Alexander Rasho

The attached dataset was collected as part of a long running study conducted by researchers at CSU Fullerton's College of Health and Human Development. The study collects data relating to the physical health of college students.

This project has two objectives:

1. Perform an exploratory data analysis of the dataset.
2. One of the variables is Total Fitness Factor Score (column with header "FF") which is computed using a formula (hidden from you). Come up with an approximation of Total Fitness Factor Score for a subject using only the other available variables.

Exploratory data analysis

Exploratory data analysis is:

1. **Generating summary statistics (mean, median, any outliers, any missing data points) for a variable**

Mean and Median

```
> summary(mydata)
     Idnum             Date               Sex                 Age              Ht              Wt             RF 2            RF 3
 Length:6276        Length:6276        Length:6276        Min.   :18.0   Min.   :55.00   Min.   : 82.0   Min.   :1.000   Min.   :1.000
 Class :character   Class :character   Class :character   1st Qu.:19.0   1st Qu.:63.00   1st Qu.:123.3   1st Qu.:1.000   1st Qu.:2.000
 Mode  :character   Mode  :character   Mode  :character   Median :19.0   Median :66.00   Median :142.0   Median :1.000   Median :2.000
                                                          Mean   :19.6   Mean   :66.21   Mean   :147.4   Mean   :2.695   Mean   :2.443
                                                          3rd Qu.:20.0   3rd Qu.:69.00   3rd Qu.:166.4   3rd Qu.:5.000   3rd Qu.:3.000
                                                          Max.   :25.0   Max.   :78.00   Max.   :329.0   Max.   :5.000   Max.   :6.000

      RF 5         BIA_percent_Fat      SF 1            SF 2            SF 3         Waist               FF              RGM              L
 Min.   :0.0000   Mode:logical    Min.   : 2.00   Min.   : 4.00   Min.   : 4.00   Mode:logical   Min.   :-7.000   Min.   :14.00   Min.
 1st Qu.:0.0000   NA's:6276       1st Qu.:10.00   1st Qu.:13.50   1st Qu.:13.80   NA's:6276      1st Qu.: 1.500   1st Qu.:30.00   1st Qu
 Median :0.0000                   Median :16.00   Median :19.50   Median :20.25                  Median : 4.000   Median :36.00   Median
 Mean   :0.1636                   Mean   :16.68   Mean   :21.14   Mean   :21.30                  Mean   : 3.573   Mean   :39.09   Mean
 3rd Qu.:0.0000                   3rd Qu.:21.90   3rd Qu.:27.50   3rd Qu.:27.30                  3rd Qu.: 6.000   3rd Qu.:48.00   3rd Qu
 Max.   :6.0000                   Max.   :50.00   Max.   :60.00   Max.   :58.00                  Max.   :12.000   Max.   :82.00   Max.
                                  NA's   :1596    NA's   :1596    NA's   :1596
       TA              PB              SBP              DBP             HR rest          Stages          PL 1             HR 1             RPE
 Min.   :18   Min.   :734.0   Min.   : 82.0   Min.   : 46.00   Min.   :46.00   Min.   :2.000   Min.   : 30.00   Min.   :  61.0   Min.
 1st Qu.:22   1st Qu.:755.0   1st Qu.:110.0   1st Qu.: 62.00   1st Qu.:72.00   1st Qu.:3.000   1st Qu.: 30.00   1st Qu.: 106.0   1st Qu.
 Median :23   Median :757.0   Median :118.0   Median : 70.00   Median :72.00   Median :3.000   Median : 50.00   Median : 117.0   Median
 Mean   :23   Mean   :757.5   Mean   :117.7   Mean   : 69.92   Mean   :71.62   Mean   :3.016   Mean   : 40.53   Mean   : 118.6   Mean
 3rd Qu.:24   3rd Qu.:760.0   3rd Qu.:125.0   3rd Qu.: 78.00   3rd Qu.:72.00   3rd Qu.:3.000   3rd Qu.: 50.00   3rd Qu.: 130.0   3rd Qu.
 Max.   :28   Max.   :772.0   Max.   :170.0   Max.   :100.00   Max.   :90.00   Max.   :4.000   Max.   :100.00   Max.   :1352.0   Max.

      HR 2            RPE 2            PL 3            HR 3            RPE 3            FF_1
 Min.   : 14.0   Min.   : 3.00   Min.   : 70   Min.   :109.0   Min.   : 6.00   Min.   :11.00
 1st Qu.:130.0   1st Qu.:10.00   1st Qu.: 70   1st Qu.:157.0   1st Qu.:13.00   1st Qu.:31.00
 Median :139.0   Median :12.00   Median :130   Median :164.0   Median :15.00   Median :37.00
 Mean   :140.4   Mean   :11.51   Mean   :128   Mean   :164.3   Mean   :14.48   Mean   :37.26
 3rd Qu.:150.0   3rd Qu.:13.00   3rd Qu.:175   3rd Qu.:172.0   3rd Qu.:16.00   3rd Qu.:44.00
 Max.   :195.0   Max.   :20.00   Max.   :200   Max.   :205.0   Max.   :20.00   Max.   :62.00
                                 NA's   :380   NA's   :380     NA's   :393
>
```

Outliers

HT

```
> boxplot(mydata$Ht) $out
numeric(0)
```

WT

```
> boxplot(mydata$wt) $out
  [1] 265.0 248.0 236.1 257.3 245.6 282.0 247.0 250.0 233.0 268.0 244.6 238.9 231.2 267.0 268.1
 [25] 250.9 247.1 262.0 278.0 233.0 245.0 243.3 252.6 235.6 241.6 260.0 239.8 251.8 236.4 258.4
 [49] 240.0 262.6 278.4 248.6 329.0 257.8 248.2 312.0 246.6 253.8 248.0 278.0 253.6 240.8 266.8
 [73] 260.6 265.2 280.0 245.8 232.0 247.6 245.4 248.6 247.2 232.0 246.6 239.8 237.6 286.2 232.8
 [97] 231.2 237.2 242.2 295.5 259.0 293.4 234.0 241.5 252.2 238.6 238.8 255.0 250.8 233.4 246.2
[121] 234.6 234.2 271.0 244.0
```

SF 1

```
> boxplot(mydata$'SF 1') $out
  [1] 49.5 50.0 40.0 40.0 46.5 40.0 40.0 40.0 45.0 45.0 42.0 45.0 45.0 4
 [31] 43.5 47.0 40.0 40.0 40.0 50.0 50.0 50.0 40.0 50.0 50.0 40.0 45.0 4
 [61] 40.0
>
```

SF 2

```
> boxplot(mydata$'SF 2') $out
  [1] 49.5 52.5 54.0 53.0 54.5 49.0 53.5 58.0 50.0 50.0 50.0 49.5 50.5 52.0 57.
 [31] 49.0 49.0 56.0 50.0 60.0
>
```

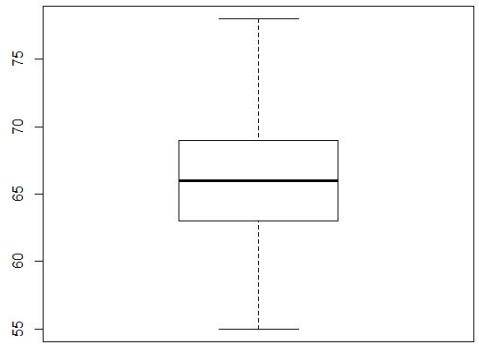
SF 3

```
> boxplot(mydata$'SF 3') $out
  [1] 50.0 50.0 48.0 50.0 48.0 48.0 51.2 53.0 48.2 49.0 52.0
 [31] 50.0 48.0 48.0 52.0 52.5 50.0 48.0 52.0 48.0
>
```

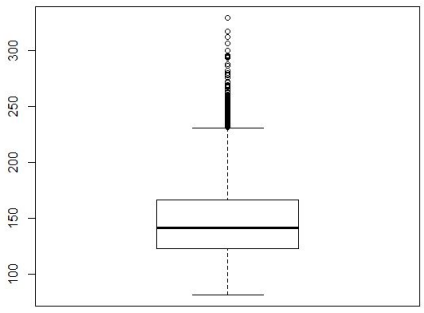
Missing data points

```
> sum(is.na(mydata))
[1] 18493
> colSums(is.na(mydata))
          Idnum            Date             Sex             Age              Ht              Wt            RF 2            RF 3
              0               0               0               0               0               0               0               0
BIA_percent_Fat           SF 1            SF 2            SF 3           Waist              FF             RGM             LGM
           6276            1596            1596            1596            6276               0               0               0
             PB             SBP             DBP        HR rest          Stages            PL 1            HR 1           RPE 1
              0               0               0               0               0               0               0               0
         RPE 2            PL 3            HR 3           RPE 3            FF_1
              0             380             380             393               0
> |
```

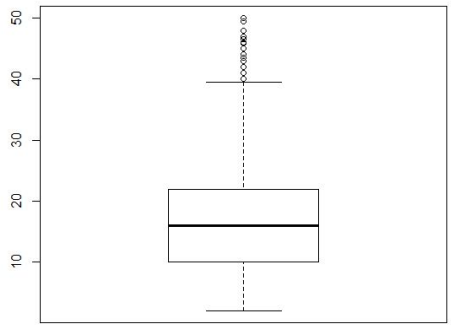
2. **Visualizing the values of a variable**
   Visualizing: Ht, Wt, SF1/2/3
   > boxplot(mydata$Ht)



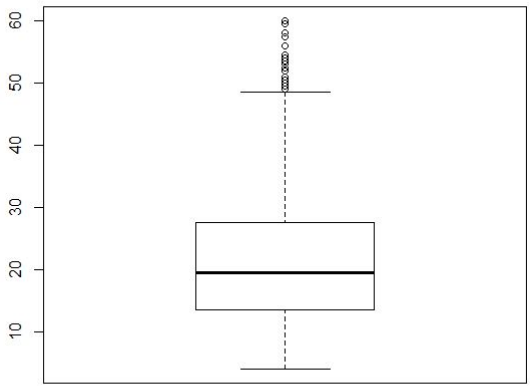
   > boxplot(mydata$Wt)



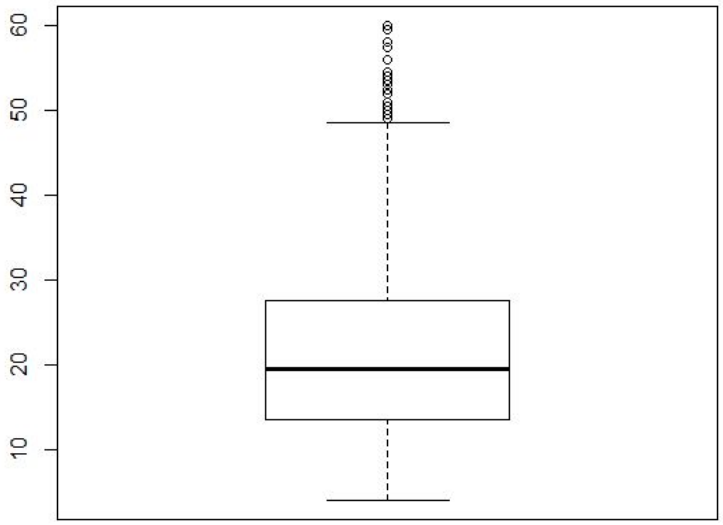
   > boxplot(mydata$'SF 1')



> boxplot(mydata$'SF 2')



> boxplot(mydata$'SF 3')

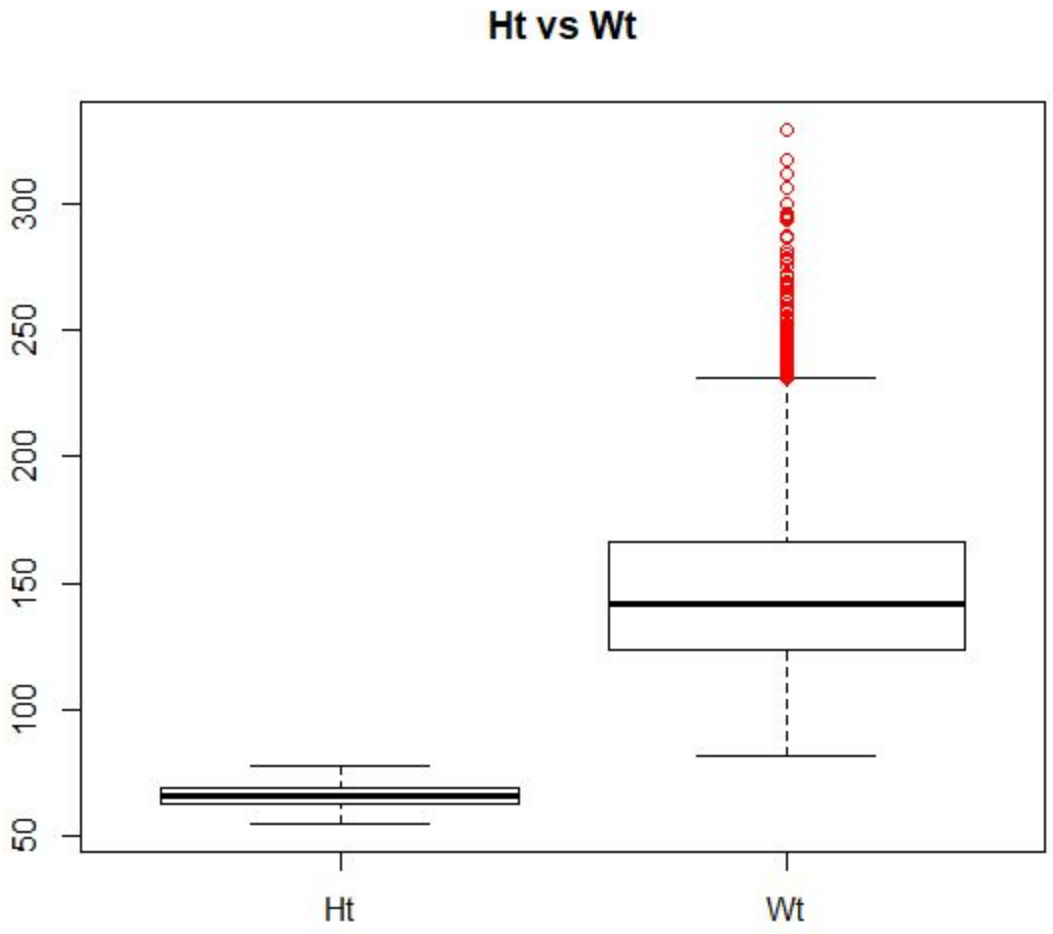


3. Visualizing the relationship between pairs of variables

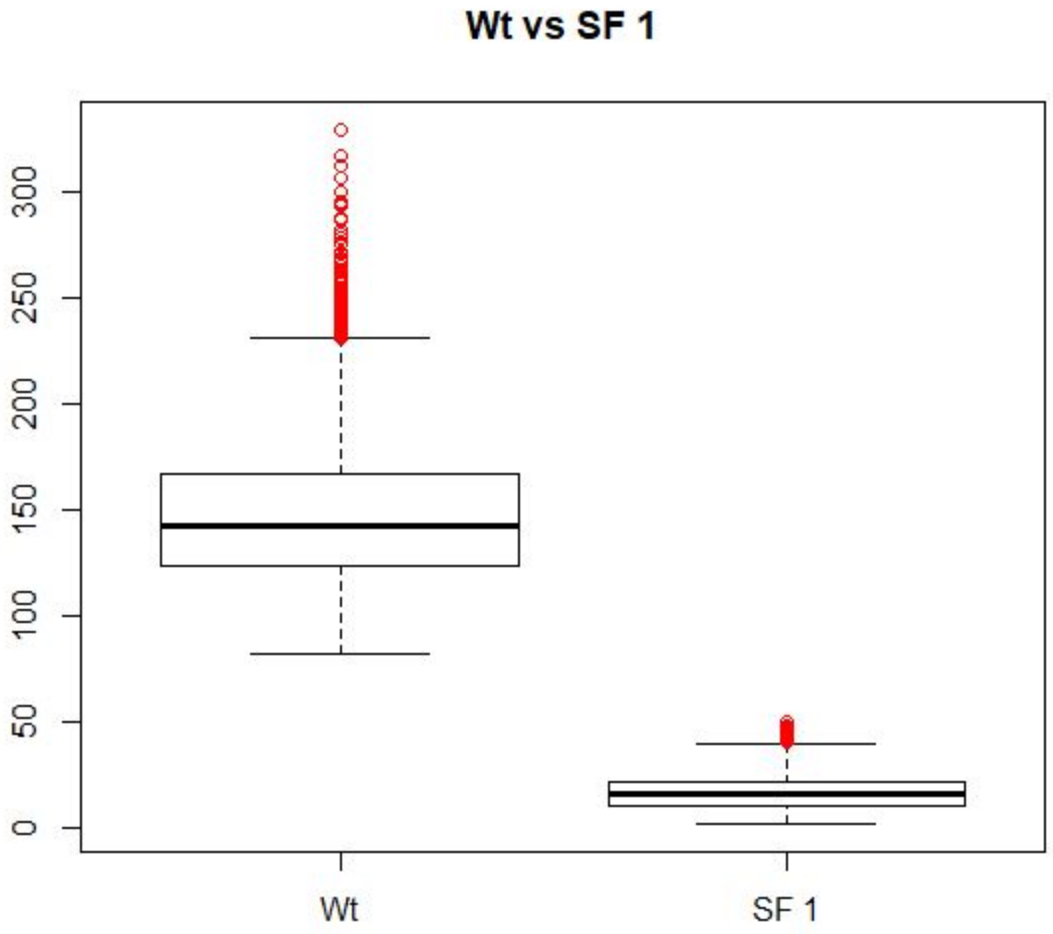
The dataset contains approximately 35 variables, so it is not expected that every variable or every pair of variables will be explored. For this project, it is sufficient to consider any 5 variables and any 5 pairs of variables.

Red circle is Outlier

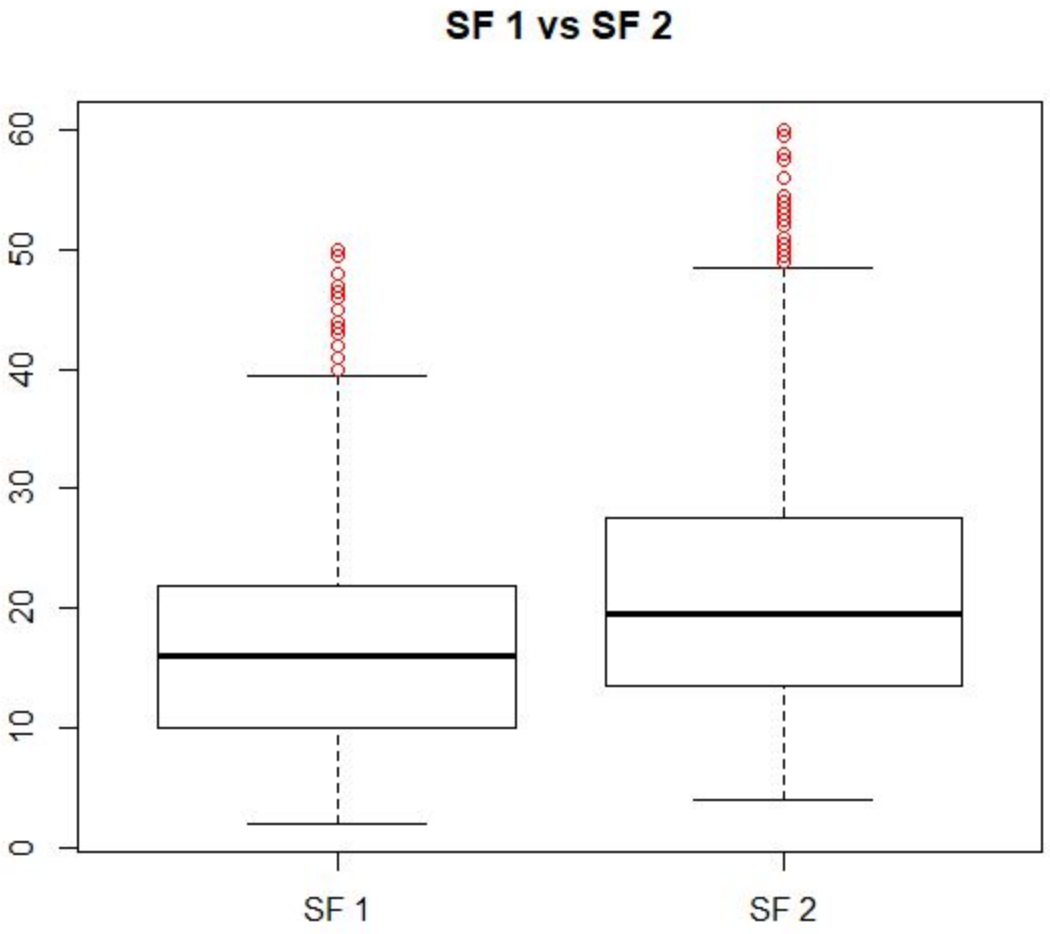
> boxplot(mydata$'Ht',mydata$'Wt', main = "Ht vs Wt",names=c('Ht', 'Wt'), outcol="red")



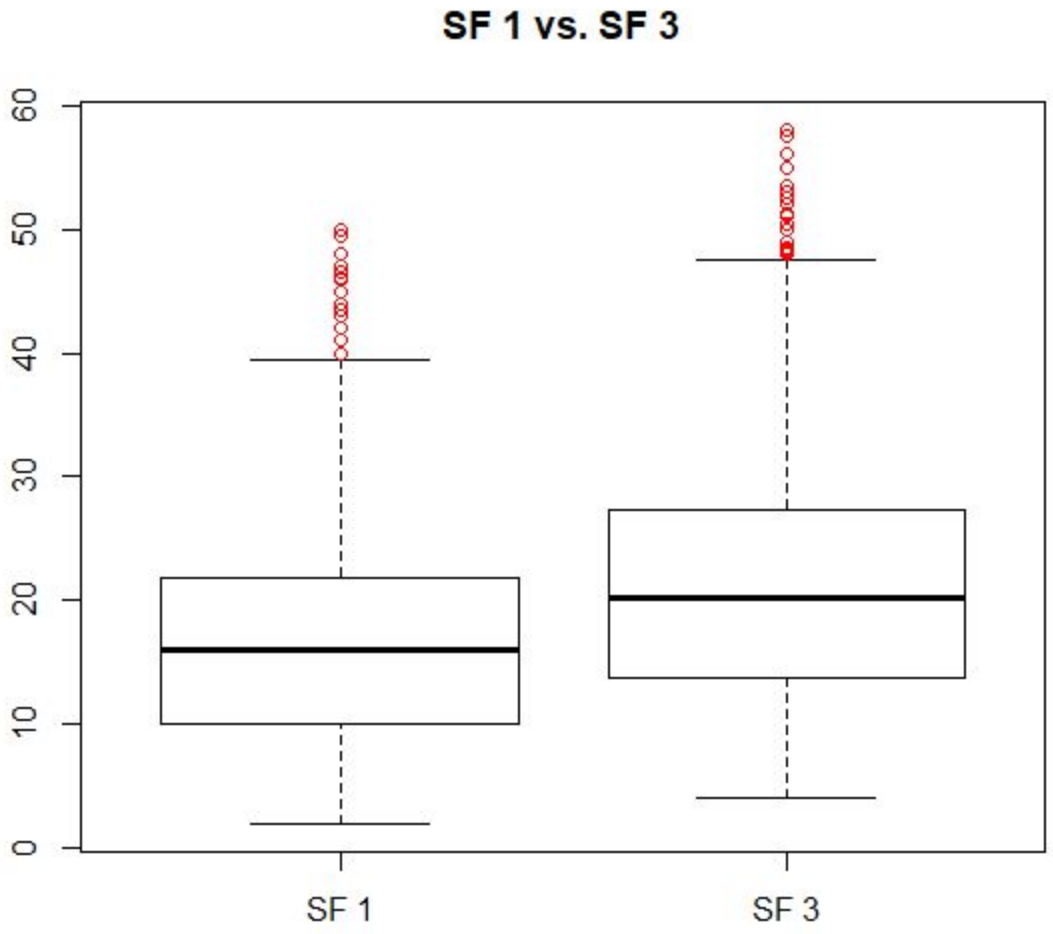
> boxplot(mydata$'Wt',mydata$'SF 1', main = "Wt vs SF 1",names=c('Wt', 'SF 1'), outcol="red")



> boxplot(mydata$'SF 1',mydata$'SF 2', main = "SF 1 vs SF 2",names=c('SF 1', 'SF 2'), outcol="red")

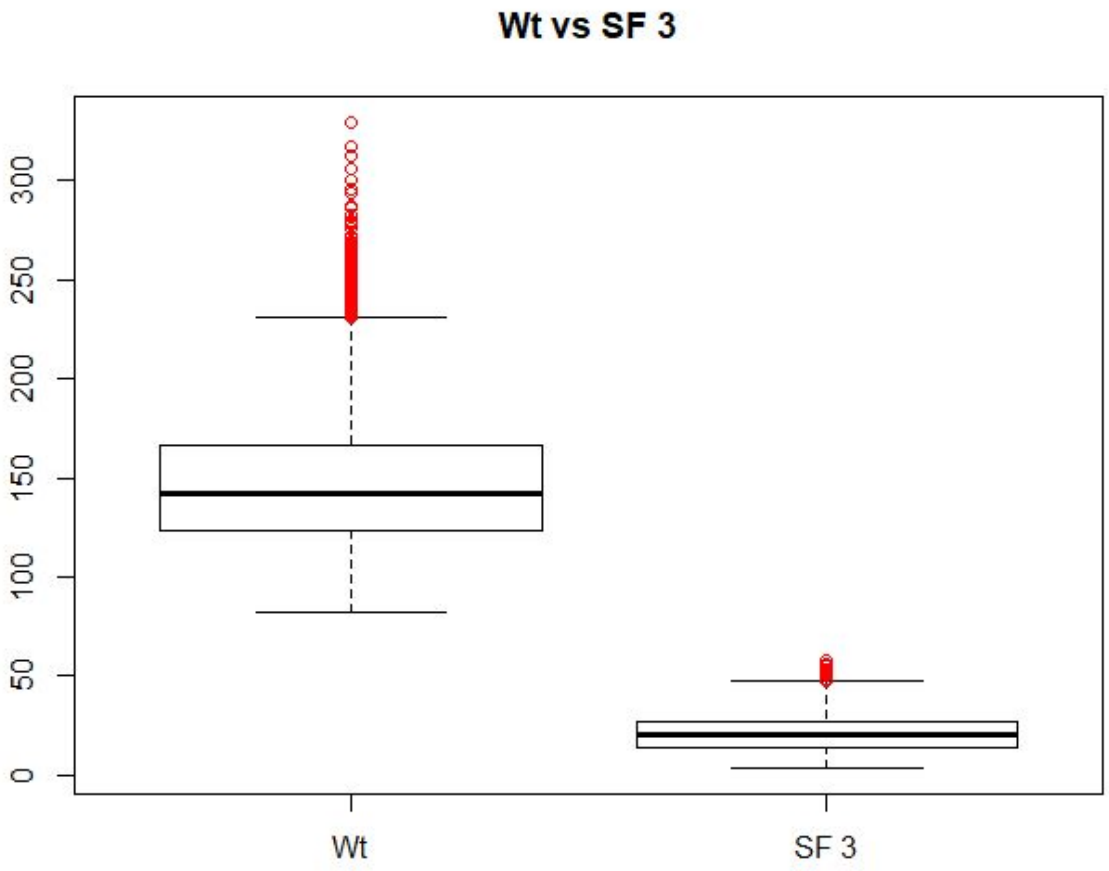


> boxplot(mydata$'SF 1',mydata$'SF 3', main = "SF 1 vs. SF 3",names=c('SF 1', 'SF 3'), outcol="red")



> boxplot(mydata$'SF 2',mydata$'SF 3', main = "SF 2 vs. SF 3",names=c('SF 2', 'SF 3'), outcol="red")

> boxplot(mydata$'Wt',mydata$'SF 3', main = "Wt vs SF 3",names=c('Wt', 'SF 3'), outcol="red")



Approximating Total Fitness Factor Score

The challenge is to identify which combination of the nearly 40 potential predictors will give the most accurate estimate and if transforming some of the variables will increase accuracy. You should explore at least 5 different combinations of predictors and choose the best combination. You will want to use some domain knowledge to pick the predictors. The attached Data dictionary file gives information on the units and meaning of the different columns. Note this file will *not* be read into the R code.

Evaluation

You should evaluate each combination of predictors using 10-fold cross-validation. Since you are estimating a continuous value, use root mean squared error (RMSE) as the evaluation metric. An example of creating folds for cross-validation using the cut function in R is here: https://stats.stackexchange.com/questions/61090/how-to-split-a-data-set-to-do-10-fold-cross-validation

The data will be searched for correlation and then transformed by using the standardizing method where the mean is 0 and the std is 1. The standardization should create a much better model.

```
> HRrest <- mydata$HR.rest
> age <- mydata$Age
> PL1 <- mydata$PL.1
> height <- mydata$Ht
> HR2 <- mydata$HR.2
> FF <- mydata$FF.1
> cor(age, FF)
[1] -0.06848912
> cor(HRrest, FF)
[1] -0.1074611
> cor(HR2, FF)
[1] -0.4025726
> cor(PL1, FF)
[1] 0.1864209
> cor(height, FF)
[1] 0.1527773
> cor(mydata$RGM, FF)
[1] 0.2724712
> cor(mydata$LGM, FF)
[1] 0.2690495
> cor(mydata$VC, FF)
[1] 0.2628523
> cor(mydata$TA, FF)
[1] 0.003856256
> cor(mydata$PB, FF)
[1] -0.004331241
> cor(mydata$SBP, FF)
[1] -0.0759372
> cor(mydata$DBP, FF)
[1] -0.1243712
> cor(mydata$Stages, FF)
[1] 0.2849326
> cor(mydata$RF.2, FF)
[1] 0.1036071
> cor(mydata$RF.3, FF)
[1] 0.06887294
```

```
> cor(mydata$RF.4, FF)
[1] 0.002560065
> cor(mydata$RF.5, FF)
[1] 0.01949157
> cor(mydata$Wt, FF)
[1] -0.2428156
> cor(mydata$PL.2, FF)
[1] 0.2479871
> cor(mydata$HR.1, FF)
[1] -0.2801663
> cor(mydata$RPE.2, FF)
[1] -0.1810074
> cor(mydata$RPE.1, FF)
[1] -0.2342282
```

High absolute Correlation Variables:

Wt
HR2
HR.1
Stages
RPE.1
RGM
LGM
VC

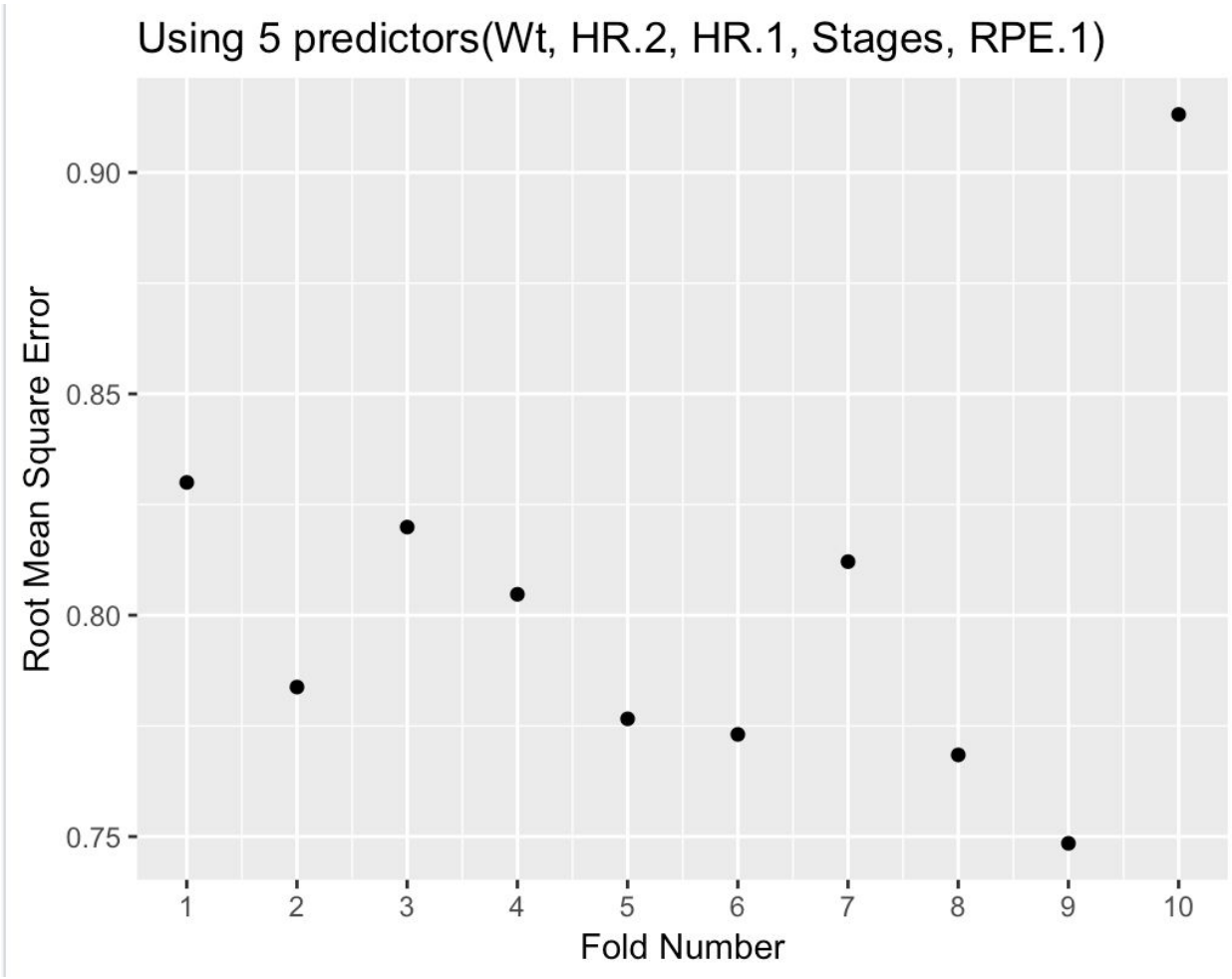
```
>library(ModelMetrics)
>library(modelr)
> mydata1 <- select(mydata, c(Wt, HR.2, HR.1,Stages,RPE.1,as.numeric(FF.1)))
> scaled.mydata1 <- scale(mydata1) #standardize the data to have the mean = 0 and sd = 1
> dfmydata1 <- as.data.frame(scaled.mydata1)
>rmse_i <- vector (length =10)
> for(i in 1:10){ testIndexes <- which(folds==i,arr.ind=TRUE)
+ testData <- dfmydata1[testIndexes, ]
+ trainData <- dfmydata1[-testIndexes, ]
+ mod <- lm(FF.1~Wt+HR.2+HR.1+Stages+RPE.1, data= trainData)
+ predicted <- predict(mod, testData)
+ rmse_i[i] <- rmse(testData$FF.1,predicted)
+ print(coefficients(mod))}

(Intercept)        Wt      HR.2      HR.1    Stages     RPE.1
-0.02442724 -0.46790999 -0.39711331 -0.07088067  0.12538318 -0.17231529
(Intercept)        Wt      HR.2      HR.1    Stages     RPE.1
-0.01413450 -0.46272302 -0.39929351 -0.06752584  0.12275469 -0.17387999
(Intercept)        Wt      HR.2      HR.1    Stages     RPE.1
-0.02233257 -0.47082952 -0.39605217 -0.07108214  0.11004439 -0.17235853
(Intercept)        Wt      HR.2      HR.1    Stages     RPE.1
-0.02350249 -0.46979351 -0.40488518 -0.07094011  0.10845226 -0.16287431
```

```
 (Intercept)          Wt        HR.2        HR.1      Stages       RPE.1
 0.002059357 -0.459430966 -0.400287440 -0.069438828  0.118802929 -0.177608927
(Intercept)          Wt        HR.2        HR.1      Stages       RPE.1
 0.01918013 -0.46535823 -0.40725866 -0.06773092  0.12465086 -0.16824954
(Intercept)          Wt        HR.2        HR.1      Stages       RPE.1
 0.02397654 -0.46804870 -0.39792718 -0.06815772  0.12807543 -0.17308433
(Intercept)          Wt        HR.2        HR.1      Stages       RPE.1
 0.01364030 -0.46687390 -0.40240411 -0.06446485  0.11940602 -0.17566873
 (Intercept)          Wt        HR.2        HR.1      Stages       RPE.1
 0.005317227 -0.457940575 -0.404584664 -0.071710963  0.110153910 -0.168777427
(Intercept)          Wt        HR.2        HR.1      Stages       RPE.1
 0.01488144 -0.47335379 -0.36848984 -0.16612060  0.11222440 -0.17009289
```

#Best standardized coefficients as of now:

```
(Intercept)          Wt        HR.2        HR.1      Stages       RPE.1
 0.005317227 -0.457940575 -0.404584664 -0.071710963  0.110153910 -0.168777427
```

```
dat <- data.frame(x = foldcount, y = rmse_i)
> ggplot(dat, aes(x,y)) +
+ geom_point() +
+ scale_x_continuous(breaks = round(seq(min(dat$x), max(dat$x), by = 1),1)) + xlab("Fold Number") + ylab("Root Mean Square Error") + ggtitle("Using 5 predictors(Wt, HR.2, HR.1, Stages, RPE.1)")
```



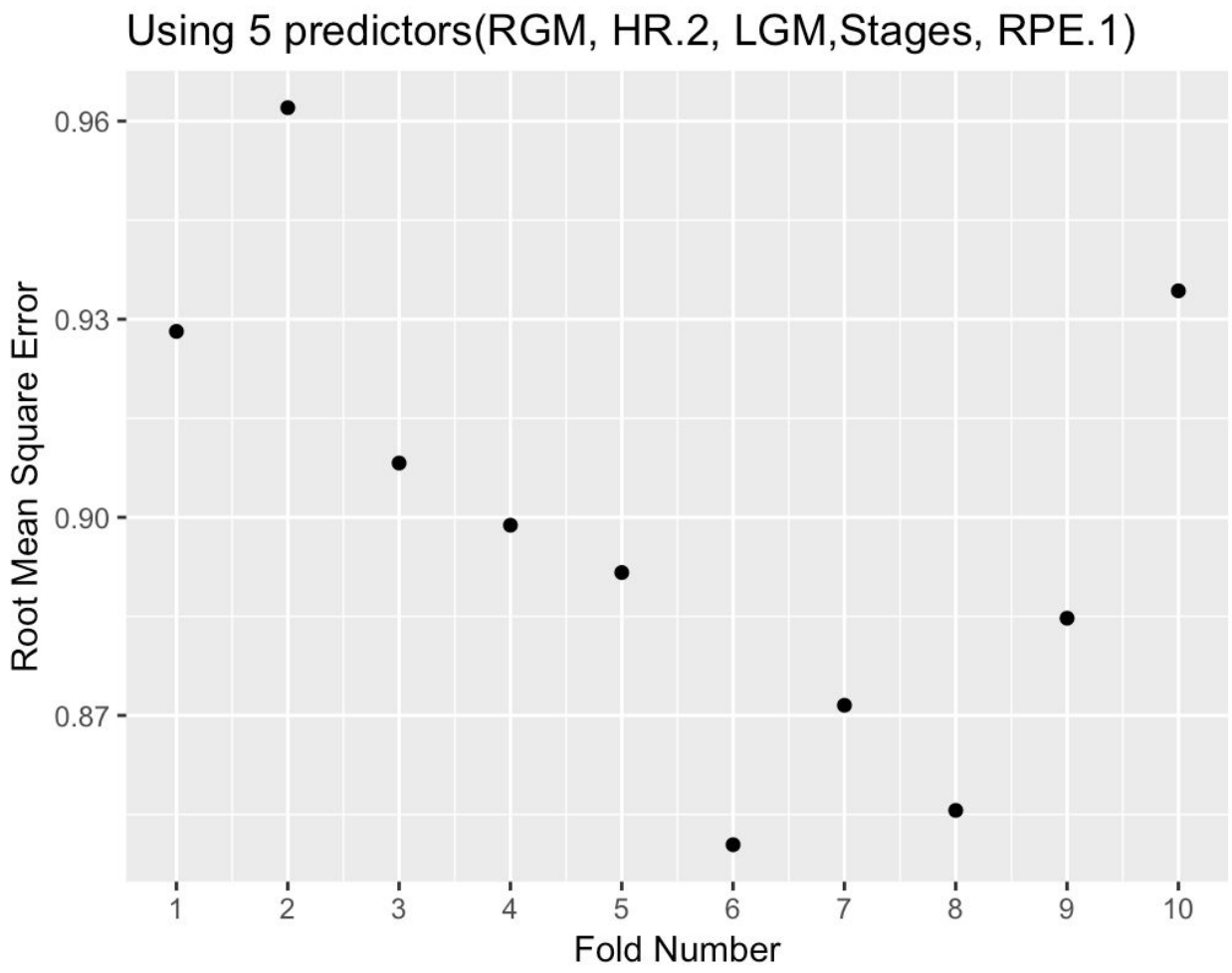

```
>mydata1 <- select(mydata, c(RGM, HR.2, LGM,Stages,RPE.1,as.numeric(FF.1)))
> scaled.mydata1 <- scale(mydata1)
> dfmydata1 <- as.data.frame(scaled.mydata1)
> for(i in 1:10){ testIndexes <- which(folds==i,arr.ind=TRUE)
+ testData <- dfmydata1[testIndexes, ]
+ trainData <- dfmydata1[-testIndexes, ]
+ mod <- lm(FF.1~RGM+HR.2+LGM+Stages+RPE.1, data= trainData)
+ predicted <- predict(mod, testData)
+ rmse_i[i] <- rmse(testData$FF.1,predicted)
+ print(coefficients(mod))}

 (Intercept)         RGM        HR.2
0.015117395  0.143998131 -0.301147506
         LGM      Stages       RPE.1
0.008553639  0.066456512 -0.082121894
 (Intercept)         RGM        HR.2
0.028300873  0.156149312 -0.303137264
         LGM      Stages       RPE.1
```

```
 0.005611394  0.064887910 -0.078562531
(Intercept)         RGM        HR.2
 0.01243600  0.11593927 -0.28924478
       LGM      Stages       RPE.1
 0.02736584  0.06966101 -0.08963321
 (Intercept)         RGM        HR.2
 0.009093478  0.123994923 -0.295152974
        LGM       Stages       RPE.1
 0.017364678  0.071218182 -0.082084999
(Intercept)         RGM        HR.2
-0.01429081  0.12312306 -0.29309947
       LGM      Stages       RPE.1
 0.01925110  0.07352065 -0.09358894
 (Intercept)         RGM        HR.2
-0.004444033  0.117589808 -0.288806148
        LGM       Stages       RPE.1
 0.024778190  0.083071529 -0.080938293
 (Intercept)         RGM        HR.2
-0.003079248  0.118003111 -0.284987691
        LGM       Stages       RPE.1
 0.019916192  0.082532468 -0.086189689
(Intercept)         RGM        HR.2
-0.00954194  0.12013867 -0.29321003
       LGM      Stages       RPE.1
 0.02189959  0.06849880 -0.08549575
(Intercept)         RGM        HR.2
-0.01741227  0.12131113 -0.30332600
       LGM      Stages       RPE.1
 0.03011691  0.05774075 -0.07228230
(Intercept)         RGM        HR.2
-0.01404469  0.11256786 -0.31113415
       LGM      Stages       RPE.1
 0.03242900  0.05766678 -0.07967391
```

#Best standardized coefficients as of now:

```
(Intercept)        Wt                HR.2              HR.1              Stages            RPE.1
 0.005317227 -0.457940575 -0.404584664 -0.071710963  0.110153910 -0.168777427
```

```
dat <- data.frame(x = foldcount, y = rmse_i)
> ggplot(dat, aes(x,y)) +
+ geom_point() +
+ scale_x_continuous(breaks = round(seq(min(dat$x), max(dat$x), by = 1),1)) + xlab("Fold Number") + ylab("Root Mean Square Error") + ggtitle("Using 5 predictors(RGM,HR.2, LGM,Stages, RPE.1)")
```

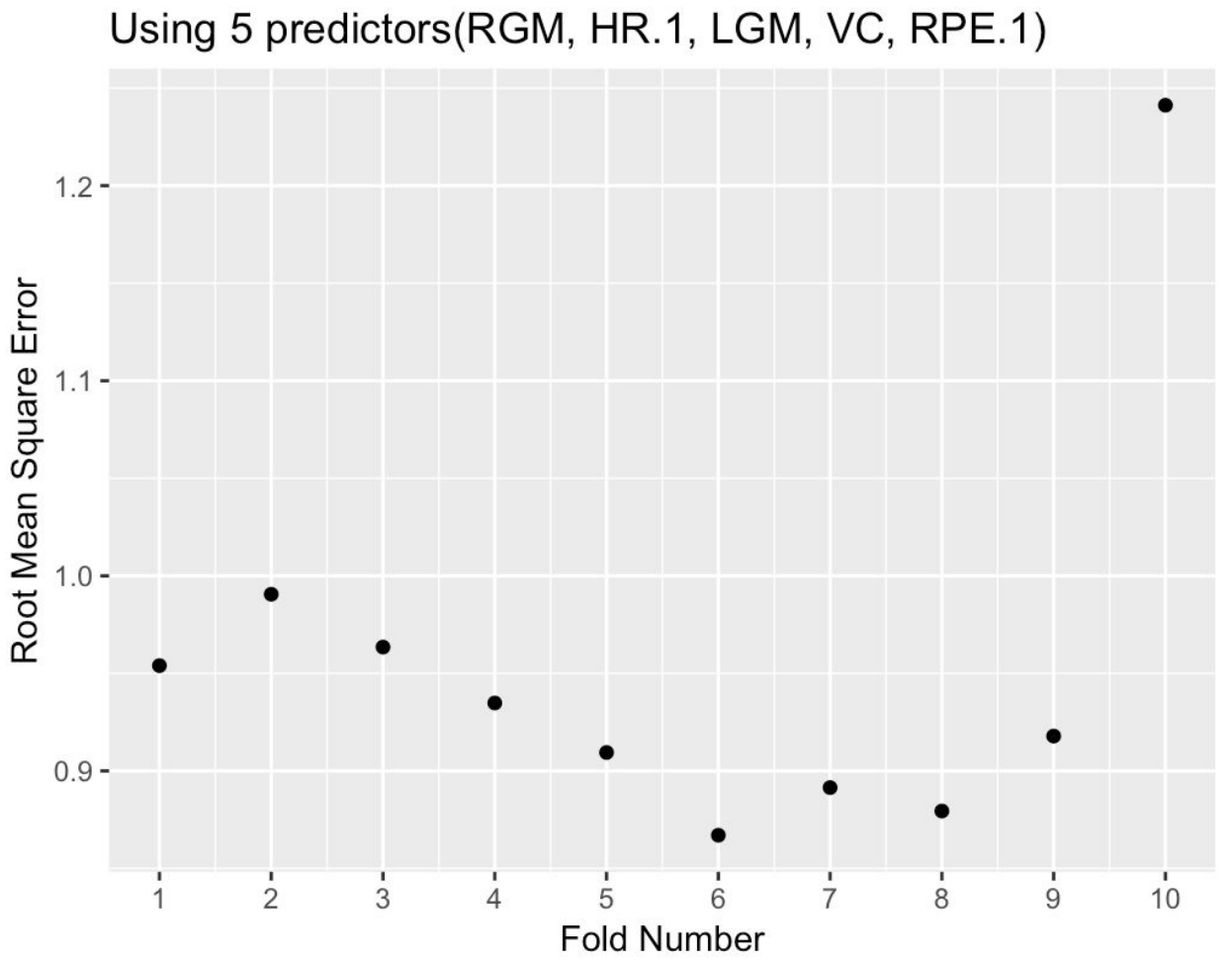



```
> mydata1 <- select(mydata, c(RGM, HR.1, LGM,VC,RPE.1,as.numeric(FF.1)))
> scaled.mydata1 <- scale(mydata1)
> dfmydata1 <- as.data.frame(scaled.mydata1)
> for(i in 1:10){ testIndexes <- which(folds==i,arr.ind=TRUE)
+ testData <- dfmydata1[testIndexes, ]
+ trainData <- dfmydata1[-testIndexes, ]
+ mod <- lm(FF.1~RGM+HR.1+LGM+VC+RPE.1, data= trainData)
+ predicted <- predict(mod, testData)
+ rmse_i[i] <- rmse(testData$FF.1,predicted)
+ print(coefficients(mod))}
(Intercept)        RGM       HR.1        LGM         VC       RPE.1
 0.02103346  0.12031529 -0.18794628  0.01767315  0.07352248 -0.12789604
 (Intercept)        RGM       HR.1        LGM         VC       RPE.1
 0.031706221  0.130781837 -0.181047321  0.002921656  0.088528694 -0.124343546
(Intercept)        RGM       HR.1        LGM         VC       RPE.1
 0.01571646  0.08436231 -0.18013859  0.03148784  0.08381213 -0.13412131
(Intercept)        RGM       HR.1        LGM         VC       RPE.1
 0.01031347  0.09831814 -0.17905006  0.01964108  0.07561392 -0.13035362
(Intercept)        RGM       HR.1        LGM         VC       RPE.1
```

-0.01389122 0.09274240 -0.17965545 0.03000112 0.07465401 -0.14123231
 (Intercept)   RGM   HR.1   LGM   VC   RPE.1
-0.004869123 0.096333214 -0.175649670 0.039365547 0.060885981 -0.128217248
 (Intercept)   RGM   HR.1   LGM   VC   RPE.1
-0.005090683 0.095176673 -0.176788769 0.037926313 0.057846721 -0.134191837
(Intercept)   RGM   HR.1   LGM   VC   RPE.1
-0.01119813 0.09433107 -0.17224947 0.02997383 0.07510485 -0.13378426
(Intercept)   RGM   HR.1   LGM   VC   RPE.1
-0.02090548 0.09270476 -0.18229918 0.03920666 0.07393068 -0.11794766
(Intercept)   RGM   HR.1   LGM   VC   RPE.1
-0.02255593 0.08956420 -0.38691080 0.02811109 0.02753540 -0.09058945

#Best standardized coefficients as of now:
(Intercept) Wt HR.2 HR.1 Stages RPE.1
 0.005317227 -0.457940575 -0.404584664 -0.071710963 0.110153910 -0.168777427

dat <- data.frame(x = foldcount, y = rmse_i)
> ggplot(dat, aes(x,y)) +
+ geom_point() +
+ scale_x_continuous(breaks = round(seq(min(dat$x), max(dat$x), by = 1),1)) + xlab("Fold Number") + ylab("Root Mean Square Error") + ggtitle("Using 5 predictors(RGM, HR.1, LGM,VC,RPE.1)")

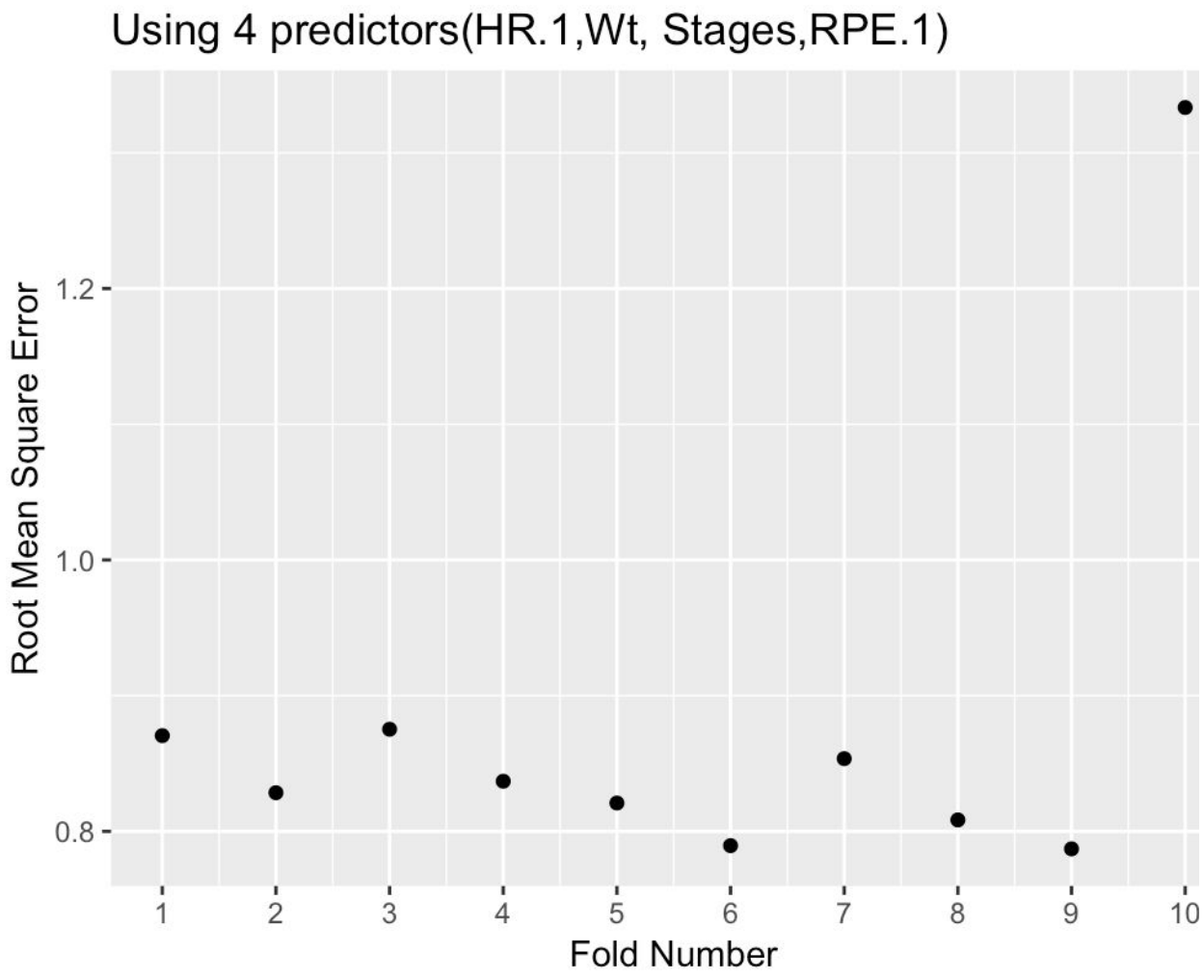



```
> mydata1 <- select(mydata, c(HR.1,Wt, Stages,RPE.1,as.numeric(FF.1)))
> scaled.mydata1 <- scale(mydata1)
> dfmydata1 <- as.data.frame(scaled.mydata1)

> for(i in 1:10){ testIndexes <- which(folds==i,arr.ind=TRUE)
+ testData <- dfmydata1[testIndexes, ]
+ trainData <- dfmydata1[-testIndexes, ]
+ mod <- lm(FF.1~HR.1+Wt+Stages+RPE.1, data= trainData)
+ predicted <- predict(mod, testData)
+ rmse_i[i] <- rmse(testData$FF.1,predicted)
+ print(coefficients(mod))}
(Intercept)       HR.1        Wt     Stages      RPE.1
-0.02403253 -0.23452294 -0.41603557  0.26976410 -0.20744824
(Intercept)       HR.1        Wt     Stages      RPE.1
-0.01406206 -0.23005429 -0.41032743  0.26679038 -0.21019233
(Intercept)       HR.1        Wt     Stages      RPE.1
```

-0.01912096 -0.23252317 -0.41580406 0.25198615 -0.20787489
(Intercept) HR.1 Wt Stages RPE.1
-0.01952707 -0.23550625 -0.41307819 0.25241279 -0.20058096
(Intercept) HR.1 Wt Stages RPE.1
0.002145374 -0.231045737 -0.405704786 0.265229021 -0.212506819
(Intercept) HR.1 Wt Stages RPE.1
0.01063664 -0.23476427 -0.40657470 0.27425502 -0.20639704
(Intercept) HR.1 Wt Stages RPE.1
0.02111296 -0.22718542 -0.41202370 0.27338475 -0.20705479
(Intercept) HR.1 Wt Stages RPE.1
0.01585976 -0.22302179 -0.41322921 0.26565610 -0.21239364
(Intercept) HR.1 Wt Stages RPE.1
0.008604526 -0.230616707 -0.404790805 0.259545853 -0.204355641
(Intercept) HR.1 Wt Stages RPE.1
0.007153235 -0.502814846 -0.444727374 0.197821165 -0.169476400

#Best standardized coefficients as of now:
(Intercept) Wt HR.2 HR.1 Stages RPE.1
0.005317227 -0.457940575 -0.404584664 -0.071710963 0.110153910 -0.168777427

```
> dat <- data.frame(x = foldcount, y = rmse_i)
> ggplot(dat, aes(x,y)) +
+ geom_point() +
+ scale_x_continuous(breaks = round(seq(min(dat$x), max(dat$x), by = 1),1)) + xlab("Fold Number") + ylab("Root Mean Square Error") + ggtitle("Using 4 predictors(HR.1,Wt, Stages,RPE.1)")
```



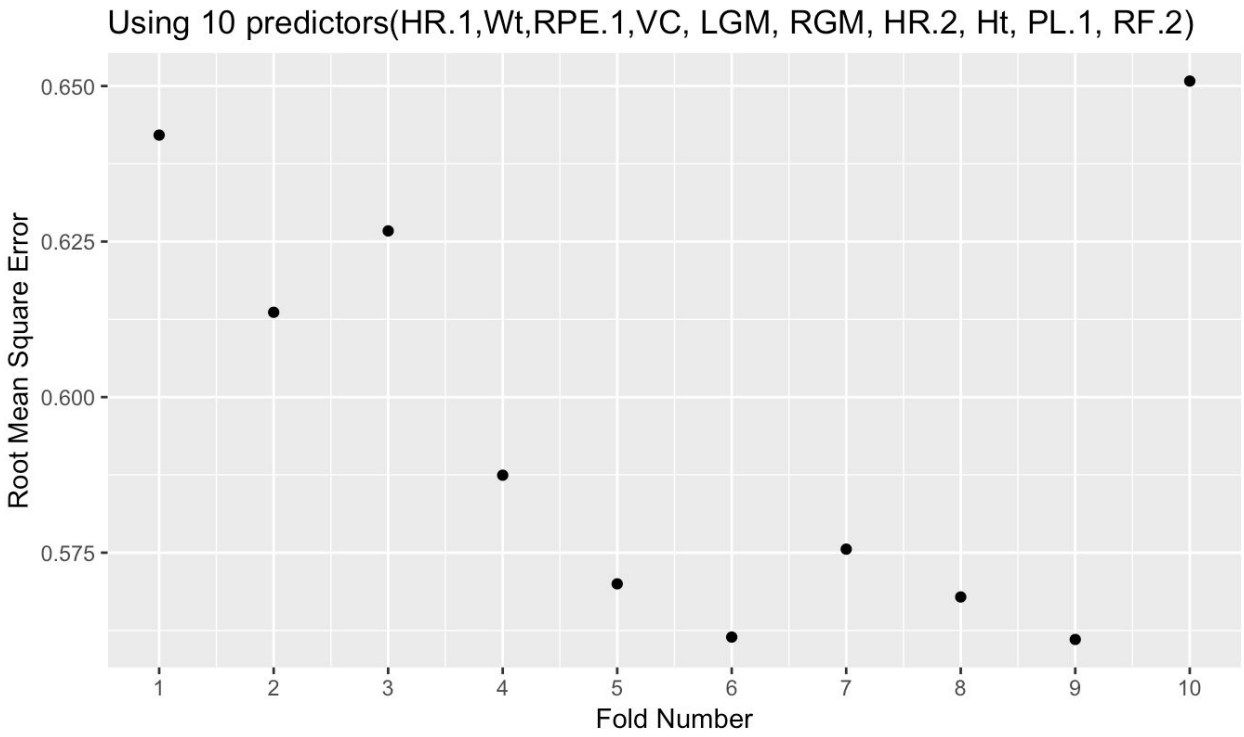

```
> mydata1 <- select(mydata, c(HR.1,Wt,RPE.1,VC, LGM, RGM, HR.2, Ht, PL.1,
RF.2,as.numeric(FF.1)))
> scaled.mydata1 <- scale(mydata1)
> dfmydata1 <- as.data.frame(scaled.mydata1)

> for(i in 1:10){ testIndexes <- which(folds==i,arr.ind=TRUE)
+ testData <- dfmydata1[testIndexes, ]
+ trainData <- dfmydata1[-testIndexes, ]
+ mod <- lm(FF.1~HR.1+Wt+RPE.1+VC+LGM+RGM+HR.2+Ht+PL.1+RF.2, data= trainData)
+ predicted <- predict(mod, testData)
+ rmse_i[i] <- rmse(testData$FF.1,predicted)
+ print(coefficients(mod))}
(Intercept)       HR.1         Wt      RPE.1         VC        LGM        RGM
 0.01270484 -0.01951793 -0.80819136 -0.07481508  0.44228514  0.29395025  0.39671051
      HR.2         Ht       PL.1       RF.2
-0.39252124 -0.01985352  0.26527669 -0.67512173
 (Intercept)       HR.1         Wt      RPE.1         VC        LGM
```

```
 0.002197257 -0.017754403 -0.817841495 -0.076223764  0.445040303  0.261155269
         RGM         HR.2           Ht         PL.1         RF.2
 0.409072721 -0.397145848 -0.016477792  0.265675463 -0.671860026
 (Intercept)         HR.1           Wt        RPE.1           VC          LGM
 0.003930901 -0.019585994 -0.820276595 -0.073005612  0.449197142  0.262532682
         RGM         HR.2           Ht         PL.1         RF.2
 0.410372665 -0.392336298 -0.020772743  0.266398991 -0.669970877
(Intercept)         HR.1           Wt        RPE.1           VC          LGM          RGM
-0.01073760 -0.01526432 -0.80961940 -0.07483331  0.44498872  0.27285000  0.40501378
       HR.2          Ht         PL.1         RF.2
-0.40073656 -0.03667419  0.26661695 -0.68341406
 (Intercept)         HR.1           Wt        RPE.1           VC          LGM
-0.004733053 -0.012036864 -0.802808478 -0.078801917  0.427459566  0.263152611
         RGM         HR.2           Ht         PL.1         RF.2
 0.400609646 -0.402581830 -0.026814417  0.261108272 -0.661958616
 (Intercept)         HR.1           Wt        RPE.1           VC          LGM
 0.004448815 -0.011196061 -0.806002321 -0.076243422  0.430438536  0.261925570
         RGM         HR.2           Ht         PL.1         RF.2
 0.386138226 -0.398829674 -0.026158083  0.274769485 -0.654491078
 (Intercept)         HR.1           Wt        RPE.1           VC          LGM
 0.008494921 -0.013252507 -0.805144417 -0.078583842  0.417675697  0.269712584
         RGM         HR.2           Ht         PL.1         RF.2
 0.388507664 -0.399028106 -0.024077804  0.278532990 -0.662821871
  (Intercept)         HR.1           Wt        RPE.1           VC          LGM
 0.0009990666 -0.0106339558 -0.8102201487 -0.0761413055  0.4281889043  0.2696477319
          RGM         HR.2           Ht         PL.1         RF.2
 0.3859962569 -0.4027650456 -0.0353285982  0.2696260282 -0.6439193309
 (Intercept)         HR.1           Wt        RPE.1           VC          LGM
-0.001386602 -0.014550106 -0.808534067 -0.069250520  0.424153562  0.261874243
         RGM         HR.2           Ht         PL.1         RF.2
 0.392091823 -0.406643686 -0.028298968  0.265675958 -0.640366565
(Intercept)         HR.1           Wt        RPE.1           VC          LGM          RGM
-0.01398105  0.02708797 -0.82085909 -0.07356822  0.42730364  0.26877383  0.39371727
       HR.2          Ht         PL.1         RF.2
-0.44545422 -0.03379407  0.25892456 -0.62110449
```

#Best standardized coefficients as of now:

```
 (Intercept)         HR.1           Wt        RPE.1           VC          LGM
-0.001386602 -0.014550106 -0.808534067 -0.069250520  0.424153562  0.261874243
         RGM         HR.2           Ht         PL.1         RF.2
 0.392091823 -0.406643686 -0.028298968  0.265675958 -0.640366565
```

```
dat <- data.frame(x = foldcount, y = rmse_i)
ggplot(dat, aes(x,y)) +
geom_point() +
scale_x_continuous(breaks = round(seq(min(dat$x), max(dat$x), by = 1),1)) + xlab("Fold Number") + ylab("Root Mean Square Error") + ggtitle("Using 10 predictors(HR.1,Wt,RPE.1,VC, LGM, RGM, HR.2, Ht, PL.1, RF.2)")
```



3.

Based on these results, the HR.1,Wt, RPE.1, VC, LGM, RGM, HR.2, Ht, PL.1, RF.2 features resulted in the lowest RMSE and the best model based on cross validation is chosen.

We unstandardized the coefficients to return a value based on unstandardized new data since we assumed that's what it would be tested upon for grading. The unstandardized version of the best model is

(Intercept)	HR.1	Wt	RPE.1	VC	LGM
73.914252940	-0.005822095	-0.225436985	-0.317382187	4.261133574	0.209097632
RGM	HR.2	Ht	PL.1	RF.2	
0.309262518	-0.237680813	-0.068163275	0.241961354	-2.980731016	

With a RMSE of 5.16257 as shown by the code below.

```
> for(i in 1:10){ testIndexes <- which(folds==i,arr.ind=TRUE)
+ testData <- mydata1[testIndexes, ]
+ trainData <- mydata1[-testIndexes, ]
+ mod <- lm(FF.1~HR.1+Wt+RPE.1+VC+LGM+RGM+HR.2+Ht+PL.1+RF.2, data= trainData)
+ predicted <- predict(mod, testData)
+ rmse_i[i] <- rmse(testData$FF.1,predicted)
+ print(coefficients(mod))}
```

```
> for(i in 1:10){print(rmse_i[i])}
[1] 5.908424
[1] 5.646373
[1] 5.766743
[1] 5.405467
[1] 5.244779
[1] 5.166067
[1] 5.296099
[1] 5.225397
[1] 5.16257
[1] 5.988335
```

The resulting code is:

```
> totalfitnessfactorscore <- function(HR.1, Wt, RPE.1, VC, LGM, RGM, HR.2, Ht, PL.1, RF.2) {
+         y <- 73.914252940 + -0.005822095*HR.1 + -0.225436985*Wt + -0.317382187*RPE.1 +
4.261133574*VC + 0.209097632*LGM + 0.309262518*RGM + -0.237680813*HR.2 +
-0.068163275*Ht + 0.241961354*PL.1 + -2.980731016*RF.2 + return(y) }
```