# Activity monitoring data analysis

Antonio Caputo

March 9th, 2023

## Introduction

It is now possible to collect a large amount of data about personal movement using activity monitoring devices such as a Fitbit, Nike Fuelband, or Jawbone Up. These type of devices are part of the "quantified self" movement – a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. But these data remain under-utilized both because the raw data are hard to obtain and there is a lack of statistical methods and software for processing and interpreting the data.

This assignment makes use of data from a personal activity monitoring device. This device collects data at 5 minute intervals through out the day. The data consists of two months of data from an anonymous individual collected during the months of October and November, 2012 and include the number of steps taken in 5 minute intervals each day.

## Load and trasform the data

Load essential library for the analysis:

We read the data and have a look at it:

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

```
##      steps            date              interval
##  Min.   :  0.00   Length:17568       Min.   :   0.0
##  1st Qu.:  0.00   Class :character   1st Qu.: 588.8
##  Median :  0.00   Mode  :character   Median :1177.5
##  Mean   : 37.38                      Mean   :1177.5
##  3rd Qu.: 12.00                      3rd Qu.:1766.2
##  Max.   :806.00                      Max.   :2355.0
##  NA's   :2304
```

The variables included in this dataset are:

- steps: Number of steps taking in a 5-minute interval (missing values are coded as NA)
- date: The date on which the measurement was taken in YYYY-MM-DD format
- interval: Identifier for the 5-minute interval in which measurement was taken

We have to transform date variable in the right format as date:

```
activity$date<- as.Date(activity$date)
```

Note that there are a number of days/intervals where there are missing values (coded as NA). The presence of missing days may introduce bias into some calculations or summaries of the data. Calculate and report the total number of missing values in the dataset.

```
# numbers of missing value
sum(is.na(activity$steps))
```

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

```
# percentage of missing value
mean(is.na(activity$steps))
```

```
## [1] 0.1311475
```

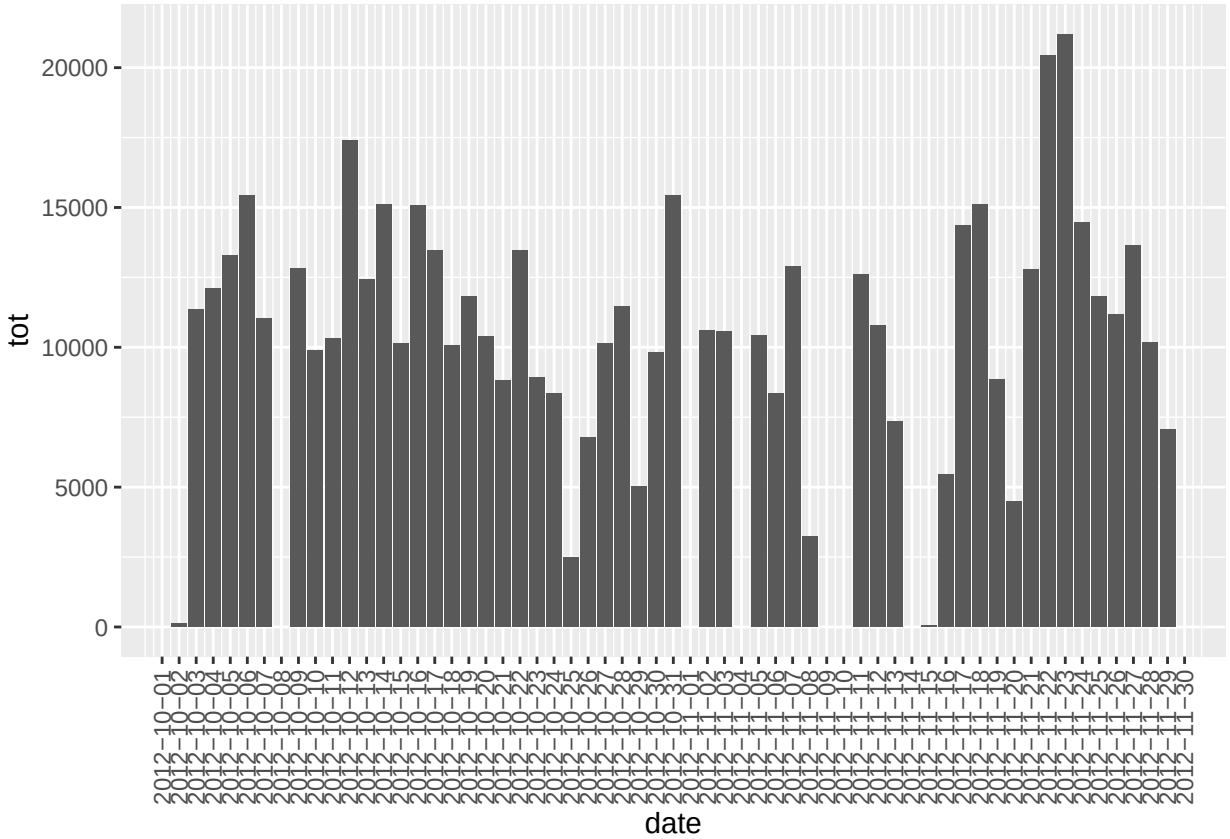
## Analysis and plot

Calculate the total number of steps taken per day. We plot the total number of steps taken each day.

```
steps_tot <-activity %>% group_by(date) %>% summarise(tot= sum(steps), mean=mean(steps), median=median(s

ggplot(data = steps_tot, aes(x=date, y=tot))+geom_col()+theme(axis.text.x = element_text(angle = 90, vj
```

```
## Warning: Removed 8 rows containing missing values (`position_stack()`).
```



Mean and median number of steps taken each day

```
mean(steps_tot$tot, na.rm = TRUE)
```

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

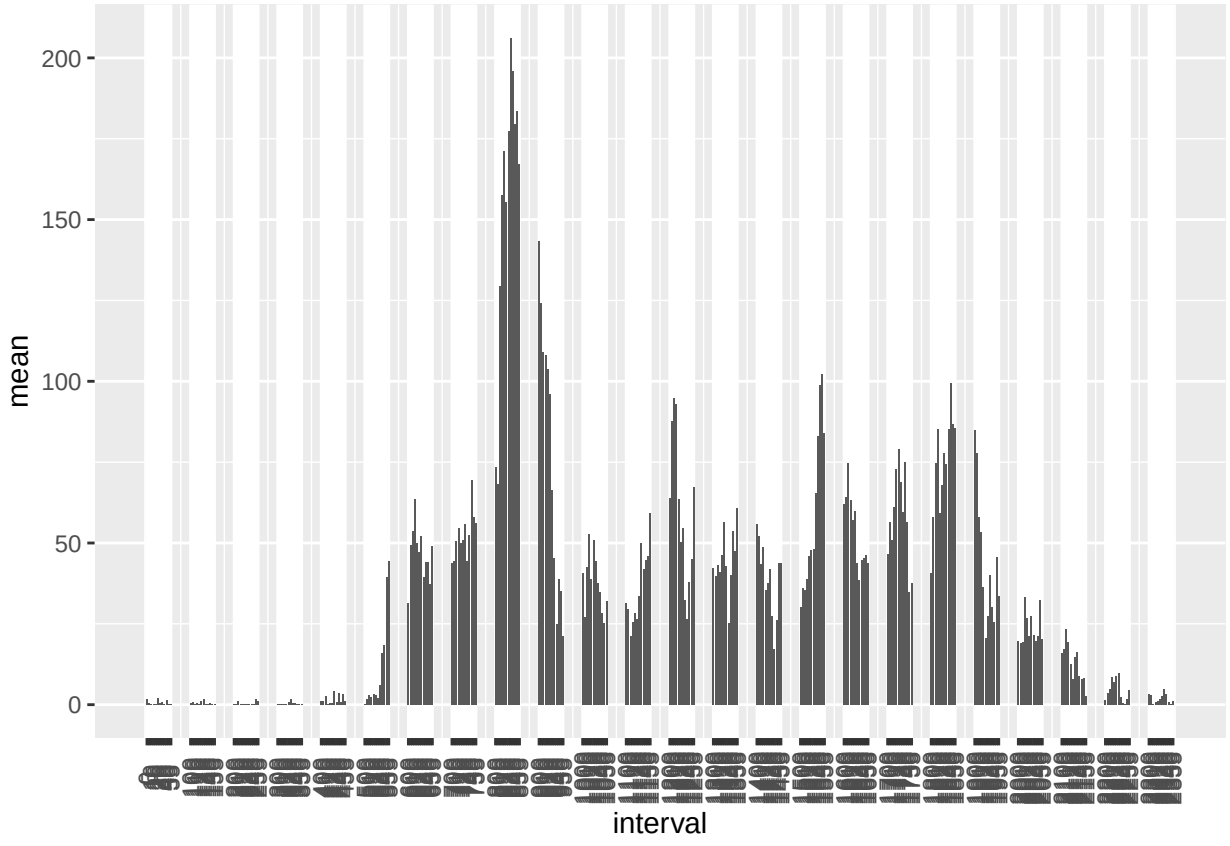
```
median(steps_tot$tot, na.rm = TRUE)
```

```
## [1] 10765
```

Time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)

```
interval_mean <-activity %>% group_by(interval) %>% summarise(tot= sum(steps, na.rm = TRUE), mean=mean(s

ggplot(data = interval_mean, aes(x=interval, y=mean))+geom_col()+theme(axis.text.x = element_text(angle
```




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

```
interval_mean$interval[which.max(interval_mean$mean)]
```

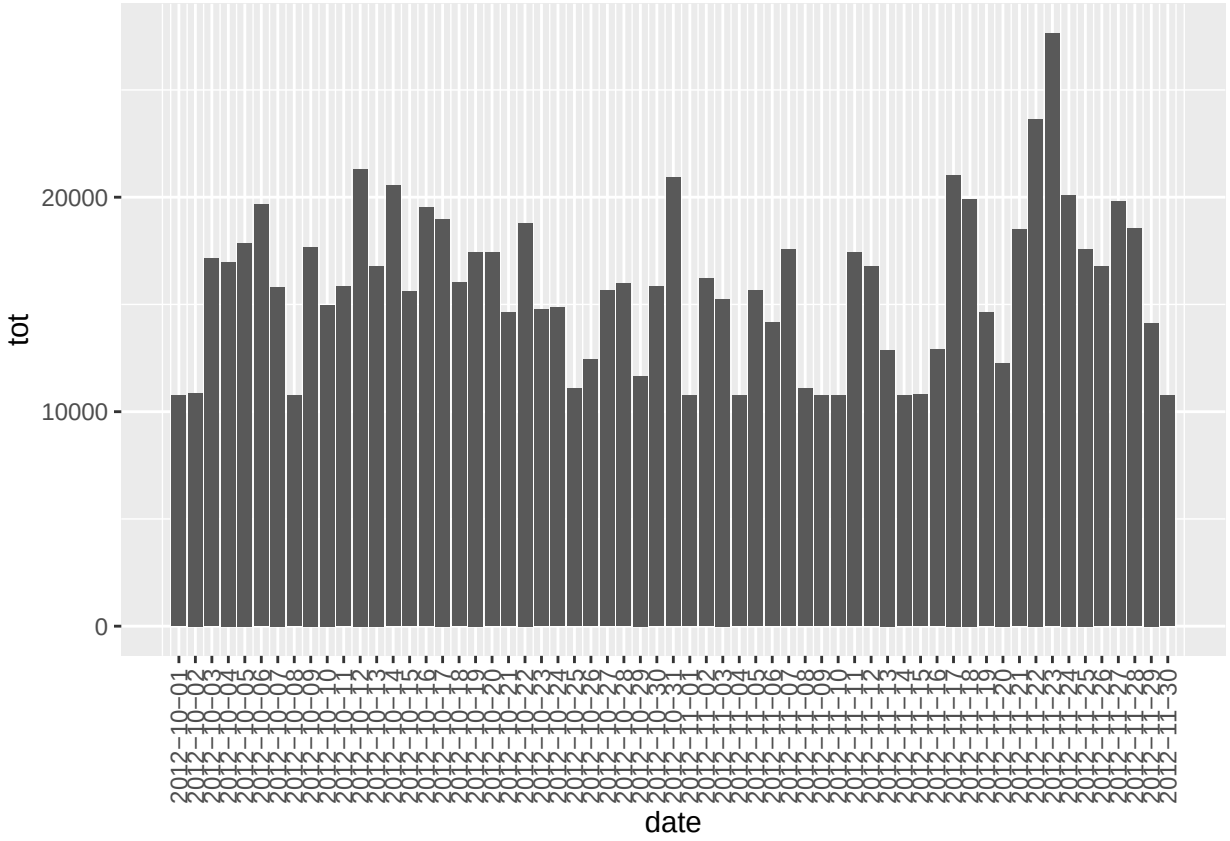
```
## [1] 835
```

Now we do the same plot but after handled the missing value. In this situation we decide to use the mean for that that 5-minute interval for filling the missing value.

```
activity_clean <- activity %>% group_by(interval) %>% mutate(steps= ifelse(is.na(steps) | steps== 0, mea

steps_clean <-activity_clean %>% group_by(date) %>% summarise(tot= sum(steps), mean=mean(steps), median=


ggplot(data = steps_clean, aes(x=date, y=tot))+geom_col()+theme(axis.text.x = element_text(angle = 90, v
```




Mean and median number of steps taken each day

```
mean(steps_clean$tot, na.rm = TRUE)
```

```
## [1] 15875.99
```

```
median(steps_clean$tot, na.rm = TRUE)
```

```
## [1] 15837.74
```

In the last analysis we compare the time series in two different situation: during weekday and during weekend day:

```
weekdays1 <- c('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday')
#Use `%in%` and `weekdays` to create a logical vector
#convert to `factor` and specify the `levels/labels`
activity_clean$wDay <- factor((weekdays(activity_clean$date) %in% weekdays1),
         levels=c(FALSE, TRUE), labels=c('weekend', 'weekday'))

interval_mean_clean <-activity_clean %>% group_by(wDay, interval) %>% summarise(tot= sum(steps, na.rm =
```

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

```
ggplot(data = interval_mean_clean, aes(x=interval, y=mean))+geom_line()+theme(axis.text.x = element_tex
```