PA1_Template_SK

Stu Kyle

August 13, 2017

first load the packages needed for the analysis

```
library(ggplot2)
library(lubridate)
```

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

## The following object is masked from 'package:base':
##
##     date
```

```
library(plyr)
```

```
##
## Attaching package: 'plyr'

## The following object is masked from 'package:lubridate':
##
##     here
```

Loading and preprocessing the data

Part 1 Import data

```
activity_data <- read.csv("activity.csv", stringsAsFactors = FALSE)
```

make the date field a date variable

```
activity_data$date <- ymd(activity_data$date)
```

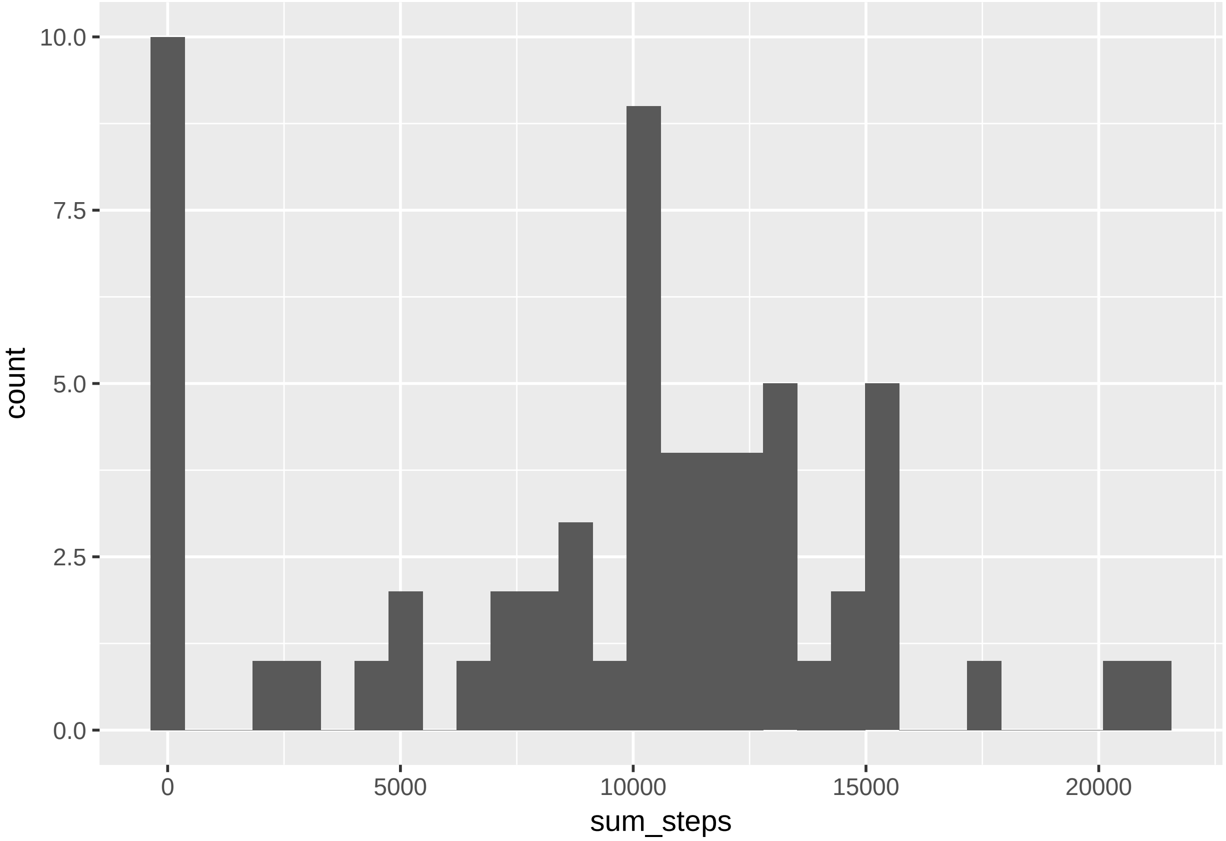
What is mean total number of steps taken per day?

Part 2 Histogram of total steps per day

```
ad_sum <- aggregate(list(sum_steps = activity_data[,1]),
list(date = activity_data$date),
sum,
na.rm = TRUE)
```

```r
plot_steps_per_day <- ggplot(data = ad_sum, aes(sum_steps)) + geom_histogram()
print(plot_steps_per_day)
```

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



Part 3 Mean and Median number of steps each day

```r
aggregate(list(mean_steps = activity_data[,1]),
list(date = activity_data$date),
mean,
na.rm = TRUE)
```

```
##          date mean_steps
## 1  2012-10-01        NaN
## 2  2012-10-02  0.4375000
## 3  2012-10-03 39.4166667
## 4  2012-10-04 42.0694444
## 5  2012-10-05 46.1597222
## 6  2012-10-06 53.5416667
## 7  2012-10-07 38.2465278
## 8  2012-10-08        NaN
## 9  2012-10-09 44.4826389
## 10 2012-10-10 34.3750000
## 11 2012-10-11 35.7777778
## 12 2012-10-12 60.3541667
## 13 2012-10-13 43.1458333
## 14 2012-10-14 52.4236111
## 15 2012-10-15 35.2048611
## 16 2012-10-16 52.3750000
## 17 2012-10-17 46.7083333
## 18 2012-10-18 34.9166667
## 19 2012-10-19 41.0729167
## 20 2012-10-20 36.0937500
## 21 2012-10-21 30.6284722
## 22 2012-10-22 46.7361111
## 23 2012-10-23 30.9652778
## 24 2012-10-24 29.0104167
## 25 2012-10-25  8.6527778
## 26 2012-10-26 23.5347222
## 27 2012-10-27 35.1354167
## 28 2012-10-28 39.7847222
## 29 2012-10-29 17.4236111
## 30 2012-10-30 34.0937500
## 31 2012-10-31 53.5208333
## 32 2012-11-01        NaN
## 33 2012-11-02 36.8055556
## 34 2012-11-03 36.7048611
## 35 2012-11-04        NaN
## 36 2012-11-05 36.2465278
## 37 2012-11-06 28.9375000
## 38 2012-11-07 44.7326389
## 39 2012-11-08 11.1770833
## 40 2012-11-09        NaN
## 41 2012-11-10        NaN
## 42 2012-11-11 43.7777778
## 43 2012-11-12 37.3784722
## 44 2012-11-13 25.4722222
## 45 2012-11-14        NaN
## 46 2012-11-15  0.1423611
## 47 2012-11-16 18.8923611
## 48 2012-11-17 49.7881944
## 49 2012-11-18 52.4652778
## 50 2012-11-19 30.6979167
## 51 2012-11-20 15.5277778
## 52 2012-11-21 44.3993056
## 53 2012-11-22 70.9270833
## 54 2012-11-23 73.5902778
## 55 2012-11-24 50.2708333
## 56 2012-11-25 41.0902778
## 57 2012-11-26 38.7569444
## 58 2012-11-27 47.3819444
## 59 2012-11-28 35.3576389
## 60 2012-11-29 24.4687500
## 61 2012-11-30        NaN
```

```
aggregate(list(median_steps = activity_data[,1]),
list(date = activity_data$date),
median,
na.rm = TRUE)
```

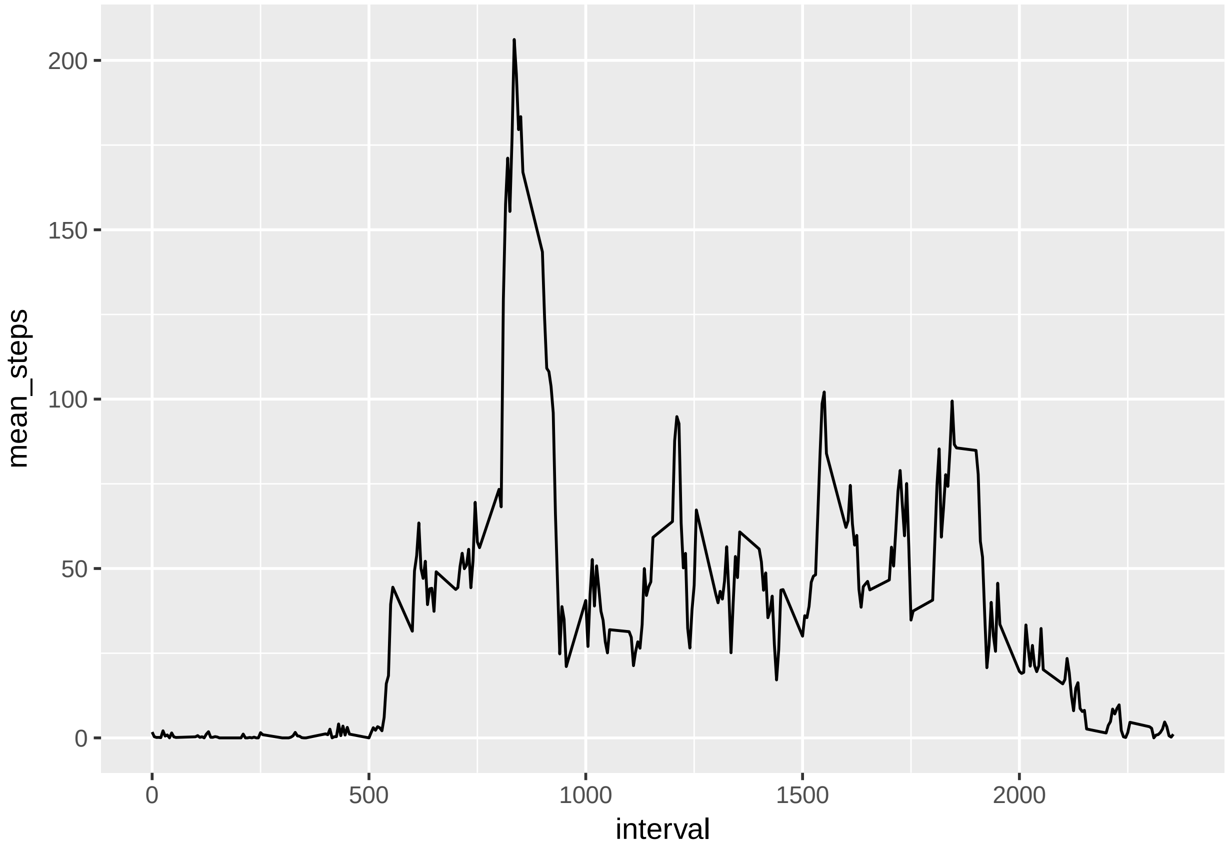
```
##          date median_steps
## 1  2012-10-01           NA
## 2  2012-10-02            0
## 3  2012-10-03            0
## 4  2012-10-04            0
## 5  2012-10-05            0
## 6  2012-10-06            0
## 7  2012-10-07            0
## 8  2012-10-08           NA
## 9  2012-10-09            0
## 10 2012-10-10            0
## 11 2012-10-11            0
## 12 2012-10-12            0
## 13 2012-10-13            0
## 14 2012-10-14            0
## 15 2012-10-15            0
## 16 2012-10-16            0
## 17 2012-10-17            0
## 18 2012-10-18            0
## 19 2012-10-19            0
## 20 2012-10-20            0
## 21 2012-10-21            0
## 22 2012-10-22            0
## 23 2012-10-23            0
## 24 2012-10-24            0
## 25 2012-10-25            0
## 26 2012-10-26            0
## 27 2012-10-27            0
## 28 2012-10-28            0
## 29 2012-10-29            0
## 30 2012-10-30            0
## 31 2012-10-31            0
## 32 2012-11-01           NA
## 33 2012-11-02            0
## 34 2012-11-03            0
## 35 2012-11-04           NA
## 36 2012-11-05            0
## 37 2012-11-06            0
## 38 2012-11-07            0
## 39 2012-11-08            0
## 40 2012-11-09           NA
## 41 2012-11-10           NA
## 42 2012-11-11            0
## 43 2012-11-12            0
## 44 2012-11-13            0
## 45 2012-11-14           NA
## 46 2012-11-15            0
## 47 2012-11-16            0
## 48 2012-11-17            0
## 49 2012-11-18            0
## 50 2012-11-19            0
## 51 2012-11-20            0
## 52 2012-11-21           0
## 53 2012-11-22           0
## 54 2012-11-23           0
## 55 2012-11-24           0
## 56 2012-11-25           0
## 57 2012-11-26           0
## 58 2012-11-27           0
## 59 2012-11-28           0
## 60 2012-11-29           0
## 61 2012-11-30          NA
```

What is the average daily activity pattern?

Part 4 Time Series Plot

```
ad_interval_mean <- aggregate(list(mean_steps = activity_data[,1]),
list(interval = activity_data$interval),
mean,
na.rm = TRUE)

plot_mean_steps_per_interval <- ggplot(data = ad_interval_mean,
aes(x = interval, y = mean_steps)) + geom_line()
print(plot_mean_steps_per_interval)
```



Part 5 Interval with most steps

```
ad_interval_mean[which.max(ad_interval_mean$mean_steps),]
```

```
##     interval mean_steps
## 104      835   206.1698
```

Imputing missing values

Part 6 Impute Values

I will set NA values equal to the average for that interval

find how many rows with missing values

```
sum(is.na(activity_data$steps))
```

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

create a data frame with rows that have missing steps

```
activity_data_missing <- subset(activity_data, is.na(steps))[,2:3]
```

set the missing steps equal to the average

```
ad_missing_imputed_values <- join(activity_data_missing,ad_interval_mean)
```

```
## Joining by: interval
```

change the column name back to steps

```
names(ad_missing_imputed_values)[3] <- c("steps")
```

put the two back together

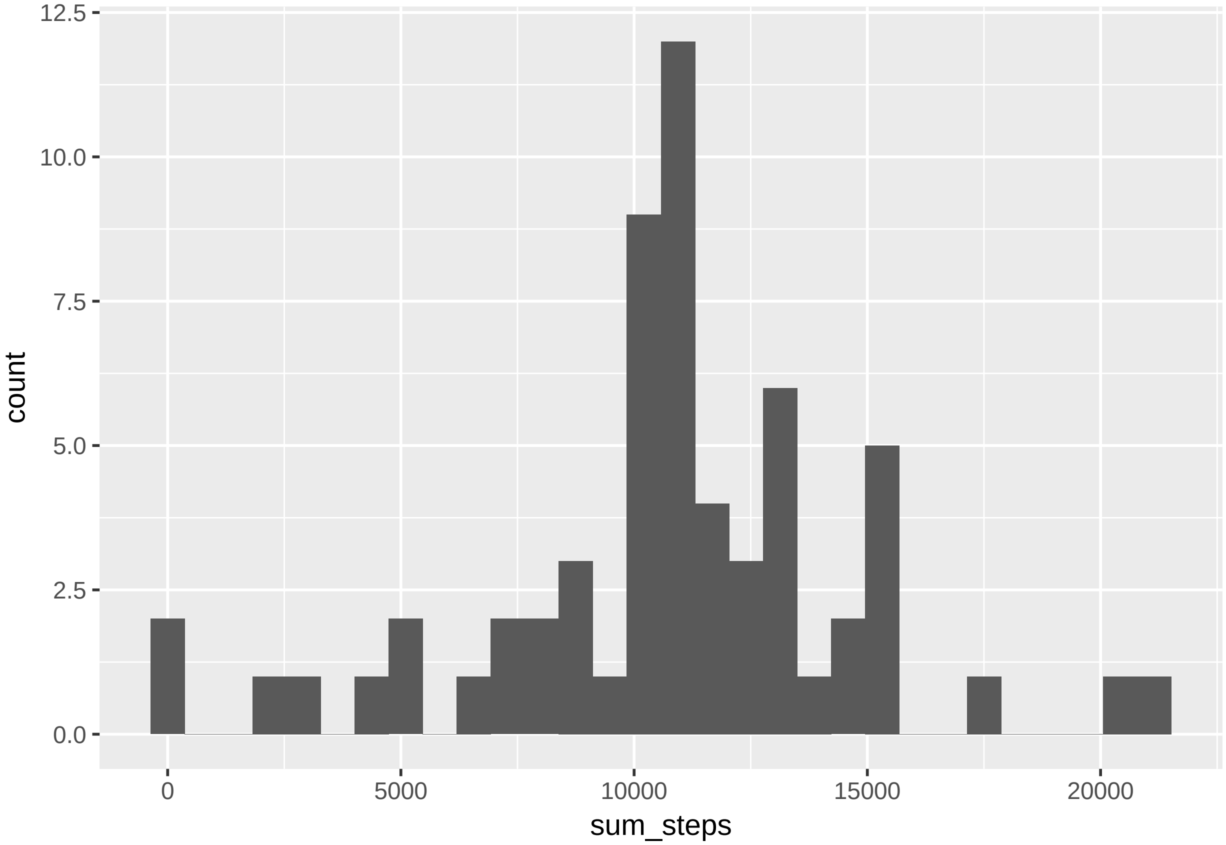
```
activity_data_impute <- rbind.fill(subset(activity_data, !is.na(steps)),
ad_missing_imputed_values)
```

Part 7 Histogram of total steps per day with imputed data

```
ad_sum_impute <- aggregate(list(sum_steps = activity_data_impute[,1]),
list(date = activity_data_impute$date),
sum)

plot_steps_per_day_impute <- ggplot(data = ad_sum_impute,
aes(sum_steps)) + geom_histogram()
print(plot_steps_per_day_impute)
```

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



Are there differences in activity patterns between weekdays and weekends?

Get average and standard deviation of steps by day of the week.

```
dow_mean <- aggregate(list(mean_steps = ad_sum_impute[,2]),
list(day_of_week = wday(ad_sum_impute$date, label = TRUE)),
mean)

dow_sd <- aggregate(list(sd_steps = ad_sum_impute[,2]),
list(day_of_week = wday(ad_sum_impute$date, label = TRUE)),
sd)
```

```
dow_mean
```

```
##   day_of_week mean_steps
## 1         Sun  12088.774
## 2         Mon  10150.709
## 3        Tues   8949.556
## 4         Wed  11676.910
## 5       Thurs   8496.465
## 6         Fri  12005.597
## 7         Sat  12314.274
```

```
dow_sd
```

```
##   day_of_week sd_steps
## 1         Sun 2154.603
## 2         Mon 2270.804
## 3        Tues 4693.636
## 4         Wed 2155.313
## 5       Thurs 6162.292
## 6         Fri 4879.720
## 7         Sat 2150.670
```

Based on the mean and standard deviation, people take slightly more steps on the weekends and the number of steps is very consistent. On weekdays, the mean number of steps can be quite different from day to day and for each weekday, the variance can be quite large. The variances are particularly high on Tuesday, Thursday, and Friday.