Exploratory Data Analysis

Brooke Beanland UTEID: btb949

Introduction

This project will explore personal health data from wearable technology. The two data sets used are Health_Data and Workout_Data. The health data was exported from the apple health app with data obtained from my apple watch. Similarly, the workout data was exported from the apple watch activity application.

These datasets have the common varaible of Date. The health data set contains the variables of total active calories, total distance, total steps, and total resting calories for each date in which it was worn. Similarly, the workout data set contains information from each date I worked out with my apple watch on, and contains the varaibles workout type, duration, distance, average heart rate, max heart rate, average pace, active energy, and total energy. These data sets will be joined using the Date columns, and relationships will be explored. The relationship expected between the variables is a higher total distance, total active calories, and total steps on the dates in which a recorded workout occured.

Exploring this data is interesting because it informs me, the user, on which workouts, and specific aspects of those workouts, results in highest total fitness acheivements that day. It will be interesting to see if specific types of workouts influence the aformentioned the health data totals in a continuous or categorical manner.

```
library(knitr)
library(tidyverse)
library(ggplot2)
library(dplyr)
library(ggplot2)
library(cluster)
library(GGally)
library(readxl)
Workout_Data <- read_excel("~/Downloads/Workout_Data.xls")
Health_Data <- read_excel("~/Downloads/Health Data.xls")
glimpse(Health_Data)
```

```
## Observations: 88
## Variables: 5
## $ Start                      <dttm> 2019-12-01, 2019-12-02, 2019-12-03, 2019...
## $ `Active Calories (kcal)`   <dbl> 0.0000000, 0.0000000, 0.0000000, 0.000000...
## $ `Distance (mi)`            <dbl> 1.414756407, 2.819737620, 4.138211731, 4....
## $ `Resting Calories (kcal)` <dbl> 4.564664, 4.564664, 4.564664, 4.564664, 4...
## $ `Steps (count)`            <dbl> 3317.000, 6344.000, 9966.000, 9506.000, 4...
```

```
glimpse(Workout_Data)
```

```
## Observations: 50
```

```
## Variables: 11
## $ Type                  <chr> "Indoor Running", "Functional Strength Trainin...
## $ Date                  <dttm> 2020-01-01, 2020-01-01, 2020-01-02, 2020-01-0...
## $ Time                  <chr> "5:14PM", "4:19PM", "5:15PM", "5:43PM", "12:34...
## $ Duration              <dttm> 1899-12-31 00:23:16, 1899-12-31 00:09:40, 189...
## $ Distance              <dbl> 2.96387190, NA, NA, NA, NA, 2.78356229, NA, NA...
## $ `Average Heart Rate` <dbl> 178.06593, 131.30973, 183.51837, 127.49147, 14...
## $ `Max Heart Rate`      <dbl> 199, 155, 199, 172, 170, 197, 165, 173, 96, 98...
## $ `Average Pace`        <dttm> 1899-12-31 00:07:51, NA, NA, NA, NA, 1899-12-...
## $ `Average Speed`       <dbl> 7.6427692, NA, NA, NA, NA, 7.2269083, NA, NA, ...
## $ `Active Energy`       <dbl> 180.661, 50.000, 193.000, 119.567, 75.000, 168...
## $ `Total Energy`        <dbl> 211.407, 76.264, 220.384, 168.354, 94.352, 199...
```

Tidying the Data

In order to demonstrate the usage of tidy functions, I first untidyed one of my datasets, Workout_Data, by using pivot_wider. I widened by data by type of workout, and used Average Heart Rate as the value selected to widen the data by. I then reversed this function by using pivot_longer to lengthen the data by workout type as well as creating a separate column for Average Heart Rate.

```
Wider_WorkoutData<-Workout_Data%>%pivot_wider(names_from = "Type", values_from = "Average Heart Rate")
glimpse(Wider_WorkoutData)
```

```
## Observations: 50
## Variables: 15
## $ Date                              <dttm> 2020-01-01, 2020-01-01, 2020-01...
## $ Time                              <chr> "5:14PM", "4:19PM", "5:15PM", "5...
## $ Duration                          <dttm> 1899-12-31 00:23:16, 1899-12-31...
## $ Distance                          <dbl> 2.96387190, NA, NA, NA, NA, 2.78...
## $ `Max Heart Rate`                  <dbl> 199, 155, 199, 172, 170, 197, 16...
## $ `Average Pace`                    <dttm> 1899-12-31 00:07:51, NA, NA, NA...
## $ `Average Speed`                   <dbl> 7.6427692, NA, NA, NA, NA, 7.226...
## $ `Active Energy`                   <dbl> 180.661, 50.000, 193.000, 119.56...
## $ `Total Energy`                    <dbl> 211.407, 76.264, 220.384, 168.35...
## $ `Indoor Running`                  <dbl> 178.0659, NA, 183.5184, NA, NA, ...
## $ `Functional Strength Training`    <dbl> NA, 131.3097, NA, NA, 141.5480, ...
## $ Yoga                              <dbl> NA, NA, NA, 127.49147, NA, NA, 1...
## $ `Cross Training`                  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ `High Intensity Interval Training` <dbl> NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ `Outdoor Walking`                 <dbl> NA, NA, NA, NA, NA, NA, NA, NA, ...
```

```
Longer_WorkoutData<- Wider_WorkoutData%>%pivot_longer(col=c("High Intensity Interval Training", "Outdoo
names_to = "Type", values_to = "Average Heart Rate")
glimpse(Longer_WorkoutData)
```

```
## Observations: 300
## Variables: 11
## $ Date                  <dttm> 2020-01-01, 2020-01-01, 2020-01-01, 2020-01-0...
## $ Time                  <chr> "5:14PM", "5:14PM", "5:14PM", "5:14PM", "5:14P...
## $ Duration              <dttm> 1899-12-31 00:23:16, 1899-12-31 00:23:16, 189...
## $ Distance              <dbl> 2.963872, 2.963872, 2.963872, 2.963872, 2.9638...
## $ `Max Heart Rate`      <dbl> 199, 199, 199, 199, 199, 199, 155, 155, 155, 1...
```

```
## $ `Average Pace`       <dttm> 1899-12-31 00:07:51, 1899-12-31 00:07:51, 189...
## $ `Average Speed`      <dbl> 7.642769, 7.642769, 7.642769, 7.642769, 7.6427...
## $ `Active Energy`      <dbl> 180.661, 180.661, 180.661, 180.661, 180.661, 1...
## $ `Total Energy`       <dbl> 211.407, 211.407, 211.407, 211.407, 211.407, 2...
## $ Type                 <chr> "High Intensity Interval Training", "Outdoor W...
## $ `Average Heart Rate` <dbl> NA, NA, 178.0659, NA, NA, NA, NA, NA, NA, NA, ...
```

Joining the Two Datasets

To join the two data sets I employed the inner_join function. This function was chosen because the goal was to create one dataset in which there were both overall health data totals and workout data for each date in the table. Inner_join acheives this becuase it joins the full datasets at the join variable, and drops any other rows that do not meet the join varaible requirment. After the inner_join, 38 rows were lost from the health data dataset, and zero rows were lost from the workout data dataset. The resulting joined data frame contained a total of 50 rows.

```
Health_Data<-Health_Data%>% rename(Date=Start)
fulldata<-inner_join(Health_Data, Workout_Data, by="Date")
nrow(Health_Data)
```

```
## [1] 88
```

```
nrow(Workout_Data)
```

```
## [1] 50
```

```
nrow(fulldata)
```

```
## [1] 50
```

```
glimpse(fulldata)
```

```
## Observations: 50
## Variables: 15
## $ Date                       <dttm> 2020-01-01, 2020-01-01, 2020-01-02, 2020...
## $ `Active Calories (kcal)`   <dbl> 206.7089, 206.7089, 117.3240, 461.7940, 4...
## $ `Distance (mi)`            <dbl> 2.2291425, 2.2291425, 0.6916531, 3.008742...
## $ `Resting Calories (kcal)` <dbl> 1504.601, 1504.601, 1556.886, 1458.973, 1...
## $ `Steps (count)`            <dbl> 4762.000, 4762.000, 1471.000, 5598.000, 5...
## $ Type                       <chr> "Indoor Running", "Functional Strength Tr...
## $ Time                       <chr> "5:14PM", "4:19PM", "5:15PM", "5:43PM", "...
## $ Duration                   <dttm> 1899-12-31 00:23:16, 1899-12-31 00:09:40...
## $ Distance                   <dbl> 2.96387190, NA, NA, NA, NA, 2.78356229, N...
## $ `Average Heart Rate`       <dbl> 178.06593, 131.30973, 183.51837, 127.4914...
## $ `Max Heart Rate`           <dbl> 199, 155, 199, 172, 170, 197, 165, 173, 9...
## $ `Average Pace`             <dttm> 1899-12-31 00:07:51, NA, NA, NA, NA, 189...
## $ `Average Speed`            <dbl> 7.6427692, NA, NA, NA, NA, 7.2269083, NA,...
## $ `Active Energy`            <dbl> 180.661, 50.000, 193.000, 119.567, 75.000...
## $ `Total Energy`             <dbl> 211.407, 76.264, 220.384, 168.354, 94.352...
```

Summary Statistics

```
#I used filter to only select the workout type and then arranged the active calories in a descending ar

fulldata %>%filter(Type=="High Intensity Interval Training")%>%arrange(desc(`Active Calories (kcal)`))%
```

```
## # A tibble: 11 x 16
##    Date                `Active Calorie~ `Distance (mi)` `Resting Calori~
##    <dttm>                         <dbl>           <dbl>            <dbl>
##  1 2020-02-03 00:00:00             986.            7.05            1678.
##  2 2020-02-19 00:00:00             950.            7.14            1625.
##  3 2020-02-05 00:00:00             928.            7.10            1622.
##  4 2020-02-10 00:00:00             916.            7.28            1626.
##  5 2020-02-12 00:00:00             878.            6.05            1588.
##  6 2020-01-27 00:00:00             858.            6.83            1580.
##  7 2020-02-15 00:00:00             836.            3.73            1620.
##  8 2020-01-29 00:00:00             808.            5.45            1609.
##  9 2020-02-24 00:00:00             792.            5.54            1587.
## 10 2020-02-26 00:00:00             788.            7.85            1613.
## 11 2020-02-17 00:00:00             742.            5.59            1580.
## # ... with 12 more variables: `Steps (count)` <dbl>, Type <chr>, Time <chr>,
## #   Duration <dttm>, Distance <dbl>, `Average Heart Rate` <dbl>, `Max Heart
## #   Rate` <dbl>, `Average Pace` <dttm>, `Average Speed` <dbl>, `Active
## #   Energy` <dbl>, `Total Energy` <dbl>, mean_active <dbl>
```

```
#I used select here and grouped by Type.
fulldata%>%select('Type', 'Average Heart Rate', 'Max Heart Rate')%>%group_by(Type)
```

```
## # A tibble: 50 x 3
## # Groups:   Type [6]
##    Type                          `Average Heart Rate` `Max Heart Rate`
##    <chr>                                        <dbl>            <dbl>
##  1 Indoor Running                               178.               199
##  2 Functional Strength Training                 131.               155
##  3 Indoor Running                               184.               199
##  4 Yoga                                         127.               172
##  5 Functional Strength Training                 142.               170
##  6 Indoor Running                               171.               197
##  7 Yoga                                         126.               165
##  8 Yoga                                         118.               173
##  9 Yoga                                          75.7               96
## 10 Yoga                                          77.6               98
## # ... with 40 more rows
```

```
#I created a new column using mutate which was a function of two separate variables in my dataset.
fulldata %>% mutate(activecal_permi = `Active Calories (kcal)`/`Distance (mi)`)%>%mutate(mean_activecal
```

```
## # A tibble: 50 x 17
##    Date                `Active Calorie~ `Distance (mi)` `Resting Calori~
##    <dttm>                         <dbl>           <dbl>            <dbl>
##  1 2020-01-01 00:00:00             207.            2.23            1505.
```

```
##  2 2020-01-01 00:00:00            207.            2.23            1505.
##  3 2020-01-02 00:00:00            117.            0.692           1557.
##  4 2020-01-03 00:00:00            462.            3.01            1459.
##  5 2020-01-04 00:00:00            475.            2.63            1589.
##  6 2020-01-05 00:00:00             48.0           2.10            1496.
##  7 2020-01-06 00:00:00            563.            4.22            1530.
##  8 2020-01-07 00:00:00            540.            2.91            1539.
##  9 2020-01-08 00:00:00            236.            1.33            1455.
## 10 2020-01-09 00:00:00            272.            2.13            1479.
## # ... with 40 more rows, and 13 more variables: `Steps (count)` <dbl>,
## #   Type <chr>, Time <chr>, Duration <dttm>, Distance <dbl>, `Average Heart
## #   Rate` <dbl>, `Max Heart Rate` <dbl>, `Average Pace` <dttm>, `Average
## #   Speed` <dbl>, `Active Energy` <dbl>, `Total Energy` <dbl>,
## #   activecal_permi <dbl>, mean_activecal_permi <dbl>
```

```
#I used summarize to get the mean value for each numeric variable in the dataset, and applied group_by
fulldata%>%group_by(Type)%>% summarise(mean_activecalories=mean(`Active Calories (kcal)`, na.rm = T), me
```

```
## # A tibble: 6 x 13
##   Type  mean_activecalo~ mean_distance mean_resting mean_steps
##   <chr>            <dbl>         <dbl>        <dbl>      <dbl>
## 1 Cros~             746.          5.19        1543.     10928.
## 2 Func~             423.          3.48        1559.      7590.
## 3 High~             862.          6.33        1612.     13620.
## 4 Indo~             448.          3.82        1530.      7923.
## 5 Outd~             128.          2.38        1193.      5326.
## 6 Yoga              451.          2.89        1505.      6120.
## # ... with 8 more variables: mean_duration <dttm>, mean_workoutdistance <dbl>,
## #   mean_avghrtrate <dbl>, mean_maxhrtrate <dbl>, mean_avgpace <dttm>,
## #   mean_avgspeed <dbl>, mean_activeenergy <dbl>, mean_totalenergy <dbl>
```

```
#I used summarize and group_by to get the maximum average heart rate and max heart rate to see which wo
fulldata%>%group_by(Type)%>% summarise(max_avgheartrate=max(`Average Heart Rate`, na.rm = T), max_maxhea
```

```
## # A tibble: 6 x 3
##   Type                              max_avgheartrate max_maxheartrate
##   <chr>                                        <dbl>            <dbl>
## 1 Indoor Running                                186.              201
## 2 Cross Training                                170.              199
## 3 High Intensity Interval Training              166.              190
## 4 Functional Strength Training                  159.              188
## 5 Yoga                                          127.              173
## 6 Outdoor Walking                               114.              146
```

```
#I used group_by to investigate the mean and standard deviation of the durations of workouts based on t
fulldata%>%group_by(Type)%>%summarise(mean_duration=mean(Duration, na.rm = T), sd_duration=sd(Duration,
```

```
## # A tibble: 6 x 3
##   Type                            mean_duration       sd_duration
##   <chr>                           <dttm>                    <dbl>
## 1 Cross Training                  1899-12-31 00:33:13        496.
## 2 Functional Strength Training    1899-12-31 00:09:49        305.
```

```
## 3 High Intensity Interval Training 1899-12-31 00:36:10        427.
## 4 Indoor Running                   1899-12-31 00:26:44        484.
## 5 Outdoor Walking                  1899-12-31 00:18:51        320.
## 6 Yoga                             1899-12-31 00:36:55        504.
```

```
#I used arrange and group by to determine the minimum resting calories burned in a day based on workout
fulldata%>%group_by(Type)%>%summarise(mean_resting=mean(`Resting Calories (kcal)`, na.rm = T ))%>%arran
```

```
## # A tibble: 6 x 2
##   Type                              mean_resting
##   <chr>                                    <dbl>
## 1 High Intensity Interval Training         1612.
## 2 Functional Strength Training             1559.
## 3 Cross Training                           1543.
## 4 Indoor Running                           1530.
## 5 Yoga                                     1505.
## 6 Outdoor Walking                          1193.
```

```
#I explored what the minimum were for all of my variables, and then grouped the minimums by workout typ

fulldata %>% summarise (min(Distance, na.rm = T), min(`Distance (mi)`, na.rm = T), min(`Active Calories
```

```
## # A tibble: 1 x 12
##   `min(Distance, ~ `min(\`Distance~ `min(\`Active C~ `min(\`Active E~
##              <dbl>            <dbl>            <dbl>            <dbl>
## 1           0.0892            0.692             48.0               25
## # ... with 8 more variables: `min(\`Resting Calories (kcal)\`, na.rm =
## #   T)` <dbl>, `min(\`Steps (count)\`, na.rm = T)` <dbl>, `min(Duration, na.rm
## #   = T)` <dttm>, `min(\`Average Heart Rate\`, na.rm = T)` <dbl>,
## #   `min(\`Average Pace\`, na.rm = T)` <dttm>, `min(\`Average Speed\`, na.rm =
## #   T)` <dbl>, `min(\`Max Heart Rate\`, na.rm = T)` <dbl>, `min(\`Total
## #   Energy\`, na.rm = T)` <dbl>
```

```
fulldata %>%group_by(Type)%>% summarise (min(Distance, na.rm = T), min(`Distance (mi)`, na.rm = T), min
```

```
## # A tibble: 6 x 13
##   Type  `min(Distance, ~ `min(\`Distance~ `min(\`Active C~ `min(\`Active E~
##   <chr>            <dbl>            <dbl>            <dbl>            <dbl>
## 1 Cros~           0.0892             2.54             554.             134.
## 2 Func~              Inf             2.23             207.              25
## 3 High~              Inf             3.73             742.             267.
## 4 Indo~             2.03             0.692             48.0             96
## 5 Outd~             0.929            1.26              69.2             38.9
## 6 Yoga               Inf             1.14             236.             107
## # ... with 8 more variables: `min(\`Resting Calories (kcal)\`, na.rm =
## #   T)` <dbl>, `min(\`Steps (count)\`, na.rm = T)` <dbl>, `min(Duration, na.rm
## #   = T)` <dttm>, `min(\`Average Heart Rate\`, na.rm = T)` <dbl>,
## #   `min(\`Average Pace\`, na.rm = T)` <dttm>, `min(\`Average Speed\`, na.rm =
## #   T)` <dbl>, `min(\`Max Heart Rate\`, na.rm = T)` <dbl>, `min(\`Total
## #   Energy\`, na.rm = T)` <dbl>
```

```
#I determined the number of distinct observations per variable.
fulldata%>%summarise_all(n_distinct)
```

```
## # A tibble: 1 x 15
##    Date `Active Calorie~ `Distance (mi)` `Resting Calori~ `Steps (count)`  Type
##   <int>            <int>           <int>            <int>           <int> <int>
## 1    49               48              48               48              48     6
## # ... with 9 more variables: Time <int>, Duration <int>, Distance <int>,
## #   `Average Heart Rate` <int>, `Max Heart Rate` <int>, `Average Pace` <int>,
## #   `Average Speed` <int>, `Active Energy` <int>, `Total Energy` <int>
```

```
#I created a correlation matrix including all my numeric variable.
fulldata2<-fulldata%>% na.omit %>% select_if(is.numeric)
fulldata2 %>%cor(fulldata2)
```

```
##                          Active Calories (kcal) Distance (mi)
## Active Calories (kcal)             1.00000000    0.87256053
## Distance (mi)                      0.87256053    1.00000000
## Resting Calories (kcal)            0.48049164    0.48199047
## Steps (count)                      0.86608114    0.99629014
## Distance                           0.06401108    0.23239642
## Average Heart Rate                 0.33673054    0.25222280
## Max Heart Rate                     0.43333642    0.30241062
## Average Speed                     -0.24824588   -0.01871333
## Active Energy                      0.70883346    0.60930317
## Total Energy                       0.71405887    0.61721736
##                          Resting Calories (kcal) Steps (count)   Distance
## Active Calories (kcal)                 0.4804916    0.86608114 0.06401108
## Distance (mi)                          0.4819905    0.99629014 0.23239642
## Resting Calories (kcal)                1.0000000    0.49671667 0.22083356
## Steps (count)                          0.4967167    1.00000000 0.20668912
## Distance                               0.2208336    0.20668912 1.00000000
## Average Heart Rate                     0.4767060    0.22514191 0.66334088
## Max Heart Rate                         0.5390049    0.27905588 0.55068193
## Average Speed                          0.1589618   -0.04130961 0.88138301
## Active Energy                          0.4640628    0.59359783 0.59060353
## Total Energy                           0.4505514    0.60226776 0.57432657
##                          Average Heart Rate Max Heart Rate Average Speed
## Active Calories (kcal)           0.3367305      0.4333364   -0.24824588
## Distance (mi)                    0.2522228      0.3024106   -0.01871333
## Resting Calories (kcal)          0.4767060      0.5390049    0.15896176
## Steps (count)                    0.2251419      0.2790559   -0.04130961
## Distance                         0.6633409      0.5506819    0.88138301
## Average Heart Rate               1.0000000      0.9413712    0.61005704
## Max Heart Rate                   0.9413712      1.0000000    0.46008176
## Average Speed                    0.6100570      0.4600818    1.00000000
## Active Energy                    0.6771266      0.7013614    0.25065730
## Total Energy                     0.6367313      0.6646166    0.21797167
##                          Active Energy Total Energy
## Active Calories (kcal)       0.7088335    0.7140589
## Distance (mi)                0.6093032    0.6172174
## Resting Calories (kcal)      0.4640628    0.4505514
## Steps (count)                0.5935978    0.6022678
```

```
## Distance                  0.5906035    0.5743266
## Average Heart Rate        0.6771266    0.6367313
## Max Heart Rate            0.7013614    0.6646166
## Average Speed             0.2506573    0.2179717
## Active Energy             1.0000000    0.9950754
## Total Energy              0.9950754    1.0000000
```

```
#I retrieved the last recording for each workout type within the dataframe.
#I then arranged by Max heart rate.
fulldata%>%group_by(Type)%>%summarise_all(last)%>%arrange(desc(`Max Heart Rate`))
```

```
## # A tibble: 6 x 15
##   Type  Date                `Active Calorie~ `Distance (mi)` `Resting Calori~
##   <chr> <dttm>                         <dbl>           <dbl>            <dbl>
## 1 Indo~ 2020-02-18 00:00:00            788.            7.85            1613.
## 2 Cros~ 2020-02-13 00:00:00            835.            6.33            1629.
## 3 Func~ 2020-01-18 00:00:00            588.            5.59            1582.
## 4 High~ 2020-02-26 00:00:00            788.            7.85            1613.
## 5 Yoga  2020-01-31 00:00:00            413.            1.14            1537.
## 6 Outd~ 2020-02-25 00:00:00            107.            1.26             685.
## # ... with 10 more variables: `Steps (count)` <dbl>, Time <chr>,
## #   Duration <dttm>, Distance <dbl>, `Average Heart Rate` <dbl>, `Max Heart
## #   Rate` <dbl>, `Average Pace` <dttm>, `Average Speed` <dbl>, `Active
## #   Energy` <dbl>, `Total Energy` <dbl>
```

```
#I created a proportion table for the categorical variable workout Type.
fulldata%>%count(Type)%>%mutate(prop=prop.table(n))
```

```
## # A tibble: 6 x 3
##   Type                              n  prop
##   <chr>                         <int> <dbl>
## 1 Cross Training                    9  0.18
## 2 Functional Strength Training      3  0.06
## 3 High Intensity Interval Training 11  0.22
## 4 Indoor Running                   16  0.32
## 5 Outdoor Walking                   3  0.06
## 6 Yoga                              8  0.16
```

The results of summary analysis displayed a lot of interesting results. When specficially looking at High Intensity Interval Training and arranging by active calories it was seen that the largest amount of active calories obtained was 985.565 while the mean active calories for high intensity interval training was 861.991. I then got a glimpse of cardiovascular output by selecting Average heart rate and max heart rate and grouping by type; there was an association between average heart rate and max heart rate which is to be expected. To create a new varaible that was a function of two other variables in my dataset I used mutate, and it resulted in a column depicting the calorie amount burned per mile. I computed the average of that calorie per mile column to see that, on average, I burn 137.585 calories per mile during my workouts. Further, to get a holisti view of the means for each number variables I calculated the averages while grouping my Type.

To extend my analysis on cardiovasular output, I then looked at the maximum average heart rate and maximum max heart rate for each workout. The result was that indoor running contained the maximum in both categories. Using mutate I saw that the standard deviations for the duration for each workout were actually quite large, which was an interesting finding. One extremely interesting finding was that the highest mean values for resting calories burned in a day by workout type was High Intensity Interval Training even

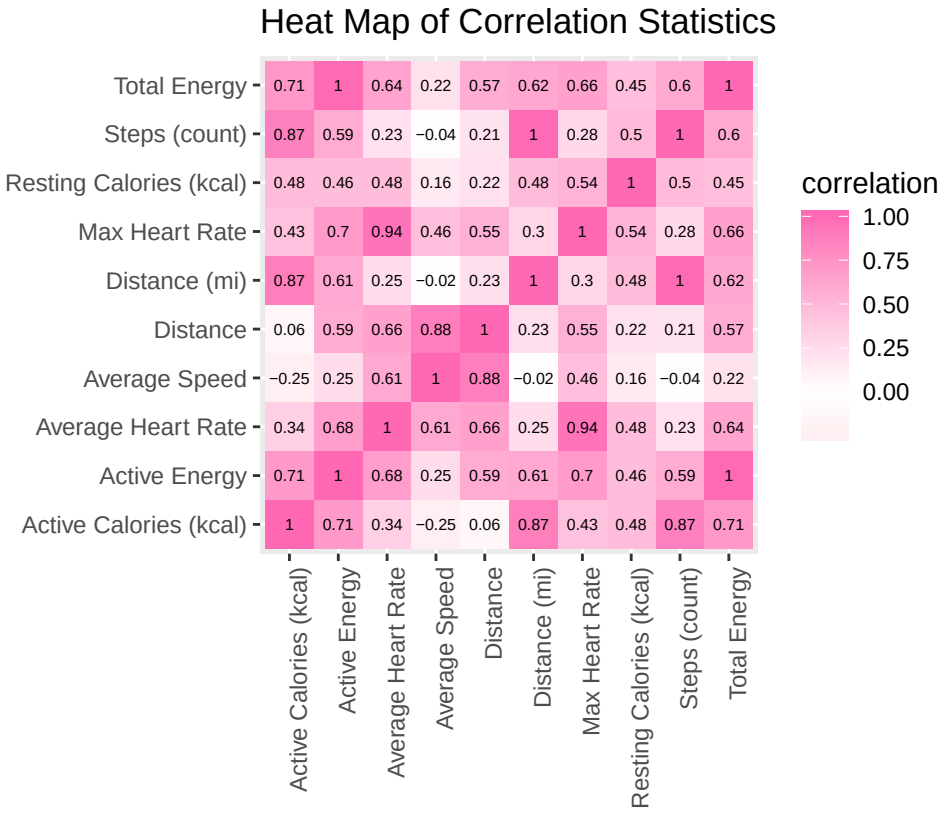
thouohg, as seen earlier, the highest max and average heart rate was seen by indoor running workouts. This finding supports claims that HIIT workouts create a higher after-burn than other workouts. I also investigated the minmums for each value based on workout and saw that there were great variations between the different minimums per variable. I looked at the last workout I did for each type and saw that during my last outdoor walk my heart rate only reached 146bpm. The correlation matrix, made into a visualize in the form of a heat map in the next section, depicted that many variables had decent-high correlations with each other. Finally, by analzying the proportion table created by workout type it was observed that my highest proportion of workout type was indoor running. This was expected as I have been working on training for a 10k run.

Visualizations

```
#I made a heat map depicting the correlation statistics.
heatmap<-cor(fulldata2)%>%as.data.frame%>%
rownames_to_column%>%
pivot_longer(-1,names_to="name",values_to="correlation")
head(heatmap)
```

```
## # A tibble: 6 x 3
##   rowname                name                     correlation
##   <chr>                  <chr>                          <dbl>
## 1 Active Calories (kcal) Active Calories (kcal)       1
## 2 Active Calories (kcal) Distance (mi)                0.873
## 3 Active Calories (kcal) Resting Calories (kcal)      0.480
## 4 Active Calories (kcal) Steps (count)                0.866
## 5 Active Calories (kcal) Distance                     0.0640
## 6 Active Calories (kcal) Average Heart Rate           0.337
```

```
heatmap%>%ggplot(aes(rowname,name,fill=correlation))+
geom_tile()+
scale_fill_gradient2(low="pink",mid="white",high="hot pink")+
geom_text(aes(label=round(correlation,2)),color = "black", size = 2)+
theme(axis.text.x = element_text(angle = 90, hjust=1))+
coord_fixed()+xlab("")+ylab("")+ggtitle("Heat Map of Correlation Statistics")
```

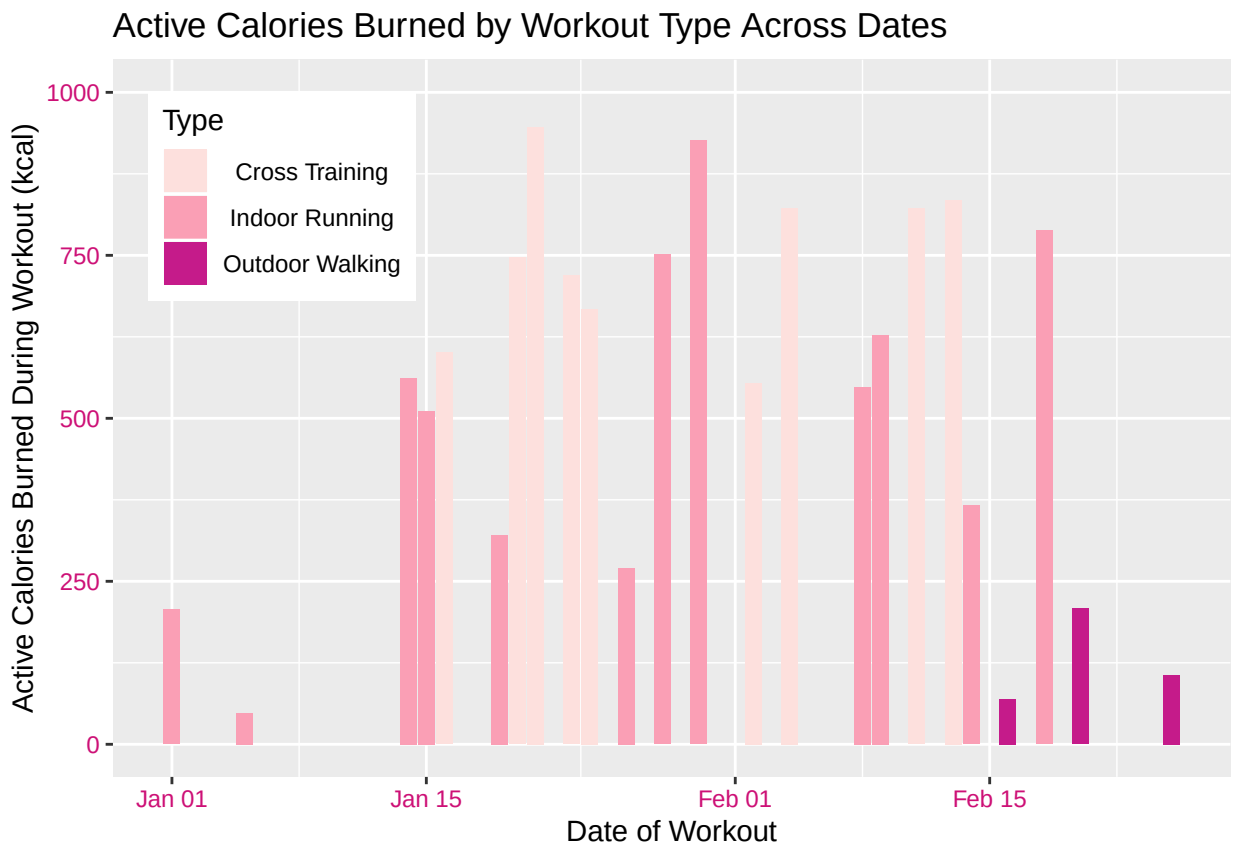


The heat map visulaization communicates that many of the variables have a good to strong correlation with each other. This is seen by the darker color pinks as those indicate a higher correlation value. The highest correlating variable, that are not redundant, are step count and miles. Further, the lowest correlating variables are distance and average speed. While the high correlation variables are expected to have that relationship, distance and average speed is a relationship that is more variable depending on the person. As for this data, reflecting my workout statistics, this correlation indicates that regardless of my mile distance my average speed is relatively static. Overall, this correlation heat map was informative in showing many relationships between variables.

```
library(RColorBrewer)
fulldata %>% drop_na()%>% ggplot(aes(`Steps (count)`, `Distance (mi)`,color=Type) )+
geom_point(aes(size=Duration))+ggtitle("Distance vs Steps by Workout Duration and Workout Type")+labs(x=
```

A scatterplot was created to visualize total distance traveled with the number of steps taken at each plot point. Further, the plot points were categorized by workout type to see if there were any assocaitions with increased steps or total distance based on workout, and workout duration was mapped to the size of each plot point. This visualization revealed that outdoor walking produced the least amount of distance and step totals while both cross training and indoor running mapped more fully from high to low in both variable. As one would expect, the larger sized points, indicating a longer workout, produce distance and step totals that are greater than smaller duration workouts. Ultimately, this visualization demonstrated that there was a clear correlation between Total Steps Taken and Distance Traveled which is logical.

```
library(RColorBrewer)
fulldata %>% drop_na%>% ggplot(aes(x = Date, y = `Active Calories (kcal)`, fill=Type))+
geom_bar(stat="summary",fun.y="mean", position="dodge")+labs(x="Date of Workout", y="Active Calories Bu
```



A bar chart was created to visualize the relationship between active calories burned by type of workout when mapping by date. First analyzing the overall shape of the bars collectively, it is apparent that at the beginning of the year my workouts started off with less intensity, peaked in intenisty half-way towards the beginning of february, and again decreased towards the end of febraury. This is interesting becuase it could suggest that a plateau in fitness could be predicted towards the end of february as my body may have adapted to the workouts I had typically been doing. Further, the categorical variable of workout type reveals that outdoor walking consistently produced low amounts of active calories burned while cross training produced active calorie burns that were consistently high. Indoor running seems to also produce active calorie counts that are substantial.

Dimensionality Reduction

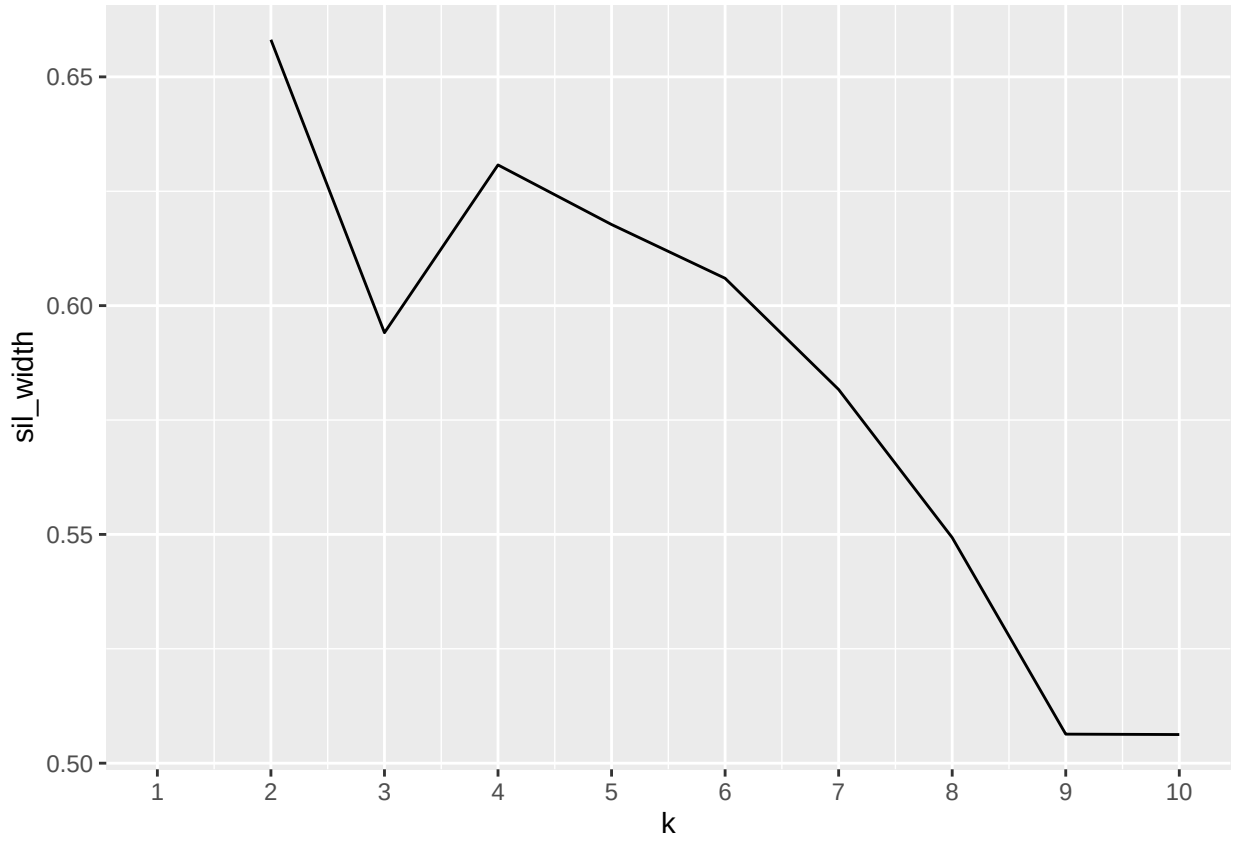
```
pam<-fulldata%>%select(`Steps (count)`, `Resting Calories (kcal)`, `Active Calories (kcal)`)%>%pam(2)

pam
```

```
## Medoids:
##      ID Steps (count) Resting Calories (kcal) Active Calories (kcal)
## [1,]  4          5598                1458.973                461.794
## [2,] 27         12896                1588.052                751.283
## Clustering vector:
##  [1] 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 2 1 2 2 2 2 1 1 1 2 2 2 2 1 1 2 2 2 1 1 2 2
## [39] 2 2 1 1 1 2 2 2 2 2 1 2
```

```
## Objective function:
##    build     swap
## 2419.933 1768.925
##
## Available components:
##  [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
##  [6] "clusinfo"   "silinfo"    "diss"       "call"       "data"
```

```
sil_width<-vector()
for(i in 2:10){
pam_fit<- fulldata%>%select(`Steps (count)`, `Resting Calories (kcal)`, `Active Calories (kcal)`)%>%pam
sil_width[i] <- pam_fit$silinfo$avg.width
}
ggplot()+geom_line(aes(x=1:10,y=sil_width))+scale_x_continuous(name="k",breaks=1:10)
```

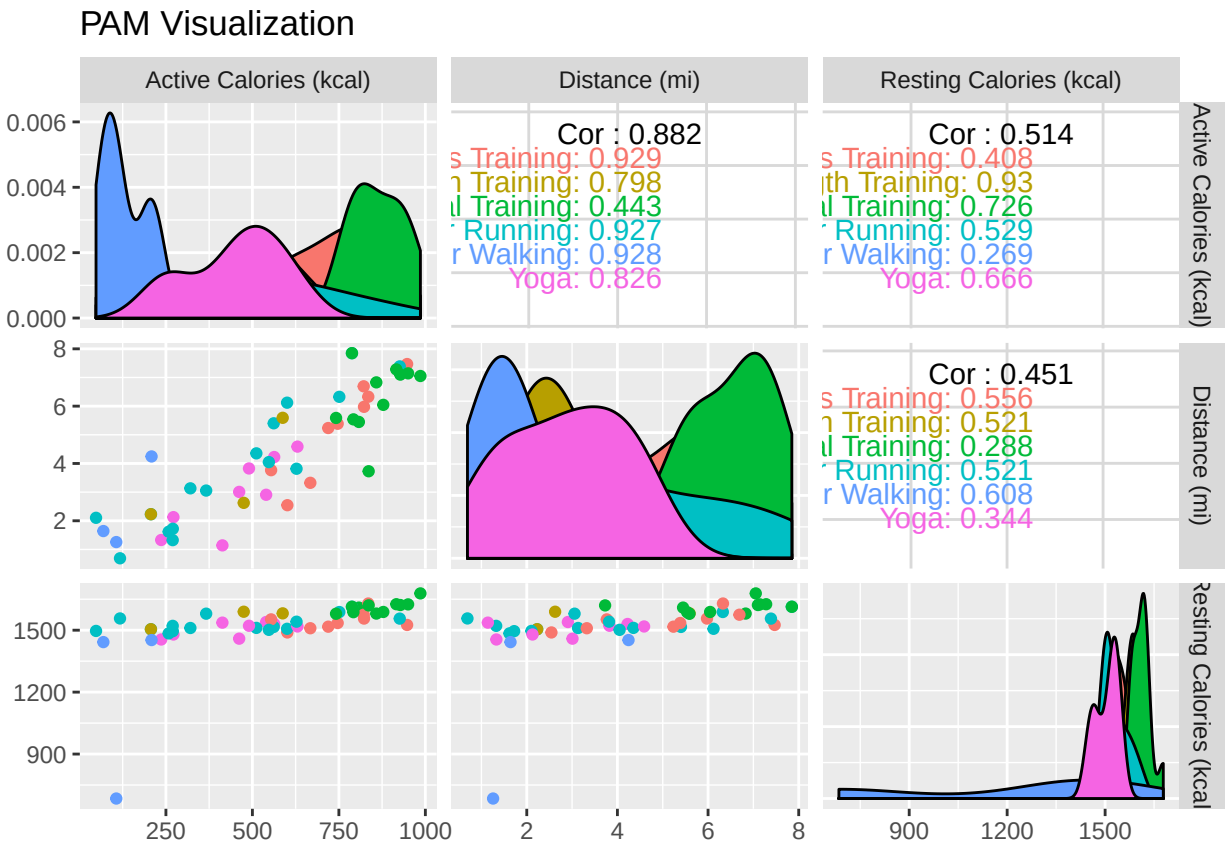



```
pamfinal<-fulldata%>% mutate(cluster=as.factor(pam$clustering))
confmat<-pamfinal%>%group_by(Type)%>%count(cluster)%>%arrange(desc(n))%>%
pivot_wider(names_from="cluster",values_from="n",values_fill = list('n'=0))
confmat
```

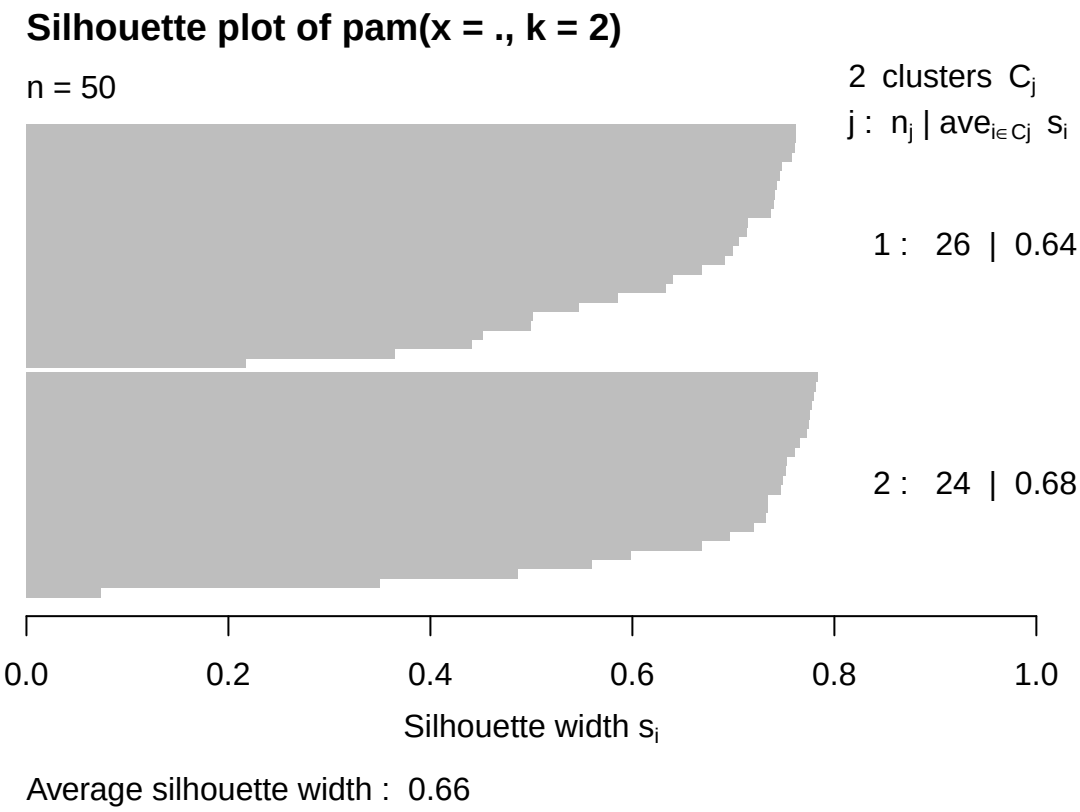
```
## # A tibble: 6 x 3
## # Groups:   Type [6]
##   Type                          `1`   `2`
##   <chr>                       <int> <int>
## 1 Indoor Running                 11     5
```

```
## 2 High Intensity Interval Training     1    10
## 3 Yoga                                 7     1
## 4 Cross Training                       3     6
## 5 Functional Strength Training         2     1
## 6 Outdoor Walking                      2     1
```

```
ggpairs(pamfinal, columns = 2:4, aes(color=Type))+ggtitle("PAM Visualization")
```




```
plot(pam, which=2)
```



A dimensional reduction using PAM was conducted by clustering the numeric variables "Active Calories (kcal)", "Distance (mi)", and "Resting Calories (kcal)". To determine the number of clusters to choose, the average silhouette width was calculated and visualized. Upon visualization, it was determined that clustering by two clusters would produce the maximum shilhouette width. After analysis, the two variables with the highest correlation were distance and active calories. This is logical becuase as distance increases, the amount of calories burned activiely should increase as well. Further, the two variables with the weakest correlation were Distance and Resting Calories.

To analyze the average shilhouette width, a plot was created to visualize with PAM when clutering by 2. The average shilhouette width was 0.66. Interpretating this value, it can be concluded that a reasonable structure was found when conducting the demintional reduction.