Reproducible Research: Peer Assessment 1

Charles Guthrie

January 9, 2016

1 Overview

It is now possible to collect a large amount of data about personal movement using activity monitoring devices such as a Fitbit, Nike Fuelband, or Jawbone Up. These type of devices are part of the "quantified self" movement – a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. But these data remain under-utilized both because the raw data are hard to obtain and there is a lack of statistical methods and software for processing and interpreting the data.

1.1 Loading and preprocessing the data

This assignment makes use of data from a personal activity monitoring device. This device collects data at 5 minute intervals through out the day. The data consists of two months of data from an anonymous individual collected during the months of October and November, 2012 and include the number of steps taken in 5 minute intervals each day.

The data was downloaded from https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip (https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip) on to a local drive for processing and analyses using the statistical program R (version 3.2.1). The dataset consists of 17,568 observations and three variables–steps (number of steps taking in a 5-minute interval with missing values are coded as NA), date (the date on which the measurement was taken in YYYY-MM-DD format), and interval (identifier for the 5-minute interval in which measurement was taken).

```
###code for loading and inital processing of dataset
##packages used:
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.2.2
```

```
##
## 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
```

```
library(ggvis)
```

```
## Warning: package 'ggvis' was built under R version 3.2.3
```

```
##Load data
wk1_data <- read.csv(file = "activity.csv", header = T, stringsAsFactors = F)
wk1_data <- tbl_df(wk1_data)
wk1_data <- wk1_data %>% mutate(date = as.Date(date)) #convert character data type to date data type
str(wk1_data)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Date, format: "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

```
summary(wk1_data)
```

```
##      steps             date               interval
##  Min.   :  0.00   Min.   :2012-10-01   Min.   :   0.0
##  1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.: 588.8
##  Median :  0.00   Median :2012-10-31   Median :1177.5
##  Mean   : 37.38   Mean   :2012-10-31   Mean   :1177.5
##  3rd Qu.: 12.00   3rd Qu.:2012-11-15   3rd Qu.:1766.2
##  Max.   :806.00   Max.   :2012-11-30   Max.   :2355.0
##  NA's   :2304
```

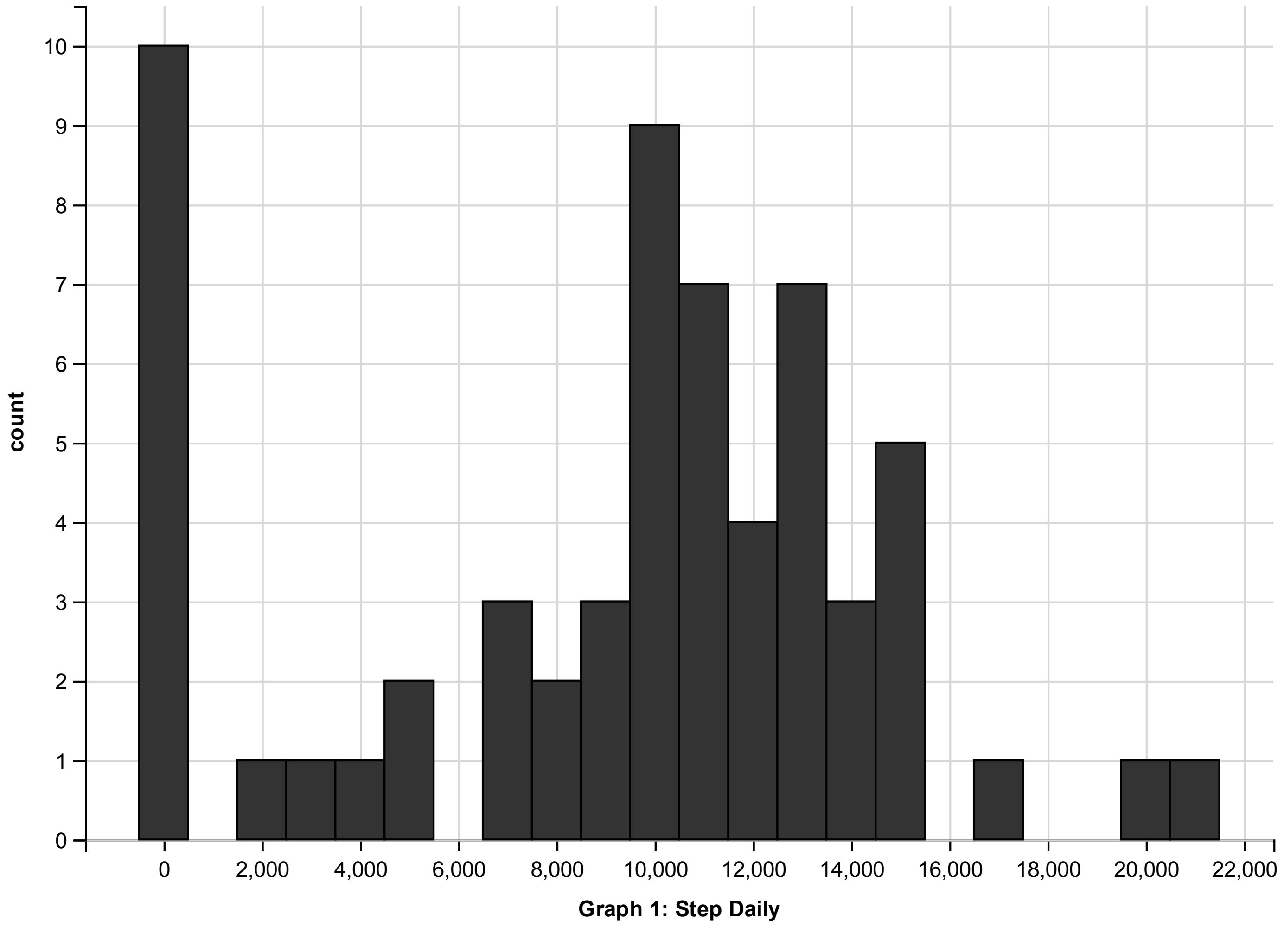
2 Questions

The present dataset was used to answer the questions in the following sections.

2.1 What is mean total number of steps taken per day?

```
##what is the mean total of steps taken per day?
steps_daily <- wk1_data %>% group_by(date) %>% summarise(steps_daily = sum(steps, na.rm
=T))
steps_daily %>% ggvis(~steps_daily) %>% layer_histograms() %>% add_axis("x", title = "Gra
ph 1: Step Daily")
```

```
## Guessing width = 1000 # range / 22
```



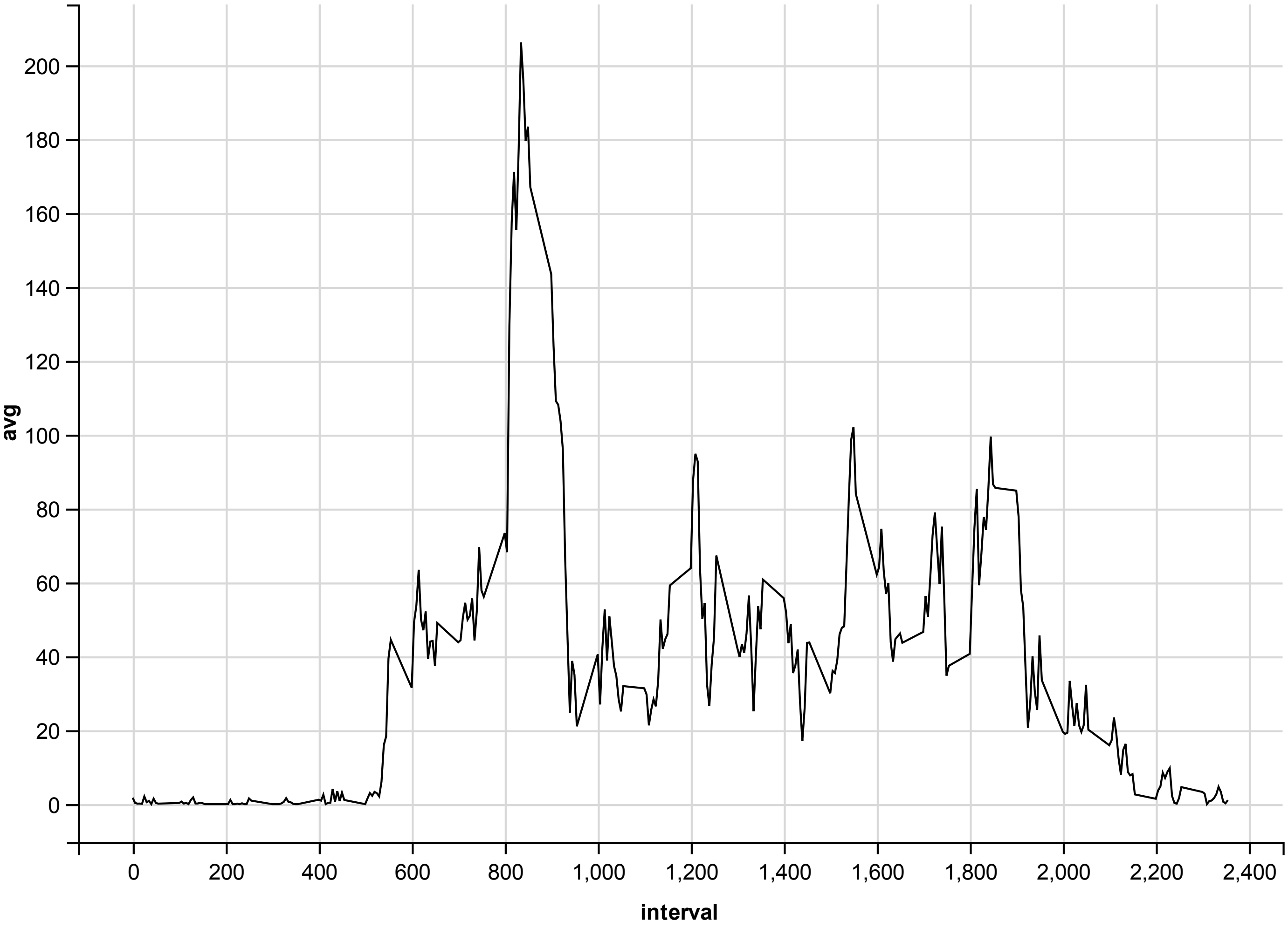
```
summary_steps_daily <- steps_daily %>% summarise(avg = mean(steps_daily, na.rm = T), medi
an = median(steps_daily, na.rm =T))
summary_steps_daily
```

```
## Source: local data frame [1 x 2]
##
##       avg median
##     (dbl)  (int)
## 1 9354.23  10395
```

The dataset was grouped by unique dates and each date’s recorded steps were summed (graph 1 shows the distribution of the total steps). The mean (9354.2295082) and median (10395) were also calculated.

2.2 What is the average daily activity pattern?

```
##what is the average daily activity pattern?
avg_daily <- wk1_data %>% group_by(interval) %>% summarise(avg = mean(steps, na.rm = T))
avg_daily %>% ggvis(~interval, ~avg) %>% layer_lines()
```



```
#Calculate and report the mean and median of the total number of steps taken per day
max_interval <- avg_daily[max(avg_daily$avg),]
```

A time series plot was created to display the average number of steps taken across the five-minute intervals (see Graph 2). The five minute interval, on average across all days in the data, that contained the maxim number of steps was found to be 1705 at an average of 56.3018868 steps.

2.3 Imputing missing values

A brief missing data analysis was conducted: the table below shows the counts of missing data (NA) by variable in the data.

```
##Imputing missing values and answer previous questions.
wk1_data %>% summarise(NAs_steps = sum(is.na(steps)), NA_date = sum(is.na(date)), NA_interval = sum(is.na(interval))) #number of NAs by variable
```

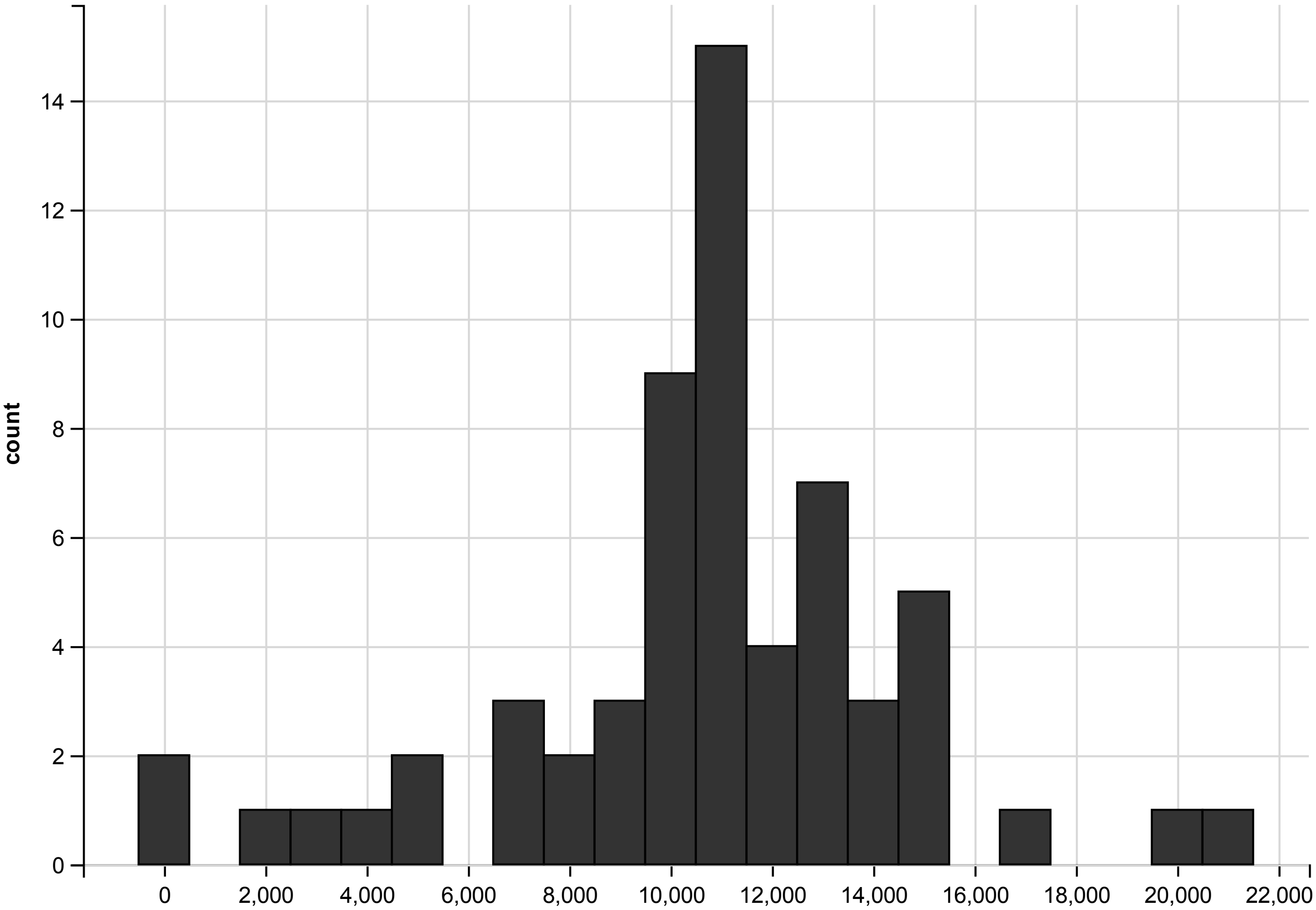
```
## Source: local data frame [1 x 3]
##
##   NAs_steps NA_date NA_interval
##       (int)   (int)       (int)
## 1      2304       0           0
```

```
#aggregated averages by unique interval number
means_by_interval <- wk1_data %>% group_by(interval) %>% summarise(steps = mean(steps, na.rm = T)) %>% as.data.frame()
#function to impute missing data with mean of corresponding interval group
impute_data <- function(steps, interval){
  imputed_result <- NA
  if(!is.na(steps)){
    imputed_result <- c(steps)
  }else{
    imputed_result <- (means_by_interval[means_by_interval$interval==interval, "steps"])
  }
  return(imputed_result)
}
wk1_data_imputed <- wk1_data
wk1_data_imputed$steps <- mapply(impute_data, wk1_data_imputed$steps, wk1_data_imputed$interval)
wk1_data_imputed
```

```
## Source: local data frame [17,568 x 3]
##
##        steps       date interval
##        (dbl)     (date)    (int)
## 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
## 7  0.5283019 2012-10-01       30
## 8  0.8679245 2012-10-01       35
## 9  0.0000000 2012-10-01       40
## 10 1.4716981 2012-10-01       45
## ..       ...        ...      ...
```

```
#
steps_daily_imputed <- wk1_data_imputed %>% group_by(date) %>% summarise(steps_daily_imputed = sum(steps, na.rm = T))
steps_daily_imputed %>% ggvis(~steps_daily_imputed) %>% layer_histograms() %>% add_axis("x", title = "Graph 3: Imputed Daily Steps Data")
```

```
## Guessing width = 1000 # range / 22
```



Graph 3: Imputed Daily Steps Data

```
summary_steps_daily_imputed <- steps_daily_imputed %>% summarise(avg = mean(steps_daily_i
mputed, na.rm = T), median = median(steps_daily_imputed, na.rm = T))
summary_steps_daily_imputed
```

```
## Source: local data frame [1 x 2]
##
##        avg   median
##      (dbl)    (dbl)
## 1 10766.19 10766.19
```

Missing data was imputed by replacing missing values (i.e. NA) with the mean value from the observations aggregate interval group. The previous questions were addressed with the imputed data. Graph 3 shows the distribution the total daily steps and the mean (1.076618910^{4}) and median (1.076618910^{4}) were calculated for the imputed data. Imputing the data with the chosen method made the distribution of the steps conform to a more normal distribution and may distort the interpretation of the data.

2.4 Are there differences in activity patterns between weekdays and weekends?

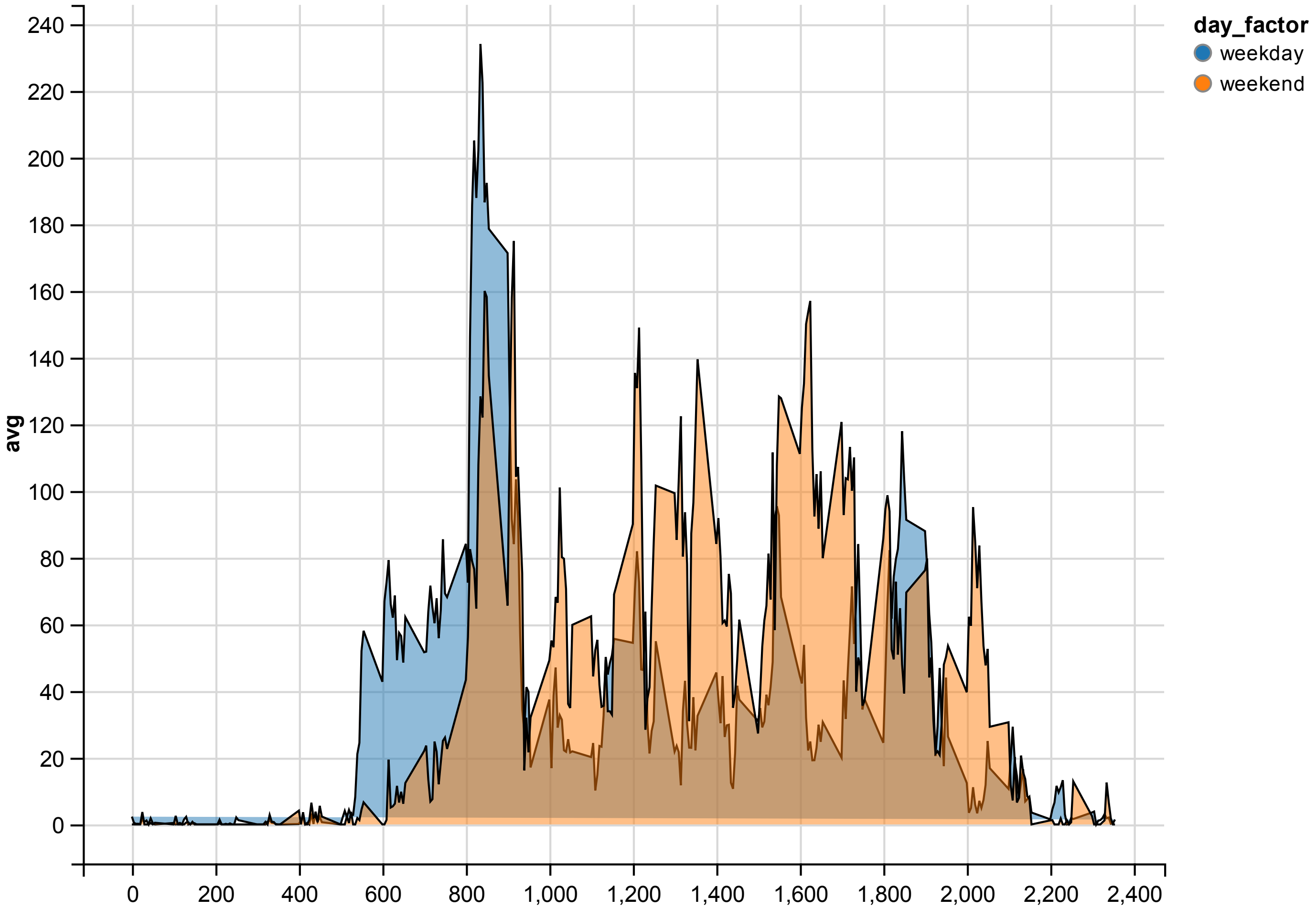
A line graph was created to explore the differences in activity patterns between weekdays and weekends (Graph 4). The graph indicates that there are larger averages of steps during the early intervals during the weekdays when compared to weekends. However, step averages are more consistent across intervals during the weekends.

```
##Are there differences in activity patterns between weekdays and weekends?
#function to convert date variable into two-level factor variable
working_for_the_weekend <- function(date) {
  day <- weekdays(date)
  if(day %in% c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday")){
    return("weekday")
  }else if(day %in% c("Saturday", "Sunday")){
    return("weekend")
  }
}

wk1_data$day_factor <- sapply(wk1_data$date, FUN=working_for_the_weekend)
wk1_data <- wk1_data %>% mutate(day_factor = factor(day_factor))
wk1_data$day_factor %>% table()
```

```
## .
## weekday weekend
##   12960    4608
```

```
activity <- wk1_data %>% group_by(interval, day_factor) %>% summarise(avg = mean(steps, n
a.rm = T))
activity %>% ggvis(~interval, ~avg, fill = ~day_factor, fillOpacity := .5) %>% group_by(d
ay_factor) %>% layer_lines() %>% add_axis("x", title = "Graph 4: Weekend and Weekday Inte
rvals")
```



Graph 4: Weekend and Weekday Intervals