weightprog

Introduction

In this vignette, we will demonstrate how to use the package, **weightprog**. This package will use data from the smartphone "Pillow" and "Health" app that provides sleep cycle and activity level statistics. The objective of the package is to allow for easy visualization and analysis for comparisons of any statistical changes in our focused variable over a time period.

Description

weightprog() specifically provides analysis through visualization through graphics. The objective of visualization in this package is to introduce a third variable (our objective variable), and present progressions of our objective variable without needing to refer to a three dimensional graph. The visualization graphics, embeds ggplots that specifically uses color and sizes as an aid to enhance basic two way graphs. The manipulation of size and color will allow an inclusion of such attributes of a third variable, while allowing distinctions between any groups of interest, but more importantly, contained in a two-dimensional graph.

Usage

There will be one primary data set that will be included in the package. The data includes an individual's weight progression with data on sleep cycles (i.e. hours of sleep, percent in deep sleep, etc.) and natural daily exercises (i.e. distance walked, steps in a day). The specific focus is on daily progression of weight, thus, the marginal change in weight from one day to the next will be the focal point of all visualization and analysis. First, we will load the package and the data. The data is imported from a publicized google document sheet.

```
library('weightprog')
```

```
library("googlesheets")
mykey <- gs_url("https://docs.google.com/spreadsheets/d/10KwAd2PL4FXbNrWkhLstzeuKvR9NYhfvegtPdjfELcM/ed
hello <- gs_read(mykey)
```

Before proceeding with the visualizations, we will first need to find a presentable objective variable that will be the focal point of our statistical analysis. The objective variable chosen for the package is the marginal weight change from one day to the next. To determine the relative size of such changes, we will take the absolute value of the marginal weight change, and also determine whether such changes were a loss, gain, or maintain in weight. There will be further explanation on what we mean by objective variable when we introduce the data in the package.

Add Absolute Weight and Weight Change Groups

```
hello$absmargweight <- abs(hello$Marginal_Weight)
hello$posnegmarg <- c("Loss","Maintain","Gain")[sign(hello$Marginal_Weight)+2]
colnames(hello)[colnames(hello)=="absmargweight"] <- "Marginal_Change"
colnames(hello)[colnames(hello)=="posnegmarg"] <- "Class"
```

We will also have to change the format of the date in order to fit the needs of our visualization functions along with organizing the days of the week to follow the sequential order. This is a quick fix:

Changing Format of Date and Organizing Days of Week

```
dd = as.Date(hello$Date, format = "%m/%d/%Y")
hello$Day <- factor(hello$Day, levels=c("Monday", "Tuesday",
                                        "Wednesday", "Thursday",
                                        "Friday", "Saturday", "Sunday"))
```

Introduction to Package Data

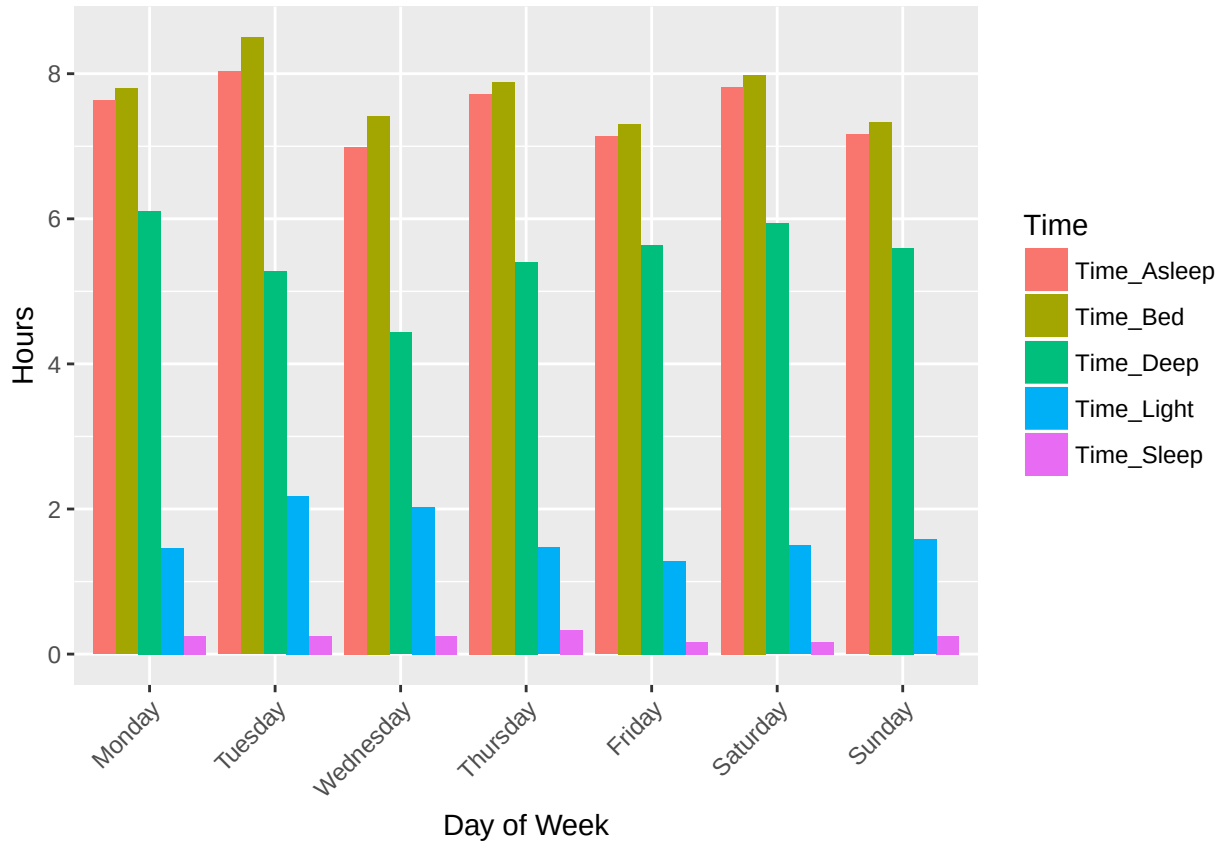
Before introducing the functions in the package, let's take a closer look at the data that is built into the package. Since the statistics consist of activity, measured only by distance walked, any functions will always use distance as the primary measurement. As for the sleep cycle, there are numerous categories of categories, and to take look at what each of the categories mean, we will first look at the summarization of the means for our sleep cycle statistics. For each statistic: Time_Asleep is hours asleep, Time_Bed is hours in bed, Time_Deep is hours in deep sleep, Time_Light is hours in light sleep, and Time_Sleep is hours till asleep. Since the mean does not tell us much about which category can be used as a primary measurement, we can proceed to using a linear regression and ANOVA.

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr
```

```
## Conflicts with tidy packages ----------------------------------------------
```

```
## filter(): dplyr, stats
## lag():    dplyr, stats
```

```
library(tidyr)
base <- hello %>%
  mutate(Time_Deep = Deep_Sleep * Time_Asleep / 100) %>%
  mutate(Time_Light = Light_Sleep * Time_Asleep / 100) %>%
  select("Day", "Time_Bed", "Time_Asleep", "Time_Sleep", "Time_Deep", "Time_Light")
base <- gather(base, Time_Asleep, value, -Day) %>%
        setNames(., c("Day", "Time", "value"))

ggplot(base, aes(Day, value)) +
geom_bar(aes(fill = Time), position = "dodge", stat="identity") +
xlab("Day of Week") + ylab("Hours") +
theme(axis.text.x=element_text(angle=45,hjust=1))
```



Choosing our Primary Variables

With plenty of variables that seem to overlap and express similar characteristics, we will specifically run a multiple regression of marginal weight change on the distance, time asleep, and time in deep sleep and light sleep. When we run a linear model through these variables, the only significant variable is that of the time in deep sleep. Hence, we will specifically use deep sleep as our primary measurement for our sleep cycle (occasionally, we will use time asleep because it is a more general measurement). Likewise, when we run an ANOVA on the change in weight with variables of distance walked (measuring activity level) and time in deep sleep (measuring sleep cycle), we observe that deep sleep is on the borderline of being a statistically significant variable. Therefore, there is convincing evidence of a statistically significant difference in the time of deep sleep amongst changes in weight. Therefore, the two primary measurements that will be utilized will be distance for activity level, and deep sleep (can be in either hours or percentage), along with the occasional use of time asleep as a more general measurement. (Note: For future changes, we will will surpress the output for the linear regression and ANOVA.)

```
base2 <- hello %>%
  mutate(Time_Deep = Deep_Sleep * Time_Asleep / 100) %>%
  mutate(Time_Light = Light_Sleep * Time_Asleep / 100) %>%
  select("Distance","Time_Bed", "Time_Asleep", "Time_Deep", "Time_Light", "Marginal_Weight")

fit <- lm(Marginal_Weight ~ Distance + Time_Asleep + Time_Deep + Time_Light + Time_Bed, data=base2)
summary(fit) # show results
```

```
##
## Call:
## lm(formula = Marginal_Weight ~ Distance + Time_Asleep + Time_Deep +
```

```
##     Time_Light + Time_Bed, data = base2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37190 -0.20339 -0.02886  0.15975  0.57743
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.26262    0.56848   0.462   0.6483
## Distance    -0.01420    0.02775  -0.512   0.6135
## Time_Asleep -0.31611    0.34245  -0.923   0.3652
## Time_Deep    0.25604    0.11011   2.325   0.0288 *
## Time_Light   0.14829    0.23174   0.640   0.5283
## Time_Bed     0.07915    0.28549   0.277   0.7840
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.284 on 24 degrees of freedom
## Multiple R-squared:  0.2516, Adjusted R-squared:  0.09572
## F-statistic: 1.614 on 5 and 24 DF,  p-value: 0.1945
```

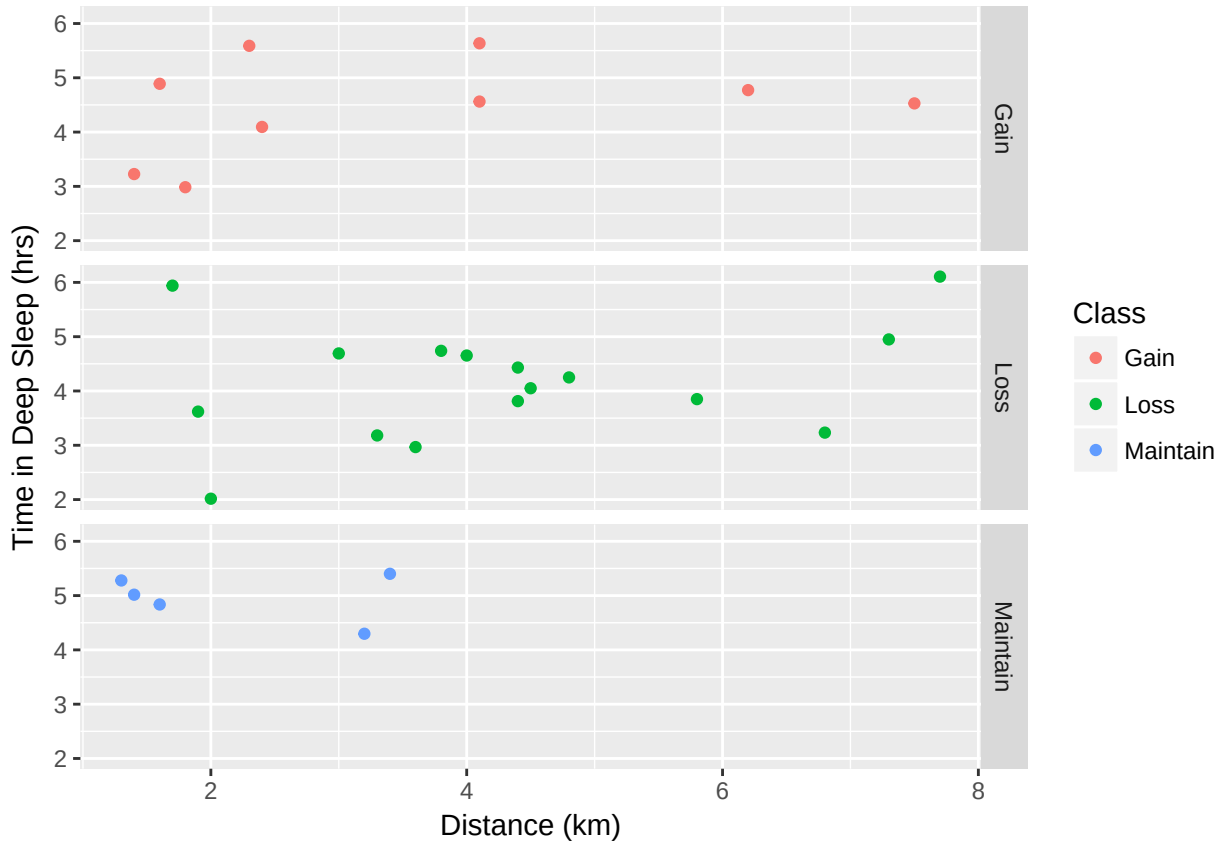
```
weightprog1 = aov(Marginal_Weight ~ Distance*Time_Deep, data = base2)
summary(weightprog1)
```

```
##                     Df Sum Sq Mean Sq F value Pr(>F)
## Distance             1 0.0096  0.0096   0.115 0.7374
## Time_Deep            1 0.4017  0.4017   4.805 0.0375 *
## Distance:Time_Deep   1 0.0020  0.0020   0.024 0.8775
## Residuals           26 2.1734  0.0836
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Understanding the Objective Variable

With the primary variables chosen to represent activity level and sleep cycle, we can now shift our focus to seeing if specific characteristics in distance and time in deep sleep seem result in some type of pattern for the changes in weight. In this case, the variable of focus will be called the objective variable, consisting of the marginal changes in weight, with subgroups (called Class) of "Gain", "Loss", and "Maintain" in weight. Here is a scatterplot that shows a general overview of the two primary variables by the objective variable's subgroups:

```
intro <- hello %>%
  mutate(Time_Deep = Deep_Sleep * Time_Asleep / 100) %>%
  select("Distance", "Time_Deep", "Class") %>% ggplot(aes(x=Distance, y=Time_Deep, col = Class)) +
  geom_point() + xlab("Distance (km)") + ylab("Time in Deep Sleep (hrs)") +
  facet_wrap(~Class , ncol=1, scales="fixed", strip.position="right")
intro
```



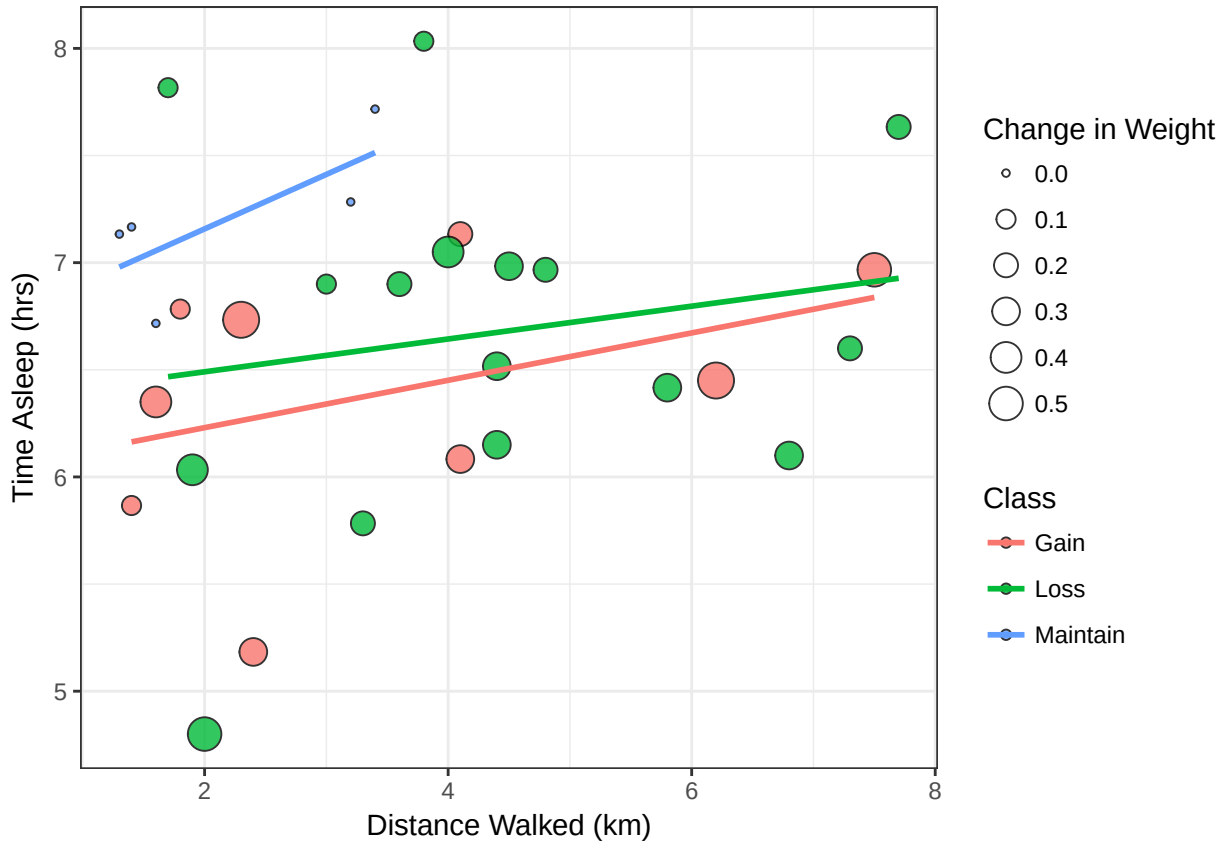
Once we have gotten a better understanding of the built-in dataset in the package, we may proceed to the two-way graphs, visualizations, and analysis in the `weightprog` package.

Two Way Scatter Feature

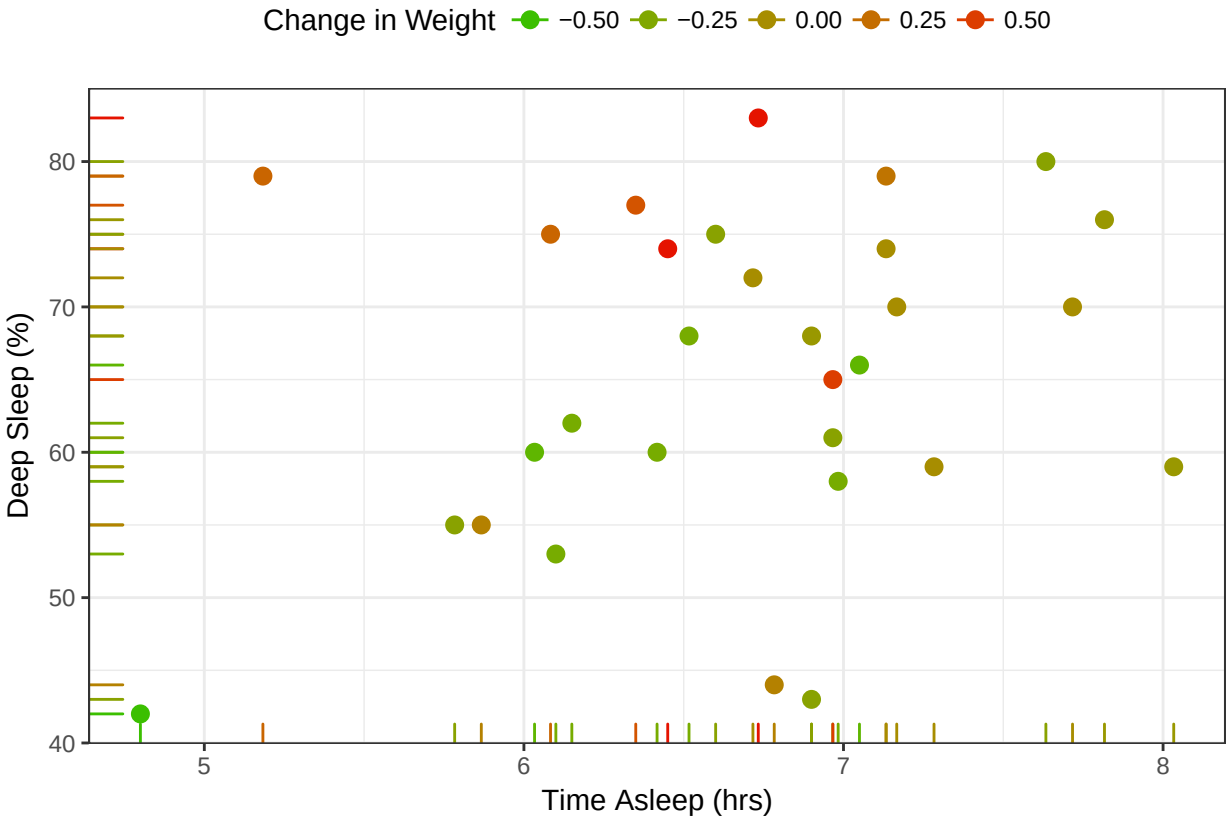
There are two different graphs within the two way scatter feature, twoway_scatterg(), and twoway_scatterc(). Both of these functions utilize a simple scatter plot between two variables of interest, and adds an additional layer for the objective variable. The two way scatter of objective variable by groups is the first of two functions, specifically aimed at using the objective variable's data points' color and size to provide information about the attributes. In addition, it runs a linear model through the specific subgroups of the objective variable. Meanwhile, the latter of the two functions, two way scatter by color, only displays any statistical changes in the objective variable through a sequential color palette.

```
library(ggplot2)
```

```
#source('/Users/Benjamin Hsu/Desktop/weightprog/R/twoway_scatter.R')
tws_plot <- twoway_scatterg(hello, hello$Distance, hello$Time_Asleep,
                            hello$Marginal_Change, hello$Class,
                            "Distance Walked (km)", "Time Asleep (hrs)",
                            "Change in Weight", "Class")
tws_plot
```




```
tws_plot2 <- twoway_scatterc(hello, hello$Time_Asleep, hello$Deep_Sleep,
                  hello$Marginal_Weight, 16, 3,
                  "Time Asleep (hrs)", "Deep Sleep (%)",
                  "Change in Weight")
tws_plot2
```



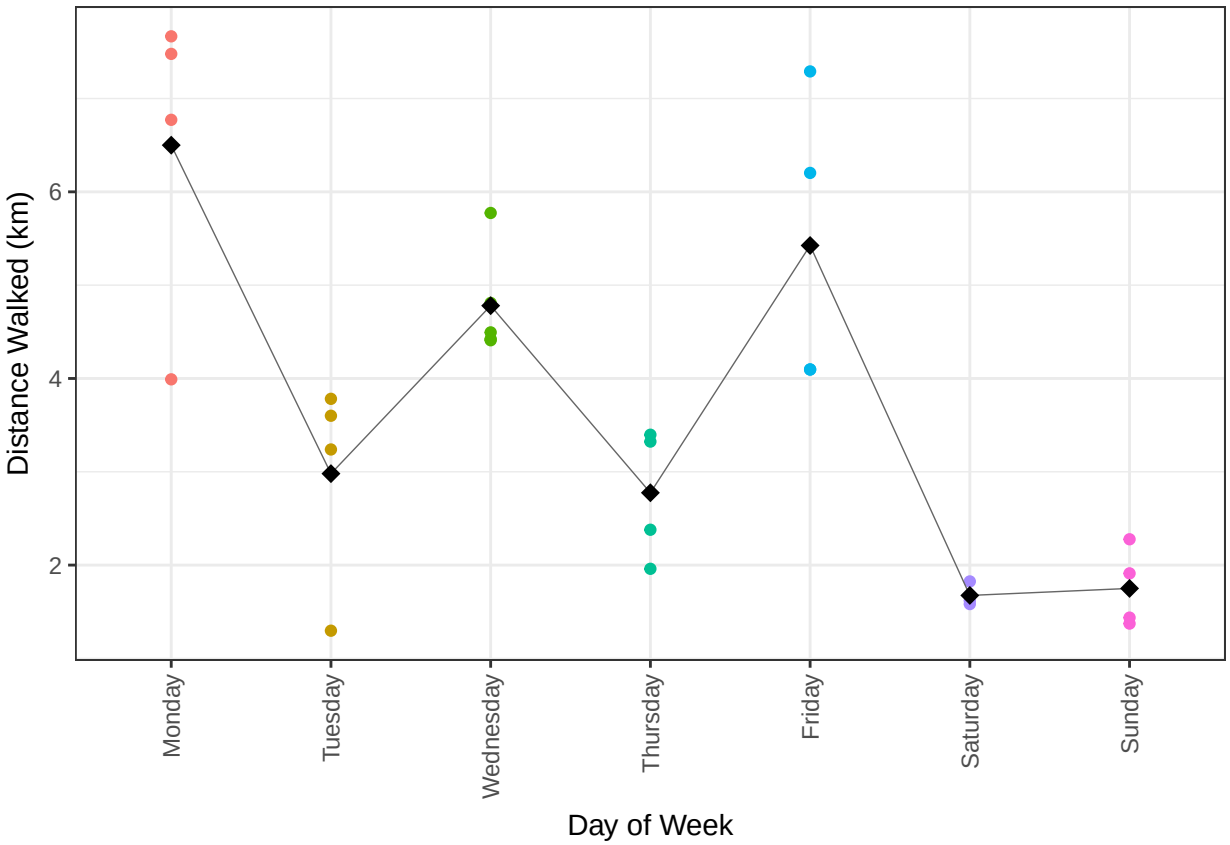
For the data inside the package, the objective variable is the marginal weight, specifically focused on the magnitude (size) of the change in marginal weight, and whether or not such changes were gain, loss or maintain in weight (color). Both scatter plots have the base scatterplot, for twoway_scatterg(), it was Time Asleep vs. Distance Walked, and for twoway_scatterc(), it was %Deep Sleep vs. Time Asleep.

The speculation for the grouped scatter plot was that the more distance walked during the day, the more sleep an individual will need. For the two way scatter of the objective variable by groups, the two subgroups of gain and loss in weight showed a poor linear fit. Meanwhie, the maintain in weight saw a good linear fit. For the two way scatter of the objective variable by color, the speculation here was that more hrs of sleep might be because of a higher proportion of deeper sleep, and thus, possible gain in weight (assuming sleeping is minimal activity). The general pattern consists of losses in weight (green) in the lower left corner, which denotes less hours of sleep and less deep sleep, and the gains in weight (red) in the upper left corner denotes less hours of sleep and greater sleep. Both these scatterplots result in interesting conclusions, but as more data is included into the package, then a more conclusive result can be deduced.

Two Way Strip Plot

The strip plot is similar to the simplified two way scatter plot, because the points' color denotes the attribute (i.e. day of the week, species, etc.) of y variable chosen. The strip plot provides a better overview of the data's patterns within a subgroup, while also providing a lineplot of the estimate mean of each group.

```
#source('/Users/Benjamin Hsu/Desktop/weightprog/R/twoway_strip.R')
twoway_strip(hello, hello$Day, hello$Distance, " ",
             hello$Day, "Day of Week", "Distance Walked (km)")
```

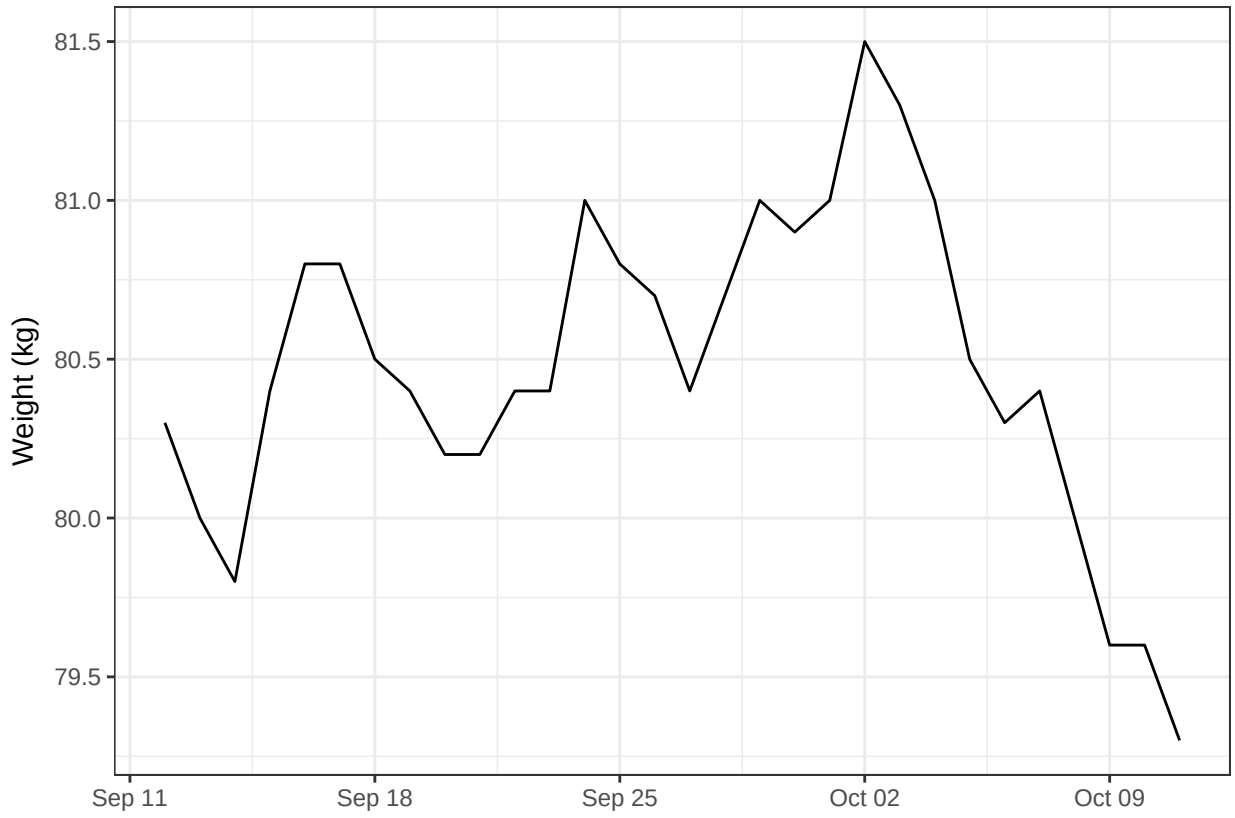


There was speculation that for different days of the week, an individual might be walked farther in distance, which might inherently mean that the activity level from one day to another differ. With different activity levels, it would be possible that this could affect the individual's marginal changes in weight. In the data, we specifically looked at the individual's activity level measured by distance walked in kilometers depending on the day of the week. We specifically see that the mean of distance walked seem to be on Monday and Friday, while the weekends seem to have significantly lower activity levels.

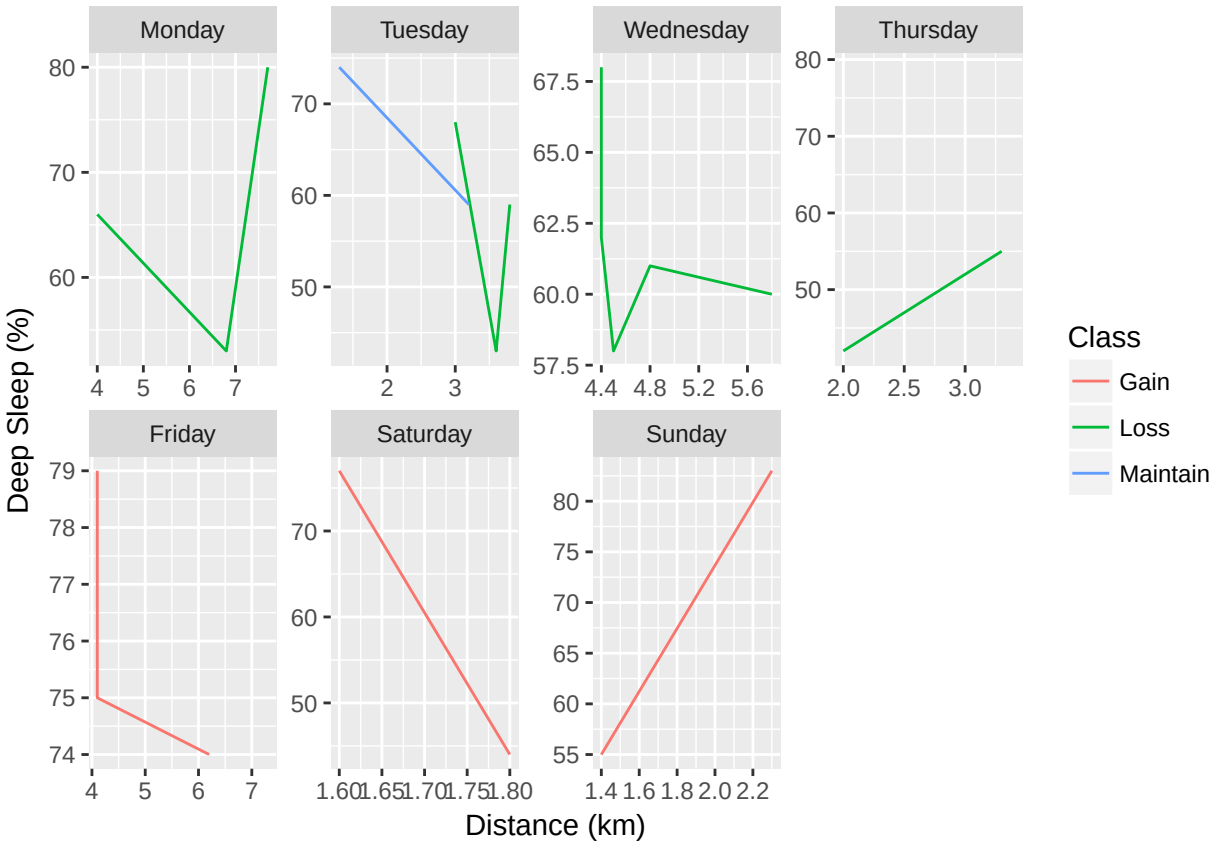
Two Way Lineplot

There are two different two way lineplots that can are generated, twoway_lineplot() and twoway_lineplotf(). The first is a connected scatterplot of a y variable observed over a time period. Though there is only one y variable in consideration, it would also be possible to add multiple other lines into the graph by re-standardizing the data to fit into the specified range. The second lineplot is used for facetting by the specific characteristic of the objective variable. The plot creates multiple lines in each characteristic, which helps identify any possibe trends in the data within each subgroup and characteristic. Both lineplots shows a basic summary of the statistical progression of the variable of interest.

```
#source('/Users/Benjamin Hsu/Desktop/weightprog/R/twoway_lineplot.R')
twl <- twoway_lineplot(hello, dd, hello$Weight, "Weight (kg)")
twl
```




```
twl2 <- twoway_lineplotf(hello, xname = "Distance", yname = "Deep_Sleep",
                 bycolor = "Class", bywrap = "Day",
                 "Distance (km)", "Deep Sleep (%)")
twl2
```



The lineplot for the data used simply shows the progression of the individual's weight over time. We see a rather constant fluctuation in the weight, between the range of approximately 79 to 82 kilograms. The overall trend of the individual's weight seems to be increasing over time. For the future, additional lines will be considered once more data is provided, which will allow for a more accurate re-standardization of the added variable.

Meanwhile, the second lineplot is facetted by the day of the week, and produces a lineplot for each of the subgroups of the objective variable (that is: loss, gain, and maintain). Each day will present a lineplot of the trend between the x and y variable amongst the subgroup. Specifically, we look at the relationship bewteen distance walked and the percent of deep sleep per day, based on weight changes. So far, we see that from Monday to Thursday, there seems to be only losses in weight, especially with higher amounts of distance walked, but an inconsistent deep sleep percentage. Meanwhile, for Friday, Saturdays, and Sundays, there are only gains in weight, with a substantially lower distance walked on weekends, but a moderate consistency in deep sleep percentages. This lineplot by facetting the objective variable allows for a specific observation of each day's progress of weight changes, and how this progression changes with relation to distance and deep sleep percentage. Though there does seem like a preliminary pattern exists amongst the day of the week and weight changes, it will be interesting to see such weight fluctuations as more data is collected.

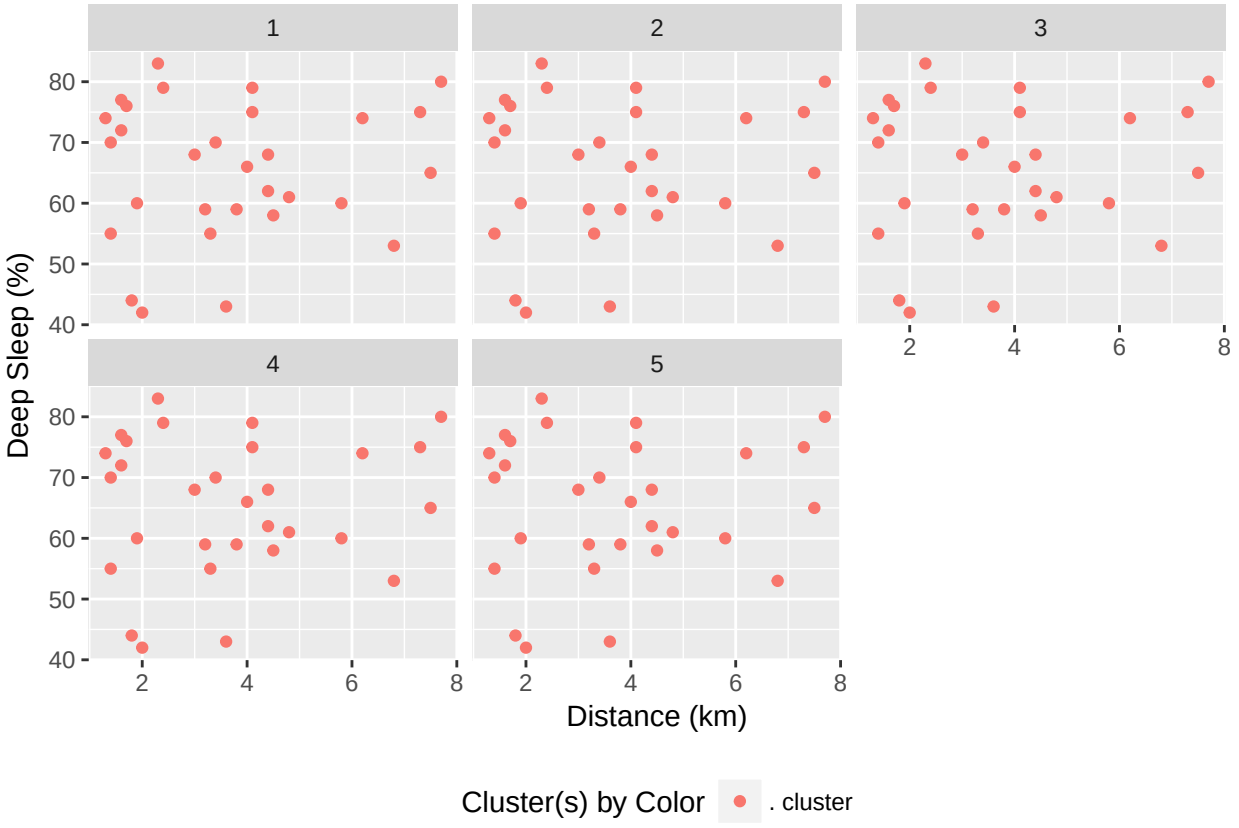
K-Means Clustering

The k-means clustering allows for us to find groups that have not been explicitly defined in the data, specifically through two variables of interest. The function creates k centroids, allowing for k clusters to be created, while plotting all these different k-means clusters to allow for a decision on how many clusters to use. In addition, this allows for a confirmation of whether or not the objective variables' subgroups really have a cluster. Though this is a less explicit way of answering whether there is a relationship between two variables

according to the objective variable, it allows us to observe almost all possibilities, from small to large amounts of clusters to confirm previous findings. (Note: this k-means is modified from the internet)

```
twoway_kmeans(hello$Distance, hello$Deep_Sleep, 5,
              "Distance (km)", "Deep Sleep (%)", "Cluster(s) by Color")
```

```
## Warning: Grouping rowwise data frame strips rowwise nature

## Warning in bind_rows_(x, .id): Unequal factor levels: coercing to character

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
```



To further see if an individual's weight changes is affected by the speculation of a relationship between distance walked and percent in deep sleep, the function can confirm how these weight changes are clustered. Theoretically, we would say that higher activity levels might induce a more tired state subsequently higher

percentage of deep sleep, which can then split into the subgroups of our objective variable. For gains in weight, we would speculate shorter distance (low activity), and losses in weight with longer distance (high activity), which would mean that for each subgroup, there should be a cluster.

Ideally, we would only need three clusters, however in the function, we do not see any clusters with a consistent subgroup, since the class (gain/loss/maintain) in each cluster seem to vary. When we do look at the three clusters created, there is a little bit of everything, which may improve as more data is collected. Specifically we see the the middle cluster consists of primary losses, with distance varying, but deep sleep percentage between 50 to 60 percent. The cluster at the top seem to consist of primarily gain, but with plenty of other classes as well. When we observe two clusters only, the bottom of the two clusters seem to be primarily defined by the subgroup for loses in weight with relative precision, and the top cluster for gain with lack of precision.

While there will always be mis-classified data, the top cluster does not show much consistency in our data, which means that we cannot generalize much about subgroups gain and maintain in weight. However, we do see that losses in weight tend to be classified together with relative consistency with distances from 0 to 8 km and deep sleep percentage from 40 to 65 percent.

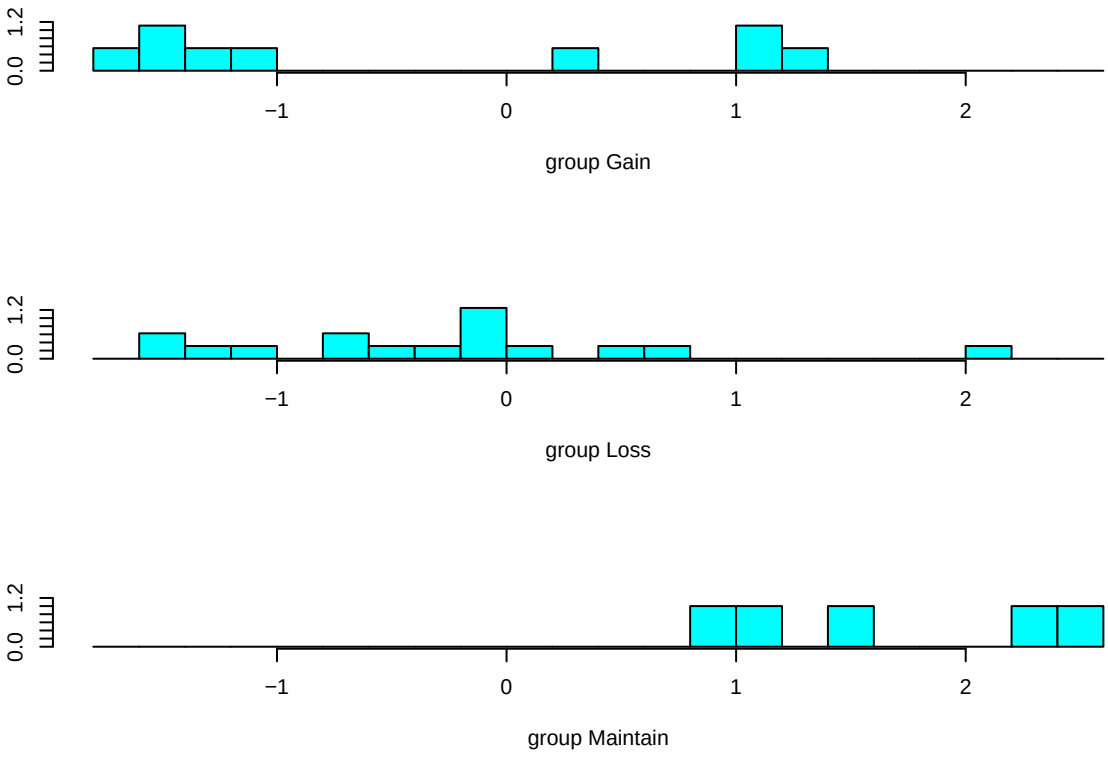
Linear Discriminant Analysis

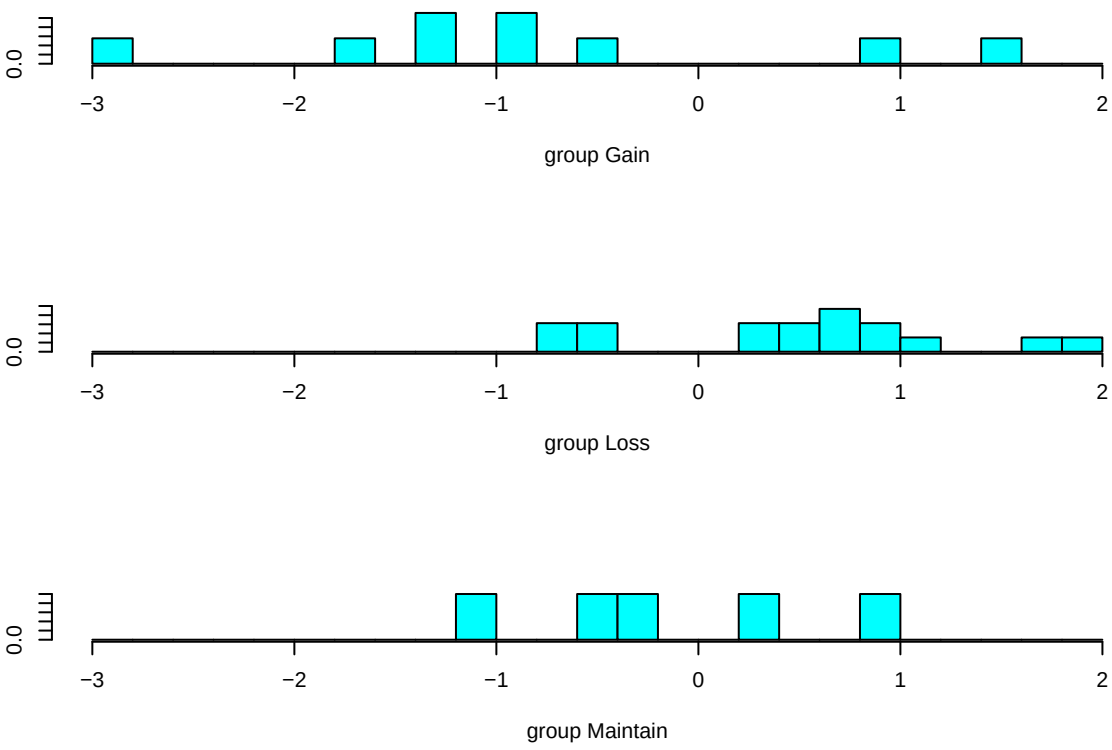
The linear discriminant analysis (LDA) allows a more definitive pattern recognition method in comparison to cluster analysis. This type of machine learning focuses on finding a linear combination of features that help separate the groups as far as possible. In addition, linear discriminant analysis might be a better way compared to that of clustered analysis because instead of determining a pattern for grouping certain variables, LDA allows us to utilize the property that the membership to each classification is established already. To use the linear discriminant analysis, the underlying assumption here is that the data is normally distributed, and each of the subgroups have equivalent variances.

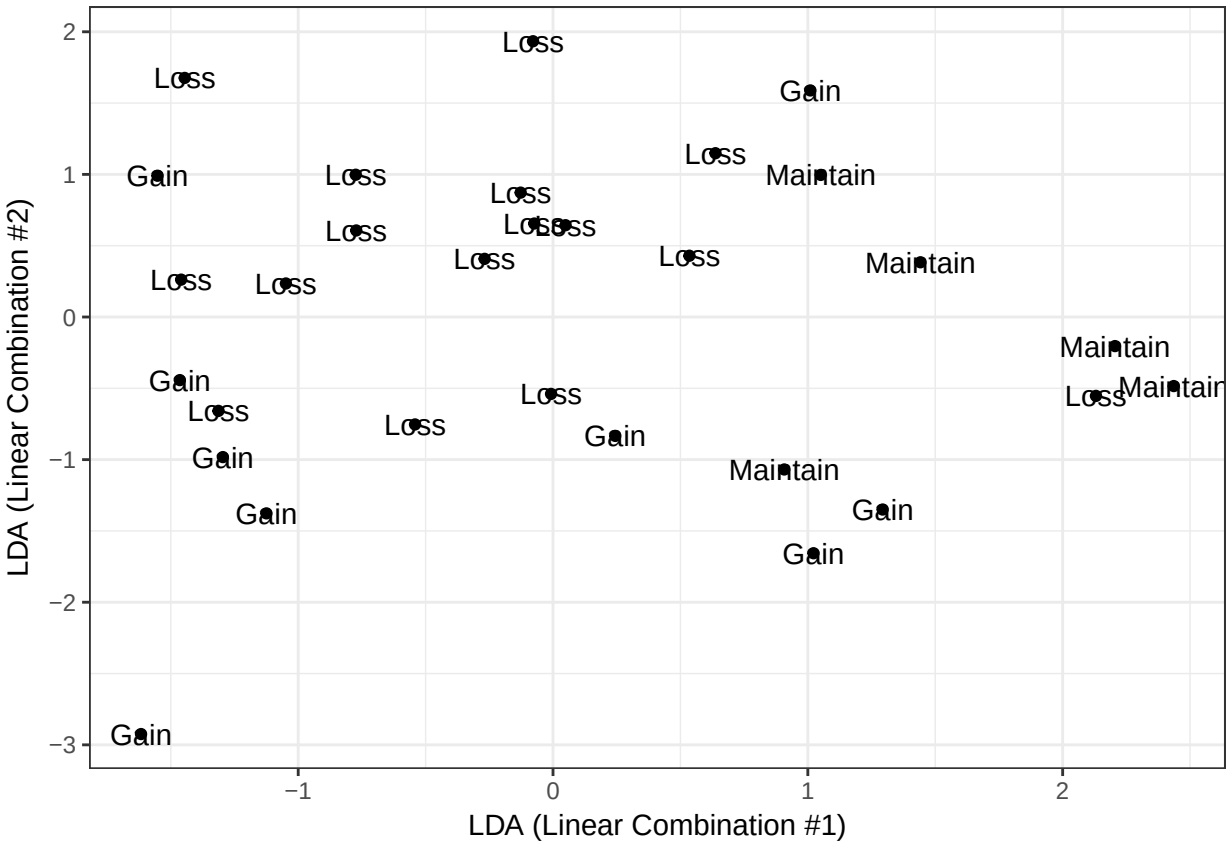
For the package data, the goal of using linear discriminant analysis was to find a linear combination for the selected variables that will provide the largest distinction from one group to the next. The variables used to provide the best separation for the class subgroups (gain/loss/maintain) included distance, time asleep, and percentage in deep and light sleep. For each of the two discriminant functions, a histogram will be used to provide visualization of each class' separation. Along with the histogram, a scatterplot of the first and second discriminant function's values for the classes will be provided. Both the histograms and scatterplots will provide visualizations to pinpoint the presence of any characteristic difference for each class.

```
library(broom)
```

```
twoway_lda(hello$Class, hello$Distance, hello$Time_Asleep,
           hello$Deep_Sleep, hello$Light_Sleep, hello, hello$Class)
```


0.0
1.2
−1
0
1
2
group Gain
0.0
1.2
−1
0
1
2
group Loss
0.0
1.2
−1
0
1
2
group Maintain


0.0
−3
−2
−1
0
1
2
group Gain
0.0
−3
−2
−1
0
1
2
group Loss
0.0
−3
−2
−1
0
1
2
group Maintain



For the first discriminant function, we speculate that the maintain class may have some separation from the loss class. To better see if there is such discrepancies, we take a look at the second discriminant function, which refutes this statement. Though the loss group does seem to be in a small class of its own, it does not seem to be distinctly separate from the other classes. When we continue and take a look at the scatterplot of the two discriminant functions, we observe that the first discriminant function does indeed have a moderate separation between the maintain and loss group along the x axis. However, for the second discriminant function, it is of the contrary, as none of the groups seem to be separate from one another. Though we do not have much conclusive evidence, the linear discriminant analysis has helped narrow down the possibility that there may exist some type of separation between the loss and maintain group with regards to the characteristics used in distance, time asleep, and percent in deep and light sleep.

Future Analysis

As more data for the package is observed over time, we will get a clearer picture of the trends and patterns in each of the visualization graphs. There are many different and interesting variables that can be paired together that could potentially suggest an influence on the objective variable of the marginal change in weight. Though we have defined the main variables we will still need to find other variables to explore.

In addition, in the future, we might be interested in further first introducing a visualization that explores the relationship between an activity level and sleep cycle variable, along with the objective variable. This will allow for a better understanding of how it is difficult to deduce a relationship between these variables, which is why we proceeded to use cluster analysis and linear discriminant analysis. We could use a contour plot first in our introduction to display how the objective variable changes in two dimensions with the chosen x and y variables. The greatest problem that using a contour plot is that there is little variation in the weight, and a serious problem in the number of observations. These problems might produce a contour plot that does not provide much introduction to our variables.

Session Info

```
sessionInfo()
```

```
## R version 3.4.1 (2017-06-30)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 15063)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] broom_0.4.2        bindrcpp_0.2       dplyr_0.7.3
##  [4] purrr_0.2.3        readr_1.1.1        tidyr_0.7.1
##  [7] tibble_1.3.4       ggplot2_2.2.1      tidyverse_1.1.1
## [10] googlesheets_0.2.2 weightprog_0.2.2
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_0.2.0 reshape2_1.4.2   haven_1.1.0      lattice_0.20-35
##  [5] colorspace_1.3-2 htmltools_0.3.6  yaml_2.1.14      rlang_0.1.2
##  [9] foreign_0.8-69   glue_1.1.1       modelr_0.1.1     readxl_1.0.0
## [13] bindr_0.1        plyr_1.8.4       stringr_1.2.0    munsell_0.4.3
## [17] gtable_0.2.0     cellranger_1.1.0 rvest_0.3.2      psych_1.7.8
## [21] evaluate_0.10.1  labeling_0.3     knitr_1.17       forcats_0.2.0
## [25] parallel_3.4.1   curl_2.8.1       Rcpp_0.12.12     scales_0.5.0
## [29] backports_1.1.0  jsonlite_1.5     mnormt_1.5-5     hms_0.3
## [33] digest_0.6.12    stringi_1.1.5    grid_3.4.1       rprojroot_1.2
## [37] tools_3.4.1      magrittr_1.5     lazyeval_0.2.0   pkgconfig_2.0.1
## [41] MASS_7.3-47      xml2_1.1.1       lubridate_1.6.0  assertthat_0.2.0
## [45] rmarkdown_1.6    httr_1.3.1       R6_2.2.2         nlme_3.1-131
## [49] compiler_3.4.1
```