CMSE 201 Final Project

Ariel Ooms

CMSE 201 - 006

December 3, 2024

Animals in the Shelter

Background and Motivation

I have owned dogs since the day I was born and I have been actively volunteering and fundraising for my local humane society since 2017, so I wanted my project to reflect my interest and passion for helping animals especially in a shelter environment. Pet overpopulation has been a problem the United States has been dealing with since the 1940s and shelters have been a key player in reducing the number of animals roaming the streets. There are several types of shelters, as outlined in "The Role of Animal Shelters in Controlling Pet Overpopulation", but I will be focusing on no-kill shelters. The article describes several statistics related to how pets enter and exit shelters as well as the motivations of pet shelters. Motivations mostly relate to reducing overpopulation by educating the public, using trap/neuter/release programs, and increasing the number of animals adopted rather than purchased. Instead of focusing on these programs, I am choosing to investigate frequency statistics of breed, species, ages, and sizes of animals in the shelter as well as the cost to adopt. I will be observing these variables compared to how long animals tend to spend in the shelter before being adopted or otherwise removed.

My key questions are:

- How does the breed, size, age, and species of an animal relate to how long it spends in the shelter?
- What are the most common reasons an animal is put in a shelter? What are the most common ways an animal is removed from a shelter?

- What species, ages, and breeds are most common in shelters?
- What time of year is most common for an animal to enter or leave a shelter?

Overall, I will be investigating different frequencies of populations within animal shelters as well as the movement in and out of the shelter.

Methodology

I used a few different data sets to answer my research questions. The "shelters" dataset includes information about the movement in and out of the shelters. It comes from the Bloomington Animal Shelter with data provided from 2017-2020. My next dataset is "austin_shelter" and contains data regarding one of the biggest no-kill shelters in the United States. The third dataset, "more_adoption" is synthetic data made specifically for data analysis objectives. Since this data was not truly collected from a shelter, the only variables I found reliable to use was cost so that was the only information I took from this dataset.

In [11]:

```
#import all required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Cleaning the Data

In [14]:

```
austin_shelter = pd.read_csv("/Users/arielooms/Documents/College/Fall 24/CMSE 201/Data/aac_intakes_outcomes
austin_shelter.head()
# https://www.kaggle.com/datasets/aaronschlegel/austin-animal-center-shelter-intakes-and-outcomes?select=aa
```

Out[14]:

	age_upon_outcome	animal_id_outcome	date_of_birth	outcome_subtype	outcome_type	sex_upon_outcome	age_upo
0	10 years	A006100	2007-07-09 00:00:00	NaN	Return to Owner	Neutered Male	
1	7 years	A006100	2007-07-09 00:00:00	NaN	Return to Owner	Neutered Male	
2	6 years	A006100	2007-07-09 00:00:00	NaN	Return to Owner	Neutered Male	
3	10 years	A047759	2004-04-02 00:00:00	Partner	Transfer	Neutered Male	
4	16 years	A134067	1997-10-16 00:00:00	NaN	Return to Owner	Neutered Male	

5 rows × 41 columns

The variables I want to look at from the Austin Shelter dataset is the age of the animals, how the animal was moved out of the shelter, the month in which the animal was adopted/moved, the species, sex, breed, and color of each animal, how the animal entered the shelter, and the amount of time spent in the shelter. All other columns could be removed to reduce complexity.

In [17]:

```
#discard any unnecessary columns
austin_shelter = austin_shelter.drop(columns=['animal_id_outcome', 'date_of_birth', 'outcome_subtype', 'se
       'outcome_number', 'dob_year', 'dob_month', 'dob_monthyear','age_upon_intake', 'animal_id_intake','f
       'age_upon_intake_age_group', 'intake_datetime','intake_year', 'intake_monthyear', 'intake_weekday',
       'intake_number'])
```

In [19]:

```
#reduced dataset
austin_shelter.head()
```

Out[19]:

	age_upon_outcome	outcome_type	age_upon_outcome_(years)	outcome_month	animal_type	breed	color
0	10 years	Return to Owner	10.0	12	Dog	Spinone Italiano Mix	Yellow/White
1	7 years	Return to Owner	7.0	12	Dog	Spinone Italiano Mix	Yellow/White
2	6 years	Return to Owner	6.0	3	Dog	Spinone Italiano Mix	Yellow/White
3	10 years	Transfer	10.0	4	Dog	Dachshund	Tricolor
4	16 years	Return to Owner	16.0	11	Dog	Shetland Sheepdog	Brown/White

In [21]:

```
#import shelter data
shelters = pd.read_csv("/Users/arielooms/Documents/College/Fall 24/CMSE 201/Data/animal-data-1.csv")
shelters.head()
# link: https://www.kaggle.com/datasets/jinbonnie/animal-data
```

Out[21]:

	id	intakedate	intakereason	istransfer	sheltercode	identichipnumber	animalname	breedname	basecolour	spe
0	15801	2009-11-28 00:00:00	Moving	0	C09115463	0A115D7358	Jadzia	Domestic Short Hair	Tortie	
1	15932	2009-12-08 00:00:00	Moving	0	D09125594	0A11675477	Gonzo	German Shepherd Dog/Mix	Tan	
2	28859	2012-08-10 00:00:00	Abandoned	0	D12082309	0A13253C7B	Maggie	Shep Mix/Siberian Husky	Various	
3	30812	2013-01-11 00:00:00	Abandoned	0	C1301091	0A13403D4D	Pretty Girl	Domestic Short Hair	Dilute tortoiseshell	
4	30812	2013-01-11 00:00:00	Abandoned	0	C1301091	0A13403D4D	Pretty Girl	Domestic Short Hair	Dilute tortoiseshell	

5 rows × 23 columns

In the "shelters" dataset, I need the reason for intake, the animal name, breed name, species name, animal age, sex of the animal, and movemenet type. I can remove all other columns.

In [24]:

```
#remove unnecessary columns
shelters = shelters.drop(columns=['id', 'intakedate', 'istransfer', 'sheltercode',
       'identichipnumber','basecolour','location', 'movementdate', 'istrial', 'returndate', 'returnedreaso
       'deceaseddate', 'deceasedreason', 'diedoffshelter', 'puttosleep',
       'isdoa'])
```

In [26]:

```
shelters.head()
```

Out[26]:

	intakereason	animalname	breedname	speciesname	animalage	sexname	movementtype
0	Moving	Jadzia	Domestic Short Hair	Cat	9 years 2 months.	Female	Adoption
1	Moving	Gonzo	German Shepherd Dog/Mix	Dog	9 years 1 month.	Male	Adoption
2	Abandoned	Maggie	Shep Mix/Siberian Husky	Dog	6 years 8 months.	Female	Adoption
3	Abandoned	Pretty Girl	Domestic Short Hair	Cat	8 years 11 months.	Female	Foster
4	Abandoned	Pretty Girl	Domestic Short Hair	Cat	8 years 11 months.	Female	Adoption

Now that the data is cleaned, I can begin calculating the frequency of my variables.

Intake and Movement

I am starting by looking at the timing of intakes and outtakes during the year. I will add up the total of incomes and outcomes of each month and plot those to observe if there is any meaningful patterns.

In [31]:

```
#uses austin shelter, intake and outtake amounts
#initializing each month intake and outtake amounts
jan_intake = 0
feb_intake = 0
mar_intake = 0
apr_intake = 0
may_intake = 0
jun_intake = 0
jul_intake = 0
aug_intake = 0
sep_intake = 0
oct_intake = 0
nov_intake = 0
dec_intake = 0

jan_out = 0
feb_out = 0
mar_out = 0
apr_out = 0
may_out = 0
```

```
jun_out = 0
jul_out = 0
aug_out = 0
sep_out = 0
oct_out = 0
nov_out = 0
dec_out = 0

#checks each row and adds 1 to whichever month the animal was taken in
for month in austin_shelter["intake_month"]:
    if month ==  1:
        jan_intake += 1
    elif month ==  2:
        feb_intake += 1
    elif month ==  3:
        mar_intake += 1
    elif month ==  4:
        apr_intake += 1
    elif month ==  5:
        may_intake += 1
    elif month ==  6:
        jun_intake += 1
    elif month ==  7:
        jul_intake += 1
    elif month ==  8:
        aug_intake += 1
    elif month ==  9:
        sep_intake += 1
    elif month ==  10:
        oct_intake += 1
    elif month ==  11:
        nov_intake += 1
    elif month ==  12:
        dec_intake += 1

#checks each row and adds 1 to whichever month the animal was moved out
for month in austin_shelter["outcome_month"]:
    if month ==  1:
        jan_out += 1
    elif month ==  2:
        feb_out += 1
    elif month ==  3:
        mar_out += 1
    elif month ==  4:
        apr_out += 1
    elif month ==  5:
        may_out += 1
    elif month ==  6:
        jun_out += 1
    elif month ==  7:
        jul_out += 1
    elif month ==  8:
        aug_out += 1
    elif month ==  9:
        sep_out += 1
    elif month ==  10:
        oct_out += 1
    elif month ==  11:
        nov_out += 1
    elif month ==  12:
        dec_out += 1

#creates categories to be plotted later
months = ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"]
month_intake_amt_austin = [jan_intake,feb_intake,mar_intake,apr_intake,may_intake,jun_intake,jul_intake,aug
month_out_amt_austin = [jan_out,feb_out,mar_out,apr_out,may_out,jun_out,jul_out,aug_out,sep_out,oct_out,nov
```

In [33]:

```
#make major classes - Austin
#Disease or injury - 'Wildlife'
#"Lifestyle of owner - 'Owner Surrender
#behavior -  'Public Assist'
# abuse - 'Euthanasia Request'
#abandoned or stray - "Stray"

sick_or_injured = 0
lifestyle_change = 0
behavior = 0
abuse = 0
abandoned = 0
other = 0

#checks the reason for intake and adds 1 to the correct category
for reason in austin_shelter["intake_type"]:
    if reason == 'Wildlife':
        sick_or_injured += 1
    elif reason == 'Owner Surrender':
        lifestyle_change += 1
    elif reason == 'Public Assist':
        behavior += 1
    elif reason == 'Euthanasia Request':
        abuse += 1
    elif reason == "Stray":
        abandoned += 1
    else:
        other += 1

intake_amounts_austin = [sick_or_injured, lifestyle_change, behavior, abuse, abandoned, other]
intake_reasons_austin = ["Sick/Injured", "Lifestyle Change", "Behavior", "Abuse", "Abandoned", "Other"]
```

In [35]:

```
#unique Bloomington Intake reasons
shelters["intakereason"].unique()
```

Out[35]:

```
array(['Moving', 'Abandoned', 'Incompatible with owner lifestyle',
       'Rabies Monitoring', 'Marriage/Relationship split',
       'Owner Deceased', 'Stray', nan, 'Police Assist', 'Biting',
       'Owner Died', 'TNR - Trap/Neuter/Release', 'Unable to Afford',
       'Unsuitable Accommodation', 'Injured Wildlife', 'Allergies',
       'Transfer from Other Shelter', 'Born in Shelter', 'Sick/Injured',
       'Landlord issues', 'Litter relinquishment',
       'Owner requested Euthanasia', 'Abuse/ neglect',
       'Incompatible with other pets', 'Behavioral Issues', 'DOA'],
      dtype=object)
```

In [37]:

```
#unique Austin Intake reasons
austin_shelter['intake_type'].unique()
```

Out[37]:

```
array(['Stray', 'Public Assist', 'Owner Surrender', 'Euthanasia Request',
       'Wildlife'], dtype=object)
```

In [39]:

```
#make major classes - Bloomington
#Disease or injury - "Rabies Monitoring", "Injured Wildlife"
#"Lifestyle of owner - "Incompatible with owner lifestyle", "Marriage/Relationship split", "Unsuitable Acc
    #"Landlord issues", "Incompatible with other pets", "Moving" , "Litter relinquishment"
#already in shelter - "Transfer from Other Shelter", "Born in Shelter"
#behavior -  "Biting", "Behavioral Issues"
# abuse - "Abuse/ neglect", "Owner requested Euthanasia", "Police Assist"
```

```
#abandoned or stray - "Abandoned", "Stray"
#owner death -  "Owner Deceased", "Owner Died"

sick_or_injured = 0
lifestyle_change = 0
in_shelter = 0
behavior = 0
abuse = 0
abandoned = 0
other = 0
owner_death = 0
TNR = 0

#checks the reason for intake and adds 1 to the correct category
for reason in shelters["intakereason"]:
    if reason == "Rabies Monitoring" or reason == "Injured Wildlife":
        sick_or_injured += 1
    elif reason == "Incompatible with owner lifestyle" or reason == "Marriage/Relationship split" or reaso
        lifestyle_change += 1
    elif reason == "Transfer from Other Shelter" or reason == "Born in Shelter":
        in_shelter += 1
    elif reason == "Biting" or reason == "Behavioral Issues":
        behavior += 1
    elif reason == "Abuse/ neglect" or reason == "Owner requested Euthanasia" or reason == "Police Assist"
        abuse += 1
    elif reason == "Abandoned" or reason == "Stray":
        abandoned += 1
    elif reason ==  "Owner Deceased" or reason == "Owner Died":
        owner_death += 1
    elif reason == 'TNR - Trap/Neuter/Release':
        TNR += 1
    else:
        other += 1

intake_amounts_bloom = [sick_or_injured, lifestyle_change, in_shelter, behavior, abuse, abandoned, owner_de
intake_reasons_bloom = ["Sick/Injured", "Lifestyle Change", "In Shelter", "Behavior", "Abuse", "Abandoned"
```

In [41]:

```
#unique Bloomington movement types
shelters["movementtype"].unique()
```

Out[41]:

```
array(['Adoption', 'Foster', 'Transfer', 'Reclaimed', 'Released To Wild',
       'Stolen', 'Escaped'], dtype=object)
```

In [43]:

```
#classes for Bloomington movement
adopted = 0
foster = 0
transfer = 0
reclaimed = 0
other = 0

#checks the reason for movement and adds 1 to the correct category
for type in shelters["movementtype"]:
    if type == "Adoption":
        adopted += 1
    elif type == "Foster":
        foster += 1
    elif type == "Transfer":
        transfer += 1
    elif type == "Reclaimed":
        reclaimed += 1
    elif type == "Released To Wild" or type == "Stolen" or type == "Escaped":
        other += 1


movement_amounts_bloom = [adopted, foster, transfer, reclaimed, other]
movement_reasons_bloom = ["Adopted", "Foster", "Transfer", "Reclaimed", "Other"]
```

Age

In [46]:

```
#function that takes in a dataframe, the range of years wanted, and which column will contain the ages in
def age_pets(petdf, years_min, years_max, col_name):
    lower_mask = petdf[col_name] >= years_min
    upper_mask = petdf[col_name] < years_max
    lower = petdf[lower_mask]
    upper_mask = upper_mask.reindex(lower.index)
    upper = lower[upper_mask]
    return upper
```

In [48]:

```
#austin shelter - changing the age column from months to years for easier comprehension
age_years = []
```

```python
#converts age from year/month to just years
for time in austin_shelter["age_upon_outcome"]:
    if 'year' in time:
        years = time.split(' year')[0] #splits the age at the word years and keeps just the first number to
        years = int(years)
    else:
        years = 0
    age_years.append(years)
austin_shelter["Years Age"] = age_years #makes a new column that marks the age in years
austin_shelter
```

Out[48]:

	age_upon_outcome	outcome_type	age_upon_outcome_(years)	outcome_month	animal_type	breed	
0	10 years	Return to Owner	10.000000	12	Dog	Spinone Italiano Mix	Yellow/
1	7 years	Return to Owner	7.000000	12	Dog	Spinone Italiano Mix	Yellow/
2	6 years	Return to Owner	6.000000	3	Dog	Spinone Italiano Mix	Yellow/
3	10 years	Transfer	10.000000	4	Dog	Dachshund	Tr
4	16 years	Return to Owner	16.000000	11	Dog	Shetland Sheepdog	Brown/
...	...	...	...	...	...	...	
79667	2 weeks	Transfer	0.038356	3	Cat	Domestic Shorthair Mix	
79668	2 years	Euthanasia	2.000000	3	Other	Bat Mix	
79669	1 year	Euthanasia	1.000000	3	Other	Bat Mix	
79670	10 months	Return to Owner	0.821918	3	Dog	Labrador Retriever Mix	Black/
79671	10 years	Euthanasia	10.000000	3	Dog	Boxer Mix	E

79672 rows × 13 columns

In [50]:

```
#creates a new DataFrame with just dogs
dogs_only_austin_mask = austin_shelter["animal_type"] == "Dog"
dogs_only_austin = austin_shelter[dogs_only_austin_mask]
```

In [84]:

```
#austin shelter - creating puppy, middle age, and old dog categories
puppy_austin = age_pets(dogs_only_austin, 0, 1, "Years Age")
mid_age_austin = age_pets(dogs_only_austin, 1, 10, "Years Age")
old_austin = age_pets(dogs_only_austin, 10, 26, "Years Age")

puppy_time_total_austin = 0
mid_age_time_total_austin = 0
old_time_total_austin = 0

#adds up the total amount of days the dogs in each class spent
for time in puppy_austin["time_in_shelter_days"]:
    puppy_time_total_austin += time
for time in mid_age_austin["time_in_shelter_days"]:
    mid_age_time_total_austin += time
for time in old_austin["time_in_shelter_days"]:
   old_time_total_austin += time

#finds the average time spent by each class
puppy_time_avg_austin = puppy_time_total_austin / len(puppy_austin["time_in_shelter_days"])
mid_age_time_avg_austin = mid_age_time_total_austin / len(mid_age_austin["time_in_shelter_days"])
old_time_avg_austin = old_time_total_austin / len(old_austin["time_in_shelter_days"])

age_times_austin = [puppy_time_avg_austin, mid_age_time_avg_austin, old_time_avg_austin]
age_classes = ["Puppies", "Middle Age", "Seniors"]

age_amounts_austin = [len(puppy_austin["Years Age"]), len(mid_age_austin["Years Age"]), len(old_austin["Yea
```

In [86]:

```
#bloomington ages
shelters["animalage"]
years_age = []
for age in shelters["animalage"]:
    if "years" in age:
        years = age.split(' years')[0]
        years = int(years)
        years_age.append(years)
    else:
        years = 0
        years = int(years)
        years_age.append(years)

shelters["Years Age"] = years_age
```

```
dogs_only_mask_bloom = shelters["speciesname"] == "Dog"
dogs_only_bloom = shelters[dogs_only_mask_bloom]
```

In [88]:

```
#bloomington age amounts
puppies_bloom = age_pets(dogs_only_bloom, 0, 1, "Years Age")
dogs_middle_age_bloom = age_pets(dogs_only_bloom, 1, 10, "Years Age")
senior_dogs_bloom = age_pets(dogs_only_bloom, 10, 20, "Years Age")

age_amounts_bloom = [len(puppies_bloom["Years Age"]), len(dogs_middle_age_bloom["Years Age"]), len(senior_
```

Breed

In [91]:

```
#Austin breeds
breeds_austin = austin_shelter["breed"]
unique_breeds = []
for breed in breeds_austin:
    if '/' in breed:
        breed = breed.split('/')[0]  #splits mixed breeds and assigns it to its primary/first breed
    if breed not in unique_breeds:
        unique_breeds.append(breed)
#creates a dictionary containing breed name, the total amount of that breed, the amount spent by all membe
#German Shepherd, Pitbull, Lab, Golden, Husky, Hound, Chihuahua, Terrier, Australian Shepherd, Border Coll
# Dachshund, Spaniel, Rottweiler, Poodle, Corgi, Great Dane
breed_dict = {
    "German Shepherd" : {"Count": 0, "TotalTime": 0 , "AvgTime": 0},
    "Pit Bull" : {"Count": 0, "TotalTime": 0 , "AvgTime": 0},
    "Lab" : {"Count": 0, "TotalTime": 0 , "AvgTime": 0},
    "Golden Retriever" : {"Count": 0, "TotalTime": 0 , "AvgTime": 0},
    "Husky": {"Count": 0, "TotalTime": 0 , "AvgTime": 0},
    "Hound": {"Count": 0, "TotalTime": 0 , "AvgTime": 0},
    "Chihuahua": {"Count": 0, "TotalTime": 0 , "AvgTime": 0},
    "Terrier": {"Count": 0, "TotalTime": 0 , "AvgTime": 0},
    "Australian Shepherd": {"Count": 0, "TotalTime": 0 , "AvgTime": 0},
    "Collie": {"Count": 0, "TotalTime": 0 , "AvgTime": 0},
    "Shih Tzu": {"Count": 0, "TotalTime": 0 , "AvgTime": 0},
    "Saint Bernard": {"Count": 0, "TotalTime": 0 , "AvgTime": 0},
    "Beagle": {"Count": 0, "TotalTime": 0 , "AvgTime": 0},
    "Pug": {"Count": 0, "TotalTime": 0 , "AvgTime": 0},
    "Dachshund": {"Count": 0, "TotalTime": 0 , "AvgTime": 0},
    "Spaniel": {"Count": 0, "TotalTime": 0 , "AvgTime": 0},
    "Rottweiler": {"Count": 0, "TotalTime": 0 , "AvgTime": 0},
    "Poodle": {"Count": 0, "TotalTime": 0 , "AvgTime": 0},
    "Corgi": {"Count": 0, "TotalTime": 0 , "AvgTime": 0},
    "Great Dane": {"Count": 0, "TotalTime": 0 , "AvgTime": 0}
}

#finds all members of each breed and caluclates the total amount of time spent for each one
for index in range(len(breeds_austin)):
    if "German" in breeds_austin[index]:
        breed_dict["German Shepherd"]["Count"] += 1
        breed_dict["German Shepherd"]["TotalTime"] += austin_shelter["time_in_shelter_days"][index]
    elif "Pit" in breeds_austin[index]:
        breed_dict["Pit Bull"]["Count"] += 1
        breed_dict["Pit Bull"]["TotalTime"] += austin_shelter["time_in_shelter_days"][index]
    elif "Lab" in breeds_austin[index]:
        breed_dict["Lab"]["Count"] += 1
        breed_dict["Lab"]["TotalTime"] += austin_shelter["time_in_shelter_days"][index]
    elif "Golden" in breeds_austin[index]:
        breed_dict["Golden Retriever"]["Count"] += 1
        breed_dict["Golden Retriever"]["TotalTime"] += austin_shelter["time_in_shelter_days"][index]
    elif "Husky" in breeds_austin[index]:
        breed_dict["Husky"]["Count"] += 1
        breed_dict["Husky"]["TotalTime"] += austin_shelter["time_in_shelter_days"][index]
    elif "Hound" in breeds_austin[index] or "hound" in breeds_austin[index]:
        breed_dict["Hound"]["Count"] += 1
        breed_dict["Hound"]["TotalTime"] += austin_shelter["time_in_shelter_days"][index]
    elif "Chihuahua" in breeds_austin[index]:
        breed_dict["Chihuahua"]["Count"] += 1
        breed_dict["Chihuahua"]["TotalTime"] += austin_shelter["time_in_shelter_days"][index]
    elif "Terrier" in breeds_austin[index]:
        breed_dict["Terrier"]["Count"] += 1
        breed_dict["Terrier"]["TotalTime"] += austin_shelter["time_in_shelter_days"][index]
    elif "Australian" in breeds_austin[index]:
        breed_dict["Australian Shepherd"]["Count"] += 1
        breed_dict["Australian Shepherd"]["TotalTime"] += austin_shelter["time_in_shelter_days"][index]
    elif "Collie" in breeds_austin[index]:
        breed_dict["Collie"]["Count"] += 1
        breed_dict["Collie"]["TotalTime"] += austin_shelter["time_in_shelter_days"][index]
    elif "Shih" in breeds_austin[index]:
        breed_dict["Shih Tzu"]["Count"] += 1
        breed_dict["Shih Tzu"]["TotalTime"] += austin_shelter["time_in_shelter_days"][index]
    elif "Bernard" in breeds_austin[index]:
        breed_dict["Saint Bernard"]["Count"] += 1
        breed_dict["Saint Bernard"]["TotalTime"] += austin_shelter["time_in_shelter_days"][index]
    elif "Beagle" in breeds_austin[index]:
        breed_dict["Beagle"]["Count"] += 1
        breed_dict["Beagle"]["TotalTime"] += austin_shelter["time_in_shelter_days"][index]
    elif "Pug" in breeds_austin[index]:
        breed_dict["Pug"]["Count"] += 1
        breed_dict["Pug"]["TotalTime"] += austin_shelter["time_in_shelter_days"][index]
    elif "Dachshund" in breeds_austin[index]:
        breed_dict["Dachshund"]["Count"] += 1
        breed_dict["Dachshund"]["TotalTime"] += austin_shelter["time_in_shelter_days"][index]
    elif "Spaniel" in breeds_austin[index]:
        breed_dict["Spaniel"]["Count"] += 1
        breed_dict["Spaniel"]["TotalTime"] += austin_shelter["time_in_shelter_days"][index]
    elif "Rottweiler" in breeds_austin[index]:
        breed_dict["Rottweiler"]["Count"] += 1
        breed_dict["Rottweiler"]["TotalTime"] += austin_shelter["time_in_shelter_days"][index]
    elif "Poodle" in breeds_austin[index]:
        breed_dict["Poodle"]["Count"] += 1
        breed_dict["Poodle"]["TotalTime"] += austin_shelter["time_in_shelter_days"][index]
    elif "Corgi" in breeds_austin[index]:
        breed_dict["Corgi"]["Count"] += 1
        breed_dict["Corgi"]["TotalTime"] += austin_shelter["time_in_shelter_days"][index]
    elif "Dane" in breeds_austin[index]:
        breed_dict["Great Dane"]["Count"] += 1
        breed_dict["Great Dane"]["TotalTime"] += austin_shelter["time_in_shelter_days"][index]

# Averages
averages_austin = []
austin_breed_counts = []
#finds the average time for each breed
for breed in breed_dict:
    avg_time = breed_dict[breed]["TotalTime"] / breed_dict[breed]["Count"]
    averages_austin.append(avg_time)
    count = breed_dict[breed]["Count"]
    austin_breed_counts.append(count)

breed_times_austin = averages_austin
breed_names = ["German Shepherd", "Pit Bull", "Lab", "Golden Retriever", "Husky", "Hound", "Chihuahua", "Te
breed_amounts_austin = austin_breed_counts
```

In [92]:
```
#bloomington drop all nan values in breedname
breeds_only = shelters["breedname"]
breeds_only.dropna(inplace=True)
```

In [93]:
```
breed_types = breeds_only.unique()
```

In [94]:
```
new_breeds = []
for breed in breed_types:
    if '/' in breed:
        breed = breed.split('/')[0]
    if breed not in new_breeds:
        new_breeds.append(breed)

new_breeds
#German Shepherd, Pitbull, Lab, Golden, Husky, Hound, Chihuahua, Terrier, Australian Shepherd, Border Coll
# Dachshund, Spaniel, Rottweiler, Poodle, Corgi, Great Dane
german_shep = 0
pit = 0
lab = 0
golden = 0
husky = 0
hound = 0
chihuahua = 0
terrier = 0
australian_shep = 0
collie = 0
shih_tzu = 0
saint_bernard = 0
beagle = 0
pug = 0
dachshund = 0
spaniel = 0
rottie = 0
poodle = 0
corgi = 0
great_dane = 0


for breed in breeds_only:
    if "German" in breed:
        german_shep += 1
    elif "Pit" in breed:
        pit += 1
    elif "Lab" in breed:
        lab += 1
    elif "Golden" in breed:
        golden += 1
    elif "Husky" in breed:
        husky += 1
    elif "Hound" in breed or "hound" in breed:
        hound += 1
    elif "Chihuahua" in breed:
        chihuahua += 1
    elif "Terrier" in breed:
        terrier += 1
    elif "Australian" in breed:
        australian_shep += 1
    elif "Collie" in breed:
        collie += 1
    elif "Shih" in breed:
        shih_tzu += 1
    elif "Bernard" in breed:
        saint_bernard += 1
    elif "Beagle" in breed:
        beagle += 1
    elif "Pug" in breed:
        pug += 1
    elif "Dachshund" in breed:
        dachshund += 1
    elif "Spaniel" in breed:
        spaniel += 1
    elif "Rottweiler" in breed:
        rottie += 1
    elif "Poodle" in breed:
        poodle += 1
    elif "Corgi" in breed:
        corgi += 1
    elif "Dane" in breed:
        great_dane += 1

breed_amounts = [german_shep, pit, lab, golden, husky, hound, chihuahua, terrier, australian_shep, collie,
```

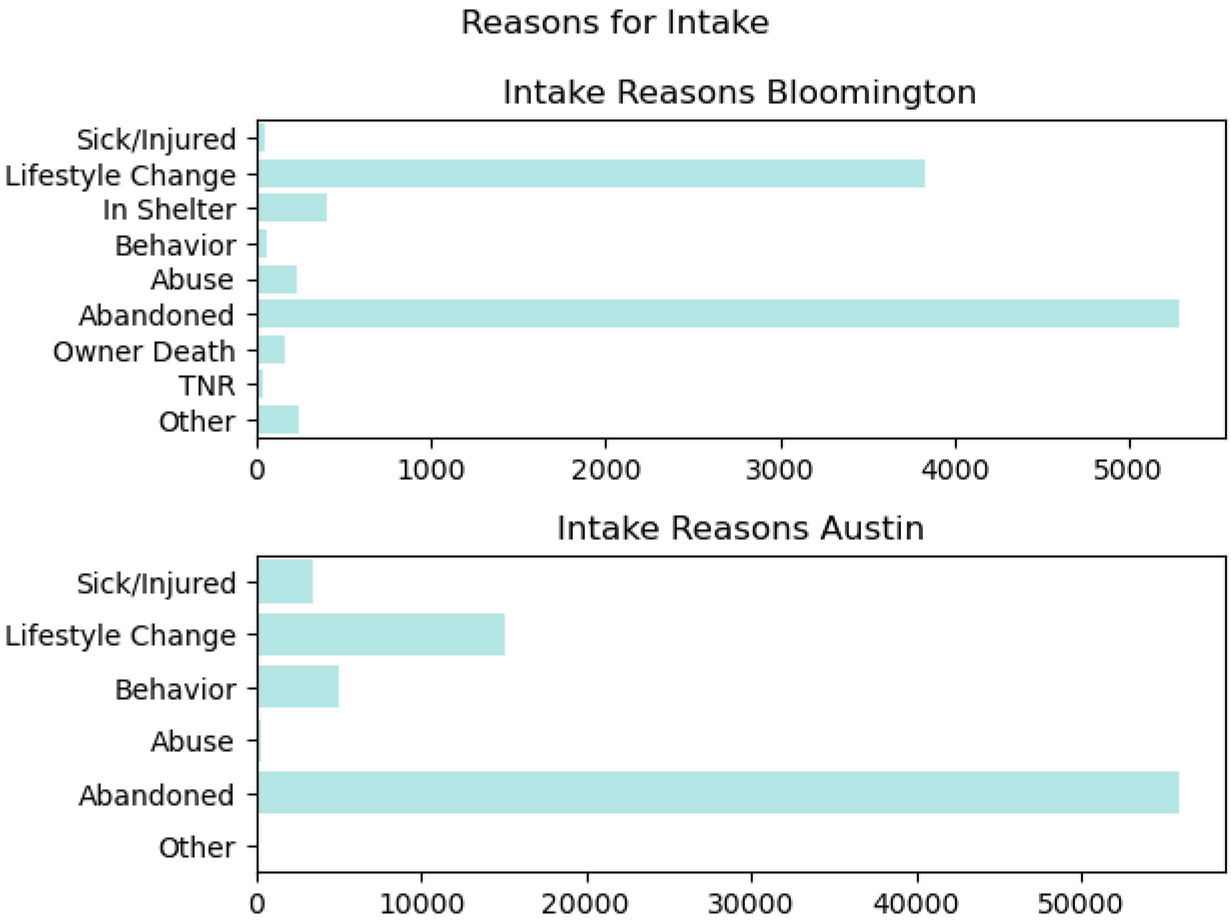
Results

In [99]:

```
colors = ["lightskyblue", 'paleturquoise', 'aquamarine', 'palegreen']
```

In [102…

```
plt.suptitle("Reasons for Intake")
#bloom
plt.subplot(2,1,1)
sns.barplot(x=intake_amounts_bloom, y=intake_reasons_bloom, orient="y", color=colors[1])
plt.title("Intake Reasons Bloomington")
#austin
plt.subplot(2,1,2)
sns.barplot(x=intake_amounts_austin, y=intake_reasons_austin, orient="y", color=colors[1])
plt.title("Intake Reasons Austin")
plt.tight_layout()
```



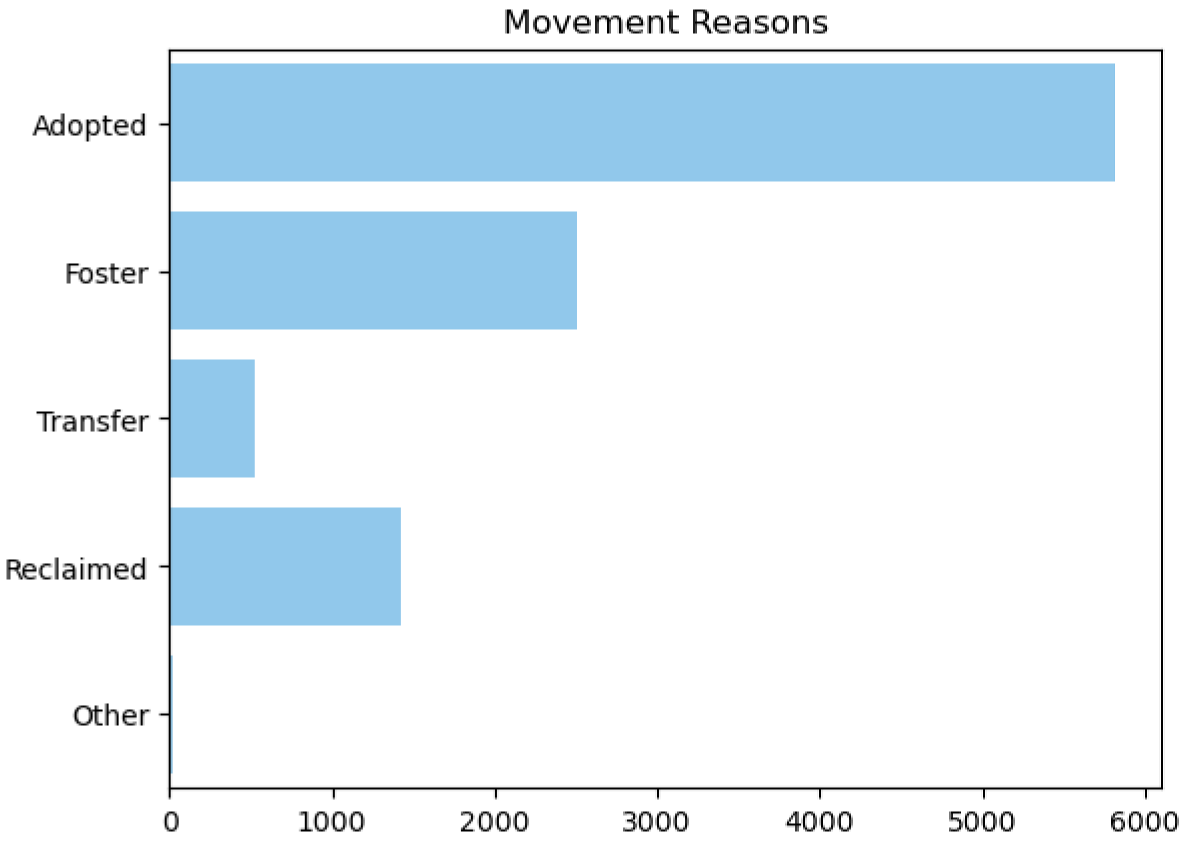
Most animals come to the shelters as strays or because they were abandoned. Second most common is a lifestyle change which could be for a variety of reasons including the death of the owner, moving or new housing, relationship split, unable to afford care, or litter relinquishment.

In [105…

```
sns.barplot(x=movement_amounts_bloom, y=movement_reasons_bloom, orient="y", color=colors[0])
plt.title("Movement Reasons")
```

Out[105…

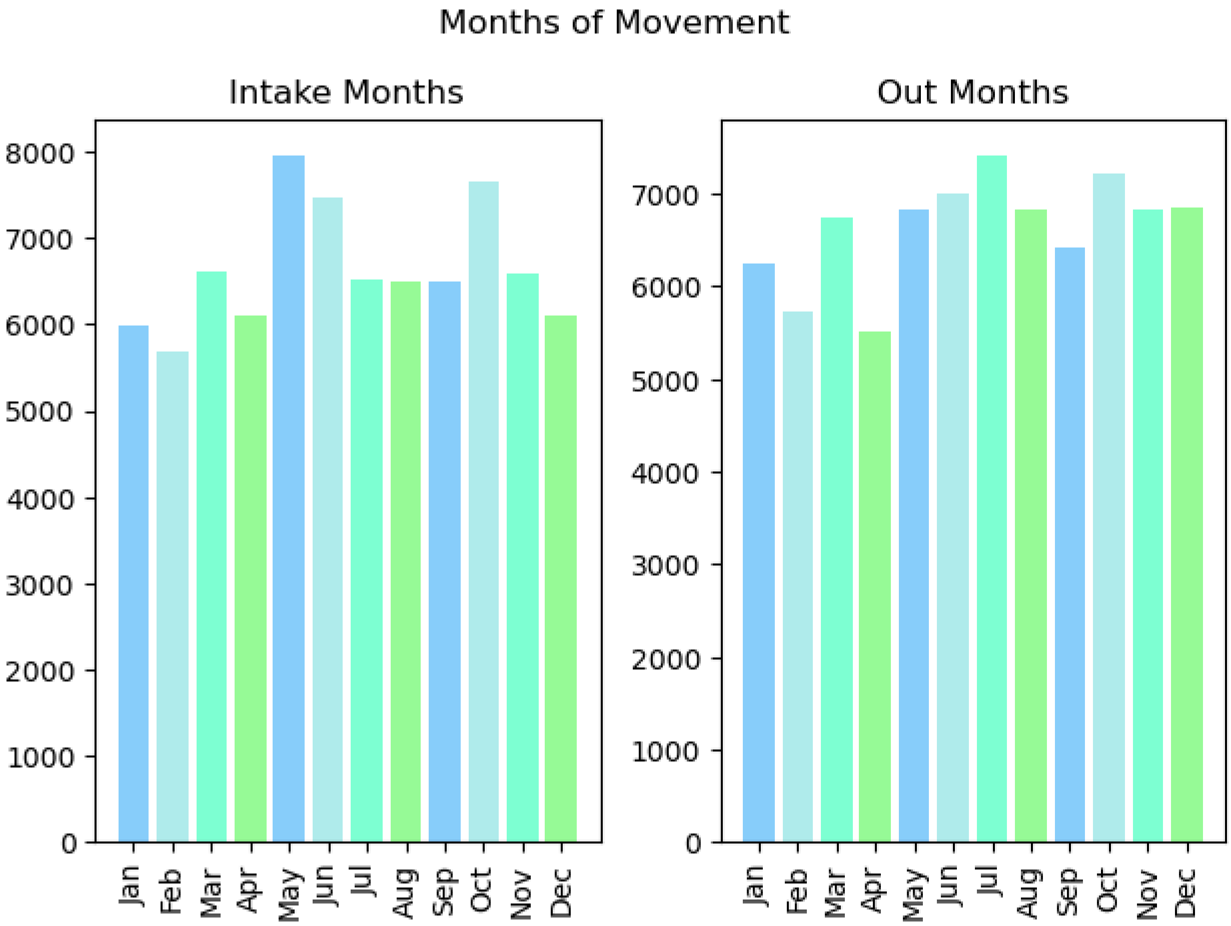
```
Text(0.5, 1.0, 'Movement Reasons')
```



Most animals leave the shelter because they are adopted or less likely, fostered.

In [108…

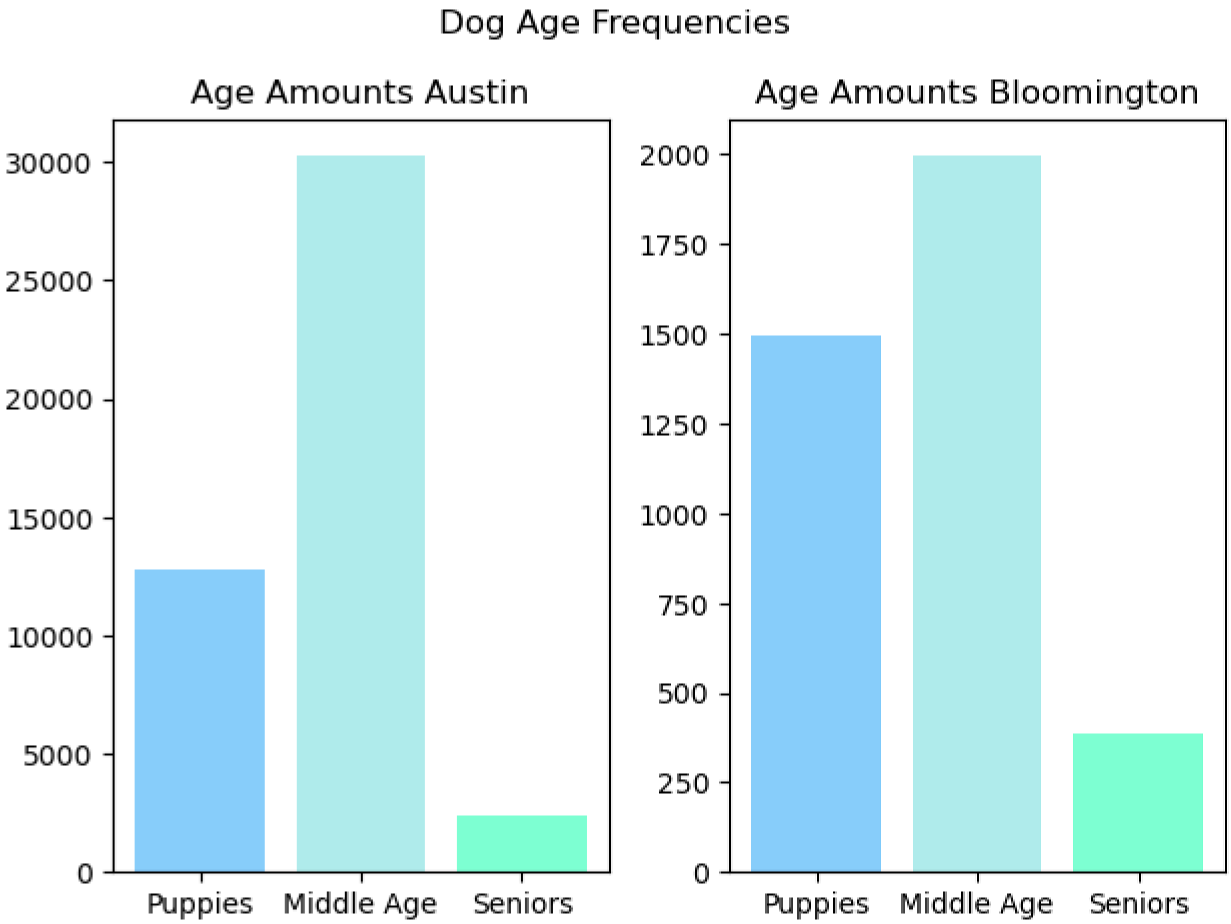
```
plt.suptitle("Months of Movement")
plt.subplot(1,2,1)
plt.bar(months,month_intake_amt_austin, color=colors)
plt.title("Intake Months")
plt.xticks(rotation=90)
plt.subplot(1,2,2)
plt.bar(months,month_out_amt_austin, color=colors)
plt.title("Out Months")
plt.xticks(rotation=90)
plt.tight_layout()
```



There appears to be a spike in animals taken into the shelter in May, June, and October and a very slight spike in adoptions/movements out of the shelter in July.

In [111…

```
plt.suptitle("Dog Age Frequencies")
plt.subplot(1,2,1)
plt.bar(age_classes, age_amounts_austin, color=colors)
plt.title("Age Amounts Austin")
plt.subplot(1,2,2)
plt.bar(age_classes, age_amounts_bloom, color=colors)
plt.title("Age Amounts Bloomington")
plt.tight_layout()
```



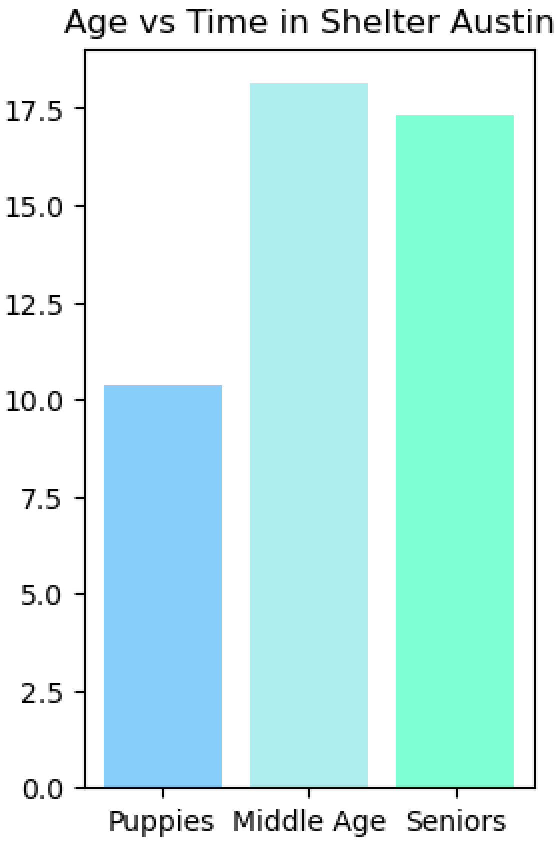
Seniors consistently make up the least of the population in both shelters and middle aged dogs between 1 and 9 make up the most. The population consisting of puppies varies more but still make up a semi large portion of the population.

In [114…

```
plt.subplot(1,2,1)
plt.bar(age_classes, age_times_austin, color=colors)
plt.title("Age vs Time in Shelter Austin")
```

Out[114…

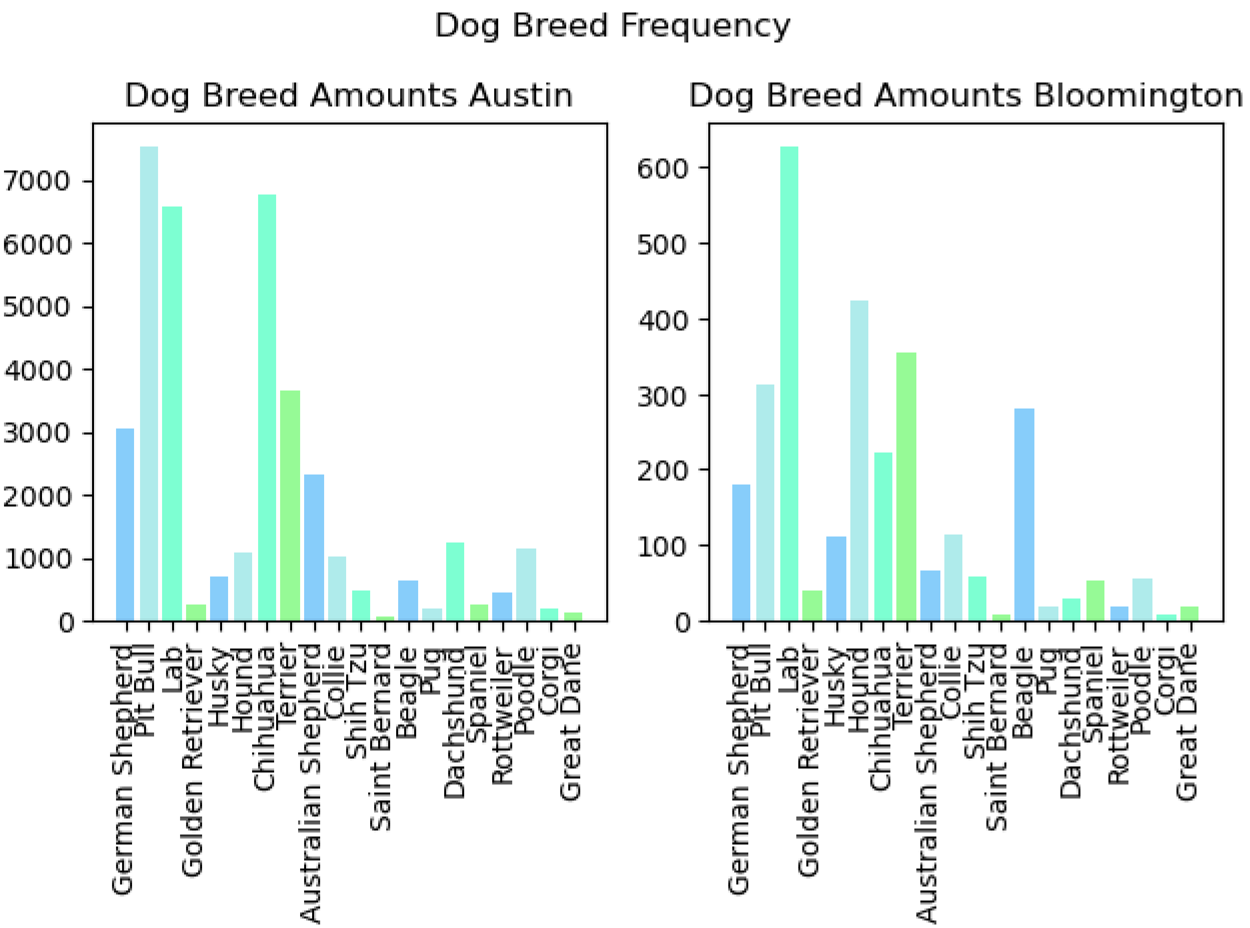
```
Text(0.5, 1.0, 'Age vs Time in Shelter Austin')
```



Despite seniors being the least frequent age group, they spend the longest time in the shelter. This could be related to the fact that many people do not want to adopt an older dog because that would mean less time spent with their pet. Puppies spend the least amount of time in the shelter which makes sense as well as most people when looking to adopt, are aiming to adopt a puppy.

In [117…

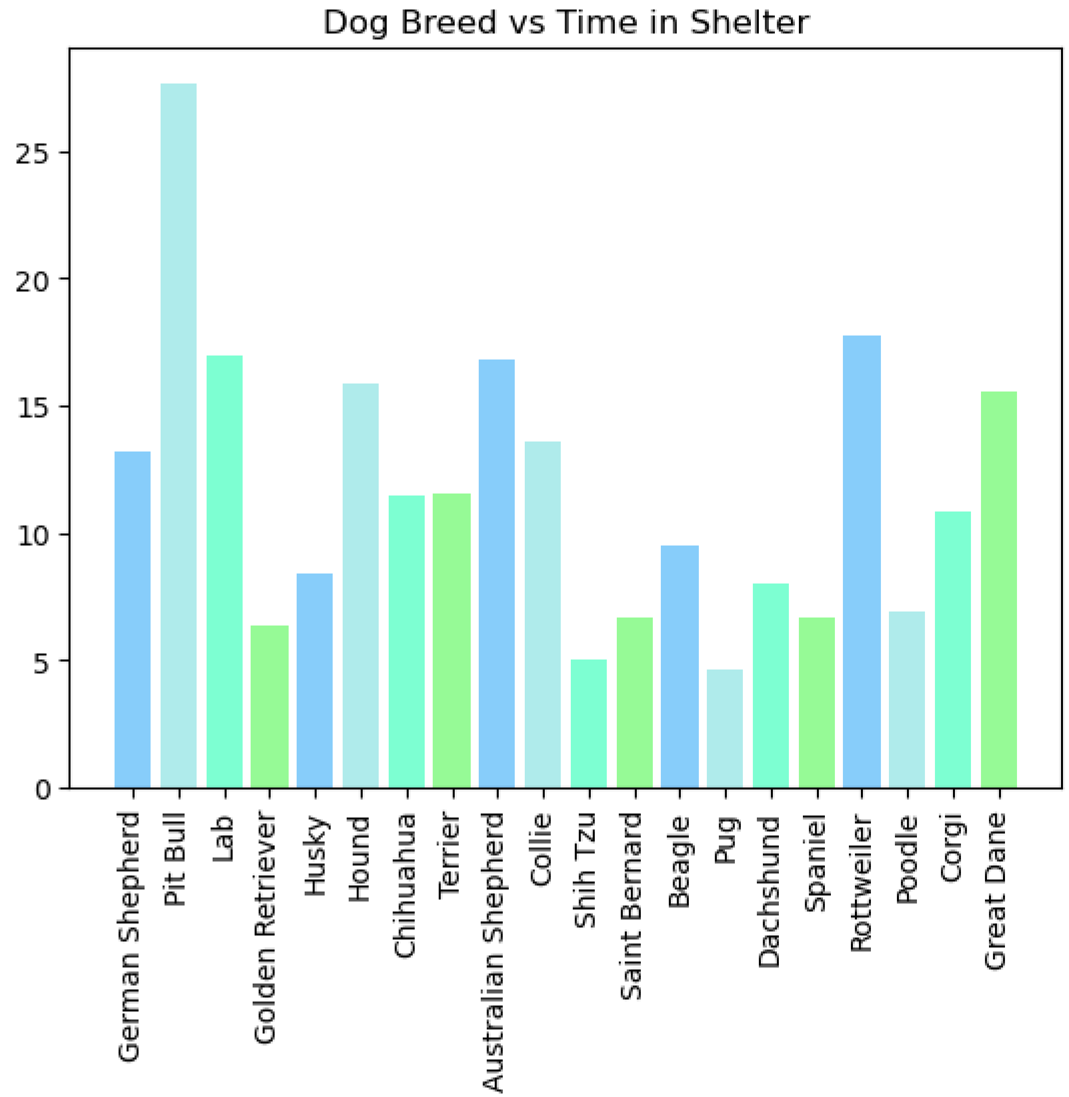
```
plt.suptitle("Dog Breed Frequency")
plt.subplot(1,2,1)
plt.bar(breed_names, breed_amounts_austin, color=colors)
plt.xticks(rotation=90)
plt.title("Dog Breed Amounts Austin")
plt.subplot(1,2,2)
plt.bar(breed_names, breed_amounts, color=colors)
plt.xticks(rotation=90)
plt.title("Dog Breed Amounts Bloomington")
plt.tight_layout()
```



Labs make up the majority of the breed population in the Bloomington Shelter and come second only to pit bulls in the Austin shelter. Chihuahuas appear often in the Austin Shelter while hounds, terriers, and beagles make up the next large chunk of dog breed in Bloomington. Great danes, corgis, golden retrievers, saint bernards, spaniels, and rottweiler's make up very little of the dog population in either shelter.

```
In [120…
plt.bar(breed_names, breed_times_austin, color=colors)
plt.xticks(rotation=90)
plt.title("Dog Breed vs Time in Shelter")
```

```
Out[120… Text(0.5, 1.0, 'Dog Breed vs Time in Shelter')
```



Pit bulls and rottweilers appear to spend the longest in the shelter in Austin which aligns with stereotypes associated with each breed. Pit bulls and rottweilers are bigger dogs and are paired with the idea of violence and aggression which could be contributing to the length of time they spend in the shelter.

Discussion and Conclusion

As shown in the charts above, there does not seem to be a strong correlation between the frequency of intakes/movements out of the shelter and the month in which those movements happen. However, there does appear to be a strong correlation between the breed and age of a dog and the time it spends in the shelter. Senior dogs spend the most time in the shelter despite there being the least amount of dogs that fall into that age range. Pit bulls and rottweilers, both considered agressive dog breeds, spend the most amount of time compared to other breeds waiting to be adopted. On average, pit bulls wait over 25 days to be adopted and rottweilers falling just under 20 while most other breeds fall between 10-15 days with many even less than that.

I would make a few changes to how I went about this analysis of shelters. First, I would do a more thorough check of my data, ensuring it is credible, reliable, and has all the information I would need to answer my research questions. I would also be more careful on my sample sizes and comparing shelters that are very different in capacity. Second, I would try to dig slightly deeper when trying to analyze the trends between certain demographics and how long they spend in the shelter or their frequencies. Finally, if I had more time, I would optimize my code more and utilize more functions. I used functions occassionally but if I had more time, I would go back and make a lot of the repeated code into functions to make the code easier to understand.

References

Moulton, Carol, et al. “PDF.” AVMA Publications, 1 Apr. 1991.

Schlegel, Aaron. “Austin Animal Center Shelter Intakes and Outcomes.” Kaggle, 9 Apr. 2018, www.kaggle.com/datasets/aaronschlegel/austin- animal-center-shelter-intakes-and-outcomes?select=aac_intakes_outcomes.csv.

Jin, Bonnie. “Animal Care and Control Adopted Animals.” Kaggle, 2 Nov. 2020, www.kaggle.com/datasets/jinbonnie/animal-data.