Reproducible Research Project 1

Brody Vogel

2/17/2019

Loading and Preprocessing the Data

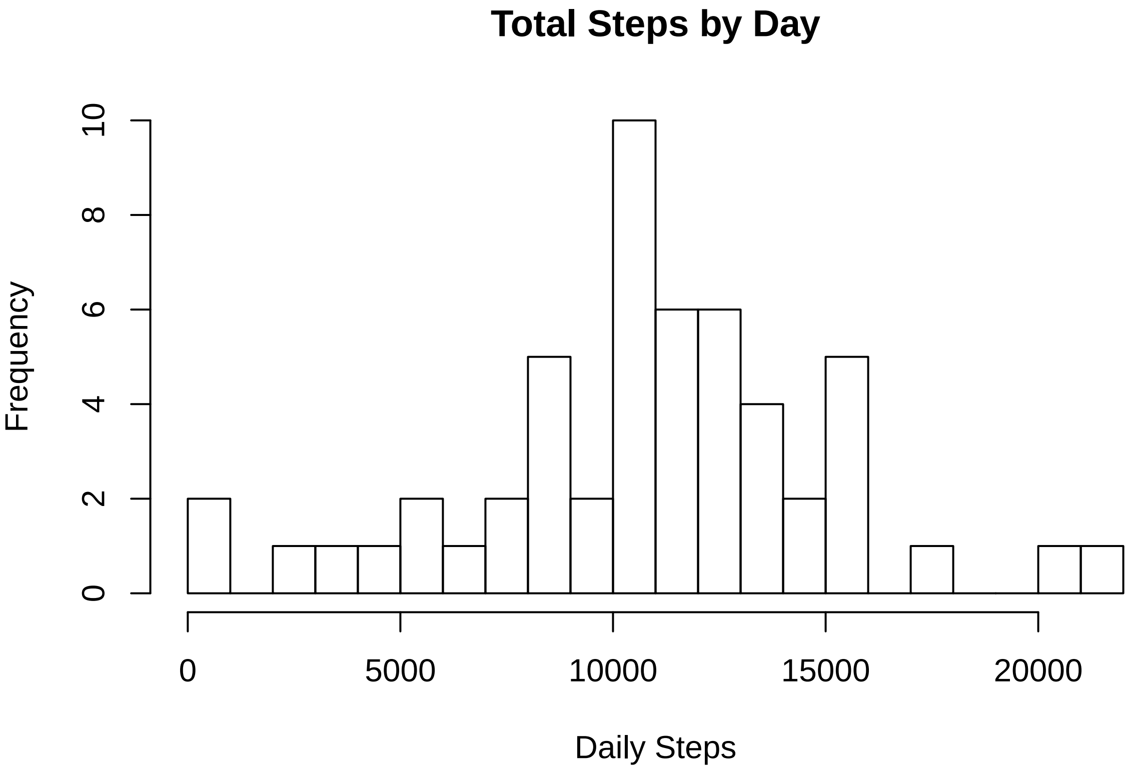
```
suppressMessages(library(tidyverse))
```

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

```
activity <- read.csv('/Users/brodyvogel/Desktop/activity.csv')

# there are a lot of NAs
no_nas <- activity[complete.cases(activity), ]
```

What is the Mean Total Number of Steps Taken Per Day?

```
# aggregate the steps by date
sums <- aggregate(activity$steps ~ activity$date, FUN = sum)
# make the histogram
hist(sums$`activity$steps`, breaks = 25, xlab = 'Daily Steps', main = 'Total Steps by Day')
```



The mean steps per day was 10766. The median number of steps per day was one less, 10765.

```
# calculate the mean and median
mean(sums[, 2])
```

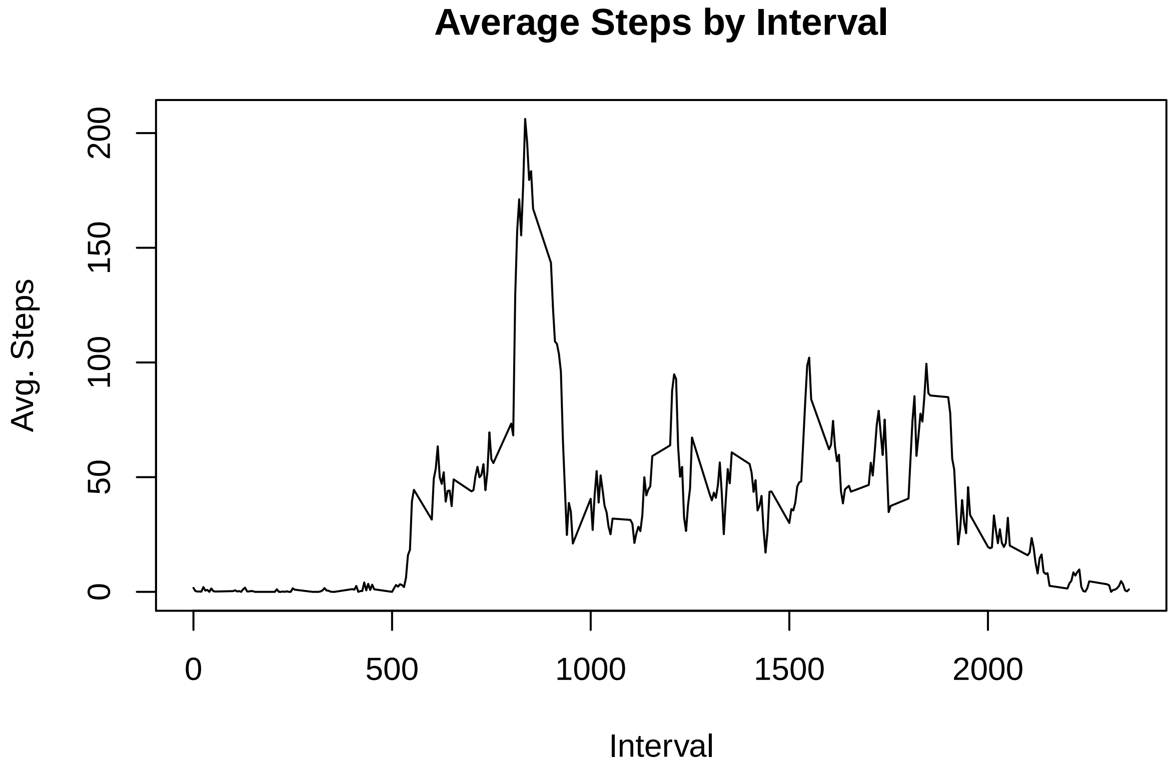
```
## [1] 10766.19
```

```
median(sums[, 2])
```

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

What is the Average Daily Activity Pattern?

```
# aggregate the steps by interval and mean
avg_by_interval <- aggregate(activity$steps ~ activity$interval, FUN = mean)
# plot the time series
plot(avg_by_interval[, 1], avg_by_interval[, 2], type = 'l',
     xlab = 'Interval', ylab = 'Avg. Steps', main = 'Average Steps by Interval')
```



The largest number of average steps in a 5-minute interval was 206. The corresponding 5-minute interval was the 835 entry.

```
# calculate the max average steps and the corresponding interval
max_steps <- max(avg_by_interval[, 2])
max_steps
```

```
## [1] 206.1698
```

```
avg_by_interval[avg_by_interval[, 2] == max_steps, 1]
```

```
## [1] 835
```

Imputing Missing Values

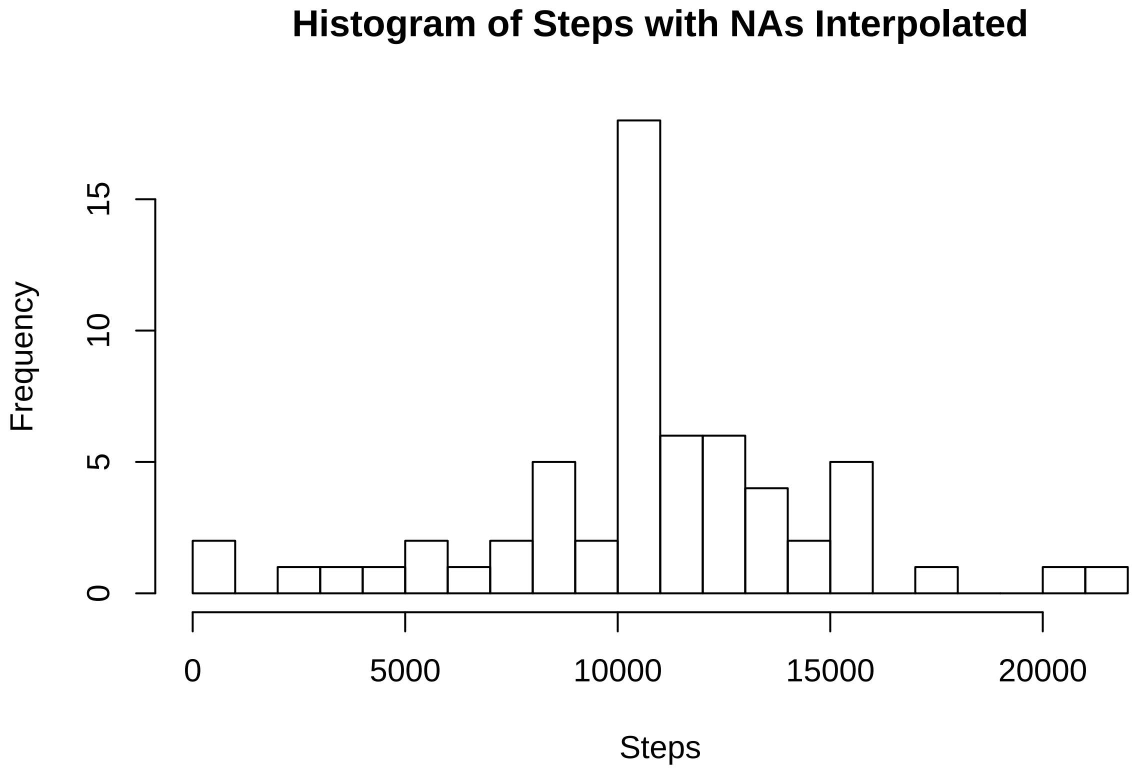
The number of rows containing an NA is 2304.

```
# get the number of rows with an NA in any column
sum(is.na(activity$steps)) + sum(is.na(activity$date)) + sum(is.na(activity$interval))
```

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

My strategy is to fill in the missing values for steps with the average number of steps for that invterval from the rest of the data set.

```
# get just the NA entries
all_nas <- activity[is.na(activity$steps), ]
# throw out the steps observation
all_nas <- all_nas %>% select(c(date, interval))

# so the merge works
names(avg_by_interval) <- c('interval', 'steps')

# create the interpolated data
interpolated_nas <- merge(all_nas, avg_by_interval, by = 'interval')

# merge the interpolated data with the clean data from before
interpolated_data <- rbind(no_nas, interpolated_nas)

# aggregate by date again
agg_interpolated_data <- aggregate(interpolated_data$steps ~ interpolated_data$date, FUN = sum)

hist(agg_interpolated_data[, 2], xlab = 'Steps', breaks = 25, main = 'Histogram of Steps with NAs Interp
```



The mean stays exactly the same, 10,766; with all the interpolated data, now, though, the median goes up one to 10,766 too.

So the data changes a little when observations are interpolated. The distribution is unchanged, however. The effect, then, is that interpolating the missing observations brings more uniformity to the data; whether this is justified or not would depend on the situation. I think it's fine, here.

```r
# calculate the mean and median
mean(agg_interpolated_data[, 2])
```

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

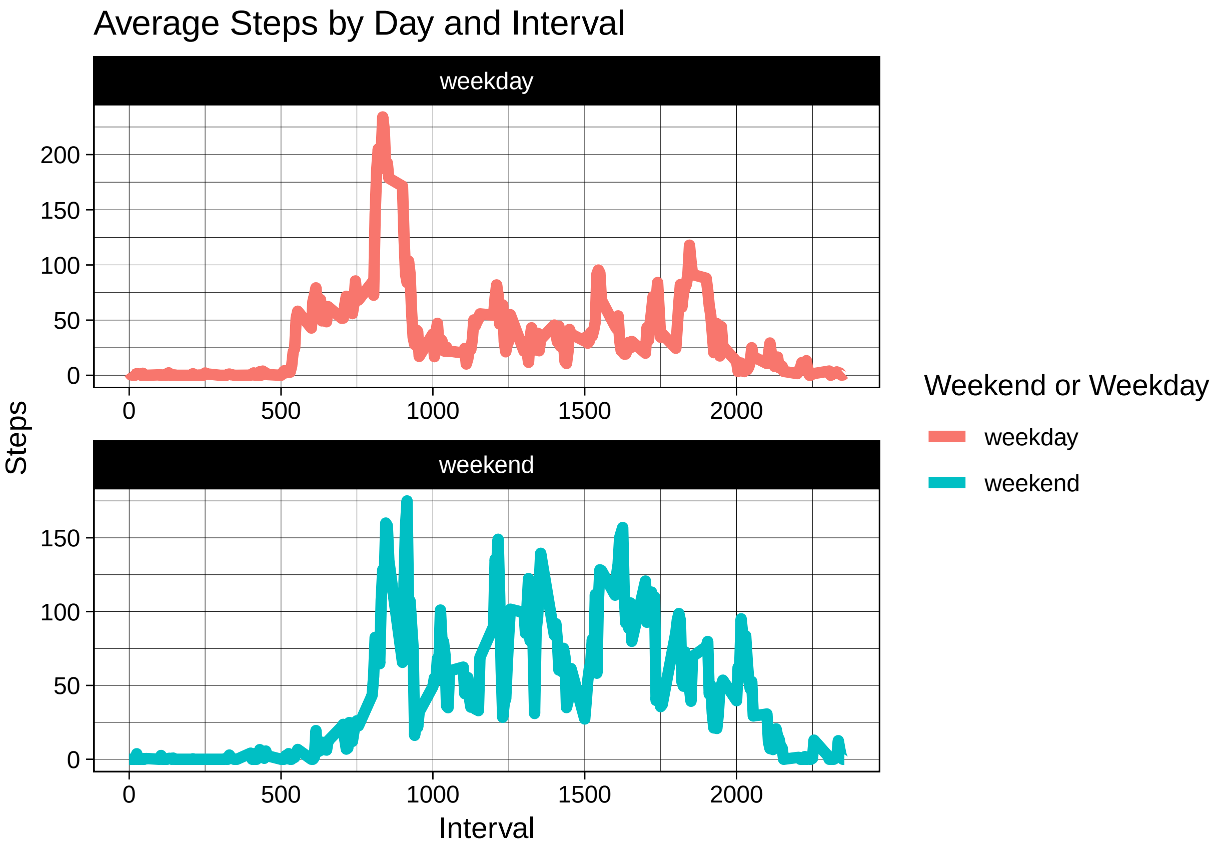
```r
median(agg_interpolated_data[, 2])
```

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

Are there Differences in Activity Patterns Between Weekdays and Weekends?

The trend lines are quite different depending on whether the activity happends on a weekend or weekday. It looks like activity is more spread out on the weekends, potentially because people are not working in a confined space.

```r
# calculate the factor variable
no_nas$day <- weekdays(as.Date(no_nas$date, '%Y-%m-%d'))
no_nas$weekend_or_not <- ifelse(no_nas$day %in% c('Saturday', 'Sunday'), 'weekend', 'weekday')

# create a grouped data frame with averages
to_plot <- no_nas %>% group_by(weekend_or_not, interval) %>%
  summarize(steps = mean(steps))

# plot it
ggplot(to_plot, aes(interval, steps, color = weekend_or_not)) +
  theme_linedraw() +
  geom_line(size = 2) +
  labs(title = 'Average Steps by Day and Interval', x = 'Interval', y = 'Steps', color = "Weekend or Wee
  facet_grid(~weekend_or_not) +
  facet_wrap(~weekend_or_not, scales = 'free', nrow = 2)
```


Average Steps by Day and Interval
weekday
200
150
100
50
0
0
500
1000
1500
2000
Steps
weekend
150
100
50
0
0
500
1000
1500
2000
Interval
Weekend or Weekday
weekday
weekend