Reproducible Research first assign

YanirFigovich

4 2018 בדצמבר

loading the data

loading the data to R and libraries

```
data<- read.csv("activity.csv")
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.4
```

```
##
## 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)
```

```
## Warning: package 'lubridate' was built under R version 3.4.4
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 3.4.4
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

```
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.4.4
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

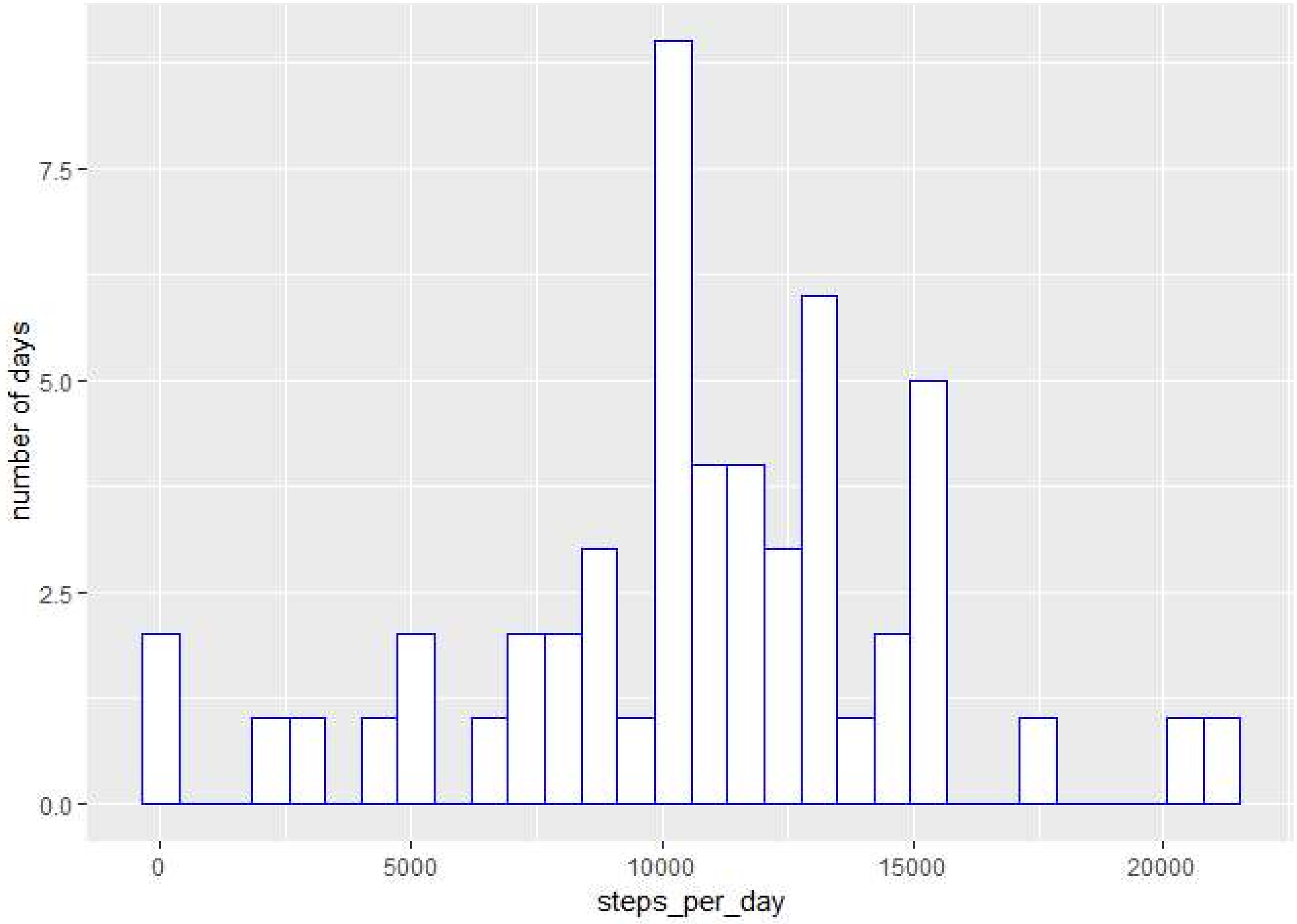
ploting number of steps

preparing and summarising the data in order to make the plot

```
data$date<-as_date(data$date)
agg_data<- group_by(data,date)
agg_data<- summarise(agg_data,steps_per_day=sum(steps))
plot1<- ggplot(agg_data)+geom_histogram(aes(steps_per_day),fill="white",color="blue")
plot1<-plot1+ labs(y="number of days")
plot(plot1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 8 rows containing non-finite values (stat_bin).
```



##find mean and median steps per day

```
mean_steps_day<-mean(agg_data$steps_per_day,na.rm = T)
mean_steps_day
```

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

```
med_steps_day<- median(agg_data$steps_per_day,na.rm = T)
med_steps_day
```

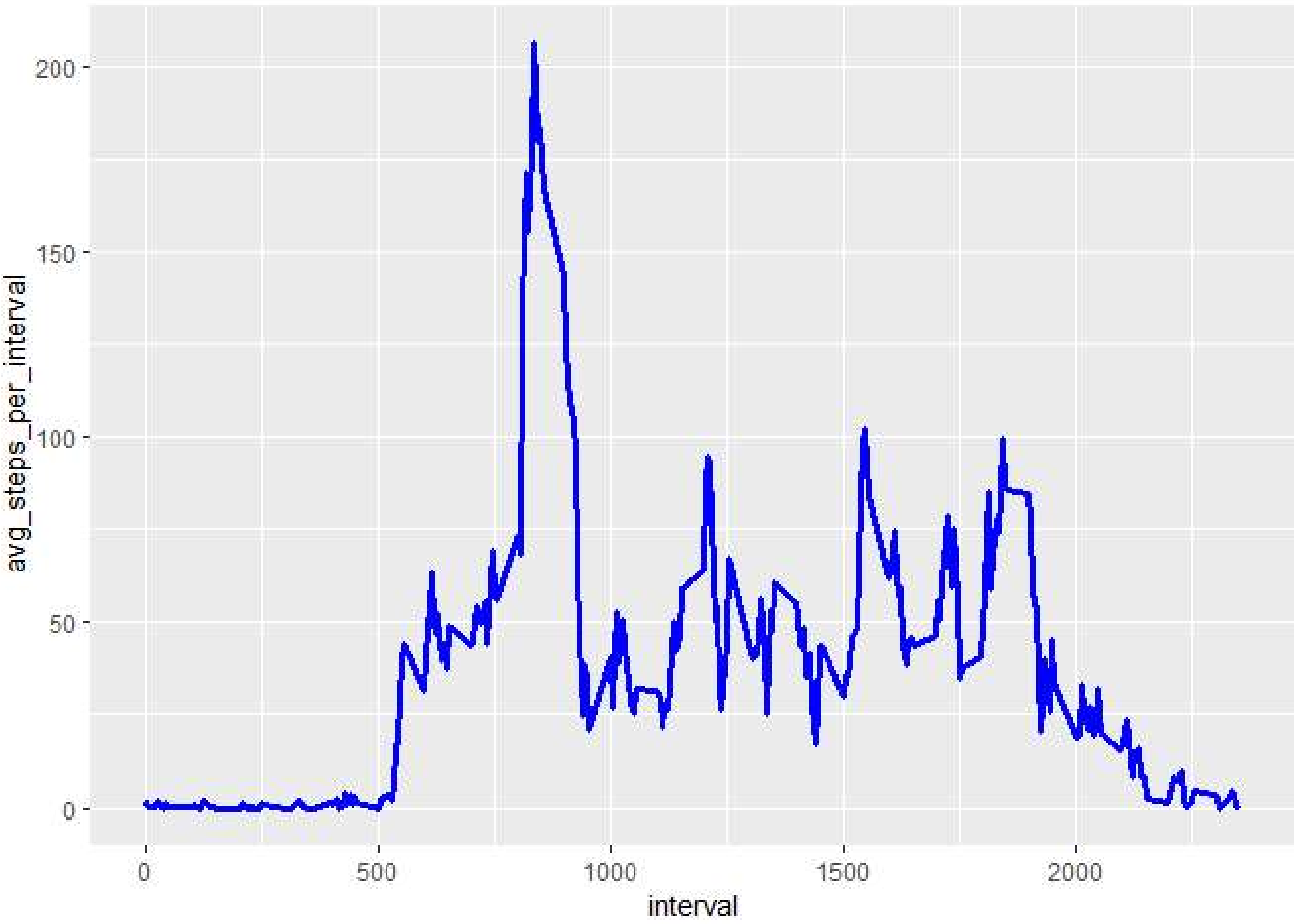
```
## [1] 10765
```

plot average steps for each interval

```
agg_data<- group_by(data,interval)
agg_data<- summarise(agg_data,avg_steps_per_interval=mean(steps,na.rm = T))
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
plot2<-ggplot(agg_data)+geom_line(aes(interval,avg_steps_per_interval),lwd=1.1,color="blue")
plot(plot2)
```



##find interval with max steps on average

```
filter(agg_data,avg_steps_per_interval==max(avg_steps_per_interval))
```

```
## # A tibble: 1 x 2
##   interval avg_steps_per_interval
##      <int>                  <dbl>
## 1      835                   206.
```

imputing NA

first we would like to know where and how much NA we have in the data. then we will impute the mean where ever we find na

```
#finding the NA
sapply(data,function(x){sum(is.na(x))})
```

```
##    steps     date interval 
##     2304        0        0
```

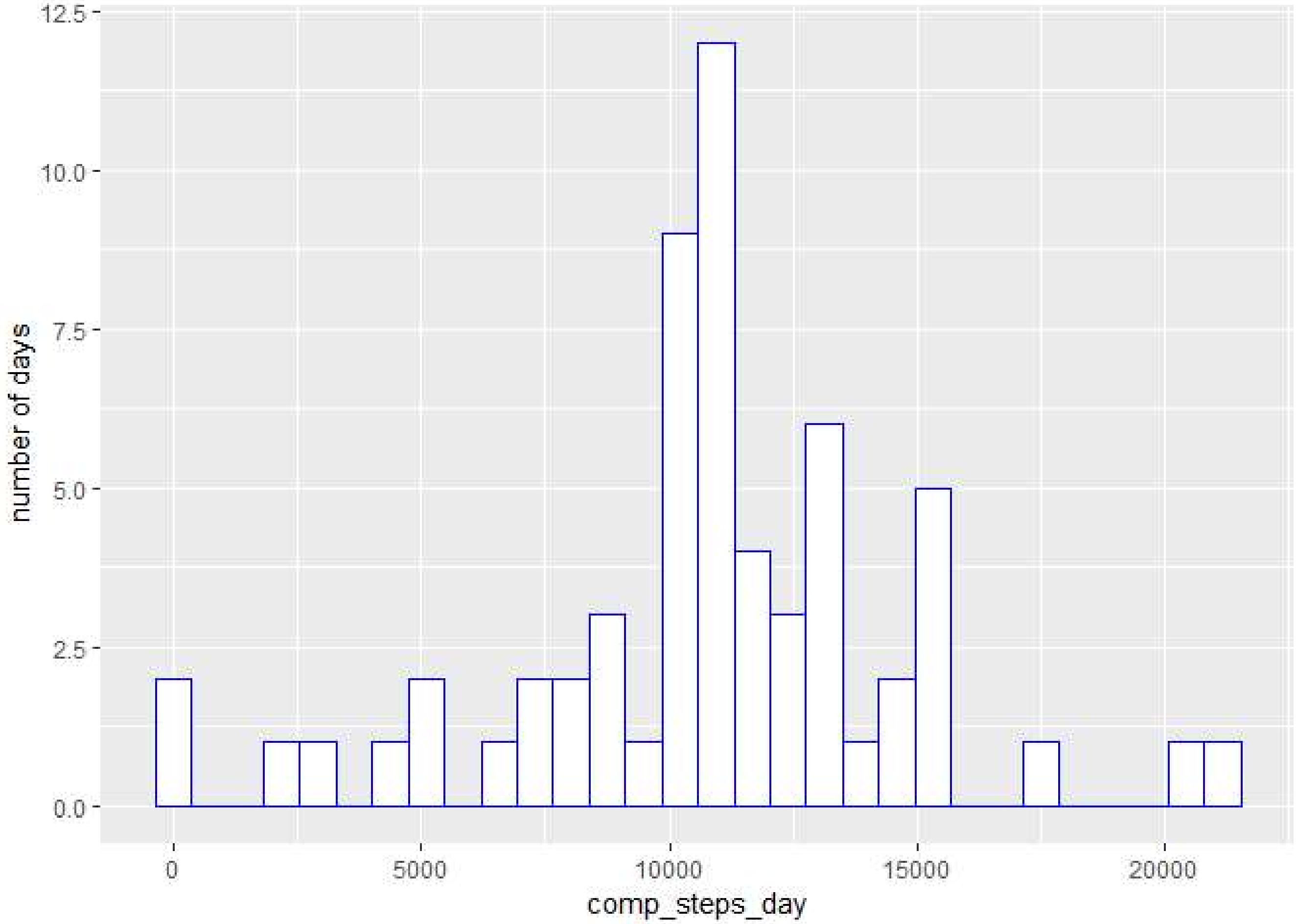
```
##imputing avg steps where steps==NA
#calculating the avg steps for each interval
agg_data<- group_by(data,interval)
agg_data<- summarise(agg_data,avg_steps_per_interval=mean(steps,na.rm = T))
#mreging the base data with the avg steps
merge_data<- left_join(data,agg_data,by="interval")
#computing a new column to select the avg if steps=NA
new_data<- mutate(merge_data,comp_steps=ifelse(is.na(steps),avg_steps_per_interval,steps))
#see some of the new data
head(new_data)
```

```
##   steps       date interval avg_steps_per_interval comp_steps
## 1    NA 2012-10-01        0              1.7169811  1.7169811
## 2    NA 2012-10-01        5              0.3396226  0.3396226
## 3    NA 2012-10-01       10              0.1320755  0.1320755
## 4    NA 2012-10-01       15              0.1509434  0.1509434
## 5    NA 2012-10-01       20              0.0754717  0.0754717
## 6    NA 2012-10-01       25              2.0943396  2.0943396
```

plot average steps for each interval after adding avg steps insted of NA

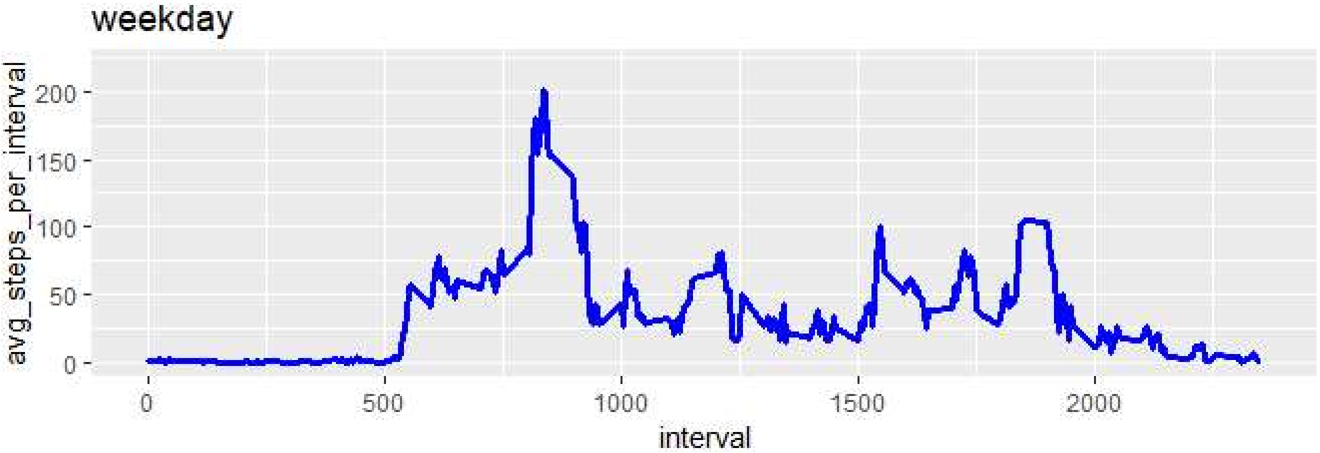
```
agg_by_day<- group_by(new_data,date)
agg_by_day<-summarise(agg_by_day,comp_steps_day=sum(comp_steps))
plot3<- ggplot(agg_by_day)+geom_histogram(aes(comp_steps_day),fill="white",color="blue")
plot3<-plot3+labs(y="number of days")
plot(plot3)
```

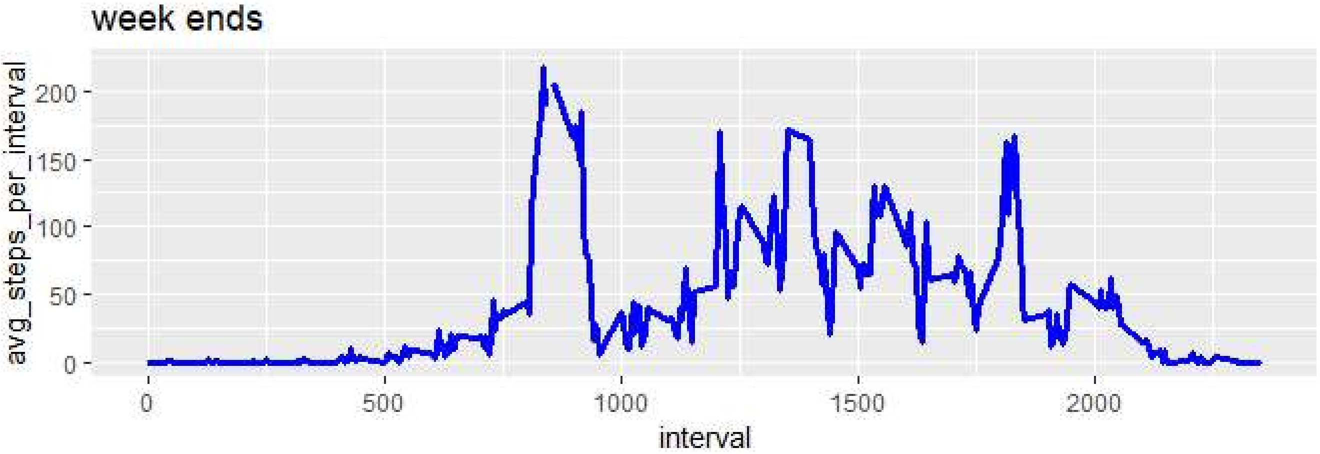
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



##plot average steps per interval comparing weekdays VS weekends we will first add anew column to our data with numeric representive of the day of the week. then we will separate for 2 tables: one for week days and the other for weekends. then we will plot a histogram for each table

```
##adding the numeric weekday
days_data<- mutate(data,week_day=wday(date))
## a table for weekdays
weekdays_data<- filter(days_data,week_day<6)
##aggragating by interval mean
agg_weekdays<- group_by(weekdays_data,interval)
agg_weekdays<- summarise(agg_weekdays,avg_steps_per_interval=mean(steps,na.rm = T))
## a table for weekends
weekdEnds_data<- filter(days_data,week_day>5)
##aggragating by interval mean
agg_weekEnds<- group_by(weekdEnds_data,interval)
agg_weekEnds<- summarise(agg_weekEnds,avg_steps_per_interval=mean(steps,na.rm = T))
plot4<-ggplot(agg_weekdays)+geom_line(aes(interval,avg_steps_per_interval),lwd=1.1,color="blu
e")+labs(title = "weekday")
plot4<-plot4 + scale_y_continuous(limits=c(0, 220))
plot5<-ggplot(agg_weekEnds)+geom_line(aes(interval,avg_steps_per_interval),lwd=1.1,color="blu
e")+labs(title = "week ends")
plot5<-plot5 + scale_y_continuous(limits=c(0, 220))
pp<-grid.arrange(plot4,plot5,nrow=2)
```


weekday
avg_steps_per_interval
200
150
100
50
0
0
500
1000
1500
2000
interval


week ends
avg_steps_per_interval
200
150
100
50
0
0
500
1000
1500
2000
interval