Visualization Code

In [56]:
```
import pandas as pd
```

This loads dataset to notebook

In [57]:
```
Data = pd.read_csv("Austin_Animal_Shelter_Dataset.csv")
Data
```

Out[57]:

	Animal ID	Name	Date of Birth	Intake Date	Intake Time	Intake Age in Days	Intake Age in Weeks	Intake Age in Months	Intake Age in Years	Outc I
0	A786884	Brock	1/3/2017	1/3/2019	16:19	730	104.29	24	2	1/8/2
1	A706918	Belle	7/5/2007	7/5/2015	12:59	2922	417.43	96	8	7/5/2
2	A724273	Runster	4/17/2015	4/14/2016	18:43	363	51.86	11	0	4/21/2
3	A665644	NaN	9/21/2013	10/21/2013	7:59	30	4.29	1	0	10/21/2
4	A682524	Rio	6/29/2010	6/29/2014	10:38	1461	208.71	48	4	7/2/2
...	...	...	...	...	...	...	...	...	...	
31311	A715363	Grace	9/4/2015	1/25/2016	14:02	143	20.43	4	0	11/17/2
31312	A740454	Peanutbutte	12/17/2015	12/17/2016	13:52	366	52.29	12	1	12/22/2
31313	A735880	Willy	7/15/2016	2/5/2017	16:27	205	29.29	6	0	2/9/2
31314	A802992	Cederic	8/28/2010	8/25/2019	16:37	3284	469.14	107	8	9/1/2
31315	A738636	Frijola	9/3/2016	11/18/2016	16:42	76	10.86	2	0	11/20/2

31316 rows × 28 columns

In [58]:

```
animal_df = pd.DataFrame(Data, columns=['Animal ID', 'Date of Birth', 'Intake Age
animal_df
```

Out[58]:

	Animal ID	Date of Birth	Intake Age in Months	Intake Date	Outcome Date	Outcome Age in Years	Length of Stay in Days	Animal Type	Sex a Outcom
0	A786884	1/3/2017	24	1/3/2019	1/8/2019	2	5.0	Dog	Neutere Mal
1	A706918	7/5/2007	96	7/5/2015	7/5/2015	8	0.0	Dog	Spaye Femal
2	A724273	4/17/2015	11	4/14/2016	4/21/2016	1	7.0	Dog	Neutere Mal
3	A665644	9/21/2013	1	10/21/2013	10/21/2013	0	0.0	Cat	Intac Femal
4	A682524	6/29/2010	48	6/29/2014	7/2/2014	4	3.0	Dog	Neutere Mal
...	...	...	...	...	...	...	...	...	.
31311	A715363	9/4/2015	4	1/25/2016	11/17/2015	0	-69.0	Cat	Spaye Femal
31312	A740454	12/17/2015	12	12/17/2016	12/22/2016	1	5.0	Dog	Neutere Mal
31313	A735880	7/15/2016	6	2/5/2017	2/9/2017	0	4.0	Cat	Neutere Mal
31314	A802992	8/28/2010	107	8/25/2019	9/1/2019	9	7.0	Dog	Neutere Mal
31315	A738636	9/3/2016	2	11/18/2016	11/20/2016	0	2.0	Cat	Neutere Mal

31316 rows × 10 columns

In [59]:

```
print(animal_df.loc[animal_df['Length of Stay in Days']<0].count())
```

```
Animal ID                 1706
Date of Birth             1706
Intake Age in Months      1706
Intake Date               1706
Outcome Date              1706
Outcome Age in Years      1706
Length of Stay in Days    1706
Animal Type               1706
Sex at Outcome            1706
Outcome                   1706
dtype: int64
```

In [60]:

```
validStay_df = animal_df.loc[(animal_df['Length of Stay in Days']>=0) & (animal_
validStay_df
```

Out[60]:

	Animal ID	Date of Birth	Intake Age in Months	Intake Date	Outcome Date	Outcome Age in Years	Length of Stay in Days	Animal Type	Sex at Outcome
0	A786884	1/3/2017	24	1/3/2019	1/8/2019	2	5.0	Dog	Neutered Male
1	A706918	7/5/2007	96	7/5/2015	7/5/2015	8	0.0	Dog	Spayed Female
2	A724273	4/17/2015	11	4/14/2016	4/21/2016	1	7.0	Dog	Neutered Male
3	A665644	9/21/2013	1	10/21/2013	10/21/2013	0	0.0	Cat	Intact Female
4	A682524	6/29/2010	48	6/29/2014	7/2/2014	4	3.0	Dog	Neutered Male
...	...	...	...	...	...	...	...	...	...
31310	A706749	1/2/2014	18	7/2/2015	7/3/2015	1	1.0	Cat	Intact Male
31312	A740454	12/17/2015	12	12/17/2016	12/22/2016	1	5.0	Dog	Neutered Male
31313	A735880	7/15/2016	6	2/5/2017	2/9/2017	0	4.0	Cat	Neutered Male
31314	A802992	8/28/2010	107	8/25/2019	9/1/2019	9	7.0	Dog	Neutered Male
31315	A738636	9/3/2016	2	11/18/2016	11/20/2016	0	2.0	Cat	Neutered Male

29594 rows × 10 columns

In [61]:
```
cat_df = validStay_df.loc[animal_df['Animal Type']=='Cat']
cat_df
```

Out[61]:

	Animal ID	Date of Birth	Intake Age in Months	Intake Date	Outcome Date	Outcome Age in Years	Length of Stay in Days	Animal Type	Sex at Outcome
3	A665644	9/21/2013	1	10/21/2013	10/21/2013	0	0.0	Cat	Intact Female
8	A818975	5/19/2020	0	6/18/2020	7/23/2020	0	35.0	Cat	Neutered Male
9	A774147	5/10/2018	1	6/11/2018	6/11/2018	0	0.0	Cat	Intact Female
10	A731435	2/22/2016	5	8/8/2016	8/13/2016	0	5.0	Cat	Neutered Male
14	A790209	3/6/2015	48	3/6/2019	3/13/2019	4	7.0	Cat	Intact Female
...	...	...	...	...	...	...	...	...	...
31306	A738837	11/22/2015	12	11/22/2016	1/29/2017	1	68.0	Cat	Spayed Female
31308	A811903	1/12/2009	131	1/11/2020	1/12/2020	11	1.0	Cat	Spayed Female
31310	A706749	1/2/2014	18	7/2/2015	7/3/2015	1	1.0	Cat	Intact Male
31313	A735880	7/15/2016	6	2/5/2017	2/9/2017	0	4.0	Cat	Neutered Male
31315	A738636	9/3/2016	2	11/18/2016	11/20/2016	0	2.0	Cat	Neutered Male

11551 rows × 10 columns

In [62]:
```
dog_df = validStay_df.loc[animal_df['Animal Type']=='Dog']
dog_df
```

Out[62]:

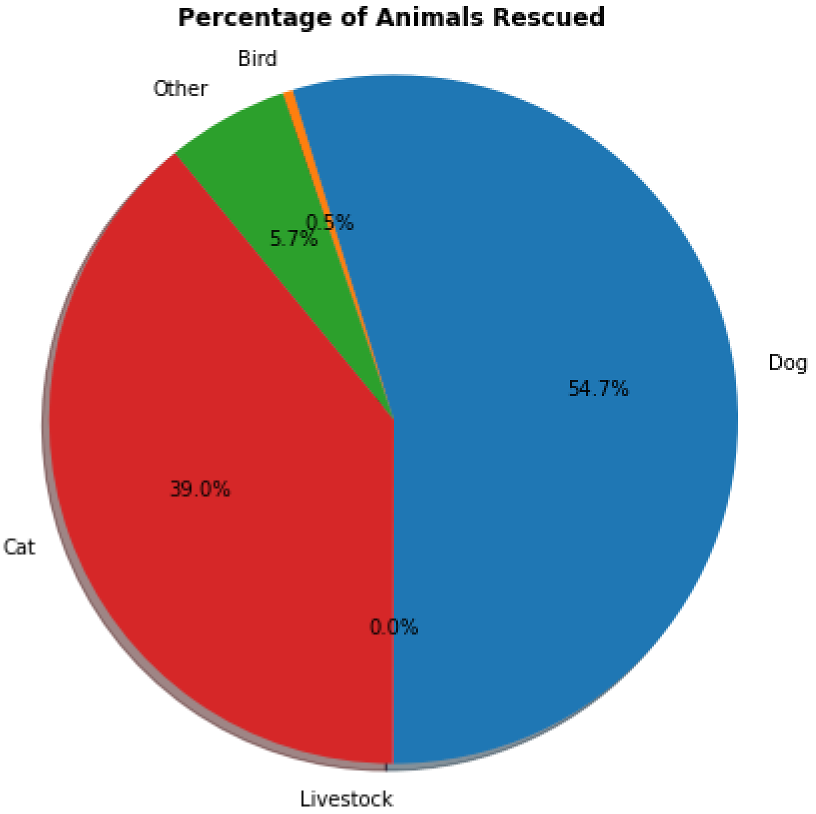
	Animal ID	Date of Birth	Intake Age in Months	Intake Date	Outcome Date	Outcome Age in Years	Length of Stay in Days	Animal Type	Sex at Outcome
0	A786884	1/3/2017	24	1/3/2019	1/8/2019	2	5.0	Dog	Neutered Male
1	A706918	7/5/2007	96	7/5/2015	7/5/2015	8	0.0	Dog	Spayed Female
2	A724273	4/17/2015	11	4/14/2016	4/21/2016	1	7.0	Dog	Neutered Male
4	A682524	6/29/2010	48	6/29/2014	7/2/2014	4	3.0	Dog	Neutered Male
5	A743852	2/18/2015	24	2/18/2017	2/21/2017	2	3.0	Dog	Neutered Male
...	...	...	...	...	...	...	...	...	...
31305	A750215	5/24/2014	36	5/24/2017	5/24/2017	3	0.0	Dog	Intact Male
31307	A625175	4/21/2011	36	5/19/2014	5/24/2014	3	5.0	Dog	Neutered Male
31309	A691935	11/12/2011	36	11/12/2014	12/9/2014	3	27.0	Dog	Neutered Male
31312	A740454	12/17/2015	12	12/17/2016	12/22/2016	1	5.0	Dog	Neutered Male
31314	A802992	8/28/2010	107	8/25/2019	9/1/2019	9	7.0	Dog	Neutered Male

16198 rows × 10 columns

In [63]:
```
#bird_df = validStay_df.loc[animal_df['AnimalType']=='Bird']
#bird_df
#other_df = validStay_df.loc[animal_df['AnimalType']=='Other']
#other_df
#  validStay_df.loc[(animal_df['AnimalType']!='Cat') & (animal_df['AnimalType']!=
#livestock_df = validStay_df.loc[animal_df['AnimalType']=='Livestock']
#livestock_df
```

In [64]:

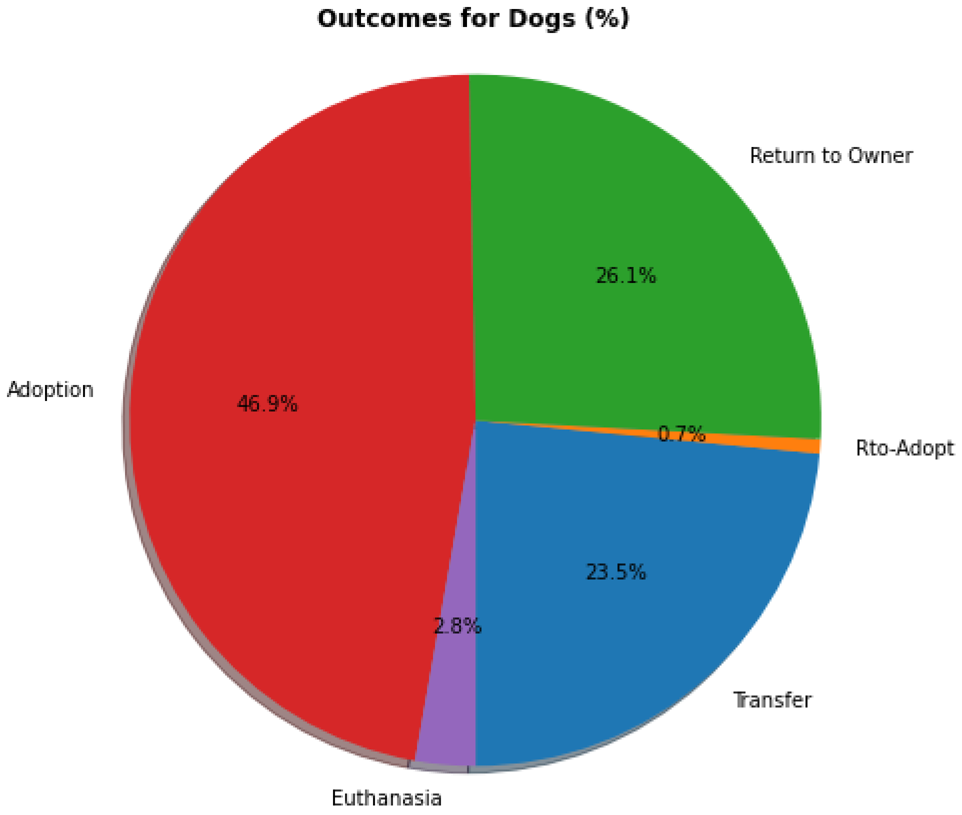
```
#Visualization of Animal Types
from matplotlib import pyplot as plt
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
import numpy as np
animals = ['Dog', 'Bird', 'Other' ,'Cat', 'Livestock']
data = [16198,142,1697,11551,6]
fig = plt.figure(figsize = (10,7))
plt.pie(data,labels=animals, autopct='%1.1f%%',shadow=True,startangle=270)
plt.title('Percentage of Animals Rescued',fontweight='bold')
plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```



In [65]:
```
#Outcome of dogs
dog_outcome_list = dog_df.Outcome.unique().tolist()
dog_cnt_list = []
for e in dog_outcome_list:
    cnt = len(dog_df[dog_df['Outcome'] == e].index)
    #print(e, cnt )
    #if cnt > 100:
    dog_cnt_list.append(cnt)
    #else:
        #dog_outcome_list.remove(e)
print(dog_outcome_list)
print(dog_cnt_list)
```

```
['Transfer', 'Return to Owner', 'Adoption', 'Died', 'Euthanasia', 'Rto-Adopt',
'Missing', 'Disposal']
[3781, 4213, 7561, 65, 450, 109, 9, 10]
```

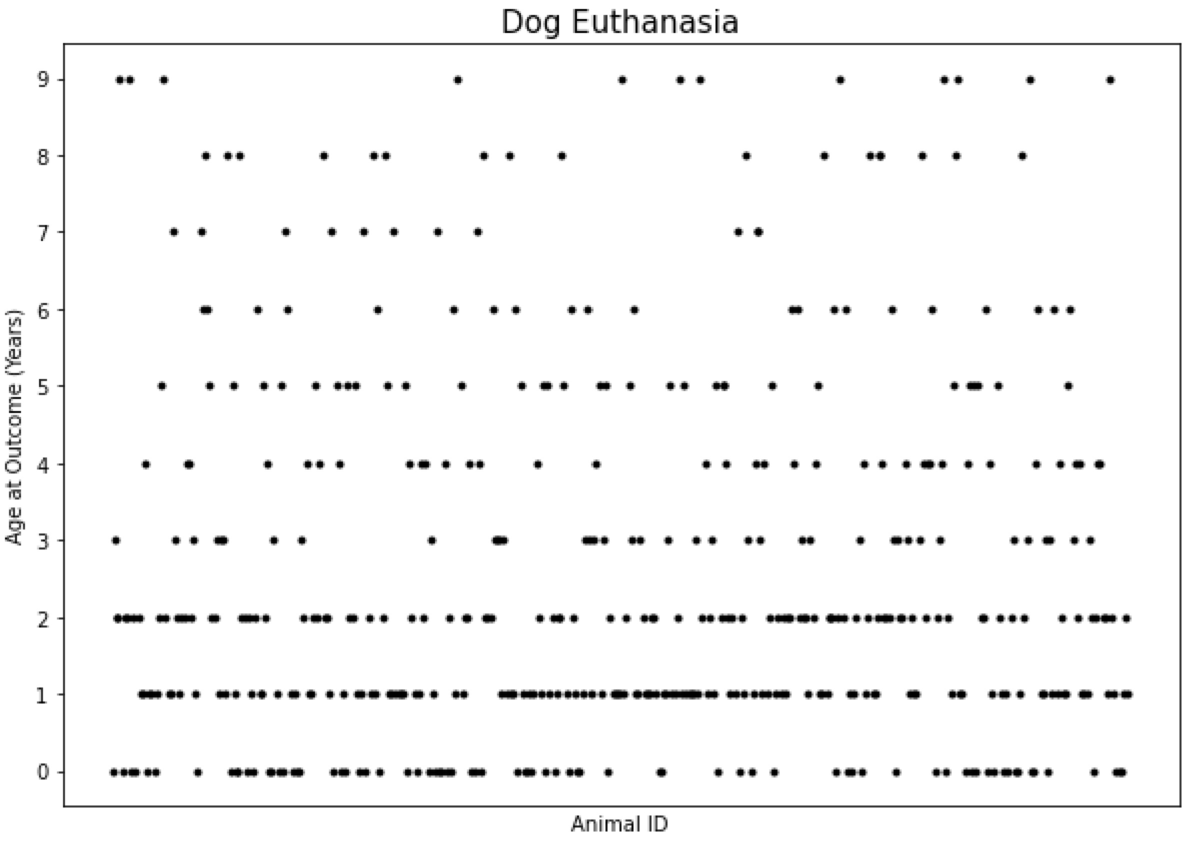
In [66]:
```
datay = [3781, 109, 4213, 7561, 450]
datax = ['Transfer', 'Rto-Adopt','Return to Owner', 'Adoption', 'Euthanasia']

fig = plt.figure(figsize = (10,7))
plt.pie(datay,labels=datax, autopct='%1.1f%%',shadow=True,startangle=270)
plt.title('Outcomes for Dogs (%)',fontweight='bold')
plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```



In [67]:
```
dog_euthanasia = validStay_df.loc[(animal_df['Animal Type']=='Dog') & (animal_df
#dog_euthanasia
```

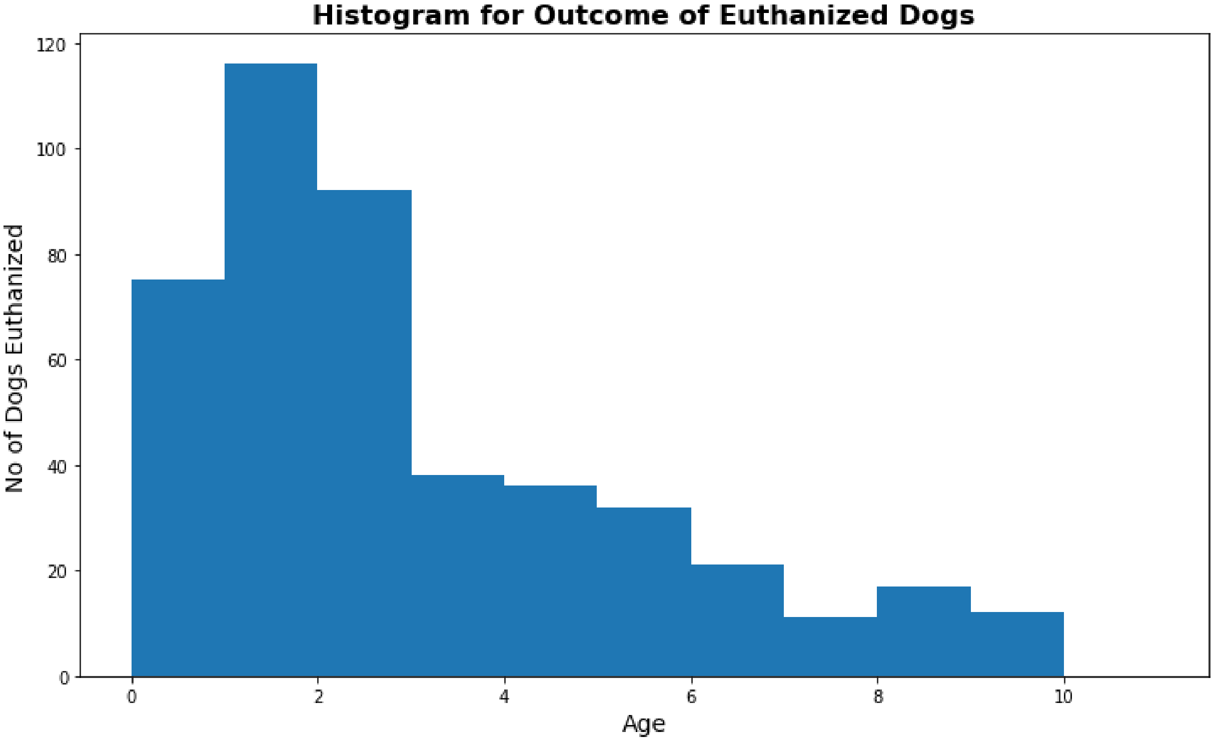
In [68]:

```
yaxis = dog_euthanasia['Outcome Age in Years'].to_numpy()
xaxis = dog_euthanasia['Animal ID'].tolist()
ydata = [float(da[0]) for da in yaxis]

fig = plt.figure(figsize = (10,7))
plt.xlabel('Animal ID')
plt.ylabel('Age at Outcome (Years)')
plt.title('Dog Euthanasia', fontsize = 15)
plt.plot(xaxis,ydata, '.', color = 'black')
plt.tick_params(axis='x', which='both', bottom=False, top=False, labelbottom=Fals
#plt.xticks(rotation = 90, fontsize = 6)
plt.yticks(np.arange(0, 10, 1.0))
plt.show()
```

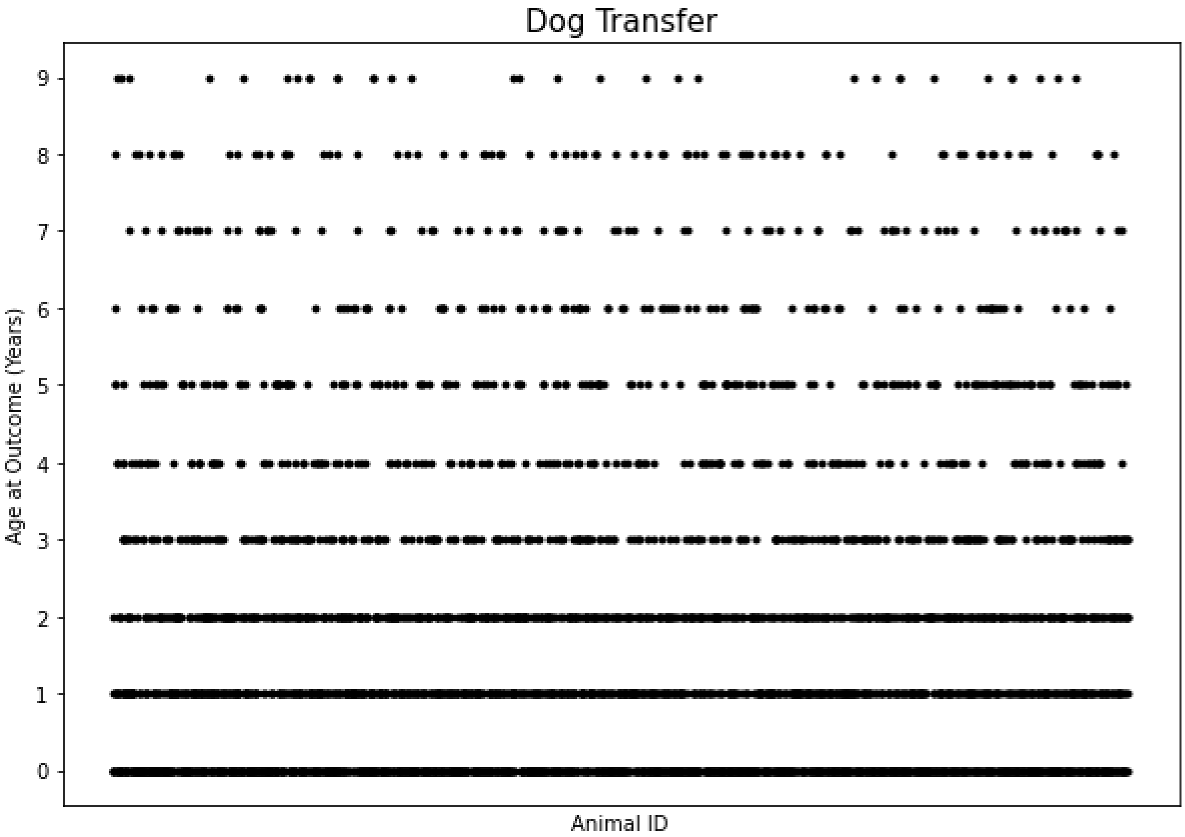


In [69]:

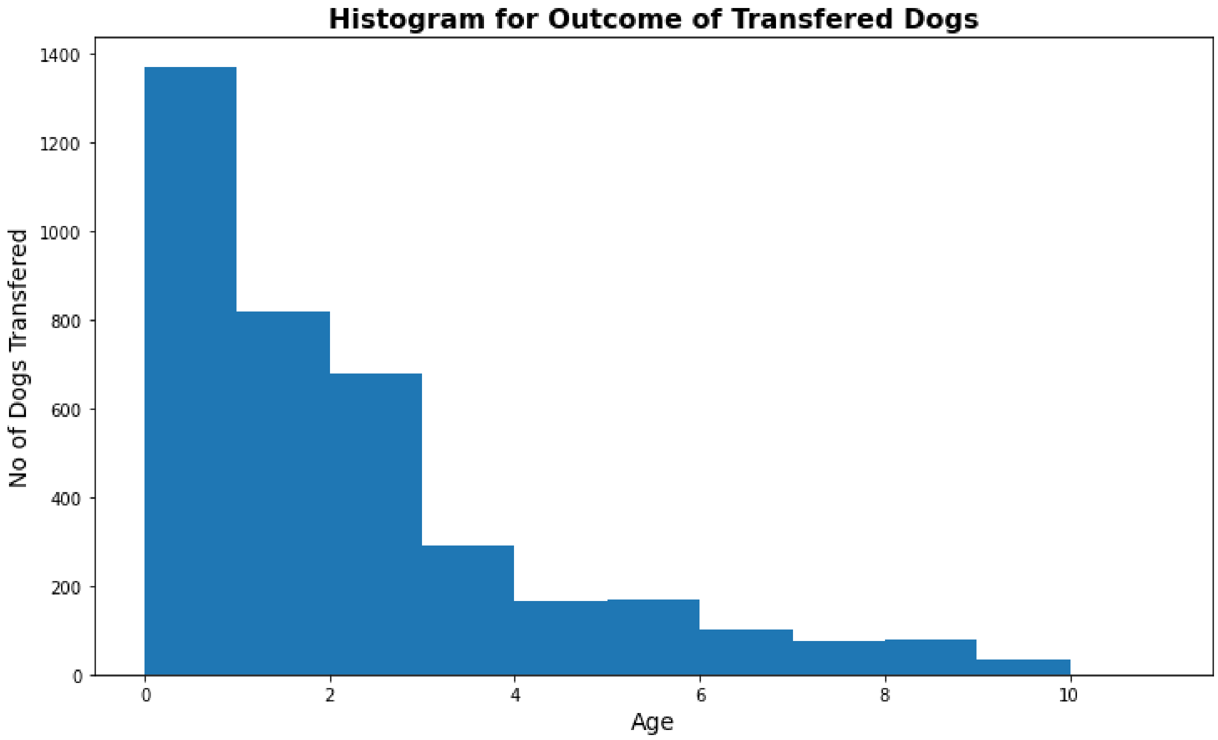
```
ydata = [float(da[0]) for da in yaxis]
n, bins, patches = plt.hist(ydata, bins = [0,1,2,3,4,5,6,7,8,9,10,11])
plt.title('Histogram for Outcome of Euthanized Dogs', fontweight= 'bold', fontsiz
plt.xlabel('Age',fontsize=14)
plt.rcParams["figure.figsize"] = (12,7)
plt.ylabel('No of Dogs Euthanized',fontsize=14)
plt.show()
```



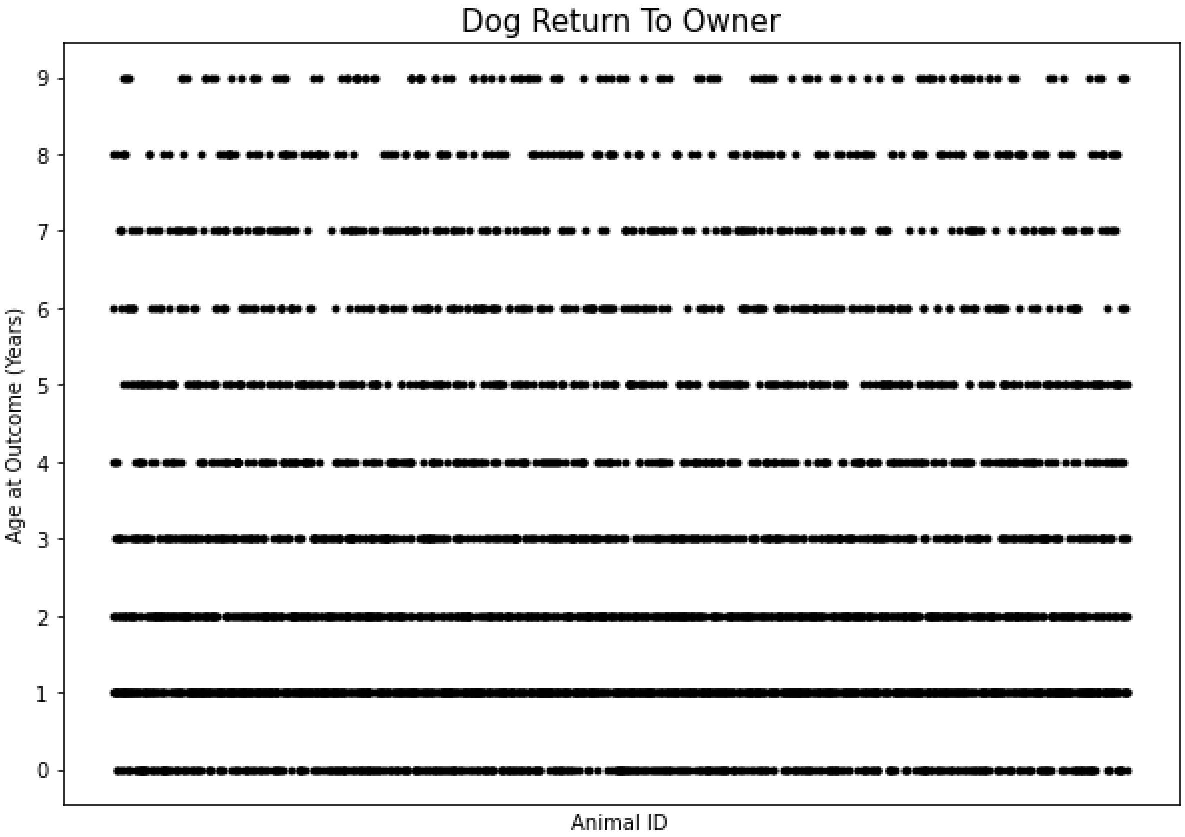
```
In [70]: dog_transfer = validStay_df.loc[(animal_df['Animal Type']=='Dog') & (animal_df['O
         yaxis = dog_transfer['Outcome Age in Years'].to_numpy()
         xaxis = dog_transfer['Animal ID'].tolist()
         ydata = [float(da[0]) for da in yaxis]

         fig = plt.figure(figsize = (10,7))
         plt.xlabel('Animal ID')
         plt.ylabel('Age at Outcome (Years)')
         plt.title('Dog Transfer',fontsize = 15)
         plt.plot(xaxis,ydata, '.', color = 'black')
         plt.tick_params(axis='x', which='both', bottom=False, top=False, labelbottom=Fals
         #plt.xticks(rotation = 90, fontsize = 6)
         plt.yticks(np.arange(0, 10, 1.0))
         plt.show()
```



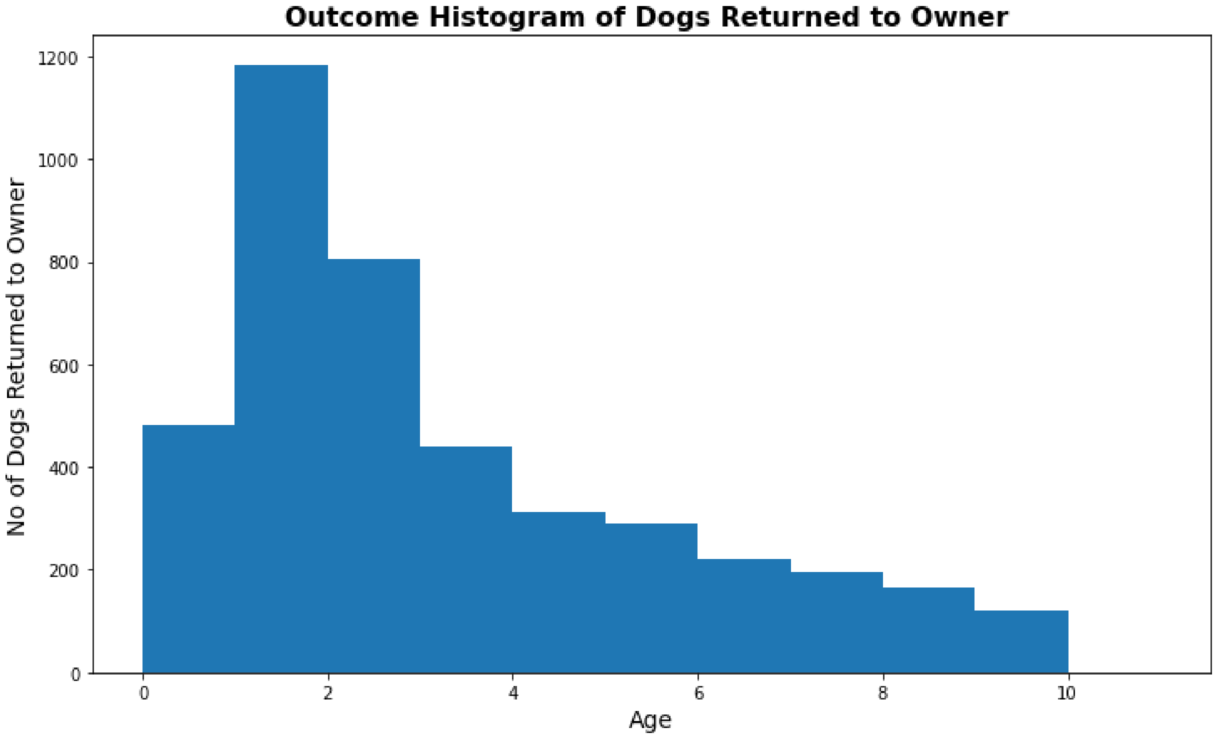
In [71]:
```
ydata = [float(da[0]) for da in yaxis]
n, bins, patches = plt.hist(ydata, bins = [0,1,2,3,4,5,6,7,8,9,10,11])
plt.title('Histogram for Outcome of Transfered Dogs', fontweight= 'bold', fontsiz
plt.xlabel('Age',fontsize=14)
plt.rcParams["figure.figsize"] = (12,7)
plt.ylabel('No of Dogs Transfered',fontsize=14)
plt.show()
```



In [72]:
```
dog_returntoowner = validStay_df.loc[(animal_df['Animal Type']=='Dog') & (animal_
yaxis = dog_returntoowner['Outcome Age in Years'].to_numpy()
xaxis = dog_returntoowner['Animal ID'].tolist()
ydata = [float(da[0]) for da in yaxis]

fig = plt.figure(figsize = (10,7))
plt.xlabel('Animal ID')
plt.ylabel('Age at Outcome (Years)')
plt.title('Dog Return To Owner',fontsize = 15)
plt.plot(xaxis,ydata, '.', color = 'black')
plt.tick_params(axis='x', which='both', bottom=False, top=False, labelbottom=Fals
#plt.xticks(rotation = 90, fontsize = 6)
plt.yticks(np.arange(0, 10, 1.0))
plt.show()
```



In [73]:
```
ydata = [float(da[0]) for da in yaxis]
n, bins, patches = plt.hist(ydata, bins = [0,1,2,3,4,5,6,7,8,9,10,11])
plt.title('Outcome Histogram of Dogs Returned to Owner ', fontweight= 'bold', for
plt.xlabel('Age',fontsize=14)
plt.rcParams["figure.figsize"] = (12,7)
plt.ylabel('No of Dogs Returned to Owner',fontsize=14)
plt.show()
```



In [74]:
```
#dog_intake =  dog_df.groupby('IntakeDate')['IntakeDate'].count()
dog_intake=dog_df[dog_df['Intake Date'].str.contains('2014', regex=True, na=False
len(dog_intake.index)
```

Out[74]:
```
2618
```

In [75]:
```
dog_outcome=dog_df[dog_df['Outcome Date'].str.contains('2014', regex=True, na=Fal
len(dog_outcome.index)
```

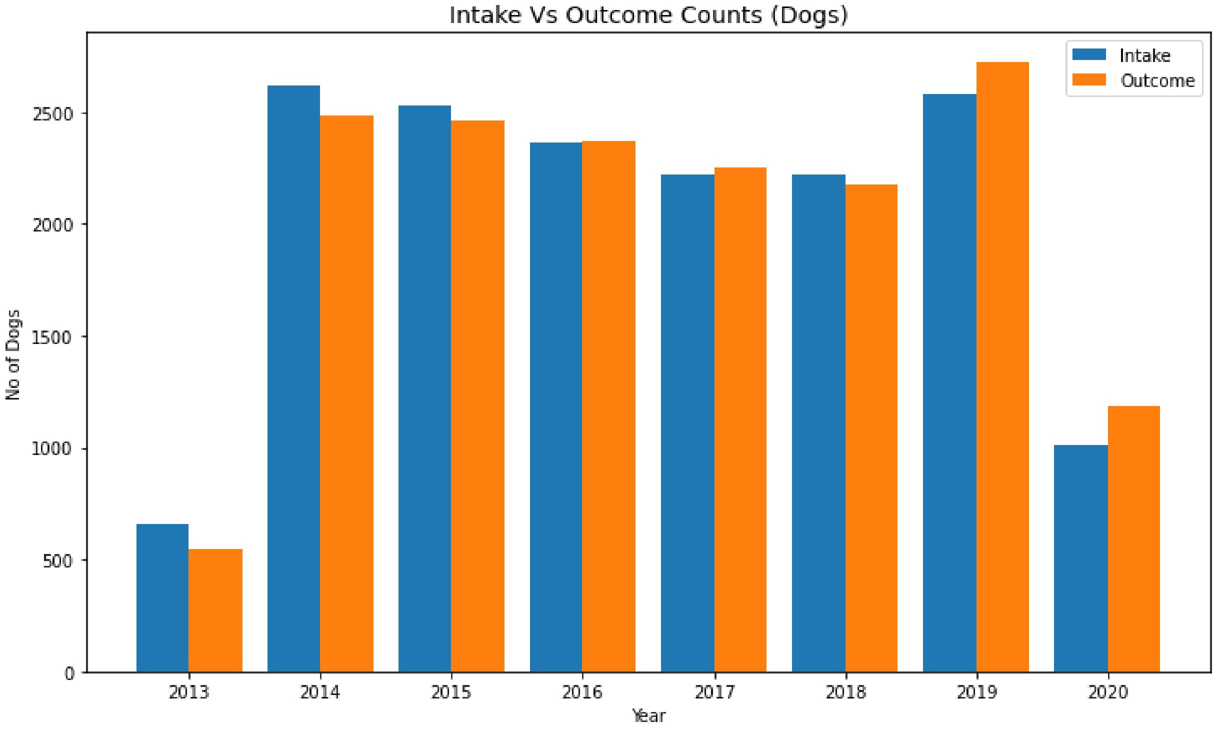
Out[75]:
```
2481
```

In [76]:

```
#Dogs intake and outcome count per year

year_list = ['2013','2014','2015','2016','2017','2018','2019','2020']
dog_intakecnt_list = []
dog_outcomecnt_list = []
for e in year_list:
    dog_intake=dog_df[dog_df['Intake Date'].str.contains( e , regex=True, na=Fals
    dog_outcome=dog_df[dog_df['Outcome Date'].str.contains( e , regex=True, na=Fa
    intakecnt = len(dog_intake.index)
    outcomecnt = len(dog_outcome.index)
    #if cnt > 100:
    dog_intakecnt_list.append(intakecnt)
    dog_outcomecnt_list.append(outcomecnt)
    #else:
        #dog_outcome_list.remove(e)
print(year_list)
print('dog intake count list is' , dog_intakecnt_list)
print('dog outcome count list is ' , dog_outcomecnt_list)
```

```
['2013', '2014', '2015', '2016', '2017', '2018', '2019', '2020']
dog intake count list is [662, 2618, 2526, 2359, 2218, 2223, 2579, 1013]
dog outcome count list is  [547, 2481, 2462, 2368, 2249, 2171, 2719, 1181]
```

In [77]:
```
x = np.arange(len(year_list))
width = 0.4
# plot data in grouped manner of bar type
plt.rcParams["figure.figsize"] = (12,7)
plt.bar(x-0.2, dog_intakecnt_list, width)
plt.bar(x+0.2, dog_outcomecnt_list, width)
plt.xticks(x, year_list)
plt.title('Intake Vs Outcome Counts (Dogs)', fontsize=14)
plt.xlabel("Year")
plt.ylabel("No of Dogs")
plt.legend(["Intake", "Outcome"])
plt.show()
```



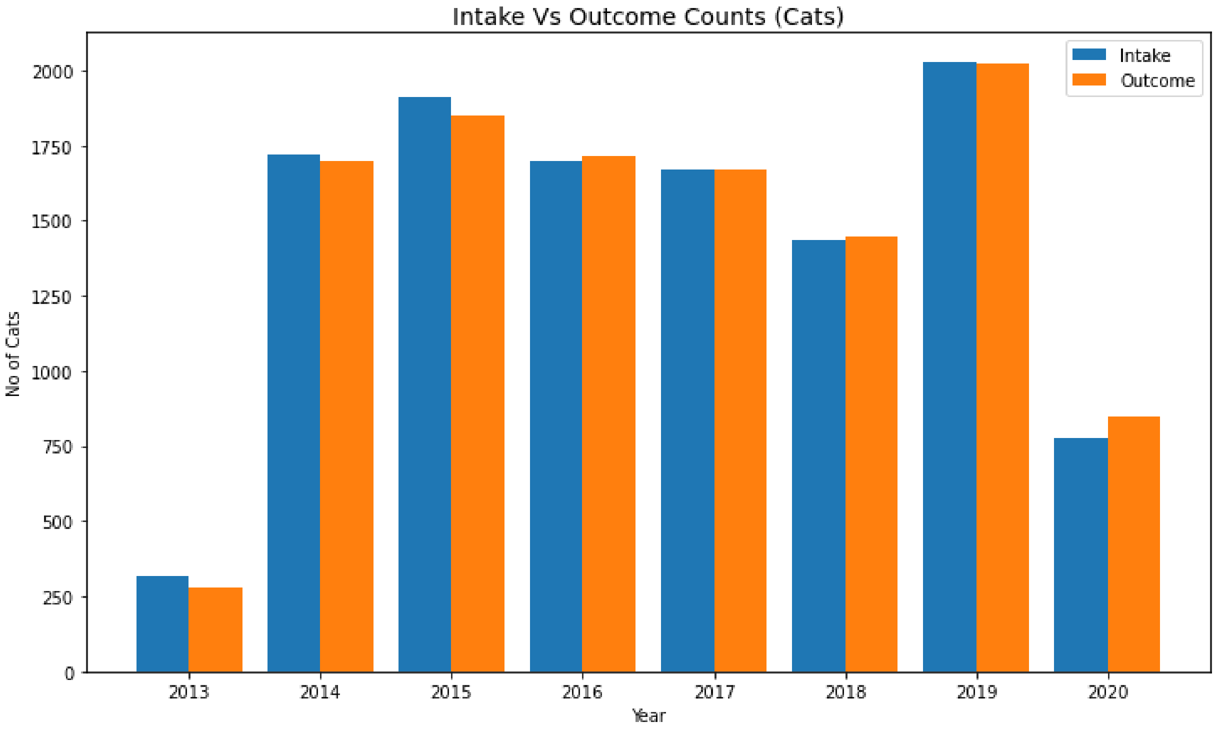
In [78]:

```
#Cats intake and outcome count per year
year_list = ['2013','2014','2015','2016','2017','2018','2019','2020']
cat_intakecnt_list = []
cat_outcomecnt_list = []
for e in year_list:
    cat_intake=cat_df[cat_df['Intake Date'].str.contains( e , regex=True, na=Fals
    cat_outcome=cat_df[cat_df['Outcome Date'].str.contains( e , regex=True, na=Fa
    intakecnt = len(cat_intake.index)
    outcomecnt = len(cat_outcome.index)
    #if cnt > 100:
    cat_intakecnt_list.append(intakecnt)
    cat_outcomecnt_list.append(outcomecnt)
    #else:
        #dog_outcome_list.remove(e)
print(year_list)
print('cat intake count list is' , cat_intakecnt_list)
print('cat outcome count list is ' , cat_outcomecnt_list)
```

```
['2013', '2014', '2015', '2016', '2017', '2018', '2019', '2020']
cat intake count list is [320, 1719, 1912, 1696, 1669, 1435, 2025, 775]
cat outcome count list is  [280, 1695, 1851, 1712, 1672, 1447, 2022, 848]
```

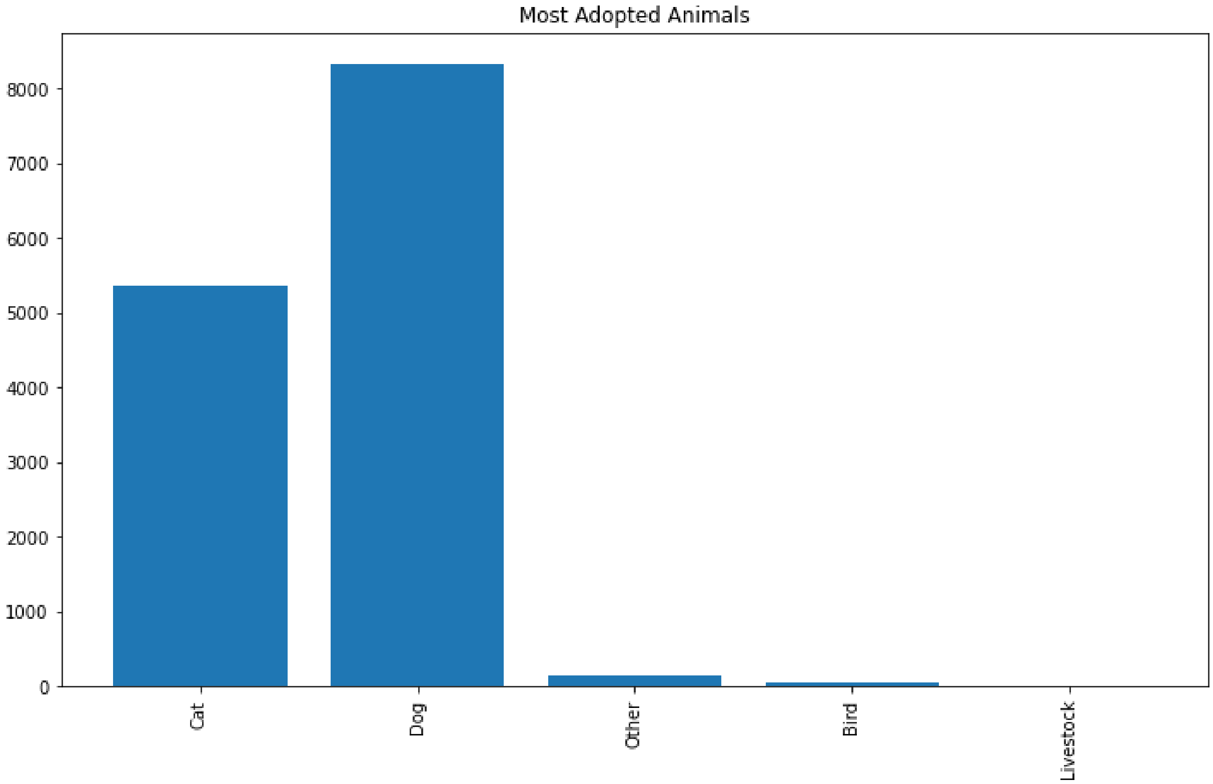
In [79]:

```
x = np.arange(len(year_list))
width = 0.4
# plot data in grouped manner of bar type
plt.rcParams["figure.figsize"] = (12,7)
plt.bar(x-0.2, cat_intakecnt_list, width)
plt.bar(x+0.2, cat_outcomecnt_list, width)
plt.xticks(x, year_list)
plt.title('Intake Vs Outcome Counts (Cats)', fontsize=14)
plt.xlabel("Year")
plt.ylabel("No of Cats")
plt.legend(["Intake", "Outcome"])
plt.show()
```



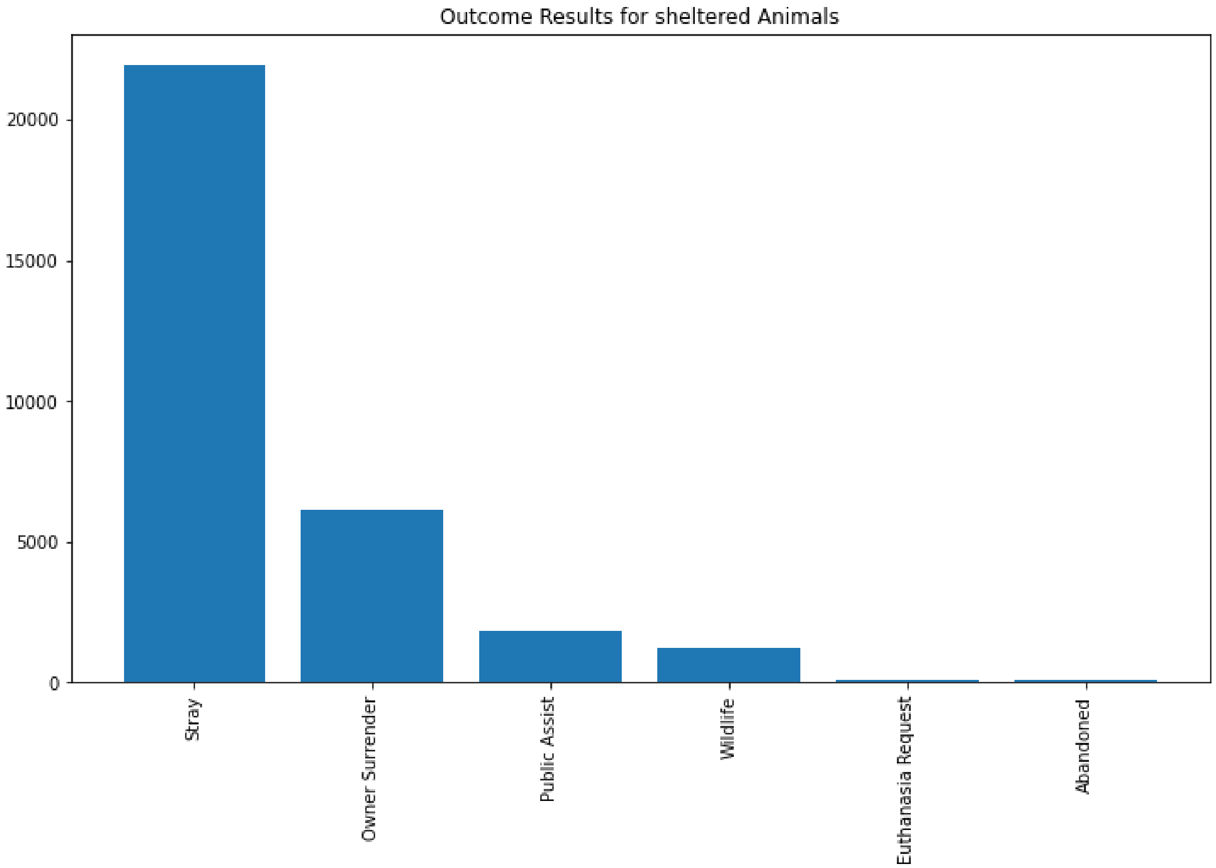
In [80]:

```
TransferedAnimalTypes = {}
AdoptedData = Data[Data["Outcome"]=="Adoption"]
for item in AdoptedData["Animal Type"]:
    if item in TransferedAnimalTypes:
        TransferedAnimalTypes[item] += 1
    else:
        TransferedAnimalTypes[item] = 1
#print(TransferedAnimalTypes)
Animalsvals = list(TransferedAnimalTypes.values())
Animalkeys = list(TransferedAnimalTypes.keys())
plt.bar(Animalkeys,Animalsvals,align = 'center')
plt.xticks(Animalkeys,rotation='90')
plt.title("Most Adopted Animals")
plt.show()
```



In [81]:

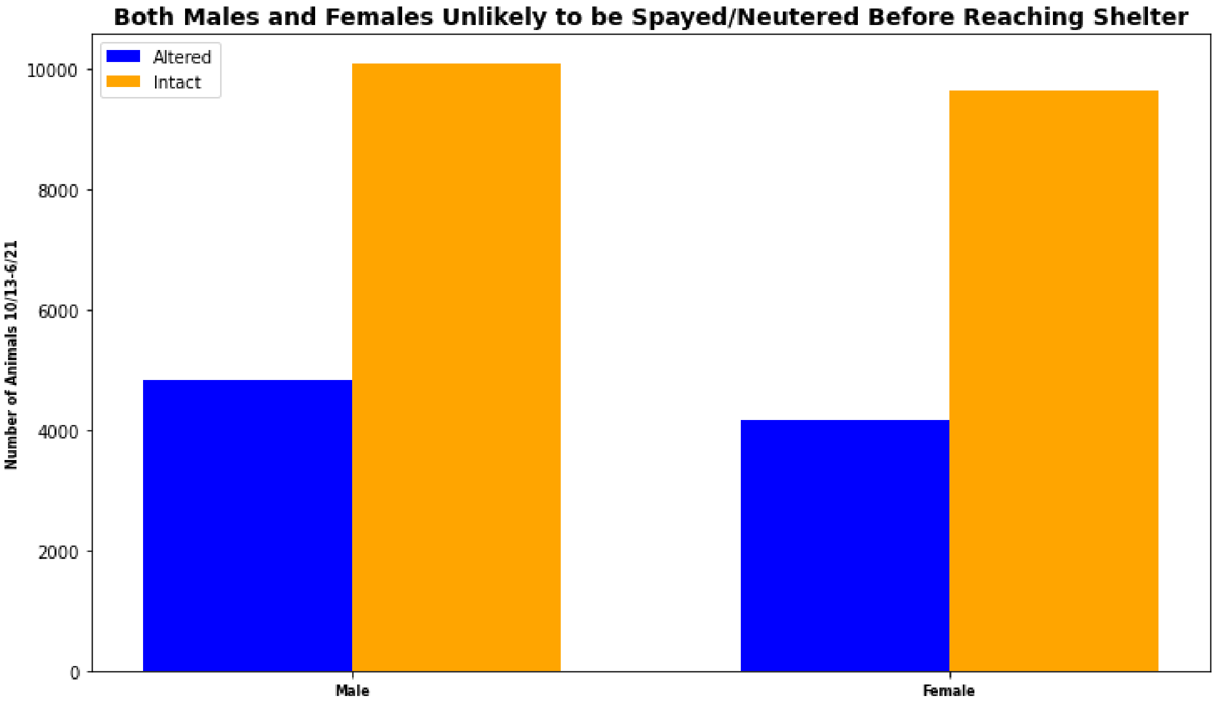
```
outcomeDict = {}
for item in Data['Intake Type']:
    if item in outcomeDict:
        outcomeDict[item] += 1
    else:
        outcomeDict[item] = 1
#print(outcomeDict)
outcomevals = list(outcomeDict.values())
outcomekeys = list(outcomeDict.keys())
plt.bar(outcomekeys,outcomevals,align = 'center')
plt.xticks(outcomekeys,rotation='90')
plt.title("Outcome Results for sheltered Animals")
plt.show()
```



In [82]:

```
# Create Side by Side Bar Graphs to see whether Males or Females
# Are More Likely to be Spayed/Neutered at Intake

Data['Sex at Intake'].value_counts()
print(Data['Sex at Intake'].value_counts())

Altered = (4826, 4152)
ind = np.arange(2)
width = 0.35
fig, ax = plt.subplots()
rects1 = ax.bar(ind, Altered, width, color='blue')

Intact = (10069, 9638)
rects2 = ax.bar(ind+width, Intact, width, color='orange')

ax.set_ylabel("Number of Animals 10/13-6/21", fontsize=8, fontweight='bold')
ax.set_title("Both Males and Females Unlikely to be Spayed/Neutered Before Reachi
ax.set_xticks(ind+width/2)
ax.set_xticklabels(('Male', 'Female'), fontsize=8, fontweight='bold')
ax.legend((rects1[0], rects2[0]), ("Altered", "Intact"))
print(" ")
plt.show()
```

```
Intact Male      10069
Intact Female     9638
Neutered Male     4826
Spayed Female     4152
Unknown           2630
Name: Sex at Intake, dtype: int64
```



In [ ]: