# Reproducible Research in R: Week 2 Assignment

syfq91

8/5/2020

### Loading and preprocessing the data

Show any code that is needed to:

1. Load the data (i.e. `read.csv()`).
2. Process/transform the data (if necessary) into a format suitable for your analysis.

```r
knitr::opts_chunk$set(echo = TRUE)

library(chron)
```

```
## NOTE: The default cutoff when expanding a 2-digit year
## to a 4-digit year will change from 30 to 69 by Aug 2020
## (as for Date and POSIXct in base R.)
```

```r
# Read in the data file.
df <- read.csv("activity.csv")

# Convert 'date' column to datetime variable.
df$date <- as.Date(df$date,format="%Y-%m-%d")

# Convert 'interval' to a time variable
# df$interval <- times(sub("(.{2})", "\\1:", sprintf("%04d:00", df$interval)))
```

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

For this part of the assignment, we can ignore the missing values in the dataset.

1. Calculate the total number of steps taken per day
2. If you do not understand the difference between a histogram and a barplot, research the difference between them. Make a histogram of the total number of steps taken each day
3. Calculate and report the mean and median of the total number of steps taken per day

```r
# Drop rows with NA in 'steps' column
df1 <- df[complete.cases(df),]

# Calculate total steps per day
total_steps_per_day <- aggregate(df1$steps, by=list(df1$date), sum)

# Generate histogram
png("Histogram of Total Steps Per Day.png")
hist(total_steps_per_day$x,breaks=20,col="blue",xlab="Total Steps Per Day",main="Histogram of Total Step
dev.off()
```

```
## pdf
##   2
```

```r
# Calculate and report the mean and median of the total number of steps taken per day

mean <- mean(total_steps_per_day$x)
median <- median(total_steps_per_day$x)
paste0("Mean Total Steps per Day: ",round(mean,2))
```

```
## [1] "Mean Total Steps per Day: 10766.19"
```

```r
paste0("Median Total Steps per Day: ",round(median,2))
```

```
## [1] "Median Total Steps per Day: 10765"
```

### What is the average daily activity pattern?

1. Make a time series plot (i.e. `type = "l"`) of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)
2. Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

```r
library(ggplot2)

# Calculate mean number of steps for each time period
mean_steps <- aggregate(df1$steps, by=list(df1$interval), mean)
names(mean_steps) <- c("time","steps")

# Plot the line graph
png("Mean Number of Steps by Time of Day.png")
ggplot(data=mean_steps, aes(x = time, y = steps))+geom_line()+xlab("Time of Day (HHMM)")+ggtitle("Mean
dev.off()
```

```
## pdf
##   2
```

```r
# Determine time interval when mean number of steps is maximum
mean_steps[which.max(mean_steps$steps), ]$time
```

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

### Imputing missing values

Note that there are a number of days/intervals where there are missing values (coded as `NA`). The presence of missing days may introduce bias into some calculations or summaries of the data.

1. Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with `NA`s).

2. Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to be sophisticated. For example, you could use the mean/median for that day, or the mean for that 5-minute interval, etc.

3. Create a new dataset that is equal to the original dataset but with the missing data filled in.

4. Make a histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day. Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

```r
# Calculate the number of rows with NAs
sum(!complete.cases(df))
```

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

```
library(plyr)
library(dplyr)
```

```
##
## Attaching package: ’dplyr’

## The following objects are masked from ’package:plyr’:
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

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

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

```
# Get rows in df where NAs are present
na_rows <- df[!complete.cases(df),]

# Rename columns
names(na_rows) <- c("steps","date","time")

# Impute mean number of steps for each time interval
na_rows <- merge(na_rows,mean_steps, by=c("time"))

# 'df1' is the original dataset with the NA rows deleted. We need to combine 'na_rows' with 'df1'. Howe

na_rows = subset(na_rows, select = -c(steps.x) )
names(na_rows) <- c("interval","date","steps")

# Combine 'df1' and 'na_rows'
df1 <- union(df1,na_rows)

# Sort 'df1' by date and then by interval
df1 <- df1 %>% arrange(date, interval)

# Calculate total steps per day
total_steps_per_day <- aggregate(df1$steps, by=list(df1$date), sum)

# Generate histogram

png("Histogram of Total Steps Per Day with Imputed Values.png")
hist(total_steps_per_day$x,breaks=20,col="blue",xlab="Total Steps Per Day",main="Histogram of Total Step
dev.off()
```

```
## pdf
##   2
```

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

mean <- mean(total_steps_per_day$x)
median <- median(total_steps_per_day$x)
paste0("Mean Total Steps per Day: ",round(mean,2))
```

```
## [1] "Mean Total Steps per Day: 10766.19"
```

```
paste0("Median Total Steps per Day: ",round(median,2))
```

```
## [1] "Median Total Steps per Day: 10766.19"
```

As we can see, imputing the mean value of the total number of steps at each time interval for missing values has very little effect on the histogram or the overall mean and median values.

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

For this part, the `weekdays()` function may be of some help here. Use the dataset with the filled-in missing values for this part.

1. Create a new factor variable in the dataset with two levels – "weekday" and "weekend" indicating whether a given date is a weekday or weekend day.
2. Make a panel plot containing a time series plot (i.e. `type = "l"`) of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

```
# Determine day of week for each date
df1$day <- weekdays(df1$date)

# Recode day of week to 'Weekday' or 'Weekend'
df1$weekday_or_weekend[df1$day %in% c("Monday","Tuesday","Wednesday","Thursday","Friday")] <- "Weekday"
df1$weekday_or_weekend[df1$day %in% c("Saturday","Sunday")] <- "Weekend"

# Calculate mean number of steps for each time interval on Weekdays and Weekends
mean_steps <- aggregate(df1$steps, by=list(df1$weekday_or_weekend,df1$interval), mean)
names(mean_steps) <- c("Weekday_Weekend","time","steps")

# Sort 'mean_steps' by weekday/weekend and then by time
mean_steps <- mean_steps %>% arrange(Weekday_Weekend, time)

# The 'Weekday_Weekend' column must be converted to a factor variable for plotting
mean_steps$Weekday_Weekend <- as.factor(mean_steps$Weekday_Weekend)

# Create plots
png("Mean Number of Steps by Time of Day Weekday Weekend.png")
ggplot(mean_steps, aes(x=time, y=steps))+geom_line()+facet_wrap(~Weekday_Weekend,nrow=2)+xlab("Time of 
dev.off()
```

```
## pdf
##   2
```