PA1\_template

TM

2023-01-27

## Loading and preprocessing the data

library(dplyr)

##   
## 次のパッケージを付け加えます: 'dplyr'

## 以下のオブジェクトは 'package:stats' からマスクされています:  
##   
## filter, lag

## 以下のオブジェクトは 'package:base' からマスクされています:  
##   
## intersect, setdiff, setequal, union

library(tidyr)  
library(lubridate)

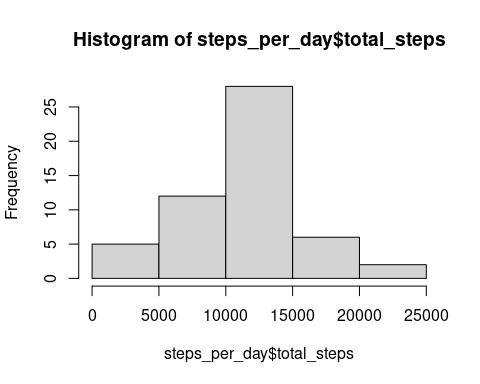
## 要求されたパッケージ timechange をロード中です

##   
## 次のパッケージを付け加えます: 'lubridate'

## 以下のオブジェクトは 'package:base' からマスクされています:  
##   
## date, intersect, setdiff, union

df <- read.csv("activity.csv")

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

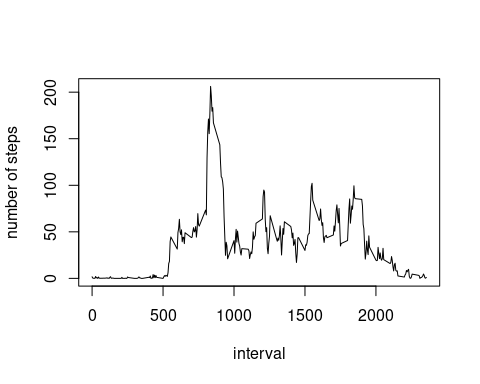


## [1] 10766.19

## What is the average daily activity pattern?

1. Make 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 days (y-axis)

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")



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

max\_step <- max(interval\_5min$int\_steps)  
max\_5min <- interval\_5min %>%   
 filter(int\_steps == max\_step)  
max\_5min

## # A tibble: 1 × 2  
## interval int\_steps  
## <int> <dbl>  
## 1 835 206.

## Imputing missing values

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

include\_NA <- df %>%   
 filter(is.na(steps))  
number\_NA <- nrow(include\_NA)  
number\_NA

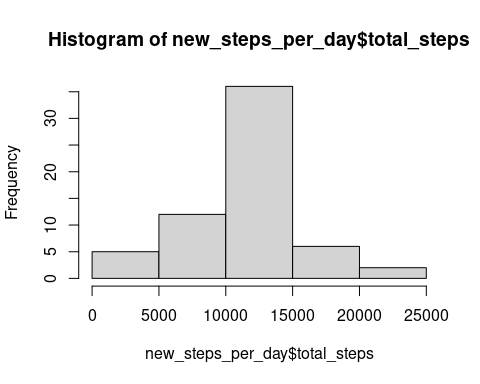
## [1] 2304

1. 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.
2. Create a new dataset that is equal to the original dataset but with the missing data filled in. “new\_dataset” is the original dataset but with the missing data filled in.

# 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)

1. 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.

new\_steps\_per\_day <- new\_dataset%>%   
 group\_by(date) %>%   
 summarize(total\_steps=sum(new\_steps))  
hist(new\_steps\_per\_day$total\_steps)



new\_mean\_steps\_per\_day <- mean(new\_steps\_per\_day$total\_steps, na.rm=TRUE)  
new\_mean\_steps\_per\_day

## [1] 10766.19

new\_median\_steps\_per\_day <- median(new\_steps\_per\_day$total\_steps, na.rm=TRUE)  
new\_median\_steps\_per\_day

## [1] 10766.19

## 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.

new\_dataset\_weekday<- new\_dataset %>%   
 mutate(weekday = wday(date)) %>%   
 mutate(weekday = ifelse(weekday %in% c(2,3,4,5,6),"weekday","weekend"))

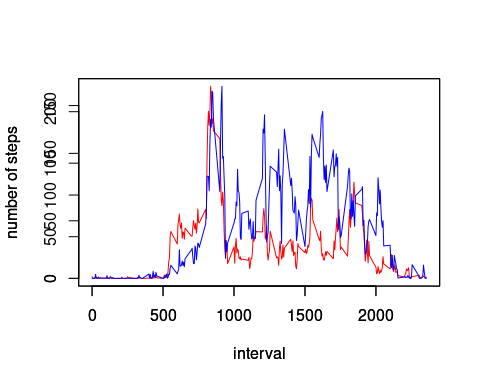
1. Make a panel plot containing a time series plot (i.e.type = “l”) of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis).

Plot weekday data as red line, and weekend data as blue line.

interval\_5min\_weekday <- new\_dataset\_weekday %>%   
 group\_by(weekday,interval) %>%   
 summarize(int\_steps=mean(new\_steps, na.rm=TRUE))

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

# abstruct weekday data. And plot.  
weekday\_data <- interval\_5min\_weekday %>%   
 filter(weekday == "weekday")  
plot(weekday\_data$interval, weekday\_data$int\_steps, type="l", ylab="number of steps", xlab="interval",col="red")  
par(new=T)  
  
#abstract weekend data. And plot.  
weekend\_data <- interval\_5min\_weekday %>%   
 filter(weekday == "weekend")  
plot(weekend\_data$interval, weekend\_data$int\_steps, type="l", ylab="number of steps", xlab="interval", col="blue")



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