PA1_template.Rmd

Emma Engler

2/24/2020

#Reproducible Research: Assignment 1

##Load and preprocess data ###Loading data

```
if(!file.exists('activity.csv')){
  unzip('activity.zip')
}
data <-read.csv('activity.csv')
summary(data)
```

```
##      steps                 date          interval     
##  Min.   :  0.00   2012-10-01:  288   Min.   :   0.0  
##  1st Qu.:  0.00   2012-10-02:  288   1st Qu.: 588.8  
##  Median :  0.00   2012-10-03:  288   Median :1177.5  
##  Mean   : 37.38   2012-10-04:  288   Mean   :1177.5  
##  3rd Qu.: 12.00   2012-10-05:  288   3rd Qu.:1766.2  
##  Max.   :806.00   2012-10-06:  288   Max.   :2355.0  
##  NA's   :2304     (Other)   :15840
```

```
head(data)
```

```
##   steps       date interval
## 1    NA 2012-10-01        0
## 2    NA 2012-10-01        5
## 3    NA 2012-10-01       10
## 4    NA 2012-10-01       15
## 5    NA 2012-10-01       20
## 6    NA 2012-10-01       25
```

What is mean total number of steps taken per day?

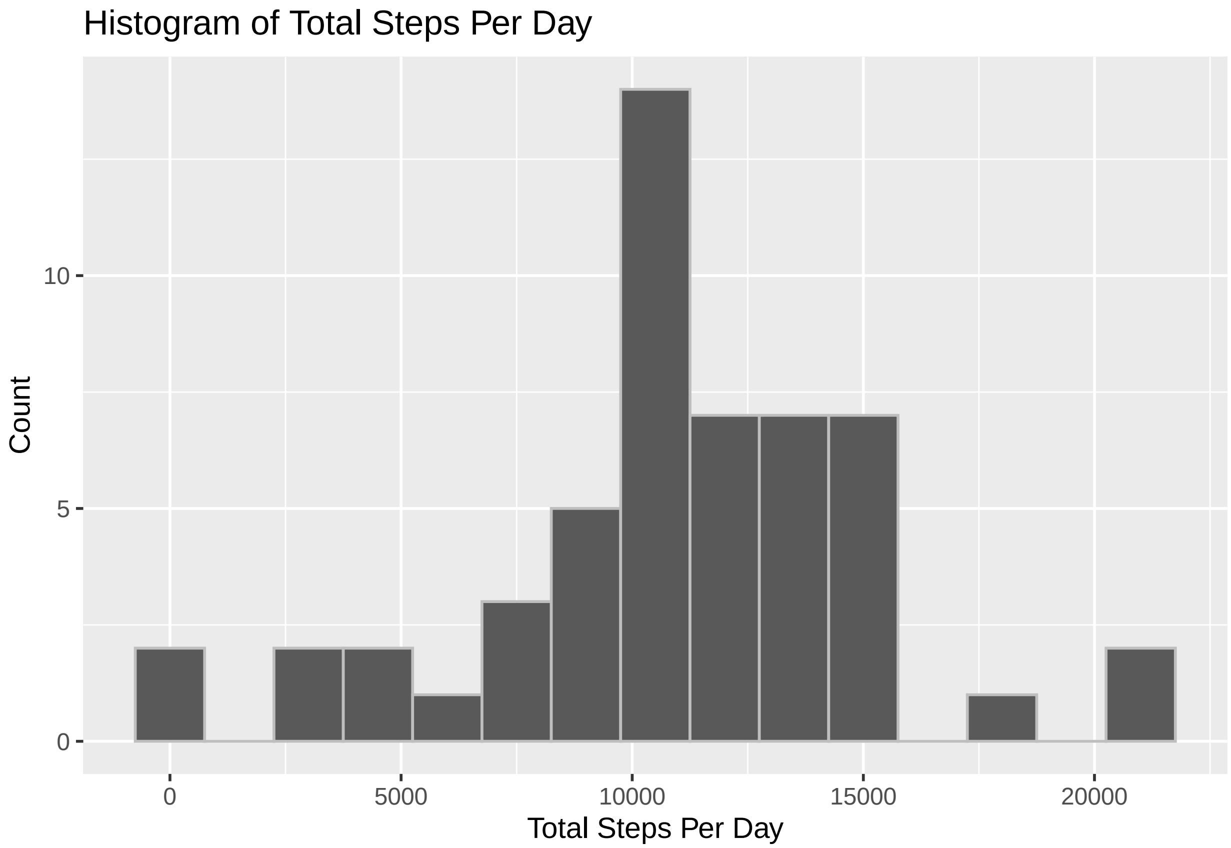
Missing values in the dataset are not used.

```
# Calculating the mean total number steps taken per day
steps <-aggregate(data$steps, by=list(Date=data$date), FUN=sum)
library(ggplot2)
names(steps)[names(steps)=="x"] <-"Total"
temp <-as.Date(steps$Date, "%Y-%m-%d")
steps$Date <-format(temp, format = "%m-%d")
head(steps)
```

```
##    Date Total
## 1 10-01    NA
## 2 10-02   126
## 3 10-03 11352
## 4 10-04 12116
## 5 10-05 13294
## 6 10-06 15420
```

Make a histogram of the total number of steps taken each day

```
hist1 <-ggplot(data=na.omit(steps), aes(Total)) +
  geom_histogram(binwidth=1500, colour="grey") +
  xlab("Total Steps Per Day") +
  ylab("Count") +
  ggtitle("Histogram of Total Steps Per Day")
print(hist1)
```



### Calculate and report mean and median of total number of steps per day

```
mean(na.omit(steps$Total))
```

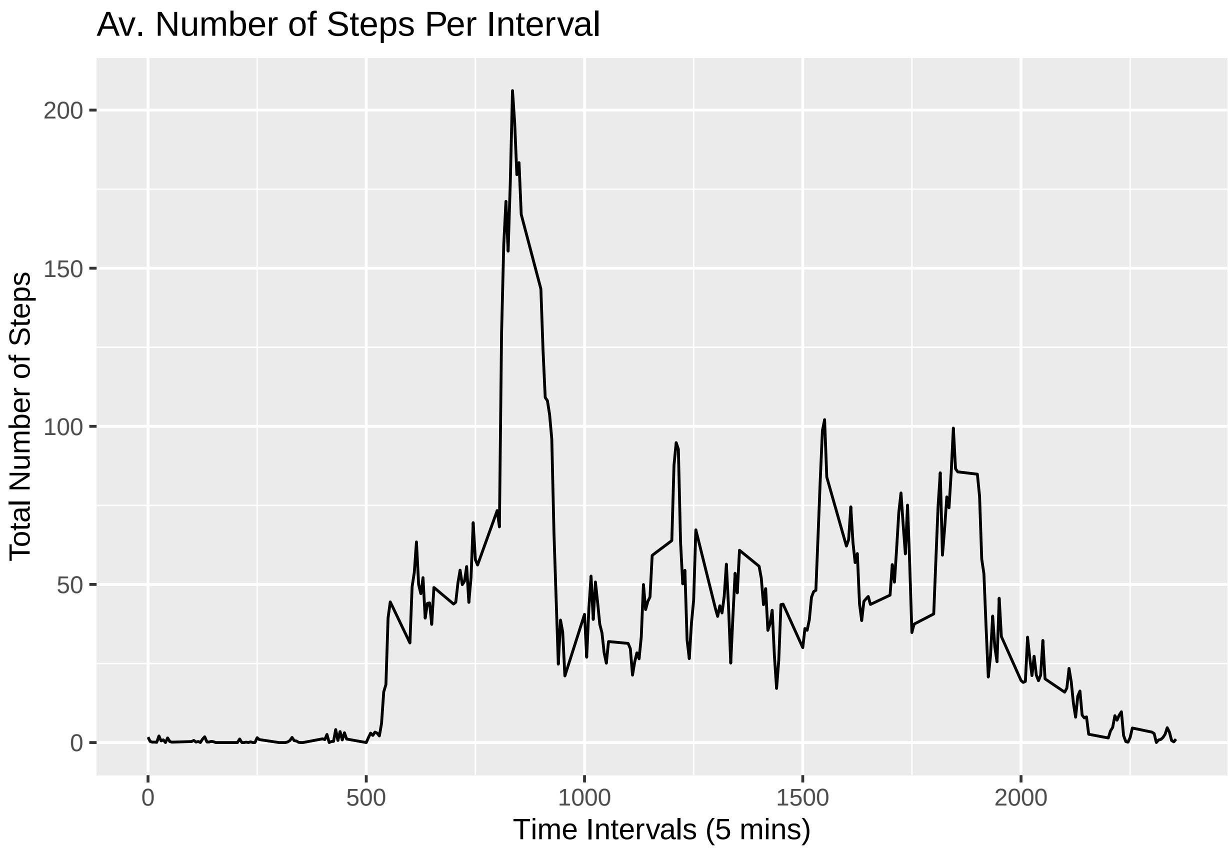
```
## [1] 10766.19
```

```
median(na.omit(steps$Total))
```

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

##What is the average daily activity pattern? Time series plot of the average number of steps taken. Shows the 5-min interval that, on average, contains the max number of steps.

```
# Make a Time Series Plot
steps_five <- aggregate(steps ~ interval, data=data, FUN=mean)
timeseries1 <-ggplot(data=steps_five, aes(x=interval, y=steps)) +
  geom_line() +
  xlab("Time Intervals (5 mins)") +
  ylab("Total Number of Steps") +
  ggtitle("Av. Number of Steps Per Interval")
print(timeseries1)
```



Which interval contains max number of steps?

```
head(steps_five)
```

```
##   interval     steps
## 1        0 1.7169811
## 2        5 0.3396226
## 3       10 0.1320755
## 4       15 0.1509434
## 5       20 0.0754717
## 6       25 2.0943396
```

```
steps_five[which(steps_five$steps==max(steps_five$steps)),]
```

```
##     interval    steps
## 104      835 206.1698
```

*Most steps: ‘r most.steps’ ## Imputing missing values There are many missing values in days/intervals. This introduces the possibility of bias into results or summaries.

Calculate and report total missing values in dataset

```
sapply(X=data, FUN=function(x) sum(is.na(x)))
```

```
##    steps     date interval
##     2304        0        0
```

*Number of missing values: ‘r missing.values’ ### Devise a strategy to fill in all missing values in dataset

```
replace_bymean <-function(num) replace(num, is.na(num), mean(num, na.rm=TRUE))
day_mean <-(data %>% group_by(interval) %>% mutate(steps=replace_bymean(steps)))
head(day_mean)
```

```
## # A tibble: 6 x 3
## # Groups:   interval [6]
##    steps date       interval
##    <dbl> <fct>         <int>
## 1 1.72   2012-10-01        0
## 2 0.340  2012-10-01        5
## 3 0.132  2012-10-01       10
## 4 0.151  2012-10-01       15
## 5 0.0755 2012-10-01       20
## 6 2.09   2012-10-01       25
```

```
sum(is.na(day_mean))
```

```
## [1] 0
```

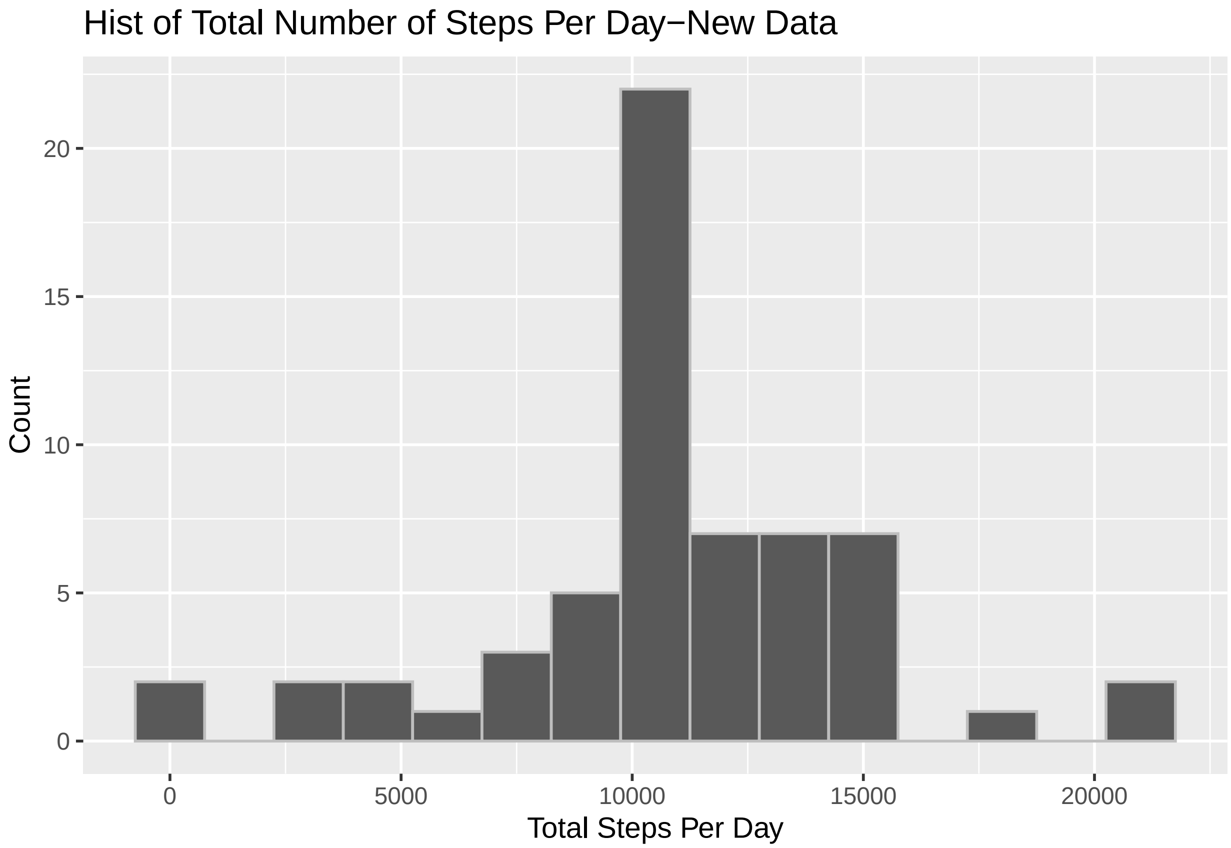
Create new dataset equal to the original dataset but with the missing data filled in

```
new_data <-as.data.frame(day_mean)
head(new_data)
```

```
##       steps       date interval
## 1 1.7169811 2012-10-01        0
## 2 0.3396226 2012-10-01        5
## 3 0.1320755 2012-10-01       10
## 4 0.1509434 2012-10-01       15
## 5 0.0754717 2012-10-01       20
## 6 2.0943396 2012-10-01       25
```

Make a histogram of the total number of steps taken each day

```r
new_steps <-aggregate(new_data$steps, by=list(new_data$date), FUN=sum)
names(new_steps)[names(new_steps)=="x"] <-"Total"
names(new_steps)[names(new_steps)=="Group1"] <-"Date"
hist2 <-ggplot(data=new_steps, aes(Total)) +
  geom_histogram(binwidth=1500, colour="grey") +
  xlab("Total Steps Per Day") +
  ylab("Count") +
  ggtitle("Hist of Total Number of Steps Per Day-New Data")
print(hist2)
```



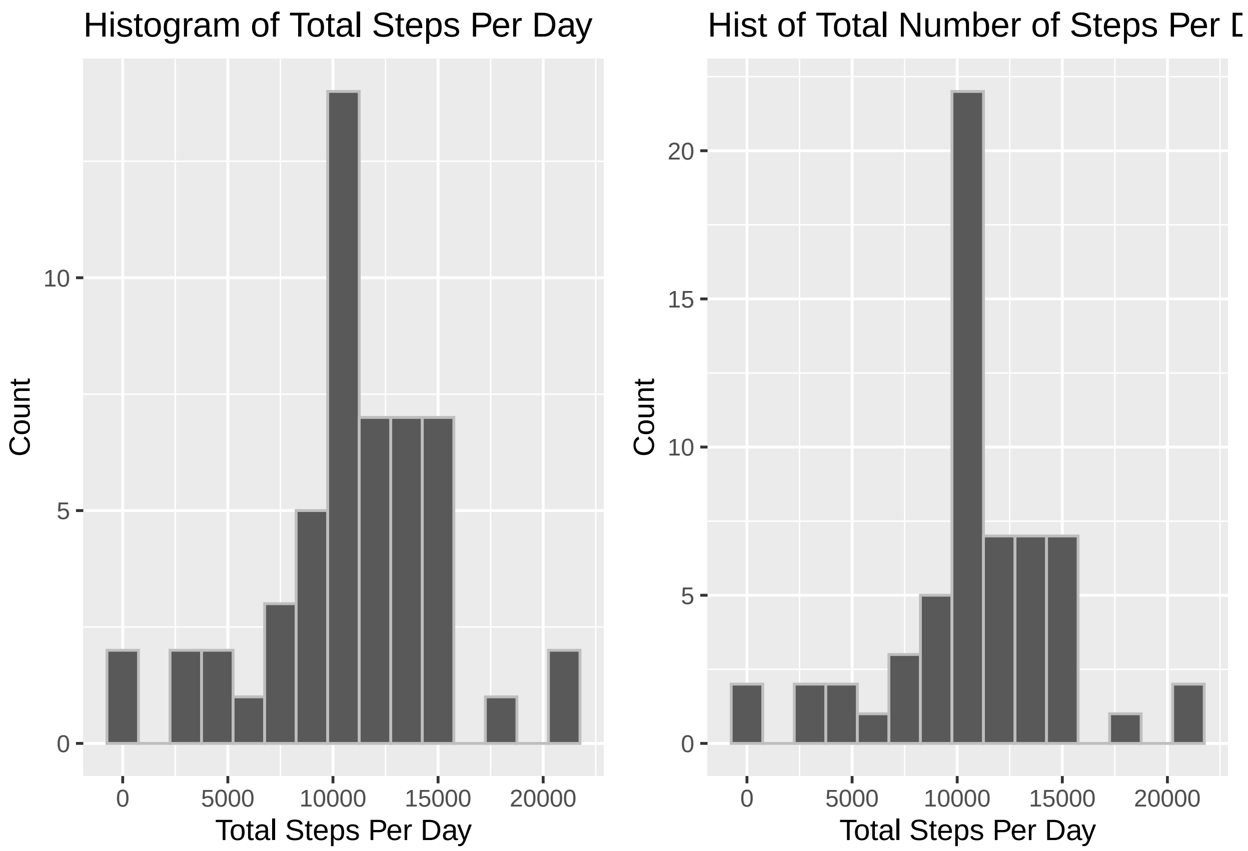
Comparison of the two plots

```r
library(grid)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
grid.arrange(hist1, hist2, ncol=2)
```



## Comparing mean and median

```r
mean(na.omit(steps$Total))
```

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

```r
median(na.omit(steps$Total))
```

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

```r
mean(new_steps$Total)
```

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

```r
median(new_steps$Total)
```

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

While the means of the dataset have remained the same, the medians of each dataset are slightly changed. The new data version shows a larger than that of the original with the NA's included.

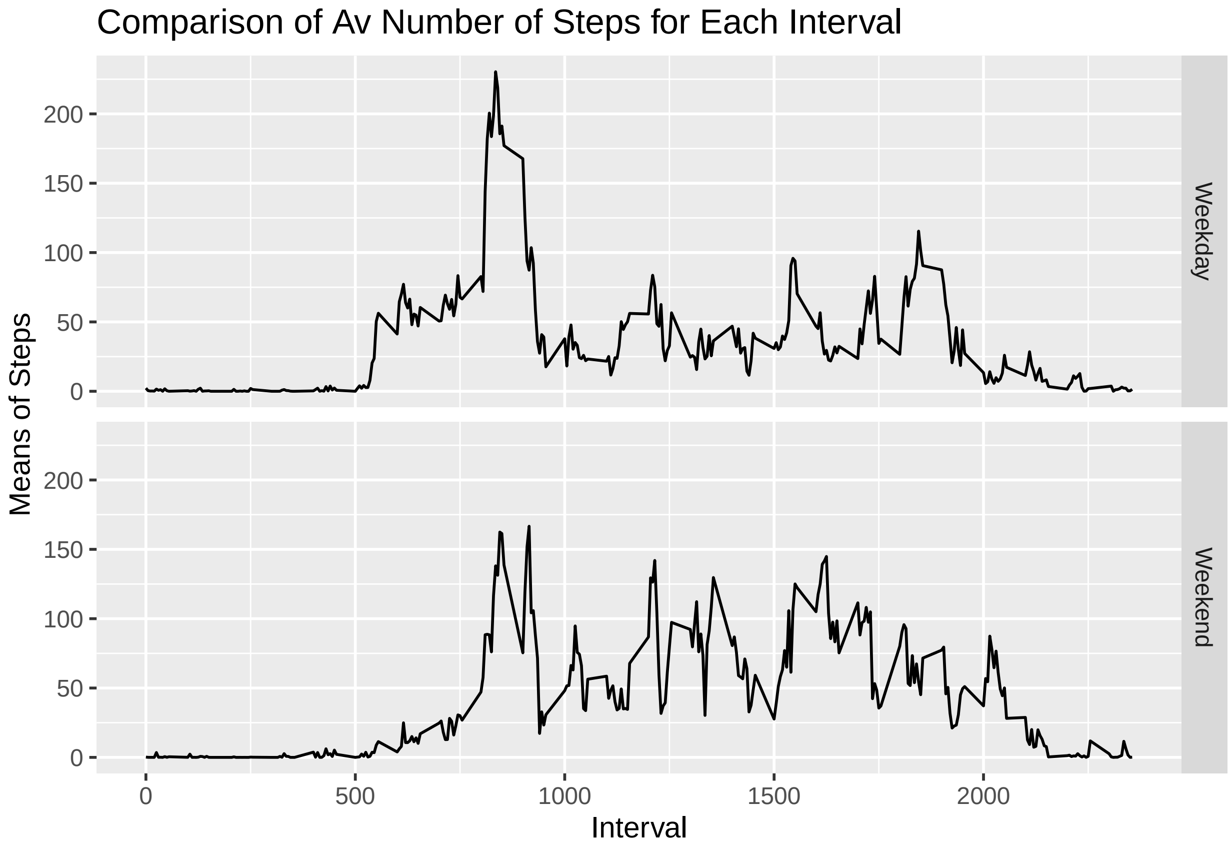
Comparing the average number of steps taken per 5-minue interval across weekdays and weekends

```
#Creating new variable in dataset of weekday and weekend
new_data$WeekendorWeekday <-ifelse(weekdays(as.Date(new_data$date)) %in% c("Monday", "Tuesday", "Wednes
head(new_data)
```

```
##       steps       date interval WeekendorWeekday
## 1 1.7169811 2012-10-01        0          Weekday
## 2 0.3396226 2012-10-01        5          Weekday
## 3 0.1320755 2012-10-01       10          Weekday
## 4 0.1509434 2012-10-01       15          Weekday
## 5 0.0754717 2012-10-01       20          Weekday
## 6 2.0943396 2012-10-01       25          Weekday
```

Making a panel plot to compare the average number of steps taken per interval across weekdays and weekends

```
new_data <-(new_data %>% group_by(interval, WeekendorWeekday) %>% summarise(Mean=mean(steps)))
ggplot(new_data, mapping=aes(x=interval, y=Mean)) +
  geom_line() +
  facet_grid(WeekendorWeekday~.) +
  xlab("Interval") +
  ylab("Means of Steps") +
  ggtitle("Comparison of Av Number of Steps for Each Interval")
```



It can be seen that there is different patterns between weekdays and weekends.