# Summary step data over the course of 2 months

### Packaging and data download

Before analysis, download key packages.

* Pacman expedites the downloading and installing packages process
* Plyr and dplyr are helpful in analyzing data frames
* Downloader expedites the downloading and unzipping data process
* Lubridate makes processing dates easier
* ggplot2 enables the creation of nicely formatted graphs
* gridExtra simplifies the development of panel plotting in ggplot

Then, download the data!

if (!require("pacman")) install.packages("pacman")

## Loading required package: pacman

p\_load(plyr, dplyr, downloader, lubridate, ggplot2, gridExtra)  
  
library(xtable)  
options(xtable.floating = FALSE)  
options(xtable.timestamp = "")  
  
download("http://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip", dest="./Activity monitoring data.zip", mode="wb")  
unzip("Activity monitoring data.zip")

### Initial data processing

The data comes in .csv format and needs very little processing for the first few questions. The first few questions do involve summary statistics (i.e., sum, mean, median), so the code below will create a dataset that contains these summary statistics grouped by day.

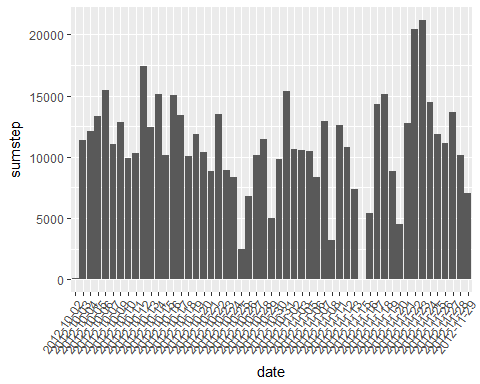
df <- read.csv("activity.csv", header=TRUE)  
sumstep\_day <- aggregate(steps ~ date, df, FUN=sum, drop=FALSE)  
meanstep\_day <- aggregate(steps ~ date, df, FUN=mean, drop=FALSE)  
medstep\_day <- aggregate(steps ~ date, df, FUN=median, drop=FALSE)  
summarystep\_day <- sumstep\_day %>% merge(meanstep\_day, by="date") %>%  
 merge(medstep\_day,by="date")  
names(summarystep\_day) <- c("date","sumstep","meanstep","medstep")

### QUESTION 1: What are the total number of steps taken each day?

Using the summary data frames created above, the script below creates quick histogram that demonstrates total steps across the different days in the dataset.

ggplot(data=summarystep\_day,aes(x=date,y=sumstep))+ geom\_histogram(stat="identity") +   
 theme(axis.text.x = element\_text(angle = 55, hjust = 1))

## Warning: Ignoring unknown parameters: binwidth, bins, pad



### QUESTION 2: What are the mean and median number of steps taken each day?

The table below illustrates the mean and median number of steps taken each day.

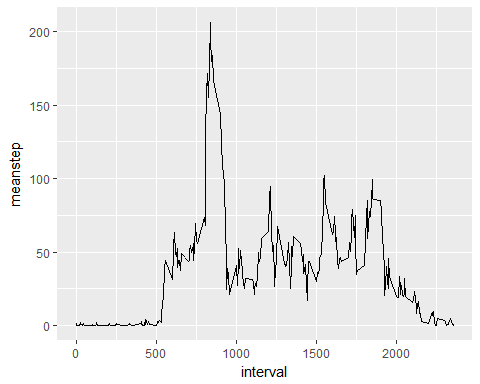
kable(summarystep\_day)

|  |  |  |  |
| --- | --- | --- | --- |
| date | sumstep | meanstep | medstep |
| 2012-10-02 | 126 | 0.4375000 | 0 |
| 2012-10-03 | 11352 | 39.4166667 | 0 |
| 2012-10-04 | 12116 | 42.0694444 | 0 |
| 2012-10-05 | 13294 | 46.1597222 | 0 |
| 2012-10-06 | 15420 | 53.5416667 | 0 |
| 2012-10-07 | 11015 | 38.2465278 | 0 |
| 2012-10-09 | 12811 | 44.4826389 | 0 |
| 2012-10-10 | 9900 | 34.3750000 | 0 |
| 2012-10-11 | 10304 | 35.7777778 | 0 |
| 2012-10-12 | 17382 | 60.3541667 | 0 |
| 2012-10-13 | 12426 | 43.1458333 | 0 |
| 2012-10-14 | 15098 | 52.4236111 | 0 |
| 2012-10-15 | 10139 | 35.2048611 | 0 |
| 2012-10-16 | 15084 | 52.3750000 | 0 |
| 2012-10-17 | 13452 | 46.7083333 | 0 |
| 2012-10-18 | 10056 | 34.9166667 | 0 |
| 2012-10-19 | 11829 | 41.0729167 | 0 |
| 2012-10-20 | 10395 | 36.0937500 | 0 |
| 2012-10-21 | 8821 | 30.6284722 | 0 |
| 2012-10-22 | 13460 | 46.7361111 | 0 |
| 2012-10-23 | 8918 | 30.9652778 | 0 |
| 2012-10-24 | 8355 | 29.0104167 | 0 |
| 2012-10-25 | 2492 | 8.6527778 | 0 |
| 2012-10-26 | 6778 | 23.5347222 | 0 |
| 2012-10-27 | 10119 | 35.1354167 | 0 |
| 2012-10-28 | 11458 | 39.7847222 | 0 |
| 2012-10-29 | 5018 | 17.4236111 | 0 |
| 2012-10-30 | 9819 | 34.0937500 | 0 |
| 2012-10-31 | 15414 | 53.5208333 | 0 |
| 2012-11-02 | 10600 | 36.8055556 | 0 |
| 2012-11-03 | 10571 | 36.7048611 | 0 |
| 2012-11-05 | 10439 | 36.2465278 | 0 |
| 2012-11-06 | 8334 | 28.9375000 | 0 |
| 2012-11-07 | 12883 | 44.7326389 | 0 |
| 2012-11-08 | 3219 | 11.1770833 | 0 |
| 2012-11-11 | 12608 | 43.7777778 | 0 |
| 2012-11-12 | 10765 | 37.3784722 | 0 |
| 2012-11-13 | 7336 | 25.4722222 | 0 |
| 2012-11-15 | 41 | 0.1423611 | 0 |
| 2012-11-16 | 5441 | 18.8923611 | 0 |
| 2012-11-17 | 14339 | 49.7881944 | 0 |
| 2012-11-18 | 15110 | 52.4652778 | 0 |
| 2012-11-19 | 8841 | 30.6979167 | 0 |
| 2012-11-20 | 4472 | 15.5277778 | 0 |
| 2012-11-21 | 12787 | 44.3993056 | 0 |
| 2012-11-22 | 20427 | 70.9270833 | 0 |
| 2012-11-23 | 21194 | 73.5902778 | 0 |
| 2012-11-24 | 14478 | 50.2708333 | 0 |
| 2012-11-25 | 11834 | 41.0902778 | 0 |
| 2012-11-26 | 11162 | 38.7569444 | 0 |
| 2012-11-27 | 13646 | 47.3819444 | 0 |
| 2012-11-28 | 10183 | 35.3576389 | 0 |
| 2012-11-29 | 7047 | 24.4687500 | 0 |

### QUESTION 3: How many steps are typically taken in different 5-minute intervals over the course of a normal day?

For the next chunk, I go back to the original data frame and aggregate using a mean function, by the 5-minute intervals. Then, I use ggplot to create a time-series line plot that tracks average steps over the course of a typical day.

meanstep\_int <- aggregate(steps ~ interval, df, FUN=mean, drop=FALSE)  
maxstep\_int <- aggregate(steps ~ interval, df, FUN=max, drop=FALSE)  
summarystep\_int <- meanstep\_int %>% merge(maxstep\_int,by="interval", drop=FALSE)  
names(summarystep\_int) <- c("interval","meanstep","maxstep")  
  
ggplot(data=summarystep\_int, aes(x=interval,y=meanstep)) + geom\_line()



### QUESTION 4: Which 5-minute interval,on average, contains the maximum number of steps?

Though this question can loosely be answered by referring to the graph above, I answer this question more directly with the table below.

tmp <- arrange(summarystep\_int,desc(meanstep))[1:6,1:2]  
kable(tmp)

|  |  |
| --- | --- |
| interval | meanstep |
| 835 | 206.1698 |
| 840 | 195.9245 |
| 850 | 183.3962 |
| 845 | 179.5660 |
| 830 | 177.3019 |
| 820 | 171.1509 |

## Impute missing data

In this code chunk I replace missing data with the mean from a given interval. This is assumed to be more appropriate than the mean for an entire day because steps vary significantly less on a per-interval basis than they do across an entire day.

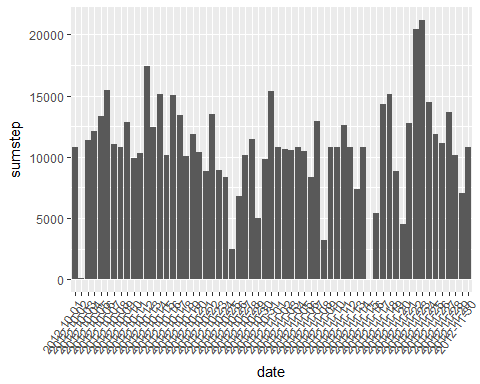
impute.mean <- function(x) replace(x, is.na(x), mean(x, na.rm = TRUE))  
df\_impute <- ddply(df, ~ interval, transform, steps = impute.mean(steps))

### QUESTION 5: After imputing missing values, how many steps are taken each day?

Having imputed the missing data, now let's run the total steps per day data again, creating an "imputed data" data frame. Using this data frame, I will then run a histogram in ggplot.

sumstep\_day\_imp <- aggregate(steps ~ date, df\_impute, FUN=sum, drop=FALSE)  
names(sumstep\_day\_imp) <- c("date", "sumstep")  
  
ggplot(data=sumstep\_day\_imp,aes(x=date,y=sumstep))+   
 geom\_histogram(stat="identity") +   
 theme(axis.text.x = element\_text(angle = 55, hjust = 1))

## Warning: Ignoring unknown parameters: binwidth, bins, pad



### QUESTION 6: How does the average number of steps per 5-minute interval change across weekdays vs. weekends?

To address this question, I will filter the above data set, breaking it down into a weekend and a weekday dataset. From there, I will calculate two line graphs and use gridExtra to plot them in a panel format.

meanstep\_day\_imp <- mutate(df\_impute,day=wday(date,label=TRUE))  
  
meanstep\_day\_imp\_wkend <- filter(meanstep\_day\_imp,day=="Sat" | day=="Sun")  
meanstep\_day\_imp\_wkend <- aggregate(steps ~ interval, meanstep\_day\_imp\_wkend, FUN=mean, drop=FALSE)  
  
meanstep\_day\_imp\_wkday <- filter(meanstep\_day\_imp,day=="Mon" | day=="Tues" |   
 day=="Wed" | day=="Thurs" | day=="Fri")  
meanstep\_day\_imp\_wkday <- aggregate(steps ~ interval, meanstep\_day\_imp\_wkday, FUN=mean, drop=FALSE)  
  
p1 <- ggplot(meanstep\_day\_imp\_wkend, aes(x=interval, y=steps)) + geom\_line() + ggtitle("Mean Weekend Steps") + ylim(0,250)  
p2 <- ggplot(meanstep\_day\_imp\_wkday, aes(x=interval,y=steps)) + geom\_line() + ggtitle("Mean Weekday Steps") + ylim(0,250)  
  
grid.arrange(p1, p2, ncol=2)

