

# Activity Data

Me

7/07/2021

Loads data and replaces NA values with null value.

```
activity <- read.csv("activity.csv", header=TRUE)
activity$interval <- as.numeric(activity$interval)
activity$steps <- as.numeric(activity$steps)
activity$date <- as.Date(activity$date)
activity[is.na(activity)] <- 0
```

Groups and sums steps taken over 5 minute intervals by date, plots histogram of daily steps, and takes summary statistics to calculate mean and median of daily steps taken.

```
library(dplyr)
```

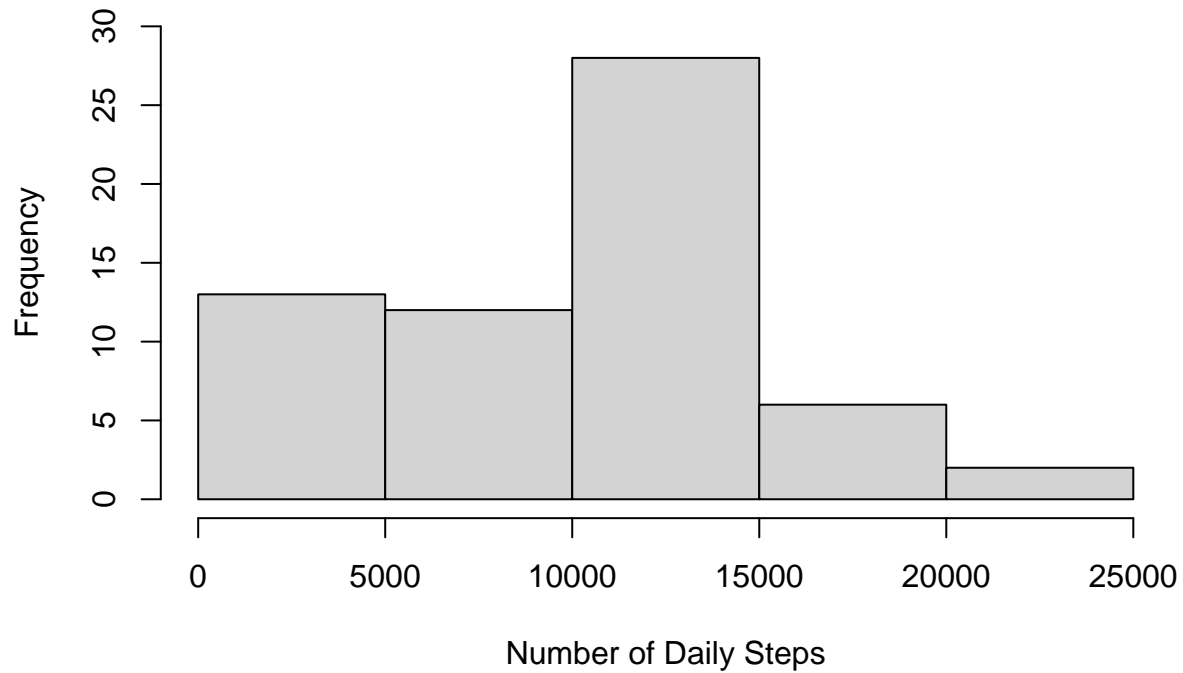
```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
activity_steps <-
  activity %>%
  group_by(date) %>%
  summarise(steps = sum(steps))
activity_steps$date <- as.Date(activity_steps$date)
hist(activity_steps$steps,
      main="Histogram of Daily Steps",
      xlab="Number of Daily Steps", ylim=c(0,30))
```

## Histogram of Daily Steps



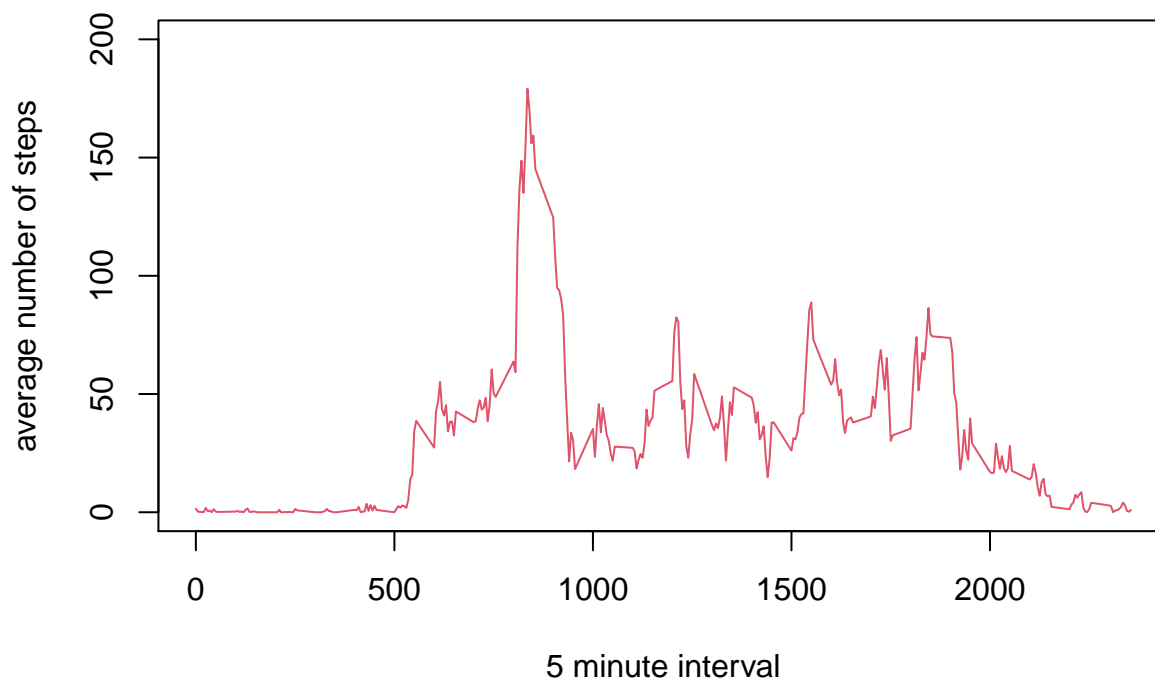
```
summary(activity_steps$steps)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##         0   6778   10395   9354   12811   21194
```

Calculates average number of steps for each interval, plots time series of average steps per interval, and calculates 5 minute interval with maximum average number of daily steps.

```
avg_steps_int <- group_by(activity, interval) %>% summarize(m = mean(steps))
names(avg_steps_int)[2] <- "steps"
plot(avg_steps_int$interval,
     avg_steps_int$steps, type = "l",
     main="Average Steps Per Interval", col=2,
     xlab="5 minute interval", ylab="average number of steps",
     ylim=c(0,200))
```

## Average Steps Per Interval



```
max_int <- which.max(avg_steps_int$interval) *5
max_int
```

```
## [1] 1440
```

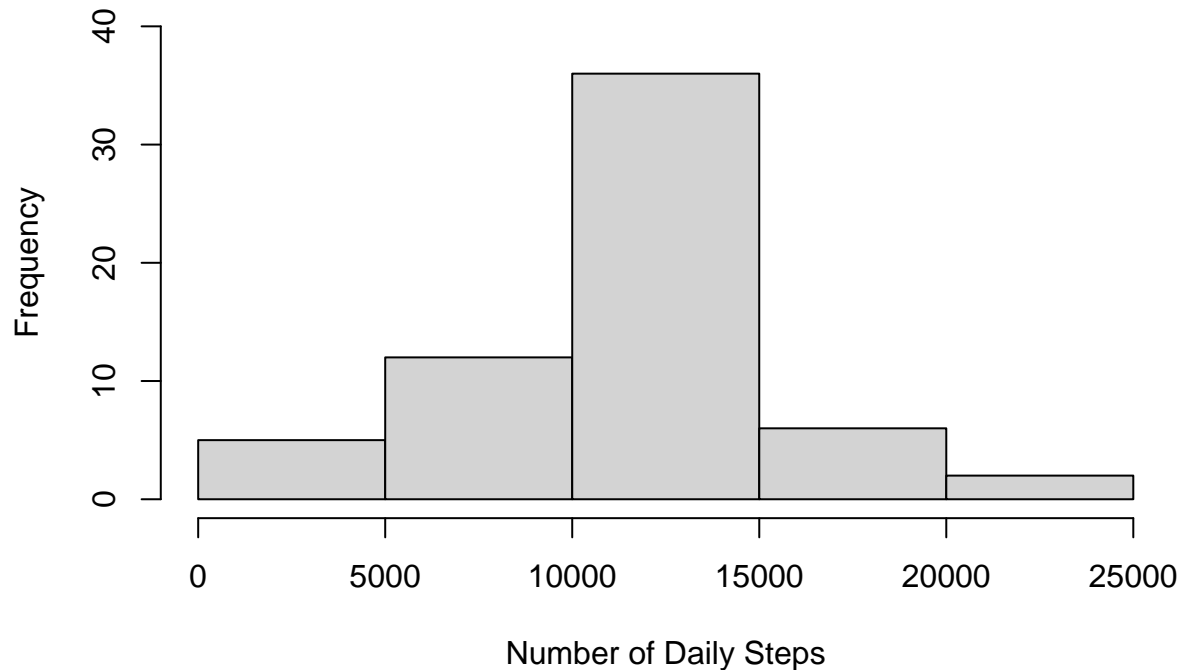
Replaces NA values of “steps” with mean of “steps” column, plots histogram of daily steps, and takes summary statistics to calculate mean and median of daily steps taken.

```
activity <- read.csv("activity.csv", header=TRUE)
sum(is.na(activity$steps))
```

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

```
library(dplyr)
activity_impute <- activity %>%
  group_by(interval) %>%
  mutate(steps=ifelse(is.na(steps), mean(steps,na.rm=TRUE), steps))
activity_impute_2 <- activity_impute %>%
  group_by(date) %>%
  summarise(steps=sum(steps))
hist(activity_impute_2$steps,
      main="Histogram of Daily Steps",
      xlab="Number of Daily Steps",
      ylim=c(0,40))
```

## Histogram of Daily Steps



```
summary(activity_impute_2$steps)
```

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

Categorizes dates as weekdays and weekend days. Averages steps according to these categories. Creates panel plots to compare average step patterns across 5 minute intervals for weekdays and weekend days.

```
activity <- read.csv("activity.csv", header=TRUE)
activity_impute <- activity %>%
  group_by(interval) %>%
  mutate(steps=ifelse(is.na(steps), mean(steps,na.rm=TRUE), steps))
```

```
activity_impute$interval <- as.numeric(activity_impute$interval)
activity_impute$steps <- as.numeric(activity_impute$steps)
activity_impute$date <- as.Date(activity_impute$date)
```

```
library(dplyr)
activity_impute$day <- weekdays(activity_impute$date)
weekend <- c("Saturday", "Sunday")
activity_impute$category[activity_impute$day==weekend] <- "weekend"
```

```
## Warning: Unknown or uninitialised column: `category`.
```

```
activity_impute$category[activity_impute$day!=weekend] <- "weekday"
```

```
avg_steps_day <- group_by(activity_impute, interval, category) %>%
  summarize(m = mean(steps))
```

```
## `summarise()` has grouped output by 'interval'. You can override using the `.groups` argument.
```

```
names(avg_steps_day)[3] <- "steps"
```

```
library(ggplot2)
```

```
p <- ggplot(data = avg_steps_day, aes(x = interval, y = steps)) + geom_line()
```

```
p + facet_wrap(~category)
```

