Activity Data

Activity Monitoring

Large amount of data are collected through fitness devices, such as fitBit, Nike Fuelband, and Jawbone Up. The data collected here is from october and November of 2012, and was collected at 5 minute intervals. It aims to measure the number of steps taken in each 5 minute interval. This report will utilize this data to create summary measures to determine when activity levels are higher/lower.

This code create summary measures of the data; specifically the mean, median, and total steps by each day.

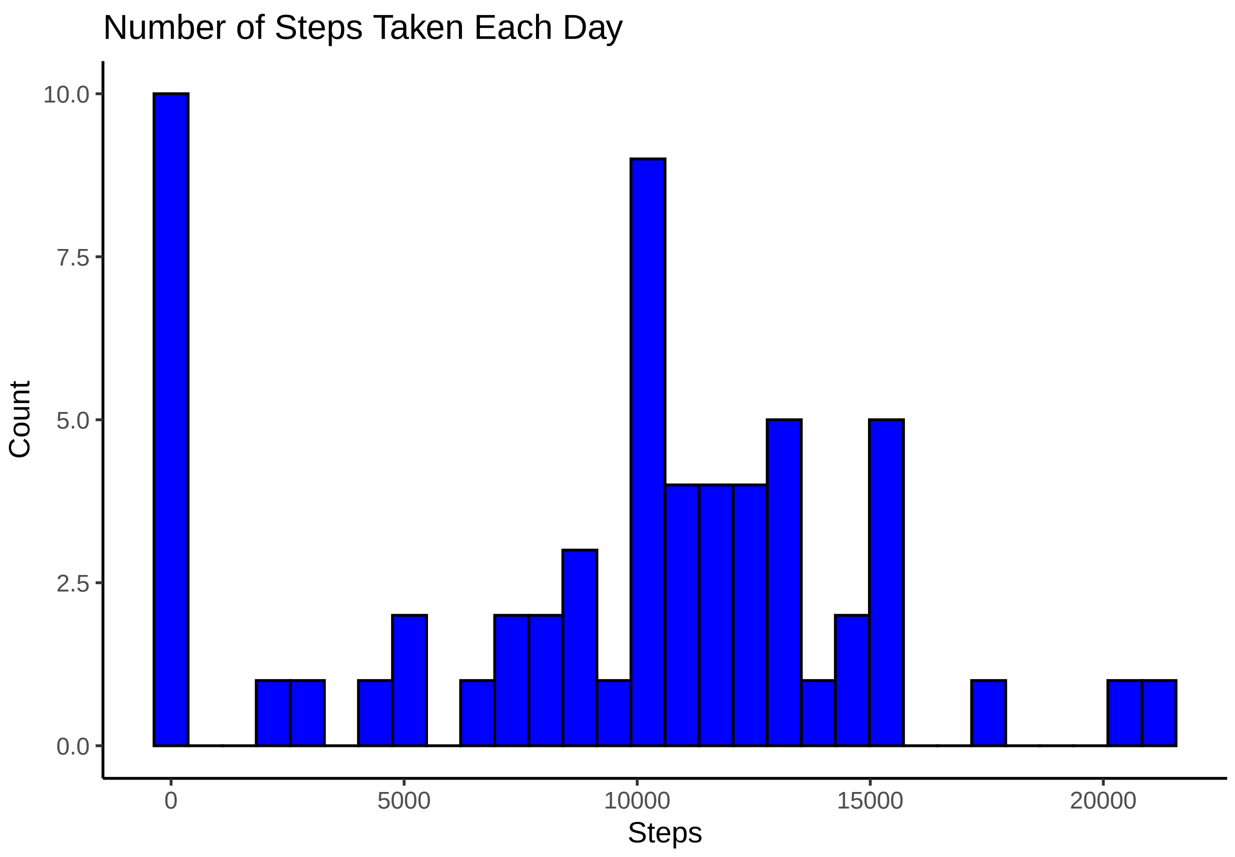
```
df<- activity %>% group_by(date) %>% summarise(mean.steps = mean(steps,na.rm = TRUE),
                                               median.steps = median(steps,na.rm = TRUE),
                                               total.steps = sum(steps,na.rm = TRUE)

)
df
```

```
## # A tibble: 61 x 4
##    date       mean.steps median.steps total.steps
##    <fct>           <dbl>        <dbl>       <int>
##  1 2012-10-01    NaN               NA           0
##  2 2012-10-02      0.438            0         126
##  3 2012-10-03     39.4              0       11352
##  4 2012-10-04     42.1              0       12116
##  5 2012-10-05     46.2              0       13294
##  6 2012-10-06     53.5              0       15420
##  7 2012-10-07     38.2              0       11015
##  8 2012-10-08    NaN               NA           0
##  9 2012-10-09     44.5              0       12811
## 10 2012-10-10     34.4              0        9900
## # ... with 51 more rows
```

Create a histogram of the total steps taken each day and a time series plot of mean steps taken each day.

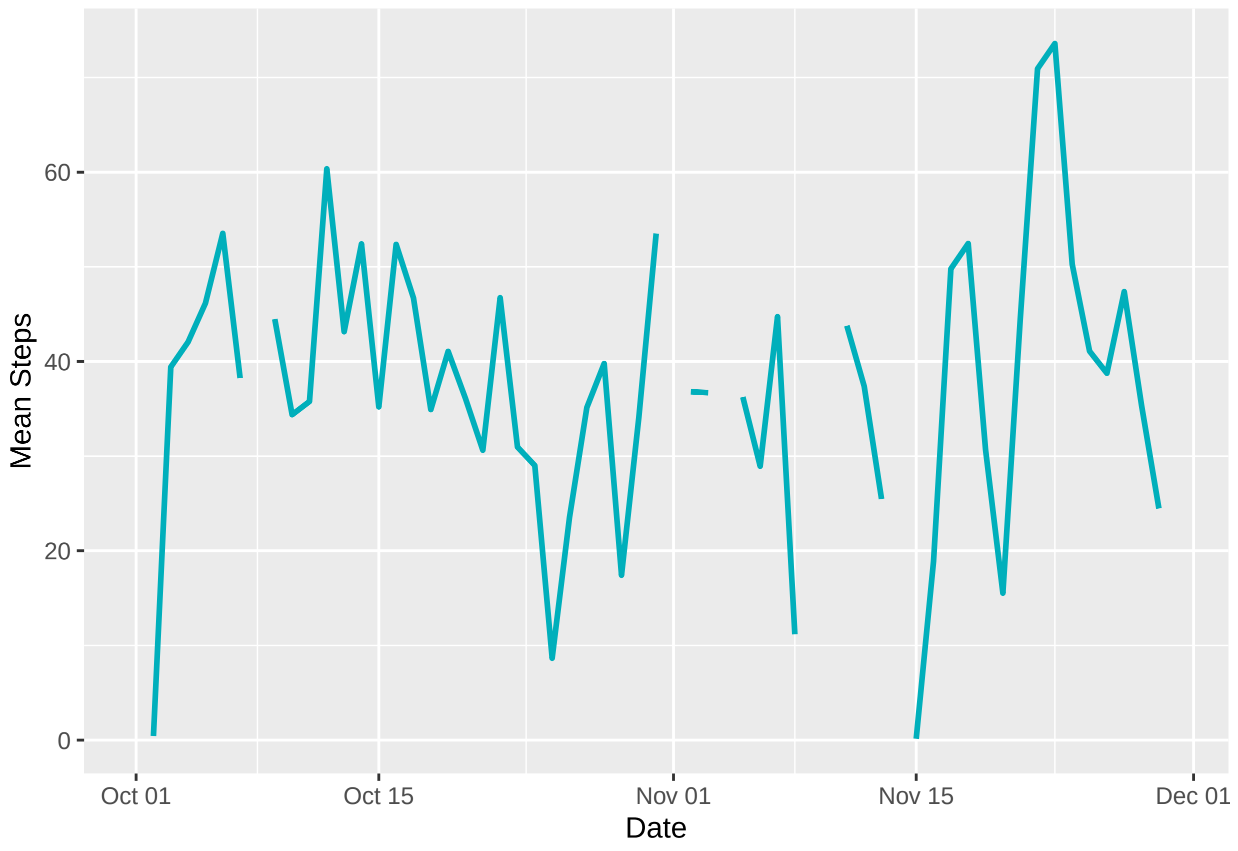
```
#Create Histogram of # steps taken
ggplot(df, aes(x=total.steps)) + geom_histogram(fill = "blue", color = "black") + labs(x = "Steps", y =
  ggtitle("Number of Steps Taken Each Day") + theme_classic()
```

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



```
#Create Plot of mean steps
ggplot(data = df, aes(x = as.Date(date), y = mean.steps))+
  geom_line(color = "#00AFBB", size = 1) + labs(x = "Date", y = "Mean Steps")
```

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



Mean Steps
60
40
20
0
Oct 01
Oct 15
Nov 01
Nov 15
Dec 01
Date

```r
# Get the interval with the hgihest number of steps
df.interval <- activity %>% group_by(interval) %>% summarise(mean.steps = mean(steps,na.rm = TRUE))
df.interval[which.max(df.interval$mean.steps),1]
```

```
## # A tibble: 1 x 1
##   interval
##      <int>
## 1      835
```

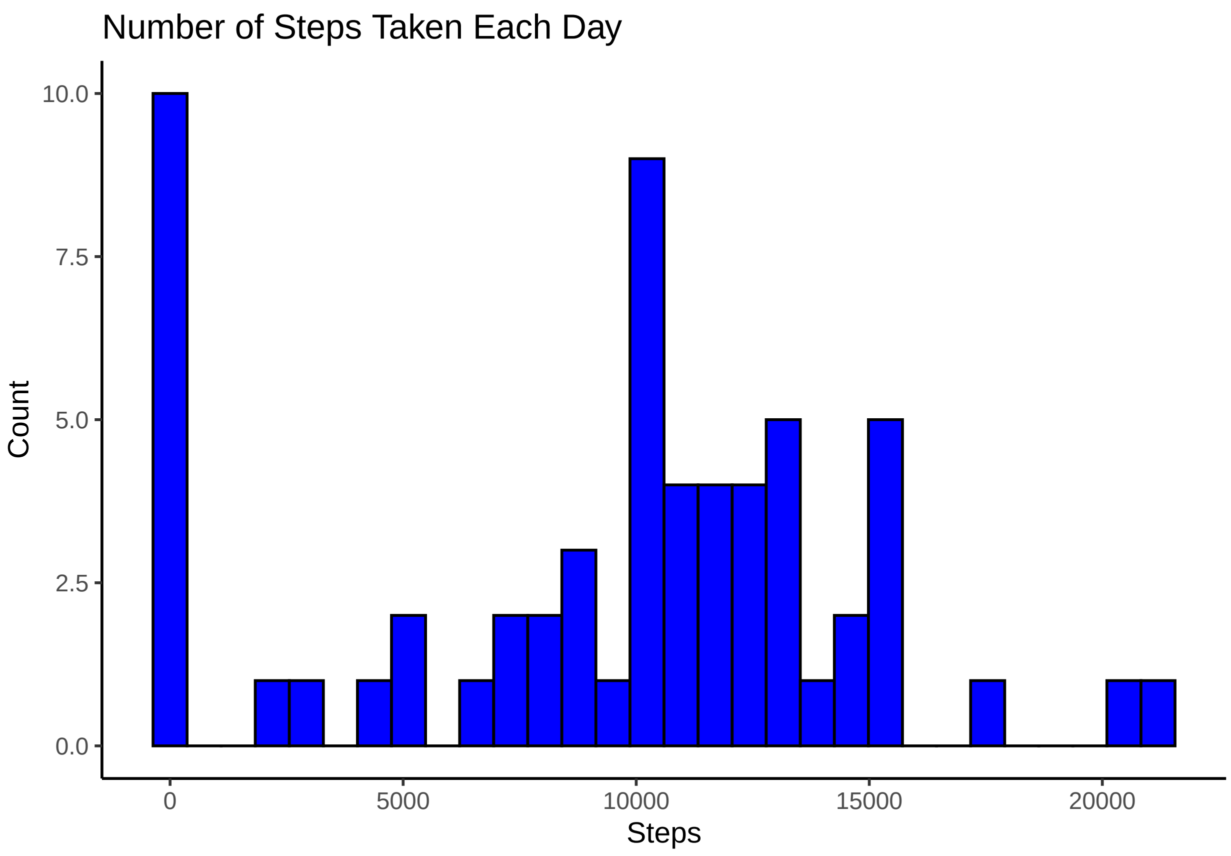
```r
df.interval
```

```
## # A tibble: 288 x 2
##    interval mean.steps
##       <int>      <dbl>
##  1        0     1.72
##  2        5     0.340
##  3       10     0.132
##  4       15     0.151
##  5       20     0.0755
##  6       25     2.09
##  7       30     0.528
##  8       35     0.868
##  9       40     0
## 10       45     1.47
## # ... with 278 more rows
```

This code imputes missing values by utilizinf the mean and then re-creates the histogram above with the imputed values.

```
replacewithmean <- function(x) replace(x, is.na(x), mean(x, na.rm = TRUE))
meandata <- activity %>% group_by(interval) %>% mutate(steps= replacewithmean(steps))

# Histogram after Imputation
df1<- activity %>% group_by(date) %>% summarise(
  total.steps = sum(steps,na.rm = TRUE)
)
ggplot(df1, aes(x=total.steps)) + geom_histogram(fill = "blue",color = "black") + labs(x = "Steps", y =
  ggtitle("Number of Steps Taken Each Day") + theme_classic()
```

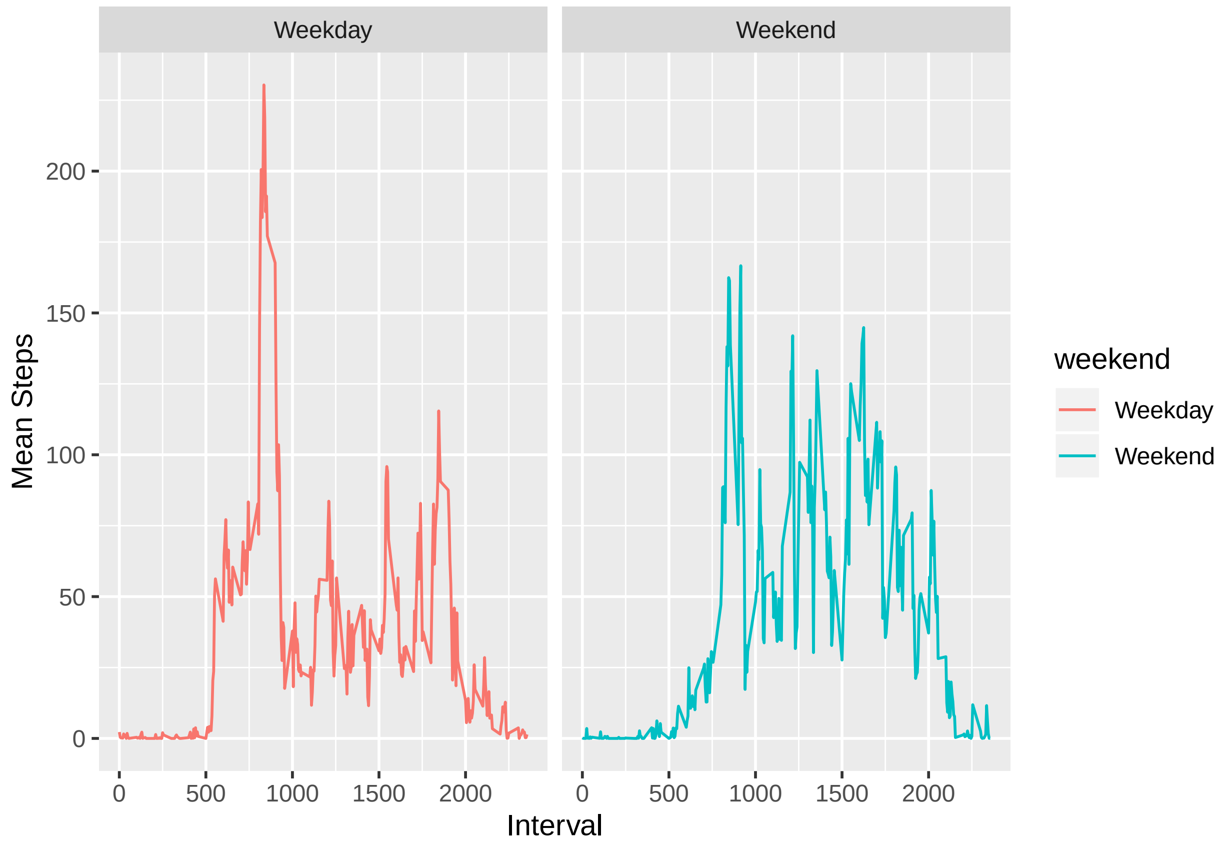
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The plots below aim to comapre the amount of activity on weekends versus weekdays based on the interval.

There is a large spike in activity in the weekdays, otherwise the number of steps taken on the weekends is generally slighlty higher.

```r
meandata$day<- weekdays(as.Date(meandata$date))
meandata$weekend <- ifelse(meandata$day == "Saturday" | meandata$day == "Sunday","Weekend","Weekday" )
#Get the mean steps for each interval by weekend and by interval
mean.steps<- meandata  %>% group_by(weekend,interval) %>% summarise(mean.steps = mean(steps))
#Plot data
ggplot(mean.steps, aes(x = interval,y= mean.steps, color = weekend)) + geom_line() +
  facet_grid(~ weekend) + labs(x = "Interval", y = "Mean Steps")
```


Weekday
Weekend
Mean Steps
Interval
weekend
Weekday
Weekend