

PA1_template

Mike Woods-DeWitt

March 15, 2019

R Markdown

```
# Installs packages
install.packages("dplyr", repos="https://cloud.r-project.org")
```

```
## package 'dplyr' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
## C:\Users\mw01166\AppData\Local\Temp\RtmpaqPdG4\downloaded_packages
```

```
library("dplyr")
```

```
## Warning: package 'dplyr' was built under R version 3.5.3
```

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

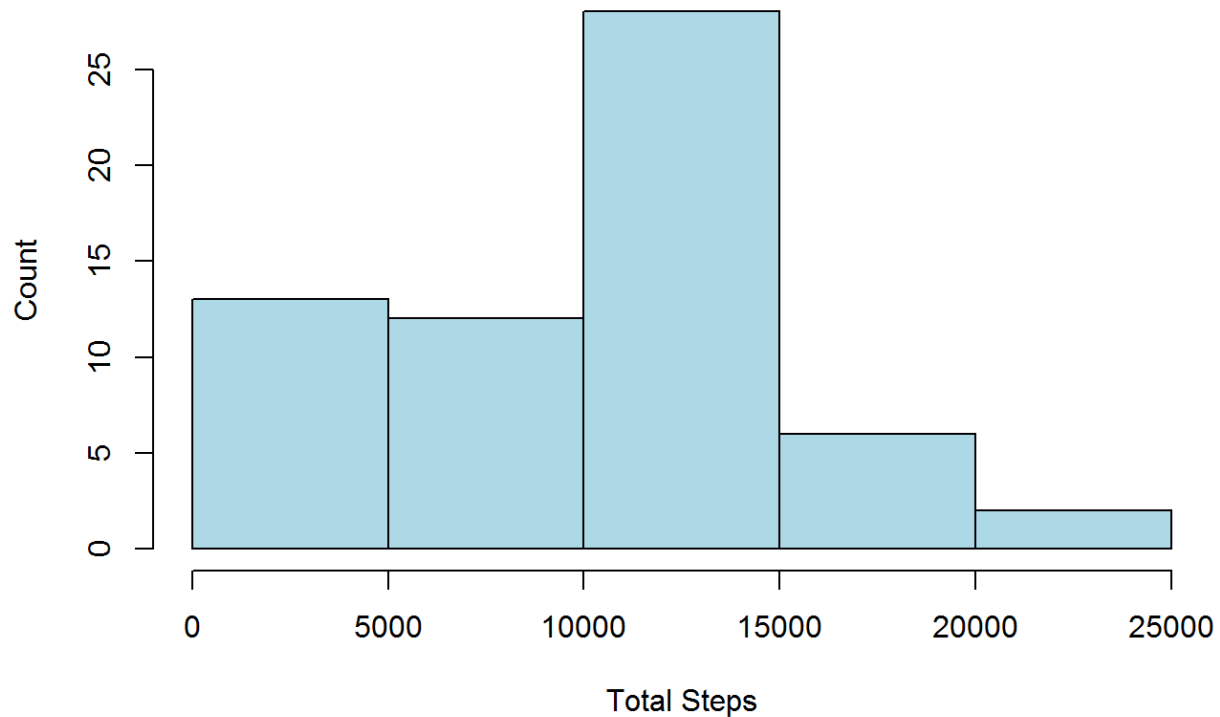
```
## The following objects are masked from 'package:stats':
##
## filter, lag
```

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

```
# sets the working directory for calls as needed to download the datasets
wd <- getwd()
download.file(url = "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip",
              , destfile = paste(wd, "repdata_data_activity.zip", sep = "/"))
# unzips and names the dataset for the assignment
unzip(zipfile = "repdata_data_activity.zip")
activity<-read.csv("./activity.csv",header = TRUE)
# Set column 2 as a date
activity[,2]<-as.Date(activity$date)
```

```
steps_q1 <- tapply(activity$steps,activity$date, sum,na.rm=TRUE)
q1 <- hist(steps_q1,col="light blue",xlab="Total Steps",ylab="Count", main = "Q
uestion 1: Histogram of the total number of steps taken each day")
```

Question 1: Histogram of the total number of steps taken each day



```
# Question 2a: Mean number of steps taken per day
answer_q2a <- mean(steps_q1)
print(answer_q2a)
```

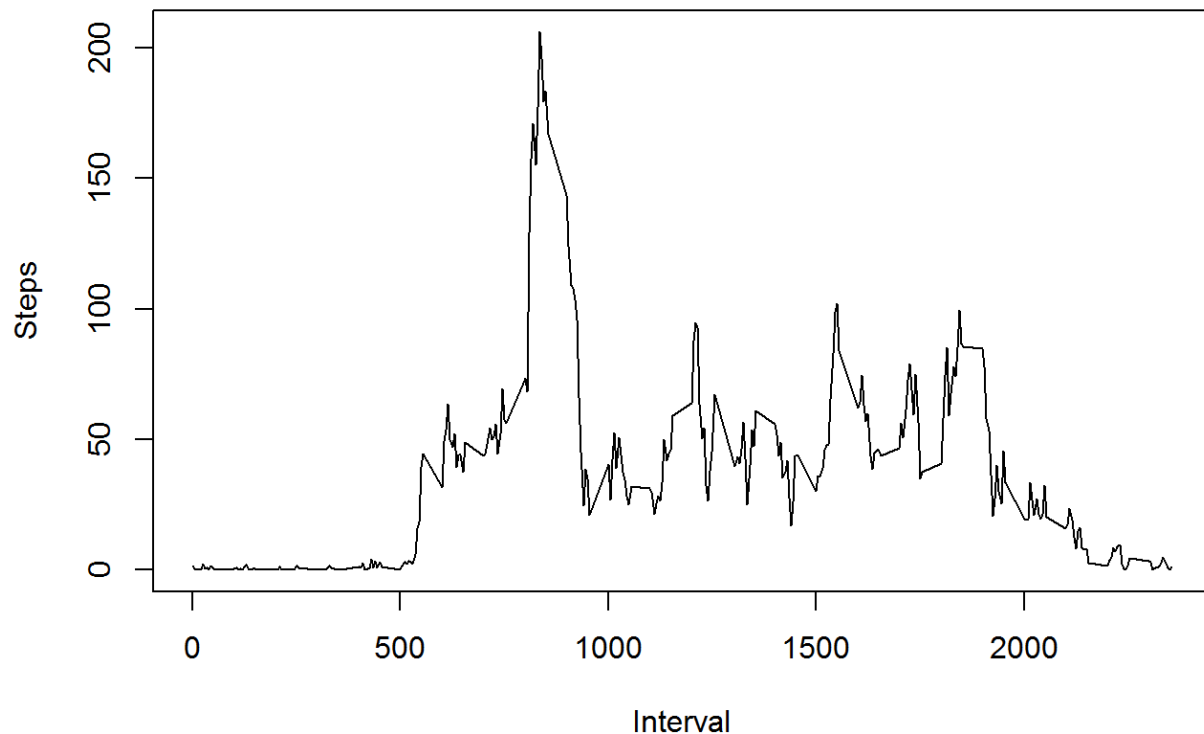
```
## [1] 9354.23
```

```
# Question 2b: Median number of steps taken per day
answer_q2b <- median(steps_q1)
print(answer_q2b)
```

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

```
Steps_time_series <- tapply(activity$steps,activity$interval, mean, na.rm=TRUE)
plot(as.numeric(names(Steps_time_series)),Steps_time_series,
     xlab = "Interval",
     ylab="Steps",
     main = "Question 3: Time Series Plot of the Average Number of Steps Take
n",
     type = "l")
```

Question 3: Time Series Plot of the Average Number of Steps Taken



```
ordered_interval <- names(sort(Steps_time_series,decreasing = TRUE))
max_interval <- ordered_interval[1]
max_interval
```

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

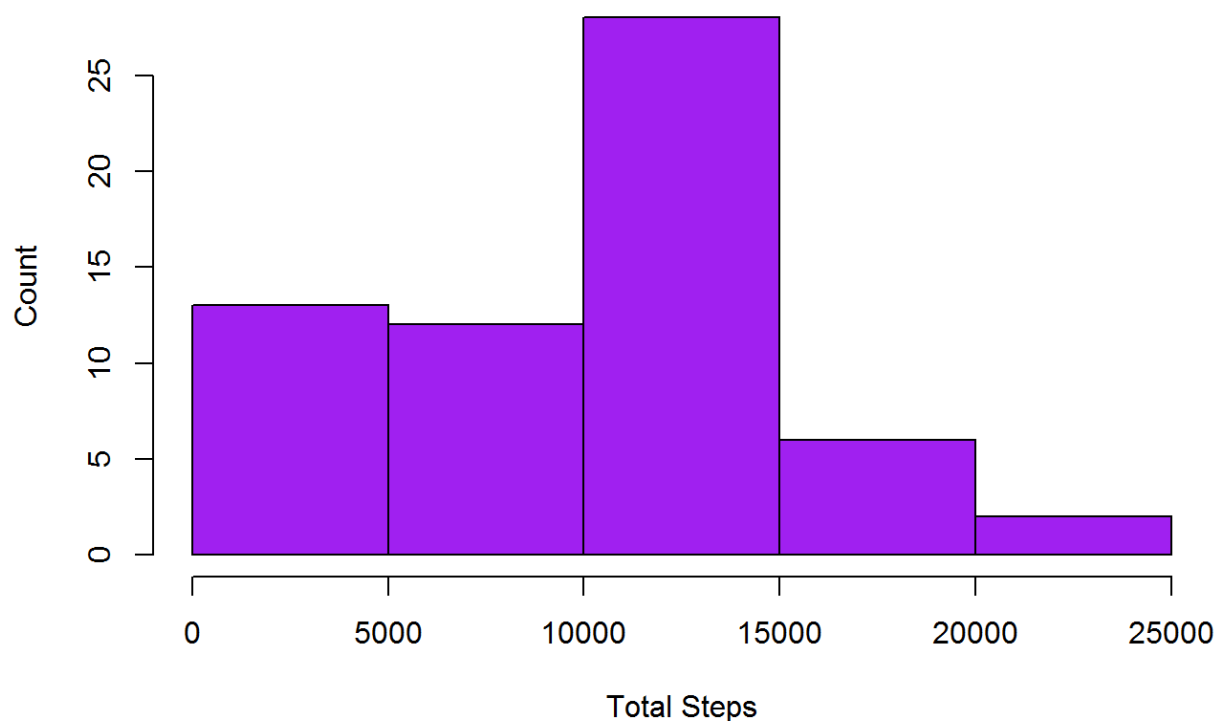
```
# Show the total number of missing values in the original dataset
NA.val_count <- sum(is.na(activity$steps))
NA.val_count
```

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

```
# Setting all missing data points to 0
# creating a new dataset to input missing values as 0
activity_revised <- activity
activity_revised[is.na(activity_revised)] <- 0
```

```
steps_q6 <- tapply(activity_revised$steps, activity_revised$date, sum, na.rm=TRUE)
q6 <- hist(steps_q6, col="purple", xlab="Total Steps", ylab="Count", main = "Question 6: Histogram of the total number of steps taken each day (revised dataset, missing data imputed)")
```

Question 6: Histogram of the total number of steps taken each day (revised dataset, missing data imputed)



```
answer_q6a <- mean(steps_q6)
print(answer_q6a)
```

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

```
# Question 6b: Median number of steps taken per day
answer_q6b <- median(steps_q6)
# By setting the missing values to 0 there is no difference between the missing value dataset and the revised dataset
```

```
# Creating a new column to differentiate weekdays from weekends
activity_day_type <- mutate(activity, weekend_or_weekday = ifelse(weekdays(activity$date) == "Saturday" | weekdays(activity$date) == "Sunday", "weekend", "weekday"))

# subsetting weekdays and weekends
weekend <- subset(activity_day_type, as.character(activity_day_type$weekend_or_weekday) == "weekend")
weekday <- subset(activity_day_type, as.character(activity_day_type$weekend_or_weekday) == "weekday")

# Set the stage for side by side plot
par(mfrow=c(1,2))

# Plot weekdays
Steps_time_series_week_days <- tapply(weekday$steps, weekday$interval, mean, na.rm=TRUE)
plot(as.numeric(names(Steps_time_series_week_days)), Steps_time_series_week_days,
     xlab = "Interval",
     ylab = "Steps",
     main = "Weekdays:
Time Series Plot
Avg Number of Steps Taken",
     type = "l")

# Plot Weekends
Steps_time_series_week_ends <- tapply(weekend$steps, weekend$interval, mean, na.rm=TRUE)
plot(as.numeric(names(Steps_time_series_week_ends)), Steps_time_series_week_ends,
     xlab = "Interval",
     ylab = "Steps",
     main = "Weekends:
Time Series Plot
Avg Number of Steps Taken",
     type = "l")
```

