Reproducible Research - Week 2 - Project 1

1. Loading and preparing the data

```
library(ggplot2)
library(dplyr)
```

Loading libraries

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
activity <- read.csv("activity.csv")
```

Loading the data (file 'activity.csv' in the working directory)

```
# printing out the first 20 rows
head(activity,20)
```

Processing for analysis

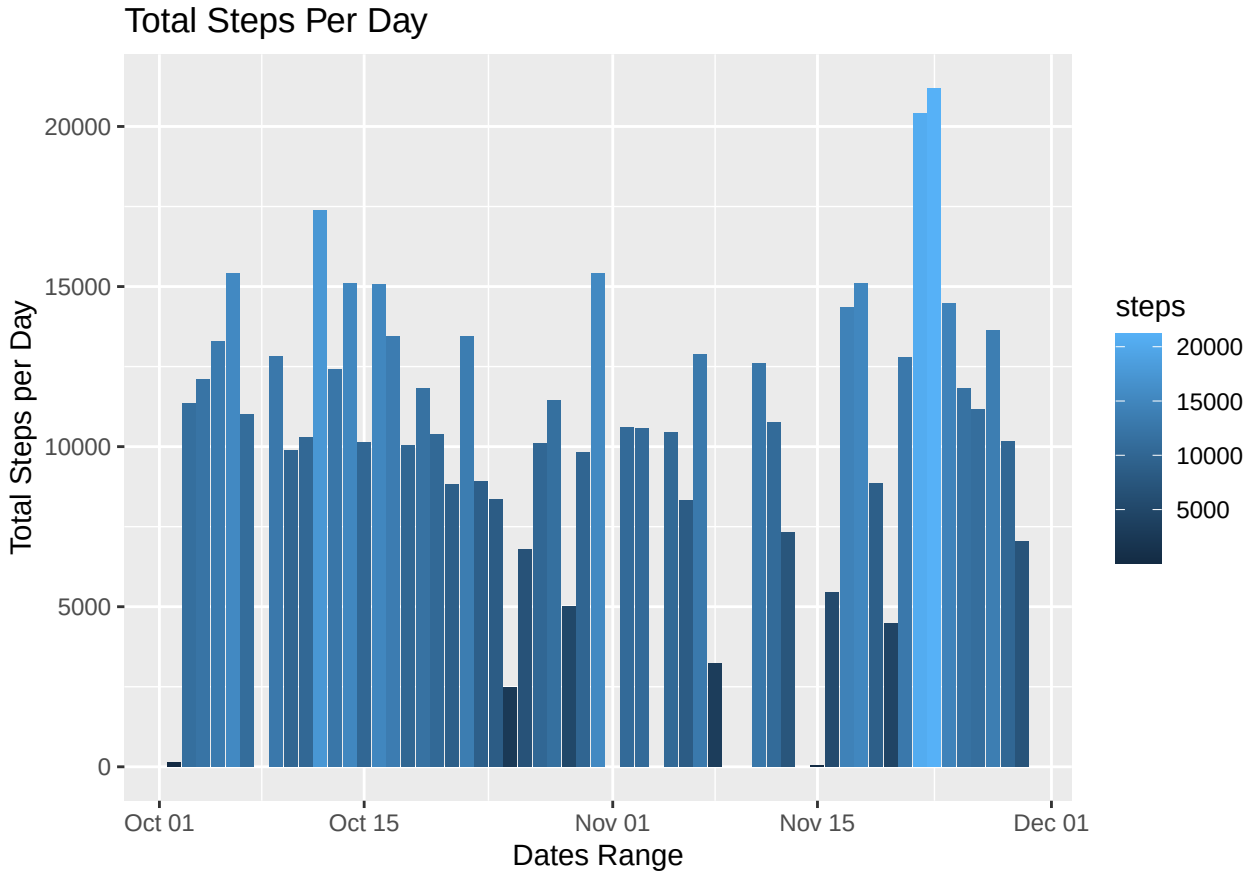
```
##    steps       date interval
## 1     NA 2012-10-01        0
## 2     NA 2012-10-01        5
## 3     NA 2012-10-01       10
## 4     NA 2012-10-01       15
## 5     NA 2012-10-01       20
## 6     NA 2012-10-01       25
## 7     NA 2012-10-01       30
## 8     NA 2012-10-01       35
## 9     NA 2012-10-01       40
## 10    NA 2012-10-01       45
## 11    NA 2012-10-01       50
## 12    NA 2012-10-01       55
## 13    NA 2012-10-01      100
## 14    NA 2012-10-01      105
```

```
## 15    NA 2012-10-01      110
## 16    NA 2012-10-01      115
## 17    NA 2012-10-01      120
## 18    NA 2012-10-01      125
## 19    NA 2012-10-01      130
## 20    NA 2012-10-01      135
```

```
# Aggregating
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),]
```

What is mean total number of steps taken per day?

```
ggplot(activity_ag) + geom_col(aes(x=date2,y=steps,group=date2,fill=steps)) +
  xlab("Dates Range")+
  ylab("Total Steps per Day")+
  ggtitle("Total Steps Per Day")+
  theme(plot.margin=unit(c(0,0,0,0),"mm"))
```



Plotting

Mean and median number of steps taken each day

```
activity_int <- aggregate(steps ~ interval, data=activity, FUN=mean)
print(paste0("Mean steps per day: ", steps_mean<-mean(activity_ag$steps)))
```

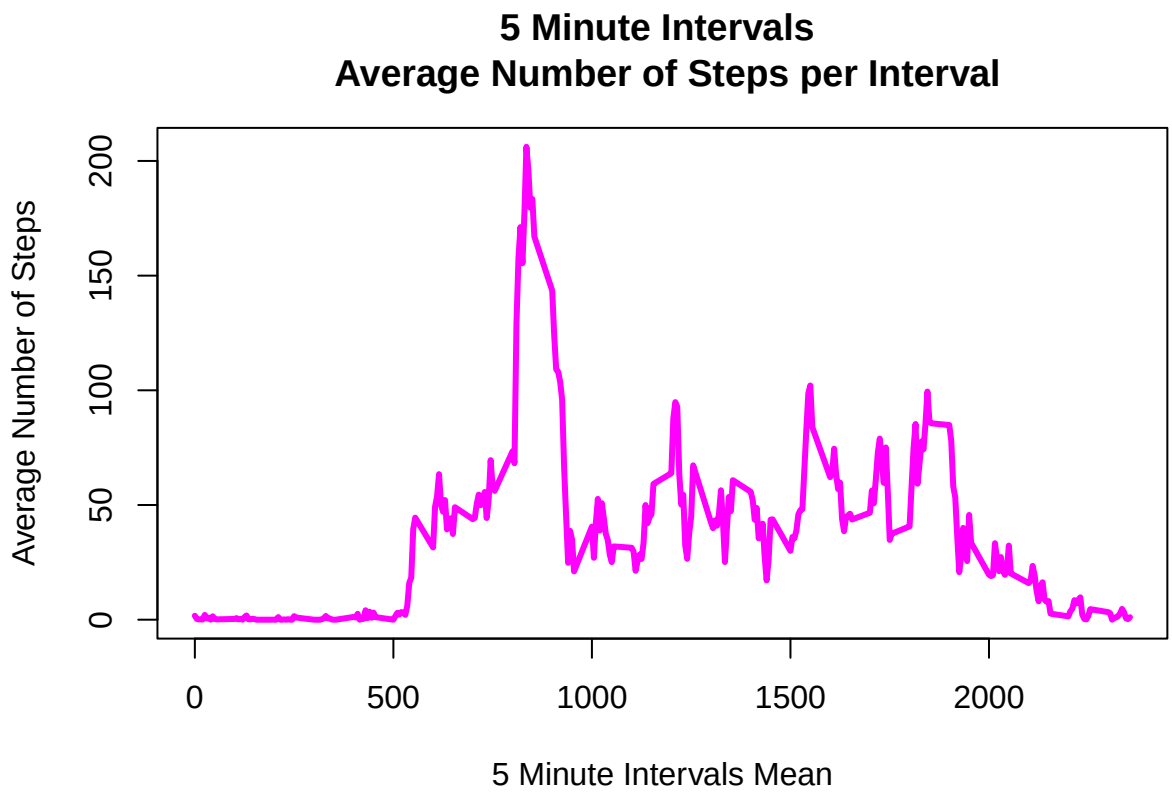
```
## [1] "Mean steps per day: 10766.1886792453"
```

```
print(paste0("Median steps per day: ", steps_median<-median(activity_ag$steps)))
```

```
## [1] "Median steps per day: 10765"
```

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="magenta",
        main = "5 Minute Intervals \n Average Number of Steps per Interval",
        xlab = "5 Minute Intervals Mean",
        ylab = "Average Number of Steps")
```



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"
```

```
max_steps <- which.max(activity_interval_ag$steps)
activity_interval_ag[max_steps, ]
```

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

Code to describe and show a strategy for imputing missing data

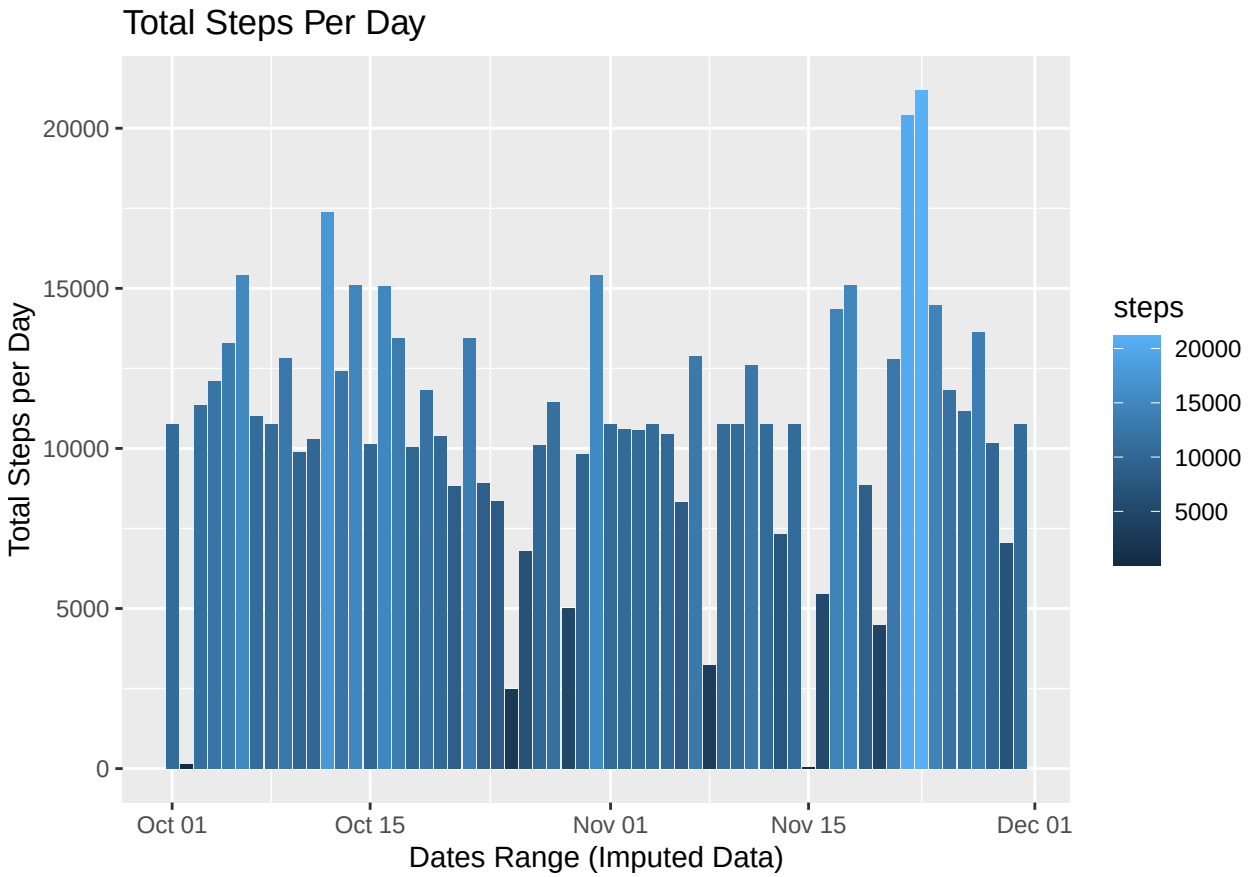
```
sum(is.na(activity))
```

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

```
#creating a new dataset for adding data to missing values
activity_merge <- activity

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

#Replacing NAs with means
activity_merge$steps[is.na(activity_merge$steps)]<- round(as.numeric(lapply(activity_merge$interval[is.

#Aggregating both datasets
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),]
```

Histogram of the total number of steps taken each day after missing values are imputed

```
ggplot(activity_merge_ag) + geom_col(aes(x=date2,y=steps,group=date2,fill=steps)) +
  xlab("Dates Range (Imputed Data)")+
  ylab("Total Steps per Day")+
  ggtitle("Total Steps Per Day")+
  theme(plot.margin=unit(c(0,0,0,0),"mm"))
```




```
summary (activity_ag$steps)
```

Mean and median total number of steps taken per day WITHOUT filling in the missing values

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      41    8841   10765   10766   13294   21194
```

```
summary (activity_merge_ag$steps)
```

Mean and median total number of steps taken per day WITH filling in the missing values

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      41    9819   10762   10766   12811   21194
```

Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

```r
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","T

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="red", main = "5-Minute Intervals \n Weekends", xlab = "5-Minute Intervals

plot(x = activity_int_weekday_ag$interval,y = activity_int_weekday_ag$steps,
        type = "l",lwd=3, col="orange", main = "5-Minute Intervals \n Weekdays", xlab = "5-Minute Inter
```