

# PA1\_Template

Omkar

21/06/2020

```
unzip(zipfile="activity.zip")
```

```
## Warning in unzip(zipfile = "activity.zip"): error 1 in extracting from zip file
```

```
activity <- read.csv("activity.csv")
```

```
library(lubridate)
```

```
##
```

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

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##     date, intersect, setdiff, union
```

```
## Determine day of the week and categorize as weekend or weekday
```

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

```
activity$weekend <- as.factor(ifelse(weekdays(activity$date)=="Saturday" | weekdays(activity$date)=="Sunday", 1, 0))
```

```
activity$dayofweek <- as.factor(weekdays(activity$date))
```

```
##Generate histogram for total number of steps taken for each day
```

```
library(dplyr)
```

```
##
```

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

```
## The following objects are masked from 'package:lubridate':
```

```
##
```

```
##     intersect, setdiff, union
```

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

```
##
```

```
##     filter, lag
```

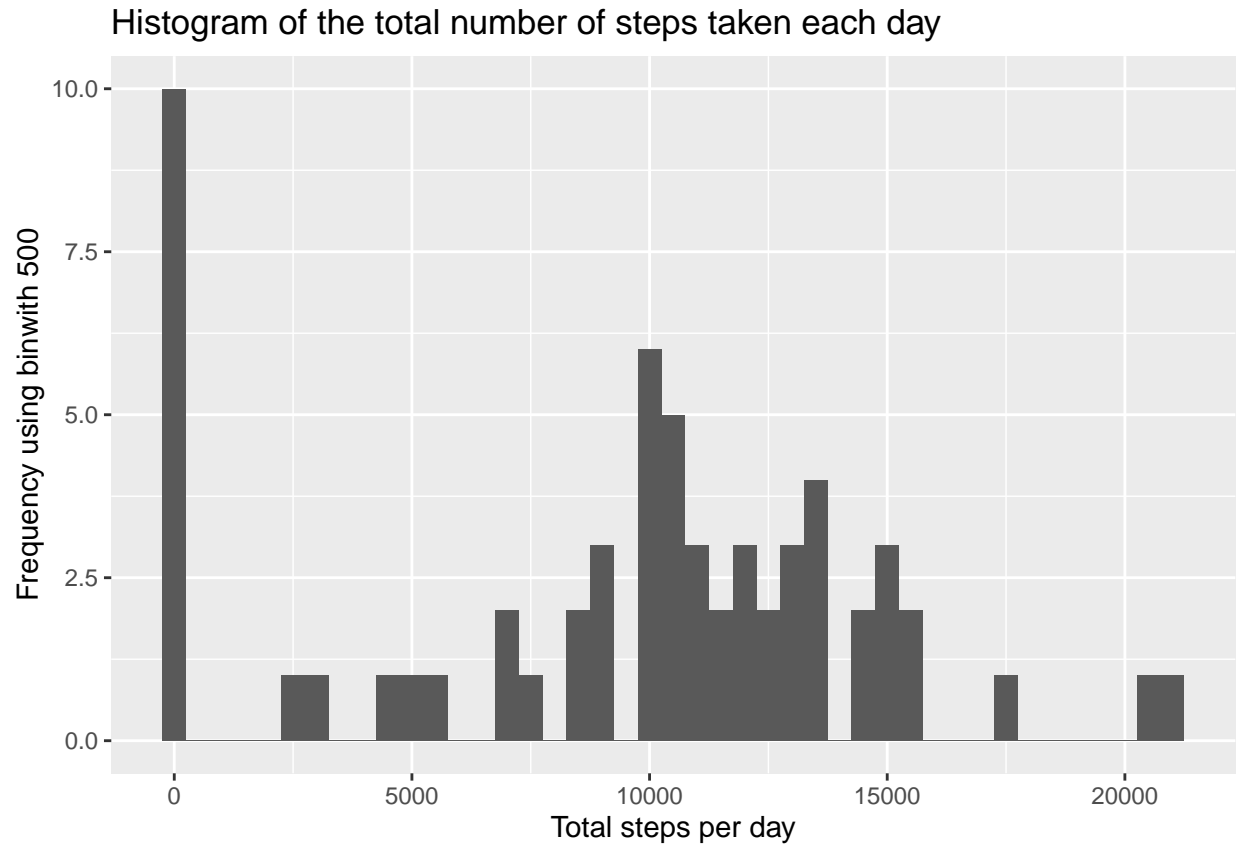
```
## The following objects are masked from 'package:base':
```

```
##
```

```
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
stepsByDay <- activity %>% group_by(date) %>% summarise(stepsperday = sum(steps,na.rm = TRUE))
qplot(stepsperday,data=stepsByDay,na.rm=TRUE,binwidth=500,xlab='Total steps per day', ylab='Frequency u
```



```
library(dplyr)
library(ggplot2)
```

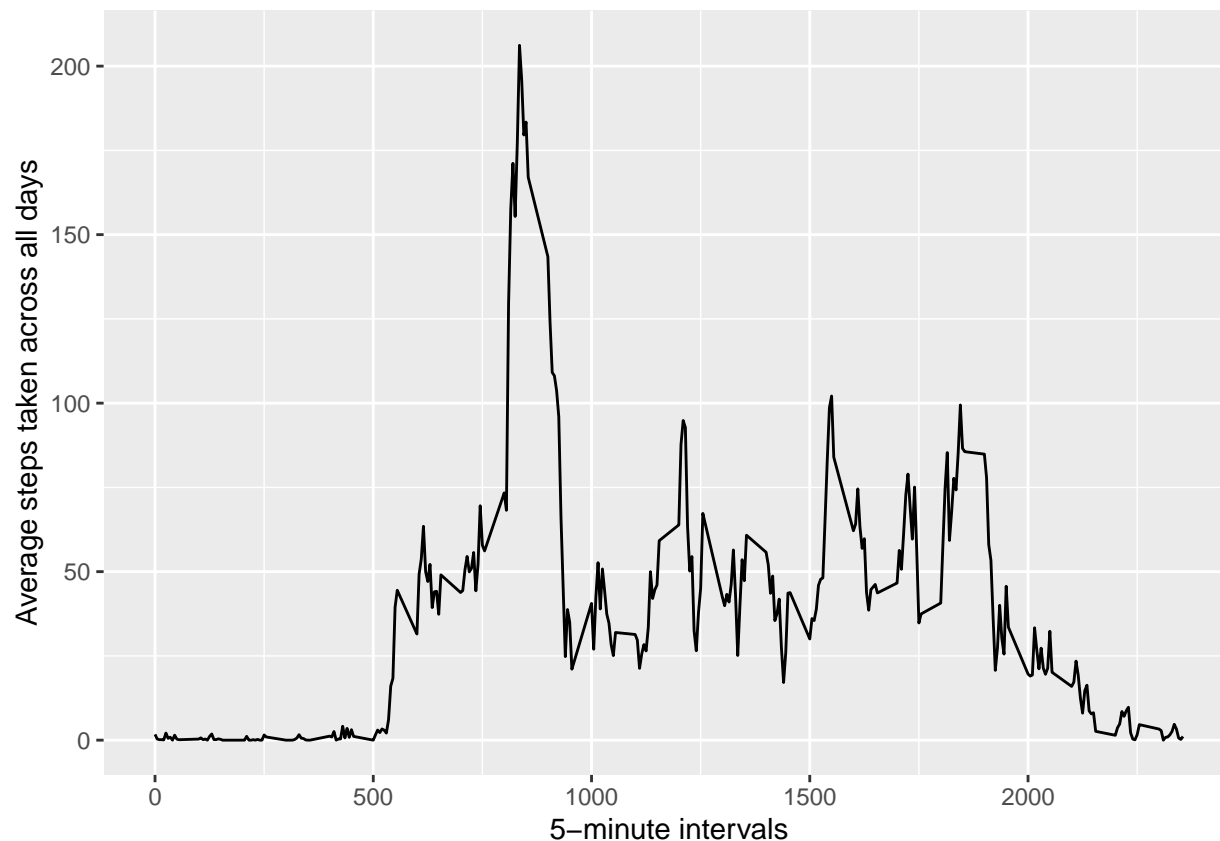
```
## Calculate mean and median steps taken for each day
```

```
meanstepsperday <- stepsByDay %>% summarise(average = mean(stepsperday,na.rm = TRUE),median=median(stepsperday))
meanstepsperday
```

```
## # A tibble: 1 x 2
##   average median
##   <dbl>   <int>
## 1  9354.  10395
```

```
## Plot average daily activity pattern
```

```
interval_average <- activity %>% group_by(interval) %>% summarise(average = mean(steps,na.rm = TRUE))
qplot(interval,average,data=interval_average,geom="line",xlab = "5-minute intervals",ylab = "Average steps per 5-minute interval")
```



```
## Calculate interval which has max number of steps on average
interval_average[which.max(interval_average$average),]
```

```
## # A tibble: 1 x 2
##   interval average
##   <int>   <dbl>
## 1     835    206.
```

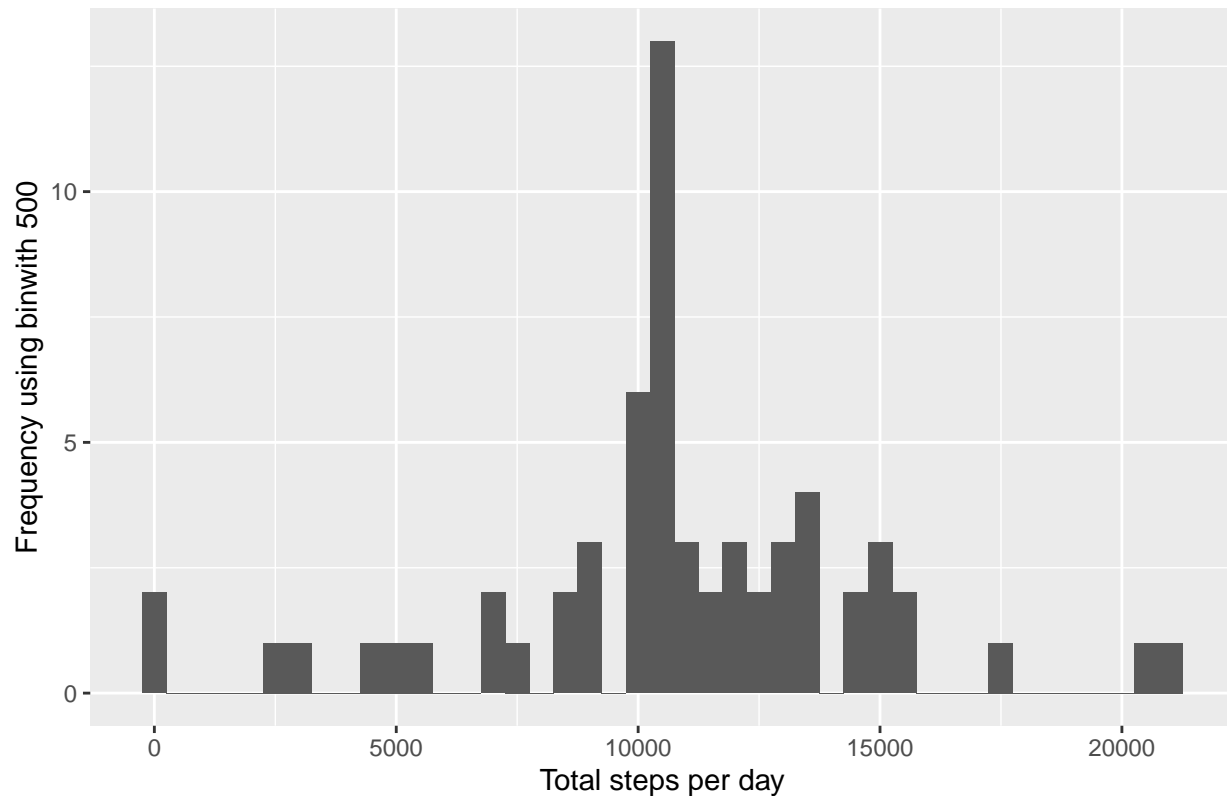
```
library(dplyr)
library(ggplot2)

## Handling missing values
## Dataset with no NAs
activity_no_NA <- activity[which(!is.na(activity$steps)),]
## Mean steps for each interval
interval_only <- activity_no_NA %>% group_by(interval) %>% summarise(average=mean(steps))
## Average to integer
interval_only$average <- as.integer(interval_only$average)
## Dataset where steps have NAs
activity_na <- activity[which(is.na(activity$steps)),]
## Replace NAs with average steps
activity_na$steps <- ifelse(activity_na$interval==interval_only$interval,interval_only$average)
## Binding data w/o NAs and data that had NAs
activity_impute <- rbind(activity_no_NA,activity_na)
## Calculate number of missing values
nrow(activity_na)
```

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

```
## Plot steps taken per day after missing values handled
stepsByDay_impute <- activity_impute %>% group_by(date) %>% summarise(stepsperday = sum(steps))
qplot(stepsperday, data=stepsByDay_impute, na.rm=TRUE, binwidth=500, xlab='Total steps per day', ylab='Frequency using binwidth 500')
```

Histogram of the total number of steps taken each day



```
## Mean and median after missing handling
totalstepsperday_impute <- activity_impute %>% group_by(date) %>% summarise(stepsperday = sum(steps))
mean_n_median <- totalstepsperday_impute %>% summarise(average=mean(stepsperday), median=median(stepsperday))
mean_n_median
```

```
## # A tibble: 1 x 2
##   average median
##   <dbl> <int>
## 1  10750.  10641
```

```
## Comparing patterns for weekdays and weekends
meansteps <- activity_impute %>% group_by(interval, weekend) %>% summarise(average = mean(steps))
qplot(interval, average, data=meansteps, geom="line", facets=weekend~., xlab="5-minute interval", ylab="average steps")
```

Average steps pattern between Weekday and Weekend

