Reproducible Research

## Data Wrangling:

windowsFont("Lucida Console")

## [1] "Lucida Console"

### Session Information

sessionInfo()

## R version 3.6.0 (2019-04-26)  
## Platform: x86\_64-w64-mingw32/x64 (64-bit)  
## Running under: Windows 10 x64 (build 17134)  
##   
## Matrix products: default  
##   
## locale:  
## [1] LC\_COLLATE=English\_United Kingdom.1252   
## [2] LC\_CTYPE=English\_United Kingdom.1252   
## [3] LC\_MONETARY=English\_United Kingdom.1252  
## [4] LC\_NUMERIC=C   
## [5] LC\_TIME=English\_United Kingdom.1252   
##   
## attached base packages:  
## [1] stats graphics grDevices utils datasets methods base   
##   
## loaded via a namespace (and not attached):  
## [1] compiler\_3.6.0 magrittr\_1.5 tools\_3.6.0 htmltools\_0.3.6  
## [5] yaml\_2.2.0 Rcpp\_1.0.1 stringi\_1.4.3 rmarkdown\_1.13   
## [9] knitr\_1.23 stringr\_1.4.0 xfun\_0.7 digest\_0.6.19   
## [13] evaluate\_0.13

### Loading Essential Packages

library(dplyr)

##   
## 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

library(ggplot2)

## Registered S3 methods overwritten by 'ggplot2':  
## method from   
## [.quosures rlang  
## c.quosures rlang  
## print.quosures rlang

### The Dataset

activity = read.csv("activity.csv")

Here are the summary statistics of the dataset

summary(activity)

## steps date interval   
## Min. : 0.00 2012-10-01: 288 Min. : 0.0   
## 1st Qu.: 0.00 2012-10-02: 288 1st Qu.: 588.8   
## Median : 0.00 2012-10-03: 288 Median :1177.5   
## Mean : 37.38 2012-10-04: 288 Mean :1177.5   
## 3rd Qu.: 12.00 2012-10-05: 288 3rd Qu.:1766.2   
## Max. :806.00 2012-10-06: 288 Max. :2355.0   
## NA's :2304 (Other) :15840

## Exploratory Data Analysis

### Numerical Summaries

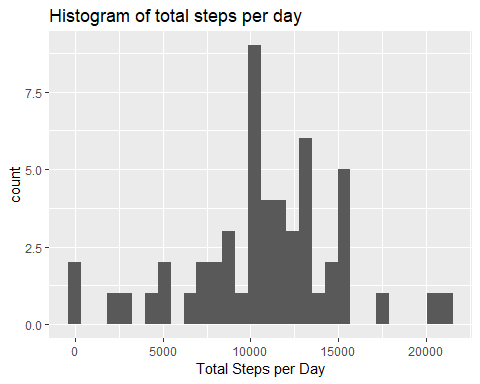
## Mean and Median number of steps taken each day by grouping daywise  
steps\_details <- activity %>%  
 group\_by(date) %>%  
 summarise(steps\_per\_day = sum(steps, na.rm = FALSE), median\_per\_day = median(steps),  
 mean\_per\_day = mean(steps))  
head(steps\_details)

## # A tibble: 6 x 4  
## date steps\_per\_day median\_per\_day mean\_per\_day  
## <fct> <int> <dbl> <dbl>  
## 1 2012-10-01 NA NA NA   
## 2 2012-10-02 126 0 0.438  
## 3 2012-10-03 11352 0 39.4   
## 4 2012-10-04 12116 0 42.1   
## 5 2012-10-05 13294 0 46.2   
## 6 2012-10-06 15420 0 53.5

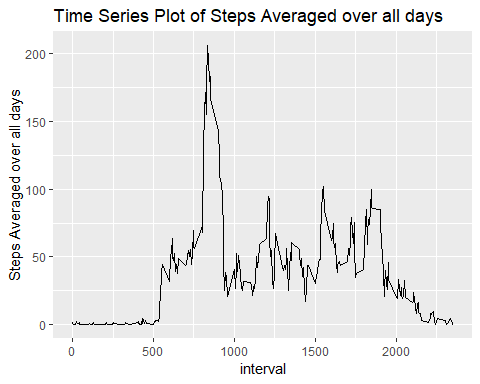
### Plots

## Plotting a Histogram  
ggplot(mapping = aes(x = steps\_details$steps\_per\_day))+  
 geom\_histogram()+   
 ggtitle("Histogram of total steps per day")+  
 xlab("Total Steps per Day")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



## Average steps during each time interval  
# Grouping the variables on the basis of the interval  
interval\_details <- activity %>%  
 group\_by(interval) %>%  
 summarise(mean\_steps = mean(steps, na.rm = TRUE))  
  
# Time series plot  
ggplot(interval\_details, mapping = aes(x=interval, y=mean\_steps))+  
 geom\_line()+  
 ggtitle("Time Series Plot of Steps Averaged over all days")+  
 ylab("Steps Averaged over all days")



Calculating the interval during which mean number of steps is the highest:

interval\_details$interval[which.max(interval\_details$mean\_steps)]

## [1] 835

## Data Cleaning

We first get a count of the missing values:

lapply(activity, function(x) sum(is.na(x)))

## $steps  
## [1] 2304  
##   
## $date  
## [1] 0  
##   
## $interval  
## [1] 0

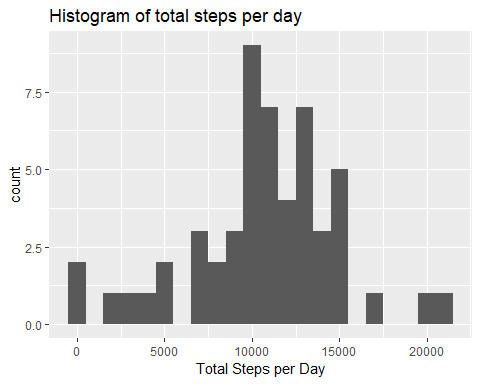
### Imputing the missing Values:

Only ‘steps’ has missing values. Hence, replace them by the mean and store the analytic data in a new dataset - ‘activity\_ana’

activity\_ana <- activity  
activity\_ana$steps[is.na(activity\_ana$steps)] <-   
 mean(activity\_ana$steps, na.rm = TRUE)

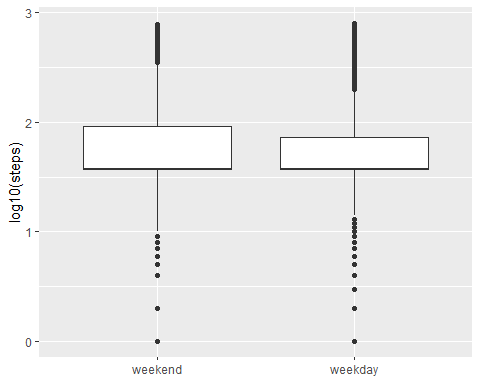
### Plots

## Mean and Median number of steps taken each day by grouping daywise  
activity\_ana%>%  
 group\_by(date) %>%  
 summarise(steps\_per\_day = sum(steps)) %>%  
   
 ggplot(steps\_per\_day, mapping = aes(x = steps\_details$steps\_per\_day))+  
 geom\_histogram(binwidth = 1000)+  
 ggtitle("Histogram of total steps per day")+  
 xlab("Total Steps per Day")



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

# Creating the factor variable 'day\_type'  
weekdays1 <- c('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday')  
activity\_ana$day\_type <-   
 factor((weekdays(as.Date(activity\_ana$date)) %in% weekdays1),   
 levels=c(FALSE, TRUE), labels=c('weekend', 'weekday'))  
  
# Boxplot to compare the steps during weekdays and weekends  
ggplot(activity\_ana, mapping = aes(x = day\_type, y = log10(steps)))+  
 geom\_boxplot() +  
 theme(axis.title.x = element\_blank())



**Narrative:** - The number of steps is less variable during weekends - The median number of steps is slightly more during weekends