Knitr Project

Jackson Pham

2024-03-01

Part 1: Loading and Reprocessing Data

```
data <- read.csv("Downloads/activity.csv")
head(data)
```

```
##   steps       date interval
## 1    NA 2012-10-01        0
## 2    NA 2012-10-01        5
## 3    NA 2012-10-01       10
## 4    NA 2012-10-01       15
## 5    NA 2012-10-01       20
## 6    NA 2012-10-01       25
```

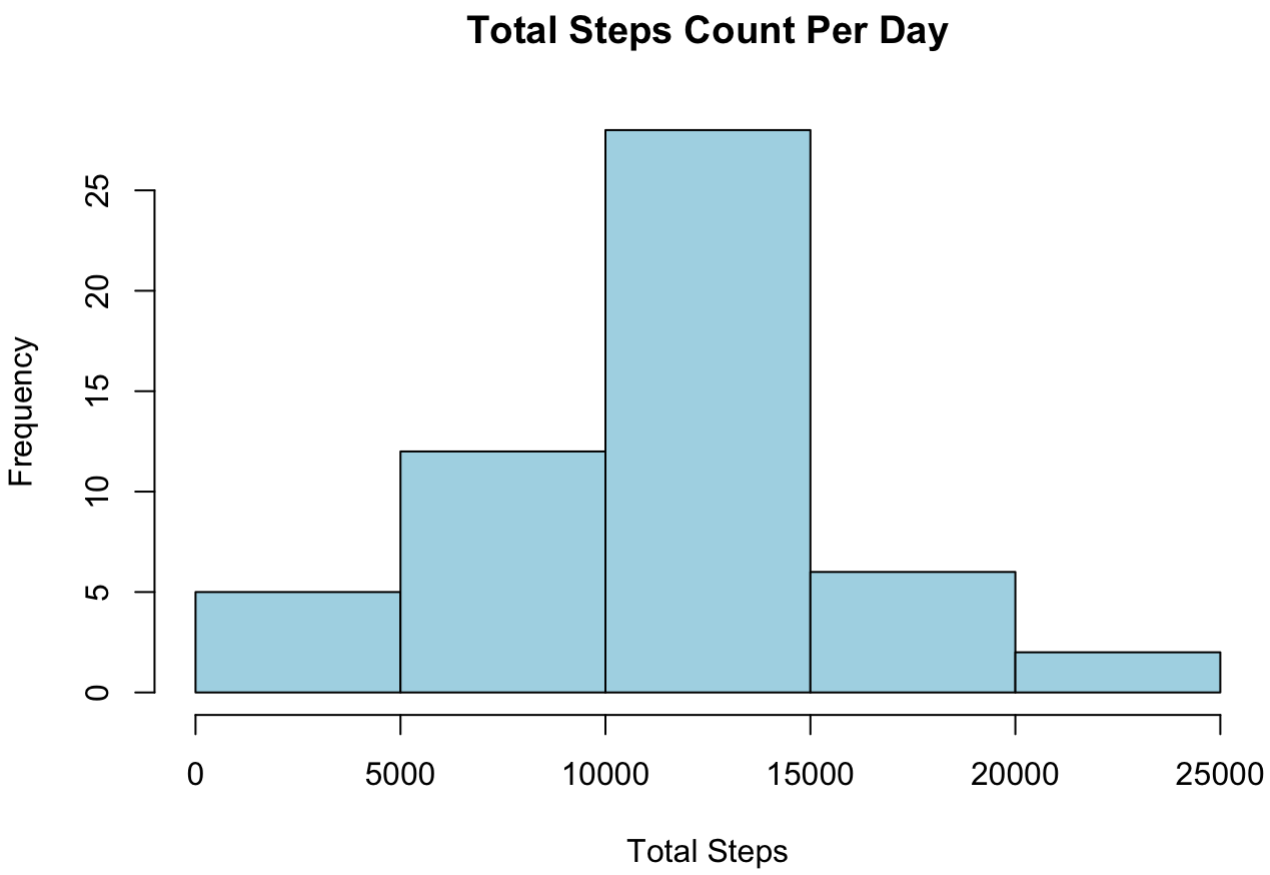
```
summary(data)
```

```
##      steps            date              interval     
##  Min.   :  0.00   Length:17568       Min.   :   0.0  
##  1st Qu.:  0.00   Class :character   1st Qu.: 588.8  
##  Median :  0.00   Mode  :character   Median :1177.5  
##  Mean   : 37.38                      Mean   :1177.5  
##  3rd Qu.: 12.00                      3rd Qu.:1766.2  
##  Max.   :806.00                      Max.   :2355.0  
##  NA's   :2304
```

```
# Determine the data frame and value, ignore all zero value
# And NA value
```

Part 2: What is mean total number of steps taken per day?

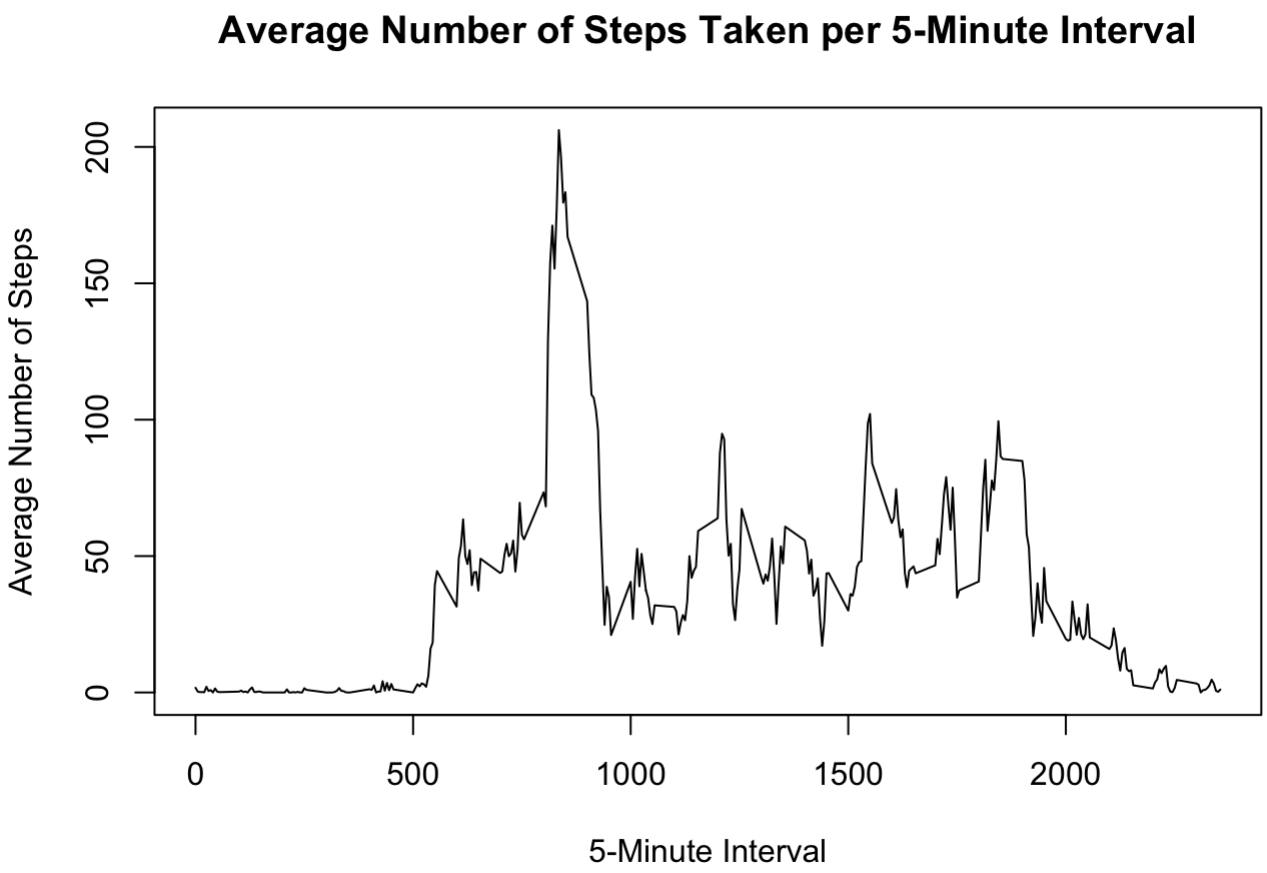
```
total_steps_per_day <- aggregate(steps ~ date, data = data, FUN = sum)

# Determine the total amount of steps per day

hist(total_steps_per_day$steps, main = "Total Steps Count Per Day", xlab = "Total Step
s", ylab = "Frequency", col = "lightblue")
```




```
# Graph the histogram of date vs total steps count, the date is counted as
# Frequency from 1 to 53 where 1 is the starting date of 2012-10-02 and 53
# Represent 2012-11-29
```

Part 3: What is the average daily activity pattern?

```
avg_steps_per_interval <- aggregate(steps ~ interval, data = data, FUN = mean)

# Determine the average steps per interval using the function aggregate with means
# This allow use to make the time series graph

plot(avg_steps_per_interval$interval, avg_steps_per_interval$steps, type = "l", xlab =
"5-Minute Interval", ylab = "Average Number of Steps", main = "Average Number of Steps T
aken per 5-Minute Interval")
```




```
# Plot the time series graph

max_interval <- avg_steps_per_interval$interval[which.max(avg_steps_per_interval$steps)]

# Find the interval with the maximum average number of steps

max_interval
```

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

Part 4: Imputing missing values

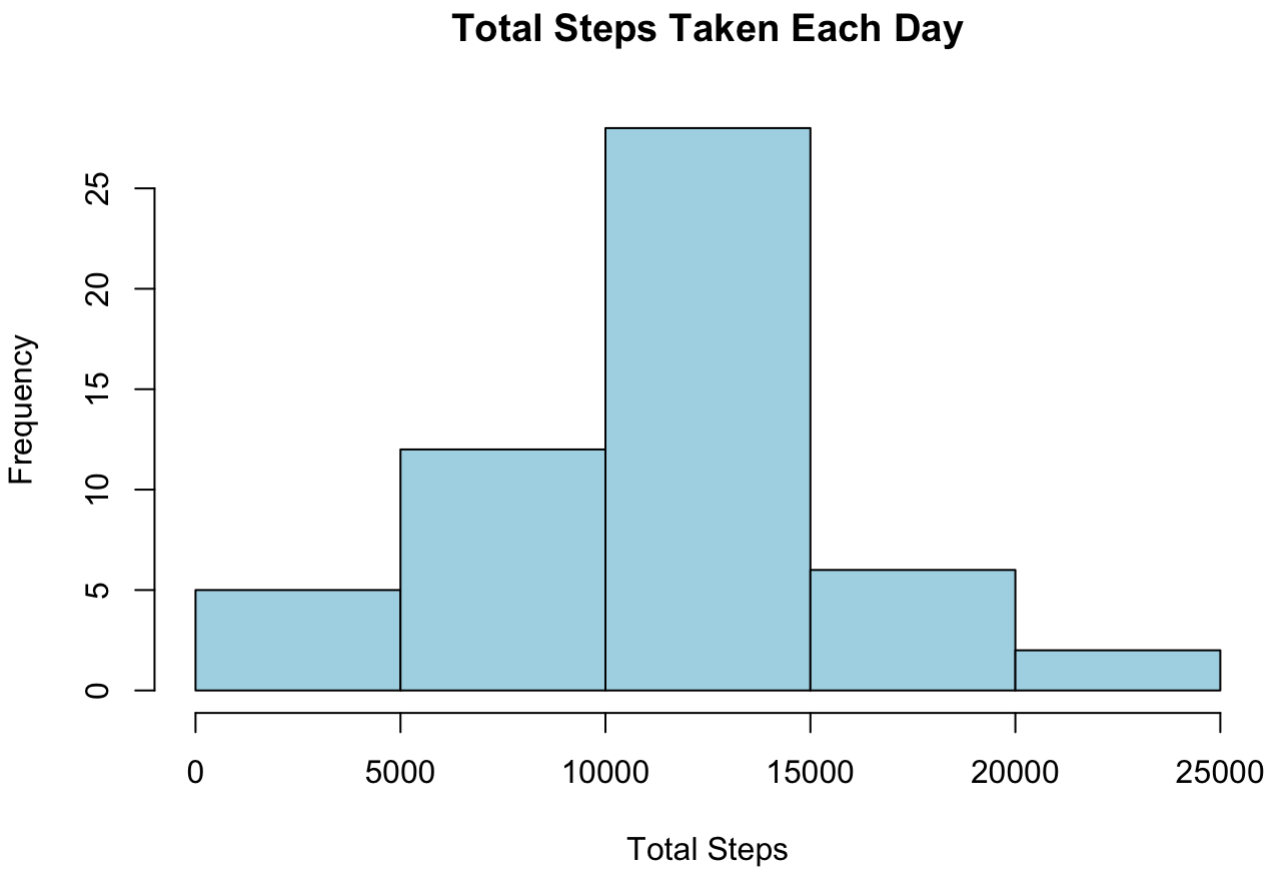
```
total_missing_value <- sum(is.na(data$steps))
total_missing_value
```

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

```
# Determine total missing values

# Create a copy of the original dataset

new_data <- total_steps_per_day

# Iterate over each row in the data frame

for (i in 1:nrow(new_data)) {
  # Check if the steps value is missing or zero
  if (is.na(new_data$steps[i]) | new_data$steps[i] == 0) {
    # Find the mean of steps for the corresponding date
    mean_steps <- mean(new_data$steps[new_data$date == new_data$date[i]], na.rm = TRUE)
    # Replace the missing or zero value with the mean
    new_data$steps[i] <- mean_steps
  }
}

# Present first few lines of the new data

head(new_data)
```

```
##         date steps
## 1 2012-10-02   126
## 2 2012-10-03 11352
## 3 2012-10-04 12116
## 4 2012-10-05 13294
## 5 2012-10-06 15420
## 6 2012-10-07 11015
```

```
# Create a histogram of the total number of steps taken each day

hist(new_data$steps, main = "Total Steps Taken Each Day", xlab = "Total Steps", ylab =
"Frequency", col = "lightblue")
```




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

mean_steps_per_day <- mean(new_data$steps, na.rm = TRUE)
median_steps_per_day <- median(new_data$steps, na.rm = TRUE)

mean_steps_per_day
```

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

```
median_steps_per_day
```

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

Part 5: Are there differences in activity patterns between weekdays and weekends?

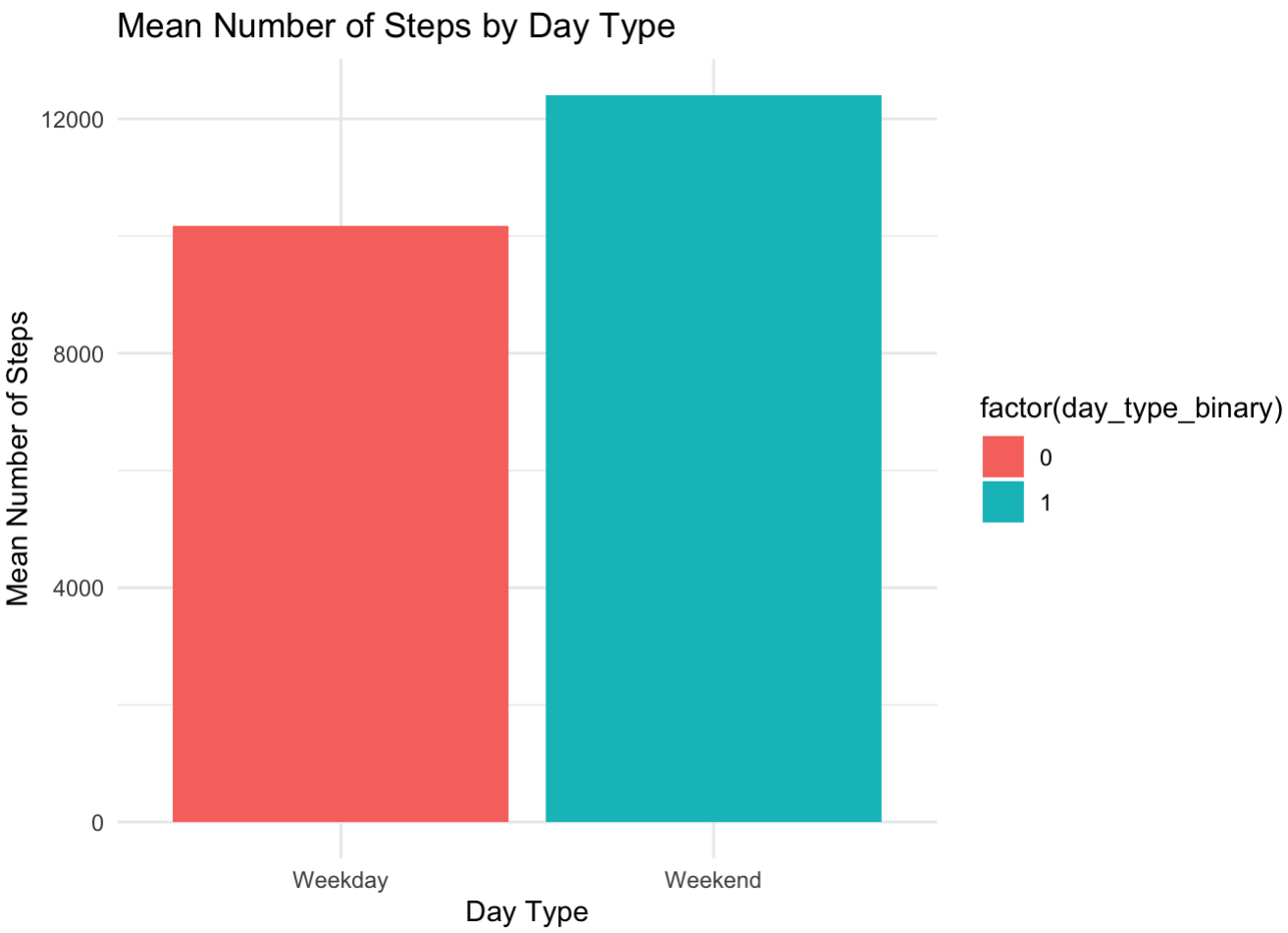
```
# Copy of new_data to a separate variable

new_data_weekday <- new_data
new_data_weekday$date <- as.Date(new_data$date)

# Create a for loop that goes through every column in the dataframe
# new_data$date using the weekdays() function to create a new coulmnn
# with the weekday inserted

new_data_weekday$day_type <- factor(weekdays(new_data_weekday$date) %in% c("Saturday",
"Sunday"),
                        levels = c(FALSE, TRUE),
                        labels = c("weekday", "weekend"))

# Print out first few row of the result

head(new_data_weekday)
```

```
##         date steps day_type
## 1 2012-10-02   126  weekday
## 2 2012-10-03 11352  weekday
## 3 2012-10-04 12116  weekday
## 4 2012-10-05 13294  weekday
## 5 2012-10-06 15420  weekend
## 6 2012-10-07 11015  weekend
```

```
# Load ggplot2

library(ggplot2)

# Turn weekday into 0, and weekend into 1 as a binary number

new_data_weekday$day_type_binary <- ifelse(new_data_weekday$day_type == "weekday", 0, 1)

head(new_data_weekday)
```

```
##         date steps day_type day_type_binary
## 1 2012-10-02   126  weekday               0
## 2 2012-10-03 11352  weekday               0
## 3 2012-10-04 12116  weekday               0
## 4 2012-10-05 13294  weekday               0
## 5 2012-10-06 15420  weekend               1
## 6 2012-10-07 11015  weekend               1
```

```
# Find the average number of steps between weekday and weekend

mean_steps <- aggregate(steps ~ day_type_binary, data = new_data_weekday, FUN = mean)

# Graph the bar graph between mean steps of weekday and weekend

ggplot(mean_steps, aes(x = factor(day_type_binary), y = steps, fill = factor(day_type_bi
nary))) +
  geom_bar(stat = "identity") +
  labs(title = "Mean Number of Steps by Day Type",
       x = "Day Type",
       y = "Mean Number of Steps") +
  scale_x_discrete(labels = c("Weekday", "Weekend")) +
  theme_minimal()
```




```
# The graph shows that more steps are taken in weekend than weekday
```