

Reproducible Research: Peer Assessment 1

Loading and preprocessing the data

###1. Carregamento **ler tabela**

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

download das bibliotecas

```
library(ggplot2)
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
```

Definição do padrão de horario

```
Sys.setlocale("LC_TIME", "English")
```

```
## [1] "English_United States.1252"
```

###2. Preprocessamento **Dados iniciais**

```
str(atividades)
```

```
## '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 ...
```

Detalhamento dos dados: “steps”: Numero de passos dados durante 5 minutos, inteiro, contam vazios “date”: Data do registro do evento, fator com 61 níveis “interval”: Intervalo ocorrido, inteiro

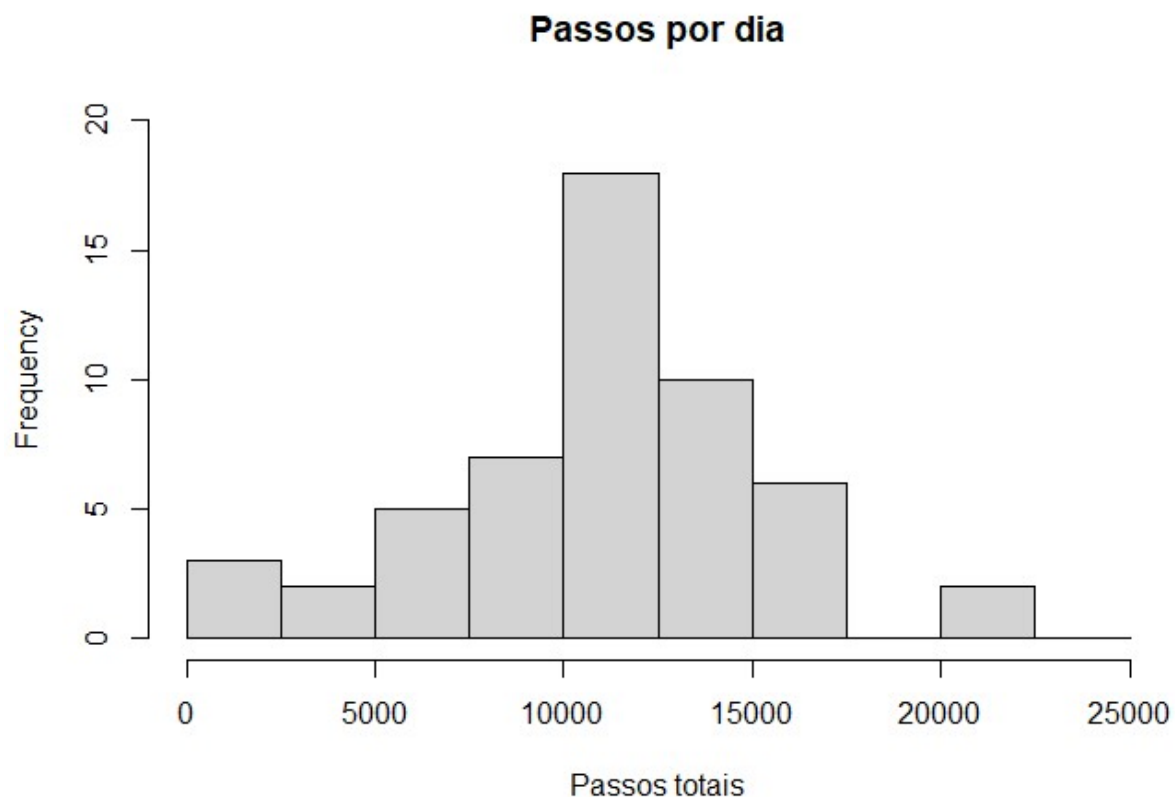
=====

What is mean total number of steps taken per day? ### 1. Criar variavel

```
PassosDiarios <- aggregate(atividades$steps, list(atividades$date), sum)
colnames(PassosDiarios) <- c("Date", "Steps")
```

2. Plotar grafico

```
hist(PassosDiarios$Steps, main = "Passos por dia", xlab = "Passos totais", ylim = c(0,20), breaks = seq(0,25000, by=2500))
```



3. Media

```
mean(PassosDiarios$Steps, na.rm=TRUE)
```

```
## [1] 10766.19
```

4. Mediana

```
median(PassosDiarios$Steps, na.rm=TRUE)
```

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

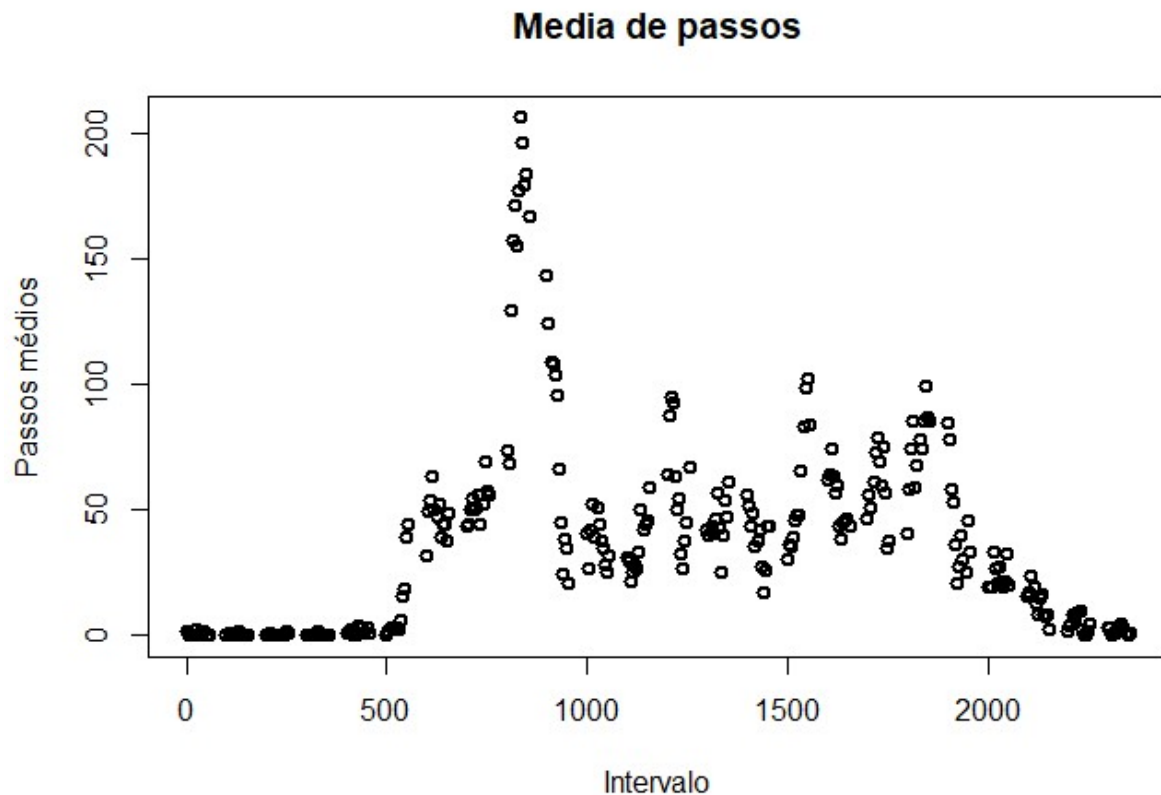
=====

What is the average daily activity pattern? ### 1. Criar variavel

```
AtividadeMedia <- aggregate(atividades$steps, by=list(atividades$interval), FUN=mean, na.rm=TRUE)
names(AtividadeMedia) <- c("interval", "mean")
```

2. Plotar grafico

```
plot(AtividadeMedia$interval, AtividadeMedia$mean, main="Media de passos", xlab="Intervalo", ylab="Passos médios", lwd = 2)
```



3. Dia com numero maximo

```
AtividadeMedia[which.max(AtividadeMedia$mean), ]$interval
```

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

=====

Imputing missing values ### 1. Numero total de valores faltantes

```
sum(is.na(atividades$steps))
```

```
## [1] 2304
```

2. Substituir os valores pela media de passsos

```
PassosMedios <- AtividadeMedia$mean[match(atividades$interval, AtividadeMedia$interval)]
```

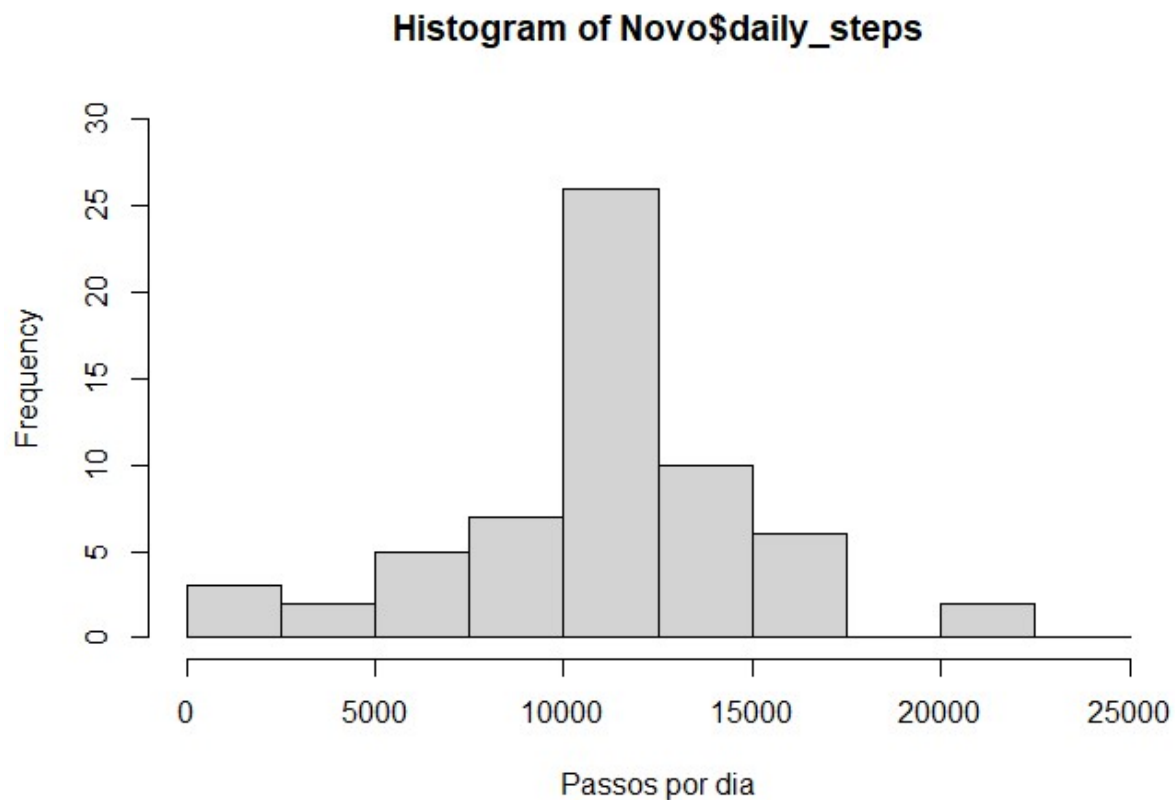
3. Criar novo conjunto inserindo os passosmedios no lugar dos NA

```
input <- transform(atividades, steps = ifelse(is.na(atividades$steps), yes = PassosMedios, no = atividades$steps))
Novo <- aggregate(steps ~ date, input, sum)
names(Novo) <- c("date", "daily_steps")
```

4. Fazer histograma

```
hist(Novo$daily_steps, xlab = "Passos por dia", ylim = c(0,30), breaks = seq(0,25000,by=2500, main = "Passos totais atualizado"))
```

```
## Warning: In seq.default(0, 25000, by = 2500, main = "Passos totais atualizado")
:
## extra argument 'main' will be disregarded
```



5. Media

```
mean(Novo$daily_steps)
```

```
## [1] 10766.19
```

6. Mediana

```
median(Novo$daily_steps)
```

```
## [1] 10766.19
```

=====

Are there differences in activity patterns between weekdays and weekends? ### 1. Formatar coluna

```
atividades$date <- as.Date(strptime(atividades$date, format="%Y-%m-%d"))
```

2. Criar variaveis

```
atividades$tipo <- sapply(atividades$date, function(x) {  
  if (weekdays(x) == "Saturday" | weekdays(x) == "Sunday")  
    {y <- "Weekend"} else  
    {y <- "Weekday"}  
  y  
})
```

4. Ver se deu certo

```
head(atividades, n=10)
```

```
##      steps      date interval  tipo  
## 1      NA 2012-10-01         0 Weekday  
## 2      NA 2012-10-01         5 Weekday  
## 3      NA 2012-10-01        10 Weekday  
## 4      NA 2012-10-01        15 Weekday  
## 5      NA 2012-10-01        20 Weekday  
## 6      NA 2012-10-01        25 Weekday  
## 7      NA 2012-10-01        30 Weekday  
## 8      NA 2012-10-01        35 Weekday  
## 9      NA 2012-10-01        40 Weekday  
## 10     NA 2012-10-01        45 Weekday
```

5. Plotar o grafico

```
concat <- aggregate(steps~interval + tipo, atividades, mean, na.rm = TRUE)
plot<- ggplot(concat, aes(x = interval , y = steps, color = tipo)) +
  geom_line() +
  labs(title = "media diaria", x = "data", y = "media") +
  facet_wrap(~tipo, ncol = 1, nrow=2)
print(plot)
```

