Reproducible Research - Course Project 1

Prabeeti Bulani

9/18/2019

Introduction

Introduction

It is now possible to collect a large amount of data about personal movement using activity monitoring devices such as a Fitbit, Nike Fuelband, or Jawbone Up. These type of devices are part of the “quantified self” movement – a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. But these data remain under-utilized both because the raw data are hard to obtain and there is a lack of statistical methods and software for processing and interpreting the data.

This assignment makes use of data from a personal activity monitoring device. This device collects data at 5 minute intervals through out the day. The data consists of two months of data from an anonymous individual collected during the months of October and November, 2012 and include the number of steps taken in 5 minute intervals each day.

The variables included in this dataset are:

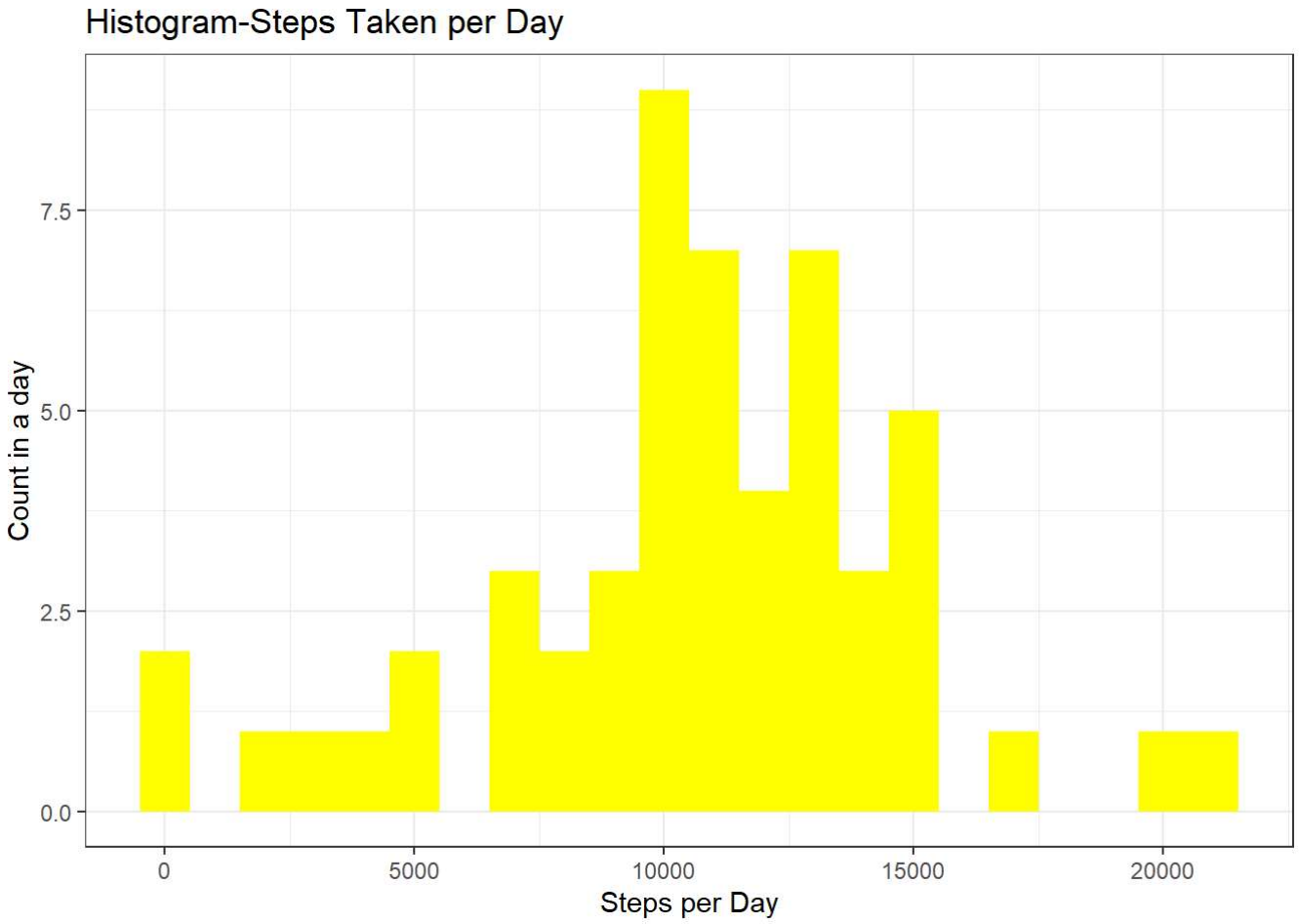
steps: Number of steps taking in a 5-minute date: The date on which the measurement was taken in YYYY-MM-DD format interval: Identifier for the 5-minute interval in which measurement was taken The dataset is stored in a comma-separated-value (CSV) file and there are a total of 17,568 observations in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

1. Loading and preprocessing the data

```
activity_data <- read.csv('activity.csv', header = TRUE, sep = ",",colClasses=c("numeric", "character", "nu
meric"))

activity_data$date <- as.Date(activity_data$date, format = "%m/%d/%Y")
activity_data$interval <- as.factor(activity_data$interval)

## What is mean total number of steps taken per day?
total_steps_1day <- aggregate(steps ~ date, activity_data, sum)
colnames(total_steps_1day) <- c("date","steps")
```

2.Make a histogram of the total number of steps taken each day

```
ggplot(total_steps_1day, aes(x=steps)) +  geom_histogram(fill="yellow",binwidth = 1000) +  xlab("Steps per
 Day") +  ylab("Count in a day") +
  ggtitle("Histogram-Steps Taken per Day")+ theme_bw()
```



3.Mean and median number of steps taken each day

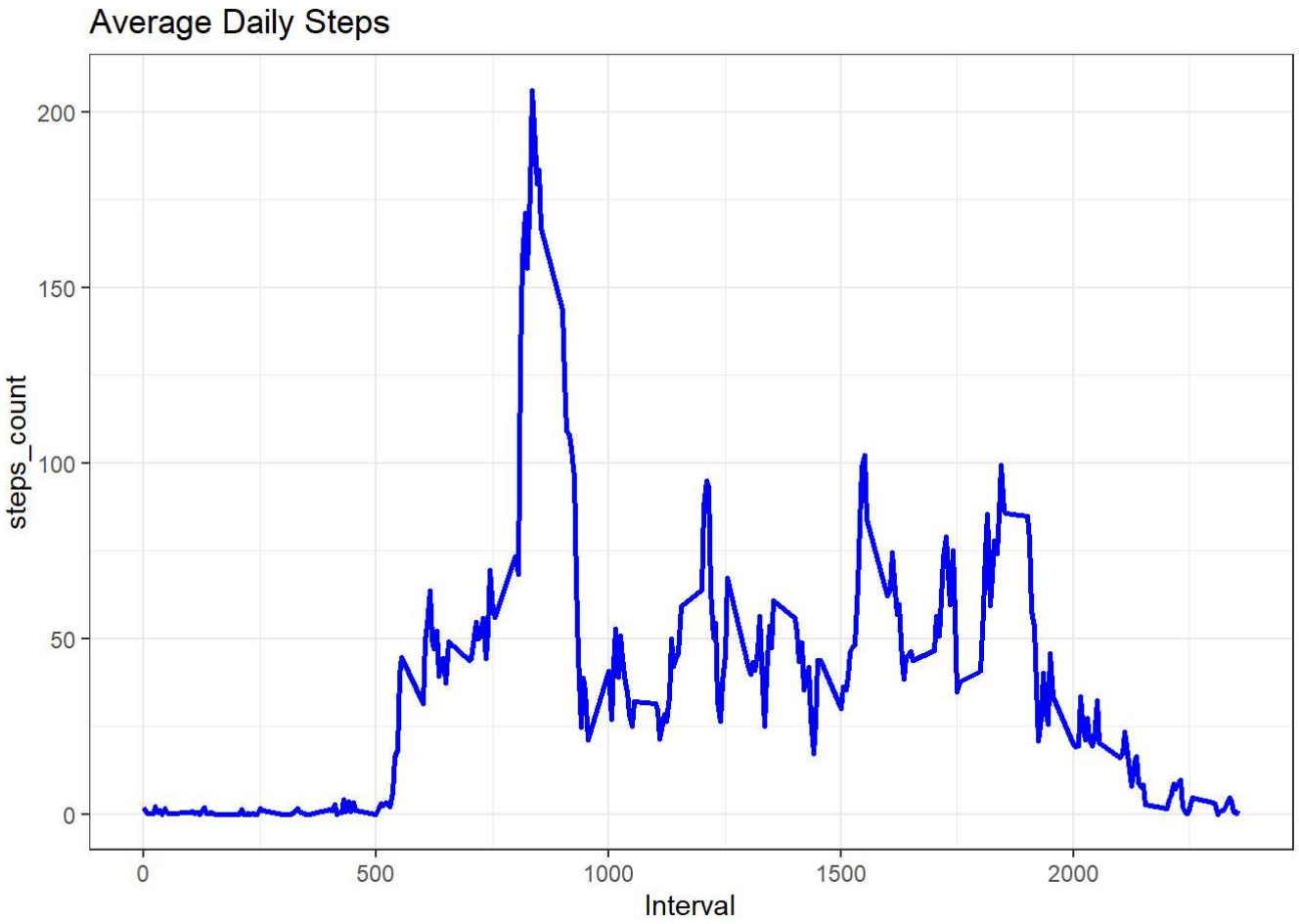
```
mean_steps_per_day   <- mean(total_steps_1day$steps, na.rm=TRUE)
median_steps_per_day <- median(total_steps_1day$steps, na.rm=TRUE)
mean_steps_per_day
median_steps_per_day
```

```
## [1] 10766.19
## [1] 10765
```

4.Time series plot of the average number of steps taken

#######What is the average daily activity pattern?

```
daily_activity_pattern <- aggregate(activity_data$steps,
                                   by = list(interval = activity_data$interval),
                                   FUN=mean, na.rm=TRUE)
#convert to integers this helps in plotting
daily_activity_pattern$interval <- as.integer(levels(daily_activity_pattern$interval)[daily_activity_patter
n$interval])
colnames(daily_activity_pattern) <- c("interval", "steps")

ggplot(daily_activity_pattern, aes(x=interval, y=steps)) +
        geom_line(color="blue", size=1) +  xlab("Interval") +  ylab("steps_count") +
  ggtitle("Average Daily Steps")+   theme_bw()
```



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

```
max_5minute_interval <- daily_activity_pattern[which.max(daily_activity_pattern$steps),]
max_5minute_interval
```

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

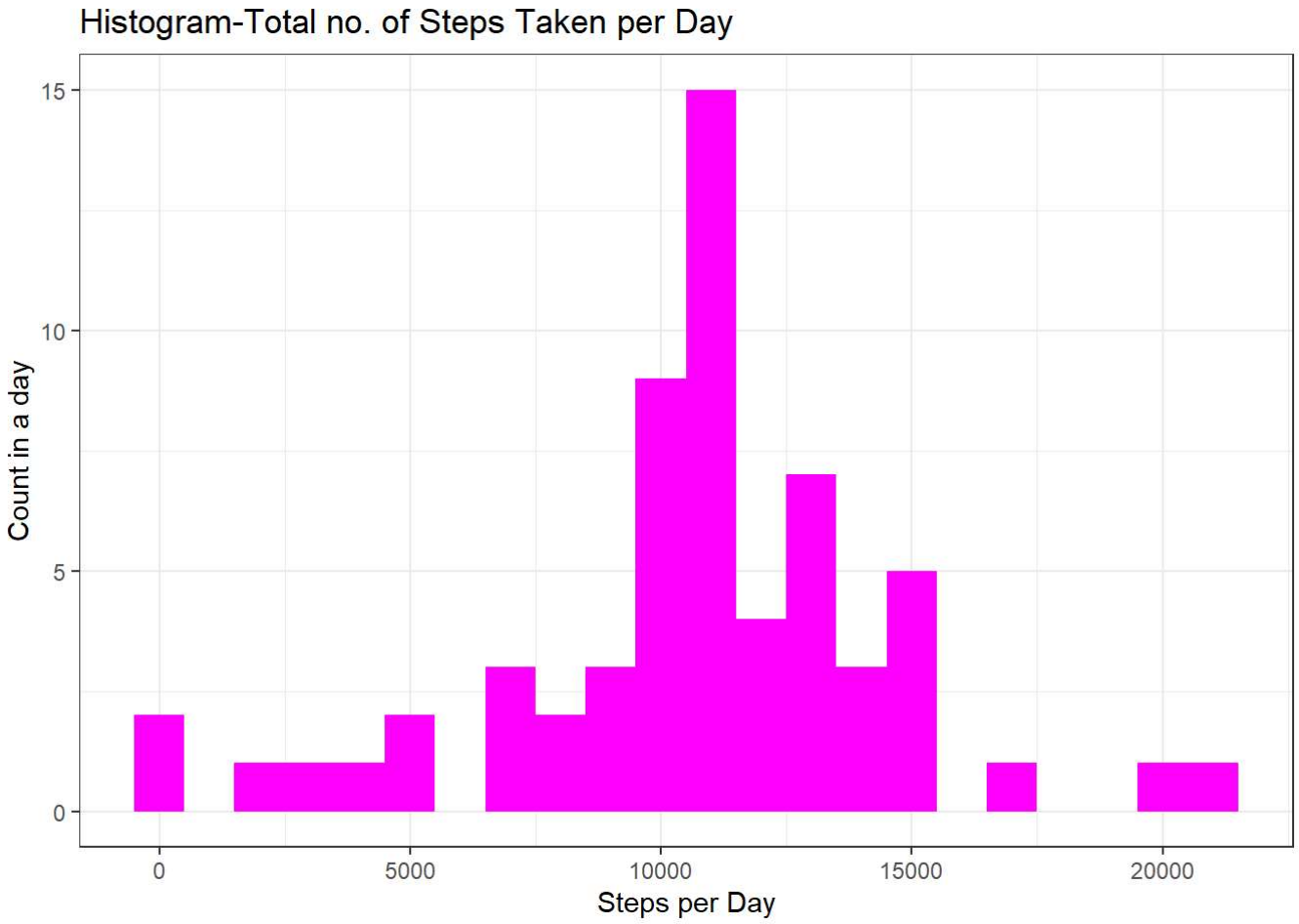
6.Code to describe and show a strategy for imputing missing data

```
#######Imputing missing values
#######Calculate and report the total number of missing values in the dataset
sum(is.na(activity_data))

#######Create a new dataset that is equal to the original dataset but with the missing data filled in.Means for the 5-minute intervals are used as fillers for missing values.
activity_data <- merge(activity_data, daily_activity_pattern, by = "interval", suffixes = c("", ".y"))
step_count <- is.na(activity_data$steps)
activity_data$steps[step_count] <- activity_data$steps.y[step_count]
activity_data <- activity_data[, c(1:3)]
str(activity_data)
```

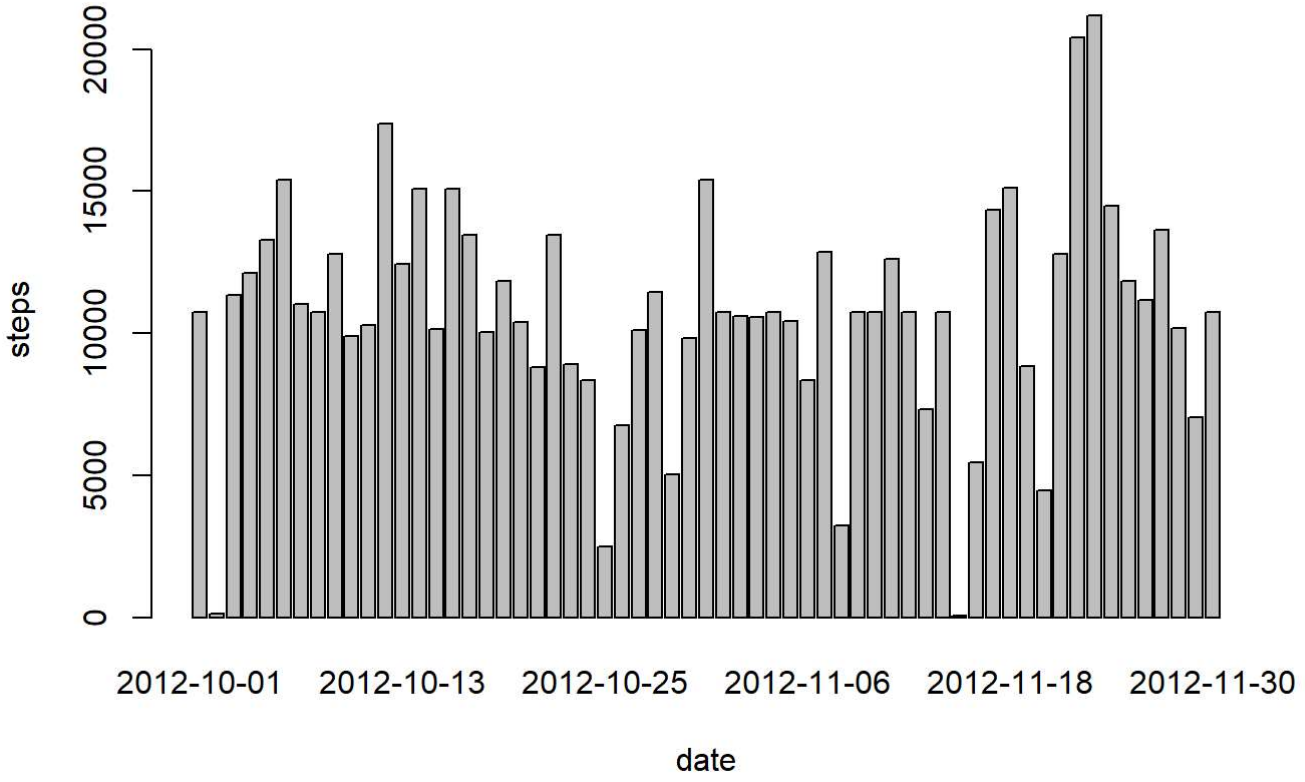
```
## [1] 2304
## 'data.frame':    17568 obs. of  3 variables:
##  $ interval: Factor w/ 288 levels "0","5","10","15",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ steps   : num  1.72 0 0 0 0 ...
##  $ date    : Date, format: "2012-10-01" "2012-11-23" ...
```

7.Histogram of the total number of steps taken each day after missing values are imputed

```
#######Make a histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day. Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

steps_taken_in_day <- aggregate(steps ~ date, data = activity_data, FUN = sum)


ggplot(steps_taken_in_day, aes(x=steps)) +  geom_histogram(fill="magenta",binwidth = 1000) +  xlab("Steps per Day") +  ylab("Count in a day") +
  ggtitle("Histogram-Total no. of Steps Taken per Day")+ theme_bw()
```




```
barplot(steps_taken_in_day$steps, names.arg = steps_taken_in_day$date, xlab = "date", ylab = "steps")
```

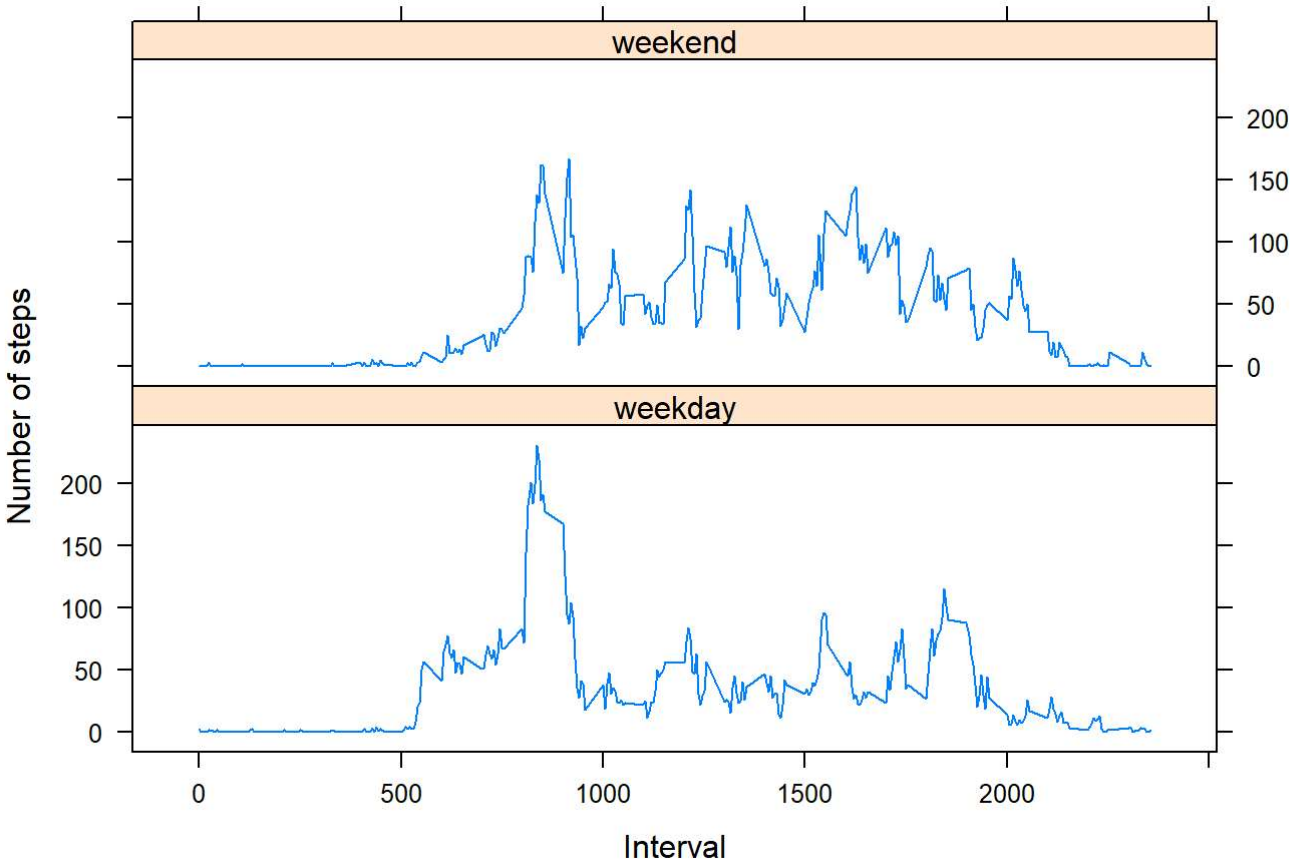



```
mean2 <- mean(steps_taken_in_day$steps)
median2 <- median(steps_taken_in_day$steps)
mean2
median2
# Mean and median are same now after imputing data earlier it was different
# Mean and Median After Imputing Data - 10766.19 and 10766.19
# Mean and Median Before Imputing Data - 10766.19 and 10765
```

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

8.Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

```
# Are there differences in activity patterns between weekdays and weekends?
#######Create a new factor variable in the dataset with two levels - "weekday" and "weekend" indicating whe
ther a given date is a weekday or weekend day.
#######Make a panel plot containing a time series plot of the 5-minute interval (x-axis) and the average nu
mber of steps taken, averaged across all weekday days or weekend days (y-axis).


activity_data$weekdays <- factor(format(activity_data$date, "%A"))
levels(activity_data$weekdays)


levels(activity_data$weekdays) <- list(weekday = c("Monday", "Tuesday",
                                           "Wednesday",
                                           "Thursday", "Friday"),
                                 weekend = c("Saturday", "Sunday"))
levels(activity_data$weekdays)


table(activity_data$weekdays)

days_average <- aggregate(activity_data$steps,
                      list(interval = as.numeric(as.character(activity_data$interval)),
                           weekdays = activity_data$weekdays),
                      FUN = "mean")
names(days_average)[3] <- "mean_Steps"

xyplot(days_average$mean_Steps ~ days_average$interval | days_average$weekdays,
       layout = c(1, 2), type = "l",
       xlab = "Interval", ylab = "Number of steps")
```




```
## [1] "Friday"    "Monday"    "Saturday"  "Sunday"    "Thursday"  "Tuesday"
## [7] "Wednesday"
## [1] "weekday" "weekend"
##
## weekday weekend
##   12960    4608
```