Final Write-Up

Azka Javaid & Caleb Ki

December 19, 2016

Introduction

In 2015, 35092 people died in motor vehicle accidents according to the National Highway Traffic Safety Association (NHTSA). This is an increase of 2348 fatalities from 2014. Part of the reason that there was an increase in the number of fatal traffic accidents overall is that the number of fatal traffic accidents involving alcohol impaired drivers increased by 322. The CDC reports that around 31% of traffic fatalities include a drunk driver and that alcohol impaired vehicle accidents cost around $44 billion.

Although the dangers of drunk driving are well documented at an individual level, the risk factors associated with drunk driving are generally not well known at an aggregate level. There is relatively little literature exploring why there are large discrepancies in rate of drunk driving between different U.S. counties. For example, in 2015, Fremont County, Wyoming, had 27 fatal traffic accidents involving alcohol per 100,000 people whereas the same statistic for Morris County, New Jersey is only 0.2. This difference in drunk driving incidence between the two counties is vast though there is little understanding of how and why these disparities are accentuated based on socioeconomic differences or additional factors.

Identifying risk factors associated with drunk driving could help develop intervention methods. More specifically, educational programs can be developed to spread awareness about what preventable factors increase the likelihood of drunk driving like time of day and weekend/weekday status. Moreover pinpointing county level disparities in drunk driving related accidents could lead to geographic level government policy implementations like increased budget allocation to spread awareness about safe driving practices in counties with high rates of drunk driving and fatalities.

In this project, we plan to determine what factors at individual and county level predict whether or not the driver involved in a fatal traffic accident was alcohol impaired. Individual level predictors include driver's sex, age, past speeding related suspensions and convictions. County level predictors of interest include population and income to poverty ratio. Lastly, we also analyze accident level attribute like time of day and week status (weekday/weekend) in potentially explaning drunk driving incidence related to fatal accidents.

Data

We used data from two different sources to identify potential risk factors associated with drunk driving related fatal accidents. Our primary source of data was from the Fatality Accident Report Service (FARS) managed by the National Highway Traffic Safety Administration (NHTSA). FARS provides information concerning every vehicle and person involved in a fatal traffic accident. The data is provided at an accident, vehicle, and person level. Each accident is given a unique identifier, and each row of the vehicle and person datasets includes the unique ID indicating which accident that the vehicle or person was a part of. This allowed us to seamlessly join the data together. The manual provided in this link contains information about each variable within the datasets.

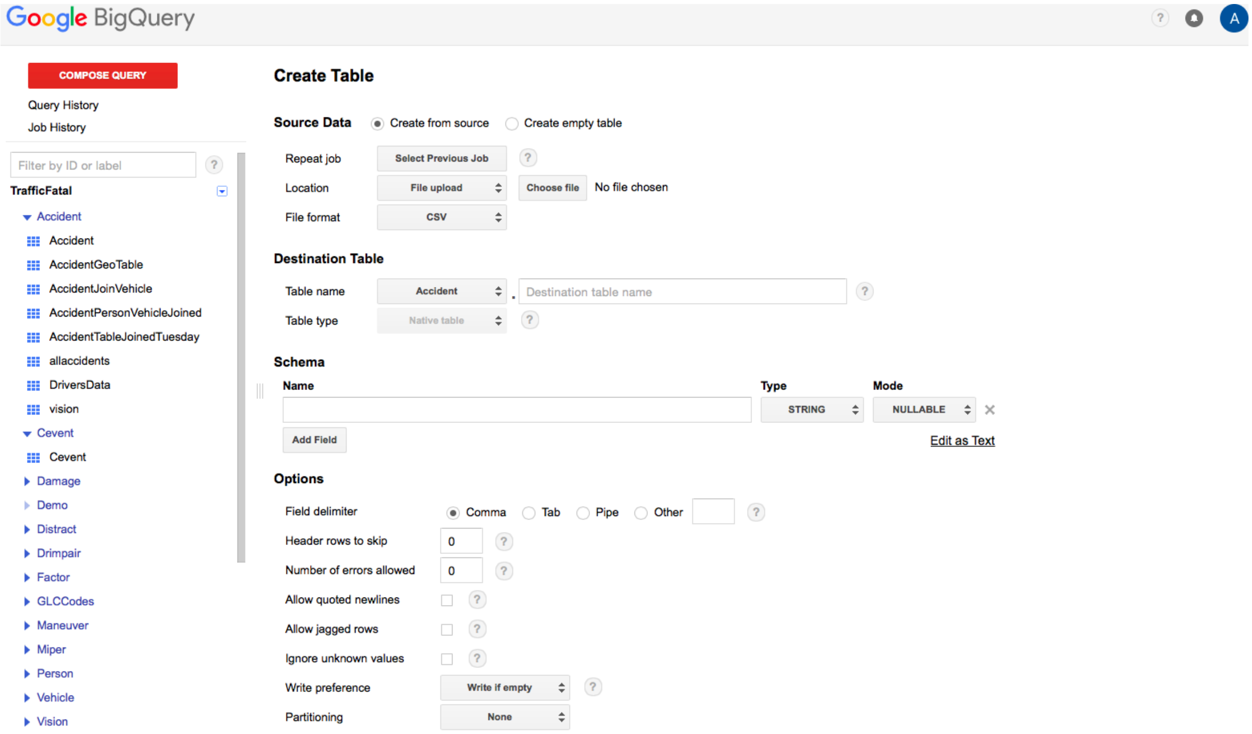
The second and supplementary source of data came from the American Community Survey (ACS) conducted by the U.S. Census Bureau. The ACS provides social, economic, and demographic data at the county, state, and national level. The FARS data contains geographic information like latitude, longitude, county, and state. By calculating FIPS codes, we were able to match the county level data we were looking for to each accident, vehicle and person. Information about the ACS can be found here.

Data Wrangling

BigQuery

We used Google's BigQuery as a hosting platform for our datasets. BigQuery is an analytics warehouse with the ability to process data on the petabyte scale. It provides a serverless and infrastructure-less environment through its cloud data storage, thereby further reducing the need for a database administrator. Its features include the ability to ingest data from different sources including Google Cloud Storage, Cloud Datastore, and livestream. Data can be read and written via Cloud Dataflow, Spark, and Hadoop and exported out in the Cloud. A key feature of BigQuery is the ability to collaborate and share queries as well as data by adding members to a project. Since we used accident, vehicle and person level data, BigQuery provided a cohesive and structured environment for managing all three. Additionally the user-friendly interface was conducive to basic exploratory analysis with SQL as well as for performing variety of joins.

Most basic and preliminary use of BigQuery entails navigating between two environments: Google BigQuery and Google Cloud Platform. Data can be uploaded in BigQuery by first creating a project and enabling both billing and the BigQuery API. Once a project is created, datasets can be added from the 'create a new dataset drop down option' on the highlighted project (available on the left side of the inteface). After specifying a dataset, it is populated by tables of interest. The specification of the table entails defining a schema (structure or data skeleton), which involves defining the variable names as well the data types for each variable as shown by Figure 1.



Create Table

Source Data ● Create from source ○ Create empty table

Repeat job: Select Previous Job

Location: File upload — Choose file — No file chosen

File format: CSV

Destination Table

Table name: Accident . Destination table name

Table type: Native table

Schema

Name	Type	Mode
	STRING	NULLABLE

Options

Field delimiter: ● Comma ○ Tab ○ Pipe ○ Other

Header rows to skip: 0

Number of errors allowed: 0

Allow quoted newlines □

Allow jagged rows □

Ignore unknown values □

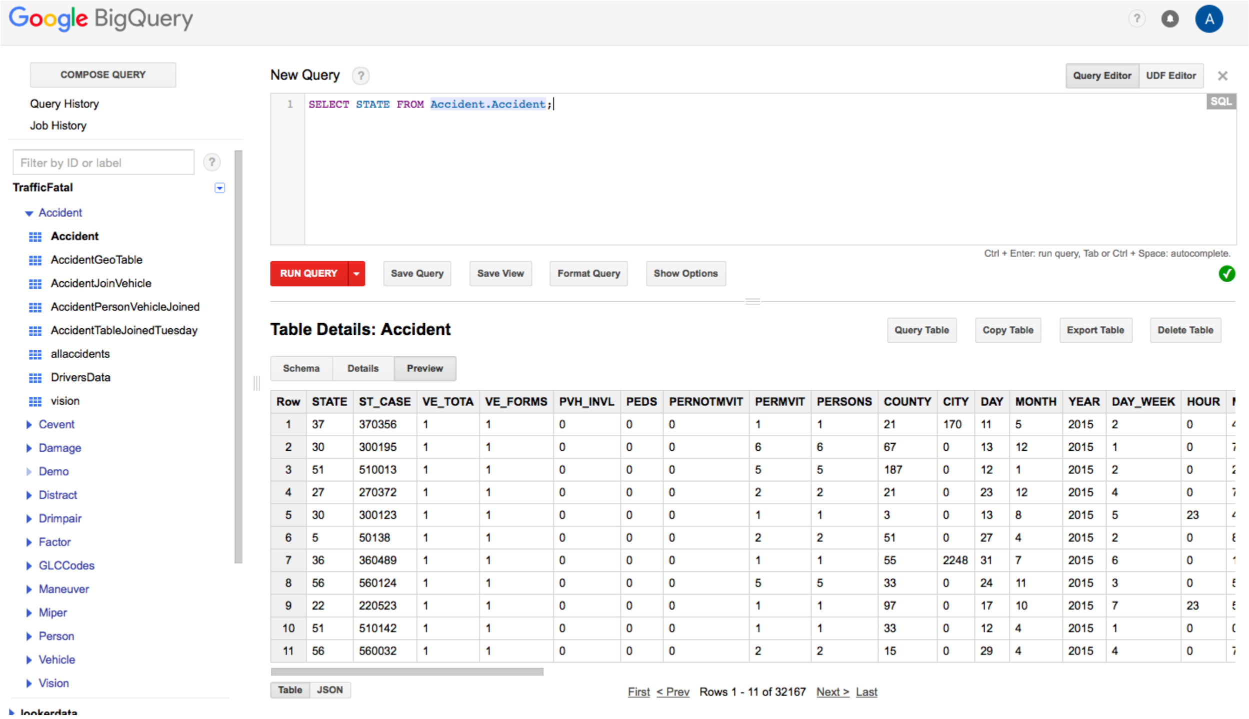
Write preference: Write if empty

Partitioning: None

Figure 1: Figure 1

Once the data is exported in a table, it can be previewed and queried through the 'compose query tab' on the left (see Figure 2).

After the data is queried, the resulting table can be exported out to the Cloud by first creating a bucket from the Cloud console. Buckets can be created by selecting the Storage option from the main Cloud Platform console page. After a bucket is created, the queried data can be exported to that bucket with the name of the bucket and filename specified in format: gs://tablejoin/accident.csv where tablejoin is the name of the folder and accident is the name of the csv formatted table (see Figure 3).



```
SELECT STATE FROM Accident.Accident;
```

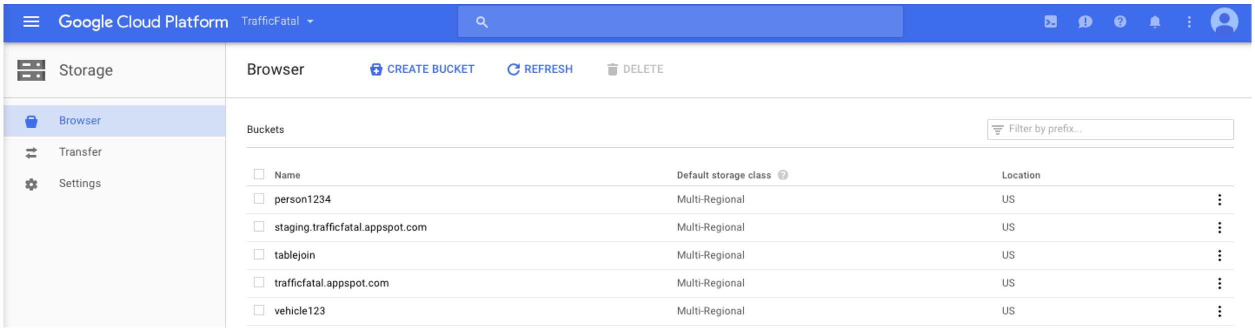
Ctrl + Enter: run query, Tab or Ctrl + Space: autocomplete.

Table Details: Accident

Row	STATE	ST_CASE	VE_TOTA	VE_FORMS	PVH_INVL	PEDS	PERNOTMVIT	PERMVIT	PERSONS	COUNTY	CITY	DAY	MONTH	YEAR	DAY_WEEK	HOUR
1	37	370356	1	1	0	0	0	1	1	21	170	11	5	2015	2	0
2	30	300195	1	1	0	0	0	6	6	67	0	13	12	2015	1	0
3	51	510013	1	1	0	0	0	5	5	187	0	12	1	2015	2	0
4	27	270372	1	1	0	0	0	2	2	21	0	23	12	2015	4	0
5	30	300123	1	1	0	0	0	1	1	3	0	13	8	2015	5	23
6	5	50138	1	1	0	0	0	2	2	51	0	27	4	2015	2	0
7	36	360489	1	1	0	0	0	1	1	55	2248	31	7	2015	6	0
8	56	560124	1	1	0	0	0	5	5	33	0	24	11	2015	3	0
9	22	220523	1	1	0	0	0	1	1	97	0	17	10	2015	7	23
10	51	510142	1	1	0	0	0	1	1	33	0	12	4	2015	1	0
11	56	560032	1	1	0	0	0	2	2	15	0	29	4	2015	4	0

First < Prev Rows 1 - 11 of 32167 Next > Last

Figure 2: Figure 2



Browser

Buckets

Name	Default storage class	Location
person1234	Multi-Regional	US
staging.trafficfatal.appspot.com	Multi-Regional	US
tablejoin	Multi-Regional	US
trafficfatal.appspot.com	Multi-Regional	US
vehicle123	Multi-Regional	US

Figure 3: Figure 3

BigQuery has additional features worth highlighting like the publicly available datasets, which include the National Oceanic and Atmospheric Administration (NOAA) global data obtained from the USAF Climatology Center, US Disease Surveillance data from the CDC, NYC Taxi and Limousine Commission (TLC) Trip Data and GDELT Internet Archive Book Data.

FARS/ACS datasets

The FARS dataset was cleaned to remove any observations with unreported sex, drug use and death scence status. A FIPS code column was created and used as the basis for joining the FARS and the ACS datasets so that each driver is matched with the county and state aggregate population and income to poverty ratio measures. We also created weekday and day status predictors. The data given from FARS already had information about the day of the week the accident occurred as well as the time. When dayweek, the variable for day of the week equaled 1 it meant Sunday and when it equaled 7 it meant Saturday so the weekend variable was coded simply to be a binary indicator of weekend using the day of week variable. We coded night for if the accident occurred between the hours of 6pm and 6am and day otherwise. Final predictors of interest as a measure of driver's drunk incidence included the driver's sex (Sex), age (Age), history of previous DWI convictions, previous speeding convictions and suspensions and revocations (PrevDWIConvictions, PrevSpeeding, PrevSuspensions), driver's police reported drug use and death scene status (ReportedDrugs, DeathSceneStatus), vehicle level attributes like speed before crash (VehicleSpeed), accident level atributes like number of fatalities, weekday status and say status (NumFatalities, WeekdayStatus, DayStatus) and county level variables like total population and income to poverty ratio in the past 12 months of the driver's state and county (TotalPopulation, IncomeToPovRatio).

Data Visualization/Exploration

In order to visualize and explore the data, the shiny, leaflet and choroplethr packages were used. The findings from the shiny app will be detailed below, but the app itself can be found at this link. The shiny app was made up of three components: a marker map made using leaflet and two choropleth maps with different forms of interactivity.

Marker Map

The first of three components was the marker map. For every fatal accident in the year 2015, there was a circle representing the accident at the location of the accident on the map (using latitude and longitude). The user could choose whether to color code the circles by whether a drunk driver was involved or not, whether it was a weekend or weekday, or whether it was night (between 6pm and 6am) or day. The user could also choose to scale the dot size by the number of fatilities or the number of people involved in the accident.

Displayed below is the marker map where the color is based on whether a drunk driver or not and the size is based on the number of fatilities in the accident.

And displayed below here is the marker map where the color is based on whether a drunk driver or not and the size is based on the number of people in the accident.

In the first map, the yellow dots are easy to make out whereas in the second map, the yellow dots are muted. This indicates that a drunk driver was not present in the accidents with largest number of people involved. On the other hand, a drunk driver was present in some of the accidents with the largest number of people who died. This indicates that there maybe a direct relationship between whether a drunk driver was involved in an accidents and the number of fatalities or an inverse relationship between whether a drunk driver was involved and the number of people who were involved. This is something we explore later.

Another feature of the marker map is that if a circle is clicked, a pop-up display will show up. An example is given below.

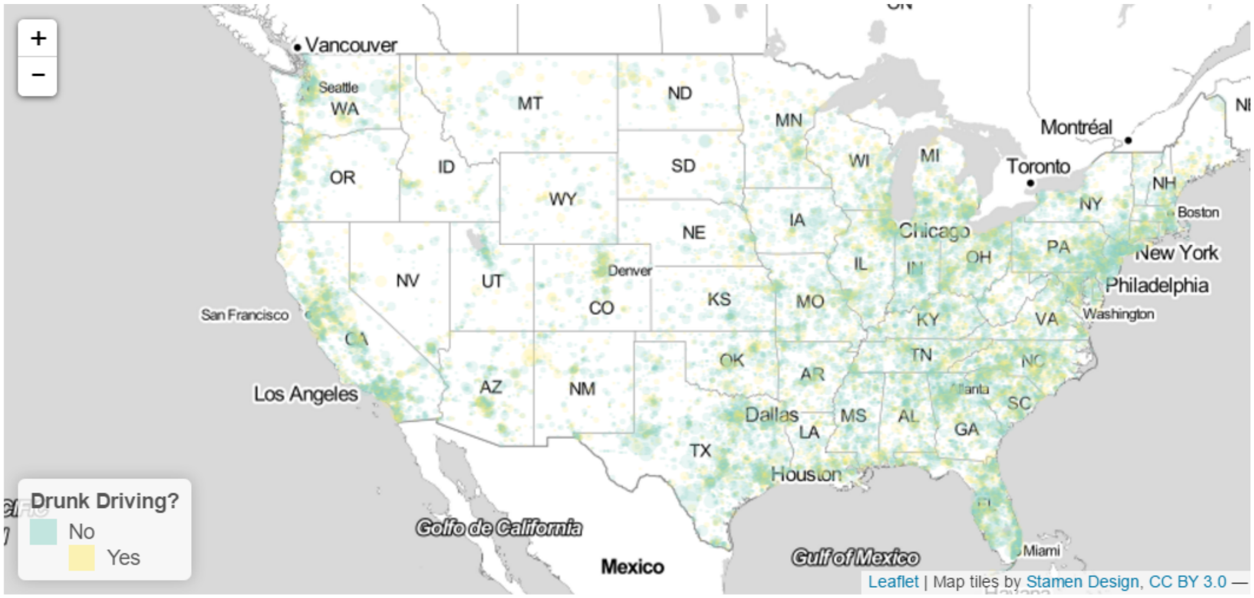


Figure 4: Figure 4

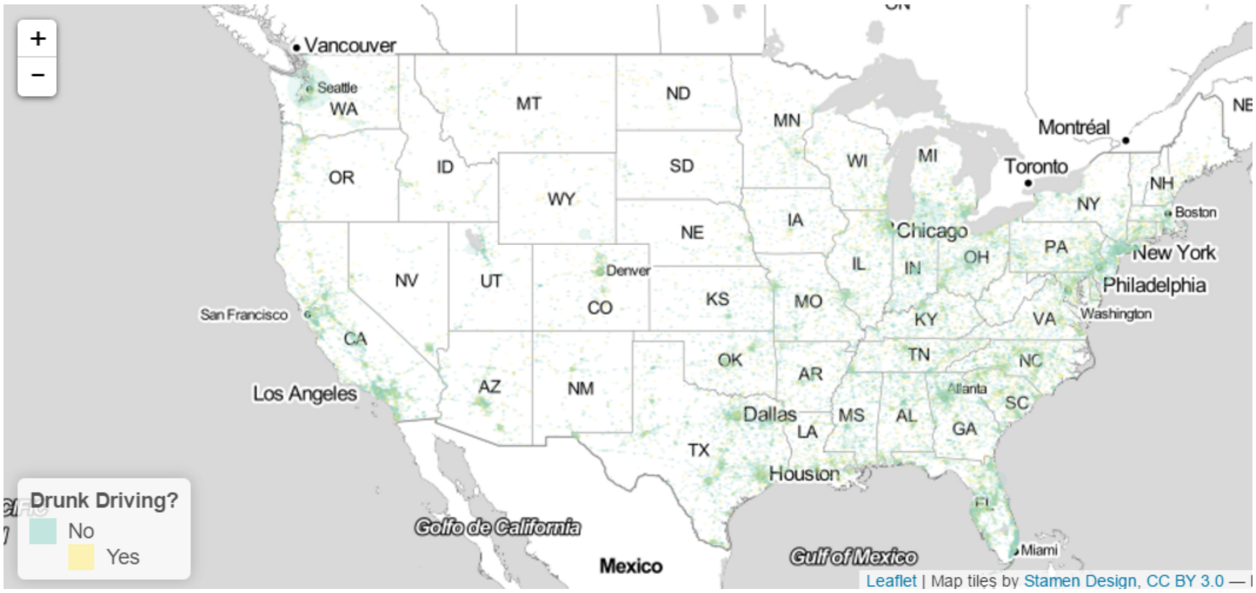
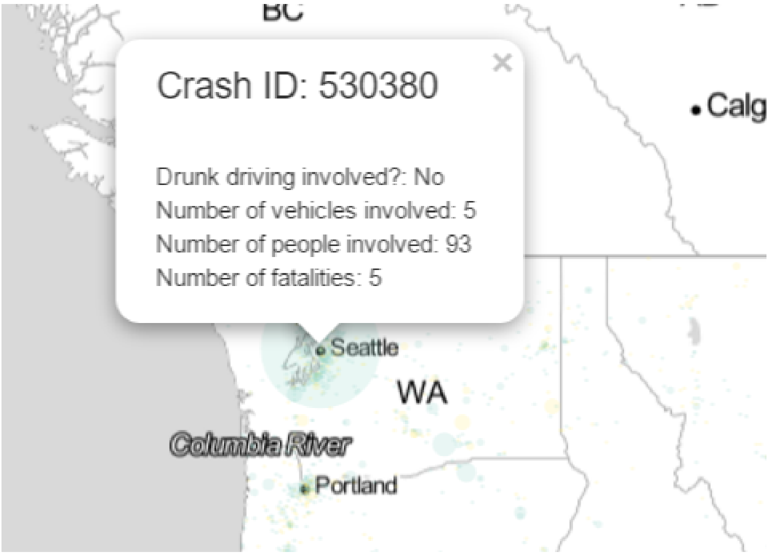


Figure 5: Figure 5



Crash ID: 530380

Drunk driving involved?: No
Number of vehicles involved: 5
Number of people involved: 93
Number of fatalities: 5

Figure 6: Figure 6

The pop-up displays relevant information regarding the accident including whether a drunk driver was involved, the number of vehicles, the number of people, and the number of fatalities. The most important feature of the pop-up is that it displays the unique ID given to each accident. If there is a particularly interesting accident displayed on the marker map and the user would like to look into it, the ID provided the pop-up is essential. The user could take the ID, and match the accident from the map to the corresponding row in the dataset.

Choropleth Map 1

The second of the three components was a choropleth map. The primary use of this component is to see how fatal traffic accidents involving drunk drivers varies from county to county and to explore possible relationships between drunk driving and other predictors. The user could select they wanted a state or a county level choropleth map. The user could also select whether they wanted the choropleth to be based on raw count of accidents or number of accidents per 100,000. Other features include selecting whether the data contained all accidents or only those involving drunk driver; whether to include accidents occurring during the weekend, the weekday, or both; and whether to include accidents occurring during the night, the day, or both.

Displayed below is a county level choropleth map of raw count of fatal accidents involving drunk drivers. The darker the green the more fatal accidents involving a drunk driver in that county.

Displayed below is a county level choropleth map of the number of fatal accidents involving drunk drivers per 100,000 inhabitants. This was to normalize for population as the larger counties are expected to have larger raw counts.

The two visuals above confirm that fatal accidents involving drunk drivers is not uniformly distributed both in terms of raw count and rate. From the differences in the first visual to the second, it seems that those counties with large populations (and therefore more raw accidents) seem to have more stable rates of fatal accidents involving drunk drivers. These displays also serve to point out the counties and regions with abnormally high rates of fatal traffic accidents involving drunk drivers. In particular, the Mountain West region seems to have the highest share of counties that are very green (and so have the largest rates of accidents involving drunk drivers).

This next display is a state level county level choropleth map of the number of fatal accidents involving drunk drivers per 100,000 inhabitants.

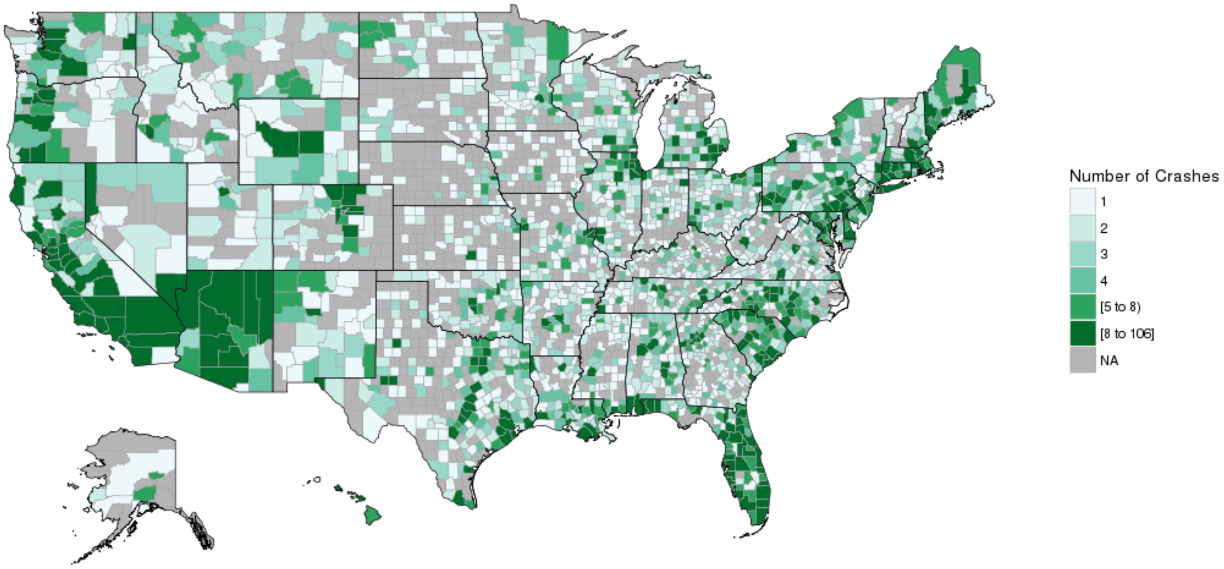


Figure 7: Figure 7

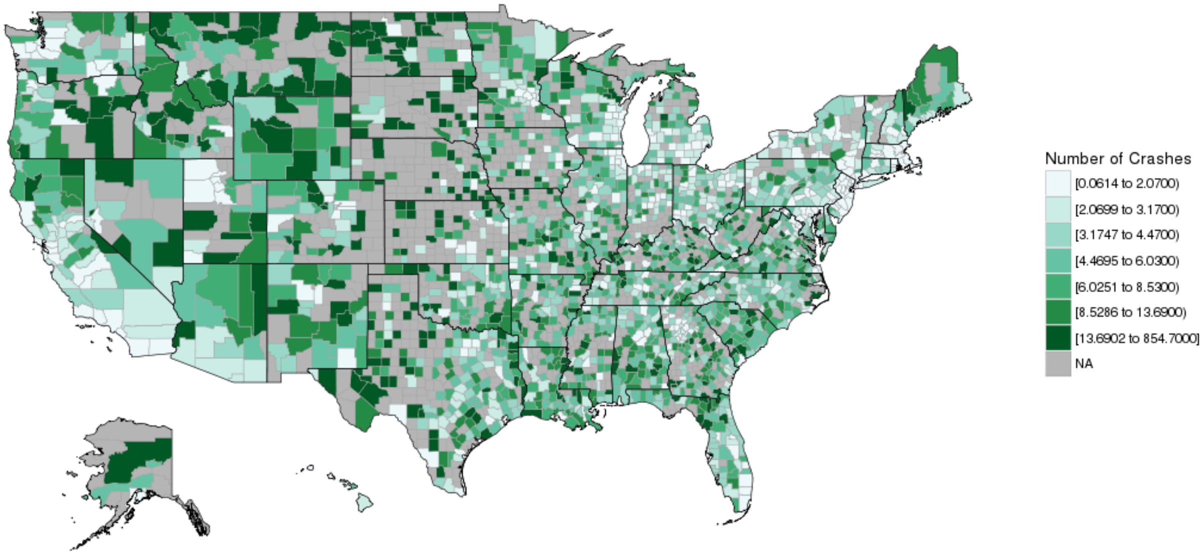


Figure 8: Figure 8

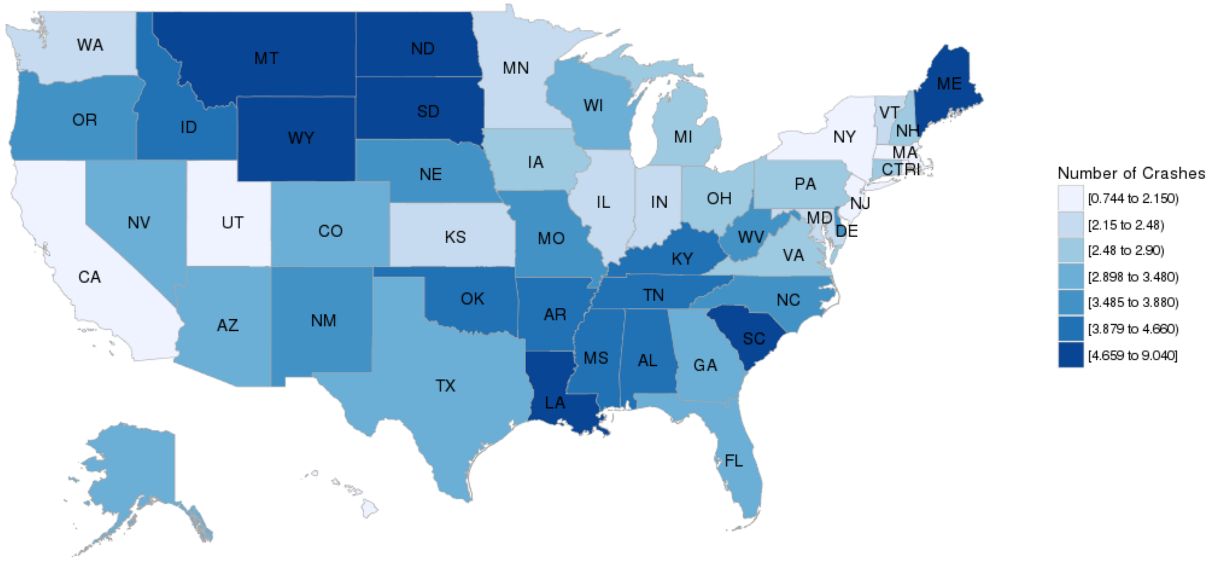
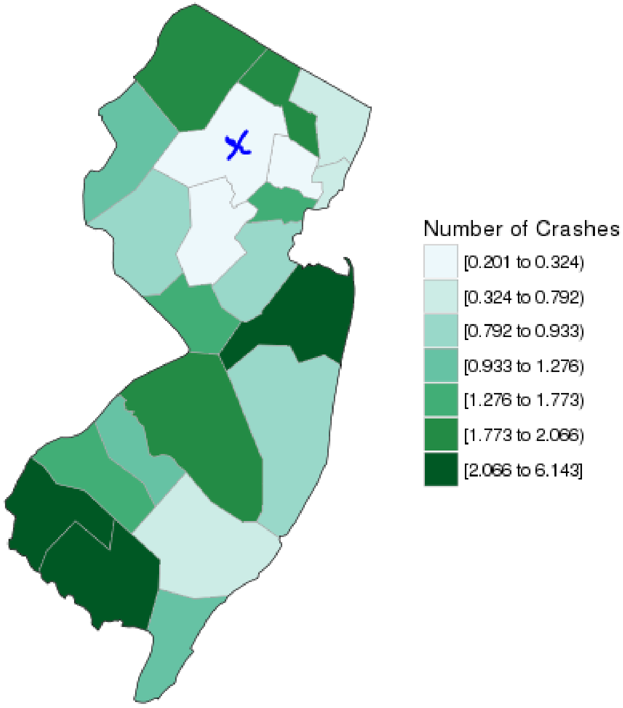
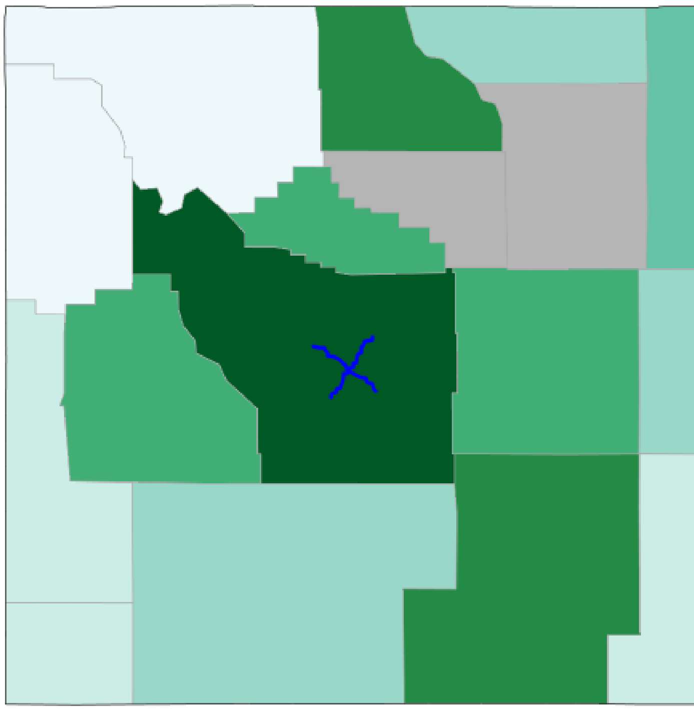


Figure 9: Figure 9

The state level display confirms that the Mountain West region has high rates of accidents involving drunk drivers. In addition, the states in the South also seem to have relatively high rates of accidents involving drunk drivers. The Mountain Northwest and the South regions happens to be two of the poorer and less populated regions of America. This indicates that perhaps population and some measure of income at the county or state level could be valuable predictors for whether a driving involved in an accident was drunk or not.

A nice feature of the second component of the shiny app is that it allows the users to zoom in on states they wish to explore further. As an example, displayed below are choropleth plots zoomed in on the states of New Jersey and Wyoming. Morris County and Fremont County (the counties discussed in the motivation section) are marked by a blue X.





The visual confirms that indeed the rates of fatal traffic accidents involving drunk drivers is much higher in Fremont County than in Morris County.

The last component of the shiny app will be discussed after our analysis is discussed.

Results/Analysis

The dataset was then split in test and training sets with a 70/30 split (70% data in training and 30% in test set) where the training set had 27,678 observations while the test set had 11,862 observations. Logistic regression was used first to assess whether drunk driving incidence can be predicted by the above mentioned predictors followed by random forest. Both models were built on the training data and then assessed on the test data via accuracy and ROC curves.

```r
set.seed(495)
load("FinalData.Rda")
n <- nrow(FinalData)
shuffled <- FinalData[sample(n),]
train <- shuffled[1:round(0.7 * n),]
test <- shuffled[(round(0.7 * n) + 1):n,]
```

Logistic Regression

Predictors associated with increased drunk driving incidence Predictors accounting for the driver's previous suspensions and revocations, previous DWI convictions, number of fatalities in the accident, the day of the crash being a weekend and time being night and total population all increase the odds of a driver being drunk.

Exponential of the coefficient estimates was assessed in the column labeled ExpEstimate (since logistic model is used, exponentiating the coefficients allows for a more intuitive interpretation). The coefficient for PrevSuspensions indicates that holding other variables constant, the odds of a driver involved in a fatal accident being drunk increase by 1.09 given an increase in the previous suspension record by one. Coefficient for PreDWIConvictions indicates that an increase in the previous DWI convictions by one produces an increase in the odds of the driver being drunk by about 1.03 holding other variables fixed.

In regards to accident and county level attributes, increase in number of fatalities by one inceases the odds of a driver involved in fatal accident being drunk by about 1.03. If the day of crash is a weekend, then there is about a 1.88 odds of a drunk driver incidence in comparison to if the crash occurred on a weekday. If the time occurred during night, then there is about a 3.96 odds of the driver being drunk than if the crash occurred during the day. Lastly the odds of a drunk driver increase by about 1.00 given a one unit increase in the total population holding other variables fixed.

Predictors associated with decreased drunk driving incidence Predictors like the driver's sex, age, history of previous speeding suspensions, number of fatalities and income to poverty ratio all decreased the odds of a driver being drunk.

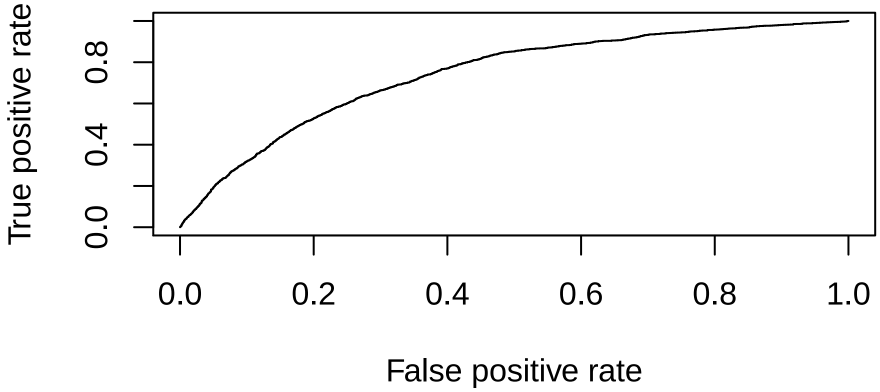
The coefficient for Sex indicates that holding other variables constant, if the driver is female, then there is about a 0.56 odds of a driver being drunk than if the driver was a male. Age indicates that the odds of a driver being drunk are about 0.99 higher given a one year increase in age. Variable for PrevSpeeding indicates that holding other variables constant, the odds of a driver being drunk are 0.89 higher given an increase in previous speeding suspensions of one holding other variables constant. Lastly an increase in income to poverty ratio of one is associated with a 0.99 increased chance of odds of the driver being drunk.

```
logmodnew <- glm(formula = DriverDrinking ~ Sex + Age + PrevSuspensions +
                PrevDWIConvictions + PrevSpeeding +
                NumFatalities + WeekdayStatus + DayStatus +
                IncomeToPovRatio + TotalPopulation,
              family=binomial(link='logit'), data = train)
#save(logmodnew, file = "logisticmodFinal.Rda")
logmod <- logmodnew
logTable <- tidy(logmod)
logTable <- logTable %>% mutate(ExpEstimate = exp(estimate))
logTable <- logTable %>% dplyr::select(term, estimate, ExpEstimate, std.error, p.value)
logTable <- rename(logTable, Estimate = estimate)
logTable <- rename(logTable, Term = term)
logTable <- rename(logTable, StdError = std.error)
logTable <- rename(logTable, PValue = p.value)
logTable  #added exponentiated estimate column in the model summary
```

```
##                    Term      Estimate ExpEstimate     StdError
## 1           (Intercept) -1.899843e+00   0.1495920 7.335652e-02
## 2             SexFemale -5.666558e-01   0.5674199 4.229024e-02
## 3                   Age -1.367535e-02   0.9864177 1.034101e-03
## 4       PrevSuspensions  7.456089e-02   1.0774109 8.140213e-03
## 5    PrevDWIConvictions  4.418873e-02   1.0451796 2.156707e-02
## 6          PrevSpeeding -1.197809e-01   0.8871148 2.138956e-02
## 7         NumFatalities  3.500594e-02   1.0356259 3.884083e-02
## 8  WeekdayStatusWeekend  6.650436e-01   1.9445753 3.360326e-02
## 9        DayStatusNight  1.416143e+00   4.1211947 3.714578e-02
## 10     IncomeToPovRatio -2.000525e-06   0.9999980 2.475694e-06
## 11      TotalPopulation  1.889423e-06   1.0000019 2.439927e-06
##           PValue
## 1  6.876144e-148
## 2   6.111013e-41
## 3   6.345507e-40
## 4   5.210196e-20
## 5   4.047203e-02
## 6   2.143889e-08
## 7   3.674467e-01
## 8   3.556150e-87
## 9   0.000000e+00
## 10  4.190525e-01
## 11  4.387080e-01
```

Logistic Regression Assessment Logistic regression was assessed with model accuracy. Initial model accuracy was about 0.74 and then cross validation with k = 100 (data is iteratively split in training and test sets 100 times and accuracy is computed and averaged) was used. The accuracy from cross validation was about about 0.82.

```
prob <- predict(logmod, newdata=test, type="response")
pred <- prediction(prob, test$DriverDrinking)
perf <- performance(pred, measure = "tpr", x.measure = "fpr")
plot(perf)
```



```
#Accuracy assessment
auc <- performance(pred, measure = "auc")
auc <- auc@y.values[[1]]
auc #75% accurate
```

```
## [1] 0.7425075
```

```
#Cross-validation for model assessment
dat <- FinalData
k <- 100
acc <- NULL
set.seed(123)

for(i in 1:k)
{
  smp_size <- floor(0.70 * nrow(dat))
  index <- sample(seq_len(nrow(dat)),size=smp_size)
  train <- dat[index, ]
  test <- dat[-index, ]
  model <- glm(formula = DriverDrinking ~ Sex + Age + PrevSuspensions +
                 PrevDWIConvictions + PrevSpeeding + NumFatalities + WeekdayStatus +
                 DayStatus + IncomeToPovRatio + TotalPopulation,
              family=binomial(link='logit'), data = train)
  results_prob <- predict(model,newdata=test,type='response')
  results <- ifelse(results_prob > 0.5,1,0)
  answers <- test$DriverDrinking
  misClasificError <- mean(answers != results)
  acc[i] <- 1-misClasificError
}
mean(acc)
```

```
## [1] 0.8147884
```

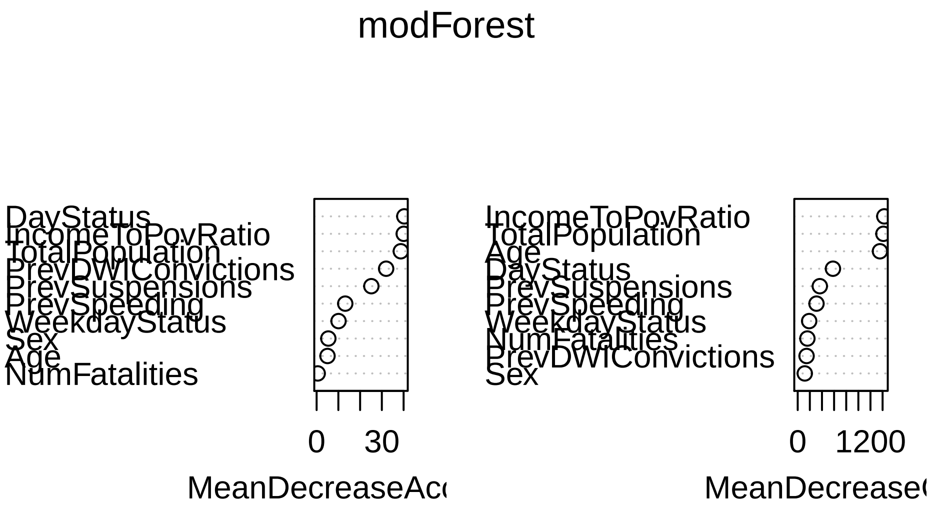
Random Forest

Variable Importance In addition to logistic regression, random forest was built to assess variable importance. Random forest is an ensemble modeling approach where output from many bootstrapped decision trees is aggregated to provide an estimate. About 100 trees were grown for the random forest model and the same predictors were regressed to predict drunk driving incidence as used for the logistic regression (Sex, Age, PrevDWIConvictions, PrevSpeeding, PrevSuspensions, NumFatalities, WeekdayStatus, DayStatus, TotalPopulation, IncomeToPovRatio).

Variable importance plot was constructed to assess which features are highly predictive in forecasting drunk driving incidence. Mean decrease in accuracy and mean decrease in gini index (measure of the model misclassification rate) were used to assess importance. County level income to poverty ratio was the most important predictor according to the mean decrease accuracy index which means that exclusion of this predictor from the model would result in the highest decrease in accuracy. In comparison, total population was the most important predictor according to the mean decrease gini index indicating that total population results in the highest node purity, which signifies a low rate of misclassifying the driver as drunk. Additionally day status and previous speeding convictions and suspension history were the next important predictors according to both the mean decrease accuracy and the mea decrease gini indices.

Least important predictors in predicting the drunk driving incident included the driver's sex and number of fatalities according to both mean decrease accuracy and mean decrease gini indices.

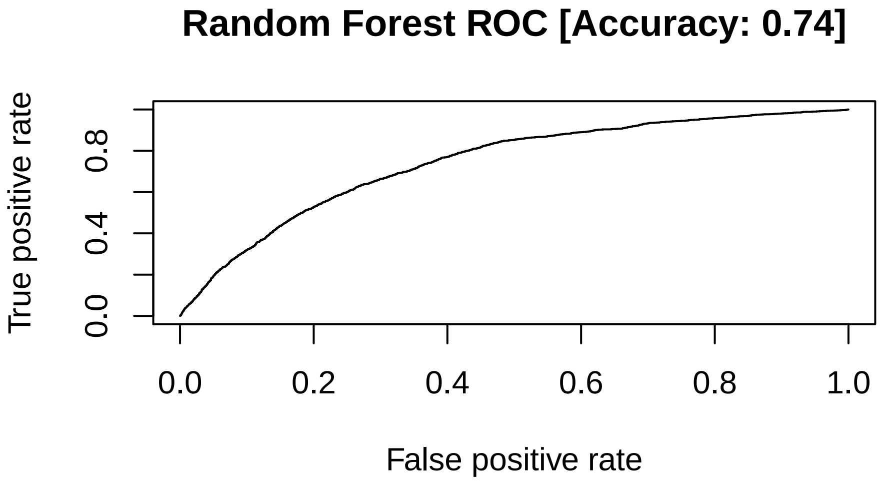
```
set.seed(1000)
modForest <- randomForest(DriverDrinking ~ Sex + Age + PrevSuspensions + PrevDWIConvictions +
                          PrevSpeeding + NumFatalities + WeekdayStatus + DayStatus + IncomeToPovRatio +
                          TotalPopulation, data = train, ntree = 100, mtry = 4,
                        keep.forest = FALSE, importance = TRUE)
#Importance plots
varImpPlot(modForest)
```



Random Forest Assessment Random Forest was also assessed via accuracy. The accuracy for the initial test and training split was 0.81. Cross-validation was then used to validate accuracy. K value of 50 was specified in cross-validation, indicating that the data was split in a test and training set 50 times and accuracy was calculated each time, producing an aggregate estimate of accuracy of about 0.83.

```
modForest1 <- randomForest(DriverDrinking ~ Sex + Age + PrevSuspensions + PrevDWIConvictions +
                            PrevSpeeding + NumFatalities + WeekdayStatus + DayStatus + IncomeToPovRatio
                            TotalPopulation, data = train, ntree = 100, mtry = 4,
                          keep.forest = TRUE, importance = FALSE)

test.forest = predict(modForest1, type = "prob", newdata = test)
forestpred = prediction(test.forest[,2], test$DriverDrinking)
forestperf = performance(forestpred, "tpr", "fpr")
plot(perf, main="Random Forest ROC [Accuracy: 0.74]")
```



```
conf <- modForest1$confusion
print(sum(diag(conf)) / sum(conf))
```

```
## [1] 0.8052709
```

```
dat <- FinalData
k <- 50
acc2 <- NULL
set.seed(123)
for(i in 1:k)
{
  smp_size <- floor(0.70 * nrow(dat))
  index <- sample(seq_len(nrow(dat)),size=smp_size)
  train <- dat[index, ]
  test <- dat[-index, ]
  modForest1 <- randomForest(DriverDrinking ~ Sex + Age + PrevSuspensions + PrevDWIConvictions +
                            PrevSpeeding + ReportedDrugs + VehicleSpeed + DeathSceneStatus +
                            NumFatalities + WeekdayStatus + DayStatus + IncomeToPovRatio +
                            TotalPopulation, data = train, ntree = 100, mtry = 4,
                          keep.forest = TRUE, importance = FALSE)
  conf <- modForest1$confusion
  acc2[i] <- (sum(diag(conf)) / sum(conf))
}
mean(acc2)
```

```
## [1] 0.8321852
```

Discussion

According to the results above, county level predictors like income to poverty ratio in the past 12 months and total population appear to be the most important predictors in predicting whether or not the driver is drunk. In regards to county level predictors, a more affluent county as indicated by a high income to poverty ratio may be more able to purchase and consume alcohol. High total population may be a confounding variable since there might naturally be an increase in the number of accidents as the population increases, irrelevant to the driver's drinking status. Additionally day status predictor is important, which seems intuitive since alcohol consumption typically occurs at night, resulting in increased drunk driving incidence during the night. Previous speeding suspensions and convictions appear to be somewhat important in predicting drunk driving incidence.

Driver's sex does not appear to be relevant in predicting drunk driving incidence. Number of fatalities is not an important variable indicating that the intensity of the crash as measured by the number of deaths in an accident does not predict the drunk driving incidence.

Limitations and Future Work

There are a few important limitations to our work. First, the FARS dataset only accounts for accidents leading to fatalities. So our models are technically predicting whether a driver involved in a fatal accident was drunk or not. The models may lose predictive value if used to determine drunk driving incidence for a driver involved in any accident (not necessarily one that involved a fatality). Second, we do not believe that observations in our dataset are independent from each other. As we noticed, there was a large difference in rates of drunk driving between certain counties. This indicates that there is some county level effect with a high chance of correlation between county level observations. This is a violation of the assumptions of logistic regression. In the future, generalized estimating equations (GEE) can be used to address this violation.

There are several natural extensions of this project that can be taken. We only included data from the year 2015 in our current analysis but would like to include additional years in consideration. Since we only included observations from counties where there were at least 10 fatal accidents, by adding more observations from different years, more counties could have been included in the analysis. Additionally, a different set of questions could have been answered by adding more years. For example, as indicated in the motivation/introduction section, there was an increase in the number of fatal traffic accidents (both overall and alcohol induced). By adding more years, one could try to ascertain why 2015 saw an increase in fatal accidents when previous years observed a decrease.

This future work extension below is slightly confusing, maybe elaborate a bit?

Another extension, is adapting the model so that it fits at an accident or county level. In order to get county estimates, we used a driver level model and aggregated the predictions. How would the model change if we only fit at a county level?

ACS Dataset

```r
data("fips.state")
data("fips.county")

######Pulling in population data
mygeo <- geo.make(state = "*", county = "*")
pop <- acs.fetch(endyear=2015, geography = mygeo,
               table.number = "B01003",
               key = "3dbabece4401ad72afa36a118e2cf777efa2afb3")
countypop <- as.data.frame(estimate(pop))
countypopT <- setDT(countypop, keep.rownames = TRUE)[]
countypopT <- countypopT %>% rename(StateCounty = rn) %>%
  mutate(StateCounty = as.factor(StateCounty))
levels(countypopT$StateCounty)[levels(countypopT$StateCounty)
                               =="Doña Ana County, New Mexico"] <- "Dona Ana County, New Mexico"

countypopT <- countypopT %>%
  separate(StateCounty, c("County.Name", "STATE_NAME"), ", ") %>% #separator by comma and space
  mutate(STATE_NAME = as.factor(STATE_NAME))

#merge county and state data
fips.state <- fips.state %>% mutate(STATE_NAME = as.factor(STATE_NAME)) %>%
  rename(State = STUSAB)

joinedData <- right_join(fips.state, fips.county, by = "State") #joinedData merge state and county by s
joinedData <- joinedData %>% mutate(County.Name = as.factor(County.Name))
countypopT <- countypopT %>% mutate(County.Name = as.factor(County.Name))

FinalPop <- inner_join(joinedData, countypopT, by = c("County.Name", "STATE_NAME")) %>%
  mutate(FIPSCode = ((1000*State.ANSI) + County.ANSI)) #find fips encoding

#####Pulling in Poverty to Income ratio data
income <- acs.fetch(endyear=2015, geography = mygeo,
               table.number = "C17002",
               key = "3dbabece4401ad72afa36a118e2cf777efa2afb3")

income <- as.data.frame(estimate(income))
income <- income %>% dplyr::select(C17002_001)
incomeT <- setDT(income, keep.rownames = TRUE)[]
incomeT <- incomeT %>% rename(StateCounty = rn) %>%
  mutate(StateCounty = as.factor(StateCounty))

levels(incomeT$StateCounty)[levels(incomeT$StateCounty)
                               =="Doña Ana County, New Mexico"] <- "Dona Ana County, New Mexico"

incomeT <- incomeT %>%
  separate(StateCounty, c("County.Name", "STATE_NAME"), ", ") %>% #separator by comma and space
  mutate(STATE_NAME = as.factor(STATE_NAME), County.Name = as.factor(County.Name))

FinalIncome <- inner_join(joinedData, incomeT, by = c("County.Name", "STATE_NAME")) %>%
  mutate(FIPSCode = ((1000*State.ANSI) + County.ANSI)) #find fips encoding


######Now merge FinalIncom1 and FinalPop1 by FipsCode
FinalCensus <- inner_join(FinalIncome, FinalPop, by = "FIPSCode")
CensusFinalD1 <- FinalCensus %>% dplyr::select(FIPSCode, C17002_001, B01003_001)
#save(CensusFinalD1, file = "CensusFinalD1.Rda")
```

Drivers Dataset

```
drivers <- read.csv("DriversData.csv") #only data for the drivers selected

#should get around 30000 observations: if more than person involved in an accident

drivers <- drivers %>% select(V_V_CONFIG, V_TRAV_SP, V_DEFORMED, V_DR_DRINK, V_PREV_ACC,
                                V_PREV_SUS, V_PREV_DWI, V_PREV_SPD, V_VALIGN, V_VPAVETYP, V_VSURCOND,
                                A_LGT_COND, A_FATALS, A_DRUNK_DR, A_WEATHER, A_WEATHER1, A_WEATHER2,
                                A_WRK_ZONE, A_MAN_COLL, P_AGE, P_SEX, P_INJ_SEV, P_SEAT_POS,
                                P_AIR_BAG, P_EJECTION, P_EJ_PATH, P_DRINKING, P_DRUGS,
                                P_LAG_HRS, P_DOA, A_COUNTY, A_STATE, A_CITY, A_DAY_WEEK, A_HOUR)

#Counting the county frequency by fips code
drivers <- drivers %>% unique() %>%
  mutate(A_STATE = readr::parse_number(A_STATE), A_COUNTY = readr::parse_number(A_COUNTY)) %>%
  mutate(FIPSCode = ((1000*A_STATE) + A_COUNTY)) #fips code

countiesL <- tally(drivers$FIPSCode) #tally by FIPSCode
filtereddrivers <- data.frame(countiesL) %>%
  rename(FIPSCode = X) %>%
  mutate(FIPSCode = readr::parse_number(FIPSCode)) %>%
  right_join(drivers)

#create new variable here: only counties
filtereddrivers <- filtereddrivers %>%
  filter(Freq > 10) %>%
  mutate(P_SEX = as.factor(P_SEX), P_DRUGS = as.factor(P_DRUGS),
         P_DOA = as.factor(P_DOA), V_DR_DRINK = as.factor(V_DR_DRINK),
         A_STATE = as.factor(A_STATE), A_COUNTY = as.factor(A_COUNTY)) %>%
  rename(state = A_STATE, county = A_COUNTY) %>%
  mutate(TravSpeed = readr::parse_number(V_TRAV_SP),
         Age = readr::parse_number(P_AGE),
         PDWI = readr::parse_number(V_PREV_DWI),
         PSuspension = readr::parse_number(V_PREV_SUS),
         PCrash = readr::parse_number(V_PREV_ACC),
         PSpeed = readr::parse_number(V_PREV_SPD))

ACS <- load("CensusFinalD1.Rda")
FinalData <- CensusFinalD1 %>% mutate(FIPSCode = readr::parse_number(FIPSCode)) %>%
  right_join(filtereddrivers, by = "FIPSCode") #40422 observations
FinalData <- na.omit(FinalData)

########Build training and test models here
FinalData <- FinalData %>%
  rename(IncomeToPovRatio = C17002_001, #Quant
         TotalPopulation = B01003_001, #Quant
         Sex = P_SEX, #Categorical
         PrevDWIConvictions = PDWI, #Quantitative
         PrevSpeeding = PSpeed, #Quantitative
         PrevSuspensions = PSuspension, #Quant
         VehicleSpeed = TravSpeed, #Quant
         ReportedDrugs = P_DRUGS, #Categorical
         NumFatalities = A_FATALS, #Quant
         DeathSceneStatus = P_DOA, #Categorical
         DriverDrinking = V_DR_DRINK) %>% #Categorical
  filter(Sex == '1' | Sex == '2')

FinalData$Sex <- droplevels(FinalData$Sex)
levels(FinalData$Sex) <- c("Male", "Female")

#Creating weekend/weekday predictor
FinalData <- FinalData %>%
  mutate(WeekdayStatus = ifelse(A_DAY_WEEK == '1' | A_DAY_WEEK == '7', "Weekend", "Weekday"))
FinalData$WeekdayStatus <- as.factor(FinalData$WeekdayStatus)

FinalData <- FinalData %>%
  mutate(DayStatus = ifelse(A_HOUR < 6 | A_HOUR >= 18, "Night", "Day"))
FinalData$DayStatus <- as.factor(FinalData$DayStatus)

#save(FinalData, file = "FinalData.Rda")
```