

COURSERA PROJECT 1

Elshan Musazade

2025-07-01

Introduction

This assignment uses data from a personal activity monitoring device that collects steps at 5-minute intervals. The dataset spans October and November 2012 and will be analyzed to explore physical activity patterns, handle missing data, and compare activity behavior between weekdays and weekends.

Load and Inspect Data

```
# Load dataset
activity <- read.csv("C:/Users/ASUS/Desktop/homework/activity.csv")

# Convert date to Date type and check structure
activity$date <- as.Date(activity$date)
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 ...
```

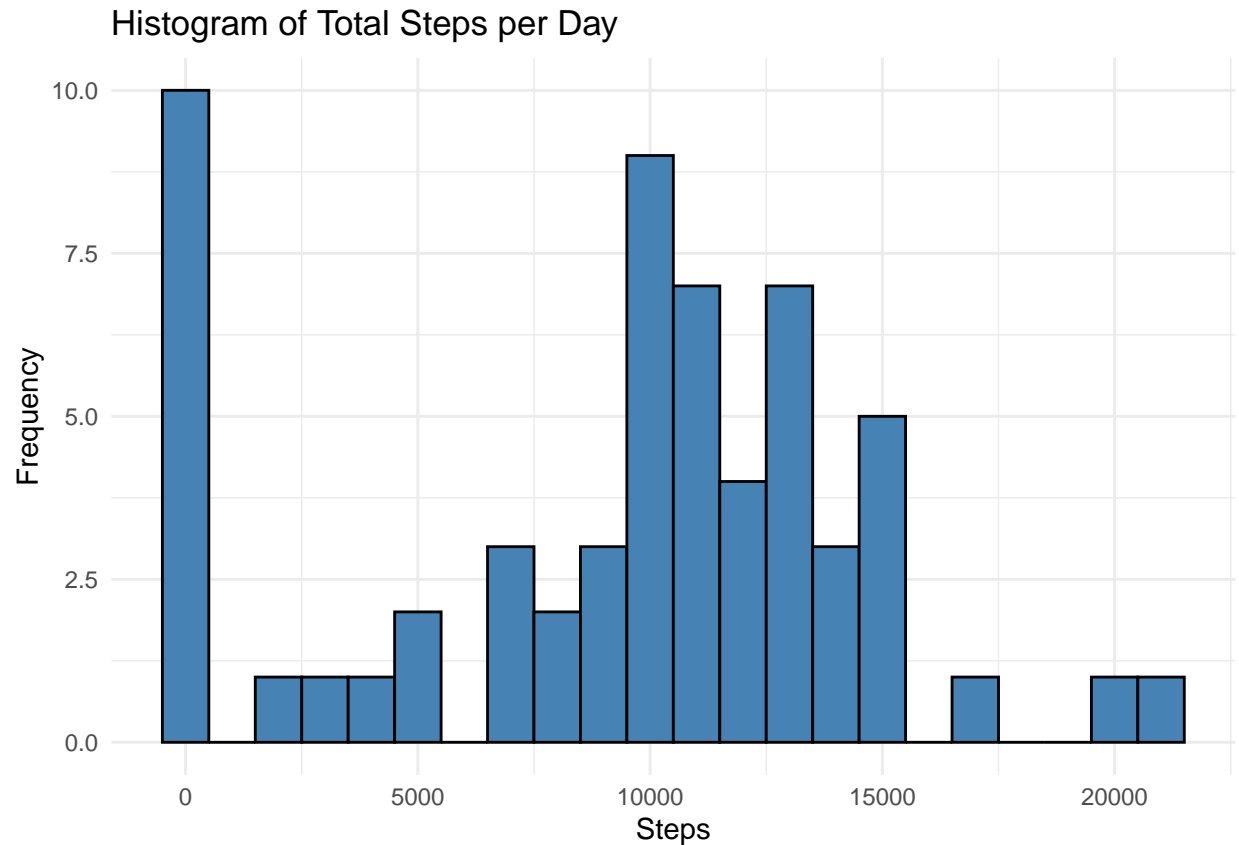
```
# Check for missing data in 'steps'
missing_steps <- sum(is.na(activity$steps))
missing_steps
```

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

Total Number of Steps per Day

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

# Plot histogram of total steps per day
ggplot(daily_steps, aes(x = total_steps)) +
  geom_histogram(binwidth = 1000, fill = "steelblue", color = "black") +
  labs(title = "Histogram of Total Steps per Day", x = "Steps", y = "Frequency") +
  theme_minimal()
```



```
# Calculate and display mean and median of daily steps
mean_steps <- mean(daily_steps$total_steps)
median_steps <- median(daily_steps$total_steps)
mean_steps
```

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

```
median_steps
```

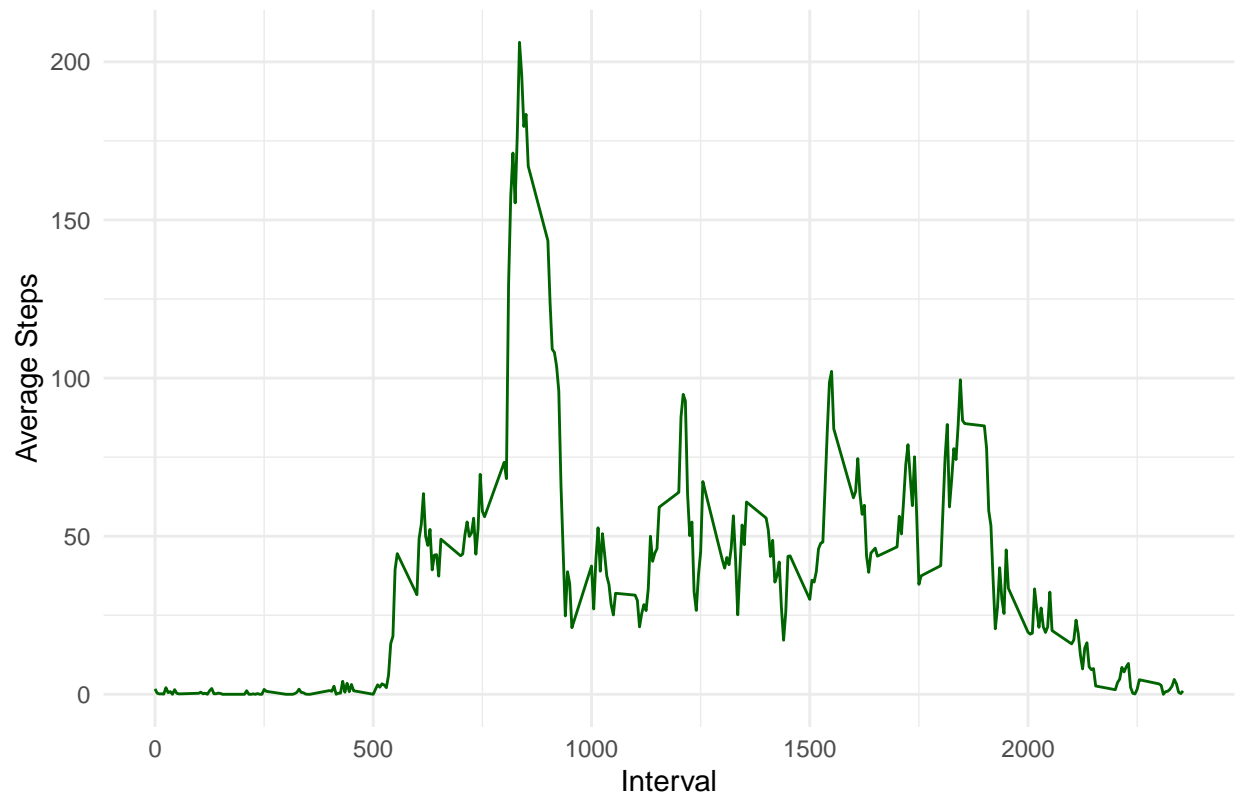
```
## [1] 10395
```

Average Daily Activity Pattern

```
# Calculate average steps by 5-minute interval
interval_avg <- activity %>%
  group_by(interval) %>%
  summarise(avg_steps = mean(steps, na.rm = TRUE))

# Plot average steps by interval
ggplot(interval_avg, aes(x = interval, y = avg_steps)) +
  geom_line(color = "darkgreen") +
  labs(title = "Average Steps by 5-Minute Interval", x = "Interval", y = "Average Steps") +
  theme_minimal()
```

Average Steps by 5–Minute Interval



```
# Find the interval with the maximum average steps
max_interval <- interval_avg[which.max(interval_avg$avg_steps), ]
max_interval
```

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

Imputing Missing Values

```
# Impute missing steps using the average steps for the same interval
activity_imputed <- activity %>%
  left_join(interval_avg, by = "interval") %>%
  mutate(steps = ifelse(is.na(steps), avg_steps, steps)) %>%
  select(date, interval, steps)

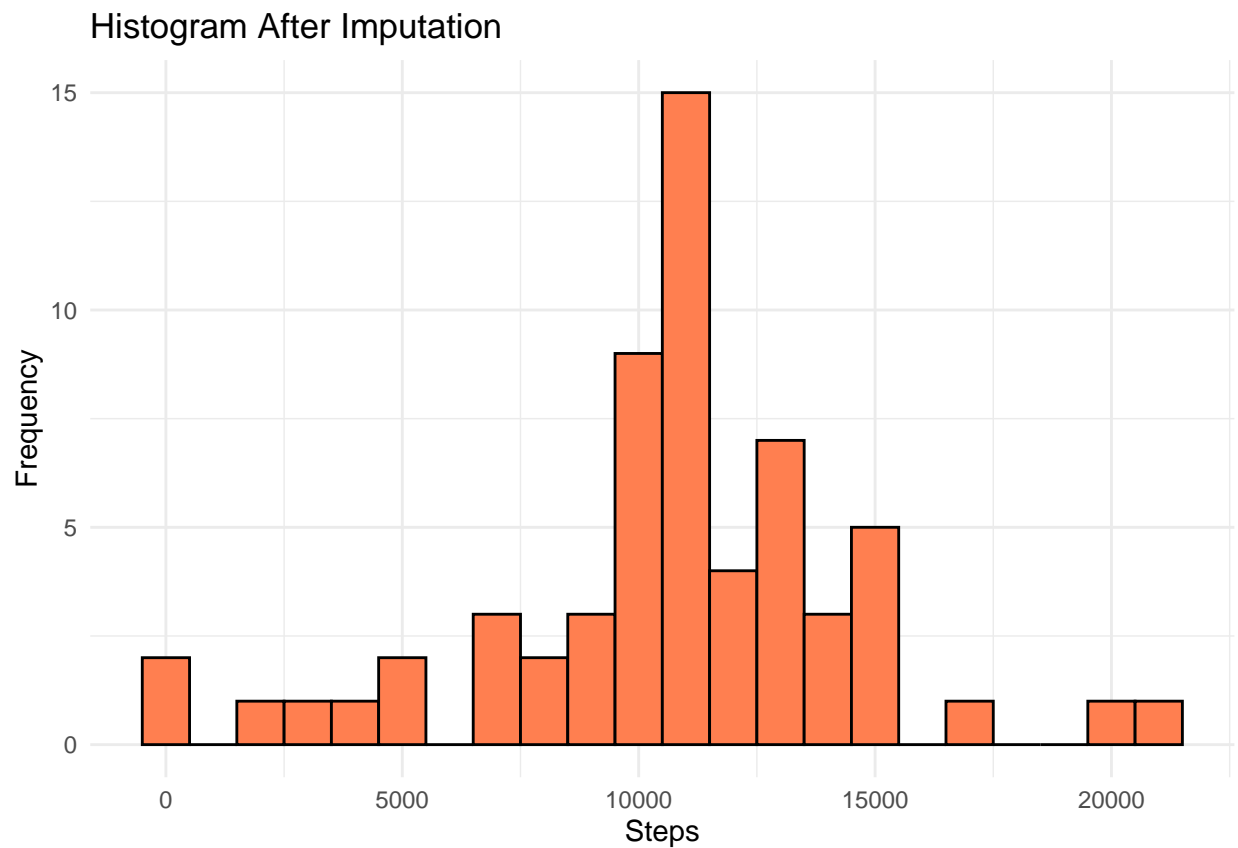
# Check for any remaining missing values
remaining_missing <- sum(is.na(activity_imputed$steps))
remaining_missing
```

```
## [1] 0
```

Total Steps After Imputation

```
# Calculate total steps per day after imputation
daily_steps_imputed <- activity_imputed %>%
  group_by(date) %>%
  summarise(total_steps = sum(steps))

# Plot histogram of total steps after imputation
ggplot(daily_steps_imputed, aes(x = total_steps)) +
  geom_histogram(binwidth = 1000, fill = "coral", color = "black") +
  labs(title = "Histogram After Imputation", x = "Steps", y = "Frequency") +
  theme_minimal()
```



```
# Calculate and display mean and median of total steps after imputation
mean_imputed_steps <- mean(daily_steps_imputed$total_steps)
median_imputed_steps <- median(daily_steps_imputed$total_steps)
mean_imputed_steps
```

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

```
median_imputed_steps
```

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

Compare Before vs After Imputation

```
# Compare mean and median before and after imputation
original_stats <- daily_steps %>%
  summarise(mean = mean(total_steps), median = median(total_steps))

imputed_stats <- daily_steps_imputed %>%
  summarise(mean = mean(total_steps), median = median(total_steps))

# Display comparison
comparison <- rbind(original_stats, imputed_stats)
comparison
```

```
## # A tibble: 2 x 2
##   mean median
##   <dbl> <dbl>
## 1  9354. 10395
## 2 10766. 10766.
```

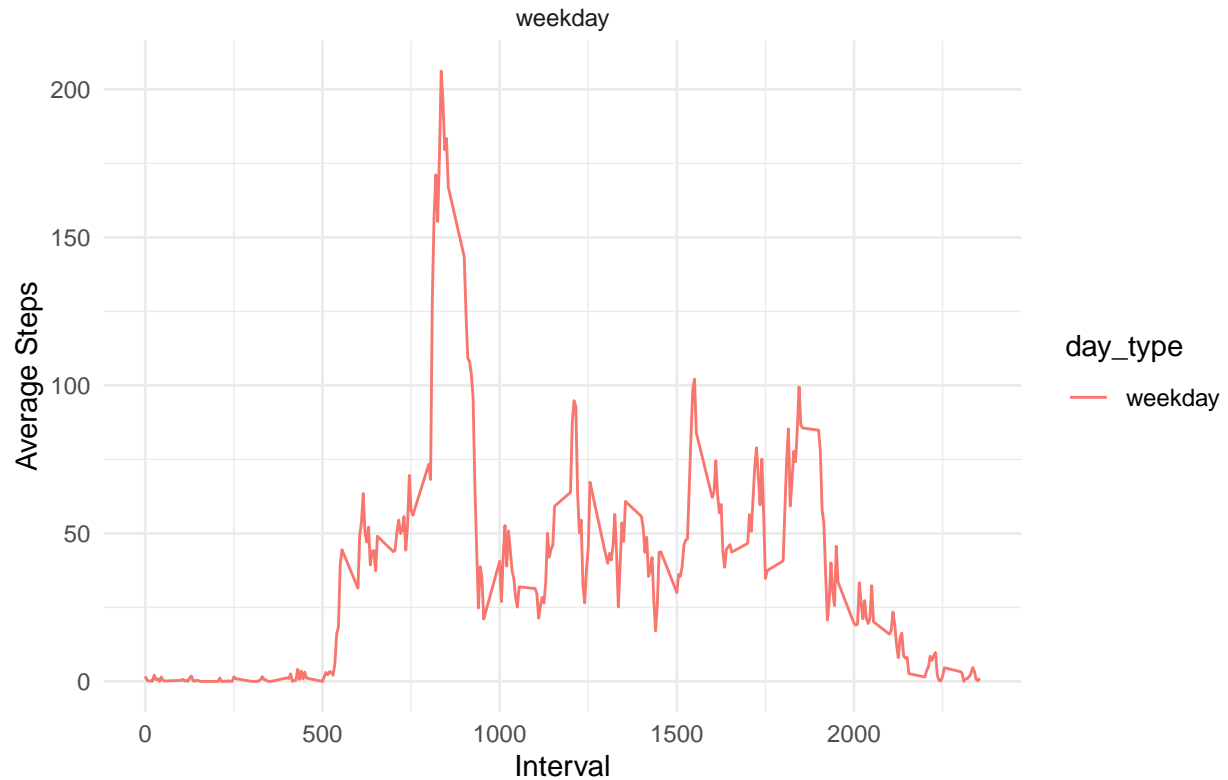
Weekday vs Weekend Patterns

```
# Add 'day_type' column to classify days as 'weekday' or 'weekend'
activity_imputed$day_type <- ifelse(weekdays(activity_imputed$date) %in% c("Saturday", "Sunday"), "weekend", "weekday")

# Calculate average steps by interval for weekdays and weekends
interval_daytype <- activity_imputed %>%
  group_by(interval, day_type) %>%
  summarise(avg_steps = mean(steps), .groups = 'drop')

# Plot steps by interval for weekdays vs weekends
ggplot(interval_daytype, aes(x = interval, y = avg_steps, color = day_type)) +
  geom_line() +
  facet_wrap(~day_type, nrow = 2) +
  labs(title = "Steps by Interval: Weekday vs Weekend", x = "Interval", y = "Average Steps") +
  theme_minimal()
```

Steps by Interval: Weekday vs Weekend



Conclusion

This report explored activity patterns, handled missing values with mean imputation, and compared behaviors between weekdays and weekends. The analysis was conducted using modern `dplyr` and `ggplot2` functions for efficient and reproducible results.