Reproducible Research: Peer Assessment 1

## Loading and preprocessing the data

# 1. Load the data (i.e. read.csv())

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

# read csv, remove NA values and store it in data.

data <- read.csv("activity.csv", sep=",", header = TRUE, stringsAsFactors = FALSE, na.strings = "NA")

# display the data, a use to sellect a number of data entries from the head is head(data, 10)

` head(data)

## What is mean total number of steps taken per day?

# extract completed cases and store them in datasteps

datasteps <- data[complete.cases(data),]

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

# compute total steps per day and store in total\_steps\_per\_day

total\_steps\_per\_day <- tapply(datastepsdate, sum, na.rm = TRUE)

total\_steps\_per\_day

# 2.hitogram showing the steps per day

hist(total\_steps\_per\_day, col="red", xlab = "Steps per Day", main = "Number of Steps per Day")

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

# mean\_total of steps

mean\_total <- mean(total\_steps\_per\_day)

mean\_total

# [1] 9354.23

# median\_total of steps

median\_total <- median(total\_steps\_per\_day)

median\_total

# [1] 10395

## What is the average daily activity pattern?

data\_pattern <- data[complete.cases(data),] daily\_activity <- tapply(data\_patterninterval, mean, na.rm = TRUE) daily\_activity

# 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)

# plot of average daily activity

plot(names(daily\_activity), daily\_activity, type = "l", xlab = "Interval", ylab = "Average Number of Steps Taken Througout the Days", main = "Average Daily Activity", col = "red")

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

largest\_n\_steps\_interval <- which.max(daily\_activity) names(largest\_n\_steps\_interval)

# [1] "835"

# ------------------------------------------------------------------------------------------

## Imputing missing values

# 1.Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

# number of missing values stores in number\_na

number\_na <- sum(!(complete.cases(data))) number\_na

# [1] 2304

# 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.

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

# asign each NA value the average number of steps for all days as calculated in previous steps

new\_data\_filling\_NA <- data for (i in 1:nrow(new\_data\_filling\_NA)){ if (is.na(new\_data\_filling\_NA$steps[i])){ new\_data\_filling\_NA$steps[i] <- mean(new\_data\_filling\_NAinterval == new\_data\_filling\_NA$interval[i]], na.rm=TRUE) } }

# 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?

total\_steps\_per\_day2 <- tapply(new\_data\_filling\_NAdate, sum) hist(total\_steps\_per\_day2, col = "blue",xlab = "Steps taken per Day", main = "Average number of Steps per DayWith Filled in Missing Values")

# mean\_total of steps

mean\_total2 <- mean(total\_steps\_per\_day2) mean\_total2

# [1] 10766.19

# median\_total of steps

median\_total2 <- median(total\_steps\_per\_day2) median\_total2

# [1] 10766.19

mean\_total2 - mean\_total

# [1] 1411.959

median\_total2 - median\_total

# [1] 371.1887

## 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.

week <- weekdays(strptime(new\_data\_filling\_NA$date, format = "%Y-%m-%d")) for (i in 1:length(week)){ if (week[i] == "Saturday" | week[i] == "Sunday"){ week[i] <- "weekend" } else { week[i] <- "weekday" } } week <- as.factor(week) new\_data\_filling\_NA$week <- week

# 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). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

final\_data <- aggregate(steps ~ interval + week, data = new\_data\_filling\_NA, mean)

library(lattice)

xyplot(steps ~ interval | week, data = final\_data, type = "l", layout = c(1, 2), xlab = "Interval", ylab = "Number of Steps")