PA1_template.html

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------------------- tidyverse 1.2.
```

```
## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  2.1.1     v dplyr   0.8.1
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.3.0
```

```
## -- Conflicts ------------------------------------------------------------------ tidyverse_conflicts(
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(lubridate)
```

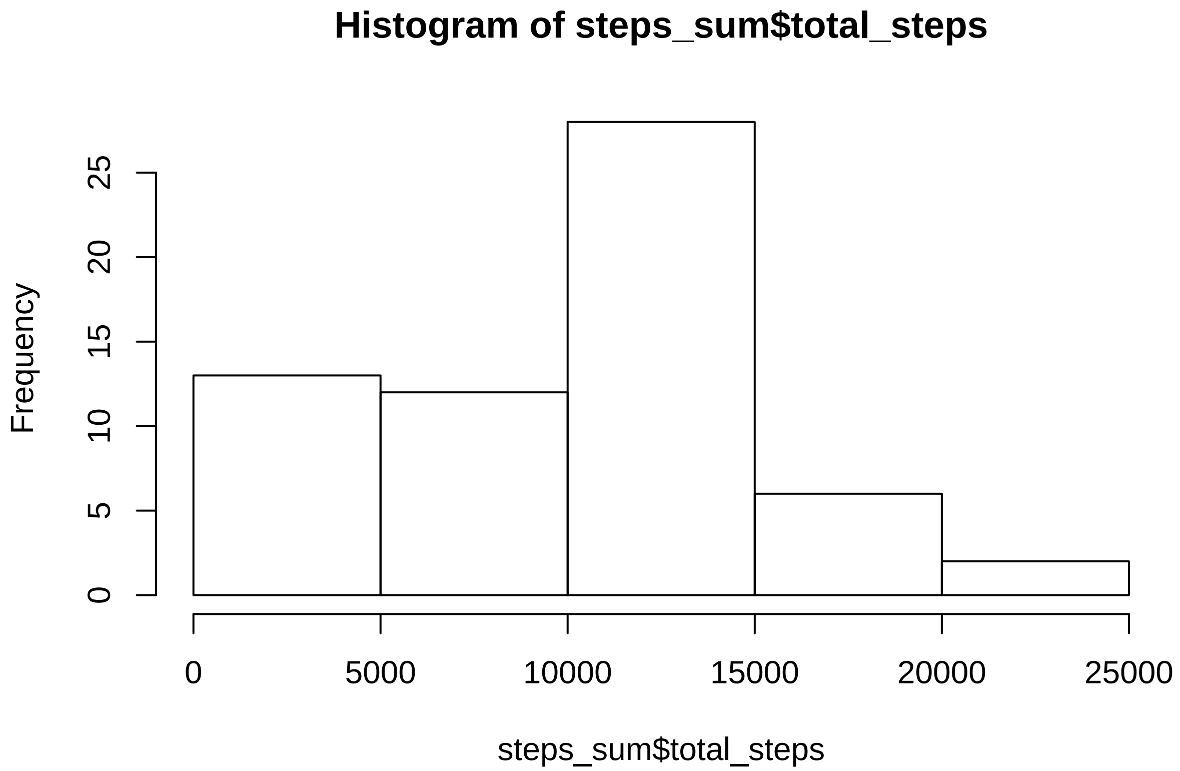
```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```r
data<-read.csv("activity.csv")
summary(data)
```

```
##      steps                date          interval
##  Min.   :  0.00   2012-10-01:  288   Min.   :   0.0
##  1st Qu.:  0.00   2012-10-02:  288   1st Qu.: 588.8
##  Median :  0.00   2012-10-03:  288   Median :1177.5
##  Mean   : 37.38   2012-10-04:  288   Mean   :1177.5
##  3rd Qu.: 12.00   2012-10-05:  288   3rd Qu.:1766.2
##  Max.   :806.00   2012-10-06:  288   Max.   :2355.0
##  NA's   :2304     (Other)   :15840
```

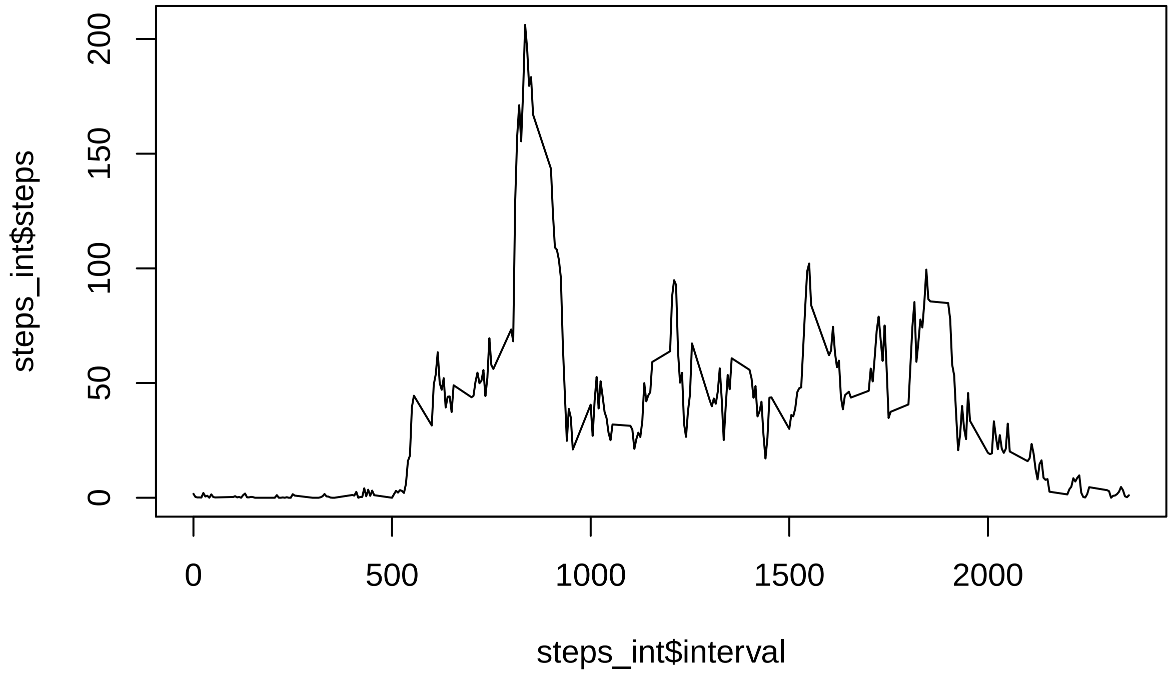
```r
##Calculate the total number of steps taken per day
steps_sum<-data%>% group_by(date)%>% summarise(total_steps=sum(steps, na.rm=TRUE))
## Make a histogram of the total number of steps taken each day
hist(steps_sum$total_steps)
```



```
##Calculate and report the mean and median of the total number of steps taken per day
steps_sum %>% summarise(mean_steps=mean(total_steps, na.rm=TRUE), median_steps=median(total_steps, na.r
```

```
## # A tibble: 1 x 2
##   mean_steps median_steps
##        <dbl>        <int>
## 1      9354.        10395
```

```
##Make a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, avera
steps_int<-data%>% group_by(interval)%>% summarise(steps=mean(steps, na.rm=TRUE))
plot(steps_int$interval, steps_int$steps, type="l")
```



```
##Which 5-minute interval, on average across all the days in the dataset, contains the maximum number o
steps_int$interval[steps_int$steps==max(steps_int$steps)]
```

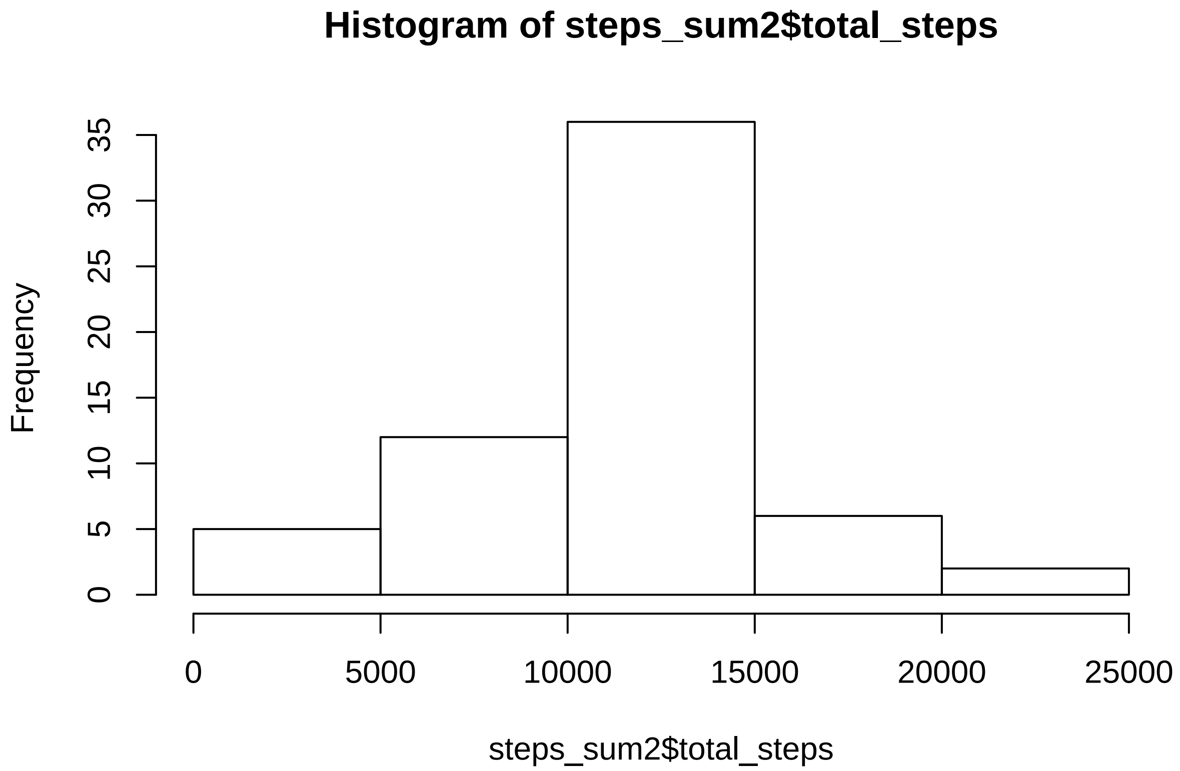
```
## [1] 835
```

```
##Calculate and report the total number of missing values in the dataset
sum(is.na(data$steps))
```

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

```
## Fill missing data with the mean for that 5-minute interval, etc.
##Create a new dataset that is equal to the original dataset but with the missing data filled in.
data2<-merge(data, steps_int, by="interval")
data2[is.na(data2$steps.x), "steps.x"]<-data2[is.na(data2$steps.x), "steps.y"]
data2<-data2[,1:3]
colnames(data2)<-c("interval", "steps", "date")

##Make a histogram of the total number of steps taken each day and Calculate and report the mean and me
steps_sum2<-data2%>% group_by(date)%>% summarise(total_steps=sum(steps, na.rm=TRUE))
hist(steps_sum2$total_steps)
```



```r
steps_sum2 %>% summarise(mean_steps=mean(total_steps, na.rm=TRUE), median_steps=median(total_steps, na.
```

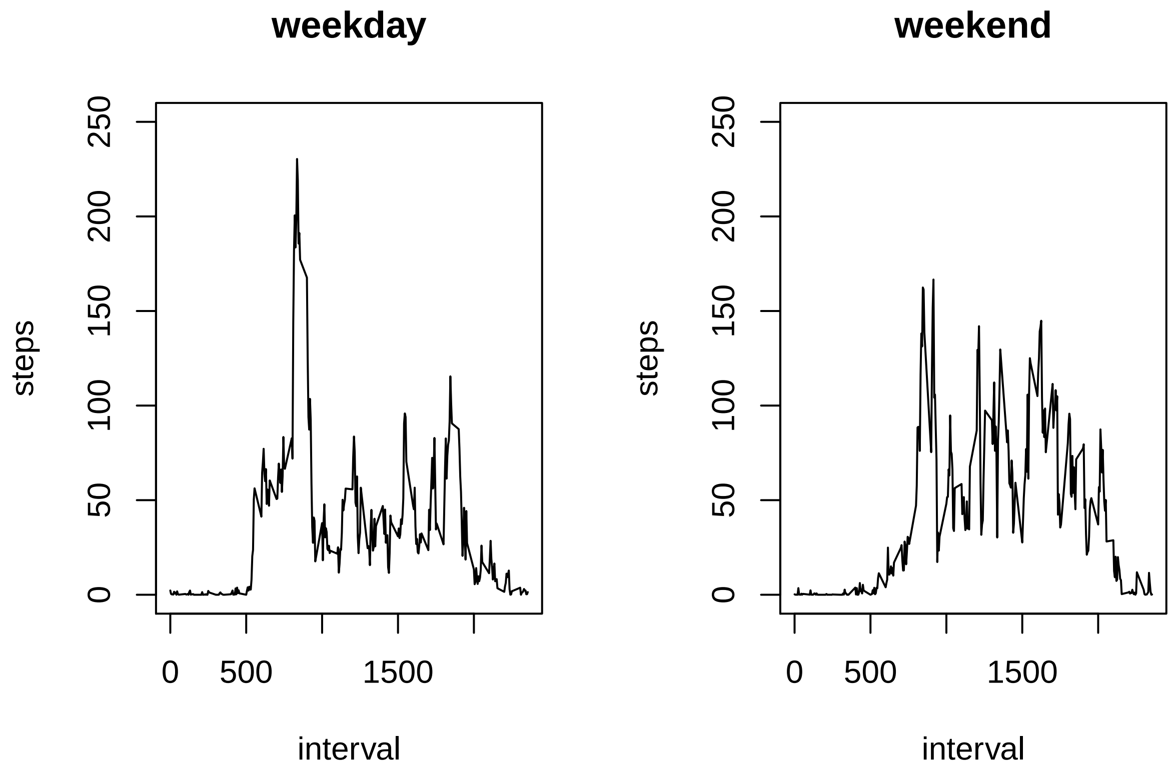
```
## # A tibble: 1 x 2
##   mean_steps median_steps
##        <dbl>        <dbl>
## 1     10766.       10766.
```

The mean and median increased, the histogram looks more normal

```r
data2$daytype<-factor(ifelse(wday(ymd(data2$date)) %in% c(1,7), 2, 1), labels=c("weekday", "weekend"))

steps_int2<-data2 %>% group_by(interval, daytype)%>% summarise(steps=mean(steps, na.rm=TRUE))


par(mfrow = c(1,2))
with(steps_int2%>% filter(daytype=="weekday"), plot(interval, steps, type="l",ylim=c(0,250), main="weekd
with(steps_int2%>% filter(daytype=="weekend"), plot(interval, steps, type="l", ylim=c(0,250), main="weel
```



Weekday has higher steps than weekend.