assignment

Rvdkroon

2025-04-22

To deal with missing data na.rm=TRUE is added

```
# Load necessary libraries
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
```

```
library(ggplot2)

# Load the dataset
activity <- read.csv("activity.csv")

# Convert date column to Date type
activity$date <- as.Date(activity$date)

# View structure of the dataset
str(activity)
```

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

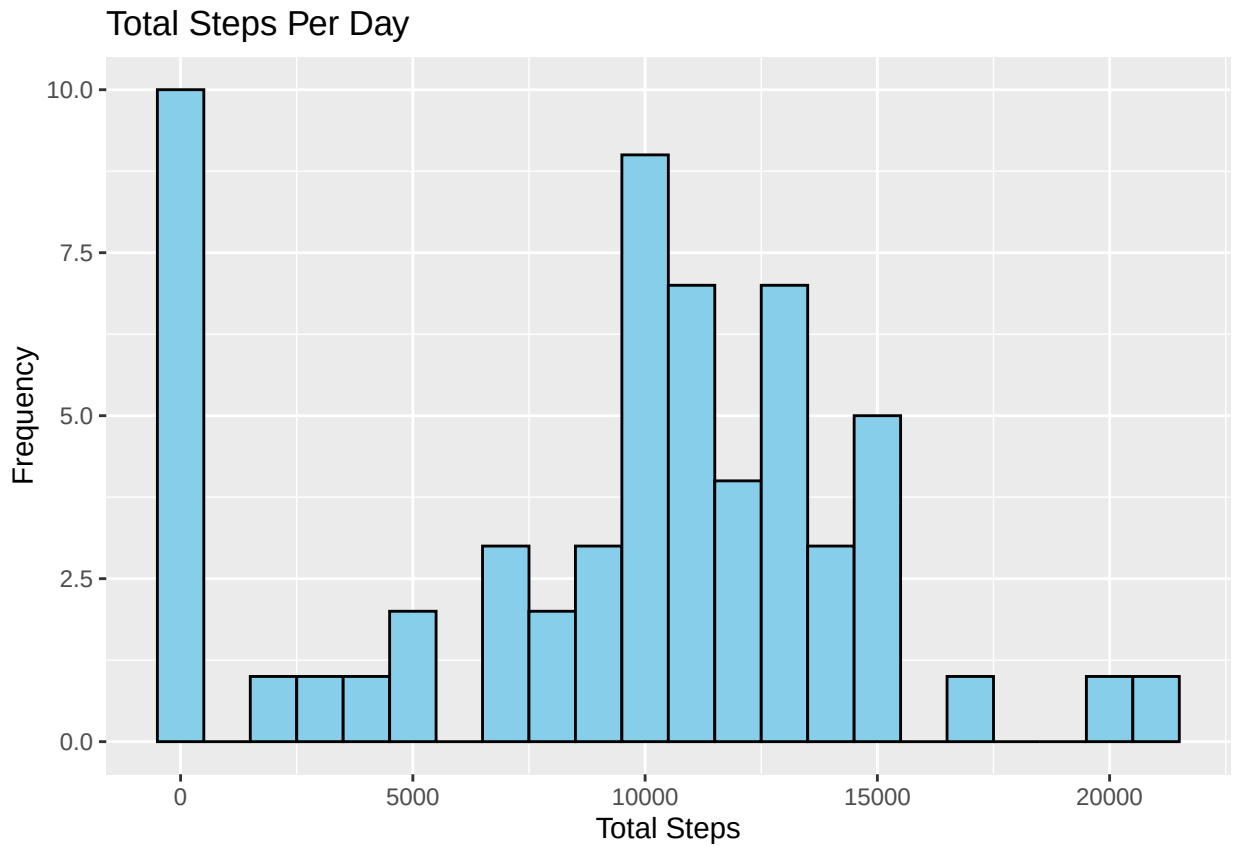
```
summary(activity)
```

```
##      steps             date               interval
##  Min.   :  0.00   Min.   :2012-10-01   Min.   :   0.0
##  1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.: 588.8
##  Median :  0.00   Median :2012-10-31   Median :1177.5
```

```
##  Mean   : 37.38   Mean   :2012-10-31   Mean   :1177.5
##  3rd Qu.: 12.00   3rd Qu.:2012-11-15   3rd Qu.:1766.2
##  Max.   :806.00   Max.   :2012-11-30   Max.   :2355.0
##  NA's   :2304
```

```
# Calculate total steps per day
total_steps_per_day <- activity %>%
  group_by(date) %>%
  summarise(total_steps = sum(steps, na.rm = TRUE))

# Plot histogram
ggplot(total_steps_per_day, aes(x = total_steps)) +
  geom_histogram(binwidth = 1000, fill = "skyblue", color = "black") +
  labs(title = "Total Steps Per Day", x = "Total Steps", y = "Frequency")
```



Mean and median steps

median steps=10395 and mean steps = 9335

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

mean_steps
```

```
## [1] 9354.23
```

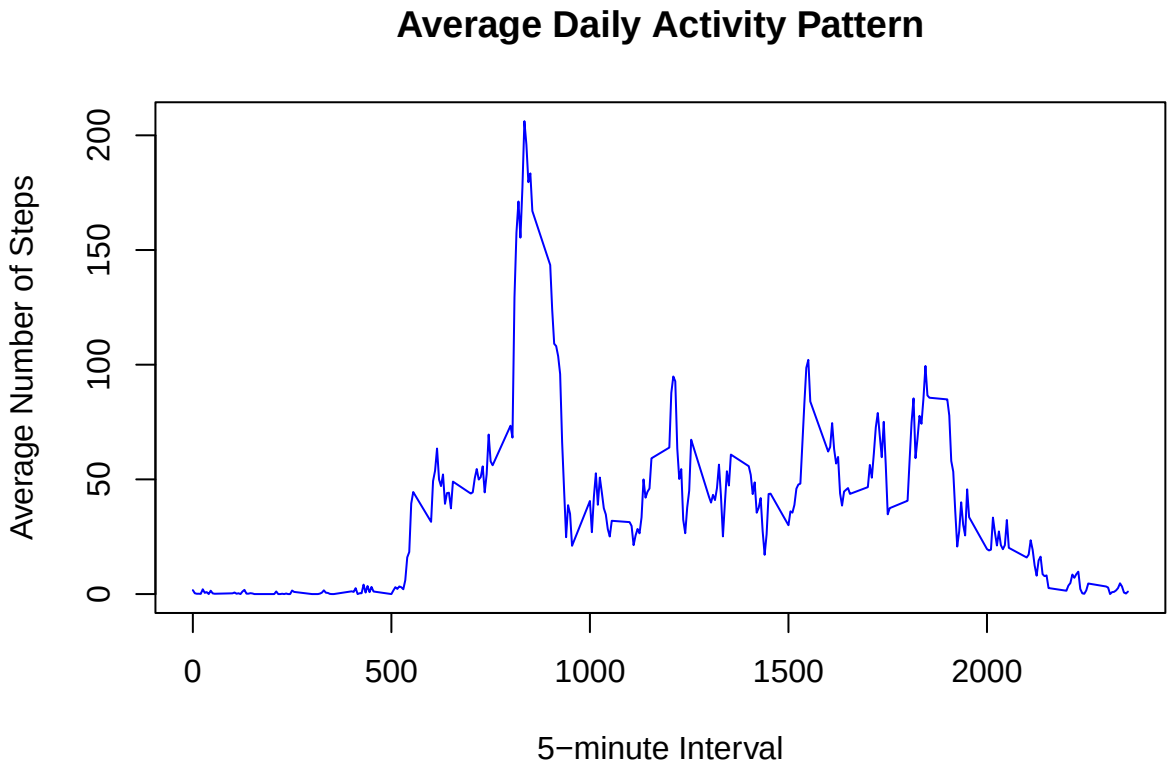
```
median_steps
```

```
## [1] 10395
```

Average steps a day and missing data

```
# Calculate average steps per interval
avg_steps_interval <- activity %>%
  group_by(interval) %>%
  summarise(mean_steps = mean(steps, na.rm = TRUE))

# Time series plot
plot(avg_steps_interval$interval, avg_steps_interval$mean_steps,
     type = "l",
     col = "blue",
     xlab = "5-minute Interval",
     ylab = "Average Number of Steps",
     main = "Average Daily Activity Pattern")
```



Max steps 5 minute interval

Interval 835 gave maximum steps

```
max_interval <- avg_steps_interval[which.max(avg_steps_interval$mean_steps), ]
max_interval
```

```
## # A tibble: 1 x 2
##   interval mean_steps
##      <int>      <dbl>
## 1      835       206.
```

Create a new factor variable for day type

activity$day_type <- ifelse(weekdays(activity$date) %in% c("Saturday", "Sunday"), "weekend", "weekday")
activity$day_type <- factor(activity$day_type, levels = c("weekday", "weekend"))

Quick check of the new factor

table(activity$day_type)

Average steps by interval and day_type

avg_steps_daytype <- activity %>% group_by(interval, day_type) %>% summarise(mean_steps = mean(steps, na.rm = TRUE), .groups = 'drop')

Panel plot using ggplot2

ggplot(avg_steps_daytype, aes(x = interval, y = mean_steps)) + geom_line(color = "darkgreen") + facet_wrap(~ day_type, ncol = 1) + labs(title = "Average Daily Activity Patterns: Weekday vs Weekend", x = "5-minute Interval", y = "Average Number of Steps") + theme_minimal() "'