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

In [ ]:
```
UVDAT = pd.read_csv('modified_uverse_data.txt', sep="|", header=None, low_memory=False, na
mes=['PERSON_GENDER_CD','PERSON_AGE_NBR','HOUSEHOLD_INCOME_ID','DOG_WHISPERER','DOGS_101
','CATS_101','BAD_DOG','PUPPIES_VS_BABIES', 'PUPPY_BOWL'])
```

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

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

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

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

FORMATTING THE DATA

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

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

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

In [ ]:
```
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', 'Uknown')
UVDAT['PERSON_GENDER_CD'] = UVDAT['PERSON_GENDER_CD'].replace('*', 'Uknown')
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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 [ ]:
```
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(15)
```

In [ ]:
```
UVDAT.loc[1:15, ['DOG_WHISPERER', 'DOGS_101', 'CATS_101', 'BAD_DOG', 'PUPPIES_VS_BABIES', 'PUPPY_
BOWL', 'All_SHOWS']]
```

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

In [ ]:
```
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()
```

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

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

In [ ]:
```
# 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()
```

In [ ]:
```
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_BA
BIES')
UVDAT_VIEWERS['PUPPY_BOWL'] = UVDAT_VIEWERS['PUPPY_BOWL'].replace(1, 'PUPPY_BOWL')
```

In [ ]:
```
# 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 [ ]:
```
# Create a column from the list
UVDAT_VIEWERS['AgeRange'] = AgeRange
# View dataframe with new column
UVDAT_VIEWERS.head(15)
```

In [ ]:
```
#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()
```

In [ ]:
```
#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 [ ]:
```
UVDAT_VIEWERS_PIVOT.head(15)
```

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

In [ ]:
```
#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()
```

In [ ]:
```
#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()
```

In [ ]:
```
#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()
```

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

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

In [ ]:
```
UVDAT_VIEWERS_PIVOT.head(15)
```

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

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

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

In [ ]:
```
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=Fals
e);
```

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

TESTING TESTNG

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

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

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

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

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

In [ ]:
```
#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_xlabels("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:], labelspacing=1.2, title="shows_watched", borderpad=1,
            frameon=True, framealpha=0.6, edgecolor="k", facecolor="w")
plt.show()
```

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

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

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

In [ ]:
```
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=Fals
e);
```