project-1

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

COURSE PROJECT 1

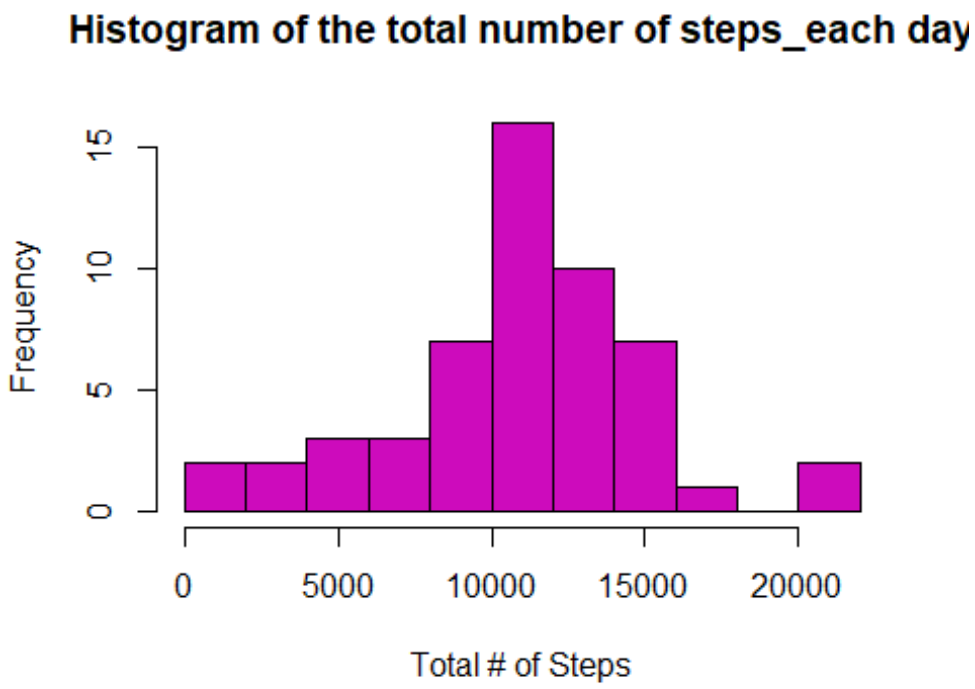
1)) Code for reading in the dataset

2)) Histogram of the total number of steps

The dataset is stored in a comma-separated-value (CSV) file and there are a total of17,568 observations in this dataset

```
#reading the code
data <- read.csv("activity.csv", header = TRUE, sep = ",", na.strings =
"NA")
#format of the code (was taken in YYYY-MM-DD format)
data$date <- as.Date(data$date, format = "%Y-%m-%d")
steps_e_day <- aggregate(steps ~ date, data = data, sum)
colnames(steps_e_day) <- c("date", "steps")

#the histogram of the number of steps
hist((steps_e_day$steps), breaks = 10, col = "78", xlab = "Total # of
Steps", main= "Histogram of the total number of steps_each day")
```



3))Mean and median number of steps(taken each day)

```
#the mean and median
m<-mean(steps_e_day$steps)
p<-median(steps_e_day$steps)

#the prints of mean and medium

print(m)

## [1] 10766.19

print(p)

## [1] 10765
```

6)) Code to describe and show a strategy for imputing missing data

7)) Histogram of the total number of steps taken each day after missing

```
#we return of the data format
data$date <- as.Date(data$date, format = "%Y-%m-%d")
data$interval <- factor(data$interval)
#we create a variable of nas character
NA_i <- is.na(as.character(data$steps))
```

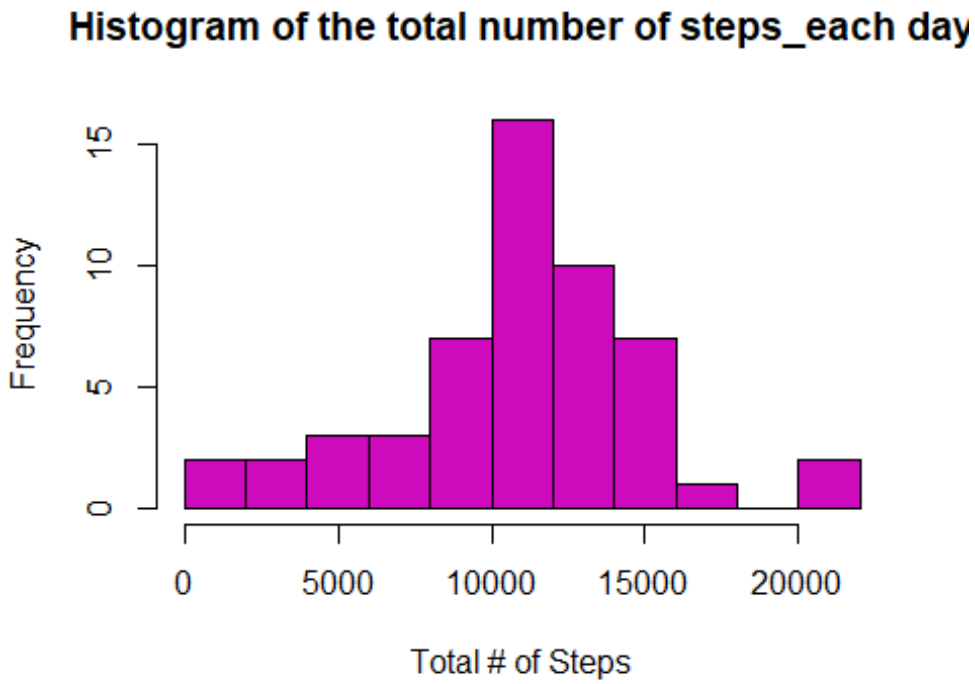
```r
#we use it in the data
data_NA <- data[!NA_i,]
steps_e_day <- aggregate(steps ~ date, data = data_NA, sum)
colnames(steps_e_day) <- c("date", "steps")
```

prove that the data dont have NAs

```r
summary(data_NA)
```

```
##      steps             date               interval
##  Min.   :  0.00   Min.   :2012-10-02   0      :   53
##  1st Qu.:  0.00   1st Qu.:2012-10-16   5      :   53
##  Median :  0.00   Median :2012-10-29   10     :   53
##  Mean   : 37.38   Mean   :2012-10-30   15     :   53
##  3rd Qu.: 12.00   3rd Qu.:2012-11-16   20     :   53
##  Max.   :806.00   Max.   :2012-11-29   25     :   53
##                                        (Other):14946
```

```r
hist((steps_e_day$steps), breaks = 10, col = "78", xlab = "Total # of
Steps", main= "Histogram of the total number of steps_each day")
```



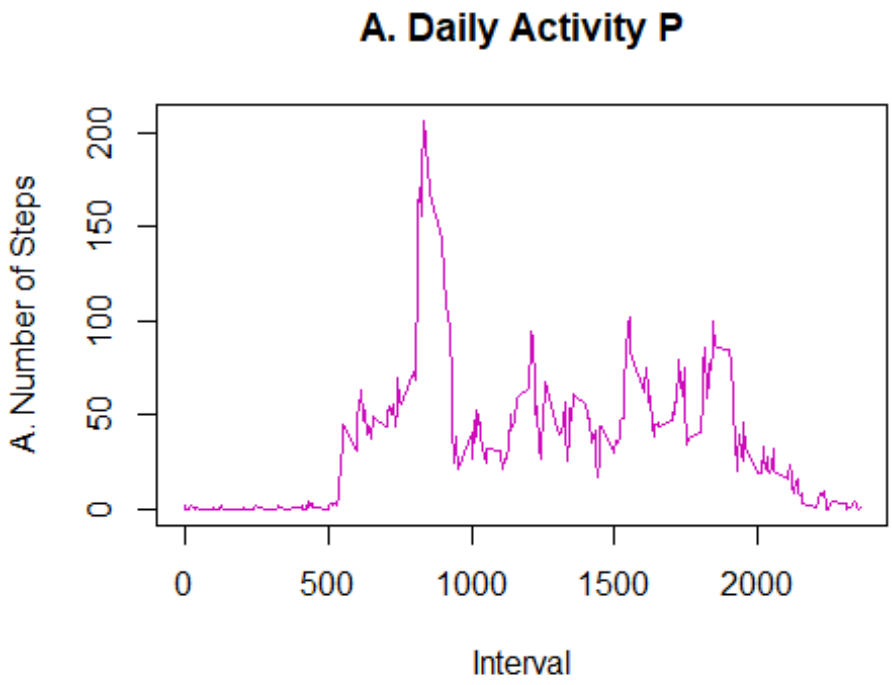
4)) Time series plot of the average number of steps taken

```r
#average of steps
steps_p_inter <- aggregate(data_NA$steps,
by=list(interval=data_NA$interval), FUN=mean)

#columns names
```

```r
colnames(steps_p_inter) <- c("interval", "average_steps")

#ploting the average
plot(as.integer(levels(steps_p_inter$interval)),
steps_p_inter$average_steps, type="l",
     xlab = "Interval", ylab = "A. Number of Steps", main = "A. Daily
Activity P",  col ="78")
```



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

```r
#average number of the steps
max(steps_p_inter$average_steps)

## [1] 206.1698

#maximum of the steps
steps_p_inter[which.max(steps_p_inter$average_steps),]$interval

## [1] 835
## 288 Levels: 0 5 10 15 20 25 30 35 40 45 50 55 100 105 110 115 120 125
... 2355
```

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

Upload the packages

```
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(lubridate)

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

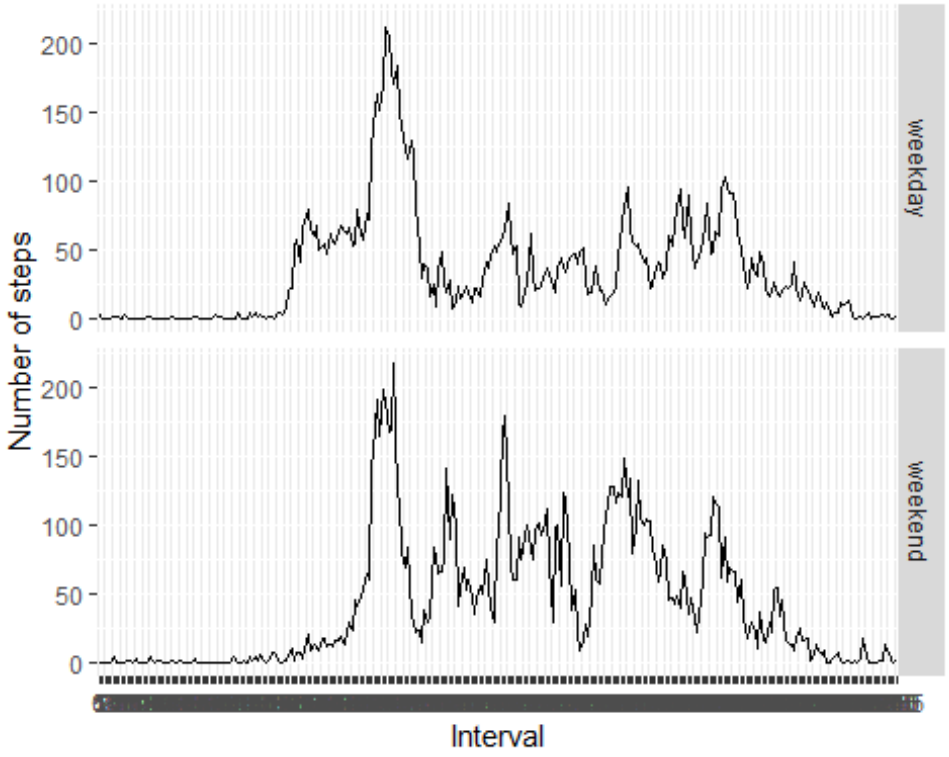
use a function and then we use plots for the comparing

```
com_data <- data_NA
#use a funtion to assign one day from week
daywk <- function(date) {
  wday <- wday(date)
  iswkend <- wday == 1 | wday == 6
  daywk <- character(length = length(date))
  daywk[iswkend] <- "weekend"
  daywk[!iswkend] <- "weekday"

  return(as.factor(daywk))
}
#we create a group for the comparison
com_data <- com_data %>% mutate(daywk = daywk(date))
stepspintervd <- com_data %>%
  group_by(interval, daywk) %>%
  summarize(mean = mean(steps, na.rm = TRUE))

## `summarise()` regrouping output by 'interval' (override with `.groups`
argument)

library(ggplot2)
#we use ggplot for look the comparison
gra <- ggplot(stepspintervd, aes(interval, mean,group =1 ))
gra <- gra + facet_grid(daywk ~ .)
gra <- gra + geom_line()
gra + xlab("Interval") + ylab("Number of steps")
```



9)) All of the R code needed to reproduce the results (numbers, plots, etc.) in the report

```
#reading the code
data <- read.csv("activity.csv", header = TRUE, sep = ",", na.strings =
"NA")
#format of the code (was taken in YYYY-MM-DD format)
data$date <- as.Date(data$date, format = "%Y-%m-%d")
steps_e_day <- aggregate(steps ~ date, data = data, sum)
colnames(steps_e_day) <- c("date", "steps")

#the histogram of the number of steps
hist((steps_e_day$steps), breaks = 10, col = "78", xlab = "Total # of
Steps", main= "Histogram of the total number of steps_each day")

#the mean and median
m<-mean(steps_e_day$steps)
p<-median(steps_e_day$steps)

#the prints of mean and medium

print(m)
print(p)

data$date <- as.Date(data$date, format = "%Y-%m-%d")
data$interval <- factor(data$interval)
```

```
NA_i <- is.na(as.character(data$steps))
data_NA <- data[!NA_i,]
steps_e_day <- aggregate(steps ~ date, data = data_NA, sum)
colnames(steps_e_day) <- c("date", "steps")
hist((steps_e_day$steps), breaks = 10, col = "78", xlab = "Total # of
Steps", main= "Histogram of the total number of steps_each day")

#average of steps
steps_p_inter <- aggregate(data_NA$steps,
by=list(interval=data_NA$interval), FUN=mean)

#columns names
colnames(steps_p_inter) <- c("interval", "average_steps")

#ploting the average
plot(as.integer(levels(steps_p_inter$interval)),
steps_p_inter$average_steps, type="l",
     xlab = "Interval", ylab = "A. Number of Steps", main = "A. Daily
Activity P",  col ="78")
#average number of the steps
max(steps_p_inter$average_steps)
#maximum of the steps
steps_p_inter[which.max(steps_p_inter$average_steps),]$interval

library(dplyr)
library(lubridate)

com_data <- data_NA
daywk <- function(date) {
  wday <- wday(date)
  iswkend <- wday == 1 | wday == 6
  daywk <- character(length = length(date))
  daywk[iswkend] <- "weekend"
  daywk[!iswkend] <- "weekday"

  return(as.factor(daywk))
}

com_data <- com_data %>% mutate(daywk = daywk(date))
stepspintervd <- com_data %>%
  group_by(interval, daywk) %>%
  summarize(mean = mean(steps, na.rm = TRUE))
library(ggplot2)
gra <- ggplot(stepspintervd, aes(interval, mean,group =1 ))
gra <- gra + facet_grid(daywk ~ .)
gra <- gra + geom_line()
gra + xlab("Interval") + ylab("Number of steps")
```