Reproducible Research: Course Project 1

Nicolás Rivera Garzón

9/11/2020

R Markdown

Loading and preprocessing the data

1. Load the data.

```
getwd()
```

```
## [1] "C:/Users/Nicolás Rivera/OneDrive/Documentos/Data Science Johns Hopkins University/Reproducible R
```

```
setwd("C:/Users/Nicolás Rivera/OneDrive/Documentos/Data Science Johns Hopkins University/Reproducible Re
activity<-read.csv("activity.csv")
```

2. Process/transform the data (if necessary) into a format suitable for your analysis. Explore

```
dim(activity)
```

```
## [1] 17568     3
```

```
str(activity)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : chr  "2012-10-01" "2012-10-01" "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

```
head(activity)
```

```
##   steps       date interval
## 1    NA 2012-10-01        0
## 2    NA 2012-10-01        5
## 3    NA 2012-10-01       10
## 4    NA 2012-10-01       15
## 5    NA 2012-10-01       20
## 6    NA 2012-10-01       25
```

```
names(activity)
```

```
## [1] "steps"    "date"     "interval"
```

```
summary(activity)
```

```
##      steps            date              interval
##  Min.   :  0.00   Length:17568       Min.   :   0.0
##  1st Qu.:  0.00   Class :character   1st Qu.: 588.8
##  Median :  0.00   Mode  :character   Median :1177.5
##  Mean   : 37.38                      Mean   :1177.5
##  3rd Qu.: 12.00                      3rd Qu.:1766.2
```

```
##  Max.   :806.00                     Max.   :2355.0
##  NA's   :2304
```

What is mean total number of steps taken per day?

1. Calculate the total number of steps taken per day.

```
StepsDaily <- aggregate(activity$steps, list(activity$date), FUN=sum)
colnames(StepsDaily) <- c("Date", "Steps")
StepsDaily
```

```
##          Date Steps
## 1  2012-10-01    NA
## 2  2012-10-02   126
## 3  2012-10-03 11352
## 4  2012-10-04 12116
## 5  2012-10-05 13294
## 6  2012-10-06 15420
## 7  2012-10-07 11015
## 8  2012-10-08    NA
## 9  2012-10-09 12811
## 10 2012-10-10  9900
## 11 2012-10-11 10304
## 12 2012-10-12 17382
## 13 2012-10-13 12426
## 14 2012-10-14 15098
## 15 2012-10-15 10139
## 16 2012-10-16 15084
## 17 2012-10-17 13452
## 18 2012-10-18 10056
## 19 2012-10-19 11829
## 20 2012-10-20 10395
## 21 2012-10-21  8821
## 22 2012-10-22 13460
## 23 2012-10-23  8918
## 24 2012-10-24  8355
## 25 2012-10-25  2492
## 26 2012-10-26  6778
## 27 2012-10-27 10119
## 28 2012-10-28 11458
## 29 2012-10-29  5018
## 30 2012-10-30  9819
## 31 2012-10-31 15414
## 32 2012-11-01    NA
## 33 2012-11-02 10600
## 34 2012-11-03 10571
## 35 2012-11-04    NA
## 36 2012-11-05 10439
## 37 2012-11-06  8334
## 38 2012-11-07 12883
## 39 2012-11-08  3219
## 40 2012-11-09    NA
## 41 2012-11-10    NA
## 42 2012-11-11 12608
## 43 2012-11-12 10765
```

```
## 44 2012-11-13  7336
## 45 2012-11-14    NA
## 46 2012-11-15    41
## 47 2012-11-16  5441
## 48 2012-11-17 14339
## 49 2012-11-18 15110
## 50 2012-11-19  8841
## 51 2012-11-20  4472
## 52 2012-11-21 12787
## 53 2012-11-22 20427
## 54 2012-11-23 21194
## 55 2012-11-24 14478
## 56 2012-11-25 11834
## 57 2012-11-26 11162
## 58 2012-11-27 13646
## 59 2012-11-28 10183
## 60 2012-11-29  7047
## 61 2012-11-30    NA
```

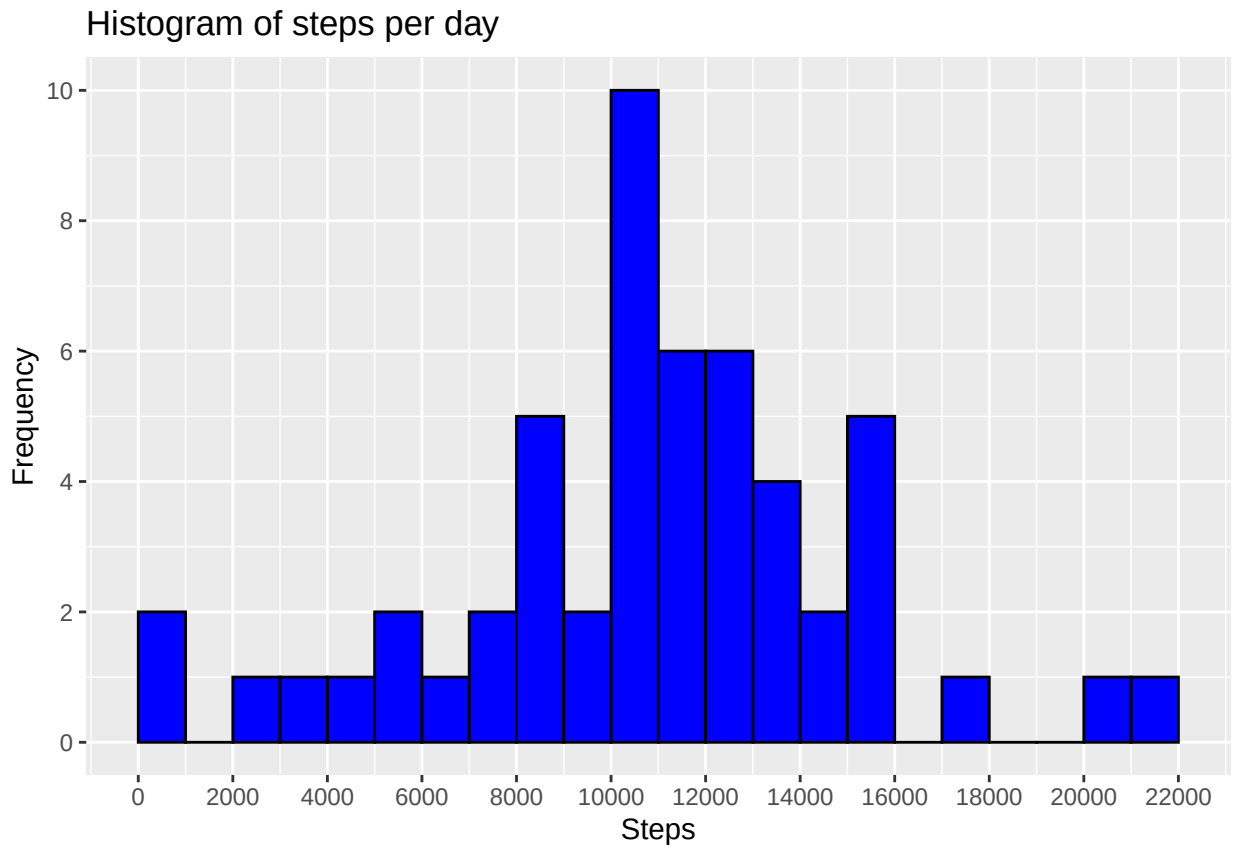
```
summary(StepsDaily)
```

```
##      Date               Steps
##  Length:61          Min.   :   41
##  Class :character   1st Qu.: 8841
##  Mode  :character   Median :10765
##                     Mean   :10766
##                     3rd Qu.:13294
##                     Max.   :21194
##                     NA's   :8
```

2. Make a histogram of the total number of steps taken each day.

```
library(ggplot2)
g <- ggplot(StepsDaily, aes(Steps))
g+geom_histogram(boundary=0, binwidth=1000, col="black",fill="blue") +ggtitle("Histogram of steps per da
```

```
## Warning: Removed 8 rows containing non-finite values (stat_bin).
```



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

```
MeanSteps <- mean(StepsDaily$Steps, na.rm=TRUE)
MeanSteps
```

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

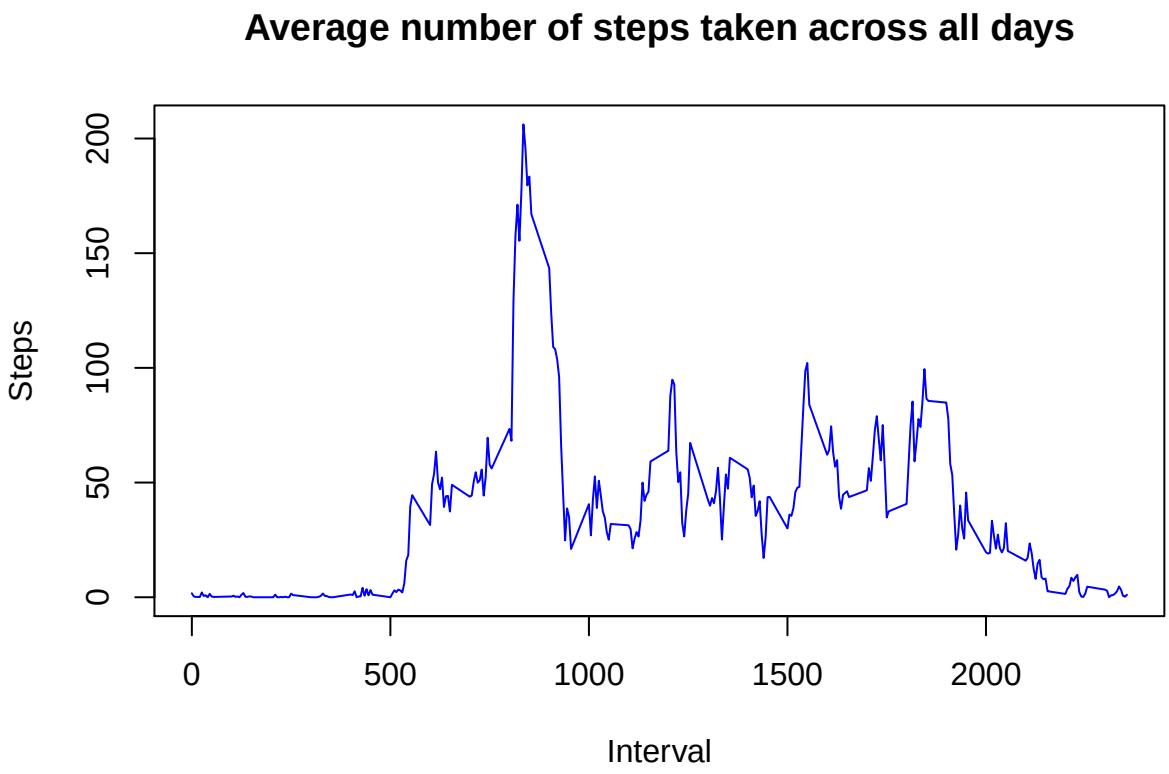
```
MedianSteps <- median(StepsDaily$Steps, na.rm=TRUE)
MedianSteps
```

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

What is the average daily activity pattern?

1. Make a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis).

```
StepsIntervals<-aggregate(steps~interval,data=activity,FUN=mean,na.action=na.omit)
StepsIntervals$time <- StepsIntervals$interval/100
plot(steps~interval, data=StepsIntervals, type="l", ylab = "Steps", xlab="Interval", main="Average numbe
```



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

```
MaxSteps <- StepsIntervals[which.max(StepsIntervals$steps),]$interval
MaxSteps
```

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

Imputing missing values

1. Calculate and report the total number of missing values in the dataset.

```
MissingValues <- sum(is.na(activity$steps))
MissingValues
```

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

2. Devise a strategy for filling in all of the missing values in the dataset.

```
activity$CompleteSteps <- ifelse(is.na(activity$steps), round(StepsIntervals$steps[match(activity$interv
```

3. Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
activityFull <- data.frame(steps=activity$CompleteSteps, interval=activity$interval, date=activity$date)
head(activityFull)
```

```
##   steps interval       date
## 1     2        0 2012-10-01
## 2     0        5 2012-10-01
## 3     0       10 2012-10-01
```

```
## 4     0       15 2012-10-01
## 5     0       20 2012-10-01
## 6     2       25 2012-10-01
```

```
str(activityFull)
```

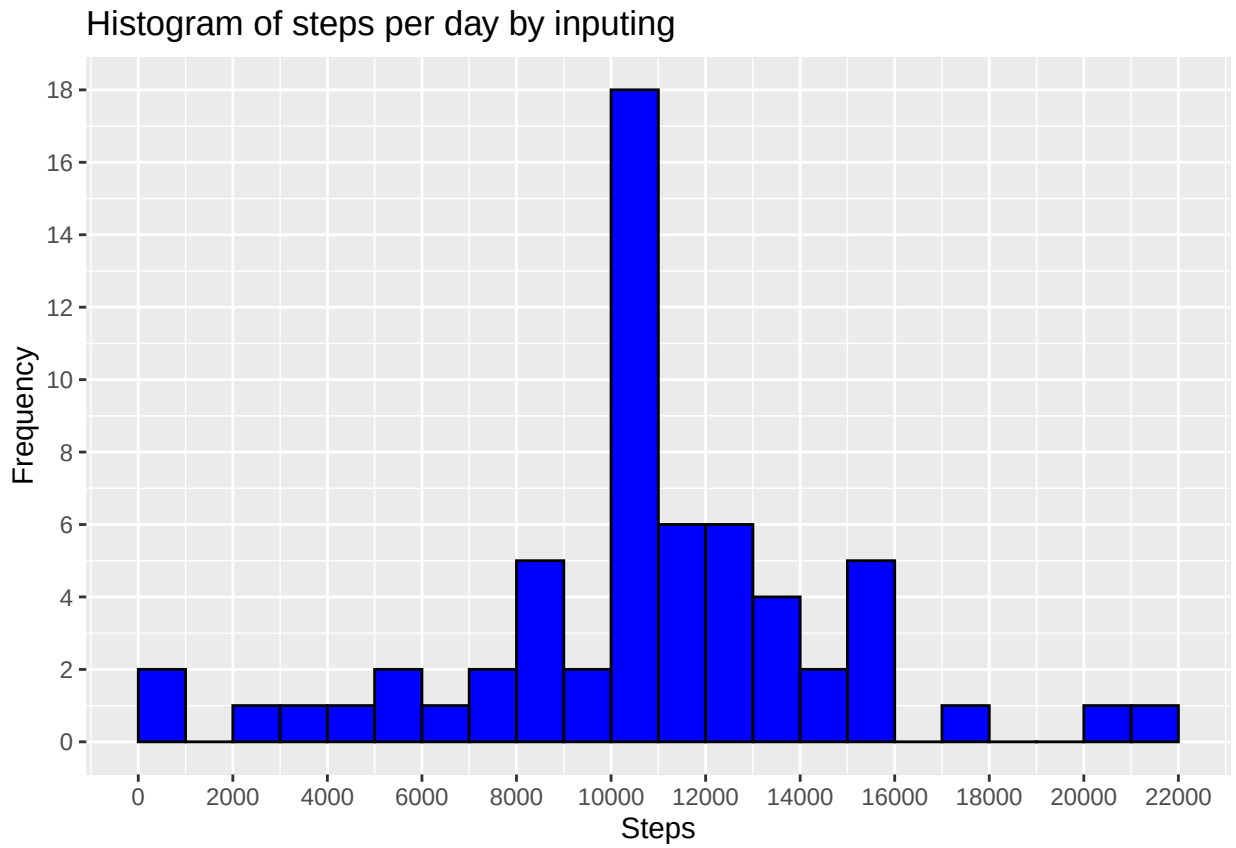
```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : num  2 0 0 0 0 2 1 1 0 1 ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
##  $ date    : chr  "2012-10-01" "2012-10-01" "2012-10-01" "2012-10-01" ...
```

```
summary(activityFull)
```

```
##      steps           interval          date
##  Min.   :  0.00   Min.   :   0.0   Length:17568
##  1st Qu.:  0.00   1st Qu.: 588.8   Class :character
##  Median :  0.00   Median :1177.5   Mode  :character
##  Mean   : 37.38   Mean   :1177.5
##  3rd Qu.: 27.00   3rd Qu.:1766.2
##  Max.   :806.00   Max.   :2355.0
```

4. Make a histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day.

```
Stepsbyinputing <- aggregate(activityFull$steps, list(activityFull$date), FUN=sum)
colnames(Stepsbyinputing) <- c("Date", "Steps")
g2 <- ggplot(Stepsbyinputing, aes(Steps))
g2+geom_histogram(boundary=0, binwidth=1000, col="black",fill="blue") +ggtitle("Histogram of steps per d
```



Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

```
MeanInput <- mean(Stepsbyinputing$Steps)
MeanInput
```

```
## [1] 10765.64
```

```
MedianInput <- median(Stepsbyinputing$Steps)
MedianInput
```

```
## [1] 10762
```

```
DiffMean <- MeanInput-MeanSteps
DiffMean
```

```
## [1] -0.549335
```

```
DiffMedian <- MedianInput-MedianSteps
DiffMedian
```

```
## [1] -3
```

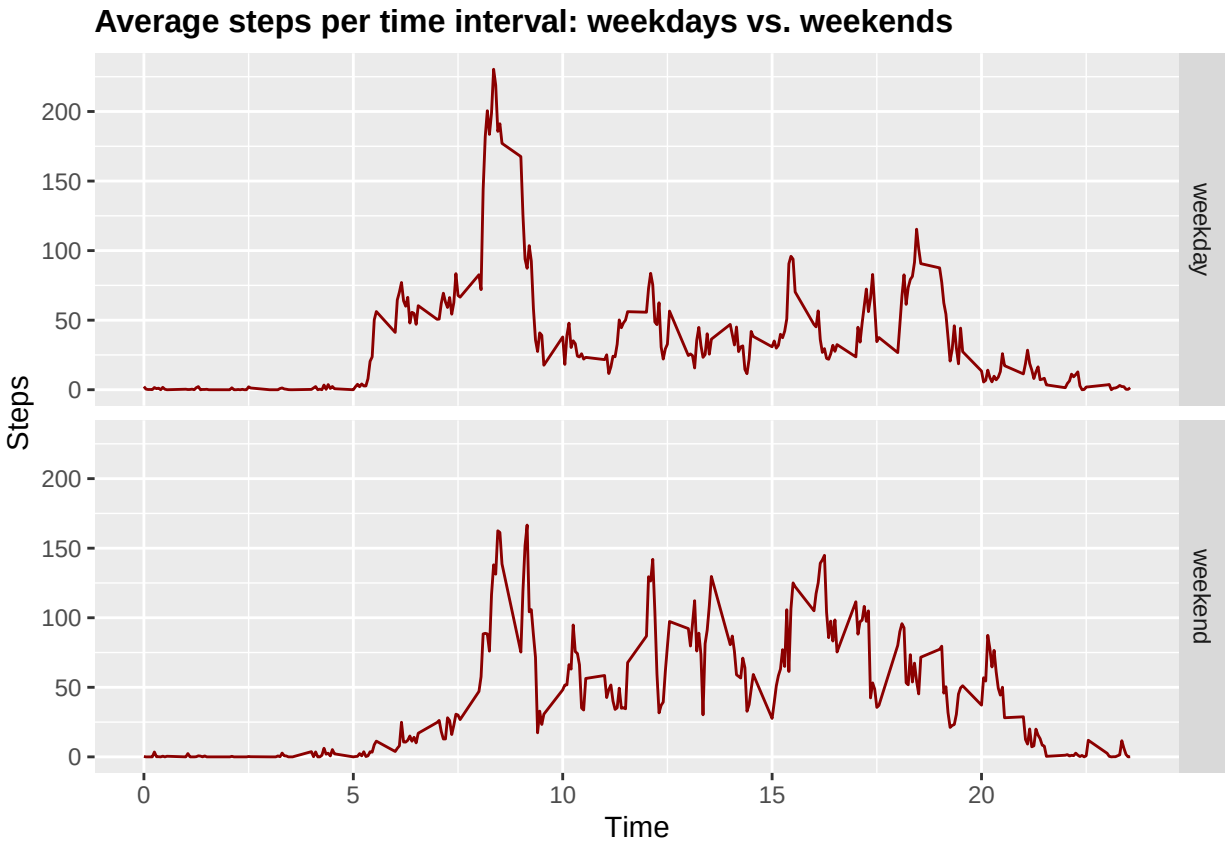
The differences are small. ## Are there differences in activity patterns between weekdays and weekends? 1. Create a new factor variable in the dataset with two levels – "weekday" and "weekend" indicating whether a given date is a weekday or weekend day.

```
activityFull$RealDate <- as.Date(activityFull$date, format = "%Y-%m-%d")
activityFull$weekday <- weekdays(activityFull$RealDate)
activityFull$DayType <- ifelse(activityFull$weekday=='sábado' | activityFull$weekday=='domingo', 'weeke
head(activityFull)
```

```
##   steps interval       date   RealDate weekday DayType
## 1     2        0 2012-10-01 2012-10-01   lunes weekday
## 2     0        5 2012-10-01 2012-10-01   lunes weekday
## 3     0       10 2012-10-01 2012-10-01   lunes weekday
## 4     0       15 2012-10-01 2012-10-01   lunes weekday
## 5     0       20 2012-10-01 2012-10-01   lunes weekday
## 6     2       25 2012-10-01 2012-10-01   lunes weekday
```

2. Make a panel plot containing a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis).

```
StepsPerTimeDT <- aggregate(steps~interval+DayType,data=activityFull,FUN=mean,na.action=na.omit)
StepsPerTimeDT$time <- StepsIntervals$interval/100
j <- ggplot(StepsPerTimeDT, aes(time, steps))
j+geom_line(col="darkred")+ggtitle("Average steps per time interval: weekdays vs. weekends")+xlab("Time
```


Average steps per time interval: weekdays vs. weekends
Steps
200
150
100
50
0
weekday
weekend
0
5
10
15
20
Time