Exploratory Data Analysis on Seattle Terry Stops

This is a continuation of the project we started by preprocessing. As stated in the first part of this project, the "Preprocess file", the objectives of this project are:

- Determine if there is a racial disparity in the Seattle Terry Stops
- Do the differences in races between the officer and the subject play a role in frisks arrests?
- Determine the most common outcome of the Seattle Terry Stops and what it means
- Develop a model that can accurately predict the likelihood of an arrest occurring during a Terry Stop

In this part we will address the first three objectives and the fourth one will be covered in the third part of this project which is the modeling part.

Loading the cleaned data

In [1]:
```
# Importing the relevant libraries for EDA
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.ticker as mtick
%matplotlib inline
```

In [2]:
```
df = pd.read_csv('data/clean_Terry_stops_data.csv')

df.head()
```

Out[2]:

	subject_age_group	subject_id	go_sc_num	terry_stop_id	stop_resolution	weapon_type
0	36_45	unassigned	20160000398323	208373	Offense Report	NaN
1	18_25	unassigned	20180000227180	559146	Citation / Infraction	NaN
2	18_25	unassigned	20180000410091	498246	Offense Report	NaN
3	Unknown	unassigned	20160000001637	146742	Field Contact	NaN
4	46_55	unassigned	20150000006037	104477	Field Contact	NaN

5 rows × 28 columns

Yearly Terry Stops pattern covering the time period in the dataset

Let us look at the pattern of the Terry stops in Seattle for general overview.

In [3]:
```
# Creating a new DataFrame with the year column and a count column
year_df = pd.DataFrame(df['incident_year'])
year_df['count'] = 1

# Group by 'incident_year' and sum the 'count' column
year_graph = year_df.groupby('incident_year', as_index=False)['count']

# Plot the number of Terry Stops by year
plt.figure(figsize=(15,10))
plt.plot(year_graph['incident_year'], year_graph['count'], color='red
plt.title('Terry Stops by Year', fontsize=20)
plt.xlabel('Year', fontsize=14)
plt.ylabel('Number of Terry Stops', fontsize=14)
plt.grid(True)  # Enable grid for better readability

plt.show()
```

The spike in Terry Stops in Seattle in 2018 was a direct response to the surge in gun violence in the city. The Seattle Police Department's decision to increase stop-and-frisk activities was part of a broader strategy to address public safety concerns and reduce violent crime.

The drastic fall in Terry Stops in Seattle from 2019 to 2021 can be attributed to a combination of factors, including increased scrutiny and accountability due to police reform measures, the impact of protests and calls for reform (following the George Floyd incident), changes in crime trends, legal and community challenges, and operational shifts influenced by the pandemic. These factors collectively contributed to a reduction in the number of Terry Stops conducted by law enforcement during this period.

While the drop in Terry stop instances between 2023 and 2024 could reflect successful reforms, shifts in policing strategy, or changes in crime trends, among other factors. Understanding the specific reasons for the decline is crucial for evaluating its significance and ensuring that policing practices align with broader goals of fairness and effectiveness.

Relationship between Terry Stops and Subject's Race

Here we will try to see if there is any relationship between the Terry Stops and the Subject's Race.

In [4]:
```
# Let us first check the stop resolutions
df.stop_resolution.value_counts()
```

Out[4]:
```
stop_resolution
Field Contact               29173
Offense Report              15555
Arrest                      14592
Referred for Prosecution      717
Citation / Infraction         215
Name: count, dtype: int64
```

Stop Data

We are trying to see if race has any role to play in the Terry Stops. To do that let us calculate what the stop ratios are for each race in the dataset in contrast to their respective population.

Race Population Percentage (of total)

White 467,390 = 0.63

Asian 123,703 = 0.17

Two or more races 69,030 = 0.09

Black or African American 49,534 = 0.07

Other race 19,235 = 0.02

Native American 4,111 = 0.005

Native Hawaiian or Pacific Islander 1,600 = 0.002 (This race is aggregated with the 'Other race' for better calculation which comes to 0.022)

This population data is available at https://worldpopulationreview.com/us-cities/washington/seattle (https://worldpopulationreview.com/us-cities/washington/seattle)

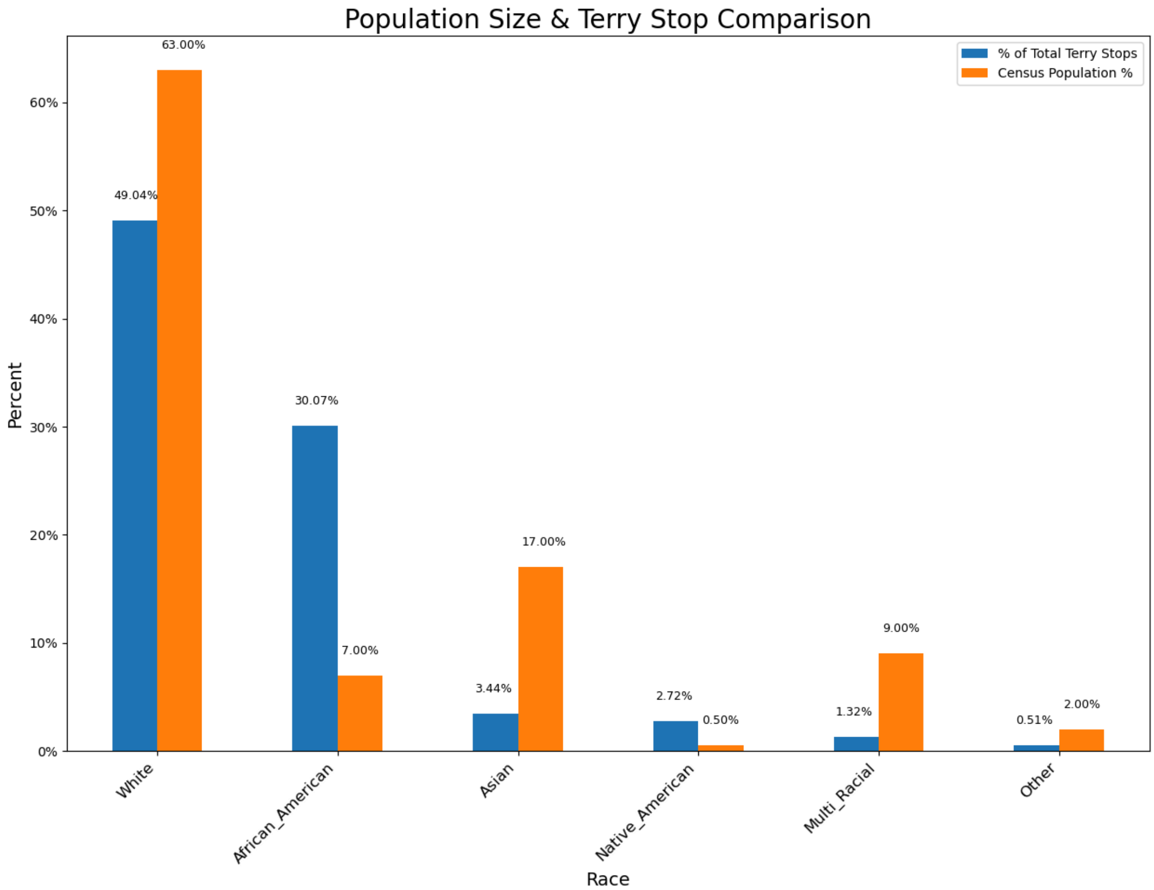
In [5]:
```
# Race count
df.subject_perceived_race.value_counts()
```

Out[5]:
```
subject_perceived_race
White                        29548
Black or African American    18119
Not Specified                 6108
Asian                         2070
Hispanic                      1666
Native American               1638
Multi-Racial                   796
Other                          307
Name: count, dtype: int64
```

In [6]:
```
# Calculating the stop ratios for each race in the dataset
df['subject_perceived_race'].value_counts(normalize=True)
```

Out[6]:
```
subject_perceived_race
White                        0.490407
Black or African American    0.300720
Not Specified                0.101374
Asian                        0.034356
Hispanic                     0.027651
Native American              0.027186
Multi-Racial                 0.013211
Other                        0.005095
Name: proportion, dtype: float64
```

In [7]:
```
# Calculating the percentage of each race in the dataset and assigning
races = ['White', 'African_American', 'Asian',
         'Native_American', 'Multi_Racial', 'Other']

stop_percents = [0.490407, 0.300720, 0.034356, 0.027186, 0.013211, 0.0
pop_percents = [0.63, 0.07, 0.17, 0.005, 0.09, 0.02] # This data is ob

race_stop_ratios = pd.DataFrame({
    'race': races,
    'stop_percent': stop_percents,
    'population_percent': pop_percents}).set_index(['race'])

# plotting terry stop and census % data by race for comparison
ax = race_stop_ratios.plot(kind = 'bar', figsize = (15,10))

for i, v in enumerate(race_stop_ratios.stop_percent):
    ax.text(i-.24, v+.02, '{:.2f}%'.format(100*v),fontsize = 9 )

for i, v in enumerate(race_stop_ratios.population_percent):
    ax.text(i+.02, v+.02, '{:.2f}%'.format(100*v),fontsize = 9 )

plt.title('Population Size & Terry Stop Comparison', fontsize = 20)
plt.xlabel('Race', fontsize = 14)
plt.ylabel('Percent', fontsize = 14)
plt.legend(loc = 'best', labels = ['% of Total Terry Stops', 'Census P
plt.xticks(rotation = 45, ha='right', fontsize = 12)
ax.yaxis.set_major_formatter(mtick.PercentFormatter(xmax=1, decimals=N
plt.show()
```



In [8]:
```
# Checking the stop ratio vs the population ratio of each race
race_stop_ratios
```

Out[8]:

race	stop_percent	population_percent
White	0.490407	0.630
African_American	0.300720	0.070
Asian	0.034356	0.170
Native_American	0.027186	0.005
Multi_Racial	0.013211	0.090
Other	0.005095	0.020

Ok so we see here that the highest stop ratio is for White with 49% and African American follows second with 30%. Even though the White is the dominant race in Seattle (63%), the proportion of the stop ratio vs the population is ok as is with all the other races except the Native Americans and the African Americans. With both these races the result is worrisome. Even though they are only 0.005% and 7% of the population respectively, the proportion of the stop ratio is 2.7% and 30%. This makes these races the most affected races by the Seattle Terry Stop.

Frisk Data

This requires further investigation of the dataset. Let us do the frisk flag test to see what happens.

In [9]:
```
# Let create a dataframe called frisked to see the racial distributior
frisked = df.copy()
frisked = frisked[frisked['frisk_flag'] == 'Y']

frisked.subject_perceived_race.value_counts()
```

Out[9]:
```
subject_perceived_race
White                        6100
Black or African American    5142
Not Specified                1566
Asian                         572
Hispanic                      435
Native American               385
Multi-Racial                  181
Other                          74
Name: count, dtype: int64
```

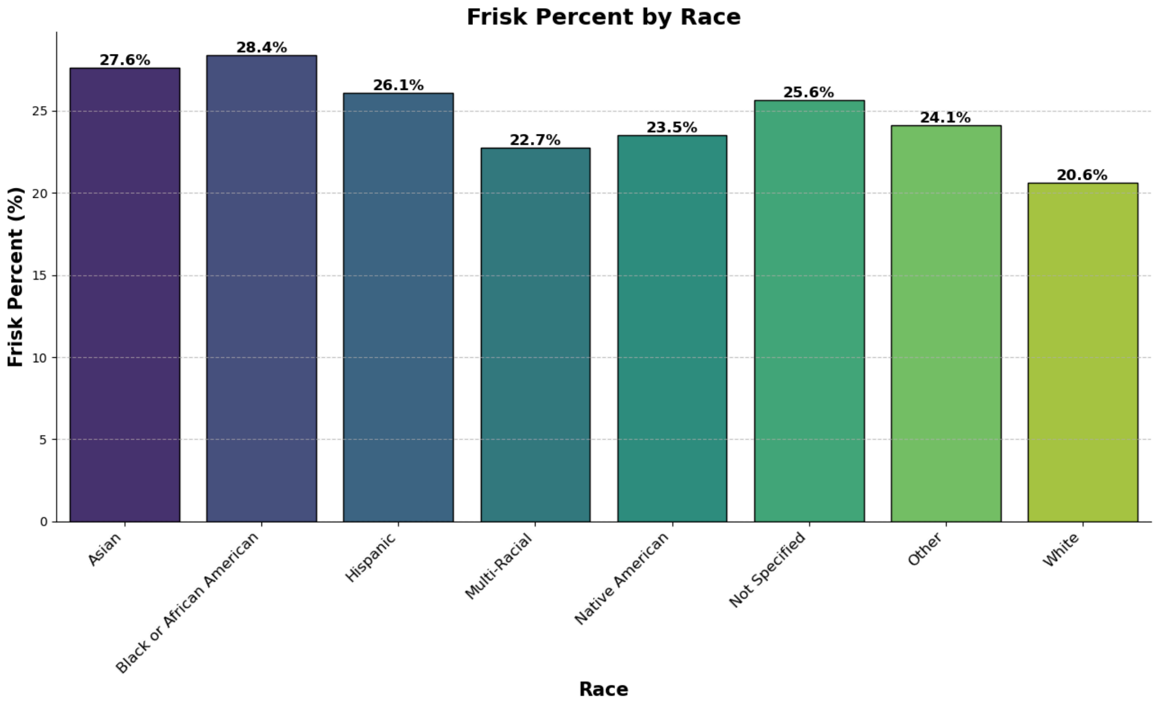
In [10]:
```
# Calculate the total number of Terry Stops and the number of frisk st
# Assume 'race' is the column indicating race and 'frisk' indicates it

total_stops = df.groupby('subject_perceived_race')['terry_stop_id'].co
frisk_stops = df[df['frisk_flag'] == 'Y'].groupby('subject_perceived_r

# Merge these two Series into a DataFrame for easier calculation
stop_data = pd.DataFrame({
    'total_stops': total_stops,
    'frisk_stops': frisk_stops})

# Calculate the frisk percentage
stop_data['frisk_percent'] = (stop_data['frisk_stops'] / stop_data['to
stop_data = stop_data.reset_index()
stop_data
```

Out[10]:

	subject_perceived_race	total_stops	frisk_stops	frisk_percent
0	Asian	2070	572	27.632850
1	Black or African American	18119	5142	28.379050
2	Hispanic	1666	435	26.110444
3	Multi-Racial	796	181	22.738693
4	Native American	1638	385	23.504274
5	Not Specified	6108	1566	25.638507
6	Other	307	74	24.104235
7	White	29548	6100	20.644375

In [11]:
```
# Plotting stop data
fig = plt.figure(figsize=(13, 8))

# Use Seaborn to create the bar plot
ax = sns.barplot(data=stop_data, x='subject_perceived_race', y='frisk_

# Annotate bars with their values
for p in ax.patches:
    height = p.get_height()
    ax.annotate(f'{height:.1f}%',
                (p.get_x() + p.get_width() / 2., height),
                ha='center', va='bottom',
                fontsize=12, fontweight='bold', color='black')

# Set labels and title with enhanced styling
ax.set_xlabel('Race', fontsize=15, fontweight='bold')
ax.set_ylabel('Frisk Percent (%)', fontsize=15, fontweight='bold')
ax.set_title('Frisk Percent by Race', fontsize=18, fontweight='bold')

# Customize x-axis labels
ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha='right', font

# Customize gridlines and remove spines
ax.grid(axis='y', linestyle='--', alpha=0.7)
sns.despine()

# Display the plot
plt.tight_layout()
plt.show()
```

```
/var/folders/m7/31mld3hn46s_y1f05nhxt1_c0000gn/T/ipykernel_20007/302
6284671.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set `legend
=False` for the same effect.

  ax = sns.barplot(data=stop_data, x='subject_perceived_race', y='fr
isk_percent', palette='viridis', edgecolor='black')
/var/folders/m7/31mld3hn46s_y1f05nhxt1_c0000gn/T/ipykernel_20007/302
6284671.py:21: UserWarning: set_ticklabels() should only be used wit
h a fixed number of ticks, i.e. after set_ticks() or using a FixedLo
cator.
  ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha='right',
fontsize=12)
```



As we can see it here as well the level of frisk instances between white and African American is almost parallel. Though the White race out numbers the African American race by a huge margin, the frisk instances are almost equal which can be an indication of the fact that there is a tendency towards racial bias. The higher frequency of frisk instances among the African American community during Terry stops in Seattle may stem from several interconnected factors:

- Historical and Systemic Bias: Long-standing racial biases and historical inequalities can influence policing practices, leading to disproportionate scrutiny of Black individuals.
- Policing Practices: Law enforcement may focus more on neighborhoods with higher crime rates, which can disproportionately affect racial minorities if those areas have higher minority populations.
- Socioeconomic Factors: Economic disparities, which often correlate with race, can result in more frequent police encounters in economically disadvantaged areas.
- Perceptions and Data: Biases in crime perception and the ways data are recorded and interpreted can affect how often stops and frisks are conducted among different racial groups.

Arrest Data

Let us further check what happens with the arrest instances.

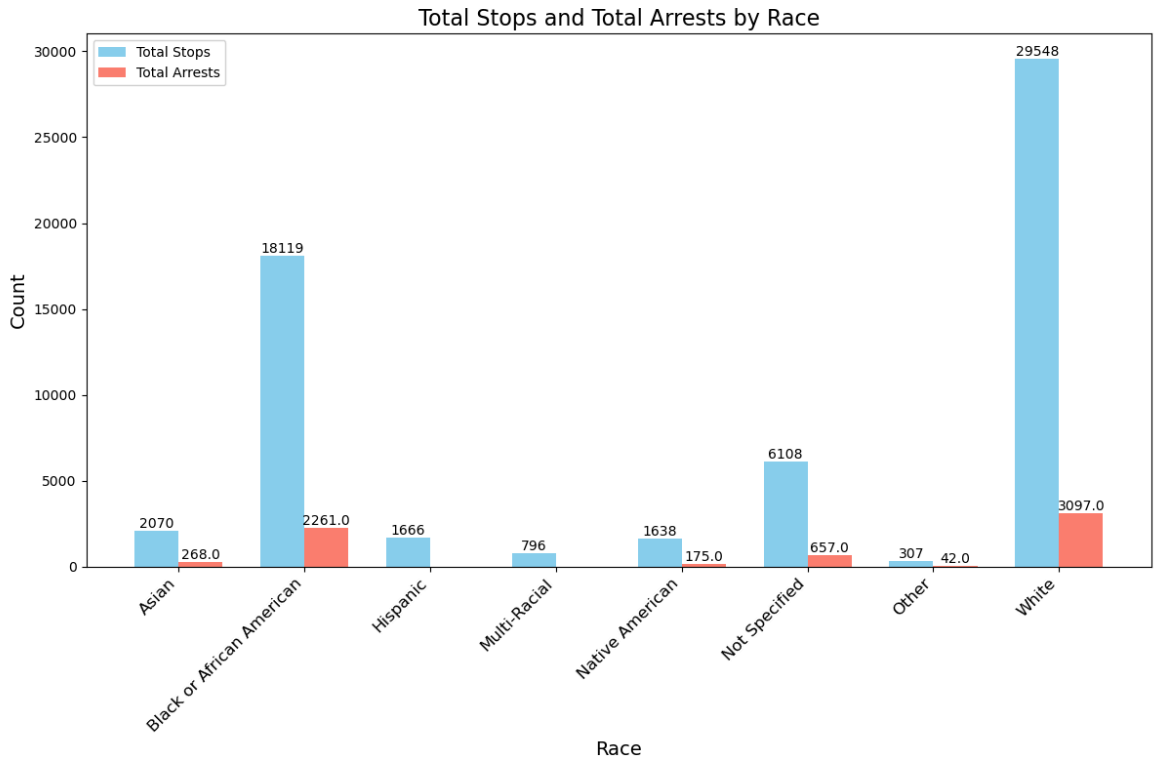
In [12]:
```
# Relationship between Terry Stops and Arrests per race
total_arrests = df[df['arrest_flag'] == 'Y'].groupby('subject_perceive
total_stops = df.groupby('subject_perceived_race')['terry_stop_id'].co
arrest_data = pd.DataFrame({
    'total_stops': total_stops,
    'total_arrests': total_arrests})
arrest_data = arrest_data.reset_index()

# arrests = df[['arrest_flag', 'frisk_flag', 'same_race']]

# print(arrests.shape)
# arrests.head()

arrest_data
```

Out[12]:

	subject_perceived_race	total_stops	total_arrests
0	Asian	2070	268.0
1	Black or African American	18119	2261.0
2	Hispanic	1666	NaN
3	Multi-Racial	796	NaN
4	Native American	1638	175.0
5	Not Specified	6108	657.0
6	Other	307	42.0
7	White	29548	3097.0

In [13]:
```
# Plotting the relationship between Terry Stops and Arrests per race
# Define data
races = arrest_data['subject_perceived_race']
total_stops = arrest_data['total_stops']
total_arrests = arrest_data['total_arrests']

# Define the width of each bar and positions
bar_width = 0.35
index = np.arange(len(races))

# Create a figure and axis
fig, ax = plt.subplots(figsize=(12, 8))

# Plot bars for total stops and total arrests
bars1 = ax.bar(index, total_stops, bar_width, label='Total Stops', co
bars2 = ax.bar(index + bar_width, total_arrests, bar_width, label='Tot

# Add labels, title, and legend
ax.set_xlabel('Race', fontsize=14)
ax.set_ylabel('Count', fontsize=14)
ax.set_title('Total Stops and Total Arrests by Race', fontsize=16)
ax.set_xticks(index + bar_width / 2)
ax.set_xticklabels(races, rotation=45, ha='right', fontsize=12)
ax.legend()

# Add value labels on bars
for bars in [bars1, bars2]:
    for bar in bars:
        height = bar.get_height()
        ax.text(
            bar.get_x() + bar.get_width() / 2.0, height,
            f'{height}',
            ha='center', va='bottom', fontsize=10
        )

# Adjust layout
plt.tight_layout()
plt.show()
```

```
posx and posy should be finite values
posx and posy should be finite values
posx and posy should be finite values
posx and posy should be finite values
```



This as well suggests the same fact. All the other races have almost 10% arrest instances out of the total stops they get while the African American race gets about 12% of arrest instances out of the total stops.

Racial Relationship Data

Ok now we'll make 2 dataframes: One where officer and subject are of the same race and another where they are not.

In [14]:
```
# Making datasets with the same officer and subject race and different
arrests = df[['arrest_flag', 'frisk_flag', 'same_race']]

same_race = arrests[arrests['same_race'] == 'Y']
diff_race = arrests[arrests['same_race'] == 'N']

print(same_race.shape)
print(diff_race.shape)
```

```
(22989, 3)
(37263, 3)
```

Now lets separate our data into categories we can compare: Same Race Arrests and Frisks & Different Race Arrests and Frisks

In [15]:

```
# Separating the data into categories
same_arrest = 0
same_frisk = 0
for i in range(len(same_race)):
    if same_race.arrest_flag.iloc[i] == 'Y':
        same_arrest += 1
    if same_race.frisk_flag.iloc[i] == 'Y':
        same_frisk += 1
print(f'Same Race: Arrests = {same_arrest}, Frisk Searches = {same_fr

diff_arrest = 0
diff_frisk = 0

for i in range(len(same_race)):
    if diff_race.arrest_flag.iloc[i] == 'Y':
        diff_arrest += 1
    if diff_race.frisk_flag.iloc[i] == 'Y':
        diff_frisk += 1
print(f'Different Race: Arrests = {diff_arrest}, Frisk Searches = {di
```

```
Same Race: Arrests = 2251, Frisk Searches = 4764
Different Race: Arrests = 2649, Frisk Searches = 5967
```

Since these are based on different sized sets, we'll do what we did with races and calculate the ratio of these values to their respective dataset sizes

In [16]:

```
# Calculating the ratio
arrest_same_ratio = same_arrest/len(same_race)
frisk_same_ratio = same_frisk/len(same_race)
arrest_diff_ratio = diff_arrest/len(diff_race)
frisk_diff_ratio = diff_frisk/len(diff_race)

# Creating a dataframe for same race arrests and same race frisks
keys = ['Same Race: Arrests', 'Same Race: Frisk Searches']
vals = [arrest_same_ratio, frisk_same_ratio]

same_race = {}

for key in keys:
    for val in vals:
        same_race[key] = val
        vals.remove(val)
        break


same_race_df = pd.DataFrame(same_race, index=[0])
same_race_df
```

Out[16]:

	Same Race: Arrests	Same Race: Frisk Searches
0	0.097916	0.20723

In [17]:
```
# Creating a dataframe for different race arrests and different race f
keys = ['Different Race: Arrests', 'Different Race: Frisk Searches']
vals = [arrest_diff_ratio, frisk_diff_ratio]

diff_race = {}

for key in keys:
    for val in vals:
        diff_race[key] = val
        vals.remove(val)
        break

diff_race_df = pd.DataFrame(diff_race, index=[0])
diff_race_df
```

Out[17]:

	Different Race: Arrests	Different Race: Frisk Searches
0	0.071089	0.160132

In [18]:
```
# Combining the two dataframes
race_relations = pd.concat([same_race_df, diff_race_df], axis=1)
race_relations
```

Out[18]:

	Same Race: Arrests	Same Race: Frisk Searches	Different Race: Arrests	Different Race: Frisk Searches
0	0.097916	0.20723	0.071089	0.160132

In [19]:
```
# Now let's plot this data
fig = plt.figure(figsize=(13,8))

ax = sns.barplot(race_relations)

ax.set_xlabel('Result of Stop', fontsize=15)
ax.set_ylabel('Arrested or Frisk Searched Percentage ', fontsize=15),
ax.set_title('Officer & Subject Relations: '+ 'Same Race v. Different
```

Out[19]:
```
Text(0.5, 1.0, 'Officer & Subject Relations: Same Race v. Different
Race')
```

There are more frisks and arrests among the same races, but fewer among different race. There could be a couple of reasons for this. One is that an officer whose race is different is more hesitant and doesn't want to risk the possibility of their actions being considered racist. Another explanation is that the officers in Seattle are assigned to beats where the local demographics match their own.

Frisk to Arrest Data

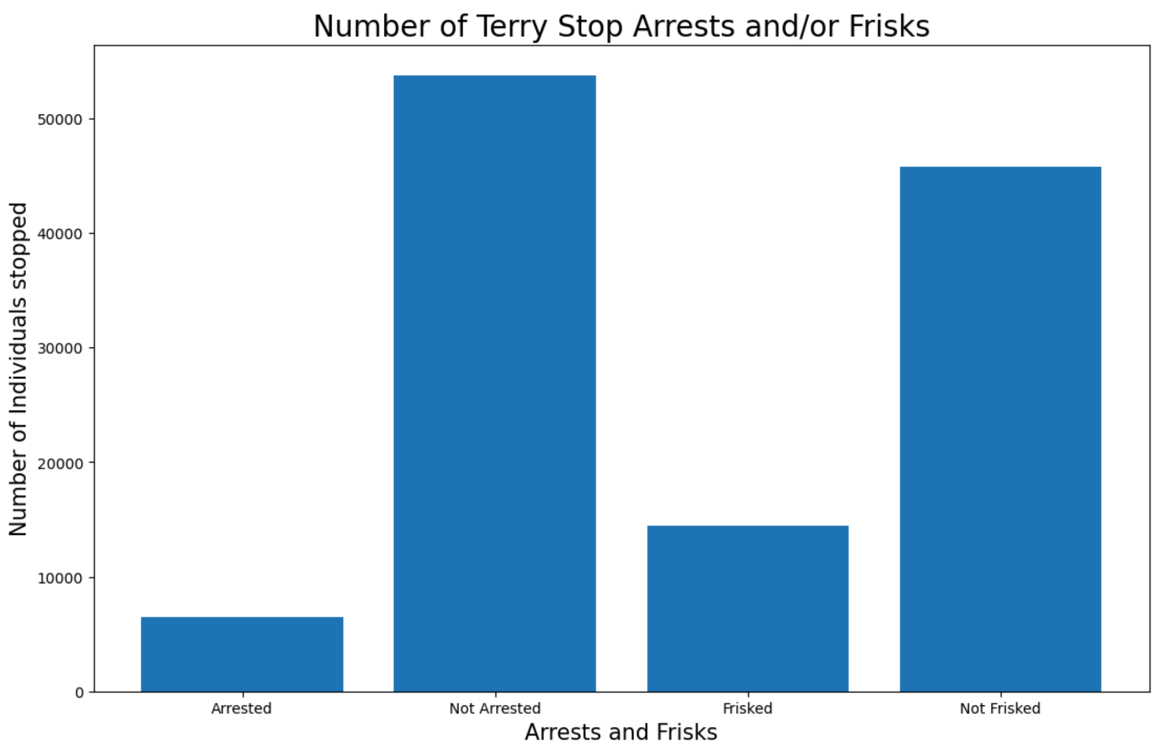
Let us do a comparison between the number of frisks and the number of arrests.

In [20]:
```
# Creating a subset of the dataset with the arrest and frisk features
arst_frsk = df[['arrest_flag', 'frisk_flag']]
print(arst_frsk.arrest_flag.value_counts())
print(arst_frsk.frisk_flag.value_counts())
```

```
arrest_flag
N    53752
Y     6500
Name: count, dtype: int64
frisk_flag
N    45797
Y    14455
Name: count, dtype: int64
```

In [21]:
```
# Plotting the comparison
arrest_yes = 6500
arrest_no = 53752
frisk_yes = 14455
frisk_no = 45797

x = ['Arrested', 'Not Arrested', 'Frisked', 'Not Frisked']
y = [arrest_yes, arrest_no, frisk_yes, frisk_no]

fig = plt.figure(figsize=(13,8))

ax = sns.barplot()
ax.bar(x, y)
ax.set_xlabel('Arrests and Frisks', fontsize=15)
ax.set_ylabel('Number of Individuals stopped', fontsize=15),
ax.set_title('Number of Terry Stop Arrests and/or Frisks', fontsize=2(
```

Out[21]:
```
Text(0.5, 1.0, 'Number of Terry Stop Arrests and/or Frisks')
```



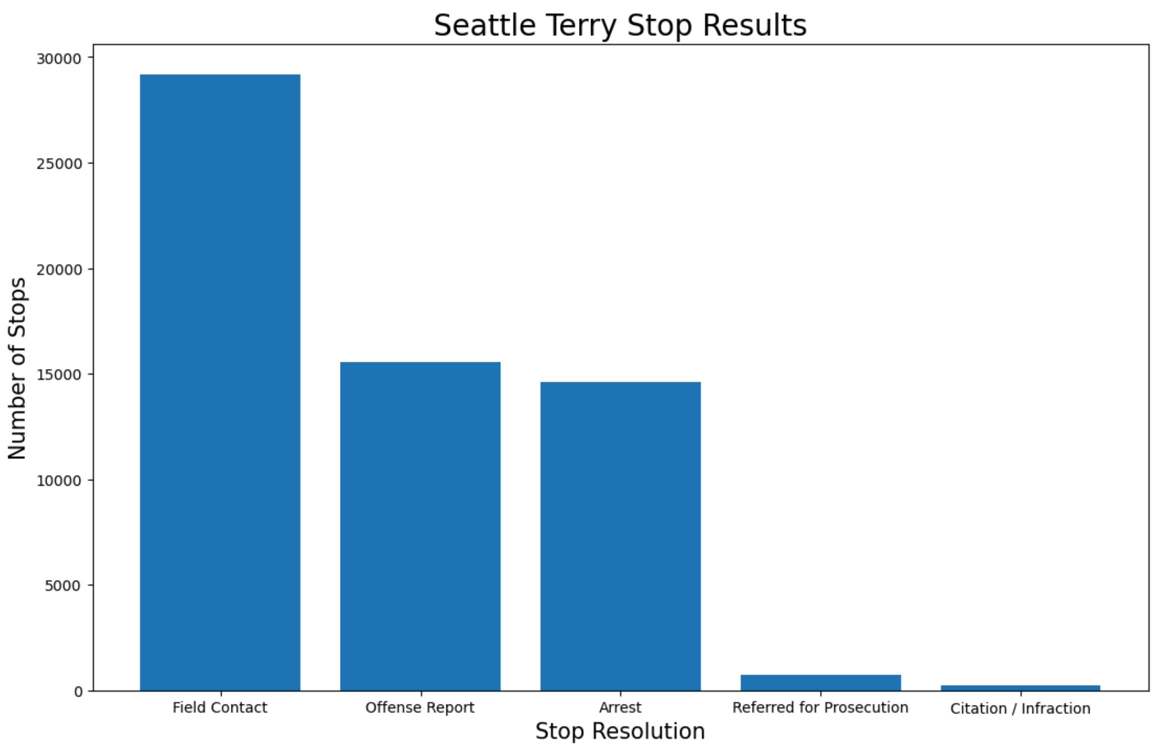
It is interesting to see that there is a huge gap between the number of people frisked and the number of people arrested. This could sound like a faulty call by the officers frisking people without enough information which could mean violation of people's 4th amendment right. On the other hand it could mean a cautious approach by officers, focusing on immediate safety concerns without escalating to arrests unless justified. However, it's important to continuously review these practices to ensure they align with legal standards and community expectations.

Terry Stop Common Outcome

Finally let us identify the common outcome of the stop resolutions.

In [22]:
```
# Creating a comparison between the different values of the stop resol
x = df['stop_resolution'].value_counts().index
y = df['stop_resolution'].value_counts()

# Plotting the comparison
fig = plt.figure(figsize=(13,8))

ax = sns.barplot()
ax.bar(x, y)
ax.set_xlabel('Stop Resolution', fontsize=15)
ax.set_ylabel('Number of Stops', fontsize=15)
ax.set_title('Seattle Terry Stop Results', fontsize=20)
```

Out[22]:
```
Text(0.5, 1.0, 'Seattle Terry Stop Results')
```



The most common outcome of the Seattle Terry Stop resolution happens to be "Field Contact". While field contacts are generally less intrusive than formal stops, a high frequency should be assessed to ensure it aligns with the goals of the resolution, maintains community trust, and is in line with fair policing practices. This result confirms that the numerous frisk instances are actually more of a preventive measures than violations of people's rights but extreme caution is necessary.

Conclusion

1. The fact that African Americans experience 30% of Terry stops despite being only 7% of the population suggests a significant racial disparity in policing. This indicates potential biases in police practices.
2. The data suggests that African Americans, while experiencing fewer overall Terry stops compared to White individuals, are subject to frisks more frequently when stopped. This disparity could indicate potential racial bias or differing police practices that result in more invasive procedures for African Americans.
3. A large gap between the number of people frisked and the number of arrests can have negative impacts on individuals, community relations, and policing effectiveness. It may lead to perceptions of unfairness, reduced trust in law enforcement, and potential legal and ethical concerns.
4. Officers exhibiting higher frisk and arrest frequencies towards individuals of their own race compared to those of a different race suggests potential issues related to racial bias or community-specific policing practices. This pattern might reflect familiarity bias, targeted enforcement, or implicit racial biases among officers.
5. The prevalence of "Field Contact" as the most common outcome of Terry stops in Seattle indicates that many stops do not result in formal legal action but rather involve brief interactions with individuals. This trend highlights the role of field contacts in proactive policing and community engagement, while also raising questions about effectiveness, resource allocation, and community perceptions.

Recommendations:

1. **Address Racial Bias:** Implement anti-bias training, revise stop policies for fairness, and increase transparency through audits and public reporting.
2. **Reform Frisk Practices:** Review frisk criteria to ensure they are justified and applied fairly, minimizing racial profiling.
3. **Evaluate Effectiveness:** Assess and adjust stop-and-frisk practices to improve outcomes and build community trust.
4. It is important to implement anti-bias training, review policing policies, and ensure equitable enforcement practices. Additionally, engaging with communities and improving data collection can help address and mitigate these disparities.
5. **Optimize Field Contacts:** Ensure field contacts are constructive and aligned with policing goals to enhance effectiveness and community engagement.

By analyzing and predicting Terry stops in Seattle, law enforcement officers and community members can make informed decisions about when and where to stop individuals, address the identified issues and concerns, and improve the overall effectiveness of their policing efforts.

In [ ]:

```

```