Reproducible Research PA1

Faisal Sardar

Saturday, May 16, 2015

Loading and preprocessing the data

Show any code that is needed to

- Load the data (i.e. read.csv())
- Process/transform the data (if necessary) into a format suitable for your analysis

```
if(!file.exists("data")){
  dir.create("data")
}

if(!"Activity.zip" %in% dir("./data/")){
  URLFile<-"https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
  download.file(URLFile,destfile="./data/Activity.zip")
  unzip("./data/Activity.zip", files = NULL, list = FALSE, overwrite = TRUE,
        junkpaths = FALSE, exdir = "./data", unzip = "internal",
        setTimes = FALSE)
  dateDownloaded<-date()
} else {print("Already downloaded!")}


AMonitor<-data.table(read.csv("./data/activity.csv"))
head(AMonitor)
str(AMonitor)

#ignoring rows with na
AMonitor<-na.omit(AMonitor)
AMonitor <- AMonitor[, date := as.Date(date)]
setkey(AMonitor, date, interval)
head(AMonitor)
```

What is mean total number of steps taken per day?

For this part of the assignment, you can ignore the missing values in the dataset.

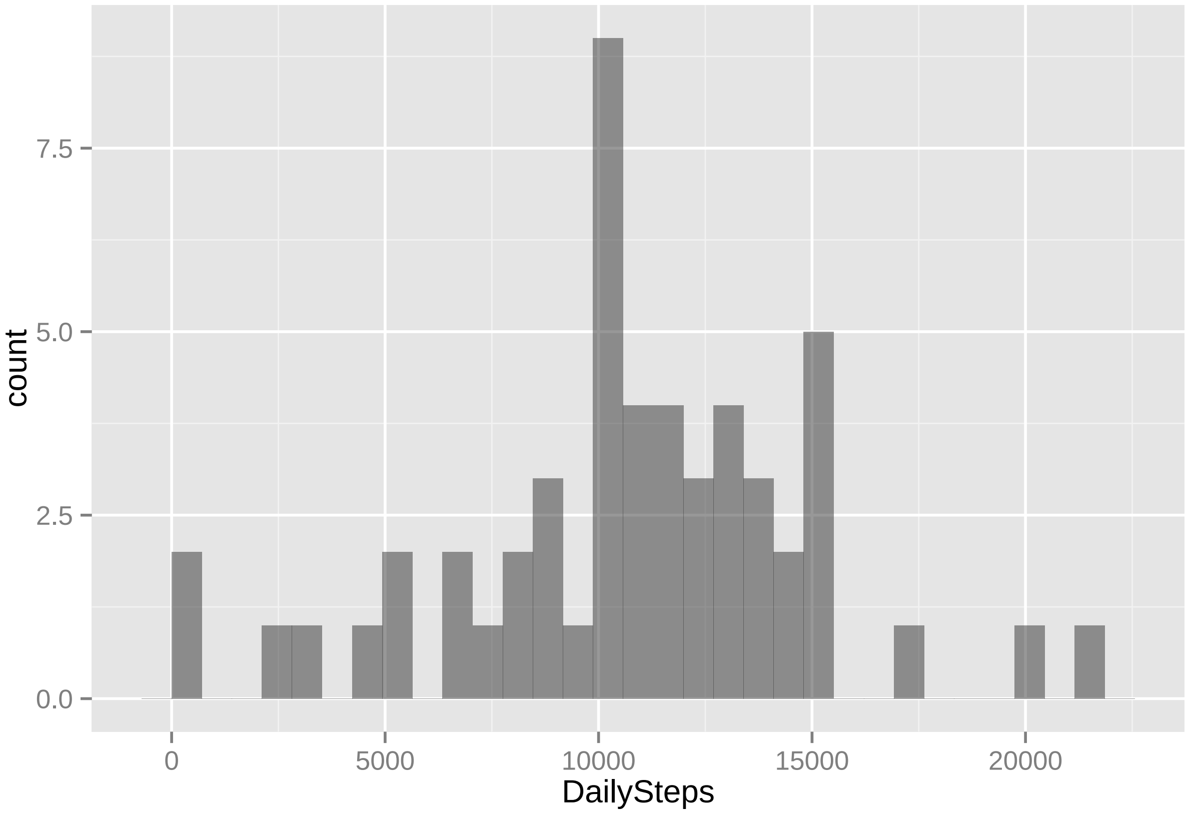
- Calculate the total number of steps taken per day

```
TotalDailySteps <- AMonitor[, list(DailySteps = sum(steps)), date]
TotalDailySteps
```

```
##           date DailySteps
##  1: 2012-10-02        126
##  2: 2012-10-03      11352
##  3: 2012-10-04      12116
##  4: 2012-10-05      13294
##  5: 2012-10-06      15420
##  6: 2012-10-07      11015
##  7: 2012-10-09      12811
##  8: 2012-10-10       9900
##  9: 2012-10-11      10304
## 10: 2012-10-12      17382
## 11: 2012-10-13      12426
## 12: 2012-10-14      15098
## 13: 2012-10-15      10139
## 14: 2012-10-16      15084
## 15: 2012-10-17      13452
## 16: 2012-10-18      10056
## 17: 2012-10-19      11829
## 18: 2012-10-20      10395
## 19: 2012-10-21       8821
## 20: 2012-10-22      13460
## 21: 2012-10-23       8918
## 22: 2012-10-24       8355
## 23: 2012-10-25       2492
## 24: 2012-10-26       6778
## 25: 2012-10-27      10119
## 26: 2012-10-28      11458
## 27: 2012-10-29       5018
## 28: 2012-10-30       9819
## 29: 2012-10-31      15414
## 30: 2012-11-02      10600
## 31: 2012-11-03      10571
## 32: 2012-11-05      10439
## 33: 2012-11-06       8334
## 34: 2012-11-07      12883
## 35: 2012-11-08       3219
## 36: 2012-11-11      12608
## 37: 2012-11-12      10765
## 38: 2012-11-13       7336
## 39: 2012-11-15         41
## 40: 2012-11-16       5441
## 41: 2012-11-17      14339
## 42: 2012-11-18      15110
## 43: 2012-11-19       8841
## 44: 2012-11-20       4472
## 45: 2012-11-21      12787
## 46: 2012-11-22      20427
## 47: 2012-11-23      21194
## 48: 2012-11-24      14478
## 49: 2012-11-25      11834
## 50: 2012-11-26      11162
## 51: 2012-11-27      13646
## 52: 2012-11-28      10183
## 53: 2012-11-29       7047
##           date DailySteps
```

- If you do not understand the difference between a histogram and a barplot, research the difference between them. Make a histogram of the total number of steps taken each day

```
ggplot(TotalDailySteps, aes(x=DailySteps)) +
  geom_histogram(alpha=.5)
```



- Calculate and report the mean and median of the total number of steps taken per day

```
mean(TotalDailySteps$DailySteps)
```

```
## [1] 10766.19
```

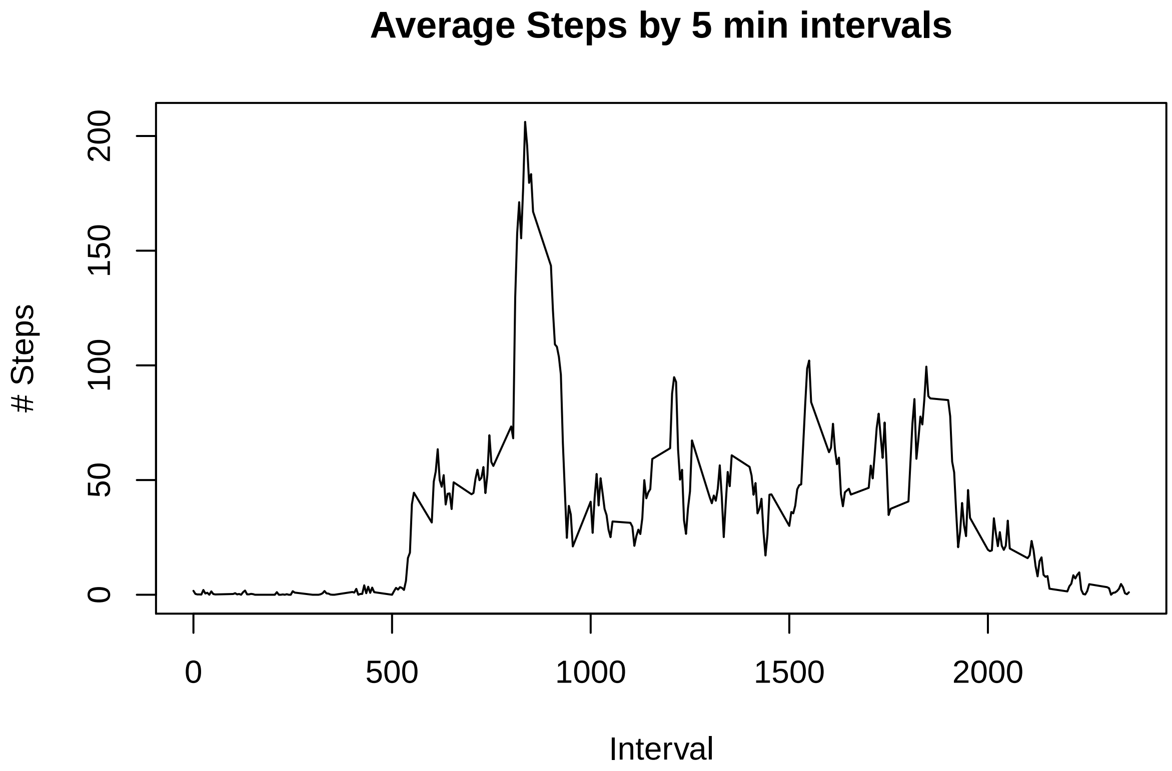
```
median(TotalDailySteps$DailySteps)
```

```
## [1] 10765
```

What is the average daily activity pattern?

- Make 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)

```
steps_by_interval <- aggregate(steps ~ interval, data=AMonitor, mean)
plot(steps_by_interval$interval,steps_by_interval$steps, type="l",
     xlab="Interval", ylab="# Steps",main="Average Steps by 5 min intervals")
```



- Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

```
MeanStepsPerInterval<-ddply(AMonitor, c("interval"),summarise,meansteps = mean(steps))
```

Max Interval:

```
MeanStepsPerInterval[which(MeanStepsPerInterval$meansteps
                     ==max(MeanStepsPerInterval$meansteps)), "interval"]
```

```
## [1] 835
```

Max Steps:

```
MeanStepsPerInterval[which(MeanStepsPerInterval$meansteps
                     ==max(MeanStepsPerInterval$meansteps)), "meansteps"]
```

```
## [1] 206.1698
```

Imputing missing values

Note that there are a number of days/intervals where there are missing values (coded as NA). The presence of missing days may introduce bias into some calculations or summaries of the data.

- Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

```
MissingValues<-data.table(read.csv("./data/activity.csv"))
CountMV <- sum(!complete.cases(MissingValues))
CountMV
```

```
## [1] 2304
```

Are there differences in activity patterns between weekdays and weekends?

For this part the weekdays() function may be of some help here. Use the dataset with the filled-in missing values for this part.

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

```
DOWLevels <- c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")
WDWELevels <- c("Weekend", "Weekday", "Weekday","Weekday","Weekday","Weekday","Weekday","Weekend")
DOWComp <- AMonitor[, DOW := factor(weekdays(date), levels=DOWLevels)]
DOWComp <- AMonitor[, WDWE := factor(WDWELevels[DOW])]
DOWComp[, .N, list(WDWE, DOW)]
WDWEIntervals <- DOWComp[, list(meanSteps = mean(steps)), list(WDWE, interval)]
```

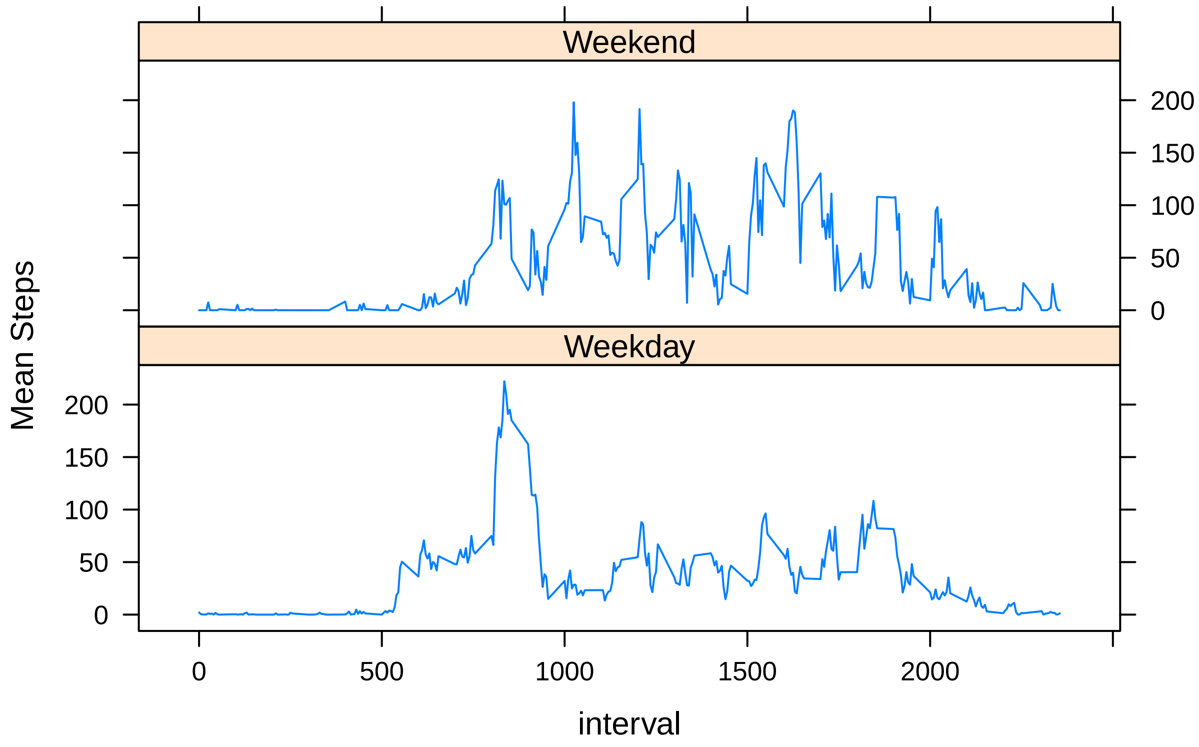
```
##    steps       date interval     DOW    WDWE
## 1:     0 2012-10-02        0 Tuesday Weekday
## 2:     0 2012-10-02        5 Tuesday Weekday
## 3:     0 2012-10-02       10 Tuesday Weekday
## 4:     0 2012-10-02       15 Tuesday Weekday
## 5:     0 2012-10-02       20 Tuesday Weekday
## 6:     0 2012-10-02       25 Tuesday Weekday
## 7:     0 2012-10-02       30 Tuesday Weekday
```

```
##       WDWE interval  meanSteps
## 1: Weekday        0 1.97826087
## 2: Weekday        5 0.39130435
## 3: Weekday       10 0.15217391
## 4: Weekday       15 0.17391304
## 5: Weekday       20 0.08695652
## 6: Weekday       25 1.28260870
## 7: Weekday       30 0.60869565
```

- Make 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). The plot should look something like the following, which was creating using simulated data:

```
xyplot(meanSteps ~ interval | WDWE,
       data=WDWEIntervals, type="l", layout=c(1,2),
       ylab = "Mean Steps", main="Weekday vs. Weekend - Mean Steps Per 5-Minute Interval")
```

Weekday vs. Weekend − Mean Steps Per 5−Minute Interval


Weekend
Weekday
Mean Steps
interval