

# Activity Assignment

*Baird*

*January 28, 2017*

## Assignment objectives

### Commit containing full submission

- 1) Code for reading in the dataset and/or processing the data
  - 2) Histogram of the total number of steps taken each day
  - 3) Mean and median number of steps taken each day
  - 4) Time series plot of the average number of steps taken
  - 5) The 5-minute interval that, on average, contains the maximum number of steps
  - 6) Code to describe and show a strategy for imputing missing data
  - 7) Histogram of the total number of steps taken each day after missing values are imputed
  - 8) Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends
  - 9) All of the R code needed to reproduce the results (numbers, plots, etc.) in the report
- load data from activity.csv file

```
#quartz()
```

```
#dev.off()  
#plot.new()  
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.2.5
```

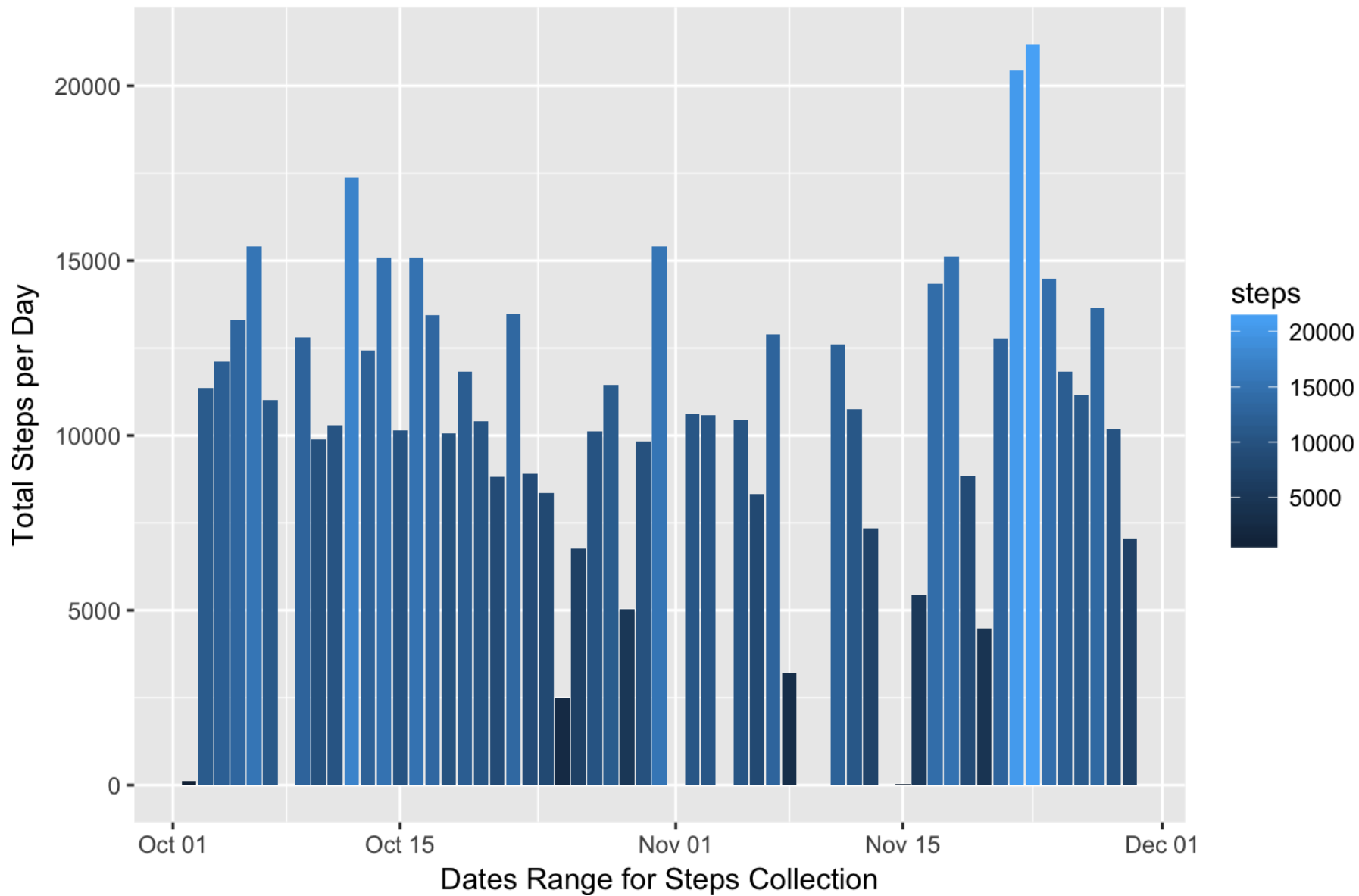
## -1)Code for reading in the dataset and/or processing the data

```
activity <- read.csv("activity.csv")  
activity$date2<-as.Date(as.character(activity$date), '%Y-%m-%d')  
activity_ag <- aggregate(steps~date2, data=activity, FUN=sum,na.rm=TRUE)  
activity_ag<-activity_ag[order(activity_ag$date2),]
```

## -2)Histogram of the total number of steps taken each day

```
#plot.new()  
#g<-g+ggtitle("This is a longer\ntitle than expected")  
ggplot(activity_ag) + geom_col(aes(x=date2,y=steps,group=date2,fill=steps)) +  
  xlab("Dates Range for Steps Collection")+  
  ylab("Total Steps per Day")+  
  ggtitle("Plot of Total Steps Taken Per Day \nDuration-Recorded Date Interval")+  
  theme(plot.margin=unit(c(0,0,0,0), "mm"))
```

Plot of Total Steps Taken Per Day  
Duration-Recorded Date Interval



`#dev.off()`

-3) Mean and median number of steps taken each day

```
activity_int <- aggregate(steps ~ interval, data=activity, FUN=mean)

print(paste0("Mean steps taken over interval: ", steps_mean<-mean(activity_ag$steps)))
```

```
## [1] "Mean steps taken over interval: 10766.1886792453"
```

```
print(paste0("Median steps taken over interval: ", steps_median<-median(activity_ag$steps)))
```

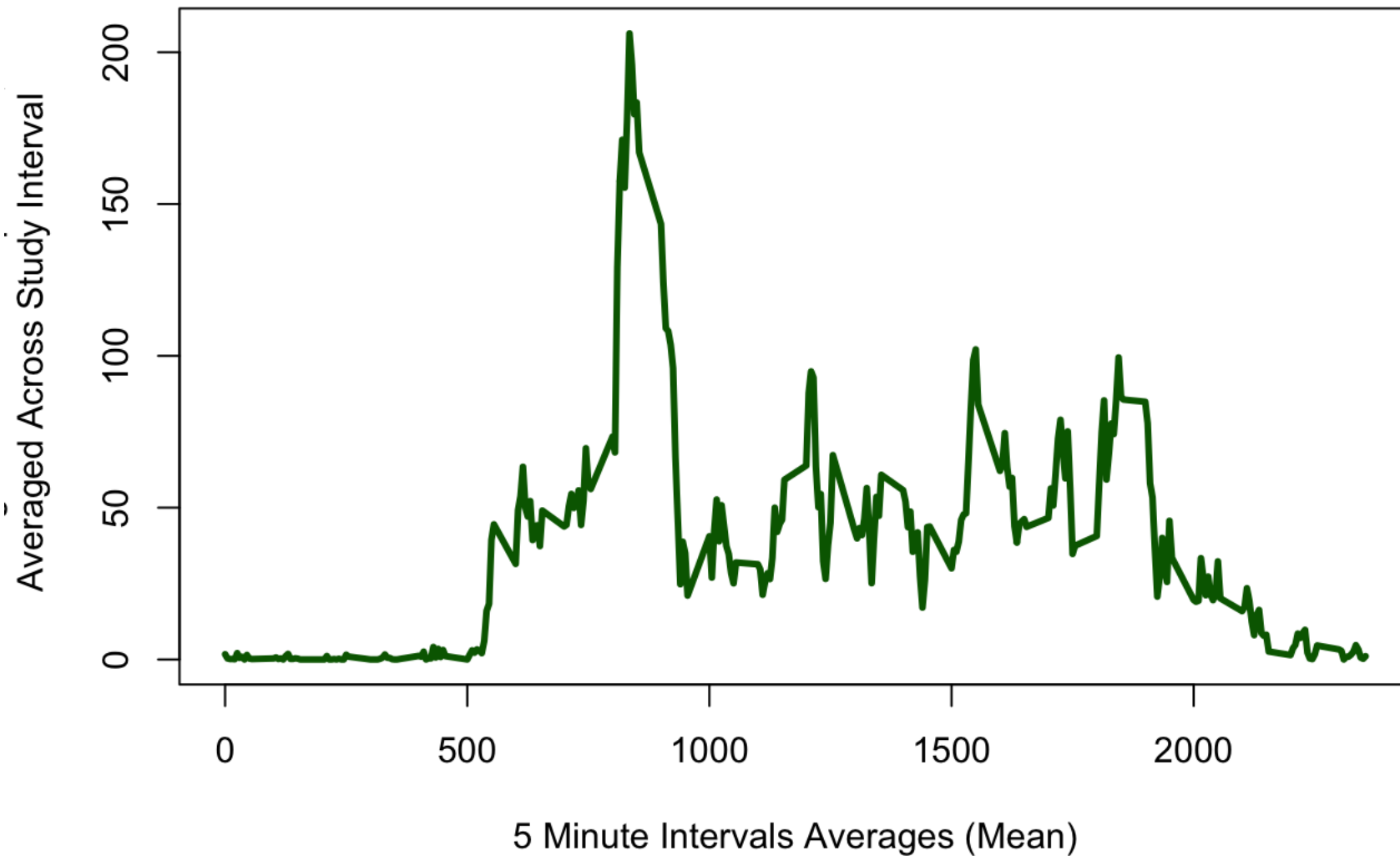
```
## [1] "Median steps taken over interval: 10765"
```

#### -4)Time series plot of the average number of steps taken

```
activity_interval_ag <- aggregate(steps ~ interval, activity, mean, rm.na=T)

plot(x = activity_interval_ag$interval,y = activity_interval_ag$steps,
     type = "l",lwd=3, col="dark green",
     main = "5 Minute Intervals \n Average Number of Steps Taken Across Study Interval",
     xlab = "5 Minute Intervals Averages (Mean)",
     ylab = "Average Number of Steps Taken,\n Averaged Across Study Interval")
```

## 5 Minute Intervals Average Number of Steps Taken Across Study Interval



`#dev.off()`

-5)The 5-minute interval that, on average, contains the maximum number of steps

```
print(paste0("The 5-minute interval that contains the maximum number of steps: ",
activity_interval_ag$interval[which.max(activity_interval_ag$steps)]))
```

```
## [1] "The 5-minute interval that contains the maximum number of steps: 835"
```

```
sum(is.na(activity))
```

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

```
#create new data group for adding fake data to missing steps
activity_merge<-activity
```

```
#create a function to return steps mean from time interval
step_inteveral <- function(int){
  activity_interval_ag$steps[activity_interval_ag$interval==int]
}
```

```
#testing function over missing data
```

```
#step_inteveral(activity_merge$interval[is.na(activity_merge$steps)])
```

```
#replace na data with mean steps
```

```
activity_merge$steps[is.na(activity_merge$steps)]<- round(as.numeric(lapply(activity_merge$interval[is.na(activity_merge$steps)],step_inteveral), digits = 0))
```

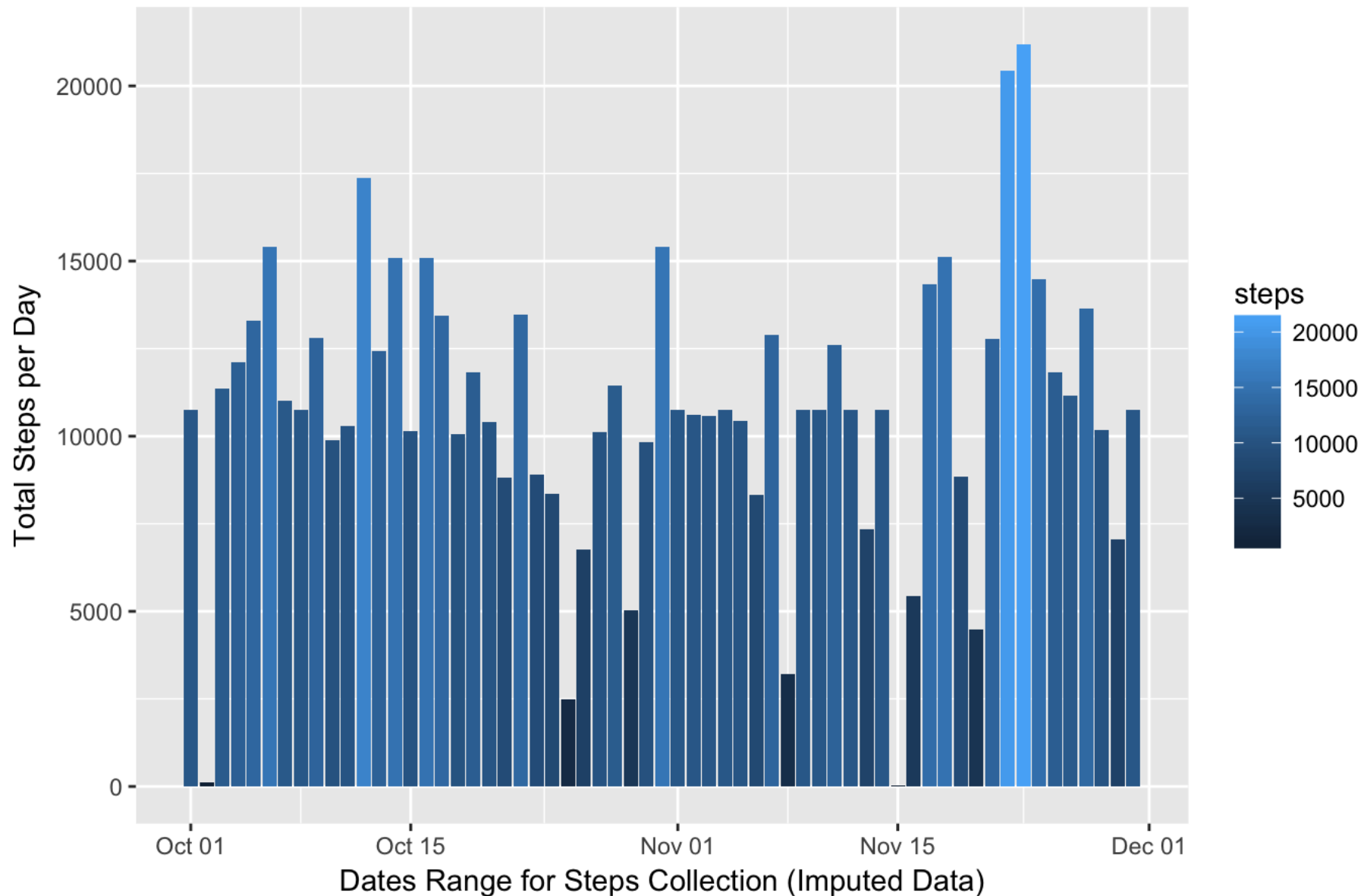
```
#activity_merge$steps[is.na(activity_merge$steps)]<- round(lapply(activity_merge$interval[is.na(activity_merge$steps)],step_inteveral), digits = 0)
```

```
activity_merge_ag <- aggregate(steps~date2, data=activity_merge, FUN=sum,na.rm=TRUE)
activity_merge_ag<-activity_merge_ag[order(activity_merge_ag$date2),]
```

## -6)Code to describe and show a strategy for imputing missing data

```
#plot.new()  
ggplot(activity_merge_ag) + geom_col(aes(x=date2,y=steps,group=date2,fill=steps)) +  
  xlab("Dates Range for Steps Collection (Imputed Data)") +  
  ylab("Total Steps per Day") +  
  ggtitle("Plot of Total Steps Taken Per Day \nDuration-Recorded Date Interval") +  
  theme(plot.margin=unit(c(0,0,0,0), "mm"))
```

Plot of Total Steps Taken Per Day  
Duration-Recorded Date Interval



```
activity_merge_int <- aggregate(steps ~ interval, data=activity_merge, FUN=mean)
```

```
print(paste0("Mean steps taken over supplemented (imputed) interval: ", steps_merge_mean<-mean(  
activity_merge_ag$steps)))
```



```
## [1] "Mean steps taken over supplemented (imputed) interval: 10765.6393442623"
```

```
print(paste0("Median steps taken over supplemented (imputed) interval: ", steps_merge_median<-median(activity_merge_ag$steps)))
```

```
## [1] "Median steps taken over supplemented (imputed) interval: 10762"
```

## -7)Histogram of the total number of steps taken each day after missing values are imputed

```
activity_interval_merge_ag <- aggregate(steps ~ interval, activity_merge, mean, rm.na=T)

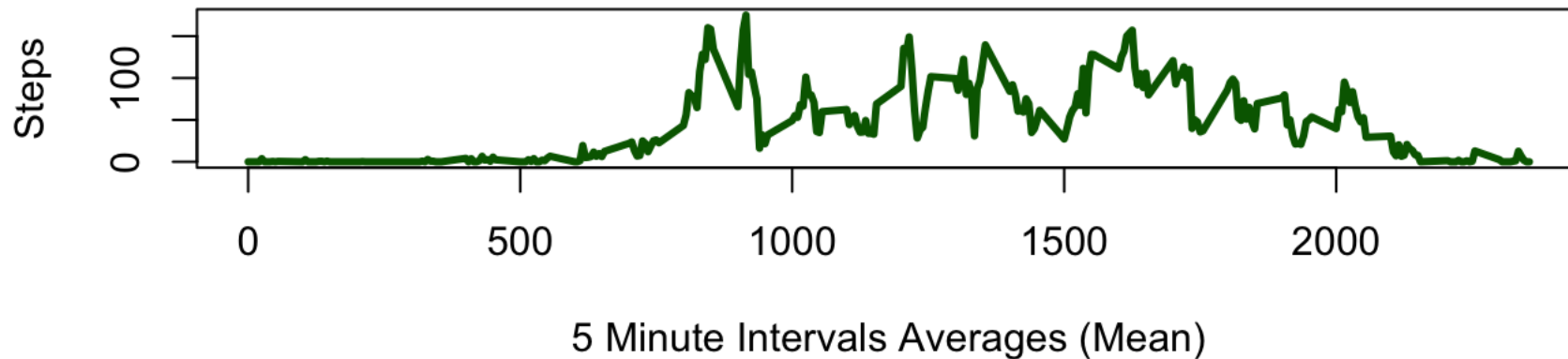
activity_weekends<-activity[weekdays(as.Date(activity$date)) %in% c("Saturday", "Sunday"),]
activity_weekdays<-activity[weekdays(as.Date(activity$date)) %in% c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday"),]

activity_int_weekend_ag <- aggregate(steps ~ interval, activity_weekends, mean, rm.na=T)
activity_int_weekday_ag <- aggregate(steps ~ interval, activity_weekdays, mean, rm.na=T)
par(mfrow = c(2, 1))
plot(x = activity_int_weekend_ag$interval,y = activity_int_weekend_ag$steps,
     type = "l",lwd=3, col="dark green",
     main = "5 Minute Intervals \n Average Number of Steps Taken Across Study Interval\nWeekends",
     xlab = "5 Minute Intervals Averages (Mean)",
     ylab = "Steps")

plot(x = activity_int_weekday_ag$interval,y = activity_int_weekday_ag$steps,
     type = "l",lwd=3, col="blue",
     main = "5 Minute Intervals \n Average Number of Steps Taken Across Study Interval\nWeekdays")
```

```
days",  
  xlab = "5 Minute Intervals Averages (Mean)",  
  ylab = "Steps")
```

### 5 Minute Intervals Average Number of Steps Taken Across Study Interval Weekends



### 5 Minute Intervals Average Number of Steps Taken Across Study Interval Weekdays

