Project_1

Byron Jones

January 19, 2017

Reproducible Research

Project 1

Submitted by Byron Jones

```
# Project dataset contains the following variables:
## *steps*:    number of steps taking in a 5-minute interval (missing values are coded as NA)
## *date*:     the date on which the measurement was taken in YYYY-MM-DD format
## *interval*: identifier for the 5-minute interval in which measurement was taken

# The dataset is stored in a comma-separated-value (CSV) file and there are a total of 17,568 observati

# Required for the assignment:
## 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 w
## 9.All of the R code needed to reproduce the results (numbers, plots, etc.) in the report
```

Set working directory, read in the dataset and look at first few rows

```
# set working directory
setwd("//Phchbs-s3159/jonesby1$/data/Byron/R/Reproducible Research/Projects")

# read in the dataset and look at first few rows
activity<-read.csv("activity.csv",head=TRUE)
print(head(activity))
```

```
##   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
```

Identify days of the week and add this new variable to existing activity list

```
week.days<-weekdays(as.Date(unlist(activity$date)))
activity[["day"]]<-week.days
names(activity)<-c("steps", "date","interval", "day")
```

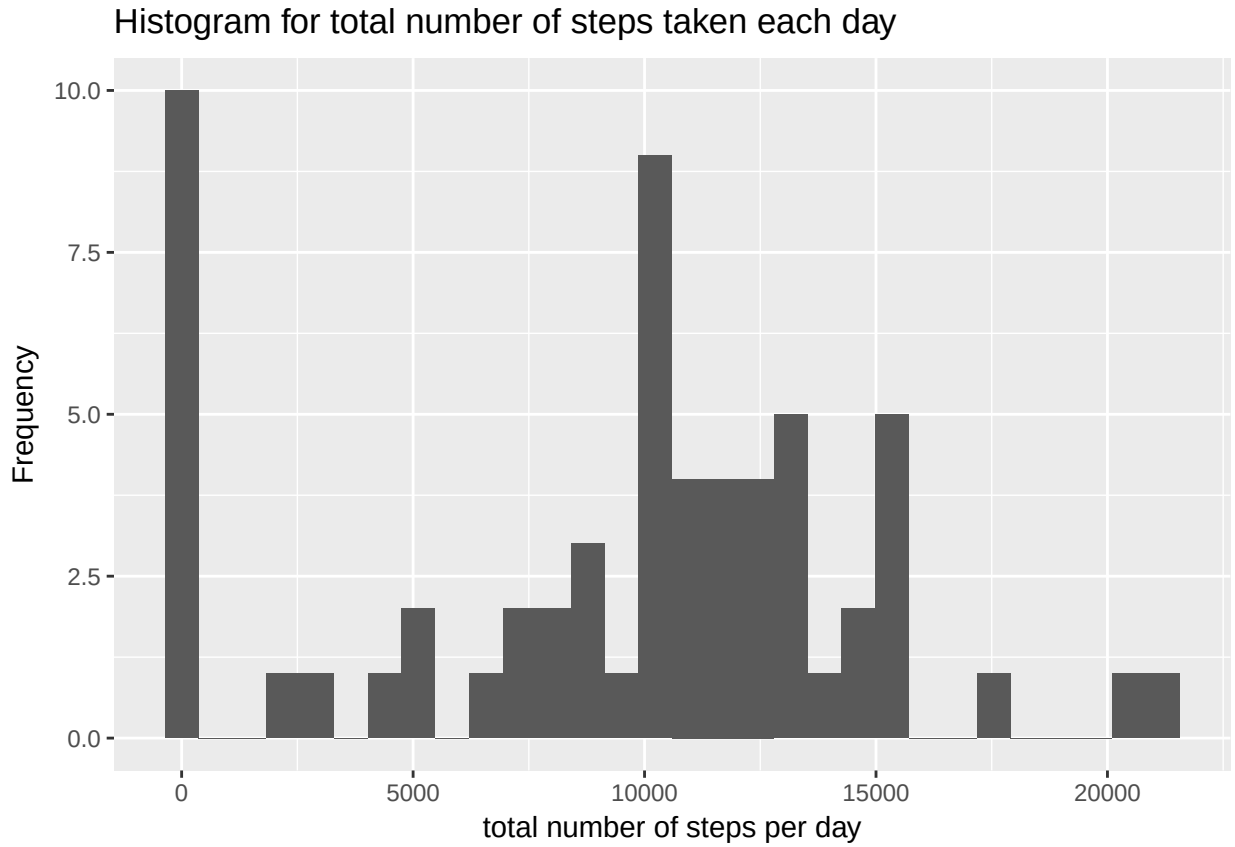
Calculate mean total number of steps taken each day and mean total number of steps in each 5-minute interval using the aggregate function

Histogram of total number of steps taken each day

```
## aggregate data for each day
day.mean<-as.numeric(unlist(aggregate(activity$steps, by=list(activity$date), FUN=sum, na.rm=TRUE)[2]))

## load library ggplot2
library(ggplot2)
```

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

```
day.mean.df<-data.frame(day.mean)
ggplot(data=day.mean.df, aes(day.mean.df$day.mean)) +
  geom_histogram(bins=30) +
   labs(title="Histogram for total number of steps taken each day") +
   labs(x="total number of steps per day", y="Frequency")
```



Note the high frequency of zero counts

Mean and median of total number per day

```
print(mean(day.mean))
```

```
## [1] 9354.23
```

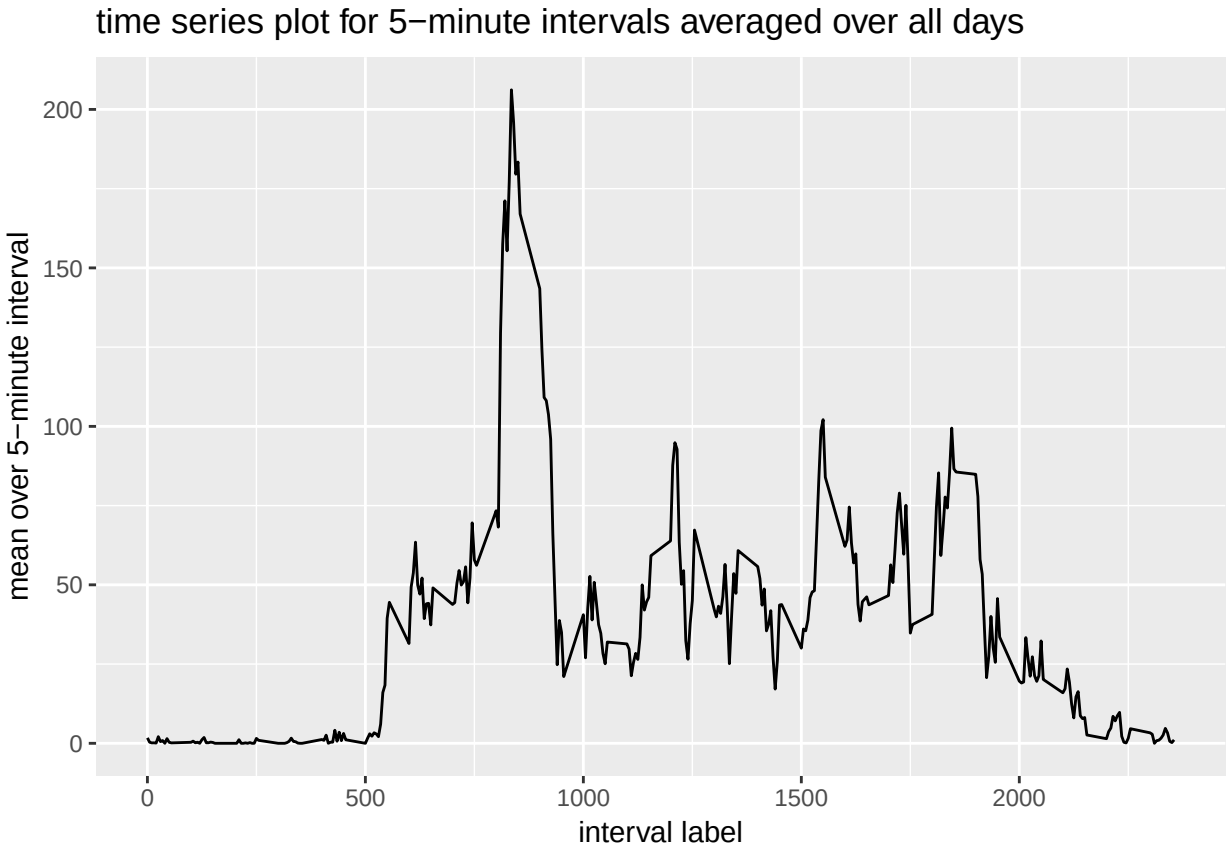
```
print(median(day.mean))
```

```
## [1] 10395
```

Mean is lower than median due to high frequency of intervals with zero steps

Time series plot of average number of steps taken

```
## calculate mean for each 5-minute interval using the aggregate function
interval.mean<-as.numeric(unlist(aggregate(activity$steps, by=list(activity$interval), FUN=mean, na.rm='

## plotting values for x-axis
interval.values<-unique(activity$interval)
## create a data frame for use in ggplot2
interval.df<-data.frame(cbind(interval.values,interval.mean))

## time series plot using ggplot2
ggplot(interval.df, aes(x=interval.values, y=interval.mean)) +
  geom_line() +
  ggtitle("time series plot for 5-minute intervals averaged over all days") +
  labs(x="interval label") +
  labs(y="mean over 5-minute interval")
```



Highest mean for 5-minute interval labels between 500 and 1000

Maximum mean number of steps

```
## maximum mean number of steps
max.mean<-max(interval.mean)
print(max.mean)
```

```
## [1] 206.1698
```

Maximum mean number of steps is 206.17

Label of the interval that contains the maximum

```
## interval containing maximum mean
max.interval<-interval.values[interval.mean==max(interval.mean)]
print(max.interval)
```

```
## [1] 835
```

Interval that contains the maximum is 835

Total number of rows with missing data

```
## total number of rows with missing data (NA)
total.NA<-sum(is.na(activity$steps))
print(total.NA)
```

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

Total number of rows with missing data is 2304

Imputation of missing data

Imputation rule: replace NA with the mean for that day

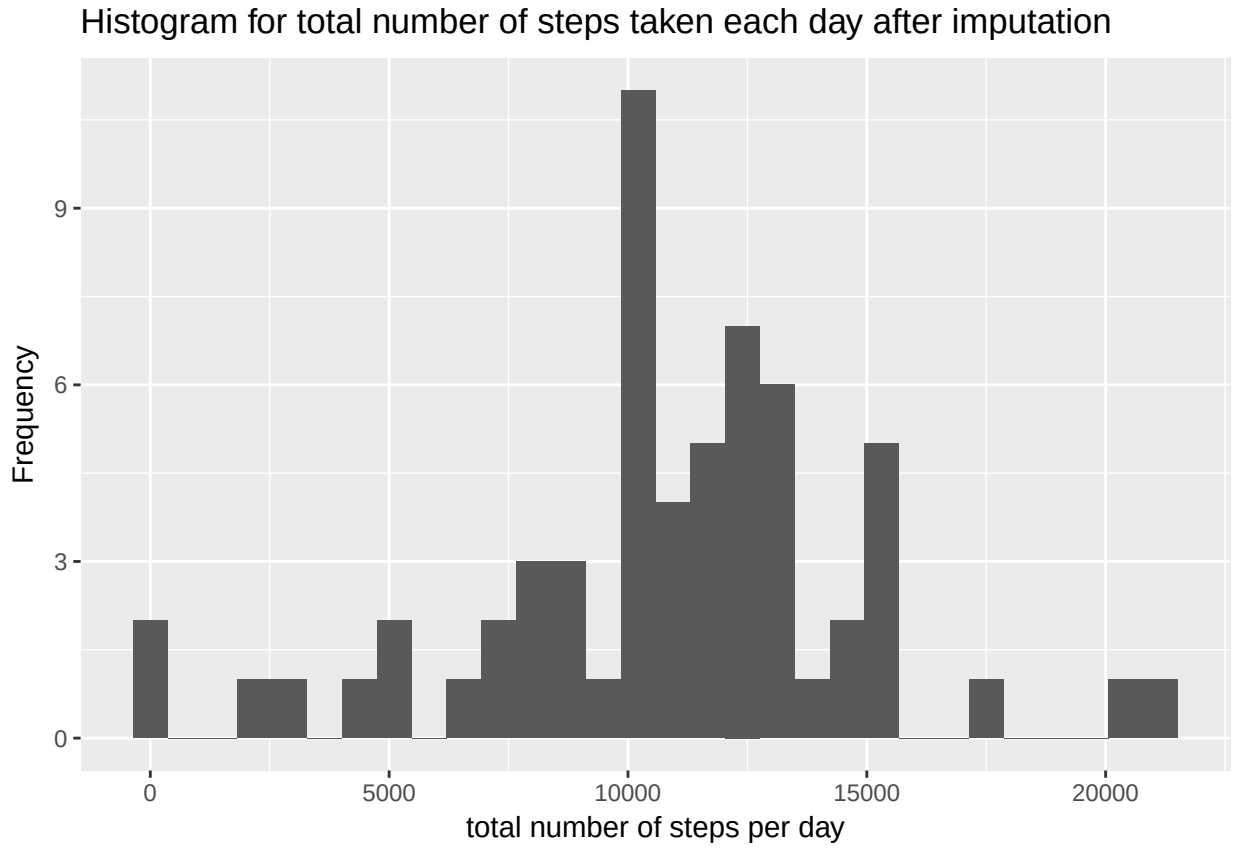
```
## imputation rule: replace NA with the mean for that day

## calculate the mean for each day of the week
Saturday.mean<-mean(activity$steps[activity$day=="Saturday"],na.rm=TRUE)
Sunday.mean<-mean(activity$steps[activity$day=="Sunday"],na.rm=TRUE)
Monday.mean<-mean(activity$steps[activity$day=="Monday"],na.rm=TRUE)
Tuesday.mean<-mean(activity$steps[activity$day=="Tuesday"],na.rm=TRUE)
Wednesday.mean<-mean(activity$steps[activity$day=="Wednesday"],na.rm=TRUE)
Thursday.mean<-mean(activity$steps[activity$day=="Thursday"],na.rm=TRUE)
Friday.mean<-mean(activity$steps[activity$day=="Friday"],na.rm=TRUE)

## impute missing data
## copy original dataset
activity.imp<-activity
activity.imp$steps[is.na(activity.imp$steps) & activity.imp$day=="Saturday"]<-Saturday.mean
activity.imp$steps[is.na(activity.imp$steps) & activity.imp$day=="Sunday"]<-Sunday.mean
activity.imp$steps[is.na(activity.imp$steps) & activity.imp$day=="Monday"]<-Monday.mean
activity.imp$steps[is.na(activity.imp$steps) & activity.imp$day=="Tuesday"]<-Tuesday.mean
activity.imp$steps[is.na(activity.imp$steps) & activity.imp$day=="Wednesday"]<-Wednesday.mean
activity.imp$steps[is.na(activity.imp$steps) & activity.imp$day=="Thursday"]<-Thursday.mean
activity.imp$steps[is.na(activity.imp$steps) & activity.imp$day=="Friday"]<-Friday.mean
```

Recalculate using imputed data set and compare with results obtained when missing data were not imputed

```
## calculate total number of steps taken each day using the aggregate function
day.mean.imp<-as.numeric(unlist(aggregate(activity.imp$steps, by=list(activity.imp$date), FUN=sum, na.r
```

Histogram of total number of steps after imputation

```
day.mean.imp.df<-data.frame(day.mean.imp)
ggplot(data=day.mean.imp.df, aes(day.mean.imp.df$day.mean.imp)) +
  geom_histogram(bins=30) +
   labs(title="Histogram for total number of steps taken each day after imputation") +
   labs(x="total number of steps per day", y="Frequency")
```



Histogram is now less skewed: high frequency of zeros has been reduced

Identify weekends and add this identifier to the existing activity list

```
## identify weekends
weekend<-(activity.imp$day=="Saturday" | activity.imp$day=="Sunday")

## add this to existing activity list
activity.imp[["daytype"]]<-weekend
activity.imp$daytype[activity.imp$daytype]<-"weekend"
activity.imp$daytype[activity.imp$daytype==FALSE]<-"weekday"
```

Panel plot for week days and weekend days separately

```
library(dplyr)
```

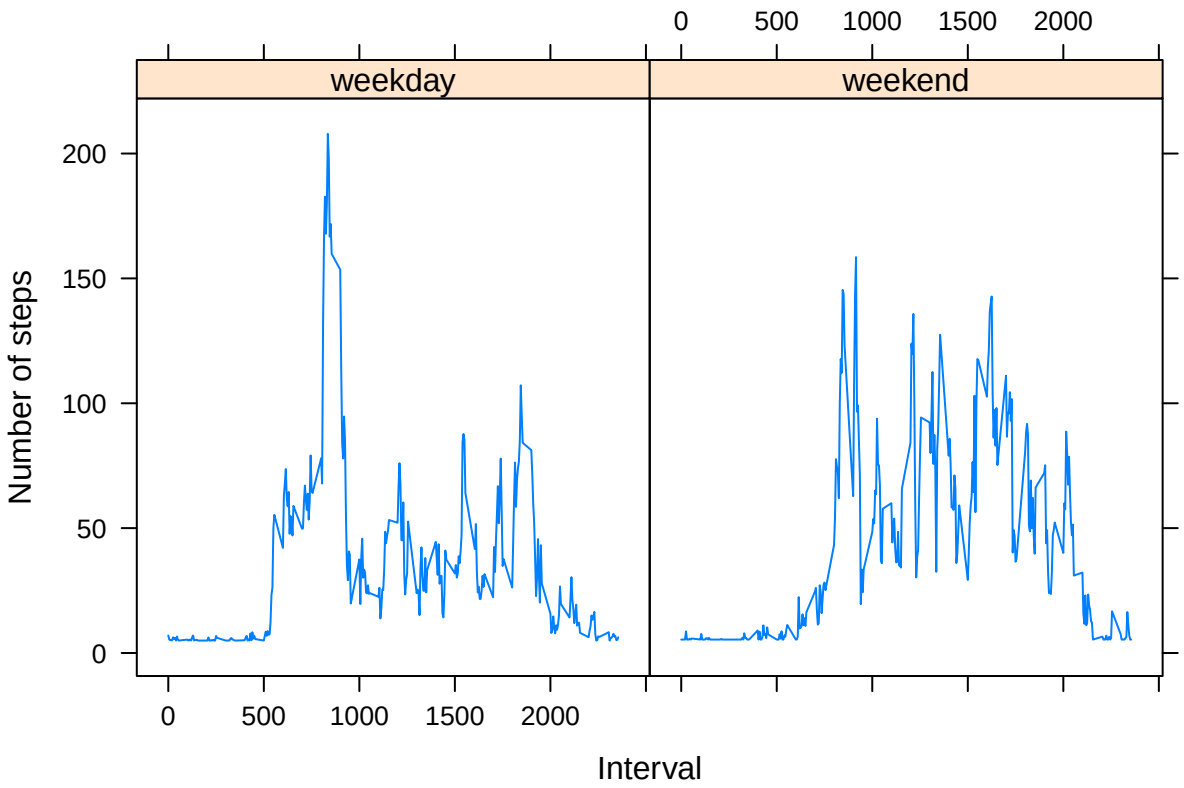
```
## Warning: package 'dplyr' was built under R version 3.2.5
```

```
##
## 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.grp <- group_by(activity.imp, daytype, interval)
avgInt <- summarise(activity.grp, AvgSteps = mean(steps))

library(lattice)
xyplot(AvgSteps ~ interval | factor(daytype), avgInt, type = "l",
       xlab = "Interval", ylab = "Number of steps")
```



Conclusion: number of steps per 5-minute interval is spread more uniformly over the weekend compared to the weekdays.