DSC 424 FINAL REPORT

Group 4: Diabetes Dataset

Christopher Lee, Edward Xu, James Robinson, Hai Ha Vu

Table of Contents

Non-Technical Summary

Diabetes is an extremely prevalent health issue within the United States that can introduce both a reduced quality of life and financial strain among patients. The Diabetes Health Indicators Dataset is a public domain collection of 253,680 responses to a CDC survey on health-related behaviors and conditions. There are 22 variables within the dataset, one of which indicating whether the person had no diabetes (or diabetes only during pregnancy), prediabetes, or diabetes. The 21 other variables can be used to analyze any trends in the data to identify latent health factors or predict presence or absence of diabetes.

We investigated the use of three different techniques to analyze diabetes with the corresponding health indicators in the dataset. The first two were methods to explore the interrelationships between the health indicators or discover hidden factors within the data. These techniques are called Correspondence Analysis (CA) and Principal Component or Principal Factor Analysis (PCA/PFA). CA can reveal an association between predictor variables and the classes of the response variable. In our case, we attempted to find health indicators most heavily associated with diabetes and prediabetes as opposed to indicators more closely related to people without diabetes. An initial analysis found that conditions such as stroke, heart disease, and difficulty walking are closely related to diabetes in the dataset whereas higher physical activity and having healthcare coverage are associated with no diabetes. Generally, negative health indicators tended to be more associated with the presence of diabetes while positive health indicators tended to be more associated with no diabetes. One surprising exception to this rule was that heavy alcohol consumption was actually most heavily associated with no diabetes rather than presence of diabetes. These results are summarized in **Figure 1** where more heavily associated indicators have smaller radial distance (i.e., smaller angles) to the diabetes class.

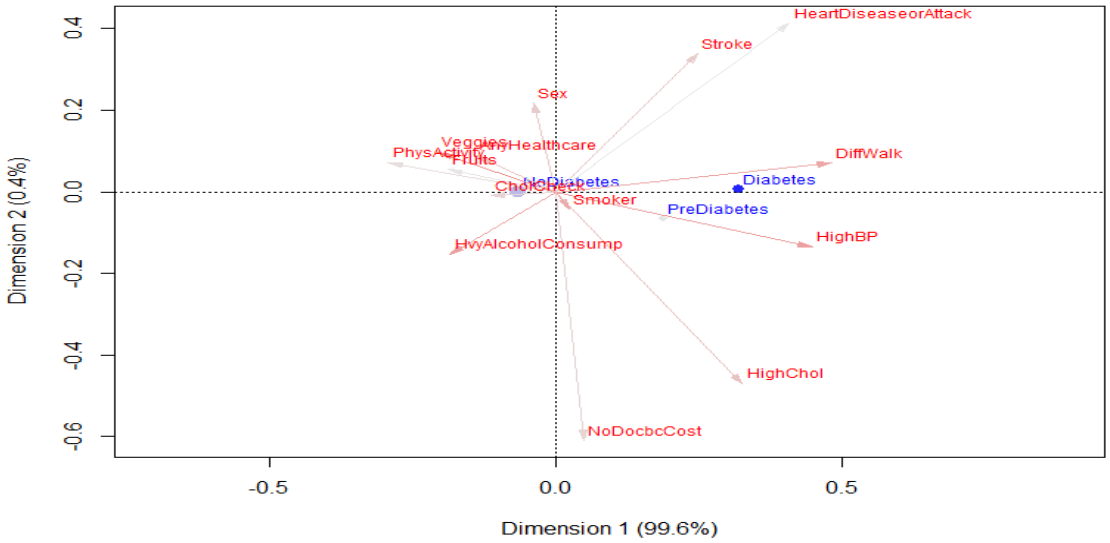


Figure 1: Correspondence analysis radial plot

PFA is a technique that is very closely related to CA but can be used to discover new or latent factors within the dataset as combinations of the existing features. We found two factors from the analysis that seemed to have interpretable meanings. One was comprised of negative health indicators such as high blood pressure or history of stroke, heart disease, or heart attack which may represent prevalent health risks, especially those that are cardiovascular in nature. The other was comprised of opposing negative

and positive health indicators which may indicate a factor representing general health with some features contributing to poor health and others contributing to good health.

The third technique that was attempted had a different purpose, that is, for classification and prediction of presence of diabetes. This method is called Linear Discriminant Analysis (LDA) which allows users to train a model for classifying instances and then use it for extrapolated prediction on unknown cases. LDA is another technique that combines variables; however, the combination is used as a classifier to separate datapoints of different classes. Unfortunately, our LDA classifier was not able to perform well at this diabetes classification task and failed to predict even a single prediabetes case. Our hypothesis is that a heavy class imbalance within the diabetes variable partially led to this poor performance. We can visualize this classification task with **Figure 2** which shows the datapoints plotted along axes representing the combined variables that were calculated via LDA. We can see that there is potentially a general difference between diabetes (black points) and no diabetes (green points) but a very large degree of mixing which makes the separation of the two classes quite difficult.

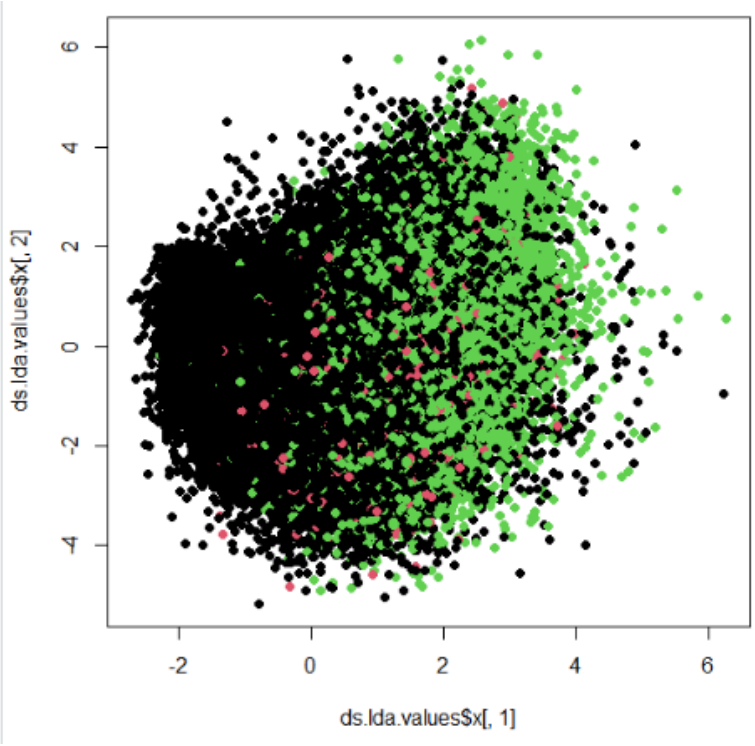


Figure 2: Plot of datapoints along the primary and secondary linear discriminants.

Technical Summary

The diabetes dataset that our group will be analyzing consists of 22 variables, of which 19 (including the response variable) are categorical/ordinal and 3 are continuous. 14 of the categorical variables are binary. Given the nature of the data, we propose two lines exploratory analysis with Principal Component Analysis (PCA) and Correspondence Analysis (CA). Furthermore, we would like to investigate diabetes classification using Linear Discriminant Analysis (LDA). The response variable has 3 categories: 0 for no diabetes or diabetes only during pregnancy, 1 for pre-diabetes, and 2 for diabetes.

Principal Factor Analysis

The 22 variables in the diabetes health indicators dataset are all coded as numeric which allows us to use not only correspondence analysis with the categorical variables, but also PCA in order to analyze variable relationships. However, the dataset is comprised of a few different types of variables. 14 are binary, 5 are ordinal, and 3 are continuous. Because of this heterogeneity, several different correlation coefficients needed to be calculated. We used polychoric correlation for ordinal-ordinal pairs, Pearson correlation for continuous-continuous pairs, and polyserial correlation for continuous-ordinal pairs. The response variable was removed and the remaining features were inputted into R's hetcor() function to compute all correlations and compile them into a correlation matrix which can be visualized in the correlation plot in **Figure 3**. The correlation plot was ordered by the angle of eigenvectors and reveals potentially 2-3 groupings of the variables. These groupings were kept in mind for the PFA since they may constitute the factors that are calculated.

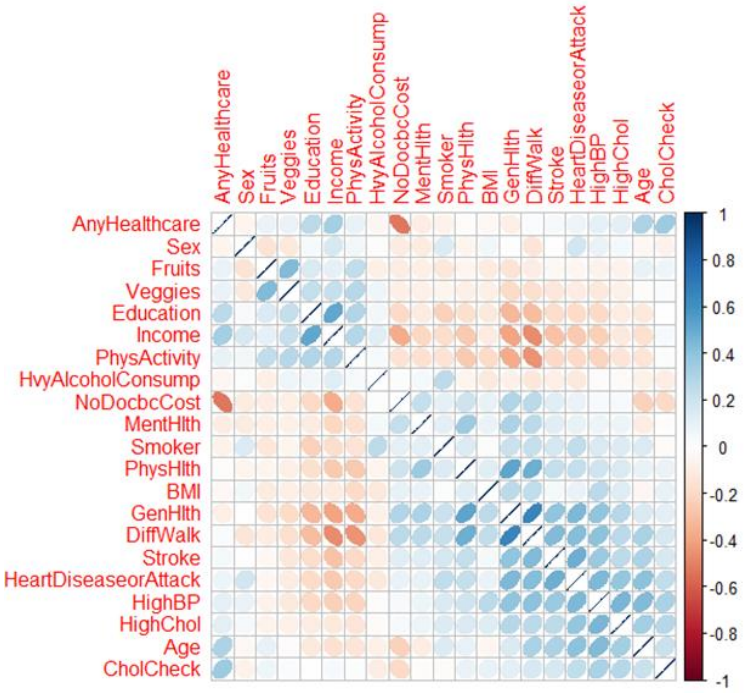


Figure 3: Correlation plot of the 21 independent variables

A correlation test was also conducted to test the significance of these calculated coefficients. Any variable that is highly correlated with a large number of other features may cause issues with the factor analysis because it may be difficult to distinguish its effect from the other features. On the other hand, any variables that are not correlated with the others at all may constitute their own factors. Therefore, both needed to be removed prior to computing the factors. In this case, CholCheck, HvyAlcoholConsump, Sex, and BMI were removed due to having 0 significant correlations with any of the other features.

The new correlation matrix was computed using the remaining 17 variables and used in PCA to determine an appropriate number of factors. The resulting variances of each component were plotted in a scree plot which showed a fairly clear elbow after the second component as seen in **Figure 4a**. Furthermore, a parallel analysis was used to compare variance captured in the diabetes data compared

to randomly generated data. This revealed that only the first two factors captured more variance than the noise as seen in **Figure 4b**.

These first two components captured 76.13% of the total variance which is sufficient for factor analysis. Thus, two factors were selected when computing principal factors using the R Psych package's principal() function. Varimax rotation was also used in order to improve the interpretability of the PFA. This was more important that preserving the orthogonal components given our goal to discover latent factors rather than to simply reduce dimensionality. This resulted in the loadings that are shown in **Figure 5a**. As a note, all values below 0.4 were filtered out due to their relatively small loading onto the factors.

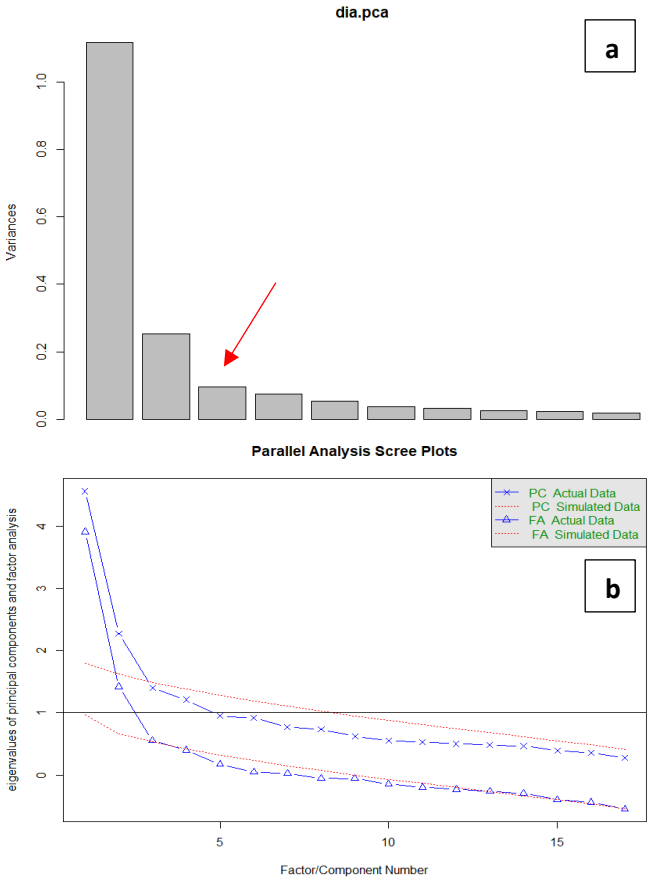


Figure 4: (a) Scree plot by plotting variance of components from prcomp(). (b) Scree plot from parallel analysis of diabetes data compared to noise

The result had a high degree of interpretability. Factor 1 had all positive contributions from variables representing poor health indicators. There seemed to be an emphasis on many cardiovascular health-related issues such as high blood pressure, high cholesterol, and stroke. Factor 2 seemed to represent a more general health measure with opposing contributions from good health indicators and poor health indicators. For example, having higher education and income along with healthcare coverage are likely correlated with better preventive or healthcare-seeking behaviors and all had positive contributions. On the other hand, indicators such as difficulty walking and poor mental or physical health had negative contributions. The removed variables can also be added back as their own factors.

Lastly, a confirmatory Common Factor Analysis (CFA) was conducted. PFA is calculated using geometric properties of the data whereas CFA is computed statistically using maximum likelihood techniques. Thus, if similar results can be found using both methods, we can have increased confidence in the factors. Using factanal() in R yielded the factor loadings shown in **Figure 5b.** Factor 1 is strikingly similar between the two methods while Factor 2 has two fewer contributing features from the CFA results. Otherwise, they have similar loading magnitudes and contributions.

As a note, the common factor analysis yielded an extremely high chi-square value with an approximate p-value of 0.0. This means that we must reject the null hypothesis that 2 factors is sufficient. However, when attempting CFA with 10 factors (the maximum allowed for this dataset with factanal), we still

receive a p-value of approximately 0.0. We believe that our extremely large dataset is what is contributing to the high chi-squre values being computed.

Correspondence Analysis

Correspondence analysis (CA) and multiple correspondence analysis (MCA) are techniques that are very similar to PCA and factor analysis. Both CA and PCA allow us to summarize patterns in data. CA is used specifically for summarizing and visualizing relative frequencies in tables. The main difference between the two is that CA is well suited for categorical data while PCA is not. MCA differs from CA in that it handles data with more than two dimensions. MCA also excels at dealing with categorical, especially survey data and it is very similar to factor analysis as well. Both are excellent for visualizing multidimensional data into a lower dimensional space.

To begin, we created a frequency table as seen in **Figure 6**. Note that there is a strong frequency imbalance in many of the variables.

We then use the ca() function in R to generate the frequency graph in **Figure 7**. By creating mental scales, we can see that Diabetes has higher frequencies to features such as HighBP and DiffWalk while PreDiabetes has a higher frequency with NoDocbcCost and HighChol. On the other hand, NoDiabetes is more closely associated with PhysAcitivity, Fruits and Veggies, and AnyHealthcare. Generally, more negative health indicators seem to be more closely associated with Diabetes and PreDiabetes while positive health indicators have higher frequencies with NoDiabetes. One glaring exception to this rule is that HvyAlcoholConsump is very highly associated with NoDiabetes. Furthermore, Diabetes and PreDiabetes are fairly close together, especially on Dimension 1 while NoDiabetes is further away and on the other side of the origin.We also note that Dimension 1 captures 99.6% of the variance and Dimension 2 captures .4%.

```
Loadings:
                        RC1    RC2
HighBP                0.713
HighChol              0.606
Stroke                0.611
HeartDiseaseorAttack  0.719
DiffWalk              0.633 -0.486
Age                   0.721
AnyHealthcare                0.630
NoDocbcCost                 -0.693
GenHlth               0.549 -0.555
Education                    0.560
Income                       0.661
Smoker
PhysActivity                 0.469
Fruits
Veggies
MentHlth                    -0.442
PhysHlth                    -0.430
```

a

```
Loadings:
                      Factor1 Factor2
HighBP                  0.629
HighChol                0.502
Stroke                  0.538
HeartDiseaseorAttack    0.638
DiffWalk                0.621   0.497
Age                     0.648
AnyHealthcare                  -0.573
NoDocbcCost                     0.664
GenHlth                 0.531   0.559
Income                         -0.606
Smoker
PhysActivity
Fruits
Veggies
Education                      -0.468
MentHlth
PhysHlth                        0.405
```

b

Figure 5: (a) Factor loadings computed using prcomp(). (b) Factor loading computed using factanal(). Both had 2 factors and Varimax rotation.

	HighBP	HighChol	CholCheck	Smoker	Stroke	HeartDiseaseorAttack	PhysActivity	Fruits	Veggies	HvyAlcoholConsump
NoDiabetes	79312	81030	204536	91824	6759	15351	166491	137416	175544	13216
PreDiabetes	2913	2875	4569	2282	265	664	3142	2789	3561	208
Diabetes	26604	23686	35105	18317	3268	7878	22287	20693	26736	832

	AnyHealthcare	NoDocbcCost	DiffWalk	Sex
NoDiabetes	202962	17013	28269	92744
PreDiabetes	4377	599	1285	2027
Diabetes	33924	3742	13121	16935

Figure 6: Frequency table of diabetes classes by binary indicators in the dataset

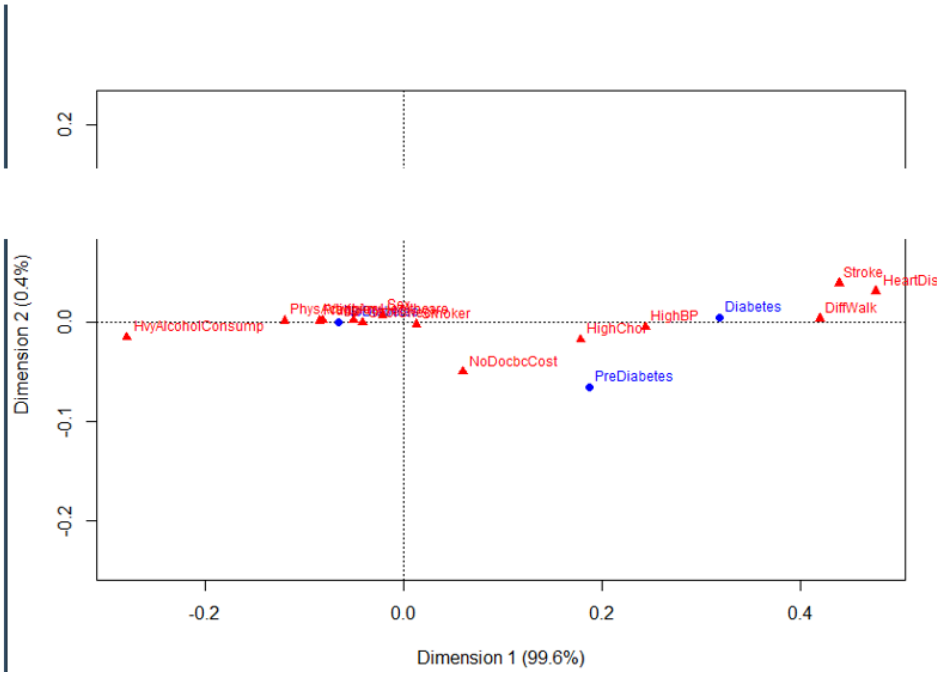


Figure 7: Correspondence Analysis plot using frequency table from Figure 6

We then moved on to attempt MCA. The preprocessing/transformation involved converting all of the 14 binary categorical columns into descriptive factor columns. The resulting frequency table in **Figure 8** is slightly different than the original frequency table.

```
     HighBP              HighChol            CholCheck          Smoker           Stroke
HighBP_N:144851   HighChol_N:146089   CholCheck_N:  9470   Smoker_N:141257   Stroke_N:243388
HighBP_Y:108829   HighChol_Y:107591   CholCheck_Y:244210   Smoker_Y:112423   Stroke_Y: 10292
          HeartDiseaseorAttack         PhysActivity            Fruits            Veggies                HvyAlcoholConsump
HeartDiseaseorAttack_N:229787    PhysActivity_N: 61760   Fruits_N: 92782   Veggies_N: 47839   HvyAlcoholConsump_N:239424
HeartDiseaseorAttack_Y: 23893    PhysActivity_Y:191920   Fruits_Y:160898   Veggies_Y:205841   HvyAlcoholConsump_Y: 14256
       AnyHealthcare          NoDocbcCost            DiffWalk          Sex
AnyHealthcare_N: 12417   NoDocbcCost_N:232326   DiffWalk_N:211005   Sex_N:141974
AnyHealthcare_Y:241263   NoDocbcCost_Y: 21354   DiffWalk_Y: 42675   Sex_Y:111706
```

Figure 8: Frequency table for multiple correspondence analysis

We then created a summary table in order to see the components generated and the variance captured as seen in **Figure 9**.

```
 Eigenvalues:
            1       2       3       4       5       6
Value       0.007347 0.000877 0.000271 0.000103 1.2e-05 2e-06
Percentage 69.95%   8.35%   2.58%   0.98%   0.11%   0.02%


 Columns:
        Diabetes_012:Diabetes Diabetes_012:NoDiabetes Diabetes_012:PreDiabetes HighBP:HighBP_N HighBP:HighBP_Y
Mass                 0.009289              0.056161              0.001217       0.038067       0.028600
ChiDist              0.712237              0.124518              1.903086       0.256970       0.342026
Inertia              0.004712              0.000871              0.004408       0.002514       0.003346
Dim. 1               3.484415             -0.620982              2.061231      -1.381485       1.838751
Dim. 2               1.230232             -0.205947              0.113951      -0.610263       0.812258
       HighChol:HighChol_N HighChol:HighChol_Y CholCheck:CholCheck_N CholCheck:CholCheck_Y Smoker:Smoker_N Smoker:Smoker_Y
Mass              0.038392            0.028275             0.002489             0.064178       0.037122       0.029545
ChiDist           0.245103            0.332806             1.340260             0.051973       0.240520       0.302208
Inertia           0.002306            0.003132             0.004470             0.000173       0.002148       0.002698
Dim. 1           -1.153918            1.566811            -1.973511             0.076529      -0.722730       0.908094
Dim. 2           -0.654823            0.889130            -6.889444             0.267160       0.506777      -0.636754
```

Figure 9: MCA summary table

Another frequency graph was created in **Figure 10**. This graph is slightly different than the previous CA graph in that it shows Diabetes having higher frequencies with heart disease and stroke.
Both techniques were helpful in visualizing the variables and their frequencies. The factors generated also lined up with the factors generated in the previous PFA section of our project. MCA was able to capture 78% of the variance in the first 2 components. Based on the summary, 2 or 3 components seem appropriate which aligns with the PCA analysis.

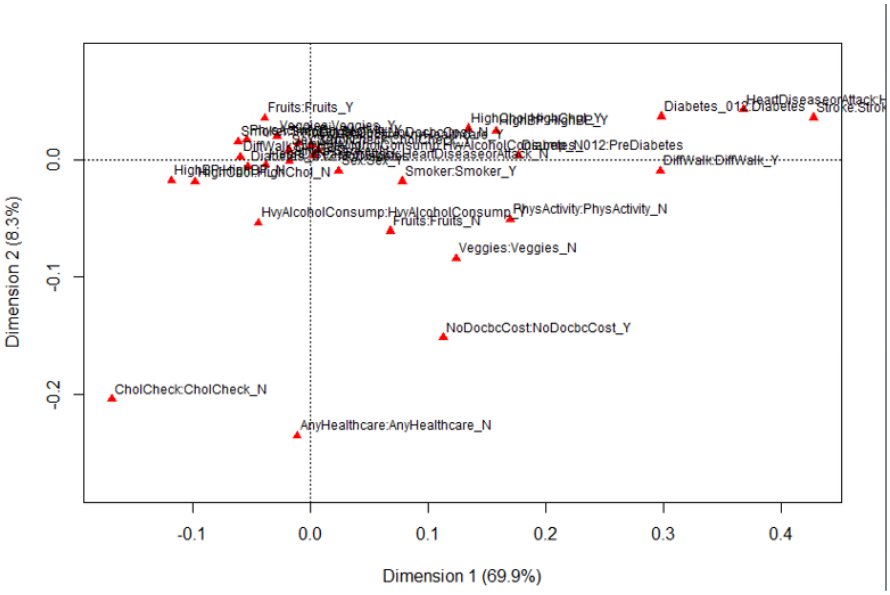


Figure 10: MCA frequency plot

Linear Discriminant Analysis

Firstly, we code all categorical variables as binary using dummy variables and then split the data set via holdout into 80% training and 20% testing. LDA is then computed on all indicators for the training set as seen in **Figure 11**. Looking at the output, we get a set of prior probabilities which is the samples that fall into each category. There is a large class imbalance with the majority of cases falling into type 0 (84%), very little of type 1 (1%), and a moderate amount of type 2 (13%). The group means are the means in each class of the independent variables. This is what LDA is attempting to separate.

```
Prior probabilities of groups:
         0          1          2
0.84265610 0.01833511 0.13900879

Group means:
    HighBP1 HighChol1 CholCheck1      BMI    Smoker1    Stroke1 HeartDiseaseorAttack1 PhysActivity1   Fruits1  Veggies1
0 0.3708979 0.3788214  0.9575585 27.74346 0.4297301 0.03161182            0.07170842     0.7782612 0.6425631 0.8210184
1 0.6248320 0.6173072  0.9865628 30.69901 0.4896533 0.05697393            0.14001612     0.6791185 0.6006450 0.7672669
2 0.7532168 0.6706604  0.9933359 31.94555 0.5180603 0.09265889            0.22430967     0.6317039 0.5853391 0.7581794
  HvyAlcoholConsump1 AnyHealthcare1 NoDocbcCost1  GenHlth2  GenHlth3   GenHlth4   GenHlth5 MentHlth PhysHlth DiffWalk1
0         0.06190794      0.9495299   0.07973125 0.3818445 0.2831263 0.09737913 0.03339532 2.942133 3.585509 0.1322773
1         0.04326794      0.9457135   0.13195378 0.2620263 0.3743617 0.22037087 0.07739855 4.553615 6.327063 0.2816447
2         0.02222537      0.9601928   0.10417922 0.1800007 0.3804544 0.27932367 0.12775159 4.413243 7.907270 0.3700684
       Sex1        Age2        Age3       Age4       Age5       Age6       Age7      Age8      Age9     Age10      Age11
0 0.4338818 0.034892288 0.049973101 0.06159802 0.06987814 0.08323393 0.10609197 0.1221201 0.1251608 0.1166760 0.08297079
1 0.4361731 0.010749798 0.017199678 0.03171191 0.03574308 0.06557377 0.09110454 0.1209352 0.1472722 0.1521096 0.13276001
2 0.4786076 0.004147318 0.008790897 0.01793627 0.03005920 0.04888164 0.08574669 0.1211939 0.1623480 0.1841126 0.14703484
       Age12      Age13 Education2 Education3 Education4 Education5 Education6    Income2    Income3    Income4    Income5
0 0.05666269 0.06429373 0.01262485 0.03229598  0.2357671  0.2726651  0.4460623 0.03914930 0.05628845 0.07359717 0.09761303
1 0.09916689 0.09056705 0.03520559 0.06799248  0.2880946  0.2897071  0.3187315 0.07632357 0.08922333 0.09836066 0.12523515
2 0.09659353 0.09113466 0.03392294 0.06454929  0.3115806  0.2939633  0.2946014 0.08769629 0.10102442 0.11357272 0.12686541
    Income6   Income7   Income8
0 0.1425689 0.1731867 0.3841485
1 0.1639344 0.1620532 0.2163397
2 0.1506150 0.1503314 0.2026514
```

Figure 11: LDA summary from initial attempt

Separation of three classes can be accomplished using 2 components. The first discriminant captures 99.14% of the trace while the second captures 0.86%. We then take the LDA scores and plot them in a histogram across the three diabetes classes. The result is displayed in **Figure 12** where we can see that there is very little separation between the three diabetes groups across both linear discriminants.

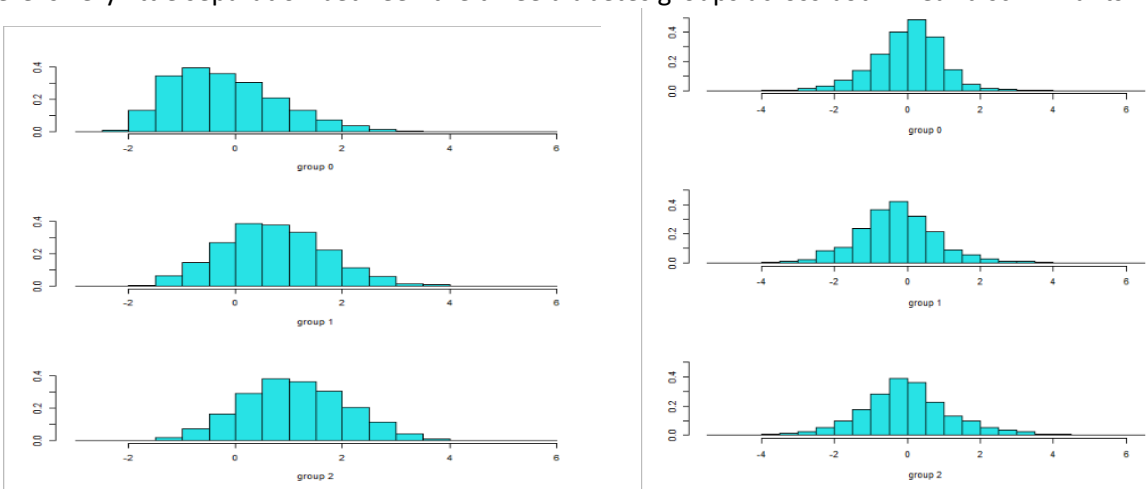


Figure 12: Histogram of scores across linear discriminant 1 (left) and 2 (right)

Another way to visualize this is to plot the scores along the two linear discriminants which can be seen in **Figure 13**. While there are perhaps general groupings of points between Diabetes (black) and No Diabetes (green), there is a large degree of mixing. Furthermore, pre-diabetes data is quite homogenously mixed among the other two classes.

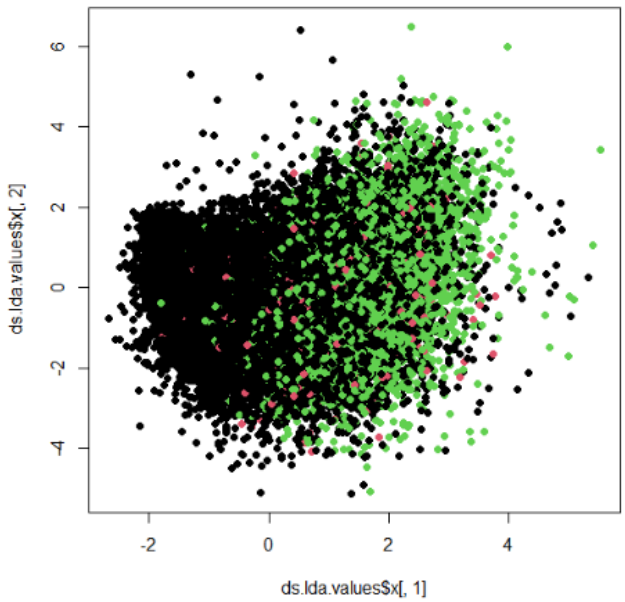


Figure 13: Scores from LD1 and LD2 plotted against each other

When we calculate the confusion matrix on the test set (see **Figure 14**), we see relatively poor performance in classifying all three classes.

	Predicted Class		
Actual Class	0	1	2
0	41022	0	1669
1	788	0	122
2	5389	0	1746

Figure 14: Confusion matrix on LDA test set

Class 0 appears to have the strongest performance of the three while class 2 has poor performance and the model failing to make a single class 1 prediction altogether. We suspect that the heavy class imbalance is contributing to this behavior since the vast majority of cases are class 0 while class 1 is the heavy minority.

Because pre-diabetes is not well separated from either class, future work could involve removing that class entirely from the dataset and focusing on the two class problem instead. Furthermore, we can investigate over- or undersampling for class balance along with data augmentation techniques such as SMOTE for creating synthetic data.

Appendices

Appendix A: Individual Reports

Edward Xu

My main contributions to the group project have been twofold. On the administrative side, I feel that I acted as a sort of project manager by organizing meetings and taking meeting minutes as well as delegating roles to the other members. On the technical side, I analyzed the correlations between the variables and used that to run a principal factor analysis (PFA) on the independent variables to identify latent factors in the data.

The data analysis began with computing the correlation matrix. Because the variables had different types, the hetcor() function was needed to handle the heterogeneity. As a note, I treated binary variables as ordinal rather than categorical in order to compute the correlation coefficients. After computing the correlation matrix, the variable relationships were visualized using corrplot() and ordering by the angle of the eigenvectors. This revealed potentially 2 or 3 groupings of variables which I though could imply 2-3 factors from the factor analysis. A correlation test was also conducted in order to isolate any variables that were too highly correlated or not correlated at all with the other variables. This identified four variables, CholCheck, HvyAlcoholConsump, Sex, and BMI that had 0 significant correlations. These were removed from the set prior to the factor analysis to constitute their own factors rather than contributing to other components. Principal component analysis (PCA) with prcomp() was then used in addition to parallel analysis to determine an appropriate number of factors. The scree plot from the PCA showed a fairly clear elbow after 2 components, and the parallel analysis showed that only the first two components captured more variance than noise. Furthermore, the first two components were able to capture 76.13% of the variance in the dataset. These confirmatory techniques allowed us to choose 2 factors to use with the principal() function for PFA. Also, to improve interpretability, Varimax rotation was implemented. This yielded two factors that were very interpretable. The first represented negative health conditions, especially related to cardiovascular health. The second appeared to be a more general health factor with negative contributions from poor health indicators and positive contributions from healthcare-seeking or preventive behaviors. Lastly, a common factor analysis (CFA) was conducted to confirm the factors that were calculated via PFA. This yielded strikingly similar factors with the first factor being almost identical between the two and the second factor simply having two fewer variables when calculated via CFA. As a note, the chi-square value from the CFA was extremely high with an p-value of approximately 0. However, we also checked with 10 factors (which was the maximum allowed with our number of variables) which also yielded a p-value of approximately 0. Our group hypothesizes that the large number of datapoints (>250,000) potentially caused the high chi-square value regardless of how many factors are used, so we still feel relatively confident in our choice of 2 factors.

Given that this class is online and asynchronous, it was difficult to connect and coordinate with everyone. My main takeaway from the administrative work was that it usually only takes one proactive voice to get the ball rolling, especially since everyone in the group is competent and willing to complete graduate level work. From the data analysis side of the project, I learned that data preparation and exploration can very often take significantly more time than the actual application of analysis techniques. In my case, I needed to learn the variables and their types. This allowed me to explore whether correlation analysis could be feasible and see what tools are available to conduct such analysis. This probably spanned 50-60% of my time working with the data. Afterwards, I could simply follow familiar steps from lectures or assignments to compute the principal components and factors.

James Robinson

Introduction

The analysis techniques I have chosen to focus on for the final project are correspondence analysis (CA) and multiple correspondence analysis (MCA). CA is a technique that is used for summarizing and visualizing relative frequencies in tables. MCA excels at dealing with categorical data (i.e. survey data as we are dealing with here) and data with more than two dimensions. It is very similar to factor analysis.

Summary of Work

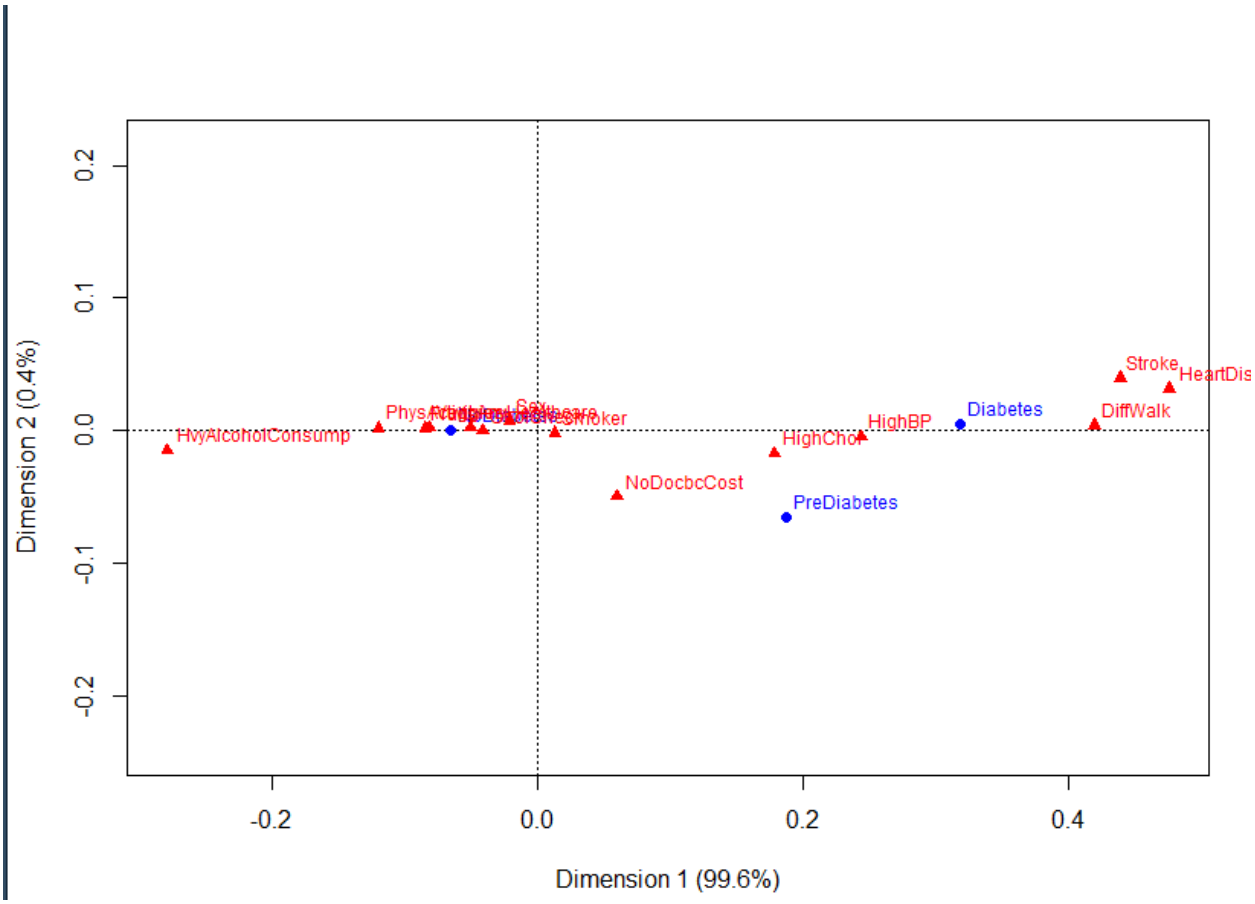
Beginning with CA, I preprocessed/transformed the data by separating 14 categorical predictors into a separate dataframe. Those predictors were:
HighBP,HighChol,CholCheck,Smoker,Stroke,HeartDiseaseorAttack,PhysActivity,Fruits,Veggies,HvyAlcoholConsump,AnyHealthcare,NoDocbcCost,DiffWalk,Sex
I then created a frequency table based on those 14 predictors and named the rows "NoDiabetes", "PreDiabetes" and "Diabetes".

```
> diabetes8 = subset(diabetes, select = c(Diabetes_012,HighBP,HighChol,CholCheck,Smoker,Stroke,HeartDiseaseorAttack,PhysActivity,Fruit
s,Veggies,HvyAlcoholConsump,AnyHealthcare,NoDocbcCost,DiffWalk,Sex) )
> finald = diabetes8 %>%
+   group_by(Diabetes_012) %>%
+   summarise(across(everything(), sum), .groups = 'drop')
> cadf = as.data.frame(finald)
> cadf$Diabetes_012 = NULL
> rownames(cadf) = c( "NoDiabetes", "PreDiabetes","Diabetes")
> head(cadf)
            HighBP HighChol CholCheck Smoker Stroke HeartDiseaseorAttack PhysActivity Fruits Veggies HvyAlcoholConsump
NoDiabetes   79312    81030    204536  91824   6759                15351       166491 137416  175544             13216
PreDiabetes   2913     2875      4569   2282    265                  664         3142   2789    3561               208
Diabetes     26604    23686     35105  18317   3268                 7878        22287  20693   26736               832
            AnyHealthcare NoDocbcCost DiffWalk   Sex
NoDiabetes         202962       17013    28269 92744
PreDiabetes          4377         599     1285  2027
Diabetes            33924        3742    13121 16935
```

I then ran the ca() function against the frequency table.

```
> fit = ca(cadf)
> fit

 Principal inertias (eigenvalues):
            1        2
Value       0.020313 8.8e-05
Percentage  99.57%   0.43%
```

The output shows the 2 dimensions generated and the amount of explained variance for each. The first component contains 99.57% and the second only .4%
One of the main reasons for using CA is that it allows easy visualization. I ran the plot() function against the CA output and it generated the below graph.



As you can see from the above graph that Diabetes has higher frequencies of HighBP and DiffWalk. PreDiabetes has a higher frequency of NoDocbcCost and HighChol.

I then moved on to MCA. The preprocessing/transformation involved converting all of the 14 binary categorical columns into descriptive factor columns as the example below illustrates.

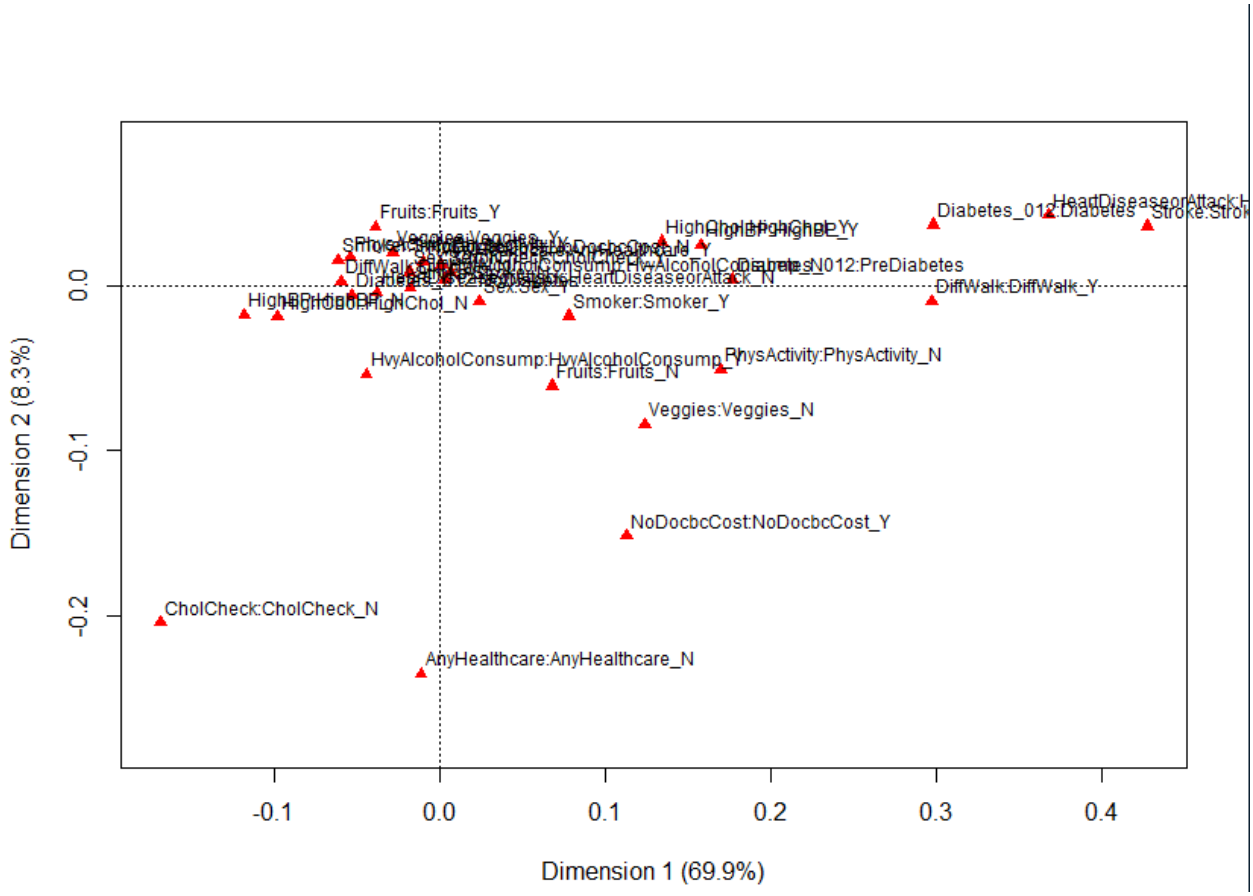
```
diabetes4$HighBP[diabetes4$HighBP == 1] <- "HighBP_Y"
diabetes4$HighBP[diabetes4$HighBP == 0] <- "HighBP_N"

diabetes4$HighBP <- as.factor(diabetes4$HighBP)
```

The frequency table generated by the summary() function shows a frequency imbalance in a few of the variables.

```
> summary(diabetes4)
      HighBP           HighChol          CholCheck          Smoker          Stroke
 HighBP_N:144851   HighChol_N:146089   CholCheck_N:  9470   Smoker_N:141257   Stroke_N:243388
 HighBP_Y:108829   HighChol_Y:107591   CholCheck_Y:244210   Smoker_Y:112423   Stroke_Y: 10292
          HeartDiseaseorAttack        PhysActivity         Fruits           Veggies                HvyAlcoholConsump
 HeartDiseaseorAttack_N:229787    PhysActivity_N: 61760   Fruits_N: 92782   Veggies_N: 47839   HvyAlcoholConsump_N:239424
 HeartDiseaseorAttack_Y: 23893    PhysActivity_Y:191920   Fruits_Y:160898   Veggies_Y:205841   HvyAlcoholConsump_Y: 14256
       AnyHealthcare          NoDocbcCost          DiffWalk          Sex
 AnyHealthcare_N: 12417   NoDocbcCost_N:232326   DiffWalk_N:211005   Sex_N:141974
 AnyHealthcare_Y:241263   NoDocbcCost_Y: 21354   DiffWalk_Y: 42675   Sex_Y:111706
```

Then I generated a summary of the components and a graph showing where the variables were associated on the first 2 components.

```
> dmca <- mjca(diabetes4)
> print(dmca)

 Eigenvalues:
           1        2        3        4        5       6
Value      0.007347 0.000877 0.000271 0.000103 1.2e-05 2e-06
Percentage 69.95%   8.35%    2.58%    0.98%    0.11%   0.02%


 Columns:
        Diabetes_012:Diabetes Diabetes_012:NoDiabetes Diabetes_012:PreDiabetes HighBP:HighBP_N HighBP:HighBP_Y
Mass                 0.009289               0.056161                0.001217       0.038067       0.028600
ChiDist              0.712237               0.124518                1.903086       0.256970       0.342026
Inertia              0.004712               0.000871                0.004408       0.002514       0.003346
Dim. 1               3.484415              -0.620982                2.061231      -1.381485       1.838751
Dim. 2               1.230232              -0.205947                0.113951      -0.610263       0.812258
        HighChol:HighChol_N HighChol:HighChol_Y CholCheck:CholCheck_N CholCheck:CholCheck_Y Smoker:Smoker_N Smoker:Smoker_Y
Mass               0.038392            0.028275              0.002489              0.064178       0.037122       0.029545
ChiDist            0.245103            0.332806              1.340260              0.051973       0.240520       0.302208
Inertia            0.002306            0.003132              0.004470              0.000173       0.002148       0.002698
Dim. 1            -1.153918            1.566811             -1.973511              0.076529      -0.722730       0.908094
Dim. 2            -0.654823            0.889130             -6.889444              0.267160       0.506777      -0.636754
```



Conclusion

Both techniques were helpful in visualizing the variables and their frequencies. The factors generated also lined up with the factors generated by Edward in the principal component analysis section of our project. MCA was able to capture 78% of the variance I the first 2 components. Based on the summary I would have picked 3 components which the PCA analysis agreed with. The main issue that I have with MCA is the messiness of the graph. The CA graph is much easier to read.

Hai Ha Vu

Diabetes Dataset – Linear Discriminant Analysis

The diabetes dataset that our group will be analyzing consists of 22 variables, of which 19 (including the response variable) are categorical/ordinal and 3 are continuous. 14 of the categorical/ordinal variables are binary. Given the nature of the data, we propose two lines of analysis with principal component analysis (PCA) and correspondence analysis (CA). Furthermore, we would like to investigate applying techniques such as clustering and linear discriminant analysis (LDA) for classification.

After getting apply PCA on dataset, I used another dimension reduction technology to analyze dataset. In our dataset, dependent variable has 3 types of category: 0 is for no diabetes or only during pregnancy, 1 is for pre-diabetes, and 2 is for diabetes.

Before running LDA, I transformed the data by transforming some variables to factors, include independent variables and dependent variable, those are:

Diabetes_012, GenHlth, Age, Education, Income: Categorical Variables
HighBP, HighChol, CholCheck, Smoker, Stroke, HeartDiseaseorAttack, PhysActivity, Fruits, Veggies, HvyAlcoholConsump, AnyHealthcare, NoDocbcCost, DiffWalk, Sex: Binary variables

Firstly, I try an initial LDA on everything and print the result:
ds.lda = lda(Diabetes_012 ~ ., data=ds)

```
Prior probabilities of groups:
         0          1          2
0.84241170 0.01825528 0.13933302

Group means:
    HighBP1 HighChol1 CholCheck1      BMI   Smoker1    Stroke1 HeartDiseaseorAttack1 PhysActivity1   Fruits1
0 0.3711319 0.3791711  0.9571040 27.74252 0.4296804 0.03162801            0.07183334     0.7790766 0.6430233
1 0.6290218 0.6208162  0.9866120 30.72447 0.4927661 0.05722306            0.14338156     0.6784712 0.6022457
2 0.7526736 0.6701183  0.9931817 31.94401 0.5182199 0.09245742            0.22288236     0.6305381 0.5854411
   Veggies1 HvyAlcoholConsump1 AnyHealthcare1 NoDocbcCost1  GenHlth2  GenHlth3   GenHlth4   GenHlth5 MentHlth
0 0.8214391         0.06184284      0.9497387   0.07961049 0.3813189 0.2829207 0.09712077 0.03346701 2.944404
1 0.7689484         0.04491471      0.9451522   0.12934571 0.2621464 0.3731376 0.22133448 0.07579357 4.529907
2 0.7564081         0.02353873      0.9597691   0.10586771 0.1805296 0.3807220 0.27697618 0.12951961 4.461806
  PhysHlth DiffWalk1      Sex1        Age2        Age3       Age4       Age5       Age6       Age7       Age8
0 3.582416 0.1322817 0.4339855 0.034646215 0.050242626 0.06108946 0.06992415 0.08312939 0.10672756 0.1217531
1 6.348305 0.2774779 0.4377024 0.011660548 0.015547398 0.03066292 0.03519758 0.06737206 0.09026128 0.1187648
2 7.954479 0.3712160 0.4791207 0.003960844 0.008883608 0.01771063 0.02973462 0.04928422 0.08736491 0.1206077
       Age9     Age10      Age11      Age12      Age13 Education2 Education3 Education4 Education5 Education6
0 0.1254498 0.1166993 0.08324637 0.05677038 0.06411234 0.01262968 0.03213806  0.2355325  0.2724482  0.4466666
1 0.1515871 0.1505074 0.12999352 0.09609156 0.09781905 0.03476571 0.06780393  0.2915137  0.2878428  0.3176420
2 0.1621966 0.1855373 0.14544786 0.09627681 0.09078821 0.03346913 0.06495785  0.3130764  0.2929327  0.2942341
     Income2    Income3    Income4    Income5   Income6   Income7   Income8
0 0.03903080 0.05617609 0.07310145 0.09729391 0.1423986 0.1741623 0.3845477
1 0.07687325 0.09090909 0.09911466 0.12675448 0.1615202 0.1587130 0.2183114
2 0.08730832 0.10094494 0.11469473 0.12742602 0.1496916 0.1489560 0.2035591
```

Looking at the output, we get a set of prior probabilities which is just the samples which fall into each category, we can see that they're not even, we have huge more of type 0 (84%), and very little of type 1 (1%) and a middle of type 2 (13%). The group means are the means in each group of the independent variables. This is what are LDA trying to separate those means.

Then we get the actual computation the coefficients of linear discriminant by print the scaling:
print(ds.lda$scaling[order(ds.lda$scaling[, 1]),])

print(ds.lda$scaling[order(ds.lda$scaling[, 2]),])

```
Coefficients of linear discriminants:
                              LD1          LD2
HighBP1                0.578099505  0.258223512
HighChol1              0.443742845 -0.618066441
CholCheck1             0.346913166 -0.551612139
BMI                    0.054097201 -0.015464464
Smoker1               -0.051786106  0.038685785
Stroke1                0.252281938  1.067062893
HeartDiseaseorAttack1  0.473386102  1.163890939
PhysActivity1         -0.056318111 -0.089475297
Fruits1               -0.008622061 -0.013787876
Veggies1              -0.026757814  0.115227136
HvyAlcoholConsump1    -0.383404981 -0.545612421
AnyHealthcare1         0.103319803  0.342415178
NoDocbcCost1          -0.023128711 -1.244827678
GenHlth2               0.067933985 -0.343677882
GenHlth3               0.534521818 -0.245603523
GenHlth4               1.164540167  0.174366588
GenHlth5               1.455534783  1.860269502
MentHlth              -0.004252380 -0.040550719
PhysHlth              -0.003773351 -0.008737074
DiffWalk1              0.294991665  0.531923632
Sex1                   0.119061568  0.154998863
Age2                  -0.030990143 -0.558782242
Age3                  -0.041719672 -0.372197368
Age4                   0.031097265 -0.834044003
Age5                   0.070988533 -0.563398522
Age6                   0.146412153 -1.214258033
Age7                   0.239940651 -0.854585383
Age8                   0.287731590 -0.951925092
Age9                   0.457422576 -1.105961234
Age10                  0.602703319 -0.852331599
Age11                  0.655305560 -1.382879266
Age12                  0.560762141 -1.914289594
Age13                  0.313970414 -2.167842208
Education2             0.150929704 -4.489210384
Education3            -0.058965084 -4.090102734
Education4            -0.174724353 -3.020523529
Education5            -0.153054916 -3.050451491
Education6            -0.187651223 -2.961264383
Income2                0.030160718  0.538271538
Income3               -0.038638408  0.741787198
Income4               -0.074995938  1.058319220
Income5               -0.141270490  0.687369654
Income6               -0.210616688  0.596262135
Income7               -0.230310438  0.820251058
Income8               -0.291122043  1.137945907
```

```
Proportion of trace:
   LD1    LD2
0.9914 0.0086
```

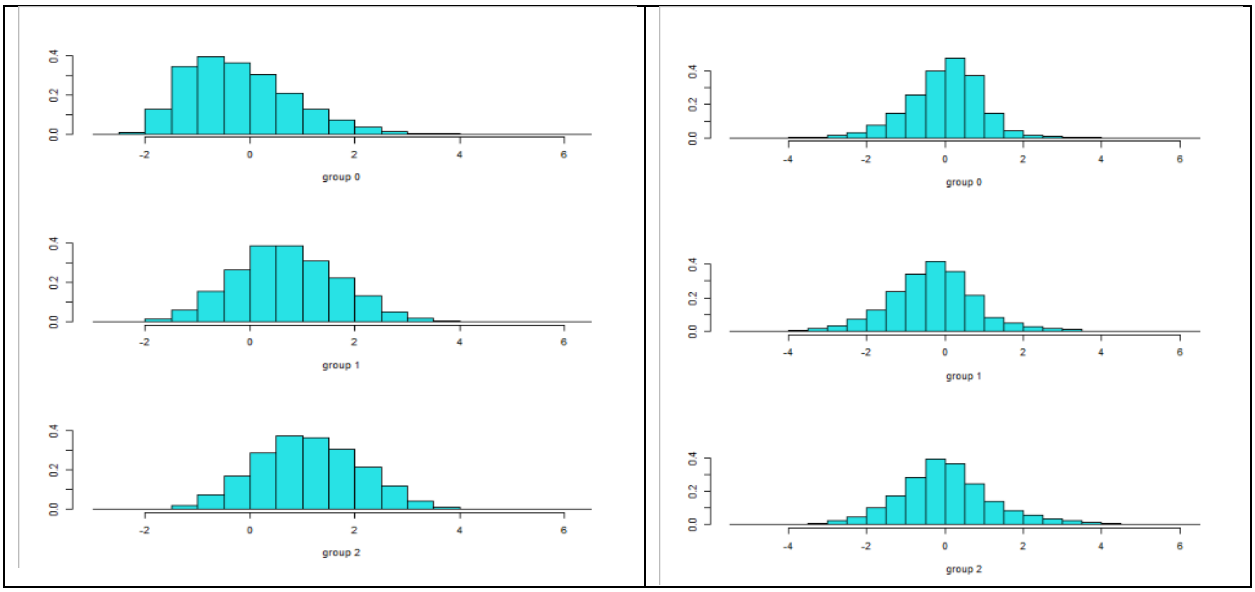
Since I separate 3 classes, 0 is for no diabetes or only during pregnancy, 1 is for pre-diabetes, and 2 is for diabetes, it accomplished by 2 components. And we get a proportion of traits for loading which is the components contribute. We can see that the first one is capturing 99.14% of the separation and then the last one separate the rest. This is the summary of how much of the available separation we're getting as far as LDA is concerned.

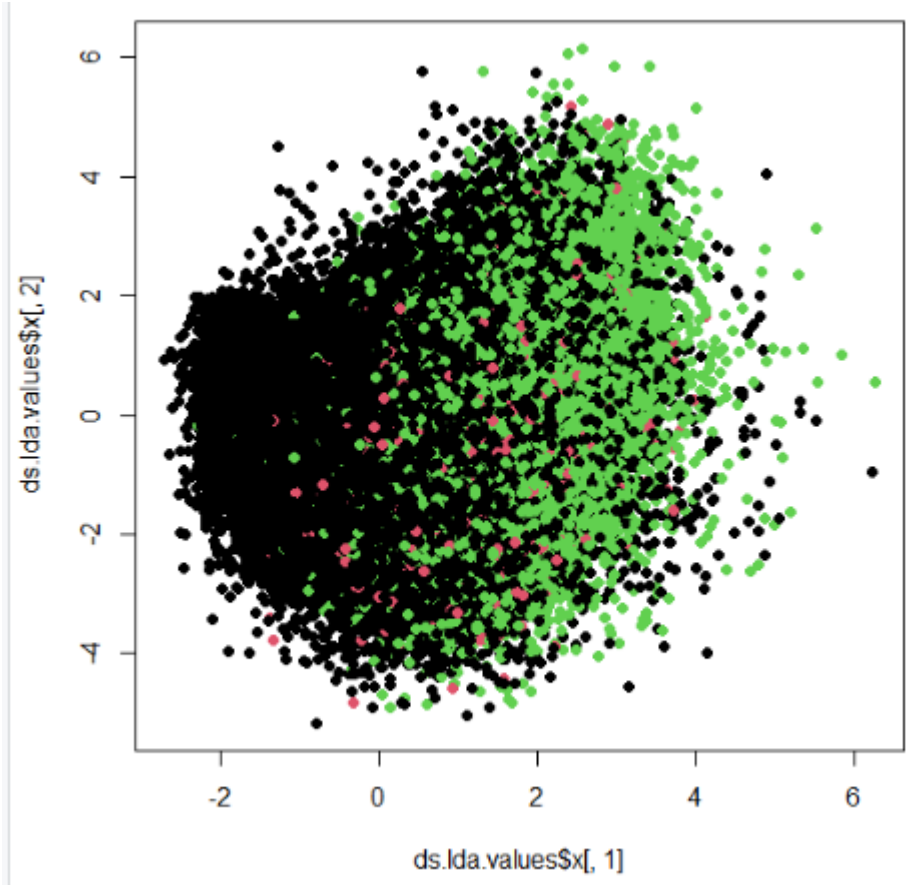
We can see the linear discriminant has the mostly income, education, mental health, physical health on the negative sides, on the positive side we got job age, sex, highBP, highChol …

After I apply it on the training dataset, and look at the histogram, there is actually a huge overlap with quite a bit more confusion between these 3 groups. Apply the transform data we can see the classification. Here is the small separation but most of the character categories are overlapping like we got the histogram.

Calculate the confusion matrix we got not perfect fit here.

```
# Look at the separation
ds.lda.values = predict(ds.lda)
ldahist(data=ds.lda.values$x[, 1], g=ds$Diabetes_012)
ldahist(data=ds.lda.values$x[, 2], g=ds$Diabetes_012)
# Plot the transformed data:
plot(ds.lda.values$x[, 1], ds.lda.values$x[, 2], col=ds$Diabetes_012, pch=16)
```






```
> # Compute a confusion matrix
> table(ds$Diabetes_012, ds.lda.values$class)

        0      1      2
  0 205452      0   8251
  1   4023      0    608
  2  26605      0   8741
```

Now we reserve a test set and a training set. We got the sample of 80% and the rest of dataset for test set. After building model for training set, look at the output: We got a similar output from the parameters, and the loading things are looking pretty stable here.

 # Now, let's separate into test and training set:

*s = sample(nrow(ds), nrow(ds) * .8)*

dsTrain = ds[s,]

dsTest = ds[-s,]

 # Build the model on the training set

ds.lda.train = lda(Diabetes_012 ~ ., data=dsTrain)

 # Look at the output

ds.lda.train

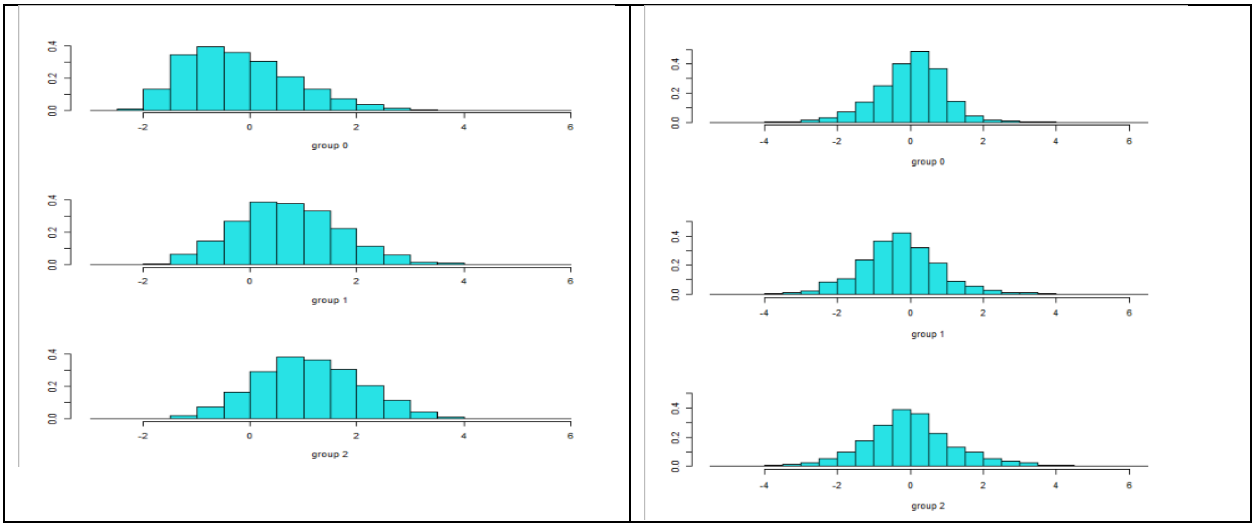
```
Call:
lda(Diabetes_012 ~ ., data = dsTrain)

Prior probabilities of groups:
         0          1          2
0.84265610 0.01833511 0.13900879

Group means:
    HighBP1 HighChol1 CholCheck1      BMI   Smoker1    Stroke1 HeartDiseaseorAttack1 PhysActivity1   Fruits1  Veggies1
0 0.3708979 0.3788214  0.9575585 27.74346 0.4297301 0.03161182            0.07170842     0.7782612 0.6425631 0.8210184
1 0.6248320 0.6173072  0.9865628 30.69901 0.4896533 0.05697393            0.14001612     0.6791185 0.6006450 0.7672669
2 0.7532168 0.6706604  0.9933359 31.94555 0.5180603 0.09265889            0.22430967     0.6317039 0.5853391 0.7581794
  HvyAlcoholConsump1 AnyHealthcare1 NoDocbcCost1  GenHlth2  GenHlth3   GenHlth4   GenHlth5 MentHlth PhysHlth DiffWalk1
0         0.06190794      0.9495299   0.07973125 0.3818445 0.2831263 0.09737913 0.03339532 2.942133 3.585509 0.1322773
1         0.04326794      0.9457135   0.13195378 0.2620263 0.3743617 0.22037087 0.07739855 4.553615 6.327063 0.2816447
2         0.02222537      0.9601928   0.10417922 0.1800007 0.3804544 0.27932367 0.12775159 4.413243 7.907270 0.3700684
       Sex1        Age2        Age3       Age4       Age5       Age6       Age7       Age8       Age9      Age10      Age11
0 0.4338818 0.034892288 0.049973101 0.06159802 0.06987814 0.08323393 0.10609197 0.1221201 0.1251608 0.1166760 0.08297079
1 0.4361731 0.010749798 0.017199678 0.03171191 0.03574308 0.06557377 0.09110454 0.1209352 0.1472722 0.1521096 0.13276001
2 0.4786076 0.004147318 0.008790897 0.01793627 0.03005920 0.04888164 0.08574669 0.1211939 0.1623480 0.1841126 0.14703484
       Age12      Age13 Education2 Education3 Education4 Education5 Education6    Income2    Income3    Income4    Income5
0 0.05666269 0.06429373 0.01262485 0.03229598  0.2357671  0.2726651  0.4460623 0.03914930 0.05628845 0.07359717 0.09761303
1 0.09916689 0.09056705 0.03520559 0.06799248  0.2880946  0.2897071  0.3187315 0.07632357 0.08922333 0.09836066 0.12523515
2 0.09659353 0.09113466 0.03392294 0.06454929  0.3115806  0.2939633  0.2946014 0.08769629 0.10102442 0.11357272 0.12686541
    Income6   Income7   Income8
0 0.1425689 0.1731867 0.3841485
1 0.1639344 0.1620532 0.2163397
2 0.1506150 0.1503314 0.2026514
```

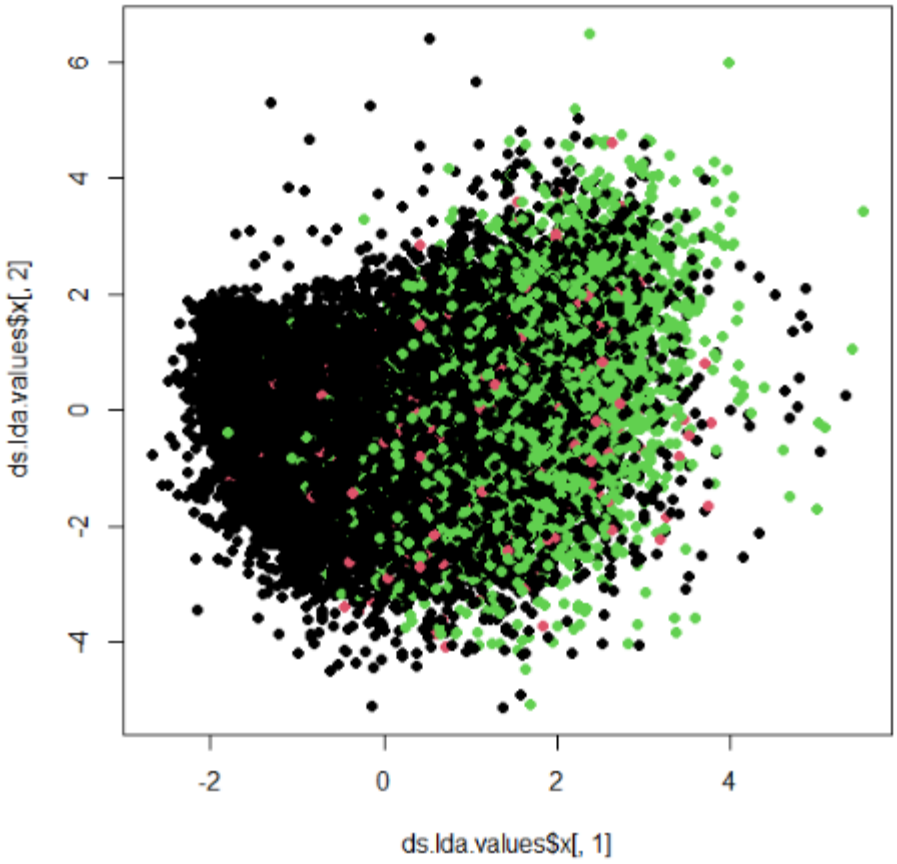
```
Coefficients of linear discriminants:
                              LD1          LD2
HighBP1                 0.579743596  0.292938055
HighChol1               0.443048584 -0.585635919
CholCheck1              0.347098896 -0.555194953
BMI                     0.054328151 -0.011581107
Smoker1                -0.049855781  0.082965911
Stroke1                 0.251680253  1.064806644
HeartDiseaseorAttack1   0.484814393  1.306055871
PhysActivity1          -0.047661962 -0.106213121
Fruits1                -0.010289889 -0.011070001
Veggies1               -0.025814317  0.167748604
HvyAlcoholConsump1     -0.400886953 -0.489688358
AnyHealthcare1          0.105229859  0.286192058
NoDocbcCost1           -0.035155431 -1.384768673
GenHlth2                0.062551922 -0.391458457
GenHlth3                0.534066500 -0.332113733
GenHlth4                1.175764647  0.133863908
GenHlth5                1.456354800  1.563461884
MentHlth               -0.004293562 -0.041853216
PhysHlth               -0.004476290 -0.004561486
DiffWalk1               0.300373613  0.333094555
Sex1                    0.112221224  0.147907961
Age2                   -0.028412280 -0.353944947
Age3                   -0.043955838 -0.399749467
Age4                    0.032683994 -0.781456097
Age5                    0.073422645 -0.497468237
Age6                    0.142611441 -1.069936806
Age7                    0.234838922 -0.836624591
Age8                    0.293033387 -0.918677089
Age9                    0.458669891 -0.947603500
Age10                   0.594256761 -0.864982679
Age11                   0.668847893 -1.403352563
Age12                   0.564262163 -2.002927327
Age13                   0.300191331 -1.777677947
Education2              0.192879323 -5.405897288
Education3             -0.064155244 -5.019215682
Education4             -0.158541296 -3.876376227
Education5             -0.135719471 -3.938943629
Education6             -0.167733428 -3.849643519
Income2                 0.044908728  0.620810831
Income3                -0.030045996  0.839557275
Income4                -0.078261719  1.065863768
Income5                -0.134666855  0.723120905
Income6                -0.192179595  0.557982089
Income7                -0.202692930  0.765312416
Income8                -0.279313222  1.143115782
```

After that I predict on the test set and build histogram:

 # Predict on the test set

ds.lda.values = predict(ds.lda.train, dsTest)
 # Look at histograms:
ldahist(data=ds.lda.values$x[, 1], g=dsTest$Diabetes_012)
ldahist(data=ds.lda.values$x[, 2], g=dsTest$Diabetes_012)
 # Plot the transformed data so we can see the classification
plot(ds.lda.values$x[, 1], ds.lda.values$x[, 2], col=dsTest$Diabetes_012, pch=16)
Looking at histogram, they are similar to the previous ones. We did little better but we really cannot separate them because this dataset is not a very separable dataset. There are lots of variables are in the similar distance with similar size and magnitude.





	0	1	2
0	41022	0	1669
1	788	0	122
2	5389	0	1746

Appendix B: R code

PFA

```
# Importing and Preparing Data
diabetes = read.csv('diabetes_health_indicators.csv')
diabetes_cat = diabetes[, c(1:4, 6:15, 18:22)]
diabetes_cat[] = lapply(diabetes_cat, factor)
diabetes_num = diabetes[, c(5, 16:17)]
diabetes = cbind(diabetes_cat, diabetes_num)
diabetes2 = diabetes[-1]
summary(diabetes2)

# Libraries
library(corrplot)
library(MASS)
library(psych)
library(polycor)

# Correlation analysis
cat.cor = hetcor(diabetes2)
print(cat.cor)
corrplot(cat.cor$correlations, method='ellipse', order='AOE')
catcortest = corr.test(cat.cor$correlations, adjust='none')
catcortest.clean = ifelse(catcortest$p < 0.01, T, F)
colSums(catcortest.clean) - 1

# Removing variables with no significant correlations and re-computing correlation matrix
diabetes3 = subset(diabetes2, select=-c(CholCheck, BMI, Sex, HvyAlcoholConsump))
cat.cor2 = hetcor(diabetes3)

# PCA to determine number of factors
dia.pca = prcomp(cat.cor2$correlations)
print(dia.pca)
summary(dia.pca)
plot(dia.pca)
pa = fa.parallel(cat.cor2$correlations, n.iter=500)

# PFA with principal()
dia.prin = principal(cat.cor2$correlations, nfactors=2)
print(dia.prin$loadings, cutoff=0.4, sort=T)

# CFA for confirmatory analysis + chi-square
dia.fact = factanal(cov=cat.cor2$correlations, factors=2, n.obs=253680)
print(dia.fact$loadings, cutoff=0.4, sort=T)
print(dia.fact)
```

LDA

```
library(tidyverse)
library(corrplot)
library(plyr)
library(ggplot2)
library(RCurl)
library(psych)
library(MASS)
library(caret)
library(car)

# Load and explore our data
ds = read.csv("diabetes_012_health_indicators_BRFSS2015.csv")
head(ds)

# Transform some variables to factors
ds$Diabetes_012 = as.factor(ds$Diabetes_012)
ds$GenHlth = as.factor(ds$GenHlth)
ds$Age = as.factor(ds$Age)
ds$Education = as.factor(ds$Education)
ds$Income = as.factor(ds$Income)

ds$HighBP = as.factor(ds$HighBP)
ds$HighChol = as.factor(ds$HighChol)
ds$CholCheck = as.factor(ds$CholCheck)
ds$Smoker = as.factor(ds$Smoker)
ds$Stroke = as.factor(ds$Stroke)
ds$HeartDiseaseorAttack = as.factor(ds$HeartDiseaseorAttack)
ds$PhysActivity = as.factor(ds$PhysActivity)
ds$Fruits = as.factor(ds$Fruits)
ds$Veggies = as.factor(ds$Veggies)
ds$HvyAlcoholConsump = as.factor(ds$HvyAlcoholConsump)
ds$AnyHealthcare = as.factor(ds$AnyHealthcare)
ds$NoDocbcCost = as.factor(ds$NoDocbcCost)
ds$DiffWalk = as.factor(ds$DiffWalk)
ds$Sex = as.factor(ds$Sex)

summary(ds$Diabetes_012)   #three types 0-no diabetes or only during pregnancy
                    #1-prediabetes and 2-diabetes

# Try an initial lda on everything
ds.lda = lda(Diabetes_012 ~ ., data=ds)
# Look at the output
print(ds.lda)
# Print the scaling:
print(ds.lda$scaling[order(ds.lda$scaling[, 1]), ])
print(ds.lda$scaling[order(ds.lda$scaling[, 2]), ])
# Look at the separation
```

```
ds.lda.values = predict(ds.lda)
ldahist(data=ds.lda.values$x[, 1], g=ds$Diabetes_012)
ldahist(data=ds.lda.values$x[, 2], g=ds$Diabetes_012)
# Plot the transformed data:
plot(ds.lda.values$x[, 1], ds.lda.values$x[, 2], col=ds$Diabetes_012, pch=16)
# Compute a confusion matrix
table(ds$Diabetes_012, ds.lda.values$class)
confusionMatrix(ds$Diabetes_012, ds.lda.values$class)

# Now, let's separate into test and training set:
s = sample(nrow(ds), nrow(ds) * .8)
dsTrain = ds[s, ]
dsTest = ds[-s, ]

# Build the model on the training set
ds.lda.train = lda(Diabetes_012 ~ ., data=dsTrain)
# Look at the output
ds.lda.train
# Print the scaling:
print(ds.lda.train$scaling[order(ds.lda.train$scaling[, 1]), ])
# Predict on the test set
ds.lda.values = predict(ds.lda.train, dsTest)
# Look at histograms:
ldahist(data=ds.lda.values$x[, 1], g=dsTest$Diabetes_012)
ldahist(data=ds.lda.values$x[, 2], g=dsTest$Diabetes_012)
# Plot the transformed data so we can see the classification
plot(ds.lda.values$x[, 1], ds.lda.values$x[, 2], col=dsTest$Diabetes_012, pch=16)
# Compute a confusion matrix
table(dsTest$Diabetes_012, ds.lda.values$class)
confusionMatrix(dsTest$Diabetes_012, ds.lda.values$class)
```

CA/MCA

```
library("FactoMineR")
library("factoextra")
library(ca)
library(dplyr)

################### Pre Processing
diabetes =
read.csv("C:/Users/devjrr/Documents/DSC424/FinalProject/diabetes_012_health_indicators/diabetes_0
12_health_indicators_BRFSS2015.csv")
head(diabetes)

########## Convert binary to factors
```

```
diabetes4 = subset(diabetes, select =
c(Diabetes_012,HighBP,HighChol,CholCheck,Smoker,Stroke,HeartDiseaseorAttack,PhysActivity,Fruits,Ve
ggies,HvyAlcoholConsump,AnyHealthcare,NoDocbcCost,DiffWalk,Sex) )

diabetes4$Diabetes_012[diabetes4$Diabetes_012 == 0] <- "NoDiabetes"
diabetes4$Diabetes_012[diabetes4$Diabetes_012 == 1] <- "PreDiabetes"
diabetes4$Diabetes_012[diabetes4$Diabetes_012 == 2] <- "Diabetes"
diabetes4$Diabetes_012 <- as.factor(diabetes4$Diabetes_012)
diabetes4$HighBP[diabetes4$HighBP == 1] <- "HighBP_Y"
diabetes4$HighBP[diabetes4$HighBP == 0] <- "HighBP_N"
diabetes4$HighBP <- as.factor(diabetes4$HighBP)
diabetes4$HighChol[diabetes4$HighChol == 1] <- "HighChol_Y"
diabetes4$HighChol[diabetes4$HighChol == 0] <- "HighChol_N"
diabetes4$HighChol <- as.factor(diabetes4$HighChol)
diabetes4$CholCheck[diabetes4$CholCheck == 1] <- "CholCheck_Y"
diabetes4$CholCheck[diabetes4$CholCheck == 0] <- "CholCheck_N"
diabetes4$CholCheck <- as.factor(diabetes4$CholCheck)
diabetes4$Smoker[diabetes4$Smoker == 1] <- "Smoker_Y"
diabetes4$Smoker[diabetes4$Smoker== 0] <- "Smoker_N"
diabetes4$Smoker <- as.factor(diabetes4$Smoker)
diabetes4$Stroke[diabetes4$Stroke == 1] <- "Stroke_Y"
diabetes4$Stroke[diabetes4$Stroke == 0] <- "Stroke_N"
diabetes4$Stroke <- as.factor(diabetes4$Stroke)
diabetes4$HeartDiseaseorAttack[diabetes4$HeartDiseaseorAttack == 1] <- "HeartDiseaseorAttack_Y"
diabetes4$HeartDiseaseorAttack[diabetes4$HeartDiseaseorAttack == 0] <- "HeartDiseaseorAttack_N"
diabetes4$HeartDiseaseorAttack <- as.factor(diabetes4$HeartDiseaseorAttack)
diabetes4$PhysActivity[diabetes4$PhysActivity == 1] <- "PhysActivity_Y"
diabetes4$PhysActivity[diabetes4$PhysActivity == 0] <- "PhysActivity_N"
diabetes4$PhysActivity <- as.factor(diabetes4$PhysActivity)
diabetes4$Fruits[diabetes4$Fruits == 1] <- "Fruits_Y"
diabetes4$Fruits[diabetes4$Fruits == 0] <- "Fruits_N"
diabetes4$Fruits <- as.factor(diabetes4$Fruits)
diabetes4$Veggies[diabetes4$Veggies == 1] <- "Veggies_Y"
diabetes4$Veggies[diabetes4$Veggies == 0] <- "Veggies_N"
diabetes4$Veggies <- as.factor(diabetes4$Veggies)
diabetes4$HvyAlcoholConsump[diabetes4$HvyAlcoholConsump == 1] <- "HvyAlcoholConsump_Y"
diabetes4$HvyAlcoholConsump[diabetes4$HvyAlcoholConsump == 0] <- "HvyAlcoholConsump_N"
diabetes4$HvyAlcoholConsump <- as.factor(diabetes4$HvyAlcoholConsump)
diabetes4$AnyHealthcare[diabetes4$AnyHealthcare == 1] <- "AnyHealthcare_Y"
diabetes4$AnyHealthcare[diabetes4$AnyHealthcare == 0] <- "AnyHealthcare_N"
diabetes4$AnyHealthcare <- as.factor(diabetes4$AnyHealthcare)
diabetes4$NoDocbcCost[diabetes4$NoDocbcCost == 1] <- "NoDocbcCost_Y"
diabetes4$NoDocbcCost[diabetes4$NoDocbcCost == 0] <- "NoDocbcCost_N"
diabetes4$NoDocbcCost <- as.factor(diabetes4$NoDocbcCost)
diabetes4$DiffWalk[diabetes4$DiffWalk == 1] <- "DiffWalk_Y"
diabetes4$DiffWalk[diabetes4$DiffWalk == 0] <- "DiffWalk_N"
diabetes4$DiffWalk <- as.factor(diabetes4$DiffWalk)
diabetes4$Sex[diabetes4$Sex == 1] <- "Sex_Y"
```

```
diabetes4$Sex[diabetes4$Sex == 0] <- "Sex_N"
diabetes4$Sex <- as.factor(diabetes4$Sex)

########################################################## CA

diabetes8 = subset(diabetes, select =
c(Diabetes_012,HighBP,HighChol,CholCheck,Smoker,Stroke,HeartDiseaseorAttack,PhysActivity,Fruits,Ve
ggies,HvyAlcoholConsump,AnyHealthcare,NoDocbcCost,DiffWalk,Sex) )
finald = diabetes8 %>%
  group_by(Diabetes_012) %>%
  summarise(across(everything(), sum), .groups = 'drop')
cadf = as.data.frame(finald)
head(cadf)
cadf$Diabetes_012 = NULL
rownames(cadf) = c( "NoDiabetes", "PreDiabetes","Diabetes")
head(cadf)
fit = ca(cadf)
fit
plot(fit)
plot(fit, mass=T, contrib="absolute",
    map="rowgreen", arrows=c(F, T))

######################### Multiple CA
summary(diabetes4)
dmca <- mjca(diabetes4)
print(dmca)
plot(dmca)
```