missing values

3.1 Calculate and report the total number of missing values in the dataset

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

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

3.2 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.

```
clean.steps<- average.daily.activity$mean[match(activity$interval,average.daily.activity$interval)]
```

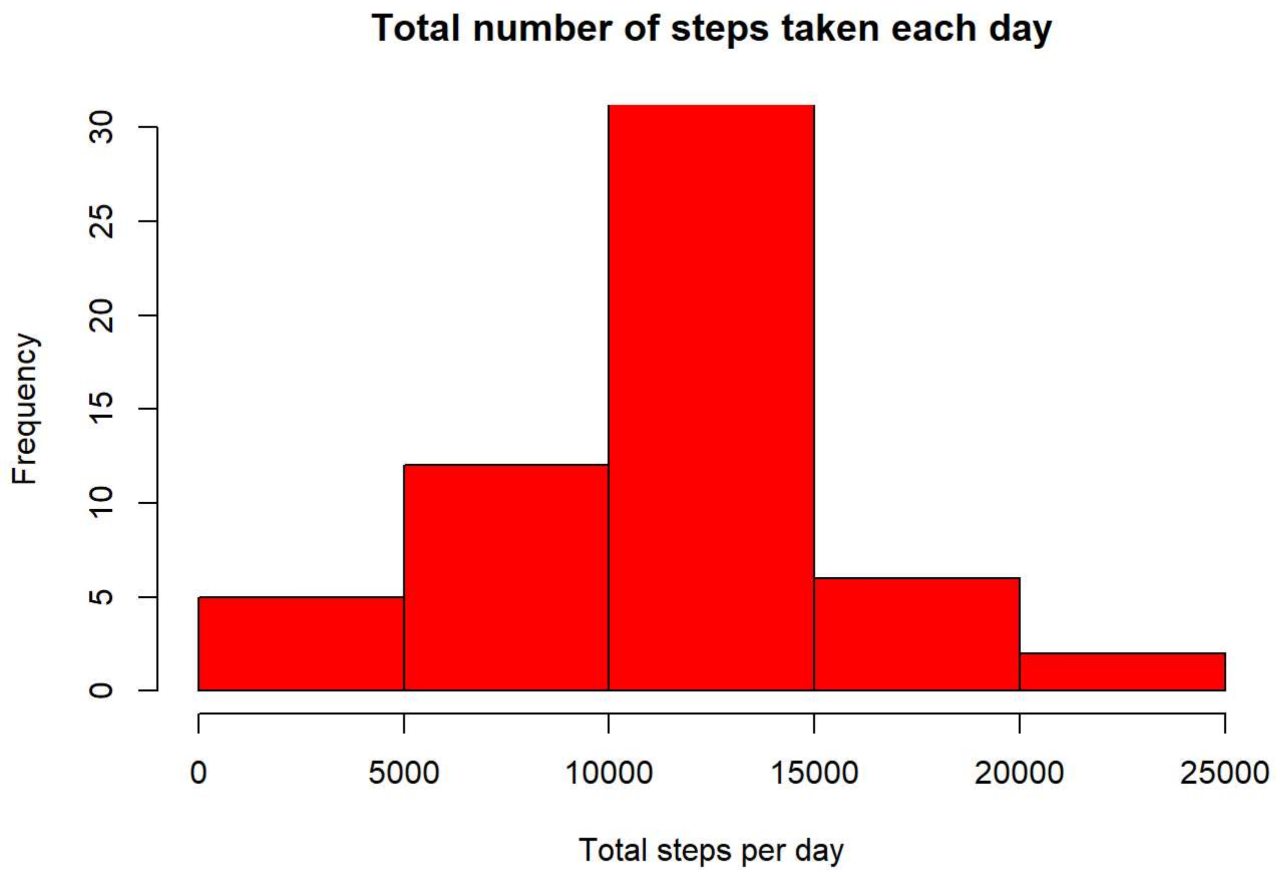
3.3 Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
activity.clean <- transform(activity, steps = ifelse(is.na(activity$steps), yes = clean.steps, no = activity$steps))

total.clean.steps<- aggregate(steps ~ date, activity.clean, sum)

names(total.clean.steps)<- c("date", "daily.steps")
```

3.4 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?

```
hist(total.clean.steps$daily.steps,
     col = "red",
     xlab = "Total steps per day",
     ylim = c(0,30),
     main = "Total number of steps taken each day")
```



```
mean(total.clean.steps$daily.steps)
```

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

```
median(total.clean.steps$daily.steps)
```

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

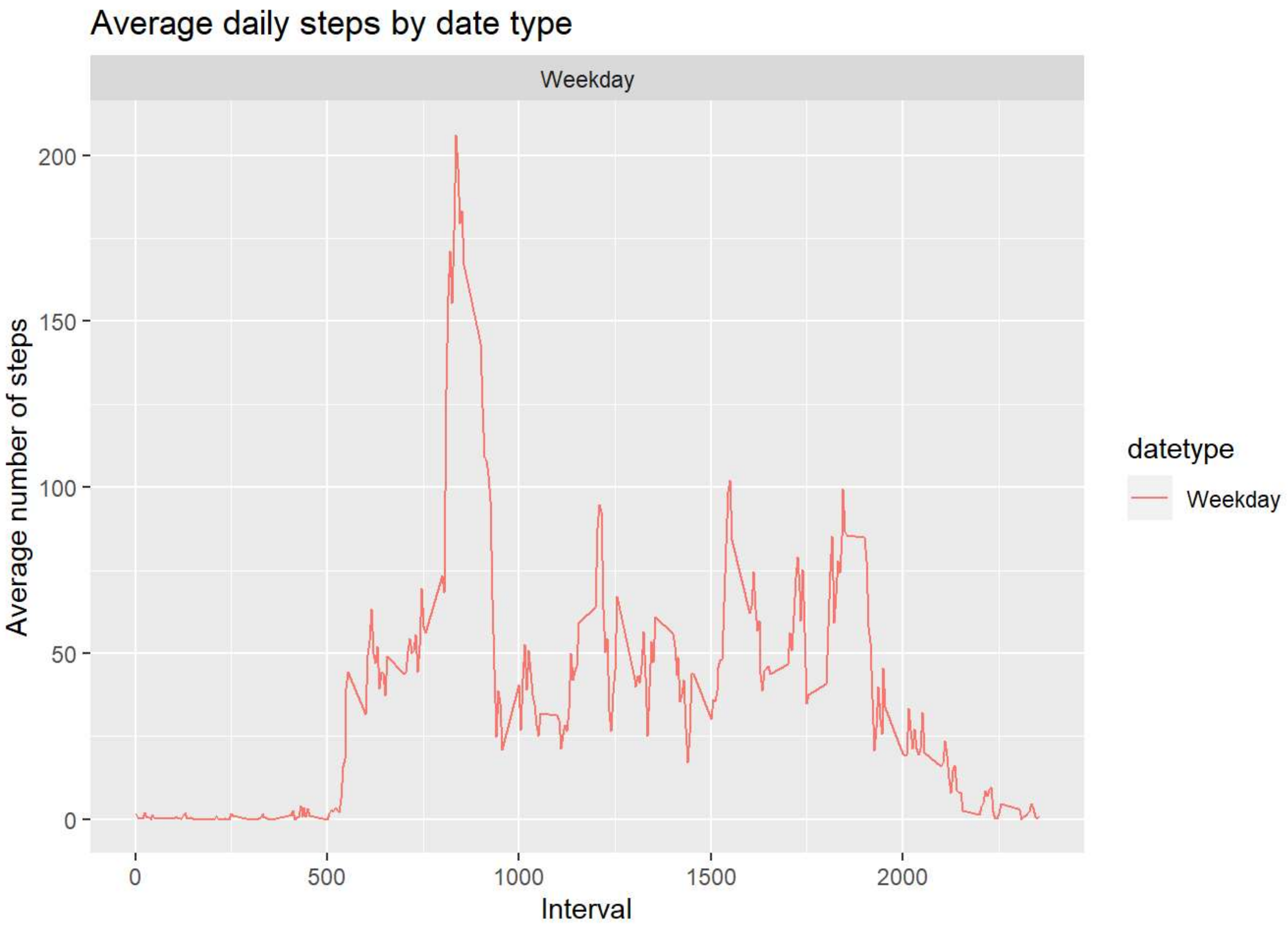
4. Are there differences in activity patterns between weekdays and weekends?

4.1 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.

```r
activity$datetype <- sapply(activity$date, function(x) {
  if (weekdays(x) == "Saturday" | weekdays(x) =="Sunday")
  {y <- "Weekend"} else
  {y <- "Weekday"}
  y
})
```

4.2 Make a panel plot containing a time series plot (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).

```r
activity.datetype<- aggregate(steps~interval+datetype, activity,mean, na.rm =TRUE)
ggplot(activity.datetype,
       aes(x = interval,
           y = steps,
           color = datetype))+
  geom_line() + labs(title = "Average daily steps by date type", x = "Interval", y = "Average
number of steps") + facet_wrap(~datetype, ncol = 1, nrow = 2)
```


Average daily steps by date type
Weekday
200
150
100
50
0
Average number of steps
0
500
1000
1500
2000
Interval
datetype
Weekday