MACS 33002 Introduction to Machine Learning
Final Report

The Income Inequality by Opportunity in the U.S. and their Roots

(Members and Main Contributions)

- **John Kim**
  Literature review, theoretical background research, exploratory data analysis, coding for age subgroup analysis, and drafting report

- **Takahiro Minami**
  Literature review, theoretical background research, data cleaning, exploratory data analysis, and coding for whole US analysis, robustness check, and drafting report

- **Xin Feng**
  Data cleaning, exploratory data analysis, coding for state level analysis, and drafting report

- **Yujiao Song**
  Data cleaning and drafting report

I. Motivation and Introduction

Income inequality in the United States had reached its peak level in 50 years, and it is still growing (Ingraham, 2019). Setting aside the scathing critique from the humanitarian and social justice sphere on severe inequality (Sen, 2000), a line of research corroborates the negative impact of income inequality on economic growth (Herzer & Vollmer, 2012; Cingano, 2014). This negative relationship is conspicuous when income inequality is entailed by inequality of opportunity instead of individuals' efforts (Marrero & Rodríguez, 2013).

In light of this view, we adopt Roemer's (1998) conception of inequality of opportunity to differentiate the causes of income inequality by social circumstances and individuals' effort. We assess the social circumstances that contribute to the current income inequality in the U.S. by utilizing the conditional inference tree method. After we analyze the structure of income inequality in the U.S. by drawing a regression tree, we elaborate the heterogeneity of inequality of opportunity across U.S. states and age groups by comparing trees across subgroups and estimating the opportunity-base Gini-coefficient measuring income inequality associated with social circumstances.

II. Theoretical Background

John Roemer attributes an individual's outcome to two types of factors: effort (factors over which individuals have control) and circumstances (environmental factors beyond one's control, such as biological characteristics). Over the past decade, scholars have debated the appropriate empirical methods to estimate inequality of opportunity (Ramos & Van de gaer, 2012). Based on them, we define inequality of opportunity as a divergence of average income between social groups with different social circumstances. For example, when the average income of Group A (White, Male, US citizen) and Group B (Black, Female, non-US citizen) are

different, we think that this income difference is associated with the difference of social circumstances (race, gender, and citizenship) and regard it as inequality of opportunity.

Traditional research has developed many regression models to classify the population based on social circumstances and measure income inequality by opportunity. However, the fundamental problem of the regression methods is that researchers have to decide the predictors and structure of interaction terms in their model a priori. Due to the arbitrariness in model selection, regression methods have concerns about estimation biases (Donni et al., 2015).

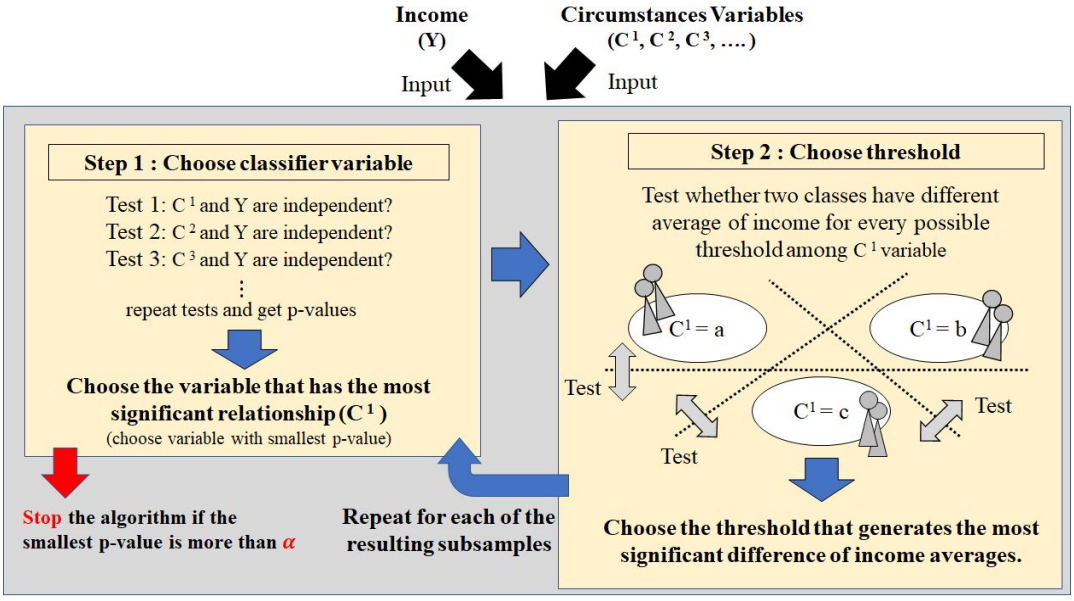
To overcome the arbitrary model selection problem, we adapt conditional inference tree algorithm, which was suggested by Hothorn et al. (2006) and applied to estimate inequality of opportunity in EU countries by Brunori et al. (2018). This regression tree method has several advantages over traditional estimations. First, the model makes no assumption on the underlying relationship between the predictors and the dependent variable. Second, conditional inference tree method will perform a sequence of hypothesis tests to prevent model overfitting, thus reducing the upward bias on the inequality of opportunity estimates. Third, the conditional inference tree is visually appealing and it can provide a more intuitive understanding of the structure of inequality of opportunities (Brunori et al., 2018).

III. Conditional Inference Tree

The conditional inference tree algorithm conducts statistical tests to check whether the output (income; Y) depends on each of predictors (social circumstances; C_p) by comparing unconditional and conditional distributions of income ($f(Y)$ and $f(Y|C_p)$), and chooses the predictor that is most strongly related with income (Step 1 in Figure 1). Then, the algorithm divides the population into two classes by all possible splits among the predictor and tests whether the binary classes have significantly different income averages. Based on the p-values

of these tests, the split associating with the biggest income discrepancy is chosen (Step 2 in Figure 1). This whole process is repeated for each of the resulting subgroups until all remaining predictors are independent of income in the subgroup. Through this process, the algorithm builds the tree to structurize the factors contributing income divergence in each of the subgroups.

(**Figure 1**: Mechanism of Conditional Inference Tree Algorithm)



In the tree algorithm, we need to set a maximum p-value α to decide when the machine should stop repeating the process. Once all p-values generated by independence tests for predictors (C_p) and output (Y) are larger than the threshold α, the algorithm stops generating the tree. A lower α increases the possibility to miss some eligible classifiers while a higher α increases the possibility to misidentify meaningless variables as classifiers.

To deal with this trade-off between type 1 and type 2 errors, we conduct cross-validation to choose optimal α. We will divide the original sample into 10, then calculate MSE using each subgroup based on the estimation from the other 9 subsamples (10-fold CV). We will repeat this

process for 19 possible α from 0.01 to 0.1, and choose the optimal α based on MSE. We conduct this cross-validation for the whole US, each state, and each age group respectively.

IV. Empirical Strategy

1. Data

We use microdata from the Current Population Survey 2019 Annual Social and Economic (ASEC) Supplement. ASEC is the nation-wide survey conducted by the U.S. Census Bureau; it contains demographic and socio-economic data for subjective individuals (180,101 obs.) and households (944,633). Since we use household income and individual level predictors, we match household and individual databases and find 68,345 households whose members are also included in individual dataset. We only use the data of householders who are over 18 years old representing each of these households which sharply reduces the size of our dataset. Our cleaned data has 58,269 individual observations.

Our outcome variable is the total household income in the unit of 1,000 dollars. The input variables include social circumstance characteristics such as sex, mother's place of birth, father's place of birth, race, citizenship, etc. Table 1 contains all the variables we use in our modeling. See Appendix for the summary statistics of the whole U.S. and each state.

(**Table 1**: The Variable List)

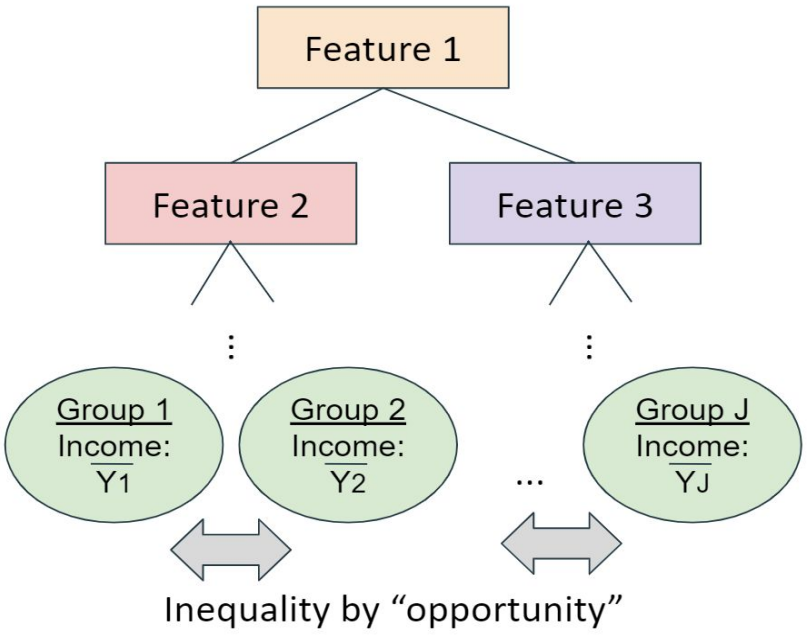
Output variable		
	Total Household Income	Annual income ($1,000)
Category variable		
	State	50 States + D.C.
	Age Group	18-34 / 35-49 / 50-64
Circumstances variable		
1	Sex	Male / Female
2	Hispanic	Yes / No
3	Race	White / Black / Other
4	Citizenship	- Citizen born in US - Citizen born out of US - Non-Citizen
5	Place of Birth	US / out of US
6	Mother's Place of Birth	US / out of US
7	Father's Place of Birth	US / out of US

2. Estimation of Income inequality by Opportunity

Using the conditional inference tree, we classify household income by various circumstances, including gender, race, and citizenship. The resulting tree is shown in Figure 2. The features at nodes are social circumstances playing a significant role in explaining the variation of income between the following subgroups. The height of the node indicates the relevant significance in income inequality.

The terminal nodes at the bottom of the tree represent groups within which individuals share the same social circumstances (i.e. opportunity is equal). We can think that income differences between these groups are associated with differences of social circumstances (i.e. opportunities), so we define income inequality by opportunity as the income difference between terminal nodes. On the other hand, the variation of income within each terminal node can be attributed to inequality by efforts although it is not our focus in this research.

(**Figure 2**: Expected Output Tree and Income Inequality by Opportunity)



Note: $\overline{Y_J}$ represents the average income of the terminal node (Group J)

In addition to visualizing the structure of income inequality by opportunity, the conditional inference tree algorithm allows us to measure the degree of inequality of opportunity numerically by Gini-coefficient. We calculate the average income for each of terminal nodes and replace individual income by the average income of their representing terminal nodes. Using the replaced income level, we calculate the Gini-coefficient following the usual Gini-coefficient formula. Mathematically, we calculate

$$G_{opportunity} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |\bar{Y}_{p[i]} - \bar{Y}_{q[j]}|}{2n^2\mu}$$

where, n is the total number of individuals in the population, $\bar{Y}_{p[i]}$ is the average income of group p which individual i belongs to, and μ is the overall average income of the population. Since the divergence of income between terminal nodes are only associated with social circumstances, this Gini-coefficient can capture the degree of income inequality by opportunity. We call this type of Gini-coefficient "opportunity-based Gini-coefficient."

V. Results

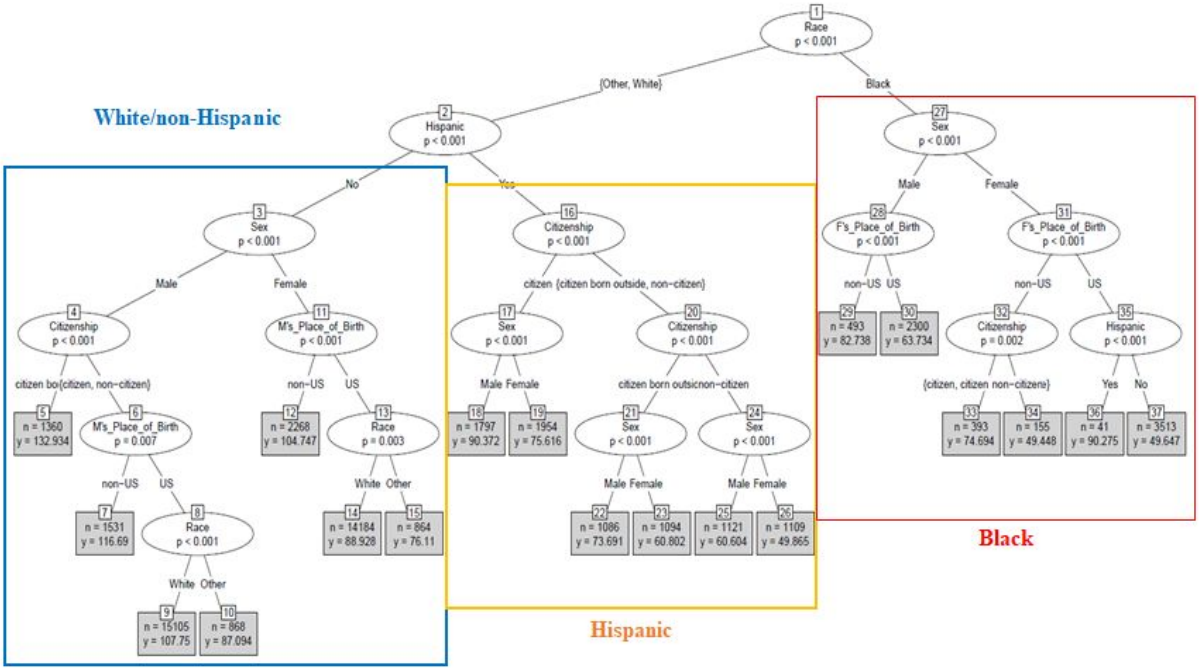
1. The Whole US Analysis

Figure 3 illustrates the tree for the whole U.S. sample generated by the conditional inference tree algorithm. Among various social circumstances, "race" and "hispanic" sit at the top of the tree, which implies that racial and ethnical income inequality is the most severe in U.S. Based on these factors, the tree can be divided into three sub-tree; white/non-hispanic, white/hispanic, and black. For white/non-hispanic and black group, "sex" is the next feature to split their tree, which suggests the existance of strong gender inequality. On the other hand, for hispanic group, "citizenship" comes first rather than "sex." It means that hispanic people who

can get U.S. citizenship have advantage over hispanic without citizenship, and the inequality associated with citizenship is stronger than the one by gender. A potential reason for this finding is that some of hispanic immigrants from Mexico or South America are not eligible to get U.S. citizenship and face difficulties.

In addition, the tree shows that the mother's place of birth or the father's place of birth influence the income of white/non-hispanic and black people although their impacts are minor. Interestingly, the so-called second-generation who have parents born outside the U.S. tend to earn more income than those whose parents were born in the U.S.

(**Figure 3**: The Whole U.S. Tree)

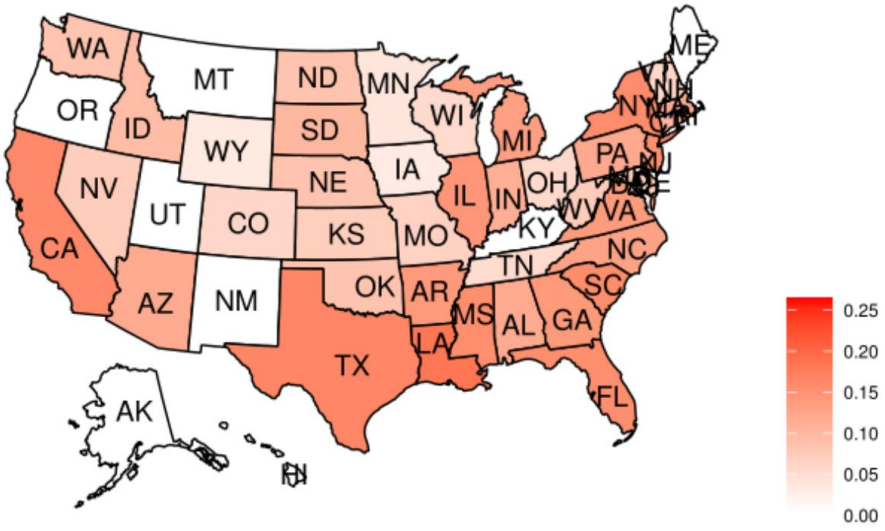


2. Analysis for States

We run our model for each of 50 states and the District of Columbia and calculate opportunity-based Gini-coefficient using average income of terminal nodes groups for each state. Figure 4 is the map of opportunity-based Gini-coefficient. The states with darker red have higher opportunity-based Gini-coefficient, implying that larger income inequality by

opportunity exists. The figure shows the geographical heterogeneity of inequality level across U.S. The east and west coast states and states at the southern border show more severe inequality than states in the middle.

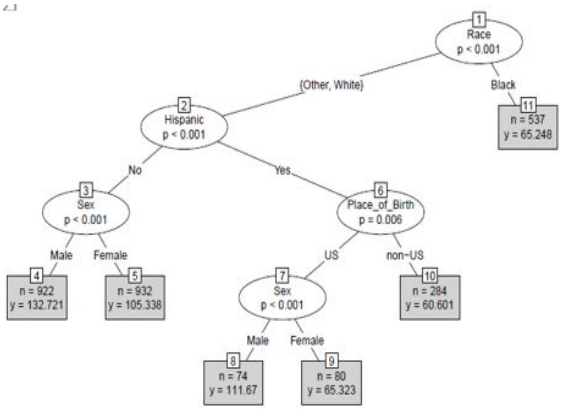
(**Figure 4**: Opportunity-based Gini-Coefficient Map by States)



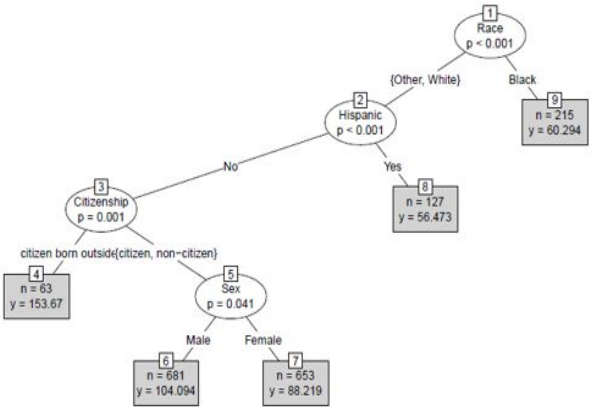
Figures 5 and 6 show the state tree of two states from the eastern region : New York and Pennsylvania. It illustrates that race is at the top of each tree and there is a significant inequality between white and black people. This characteristic is commonly observed in other states in this region. Moreover, it is also common that in east and midwest states the sub-tree of black people has no additional nodes. This indicates that the inequality black people are suffering from is severe, but it's structure is simple.

Figures 7 and 8 show the state tree of Texas and Arizona representing the southwest region. Instead of race, ethnicity (hispanic or not) is commonly at the top of trees for these states. One potential explanation is that these states are geographically close to Mexico and they have more hispanic immigrants.

(**Figure 5**: New York State Tree)

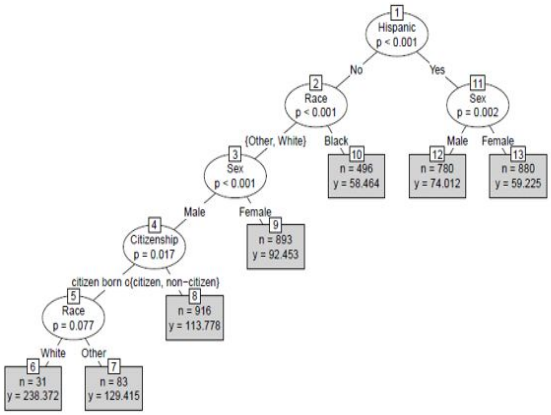


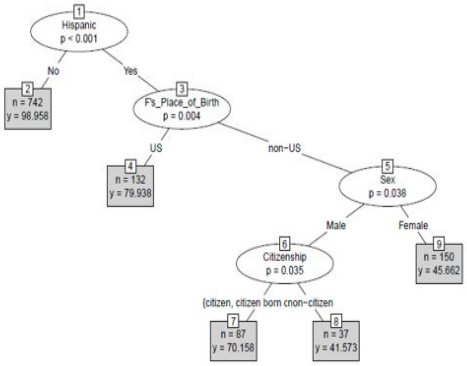
(**Figure 6:** Pennsylvania State Tree)



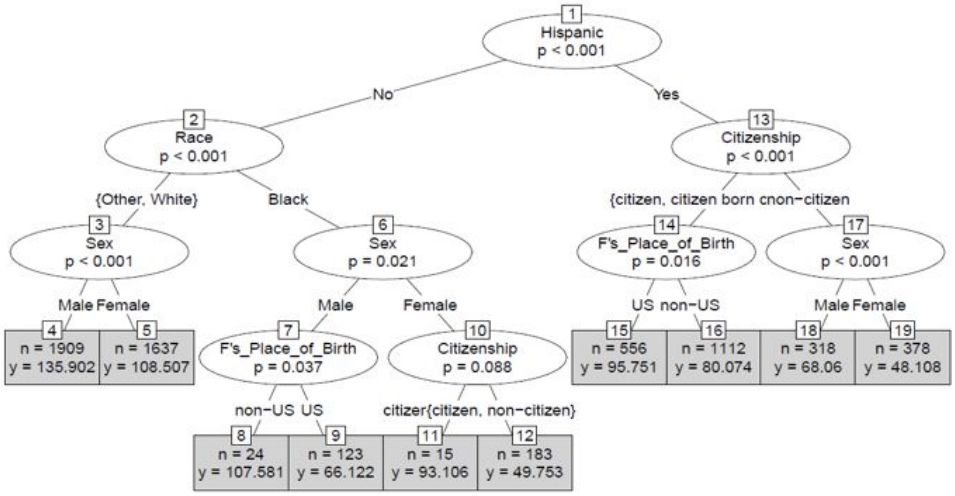
(**Figure 7**: Texas State Tree)

(**Figure 8**: Arizona State Tree)





(**Figure 9**: California State Tree)

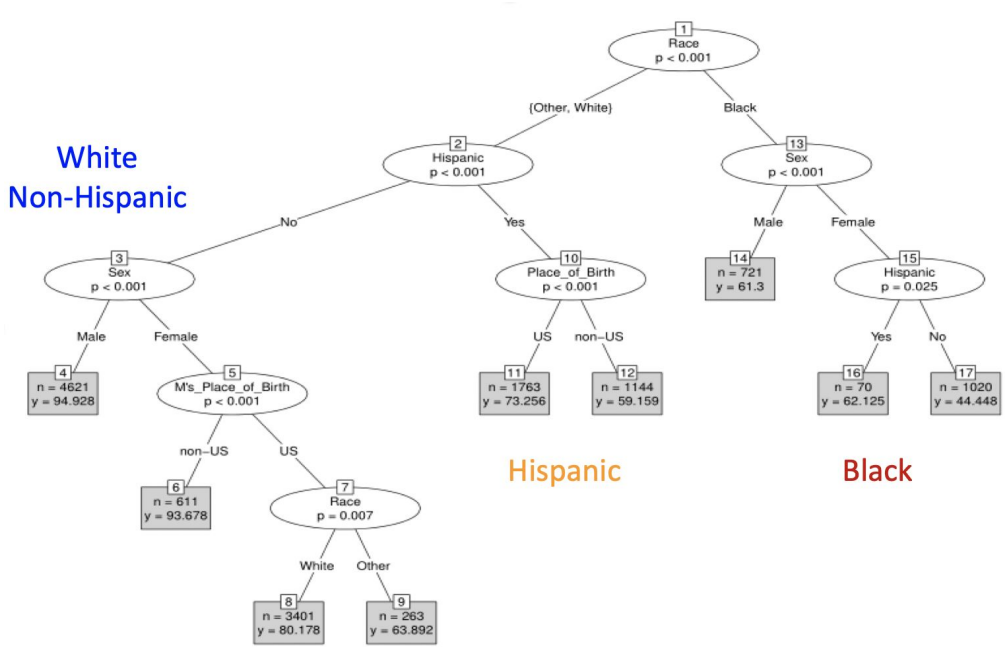


Among all 51 states, California has the most complicated structure of inequality (Figure 9). Ethnicity is at the top of the tree as in other southern states, but both hispanic and non-hispanic people suffer from inequality by gender, citizenship, or parent's place of birth even among their ethnicity group. While California is one of the largest economic hubs in the U.S. along with states on the east, the structure of inequality is unique and complex compared. We don't have a certain clue to judge this reason, but the distinctive industry structure and diversity of population may generate this multilayer inequality in California.

3. Analysis for Age Groups

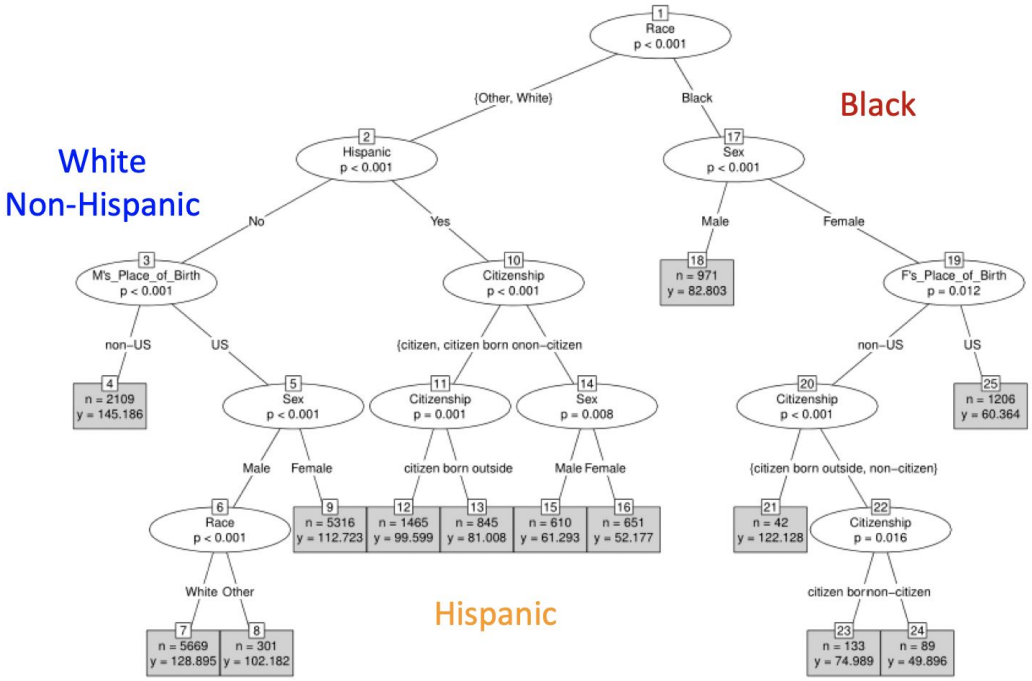
We conducted a subgroup analysis by dividing the whole dataset into three age groups. Because age can have a substantial impact on income level (e.g., through accumulated skillsets or knowledge), we can observe the correlation with more precision by controlling age. This is a common technique in economics to investigate income inequality by setting different age groups (see Chetty, 2017, for example).

(**Figure 10**: Age 18-34 Tree)



Figures 10 to 12 show the trees for age group 18-34, 35-49, and 50-64 respectively. Although there exists a slight difference in the complexity or the order of nodes in the figures, we can observe an analogous pattern across the different age groups. The 'race' variable appears on the top node, followed either by 'hispanic' or 'sex' (or 'citizenship' for the eldest group), dividing each tree into three clusters: White (non-Hispanic), Hispanic, and Black. White (non-hispanic) group has the highest mean income, followed by Hispanic U.S. citizens and then Black. The result is in line with the real data published by the US Census Bureau (2018): White, non-hispanic ($68,145), Hispanic ($50,486), Black ($40,258).

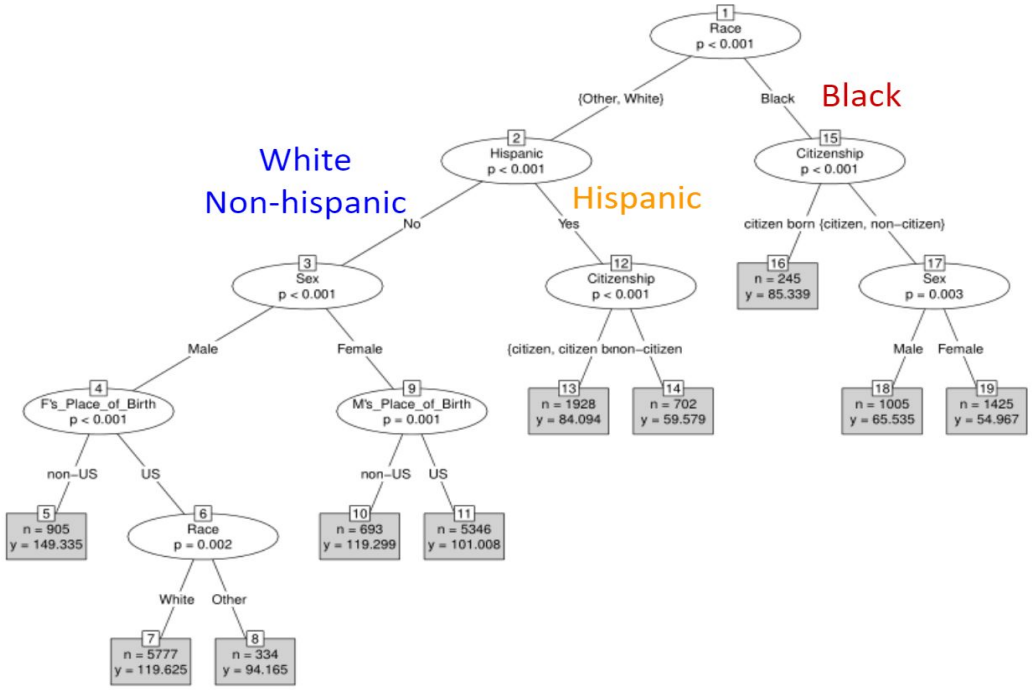
(**Figure 11**: Age 35-49 Tree)



Throughout the figures, we can see that either 'place of birth' or 'citizenship' appeared as significant variables for the Hispanic group. If we assume that the 'place of birth' variable is equivalent to the 'citizenship' variable, due to the policy of universal birthright citizenship in the U.S., we can conclude that the most determinant factor for the Hispanic group's income is

citizenship status. Data from Pew Research Center (2017) show possible explanations behind the trend. There exists a gap between the U.S. born hispanics and the hispanics not born in the U.S. in English proficiency (90% v.s. 36%) and educational attainment (college or higher: 53% v.s. 29%) which may systematically create the income gap.

(**Figure 12**: Age 49-64 Tree)



For White group, on the other hand, the 'sex' and 'parents' place of birth' variables are critical. Individuals whose parents are born outside the U.S. have larger income compared to *leaves* in the tree for the two older groups. Although it might be hasty to conclude a trend along the timeline with cross-sectional data, if we assume that we can observe at least some cumulative effect throughout the time by delving into different age groups, we could argue that sex-induced income inequality gap is closing while that of race is widening. This tendency is in line with the the critiques that affirmative action programs have focused mostly on white female instead of minor ethnicities in the U.S. throughout history (Williams & Cooper, 2019).

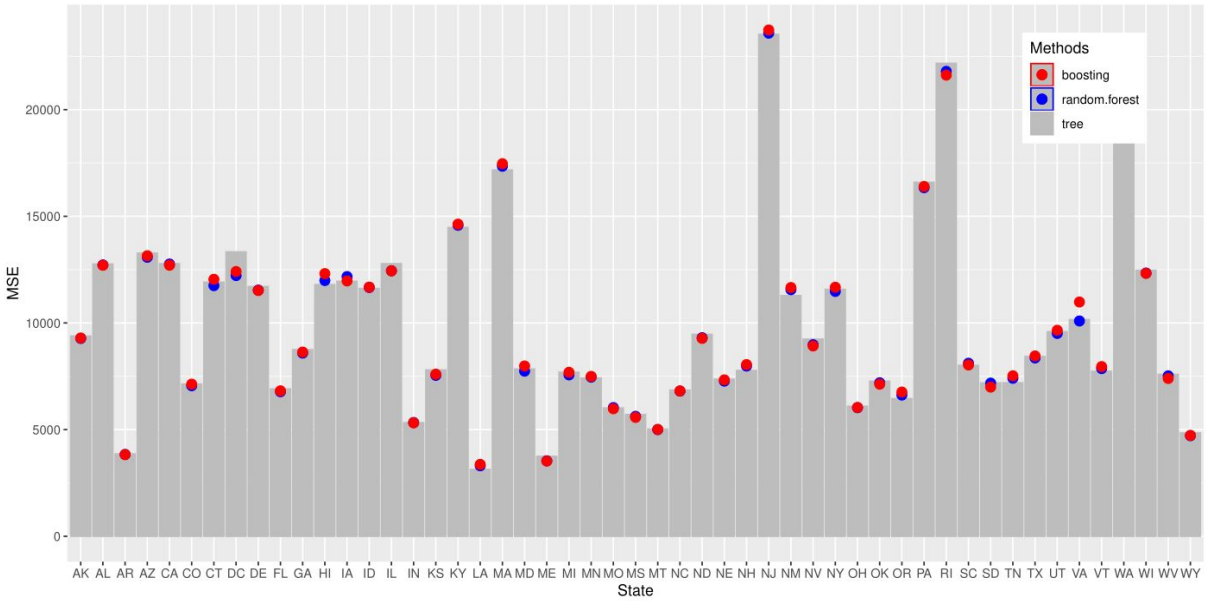
VI. Discussion

Although our conditional inference tree method is a powerful tool to estimate income inequality and visualize it, we acknowledge that the tree method may be less accurate than other ensemble methods. To check the robustness of our result, we run random forest and boosting methods for the whole U.S. sample, state level, and age subgroup analysis. As Table 2 and Figures 13 to14 show, we confirm that MSE of our baseline estimation (tree method) is very close to MSEs of the other two methods. This makes us more confident about the precision of our baseline model.

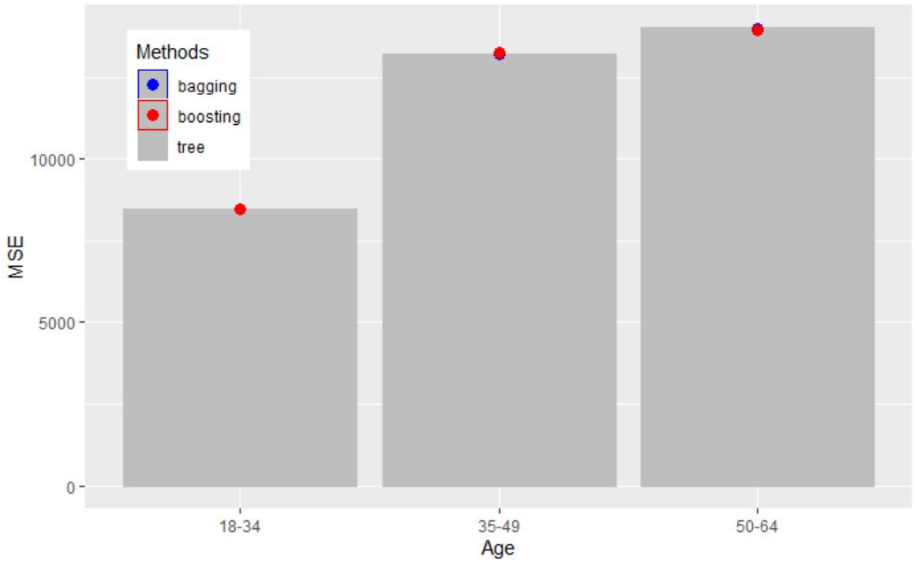
(**Table 2**: Comparison of MSE)

tree	random.forest	boosting
11,116	11,140	10,711

(**Figure 13**: Comparison of MSE)



(**Figure 14**: Comparison of MSE)



Another potential pitfall of our estimation strategy is that some effort features may be correlated with circumstance features. Although we assume effort and opportunity affects income level independently, it may not be the case in the real world. In that case, the income differences between terminal node groups are associated with differences in both opportunity and effort, and we cannot estimate pure inequality by opportunity in our strategy. This problem is also related to the definition of opportunity features and effort features. Acknowledging these issues are theoretically important research topics for the future, we follow John Roemer's simple binary division in this paper as many previous empirical literature did.

VII. Conclusion

In this research, we divide income inequality into inequality associated with opportunity and effort, and explore the structure behind the inequality by opportunity in U.S.. Using conditional inference tree to classify household income for the whole country, states, and age subgroups, we find three key findings in our analysis.

First, the analysis at country level shows that race and ethnicity play the most essential roles to generate income inequality in the U.S. As expected, white/non-hispanic people earn the highest income on average, and hispanic and black people are behind. Among these three racial/ethnic groups, the factors contributing to inequality are different. Gender matters for white/non-hispanic and black groups while citizenship is more critical for hispanic people. From policy perspectives, our research makes a great contribution by elaborating the complex structure of inequality among social minorities and prioritizing social factors causing inequality.

Second, from state level analysis, we find there exist heterogeneous characteristics on the inequality structure based on the spatial allocation of the states. In brief, in east states, race (white/black) tends to be the most significant factor to contribute to income inequality, while ethnicity (whether Hispanic or not) is more crucial in southwestern states with many Hispanic immigrants. Although further research is necessary to specify causal relationship, this result implies that the social policy to mitigate income inequality should prioritize different minorities depending on geographic and social characteristics of states.

Lastly, a significant difference is not found among inequality structures across different age groups. Here, we found citizenship as a critical feature for the Hispanic group in income, whereas sex and parents' place of birth play a substantial role for the White non-Hispanic group. Also, we suggest that there is a trend that income gap between sexes is reducing while that among races is enhancing as people get older. From this result, it is confirmed that the racial minorities are suffering from income inequality consistently across generations.

References

Brunori, P., Hufe, P., & Mahler, D. G. (2018). The roots of inequality: estimating inequality of opportunity from regression trees. The World Bank.

Chetty, R., Grusky, D., Hell, M., Hendren, N., Manduca, R., & Narang, J. (2017). The fading American dream: Trends in absolute income mobility since 1940. Science, 356(6336), 398-406.

Cingano, F. (2014). Trends in income inequality and its impact on economic growth. Paris: OECD.

Donni, P. L., Rodríguez, J. G., & Dias, P. R. (2015). Empirical definition of social types in the analysis of inequality of opportunity: a latent classes approach. Social Choice and Welfare, 44(3), 673-701.

Herzer, D., & Vollmer, S. (2012). Inequality and growth: evidence from panel cointegration. The Journal of Economic Inequality, 10(4), 489-503.

Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. Journal of Computational and Graphical Statistics, 15(3), 651-674.

Ingraham, C. (2019, October 9). For the first time in history, U.S. billionaires paid a lower tax rate than the working class last year. The Washington Post. Retrieved from: https://www.washingtonpost.com/business/2019/10/08/first-time-history-us-billionaires-paid-lower-tax-rate-than-working-class-last-year/

Pew Research Center. (2017). Pew Research Center tabulations of 2017 American Community Surveys (1% IPUMS). (Accessed on 11 March 2020) https://www.pewresearch.org/hispanic/fact-sheet/latinos-in-the-u-s-fact-sheet/.

Ramos, X., & Van de Gaer, D. (2012). Empirical approaches to inequality of opportunity: Principles, measures, and evidence.

Roemer, J. E. (1998). Equality of opportunity. Cambridge, MA: Harvard University Press.

Sen, A. K. (2000). Social justice and distribution of income. In A.B. Atkinson & F. Bourguignon (Eds.), Handbooks in economics, 16, 59-86.

U.S. Census Bureau. (2018). Current population survey, 1968 to 2018 annual social and economic supplements. (Accessed on 11 March 2020). https://www.census.gov/content/dam/Census/library/visualizations/2018/demo/p60-263/figure1.pdf.

Williams, D. R., & Cooper, L. A. (2019). Reducing racial inequities in health: Using what we already know to take action. International journal of environmental research and public health, 16(4), 606.

Appendix (Summary Table)

state	n	ave.income	25%tile	75%tile	s.d.	Gini	White ratio	Male ratio	Hispanic ratio
Whole	68,315	90.3	31.7	114.0	106.2	0.481	77.0	50.1	16.4
AK	782	94.0	36.1	128.2	92.4	0.452	69.2	55.1	6.6
AL	1,399	70.0	25.0	92.0	85.9	0.480	65.7	49.5	4.7
AR	1,269	70.4	25.7	92.2	74.0	0.466	77.4	49.5	6.8
AZ	1,148	85.8	30.3	111.3	91.5	0.474	85.6	51.6	35.4
CA	6,255	102.4	35.0	128.0	123.1	0.491	74.1	51.3	37.8
CO	867	102.7	42.0	129.5	99.1	0.445	89.3	51.8	21.6
CT	690	105.8	34.2	130.9	145.0	0.499	80.7	48.1	20.0
DC	1,280	131.4	32.3	172.6	158.1	0.531	48.0	43.7	11.6
DE	726	95.0	36.8	121.5	114.1	0.461	68.2	52.9	7.9
FL	3,291	78.7	27.0	100.0	97.8	0.491	78.4	49.3	30.1
GA	1,572	79.5	25.0	100.2	96.0	0.505	54.5	48.3	10.7
HI	1,029	108.3	40.0	137.8	125.9	0.469	26.9	53.6	6.8
IA	780	89.4	36.6	113.1	87.0	0.420	90.5	53.5	7.6
ID	1,052	79.3	34.0	102.1	84.8	0.435	95.2	54.6	14.4
IL	1,830	100.2	37.0	126.6	113.8	0.477	77.2	53.6	18.5
IN	1,091	80.4	32.2	105.0	80.4	0.439	84.0	53.3	8.7
KS	851	89.1	35.0	111.7	108.1	0.452	86.7	48.6	11.8
KY	754	81.8	26.0	99.9	118.8	0.518	87.9	51.5	3.4
LA	1,648	72.4	23.5	90.8	89.9	0.496	61.6	46.0	6.5
MA	1,354	120.5	39.3	160.0	127.8	0.483	80.7	48.3	12.6
MD	837	111.3	46.5	149.2	95.7	0.432	57.6	49.5	10.4
ME	522	78.4	28.0	107.5	65.7	0.437	92.1	49.8	1.7
MI	1,524	86.1	32.6	109.2	93.8	0.460	79.1	48.9	4.1
MN	872	100.6	39.3	132.6	100.3	0.446	86.7	51.5	4.5
MO	897	86.9	32.5	115.0	99.9	0.455	81.4	48.3	5.0
MS	1,322	60.8	20.2	76.5	78.4	0.492	54.6	43.9	2.8
MT	1,157	77.1	30.5	99.2	74.1	0.439	93.3	53.2	4.4
NC	1,655	77.4	28.2	97.6	88.3	0.484	68.3	49.6	9.6
ND	947	90.0	36.1	109.8	107.3	0.455	88.0	54.0	3.6
NE	811	91.4	34.0	117.4	83.8	0.446	89.1	48.2	13.8
NH	819	106.3	45.3	139.8	91.5	0.414	94.1	54.1	3.5
NJ	1,359	115.4	35.2	143.6	143.6	0.514	74.1	51.2	21.0
NM	1,349	75.5	24.0	88.0	145.7	0.521	85.1	46.6	50.1
NV	944	84.4	33.0	106.0	99.1	0.464	73.8	51.4	25.4
NY	2,829	101.2	32.0	127.5	122.4	0.511	69.1	46.9	19.5
OH	1,605	86.9	31.0	111.8	92.9	0.464	83.7	50.1	3.2
OK	1,036	77.3	30.0	100.3	78.4	0.453	77.0	51.5	12.5
OR	989	93.9	38.1	120.2	94.6	0.445	87.6	50.5	12.4
PA	1,739	91.0	32.0	114.7	107.9	0.473	82.2	49.8	7.8
RI	621	95.1	32.4	124.8	108.2	0.492	84.2	43.5	17.9
SC	1,167	79.0	28.9	99.4	89.8	0.480	65.4	49.5	5.1
SD	766	86.9	31.4	108.4	99.1	0.462	86.7	47.8	4.3
TN	1,338	80.4	31.3	102.6	101.7	0.463	78.0	49.9	5.8
TX	4,079	84.3	30.0	106.0	102.2	0.487	79.6	49.6	40.7
UT	965	101.3	47.1	125.0	100.1	0.416	93.2	50.6	15.6
VA	1,266	107.5	38.6	140.0	114.8	0.470	67.9	52.1	11.0
VT	841	88.0	36.4	118.5	76.2	0.420	94.4	49.8	2.4
WA	1,242	109.0	42.2	135.3	123.3	0.458	79.1	55.3	13.4
WI	883	84.5	35.5	110.0	89.4	0.427	88.9	47.1	6.0
WV	1,351	69.4	25.0	90.1	79.7	0.458	92.5	47.7	1.9
WY	915	83.7	34.0	107.5	83.2	0.431	94.6	54.9	10.2

Appendix (Code)

```
library(knitr)
library(kableExtra)
library(gridExtra)
library(tree)
library(rsample)
library(party)
```

```
## Warning: package 'party' was built under R version 3.6.3
```

```
library(reldist)
library(skimr)
library(gbm)
library(randomForest)
library(tidyverse)
```

Data Cleaning

Import Dataset

```
# Household data
hh <- read.csv("hhpub19.csv", header=T)

# Person data
pp <- read.csv("pppub19.csv", header=T)

# state db
fips = read.csv("fipscodes.csv", header = T)
```

Add State names to Household database

```
# matching state name and fips code
hh <- left_join(hh,fips,by="GESTFIPS")
head(hh$GESTFIPS)
```

```
## [1] 23 23 23 23 23 23
```

```
head(hh$state_full)
```

```
## [1] MAINE MAINE MAINE MAINE MAINE MAINE
## 55 Levels: ALABAMA ALASKA AMERICAN SAMOA ARIZONA ARKANSAS ... WYOMING
```

Matching household and people database by Household ID (H_SEQ and PH_SEQ)

```
# matching key between person data and household
#PH_SEQ is the sequence number of family record in household
```

```r
match_hh_per <- unique(pp$PH_SEQ)

# hh database we use
new_hh <- hh %>%  filter(H_SEQ %in% match_hh_per)

# pp database we use
# we use only householders
new_pp <- pp %>% filter(HHDREL==1)

# match hh and pp
df <- inner_join(new_hh, new_pp, by=c("H_SEQ" = "PH_SEQ"))
```

Select all variables we use

```r
##Variables:
#HTOTVAL: household income
#A_SEX: 1 = Male, 2 = Female
#A_AGE
#PRCITSHP: 1 = Native, born in US;
          #2 = Native, born in PR or US outlying area
               #/ Native, born abroad of US parent(s)
               #/ Foreign born, US cit by naturalization;
          #3 = Foreign born, not a US citizen
#PRDTRACE: 1 = White only; 2 = Black only; 3 = Others
#PEHSPNON: Are you Spanish, Hispanic, or Latino?  1 = Yes 2 = No
#PENATVTY: Place of Birth 1 = U.S., 2 = Outside of U.S.
#PEFNTVTY: Father's Place of Birth 1 = U.S., 2 = Outside of U.S.
#PEMNTVTY: Mother's Place of Birth 1 = U.S., 2 = Outside of U.S.
#GESTFIPS
#state
#state_full

df1 <- dplyr::select (df,
                      HTOTVAL, A_SEX, A_AGE, PRCITSHP, PRDTRACE, PEHSPNON,
                      PENATVTY, PEFNTVTY, PEMNTVTY, GESTFIPS, state, state_full)
```

Data Cleaning

```r
#There is no NA data in this dataset.

#PRCITSHP:
for (i in 1:length(df1$PRCITSHP)) {
    if (df1$PRCITSHP[i] == 3 | df1$PRCITSHP[i] == 4){
      df1$PRCITSHP[i] = 2
    }
    else if (new_pp$PRCITSHP[i] == 5){
      df1$PRCITSHP[i] = 3
    }
}

#PRDTRACE:
for (i in 1:length(df1$PRDTRACE)) {
```

```
    if(!(df1$PRDTRACE[i] == 1 | df1$PRDTRACE[i] == 2)) {
      df1$PRDTRACE[i] = 3
    }
}

#PENATVTY: Place of Birth 1 = U.S., 2 = Outside of U.S.
df1$PENATVTY <- ifelse(df1$PENATVTY == 57, 1, 2)

##PEFNTVTY: Father's Place of Birth 1 = U.S., 2 = Outside of U.S.
df1$PEFNTVTY <- ifelse(df1$PEFNTVTY == 57, 1, 2)

#PEMNTVTY: Mother's Place of Birth 1 = U.S., 2 = Outside of U.S.
df1$PEMNTVTY <- ifelse(df1$PEMNTVTY == 57, 1, 2)
```

```
colnames(df1) <- c("HHincome","Sex","A_Age", "Citizenship","Race","Hispanic",
                   "Place_of_Birth","F_Place_of_Birth","M_Place_of_Birth",
                   "GESTFIPS","state","state_full")
df1$HHincome <- as.integer(df1$HHincome)/1000
df1$Sex <- as.factor(ifelse(df1$Sex==1,"Male","Female"))
df1$Citizenship <- as.factor(ifelse(df1$Citizenship==1,"citizen",
                   ifelse(df1$Citizenship==2,"citizen born outside","non-citizen")))
df1$Race <- as.factor(ifelse(df1$Race==1, "White",
            ifelse(df1$Race==2,"Black","Other")))
df1$Hispanic <- as.factor(ifelse(df1$Hispanic==1,"Yes","No"))
df1$Place_of_Birth <- as.factor(ifelse(df1$Place_of_Birth==1,"US","non-US"))
df1$`F_Place_of_Birth` <- as.factor(ifelse(df1$`F_Place_of_Birth`==1,"US","non-US"))
df1$`M_Place_of_Birth` <- as.factor(ifelse(df1$`M_Place_of_Birth`==1,"US","non-US"))
head(df1)
```

```
##   HHincome    Sex A_Age Citizenship  Race Hispanic Place_of_Birth
## 1   18.000   Male    21     citizen White       No             US
## 2   21.780 Female    85     citizen White       No             US
## 3   12.000 Female    61     citizen White       No             US
## 4   22.727 Female    73     citizen White       No             US
## 5   20.954 Female    80     citizen White       No             US
## 6   55.000 Female    53     citizen White       No             US
##   F_Place_of_Birth M_Place_of_Birth GESTFIPS state state_full
## 1               US               US       23    ME      MAINE
## 2               US               US       23    ME      MAINE
## 3               US               US       23    ME      MAINE
## 4               US           non-US       23    ME      MAINE
## 5               US               US       23    ME      MAINE
## 6               US               US       23    ME      MAINE
```

Summary Statistics

```
sum_table.whole <- df1 %>%
  summarize(state="Whole",
            n=n(),
            ave.income=round(mean(HHincome),1),
            `25%tile`=round(quantile(HHincome,0.25),1),
            `75%tile`=round(quantile(HHincome,0.75),1),
            `s.d.`=round(sd(HHincome),1),
```

```
            `Gini`=round(gini(HHincome),3),
            `White ratio`=round(mean(I(Race=="White"))*100,1),
            `Male ratio`=round(mean(I(Sex=="Male"))*100,1),
            `Hispanic ratio`=round(mean(I(Hispanic=="Yes"))*100,1))

sum_table.state <- df1 %>%
  group_by(state)%>%
  summarise(n=n(),
            ave.income=round(mean(HHincome),1),
            `25%tile`=round(quantile(HHincome,0.25),1),
            `75%tile`=round(quantile(HHincome,0.75),1),
            `s.d.`=round(sd(HHincome),1),
            `Gini`=round(gini(HHincome),3),
            `White ratio`=round(mean(I(Race=="White"))*100,1),
            `Male ratio`=round(mean(I(Sex=="Male"))*100,1),
            `Hispanic ratio`=round(mean(I(Hispanic=="Yes"))*100,1))

sum_table <- as.data.frame(rbind(sum_table.whole,sum_table.state))
sum_table$n <- formatC(sum_table$n, format="d", big.mark=',')
kable(sum_table, align=rep('c', 10),booktabs = T)
```

state	n	ave.income	25%tile	75%tile	s.d.	Gini	White ratio	Male ratio	Hispanic ratio
Whole	68,315	90.3	31.7	114.0	106.2	0.481	77.0	50.1	16.4
AK	782	94.0	36.1	128.2	92.4	0.452	69.2	55.1	6.6
AL	1,399	70.0	25.0	92.0	85.9	0.480	65.7	49.5	4.7
AR	1,269	70.4	25.7	92.2	74.0	0.466	77.4	49.5	6.8
AZ	1,148	85.8	30.3	111.3	91.5	0.474	85.6	51.6	35.4
CA	6,255	102.4	35.0	128.0	123.1	0.491	74.1	51.3	37.8
CO	867	102.7	42.0	129.5	99.1	0.445	89.3	51.8	21.6
CT	690	105.8	34.2	130.9	145.0	0.499	80.7	48.1	20.0
DC	1,280	131.4	32.3	172.6	158.1	0.531	48.0	43.7	11.6
DE	726	95.0	36.8	121.5	114.1	0.461	68.2	52.9	7.9
FL	3,291	78.7	27.0	100.0	97.8	0.491	78.4	49.3	30.1
GA	1,572	79.5	25.0	100.2	96.0	0.505	54.5	48.3	10.7
HI	1,029	108.3	40.0	137.8	125.9	0.469	26.9	53.6	6.8
IA	780	89.4	36.6	113.1	87.0	0.420	90.5	53.5	7.6
ID	1,052	79.3	34.0	102.1	84.8	0.435	95.2	54.6	14.4
IL	1,830	100.2	37.0	126.6	113.8	0.477	77.2	53.6	18.5
IN	1,091	80.4	32.2	105.0	80.4	0.439	84.0	53.3	8.7
KS	851	89.1	35.0	111.7	108.1	0.452	86.7	48.6	11.8
KY	754	81.8	26.0	99.9	118.8	0.518	87.9	51.5	3.4
LA	1,648	72.4	23.5	90.8	89.9	0.496	61.6	46.0	6.5
MA	1,354	120.5	39.3	160.0	127.8	0.483	80.7	48.3	12.6
MD	837	111.3	46.5	149.2	95.7	0.432	57.6	49.5	10.4
ME	522	78.4	28.0	107.5	65.7	0.437	92.1	49.8	1.7
MI	1,524	86.1	32.6	109.2	93.8	0.460	79.1	48.9	4.1
MN	872	100.6	39.3	132.6	100.3	0.446	86.7	51.5	4.5
MO	897	86.9	32.5	115.0	99.9	0.455	81.4	48.3	5.0
MS	1,322	60.8	20.2	76.5	78.4	0.492	54.6	43.9	2.8
MT	1,157	77.1	30.5	99.2	74.1	0.439	93.3	53.2	4.4
NC	1,655	77.4	28.2	97.6	88.3	0.484	68.3	49.6	9.6
ND	947	90.0	36.1	109.8	107.3	0.455	88.0	54.0	3.6
NE	811	91.4	34.0	117.4	83.8	0.446	89.1	48.2	13.8
NH	819	106.3	45.3	139.8	91.5	0.414	94.1	54.1	3.5
NJ	1,359	115.4	35.2	143.6	143.6	0.514	74.1	51.2	21.0
NM	1,349	75.5	24.0	88.0	145.7	0.521	85.1	46.6	50.1
NV	944	84.4	33.0	106.0	99.1	0.464	73.8	51.4	25.4
NY	2,829	101.2	32.0	127.5	122.4	0.511	69.1	46.9	19.5
OH	1,605	86.9	31.0	111.8	92.9	0.464	83.7	50.1	3.2
OK	1,036	77.3	30.0	100.3	78.4	0.453	77.0	51.5	12.5
OR	989	93.9	38.1	120.2	94.6	0.445	87.6	50.5	12.4
PA	1,739	91.0	32.0	114.7	107.9	0.473	82.2	49.8	7.8
RI	621	95.1	32.4	124.8	108.2	0.492	84.2	43.5	17.9
SC	1,167	79.0	28.9	99.4	89.8	0.480	65.4	49.5	5.1
SD	766	86.9	31.4	108.4	99.1	0.462	86.7	47.8	4.3
TN	1,338	80.4	31.3	102.6	101.7	0.463	78.0	49.9	5.8
TX	4,079	84.3	30.0	106.0	102.2	0.487	79.6	49.6	40.7
UT	965	101.3	47.1	125.0	100.1	0.416	93.2	50.6	15.6
VA	1,266	107.5	38.6	140.0	114.8	0.470	67.9	52.1	11.0
VT	841	88.0	36.4	118.5	76.2	0.420	94.4	49.8	2.4
WA	1,242	109.0	42.2	135.3	123.3	0.458	79.1	55.3	13.4
WI	883	84.5	35.5	110.0	89.4	0.427	88.9	47.1	6.0
WV	1,351	69.4	25.0	90.1	579.7	0.458	92.5	47.7	1.9
WY	915	83.7	34.0	107.5	83.2	0.431	94.6	54.9	10.2

Tree for the whole US

```
df1.whole <- df1 %>%
  select(HHincome,Sex,Citizenship,Race,Hispanic,
         Place_of_Birth,`F_Place_of_Birth`,`M_Place_of_Birth`)
head(df1.whole)
```

```
##   HHincome    Sex Citizenship  Race Hispanic Place_of_Birth F_Place_of_Birth
## 1   18.000   Male     citizen White       No             US               US
## 2   21.780 Female     citizen White       No             US               US
## 3   12.000 Female     citizen White       No             US               US
## 4   22.727 Female     citizen White       No             US               US
## 5   20.954 Female     citizen White       No             US               US
## 6   55.000 Female     citizen White       No             US               US
##   M_Place_of_Birth
## 1               US
## 2               US
## 3               US
## 4           non-US
## 5               US
## 6               US
```

Split the df into training and test set

```
# split the df into training and test set
set.seed(12345)
train <- sample(c(1:nrow(df1.whole)),nrow(df1.whole)*0.75,replace = FALSE)
test <- c(1:nrow(df1.whole))[!(c(1:nrow(df1.whole)) %in% train)]
df_train <- df1.whole[train,]
df_test <- df1.whole[test,]
```

Cross validation for deciding alpha. We search among potential alpha from 0.01 to 0.1.

```
# Develop function for CV (get MSE)

holdout_results <- function(splits,alpha) {
  # Fit the model to the training set

  mod <- ctree(HHincome ~ .,
               data=analysis(splits),
               control = ctree_control(mincriterion=1-alpha,
                                       testtype="Bonferroni"))
  # Get MSE based on test set
  test <- assessment(splits)
  mse <-
    mean((test$HHincome - predict(mod,newdata = test))^2)

  mse
}

# set potential alpha
set.seed(23456)
```

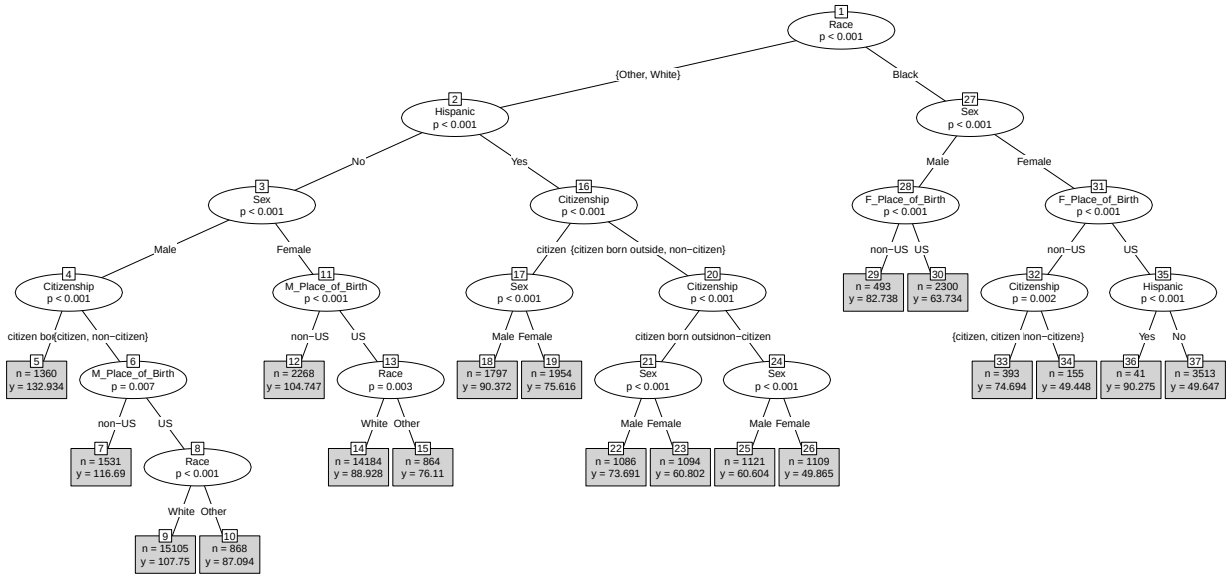
```
a <-  seq(0.01, 0.1, 0.005)
cv_result <- data.frame(alpha=a)

# repeat CV for each alpha
for (i in 1:length(a)) {

 ## 10 folded CV
cv10 <- vfold_cv(data = df_train,v = 10) %>%
  mutate(results = pmap(list(splits,a[i]),holdout_results)) %>%
  unnest(results)

cv_result$mse[i] <- mean(cv10$results)
}

# optimal alpha
opt_alpha <- cv_result$alpha[cv_result$mse==min(cv_result$mse)]
opt_alpha
```

```
## [1] 0.035
```

Draw tree

```
# Baseline case
citree <- ctree(HHincome ~ .,
                data=df_train,
                control = ctree_control(mincriterion=1-opt_alpha,
                                        testtype="Bonferroni"))
```

```
# Plot
plot(citree, type="simple")
```



MSE and Opportunity based Gini coefficient

```
# mse
mse <- mean((df_test$HHincome - predict(citree,newdata = df_test))^2)
mse
```

```
## [1] 11115.68
```

```
# opportunity based gini coefficient
gini_opp <- gini(predict(citree,newdata = df1.whole))
gini <- gini(df1.whole$HHincome)
kable(data.frame(gini.opp=gini_opp,gini=gini))
```

gini.opp	gini
0.1230618	0.4811

Robustness check

Conditional Inference Random Forest

```
ciforest <-
  partykit::cforest(HHincome ~ .,
          data=df_train,
            control = partykit::ctree_control(alpha=opt_alpha,
                                testtype="Bonferroni"))
```

```
#mse
memory.limit(size=10000)
mse_forest <- mean((df_test$HHincome -
                      predict(ciforest,newdata = df_test,type = "response"))^2)
```

Boosting

```
set.seed(1000)

boost <- gbm(HHincome ~ .,
                    data=df_train,
                    distribution="gaussian",
                    n.trees=1000,
                    shrinkage=0.01,
                    interaction.depth = 4)
test.pred <- predict(boost, newdata=df_test, n.trees=1000)
mse_boost <- mean((df_test$HHincome - test.pred)^2)
```

State-level Analysis

```
df1.state <- df1 %>%
  select(HHincome,Sex,Citizenship,Race,Hispanic,Place_of_Birth,
         `F_Place_of_Birth`,`M_Place_of_Birth`,state)
head(df1.state)
```

```
##   HHincome    Sex Citizenship  Race Hispanic Place_of_Birth F_Place_of_Birth
## 1   18.000   Male     citizen White       No             US               US
## 2   21.780 Female     citizen White       No             US               US
```

```
## 3   12.000 Female     citizen White       No             US               US
## 4   22.727 Female     citizen White       No             US               US
## 5   20.954 Female     citizen White       No             US               US
## 6   55.000 Female     citizen White       No             US               US
##   M_Place_of_Birth state
## 1               US    ME
## 2               US    ME
## 3               US    ME
## 4           non-US    ME
## 5               US    ME
## 6               US    ME
```

Split the df into training and test set for each state

```
# Generate subsample for each state
state_list <- vector(mode = "list", length = 51)
state_name <- unique(df1.state$state)
for (i in 1:51) {
    state_list[[i]] <-
      df1.state %>% filter(state == state_name[i]) %>%
      dplyr::select(-state)
}


# Split the df into training and test set for each state
train_length <- 51
test_length <- 51
train_list <- vector(mode = "list", length = 51)
test_list <- vector(mode = "list", length = 51)

set.seed(3000)
for (i in 1:train_length) {
  train_list[[i]]=sample(1:nrow(state_list[[i]]),
                         nrow(state_list[[i]])*0.75)

  test_list[[i]]=
    c(1:nrow(state_list[[i]]))[!(c(1:nrow(state_list[[i]])) %in% train_list[[i]])]
}
```

Cross validation for deciding alpha for each state

```
# function fot set potential alpha
a <-  seq(0.01, 0.1, 0.005)

cv_alpha <- function(df_train){
 cv_result <- data.frame(alpha=a)

 ## repeat CV for each alpha
 for (i in 1:length(a)) {
 ## 10 folded CV
 cv10 <- vfold_cv(data = df_train,v = 10) %>%
  mutate(results = pmap(list(splits,a[i]),holdout_results)) %>%
  unnest(results)
```

```
 cv_result$mse[i] <- mean(cv10$results)
 }

# optimal alpha
opt_alpha.state <- cv_result$alpha[cv_result$mse==min(cv_result$mse)]
opt_alpha.state
}

# CV for all states
set.seed(1234)
opt_alpha.state <- data.frame(state_group=c(1:51),state=state_name)
for (i in 1:train_length) {

  train <- train_list[[i]]
  opt_alpha.state$alpha[i] <- cv_alpha(state_list[[i]][train, ])
}
```

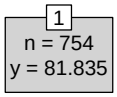
Using optimal alpha, estimate Tree of each state

```
# using only training set
tree_list.train <- vector(mode = "list", length = 51)
for (i in 1:length(tree_list.train)) {

 train <- train_list[[i]]
 citree <- ctree(HHincome ~ .,
                 data=state_list[[i]][train,],
                 control =
                   ctree_control(mincriterion=1-opt_alpha.state$alpha[i],
                                 testtype="Bonferroni"))

 tree_list.train[[i]] <- citree
}
```

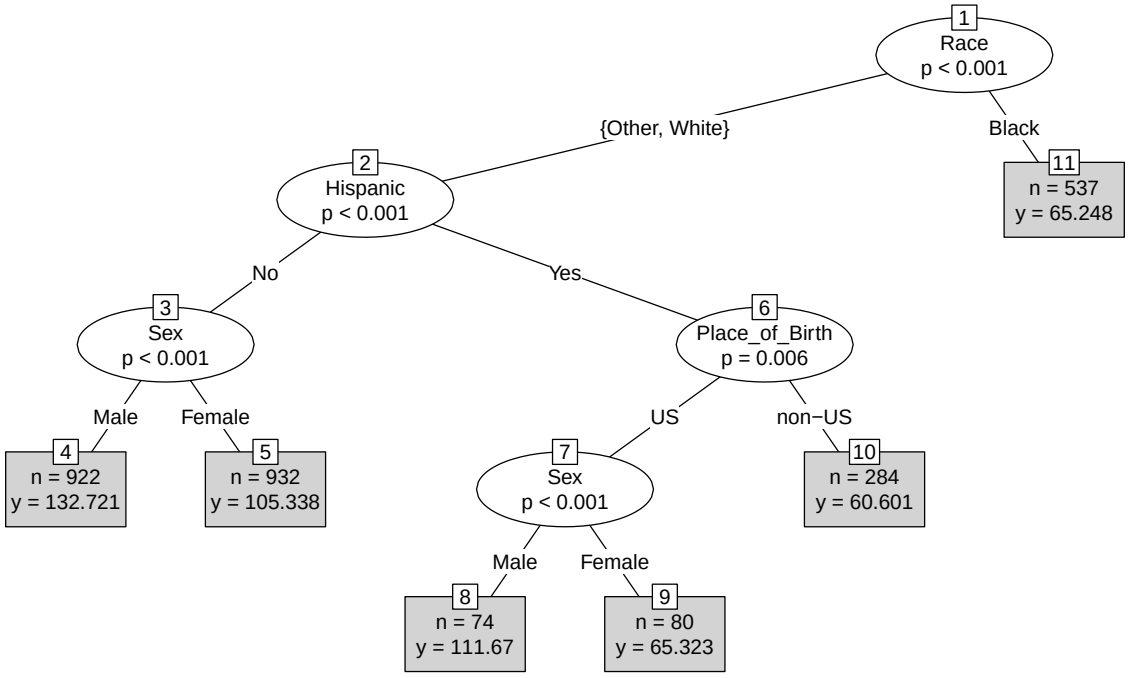
```
# use whole sample
tree_list <- vector(mode = "list", length = 51)
for (i in 1:length(tree_list)) {

 citree <- ctree(HHincome ~ .,
                 data=state_list[[i]],
                 control =
                   ctree_control(mincriterion=1-opt_alpha.state$alpha[i],
                                 testtype="Bonferroni"))

 tree_list[[i]] <- citree
}
```

Draw Trees for sample states

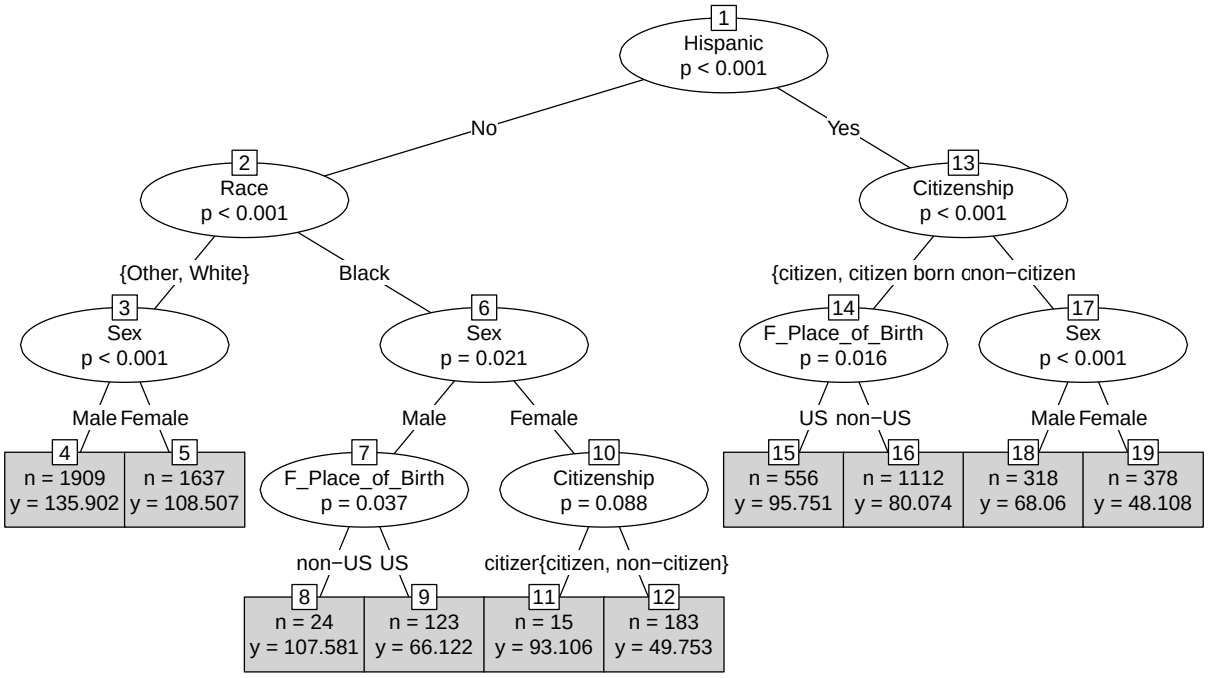
```
plot(tree_list[[31]],type="simple") #KY
```



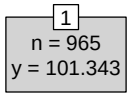

```
plot(tree_list[[7]],type="simple") #NY
```



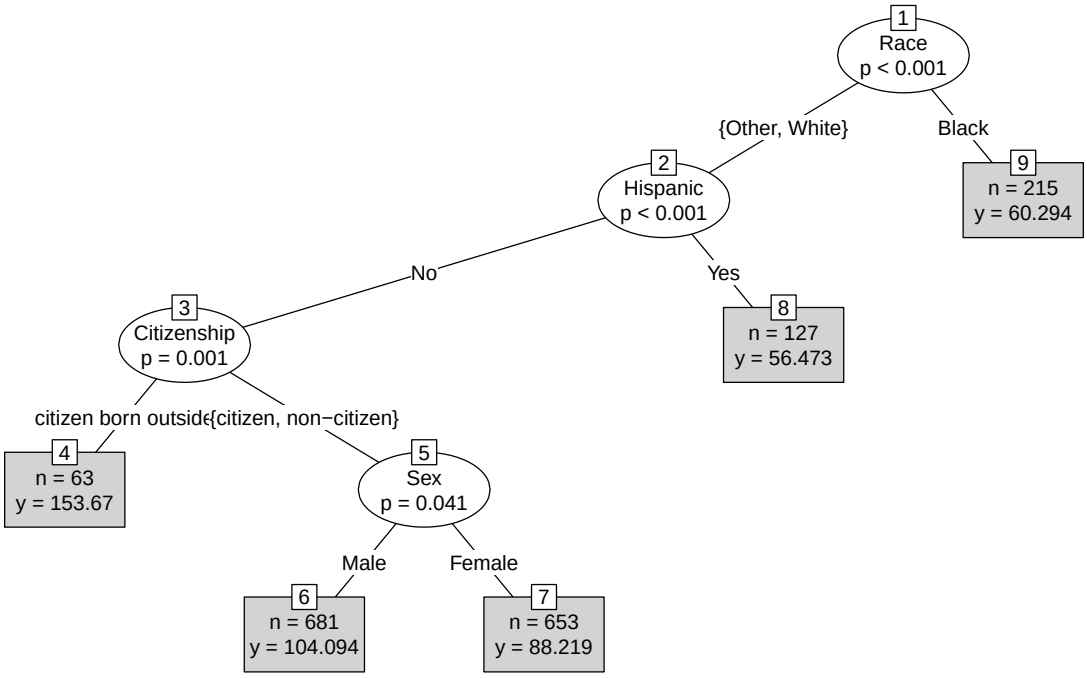

```
plot(tree_list[[49]],type="simple") #CA
```



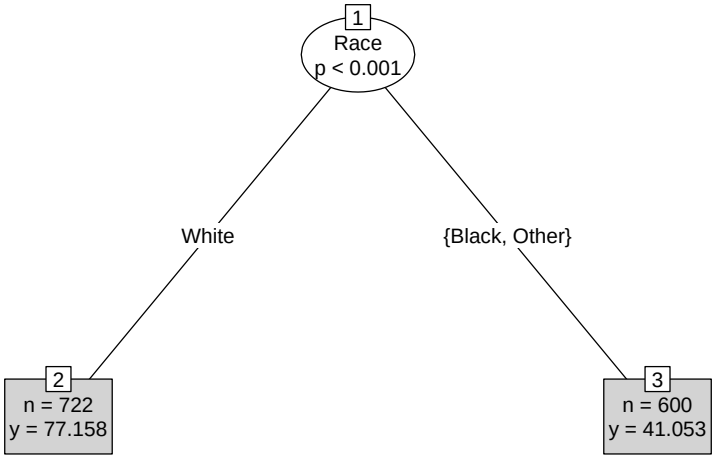

```
plot(tree_list[[45]],type="simple") #UT
```



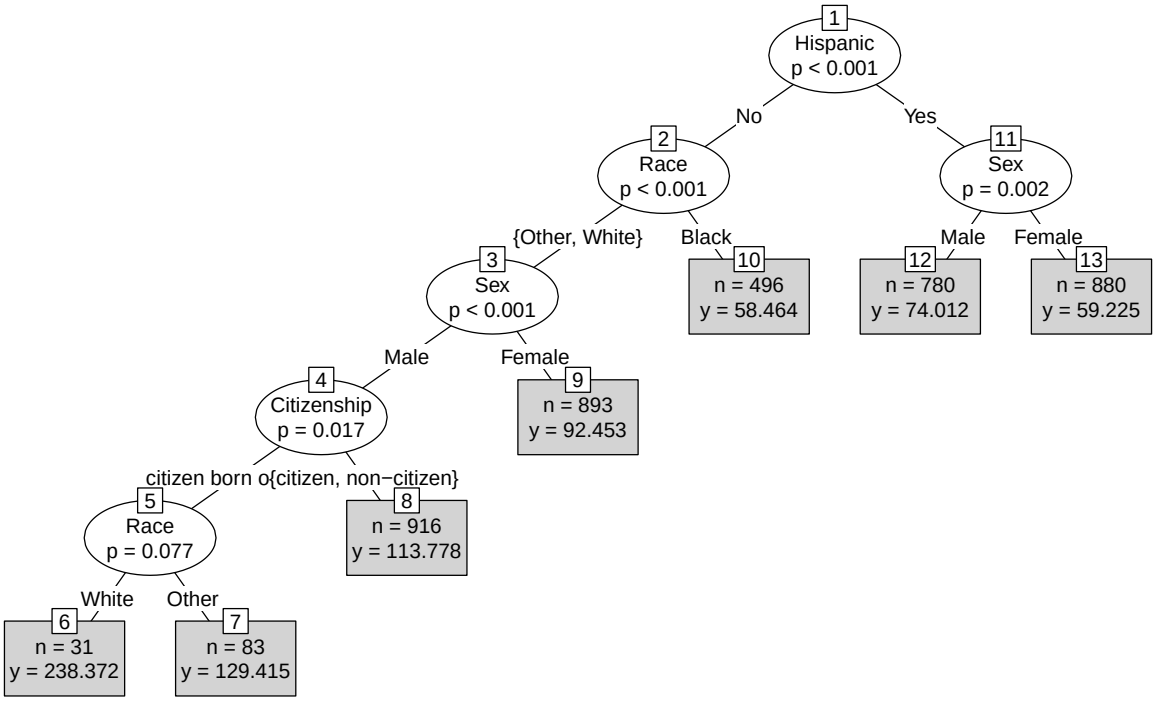

```
plot(tree_list[[9]],type="simple") #PA
```



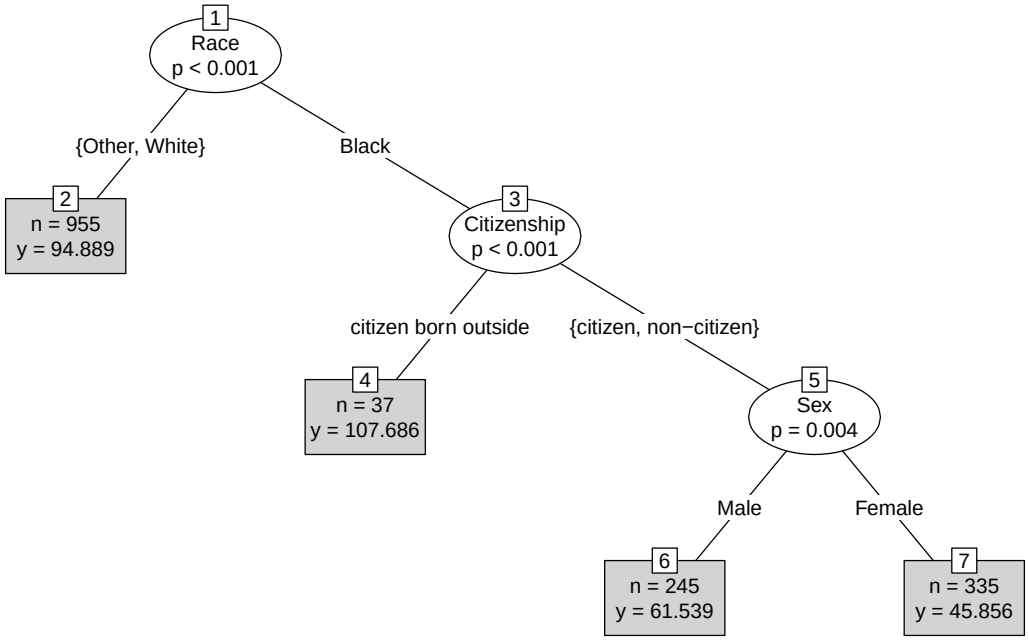

```
plot(tree_list[[34]],type="simple") #MS
```



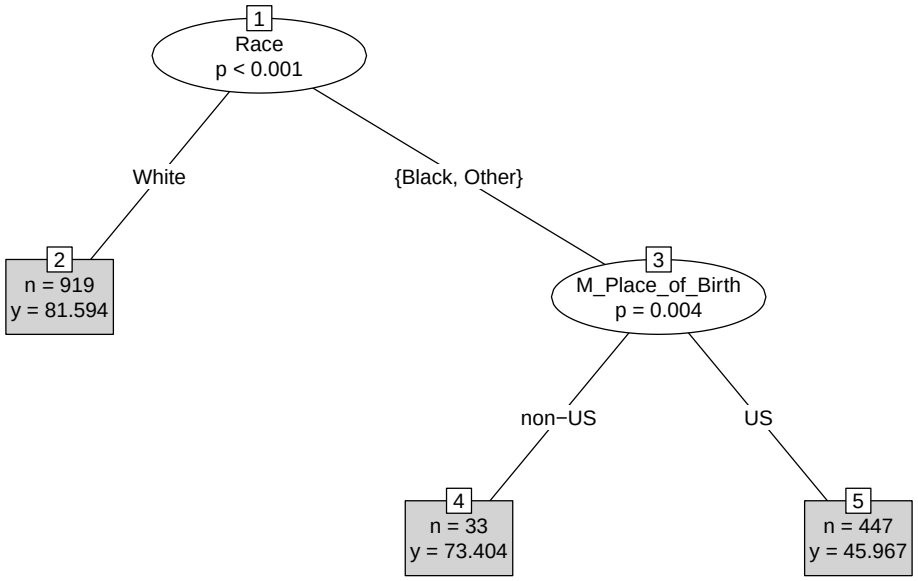

```
plot(tree_list[[38]],type="simple") #TX
```



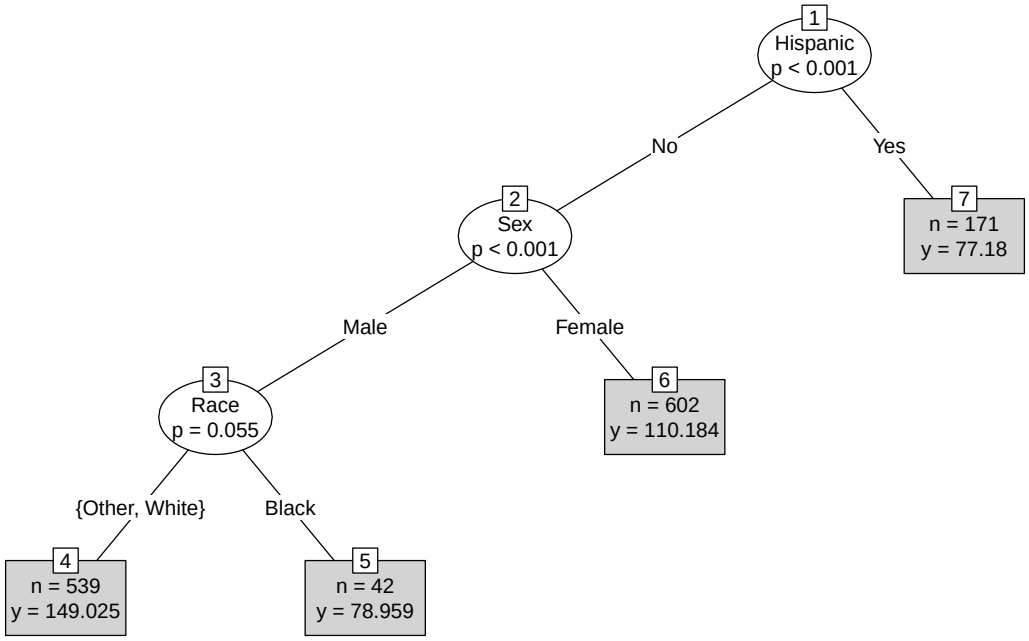

```
plot(tree_list[[29]],type="simple") #GA
```



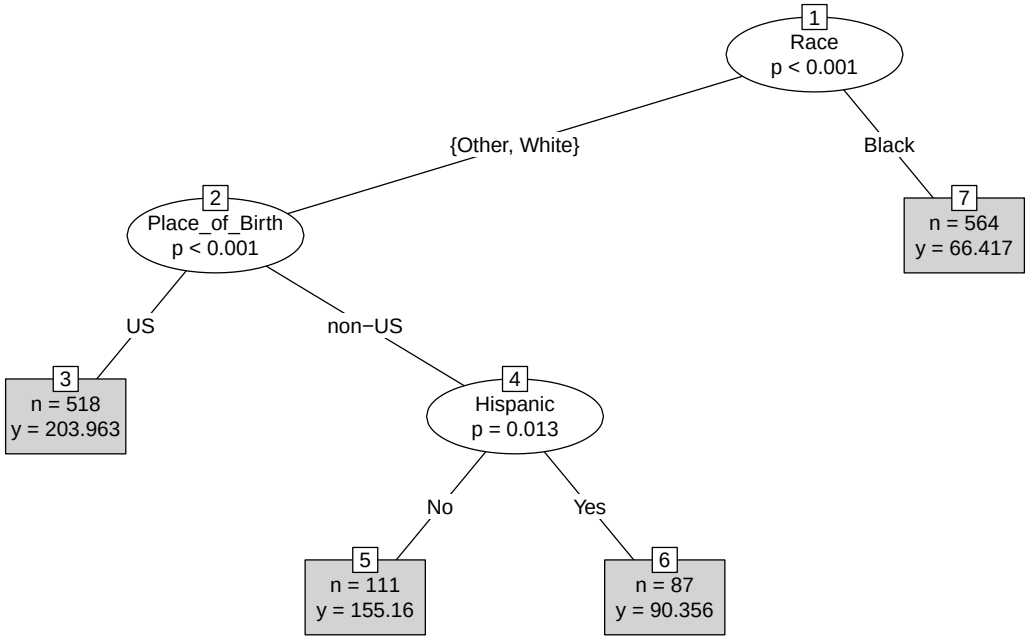

```
plot(tree_list[[33]],type="simple") #AL
```



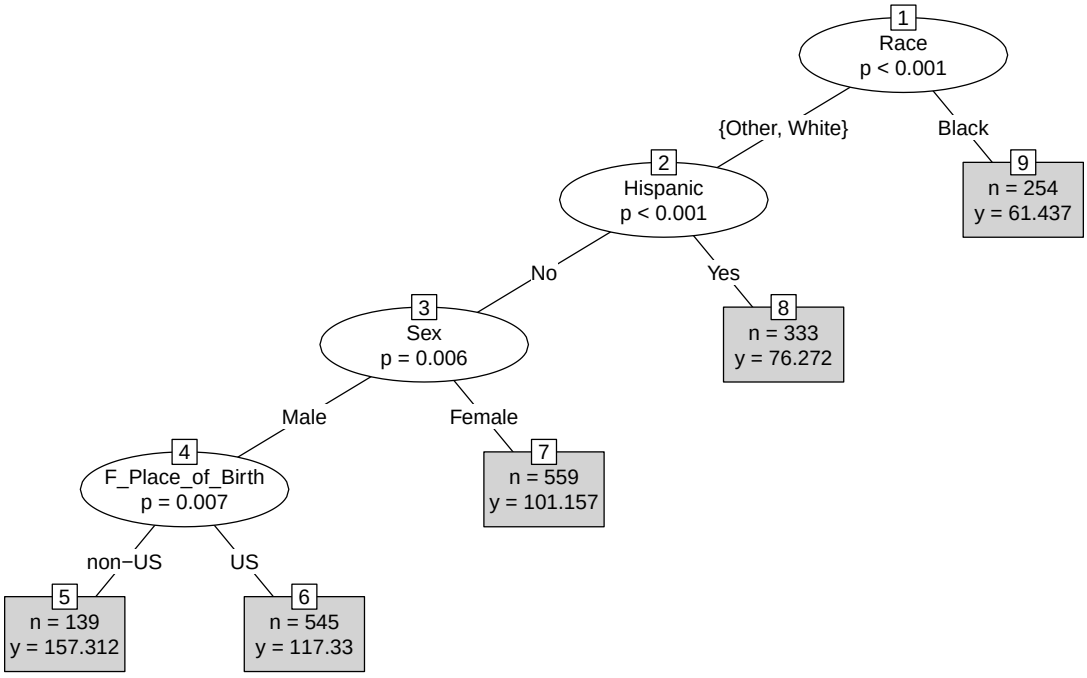

```
plot(tree_list[[4]],type="simple") #MA
```



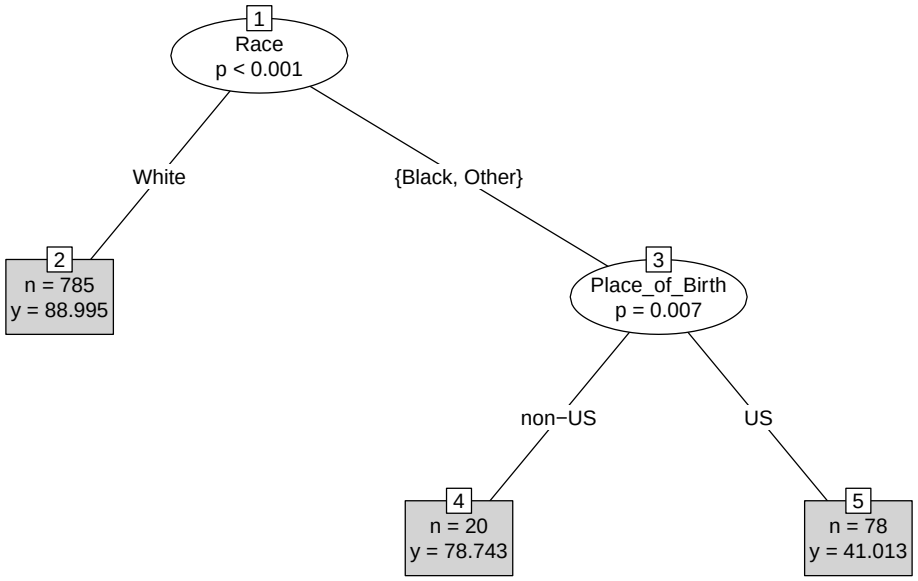

```
plot(tree_list[[24]],type="simple") #DC
```



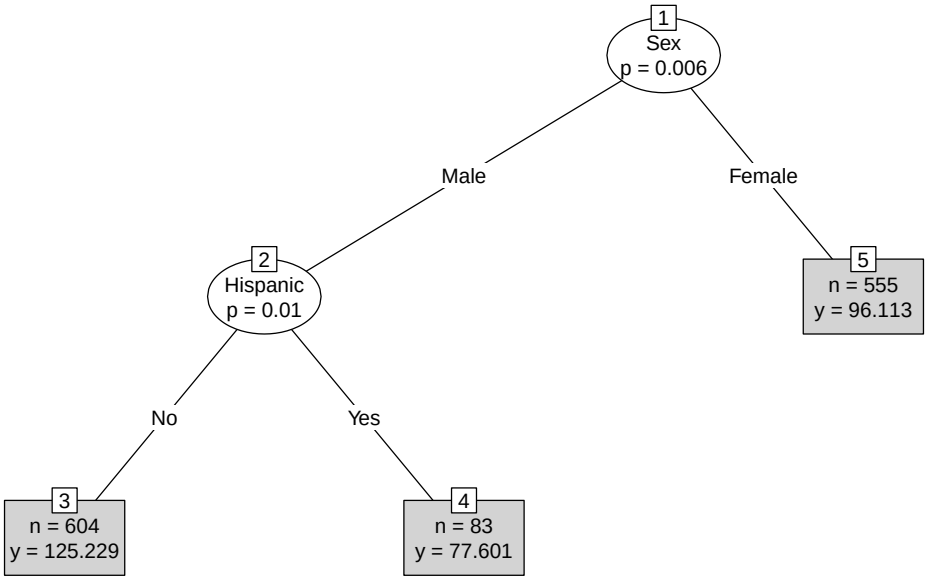

```
plot(tree_list[[12]],type="simple") #IL
```



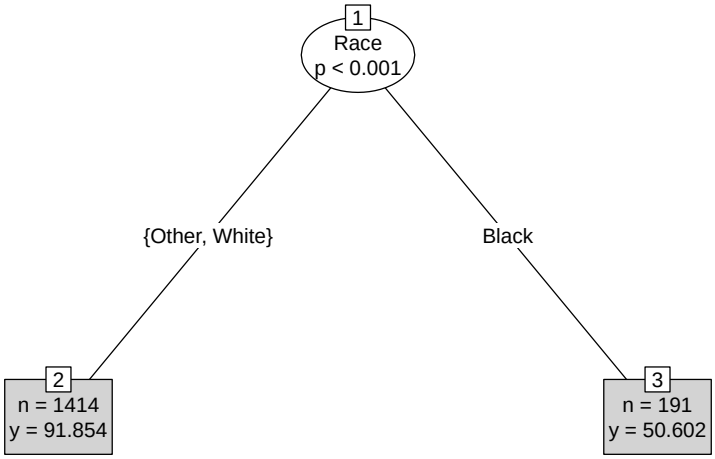

```
plot(tree_list[[14]],type="simple") #WI
```



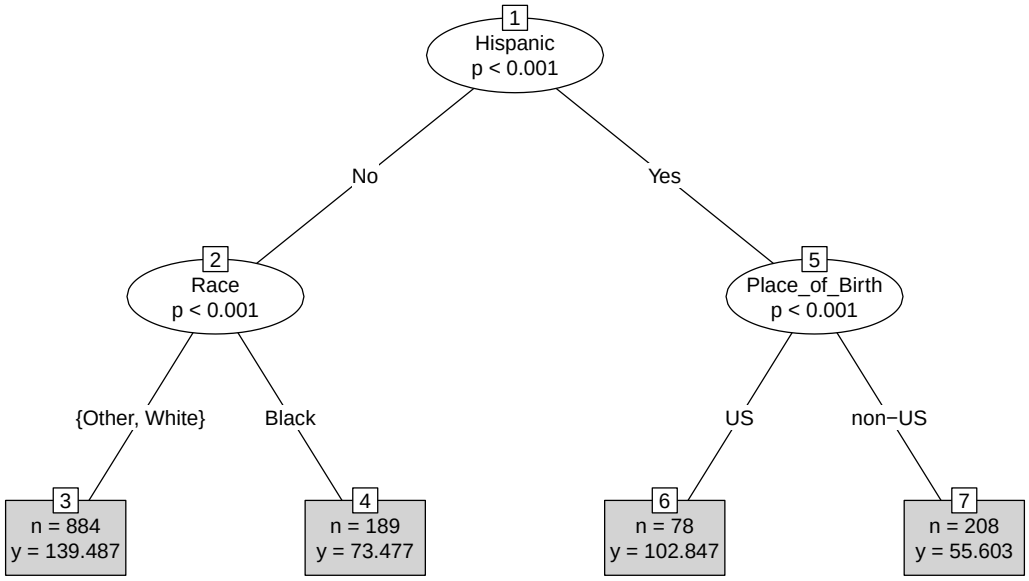

```
plot(tree_list[[47]],type="simple") #WA
```



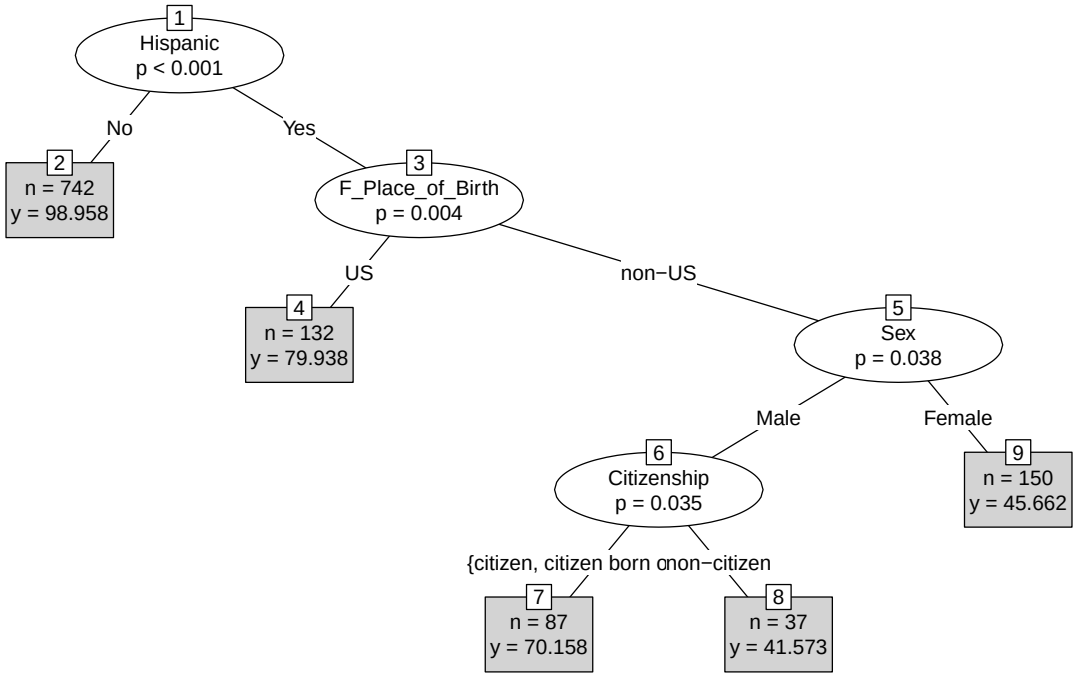

```
plot(tree_list[[10]],type="simple") #OH
```




```
plot(tree_list[[8]],type="simple") #NJ
```




```
plot(tree_list[[44]],type="simple") #AZ
```



MSE and Opportunity based Gini coefficient of each state

```
# mse
mse.state <- c()
for (i in 1:51) {
  test <- test_list[[i]]
  mse.state[i] <- mean((state_list[[i]][test, ]$HHincome-
                  predict(tree_list.train[[i]],
                          newdata = state_list[[i]][test, ]))^2)
}
```

```
# opportunity based gini coefficient
gini_opp.state <- c()
for (i in 1:51) {
  gini_opp.state[i] <- gini(predict(tree_list[[i]],newdata = state_list[[i]]))
}
# normal gini-coefficient
gini.state <- c()
for (i in 1:51) {
  gini.state[i] <- gini(state_list[[i]]$HHincome)
}
```

```
gini.my <- data.frame(state=state_name,
                      my.gini=round(gini.state,3),
                      my.gini.opp=round(gini_opp.state,3),
                      mse=round(mse.state,0),
                      opt_alpha=as.numeric(opt_alpha.state$alpha))
gini.official <- fips %>% select(state,gini)

gini.comp <- left_join(gini.my,gini.official,by="state")
```

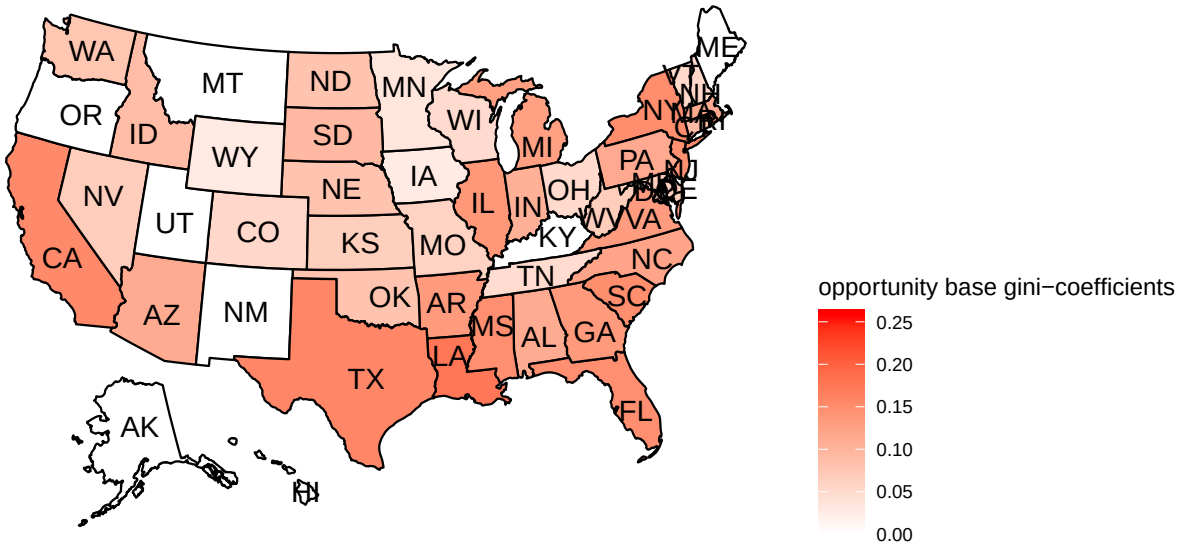
```r
kable(gini.comp,booktabs = T)
```

state	my.gini	my.gini.opp	mse	opt_alpha	gini
ME	0.437	0.000	3747	0.015	0.45
NH	0.414	0.053	7769	0.080	0.45
VT	0.420	0.052	7735	0.070	0.45
MA	0.483	0.113	17167	0.080	0.49
RI	0.492	0.155	22169	0.075	0.47
CT	0.499	0.092	11911	0.060	0.50
NY	0.511	0.150	11575	0.080	0.51
NJ	0.514	0.145	23525	0.035	0.48
PA	0.473	0.113	16592	0.100	0.47
OH	0.464	0.050	6091	0.015	0.47
IN	0.439	0.105	5324	0.085	0.45
IL	0.477	0.138	12780	0.010	0.49
MI	0.460	0.126	7689	0.085	0.47
WI	0.427	0.048	12470	0.060	0.45
MN	0.446	0.034	7419	0.070	0.45
IA	0.420	0.026	11951	0.040	0.44
MO	0.455	0.059	6012	0.095	0.47
ND	0.455	0.079	9465	0.095	0.44
SD	0.462	0.095	7189	0.085	0.44
NE	0.446	0.080	7365	0.045	0.45
KS	0.452	0.065	7796	0.100	0.46
DE	0.461	0.069	11706	0.095	0.46
MD	0.432	0.065	7840	0.010	0.45
DC	0.531	0.258	13330	0.020	0.52
VA	0.470	0.131	10156	0.065	0.48
WV	0.458	0.067	7587	0.060	0.47
NC	0.484	0.121	6849	0.015	0.48
SC	0.480	0.149	8010	0.040	0.48
GA	0.505	0.134	8749	0.010	0.48
FL	0.491	0.148	6906	0.085	0.49
KY	0.518	0.000	14475	0.040	0.48
TN	0.463	0.048	7197	0.080	0.48
AL	0.480	0.112	12760	0.060	0.49
MS	0.492	0.147	5707	0.030	0.48
AR	0.466	0.134	3860	0.095	0.48
LA	0.496	0.176	3131	0.055	0.49
OK	0.453	0.080	7267	0.080	0.47
TX	0.487	0.157	8425	0.085	0.48
MT	0.439	0.000	5022	0.080	0.45
ID	0.435	0.089	11613	0.100	0.45
WY	0.431	0.028	4844	0.050	0.46
CO	0.445	0.052	7134	0.065	0.46
NM	0.521	0.000	11281	0.010	0.49
AZ	0.474	0.112	13271	0.085	0.46
UT	0.416	0.000	9587	0.020	0.43
NV	0.464	0.067	9240	0.055	0.47
WA	0.458	0.077	23078	0.100	0.46
OR	0.445	0.000	6449	0.010	0.46
CA	0.491	0.153	12780	0.095	0.49
AK	0.452	0.000	9383	0.010	0.43
HI	0.469	0.000	11796	0.015	0.45

State Mapping (opp base)

```
library(usmap)
df.gini.opp <- data.frame(state=state_name,
                      gini_opp=gini_opp.state)
df.gini.new <- df.gini.opp[order(-gini_opp.state),]
df.gini.new$order = c(1:51)
plot_usmap(data=df.gini.new, values="gini_opp", color='black',
           labels = state_name,include=state_name)+
  scale_fill_continuous(
    low='white', high='red', name="opportunity base gini-coefficients")+
  theme(legend.position = 'right')
```

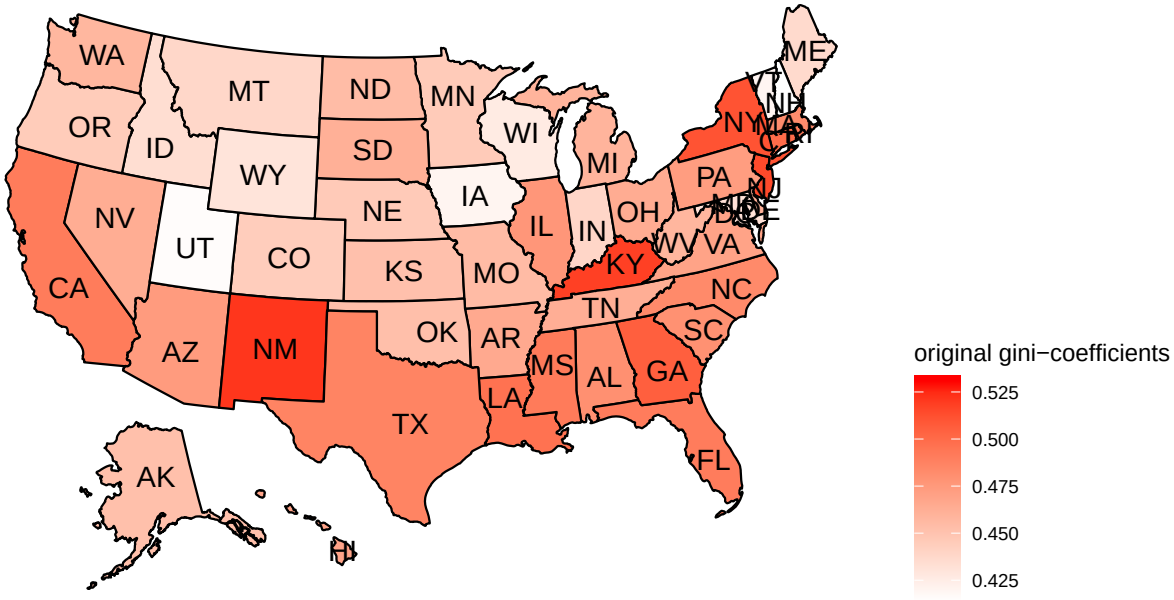
```
## Warning in if (labels) {: the condition has length > 1 and only the first
## element will be used
```



Mapping (original base)

```
library(usmap)
df.gini <- data.frame(state=state_name,
                      gini=gini.state)
plot_usmap(data=df.gini, values="gini", color='black',
           labels=state_name,include=state_name)+
  scale_fill_continuous(
    low='white', high='red', name="original gini-coefficients")+
  theme(legend.position = 'right')
```

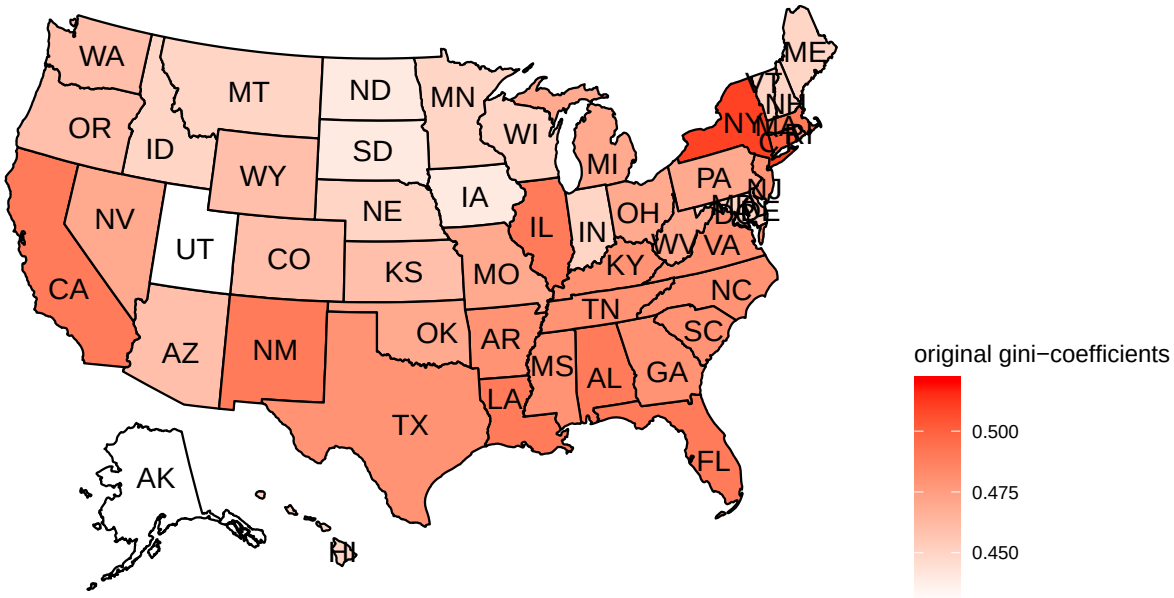
```
## Warning in if (labels) {: the condition has length > 1 and only the first
## element will be used
```



Mapping (official base)

```
library(usmap)
df.gini <- data.frame(state=state_name,
                      gini=as.numeric(gini.comp$gini))
plot_usmap(data=df.gini, values="gini", color='black',
           labels=state_name,include=state_name)+
  scale_fill_continuous(
    low='white', high='red', name="original gini-coefficients")+
  theme(legend.position = 'right')
```

```
## Warning in if (labels) {: the condition has length > 1 and only the first
## element will be used
```



bagging

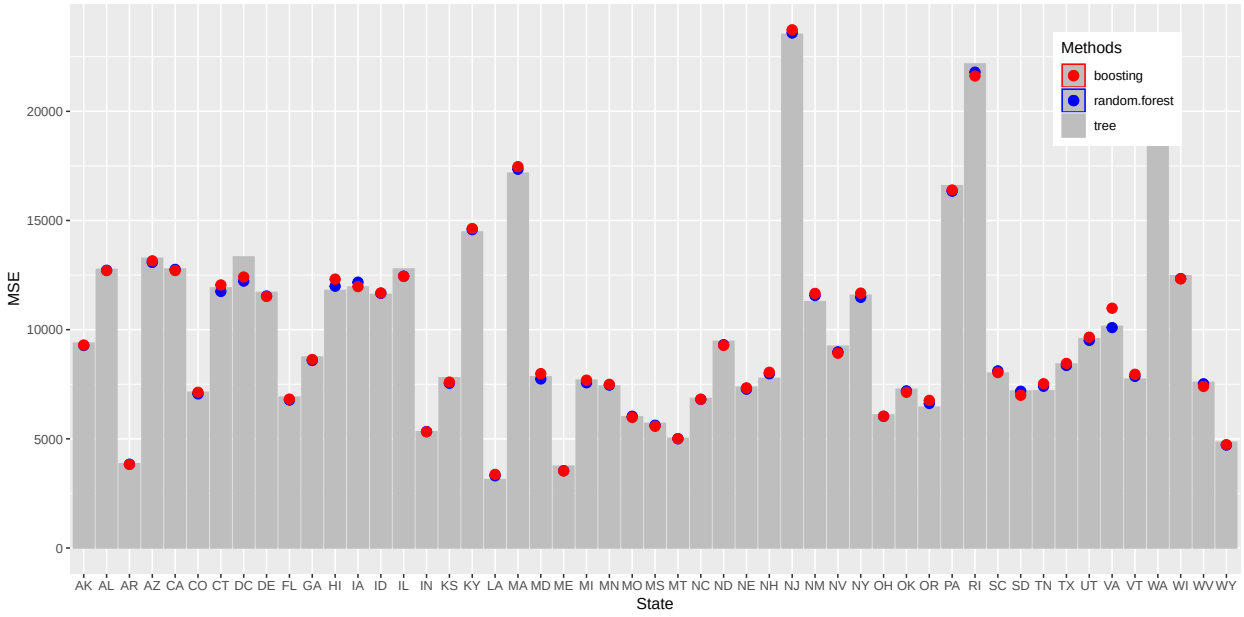
```
# bagging
set.seed(5000)
bag_list.train <- vector(mode = "list", length = 51)
for (i in 1:length(bag_list.train)){
  train <- train_list[[i]]
  bagtree <- randomForest(HHincome ~., data = state_list[[i]],
                          subset = train)
  bag_list.train[[i]] <- bagtree
}

#calculating mse
bag.mse.state <- c()
for (i in 1:length(bag_list.train)) {
  test <- test_list[[i]]
  bag.mse.state[i] <- mean((state_list[[i]][test, ]$HHincome-
                    predict(bag_list.train[[i]],
                            newdata = state_list[[i]][test, ]))^2)
}
```

Boosting

```
set.seed(10000)
boost_list.train <- vector(mode = "list", length = 51)
```

```
boost.mse.state <- c()
for (i in 1:length(boost_list.train)) {

 train <- train_list[[i]]
 test <- test_list[[i]]
 boost.state <- gbm(HHincome ~ .,
                    data=state_list[[i]][train,],
                    distribution="gaussian",
                    n.trees=1000,
                    shrinkage=0.01,
                    interaction.depth = 4)
 boost_list.train[[i]] <- boost.state
 test.pred <- predict(boost.state, newdata=state_list[[i]][test,], n.trees=1000)
 boost.mse.state[i] <- mean((state_list[[i]][test,]$HHincome - test.pred)^2)
}
```

```
validity.state <- data.frame(state=state_name,
                             baseline = mse.state,
                             bagging = bag.mse.state,
                             boosting = boost.mse.state) %>%
                  arrange(state)


colors <- c("tree"="grey",
            "random.forest"="blue",
            "boosting"="red")

ggplot(validity.state, aes(x=state,y=baseline,color="tree"))+
  geom_bar(stat = "identity",fill="grey")+
  geom_point(aes(x=state,y=bagging,color="random.forest"),size=3)+
  geom_point(aes(x=state,y=boosting,color="boosting"),size=3)+
  labs(x = "State",y = "MSE",
       color = "Methods")  +
  scale_color_manual(values = colors)+
  theme(legend.position = c(.95, .95),
        legend.justification = c("right", "top"),
        legend.box.just = "right")
```



Age-level Analysis

```
df1.age <- df1 %>%
  select(HHincome,Sex,Citizenship,Race,Hispanic,Place_of_Birth,
         `F_Place_of_Birth`,`M_Place_of_Birth`,A_Age)
head(df1.age)
```

```
##   HHincome    Sex Citizenship  Race Hispanic Place_of_Birth F_Place_of_Birth
## 1   18.000   Male     citizen White       No             US               US
## 2   21.780 Female     citizen White       No             US               US
## 3   12.000 Female     citizen White       No             US               US
## 4   22.727 Female     citizen White       No             US               US
## 5   20.954 Female     citizen White       No             US               US
## 6   55.000 Female     citizen White       No             US               US
##   M_Place_of_Birth A_Age
## 1               US    21
## 2               US    85
## 3               US    61
## 4           non-US    73
## 5               US    80
## 6               US    53
```

Generate age sub-group

```
# generate age flag
df1.age$AGE_flag[df1.age$A_Age>=18 & df1.age$A_Age<35]<-1
df1.age$AGE_flag[df1.age$A_Age>=35 & df1.age$A_Age<50]<-2
df1.age$AGE_flag[df1.age$A_Age>=50 & df1.age$A_Age<65]<-3
df1.age$AGE_flag[is.na(df1.age$AGE_flag)]<-0
```

Summary Statistics (age)

```
age.category <- c("18-34","35-49","50-64")

sum_table.age <- df1.age %>%
  group_by(AGE_flag)%>%
  summarise(sample.size=n(),
            ave.income=round(mean(HHincome),1),
            `25%tile`=round(quantile(HHincome,0.25),1),
            `75%tile`=round(quantile(HHincome,0.75),1),
            std.dev=round(sd(HHincome),1),
            Gini=round(gini(HHincome),3),
            male=round(mean(I(Sex=="Male"))*100,1),
            white=round(mean(I(Race=="White"))*100,1)) %>%
  filter(AGE_flag>0) %>%
  as.data.frame()


sum_table.age[,1] <- age.category

kable(sum_table.age, booktabs = T)
```

AGE_flag	sample.size	ave.income	25%tile	75%tile	std.dev	Gini	male	white
18-34	13614	79.0	34	99.5	91.7	0.442	49.5	74.6
35-49	19407	109.5	45	138.0	116.8	0.446	50.9	77.2
50-64	18360	100.7	35	129.2	117.5	0.485	52.1	76.9

Split the df into training and test set for each age group

```
# generate age sub-group dataset
age_list <- vector(mode = "list", length = 3)
for (i in 1:3) {
    age_list[[i]] <- df1.age %>% filter(AGE_flag == i) %>% select(-c(A_Age,AGE_flag))
}

# Generate Training and Testing Set List. We assign 75 % of sample to Training.
train_length.age <- 3
test_length.age <- 3
train_list.age <- vector(mode = "list", length = 3)
test_list.age <- vector(mode = "list", length = 3)
set.seed(5000)
for (i in 1:train_length.age) {
  train_list.age[[i]]=sample(1:nrow(age_list[[i]]),
                        nrow(age_list[[i]])*0.75)

  test_list.age[[i]]=
    c(1:nrow(age_list[[i]]))[!(c(1:nrow(age_list[[i]])) %in% train_list.age[[i]])]
}
```

Conduct CV to set optimal alpha

```
# CV for all age group
set.seed(12345)
```

```r
opt_alpha.age <- data.frame(age_group=c(1:3))

for (i in 1:train_length.age) {

  train <- train_list.age[[i]]
  opt_alpha.age$alpha[i] <- cv_alpha(age_list[[i]][train, ])
}
```

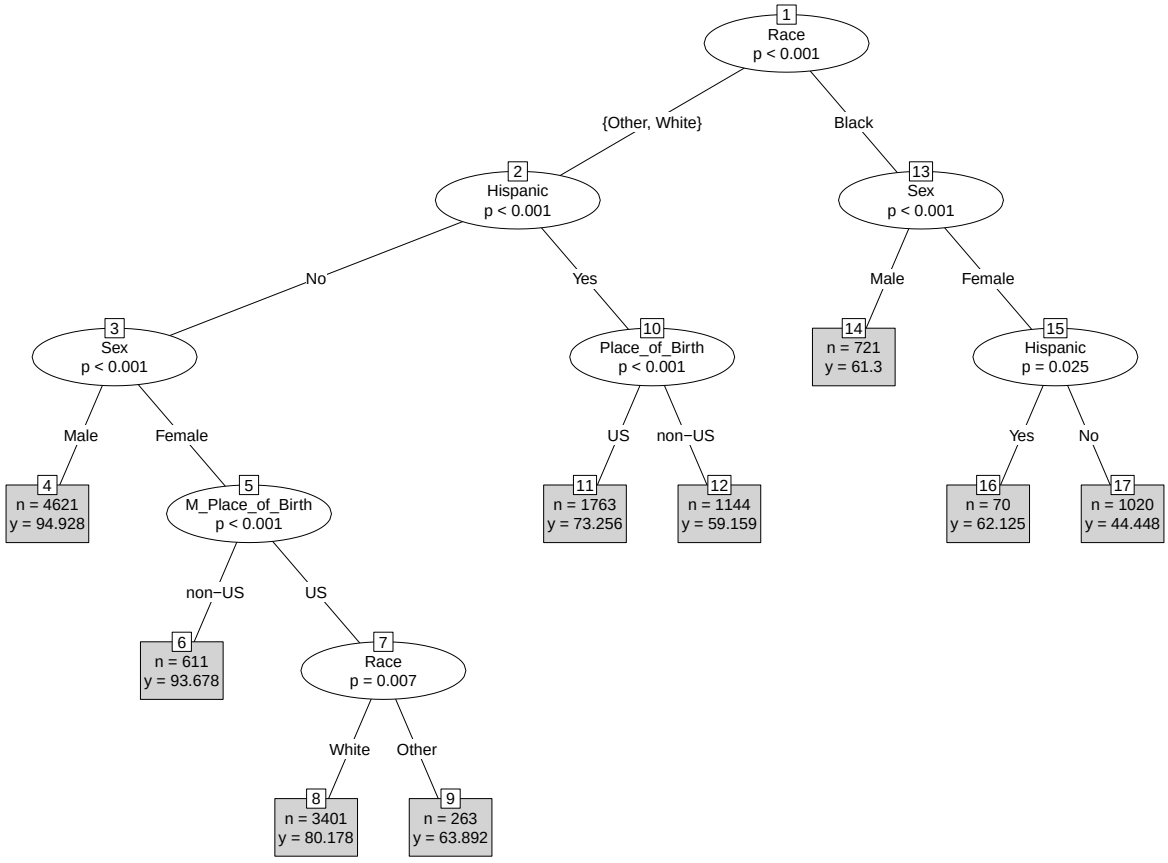
Using optimal alpha, estimate Tree of each age group

```r
# using only training set
tree_list.age.train <- vector(mode = "list", length = 3)
for (i in 1:length(tree_list.age.train)) {

 train <- train_list.age[[i]]
 citree <- ctree(HHincome ~ .,
                 data=age_list[[i]][train,],
                 control =
                   ctree_control(mincriterion=1-opt_alpha.age$alpha[i],
                                 testtype="Bonferroni"))

 tree_list.age.train[[i]] <- citree
}
```

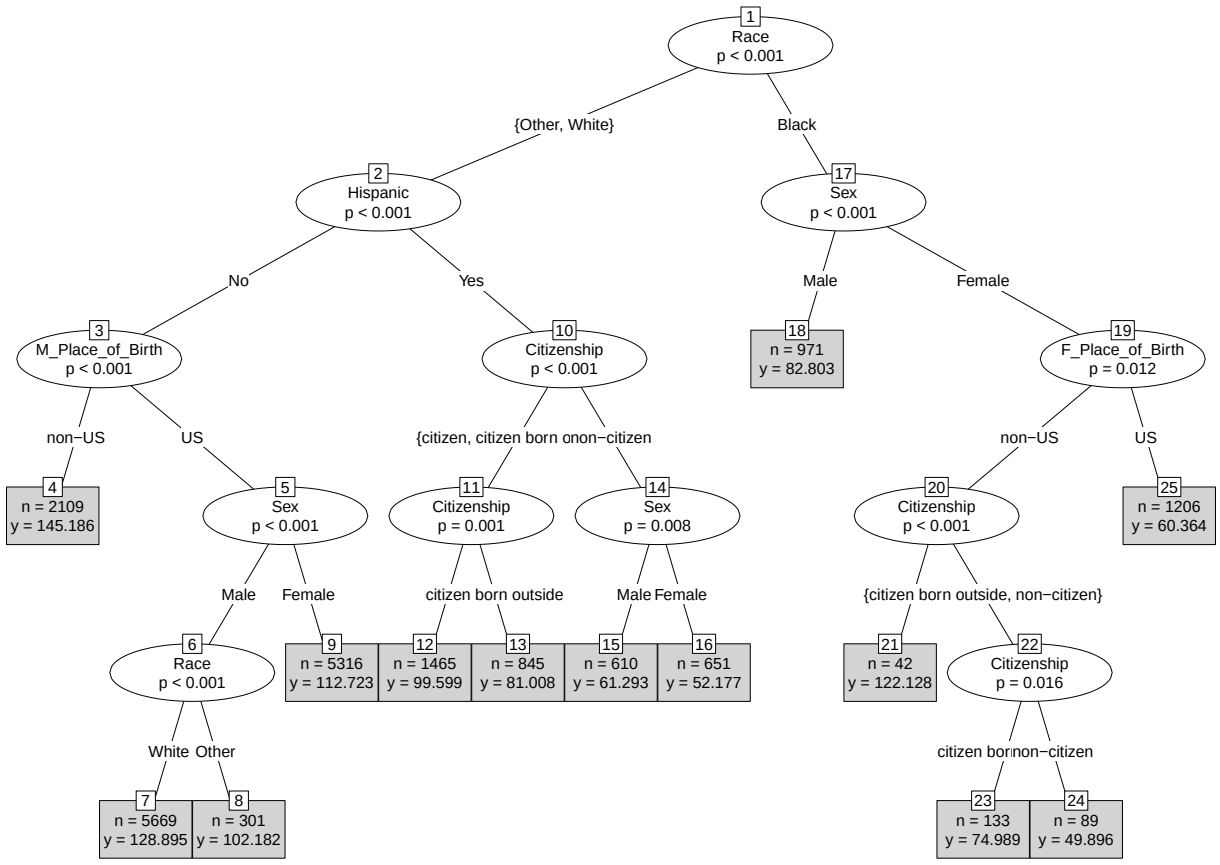
```r
# use whole sample
tree_list.age <- vector(mode = "list", length = 3)
for (i in 1:length(tree_list.age)) {
 train <- train_list.age[[i]]
 citree <- ctree(HHincome ~ .,
                 data=age_list[[i]],
                 control = ctree_control(mincriterion=1-opt_alpha.age$alpha[i],
                                         testtype="Bonferroni"))
 tree_list.age[[i]] <- citree
}
```

Draw Tree of each age group

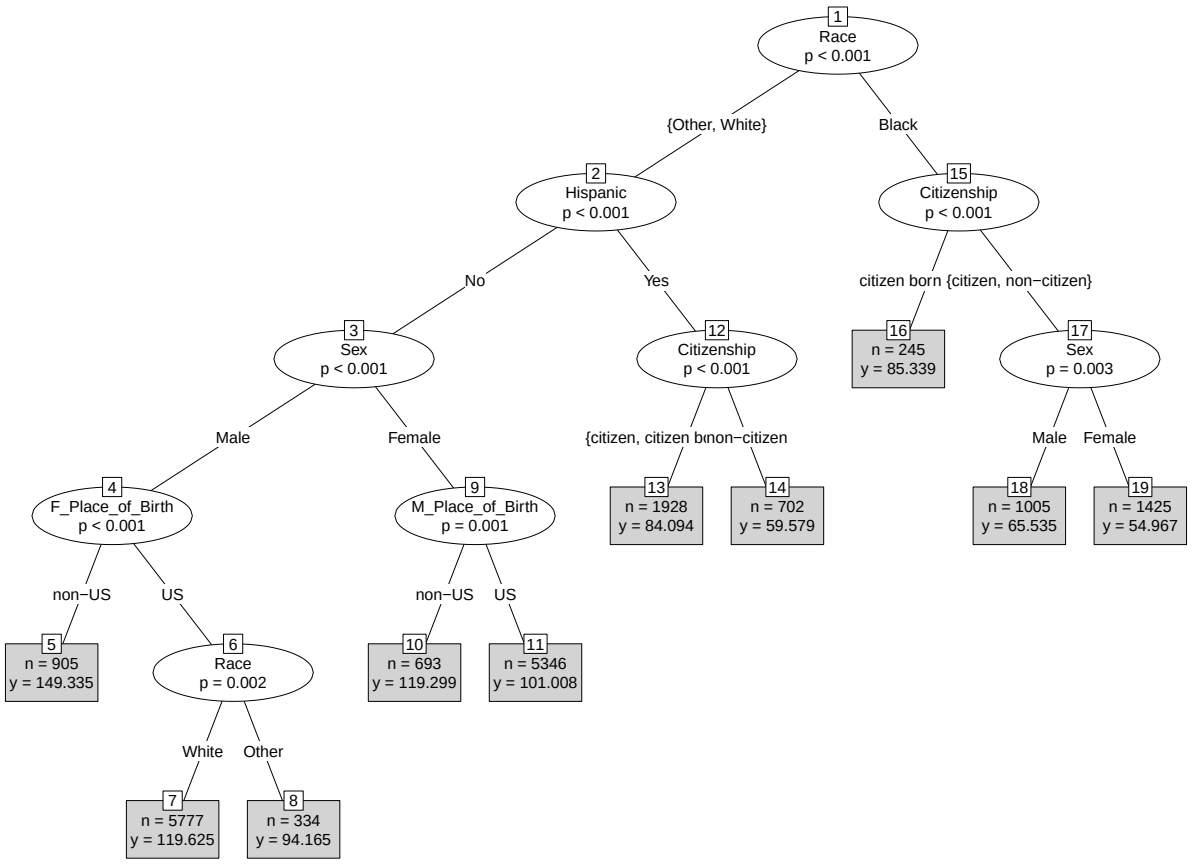
```r
plot(tree_list.age[[1]],type="simple")
```




```
plot(tree_list.age[[2]],type="simple")
```




```
plot(tree_list.age[[3]],type="simple")
```



MSE and Opportunity based Gini coefficient for 3 age groups

```
# mse
mse.age <- c()
for (i in 1:3) {
  test <- test_list.age[[i]]
  mse.age[i] <- mean((age_list[[i]][test, ]$HHincome-
                      predict(tree_list.age.train[[i]],
                              newdata = age_list[[i]][test, ]))^2)
}
```

```
# opportunity based gini coefficient
gini_opp.age <- c()
for (i in 1:3) {
  gini_opp.age[i] <- gini(predict(tree_list.age[[i]],newdata = age_list[[i]]))
}
```

```
gini.comp.age <- data.frame(age=age.category,
                        gini=round(as.numeric(sum_table.age$Gini),3),
                        my.gini.opp=round(gini_opp.age,3),
                        mse=round(mse.age,0),
                        opt_alpha=as.numeric(opt_alpha.age$alpha))
```

```
kable(gini.comp.age,booktabs = T)
```

age	gini	my.gini.opp	mse	opt_alpha
18-34	0.442	0.108	5690	0.050
35-49	0.446	0.130	13102	0.045
50-64	0.485	0.129	14825	0.020

bagging

```
# bagging
set.seed(5000)
bag_list.age.train <- vector(mode = "list", length = 3)
for (i in 1:length(bag_list.age.train)){
  train <- train_list.age[[i]]
  bagtree <- randomForest(HHincome ~., data = age_list[[i]],
                          subset = train)
  bag_list.age.train[[i]] <- bagtree
}

#calculating mse
bag.mse.age <- c()
for (i in 1:length(bag_list.age.train)) {
  test <- test_list.age[[i]]
  bag.mse.age[i] <- mean((age_list[[i]][test, ]$HHincome-
                    predict(bag_list.age.train[[i]],
                            newdata = age_list[[i]][test, ]))^2)
}
```

Boosting

```
set.seed(1000)
boost_list.age.train <- vector(mode = "list", length = 3)
boost.mse.age <- c()
for (i in 1:length(boost_list.age.train)) {

 train <- train_list.age[[i]]
 test <- test_list.age[[i]]
 boost.age <- gbm(HHincome ~ .,
                    data=age_list[[i]][train,],
                    distribution="gaussian",
                    n.trees=1000,
                    shrinkage=0.01,
                    interaction.depth = 4)
 boost_list.age.train[[i]] <- boost.age
 test.pred <- predict(boost.age, newdata=age_list[[i]][test,], n.trees=1000)
 boost.mse.age[i] <- mean((age_list[[i]][test,]$HHincome - test.pred)^2)
}
```

```
validity.age <- data.frame(age=age.category,
                           baseline = mse.age,
                           bagging = bag.mse.age,
                           boosting = boost.mse.age)
```

```
colors <- c("tree"="grey",
            "random.forest"="blue",
            "boosting"="red")

ggplot(validity.age, aes(x=age,y=baseline,color="tree"))+
  geom_bar(stat = "identity",fill="grey")+
  geom_point(aes(x=age,y=bagging,color="random.forest"),size=3)+
  geom_point(aes(x=age,y=boosting,color="boosting"),size=3)+
  labs(x = "Age",y = "MSE",
       color = "Methods")  +
  scale_color_manual(values = colors)+
  theme(legend.position = c(.2, .95),
        legend.justification = c("right", "top"),
        legend.box.just = "right")
```