

PA1_template

TM

2023-02-12

Loading and preprocessing the data

Question 1: Code for reading in the dataset and/or processing the data

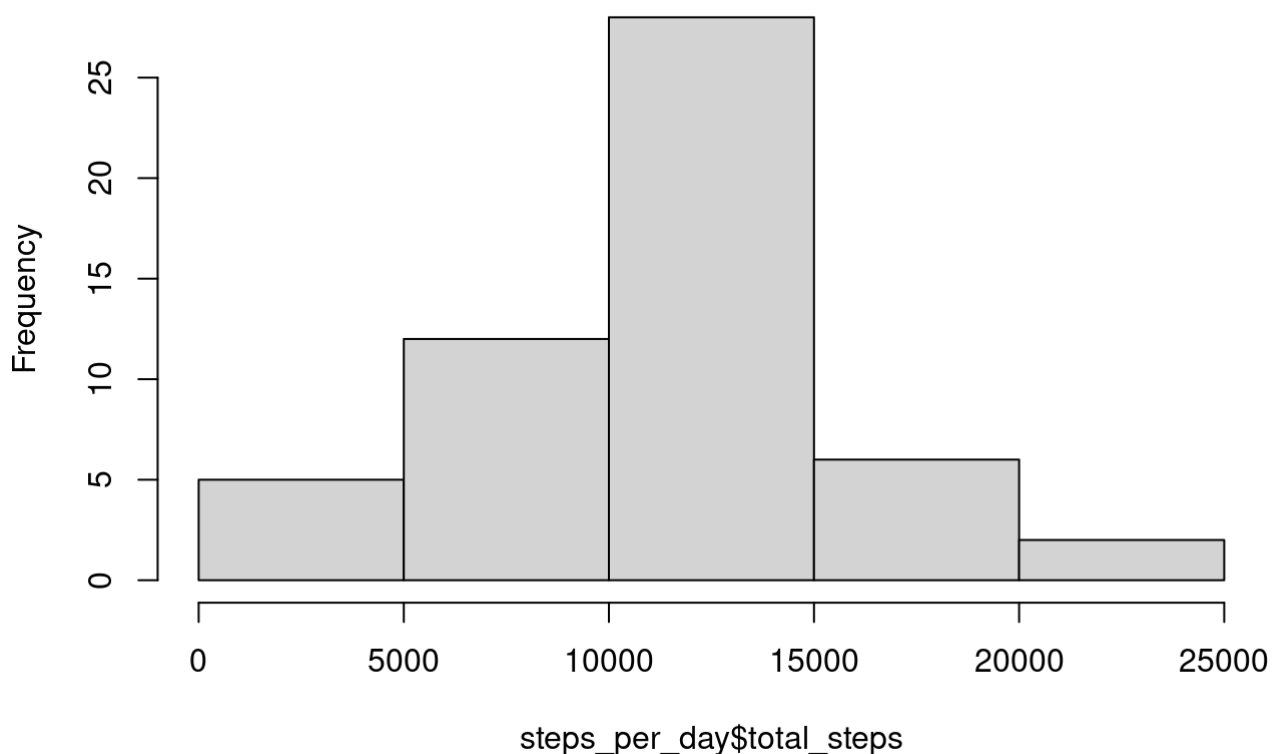
```
suppressMessages(library(dplyr))  
suppressMessages(library(lubridate))  
suppressMessages(library(ggplot2))  
df <- read.csv("activity.csv")
```

What is mean total number of steps taken per day?

Question 2: Histogram of the total number of steps taken each day Question 3: Mean and median number of steps taken each day

```
steps_per_day <- df %>%  
  group_by(date) %>%  
  summarize(total_steps=sum(steps))  
hist(steps_per_day$total_steps)
```

Histogram of steps_per_day\$total_steps



```
mean_steps_per_day <- mean(steps_per_day$total_steps, na.rm=TRUE)
mean_steps_per_day
```

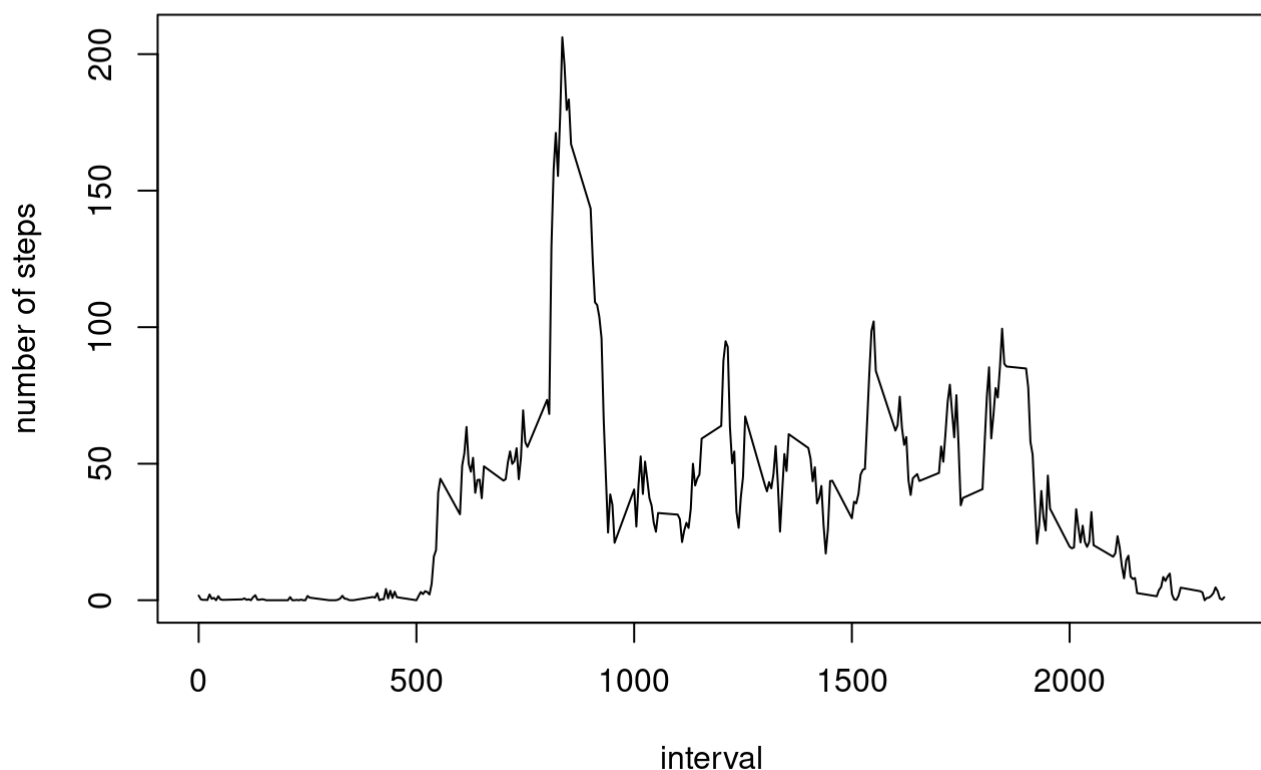
```
## [1] 10766.19
```

What is the average daily activity pattern?

Question 4: Time series plot of the average number of steps taken.

```
interval_5min <- df %>%
  group_by(interval) %>%
  summarize(int_steps=mean(steps, na.rm=TRUE))

plot(interval_5min$interval, interval_5min$int_steps, type="l", ylab="number of steps", xlab="interval")
```



Question 5: The 5-minute interval that, on average, contains the maximum number of steps

```
max_step <- max(interval_5min$int_steps)
max_5min <- interval_5min %>%
  filter(int_steps == max_step)
colnames(max_5min) <- c("The interval that contains the maximum number", "Maximum number of steps")
max_5min
```

```
## # A tibble: 1 × 2
##   `The interval that contains the maximum number` `Maximum number of steps`
##                                     <int>                                <dbl>
## 1                                     835                                206.
```

Imputing missing values

Question 6-1: Code to describe and show a strategy for imputing missing data. The number of row which contains missing values.

```
include_NA <- df %>%
  filter(is.na(steps))
number_NA <- nrow(include_NA)
paste("The number of missing values in steps is ", number_NA, ".")
```

```
## [1] "The number of missing values in steps is 2304 ."
```

Question 6-2: Devise a strategy for filling in all of the missing values in the dataset – I use the mean steps to fill the missin value. Question 6-3: Create a new dataset – The name of new dataset is “new_dataset”.

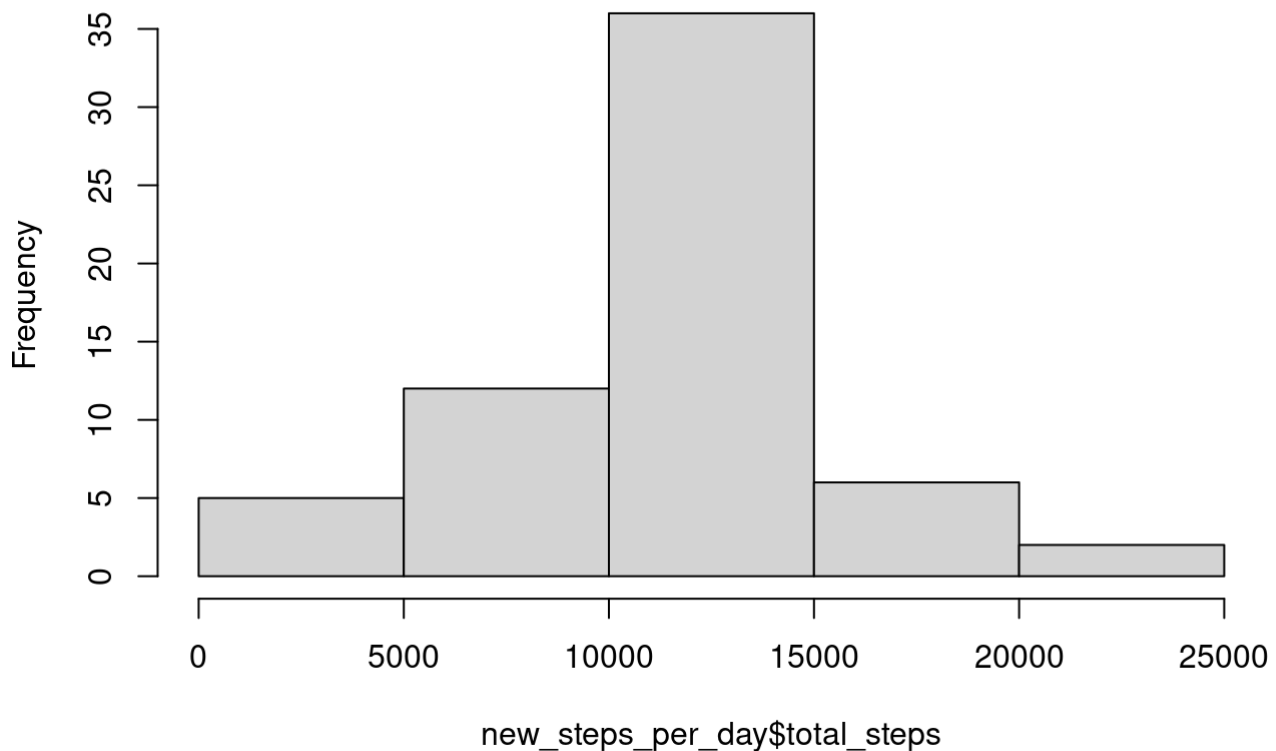
```
# I use mean values of day to fill the missing value
mean_steps <- df %>%
  group_by(interval) %>%
  summarize(mean_steps=mean(steps, na.rm=TRUE))

df_add_mean <- left_join(df, mean_steps, by="interval")
new_dataset <- df_add_mean %>%
  mutate(new_steps=ifelse(is.na(steps),mean_steps,steps)) %>%
  select(new_steps, date, interval)
```

Question 7: Make a histogram of the total number of steps in new dataset. Calculate the mean and median.

```
new_steps_per_day <- new_dataset%>%
  group_by(date) %>%
  summarize(total_steps=sum(new_steps))
hist(new_steps_per_day$total_steps)
```

Histogram of new_steps_per_day\$total_steps



```
new_mean_steps_per_day <- mean(new_steps_per_day$total_steps, na.rm=TRUE)
paste("The mean total number of steps taken per day is ",round(new_mean_steps_per_day), ".")
```

```
## [1] "The mean total number of steps taken per day is 10766 ."
```

```
new_median_steps_per_day <- median(new_steps_per_day$total_steps, na.rm=TRUE)
paste("The mean total number of steps taken per day is ",round(new_median_steps_per_day), ".")
```

```
## [1] "The mean total number of steps taken per day is 10766 ."
```

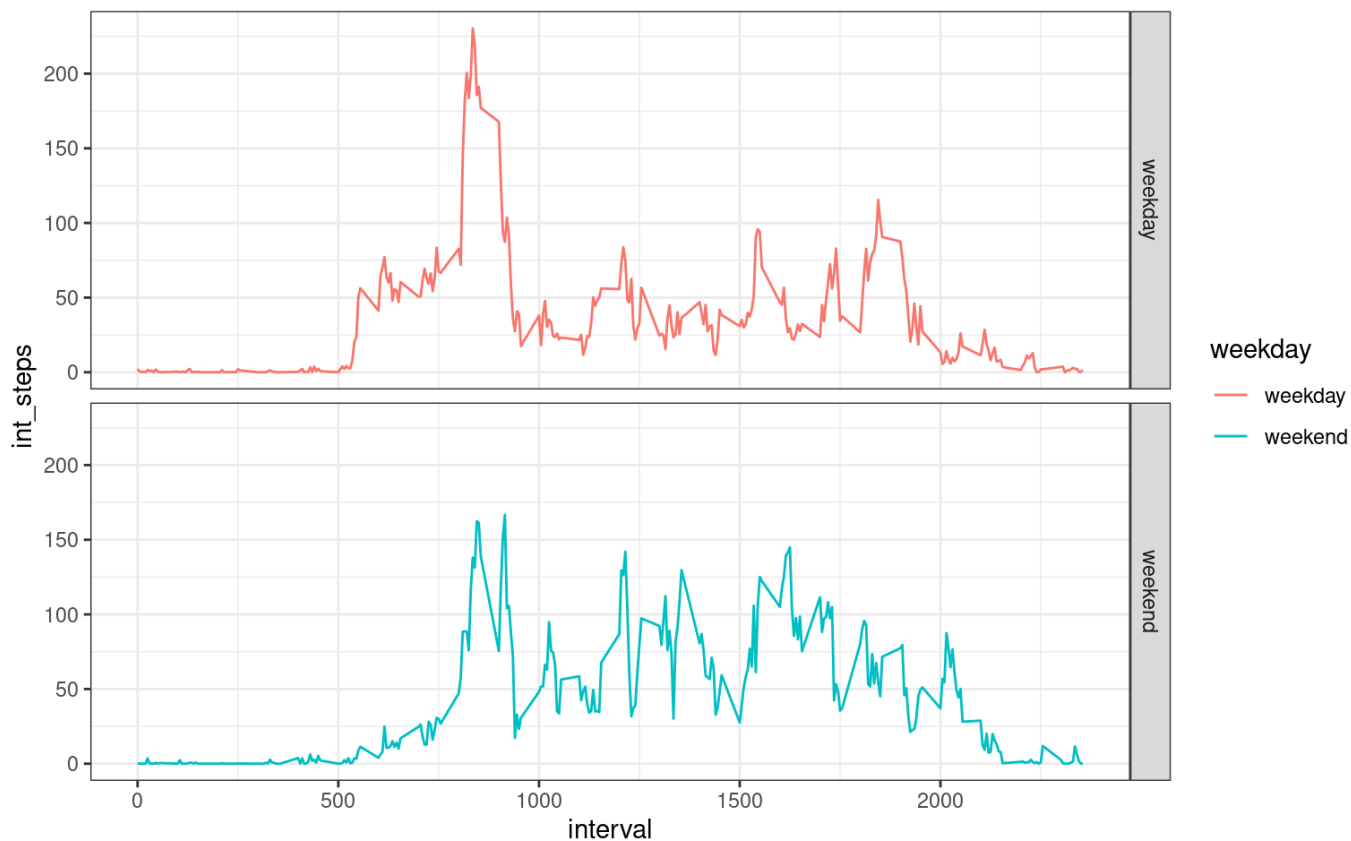
Are there differences in activity patterns between weekdays and weekends?

Question 8-1: Create a new factor variable in the dataset with two levels – “weekday” and “weekend”

```
new_dataset_weekday<- new_dataset %>%
  mutate(weekday = wday(date)) %>%
  mutate(weekday = ifelse(weekday %in% c(2,3,4,5,6), "weekday", "weekend"))
```

Question 8-2: Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

```
options(dplyr.summarise.inform = FALSE)
interval_5min_weekday <- new_dataset_weekday %>%
  group_by(weekday, interval) %>%
  summarize(int_steps=mean(new_steps, na.rm=TRUE))
interval_5min_weekday %>%
  ggplot(aes(x=interval, y=int_steps, color=weekday)) +
  geom_line() +
  facet_grid(weekday ~ .) +
  theme_bw()
```



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.