APPENDIX

FEATURE SELECTION AND MULTILEVEL MODELS

1. library(ggplot2)
2. library(ggcorrplot)
3. library(corrplot)
4. library(lme4)
5. library(nlme)
6. library(knitr)
7. library(dplyr)
8. library(MASS)
9. library(bestglm)
10. require(reshape2)
11. require(compiler)
12. require(parallel)
13. require(boot)
14. require(lattice)

17. #setwd('Users/stuartgeman/Desktop/data2020/Final Project')
18. #Get the Police killing data ready (cleaned and remove columns)
19. #police = read.csv("police\_killings\_cleaned.csv")
20. #drop = c("X","name","month","day","year","streetaddress", "city","latitude","longitude",
21. #       "state\_fp","county\_fp","tract\_ce","county\_id","namelsad","lawenforcementagency",
22. #        "pop","state")
23. #police = police[,!(names(police) %in% drop)]
24. #police$age = police$age + 15
25. #police = na.omit(police)
26. #police = police[ ! police$raceethnicity %in% "Unknown", ]
28. police = read.csv("police\_killings\_cleaned.csv")
29. police$X = NULL
30. police$age = police$age + 15
31. police = na.omit(police)
32. police = police[ ! police$raceethnicity %in% "Unknown", ]
34. acs = read.csv("acs2015\_census\_tract\_data.csv")
35. names(acs)[names(acs) == 'CensusTract'] <- 'geo\_id'
37. #We merge the acs dataframe and police dataframe
38. total <- merge(acs,police,by="geo\_id")
39. total <-na.omit(total)
40. total$White = total$White/100
41. total$Black = total$Black/100
42. total$Hispanic = total$Hispanic/100
43. total$Pacific = total$Pacific/100
44. total$Asian = total$Asian/100
45. total$Native - total$Native/100

48. total$raceethnicity <- as.character(total$raceethnicity)
49. total$raceethnicity[total$raceethnicity== "Hispanic/Latino"] <- "Hispanic"
50. total$raceethnicity[total$raceethnicity== "Native American"] <- "Native"
51. total$raceethnicity[total$raceethnicity== "Asian/Pacific Islander"] <- "Asian\_Pacific"
53. # Convert race into binary variable
54. total$raceethnicity[which(total$raceethnicity == "Black")] = "1"
55. total$raceethnicity[which(total$raceethnicity == "White")] = "0"
56. total$raceethnicity[which(total$raceethnicity == "Hispanic")] = "0"
57. total$raceethnicity[which(total$raceethnicity == "Asian\_Pacific")] = "0"
58. total$raceethnicity[which(total$raceethnicity == "Hispanic/Latino")] = "0"
59. total$raceethnicity[which(total$raceethnicity == "Native")] = "0"
61. total$raceethnicity <- as.numeric(as.character(total$raceethnicity))
63. #Get the census data ready (including ready to merge with police)
64. acs = read.csv("acs2015\_census\_tract\_data.csv")
65. acs <- na.omit(acs)
66. acs$Asian\_Pacific = acs$Asian + acs$Pacific
68. names(acs)[names(acs) == 'CensusTract'] <- 'geo\_id'
69. #get rid of columns that are redundant after merge
71. total$Men = total$Men/total$TotalPop
72. total$Women = NULL
73. #state\_pop = aggregate(TotalPop~State,acs,sum)
74. #rownames(total) <- 1:nrow(total)
76. #Get the state level Aggregate Data
77. #Notice we don't do Native or Asian, since these features correspond to very few killings
78. #and will simply increase the "dependency" between our races
79. acs$AsianPacific = acs$Asian + acs$Pacific
80. state\_levels = acs %>%
81. group\_by(State) %>%
82. summarise(TotalState = sum(TotalPop),
83. #Total\_State\_Women = sum(Women)/TotalState,
84. State\_Men = sum(Men)/TotalState,
85. State\_Unemployment =  (sum(TotalPop\*Unemployment\*.01))/TotalState,
86. State\_IncomePerCap = sum(TotalPop\*IncomePerCapErr)/TotalState,
87. State\_Poverty = (sum(TotalPop\*Poverty\*.01))/TotalState,
88. State\_Drive = (sum(TotalPop\*Drive\*.01))/TotalState,
89. State\_Child\_Poverty = (sum(TotalPop\*.01\*ChildPoverty))/TotalState,
90. State\_Hispanic = (sum(TotalPop\*(Hispanic\*.01)))/TotalState,
91. State\_Black = (sum(TotalPop\*(Black\*.01)))/TotalState,
92. State\_White = (sum(TotalPop\*(White\*.01)))/TotalState,
93. State\_Asian\_Pacific = (sum(TotalPop\*(AsianPacific\*.01)))/TotalState,
94. State\_Native = (sum(TotalPop\*(Native\*.01)))/TotalState)


98. #I will now subset the states into regions Northeast, South, West, Midwest (I had a fifth, Mountain,
99. #but there weren't enough observations for it so I distributed the mountain states into the others)
100. NorthEast = c("Connecticut", "Maine", "Massachusetts", "New Hampshire", "Rhode Island", "Vermont",
101. "New Jersey", "New York","Delaware","District of Columbia", "Maryland",
102. "Pennsylvania")
104. South = c("Alabama", "Florida", "Georgia", "Kentucky",
105. "Mississippi", "North Carolina", "South Carolina", "Tennessee","Virginia",
106. "West Virginia","Arkansas", "Louisiana","Oklahoma", "Texas")
108. Midwest = c("Illinois", "Indiana", "Michigan", "Minnesota", "Ohio", "Wisconsin",
109. "Iowa", "Kansas", "Missouri", "Nebraska","North Dakota", "South Dakota")

112. West = c("Arizona", "California", "Hawaii", "Nevada","Alaska",
113. "Idaho", "Oregon", "Washington","Colorado","New Mexico","Utah","Montana","Wyoming")

116. Demographics\_NorthEast =  subset(state\_levels, (state\_levels$State %in% NorthEast))
118. Demographics\_South = subset(state\_levels, (state\_levels$State %in% South))
120. Demographics\_Midwest = subset(state\_levels, (state\_levels$State %in% Midwest))
122. Demographics\_West = subset(state\_levels, (state\_levels$State %in% West))

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)/TotalRegion,
131. Region\_Poverty = (sum(TotalState\*State\_Poverty))/TotalRegion,
132. Region\_Hispanic = (sum(TotalState\*State\_Hispanic))/TotalRegion,
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)

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)/TotalRegion,
144. Region\_Poverty = (sum(TotalState\*State\_Poverty))/TotalRegion,
145. Region\_Hispanic = (sum(TotalState\*State\_Hispanic))/TotalRegion,
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)
151. Demographics\_South = Demographics\_South %>%
152. summarise(TotalRegion = sum(TotalState),
153. Region\_Men = sum(TotalState\*State\_Men)/TotalRegion,
154. Region\_Unemployment =  (sum(TotalState\*State\_Unemployment))/TotalRegion,
155. Region\_IncomePerCap = sum(TotalState\*State\_IncomePerCap)/TotalRegion,
156. Region\_Poverty = (sum(TotalState\*State\_Poverty))/TotalRegion,
157. Region\_Hispanic = (sum(TotalState\*State\_Hispanic))/TotalRegion,
158. Region\_Black = (sum(TotalState\*State\_Black))/TotalRegion,
159. Region\_White = (sum(TotalState\*State\_White))/TotalRegion,
160. Region\_Drive = (sum(TotalState\*State\_Drive))/TotalRegion,
161. Region\_Child\_Poverty = (sum(TotalState\*State\_Child\_Poverty))/TotalRegion)

164. Demographics\_West = Demographics\_West %>%
165. summarise(TotalRegion = sum(TotalState),
166. Region\_Men = sum(TotalState\*State\_Men)/TotalRegion,
167. Region\_Unemployment =  (sum(TotalState\*State\_Unemployment))/TotalRegion,
168. Region\_IncomePerCap = sum(TotalState\*State\_IncomePerCap)/TotalRegion,
169. Region\_Poverty = (sum(TotalState\*State\_Poverty))/TotalRegion,
170. Region\_Hispanic = (sum(TotalState\*State\_Hispanic))/TotalRegion,
171. Region\_Black = (sum(TotalState\*State\_Black))/TotalRegion,
172. Region\_White = (sum(TotalState\*State\_White))/TotalRegion,
173. Region\_Drive = (sum(TotalState\*State\_Drive))/TotalRegion,
174. Region\_Child\_Poverty = (sum(TotalState\*State\_Child\_Poverty))/TotalRegion)
176. drop = c("name","month","day","year","streetaddress", "city","latitude","longitude",
177. "state\_fp","county\_fp","tract\_ce","county\_id","namelsad","lawenforcementagency",
178. "pop","state")
180. #Merge Regional Data On
181. total = total[,!(names(total) %in% drop)]
182. total = merge(total, state\_levels, by= "State")
184. totalNE = subset(total, (total$State %in% NorthEast))
185. totalNE$Region = "NE"
186. Demographics\_NorthEast$Region = "NE"
187. totalNE = merge(totalNE, Demographics\_NorthEast, by= "Region")
189. totalS = subset(total, (total$State %in% South))
190. totalS$Region = "South"
191. Demographics\_South$Region = "South"
192. totalS = merge(totalS, Demographics\_South, by= "Region")
194. totalMid = subset(total, (total$State %in% Midwest))
195. totalMid$Region = "Mid"
196. Demographics\_Midwest$Region = "Mid"
197. totalMid = merge(totalMid, Demographics\_Midwest, by= "Region")
199. totalW = subset(total, (total$State %in% West))
200. 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", "IncomePerCapErr","Native","Asian",
207. "Employed", "PrivateWork","WorkAtHome","share\_black", "share\_hispanic","share\_white",
208. "p\_income","h\_income","county\_income", "TotalState.x", "State\_Men.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\_White.y","State\_Drive.y","State\_Child\_Poverty.y", "cause")
212. total = total[,!(names(total) %in% drop)]

215. #This one is not clear so I leave it out of the above list (same goes for 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
230. step <- stepAIC(full,direction = "both", trace = FALSE)
231. step$anova
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

242. forward <- stepAIC(full,direction = "forward", trace = FALSE)
243. forward$anova
244. #Somewhat dissapointingly gives the same variables back for final
245. #model (in fact, if both can't be applied reverts to backward)!
247. backward <-stepAIC(full, direction = "backward", trace = FALSE)
248. backward$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 variables.
257. #A TRAIN TEST SPLIT
258. ## 75% of the sample size
259. smp\_size <- floor(0.75 \* nrow(total))
261. ## set the seed to make your partition reproducible
262. set.seed(123)
263. train\_ind <- sample(seq\_len(nrow(total)), size = smp\_size)
265. train <- total[train\_ind, ]
266. test <- total[-train\_ind, ]
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)
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)
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()
286. #print(gg)

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$predicted.prob)
298. #png("1d.png", width = 400, height =400, res = 110)
299. #ggplot(test, aes(d = raceethnicity, m = predicted.prob)) +
300. # geom\_abline(slope = 1, intercept = 0) +
301. #labs(x = "1 - Specificity", y = "Sensitivity")
302. #dev.off()
304. #ggplot(test$raceethnicity, p) +
305. # labs(x = "1 - Specificity", y = "Sensitivity")
307. Xy=total
308. Xy$raceethnicity = NULL
310. Xy = cbind(Xy, total$raceethnicity)
311. Xy$gender = NULL
312. myglm <-bestglm(Xy,nvmax = 8)
313. Xy = Xy[,c("TotalPop", "White","Black","Professional","Unemployment",
314. "Service","Construction","State\_IncomePerCap","comp\_income",
315. "age","college","Office", "nat\_bucket","IncomePerCap","State\_Hispanic")]
316. Xy$black\_killed = total$raceethnicity
317. #Xy$nonBlack = 1 - total$raceethnicity
318. myglm\_logit <-bestglm(Xy,family = binomial(), nvmax = 5)

321. #Now that we have looked at a few different glm's to predict whether a person
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)

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$State = STATE
337. train <- total[train\_ind, ]
338. test <- total[-train\_ind, ]
340. #Test with states that were includeded in the training data
341. test = test[test$State %in% unique(train$State),]
343. #STATE INTERCEPT
344. model.state.intercept = glmer(raceethnicity ~ Scale\_TotalPop + age\_scale + Professional+ nat\_bucket+Black + White  +comp\_income
345. + (1|Scale\_State\_IncomePerCap),
346. family = binomial("logit"),REML = FALSE, data=train,
347. glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000)))
349. se1 <- sqrt(diag(vcov(model.state.intercept)))
350. # table of estimates with 95% CI
351. (tab <- cbind(Est = fixef(model.state.intercept),
352. LL = fixef(model.state.intercept) - 1.96 \* se1, UL = fixef(model.state.intercept) + 1.96 \*se1))
354. #print(model.state.intercept, corr = FALSE)
355. predictions.state.intercept <- predict(model.state.intercept, test, type = "response")
356. roc\_model.state.intercept <- roc(test$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)



364. #We have a STATE slope model
365. model.state.slope = glmer(raceethnicity ~ Scale\_TotalPop + age\_scale+ Professional+ nat\_bucket+Black + White  +comp\_income
366. + (Black+age\_scale + nat\_bucket|Scale\_State\_IncomePerCap),
367. family = binomial("logit"),REML = FALSE, data=train,
368. glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000)))


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)
378. predictions.state.slope <- predict(model.state.slope, test, type = "response")
379. roc\_model.state.slope <- roc(test$raceethnicity ~ predictions.state.slope)
380. auc2 = auc(roc\_model.state.slope)
381. plot(roc\_model.state.slope)
382. title(main = "ROC Predictions State Slope Model", line = +3)
384. # STATE SLOPE
385. model.state.slope.intercept = glmer(raceethnicity ~ Scale\_TotalPop + age\_scale+ Professional+ nat\_bucket+Black + White + gender +comp\_income
386. + (1+Black+age\_scale + nat\_bucket|Scale\_State\_IncomePerCap),
387. family = binomial("logit"),REML = FALSE, data=train,
388. glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000)))
390. se3 <- sqrt(diag(vcov(model.state.slope.intercept)))
391. # table of estimates with 95% CI
392. (tab <- cbind(Est = fixef(model.state.slope.intercept),
393. LL = fixef(model.state.slope.intercept) - 1.96 \* se3, UL = fixef(model.state.slope.intercept) + 1.96 \*se3))
394. print(model.state.slope.intercept, corr = FALSE)
395. predictions.state.slope.intercept <- predict(model.state.slope.intercept, test, type = "response")
396. roc\_model.state.slope.intercept <- roc(test$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)
401. anova(model.state.intercept,model.state.slope,model.state.slope.intercept)
403. #REGION INTERCEPT
404. model.region.intercept = glmer(raceethnicity ~ Scale\_TotalPop+age\_scale+ Professional+ nat\_bucket+Black + White + gender +comp\_income
405. + Scale\_State\_IncomePerCap+ State\_Hispanic+ (1|Scale\_TotalRegion),
406. family = binomial("logit"),REML = FALSE, data=train,
407. glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000)))
409. se4 <- sqrt(diag(vcov(model.region.intercept)))
410. # table of estimates with 95% CI
411. (tab <- cbind(Est = fixef(model.region.intercept),
412. LL = fixef(model.region.intercept) - 1.96 \* se4, UL = fixef(model.region.intercept) + 1.96 \*se4))
413. print(model.region.intercept, corr = FALSE)
415. predictions.model.region.intercept <- predict(model.region.intercept, test, type = "response")
416. roc\_model.region.intercept <- roc(test$raceethnicity ~ predictions.model.region.intercept)
417. auc4= auc(roc\_model.region.intercept)
418. plot(roc\_model.region.intercept)
419. title(main = "ROC Predictions Region Intercept Model", line = +3)



424. #REGION VARYING SLOPE
426. model.region.slope = glmer(raceethnicity ~ Scale\_TotalPop+age\_scale+ Professional+ nat\_bucket+Black + White + gender +comp\_income
427. + Scale\_State\_IncomePerCap+ State\_Hispanic+ (Scale\_State\_IncomePerCap+ comp\_income+age\_scale + Black|Scale\_TotalRegion),
428. family = binomial("logit"),REML = FALSE, data=train,
429. glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000)))
431. se5 <- sqrt(diag(vcov(model.region.slope)))
432. # table of estimates with 95% CI
433. (tab <- cbind(Est = fixef(model.region.slope),
434. LL = fixef(model.region.slope) - 1.96 \* se5, UL = fixef(model.region.slope) + 1.96 \*se5))
435. print(model.region.slope, corr = FALSE)
436. prediction.model.region.slope <- predict(model.region.slope, test, type = "response")
437. roc\_model.region.slope <- roc(test$raceethnicity ~ prediction.model.region.slope)
438. auc5= auc(roc\_model.region.slope)
439. plot(roc\_model.region.slope)
440. title(main = "ROC Predictions Region Slope Model", line = +3)
442. #REGION VARYING SLOPE INTERCEPT
444. model.region.slope.intercept = glmer(raceethnicity ~Scale\_TotalPop + age\_scale+ Professional+ nat\_bucket+Black + White + gender +comp\_income
445. + Scale\_State\_IncomePerCap+ State\_Hispanic+ (1 +Scale\_State\_IncomePerCap+ comp\_income+age\_scale + Black|Scale\_TotalRegion),
446. family = binomial("logit"),REML = FALSE, data=train,
447. glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 200000)))
449. se6 <- sqrt(diag(vcov(model.region.slope.intercept)))
450. # table of estimates with 95% CI
451. (tab <- cbind(Est = fixef(model.region.slope.intercept),
452. LL = fixef(model.region.slope.intercept) - 1.96 \* se6, UL = fixef(model.region.slope.intercept) + 1.96 \*se6))
453. print(model.region.slope.intercept, corr = FALSE)
454. predictions.region.region.slope.intercept <- predict(model.region.slope.intercept, test, type = "response")
455. roc\_region.slope.intercept <- roc(test$raceethnicity ~ predictions.region.region.slope.intercept)
456. auc6= auc(roc\_region.slope.intercept)
457. plot(roc\_region.slope.intercept)
458. title(main = "ROC Predictions Region Slope Intercept Model", line = +3)

461. anova(model.state.intercept,model.state.slope,model.state.slope.intercept,
462. model.region.intercept,model.region.slope ,model.region.slope.intercept)

FIGURES/EDA

1. library(arm)
2. library(sjPlot)
3. library(sjmisc)
4. library(lme4)
5. png("4a.png", width = 600, height = 450, res = 100)
6. par(mai = c(0.8, 0.8, 0.1, 0.1))
7. plot(coef(fit.sub.4a)$county[,1], type = "l", lwd = 2, col = "pink",
8. ylab = "County-level Intercepts", xlab = "Counties in the Subset")
9. tmp <- rownames(coef(fit.full.4a)$county) %in%
10. rownames(coef(fit.sub.4a)$county)
11. lines(coef(fit.full.4a)$county[tmp,1], lty = 2, lwd = 2, col = "yellowgreen")
12. legend("bottomleft", c("Subset", "Entire"),
13. 49 lty = 1:2,
14. 50 col = c("pink", "yellowgreen"),
15. 51 lwd = 2)
16. dev.off()

19. coef(model.state.intercept)$State\_IncomePerCap[,1]
21. gg <- ggplot(total, aes(x = State\_IncomePerCap, y = raceethnicity, group = State\_IncomePerCap)) +
22. # geom\_line(aes(y = PooledPredictions), color = "darkgrey") +
23. #geom\_line(aes(y = VaryingInterceptPredictions), color = "blue") +
24. #geom\_line(aes(y = VaryingSlopePredictions), color = "red") +
25. #geom\_line(aes(y = InteractionPredictions), color = "black") +
26. geom\_point(alpha = 0.3, size = 3) +
27. facet\_wrap(~total) +
28. theme\_bw()
30. fit <- fitted(model.state.intercept, total, type="response")
31. State\_IncomePerCap = total$State\_IncomePerCap
32. Race\_Of\_Deceased = total$raceethnicity
33. Race\_Of\_Deceased = ifelse(Race\_Of\_Deceased == 0, "Not Black", "Black")
34. df = data.frame(State\_IncomePerCap, Race\_Of\_Deceased, fit)
36. ggplot(df, aes(fit, State\_IncomePerCap)) +
37. geom\_point(aes(color = Race\_Of\_Deceased))

40. fit <- fitted(model.state.intercept, total, type="response")
41. Census\_Level\_Income = total$IncomePerCap
42. Race\_Of\_Deceased = total$raceethnicity
43. Race\_Of\_Deceased = ifelse(Race\_Of\_Deceased == 0, "Not Black", "Black")
45. df = data.frame(Census\_Level\_Income, Race\_Of\_Deceased, fit)
47. ggplot(df, aes(fit, Census\_Level\_Income)) +
48. geom\_point(aes(color = factor(Race\_Of\_Deceased)))


52. fit <- fitted(model.state.intercept, total, type="response")
53. State\_IncomePerCap = total$State\_IncomePerCap
54. Race\_Of\_Deceased = total$raceethnicity
55. Race\_Of\_Deceased = ifelse(Race\_Of\_Deceased == 0, "Not Black", "Black")
56. df = data.frame(State\_IncomePerCap, Race\_Of\_Deceased, fit)
58. ggplot(df, aes(fit, State\_IncomePerCap)) +
59. geom\_point(aes(color = Race\_Of\_Deceased))

62. fit <- fitted(model.state.intercept, total, type="response")
63. Census\_Level\_Income = total$IncomePerCap
64. Race\_Of\_Deceased = total$raceethnicity
65. Race\_Of\_Deceased = ifelse(Race\_Of\_Deceased == 0, "Not Black", "Black")
67. df = data.frame(Census\_Level\_Income, Race\_Of\_Deceased, fit)
69. ggplot(df, aes(fit, Census\_Level\_Income)) +
70. geom\_point(aes(color = factor(Race\_Of\_Deceased)))
72. police = read.csv("police\_killings\_cleaned.csv")
73. ggplot(police, aes(x = nat\_bucket,fill = raceethnicity)) +geom\_bar(position = "dodge")
75. ggplot(police, aes(x = nat\_bucket,fill = raceethnicity)) +geom\_bar()
77. ggplot(police, aes(x = county\_bucket,fill = raceethnicity)) +geom\_bar(position = "dodge")
79. ggplot(police, aes(x = county\_bucket,fill = raceethnicity)) +geom\_bar()
81. acs = read.csv("acs2015\_census\_tract\_data.csv")
82. acs = na.omit(acs)
83. counties = acs
85. Counties\_Income = acs %>%
86. group\_by(County) %>%
87. summarise(TotalCounty = sum(TotalPop),
88. County\_IncomePerCap = sum(TotalPop\*IncomePerCap)/TotalCounty,
89. twenty = quantile(County\_IncomePerCap,0.2,na.rm=TRUE),
90. forty=quantile(County\_IncomePerCap,0.4,na.rm=TRUE),
91. sixty=quantile(County\_IncomePerCap,0.6,na.rm=TRUE),
92. eighty=quantile(County\_IncomePerCap,0.8,na.rm=TRUE))
94. counties = merge(Counties\_Income,counties, by= "County")
95. counties$county\_bucket[counties$IncomePerCap < counties$twenty] = 1
96. counties$county\_bucket[counties$IncomePerCap >= counties$twenty & counties$IncomePerCap < counties$forty] = 2
97. counties$county\_bucket[counties$IncomePerCap>=counties$forty & counties$IncomePerCap < counties$sixty] = 3
98. counties$county\_bucket[counties$IncomePerCap>=counties$sixty & counties$IncomePerCap < counties$eighty] = 4
99. counties$county\_bucket[counties$IncomePerCap>=counties$eighty]=5

102. ggplot(counties, aes(x = county\_bucket)) +geom\_bar()

MORE FIGURES/EDA

1. library(ggplot2)
2. library(ggcorrplot)
3. library(corrplot)


7. #setwd('Users/stuartgeman/Desktop/data2020/Final Project')
8. police = read.csv("police\_killings\_cleaned.csv")
9. police$X = NULL
10. #Remove shootings where race of victim is unkown
11. police = police[ ! police$raceethnicity %in% "Unknown", ]
13. sum(police$raceethnicity =="Unknown")
15. acs = read.csv("acs2015\_census\_tract\_data.csv")
16. names(acs)[names(acs) == 'CensusTract'] <- 'geo\_id'
17. P = police[sapply(police, is.numeric)]
18. c = acs[sapply(aes, is.numeric)]
19. P = na.omit(P)
20. c = na.omit(acs)
21. total <- merge(police,acs ,by="geo\_id")
23. P$geo\_id = NULL
24. P$latitude= NULL
25. P$longitude = NULL
26. P$tract\_ce = NULL
27. P$county\_fp = NULL
28. P$state\_fp = NULL
29. P$county\_id = NULL
30. P$year = NULL
31. police = read.csv("police\_killings\_cleaned.csv")

34. M = cor(P)
35. corrplot(M,method="circle")
36. ggcorrplot(M, method = "circle")
37. total <-merge(aes, police, by = "geo\_id")
39. library(ggplot2)
40. police$age = police$age +15
42. police$raceethnicity <- as.character(police$raceethnicity)
43. police$gender = as.character(police$gender)
44. police$cause = as.character(police$cause)
45. police$armed = as.character(police$armed)
46. ggplot(police, aes(x = raceethnicity,fill = gender)) +geom\_bar()
47. ggplot(police, aes(x = age, fill = raceethnicity)) + geom\_bar()
48. ggplot(police, aes(x = cause, fill= raceethnicity)) + geom\_bar()
49. ggplot(police, aes(x = cause, fill= armed)) + geom\_bar()
50. ggplot(police, aes(x = raceethnicity, fill = armed)) + geom\_bar()
51. police$unarmed = ifelse(police$armed == "No", "No","Yes")
52. ggplot(police, aes(x = raceethnicity, fill = unarmed)) + geom\_bar()
53. ggplot(police, aes(x = raceethnicity, fill = county\_bucket)) + geom\_bar()
55. ggplot(police, aes(x = raceethnicity, fill = county\_bucket)) + geom\_bar()
57. ggplot(police, aes(x = raceethnicity, ))
58. barplot(prop.table(table(victims)))
59. gender <- as.character(police$gender)
61. barplot(prop.table(table(gender)))
63. armed <- as.character(police$armed)
65. barplot(prop.table(table(armed)))
67. cause <- as.character(police$cause)
69. barplot(prop.table(table(cause)))
70. barplot(prop.table(table(police$age)))
71. barplot(prop.table(table(police$county\_bucket, main = "Killings Across Income Level")))
73. Public\_other\_transit\_wealth <- ggplot(police, aes(x = share\_white, y = h\_income))
75. white\_poverty\_graph <- ggplot(acs, aes(x = White, y = Unemployment))
76. white\_poverty\_graph + geom\_line(aes(color = White))
78. black\_h\_inome\_graph <- ggplot(police, aes(x = share\_black, y = h\_income))
79. black\_h\_inome\_graph + geom\_line(aes(color = raceethnicity))

DATA PROCESSING FOR MATLAB HYPOTHESIS TEST

1. library(ggplot2)
2. library(ggcorrplot)
3. library(corrplot)


7. #setwd('Users/stuartgeman/Desktop/data2020/Final Project')
8. police = read.csv("police\_killings\_cleaned.csv")
9. police$X = NULL
10. acs = read.csv("acs2015\_census\_tract\_data.csv")
11. names(acs)[names(acs) == 'CensusTract'] <- 'geo\_id'
12. total <- merge(acs,police,by="geo\_id")
13. total <-na.omit(total)
14. total = total[ ! total$raceethnicity %in% "Unknown", ]
16. total = na.omit(total)
17. rownames(total) <- 1:nrow(total)
18. stat = subset(total, select=c("geo\_id", "raceethnicity", "Hispanic","White", "Black", "Native","Asian",
19. "Pacific"))
21. #Recategorize data as strings/Rename To Match ACS
22. stat$raceethnicity <- as.character(stat$raceethnicity)
23. stat$raceethnicity[stat$raceethnicity== "Hispanic/Latino"] <- "Hispanic"
24. stat$raceethnicity[stat$raceethnicity== "Native American"] <- "Native"
25. stat$raceethnicity[stat$raceethnicity== "Asian/Pacific Islander"] <- "Asian\_Pacific"
27. #stat$raceethnicity <- as.factor(stat$raceethnicity)
28. stat$Asian\_Pacific = stat$Asian + stat$Pacific
30. Races = subset(stat, select = c("geo\_id","Hispanic", "White", "Black", "Native", "Asian\_Pacific"))
31. Races = Races[!duplicated(Races$geo\_id),]
32. rownames(Races) <- 1:nrow(Races)
34. Shootings <- matrix(0, ncol = 6, nrow = 415)
35. Shootings <- data.frame(Shootings)
36. names(Shootings)[names(Shootings) == 'X1'] <- 'geo\_id'
37. Shootings$geo\_id=Races$geo\_id
38. names(Shootings)[names(Shootings) == 'X2'] <- 'Hispanic'
39. names(Shootings)[names(Shootings) == 'X3'] <- 'White'
40. names(Shootings)[names(Shootings) == 'X4'] <- 'Black'
41. names(Shootings)[names(Shootings) == 'X5'] <- 'Native'
42. names(Shootings)[names(Shootings) == 'X6'] <- 'Asian\_Pacific'

45. **for**(i in 1:nrow(stat)) {
46. **if**(stat$raceethnicity[i] == "Hispanic"){
47. index <- Shootings$geo\_id == stat$geo\_id[i]
48. Shootings$Hispanic[index] <- Shootings$Hispanic[index] + 1
49. }
50. **if**(stat$raceethnicity[i] == "White"){
51. index <- Shootings$geo\_id == stat$geo\_id[i]
52. Shootings$White[index] <- Shootings$White[index] + 1
53. }
54. **if**(stat$raceethnicity[i] == "Black"){
55. index <- Shootings$geo\_id == stat$geo\_id[i]
56. Shootings$Black[index] <- Shootings$Black[index] + 1
57. }
58. **if**(stat$raceethnicity[i] == "Native"){
59. index <- Shootings$geo\_id == stat$geo\_id[i]
60. Shootings$Native[index] <- Shootings$Native[index] + 1
61. }
62. **if**(stat$raceethnicity[i] == "Asian\_Pacific"){
63. index <- Shootings$geo\_id == stat$geo\_id[i]
64. Shootings$Asian\_Pacific[index] <- Shootings$Asian\_Pacific[index] + 1
65. }
67. }
68. Races$geo\_id = NULL
69. #Make Fractions
70. Races = Races/100
72. Shootings$geo\_id = NULL
74. # Write CSV's of Shootings and Races dataframes to export to Matlab
75. write.csv(Shootings, file = "csvs/Tract\_RacesOfVictims.csv",row.names=FALSE)
76. write.csv(Races, file = "csvs/Tract\_RacesOfCounties.csv",row.names=FALSE)

MORE DATAPROCESSING FOR MATLAB CODE

1. %Start with The NorthEast
2. %Are **using** NewAGG R file
3. C=csvread('csvs/RacesOfNortheast.csv',1);
4. E=csvread('csvs/NortheEast\_RacesOfVictims.csv',1);
5. %E = E';
6. NumSamples=1000;
7. [NumTracts,NTypes]=size(C);
9. % Clean C, meaning force total in each row to be 1
11. CSums=sum(C,2);
12. **for** col=1:NTypes
13. C(:,col)=C(:,col)./CSums;
14. end
16. % Get number of attacks in each row, and then normalize rows of E
18. ESums=sum(E,2);
19. **for** col=1:NTypes
20. E(:,col)=E(:,col)./ESums;
21. end
23. S=sum(sum(abs(E-C))); % The observed value of the statistic
25. % Make surrogate NumSamples surrogate E's and compute, **for** each, a
26. % surrogate S ('SS')
28. SS=zeros(NumSamples,1);
30. **for** samp=1:NumSamples
32. ES=zeros(NumTracts,NTypes);  % ES will hold the current surrogate E
34. **for** row=1:NumTracts
35. r = mnrnd(ESums(row),C(row,:)); % Multinomial selection of
36. % number of victims, but from the distribution in C
38. % load the victim numbers into the surrogate, ES
39. **for** col=1:NTypes
40. ES(row,col)=r(col);
41. end
42. end
44. % Normalize the rows of ES (make them probability distributions
45. ESSums=sum(ES,2);
46. **for** col=1:NTypes
47. ES(:,col)=ES(:,col)./ESSums;
48. end
50. % Compute the L1 distance between C and the surrogate ES
51. SS(samp)=sum(sum(abs(ES-C)));
53. end
55. % Display Results
57. figure(4)
58. close(4)
59. figure(4)
60. subplot(2,2,1)
61. hist(SS);
62. hold on
63. scatter(S,0,100,'filled','r')
64. hold off
65. pvalue=(sum(SS>=S)+1)/(NumSamples+1);
66. disp(['p-value: ',num2str(pvalue)])
67. title(['Agg Northeast H-Test p-value <= ',num2str(pvalue)])
69. %NEW REGION
70. %Now the South
71. C=csvread('csvs/RacesOfSouth.csv',1);
72. E=csvread('csvs/South\_RacesOfVictims.csv',1);
73. %E = E';
74. NumSamples=1000;
75. [NumTracts,NTypes]=size(C);
77. % Clean C, meaning force total in each row to be 1
79. CSums=sum(C,2);
80. **for** col=1:NTypes
81. C(:,col)=C(:,col)./CSums;
82. end
84. % Get number of attacks in each row, and then normalize rows of E
86. ESums=sum(E,2);
87. **for** col=1:NTypes
88. E(:,col)=E(:,col)./ESums;
89. end
91. S=sum(sum(abs(E-C))); % The observed value of the statistic
93. % Make surrogate NumSamples surrogate E's and compute, **for** each, a
94. % surrogate S ('SS')
96. SS=zeros(NumSamples,1);
98. **for** samp=1:NumSamples
100. ES=zeros(NumTracts,NTypes);  % ES will hold the current surrogate E
102. **for** row=1:NumTracts
103. r = mnrnd(ESums(row),C(row,:)); % Multinomial selection of
104. % number of victims, but from the distribution in C
106. % load the victim numbers into the surrogate, ES
107. **for** col=1:NTypes
108. ES(row,col)=r(col);
109. end
110. end
112. % Normalize the rows of ES (make them probability distributions
113. ESSums=sum(ES,2);
114. **for** col=1:NTypes
115. ES(:,col)=ES(:,col)./ESSums;
116. end
118. % Compute the L1 distance between C and the surrogate ES
119. SS(samp)=sum(sum(abs(ES-C)));
121. end
123. % Display Results
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)])
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);
142. % Clean C, meaning force total in each row to be 1
144. CSums=sum(C,2);
145. **for** col=1:NTypes
146. C(:,col)=C(:,col)./CSums;
147. end
149. % Get number of attacks in each row, and then normalize rows of E
151. ESums=sum(E,2);
152. **for** col=1:NTypes
153. E(:,col)=E(:,col)./ESums;
154. end
156. S=sum(sum(abs(E-C))); % The observed value of the statistic
158. % Make surrogate NumSamples surrogate E's and compute, **for** each, a
159. % surrogate S ('SS')
161. SS=zeros(NumSamples,1);
163. **for** samp=1:NumSamples
165. ES=zeros(NumTracts,NTypes);  % ES will hold the current surrogate E
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
171. % load the victim numbers into the surrogate, ES
172. **for** col=1:NTypes
173. ES(row,col)=r(col);
174. end
175. end
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
183. % Compute the L1 distance between C and the surrogate ES
184. SS(samp)=sum(sum(abs(ES-C)));
186. end
188. % Display Results
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)])
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);
207. % Clean C, meaning force total in each row to be 1
209. CSums=sum(C,2);
210. **for** col=1:NTypes
211. C(:,col)=C(:,col)./CSums;
212. end
214. % Get number of attacks in each row, and then normalize rows of E
216. ESums=sum(E,2);
217. **for** col=1:NTypes
218. E(:,col)=E(:,col)./ESums;
219. end
221. S=sum(sum(abs(E-C))); % The observed value of the statistic
223. % Make surrogate NumSamples surrogate E's and compute, **for** each, a
224. % surrogate S ('SS')
226. SS=zeros(NumSamples,1);
228. **for** samp=1:NumSamples
230. ES=zeros(NumTracts,NTypes);  % ES will hold the current surrogate E
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
236. % load the victim numbers into the surrogate, ES
237. **for** col=1:NTypes
238. ES(row,col)=r(col);
239. end
240. end
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
248. % Compute the L1 distance between C and the surrogate ES
249. SS(samp)=sum(sum(abs(ES-C)));
251. end
253. % Display Results
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).
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.
10. % Define a statistic 'S', which is the L1 distance between E and C
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.
19. % Get the data (Tract data and victim data)
21. C=csvread('csvs/Tract\_RacesOfCounties.csv',1);
22. E=csvread('csvs/Tract\_RacesOfVictims.csv',1);
24. NumSamples=1000;
25. [NumTracts,NTypes]=size(C);
27. % Clean C, meaning force total in each row to be 1
29. CSums=sum(C,2);
30. **for** col=1:NTypes
31. C(:,col)=C(:,col)./CSums;
32. end
34. % Get number of attacks in each row, and then normalize rows of E
36. ESums=sum(E,2);
37. **for** col=1:NTypes
38. E(:,col)=E(:,col)./ESums;
39. end
41. S=sum(sum(abs(E-C))); % The observed value of the statistic
43. % Make surrogate NumSamples surrogate E's and compute, **for** each, a
44. % surrogate S ('SS')
46. SS=zeros(NumSamples,1);
48. **for** samp=1:NumSamples
50. ES=zeros(NumTracts,NTypes);  % ES will hold the current surrogate E
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
56. % load the victim numbers into the surrogate, ES
57. **for** col=1:NTypes
58. ES(row,col)=r(col);
59. end
60. end
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
68. % Compute the L1 distance between C and the surrogate ES
69. SS(samp)=sum(sum(abs(ES-C)));
71. end
73. % Display Results
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)])