# PA1

Wei Jiang

2023-12-26

```
knitr::opts_chunk$set(echo = TRUE, warning = FALSE)
```

### 1. Load the data.

```
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)
# load data and transfer date from char to Date
raw <- read.csv("/Users/weijiang/Downloads/activity.csv")
df <- raw
```

### 2.Process/transform the data (if necessary) into a format suitable for your analysis.

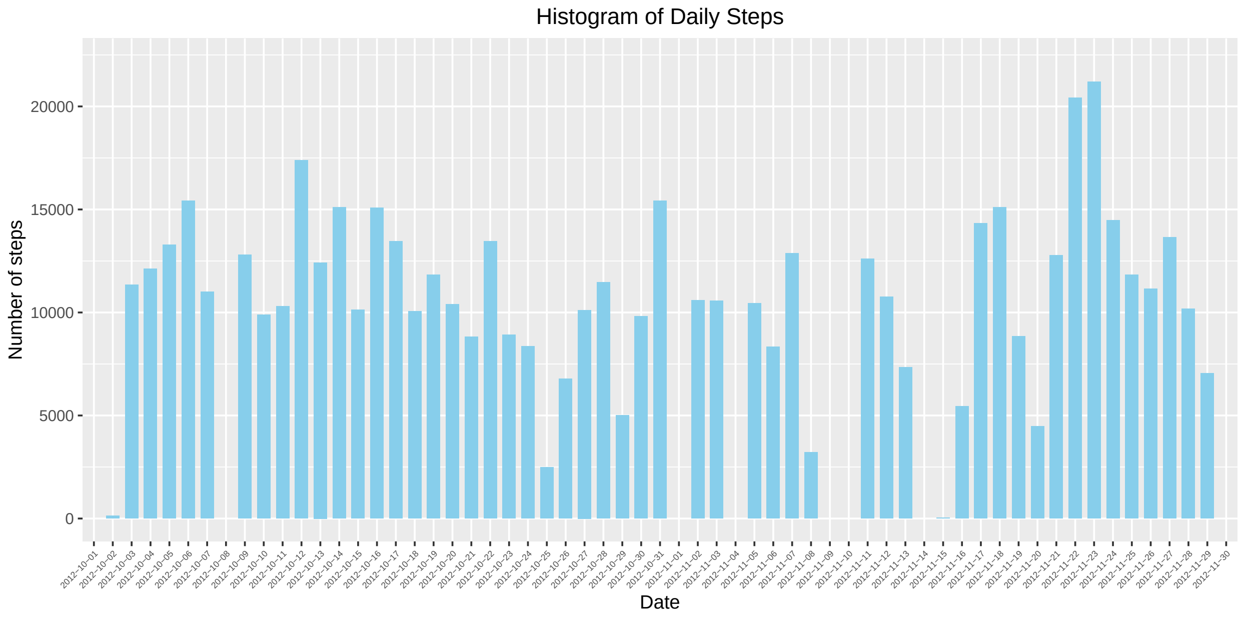
```
df$date <- as.Date(df$date, format= "%Y-%m-%d")
df <- df %>% mutate(df, day = weekdays(df$date))
```

### 3.Make a histogram of the total number of steps taken each day.

```
# Calculate and report the mean and median total number of steps taken per day
data <- df %>% group_by(date) %>% summarise(total=sum(steps))
Mean <- as.integer(mean(data$total, na.rm = TRUE))
Median  <- median(data$total, na.rm = TRUE)
```

### The mean of daily steps is 10766, median of daily steps is 10765.

```r
# Plotting the histogram
ggplot(data, aes(x = as.character(date), y = total)) +
  geom_bar(stat = "identity", fill = "skyblue", width = 0.7) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 5)) +
  scale_y_continuous(limits = c(0, max(data$total, na.rm = TRUE) + 1000)) +
  labs(x = "Date", y = "Number of steps", title = "Histogram of Daily Steps") +
  theme(plot.title = element_text(hjust = 0.5))
```



**Imputing missing values.**

**1.Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)..**

```r
n_miss <- sum(is.na(df)) # num of missing values
```

There are total 2304 missing values in the dataset..

**2.Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to be sophisticated. For example, you could use the mean/median for that day, or the mean for that 5-minute interval, etc..**

```r
# replacing missing values by the median steps in that time frame of the day

# remove rows with "NA" in it
df_na_free <- na.omit(df)

# median steps of each time frame per day in whole null missing data
d1 <- df_na_free %>% mutate(day=weekdays(date))

d_med <- d1 %>% group_by(day, interval) %>% summarise(med_steps = median(steps))
```

```
## ‘summarise()‘ has grouped output by ’day’. You can override using the ‘.groups‘
## argument.
```

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

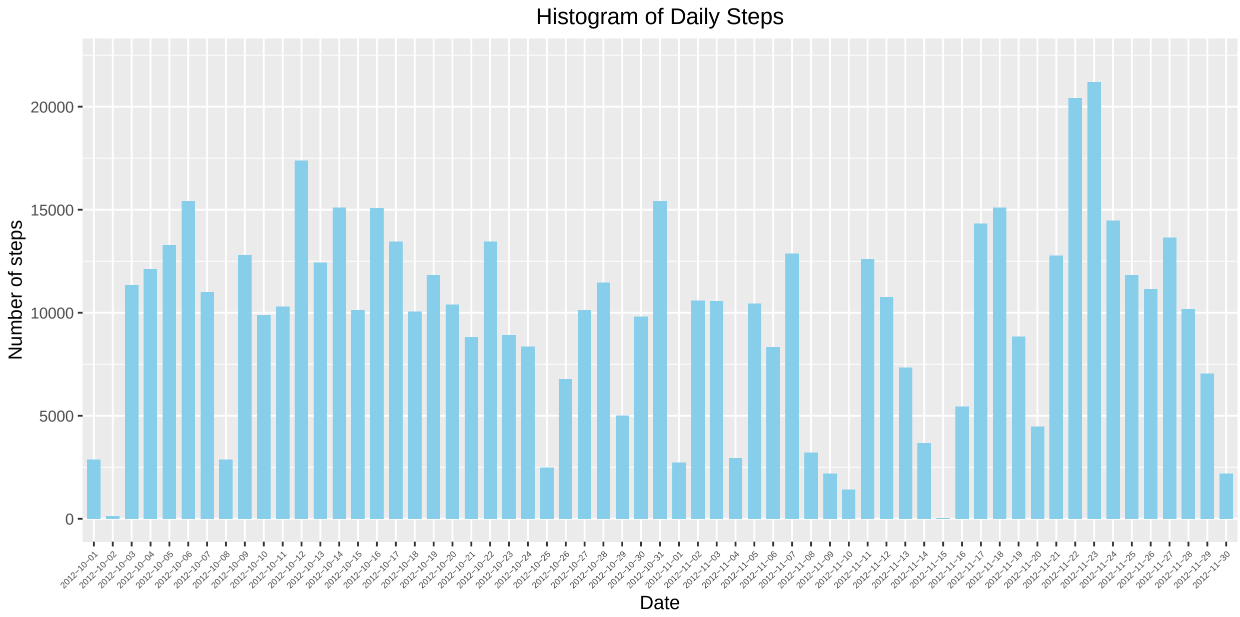
```
# use the median steps of each time frame per day to replace the missing time frame date
# for example interval 0 in "2012-12-01" is "NA", "2012-12-01" is Monday
# replacing it with median steps at interval 0  on Monday

df_na_free <- df %>%
    left_join (d_med, by = c("interval","day")) %>% # left join the column with median steps to origina
    mutate(
        steps = ifelse(is.na(steps), d_med$med_steps, steps) # replacing "NA" by median steps
    ) %>%
    select(-med_steps)

head(df_na_free)
```

```
##   steps       date interval    day
## 1     0 2012-10-01        0 Monday
## 2     0 2012-10-01        5 Monday
## 3     0 2012-10-01       10 Monday
## 4     0 2012-10-01       15 Monday
## 5     0 2012-10-01       20 Monday
## 6     0 2012-10-01       25 Monday
```

### 4.Make a histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day. Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?..

```
# Calculate and report the mean and median total number of steps taken per day
data <- df_na_free %>% group_by(date) %>% summarise(total=sum(steps))
Mean <- as.integer(mean(data$total, na.rm = TRUE))
Median  <- median(data$total, na.rm = TRUE)
```

```
# Plotting the histogram
ggplot(data, aes(x = as.character(date), y = total)) +
  geom_bar(stat = "identity", fill = "skyblue", width = 0.7) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 5)) +
  scale_y_continuous(limits = c(0, max(data$total, na.rm = TRUE) + 1000)) +
  labs(x = "Date", y = "Number of steps", title = "Histogram of Daily Steps") +
  theme(plot.title = element_text(hjust = 0.5))
```



**Are there differences in activity patterns between weekdays and weekends?..**

** 1.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.**

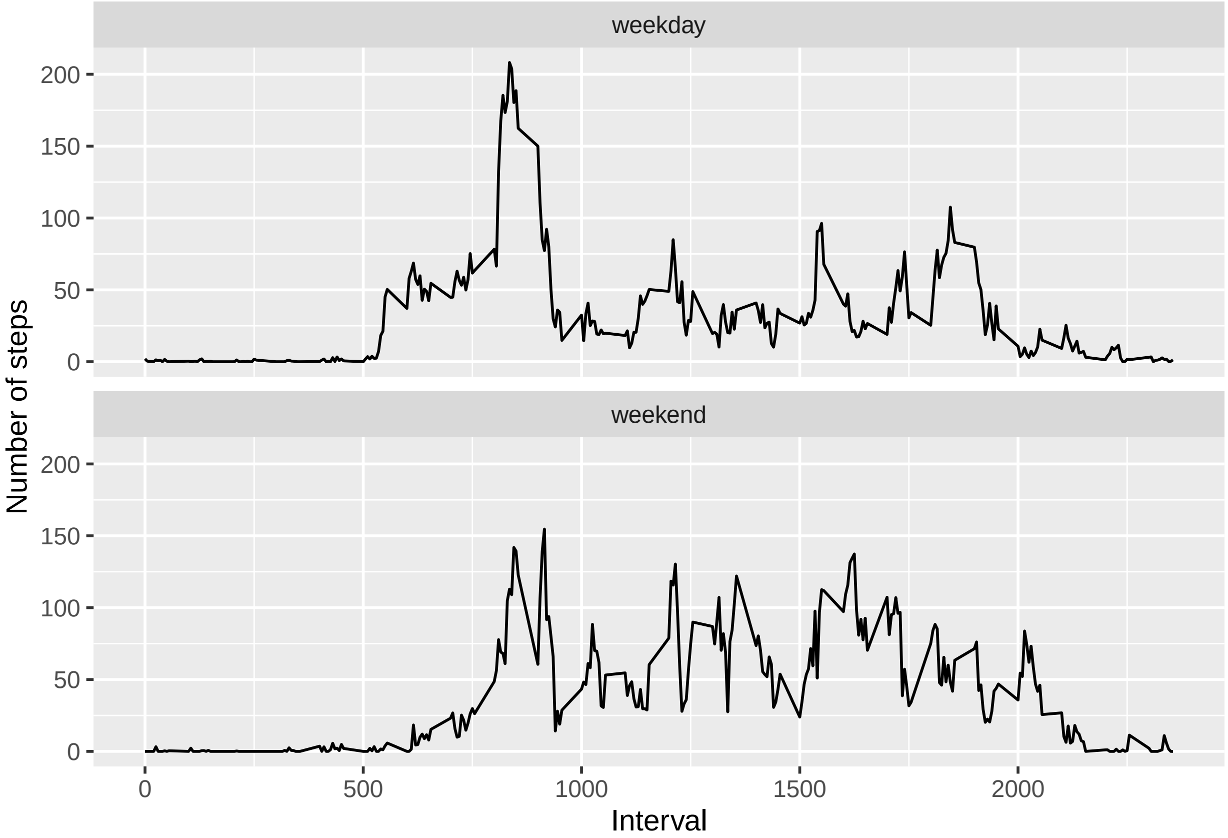
```
df_na_free <- df_na_free %>% mutate(day_type = ifelse(df_na_free$day %in% c("Saturday", "Sunday"), "week
head(df_na_free)
```

```
##   steps       date interval    day day_type
## 1     0 2012-10-01        0 Monday  weekday
## 2     0 2012-10-01        5 Monday  weekday
## 3     0 2012-10-01       10 Monday  weekday
## 4     0 2012-10-01       15 Monday  weekday
## 5     0 2012-10-01       20 Monday  weekday
## 6     0 2012-10-01       25 Monday  weekday
```

```
group_day_type <- df_na_free %>% group_by(interval, day_type) %>% summarise(ave=mean(steps))
```

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

```
ggplot(group_day_type,aes(x=interval,y=ave)) +
    geom_line() +
    facet_wrap(~ day_type, nrow=3, ncol=1) +
    labs(x="Interval", y="Number of steps")
```


weekday
weekend
Number of steps
200
150
100
50
0
200
150
100
50
0
0
500
1000
1500
2000
Interval