Who Let the Dogs Out?

Group 3:
Katie Hobbs, Gabie Lorenzo,
Anisha Naidu, Jason Noble

Analysis on the population of adoptable dogs advertised on Petfinder in Georgia.

Agenda

- Context
- Data Collection and Data Cleaning
- Data Visualization and Key Results
  - Size
  - Gender
  - Age
  - Location
- Limitations and Future Studies
- Appendix

Context

- **Petfinder** is an internet company that operates the largest online pet adoption website serving all of North America.
- Our group wanted to understand more about the population of adoptable dogs advertised on **Petfinder in Georgia**, specifically if differences in certain parameters have a relationship with the length of time dogs spend on Petfinder.
- The analysis could help Petfinder focus efforts on finding compatible homes for dogs.
- Parameters analyzed in this analysis:
  - Size
  - Gender
  - Age
  - Location

Data Collection and Preparation

Step 1

API Calls

API calls were made to the Petfinder API

Step 2

Create DataFrame

Collected data was organized into a DataFrame

Step 3

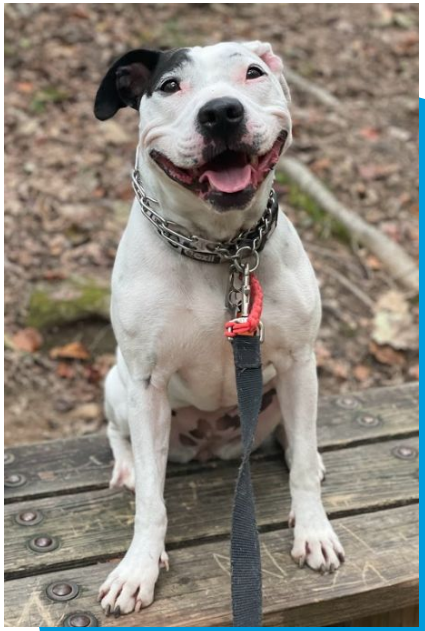
Clean the Data

Data was cleaned to reduce chance of errors during analysis.

Step 4

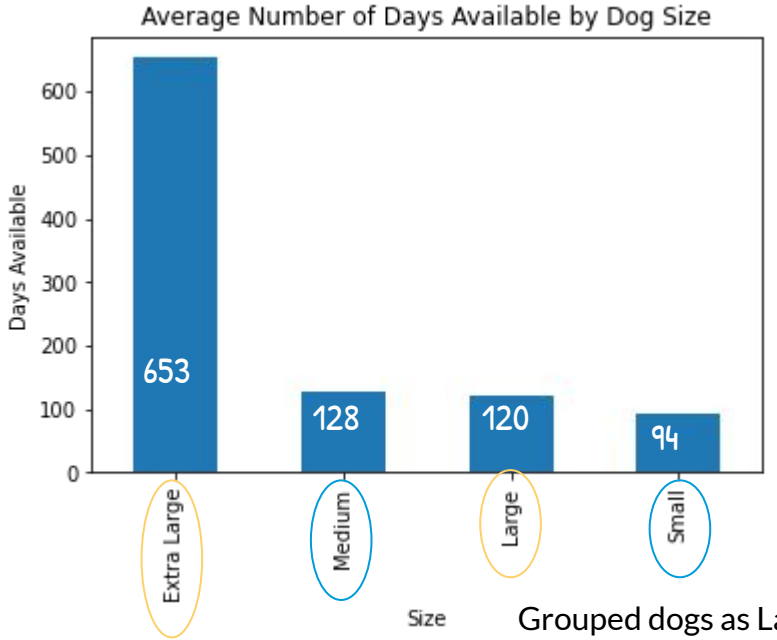
Create Column with Days Available

Now we are ready to perform our analysis!



Question 1: Size

Do larger dogs spend more time on Petfinder than smaller dogs?

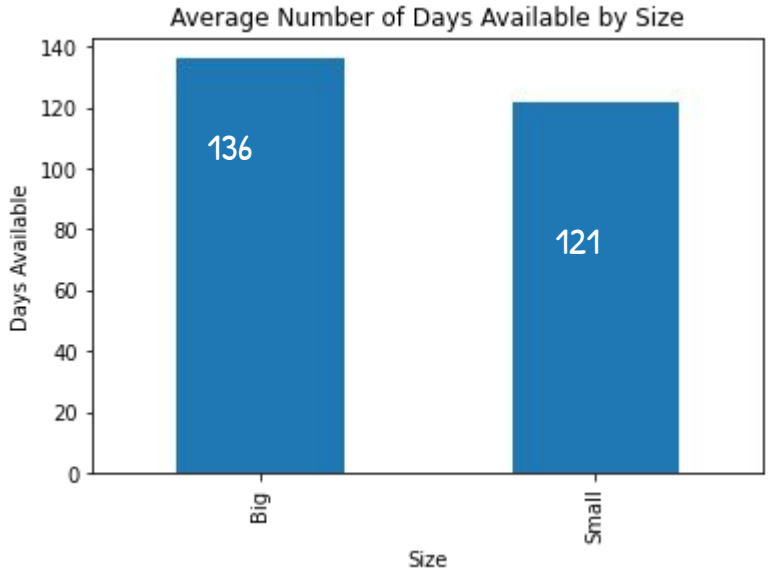


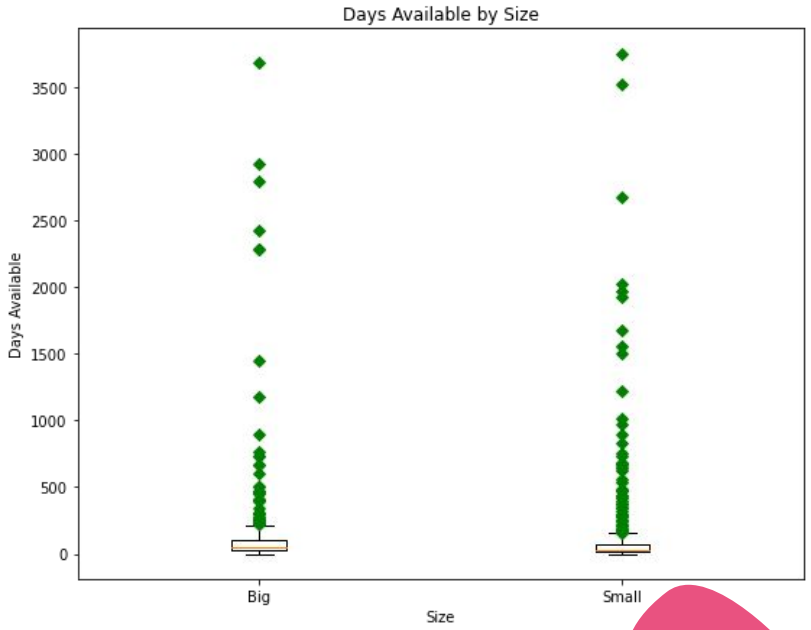
Size	Dog Count
Extra Large (101 lbs or more)	11
Large (61-100 lbs)	371
Medium (26-60 lbs)	358
Small (0-25 lbs)	92

Grouped dogs as Larger (Extra Large and Large) vs Smaller (Medium and Small) to get a more comparable sample size of each group.

Result

* H1_o: Large and x-large dogs spend the same of time on average on Petfinder as small and medium dogs.
* H1_a1: **Large and x-large dogs spend more average time on Petfinder than small and medium dogs.**
* H1_a2: Large and x-large dogs spend less average time on Petfinder than small and medium dogs.



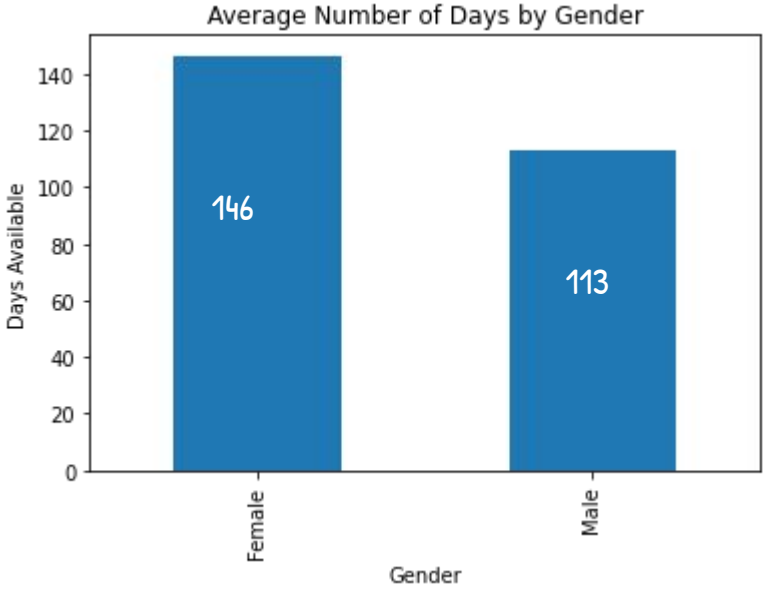


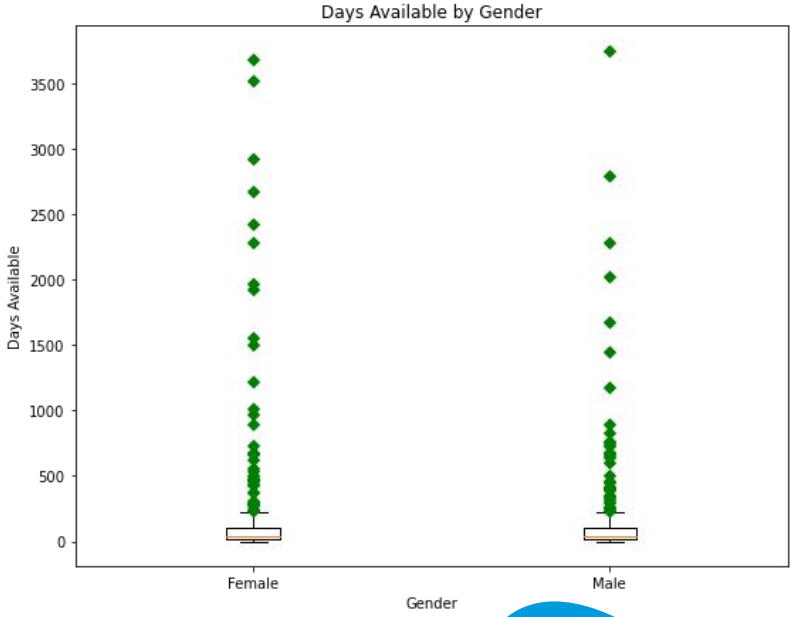
Question 2: Gender

Do female dogs spend more time on average on Petfinder than male dogs?

Result

* H2_o: Male dogs spend on average the same amount of time on Petfinder as female dogs.
* H2_a1: Male dogs spend more time on average on Petfinder than female dogs.
* H2_a2: **Male dogs spend less time on average on Petfinder than female dogs.**

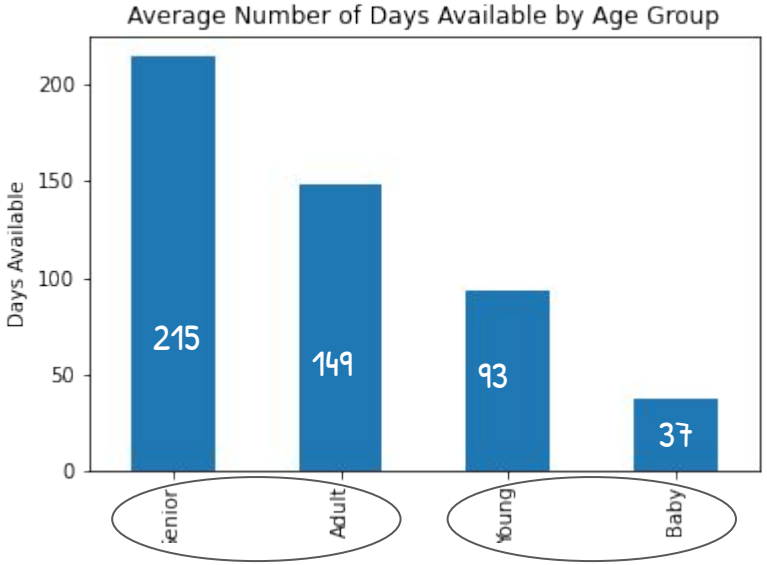




However, the male dog spends a median 38 days on Petfinder, while the female dog spends a median 35 days on Petfinder.

Question 3: Age

Do older dogs spend more time on Petfinder than younger dogs?

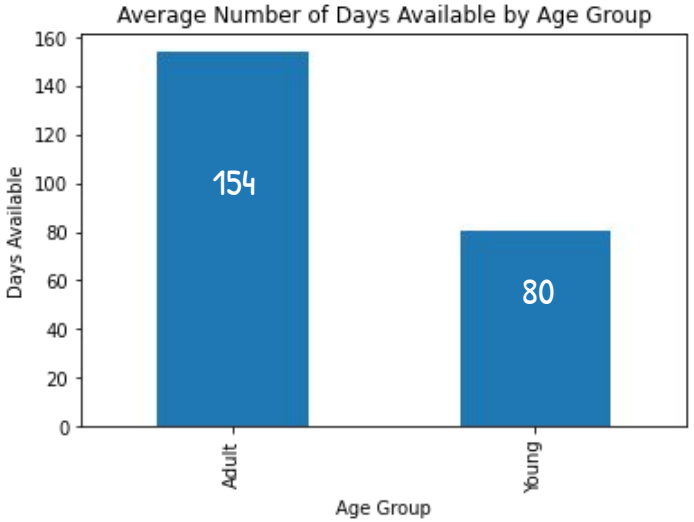


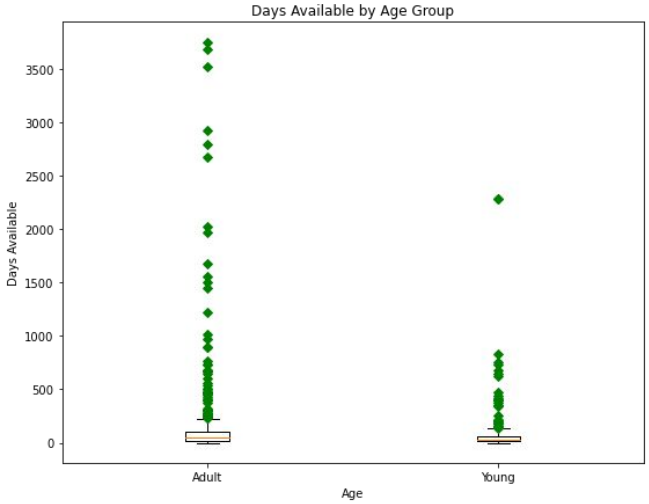
Age	Dog Count
Senior	44
Adult	495
Young	225
Baby	68

Grouped dogs as Adult (Senior and Adult) vs Young (Young and Baby) to get a more comparable sample size of each group.

Result

* H3_o: Adult and senior dogs spend the same amount of time on average on Petfinder than baby or young dogs in Georgia.
* H3_a1: **Adult and senior dogs spend more average time on Petfinder in Georgia than baby or young dogs.**
* H3_a2: Adult and senior dogs spend less average time on Petfinder in Georgia than baby or young dogs.





Question 4: Location

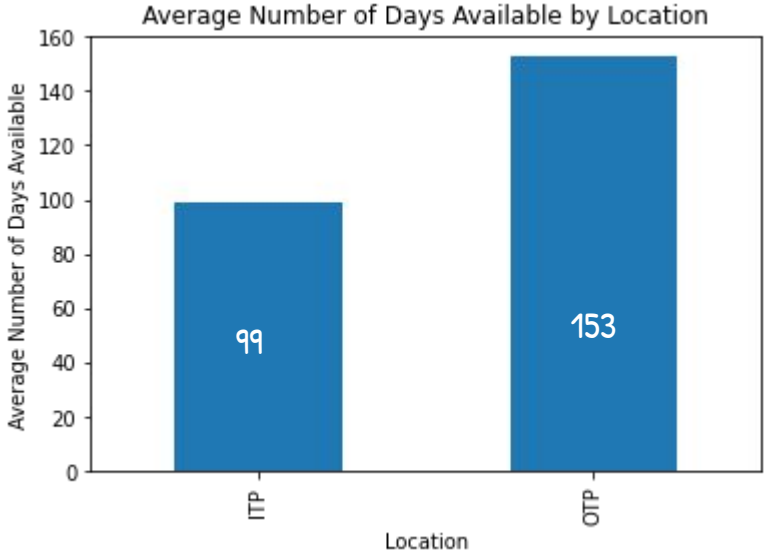
Do dogs located inside the perimeter of Atlanta spend more time on average on Petfinder than dogs located outside the perimeter of Atlanta?

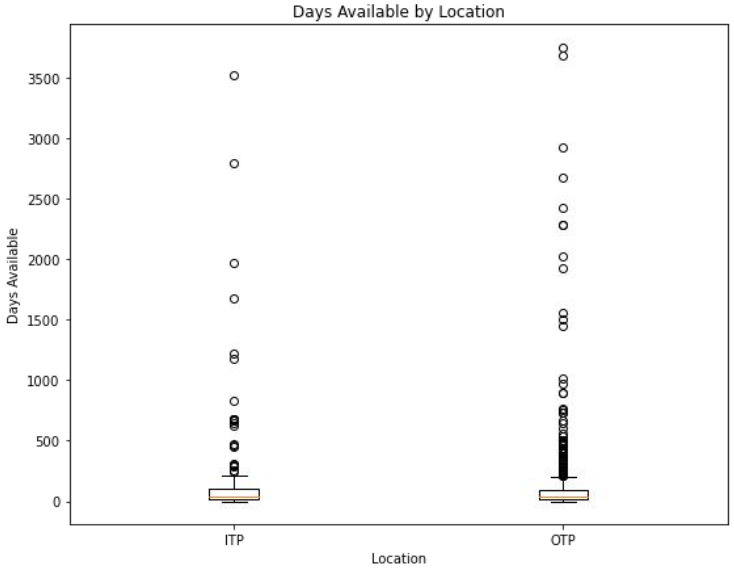

```
itp_atlanta = ["Atlanta","East Point", "College Park",
               "Hapeville", "Decatur","Belvedere Park","Chamblee","Doraville",
               "Panthersville","Candler-McAfee", "Scottsdale",
               "North Decatur","North Druid Hills","Brookhaven","Vinings"]
```

Size	Dog Count
ITP	383
OTP	449

Result

* H4_o: Dogs based inside the perimeter of Atlanta spend on average the same amount of time on Petfinder than dogs that are outside the perimeter of Atlanta.
* H4_1a: Dogs based inside the perimeter of Atlanta cities spend more time on average on Petfinder than dogs based outside the perimeter of Atlanta.
* H4_2a: **Dogs based inside the perimeter of Atlanta cities spend less time on average on Petfinder than dogs based outside the perimeter of Atlanta.**





However, the median dog ITP spends 39 days on Petfinder, while the median dog OTP spends 38 dogs on Petfinder.

Limitations and Future Studies

- Data Set:
  - A **snapshot of dogs available on Petfinder was used** to complete the analysis because historic data of dogs that had already been adopted was not available. A study that evaluated adopted dogs with known durations on Petfinder would be a more accurate study.
  - Petfinder's API provided **categorical data** for weight (big or small) and age (adult or young). Having **numerical data** for age and weight would allow us to perform a study of the strength of the relationships between these variables.
- Method:
  - By comparing **average days** on Petfinder instead of **median days,** the data is skewed by **outliers**. Median days or reducing outliers could be used to improve robustness of findings.
  - Dog sample sizes are not the same. Study could take a random sample of the same number of dogs in each grouping to compare.
- Additional Studies:
  - Future studies based on organization could help optimize resource sharing between the groups to ultimately improve overall dog adoption rate.
  - A deeper study analyzing all parameters affecting the likelihood to be adopted could be performed to predict the length of time a dog will be available for adoption on Petfinder.

Questions?

Appendix

- Calls to Petfinder API
- Create DataFrame of unique dogs in Georgia

In [1]:
```
# Dependencies
import requests
import json
import pandas as pd

# Import API keys
from api_keys import api_key
from api_keys import secret
from api_keys import token_type
from api_keys import access_token
```

In [6]:
```
# create empty lists of data to collect
pet_id = []
name = []
status = []
publish_date = []
org_id = []
city = []
state = []
breed = []
age = []
gender = []
size = []
coat = []
spayed_neutered = []
house_trained = []
special_needs = []
shots_current = []
```

```
# loop to gather data from API with separate API calls for individual random dogs in Georgia
for i in range(1000):

    # construct data request
    url = f'https://api.petfinder.com/v2/animals?type=dog&location=georgia&sort=random&limit=1'

    # get response and jsonify
    response = requests.get(url, headers=headers).json()

    # if pet_id from response is not in pet_id list, append the data
    if response['animals'][0]['id'] not in pet_id:
        try:
            pet_id.append(response['animals'][0]['id'])
            name.append(response['animals'][0]['name'])
            status.append(response['animals'][0]['status'])
            publish_date.append(response['animals'][0]['published_at'])
            org_id.append(response['animals'][0]['organization_id'])
            breed.append(response['animals'][0]['breeds']['primary'])
            age.append(response['animals'][0]['age'])
            gender.append(response['animals'][0]['gender'])
            size.append(response['animals'][0]['size'])
            coat.append(response['animals'][0]['coat'])
            spayed_neutered.append(response['animals'][0]['attributes']['spayed_neutered'])
            house_trained.append(response['animals'][0]['attributes']['house_trained'])
            special_needs.append(response['animals'][0]['attributes']['special_needs'])
            shots_current.append(response['animals'][0]['attributes']['shots_current'])
            city.append(response['animals'][0]['contact']['address']['city'])
            state.append(response['animals'][0]['contact']['address']['state'])
            print(f'Dog was found! Appending data...')
        except:
            print(f'Dog was not found!')

    # if pet_id from response is already in pet_id list, skip it
    else:
        pass
```

Appendix

In [12]:

```
# create dictionary to pass into DataFrame
dog_dict = {
    'Pet ID': pet_id,
    'Name': name,
    'Status': status,
    'Publish Date': publish_date,
    'Organization ID': org_id,
    'City': city,
    'State': state,
    'Breed': breed,
    'Age': age,
    'Gender': gender,
    'Size': size,
    'Coat': coat,
    'Spayed or Neutered': spayed_neutered,
    'House Trained': house_trained,
    'Special Needs': special_needs,
    'Shots Current': shots_current,
}

# create DataFrame from dictionary
georgia_dogs = pd.DataFrame(dog_dict)
georgia_dogs.head()
```

Out[12]:

	Pet ID	Name	Status	Publish Date	Organization ID	City	State	Breed	Age	Gender	Size	Coat	Spayed or Neutered	House Trained	Special Needs	Shots Current
0	54858447	Boon	adoptable	2022-03-04T18:10:57+0000	GA862	Athens	GA	Mixed Breed	Adult	Male	Large	None	True	False	False	False
1	54802620	CASPER	adoptable	2022-03-01T19:01:22+0000	GA99	Macon	GA	Mixed Breed	Young	Male	Medium	None	True	False	False	False
2	55368350	Giacomo (in foster)	adoptable	2022-04-22T00:06:14+0000	GA335	Atlanta	GA	Mixed Breed	Young	Male	Small	None	True	False	False	False
3	55179362	Loki	adoptable	2022-04-01T16:52:47+0000	GA340	Tucker	GA	Labrador Retriever	Adult	Male	Large	Short	True	True	True	True
4	55331643	QUEEN	adoptable	2022-04-18T16:56:32+0000	GA217	Atlanta	GA	Pit Bull Terrier	Young	Female	Medium	None	False	False	False	False

Appendix

- Data Cleaning
  - Minor capitalization formatting
  - Looked for NaN values
- Calculating the target (Average days on Petfinder)
  - API included parameter of "published_at" that we included in our DataFrame
  - Use datetime library
  - Date Collected - Publish Date = Days Available on Petfinder

Name	Status	Publish Date	Organization ID	City	State	Breed	Age	Gender	Size	Coat	Spayed or Neutered	House Trained	Special Needs	Shots Current	Date Collected	Days Available
Boon	adoptable	2022-03-04	GA862	Athens	GA	Mixed Breed	Adult	Male	Large	NaN	True	False	False	False	2022-04-26	53
Casper	adoptable	2022-03-01	GA99	Macon	GA	Mixed Breed	Young	Male	Medium	NaN	True	False	False	False	2022-04-26	56
Giacomo (In Foster)	adoptable	2022-04-22	GA335	Atlanta	GA	Mixed Breed	Young	Male	Small	NaN	True	False	False	False	2022-04-26	4
Loki	adoptable	2022-04-01	GA340	Tucker	GA	Labrador Retriever	Adult	Male	Large	Short	True	True	True	True	2022-04-26	25
Queen	adoptable	2022-04-18	GA217	Atlanta	GA	Pit Bull Terrier	Young	Female	Medium	NaN	False	False	False	False	2022-04-26	8

Appendix

- Size Analysis:

In [6]:
```
#Using the aggregation method, look at average
initial_stats = dogs_df.groupby(['Size'])['Days Available'].agg(['mean', 'median', 'var', 'std','sem'])
initial_stats['mean']
```

Out[6]:
```
Size
Extra Large    653.272727
Large          120.811321
Medium         128.438547
Small           93.750000
Name: mean, dtype: float64
```

In [7]:
```
#Generate a bar plot showing the days vs avg size
total_days = dogs_df.groupby(['Size'])['Days Available'].mean()
bar_plot = total_days.sort_values(ascending=False).plot(kind='bar')

# Set the xlabel, ylabel, and title
bar_plot.set_ylabel("Days Available")
bar_plot.set_title("Average Number of Days Available by Dog Size")

plt.savefig("Figures/bar_dog_size_all.png")
plt.show()
```

Appendix

- Size Analysis:

In [10]:

```
#There are only a small number of "Extra Large" dogs.
#This small sample size could be skewing the data.
#Modify grouping lists to be either big (Extra large and Large) or small (Medium and Small).

#create a new dataframe for dogs_df
dogs_df2 = dogs_df

size_group_df ={
    "Extra Large":"Big",
    "Large": "Big",
    "Medium":"Small",
    "Small":"Small"
}

dogs_df2['Group']=dogs_df2['Size'].map(size_group_df)
dogs_df2
```

Appendix

- Gender Analysis:

In [16]:
```
#Using the aggregation method, look at average
initial_stats = dogs_df.groupby(['Gender'])['Days Available'].agg(['mean', 'median', 'var', 'std','sem'])
initial_stats
```

Out[16]:

Gender	mean	median	var	std	sem
Female	146.659517	35.0	184237.278921	429.228702	22.224620
Male	113.091503	38.0	93237.930473	305.348867	14.252465

In [17]:
```
#Generate a bar plot showing the days vs avg size
total_days = dogs_df.groupby(['Gender'])['Days Available'].mean()
bar_plot = total_days.sort_values(ascending=False).plot(kind='bar')

# Set the xlabel, ylabel, and title
bar_plot.set_ylabel("Days Available")
bar_plot.set_title("Average Number of Days by Gender")

plt.savefig("Figures/bar_dog_gender.png")
plt.show()
```

Appendix

- Age Analysis:

In [21]:
```
#Using the aggregation method, look at average
initial_stats = dogs_df.groupby(['Age'])['Days Available'].agg(['mean', 'median', 'var', 'std','sem'])
initial_stats
```

Out[21]:

Age	mean	median	var	std	sem
Adult	148.686869	45.0	173708.507013	416.783525	18.733026
Baby	37.058824	14.0	6299.130817	79.367064	9.624670
Senior	214.909091	82.0	232801.293869	482.494864	72.738838
Young	93.497778	30.0	61840.331468	248.677163	16.578478

In [22]:
```
#Generate a bar plot showing the days vs avg size
total_days = dogs_df.groupby(['Age'])['Days Available'].mean()
bar_plot = total_days.sort_values(ascending=False).plot(kind='bar')

# Set the xlabel, ylabel, and title
bar_plot.set_ylabel("Days Available")
bar_plot.set_title("Average Number of Days Available by Age Group")

plt.savefig("Figures/bar_dog_age_all.png")
plt.show()
```

Appendix

- Age Analysis:

In [25]:

```
#Modify grouping lists to be either big (Extra large and Large) or small (Medium and Small).

#create a new dataframe for dogs_df
dogs_df_age = dogs_df

age_group_df ={
    "Senior":"Adult",
    "Adult": "Adult",
    "Young":"Young",
    "Baby":"Young"
}

dogs_df_age['Age_Group']=dogs_df_age['Age'].map(age_group_df)
dogs_df_age
```

Appendix

- Location Analysis:

In [30]:

```
#Modify grouping lists to be either ITP (in the perimeter) or not.
#cities inside the perimeter of Atlanta

itp_atlanta = ["Atlanta","East Point", "College Park",
               "Hapeville", "Decatur","Belvedere Park","Chamblee","Doraville",
               "Panthersville","Candler-McAfee", "Scottsdale",
               "North Decatur","North Druid Hills","Brookhaven","Vinings"]
itp_cities = []
for city in dogs_df['City']:

    if city in itp_atlanta:
        itp_cities.append("ITP")
    else:
        itp_cities.append("OTP")
dogs_df['Location'] = itp_cities
dogs_df.head()
```

Appendix

- Location Analysis:

In [32]:
```
#Using the aggregation method, look at average
ITP_stats = dogs_df.groupby(['Location'])['Days Available'].agg(['mean', 'median', 'var', 'std','sem'])
ITP_stats
```

Out[32]:

Location	mean	median	var	std	sem
ITP	99.140992	39.0	84701.681640	291.035533	14.871221
OTP	152.877506	38.0	175263.897908	418.645313	19.757094

In [33]:
```
#Generate a bar plot showing the days vs location
total_days = dogs_df.groupby(['Location'])['Days Available'].mean()
bar_plot = total_days.sort_values(ascending=True).plot(kind='bar')

# Set the xlabel, ylabel, and title
bar_plot.set_ylabel("Average Number of Days Available")
bar_plot.set_xlabel("Location")
bar_plot.set_title("Average Number of Days Available by Location")

plt.savefig("Figures/bar_dog_city_grouped.png")
plt.show()
```