Reproducible Research - Week 2 Assignment

AGF

2022-06-06

Question/Task 1

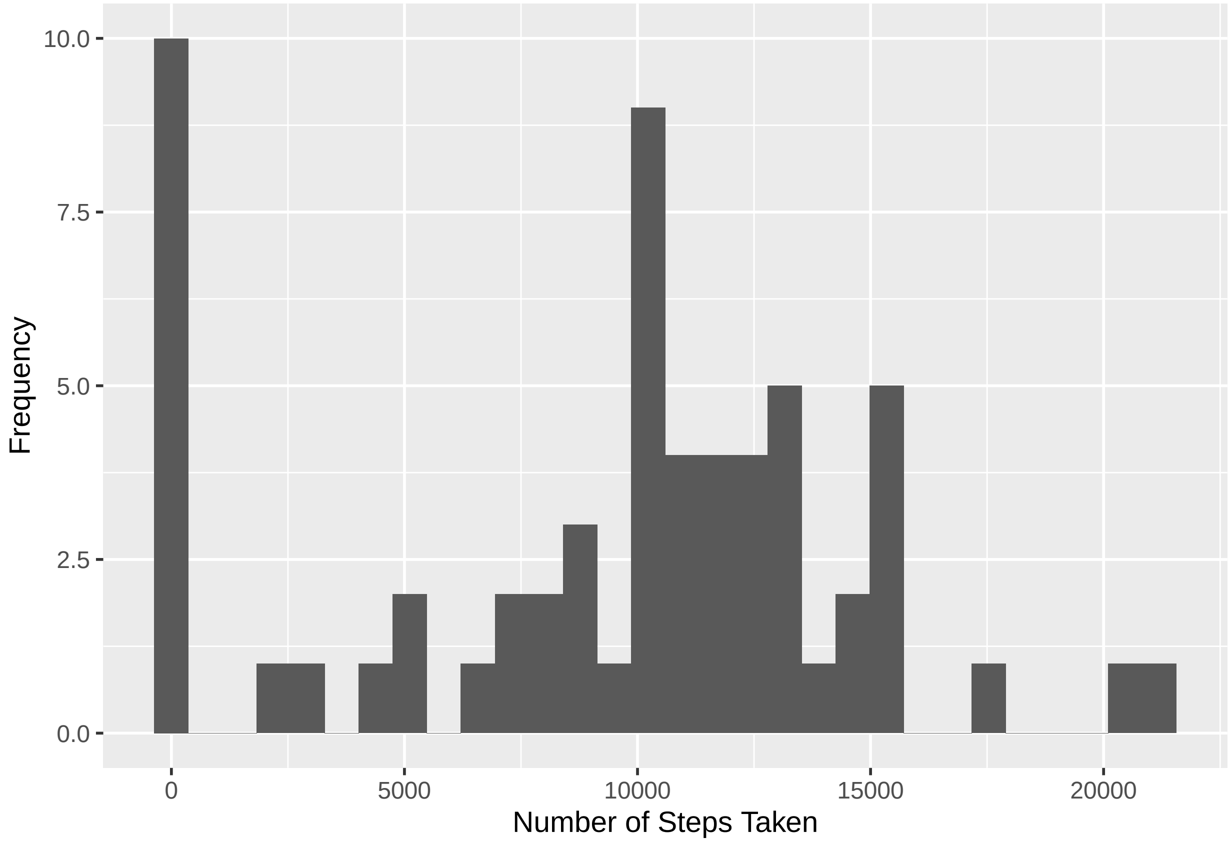
Read in data

```
# DS stands for "dataset"

dsRaw <- read.csv(unz('repdata_data_activity.zip','activity.csv'), header = T)
```

Question/Task 2

Create histogram of total number of steps taken each day

```
dsAggByDay <-aggregate(dsRaw$steps, by=list(dsRaw$date), FUN=sum, na.rm=TRUE)

library(ggplot2)

p <- (ggplot(dsAggByDay, aes(x))
      + geom_histogram(bins=30)
      + xlab("Number of Steps Taken")
      + ylab("Frequency"))

print(p)
```



Question/Task 3

Calculate the mean and median number of steps taken per day

```
medianSteps <- median(dsAggByDay$x, na.rm=FALSE)
meanSteps <- mean(dsAggByDay$x, na.rm=FALSE)
```

The median number of steps taken per day was 10,395 and the mean was 9,354.23.

Question/Task 4

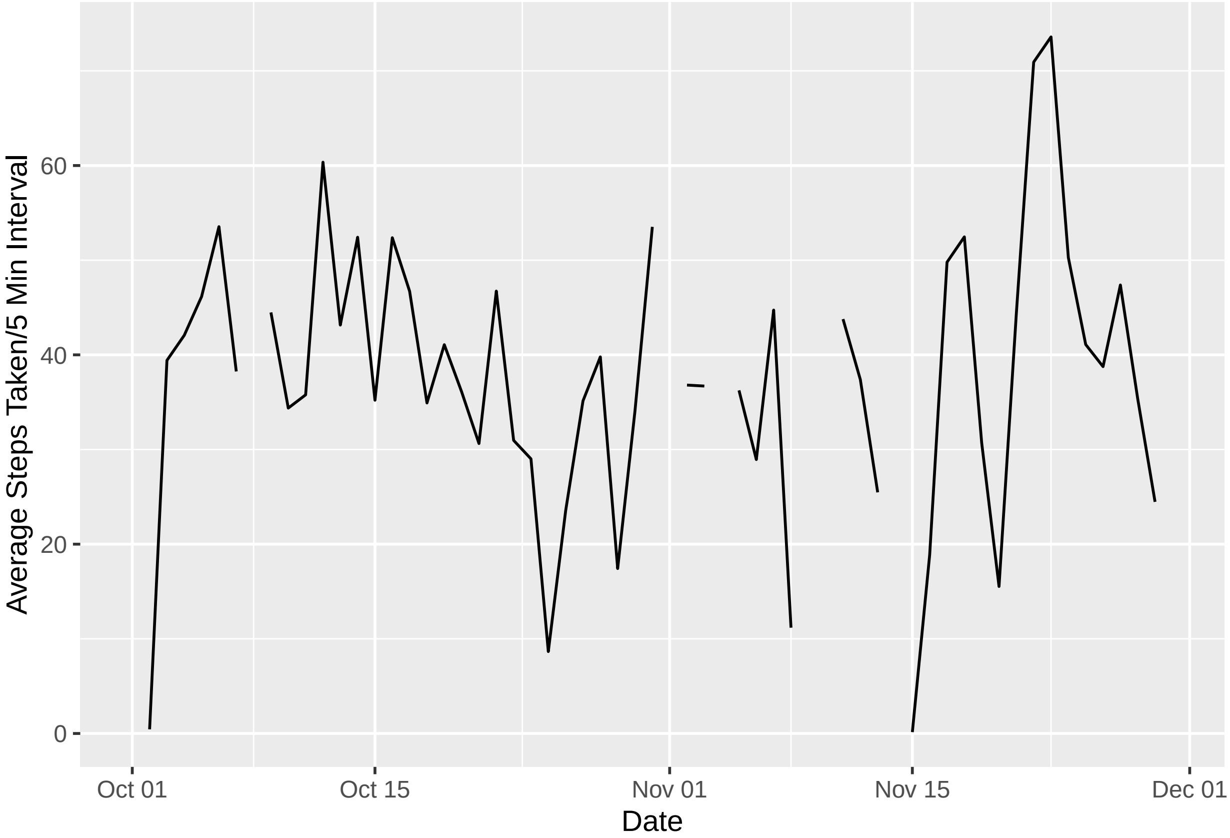
Next, create a time series plot average number of steps per day

```
dsAvgByDay <-aggregate(dsRaw$steps, by=list(dsRaw$date), FUN=mean, na.rm=TRUE)

dsAvgByDay$date <- as.Date(dsAvgByDay$Group.1, format="%Y-%m-%d")

class(dsAvgByDay$Group.1)
```

```
## [1] "character"
```

```
p <- (ggplot(dsAvgByDay, aes(x=date, y=x))
      + geom_line()
      + xlab("Date")
      + ylab("Average Steps Taken/5 Min Interval"))

print(p)
```

```
## Warning: Removed 2 row(s) containing missing values (geom_path).
```



Question/Task 5

Next, calculate the 5-minute interval that on average contains the most steps

```
dsAvgByInterval <-aggregate(dsRaw$steps, by=list(dsRaw$interval), FUN=mean, na.rm=TRUE)

maxInterval <- which.max(dsAvgByInterval$x)
```

The maximum average number of steps occurred at interval 835 and the average number of steps taken at interval 835 was 206.1698

Question/Task 6

There are several major ways to impute missing data, each with its benefits and drawbacks.

1. Delete records with missing observations
2. Impute the average of the non-missing values of a variable for records with missing values
3. Impute missing values stochastically in a way that mimics the distribution of the non-missing values
4. Build a model predict what the missing values are, often as a function of other variables in the dataset

Regarding (1), this is fine if the observation are Missing at Random (MAR)[1], which is seldom the case. Deleting missing values can bias the data if the missing values aren't missing at random.

Regarding (2), this method usually results in a dataset that understates the variation in the true data. (Practitioners can also impute the minimum, maximum, median, etc. value depending on what the analysis is trying to accomplish.)

Regarding (3), this is usually the second-best option when (4) isn't possible. Doing multiple stochastic imputations and then reporting a range of possible outcomes is an even stronger method. The downside to this method is that it is usually programmatically and computationally more involved than (1) and (2), but computational power and off-the-shelf ("canned") functions in statistical software programs make this easier.

Regarding (4), this is usually the preferred method when it's possible. It requires enough data to model the missing values in a way that's better than random guessing. It also requires building a model with all the time and resources that that entails, which isn't always fesible or desirable.

This analysis will use (2), imputing the average value.

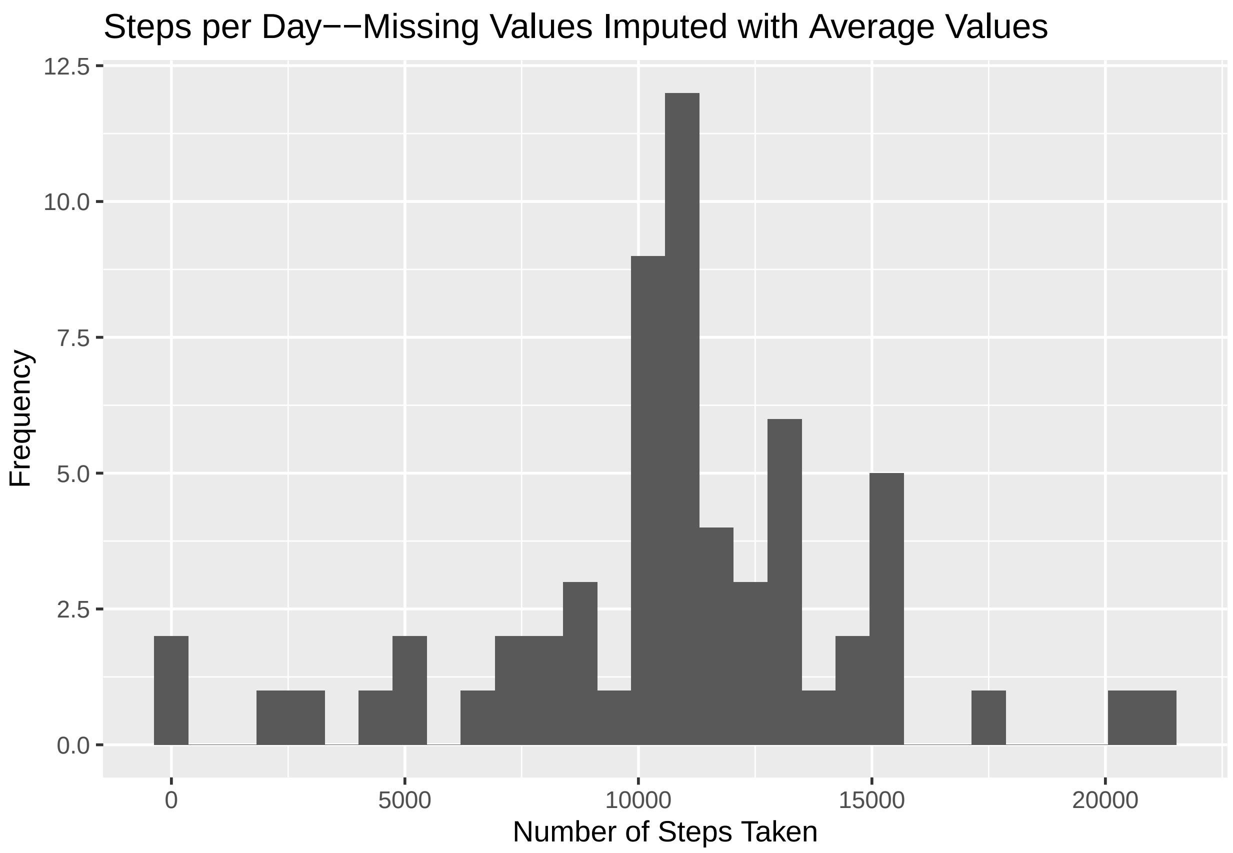
[1] Rubin, Donald B. "Inference and Missing Data." Biometrika 63, no. 3 (1976): 581–92. https://doi.org/10.2307/2335739.

```
dsImputed <- dsRaw

dsImputed$steps[is.na(dsImputed$steps)] <- mean(dsImputed$steps, na.rm = TRUE)

dsAggByDayImputed <-aggregate(dsImputed$steps,
                              by=list(dsImputed$date), FUN=sum, na.rm=TRUE)
```

Question/Task 7

Here is a histogram of the number of steps taken per day after imputation.

```
library(ggplot2)

p <- (ggplot(dsAggByDayImputed, aes(x))
      + geom_histogram(bins=30)
      + ggtitle("Steps per Day--Missing Values Imputed with Average Values")
      + xlab("Number of Steps Taken")
      + ylab("Frequency"))

print(p)
```



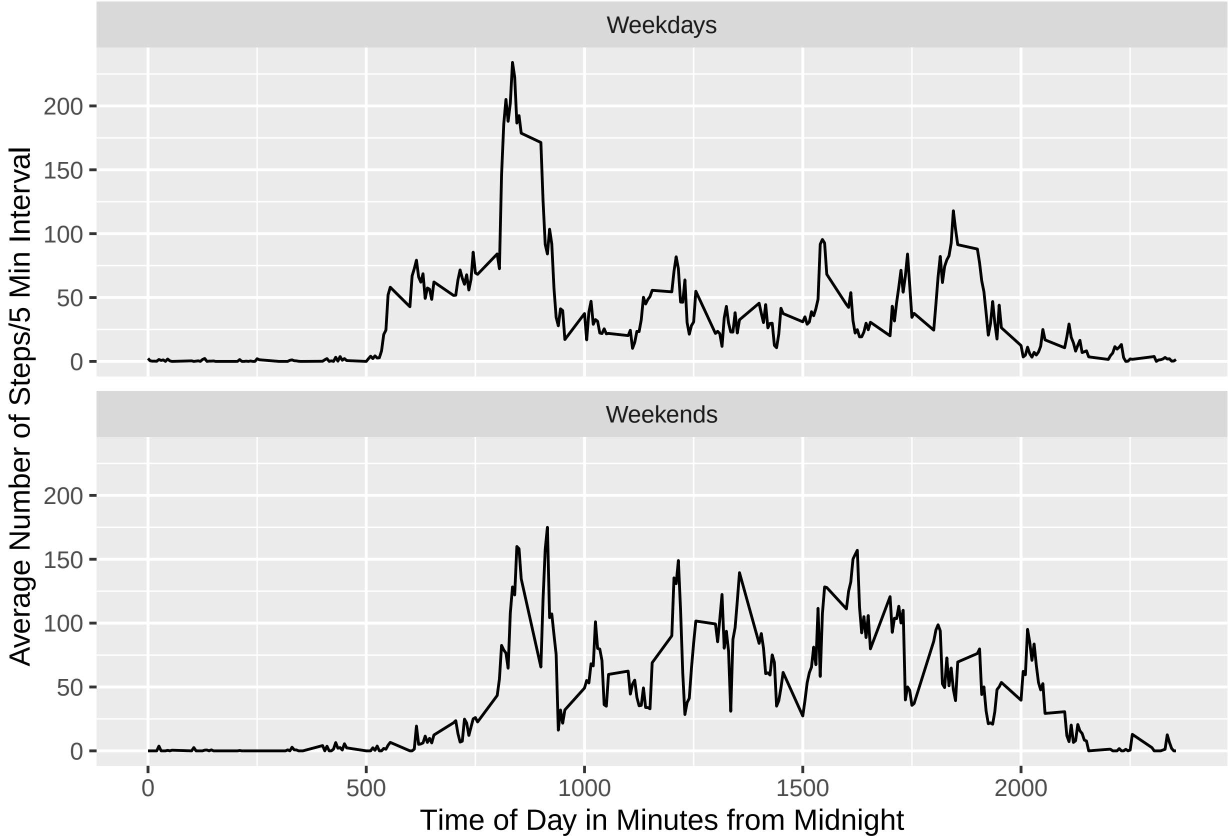
Question/Task 8

Create a panel plot comparing average steps taken per 5-minute interval on weekdays versus weekends

```
#install.packages("timeDate", repos = "http://cran.us.r-project.org")

library(timeDate)
```

```
## Warning: package 'timeDate' was built under R version 4.1.2
```

```
dsRaw$date <- as.Date(dsRaw$date, format="%Y-%m-%d")

dsRaw$weekend <- isWeekend(dsRaw$date)

dsAvgByDoW <-aggregate(dsRaw$steps, by=list(dsRaw$interval,dsRaw$weekend),
                       FUN=mean, na.rm=TRUE)

colnames(dsAvgByDoW) <- c("interval","weekend","avgSteps")

facet_labels <- as_labeller(c('TRUE' = "Weekends", 'FALSE' = "Weekdays"))

p <- (ggplot(dsAvgByDoW, aes(x=interval, y=avgSteps))
     + geom_line()
     + facet_wrap(dsAvgByDoW$weekend, nrow=2, labeller=facet_labels)
     + ylab("Average Number of Steps/5 Min Interval")
     + xlab("Time of Day in Minutes from Midnight"))

print(p)
```


Weekdays
Weekends
Average Number of Steps/5 Min Interval
Time of Day in Minutes from Midnight