A Push For Fair Wages - Anlysis Of Income Inequality In The US

Omar Choudhry (oachoud2), Matthew Hamburger (mhambur2), Donghan Liu (donghan2), Ruilin Zhao (rzhao15)

May 9, 2018

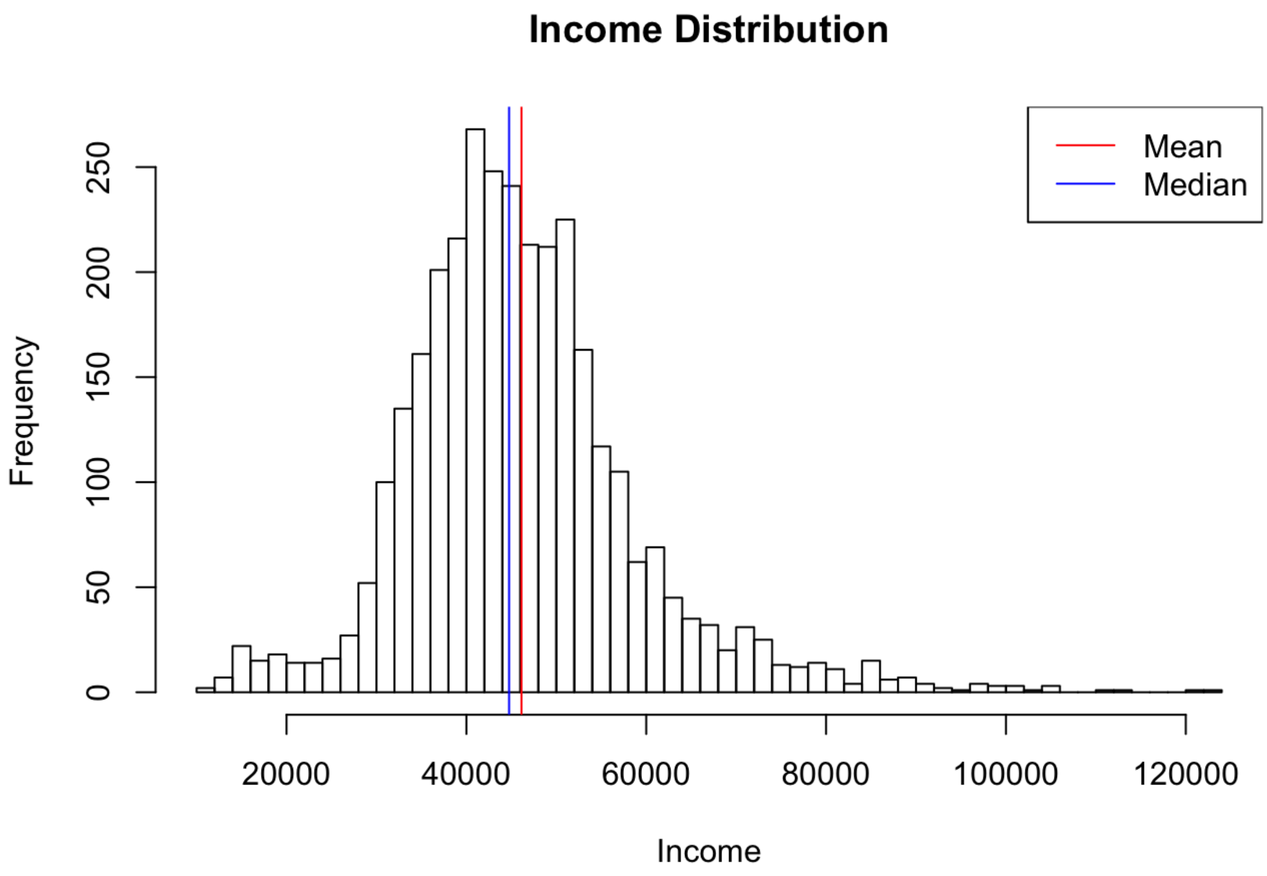
Abstract

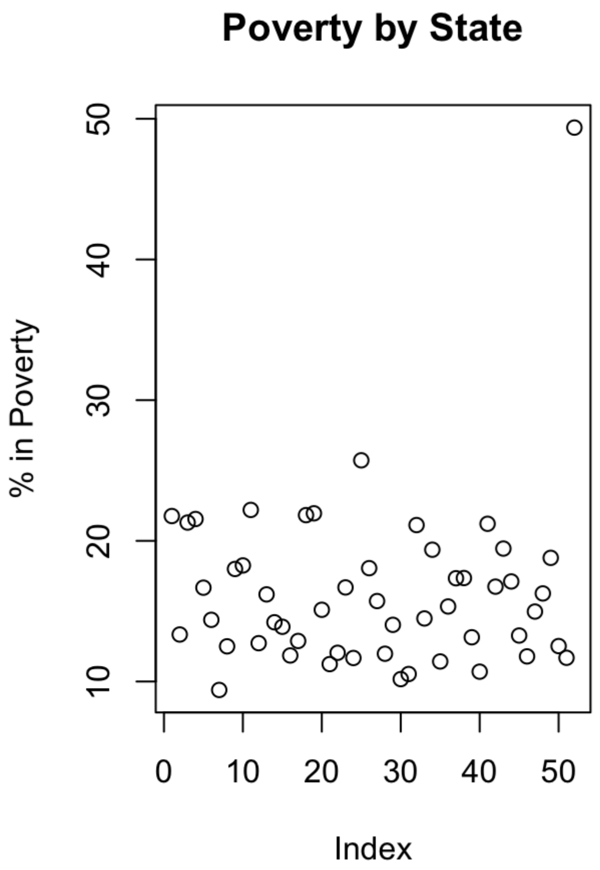
This study uses demographic variables from US census data to predict income in counties across the US. The motivation behind predicting income is to see if there are any Income Inequalties that stem from these demographics that could be used as part of social movemements towards income equality. We fit many variable selection regression models to narrow down our models to the most significant variables, using RMSE as an indicator of model accuracy. After testing all our models, we found that a polynomial regression gave the best results and omitted enough variables to make the model interpretable. We concluded that demographic variables are significant when predicting income, and that there is some evidence of income inquality when it comes to gender and race.

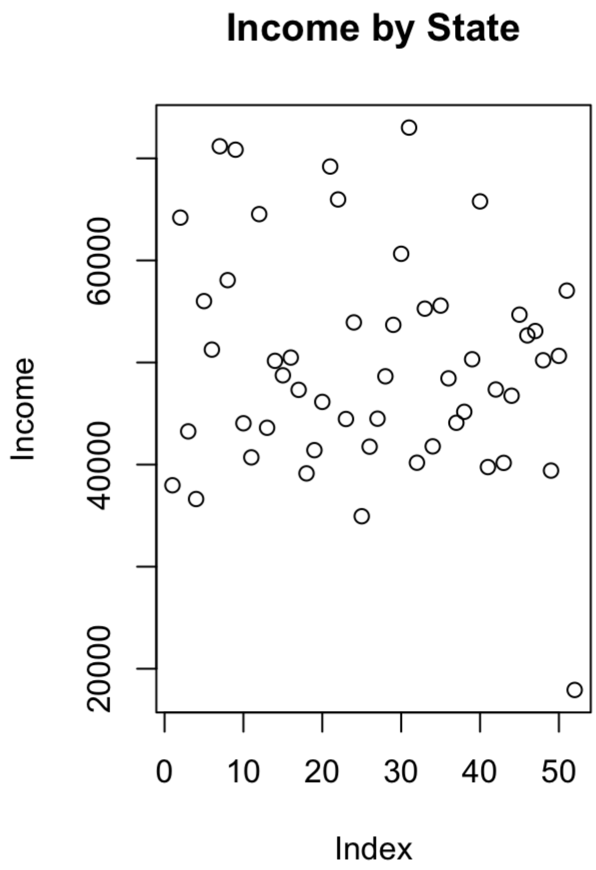
Introduction

The dataset we used for this project was US Census data taken from Kaggle. It contains 1 entry for each county in the United States. This comes out to 3218 rows and 35 columns/variables describing each county. Some of the variables include, race, occupation, transportation, gender, average income, and population. The purpose of our analysis is to predict income based on the some attributes in the dataset. Our goal is to most accurately predict the Income of a given county in order to see if there is a significant inequality due to things like race, gender, occupation, and means of commute. This is a very relevant social question, and quantitative significance could be used as proof for income inequality movements in the US.

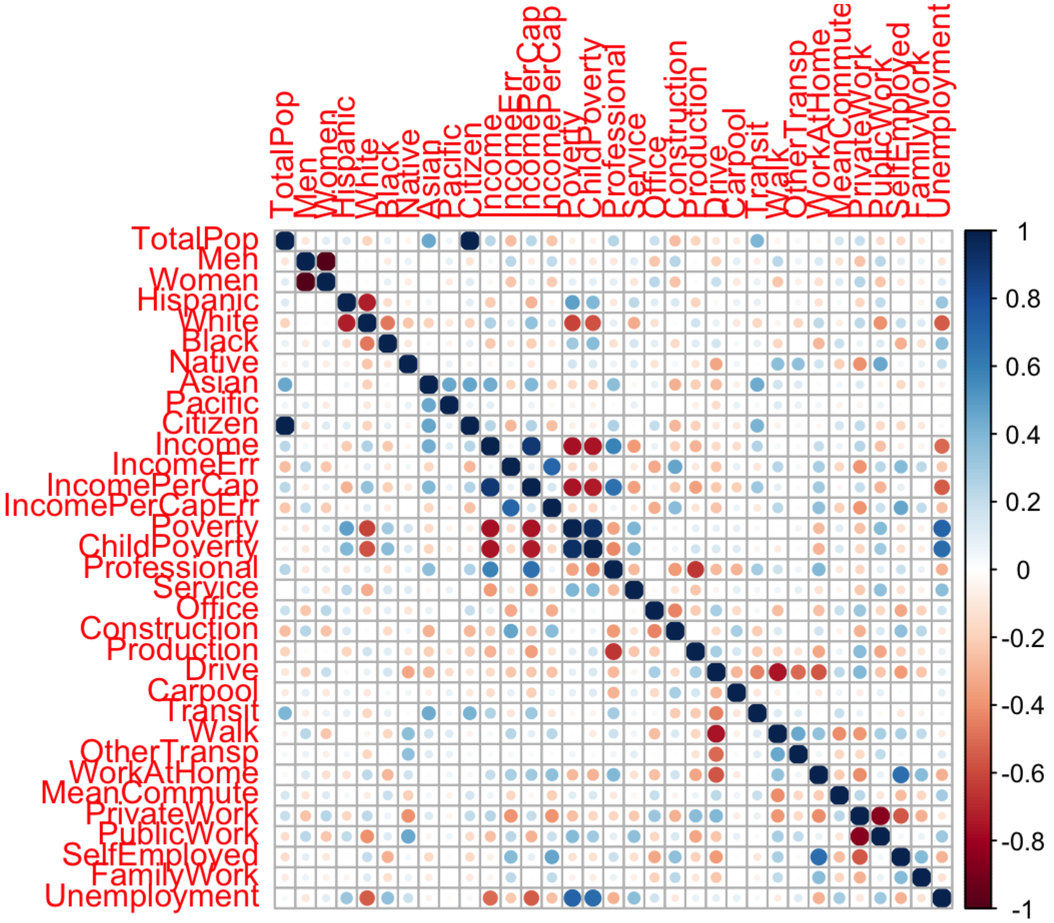
Exploratory Data Analysis


Income Distribution
Mean
Median
Frequency
0
50
100
150
200
250
20000
40000
60000
80000
100000
120000
Income


Poverty by State
% in Poverty
10
20
30
40
50
0
10
20
30
40
50
Index


Income by State
Income
20000
40000
60000
0
10
20
30
40
50
Index

Correlation Between Variables
Er



From the histogram of Income, we can observe that the income from the data roughly follows a normal distribution. Both the mean and median lines are close to each other, and it is shaped like a bell curve. However, we detected, there are some outliers in the data, which could potentially influence our model fitting. Puerto Rico, bottom right of the income graph and top right of the poverty graph should definitely be removed, because it is unlike any of the other data points in our set.

We also created a correlation plot for the full data frame to see if there was any potential problems for multicollinearity and which variables are most highly correlated. Based off intuition poverty and unemployment will be variables to closely watch. They are both useful indicators, but are also candidates for correlation with each other as well as other predictors. In order the top 5 predictors that have the highest correlation with income are; poverty, child poverty, professional, Asian and unemployment rate. However, some concerns are proved to be true as poverty, child poverty and unemployment are all correlated with one another. This will be something that we will keep an eye on as our work progress, but hopefully some of the more advanced models will pull two of the correlated factors out.

Methods

Data

We had to perform some data manipulation and processing in order to prepare the data for model fitting. First we dropped all non-numeric variables, State and County, which will not be helpful when we fit the models as County is unique per row, and State has many categories. We then changed the variables Men, Women and Citizen from population counts to percentages of population to make them more comparable across counties. This also solved the multicollinearity issue between the 3 variables and Population. We also dropped a variable from each percentage category. For example, we dropped Men because Men and Women of each observation will add up to 1. If we included both variables in our models, there would be perfect correlation between variables in the same category. Similarly, we dropped Pacific from races category, OtherTransp from transportation methods and Office from professions. We also dropped observation ID, CensusId, IncomeErr, and IncomePerCapErr, which will be irrelevant in predicting income. Finally, we removed Puerto Rico from our analysis entirely, as our EDA showed that it was a major outlier.

Models

Our goal is to find a model that not only brings us good Income predictions (low RMSE), but also are easy to interpret. Based on this, we fitted quite a few models that should be easy to interpret in order to see which would return the best predictions. We applied simple linear regression models with forward/backward/stepwise selections, a linear model with interaction terms, a polynomial model, a ridge model and a lasso model. We also did two KNN models with scaled/unscaled variables and a random forest model. Because random forest models are hard to interpret, we will just use those results as comparisons to see how well our interpretable models perform.

- Linear model(Forward Selection or Stepwise Selection)

```
set.seed(432)
# stepwise and forward are the same
fwd_lm = train(
  Income ~ Poverty + Professional + Citizen + MeanCommute + SelfEmployed + Asian +
 Walk + Hispanic + Native + PrivateWork + TotalPop + Women + Carpool,
  data = county_trn,
  trControl = trainControl(method = "cv", number = 10),
  method = "lm"
)
```

- Linear model(Backward Selection)

```
set.seed(432)
bwd_lm = train(
  Income ~ TotalPop + Women + Hispanic + Native + Asian + Citizen + Poverty + Prof
essional + Service + Production + Walk + MeanCommute + PublicWork + SelfEmployed,
  data = county_trn,
  trControl = trainControl(method = "cv", number = 10),
  method = "lm"
)
```

- Linear model with interaction terms (Stepwise Selection)

```
set.seed(432)
# stepwise and forward are the same
inter_lm = train(
  Income ~ (Poverty + Professional + Citizen + MeanCommute + SelfEmployed + Asian
 + Walk + Hispanic + Native + PrivateWork + Women + Carpool + Professional:MeanCom
mute + Poverty:Professional + Poverty:Asian + Poverty:PrivateWork + Asian:PrivateW
ork + Poverty:Citizen + Poverty:Women + SelfEmployed:Native + PrivateWork:Women +
 Native:Women + Professional:SelfEmployed + Walk:Women + Native:Carpool + MeanComm
ute:Walk + Poverty:MeanCommute + PrivateWork:Hispanic + Poverty:Hispanic + MeanCom
mute:Hispanic + Native:PrivateWork)^2,
  data = county_trn,
  trControl = trainControl(method = "cv", number = 10),
  method = "lm"
)
```

- Polynomial model(Stepwise Selection)

```
set.seed(432)
# stepwise and forward are the same
poly_fit = train(
  Income ~ Poverty + I(Professional^2) + I(Citizen^2)  + I(Poverty^2) + SelfEmploy
ed + I(MeanCommute^2) + Asian + Native + I(Asian^2) + I(Women^2) + PrivateWork + T
otalPop + I(PrivateWork^2) + Citizen + Professional,
  data = county_trn,
  trControl = trainControl(method = "cv", number = 10),
  method = "lm"
)
```

- KNN model with unscaled predictors (1, 5, 10, 15, 20 and 25 are used as numbers of nearest neighbors)

```
set.seed(432)
knn_unscaled_mod = train(
  Income ~ .,
  data = county_trn,
  trControl = trainControl(method = "cv", number = 10),
  method = "knn",
  tuneGrid = expand.grid(k = c(1, 5, 10, 15, 20, 25))
)
```

- KNN model with scaled predictors

```
set.seed(432)
knn_scaled_mod = train(
  Income ~ .,
  data = county_trn,
  trControl = trainControl(method = "cv", number = 10),
  preProcess = c("center", "scale"),
  method = "knn",
  tuneGrid = expand.grid(k = c(1, 5, 10, 15, 20, 25))
)
```

- Random Forest Model

```
set.seed(432)
rf_mod = train(
  Income ~ .,
  data = county_trn,
  trControl = trainControl(method = "cv", number = 10),
  method = "rf"
)
```

- Ridge Model

```
set.seed(432)
trn_X = model.matrix(Income ~ ., county_trn)[, -9]
trn_y = county_trn$Income
tst_X = model.matrix(Income ~ ., county_tst)[, -9]
tst_y = county_tst$Income

ridge_mod = cv.glmnet(trn_X, trn_y, alpha = 0)
ridge_rmse = sqrt(mean((tst_y - predict(ridge_mod, tst_X, s = "lambda.min")) ^ 2))
ridge_cv_rmse = sqrt(ridge_mod$cvm[ridge_mod$lambda == ridge_mod$lambda.min])
```

- Lasso Model

```
set.seed(432)
lasso_mod = cv.glmnet(trn_X, trn_y, alpha = 1)
lasso_rmse = sqrt(mean((tst_y - predict(lasso_mod, tst_X, s = "lambda.min")) ^ 2))
lasso_cv_rmse = sqrt(lasso_mod$cvm[lasso_mod$lambda == lasso_mod$lambda.min])
```

Results

Method	CV RMSE	Test RMSE
Linear Forward/Stepwise Selection	5594.33	4973.65
Linear backward Selection	5550.47	5005.46

Method	CV RMSE	Test RMSE
Linear Model with Interactions	37862.34	51433.55
Polynomial model	7798.96	4657.19
KNN Unscaled	7156.87	6797.51
KNN Scaled	6067.34	6598.55
Random Forest	3793.20	3661.66
Ridge Model	4036.90	4008.58
Lasso Model	3853.11	3904.95

CV RMSE and Test RMSE are the criteria we are using to determine how well the models make predictions. From the result table, we can see that the random forest model produces the best prediction result and the polynomial model produces the second best prediction. Since random forest model is always hard to interpret and the polynomial model is interpretable, we chose the polynomial model as our best model.

Discussion

Our two best models in terms of RMSE of Income were Random Forest and a second degree Polynomial Regression. Being that our goal is to have an interpretable model for predicting income we have chosen the Polynomial Regression as our best model. The inclusion of the squared terms indicates that income has a non-linear relationship with some of the independent variables. This means that for certain variables there is a peak influence, before larger values start to have a diminishing effect (like a parabola). When applying stepwise selection to he model, it did a good job of taking out factors that we expected to have high multicollinearity.

The final model has narrowed down the races and professions to two variables and only one commuter variable. Looking at our model summary you can see what variables our methods have deemed most statistically significant. Some interesting conclusions are; the Asian and Native variables stayed in the model, mean commute was the only significant variable in terms of transportation and poverty was chosen over unemployment. We also saw that the squared women term stayed in the model.

These are interesting because, the Asian variable has a very positive coefficient, meaning higher Asian population results in significantly higher income. We, see the opposite with the Women variable, where a higher percentage of women in a county points to lower income due to the negative coefficient.

We think Asian and Native stayed in the models because they’re smaller populations, with less variance than the other races. Their populations are also more concentrated, so income differences in those areas could be revealed by the Asian and Native American races. Mean commute could be used as an indicator of location. We would expect people who live in more urban areas to longer commutes than people who live in smaller towns.

In this study we have seen that certain demographic variables are very significant when predicting income. The appearance of the variables Asian and Women in the final model are slightly concerning in terms of Income inequality. This model could be presented as part of petitions and movements towards income inequality to provide quantitative evidence that this is a real issue.

```
## 
## Call:
## lm(formula = .outcome ~ ., data = dat)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -27581  -2896   -277   2414  38169 
## 
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)    
## (Intercept)          5.133e+04  5.792e+03   8.862  < 2e-16 ***
## Poverty             -2.019e+03  4.499e+01 -44.886  < 2e-16 ***
## `I(Professional^2)`  1.503e+01  1.505e+00   9.986  < 2e-16 ***
## `I(Citizen^2)`      -8.426e-10  2.756e-10  -3.057  0.00226 ** 
## `I(Poverty^2)`       1.908e+01  7.794e-01  24.485  < 2e-16 ***
## SelfEmployed        -3.519e+02  3.777e+01  -9.318  < 2e-16 ***
## `I(MeanCommute^2)`   4.964e+00  4.089e-01  12.139  < 2e-16 ***
## Asian                1.635e+03  1.081e+02  15.127  < 2e-16 ***
## Native               2.473e+02  1.897e+01  13.040  < 2e-16 ***
## `I(Asian^2)`        -3.342e+01  3.645e+00  -9.169  < 2e-16 ***
## `I(Women^2)`        -2.130e+04  5.200e+03  -4.096 4.35e-05 ***
## PrivateWork          6.317e+02  1.540e+02   4.103 4.22e-05 ***
## TotalPop             1.972e-02  4.895e-03   4.029 5.77e-05 ***
## `I(PrivateWork^2)`  -3.449e+00  1.086e+00  -3.177  0.00150 ** 
## Citizen             -2.917e-02  7.455e-03  -3.913 9.37e-05 ***
## Professional        -5.294e+02  1.047e+02  -5.056 4.60e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 5097 on 2398 degrees of freedom
## Multiple R-squared:  0.8505, Adjusted R-squared:  0.8495 
## F-statistic: 909.2 on 15 and 2398 DF,  p-value: < 2.2e-16
```

Appendix

```
## 'data.frame':    3220 obs. of  37 variables:
##  $ CensusId       : int  1001 1003 1005 1007 1009 1011 1013 1015 1017 1019 ...
##  $ State          : Factor w/ 52 levels "Alabama","Alaska",..: 1 1 1 1 1 1 1 1
1 1 ...
##  $ County         : Factor w/ 1928 levels "Abbeville","Acadia",..: 90 97 108 16
0 175 236 246 259 312 334 ...
##  $ TotalPop       : int  55221 195121 26932 22604 57710 10678 20354 116648 3407
9 26008 ...
##  $ Men            : int  26745 95314 14497 12073 28512 5660 9502 56274 16258 12
975 ...
##  $ Women          : int  28476 99807 12435 10531 29198 5018 10852 60374 17821 1
3033 ...
##  $ Hispanic       : num  2.6 4.5 4.6 2.2 8.6 4.4 1.2 3.5 0.4 1.5 ...
##  $ White          : num  75.8 83.1 46.2 74.5 87.9 22.2 53.3 73 57.3 91.7 ...
##  $ Black          : num  18.5 9.5 46.7 21.4 1.5 70.7 43.8 20.3 40.3 4.8 ...
##  $ Native         : num  0.4 0.6 0.2 0.4 0.3 1.2 0.1 0.2 0.2 0.6 ...
##  $ Asian          : num  1 0.7 0.4 0.1 0.1 0.2 0.4 0.9 0.8 0.3 ...
##  $ Pacific        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Citizen        : int  40725 147695 20714 17495 42345 8057 15581 88612 26462
20600 ...
##  $ Income         : num  51281 50254 32964 38678 45813 ...
##  $ IncomeErr      : num  2391 1263 2973 3995 3141 ...
##  $ IncomePerCap   : int  24974 27317 16824 18431 20532 17580 18390 21374 21071
21811 ...
##  $ IncomePerCapErr: int  1080 711 798 1618 708 2055 714 489 1366 1556 ...
##  $ Poverty        : num  12.9 13.4 26.7 16.8 16.7 24.6 25.4 20.5 21.6 19.2 ...
##  $ ChildPoverty   : num  18.6 19.2 45.3 27.9 27.2 38.4 39.2 31.6 37.2 30.1 ...
##  $ Professional   : num  33.2 33.1 26.8 21.5 28.5 18.8 27.5 27.3 23.3 29.3 ...
##  $ Service        : num  17 17.7 16.1 17.9 14.1 15 16.6 17.7 14.5 16 ...
##  $ Office         : num  24.2 27.1 23.1 17.8 23.9 19.7 21.9 24.2 26.3 19.5 ...
##  $ Construction   : num  8.6 10.8 10.8 19 13.5 20.1 10.3 10.5 11.5 13.7 ...
##  $ Production     : num  17.1 11.2 23.1 23.7 19.9 26.4 23.7 20.4 24.4 21.5 ...
##  $ Drive          : num  87.5 84.7 83.8 83.2 84.9 74.9 84.5 85.3 85.1 83.9 ...
##  $ Carpool        : num  8.8 8.8 10.9 13.5 11.2 14.9 12.4 9.4 11.9 12.1 ...
##  $ Transit        : num  0.1 0.1 0.4 0.5 0.4 0.7 0 0.2 0.2 0.2 ...
##  $ Walk           : num  0.5 1 1.8 0.6 0.9 5 0.8 1.2 0.3 0.6 ...
##  $ OtherTransp    : num  1.3 1.4 1.5 1.5 0.4 1.7 0.6 1.2 0.4 0.7 ...
##  $ WorkAtHome     : num  1.8 3.9 1.6 0.7 2.3 2.8 1.7 2.7 2.1 2.5 ...
##  $ MeanCommute    : num  26.5 26.4 24.1 28.8 34.9 27.5 24.6 24.1 25.1 27.4 ...
##  $ Employed       : int  23986 85953 8597 8294 22189 3865 7813 47401 13689 1015
5 ...
##  $ PrivateWork    : num  73.6 81.5 71.8 76.8 82 79.5 77.4 74.1 85.1 73.1 ...
##  $ PublicWork     : num  20.9 12.3 20.8 16.1 13.5 15.1 16.2 20.8 12.1 18.5 ...
##  $ SelfEmployed   : num  5.5 5.8 7.3 6.7 4.2 5.4 6.2 5 2.8 7.9 ...
##  $ FamilyWork     : num  0 0.4 0.1 0.4 0.4 0 0.2 0.1 0 0.5 ...
##  $ Unemployment   : num  7.6 7.5 17.6 8.3 7.7 18 10.9 12.3 8.9 7.9 ...
```

```r
library(corrplot)
library(caret)
library(MASS)
library(glmnet)
library(knitr)
library(kableExtra)

# read in the data
county_dat = read.csv('acs2015_county_data.csv')
full_data = county_dat
# drop NAs
unique(unlist(lapply(county_dat, function(x) which(is.na(x)))))
county_dat = county_dat[-c(549, 2674),]

#hist(county_dat$Income, breaks = 50, main = 'Income Distribution', xlab = "Incom
e") # hist of income
#abline(v = mean(county_dat$Income, na.rm = TRUE), col = 'red')
#abline(v = median(county_dat$Income, na.rm = TRUE), col = 'blue')
#legend("topright", legend = c("Mean","Median"),lty = 1, col = c("red","blue"))

par(mfrow = c(1,2))
poverty = c()
for (i in unique(county_dat$State)){
  poverty = c(poverty, mean(county_dat[county_dat$State == i,]$Poverty))
}
#plot(poverty, main = "Poverty by State",ylab="% in Poverty")

income = c()
for (i in unique(county_dat$State)){
  income = c(income, mean(county_dat[county_dat$State == i,]$Income))
}
#plot(income, main = "Income by State", ylab="Income")


par(mfrow = c(1,1))

county_dat = county_dat[,-c(1,3,32)]
county_dat$Men = county_dat$Men/county_dat$TotalPop
county_dat$Women = county_dat$Women/county_dat$TotalPop
M = cor(county_dat[,-c(1)])
#corrplot(M, method="circle")

set.seed(432)
county_idx = createDataPartition(county_dat$Income, p = 0.75, list = FALSE)
county_trn = county_dat[county_idx, ]
county_tst = county_dat[-county_idx, ]

# IncomeErr, IncomePerCap, IncomePerCapErr
```

```
county_dat = county_dat[,-which(names(county_dat) %in% c('Men','Pacific', 'OtherTr
ansp', 'Office', 'IncomeErr', 'IncomePerCap', 'IncomePerCapErr'))]

county_dat$Citizen = county_dat$Citizen/county_dat$TotalPop
county_dat = county_dat[,-c(1)]
county_dat = county_dat[!(county_dat$State == 'Puerto Rico'),]

# BIC
fwd_lm_bic = step(lm(Income ~ 1, data = county_trn), direction = 'forward', scope
 = formula(lm(Income ~ ., data = county_trn)), k = log(nrow(county_trn)))

bwd_lm_bic = step(lm(Income ~ ., data = county_trn), direction = 'backward', k = l
og(nrow(county_trn)))

step_lm_bic = step(lm(Income ~ 1, data = county_trn), direction = 'both', scope =
 formula(lm(Income ~ ., data = county_trn)), k = log(nrow(county_trn)))

step_inter_lm_bic = step(lm(Income ~ 1, data = county_trn), direction = 'both', sc
ope = formula(lm(Income ~ (Poverty + Professional + Citizen + MeanCommute + SelfEm
ployed + Asian + Walk + Hispanic + Native + PrivateWork + TotalPop + Women + Carpo
ol)^2, data = county_trn)), k = log(nrow(county_trn)))

step_poly_bic = step(lm(Income ~ 1, data = county_trn), direction = 'both', scope
 = formula(lm(Income ~ Poverty + Professional + Citizen + MeanCommute + SelfEmploy
ed + Asian + Walk + Hispanic + Native + PrivateWork + TotalPop + Women + Carpool +
 I(Poverty^2) + I(Professional^2) + I(Citizen^2) + I(MeanCommute^2) + I(SelfEmploy
ed^2) + I(Asian^2) + I(Walk^2) + I(Hispanic^2) + I(Native^2) + I(PrivateWork^2) +
 I(TotalPop^2) + I(Women^2) + I(Carpool^2), data = county_trn)), k = log(nrow(coun
ty_trn)))

calc_rmse = function(actual, predicted) {
  sqrt(mean((actual - predicted) ^ 2))
}

get_best_result = function(caret_fit) {
  best = which(rownames(caret_fit$results) == rownames(caret_fit$bestTune))
  best_result = caret_fit$results[best, ]
  rownames(best_result) = NULL
  best_result
}

results = data.frame(
  method = c("Linear Forward/Stepwise Selection", "Linear backward Selection", "Li
near Model with Interactions", "Polynomial model", "KNN Unscaled", "KNN Scaled",
"Ridge Model", "Lasso Model"),
  cv = c(
    get_best_result(fwd_lm)$RMSE,
    get_best_result(bwd_lm)$RMSE,
    get_best_result(inter_lm)$RMSE,
    get_best_result(poly_fit)$RMSE,
    get_best_result(knn_unscaled_mod)$RMSE,
    get_best_result(knn_scaled_mod)$RMSE,
    #get_best_result(rf_mod)$RMSE,
    ridge_cv_rmse,
    lasso_cv_rmse
  ),
  test = c(
    calc_rmse(county_tst$Income, predict(fwd_lm, county_tst)),
    calc_rmse(county_tst$Income, predict(bwd_lm, county_tst)),
    calc_rmse(county_tst$Income, predict(inter_lm, county_tst)),
    calc_rmse(county_tst$Income, predict(poly_fit, county_tst)),
    calc_rmse(county_tst$Income, predict(knn_unscaled_mod, county_tst)),
    calc_rmse(county_tst$Income, predict(knn_scaled_mod, county_tst)),
    #calc_rmse(county_tst$Income, predict(rf_mod, county_tst)),
    ridge_rmse,
    lasso_rmse
  )
)
colnames(results) = c("Method", "CV RMSE", "Test RMSE")
kable_styling(kable(results, format = "html", digits = 2), full_width = FALSE)
```