# Count the unique values in 'violation'

print(ri.violation.value\_counts())

# Express the counts as proportions

print(ri.violation.value\_counts(normalize= True))

# Create a DataFrame of female drivers

female = ri[ri.driver\_gender =='F']

# Create a DataFrame of male drivers

male = ri[ri.driver\_gender == 'M']

# Compute the violations by female drivers (as proportions)

print(female.violation.value\_counts(normalize = True))

# Compute the violations by male drivers (as proportions)

print(male.violation.value\_counts(normalize = True))

# Create a DataFrame of female drivers stopped for speeding

female\_and\_speeding = ri[(ri.driver\_gender == 'F') & (ri.violation == 'Speeding')]

# Create a DataFrame of male drivers stopped for speeding

male\_and\_speeding = ri[(ri.driver\_gender =='M') & (ri.violation == 'Speeding')]

# Compute the stop outcomes for female drivers (as proportions)

print(female\_and\_speeding.stop\_outcome.value\_counts( normalize = True))

# Compute the stop outcomes for male drivers (as proportions)

print(male\_and\_speeding.stop\_outcome.value\_counts( normalize = True))

# Check the data type of 'search\_conducted'

print(ri.search\_conducted.dtype)

# Calculate the search rate by counting the values

print(ri.search\_conducted.value\_counts(normalize = True))

# Calculate the search rate by taking the mean

print(ri.search\_conducted.mean())

# Calculate the search rate for both groups simultaneously

print(ri.groupby(ri.driver\_gender).search\_conducted.mean())

# Calculate the search rate for each combination of gender and violation

print(ri.groupby(['driver\_gender', 'violation']).search\_conducted.mean())