RepData_PeerAssessment1

Loading and preprocessing the data 1. Load the data

```
setwd("/Users/chen/Downloads")
library(plyr)
library(ggplot2)
activity <- read.csv("activity.csv")
```

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

```
activity$day <- weekdays(as.Date(activity$date))
activity$DateTime<- as.POSIXct(activity$date, format="%Y-%m-%d")
clean <- activity[!is.na(activity$steps),]
```
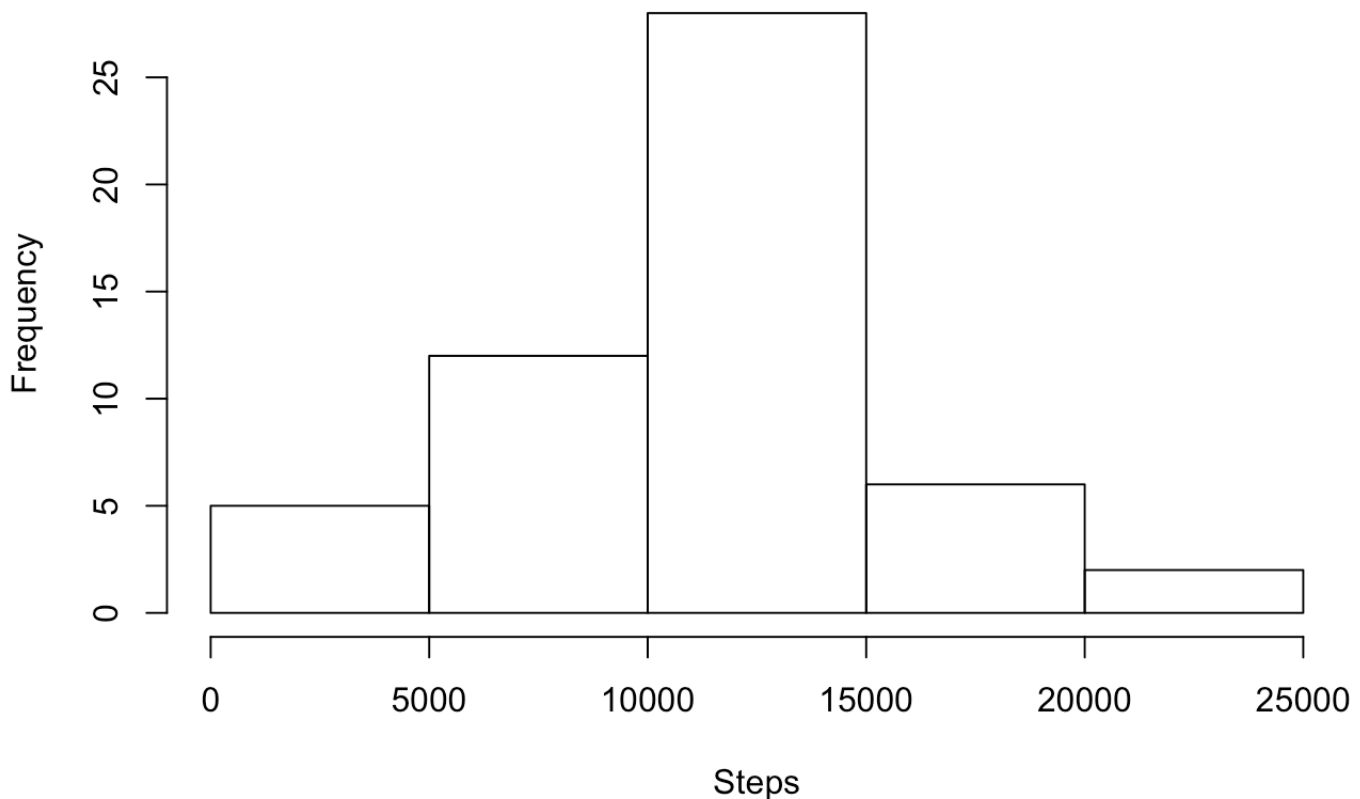
What is mean total number of steps taken per day? 1. Calculate the total number of steps taken per day

```
sumTable <- aggregate(activity$steps ~ activity$date, FUN=sum, )
colnames(sumTable)<- c("Date", "Steps")
```

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

```
hist(sumTable$Steps, breaks=5, xlab="Steps", main = "Total Steps per Day")
```

Total Steps per Day

3.

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

```
as.integer(mean(sumTable$Steps))
```

```
## [1] 10766
```

```
as.integer(median(sumTable$Steps))
```

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

What is the average daily activity pattern? 1. Make a time series plot

```
clean <- activity[!is.na(activity$steps),]
intervalTable <- ddply(clean, .(interval), summarize, Avg = mean(steps))
p <- ggplot(intervalTable, aes(x=interval, y=Avg), xlab = "Interval", ylab="Average N
umber of Steps")
p + geom_line()+xlab("Interval")+ylab("Average Number of Steps")+ggtitle("Average Num
ber of Steps per Interval")
```

Average Number of Steps per Interval
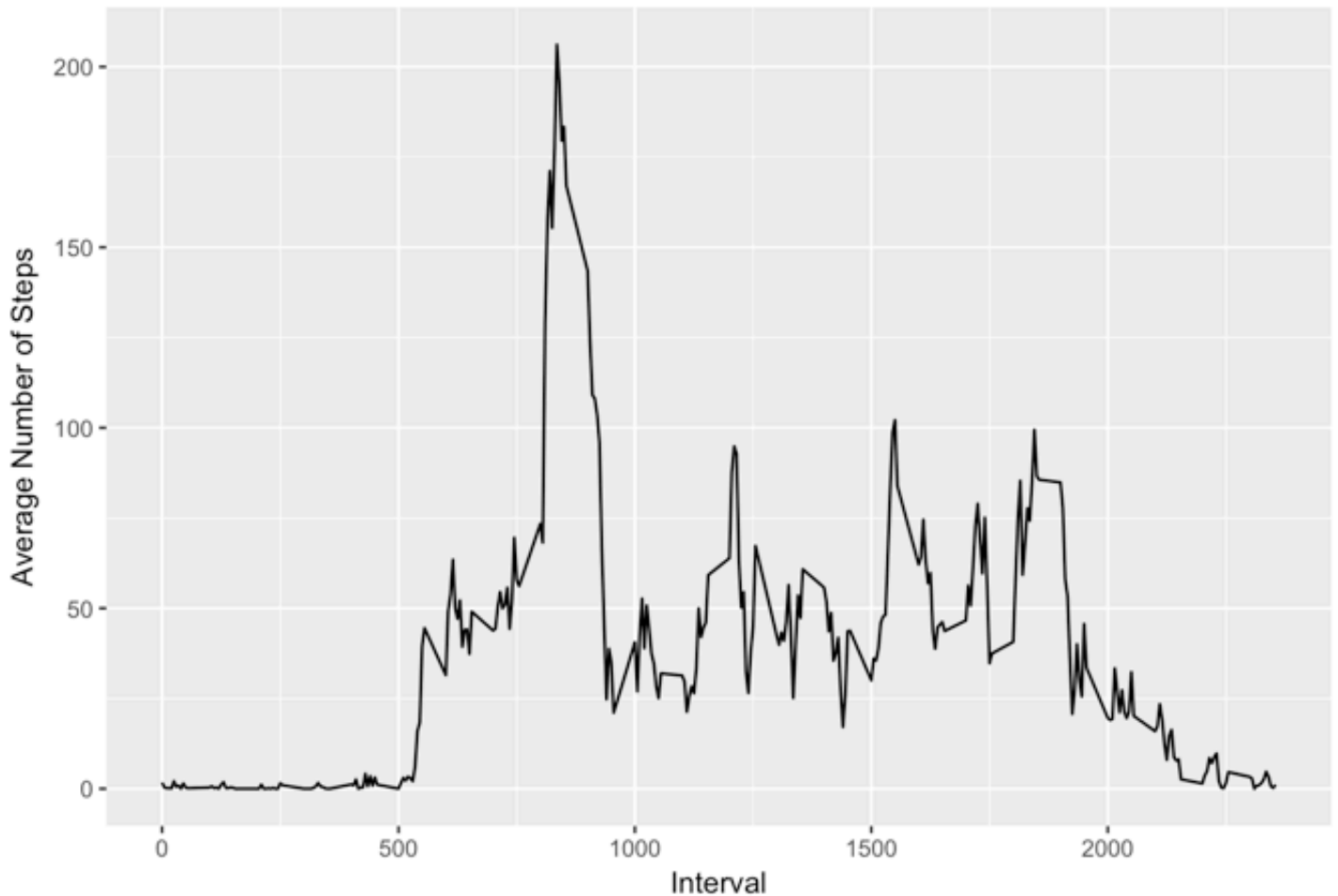
2.

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

```
maxSteps <- max(intervalTable$Avg)
intervalTable[intervalTable$Avg==maxSteps,1]
```

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

Imputing missing values 1. Calculate and report the total number of missing values in the dataset

```
nrow(activity[is.na(activity$steps),])
```

```
## [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.

```
avgTable <- ddply(clean, .(interval, day), summarize, Avg = mean(steps))
nadata<- activity[is.na(activity$steps),]
newdata<-merge(nadata, avgTable, by=c("interval", "day"))
```

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

```
newdata2<- newdata[,c(6,4,1,2,5)]
colnames(newdata2)<- c("steps", "date", "interval", "day", "DateTime")
##Merge the NA averages and non NA data together
mergeData <- rbind(clean, newdata2)
```

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?

```
sumTable2 <- aggregate(mergeData$steps ~ mergeData$date, FUN=sum, )
colnames(sumTable2)<- c("Date", "Steps")
as.integer(mean(sumTable2$Steps))
```
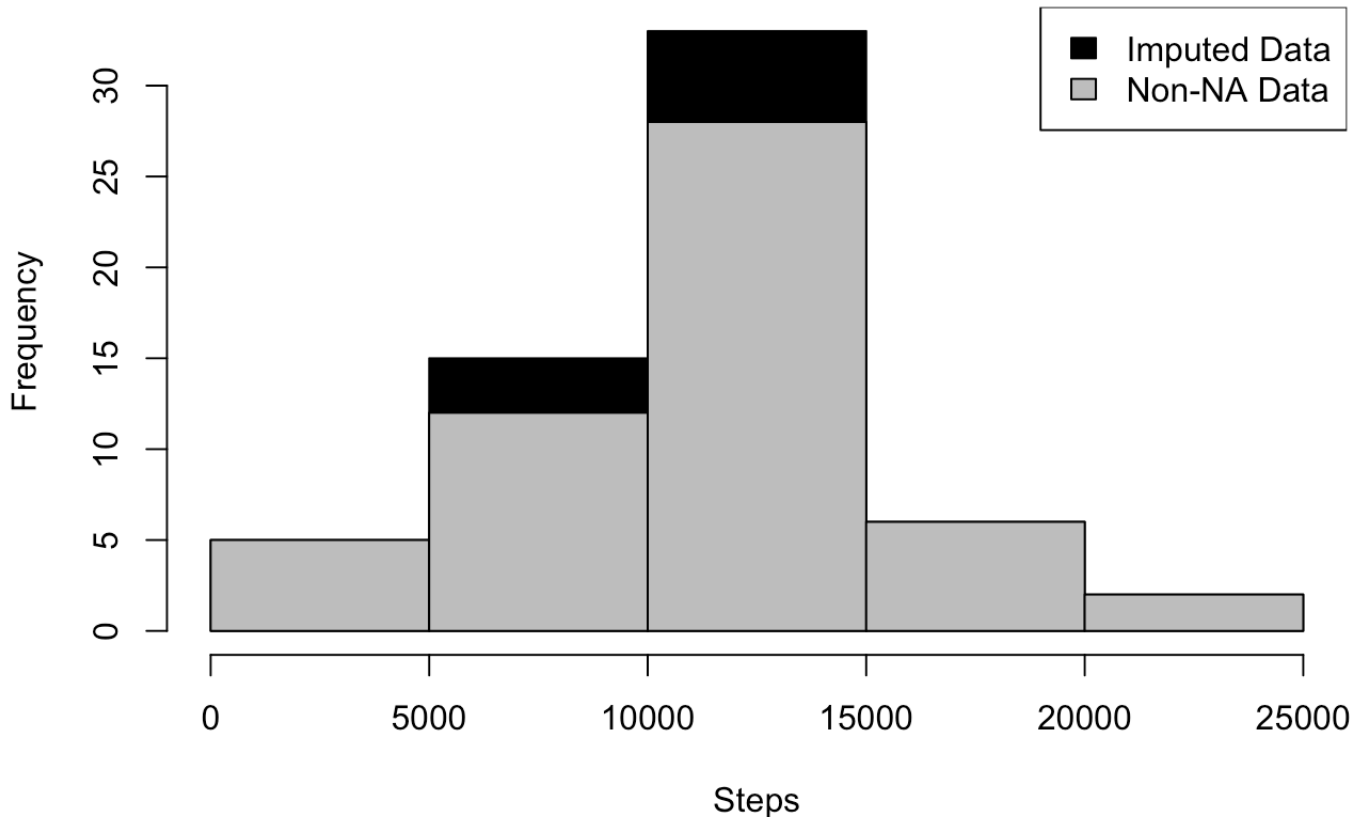
```
## [1] 10821
```

```
as.integer(median(sumTable2$Steps))
```

```
## [1] 11015
```

```
hist(sumTable2$Steps, breaks=5, xlab="Steps", main = "Total Steps per Day with NAs Fi
xed", col="Black")
hist(sumTable$Steps, breaks=5, xlab="Steps", main = "Total Steps per Day with NAs Fix
ed", col="Grey", add=T)
legend("topright", c("Imputed Data", "Non-NA Data"), fill=c("black", "grey") )
```

Total Steps per Day with NAs Fixed

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.

```
mergeData$DayCategory <- ifelse(mergeData$day %in% c("Saturday", "Sunday"), "Weekend"
, "Weekday")
```

2. Make a panel plot containing a time series plot 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.

```
library(lattice)
intervalTable2 <- ddply(mergeData, .(interval, DayCategory), summarize, Avg = mean(st
eps))

##Plot data in a panel plot
xyplot(Avg~interval|DayCategory, data=intervalTable2, type="l",  layout = c(1,2),
       main="Average Steps per Interval Based on Type of Day",
       ylab="Average Number of Steps", xlab="Interval")
```

Average Steps per Interval Based on Type of Day