**Reproducible Research: Peer Assessment 1**

**Loading and preprocessing the data**

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

dataset <- read.csv("activity.csv", colClasses = "character")

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

steps\_as\_numeric <- as.numeric(dataset$steps)

dataset[,1] <- steps\_as\_numeric

date\_as\_date <- as.Date(dataset$date)

dataset[,2] <- date\_as\_date

interval\_as\_numeric <- as.numeric(dataset$interval)

dataset[,3] <- interval\_as\_numeric

A version of the dataset with NAs removed

dataset\_noNAs <- dataset[complete.cases(dataset),]

set up the margins and graphing parameters

par(mar=c(3,4,1,1))

par(mfrow=c(1,1))

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

Consider the dataset after removing all the NAs: dataset\_noNAs

Make a histogram of the total number of steps taken each day

total\_steps\_per\_day <- sapply(split(dataset\_noNAs$steps, dataset\_noNAs$date), sum)

hist(total\_steps\_per\_day, col="Red")

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

Calculate the average/mean steps taken per day

average\_steps\_per\_day <- sapply(split(dataset\_noNAs$steps, dataset\_noNAs$date), mean)

average\_steps\_per\_day

Calculate the median steps taken per day

median\_steps\_per\_day <- sapply(split(dataset\_noNAs$steps, dataset\_noNAs$date), median)

median\_steps\_per\_day

**What is the average daily activity pattern?**

Consider the dataset after removing all the NAs: dataset\_noNAs

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)

Calculation of the x-(average\_steps\_per\_interval) and y- (intervals) values for the plot

average\_steps\_per\_interval <- sapply(split(dataset\_noNAs$steps, dataset\_noNAs$interval), mean)

intervals\_list <- split(dataset\_noNAs$interval, dataset\_noNAs$interval)

intervals <- names(intervals\_list)

The plot of average steps taken over every interval in the dataset across all days

plot(intervals, average\_steps\_per\_interval, type="l", xlab="Intervals", ylab="Average steps", col="Blue")

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

maximum\_steps\_interval <- max(average\_steps\_per\_interval)

index\_of\_maximum\_steps\_interval <- match(maximum\_steps\_interval, average\_steps\_per\_interval)

maximum\_steps\_5\_minute\_interval <- intervals[index\_of\_maximum\_steps\_interval]

maximum\_steps\_5\_minute\_interval

**Imputing missing values**

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

total\_number\_of\_missing\_values <- sum(is.na(dataset))

total\_number\_of\_missing\_values

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. My strategy: replace all the NAs with the mean for the corresponding 5-minute interval in that row

Create a copy of the original dataset that will have the missing data (i.e., the NA values) filled in.

dataset\_NAsfilled <- read.csv("activity.csv", colClasses = "character")

Process/transform the data into a format suitable for your analysis

steps\_as\_numeric <- as.numeric(dataset\_NAsfilled$steps)

dataset\_NAsfilled[,1] <- steps\_as\_numeric

date\_as\_date <- as.Date(dataset\_NAsfilled$date)

dataset\_NAsfilled[,2] <- date\_as\_date

interval\_as\_numeric <- as.numeric(dataset\_NAsfilled$interval)

dataset\_NAsfilled[,3] <- interval\_as\_numeric

Process/transform the dataset by replacing the NA values with the mean for the corresponding 5-minute interval

for (i in seq(intervals)) {

interval <- as.numeric(intervals[i])

for (row in seq(dataset\_NAsfilled$steps)) {

if (is.na(dataset\_NAsfilled$steps[row]) && dataset\_NAsfilled$interval[row] == interval) {

dataset\_NAsfilled$steps[row] = average\_steps\_per\_interval[i]

}

}

}

Make a histogram of the total number of steps taken each day

total\_steps\_per\_day\_NAsfilled <- sapply(split(dataset\_NAsfilled$steps, dataset\_NAsfilled$date), sum)

hist(total\_steps\_per\_day\_NAsfilled, col="Green")

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

average\_steps\_per\_day\_NAsfilled <- sapply(split(dataset\_NAsfilled$steps, dataset\_NAsfilled$date), mean)

average\_steps\_per\_day\_NAsfilled

median\_steps\_NAsfilled <- sapply(split(dataset\_NAsfilled$steps, dataset\_NAsfilled$date), median)

median\_steps\_NAsfilled

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?

Yes, the values differ from the estimates from the first part of the assignment. The frequency of steps has gone up in every 5-minute interval category as seen in the green histogram. Moreover, the replacement of NA values with the corresponding row's 5-minute interval mean has given us higher mean and median values. Please note that a lot of the median values for both parts of the assignment are 0 since 0 appears a lot in the given data itself.

**Are there differences in activity patterns between weekdays and weekends?**

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.

weekday <- vector(mode="character")

day\_of\_the\_week <- weekdays(dataset\_NAsfilled$date)

for (i in seq(dataset\_NAsfilled$steps)) {

#print(i)

day <- day\_of\_the\_week[i]

if (day == "Saturday" || day == "Sunday") {

weekday <- append(weekday, "weekend")

}

else {

weekday <- append(weekday, "weekday")

}

}

Modify the data frame with the weekday/weekend column/factor

dataset\_modified <- cbind(dataset\_NAsfilled, weekday)

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).

Weekday data

weekday\_dataframe <- dataset\_modified[dataset\_modified$weekday == "weekday",]

average\_steps\_weekday <- sapply(split(weekday\_dataframe$steps, weekday\_dataframe$interval), mean)

interval1\_list <- split(weekday\_dataframe$interval, weekday\_dataframe$interval)

interval1 <- names(interval1\_list)

Weekend data

weekend\_dataframe <- dataset\_modified[dataset\_modified$weekday == "weekend",]

average\_steps\_weekend <- sapply(split(weekend\_dataframe$steps, weekend\_dataframe$interval), mean)

interval2\_list <- split(weekend\_dataframe$interval, weekend\_dataframe$interval)

interval2 <- names(interval2\_list)

Set up the margins and plot

par(mar=c(4,4,2,1))

par(mfrow=c(2,1))

plot(interval2, average\_steps\_weekend, type="l", xlab="Intervals", ylab="Number of average steps", col="Blue", main="weekend")

plot(interval1, average\_steps\_weekday, type="l", xlab="Intervals", ylab="Number of average steps", col="Blue", main="weekday")