instructions

TSAI -YI FAN

2018年4月16日

knitr::opts_chunk$set(echo = TRUE)

Loading and preprocessing the data

- Load the data
- Process/transform the data into a format suitable for your analysis.

```
tbl <- read.csv(file = "activity.csv")
head(tbl)
```

```
##   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
```

What is mean total number of steps taken per day?

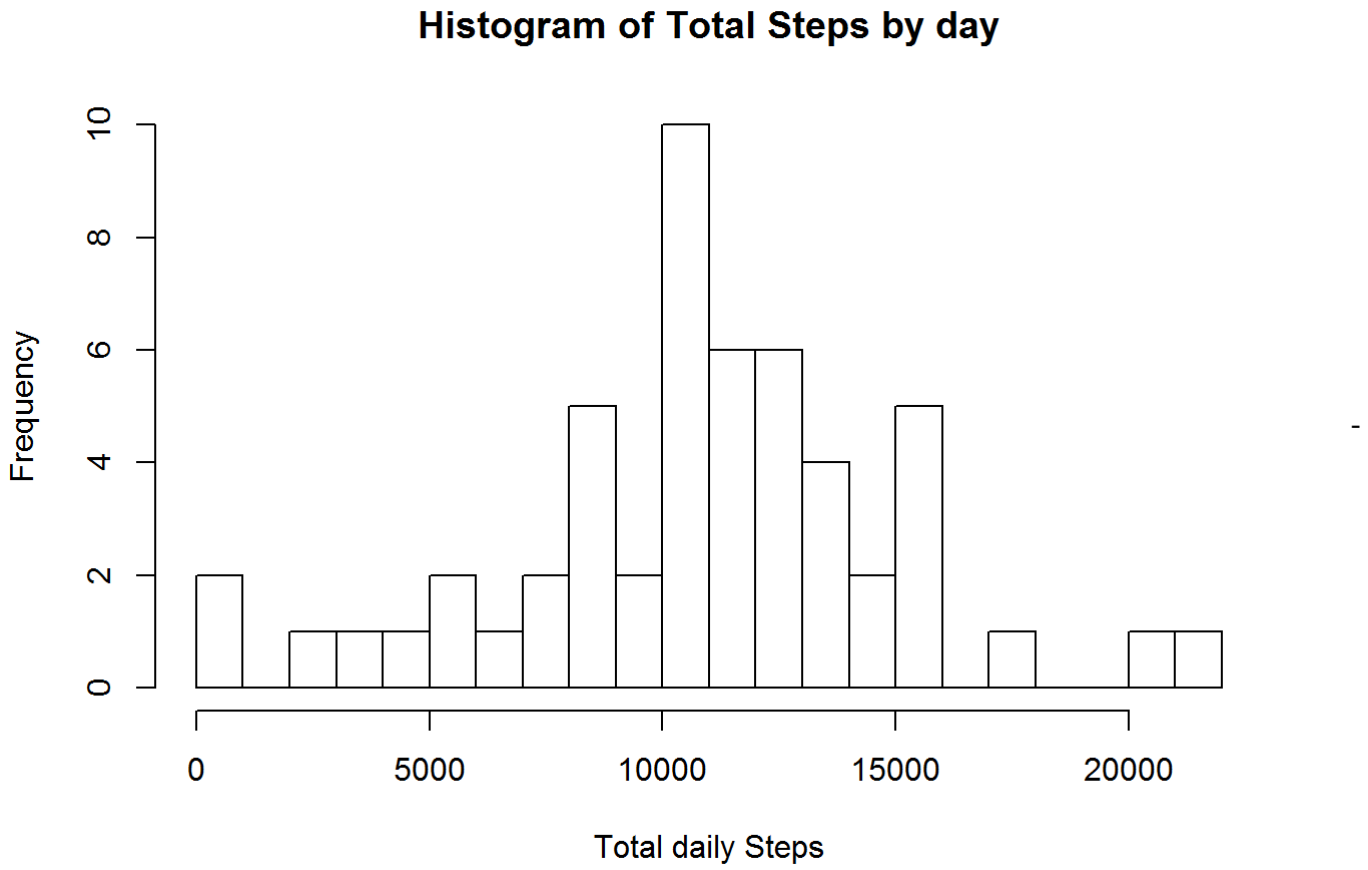
- Calculate the total number of steps taken per day

```
daily_steps <- tapply(tbl$steps, tbl$date, sum)
head(daily_steps)
```

```
## 2012-10-01 2012-10-02 2012-10-03 2012-10-04 2012-10-05 2012-10-06
##         NA        126      11352      12116      13294      15420
```

- Make a histogram of the total number of steps taken each day

```
hist(daily_steps,breaks = 30, xlab = "Total daily Steps", main = "Histogram of Total Steps by
 day")
```



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

```
median(daily_steps,na.rm = T)
```

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

```
mean(daily_steps,na.rm = T)
```

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

What is the average daily activity 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).

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

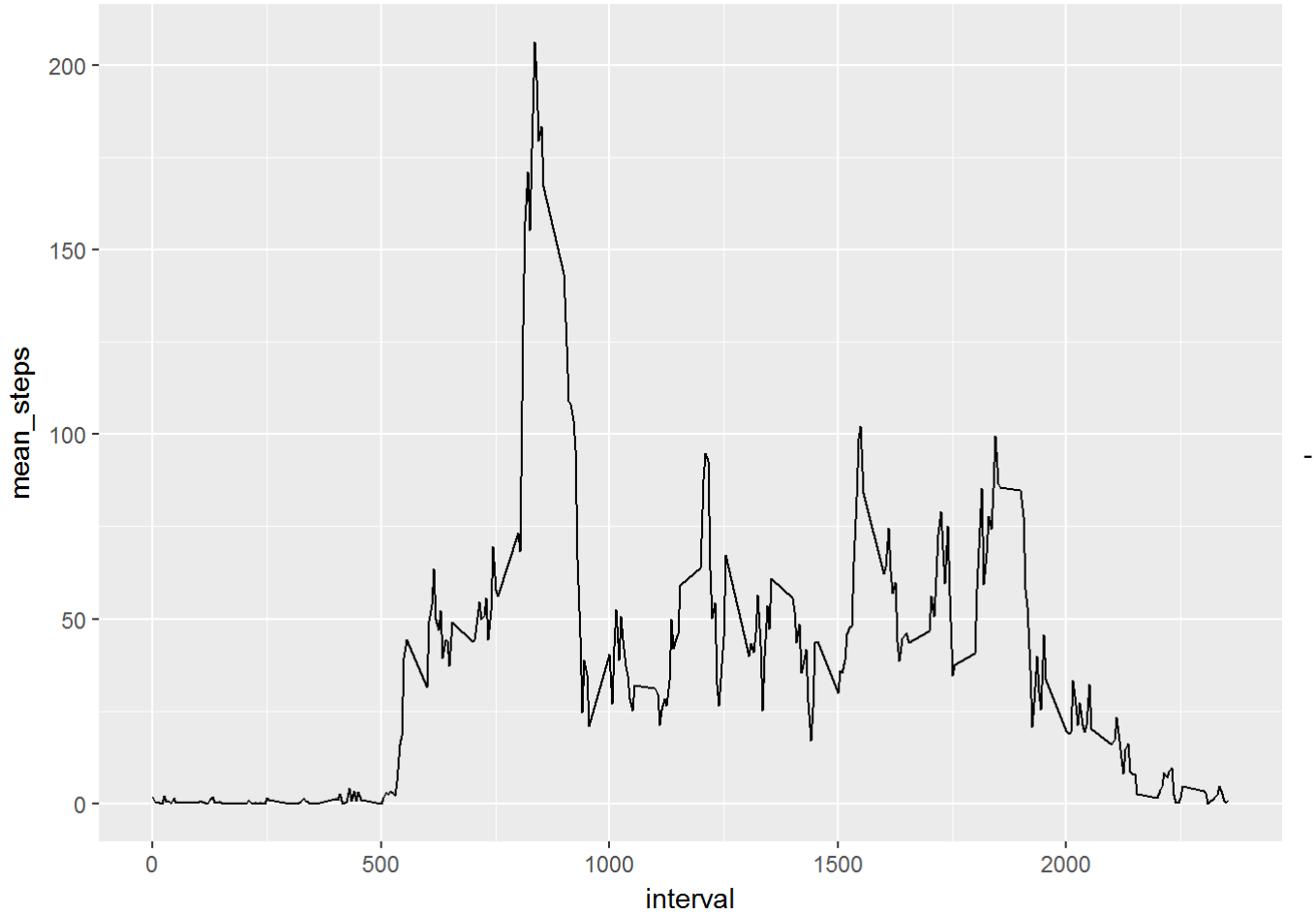
```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```
tbl_interval <- tbl %>% na.omit() %>% group_by(interval) %>% summarize(mean_steps= mean(step
s))
ggplot(tbl_interval, aes(x=interval, y=mean_steps))+ geom_line()
```



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

```
max_5_minute <- tbl_interval[which(tbl_interval$mean_steps == max(tbl_interval$mean_steps)),]
print(max_5_minute)
```

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

Imputing missing values

Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

```
sum(is.na(tbl))
```

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

- Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to be sophisticated. For example, you could use the mean/median for that day, or the mean for that 5-minute interval, etc.

```
tbl_mgr <- merge(tbl, tbl_interval, by="interval")
tbl_mgr$steps[is.na(tbl_mgr$steps)] <- tbl_mgr$mean_steps[is.na(tbl_mgr$steps)]
tbl_mgr <- tbl_mgr[order(tbl_mgr$date),]
tbl_mgr <- tbl_mgr[,-c(4)]
head(tbl_mgr)
```

```
##     interval     steps       date
## 1          0 1.7169811 2012-10-01
## 63         5 0.3396226 2012-10-01
## 128       10 0.1320755 2012-10-01
## 205       15 0.1509434 2012-10-01
## 264       20 0.0754717 2012-10-01
## 327       25 2.0943396 2012-10-01
```

Create a new dataset that is equal to the original dataset but with the missing data filled in.

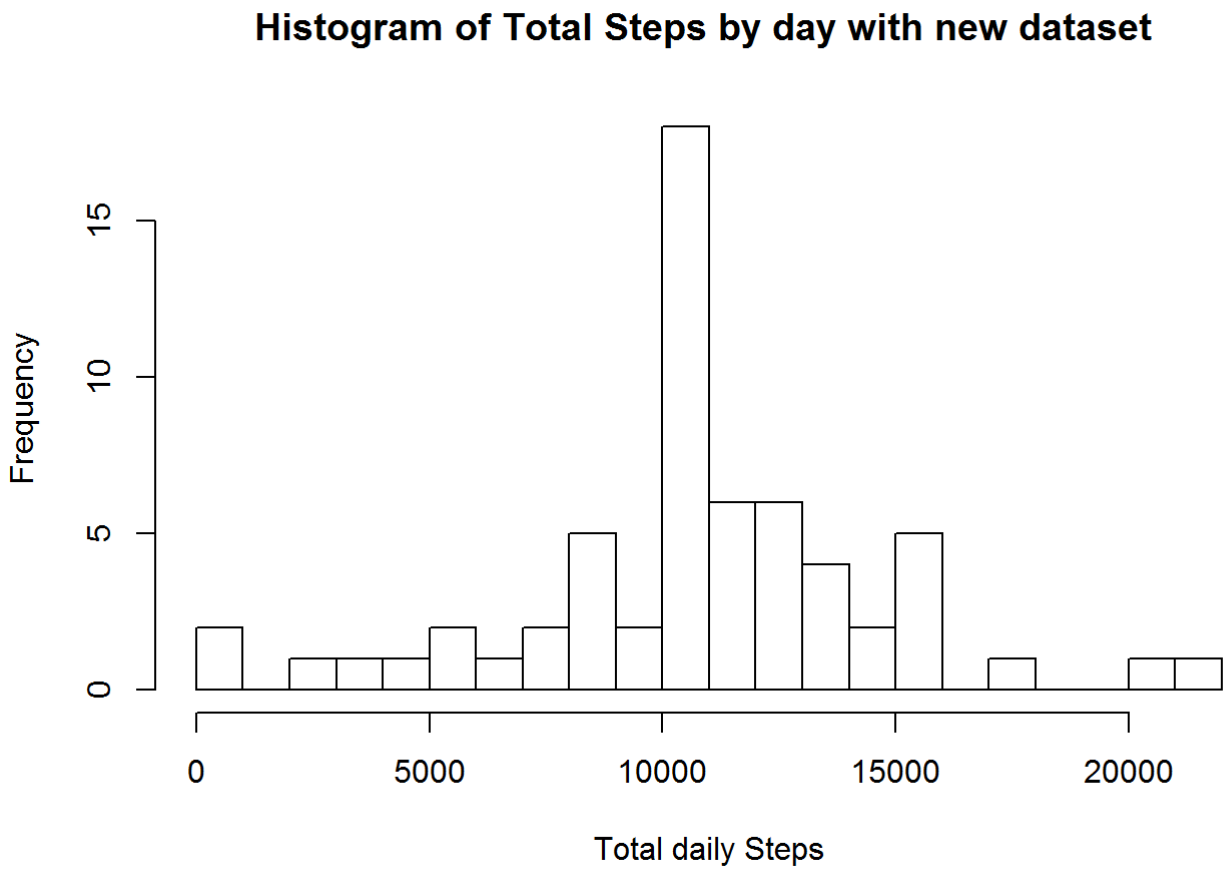
```
tbl_2 <- tbl_mgr[,c(2,1,3)]
head(tbl_2)
```

```
##         steps interval       date
## 1   1.7169811        0 2012-10-01
## 63  0.3396226        5 2012-10-01
## 128 0.1320755       10 2012-10-01
## 205 0.1509434       15 2012-10-01
## 264 0.0754717       20 2012-10-01
## 327 2.0943396       25 2012-10-01
```

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.

- The histogram differs from the very first verison with missing values.
- The median value has been increased and equals to the mean value with the new dataset.

```
daily_steps2 <- tapply(tbl_2$steps, tbl_2$date, sum)
hist(daily_steps2,breaks = 30, xlab = "Total daily Steps", main = "Histogram of Total Steps b
y day with new dataset ")
```




```
median(daily_steps2)
```

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

```
mean(daily_steps2)
```

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

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 whether a given date is a weekday or weekend day.

```
tbl_2$whichday <- ifelse(weekdays(as.Date(tbl_2$date)) %in% c("Samstag", "Sonntag"), "weeken
d", "weekday")
tbl_2$whichday <- as.factor(tbl_2$whichday)
head(tbl_2)
```

```
##         steps interval       date whichday
## 1   1.7169811        0 2012-10-01  weekday
## 63  0.3396226        5 2012-10-01  weekday
## 128 0.1320755       10 2012-10-01  weekday
## 205 0.1509434       15 2012-10-01  weekday
## 264 0.0754717       20 2012-10-01  weekday
## 327 2.0943396       25 2012-10-01  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 across all weekday Makedays or weekend days (y-axis).

```
tbl_interval2 <- tbl_2 %>% group_by(interval, whichday) %>% summarize(mean_steps= mean(step
s))
ggplot(tbl_interval2, aes(x=interval, y=mean_steps))+
geom_line() +
facet_grid(whichday ~.) +
xlab("Interval") +
ylab("Mean Steps") +
ggtitle("Comparison of Average Number of Steps seprated by weekday and weekend")
```