

```
In [573]: import pandas as pd
df = pd.read_csv('Terry_Stops.csv')
df.head()
```

Out[573]:

	Subject Age Group	Subject ID	GO / SC Num	Terry Stop ID	Stop Resolution	Weapon Type	Officer ID	Officer YOB	Officer Gender	Officer Race	...	Reported Time	Initial Call Type	Final Ca
0	36 - 45	8597903457	20190000243853	8597867579	Field Contact	-	5469	1967	M	White	...	06:45:50.0000000	TRESPASS	--SUSP CIF SUSP P
1	26 - 35	-1	20160000003391	181276	Field Contact	None	7591	1985	M	Hispanic or Latino	...	22:48:00.0000000	-	
2	18 - 25	7774286580	20210000118915	24056783769	Field Contact	-	7459	1973	M	White	...	12:16:51.0000000	SFD - ASSIST ON FIRE OR MEDIC RESPONSE	DISTUR -
3	26 - 35	-1	20180000047173	400437	Offense Report	None	6680	1972	M	Hispanic or Latino	...	18:31:00.0000000	-	
4	18 - 25	-1	20160000085711	136669	Offense Report	None	7560	1986	M	White	...	11:44:00.0000000	SUSPICIOUS PERSON, VEHICLE OR INCIDENT	--SUSP CIF SUSP P

5 rows × 23 columns

```
In [574]: df.shape
```

Out[574]: (58157, 23)

In [575]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58157 entries, 0 to 58156
Data columns (total 23 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Subject Age Group                    58157 non-null  object
1   Subject ID                          58157 non-null  int64
2   GO / SC Num                        58157 non-null  int64
3   Terry Stop ID                      58157 non-null  int64
4   Stop Resolution                    58157 non-null  object
5   Weapon Type                        58157 non-null  object
6   Officer ID                        58157 non-null  object
7   Officer YOB                       58157 non-null  int64
8   Officer Gender                    58157 non-null  object
9   Officer Race                      58157 non-null  object
10  Subject Perceived Race              58157 non-null  object
11  Subject Perceived Gender            58157 non-null  object
12  Reported Date                      58157 non-null  object
13  Reported Time                      58157 non-null  object
14  Initial Call Type                  58157 non-null  object
15  Final Call Type                    58157 non-null  object
16  Call Type                          58157 non-null  object
17  Officer Squad                      57613 non-null  object
18  Arrest Flag                        58157 non-null  object
19  Frisk Flag                         58157 non-null  object
20  Precinct                          58157 non-null  object
21  Sector                            58157 non-null  object
22  Beat                              58157 non-null  object
dtypes: int64(4), object(19)
memory usage: 10.2+ MB
```

```
In [576]: df.isna().sum()
```

```
Out[576]: Subject Age Group      0
Subject ID      0
GO / SC Num     0
Terry Stop ID   0
Stop Resolution  0
Weapon Type     0
Officer ID      0
Officer YOB     0
Officer Gender  0
Officer Race    0
Subject Perceived Race  0
Subject Perceived Gender  0
Reported Date   0
Reported Time   0
Initial Call Type  0
Final Call Type  0
Call Type       0
Officer Squad   544
Arrest Flag     0
Frisk Flag      0
Precinct        0
Sector          0
Beat            0
dtype: int64
```

```
In [577]: df.dropna(inplace=True)
df.isna().sum()
```

```
Out[577]: Subject Age Group      0
Subject ID      0
GO / SC Num     0
Terry Stop ID   0
Stop Resolution  0
Weapon Type     0
Officer ID      0
Officer YOB     0
Officer Gender  0
Officer Race    0
Subject Perceived Race  0
Subject Perceived Gender  0
Reported Date   0
Reported Time   0
Initial Call Type  0
Final Call Type  0
Call Type       0
Officer Squad   0
Arrest Flag     0
Frisk Flag      0
Precinct        0
Sector          0
Beat            0
dtype: int64
```

In [578]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 57613 entries, 1 to 58156
Data columns (total 23 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Subject Age Group                    57613 non-null  object
1   Subject ID                          57613 non-null  int64
2   GO / SC Num                        57613 non-null  int64
3   Terry Stop ID                      57613 non-null  int64
4   Stop Resolution                    57613 non-null  object
5   Weapon Type                        57613 non-null  object
6   Officer ID                        57613 non-null  object
7   Officer YOB                       57613 non-null  int64
8   Officer Gender                    57613 non-null  object
9   Officer Race                      57613 non-null  object
10  Subject Perceived Race              57613 non-null  object
11  Subject Perceived Gender            57613 non-null  object
12  Reported Date                      57613 non-null  object
13  Reported Time                      57613 non-null  object
14  Initial Call Type                  57613 non-null  object
15  Final Call Type                    57613 non-null  object
16  Call Type                          57613 non-null  object
17  Officer Squad                      57613 non-null  object
18  Arrest Flag                        57613 non-null  object
19  Frisk Flag                         57613 non-null  object
20  Precinct                          57613 non-null  object
21  Sector                            57613 non-null  object
22  Beat                              57613 non-null  object
dtypes: int64(4), object(19)
memory usage: 10.5+ MB
```

In [579]: `df = df.drop_duplicates()`

```
In [580]: df["Stop Resolution"]
```

```
Out[580]: 1      Field Contact
          2      Field Contact
          3      Offense Report
          4      Offense Report
          5      Arrest
          ...
          58152      Arrest
          58153      Arrest
          58154      Field Contact
          58155      Field Contact
          58156      Offense Report
          Name: Stop Resolution, Length: 57613, dtype: object
```

```
In [581]: df["Stop Resolution"].unique()
```

```
Out[581]: array(['Field Contact', 'Offense Report', 'Arrest',
                  'Referred for Prosecution', 'Citation / Infraction'], dtype=object)
```

```
In [582]: new_df = df[["Subject Age Group", "Stop Resolution", "Officer Gender", "Officer Race", "Subject Perceived Gender", "Subject Perceived Race", "Arrest Flag", "Frisk Flag"]]
```

Out[582]:

	Subject Age Group	Stop Resolution	Officer Gender	Officer Race	Subject Perceived Gender	Subject Perceived Race	Arrest Flag	Frisk Flag
1	26 - 35	Field Contact	M	Hispanic or Latino	Female	Unknown	N	N
2	18 - 25	Field Contact	M	White	Male	White	N	Y
3	26 - 35	Offense Report	M	Hispanic or Latino	Male	White	N	N
4	18 - 25	Offense Report	M	White	Male	Black or African American	N	N
5	26 - 35	Arrest	M	White	Male	Black or African American	Y	N
...
58152	18 - 25	Arrest	M	White	Male	Black or African American	N	Y
58153	18 - 25	Arrest	M	White	Male	White	N	N
58154	36 - 45	Field Contact	M	Black or African American	Male	White	N	N
58155	36 - 45	Field Contact	M	White	Male	Black or African American	N	N
58156	36 - 45	Offense Report	M	White	Male	Black or African American	N	N

57613 rows × 8 columns

In [583]: `from sklearn.preprocessing import OneHotEncoder`

```
# Columns to one-hot encode
columns_to_encode = ["Arrest Flag", "Frisk Flag"]

# Initialize the OneHotEncoder
encoder = OneHotEncoder(sparse=False, drop='first')

# Fit and transform the DataFrame
one_hot_encoded = encoder.fit_transform(new_df[columns_to_encode])

# Create a new DataFrame with the one-hot encoded values
one_hot_df = pd.DataFrame(one_hot_encoded, columns=encoder.get_feature_names_out(columns_to_encode))

# Concatenate the new DataFrame with the original DataFrame
new_df = pd.concat([new_df, one_hot_df], axis=1)

# Drop the original categorical columns from the new DataFrame
new_df.drop(columns=columns_to_encode, inplace=True)

# Print the result
new_df
```

C:\Users\admin\anaconda3\Lib\site-packages\sklearn\preprocessing_encoders.py:972: FutureWarning: `sparse` was renamed to `sparse_output` in version 1.2 and will be removed in 1.4. `sparse_output` is ignored unless you leave `sparse` to its default value.

warnings.warn(

Out[583]:

	Subject Age Group	Stop Resolution	Officer Gender	Officer Race	Subject Perceived Gender	Subject Perceived Race	Arrest Flag_Y	Frisk Flag_N	Frisk Flag_Y
1	26 - 35	Field Contact	M	Hispanic or Latino	Female	Unknown	0.0	0.0	1.0
2	18 - 25	Field Contact	M	White	Male	White	0.0	1.0	0.0
3	26 - 35	Offense Report	M	Hispanic or Latino	Male	White	0.0	1.0	0.0
4	18 - 25	Offense Report	M	White	Male	Black or African American	1.0	1.0	0.0
5	26 - 35	Arrest	M	White	Male	Black or African American	0.0	0.0	1.0
...
57158	NaN	NaN	NaN	NaN	NaN	NaN	0.0	1.0	0.0
57174	NaN	NaN	NaN	NaN	NaN	NaN	0.0	1.0	0.0
57175	NaN	NaN	NaN	NaN	NaN	NaN	0.0	1.0	0.0
57278	NaN	NaN	NaN	NaN	NaN	NaN	0.0	1.0	0.0
57566	NaN	NaN	NaN	NaN	NaN	NaN	0.0	1.0	0.0

58145 rows × 9 columns

```
In [584]: new_df["Officer Race"].unique()
```

```
Out[584]: array(['Hispanic or Latino', 'White', 'Nat Hawaiian/Oth Pac Islander',  
                'Two or More Races', 'Not Specified', 'Black or African American',  
                'Asian', 'American Indian/Alaska Native', 'Unknown', nan],  
              dtype=object)
```

```
In [585]: new_df["Subject Age Group"].unique()
```

```
Out[585]: array(['26 - 35', '18 - 25', '-', '1 - 17', '36 - 45', '56 and Above',  
                '46 - 55', nan], dtype=object)
```

```
In [ ]:
```

```
In [586]: ▶ import statsmodels as sm
import sklearn.preprocessing as preprocessing
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from scipy import stats
import numpy as np
```

```
In [587]: ▶ # Convert race and sex using get_dummies()
x_feats = ["Officer Race", "Subject Perceived Race", "Officer Gender", "Subject Perceived Gender"]
X = pd.get_dummies(new_df[x_feats], drop_first=True, dtype=float)

# Convert target using get_dummies
y = pd.get_dummies(new_df["Arrest Flag_Y"], drop_first=True, dtype=float)

# Rename the column to be consistent with the Logistic regression assumption
y.columns = ['Arrest_Flag']
```

```
In [588]: ▶ from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
```

In [589]: `import statsmodels.api as sm`

```
# Create intercept term required for sm.Logit, see documentation for more information
X = sm.add_constant(X_train)

# Fit model
logit_model = sm.Logit(y_train, X_train)

# Get results of the fit
result = logit_model.fit()
result.summary()
```

Warning: Maximum number of iterations has been exceeded.
Current function value: 0.334629
Iterations: 35

C:\Users\admin\anaconda3\Lib\site-packages\statsmodels\base\model.py:607: ConvergenceWarning: Maximum Likelihood optimization failed to converge. Check mle_retvals
warnings.warn("Maximum Likelihood optimization failed to "

Out[589]:

Logit Regression Results

Dep. Variable: Arrest_Flag **No. Observations:** 43608
Model: Logit **Df Residuals:** 43584
Method: MLE **Df Model:** 23
Date: Sat, 02 Dec 2023 **Pseudo R-squ.:** -0.01127
Time: 00:06:09 **Log-Likelihood:** -14592.
converged: False **LL-Null:** -14430.
Covariance Type: nonrobust **LLR p-value:** 1.000

	coef	std err	z	P> z	[0.025	0.975]
Officer Race_Asian	-1.0741	0.158	-6.803	0.000	-1.384	-0.765
Officer Race_Black or African American	-1.1662	0.160	-7.310	0.000	-1.479	-0.854
Officer Race_Hispanic or Latino	-1.0218	0.153	-6.659	0.000	-1.322	-0.721
Officer Race_Nat Hawaiian/Oth Pac Islander	-1.2047	0.227	-5.305	0.000	-1.650	-0.760
Officer Race_Not Specified	-1.0684	0.189	-5.642	0.000	-1.440	-0.697
Officer Race_Two or More Races	-1.0540	0.176	-5.982	0.000	-1.399	-0.709
Officer Race_Unknown	-12.1832	156.352	-0.078	0.938	-318.627	294.261
Officer Race_White	-1.0138	0.144	-7.055	0.000	-1.295	-0.732
Subject Perceived Race_American Indian or Alaska Native	-0.0271	0.040	-0.679	0.497	-0.105	0.051
Subject Perceived Race_Asian	-0.1672	0.052	-3.217	0.001	-0.269	-0.065
Subject Perceived Race_Black or African American	-0.0305	nan	nan	nan	nan	nan
Subject Perceived Race_DUPLICATE	0.1101	nan	nan	nan	nan	nan
Subject Perceived Race_Hispanic	-0.1472	0.084	-1.746	0.081	-0.312	0.018
Subject Perceived Race_Multi-Racial	-0.1465	0.105	-1.400	0.162	-0.352	0.059
Subject Perceived Race_Native Hawaiian or Other Pacific Islander	-0.2922	0.355	-0.822	0.411	-0.989	0.404
Subject Perceived Race_Other	0.5015	0.231	2.169	0.030	0.048	0.955
Subject Perceived Race_Unknown	0.0568	0.036	1.600	0.110	-0.013	0.126
Subject Perceived Race_White	-0.0067	nan	nan	nan	nan	nan
Officer Gender_M	-0.0485	0.045	-1.080	0.280	-0.137	0.039
Subject Perceived Gender_DUPLICATE	0.1101	nan	nan	nan	nan	nan
Subject Perceived Gender_Female	-1.0902	0.053	-20.703	0.000	-1.193	-0.987
Subject Perceived Gender_Gender Diverse (gender non-conforming and/or transgender)	-0.7559	0.540	-1.399	0.162	-1.815	0.303

Subject Perceived Gender_Male	-1.0825	nan	nan	nan	nan	nan
Subject Perceived Gender_Unable to Determine	-1.1305	0.243	-4.661	0.000	-1.606	-0.655
Subject Perceived Gender_Unknown	-0.9163	0.379	-2.418	0.016	-1.659	-0.174

```
In [590]: np.exp(result.params)
```

```
Out[590]: Officer Race_Asian                                0.341615
Officer Race_Black or African American                    0.311548
Officer Race_Hispanic or Latino                          0.359964
Officer Race_Nat Hawaiian/Oth Pac Islander               0.299780
Officer Race_Not Specified                               0.343571
Officer Race_Two or More Races                           0.348527
Officer Race_Unknown                                     0.000005
Officer Race_White                                        0.362843
Subject Perceived Race_American Indian or Alaska Native  0.973274
Subject Perceived Race_Asian                             0.846038
Subject Perceived Race_Black or African American         0.969966
Subject Perceived Race_DUPLICATE                         1.116378
Subject Perceived Race_Hispanic                         0.863159
Subject Perceived Race_Multi-Racial                     0.863711
Subject Perceived Race_Native Hawaiian or Other Pacific  0.746617
Islander
Subject Perceived Race_Other                             1.651116
Subject Perceived Race_Unknown                          1.058490
Subject Perceived Race_White                             0.993309
Officer Gender_M                                         0.952651
Subject Perceived Gender_DUPLICATE                     1.116378
Subject Perceived Gender_Female                        0.336144
Subject Perceived Gender_Gender Diverse (gender non-con 0.469589
forming and/or transgender)
Subject Perceived Gender_Male                           0.338745
Subject Perceived Gender_Unable to Determine            0.322862
Subject Perceived Gender_Unknown                       0.400014
dtype: float64
```

```
In [591]: model_log.coef_
```

```
Out[591]: array([[ -1.07380115,  -1.16592594,  -1.02148029,  -1.20435648,  -1.06808804,
    -1.0537694 ,  -4.78110185,  -1.01352402,  -0.02726226,  -0.16736489,
    -0.03064991,   0.11121594,  -0.14733636,  -0.14671854,  -0.29248295,
     0.50067857,   0.05669608,  -0.00686996,  -0.04851256,   0.11121594,
    -1.09031518,  -0.75733462,  -1.08260815,  -1.13068   ,  -0.91712143]])
```

```
In [592]: logreg = LogisticRegression(fit_intercept=False, C=1e15, solver='liblinear')
model_log = logreg.fit(X_train, y_train)

y_hat_train = logreg.predict(X_train)
y_hat_test = logreg.predict(X_test)
```

C:\Users\admin\anaconda3\Lib\site-packages\sklearn\utils\validation.py:1184: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().

```
y = column_or_1d(y, warn=True)
```

```
In [593]: from sklearn.metrics import mean_squared_error

train_mse = mean_squared_error(y_train, y_hat_train)
test_mse = mean_squared_error(y_test, y_hat_test)
print('Train Mean Squared Error:', train_mse)
print('Test Mean Squared Error:', test_mse)
```

Train Mean Squared Error: 0.10266464868831407
Test Mean Squared Error: 0.10462956593519983

```
In [594]: import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import OneHotEncoder
from sklearn import tree
```

```
In [595]: clf = DecisionTreeClassifier(criterion='entropy')

clf.fit(X_train, y_train)

other_train_mse = mean_squared_error(y_train, clf.predict(X_train))
other_test_mse = mean_squared_error(y_test, clf.predict(X_test))
print('Train Mean Squared Error:', other_train_mse)
print('Test Mean Squared Error:', other_test_mse)
```

Train Mean Squared Error: 0.10252705925518253
Test Mean Squared Error: 0.10476714590355644

```
In [596]: y_pred = clf.predict(X_test)
y_pred
```

Out[596]: array([0., 0., 0., ..., 0., 0., 0.]

```
In [597]: ▶ acc = accuracy_score(y_test,y_pred) * 100  
print("Accuracy: {0}" .format(acc))
```

Accuracy: 89.52328540964436

```
In [598]: ▶ clf.score(X_train, y_train)
```

Out[598]: 0.8974729407448174

In [599]: `from sklearn.ensemble import AdaBoostClassifier`

```
# Create a base decision tree classifier
base_dt_classifier_boost = DecisionTreeClassifier(max_depth=1)

# Create an AdaBoostClassifier
adaboost_classifier = AdaBoostClassifier(base_estimator=base_dt_classifier_boost, n_estimators=50, random_state=42)

# Train the AdaBoostClassifier
adaboost_classifier.fit(X_train, y_train)

# Evaluate AdaBoost performance
adaboost_predictions = adaboost_classifier.predict(X_test)
print("AdaBoost Accuracy:", accuracy_score(y_test, adaboost_predictions))
print("AdaBoost Classification Report:")
print(classification_report(y_test, adaboost_predictions))
```

C:\Users\admin\anaconda3\Lib\site-packages\sklearn\utils\validation.py:1184: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().

`y = column_or_1d(y, warn=True)`

C:\Users\admin\anaconda3\Lib\site-packages\sklearn\ensemble_base.py:156: FutureWarning: `base_estimator` was renamed to `estimator` in version 1.2 and will be removed in 1.4.

`warnings.warn(`

AdaBoost Accuracy: 0.8953704340648002

AdaBoost Classification Report:

	precision	recall	f1-score	support
0.0	0.90	1.00	0.94	13016
1.0	0.00	0.00	0.00	1521
accuracy			0.90	14537
macro avg	0.45	0.50	0.47	14537
weighted avg	0.80	0.90	0.85	14537

C:\Users\admin\anaconda3\Lib\site-packages\sklearn\metrics_classification.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

`_warn_prf(average, modifier, msg_start, len(result))`

C:\Users\admin\anaconda3\Lib\site-packages\sklearn\metrics_classification.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

`_warn_prf(average, modifier, msg_start, len(result))`

C:\Users\admin\anaconda3\Lib\site-packages\sklearn\metrics_classification.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

`_warn_prf(average, modifier, msg_start, len(result))`

In []:

In [600]:

new_df

Out[600]:

	Subject Age Group	Stop Resolution	Officer Gender	Officer Race	Subject Perceived Gender	Subject Perceived Race	Arrest Flag_Y	Frisk Flag_N	Frisk Flag_Y
1	26 - 35	Field Contact	M	Hispanic or Latino	Female	Unknown	0.0	0.0	1.0
2	18 - 25	Field Contact	M	White	Male	White	0.0	1.0	0.0
3	26 - 35	Offense Report	M	Hispanic or Latino	Male	White	0.0	1.0	0.0
4	18 - 25	Offense Report	M	White	Male	Black or African American	1.0	1.0	0.0
5	26 - 35	Arrest	M	White	Male	Black or African American	0.0	0.0	1.0
...
57158	NaN	NaN	NaN	NaN	NaN	NaN	0.0	1.0	0.0
57174	NaN	NaN	NaN	NaN	NaN	NaN	0.0	1.0	0.0
57175	NaN	NaN	NaN	NaN	NaN	NaN	0.0	1.0	0.0
57278	NaN	NaN	NaN	NaN	NaN	NaN	0.0	1.0	0.0
57566	NaN	NaN	NaN	NaN	NaN	NaN	0.0	1.0	0.0

58145 rows × 9 columns

In [601]:

```
filtered_df = new_df[new_df["Arrest Flag_Y"] == 1.0]

# Now you can work with the filtered_df, for example:
print(filtered_df.groupby("Frisk Flag_Y").size())
```

```
Frisk Flag_Y
0.0    3837
1.0    2161
dtype: int64
```

```
In [602]: print(filtered_df.groupby("Subject Perceived Race").size())
```

```
Subject Perceived Race
-                        186
American Indian or Alaska Native    163
Asian                               190
Black or African American    1750
DUPLICATE                           4
Hispanic                          147
Multi-Racial                      81
Native Hawaiian or Other Pacific Islander    11
Other                             25
Unknown                          438
White                           2946
dtype: int64
```

```
In [603]: print(filtered_df.groupby("Officer Race").size())
```

```
Officer Race
American Indian/Alaska Native    41
Asian                          273
Black or African American    222
Hispanic or Latino            384
Nat Hawaiian/Oth Pac Islander    46
Not Specified                 267
Two or More Races            386
White                        4322
dtype: int64
```

```
In [604]: # Assuming new_df is your DataFrame
grouped_df = new_df[new_df["Arrest Flag_Y"] == 1.0].groupby(["Subject Perceived Gender", "Officer Gender"]).size()

# Now you can work with the grouped_df
grouped_df
```

Out[604]:

Subject Perceived Gender	Officer Gender	
-	F	1
	M	19
DUPLICATE	F	1
	M	3
Female	F	130
	M	1031
Gender Diverse (gender non-conforming and/or transgender)	F	1
	M	3
Male	F	524
	M	4186
Unable to Determine	F	2
	M	29
Unknown	M	11

dtype: int64

```
In [605]: # Now you can work with the filtered_df
filtered_df = new_df[(new_df["Arrest Flag_Y"] == 1.0) & (new_df["Subject Perceived Gender"] == "Male")].groupby(["Subject Perceived Race", "Subject Age Group"]).size()

print(filtered_df)
```

Subject Perceived Race	Subject Age Group	
-	-	7
	1 - 17	5
	18 - 25	30
	26 - 35	39
	36 - 45	29
	...	
White	18 - 25	355
	26 - 35	807
	36 - 45	555
	46 - 55	345
	56 and Above	161

Length: 62, dtype: int64

```
In [606]: Races = ["White", "Black or African American"]
filtered_df = new_df[(new_df["Arrest Flag_Y"] == 1.0) & (new_df["Subject Perceived Race"] == "Races")].groupby(["Frisk Flag_Y"])

Now you can work with the filtered_df
print(filtered_df)

Series([], dtype: int64)
```

In []:

In []:

In []: