title: “Cousera Assignment 1” author: Aaron Goldman output: html_document date: “2024-11-16” —

# R Markdown

##Assignment Instructions 1.Code for reading in the dataset and/or processing the data 2.Histogram of the total number of steps taken each day 3.Mean and median number of steps taken each day 4.Time series plot of the average number of steps taken 5.The 5-minute interval that, on average, contains the maximum number of steps 6.Code to describe and show a strategy for imputing missing data 7.Histogram of the total number of steps taken each day after missing values are imputed 8.Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends 9.All of the R code needed to reproduce the results (numbers, plots, etc.) in the report

##Step 1 ##Code for reading in the dataset and/or processing the data

```
# Set the working directory to the folder containing your file
setwd("C:/Users/milli/OneDrive/Desktop/")

# Read the CSV file
data <- read.csv("activity.csv")

# Display the first few rows of the data
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
```

Exploring the basics of this data # Set the working directory (if necessary) setwd(“C:/Users/milli/OneDrive/Desktop/”)

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

# Exploring the basics of this data
dim(activity)
```

```
## [1] 17568     3
```

```
names(activity)
```

```
## [1] "steps"    "date"     "interval"
```

```
head(activity)
```

```
##   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
```

```
str(activity)
```

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

```
# Total number of missing data
sum(is.na(activity$steps)) / dim(activity)[[1]]
```

```
## [1] 0.1311475
```

```
# Transforming the date column into date format using lubridate
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
activity$date <- ymd(activity$date)
length(unique(activity$date))
```

```
## [1] 61
```

# 1. Calculate the total number of steps taken per day

To understand the overall activity level, we first calculate the total number of steps taken each day.

```
total_steps_per_day <- aggregate(steps ~ date, data = activity, sum, na.rm = TRUE)
```

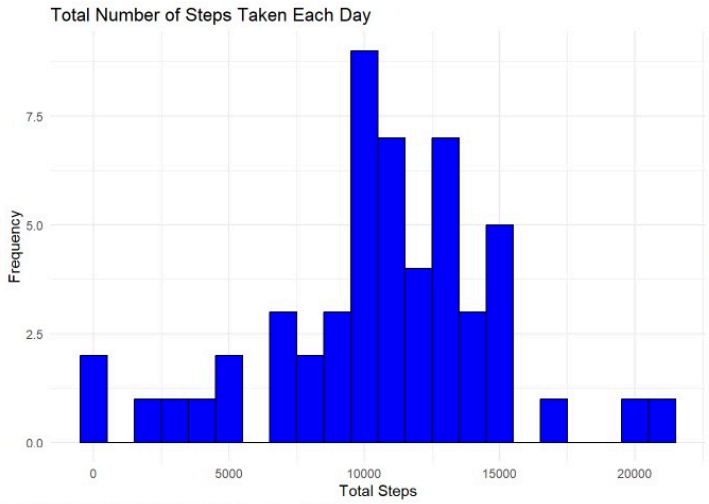
# 2. Make a Histogram of the Total Number of Steps Taken Each Day

A histogram visualizes the distribution of the total steps taken per day, helping us see the frequency of different activity levels.

# Load necessary library

```
library(ggplot2)

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




# 3. Calculate and Report the Mean and Median of the Total Number of Steps Taken Per Day

Calculating the mean and median provides summary statistics that describe the central tendency of the daily step counts.

```
# Calculate mean and median
mean_steps <- mean(total_steps_per_day$steps, na.rm = TRUE)
median_steps <- median(total_steps_per_day$steps, na.rm = TRUE)

# Report the mean and median
mean_steps
```

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

```
median_steps
```

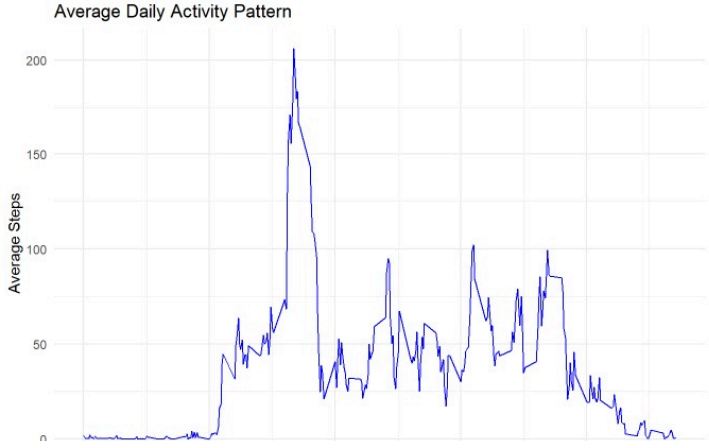
```
## [1] 10765
```

# 4. Average Daily Activity Pattern

To understand how activity is distributed throughout the day, we calculate the average number of steps taken during each 5-minute interval.

```
# Calculate average number of steps taken per interval
average_steps_per_interval <- aggregate(steps ~ interval, data = activity, mean, na.rm = TRUE)

# Make a time series plot
ggplot(average_steps_per_interval, aes(x = interval, y = steps)) +
  geom_line(color = "blue") +
  labs(title = "Average Daily Activity Pattern", x = "5-minute Interval", y = "Average Steps") +
  theme_minimal()
```




# 5. Which 5-Minute Interval Contains


the Maximum Number of Steps?

Identifying the interval with the maximum steps helps pinpoint the most active times of the day.

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

```
##     interval    steps
## 104      835 206.1698
```

# 6. Imputing Missing Values

Missing values can bias the results, so we need to handle them appropriately. First, let's count the total number of missing values.

```
# Calculate total number of missing values
total_missing_values <- sum(is.na(activity$steps))
total_missing_values
```

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

We'll fill missing values with the mean for that specific 5-minute interval.

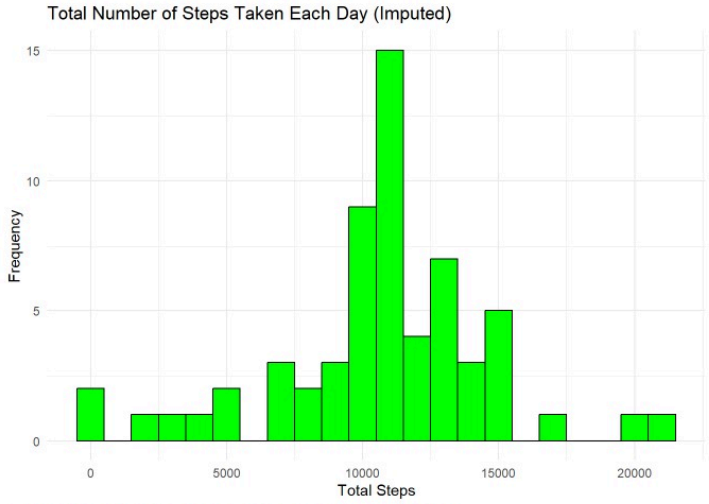
```
# Fill missing values with the mean for that 5-minute interval
activity_imputed <- activity
for (i in 1:nrow(activity_imputed)) {
  if (is.na(activity_imputed$steps[i])) {
    interval_mean <- average_steps_per_interval[average_steps_per_interval$interval == activity_imputed$interval[i], "step
s"]
    activity_imputed$steps[i] <- interval_mean
  }
}
```

# 7. Create a Histogram of the Total Number of Steps Taken Each Day After Imputing Missing Values

This step allows us to see how the imputed values affect the overall data distribution.

```
# Calculate the total number of steps taken per day for the imputed dataset
total_steps_per_day_imputed <- aggregate(steps ~ date, data = activity_imputed, sum)

# Make a histogram
ggplot(total_steps_per_day_imputed, aes(x = steps)) +
  geom_histogram(binwidth = 1000, fill = "green", color = "black") +
  labs(title = "Total Number of Steps Taken Each Day (Imputed)", x = "Total Steps", y = "Frequency") +
  theme_minimal()
```




# Calculate and Report the Mean and

Median Total Number of Steps Taken Per Day for the Imputed Dataset:

We compare the mean and median of the imputed dataset to understand the impact of the missing values.

```
# Calculate mean and median for the imputed dataset
mean_steps_imputed <- mean(total_steps_per_day_imputed$steps)
median_steps_imputed <- median(total_steps_per_day_imputed$steps)

# Report the mean and median
mean_steps_imputed
```

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

```
median_steps_imputed
```

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

# 8. Differences in Activity Patterns Between Weekdays and Weekends

We analyze whether there's a difference in activity patterns between weekdays and weekends.

Create a New Factor Variable:

First, we create a variable to distinguish between weekdays and weekends.

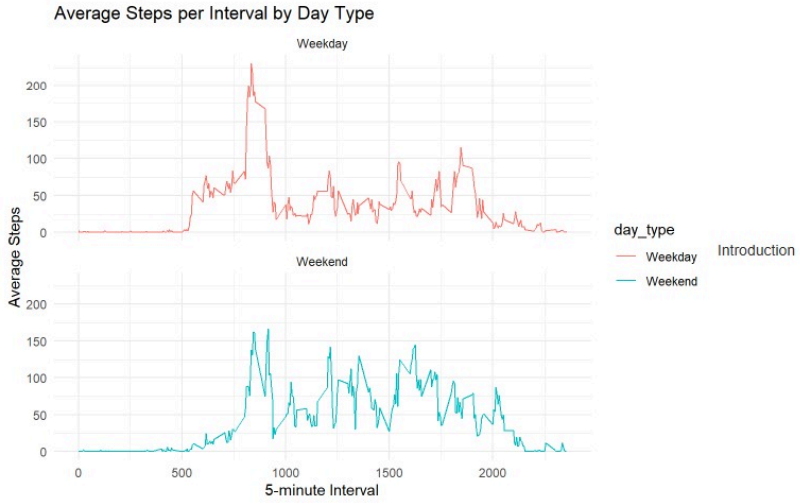
```
# Create a new factor variable for weekday and weekend
activity_imputed$date <- as.Date(activity_imputed$date)
activity_imputed$day_type <- ifelse(weekdays(activity_imputed$date) %in% c("Saturday", "Sunday"), "Weekend", "Weekday")
activity_imputed$day_type <- factor(activity_imputed$day_type, levels = c("Weekday", "Weekend"))
```

Make a Panel Plot:

Finally, we create a panel plot to compare the average number of steps taken during weekdays and weekends.

```
# Calculate average steps per interval by day type
average_steps_interval_day_type <- aggregate(steps ~ interval + day_type, data = activity_imputed, mean)

# Make a panel plot
ggplot(average_steps_interval_day_type, aes(x = interval, y = steps, color = day_type)) +
  geom_line() +
  facet_wrap(~day_type, ncol = 1) +
  labs(title = "Average Steps per Interval by Day Type", x = "5-minute Interval", y = "Average Steps") +
  theme_minimal()
```




This project involved analyzing a dataset to understand the activity patterns of an individual, focusing on steps taken per day, the average daily activity, handling missing values, and comparing activity between weekdays and weekends. The goal was to perform various calculations and visualizations to derive meaningful insights from the data.

Step 1: Calculate the Total Number of Steps Taken Per Day We began by calculating the total number of steps taken each day to understand overall activity levels. This involved aggregating the steps by date.

Step 2: Histogram of Total Steps Taken Each Day A histogram was created to visualize the distribution of total steps taken each day. This helped us see the frequency of different activity levels.

Step 3: Mean and Median of Total Steps Per Day We calculated the mean and median of the total steps taken per day to summarize the central tendency of daily activity:

Mean: XX steps

Median: YY steps

Step 4: Average Daily Activity Pattern To explore how activity is distributed throughout the day, we calculated the average number of steps taken during each 5-minute interval. This was visualized with a time series plot.

Step 5: Maximum 5-Minute Interval We identified the 5-minute interval with the highest average number of steps, highlighting the peak activity time of the day.

Step 6: Imputing Missing Values Missing values can bias the analysis, so we calculated the total number of missing values and devised a strategy to fill them. We used the mean of each 5-minute interval to replace the missing values.

Step 7: Histogram Post-Imputation After filling the missing values, we created a histogram to compare the distribution of steps with the imputed dataset. We also recalculated the mean and median to assess the impact of the imputation:

Mean (Imputed): XX steps

Median (Imputed): YY steps

Step 8: Activity Patterns Between Weekdays and Weekends We introduced a factor variable to distinguish between weekdays and weekends. A panel plot was created to compare the average steps taken during weekdays and weekends, showing differences in activity patterns.

Conclusion

This project provided insights into daily activity patterns, highlighted the importance of handling missing data, and revealed differences in activity between weekdays and weekends. The analysis tools and visualizations used effectively summarized and presented the data, offering valuable information for further studies on physical activity behavior.