Reproducible Research: Assignment 1

Meskerem

2023-10-14

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.

```
### Code for reading in the dataset and/or processing the data
### Loading and preprocessing the data
# Imputing missing values

activity <- read.csv("activity.csv")
activity$date <- as.Date(activity$date, format = "%Y-%m-%d")

dim(activity)
```

```
## [1] 17568     3
```

```
names(activity)
```

```
## [1] "steps"    "date"     "interval"
```

```
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
```

```
str(activity)
```

```
## '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 ...
```

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

sum(is.na(activity$steps))/dim(activity)[[1]]
```

```
## [1] 0.1311475
```

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.2.3
```

```
##
## Attaching package: 'lubridate'
```

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

```
activity$date <- ymd(activity$date)
length(unique(activity$date))
```

```
## [1] 61
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
## What is the average daily activity pattern?

AM <- data.frame(tapply(activity$steps, activity$date, sum, na.rm = TRUE))
AM$date <- rownames(AM)
rownames(AM) <- NULL
names(AM)[[1]] <- "Total Steps"
png("plot1.png")
ggplot(AM, aes(y = AM$`Total Steps`, x = AM$date)) + geom_bar(stat = "identity") + ylab("Total Steps") +
```

```
## Warning: Use of 'AM$date' is discouraged.
## i Use 'date' instead.
```

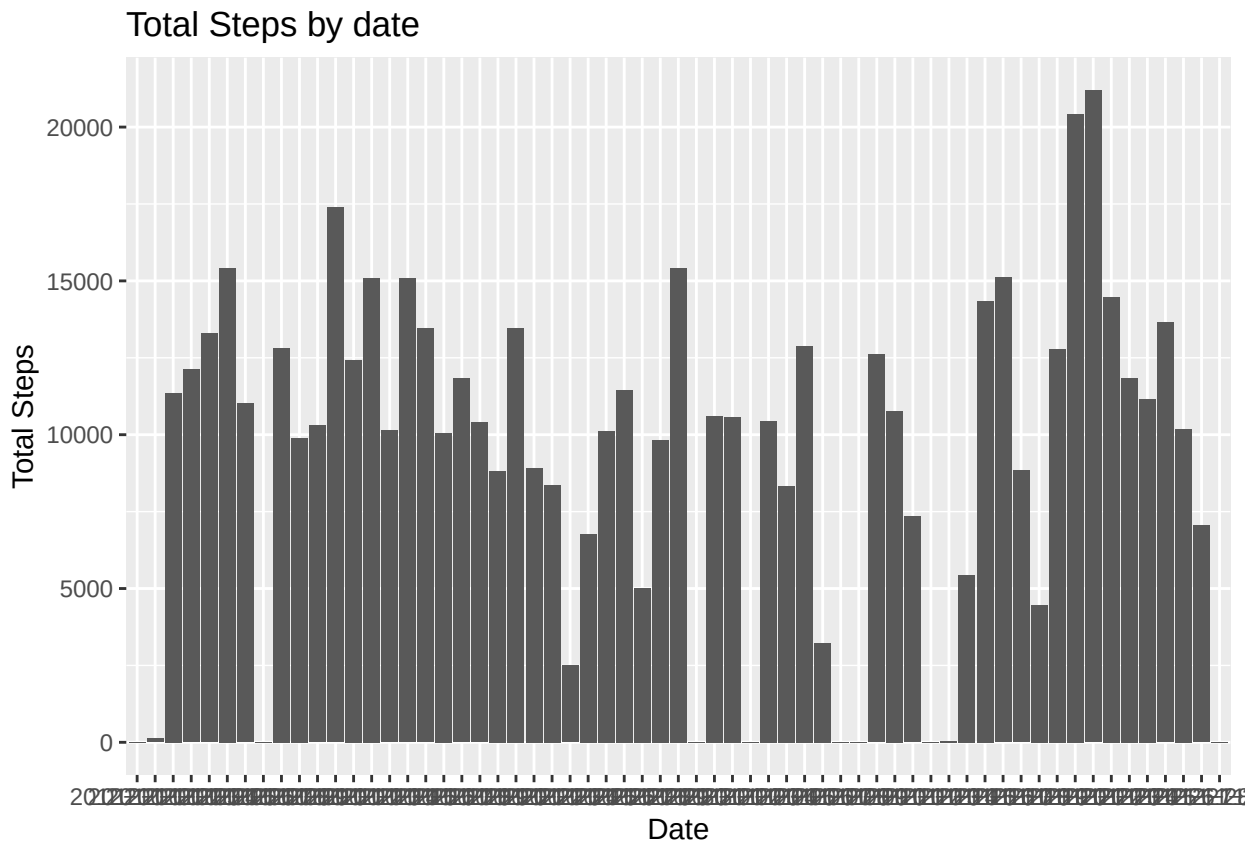
```
## Warning: Use of '' AM$'Total Steps' '' is discouraged.
## i Use 'Total Steps' instead.
```

```
dev.off()
```

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

```
ggplot(AM, aes(y = AM$`Total Steps`, x = AM$date)) + geom_bar(stat = "identity") + ylab("Total Steps") +
```

```
## Warning: Use of ‘AM$date‘ is discouraged.
## i Use ‘date‘ instead.
## Use of ‘‘ AM$‘Total Steps‘ ‘‘ is discouraged.
## i Use ‘Total Steps‘ instead.
```

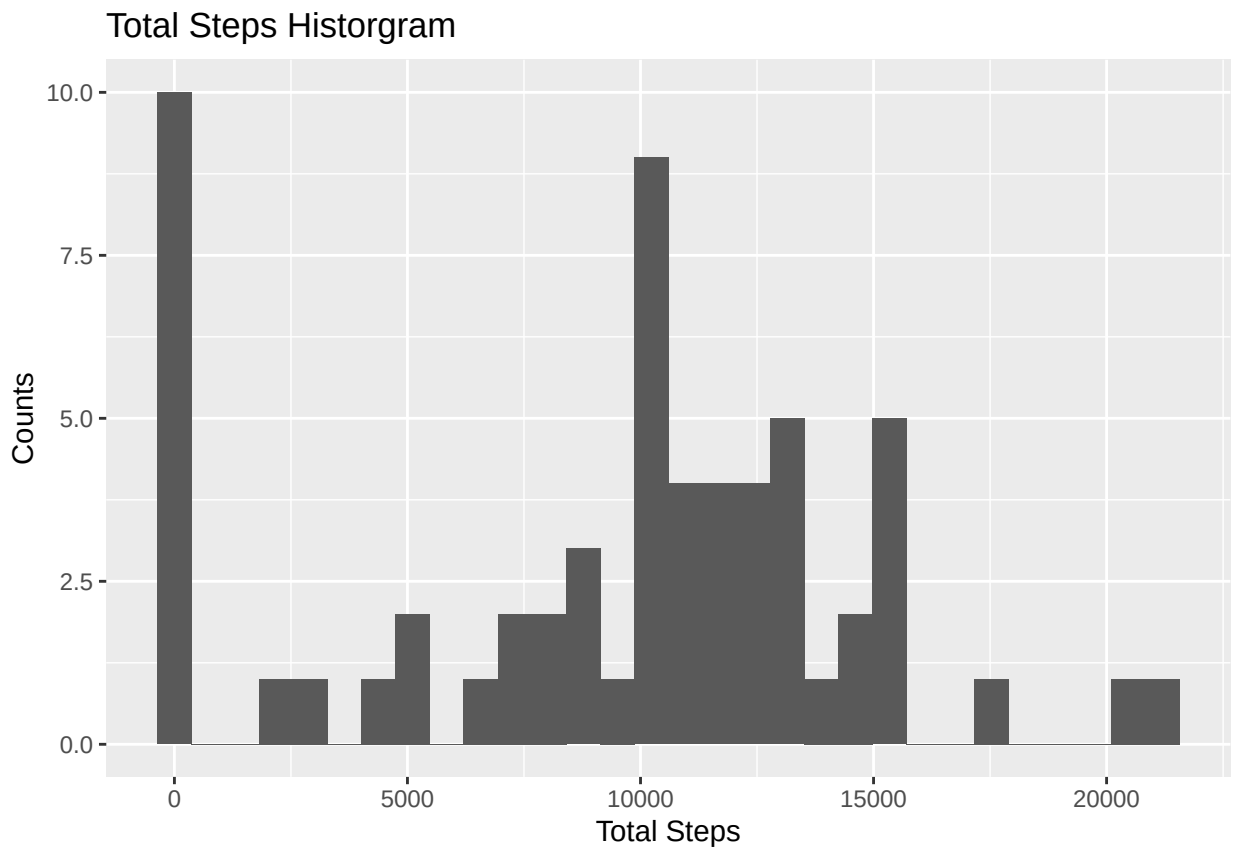



```
# Make a histogram of the total number of steps taken each day and Calculate and report the mean and me

qplot(AM$`Total Steps`, geom = "histogram", xlab = "Total Steps", ylab = "Counts", main = "Total Steps
```

```
## Warning: ‘qplot()‘ was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```




```
png("plot2.png")
qplot(AM$`Total Steps`, geom = "histogram", xlab = "Total Steps", ylab = "Counts", main = "Total Steps H
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
dev.off()
```

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

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

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

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

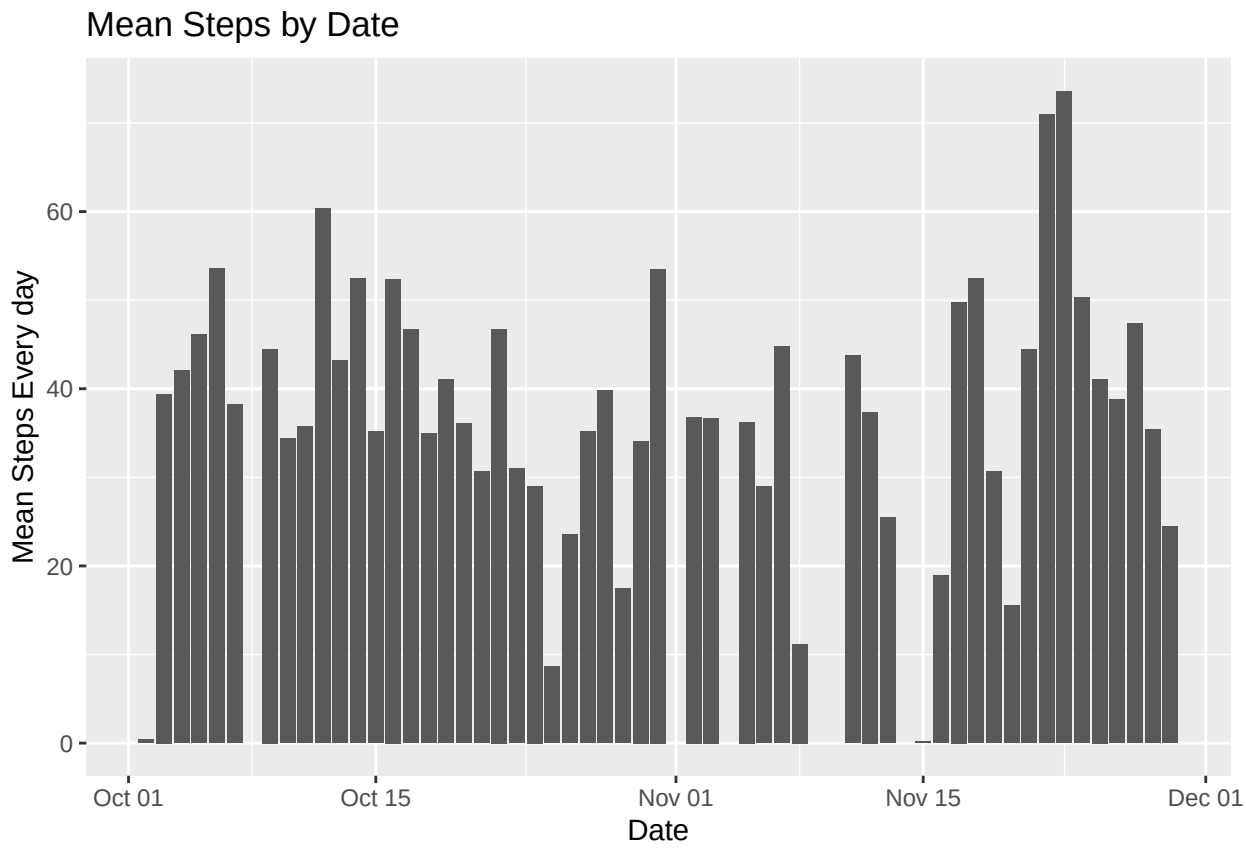
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
AM2 <- data.frame(round(tapply(activity$steps, activity$date, mean, na.rm = TRUE), 2))
AM2$date <- rownames(AM2)
rownames(AM2) <- NULL
names(AM2)[[1]] <- "Mean Steps"
temp <- activity %>% select(date, steps) %>% group_by(date) %>% summarise(median(steps))
names(temp)[[2]] <- "Median Steps"
AM2$median <- temp$`Median Steps`
AM2 <- AM2 %>% select(date, `Mean Steps`, median)
AM3 <- AM2
AM3$date <- as.Date(AM3$date, format = "%Y-%m-%d")
ggplot(AM3, aes(x = AM3$date, y = AM3$`Mean Steps`)) + geom_bar(stat = "identity") + scale_x_date() + yl
```

```
## Warning: Use of ‘AM3$date‘ is discouraged.
## i Use ‘date‘ instead.

## Warning: Use of ‘‘ AM3$‘Mean Steps‘ ‘‘ is discouraged.
## i Use ‘Mean Steps‘ instead.

## Warning: Removed 8 rows containing missing values (‘position_stack()‘).
```




```
png("plot3.png")
ggplot(AM3, aes(x = AM3$date, y = AM3$`Mean Steps`)) + geom_bar(stat = "identity") + scale_x_date() + yl
```

```
## Warning: Use of ‘AM3$date‘ is discouraged.
## i Use ‘date‘ instead.
```

```
## Warning: Use of `` AM3$`Mean Steps` `` is discouraged.
## i Use `Mean Steps` instead.
```

```
## Warning: Removed 8 rows containing missing values (`position_stack()`).
```

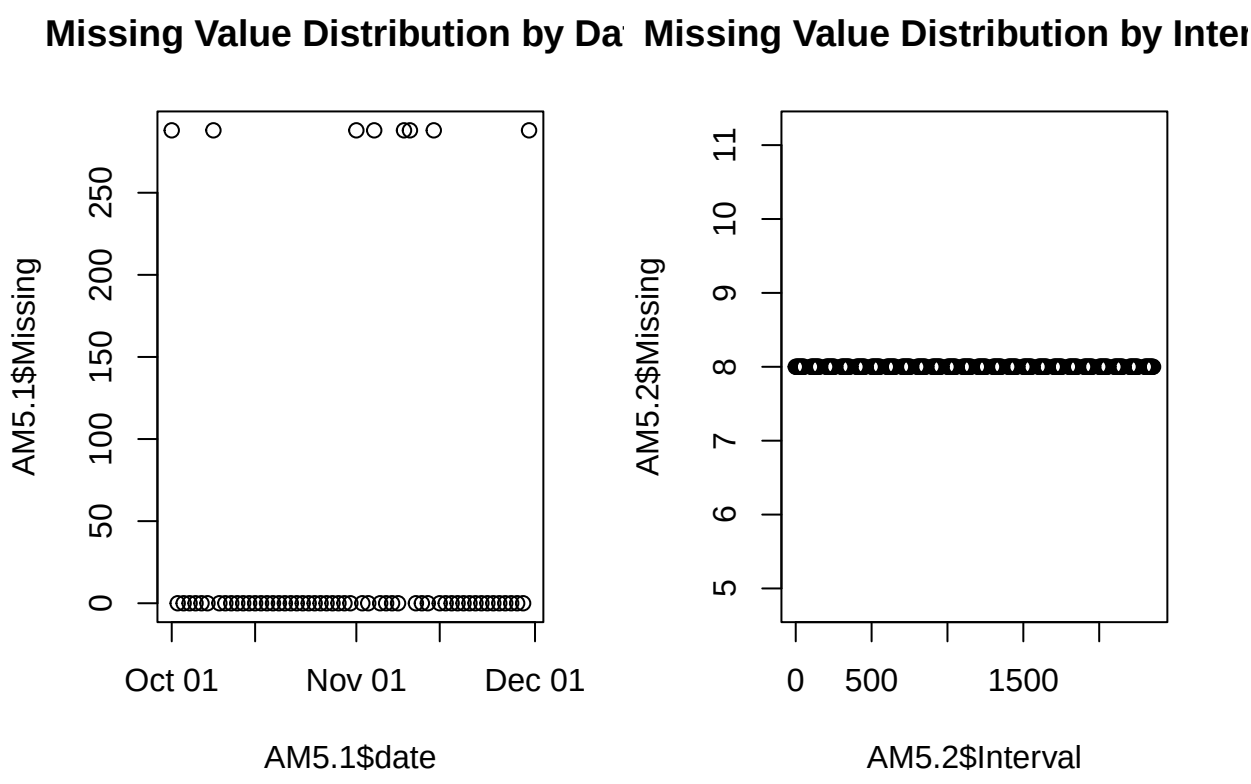
```
dev.off()
```

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

```
activity$interval <- factor(activity$interval)
AM4 <- aggregate(data = activity, steps ~ date + interval, FUN = "mean")
AM4 <- aggregate(data = AM4, steps ~ interval, FUN = "max")
AM5 <- activity
AM5$Missing <- is.na(AM5$steps)
AM5 <- aggregate(data = AM5, Missing ~ date + interval, FUN = "sum")
AM5.1 <- data.frame(tapply(AM5$Missing, AM5$date, sum))
AM5.1$date <- rownames(AM5.1)
rownames(AM5.1) <- NULL
names(AM5.1) <- c("Missing", "date")
AM5.1$date <- as.Date(AM5.1$date, format = "%Y-%m-%d")

AM5.2 <- data.frame(tapply(AM5$Missing, AM5$interval, sum))
AM5.2$date <- rownames(AM5.2)
rownames(AM5.2) <- NULL
names(AM5.2) <- c("Missing", "Interval")

par(mfrow = c(1, 2))
plot(y = AM5.1$Missing, x = AM5.1$date, main = "Missing Value Distribution by Date")
plot(y = AM5.2$Missing, x = AM5.2$Interval, main = "Missing Value Distribution by Interval")
```




```
table(activity$date)
```

```
##
## 2012-10-01 2012-10-02 2012-10-03 2012-10-04 2012-10-05 2012-10-06 2012-10-07
##        288        288        288        288        288        288        288
## 2012-10-08 2012-10-09 2012-10-10 2012-10-11 2012-10-12 2012-10-13 2012-10-14
##        288        288        288        288        288        288        288
## 2012-10-15 2012-10-16 2012-10-17 2012-10-18 2012-10-19 2012-10-20 2012-10-21
##        288        288        288        288        288        288        288
## 2012-10-22 2012-10-23 2012-10-24 2012-10-25 2012-10-26 2012-10-27 2012-10-28
##        288        288        288        288        288        288        288
## 2012-10-29 2012-10-30 2012-10-31 2012-11-01 2012-11-02 2012-11-03 2012-11-04
##        288        288        288        288        288        288        288
## 2012-11-05 2012-11-06 2012-11-07 2012-11-08 2012-11-09 2012-11-10 2012-11-11
##        288        288        288        288        288        288        288
## 2012-11-12 2012-11-13 2012-11-14 2012-11-15 2012-11-16 2012-11-17 2012-11-18
##        288        288        288        288        288        288        288
## 2012-11-19 2012-11-20 2012-11-21 2012-11-22 2012-11-23 2012-11-24 2012-11-25
##        288        288        288        288        288        288        288
## 2012-11-26 2012-11-27 2012-11-28 2012-11-29 2012-11-30
##        288        288        288        288        288
```

```
library(lubridate)
AM5.3 <- as.data.frame(AM5.1) %>% select(date, Missing) %>% arrange(desc(Missing))
AM5.3 <- AM5.3[which(AM5.3$Missing != 0),]
```

```
AM5.3$Weekday <- wday(AM5.3$date, label = TRUE)
AM5.4 <- activity
AM5.4$weekday <- wday(AM5.4$date, label = TRUE)
# What is mean total number of steps taken per day?
AM5.5 <- aggregate(data = AM5.4, steps ~ interval + weekday, FUN = "mean", na.rm = TRUE)
AM5.6 <- merge(x = AM5.4, y = AM5.5, by.x = c("interval", "weekday"), by.y = c("interval", "weekday"), a
AM5.6$Steps.Updated <- 0
for (i in 1:dim(AM5.6)[[1]]) {
  if (is.na(AM5.6[i, 3])) {
    AM5.6[i, 6] = AM5.6[i, 5]
  } else {
    AM5.6[i, 6] = AM5.6[i, 3]
  }
}

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

AM5.6 <- AM5.6 %>% select(date, weekday, interval, Steps.Updated)
names(AM5.6)[[4]] <- "Steps"
png("plot4.png")
qplot(AM5.6$Steps, geom = "histogram", main = "Total steps taken histogram post imputation", xlab = "Ste
```

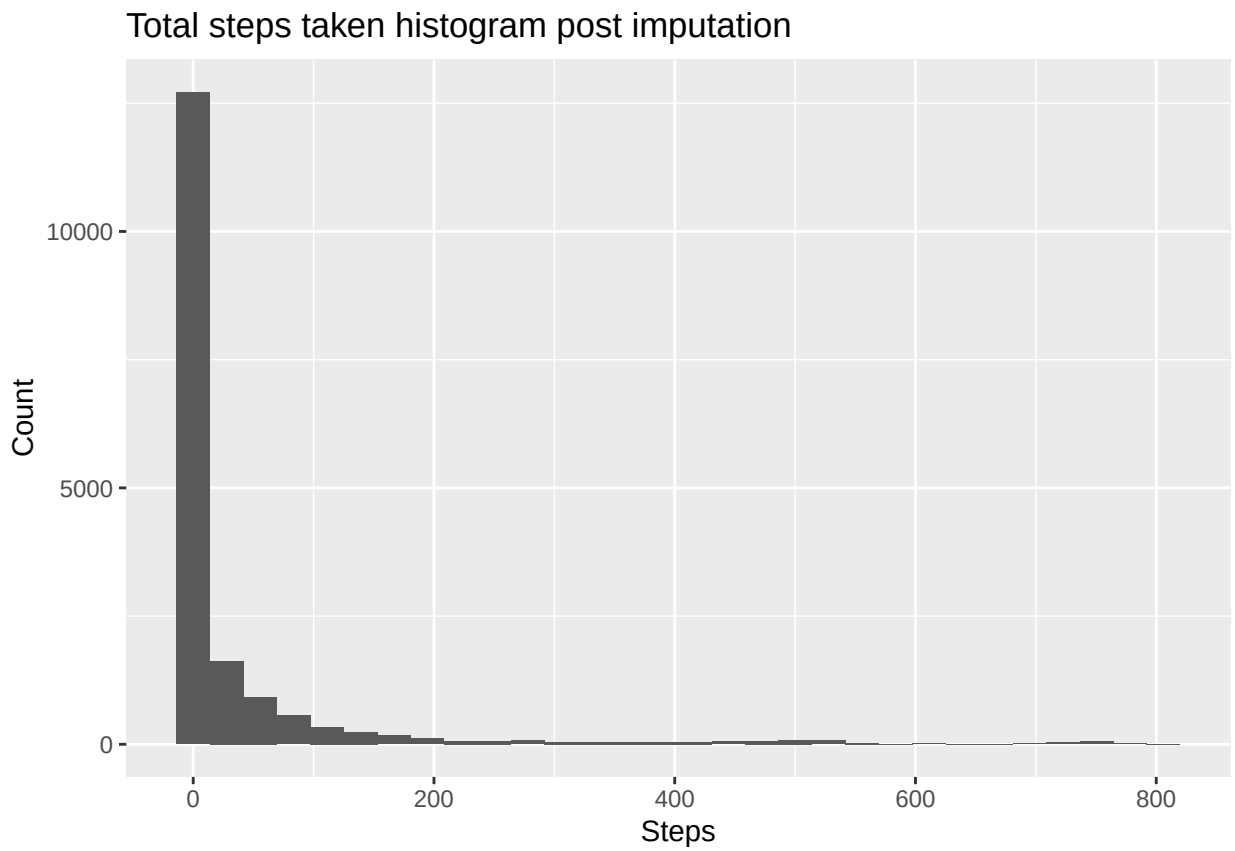
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
dev.off()
```

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

```
qplot(AM5.6$Steps, geom = "histogram", main = "Total steps taken histogram post imputation", xlab = "Ste
```

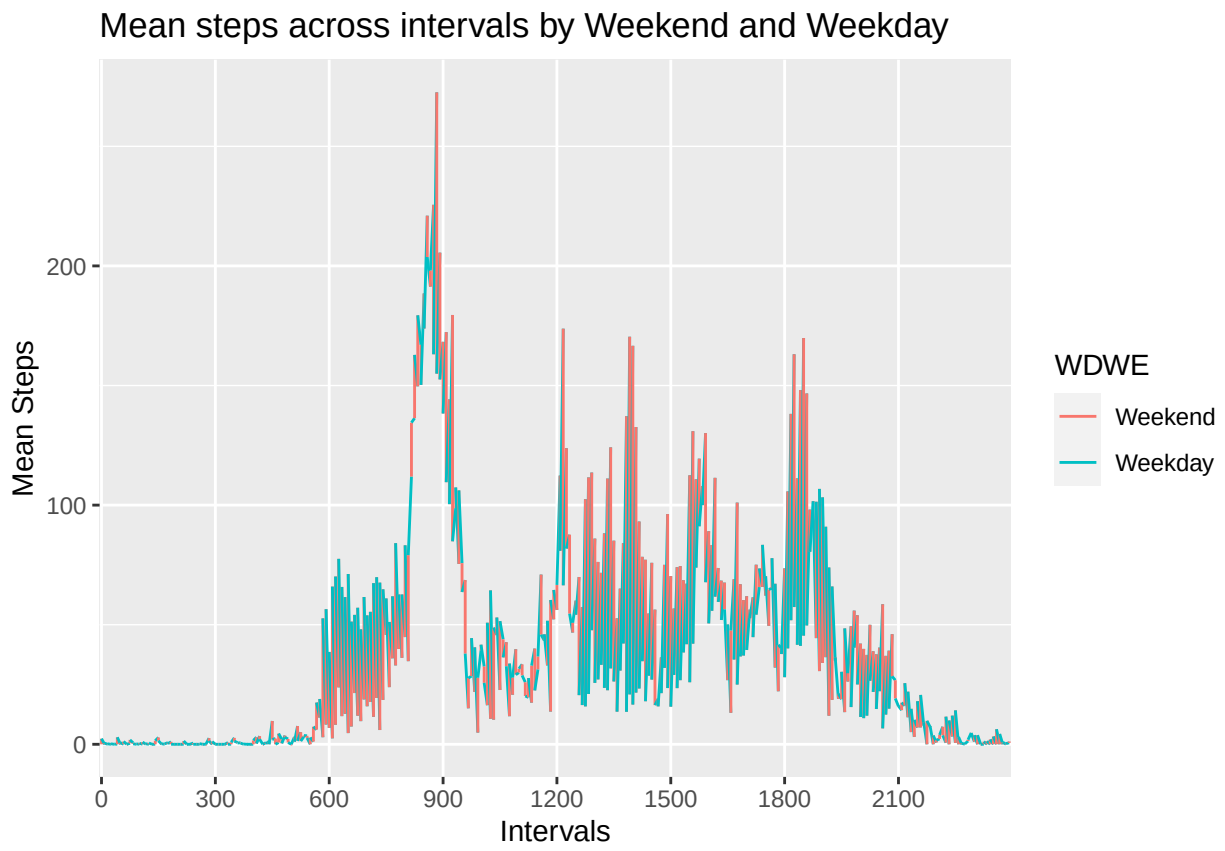
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
AM6 <- AM5.6
levels(AM6$weekday) <- c(1, 2, 3, 4, 5, 6, 7)
AM6$WDWE <- AM6$weekday %in% c(1, 2, 3, 4, 5)
AM6.1 <- aggregate(data = AM6, Steps ~ interval + WDWE, mean, na.rm = TRUE)
AM6.1$WDWE <- as.factor(AM6.1$WDWE)
levels(AM6.1$WDWE) <- c("Weekend", "Weekday")
png("plot5.png")
ggplot(data = AM6.1, aes(y = Steps, x = interval, group = 1, color = WDWE)) + geom_line() + scale_x_dis
dev.off()
```

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

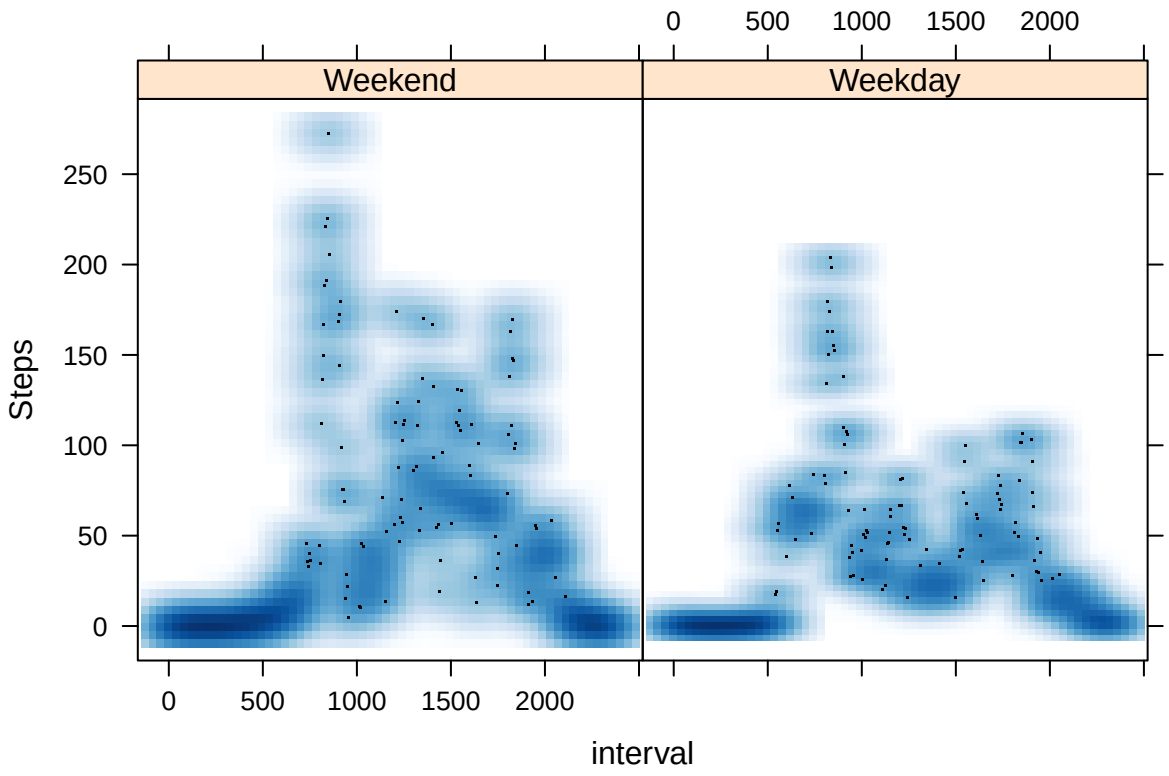
```
ggplot(data = AM6.1, aes(y = Steps, x = interval, group = 1, color = WDWE)) + geom_line() + scale_x_dis
```




```
# Make a panel plot containing a time series plot

AM6.1$interval <- as.numeric(as.character(AM6.1$interval))
library(lattice)
xyplot(data = AM6.1, Steps ~ interval | WDWE, grid = TRUE, type = c("p", "smooth"), lwd = 4, panel = pa
```

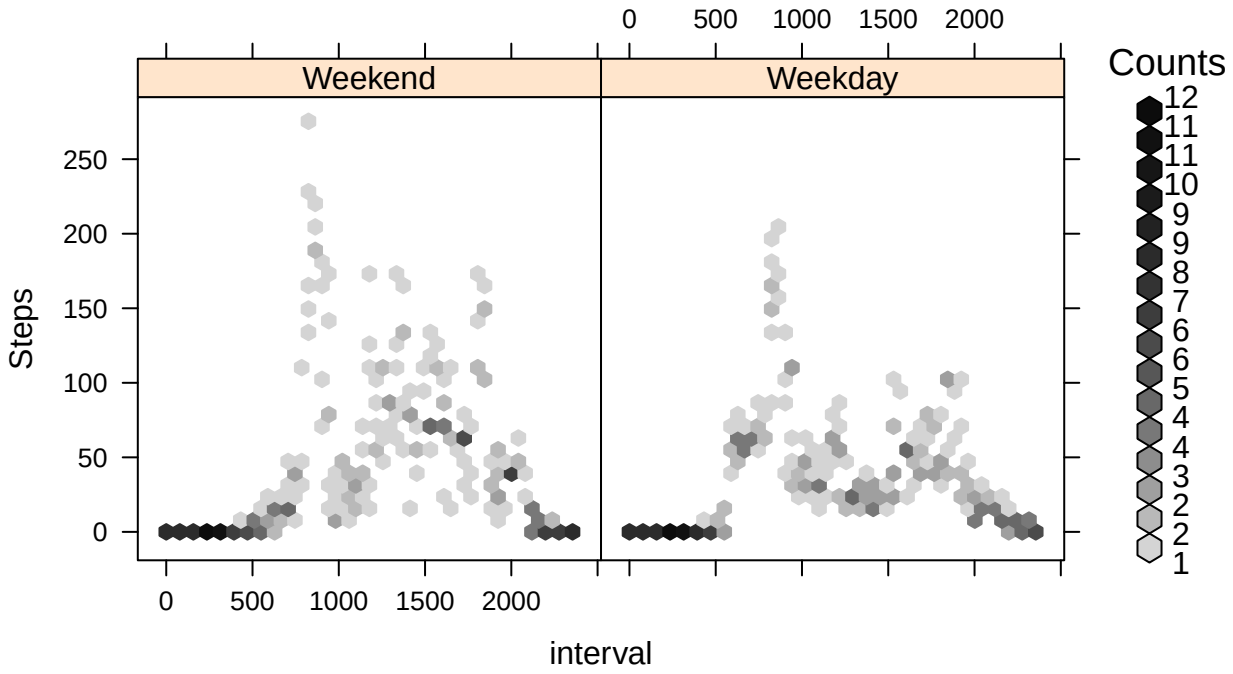
```
## (loaded the KernSmooth namespace)
```




```
library(hexbin)
```

```
## Warning: package 'hexbin' was built under R version 4.2.3
```

```
hexbinplot(data = AM6.1, Steps ~ interval | WDWE, aspect = 1, bins = 50)
```




```
png("plot6.png")
xyplot(data = AM6.1, Steps ~ interval | WDWE, grid = TRUE, type = c("p", "smooth"), lwd = 4, panel = pa
dev.off()
```

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

```
png("plot7.png")
hexbinplot(data = AM6.1, Steps ~ interval | WDWE, aspect = 1, bins = 50)
dev.off()
```

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