# Reproducible Research: HW 1

## Loading and preprocessing the data

read data from “activity.csv” to a data.frame, “alldata”.

```
alldata<-read.csv("activity.csv")
alldata$interval<-sapply(alldata$interval,function(x) paste(paste(rep("0",4-nchar(as.char
acter(x))),sep="",collapse = ''),as.character(x),sep=""))
alldata$date <- as.Date(alldata$date, format = "%Y-%m-%d")
```

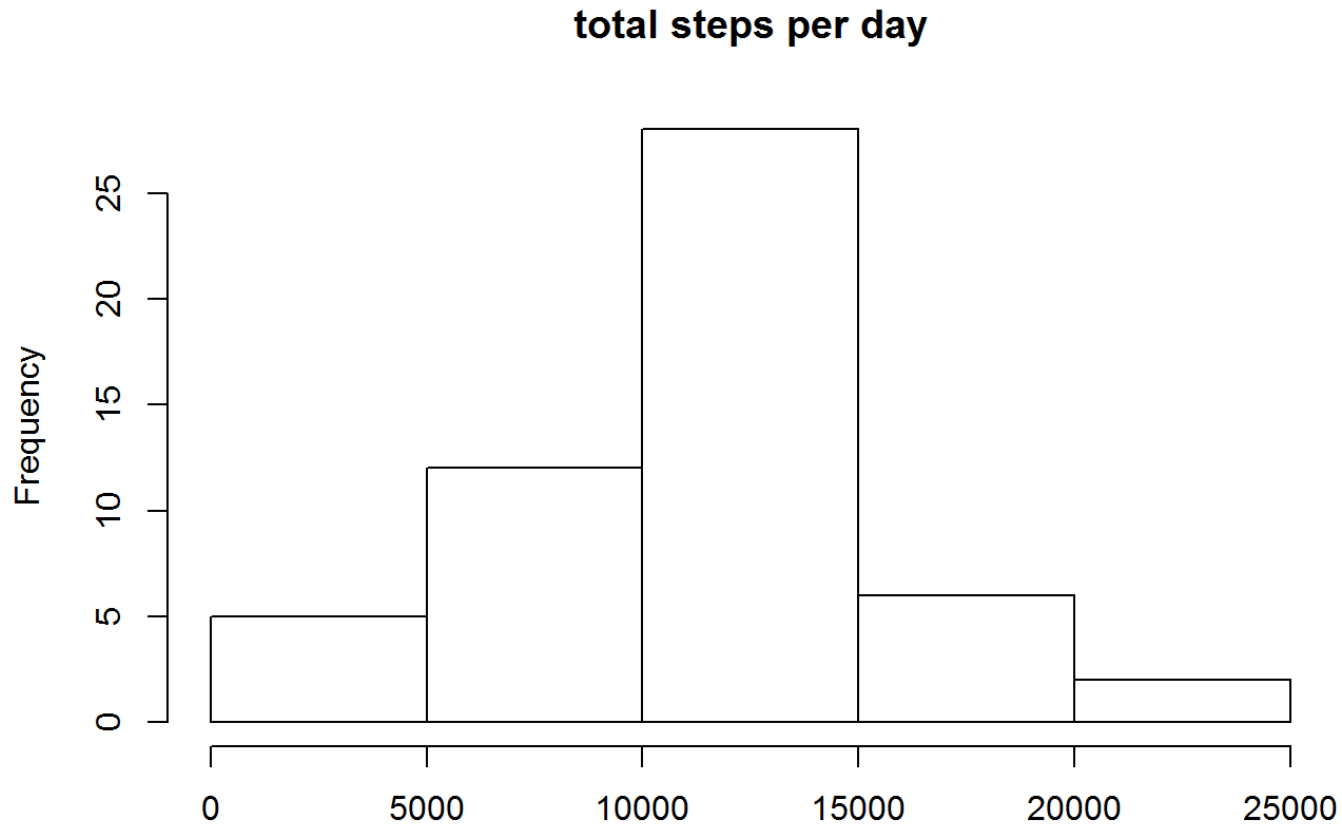
copy all the non-NA observation to a data.frame, “cleandata”.

```
cleandata <- alldata[!is.na(alldata$steps),]
```

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

Calculate the averaged total steps per day and put them in “data_steps”.

```
date_steps<-aggregate(cleandata$steps,list(date=cleandata$date),sum)
colnames(date_steps)[2]<-"total_steps"
hist(date_steps$total_steps,main="total steps per day",xlab="")
```




```
mean(date_steps$total_steps)
```

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

The mean of total steps per day are 11766.19.

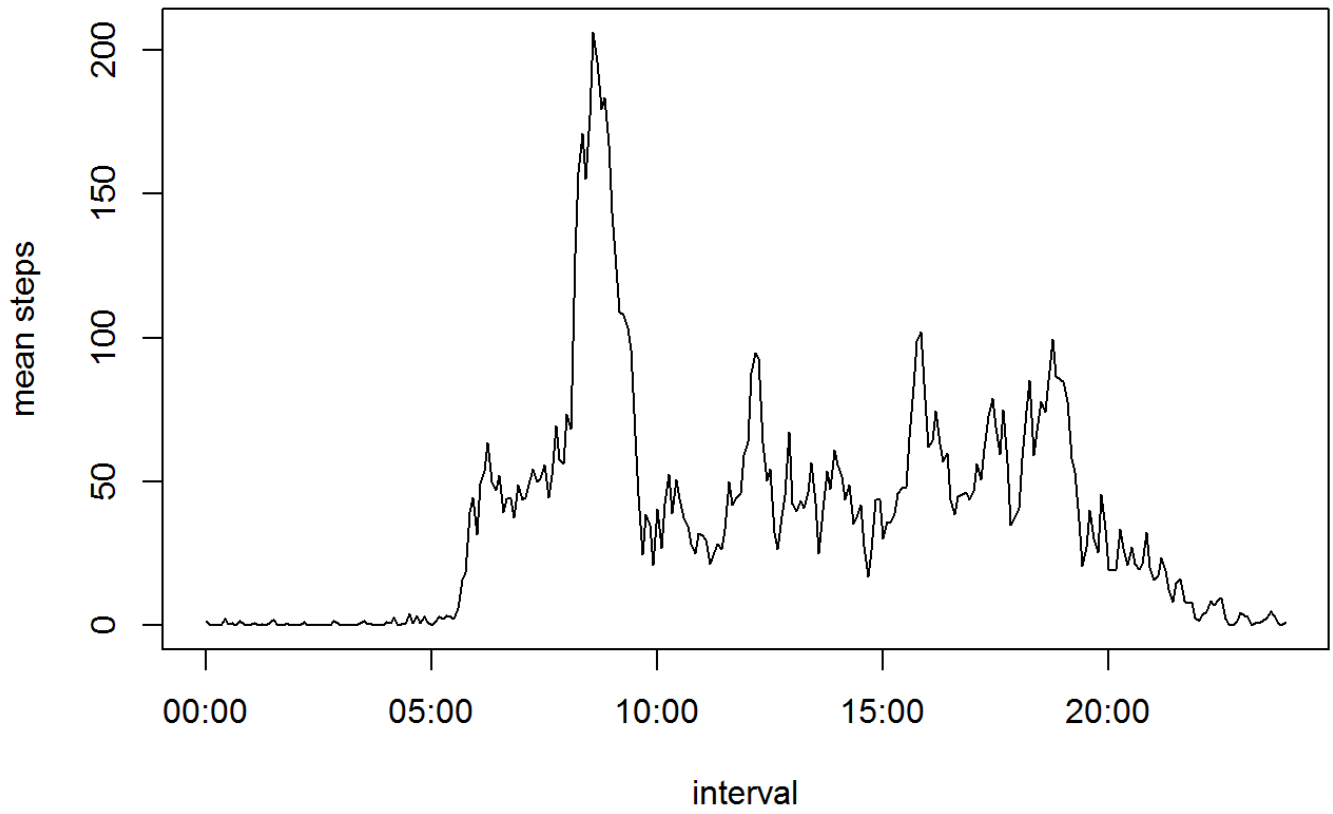
```
median(date_steps$total_steps)
```

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

The median of total steps per day are 10765.

# What is the average daily activity pattern?

```
interval_steps<-aggregate(cleandata$steps,list(inte=cleandata$interval),mean)
plot(strptime(interval_steps$inte,"%H%M"),interval_steps$x,type = "l",xlab="interval",yla
b="mean steps")
```




Calculate the interval which has the maximum mean steps:

```
idx<-which(interval_steps$x==max(interval_steps$x))
interval_steps$inte[idx]
```

```
## [1] "0835"
```

The mean steps has its maximum at interval 8:35.

# Imputing missing values

Calculate the total number of missing values in the dataset:

```
sum(is.na(alldata$steps))
```

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

The total number of missing values is 2304.

Fill the NA slots with the mean value of the steps at the same interval of other days.

First we create a function that takes values of steps from the corresponding time period if the value of steps is NA.

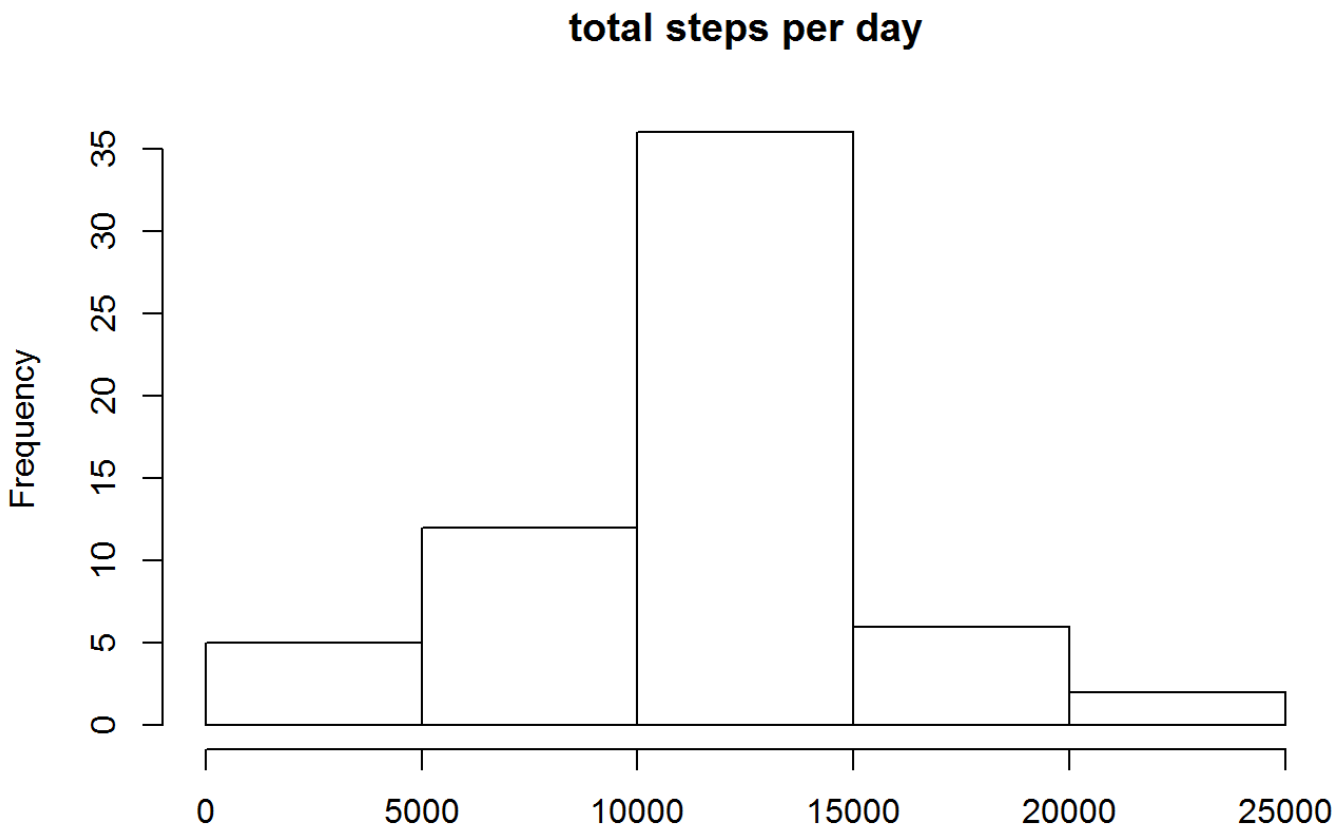
```
rmna <- function (x,y) {
  if(is.na(x)) {
    interval_steps[which(interval_steps$inte==(y)),2]
  } else {x}
}
```

Now mapply the function to create a data set “alldata2” with all NA value filled.

```
alldata2<-alldata
alldata2$steps<-mapply(rmna,alldata$steps,alldata$interval)
```

plot the histogram again with alldata2.

```
date_steps2<-aggregate(alldata2$steps,list(date=alldata2$date),sum)
colnames(date_steps2)[2]<-"total_steps"
hist(date_steps2$total_steps,main="total steps per day",xlab="")
```




```
mean(date_steps2$total_steps)
```

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

The mean of total steps per day are the same as before, 11766.19.

```
median(date_steps2$total_steps)
```

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

The median of total steps per day becomes 11766.19, the same as the mean value.

# Are there differences in activity patterns between weekdays and weekends?

add one more column to “alldata2

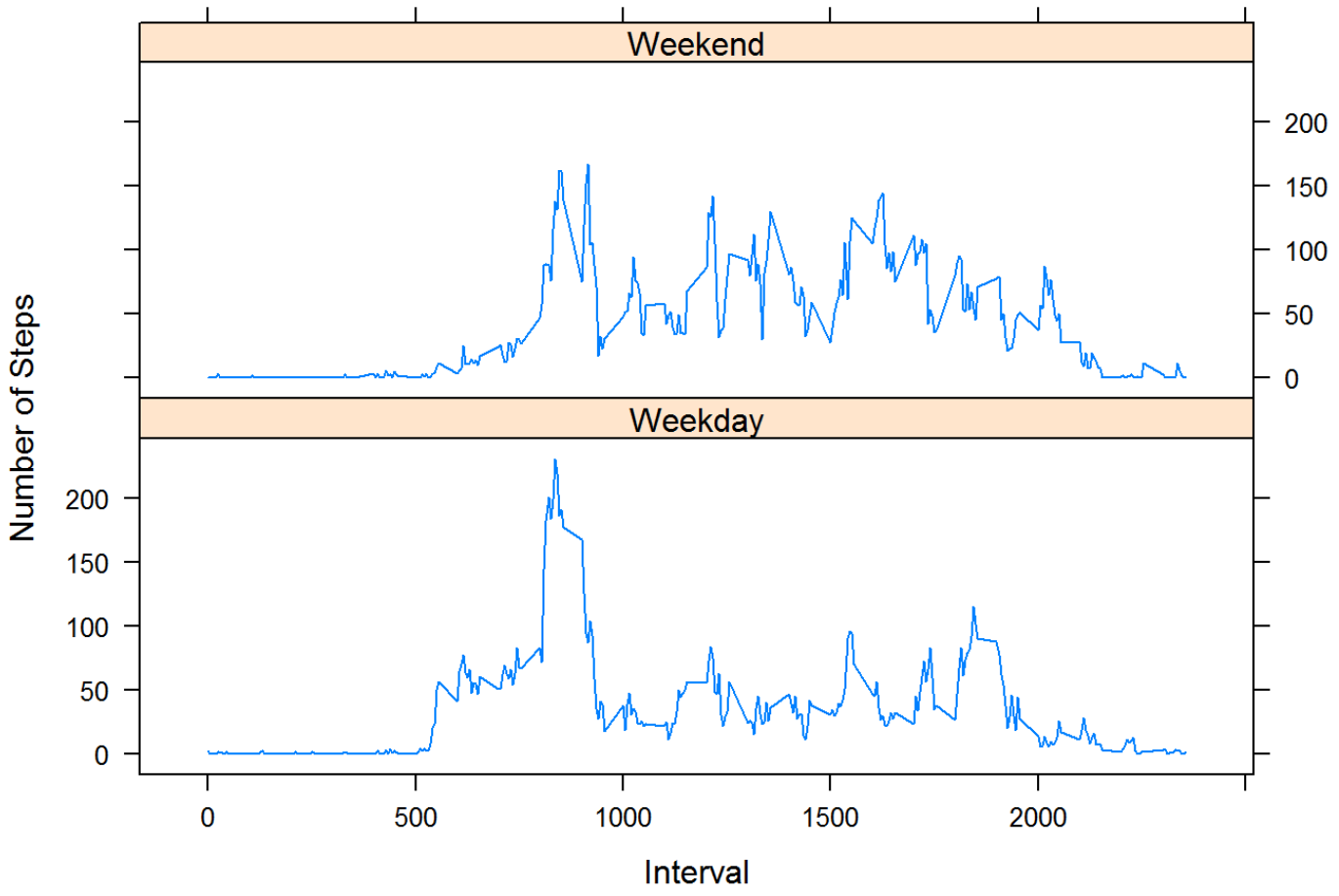
```
tmp<-sapply(weekdays(alldata2$date),function(x) if (x=="Sunday"|x=="Saturday") {"Weeken
d"} else {"Weekday"})
alldata2<-cbind(alldata2,wd=as.factor(tmp))
```

calculate the mean values of steps for the weekend and weekday.

```
alldata3<-aggregate(alldata2$steps,list(inte=alldata2$interval,wd=alldata2$wd),mean)
```

create the panel plot.

```
#alldata3$inte<-strptime(alldata3$inte,"%H%M")
alldata3$inte<-as.numeric(alldata3$inte)
library(lattice)
xyplot(x~inte|wd,alldata3, layout=c(1,2),xlab="Interval",ylab="Number of Steps",type='l')
```




The difference is obvious.

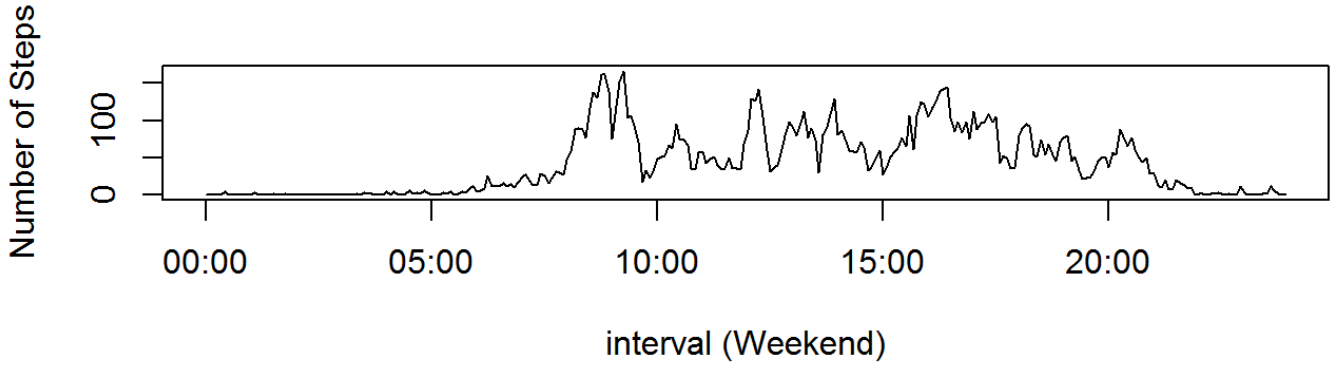
# Some comments

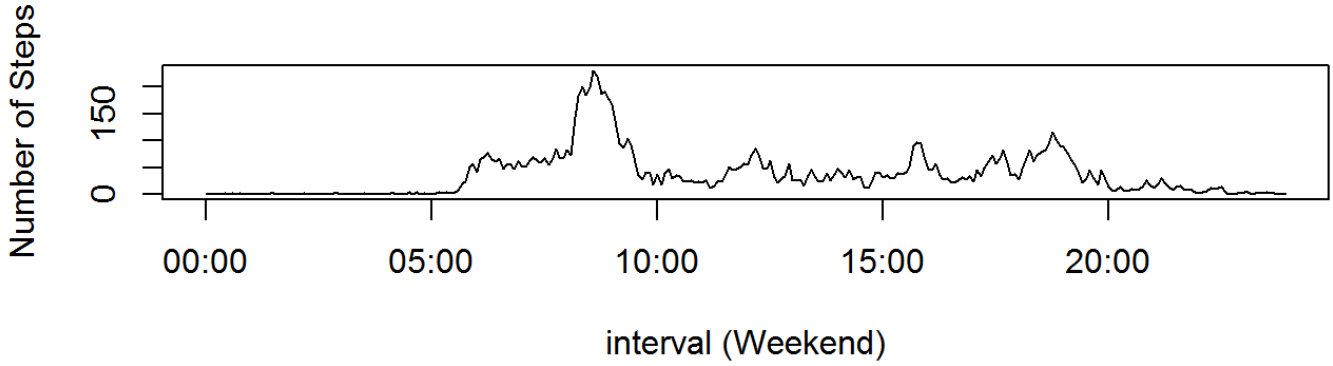
In the previous figure, there are some funny discontinuity between every xx55 to xx00 Interval (You can see that in the instructor's simulation figure, too). This is the result of the time format string "HHMM" being interpret as integer numbers.

I avoid this problem by phrasing the time string correctly to POSIXlt format in other figures. However, it appears the lattice plotting system doesn't accept POSIXlt as the x-axis, so I have no choice. The following is the correct figure using the base plotting system.

```
alldata4<-aggregate(alldata2$steps,list(inte=alldata2$interval,wd=alldata2$wd),mean)
alldata4$inte<-strptime(alldata4$inte,"%H%M")

par(mfrow=c(2,1))
plot(alldata4$inte[alldata4$wd=="Weekend"],alldata4$x[alldata4$wd=="Weekend"],type = "l",
xlab="interval (Weekend)",ylab="Number of Steps")
plot(alldata4$inte[alldata4$wd=="Weekday"],alldata4$x[alldata4$wd=="Weekday"],type = "l",
xlab="interval (Weekend)",ylab="Number of Steps")
```


Number of Steps
0
100
00:00
05:00
10:00
15:00
20:00
interval (Weekend)


Number of Steps
0
150
00:00
05:00
10:00
15:00
20:00
interval (Weekend)