Peer_graded assingment, Week 2, Reproducible Research

Aranda N. Alfredo

Reading, cleaning and undesrtanding data set files

```
# Undertanding datasets
activity <- read.csv("activity.csv", sep = ',', header = TRUE)
View(activity)
head(activity)
```

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

```
dim(activity)
```

```
## [1] 17568     3
```

```
class(activity)
```

```
## [1] "data.frame"
```

```
str(activity)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

```
activity$steps <- as.numeric(activity$steps)
activity$interval <- as.numeric(activity$interval)
activityNaRm <- activity[complete.cases(activity), ]
```

What is mean total number of steps taken per day?

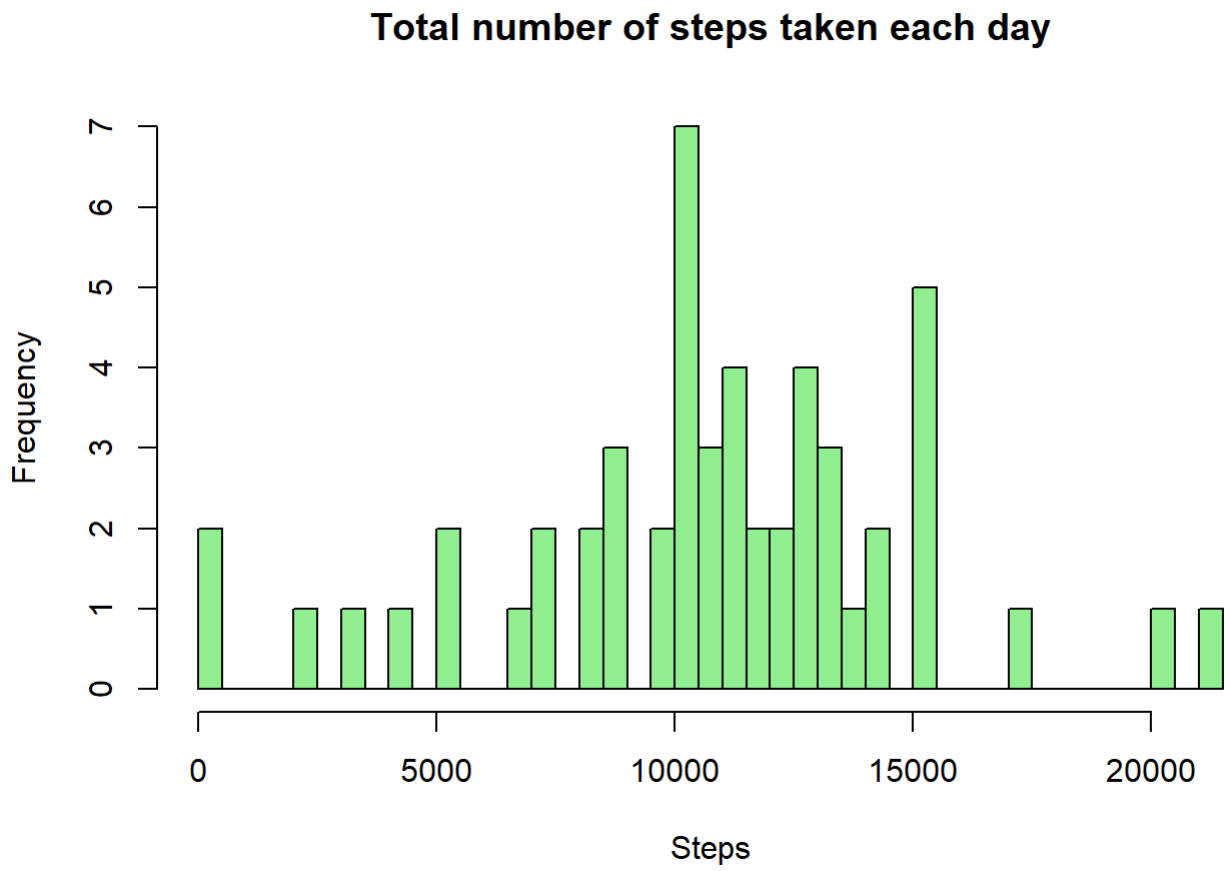
```
meanSteps <-aggregate(x = activityNaRm[c("steps")],
                      FUN = mean,
                      by = list(activityNaRm$date))
names(meanSteps) <- c("dates", "steps")
```

1.Calculate the total number of steps taken per day

```
sumSteps <-aggregate(x = activityNaRm[c("steps")],
                      FUN = sum,
                      by = list(activityNaRm$date))
names(sumSteps) <- c("dates", "steps")
```

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

```
hist(sumSteps$steps,
     breaks=53,
     main = "Total number of steps taken each day",
     xlab = "Steps",
     col = "lightgreen",
     border = "black"
)
```



3.Calculate and report the mean and median of the total number of steps taken per day

```
dim <- length(sumSteps$steps)
totalMean <- sum(sumSteps$steps)/dim
totalMedian <- median(sum(sumSteps$steps)/dim)
```

The total mean of the number of steps taken per day is 1.076618910^{4} and the total median of the number of steps taken per day is 1.076618910^{4}

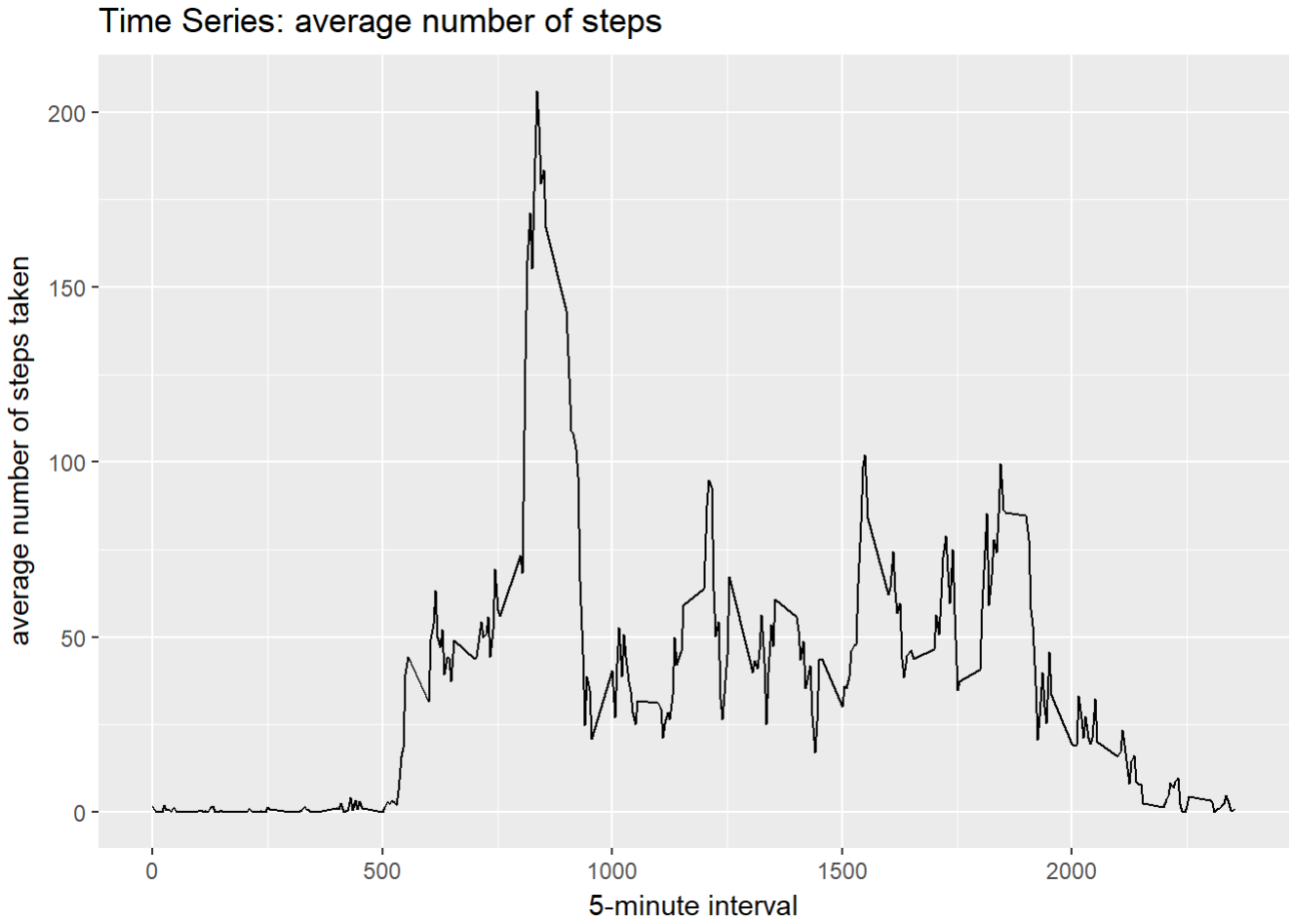
What is the average daily activity pattern?

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)

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.3
```

```
averages <- aggregate(x=list(steps=activity$steps), by=list(interval=activity$interval),FUN=mea
n, na.rm=TRUE)
ggplot(data=averages, aes(x=interval, y=steps)) +
      geom_line() +
      ggtitle("Time Series: average number of steps") +
      xlab("5-minute interval") +
      ylab("average number of steps taken")
```



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

```
maxSteps <- max(meanSteps$steps)
maxS <- meanSteps[meanSteps$steps == maxSteps,]
maxD <- maxS$dates
```

The day it constains the maximum number of steps is 2012-11-23

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)

```
totalNA <- length(activity$steps) - length(activityNaRm$steps)
```

The total numbers of missing values (NA) in the datasets is 2304

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.

I realized a strategy as follow: If the first or last day have NA then replace it with the global average without NA. Next, if the position i (do not consider first and last positions) has NA then this one takes the averege between i+1 and i-1. Finally, if 1+i or i-1 have NA then replace it with the global average without NA too.

```
activity2 <- activity
d <- length(activity$steps)
for (i in 1:d) {
      if (is.na(activity2[i,1])) {
            activity2[i,1] <- averages[averages$interval == activity2[i,3],2]
            activity2[i,4] <- "check"
      }
}
```

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

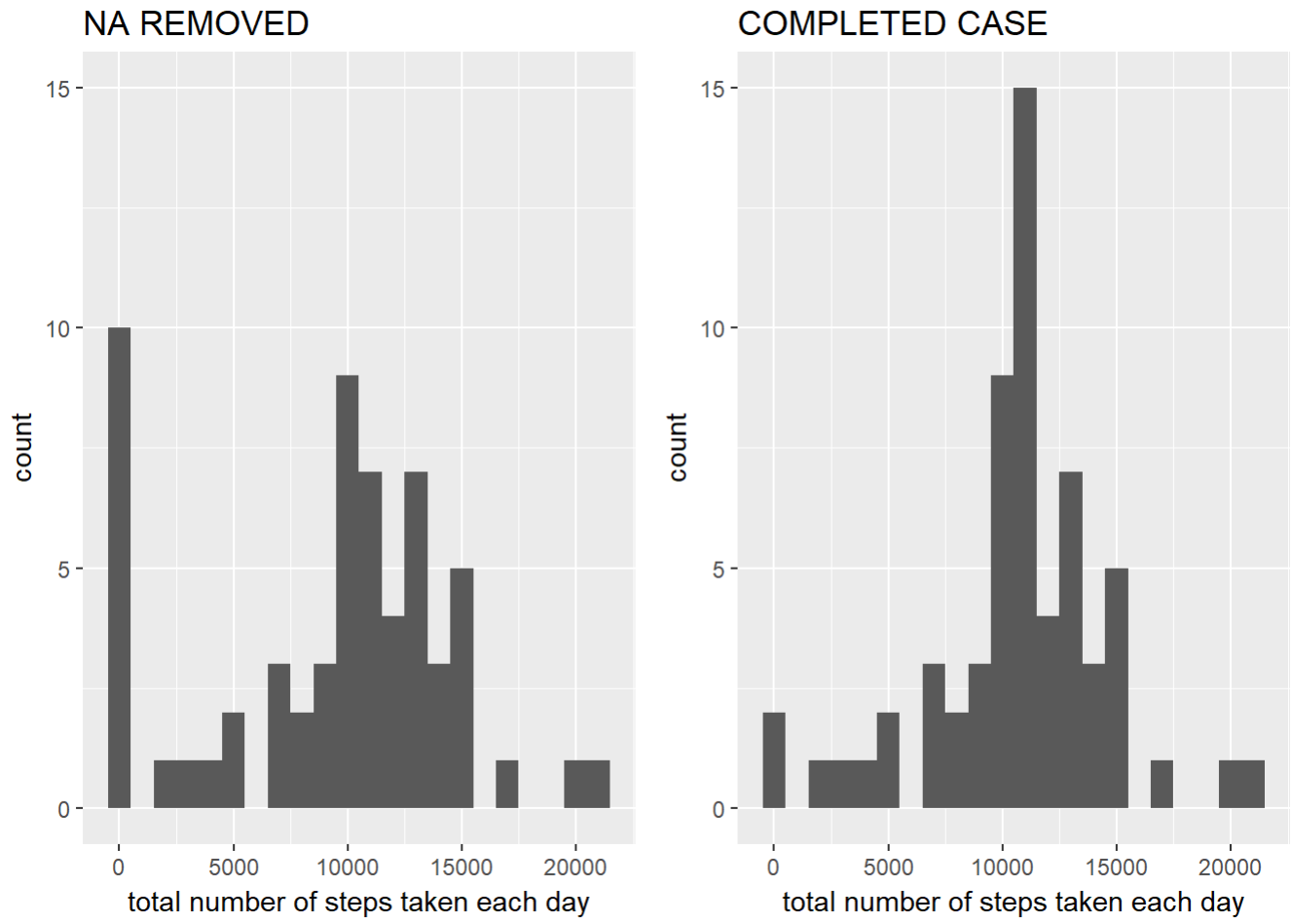
```
activity2 <- activity2
```

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.

```
# Preparing data for histogram
actsteps <- tapply(activity$steps, activity$date, FUN=sum, na.rm=TRUE)
stepsMean <- mean(actsteps, na.rm=TRUE)
actsteps2 <- tapply(activity2$steps, activity2$date, FUN=sum)
stepsMean2 <- mean(actsteps2)
dif <-abs(stepsMean - stepsMean2)
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.4.3
```

```
library(ggplot2)

# Plots
p1<-qplot(actsteps,
               binwidth=1000,
               ylim=c(0,15),
               main="NA REMOVED",
               xlab="total number of steps taken each day")

p2<-qplot(actsteps2,
               binwidth=1000,
               ylim=c(0,15),
               main="COMPLETED CASE",
               xlab="total number of steps taken each day")
grid.arrange(p1, p2, ncol=2)
```



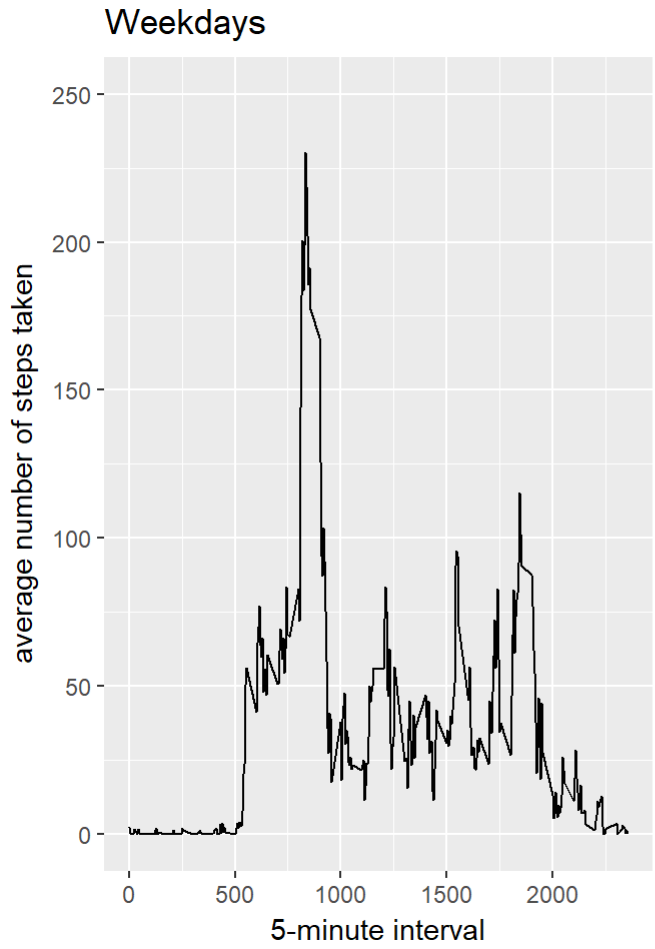
Are there differences in activity patterns between weekdays and weekends?

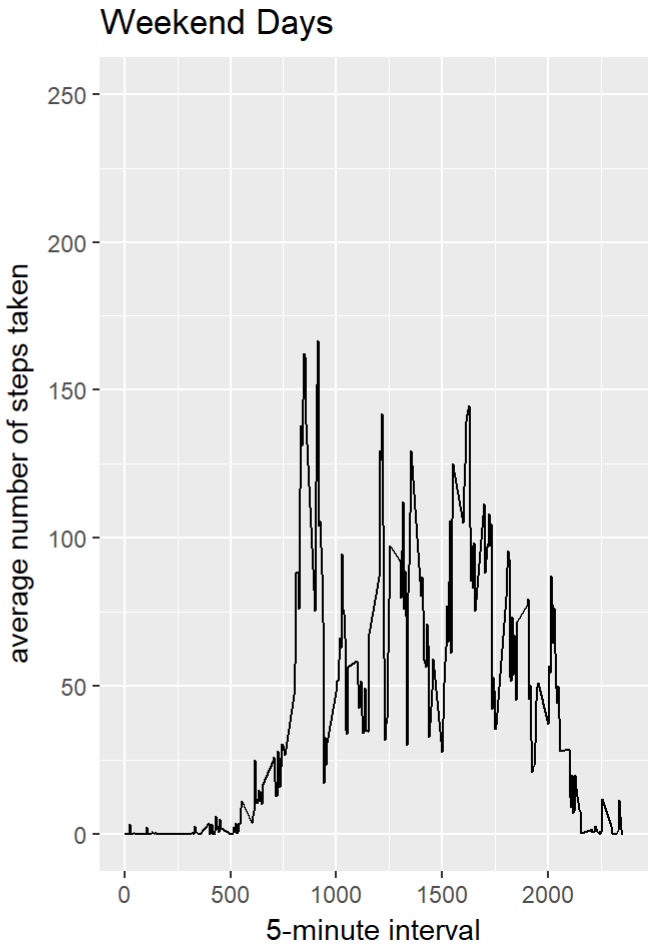
1.Create a new factor variable in the dataset with two levels - “weekday” and “weekend” indicating whether a given date is a weekday or weekend day.

```
# Taking as factor date
activity2$date <- as.Date(activity2$date)
activity2$WD <- weekdays(activity2$date)
activity2$WDG <- "week"

# Days names in Chile
for (i in 1:length(activity2$steps)) {
      if (activity2[i,5] == "sábado" | activity2[i,5] == "domingo") {
            activity2[i,6] <- "weekend"
      }
}
activity2[,6] <- as.factor(activity2[,6])
```

2.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)

```
# Compute means by intervals
activity2w <-subset(activity2,activity2[,6]=="week")
activity2we <-subset(activity2,activity2[,6]=="weekend")
averagesW <- aggregate(steps ~ interval, activity2w, FUN=mean)
averagesWe <- aggregate(steps ~ interval, activity2we, FUN=mean)

# Preparing plots
plot1 <- ggplot(data=averagesW, aes(x=interval, y=steps)) +
         geom_line() +
         ylim(0, 250) +
          ggtitle("Weekdays") +
         xlab("5-minute interval") +
         ylab("average number of steps taken")

plot2 <- ggplot(data=averagesWe, aes(x=interval, y=steps)) +
         geom_line() +
         ylim(0, 250) +
         ggtitle("Weekend Days") +
         xlab("5-minute interval") +
         ylab("average number of steps taken")
library(gridExtra)
grid.arrange(plot1, plot2, ncol=2)
```





The maximum steps is in Weekdays.