

Course Project 1 - Reproducible Research

1. Code for reading in the dataset and/or processing the data

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.0 --
## v ggplot2 3.3.2      v purrr  0.3.4
## v tibble  3.0.3      v dplyr  1.0.2
## v tidyr   1.1.2      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.5.0

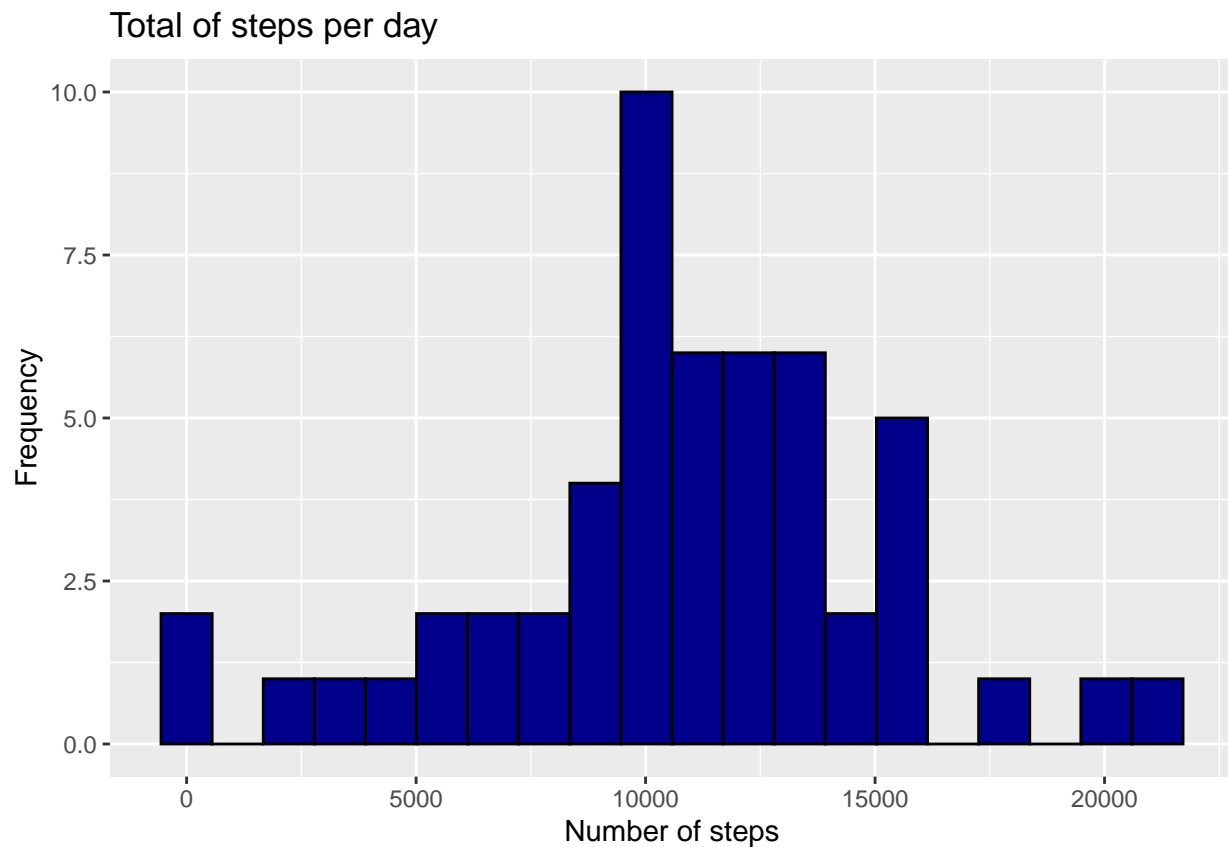
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

activity_url <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
temp <- tempfile()
download.file(activity_url, temp)

unzip(temp, "activity.csv")
mydata <- read.csv("activity.csv")
unlink(temp)
activity_omit <- na.omit(mydata)
total_ac_omit <- aggregate(steps ~ date, activity_omit, sum)
```

2. Histogram of the total number of steps taken each day

```
ggplot(data = total_ac_omit, aes(x=steps)) +
  geom_histogram(bins = 20, fill = "darkblue", color = "black") +
  ggtitle("Total of steps per day") +
  ylab("Frequency") + xlab("Number of steps")
```



3. Mean and median number of steps taken each day

```
steps_mean <- mean(total_ac_omit$steps)
steps_median <- median(total_ac_omit$steps)
```

```
print(steps_mean)
```

```
## [1] 10766.19
```

```
print(steps_median)
```

```
## [1] 10765
```

The mean is `steps_mean[1]` and the median is `steps_median[1]`

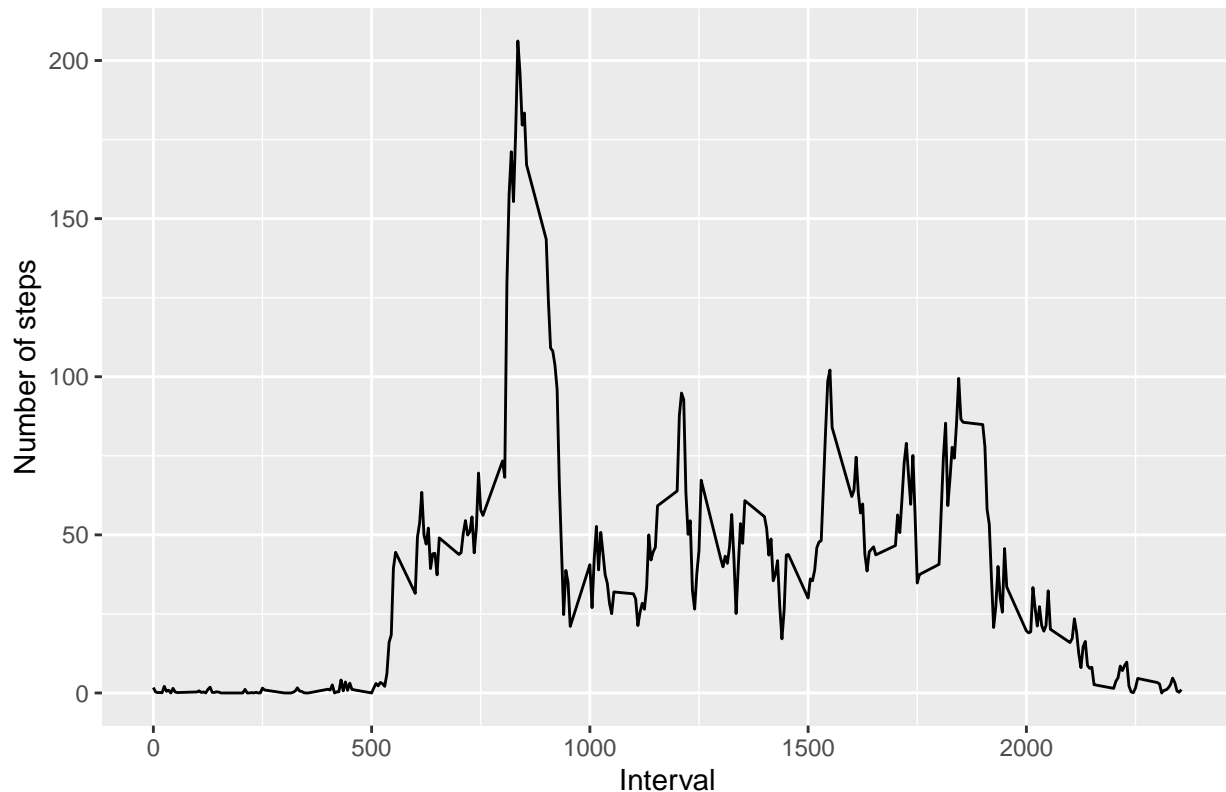
4. Time series plot of the average number of steps taken

Prepare data

```
total_ac_average <- aggregate(steps ~ interval, activity_omit, mean)
```

```
ggplot(total_ac_average, aes(interval, steps)) +
  geom_line() +
  ggtitle("Time series plot of the average number of steps taken") +
  xlab("Interval") +
  ylab("Number of steps")
```

Time series plot of the average number of steps taken



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

```
max_number_steps <- which.max(total_ac_average$steps)
print(total_ac_average[max_number_steps,])
```

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

6. Code to describe and show a strategy for imputing missing data

```
fill <- function(steps, interval) {
  fill_ac <- NA
  if (!is.na(steps))
    fill_a <- c(steps)
  else
    fill_a <- (total_ac_average[total_ac_average$interval==interval, "steps"])
  return(fill_a)
}
```

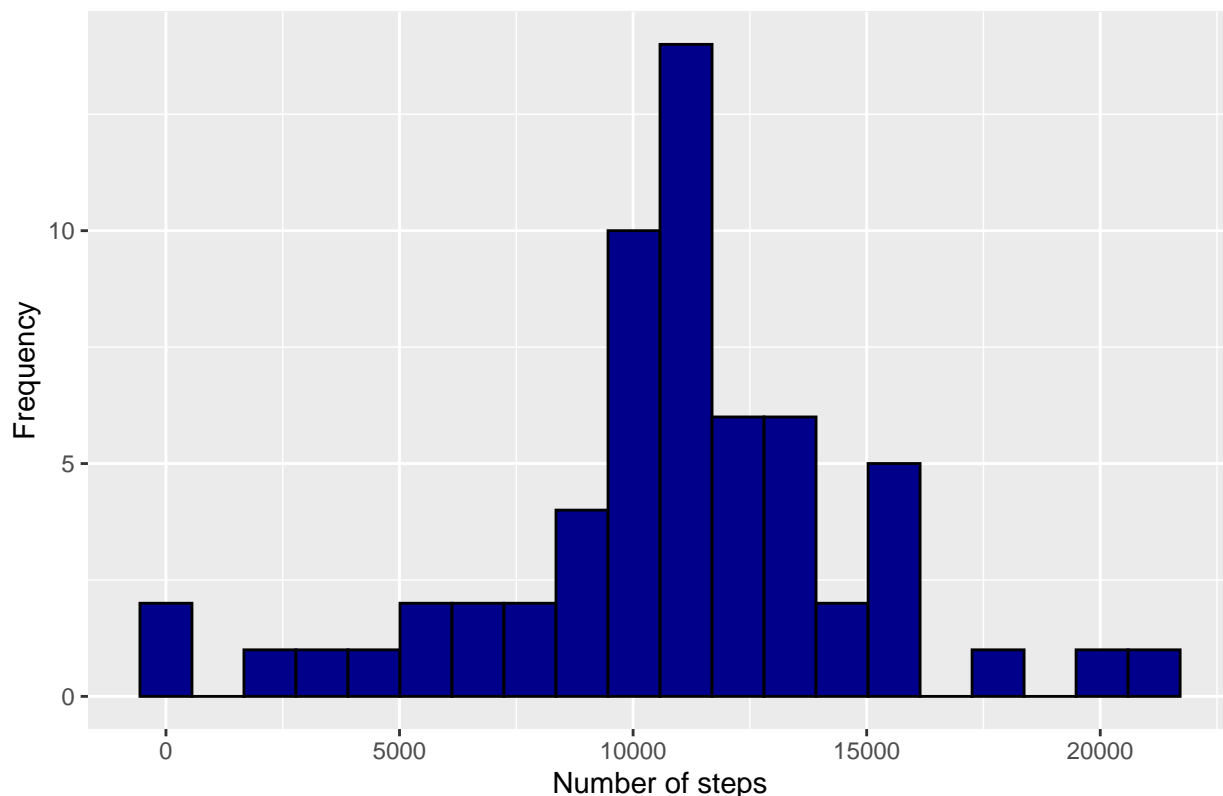
```
new_activity <- mydata
new_activity$steps <- mapapply(fill, new_activity$steps, new_activity$interval)

total_ac_steps <- aggregate(steps ~ date, new_activity, sum)
```

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

```
ggplot(data = total_ac_steps, aes(x=steps)) +
  geom_histogram(bins = 20, fill = "darkblue", color = "black") +
  ggtitle("Total of steps per day") +
  ylab("Frequency") + xlab("Number of steps")
```

Total of steps per day



```
total_ac_steps_mean <- mean(total_ac_steps$steps)
total_ac_steps_median <- median(total_ac_steps$steps)
print(total_ac_steps_mean)
```

```
## [1] 10766.19
```

```
print(total_ac_steps_median)
```

```
## [1] 10766.19
```

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

```
week_activity <- new_activity
```

```
week_activity$date <- as.Date(strptime(week_activity$date, format="%Y-%m-%d"))
```

```
week_activity$day <- factor(ifelse(as.POSIXlt(week_activity$date)$wday %in% c(0,6), 'weekend', 'weekday'))
averaged_week_activity <- aggregate(steps ~ interval + day, data=week_activity, mean)
```

```
ggplot(averaged_week_activity, aes(interval, steps)) +
  geom_line() +
  facet_grid(day ~ .) +
  xlab("5-minute interval") +
  ylab("avarage number of steps")
```

