Data 2020: Final Project

Outline:

I. **Exploratory Analysis**
II. **Models and Hypothesis Tests**
III. **Prediction**
IV. **Conclusion**

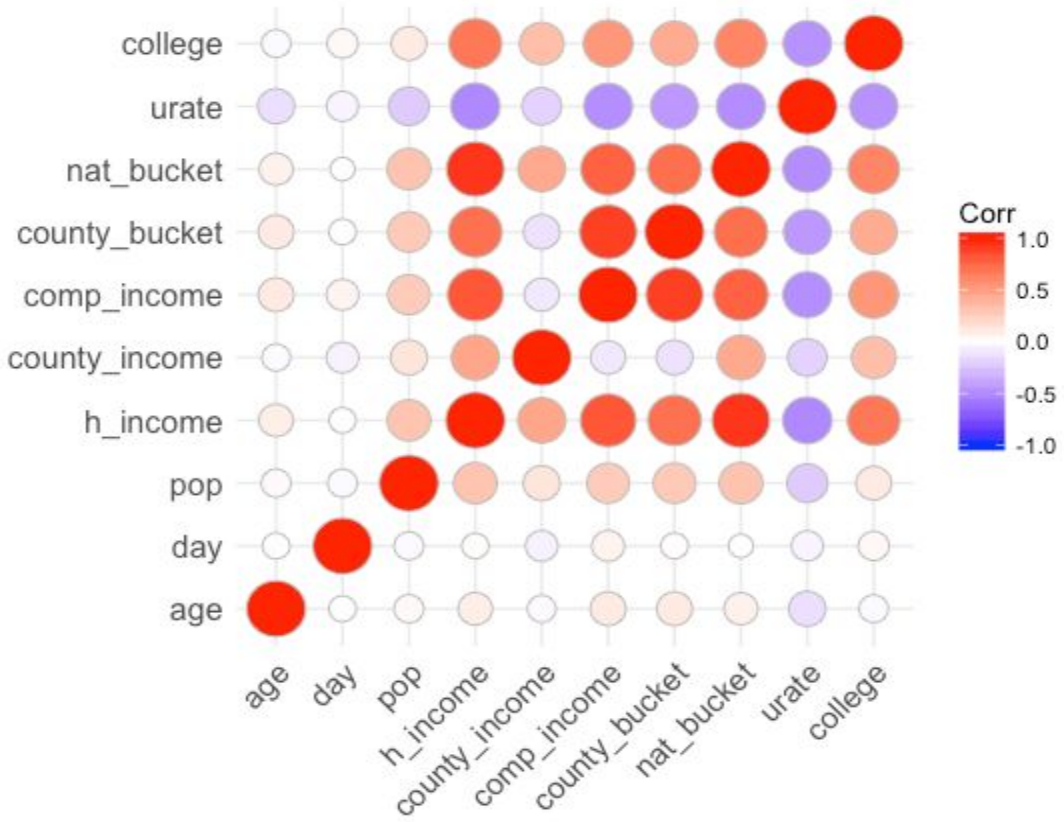
The goal of this project is to see to how much of a role race plays in police killings. Since we do not have the racial profile of the police officers who are involved in these killings we must limit our conclusions to how the race of those killed by police plays an explanatory role.

I. Exploratory Analysis

In an effort to become better acquainted with the data I created a number of graphs and plots. The idea is to use visualization tools to discover correlations, trends or oddities that might otherwise go unnoticed if I began by just building models.

I started by using the police killings dataframe (non-numeric values are not included) to look at the covariances among some of the variables.[1]

[1] I will often use “victims” to refer to the population of people represented in the police killings dataframe. This is strictly for convenience and not meant to pre-judge the causes or circumstances behind these deaths.

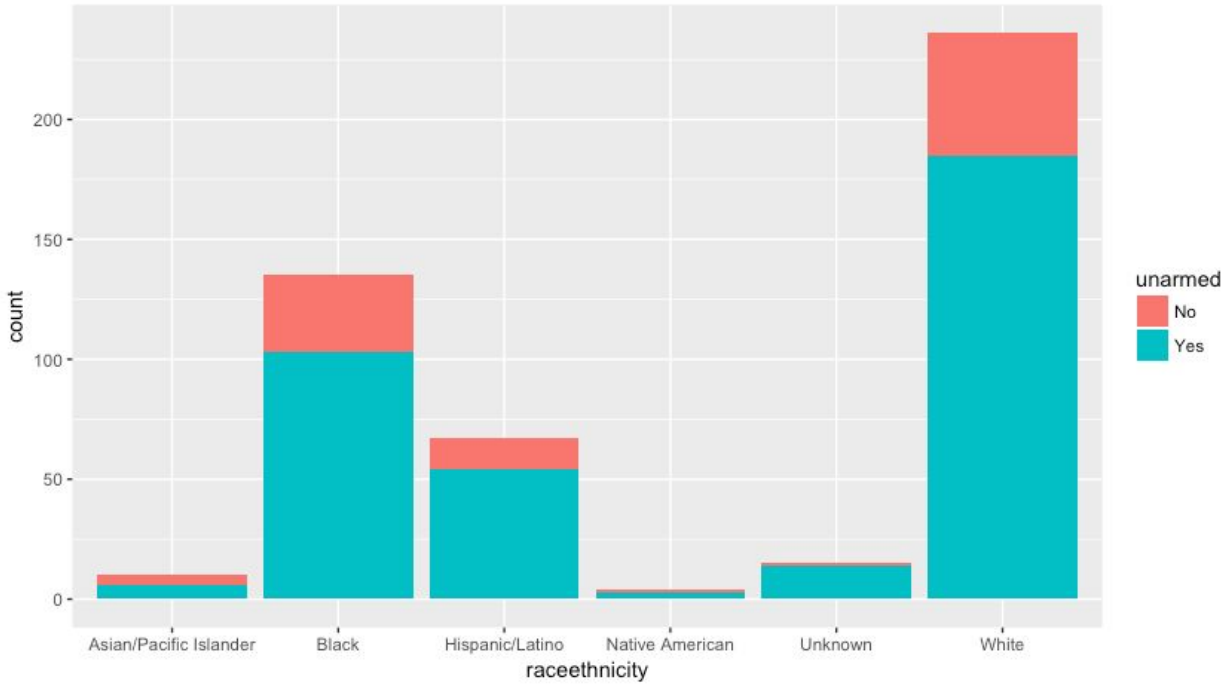


Covariance matrix: Nothing particularly surprising here. We see correlation where we would expect. For example, the level of education is positively correlated with median household income and negatively correlated with the rate of unemployment.

I next made a number of histograms to try and see what types of trends might exist among the victims.

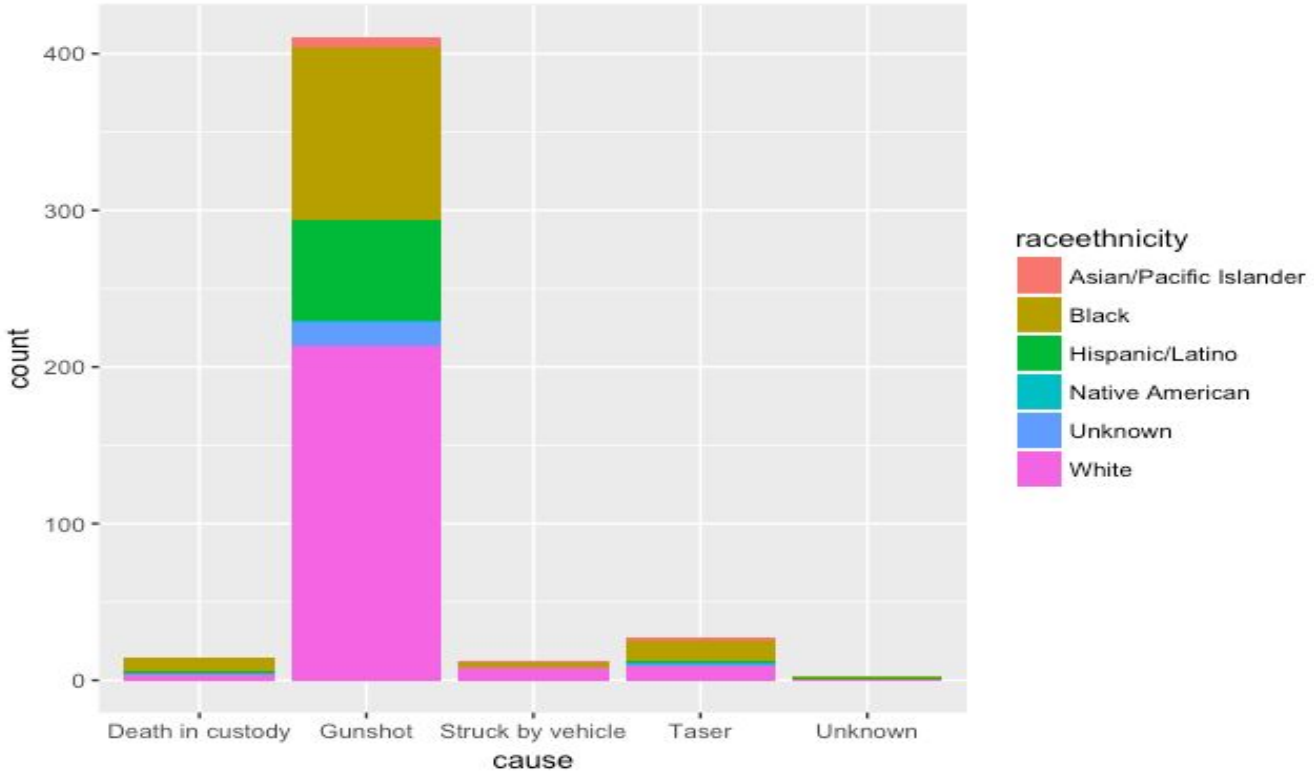
How/whether the deceased was armed. Most often those killed were armed. But an unarmed police killing is the second most frequent type of police killing. We'll see this again below.

Jesse Geman
Data 2020: Final Project

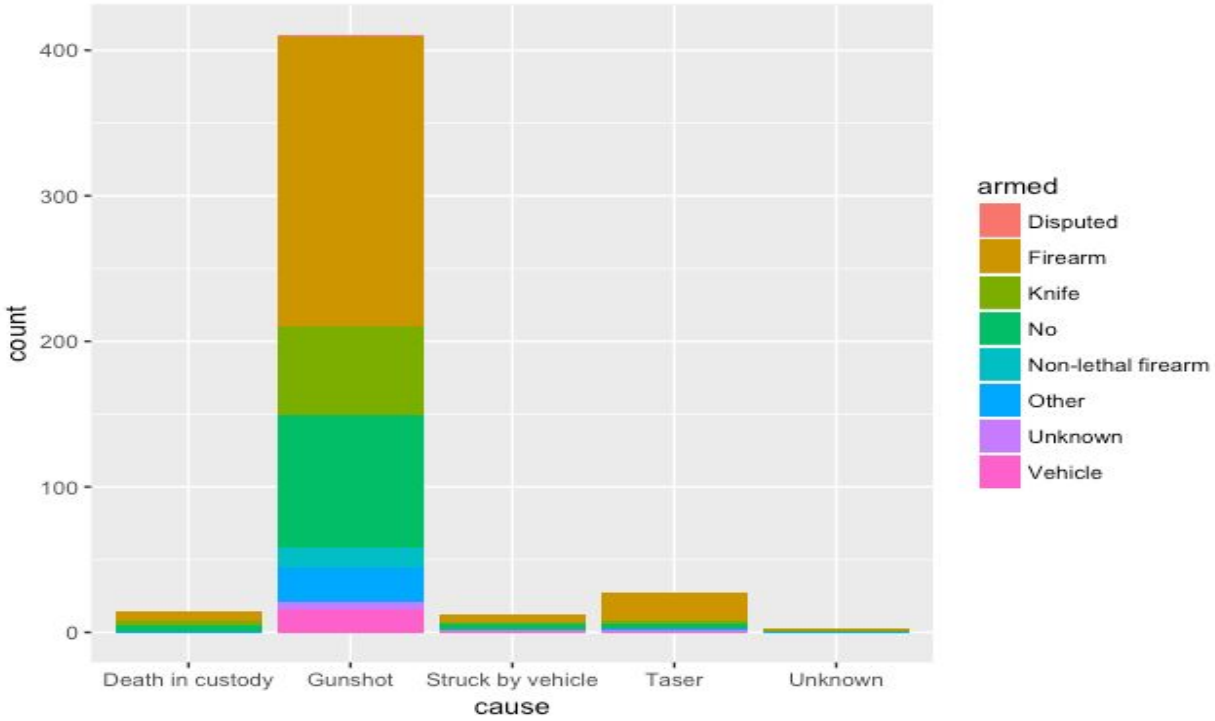


Victims by race: For comparison: 72% of Americans are White, 17% are Hispanic/Latino, 12.6% are Black, 5% are Asian/Pacific Islander, and 0.9% are Native. (In some cases, Hispanics/Latinos might also have been categorized as White or Black.) The most striking features of this graph are: (i) the relatively low fraction of the victims who are White (about 55%), as compared to their prevalence in the American population (about 72%), and (ii) the relatively high fraction of the victims who are Black (about 30%), as compared to their prevalence in the American population (about 13%). NOTE: The above plot indicates unarmed Blacks are killed disproportionately often.

Jesse Geman
Data 2020: Final Project

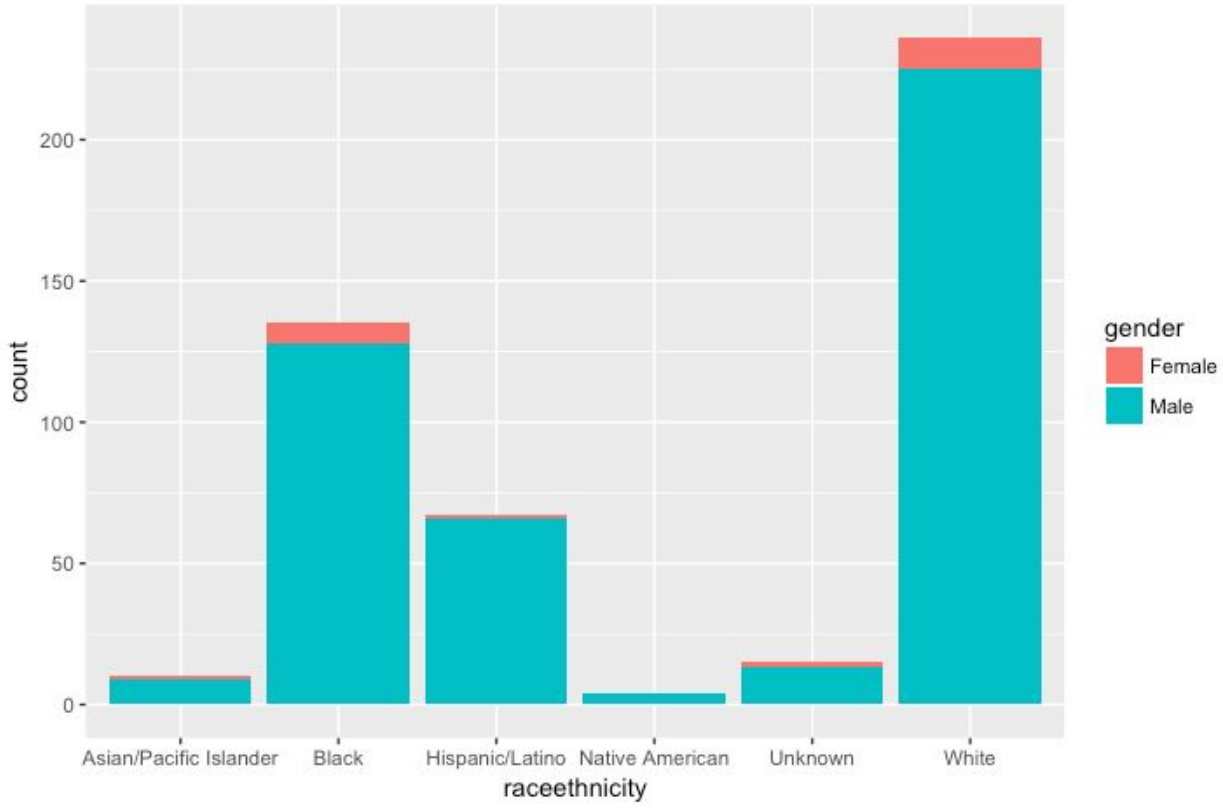


Cause of death. The vast majority of police killings were the result of gunshots.

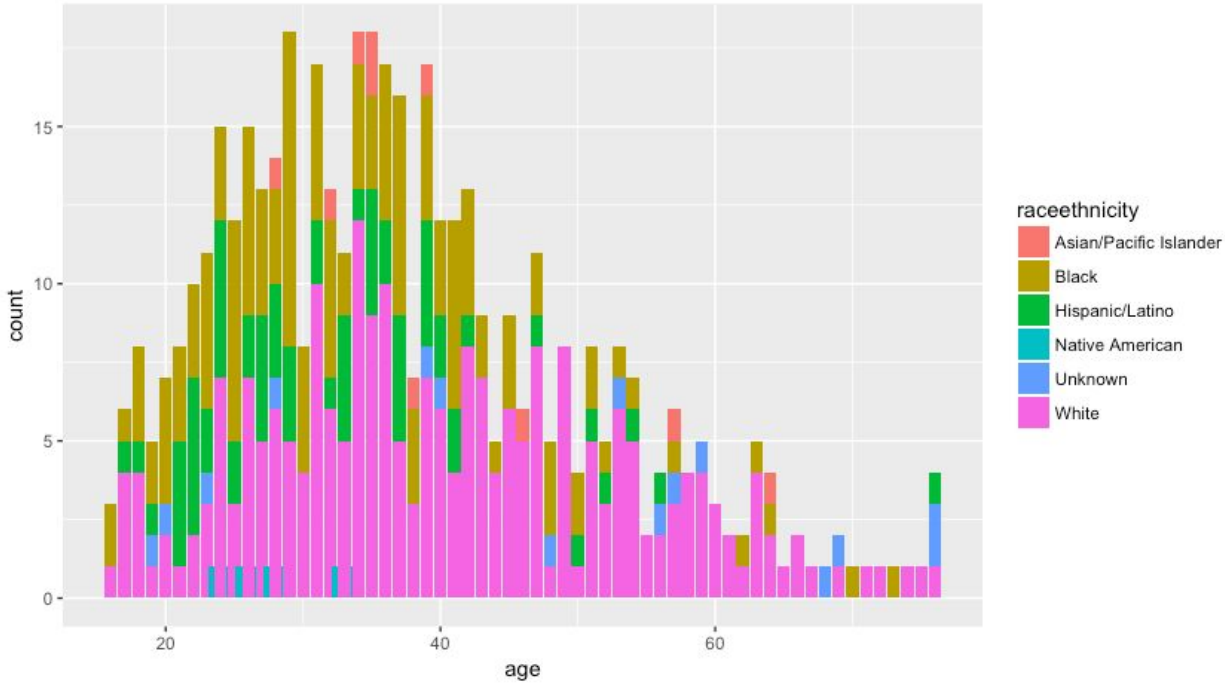


Cause of death. Of those shot by police, roughly half were armed with firearms. About a quarter were unarmed.

Jesse Geman
Data 2020: Final Project



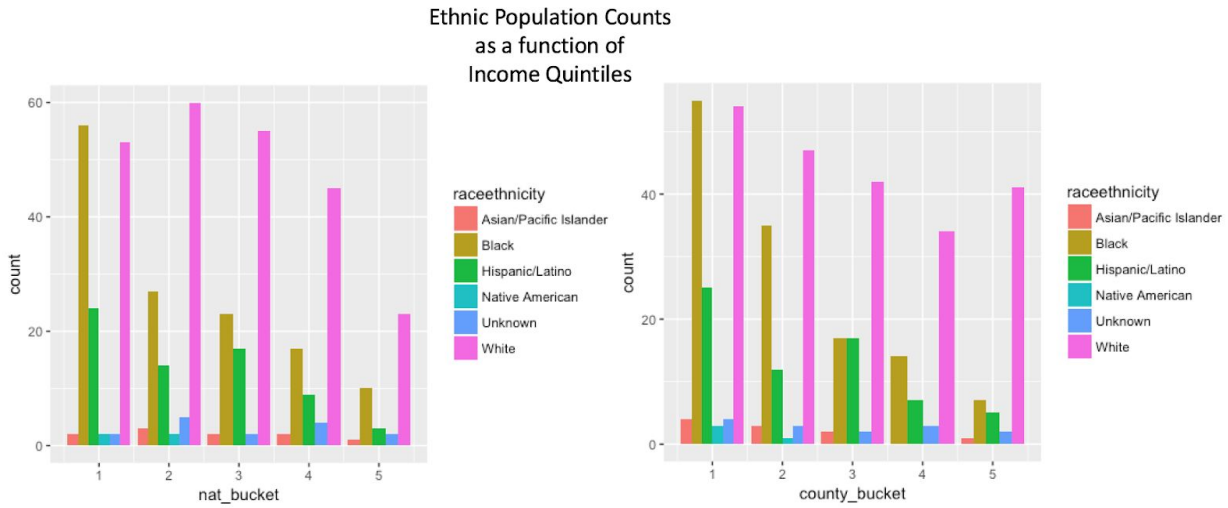
Victims by Race and Gender (the vast majority of the deceased were men).



Victims by age. Note, this plot has been shifted since the draft. The age of each victim has been increased by fifteen. The average age is now closer thirty as opposed to twenty.

Based on these histograms, and looking ahead: Of these variables, in an effort to avoid small fragments of a not-very-large data set, I did not make further use of the non-binary categorical variables: 'how and whether the victim was armed' and 'cause of death'. Furthermore, as we can see above, gender does not appear to be a particularly informative variable in a predictive model, which is hardly surprising the small percentage of female victims. But age was selected. As for race, that will be the variable for which we will build models, both for testing hypotheses and for generating predictions.

Before moving on to section II (Models and Hypothesis Tests), I wish to present one more exploratory figure, in order to demonstrate, visually, some of the hazards of aggregating inhomogeneous data.



Victim counts versus income, aggregated globally (left-hand panel) and locally (right-hand panel)

Consider first the left-hand panel. Five household "income buckets" are indicated on the horizontal access, lowest ('1') to highest ('5'). Buckets are defined by quintiles, computed from national household income data. Each victim belongs to one of the five buckets. Colored bars within each bucket show the number of victims in the data set whose incomes fell in that bucket and who belonged to the corresponding ethnic classification, as indicated by the legend. Notice in particular that Blacks are substantially overrepresented in the lower income brackets and underrepresented in the upper income brackets.

Now consider the right-hand panel, which depicts a similar breakdown of the victim data, but with one important difference: The buckets are defined *locally*, county-by-county. In the event of a victim from a relatively poor neighborhood, the victim might be in the second lowest income bucket (bucket 2) nationally, but the highest income bucket (bucket 5) locally. The effect of this more fine-grained (more local) scale of comparison is to reveal a considerably greater imbalance in the following sense: in terms of *relative* income, Blacks are even less well

represented in the highest income bracket. If it were to turn out that low income was predictive of the likelihood of becoming a victim, then it would be logical to expect that the *local, relative* income would be the better measure, since effects like cost-of-living are in a sense "controlled for" by using the local data. Indeed, the relationship between low income and the likelihood of becoming a victim might be substantially muted or even completely hidden by using the aggregated data on the left rather than the fine-grained data on the right.

This will come up again, shortly. Let me define two terms: by a "county effect" I will mean that certain types of counties (e.g. low-income counties) may have more victims per capita. The term does not imply any specific cause, though we can imagine many possibilities. The second term is "race effect", by which I will mean the overrepresentation, among victims, of a particular race, as measured *relative to the fraction of the county population occupied by that same race.*

II. Models and Hypothesis Tests

I designed a sequence of hypothesis tests all aimed at seeing whether and to what extent the demographics of the victims mirrored those of the populations.

The idea is to try to quantify how "surprising" these killings were given the racial makeup of the counties from which they came. Or, put differently, the idea is to try to measure the extent of a race effect among victims. I decided to create a simple, Monte Carlo hypothesis test with the following null hypothesis: the distribution on the racial makeup of those killed by police in a given county is not different from the distribution on the racial makeup of the county's population.

To begin, I removed from police_killings_cleaned.csv all entries (rows) where the race of the victim was unknown or the row contained NA values. This left 421 out of the original 467 entries. These remaining entries belonged to 415 unique counties. For each of these counties, I used the races of the victims (either Asian/Pacific, Black, Hispanic, White, or Native American) to create a (victim) empirical distribution on race. I had, then, 415 (victim) empirical distributions on race.

It turns out that there were only six counties with more than one recorded killing in the dataset. Thus, most of the empirical, county-level, victim distributions had one race that had probability 1 and four races that had probability 0 of being killed by the police. To reiterate, then, I ended up with 415 vectors (there were 415 unique counties with recorded police killings) of length five, where most of the elements were 0 and either one element was 1, or two elements were 0.5 (namely, the cases where there were two recorded police killings and the two victims were of different races). For each of these vectors I also had the corresponding population distributions for the same county, as represented by a second vector also of length five, and derived from the census data.

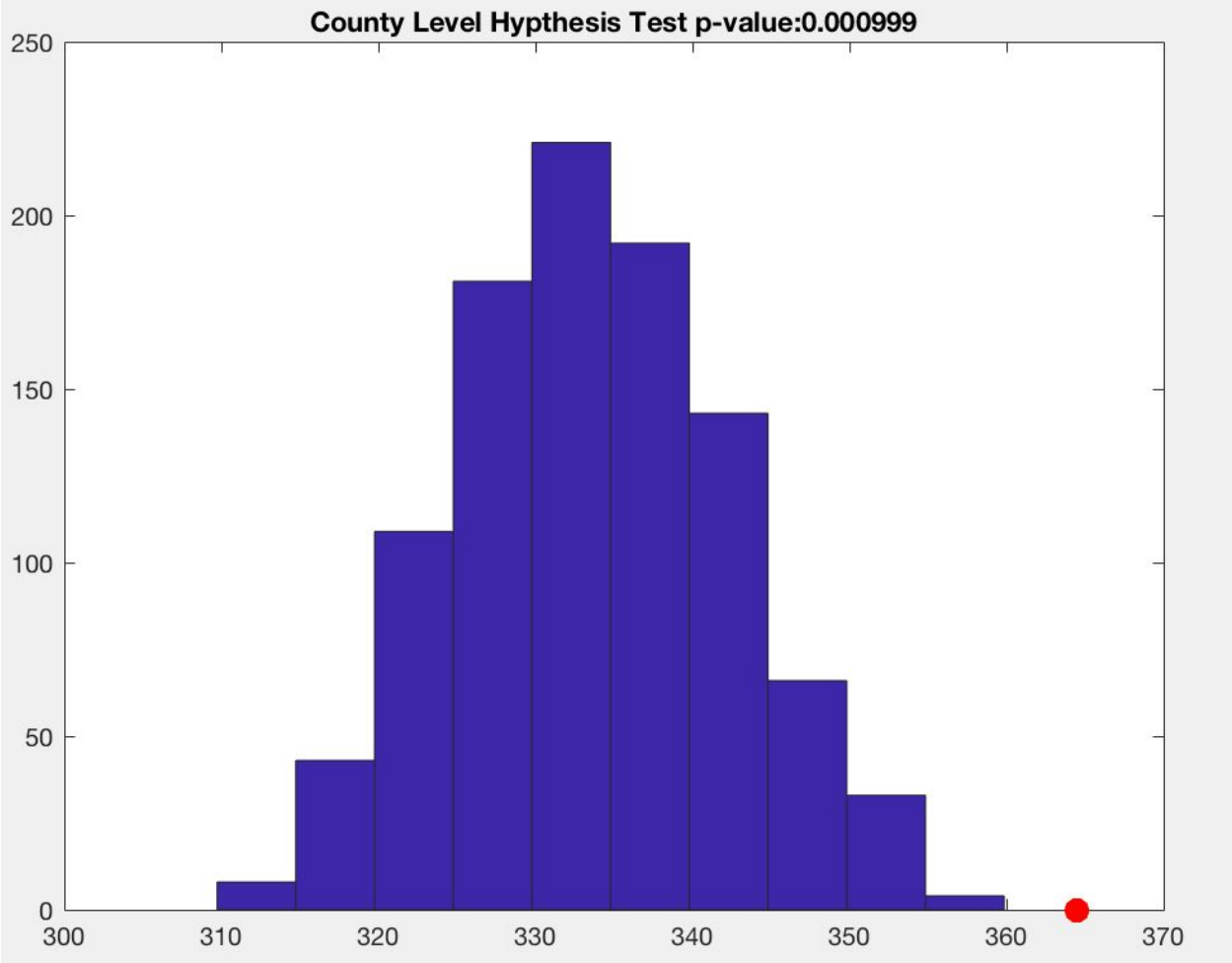
The sparsity of the empirical distribution vectors does not in any way invalidate the use of the permutation-type test that I will shortly describe. Putting aside the particular statistic that I chose to work with, the null hypothesis can now be stated more precisely: the races of the victims, as described by the county-based empirical distributions, were randomly chosen from the actual distributions on race in those same counties.

As for the statistic, I based it on the 415 L1 distances (sums of absolute differences) between the county-level (victim) empirical distribution vectors and the corresponding county-level census-derived distribution vectors. The test statistic itself was just the sum of the L1 distances, summed over the 415 counties.

After calculating the observed value of the statistic, I generated 1,000 sample statistics under the assumption of the null hypothesis. That is, for each county I used the demographic distribution to sample the race for each hypothetical victim in that county (typically just one victim, as already mentioned). I then calculated the L1 distance between the generated empirical sample distribution (obtained the same way I obtained the empirical distribution for the observed killings) and the demographic distribution for that county. Finally, I summed the 415 L1 distances to get a new generated sample statistic value. It is worth reiterating that these statistics were all generated using the model given by the null hypothesis: For each county, the race of each individual killed by police was drawn from the demographic distribution on race; victims were killed in proportion to their numbers in the county. As I mentioned above, I generated 1,000 sample statistics.

Below is a histogram of the 1,000 statistic values. The red dot corresponds to the statistic we got from the actual data. Obviously, it is significant and we can conclude the distribution on the races of those killed by police differs quite a bit from the distribution on the racial makeup of the county. The p-value for this test was 0.000999. [2]

[2] The code for my hypothesis test can be found at the end of this section on “Models and Hypothesis Tests”. It was done in Matlab.



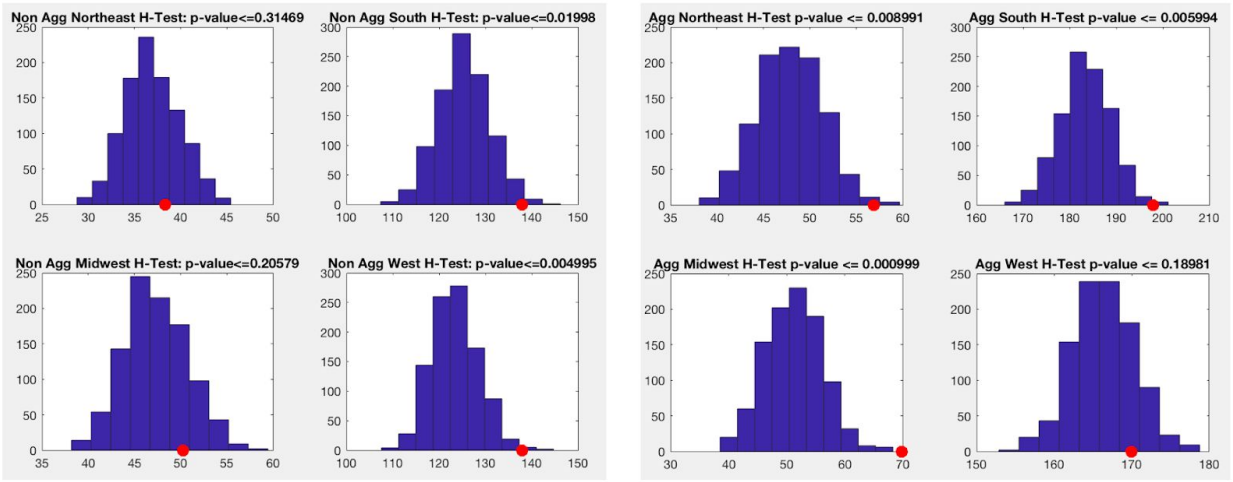
Many other statistics could have been used—-indeed, almost any measure of the difference between two discrete probabilities on five outcomes. I doubt that the result is particularly sensitive to this choice, though I did not try experimenting with other statistics.

We appear to have strong statistical evidence against the notion of police killings being unrelated to race, per se. Rather than moving on to more predictive models, I decided to first dig a bit more deeply, in the hope of gaining more insight into this disturbing result.

Two Approaches to Region-Level Analyses

How do these results hold up when the test is performed more regionally? To explore this, I divided the entries, somewhat arbitrarily, into four disjoint sets based on the locations of the killings: NorthEast, MidWest, West, and South. (I used designations and groupings similar to those defined by the United States Census Bureau.)

The first experiment was to simply apply the same steps that were used to generate the test summarized in the previous figure (representing a nation-wide, analysis), but this time separately to each of the four subregions. This involved nothing more than sub-selecting the appropriate entries (rows) for each of the four regions. The results are summarized in the left-hand panel of the following figure:



The first takeaway is that the significance levels are lower than for the same data grouped into a single national-level test. The reason is simple and predictable: the results for each of the four subregions are based upon fewer samples. This, then, is a demonstration of the difference between type I and type II errors, or, more specifically, the inevitable relationship between sample size and power. Nevertheless, it is possibly instructive to look at the *relative* levels of significance: the “race effect” appears, at least on first inspection, to be weaker in the NorthEast and MidWest than in the West or South. Obviously, this deserves a more detailed analysis before it is to be taken seriously, which I have not done.

Before moving on to some different hypothesis tests, I decided to look again at the regional data but from a somewhat different viewpoint: perhaps the census data at the county level is too refined and noisy to be meaningful. What would the results look like if I replaced the county-level census-derived distribution vectors by the single region-level census-derived distribution? In other words, the null hypothesis becomes this: within a given region the races of the victims, as described by the county-based empirical distributions, were randomly chosen from the census-derived *regional-level* distribution on race. The code is essentially the same, if not in fact a bit simpler since there is now only one census-based probability vector. The results of using this “aggregated” census data, one for each of the four regions, are summarized in the right-hand panel of the previous figure.

Why have the p-values for the NorthWest and MidWest regions become so dramatically lower? There are, I suppose, many possible explanations *but consider the remarkably different conclusions that follow from the two approaches*: Had we only seen the aggregated region-level

data (right-hand panel), say for the NorthEast, we would have been tempted to conclude that race substantially influences whether or not a particular individual will become the victim of a police killing. On the other hand, if we were to then perform the county level analysis, resulting in the left-hand panel, then it would be logical, in the case of the NorthEast, to conclude that there is little discernible evidence, from this data alone, for a direct (causal) connection between race and becoming a victim.

What's going on? One particularly logical explanation might be that either there are more police actions in certain counties or these actions are much more likely to result in police killings than in other counties. Either way, we have a "county effect", as defined in section I. *But, within these counties, the likelihood of a victim being of a particular race is essentially the same as the likelihood of any person in the county being of that race.* In other words, a county effect but no race effect! Perhaps, then, there is an important confounding variable. Perhaps, for example, police are more likely to use deadly force in certain neighborhoods (e.g. poor neighborhoods), and in these neighborhoods there is a systematic shift in the demographics of race (e.g. towards Black people).

The disparity between the two panels gives us another example of what I earlier called the hazards of aggregating inhomogeneous data.

Same data, two similar approaches, two dramatically different takehomes, each with its own implications for addressing the same victim/race disparity!

I need to move on, though I am well aware of the many angles from which this could be studied and the superficiality of my efforts so far. But I learned a lesson: it's not hard to make proper statistical tests, but it can be awfully hard to draw proper conclusions.

Exploring the Alternative Hypothesis

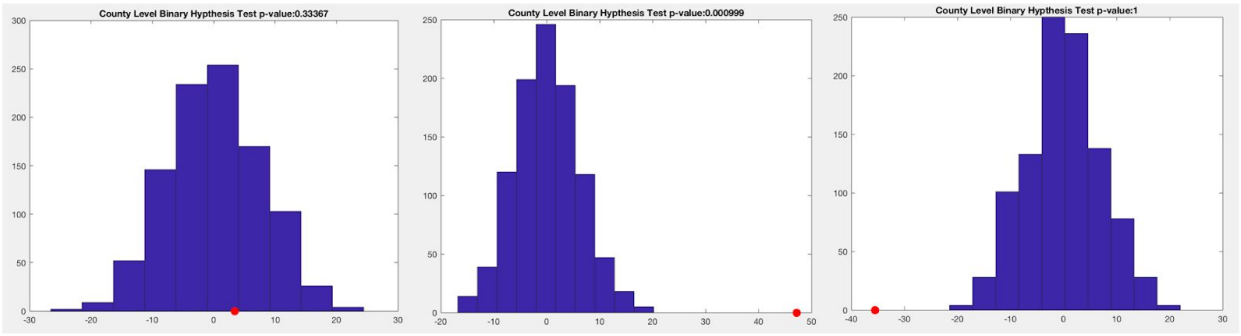
The weight of evidence suggests that being victimized is not race neutral. But there are many ways for two distributions (say the empirical distribution on the race of victims and the local race distribution of the population) to differ. Are the observed differences in our data systematically different in a particular way? Already from just histograms and basic demographic information we have good reason to suspect that victims are systematically less likely to be White and more likely to be Black. One way to take a closer look is to use one-tailed tests of the basic type explored in the previous paragraphs.

Suppose that we were to divide the population into just two race-derived groups, instead of four. To be concrete, suppose we define group W1 to be the subpopulation of White victims, and group W2 to be the remaining victims—-made up of Blacks, Hispanic/Latinos, Asian/Pacific Islanders, and Native Americans. Consider the null hypothesis that the group labels (W1 or W2) of the victims, as described by the county-based empirical distributions, were randomly chosen from the actual distributions on W1 and W2 in those same counties. I want to test this null

hypothesis against the alternative hypothesis of a race effect, i.e. that victims are more likely to be in the W2 (non-white) group than are other people in the county. A suitable statistic, which is essentially a one-tailed version of the statistic used above, would be *the sum, over counties, of the differences, one for each county, between the fraction of victims in the W2 group and the fraction of the population in the W2 group.* Large values of the statistic would be (preliminary!) evidence towards a systematic race-based bias among police victims, i.e. a race effect.

The distribution of the statistic, under the null hypothesis, is again based on the empirical distribution of the statistic derived from 1,000 populations of "virtual" victims. Each virtual victim replaces a real victim, but with a group label (W1 or W2) chosen according to the county proportions of W1 and W2.

The following figure shows the result of this test performed three times, once for each of three different partitionings of race: W1 and W2, as already described, B1 and B2, wherein B2 individuals are Black and B1 are non-Black, and H1 and H2, representing non-Hispanic and Hispanic, respectively. In each case the alternative hypothesis is that the the second group, the "2 group", is overrepresented among victims, and the displayed p-value represents rejection of the null hypothesis (no race effect) in the direction of this alternative (race effect).



The results are all over the place, but not illogical. There is no evidence that victims are disproportionately non-White, strong evidence that victims are disproportionately Black, and no evidence that they are disproportionately Hispanic/Latino. Concerning the Hispanic/Latino population, to the contrary, the evidence appears to indicate that victims are in fact disproportionately non-Hispanic/Latino.

Here is one possible interpretation: There is indeed a race effect in play, whereby victims are more likely to be Black than would be predicted under the local population. As suggested earlier, much of the overrepresentation of Blacks among victims (which is undeniable, given just the histograms---see section I) might be due to the "county effect", in which there are more deaths resulting from police actions in certain counties, most likely the more poor counties, and Blacks are disproportionately represented in the poor counties. At the same time, the one-tailed

test results displayed in the middle panel, strongly suggest that, additionally, Blacks are victimized out of proportion to their representation, even in these select ("county effect") counties.

The situation for Whites is different. They are clearly under-represented among victims relative to their overall numbers, but after "correcting for" their county-level proportions there is no evidence that they are actually "favored" out of proportion to their local representations. And finally, the Hispanic/Latinos are perhaps overrepresented in those same counties in which Blacks are overrepresented, in which case the victimization of Blacks and the favoring of Hispanics might be one and the same thing. But, of course, it is not so hard to come up with alternative explanations!

While it is evident that certain racial groups are disproportionately killed by police, we cannot make any causal inferences just yet. There could be confounding variables. With this in mind, it would be interesting and useful to see how other factors might relate to police killings. For example income levels. It would also be useful to dig deeper, for example what if we just looked at killings where the victim was unarmed. It's unfortunate that we don't know more about the deceased. What levels of education had they obtained? What income bracket were they in? Still, we do have enough data from which to glean more insights. For this purpose, we turn our attention to predictive models.

III. Prediction

What shall we predict? What sorts of models are suggested by the analyses described above? Whereas there are many directions that could be followed, perhaps the most natural, in light of the results seen so far, is to explore the prediction of race as a function of a list of candidate confounding variables, conditioned on the individual having been killed by the police. To narrow the scope even further, and taking into account the evidence that I have presented, I decided to *seek a predictive model for the probability of a person being Black given that they were killed by the police.*

Getting Started

Logistic regression was the natural choice of model given what I was trying to predict a probability. The challenge was choosing the features that would yield the best results. I took a two pronged approach: I used the *stepAIC* stepwise algorithm, included in the MASS library, and the *bestglm* package (and considered the BIC scores as opposed to the AIC scores) to select features.

StepAIC

This tool is incredibly simple to use. I built a regular glm using a majority of the features (including both the census features and the features from the police killings csv). I then used the

stepAIC function to perform forward, backward stepwise selection.[3] The backward stepwise selection was more convincing and settled on 20 variables (out of nearly 60). I list them below. Black, the feature that gives the fraction of the census tract that is Black, was, by far the most significant with a p-value of $8.39e^{-5}$. Age was also deemed to be important as was unemployment. Surprisingly, so was the population of the region to which the county belonged. Intuitively, this seems suspicious, but, I decided to stick with the stepwise results:

```
Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)        4.433e+02  2.409e+02   1.841  0.06567 .
TotalPop          -1.620e-04  6.730e-05  -2.407  0.01607 *
White             -1.839e+00  9.335e-01  -1.970  0.04888 *
Black              4.261e+00  1.083e+00   3.933 8.39e-05 ***
Professional      -4.471e+00  2.407e+00  -1.858  0.06322 .
Service           -4.457e+00  2.406e+00  -1.852  0.06402 .
Office            -4.491e+00  2.408e+00  -1.865  0.06216 .
Construction      -4.407e+00  2.404e+00  -1.833  0.06683 .
Production        -4.472e+00  2.409e+00  -1.856  0.06338 .
Carpool           -7.204e-02  2.912e-02  -2.474  0.01336 *
OtherTransp       -9.793e-02  6.021e-02  -1.626  0.10385
PublicWork        -3.865e-02  2.412e-02  -1.602  0.10916
SelfEmployed      -1.152e-01  4.713e-02  -2.445  0.01450 *
Unemployment       6.917e-02  3.197e-02   2.164  0.03050 *
age               -3.797e-02  1.254e-02  -3.028  0.00246 **
comp_income       -1.227e+00  7.810e-01  -1.571  0.11612
nat_bucket         4.752e-01  2.208e-01   2.152  0.03139 *
college            4.316e+00  2.159e+00   1.999  0.04557 *
State_IncomePerCap 1.115e-04  5.650e-05   1.973  0.04850 *
State_Hispanic    -3.259e+00  1.549e+00  -2.104  0.03540 *
TotalRegion        2.794e-08  9.722e-09   2.874  0.00405 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Coefficients selected using the Backwards stepAIC algorithm

```
    Null deviance: 515.53  on 420  degrees of freedom
Residual deviance: 329.54  on 400  degrees of freedom
AIC: 371.54
```

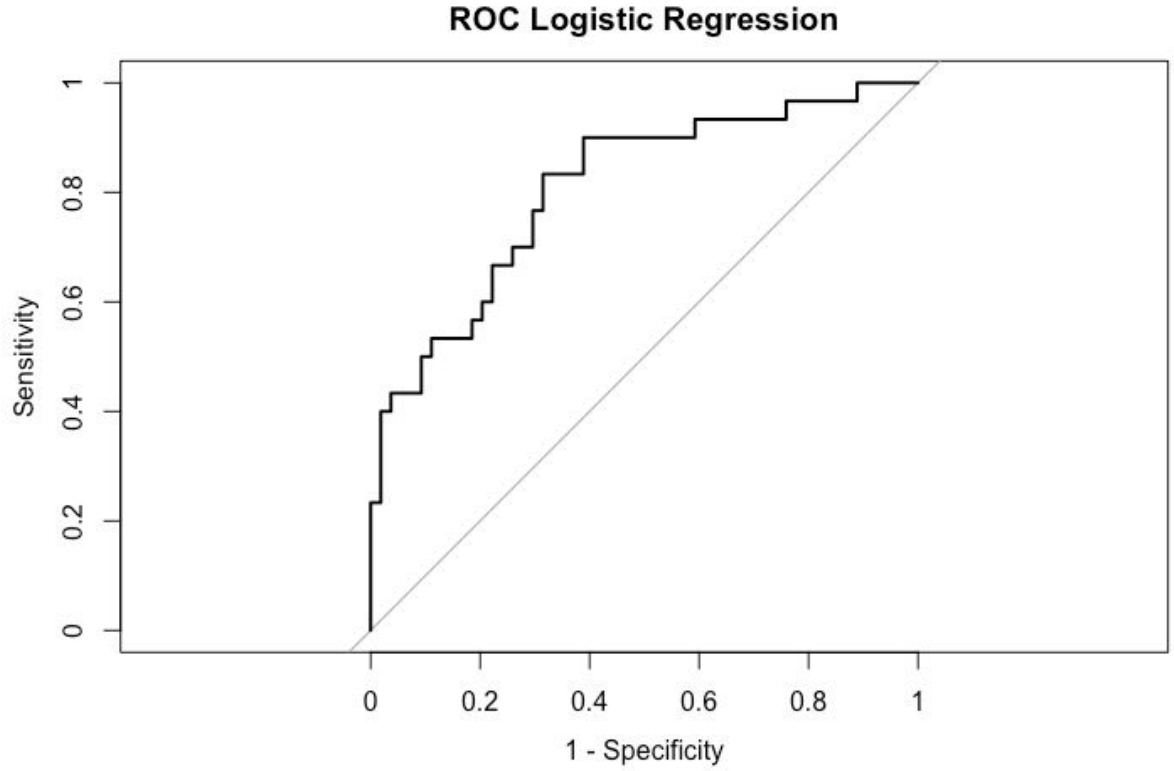
The model’s AIC score. (Which, as we will see, is relatively good.)

[3] In this case, given the complexity and the size of the data, I believe the option “both” is the same as backward stepwise selection.

Jesse Geman
Data 2020: Final Project

Testing the Model

I proceeded to build a logistic regression model using the variables above with a training set made up of 75% of the data. I then predicted on the remaining 25%. Below is the resulting ROC curve based on the predictions for the test data.



The results are respectable but not spectacular. (If we had more data we would see a smoother ROC curve.) The AUC for the curve was a respectable 0.808. (Interpretation: the probability that a randomly chosen victim has a higher predicted probability of being Black than a randomly chosen non-victim, is estimated to be 0.808.)

BestGLM

The bestGLM model was slow to produce results. I ended up setting nvmax to 5. The model selected just 2 features: Black, and Age.

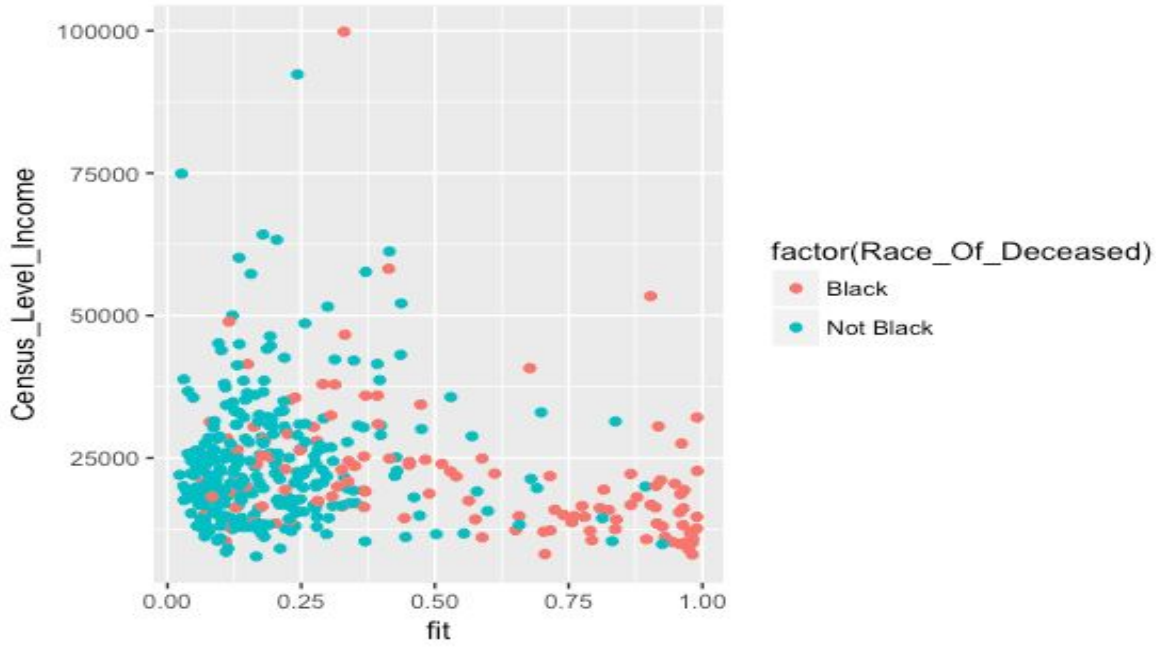
	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-0.7212133	0.41654207	-1.731430	8.337517e-02
Black	5.4578477	0.65232006	8.366825	5.919155e-17
age	-0.0326998	0.01108631	-2.949566	3.182201e-03

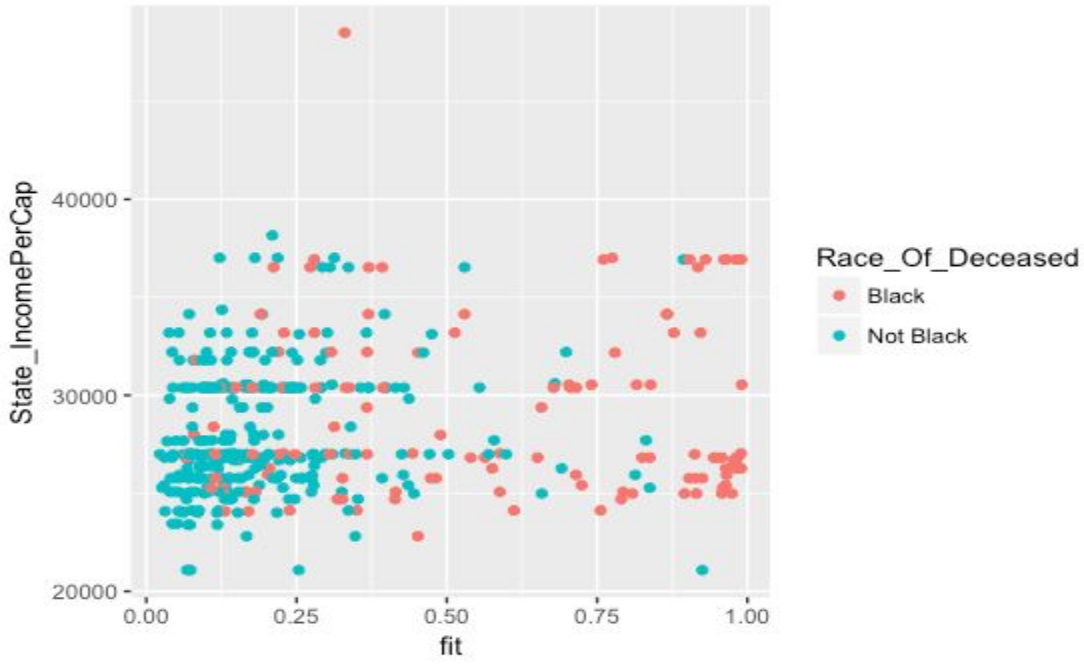
Based on the two logistic models so far, it seems that the race distribution of the census tract to which a victim belongs, and the age of the victim, are the two most important predictors.

Jesse Geman
Data 2020: Final Project

Multilevel Models

Using the features deemed significant by the prior models I built six multilevel models. For the first three I used the state-per-capita-income (rescaled) as a level. And, for the next three I used the regional population, which, as I have already noted, strikes me as a peculiar choice. But it was deemed significant by the stepAIC algorithm. For both sets of levels (the state and the region), I built a varying intercept model, a varying slope, and a varying slope/varying intercept model. Hence, I had six models.





The above images show my predicted probabilities for the the varying state intercept model. In the first image the predictions are plotted against the census level income. In the second they are plotted against the fixed, state level per capita income. The images are not immediately illuminating, and my initial take was that I had done nothing more than to make some pretty plots.

Luckily, upon further inspection, I noticed a few things of possible importance. Both images indicate that police shootings, across all races, happen in areas of lower incomes. This holds true at the census level, and even at the state level. Thus, we might expect to see more people killed by police from demographic groups that are disproportionately low income. This is equivalent to what I have termed a “county effect” in my earlier discussions. It is certainly for Blacks in America.

However, when we look at the bottom image we see that when we hold income constant (at the state level), Blacks are disproportionately represented among those killed by police. This remains true even in the states with higher income levels. But, turning again to the first graph, we see that many of those same individuals correspond to low income census tracts.

(So what’s the point? I’m not sure, but it seemed like a worthwhile tangent.)

Assessing the Multilevel Models

None of the models did particularly well, though they were all better than the models I built for the draft I submitted on the 3rd. Below are some of the results I got from an anova comparison of the models.

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df
model.state.intercept	9	302.06	336.41	-142.03	284.06		
model.region.intercept	12	297.92	343.73	-136.96	273.92	10.1379	3
model.state.slope	18	320.25	388.96	-142.13	284.25	0.0000	6
model.state.slope.intercept	19	319.92	392.45	-140.96	281.92	2.3313	1
model.region.slope	26	319.47	418.72	-133.74	267.47	14.4521	7
model.region.slope.intercept	26	319.47	418.72	-133.74	267.47	0.0000	0

Anova Comparison of My Six Multilevel Models

Both varying intercept models performed better than any of the other models. (Remember that in both AIC and BIC, the parameters, in this setup, will yield lower log likelihoods.) Whether or not to go with the state intercept model or the region intercept model is a toss up.

In addition to the anova comparison test, I calculated the area-under-the-curve scores for each of the models using a holdout test set. I've listed the scores below. The results might be different than the above plot would have you guess.

AUC SCORES
Model: The State Intercept AUC 0.792
Model: State Slope AUC 0.7914
Model: State Slope intercept AUC 0.75
Model: Region Intercept AUC 0.7691
Model: Region Slope AUC 0.7642
Model: Region Slope Intercept AUC 0.7642

The AUC scores serve as a reminder that there are many different criteria for evaluating models, and there is no one sure way to select a model. In this case, the low AIC score and the high AUC value associated with the state intercept model would lead me to choose it.[4] All of this makes sense, and is consistent with what we have seen. The more macroscopic we get, the more we miss. Individual level predictors reign supreme, and above that it is census level predictors.

IV Conclusion

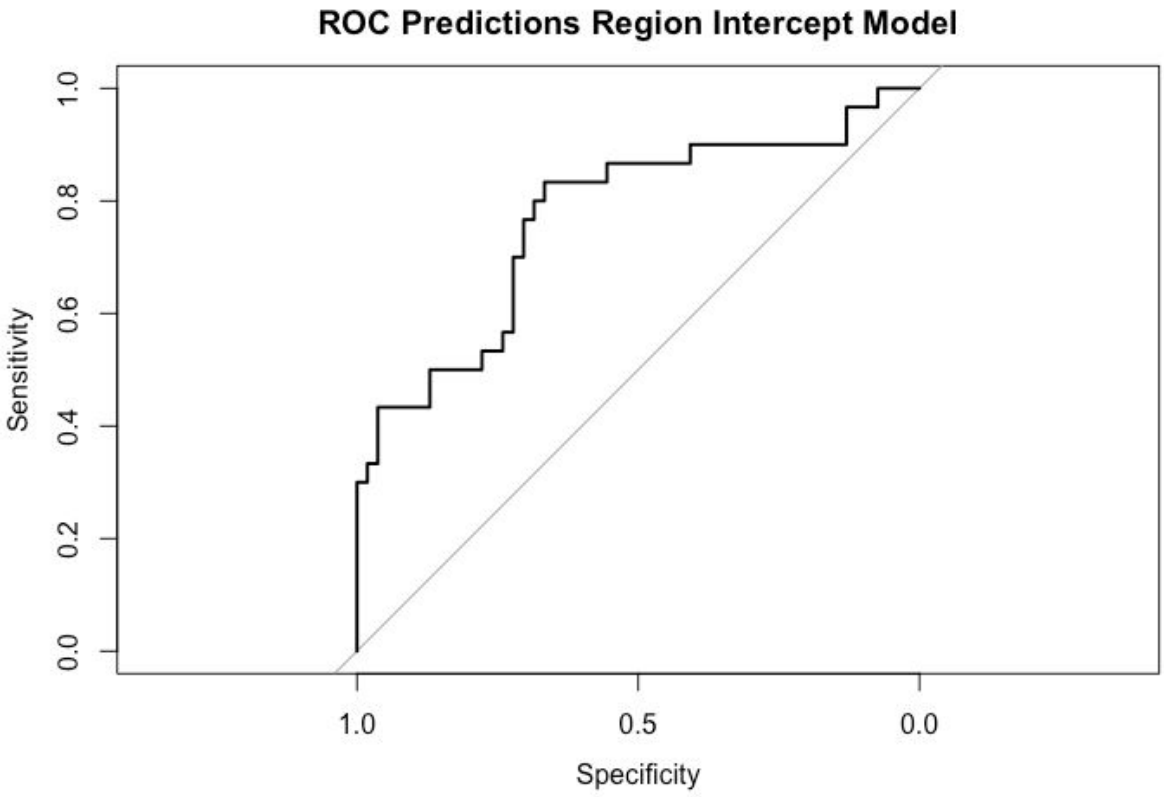
This project was illuminating. We found there to be nearly incontrovertible evidence that police killings occur at rates disproportionate to those one would expect to see in an idealized, "color-blind" world. Specifically, given that a police shooting occurs, the victim is disproportionately likely to be Black. Beyond this, conclusions are hard to come by and any answers seem unlikely to be found through multilevel modeling. If I had more time, which I most certainly do not, I would continue down the road of "exploratory" hypothesis tests before attempting to construct any predictive models.

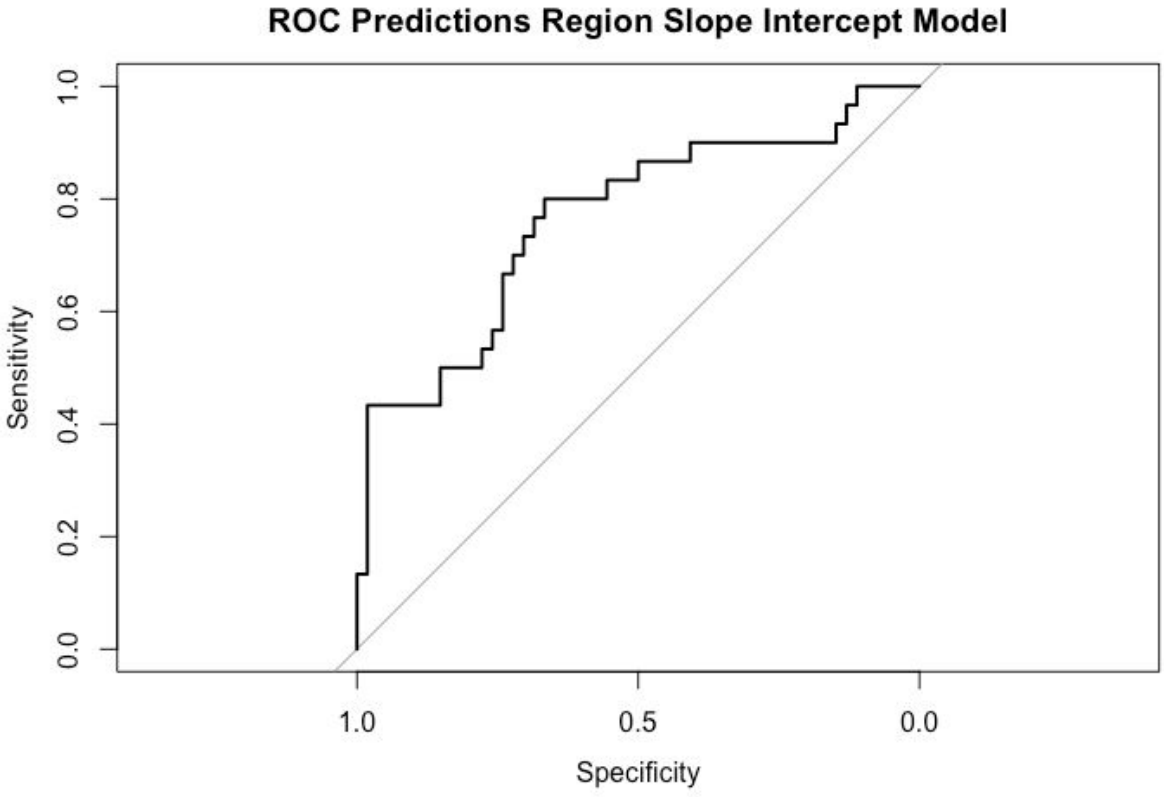
Nonetheless, the evidence for what we have found is strong and a fine example of both how powerful statistical methods and tools can be, and their limitations.

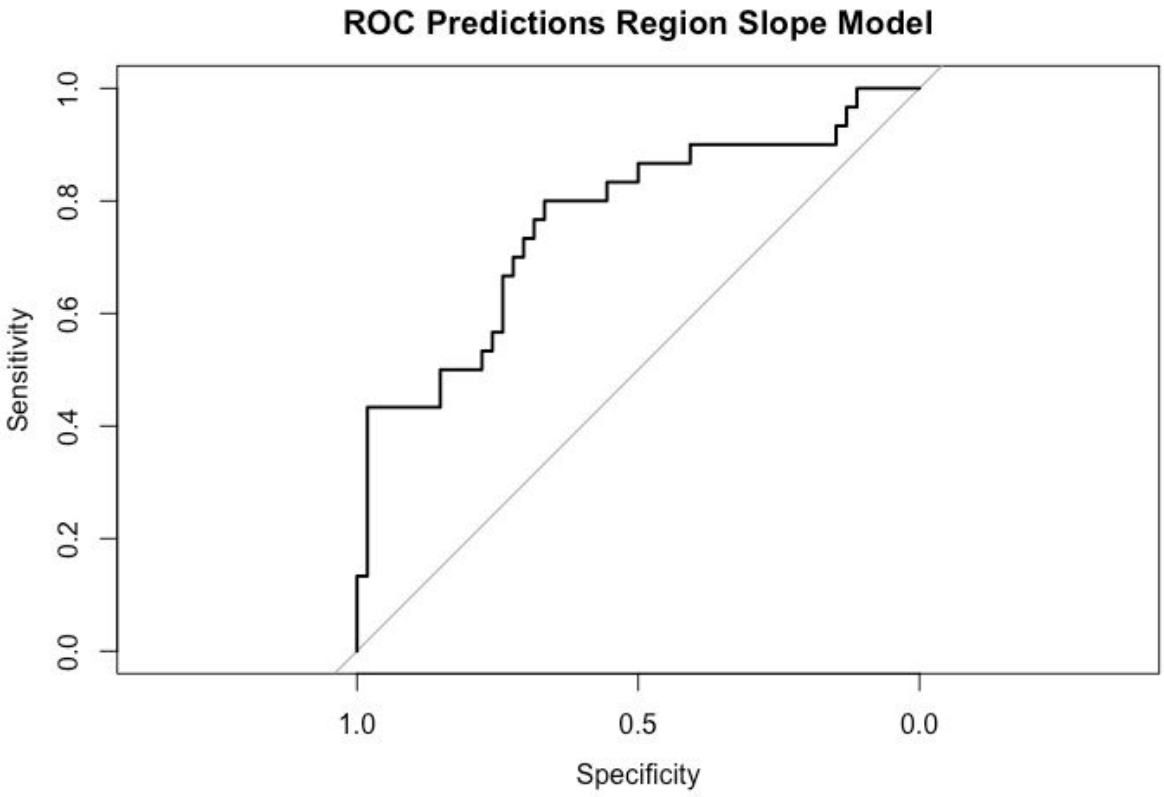
Thanks for a really good course. This project reminded me of just how much I learned.

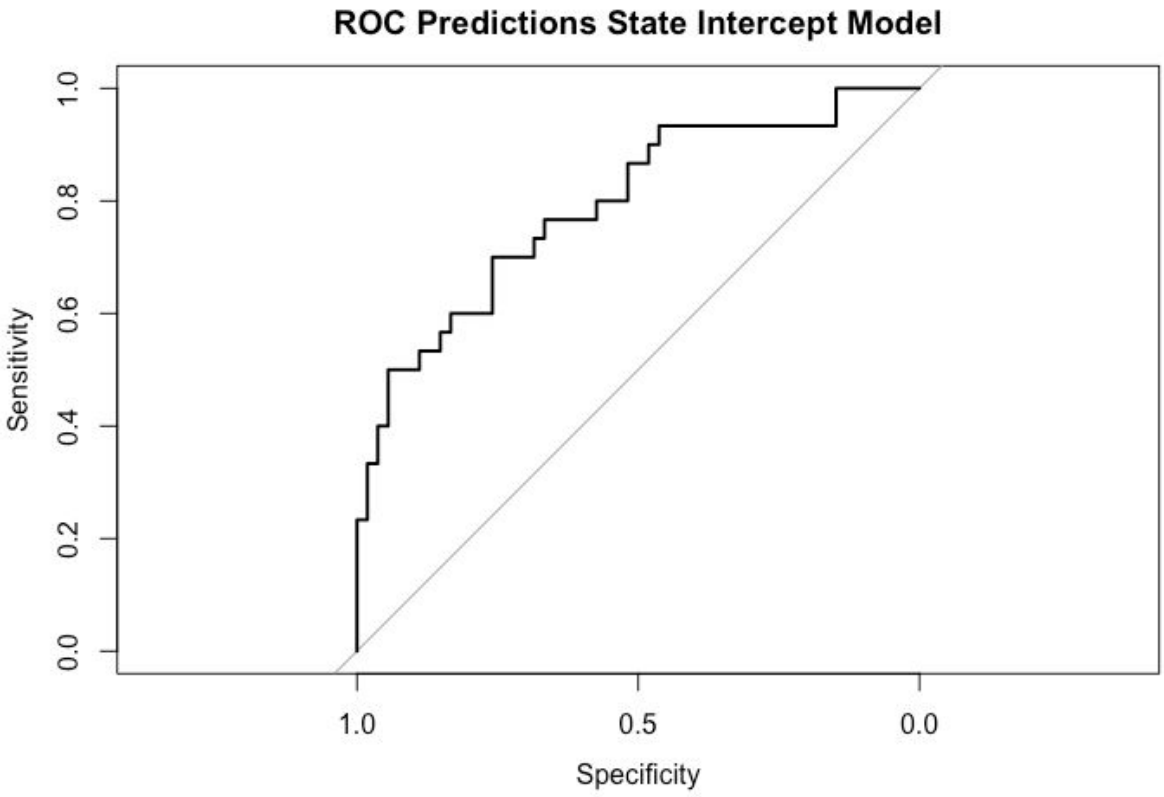
[4] I've included the ROC curves and tables with confidence intervals for the coefficients associated with each model in the appendix as well.

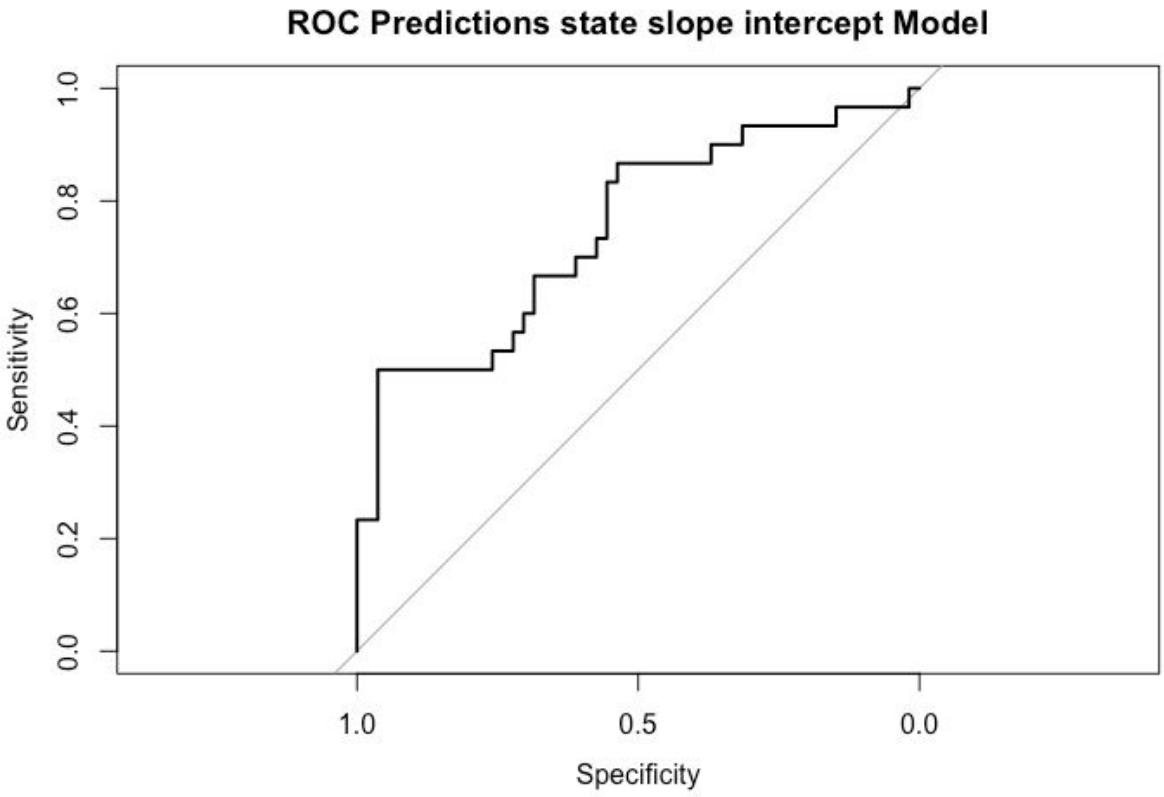
Jesse Geman
Data 2020: Final Project

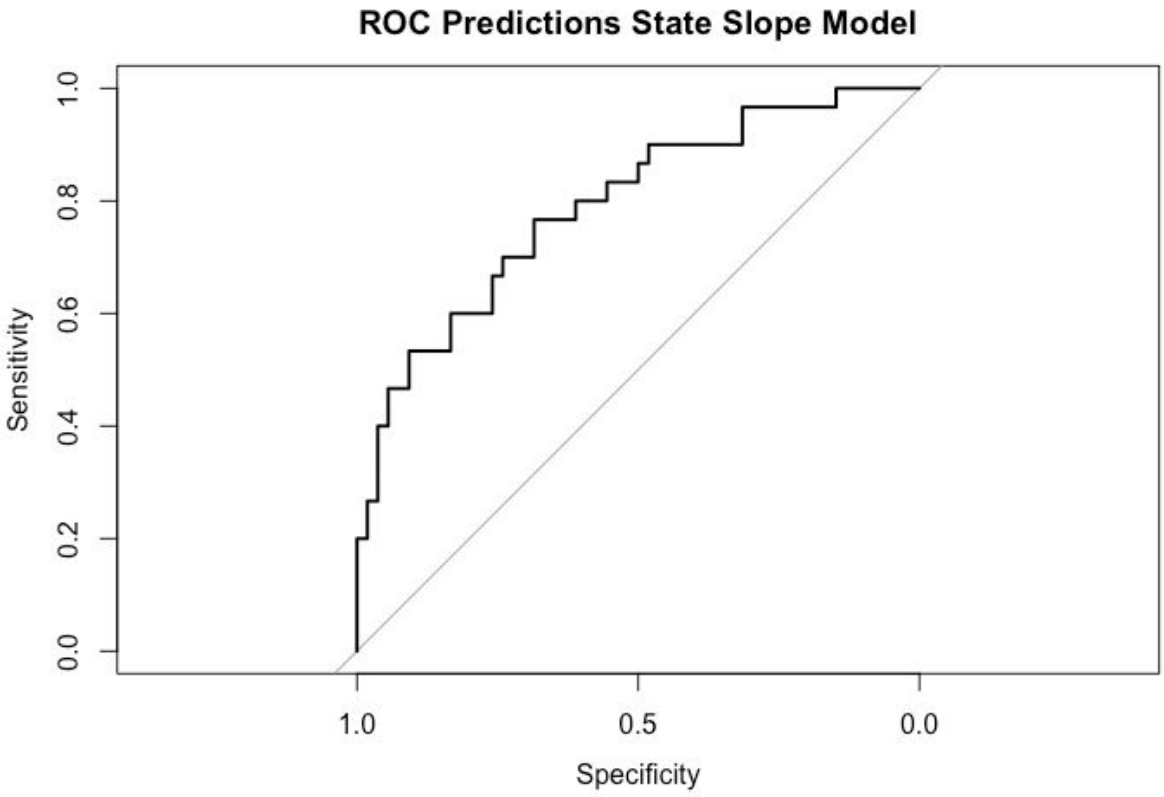

ROC Predictions Region Intercept Model
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.0
0.5
0.0
Specificity


ROC Predictions Region Slope Intercept Model
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.0
0.5
0.0
Specificity


ROC Predictions Region Slope Model
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.0
0.5
0.0
Specificity


ROC Predictions State Intercept Model
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.0
0.5
0.0
Specificity


ROC Predictions state slope intercept Model
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.0
0.5
0.0
Specificity


ROC Predictions State Slope Model
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.0
0.5
0.0
Specificity

Jesse Geman
Data 2020: Final Project

CI's FOR STATE LEVEL INTERCEPT MODEL:

	Est	LL	UL
(Intercept)	-2.20132203	-2.208747194	-2.19389687
Scale_TotalPop	-0.37175988	-0.706589717	-0.03693005
age_scale	-0.51273825	-0.857612440	-0.16786406
Professional	0.01418887	0.008007714	0.02037003
nat_bucket	0.44907974	0.442032567	0.45612692
Black	5.88734642	4.497340120	7.27735273
White	-1.25361706	-1.260661616	-1.24657251
comp_income	-0.97188223	-0.979307318	-0.96445713

CI's FOR STATE LEVEL SLOPE MODEL:

	Est	LL	UL
(Intercept)	-2.23581318	-3.64426743	-0.82735892
Scale_TotalPop	-0.37651185	-0.73386247	-0.01916122
age_scale	-0.53472799	-0.89503633	-0.17441966
Professional	0.01408095	-0.01549561	0.04365752
nat_bucket	0.48720563	0.04966856	0.92474270
Black	6.04335830	3.75590520	8.33081140
White	-1.28230682	-2.81519976	0.25058611
comp_income	-1.06042509	-2.66531499	0.54446481

CI's FOR STATE LEVEL SLOPE/INTERCEPT MODEL:

	Est	LL	UL
(Intercept)	-4.17326142	-6.88753988	-1.45898297
Scale_TotalPop	-0.39893694	-0.78381007	-0.01406381
age_scale	-0.56052964	-0.96636881	-0.15469046
Professional	0.01540595	-0.01456591	0.04537781
nat_bucket	0.57479068	0.06886601	1.08071535
Black	6.91244281	3.68923330	10.13565233
White	-1.00902950	-2.86055044	0.84249143
genderMale	1.62876097	-0.44860827	3.70613021
comp_income	-1.30228260	-3.04468510	0.44011991

CI's FOR REGIONAL LEVEL INTERCEPT MODEL:

	Est	LL	UL
(Intercept)	-2.40750817	-5.09187414	0.276857809
Scale_TotalPop	-0.37738515	-0.72560668	-0.029163616
age_scale	-0.51159383	-0.85604154	-0.167146123
Professional	0.01864476	-0.01145743	0.048746952
nat_bucket	0.52793656	0.07900980	0.976863312
Black	4.67057017	2.34820980	6.992930545
White	-1.84893680	-3.68878384	-0.009089769
genderMale	1.26568182	-0.65431898	3.185682609
comp_income	-1.39271439	-3.05444785	0.269019061
Scale_State_IncomePerCap	0.19894045	-0.18364204	0.581522952
State_Hispanic	-2.78310699	-6.13629404	0.570080069

Jesse Geman
Data 2020: Final Project

CI's FOR REGIONAL LEVEL SLOPE MODEL:

	Est	LL	UL
(Intercept)	-3.15148087	-6.53464219	0.23168045
Scale_TotalPop	-0.33017539	-0.69244403	0.03209326
age_scale	-0.67244218	-1.14238971	-0.20249464
Professional	0.01535673	-0.01571819	0.04643164
nat_bucket	0.63030199	0.15388895	1.10671504
Black	6.44271971	2.52857414	10.35686527
White	-1.65571074	-3.53402464	0.22260316
genderMale	1.18996110	-0.72392918	3.10385139
comp_income	-1.38215605	-3.41764369	0.65333160
Scale_State_IncomePerCap	0.35728567	-0.21745338	0.93202473
State_Hispanic	-2.42487055	-5.91607621	1.06633510

CI's FOR REGIONAL LEVEL SLOPE/INTERCEPT MODEL:

	Est	LL	UL
(Intercept)	-3.15148087	-6.53464219	0.23168045
Scale_TotalPop	-0.33017539	-0.69244403	0.03209326
age_scale	-0.67244218	-1.14238971	-0.20249464
Professional	0.01535673	-0.01571819	0.04643164
nat_bucket	0.63030199	0.15388895	1.10671504
Black	6.44271971	2.52857414	10.35686527
White	-1.65571074	-3.53402464	0.22260316
genderMale	1.18996110	-0.72392918	3.10385139
comp_income	-1.38215605	-3.41764369	0.65333160
Scale_State_IncomePerCap	0.35728567	-0.21745338	0.93202473
State_Hispanic	-2.42487055	-5.91607621	1.06633510

APPENDIX

FEATURE SELECTION AND MULTILEVEL MODELS

```
library(ggplot2)
library(ggcorrplot)
library(corrplot)
library(lme4)
library(nlme)
library(knitr)
library(dplyr)
library(MASS)
library(bestglm)
require(reshape2)
require(compiler)
require(parallel)
require(boot)
require(lattice)


#setwd('Users/stuartgeman/Desktop/data2020/Final Project')
#Get the Police killing data ready (cleaned and remove columns)
#police = read.csv("police_killings_cleaned.csv")
#drop = c("X","name","month","day","year","streetaddress", "city","latitude","lo
ngitude",
  #       "state_fp","county_fp","tract_ce","county_id","namelsad","lawenforceme
ntagency",
 #        "pop","state")
#police = police[,!(names(police) %in% drop)]
#police$age = police$age + 15
#police = na.omit(police)
#police = police[ ! police$raceethnicity %in% "Unknown", ]

police = read.csv("police_killings_cleaned.csv")
police$X = NULL
police$age = police$age + 15
police = na.omit(police)
police = police[ ! police$raceethnicity %in% "Unknown", ]

acs = read.csv("acs2015_census_tract_data.csv")
names(acs)[names(acs) == 'CensusTract'] <- 'geo_id'

#We merge the acs dataframe and police dataframe
total <- merge(acs,police,by="geo_id")
total <-na.omit(total)
total$White = total$White/100
total$Black = total$Black/100
total$Hispanic = total$Hispanic/100
total$Pacific = total$Pacific/100
total$Asian = total$Asian/100
total$Native - total$Native/100


total$raceethnicity <- as.character(total$raceethnicity)
total$raceethnicity[total$raceethnicity== "Hispanic/Latino"] <- "Hispanic"
total$raceethnicity[total$raceethnicity== "Native American"] <- "Native"
total$raceethnicity[total$raceethnicity== "Asian/Pacific Islander"] <-
  "Asian_Pacific"

# Convert race into binary variable
```

```r
total$raceethnicity[which(total$raceethnicity == "Black")] = "1"
total$raceethnicity[which(total$raceethnicity == "White")] = "0"
total$raceethnicity[which(total$raceethnicity == "Hispanic")] = "0"
total$raceethnicity[which(total$raceethnicity == "Asian_Pacific")] = "0"
total$raceethnicity[which(total$raceethnicity == "Hispanic/Latino")] = "0"
total$raceethnicity[which(total$raceethnicity == "Native")] = "0"

total$raceethnicity <- as.numeric(as.character(total$raceethnicity))

#Get the census data ready (including ready to merge with police)
acs = read.csv("acs2015_census_tract_data.csv")
acs <- na.omit(acs)
acs$Asian_Pacific = acs$Asian + acs$Pacific

names(acs)[names(acs) == 'CensusTract'] <- 'geo_id'
#get rid of columns that are redundant after merge

total$Men = total$Men/total$TotalPop
total$Women = NULL
#state_pop = aggregate(TotalPop~State,acs,sum)
#rownames(total) <- 1:nrow(total)

#Get the state level Aggregate Data
#Notice we don't do Native or Asian, since these features correspond to very few killings
#and will simply increase the "dependency" between our races
acs$AsianPacific = acs$Asian + acs$Pacific
state_levels = acs %>%
  group_by(State) %>%
  summarise(TotalState = sum(TotalPop),
            #Total_State_Women = sum(Women)/TotalState,
            State_Men = sum(Men)/TotalState,
            State_Unemployment =  (sum(TotalPop*Unemployment*.01))/TotalState,
            State_IncomePerCap = sum(TotalPop*IncomePerCapErr)/TotalState,
            State_Poverty = (sum(TotalPop*Poverty*.01))/TotalState,
            State_Drive = (sum(TotalPop*Drive*.01))/TotalState,
            State_Child_Poverty = (sum(TotalPop*.01*ChildPoverty))/TotalState,
            State_Hispanic = (sum(TotalPop*(Hispanic*.01)))/TotalState,
            State_Black = (sum(TotalPop*(Black*.01)))/TotalState,
            State_White = (sum(TotalPop*(White*.01)))/TotalState,
            State_Asian_Pacific = (sum(TotalPop*(AsianPacific*.01)))/TotalState,

            State_Native = (sum(TotalPop*(Native*.01)))/TotalState)


#I will now subset the states into regions Northeast, South, West, Midwest (I had a fifth, Mountain,
#but there weren't enough observations for it so I distributed the mountain states into the others)
       NorthEast = c("Connecticut", "Maine", "Massachusetts", "New Hampshire", "Rhode Island", "Vermont",
                     "New Jersey", "New York","Delaware","District of Columbia", "Maryland",
                     "Pennsylvania")

       South = c("Alabama", "Florida", "Georgia", "Kentucky",
                 "Mississippi", "North Carolina", "South Carolina", "Tennessee","Virginia",
                 "West Virginia","Arkansas", "Louisiana","Oklahoma", "Texas")

```

```
108.       Midwest = c("Illinois", "Indiana", "Michigan", "Minnesota", "Ohio", "Wis
    consin",
109.                   "Iowa", "Kansas", "Missouri", "Nebraska","North Dakota", "So
    uth Dakota")
110.
111.
112.       West = c("Arizona", "California", "Hawaii", "Nevada","Alaska",
113.                "Idaho", "Oregon", "Washington","Colorado","New Mexico","Utah",
    "Montana","Wyoming")
114.
115.
116.       Demographics_NorthEast =  subset(state_levels, (state_levels$State %in%
    NorthEast))
117.
118.       Demographics_South = subset(state_levels, (state_levels$State %in% South
    ))
119.
120.       Demographics_Midwest = subset(state_levels, (state_levels$State %in% Mid
    west))
121.
122.       Demographics_West = subset(state_levels, (state_levels$State %in% West))

123.
124.
125.       #For Each Demographic Region We aggregate the demographic data
126.       Demographics_NorthEast = Demographics_NorthEast %>%
127.         summarise(TotalRegion = sum(TotalState),
128.                   Region_Men = sum(TotalState*State_Men)/TotalRegion,
129.                   Region_Unemployment =  (sum(TotalState*State_Unemployment))/
    TotalRegion,
130.                   Region_IncomePerCap = sum(TotalState*State_IncomePerCap)/Tot
    alRegion,
131.                   Region_Poverty = (sum(TotalState*State_Poverty))/TotalRegion
    ,
132.                   Region_Hispanic = (sum(TotalState*State_Hispanic))/TotalRegi
    on,
133.                   Region_Black = (sum(TotalState*State_Black))/TotalRegion,
134.                   Region_White = (sum(TotalState*State_White))/TotalRegion,
135.                   Region_Drive = (sum(TotalState*State_Drive))/TotalRegion,
136.                   Region_Child_Poverty = (sum(TotalState*State_Child_Poverty))
    /TotalRegion)
137.
138.
139.       Demographics_Midwest = Demographics_Midwest %>%
140.         summarise(TotalRegion = sum(TotalState),
141.                   Region_Men = sum(TotalState*State_Men)/TotalRegion,
142.                   Region_Unemployment =  (sum(TotalState*State_Unemployment))/
    TotalRegion,
143.                   Region_IncomePerCap = sum(TotalState*State_IncomePerCap)/Tot
    alRegion,
144.                   Region_Poverty = (sum(TotalState*State_Poverty))/TotalRegion
    ,
145.                   Region_Hispanic = (sum(TotalState*State_Hispanic))/TotalRegi
    on,
146.                   Region_Black = (sum(TotalState*State_Black))/TotalRegion,
147.                   Region_White = (sum(TotalState*State_White))/TotalRegion,
148.                   Region_Drive = (sum(TotalState*State_Drive))/TotalRegion,
149.                   Region_Child_Poverty = (sum(TotalState*State_Child_Poverty))
    /TotalRegion)
150.
151.       Demographics_South = Demographics_South %>%
```

```
        summarise(TotalRegion = sum(TotalState),
                  Region_Men = sum(TotalState*State_Men)/TotalRegion,
                  Region_Unemployment =  (sum(TotalState*State_Unemployment))/TotalRegion,
                  Region_IncomePerCap = sum(TotalState*State_IncomePerCap)/TotalRegion,
                  Region_Poverty = (sum(TotalState*State_Poverty))/TotalRegion,
                  Region_Hispanic = (sum(TotalState*State_Hispanic))/TotalRegion,
                  Region_Black = (sum(TotalState*State_Black))/TotalRegion,
                  Region_White = (sum(TotalState*State_White))/TotalRegion,
                  Region_Drive = (sum(TotalState*State_Drive))/TotalRegion,
                  Region_Child_Poverty = (sum(TotalState*State_Child_Poverty))/TotalRegion)


      Demographics_West = Demographics_West %>%
        summarise(TotalRegion = sum(TotalState),
                  Region_Men = sum(TotalState*State_Men)/TotalRegion,
                  Region_Unemployment =  (sum(TotalState*State_Unemployment))/TotalRegion,
                  Region_IncomePerCap = sum(TotalState*State_IncomePerCap)/TotalRegion,
                  Region_Poverty = (sum(TotalState*State_Poverty))/TotalRegion,
                  Region_Hispanic = (sum(TotalState*State_Hispanic))/TotalRegion,
                  Region_Black = (sum(TotalState*State_Black))/TotalRegion,
                  Region_White = (sum(TotalState*State_White))/TotalRegion,
                  Region_Drive = (sum(TotalState*State_Drive))/TotalRegion,
                  Region_Child_Poverty = (sum(TotalState*State_Child_Poverty))/TotalRegion)

      drop = c("name","month","day","year","streetaddress", "city","latitude", "longitude",
               "state_fp","county_fp","tract_ce","county_id","namelsad","lawenforcementagency",
                "pop","state")

      #Merge Regional Data On
      total = total[,!(names(total) %in% drop)]
      total = merge(total, state_levels, by= "State")

      totalNE = subset(total, (total$State %in% NorthEast))
      totalNE$Region = "NE"
      Demographics_NorthEast$Region = "NE"
      totalNE = merge(totalNE, Demographics_NorthEast, by= "Region")

      totalS = subset(total, (total$State %in% South))
      totalS$Region = "South"
      Demographics_South$Region = "South"
      totalS = merge(totalS, Demographics_South, by= "Region")

      totalMid = subset(total, (total$State %in% Midwest))
      totalMid$Region = "Mid"
      Demographics_Midwest$Region = "Mid"
      totalMid = merge(totalMid, Demographics_Midwest, by= "Region")

      totalW = subset(total, (total$State %in% West))
      totalW$Region = "West"
```

```
201.       Demographics_West$Region = "West"
202.       totalW = merge(totalW, Demographics_West, by= "Region")
203.       total = rbind(totalNE,totalMid,totalS,totalW)
204.       #We Drop colnames that are useless
205.       STATE = total$State
206.       drop = c("Region", "State", "geo_id", "County","IncomeErr", "IncomePerCa
    pErr","Native","Asian",
207.                "Employed", "PrivateWork","WorkAtHome","share_black", "share_hi
    spanic","share_white",
208.                "p_income","h_income","county_income", "TotalState.x", "State_M
    en.x","State_Unemployment.x",
209.                "State_IncomePerCap.x",  "State_Poverty.x","State_Hispanic.x","
    State_Black.x","State_White.x",
210.                "State_Drive.x","State_Child_Poverty.x", "TotalState.y","State_
    Men.y","State_Unemployment.y","State_IncomePerCap.y",
211.                "State_Poverty.y", "State_Hispanic.y","State_Black.y","State_Wh
    ite.y","State_Drive.y","State_Child_Poverty.y", "cause")
212.       total = total[,!(names(total) %in% drop)]
213.
214.
215.       #This one is not clear so I leave it out of the above list (same goes fo
    r cause maybe..)
216.       total$armed = NULL
217.       total$pov = NULL
218.       #Turn Gender into 1's and zeros
219.       cols <- sapply(total, is.logical)
220.       total[,cols] <- lapply(total[,cols], as.numeric)
221.       #Okay Time for some feature selection:
222.       #We will use logistic regression to figure out which features we should

223.       #consider using.
224.       full <- glm(raceethnicity ~.,data = total, family = binomial())
225.       #Degrees of Freedom: 420 Total (i.e. Null);  375 Residual
226.       #Null Deviance:     515.5
227.       #Residual Deviance: 318.1   AIC: 410.1
228.       #Not good but better than our scores multilevel
229.
230.       step <- stepAIC(full,direction = "both", trace = FALSE)
231.       step$anova
232.
233.       #Final Selection Variables:
234.       #Final Model:
235.        #raceethnicity ~ TotalPop + White + Black + Professional + Service +
236.          #Office + Construction + Production + Carpool + OtherTransp +
237.          #PublicWork + SelfEmployed + Unemployment + age + comp_income +
238.          #nat_bucket + college + State_IncomePerCap + State_Hispanic +
239.          #TotalRegion
240.
241.
242.       forward <- stepAIC(full,direction = "forward", trace = FALSE)
243.       forward$anova
244.       #Somewhat dissapointingly gives the same variables back for final
245.       #model (in fact, if both can't be applied reverts to backward)!
246.
247.       backward <-stepAIC(full, direction = "backward", trace = FALSE)
248.       backward$anova
249.       #Final Selection Variables:
250.       #Final Model:
251.         #raceethnicity ~ TotalPop + White + Black + Professional + Service +

252.         #Office + Construction + Production + Carpool + OtherTransp +
```

```
253.         #PublicWork + SelfEmployed + Unemployment + age + comp_income +
254.         #nat_bucket + college + State_IncomePerCap + State_Hispanic +
255.         #TotalRegion
256.       #We now build a normal logistic model with some of the recommended varia
   bles.
257.       #A TRAIN TEST SPLIT
258.       ## 75% of the sample size
259.       smp_size <- floor(0.75 * nrow(total))
260.
261.       ## set the seed to make your partition reproducible
262.       set.seed(123)
263.       train_ind <- sample(seq_len(nrow(total)), size = smp_size)
264.
265.       train <- total[train_ind, ]
266.       test <- total[-train_ind, ]
267.
268.       glm.logit = glm(raceethnicity ~ White + Black + Professional + Service +
    Office +
269.                         Construction + Production + Carpool + OtherTransp+ age
     +
270.                         comp_income + nat_bucket + college +State_IncomePerCap
    + State_Hispanic + TotalRegion,
271.                       family = binomial, data = train)
272.
273.       library(gridExtra)
274.       library(pROC)
275.       p <- predict(glm.logit, newdata=test, type="response")
276.       plot(roc(test$raceethnicity,p), legacy.axes = TRUE)
277.       auc_logit =auc(roc(test$raceethnicity,p))
278.       title(main = "ROC Logistic Regression", line = +3)
279.
280.       #The deviance residuals for the predictions on the trianed data)
281.       #gg <- qplot(x = fitted(glm.logit), y = residuals(glm.logit)) +
282.         #geom_smooth(method = "glm", se = FALSE) +
283.        # geom_point(alpha = 0.3, size = 3) +
284.         #theme_bw()
285.
286.       #print(gg)
287.
288.
289.       #test <- test %>%
290.        # mutate(test$raceethnicity = test$raceethnicity)
291.       #test <- test %>%
292.        # mutate(predicted.prob = p)
293.       #test <- test %>%
294.        # mutate(predicted = ifelse(predicted.prob >0.5, 1, 0))
295.       #table(test$raceethnicity, test$predicted)
296.       #table(test$raceethnicity)
297.       #roc(test$raceethnicity, test$predicted.prob)
298.       #png("1d.png", width = 400, height =400, res = 110)
299.       #ggplot(test, aes(d = raceethnicity, m = predicted.prob)) +
300.        # geom_abline(slope = 1, intercept = 0) +
301.         #labs(x = "1 - Specificity", y = "Sensitivity")
302.       #dev.off()
303.
304.       #ggplot(test$raceethnicity, p) +
305.        # labs(x = "1 - Specificity", y = "Sensitivity")
306.
307.       Xy=total
308.       Xy$raceethnicity = NULL
309.
```

```
310.        Xy = cbind(Xy, total$raceethnicity)
311.        Xy$gender = NULL
312.        myglm <-bestglm(Xy,nvmax = 8)
313.        Xy = Xy[,c("TotalPop", "White","Black","Professional","Unemployment",
314.                        "Service","Construction","State_IncomePerCap","comp_income
    ",
315.                        "age","college","Office", "nat_bucket","IncomePerCap","Sta
    te_Hispanic")]
316.        Xy$black_killed = total$raceethnicity
317.        #Xy$nonBlack = 1 - total$raceethnicity
318.        myglm_logit <-bestglm(Xy,family = binomial(), nvmax = 5)
319.
320.
321.        #Now that we have looked at a few different glm's to predict whether a p
    erson
322.        #shot was black or not we build a multilevel model. We start by scaling
    the data where appropriate
323.        total$Scale_State_IncomePerCap = scale(total$State_IncomePerCap)
324.        #total$State_Hispanic = scale(total$State_Hispanic)
325.        total$Scale_TotalPop = scale(total$TotalPop)
326.        total$age_scale = scale(total$age)
327.        total$Scale_TotalRegion = scale(total$TotalRegion)
328.
329.
330.        #A TRAIN TEST SPLIT
331.        ## 75% of the sample size
332.        smp_size <- floor(0.8 * nrow(total))
333.        set.seed(123)
334.        train_ind <- sample(seq_len(nrow(total)), size = smp_size)
335.        total$State = STATE
336.
337.        train <- total[train_ind, ]
338.        test <- total[-train_ind, ]
339.
340.        #Test with states that were includeded in the training data
341.        test = test[test$State %in% unique(train$State),]
342.
343.        #STATE INTERCEPT
344.        model.state.intercept = glmer(raceethnicity ~ Scale_TotalPop + age_scale
    + Professional+ nat_bucket+Black + White  +comp_income
345.                                        + (1|Scale_State_IncomePerCap),
346.                              family = binomial("logit"),REML = FALSE, data=train,

347.                              glmerControl(optimizer = "bobyqa", optCtrl = list(max
    fun = 200000)))
348.
349.        se1 <- sqrt(diag(vcov(model.state.intercept)))
350.        # table of estimates with 95% CI
351.        (tab <- cbind(Est = fixef(model.state.intercept),
352.                        LL = fixef(model.state.intercept) -
    1.96 * se1, UL = fixef(model.state.intercept) + 1.96 *se1))
353.
354.        #print(model.state.intercept, corr = FALSE)
355.        predictions.state.intercept <-
    predict(model.state.intercept, test, type = "response")
356.        roc_model.state.intercept <-
    roc(test$raceethnicity ~ predictions.state.intercept)
357.        auc1 = auc(roc_model.state.intercept)
358.        plot(roc_model.state.intercept)
359.        title(main = "ROC Predictions State Intercept Model", line = +3)
360.
```

```
361.
362.
363.
364.       #We have a STATE slope model
365.       model.state.slope = glmer(raceethnicity ~ Scale_TotalPop + age_scale+ Pr
    ofessional+ nat_bucket+Black + White  +comp_income
366.                                 + (Black+age_scale + nat_bucket|Scale_State_Inc
    omePerCap),
367.                                 family = binomial("logit"),REML = FALSE, data=t
    rain,
368.                                 glmerControl(optimizer = "bobyqa", optCtrl = li
    st(maxfun = 200000)))
369.
370.
371.
372.       se2 <- sqrt(diag(vcov(model.state.slope)))
373.       # table of estimates with 95% CI
374.       (tab <- cbind(Est = fixef(model.state.slope),
375.                     LL = fixef(model.state.slope) -
     1.96 * se2, UL = fixef(model.state.slope) + 1.96 *se2))
376.       print(model.state.slope, corr = FALSE)
377.
378.       predictions.state.slope <-
     predict(model.state.slope, test, type = "response")
379.       roc_model.state.slope <-
     roc(test$raceethnicity ~ predictions.state.slope)
380.       auc2 = auc(roc_model.state.slope)
381.       plot(roc_model.state.slope)
382.       title(main = "ROC Predictions State Slope Model", line = +3)
383.
384.       # STATE SLOPE
385.       model.state.slope.intercept = glmer(raceethnicity ~ Scale_TotalPop + age
    _scale+ Professional+ nat_bucket+Black + White + gender +comp_income
386.                                 + (1+Black+age_scale + nat_bucket|Scale_State_
    IncomePerCap),
387.                                 family = binomial("logit"),REML = FALSE, data=
    train,
388.                                 glmerControl(optimizer = "bobyqa", optCtrl = l
    ist(maxfun = 200000)))
389.
390.       se3 <- sqrt(diag(vcov(model.state.slope.intercept)))
391.       # table of estimates with 95% CI
392.       (tab <- cbind(Est = fixef(model.state.slope.intercept),
393.                     LL = fixef(model.state.slope.intercept) -
     1.96 * se3, UL = fixef(model.state.slope.intercept) + 1.96 *se3))
394.       print(model.state.slope.intercept, corr = FALSE)
395.       predictions.state.slope.intercept <-
     predict(model.state.slope.intercept, test, type = "response")
396.       roc_model.state.slope.intercept <-
     roc(test$raceethnicity ~ predictions.state.slope.intercept)
397.       auc3 = auc(roc_model.state.slope.intercept)
398.       plot(roc_model.state.slope.intercept)
399.       title(main = "ROC Predictions state slope intercept Model" ,line = +3)
400.
401.       anova(model.state.intercept,model.state.slope,model.state.slope.intercep
    t)
402.
403.       #REGION INTERCEPT
404.       model.region.intercept = glmer(raceethnicity ~ Scale_TotalPop+age_scale+
     Professional+ nat_bucket+Black + White + gender +comp_income
```

```
                                                + Scale_State_IncomePerCap+ State_Hi
spanic+ (1|Scale_TotalRegion),
                                                family = binomial("logit"),REML = FA
LSE, data=train,
                                                glmerControl(optimizer = "bobyqa", o
ptCtrl = list(maxfun = 200000)))

       se4 <- sqrt(diag(vcov(model.region.intercept)))
       # table of estimates with 95% CI
       (tab <- cbind(Est = fixef(model.region.intercept),
                     LL = fixef(model.region.intercept) -
 1.96 * se4, UL = fixef(model.region.intercept) + 1.96 *se4))
       print(model.region.intercept, corr = FALSE)

       predictions.model.region.intercept <-
 predict(model.region.intercept, test, type = "response")
       roc_model.region.intercept <-
 roc(test$raceethnicity ~ predictions.model.region.intercept)
       auc4= auc(roc_model.region.intercept)
       plot(roc_model.region.intercept)
       title(main = "ROC Predictions Region Intercept Model", line = +3)



       #REGION VARYING SLOPE

       model.region.slope = glmer(raceethnicity ~ Scale_TotalPop+age_scale+ Pro
fessional+ nat_bucket+Black + White + gender +comp_income
                                          + Scale_State_IncomePerCap+ State_Hispani
c+ (Scale_State_IncomePerCap+ comp_income+age_scale + Black|Scale_TotalRegion),

                                          family = binomial("logit"),REML = FALSE,
data=train,
                                          glmerControl(optimizer = "bobyqa", optCtr
l = list(maxfun = 200000)))

       se5 <- sqrt(diag(vcov(model.region.slope)))
       # table of estimates with 95% CI
       (tab <- cbind(Est = fixef(model.region.slope),
                     LL = fixef(model.region.slope) -
 1.96 * se5, UL = fixef(model.region.slope) + 1.96 *se5))
       print(model.region.slope, corr = FALSE)
       prediction.model.region.slope <-
 predict(model.region.slope, test, type = "response")
       roc_model.region.slope <-
 roc(test$raceethnicity ~ prediction.model.region.slope)
       auc5= auc(roc_model.region.slope)
       plot(roc_model.region.slope)
       title(main = "ROC Predictions Region Slope Model", line = +3)

       #REGION VARYING SLOPE INTERCEPT

       model.region.slope.intercept = glmer(raceethnicity ~Scale_TotalPop + age
_scale+ Professional+ nat_bucket+Black + White + gender +comp_income
                                     + Scale_State_IncomePerCap+ State_Hispanic+ (
1 +Scale_State_IncomePerCap+ comp_income+age_scale + Black|Scale_TotalRegion),

                                     family = binomial("logit"),REML = FALSE, data
=train,
```

```
                                    glmerControl(optimizer = "bobyqa", optCtrl =
    list(maxfun = 200000)))

       se6 <- sqrt(diag(vcov(model.region.slope.intercept)))
       # table of estimates with 95% CI
       (tab <- cbind(Est = fixef(model.region.slope.intercept),
                     LL = fixef(model.region.slope.intercept) -
     1.96 * se6, UL = fixef(model.region.slope.intercept) + 1.96 *se6))
       print(model.region.slope.intercept, corr = FALSE)
       predictions.region.region.slope.intercept <-
     predict(model.region.slope.intercept, test, type = "response")
       roc_region.slope.intercept <-
     roc(test$raceethnicity ~ predictions.region.region.slope.intercept)
       auc6= auc(roc_region.slope.intercept)
       plot(roc_region.slope.intercept)
       title(main = "ROC Predictions Region Slope Intercept Model", line = +3)


       anova(model.state.intercept,model.state.slope,model.state.slope.intercep
    t,
             model.region.intercept,model.region.slope ,model.region.slope.inte
    rcept)
```

FIGURES/EDA

```
library(arm)
library(sjPlot)
library(sjmisc)
library(lme4)
png("4a.png", width = 600, height = 450, res = 100)
par(mai = c(0.8, 0.8, 0.1, 0.1))
plot(coef(fit.sub.4a)$county[,1], type = "l", lwd = 2, col = "pink",
          ylab = "County-level Intercepts", xlab = "Counties in the Subset")
 tmp <- rownames(coef(fit.full.4a)$county) %in%
  rownames(coef(fit.sub.4a)$county)
lines(coef(fit.full.4a)$county[tmp,1], lty = 2, lwd = 2, col = "yellowgreen")
 legend("bottomleft", c("Subset", "Entire"),
          49 lty = 1:2,
          50 col = c("pink", "yellowgreen"),
          51 lwd = 2)
dev.off()


coef(model.state.intercept)$State_IncomePerCap[,1]

gg <-
 ggplot(total, aes(x = State_IncomePerCap, y = raceethnicity, group = State_Inco
mePerCap)) +
 # geom_line(aes(y = PooledPredictions), color = "darkgrey") +
  #geom_line(aes(y = VaryingInterceptPredictions), color = "blue") +
  #geom_line(aes(y = VaryingSlopePredictions), color = "red") +
  #geom_line(aes(y = InteractionPredictions), color = "black") +
  geom_point(alpha = 0.3, size = 3) +
  facet_wrap(~total) +
  theme_bw()

fit <- fitted(model.state.intercept, total, type="response")
State_IncomePerCap = total$State_IncomePerCap
Race_Of_Deceased = total$raceethnicity
```

```
Race_Of_Deceased = ifelse(Race_Of_Deceased == 0, "Not Black", "Black")
df = data.frame(State_IncomePerCap, Race_Of_Deceased, fit)

ggplot(df, aes(fit, State_IncomePerCap)) +
  geom_point(aes(color = Race_Of_Deceased))


fit <- fitted(model.state.intercept, total, type="response")
Census_Level_Income = total$IncomePerCap
Race_Of_Deceased = total$raceethnicity
Race_Of_Deceased = ifelse(Race_Of_Deceased == 0, "Not Black", "Black")

df = data.frame(Census_Level_Income, Race_Of_Deceased, fit)

ggplot(df, aes(fit, Census_Level_Income)) +
  geom_point(aes(color = factor(Race_Of_Deceased)))


fit <- fitted(model.state.intercept, total, type="response")
State_IncomePerCap = total$State_IncomePerCap
Race_Of_Deceased = total$raceethnicity
Race_Of_Deceased = ifelse(Race_Of_Deceased == 0, "Not Black", "Black")
df = data.frame(State_IncomePerCap, Race_Of_Deceased, fit)

ggplot(df, aes(fit, State_IncomePerCap)) +
  geom_point(aes(color = Race_Of_Deceased))


fit <- fitted(model.state.intercept, total, type="response")
Census_Level_Income = total$IncomePerCap
Race_Of_Deceased = total$raceethnicity
Race_Of_Deceased = ifelse(Race_Of_Deceased == 0, "Not Black", "Black")

df = data.frame(Census_Level_Income, Race_Of_Deceased, fit)

ggplot(df, aes(fit, Census_Level_Income)) +
  geom_point(aes(color = factor(Race_Of_Deceased)))

police = read.csv("police_killings_cleaned.csv")
ggplot(police, aes(x = nat_bucket,fill = raceethnicity)) +geom_bar(position = "d
odge")

ggplot(police, aes(x = nat_bucket,fill = raceethnicity)) +geom_bar()

ggplot(police, aes(x = county_bucket,fill = raceethnicity)) +geom_bar(position =
 "dodge")

ggplot(police, aes(x = county_bucket,fill = raceethnicity)) +geom_bar()

acs = read.csv("acs2015_census_tract_data.csv")
acs = na.omit(acs)
counties = acs

Counties_Income = acs %>%
  group_by(County) %>%
  summarise(TotalCounty = sum(TotalPop),
            County_IncomePerCap = sum(TotalPop*IncomePerCap)/TotalCounty,
            twenty = quantile(County_IncomePerCap,0.2,na.rm=TRUE),
            forty=quantile(County_IncomePerCap,0.4,na.rm=TRUE),
            sixty=quantile(County_IncomePerCap,0.6,na.rm=TRUE),
```

```
            eighty=quantile(County_IncomePerCap,0.8,na.rm=TRUE))

counties = merge(Counties_Income,counties, by= "County")
counties$county_bucket[counties$IncomePerCap < counties$twenty] = 1
counties$county_bucket[counties$IncomePerCap >= counties$twenty & counties$IncomePerCap < counties$forty] = 2
counties$county_bucket[counties$IncomePerCap>=counties$forty & counties$IncomePerCap < counties$sixty] = 3
counties$county_bucket[counties$IncomePerCap>=counties$sixty & counties$IncomePerCap < counties$eighty] = 4
counties$county_bucket[counties$IncomePerCap>=counties$eighty]=5


      ggplot(counties, aes(x = county_bucket)) +geom_bar()
```

MORE FIGURES/EDA

```
library(ggplot2)
library(ggcorrplot)
library(corrplot)


#setwd('Users/stuartgeman/Desktop/data2020/Final Project')
police = read.csv("police_killings_cleaned.csv")
police$X = NULL
#Remove shootings where race of victim is unkown
police = police[ ! police$raceethnicity %in% "Unknown", ]

sum(police$raceethnicity =="Unknown")

acs = read.csv("acs2015_census_tract_data.csv")
names(acs)[names(acs) == 'CensusTract'] <- 'geo_id'
P = police[sapply(police, is.numeric)]
c = acs[sapply(aes, is.numeric)]
P = na.omit(P)
c = na.omit(acs)
total <- merge(police,acs ,by="geo_id")

P$geo_id = NULL
P$latitude= NULL
P$longitude = NULL
P$tract_ce = NULL
P$county_fp = NULL
P$state_fp = NULL
P$county_id = NULL
P$year = NULL
police = read.csv("police_killings_cleaned.csv")


M = cor(P)
corrplot(M,method="circle")
ggcorrplot(M, method = "circle")
total <-merge(aes, police, by = "geo_id")

library(ggplot2)
police$age = police$age +15

police$raceethnicity <- as.character(police$raceethnicity)
police$gender = as.character(police$gender)
```

```
police$cause = as.character(police$cause)
police$armed = as.character(police$armed)
ggplot(police, aes(x = raceethnicity,fill = gender)) +geom_bar()
ggplot(police, aes(x = age, fill = raceethnicity)) + geom_bar()
ggplot(police, aes(x = cause, fill= raceethnicity)) + geom_bar()
ggplot(police, aes(x = cause, fill= armed)) + geom_bar()
ggplot(police, aes(x = raceethnicity, fill = armed)) + geom_bar()
police$unarmed = ifelse(police$armed == "No", "No","Yes")
ggplot(police, aes(x = raceethnicity, fill = unarmed)) + geom_bar()
ggplot(police, aes(x = raceethnicity, fill = county_bucket)) + geom_bar()

ggplot(police, aes(x = raceethnicity, fill = county_bucket)) + geom_bar()

ggplot(police, aes(x = raceethnicity, ))
barplot(prop.table(table(victims)))
gender <- as.character(police$gender)

barplot(prop.table(table(gender)))

armed <- as.character(police$armed)

barplot(prop.table(table(armed)))

cause <- as.character(police$cause)

barplot(prop.table(table(cause)))
barplot(prop.table(table(police$age)))
barplot(prop.table(table(police$county_bucket, main = "Killings Across Income Le
vel")))

Public_other_transit_wealth <-
 ggplot(police, aes(x = share_white, y = h_income))

white_poverty_graph <- ggplot(acs, aes(x = White, y = Unemployment))
white_poverty_graph + geom_line(aes(color = White))

black_h_inome_graph <- ggplot(police, aes(x = share_black, y = h_income))
black_h_inome_graph + geom_line(aes(color = raceethnicity))
```

DATA PROCESSING FOR MATLAB HYPOTHESIS TEST

```
library(ggplot2)
library(ggcorrplot)
library(corrplot)


#setwd('Users/stuartgeman/Desktop/data2020/Final Project')
police = read.csv("police_killings_cleaned.csv")
police$X = NULL
acs = read.csv("acs2015_census_tract_data.csv")
names(acs)[names(acs) == 'CensusTract'] <- 'geo_id'
total <- merge(acs,police,by="geo_id")
total <-na.omit(total)
total = total[ ! total$raceethnicity %in% "Unknown", ]

total = na.omit(total)
rownames(total) <- 1:nrow(total)
stat = subset(total, select=c("geo_id", "raceethnicity", "Hispanic","White", "Bl
ack", "Native","Asian",
```

```
                            "Pacific"))

#Recategorize data as strings/Rename To Match ACS
stat$raceethnicity <- as.character(stat$raceethnicity)
stat$raceethnicity[stat$raceethnicity== "Hispanic/Latino"] <- "Hispanic"
stat$raceethnicity[stat$raceethnicity== "Native American"] <- "Native"
stat$raceethnicity[stat$raceethnicity== "Asian/Pacific Islander"] <-
  "Asian_Pacific"

#stat$raceethnicity <- as.factor(stat$raceethnicity)
stat$Asian_Pacific = stat$Asian + stat$Pacific

Races = subset(stat, select = c("geo_id","Hispanic", "White", "Black", "Native",
  "Asian_Pacific"))
Races = Races[!duplicated(Races$geo_id),]
rownames(Races) <- 1:nrow(Races)

Shootings <- matrix(0, ncol = 6, nrow = 415)
Shootings <- data.frame(Shootings)
names(Shootings)[names(Shootings) == 'X1'] <- 'geo_id'
Shootings$geo_id=Races$geo_id
names(Shootings)[names(Shootings) == 'X2'] <- 'Hispanic'
names(Shootings)[names(Shootings) == 'X3'] <- 'White'
names(Shootings)[names(Shootings) == 'X4'] <- 'Black'
names(Shootings)[names(Shootings) == 'X5'] <- 'Native'
names(Shootings)[names(Shootings) == 'X6'] <- 'Asian_Pacific'


for(i in 1:nrow(stat)) {
  if(stat$raceethnicity[i] == "Hispanic"){
    index <- Shootings$geo_id == stat$geo_id[i]
    Shootings$Hispanic[index] <- Shootings$Hispanic[index] + 1
  }
  if(stat$raceethnicity[i] == "White"){
    index <- Shootings$geo_id == stat$geo_id[i]
    Shootings$White[index] <- Shootings$White[index] + 1
  }
  if(stat$raceethnicity[i] == "Black"){
    index <- Shootings$geo_id == stat$geo_id[i]
    Shootings$Black[index] <- Shootings$Black[index] + 1
  }
  if(stat$raceethnicity[i] == "Native"){
    index <- Shootings$geo_id == stat$geo_id[i]
    Shootings$Native[index] <- Shootings$Native[index] + 1
  }
  if(stat$raceethnicity[i] == "Asian_Pacific"){
    index <- Shootings$geo_id == stat$geo_id[i]
    Shootings$Asian_Pacific[index] <- Shootings$Asian_Pacific[index] + 1
  }

}
Races$geo_id = NULL
#Make Fractions
Races = Races/100

Shootings$geo_id = NULL

# Write CSV's of Shootings and Races dataframes to export to Matlab
write.csv(Shootings, file = "csvs/Tract_RacesOfVictims.csv",row.names=FALSE)
write.csv(Races, file = "csvs/Tract_RacesOfCounties.csv",row.names=FALSE)
```

MORE DATAPROCESSING FOR MATLAB CODE

```
%Start with The NorthEast
%Are using NewAGG R file
C=csvread('csvs/RacesOfNortheast.csv',1);
E=csvread('csvs/NortheEast_RacesOfVictims.csv',1);
%E = E';
NumSamples=1000;
[NumTracts,NTypes]=size(C);

% Clean C, meaning force total in each row to be 1

CSums=sum(C,2);
for col=1:NTypes
    C(:,col)=C(:,col)./CSums;
end

% Get number of attacks in each row, and then normalize rows of E

ESums=sum(E,2);
for col=1:NTypes
    E(:,col)=E(:,col)./ESums;
end

S=sum(sum(abs(E-C))); % The observed value of the statistic

% Make surrogate NumSamples surrogate E's and compute, for each, a
% surrogate S ('SS')

SS=zeros(NumSamples,1);

for samp=1:NumSamples

    ES=zeros(NumTracts,NTypes);  % ES will hold the current surrogate E

    for row=1:NumTracts
        r = mnrnd(ESums(row),C(row,:)); % Multinomial selection of
                    % number of victims, but from the distribution in C

        % load the victim numbers into the surrogate, ES
        for col=1:NTypes
            ES(row,col)=r(col);
        end
    end

    % Normalize the rows of ES (make them probability distributions
    ESSums=sum(ES,2);
    for col=1:NTypes
        ES(:,col)=ES(:,col)./ESSums;
    end

    % Compute the L1 distance between C and the surrogate ES
    SS(samp)=sum(sum(abs(ES-C)));

end

% Display Results

figure(4)
```

```
close(4)
figure(4)
subplot(2,2,1)
hist(SS);
hold on
scatter(S,0,100,'filled','r')
hold off
pvalue=(sum(SS>=S)+1)/(NumSamples+1);
disp(['p-value: ',num2str(pvalue)])
title(['Agg Northeast H-Test p-value <= ',num2str(pvalue)])

%NEW REGION
%Now the South
C=csvread('csvs/RacesOfSouth.csv',1);
E=csvread('csvs/South_RacesOfVictims.csv',1);
%E = E';
NumSamples=1000;
[NumTracts,NTypes]=size(C);

% Clean C, meaning force total in each row to be 1

CSums=sum(C,2);
for col=1:NTypes
    C(:,col)=C(:,col)./CSums;
end

% Get number of attacks in each row, and then normalize rows of E

ESums=sum(E,2);
for col=1:NTypes
    E(:,col)=E(:,col)./ESums;
end

S=sum(sum(abs(E-C))); % The observed value of the statistic

% Make surrogate NumSamples surrogate E's and compute, for each, a
% surrogate S ('SS')

SS=zeros(NumSamples,1);

for samp=1:NumSamples

            ES=zeros(NumTracts,NTypes);  % ES will hold the current surrogate E

            for row=1:NumTracts
                r = mnrnd(ESums(row),C(row,:)); % Multinomial selection of
                            % number of victims, but from the distribution in C

                % load the victim numbers into the surrogate, ES
                for col=1:NTypes
                    ES(row,col)=r(col);
                end
            end

            % Normalize the rows of ES (make them probability distributions
            ESSums=sum(ES,2);
            for col=1:NTypes
                ES(:,col)=ES(:,col)./ESSums;
            end
```

```
117.
118.            % Compute the L1 distance between C and the surrogate ES
119.            SS(samp)=sum(sum(abs(ES-C)));
120.
121.        end
122.
123.        % Display Results
124.
125.        subplot(2,2,2)
126.        hist(SS);
127.        hold on
128.        scatter(S,0,100,'filled','r')
129.        hold off
130.        pvalue=(sum(SS>=S)+1)/(NumSamples+1);
131.        disp(['p-value: ',num2str(pvalue)])
132.        title(['Agg South H-Test p-value <= ',num2str(pvalue)])
133.
134.        %NEW REGION
135.        %Now the Midwest
136.        C=csvread('csvs/RacesOfMidwest.csv',1);
137.        E=csvread('csvs/Midwest_RacesOfVictims.csv',1);
138.        %E = E';
139.        NumSamples=1000;
140.        [NumTracts,NTypes]=size(C);
141.
142.        % Clean C, meaning force total in each row to be 1
143.
144.        CSums=sum(C,2);
145.        for col=1:NTypes
146.            C(:,col)=C(:,col)./CSums;
147.        end
148.
149.        % Get number of attacks in each row, and then normalize rows of E
150.
151.        ESums=sum(E,2);
152.        for col=1:NTypes
153.            E(:,col)=E(:,col)./ESums;
154.        end
155.
156.        S=sum(sum(abs(E-C))); % The observed value of the statistic
157.
158.        % Make surrogate NumSamples surrogate E's and compute, for each, a
159.        % surrogate S ('SS')
160.
161.        SS=zeros(NumSamples,1);
162.
163.        for samp=1:NumSamples
164.
165.            ES=zeros(NumTracts,NTypes);  % ES will hold the current surrogate E
166.
167.            for row=1:NumTracts
168.                r = mnrnd(ESums(row),C(row,:)); % Multinomial selection of
169.                            % number of victims, but from the distribution in C
170.
171.                % load the victim numbers into the surrogate, ES
172.                for col=1:NTypes
173.                    ES(row,col)=r(col);
174.                end
175.            end
```

```
176.
177.            % Normalize the rows of ES (make them probability distributions
178.            ESSums=sum(ES,2);
179.            for col=1:NTypes
180.                ES(:,col)=ES(:,col)./ESSums;
181.            end
182.
183.            % Compute the L1 distance between C and the surrogate ES
184.            SS(samp)=sum(sum(abs(ES-C)));
185.
186.        end
187.
188.        % Display Results
189.
190.        subplot(2,2,3)
191.        hist(SS);
192.        hold on
193.        scatter(S,0,100,'filled','r')
194.        hold off
195.        pvalue=(sum(SS>=S)+1)/(NumSamples+1);
196.        disp(['p-value: ',num2str(pvalue)])
197.        title(['Agg Midwest H-Test p-value <= ',num2str(pvalue)])
198.
199.        %NEW REGION
200.        %Now the West
201.        C=csvread('csvs/RacesOfWest.csv',1);
202.        E=csvread('csvs/West_RacesOfVictims.csv',1);
203.        %E = E';
204.        NumSamples=1000;
205.        [NumTracts,NTypes]=size(C);
206.
207.        % Clean C, meaning force total in each row to be 1
208.
209.        CSums=sum(C,2);
210.        for col=1:NTypes
211.            C(:,col)=C(:,col)./CSums;
212.        end
213.
214.        % Get number of attacks in each row, and then normalize rows of E
215.
216.        ESums=sum(E,2);
217.        for col=1:NTypes
218.            E(:,col)=E(:,col)./ESums;
219.        end
220.
221.        S=sum(sum(abs(E-C))); % The observed value of the statistic
222.
223.        % Make surrogate NumSamples surrogate E's and compute, for each, a
224.        % surrogate S ('SS')
225.
226.        SS=zeros(NumSamples,1);
227.
228.        for samp=1:NumSamples
229.
230.            ES=zeros(NumTracts,NTypes);  % ES will hold the current surrogate E
231.
232.            for row=1:NumTracts
233.                r = mnrnd(ESums(row),C(row,:)); % Multinomial selection of
234.                            % number of victims, but from the distribution in C
```

```
235.
236.                 % load the victim numbers into the surrogate, ES
237.                 for col=1:NTypes
238.                     ES(row,col)=r(col);
239.                 end
240.             end
241.
242.             % Normalize the rows of ES (make them probability distributions
243.             ESSums=sum(ES,2);
244.             for col=1:NTypes
245.                 ES(:,col)=ES(:,col)./ESSums;
246.             end
247.
248.             % Compute the L1 distance between C and the surrogate ES
249.             SS(samp)=sum(sum(abs(ES-C)));
250.
251.         end
252.
253.         % Display Results
254.
255.         subplot(2,2,4)
256.         hist(SS);
257.         hold on
258.         scatter(S,0,100,'filled','r')
259.         hold off
260.         pvalue=(sum(SS>=S)+1)/(NumSamples+1);
261.         disp(['p-value: ',num2str(pvalue)])
262.         title(['Agg West H-Test p-value <= ',num2str(pvalue)])
```

If you wish to see all data processing code for the Matlab tests, check out my Git: https://github.com/JKG114/2020-Final-Project. Otherwise I'll spare you what's left…

Matlab Hypothesis test (I wrote several tests all are similar, this is the first one and is documented the most clearly):

```
1.  % C: an nx5 table of probabilities. Each row is five non-negative numbers
2.  % that add up to 1, representing the proportions of the five ethnicities in
3.  % one census tract. There are as 415 rows(we removed rows containing NA
4.  % values and rows where the race of the victim was unknown).
5.
6.  % E: same as C, except that the proportions come from the population of
7.  % victims. Rows in E correspond to the rows in C, i.e. they come
8.  % from the same census tract.
9.
10. % Define a statistic 'S', which is the L1 distance between E and C
11.
12. % Build a null distribution for S by creating "surrogate" versions of E.
13. % For example, let SE be an nx5 table with rows that correspond to the rows
14. % of C and E, except that the entries are random and come from random
15. % samples from the distributions represented in C. Each row of SE is
16. % determined from 'ESums' selections from the C distribution, where ESums
17. % is the number of victims recorded in the corresponding tract.
18.
19. % Get the data (Tract data and victim data)
20.
21. C=csvread('csvs/Tract_RacesOfCounties.csv',1);
22. E=csvread('csvs/Tract_RacesOfVictims.csv',1);
```

```
23.
24. NumSamples=1000;
25. [NumTracts,NTypes]=size(C);
26.
27. % Clean C, meaning force total in each row to be 1
28.
29. CSums=sum(C,2);
30. for col=1:NTypes
31.     C(:,col)=C(:,col)./CSums;
32. end
33.
34. % Get number of attacks in each row, and then normalize rows of E
35.
36. ESums=sum(E,2);
37. for col=1:NTypes
38.     E(:,col)=E(:,col)./ESums;
39. end
40.
41. S=sum(sum(abs(E-C))); % The observed value of the statistic
42.
43. % Make surrogate NumSamples surrogate E's and compute, for each, a
44. % surrogate S ('SS')
45.
46. SS=zeros(NumSamples,1);
47.
48. for samp=1:NumSamples
49.
50.     ES=zeros(NumTracts,NTypes);  % ES will hold the current surrogate E
51.
52.     for row=1:NumTracts
53.         r = mnrnd(ESums(row),C(row,:)); % Multinomial selection of
54.                       % number of victims, but from the distribution in C
55.
56.         % load the victim numbers into the surrogate, ES
57.         for col=1:NTypes
58.             ES(row,col)=r(col);
59.         end
60.     end
61.
62.     % Normalize the rows of ES (make them probability distributions
63.     ESSums=sum(ES,2);
64.     for col=1:NTypes
65.         ES(:,col)=ES(:,col)./ESSums;
66.     end
67.
68.     % Compute the L1 distance between C and the surrogate ES
69.     SS(samp)=sum(sum(abs(ES-C)));
70.
71. end
72.
73. % Display Results
74.
75. figure(1)
76. close(1)
77. figure(1)
78. hist(SS);
79. hold on
80. scatter(S,0,100,'filled','r')
81. hold off
82. pvalue=(sum(SS>=S)+1)/(NumSamples+1);
83. disp(['p-value: ',num2str(pvalue)])
```

```
84. title(['County Level Hypthesis Test p-value:',num2str(pvalue)])
85.
86.
87.
```