Week2 C5 assignment

letspairup

6/21/2021

Load data and create data frame

- download data from source
- unzip it
- convert it to data frame

```
download.file('https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip',
              destfile = 'activity.zip', mode='wb')

unzip('activity.zip')

repdata <- read.csv('activity.csv', header = TRUE)
```

Sample data

```
knitr::kable(head(repdata))
```

steps	date	interval
NA	2012-10-01	0
NA	2012-10-01	5
NA	2012-10-01	10
NA	2012-10-01	15
NA	2012-10-01	20
NA	2012-10-01	25

Total number of steps by day

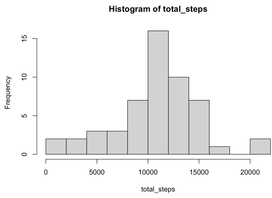
```
library('dplyr')

dailystep_count <- repdata %>%
  select(date, steps) %>%
  filter(!is.na(steps)) %>%
  group_by(date) %>%
  summarise(total_steps = sum(steps))

head(dailystep_count)
```

```
## # A tibble: 6 x 2
##   date       total_steps
##   <chr>            <int>
## 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
```

```
with(dailystep_count,
     hist(total_steps, breaks = 15))
```




```
summary(dailystep_count)
```

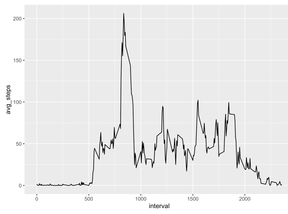
```
##      date           total_steps
##  Length:53          Min.   :   41
##  Class :character   1st Qu.: 8841
##  Mode  :character   Median :10765
##                     Mean   :10766
##                     3rd Qu.:13294
##                     Max.   :21194
```

High level summary of daily step counts. You can see **mean is 10766** and **median is 10765.**

Average daily activity pattern - Time series plot

```
library(ggplot2)
repdata_by_interval <- repdata %>% na.omit() %>%
                                   group_by(interval) %>% summarise(avg_steps= mean(steps))

ggplot(repdata_by_interval, aes(x=interval, y=avg_steps))+ geom_line()
```



Five minute inteval with max steps - across all days

```
repdata_by_interval[which.max(repdata_by_interval$avg_steps),]
```

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

Total number of missing values

```
sum(is.na(repdata))
```

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

Total number of missing values

```
sum(is.na(repdata))
```

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

Strategy to fill missing values

Here we are going to use daily mean to fill missing values.

```
fill_with_mean <- function(x)
{
  replace(x, is.na(x), mean(x, na.rm = TRUE))
}

complete_data <- repdata %>% mutate(steps = fill_with_mean(steps))

sum(is.na(complete_data))
```

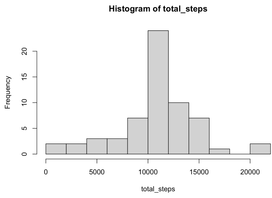
```
## [1] 0
```

complete_data is not having any missing values..!

Total number of steps by day and mean/median with complete data set

```
dailystep_count_with_complete_data <- complete_data %>%
  group_by(date) %>%
  summarise(total_steps = sum(steps))

with(dailystep_count_with_complete_data,
     hist(total_steps, breaks = 15))
```




```
summary(dailystep_count_with_complete_data)
```

```
##      date           total_steps
##  Length:61          Min.   :   41
##  Class :character   1st Qu.: 9819
##  Mode  :character   Median :10766
##                     Mean   :10766
##                     3rd Qu.:12811
##                     Max.   :21194
```

You can see **mean is 10766** and **median is 10766.** *Not much different from when we removed NA values.*

Activity pattern between weekdays and weekend

- Convert text based date column to Date
- Add new column day_category; marked row as either weekend or weekday based on the day of date.
- Group by data by day_category and interval
- Calculate average steps in avg_steps

```
str(repdata)
```

```
## '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 ...
```

```
repdata$date <- as.Date(repdata$date)
str(repdata)
```

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

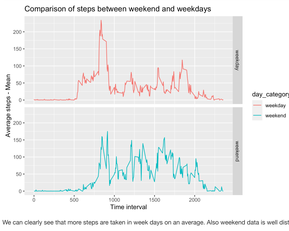
```
get_day_category <- function(x) {
  if("Saturday" == weekdays(x) | "Sunday" == weekdays(x))
  {
    return("weekend")
  }
  else
  {
    return("weekday")
  }
}

comparedata <- repdata %>% filter(!is.na(steps)) %>%
  mutate(day_category = sapply(date,get_day_category)) %>%
  select(steps,interval, day_category) %>%
  group_by(day_category, interval) %>%
  mutate(avg_steps = mean(steps, na.rm = TRUE))

#str(comparedata)
```

Plot time series graph

```
ggplot(comparedata, aes(x=interval, y=avg_steps, color=day_category)) + geom_line() +
facet_grid(day_category ~.) + xlab("Time interval") + ylab("Average steps - Mean") +
    ggtitle("Comparison of steps between weekend and weekdays")
```



We can clearly see that more steps are taken in week days on an average. Also weekend data is well distributed across active time intervals.