Reproducible Research: Peer Assessment 1

Dhruvin Shah

5/21/2020

Loading and preprocessing the data

1. Loading data using read.csv() function

```
data_raw <- read.csv("activity.csv", header = TRUE)
head(data_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
```

Dimension of Data:

```
dim(data_raw)
```

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

Structure of Data:

```
str(data_raw)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

2. Preprocessing the data

```
data <- data_raw
data$date <- as.Date(data$date)
str(data)
```

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

What is mean total number of steps taken per day?

Finding total number of steps by using group_by() and summarise() function of dplyr library.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

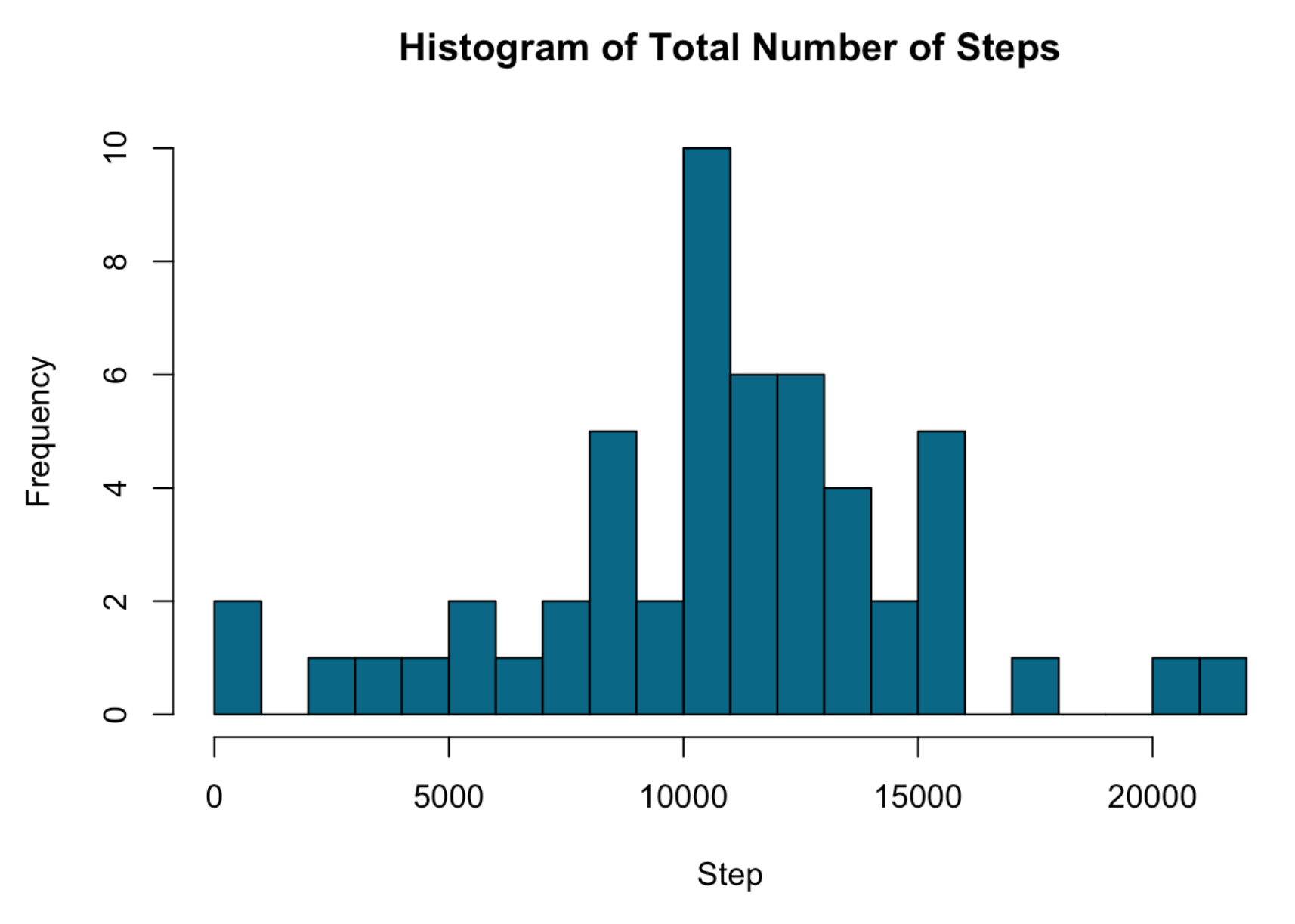
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
TotalStepsData <- data %>%
    na.omit() %>%
    group_by(date) %>%
    summarise(Steps = sum(steps))
TotalStepsData <- as.data.frame(TotalStepsData)
head(TotalStepsData)
```

```
##         date Steps
## 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
```

Histogram of Total Steps is

```
hist(TotalStepsData$Steps,
     main = "Histogram of Total Number of Steps",
     breaks = 25,
     col = "#086A87",
     xlab = "Step")
```



Mean of Total Steps is:

```
mean(TotalStepsData$Steps)
```

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

Median of Total Steps is:

```
median(TotalStepsData$Steps)
```

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

What is the average daily activity pattern?

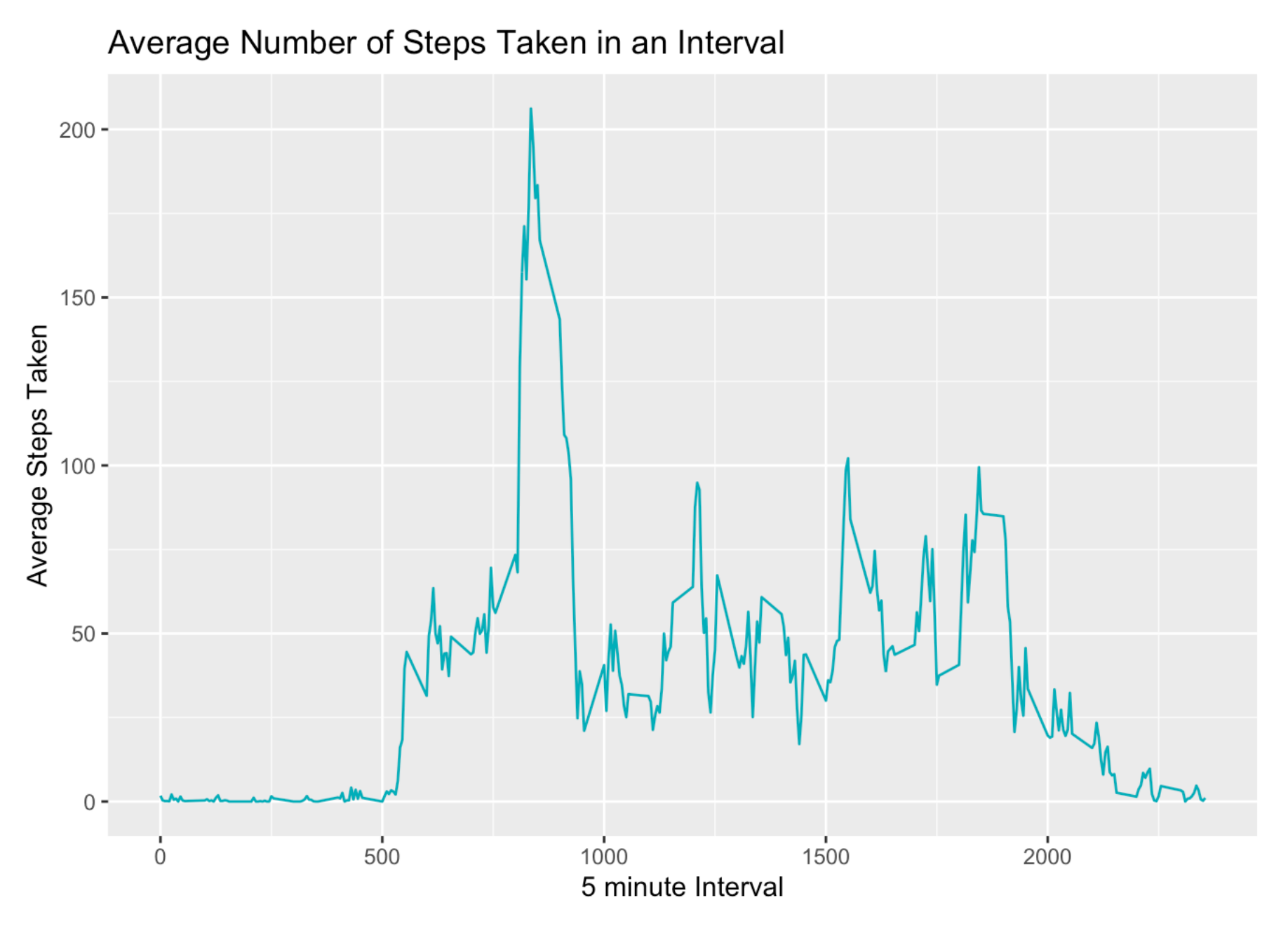
Finding Average steps during the 5 - min interval using group_by() and summarise() functions of dplyr library.

```
AverageStepsData <- data %>%
    na.omit() %>%
    group_by(interval) %>%
    summarise(Steps = mean(steps))
AverageStepsData <- as.data.frame(AverageStepsData)
head(AverageStepsData)
```

```
##   interval     Steps
## 1        0 1.7169811
## 2        5 0.3396226
## 3       10 0.1320755
## 4       15 0.1509434
## 5       20 0.0754717
## 6       25 2.0943396
```

Plotting the Time-series using ggplot() function

```
library(ggplot2)
ggplot(data = AverageStepsData,
       aes(x = interval,
           y = Steps)) +
    geom_line(color = "#00AFBB", size = 0.5) +
    labs(title = "Average Number of Steps Taken in an Interval") +
    xlab("5 minute Interval") +
    ylab("Average Steps Taken")
```



Imputing missing values

Checking for Missing Values:

```
any(is.na(data))
```

```
## [1] TRUE
```

Total Number of Missing Values:

```
sum(is.na(data))
```

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

Missing Values in each column:

```
sapply(data, function(x) sum(length(which(is.na(x)))))
```

```
##    steps     date interval
##     2304        0        0
```

Percentage of Missing Values:

```
sapply(data, function(x) (mean(is.na(x)))*100)
```

```
##    steps     date interval
## 13.11475  0.00000  0.00000
```

Replacing missing values with average of interval

```
cleandata <- data
 for(i in AverageStepsData$interval) {
     cleandata[cleandata$interval == i & is.na(cleandata$steps), ]$steps <- AverageStepsData$Steps[AverageStepsDa
ta$interval == i]
 }
```

Checking for any NA values in updated dataset

```
any(is.na(cleandata))
```

```
## [1] FALSE
```

```
head(cleandata)
```

```
##       steps       date interval
## 1 1.7169811 2012-10-01        0
## 2 0.3396226 2012-10-01        5
## 3 0.1320755 2012-10-01       10
## 4 0.1509434 2012-10-01       15
## 5 0.0754717 2012-10-01       20
## 6 2.0943396 2012-10-01       25
```

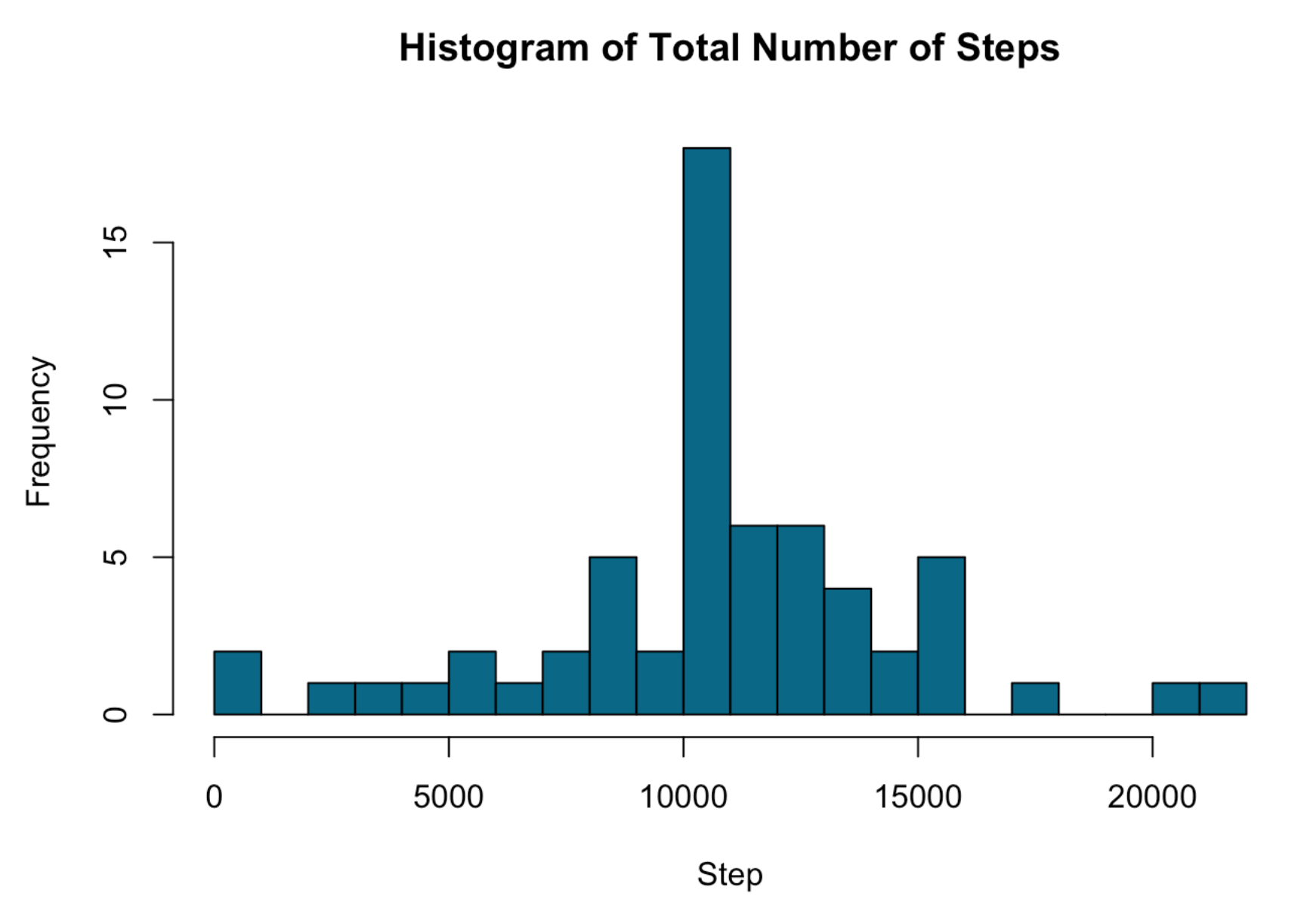
Finding total number of steps in clean data by using group_by() and summarise() function of dplyr library.

```
TotalStepsCleanData <- cleandata %>%
    group_by(date) %>%
    summarise(steps = sum(steps))
TotalStepsCleanData <- as.data.frame(TotalStepsCleanData)
head(TotalStepsCleanData)
```

```
##         date    steps
## 1 2012-10-01 10766.19
## 2 2012-10-02   126.00
## 3 2012-10-03 11352.00
## 4 2012-10-04 12116.00
## 5 2012-10-05 13294.00
## 6 2012-10-06 15420.00
```

Histogram of Total Steps of Clean Data is

```
hist(TotalStepsCleanData$steps,
     main = "Histogram of Total Number of Steps",
     breaks = 25,
     col = "#086A87",
     xlab = "Step")
```



Mean of Total Steps is:

```
mean(TotalStepsCleanData$steps)
```

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

Median of Total Steps is:

```
median(TotalStepsCleanData$steps)
```

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

Are there differences in activity patterns between weekdays and weekends?

Determing the Phase of Week using weekdays() function

```
cleandata$phase <- ""
cleandata$date <- as.Date(cleandata$date)
cleandata[weekdays(cleandata$date) == "Saturday" | weekdays(cleandata$date) == "Saturday",]$phase <- "weekend"
cleandata[!(weekdays(cleandata$date) == "Saturday" | weekdays(cleandata$date) == "Saturday"),]$phase <- "weekday"
head(cleandata)
```

```
##       steps       date interval   phase
## 1 1.7169811 2012-10-01        0 weekday
## 2 0.3396226 2012-10-01        5 weekday
## 3 0.1320755 2012-10-01       10 weekday
## 4 0.1509434 2012-10-01       15 weekday
## 5 0.0754717 2012-10-01       20 weekday
## 6 2.0943396 2012-10-01       25 weekday
```

Finding Average steps of Week Phase during the 5 - min interval using group_by() and summarise() functions of dplyr library.

```
AverageStepsWeekPhase <- cleandata %>%
    group_by(phase, interval) %>%
    summarise(steps = mean(steps))
AverageStepsWeekPhase <- as.data.frame(AverageStepsWeekPhase)
head(AverageStepsWeekPhase)
```

```
##     phase interval      steps
## 1 weekday        0 1.94375222
## 2 weekday        5 0.38447846
## 3 weekday       10 0.14951940
## 4 weekday       15 0.17087932
## 5 weekday       20 0.08543966
## 6 weekday       25 2.37095052
```

Subsetting the Weekday and Weekend Data from Weekphase

```
AverageStepsWeekday <- AverageStepsWeekPhase %>%
    filter(phase == "weekday")
head(AverageStepsWeekday)
```

```
##     phase interval      steps
## 1 weekday        0 1.94375222
## 2 weekday        5 0.38447846
## 3 weekday       10 0.14951940
## 4 weekday       15 0.17087932
## 5 weekday       20 0.08543966
## 6 weekday       25 2.37095052
```

```
AverageStepsWeekend <- AverageStepsWeekPhase %>%
    filter(phase == "weekend")
head(AverageStepsWeekend)
```

```
##     phase interval       steps
## 1 weekend        0 0.214622642
## 2 weekend        5 0.042452830
## 3 weekend       10 0.016509434
## 4 weekend       15 0.018867925
## 5 weekend       20 0.009433962
## 6 weekend       25 0.261792453
```

Plotting the Time Series of Weekdays and Weekends Activity

```
par(mfrow = c(1,2))
plot(AverageStepsWeekday$interval, AverageStepsWeekday$steps,
     main = "Weekday",
     xlab = "5 minute Interval",
     ylab = "Average Steps Taken",
     type = "l", lwd = 2, col = "navy")

plot(AverageStepsWeekend$interval, AverageStepsWeekend$steps,
     main = "Weekend",
     xlab = "5 minute Interval",
     ylab = "Average Steps Taken",
     type = "l", lwd = 2, col = "red")
```