Week 2 Homework

Kyle

Monday, April 24, 2017

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

Read in the dataset Activity monitoring data as a csv file.

library(plyr)  
library(ggplot2)  
stepData <- read.csv("C:/Users/1405249584A/Documents/R/CourseraData/activity.csv")  
stepData$date <- as.Date(stepData$date,"%Y-%m-%d")

Find the mean and median:

mean(stepData$steps,na.rm=TRUE)

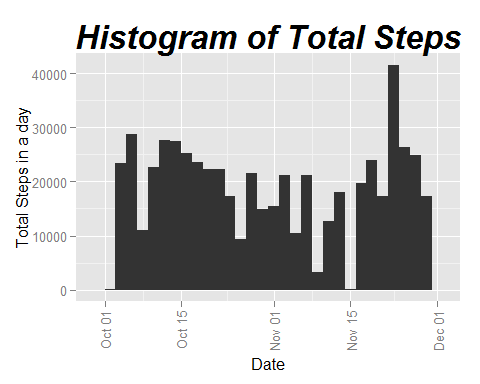
## [1] 37.3826

median(stepData$steps,na.rm=TRUE)

## [1] 0

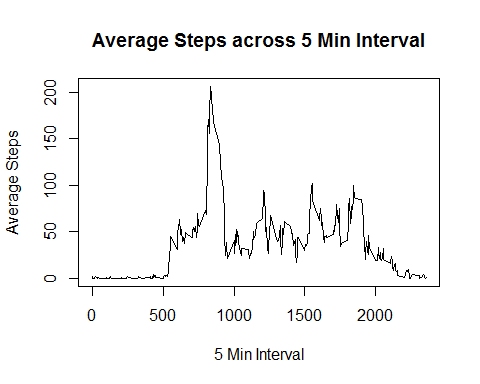
Plot a histogram of the total number of steps taken each day:

hist <- ggplot(stepData)+  
 geom\_histogram(aes(x=date,weight=steps))+  
 ggtitle("Histogram of Total Steps")+  
 ylab("Total Steps in a day")+  
 xlab("Date")+  
 theme(plot.title = element\_text(color="black", size=25, face="bold.italic"),  
 axis.text.x=element\_text(angle=90,vjust=.5,hjust=1))  
hist



Time series plot of the average number of steps taken. The first graph is to verify the line (second) graph:

average <- ddply(stepData,~date,summarise,avg=mean(steps,na.rm=TRUE))  
gplotT <- ggplot(average)+  
 geom\_histogram(aes(x=date,weight=avg))+  
 ggtitle("Histogram of Average Steps Each Day")+  
 ylab("Average Steps in a day")+  
 xlab("Date")+  
 theme(plot.title = element\_text(color="black", size=25, face="bold.italic"),  
 axis.text.x=element\_text(angle=90,vjust=.5,hjust=1))  
  
intervals <- ddply(stepData,~interval,summarise,avg=mean(steps,na.rm=TRUE))  
plot(intervals$interval,intervals$avg,type="l",  
 main="Average Steps across 5 Min Interval", ylab="Average Steps", xlab="5 Min Interval")



5-minute interval that, on average, contains the maximum number of steps:

intervals[intervals$avg==max(intervals$avg),]

## interval avg  
## 104 835 206.1698

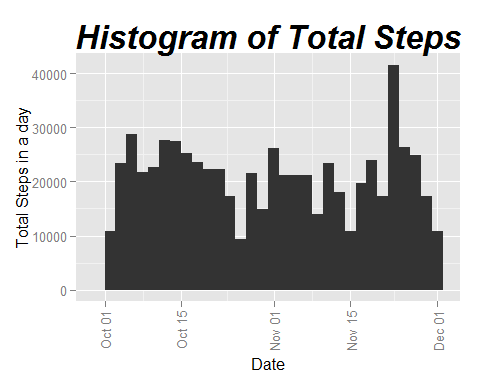
Code to describe and show a strategy for imputing missing data:

Keeping it simple and using the mean of the specific interval to impute the missing values.

avgDay <- ddply(stepData,~date+interval,avg=mean(steps,na.rm=TRUE))  
dailyAvg <- aggregate(avgDay$steps,format(avgDay['date'],'%d'),mean)  
stepData2 <- stepData  
stepData2[is.na(stepData2$steps),'steps'] <- intervals$avg

Histogram of the new dataset. Also find the new mean and median:

hist2 <- ggplot(stepData2)+  
 geom\_histogram(aes(x=date,weight=steps))+  
 ggtitle("Histogram of Total Steps")+  
 ylab("Total Steps in a day")+  
 xlab("Date")+  
 theme(plot.title = element\_text(color="black", size=25, face="bold.italic"),  
 axis.text.x=element\_text(angle=90,vjust=.5,hjust=1))  
hist2



mean(stepData2$steps)

## [1] 37.3826

median(stepData2$steps)

## [1] 0

Compare the old average steps with the new average steps with the imputed data.

avgDay2 <- ddply(stepData2,~date+interval,avg=mean(steps))  
View(cbind(avgDay2,avgDay))

The median and mean from the old dataset and new dataset are the same.

mean(stepData2$steps)

## [1] 37.3826

median(stepData2$steps)

## [1] 0

mean(stepData$steps,na.rm=TRUE)

## [1] 37.3826

median(stepData$steps,na.rm=TRUE)

## [1] 0

Add new column indicating if a date is a weekday or weekend.

weekdays <- c("Monday","Tuesday","Wednesday","Thursday","Friday")  
weekends <- c("Saturday","Sunday")  
stepData2$day <- weekdays(stepData2$date)  
stepData2[stepData2$day%in%weekdays,'day'] <- "weekday"  
stepData2[stepData2$day%in%weekends,'day'] <- "weekend"  
stepData2$day <- factor(stepData2$day)

Make a panel plot with a time series of average steps based on weekday or weekend.

avgDay3 <- ddply(stepData2,~interval+day,summarise,steps=as.double(format(mean(steps),option=4)))  
lineplot <- ggplot(avgDay3)+  
 geom\_line(aes(x=interval,y=steps,colour=day,group=day))+  
 ggtitle("Line Plot of Average Steps for Each Interval")+  
 facet\_grid(day~.)  
lineplot

