Reproducible Research Peer Assessment 1

Loading and preprocessing the data

Show any code that is needed to

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

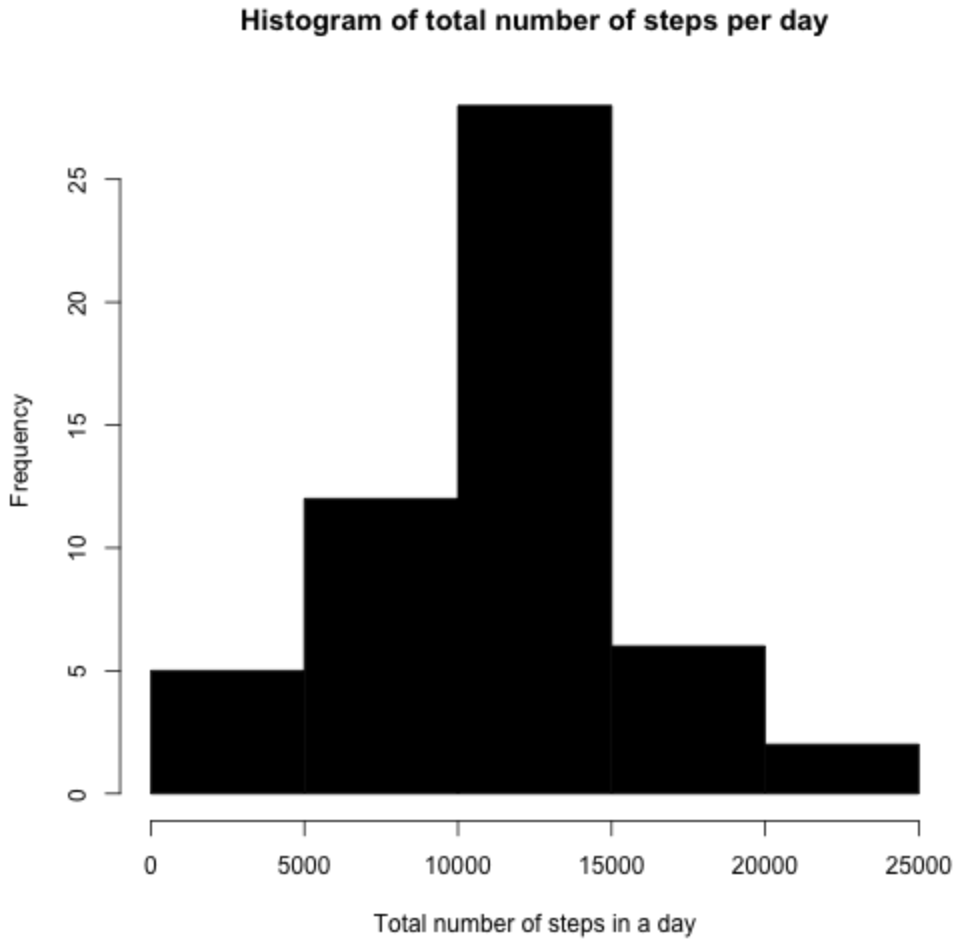
```
# set the file url
fileurl <- "
https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"

# create a temporary directory
td = tempdir()

# create the placeholder file
tf = tempfile(tmpdir=td, fileext=".zip")

# download into the placeholder file (curl method needed for Mac OS X)
download.file(fileurl, tf, method="curl")

# get the name of the first file in the zip archive
fname = unzip(tf, list=TRUE)$Name[1]

# unzip the file to the temporary directory
unzip(tf, files=fname, exdir=td, overwrite=TRUE)

# fpath is the full path to the extracted file
fpath = file.path(td, fname)

# load the csv in data frame
df <- read.csv(fpath, as.is=TRUE)
```

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.

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

```
# generate df2 with complete cases only
df2 <- na.omit(df)

# aggregate steps as per date to get total number of steps in a day
table_date_steps <- aggregate(steps ~ date, df2, sum)

# create histogram of total number of steps in a day
hist(table_date_steps$steps, col=1, main="Histogram of total number of steps
per day",
     xlab="Total number of steps in a day")
```




```
# get mean and median total number of steps per day
mean(table_date_steps$steps)
## [1] 10766
median(table_date_steps$steps)
## [1] 10765
```

The mean and median total number of steps per day are 10766 and 10765 steps respectively.

What is the average daily activity pattern?

1. 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).
2. Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

```
# aggregate steps as interval to get average number of steps in an interval
across all days
table_interval_steps <- aggregate(steps ~ interval, df2, mean)

# generate the line plot of the 5-minute interval (x-axis) and the average
number of
# steps taken, averaged across all days (y-axis)
```

```
plot(table_interval_steps$interval, table_interval_steps$steps, type='l',
col=1,
     main="Average number of steps averaged over all days", xlab="Interval",
     ylab="Average number of steps")
# find row id of maximum average number of steps in an interval
max_ave_steps_row_id <- which.max(table_interval_steps$steps)

# get the interval with maximum average number of steps in an interval
table_interval_steps [max_ave_steps_row_id, ]
##     interval steps
## 104      835 206.2
```

The interval 835 has the maximum average number of steps (206.2).

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.

1. Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs).
2. Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to be sophisticated. For example, you could use the mean/median for that day, or the mean for that 5-minute interval, etc.
3. Create a new dataset that is equal to the original dataset but with the missing data filled in.
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. 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?

```
# get rows with NA's
df_NA <- df[!complete.cases(df),]

# number of rows
nrow(df_NA)
## [1] 2304
```

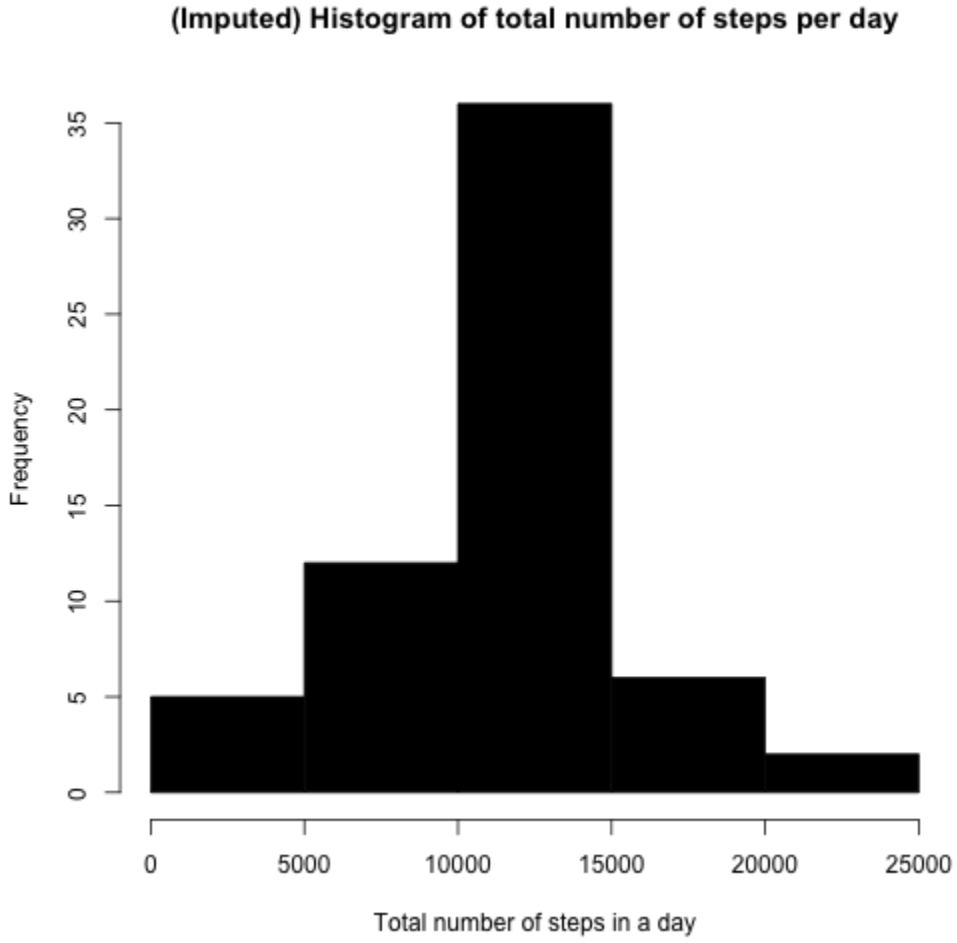
The total number of rows with NA's is 2304 as shown above.

For performing imputation, we replace the NA by the mean for that 5-minute interval. We already have this data in the data frame “table_interval_steps”.

We loop across the rows of the data frame “df”. If the steps value is NA for a row, we find the corresponding value of interval. We then look up the steps value from the other data frame “table_interval_steps” for this value of interval and replace the NA value with it.

```
# perform the imputation
for (i in 1:nrow(df)){
  if (is.na(df$steps[i])){
    interval_val <- df$interval[i]
    row_id <- which(table_interval_steps$interval == interval_val)
    steps_val <- table_interval_steps$steps[row_id]
    df$steps[i] <- steps_val
  }
}

# aggregate steps as per date to get total number of steps in a day
table_date_steps_imputed <- aggregate(steps ~ date, df, sum)
```

```
# create histogram of total number of steps in a day
hist(table_date_steps_imputed$steps, col=1, main="(Imputed) Histogram of
total number of steps per day", xlab="Total number of steps in a day")
```




```
# get mean and median of total number of steps per day
mean(table_date_steps_imputed$steps)
## [1] 10766
median(table_date_steps_imputed$steps)
## [1] 10766
# get mean and median of total number of steps per day for data with NA's
removed
mean(table_date_steps$steps)
## [1] 10766
median(table_date_steps$steps)
## [1] 10765
```

Due to data imputation, the means remain same whereas there is slight change in median value.

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.

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.
2. 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).

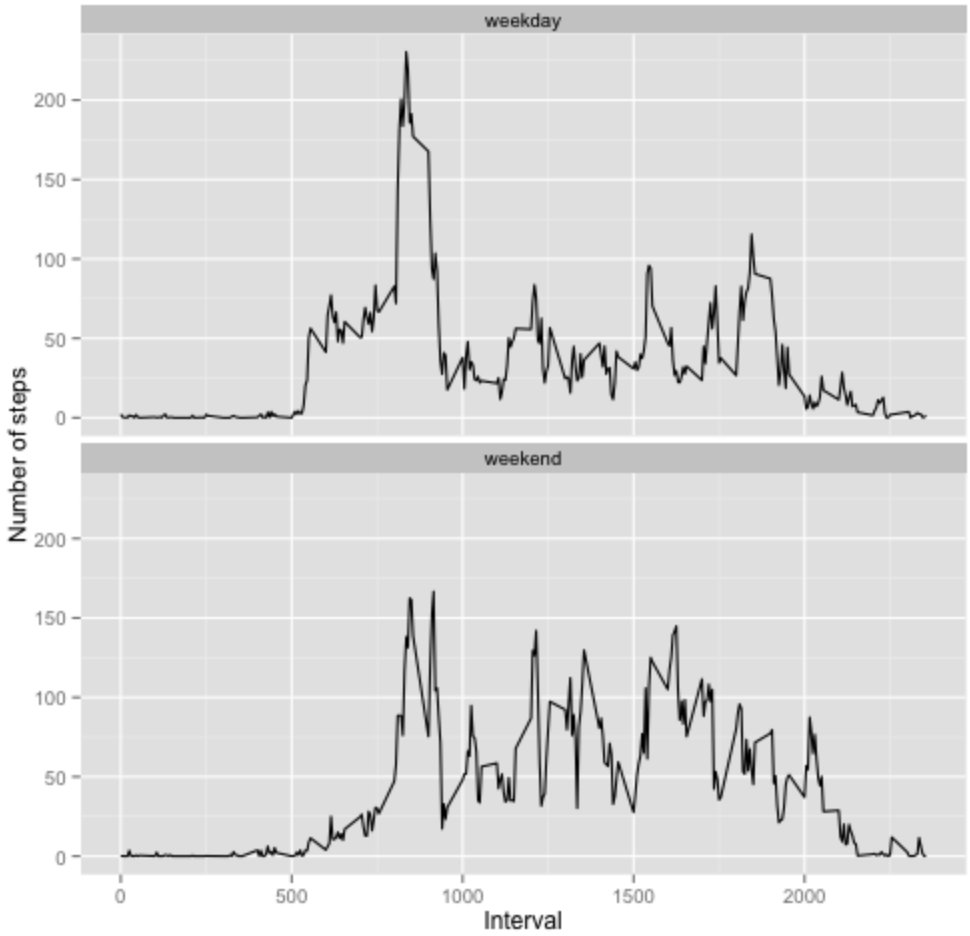
```
# convert date from string to Date class
df$date <- as.Date(df$date, "%Y-%m-%d")

# add a new column indicating day of the week
df$day <- weekdays(df$date)

# add a new column called day type and initialize to weekday
df$day_type <- c("weekday")

# If day is Saturday or Sunday, make day_type as weekend
for (i in 1:nrow(df)){
  if (df$day[i] == "Saturday" || df$day[i] == "Sunday"){
    df$day_type[i] <- "weekend"
  }
}

# convert day_time from character to factor
df$day_type <- as.factor(df$day_type)

# aggregate steps as interval to get average number of steps in an interval
across all days
table_interval_steps_imputed <- aggregate(steps ~ interval+day_type, df,
mean)

# make the panel plot for weekdays and weekends
library(ggplot2)
##
## Attaching package: 'ggplot2'
##
## The following object is masked from 'package:psych':
##
##     %+%
qplot(interval, steps, data=table_interval_steps_imputed, geom=c("line"),
xlab="Interval",
      ylab="Number of steps", main="") + facet_wrap(~ day_type, ncol=1)
```

Finally, we remove all the data frames to free the memory.

```
# remove the data frames to free memory
rm(df, df2, table_date_steps, table_interval_steps, df_NA,
table_date_steps_imputed,
   table_interval_steps_imputed)
```

Finally, we remove all the data frames to free the memory.

```
# remove the data frames to free memory
rm(df, df2, table_date_steps, table_interval_steps, df_NA,
table_date_steps_imputed,
   table_interval_steps_imputed)
```



Finally, we remove all the data frames to free the memory.

```
# remove the data frames to free memory
rm(df, df2, table_date_steps, table_interval_steps, df_NA,
table_date_steps_imputed,
   table_interval_steps_imputed)
```