

# Loading and preprocessing the data

Read the data file 'activity.csv' convert date from factor format to date format 'ymd'

```
library(lubridate)
library(dplyr)
library(ggplot2)
```

```
actv <- read.csv('~/.desktop/activity.csv')
```

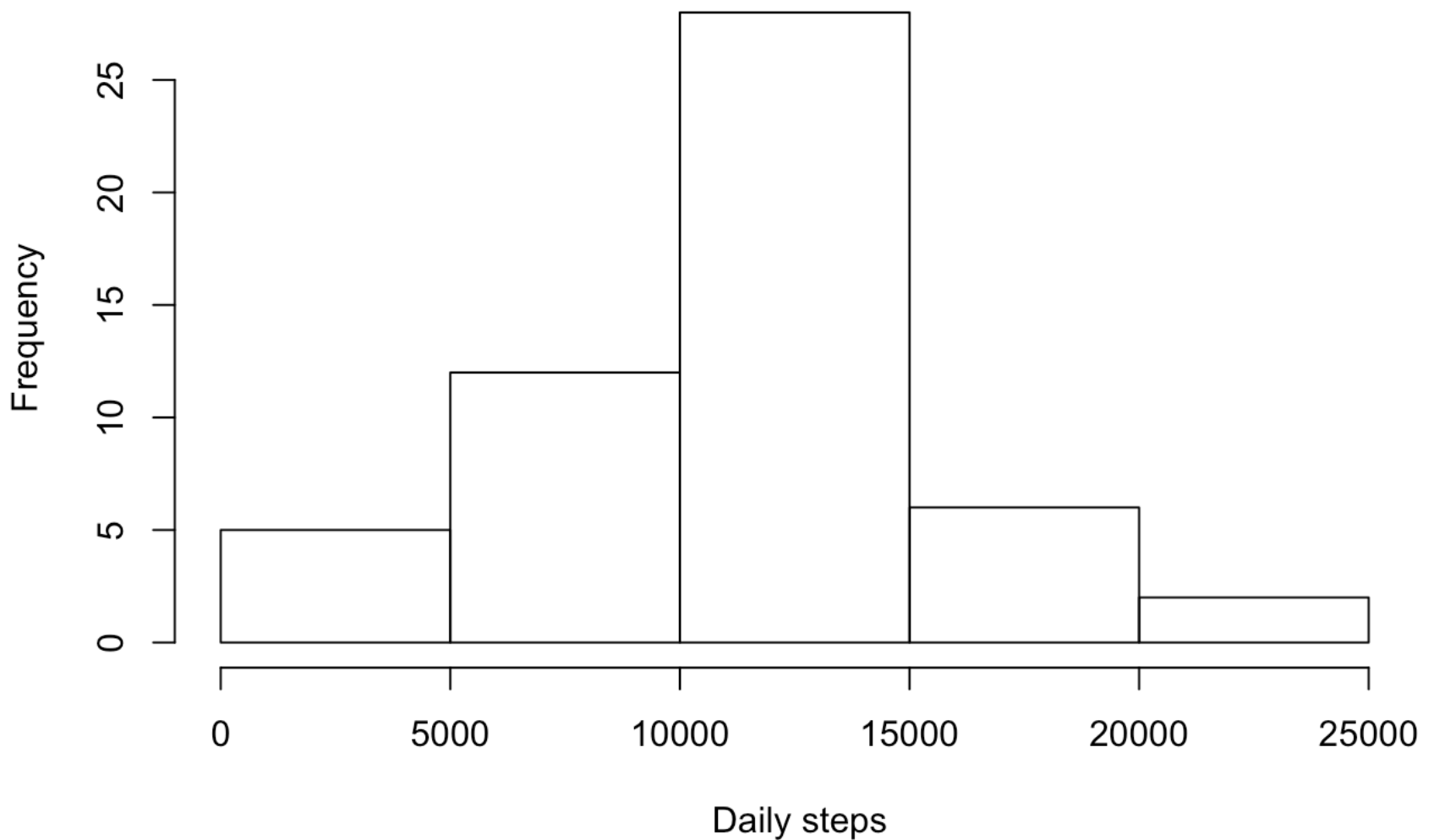
```
actv$date <- ymd(as.Date(as.character(actv$date)))
a<- actv
actv <- filter(actv, !is.na(steps) )
```

## What is mean total number of steps taken per day?

Calculate the total number of steps taken per day Make a histogram of the total number of steps taken each day Calculate and report the mean and median of the total number of steps taken per day

```
dailysteps <- actv %>% group_by(date) %>%
  summarise(totalsteps = sum(steps))
hist(dailysteps$totalsteps, freq = NULL, axes = T,
      main = 'steps taken every day', xlab = ' Daily steps')
```

## steps taken every day



```
mediandailysteps <- summary(dailysteps$totalsteps)[3]
meandailysteps <- summary(dailysteps$totalsteps)[4]
print(c('Mean Daily Steps =', meandailysteps))
```

```
##                               Mean
## "Mean Daily Steps ="      "10766.1886792453"
```

```
print(c('Median Daily Steps =', mediandailysteps))
```

```
##                               Median
## "Median Daily Steps ="      "10765"
```

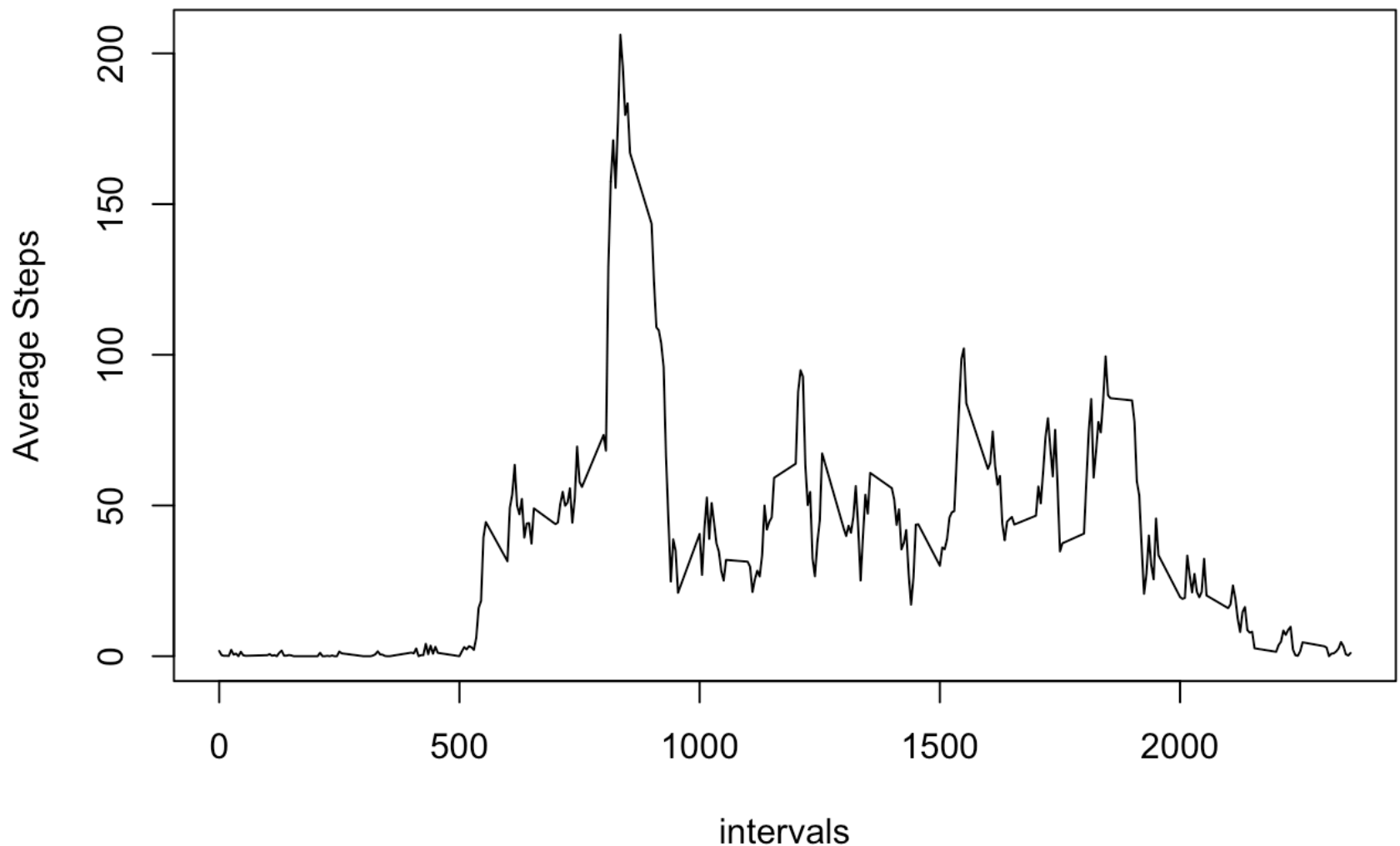
## What is the average daily activity pattern?

Make a time series plot (“type=”l”) of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis) Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

```
stp_int <- actv %>% group_by(interval) %>%
  summarise(avg_steps = mean(steps) )

plot(stp_int$interval, stp_int$avg_steps,
     type = 'l', xlab = 'intervals', ylab = 'Average Steps',
     main = 'Average steps over the day')
```

## Average steps over the day



```
mint <- stp_int %>%
  filter(avg_steps == max(avg_steps) )
print(c("The Interval in which Max steps are taken = ", mint[1]) )
```

```
## [[1]]
## [1] "The Interval in which Max steps are taken = "
##
## $interval
## [1] 835
```

## Imputing missing values

Note that there are a number of days/intervals where there are missing values (coded as NA). Calculate and report the total number of missing values (i.e. NAs)

```

actv <- a
missingvals <- sum(is.na(actv$steps))
print(c("Total Missing Values", missingvals))

```

```
## [1] "Total Missing Values" "2304"
```

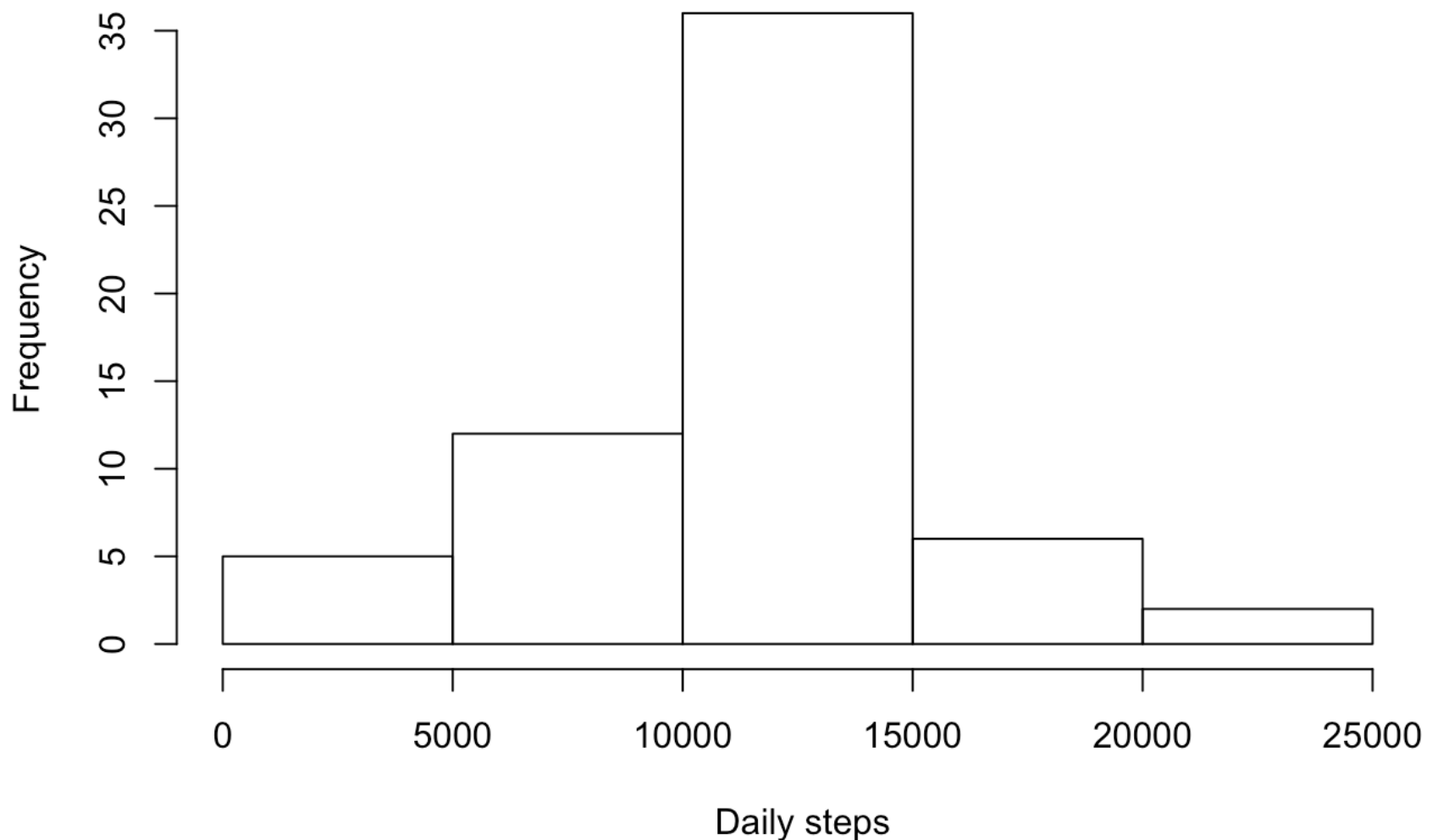
```

for (i in 1:dim(actv)[1]) {
  if (is.na(actv[i,1] ) )
  {
    const <- filter(stp_int, interval == actv[i,3])
    actv[i,1] <- const[1,2]
  }
}

dailysteps <- actv %>% group_by(date) %>% summarise(sum(steps))
names(dailysteps)[2] <- 'totalsteps'
hist(dailysteps$totalsteps, freq = NULL, axes = T,
     main = 'steps taken every day', xlab = ' Daily steps')

```

## steps taken every day



```

mediandailysteps <- summary(dailysteps$totalsteps)[3]
meandailysteps <- summary(dailysteps$totalsteps)[4]
print(c('Mean Daily Steps after replacement=', meandailysteps))

```

```
##  
## "Mean Daily Steps after replacement="  
##                               Mean  
##                               "10766.1886792453"
```

```
print(c('Median Daily Steps after replacement =', mediandailysteps))
```

```
##  
## "Median Daily Steps after replacement ="  
##                               Median  
##                               "10766.1886792453"
```

## Are there differences in activity patterns between weekdays and weekends?

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. 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).

```
actv <- mutate(actv, dday = case_when(  
  weekdays(date) %in% c("Monday", "Tuesday", "Wednesday", "Thursday",  
"Friday") ~ "weekday",  
  weekdays(date) %in% c("Saturday", "Sunday") ~ "weekend" ) )  
  
stp_week <- actv %>% group_by(dday, interval) %>%  
  summarise(week_steps = mean(steps) )  
  
par(mfrow = c(1,2) )  
qplot(interval, week_steps, data=stp_week, facets=~dday,  
  geom = c("line"))
```

