Project1

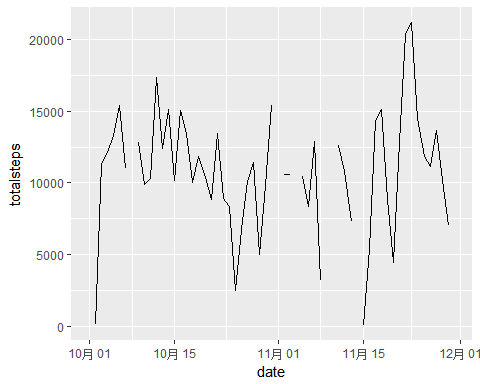
## 1.Code for reading and loading the data

AMdata<-read.csv("C:/Users/Steve/Desktop/Data Science Cousera/Reproducible Analysis/project1/activity.csv",header=TRUE)

## 2.Histogram of the total number of steps taken each day

library(plyr)  
totalstep<-ddply(AMdata,.(date),function(x)sum(x$steps))  
colnames(totalstep)[2]<-"totalsteps"  
totalstep$date<-as.Date(totalstep$date)  
library(ggplot2)  
ggplot(totalstep,aes(x=date,y=totalsteps))+geom\_line()

## Warning: Removed 2 rows containing missing values (geom\_path).



## 3.Mean and median number of steps taken each day

calculate the mean and median

meanstep<-mean(totalstep$totalsteps,na.rm = TRUE)  
medianstep<-median(totalstep$totalsteps,na.rm = TRUE)  
print(meanstep)

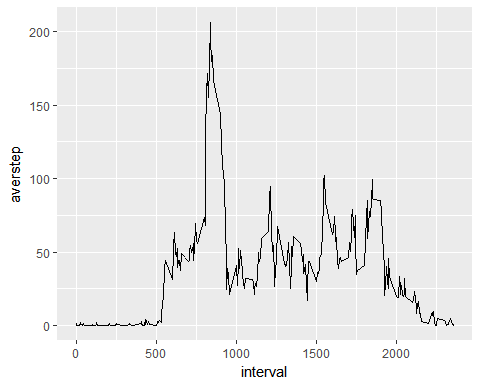
## [1] 10766.19

print(medianstep)

## [1] 10765

## 4.Time series plot of the average number of steps taken

intervalstep<-ddply(AMdata,.(interval),function(x)mean(x$steps,na.rm = TRUE))  
colnames(intervalstep)[2]<-"averstep"  
ggplot(intervalstep,aes(x=interval,y=averstep))+geom\_line()

 ##5.The maximum number of steps in 5-minute interval

maxstep<-max(intervalstep$averstep)  
print(maxstep)

## [1] 206.1698

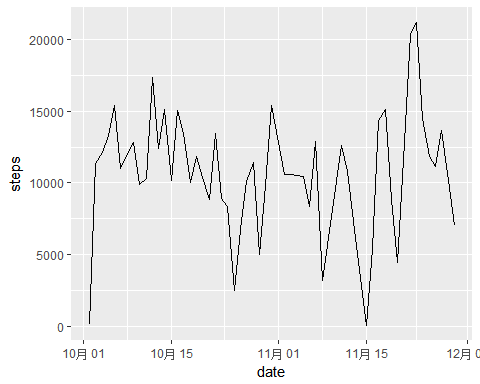
## 6.Code to describe and show a strategy for imputing missing data

I have used ddply to calculate without the missing value

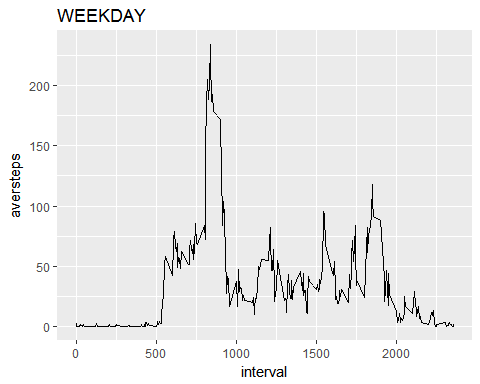
AMnoNA<-AMdata[!is.na(AMdata$steps),]

## 7.Histogram of total number of steps taken each day after imputing missing values

totalstep2<-ddply(AMnoNA,.(date),function(x)sum(x$steps))  
colnames(totalstep2)[2]<-"steps"  
totalstep2$date<-as.Date(totalstep2$date)  
ggplot(totalstep2,aes(x=date,y=steps))+geom\_line()

 ##8.Panel plot comparing average of steps accross weekdays and weekends

AMdata$date<-as.Date(AMdata$date)  
AMdata$weekdays<-weekdays(AMdata$date)  
weekday<-c("星期一","星期二","星期三","星期四","星期五")  
weekend<-c("星期六","星期日")  
AMweekday<-AMdata[AMdata$weekdays %in% weekday,]  
AMweekend<-AMdata[AMdata$weekdays %in% weekend,]  
weekdayaver<-ddply(AMweekday,.(interval),function(x)mean(x$steps,na.rm = TRUE))  
colnames(weekdayaver)[2]<-"aversteps"  
weekendaver<-ddply(AMweekend,.(interval),function(x)mean(x$steps,na.rm = TRUE))  
colnames(weekendaver)[2]<-"aversteps"  
ggplot(weekdayaver,aes(x=interval,y=aversteps))+geom\_line()+ggtitle("WEEKDAY")



ggplot(weekendaver,aes(x=interval,y=aversteps))+geom\_line()+ggtitle("WEEKEND")

