



Data Analytics Bootcamp

NRS DUI Data Analysis

TEAM # 4

BENJAMIN, ELIZABETH, JAGRATI, NICK

Agenda

- Motivation & Summary
- Questions & Data
- Data Cleanup & Exploration
- Data Analysis
- Conclusion
- Post Mortem
- Q&A



Core Message

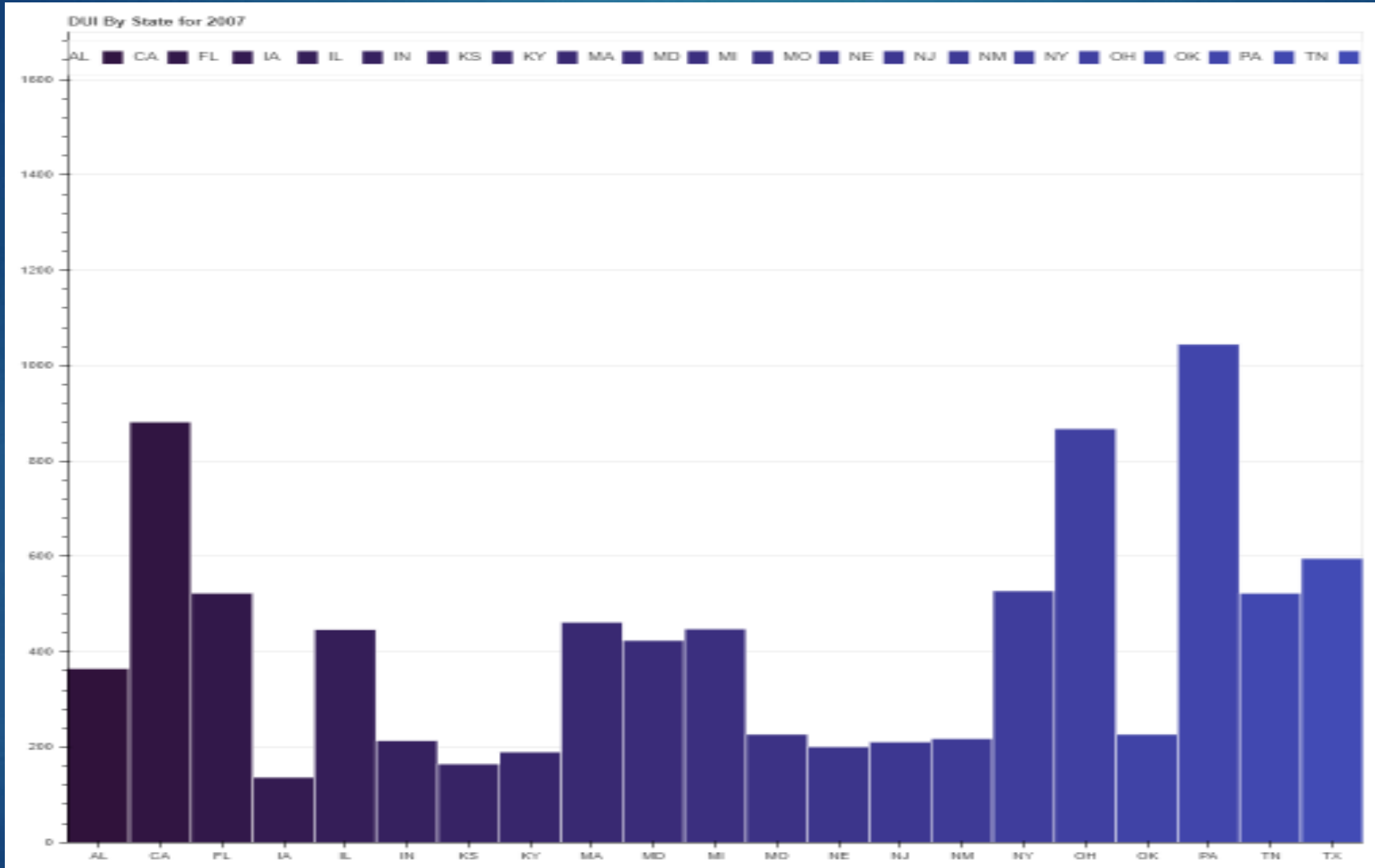
Core Message

- ▶ Do not drink and drive, otherwise you will become a statistic! Here is some info on where the worst offenders of DUI's are in the United states, and facts about demographics and comparing them to census statistics.



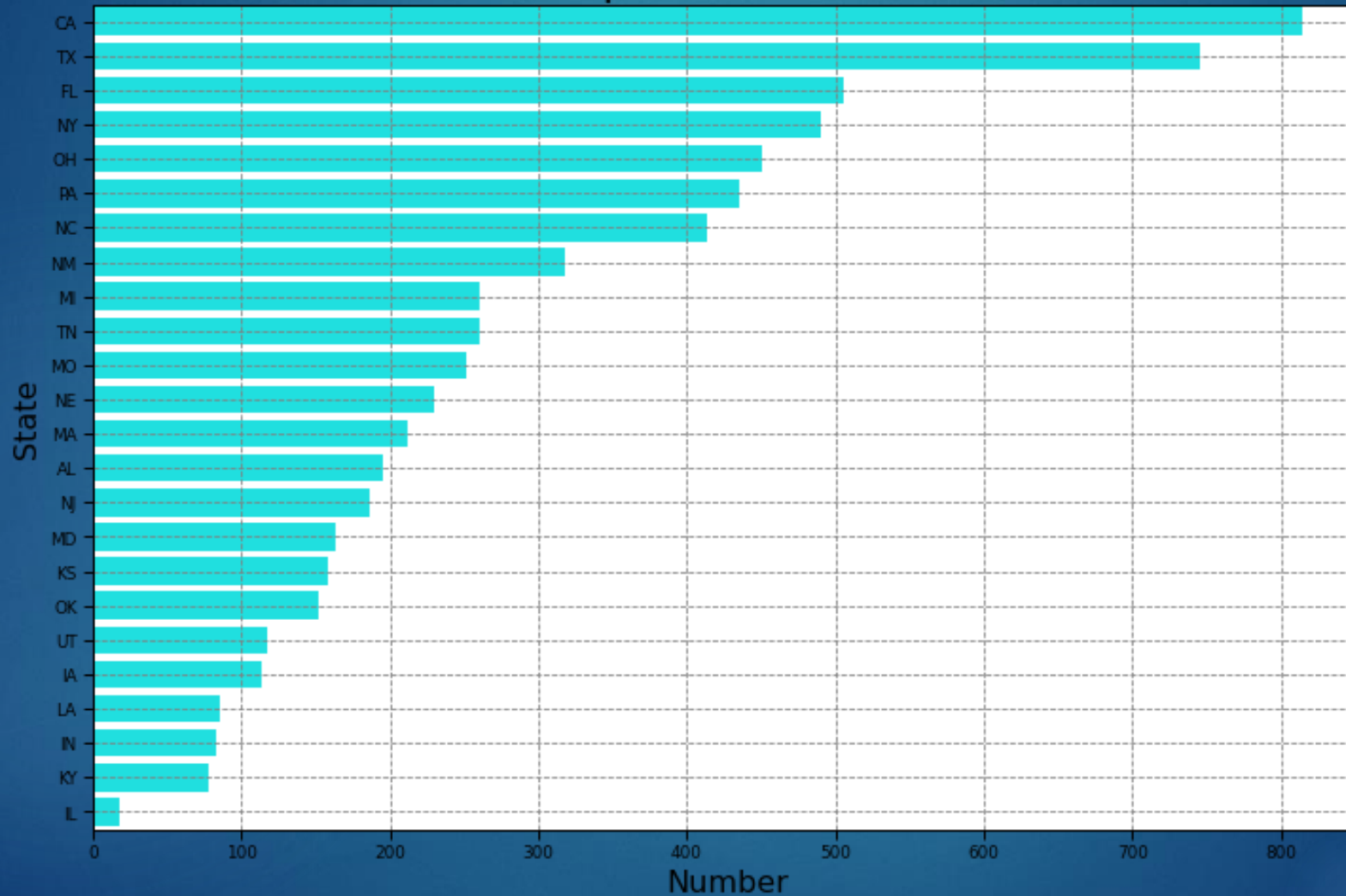
Questions & Data

DUI by State for 2007

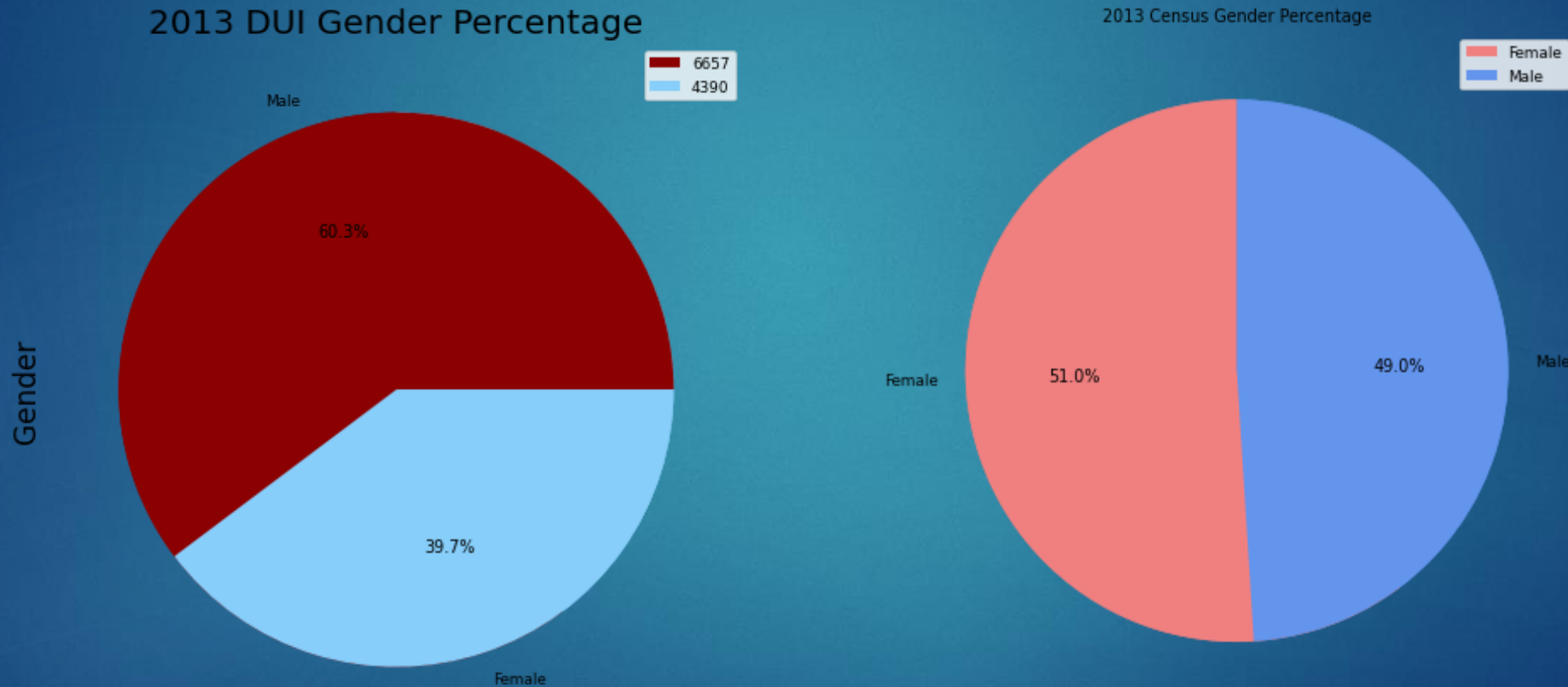


DUI by State for 2013

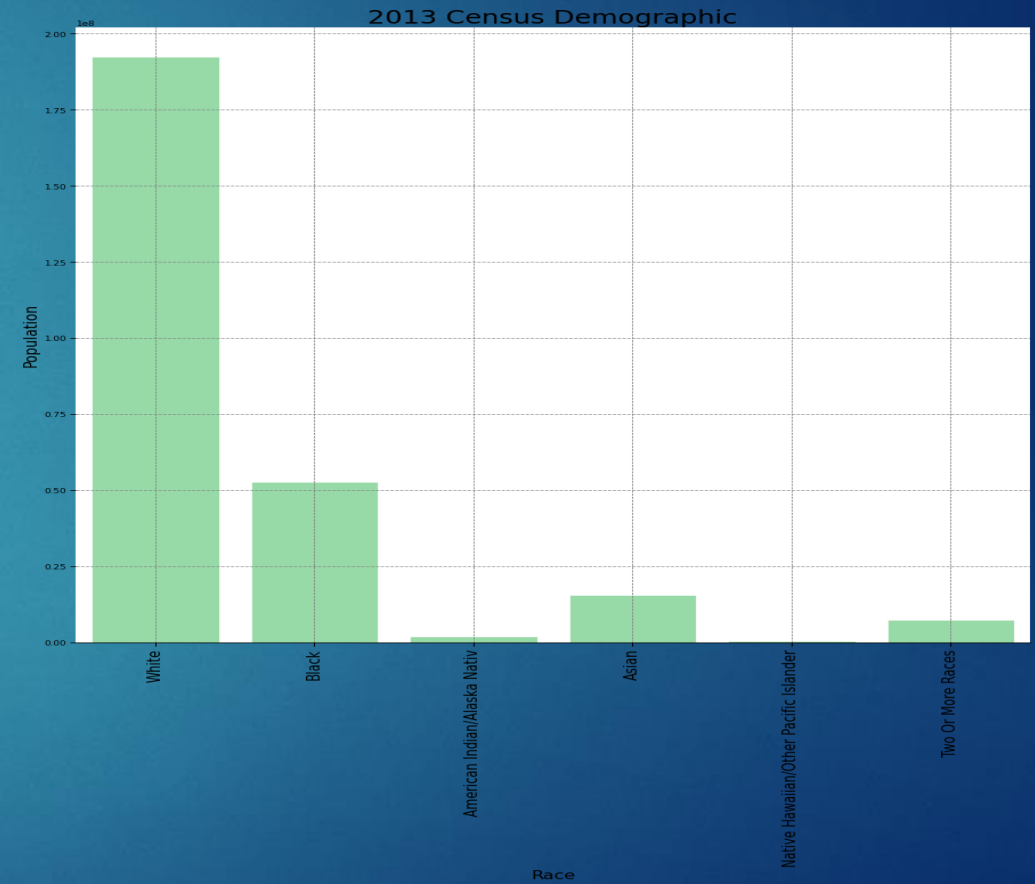
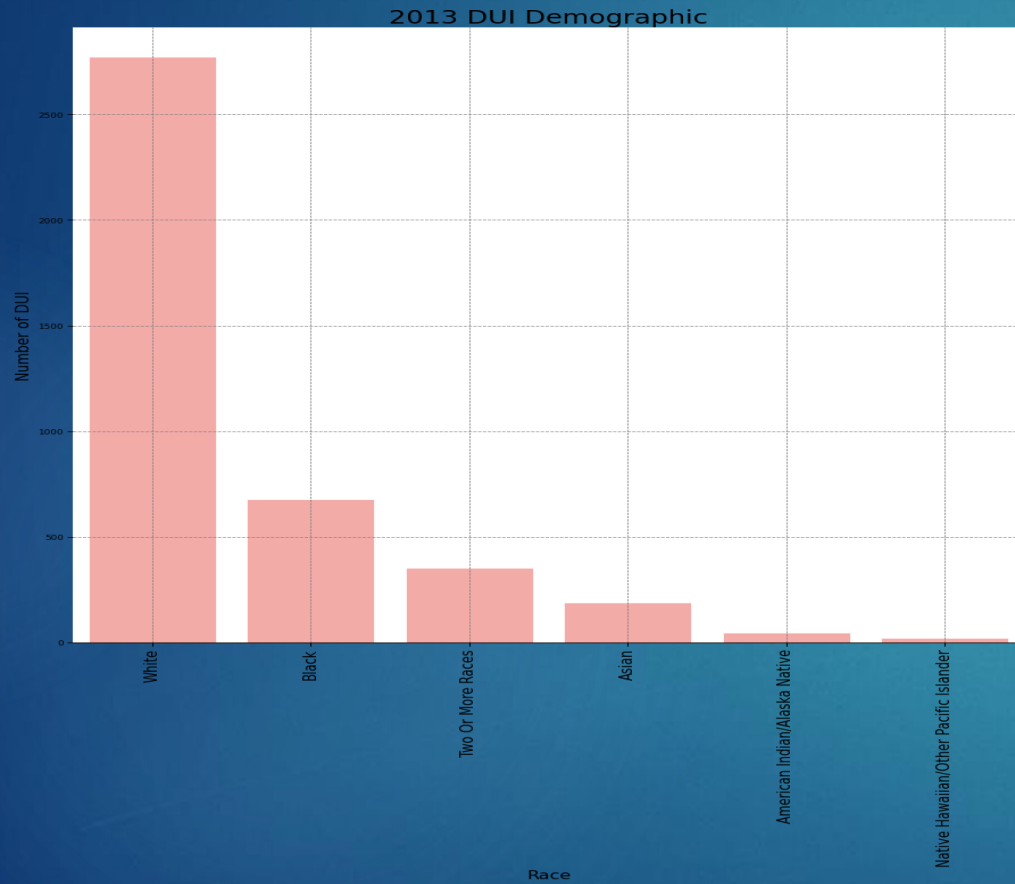
DUI Comparison of 27 states



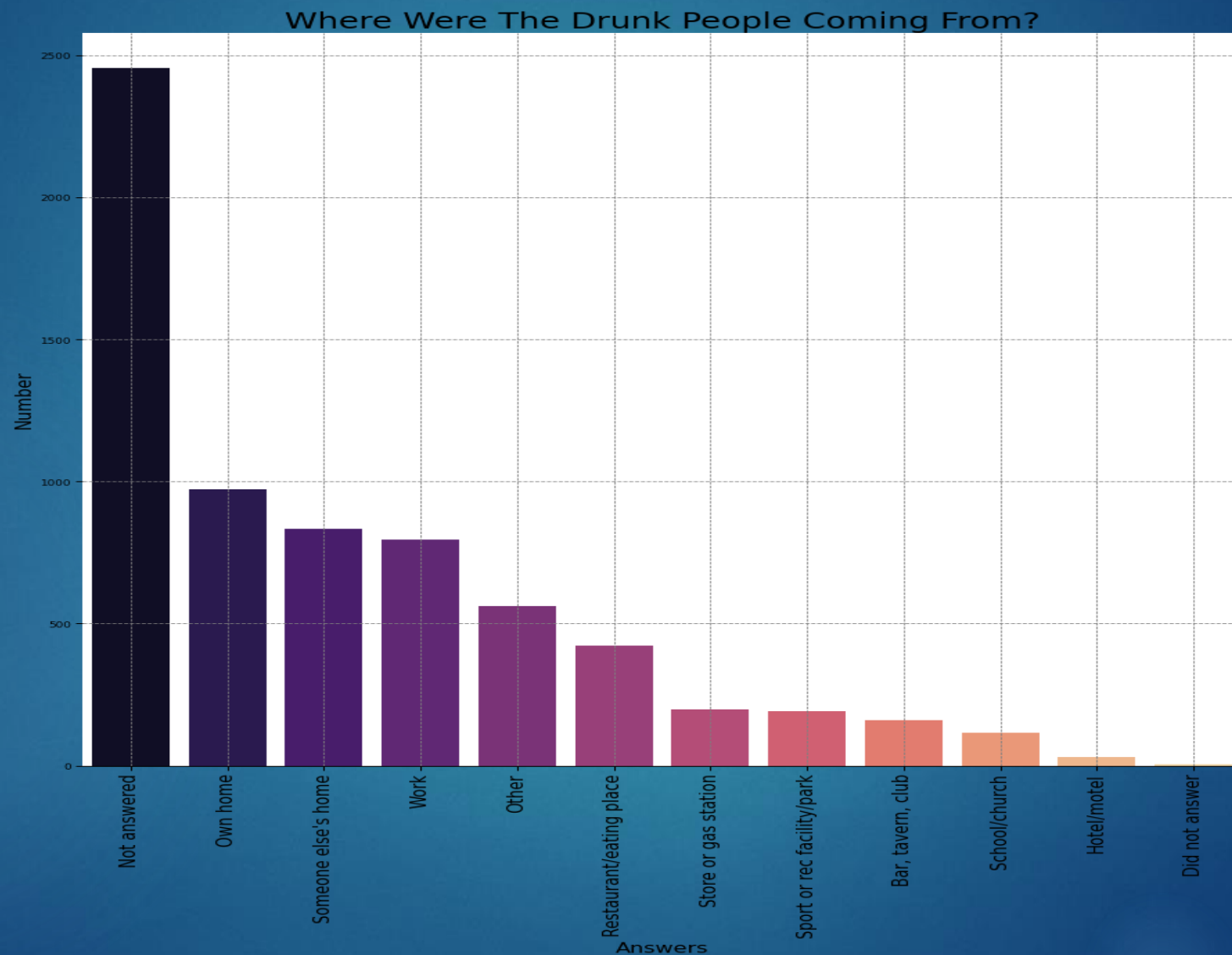
What are gender percent of DUIs? How does this compare with census population?



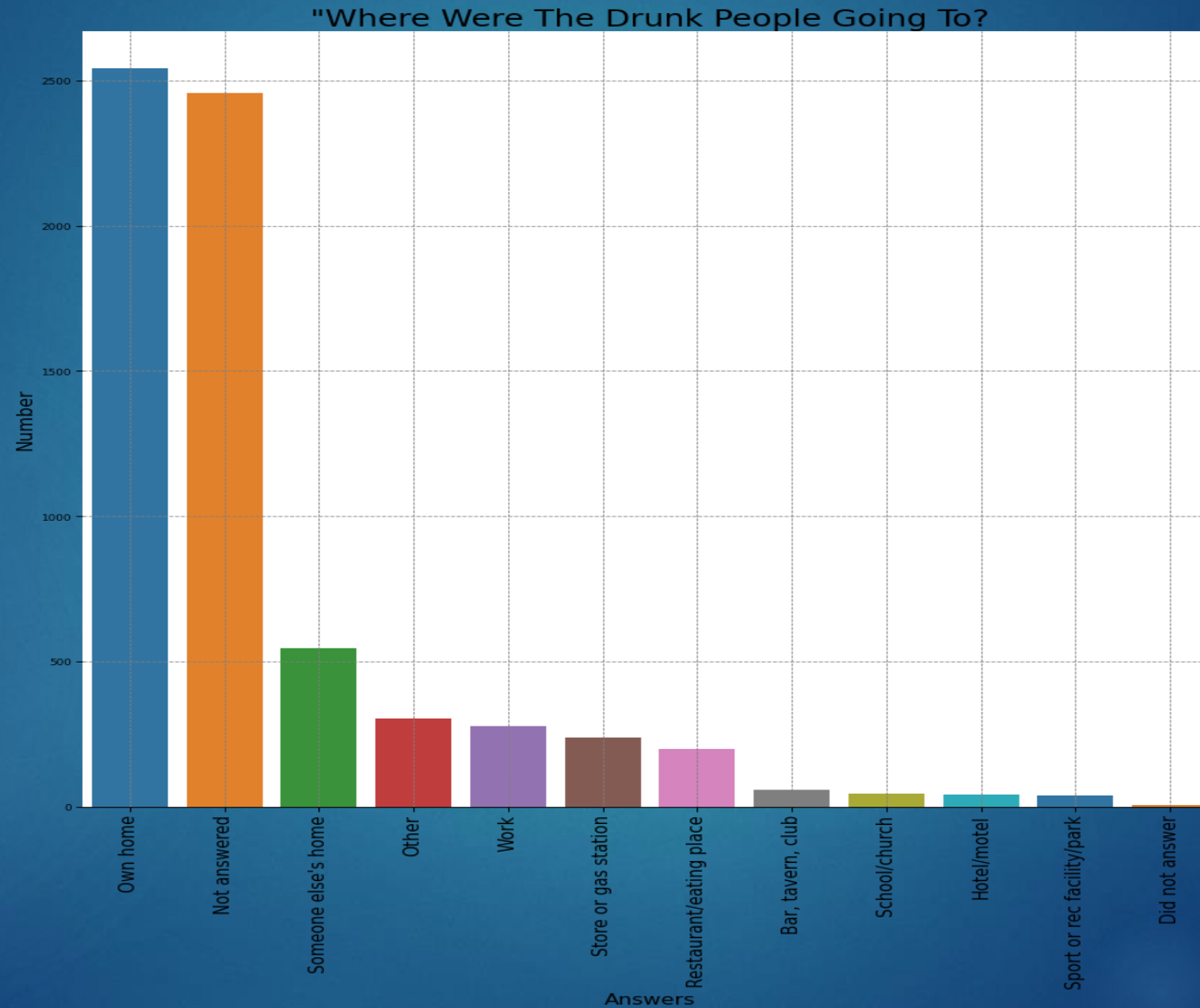
How do census demographics compare to DUI demographic statistics?



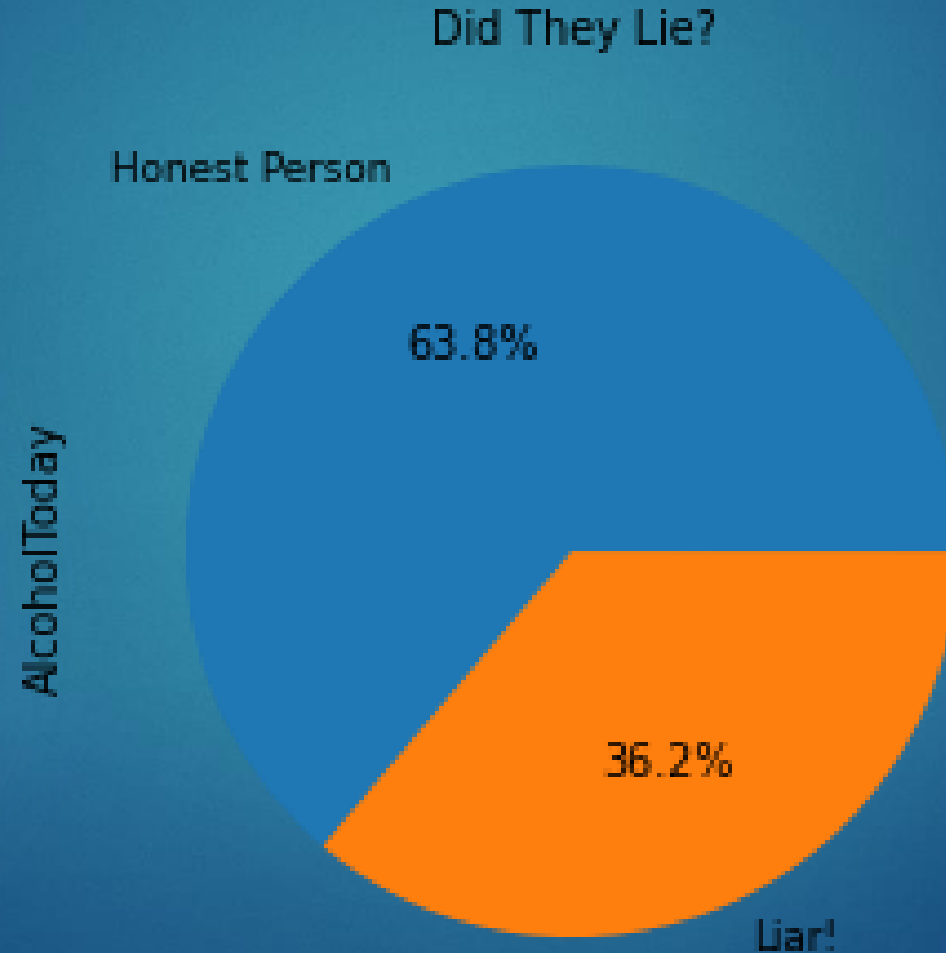
Where were most people coming from?



Where Were The Drunk People Going To?

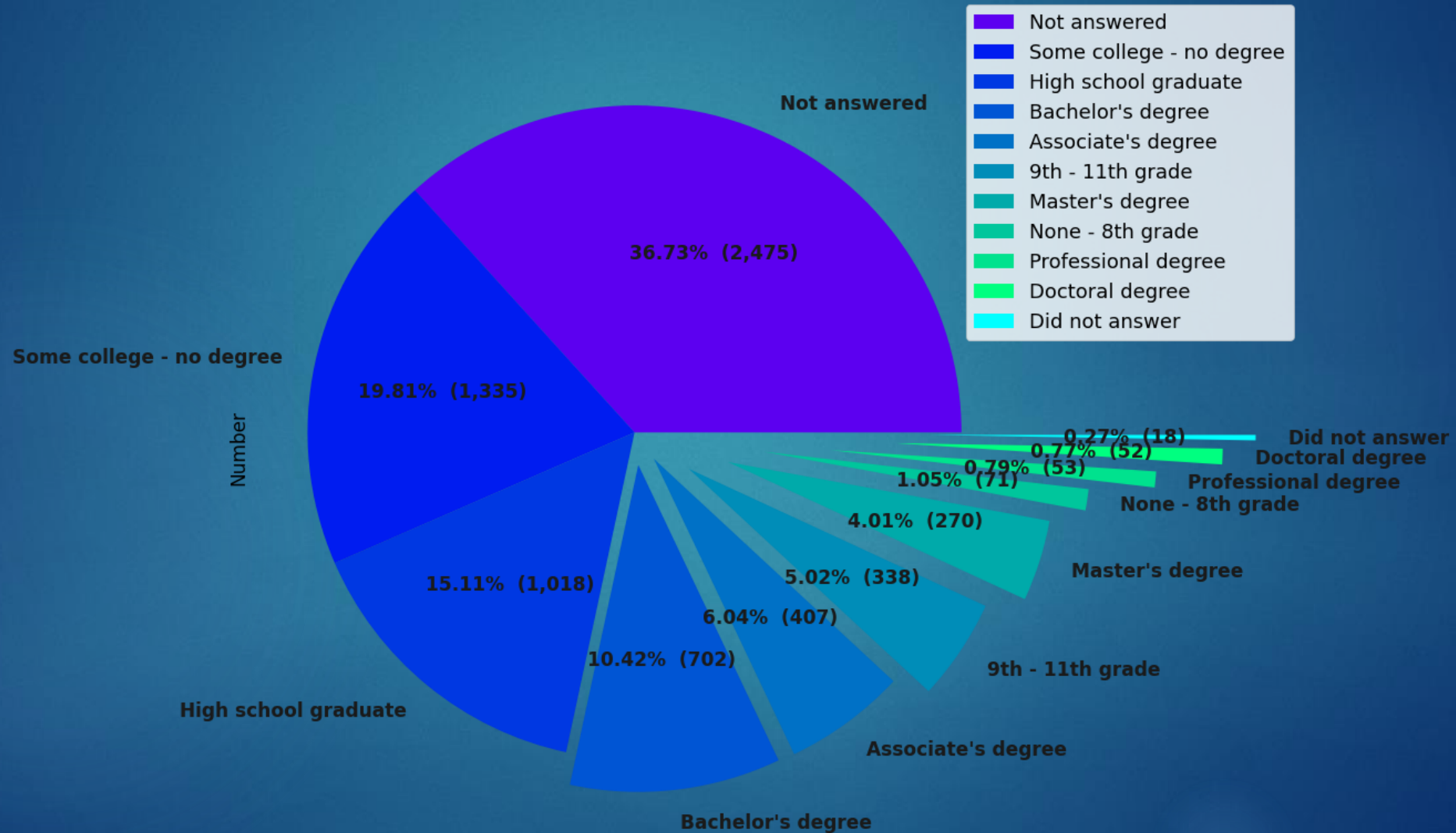


How many people who drove drunk told the truth about drinking that day?

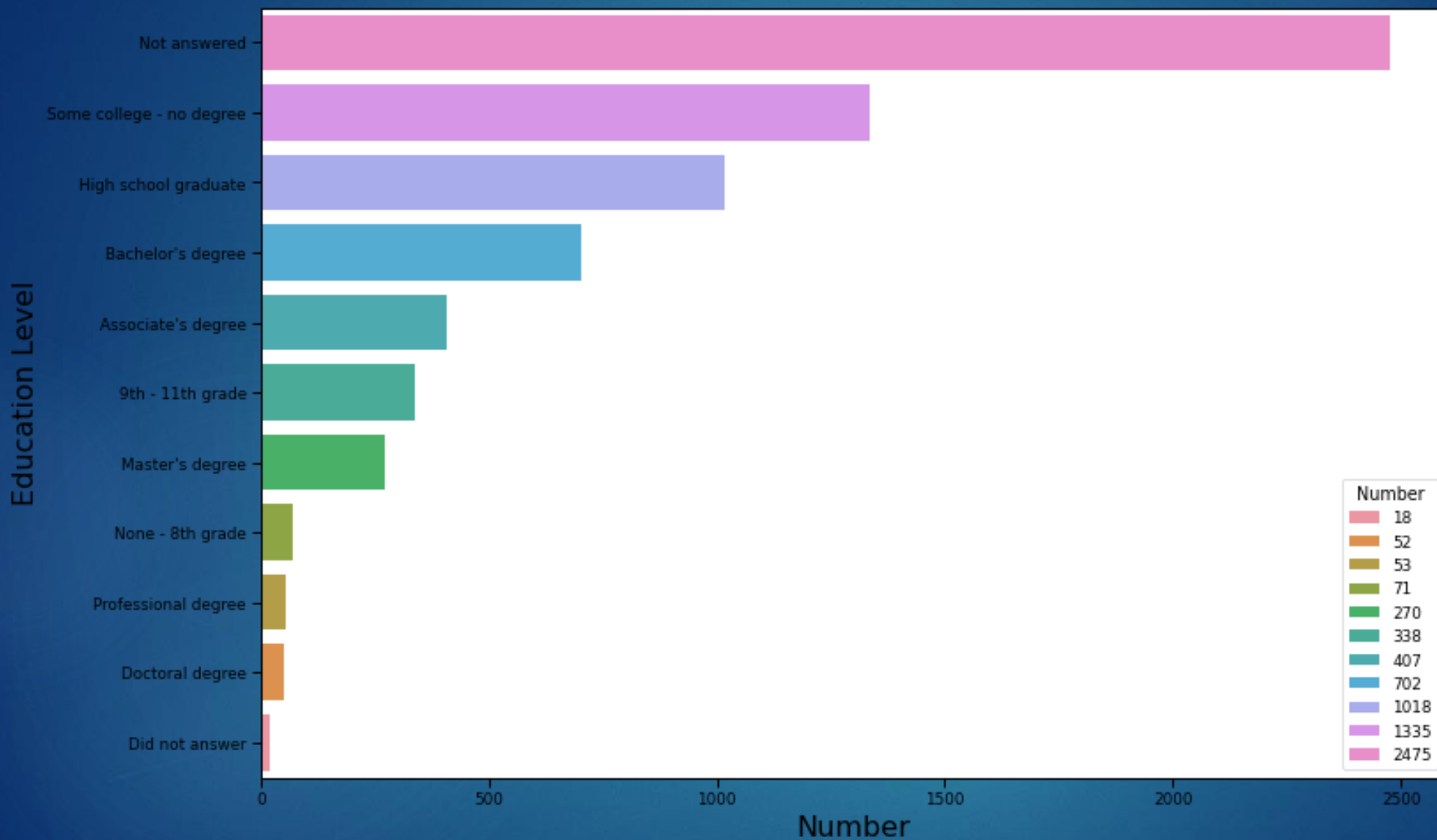


Are the educated more likely to drink?

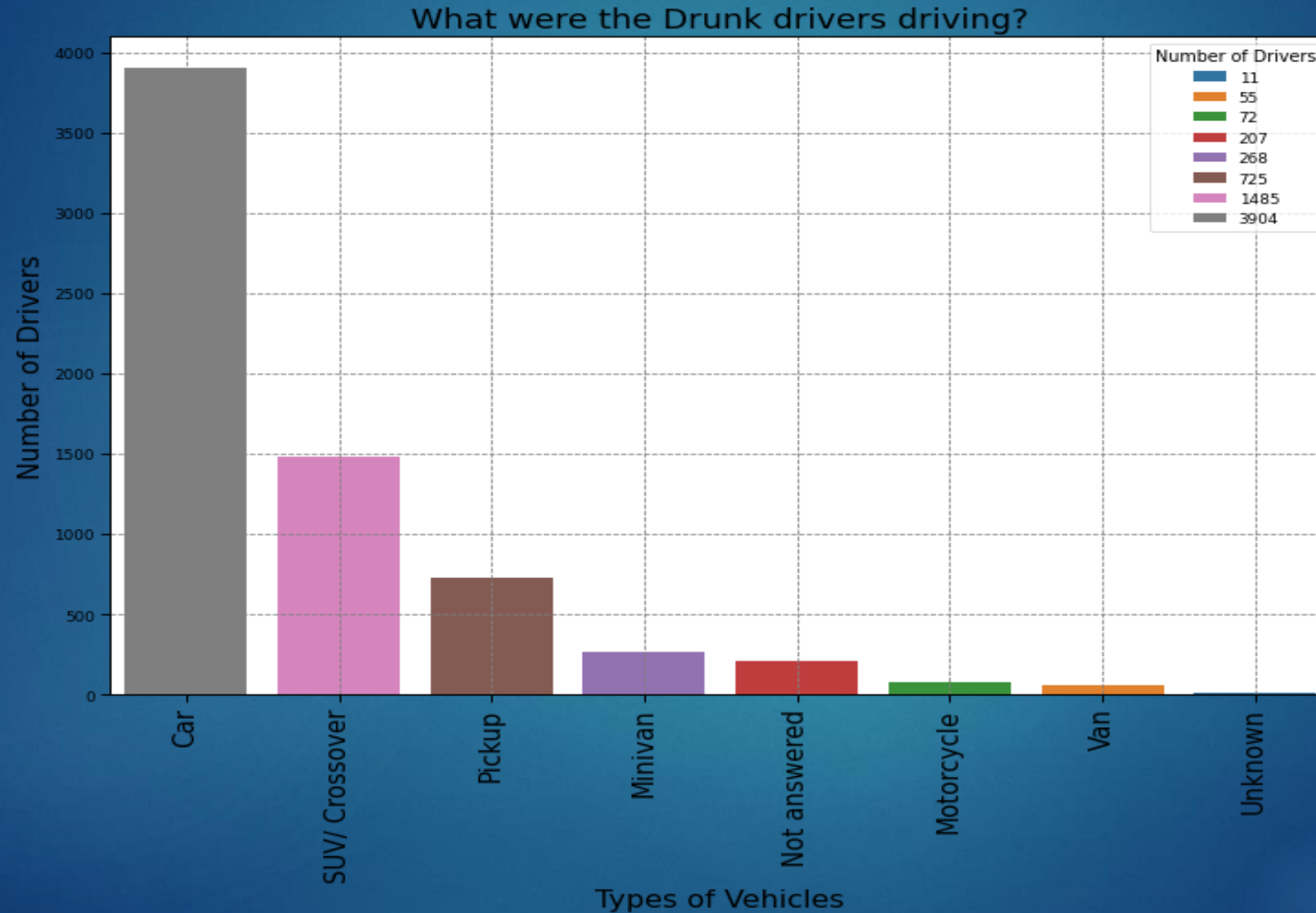
How Educated Are Drunk Drivers?

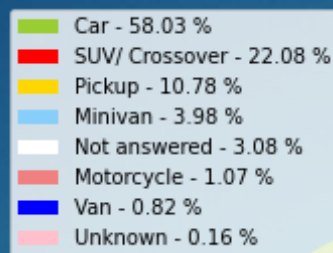


Education of Drunk Drivers 2013



What were the Drunk drivers driving?





Car - 58.03 %

SUV/ Crossover - 22.08 %

Pickup - 10.78 %

Minivan - 3.98 %

Not answered - 3.08 %

Motorcycle - 1.07 %


Van - 0.82 %

Unknown - 0.16 %











Data Cleanup & Exploration

Raw Data from NRS for 2007 and 2013

localhost:8888/notebooks/GitLab/Team-4-Beer-API/analysis.ipynb

jupyter analysis Last Checkpoint: 16 hours ago (autosaved)  Logo

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

          Code

```
In [9]: # Dependencies and Setup
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

In [19]: # Reading raw csv files
data07_path = "Resource/nrs_2007_final.csv"
data13_path = "Resource/nrs_2013_combined.csv"

data07_raw_df = pd.read_csv(data07_path)
print(f"2007 dataframe size is {data07_raw_df.shape}")
data13_raw_df = pd.read_csv(data13_path)
print(f"2013 dataframe size is {data13_raw_df.shape}")

2007 dataframe size is (11120, 313)
2013 dataframe size is (11322, 995)

In [21]: # Getting selected columns for the 2007 data
col_list = ["STATE", "DATE", "weight", "NRS_DAGE", "NRS_DRACE", "NRS_PERDR",
            "NRS_LIVEMI", "NRS_FROM", "NRS_HEADED", "NRS_BETMI", "NRS_TODRK",
            "NRS_AGE", "NRS_ZIPCODE", "NRS_SCHOOL", "NRS_EMPLOY", "NRS_RACE",
            "NRS_VTYPE", "NRS_DSEX", "DQP2Q28", "DQP2Q29C", "DQP2Q32", "BLD_DRUG",
            "BLD_ETHANOL", "OF_DRUG", "resulbac2"]
data07_df = pd.read_csv(data07_path, low_memory=False, usecols=col_list)

# Checking list of columns imported from CSV to ensure matchh with sort listed columns
col_list_check = data07_df.columns.values.tolist()
col_list_check
```


Selected Data Import

```
2007 dataframe size is (11120, 313)
2013 dataframe size is (11322, 995)
```

```
In [11]: # Getting selected columns for the 2007 data
col_list = ["STATE", "DATE", "weight", "NRS_DAGE", "NRS_DRACE", "NRS_PERDR",
            "NRS_LIVEMI", "NRS_FROM", "NRS_HEADED", "NRS_BETMI", "NRS_TODRK",
            "NRS_AGE", "NRS_ZIPCODE", "NRS_SCHOOL", "NRS_EMPLOY", "NRS_RACE",
            "NRS_VTYPE", "NRS_DSEX", "DQP2Q28", "DQP2Q29C", "DQP2Q32", "BLD_DRUG",
            "BLD_ETHANOL", "OF_DRUG", "resulbac2"]
data07_df = pd.read_csv(data07_path, usecols=col_list)

# Checking list of columns imported from CSV to ensure matchh with sort listed columns
col_list_check = data07_df.columns.values.tolist()
col_list_check
```

```
C:\Users\Jagrati\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3071: DtypeWarning: Columns (38) have mixed ty
pes.Specify dtype option on import or set low_memory=False.
```

```
has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

```
Out[11]: ['STATE',
          'DATE',
          'weight',
          'NRS_DAGE',
          'NRS_DRACE',
          'NRS_PERDR',
          'NRS_LIVEMI',
          'NRS_FROM',
```

Mismatch in data columns

Before

	A	B	C	D	E	F
1	2013	2007	matching cloum name			
2	'State',	['STATE',	'State',			
3	'nighttime',	'DATE',	Time_Stamp_n			
4	'nightday',			protocol49_only_NEW_wgt		
5	'Alcohol',	'NRS_DAGE',				
6	'race_obs',	'NRS_DRACE',				
7	'Gender',	'NRS_PERDR',	DayTimePct			
8	'VehicleType',	'NRS_LIVEMI',	MilesTraveling			
9	'FromWhere',	'NRS_FROM',	'FromWhere',			
10	'ToWhere',	'NRS_HEADED',	'ToWhere',			
11	'AlcoholToday',	'NRS_BETMI',				
12	'DriverAgeYears',	'NRS_TODRK',	'AlcoholToday',			
13	'EducationLevel',	'NRS_AGE',	'DriverAgeYears',			
14	'AreYouStudent',	'NRS_ZIPCODE',	MyZipCodeIs			
15	'AreYouEmployed',	'NRS_SCHOOL',	'EducationLevel',			
16	'Race',	'NRS_EMPLOY',	'AreYouEmployed',			
17	'bac_from_blood']	'NRS_RACE',	race_n			
18		'NRS_VTYPE',	'VehicleType',			
19		'NRS_DSEX',	'Gender',			
20		'DQP2Q28',	GotArrested			
21		'DQP2Q29C',				
22		'DQP2Q32',				
23		'BLD_DRUG',				
24		'BLD_ETHANOL',	mg_dL2			
25		'OF_DRUG',				
26		'resulbac2']	bac_from_blood			
27			'nighttime',			
28			'nightday',			
29			'Alcohol',			

After

Matching Column Header Name	2007	2013
'State'	STATE	State
'Zipcode'	NRS_ZIPCODE	MyZipCodeIs
'Date'	DATE	Time_Stamp
'Day or Night?'	session	data_session_original
'Race'	NRS_RACE	race_n
'Gender'	NRS_DSEX	Gender
'Age Range'	NRS_AGE	DriverAgeYears
'Education Level'	NRS_SCHOOL	EducationLevel
'Student'	NRS_EMPLOY	AreYouStudent
'Employed'	NRS_EMPLOY	AreYouEmployed
'Vehicle Type'	NRS_VTYPE	VehicleType
'From Where'	NRS_FROM	FromWhere
'To Where'	NRS_HEADED	ToWhere
'Trip distance'	NRS_BETMI	MilesTraveling
'AlcoholToday'	NRS_TODRK	AlcoholToday
'Blood Alcohol Content'	resulbac2	bac_from_blood

NaN values in the data

Before

```
11118    1    PA    NaN    Restaurant / eating place    Home (own home)    16 - 20
11119    3    IN    NaN    Work    Home (own home)    16 - 20

11120 rows × 15 columns
```

```
In [45]: # Checking count of NA or NaN values in the dataframe
data07_df.isna().sum()
```

```
Out[45]: session          0
STATE              0
DATE              7646
NRS_FROM          2147
NRS_HEADED        2148
NRS_BETMI         2147
NRS_TODRK         5074
NRS_AGE           0
NRS_ZIPCODE       2202
NRS_SCHOOL        2174
NRS_EMPLOY        2177
NRS_RACE          2179
NRS_VTYPE         73
NRS_DSEX          73
resulbac2         0
dtype: int64
```

After

```
11102    1    VA    8/10/2007    Home (own home)    Store or stat

1914 rows × 15 columns
```

```
In [47]: # Checking count of NA or NaN values in the data
data07_df.isna().sum()
```

```
Out[47]: session          0
STATE              0
DATE              0
NRS_FROM          0
NRS_HEADED        0
NRS_BETMI         0
NRS_TODRK         0
NRS_AGE           0
NRS_ZIPCODE       0
NRS_SCHOOL        0
NRS_EMPLOY        0
NRS_RACE          0
NRS_VTYPE         0
NRS_DSEX          0
resulbac2         0
dtype: int64
```

#NULL! values in 1 column

```
In [59]: # Checking data after cleanup
data13_df
```

Out[59]:

Where	ToWhere	MilesTraveling	AlcoholToday	DriverAgeYears	MyZipCodeIs	EducationLevel	AreYouStudent	AreYouEmployed	bac_from_blood
Other	Other	0-5	No	43.0	93003.0	High school graduate	No	Employed Full-time	#NULL!
Other	Own home	More than 20	No	21.0	93033.0	Some college - no degree	No	Employed Full-time	#NULL!
Other	Own home	0-5	Yes	23.0	93035.0	Some college - no degree	Other/ Technical or Trade Program	Other	#NULL!
else's home	Own home	0-5	No	19.0	93030.0	Some college - no degree	College/ Grad or Law School	Employed Part-time	0
Other	Other	More than 20	No	23.0	93033.0	Some college - no degree	No	Employed Full-time	0
...
Work	Own home	11-20	No	21.0	84041.0	High school graduate	No	Employed Part-time	#NULL!
home	Someone else's home	0-5	No	19.0	84010.0	High school graduate	High School	Employed Full-time	#NULL!
home	School/church	More than 20	No	23.0	84321.0	Associate's degree	College/ Grad or Law School	Employed Full-time	#NULL!
eating place	Own home	0-5	No	40.0	84010.0	High school graduate	No	Employed Full-time	0
or gas station	Other	6-10	No	20.0	84606.0	Some college - no degree	College/ Grad or Law School	Employed Part-time	0

Adding City Name based on Zipcode

```
In [19]: search = SearchEngine(simple_zipcode=True)
for index, row in data_13.iterrows():
    try:
        zipcode = row["MyZipCodeIs"]
        getCityData = search.by_zipcode(zipcode)
        data_13.loc[index, "CityName"] = getCityData.major_city
    except ValueError:
        print(f"Skipping {index} due to invalid zipcode data")

data_13.head()
```

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexing.py:845: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

self.obj[key] = _infer_fill_value(value)
C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexing.py:966: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
self.obj[item] = s

Out[19]:

	data_session_original	State	Time_Stamp	race_n	Gender	VehicleType	FromWhere	ToWhere	MilesTraveling	AlcoholToday	DriverAgeYears	MyZipCodeIs
3	3	CA	12/07/2013 02:34:15	8	Male	Car	Someone else's home	Own home	0-5	No	19.0	93030.0
5	3	CA	12/07/2013 13:28:52	1	Male	Car	Other	Other	More than 20	No	23.0	93033.0

Distribute NRS_EMPLOY column to 2 parts

```
In [13]: # Renaming NRS_EMPLOY to AreYouEmployed to match data with 2013
data07_df = data07_df.rename(columns={"NRS_EMPLOY": "AreYouEmployed"})
data07_df.head()
```

```
Out[13]:
```

	session	STATE	DATE	NRS_FROM	NRS_HEADED	NRS_BETMI	NRS_TODRK	NRS_AGE	NRS_ZIPCODE	NRS_SCHOOL	AreYouEmployed	NRS_RA
1968	4	IL	7/22/2007	Work	Home (own home)	0 - 5	No	20	60154	Some college	Employed/self-employed	Hispa
1990	3	CA	7/28/2022	Other	Home (own home)	6 - 10	Yes	23	90732	High school graduate	Employed/self-employed	Wh
1992	1	CA	7/27/2007	Home (own home)	Restaurant / eating place	0 - 5	No	31	90503	Some college	Employed/self-employed	Wh
1993	3	WI	8/4/2007	Work	Someone else's home	6 - 10	No	46	53018	Some college	Employed/self-employed	Wh
1994	2	IL	7/21/2007	Work	Someone else's home	0 - 5	No	21	60613	College graduate	Employed/self-employed	Asi

```
In [14]: # Creating a copy of AreYouEmployed column to match with 2013 AreYouStudent column
data07_df["AreYouStudent"] = data07_df["AreYouEmployed"]
data07_df.head()
```

```
Out[14]:
```

	session	STATE	DATE	NRS_FROM	NRS_HEADED	NRS_BETMI	NRS_TODRK	NRS_AGE	NRS_ZIPCODE	NRS_SCHOOL	AreYouEmployed	NRS_RA
1968	4	IL	7/22/2007	Work	Home (own home)	0 - 5	No	20	60154	Some college	Employed/self-employed	Hispa
1990	3	CA	7/28/2022	Other	Home (own home)	6 - 10	Yes	23	90732	High school graduate	Employed/self-employed	Wh
1992	1	CA	7/27/2007	Home (own home)	Restaurant / eating place	0 - 5	No	31	90503	Some college	Employed/self-employed	Wh
1993	3	WI	8/4/2007	Work	Someone else's home	6 - 10	No	46	53018	Some college	Employed/self-employed	Wh
1994	2	IL	7/21/2007	Work	Someone else's home	0 - 5	No	21	60613	College graduate	Employed/self-employed	Asi

```
In [15]: # Checking unique values in AreYouEmployed column
data07_df["AreYouEmployed"].unique()
```

```
Out[15]: array(['Employed/self-employed', 'Student', 'On Disability', 'Retired',
        'Unemployed', 'Homemaker', 'Other'], dtype=object)
```

```
In [16]: # Converting AreYouEmployed to binary
data07_df["AreYouEmployed"] = data07_df["AreYouEmployed"].apply(lambda x: 1 if x == "Employed/self-employed" else 0)
data07_df
```

```
Out[16]:
```

	session	STATE	DATE	NRS_FROM	NRS_HEADED	NRS_BETMI	NRS_TODRK	NRS_AGE	NRS_ZIPCODE	NRS_SCHOOL	AreYouEmployed	NRS_RA
1968	4	IL	7/22/2007	Work	Home (own home)	0 - 5	No	20	60154	Some college	1	Hispa
					Home (own					High school		

Change gender from numeric to string

```
In [20]: # Converting gender from numeric to string
data07_df["NRS_DSEX"] = data07_df["NRS_DSEX"].apply(lambda x: "Male" if x == 1.0 else "Female")
data07_df
```

```
Out[20]:
```

	session	STATE	DATE	NRS_FROM	NRS_HEADED	NRS_BETMI	NRS_TODRK	NRS_AGE	NRS_ZIPCODE	NRS_SCHOOL	AreYouEmployed	NRS_RA
1968	4	IL	7/22/2007	Work	Home (own home)	0 - 5	No	20	60154	Some college	1	Hispi
1990	3	CA	7/28/2022	Other	Home (own home)	6 - 10	Yes	23	90732	High school graduate	1	W
1992	1	CA	7/27/2007	Home (own home)	Restaurant / eating place	0 - 5	No	31	90503	Some college	1	W
1993	3	WI	8/4/2007	Work	Someone else's home	6 - 10	No	46	53018	Some college	1	W
1994	2	IL	7/21/2007	Work	Someone else's home	0 - 5	No	21	60613	College graduate	1	A
...
11091	4	IL	7/22/2022	Other	Someone else's home	6 - 10	Yes	46	60077	High school graduate	1	W
11095	5	NE	8/5/2007	Someone else's home	Someone else's home	0 - 5	Yes	25	68154	College graduate	1	W
11097	2	AL	7/21/2007	Home (own home)	Restaurant / eating place	More than 20	No	21	35150	High school graduate	1	Blac Afri Ameri
11101	2	CA	7/28/2007	Someone else's home	Home (own home)	More than 20	No	28	90810	Some college	1	Blac Afri Ameri
11102	1	VA	8/10/2007	Home (own home)	Store or gas station	0 - 5	No	41	23227	High school graduate	1	Blac Afri Ameri

1914 rows × 17 columns

Change race from numeric to string

In [38]: *# Replacing numeric race values with string*

```
race_dict = {1: "White",
             2: "Black/African American",
             3: "Asian",
             4: "Native American/Alaskan",
             5: "Native Hawaiian/Other Pacific Islander",
             6: "Unknown",
             7: "Other",
             8: "More than one",
             9: "no response"}

data13_df["race_n"] = data13_df["race_n"].map(lambda x: race_dict.get(x) if x in race_dict else x)
data13_df
```


Out[38]:

	data_session_original	State	Time_Stamp	race_n	Gender	VehicleType	FromWhere	ToWhere	MilesTraveling	AlcoholToday	DriverAgeYears	
	3	3	CA	12/07/2013 02:34:15	More than one	Male	Car	Someone else's home	Own home	0-5	No	19
	5	3	CA	12/07/2013 13:28:52	White	Male	Car	Other	Other	More than 20	No	21
	9	1	FL	06/28/2013 10:54:21	White	Male	SUV/ Crossover	Own home	Other	6-10	No	41
	15	4	FL	06/29/2013 23:13:17	White	Male	Car	Own home	Own home	11-20	Yes	21
	22	5	FL	06/30/2013 02:30:54	Unknown	Female	Car	Someone else's home	Own home	0-5	No	21

	11287	4	OK	09/07/2013 23:42:59	White	Male	Pickup	Someone else's home	Own home	6-10	No	21
	11292	5	OK	09/08/2013 03:08:10	White	Male	Pickup	Restaurant/eating place	Own home	0-5	Did not answer	21
	11299	1	UT	02/07/2014 16:37:07	White	Male	Car	Store or gas station	Own home	0-5	No	21
	11314	4	UT	02/08/2014 23:01:07	Native Hawaiian/Other Pacific Islander	Female	Car	Restaurant/eating place	Own home	0-5	No	41
	11320	5	UT	02/09/2014 02:03:32	White	Female	Car	Store or gas station	Other	6-10	No	21

3177 rows × 17 columns

Data Analysis & Conclusions

- 
- ▶ Higher populations, as expected, have more DUI's
 - ▶ However it seems that Southern States have a much higher percentage of DUI vs population than other regions of the United states
 - ▶ Although there are more females than males in the United States, men tend to get DUI's at a nearly double the rate of women
 - ▶ As for ethnicities, DUI data seems to fit the census data, showing that race does not play a role in DUI rates.
 - ▶ You shouldn't drive to or from anywhere drunk, but it looks like most people who drink are coming and going from their home or someone elses. Somehow, a good amount are coming or going to work.
 - ▶ Surprisingly, 64% people tell the truth to officers when asked if they were drinking that day. Our hypothesis was that more would lie, but most people are honest drunks!
 - ▶ It seems DUI's are given to less educated at a higher rate, but this might account for the fact that there are less in the population that have higher education.
 - ▶ Cars are the most common offenders, most likely due to cars being the most common vehicle, but SUV's were a pretty big offender as well.

Post Mortem

Q&A

The background of the slide features a dark blue field filled with numerous bright, diagonal light streaks that create a sense of motion and depth. In the upper right corner, there is a solid yellow rectangular block.

Thank You!