Reproducible Research Course: Assignment 1

By Kamran Asgarov.

In this assignment, we are tasked with generating a report to answer a list of questions regarding the data from a Personal Activity Monitor (PAM). The code will make use of ggplot2, dplyr, scales and lubridate R packages, so lets make sure that they are installed and loaded.

This can be accomplished by a simple if loop, which checks if a package is installed and downloads it if it is not.

```
if(!require(ggplot2)){
    install.packages("ggplot2")
    }

if(!require(scales)){
    install.packages("scales")
    }

if(!require(dplyr)){
    install.packages("dplyr")
    }


if(!require(lubridate)){
    install.packages("lubridate")
    }

library(ggplot2)

library(scales)

library(dplyr)

library(lubridate)
```

Next, lets read the raw data from the PAM.

```
if(!"activity.csv" %in% list.files()){

    download.file(url = "https://github.com/askerovk/RepData_PeerAssessment1/blob/master/acti
vity.zip", destfile = "activity.zip")
    unzip(zipfile = "repdata_data_activity.zip")
}
raw<-read.csv(file = "activity.csv", header = TRUE, na.strings = "NA")

head(raw)
```

```
##   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
```

AS you can see, the interval is not stored as a time variable, which will complicate further research, so lets convert to a more suitable format.

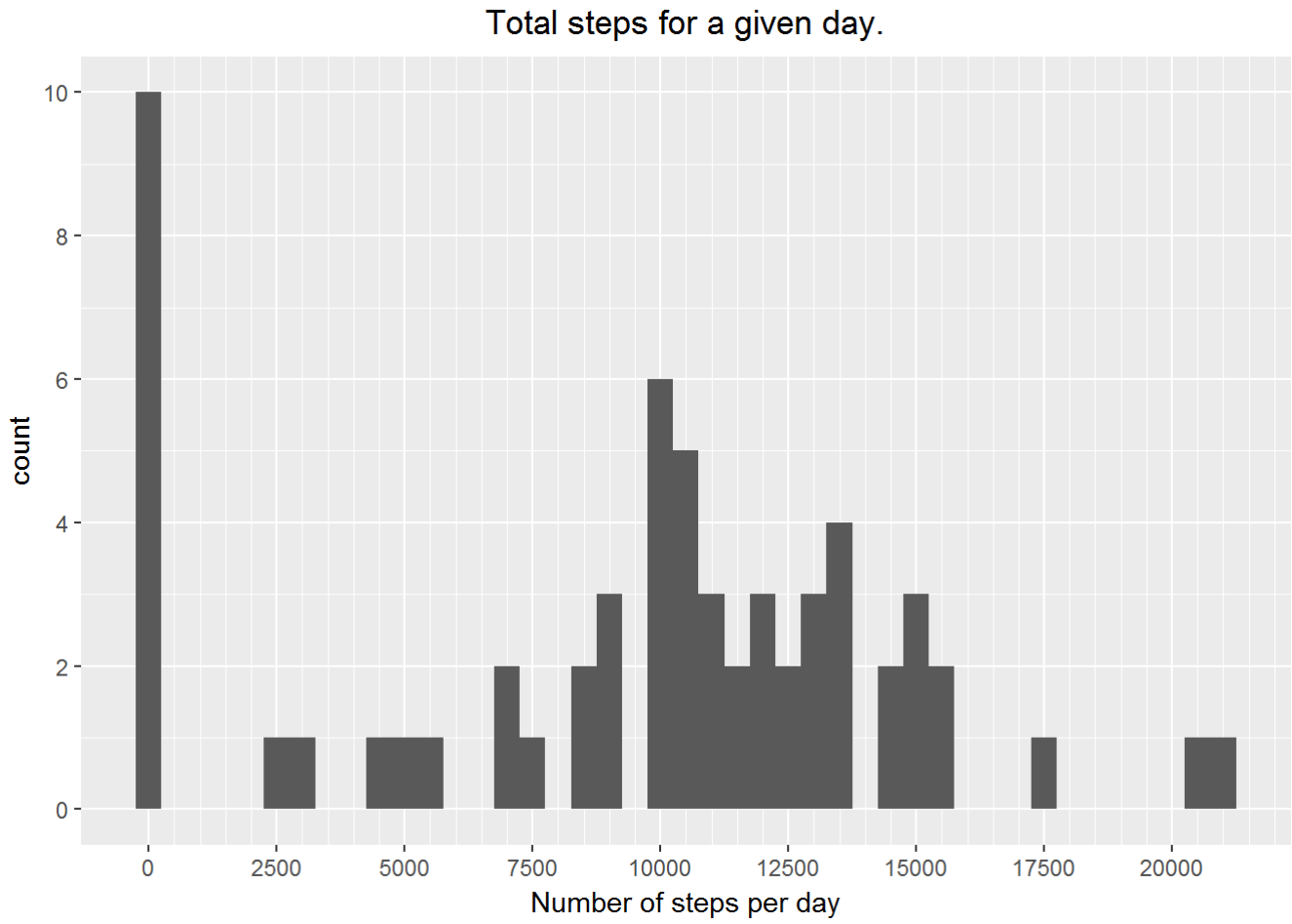
```
raw$interval<-formatC(raw$interval, width = 4, format = "d", flag = "0")

raw$interval<-format(strptime(raw$interval, format = "%H%M"), format = "%H:%M")

raw$interval<-sapply(raw$interval, function(x) {paste("2017-02-22", x, sep = " ")})

head(raw)
```

```
##   steps       date         interval
## 1    NA 2012-10-01 2017-02-22 00:00
## 2    NA 2012-10-01 2017-02-22 00:05
## 3    NA 2012-10-01 2017-02-22 00:10
## 4    NA 2012-10-01 2017-02-22 00:15
## 5    NA 2012-10-01 2017-02-22 00:20
## 6    NA 2012-10-01 2017-02-22 00:25
```

As you can see above, the data has 3 variables: steps, date and interval. Each column has only 1 variable, each row has 1 observation and the variable names lower case, so the data set is tidy.

Now, let us build the histogram of the total number of steps taken per day, as required by the assignment.

```
data1<- group_by(.data = raw, date)

day<- summarise(data1, daily=sum(steps, na.rm = TRUE))

ggplot(day, aes(x = daily))+geom_histogram(binwidth = 500)+ggtitle("Total steps for a given d
ay.")+scale_x_continuous(name = "Number of steps per day", breaks = seq(0, 22000, 2500), mino
r_breaks = seq(0, 22000, 500) )+ scale_y_continuous(breaks=seq(0,11,2)) +theme(plot.title = e
lement_text(hjust = 0.5))
```



The next task is to calculate the mean and median number of steps taken per day.

```
stepsmean<- mean(day$daily, na.rm = TRUE)

stepsmedian<-median(day$daily, na.rm = TRUE)

stepsmean
```

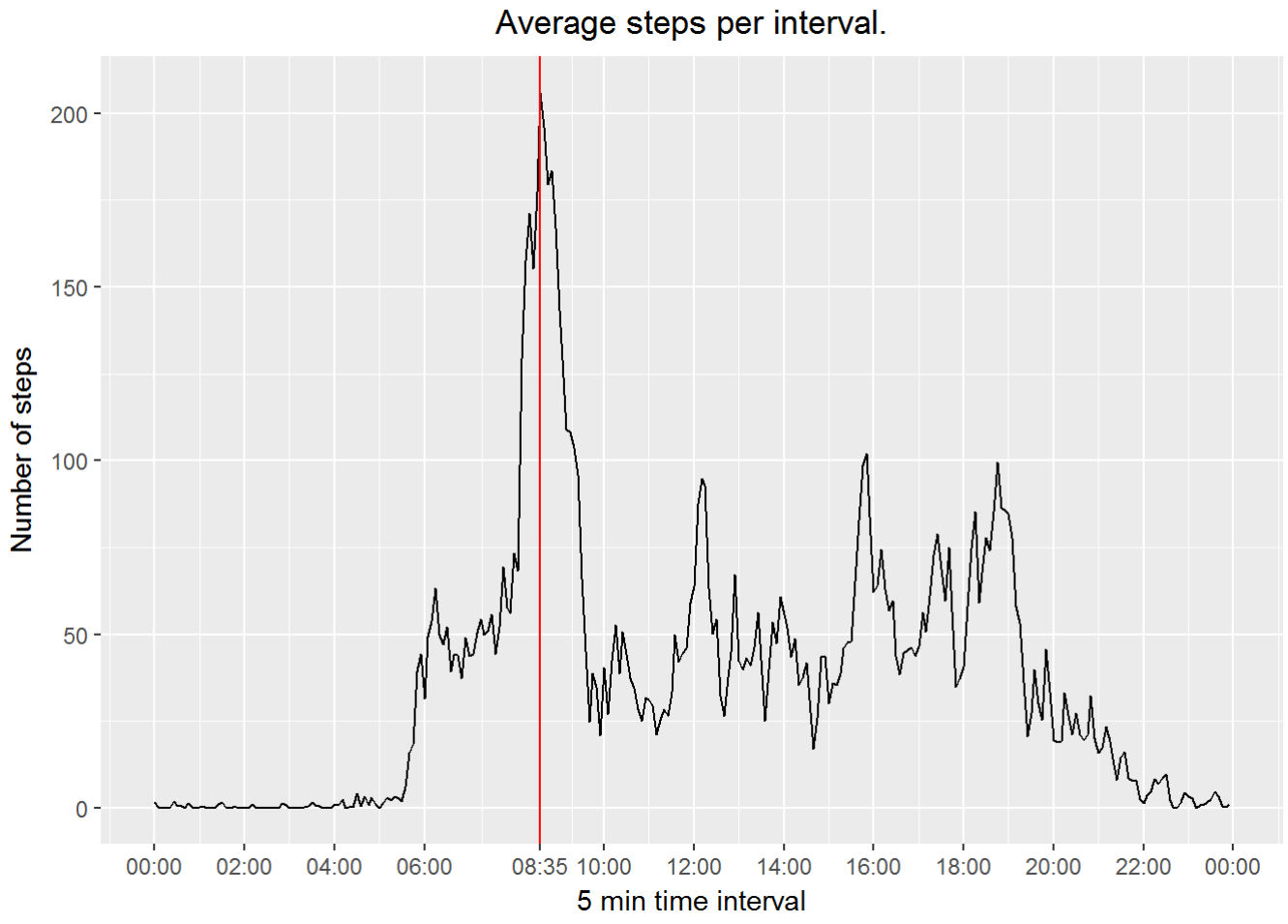
```
## [1] 9354.23
```

```
stepsmedian
```

```
## [1] 10395
```

As requested by the assignment, we need to make a make a time plot of the average number of steps taken per the 5 min interval and to highlight the interval which has the highest number of steps, on average

```
data2<-group_by(.data = raw, interval)

average<- summarise(.data = data2, av=mean(steps, na.rm = TRUE))

xaxis1<- seq(as.POSIXct("2017-02-22 00:00:00", tz = "UTC"), as.POSIXct("2017-02-22 06:00:00",
 tz = "UTC"), by = "2 hours")

xaxis2<-as.POSIXct("2017-02-22 08:35:00", tz = "UTC")

xaxis3<- seq(as.POSIXct("2017-02-22 10:00:00", tz = "UTC"), as.POSIXct("2017-02-23 00:00:00",
 tz = "UTC"), by = "2 hours")

xaxis<-c(xaxis1, xaxis2, xaxis3)


ggplot(average, aes(as.POSIXct(average$interval, format="%Y-%m-%d %H:%M", tz = "UTC"), av))+

    geom_line()+ggtitle("Average steps per interval.")+theme(plot.title = element_text(hjust
= 0.5))+

    ylab("Number of steps")+geom_vline(xintercept = as.numeric(xaxis2), col="red")+

    scale_x_datetime(breaks = xaxis ,labels = date_format("%H:%M"))+xlab("5 min time interva
l")
```



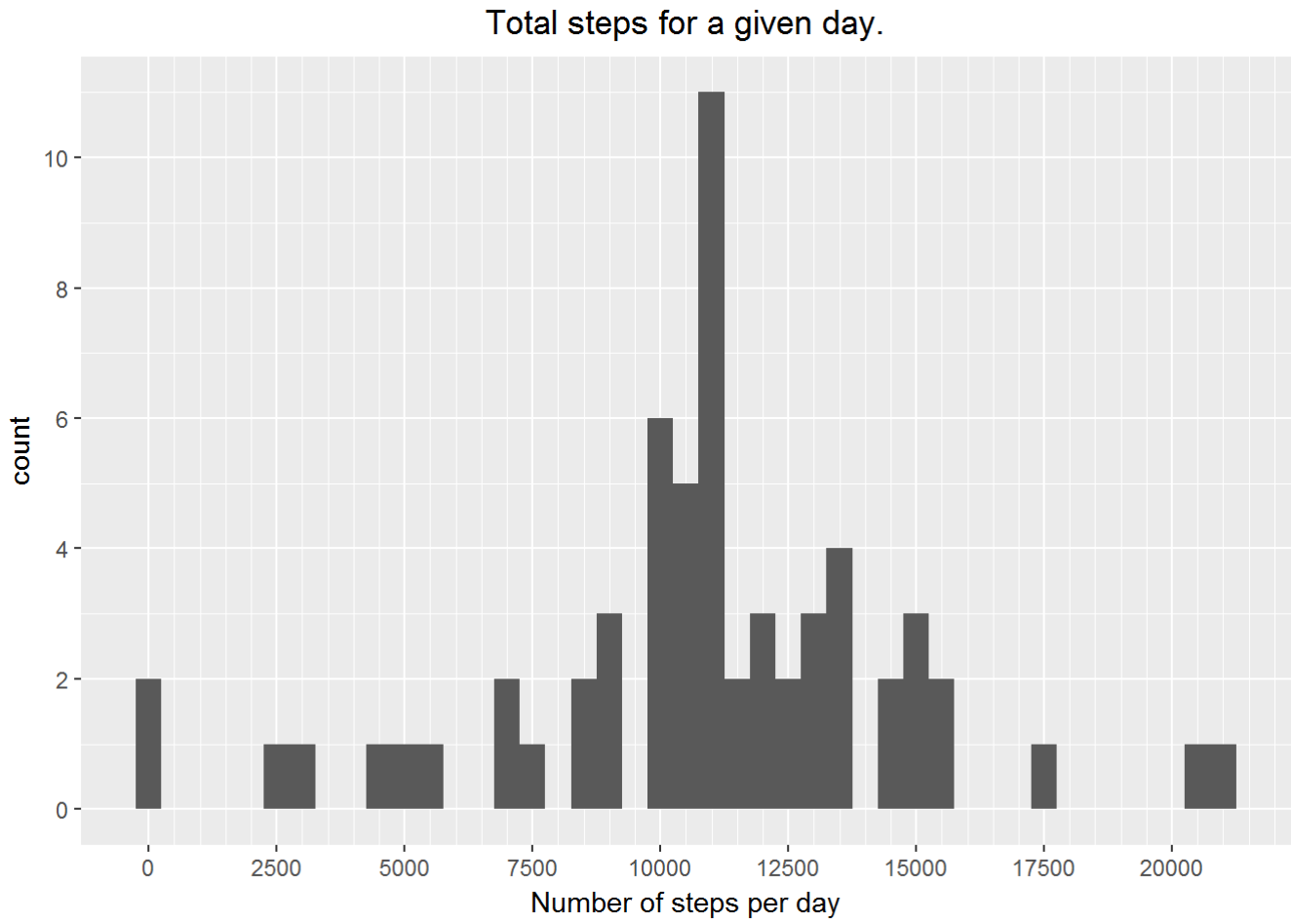
From the time plot above, the 08:35-08:40 is the time interval with the highest number of steps on average.

The following task requires us to impute the missing values in the “steps” variable. I have chosen to do this by replacing the NA values of steps taken for a given 5 min interval with the mean value of that interval, as previously determined.

```
impute<-merge(x = raw, y = average, by= "interval", sort = TRUE)

impute<- select(.data = impute, date, interval, steps, av)

var1<-impute$steps

var2<-impute$av

for(i in 1:length(var1)) {

         if(is.na(var1[i])) {

             var1[i]<-var2[i]
}}

impute$steps<-var1
```

Now that missing values have been imputed, lets make a new histogram, to see if this changes the distribution.

```
data3<-group_by(.data = impute, date)

iday<- summarise(.data = data3, daily=sum(steps))

ggplot(iday, aes(x = daily))+geom_histogram(binwidth = 500)+ggtitle("Total steps for a given
 day.")+

    scale_x_continuous(name = "Number of steps per day", breaks = seq(0, 22000, 2500), minor_
breaks = seq(0, 22000, 500) )+

    scale_y_continuous(breaks = seq(0,11,2)) +theme(plot.title = element_text(hjust = 0.5))
```



As seen from the diagram, the most populated bin of changed form the “0” to the “11000” bin.

The final task of this assignment involves comparing the mean number of steps taken for a given 5 min time interval on weekends vs weekdays. To accomplish this, let us first create the necessary factor variable.

```
caldates<- strptime(impute$date, format = "%Y-%m-%d", tz = "UTC")

var3<-seq_along(impute$date)

for(i in 1:length(var3)) {

    if(wday(caldates[i], label = TRUE) %in% c("Sat", "Sun")) {

        var3[i]<-"weekend"
    }

    else {

        var3[i]<-"weekday"
    }
}

impute$day<-var3
```

Now that the variable has been created lets graph the results.

```
data4<- group_by(impute, interval, day)

iwday<- summarise(data4, av=mean(steps))


xaxis4<- seq(as.POSIXct("2017-02-22 00:00:00", tz = "UTC"), as.POSIXct("2017-02-23 00:00:00",
 tz = "UTC"), by = "2 hours")

ggplot(iwday, aes(as.POSIXct(iwday$interval, format="%Y-%m-%d %H:%M", tz = "UTC"), av, col=da
y))+geom_line()+

    ggtitle("Average steps per interval.")+theme(plot.title = element_text(hjust =
0.5))+ylab("Number of steps")+

    scale_x_datetime(breaks = xaxis4, labels = date_format("%H:%M"))+xlab("5 min time interva
l")
```