Reproducible Research: Peer Assessment 1

Yanan Dong

2023-08-17

##Assignment Instructions 1. Format: a single R markdown document that can be processed by knitr and be transformed into an HTML file

Questions: 1.Loading and preprocessing the data 2.Histogram of mean total number of steps taken per day and report both mean and median steps taken each day 3.Time series plot of the average daily data pattern, output which 5-minute interval that, on average, contains the maximum number of steps 4.Missing data imputation, output histogram of the total number of steps taken each day after imputation. 5.Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

Loading and preprocessing the data

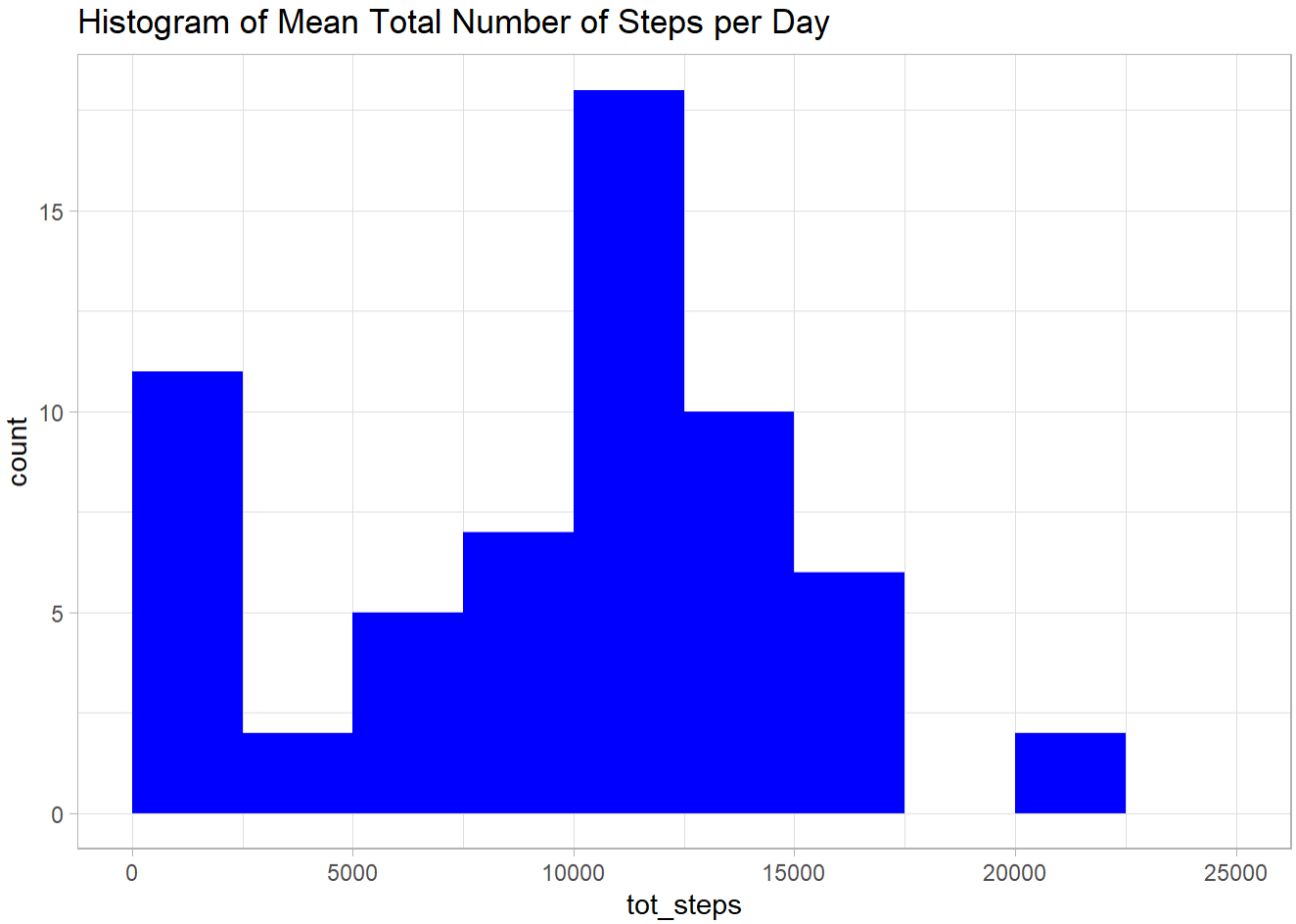
```
data <- read_csv("activity.csv", col_types = cols(steps = col_number(),
    date = col_date(format = "%Y-%m-%d"),
    interval = col_number()), na = "NA")

dim(data)
```

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

```
# The variables included in this dataset are:
# steps: Number of steps taking in a 5-minute interval (missing values are coded as NA)
# date: The date on which the measurement was taken in YYYY-MM-DD format
# interval: Identifier for the 5-minute interval in which measurement was taken
# The dataset is stored in a comma-separated-value (CSV) file and there are a total of 17,568 observations in this dataset.
```

What is mean total number of steps taken per day?

```
#summarise data as needed, ignore the missing values in the dataset
d1<-data %>% group_by(date) %>%
  summarise(tot_steps = sum(steps, na.rm=T),
            mean_steps = mean(steps, na.rm=T),
            median_steps = median(steps, na.rm=T))

#Histogram of the total number of steps taken each day
ggplot(d1, aes(x = tot_steps))+
  geom_histogram(fill = "blue", breaks = seq(0, 25000, by = 2500))+
  labs(title = "Histogram of Mean Total Number of Steps per Day",
       ylab = "Frequency", xlab = "Total Steps per Day")+
  theme_light()
```



The mean and median of the total number of steps taken per day are:

```
paste("mean is :", round(mean(d1$tot_steps, na.rm = T),3))
```

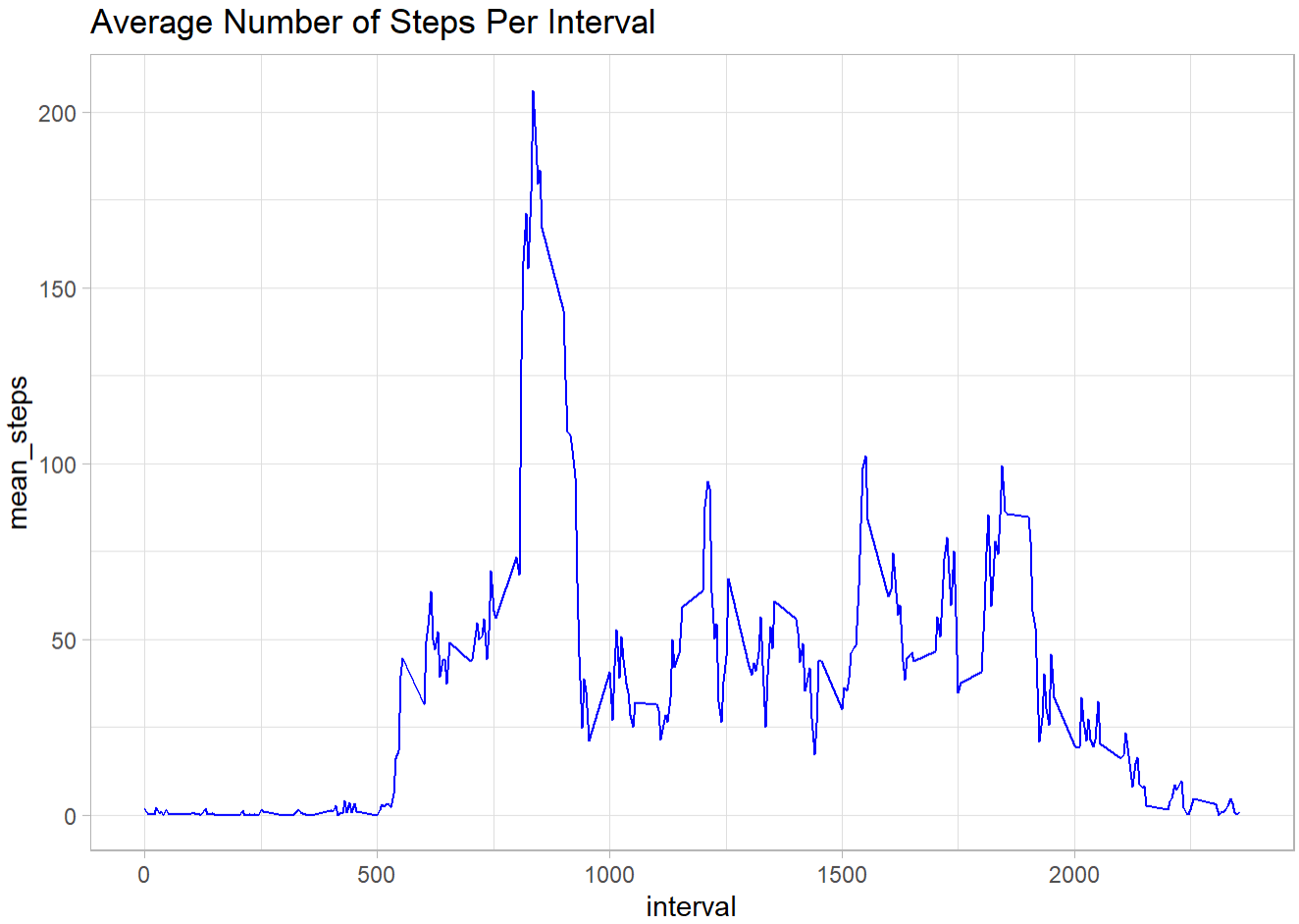
```
## [1] "mean is : 9354.23"
```

```
paste("median is :", round(median(d1$tot_steps, na.rm = T),3))
```

```
## [1] "median is : 10395"
```

What is the average daily data pattern?

```
# Make a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)
d2<-data %>% group_by(interval) %>%
  summarise(tot_steps = sum(steps, na.rm=T),
            mean_steps = mean(steps, na.rm=T),
            median_steps = median(steps, na.rm=T))

ggplot(d2, aes(interval, mean_steps)) +
  geom_line(col = "blue")+
  labs(title = "Average Number of Steps Per Interval",
       ylab = "Number of Steps", xlab = "Interval")+
  theme_light()
```



Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

```
paste("On average, inverval",d2[which.max(d2$mean_steps), ]$interval, "contains the maximum number of steps")
```

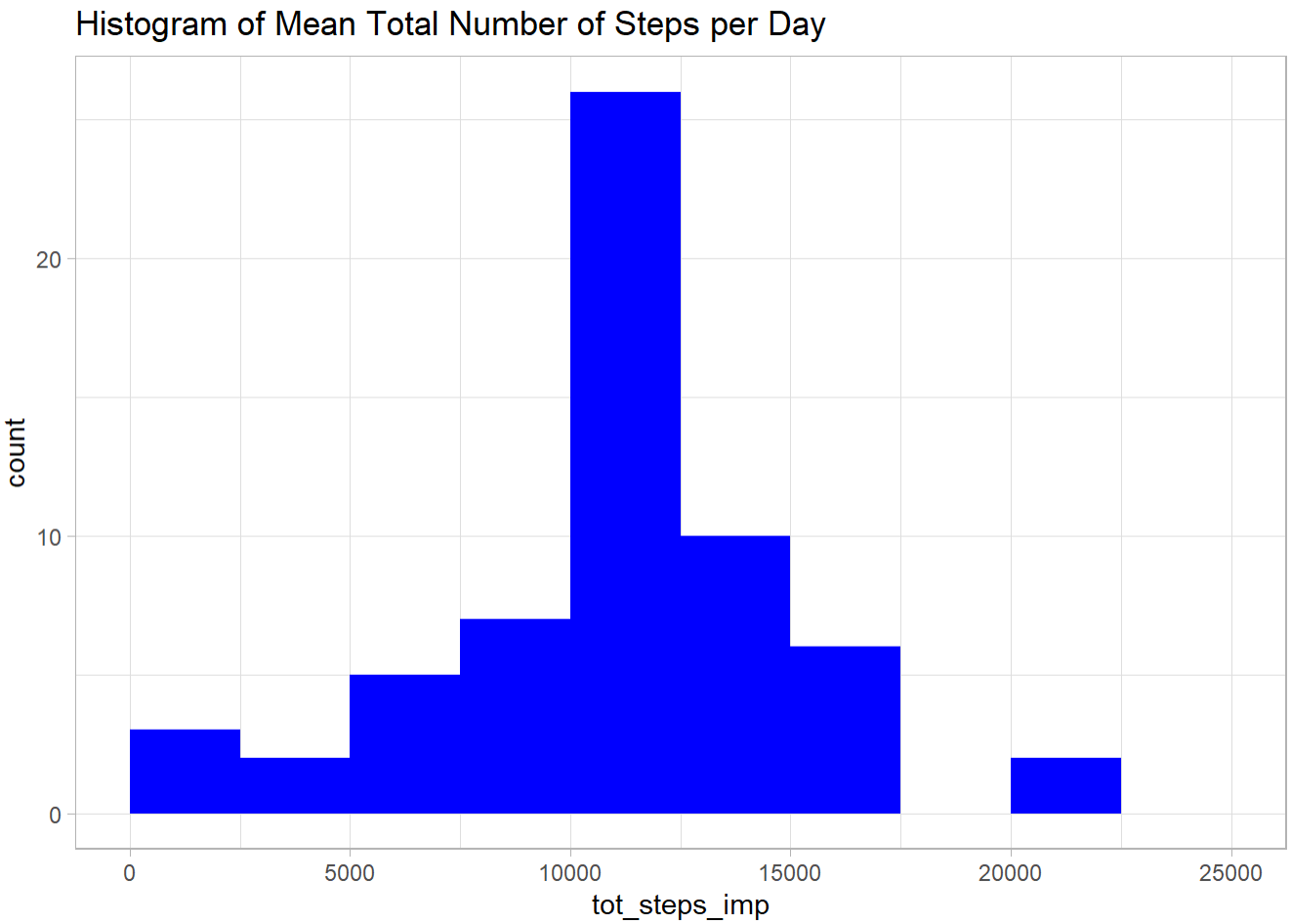
```
## [1] "On average, inverval 835 contains the maximum number of steps"
```

Imputing missing values

```
# 1.Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)
sum(is.na(data$steps))
```

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

```
# 2. Imputation, fill in missng data, then histogram of the total number of steps taken each day and Calculate and report the mean an
d median total number of steps taken per day.

data_imp<-data %>%
  group_by(interval) %>%
  mutate(steps_imp = replace_na(steps, mean(steps, na.rm = TRUE)))

d3<-data_imp %>% group_by(date) %>%
  summarise(tot_steps_imp = sum(steps_imp, na.rm=T),
            mean_steps_imp = mean(steps_imp, na.rm=T),
            median_steps_imp = median(steps_imp, na.rm=T))

#Histogram after imputation
ggplot(d3, aes(x = tot_steps_imp))+
  geom_histogram(fill = "blue", breaks = seq(0, 25000, by = 2500))+
  labs(title = "Histogram of Mean Total Number of Steps per Day",
       ylab = "Frequency", xlab = "Total Steps per Day")+
  theme_light()
```



The mean and median of the total number of steps taken per day are:

```
paste("mean is :", round(mean(d3$tot_steps_imp, na.rm = T),3))
```

```
## [1] "mean is : 10766.189"
```

```
paste("median is :", round(median(d3$tot_steps_imp, na.rm = T),3))
```

```
## [1] "median is : 10766.189"
```

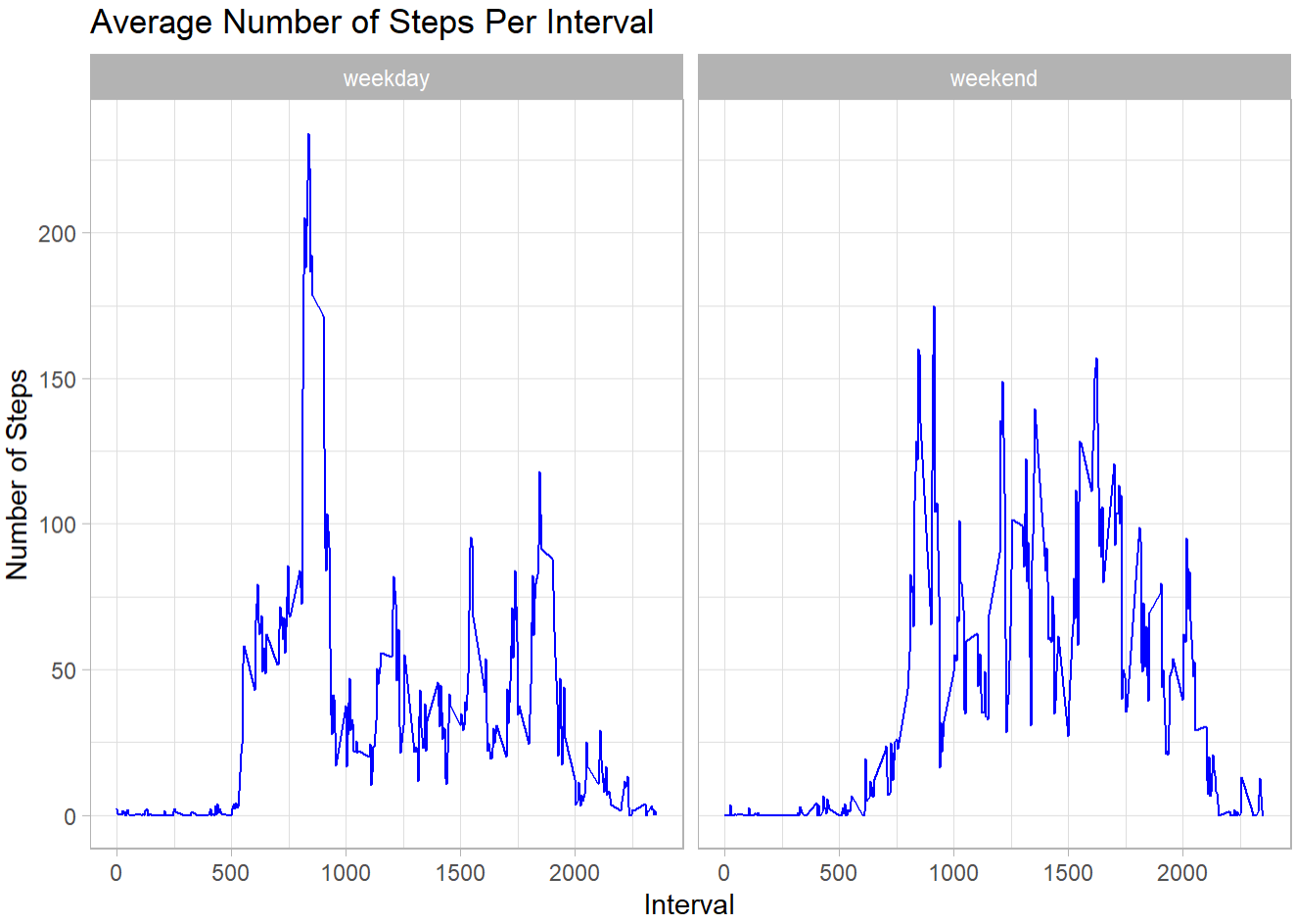
Value differes from the first assisgnment. After imputation with mean values per day, the distribution looks a lot more closer to a normal disctibution.

Are there differences in data patterns between weekdays and weekends?

```
# For this part the weekdays() function may be of some help here. Use the dataset with the filled-in missing values for this part.

# 1. Create a new factor variable in the dataset with two levels – “weekday” and “weekend” indicating whether a given date is a week day or weekend day.
data_imp<-data_imp %>%
  mutate(weekday = as.factor(weekdays(date))
         , weekday_flag = ifelse(weekday == "Saturday"| weekday =="Sunday", "weekend", "weekday"))

# Make a panel plot containing a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged a cross all weekday days or weekend days (y-axis). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

d4<-data_imp %>% group_by(weekday_flag, interval) %>%
  summarise(tot_steps = sum(steps, na.rm=T),
            mean_steps = mean(steps, na.rm=T))
```

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

```
ggplot(d4, aes(interval, mean_steps)) +
  geom_line(col = "blue")+
  facet_grid(~weekday_flag)+
  labs(title = "Average Number of Steps Per Interval")+
  xlab("Interval")+ ylab("Number of Steps")+
  theme_light()
```


Average Number of Steps Per Interval
weekday
weekend
Number of Steps
0
50
100
150
200
0
500
1000
1500
2000
Interval