# Course Project 1 Reproducible Research

### Code for reading in the dataset and/or processing the data

tabla<-read.csv("./activity.csv",header=TRUE)  
names(tabla)

## [1] "steps" "date" "interval"

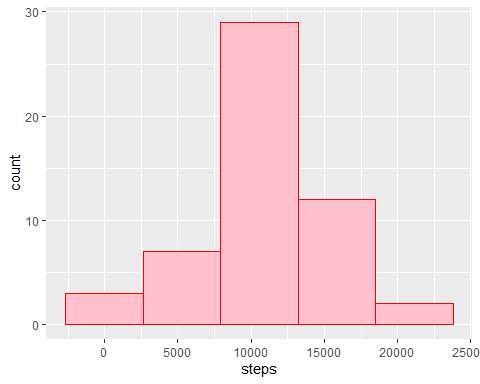
str(tabla)

## 'data.frame': 17568 obs. of 3 variables:  
## $ steps : int NA NA NA NA NA NA NA NA NA NA ...  
## $ date : chr "2012-10-01" "2012-10-01" "2012-10-01" "2012-10-01" ...  
## $ interval: int 0 5 10 15 20 25 30 35 40 45 ...

#Tabla que no tiene datos faltantes  
tabla1<-tabla[complete.cases(tabla),]

### Histogram of the total number of steps taken each day

library(ggplot2)  
totalpasos<- aggregate(steps ~ date, tabla1, sum)  
ggplot(totalpasos,aes(x=steps))+geom\_histogram(bins=5,fill="pink",color="red")



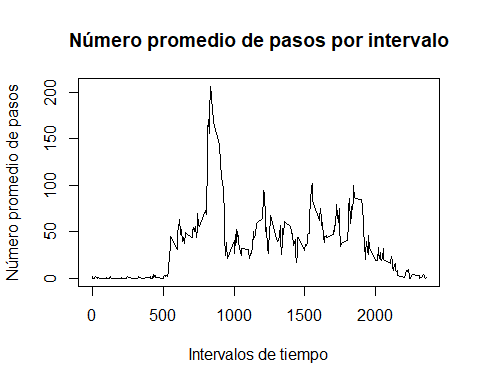
### Mean and median number of steps taken each day

summary(totalpasos$steps)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 41 8841 10765 10766 13294 21194

### Time series plot of the average number of steps taken

avg\_steps\_interval<-aggregate(steps~interval,tabla1,mean)  
avg\_steps\_day<-aggregate(steps~date,tabla1,mean)  
plot(avg\_steps\_interval$interval, avg\_steps\_interval$steps, type='l', col=1, main="Número promedio de pasos por intervalo", xlab="Intervalos de tiempo", ylab="Número promedio de pasos")



### The 5-minute interval that, on average, contains the maximum number of steps

Intervalo<-which.max(avg\_steps\_interval$steps)  
Pasos<-avg\_steps\_interval[Intervalo,]$steps

El intervalo de 5 minutos que contiene el máximo número de pasos es 104 con 206.1698113.

### Code to describe and show a strategy for imputing missing data

#se buscan cuantos datos faltantes hay  
datos\_faltantes<-tabla[!complete.cases(tabla),]  
nrow(datos\_faltantes)

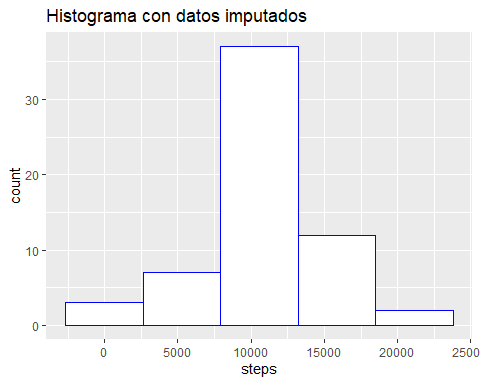
## [1] 2304

#Se ingresan valores promedio en los datos faltantes  
for (i in 1:nrow(tabla)) {  
 if(is.na(tabla$steps[i])) {  
 val <- avg\_steps\_interval$steps[which(avg\_steps\_interval$interval == tabla$interval[i])]  
 tabla$steps[i] <- val   
 }  
}  
steps\_day\_impute <- aggregate(steps ~ date, tabla, sum)  
str(steps\_day\_impute)

## 'data.frame': 61 obs. of 2 variables:  
## $ date : chr "2012-10-01" "2012-10-02" "2012-10-03" "2012-10-04" ...  
## $ steps: num 10766 126 11352 12116 13294 ...

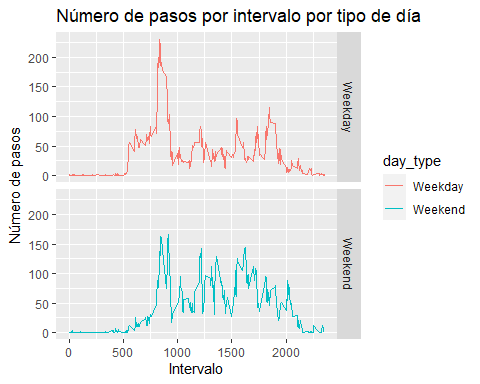
### Histogram of the total number of steps taken each day after missing values are imputed

ggplot(steps\_day\_impute,aes(x=steps))+geom\_histogram(bins=5,fill="white",color="blue")+ggtitle("Histograma con datos imputados")



### Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

#Validamos si es un día laboral o de trabajo  
week\_day <- function(date\_val) {  
 wd <- weekdays(as.Date(date\_val, '%Y-%m-%d'))  
 if (!(wd == 'sábado' || wd == 'domingo')) {  
 x <- 'Weekday'  
 } else {  
 x <- 'Weekend'  
 }  
 x  
}  
#Se grafica por tipo de día  
tabla$day\_type <- as.factor(sapply(tabla$date, week\_day))  
#Se revisa a qué tipo de día pertenece  
steps\_day\_impute <- aggregate(steps ~ interval+day\_type, tabla, mean)  
plt <- ggplot(steps\_day\_impute, aes(interval, steps)) +  
 geom\_line(stat = "identity", aes(colour = day\_type)) +  
 facet\_grid(day\_type ~ ., scales="fixed", space="fixed") +  
 labs(x="Intervalo", y=expression("Número de pasos")) +  
 ggtitle("Número de pasos por intervalo por tipo de día")  
print(plt)



All of the R code needed to reproduce the results (numbers, plots, etc.) in the report