## Cargando Biblioteca

```
library(ggplot2)
```

## Cargar y preprocesar los datos

### Cargar los datos

```
fullData <- read.csv("activity.csv")
```

## Procesar/transformar los datos

```
fullData$date <- as.Date(fullData$date, "%Y-%m-%d")
```

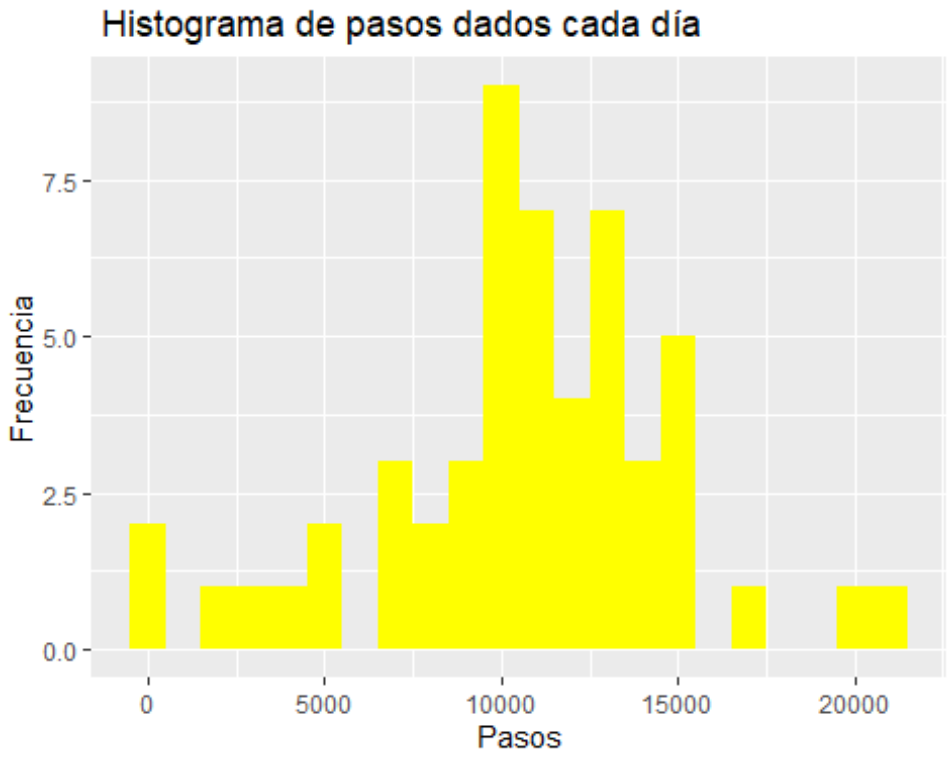
## A. ¿Cuál es el número total medio de pasos dados por día?

### Calcular el total de pasos por día

```
stepsPerDay <- aggregate(steps ~ date, fullData, FUN = sum)
```

### Haz un histograma del número total de pasos dados cada día.

```
library(ggplot2)

g <- ggplot (stepsPerDay, aes (x = steps))

g + geom_histogram(fill = "yellow", binwidth = 1000) +
    labs(title = " Histograma de pasos dados cada día ", x = "Pasos", y =
"Frecuencia")
```




### Calcule e informe la media y la mediana del número total de pasos dados por día

```
stepsMean <- mean(stepsPerDay$steps, na.rm=TRUE)
stepsMean

## [1] 10766.19

stepsMedian <- median(stepsPerDay$steps, na.rm=TRUE)
stepsMedian

## [1] 10765
```

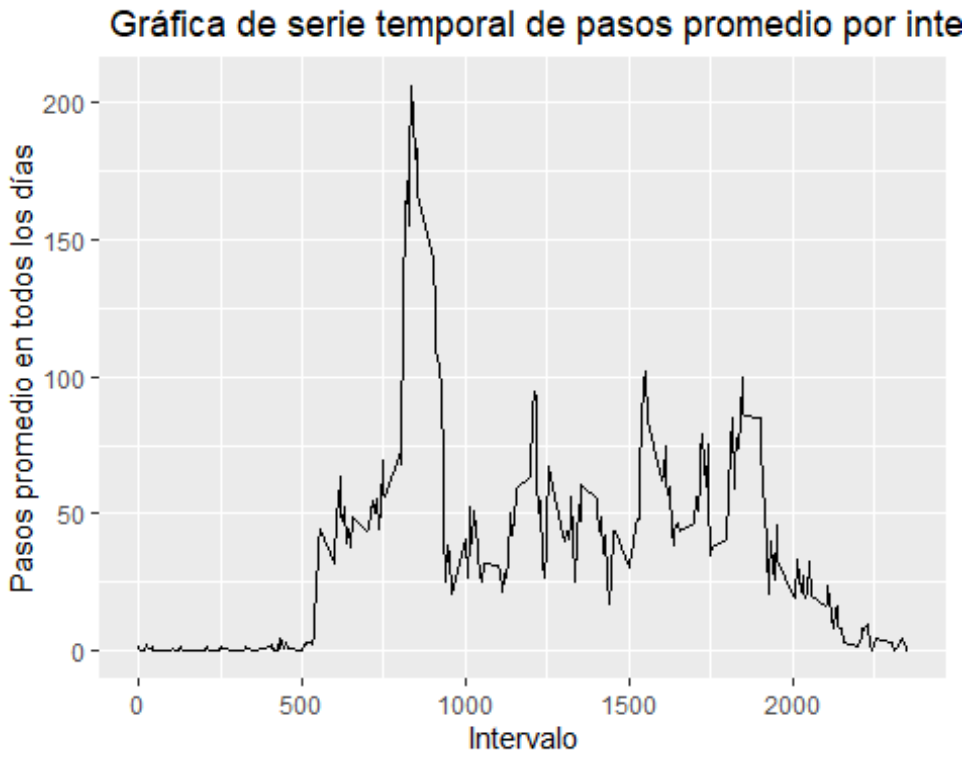
### La media y la mediana del número total de pasos dados por día son 10766.19 y 10765 respectivamente

## B. ¿Cuál es el patrón de actividad diario promedio?

### Haga un gráfico de serie de tiempo (es decir, tipo = "l") del intervalo de 5 minutos (eje x) y el número promedio de pasos dados, promediado en todos los días (eje y)

```
stepsPerInterval <- aggregate(steps ~ interval, fullData, mean)

# Cree una gráfica de serie temporal del número promedio de pasos por
intervalo, anote la gráfica
```

```
h <- ggplot (stepsPerInterval, aes(x=interval, y=steps))
h + geom_line()+ labs(title = " Gráfica de serie temporal de pasos promedio
por intervalo", x = "Intervalo", y = "Pasos promedio en todos los días")
```




### ¿Qué intervalo de 5 minutos, en promedio de todos los días del conjunto de datos, contiene la cantidad máxima de pasos?

```
maxInterval <- stepsPerInterval[which.max(stepsPerInterval$steps), ]
maxInterval

##     interval    steps
## 104      835 206.1698
```

## C. Imputación del valor faltante

### Calcule e informe el número total de valores faltantes en el conjunto de datos (es decir, el número total de filas con NA)

```
noMissingValue <- nrow(fullData[is.na(fullData$steps),])
noMissingValue

## [1] 2304
```

409 / 5.000

## Diseñe una estrategia para completar todos los valores que faltan en el conjunto de datos. La estrategia no necesita ser sofisticada. Por ejemplo, podría usar la media/mediana de ese día, o la media de ese intervalo de 5 minutos, etc.

## Mi estrategia para completar los valores faltantes (NA) es sustituir los valores faltantes (pasos) con el número promedio de pasos según el intervalo de 5 minutos y el día de la semana.

```
fullData1 <- read.csv("activity.csv", header=TRUE,sep=",")

# Cree una variable/columna con el nombre de los días de la semana
fullData1$day <- weekdays(as.Date(fullData1$date))

# crear un número promedio de pasos por intervalo de 5 minutos y día
stepsAvg1 <- aggregate(steps ~ interval + day, fullData1, mean)

# Cree un conjunto de datos con todas las NA para la sustitución
nadata <- fullData1 [is.na(fullData1$steps),]

# Combine el conjunto de datos NA con los pasos promedio basados en
intervalos de 5 minutos + días de la semana, para sustituciones
newdata1 <- merge(nadata, stepsAvg1, by=c("interval", "day"))
```

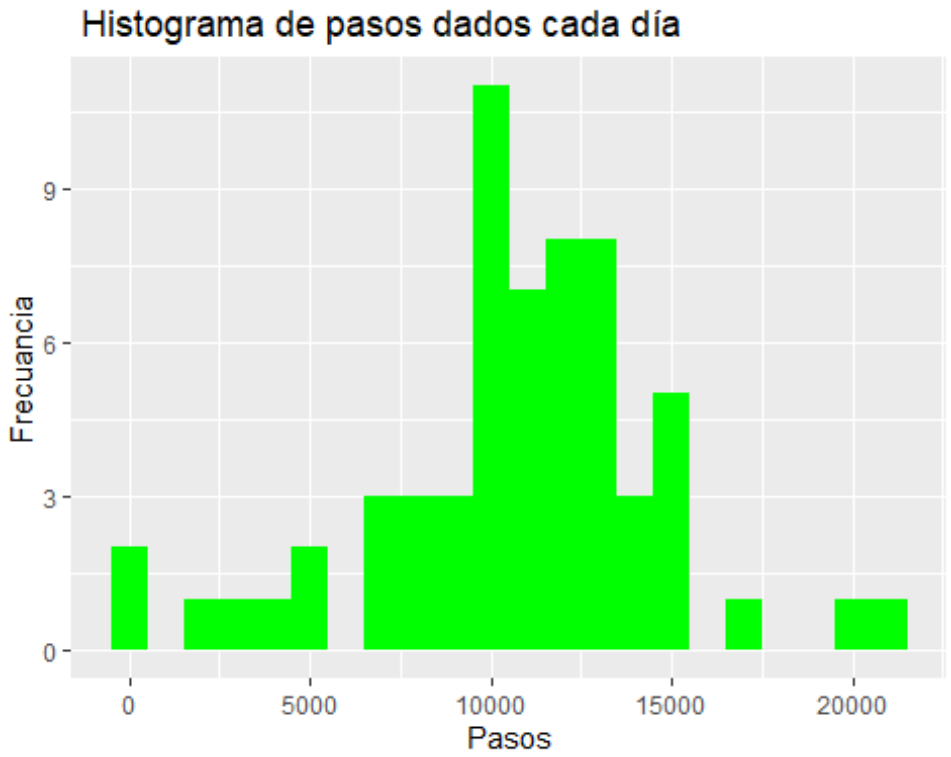
## Cree un nuevo conjunto de datos que sea igual al conjunto de datos original pero con los datos faltantes completados.

```
cleanData <- fullData1 [!is.na(fullData1$steps),]

#Vuelva a ordenar los nuevos datos sustituidos en el mismo formato que el
conjunto de datos limpio (omita la columna NA que se sustituirá por los pasos
promedio en función del intervalo de 5 minutos + día)
newdata2 <- newdata1[,c(5,4,1,2)]
colnames(newdata2) <- c("steps", "date", "interval", "day")

# Combinar los nuevos datos promedio (NA) con el conjunto de datos sin NA
mergeData <- rbind (cleanData, newdata2)
```

## Haga un histograma del número total de pasos dados cada día y Calcule e informe la media y la mediana del número total de pasos dados por día. ¿Estos valores difieren de las estimaciones de la primera parte de la tarea? ¿Cuál es el impacto de imputar los datos que faltan en las estimaciones del número total diario de pasos?

```
# Calcule el total de pasos por día en los datos combinados
stepsPerDayFill <- aggregate(steps ~ date, mergeData, FUN = sum)

# Crear el histograma
g1 <- ggplot (stepsPerDayFill, aes (x = steps))
g1 + geom_histogram(fill = "green", binwidth = 1000) +
```

```
    labs(title = " Histograma de pasos dados cada día ", x = "Pasos", y =
"Frecuancia")
```




```
stepsMeanFill <- mean(stepsPerDayFill$steps, na.rm=TRUE)
stepsMeanFill

## [1] 10821.21

# Mediana del total de pasos con datos imputados
stepsMedianFill <- median(stepsPerDayFill$steps, na.rm=TRUE)
stepsMedianFill

## [1] 11015
```

**La nueva media de los datos imputados es 1,082121^{4} pasos en comparación con la media anterior de 1,0766189^{4} pasos. Eso crea una diferencia de 55.0209226 pasos en promedio por día.**

**La nueva mediana de los datos imputados es 1,1015^{4} pasos en comparación con la mediana anterior de 10765 pasos. Eso crea una diferencia de 250 pasos para la mediana.**

**Sin embargo, la forma general de la distribución no ha cambiado.**

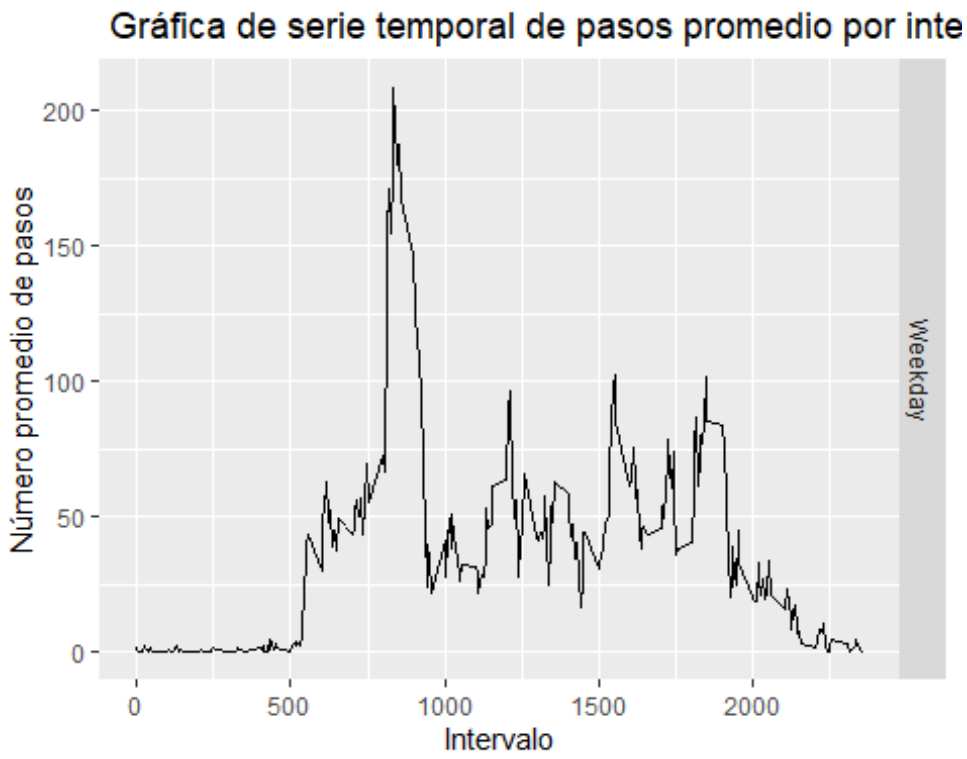
## D. ¿Existen diferencias en los patrones de actividad entre los días de semana y los fines de semana?

**Cree una nueva variable de factor en el conjunto de datos con dos niveles: "día de la semana" y "fin de semana", que indica si una fecha determinada es un día de la semana o un día de fin de semana.**

```
#cree una nueva variable/columna que indique el día de la semana o el fin de semana
mergeData$DayType <- ifelse(mergeData$day %in% c("Saturday", "Sunday"), "Weekend", "Weekday")
```

**Haz un gráfico de panel que contenga un gráfico de series de tiempo (es decir, tipo = "l") del intervalo de 5 minutos (eje x) y el número promedio de pasos dados, promediados entre todos los días de la semana o los días de fin de semana (eje y). Consulte el archivo README en el repositorio de GitHub para ver un ejemplo de cómo debería verse este gráfico usando datos simulados.**

```
# crear una tabla con pasos promedio por intervalo de tiempo entre los días de semana o los días de fin de semana
stepsPerIntervalDT <- aggregate(steps ~ interval+DayType, mergeData, FUN = mean)
# Haz el diagrama de panel
j <- ggplot (stepsPerIntervalDT, aes(x=interval, y=steps))
j + geom_line()+ labs(title = " Gráfica de serie temporal de pasos promedio por intervalo: días de semana frente a fines de semana", x = "Intervalo", y = "Número promedio de pasos") + facet_grid(DayType ~ .)
```


Gráfica de serie temporal de pasos promedio por inte
Número promedio de pasos
200
150
100
50
0
0
500
1000
1500
2000
Intervalo
Weekday