Activity Monitoring Data Analysis

Your Name

2025-03-24

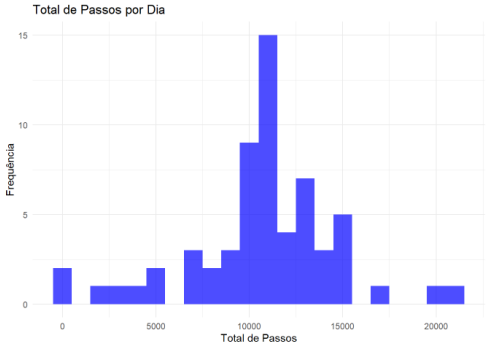
Loading and Preprocessing the Data

```
# Carregar os dados
activity_data <- read.csv("activity.csv")

# Converter a coluna date para formato Date
activity_data$date <- as.Date(activity_data$date, format="%Y-%m-%d")

# Visualizar os primeiros registros
head(activity_data)
```

```
##       steps       date interval
## 1 1.7169811 2012-10-01        0
## 2 0.3396226 2012-10-01        5
## 3 0.1320755 2012-10-01       10
## 4 0.1509434 2012-10-01       15
## 5 0.0754717 2012-10-01       20
## 6 2.0943396 2012-10-01       25
```

What is the total number of steps taken per day?

```
# Total de passos por dia (ignorando NAs)
total_steps_per_day <- activity_data %>%
  group_by(date) %>%
  summarize(total_steps = sum(steps, na.rm = TRUE))

# Criar histograma
ggplot(total_steps_per_day, aes(x = total_steps)) +
  geom_histogram(binwidth = 1000, fill = "blue", alpha = 0.7) +
  labs(title = "Total de Passos por Dia", x = "Total de Passos", y = "Frequência") +
  theme_minimal()
```




```
# Média e mediana
mean_steps <- mean(total_steps_per_day$total_steps, na.rm = TRUE)
median_steps <- median(total_steps_per_day$total_steps, na.rm = TRUE)

mean_steps
```

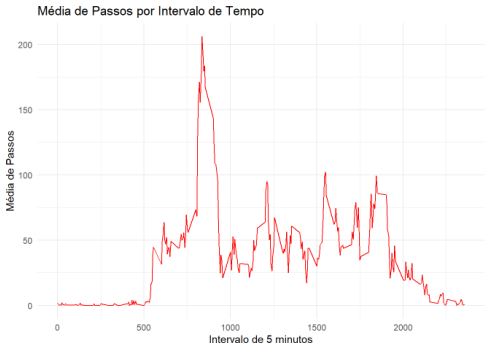
```
## [1] 10766.19
```

```
median_steps
```

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

What is the average daily activity pattern?

```
# Média de passos por intervalo de 5 minutos
average_steps_interval <- activity_data %>%
  group_by(interval) %>%
  summarize(avg_steps = mean(steps, na.rm = TRUE))

# Criar gráfico
ggplot(average_steps_interval, aes(x = interval, y = avg_steps)) +
  geom_line(color = "red") +
  labs(title = "Média de Passos por Intervalo de Tempo", x = "Intervalo de 5 minutos", y = "Média de Passos") +
  theme_minimal()
```




```
# Intervalo com o maior número médio de passos
max_interval <- average_steps_interval[which.max(average_steps_interval$avg_steps), ]
max_interval
```

```
## # A tibble: 1 × 2
##   interval avg_steps
##      <int>     <dbl>
## 1      835      206.
```

Imputing Missing Values

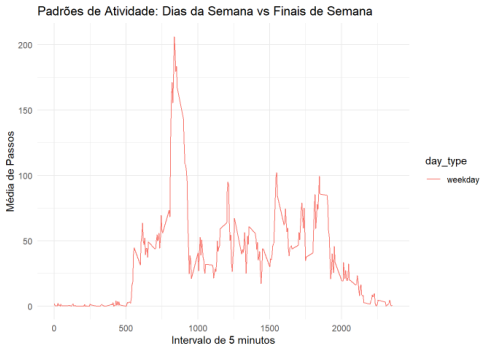
```
# Número de valores ausentes
num_missing <- sum(is.na(activity_data$steps))

# Substituir NAs pela média do respectivo intervalo
activity_data_imputed <- activity_data %>%
  group_by(interval) %>%
  mutate(steps = ifelse(is.na(steps), mean(steps, na.rm = TRUE), steps))

# Criar novo dataset
write.csv(activity_data_imputed, "activity_imputed.csv", row.names = FALSE)
```

Are there differences in activity patterns between weekdays and weekends?

```
# Criar coluna para diferenciar dia da semana e fim de semana
activity_data_imputed$day_type <- ifelse(weekdays(activity_data_imputed$date) %in% c("Saturday", "Sunday"), "weekend", "weekday")

# Média de passos por intervalo e tipo de dia
steps_weekday_weekend <- activity_data_imputed %>%
  group_by(interval, day_type) %>%
  summarize(avg_steps = mean(steps))
```

```
## `summarise()` has grouped output by 'interval'. You can override using
## the `.groups` argument.
```

```
# Criar gráfico
ggplot(steps_weekday_weekend, aes(x = interval, y = avg_steps, color = day_type)) +
  geom_line() +
  labs(title = "Padrões de Atividade: Dias da Semana vs Finais de Semana",
       x = "Intervalo de 5 minutos", y = "Média de Passos") +
  theme_minimal()
```



Conclusion

Os gráficos mostram padrões claros de atividade ao longo do dia, destacando diferenças entre dias úteis e finais de semana.