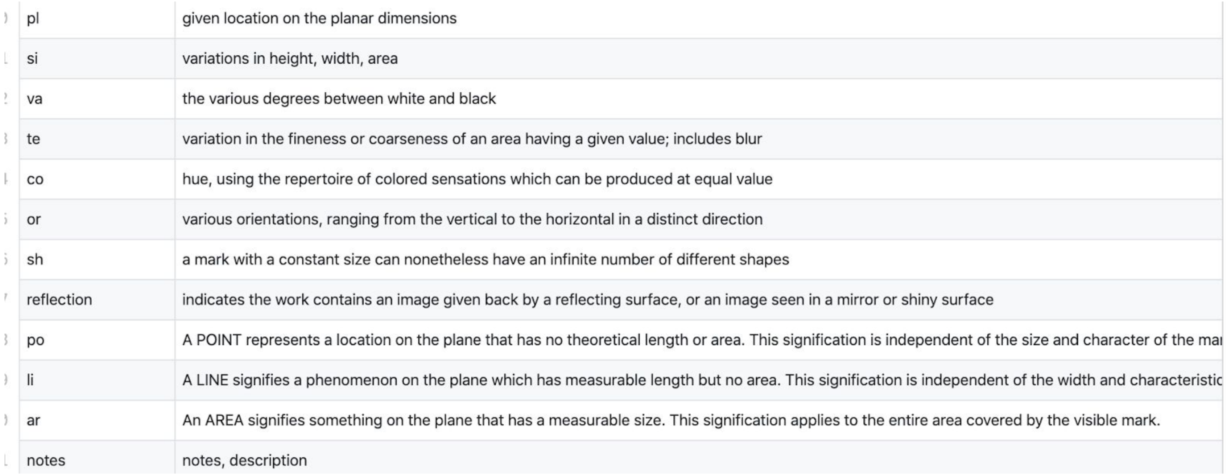
Machine Learning Assignment 3

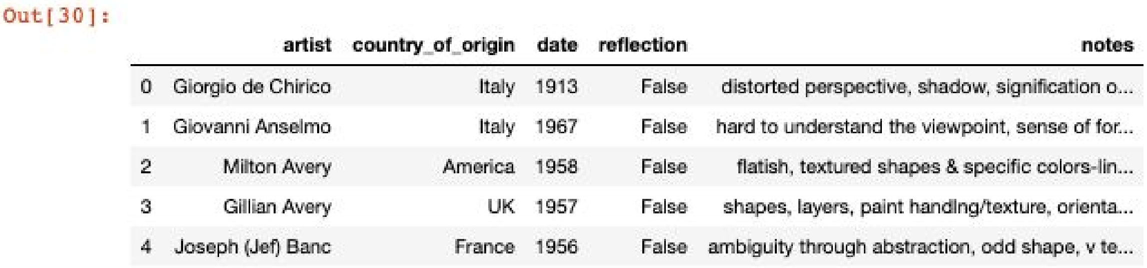
Lee Kuczewski
4.15.20-4.20.20
ML_Assignment3_Option2
First Draft

Looking at metadata for features:



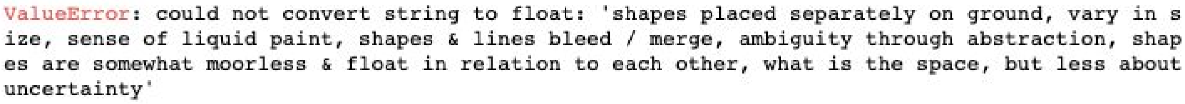
pl	given location on the planar dimensions
si	variations in height, width, area
va	the various degrees between white and black
te	variation in the fineness or coarseness of an area having a given value; includes blur
co	hue, using the repertoire of colored sensations which can be produced at equal value
or	various orientations, ranging from the vertical to the horizontal in a distinct direction
sh	a mark with a constant size can nonetheless have an infinite number of different shapes
reflection	indicates the work contains an image given back by a reflecting surface, or an image seen in a mirror or shiny surface
po	A POINT represents a location on the plane that has no theoretical length or area. This signification is independent of the size and character of the mai
li	A LINE signifies a phenomenon on the plane which has measurable length but no area. This signification is independent of the width and characteristic
ar	An AREA signifies something on the plane that has a measurable size. This signification applies to the entire area covered by the visible mark.
notes	notes, description

In order to read in the metadata, we will need to transform the datatype.



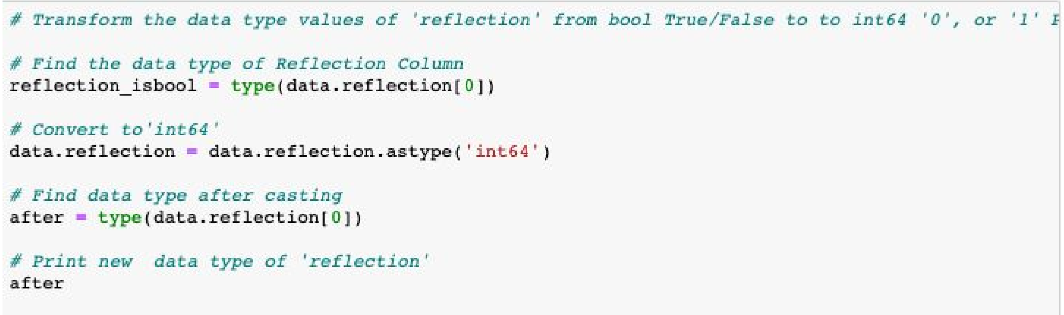
Out[30]:

	artist	country_of_origin	date	reflection	notes
0	Giorgio de Chirico	Italy	1913	False	distorted perspective, shadow, signification o...
1	Giovanni Anselmo	Italy	1967	False	hard to understand the viewpoint, sense of for...
2	Milton Avery	America	1958	False	flatish, textured shapes & specific colors-lin...
3	Gillian Avery	UK	1957	False	shapes, layers, paint handlng/texture, orienta...
4	Joseph (Jef) Banc	France	1956	False	ambiguity through abstraction, odd shape, v te...



```
ValueError: could not convert string to float: 'shapes placed separately on ground, vary in s
ize, sense of liquid paint, shapes & lines bleed / merge, ambiguity through abstraction, shap
es are somewhat moorless & float in relation to each other, what is the space, but less about
uncertainty'
```

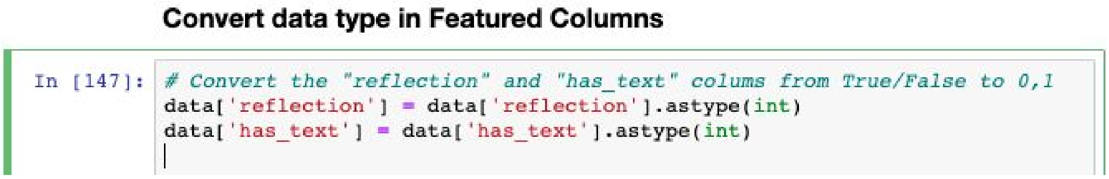
Part 1 to change data type of reflection from Bool to int64 using astype( )



```
# Transform the data type values of 'reflection' from bool True/False to to int64 '0', or '1' F

# Find the data type of Reflection Column
reflection_isbool = type(data.reflection[0])

# Convert to'int64'
data.reflection = data.reflection.astype('int64')

# Find data type after casting
after = type(data.reflection[0])

# Print new  data type of 'reflection'
after
```

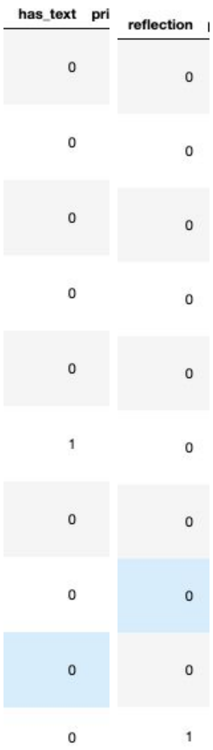
Part 2 Set (True, False) values to (1, 0)

Transform Data type for columns ‘reflection’ & ‘has_text’



Convert data type in Featured Columns

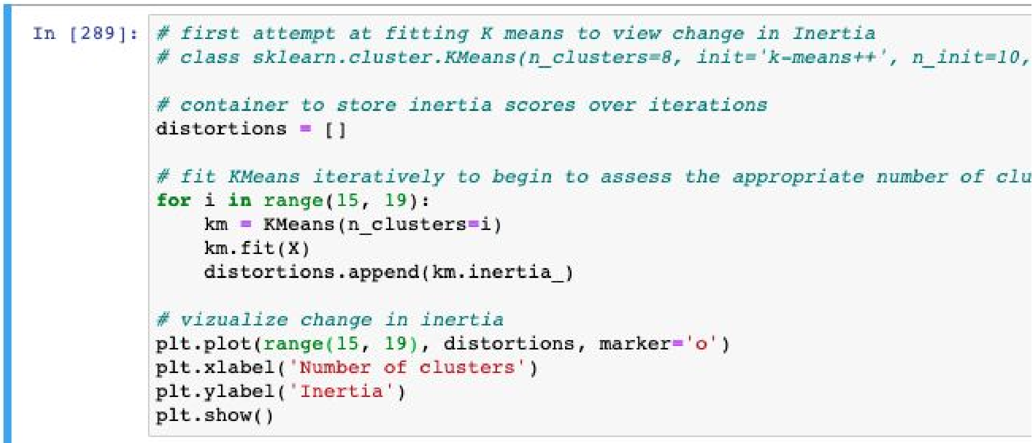
```
In [147]: # Convert the "reflection" and "has_text" colums from True/False to 0,1
          data['reflection'] = data['reflection'].astype(int)
          data['has_text'] = data['has_text'].astype(int)
```



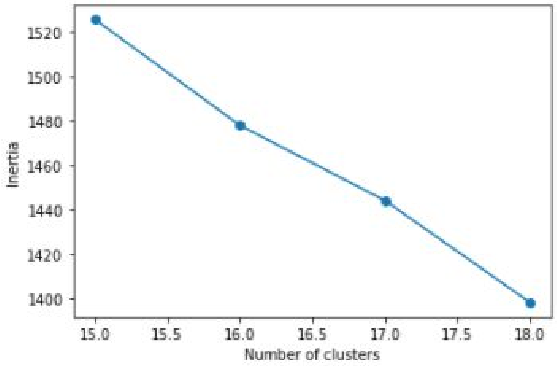
has_text	pri
0	
0	
0	
0	
0	
1	
0	
0	
0	
0	

reflection
0
0
0
0
0
0
0
0
0
1

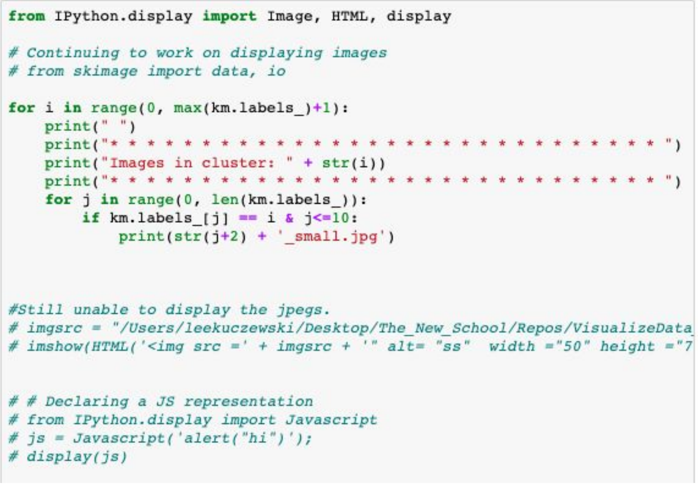
plot inertia scores by number of clusters



```
In [289]: # first attempt at fitting K means to view change in Inertia
          # class sklearn.cluster.KMeans(n_clusters=8, init='k-means++', n_init=10,

          # container to store inertia scores over iterations
          distortions = []

          # fit KMeans iteratively to begin to assess the appropriate number of clu
          for i in range(15, 19):
              km = KMeans(n_clusters=i)
              km.fit(X)
              distortions.append(km.inertia_)

          # vizualize change in inertia
          plt.plot(range(15, 19), distortions, marker='o')
          plt.xlabel('Number of clusters')
          plt.ylabel('Inertia')
          plt.show()
```



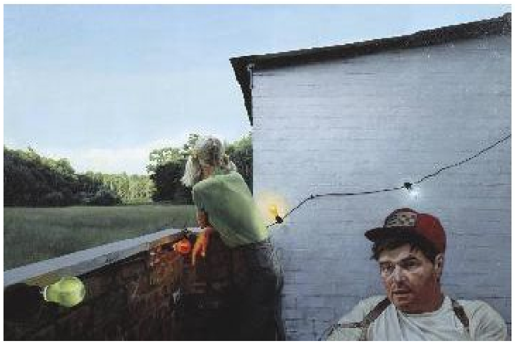
More Work is to be done on printing images, using IPython.display

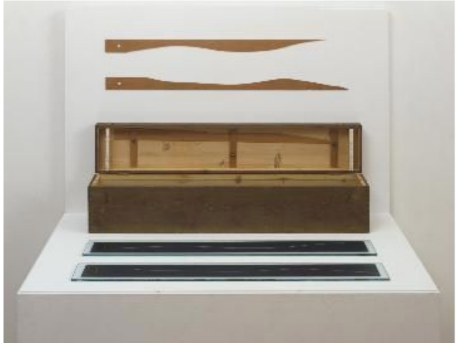


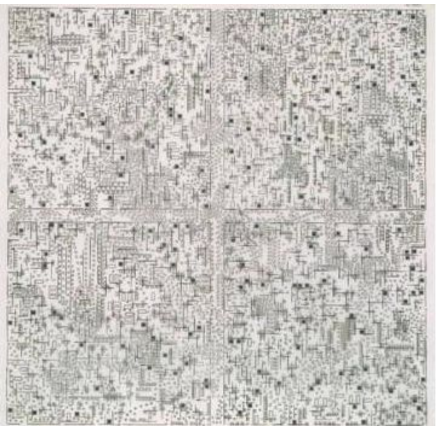
```
from IPython.display import Image, HTML, display

# Continuing to work on displaying images
# from skimage import data, io

for i in range(0, max(km.labels_)+1):
    print(" ")
    print("* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ")
    print("Images in cluster: " + str(i))
    print("* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ")
    for j in range(0, len(km.labels_)):
        if km.labels_[j] == i & j<=10:
            print(str(j+2) + '_small.jpg')



#Still unable to display the jpegs.
# imgsrc = "/Users/leekuczewski/Desktop/The_New_School/Repos/VisualizeData_
# imshow(HTML('<img src =' + imgsrc + '" alt= "ss"  width ="50" height ="7


# # Declaring a JS representation
# from IPython.display import Javascript
# js = Javascript('alert("hi")');
# display(js)
```

Many of the images are ending up in multiple clusters, so my next iteration will incorporate additional changes, where each artwork can only be clustered in one place, also it will be critical to see the data (images) and search for patterns using different features.

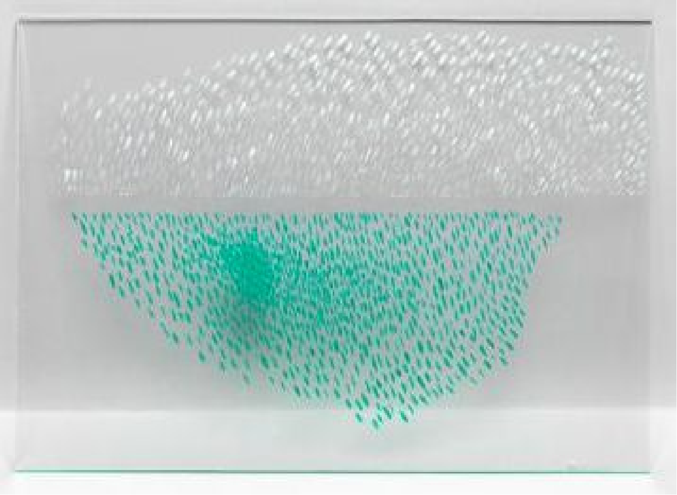
Cluster 0 Samples:



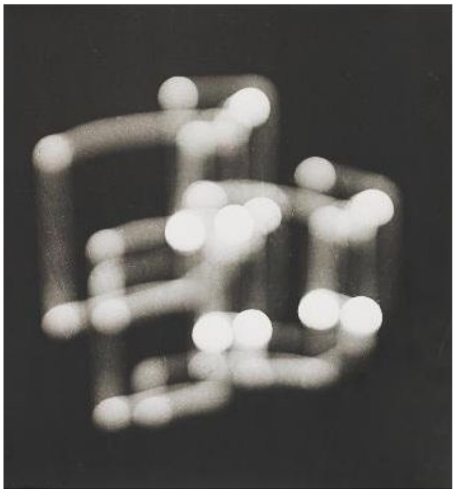




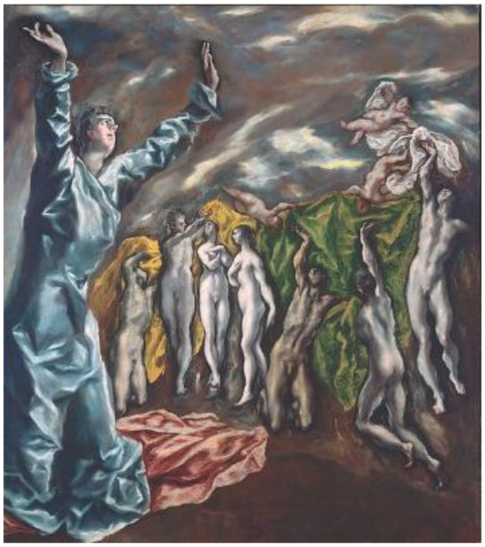
Cluster 1:

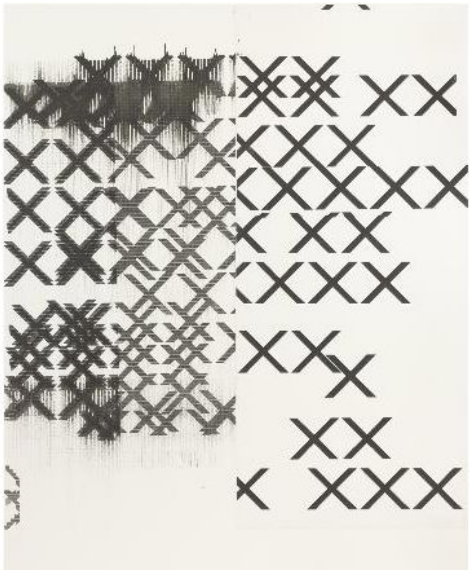


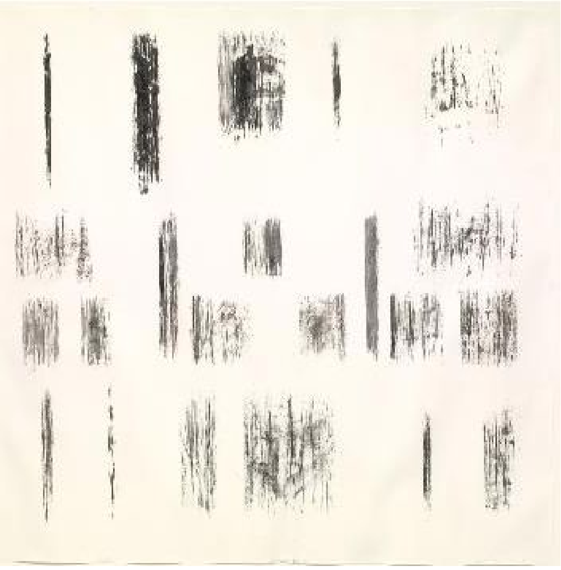
Cluster 2:

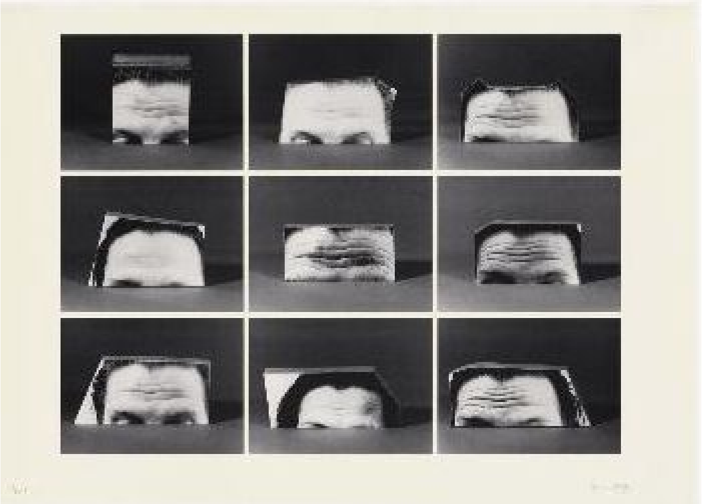


Cluster 3:

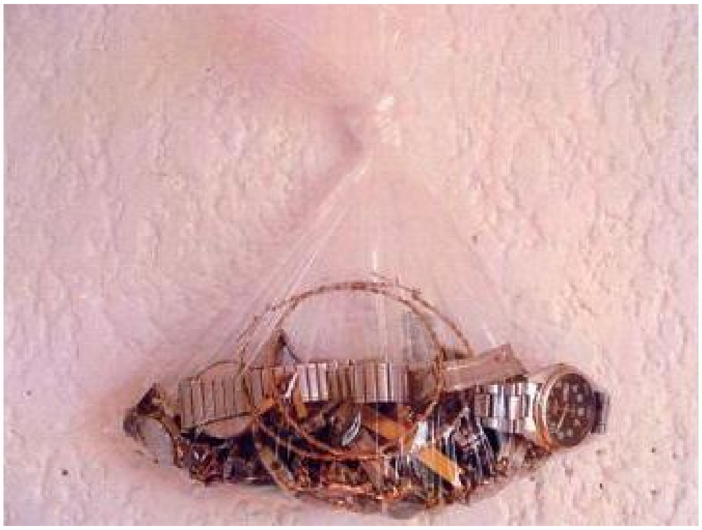


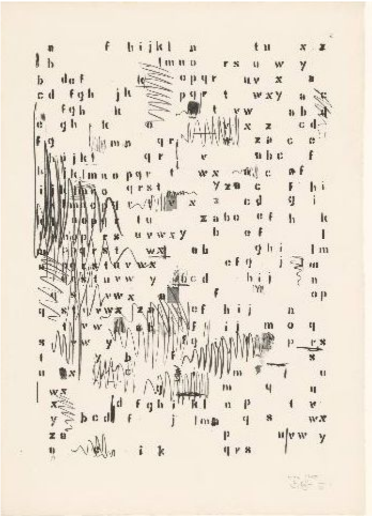






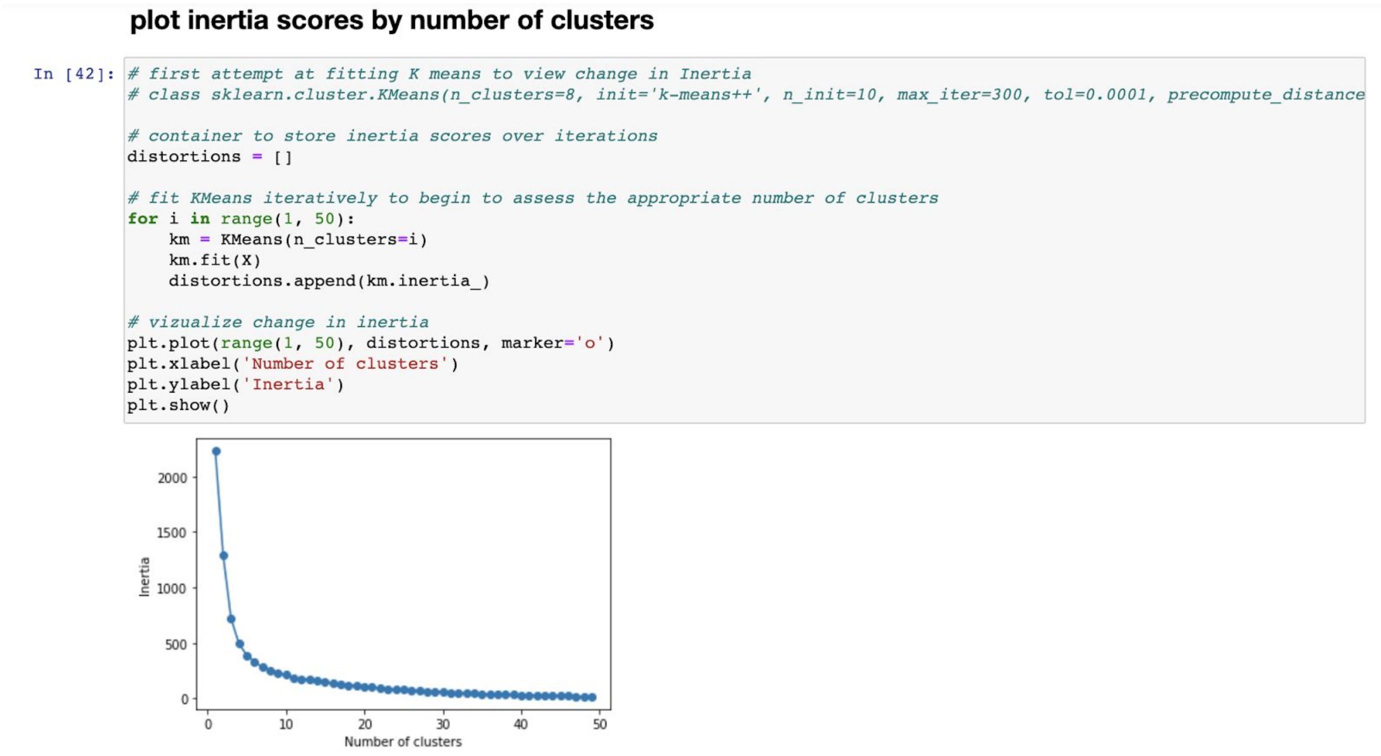
Cluster 4:





Continuation of Work 4/21/20- 5/3/20

Reconsidering the Number of Clusters

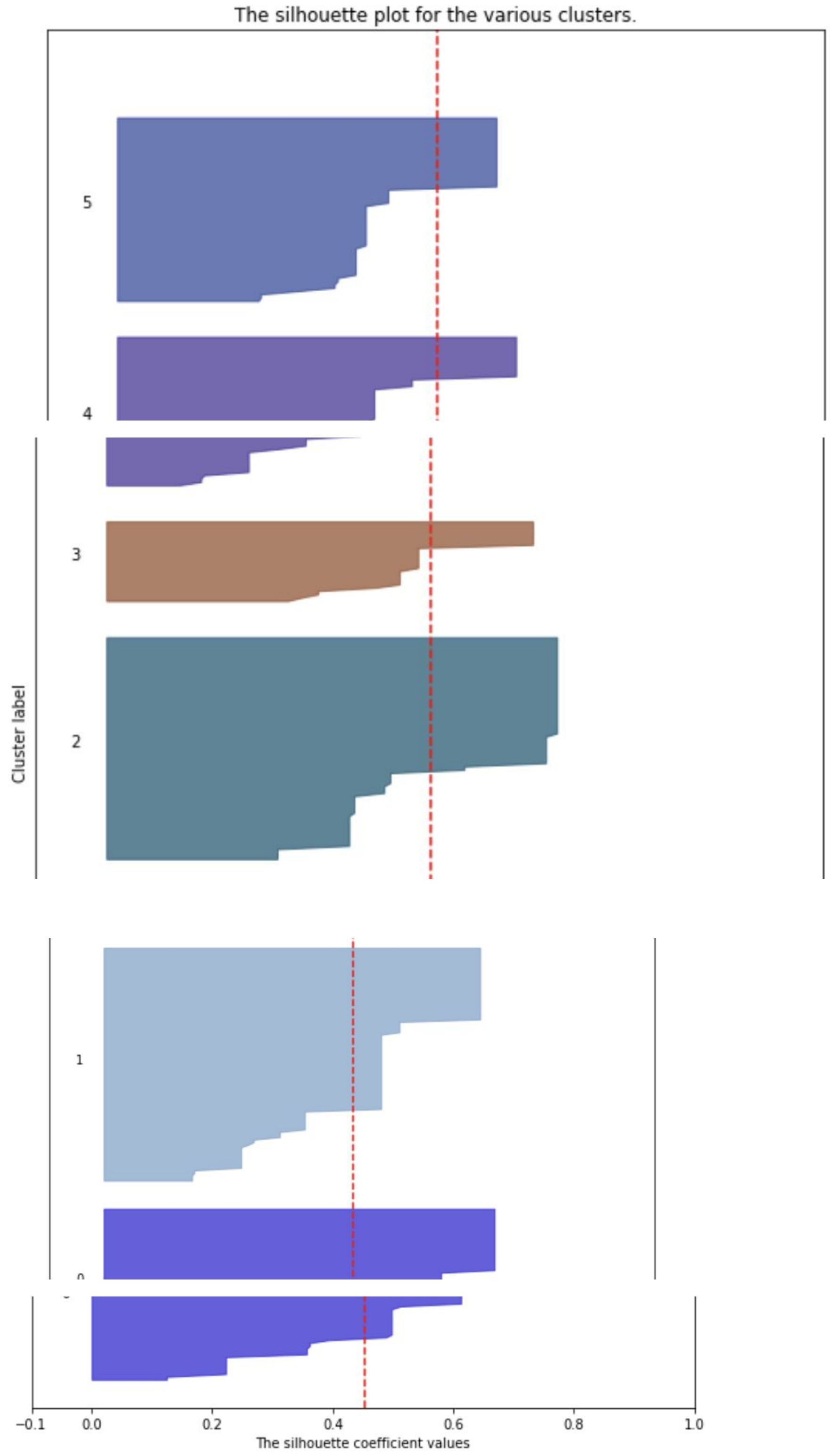


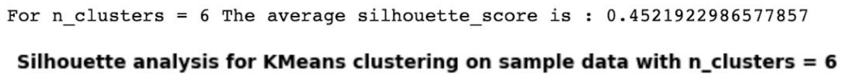
plot inertia scores by number of clusters

In [42]:

```
# first attempt at fitting K means to view change in Inertia
# class sklearn.cluster.KMeans(n_clusters=8, init='k-means++', n_init=10, max_iter=300, tol=0.0001, precompute_distance

# container to store inertia scores over iterations
distortions = []

# fit KMeans iteratively to begin to assess the appropriate number of clusters
for i in range(1, 50):
    km = KMeans(n_clusters=i)
    km.fit(X)
    distortions.append(km.inertia_)

# vizualize change in inertia
plt.plot(range(1, 50), distortions, marker='o')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.show()
```

Change the number range of the Cluster to average 6


The silhouette plot for the various clusters.

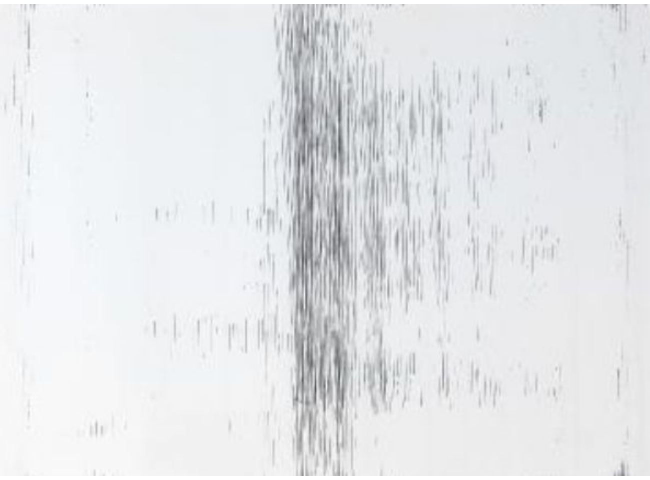


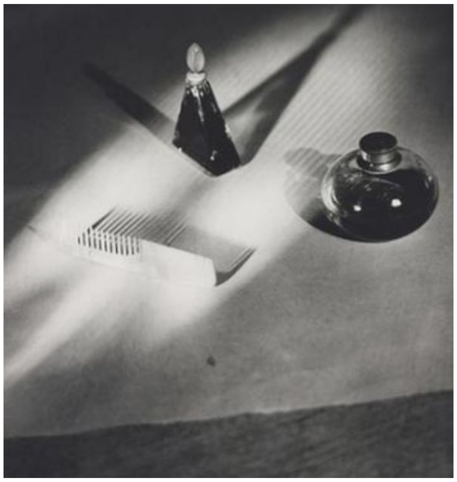
```
For n_clusters = 6 The average silhouette_score is : 0.4521922986577857
```

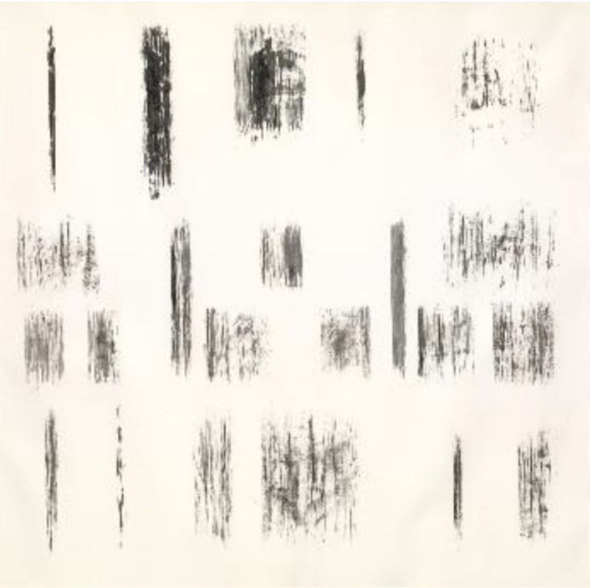
Silhouette analysis for KMeans clustering on sample data with n_clusters = 6

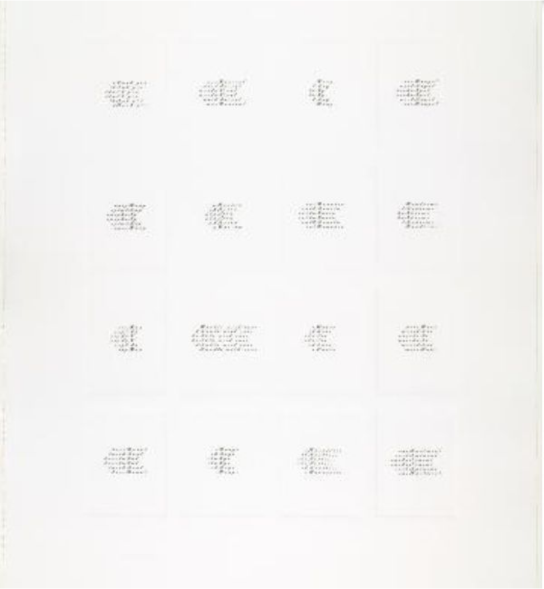
Some photos within a few clusters:

Cluster1:

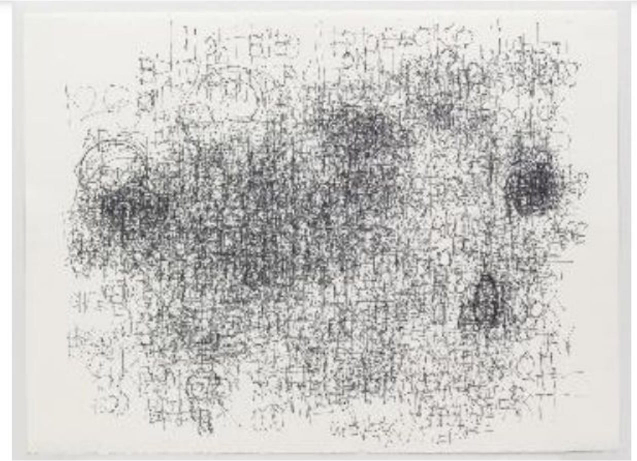


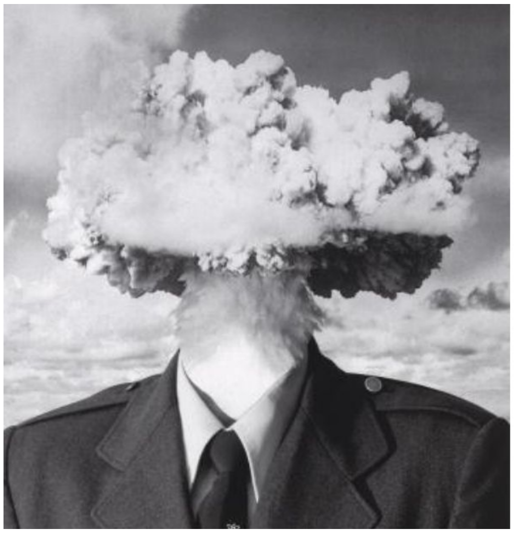






Cluster:





Cluster: