course project 1

Chris Teceno

2022-07-28

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

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

```
setwd("/Users/teceno/Desktop/r-reproducible-research/course-project-1")
activity<-read.csv("activity.csv")
library(lubridate)
```

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

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

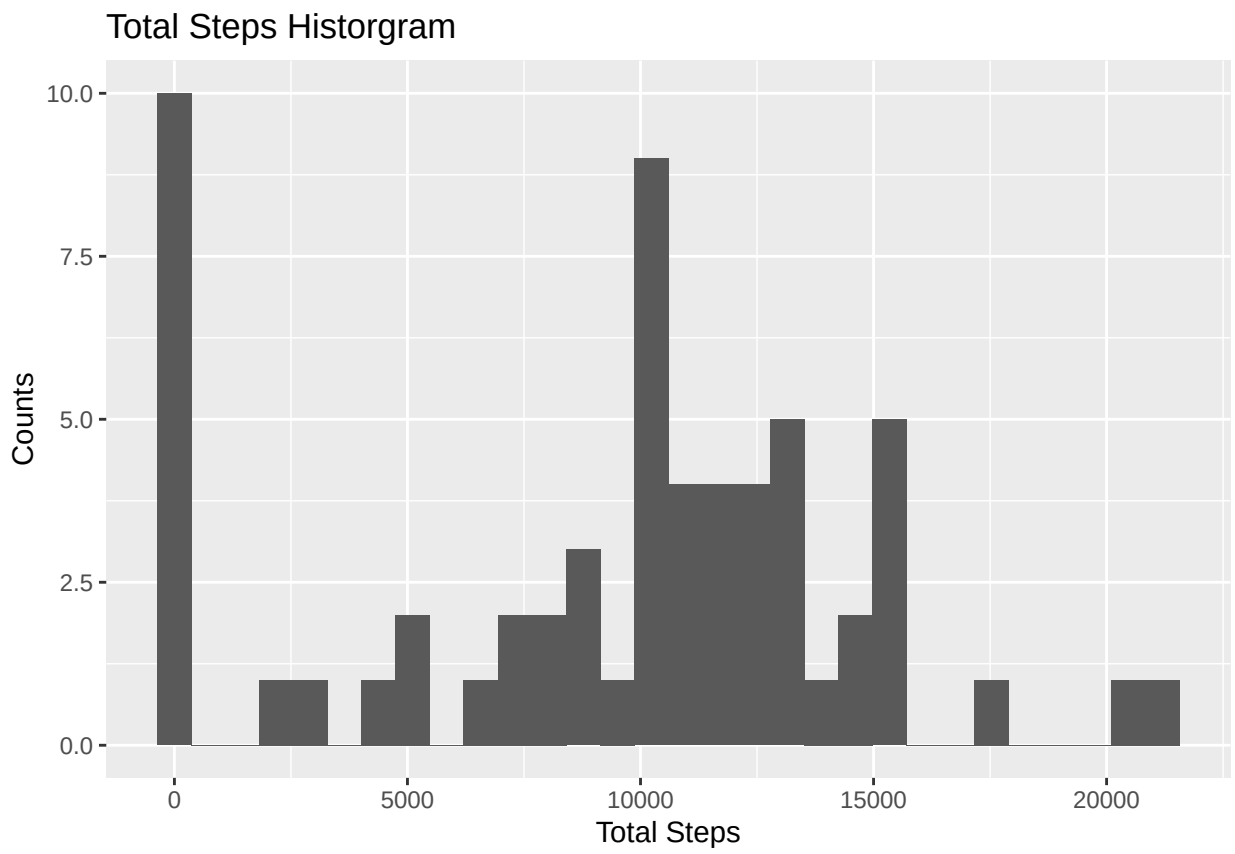
```
# make date a date object
activity$date<-ymd(activity$date)
#fill na with 0
activity[is.na(activity)] <-0
```

2. Histogram of the total number of steps taken each day

```
#import ggplot
library(ggplot2)
#make df with daily step total
```

```
dailySum <-data.frame(tapply(activity$steps,activity$date,sum,na.rm=TRUE))
#change column name
names(dailySum)[[1]]<-"TotalSteps"
qplot(dailySum$TotalSteps,geom="histogram",xlab="Total Steps",ylab="Counts",main="Total Steps Historgra
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



3. Mean and median number of steps taken each day

```
daily <- activity %>% group_by(date) %>% summarise(
        total_steps = sum(steps),
        mean_steps = mean(steps),
        median_steps =median(steps))
daily
```

```
## # A tibble: 61 x 4
##    date       total_steps mean_steps median_steps
##    <date>           <dbl>      <dbl>        <dbl>
##  1 2012-10-01           0      0                0
##  2 2012-10-02         126      0.438            0
##  3 2012-10-03       11352     39.4              0
##  4 2012-10-04       12116     42.1              0
```

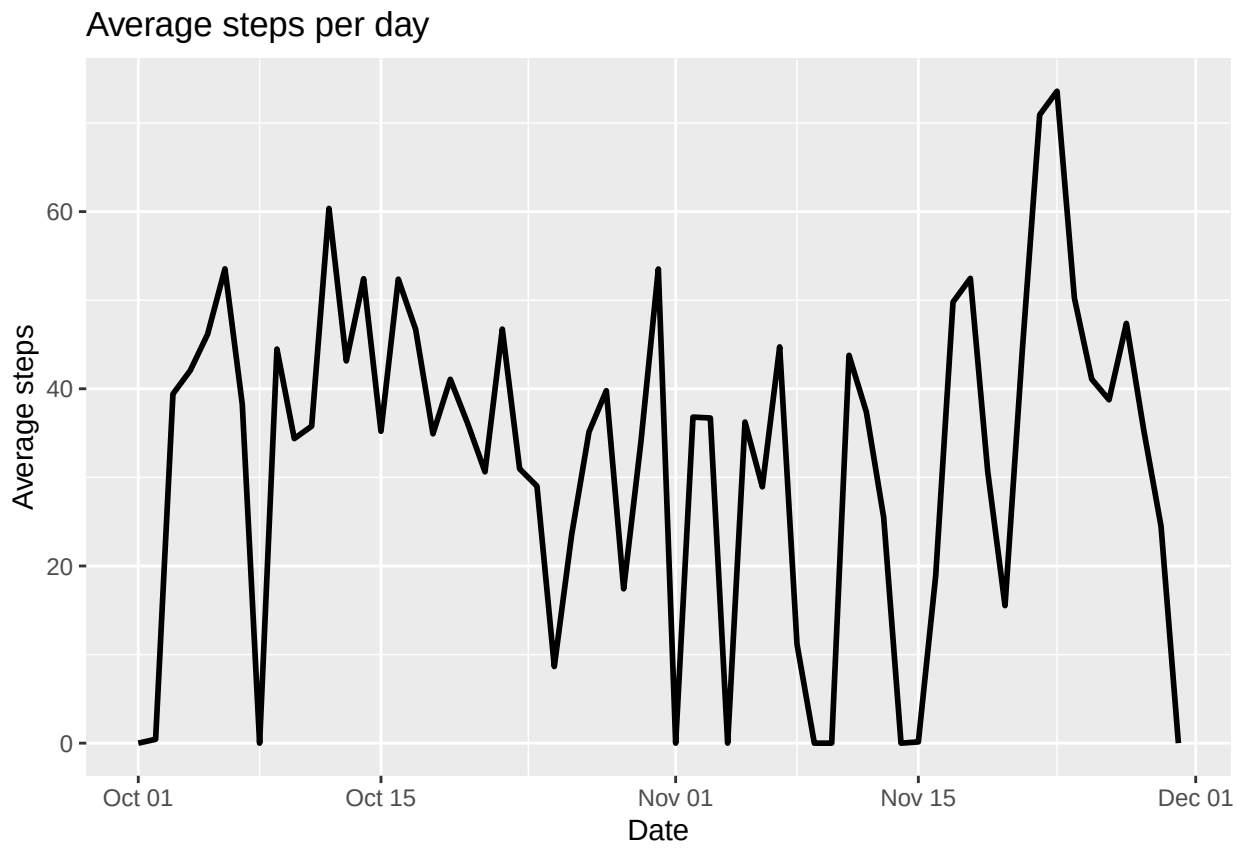
```
##  5 2012-10-05       13294     46.2              0
##  6 2012-10-06       15420     53.5              0
##  7 2012-10-07       11015     38.2              0
##  8 2012-10-08           0      0                0
##  9 2012-10-09       12811     44.5              0
## 10 2012-10-10        9900     34.4              0
## # ... with 51 more rows
```

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

```
plot <- ggplot(daily,aes(x = daily$'date',y = daily$mean_steps))+
    geom_line(size=1)+
    scale_x_date()+
    ylab("Average steps")+
    xlab("Date")+
    ggtitle("Average steps per day")
plot
```

```
## Warning: Use of ‘daily$date‘ is discouraged. Use ‘date‘ instead.
```

```
## Warning: Use of ‘daily$mean_steps‘ is discouraged. Use ‘mean_steps‘ instead.
```



5. The 5-minute interval that, on average, contains the maximum number of steps

```
# get the mean per interval
intervals <-aggregate(data = activity,steps~interval, FUN="mean")
# sort descending and return the first row
intervals[order(-intervals$steps),][1,]
```

```
##     interval    steps
## 104      835 179.1311
```

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

```
# this was done above however, there are several rows where
# steps are na, it can be assumed there is 0 data here. since it is not a # large portion of the data s

#load data and count na's
temp<-read.csv("activity.csv")
sum(is.na(temp$steps)) #2304
```

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

```
#percent of dataframe
sum(is.na(temp$steps))/dim(temp)[1] # 13%
```

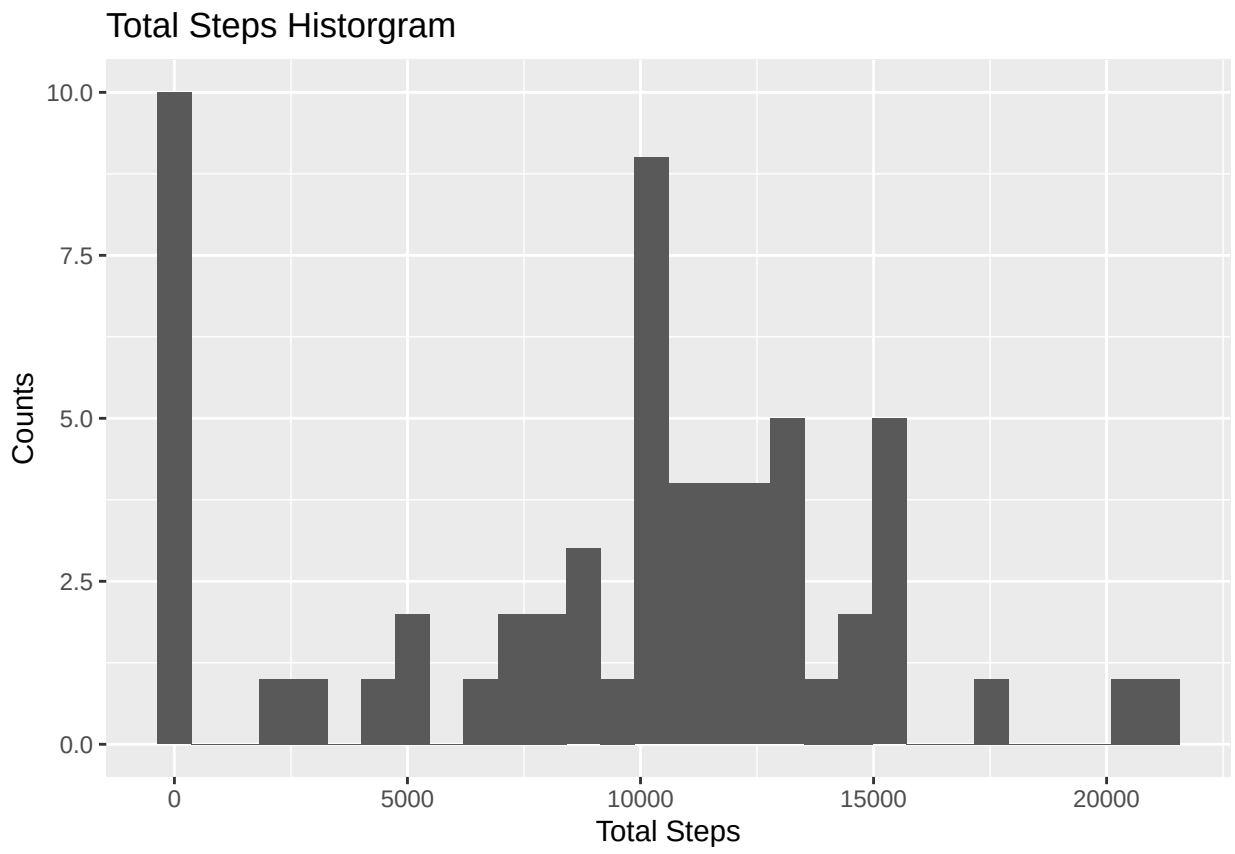
```
## [1] 0.1311475
```

```
#fill na with 0
temp[is.na(temp)] <-0
```

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

```
dailySum <-data.frame(tapply(activity$steps,activity$date,sum))
#change column name
names(dailySum)[[1]]<-"TotalSteps"
qplot(dailySum$TotalSteps,geom="histogram",xlab="Total Steps",ylab="Counts",main="Total Steps Historgra
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



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

```
# add label for weekday/weekend
activity$weekday <- ifelse(!wday(activity$date) %in% c(1,7),                activity$weekday <- "weekda
    activity$weekday <- "weekend")
#group_by interval/weekday and get mean steps
intervals_mod <- activity %>% group_by(interval,weekday) %>% summarise(
     mean_steps = mean(steps))
```

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

```
#make plot
qplot(mean_steps, interval, data = intervals_mod)+
  facet_wrap(~weekday)+
  ggtitle("Average steps per interval: weekday vs weekend")
```


Average steps per interval: weekday vs weekend
weekday
weekend
interval
2000
1500
1000
500
0
0
50
100
150
200
mean_steps