data<- read.csv("activity.csv")

> data$date<- as.Date(data$date)

>

> #mean total number steps per day

> stepsbyday<- tapply(data$steps, data$date, sum, na.rm=TRUE)

> library(ggplot2)

> qplot(stepsbyday, xlab="No. of Steps Taken Each Day", ylab="Total Frequency", binwidth=500)

>

> #mean and median steps per day

> meanbyday<- mean(stepsbyday)

> medianbyday<- median(stepsbyday)

>

> medianbyday

[1] 10505

> meanbyday

[1] 8047.458

>

> #5 min intervals aver across all days

> avg<- tapply(data$steps, data$interval, mean, na.rm=TRUE)

> #time series plot 5 min intervals

> plot(names(avg), avg, xlab="5-min interval", type="l", ylab="Average no. of steps")

>

> #interval with max number steps

> maxavg<- max(avg)

> maxinterval<- as.numeric(names(avg)[which(avg==max(avg))])

> maxinterval

[1] 835

>

> #number missing values in dataset

> totalna<- sum(is.na(data$steps))

> totalna

[1] 2304

> #impute missing data

> imputedata<- data

> #use mean for all missing data

> imputedata$steps[which(is.na(data$steps))]<- as.vector(avg[as.character(data[which(is.na(data$steps)),3])])

> #number steps per day imputed data

> stepseachday<- tapply(imputedata$steps, imputedata$date, sum, na.rm=TRUE)

> #histogram steps per day imputed data

> qplot(stepseachday, xlab="No. of Steps Taken Each Day", ylab="Total Frequency", binwidth=500)

>

> #mean and median total steps per day with imputed data

> meanEachDayImputed<- mean(stepseachday)

> medianEachDayImputed<- median(stepseachday)

> meanEachDayImputed

[1] 10739.01

> medianEachDayImputed

[1] 10766.19

> ##values are higher with imputed data

>

> #difference in weekend and weekdays

> imputedata$dayType<- ifelse(as.POSIXlt(imputedata$date)$wday %in% c(0,6), "weekends","weekdays")

> #aggregate data by interval and type

> aggregateData<- aggregate(steps ~ interval + dayType, data=imputedata, mean)

> #time series plot weekends and weekdays

> ggplot(aggregateData, aes(interval, steps)) + geom\_line() + facet\_grid(dayType ~ .) + xlab("5-minute interval") + ylab("avarage number of steps")

>