PA1\_resub20180901

Harold Pollack

9/2/2018

This R markdown file explores an activity dataset. It compares activity on the weekdays and weekends. It imputes values, and gives an opportunity to use GitHub. This is a resubmission.

First I load libraries

Now I read read and processe activity.csv to read the activity dataset. I also construct a logged version of steps, which I will use below.

activity <- read.csv("/Users/haroldpollack/Documents/coursera\_datascience3/activity.csv")  
activity$lsteps<-log(1+activity$steps)  
activity$day <-weekdays(as.Date(activity$date))  
table(activity$day)

##   
## Friday Monday Saturday Sunday Thursday Tuesday Wednesday   
## 2592 2592 2304 2304 2592 2592 2592

str(activity)

## 'data.frame': 17568 obs. of 5 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 ...  
## $ lsteps : num NA NA NA NA NA NA NA NA NA NA ...  
## $ day : chr "Monday" "Monday" "Monday" "Monday" ...

summary(activity)

## steps date interval lsteps   
## Min. : 0.00 2012-10-01: 288 Min. : 0.0 Min. :0.000   
## 1st Qu.: 0.00 2012-10-02: 288 1st Qu.: 588.8 1st Qu.:0.000   
## Median : 0.00 2012-10-03: 288 Median :1177.5 Median :0.000   
## Mean : 37.38 2012-10-04: 288 Mean :1177.5 Mean :1.155   
## 3rd Qu.: 12.00 2012-10-05: 288 3rd Qu.:1766.2 3rd Qu.:2.565   
## Max. :806.00 2012-10-06: 288 Max. :2355.0 Max. :6.693   
## NA's :2304 (Other) :15840 NA's :2304   
## day   
## Length:17568   
## Class :character   
## Mode :character   
##   
##   
##   
##

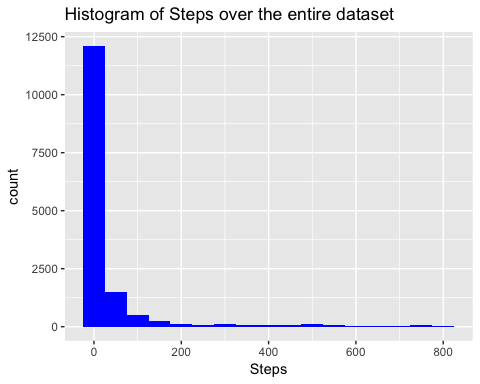
step\_mean <- mean(activity$steps, na.rm = TRUE)  
step\_median <- median(activity$steps, na.rm = TRUE)

Now the mean number of steps is 37.3825996.

The median number of steps is 0.

## Now let’s show some graphs of the step distribution

## Warning: Removed 2304 rows containing non-finite values (stat\_bin).



## Find which days are weekends

charday <- as.character(activity$day)  
table(charday)

## charday  
## Friday Monday Saturday Sunday Thursday Tuesday Wednesday   
## 2592 2592 2304 2304 2592 2592 2592

dSunday<-as.numeric((activity$day=="Sunday"))  
dMonday<-as.numeric((activity$day=="Monday"))  
dTuesday<-as.numeric((activity$day=="Tuesday"))  
dWednesday<-as.numeric((activity$day=="Wednesday"))  
dThursday<-as.numeric((activity$day=="Thursday"))  
dFriday<-as.numeric((activity$day=="Friday"))  
dSaturday<-as.numeric((activity$day=="Saturday"))  
dweekend<-dSunday+dSaturday  
activity$dweekend <- dweekend  
table(dweekend)

## dweekend  
## 0 1   
## 12960 4608

Now we have to handle missing values. I replaced with nonnegative predicted values based on a simple regression with day dummies and interval dummies. I then forced the values to be non-negative. A slightly better model would employ censored regression or something like that.

#  
# regression model with non-interacted day and interval dummies  
#  
model\_steps <- lm(steps ~ as.factor(day)+as.factor(interval), data=activity)  
# summary(model\_steps)  
activity$missing\_steps <- as.numeric(is.na(activity$steps))  
table(activity$missing\_steps)

##   
## 0 1   
## 15264 2304

n\_missing <- sum(activity$missing\_steps)  
# n\_missing  
activity$pred\_steps=predict(model\_steps,activity)  
summary(activity$pred\_steps)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -8.866 6.308 33.237 37.574 53.762 212.313

str(activity$pred\_steps)

## num [1:17568] -1.03 -2.41 -2.62 -2.6 -2.67 ...

activity$imputed\_steps<-activity$steps  
summary(activity$imputed\_steps)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.00 0.00 0.00 37.38 12.00 806.00 2304

str(activity$imputed\_steps)

## int [1:17568] NA NA NA NA NA NA NA NA NA NA ...

#  
# loop through each observation  
#  
for(i in 1:nrow(activity))  
{  
 activity$imputed\_steps[i]<- activity$steps[i]  
 if(activity$missing\_steps[i] == 1)  
 {  
 #   
 # Note that I want predicted values to be nonnegative. Which I do below.  
 #  
 activity$imputed\_steps[i] <- max(0,activity$pred\_steps[i])  
 }  
}  
summary(activity$imputed\_steps\_missing)

## Length Class Mode   
## 0 NULL NULL

str(activity$imputed\_steps\_missing)

## NULL

summary(activity$steps)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.00 0.00 0.00 37.38 12.00 806.00 2304

summary(activity$imputed\_steps)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 0.00 0.00 37.63 27.27 806.00

summary(activity$missing\_steps)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 0.0000 0.1311 0.0000 1.0000

str(activity$steps)

## int [1:17568] NA NA NA NA NA NA NA NA NA NA ...

str(activity$imputed\_steps)

## num [1:17568] 0 0 0 0 0 0 0 0 0 0 ...

str(activity$missing\_steps)

## num [1:17568] 1 1 1 1 1 1 1 1 1 1 ...

imputed\_steps <- activity$imputed\_steps  
activity\_date <- activity$date  
interval <-activity$interval

Note that there are `r n\_missing’ missing steps. Now deal with weekend stuff in a clunky way by defining two variables.

#  
# create weekend and weekday imputed step stuff. Loop through all the data and  
# fill in the weekend and weekday vales.  
#  
for(i in 1:nrow(activity))  
{  
 if(activity$dweekend[i] == 1)  
 {  
 #   
 # Note that I want predicted values to be nonnegative. Which I do below.  
 #  
 activity$weekend\_steps[i] <- activity$imputed\_steps[i]  
 activity$weekday\_steps[i] <- NA  
 activity$day\_class\_factor <- "Weekend"  
 }  
 if(activity$dweekend[i] == 0)  
 {  
 #   
 # Note that I want predicted values to be nonnegative. Which I do below.  
 #  
 activity$weekday\_steps[i] <- activity$imputed\_steps[i]  
 activity$weekend\_steps[i] <- NA  
 activity$day\_class\_factor <- "Weekday"  
 }  
}  
summary(activity$weekday\_steps)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.00 0.00 0.00 35.69 24.00 806.00 4608

#  
#  
#  
weekend\_activity <- subset (activity,dweekend==1)  
str(weekend\_activity)

## 'data.frame': 4608 obs. of 12 variables:  
## $ steps : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ date : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 6 6 6 6 6 6 6 6 6 6 ...  
## $ interval : int 0 5 10 15 20 25 30 35 40 45 ...  
## $ lsteps : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ day : chr "Saturday" "Saturday" "Saturday" "Saturday" ...  
## $ dweekend : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ missing\_steps : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ pred\_steps : num 7.86 6.48 6.28 6.29 6.22 ...  
## $ imputed\_steps : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ weekday\_steps : num NA NA NA NA NA NA NA NA NA NA ...  
## $ weekend\_steps : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ day\_class\_factor: chr "Weekday" "Weekday" "Weekday" "Weekday" ...

summary(weekend\_activity)

## steps date interval lsteps   
## Min. : 0.00 2012-10-06: 288 Min. : 0.0 Min. :0.000   
## 1st Qu.: 0.00 2012-10-07: 288 1st Qu.: 588.8 1st Qu.:0.000   
## Median : 0.00 2012-10-13: 288 Median :1177.5 Median :0.000   
## Mean : 43.08 2012-10-14: 288 Mean :1177.5 Mean :1.343   
## 3rd Qu.: 24.00 2012-10-20: 288 3rd Qu.:1766.2 3rd Qu.:3.219   
## Max. :785.00 2012-10-21: 288 Max. :2355.0 Max. :6.667   
## NA's :576 (Other) :2880 NA's :576   
## day dweekend missing\_steps pred\_steps   
## Length:4608 Min. :1 Min. :0.000 Min. : 5.248   
## Class :character 1st Qu.:1 1st Qu.:0.000 1st Qu.: 8.182   
## Mode :character Median :1 Median :0.000 Median : 39.809   
## Mean :1 Mean :0.125 Mean : 43.078   
## 3rd Qu.:1 3rd Qu.:0.000 3rd Qu.: 58.654   
## Max. :1 Max. :1.000 Max. :212.313   
##   
## imputed\_steps weekday\_steps weekend\_steps day\_class\_factor   
## Min. : 0.00 Min. : NA Min. : 0.00 Length:4608   
## 1st Qu.: 0.00 1st Qu.: NA 1st Qu.: 0.00 Class :character   
## Median : 0.00 Median : NA Median : 0.00 Mode :character   
## Mean : 43.08 Mean :NaN Mean : 43.08   
## 3rd Qu.: 37.57 3rd Qu.: NA 3rd Qu.: 37.57   
## Max. :785.00 Max. : NA Max. :785.00   
## NA's :4608

weekday\_activity <- subset (activity,dweekend==0)  
str(weekday\_activity)

## 'data.frame': 12960 obs. of 12 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 ...  
## $ lsteps : num NA NA NA NA NA NA NA NA NA NA ...  
## $ day : chr "Monday" "Monday" "Monday" "Monday" ...  
## $ dweekend : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ missing\_steps : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ pred\_steps : num -1.03 -2.41 -2.62 -2.6 -2.67 ...  
## $ imputed\_steps : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ weekday\_steps : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ weekend\_steps : num NA NA NA NA NA NA NA NA NA NA ...  
## $ day\_class\_factor: chr "Weekday" "Weekday" "Weekday" "Weekday" ...

summary(weekday\_activity)

## steps date interval lsteps   
## Min. : 0.00 2012-10-01: 288 Min. : 0.0 Min. :0.000   
## 1st Qu.: 0.00 2012-10-02: 288 1st Qu.: 588.8 1st Qu.:0.000   
## Median : 0.00 2012-10-03: 288 Median :1177.5 Median :0.000   
## Mean : 35.34 2012-10-04: 288 Mean :1177.5 Mean :1.088   
## 3rd Qu.: 8.00 2012-10-05: 288 3rd Qu.:1766.2 3rd Qu.:2.197   
## Max. :806.00 2012-10-08: 288 Max. :2355.0 Max. :6.693   
## NA's :1728 (Other) :11232 NA's :1728   
## day dweekend missing\_steps pred\_steps   
## Length:12960 Min. :0 Min. :0.0000 Min. : -8.866   
## Class :character 1st Qu.:0 1st Qu.:0.0000 1st Qu.: 5.533   
## Mode :character Median :0 Median :0.0000 Median : 31.654   
## Mean :0 Mean :0.1333 Mean : 35.616   
## 3rd Qu.:0 3rd Qu.:0.0000 3rd Qu.: 51.726   
## Max. :0 Max. :1.0000 Max. :211.703   
##   
## imputed\_steps weekday\_steps weekend\_steps day\_class\_factor   
## Min. : 0.00 Min. : 0.00 Min. : NA Length:12960   
## 1st Qu.: 0.00 1st Qu.: 0.00 1st Qu.: NA Class :character   
## Median : 0.00 Median : 0.00 Median : NA Mode :character   
## Mean : 35.69 Mean : 35.69 Mean :NaN   
## 3rd Qu.: 24.00 3rd Qu.: 24.00 3rd Qu.: NA   
## Max. :806.00 Max. :806.00 Max. : NA   
## NA's :12960

Set up a weekend factor

activity$weekend\_factor <- as.factor(activity$dweekend)

This code is to be ignored. Left here as a template for me for future use

#xyplot(week\_activity$imputed\_steps ~ week\_activity$interval | factor(as.factor(week\_activity$dweekend),type="l", xlab = "Interval",ylab = "Steps"))  
# xyplot(week\_activity$imputed\_steps ~ week\_activity$interval , xlab = "Interval",ylab = "Steps")

Now deal with daily averages using aggregate.

#  
# First compute average by day  
#  
day\_imputed\_steps <- aggregate(imputed\_steps,list(activity\_day= activity\_date),sum)  
day\_imputed\_steps$daily\_steps <- day\_imputed\_steps$x  
str(day\_imputed\_steps)

## 'data.frame': 61 obs. of 3 variables:  
## $ activity\_day: Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ x : num 10130 126 11352 12116 13294 ...  
## $ daily\_steps : num 10130 126 11352 12116 13294 ...

summary(day\_imputed\_steps$daily\_steps)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 41 8918 11015 10838 12811 21194

#  
# now average intervals  
#  
interval\_imputed\_steps <- aggregate(imputed\_steps,list(activity\_interval= interval),mean)  
interval\_imputed\_steps$interval\_steps\_imputed <- interval\_imputed\_steps$x  
str(interval\_imputed\_steps)

## 'data.frame': 288 obs. of 3 variables:  
## $ activity\_interval : int 0 5 10 15 20 25 30 35 40 45 ...  
## $ x : num 2.059 0.749 0.552 0.57 0.498 ...  
## $ interval\_steps\_imputed: num 2.059 0.749 0.552 0.57 0.498 ...

summary(interval\_imputed\_steps)

## activity\_interval x interval\_steps\_imputed  
## Min. : 0.0 Min. : 0.4265 Min. : 0.4265   
## 1st Qu.: 588.8 1st Qu.: 2.7901 1st Qu.: 2.7901   
## Median :1177.5 Median : 34.3043 Median : 34.3043   
## Mean :1177.5 Mean : 37.6306 Mean : 37.6306   
## 3rd Qu.:1766.2 3rd Qu.: 53.0260 3rd Qu.: 53.0260   
## Max. :2355.0 Max. :206.3609 Max. :206.3609

#  
# day\_imputed\_steps

Now show weekend and weekday separately

interval\_imputed\_weekend\_steps <- aggregate(activity$weekend\_steps,list(activity\_interval= interval),mean)  
interval\_imputed\_weekend\_steps <- interval\_imputed\_weekend\_steps$x  
str(interval\_imputed\_weekend\_steps)

## num [1:288] NA NA NA NA NA NA NA NA NA NA ...

summary(interval\_imputed\_weekend\_steps)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## NA NA NA NaN NA NA 288

Now find the interval with the largest average number of (imputed) steps, and the corresponding number of steps.

One notice. This is a resubmission. I noticed during peer grading that I had made a careless error in my original code. The maximum answer was correct, but my answer of“835” had the benefit of seeing the correct code.

#  
# find the maximum imputed steps  
#  
max\_interval <- interval\_imputed\_steps[which.max(interval\_imputed\_steps$interval\_steps\_imputed),1]   
max\_interval

## [1] 835

#   
# maximum value of steps  
#  
max(interval\_imputed\_steps$interval\_steps\_imputed)

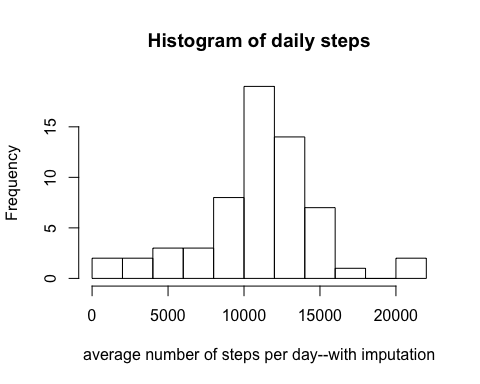
## [1] 206.3609

interval\_imputed\_steps[which.max(interval\_imputed\_steps$interval\_steps\_imputed),3]

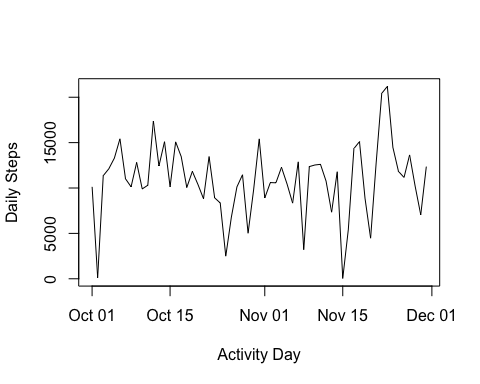
## [1] 206.3609

## Now show some graphs

## log of steps  
#  
# I commented out the logs here  
#  
hist(day\_imputed\_steps$daily\_steps ,   
 breaks = 10,   
 main="Histogram of daily steps",   
 xlab = "average number of steps per day--with imputation",  
 )



#  
# time series by day  
#  
plot(as.Date(day\_imputed\_steps$activity\_day),day\_imputed\_steps$daily\_steps, type="l", xlab = "Activity Day",ylab = "Daily Steps")



# xyplot(as.Date(day\_imputed\_steps$activity\_day),day\_imputed\_steps$daily\_steps|day\_imputed\_steps$dweekend)  
day\_imputed\_steps$day\_of\_week <-weekdays(as.Date(day\_imputed\_steps$activity\_day))  
# ggplot(day\_imputed\_steps)+geom\_line(aes(x=activity\_day,y=daily\_steps))  
#  
# Time series by 5-minute interval  
#  
table(day\_imputed\_steps$day\_of\_week)

##   
## Friday Monday Saturday Sunday Thursday Tuesday Wednesday   
## 9 9 8 8 9 9 9

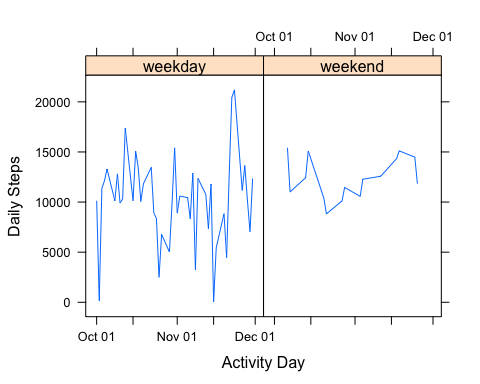
dSunday<-as.numeric((day\_imputed\_steps$day\_of\_week=="Sunday"))  
dSaturday<-as.numeric((day\_imputed\_steps$day\_of\_week=="Saturday"))  
day\_imputed\_steps$dweekend<-as.numeric(dSunday+dSaturday)  
table(day\_imputed\_steps$dweekend)

##   
## 0 1   
## 45 16

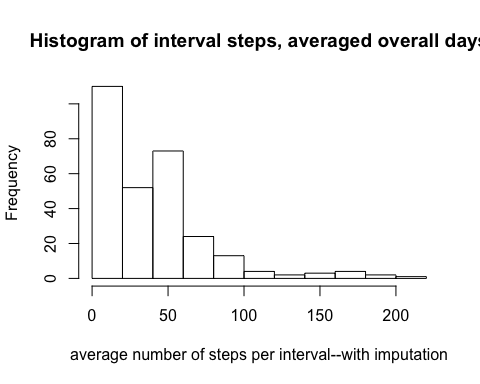
for(i in 1:nrow(day\_imputed\_steps))  
{  
 day\_imputed\_steps$day\_class[i]<- "weekday"  
 if(day\_imputed\_steps$dweekend[i] == 1)  
 {  
 #   
 # Note that I want predicted values to be nonnegative. Which I do below.  
 #  
 day\_imputed\_steps$day\_class[i]<- "weekend"  
 }  
}  
day\_imputed\_steps$day\_class<- as.factor(day\_imputed\_steps$day\_class)  
table(day\_imputed\_steps$day\_class)

##   
## weekday weekend   
## 45 16

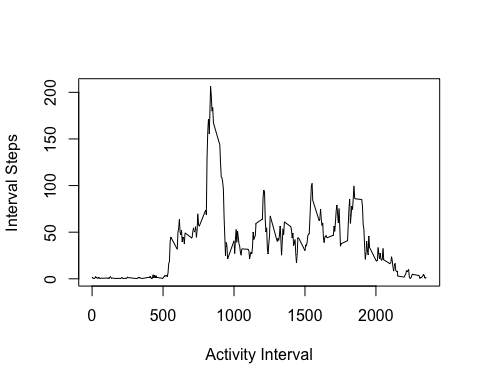
xyplot(day\_imputed\_steps$daily\_steps ~ as.Date(day\_imputed\_steps$activity\_day) | factor(day\_imputed\_steps$day\_class),type="l", xlab = "Activity Day",ylab = "Daily Steps")



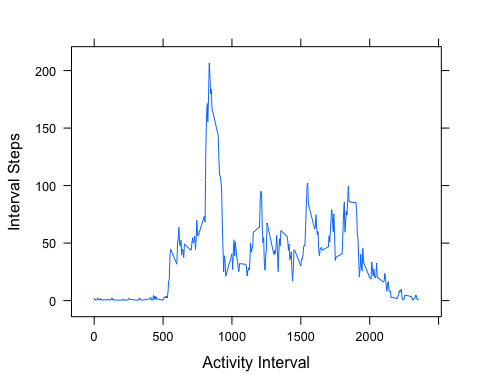
hist(interval\_imputed\_steps$interval\_steps\_imputed,   
 breaks = 10,   
 main="Histogram of interval steps, averaged overall days",  
 xlab = "average number of steps per interval--with imputation",  
 )



plot(interval\_imputed\_steps$activity\_interval,interval\_imputed\_steps$interval\_steps\_imputed, type="l",xlab = "Activity Interval",ylab = "Interval Steps")



xyplot(interval\_imputed\_steps$interval\_steps\_imputed ~ interval\_imputed\_steps$activity\_interval, type="l",xlab = "Activity Interval",ylab = "Interval Steps")



Now weekend and weekday intervals

#  
# now average intervals  
#  
weekend\_interval\_imputed\_steps <- aggregate(weekend\_activity$imputed\_steps,list(weekend\_activity\_interval= weekend\_activity$interval),mean)  
weekend\_interval\_imputed\_steps$interval\_steps\_imputed <- weekend\_interval\_imputed\_steps$x  
#  
# remove x  
#  
weekend\_interval\_imputed\_steps$x <- NULL  
#  
# descriptives on weekend   
#  
str(weekend\_interval\_imputed\_steps)

## 'data.frame': 288 obs. of 2 variables:  
## $ weekend\_activity\_interval: int 0 5 10 15 20 25 30 35 40 45 ...  
## $ interval\_steps\_imputed : num 0.927 0.754 0.728 0.731 0.721 ...

summary(weekend\_interval\_imputed\_steps)

## weekend\_activity\_interval interval\_steps\_imputed  
## Min. : 0.0 Min. : 0.712   
## 1st Qu.: 588.8 1st Qu.: 1.953   
## Median :1177.5 Median : 33.052   
## Mean :1177.5 Mean : 43.078   
## 3rd Qu.:1766.2 3rd Qu.: 75.366   
## Max. :2355.0 Max. :167.351

Now weekday code

#  
# now average intervals  
#  
weekday\_interval\_imputed\_steps <- aggregate(weekday\_activity$imputed\_steps,list(weekday\_activity\_interval= weekday\_activity$interval),mean)  
weekday\_interval\_imputed\_steps$interval\_steps\_imputed <- weekday\_interval\_imputed\_steps$x  
str(weekday\_interval\_imputed\_steps)

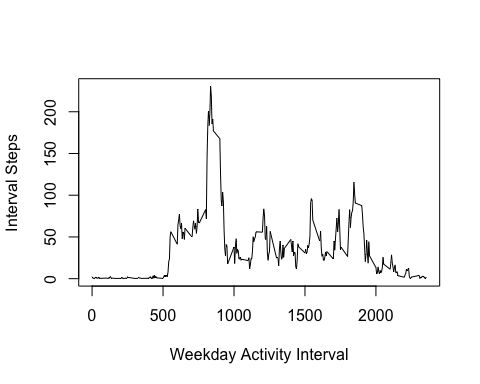
## 'data.frame': 288 obs. of 3 variables:  
## $ weekday\_activity\_interval: int 0 5 10 15 20 25 30 35 40 45 ...  
## $ x : num 2.462 0.748 0.489 0.513 0.419 ...  
## $ interval\_steps\_imputed : num 2.462 0.748 0.489 0.513 0.419 ...

summary(weekday\_interval\_imputed\_steps)

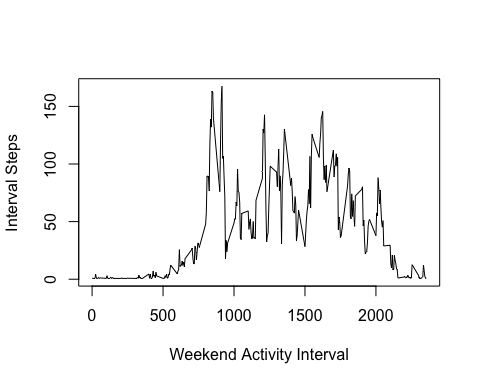
## weekday\_activity\_interval x interval\_steps\_imputed  
## Min. : 0.0 Min. : 0.325 Min. : 0.325   
## 1st Qu.: 588.8 1st Qu.: 2.415 1st Qu.: 2.415   
## Median :1177.5 Median : 25.809 Median : 25.809   
## Mean :1177.5 Mean : 35.694 Mean : 35.694   
## 3rd Qu.:1766.2 3rd Qu.: 50.860 3rd Qu.: 50.860   
## Max. :2355.0 Max. :230.384 Max. :230.384

## Now show the weeken/weekday graphs and the multi-panel

p1 <- plot(weekday\_interval\_imputed\_steps$weekday\_activity\_interval,weekday\_interval\_imputed\_steps$interval\_steps\_imputed, type="l",xlab = "Weekday Activity Interval",ylab = "Interval Steps")



p2 <- plot(weekend\_interval\_imputed\_steps$weekend\_activity\_interval,weekend\_interval\_imputed\_steps$interval\_steps\_imputed, type="l",xlab = "Weekend Activity Interval",ylab = "Interval Steps")



q1 <- qplot(weekday\_interval\_imputed\_steps$weekday\_activity\_interval,weekday\_interval\_imputed\_steps$interval\_steps\_imputed,xlab = "Weekday Activity Interval",ylab = "Interval Steps", geom=c("point", "line"))  
q2 <- qplot(weekend\_interval\_imputed\_steps$weekend\_activity\_interval,weekend\_interval\_imputed\_steps$interval\_steps\_imputed,xlab = "Weekend Activity Interval",ylab = "Interval Steps", geom=c("point", "line"))  
  
grid.arrange (q1, q2, nrow = 1)

