

0. Loading and preprocessing the data

0.1. Read activity.csv file

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

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

1.1. Calculate the total number of steps taken per day

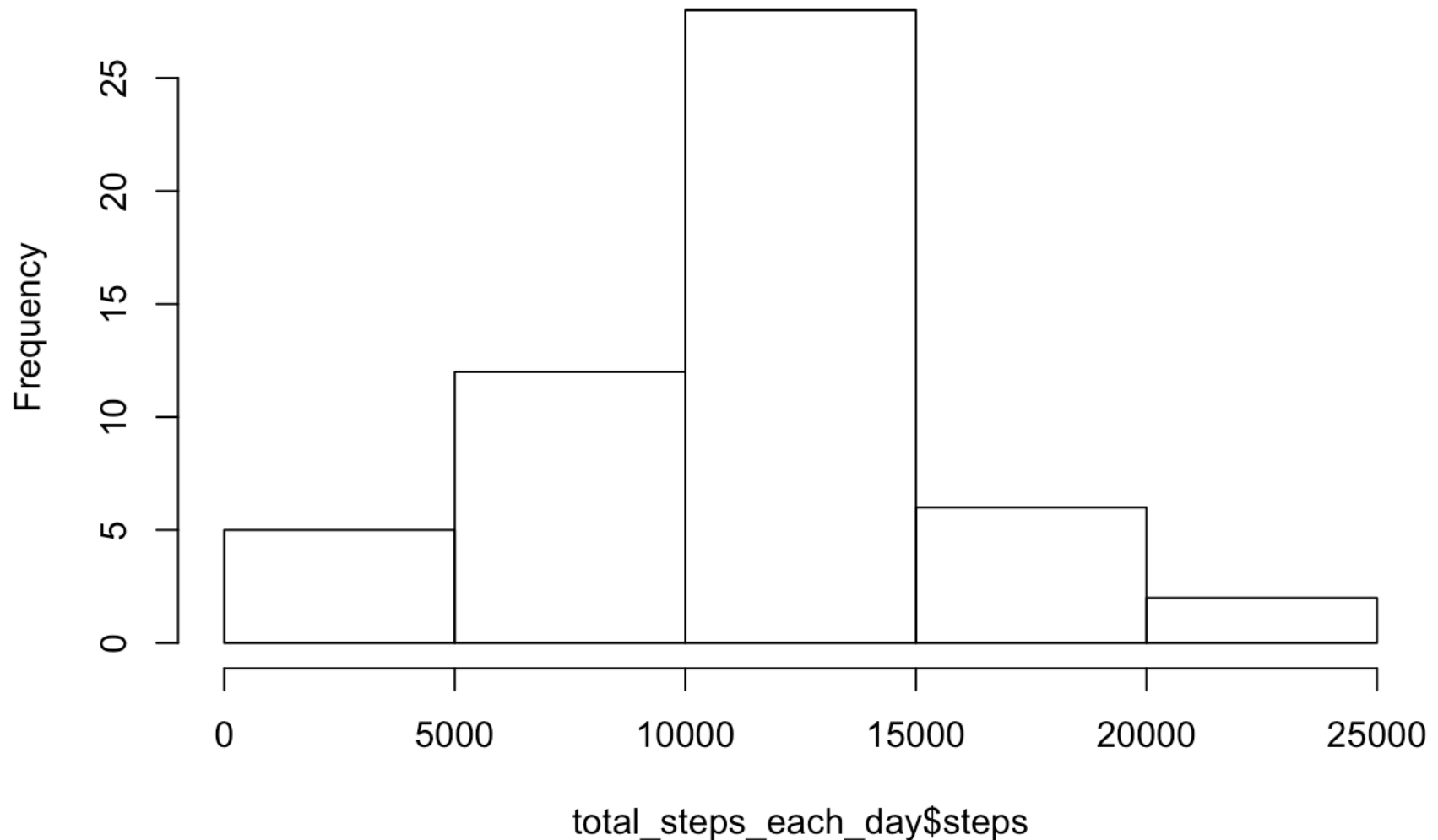
```
total_of_steps_per_day <- sum(activity$steps, na.rm = TRUE)  
total_of_steps_per_day
```

```
## [1] 570608
```

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

```
total_steps_each_day <- aggregate(steps~date, data=activity, FUN=sum, na.rm=TRUE)  
hist(total_steps_each_day$steps)
```

Histogram of total_steps_each_day\$steps



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

```
total_steps_each_day_mean <- mean(total_steps_each_day$steps)
total_steps_each_day_mean
```

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

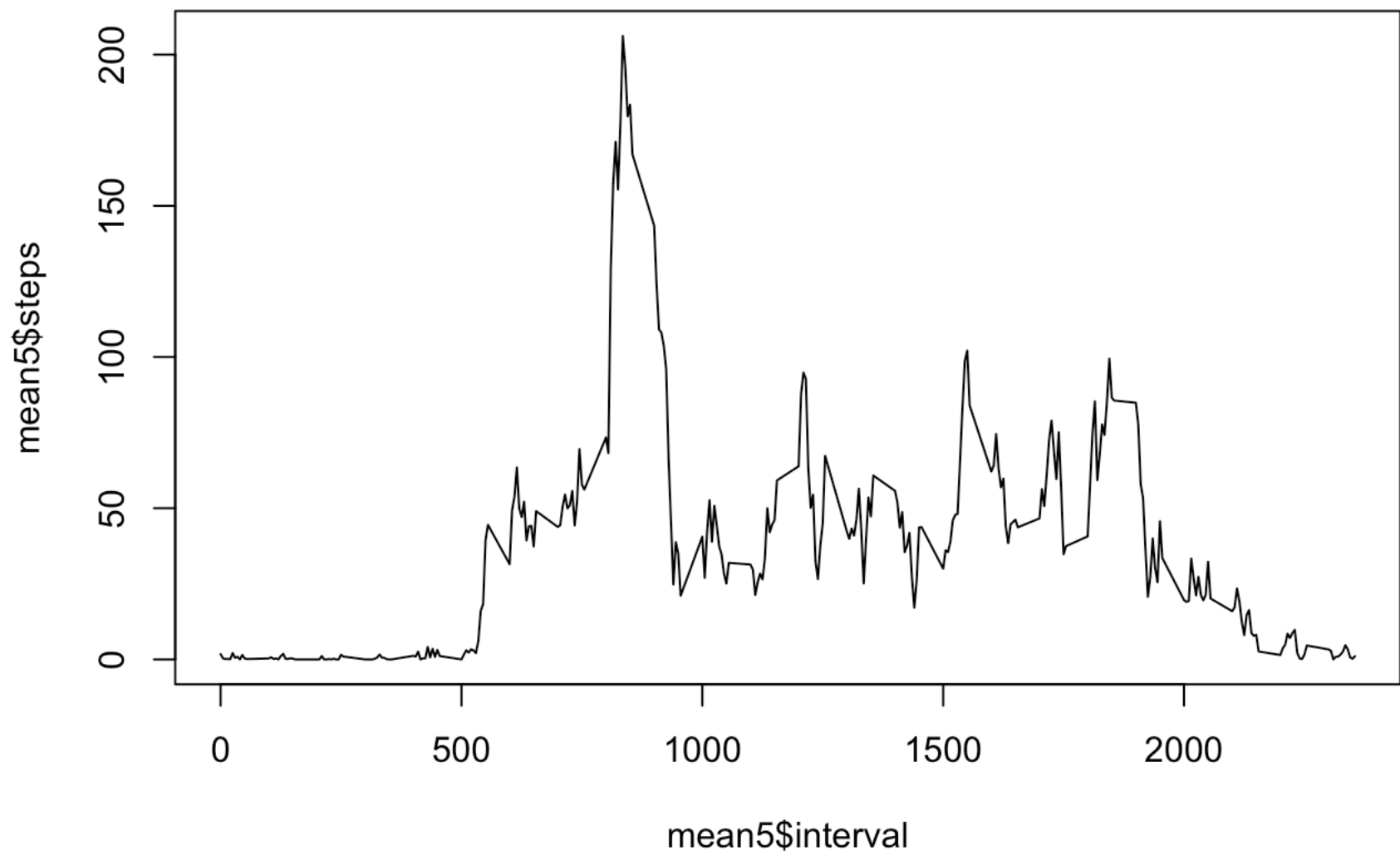
```
total_steps_each_day_median <- median(total_steps_each_day$steps)
total_steps_each_day_median
```

```
## [1] 10765
```

2. What is the average daily activity pattern?

2.1. 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)

```
mean5 <- aggregate(steps~interval, data=activity, FUN=mean, na.rm=TRUE)
plot(x = mean5$interval, y = mean5$steps, type = "l")
```



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

```
max_steps <- max(mean5$steps)
for (i in 1:288)
{
  if (mean5$steps[i] == max_steps)
    five_minute_interval_at_max_steps <- mean5$interval[i]
}
five_minute_interval_at_max_steps
```

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

3. Imputing missing values

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

```
total_na <- 0
for (i in 1:17568)
{
  if(is.na(activity$steps[i]))
    total_na <- total_na+1
}
total_na
```

```
## [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.

Answer - Fill mean for the 5-minute interval replacing the missing values

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

```

activity_missing_filled <- activity

for (i in 1:17568)
{
  if(is.na(activity_missing_filled$steps[i]))
  ## if NA/missing value found
  {
    ## storing the missing indices
    x <- activity_missing_filled$interval[i]
    for (j in 1:288)
    {
      if (mean5$interval[j] == x)
      ## replacing NA by 5 minute interval mean
      activity_missing_filled$steps[i] <- mean5$steps[j]
    }
  }
}

```

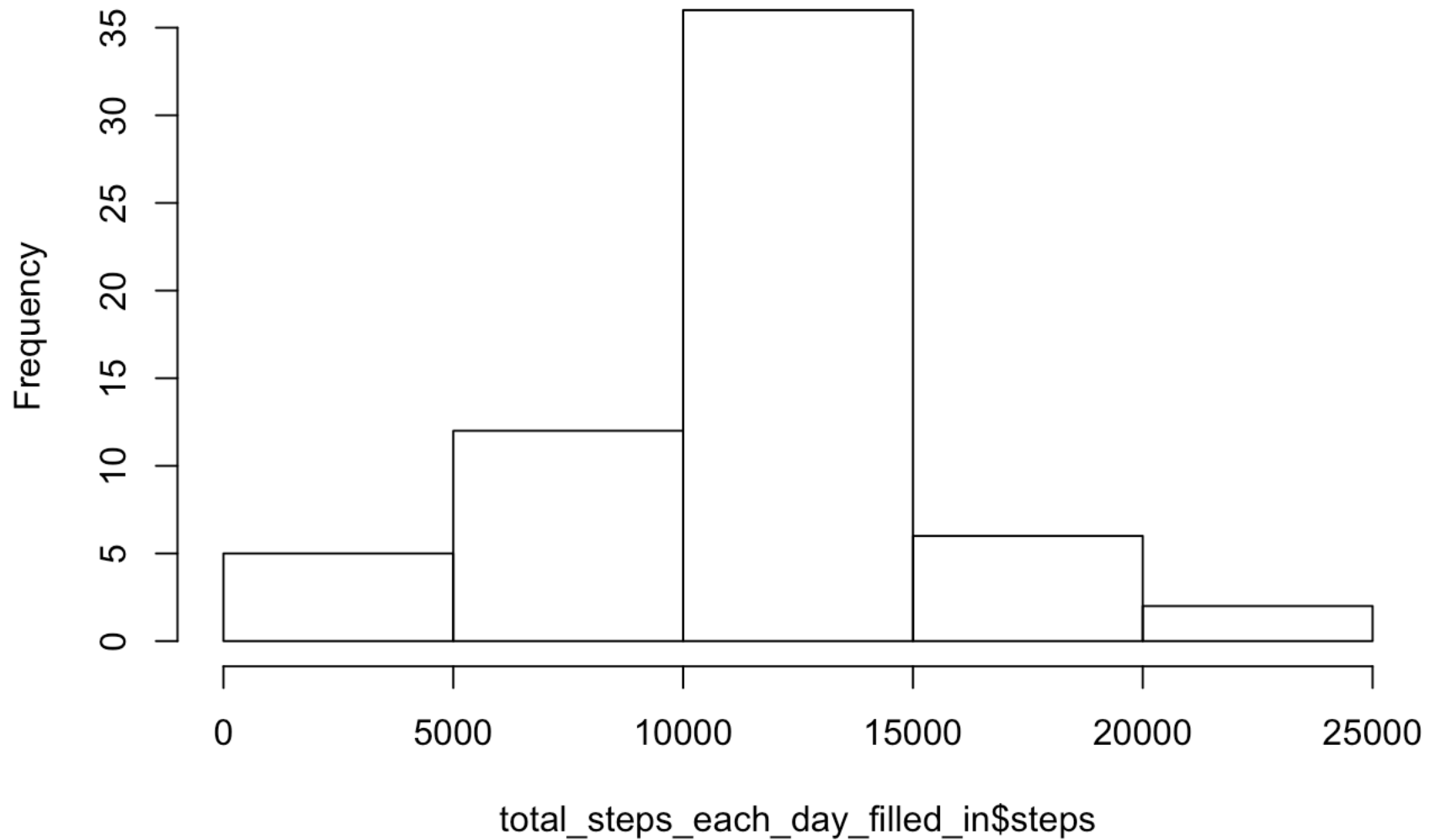
3.4. Make a histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day. Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

```

total_steps_each_day_filled_in <- aggregate(steps~date, data=activity_missing_filled,
FUN=sum, na.rm=TRUE)
hist(total_steps_each_day_filled_in$steps)

```

Histogram of total_steps_each_day_filled_in\$steps



```
total_steps_each_day_mean_filled_in <- mean(total_steps_each_day_filled_in$steps)
total_steps_each_day_mean_filled_in
```

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

```
total_steps_each_day_median_filled_in <- median(total_steps_each_day_filled_in$steps)
total_steps_each_day_median_filled_in
```

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

Answer - No impact as we used mean value to replace the missing values

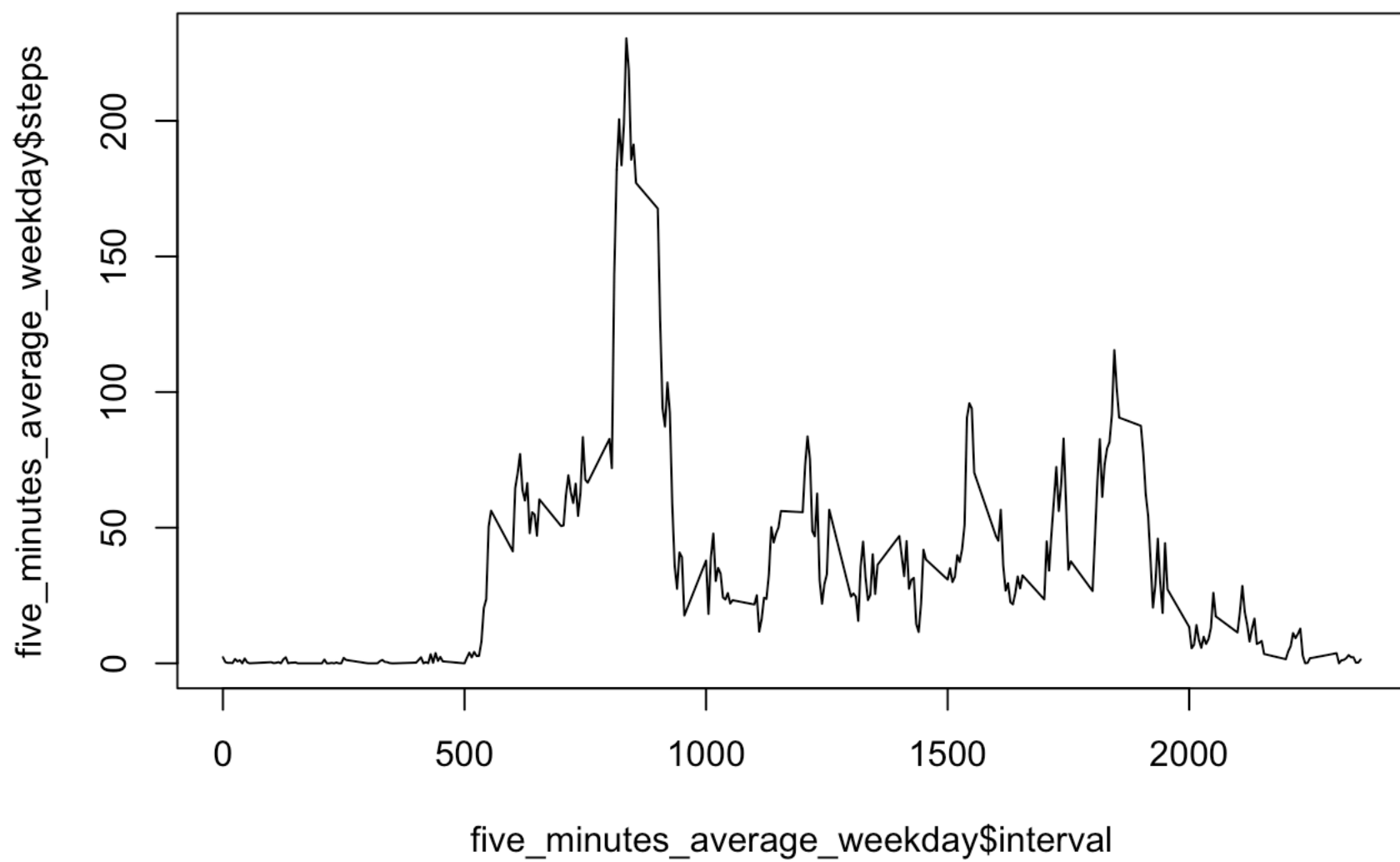
4. Are there differences in activity patterns between weekdays and weekends?

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.

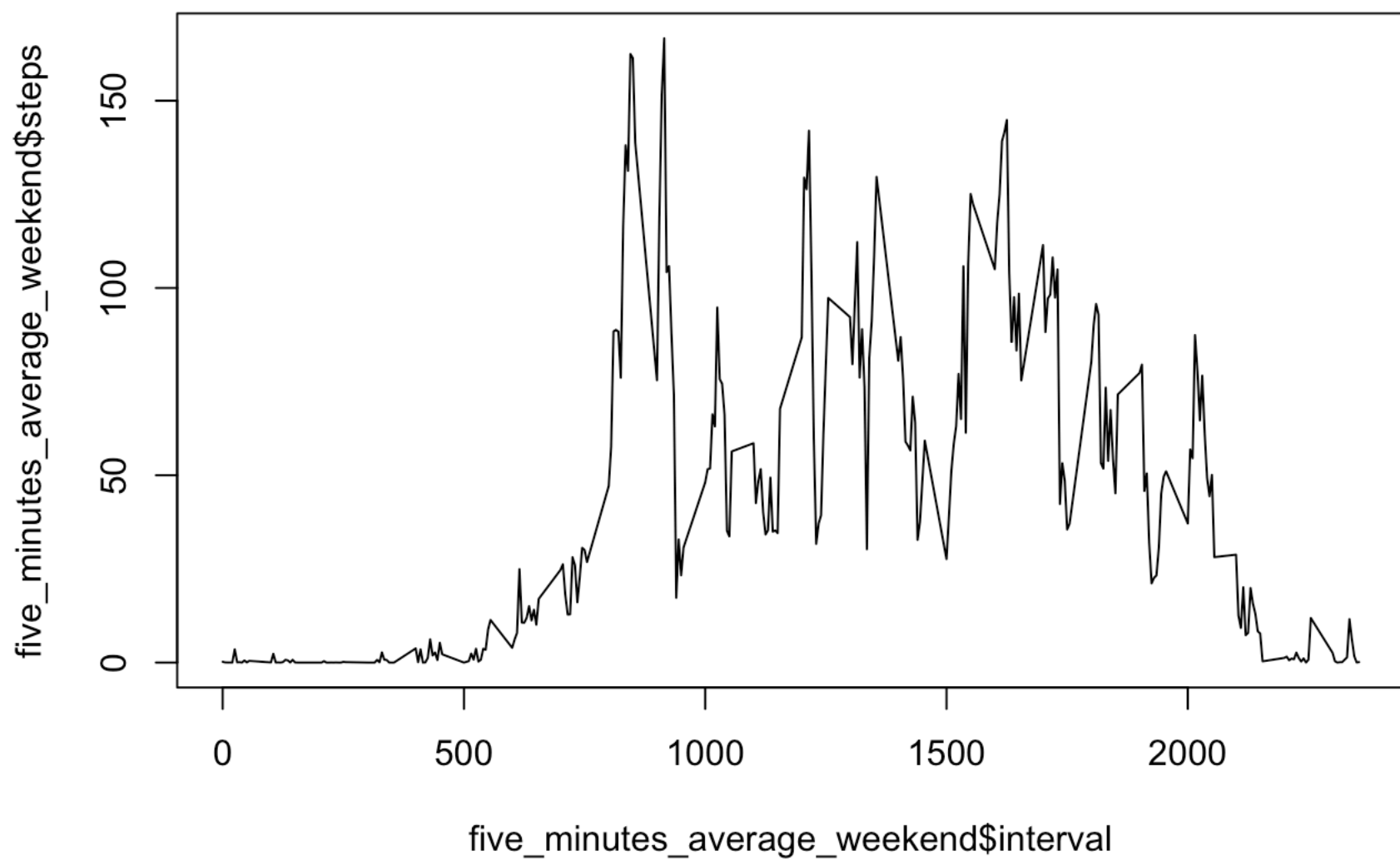
```
dayofweek <- function(date)
{
  if (weekdays(as.Date(date)) %in% c("Saturday", "Sunday"))
  {
    "weekend"
  } else {
    "weekday"
  }
}
activity_missing_filled$daytype <- as.factor(sapply(activity_missing_filled$date, day
ofweek))
```

4.2. 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.

```
weekday <- grep("weekday",activity_missing_filled$daytype)
weekday_frame <- activity_missing_filled[weekday,]
weekend_frame <- activity_missing_filled[-weekday,]
five_minutes_average_weekday <- aggregate(steps~interval, data=weekday_frame, FUN=mea
n, na.rm=TRUE)
five_minutes_average_weekend <- aggregate(steps~interval, data=weekend_frame, FUN=mea
n, na.rm=TRUE)
plot(x = five_minutes_average_weekday$interval, y = five_minutes_average_weekday$step
s, type = "l")
```



```
plot(x = five_minutes_average_weekend$interval, y = five_minutes_average_weekend$steps, type = "l")
```

KM7