Reproducible Research, Project 1

Data loading portion:

## Load Packages  
library(dplyr)

##   
## 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

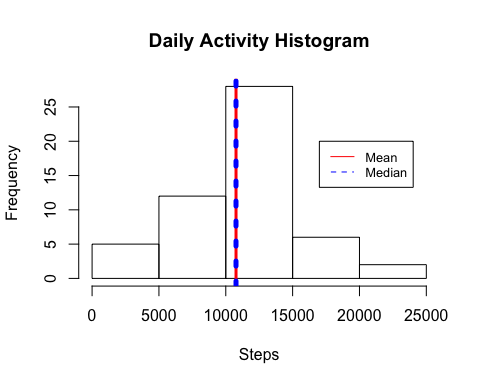
library(ggplot2)  
  
## Load Data  
Raw <- read.csv("./activity.csv", header = TRUE)

Initial sort of Data

## Separate Complete and Incomplete Cases  
Complete <- Raw[complete.cases(Raw),]  
Incomplete <- Raw[!complete.cases(Raw),]

For first part of the assignment:

## Summary of Daily Activity by Date  
Daily <- Complete %>% group\_by(date) %>% summarize("Activity" = sum(steps))  
  
## Produce Histogram, Add Labels, Title, & Legend   
hist(Daily$Activity, xlab = "Steps", main = "Daily Activity Histogram")  
abline(v = mean(Daily$Activity), lty = 1, col = "red", lwd = 3)  
abline(v = median(Daily$Activity), lty = 3, col = "blue", lwd = 5)  
legend(17000, 20, legend=c("Mean", "Median"), col=c("red", "blue"), lty = 1:3, cex = .8)



## Get Mean and Median of Daily Activity by Date  
mean(Daily$Activity)

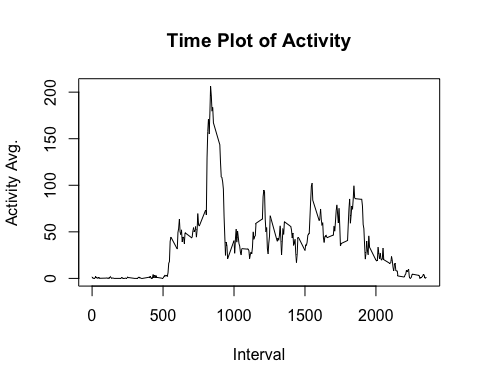
## [1] 10766.19

median(Daily$Activity)

## [1] 10765

Second part of assignment:

## Summary of Activity by Interval  
TimePlot <- Complete %>% group\_by(interval) %>% summarize("Avg" = mean(steps))  
  
## Produce Line Plot  
plot(TimePlot$interval, TimePlot$Avg, type = "l", xlab = "Interval", ylab = "Activity Avg.", main = "Time Plot of Activity")

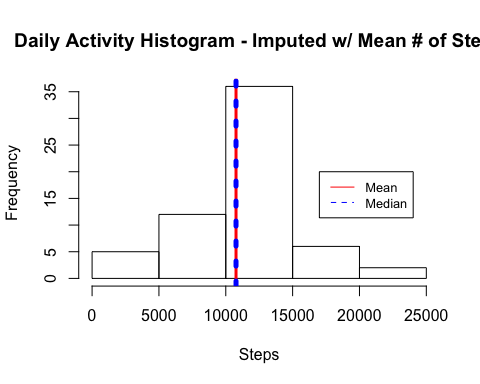


## Find Interval with Highest Activity  
TimePlot[which.max(TimePlot$Avg), 1]

## # A tibble: 1 x 1  
## interval  
## <int>  
## 1 835

Imputing Data and recreating graphs (note I’ve used mean of initial data) For unknown reason, knitr is not able to produce mean / median lines as RStudio does (see png file).

## Impute data with mean of steps from initial data set & create a new data frame   
Incomplete$steps[is.na(Incomplete$steps)] = mean(Raw$steps, na.rm = TRUE)  
ImpMerge <- rbind(Incomplete, Complete)  
  
## Summarize Imputed Data  
ImpDaily <- ImpMerge %>% group\_by(date) %>% summarize("Activity" = sum(steps))  
ImpDaily$Activity[is.na(ImpDaily$Activity)] <- mean(Daily$Activity)  
  
## Histogram of Imputed Data  
hist(ImpDaily$Activity, xlab = "Steps", main = "Daily Activity Histogram - Imputed w/ Mean # of Steps")  
abline(v = mean(ImpDaily$Activity), lty = 1, col = "red", lwd = 3)  
abline(v = median(ImpDaily$Activity), lty = 3, col = "blue", lwd = 5)  
legend(17000, 20, legend=c("Mean", "Median"), col=c("red", "blue"), lty = 1:3, cex = .8)



Adding days and classifying as Weekday or Weekend

## Add Weekday Info & Classify as Weekday or Weekend  
ImpMerge$Day <- weekdays(as.Date(ImpMerge$date, format = "%Y-%m-%d"))  
ImpMerge$DayClass <- ifelse(ImpMerge$Day == "Saturday" | ImpMerge$Day == "Sunday", "Weekend", "Weekday")

Last part of assignment:

## Summarize Modified Data  
ImpTimePlot <- ImpMerge %>% group\_by(interval, DayClass) %>% summarize("Avg" = mean(steps))

Plot for Activity Weekend vs. Weekday

a <- ggplot(ImpTimePlot, aes(interval, Avg, color = DayClass))  
a + geom\_line() + ggtitle("Activity Time Plot, Weekday v Weekend") + xlab("Intervals") + ylab("Activity")

