US Census 2017 Analysis

DCS 424 Advanced Data Analysis Winter 2020
Dr. John McDonald
Final Project
Group 7

Adam Frink, Jingxuan Yang, Sam Song

INDEX

Non-technical Summary

Abstract

The U.S. Constitution mandates that a census be taken every 10 years to count all people—both citizens and noncitizens—living in the United States. An accurate count of the population serves as the basis for fair political representation and plays a vital role in many areas of public life. For example, census totals help determine the amount of funding that state governments and local communities receive from the federal government for the next decade. Accurate census counts ensure that funding is equitably distributed for numerous programs such as Medicaid, highway planning and construction, special education grants to states, the National School Lunch Program, and Head Start (Mather & Scommegna, 2019). In this report, we perform two lines of analyses on the situation of unemployment by US census demographic data in 2017. We apply Principal Component Analysis (PCA) and Factor Analysis (FA) to reduce the number of variables of the final model and the discovery of latent factors. We analyze the correspondence between job and commute by using canonical correlation analysis (CCA). Then, using six factors from PCA, we perform linear regression with the situation of unemployment as a dependent variable to determine which variables have the highest impact on the performance of unemployment. The result shows that a moderate model could be created using the performance of unemployment and related predictors. Additionally, it shows that 61.03% of the variation in the unemployment variable can be explained by the regression model. The conclusion confirms that the income situation and job type are the most significant predictors to affect the performance of unemployment.

Data Set

The dataset used in this analysis is “US Census Demographic Data 2017” obtained from Kaggle.com[3]. This data set contains 37 predictors(columns) and 3221 rows one row for every county in the United States. Data points are aggregated on a per capita percentage grouped by county.

Technical Summary

- Data Cleaning:
  Before we completed our analysis, we first checked the values, found the missing data and performed data cleaning. Even though our data has only one row, it is critical to ensure that missing, incomplete, or otherwise defective data adversely affect results.

- CFA:

Factor analysis can help us find the latent factors which are correlated to our variables. We have more than 30 variables in our data. By using factor analysis, we confirmed these factors in data with a low dimension.

- PCA:
  Principal Component Analysis is a tool to help us view the correlation between variables and reduce the dimension of those large data. PCA rotates the observations, and in our data, helped us to check the multicollinearity situation and reduce the dimension of our data.

- Linear Regression Model:
  Linear regression model is a basic method of data analysis. We performed linear regression with the situation of unemployment as a dependent variable to determine which variables have the highest impact on the performance of unemployment. We used Ridge and Lasso to reduce the complexity and check the situation of overfitting. We applied the result of principal component analysis to replace most parts of original variables to solve the issue of multicollinearity and to reduce the number of variables.

- CCA:
  We used Canonical Correlations Analysis to visualize the correlation between several categorical variables. In our introduction of our data set, we mentioned that almost all variables are numeric or integers, so in our data, we used its applications on confirmation of the correlation between commute and job.

- LDA:
  We used Linear Discriminant Analysis for cases where there were two or more objects in data. For our data and goals, it helped to provide better interpretations for our variables like PFA. For our data, used it to build a better predictive model.

- Gradient Boosting (extra):
  A regression used for building prediction models which will help us to reach our goal: to predict the situation of unemployment and self-employed.

Common Factor Analysis (CFA)

Our data is about the census of America in 2017. There are many variables in our data set, which means it may be hard for us to study the correlations among them. By using CFA, we can reduce the number of factors and variables we need to observe, helping us find correlations between variables. In addition, like PCA, CFA also can be used for searching those hidden or latent variables. In our data with so many variables, CFA is critical to cut through noise and see correlations. For our directions, we hope to predict the Unemployment and Self-employed in our data. The correlation between them and other variables is the core part in our analysis.

The correlations among variables in our data are not very clear for us. We hope to use CFA to show the correlations and find those latent factors. We want to find latent variables because some of the information cannot be collected in the census but they are still important for solving our directions. First of all, we checked our data set and found a row with missing value, then we deleted it. All of our data are numeric so we do not need to create any dummy variable. To start factor analysis, we created the scree plot (Fig. A-1) to confirm how many factors we need. In Scree Plot, the knee point appears at the fifth factor so that we only need four factors in our factor analysis. These four factors are the latent factors and we would explain them later in the analysis and show what these factors represent.

In the first time of factor analysis, we did not rotate the factors and our analysis did not produce a useful result (The output of f1 in Fig.A-2). To get a better SS loading and explanation of factors, we tried varimax rotation in second factor analysis. At this time, we got a better result and the factors can be explained better with variables. From the difference of the model, we can also see the changes between f1 and f2 (Fig.A-3). The last step was to study the correlations among variables. We focused primarily upon the loadings (Fig.A-4). In loadings, for each factor, those non-relative variables are removed so that they can be explained with less variables. With the factors explained better, the relationship and weights of each variable become clear in our analysis.

From the output of loadings in R, we can explain 4 factors:

MR1=
0.984TotalPop+0.983Men+0.984Women+0.976VotingAgeCitizen+0.981Employed

MR2=
-0.687White-0.782Income-0.821IncomePerCap+0.920Poverty+0.887ChildPoverty+0.508PublicWork+0.741Unemployment

MR3=
0.566Professional-0.573Production-0.625Drive+0.548Walk-0.514PrivateWork+0.535PublicWork

MR4=
0.558IncomeErr+0.602IncomePerCapErr+0.604Construction+0.546WorkAtHome-0.577PrivateWork+0.726SelfEmployed

We set the cutoff value as .5. With this cutoff value, we think the number of variables left is suitable for most factors. Doing so enabled us to know what each factor represents and give each a name. The first factor mainly represents population, the second covers the economic situation, the third factor is relative to people with work and the last factor covers job situation.

In the PCA part, we used 6 factors in our analysis. To compare the difference between their results. We tried 6 factors in our factor analysis and repeated the process again. In this analysis, we found the sixth factor has a significant influence on the whole data (Fig.A-5). We kept using varimax in our factor analysis. From the result, we found the proportion of variance does not have a big change at this time. Finally, we checked the score and evaluated our hypothesis to see whether the factors are sufficient. From the chi-square test, we saw that both 4 and 6 factors are sufficient for the model. From the results, we think PCA has a better performance on our data.

Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is the general name for a technique that uses sophisticated underlying mathematical principles to transform a number of possibly correlated variables into a smaller number of variables called principal components (Adewumi, 2019). In this report, I apply Principal Component Analysis (PCA) to reduce the number of variables of the final model and discover latent factors. I implemented adequacy tests on our data to test if it is suitable for PCA. Adequacy tests include the correlation matrix, the Kaiser-Meyer-Olkin (KMO) test, and Bartlett's test of sphericity. The Kaiser-Meyer-Olkin Measure of Sampling Adequacy is a statistic that indicates the proportion of variance in your variables that might be caused by underlying factors. High values (close to 1.0) generally indicate that factor analysis may be useful with your data. If the value is less than 0.50, the results of the factor analysis probably won't be very useful [1].

Bartlett's test of sphericity tests the hypothesis that your correlation matrix is an identity matrix, which would indicate that your variables are unrelated and therefore unsuitable for structure detection. Small values (less than 0.05) of the significance level indicate that factor analysis may be useful with your data [2].

I apply the Pearson r correlation and Spearman rank correlation to analyze the association between each pair of numeric variables. According to the correlation matrix of spearman rank correlation with its visualization plot (Appendix B1), I discovered that the issue of multicollinearity is significant. There are more than 10 pairs of variables in which the R-value is larger than 0.8. According to the correlation matrix of Pearson r correlation with its visualization plot (Appendix B2), the associations between each pair of variables are strong. For example, the “men” variable has a very strong positive association with the “employed” variable. The R-value between the “men” variable and the “employed” variable is close to 1. The “women” variable also has a very strong positive association with the “employed” variable. The R-value between the “women” variable and the “employed” variable is also close to 1. They mean that the model exists in the issue of multicollinearity.

Next, we measured how many significant correlations each of the variables has. According to the test (Appendix B3), the “income” variable has the largest amount of significant correlations with other variables. “Income” variable has 19 significant correlations. “CountyId” variable and “Pacific” variable have the smallest amount of significant correlations with other variables. There exists one significant correlation.

After measuring the correlations, I implemented adequacy tests on our data to test if it was suitable for PCA. According to the result of Bartlett's test of sphericity test (Appendix B4), the p-value is lower than 2.22e-16 which indicates that factor analysis may be useful with our data. According to the result of the KMO test (Appendix B5), the KMO value is 0.5, indicating the sampling is not adequate and that remedial action was required.

Before performing PCA on a dataset, had to determine the number of principal components that represent the dataset. According to the scree plot graph (Appendix B6) and parallel analysis scree plot graph (Appendix B7), we could keep only the top 6 components whose cumulative proportion is 64.23%. To implement a cutoff of 0.5, we generated the factor loadings for all relevant factors (Appendix B8&B9).

Factor 1: Population Information

Factor 1 = 0.983*TotalPop + 0.983*Men + 0.983*Women + 0.978*VotingAgeCitizen + 0.981*Employed

Factor 2: Income Information

Factor 2 = 0.605*Hispanic – 0.762*White – 0.741*Income – 0.777*IncomePerCap + 0.914*Poverty + 0.882*ChildPoverty + 0.507*Service + 0.528*PublicWork +0.783*Unemployment

Factor 3: Work Information

Factor 3 = 0.685*IncomeErr + 0.695*IncomeCapErr – 0.516*Office + 0.719*Construction – 0.505*PrivateWork + 0.702*SelfEmployed

Factor 4: Getting Around

Factor 4 = 0.671*Native – 0.643*Drive + 0.710*Walk + 0.626*OtherTransp

Factor 5: Professional & Production

Factor 5 = 0.700*Professional – 0.787*Production

Factor 6: Race

Factor 6 = 0.604*Asian + 0.507*Pacific

The principal component analysis is a technique for feature extraction — so it combines our input variables in a specific way, then we can drop the “least important” variables while still retaining the most valuable parts of all of the variables (Adewumi, 2019). It could help us to discover the underlying factors and eliminate the issue of multicollinearity. In this report, we retained six principal components which replace 27 of the original variables. We could use the result of PCA to build a linear regression model for predicting the performance of “unemployment”.

Linear Regression model

In this report, we combined the previous knowledge and the concept of what we learned in this quarter to analyze the linear regression model. We applied the regularization regression model to solve the issue of overfitting. We also applied the result of principal component analysis to replace most parts of original variables for solving the issue of multicollinearity and reducing the number of variables. Our full model includes the six principal components which replace 27 of original variables, as well as seven variables which exclude the principal components.

We applied the “unemployment” variable as a dependent variable in the first direction. The dataset needs to be split into a training set and testing set. In this case, the training set takes 70% of the dataset. The testing set takes 30% of the dataset. The sample size is 2,226. We created a linear regression model whose dependent variable is the “unemployment” variable. In the OLS regression model of “unemployment” (Appendix B10, 11, 12), the R^2 of the training set is 62.48%. The RMSE of the training set is 2.3637. The RMSE of the testing set is 2.1478. We applied the regularization regression analysis to handle the “unemployment” model. The regularization regression analysis includes ridge regression analysis, lasso regression analysis, and elastic net regression analysis.

Table 1
The Result of Regularization Regression Model

Unemployment	RMSE (Train)	RMSE (Test)	R^2
OLS	2.3637	2.1478	0.6248
Ridge	2.3844	2.1568	0.6182
Lasso	2.3752	2.1442	0.6211

Elastic Net	2.3774	2.1421	0.6204

According to the result of table 1, it is obvious the results are very similar. The difference is not significant, and could reasonably be ignored. The “unemployment” model doesn’t address the issue of overfitting.

The distribution of the “unemployment” variable is skewed, so we applied the log transformation to normalize it.

Next, we computed the VIF values to check the issue of multicollinearity. There is no variable in which the VIF value is larger than 10. This means that the model doesn’t exist in the issue of multicollinearity. The correlation plot confirms this conclusion (Appendix B13).

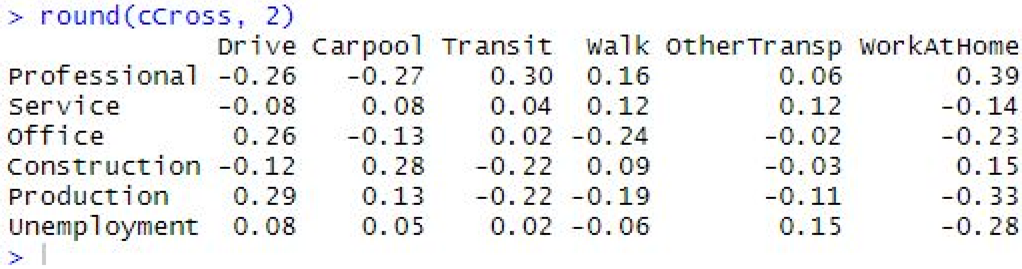
Then, we applied the forward model selection to remove the insignificant predictors and use the goodness of fit to test the performance of the final model. After applying forward model selection, the p-value of every variable is lower than 0.05 (Appendix B14). The RMSE of the training set is 0.336. The RMSE of the testing set is 0.308. The F-value is 320. Adj-R^2 is 0.6103. This means that 61.03% of the variation in the unemployment variable can be explained by the multiple regression model.

Unemployment_log = 1.198 – 0.002*Black + 0.007*Carpool + 0.019*WorkAtHome + 0.017*MeanCommute + 0.095*FamilyWork – 0.006*Population_Information + 0.063*Income_Information – 0.064*Work_Information + 0.029*Getting_Around – 0.022*Professional_Information + 0.023*Race

I tested the values of the standardized coefficient to check which predictor has the greatest influence on the “unemployment_log” variable. The “income_information” variable and “work_information” variable have the largest absolute value of the standardized coefficient (Appendix B15). They mean that the income situation and job type are the most significant predictors to affect the performance of unemployment.

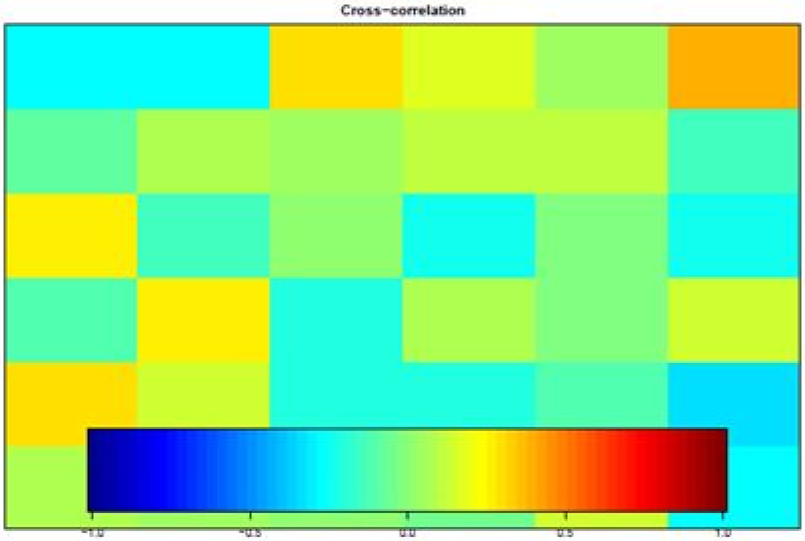
Canonical Correlation

The second line of investigation chosen was to investigate if a relationship existed between job category and commute category. The first technique employed to analyze this relationship was canonical correlation. An initial corrplot [Appendix C] and analysis of the correlation matrix showed some slight, but interesting correlations. For example, the *Professional* job category has the highest correlation with the *WorkAtHome* commute type than any other job type with a correlation of 0.39. The *Professional* job category also has a slight correlation with taking *Transit* 0.3 and a slight negative correlation with *Carpooling* -0.3. There are other interesting relationships such as *Constructio*n and *Production* are the only two job categories to have a negative correlation with taking *Transit* both at -0.22. Observing the correlation matrix between job type and commute type we can see some profiles emerge of how certain professions commute to work though the correlations are modest.

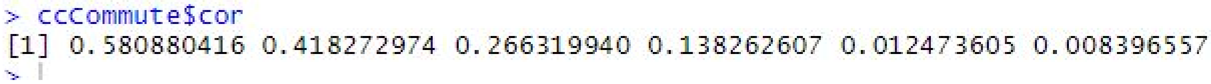


```
> round(cCross, 2)
             Drive Carpool Transit  Walk OtherTransp WorkAtHome
Professional -0.26   -0.27    0.30  0.16        0.06       0.39
Service      -0.08    0.08    0.04  0.12        0.12      -0.14
Office        0.26   -0.13    0.02 -0.24       -0.02      -0.23
Construction -0.12    0.28   -0.22  0.09       -0.03       0.15
Production    0.29    0.13   -0.22 -0.19       -0.11      -0.33
Unemployment  0.08    0.05    0.02 -0.06        0.15      -0.28
> 
```

The cross correlation plot of the matcor function gives a color ramp graphic of this.

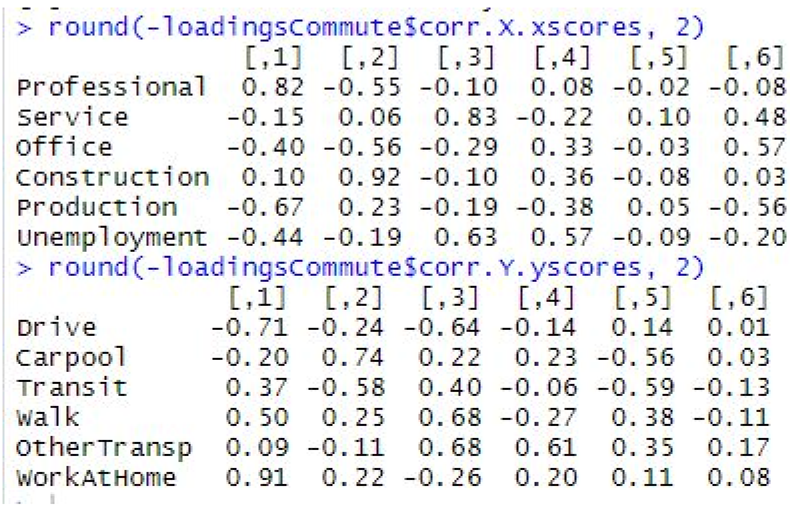


Computing the canonical correlation between the job and commute categories we seem to have two strong variates V1 = .58 and V2 = .41. A Wilks test would be applied here to determine which variants are significant. However, the Wilks function from class errors out on this data set (see appendix C for more detail), so this is an estimate.



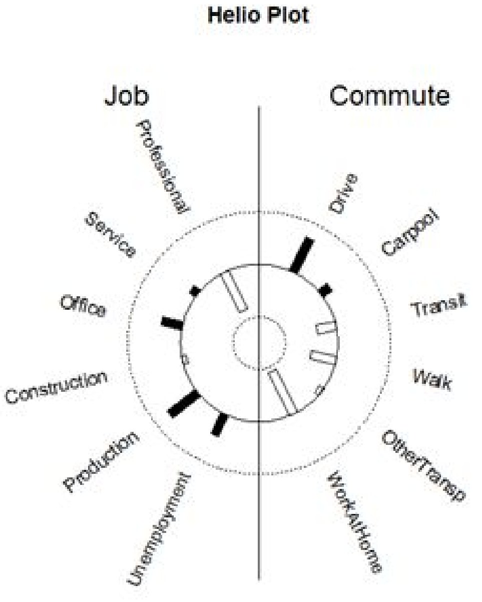
```
> ccCommute$cor
[1] 0.580880416 0.418272974 0.266319940 0.138262607 0.012473605 0.008396557
> 
```

Analyzing the loadings of the canonical correlation we can see that Variant 1 includes the relationship between *Professionals* 0.82 and *WorkingAtHome* 0.91, which was the strongest correlation observed in the initial assessment. Variant 1 also includes the correlation between *Office* -0.4 and *Production* -0.67 workers positive correlation with *Driving* -0.71. Variant 2 represents the relationship between *Construction* 0.92 workers and their positive correlation with *Carpooling* 0.74.

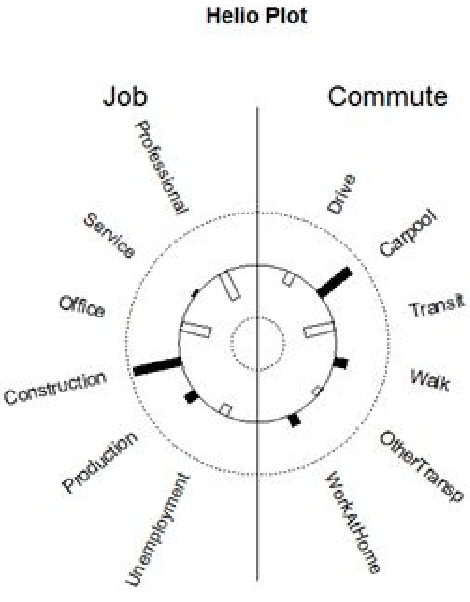


```
> round(-loadingsCommute$corr.X.xscores, 2)
              [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
Professional  0.82 -0.55 -0.10  0.08 -0.02 -0.08
Service      -0.15  0.06  0.83 -0.22  0.10  0.48
Office       -0.40 -0.56 -0.29  0.33 -0.03  0.57
Construction  0.10  0.92 -0.10  0.36 -0.08  0.03
Production   -0.67  0.23 -0.19 -0.38  0.05 -0.56
Unemployment -0.44 -0.19  0.63  0.57 -0.09 -0.20
> round(-loadingsCommute$corr.Y.yscores, 2)
             [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
Drive       -0.71 -0.24 -0.64 -0.14  0.14  0.01
Carpool     -0.20  0.74  0.22  0.23 -0.56  0.03
Transit      0.37 -0.58  0.40 -0.06 -0.59 -0.13
Walk         0.50  0.25  0.68 -0.27  0.38 -0.11
OtherTransp  0.09 -0.11  0.68  0.61  0.35  0.17
WorkAtHome   0.91  0.22 -0.26  0.20  0.11  0.08
```

The relationship of the variants to their primary coefficients can more easily be observed in the following heilo plots. In this model, the dark bars represent positive correlation, the white bars represent negative correlation. The size of the bars represent the strength of correlation.



Canonical Variate1



Canonical Variate2

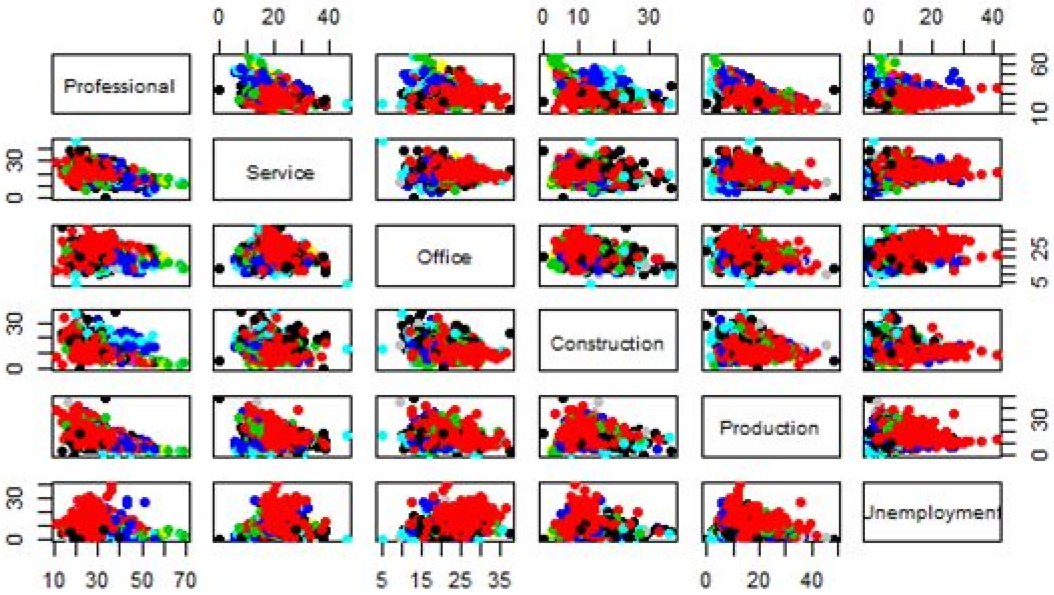
Canonical correlation shows that relationships between job type and commute type do exist. However, the strength of those correlations is slight. Also, validation of the significance of these correlations is incomplete due to issues with the Wilks test function used in class (see more on this in Appendix C). Also, chi square will not work as a validation tool for this data set since it is meant to measure frequency data and the data in this analysis is aggregated.

LDA

The second technique used in analysis of the job type and commute type data was LDA (Linear Discriminant Analysis). To facilitate this the data was grouped into a new categorical variable named ‘region’. LDA was then implemented to see if regions could be distinguished by the per capita percentage employed in a job type or commute type.

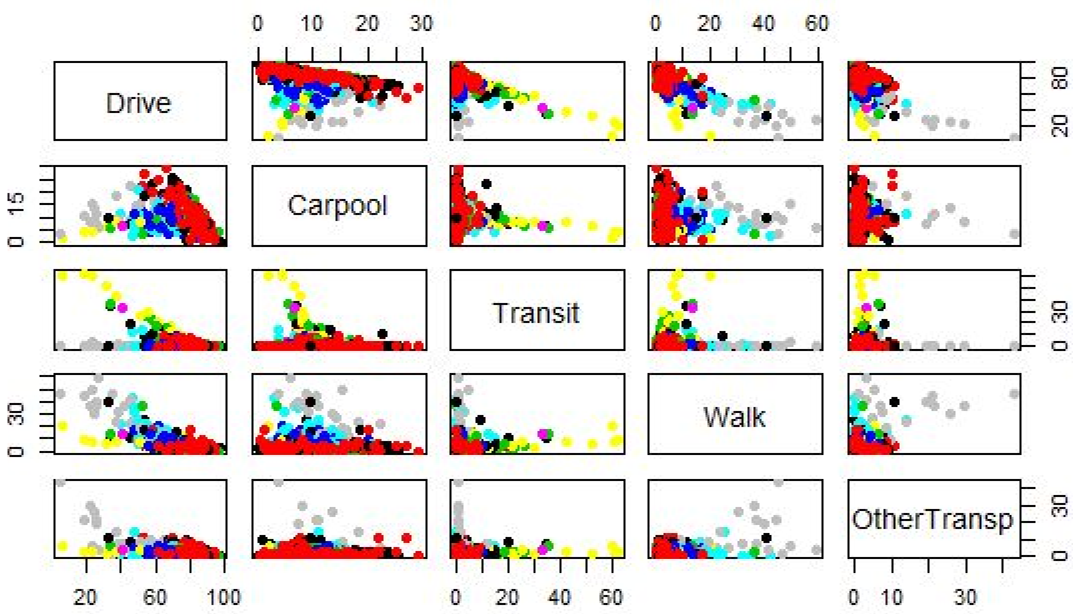
A new categorical variable named region was created by classifying the data by state name into one of the 10 region partitions used by the Law School Admission Council LSAC on their website22. LSAC, however, does not group Alaska into a region. Therefore, Alaska was added to the NW region with Oregon and Washington.

Initial plot of region (colored) by job type



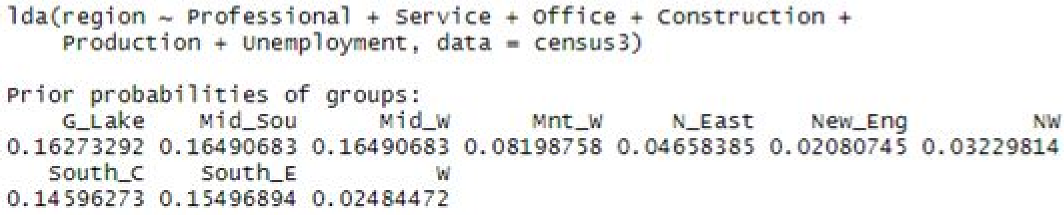
Analysis: The regions overlap each other. No observable grouping by job type.

Initial plot of region (colored) by commute type

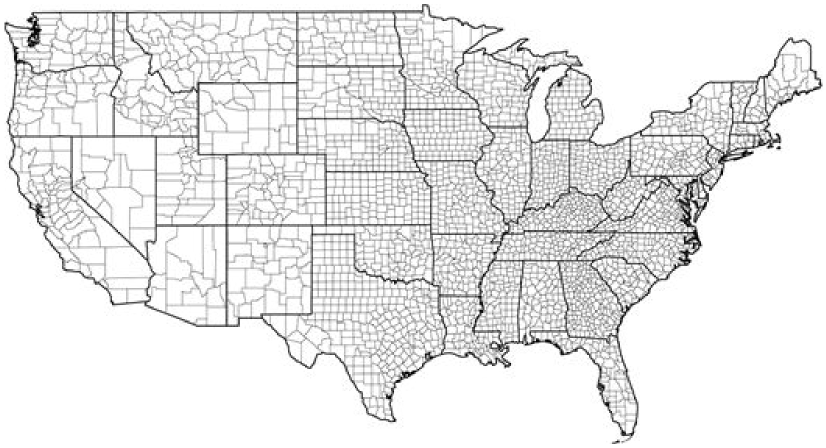


Analysis: The regions overlap each other, but there appears to be a little bit of separation by region based on commute type.

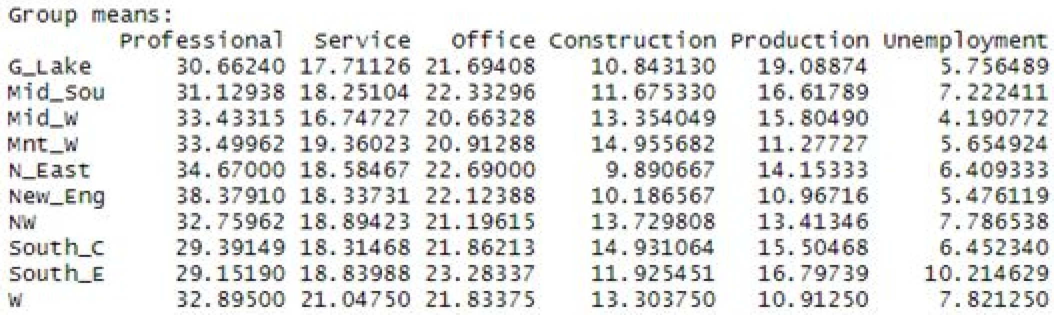
The output from the lda() function provides some interesting insight. First, the probabilities of the regions are not evenly distributed. The West and NorthWest are the least represented. This is likely due to the data’s grouping by county. The Western part of the United States has far larger county sizes in terms of geographical size than the rest of the country.



```
lda(region ~ Professional + Service + Office + Construction + 
    Production + Unemployment, data = census3)

Prior probabilities of groups:
    G_Lake    Mid_Sou      Mid_W      Mnt_W     N_East    New_Eng         NW 
0.16273292 0.16490683 0.16490683 0.08198758 0.04658385 0.02080745 0.03229814 
   South_C    South_E          W 
0.14596273 0.15496894 0.02484472 
```

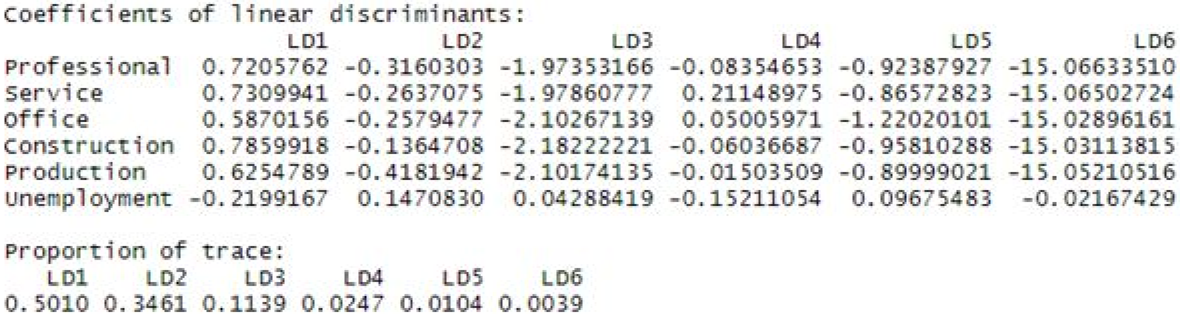


Second, the group means for the job types are all VERY similar. There are a couple that stand out like the Great lake region has a high production average and the Southeast has high unemployment, but even these extremes are not that much different from the other regions and this is the underlying reason why job type value will fail to differentiate the different regions.



```
Group means:
         Professional  Service   Office Construction Production Unemployment
G_Lake       30.66240 17.71126 21.69408    10.843130   19.08874     5.756489
Mid_Sou      31.12938 18.25104 22.33296    11.675330   16.61789     7.222411
Mid_W        33.43315 16.74727 20.66328    13.354049   15.80490     4.190772
Mnt_W        33.49962 19.36023 20.91288    14.955682   11.27727     5.654924
N_East       34.67000 18.58467 22.69000     9.890667   14.15333     6.409333
New_Eng      38.37910 18.33731 22.12388    10.186567   10.96716     5.476119
NW           32.75962 18.89423 21.19615    13.729808   13.41346     7.786538
South_C      29.39149 18.31468 21.86213    14.931064   15.50468     6.452340
South_E      29.15190 18.83988 23.28337    11.925451   16.79739    10.214629
W            32.89500 21.04750 21.83375    13.303750   10.91250     7.821250
```

Third, the coefficients of linear discriminants, like the group means above, are similar and they generally move in the same direction (are all positive or negative at the same time) for a given LD. The exception to this is Unemployment which generally has an opposite sign as the other job type confidents. Perhaps this is because unemployment more represents a lack of a job type than being a job type itself.



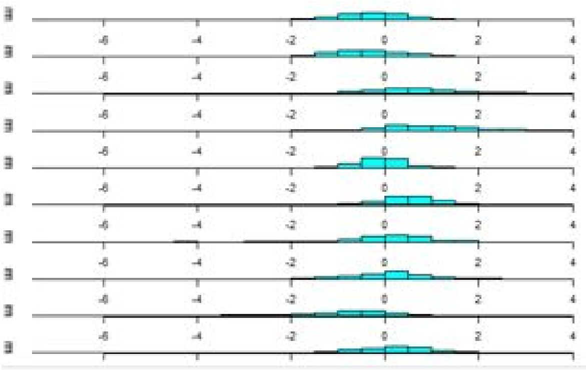
```
Coefficients of linear discriminants:
                    LD1        LD2         LD3         LD4         LD5          LD6
Professional  0.7205762 -0.3160303 -1.97353166 -0.08354653 -0.92387927 -15.06633510
Service       0.7309941 -0.2637075 -1.97860777  0.21148975 -0.86572823 -15.06502724
Office        0.5870156 -0.2579477 -2.10267139  0.05005971 -1.22020101 -15.02896161
Construction  0.7859918 -0.1364708 -2.18222221 -0.06036687 -0.95810288 -15.03113815
Production    0.6254789 -0.4181942 -2.10174135 -0.01503509 -0.89999021 -15.05210516
Unemployment -0.2199167  0.1470830  0.04288419 -0.15211054  0.09675483  -0.02167429

Proportion of trace:
   LD1    LD2    LD3    LD4    LD5    LD6 
0.5010 0.3461 0.1139 0.0247 0.0104 0.0039
```

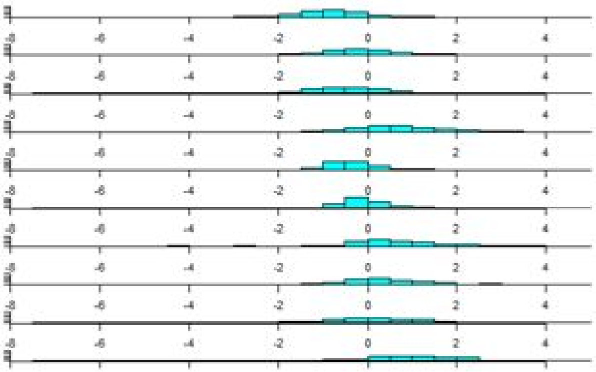
Fourth, the majority of the trace can be found in the first two linear discriminates. LD1 represents ~50% and LD2 ~34%. Together they account for 84% of the trace.

Looking at the discriminant histograms we see an illustration of what is observed with the group means. The distribution of job types are very similar throughout the regions. Some might be skewed and have a long tail in one direction, but for the most part they overlap making the regions impossible to separate on job type alone.

LD1

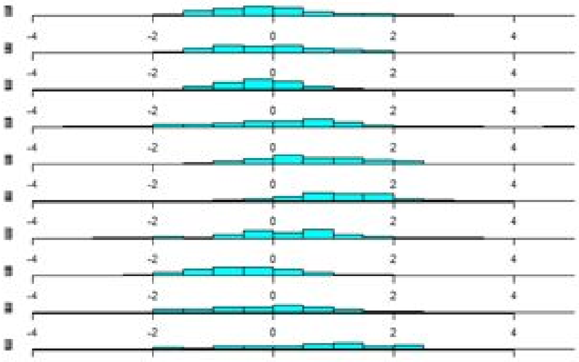
LD2

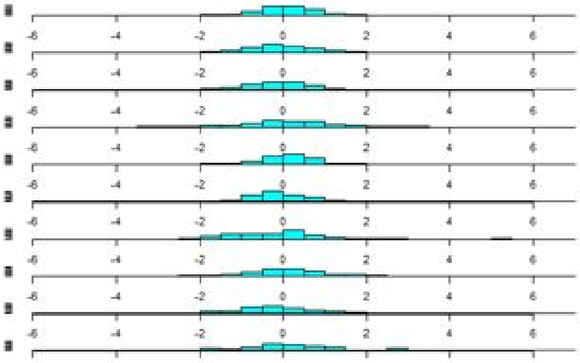




LD3

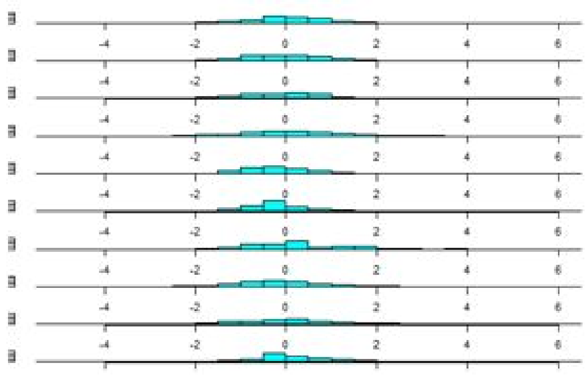
LD4

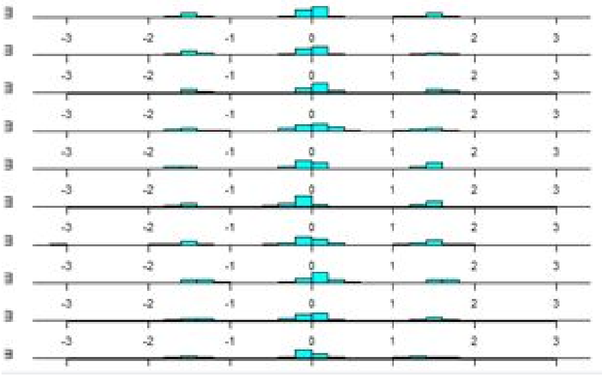




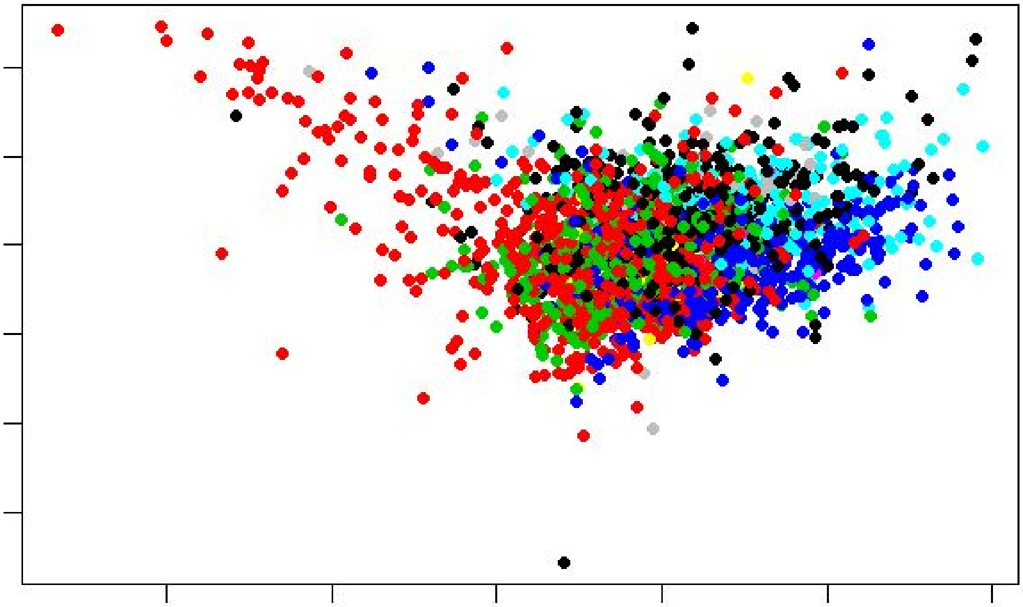
LD5

LD6





Plotting the loading of LD1 and LD2, which together represent 84% of our trace, we get the following plot. The regions are overlapping based on job type, with a few of the job types that are skewed one way or the other having some separation with part of its data points, with the bulk of the data points still overlapping and bunched together.



In conclusion U.S. regions in this data set cannot be partitioned into distinct grouping based on job type using LDA because the distribution of job type is too similar across the various regions. The same is true for commute type. The analysis for using commute type independent variables to separate regions was not included in this analysis, because the result is the same as what was observed using job type. One potential issue might be the format of the data in that we are analyzing aggregate data.

Gradient Boosting (Extra)

Gradient Boosting is a tool used for helping us build a predictive model of our data set. It is a kind of machine learning technique. We want to use it because we think it will be very helpful in predicting the situation of Self-Employed and Unemployment. This technique is relative to several functions such as iteratively.

In R, the package “gbm” is required for this part. This package contains a function called gbm. In this function, we have to call the data set we used, the distribution, the shrinkage and n.trees. The distribution means the assumptions about the data distribution of the output variables, which is required when we calculate the loss function. The shrinkage is the speed of moving to gradient descent in each iteration. In most situations, the smaller shrinkage means the better model of boosting. The n.trees, which represents number of iterations, an important element related to learning rate and performance of model.

Before we built our gradient boosting model, we divided our data set into train sets and testing sets, like what we did in Ridge and Lasso Regression. Then we use the train set and call the predictor Self-Employed dependent variable. We used “gaussian” as distribution which is mostly used in categorical objects. The shrinkage is 0.01 which is small enough for our predictive model. We let the number of iterations equal to 10000 and its depth equal to 4.

Then we can get our predictive model (Fig.D-1). From the plot, we can find that the total population, employed and drive commute have the most influence on Self-Employed. If we compared the result with the prediction of Ridge and Lasso, we can find some difference between them. If we want to know more about its performance, we need to test its accuracy of prediction and that is why we splitted train set and testing set. Then let check the result of testing (Fig.D-2). By comparing these 2 results and calculating the accuracy of prediction, we can know that the model does not have a good performance on our data.

Future Work

Additional validation of the canonical correlation variants using a wilks test as well as other validation techniques.It would also be interesting to perform our analysis on the full, non-aggregated, US census dataset. This dataset would be huge! In the millions of rows. However, it would allow us to analyze a finer grain of the data and it would be interesting to see if the results are any different.

Conclusion

We apply PCA and factor analysis to reduce the number of variables and discover the underlying factors. Because of the result of the parallel scree plot, PCA retains 6 principal components and factor analysis retains 4 components. However, the difference is not significant. Because they don’t have a good performance according to their low KMO values. We use the result of PCA to build a linear regression model for predicting the performance of “unemployment”. The result shows that the income situation and job type have the most significant effect on the performance of unemployment.

GBM is a new technique we did not learn yet in our class. We used it because we believe it will be helpful in our prediction. Even though the model does not have good performance on our data, it helped us confirm some information and details in our data. For example, from the model, we can learn that the total population has the greatest influence on our dependent variables.

References

1.(n.d.). Retrieved from https://www.ibm.com/support/knowledgecenter/SSLVMB_23.0.0/spss/tutorials/fac_telco_kmo_ 01.html

2.(n.d.). Retrieved from https://www.ibm.com/support/knowledgecenter/SSLVMB_23.0.0/spss/tutorials/fac_telco_kmo_ 01.html

3.List of States in Regions, officialguide.lsac.org/release/Search/RegionList.aspx. https://officialguide.lsac.org/release/Search/RegionList.aspx

4. “US Census Demographic Data 2017.” Kaggle, 3 Mar. 2019, https://www.kaggle.com/muonneutrino/us-census-demographic-data

Adewumi, J. (2019, March 26). Understanding the Role of Eigenvectors and Eigenvalues in PCA Dimensionality Reduction. Retrieved from https://medium.com/@dareyadewumi650/understanding-the-role-of-eigenvectors-and-eigenvalues-in-pca-dimensionality-reduction-10186dad0c5c

Mather, M., & Scommegna, P. (2019, September 17). Why Is the U.S. Census So Important? Retrieved March 17, 2020, from https://www.prb.org/importance-of-us-census/

Appendix

Variables

Variable Name	Description	Data Type
CountyId	Census tract ID	Int
State	States of US	String
County	County or county equivalent	String
TotalPop	Total Population	Int
Men	Number of men	Int
Women	Number of women	Int
Hispanic	% of population that is Hispanic/Latino	Numeric
White	% of population that is White	Numeric
Black	% of population that is Black	Numeric

Native	% of population that is Native American or Alaskan	Numeric
Asian	% of population that is Asian	Numeric
Pacific	% of population that is Native Hawaiian or islander	Numeric
VotingAgeCitizen	Number of citizens can vote	Int
Income	Median household income ($)	Int
IncomeErr	Median household income error ($)	Int
IncomePerCap	Income per capita ($)	Int
IncomePerCapErr	Income per capita error ($)	Int
Poverty	% under poverty level	Numeric
ChildPoverty	% of child under poverty level	Numeric
Professional	% employed in management, business, science and arts	Numeric
Service	% employed in service jobs	Numeric
Office	% employed in sales and office jobs	Numeric
Construction	% employed in natural resources, construction and maintenance	Numeric
Production	% employed in production, transportation and material movement	Numeric

Drive	% commuting alone in a car, van or truck	Numeric
Carpool	% carpooling in a car, van or truck	Numeric
Transit	% commuting on public transportation	Numeric
Walk	% walking to work	Numeric
OtherTransp	% commuting via other means	Numeric
WorkAtHome	% working at home	Numeric
MeanCommute	Mean commute time (min)	Int
Employed	Number of employed (16+)	Int
PrivateWork	% employed in private industry	Numeric
PublicWork	% employed in public jobs	Numeric
SelfEmployed	% self-employed	Numeric
FamilyWork	% in unpaid family work	Numeric
Unemployed	Unemployed rate (%)	Numeric

Appendix A

Individual Report Jingxuan Yang

In this project, I discussed with my teammates about which data we can use. Most of the data sets are interesting. After we confirmed the data set, I helped my team build the histograms of our variables and clear up them. I did the part of factor analysis, compared its results with the output of PCA. I tried a different number of factors and rotations on factor analysis. I analyzed the results and shared them with my group members to discuss them. At last, I tried the gradient boosting for our project. I selected this technique because it can help build a predictive model and our goals is to predict and build models for some variables in our data set. I learned the technique and its code online. Then I applied it to our data. Fortunately, the model can provide some useful information for us.

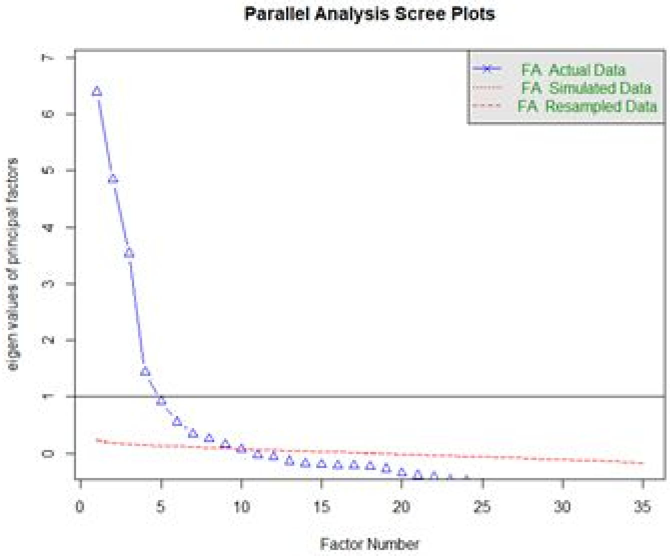
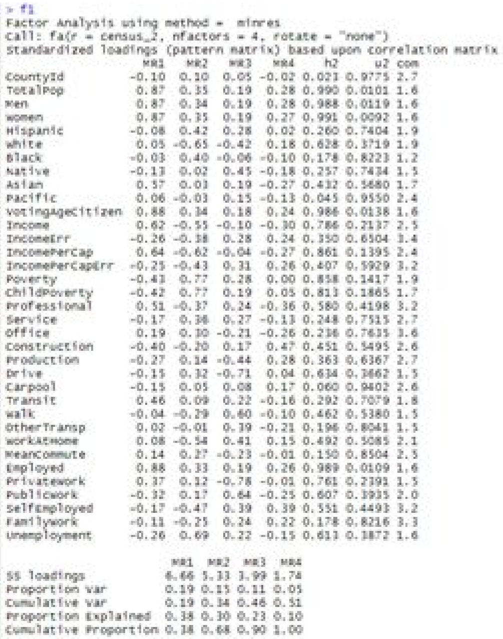


Fig.A-1 Scree Plot



```
> f1
Factor Analysis using method =  minres
Call: fa(r = census_2, nfactors = 4, rotate = "none")
Standardized loadings (pattern matrix) based upon correlation matrix
                   MR1   MR2   MR3   MR4    h2     u2 com
CountyId         -0.10  0.10  0.05 -0.02 0.023 0.9775 2.7
TotalPop          0.87  0.35  0.19  0.28 0.990 0.0101 1.6
Men               0.87  0.34  0.19  0.28 0.988 0.0119 1.6
Women             0.87  0.35  0.19  0.27 0.991 0.0092 1.6
Hispanic         -0.08  0.42  0.28  0.02 0.260 0.7404 1.9
White             0.05 -0.65 -0.42  0.18 0.628 0.3719 1.9
Black            -0.03  0.40 -0.06 -0.10 0.178 0.8223 1.2
Native           -0.13  0.02  0.45 -0.18 0.257 0.7434 1.5
Asian             0.57  0.03  0.19 -0.27 0.432 0.5680 1.7
Pacific           0.06 -0.03  0.15 -0.13 0.045 0.9550 2.4
VotingAgeCitizen  0.88  0.34  0.18  0.24 0.986 0.0138 1.6
Income            0.62 -0.55 -0.10 -0.30 0.786 0.2137 2.5
IncomeErr        -0.26 -0.38  0.28  0.24 0.350 0.6504 3.4
IncomePerCap      0.64 -0.62 -0.04 -0.27 0.861 0.1395 2.4
IncomePerCapErr  -0.25 -0.43  0.31  0.26 0.407 0.5929 3.2
Poverty          -0.43  0.77  0.28  0.00 0.858 0.1417 1.9
ChildPoverty     -0.42  0.77  0.19  0.05 0.813 0.1865 1.7
Professional      0.51 -0.37  0.24 -0.36 0.580 0.4198 3.2
Service          -0.17  0.38  0.27 -0.13 0.248 0.7515 2.7
Office            0.19  0.30 -0.21 -0.26 0.236 0.7635 3.6
Construction     -0.40 -0.20  0.17  0.47 0.451 0.5495 2.6
Production       -0.27  0.14 -0.44  0.28 0.363 0.6367 2.7
Drive            -0.15  0.32 -0.71  0.04 0.634 0.3662 1.5
Carpool          -0.15  0.05  0.08  0.17 0.060 0.9402 2.6
Transit           0.46  0.09  0.22 -0.16 0.292 0.7079 1.8
Walk             -0.04 -0.29  0.60 -0.10 0.462 0.5380 1.5
OtherTransp       0.02 -0.01  0.39 -0.21 0.196 0.8041 1.5
WorkAtHome        0.08 -0.54  0.41  0.15 0.492 0.5085 2.1
MeanCommute       0.14  0.27 -0.23 -0.01 0.150 0.8504 2.5
Employed          0.88  0.33  0.19  0.26 0.989 0.0109 1.6
PrivateWork       0.37  0.12 -0.78 -0.01 0.761 0.2391 1.5
PublicWork       -0.32  0.17  0.64 -0.25 0.607 0.3935 2.0
SelfEmployed     -0.17 -0.47  0.39  0.39 0.551 0.4493 3.2
FamilyWork       -0.11 -0.25  0.24  0.22 0.178 0.8216 3.3
Unemployment     -0.26  0.69  0.22 -0.15 0.613 0.3872 1.6

                       MR1  MR2  MR3  MR4
SS loadings           6.66 5.33 3.99 1.74
Proportion Var        0.19 0.15 0.11 0.05
Cumulative Var        0.19 0.34 0.46 0.51
Proportion Explained  0.38 0.30 0.23 0.10
Cumulative Proportion 0.38 0.68 0.90 1.00
```

Fig.A-2 Factor Analysis Output 1

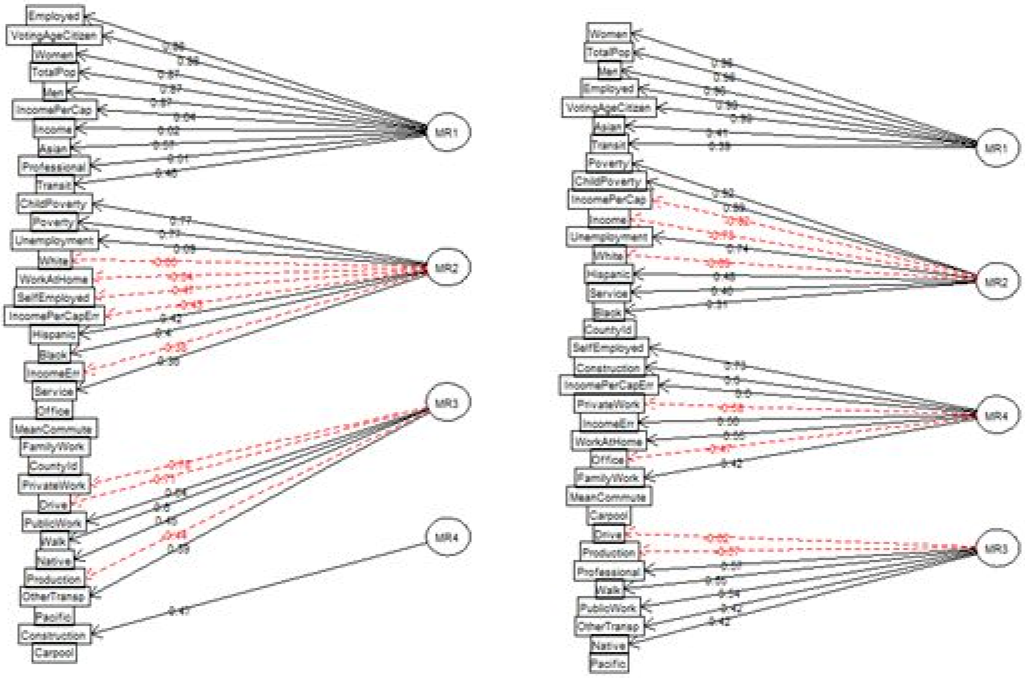
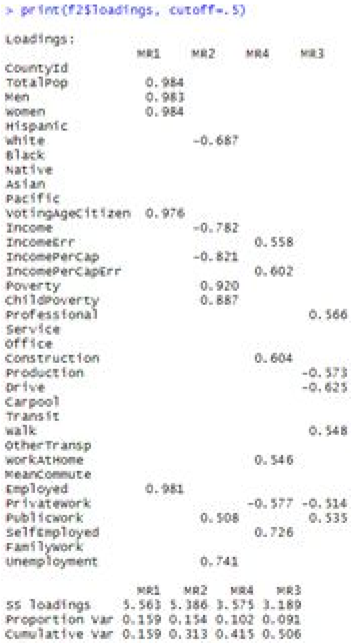
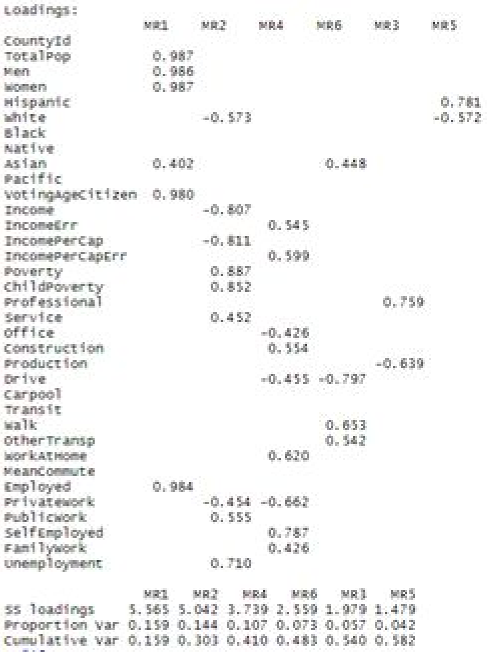


Fig.A-3 Factor Analysis Model



```
> print(f2$loadings, cutoff=.5)

Loadings:
                    MR1    MR2    MR4    MR3
CountyId
TotalPop          0.984
Men               0.983
Women             0.984
Hispanic
White                   -0.687
Black
Native
Asian
Pacific
VotingAgeCitizen  0.976
Income                  -0.782
IncomeErr                       0.558
IncomePerCap            -0.821
IncomePerCapErr                 0.602
Poverty                  0.920
ChildPoverty             0.887
Professional                           0.566
Service
Office
Construction                    0.604
Production                            -0.573
Drive                                 -0.625
Carpool
Transit
Walk                                   0.548
OtherTransp
WorkAtHome                      0.546
MeanCommute
Employed          0.981
PrivateWork                    -0.577 -0.514
PublicWork               0.508         0.535
SelfEmployed                    0.726
FamilyWork
Unemployment             0.741

                 MR1   MR2   MR4   MR3
SS loadings    5.563 5.386 3.575 3.189
Proportion Var 0.159 0.154 0.102 0.091
Cumulative Var 0.159 0.313 0.415 0.506
```

Fig.A-4 Factor Analysis Loading



```
Loadings:
                    MR1    MR2    MR4    MR6    MR3    MR5
CountyId
TotalPop          0.987
Men               0.986
Women             0.987
Hispanic                                               0.781
White                   -0.573                        -0.572
Black
Native
Asian             0.402                0.448
Pacific
VotingAgeCitizen  0.980
Income                  -0.807
IncomeErr                       0.545
IncomePerCap            -0.811
IncomePerCapErr                 0.599
Poverty                  0.887
ChildPoverty             0.852
Professional                                  0.759
Service                  0.452
Office                         -0.426
Construction                    0.554
Production                                   -0.639
Drive                          -0.455 -0.797
Carpool
Transit
Walk                                   0.653
OtherTransp                            0.542
WorkAtHome                      0.620
MeanCommute
Employed          0.984
PrivateWork             -0.454 -0.662
PublicWork               0.555
SelfEmployed                    0.787
FamilyWork                      0.426
Unemployment             0.710

                 MR1   MR2   MR4   MR6   MR3   MR5
SS loadings    5.565 5.042 3.739 2.559 1.979 1.479
Proportion Var 0.159 0.144 0.107 0.073 0.057 0.042
Cumulative Var 0.159 0.303 0.410 0.483 0.540 0.582
```

Fig.A-5 Result of nfactor6

Code:

```
# read data American Census 2017

census = read.csv("acs2017.csv")
head(census)

census_2 = as.data.frame(census[,-c(2,3)])
head(census_2)
```

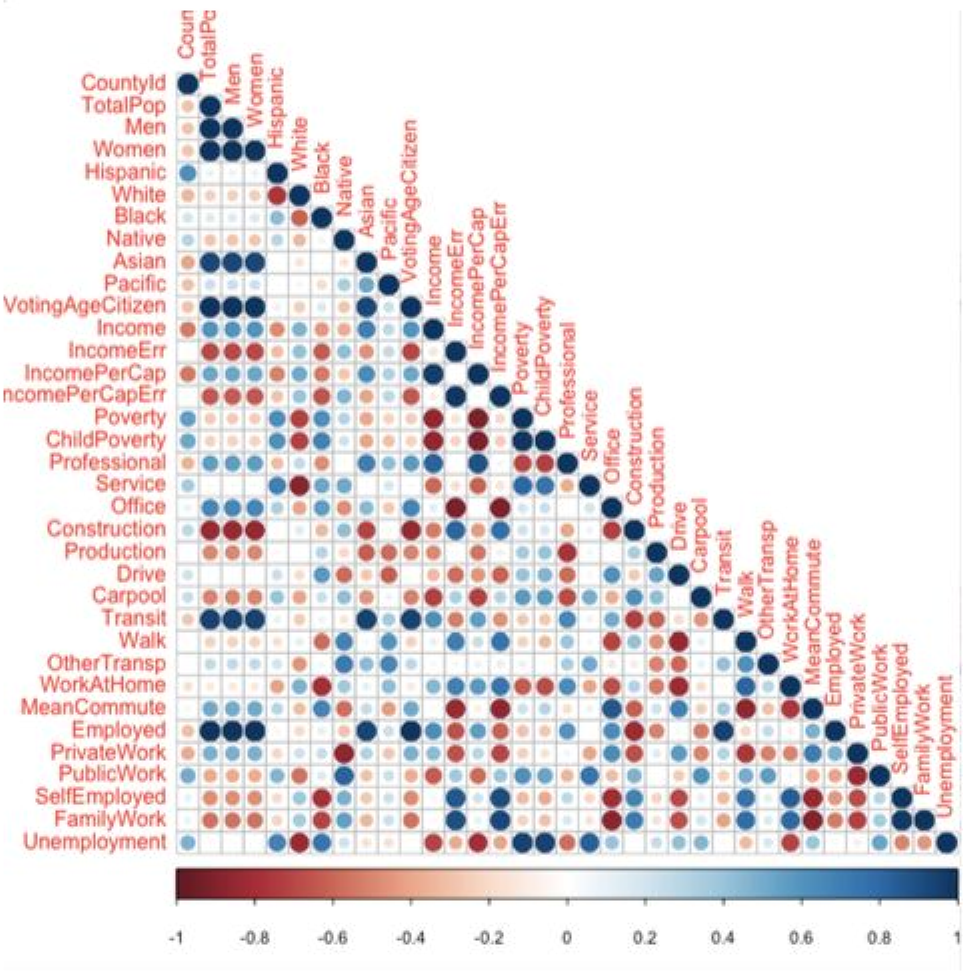
```
# we need the library psych
library(psych)

# Confirm how many factors we need in CFA
fn = fa.parallel(census_2,fm = 'minres',fa = 'fa')


# FA: use fa() function
f1 = fa(census_2, nfactor = 4, rotate = "none")
f1
factor.plot(f1,labels=rownames(f1$loadings))
fa.diagram(f1,digits = 2)
print(f1$loadings, cutoff=.4)

f1s = f1$scores
f1s

f2 = fa(census_2, nfactor = 4, rotate = "varimax")
f2
fa.diagram(f2,digits = 2)

print(f2$loadings, cutoff=.5)

f3 = fa(census_2, nfactor = 6, rotate = "none")
f3
fa.diagram(f3,digits = 2)
print(f3$loadings, cutoff=.4)

f4 = fa(census_2, nfactor = 6, rotate = "varimax")
f4
fa.diagram(f4,digits = 2)
print(f4$loadings, cutoff=.4)

f4$scores
```

Appendix B

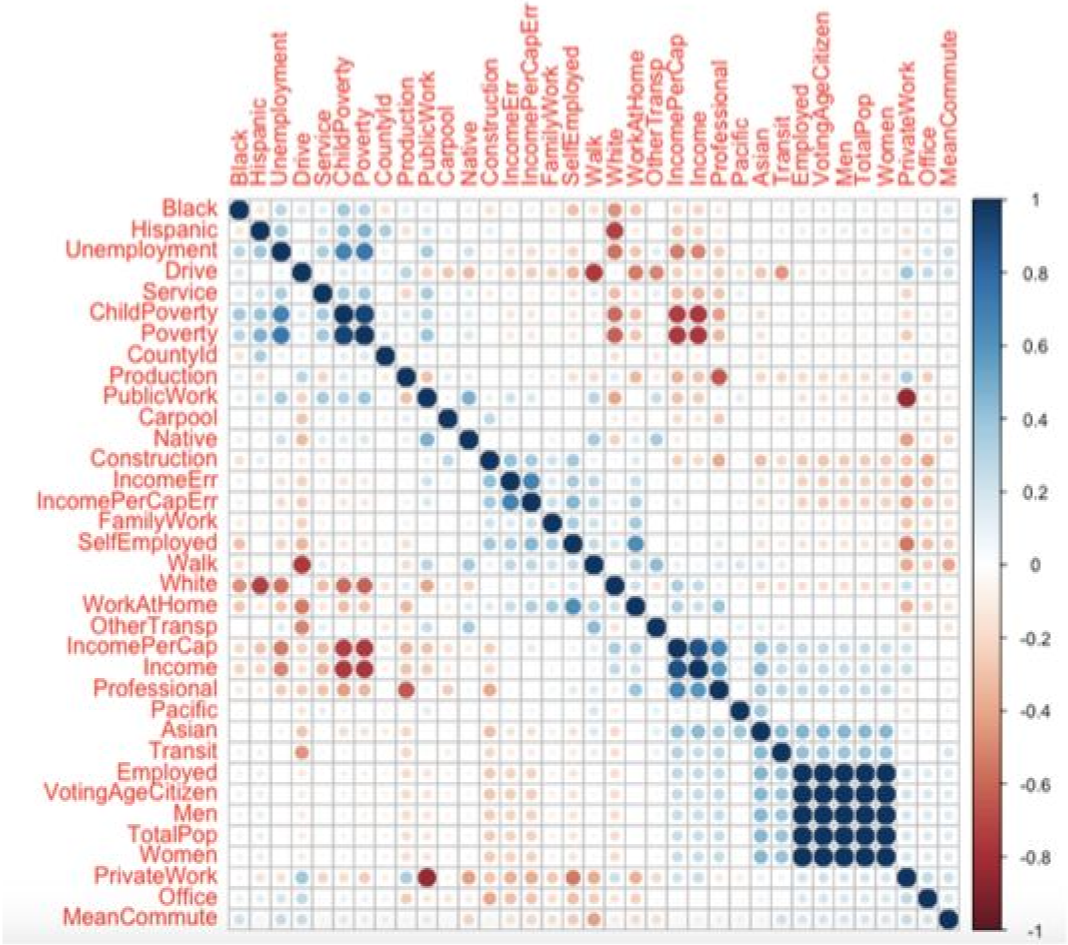
Individual Report Sam Song

In this project, I cooperated with my group partners for selecting the dataset and cleaning the data. I chose to focus on applying PCA and combine the result of PCA with the multiple regression model. Because of the low KMO value, the performance of PCA is not good. I use the result of PCA to build a linear regression model for predicting the performance of “unemployment”. The result shows that the income situation and job type have the most significant effect on the performance of unemployment. In this report, I apply multiple regression models, which is previous knowledge, and combine it with the concept that I learned in this class. I learned too much new technology about data analysis. This experiment is very useful in future work.

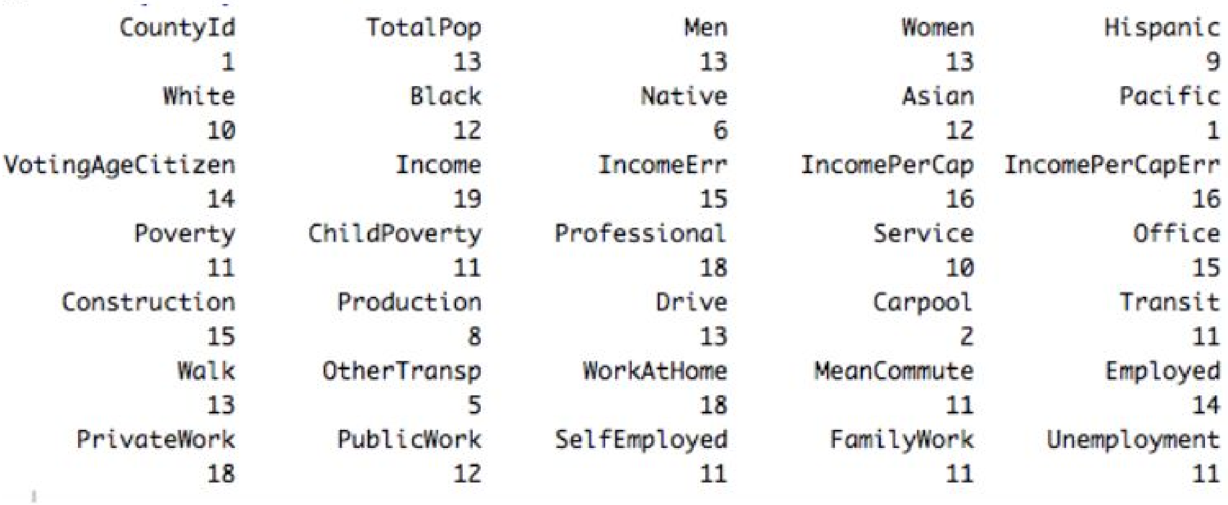
B1


Coun
TotalPo
Men
Women
Hispanic
White
Black
Native
Asian
Pacific
VotingAgeCitizen
Income
IncomeErr
IncomePerCap
IncomePerCapErr
Poverty
ChildPoverty
Professional
Service
Office
Construction
Production
Drive
Carpool
Transit
Walk
OtherTransp
WorkAtHome
MeanCommute
Employed
PrivateWork
PublicWork
SelfEmployed
FamilyWork
Unemployment
CountyId
TotalPop
Men
Women
Hispanic
White
Black
Native
Asian
Pacific
VotingAgeCitizen
Income
IncomeErr
IncomePerCap
ncomePerCapErr
Poverty
ChildPoverty
Professional
Service
Office
Construction
Production
Drive
Carpool
Transit
Walk
OtherTransp
WorkAtHome
MeanCommute
Employed
PrivateWork
PublicWork
SelfEmployed
FamilyWork
Unemployment
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

B2

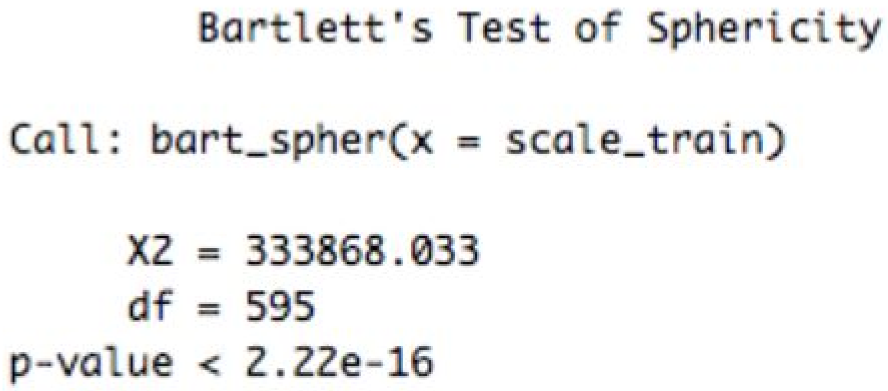


B3



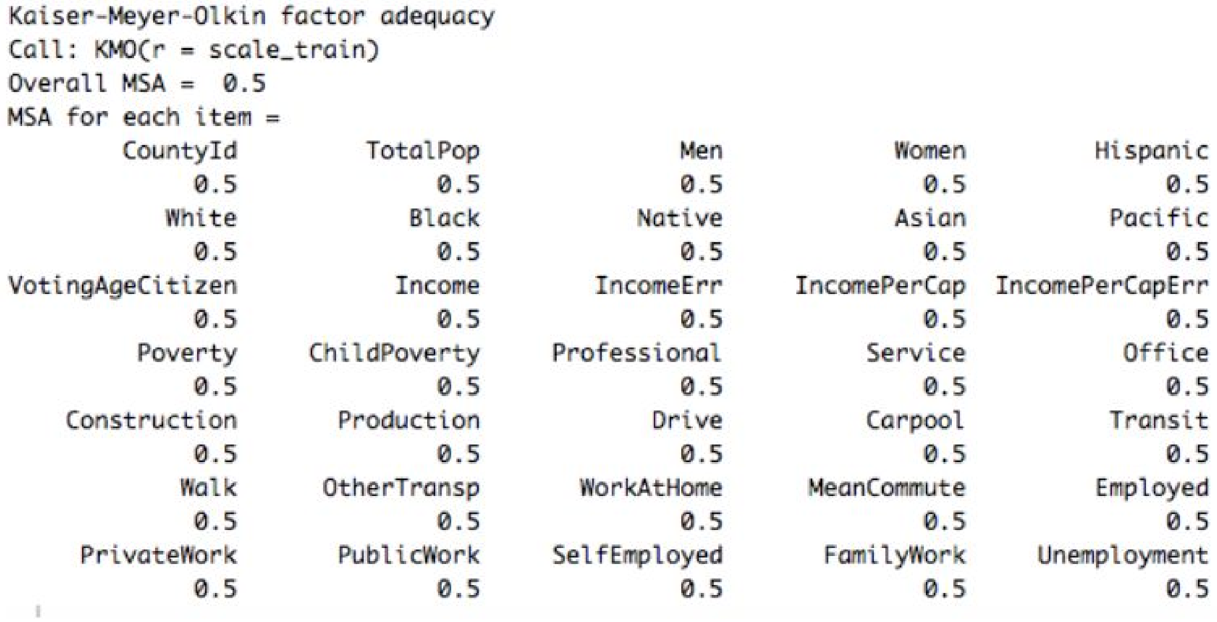
```
        CountyId         TotalPop              Men            Women         Hispanic
               1               13               13               13                9
           White            Black           Native            Asian          Pacific
              10               12                6               12                1
VotingAgeCitizen           Income        IncomeErr     IncomePerCap  IncomePerCapErr
              14               19               15               16               16
         Poverty     ChildPoverty     Professional          Service           Office
              11               11               18               10               15
    Construction       Production            Drive          Carpool          Transit
              15                8               13                2               11
            Walk      OtherTransp       WorkAtHome      MeanCommute         Employed
              13                5               18               11               14
     PrivateWork       PublicWork     SelfEmployed       FamilyWork     Unemployment
              18               12               11               11               11
```

B4



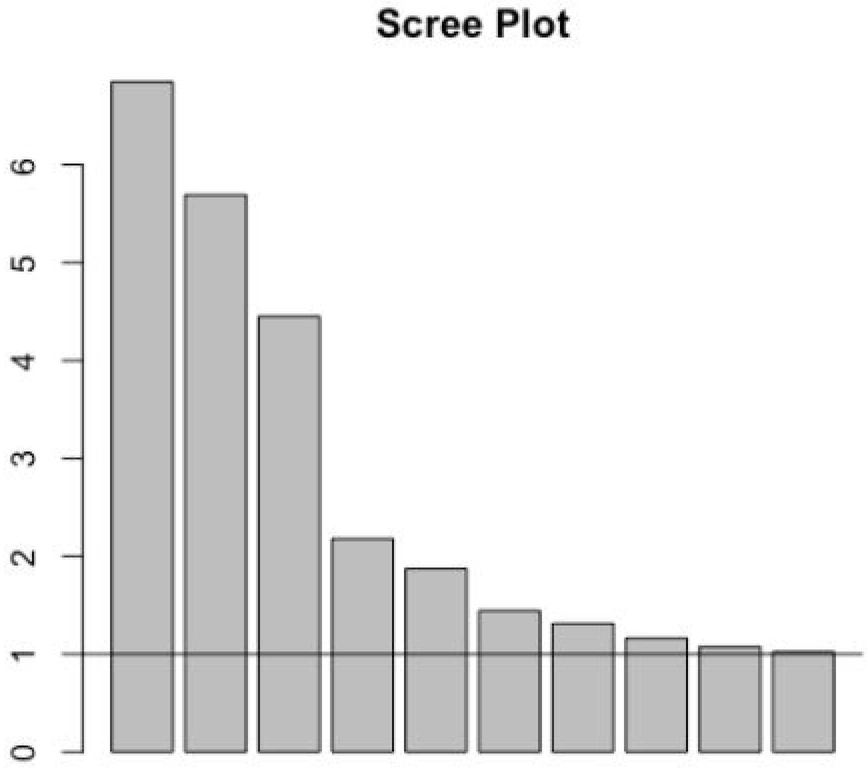
```
        Bartlett's Test of Sphericity

Call: bart_spher(x = scale_train)

     X2 = 333868.033
     df = 595
p-value < 2.22e-16
```

B5

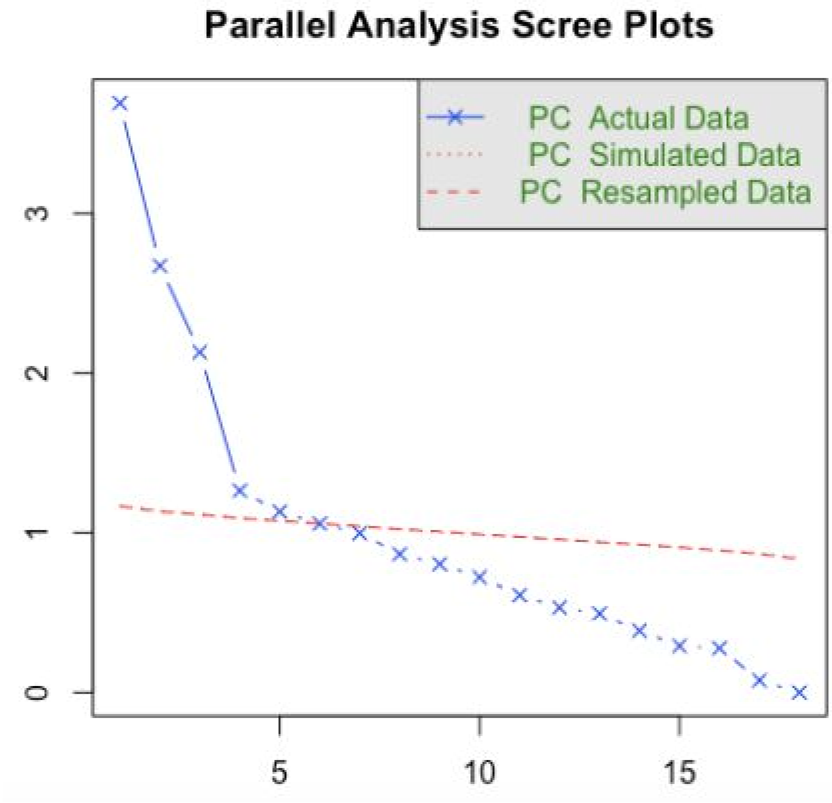


```
Kaiser-Meyer-Olkin factor adequacy
Call: KMO(r = scale_train)
Overall MSA =  0.5
MSA for each item = 
        CountyId         TotalPop              Men            Women         Hispanic 
             0.5              0.5              0.5              0.5              0.5 
           White            Black           Native            Asian          Pacific 
             0.5              0.5              0.5              0.5              0.5 
VotingAgeCitizen           Income        IncomeErr     IncomePerCap  IncomePerCapErr 
             0.5              0.5              0.5              0.5              0.5 
         Poverty     ChildPoverty     Professional          Service           Office 
             0.5              0.5              0.5              0.5              0.5 
    Construction       Production            Drive          Carpool          Transit 
             0.5              0.5              0.5              0.5              0.5 
            Walk       OtherTransp       WorkAtHome      MeanCommute         Employed 
             0.5              0.5              0.5              0.5              0.5 
     PrivateWork       PublicWork     SelfEmployed       FamilyWork     Unemployment 
             0.5              0.5              0.5              0.5              0.5 
```

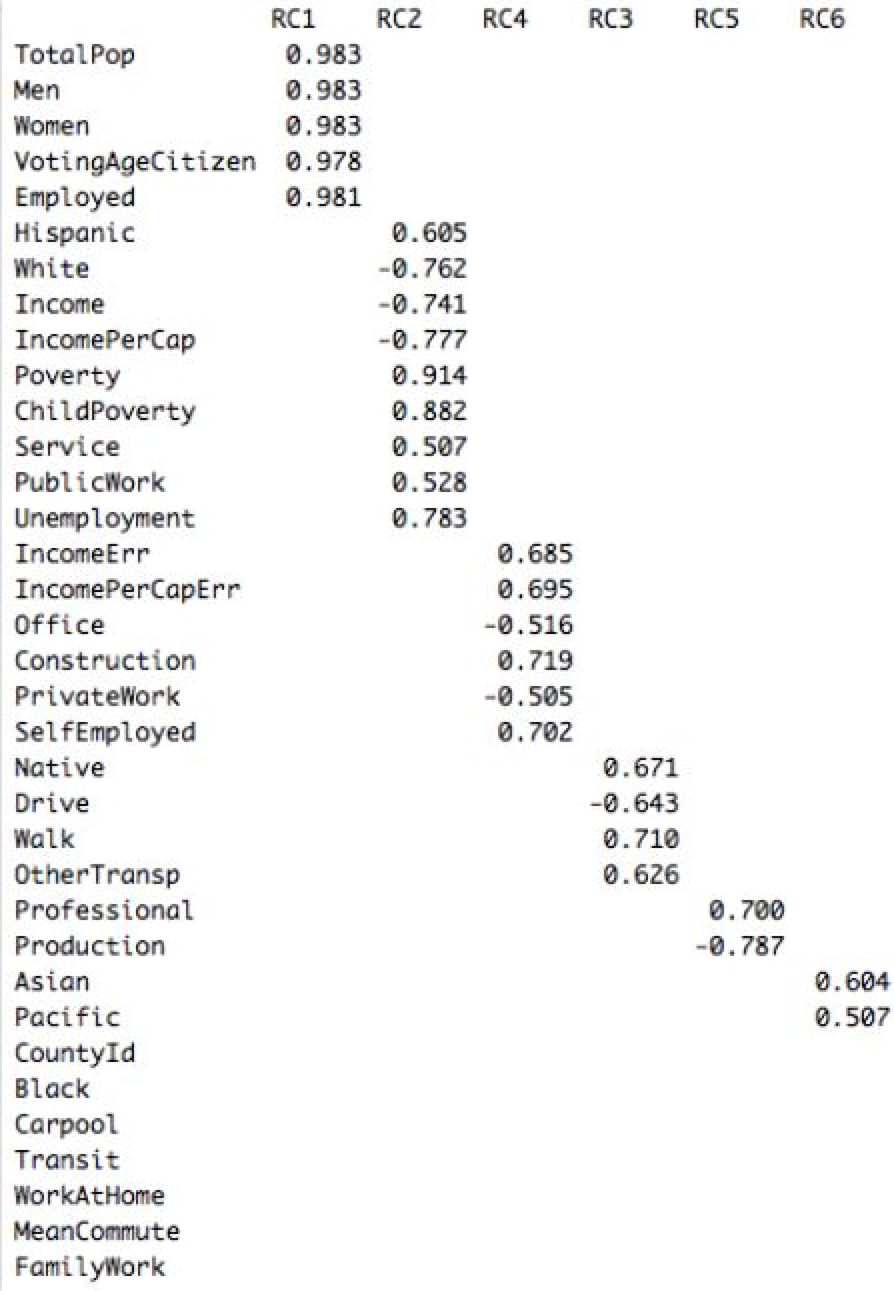
B6



B7

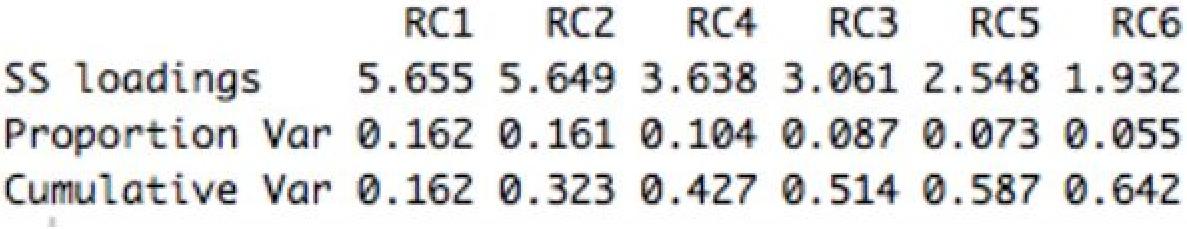

Parallel Analysis Scree Plots
PC Actual Data
PC Simulated Data
PC Resampled Data
0
1
2
3
5
10
15

B8



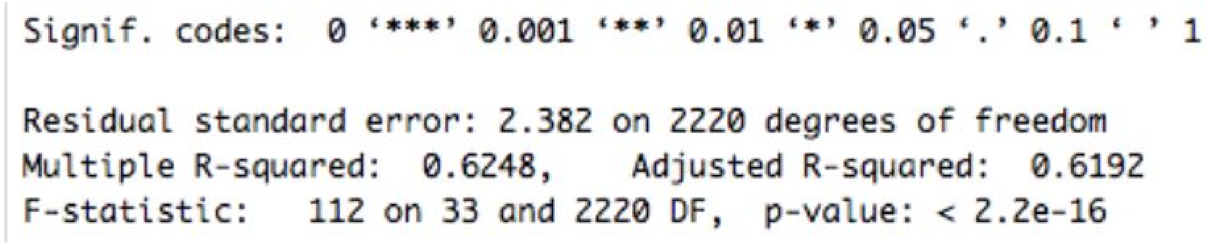
```
                 RC1    RC2    RC4    RC3    RC5    RC6
TotalPop          0.983
Men               0.983
Women             0.983
VotingAgeCitizen  0.978
Employed          0.981
Hispanic                 0.605
White                   -0.762
Income                  -0.741
IncomePerCap            -0.777
Poverty                  0.914
ChildPoverty             0.882
Service                  0.507
PublicWork               0.528
Unemployment             0.783
IncomeErr                       0.685
IncomePerCapErr                 0.695
Office                         -0.516
Construction                    0.719
PrivateWork                    -0.505
SelfEmployed                    0.702
Native                                 0.671
Drive                                 -0.643
Walk                                   0.710
OtherTransp                            0.626
Professional                                  0.700
Production                                   -0.787
Asian                                                0.604
Pacific                                              0.507
CountyId
Black
Carpool
Transit
WorkAtHome
MeanCommute
FamilyWork
```

B9



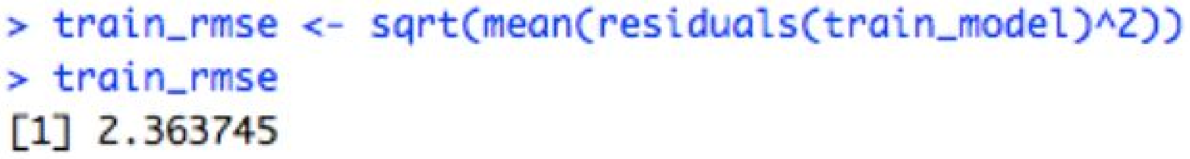
```
                 RC1   RC2   RC4   RC3   RC5   RC6
SS loadings    5.655 5.649 3.638 3.061 2.548 1.932
Proportion Var 0.162 0.161 0.104 0.087 0.073 0.055
Cumulative Var 0.162 0.323 0.427 0.514 0.587 0.642
```

B10



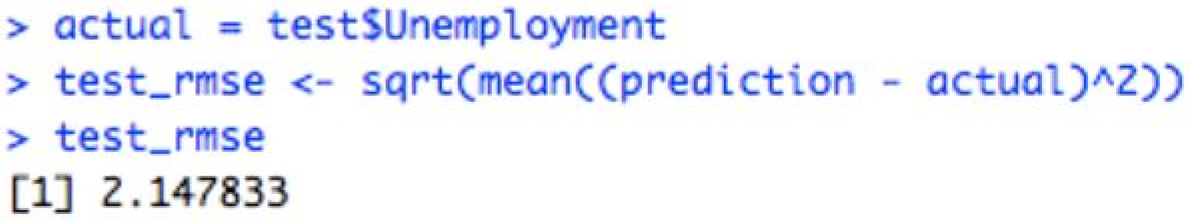
```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.382 on 2220 degrees of freedom
Multiple R-squared:  0.6248,	Adjusted R-squared:  0.6192 
F-statistic:   112 on 33 and 2220 DF,  p-value: < 2.2e-16
```

B11



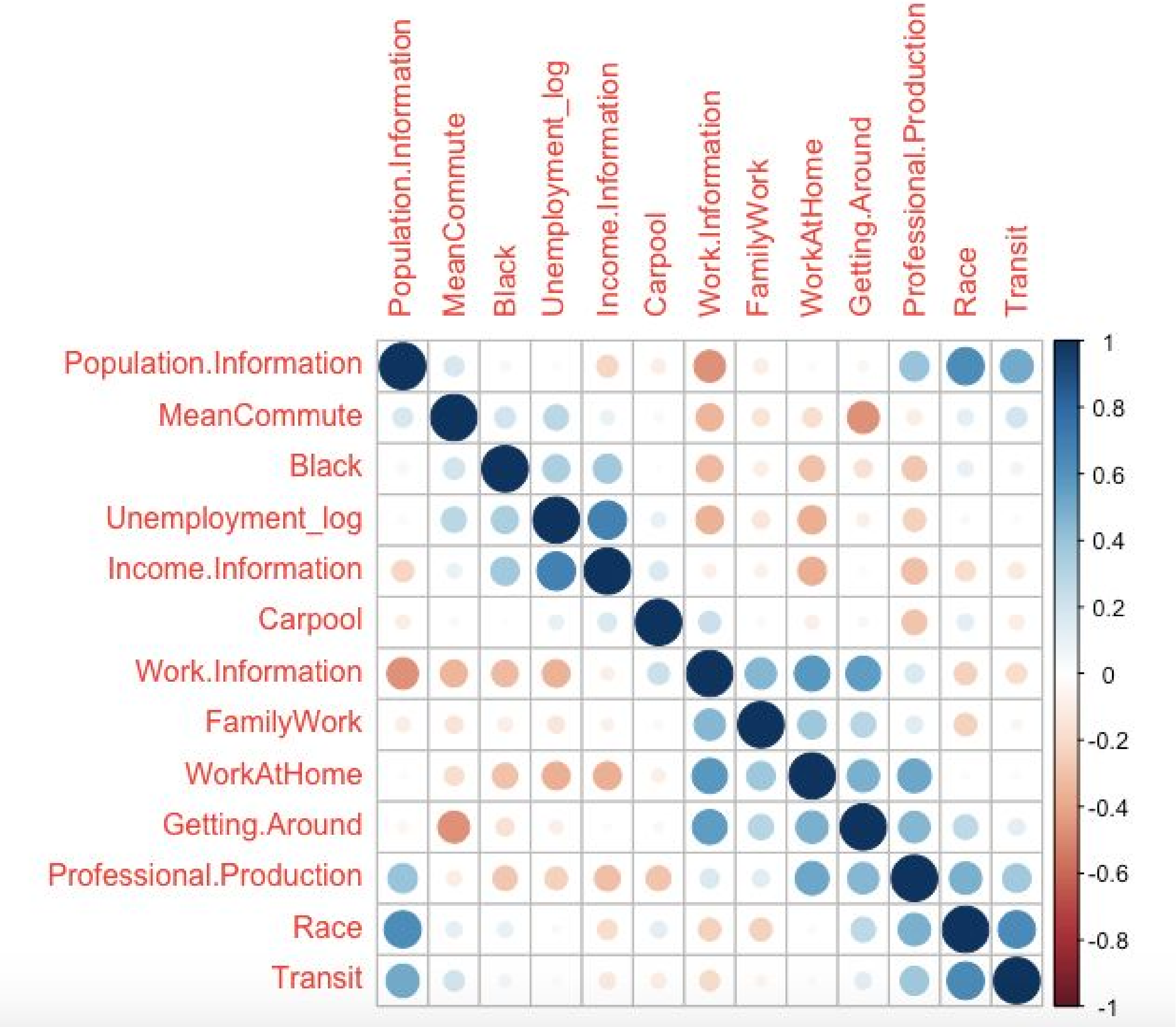
```
> train_rmse <- sqrt(mean(residuals(train_model)^2))
> train_rmse
[1] 2.363745
```

B12

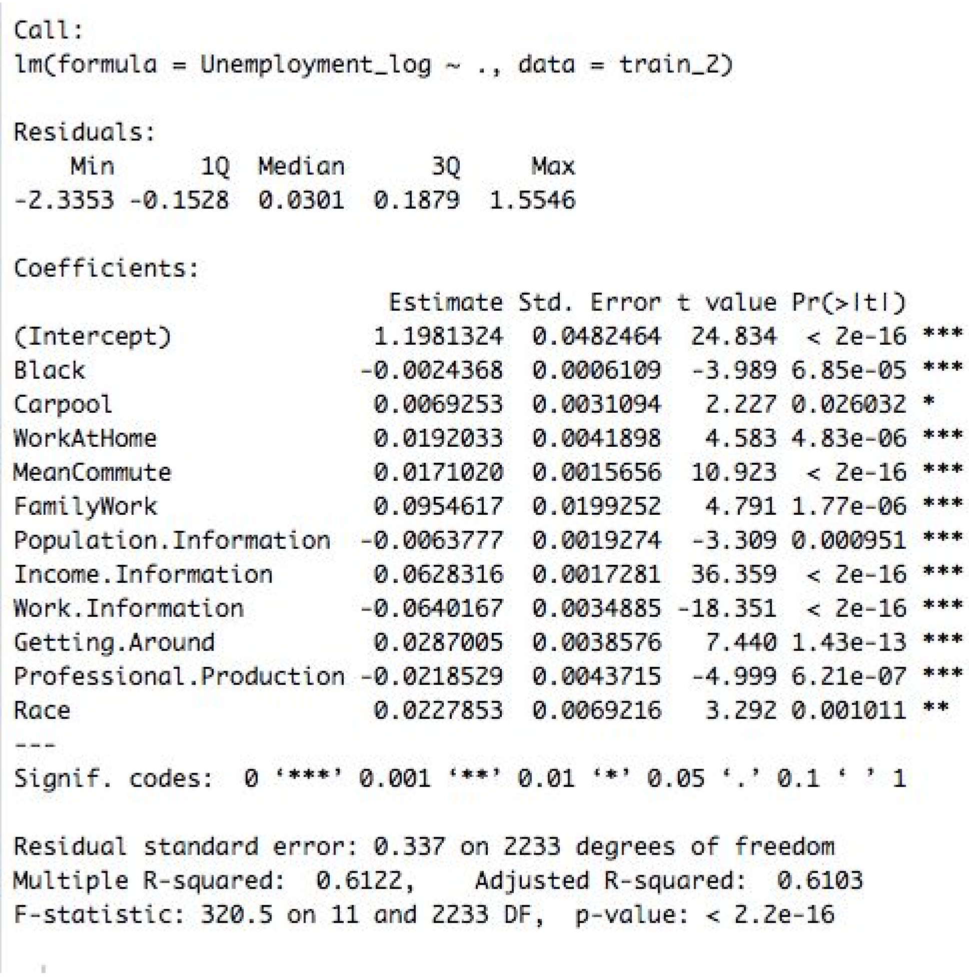


```
> actual = test$Unemployment
> test_rmse <- sqrt(mean((prediction - actual)^2))
> test_rmse
[1] 2.147833
```

B13

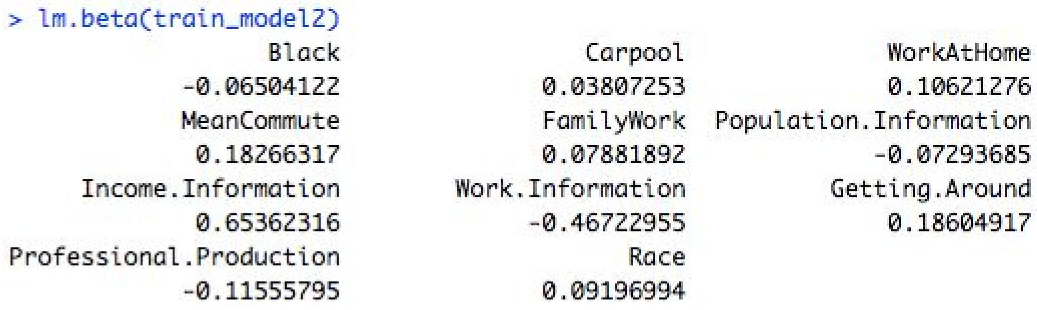

Population.Information
MeanCommute
Black
Unemployment_log
Income.Information
Carpool
Work.Information
FamilyWork
WorkAtHome
Getting.Around
Professional.Production
Race
Transit
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

B14



```
Call:
lm(formula = Unemployment_log ~ ., data = train_2)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.3353 -0.1528  0.0301  0.1879  1.5546 

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)    
(Intercept)             1.1981324  0.0482464  24.834  < 2e-16 ***
Black                  -0.0024368  0.0006109  -3.989 6.85e-05 ***
Carpool                 0.0069253  0.0031094   2.227 0.026032 *  
WorkAtHome              0.0192033  0.0041898   4.583 4.83e-06 ***
MeanCommute             0.0171020  0.0015656  10.923  < 2e-16 ***
FamilyWork              0.0954617  0.0199252   4.791 1.77e-06 ***
Population.Information -0.0063777  0.0019274  -3.309 0.000951 ***
Income.Information      0.0628316  0.0017281  36.359  < 2e-16 ***
Work.Information       -0.0640167  0.0034885 -18.351  < 2e-16 ***
Getting.Around          0.0287005  0.0038576   7.440 1.43e-13 ***
Professional.Production -0.0218529  0.0043715  -4.999 6.21e-07 ***
Race                    0.0227853  0.0069216   3.292 0.001011 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.337 on 2233 degrees of freedom
Multiple R-squared:  0.6122,	Adjusted R-squared:  0.6103 
F-statistic: 320.5 on 11 and 2233 DF,  p-value: < 2.2e-16
```

B15



```
> lm.beta(train_model2)
                  Black                 Carpool              WorkAtHome 
            -0.06504122              0.03807253              0.10621276 
            MeanCommute              FamilyWork  Population.Information 
             0.18266317              0.07881892             -0.07293685 
     Income.Information        Work.Information          Getting.Around 
             0.65362316             -0.46722955              0.18604917 
Professional.Production                    Race 
            -0.11555795              0.09196994 
```

CODE:
library(corrplot)
library(psych)

#read file
census = read.csv("acs2017.csv")
head(census)
census_2 = as.data.frame(census[,-c(2,3)])

```
head(census_2)

#scaling the data
scale_train = scale(census_2)

# Adequecy Test
cor.data = cor(scale_train)
cor.data
#corrplot 1 with value
round(corrplot(cor(cor.data,method="spearman"), method = "number",
          type = "lower"), 2)
#corrplot 2 with circle graph
round(corrplot(cor(cor.data,method="spearman"), method = "circle",
          type = "lower"), 2)
#corrplot 3 with aoe order
corrplot(cor.data, order="AOE")

# Run a correlation test to see how correlated the variables are.  Which correlations are
significant
CorrTest = corr.test(cor.data, adjust="none")
round(CorrTest$p, 2)

M = CorrTest$p
M

# Now, for each element, see if it is < .01 (or whatever significance) and set the entry to
# true = significant or false
MTest = ifelse(M < .01, T, F)
MTest

# Now lets see how many significant correlations there are for each variable.  We can
do
# this by summing the columns of the matrix
colSums(MTest) - 1  # We have to subtract 1 for the diagonal elements (self-correlation)

install.packages('REdaS')
install.packages('grid')
library(REdaS)
library(grid)
#check Bartlett's Test of Sphericity p-value
bart_spher(scale_train)  #p-value < 2.22e-16
#kmo test
library(psych)
KMO(scale_train) #Overall MSA =  0.5
```

```
# PCA
p = prcomp(scale_train, center = T)
summary(p)
plot(p, main = 'Scree Plot')
abline(1,0)
fa.parallel(train,fa="pc")

# Rotation and Interpretation
p1 = principal(scale_train, rotate = 'varimax', nfactors = 6, scores = TRUE)
print(p1$loadings, cutoff = 0.5, sort = T)

score = p1$scores
score

colnames(score) <- c("Population Information", "Income Information", "Work
Information",
                 "Getting Around", "Professional&Production", "Race")
head(score)
install.packages('xlsx')
library(xlsx)

write.csv(score, "6pc_scores.csv")

census = read.csv("acs2017_Unem.csv")
head(census)

hist(census$Unemployment)
hist(log(census$Unemployment))

install.packages("rcompanion")
library(rcompanion)
library(MASS)
library(car)

Unemployment_log=log(census$Unemployment)
census$Unemployment_log <- Unemployment_log
census
head(census)
census_2 = as.data.frame(census[,-c(7)])
head(census_2)
census_3 <- census_2[-c(2,6,18,102,657,1235,1635,1623,2136,2528,2722,3220),]
census_3
```

```
#split train and test

## 70% of the sample size
smp_size <- floor(0.7 * nrow(census_3))
## set the seed to make your partition reproducible
set.seed(321)
train_ind <- sample(seq_len(nrow(census_3)), size = smp_size)
train <- census_3[train_ind, ]
test <- census_3[-train_ind, ]


#RMSE of train and test (Unemployment)
train_model = lm(Unemployment_log ~ ., data =train)
summary(train_model)
train_rmse <- sqrt(mean(residuals(train_model)^2))
train_rmse
prediction <- predict(train_model, test)
actual = test$Unemployment_log
test_rmse <- sqrt(mean((prediction - actual)^2))
test_rmse


# Adequecy Test
cor.data = cor(train)
cor.data
#corrplot 1 with value
round(corrplot(cor(cor.data,method="spearman"), method = "number",
            type = "lower"), 2)
#corrplot 2 with circle graph
round(corrplot(cor(cor.data,method="spearman"), method = "circle",
            type = "lower"), 2)
#corrplot 3 with aoe order
corrplot(cor.data, order="AOE")

#check multicollinearity by VIF
vif(train_model)


#forward
install.packages("leaps")
null=lm(Unemployment_log ~ 1, data=train)
null
full=lm(Unemployment_log ~., data=train)
summary(full)
forward=step(null, scope=list(lower=null,upper=full), direction='forward', trace=F)
```

```
summary(forward)

#reset the training and testing data
train_2 = as.data.frame(train[,-c(3)])
head(train_2)
test_2 = as.data.frame(test[,-c(3)])
head(test_2)

#RMSE of train and test 2(Unemployment)
train_model2 = lm(Unemployment_log ~ ., data =train_2)
summary(train_model2)
train_rmse2 <- sqrt(mean(residuals(train_model2)^2))
train_rmse2
prediction <- predict(train_model2, test)
actual = test$Unemployment_log
test_rmse2 <- sqrt(mean((prediction - actual)^2))
test_rmse2

#new model (SelfEmployed)
train_model2 = lm(Unemployment_log ~ ., data =train_2)
summary(train_model2)

#check standardized estimate
library(QuantPsyc)
lm.beta(train_model2)
```

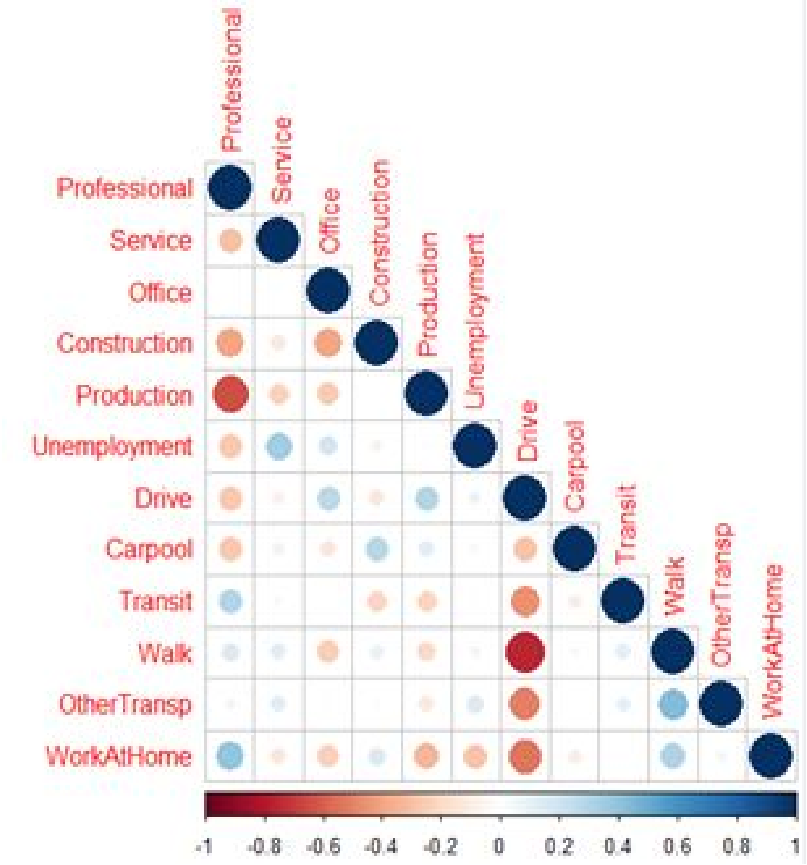
Appendix C

Individual Report Adam Frink

This semester I worked with my teammates to select a dataset and decide on which lines of investigation would be interesting to model. I chose to focus on the job category vs commute category line of investigation. My goal was to use methods learned after the midterm to analyze this potential relationship. Since the topics were recently introduced it made analysis more of a challenge, but helped to cement the material. I initially attempted correspondence analysis and Cluster analysis. However, these methods weren't appropriate for my dataset which is aggregated per US county and had 1 categorical variable 'State'. Canonical correlation, the final method covered in class, seemed to be just what I was looking for to analyze the relationship between two groups of data points, which was the question I was trying to answer. I found some worthwhile relationships here, but had continued problems running the wilks function. Since there were issues with the canonical correlation technique I also chose to create a categorical variable to enable investigation using LDA.

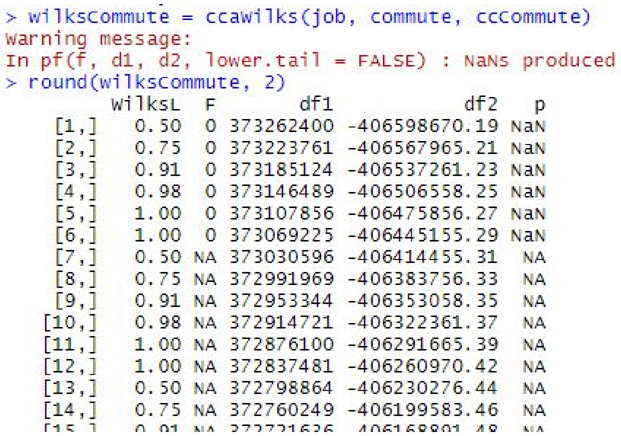
Canonical Correlation

Original Data Assessment Corrplot Job vs Commute



Issues With Wilks

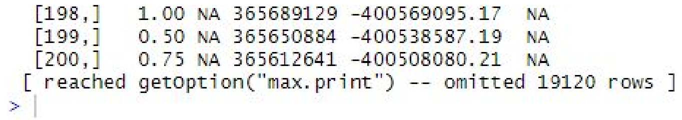
My wilks function should only produce 6 rows since i have a 6X6 matrix of x y variables and therefore 6 variants. However, the wilks function when Irun it with my dataset produces 200 rows and then cuts out.



```
> wilksCommute = ccaWilks(job, commute, ccCommute)
Warning message:
In pf(f, d1, d2, lower.tail = FALSE) : NaNs produced
> round(wilksCommute, 2)
         WilksL  F       df1           df2   p
    [1,]   0.50  0 373262400 -406598670.19 NaN
    [2,]   0.75  0 373223761 -406567965.21 NaN
    [3,]   0.91  0 373185124 -406537261.23 NaN
    [4,]   0.98  0 373146489 -406506558.25 NaN
    [5,]   1.00  0 373107856 -406475856.27 NaN
    [6,]   1.00  0 373069225 -406445155.29 NaN
    [7,]   0.50 NA 373030596 -406414455.31  NA
    [8,]   0.75 NA 372991969 -406383756.33  NA
    [9,]   0.91 NA 372953344 -406353058.35  NA
   [10,]   0.98 NA 372914721 -406322361.37  NA
   [11,]   1.00 NA 372876100 -406291665.39  NA
   [12,]   1.00 NA 372837481 -406260970.42  NA
   [13,]   0.50 NA 372798864 -406230276.44  NA
   [14,]   0.75 NA 372760249 -406199583.46  NA
   [15,]   0.91 NA 372721636 -406168891.48  NA
```

.

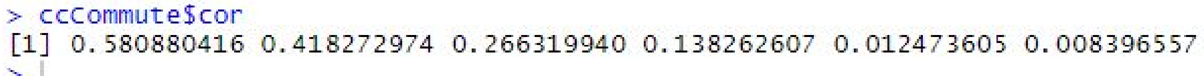
.



```
  [198,]   1.00 NA 365689129 -400569095.17   NA
  [199,]   0.50 NA 365650884 -400538587.19   NA
  [200,]   0.75 NA 365612641 -400508080.21   NA
 [ reached getOption("max.print") -- omitted 19120 rows ]
> 
```

My canonical correlation variable has the correct number of variants 6 and when Irun my wilks function with the iris data set it works correctly. I spent some time trying to figure it out and couldt not find a solution.

```
# Next, we run the canonical correlation
        ccCommute = cc(job, commute)
        ccCommute
        ccCommute$cor
```



```
> ccCommute$cor
[1] 0.580880416 0.418272974 0.266319940 0.138262607 0.012473605 0.008396557
> 
```

Canonical Correlation R Code

```
library(sqldf)
library(corrplot)


setwd("C:/Data/DCS 424/Final Project")

####################
# pull in columns want to compare (in this case job types and commute types)
####################
census = read.csv("acs2017_county_data.csv")
census = as.matrix(census[,c(20,21,22,23,24,37,25,26,27,28,29,30)]) # job types and commute types
head(census)
# TotalPop
# Income
# IncomePerCap
# Professional
# Service
# Office
# Construction
# Production
# FamilyWork
# Unemployment
#
# Drive
# Carpool
# Transit
# Walk
# OtherTransp

#####################
# quick corr plot
#####################

cor = cor(census)
corrplot(cor, type="lower")


#####################
# Seperating the data into job and commute
#####################
job = census[,1:6]  # the tpe of work
commute = census[,7:12]   # commute type
```

```
###################################################################
# Now, lets do some computation
###################################################################

# The CCA library has more extensive functionality
library(CCA)

# First invesitgate the combined correlation matrix, and test the
# cross correlations.  To do this we use the "matcor" function which
# computes correlation matrices between two datasets
c = matcor(job, commute)
c
img.matcor(c, type = 2)


# Then we pull out the upper right block of correlations that compares
# the job and commute  --gives just the diagnol we're interested in
cCross = c$XYcor[1:6, 7:12]
round(cCross, 2)

# Now, run a correlation test
library(psych)
p = corr.p(cCross, nrow(job))
p

# Next, we run the canonlical correlation
ccCommute = cc(job, commute)
ccCommute
ccCommute$cor

# This gives us the cannonical correlates, but no significance tests
c_sig = cancor(job, commute)
c_sig

# First, let's test the model for significance.  We are quite sure that
# We will find some correlation here because our matrix above showed
# Significance, but the Wilks test confirms this

ccaWilks = function(set1, set2, cca)
{
 ev = ((1 - cca$cor^2))
 ev

 n = dim(set1)[1]
 p = length(set1)
 q = length(set2)
 k = min(p, q)
 m = n - 3/2 - (p + q)/2
 m

 w = rev(cumprod(rev(ev)))

 # initialize
 d1 = d2 = f = vector("numeric", k)

 for (i in 1:k)
 {
  s = sqrt((p^2 * q^2 - 4)/(p^2 + q^2 - 5))
  si = 1/s
  d1[i] = p * q
  d2[i] = m * s - p * q/2 + 1
  r = (1 - w[i]^si)/w[i]^si
  f[i] = r * d2[i]/d1[i]
  p = p - 1
  q = q - 1
  }

  pv = pf(f, d1, d2, lower.tail = FALSE)
  dmat = cbind(WilksL = w, F = f, df1 = d1, df2 = d2, p = pv)
}

wilksCommute = ccaWilks(job, commute, ccCommute)
round(wilksCommute, 2)

# To understand the canonical correlates, we look at the raw coefficients
# These can be interpreted exactly like components in PCA except that we
# have two sets of them.  Notice, however, that it is the RELATIVE
# contributions of each variable that are important, not the absolute
# size of the contribution
names(ccCommute)
round(-ccCommute$xcoef, 3)  # Since the first column is negative, we negate these
round(-ccCommute$ycoef, 3)  # Notice that this makes no difference in the relationship between them
ccCommute


# To help us better understand the components, we look at the correlations
# between each of the variates and the original variables that make them up.
# This creates two "loadings" matrices very much like the loadings matrix
# of PCA
#
# You will notice that this gives a different picture of the relationship between the x's and
# their variates.  Math was a lower "coefficient" but has the highest correlation with the
# variate.  This means in a real sense that math contributes most highly to this variate.
# The reason for this difference is in the size of the contributions and in how the other variables
# also contribute.  If Math swings more, it might contribute more even though its coefficient is
# smaller
loadingsCommute = comput(job, commute, ccCommute)
ls(loadingsCommute)
round(-loadingsCommute$corr.X.xscores, 2)
round(-loadingsCommute$corr.Y.yscores, 2)

# Let's plot the first two variates against each other.  We do this by looking at the scores.
plot(loadingsCommute$xscores[,2], loadingsCommute$yscores[,2])
cor(loadingsCommute$xscores[,1], loadingsCommute$yscores[,1])

# Last we look at the relationship between the variables and the correlates from the other dataset
# This gives us a better view of how the variables from one set relate to the other correlate and
# how we might predict one from the other
loadingsCommute$corr.X.yscores   # How do the y-variates depend on the x-variables.  Most important for prediction
loadingsCommute$corr.Y.xscores

# A basic visualization of the cannonical correlation
plt.cc(ccCommute)
```

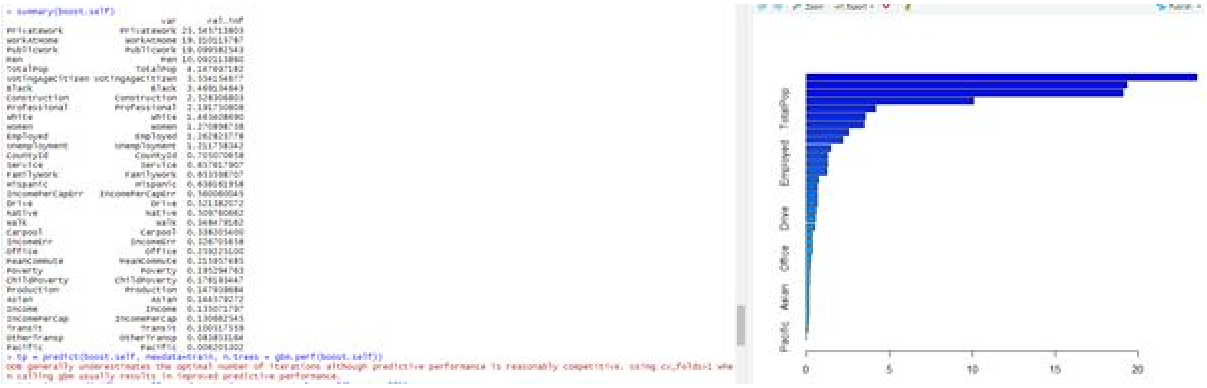
LDA Code

```
#####################
#SQL ZONE
#####################
census = read.csv("acs2017_county_data.csv")

census = data.frame(census)

census2 =
  sqldf('SELECT *

        /*Add REGIONS*/
         ,CASE WHEN State IN ("California","Hawaii", "Nevada") THEN "W"
```

```
     WHEN State IN ("Illinois", "Indiana", "Michigan", "Minnesota", "Ohio", "Wisconsin") THEN "G_Lake"
     WHEN State IN ("Delaware", "District of Columbia", "Kentucky", "Maryland", "North Carolina", "Tennessee", "Virginia", "West
Virginia") THEN "Mid_Sou"
     WHEN State IN ("Iowa", "Kansas", "Missouri", "Nebraska", "North Dakota", "South Dakota") THEN "Mid_W"
     WHEN State IN ("Arizona", "Colorado", "Idaho", "Montana", "New Mexico", "Utah", "Wyoming")THEN "Mnt_W"
     WHEN State IN ("Connecticut", "Maine", "Massachusetts", "New Hampshire", "Rhode Island", "Vermont") THEN "New_Eng"
     WHEN State IN ("New Jersey", "New York", "Pennsylvania") THEN "N_East"
     WHEN State IN ("Oregon", "Washington", "Alaska") THEN "NW"
     WHEN State IN ("Arkansas", "Louisiana", "Oklahoma", "Texas") THEN "South_C"
     WHEN State IN ("Alabama", "Florida", "Georgia", "Mississippi", "South Carolina", "Puerto Rico") THEN "South_E" END AS
region

    /*Add REGION CODE AS INT*/
     ,CASE WHEN State IN ("California","Hawaii", "Nevada") THEN 1
     WHEN State IN ("Illinois", "Indiana", "Michigan", "Minnesota", "Ohio", "Wisconsin") THEN 2
     WHEN State IN ("Delaware", "District of Columbia", "Kentucky", "Maryland", "North Carolina", "Tennessee", "Virginia", "West
Virginia") THEN 3
     WHEN State IN ("Iowa", "Kansas", "Missouri", "Nebraska", "North Dakota", "South Dakota") THEN 4
     WHEN State IN ("Arizona", "Colorado", "Idaho", "Montana", "New Mexico", "Utah", "Wyoming")THEN 5
     WHEN State IN ("Connecticut", "Maine", "Massachusetts", "New Hampshire", "Rhode Island", "Vermont") THEN 6
     WHEN State IN ("New Jersey", "New York", "Pennsylvania") THEN 7
     WHEN State IN ("Oregon", "Washington", "Alaska") THEN 8
     WHEN State IN ("Arkansas", "Louisiana", "Oklahoma", "Texas") THEN 9
     WHEN State IN ("Alabama", "Florida", "Georgia", "Mississippi", "South Carolina", "Puerto Rico") THEN 10 END AS
region_code


    /*Add CITY SIZE*/
     ,CASE WHEN TotalPop < 11214 THEN "x_small"
     WHEN TotalPop >= 11214 AND TotalPop < 25848 THEN "small"
     WHEN TotalPop >= 25848 AND TotalPop < 66609 THEN "medium"
     WHEN TotalPop > 66608 THEN "large" END AS size_category


     /*Add CITY SIZE*/
     ,CASE WHEN TotalPop < 11214 THEN 1
     WHEN TotalPop >= 11214 AND TotalPop < 25848 THEN 2
     WHEN TotalPop >= 25848 AND TotalPop < 66609 THEN 3
     WHEN TotalPop > 66608 THEN 4 END AS size_category_code
     FROM census'
 )


        #
        # sql_table = sqldf('SELECT AVG(TotalPop) AS avg_TotalPop
        #      ,AVG(Income)          AS avg_Income
        #      ,AVG(IncomePerCap)    AS avg_IncomePerCap
        #      ,AVG(Professional)    AS avg_Professional
        #      ,AVG(Service)         AS avg_Service
        #      ,AVG(Office)          AS avg_Office
        #      ,AVG(Construction)    AS avg_Construction
        #      ,AVG(Production)      AS avg_Production
        #      ,AVG(Drive)           AS avg_Drive
        #      ,AVG(Carpool)         AS avg_Carpool
        #      ,AVG(Transit)         AS avg_Transit
        #      ,AVG(Walk)            AS avg_Walk
        #      ,AVG(OtherTransp)     AS avg_OtherTransp
        #      ,AVG(Unemployment)    AS avg_Unemployment
        #
        #      FROM source'
        # )

        # avg_TotalPop     --100,786.1
        # avg_Income       --48,994.97
        # avg_IncomePerCap --25657
        # avg_Professional 31.47981
        # avg_Service  18.21429
        # avg_Office 21.87894
        # avg_Construction  12.59236
        # avg_Production  15.83575
        # avg_Drive  79.63096
        # avg_Carpool  9.851646
        # avg_Transit  0.9389752
        # avg_Walk  3.244472
        # avg_OtherTransp  1.598696
        # avg_FamilyWork  0.2788199
        # avg_Unemployment   6.66559


####################################################
#  LDA
####################################################

library(car)
install.packages("rattle")
head(census2)

census3 = as.data.frame(census2[,c(2,20,21,22,23,24,37, 25,26,27,28,29,30, 38,39,40,41)]) # job types and commute types


summary(census3$region)
summary(census3$state)

#BASED ON REGION
##########################

#plot all
plot(census3[2:12], pch=16, col=census3$region_code)  # Not much separation

#plot job type
plot(census3[2:7], pch=16, col=census3$region_code)  # more separation, but overlap

#plot commute type
plot(census3[8:12], pch=16, col=census3$region_code)  # more separation, but overlap


library(MASS)

# Try an initial lda on everything
censuslda = lda(region ~ ., data=census3[2:13])

# Try on job type
censuslda_job = lda(region ~ Professional + Service+ Office+ Construction+ Production+ Unemployment, data=census3)
  censuslda_job1 = lda(region ~ Production, data=census3)


# Try on commute type
censuslda_commute = lda(region ~ Drive+ Carpool+ Transit+ Walk+ OtherTransp + WorkAtHome, data=census3)


# We can use "predict" just like "lm".  Note here that we are predicting on the
# training set!
censuslda.values = predict(censuslda)
censuslda_job.values = predict(censuslda_job)
censuslda_commute.values = predict(censuslda_commute)



censuslda.values$x  # The scores are stored in the x-parameter
censuslda_job.values$x
censuslda_commute.values$x
```

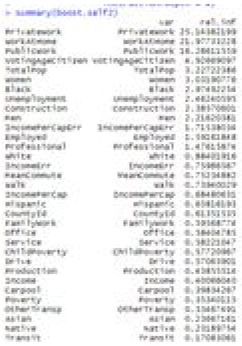
```
##############
#Job
##############
# Now, let's predict our training data
p2 = predict(censuslda_job, newdata=census3[,c(2:7)])
par(mar=c(1,1,1,1))
ldahist(p2$x[, 1], g=census3$region)
ldahist(p2$x[, 2], g=census3$region)
ldahist(p2$x[, 3], g=census3$region)
ldahist(p2$x[, 4], g=census3$region)
ldahist(p2$x[, 5], g=census3$region)
ldahist(p2$x[, 6], g=census3$region)


plot(p2$x[, 1:2], col=census3$region_code, pch=16)


##############
#Commute
##############
# Now, let's predict our training data
p3 = predict(censuslda_commute, newdata=census3[,c(8:13)])
par(mar=c(1,1,1,1))
ldahist(p3$x[, 1], g=census3$region)
ldahist(p3$x[, 2], g=census3$region)  # Separates diffuse nebulae with open clusters
ldahist(p3$x[, 3], g=census3$region)
ldahist(p3$x[, 4], g=census3$region)
ldahist(p3$x[, 5], g=census3$region)
ldahist(p3$x[, 6], g=census3$region)


plot(p3$x[, 1:2], col=census3$region_code, pch=16)
```

Appendix D



```
> summary(boost.self)
                                var      rel.inf
PrivateWork             PrivateWork 21.565713803
WorkAtHome               WorkAtHome 19.350113787
PublicWork               PublicWork 19.099582543
Men                             Men 10.092113860
TotalPop                   TotalPop  4.187007182
VotingAgeCitizen VotingAgeCitizen  3.556156877
Black                         Black  3.469534843
Construction           Construction  3.328306803
Professional           Professional  2.191750808
White                         White  1.445608690
Women                         Women  1.270898738
Employed                   Employed  1.262823778
Unemployment           Unemployment  1.251758342
CountyId                   CountyId  0.705070858
Service                     Service  0.657017907
FamilyWork               FamilyWork  0.653108707
Hispanic                   Hispanic  0.638161958
IncomePerCapErr     IncomePerCapErr  0.560060045
Drive                         Drive  0.521182072
Native                       Native  0.509760062
Walk                           Walk  0.368479162
Carpool                     Carpool  0.338305400
IncomeErr                 IncomeErr  0.326705658
Office                       Office  0.259271100
MeanCommute             MeanCommute  0.215957685
Poverty                     Poverty  0.185294763
ChildPoverty           ChildPoverty  0.179183447
Production               Production  0.147939684
Asian                         Asian  0.144179272
Income                       Income  0.135071797
IncomePerCap           IncomePerCap  0.130682543
Transit                     Transit  0.100317559
OtherTransp             OtherTransp  0.083853164
Pacific                     Pacific  0.006201302
> tp = predict(boost.self, newdata=train, n.trees = gbm.perf(boost.self))
OOB generally underestimates the optimal number of iterations although predictive performance is reasonably competitive. Using cv_folds>1 whe
n calling gbm usually results in improved predictive performance.
```

Fig.D-1 Output of gbm



```
> summary(boost.self2)
                                  var    rel.inf
PrivateWork             PrivateWork 25.14382199
WorkAtHome               WorkAtHome 21.97731228
PublicWork               PublicWork 16.28611319
VotingAgeCitizen   VotingAgeCitizen  4.92669097
TotalPop                   TotalPop  3.22722388
Women                         Women  3.03190778
Black                         Black  2.87482216
Unemployment           Unemployment  2.48240595
Construction           Construction  2.38570801
Men                             Men  2.21620581
IncomePerCapErr     IncomePerCapErr  1.71538034
Employed                   Employed  1.59281848
Professional           Professional  1.47615879
White                         White  0.88401916
IncomeErr                 IncomeErr  0.75986567
MeanCommute             MeanCommute  0.75234882
Walk                           Walk  0.73940029
IncomePerCap           IncomePerCap  0.68480631
Hispanic                   Hispanic  0.65814193
CountyId                   CountyId  0.61351319
FamilyWork               FamilyWork  0.39568774
Office                       Office  0.38604785
Service                     Service  0.38221847
ChildPoverty           ChildPoverty  0.37720967
Drive                         Drive  0.37063901
Production               Production  0.43855514
Income                       Income  0.40066640
Carpool                     Carpool  0.39834267
Poverty                     Poverty  0.35340513
OtherTransp             OtherTransp  0.33467691
Asian                         Asian  0.23067181
Native                       Native  0.23189754
Transit                     Transit  0.17083060
```

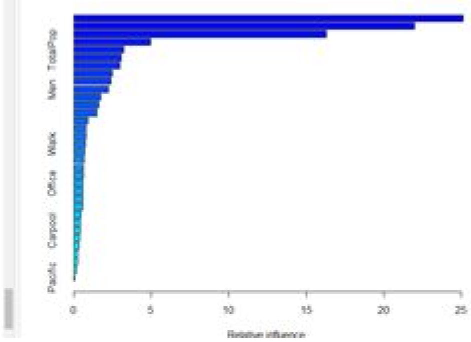


Fig.D-2 Output of test

Code:

```
# read data American Census 2017

census = read.csv("acs2017.csv")
head(census)

census_2 = as.data.frame(census[,-c(2,3)])
head(census_2)


install.packages("gbm")
library(gbm)
set.seed(86)

boost.self = gbm(SelfEmployed ~ ., data=census_2, distribution="gaussian", n.trees = 10000,
               shrinkage = 0.01, interaction.depth = 4)
```