

```
In [4]: import pandas as pd
import numpy as np
```

```
In [6]: UVDAT = pd.read_csv('U:/Documents/Gunn Notes/Data Analyst Training/CELL/modified_uv
erse_data.txt', sep="|", header=None, low_memory=False, names=['PERSON_GENDER_CD', '
PERSON_AGE_NBR', 'HOUSEHOLD_INCOME_ID', 'DOG_WHISPERER', 'DOGS_101', 'CATS_101', 'BAD_DO
G', 'PUPPIES_VS_BABIES', 'PUPPY_BOWL'])
```

```
In [7]: print(UVDAT.dtypes)
```

```
PERSON_GENDER_CD      object
PERSON_AGE_NBR         float64
HOUSEHOLD_INCOME_ID   object
DOG_WHISPERER          int64
DOGS_101               int64
CATS_101               int64
BAD_DOG                int64
PUPPIES_VS_BABIES      int64
PUPPY_BOWL             int64
dtype: object
```

```
In [8]: UVDAT.describe()
```

Out [8]:

| | PERSON_AGE_NBR | DOG_WHISPERER | DOGS_101 | CATS_101 | BAD_DOG | PUPPIES_VS_BABIES |
|-------|----------------|---------------|--------------|--------------|--------------|-------------------|
| count | 3.092875e+06 | 4.041319e+06 | 4.041319e+06 | 4.041319e+06 | 4.041319e+06 | 4.041319e+06 |
| mean | 5.265317e+01 | 1.526425e-01 | 6.585498e-02 | 2.506409e-02 | 3.206626e-03 | 1.709442e-02 |
| std | 1.490502e+01 | 3.596426e-01 | 2.480285e-01 | 1.563198e-01 | 5.653622e-02 | 1.296233e-01 |
| min | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 25% | 4.100000e+01 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 50% | 5.200000e+01 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 75% | 6.300000e+01 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| max | 1.050000e+02 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 |

```
In [ ]: #UVDAT.sum(axis = 0, skipna = True)
```

```
In [9]: UVDAT.head()
```

Out [9]:

| | PERSON_GENDER_CD | PERSON_AGE_NBR | HOUSEHOLD_INCOME_ID | DOG_WHISPERER | DOGS_101 | CATS_101 |
|---|------------------|----------------|---------------------|---------------|----------|----------|
| 0 | 1 | 47.0 | | C | 1 | 0 |
| 1 | 1 | 51.0 | | D | 0 | 0 |
| 2 | 1 | 26.0 | | 9 | 0 | 0 |
| 3 | 2 | 41.0 | | 9 | 0 | 0 |
| 4 | 2 | 48.0 | | A | 0 | 0 |

****FORMATTING THE DATA****

```
In [10]: UVDAT.isnull().sum()
```

```
Out[10]: PERSON_GENDER_CD      800215  
PERSON_AGE_NBR      948444  
HOUSEHOLD_INCOME_ID  730856  
DOG_WHISPERER        0  
DOGS_101             0  
CATS_101             0  
BAD_DOG              0  
PUPPIES_VS_BABIES    0  
PUPPY_BOWL           0  
dtype: int64
```

```
In [11]: UVDAT = UVDAT.dropna()
```

```
In [12]: UVDAT.isnull().sum()
```

```
Out[12]: PERSON_GENDER_CD      0  
PERSON_AGE_NBR      0  
HOUSEHOLD_INCOME_ID    0  
DOG_WHISPERER        0  
DOGS_101             0  
CATS_101             0  
BAD_DOG              0  
PUPPIES_VS_BABIES    0  
PUPPY_BOWL           0  
dtype: int64
```

```
In [13]: UVDAT['PERSON_GENDER_CD'] = UVDAT['PERSON_GENDER_CD'].replace('1', 'Male')  
UVDAT['PERSON_GENDER_CD'] = UVDAT['PERSON_GENDER_CD'].replace('2', 'Female')  
UVDAT['PERSON_GENDER_CD'] = UVDAT['PERSON_GENDER_CD'].replace('3', 'Unknown')  
UVDAT['PERSON_GENDER_CD'] = UVDAT['PERSON_GENDER_CD'].replace('*', 'Unknown')
```

```
In [14]: UVDAT['PERSON_AGE_NBR'] = UVDAT['PERSON_AGE_NBR'].astype(np.int64)
```

```
In [15]: print(UVDAT.dtypes)
```

```
PERSON_GENDER_CD      object  
PERSON_AGE_NBR        int64  
HOUSEHOLD_INCOME_ID    object  
DOG_WHISPERER         int64  
DOGS_101              int64  
CATS_101              int64  
BAD_DOG               int64  
PUPPIES_VS_BABIES     int64  
PUPPY_BOWL            int64  
dtype: object
```

```
In [16]: print (UVDAT.columns)
         UVDAT.head(5)
```

```
Index(['PERSON_GENDER_CD', 'PERSON_AGE_NBR', 'HOUSEHOLD_INCOME_ID',
      'DOG_WHISPERER', 'DOGS_101', 'CATS_101', 'BAD_DOG', 'PUPPIES_VS_BABIES',
      'PUPPY_BOWL'],
      dtype='object')
```

Out [16]:

| | PERSON_GENDER_CD | PERSON_AGE_NBR | HOUSEHOLD_INCOME_ID | DOG_WHISPERER | DOGS_101 | CATS_101 |
|---|------------------|----------------|---------------------|---------------|----------|----------|
| 0 | Male | 47 | C | 1 | 0 | |
| 1 | Male | 51 | D | 0 | 0 | |
| 2 | Male | 26 | 9 | 0 | 0 | |
| 3 | Female | 41 | 9 | 0 | 0 | |
| 4 | Female | 48 | A | 0 | 0 | |

```
In [17]: UVDAT.describe()
```

Out [17]:

| | PERSON_AGE_NBR | DOG_WHISPERER | DOGS_101 | CATS_101 | BAD_DOG | PUPPIES_VS_BABIES |
|-------|----------------|---------------|--------------|--------------|--------------|-------------------|
| count | 3.055774e+06 | 3.055774e+06 | 3.055774e+06 | 3.055774e+06 | 3.055774e+06 | 3.055774e+06 |
| mean | 5.272658e+01 | 1.667568e-01 | 7.156387e-02 | 2.708054e-02 | 3.419428e-03 | 1.858809e-02 |
| std | 1.488195e+01 | 3.727586e-01 | 2.577644e-01 | 1.623182e-01 | 5.837582e-02 | 1.350651e-01 |
| min | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 25% | 4.100000e+01 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 50% | 5.200000e+01 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 75% | 6.300000e+01 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| max | 1.050000e+02 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 |

```
In [18]: UVDAT.PERSON_GENDER_CD.unique()
```

```
Out [18]: array(['Male', 'Female', 'Unknown'], dtype=object)
```

```
In [19]: UVDAT.HOUSEHOLD_INCOME_ID.unique()
```

```
Out [19]: array(['C', 'D', '9', 'A', '7', 'B', '6', '3', '8', '5', '4', '1', '2'],
              dtype=object)
```

```
In [20]: #Valid TV Shows:
         # 1 = Positive Observation
         # 0 = Negative Observation
```

QUESTIONS

****Question 1: How many viewers have the income code 4?**** Answer: 204,865 viewers

```
In [21]: print (len(UVDAT[UVDAT['HOUSEHOLD_INCOME_ID'] == '4']))
```

204865

```
In [22]: UVDAT['HOUSEHOLD_INCOME_ID'].value_counts()
```

```
Out [22]: 6      599827
          7      512761
          8      386581
          9      291091
          A      257714
          5      256840
          4      204865
          3      154941
          B      130628
          1       98269
          C       81433
          2       53919
          D       26905
          Name: HOUSEHOLD_INCOME_ID, dtype: int64
```

****Question 2: How many females are in the population? Answer: 1,481,375 females****

```
In [23]: print (len(UVDAT[UVDAT['PERSON_GENDER_CD'] == 'Female']))
```

```
1481375
```

```
In [24]: UVDAT['PERSON_GENDER_CD'].value_counts()
```

```
Out [24]: Male      1574334
          Female    1481375
          Unknown      65
          Name: PERSON_GENDER_CD, dtype: int64
```

****Question 3: What is the median age?** Answer: Median age is 52**

```
In [25]: UVDAT_filtered_AGE = UVDAT.PERSON_AGE_NBR[UVDAT['PERSON_AGE_NBR'] > 0]
```

```
In [26]: UVDAT_filtered_AGE.describe()
```

```
Out [26]: count      3.055199e+06
          mean      5.273651e+01
          std       1.486576e+01
          min       1.800000e+01
          25%       4.100000e+01
          50%       5.200000e+01
          75%       6.300000e+01
          max       1.050000e+02
          Name: PERSON_AGE_NBR, dtype: float64
```

```
In [27]: #UVDAT_filtered_AGE.sort_values(ascending = True)
```

****Question 4: What is the mode of the income level code?** Answer: Mode of income level code is 6**

```
In [28]: UVDAT_filtered_INCOME = UVDAT['HOUSEHOLD_INCOME_ID']
```

```
In [29]: UVDAT_filtered_INCOME.unique()
```

```
Out [29]: array(['C', 'D', '9', 'A', '7', 'B', '6', '3', '8', '5', '4', '1', '2'],
              dtype=object)
```

```
In [30]: UVDAT_filtered_INCOME.value_counts()
```

```
Out [30]: 6      599827
          7      512761
          8      386581
          9      291091
          A      257714
          5      256840
          4      204865
          3      154941
          B      130628
          1       98269
          C       81433
          2       53919
          D       26905
          Name: HOUSEHOLD_INCOME_ID, dtype: int64
```

****Question 5: How many viewers are under the age of 18?**** Answer: There are no viewers under age 18

```
In [31]: UVDAT_filtered_UNDERAGE = UVDAT[(UVDAT.PERSON_AGE_NBR > 0) & (UVDAT.PERSON_AGE_NBR < 18)]
          UVDAT_filtered_UNDERAGE.count()
```

```
Out [31]: PERSON_GENDER_CD      0
          PERSON_AGE_NBR      0
          HOUSEHOLD_INCOME_ID  0
          DOG_WHISPERER      0
          DOGS_101           0
          CATS_101           0
          BAD_DOG            0
          PUPPIES_VS_BABIES  0
          PUPPY_BOWL         0
          dtype: int64
```

```
In [32]: UVDAT_filtered_UNDERAGE.describe()
```

```
Out [32]:
```

| | PERSON_AGE_NBR | DOG_WHISPERER | DOGS_101 | CATS_101 | BAD_DOG | PUPPIES_VS_BABIES | PUF |
|-------|----------------|---------------|----------|----------|---------|-------------------|-----|
| count | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| mean | NaN | NaN | NaN | NaN | NaN | NaN | |
| std | NaN | NaN | NaN | NaN | NaN | NaN | |
| min | NaN | NaN | NaN | NaN | NaN | NaN | |
| 25% | NaN | NaN | NaN | NaN | NaN | NaN | |
| 50% | NaN | NaN | NaN | NaN | NaN | NaN | |
| 75% | NaN | NaN | NaN | NaN | NaN | NaN | |
| max | NaN | NaN | NaN | NaN | NaN | NaN | |

Predictive Models

Regression

****1. Create a linear model showing the relationship between a viewer's age and the total number of TV shows watched.****

```
In [33]: UVDAT['All_SHOWS'] = UVDAT['DOG_WHISPERER'] + UVDAT['DOGS_101'] + UVDAT['CATS_101'] + UVDAT['BAD_DOG'] + UVDAT['PUPPIES_VS_BABIES'] + UVDAT['PUPPY_BOWL']
print (UVDAT.columns)
UVDAT.head(10)
```

```
Index(['PERSON_GENDER_CD', 'PERSON_AGE_NBR', 'HOUSEHOLD_INCOME_ID',
      'DOG_WHISPERER', 'DOGS_101', 'CATS_101', 'BAD_DOG', 'PUPPIES_VS_BABIES',
      'PUPPY_BOWL', 'All_SHOWS'],
      dtype='object')
```

Out [33]:

| | PERSON_GENDER_CD | PERSON_AGE_NBR | HOUSEHOLD_INCOME_ID | DOG_WHISPERER | DOGS_101 | CATS_101 |
|---|------------------|----------------|---------------------|---------------|----------|----------|
| 0 | Male | 47 | C | 1 | 0 | 0 |
| 1 | Male | 51 | D | 0 | 0 | 0 |
| 2 | Male | 26 | 9 | 0 | 0 | 0 |
| 3 | Female | 41 | 9 | 0 | 0 | 0 |
| 4 | Female | 48 | A | 0 | 0 | 0 |
| 5 | Male | 42 | 7 | 1 | 1 | 0 |
| 6 | Female | 60 | B | 1 | 1 | 0 |
| 7 | Male | 57 | 7 | 0 | 0 | 0 |
| 8 | Male | 44 | D | 0 | 0 | 0 |
| 9 | Male | 54 | 6 | 0 | 0 | 0 |

```
In [34]: UVDAT.loc[1:10, ['DOG_WHISPERER', 'DOGS_101', 'CATS_101', 'BAD_DOG', 'PUPPIES_VS_BABIES',
      'PUPPY_BOWL', 'All_SHOWS']]
```

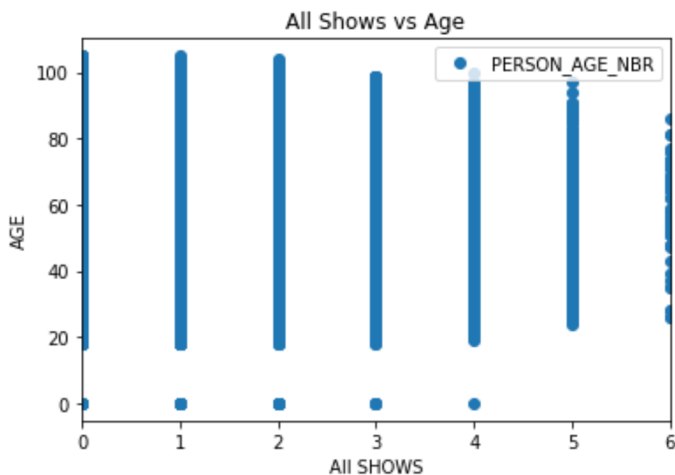
Out [34]:

| | DOG_WHISPERER | DOGS_101 | CATS_101 | BAD_DOG | PUPPIES_VS_BABIES | PUPPY_BOWL | All_SHOWS |
|----|---------------|----------|----------|---------|-------------------|------------|-----------|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 1 | 1 | 0 | 0 | 0 | 0 | 2 |
| 6 | 1 | 1 | 1 | 0 | 0 | 0 | 3 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

```
In [35]: import matplotlib.pyplot as plt
         %matplotlib inline
```

```
In [36]: plt.figure(figsize=(13,10))
         UVDAT.plot(x='All_SHOWS', y='PERSON_AGE_NBR', style='o')
         plt.title('All Shows vs Age')
         plt.xlabel('All SHOWS')
         plt.ylabel('AGE')
         plt.show()
```

<Figure size 936x720 with 0 Axes>



****Classification / Clustering****

****1. Create a cluster about each show (might be best to do a subset for each TV show) to show which age/age range is most likely to watch that show.****

```
In [37]: #copy columns into new dataframe
         UVDAT_VIEWERS = UVDAT[['PERSON_AGE_NBR', 'DOG_WHISPERER', 'DOGS_101', 'CATS_101', 'BAD_DOG',
                                'PUPPIES_VS_BABIES', 'PUPPY_BOWL']].copy()
```

```
In [38]: #review new dataframe columns and values
         UVDAT_VIEWERS.describe()
```

Out [38]:

| | PERSON_AGE_NBR | DOG_WHISPERER | DOGS_101 | CATS_101 | BAD_DOG | PUPPIES_VS_BABI |
|-------|----------------|---------------|--------------|--------------|--------------|-----------------|
| count | 3.055774e+06 | 3.055774e+06 | 3.055774e+06 | 3.055774e+06 | 3.055774e+06 | 3.055774e+06 |
| mean | 5.272658e+01 | 1.667568e-01 | 7.156387e-02 | 2.708054e-02 | 3.419428e-03 | 1.858809e-03 |
| std | 1.488195e+01 | 3.727586e-01 | 2.577644e-01 | 1.623182e-01 | 5.837582e-02 | 1.350651e-02 |
| min | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 25% | 4.100000e+01 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 50% | 5.200000e+01 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 75% | 6.300000e+01 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| max | 1.050000e+02 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 |

```
In [39]: # Get name of indexes for which person's age is equal 0
indexNames = UVDAT_VIEWERS[ UVDAT_VIEWERS['PERSON_AGE_NBR'] == 0 ].index
# Delete these row indexes from dataFrame
UVDAT_VIEWERS.drop(indexNames , inplace=True)
UVDAT_VIEWERS.describe()
```

Out [39]:

| | PERSON_AGE_NBR | DOG_WHISPERER | DOGS_101 | CATS_101 | BAD_DOG | PUPPIES_VS_BABI |
|-------|----------------|---------------|--------------|--------------|--------------|-----------------|
| count | 3.055199e+06 | 3.055199e+06 | 3.055199e+06 | 3.055199e+06 | 3.055199e+06 | 3.055199e+06 |
| mean | 5.273651e+01 | 1.667551e-01 | 7.156588e-02 | 2.708072e-02 | 3.419417e-03 | 1.858864e-03 |
| std | 1.486576e+01 | 3.727571e-01 | 2.577678e-01 | 1.623187e-01 | 5.837573e-02 | 1.350671e-02 |
| min | 1.800000e+01 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 25% | 4.100000e+01 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 50% | 5.200000e+01 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 75% | 6.300000e+01 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| max | 1.050000e+02 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 | 1.000000e+00 |

```
In [40]: UVDAT_VIEWERS['DOG_WHISPERER'] = UVDAT_VIEWERS['DOG_WHISPERER'].replace(1, 'DOG_WHISPERER')
UVDAT_VIEWERS['DOGS_101'] = UVDAT_VIEWERS['DOGS_101'].replace(1, 'DOGS_101')
UVDAT_VIEWERS['CATS_101'] = UVDAT_VIEWERS['CATS_101'].replace(1, 'CATS_101')
UVDAT_VIEWERS['BAD_DOG1'] = UVDAT_VIEWERS['BAD_DOG'].replace(1, 'BAD_DOG')
UVDAT_VIEWERS['PUPPIES_VS_BABIES'] = UVDAT_VIEWERS['PUPPIES_VS_BABIES'].replace(1, 'PUPPIES_VS_BABIES')
UVDAT_VIEWERS['PUPPY_BOWL'] = UVDAT_VIEWERS['PUPPY_BOWL'].replace(1, 'PUPPY_BOWL')
```

```
In [41]: # Create a function to assign age range
AgeRange = []
for row in UVDAT_VIEWERS['PERSON_AGE_NBR']:
    if row >= 90: AgeRange.append('90 and above')
    elif row >= 75: AgeRange.append('75-89')
    elif row >= 60: AgeRange.append('60-74')
    elif row >= 45: AgeRange.append('45-59')
    elif row >= 30: AgeRange.append('30-44')
    elif row >= 15: AgeRange.append('15-29')
    else: AgeRange.append('Failed')
```



```
In [42]: # Create a column from the list
UVDAT_VIEWERS['AgeRange'] = AgeRange
# View dataframe with new column
UVDAT_VIEWERS.head(10)
```

```
Out [42]:
```

| | PERSON_AGE_NBR | DOG_WHISPERER | DOGS_101 | CATS_101 | BAD_DOG | PUPPIES_VS_BABIES | PUPPY_ |
|---|----------------|---------------|----------|----------|---------|-------------------|--------|
| 0 | 47 | DOG_WHISPERER | 0 | 0 | 0 | | 0 |
| 1 | 51 | 0 | 0 | 0 | 0 | | 0 |
| 2 | 26 | 0 | 0 | 0 | 0 | | 0 |
| 3 | 41 | 0 | 0 | 0 | 0 | | 0 |
| 4 | 48 | 0 | 0 | 0 | 0 | | 0 |
| 5 | 42 | DOG_WHISPERER | DOGS_101 | 0 | 0 | | 0 |
| 6 | 60 | DOG_WHISPERER | DOGS_101 | CATS_101 | 0 | | 0 |
| 7 | 57 | 0 | 0 | 0 | 0 | | 0 |
| 8 | 44 | 0 | 0 | 0 | 0 | | 0 |
| 9 | 54 | 0 | 0 | 0 | 0 | | 0 |

```
In [43]: #test checking to see if correct number of shows has been assinged to DOG_WHISPERER column
TEST = UVDAT_VIEWERS[UVDAT_VIEWERS.DOG_WHISPERER == 'DOG_WHISPERER']
TEST.DOG_WHISPERER.value_counts()
```

```
Out [43]: DOG_WHISPERER    509470
Name: DOG_WHISPERER, dtype: int64
```

```
In [44]: #moving show data from columns to rows
UVDAT_VIEWERS_PIVOT = UVDAT_VIEWERS.melt(id_vars=["PERSON_AGE_NBR", "AgeRange"],
      var_name="Delete",
      value_name="Value")
```

```
In [45]: UVDAT_VIEWERS_PIVOT.head(10)
```

```
Out [45]:
```

| | PERSON_AGE_NBR | AgeRange | Delete | Value |
|---|----------------|----------|---------------|---------------|
| 0 | 47 | 45-59 | DOG_WHISPERER | DOG_WHISPERER |
| 1 | 51 | 45-59 | DOG_WHISPERER | 0 |
| 2 | 26 | 15-29 | DOG_WHISPERER | 0 |
| 3 | 41 | 30-44 | DOG_WHISPERER | 0 |
| 4 | 48 | 45-59 | DOG_WHISPERER | 0 |
| 5 | 42 | 30-44 | DOG_WHISPERER | DOG_WHISPERER |
| 6 | 60 | 60-74 | DOG_WHISPERER | DOG_WHISPERER |
| 7 | 57 | 45-59 | DOG_WHISPERER | 0 |
| 8 | 44 | 30-44 | DOG_WHISPERER | 0 |
| 9 | 54 | 45-59 | DOG_WHISPERER | 0 |

```
In [46]: #checking values in the Value column
UVDAT_VIEWERS_PIVOT.Value.unique()
```

```
Out [46]: array(['DOG_WHISPERER', 0, 'DOGS_101', 'CATS_101', 1, 'PUPPIES_VS_BABIES',
                'PUPPY_BOWL', 'BAD_DOG'], dtype=object)
```

```
In [47]: #test checking to see if correct number of shows has been assinged to DOG_WHISPERER column
TEST1 = UVDAT_VIEWERS_PIVOT[UVDAT_VIEWERS_PIVOT.Value == 'DOG_WHISPERER']
TEST1.Value.value_counts()
```

```
Out [47]: DOG_WHISPERER    509470
Name: Value, dtype: int64
```

```
In [48]: #Delete "0L" values in Value column
# Get name of indexes
indexNames = UVDAT_VIEWERS_PIVOT[UVDAT_VIEWERS_PIVOT['Value'] == 0].index
# Delete these row indexes from dataFrame
UVDAT_VIEWERS_PIVOT.drop(indexNames , inplace=True)
UVDAT_VIEWERS_PIVOT.Value.unique()
```

```
Out [48]: array(['DOG_WHISPERER', 'DOGS_101', 'CATS_101', 1, 'PUPPIES_VS_BABIES',
                'PUPPY_BOWL', 'BAD_DOG'], dtype=object)
```

```
In [49]: #Delete "1L" values in Value column
# Get name of indexes
indexNames = UVDAT_VIEWERS_PIVOT[UVDAT_VIEWERS_PIVOT['Value'] == 1].index
# Delete these row indexes from dataFrame
UVDAT_VIEWERS_PIVOT.drop(indexNames , inplace=True)
UVDAT_VIEWERS_PIVOT.Value.unique()
```

```
Out [49]: array(['DOG_WHISPERER', 'DOGS_101', 'CATS_101', 'PUPPIES_VS_BABIES',
                'PUPPY_BOWL', 'BAD_DOG'], dtype=object)
```

```
In [50]: #delete column "Delete"
del UVDAT_VIEWERS_PIVOT['Delete']
```

```
In [51]: #delete column "PERSON_AGE_NBR"
del UVDAT_VIEWERS_PIVOT['PERSON_AGE_NBR']
```

```
In [52]: UVDAT_VIEWERS_PIVOT.head(10)
```

Out [52]:

| | AgeRange | Value |
|----|----------|---------------|
| 0 | 45-59 | DOG_WHISPERER |
| 5 | 30-44 | DOG_WHISPERER |
| 6 | 60-74 | DOG_WHISPERER |
| 17 | 45-59 | DOG_WHISPERER |
| 24 | 60-74 | DOG_WHISPERER |
| 26 | 60-74 | DOG_WHISPERER |
| 30 | 45-59 | DOG_WHISPERER |
| 36 | 60-74 | DOG_WHISPERER |
| 37 | 45-59 | DOG_WHISPERER |
| 48 | 45-59 | DOG_WHISPERER |

```
In [53]: #group by and store totals in new dataframe
UVDAT_VIEWERS_PIVOT1 = UVDAT_VIEWERS_PIVOT.groupby(['AgeRange', 'Value']).size().unstack('V
alue', fill_value=0)
UVDAT_VIEWERS_PIVOT1.head(15)
```

Out [53]:

| | Value | BAD_DOG | CATS_101 | DOGS_101 | DOG_WHISPERER | PUPPIES_VS_BABIES | PUPPY_BOWL |
|----------|--------------|---------|----------|----------|---------------|-------------------|------------|
| AgeRange | | | | | | | |
| | 15-29 | 321 | 2762 | 8786 | 17694 | 2160 | 697 |
| | 30-44 | 1968 | 21153 | 62866 | 129660 | 14924 | 3339 |
| | 45-59 | 3992 | 29868 | 83430 | 195191 | 20559 | 5752 |
| | 60-74 | 3219 | 21803 | 48649 | 129260 | 14114 | 3916 |
| | 75-89 | 871 | 6585 | 13684 | 34655 | 4616 | 1139 |
| | 90 and above | 76 | 566 | 1233 | 3010 | 419 | 104 |

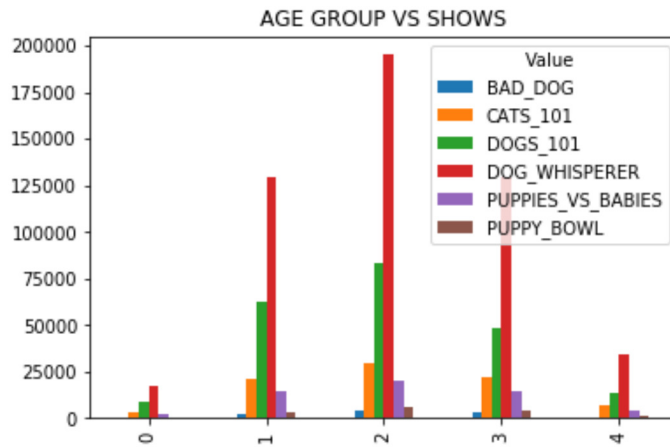
```
In [54]: #NOT REQUIRED TO RUN
UVDAT_VIEWERS_PIVOT1 = UVDAT_VIEWERS_PIVOT1.reset_index()
```

```
In [55]: #NOT REQUIRED TO RUN
UVDAT_VIEWERS_PIVOT1.head()
```

Out [55]:

| Value | AgeRange | BAD_DOG | CATS_101 | DOGS_101 | DOG_WHISPERER | PUPPIES_VS_BABIES | PUPPY_BOW |
|-------|----------|---------|----------|----------|---------------|-------------------|-----------|
| 0 | 15-29 | 321 | 2762 | 8786 | 17694 | 2160 | 69 |
| 1 | 30-44 | 1968 | 21153 | 62866 | 129660 | 14924 | 333 |
| 2 | 45-59 | 3992 | 29868 | 83430 | 195191 | 20559 | 575 |
| 3 | 60-74 | 3219 | 21803 | 48649 | 129260 | 14114 | 391 |
| 4 | 75-89 | 871 | 6585 | 13684 | 34655 | 4616 | 113 |

```
In [56]: UVDAT_VIEWERS_PIVOT1.head().plot(kind='bar', legend=True, title='AGE GROUP VS SHOWS')
from matplotlib.pyplot import figure
figure(num=None, figsize=(20, 4), dpi=50, facecolor='w', edgecolor='k', frameon=True, clear=False);
```



<Figure size 1000x200 with 0 Axes>

TESTING TESTNG

```
In [57]: import seaborn as sns
import matplotlib.pyplot as plt
```

```
In [59]: #group by and store totals in new dataframeVa
mmmm = UVDAT_VIEWERS_PIVOT.groupby(['AgeRange', 'Value']).size()
mmmm.head(15)
```

```
Out [59]: AgeRange  Value
15-29      BAD_DOG           321
          CATS_101          2762
          DOGS_101          8786
          DOG_WHISPERER      17694
          PUPPIES_VS_BABIES    2160
          PUPPY_BOWL           697
30-44      BAD_DOG          1968
          CATS_101         21153
          DOGS_101         62866
          DOG_WHISPERER     129660
          PUPPIES_VS_BABIES   14924
          PUPPY_BOWL          3339
45-59      BAD_DOG          3992
          CATS_101         29868
          DOGS_101         83430
dtype: int64
```

```
In [60]: mmmm = mmmm.reset_index()
```

```
In [61]: mmmm.head(10)
```

Out [61]:

| | AgeRange | Value | 0 |
|---|----------|-------------------|--------|
| 0 | 15-29 | BAD_DOG | 321 |
| 1 | 15-29 | CATS_101 | 2762 |
| 2 | 15-29 | DOGS_101 | 8786 |
| 3 | 15-29 | DOG_WHISPERER | 17694 |
| 4 | 15-29 | PUPPIES_VS_BABIES | 2160 |
| 5 | 15-29 | PUPPY_BOWL | 697 |
| 6 | 30-44 | BAD_DOG | 1968 |
| 7 | 30-44 | CATS_101 | 21153 |
| 8 | 30-44 | DOGS_101 | 62866 |
| 9 | 30-44 | DOG_WHISPERER | 129660 |

```
In [62]: mmmm.rename(columns = {"Value": 'Shows'}, inplace = True)
mmm.rename(columns = {0: 'Value'}, inplace = True)
mmm.head()
```

Out [62]:

| | AgeRange | Shows | Value |
|---|----------|-------------------|-------|
| 0 | 15-29 | BAD_DOG | 321 |
| 1 | 15-29 | CATS_101 | 2762 |
| 2 | 15-29 | DOGS_101 | 8786 |
| 3 | 15-29 | DOG_WHISPERER | 17694 |
| 4 | 15-29 | PUPPIES_VS_BABIES | 2160 |

```
In [63]: #UVDAT_VIEWERS_PIVOT
#UVDAT_VIEWERS_PIVOT1.head().plot(kind='bar')
g = sns.FacetGrid(mmmm, col="Shows", size=3)
g.map(plt.scatter, "AgeRange", "Value", cmap="viridis")
ax = plt.gca()
#plt.colorbar(label="Value")
#g.map(sns.barplot, 'AgeRange')
#g.map(sns.regplot, "AgeRange", "Value");
g.add_legend(title="Shows")
#g.set_xlabel("Age Range")
g.set_titles("{col_name}");

for pw in pws:
    plt.scatter([], [], s=(pw**2)*60, c="k", label=str(pw))

h, l = plt.gca().get_legend_handles_labels()
plt.legend(h[1:], l[1:], labelspace=1.2, title="shows_watched", borderpad=1,
           frameon=True, framealpha=0.6, edgecolor="k", facecolor="w")
plt.show()
```

P:\Anaconda\lib\site-packages\seaborn\axisgrid.py:230: UserWarning: The `size` parameter has been renamed to `height`; please update your code.

warnings.warn(msg, UserWarning)

NameError Traceback (most recent call last)

<ipython-input-63-c2f204459a19> in <module>

11 g.set_titles("{col_name}");

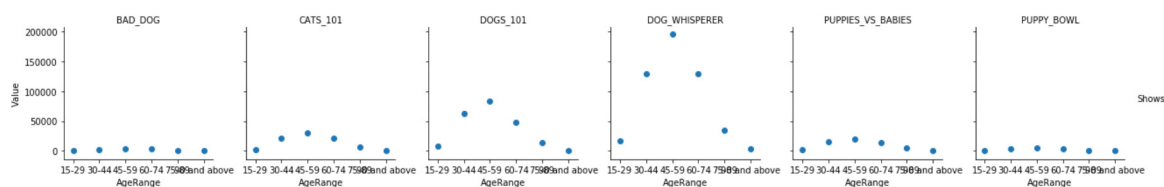
12

---> 13 for pw in pws:

14 plt.scatter([], [], s=(pw**2)*60, c="k", label=str(pw))

15

NameError: name 'pws' is not defined



****2. Create a cluster showing which gender code is more likely to have a certain income level. Hint: you can take a similar approach to question 1 for the graphs or you can find a way to put all income codes in 1 graph.****

```
In [64]: #copy columns with Income into new dataframe
UVDAT_INCOME = UVDAT[['PERSON_GENDER_CD', 'HOUSEHOLD_INCOME_ID']].copy()
```

In [65]: `UVDAT_INCOME.head(5)`

Out [65]:

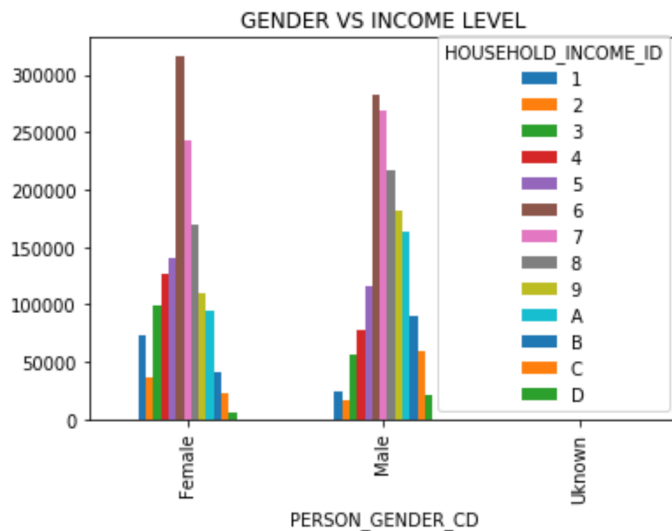
| | PERSON_GENDER_CD | HOUSEHOLD_INCOME_ID |
|---|------------------|---------------------|
| 0 | Male | C |
| 1 | Male | D |
| 2 | Male | 9 |
| 3 | Female | 9 |
| 4 | Female | A |

In [66]: `#group by and store totals in new dataframe`
`UVDAT_INCOME = UVDAT_INCOME.groupby(['PERSON_GENDER_CD', 'HOUSEHOLD_INCOME_ID']).size().unstack('HOUSEHOLD_INCOME_ID', fill_value=0)`
`UVDAT_INCOME.head(5)`

Out [66]:

| HOUSEHOLD_INCOME_ID | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A |
|---------------------|-------|-------|-------|--------|--------|--------|--------|--------|--------|--------|
| PERSON_GENDER_CD | | | | | | | | | | |
| Female | 73716 | 37204 | 98806 | 127143 | 140908 | 316668 | 243602 | 169856 | 109787 | 94025 |
| Male | 24546 | 16715 | 56129 | 77717 | 115928 | 283144 | 269146 | 216715 | 181303 | 163687 |
| Unknown | 7 | 0 | 6 | 5 | 4 | 15 | 13 | 10 | 1 | 2 |

In [67]: `UVDAT_INCOME.head().plot(kind='bar', legend=True, title='GENDER VS INCOME LEVEL')`
`from matplotlib.pyplot import figure`
`figure(num=None, figsize=(20, 4), dpi=50, facecolor='w', edgecolor='k', frameon=True, clear=False);`



<Figure size 1000x200 with 0 Axes>

In []:

In []:

In []:

In []:

In []: