Boat Safety-Sutow Brett

July 28, 2021

[3]:
```
#Boat Safety#
#Setup#
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import squarify as sq
import statsmodels.api as sm
```

[4]:
```
#Loads Data#
VesselAccidents= pd.read_csv('/Users/Brett/Desktop/Vessel Type Accidents 2020.
 ↪csv')
Factors= pd.read_csv('/Users/Brett/Desktop/2020 Factors for Accident.csv')
PersonalWaterCraft= pd.read_csv('/Users/Brett/Desktop/watercraft.csv')
AccidentType= pd.read_csv('/Users/Brett/Desktop/Accident Type Boating Datam2020.
 ↪csv')
Accidents2001= pd.read_csv('/Users/Brett/Desktop/Data since 2001.csv')
```

[41]:
```
#Prints Heads#
print(VesselAccidents)
print(Factors)
print(PersonalWaterCraft)
print(AccidentType)
print(Accidents2001)
```

```
          Type of Boat  Drownings  Other Deaths  Total Deaths  Total Injuries
0       Open Motorboat        260           116           376            1520
1  Personal Watercraft         19            47            66             896
2      Cabin Motorboat         16            17            33             298
3              Pontoon         53            14            67             232
4          Canoe/Kayak        131            23           154             117
                          Crash Reason  Accidents  Deaths  Injuries
0                          Alcohol Use        296     115       260
1                             Drug Use         12       8         2
2                       Excessive Speed        418      32       345
3                       Failure to Vent         54       5        89
4                      Improper Lookout        578      28       409
5  Inadqeuate Onboard Navigation Lights         21       0        16
```

6	Navigation Rules Violation	316	26	220	
7	Operator Inattention	664	55	383	
8	Operator Inexperience	612	56	343	
9	Restricted Vision	67	2	47	
10	Sharp Turn	67	7	66	
11	Starting in Gear	4	0	2	

	Year	Fatalities	Injured persons	Accidents	Sales	Number in use
0	1987	5	156	376	29000.0	92756.0
1	1988	20	254	650	48000.0	126881.0
2	1989	20	402	844	64000.0	178510.0
3	1990	28	532	1162	72000.0	241376.0
4	1991	26	708	1513	68000.0	305915.0
5	1992	34	730	1650	79000.0	372283.0
6	1993	35	915	2236	107000.0	454545.0
7	1994	56	1338	3002	142000.0	600000.0
8	1995	68	1617	3986	200000.0	760000.0
9	1996	57	1837	4099	191000.0	900000.0
10	1997	84	1812	4070	176000.0	1000000.0
11	1998	78	1743	3607	130000.0	1180000.0
12	1999	66	1614	3374	106000.0	1200000.0
13	2000	68	1580	3268	92000.0	1230000.0
14	2001	50	1424	2562	80900.0	1220000.0
15	2002	71	1362	2225	79300.0	1220000.0
16	2003	57	1228	1994	80600.0	1170000.0
17	2004	56	952	1664	79500.0	1250000.0
18	2005	65	1007	1692	80200.0	1230000.0
19	2006	68	919	1631	82200.0	1190000.0
20	2007	67	982	1655	79900.0	1190000.0
21	2008	45	920	1459	62600.0	1240000.0
22	2009	42	878	1332	44500.0	1330000.0
23	2010	38	776	1221	41600.0	1270000.0
24	2011	44	764	1158	42900.0	1270000.0
25	2012	58	721	1111	38500.0	1250000.0
26	2013	36	601	954	39400.0	1270000.0
27	2014	34	592	891	47900.0	1200000.0
28	2015	33	623	1023	54900.0	NaN
29	2016	46	675	1072	59000.0	NaN
30	2017	46	624	1028	NaN	NaN
31	2018	42	634	1055	69000.0	NaN
32	2019	46	614	1062	73000.0	NaN

	Accident Type	Number of Accidents	Number of Deaths	\
0	Collision with recreational vessel	1379	66	
1	Flooding/Swamping	589	84	
2	Collision with fixed object	542	62	
3	Grounding	484	14	
4	Falls overboard	335	181	

Number of Injuries

```
0                 831
1                 128
2                 389
3                 255
4                 161
```

	Year	Deaths	Injuries	Accidents
0	2001	681	4274	6419
1	2002	750	4062	5705
2	2003	703	3888	5438
3	2004	676	3363	4904
4	2005	697	3451	4969
5	2006	710	3474	4967
6	2007	685	3673	5191
7	2008	709	3331	4789
8	2009	736	3358	4730
9	2010	672	3153	4604
10	2011	758	3081	4588
11	2012	651	3000	4515
12	2013	560	2620	4062
13	2014	610	2678	4064
14	2015	626	2613	4158
15	2016	701	2903	4463
16	2017	658	2629	4291
17	2018	633	2511	4145
18	2019	613	2559	4168
19	2020	767	3191	5265

[42]:
```
#Looks at the detailed described information for each dataset#
Accidents2001.describe()
```

[42]:

	Year	Deaths	Injuries	Accidents
count	20.00000	20.000000	20.000000	20.000000
mean	2010.50000	679.800000	3190.600000	4771.750000
std	5.91608	53.818995	518.517766	612.881449
min	2001.00000	560.000000	2511.000000	4062.000000
25%	2005.75000	646.500000	2665.750000	4260.250000
50%	2010.50000	683.000000	3172.000000	4667.000000
75%	2015.25000	709.250000	3456.750000	5024.500000
max	2020.00000	767.000000	4274.000000	6419.000000

[43]:
```
VesselAccidents.describe()
```

[43]:

	Drownings	Other Deaths	Total Deaths	Total Injuries
count	5.00000	5.000000	5.000000	5.000000
mean	95.80000	43.400000	139.200000	612.600000
std	102.83336	42.606338	139.770168	590.156589
min	16.00000	14.000000	33.000000	117.000000

```
25%     19.00000     17.000000     66.000000      232.000000
50%     53.00000     23.000000     67.000000      298.000000
75%    131.00000     47.000000    154.000000      896.000000
max    260.00000    116.000000    376.000000     1520.000000
```

[44]:
```
Factors.describe()
```

[44]:
```
         Accidents      Deaths    Injuries
count   12.000000   12.000000   12.000000
mean   259.083333   27.833333  181.833333
std    255.614219   33.999554  160.859528
min      4.000000    0.000000    2.000000
25%     45.750000    4.250000   39.250000
50%    181.500000   17.000000  154.500000
75%    458.000000   37.750000  343.500000
max    664.000000  115.000000  409.000000
```

[45]:
```
PersonalWaterCraft.dropna()
PersonalWaterCraft.describe()
```

[45]:
```
              Year  Fatalities  Injured persons    Accidents          Sales  \
count    33.00000   33.000000        33.000000    33.000000      32.000000
mean   2003.00000   48.151515       955.575758  1837.151515   82496.875000
std       9.66954   18.313453       455.549807  1064.888026   43173.808401
min    1987.00000    5.000000       156.000000   376.000000   29000.000000
25%    1995.00000   35.000000       624.000000  1062.000000   53175.000000
50%    2003.00000   46.000000       878.000000  1513.000000   76000.000000
75%    2011.00000   65.000000      1338.000000  2236.000000   84650.000000
max    2019.00000   84.000000      1837.000000  4099.000000  200000.000000

       Number in use
count   2.800000e+01
mean    9.265095e+05
std     4.308375e+05
min     9.275600e+04
25%     5.636362e+05
50%     1.190000e+06
75%     1.232500e+06
max     1.330000e+06
```

[46]:
```
AccidentType.describe()
```

[46]:
```
       Number of Accidents  Number of Deaths  Number of Injuries
count             5.000000           5.00000            5.000000
mean            665.800000          81.40000          352.800000
std             409.987439          61.40684          285.863954
min             335.000000          14.00000          128.000000
```

```
25%          484.000000        62.00000        161.000000
50%          542.000000        66.00000        255.000000
75%          589.000000        84.00000        389.000000
max         1379.000000       181.00000        831.000000
```

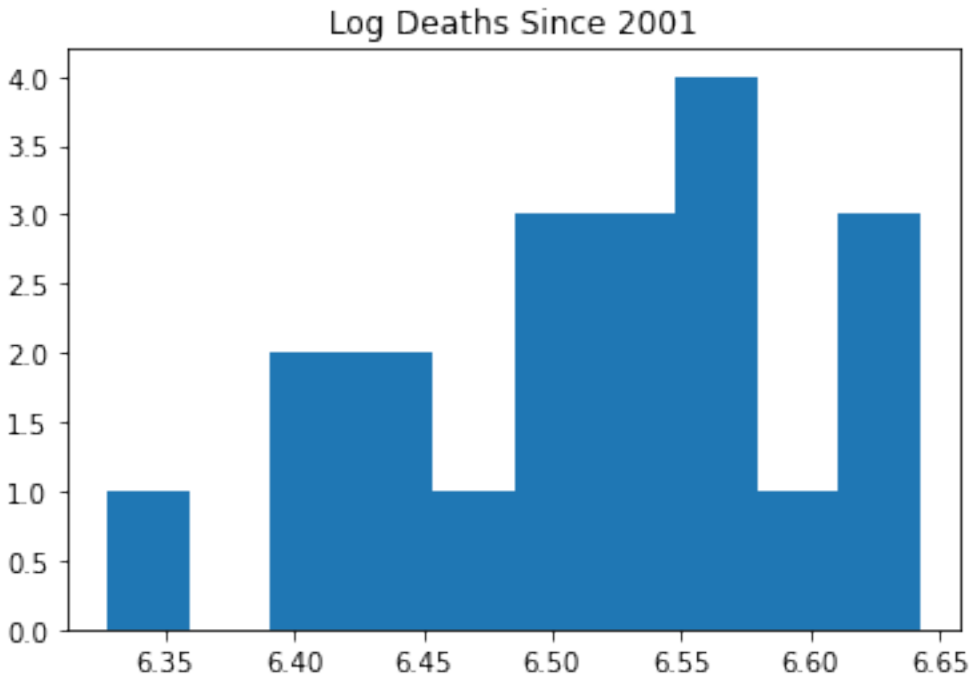
[47]:
```
#Creates Regression Analysis for deaths since 2001. Provides outlook for the
 ↪dat we are working with#

Accidents2001['LogDeaths']= np.log(Accidents2001).Deaths
X1 = Accidents2001.Year
X1 = sm.add_constant(X1)
y1 = Accidents2001.LogDeaths
mod = sm.OLS(y1,X1)
res = mod.fit()
print(res.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              LogDeaths   R-squared:                       0.125
Model:                            OLS   Adj. R-squared:                  0.076
Method:                 Least Squares   F-statistic:                     2.564
Date:                Tue, 27 Jul 2021   Prob (F-statistic):              0.127
Time:                        17:54:47   Log-Likelihood:                 23.816
No. Observations:                  20   AIC:                            -43.63
Df Residuals:                      18   BIC:                            -41.64
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         16.1978      6.045      2.680      0.015       3.498      28.898
Year          -0.0048      0.003     -1.601      0.127      -0.011       0.002
==============================================================================
Omnibus:                        1.733   Durbin-Watson:                   1.552
Prob(Omnibus):                  0.420   Jarque-Bera (JB):                0.418
Skew:                           0.037   Prob(JB):                        0.811
Kurtosis:                       3.704   Cond. No.                     7.01e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 7.01e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

[48]:
```
#Plots Histogram for above data#
plt.hist(y1, bins = 10)
```

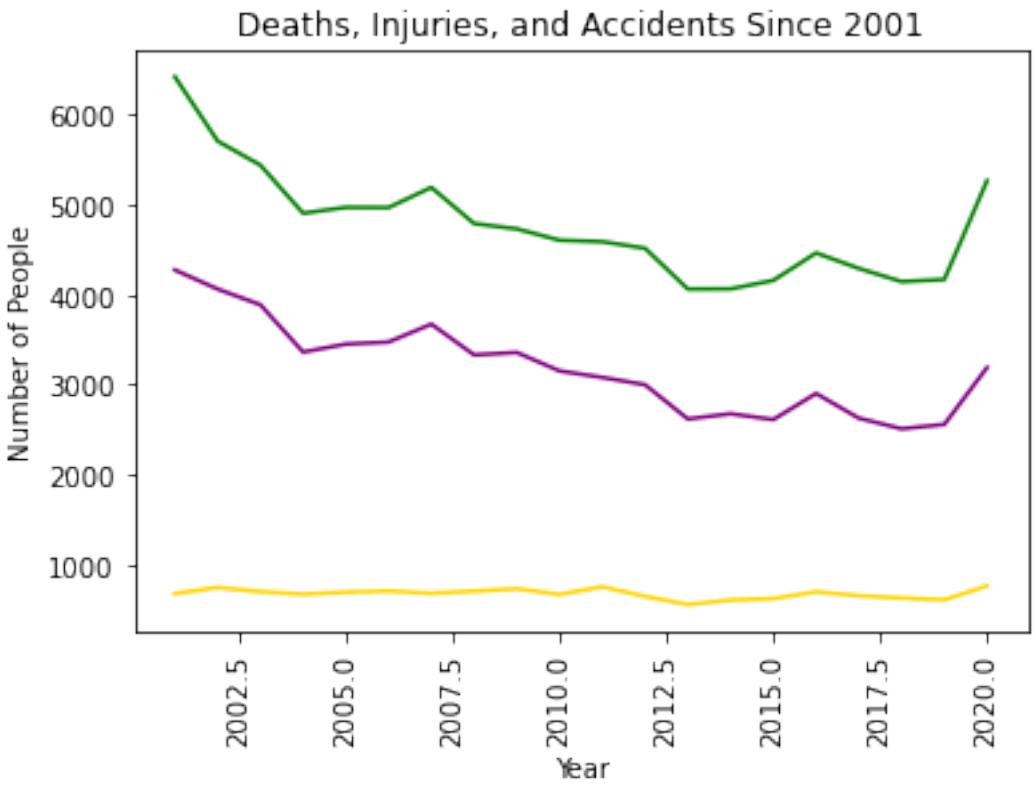
```
plt.title('Log Deaths Since 2001')
```

[48]: Text(0.5, 1.0, 'Log Deaths Since 2001')



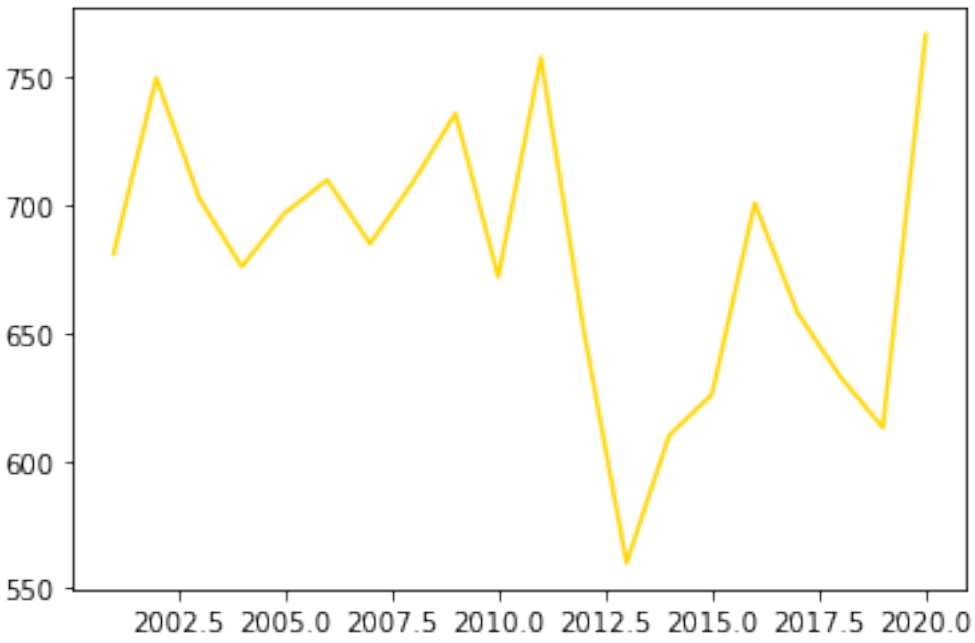
[20]:
```
#Shows graph of data over the years#
x1= Accidents2001.Year
y2= Accidents2001.Deaths
y3=Accidents2001.Accidents
y4= Accidents2001['Injuries ']
plt.title('Deaths, Injuries, and Accidents Since 2001')
plt.xlabel('Year')
plt.ylabel('Number of People')
plt.xticks(rotation=90)
plt.plot(x1,y2, color= 'gold')
plt.plot(x1,y3, color= 'green')
plt.plot(x1,y4, color= 'purple')
print("Legend: Green is Accidents, Purple is Injuries, Gold is Deaths")
```

Legend: Green is Accidents, Purple is Injuries, Gold is Deaths



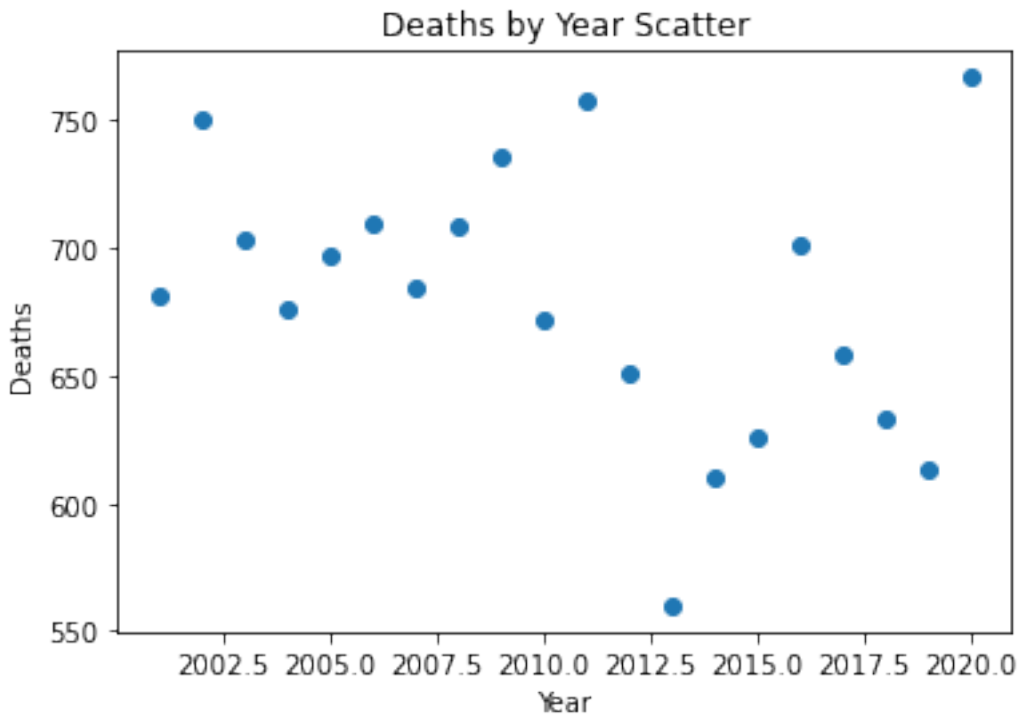
[70]:
```
x1= Accidents2001.Year
y2= Accidents2001.Deaths
plt.plot(x1,y2, color= 'gold')
```

[70]: [<matplotlib.lines.Line2D at 0x7f8582e14820>]



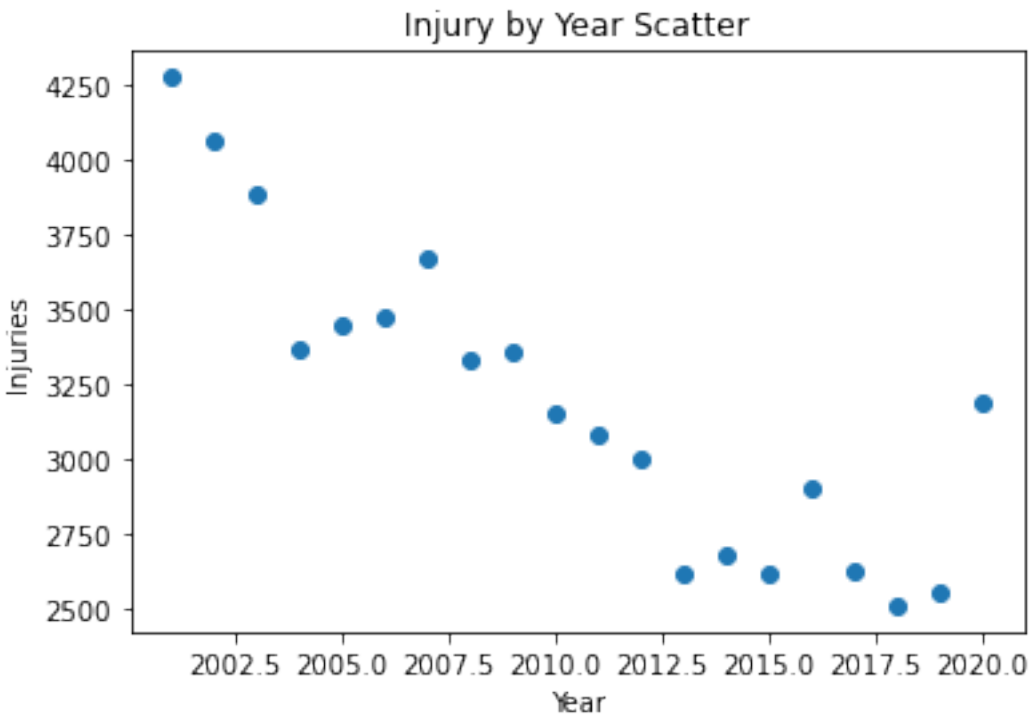
[64]:
```
plt.scatter(x=Accidents2001['Year'], y=Accidents2001['Deaths'])
plt.xlabel('Year')
plt.ylabel('Deaths')
plt.title('Deaths by Year Scatter')
```

[64]:
```
Text(0.5, 1.0, 'Deaths by Year Scatter')
```



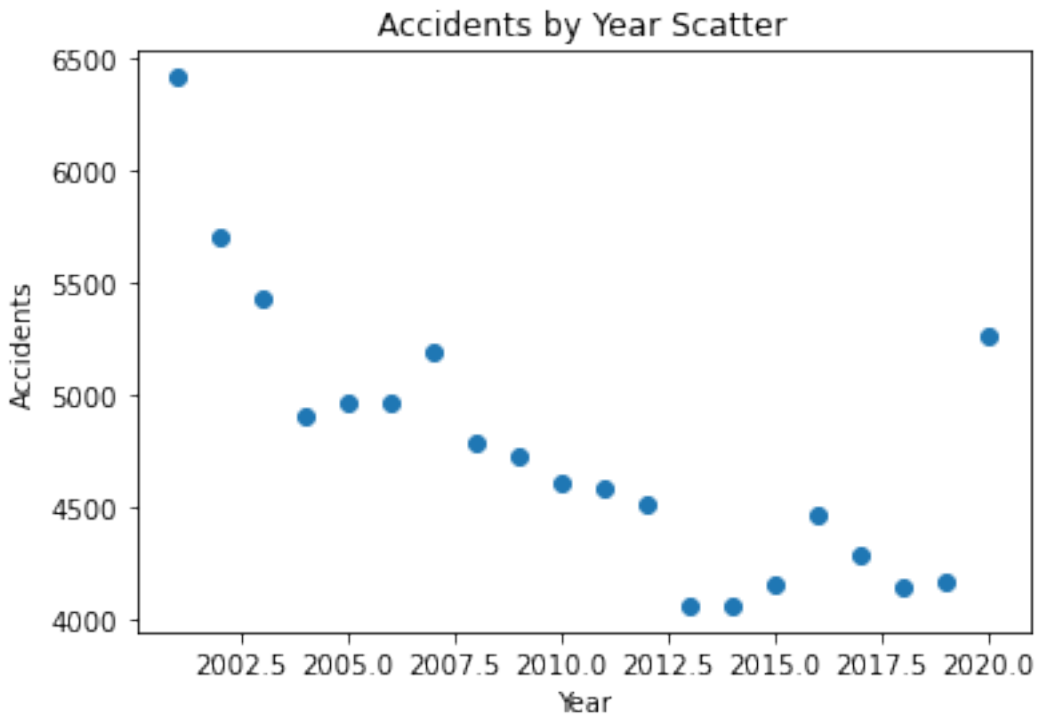
[66]:
```
plt.scatter(x=Accidents2001['Year'], y=Accidents2001['Injuries '])
plt.xlabel('Year')
plt.ylabel('Injuries')
plt.title('Injury by Year Scatter')
```

[66]: Text(0.5, 1.0, 'Injury by Year Scatter')



[71]:
```
plt.scatter(x=Accidents2001['Year'], y=Accidents2001['Accidents'])
plt.xlabel('Year')
plt.ylabel('Accidents')
plt.title('Accidents by Year Scatter')
```

[71]: Text(0.5, 1.0, 'Accidents by Year Scatter')



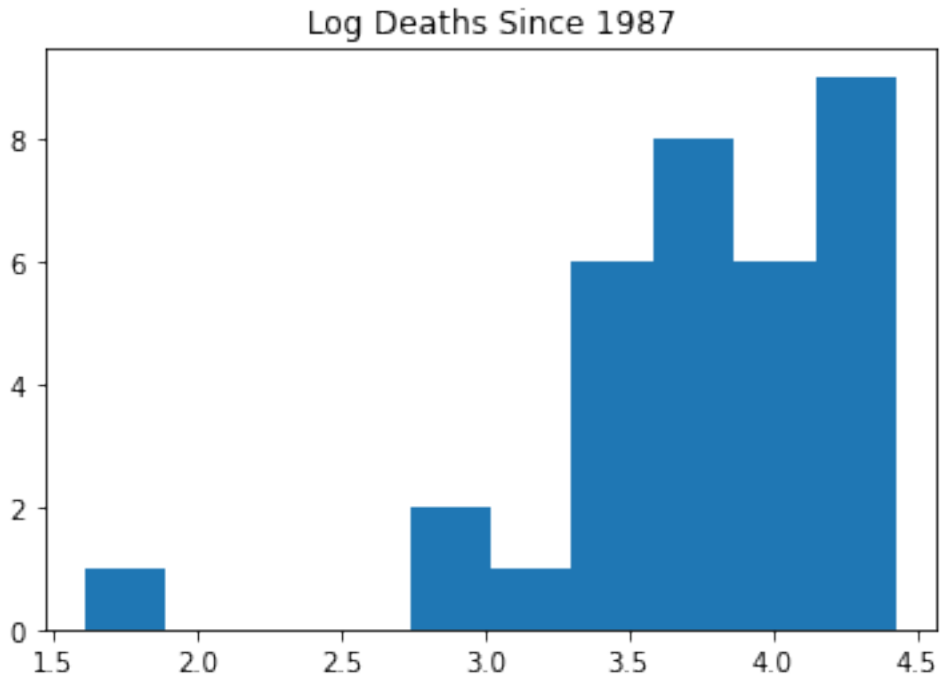
[50]:
```
#Personal Watercrafts#
PersonalWaterCraft['LogDeaths']= np.log(PersonalWaterCraft).Fatalities
X1 = PersonalWaterCraft.Year
X1 = sm.add_constant(X1)
y1 = PersonalWaterCraft.LogDeaths
mod = sm.OLS(y1,X1)
res = mod.fit()
print(res.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              LogDeaths   R-squared:                       0.112
Model:                            OLS   Adj. R-squared:                  0.083
Method:                 Least Squares   F-statistic:                     3.893
Date:                Tue, 27 Jul 2021   Prob (F-statistic):             0.0574
Time:                        17:54:49   Log-Likelihood:                -23.742
No. Observations:                  33   AIC:                             51.48
Df Residuals:                      31   BIC:                             54.48
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        -33.2673     18.771     -1.772      0.086     -71.551       5.017
```

```
Year            0.0185      0.009       1.973       0.057       -0.001        0.038
==============================================================================
Omnibus:                       18.404   Durbin-Watson:                   0.422
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               28.486
Skew:                          -1.345   Prob(JB):                     6.52e-07
Kurtosis:                       6.671   Cond. No.                     4.21e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 4.21e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

[51]:
```
#Plots Histogram for above data#
plt.hist(y1, bins = 10)
plt.title('Log Deaths Since 1987')
```

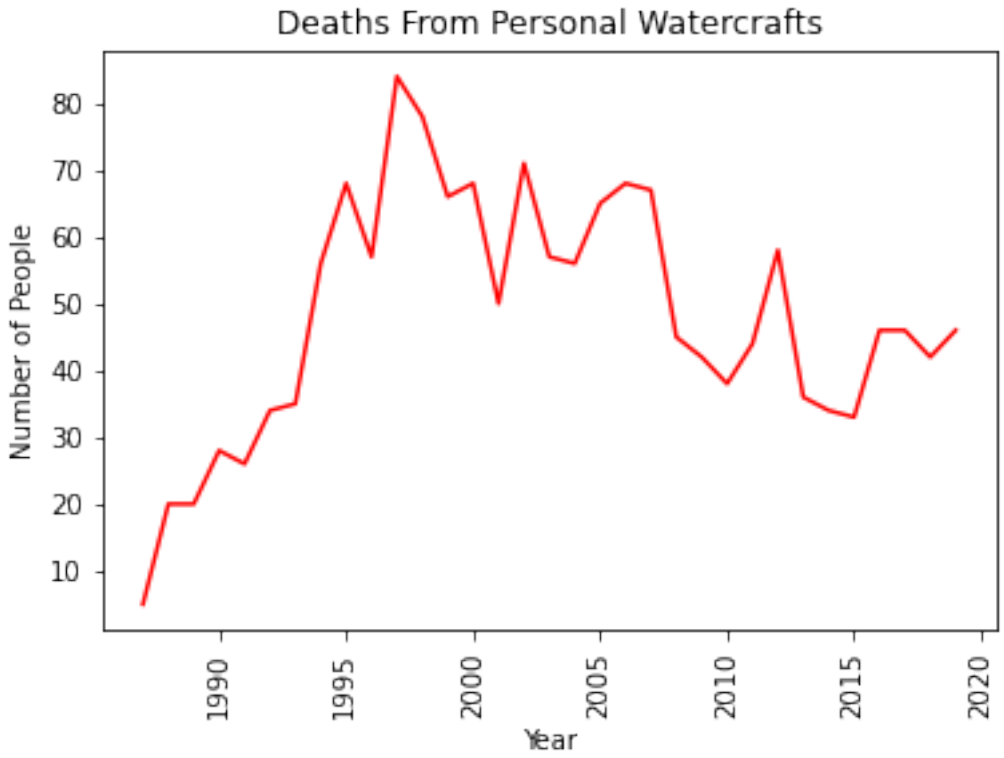
[51]:
```
Text(0.5, 1.0, 'Log Deaths Since 1987')
```



[52]:
```
#Personal Watercraft Deaths#
x1= PersonalWaterCraft.Year
y2= PersonalWaterCraft.Fatalities
plt.title('Deaths From Personal Watercrafts')
plt.xlabel('Year')
```

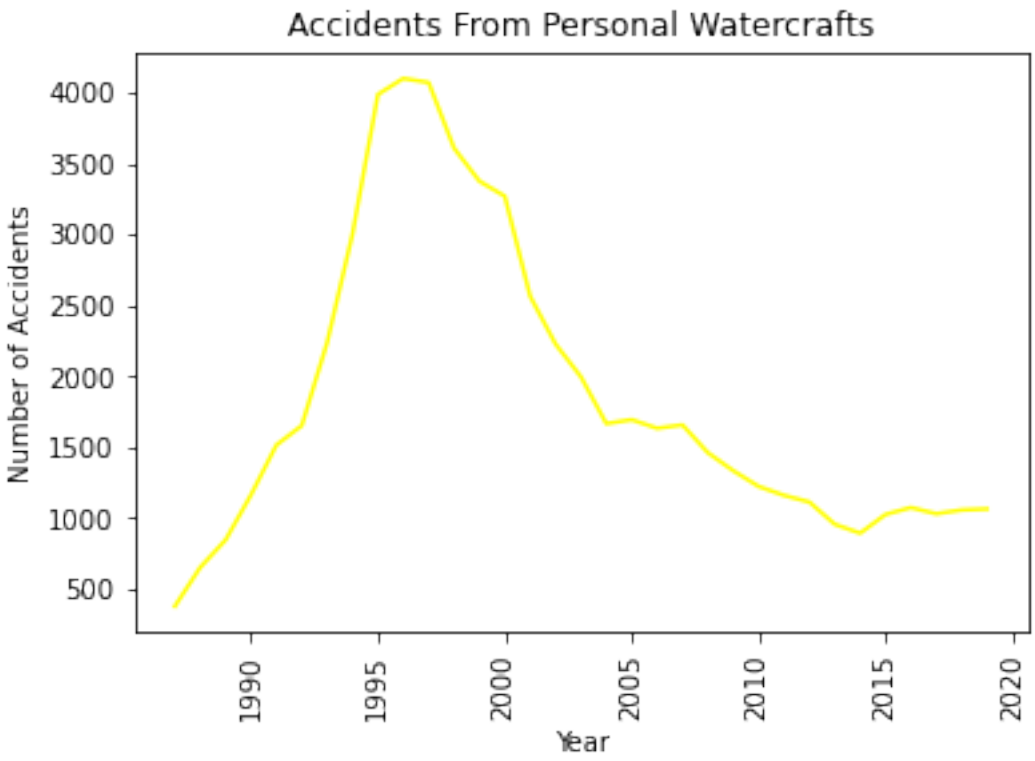
```
plt.ylabel('Number of People')
plt.xticks(rotation=90)
plt.plot(x1,y2, color= 'red')
```

[52]: [<matplotlib.lines.Line2D at 0x7f8581ba3cd0>]



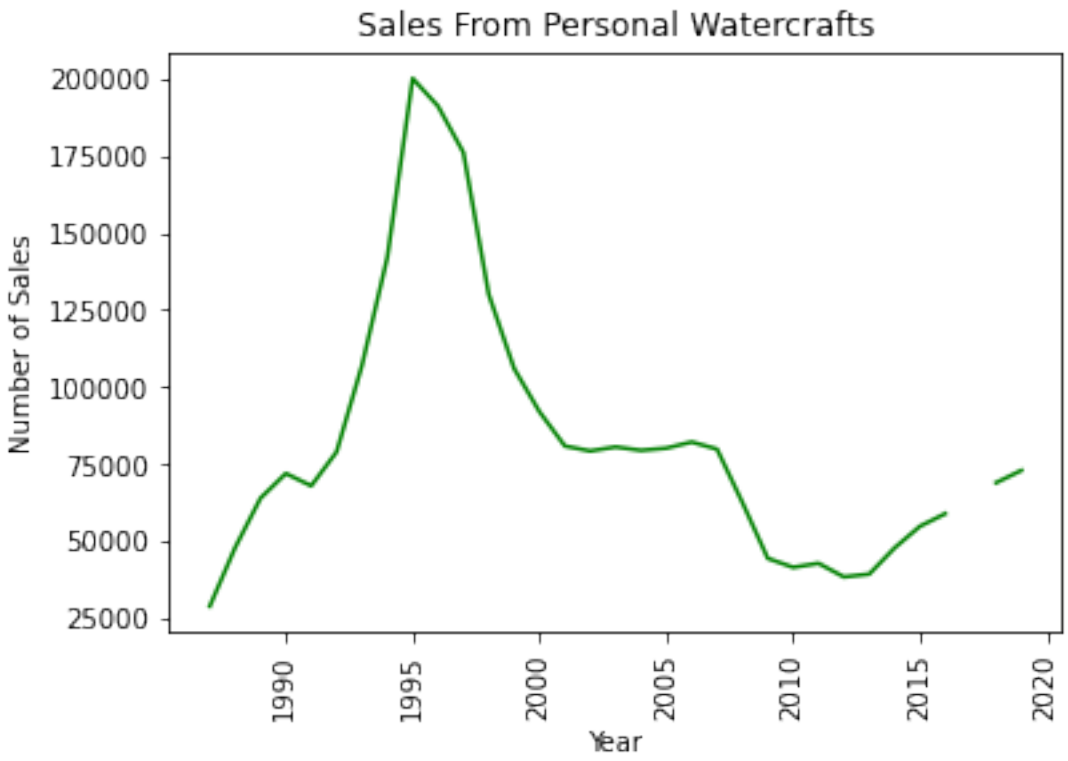
[53]:
```
#Personal Watercraft Accidents#
x1= PersonalWaterCraft.Year
y2= PersonalWaterCraft.Accidents
plt.title('Accidents From Personal Watercrafts')
plt.xlabel('Year')
plt.ylabel('Number of Accidents')
plt.xticks(rotation=90)
plt.plot(x1,y2, color= 'yellow')
```

[53]: [<matplotlib.lines.Line2D at 0x7f8581ccc340>]



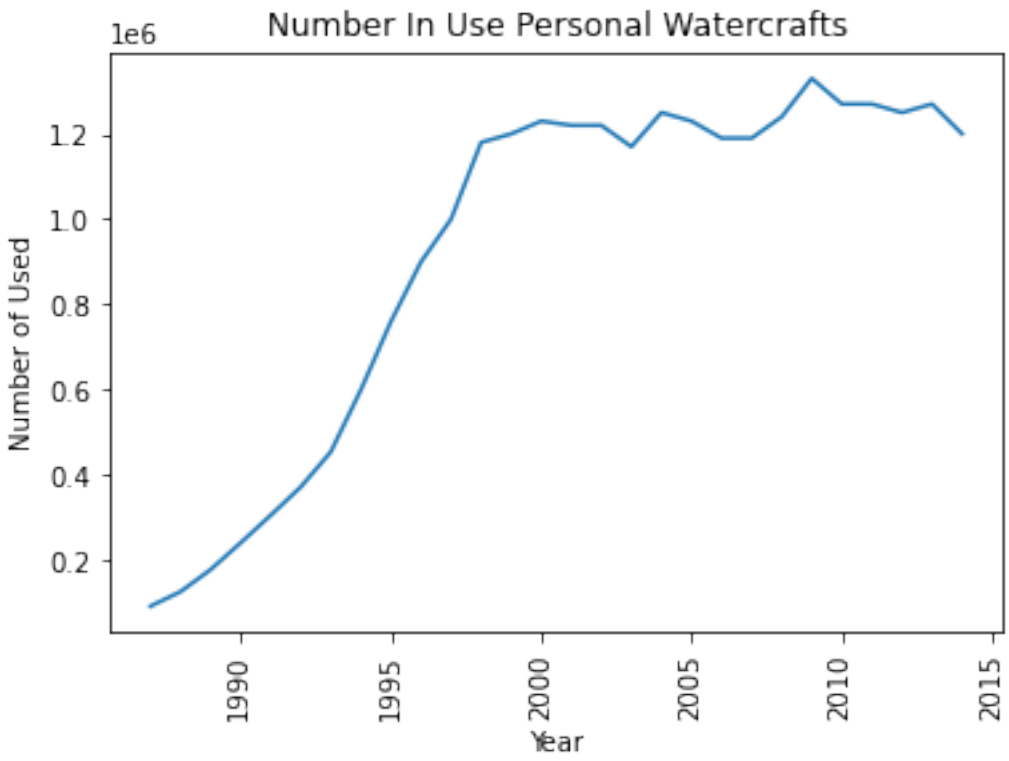
[54]:
```
#Sales Personal Watercraft#
x1= PersonalWaterCraft.Year
y2= PersonalWaterCraft['Sales']
plt.title('Sales From Personal Watercrafts')
plt.xlabel('Year')
plt.ylabel('Number of Sales')
plt.xticks(rotation=90)
plt.plot(x1,y2, color= 'green')
```

[54]:
```
[<matplotlib.lines.Line2D at 0x7f8581dc9b20>]
```



[55]:
```
#In Use Personal Watercraft#
x= PersonalWaterCraft.Year
y= PersonalWaterCraft['Number in use']
plt.title('Number In Use Personal Watercrafts')
plt.xlabel('Year')
plt.ylabel('Number of Used')
plt.xticks(rotation=90)
plt.plot(x,y)
```

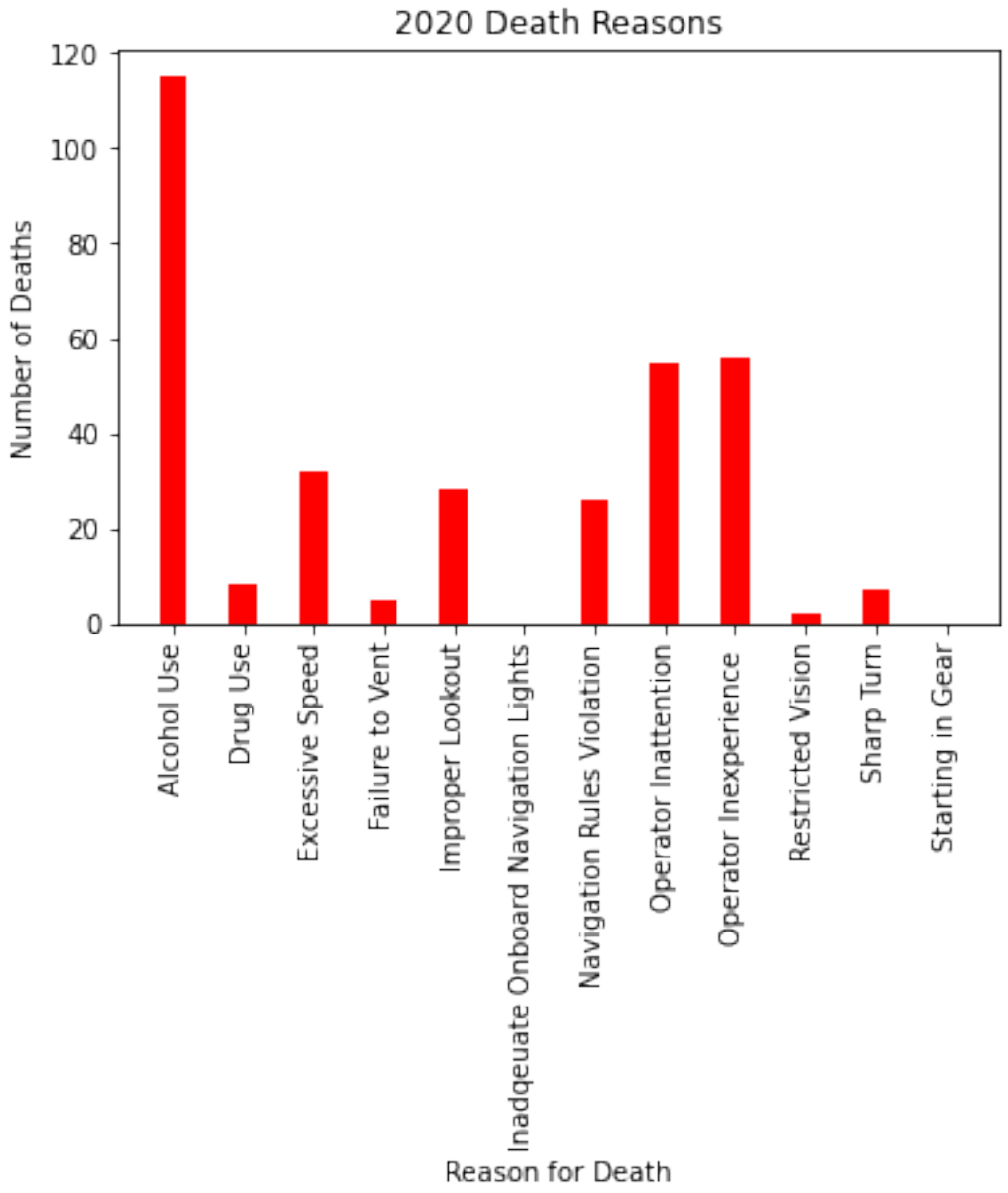
[55]:
```
[<matplotlib.lines.Line2D at 0x7f8581f0a370>]
```



[56]:
```
#2020 Reasons#
print(Factors)

plt.bar(Factors['Crash Reason'], Factors['Deaths'], color ='red',
        width = 0.4)

plt.xlabel("Reason for Death")
plt.ylabel("Number of Deaths")
plt.title("2020 Death Reasons")
plt.xticks(rotation=90)
plt.show()
```

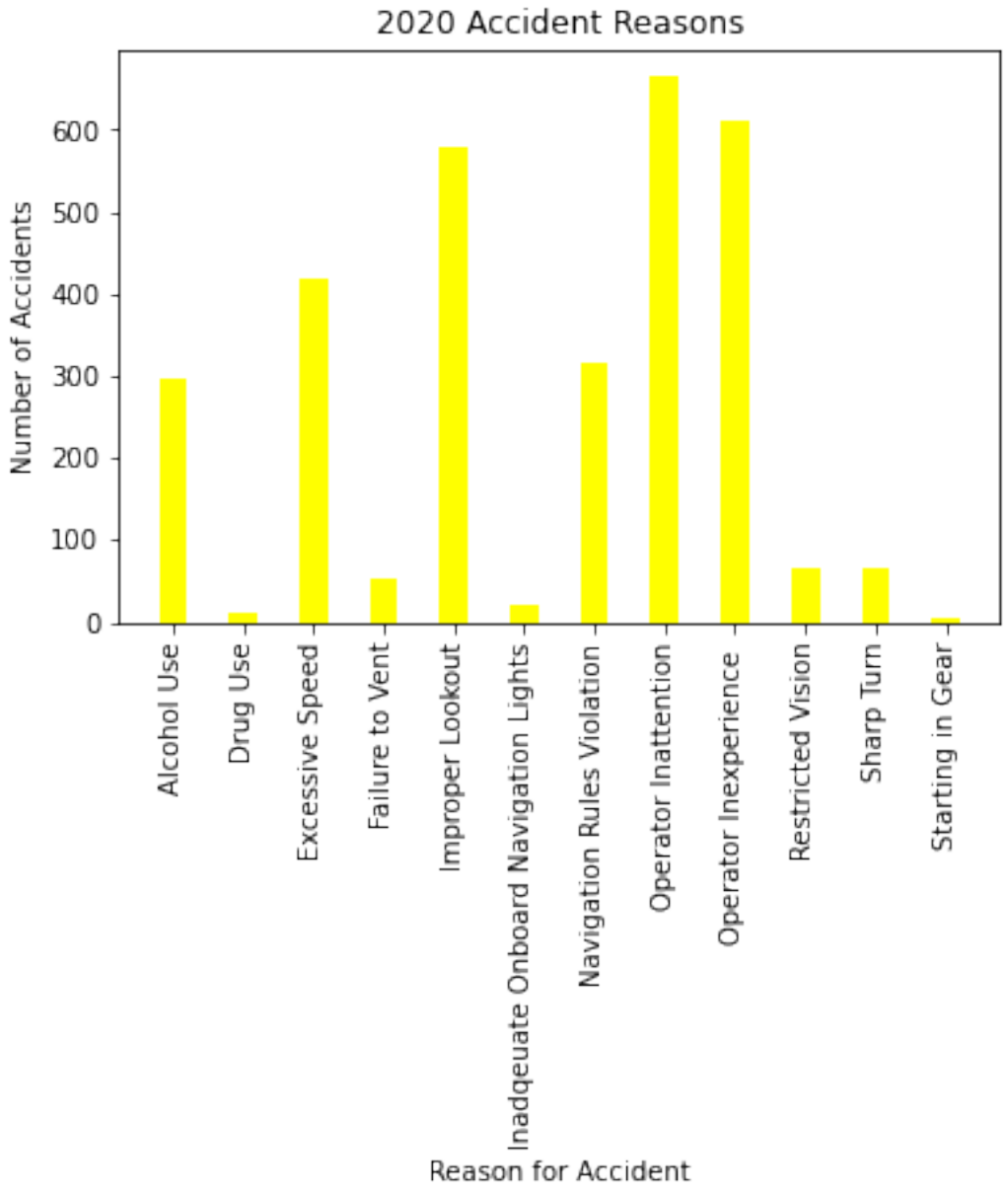
	Crash Reason	Accidents	Deaths	Injuries
0	Alcohol Use	296	115	260
1	Drug Use	12	8	2
2	Excessive Speed	418	32	345
3	Failure to Vent	54	5	89
4	Improper Lookout	578	28	409
5	Inadqeuate Onboard Navigation Lights	21	0	16
6	Navigation Rules Violation	316	26	220
7	Operator Inattention	664	55	383
8	Operator Inexperience	612	56	343
9	Restricted Vision	67	2	47

10	Sharp Turn	67	7	66
11	Starting in Gear	4	0	2

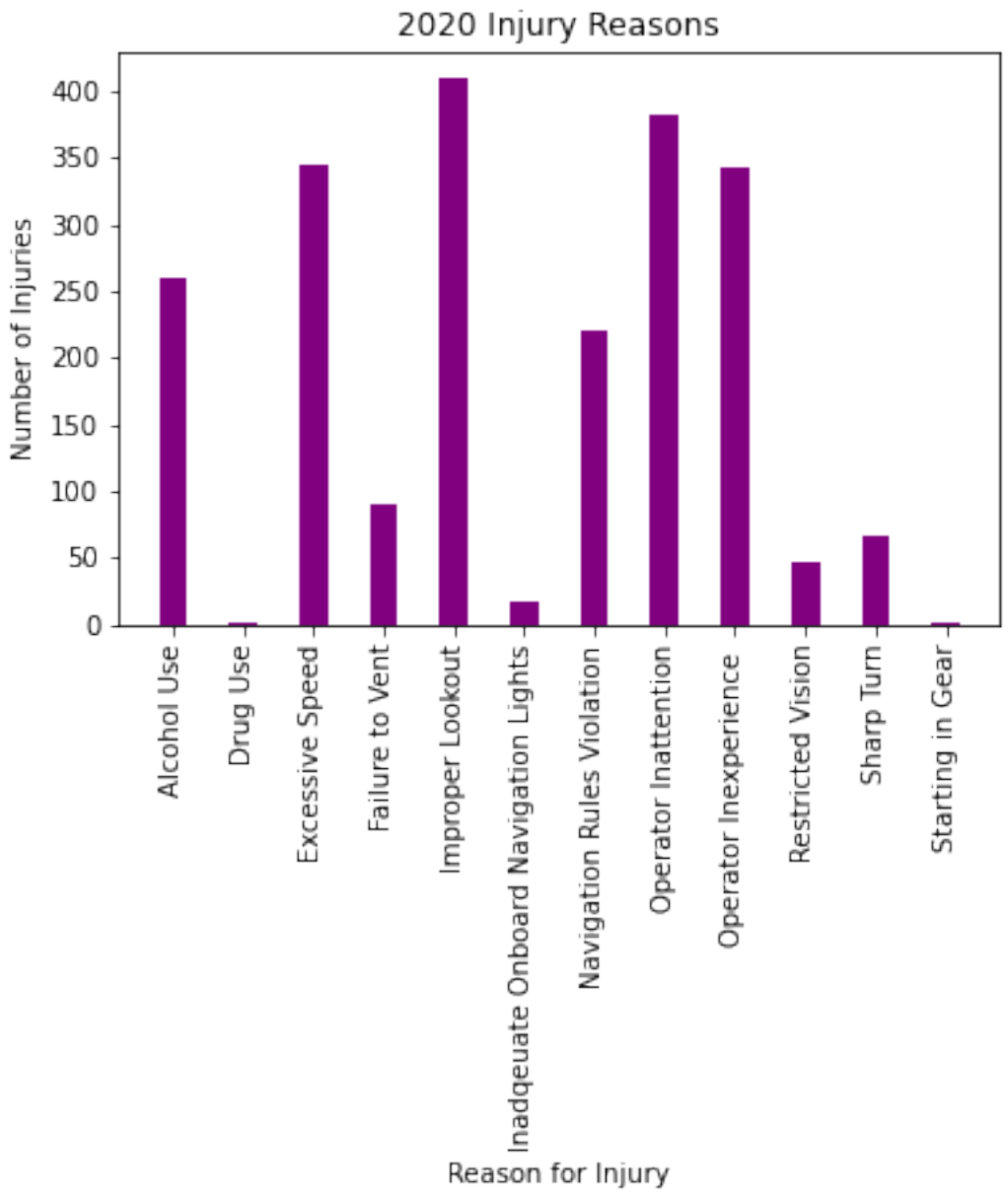


[22]:
```
plt.bar(Factors['Crash Reason'], Factors['Accidents'], color ='yellow',
        width = 0.4)

plt.xlabel("Reason for Accident")
plt.ylabel("Number of Accidents")
plt.title("2020 Accident Reasons")
plt.xticks(rotation=90)
```

```
plt.show()
```



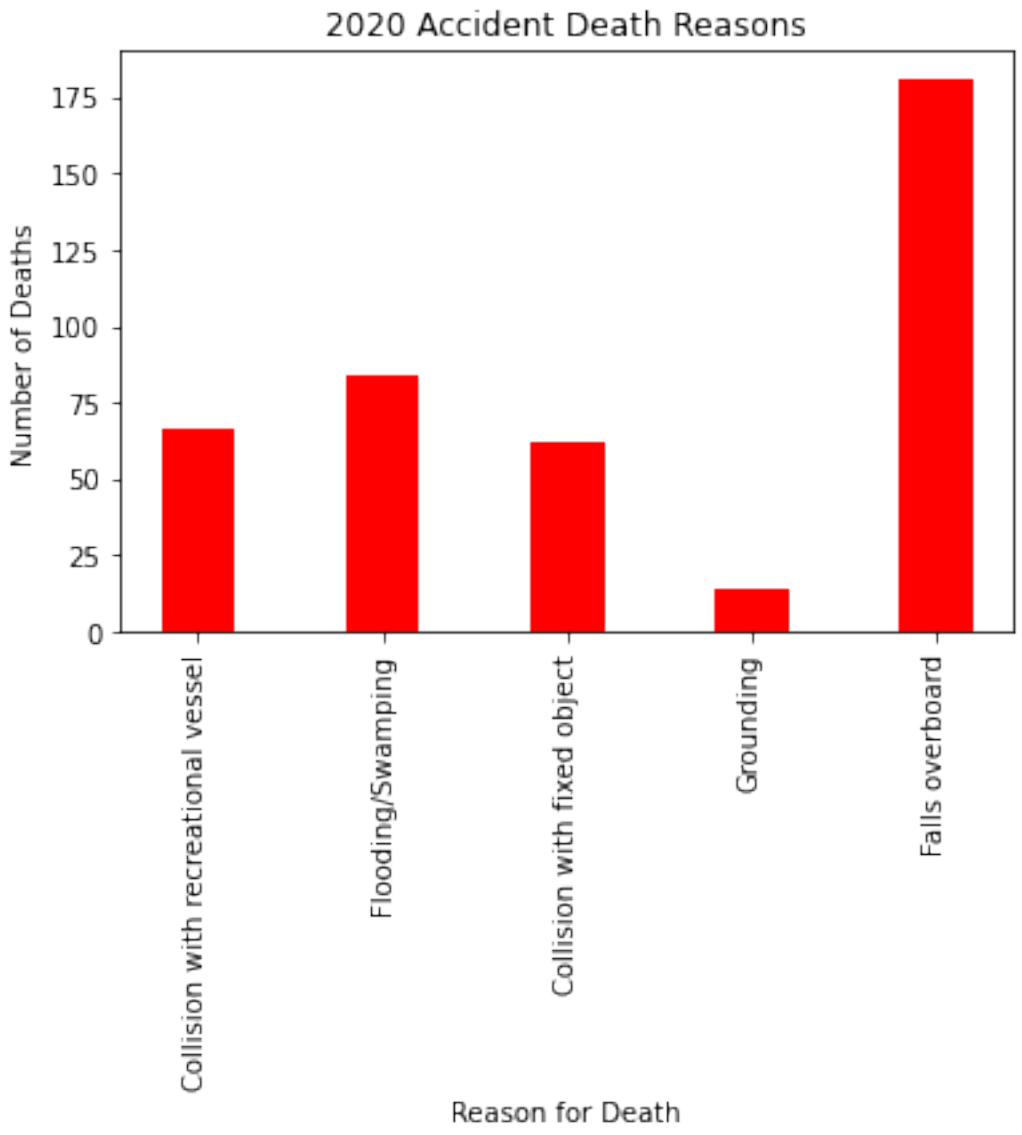
[58]:
```
plt.bar(Factors['Crash Reason'], Factors['Injuries'], color ='purple',
        width = 0.4)

plt.xlabel("Reason for Injury")
plt.ylabel("Number of Injuries")
plt.title("2020 Injury Reasons")
plt.xticks(rotation=90)
plt.show()
```



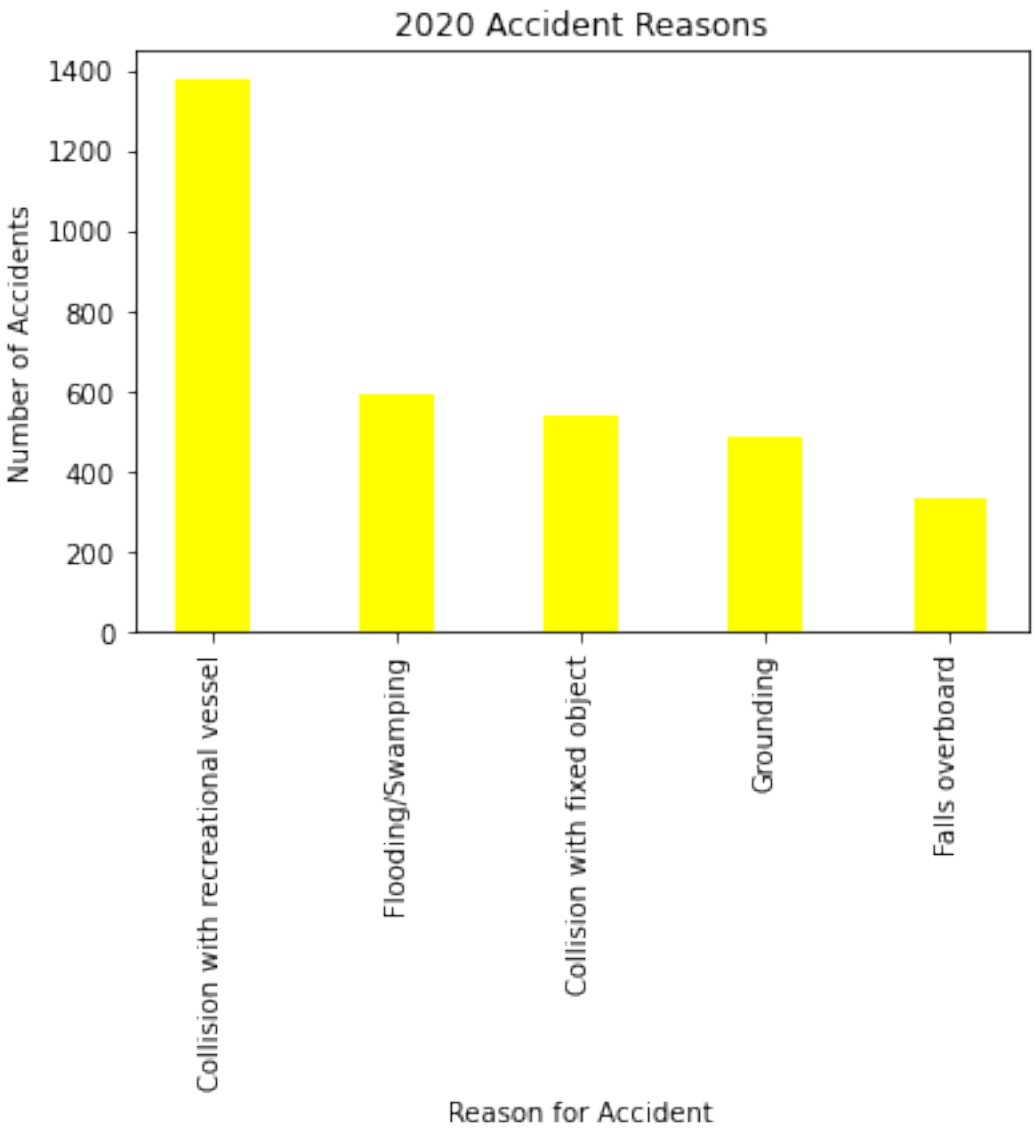
[59]:
```
#Type of Accidents that occurred in 2020#
print(AccidentType)

plt.bar(AccidentType['Accident Type'], AccidentType['Number of Deaths'], color␣
 ↪='red',
        width = 0.4)

plt.xlabel("Reason for Death")
plt.ylabel("Number of Deaths")
plt.title("2020 Accident Death Reasons")
```

```
plt.xticks(rotation=90)
plt.show()
```

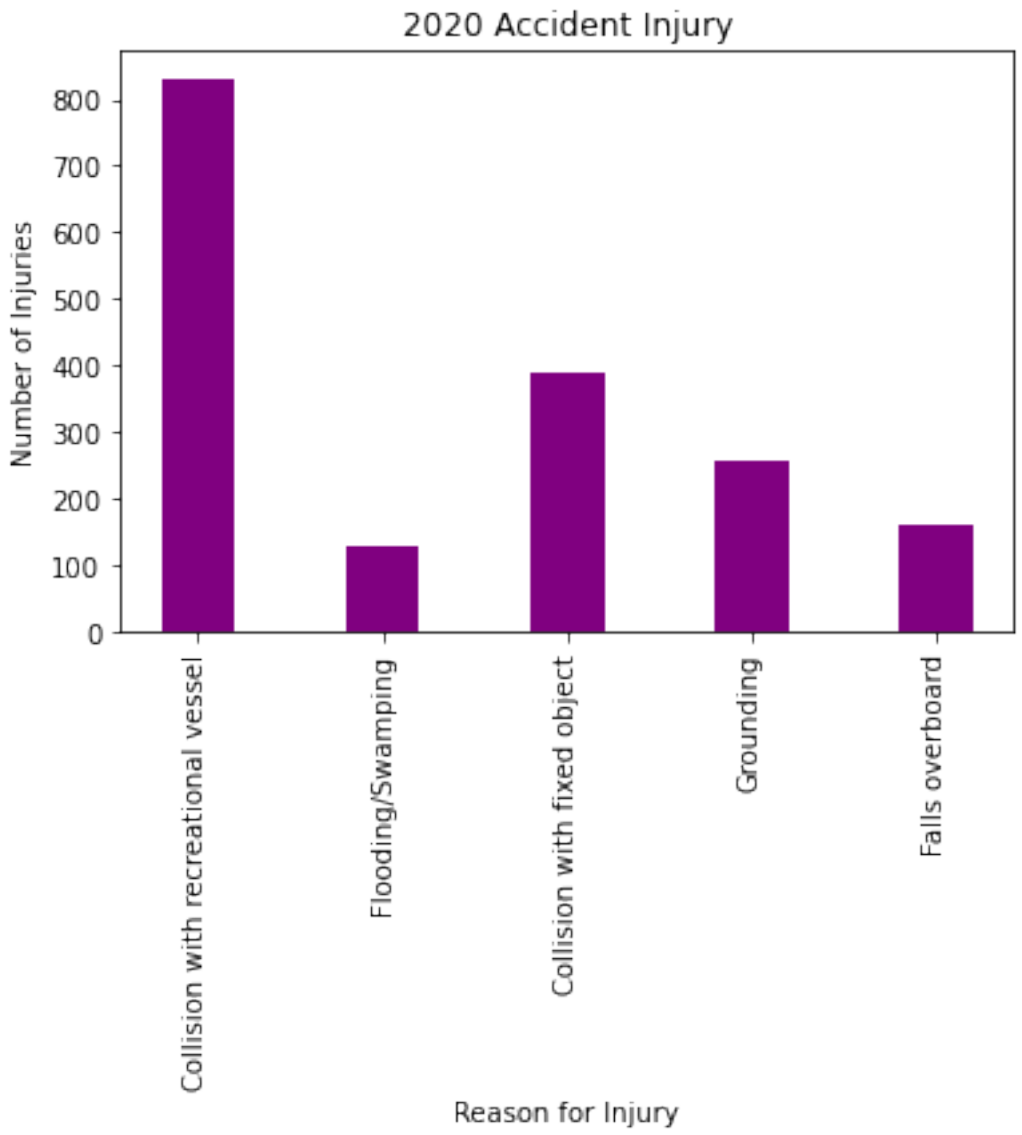
```
                      Accident Type  Number of Accidents  Number of Deaths  \
0  Collision with recreational vessel                 1379                66
1                   Flooding/Swamping                  589                84
2         Collision with fixed object                  542                62
3                            Grounding                  484                14
4                     Falls overboard                  335               181

   Number of Injuries
0                 831
1                 128
2                 389
3                 255
4                 161
```



[60]:
```
plt.bar(AccidentType['Accident Type'], AccidentType['Number of Accidents'],␣
 ↪color ='yellow',
        width = 0.4)

plt.xlabel("Reason for Accident")
plt.ylabel("Number of Accidents")
plt.title("2020 Accident Reasons")
plt.xticks(rotation=90)
plt.show()
```



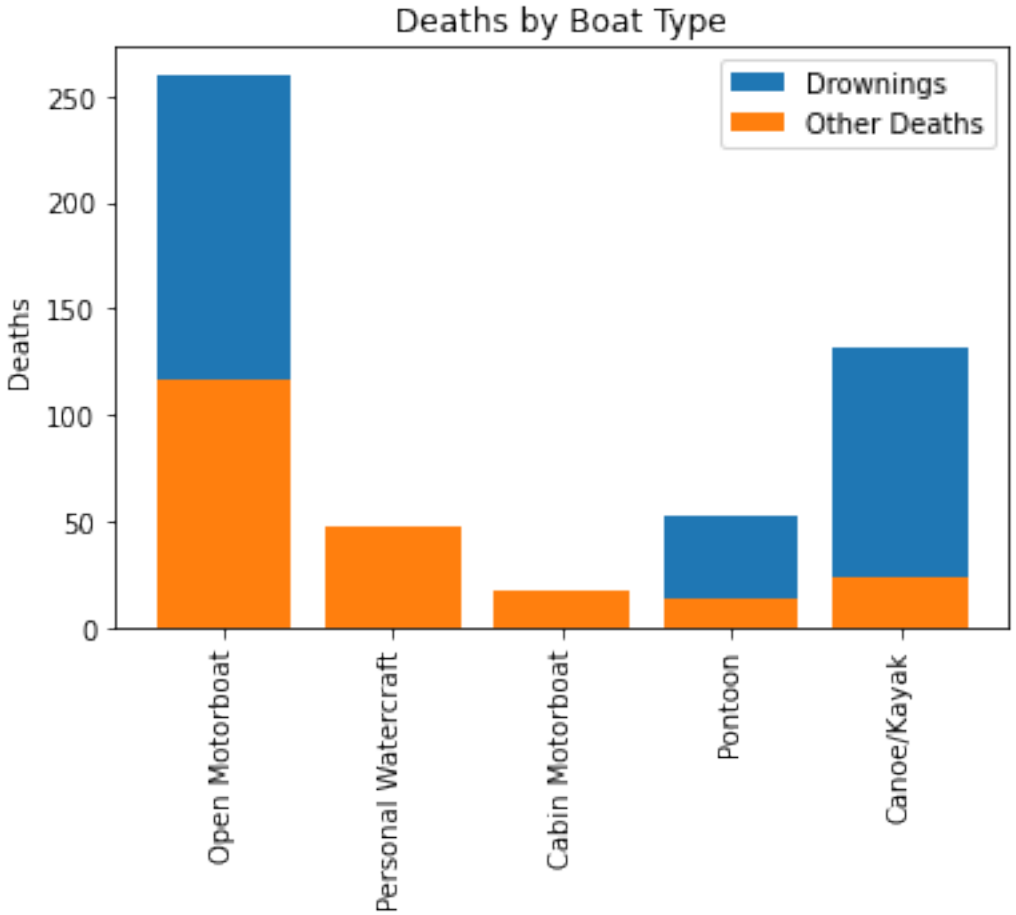
[61]:
```
plt.bar(AccidentType['Accident Type'], AccidentType['Number of Injuries'],␣
 ↪color ='purple',
        width = 0.4)

plt.xlabel("Reason for Injury")
plt.ylabel("Number of Injuries")
plt.title("2020 Accident Injury")
plt.xticks(rotation=90)
plt.show()
```



[63]:
```
#Vessel of Accidents that occurred in 2020#
print(VesselAccidents)
fig, ax = plt.subplots()

ax.bar(VesselAccidents['Type of Boat'], VesselAccidents['Drownings'],␣
 ↪label='Drownings')
ax.bar(VesselAccidents['Type of Boat'], VesselAccidents['Other Deaths'],␣
 ↪label='Other Deaths')

ax.set_ylabel('Deaths')
ax.set_title('Deaths by Boat Type')
ax.legend()
```

```
plt.xticks(rotation=90)

plt.show()
```

	Type of Boat	Drownings	Other Deaths	Total Deaths	Total Injuries
0	Open Motorboat	260	116	376	1520
1	Personal Watercraft	19	47	66	896
2	Cabin Motorboat	16	17	33	298
3	Pontoon	53	14	67	232
4	Canoe/Kayak	131	23	154	117



[ ]: