IT 270 Group Presentation 2016 & 2020 Presidential Election Analysis

Group Members: Jingyi Wang, Omar Qureshi, Ying Zhang, Ohoud Albabtin, Adam Manson, Siraj

Introduction

- The 2016 & 2020 Presidential Election data set contains data about United States county voting statistics such as demographics, county information, income, etc, as well as covid cases and deaths.
- We'd like to investigate the following question in this analysis:
    - What factors influenced these presidential elections?
- The Data Set contains approximately 5,000 observations and 51 variables

Analysis Techniques

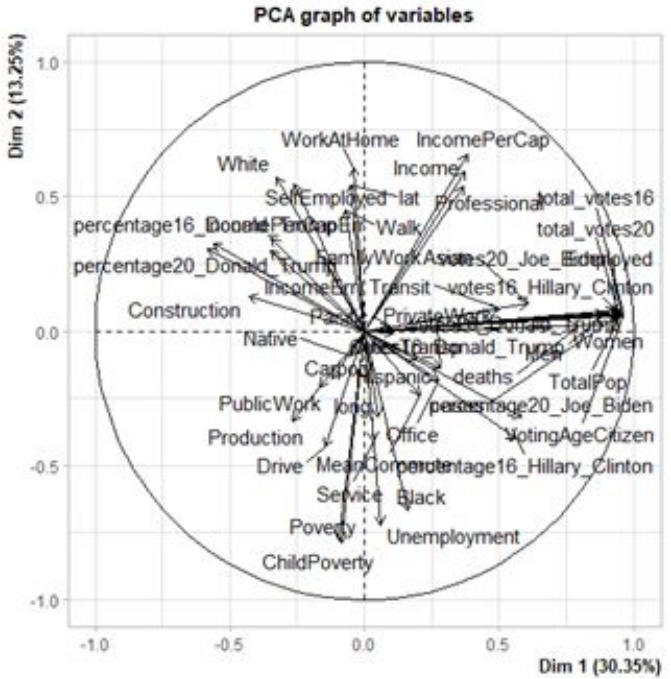
- Principal Components Analysis
- Factor Analysis
- Linear Regression
- K-Nearest Neighbor
- Cluster Analysis
- Decision tree Analysis
- Neural Networks

Components Analysis (PCA)

- Purpose:
  - PCA was used to determine which variables play a larger role in explaining the total variance
  - Also used to find the number of composite variables that can be created using the original data set
- The original data set contained 51 variables

PCA - Number of Components

- The graph to the right shows all of the variables in relation to the first two dimensions
- The following four criterion were used to determine the PCs:
  - Eigenvalue Criterion
  - Proportion of Variance Explained
  - Scree plot criterion
  - Minimum Communality Criterion
- Using the PCA guidelines, it was determined that 11 Dimensions were needed to get a good solution



PCA - Dimension Names per Variable Loadings

- Using varimax rotation we were able to name the dimensions.

Dimensions described based on variable loading:

- Dim 1: Voting Party Characteristics
- Dim 2: Voter Finances/Employment Status
- Dim 3: Public Sector Worker Voters
- Dim 4: Voting Candidate Percentage
- Dim 5: White, Black, Hispanic Voters
- Dim 6: Private Sector Worker Voters
- Dim 7: Voter Income
- Dim 8: County Accessibility (Commute Times)
- Dim 9: Service/Professional Worker Voters
- Dim 10: Asian/Pacific Voters
- Dim 11: Voter Working Status (i.e. self employed)

- Using PCA we were able to reduce the number of variables from 51 to 11 PCs
- PCA also helped us understand the way variables were loaded on the components which also gives us initial insight on variable relationships

Factor Analysis

- Purpose:
  - Analyze who participated in the vote
  - Connect with the policies of presidential candidates

- Solution
  - Use Oblique Rotation (Promax) which allows for more of a correlation between factors.

Factor Choose

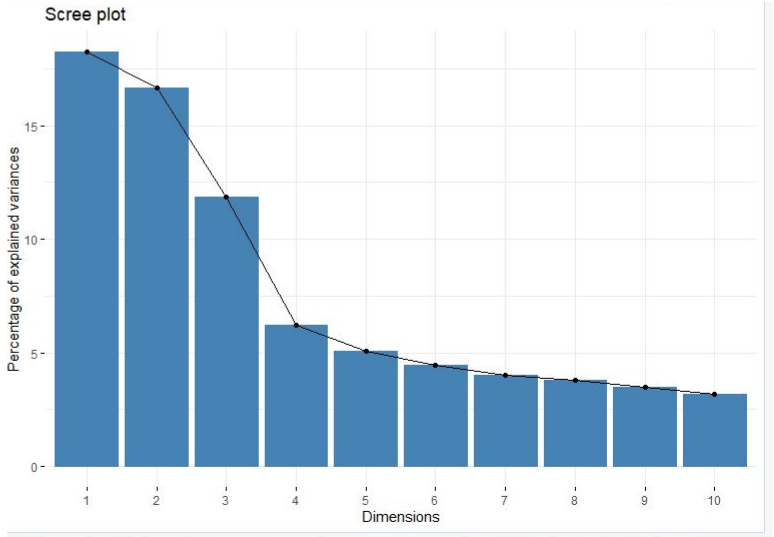
- All eigenvalues are greater than 1 (SS Loading).
- Cumulative variances are below 0.5 .
- Used screeplot confirms 4 are necessary.

4 factor has more clearly.

	Factor1	Factor2	Factor3	Factor4
SS loadings	4.681	3.466	3.040	2.495
Proportion Var	0.156	0.116	0.101	0.083
Cumulative Var	0.156	0.272	0.373	0.456

	Factor1	Factor2	Factor3	Factor4	Factor5
SS loadings	3.447	3.396	2.894	2.555	2.367
Proportion Var	0.115	0.113	0.096	0.085	0.079
Cumulative Var	0.115	0.228	0.325	0.410	0.489

	Factor1	Factor2	Factor3	Factor4	Factor5	Factor6
SS loadings	3.117	3.059	3.015	2.607	2.468	2.364
Proportion Var	0.104	0.102	0.100	0.087	0.082	0.079
Cumulative Var	0.104	0.206	0.306	0.393	0.476	0.554



Factor Analysis

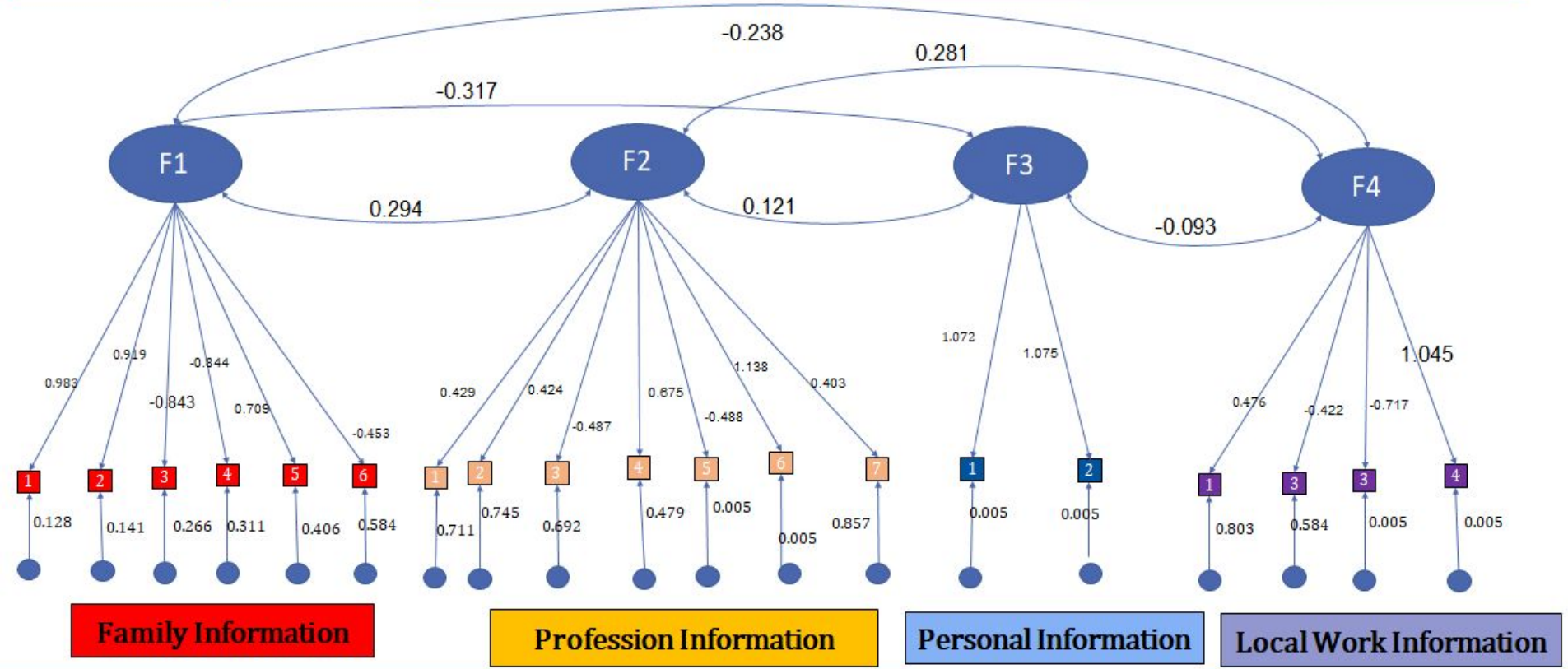
- The lower the uniqueness, the greater the relevance of the variable in the model.
  - IncomePerCap
  - Employed
  - PrivateWork
  - PublicWork
  - SelfEmployed

```
Call:
factanal(x = statistics, factors = 4, rotation = "promax")

Uniquenesses:
       Hispanic           White           Black          Native           Asian         Pacific VotingAgeCitizen          Income       IncomeErr
          0.945           0.707           0.769           0.803           0.632           0.988           0.005           0.128           0.776
   IncomePerCap    IncomePerCapErr         Poverty   ChildPoverty    Professional         Service          Office    Construction      Production
          0.141           0.711           0.266           0.311           0.406           0.787           0.847           0.745           0.584
       Drive         Carpool         Transit            Walk       OtherTransp     WorkAtHome     MeanCommute        Employed     PrivateWork
          0.692           0.935           0.787           0.771           0.982           0.479           0.908           0.005           0.005
  PublicWork    SelfEmployed      FamilyWork
          0.005           0.005           0.857
```

loadings:

```
Loadings:
                Factor1 Factor2 Factor3 Factor4
Hispanic
White
Black
Native                                  0.426
Asian
Pacific
VotingAgeCitizen                1.072
Income          0.983
IncomeErr
IncomePerCap    0.919
IncomePerCapErr         0.429
Poverty        -0.843
ChildPoverty   -0.844
Professional    0.709
Service
Office
Construction            0.424
Production     -0.453                   -0.422
Drive                  -0.487
Carpool
Transit
Walk
OtherTransp
WorkAtHome              0.675
MeanCommute
Employed                        1.075
PrivateWork            -0.488   -0.717
PublicWork                               1.045
SelfEmployed            1.138
FamilyWork              0.403

                Factor1 Factor2 Factor3 Factor4
SS loadings       4.681   3.466   3.040   2.495
Proportion Var    0.156   0.116   0.101   0.083
Cumulative Var    0.156   0.272   0.373   0.456
```

1. Income	4. ChildPoverty	1. IncomePerCapErr	4. WorkAtHome	7.FamilyWork	1.Native	4. PublicWork
2. IncomePerCap	5. Professional	2. Construction	5. PrivateWork	1. VotingAgeCitizen	2. Production	
3. Poverty	6. Production	3. Drive	6. SelfEmployed	2. Employed	3. PrivateWork	



Policy Connect

Family Information	Profession Information	Personal Information	Local Work Information
Income	SelfEmployed	VotingAgeCitizen	Native
Poverty	WorkAtHome	Employed	PublicWork
IncomePerCap			

Same: Both want to improve the economy of the country. **Family Information**

Different:

Joe Biden	Donald Trump
Biden opposes reopening the economy without the COVID-19 test. **Profession Information: WorkAtHome**	Trump pushes for the reopening of the economy. **Local Work Information: PublicWork**
Biden pledges to enhance diversity and inclusion in all the appointed opportunity regardless of the race. **Personal Information: Employee**	Trump's slogan is "Make America Great Again" and offer jobs to American workers first. **Local Work Information: Native**

Regression Analysis

- **Purpose: To determine if there are any overarching trends between demographic information and voter behavior**

Regression Analysis (Republican)

- Complete model contained all variables involving covid case numbers, gender, race, income, job type, job sector, and transportation methods.
- After the complete model was compiled in r, a backward selection was run on the model to remove insignificant coefficients.

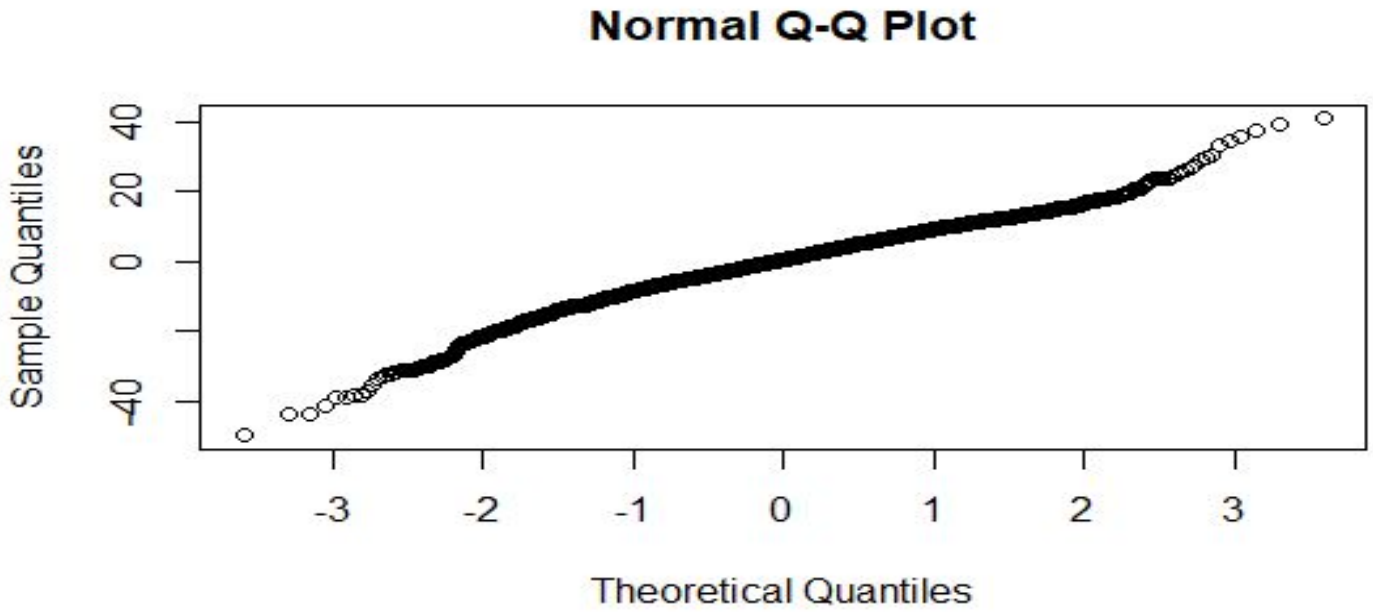
Percent of People who voted for Donald Trump =173-.49(female)-.31(hispanic)-.51(black)-.44(native)-1.17(asian)+.11(child poverty)+1.23(construction)-1.74(Private Work)-1.64(public work)-.85(self employed)+.83(Drive)+.78(Carpool)+.44(Transit)-.30(Work at Home)

r-squared	.623
Adjusted r-squared	.622
Median of Residuals	.4
Model p-value	2.2e-16

Vif() Analysis for Republican Model

Percentage Female	1.16	Private Work	320.5
Hispanic	1.27	Public Work	209.37
Black	1.67	Self Employed	93.60
Native	1.44	Drive	6.05
Asian	1.52	Carpool	2.23
Child Poverty	1.73	Transit	3.21
Construction	1.61	Work at Home	3.96

Normality (Republican)



Regression Analysis (Democrat)

- Complete model contained all variables involving covid case numbers, gender, race, income, job type, job sector, and transportation methods.
- After the complete model was compiled in r, a backward selection was run on the model to remove insignificant coefficients.

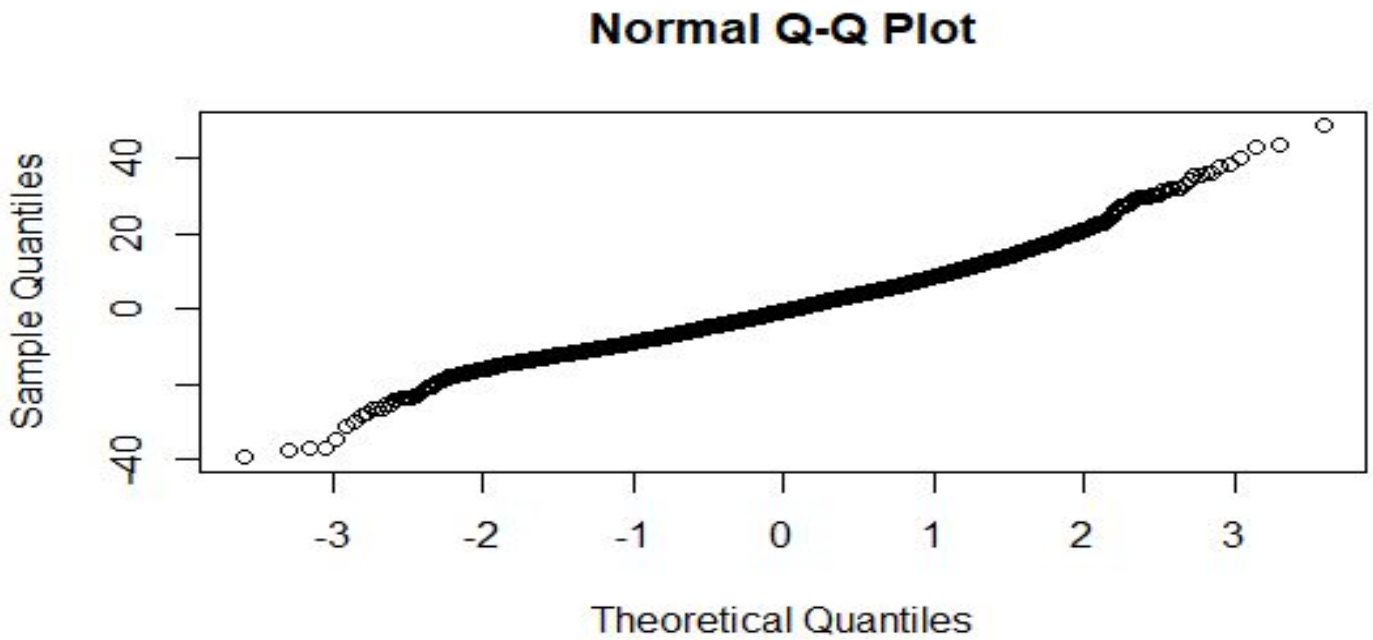
  Percent of People who voted for Joe Biden =-74.7+.49(female)+.31(hispanic)+.52(black)+.43(native)+1.17(asian)-.16(Poverty)-1.27(construction)+1.72(Private Work)+1.62(public work)+.86(self employed)-.81(Drive)-.76(Carpool)-.41(Transit)+.28(Work at Home)

R squared	.632
Adjusted r squared	.630
Median of residuals	-.56
Model p-value	2.2e-16

Vif() Analysis for Democrat Model

Percent Female	1.16	Private Work	320.8
Hispanic	1.27	Public Work	209.4
Black	1.62	Self Employed	93.69
Native	1.46	Drive	6.07
Asian	1.52	Carpool	2.22
Poverty	1.78	Transit	3.22
Construction	1.60	Work at Home	4.03

Normality (Democrat)



Voting Patterns Insights

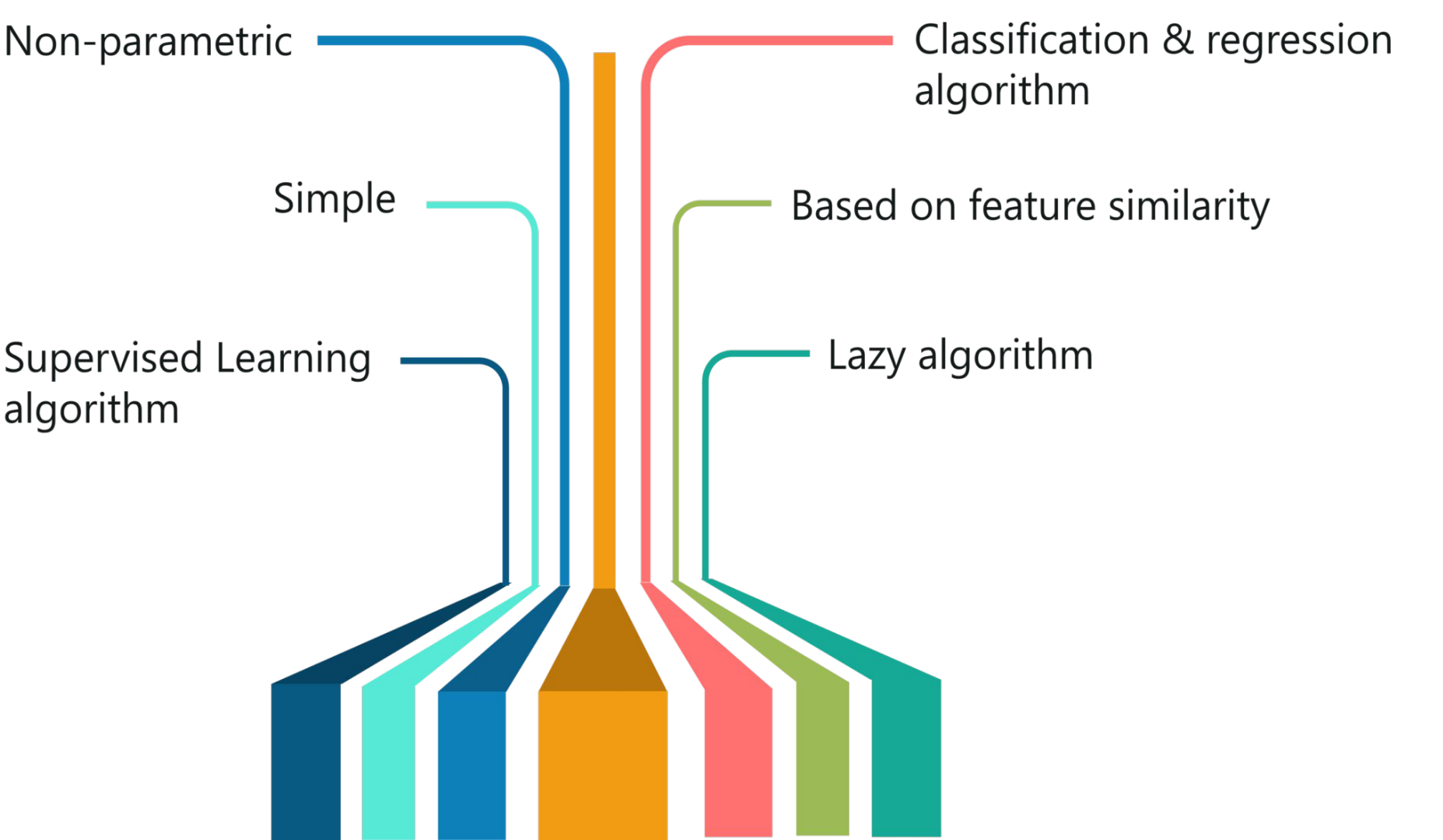
- An increase in female voting resulted in a higher level of democratic voters per county
- Counties with a higher percentage of non-white voters generally had a higher percentage of democratic voters.
- Counties who's population worked a large number of construction oriented position generally provided a higher republican turnout.
- Higher percentages of employment in all job sectors resulted in a lower number of republican votes (Strong negative correlation of -.88 between public work and private work)
- Since the coefficients of all job sector variables are moving in the same direction in each model, this is most likely due to overall employment levels and not job sector
- Counties that had high levels of driving and carpool based commuting had a higher republican vote count.

K-Nearest Neighbor (KNN)

KNN algorithm is one of the most widely used algorithms due to its simplicity. KNN assumes no specific parameters or distribution of data. This serves as a key advantage since the data does not need to conform to a theoretical distribution or be transformed.

KNN is also a lazy algorithm. Lazy algorithms do not use the training data to do generalization. KNN uses all the data for training.

Therefore, KNN is an excellent choice for a classification study when there is little or no prior knowledge about the distribution data, or when the analyst simply is looking to explore the data and new data.


Non-parametric
Classification & regression algorithm
Simple
Based on feature similarity
Supervised Learning algorithm
Lazy algorithm

```
vote16_norm$rndnum = runif(3109, 1,100)
vote_train = vote16_norm [ vote16_norm[, "rndnum"] <= 95    ,     ]
vote_test = vote16_norm [vote16_norm[, "rndnum"] > 95    ,     ]
```

Splitting the data into Training data of 95% and testing data of 5%

```
library(class)
vote_pred <- knn(vote_train, vote_test, target, k= 55, prob = T)
```

Knn Prediction

```
> vote_pred
  [1] Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes No  Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes No  Yes Yes Yes Yes Yes
 [35] Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes No  No  No  Yes
 [69] Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes No  Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes No  Yes Yes Yes Yes
[103] Yes Yes Yes Yes Yes No  Yes Yes Yes Yes Yes Yes No  Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes
[137] Yes Yes No  Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes
attr(,"prob")
  [1] 0.9272727 1.0000000 0.9636364 1.0000000 1.0000000 0.9454545 1.0000000 0.9090909 0.8727273 0.9454545 1.0000000 0.7272727 0.8909091
 [14] 0.9818182 1.0000000 1.0000000 1.0000000 0.9272727 1.0000000 0.9818182 0.9818182 0.9818182 1.0000000 0.9818182 0.9272727 1.0000000
 [27] 0.9636364 0.7272727 0.9818182 0.9272727 0.9636364 0.9818182 0.9272727 0.5454545 0.8363636 0.9642857 0.6000000 1.0000000 0.9636364
 [40] 0.8363636 1.0000000 0.9272727 1.0000000 0.6727273 0.9818182 1.0000000 0.9636364 1.0000000 0.9636364 0.7272727 0.9818182 1.0000000
 [53] 0.9090909 0.8727273 1.0000000 0.9636364 1.0000000 0.9636364 1.0000000 0.7636364 0.7636364 0.8909091 0.6727273 0.9636364 0.5272727
 [66] 0.6181818 0.6785714 1.0000000 1.0000000 1.0000000 0.9454545 1.0000000 1.0000000 0.9272727 0.9818182 0.9818182 0.8363636 1.0000000
 [79] 1.0000000 1.0000000 0.8000000 0.9636364 0.9636364 0.9454545 1.0000000 1.0000000 0.8909091 0.9818182 1.0000000 1.0000000 1.0000000
 [92] 1.0000000 0.8727273 1.0000000 0.9818182 0.9454545 0.6727273 0.7636364 1.0000000 0.8727273 1.0000000 0.8181818 0.9090909 1.0000000
[105] 0.8181818 0.9818182 0.9272727 0.6909091 1.0000000 1.0000000 0.9454545 0.9818182 1.0000000 0.9636364 0.7272727 1.0000000 0.9272727
[118] 0.9818182 0.6000000 1.0000000 1.0000000 0.9454545 0.9818182 1.0000000 0.9272727 1.0000000 1.0000000 1.0000000 0.9454545 1.0000000
[131] 0.9818182 1.0000000 0.9818182 0.8181818 0.9272727 1.0000000 1.0000000 1.0000000 0.8909091 1.0000000 1.0000000 0.9818182
[144] 1.0000000 1.0000000 1.0000000 0.9818182 0.8363636 0.9818182 0.9090909 0.8909091 0.8000000 1.0000000 0.8545455 0.9636364 0.8545455
[157] 1.0000000 0.9818182 0.9272727 1.0000000 0.9818182 1.0000000
Levels: No Yes
```

```
> table(vote_pred, targettest)
         targettest
vote_pred  No Yes
      No    9   1
      Yes   8 144
```

The accuracy is Trump's win prediction is of 94.69%

```
d_norm <- function(x) { ((x-min(x))/(max(x)-min(x)))}

vote20_norm <- as.data.frame(lapply(vote20, d_norm))
vote20_norm$Trump_Lose <- truetrumplose

vote20_norm$rndnum = runif(3048, 1,100)
vote_train = vote20_norm [ vote20_norm[, "rndnum"] <= 95     ,      ]
vote_test = vote20_norm [vote20_norm[, "rndnum"] > 95    ,      ]
```

Splitting the data to training set and testing set with 95% and 5%

```
library(class)
vote_pred <- knn(vote_train, vote_test, target, k= 52,prob = T)
```

KNN prediction

```
> vote_pred <- knn(vote_train, vote_test, target, k= 52,prob = T)
> vote_pred
  [1] No  No  Yes No  No  No  No  No  No  No  No  No  No  No  No  Yes No  No  No  No  No  No  No  No  No  No  No  Yes No  No  No  No  No  Yes
 [35] No  No  No  No  Yes No  No  No  No  No  No  No  Yes No  No  No  No  No  Yes No  No  No  No  Yes No  No  No  No  No  No  No  No  No  No
 [69] No  No  No  No  Yes Yes No  No  No  No  No  No  No  No  Yes No  No  No  No  No  No  No  No  No  No  No  Yes No  No  No  No  No  No
[103] No  No  No  No  Yes No  No  No  No  No  No  No  No  No  No  No  No  No  No  No  Yes No  No  No  No  No  No  No  No  No  No  No  No  No
[137] No
attr(,"prob")
  [1] 0.9807692 1.0000000 0.9038462 1.0000000 0.9423077 0.9807692 1.0000000 0.9423077 1.0000000 1.0000000 1.0000000 0.8076923 0.8461538
 [14] 1.0000000 1.0000000 0.5192308 0.7307692 0.9423077 0.8653846 1.0000000 0.8269231 0.8269231 0.9807692 0.9615385 0.9423077 1.0000000
 [27] 1.0000000 0.7115385 0.9230769 1.0000000 0.9807692 0.9807692 0.9423077 0.7115385 0.9615385 1.0000000 0.9615385 0.9807692 0.6538462
 [40] 0.9038462 0.9230769 0.9423077 1.0000000 1.0000000 0.9230769 0.8076923 0.9230769 0.7115385 0.9807692 1.0000000 0.9615385 1.0000000
 [53] 0.6538462 1.0000000 1.0000000 0.9423077 1.0000000 0.5576923 0.9038462 0.8653846 0.9038462 0.8846154 0.9423077 0.9615385 1.0000000
 [66] 0.9423077 1.0000000 0.9807692 1.0000000 0.9807692 1.0000000 0.8076923 0.8269231 0.5576923 0.9615385 0.9230769 0.9230769 1.0000000
 [79] 0.6153846 0.9615385 1.0000000 1.0000000 0.8461538 1.0000000 0.9038462 1.0000000 0.9615385 1.0000000 0.9230769 1.0000000 0.9423077
 [92] 1.0000000 1.0000000 1.0000000 0.6730769 0.8846154 0.9038462 0.8269231 0.9615385 0.6538462 0.6923077 0.6730769 1.0000000 0.9423077
[105] 0.9615385 1.0000000 0.5769231 1.0000000 0.9230769 0.5384615 0.8846154 0.9615385 0.9230769 0.9807692 0.9615385 0.8846154 1.0000000
[118] 0.9807692 0.9615385 1.0000000 0.9423077 0.6923077 1.0000000 0.9807692 0.9807692 1.0000000 0.8461538 0.8653846 0.8679245 0.7307692
[131] 1.0000000 0.9615385 0.8076923 1.0000000 1.0000000 0.8269231 0.8269231
Levels: No Yes
```

```
> table(vote_pred,targettest)
         targettest
vote_pred  No Yes
      No  115   8
      Yes   2  12
```

93.04% of accuracy of Trump's lose in election

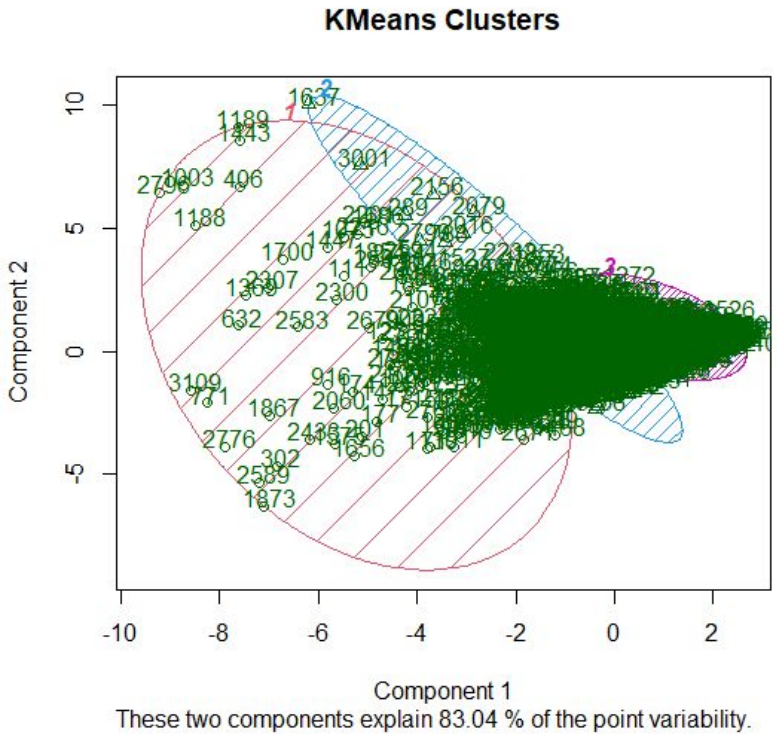
Cluster Analysis

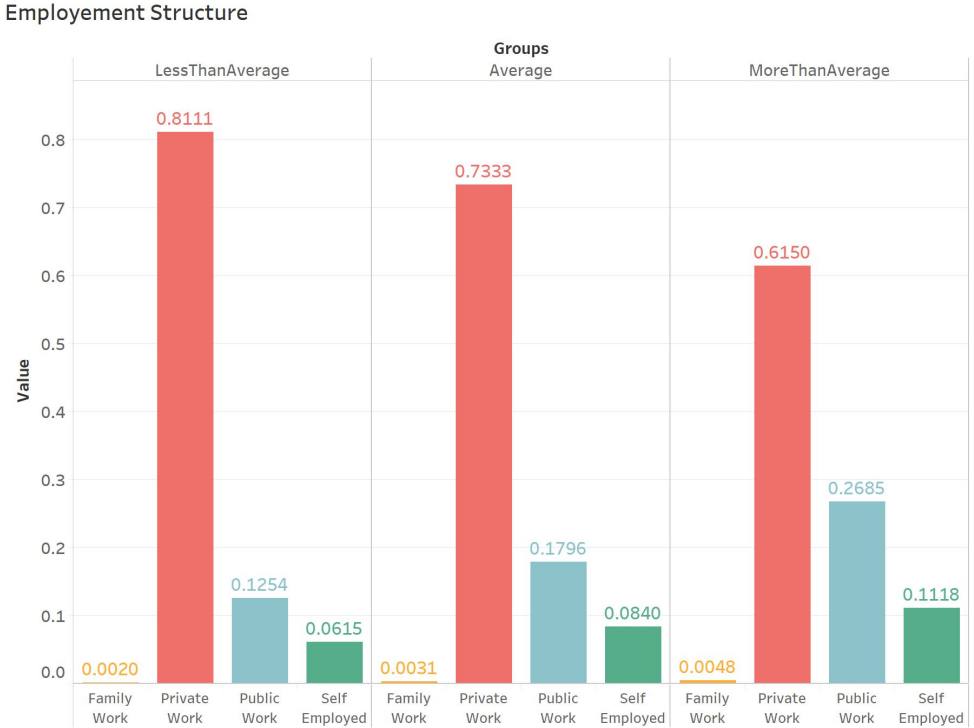
- Purpose:

  Try to find out the relationship between employment structure, especially self-employed people, and the voting choice.

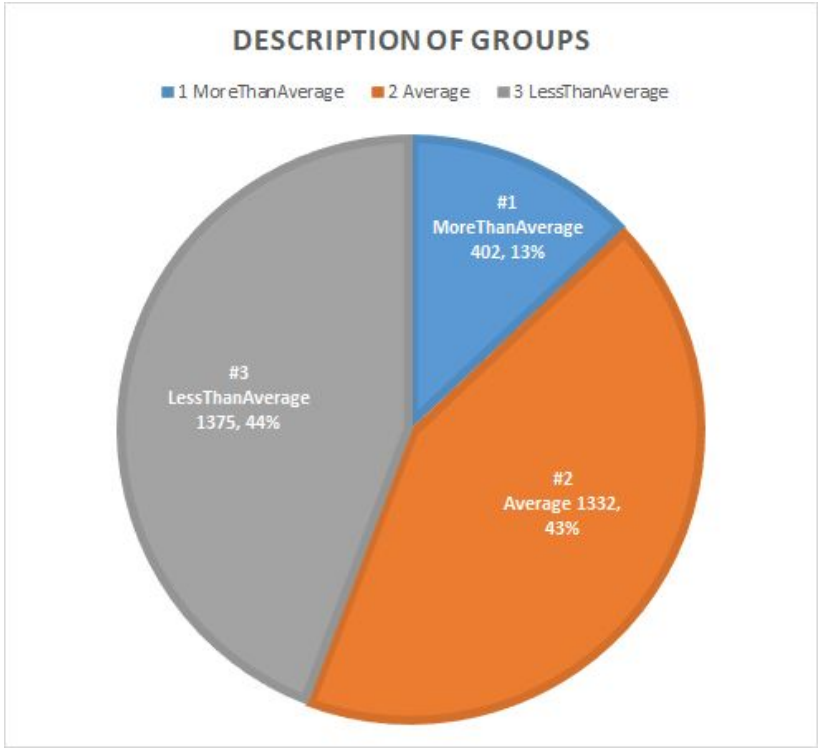
- Assumptions:
  1. Same influence
  2. Different Policy Preferences

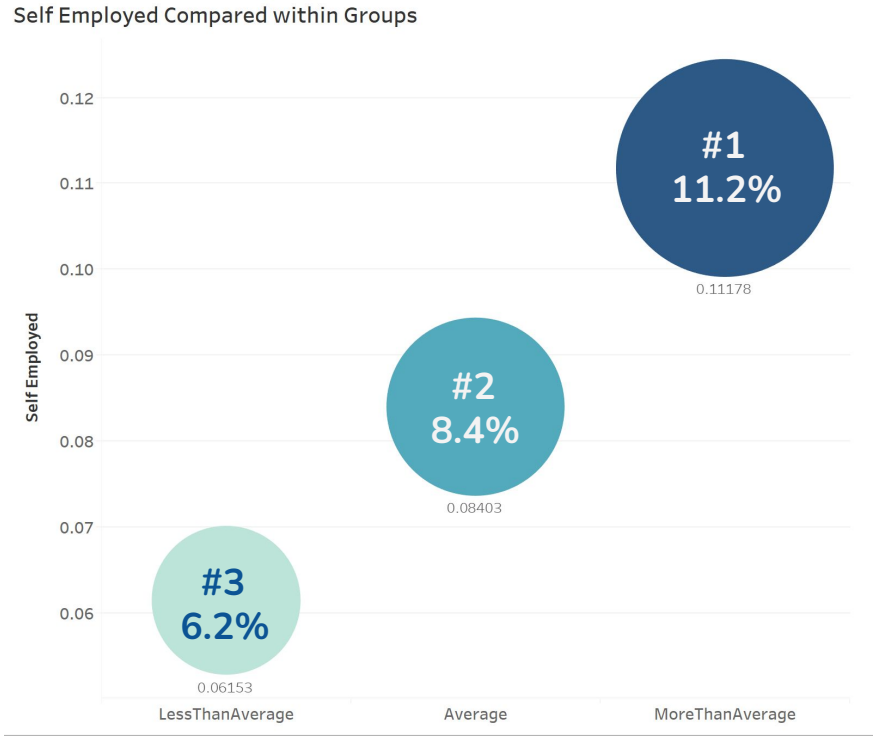
Cluster Analysis -- Groups description



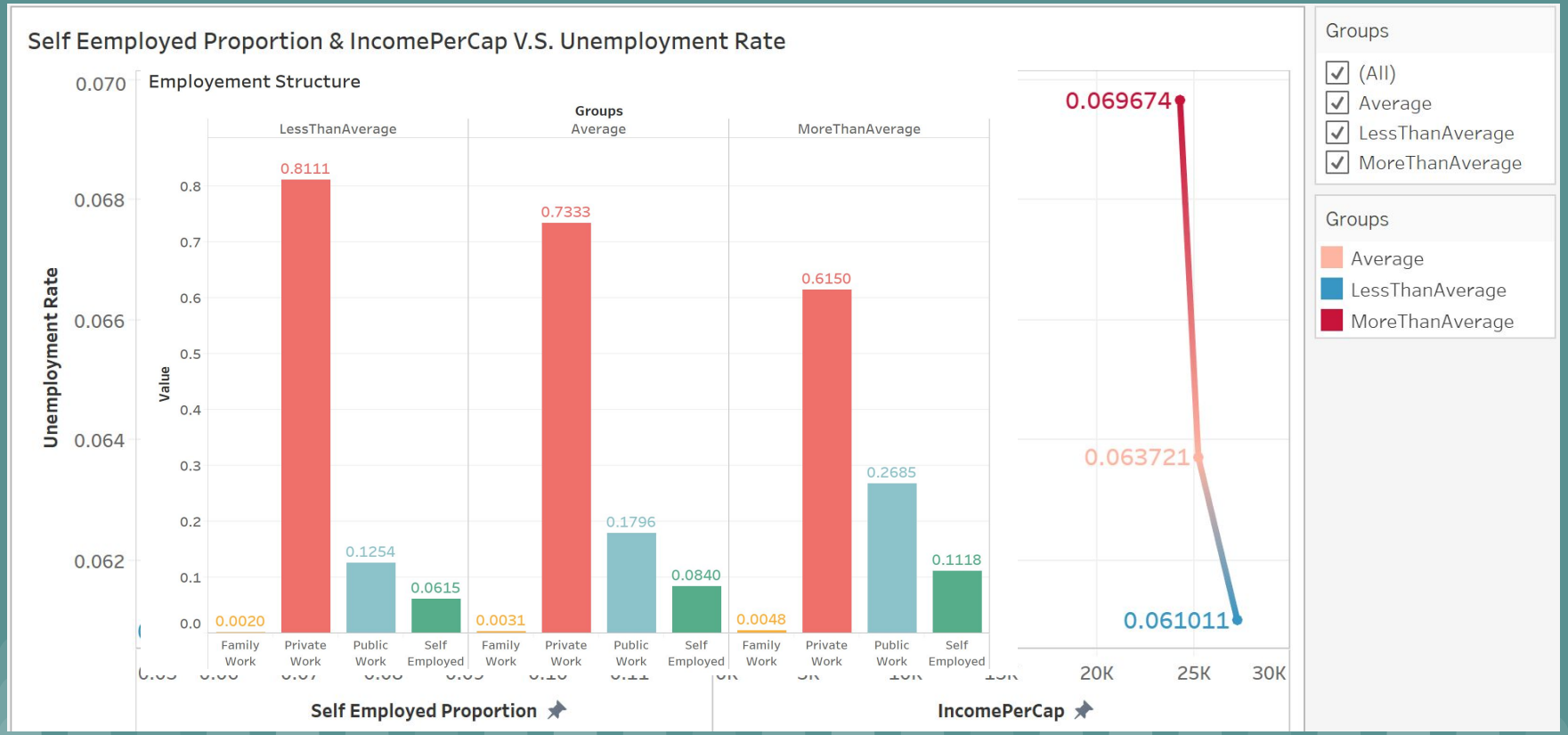


Cluster Analysis -- Groups description

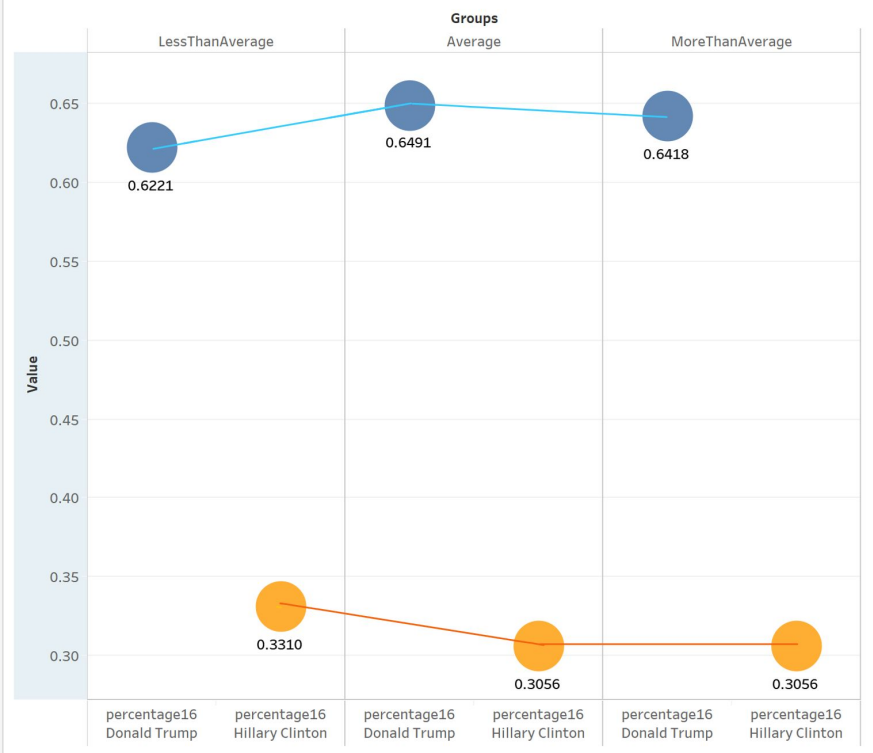




Cluster Analysis: Unemployment rate and IncomePerCap



Cluster Analysis: Voting Rate



- Conclusion:

the employment structure is a factor that electors must consider when making campion strategy.

Decision Tree Analysis

- Purpose:

  Decision tree aimed at classifying two groups of people such as poor or rich depending on the income.

- A county survey statistics dataset consisting of 51 variables was used.
- The data contained employment information as well as demographic data of the employees.
- There were employees from private, government and self-employed respondents.

Decision Tree Analysis steps

- To build our Decision tree, we will proceed as follow steps:
- Step1-Data processing and manipulation:

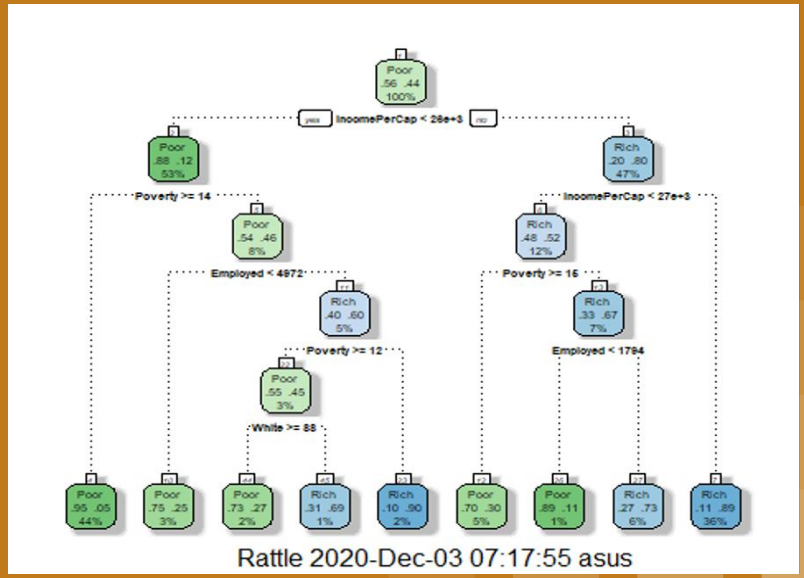
```
Project<-read.csv("C:/Users/asus/Desktop/county_statistics.csv",stringsAsFact
ors=TRUE)

summary(Project[,28])

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   19264   41123   48066   49754   55764  129588   17253
Project$Income<-ifelse(Project$Income<=49700,0,1)
Project$Income<-factor(Project$Income,levels=c(0,1),labels=c("Poor","Rich"))
```

- Step2-Data Cleaning

```
Project1<-na.omit(Project)
Project2<-Project1[,-c(1:18,27)]
```

Tree Regression Analysis

```
#Decision Trees
library(tree)
library(rpart)
#Decision tree regression analysis

Decision<-rpart(formula=Income~.,data=Project2, method="class")

library(rattle)

library(rpart.plot)
library(RColorBrewer)
fancyRpartPlot(Decision)
```

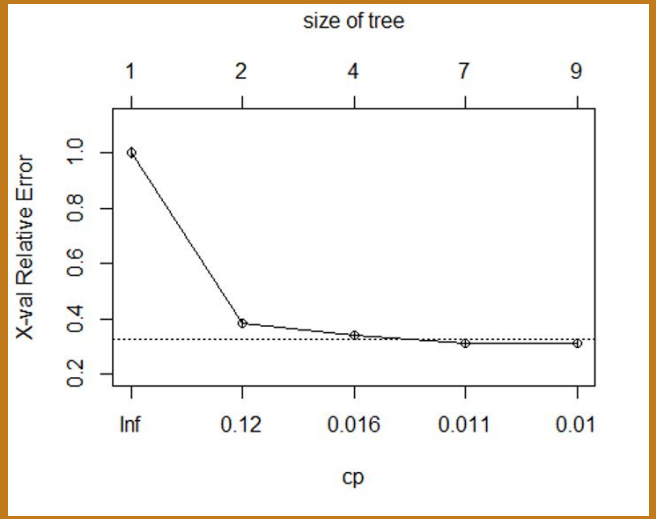


Tree regression analysis

```
printcp(Decision)

##
## Classification tree:
## rpart(formula = Income ~ ., data = Project2, method = "class")
##
## Variables actually used in tree construction:
## [1] Employed     IncomePerCap Poverty      White
##
## Root node error: 1342/3046 = 0.44058
##
## n= 3046
##
##         CP nsplit rel error  xerror     xstd
## 1 0.639344      0   1.00000 1.00000 0.020417
## 2 0.022355      1   0.36066 0.38301 0.015403
## 3 0.011177      3   0.31595 0.34277 0.014726
## 4 0.010060      6   0.28241 0.31148 0.014151
## 5 0.010000      8   0.26230 0.31148 0.014151
```

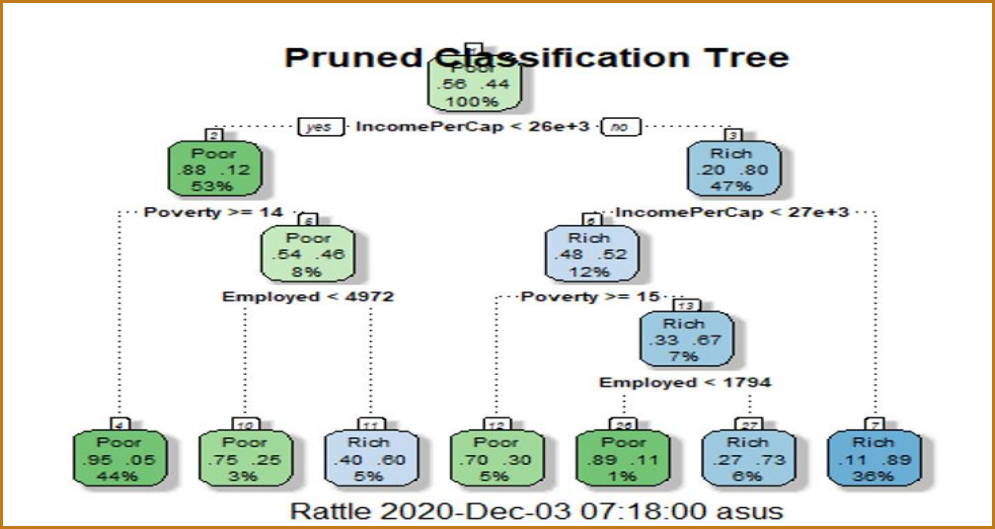
Validating (decision tree)

```
Decision$cptable[which.min(Decision$cptable[,"xerror"]),"CP"]

## [1] 0.01005961

plotcp(Decision)
```



Pruned decision tree

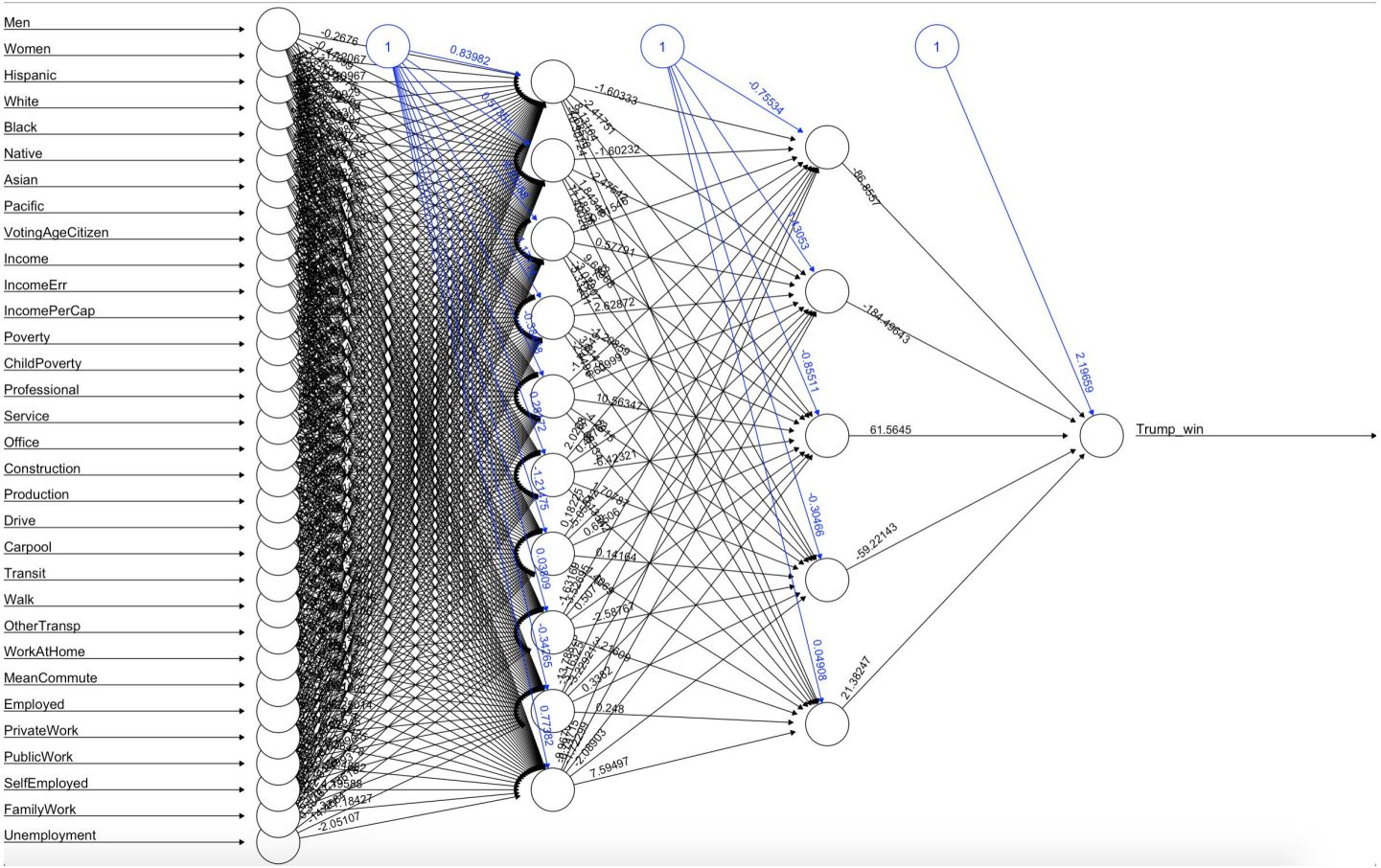
```
#pruned optimal decision will be
pDecision<- prune(Decision,cp=Decision$cptable[which.min(Decision$cptable[,"x
error"]),"CP"])

#ploting pruned optimal decision tree
fancyRpartPlot(pDecision, uniform=TRUE,main="Pruned Classification Tree")
```



Neural Network

By using all attributes (columns) associated with every county, we were able to predict the election results using a neural network.

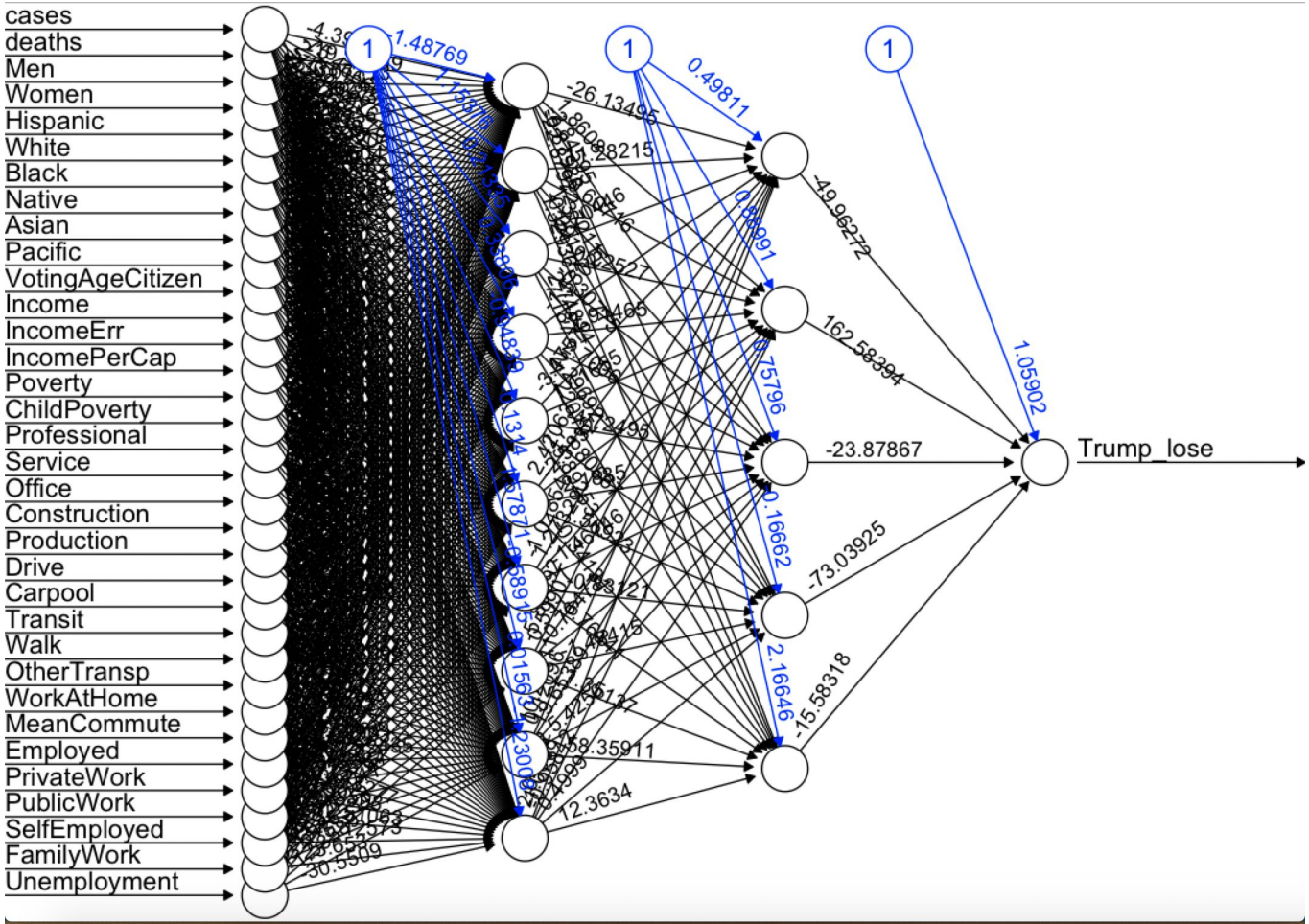
Election Prediction 2016- Hidden Layers (10, 5)



	Election16_prediction	Trump_win	Error
1	1.0	1	1
2	1.0	1	1
3	1.0	1	1
4	1.0	1	1
5	1.0	1	1
6	1.0	1	1
7	1.0	1	1
8	1.0	1	1
9	0.0	0	1
10	1.0	1	1
11	1.0	1	1
12	1.0	1	1
13	1.0	1	1
14	0.0	0	1
15	1.0	1	1
16	1.0	1	1
17	1.0	1	1
18	1.0	1	1
19	1.0	1	1
20	1.0	1	1
21	0.0	0	1
22	1.0	1	1
23	1.0	1	1
24	0.0	0	1
25	1.0	1	1
26	0.0	0	1
27	1.0	0	0
28	0.0	0	1
29	1.0	1	1
30	0.0	0	1
31	1.0	1	1
32	1.0	1	1
33	1.0	1	1

We were able to predict the election results within an error of 1.25%

Election 20 Prediction - Hidden Layers (10, 5)



row names	Election20_prediction	Trump_lose	Error
2	0	0	1
3	0	0	1
4	0	0	1
5	0	0	1
6	0	0	1
7	0	0	1
8	0	0	1
9	1	1	1
10	0	0	1
11	0	0	1
12	0	0	1
13	0	0	1
14	1	1	1
15	0	0	1
16	0	0	1
17	0	0	1
18	0	0	1
19	0	0	1

We were able to predict the election results within an error of 1.25%