peer\_assessment

Student

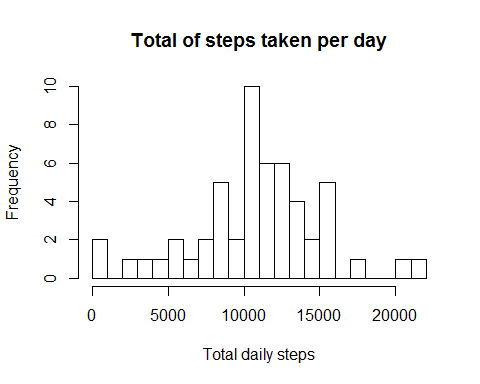
Friday, November 14, 2014

This is an R Markdown document. The first step is to read the file and create a histogram of the total number of steps taken each day #These steps show loading and initial processing of data library(knitr) setwd("~/R/R\_lectures/Data\_cousera/reproducible/) keep\_md = TRUE in html\_document() knit2html("PA1\_template.Rmd") browseURL("PA1\_template.html")

activity <- read.csv("~/R/R\_lectures/Data\_cousera/reproducible/activity.csv")  
activity$date <- as.Date(activity[,2], format="%m/%d/%Y")

# Ignoring the missing values a histogram of the total number of steps taken each day has been created

hist(tapply(activity$steps, activity$date, sum), xlab = "Total daily steps", breaks = 20,   
 main = "Total of steps taken per day")

 #mean and median total number of steps taken per day is calculated

total\_steps <- as.numeric(tapply(activity$steps, activity$date, sum))  
step\_mean <- mean(total\_steps, na.rm = TRUE)  
step\_median <- median(total\_steps, na.rm = TRUE)  
step\_mean

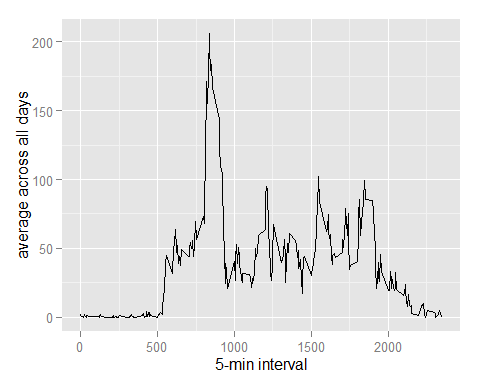
## [1] 10766

step\_median

## [1] 10765

# Now, a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis) is created here

library(ggplot2)  
average <- aggregate(x=list(steps=activity$steps), by=list(interval=activity$interval), FUN=mean, na.rm=TRUE)  
ggplot(data=average, aes(x=interval, y=steps))+geom\_line()+xlab("5-min interval")+ ylab("average across all days")



max(average$steps, na.rm=TRUE)

## [1] 206.2

# Number of days/intervals where there are missing values (coded as NA). The mean for the 5 min interval is used for filling in the missing values. The sum of the missing rows is calculated here.The dataset activity is split. Omitting the rows containing the dataset, a new dataset(nona\_activity) is created.Another dataset with na containing rows is created(na\_activity). the na containing dataset is recoded with the interval averages.Both the dataset are merged together as "imputed" dataset

sum(is.na(activity$steps))

## [1] 2304

nona\_activity<-na.omit(activity)  
na\_activity<-activity[is.na(activity[]),]  
na\_activity[,1]<-sapply(na\_activity[,3],function(x) round(average[average$interval==x,2]))  
imputed<-rbind(nona\_activity,na\_activity)  
imputed\_activity<- tapply(imputed$steps,imputed$date,sum)

# Mean, median is calculated for the imputed datasets

step\_mean

## [1] 10766

step\_median

## [1] 10765

mean(imputed\_activity)

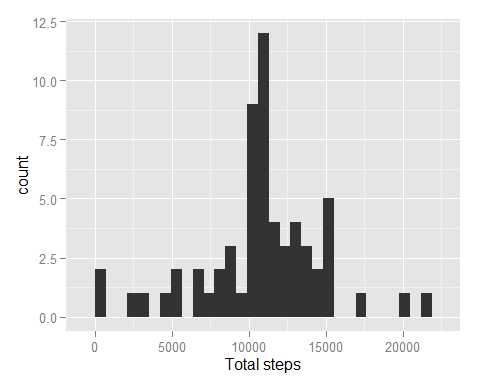
## [1] 10766

median(imputed\_activity)

## [1] 10762

qplot(imputed\_activity, xlab='Total steps')

## stat\_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.

 #After imputation the mean and median of total steps did not change much!

# Created a new factor variable in the dataset with two levels - "weekday" and "weekend" indicating whether a given date is a weekday or weekend day.

# Made a panel plot containing a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis).

library(lubridate)  
imputed$day<- wday(imputed$date, label=TRUE)  
imputed$daytype<- imputed$day  
levels(imputed$daytype) <- list(  
 weekday = c("Mon", "Tues", "Wed", "Thurs", "Fri"),  
 weekend = c("Sun", "Sat"))  
library(dplyr)

##   
## Attaching package: 'dplyr'  
##   
## The following objects are masked from 'package:lubridate':  
##   
## intersect, setdiff, union  
##   
## The following object is masked from 'package:stats':  
##   
## filter  
##   
## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

activity.typeday<- imputed %>%  
 group\_by(daytype, interval) %>%   
 summarize(total.steps=sum(steps, na.rm=TRUE), average.steps=mean(steps, na.rm=TRUE))  
library(lattice)  
xyplot(average.steps~interval|daytype, data=activity.typeday, type='l', layout=(c(1,2)),  
 main="Average Daily Activity Pattern by Type of Day",  
 ylab="Average Number of Steps Taken per Interval", xlab="5-minute Time Interval")

