Fitness Data Classification

MATH 398 Machine Learning Final Project

By: Ethan Perry and Jeffrey Stewart

Table of Contents

Introduction

This data set comes from the Korea Sports Promotion Formation in South Korea. The data collection aimed to classify the 13393 participants into 1 of 4 graded categories ranging from A (the most fit) to D (the least fit or participation). Participants ranged between the ages of 21 and 64 with a variety of physical characteristics with tests being run to classify their fitness level. The publisher of this data set onto the data forum Kaggle states that "[t]he criteria for classifying grades are not known precisely because they are set by the institution" (kukuroo3). We hope to uncover if there was any logical or algorithmic classifying process or if these data were subjectively placed into groups by the test varying administrators. Our goal is to approximate the function the institution might've used or be able to approximate characteristics the subjective graders used as a classification boundary.

Below are the available features from the post-processed data set. From further discovery it was uncovered that there were some omitted features that might have proven more elucidating in a predictive models creation however, we will proceed with the following features given in the processed data (Korea Sports Promotion Foundation 2022):

Description of Features

Age: age of the participant
Gender: binary feature, male or female
Height_cm: height of the participant in centimeters
Weight_kg: weight of the participant in kilograms
body.fat_%: The percentage of the participant's weight that is fat.
Diastolic: The diastolic blood pressure of the participant mmHg
Systolic: The systolic blood pressure of the participant in mmHg
gripForce: The grip force of the participant in kilograms
Sit.and.bend.forward_cm: Distance achieved by the participant on the sit and bend forward test
Sit.ups.counts: the amount of sit ups completed by the participant in two minutes
Broad.jump_cm: Broad jump length of the participant in centimeters. Broad jump is also referred to as standing long jump

Outliers

There were a variety of outliers in this data set that needed to be removed because they were clearly miss entries in some form or another. The following numbers for each participant coordinate to the row number they happened to be in when cleaning the data. Participants 2658 and 3356 had physically impossible values for sit-and-reach with 213cm and 185cm respectively. That equates to several feet and a distance taller than themselves. Participant 736 had a body fat of 78.4% which is not entirely impossible but given they were placed in class A and the spread of the rest of the participants body fat, it was a clear outlier. Participant 10625 had a systolic and diastolic blood pressure of 0 which was treated as an NA and removed. Participants 337 and 3525 had diastolic blood pressures of 8 and 6 respectively so they were removed as well. Participant 8252 was removed as an outlier for height being around 125cm (4 feet). This is not absolutely an outlier but given the assumption of normality needed for LDA and the large data set, we decided to remove their entry. Finally, participants 8563, 9487, and 9769 were removed because they had 0 recorded for grip force, so it was treated as an NA entry and removed. Those are all the clear outliers that aided in less variant data and more realistic predictions.

Initial Analysis

Ultimately we wish to find a formula or coefficients for an algorithm that best approximate the potentially subjective classification from the data collection. Being able to generalize a model to classify entirely new data would allow for more standard and repeatable classifying results. Since these data were collected in a subjective manner and we lack access to the unprocessed data set, working with what is given caused some difficulties in the initial analysis. Exploring the data set revealed some decently variant data and attempting to uncover the features responsible for this interference in making class predictions turned out to be non-trivial. Exploring the feature set and understanding the shortcomings of our data allowed for us to better predict class for an individual given strategic modeling selections.

We began our analysis by fitting histograms of the various features to get a better grasp on the data. Immediately there were some odd results, as a few of the histograms seemed like they had two peaks. We thought that each peak might correspond with male and female individuals, and after splitting the histograms by gender, that seemed to be the case. Below are histograms of body fat percentage and grip force split by gender.

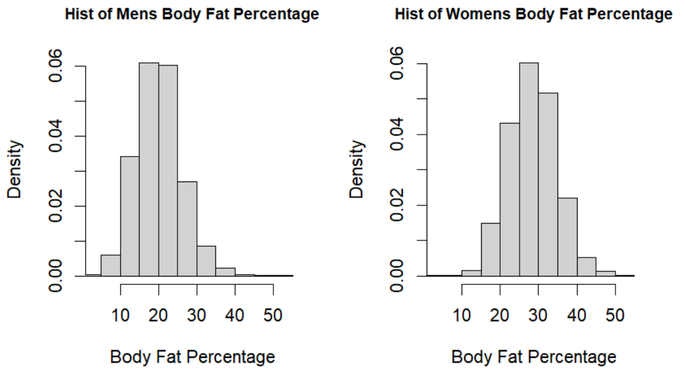


Fig.1: Histogram of Grip Force split by gender.

Fig. 2: Histogram of Body Fat % split by gender.

It is clear that the mean of these data differ for the gender groups which will likely need to be accounted for in the models. Logically, many of these physical attributes physiologically differ between genders, which brought up the question of whether the requirements to achieve a specific fitness class are the same between genders. Looking at men and women specifically classified as A level fitness, the median grip force among A class men is 46.8 kg and the median grip force among A class women is 27.9 kg. The median body fat percentage among A class men is 16.6% and the median body fat percentage among A class women is 25.1%. This suggests that the characteristics for an individual to receive a specific class is indeed different depending on gender.

Continuing Analysis

With the intention of confirming our idea that gender would be an important feature for predicting class, we decided to fit a Random Forest model. Throughout our experience of the course, Random Forest models seem to have consistently done the best in a classification setting similar to ours. Fitting one allows us to view the variable importance plot which measures the most useful features for predicting class.

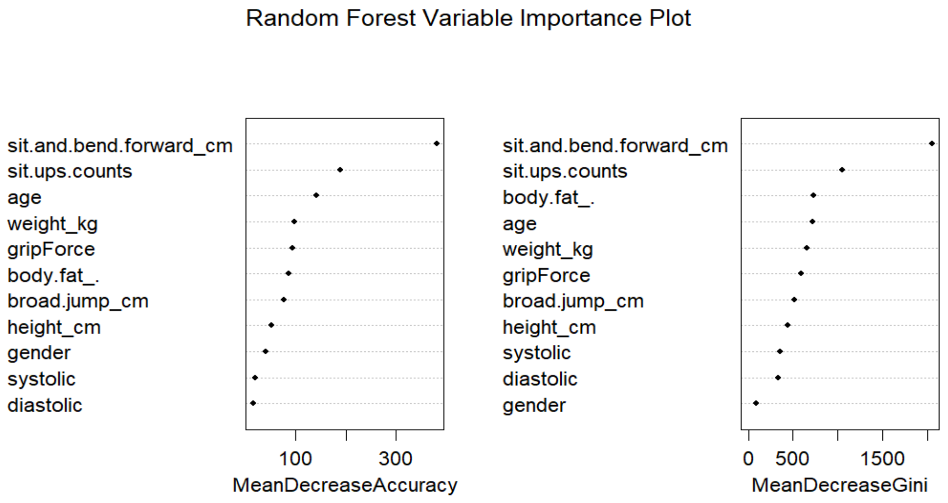


Fig. 3: Variable Importance Plot from Random Forest predicting class.

The plot on the left measures “how much accuracy the model loses by excluding each variable”(Martinez-Taboada). The plot on the right measures the mean decrease in Gini coefficient associated with each variable. For both plots, variables with larger values have higher importance (Martinez-Taboada). In both plots, gender is one of the least important variables for predicting class. This was puzzling given the histogram results above clearly showing different spread of feature data between the genders. We then proceeded to split up class and gender among each of the continuous features to understand how this difference may be interacting with class prediction.

Visualizing Gender Difference

To be able to visualize the difference between gender for a given feature, it was important to also look at the variation between classes. Below are the three dimensional plots that allowed further clarification on what was occuring. Below, the female group is identified by orange points and the red lowess line where the male group is the blue point with the corresponding blue lowess line. The points for men are shifted up slightly to avoid overlap, but the lowess curve is not shifted up for the sake of comparison. Classes A through D are represented as 4 through 1 on the graph.

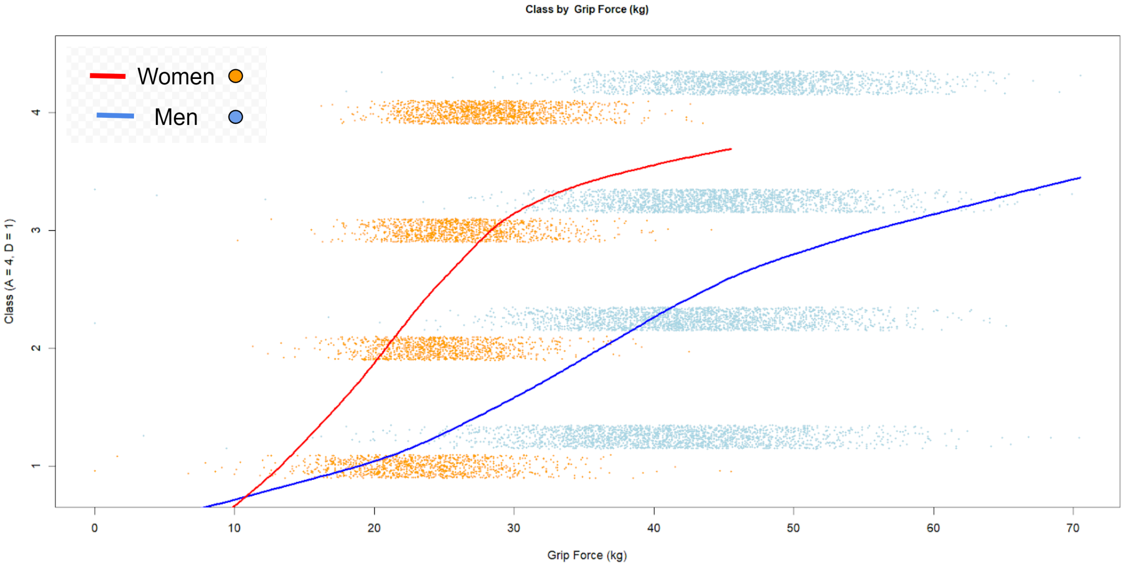


Fig.4: Class by Grip Force showing gender difference.

Looking at this graph, it is clear that there is a significant difference between the grip force among women and men in each class. The range of grip force among men seems to be wider than for women, and the range for women in each class is generally lower than the range for men in the same class. There is a slight trend of fitness level increasing as grip force increases, but based on the lowess curves, the rate of increase differs between genders. Thus, we would expect grip force to have a different effect on class predictions depending on gender.

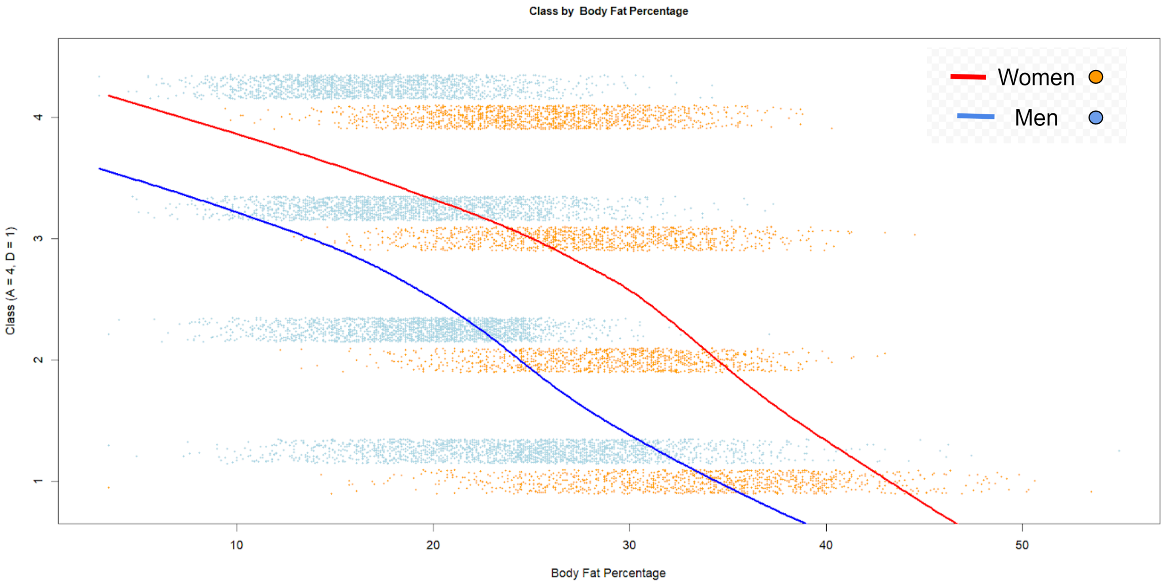


Fig. 4: Class by Body Fat % showing gender difference.

The plot of body fat percentage also shows a clear difference between men and women within each class. Generally, as body fat percentage increases, fitness level decreases. Looking at the lowess curves, the slopes are fairly similar, but the range of body fat percentage is higher for women compared to men of the same class. For example, the majority of A class men have a body fat percentage lower than 25%, but 25.1% is the median among A class women.

In both graphs, despite the lowess curves indicating that there is a relationship between the feature and class, there is significant overlap between the classes, which puts into question how useful these variables on their own would be for predicting class. We decided to also look at the most significant feature according to the variable importance plot to see if this pattern was consistent.

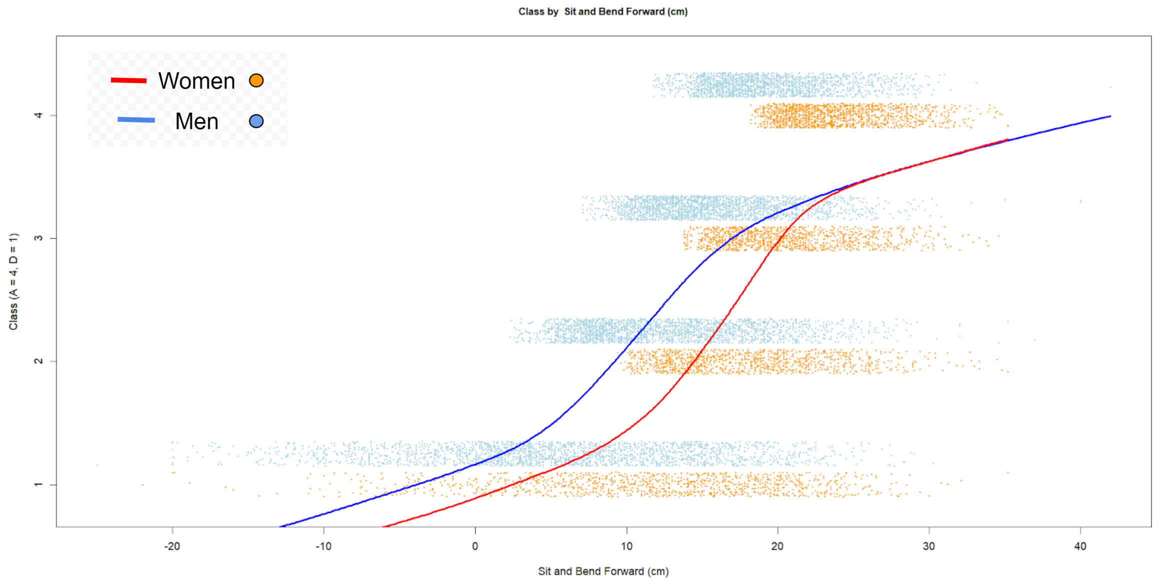
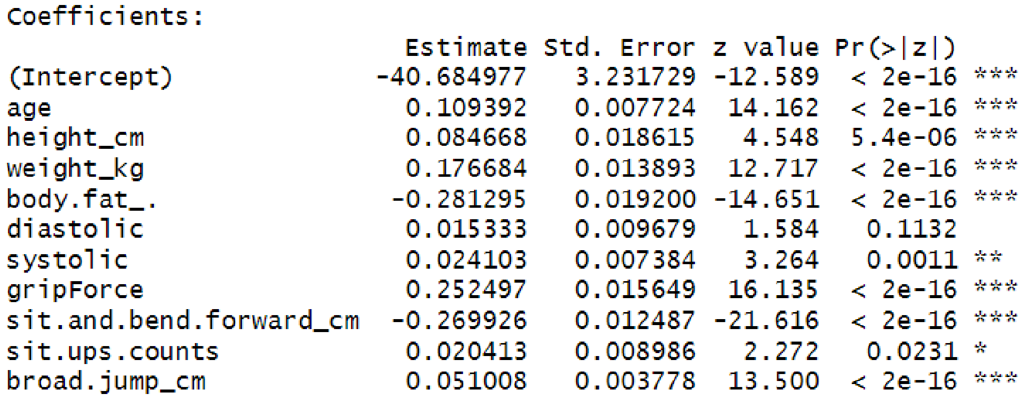


Fig. 5: Class by Sit-and-bend-forward showing gender difference.

While there is still overlap between classes in this graph, the points of distinction between classes are much more pronounced. This is most significant for the distinction between D and C class. There are no men with a sit and bend forward distance of less than about 2 cm in C class, and it looks like nearly half of D class men have a distance less than than 2 cm. Even in the most important feature, there is a noticeable difference between genders, with the range of distance achieved by women being higher than that of men in the same class.

Gender’s Significance

The results of the analysis so far seem contradictory. All of our graphical evidence suggests that gender should be significant, as most of the features have significantly different values depending on gender, but the variable importance plot labels gender as insignificant. In an attempt to explain this, we wanted to test how well gender could be modeled by the other features, and by doing so see how much of the gender difference is explained by the other features. This was done by fitting a logistic model that used all the features and not class, to predict gender. Below is the output:



```
Coefficients:
                         Estimate Std. Error z value Pr(>|z|)    
(Intercept)            -40.684977   3.231729 -12.589  < 2e-16 ***
age                      0.109392   0.007724  14.162  < 2e-16 ***
height_cm                0.084668   0.018615   4.548 5.4e-06 ***
weight_kg                0.176684   0.013893  12.717  < 2e-16 ***
body.fat_.              -0.281295   0.019200 -14.651  < 2e-16 ***
diastolic                0.015333   0.009679   1.584   0.1132    
systolic                 0.024103   0.007384   3.264   0.0011 ** 
gripForce                0.252497   0.015649  16.135  < 2e-16 ***
sit.and.bend.forward_cm -0.269926   0.012487 -21.616  < 2e-16 ***
sit.ups.counts           0.020413   0.008986   2.272   0.0231 *  
broad.jump_cm            0.051008   0.003778  13.500  < 2e-16 ***
```

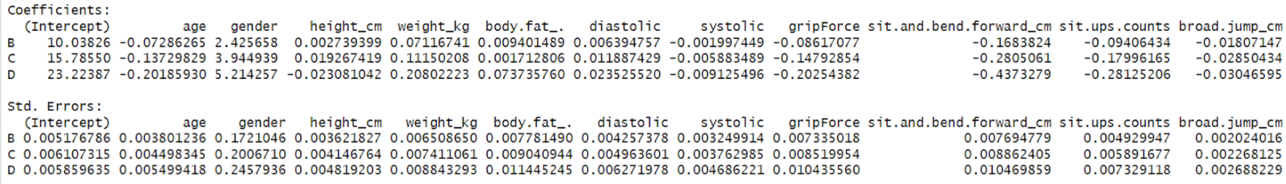
Fig. 6: Logistic regression output predicting gender from all the features.

It is clear that most of these features are significant in predicting a person's gender with most of the variables being highly significant. This is evidence to show that gender is highly correlated with each of these features respectively. The output indicates that there is multicollinearity and is giving similar information as a variance inflation factor analysis. Since so much of the variation in gender is explained by the remaining biological markers, this explains why gender has not been appearing as significant in our initial analyses. Continuing on this vein, attempting to predict gender on a test set when giving a trained model for predicting gender has a miss classification rate of roughly 1.4%.

```
            test
prediction  female male
    female    1204   21
    male        27 2093
```

Fig. 7: Confusion matrix for fitting test set on above model.

These findings further support that the features abilities to explain gender and therefore gender, as a variable for modeling, could be potentially insignificant. To gauge the significance of these variables in relation to their importance for predicting class, a multinomial logistic regression was done and the standard errors for the coefficients were analyzed. Since only the error terms were given, they were treated as a quasi p-value with a larger error suggesting less significance in predictive ability for class.



```
Coefficients:
  (Intercept)         age   gender    height_cm  weight_kg  body.fat_.   diastolic     systolic   gripForce sit.and.bend.forward_cm sit.ups.counts broad.jump_cm
B    10.03826 -0.07286265 2.425658  0.002739399 0.07116741 0.009401489 0.006394757 -0.001997449 -0.08617077              -0.1683824    -0.09406434   -0.01807147
C    15.78550 -0.13729829 3.944939  0.019267419 0.11150208 0.001712806 0.011887429 -0.005883489 -0.14792854              -0.2805061    -0.17996165   -0.02850434
D    23.22387 -0.20185930 5.214257 -0.023081042 0.20802223 0.073735760 0.023525520 -0.009125496 -0.20254382              -0.4373279    -0.28125206   -0.03046595

Std. Errors:
  (Intercept)         age    gender   height_cm   weight_kg  body.fat_.   diastolic    systolic   gripForce sit.and.bend.forward_cm sit.ups.counts broad.jump_cm
B 0.005176786 0.003801236 0.1721046 0.003621827 0.006508650 0.007781490 0.004257378 0.003249914 0.007335018             0.007694779    0.004929947   0.002024016
C 0.006107315 0.004498345 0.2006710 0.004146764 0.007411061 0.009040944 0.004963601 0.003762985 0.008519954             0.008862405    0.005891677   0.002268125
D 0.005859635 0.005499418 0.2457936 0.004819203 0.008843293 0.011445245 0.006271978 0.004686221 0.010435560             0.010469859    0.007329118   0.002688229
```

Fig 8: Multinomial Regression output predicting class and showing feature coefficient significance.

It is noticeable that every coefficient for each class has a lower standard error of 0.01 or less whereas gender is much more variable with values of 0.17, 0.20, and 0.25. It is now clear that gender is not a significant predictor for class and that is caused by the other variables' ability to account for it inherently.

We now wish to see how fitting models with and without splitting by gender do in a classification setting. Given the findings above, we expect that, when accounting for gender by separating them into two groups, the models will have similar misclassification rates as models without this split. This split by gender can also aid in the final models interpretability because the coefficients will be gender specific.

Linear Discriminant Analysis Theory

Linear Discriminant Analysis (LDA) is an approximation to the Bayes Classifier which aims to classify a discrete label by dividing the feature space with discriminant functions. “The Bayes classifier produces the lowest possible test error rate, called the Bayes error rate” (James et al. 38). Since we are approximating an algorithm with the lowest error rate, the hope is to classify with a similarly low error rate.

There are some theoretical assumptions that are required about the data before moving forward with the computation. Each feature is assumed to be normally distributed and therefore will be scaled down with a normal distribution of mean zero and variance of one. This is important because there is a shared covariance matrix among all features so scaling these features allows for this matrix to be applied in the formula. LDA takes a similar form to the Bayes Classifier as the formula below indicates.

$$\delta_k(x) = x^T \boldsymbol{\Sigma}^{-1} \mu_k - \frac{1}{2}\mu_k^T \boldsymbol{\Sigma}^{-1} \mu_k + \log \pi_k$$

Fig. 9: Linear DIscriminant Analysis classification function.

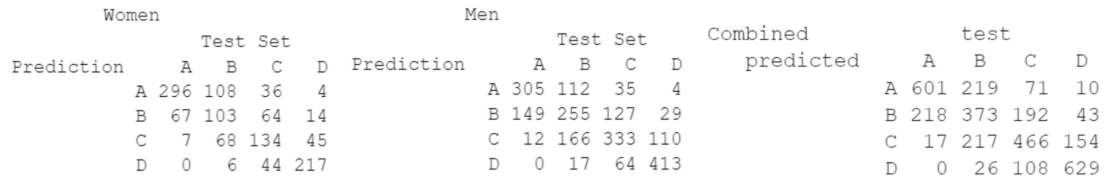
x^T: vector of feature values, transpose is taken for multiplication
Σ^{-1}: shared covariance matrix for the given data, (inverted)
μ_k: vector of means for each feature for class k
π_k: defined as n_k / n, the proportion of total subjects in class k
Note that in practice, these values are estimations of the true population values and are approximating the discriminant function for the population.

This formula returns a value for each of the k classes and the largest one is the class to which the entry is assigned. It can be thought of as a type of softmax function. In a higher dimensional space, a hyperplane is the discriminant space and best separates the k-classes by maximizing between group variance. The boundary lines are defined where there is a tie for the max value between classes and that is where the term decision boundary is derived.

Additionally, there are discriminant coefficients returned for each corresponding feature allowing for a form of dimensionality reduction. Each LD aims to describe the variation between classes which can be found by the trace values associated with each. Being able to work with a dimensionally reduced variable allows for plotting of the discriminant functions separated by class.

Fitting LDA Models

Using the lda() R command in the MASS package, we were able to fit a LDA with reported discriminant functions. We hope to further explain how our knowledge of the theory aided our analysis of these models. Firstly, below are the confusion matrices for our models. The misclassification rate for women is 38.17%, for men it is 38.71%, and the rate for combined is 38.13%.



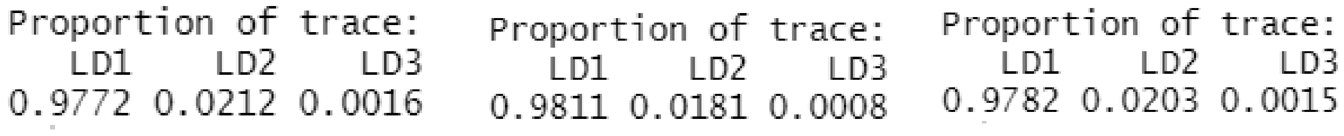
```
      Women
                Test Set
Prediction     A   B   C   D
         A 296 108  36   4
         B  67 103  64  14
         C   7  68 134  45
         D   0   6  44 217
```

```
      Men
                Test Set
Prediction     A   B   C   D
         A 305 112  35   4
         B 149 255 127  29
         C  12 166 333 110
         D   0  17  64 413
```

```
Combined             test
   predicted     A   B   C   D
             A 601 219  71  10
             B 218 373 192  43
             C  17 217 466 154
             D   0  26 108 629
```

Fig. 10: Confusion matrices of the three LDA models

The fact that the misclassification rates are so similar indicates that in the LDA setting, gender does not aid in separating the groups as expected. Since so much of this variance is already described in the features and the features are weighted in the dimensionality reduction or discriminant functions, accounting for gender by splitting models is not aiding the predictive capabilities of these LDA models. Given that our data is rather noisy and some of the assumptions were not extremely accurate, the misclassification rate was higher than we would've liked. Below are the trace values for each LDA training model.Trace is the "proportion of between-class variance that is explained by successive discriminant functions" (*LDA and Model assessment in R*).



Men

```
Proportion of trace:
   LD1    LD2    LD3 
0.9772 0.0212 0.0016 
```

Women

```
Proportion of trace:
   LD1    LD2    LD3 
0.9811 0.0181 0.0008 
```

Combined

```
Proportion of trace:
   LD1    LD2    LD3 
0.9782 0.0203 0.0015 
```

Fig 11: Trace values for each LD function respective to each model.
See full output from LDA model including coefficients on page 23-24

Noticeably, the coefficients for the discriminant functions are different for the male, female, and combined functions. This makes sense granted the conclusions above about the physiological differences between the male and female groups.

The main discriminant function, LD1, is the discriminating hyperplane that is able to separate the between group variance the best and accounts for roughly 98% of the variance explained by the model. This means that the dimensional reduction done and the weights reported from this function can be treated as its own form of classification and is what would be used to predict on new data. Granted, some of the variation is not explained without the inclusion of the other discriminant functions but since 98% is accounted for, LD1 on its own can be used to classify.

Another piece to note is the inverse covariance matrices for the three models (see page 29 for inverse of those matrices). These are used in the LDA decision boundary algorithm and it's helpful to get a sense of the values. Being able to report the inverse of the covariance matrix allows for some values to be calculated by hand and see the process which this algorithm follows.

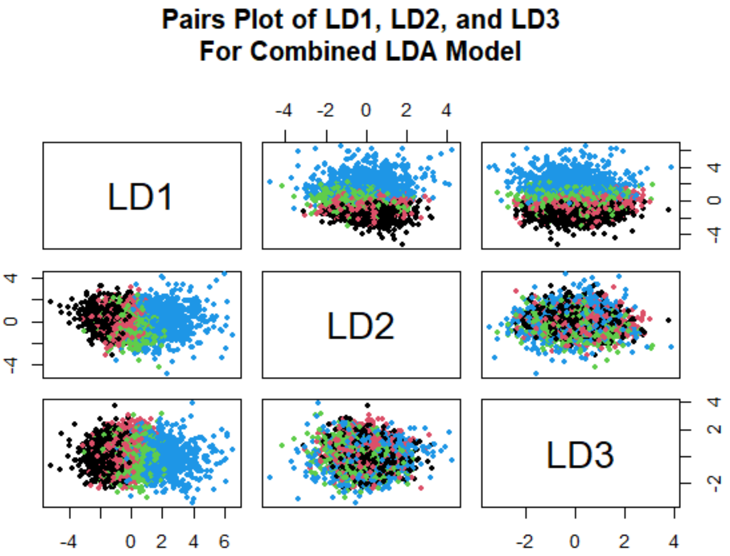


Fig. 12: Pairs plot of LD1, LD2, and LD3 for the training LDA model combined group.
Class A = black, red = B, green = C, and blue = D

Here are the plots of the discriminant functions treated as variables and colored by class where. It is rather clear that LD1 can distinguish between these classes the best given the plots of LD1 vs. LD2 as well as LD1 vs. LD3. The LD2 vs. LD3 fails to effectively separate classes. This shows that neither LD2 nor LD3 are particularly helpful for predicting class, which is consistent with the percentage trace explained from the model summary. Having been able to explain 98% of the variance between classes with LD1, this new single variable, which is a fitted discriminant function value for each individual, allows for much more visualization and interpretability in higher dimensional applications of LDA such as this.

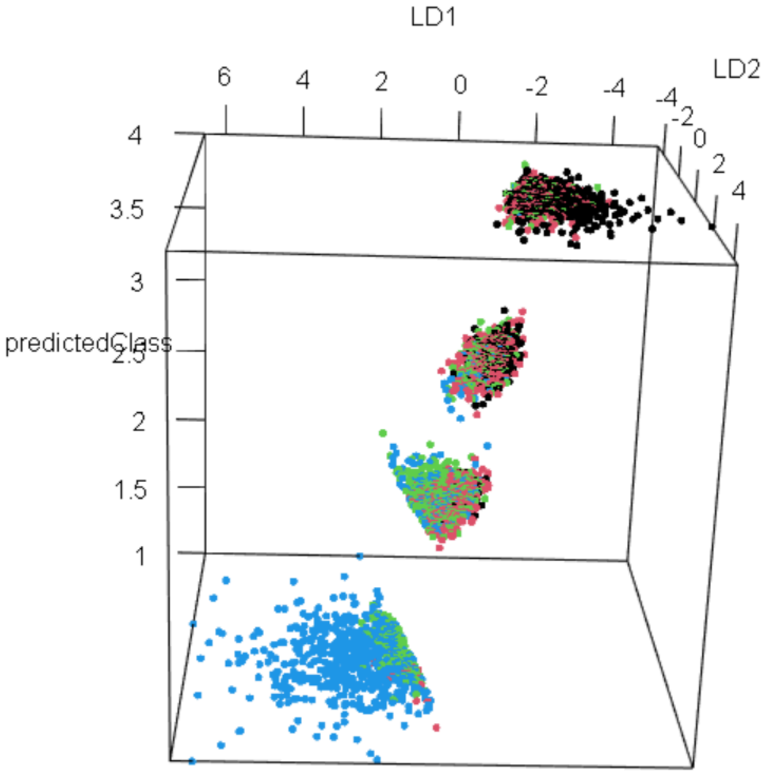
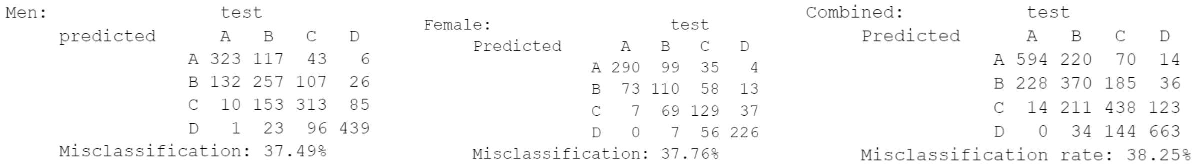


Fig. 13: 3-D plot of LD1 by LD2 split by predicted class.
Actual class is coded A = black, red = B, green = C, and blue = D.

It is clear that class D is the easiest to separate out prediction wise given the discriminant functions. In the combined model, D class has a misclassification rate of 24.8%. A class seems to be the next best with a misclassification rate of 28.1%. Classes B and C are much harder to predict with misclassification rates of 55.7% and 44.3% respectively. It is not clear where much of the variation and noise in our models predictive ability is derived from. Note how much of classes B and C are miss classified.

Polytomous Regression

We fit multinomial logistic regression models to predict class for each of the three groupings. The hope was to find a better fit with an approach more similar to regression than space separation and between group variance maximization of LDA. Being able to report coefficients without dimensional reduction and have a classification algorithm for each of the four distinct classes was a positive. LDA has the dimensionality reduction equation but there still must be a decision made about class whereas polytomous regression allows for more clear equations to classify. Below are the confusion matrices for the male, female, and combined groups as well as the misclassification rate reported.



```
Men:                    test
     predicted       A   B   C   D
                  A 323 117  43   6
                  B 132 257 107  26
                  C  10 153 313  85
                  D   1  23  96 439
     Misclassification: 37.49%
```

```
Female:                   test
     Predicted       A   B   C   D
                  A 290  99  35   4
                  B  73 110  58  13
                  C   7  69 129  37
                  D   0   7  56 226
     Misclassification: 37.76%
```

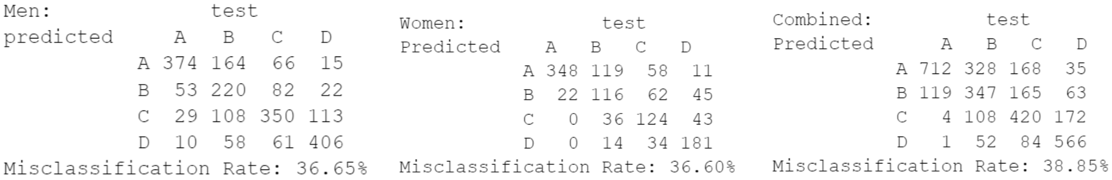
```
Combined:              test
     Predicted       A   B   C   D
                  A 594 220  70  14
                  B 228 370 185  36
                  C  14 211 438 123
                  D   0  34 144 663
     Misclassification rate: 38.25%
```

Fig. 14: Confusion matrices for the three LDA models.

When splitting by gender, the misclassification rate does improve slightly but this could just as well be a result of the randomization done in the cross validation partitioning than it is gender explaining variance. Considering just how close these three are, it is clear that gender is being accounted for by the other features in this model as well. (See full polytomous output on page 38).

Decision Trees

Since our data and classification goal is to separate fitness class by the variety of features, attempting to fit and then prune back a decision tree seems to be a sensical next-step. Below we report the confusion matrices and misclassification rates, and the actual decision tree graphs are on pages 30-31, in the appendix.



```
Men:                test
predicted       A   B   C   D
              A 374 164  66  15
              B  53 220  82  22
              C  29 108 350 113
              D  10  58  61 406
Misclassification Rate: 36.65%
```

```
Women:              test
Predicted      A   B   C   D
             A 348 119  58  11
             B  22 116  62  45
             C   0  36 124  43
             D   0  14  34 181
Misclassification Rate: 36.60%
```

```
Combined:             test
Predicted         A   B   C   D
                A 712 328 168  35
                B 119 347 165  63
                C   4 108 420 172
                D   1  52  84 566
Misclassification Rate: 38.85%
```

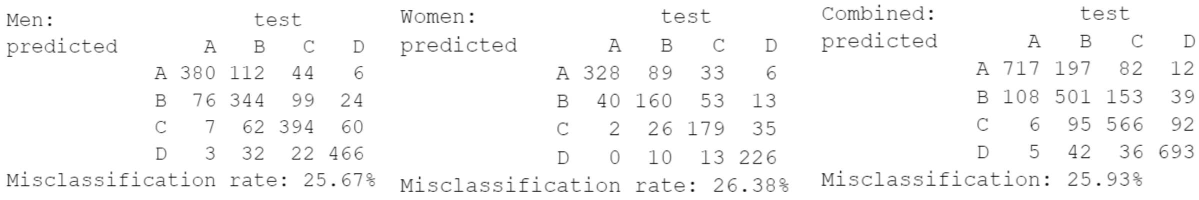
Fig. 15: Confusion matrices for the three decision tree models.

Again, the amount of variance explained by accounting for the gender term allows for slightly better prediction power, but the rate is still rather high at over a third incorrect when predicting. In the appendix, note that the tree splits have different coefficients which match the findings of varying sample means for the features split by men and women. Regardless of gender, the similar variable splits crop up around the same levels of the tree so this reinforces the finding that important variables are important because of their ability to classify fitness class more so than their ability to account for gender. Note that reporting these trees for future classification can be done as separate functions or combined with gender being the first split.

We were able to account for slightly more variation given a decision tree, however LDA, multinomial regression, and decision trees have done about the same predictively with these data. We wish to use a different algorithm geared for classification that could potentially make better predictions than the similar rates of the previous models.

Random Forest

We finally decided to attempt random forest models which have proven to be powerful in classification settings. Below, again, are the confusion matrices and misclassification rates for the three groupings.



```
Men:                 test
predicted       A   B   C   D
          A 380 112  44   6
          B  76 344  99  24
          C   7  62 394  60
          D   3  32  22 466
Misclassification rate: 25.67%
```

```
Women:               test
predicted       A   B   C   D
          A 328  89  33   6
          B  40 160  53  13
          C   2  26 179  35
          D   0  10  13 226
Misclassification rate: 26.38%
```

```
Combined:            test
predicted       A   B   C   D
          A 717 197  82  12
          B 108 501 153  39
          C   6  95 566  92
          D   5  42  36 693
Misclassification: 25.93%
```

Fig. 16: Confusion Matrices for Random Forest models.

Random Forest models see a 10% lower misclassification rate than that of the previous models. This is an important finding because we have been able to explain much more of the variation occurring in these data. Misclassifying one in four is much better than one in three so in terms of modeling, this was a great success. However, there is still a large portion of variance between these classes that has been left unexplained and it is likely due to systemic issues in the data rather than finding better methods. Being able to classify multi-factor problems becomes difficult but there are other conclusions to be made.

Conclusion

In all, we attribute our models inability to classify with high precision to a few factors. Foremost is that we were not able to access the unprocessed data set that contains additional features which could have likely been able to improve predictive power. Additionally, the nature of the classification criteria from the institution is unknown but it is our belief that it was done generally subjectively. Given, someone's judgment on a very fit or very unhealthy individual is probably realistic but defining the boundaries around classes B and C could contribute to their high misclassifications.

In terms of reporting our models, with the rate at which they misclassify, these likely are not ready to be used for more data. Having access to more features would be our biggest improvement if we were to conduct a new round of the data collection. Given how many subjects we had, the lack of explanation deriving from the features caused this predictive weakness of final models.

In all, given the data we had, we created models that were able to predict decently well. There was very high correct classification for classes A and D so fine tuning those ends was a positive. Uncovering and understanding the interaction gender had in this data set was a large obstacle but our proof allowed us a greater understanding of the data and modeling as a whole. Being able to parse apart data and understand how to gear models to best work on them is an important skill to have in machine learning. In all, we are proud of our analysis and work.

Bibliography

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2022). Chapter 2, Classification. In *An introduction to statistical learning: With applications in R* (2nd ed.). essay, Springer.

Humboldt-Universität zu Berlin | Geography Department. (n.d.). "*LDA and Model assessment in R.*" Retrieved May 7, 2023, from https://pages.cms.hu-berlin.de/EOL/gcg_quantitative-methods/Lab11_LDA_Model-assessment.html

Kukuroo3. (2022, June 29). Body performance data. Kaggle. Retrieved April 30, 2023 from https://www.kaggle.com/datasets/kukuroo3/body-performance-data

Korea Sports Promotion Foundation. (n.d.). *문화빅데이터 플랫폼*. 문화 빅데이터 - 포털- 전체 상품. Retrieved April 30, 2023 from https://www.bigdata-culture.kr/bigdata/user/data_market/detail.do?id=ace0aea7-5eee-48b9-b616-637365d665c1

Martinez-Taboada, Fernando; Redondo, Jose Ignacio (2020): Variable importance plot (mean decrease accuracy and mean decrease Gini).. PLOS ONE. Figure. https://doi.org/10.1371/journal.pone.0230799.g002

R-Code Appendix

Appendix A: R-Code

```r
library(randomForest)   # used for random forest models
library(caret)          # used for create data partition
library(glmnet)         # used for glmnet and cv.glmnet
library(MASS)           # lda() command from here
library(scatterplot3d)  # 3d plotting
library(rgl)            # used for 3d plotting
library(rpart)          # decisions trees
library(rpart.plot)     # plotting decision trees
library(nnet)           # multinom() command
```

```r
# load data
heal <- read.csv("C:~/bodyPerformance.csv", header = T, stringsAsFactors = T)

# remove outliers
heal <- heal[-c(10625, 736, 2658, 3356, 337, 3525, 8252, 8563, 9487, 9769), ]
heal$gender <- ifelse(heal$gender == "M", 1, 0)
attach(heal)

# split into training and test
set.seed(5)
heal.ind <- createDataPartition(heal$class, p = 0.75, list = F)
htrain <- heal[heal.ind, ]
htest <- heal[-heal.ind, ]

# split by gender
# health data men
hmen <- heal[heal$gender == 1,]
# health data women
hwo <- heal[heal$gender == 0,]

# training for the men
htrainMen <- htrain[htrain$gender == 1,]
# remove gender column
htrainMen <- htrainMen[,-2]
# test set for the men
htestMen <- htest[htest$gender == 1,]
# remove gender column
htestMen <- htestMen[,-2]

# training for the women
htrainWo <- htrain[htrain$gender == 0,]
# remove gender column
```

```
htrainWo <- htrainWo[,-2]
# test set for the women
htestWo <- htest[htest$gender == 0,]
# remove gender column
htestWo <- htestWo[,-2]
```

 # histograms loop (actual histograms on page 4 of Appendix B)

```
vartitles <- c("Age", "Height (in cm)", "Weigth (in kg)", "Body Fat
            Percentage", "Diastolic Blood Pressure (mmHg)", "Systolic Blood
            Pressure (mmHg)", "Grip Force (kg)", "Sit and Bend Forward (cm)",
            "Sit-Ups (max in 2min)", "Broad Jump (in cm)")

par(ask = T)
for(i in 1:10){
  Xmin <- min(min(hmen[, i]), min(hwo[, i]))
  Xmax <- max(max(hmen[, i]), max(hwo[, i]))
  hist(hmen[, i], freq = F, main = paste("Hist of Mens", vartitles[i]),
       xlab = vartitles[i], xlim = c(Xmin, Xmax), cex.main = 0.9)
  hist(hwo[, i], freq = F, main = paste("Hist of Womens", vartitles[i]),
       xlab = vartitles[i], xlim = c(Xmin, Xmax), cex.main = 0.9)
}
```

 # variable importance plot with matching RF model

```
set.seed(1)
rand <- randomForest(as.factor(class) ~ ., data = htrain, mtry =
5, ntree = 500,importance = T)

pred <- predict(rand, htest, response = "class")
tableRand <- table(pred, htest$class)
tableRand
RANDrate <- (sum(tableRand[c(2, 3, 4, 5, 7, 8, 9, 10, 12, 13,
14, 15)]))/length(htest$class)
RANDrate
varImpPlot(rand,main = "Random Forest Variable Importance")
```

 # class by predictor plots split by gender (plots on page 5)

```
varnames <- hmen %>% names()

classNumMen = 5-as.numeric(hmen$class)
classNumWo = 5-as.numeric(hwo$class)
classNum = 5-as.numeric(class)

par(ask = T)
for(i in 1:10){
```

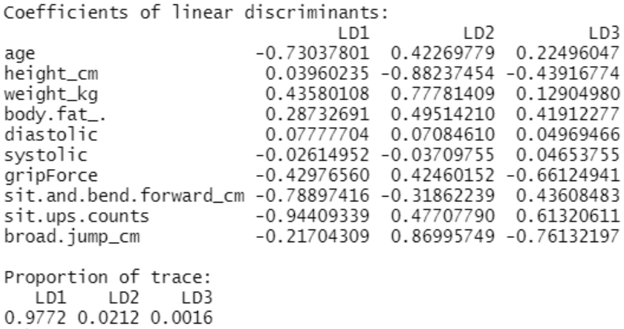
```
  Xmin <- min(min(hmen[, i]), min(hwo[, i]))
  Xmax <- max(max(hmen[, i]), max(hwo[, i]))
  plot(heal[, varnames[i]], classNum, cex.main = 0.9,
       main = paste("Class by ", vartitles[i]),
       xlab = vartitles[i], ylab="Class (A = 4, D = 1)",
       type = "n",ylim = c(.8,4.5), xlim = c(Xmin, Xmax))
  points(hmen[, i], defaultjitter <-
.25+classNumMen+runif(length(classNumMen),-.1,.1),col="light blue",pch =
20,cex =.5)
  points(hwo[, i],defaultjitter <-
classNumWo+runif(length(classNumWo),-.1,.1),col="orange",pch=20,cex = .5)
  lines(lowess(hmen[, i],classNumMen),lwd =3, col = "blue")
  lines(lowess(hwo[,i], classNumWo),lwd =3, col = "red")
}

par(ask=F)

# gender ~ all the features (not class)

healnoclass <- heal[, -12]
set.seed(1)
summary(genderLogit <- glm(gender ~ ., family = "binomial",
                        data = healnoclass))

# test/train table for that model

heal.ind <- createDataPartition(healnoclass$gender, p = 0.75, list = F)
gendertrain <- healnoclass[heal.ind, ]
gendertest <- healnoclass[-heal.ind, ]
summary(modell <- glm(gender ~ ., family = "binomial", data = gendertrain))
predict_gender <- predict(modell, newdata = gendertest, type = "response")

prediction <- ifelse((predict_gender > 0.5) == T, 1, 0)
validation <- gendertest[, 2]
gendertable <- table(prediction, validation)
gendertable

# multinomial for that model
multinomtrain <- multinom(class ~ ., data = as.data.frame(htrain))
summary(multinomtrain)

multipredict <- predict(multinomtrain, newdata = as.data.frame(htest), type =
"class")
tablemulti <- table(multipredict, htest[, 12])
multirate <- (sum(tablemulti[c(2, 3, 4, 5, 7, 8, 9, 10, 12, 13, 14,
15)]))/dim(ytest)[1]
```

```
########## LDA ##########
### Scale data for lda
scaletrain <- scale(htrain[, -c(2, 12)])
htrainclass <- htrain[, 12]
htrain <- cbind(scaletrain[, 1], htrain[, 2], scaletrain[, 2:10])
htrain <- as.data.frame(htrain)


htrain <- cbind(htrain, htrainclass)
colnames(htrain) <- c("age", "gender", "height_cm", "weight_kg", "body.fat_.",
"diastolic",
                      "systolic", "gripForce", "sit.and.bend.forward_cm",
"sit.ups.counts",
                      "broad.jump_cm", "class")


scaletest <- scale(htest[, -c(2, 12)])
htestclass <- htest[, 12]
htest <- cbind(scaletest[, 1], htest[, 2], scaletest[, 2:10])
htest <- as.data.frame(htest)
htest <- cbind(htest, htestclass)
colnames(htest) <- c("age", "gender", "height_cm", "weight_kg", "body.fat_.",
"diastolic",
                     "systolic", "gripForce", "sit.and.bend.forward_cm",
"sit.ups.counts",
                     "broad.jump_cm", "class")

htrainMen <- htrain[htrain$gender == 1,]
htestMen <- htest[htest$gender == 1,]
htrainMen <- htrainMen[,-2]
htestMen <- htestMen[,-2]

htrainWo <- htrain[htrain$gender == 0,]
htestWo <- htest[htest$gender == 0,]
htrainWo <- htrainWo[,-2]
htestWo <- htestWo[,-2]

# Code for the models
#men
set.seed(1)
lda_modelMen <- lda(class ~ ., data = htrainMen)
lda_predMen <- predict(lda_modelMen, newdata = htestMen)

#women
set.seed(1)
lda_modelWo <- lda(class ~ ., data = htrainWo)
lda_predWo <- predict(lda_modelWo, newdata = htestWo)
```

```
#combined
set.seed(1)
lda_model <- lda(class ~ ., data = htrain)
lda_pred <- predict(lda_model, newdata = htest)
```

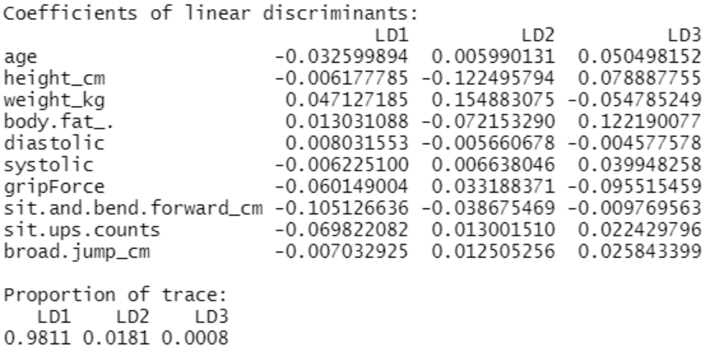
```
# Code for confusion matrices
ldatableMen <- table(lda_predMen$class, htestMen$class)
ldarateMen <- (sum(ldatableMen[c(2, 3, 4, 5, 7, 8, 9, 10, 12, 13, 14,
15)]))/length(htestMen$class)
ldarateMen

ldatableWo <- table(lda_predWo$class, htestWo$class)
ldarateWo <- (sum(ldatableWo[c(2, 3, 4, 5, 7, 8, 9, 10, 12, 13, 14,
15)]))/length(htestWo$class)
ldarateWo

ldatable <- table(lda_pred$class, htest$class)
ldarate <- (sum(ldatable[c(2, 3, 4, 5, 7, 8, 9, 10, 12, 13, 14,
15)]))/length(htest$class)
ldarate
```

```
# the model outputs with trace and coefficients
## Men model
lda_modelMen
```



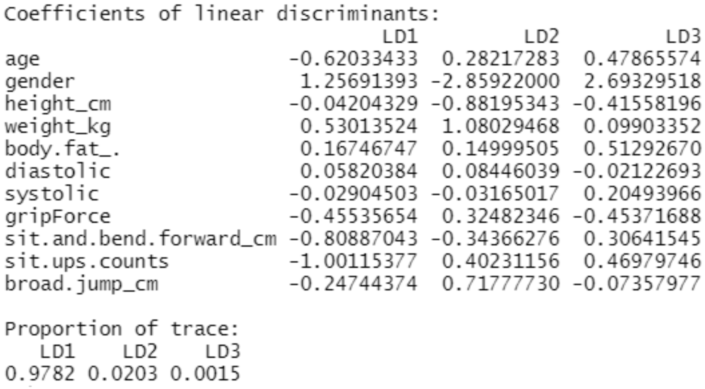
```
Coefficients of linear discriminants:
                               LD1         LD2         LD3
age                    -0.73037801  0.42269779  0.22496047
height_cm               0.03960235 -0.88237454 -0.43916774
weight_kg               0.43580108  0.77781409  0.12904980
body.fat_.              0.28732691  0.49514210  0.41912277
diastolic               0.07777704  0.07084610  0.04969466
systolic               -0.02614952 -0.03709755  0.04653755
gripForce              -0.42976560  0.42460152 -0.66124941
sit.and.bend.forward_cm -0.78897416 -0.31862239  0.43608483
sit.ups.counts         -0.94409339  0.47707790  0.61320611
broad.jump_cm          -0.21704309  0.86995749 -0.76132197

Proportion of trace:
   LD1    LD2    LD3
0.9772 0.0212 0.0016
```

```
## women model
lda_modelWo
```



```
Coefficients of linear discriminants:
                                 LD1          LD2          LD3
age                     -0.032599894  0.005990131  0.050498152
height_cm               -0.006177785 -0.122495794  0.078887755
weight_kg                0.047127185  0.154883075 -0.054785249
body.fat_.               0.013031088 -0.072153290  0.122190077
diastolic                0.008031553 -0.005660678 -0.004577578
systolic                -0.006225100  0.006638046  0.039948258
gripForce               -0.060149004  0.033188371 -0.095515459
sit.and.bend.forward_cm -0.105126636 -0.038675469 -0.009769563
sit.ups.counts          -0.069822082  0.013001510  0.022429796
broad.jump_cm           -0.007032925  0.012505256  0.025843399

Proportion of trace:
   LD1    LD2    LD3
0.9811 0.0181 0.0008
```

```
## combined model:
lda_model
```



```
Coefficients of linear discriminants:
                                LD1         LD2         LD3
age                     -0.62033433  0.28217283  0.47865574
gender                   1.25691393 -2.85922000  2.69329518
height_cm               -0.04204329 -0.88195343 -0.41558196
weight_kg                0.53013524  1.08029468  0.09903352
body.fat_.               0.16746747  0.14999505  0.51292670
diastolic                0.05820384  0.08446039 -0.02122693
systolic                -0.02904503 -0.03165017  0.20493966
gripForce               -0.45535654  0.32482346 -0.45371688
sit.and.bend.forward_cm -0.80887043 -0.34366276  0.30641545
sit.ups.counts          -1.00115377  0.40231156  0.46979746
broad.jump_cm           -0.24744374  0.71777730 -0.07357977

Proportion of trace:
   LD1    LD2    LD3 
0.9782 0.0203 0.0015 
```

 # lda pairs graph

```
pairs(lda_pred$x, pch = 20, col = as.numeric(htest$class),
      main = "Pairs Plot of LD1, LD2, and LD3\nFor Combined LDA Model",
cex.main = 0.95)
```

 # 3d plot

```
LD1 <- preds[, 1]
LD2 <- preds[, 2]
predictedClass <- (5 - (lda_pred$class %>% as.numeric())) %>% as.factor()
plot3d(LD1, LD2, predictedClass, col = as.numeric(htest$class), size = 5,)
```

Reload unscaled data

```
htrain <- heal[heal.ind, ]
htest <- heal[-heal.ind, ]

hmen <- heal[heal$gender == 1,]
hwo <- heal[heal$gender == 0,]

htrainMen <- htrain[htrain$gender == 1,]
htestMen <- htest[htest$gender == 1,]
htrainMen <- htrainMen[,-2]
htestMen <- htestMen[,-2]

htrainWo <- htrain[htrain$gender == 0,]
htestWo <- htest[htest$gender == 0,]
htrainWo <- htrainWo[,-2]
htestWo <- htestWo[,-2]
```

```
########## polytomous regression ##########

# Code for the three models
#men
set.seed(1)
polyLMtrainMen <- glmnet(x = as.matrix(htrainMen[, 1:10]),y = htrainMen[, 11],
                      family = "multinomial",
                      lambda = 0,
                      type.multinomial = "ungrouped")

#women
set.seed(1)
polyLMtrainWo <- glmnet(x = as.matrix(htrainWo[, 1:10]), y = htrainWo[, 11],
                      family = "multinomial",
                      lambda = 0,
                      type.multinomial = "ungrouped")

#combined
set.seed(1)
polyLMtrain <- glmnet(x = as.matrix(htrain[,1:11]), y = htrain[,12] ,
                              family = "multinomial",
                              lambda = 0,
                              type.multinomial = "ungrouped")

# Code for confusion matrices
# predict Men
polyLMpredictMen <- predict(polyLMtrainMen, newx = as.matrix(htestMen[,
1:10]), type = "class")
#table
tablePOLYMen <- table(polyLMpredictMen, as.matrix(htestMen[, 11]))
tablePOLYMen
polyLMrateMen <- (sum(tablePOLYMen[c(2, 3, 4, 5, 7, 8, 9, 10, 12, 13, 14,
15)]))/length(htestMen$class)
polyLMrateMen

# predict women
polyLMpredictWo <- predict(polyLMtrainWo, newx = as.matrix(htestWo[, 1:10]),
type = "class")
#table
tablePOLYWo <- table(polyLMpredictWo, htestWo[, 11])
tablePOLYWo
polyLMrateWo <- (sum(tablePOLYWo[c(2, 3, 4, 5, 7, 8, 9, 10, 12, 13, 14,
15)]))/length(htestWo$class)
polyLMrateWo

# predict combined
```

```
polyLMpredict <- predict(polyLMtrain, newx = as.matrix(htest[1:11]), type =
"class")
#table
tablePOLY <- table(polyLMpredict, htest[,12])
tablePOLY
polyLMrate <- (sum(tablePOLY[c(2, 3, 4, 5, 7, 8, 9, 10, 12, 13, 14,
15)]))/length(htest$class)
polyLMrate

# Polytomous model coefficients (coefficients on page 10 of
Appendix B)

coef(polyLMtrainMen)
coef(polyLMtrainWo)
coef(polyLMtrain)

########## decision tree ##########

# Code for the tree models
# Men Tree:
set.seed(1)
men.tree <- rpart(class~.,data = htrainMen,
                       method='class',
                       control=rpart.control(minsplit=300,cp=.001,xval=10))

best <- men.tree$cptable[which.min(men.tree$cptable[,"xerror"]),"CP"]

# store pruned tree
men.tree.pruned <- prune(men.tree, cp=best)

# Women tree:
set.seed(1)
women.tree <- rpart(class~.,data = htrainWo,
                       method='class',
                       control=rpart.control(minsplit=300,cp=.001,xval=10))

best <- women.tree$cptable[which.min(women.tree$cptable[,"xerror"]),"CP"]

# store pruned tree
women.tree.pruned <- prune(women.tree, cp=best)

#combined tree
set.seed(1)
full.tree <- rpart(class~.,data = htrain,
                    method='class',
                    control=rpart.control(minsplit=300,cp=.001,xval=10))

best <- full.tree$cptable[which.min(full.tree$cptable[,"xerror"]),"CP"]
```

```
# store pruned tree
full.tree.pruned <- prune(full.tree, cp=best)

# code for the tree confusion matrices
prunePredictMen <- predict(men.tree.pruned,newdata = htestMen,type="class")
treeTable <- table(prunePredictMen,htestMen$class)
treeMCRateMen <- (sum(treeTable[c(2, 3, 4, 5, 7, 8, 9, 10, 12, 13, 14,
15)]))/length(htestMen$class)
treeMCRateMen

prunePredictWo <- predict(women.tree.pruned,newdata = htestWo,type="class")
treeTable <- table(prunePredictWo,htestWo$class)
treeMCRateWo <- (sum(treeTable[c(2, 3, 4, 5, 7, 8, 9, 10, 12, 13, 14,
15)]))/length(htestWo$class)
treeMCRateWo

prunePredict <- predict(full.tree.pruned,newdata = htest,type="class")
treeTable <- table(prunePredict,htest$class)
treeMCRate <- (sum(treeTable[c(2, 3, 4, 5, 7, 8, 9, 10, 12, 13, 14,
15)]))/length(htest$class)
treeMCRate

# code for tree plots (plots on page 2 of Appendix B)
prp(men.tree.pruned,
    faclen=0,
    extra=1,
    roundint=F,
    digits=5)
prp(women.tree.pruned,
    faclen=0,
    extra=1,
    roundint=F,
    digits=5)
prp(full.tree.pruned,
    faclen=0,
    extra=1,
    roundint=F,
    digits=5)
########## Random Forest ##########

# Code for RandForest Models
set.seed(1)
randMen <- randomForest(as.factor(class) ~ ., data = htrainMen, mtry = 5,
ntree = 500,importance = T)
set.seed(1)
randWo <- randomForest(as.factor(class) ~ ., data = htrainWo, mtry = 5, ntree
= 500,importance = T)
```

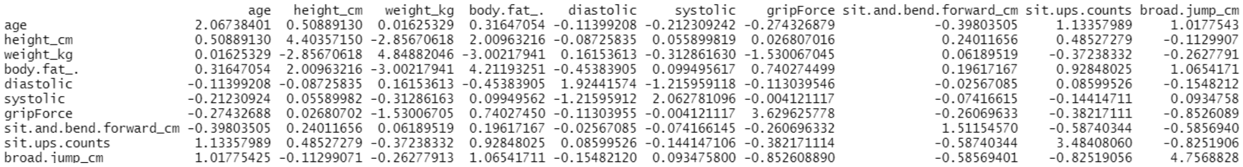
```
set.seed(1)
rand <- randomForest(as.factor(class) ~ ., data = htrain, mtry = 5, ntree =
500,importance = T)

# Code for RandForest confusion matrices
predMen <- predict(rand, htestMen, response = "class")
tableRandMen <- table(pred, htestMen$class)
tableRandMen
RANDrateMen <- (sum(tableRandMen[c(2, 3, 4, 5, 7, 8, 9, 10, 12, 13, 14,
15)]))/length(htestMen$class)
RANDrateMen

predWo <- predict(randWo, htestWo, response = "class")
tableRandWo <- table(predWo, htestWo$class)
tableRandWo
RANDrateWo <- (sum(tableRandWo[c(2, 3, 4, 5, 7, 8, 9, 10, 12, 13, 14,
15)]))/length(htestWo$class)
RANDrateWo

pred <- predict(rand, htest, response = "class")
tableRand <- table(pred, htest$class)
tableRand
RANDrate <- (sum(tableRand[c(2, 3, 4, 5, 7, 8, 9, 10, 12, 13, 14,
15)]))/length(htest$class)
RANDrate
```

Plotting Appendix

Appendix B: R-Code

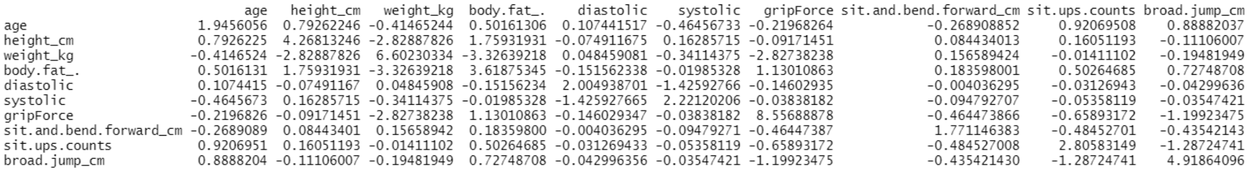
`Inverse Covariance matrices`

Male:



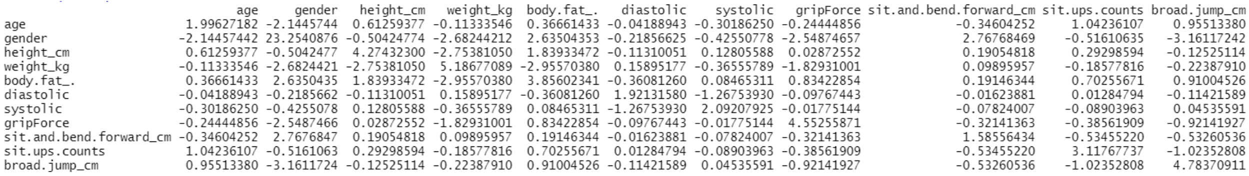
```
                                age   height_cm   weight_kg  body.fat_.   diastolic     systolic    gripForce sit.and.bend.forward_cm sit.ups.counts broad.jump_cm
age                      2.06738401  0.50889130  0.01625329  0.31647054 -0.11399208 -0.212309242 -0.274326879             -0.39803505     1.13357989     1.0177543
height_cm                0.50889130  4.40357150 -2.85670618  2.00963216 -0.08725835  0.055899819  0.026807016              0.24011656     0.48527279    -0.1129907
weight_kg                0.01625329 -2.85670618  4.84882046 -3.00217941  0.16153613 -0.312861630 -1.530067045              0.06189519    -0.37238332    -0.2627791
body.fat_.               0.31647054  2.00963216 -3.00217941  4.21193251 -0.45383905  0.099495617  0.740274499              0.19617167     0.92848025     1.0654171
diastolic               -0.11399208 -0.08725835  0.16153613 -0.45383905  1.92441574 -1.215959118 -0.113039546             -0.02567085     0.08599526    -0.1548212
systolic                -0.21230924  0.05589982 -0.31286163  0.09949562 -1.21595912  2.062781096 -0.004121117             -0.07416615    -0.14414711     0.0934758
gripForce               -0.27432688  0.02680702 -1.53006705  0.74027450 -0.11303955 -0.004121117  3.629625778             -0.26069633    -0.38217111    -0.8526089
sit.and.bend.forward_cm -0.39803505  0.24011656  0.06189519  0.19617167 -0.02567085 -0.074166145 -0.260696332              1.51154570    -0.58740344    -0.5856940
sit.ups.counts           1.13357989  0.48527279 -0.37238332  0.92848025  0.08599526 -0.144147106 -0.382171114             -0.58740344     3.48408060    -0.8251906
broad.jump_cm            1.01775425 -0.11299071 -0.26277913  1.06541711 -0.15482120  0.093475800 -0.852608890             -0.58569401    -0.82519056     4.7568828
```

Female:



```
                               age   height_cm   weight_kg  body.fat_.    diastolic    systolic   gripForce sit.and.bend.forward_cm sit.ups.counts broad.jump_cm
age                      1.9456056  0.79262246 -0.41465244  0.50161306  0.107441517 -0.46456733 -0.21968264            -0.268908852     0.92069508    0.88882037
height_cm                0.7926225  4.26813246 -2.82887826  1.75931931 -0.074911675  0.16285715 -0.09171451             0.084434013     0.16051193   -0.11106007
weight_kg               -0.4146524 -2.82887826  6.60230334 -3.32639218  0.048459081 -0.34114375 -2.82738238             0.156589424    -0.01411102   -0.19481949
body.fat_.               0.5016131  1.75931931 -3.32639218  3.61875345 -0.151562338 -0.01985328  1.13010863             0.183598001     0.50264685    0.72748708
diastolic                0.1074415 -0.07491167  0.04845908 -0.15156234  2.004938701 -1.42592766 -0.14602935            -0.004036295    -0.03126943   -0.04299636
systolic                -0.4645673  0.16285715 -0.34114375 -0.01985328 -1.425927665  2.22120206 -0.03838182            -0.094792707    -0.05358119   -0.03547421
gripForce               -0.2196826 -0.09171451 -2.82738238  1.13010863 -0.146029347 -0.03838182  8.55688878            -0.464473866    -0.65893172   -1.19923475
sit.and.bend.forward_cm -0.2689089  0.08443401  0.15658942  0.18359800 -0.004036295 -0.09479271 -0.46447387             1.771146383    -0.48452701   -0.43542143
sit.ups.counts           0.9206951  0.16051193 -0.01411102  0.50264685 -0.031269433 -0.05358119 -0.65893172            -0.484527008     2.80583149   -1.28724741
broad.jump_cm            0.8888204 -0.11106007 -0.19481949  0.72748708 -0.042996356 -0.03547421 -1.19923475            -0.435421430    -1.28724741    4.91864096
```

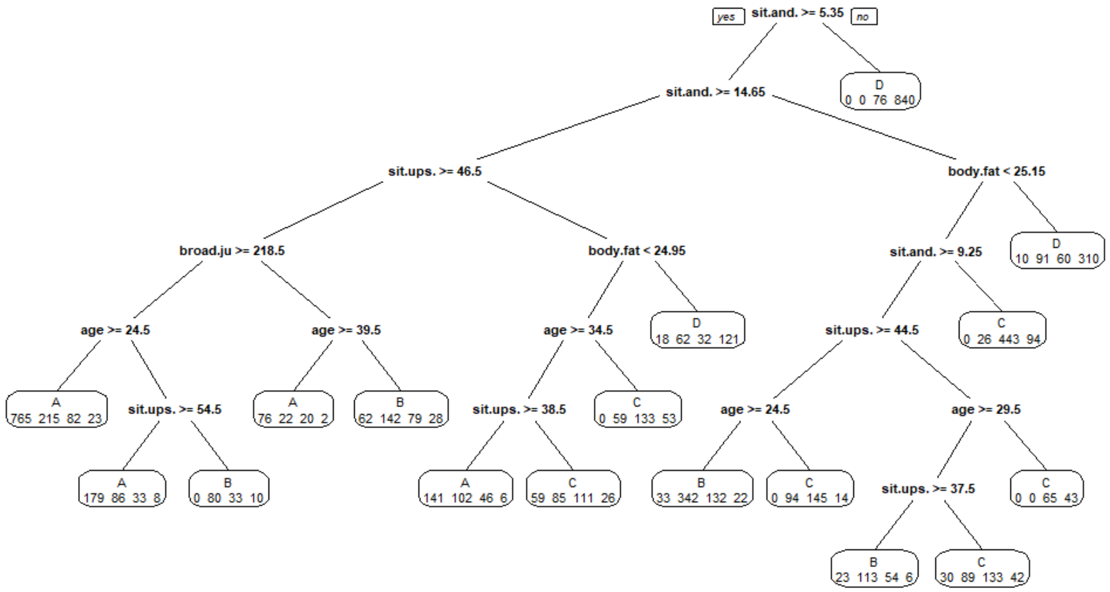
Combined:



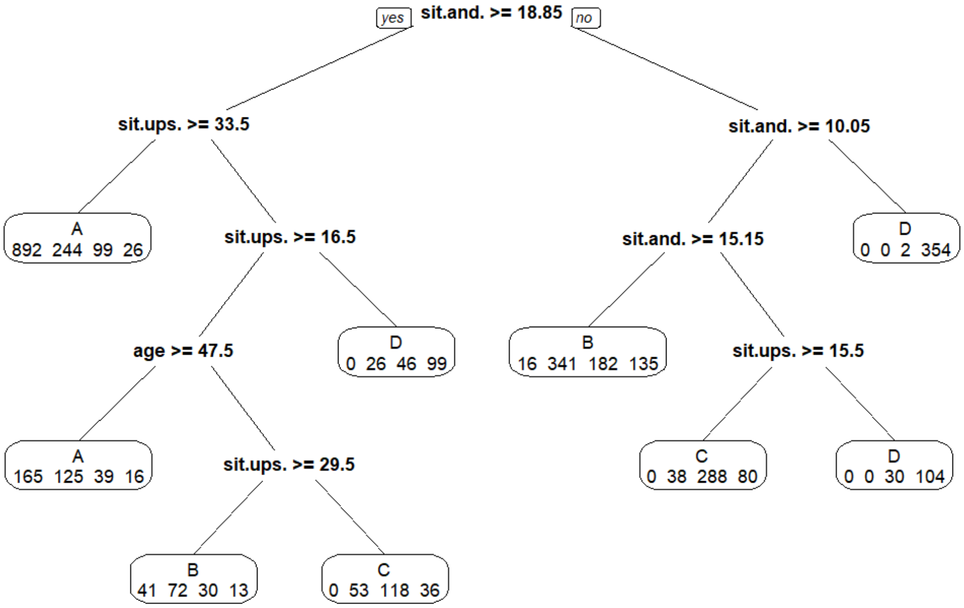
```
                                age     gender   height_cm   weight_kg  body.fat_.   diastolic    systolic   gripForce sit.and.bend.forward_cm sit.ups.counts broad.jump_cm
age                      1.99627182 -2.1445744  0.61259377 -0.11333546  0.36661433 -0.04188943 -0.30186250 -0.24444856             -0.34604252     1.04236107    0.95513380
gender                  -2.14457442 23.2540876 -0.50424774 -2.68244212  2.63504353 -0.21856625 -0.42550778 -2.54874657              2.76768469    -0.51610635   -3.16117242
height_cm                0.61259377 -0.5042477  4.27432300 -2.75381050  1.83933472 -0.11310051  0.12805588  0.02872552              0.19054818     0.29298594   -0.12525114
weight_kg               -0.11333546 -2.6824421 -2.75381050  5.18677089 -2.95570380  0.15895177 -0.36555789 -1.82931001              0.09895957    -0.18577816   -0.22387910
body.fat_.               0.36661433  2.6350435  1.83933472 -2.95570380  3.85602341 -0.36081260  0.08465311  0.83422854              0.19146344     0.70255671    0.91004526
diastolic               -0.04188943 -0.2185662 -0.11310051  0.15895177 -0.36081260  1.92131580 -1.26753930 -0.09767443             -0.01623881     0.01284794   -0.11421589
systolic                -0.30186250 -0.4255078  0.12805588 -0.36555789  0.08465311 -1.26753930  2.09207925 -0.01775144             -0.07824007    -0.08903963    0.04535591
gripForce               -0.24444856 -2.5487466  0.02872552 -1.82931001  0.83422854 -0.09767443 -0.01775144  4.55255871             -0.32141363    -0.38561909   -0.92141927
sit.and.bend.forward_cm -0.34604252  2.7676847  0.19054818  0.09895957  0.19146344 -0.01623881 -0.07824007 -0.32141363              1.58556434    -0.53455220   -0.53260536
sit.ups.counts           1.04236107 -0.5161063  0.29298594 -0.18577816  0.70255671  0.01284794 -0.08903963 -0.38561909             -0.53455220     3.11767737   -1.02352808
broad.jump_cm            0.95513380 -3.1611724 -0.12525114 -0.22387910  0.91004526 -0.11421589  0.04535591 -0.92141927             -0.53260536    -1.02352808    4.78370911
```

Decision Trees:

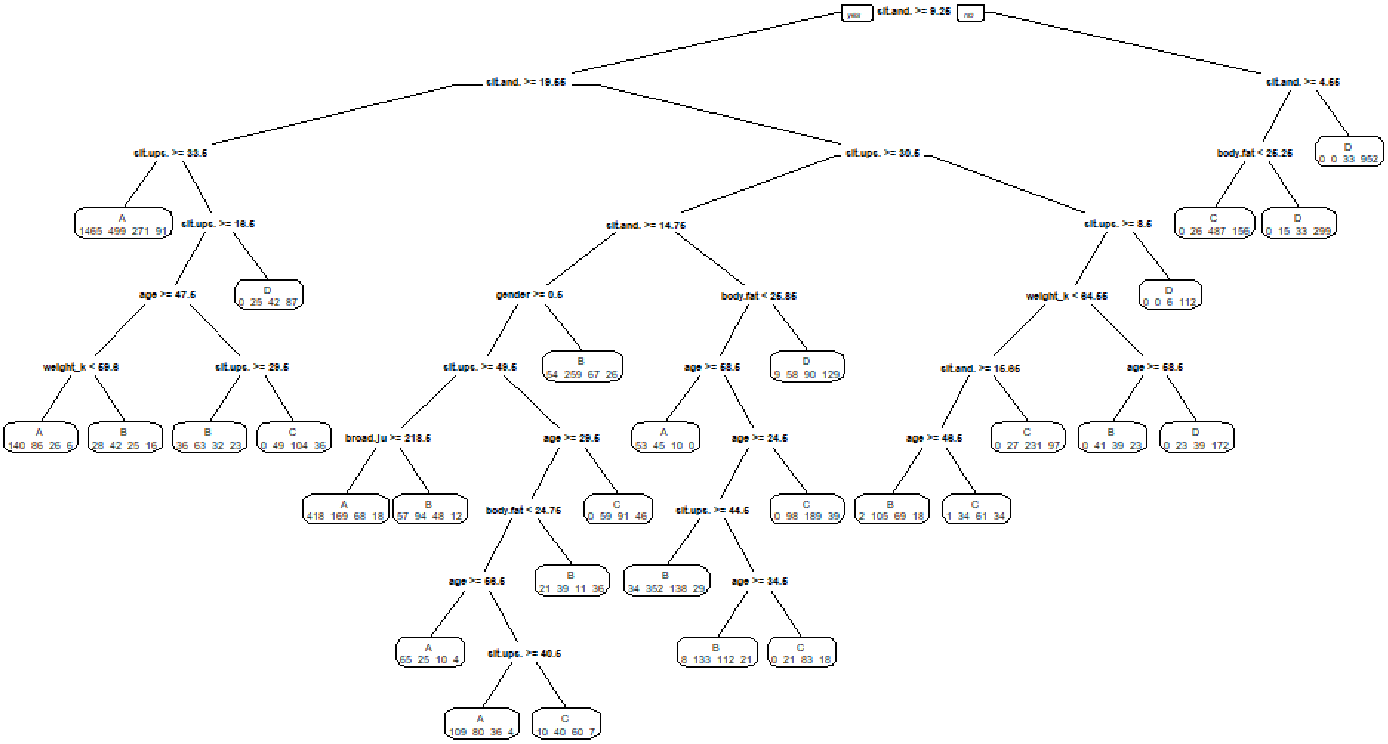
Men:



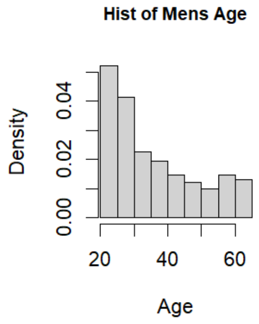
Women:

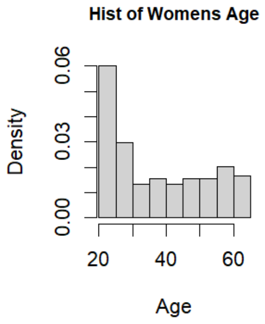


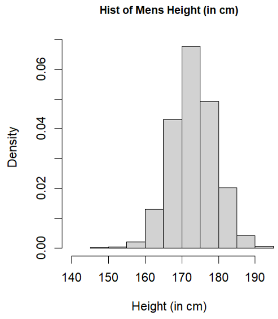
Combined:

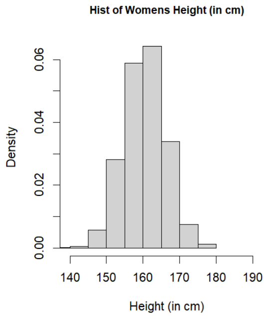


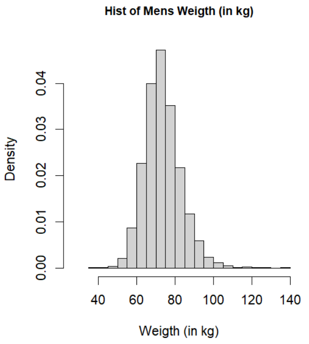
Histograms:

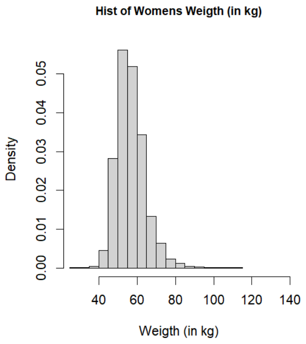


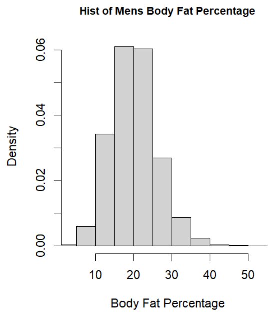


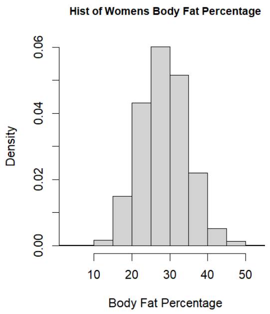


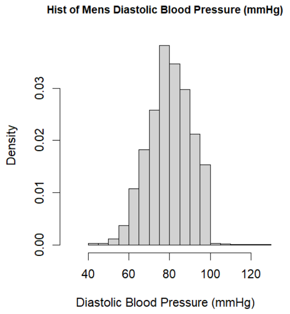


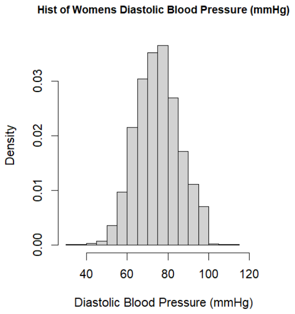

Hist of Mens Weigth (in kg)
Density
0.00
0.01
0.02
0.03
0.04
40
60
80
100
120
140
Weigth (in kg)

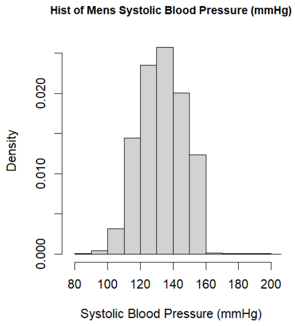

Hist of Womens Weigth (in kg)
Density
0.00
0.01
0.02
0.03
0.04
0.05
40
60
80
100
120
140
Weigth (in kg)

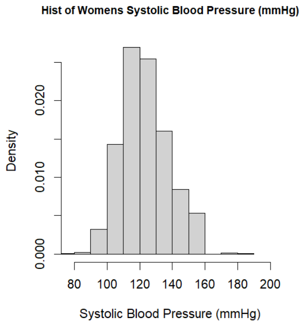

Hist of Mens Body Fat Percentage
Density
0.00
0.02
0.04
0.06
10
20
30
40
50
Body Fat Percentage

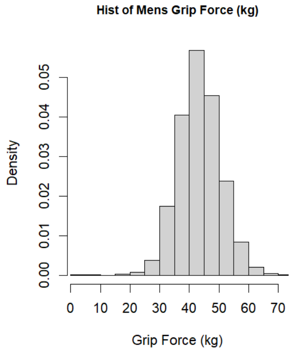

Hist of Womens Body Fat Percentage
Density
0.00
0.02
0.04
0.06
10
20
30
40
50
Body Fat Percentage

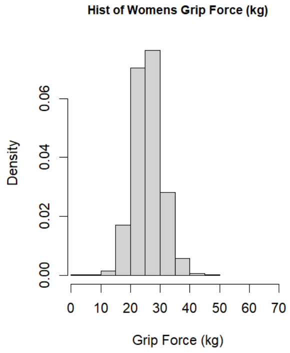

Hist of Mens Diastolic Blood Pressure (mmHg)
Density
0.00
0.01
0.02
0.03
40
60
80
100
120
Diastolic Blood Pressure (mmHg)

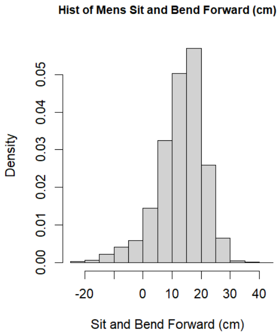

Hist of Womens Diastolic Blood Pressure (mmHg)
Density
0.00
0.01
0.02
0.03
40
60
80
100
120
Diastolic Blood Pressure (mmHg)

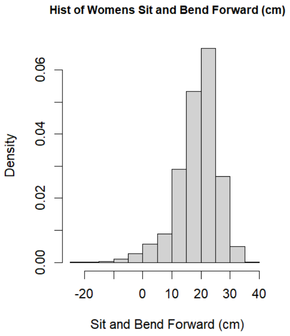

Hist of Mens Systolic Blood Pressure (mmHg)
Density
0.000
0.010
0.020
80
100
120
140
160
180
200
Systolic Blood Pressure (mmHg)

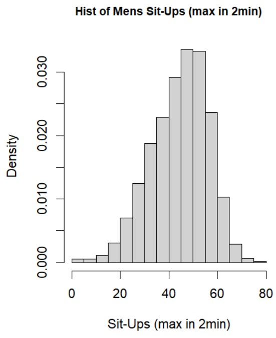

Hist of Womens Systolic Blood Pressure (mmHg)
Density
0.000
0.010
0.020
80
100
120
140
160
180
200
Systolic Blood Pressure (mmHg)

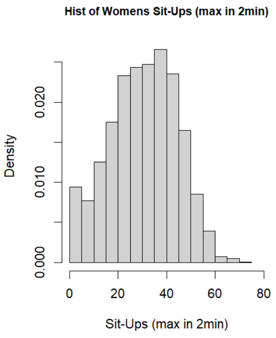

Hist of Mens Grip Force (kg)
Density
0.00
0.01
0.02
0.03
0.04
0.05
0
10
20
30
40
50
60
70
Grip Force (kg)


Hist of Womens Grip Force (kg)
Density
0.00
0.02
0.04
0.06
0
10
20
30
40
50
60
70
Grip Force (kg)

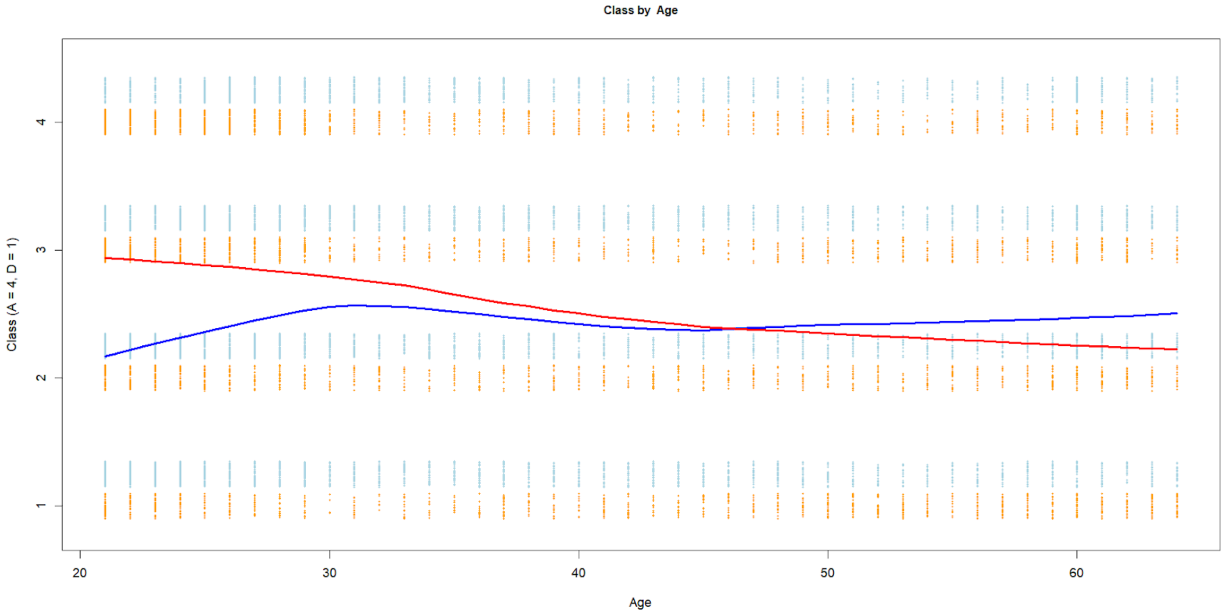

Hist of Mens Sit and Bend Forward (cm)
Density
0.00
0.01
0.02
0.03
0.04
0.05
-20
0
10
20
30
40
Sit and Bend Forward (cm)

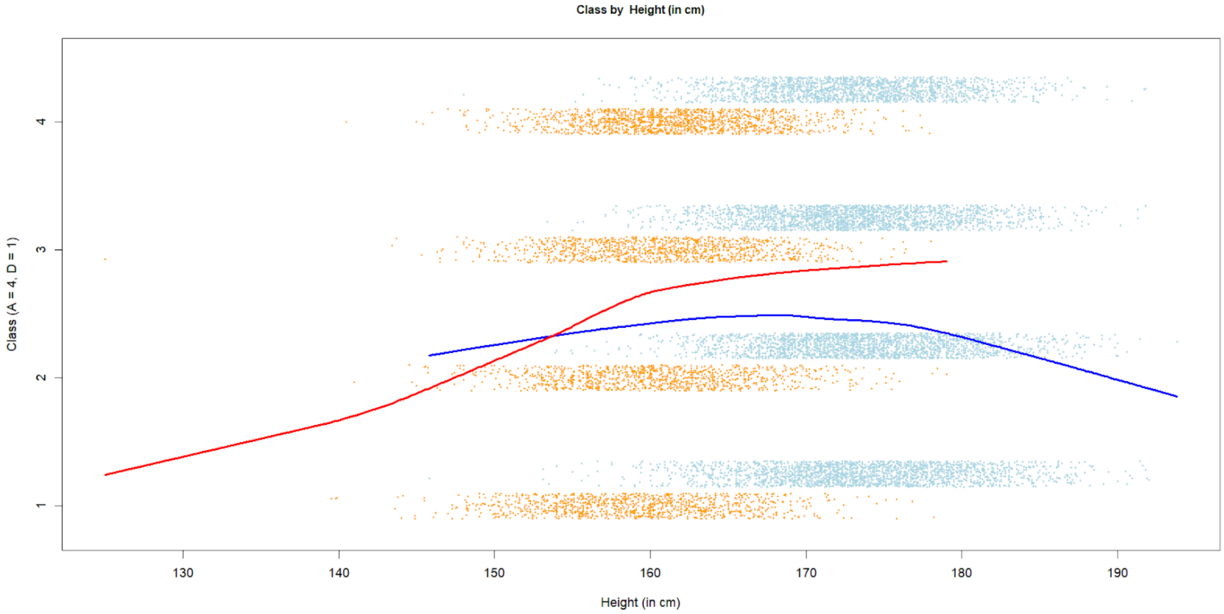

Hist of Womens Sit and Bend Forward (cm)
Density
0.00
0.02
0.04
0.06
-20
0
10
20
30
40
Sit and Bend Forward (cm)

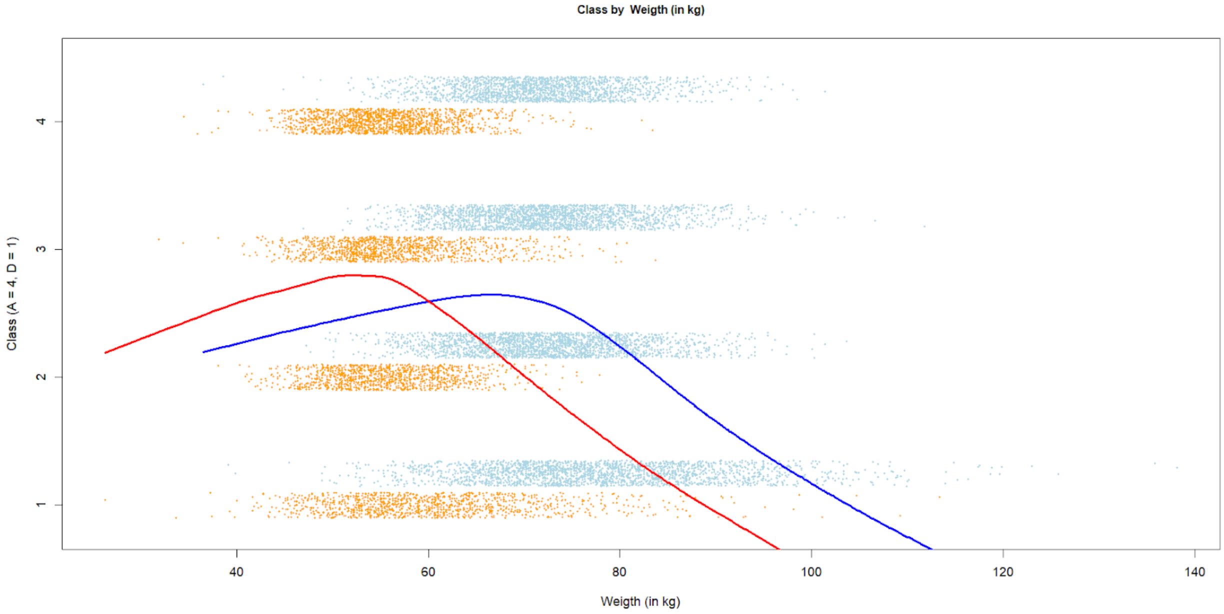

Hist of Mens Sit-Ups (max in 2min)
Density
0.000
0.010
0.020
0.030
0
20
40
60
80
Sit-Ups (max in 2min)

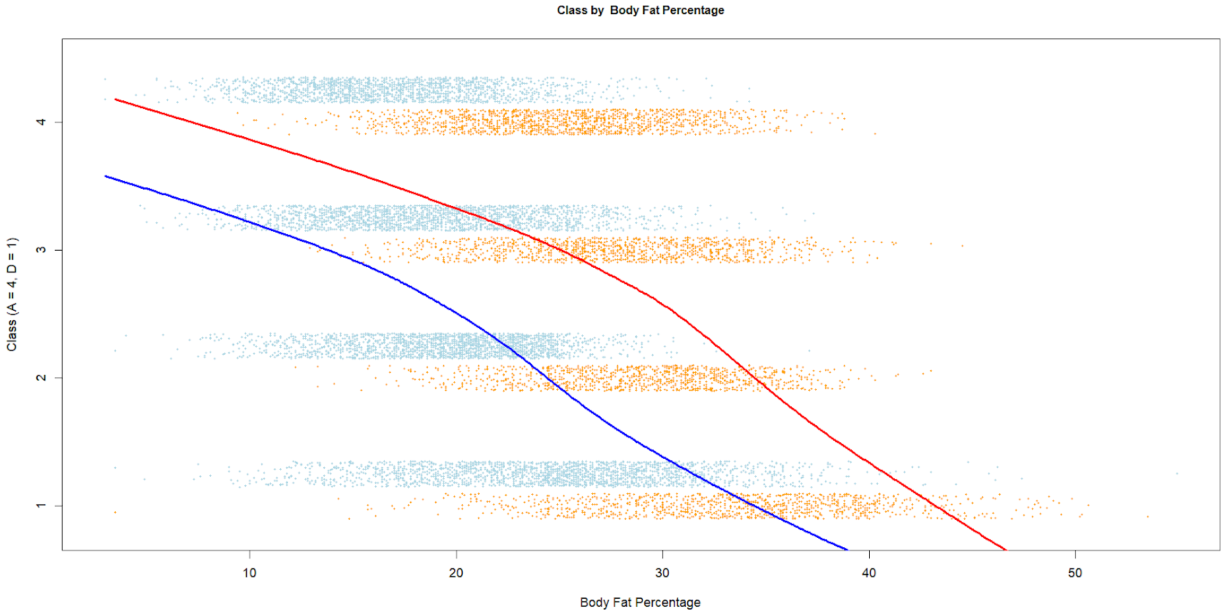

Hist of Womens Sit-Ups (max in 2min)
Density
0.000
0.010
0.020
0
20
40
60
80
Sit-Ups (max in 2min)

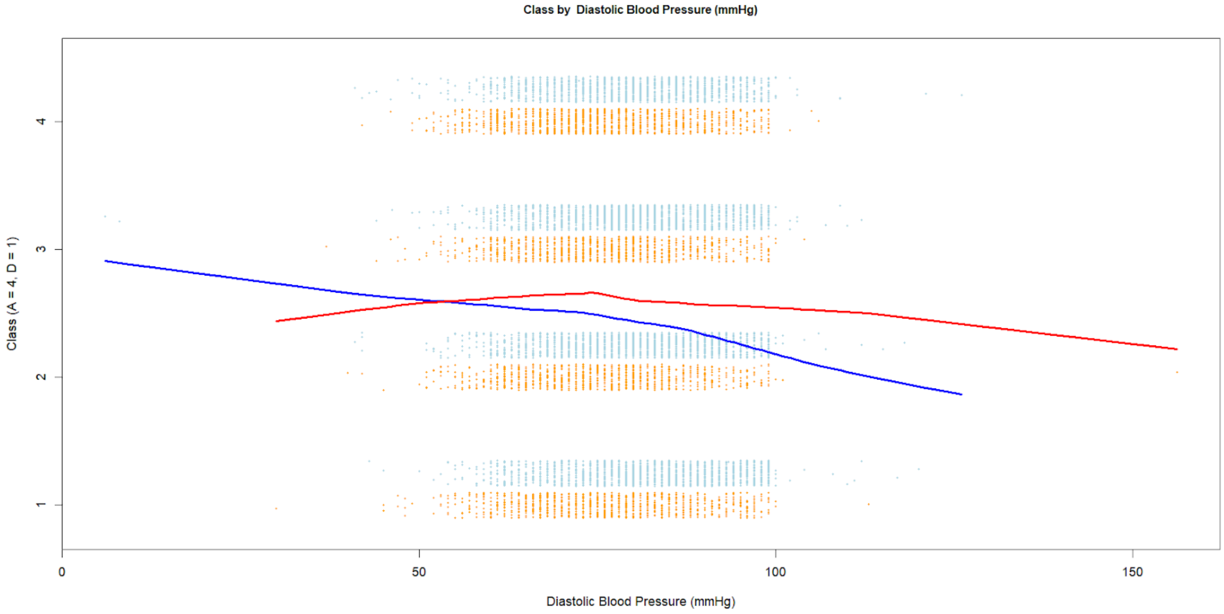
Class by Predictor plots split by gender:

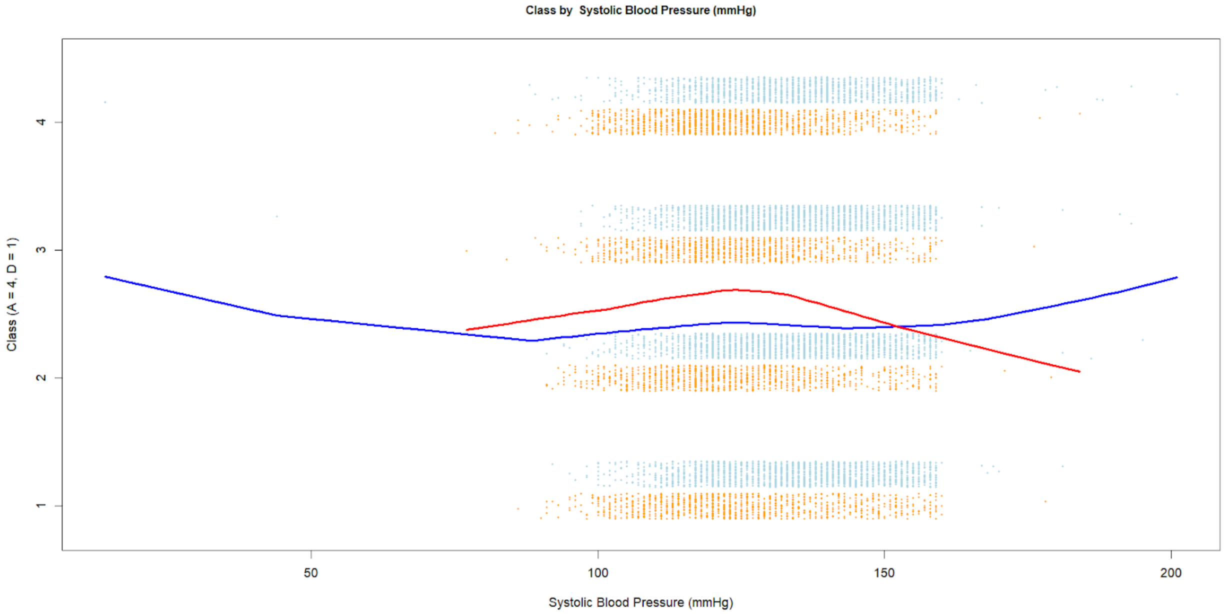


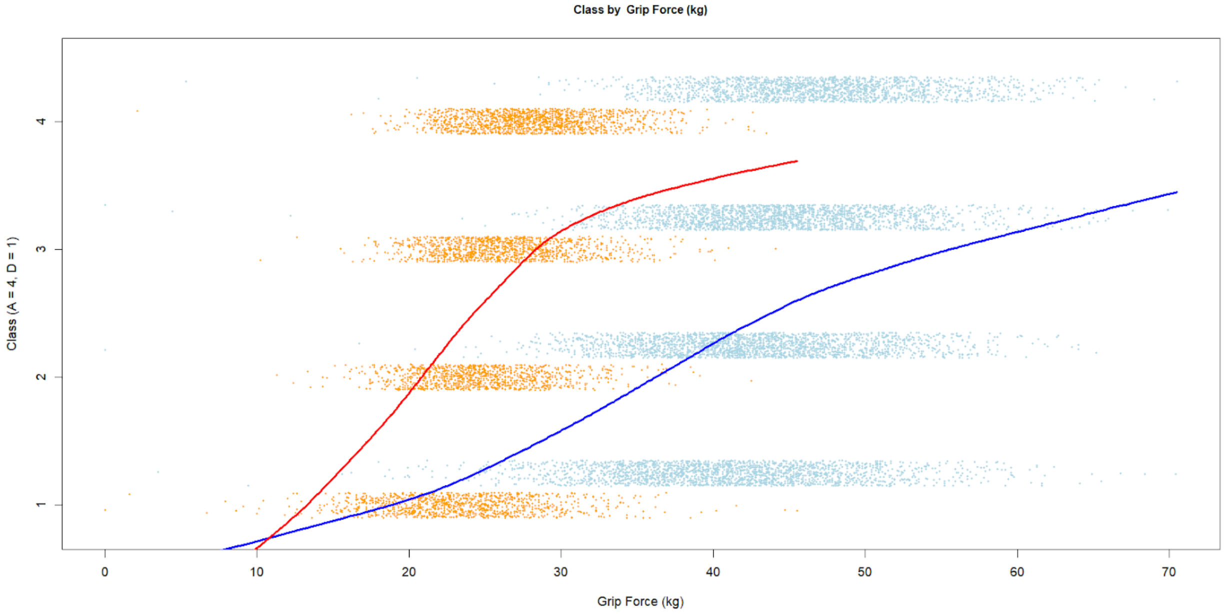


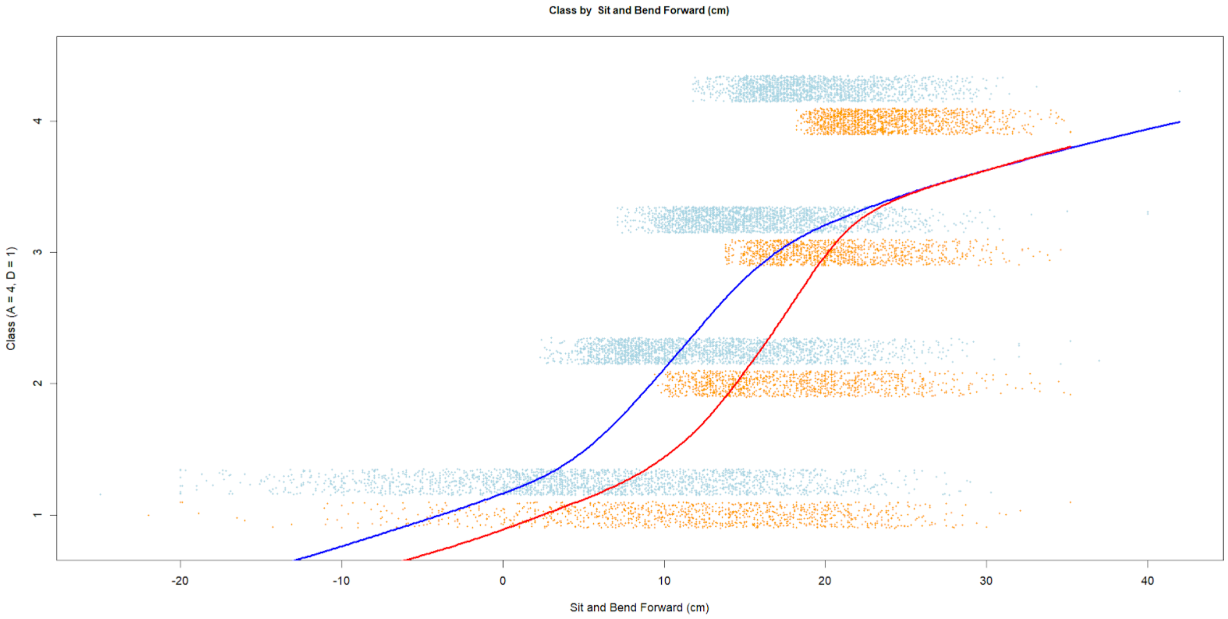

Class by Weigth (in kg)
Class (A = 4, D = 1)
1
2
3
4
40
60
80
100
120
140
Weigth (in kg)

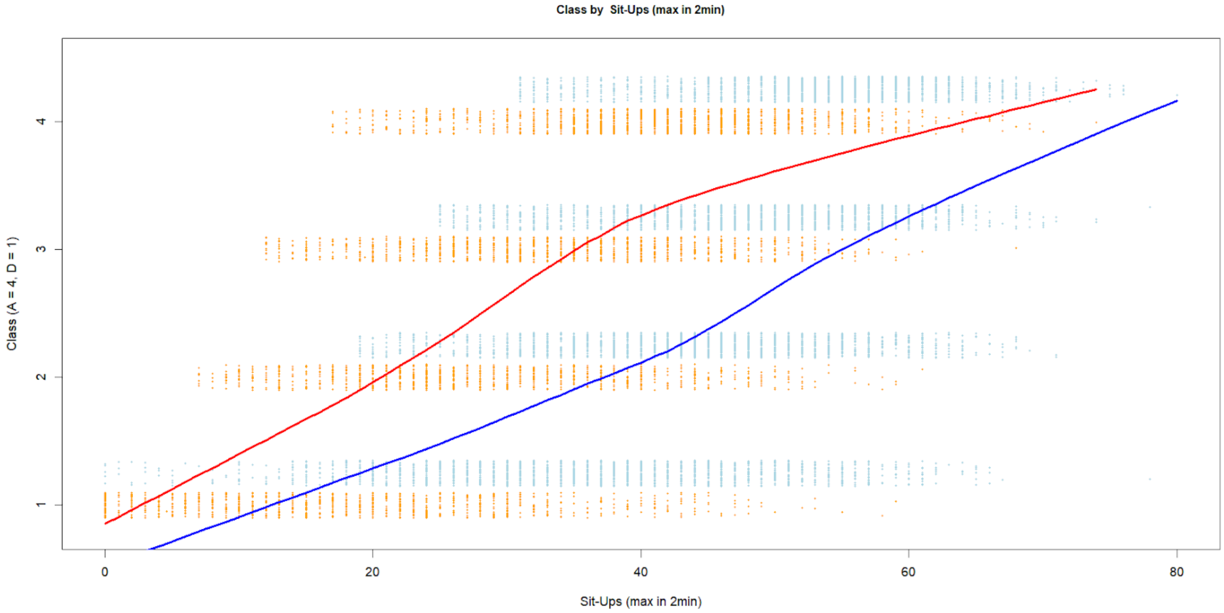

Class by Body Fat Percentage
Class (A = 4, D = 1)
1
2
3
4
10
20
30
40
50
Body Fat Percentage

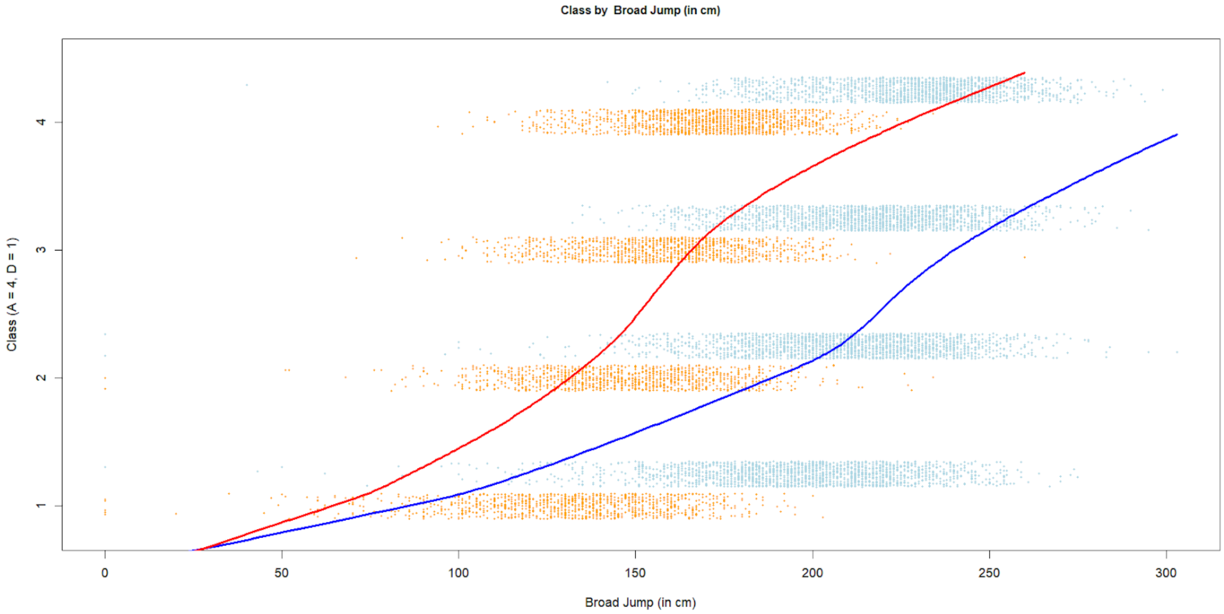

Class by Diastolic Blood Pressure (mmHg)
Class (A = 4, D = 1)
1
2
3
4
0
50
100
150
Diastolic Blood Pressure (mmHg)


Class by Systolic Blood Pressure (mmHg)
Class (A = 4, D = 1)
1
2
3
4
50
100
150
200
Systolic Blood Pressure (mmHg)


Class by Grip Force (kg)
Class (A = 4, D = 1)
1
2
3
4
0
10
20
30
40
50
60
70
Grip Force (kg)

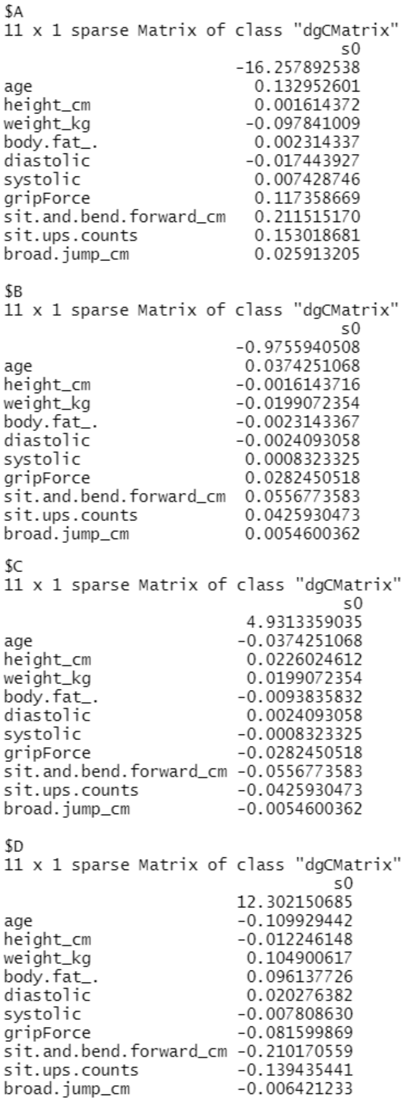

Class by Sit and Bend Forward (cm)
Class (A = 4, D = 1)
1
2
3
4
-20
-10
0
10
20
30
40
Sit and Bend Forward (cm)


Class by Sit-Ups (max in 2min)
Class (A = 4, D = 1)
1
2
3
4
0
20
40
60
80
Sit-Ups (max in 2min)


Class by Broad Jump (in cm)
Class (A = 4, D = 1)
1
2
3
4
0
50
100
150
200
250
300
Broad Jump (in cm)

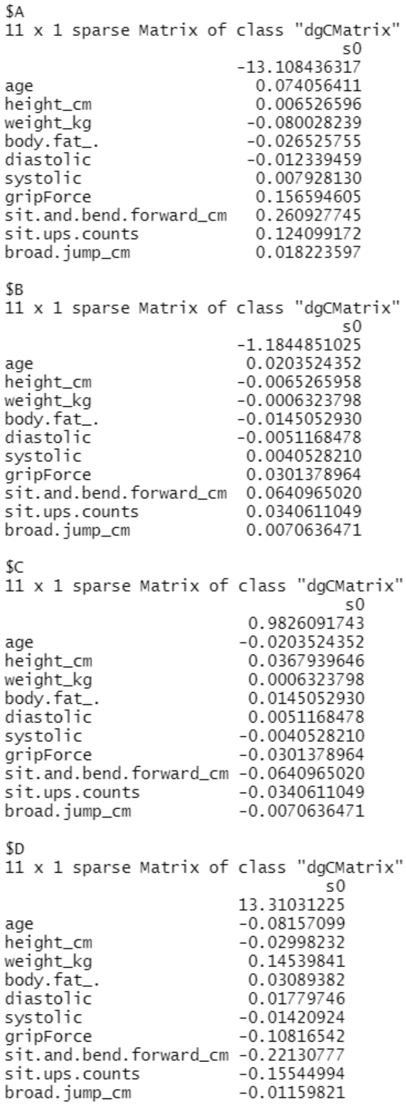
Polytomous model Coefficients

Men



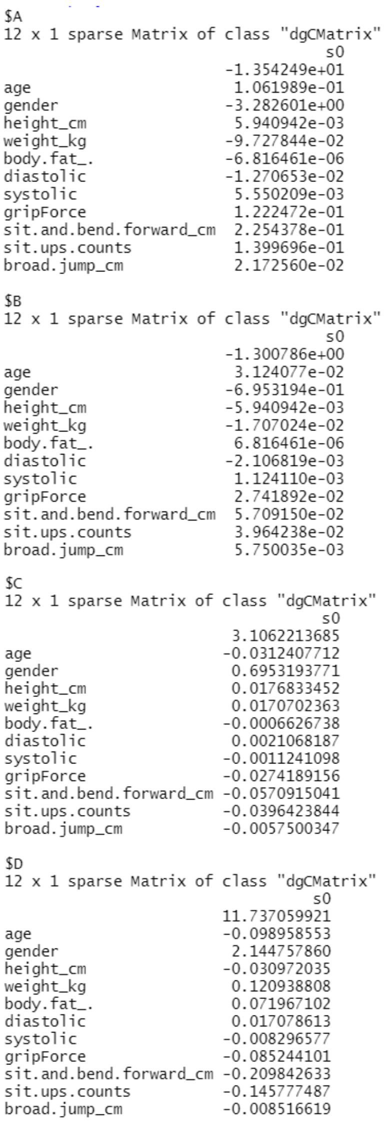
```
$A
11 x 1 sparse Matrix of class "dgCMatrix"
                                   s0
                        -16.257892538
age                       0.132952601
height_cm                 0.001614372
weight_kg                -0.097841009
body.fat_.                0.002314337
diastolic                -0.017443927
systolic                  0.007428746
gripForce                 0.117358669
sit.and.bend.forward_cm   0.211515170
sit.ups.counts            0.153018681
broad.jump_cm             0.025913205

$B
11 x 1 sparse Matrix of class "dgCMatrix"
                                   s0
                        -0.9755940508
age                      0.0374251068
height_cm               -0.0016143716
weight_kg               -0.0199072354
body.fat_.              -0.0023143367
diastolic               -0.0024093058
systolic                 0.0008323325
gripForce                0.0282450518
sit.and.bend.forward_cm  0.0556773583
sit.ups.counts           0.0425930473
broad.jump_cm            0.0054600362

$C
11 x 1 sparse Matrix of class "dgCMatrix"
                                   s0
                         4.9313359035
age                     -0.0374251068
height_cm                0.0226024612
weight_kg                0.0199072354
body.fat_.              -0.0093835832
diastolic                0.0024093058
systolic                -0.0008323325
gripForce               -0.0282450518
sit.and.bend.forward_cm -0.0556773583
sit.ups.counts          -0.0425930473
broad.jump_cm           -0.0054600362

$D
11 x 1 sparse Matrix of class "dgCMatrix"
                                  s0
                        12.302150685
age                     -0.109929442
height_cm               -0.012246148
weight_kg                0.104900617
body.fat_.               0.096137726
diastolic                0.020276382
systolic                -0.007808630
gripForce               -0.081599869
sit.and.bend.forward_cm -0.210170559
sit.ups.counts          -0.139435441
broad.jump_cm           -0.006421233
```

Women



```
$A
11 x 1 sparse Matrix of class "dgCMatrix"
                                   s0
                        -13.108436317
age                       0.074056411
height_cm                 0.006526596
weight_kg                -0.080028239
body.fat_.               -0.026525755
diastolic                -0.012339459
systolic                  0.007928130
gripForce                 0.156594605
sit.and.bend.forward_cm   0.260927745
sit.ups.counts            0.124099172
broad.jump_cm             0.018223597

$B
11 x 1 sparse Matrix of class "dgCMatrix"
                                   s0
                        -1.1844851025
age                      0.0203524352
height_cm               -0.0065265958
weight_kg               -0.0006323798
body.fat_.              -0.0145052930
diastolic               -0.0051168478
systolic                 0.0040528210
gripForce                0.0301378964
sit.and.bend.forward_cm  0.0640965020
sit.ups.counts           0.0340611049
broad.jump_cm            0.0070636471

$C
11 x 1 sparse Matrix of class "dgCMatrix"
                                   s0
                         0.9826091743
age                     -0.0203524352
height_cm                0.0367939646
weight_kg                0.0006323798
body.fat_.               0.0145052930
diastolic                0.0051168478
systolic                -0.0040528210
gripForce               -0.0301378964
sit.and.bend.forward_cm -0.0640965020
sit.ups.counts          -0.0340611049
broad.jump_cm           -0.0070636471

$D
11 x 1 sparse Matrix of class "dgCMatrix"
                                 s0
                        13.31031225
age                     -0.08157099
height_cm               -0.02998232
weight_kg                0.14539841
body.fat_.               0.03089382
diastolic                0.01779746
systolic                -0.01420924
gripForce               -0.10816542
sit.and.bend.forward_cm -0.22130777
sit.ups.counts          -0.15544994
broad.jump_cm           -0.01159821
```

Combined



```
$A
12 x 1 sparse Matrix of class "dgCMatrix"
                                   s0
                        -1.354249e+01
age                      1.061989e-01
gender                  -3.282601e+00
height_cm                5.940942e-03
weight_kg               -9.727844e-02
body.fat_.              -6.816461e-06
diastolic               -1.270653e-02
systolic                 5.550209e-03
gripForce                1.222472e-01
sit.and.bend.forward_cm  2.254378e-01
sit.ups.counts           1.399696e-01
broad.jump_cm            2.172560e-02

$B
12 x 1 sparse Matrix of class "dgCMatrix"
                                   s0
                        -1.300786e+00
age                      3.124077e-02
gender                  -6.953194e-01
height_cm               -5.940942e-03
weight_kg               -1.707024e-02
body.fat_.               6.816461e-06
diastolic               -2.106819e-03
systolic                 1.124110e-03
gripForce                2.741892e-02
sit.and.bend.forward_cm  5.709150e-02
sit.ups.counts           3.964238e-02
broad.jump_cm            5.750035e-03

$C
12 x 1 sparse Matrix of class "dgCMatrix"
                                   s0
                         3.1062213685
age                     -0.0312407712
gender                   0.6953193771
height_cm                0.0176833452
weight_kg                0.0170702363
body.fat_.              -0.0006626738
diastolic                0.0021068187
systolic                -0.0011241098
gripForce               -0.0274189156
sit.and.bend.forward_cm -0.0570915041
sit.ups.counts          -0.0396423844
broad.jump_cm           -0.0057500347

$D
12 x 1 sparse Matrix of class "dgCMatrix"
                                  s0
                        11.737059921
age                     -0.098958553
gender                   2.144757860
height_cm               -0.030972035
weight_kg                0.120938808
body.fat_.               0.071967102
diastolic                0.017078613
systolic                -0.008296577
gripForce               -0.085244101
sit.and.bend.forward_cm -0.209842633
sit.ups.counts          -0.145777487
broad.jump_cm           -0.008516619
```