PA1_template

Hazem Haffouz

2023-12-25

Load required libraries

```
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
```

Load data

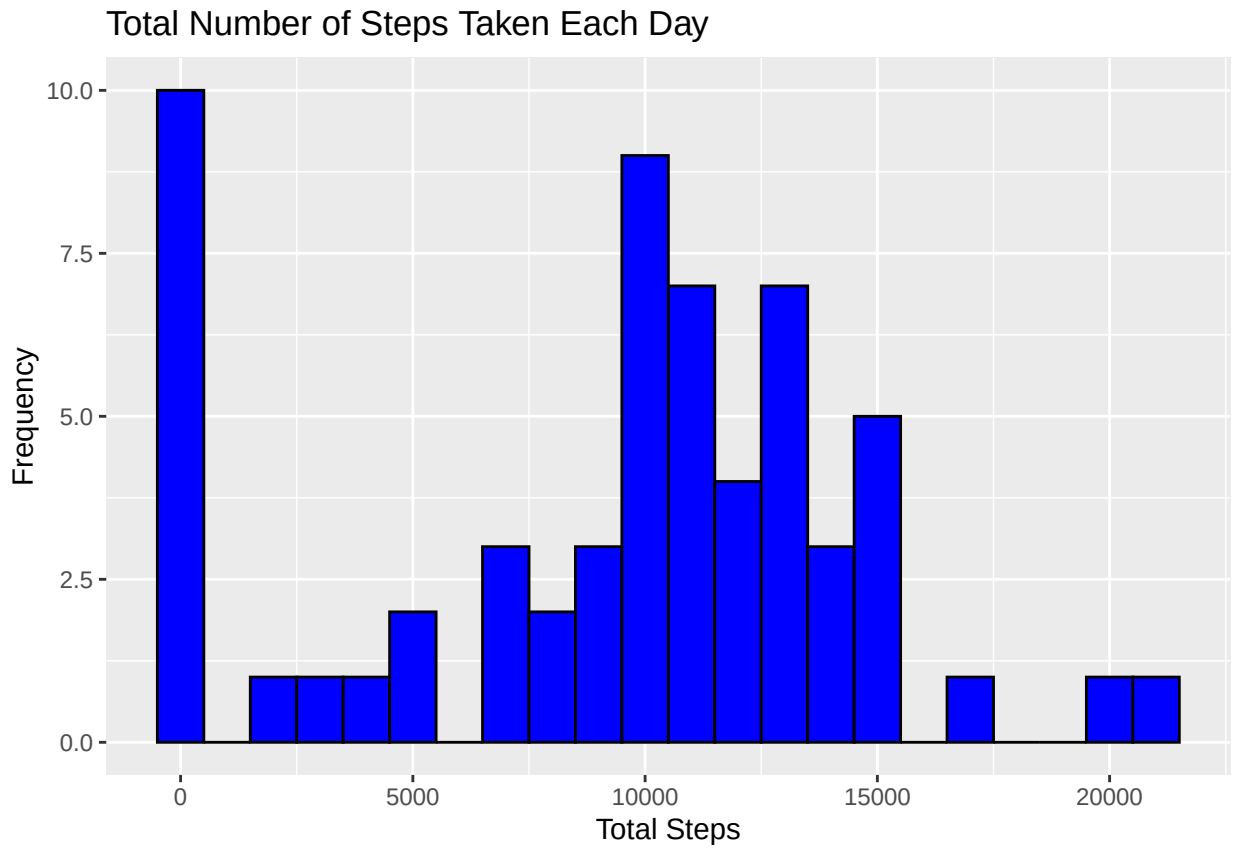
```
activity_data <- read.csv("activity.csv")
```

Calculate the total number of steps taken per day

```
total_steps_per_day <- activity_data %>%
  group_by(date) %>%
  summarise(total_steps = sum(steps, na.rm = TRUE))
```

Histogram of the total number of steps taken each day

```
ggplot(total_steps_per_day, aes(x = total_steps)) +
  geom_histogram(binwidth = 1000, fill = "blue", color = "black") +
  labs(title = "Total Number of Steps Taken Each Day",
       x = "Total Steps",
       y = "Frequency")
```



Calculate and report the mean and median of the total number of steps taken per day

```
mean_steps <- mean(total_steps_per_day$total_steps)
median_steps <- median(total_steps_per_day$total_steps)
```

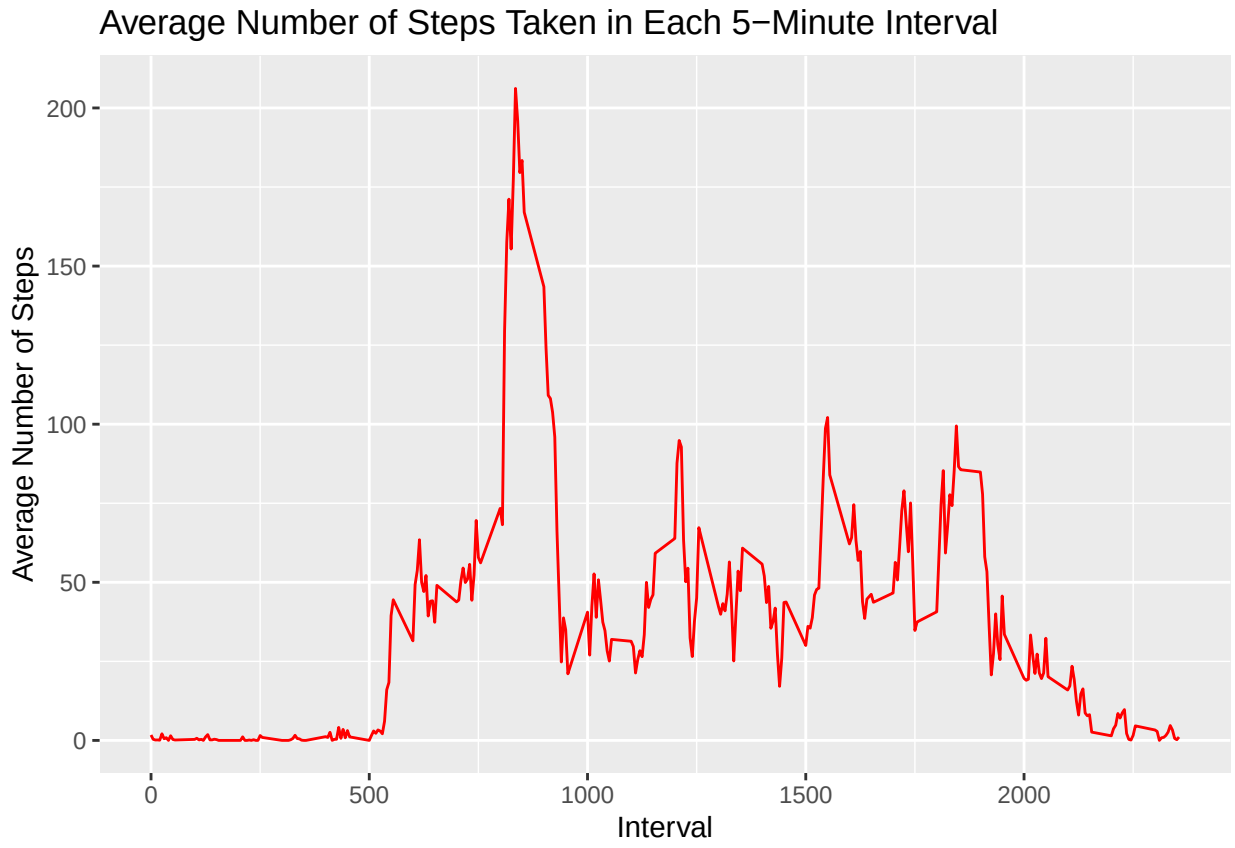
Average Daily Activity Pattern

```
average_daily_pattern <- activity_data %>%
  group_by(interval) %>%
  summarise(average_steps = mean(steps, na.rm = TRUE))
```

Time series plot of the average number of steps taken

```
ggplot(average_daily_pattern, aes(x = interval, y = average_steps)) +
  geom_line(color = "red") +
  labs(title = "Average Number of Steps Taken in Each 5-Minute Interval",
```

```
       x = "Interval",
       y = "Average Number of Steps")
```



Identify the 5-minute interval that has the maximum number of steps on average

```
max_interval <- average_daily_pattern[which.max(average_daily_pattern$average_steps), ]$interval
```

Imputing missing values

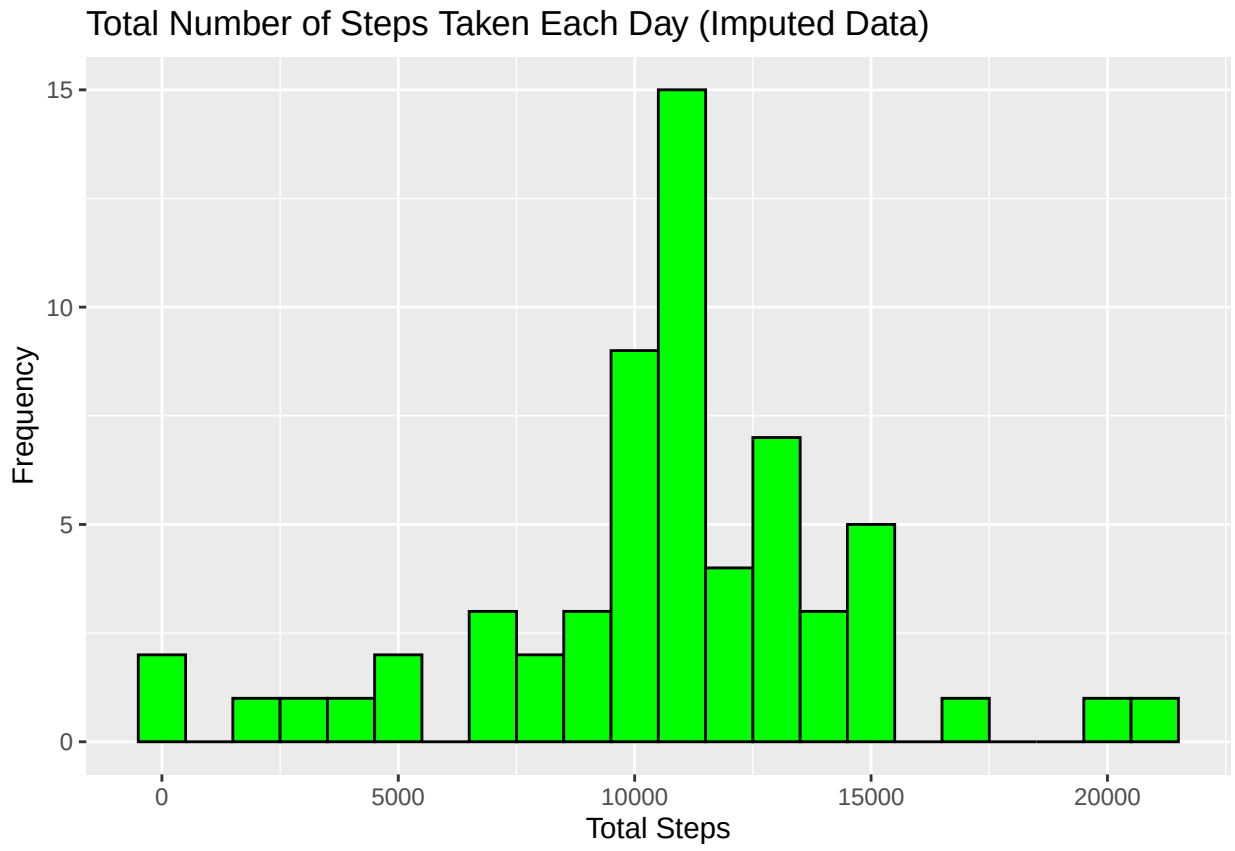
Strategy: Replace NA with mean for that 5-minute interval

```
imputed_data <- activity_data
for(i in 1:nrow(imputed_data)) {
  if(is.na(imputed_data$steps[i])) {
    imputed_data$steps[i] <- average_daily_pattern[average_daily_pattern$interval == imputed_data$inter
  }
}
```

Histogram of the total number of steps taken each day after imputing missing values

```
total_steps_per_day_imputed <- imputed_data %>%
  group_by(date) %>%
  summarise(total_steps = sum(steps))
```

```
ggplot(total_steps_per_day_imputed, aes(x = total_steps)) +
  geom_histogram(binwidth = 1000, fill = "green", color = "black") +
  labs(title = "Total Number of Steps Taken Each Day (Imputed Data)",
       x = "Total Steps",
       y = "Frequency")
```



Are there differences in activity patterns between weekdays and weekends?

Create a new factor variable

```r
imputed_data$date <- as.Date(imputed_data$date)
imputed_data$day_type <- ifelse(weekdays(imputed_data$date) %in% c("Saturday", "Sunday"), "weekend", "w
```

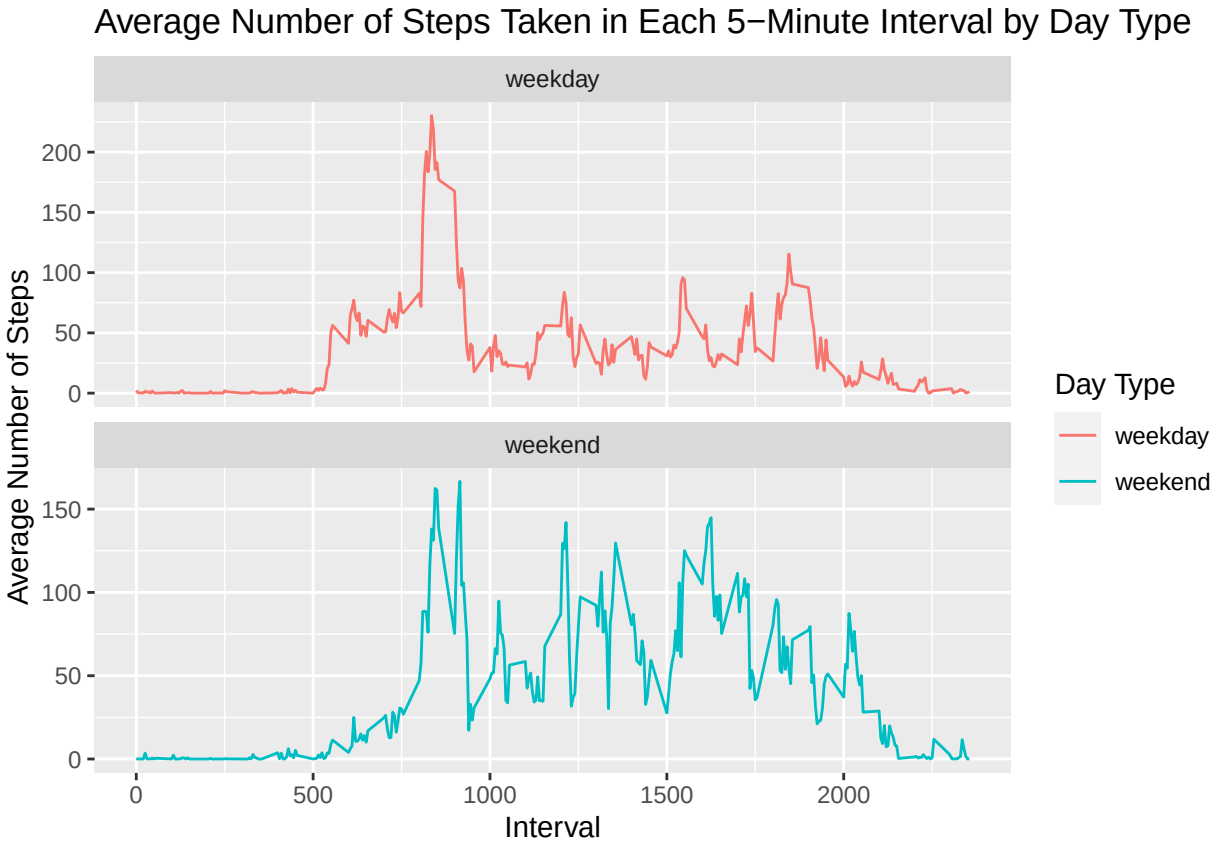
Average steps taken per interval across weekdays and weekends

```r
average_steps_by_day_type <- imputed_data %>%
  group_by(interval, day_type) %>%
  summarise(average_steps = mean(steps))
```

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

Panel plot

```r
ggplot(average_steps_by_day_type, aes(x = interval, y = average_steps, color = day_type)) +
  geom_line() +
  facet_wrap(~ day_type, ncol = 1, scales = "free_y") +
  labs(title = "Average Number of Steps Taken in Each 5-Minute Interval by Day Type",
       x = "Interval",
       y = "Average Number of Steps",
       color = "Day Type")
```


Average Number of Steps Taken in Each 5−Minute Interval by Day Type
weekday
weekend
Average Number of Steps
0
50
100
150
200
Interval
0
500
1000
1500
2000
Day Type
weekday
weekend