Reproducible Research: Peer Assessment 1

Set global option, echo = TRUE, so that someone else will be able to read the code.

```
knitr::opts_chunk$set(echo=TRUE)
```

Loading and preprocessing the data

```
setwd("~/Desktop/coursera/reproducible-research/project1/RepData_PeerAssessment1")
dt<-read.csv("activity.csv")
str(dt)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1 1 1 1 1
...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

convert date to POSIXct using lubricate package

```
library(lubridate)
dt$date<-ymd(dt$date)
```

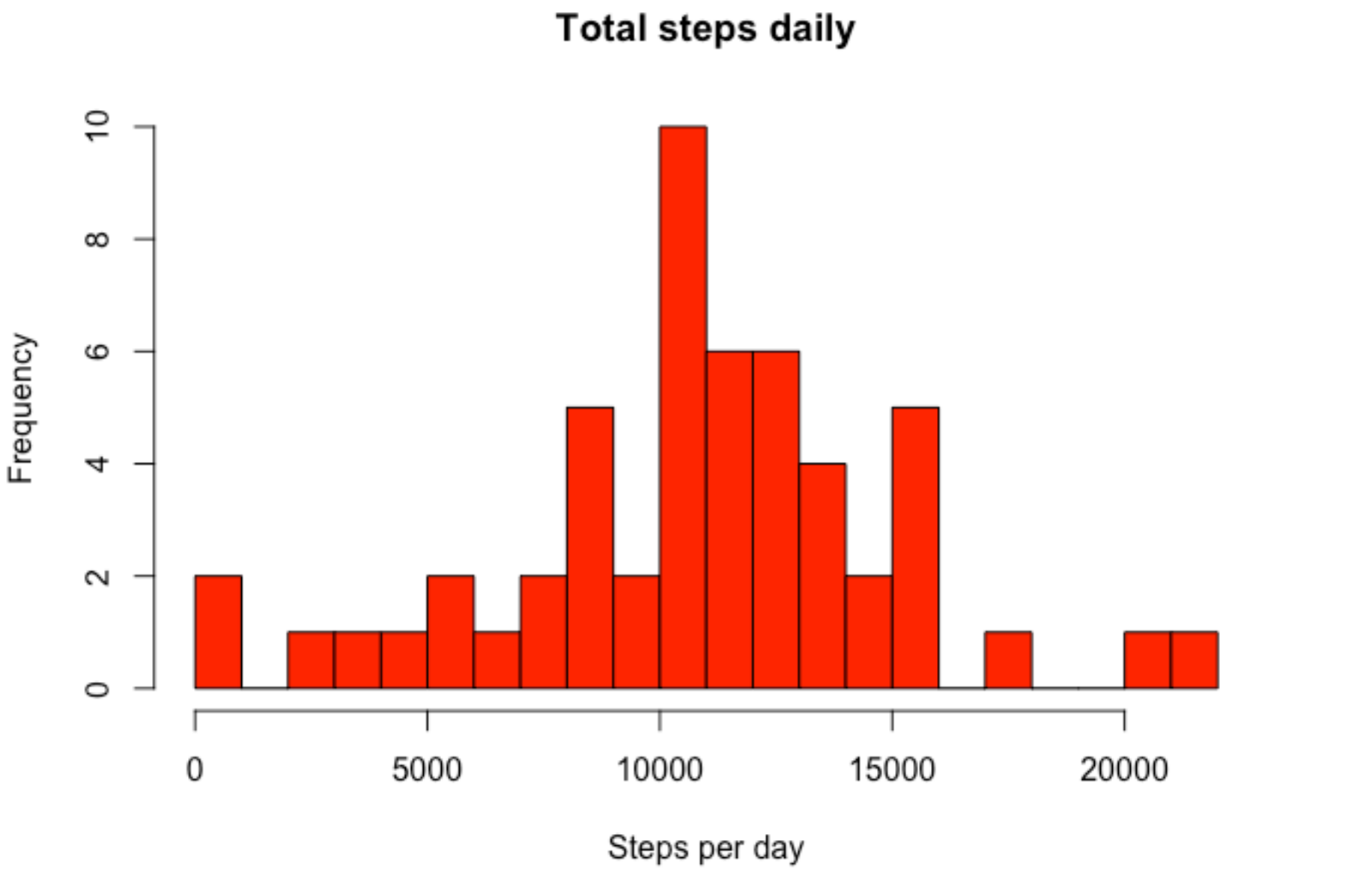
What is mean total number of steps taken per day? For this part of the assignment, you can ignore the missing values in the dataset.

1.Calculate the total number of steps taken per day

```
library(dplyr)
total_daily<-dt%>%group_by(date)%>%summarise(steps_daily=sum(steps,na.rm=TRUE),na=mea
n(is.na(steps)))
```

2.Histogram of the total number of steps taken each day

```
total_daily<-filter(total_daily,na<1)
hist(total_daily$steps_daily,col="red",breaks=20,main = "Total steps daily",xlab = "S
teps per day")
```



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

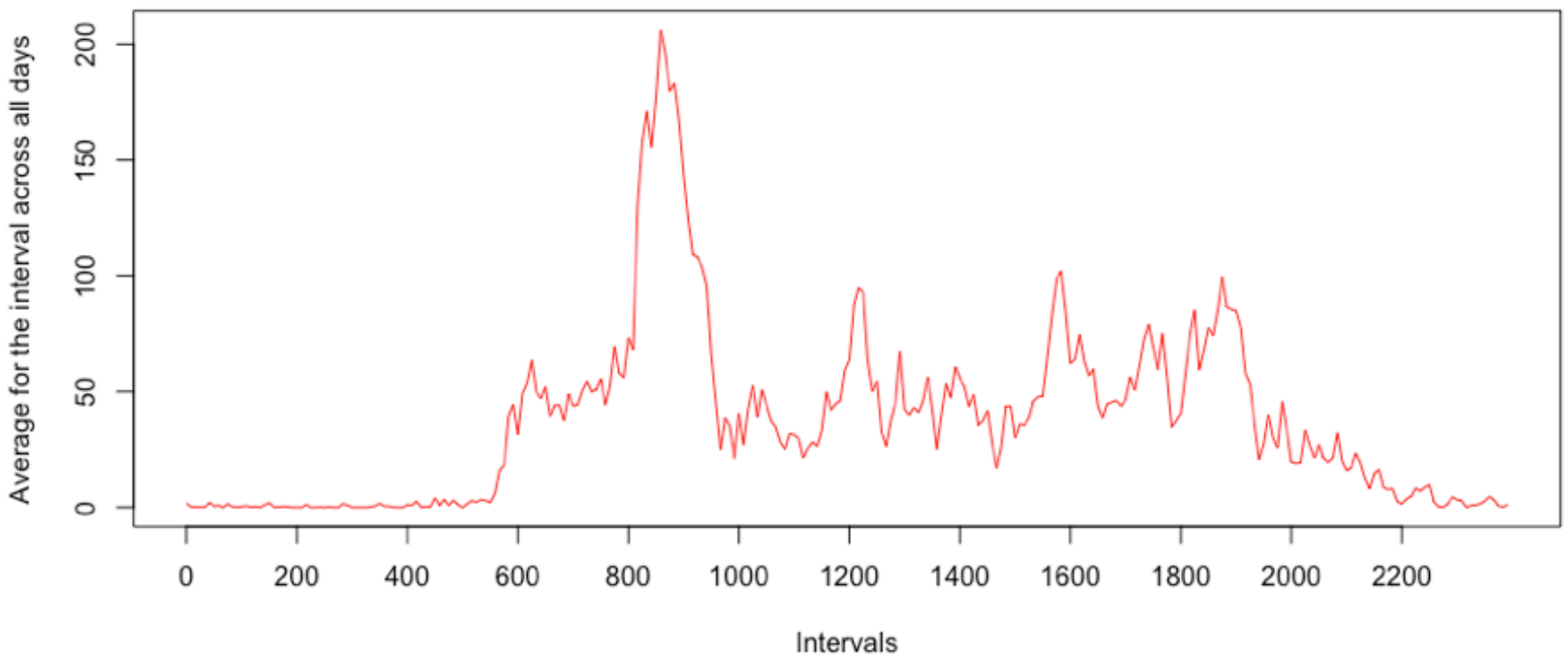
```
mean_steps<-mean(total_daily$steps_daily)
median_steps<-median(total_daily$steps_daily)
```

Mean and median of the total number of steps taken per day are 10766.19 steps,10765 steps,respectively.

What is the average daily activity pattern?

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)

```
library(dplyr,quietly = TRUE)
pattern_daily <- dt %>% group_by(interval) %>% summarise(average=mean(steps,na.rm=TRU
E))
plot(x = 1:nrow(pattern_daily),y = pattern_daily$average,type = "l",
     col = "red", xaxt = "n",xlab="Intervals",
     ylab = "Average for the interval across all days")
axis(1,labels=pattern_daily$interval[seq(1,288,24)],
     at = seq_along(pattern_daily$interval)[seq(1,288,24)])
```



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

```
Maxi_steps<-filter(pattern_daily,average==max(average))
```

Interval 835 contains on average the maximum of steps of 206

Imputing missing values

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

```
na_total <- sum(is.na(dt$steps))
na_total
```

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

```
na_percentage <- mean(is.na(dt$steps))
na_percentage
```

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

Total number of missing values in the dataset amounts to **2304** (which is **13.1** % of total observations).

2,3.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.

```
na_filling <- numeric(nrow(dt))
for (i in 1:nrow(dt))
{
        if (is.na(dt[i,"steps"])==TRUE)
            {
                    na_filling[i]<-filter(pattern_daily,interval==dt[i,"interval"]) %
>% select(average)
            }
        else
            {
                    na_filling[i]<-dt[i,"steps"]
            }

}
activity_without_NAs<-mutate(dt,steps_no_NAs=na_filling)
head(activity_without_NAs)
```

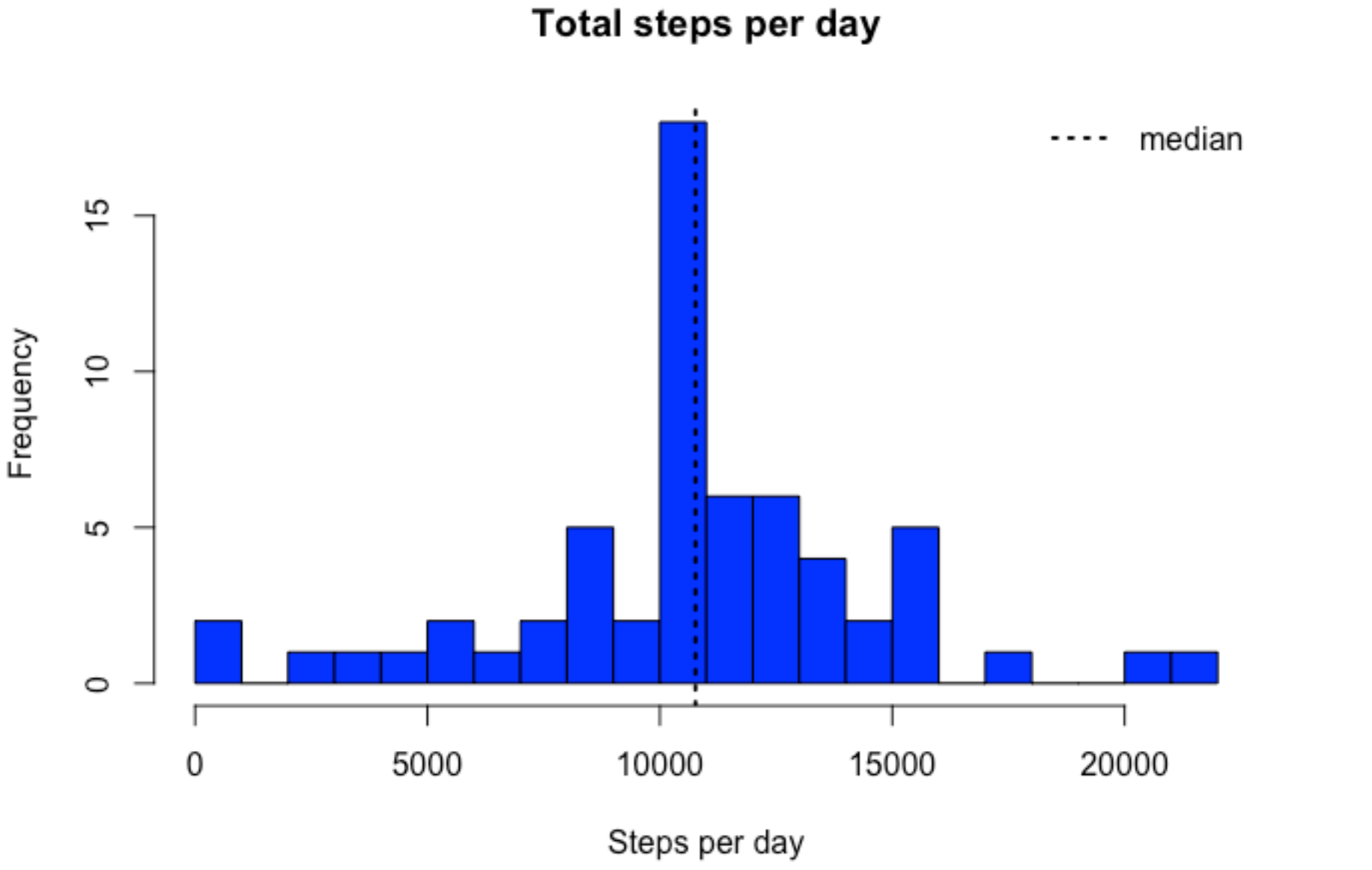
```
##   steps       date interval steps_no_NAs
## 1    NA 2012-10-01        0     1.716981
## 2    NA 2012-10-01        5    0.3396226
## 3    NA 2012-10-01       10    0.1320755
## 4    NA 2012-10-01       15    0.1509434
## 5    NA 2012-10-01       20    0.0754717
## 6    NA 2012-10-01       25      2.09434
```

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?

```
noNA_daily <- activity_without_NAs %>% mutate(steps_no_NAs=as.numeric(na_filling)) %>
% group_by(date) %>% summarise(total_steps=sum(steps_no_NAs))
head(noNA_daily)
```

```
## # A tibble: 6 x 2
##   date       total_steps
##   <date>           <dbl>
## 1 2012-10-01      10766.
## 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
```

```
hist(noNA_daily$total_steps,col="blue",breaks=20,main="Total steps per day",xlab="Ste
ps per day")
abline(v=median(noNA_daily$total_steps),lty=3, lwd=2, col="black")
legend(legend="median","topright",lty=3,lwd=2,bty = "n")
```




```
summary(noNA_daily$total_steps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      41    9819   10766   10766   12811   21194
```

Imputing missing values, mean and median of the total number of steps taken per day are similar with the estimates from the first part (ingoring missing values).

Are there differences in activity patterns between weekdays and weekends?

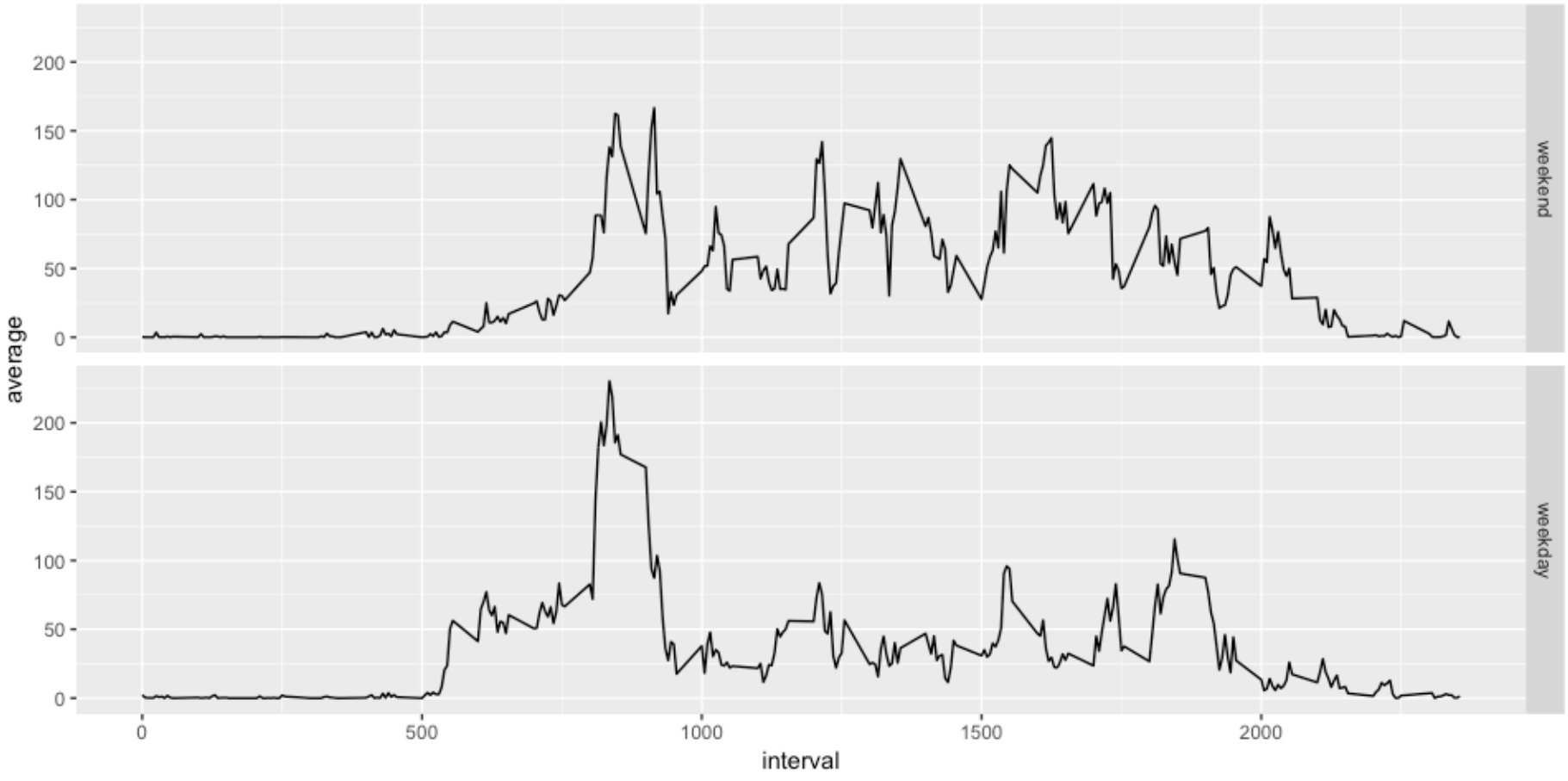
1 Create a new factor variable in the dataset with two levels – “weekday” and “weekend” indicating whether a given date is a weekday or weekend day.

```
library(lubridate)
is_weekday <-function(date){
        if(wday(date)%in%c(1,7)) result<-"weekend"
        else
                result<-"weekday"
        result
}
activity_without_NAs <- mutate(activity_without_NAs,date=ymd(date)) %>% mutate(day=sa
pply(date,is_weekday))
table(activity_without_NAs$day)
```

```
##
## weekday weekend
##   12960    4608
```

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). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

```
library(ggplot2)
daily_patterns <- activity_without_NAs %>% mutate(day=factor(day,levels=c("weekend","
weekday")),steps_no_NAs=as.numeric(steps_no_NAs)) %>% group_by(interval,day) %>% summ
arise(average=mean(steps_no_NAs))
qplot(interval,average,data=daily_patterns,geom="line",facets=day~.)
```


average
200
150
100
50
0
weekend
weekday
0
500
1000
1500
2000
interval