NYPD Civilian Complaints

This project contains data on 12,000 civilian complaints filed against New York City police officers. Interesting questions to consider include:

- Does the length that the complaint is open depend on ethnicity/age/gender?
- Are white-officer vs non-white complaintant cases more likely to go against the complainant?
- Are allegations more severe for cases in which the officer and complaintant are not the same ethnicity?
- Are the complaints of women more succesful than men (for the same allegations?)

There are a lot of questions that can be asked from this data, so be creative! You are not limited to the sample questions above.

Getting the Data

The data and its corresponding data dictionary is downloadable here. The data dictionary is in the project03 folder.

Note: you don't need to provide any information to obtain the data. Just agree to the terms of use and click "submit."

Cleaning and EDA

- Clean the data.
  - Certain fields have "missing" data that isn't labeled as missing. For example, there are fields with the value "Unknown." Do some exploration to find those values and convert them to null values.
  - You may also want to combine the date columns to create a `datetime` column for time-series exploration.
- Understand the data in ways relevant to your question using univariate and bivariate analysis of the data as well as aggregations.

Assessment of Missingness

- Assess the missingness per the requirements in `project03.ipynb`

Hypothesis Test / Permutation Test

Find a hypothesis test or permutation test to perform. You can use the questions at the top of the notebook for inspiration.

----------Start Here ----------

Summary of Findings

Introduction

The dataset is from ProPublica and records more than 12,000 civilian complaints filed against New York City police officers. The records span decades, from September 1985 to January 2020, and contains the information including complaint date, complaint type, age, ethnicity, and gender of both police officers and complainants.

ethnicity, and gender of both police officers and complainants.

The objective/research question of this project is to examine:Are the complaints made agaisnt the police officers more easy to be resolved for Men than to be resolved for Women?

We will answer this question by creating some exploratory data analysis (analyzing the statistics would help explain our questions) and conducting a few permutaiton tests for each category of complaint.

Based on our research setting, the most directly factors affecting the outcome (complain's disposition: Substantiated/ Unsubstantiated/ Exonerated) of a complaint is gender of complainants. On the other hand, we also recognize that there are other variables that might affect the substantiated rate, for example race of complainants, race of the police officer; however, we would only focused on the relationship between gender and complainant substantiated rate in this project.

We also regnozied that the result, inevitably, will come with certain bias as we ignore other genders beside female and male in the research question. We offer a way of thinking here but hopefully it will also provide more insights to the readers on the concerned matter.

I have labelled each finding with a tag in a pair of square brackets so that you can easily find the corresponding code :)

Cleaning and EDA

Brief discription about our data cleaning processes

we cleaned the following variables:

- complainant_age_incident: replace ages that are under 12 to np.nan to exclude negative ages and unrealistic ages. [CLEAN I]
- board_disposition: keep general information, we decided to aggregate small sub-categories into one large categoriesonly and only include Substantiated, Exonerated, Unsubstantiated for further analysis. This greatly simplify our analysis while bear no undesirable repurcussion on the final coutcome. [CLEAN II]
- shield_no: replace invaild shield_no (<=0) to np.nan [CLEAN III]

[This cleaning step comes with ethical concern.so I only implement once] Since our analysis mainly focus on the gender inequality and racial discrimination, and also because of the fact that we do not have enough sample representing the minority gender here. We have decided to remove all minority gender for the sake of our analysis.(I will implement this step right before hypothesis and I will not do it at the begining so that every minority gender get represented fully during the EDA.)

let's ananlyze the demographic of our sample first through some diagrams and graphs

As you can observe from this diagram, the race composition of the officers who are being accused actually suggests what is opposite to a commonly held perception that White officer are more probable of being accused from the alleged wrongdoing.[Diagram 1]

Secondly, let's look at the ethinities of people who file the complaints against the officers. The mojority of the complainant are black people here, which mean black people may be more susceptible to injustice or unfair treatment than any other races are. [Diagram 2]

In our third analysis, we find that male are more likely to submit a complaint than female are. We need to take this factor into consideration. Because this might potentially bias our analysis regarding genders since men women or other genders are not equally represented in the given sample. [Diagram3]

Then we also look at the "Age Distribution among Complainant Genders". The majority of people who filed a complaint are round 25 to 45 regardless of their genders. [Diagram4] [Diagram 5]

Here is the interesting finding [Chart1 and Diagram 6]

Also, it worth noticing that among all the disposition of the filed complaints, the proportions that complaints are disposed as "substantiated" between men and women differ in a magnitude that we could not neglect, which is about 5 percent of difference. That is being said, the complaint brought by women might less likely to be "substantiated" than men are. Aside from that, data also shows that proportion of women whose complaints are unsubstantiated is higher than the corresponding prortion of their counterpart.Proportion which offciers who are substantiated of wrongdoing got away with those charges is much highr for the women than for the man. All those differences in our statistics suggest that there might be discrimination against general women population when comes to the disposition of a filed complaint. For that, we will conduct a hypothesis test to further investigate this finding.

This is a fllow-up investigation on the issue mentioned aboved. But this time, we process the imformation at another level of granularity. Take an example to illustrate this, Say, what will the distribution of dispositions be like among the complaints that are about "Abuse of

Power"?

Assessment of Missingness

Handle NI (non-ignorable) missingness

Both my partner and I believe that the variable "gender" have non-ignorable missingness since sometimes it is hard distinguish one's gender just by his or her appearence or you can hardly acquire this information without confronting any subjective issues. There might be a ambiguity in deciding what real gender is or someone just do not want their gender to be known by other people, which leads to the missing value In short, the missingness here actually depends on the variable itself. If we can have some additional information such as the complainant's pronounce (provided by the complainant themself), we might be able to decisively figure out what's missing here.

Using permuation test to assess the missingness

case 1 (complainant_ethnicity vs mos_age_incident) (NO Dependence) (significance level 0.05)

We found that the missingness of the "complainant_ethnicity" is actually independent from variable "mos_age_incident". Graphically speaking, the distributions of missingness against variable "mos_age_incident" are quite similar, which suggests that those two distributions may be the same.(i.e they come from the same data generating process) Since the p-value is greater than the 0.05, our permutation test supports the null and suggests that two distribution are likely to be identical.

(complainant_ethnicity vs year_received) (Strong Dependence) (significance level 0.05)

The missingness of "complainant_ethnicity" is strongly dependent on the variable "year_received". Graphically speaking, the distributions of missingness against variable "year_received" are quite different. Since its p-value (0) is much smaller than 0.05, our permutation test rejects the null and suggests that two distribution are likely to be different.

Hypothesis Test (Permutation test)

Because we want to examine if gender plays an important role in disposition substantiated

rate (for the same allegation), we would conduct a permutation test and shuffle the complainant_gender to compare the percentage of complaint made by male and female that are substantiated for each allegation categories. Since there are four allegation categories (Abuse of Authority, Discourtesy, Force, Offensive Language), we would perform four permutation tests in total and compare their outcomes.

One of the appropriate test statistics to use is difference in proportion. More explicitly, the difference in male substantiated rate and women substantiated rate. And the significance level we choose for the permutation tests is 0.05.

Null hypothesis: In the population, disposition substantiated rate of women complainant and men complainant have the same distribution.

- p1=p2 or p1-p2=0
  - p1 stands for male complainant substantiated rate
  - p2 stands for female complainant substantiated rate

Alternative hypothesis: In the population, women usually have less complainant substantiated rate than men have. p1-p2 > 0

Results: p-values for Abuse of Authority, Discourtesy, Force and Offensive Language are 0.0, 0.008, 0.0015, and 0.478

Conclusion:

- Because the p-values for Abuse of Authority, Discourtesy and Force are all much smaller than the 0.05, we have sufficient evidence to reject the null, which states disposition substantiated rate of women complainant and men complainant have the same distribution.
- However, since the p-value for Offensive Language (0.478) is greater than 0.05, we are failed to reject the null hypothesis test for this category of complaint and agree that disposition substantiated rate of women complainant and men complainant have the same distribution.

Code

In [1]:
```
import matplotlib.pyplot as plt
import numpy as np
import os
import pandas as pd
import seaborn as sns
%matplotlib inline
%config InlineBackend.figure_format = 'retina'  # Higher resolution figures
```

In [2]:
```
# Before doing anyting, let's import the data
path=os.path.join("data","allegations_202007271729.csv")
allegations=pd.read_csv(path)
```

Cleaning and EDA

we cleaned the following variables:

- complainant_age_incident: replace ages that are under 12 to np.nan to exclude negative ages and unrealistic ages.
- board_disposition: keep general information, only including Substantiated, Exonerated, Unsubstantiated for further analysis.
- shield_no: replace invaild shield_no (<=0) to np.nan

EDA Graphs:

- Univariate Analysis:
  - distribution of officers ethnicity
  - distribution of ethnicity of complainant
  - distribution of complainant gender
  - Distribution of complainant age
- Bivariate Analysis:
  - age distribution among Complainant Genders
  - Disposition result over famle and male
- Bivariate Analysis & Interesting Aggregates
  - Stacked barplot of substantiated rate among different fado reasons (category of complaint)
  - Chart1

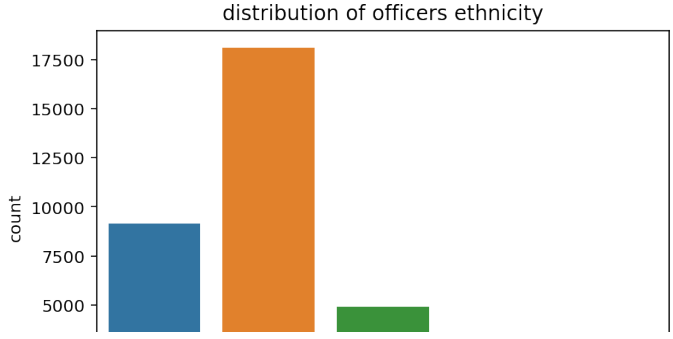
In [3]:
```
# CLEAN 1
# Some cleaning work are necessary
# remove the invalid age first. Remove all ages that are under 12
allegations['complainant_age_incident'] = allegations['complainant_age_incide
```

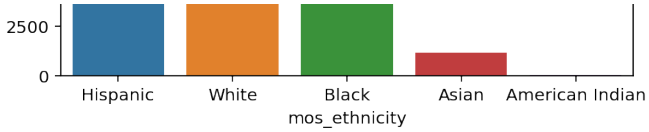
In [4]:
```
# CLEAN II
# cleaning disposition: keep general disposition -- Substantiated, Exonerated
allegations['board_disposition'] = allegations['board_disposition'].apply(lam
```

In [5]:
```
# ClEAN III
# replace the shield_no which is equal or less than 0 to np.nan
allegations['shield_no'] = allegations['shield_no'].apply(lambda x: x if x >
```

In [6]:
```
# Diagram 1 - Univariate Analysis
# ethnicity composition of complainted officers in NYPD

ax = sns.countplot(x = "mos_ethnicity", data = allegations)
ax.set_title('distribution of officers ethnicity')
```

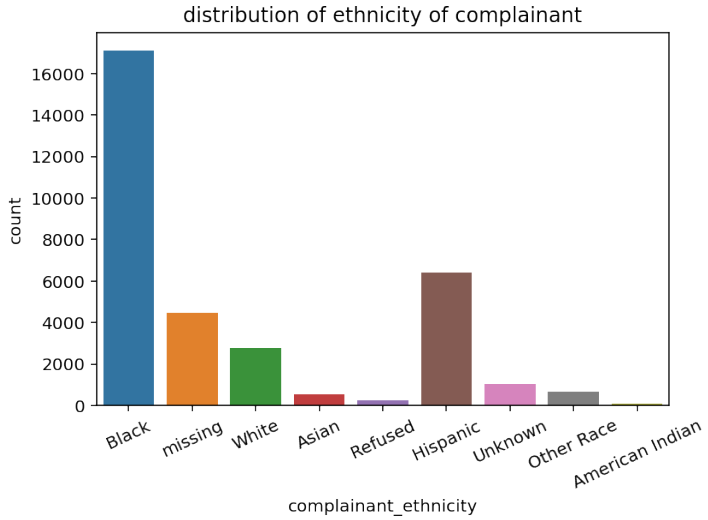
Out[6]:
```
Text(0.5, 1.0, 'distribution of officers ethnicity')
```





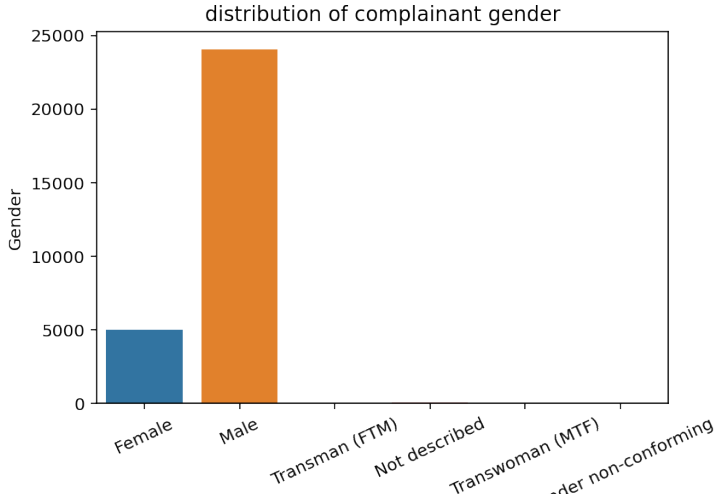
In [7]:
```
# Diagram 2 - Univariate Analysis
# the ethnicity of complainant
temp = allegations.copy()
temp['complainant_ethnicity'] = temp["complainant_ethnicity"].replace(np.nan,
ax = sns.countplot(x = "complainant_ethnicity", data = temp)
plt.xticks(rotation=25)
ax.set_title('distribution of ethnicity of complainant')
```

Out[7]: Text(0.5, 1.0, 'distribution of ethnicity of complainant')



In [8]:
```
# disgram 3 - Univariate Analysis
# Let's look at the gender distribution of complainants
ax = sns.countplot(x='complainant_gender', data=allegations);
ax.set_title('distribution of complainant gender')
plt.xticks(rotation=25)
ax.set_ylabel('Gender')
ax.set_xlabel('Counts')
```

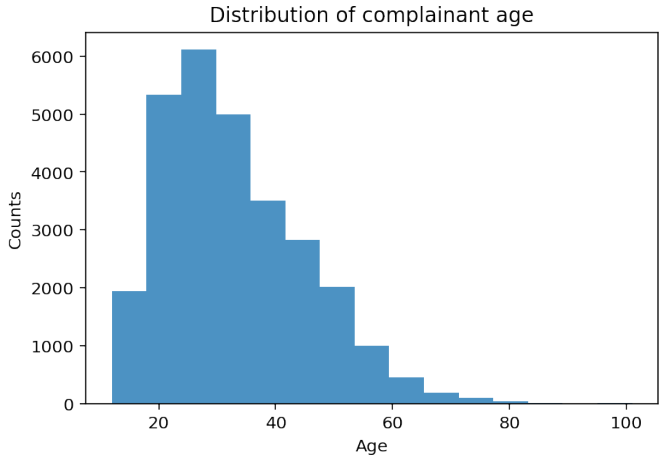
Out[8]: Text(0.5, 0, 'Counts')



Counts

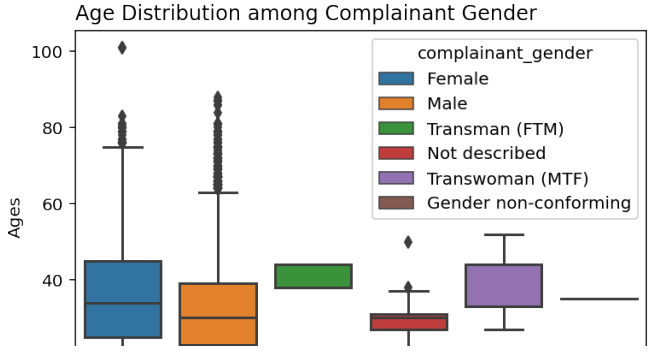
In [9]:
```
# disgram 4  - Univariate Analysis
# distribution of age in total
plt.hist(allegations['complainant_age_incident'], bins=15, alpha=0.8)
plt.title('Distribution of complainant age')
plt.xlabel('Age')
plt.ylabel('Counts')
```

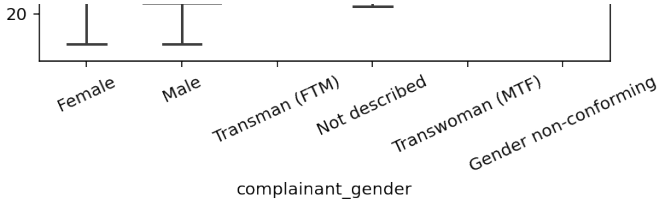
Out[9]: Text(0, 0.5, 'Counts')



In [10]:
```
# diagram 5 - Bivariate Analysis
# Age Distribution among Complainant Genders - Bivariate Analysis
ax = sns.boxplot(data = allegations,x = 'complainant_gender', y = 'complainan
                 hue = 'complainant_gender',dodge=False)
plt.xticks(rotation=25)
ax.set_title('Age Distribution among Complainant Gender', loc='left')
ax.set_ylabel('Ages')
```

Out[10]: Text(0, 0.5, 'Ages')





In [11]:
```
# Chart1
# get a table representing of the results regarding those filed complaints
# normalize the counts
# compare those data of men and women
data = allegations.pivot_table(
        values="unique_mos_id",
        index="complainant_gender",
        columns="board_disposition",
        aggfunc="count"
    )
cleaned_data=data.dropna()
cleaned_data.iloc[[0,1]].T.apply(lambda x:x/sum(x)).T
```

Out[11]:

board_disposition	Exonerated	Substantiated	Unsubstantiated
complainant_gender			
Female	0.281816	0.205537	0.512647
Male	0.273132	0.255674	0.471195

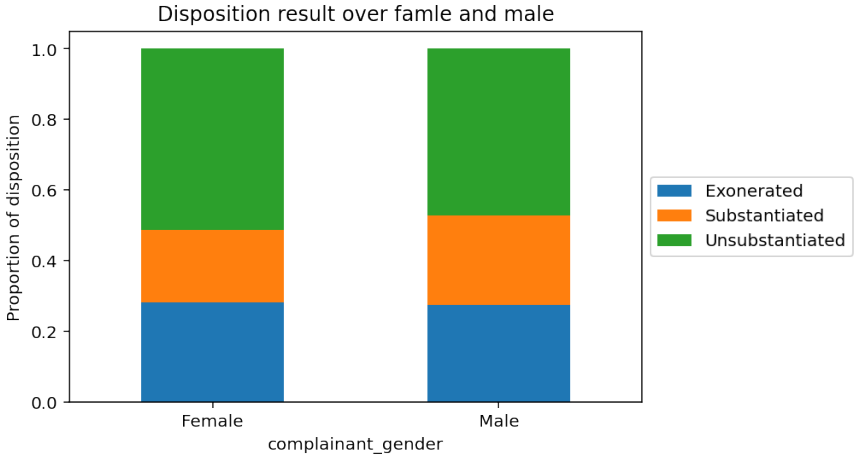
Graphs directly related to our hypothesis:

Because our research question (Are the complaints of women more succesful than men (for the same allegations?)) only focus on women and men, in our complainant_gender category, we would also just focus on the female and male to exam.

- Graphs to include:
  - Bar graph of count of female and male complains over final disposition
  - Bar graph of normalized female and male complains over final disposition

In [12]:
```
# only keep the rows there complainant_gender is female or male
gender = allegations[(allegations['complainant_gender'] == 'Female')
                     | (allegations['complainant_gender'] == 'Male')]
```

In [13]:
```
# generate the data
data = gender.pivot_table(
        values="unique_mos_id",
        index="board_disposition",
        columns="complainant_gender",
        aggfunc="count"
    )
data = (data/data.sum()).T

# generate plot
ax = data.plot(kind='bar', stacked=True, rot=0,
               title='Disposition result over famle and male')

# customize plot
ax.legend(('Exonerated','Substantiated','Unsubstantiated'), loc='center left'
ax.set_ylabel("Proportion of disposition");
```

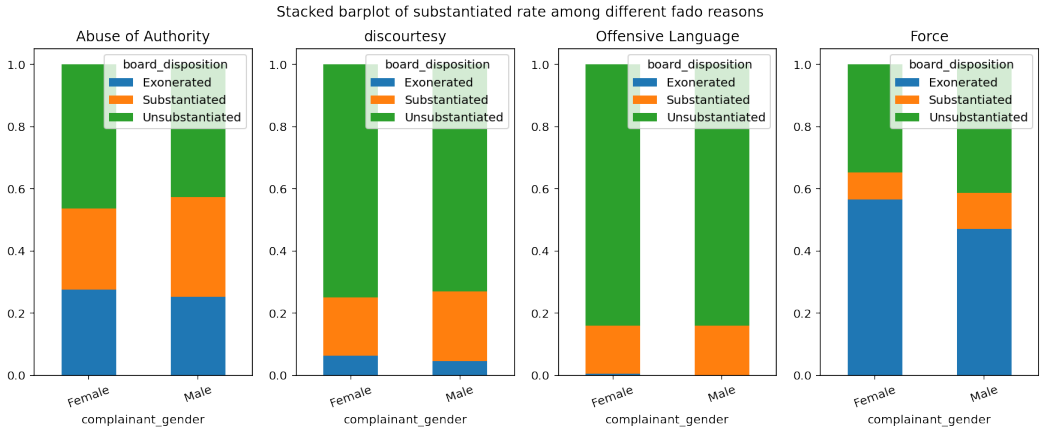


In [14]:

```
# diagram 6 - Bivariate Analysis & Interesting Aggregates
# multivariate analysis. does our statistics looks different at different lev
# generate the dataset among four different fado types(category of complaint)
abuse_data = gender[gender['fado_type'] == 'Abuse of Authority']
abuse = abuse_data.pivot_table(
        values="unique_mos_id",
        index="board_disposition",
        columns="complainant_gender",
        aggfunc="count"
    )
abuse = (abuse/abuse.sum()).T


discourtesy_data = gender[gender['fado_type'] == 'Discourtesy']
```

```
discourtesy_data = gender[gender['fado_type'] == 'Discourtesy']
discourtesy = discourtesy_data.pivot_table(
        values="unique_mos_id",
        index="board_disposition",
        columns="complainant_gender",
        aggfunc="count"
    )
discourtesy = (discourtesy/discourtesy.sum()).T
discourtesy

offensive_data = gender[gender['fado_type'] == 'Offensive Language']
offensive = offensive_data.pivot_table(
        values="unique_mos_id",
        index="board_disposition",
        columns="complainant_gender",
        aggfunc="count"
    )
offensive = (offensive/offensive.sum()).T
offensive

force_data= gender[gender['fado_type'] == 'Force']
force = force_data .pivot_table(
        values="unique_mos_id",
        index="board_disposition",
        columns="complainant_gender",
        aggfunc="count"
    )
force = (force/force.sum()).T
force

# generate the plot
fig, axes = plt.subplots(1, 4)
abuse.plot(ax = axes[0],kind='bar', stacked=True, rot=20,title = 'Abuse of Au
discourtesy.plot(ax = axes[1],kind='bar', stacked=True, rot=20, title = 'disc
offensive.plot(ax = axes[2],kind='bar', stacked=True, rot=20, title = 'Offens
force.plot(ax = axes[3],kind='bar', stacked=True, rot=20, title = 'Force')
fig.suptitle('Stacked barplot of substantiated rate among different fado reas
```

Out[14]: Text(0.5, 0.98, 'Stacked barplot of substantiated rate among different fado reasons')

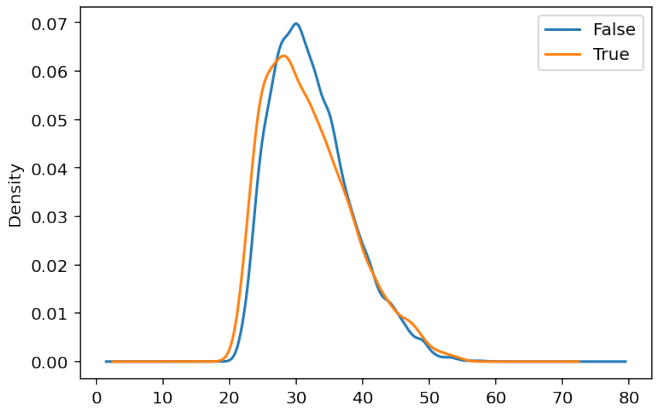


Assessment of Missingness

In [15]:
```
# permutation test [complainant_ethnicity vs. mos_age_incident]
# Basic plan: plot the distribution of mos_age_incident vs complainant_ethnic

null_vector=allegations["complainant_ethnicity"].isnull()
test_df=allegations.assign(is_null=null_vector)

test_df.groupby("is_null").mos_age_incident.plot(kind="kde",legend=True)
```

```
test_df.groupby("is_null").mos_age_incident.plot(kind="kde",legend=True)
```

Out[15]:
```
is_null
False    AxesSubplot(0.125,0.125;0.775x0.755)
True     AxesSubplot(0.125,0.125;0.775x0.755)
Name: mos_age_incident, dtype: object
```



In [16]:
```
# since two means are so close, use KS statistic to assess whether two distri
from scipy.stats import ks_2samp
```

In [17]:
```
group_A=test_df[test_df["is_null"]==True]["mos_age_incident"]
group_B=test_df[test_df["is_null"]==False]["mos_age_incident"]
ks_statistic=ks_2samp(group_A, group_B).statistic
ks_statistic
```

Out[17]:
```
0.06591802120369275
```

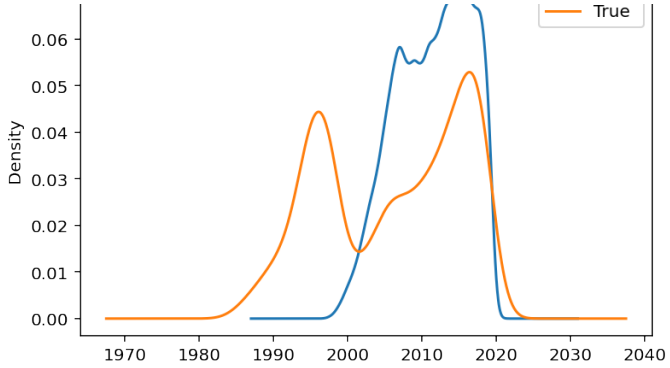
In [18]:
```
num_repetitions=1000
simulations2=[]
for i in range(num_repetitions):
    # first and foremost, we must shuffle the column of age
    shuffled_col=test_df["mos_age_incident"].sample(replace=False, frac=1).re
    # put them in a table
    shuffled_table=allegations.assign(**{"mos_age_incident":shuffled_col,"is_
    # compute a different statistic
    group_A_null=shuffled_table[shuffled_table["is_null"]==True]["mos_age_inc
    group_B_null=shuffled_table[shuffled_table["is_null"]==False]["mos_age_in
    ks_statistic_null=shuffled_table.groupby("is_null")["mos_age_incident"].m
    simulations2.append(ks_statistic_null)
```

In [19]:
```
# get the p_value
vectorized_simulations2=pd.Series(simulations2)
p_value=(ks_statistic<=vectorized_simulations2).mean()
p_value
```

Out[19]:
```
0.255
```

In [20]:
```
# permutation test [complainant_ethnicity vs. year_received]
# Basic plan: plot the distribution of mos_age_incident vs complainant_ethnic

null_vector=allegations["complainant_ethnicity"].isnull()
test_df=allegations.assign(is_null=null_vector)

test_df.groupby("is_null").year_received.plot(kind="kde",legend=True)
```

Out[20]:
```
is_null
False    AxesSubplot(0.125,0.125;0.775x0.755)
True     AxesSubplot(0.125,0.125;0.775x0.755)
Name: year_received, dtype: object
```



In [21]:
```
# find ks_statistics
group_A=test_df[test_df["is_null"]==True]["year_received"]
group_B=test_df[test_df["is_null"]==False]["year_received"]
ks_statistic=ks_2samp(group_A, group_B).statistic
ks_statistic
```

Out[21]: 0.3466206227037251

In [22]:
```
num_repetitions=1000
simulations2=[]
for i in range(num_repetitions):
    # first and foremost, we must shuffle the column of age
    shuffled_col=test_df["year_received"].sample(replace=False, frac=1).reset_
    # put them in a table
    shuffled_table=allegations.assign(**{"year_received":shuffled_col,"is_nul
    # compute a different statistic
    group_A_null=shuffled_table[shuffled_table["is_null"]==True]["year_receiv
    group_B_null=shuffled_table[shuffled_table["is_null"]==False]["year_recei
    ks_statistic_null=shuffled_table.groupby("is_null")["year_received"].mean
    simulations2.append(ks_statistic_null)
```

In [23]:
```
vectorized_simulations2=pd.Series(simulations2)
p_value=(ks_statistic<=vectorized_simulations2).mean()
p_value
```

Out[23]: 0.0

Hypothesis/Permutation Test

Four permutaion tests for each disposition:

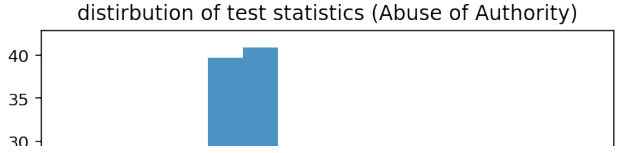
- Abuse of Authority
- Discourtesy
- Offensive Language
- Force

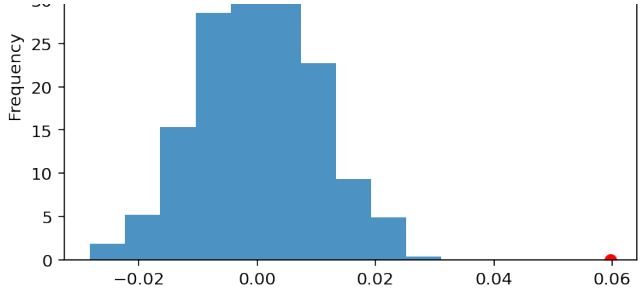
In [24]:

```
# allegation type: 'abuse'
display(abuse)
abuse_data = abuse_data[['fado_type','complainant_gender','board_disposition'
# find test statistics: difference in proportion (male - female)
size = abuse_data.shape[0]
obs = abuse.diff().iloc[-1][1]
stats = []
for _ in range(1000):
    # shuffle the gender
    shuffled_gender = (
        abuse_data['complainant_gender']
        .sample(replace=False, frac=1)
        .reset_index(drop=True)
    )


    # put them in a table
    shuffled = (
        abuse_data
        .assign(**{'shuffled_gender': shuffled_gender})
    )

    group_counts =(shuffled.pivot_table(
        values="fado_type",
        index="board_disposition",
        columns="shuffled_gender",
        aggfunc="count"
    ))
    group_means = (group_counts/group_counts.sum()).T
    difference = group_means.diff().iloc[-1][1]

    stats.append(difference)

p_value = (pd.Series(stats) >= obs).mean()
p_value_abuse = p_value
print('p-value: %f' % p_value)
pd.Series(stats).plot(kind='hist', density=True, alpha=0.8, title = 'distirbu
plt.scatter(obs, 0, color='red', s=40);
```

board_disposition	Exonerated	Substantiated	Unsubstantiated
complainant_gender			
Female	0.275875	0.261297	0.462828
Male	0.252910	0.320964	0.426126

```
p-value: 0.000000
```





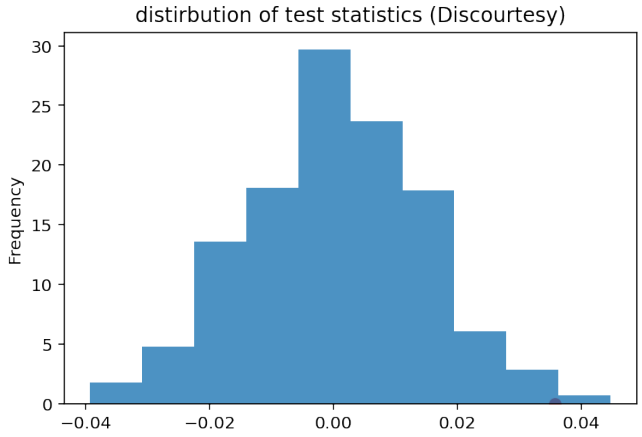
In [25]:

```
# allegation type: 'discourtesy'
display(discourtesy)
discourtesy_data = discourtesy_data[['fado_type','complainant_gender','board_
# find test statistics: difference in proportion (male - female)
size = discourtesy_data.shape[0]
obs = discourtesy.diff().iloc[-1][1]
stats = []
for _ in range(1000):
    # shuffle the gender
    shuffled_gender = (
        discourtesy_data['complainant_gender']
        .sample(replace=False, frac=1)
        .reset_index(drop=True)
    )

    # put them in a table
    shuffled = (
        discourtesy_data
        .assign(**{'shuffled_gender': shuffled_gender})
    )

    group_counts =(shuffled.pivot_table(
        values="fado_type",
        index="board_disposition",
        columns="shuffled_gender",
        aggfunc="count"
    ))
    group_means = (group_counts/group_counts.sum()).T
```

```
    difference = group_means.diff().iloc[-1][1]

    stats.append(difference)

p_value = (pd.Series(stats) >= obs).mean()
p_value_discourtesy = p_value
print('p-value: %f' % p_value)
pd.Series(stats).plot(kind='hist', density=True, alpha=0.8, title = 'distirbu
plt.scatter(obs, 0, color='red', s=40);
```

board_disposition	Exonerated	Substantiated	Unsubstantiated
complainant_gender			
Female	0.063851	0.187623	0.748527
Male	0.046144	0.223289	0.730567

```
p-value: 0.006000
```

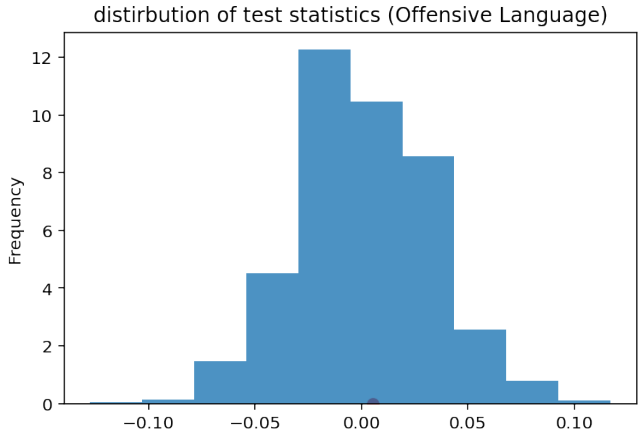


In [26]:

```
# allegation type: 'offensive'
display(offensive)
offensive_data = offensive_data[['fado_type','complainant_gender','board_disp
# find test statistics: difference in proportion (male - female)
size = offensive_data.shape[0]
obs = offensive.diff().iloc[-1][1]
stats = []
for _ in range(2000):
    # shuffle the gender
    shuffled_gender = (
        offensive_data['complainant_gender']
        .sample(replace=False, frac=1)
        .reset_index(drop=True)
    )

    # put them in a table
    shuffled = (
        offensive_data
        .assign(**{'shuffled_gender': shuffled_gender})
    )

    group_counts =(shuffled.pivot_table(
        values="fado_type",
        index="board_disposition",
        columns="shuffled_gender",
        aggfunc="count"
    ))
    group_means = (group_counts/group_counts.sum()).T
    difference = group_means.diff().iloc[-1][1]

    stats.append(difference)

p_value = (pd.Series(stats) >= obs).mean()
p_value_offensive = p_value
print('p-value: %f' % p_value)
pd.Series(stats).plot(kind='hist', density=True, alpha=0.8, title = 'distirbu
plt.scatter(obs, 0, color='red', s=40);
```

board_disposition	Exonerated	Substantiated	Unsubstantiated
complainant_gender			
Female	0.004695	0.154930	0.840376
Male	NaN	0.160183	0.839817

```
p-value: 0.463500
```



In [27]:

```
# allegation type: 'Force'
display(force)
force_data = force_data[['fado_type','complainant_gender','board_disposition'
# find test statistics: difference in proportion (male - female)
size = force_data.shape[0]
obs = force.diff().iloc[-1][1]
stats = []
for _ in range(2000):
    # shuffle the gender
    shuffled_gender = (
        force_data['complainant_gender']
        .sample(replace=False, frac=1)
        .reset_index(drop=True)
```

```
    )

    # put them in a table
    shuffled = (
        force_data
        .assign(**{'shuffled_gender': shuffled_gender})
    )

    group_counts =(shuffled.pivot_table(
        values="fado_type",
        index="board_disposition",
        columns="shuffled_gender",
        aggfunc="count"
    ))
    group_means = (group_counts/group_counts.sum()).T
    difference = group_means.diff().iloc[-1][1]

    stats.append(difference)

p_value = (pd.Series(stats) >= obs).mean()
p_value_force = p_value
print('p-value: %f' % p_value)
pd.Series(stats).plot(kind='hist', density=True, alpha=0.8, title = 'distirbu
plt.scatter(obs, 0, color='red', s=40);
```

board_disposition	Exonerated	Substantiated	Unsubstantiated
complainant_gender			
Female	0.565966	0.086998	0.347036
Male	0.471483	0.116142	0.412375

```
p-value: 0.002000
```

In [28]:
```
# generate data for dataframe
data = {'p_value':[p_value_abuse,p_value_discourtesy,p_value_force,p_value_of
```

```
summary_pvalue = pd.DataFrame(data)
summary_pvalue = summary_pvalue.assign(disposition = ['Abuse of Authority','D
                                                      'Force','Offensive Lang
# add column reject/ftr
summary_pvalue = summary_pvalue.assign(**{'Reject/FTR': summary_pvalue['p_val
                                       apply(lambda x: 'Reject' if x< 0.05
summary_pvalue
```

Out[28]:

disposition	p_value	Reject/FTR
Abuse of Authority	0.0000	Reject
Discourtesy	0.0060	Reject
Force	0.0020	Reject
Offensive Language	0.4635	FTR