Reproducible-Research-Course-project-1

D.Dashinov

10/21/2020

Getting the data

First I'm going to write some code for downloading the data

```
fileURL <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
download.file(fileURL, destfile = "data.zip")

unzip("data.zip", files = NULL, list = FALSE, overwrite = TRUE,
      junkpaths = FALSE, exdir = ".", unzip = "internal",
      setTimes = FALSE)

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

Question 1

What is mean total number of steps taken per day?

```
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
```

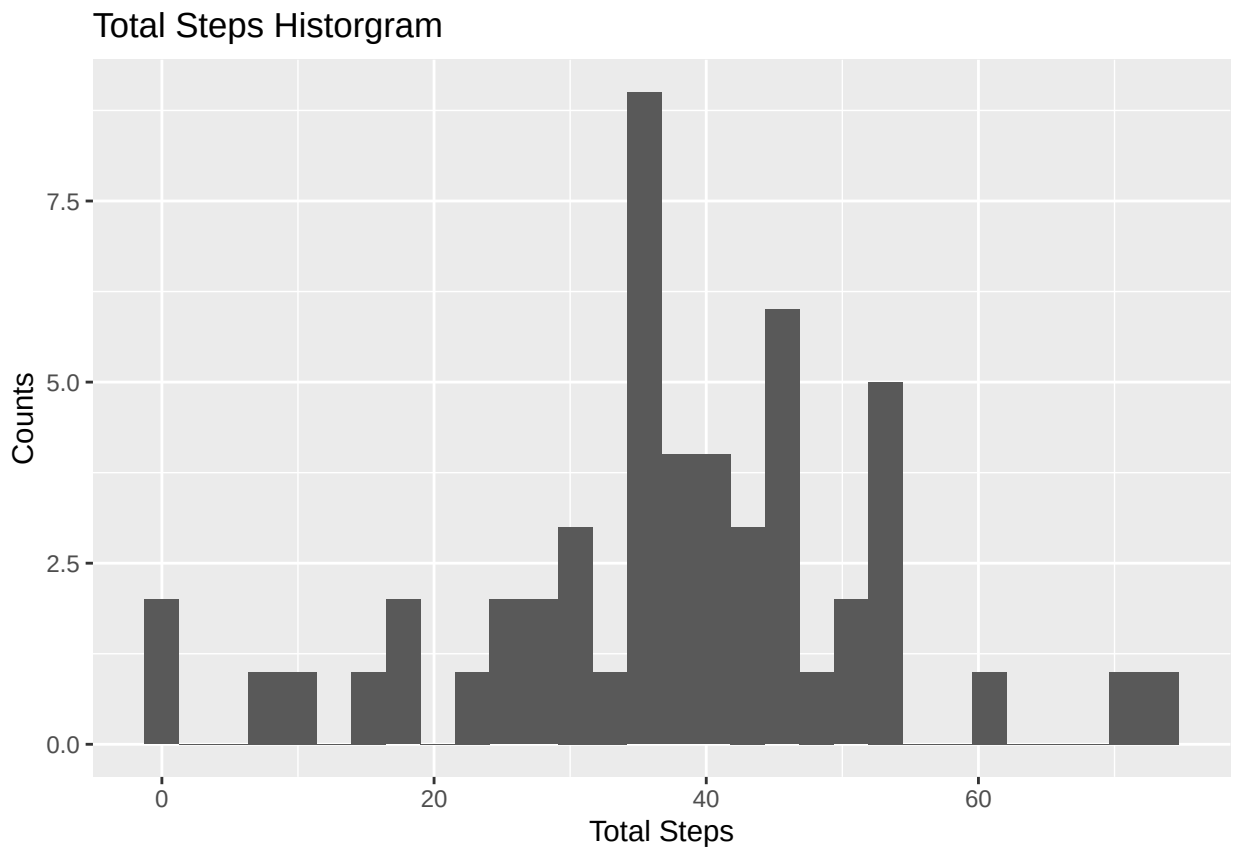
```
library(ggplot2)

Q1 <- DataActivity %>% group_by(date) %>% summarise(Mean = mean(steps, na.rm = T))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
qplot(Q1$Mean, geom="histogram",xlab="Total Steps",ylab="Counts",main="Total Steps Historgram")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 8 rows containing non-finite values (stat_bin).
```




```
summary(Q1)
```

```
##       date                Mean
##  Length:61          Min.   : 0.1424
##  Class :character   1st Qu.:30.6979
##  Mode  :character   Median :37.3785
##                     Mean   :37.3826
##                     3rd Qu.:46.1597
##                     Max.   :73.5903
##                     NA's   :8
```

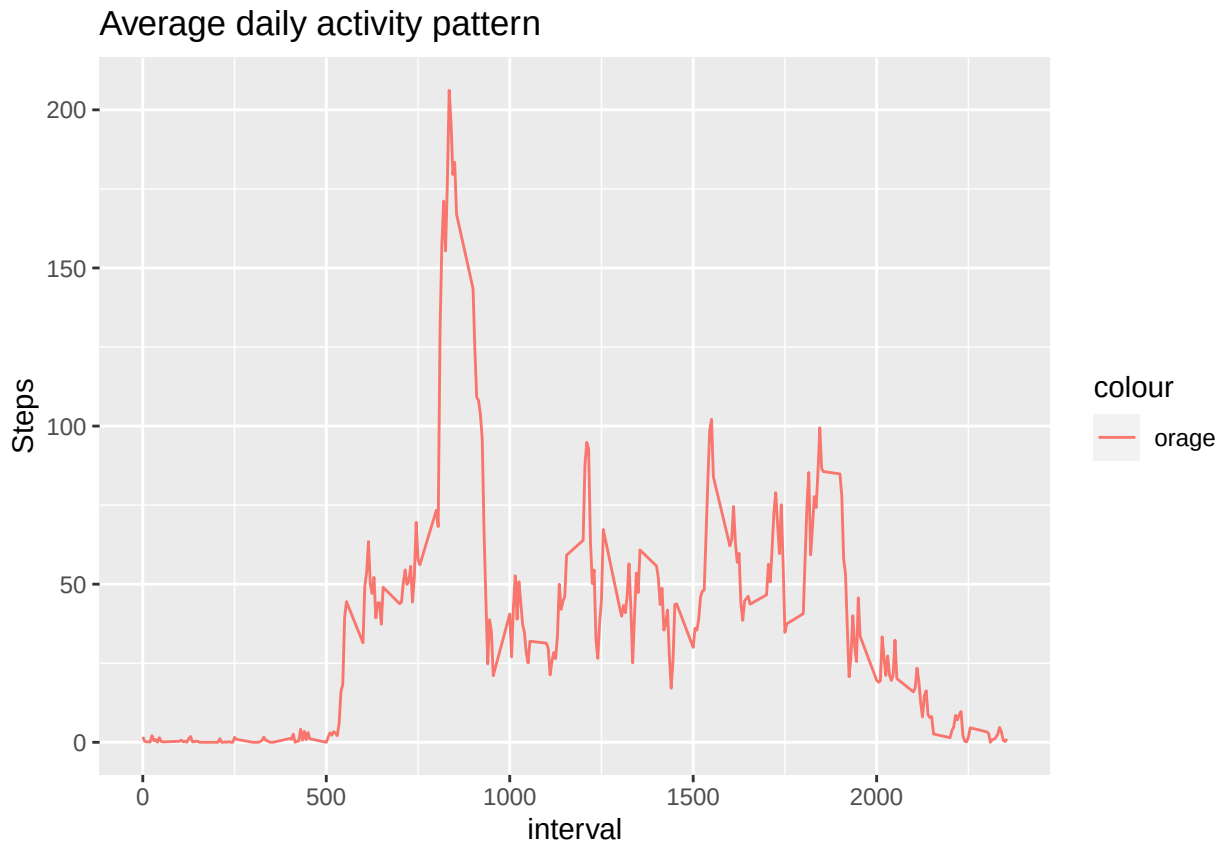
Question 2

What is the average daily activity pattern?

```
library(ggplot2)

df <- DataActivity %>% group_by(interval) %>% summarise(Mean = mean(steps, na.rm = T))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
ggplot(df, aes(x = interval, y = Mean, group=1)) +
        geom_path(aes(color = "orage")) +
        ggtitle("Average daily activity pattern") +
        ylab("Steps")
```



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

```
df[which.max(df$Mean), ]$interval
```

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

Question 3

Imputing missing values

```
sum(is.na(DataActivity$steps))
```

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

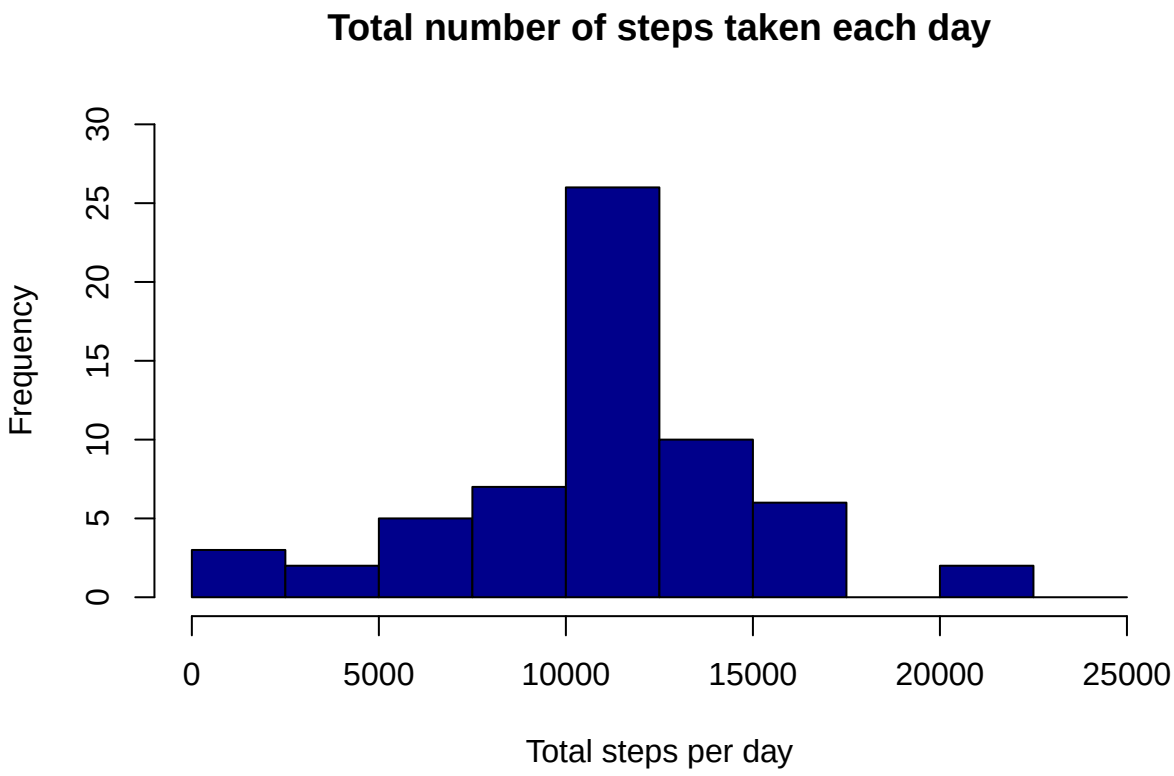
```
imputed_steps <- df$Mean[match(DataActivity$interval, df$interval)]
```

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

```
activity_imputed <- transform(DataActivity, steps = ifelse(is.na(DataActivity$steps),
                                        yes = imputed_steps, no = DataActivity$steps))
total_steps_imputed <- aggregate(steps ~ date, activity_imputed, sum)
names(total_steps_imputed) <- c("date", "daily_steps")
```

Histogram of the total number of steps taken each day with and report the mean and median total number of steps taken per day.

```
hist(total_steps_imputed$daily_steps, col = "darkblue", xlab = "Total steps per day", ylim = c(0,30), ma
```



Here is the mean

```
mean(total_steps_imputed$daily_steps)
```

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

Here is the median

```
median(total_steps_imputed$daily_steps)
```

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

Creating a new factor variable in the dataset with two levels – "weekday" and "weekend" indicating whether a given date is a weekday or weekend day.

```
DataActivity$date <- as.Date(strptime(DataActivity$date, format="%Y-%m-%d"))

DataActivity$datetype <- sapply(DataActivity$date, function(x) {
        if (weekdays(x) == "Saturday" | weekdays(x) =="Sunday")
        {y <- "Weekend"} else
        {y <- "Weekday"}
        y
})
```

Plotting by weekdays and weekends

```
activity_by_date <- aggregate(steps~interval + datetype, DataActivity, mean, na.rm = TRUE)

activity_by_date$datetype <- as.factor(activity_by_date$datetype)
activity_by_date$interval <- as.numeric(activity_by_date$interval)
```

```
plot<- ggplot(activity_by_date, aes(x = interval, y = steps, color = datetype)) +
        geom_line() +
        labs(title = "Average daily steps by type of date", x = "Interval", y = "Average number of steps
        facet_wrap(~datetype, ncol = 1, nrow=2)
print(plot)
```