ChallangeQ1

July 23, 2018

In [1]:
```
#I find this confusing
#You say to use the 2016 data and then link to 2017 data
#I'm guessing I should procees with 2017 dat but I wish I could ask
#This First.....................................
import pandas as pd
import numpy as np
pd.set_option('display.notebook_repr_html', False)
pd.set_option('display.max_columns', 8)
pd.set_option('display.max_rows', 8)
np.random.seed(1)
s = pd.Series(np.random.randn(100))
s
s[2]
s[[2, 5, 20]]
s[3:8]
s.tail()
s.index
s.values
s2 = pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd'])
s2
s = pd.Series([10, 0, 1, 1, 2, 3, 4, 5, 6, np.nan])
len(s)

sp500=pd.read_csv("sp500_2.csv",index_col='Symbol', usecols=[0, 2, 3, 7])
```

In [2]:
```
#This Second
#cc=pd.read_csv("cc.csv")#,index_col='UniqueComplaintId')
cc=pd.read_csv("cc193.csv")#,index_col='UniqueComplaintId')
```

In [3]:
```
#This Third
cc.head()
```

Out[3]:

	DateStamp	UniqueComplaintId	Close Year	Received Year	\
0	02/07/2017	70245	2009	2008	
1	02/07/2017	70244	2007	2006	
2	02/07/2017	70244	2007	2006	
3	02/07/2017	70244	2007	2006	
4	02/07/2017	70244	2007	2006	

```
            ...              Encounter Outcome  Reason For Initial Contact  \
0           ...           No Arrest or Summons                       Other
1           ...                         Arrest   Report-noise/disturbance
2           ...                         Arrest   Report-noise/disturbance
3           ...                         Arrest   Report-noise/disturbance
4           ...                         Arrest   Report-noise/disturbance

   Allegation FADO Type Allegation Description
0    Offensive Language    Physical disability
1    Abuse of Authority       Threat of arrest
2    Abuse of Authority   Question and/or stop
3    Abuse of Authority     Search (of person)
4    Abuse of Authority                  Frisk

[5 rows x 16 columns]
```

In [4]:
```
#Fourth
cd=cc.drop_duplicates(subset=['UniqueComplaintId'],keep='first')
```

In [5]:
```
#Fifth
cd.set_index('UniqueComplaintId')
cd.head(4)
```

Out[5]:
```
     DateStamp  UniqueComplaintId  Close Year  Received Year  \
0   02/07/2017              70245        2009           2008
1   02/07/2017              70244        2007           2006
12  02/07/2017              70243        2011           2011
13  02/07/2017              70242        2010           2009

                  ...                  Encounter Outcome  \
0                 ...               No Arrest or Summons
1                 ...                             Arrest
12                ...                            Summons
13                ...               No Arrest or Summons

    Reason For Initial Contact  Allegation FADO Type  \
0                        Other    Offensive Language
1     Report-noise/disturbance    Abuse of Authority
12      Other violation of VTL           Discourtesy
13                       Other    Abuse of Authority

              Allegation Description
0               Physical disability
1                  Threat of arrest
12                             Word
13  Premises entered and/or searched

[4 rows x 16 columns]
```

```
In [6]: numbercomplaint=[len(cd.loc[cd['Borough of Occurrence'] == 'Brooklyn'].index),len(cd.loc

In [7]: #This is the answer to the second question
        max(numbercomplaint)/sum(numbercomplaint)

Out[7]: 0.3421768512685413

In [151]: cd2016=cd.loc[cd['Incident Year'] == 2016]
          print(cd2016.head())

      DateStamp  UniqueComplaintId  Close Year  Received Year  \
135  02/07/2017              70198        2016           2016
244  02/07/2017              70164        2016           2016
247  02/07/2017              70161        2016           2016
291  02/07/2017              70147        2016           2016
365  02/07/2017              70126        2016           2016

           ...                Reason For Initial Contact  \
135        ...                     CV already in custody
244        ...                           Report-dispute
247        ...         Report-gun possession/shots fired
291        ...               Execution of search warrant
365        ...                     CV already in custody

     Allegation FADO Type            Allegation Description ComplaintDuration
135                 Force                   Physical force                  0
244    Abuse of Authority                 Threat of arrest                  0
247    Abuse of Authority                       Gun Drawn                   0
291    Abuse of Authority  Premises entered and/or searched                 0
365                 Force                   Physical force                  0

[5 rows x 17 columns]


In [152]: #Sixth
          cdbrook=cd2016.loc[cd2016['Borough of Occurrence'] == 'Brooklyn']

In [153]: #cdbrook
          len(cd2016.loc[cd2016['Borough of Occurrence'] == 'Brooklyn'].index)

Out[153]: 1102

In [154]: #number of rows is displayed in last row of above output
          brookrate=1102/2648771#denomenator is population of borough from wikipedia
          print(100000*brookrate)

41.604200589631944
```

```
In [155]: #Bronx rate
          #cdbronx=cd.loc[cd['Borough of Occurrence'] == 'Bronx']
          #print(cdbronx)
          len(cd2016.loc[cd2016['Borough of Occurrence'] == 'Bronx'].index)

Out[155]: 792

In [156]: bronxrate=792/1471160
          print(100000*bronxrate)

53.83506892520188


In [157]: #cdman=cd.loc[cd['Borough of Occurrence'] == 'Manhattan']
          #print(cdman)
          len(cd2016.loc[cd2016['Borough of Occurrence'] == 'Manhattan'].index)

Out[157]: 862

In [158]: manrate=862/1664727
          print(100000*manrate)

51.780261868762864


In [159]: #cdqueens=cd.loc[cd['Borough of Occurrence'] == 'Queens']
          #print(cdqueens)
          len(cd2016.loc[cd2016['Borough of Occurrence'] == 'Queens'].index)

Out[159]: 601

In [160]: queensrate=601/2358582
          print(100000*queensrate)

25.48141213661429


In [161]: #cdstaten=cd.loc[cd['Borough of Occurrence'] == 'Staten Island']
          #print(cdstaten)
          len(cd2016.loc[cd2016['Borough of Occurrence'] == 'Staten Island'].index)

Out[161]: 156

In [162]: statenrate=156/479458
          print(100000*statenrate)

32.53673940157428


In [163]: question2=max([statenrate,queensrate,manrate,bronxrate,queensrate])
          print([statenrate,queensrate,manrate,bronxrate,brookrate])
```

```
[0.0003253673940157428, 0.0002548141213661429, 0.0005178026186876286, 0.0005383506892520188, 0.0
```

```
In [164]: #This is the answer to the third question
          question2*100000
```

```
Out[164]: 53.83506892520188
```

```
In [21]: #This could take a few minutes
         cddiff=cd
         cddiff['ComplaintDuration'] =cddiff['Close Year']-cddiff['Received Year']
```

```
/home/nbuser/anaconda3_501/lib/python3.6/site-packages/ipykernel/__main__.py:3: SettingWithCopyW
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: http://pandas.pydata.org/pandas-docs/stable/indexing.html#
  app.launch_new_instance()
```

```
In [22]: cddiff
```

```
Out[22]:          DateStamp  UniqueComplaintId  Close Year  Received Year  \
         0       02/07/2017              70245        2009           2008
         1       02/07/2017              70244        2007           2006
         12      02/07/2017              70243        2011           2011
         13      02/07/2017              70242        2010           2009
         ...            ...                ...         ...            ...
         206707  02/07/2017                  4        2013           2013
         206710  02/07/2017                  3        2010           2010
         206711  02/07/2017                  2        2009           2009
         206713  02/07/2017                  1        2014           2014

                        ...                          Reason For Initial Contact  \
         0              ...                                               Other
         1              ...                         Report-noise/disturbance
         12             ...                           Other violation of VTL
         13             ...                                               Other
         ...            ...                                                 ...
         206707         ...                          Execution of search warrant
         206710         ...                                        Report-dispute
         206711         ...          PD suspected C/V of violation/crime - street
         206713         ...            PD suspected C/V of violation/crime - auto

                 Allegation FADO Type            Allegation Description  \
         0         Offensive Language               Physical disability
         1         Abuse of Authority                  Threat of arrest
         12               Discourtesy                              Word
         13        Abuse of Authority  Premises entered and/or searched
```

```
...                        ...                                ...
206707    Abuse of Authority  Premises entered and/or searched
206710          Discourtesy                              Word
206711    Abuse of Authority                          Question
206713    Abuse of Authority                    Vehicle search

       ComplaintDuration
0                      1
1                      1
12                     0
13                     1
...                  ...
206707                 0
206710                 0
206711                 0
206713                 0

[70245 rows x 17 columns]
```

In [23]:
```
#This is the answer to the fourth question
cddiff['ComplaintDuration'].mean()
```

Out[23]:
```
0.47268844757633993
```

In [24]:
```
print(3)
```

```
3
```

In [25]:
```
#len(cd.loc[cd['Borough of Occurrence'] == 'Staten Island'].index)
#pd.to_datetime('2005')
complaints2005=len(cd.loc[cd['Incident Year'] == 2005].index)
complaints2006=len(cd.loc[cd['Incident Year'] == 2006].index)
complaints2007=len(cd.loc[cd['Incident Year'] == 2007].index)
complaints2008=len(cd.loc[cd['Incident Year'] == 2008].index)
complaints2009=len(cd.loc[cd['Incident Year'] == 2009].index)
complaints2010=len(cd.loc[cd['Incident Year'] == 2010].index)
complaints2011=len(cd.loc[cd['Incident Year'] == 2011].index)
complaints2012=len(cd.loc[cd['Incident Year'] == 2012].index)
complaints2013=len(cd.loc[cd['Incident Year'] == 2013].index)
complaints2014=len(cd.loc[cd['Incident Year'] == 2014].index)
complaints2015=len(cd.loc[cd['Incident Year'] == 2015].index)
complaints2016=len(cd.loc[cd['Incident Year'] == 2016].index)
complaintsbyyear=[[2005,complaints2005],[2006,complaints2006],[2007,complaints2007],[20
```

In [26]:
```
complaintsbyyear
```

Out[26]:
```
[[2005, 3424],
 [2006, 7698],
```

```
          [2007, 7544],
          [2008, 7343],
          [2009, 7629],
          [2010, 6461],
          [2011, 6012],
          [2012, 5755],
          [2013, 5409],
          [2014, 4750],
          [2015, 4410],
          [2016, 3561]]
```

In [27]:
```
print(2)
```

```
2
```

In [28]:
```
AAAA=np.array(complaintsbyyear[(len(complaintsbyyear)-8):(len(complaintsbyyear))])
print(AAAA)
```

```
[[2009 7629]
 [2010 6461]
 [2011 6012]
 [2012 5755]
 [2013 5409]
 [2014 4750]
 [2015 4410]
 [2016 3561]]
```

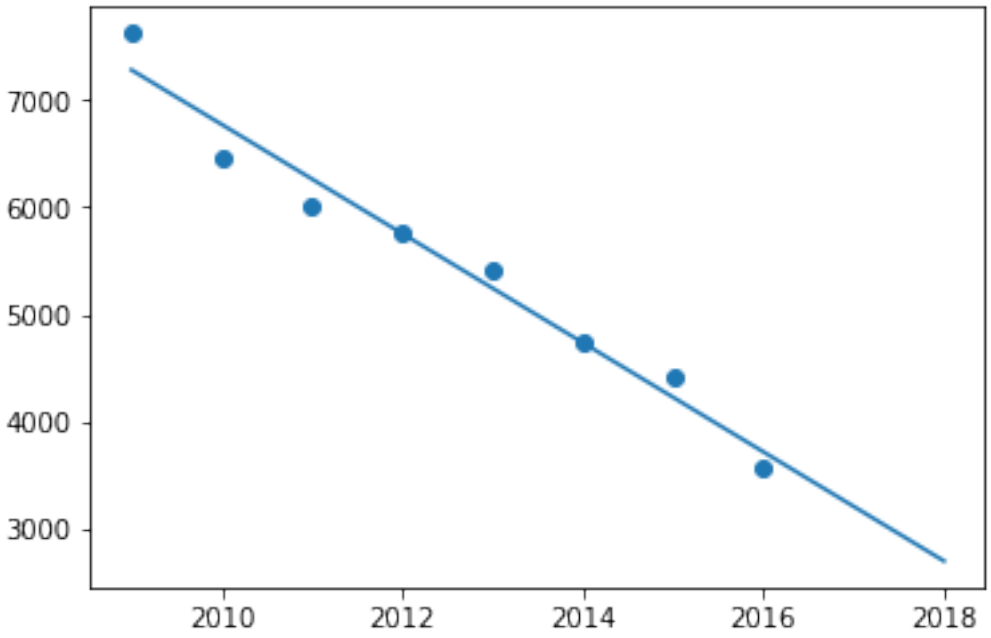
In [30]:
```
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
x = AAAA[:,0]
y = AAAA[:,1]
model = LinearRegression(fit_intercept=True)

model.fit(x[:, np.newaxis], y)

xfit = np.linspace(2009,2018,10)
yfit = model.predict(xfit[:, np.newaxis])

plt.scatter(x, y)
plt.plot(xfit, yfit)
print(model.coef_[0])
print(model.intercept_)
Mslope=model.coef_[0]
Bintercept=model.intercept_
```

```
-510.27380952380946
1032424.4166666665
```




```
In [31]: print(Mslope*2014+Bintercept)
```

```
4732.964285714203
```

```
In [32]: #This is the answer to question 5
         print(Mslope*2018+Bintercept)
```

```
2691.869047619053
```

```
In [33]: #IGNORE ME!
         #len(cd.loc[cd['Borough of Occurrence'] == 'Staten Island'].index)
         #pd.to_datetime('2005')
         complaints2005=cd['Incident Year'] == 2005].index
         complaints2006=len(cd.loc[cd['Incident Year'] == 2006].index)
         complaints2007=len(cd.loc[cd['Incident Year'] == 2007].index)
         complaints2008=len(cd.loc[cd['Incident Year'] == 2008].index)
         complaints2009=len(cd.loc[cd['Incident Year'] == 2009].index)
         complaints2010=len(cd.loc[cd['Incident Year'] == 2010].index)
         complaints2011=len(cd.loc[cd['Incident Year'] == 2011].index)
         complaints2012=len(cd.loc[cd['Incident Year'] == 2012].index)
         complaints2013=len(cd.loc[cd['Incident Year'] == 2013].index)
         complaints2014=len(cd.loc[cd['Incident Year'] == 2014].index)
         complaints2015=len(cd.loc[cd['Incident Year'] == 2015].index)
         complaints2016=len(cd.loc[cd['Incident Year'] == 2016].index)
```

```
        #Forgot about the stop and frisk condition
        complaints2005=len(cd.loc[cd['Incident Year'] == 2005].index)
        complaints2006=len(cd.loc[cd['Incident Year'] == 2006].index)
        complaints2007=len(cd.loc[cd['Incident Year'] == 2007].index)
        complaints2008=len(cd.loc[cd['Incident Year'] == 2008].index)
        complaints2009=len(cd.loc[cd['Incident Year'] == 2009].index)
        complaints2010=len(cd.loc[cd['Incident Year'] == 2010].index)
        complaints2011=len(cd.loc[cd['Incident Year'] == 2011].index)
        complaints2012=len(cd.loc[cd['Incident Year'] == 2012].index)
        complaints2013=len(cd.loc[cd['Incident Year'] == 2013].index)
        complaints2014=len(cd.loc[cd['Incident Year'] == 2014].index)
        complaints2015=len(cd.loc[cd['Incident Year'] == 2015].index)
        complaints2016=len(cd.loc[cd['Incident Year'] == 2016].index)
        complaintsbyyear=[[2005,complaints2005],[2006,complaints2006],[2007,complaints2007],[20
```

```
  File "<ipython-input-33-0a15437c6586>", line 4
    complaints2005=cd['Incident Year'] == 2005].index
                                               ^
SyntaxError: invalid syntax
```

```
In [34]: #Wait. I was supposed to only look at complaints involving frisking
         #print(np.bool(cd['Complaint Contains Stop & Frisk Allegations'][0])==True)
         frisked=cd.loc[cd['Complaint Contains Stop & Frisk Allegations']==True]
         complaints2005=len(frisked.loc[frisked['Incident Year'] == 2005].index)
         complaints2006=len(frisked.loc[frisked['Incident Year'] == 2006].index)
         complaints2007=len(frisked.loc[frisked['Incident Year'] == 2007].index)
         complaints2008=len(frisked.loc[frisked['Incident Year'] == 2008].index)
         complaints2009=len(frisked.loc[frisked['Incident Year'] == 2009].index)
         complaints2010=len(frisked.loc[frisked['Incident Year'] == 2010].index)
         complaints2011=len(frisked.loc[frisked['Incident Year'] == 2011].index)
         complaints2012=len(frisked.loc[frisked['Incident Year'] == 2012].index)
         complaints2013=len(frisked.loc[frisked['Incident Year'] == 2013].index)
         complaints2014=len(frisked.loc[frisked['Incident Year'] == 2014].index)
         complaints2015=len(frisked.loc[frisked['Incident Year'] == 2015].index)
         complaints2016=len(frisked.loc[frisked['Incident Year'] == 2016].index)
         complaintsbyyear=[[2005,complaints2005],[2006,complaints2006],[2007,complaints2007],[20
         #cd
         #complaints2005=cd.loc[cd['Incident Year'] == 2005 & np.bool(cd['Complaint Contains Sto
         #print(complaints2005)

In [35]: print(complaintsbyyear)

[[2005, 1157], [2006, 2446], [2007, 2566], [2008, 2265], [2009, 2264], [2010, 1889], [2011, 1679

In [36]: AAAA=np.array(complaintsbyyear[(len(complaintsbyyear)-10):(len(complaintsbyyear))])
         print(AAAA)
```

```
[[2007 2566]
 [2008 2265]
 [2009 2264]
 [2010 1889]
 [2011 1679]
 [2012 1492]
 [2013 1260]
 [2014  987]
 [2015  891]
 [2016  709]]
```

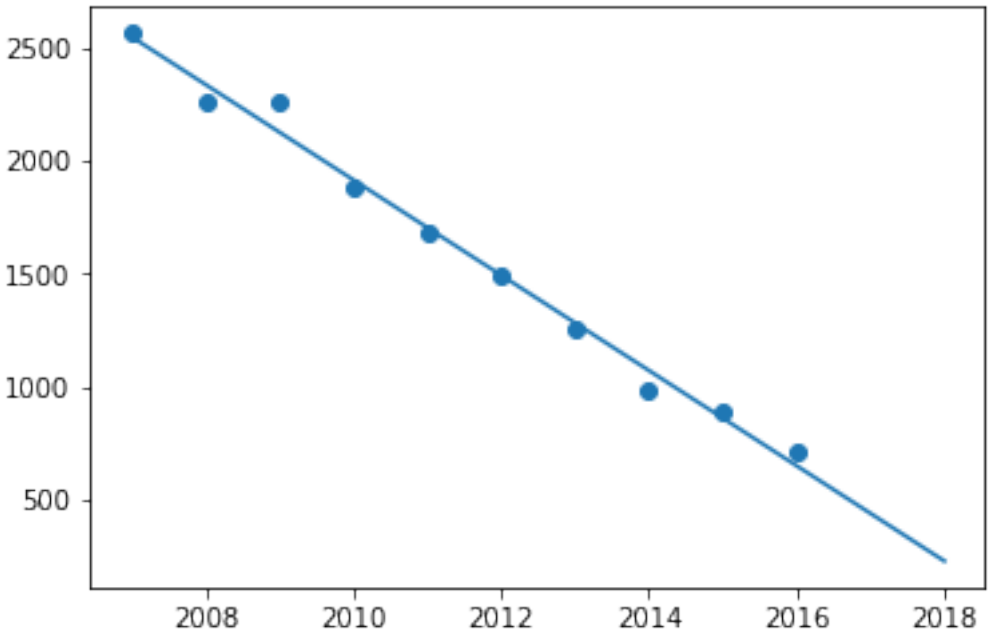
```
In [37]: #
         import matplotlib.pyplot as plt
         from sklearn.linear_model import LinearRegression
         x = AAAA[:,0]
         y = AAAA[:,1]
         model = LinearRegression(fit_intercept=True)

         model.fit(x[:, np.newaxis], y)

         xfit = np.linspace(2007,2018,11)
         yfit = model.predict(xfit[:, np.newaxis])

         plt.scatter(x, y)
         plt.plot(xfit, yfit)
         print(model.coef_[0])
         print(model.intercept_)
         Mslope=model.coef_[0]
         Bintercept=model.intercept_
```

```
-210.8484848484848
425721.9272727272
```




```
In [38]: #Okay.  This is the real answer to question 5
         print(Mslope*2018+Bintercept)
         #Use this one
```

```
229.68484848487424
```

```
In [39]: video=cd.loc[cd['Complaint Has Video Evidence'] == True]#All rows where there is video
         novideo=cd.loc[cd['Complaint Has Video Evidence'] == False]#All rows where there is no
```

```
In [40]: novideo.head(3)
```

```
Out[40]:      DateStamp  UniqueComplaintId  Close Year  Received Year  \
         0   02/07/2017              70245        2009           2008
         1   02/07/2017              70244        2007           2006
         12  02/07/2017              70243        2011           2011

                   ...        Reason For Initial Contact  Allegation FADO Type  \
         0         ...                             Other    Offensive Language
         1         ...       Report-noise/disturbance    Abuse of Authority
         12        ...          Other violation of VTL           Discourtesy

             Allegation Description ComplaintDuration
         0      Physical disability                 1
         1         Threat of arrest                 1
         12                   Word                 0

         [3 rows x 17 columns]
```

```
In [41]: videoarray=video['Is Full Investigation'].values #All rows where there is a video. valu
         novideoarray=novideo['Is Full Investigation'].values #All rows where there is not a vid
         print(videoarray)
         print(novideoarray)
```

```
[ True False False ... False  True  True]
[False  True False ... False False False]
```

```
In [85]: #Observed
         video=cd.loc[cd['Complaint Has Video Evidence'] == True]#All rows where there is video
         vidinvest=video.loc[video['Is Full Investigation'] == True]#All rows where there is vid
         vTrueItrue=len(vidinvest)
         print(vTrueItrue)
         videofalse=cd.loc[cd['Complaint Has Video Evidence'] == False]#All rows where there is
         vidfalseinvestfalse=videofalse.loc[videofalse['Is Full Investigation'] == False]#All ro
         vFalseIFalse=len(vidfalseinvestfalse)
         print(vFalseIFalse)

         video=cd.loc[cd['Complaint Has Video Evidence'] == True]#All rows where there is video
         vidinvestfalse=video.loc[video['Is Full Investigation'] == False]#All rows where there
         vTrueIfalse=len(vidinvestfalse)
         print(vTrueIfalse)
         videofalse=cd.loc[cd['Complaint Has Video Evidence'] == False]#All rows where there is
         vidfalseinvest=videofalse.loc[videofalse['Is Full Investigation'] == True]#All rows whe
         vFalseITrue=len(vidfalseinvest)
         print(vFalseITrue)
         Observed= [vTrueItrue, vTrueIfalse, vFalseITrue,vFalseIFalse]
```

```
1476
45987
596
22186
```

```
In [42]: vdmean=np.mean(videoarray)#This is just to get an idea of what the chi-sq should be
         vdstdev=np.std(videoarray)
         vnum=len(videoarray) #number of rows with a video
         print(vdmean,vdstdev,vnum)
```

```
0.7123552123552124 0.45266462616994124 2072
```

```
In [43]: novdmean=np.mean(novideoarray)#This is just to get an idea of what the chi-sq should be
         novdstdev=np.std(novideoarray)
         novnum=len(novideoarray) #number of rows without a video
         print(novdmean,novdstdev,novnum)
```

```
0.32543675648717235 0.4685378042530627 68173
```

In [44]:
```
investigatetruevideo=video.loc[video['Is Full Investigation'] == True] #investigations
investigatetruenovideo=novideo.loc[novideo['Is Full Investigation'] == True] #investiga
vidinvtrue=len(investigatetruevideo) #number of investigations when there is a video
novidinvtrue=len(investigatetruenovideo) #number of investigations where there is no vi
suminvestigations=vidinvtrue+novidinvtrue #Sum total of investigations
print(suminvestigations)
```

```
23662
```

In [46]:
```
investigatefalsevideo=video.loc[video['Is Full Investigation'] == False] #investigation
investigatefalsenovideo=novideo.loc[novideo['Is Full Investigation'] == False] #investi
vidinvfalse=len(investigatefalsevideo) #number of investigations when there is a video
novidinvfalse=len(investigatefalsenovideo) #number of investigations where there is no
sumnoinvestigations=vidinvfalse+novidinvfalse#Sum total of investigations
print(suminvestigations)
```

```
23662
```

In [47]:
```
print(2)
```

```
2
```

In [48]:
```
#Okay.  Chi-squared means we test that there's interaction between whether theres a vid
sumtotal=len(novideo)+len(video)
u1=len(video)/sumtotal #with video / total rows
u2=len(novideo)/sumtotal #without video / total rows
v1=(suminvestigations)/sumtotal #number of investigations when there is a video / total
v2=(sumnoinvestigations)/sumtotal #number of investigations when there is no video /tot
print([u1,u2,v1,v2])
```

```
[0.029496761335326357, 0.9705032386646737, 0.33684959783614493, 0.6631504021638551]
```

In [49]:
```
print([sumtotal,investigatetruevideo,investigatetruenovideo])
```

```
[70245,          DateStamp  UniqueComplaintId  Close Year  Received Year  \
133     02/07/2017              70199        2015           2015
563     02/07/2017              70065        2016           2016
593     02/07/2017              70059        2016           2015
618     02/07/2017              70054        2015           2014
...            ...                ...         ...            ...
205835  02/07/2017                280        2016           2015
205901  02/07/2017                260        2016           2016
206680  02/07/2017                 16        2016           2015
206690  02/07/2017                 10        2014           2014
```

```
             ...                          Reason For Initial Contact  \
133          ...                                        Aided case
563          ...        PD suspected C/V of violation/crime - street
593          ...                            Other violation of VTL
618          ...                                        Aided case
...          ...                                              ...
205835       ...                                 Moving violation
205901       ...                            Other violation of VTL
206680       ...                                            Other
206690       ...                            Other violation of VTL

       Allegation FADO Type                Allegation Description  \
133      Abuse of Authority                                 Other
563      Abuse of Authority                      Strip-searched
593      Abuse of Authority                        Vehicle stop
618      Abuse of Authority                    Property damaged
...                     ...                                 ...
205835   Abuse of Authority                      Vehicle search
205901   Abuse of Authority                    Threat of arrest
206680   Abuse of Authority  Refusal to provide name/shield number
206690              Force                          Pepper spray

      ComplaintDuration
133                   0
563                   0
593                   1
618                   1
...                 ...
205835                1
205901                0
206680                1
206690                0

[1476 rows x 17 columns],         DateStamp  UniqueComplaintId  Close Year  Received Year  \
1       02/07/2017              70244        2007           2006
15      02/07/2017              70241        2011           2011
21      02/07/2017              70238        2007           2006
28      02/07/2017              70235        2006           2005
...            ...                ...         ...            ...
206683  02/07/2017                 14        2012           2011
206685  02/07/2017                 13        2009           2008
206692  02/07/2017                  8        2007           2006
206707  02/07/2017                  4        2013           2013

             ...                          Reason For Initial Contact  \
1            ...                            Report-noise/disturbance
15           ...                    Report-gun possession/shots fired
21           ...            PD suspected C/V of violation/crime - bldg
```

```
28            ...                                Report of other crime
...           ...                                                  ...
206683        ...                                                Other
206685        ...         PD suspected C/V of violation/crime - subway
206692        ...         PD suspected C/V of violation/crime - street
206707        ...                           Execution of search warrant

        Allegation FADO Type            Allegation Description  \
1         Abuse of Authority                  Threat of arrest
15        Abuse of Authority  Premises entered and/or searched
21        Abuse of Authority              Question and/or stop
28        Abuse of Authority                  Threat of arrest
...                      ...                               ...
206683    Offensive Language                            Gender
206685    Abuse of Authority                              Stop
206692    Abuse of Authority                 Search (of person)
206707    Abuse of Authority  Premises entered and/or searched

       ComplaintDuration
1                      1
15                     0
21                     1
28                     1
...                  ...
206683                 1
206685                 1
206692                 1
206707                 0

[22186 rows x 17 columns]]
```

```
In [90]: u1=(1/sumtotal)*(vidinvtrue+vidinvfalse)#[sumtotal,investigatetruevideo,investigatetrue
         u2=(1/sumtotal)*(novidinvtrue+novidinvfalse)#[sumtotal,investigatetruevideo,investigate
         v1=(1/sumtotal)*(vidinvtrue+novidinvtrue)
         v2=(1/sumtotal)*(vidinvfalse+novidinvfalse)
         print([u1,u2,v1,v2])
```

```
[0.029496761335326357, 0.9705032386646737, 0.33684959783614493, 0.6631504021638551]
```

```
In [91]: E11=float(sumtotal*(u1)*(v1))#vidtrueinvtrue
         E22=float(sumtotal*(u2)*(v2))#vidfalseinvfalse
         E12=float(sumtotal*(u1)*(v2))#vidtrueinvfalse
         E21=float(sumtotal*(u2)*(v1))#vidfalseinvtrue
         print([E11,E22,E12,E21])
```

```
[697.9523667164923, 45208.9523667165, 1374.0476332835078, 22964.04763328351]
```

```
In [93]: Observed= [vTrueItrue, vTrueIfalse, vFalseITrue,vFalseIFalse]
         print(Observed)
```

```
[1476, 596, 22186, 45987]
```

```
In [96]: #Finally, the answer to question six
         chisqtest=(((vTrueItrue-E11)**2)/E11)+(((vFalseIFalse-E22)**2)/E22)+(((vFalseITrue-E21)
         print(chisqtest)
         #So, yes, it has an effect
```

```
1347.6513831287339
```

```
In [140]: #Starting question 8
          complaints2016=cd.loc[cd['Received Year'] == 2016]
          #print(complaints2016.head())
          officers=36000
          ncomplaints2016=len(complaints2016)
          print(ncomplaints2016)
          complaintspercapitaNY=ncomplaints2016/officers
          complaintsperofficer2016=ncomplaints2016/36000
          officerspercomplaint=1/complaintsperofficer2016
          c2016brook=len(complaints2016.loc[complaints2016['Borough of Occurrence']=='Brooklyn']
          c2016bronx=len(complaints2016.loc[complaints2016['Borough of Occurrence']=='Bronx'])
          c2016man=len(complaints2016.loc[complaints2016['Borough of Occurrence']=='Manhattan'])
          c2016queens=len(complaints2016.loc[complaints2016['Borough of Occurrence']=='Queens'])
          c2016staten=len(complaints2016.loc[complaints2016['Borough of Occurrence']=='Staten Is
          print(c2016staten)
          manpercap=c2016man/1664727
          brookpercap=c2016brook/2648771
          statenpercap=c2016staten/479458
          bronxpercap=c2016bronx/1471160
          queenspercap=c2016queens/2358582
          #find officers per borough
          offbrook=c2016brook*officerspercomplaint
          offbronx=c2016bronx*officerspercomplaint
          offman=c2016man*officerspercomplaint
          offqueens=c2016queens*officerspercomplaint
          offstaten=c2016staten*officerspercomplaint
```

```
3699
164
```

```
In [145]: brook=offbrook/23
          bronk=offbronx/12
          man=offman/22
          queen=offqueens/16
```

```
staten=offstaten/4
print([brook,bronk,man,queen,staten])
```

```
[483.65598222786423, 664.2335766423358, 399.4691439946915, 378.345498783455, 399.0267639902677]
```

In [146]:
```
#This is the last question
bronk/staten
```

Out[146]:
```
1.6646341463414633
```

In [138]:
```
q=1
r='1'

print(str('1')+'a')
print(str(1)+'b')
d={'a':1,'b':2}
print(len(d))
e={'a':4}
type(d)
sss = pd.Series(d, name='bow')
print(sss)
sss.to_csv('dicttest.csv')
#pd.DataFrame(d.items(), columms=['word', 'Value'])


from collections import Counter
A = Counter({'a':1, 'b':2, 'c':3})
B = Counter({'b':3, 'c':4, 'd':5})
C=A + B
print(C)
```

```
1a
1b
2
a    1
b    2
Name: bow, dtype: int64
Counter({'c': 7, 'b': 5, 'd': 5, 'a': 1})
```

In [62]:
```
#variable is likelyhood of investigation
#H0 investigation is independent of presence ofvideo
#H1 investigation is not independent of presence of video
totalinvestigated=vidinvtrue+novidinvtrue
proportioninvestigated=totalinvestigated/(sumtotal)
proportionnoinvestigated=1-totalinvestigated/(sumtotal)
print(proportioninvestigated)
```

```
0.33684959783614493
```

```
In [63]: expvidinv=proportioninvestigated*vnum #expected investigations when there is a video
         expvidnoinv=(proportionnoinvestigated)*vnum #expected no investigations where there is
         expnovidnoinv=(proportionnoinvestigated)*novnum #start checking here
         expnovidinv=proportioninvestigated*novnum
         print([expvidinv,expvidnoinv,expnovidnoinv,expnovidinv]) #expectinvestwithvideo expectn
```

```
[697.9523667164923, 1374.0476332835076, 45208.95236671649, 22964.04763328351]
```

```
In [86]: Observed= [vTrueItrue, vTrueIfalse, vFalseITrue,vFalseIFalse]
         print(Observed)
```

```
[1476, 596, 22186, 45987]
```

```
In [58]: (((expnovidinv-E11)**2)/E11)
```

```
Out[58]: 710333.5729803795
```

```
In [61]: (((expvidnoinv-E22)**2)/E22)
```

```
Out[61]: 42502.61889288507
```

```
In [67]: chisqtest=(((expvidinv-E11)**2)/E11)+(((expnovidnoinv-E22)**2)/E22)+(((expvidnoinv-E21)
         print(chisqtest)
```

```
359535.3886750671
```

```
In [ ]: numbercomplaint=[len(cd.loc[cd['Borough of Occurrence'] == 'Brooklyn'].index),len(cd.loc
```

```
In [66]: len(cd.index)
```

```
Out[66]: 70245
```

```
In [ ]: #This is the answer to the second question
        max(numbercomplaint)/sum(numbercomplaint)
```

```
In [ ]: df = pd.DataFrame({"A":["foo", "foo", "foo", "bar"], "B":[0,1,1,1], "C":["A","A","B","A"
        df.head()
```

```
In [ ]: print()
```

```
In [ ]: df.drop_duplicates(keep='first')
```

```
In [ ]: df.drop_duplicates(subset=['A', 'C'], keep=False)
        df.head()
```

```
In [ ]: df = pd.DataFrame({"A":["foo", "foo", "foo", "bar"], "B":[0,1,1,1], "C":["A","A","B","A"
        df.head()

In [ ]: df = pd.DataFrame({"A":["foo", "foo", "foo", "bar"], "B":[0,1,1,1], "C":["A","A","B","A"
        df=df.drop_duplicates(subset=['A'], keep='first')
        df.head()

In [ ]: df = df[~df.index.duplicated(keep='first')]

In [ ]: df.head()

In [ ]: df=df[~df.index.duplicated(keep='first')]

In [ ]: df

In [ ]: df.groupby(level=df.index.names).last()

In [ ]: df = df.reset_index().drop_duplicates(subset='index', keep='last').set_index('index')

In [ ]: df.head()

In [ ]: idx = pd.Index(['lama', 'cow', 'lama', 'beetle', 'lama', 'hippo'])
        idx.drop_duplicates(keep='first')
```