Import required packages

24-nov

```
- add dbscan to basic clustering ...
```

In [ ]:

TOC

- Modeling and Evaluation 1 :

```
Train and adjust parameters
```

  - Kmeans - LDA
  - Kmeans - all_in
  - DBScan
  - Spectral Clustering
  - end of file

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.simplefilter('ignore',DeprecationWarning)
import seaborn as sns
import time
import copy

from pylab import rcParams
#import hdbscan

from sklearn.model_selection import ShuffleSplit
from sklearn.preprocessing import StandardScaler

#from sklearn.datasets import make_blobs

from sklearn.ensemble import RandomForestClassifier
from sklearn.calibration import CalibratedClassifierCV
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import StratifiedKFold, cross_val_score

from sklearn import metrics
from sklearn import metrics as mt
from sklearn.metrics import log_loss
from sklearn.metrics import accuracy_score as acc
from sklearn.metrics import confusion_matrix as conf
from sklearn.metrics import f1_score, precision_score, recall_score, classi
fication_report
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_recall_fscore_support as score

from sklearn.cluster import KMeans

from tabulate import tabulate

from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"

from __future__ import print_function
```

Read in cleaned dataset from .csv file

In [2]:
```
data_dir = '../data/'
data_file = 'mashable_clean_dataset_for_lab_02_task_02.csv'

file_2_read = data_dir + data_file
df = pd.read_csv(file_2_read)

df_cluster = copy.deepcopy(df)

del df_cluster['data_channel']
```

In [3]:
```
for column in ['LDA_00', 'LDA_01', 'LDA_02', 'LDA_03', 'LDA_04']:
    new_col_name = 'ln_' + column
    print (new_col_name)
    df_cluster[new_col_name] = np.log(df_cluster[column]+1)
```

```
ln_LDA_00
ln_LDA_01
ln_LDA_02
ln_LDA_03
ln_LDA_04
```

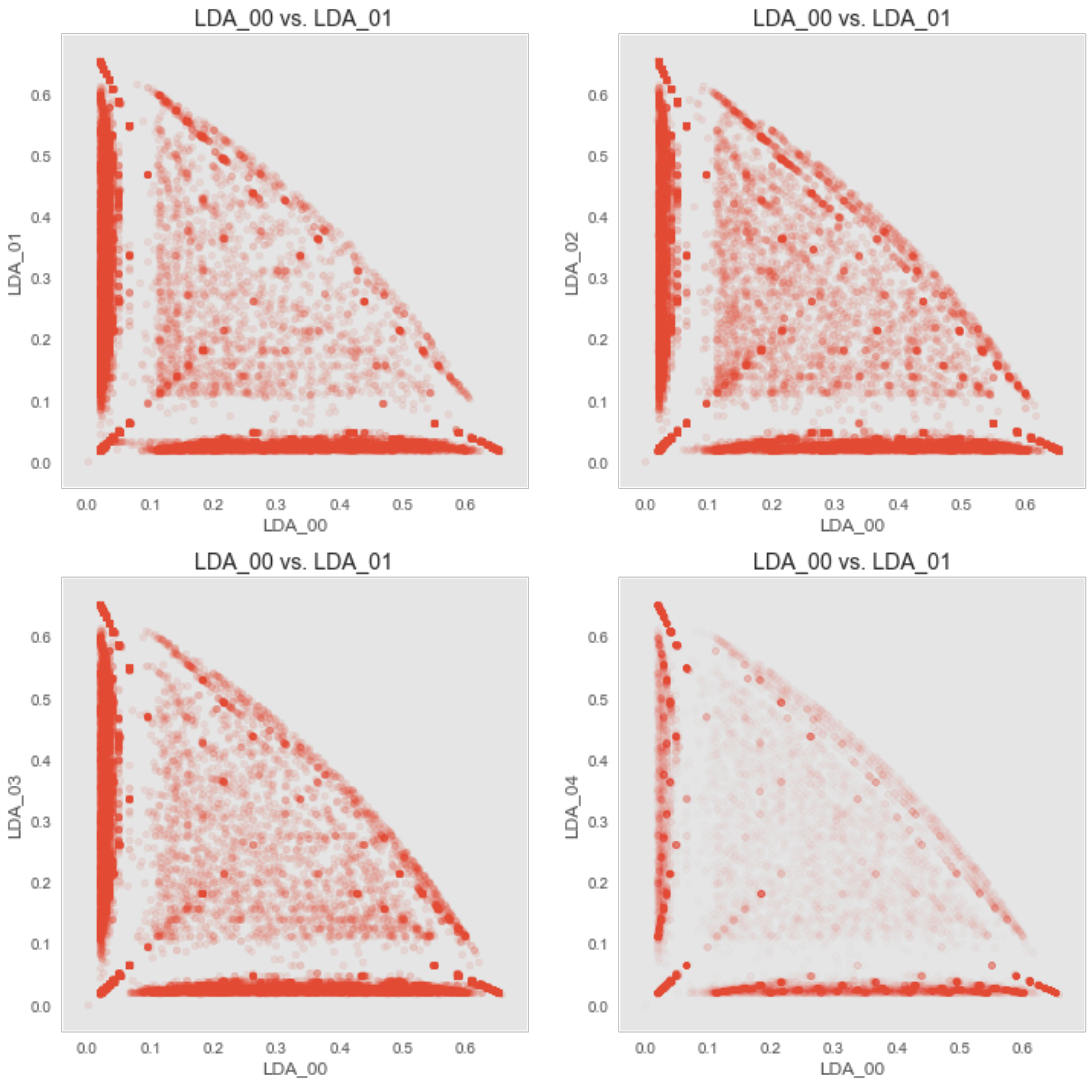
In [4]:
```
col_names = df_cluster.columns.values.tolist()
col_names
df_cluster.describe().T
```

```
Out[4]: ['n_tokens_title',
         'num_keywords',
         'kw_avg_max',
         'weekday_is_monday',
         'weekday_is_tuesday',
         'weekday_is_wednesday',
         'weekday_is_thursday',
         'weekday_is_friday',
         'is_weekend',
         'LDA_00',
         'LDA_01',
         'LDA_02',
         'LDA_03',
         'LDA_04',
         'global_subjectivity',
         'global_rate_positive_words',
         'rate_positive_words',
         'max_positive_polarity',
         'min_negative_polarity',
         'max_negative_polarity',
         'title_sentiment_polarity',
         'abs_title_subjectivity',
         'data_channel_n',
         'ln_n_tokens_content',
         'ln_num_hrefs',
         'ln_num_imgs',
         'ln_num_videos',
         'ln_kw_min_min',
         'ln_kw_avg_min',
         'ln_kw_min_max',
         'ln_kw_avg_avg',
         'ln_self_reference_avg_sharess',
         'ln_global_rate_negative_words',
         'ln_min_positive_polarity',
         'ln_abs_title_sentiment_polarity',
         'ln_LDA_00',
         'ln_LDA_01',
         'ln_LDA_02',
         'ln_LDA_03',
         'ln_LDA_04']
```

Out[4]:

	count	mean	std	min	25%	50%	75%
n_tokens_title	39644.0	10.398749	2.114037	2.0	9.000000	10.000000	12.000000
num_keywords	39644.0	7.223767	1.909130	1.0	6.000000	7.000000	9.000000
kw_avg_max	39644.0	1.913205	1.000000	0.0	1.271003	1.800325	2.442234
weekday_is_monday	39644.0	0.168020	0.373889	0.0	0.000000	0.000000	0.000000
weekday_is_tuesday	39644.0	0.186409	0.389441	0.0	0.000000	0.000000	0.000000
weekday_is_wednesday	39644.0	0.187544	0.390353	0.0	0.000000	0.000000	0.000000
weekday_is_thursday	39644.0	0.183306	0.386922	0.0	0.000000	0.000000	0.000000
weekday_is_friday	39644.0	0.143805	0.350896	0.0	0.000000	0.000000	0.000000
is_weekend	39644.0	0.130915	0.337312	0.0	0.000000	0.000000	0.000000
LDA_00	39644.0	0.184599	0.262975	0.0	0.025051	0.033387	0.240958
LDA_01	39644.0	0.141256	0.219707	0.0	0.025012	0.033345	0.150831
LDA_02	39644.0	0.216321	0.282145	0.0	0.028571	0.040004	0.334218
LDA_03	39644.0	0.223770	0.295191	0.0	0.028571	0.040001	0.375763
LDA_04	39644.0	0.234029	0.289183	0.0	0.028574	0.040727	0.399986
global_subjectivity	39644.0	0.443370	0.116685	0.0	0.396167	0.453457	0.508333
global_rate_positive_words	39644.0	0.039625	0.017429	0.0	0.028384	0.039023	0.050279
rate_positive_words	39644.0	0.682150	0.190206	0.0	0.600000	0.710526	0.800000
max_positive_polarity	39644.0	0.756728	0.247786	0.0	0.600000	0.800000	1.000000
min_negative_polarity	39644.0	0.478056	0.290290	0.0	0.300000	0.500000	0.700000
max_negative_polarity	39644.0	0.892500	0.095373	0.0	0.875000	0.900000	0.950000
title_sentiment_polarity	39644.0	1.071425	0.265450	0.0	1.000000	1.000000	1.150000
abs_title_subjectivity	39644.0	0.341843	0.188791	0.0	0.166667	0.500000	0.500000
data_channel_n	39644.0	4.184366	2.205607	1.0	2.000000	4.000000	6.000000
ln_n_tokens_content	39644.0	5.889971	1.255442	0.0	5.509388	6.016157	6.575076
ln_num_hrefs	39644.0	2.156564	0.809445	0.0	1.609438	2.197225	2.708050
ln_num_imgs	39644.0	1.116427	0.973755	0.0	0.693147	0.693147	1.609438
ln_num_videos	39644.0	0.400420	0.680486	0.0	0.000000	0.000000	0.693147
ln_kw_min_min	39644.0	1.174410	1.733030	0.0	0.000000	0.000000	1.791759
ln_kw_avg_min	39644.0	5.302209	1.132463	0.0	4.968076	5.470168	5.883322
ln_kw_min_max	39644.0	5.045209	4.521016	0.0	0.000000	7.244942	8.974745
ln_kw_avg_avg	39644.0	7.976327	0.489467	0.0	7.776304	7.962442	8.189031
ln_self_reference_avg_sharess	39644.0	6.667697	3.280186	0.0	6.889782	7.696667	8.556606
ln_global_rate_negative_words	39644.0	0.016419	0.010571	0.0	0.009569	0.015221	0.021506
ln_min_positive_polarity	39644.0	0.089255	0.060260	0.0	0.048790	0.095310	0.095310
ln_abs_title_sentiment_polarity	39644.0	0.128709	0.173844	0.0	0.000000	0.000000	0.223144

In [5]:
```
from matplotlib import pyplot as plt
plt.style.use("ggplot")

%matplotlib inline


X1 = df_cluster[['ln_LDA_00','ln_LDA_01', 'ln_LDA_02', 'ln_LDA_03', 'ln_LDA
_04']].values


plt.figure(figsize = (12,12))
plt.subplot(221)

plt.scatter(X1[:, 1], X1[:, 0],
            s = 20,
           alpha = 0.10)
plt.xlabel('LDA_00'), plt.ylabel('LDA_01')
plt.grid()
plt.title('LDA_00 vs. LDA_01')


plt.subplot(222)
plt.scatter(X1[:, 2], X1[:, 0],
            s = 20,
           alpha = 0.10)
plt.xlabel('LDA_00'), plt.ylabel('LDA_02')
plt.grid()
plt.title('LDA_00 vs. LDA_01')


plt.subplot(223)
plt.scatter(X1[:, 3], X1[:, 0],
            s = 20,
            alpha = 0.10)
plt.xlabel('LDA_00'), plt.ylabel('LDA_03')
plt.grid()
plt.title('LDA_00 vs. LDA_01')


plt.subplot(224)
plt.scatter(X1[:, 4], X1[:, 0],
            s = 20,
           alpha = 0.01)
plt.xlabel('LDA_00'), plt.ylabel('LDA_04')
plt.grid()
plt.title('LDA_00 vs. LDA_01')

plt.show();
```



In [6]:
```
# set required variables for model comparison

comparison_tbl = pd.DataFrame(columns = [
    'model_name',
    'n_clusters',
    'inertia',
    'silhouette',
    'process_time'])

i_index = []
i_index = 0

# preparation for cross validation and model comparison, each classifier is
appended once model is fit

models = []
```

Table of Contents

KMeans - LDA

K-Means - LDA scores

In [7]:
```
for n_lda in range(2, 12):

    tic = time.clock()

    print ("n_lda = ", n_lda)

    X1 = df_cluster[['ln_LDA_00','ln_LDA_01', 'ln_LDA_02', 'ln_LDA_03', 'ln_LDA_04']]

    cls_lda = KMeans(n_clusters = n_lda,
                     init = 'k-means++',
                     random_state = 1);

    cls_lda.fit(X1)

    kmeans_labels = cls_lda.labels_ # the labels from kmeans clustering
    kmeans_centers = cls_lda.cluster_centers_

    kmeans_inertia = cls_lda.inertia_
    print ("inertia = ", kmeans_inertia)

    kmeans_silhouette = metrics.silhouette_score(X1,
                                                 kmeans_labels,
                                                 metric = 'euclidean',
                                                 sample_size = 10000)
    print ("silhouette = ", kmeans_silhouette)

    toc =  time.clock()
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ... - save statistics for model comparison
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    exe_time = '{0:.4f}'.format(toc-tic)

    raw_data = {
    'model_name' : 'KMeans - LDA features',
    'n_clusters' : n_lda,
    'inertia': kmeans_inertia,
    'silhouette': kmeans_silhouette,
    'process_time' : exe_time
    }

    df_tbl = pd.DataFrame(raw_data,
    columns = ['model_name', 'n_clusters', 'inertia', 'silhouette', 'process_time'],
    index = [i_index + 1])

    comparison_tbl = comparison_tbl.append(df_tbl)

# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ... - make some plots of clusters
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    plt.figure(figsize=(12, 12));
    plt.subplot(221);
    X1 = X1.values;
    plt.scatter(X1[:, 0], X1[:, 1],
```

```
n_lda =  2
```

Out[7]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=2, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=1, tol=0.0001, verbose=0)
```

```
inertia =  6016.62121709
silhouette =  0.266625232023
```

Out[7]:
```
<matplotlib.figure.Figure at 0x7f8e35e8a390>
```

Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e35972978>
```

Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e354a79e8>
```

Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e354ee4e0>
```

Out[7]:
```
<matplotlib.text.Text at 0x7f8e359449b0>
```

Out[7]:
```
(<matplotlib.text.Text at 0x7f8e3594a5c0>,
 <matplotlib.text.Text at 0x7f8e354d7160>)
```

Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e359443c8>
```

Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e3541d748>
```

Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e354270f0>
```

Out[7]:
```
(<matplotlib.text.Text at 0x7f8e354aec50>,
 <matplotlib.text.Text at 0x7f8e354bbe80>)
```

Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e354570f0>
```

Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e35af9240>
```

Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e35ae8668>
```

Out[7]:
```
(<matplotlib.text.Text at 0x7f8e354ca7f0>,
 <matplotlib.text.Text at 0x7f8e35a9da58>)
```

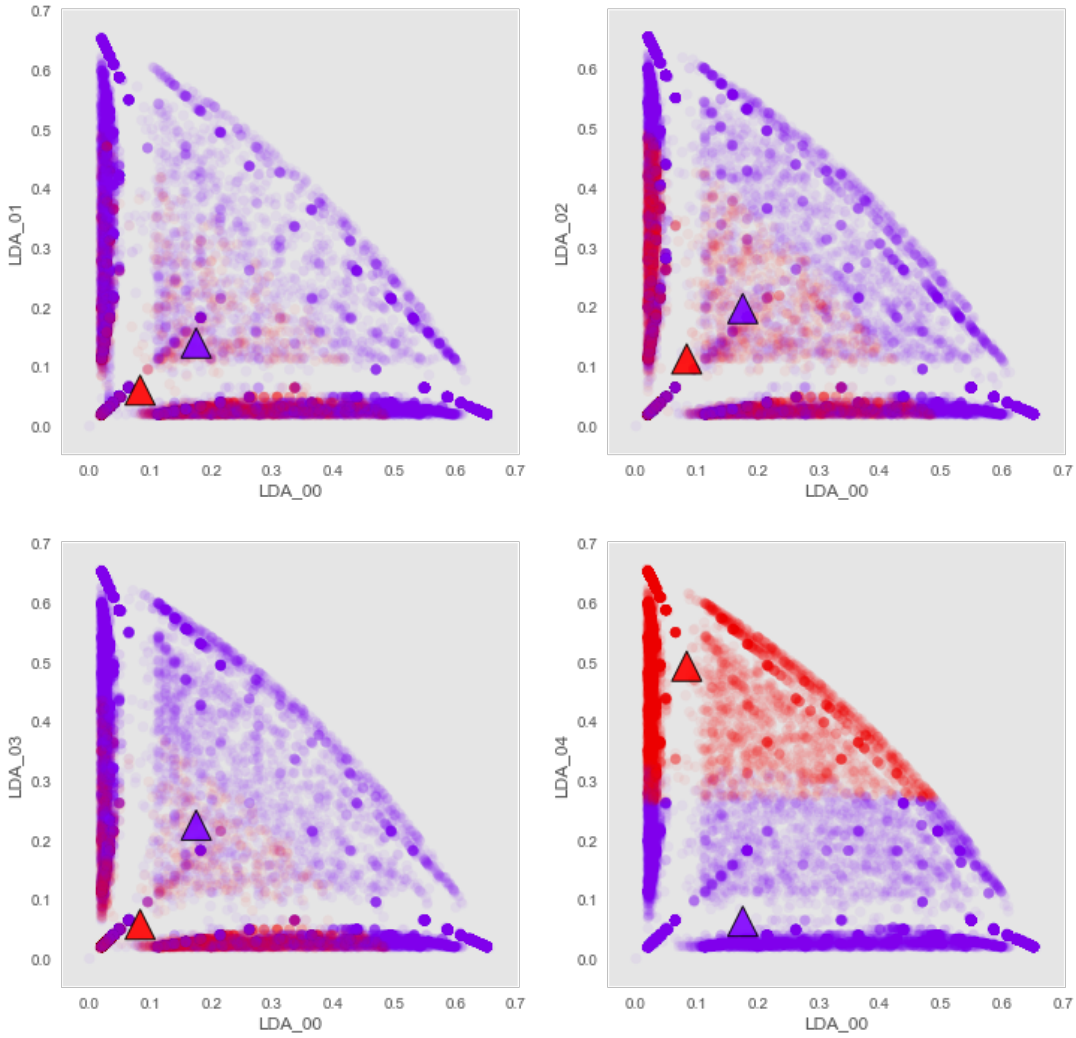
Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e35af9710>
```

Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e35b3e128>
```

Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e359aedd8>
```

Out[7]:
```
(<matplotlib.text.Text at 0x7f8e35ae87f0>,
 <matplotlib.text.Text at 0x7f8e354276d8>)
```

6016.62121709




```
n_lda =  3
```

Out[7]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=3, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=1, tol=0.0001, verbose=0)
```

```
inertia =  4120.02063847
silhouette =  0.384511692224
```

Out[7]:
```
<matplotlib.figure.Figure at 0x7f8e35941390>
```

Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e35a762b0>
```

Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e35b84cc0>
```

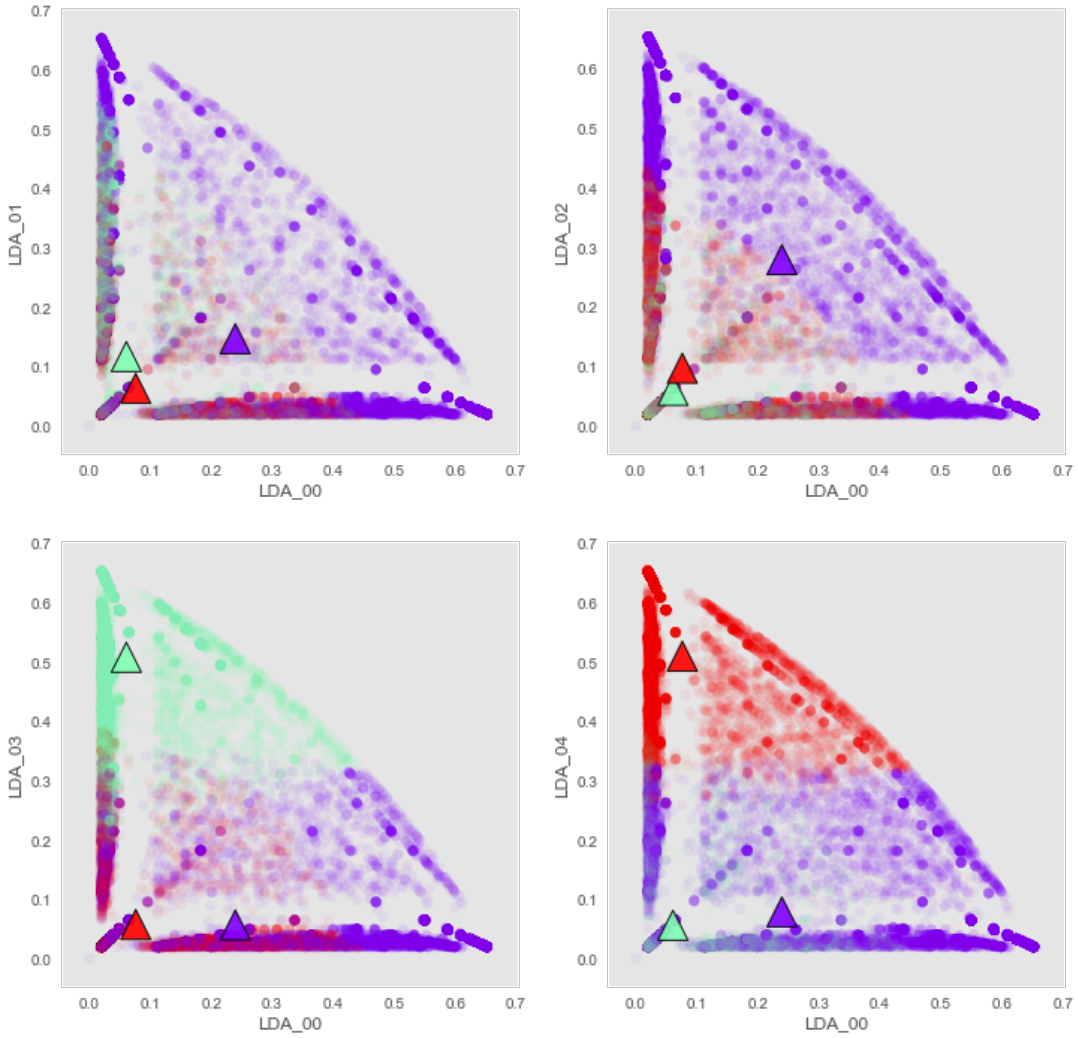
Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e35b94550>
```

Out[7]:
```
<matplotlib.text.Text at 0x7f8e35b84cf8>
```

Out[7]:
```
(<matplotlib.text.Text at 0x7f8e35b6c978>,
 <matplotlib.text.Text at 0x7f8e35bb9e80>)
```

```
Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e35b94d30>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e35b48f60>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e35b48f98>

Out[7]: (<matplotlib.text.Text at 0x7f8e359a1438>,
         <matplotlib.text.Text at 0x7f8e35b08f28>)

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e3427b080>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e34240278>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e34240c18>

Out[7]: (<matplotlib.text.Text at 0x7f8e3425e358>,
         <matplotlib.text.Text at 0x7f8e35afc240>)

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e342475c0>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e341a7dd8>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e341ad668>

Out[7]: (<matplotlib.text.Text at 0x7f8e3424f5f8>,
         <matplotlib.text.Text at 0x7f8e341e7978>)
```

4120.02063847




```
n_lda =  4
```

Out[7]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=4, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=1, tol=0.0001, verbose=0)
```

```
inertia =  2523.83403252
silhouette =  0.501645423645
```

Out[7]:
```
<matplotlib.figure.Figure at 0x7f8e35941780>
```

Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e35937320>
```

Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e3405d4a8>
```

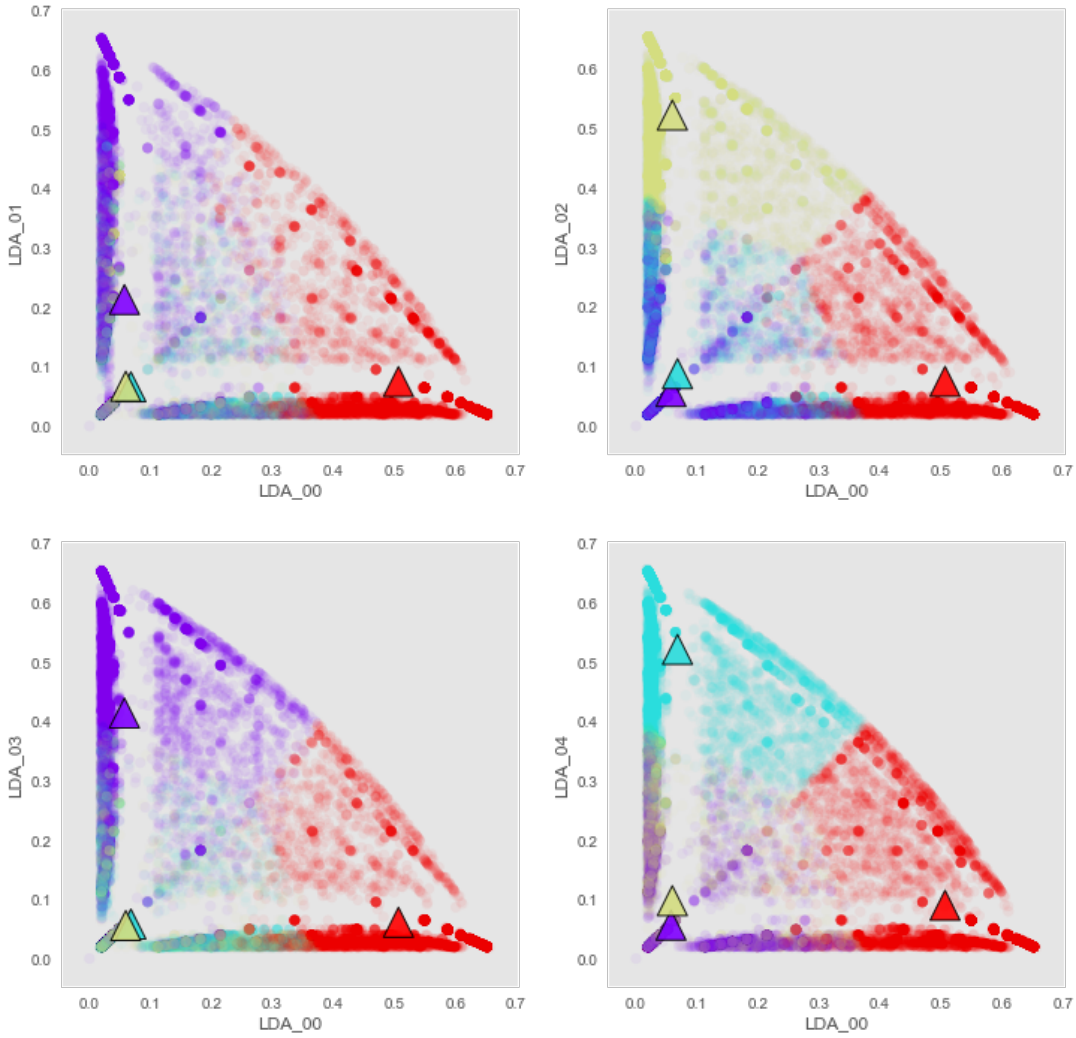
Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e340a9da0>
```

Out[7]:
```
<matplotlib.text.Text at 0x7f8e340665f8>
```

Out[7]:
```
(<matplotlib.text.Text at 0x7f8e34129c18>,
 <matplotlib.text.Text at 0x7f8e34184c50>)
```

```
Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e34066a90>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e340528d0>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e34059160>

Out[7]: (<matplotlib.text.Text at 0x7f8e34109358>,
         <matplotlib.text.Text at 0x7f8e3406c278>)

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e34052e48>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e3414d8d0>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e34155f60>

Out[7]: (<matplotlib.text.Text at 0x7f8e2c706b00>,
         <matplotlib.text.Text at 0x7f8e3408d7b8>)

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e35449710>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e34190048>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e34109eb8>

Out[7]: (<matplotlib.text.Text at 0x7f8e3421d048>,
         <matplotlib.text.Text at 0x7f8e34068278>)
```

2523.83403252




```
n_lda =  5
```

Out[7]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=5, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=1, tol=0.0001, verbose=0)
```

```
inertia =  1418.81957692
silhouette =  0.564549085098
```

Out[7]:
```
<matplotlib.figure.Figure at 0x7f8e341fc5f8>
```

Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e35429588>
```

Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e35b26f98>
```

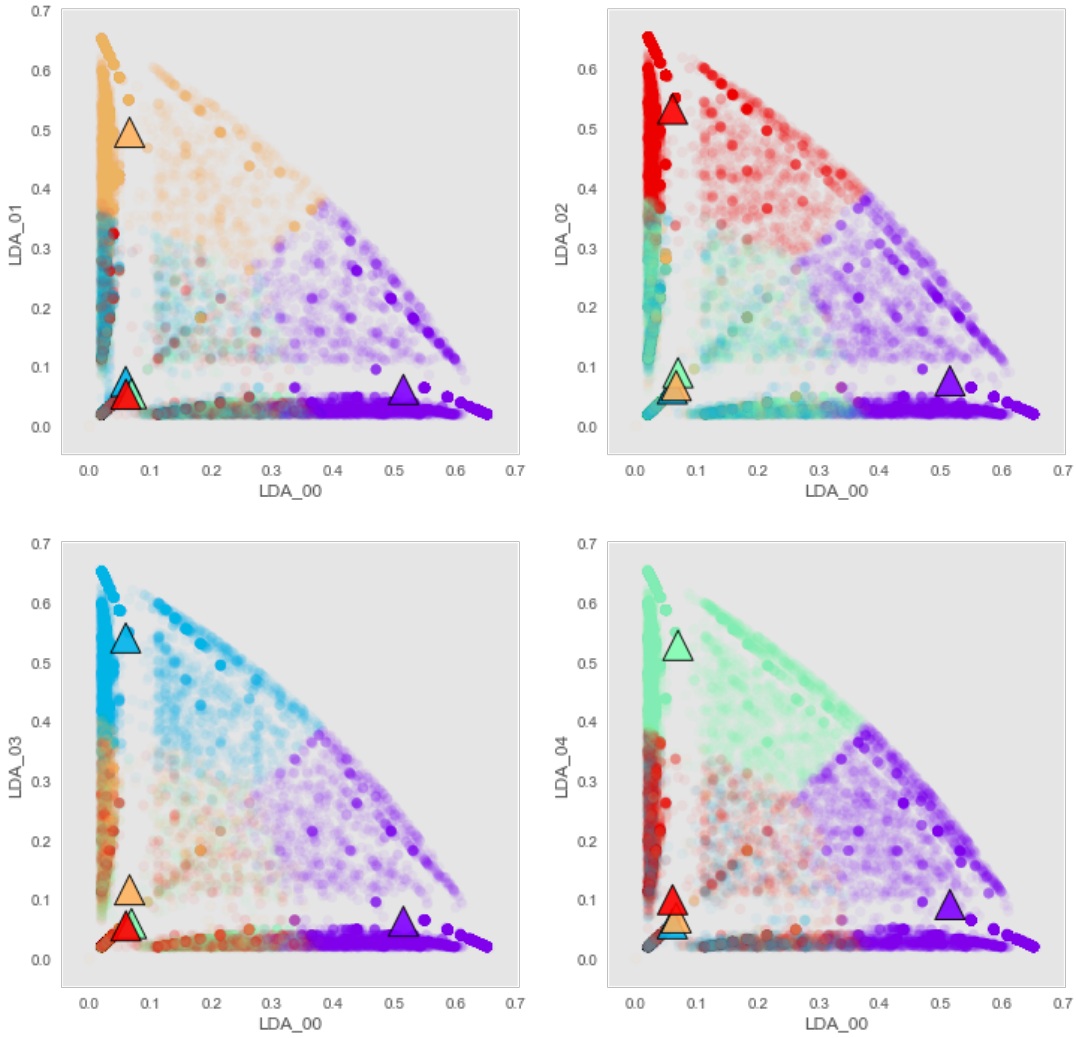
Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e359ae3c8>
```

Out[7]:
```
<matplotlib.text.Text at 0x7f8e35b26358>
```

Out[7]:
```
(<matplotlib.text.Text at 0x7f8e341f64a8>,
 <matplotlib.text.Text at 0x7f8e35435198>)
```

```
Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e35a9d240>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e35a56e10>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e359f4cc0>

Out[7]: (<matplotlib.text.Text at 0x7f8e359ae438>,
         <matplotlib.text.Text at 0x7f8e35ac8eb8>)

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e35a51a58>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e35bd4c18>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e35a34198>

Out[7]: (<matplotlib.text.Text at 0x7f8e35a1c4e0>,
         <matplotlib.text.Text at 0x7f8e35a48c88>)

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e35a2aeb8>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e359d0400>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e359d0c50>

Out[7]: (<matplotlib.text.Text at 0x7f8e35a691d0>,
         <matplotlib.text.Text at 0x7f8e359729e8>)
```

1418.81957692




```
n_lda =  6
```

Out[7]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=6, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=1, tol=0.0001, verbose=0)
```

```
inertia =  1257.64273723
silhouette =  0.521216884499
```

Out[7]:
```
<matplotlib.figure.Figure at 0x7f8e3422fc88>
```

Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e35a12518>
```

Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e3544da90>
```

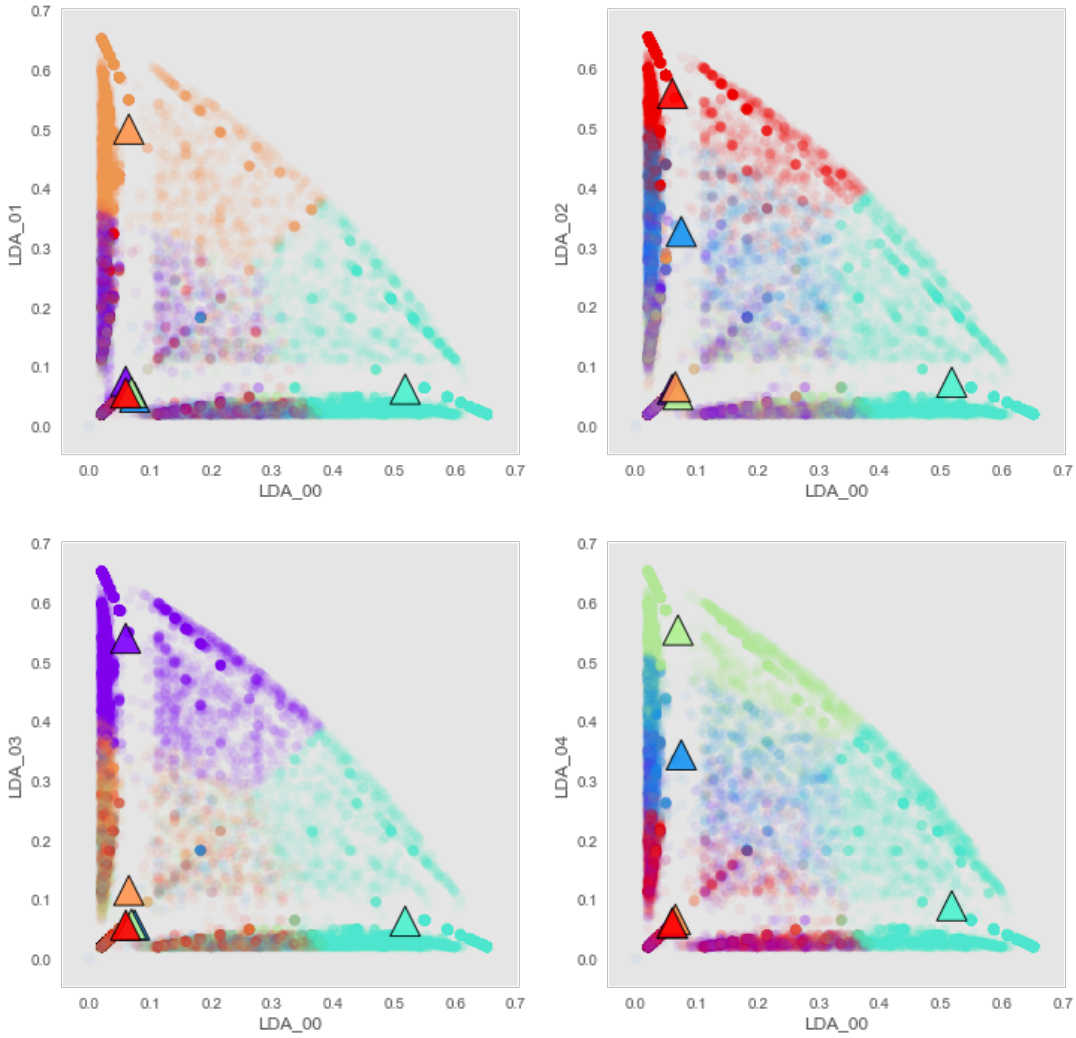
Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e35437320>
```

Out[7]:
```
<matplotlib.text.Text at 0x7f8e3544dac8>
```

Out[7]:
```
(<matplotlib.text.Text at 0x7f8e35a8ee48>,
 <matplotlib.text.Text at 0x7f8e2c741198>)
```

```
Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e35437fd0>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e354aceb8>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e35465748>

Out[7]: (<matplotlib.text.Text at 0x7f8e35446128>,
         <matplotlib.text.Text at 0x7f8e354221d0>)

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e354acd30>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e2c6f8160>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e2c6f8b38>

Out[7]: (<matplotlib.text.Text at 0x7f8e3545c128>,
         <matplotlib.text.Text at 0x7f8e35514be0>)

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c6fd358>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e35af03c8>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e3599c5f8>

Out[7]: (<matplotlib.text.Text at 0x7f8e2c6a13c8>,
         <matplotlib.text.Text at 0x7f8e2c68d5f8>)
```

1257.64273723




```
n_lda =  7
```

Out[7]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=7, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=1, tol=0.0001, verbose=0)
```

```
inertia =  1135.46696089
silhouette =  0.506603738229
```

Out[7]:
```
<matplotlib.figure.Figure at 0x7f8e359de048>
```

Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c6e59e8>
```

Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e35a12da0>
```

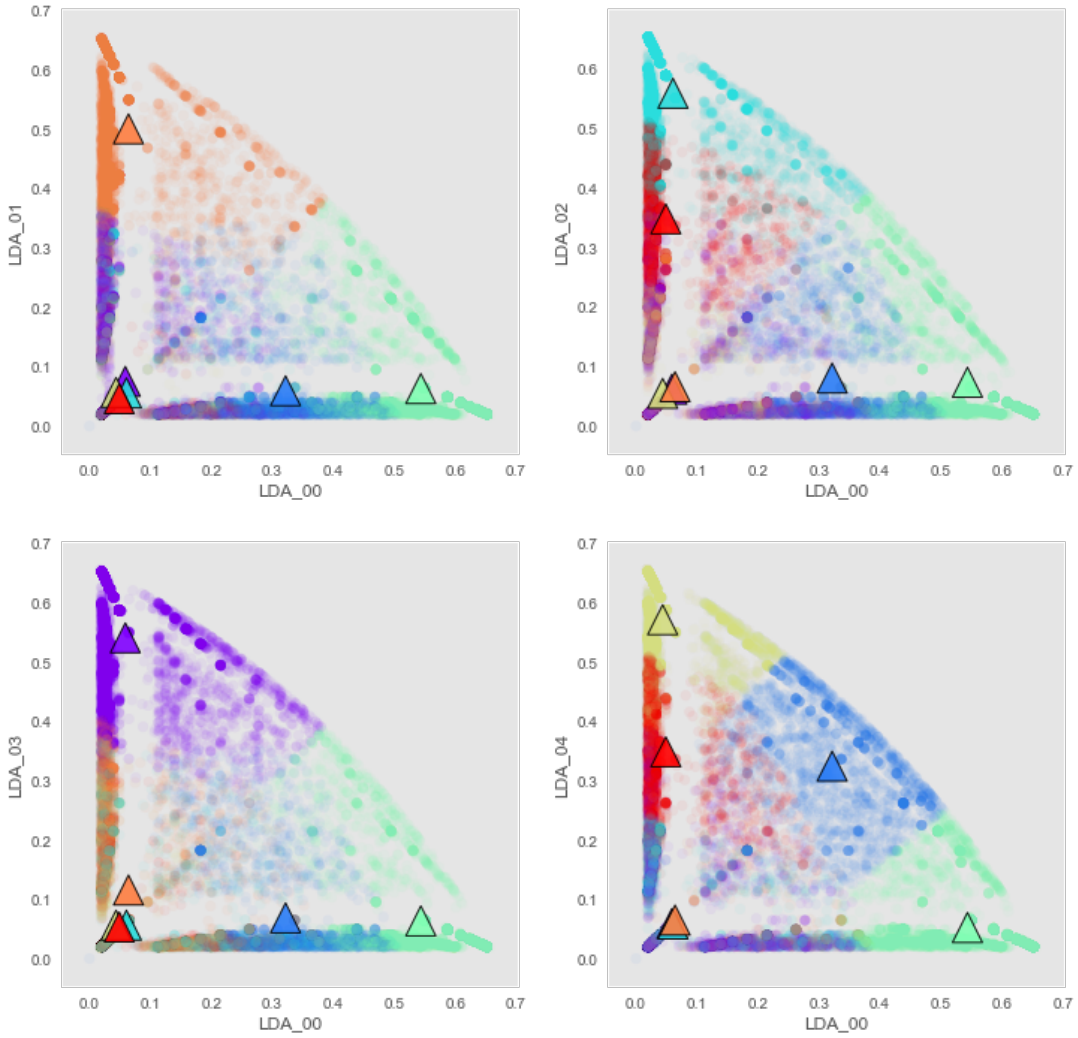
Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e35af7d30>
```

Out[7]:
```
<matplotlib.text.Text at 0x7f8e35a12d68>
```

Out[7]:
```
(<matplotlib.text.Text at 0x7f8e35a8ea20>,
 <matplotlib.text.Text at 0x7f8e2c741be0>)
```

```
Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e35ab6e10>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e359ff320>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e35b26898>

Out[7]: (<matplotlib.text.Text at 0x7f8e35a5ba20>,
         <matplotlib.text.Text at 0x7f8e35addbe0>)

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e35a24400>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e35c08e48>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e35b9fc88>

Out[7]: (<matplotlib.text.Text at 0x7f8e340b9198>,
         <matplotlib.text.Text at 0x7f8e35a1c128>)

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e35c08b38>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e35b4af98>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e341cdd30>

Out[7]: (<matplotlib.text.Text at 0x7f8e35b9f940>,
         <matplotlib.text.Text at 0x7f8e3421b908>)
```

1135.46696089




```
n_lda =  8
```

Out[7]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=8, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=1, tol=0.0001, verbose=0)
```

```
inertia =  1023.64621652
silhouette =  0.470087897868
```

Out[7]:
```
<matplotlib.figure.Figure at 0x7f8e359de668>
```

Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e35b9fb70>
```

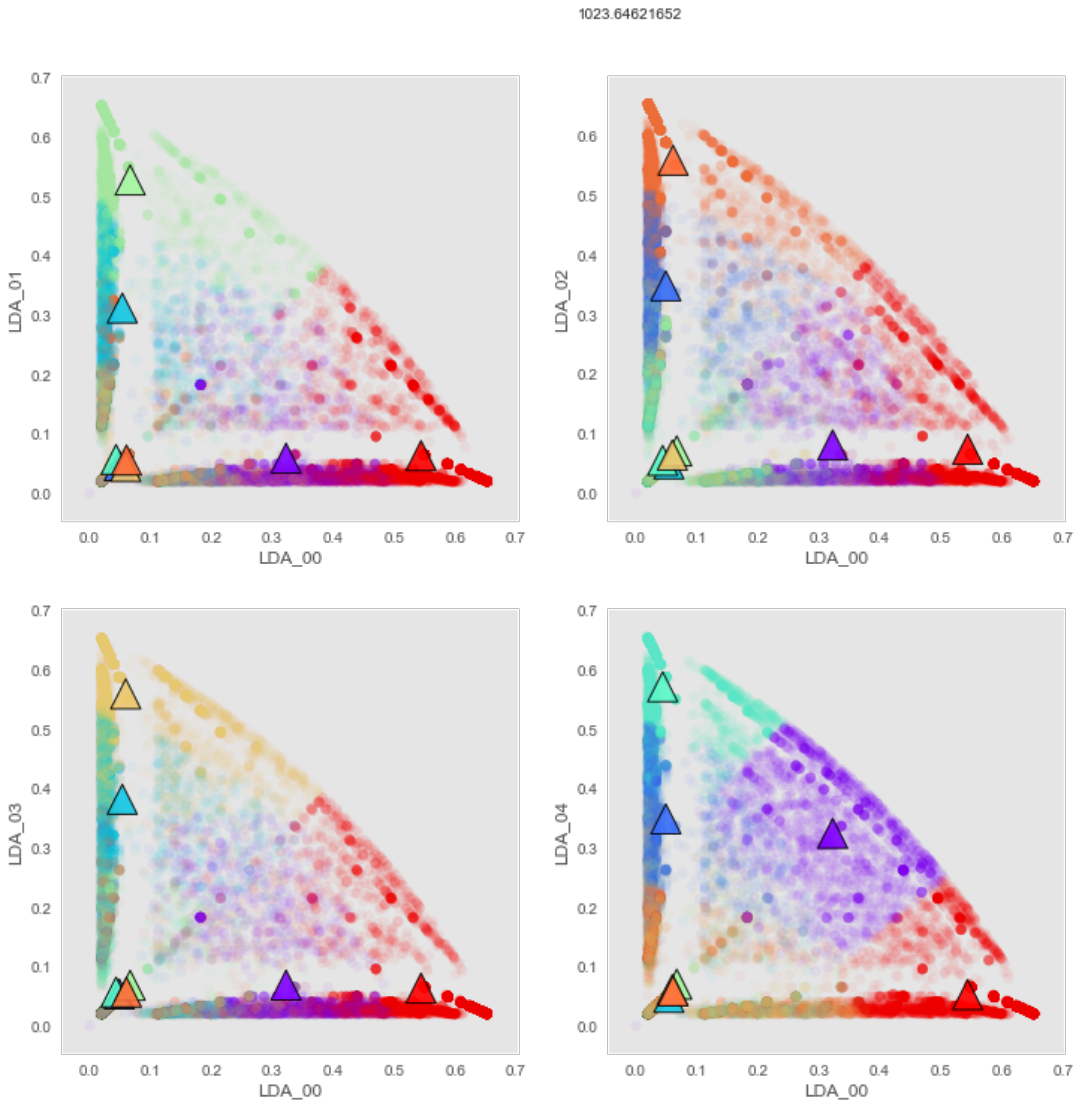
Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e3413f940>
```

Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e3414a240>
```

Out[7]:
```
<matplotlib.text.Text at 0x7f8e3413feb8>
```

Out[7]:
```
(<matplotlib.text.Text at 0x7f8e35a1f908>,
 <matplotlib.text.Text at 0x7f8e35a16fd0>)
```

```
Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c720ac8>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e340e7d68>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e340f35f8>

Out[7]: (<matplotlib.text.Text at 0x7f8e3414a400>,
         <matplotlib.text.Text at 0x7f8e341d2240>)

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e340e7da0>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e34091f28>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e34091fd0>

Out[7]: (<matplotlib.text.Text at 0x7f8e340f5f98>,
         <matplotlib.text.Text at 0x7f8e341d1860>)

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e340800f0>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e2c654908>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e2c6593c8>

Out[7]: (<matplotlib.text.Text at 0x7f8e3405e278>,
         <matplotlib.text.Text at 0x7f8e340505f8>)
```




```
n_lda =  9
```

Out[7]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=9, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=1, tol=0.0001, verbose=0)
```

```
inertia =  937.873675896
silhouette =  0.474417563079
```

Out[7]:
```
<matplotlib.figure.Figure at 0x7f8e3401e6d8>
```

Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e35a9d550>
```

Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e359aec50>
```

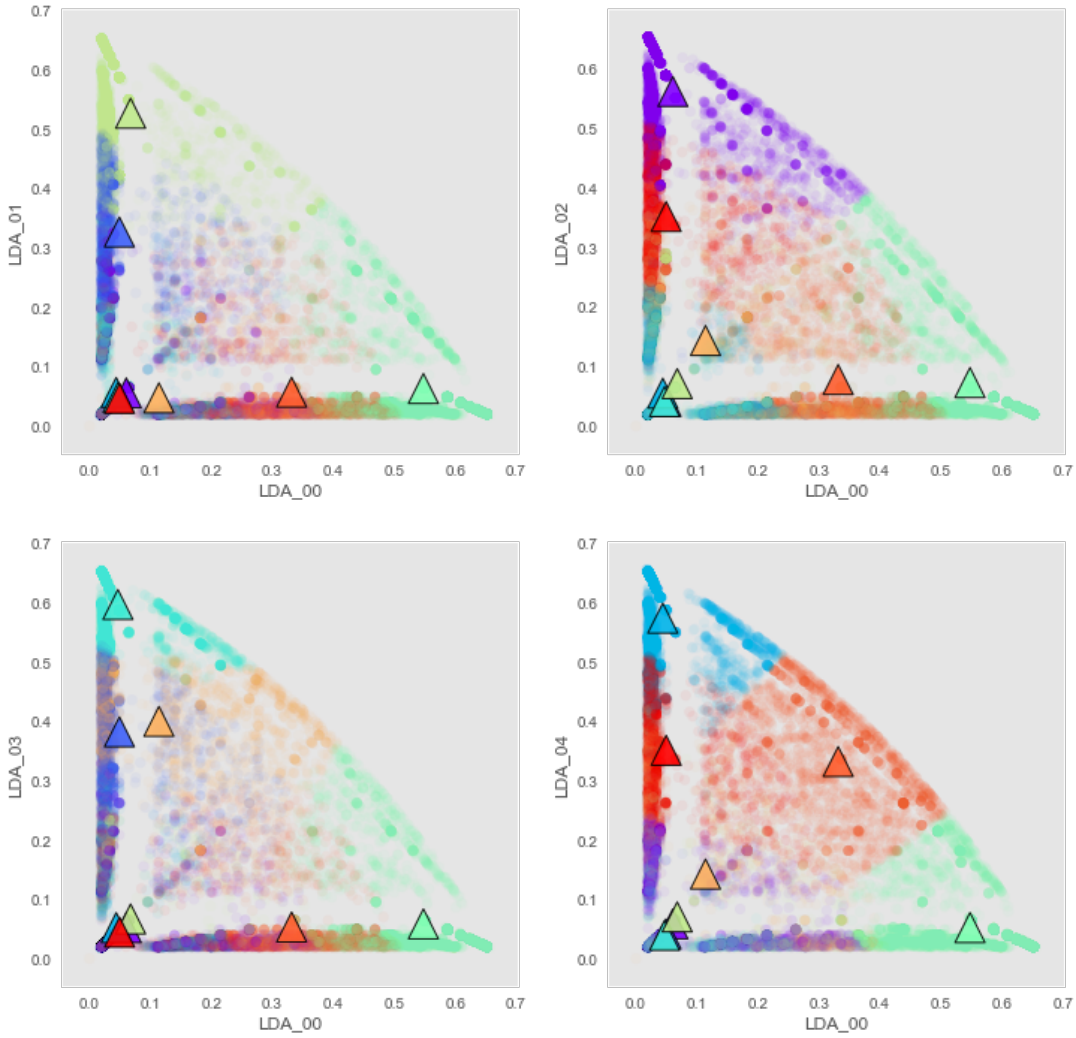
Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e35a24da0>
```

Out[7]:
```
<matplotlib.text.Text at 0x7f8e2c632c18>
```

Out[7]:
```
(<matplotlib.text.Text at 0x7f8e35bba208>,
 <matplotlib.text.Text at 0x7f8e340617f0>)
```

```
Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e341af390>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e3543fb70>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e6c89c2b0>

Out[7]: (<matplotlib.text.Text at 0x7f8e35b26400>,
         <matplotlib.text.Text at 0x7f8e359ffdd8>)

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e35adea20>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e35b240f0>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e2c6d2b00>

Out[7]: (<matplotlib.text.Text at 0x7f8e3543ff28>,
         <matplotlib.text.Text at 0x7f8e34136828>)

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e35b24588>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e35437550>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e3597ea90>

Out[7]: (<matplotlib.text.Text at 0x7f8e35add860>,
         <matplotlib.text.Text at 0x7f8e35a7c390>)
```

937.873675896




```
n_lda =  10
```

Out[7]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=10, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=1, tol=0.0001, verbose=0)
```

```
inertia =  859.164016267
silhouette =  0.47973633816
```

Out[7]:
```
<matplotlib.figure.Figure at 0x7f8e3401e550>
```

Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e35aa3908>
```

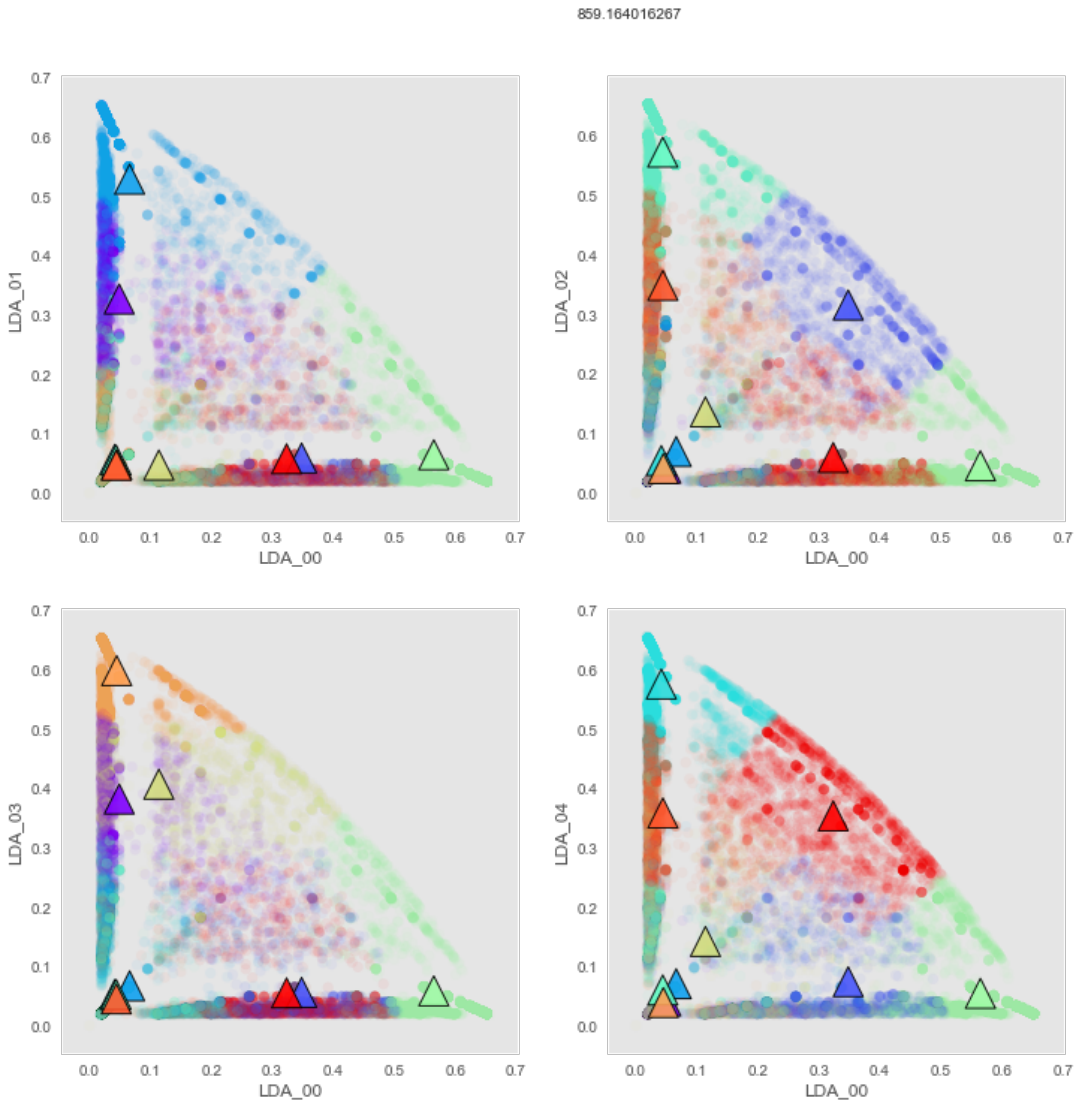
Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e35ab7d68>
```

Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e35a995f8>
```

Out[7]:
```
<matplotlib.text.Text at 0x7f8e35ab7da0>
```

Out[7]:
```
(<matplotlib.text.Text at 0x7f8e35984550>,
 <matplotlib.text.Text at 0x7f8e341090f0>)
```

```
Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e35aa6390>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e35a93198>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e35a93b70>

Out[7]: (<matplotlib.text.Text at 0x7f8e359d4be0>,
         <matplotlib.text.Text at 0x7f8e34109240>)

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e35a86320>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e354b8470>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e354b8cc0>

Out[7]: (<matplotlib.text.Text at 0x7f8e35a8f400>,
         <matplotlib.text.Text at 0x7f8e35ae8ba8>)

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e354d2630>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e35458e80>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e2c6bb710>

Out[7]: (<matplotlib.text.Text at 0x7f8e355046a0>,
         <matplotlib.text.Text at 0x7f8e355108d0>)
```




```
n_lda =  11
```

Out[7]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=11, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=1, tol=0.0001, verbose=0)
```

```
inertia =  794.23042098
silhouette =  0.493377090487
```

Out[7]:
```
<matplotlib.figure.Figure at 0x7f8e2c5d3940>
```

Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c5fa9b0>
```

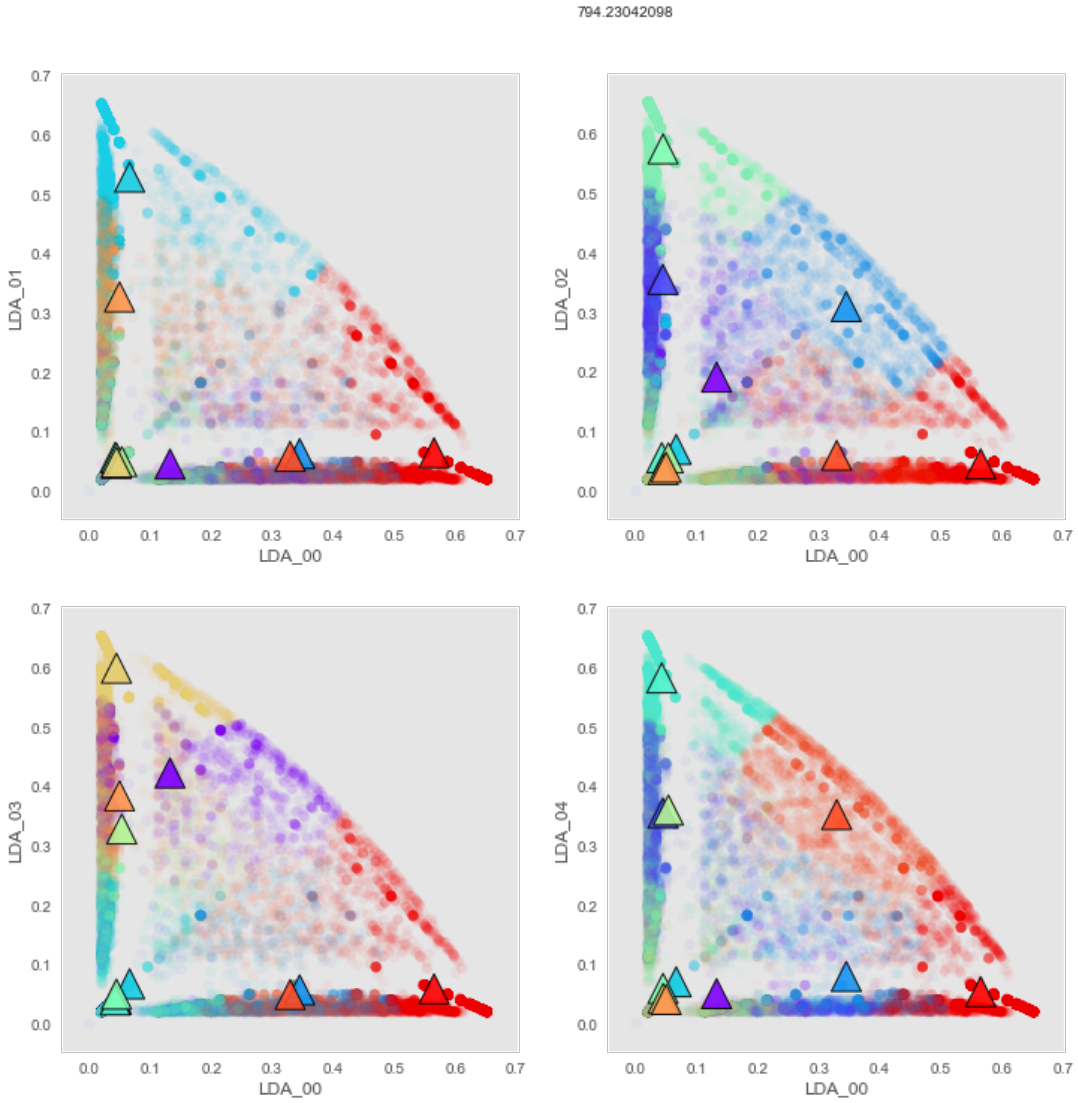
Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e35b4a8d0>
```

Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f8e35bc21d0>
```

Out[7]:
```
<matplotlib.text.Text at 0x7f8e34288f60>
```

Out[7]:
```
(<matplotlib.text.Text at 0x7f8e3545c5f8>,
 <matplotlib.text.Text at 0x7f8e3592a470>)
```

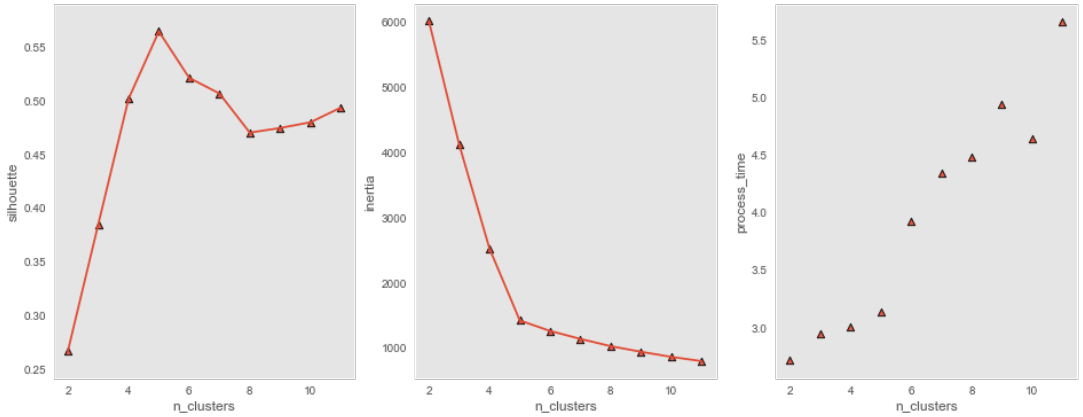
```
Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e34288518>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e35ac8b70>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e35c084a8>

Out[7]: (<matplotlib.text.Text at 0x7f8e359d4a58>,
         <matplotlib.text.Text at 0x7f8e359cecc0>)

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e35af02e8>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e35af9a90>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e35b70588>

Out[7]: (<matplotlib.text.Text at 0x7f8e35a1c828>,
         <matplotlib.text.Text at 0x7f8e35ae8400>)

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e36d0e518>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e34281128>

Out[7]: <matplotlib.collections.PathCollection at 0x7f8e35b4e7b8>

Out[7]: (<matplotlib.text.Text at 0x7f8e36cf5b70>,
         <matplotlib.text.Text at 0x7f8e35b8ce48>)
```



Out[7]:

	model_name	n_clusters	inertia	silhouette	process_time
1	KMeans - LDA features	2	6016.621217	0.266625	2.7176
1	KMeans - LDA features	3	4120.020638	0.384512	2.9431
1	KMeans - LDA features	4	2523.834033	0.501645	3.0021
1	KMeans - LDA features	5	1418.819577	0.564549	3.1371
1	KMeans - LDA features	6	1257.642737	0.521217	3.9218
1	KMeans - LDA features	7	1135.466961	0.506604	4.3356
1	KMeans - LDA features	8	1023.646217	0.470088	4.4810
1	KMeans - LDA features	9	937.873676	0.474418	4.9386
1	KMeans - LDA features	10	859.164016	0.479736	4.6344
1	KMeans - LDA features	11	794.230421	0.493377	5.6503

In [8]:
```
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ... - plot metrics across models for comparison
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

plt.figure(figsize=(16, 6));

# ... silhouette values

plt.subplot(131);
plt.scatter(comparison_tbl['n_clusters'],
            comparison_tbl['silhouette'],
            s = 40,
            linewidths = 1.0,
            marker = '^',
            edgecolors = 'black',
            alpha = 0.90);

plt.plot(comparison_tbl['n_clusters'],
         comparison_tbl['silhouette'])

plt.xlabel('n_clusters'), plt.ylabel('silhouette');
plt.grid();

# ... inertia values

plt.subplot(132);
plt.scatter(comparison_tbl['n_clusters'],
            comparison_tbl['inertia'],
            s = 40,
            linewidths = 1.0,
            marker = '^',
            edgecolors = 'black',
            alpha = 0.90);

plt.plot(comparison_tbl['n_clusters'],
         comparison_tbl['inertia'])

plt.xlabel('n_clusters'), plt.ylabel('inertia');
plt.grid();

# ... process time

plt.subplot(133);
plt.scatter(comparison_tbl['n_clusters'],
            comparison_tbl['process_time'],
            s = 40,
            linewidths = 1.0,
            marker = '^',
            edgecolors = 'black',
            alpha = 0.90);

#plt.plot(comparison_tbl['n_clusters'],
#         comparison_tbl['process_time'])

plt.xlabel('n_clusters'), plt.ylabel('process_time');
```


silhouette
0.55
0.50
0.45
0.40
0.35
0.30
0.25
2
4
6
8
10
n_clusters
inertia
6000
5000
4000
3000
2000
1000
2
4
6
8
10
n_clusters
process_time
5.5
5.0
4.5
4.0
3.5
3.0
2
4
6
8
10
n_clusters

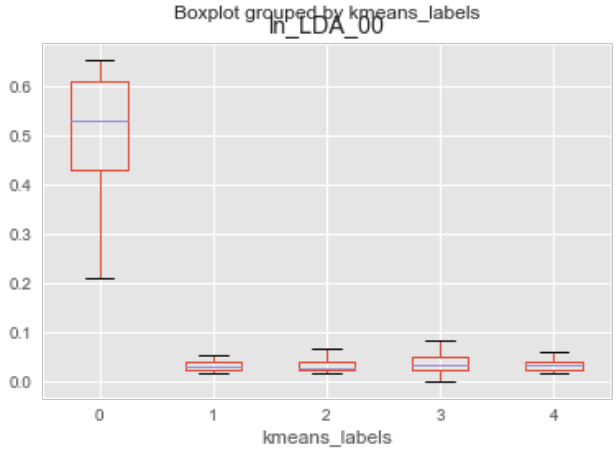
In [49]:
```
# ... recreate cluster model with max silhouette value

X1 = df_cluster[['ln_LDA_00','ln_LDA_01', 'ln_LDA_02', 'ln_LDA_03', 'ln_LDA_04']]

n_lda = 5

tic = time.clock()

cls_lda = KMeans(n_clusters = n_lda,
                 init = 'k-means++',
                 random_state = 1);
cls_lda.fit(X1);

kmeans_labels = cls_lda.labels_ # the labels from kmeans clustering
kmeans_centers = cls_lda.cluster_centers_
kmeans_inertia = cls_lda.inertia_

print ("n_lda, inertia ", n_lda, kmeans_inertia)

kmeans_silhouette = metrics.silhouette_score(X1,
                                             kmeans_labels,
                                             metric = 'euclidean',
                                             sample_size = 10000)
print ("silhouette = ", kmeans_silhouette)

toc =  time.clock()

X1['kmeans_labels'] = kmeans_labels

grouped = X1.groupby('kmeans_labels')

grouped.agg([np.mean, np.median, np.std]).T

# boxplot across clusters for each feature ...

col_names = X1.columns.values.tolist()

for col in col_names :
    fig = plt.figure()
    X1.boxplot(column = col, by = 'kmeans_labels')
#    ax.set_xticklabels(X1['kmeans_labels'], rotation=90)
    plt.show();
```

Out[49]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=5, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=1, tol=0.0001, verbose=0)

n_lda, inertia  5 1418.81957692
silhouette =  0.567669007332

/home/mcdevitt/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:30: SettingWithC
opyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.
html#indexing-view-versus-copy
```

Out[49]:

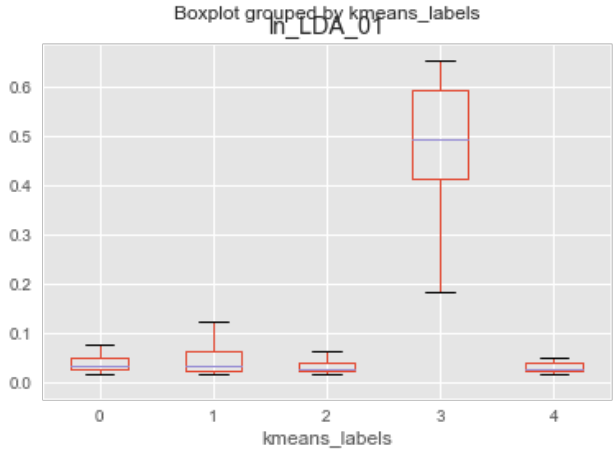
	kmeans_labels	0	1	2	3	4
ln_LDA_00	**mean**	0.514983	0.059556	0.068896	0.065855	0.059937
	median	0.531157	0.028840	0.028220	0.032791	0.032790
	std	0.100783	0.069824	0.084086	0.078472	0.071643
ln_LDA_01	**mean**	0.061619	0.075059	0.054507	0.494508	0.053473
	median	0.032832	0.032808	0.028173	0.495017	0.028236
	std	0.068309	0.086205	0.067439	0.107092	0.065778
ln_LDA_02	**mean**	0.075931	0.062138	0.089335	0.067757	0.531125
	median	0.039221	0.028257	0.032791	0.032790	0.550041
	std	0.084826	0.076345	0.097787	0.081612	0.094318
ln_LDA_03	**mean**	0.062824	0.540089	0.059080	0.115776	0.055090
	median	0.032914	0.555596	0.028172	0.040008	0.029813
	std	0.074425	0.090233	0.075166	0.115127	0.067240
ln_LDA_04	**mean**	0.091096	0.058594	0.527892	0.068599	0.099219
	median	0.039222	0.028369	0.540225	0.032790	0.039223
	std	0.097795	0.072578	0.096568	0.082624	0.103681

Out[49]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e26eb2b00>

<matplotlib.figure.Figure at 0x7f8e2c2e4e80>
```



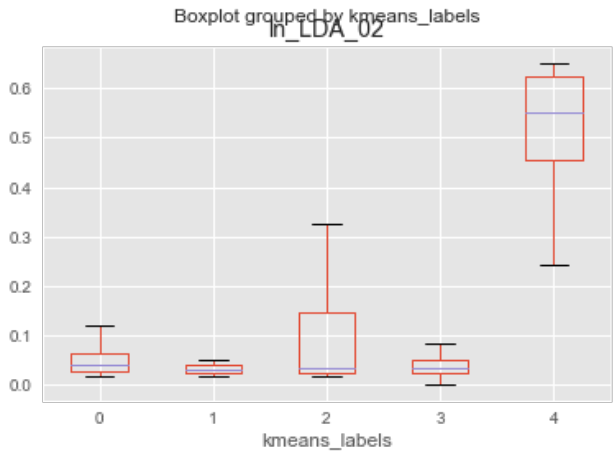
Out[49]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e26eb5400>

<matplotlib.figure.Figure at 0x7f8e340f8908>



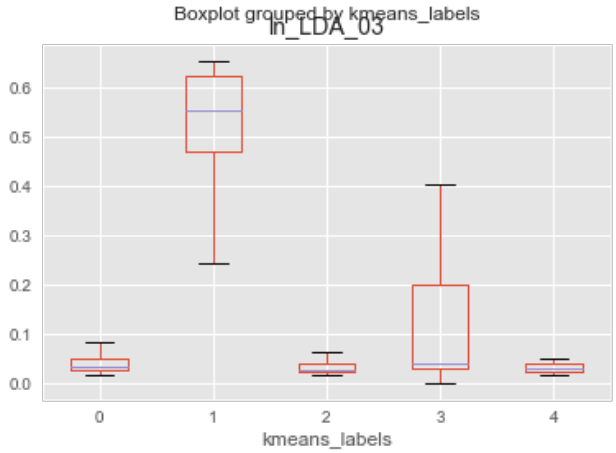
Out[49]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e3547f7b8>

<matplotlib.figure.Figure at 0x7f8e26e95cf8>



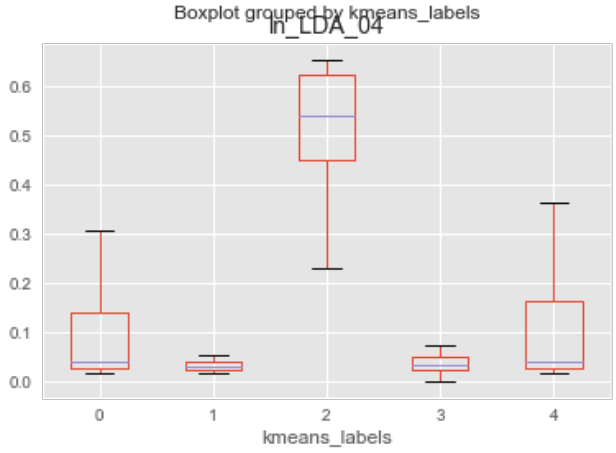
Out[49]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c1781d0>

```
<matplotlib.figure.Figure at 0x7f8e35b15518>
```



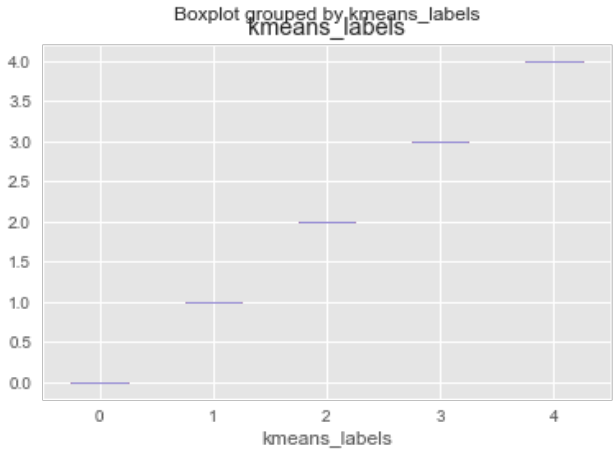
Out[49]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e26f89080>
```

```
<matplotlib.figure.Figure at 0x7f8e359843c8>
```



Out[49]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c075320>
```

```
<matplotlib.figure.Figure at 0x7f8e26df81d0>
```



K-Means - images & videos

In [ ]:

In [9]:
```
for n_lda in range(2, 10):

    tic = time.clock()

    X1 = df_cluster[['ln_num_imgs','ln_num_videos', 'ln_num_hrefs']]

    cls_lda = KMeans(n_clusters = n_lda,
                     init = 'k-means++',
                     random_state = 1)

    cls_lda.fit(X1)

    kmeans_labels = cls_lda.labels_ # the labels from kmeans clustering
    kmeans_centers = cls_lda.cluster_centers_
    kmeans_inertia = cls_lda.inertia_

    print ("n_lda = ", n_lda)
    print ("inertia = ", kmeans_inertia)

    kmeans_silhouette = metrics.silhouette_score(X1,
                                                 kmeans_labels,
                                                 metric = 'euclidean',
                                                 sample_size = 10000)
    print ("silhouette = ", kmeans_silhouette)

    toc =  time.clock()

# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ... - save statistics for model comparison
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    exe_time = '{0:.4f}'.format(toc-tic)

    raw_data = {
    'model_name' : 'KMeans - images_videos_hrefs features',
    'n_clusters' : n_lda,
    'inertia': kmeans_inertia,
    'silhouette': kmeans_silhouette,
    'process_time' : exe_time
    }

    df_tbl = pd.DataFrame(raw_data,
    columns = ['model_name', 'n_clusters', 'inertia', 'silhouette', 'process_time'],
    index = [i_index + 1])

    comparison_tbl = comparison_tbl.append(df_tbl)

# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ... - make some plots of clusters
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    plt.figure(figsize = (16, 6));

    plt.subplot(131);
    X1 = X1.values;
```

Out[9]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=2, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=1, tol=0.0001, verbose=0)

n_lda =  2
inertia =  49187.3739481
silhouette =  0.460287122859
```

Out[9]:
```
<matplotlib.figure.Figure at 0x7f8e2c5d3ba8>
```

Out[9]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e3413cc88>
```

Out[9]:
```
<matplotlib.collections.PathCollection at 0x7f8e340b2550>
```

Out[9]:
```
<matplotlib.collections.PathCollection at 0x7f8e340b2da0>
```

Out[9]:
```
<matplotlib.text.Text at 0x7f8e341c0e48>
```

Out[9]:
```
(<matplotlib.text.Text at 0x7f8e341399b0>,
 <matplotlib.text.Text at 0x7f8e341af550>)
```

Out[9]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e341af940>
```

Out[9]:
```
<matplotlib.collections.PathCollection at 0x7f8e3417e160>
```

Out[9]:
```
<matplotlib.collections.PathCollection at 0x7f8e3416efd0>
```

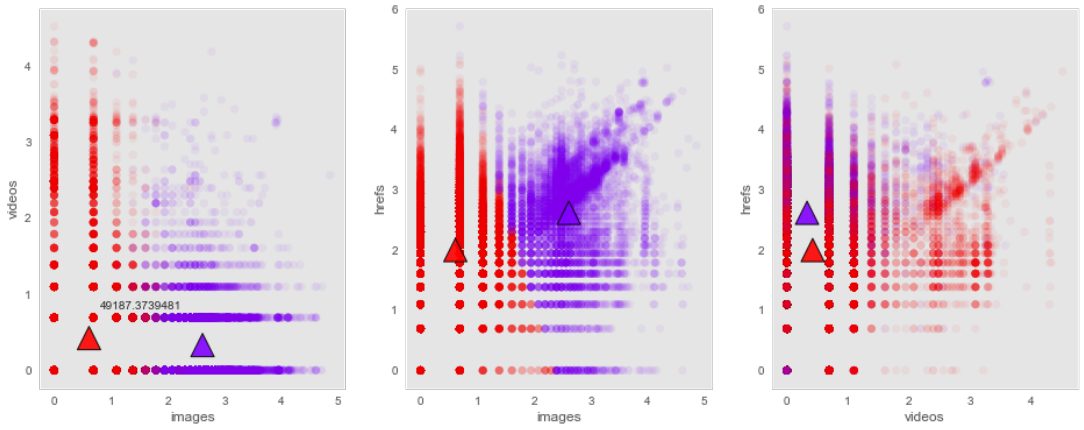
Out[9]:
```
(<matplotlib.text.Text at 0x7f8e3403e4e0>,
 <matplotlib.text.Text at 0x7f8e34037b70>)
```

Out[9]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e34187358>
```

Out[9]:
```
<matplotlib.collections.PathCollection at 0x7f8e2c517828>
```

Out[9]:
```
<matplotlib.collections.PathCollection at 0x7f8e2c51d0b8>
```

Out[9]:
```
(<matplotlib.text.Text at 0x7f8e3417e710>,
 <matplotlib.text.Text at 0x7f8e3416a278>)
```



Out[9]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=3, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=1, tol=0.0001, verbose=0)

n_lda =  3
inertia =  37625.0279826
silhouette =  0.471461115021
```

Out[9]: <matplotlib.figure.Figure at 0x7f8e34184828>

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e35bff048>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e35ea0a20>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e35b950b8>

Out[9]: <matplotlib.text.Text at 0x7f8e36165e48>

Out[9]: (<matplotlib.text.Text at 0x7f8e34238208>,
 <matplotlib.text.Text at 0x7f8e2c57d4e0>)

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e35b4e550>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e35b248d0>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e2c636b00>

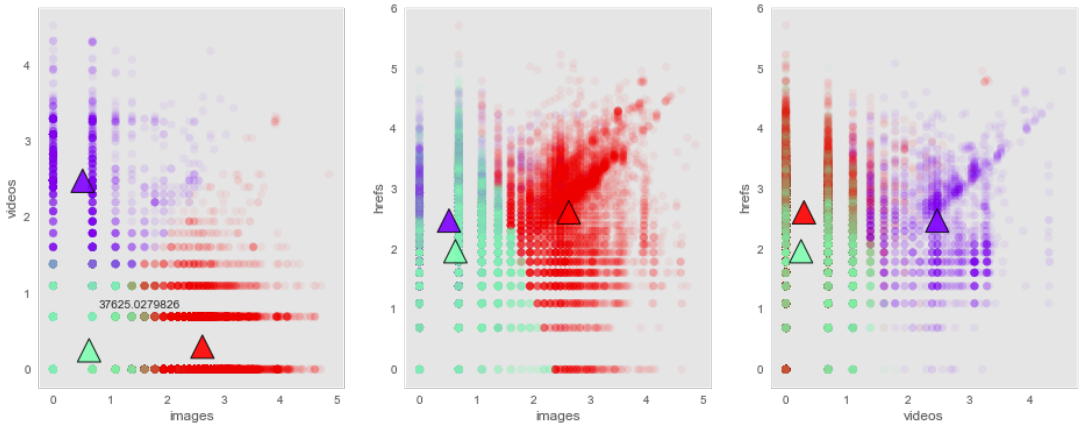
Out[9]: (<matplotlib.text.Text at 0x7f8e35b955c0>,
 <matplotlib.text.Text at 0x7f8e2c63f3c8>)

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e3617ce10>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e35aa3710>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e342814a8>

Out[9]: (<matplotlib.text.Text at 0x7f8e35b245f8>,
 <matplotlib.text.Text at 0x7f8e2c710940>)



Out[9]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=4, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=1, tol=0.0001, verbose=0)

```
n_lda =  4
inertia =  28917.9668688
silhouette =  0.346091511814
```

Out[9]: <matplotlib.figure.Figure at 0x7f8e2c63c0b8>

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e359728d0>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e3599c978>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e35a48e48>

Out[9]: <matplotlib.text.Text at 0x7f8e3599c1d0>

Out[9]: (<matplotlib.text.Text at 0x7f8e35a69a20>,
<matplotlib.text.Text at 0x7f8e359a1630>)

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e359f49b0>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e35a9d2b0>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e35bc5c50>

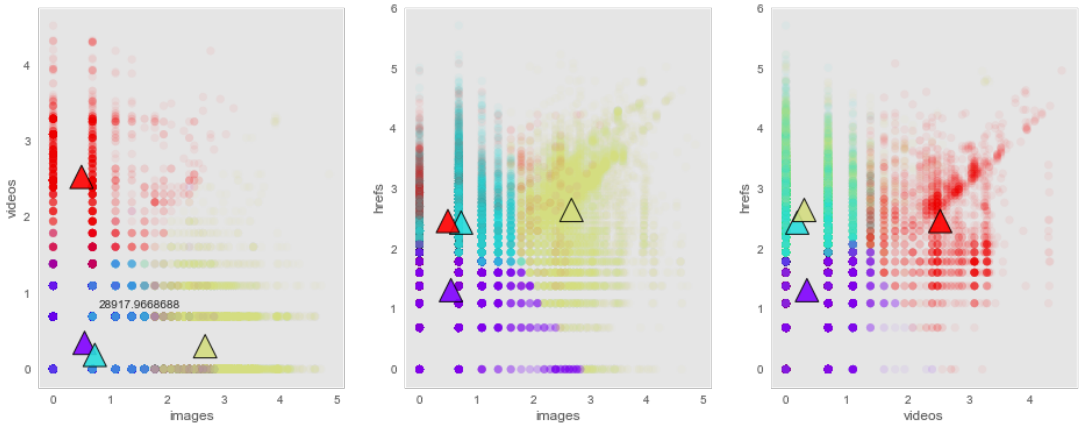
Out[9]: (<matplotlib.text.Text at 0x7f8e35a484a8>,
<matplotlib.text.Text at 0x7f8e359a11d0>)

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c670358>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e2c554b70>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e35b5a400>

Out[9]: (<matplotlib.text.Text at 0x7f8e35e83748>,
<matplotlib.text.Text at 0x7f8e35bc25c0>)



Out[9]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
n_clusters=5, n_init=10, n_jobs=1, precompute_distances='auto',
random_state=1, tol=0.0001, verbose=0)

n_lda = 5
inertia = 24309.3293114
silhouette = 0.343589314837

Out[9]: <matplotlib.figure.Figure at 0x7f8e2c50f0f0>

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e359227b8>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e35a916d8>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e35a6f080>

Out[9]: <matplotlib.text.Text at 0x7f8e35a91c50>

Out[9]: (<matplotlib.text.Text at 0x7f8e2c62aa58>,
<matplotlib.text.Text at 0x7f8e35ae3320>)

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e35a6fd68>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e2c589f98>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e2c58eb00>

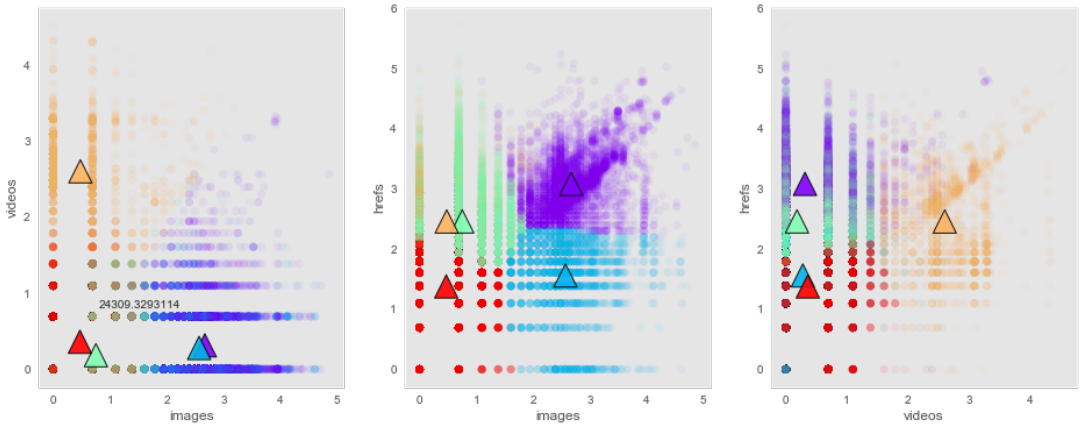
Out[9]: (<matplotlib.text.Text at 0x7f8e359ca2e8>,
 <matplotlib.text.Text at 0x7f8e35a71b70>)

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c590160>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e35bbed30>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e35c02198>

Out[9]: (<matplotlib.text.Text at 0x7f8e2c58d390>,
 <matplotlib.text.Text at 0x7f8e35a909b0>)



Out[9]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=6, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=1, tol=0.0001, verbose=0)

n_lda = 6
inertia = 21231.9882546
silhouette = 0.324759057672

Out[9]: <matplotlib.figure.Figure at 0x7f8e2c556fd0>

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e3595d5c0>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e35ab8208>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e359d4be0>

Out[9]: <matplotlib.text.Text at 0x7f8e35b5a5f8>

Out[9]: (<matplotlib.text.Text at 0x7f8e35b83438>,
 <matplotlib.text.Text at 0x7f8e35bbeb00>)

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e35b5abe0>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e2c7101d0>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e35b4ae10>

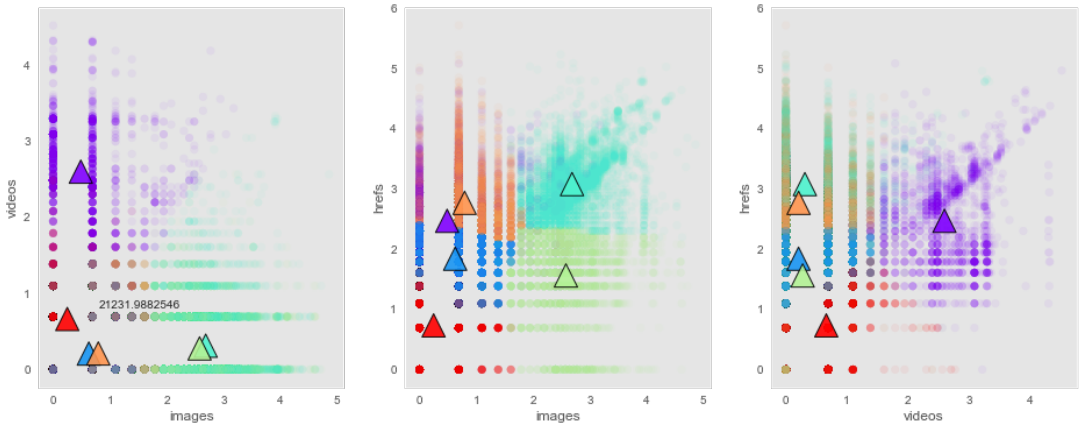
Out[9]: (<matplotlib.text.Text at 0x7f8e35addd68>,
 <matplotlib.text.Text at 0x7f8e2c660898>)

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c7109b0>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e341273c8>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e35989c88>

Out[9]: (<matplotlib.text.Text at 0x7f8e35b4a1d0>,
 <matplotlib.text.Text at 0x7f8e35bbe550>)



Out[9]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=7, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=1, tol=0.0001, verbose=0)

n_lda = 7
inertia = 18723.7392632
silhouette = 0.350159356293

Out[9]: <matplotlib.figure.Figure at 0x7f8e2c556a58>

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c61e748>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e2c63f390>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e342644a8>

Out[9]: <matplotlib.text.Text at 0x7f8e2c63f3c8>

Out[9]: (<matplotlib.text.Text at 0x7f8e2c63d748>,
 <matplotlib.text.Text at 0x7f8e35b4ecf8>)

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c63f128>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e2c67acf8>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e3596ae48>

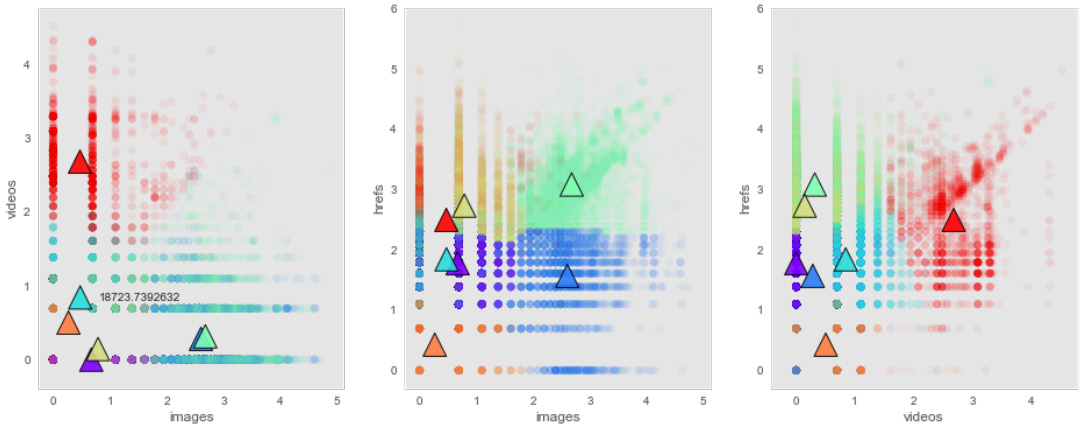
Out[9]: (<matplotlib.text.Text at 0x7f8e340a6710>,
 <matplotlib.text.Text at 0x7f8e2c63e358>)

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c67aeb8>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e34146fd0>

Out[9]: <matplotlib.collections.PathCollection at 0x7f8e341d99b0>

Out[9]: (<matplotlib.text.Text at 0x7f8e3596a5c0>,
 <matplotlib.text.Text at 0x7f8e35bba390>)



Out[9]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=8, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=1, tol=0.0001, verbose=0)

n_lda =  8
inertia =  17210.2410622
silhouette =  0.331527856696
```

Out[9]:
```
<matplotlib.figure.Figure at 0x7f8e35ac8710>
```

Out[9]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e35b83f60>
```

Out[9]:
```
<matplotlib.collections.PathCollection at 0x7f8e35504b38>
```

Out[9]:
```
<matplotlib.collections.PathCollection at 0x7f8e354ec3c8>
```

Out[9]:
```
<matplotlib.text.Text at 0x7f8e35504b70>
```

Out[9]:
```
(<matplotlib.text.Text at 0x7f8e35b37e48>,
 <matplotlib.text.Text at 0x7f8e34169d30>)
```

Out[9]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e340565c0>
```

Out[9]:
```
<matplotlib.collections.PathCollection at 0x7f8e34205f28>
```

Out[9]:
```
<matplotlib.collections.PathCollection at 0x7f8e3596a080>
```

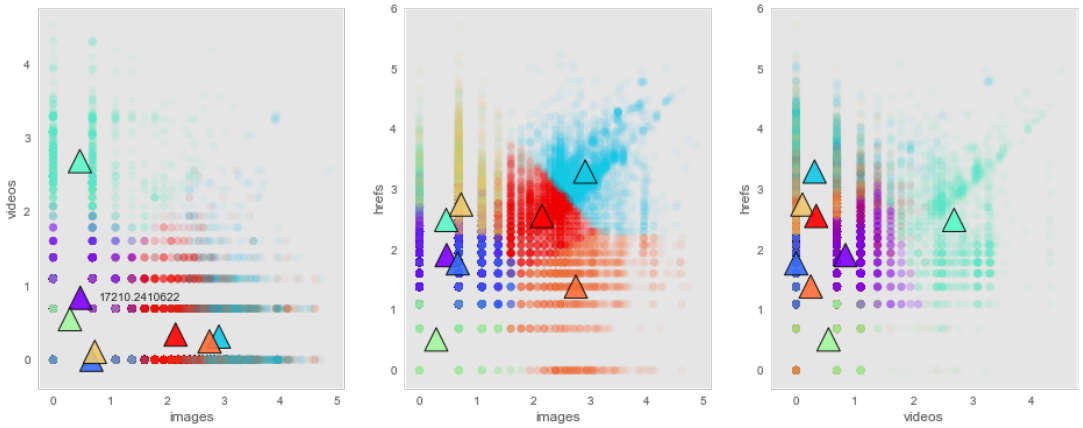
Out[9]:
```
(<matplotlib.text.Text at 0x7f8e2c51f2b0>,
 <matplotlib.text.Text at 0x7f8e2c57def0>)
```

Out[9]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e34205fd0>
```

Out[9]:
```
<matplotlib.collections.PathCollection at 0x7f8e341c4438>
```

Out[9]:
```
<matplotlib.collections.PathCollection at 0x7f8e341adc88>
```

Out[9]:
```
(<matplotlib.text.Text at 0x7f8e3596add8>,
 <matplotlib.text.Text at 0x7f8e35a12a58>)
```



Out[9]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=9, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=1, tol=0.0001, verbose=0)

n_lda =  9
inertia =  15960.0254427
silhouette =  0.318011435102
```

Out[9]:
```
<matplotlib.figure.Figure at 0x7f8e2c66e9e8>
```

Out[9]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e341d9eb8>
```

Out[9]:
```
<matplotlib.collections.PathCollection at 0x7f8e35b9fb70>
```

Out[9]:
```
<matplotlib.collections.PathCollection at 0x7f8e35a699e8>
```

Out[9]:
```
<matplotlib.text.Text at 0x7f8e35b9ff28>
```

Out[9]:
```
(<matplotlib.text.Text at 0x7f8e3596a0f0>,
 <matplotlib.text.Text at 0x7f8e35c08278>)
```

Out[9]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e35b9f240>
```

Out[9]:
```
<matplotlib.collections.PathCollection at 0x7f8e35add470>
```

Out[9]:
```
<matplotlib.collections.PathCollection at 0x7f8e35bba4e0>
```

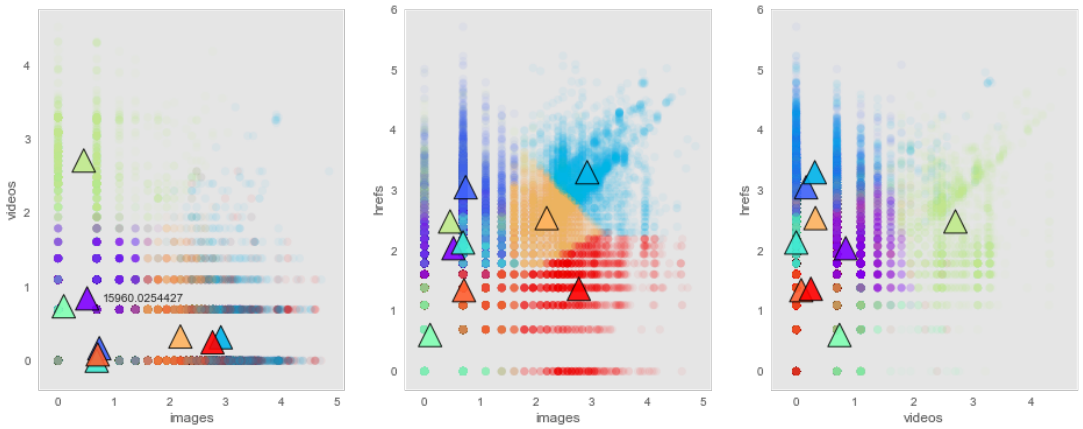
Out[9]:
```
(<matplotlib.text.Text at 0x7f8e35a69908>,
 <matplotlib.text.Text at 0x7f8e35a24d68>)
```

Out[9]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c64dc18>
```

Out[9]:
```
<matplotlib.collections.PathCollection at 0x7f8e35bc52e8>
```

Out[9]:
```
<matplotlib.collections.PathCollection at 0x7f8e35992470>
```

Out[9]:
```
(<matplotlib.text.Text at 0x7f8e34127be0>,
 <matplotlib.text.Text at 0x7f8e35bd46d8>)
```



Out[9]:

	model_name	n_clusters	inertia	silhouette	process_time
1	KMeans - LDA features	2	6016.621217	0.266625	2.7176
1	KMeans - LDA features	3	4120.020638	0.384512	2.9431
1	KMeans - LDA features	4	2523.834033	0.501645	3.0021
1	KMeans - LDA features	5	1418.819577	0.564549	3.1371
1	KMeans - LDA features	6	1257.642737	0.521217	3.9218
1	KMeans - LDA features	7	1135.466961	0.506604	4.3356
1	KMeans - LDA features	8	1023.646217	0.470088	4.4810
1	KMeans - LDA features	9	937.873676	0.474418	4.9386
1	KMeans - LDA features	10	859.164016	0.479736	4.6344
1	KMeans - LDA features	11	794.230421	0.493377	5.6503
1	KMeans - images_videos_hrefs features	2	49187.373948	0.460287	2.6638
1	KMeans - images_videos_hrefs features	3	37625.027983	0.471461	3.0407
1	KMeans - images_videos_hrefs features	4	28917.966869	0.346092	3.2196
1	KMeans - images_videos_hrefs features	5	24309.329311	0.343589	3.5388
1	KMeans - images_videos_hrefs features	6	21231.988255	0.324759	3.7828
1	KMeans - images_videos_hrefs features	7	18723.739263	0.350159	3.8868
1	KMeans - images_videos_hrefs features	8	17210.241062	0.331528	4.6341
1	KMeans - images_videos_hrefs features	9	15960.025443	0.318011	5.0193

In [10]:
```
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ... - plot metrics across models for comparison
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

plt.figure(figsize=(16, 6));

# ... silhouette values

plt.subplot(131);
plt.scatter(comparison_tbl['n_clusters'],
            comparison_tbl['silhouette'],
            s = 40,
            linewidths = 1.0,
            marker = '^',
            edgecolors = 'black',
            alpha = 0.90);

plt.plot(comparison_tbl['n_clusters'],
         comparison_tbl['silhouette'])

plt.xlabel('n_clusters'), plt.ylabel('silhouette');
plt.grid();

# ... inertia values

plt.subplot(132);
plt.scatter(comparison_tbl['n_clusters'],
            comparison_tbl['inertia'],
            s = 40,
            linewidths = 1.0,
            marker = '^',
            edgecolors = 'black',
            alpha = 0.90);

plt.plot(comparison_tbl['n_clusters'],
         comparison_tbl['inertia'])

plt.xlabel('n_clusters'), plt.ylabel('inertia');
plt.grid();

# ... process time

plt.subplot(133);
plt.scatter(comparison_tbl['n_clusters'],
            comparison_tbl['process_time'],
            s = 40,
            linewidths = 1.0,
            marker = '^',
            edgecolors = 'black',
            alpha = 0.90);

#plt.plot(comparison_tbl['n_clusters'],
#         comparison_tbl['process_time'])

plt.xlabel('n_clusters'), plt.ylabel('process_time');
```

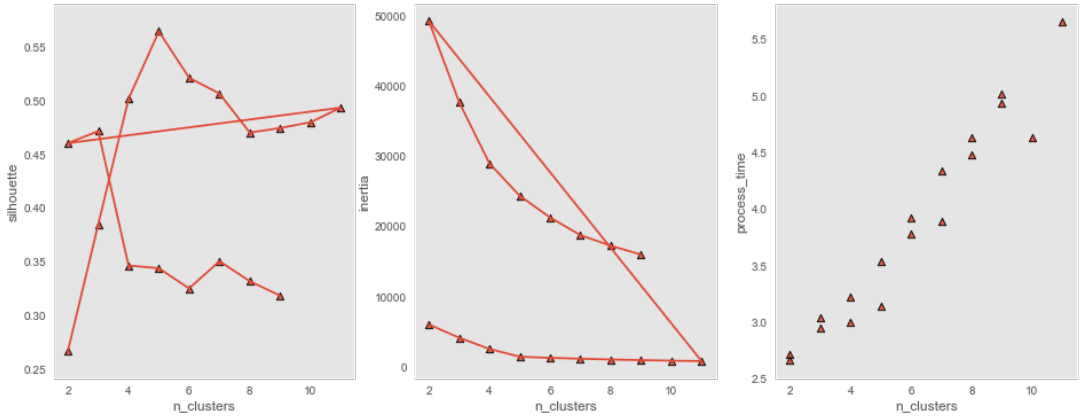


Table of Contents

KMeans - all_in

K-Means - All in

In [11]:
```
# set required variables for model comparison

all_in_tbl = comparison_tbl.reset_index(drop = True)

all_in_tbl = pd.DataFrame(columns = [
    'model_name',
    'n_clusters',
    'inertia',
    'silhouette',
    'process_time'])

i_index = []
i_index = 0

# preparation for cross validation and model comparison, each classifier is appended once mode
l is fit

models = []
```

In [12]:
```
X1 = df_cluster

for n_lda in range(2, 12):

    tic = time.clock()

    cls_lda = KMeans(n_clusters = n_lda,
                     init = 'k-means++',
                     random_state = 1);
    cls_lda.fit(X1);

    kmeans_labels = cls_lda.labels_ # the labels from kmeans clustering
    kmeans_centers = cls_lda.cluster_centers_
    kmeans_inertia = cls_lda.inertia_

    print ("n_lda, inertia ", n_lda, kmeans_inertia)

    kmeans_silhouette = metrics.silhouette_score(X1,
                                                 kmeans_labels,
                                                 metric = 'euclidean',
                                                 sample_size = 10000)
    print ("silhouette = ", kmeans_silhouette)

    toc =  time.clock()
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ... - save statistics for model comparison
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    exe_time = '{0:.4f}'.format(toc-tic)

    raw_data = {
    'model_name' : 'KMeans - all_in',
    'n_clusters' : n_lda,
    'inertia': kmeans_inertia,
    'silhouette': kmeans_silhouette,
    'process_time' : exe_time
    }

    df_tbl = pd.DataFrame(raw_data,
    columns = ['model_name', 'n_clusters', 'inertia', 'silhouette', 'process_time'],
    index = [i_index + 1])

    all_in_tbl = all_in_tbl.append(df_tbl)

# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ... - make some plots of clusters
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

all_in_tbl
```

```
Out[12]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
             n_clusters=2, n_init=10, n_jobs=1, precompute_distances='auto',
             random_state=1, tol=0.0001, verbose=0)

         n_lda, inertia  2 1400523.17552
         silhouette =  0.340781525559

Out[12]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
             n_clusters=3, n_init=10, n_jobs=1, precompute_distances='auto',
             random_state=1, tol=0.0001, verbose=0)

         n_lda, inertia  3 1144630.43362
         silhouette =  0.347675692376

Out[12]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
             n_clusters=4, n_init=10, n_jobs=1, precompute_distances='auto',
             random_state=1, tol=0.0001, verbose=0)

         n_lda, inertia  4 999391.40361
         silhouette =  0.355824631818

Out[12]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
             n_clusters=5, n_init=10, n_jobs=1, precompute_distances='auto',
             random_state=1, tol=0.0001, verbose=0)

         n_lda, inertia  5 920707.821029
         silhouette =  0.262109105863

Out[12]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
             n_clusters=6, n_init=10, n_jobs=1, precompute_distances='auto',
             random_state=1, tol=0.0001, verbose=0)

         n_lda, inertia  6 865749.225149
         silhouette =  0.191145016442

Out[12]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
             n_clusters=7, n_init=10, n_jobs=1, precompute_distances='auto',
             random_state=1, tol=0.0001, verbose=0)

         n_lda, inertia  7 826467.6524
         silhouette =  0.18100865321

Out[12]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
             n_clusters=8, n_init=10, n_jobs=1, precompute_distances='auto',
             random_state=1, tol=0.0001, verbose=0)

         n_lda, inertia  8 790950.316967
         silhouette =  0.184835919155

Out[12]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
             n_clusters=9, n_init=10, n_jobs=1, precompute_distances='auto',
             random_state=1, tol=0.0001, verbose=0)

         n_lda, inertia  9 760728.643599
         silhouette =  0.190296465922

Out[12]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
             n_clusters=10, n_init=10, n_jobs=1, precompute_distances='auto',
             random_state=1, tol=0.0001, verbose=0)
```

```
n_lda, inertia  10 737976.545004
silhouette =  0.185645502704
```

Out[12]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=11, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=1, tol=0.0001, verbose=0)
```

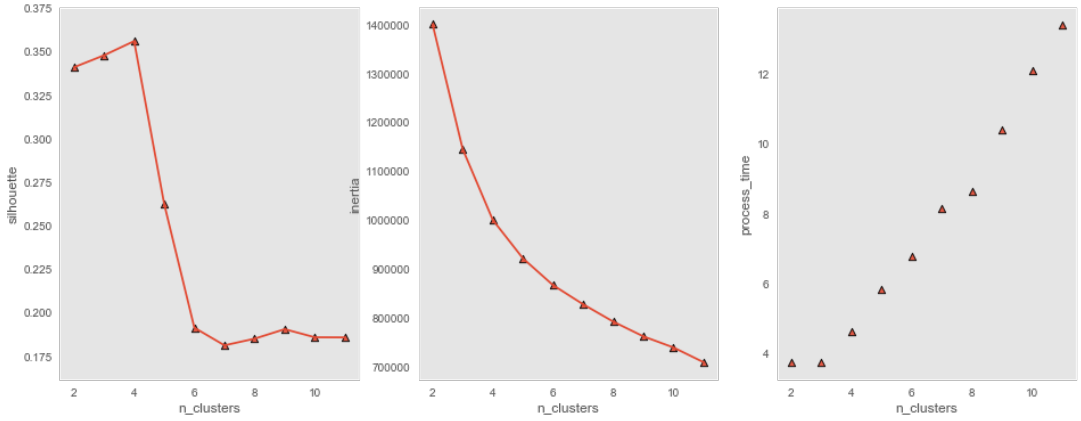
```
n_lda, inertia  11 707754.988617
silhouette =  0.18557980475
```

Out[12]:

	model_name	n_clusters	inertia	silhouette	process_time
1	KMeans - all_in	2	1.400523e+06	0.340782	3.7438
1	KMeans - all_in	3	1.144630e+06	0.347676	3.7416
1	KMeans - all_in	4	9.993914e+05	0.355825	4.6129
1	KMeans - all_in	5	9.207078e+05	0.262109	5.8243
1	KMeans - all_in	6	8.657492e+05	0.191145	6.7605
1	KMeans - all_in	7	8.264677e+05	0.181009	8.1327
1	KMeans - all_in	8	7.909503e+05	0.184836	8.6197
1	KMeans - all_in	9	7.607286e+05	0.190296	10.4076
1	KMeans - all_in	10	7.379765e+05	0.185646	12.0887
1	KMeans - all_in	11	7.077550e+05	0.185580	13.4126

In [13]:

```
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ... - plot metrics across models for comparison
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

plt.figure(figsize=(16, 6));

# ... silhouette values

plt.subplot(131);
plt.scatter(all_in_tbl['n_clusters'],
            all_in_tbl['silhouette'],
            s = 40,
            linewidths = 1.0,
            marker = '^',
            edgecolors = 'black',
            alpha = 0.90);

plt.plot(all_in_tbl['n_clusters'],
         all_in_tbl['silhouette'])

plt.xlabel('n_clusters'), plt.ylabel('silhouette');
plt.grid();

# ... inertia values

plt.subplot(132);
plt.scatter(all_in_tbl['n_clusters'],
            all_in_tbl['inertia'],
            s = 40,
            linewidths = 1.0,
            marker = '^',
            edgecolors = 'black',
            alpha = 0.90);

plt.plot(all_in_tbl['n_clusters'],
         all_in_tbl['inertia'])

plt.xlabel('n_clusters'), plt.ylabel('inertia');
plt.grid();

# ... process time

plt.subplot(133);
plt.scatter(all_in_tbl['n_clusters'],
            all_in_tbl['process_time'],
            s = 40,
            linewidths = 1.0,
            marker = '^',
            edgecolors = 'black',
            alpha = 0.90);

#plt.plot(all_in_tbl['n_clusters'],
#         all_in_tbl['process_time'])

plt.xlabel('n_clusters'), plt.ylabel('process_time');
```


silhouette
0.375
0.350
0.325
0.300
0.275
0.250
0.225
0.200
0.175
2
4
6
8
10
n_clusters
inertia
1400000
1300000
1200000
1100000
1000000
900000
800000
700000
2
4
6
8
10
n_clusters
process_time
12
10
8
6
4
2
4
6
8
10
n_clusters

In [14]:
```
# ... recreate cluster model with max silhouette value

X1 = df_cluster

n_lda = 4

tic = time.clock()

cls_lda = KMeans(n_clusters = n_lda,
                 init = 'k-means++',
                 random_state = 1);
cls_lda.fit(X1);

kmeans_labels = cls_lda.labels_ # the labels from kmeans clustering
kmeans_centers = cls_lda.cluster_centers_
kmeans_inertia = cls_lda.inertia_

print ("n_lda, inertia ", n_lda, kmeans_inertia)

kmeans_silhouette = metrics.silhouette_score(X1,
                                             kmeans_labels,
                                             metric = 'euclidean',
                                             sample_size = 10000)
print ("silhouette = ", kmeans_silhouette)

toc =  time.clock()

kmeans_labels

len(kmeans_labels)

X1['kmeans_labels'] = kmeans_labels

col_names = X1.columns.values.tolist()
```

Out[14]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=4, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=1, tol=0.0001, verbose=0)

n_lda, inertia  4 999391.40361
silhouette =  0.357899895468
```

In [46]:
```
pd.options.display.max_rows = 999

grouped = X1.groupby('kmeans_labels')

grouped.agg([np.mean, np.median, np.std]).T
```

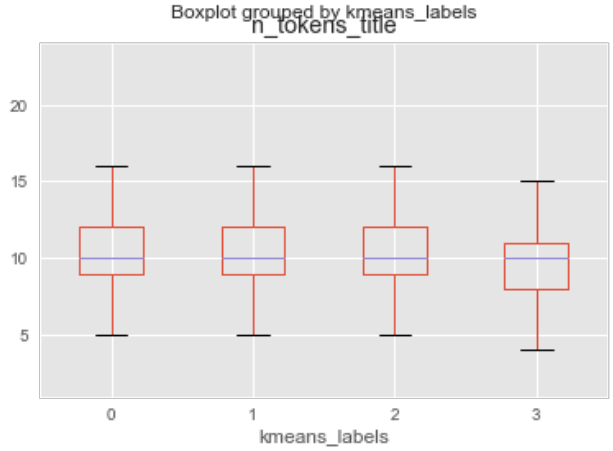
Out[46]:

	kmeans_labels	0	1	2	
n_tokens_title	**mean**	10.452100	10.467491	10.320638	9.939
	median	10.000000	10.000000	10.000000	10.00
	std	2.140902	2.098862	2.030432	2.050
num_keywords	**mean**	6.861258	7.698386	6.762251	7.807
	median	7.000000	8.000000	7.000000	8.000
	std	1.975838	1.675623	2.014037	1.710
kw_avg_max	**mean**	2.173935	1.636641	2.143115	1.351
	median	2.052661	1.571041	1.934420	1.322
	std	0.981161	0.853675	1.186846	0.892
weekday_is_monday	**mean**	0.165594	0.177133	0.157939	0.156
	median	0.000000	0.000000	0.000000	0.000
	std	0.371726	0.381795	0.364735	0.363
weekday_is_tuesday	**mean**	0.186643	0.188463	0.169700	0.194
	median	0.000000	0.000000	0.000000	0.000
	std	0.389635	0.391096	0.375422	0.395
weekday_is_wednesday	**mean**	0.185007	0.189129	0.185102	0.197
	median	0.000000	0.000000	0.000000	0.000
	std	0.388314	0.391626	0.388435	0.398
weekday_is_thursday	**mean**	0.178044	0.184612	0.192943	0.196
	median	0.000000	0.000000	0.000000	0.000
	std	0.382560	0.387997	0.394664	0.397
weekday_is_friday	**mean**	0.145706	0.139514	0.145617	0.148
	median	0.000000	0.000000	0.000000	0.000
	std	0.352820	0.346495	0.352772	0.355
is_weekend	**mean**	0.139006	0.121149	0.148698	0.107
	median	0.000000	0.000000	0.000000	0.000
	std	0.345962	0.326313	0.355840	0.309
LDA_00	**mean**	0.178494	0.173953	0.217198	0.224
	median	0.040000	0.033334	0.040000	0.033
	std	0.251981	0.257372	0.298051	0.295
LDA_01	**mean**	0.141669	0.149068	0.125422	0.125
	median	0.034243	0.029911	0.033454	0.028
	std	0.215490	0.233111	0.202931	0.203
LDA_02	**mean**	0.206819	0.206952	0.258379	0.259
	median	0.040076	0.033393	0.040354	0.050

In [38]:
```
# boxplot across clusters for each feature ...

for col in col_names :
    fig = plt.figure()
    X1.boxplot(column = col, by = 'kmeans_labels')
#    ax.set_xticklabels(X1['kmeans_labels'], rotation=90)
    plt.show();
```

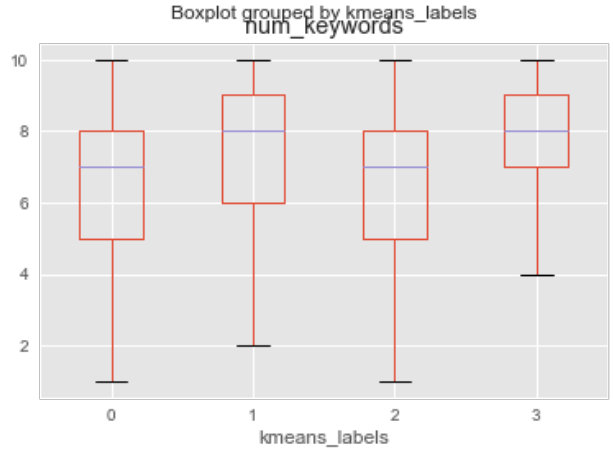
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c6fd828>

<matplotlib.figure.Figure at 0x7f8e35a6a080>



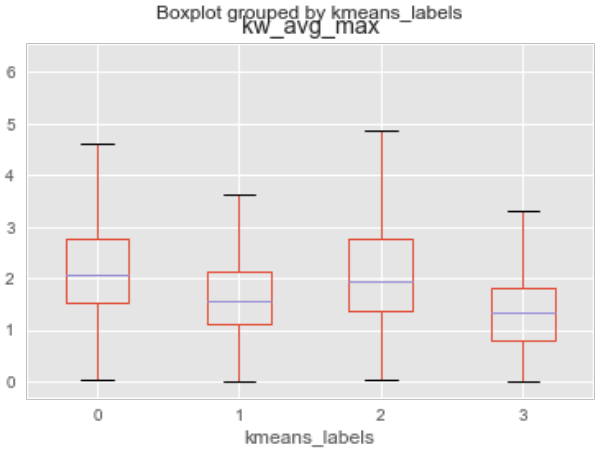
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e359a6b38>

<matplotlib.figure.Figure at 0x7f8e2c3e4470>



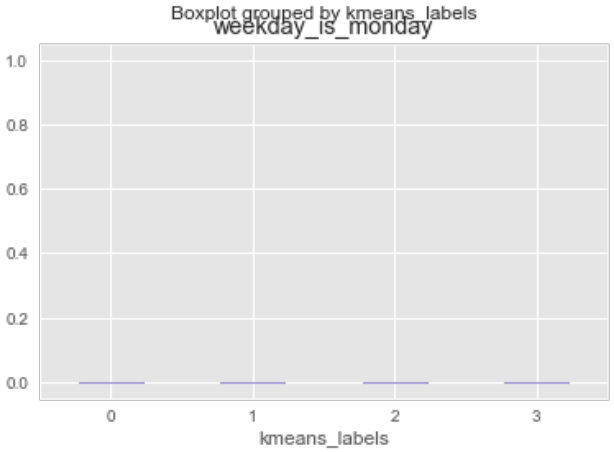
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e35ac4550>

<matplotlib.figure.Figure at 0x7f8e359c19b0>



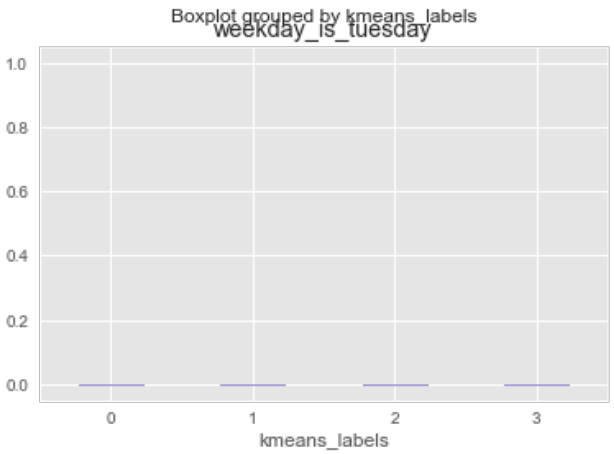
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c0dd400>

<matplotlib.figure.Figure at 0x7f8e2c396c18>



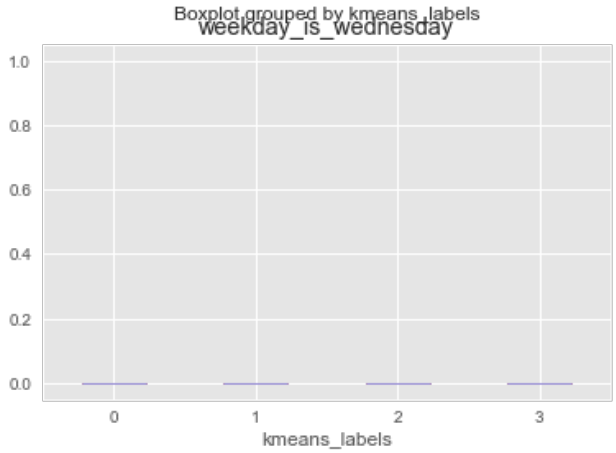
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e26fb0198>

<matplotlib.figure.Figure at 0x7f8e35e8af28>



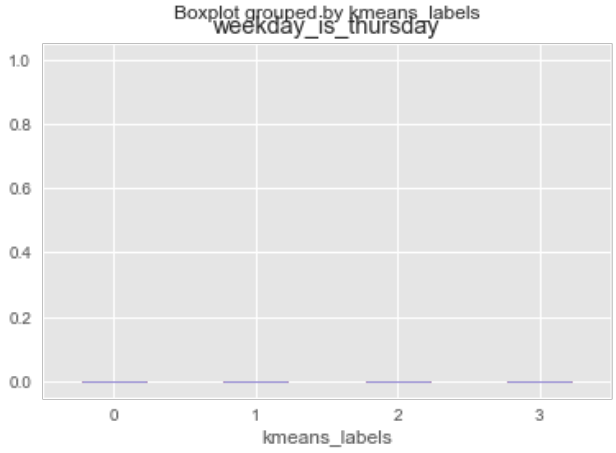
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c314e10>

```
<matplotlib.figure.Figure at 0x7f8e26edbfd0>
```



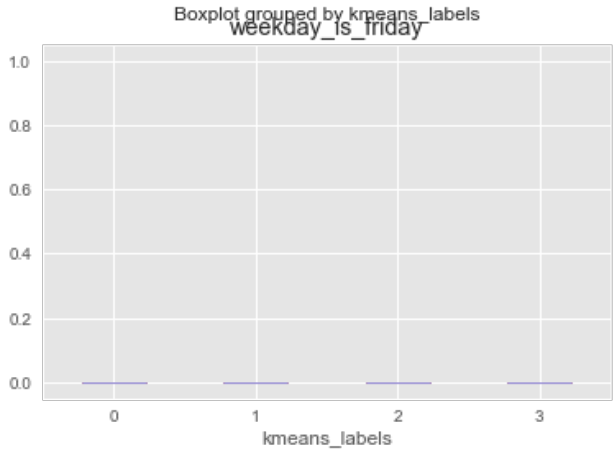
Out[38]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c06f400>
```

```
<matplotlib.figure.Figure at 0x7f8e26ef8c88>
```



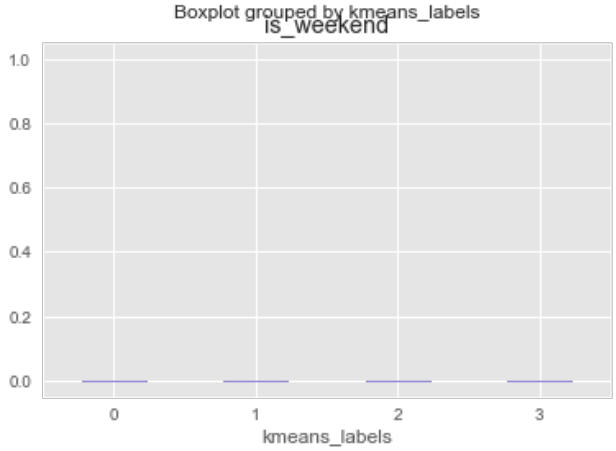
Out[38]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c09dc50>
```

```
<matplotlib.figure.Figure at 0x7f8e2c366828>
```



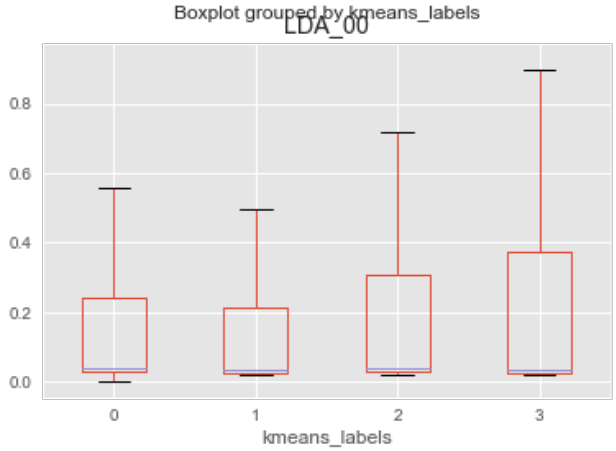
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e26e78f98>

<matplotlib.figure.Figure at 0x7f8e2c213a58>



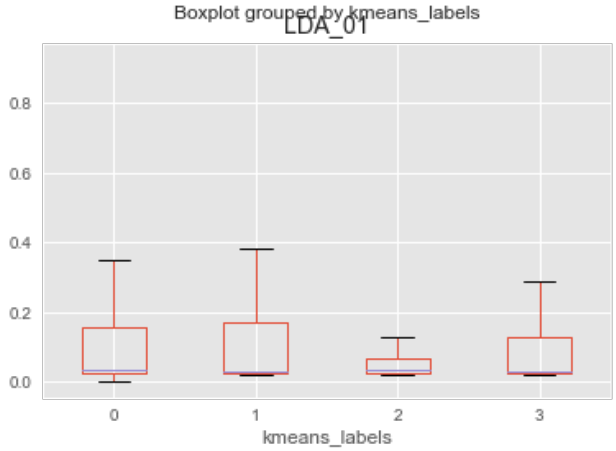
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e26d09908>

<matplotlib.figure.Figure at 0x7f8e2c2bfa90>



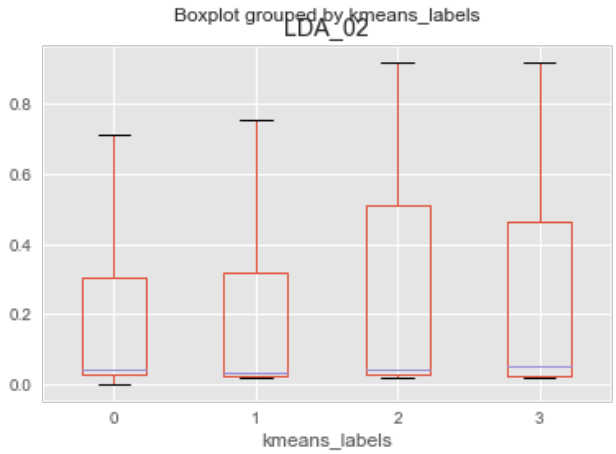
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e26efd710>

<matplotlib.figure.Figure at 0x7f8e26eb3b70>



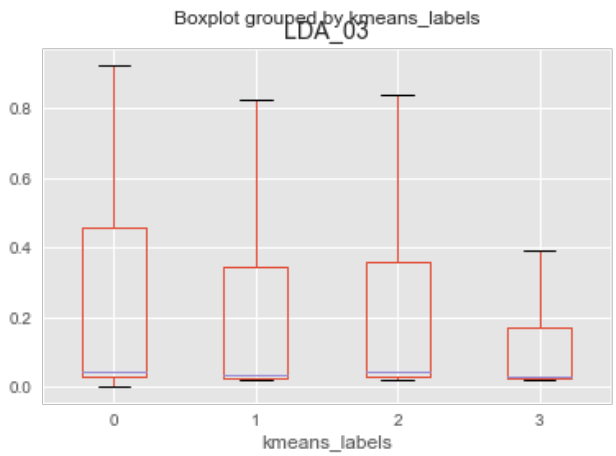
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c3fa470>

<matplotlib.figure.Figure at 0x7f8e6c881160>



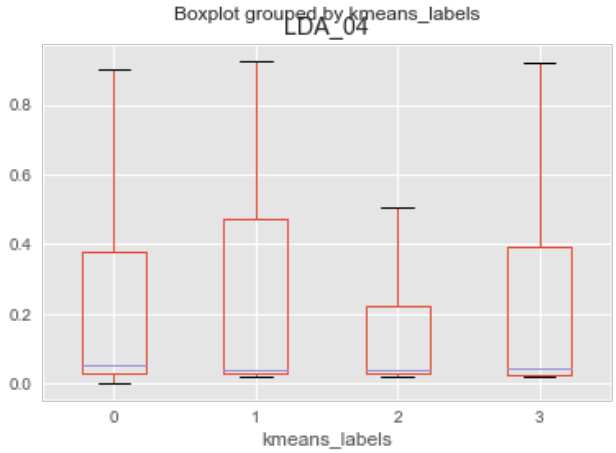
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e35a34978>

<matplotlib.figure.Figure at 0x7f8e2c589198>



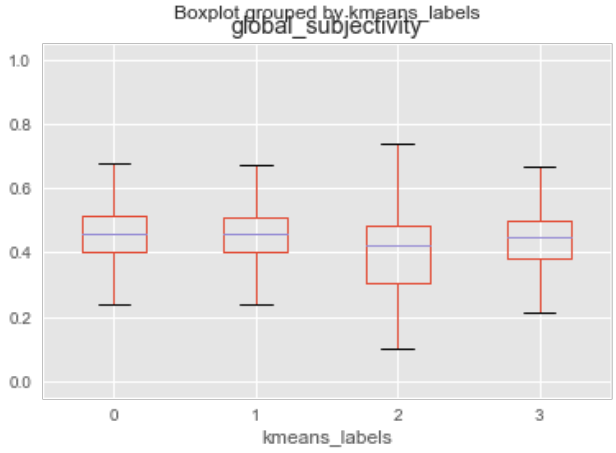
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e26f10ef0>

```
<matplotlib.figure.Figure at 0x7f8e35a93400>
```



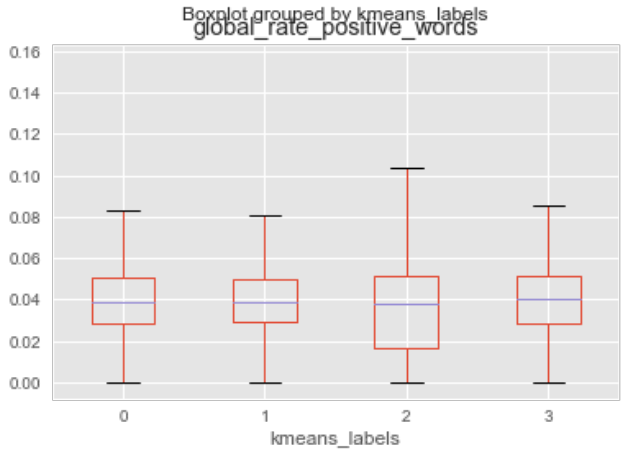
Out[38]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c0f4198>
```

```
<matplotlib.figure.Figure at 0x7f8e2c3fa588>
```



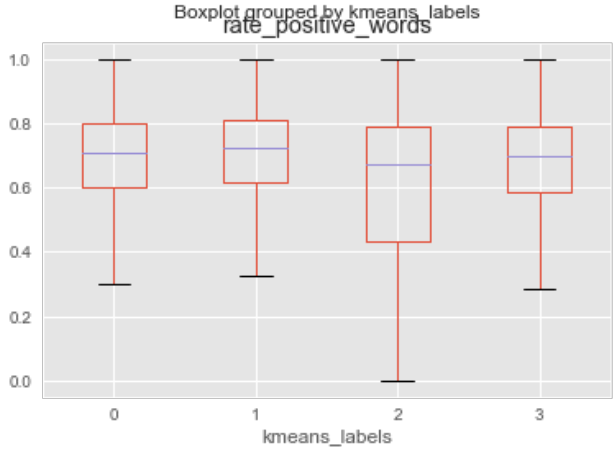
Out[38]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c20c1d0>
```

```
<matplotlib.figure.Figure at 0x7f8e34173ba8>
```



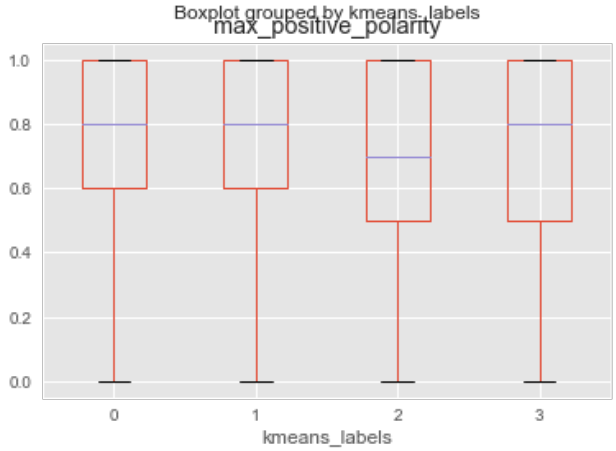
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c34f9e8>

<matplotlib.figure.Figure at 0x7f8e26e9c908>



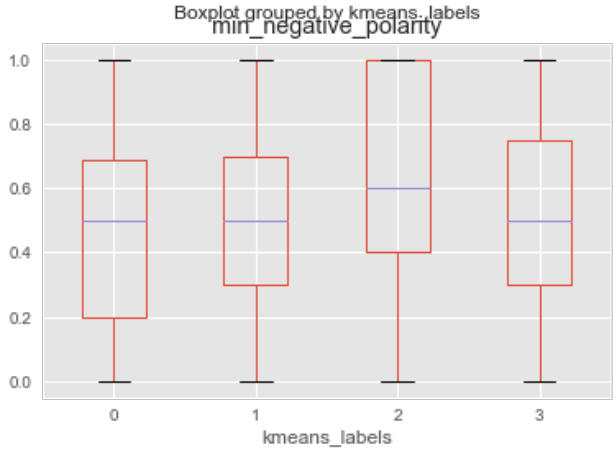
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c0dc898>

<matplotlib.figure.Figure at 0x7f8e26dc6908>



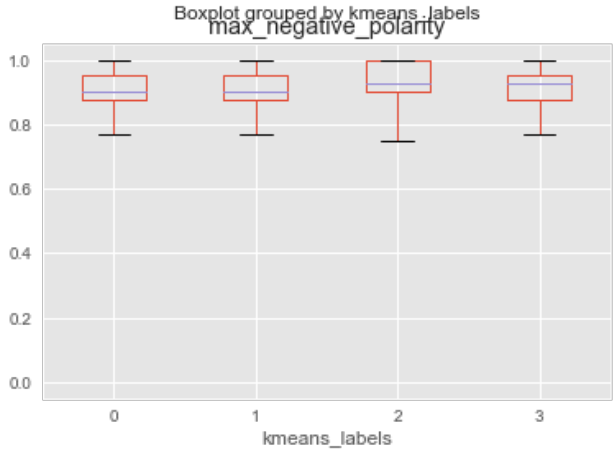
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e26fc0b70>

<matplotlib.figure.Figure at 0x7f8e2c5894a8>



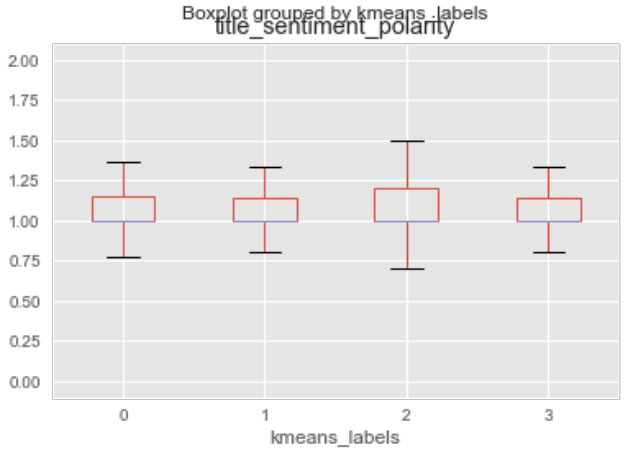
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c41ecf8>

<matplotlib.figure.Figure at 0x7f8e26fa9e80>



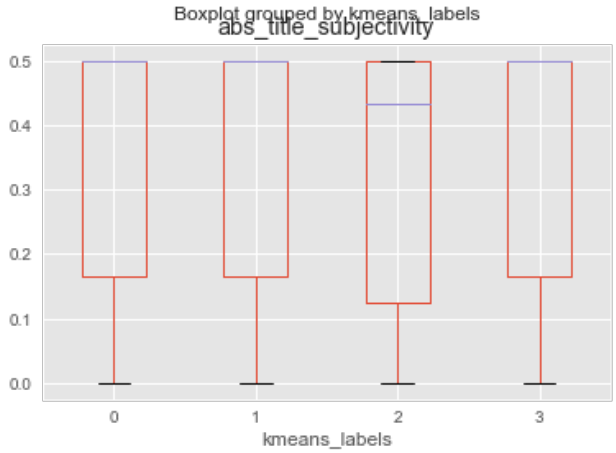
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e26d917b8>

<matplotlib.figure.Figure at 0x7f8e26f45be0>



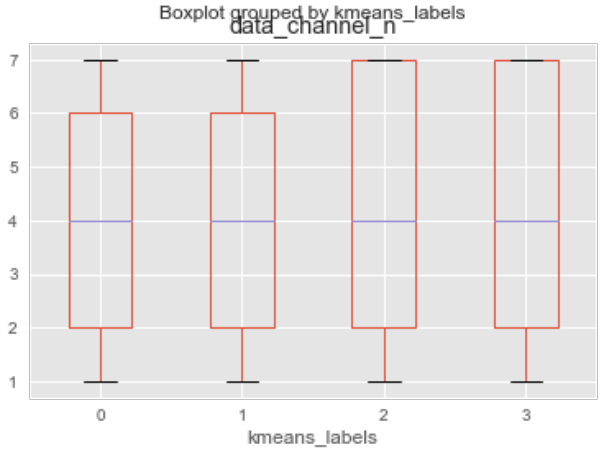
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e26e100f0>

```
<matplotlib.figure.Figure at 0x7f8e26d0cbe0>
```



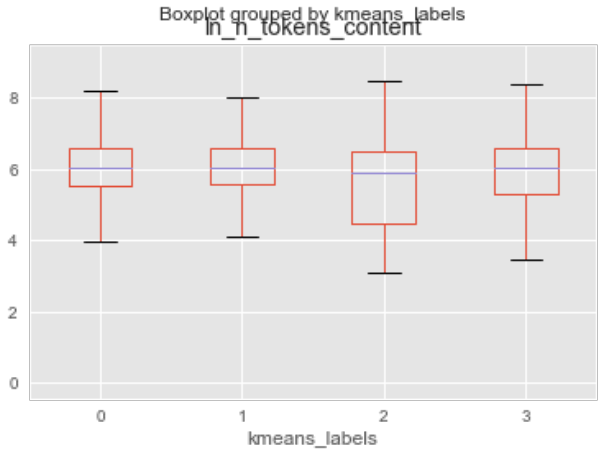
Out[38]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c461b38>
```

```
<matplotlib.figure.Figure at 0x7f8e26fc6eb8>
```



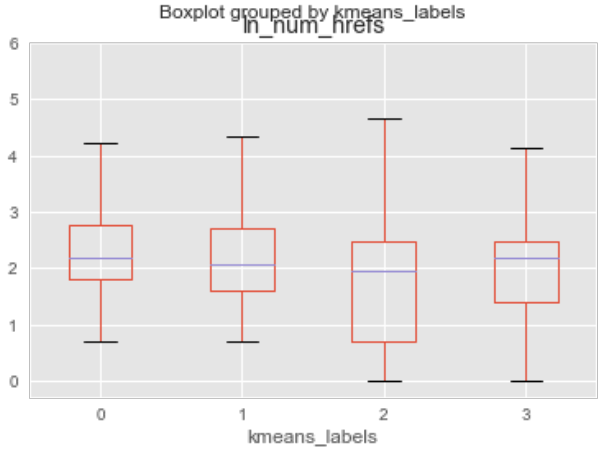
Out[38]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e26fd1780>
```

```
<matplotlib.figure.Figure at 0x7f8e2c55d518>
```



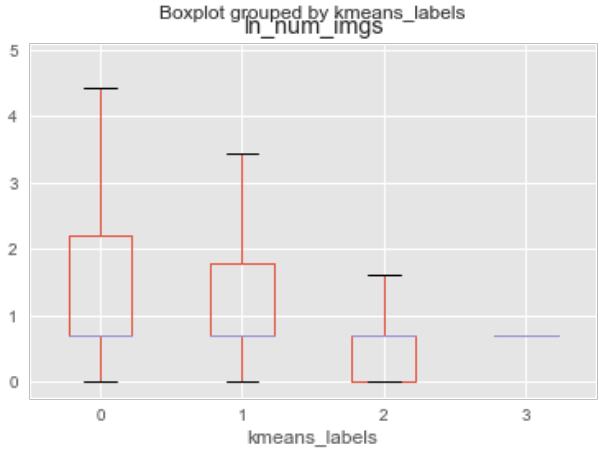
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c36bcf8>

<matplotlib.figure.Figure at 0x7f8e2c461fd0>



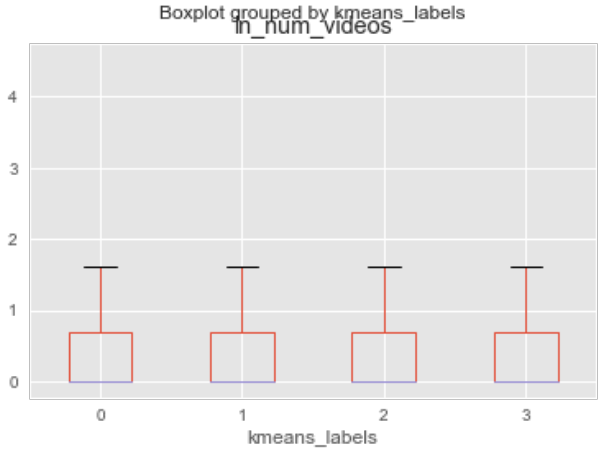
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e3411a0b8>

<matplotlib.figure.Figure at 0x7f8e2c0f9dd8>



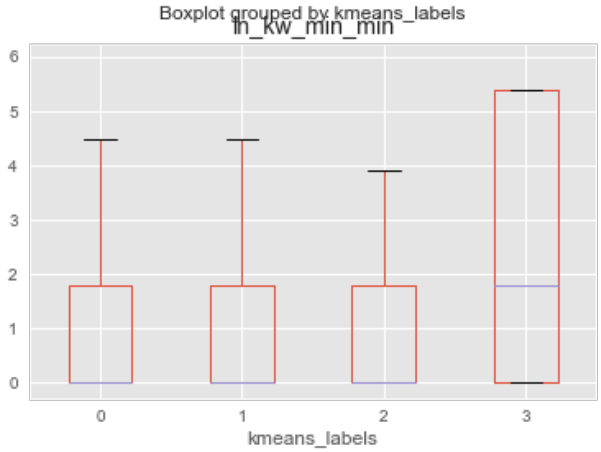
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e3424e5c0>

<matplotlib.figure.Figure at 0x7f8e2c2304a8>



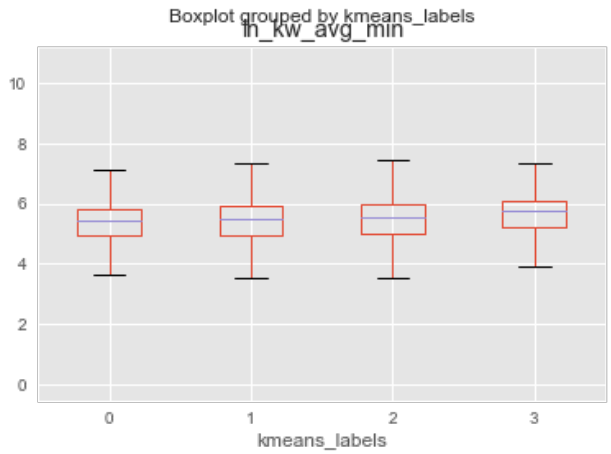
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e26f01b00>

<matplotlib.figure.Figure at 0x7f8e35e8a160>



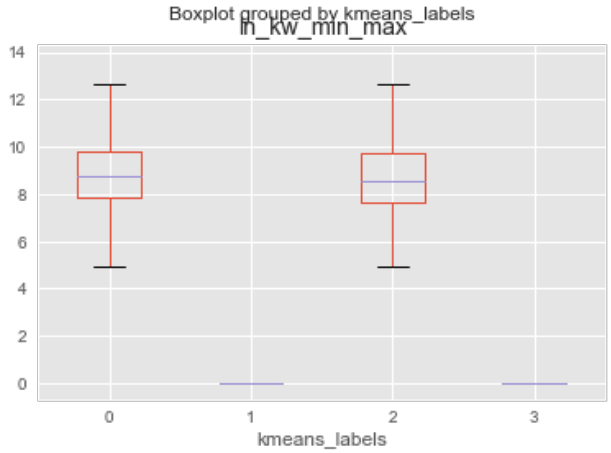
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e26eca0b8>

<matplotlib.figure.Figure at 0x7f8e3541bb38>



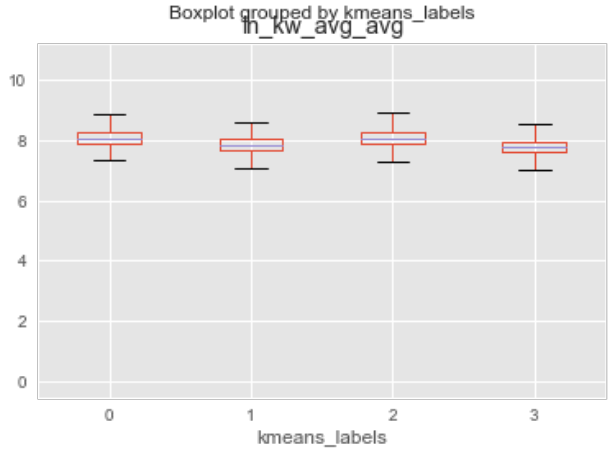
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e26e9fb00>

```
<matplotlib.figure.Figure at 0x7f8e26f01908>
```



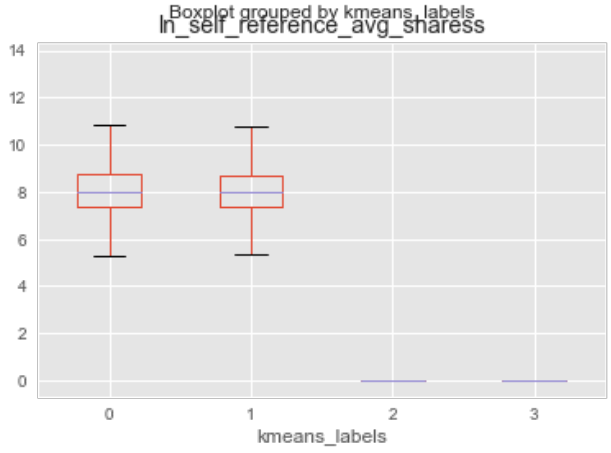
Out[38]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e26f60898>
```

```
<matplotlib.figure.Figure at 0x7f8e26e38f98>
```



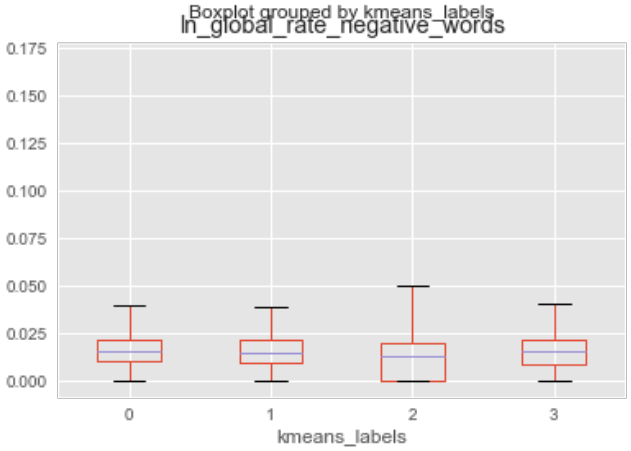
Out[38]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e340b4b38>
```

```
<matplotlib.figure.Figure at 0x7f8e26dcd7b8>
```



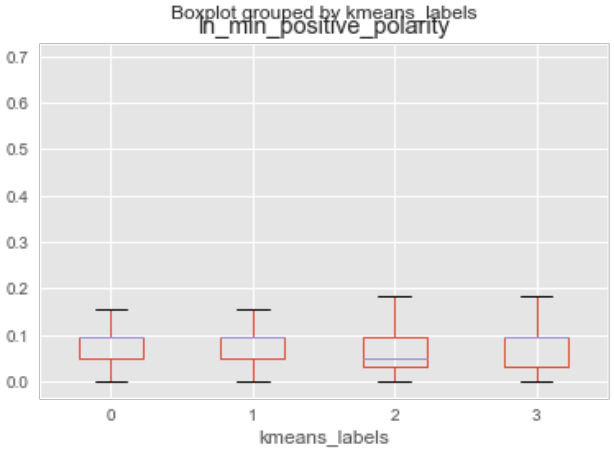
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e359ab898>

<matplotlib.figure.Figure at 0x7f8e2c6474e0>



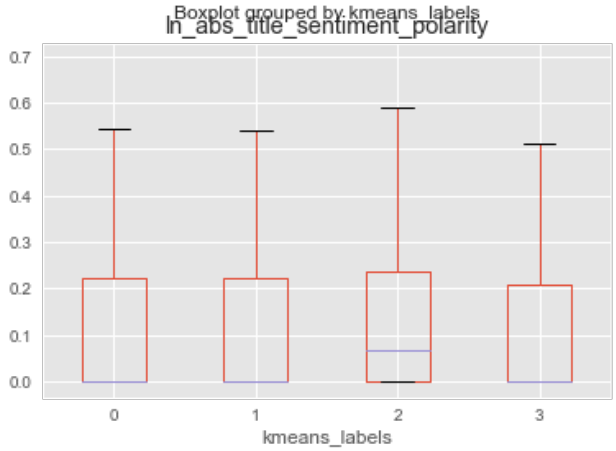
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e3411ab00>

<matplotlib.figure.Figure at 0x7f8e26de1198>



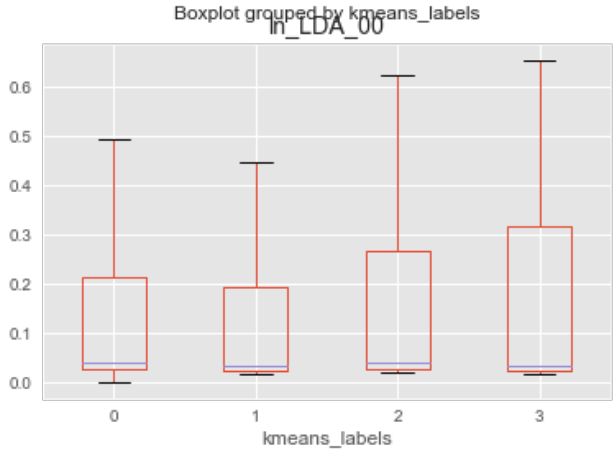
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e359de7f0>

<matplotlib.figure.Figure at 0x7f8e26dc6d68>



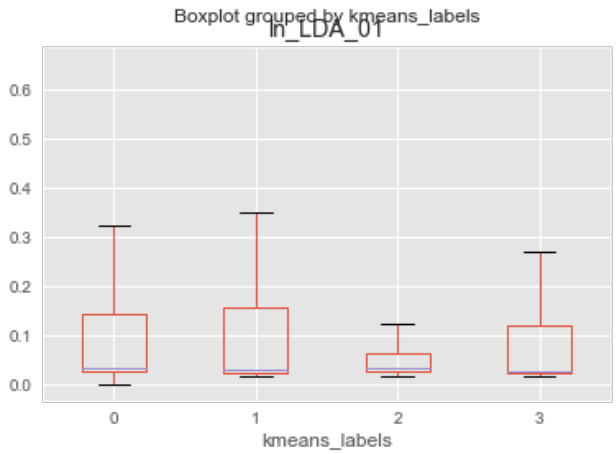
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e26e47b00>

<matplotlib.figure.Figure at 0x7f8e26e8a400>



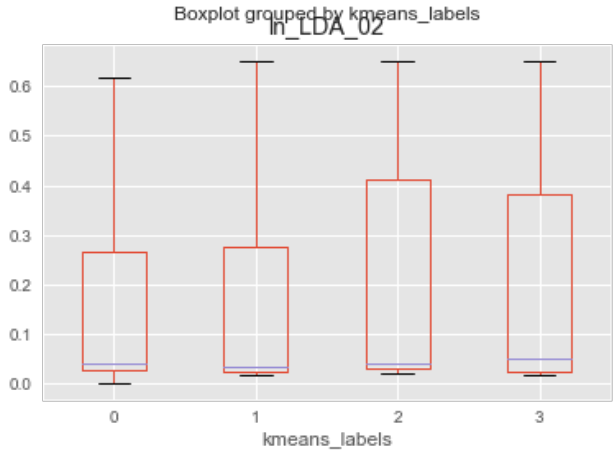
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c0d6fd0>

<matplotlib.figure.Figure at 0x7f8e2c00bf98>



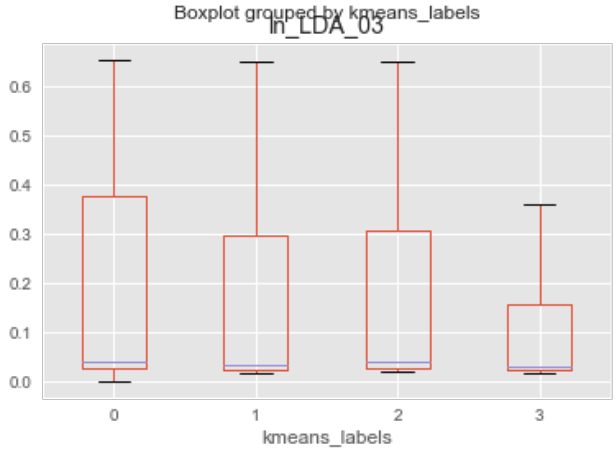
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c22ff60>

```
<matplotlib.figure.Figure at 0x7f8e26eb55c0>
```



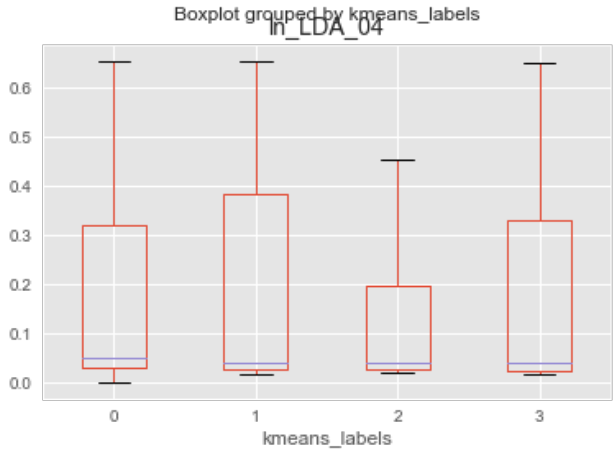
Out[38]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c2d5898>
```

```
<matplotlib.figure.Figure at 0x7f8e35a70b70>
```



Out[38]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8e26eca6d8>
```

```
<matplotlib.figure.Figure at 0x7f8e2c0829e8>
```



Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8e2c501978>

<matplotlib.figure.Figure at 0x7f8e26eb2ac8>

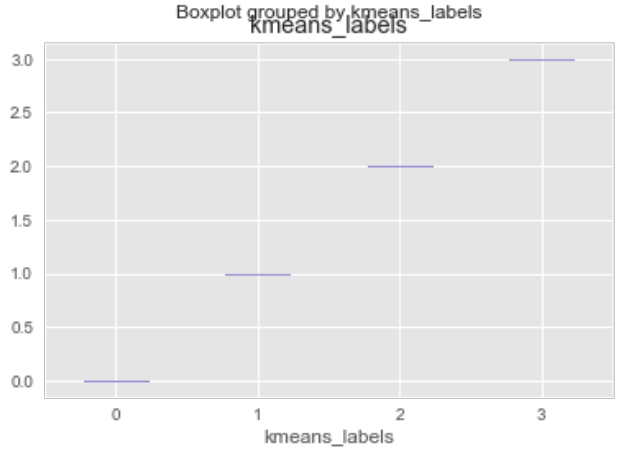


Table of Contents

DBScan

DBSCAN

http://hdbscan.readthedocs.io/en/latest/comparing_clustering_algorithms.html (http://hdbscan.readthedocs.io/en/latest/comparing_clustering_algorithms.html)

DBSCAN is a density based algorithm – it assumes clusters for dense regions. It is also the first actual clustering algorithm we've looked at: it doesn't require that every point be assigned to a cluster and hence doesn't partition the data, but instead extracts the 'dense' clusters and leaves sparse background classified as 'noise'.
In practice DBSCAN is related to agglomerative clustering.
As a first step DBSCAN transforms the space according to the density of the data: points in dense regions are left alone, while points in sparse regions are moved further away. Applying single linkage clustering to the transformed space results in a dendrogram, which we cut according to a distance parameter (called epsilon or eps in many implementations) to get clusters. Importantly any singleton clusters at that cut level are deemed to be 'noise' and left unclustered. This provides several advantages: we get the manifold following behaviour of agglomerative clustering, and we get actual clustering as opposed to partitioning. Better yet, since we can frame the algorithm in terms of local region queries we can use various tricks such as kdtrees to get exceptionally good performance and scale to dataset sizes that are otherwise unapproachable with algorithms other than K-Means.
There are some catches however. Obviously epsilon can be hard to pick; you can do some data analysis and get a good guess, but the algorithm can be quite sensitive to the choice of the parameter. The density based transformation depends on another parameter (min_samples in sklearn).
Finally the combination of min_samples and eps amounts to a choice of density and the clustering only finds clusters at or above that density; if your data has variable density clusters then DBSCAN is either going to miss them, split them up, or lump some of them together depending on your parameter choices.

So, in summary:

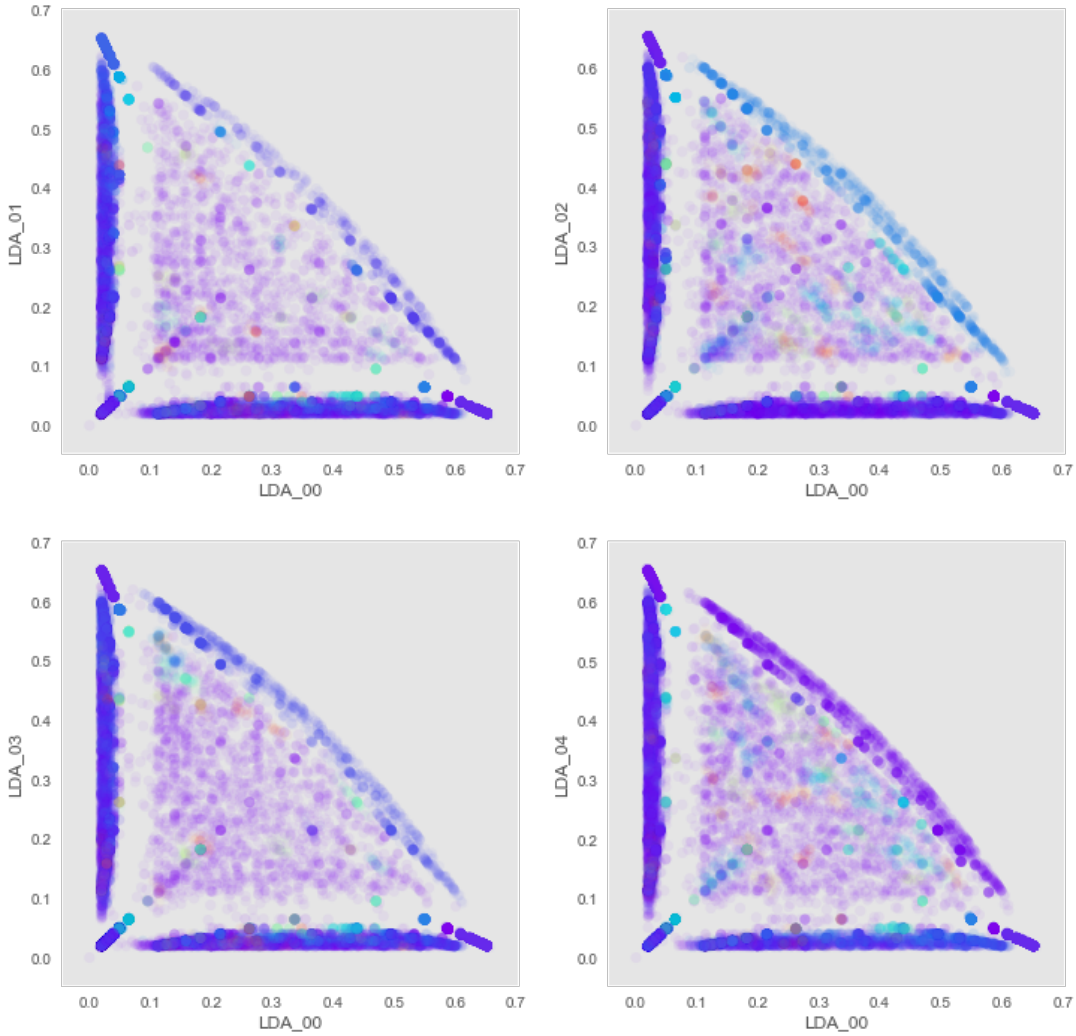
- **Don't be wrong!**: Clusters don't need to be globular, and won't have noise lumped in; varying density clusters may cause problems, but that is more in the form of insufficient detail rather than explicitly wrong. DBSCAN is the first clustering algorithm we've looked at that actually meets the 'Don't be wrong!' requirement.
  - **Intuitive parameters**: Epsilon is a distance value, so you can survey the distribution of distances in your dataset to attempt to get an idea of where it should lie. In practice, however, this isn't an especially intuitive parameter, nor is it easy to get right.
  - **Stability**: DBSCAN is stable across runs (and to some extent subsampling if you re-parameterize well); stability over varying epsilon and min samples is not so good.
  - **Performance**: This is DBSCAN's other great strength; few clustering algorithms can tackle datasets as large as DBSCAN can.

So how does it cluster our test dataset? I played with a few epsilon values until I got somethign reasonable, but there was little science to this – getting the parameters right can be hard.

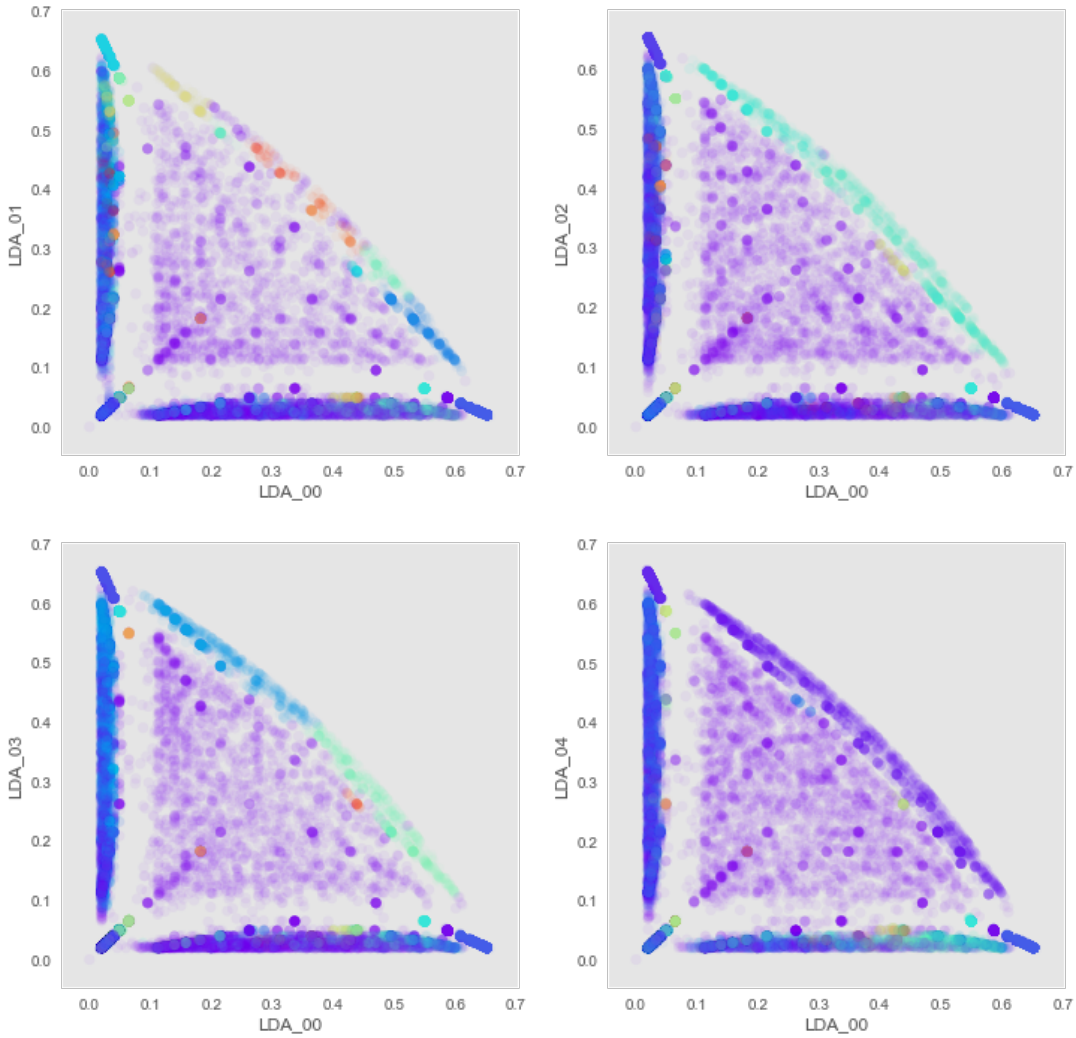
In [61]:
```
# set required variables for model comparison

dbscan_tbl = pd.DataFrame(columns = [
    'model_name',
    'n_clusters',
    'epsilon',
    'min_points',
    'inertia',
    'silhouette',
    'process_time'])

i_index = []
i_index = 0

# preparation for cross validation and model comparison, each classifier is appended once mode
l is fit

models = []
```

In [62]:
```
%%time

from sklearn.cluster import DBSCAN

params = []
for epsilon in [0.020, 0.03, 0.05, 0.06, 0.07]:
    for min_pts in range (10, 200, 20):

        tic = time.clock()

        X1 = df_cluster[['ln_LDA_00','ln_LDA_01', 'ln_LDA_02', 'ln_LDA_03', 'ln_LDA_04']]

        # append on the clustering

        cls_fam = DBSCAN(eps = epsilon,
                         min_samples = min_pts,
                         n_jobs = -1)
        cls_fam.fit(X1)

        dbscan_labels = cls_fam.labels_ # the labels from kmeans clustering

        dbscan_nclusters = len(set(dbscan_labels))

        print ("eps, min_pts, nclusters = ", epsilon, min_pts, dbscan_nclusters)

        dbscan_silhouette = metrics.silhouette_score(X1,
                                                     dbscan_labels,
                                                     metric = 'euclidean',
                                                     sample_size = 10000)
        print ("silhouette = ", dbscan_silhouette)

        toc =  time.clock()

# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ... - save statistics for model comparison
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

        exe_time = '{0:.4f}'.format(toc-tic)

        raw_data = {
        'model_name' : 'DBScan - LDA features',
        'n_clusters' : dbscan_nclusters,
        'epsilon' : epsilon,
        'min_points' : min_pts,
        'inertia': 0,
        'silhouette': dbscan_silhouette,
        'process_time' : exe_time
        }

        df_tbl = pd.DataFrame(raw_data,
        columns = ['model_name', 'n_clusters', 'epsilon', 'min_points', 'inertia', 'silhouette
', 'process_time'],
        index = [i_index + 1])

        dbscan_tbl = dbscan_tbl.append(df_tbl)
```

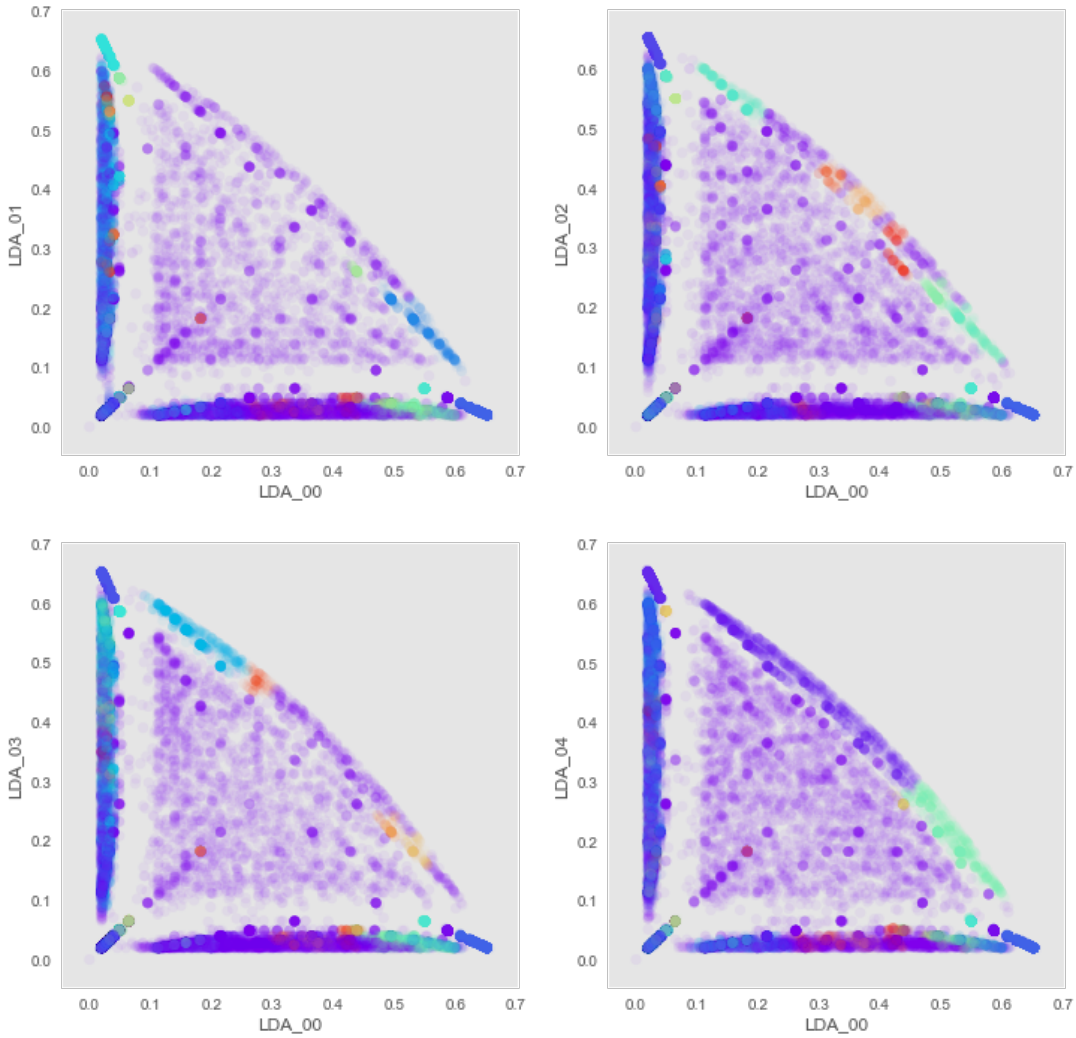
```
eps, min_pts, nclusters =  0.02 10 153
silhouette =  -0.0855960794612
```



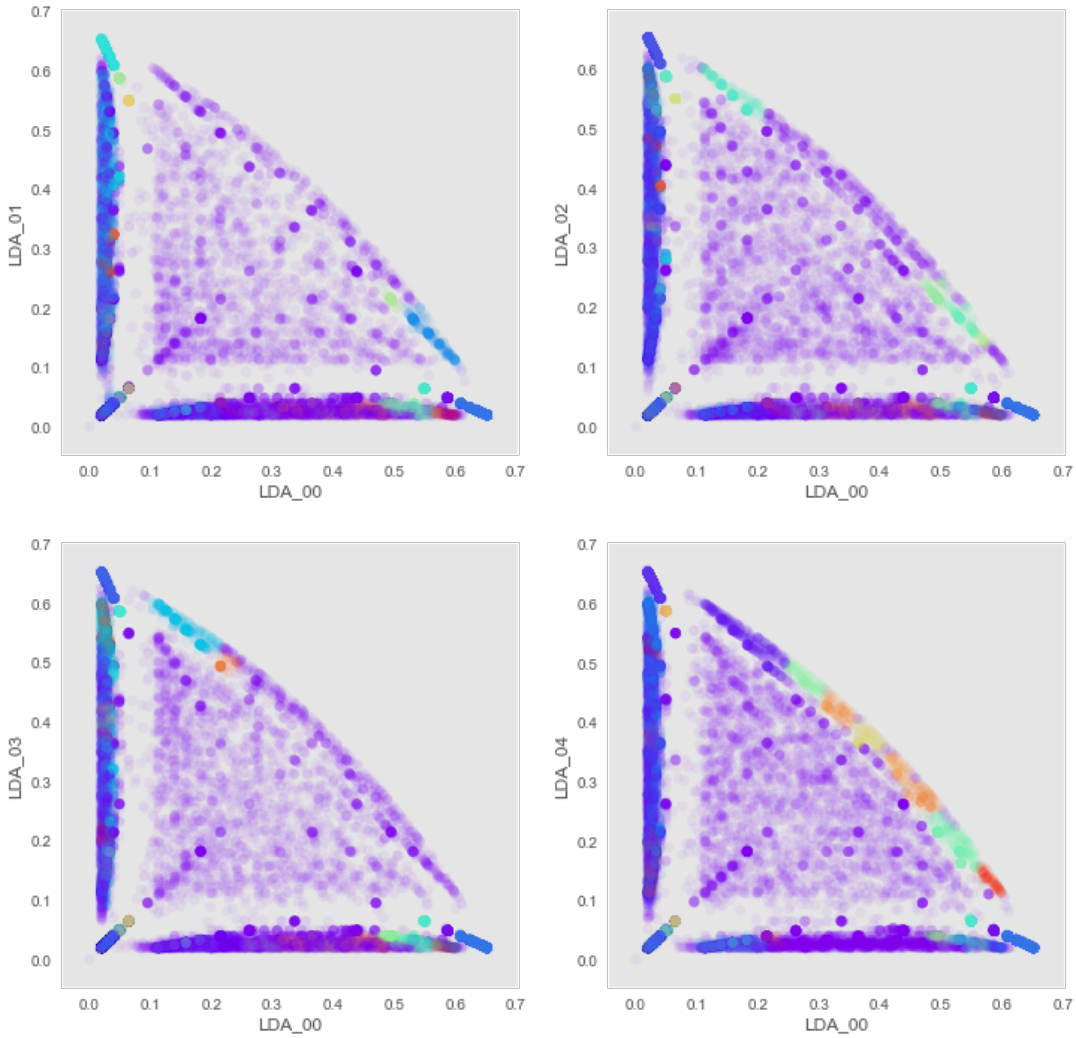

```
eps, min_pts, nclusters =  0.02 30 60
silhouette =  0.031905900258
```



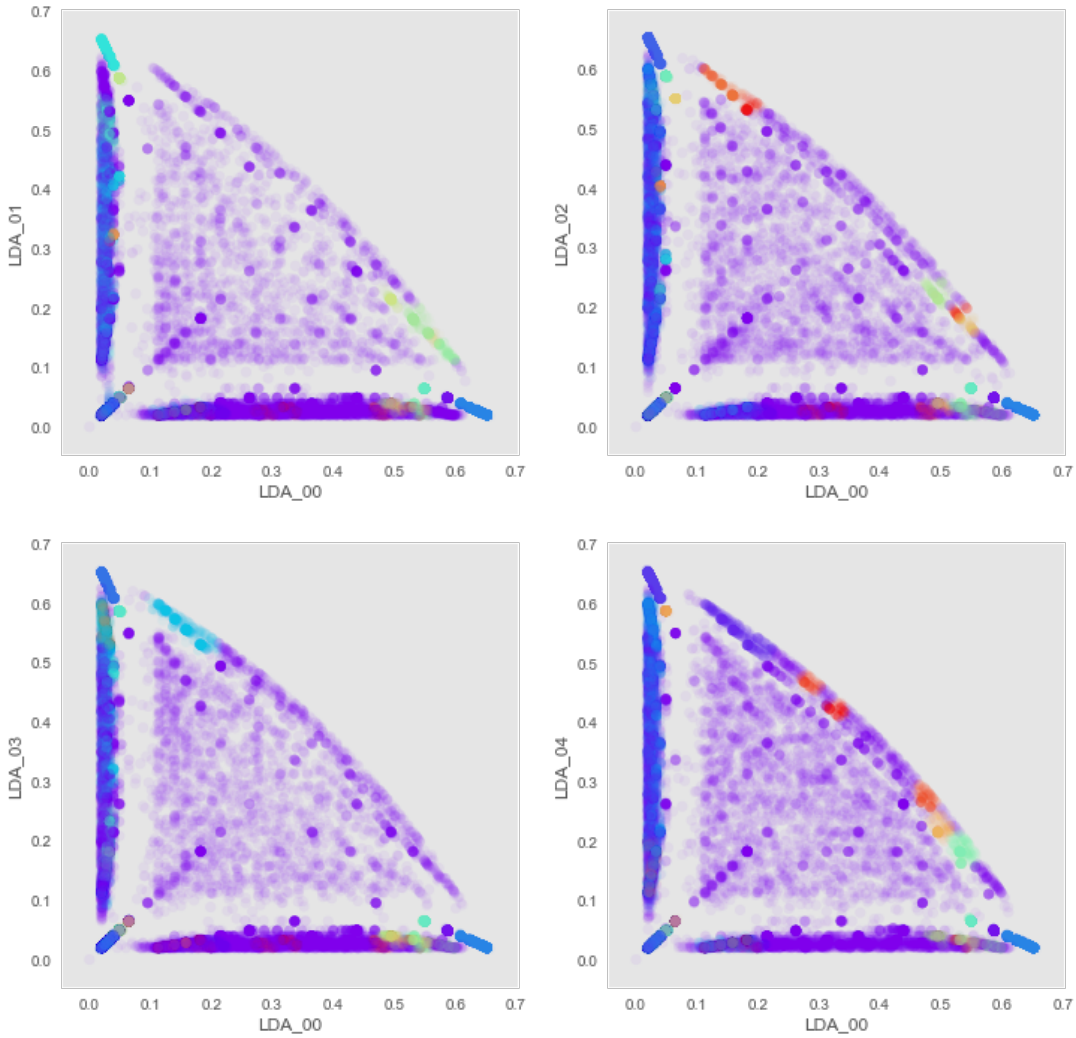

```
eps, min_pts, nclusters =  0.02 50 56
silhouette =  0.101426571856
```



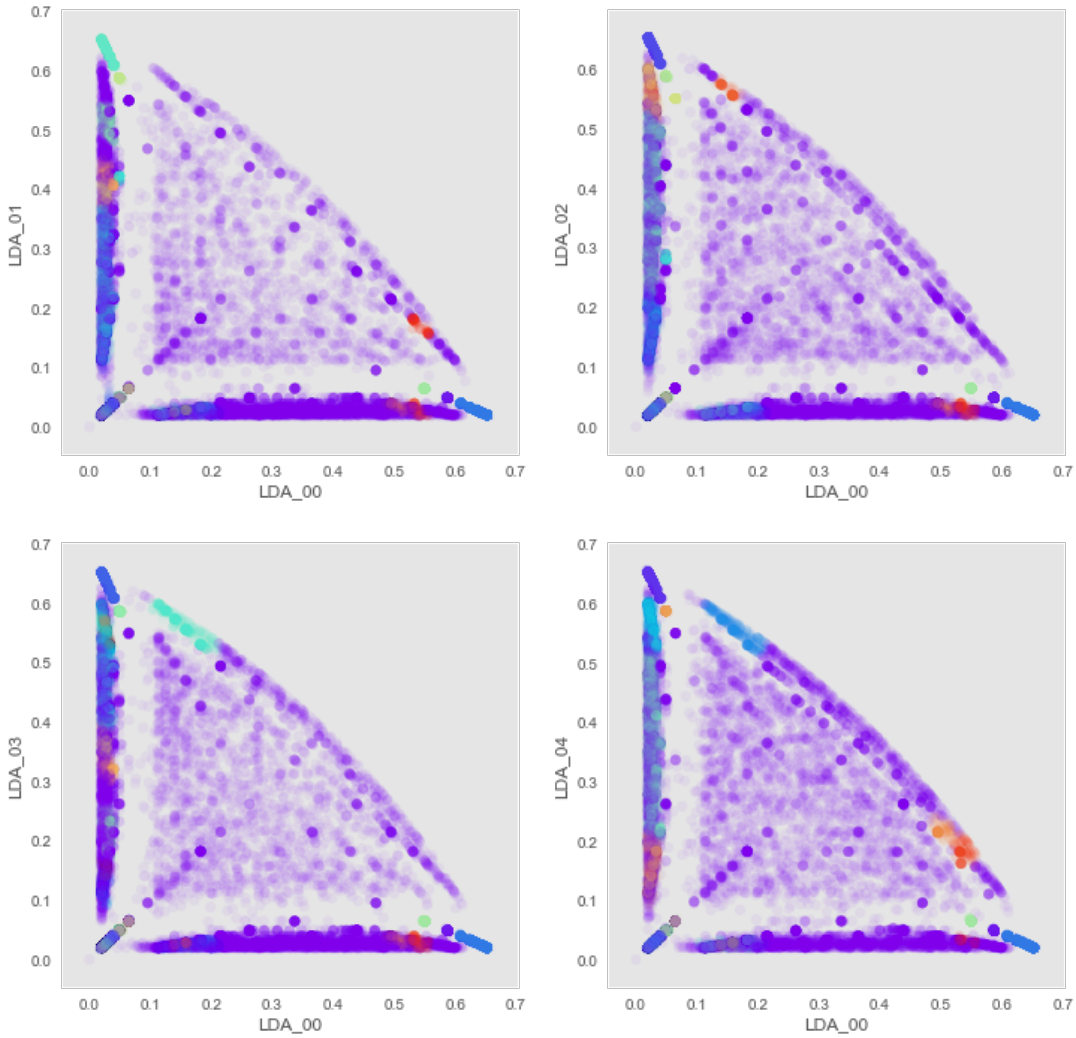

```
eps, min_pts, nclusters =  0.02 70 48
silhouette =  0.0771194097828
```



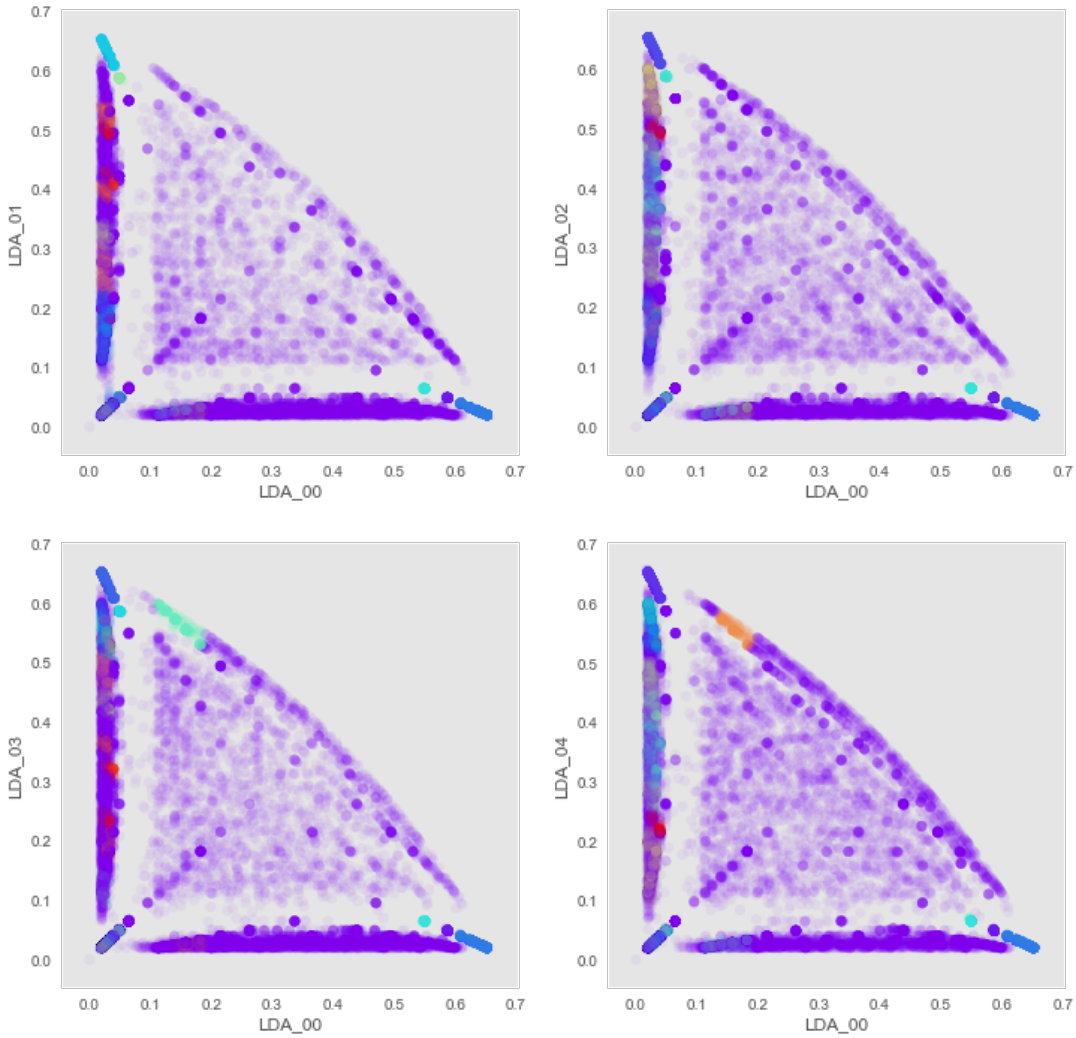

```
eps, min_pts, nclusters =  0.02 90 41
silhouette =  0.036829602996
```



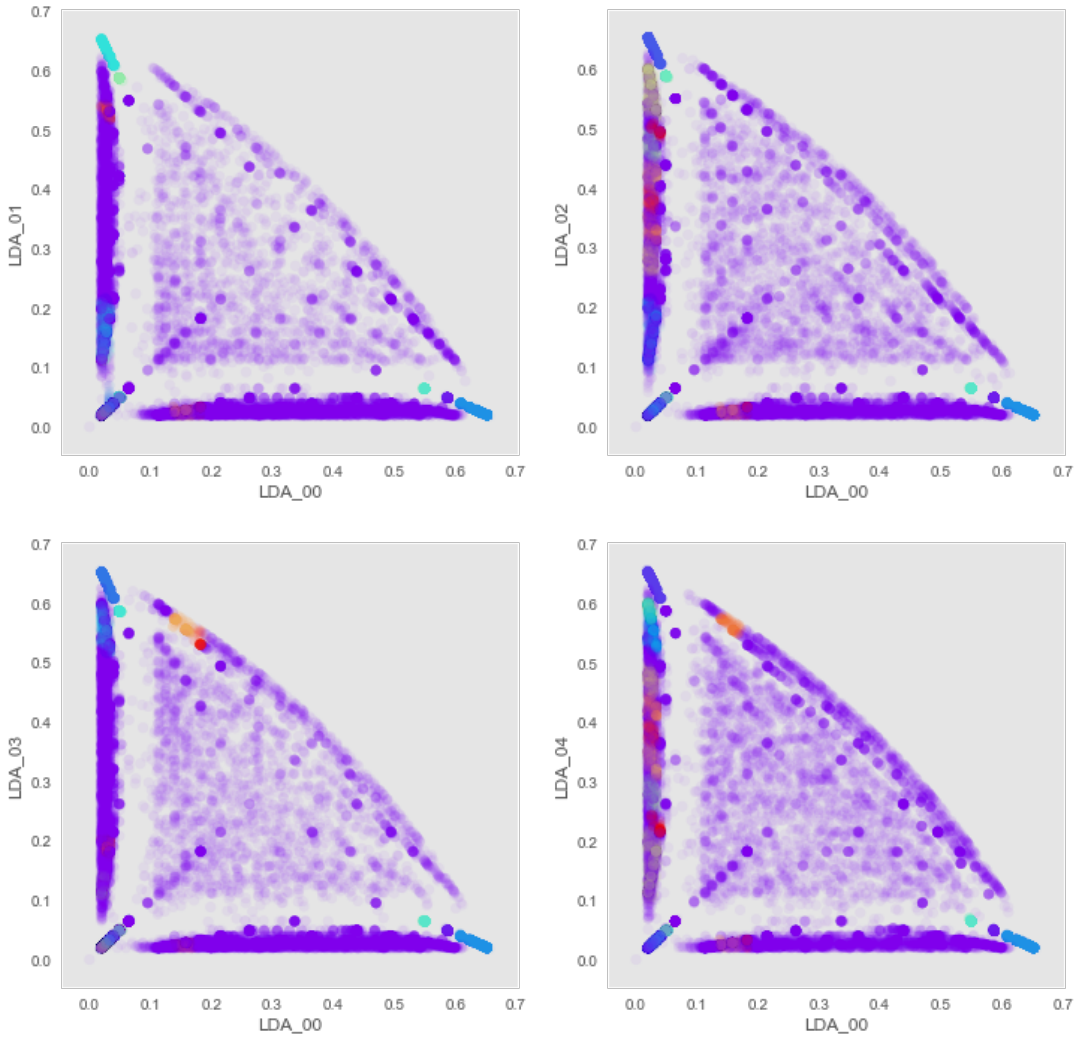

```
eps, min_pts, nclusters =  0.02 110 33
silhouette =  0.0431833243125
```



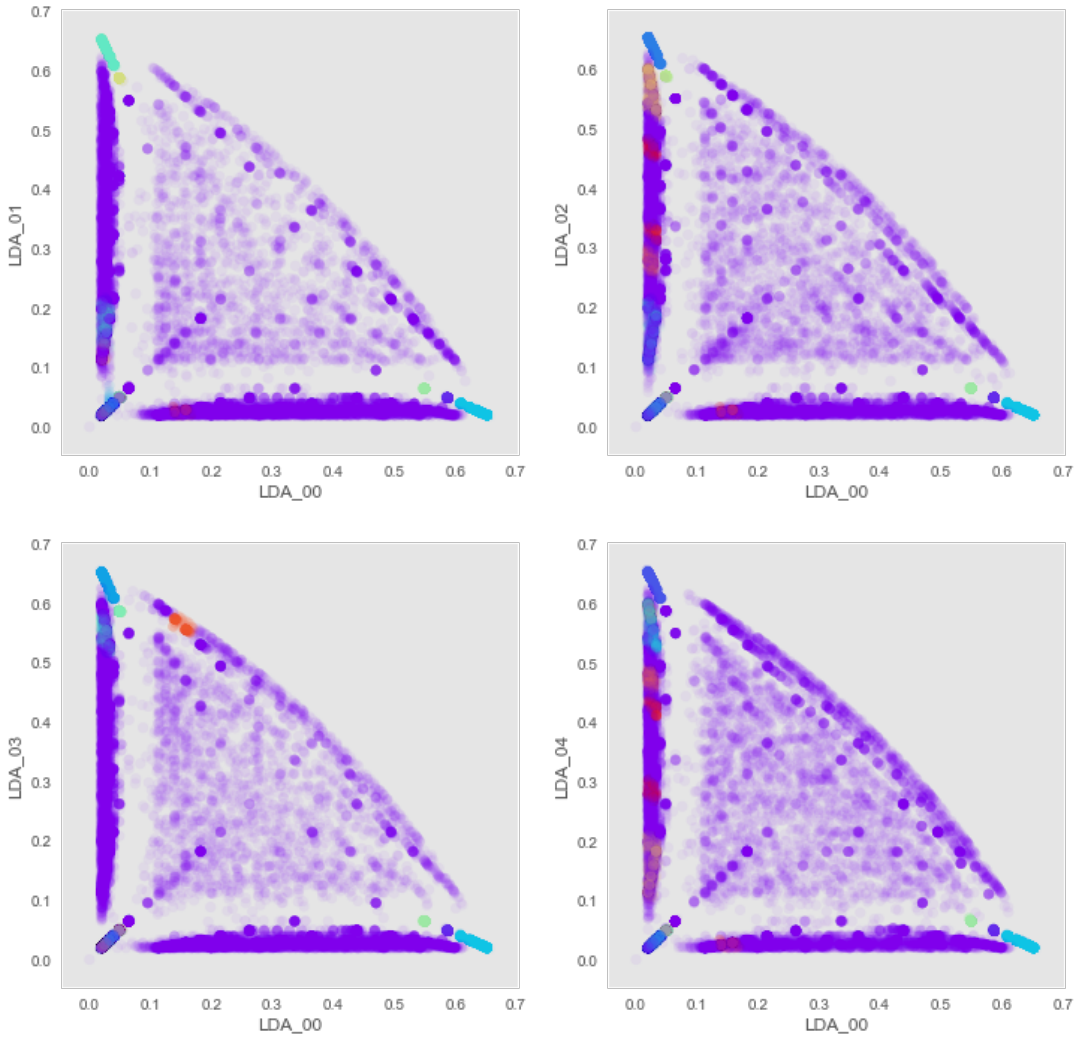

```
eps, min_pts, nclusters =  0.02 130 32
silhouette =  0.0250743431644
```



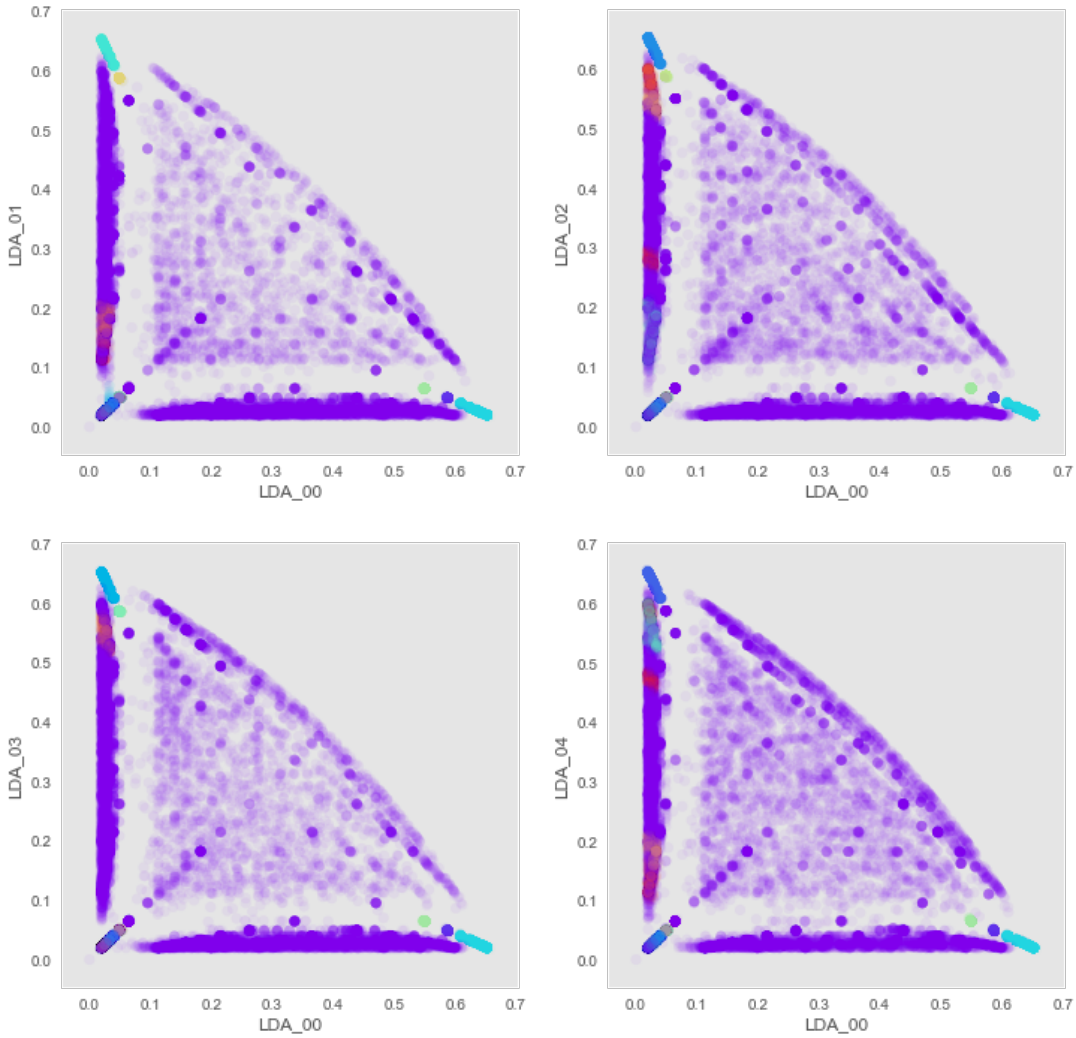

```
eps, min_pts, nclusters =  0.02 150 27
silhouette =  -0.012064911841
```



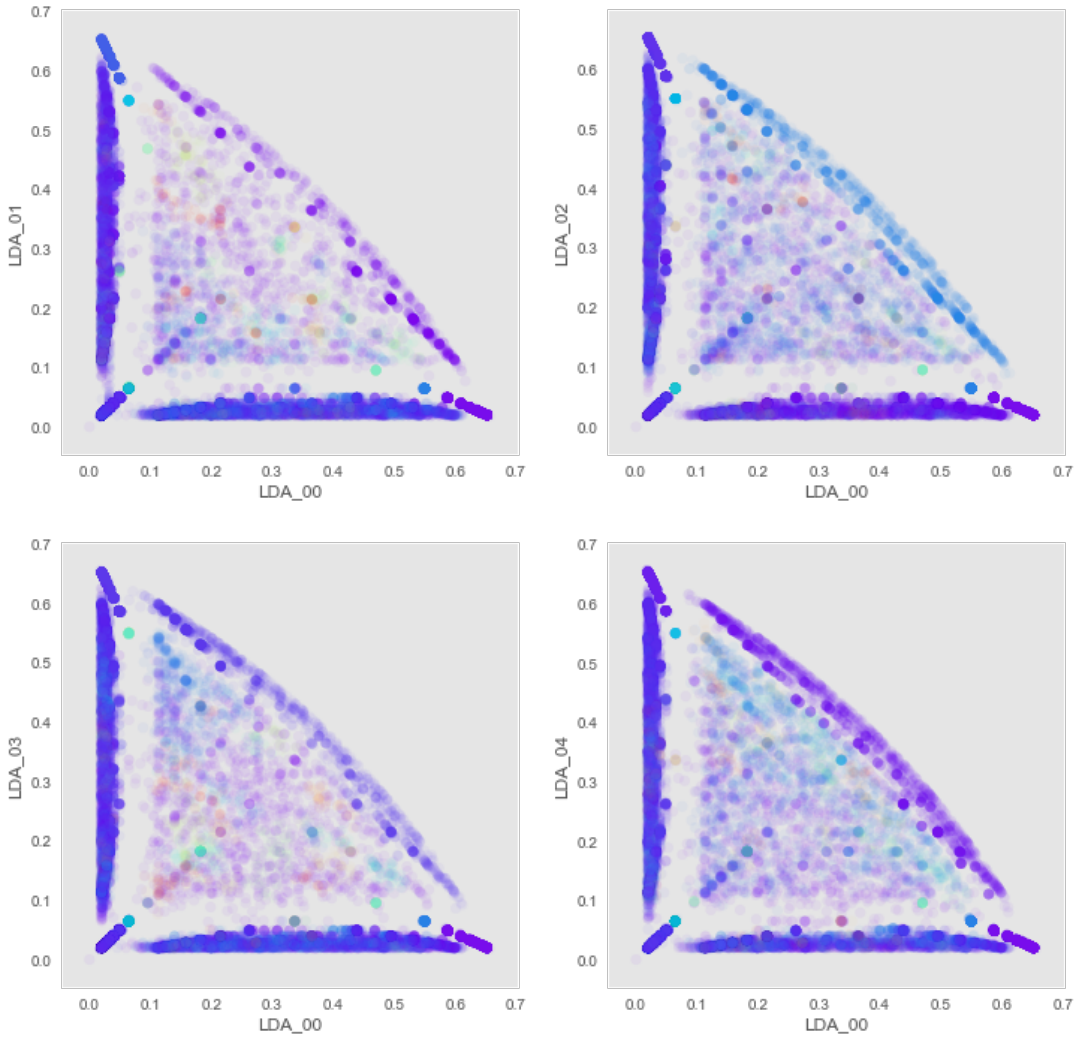

```
eps, min_pts, nclusters =  0.02 170 19
silhouette =  -0.0598985059903
```



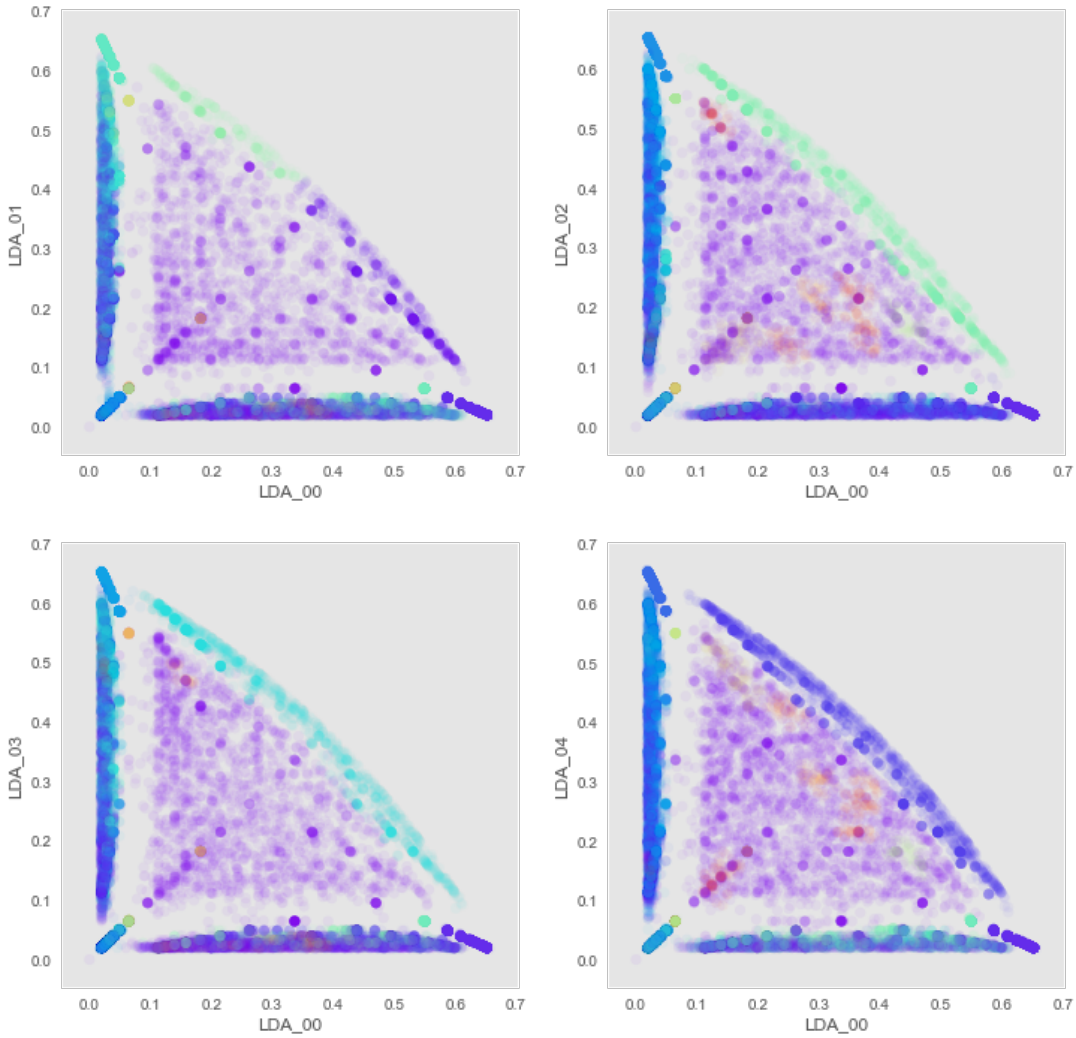

```
eps, min_pts, nclusters =  0.02 190 17
silhouette =  -0.0711635056725
```



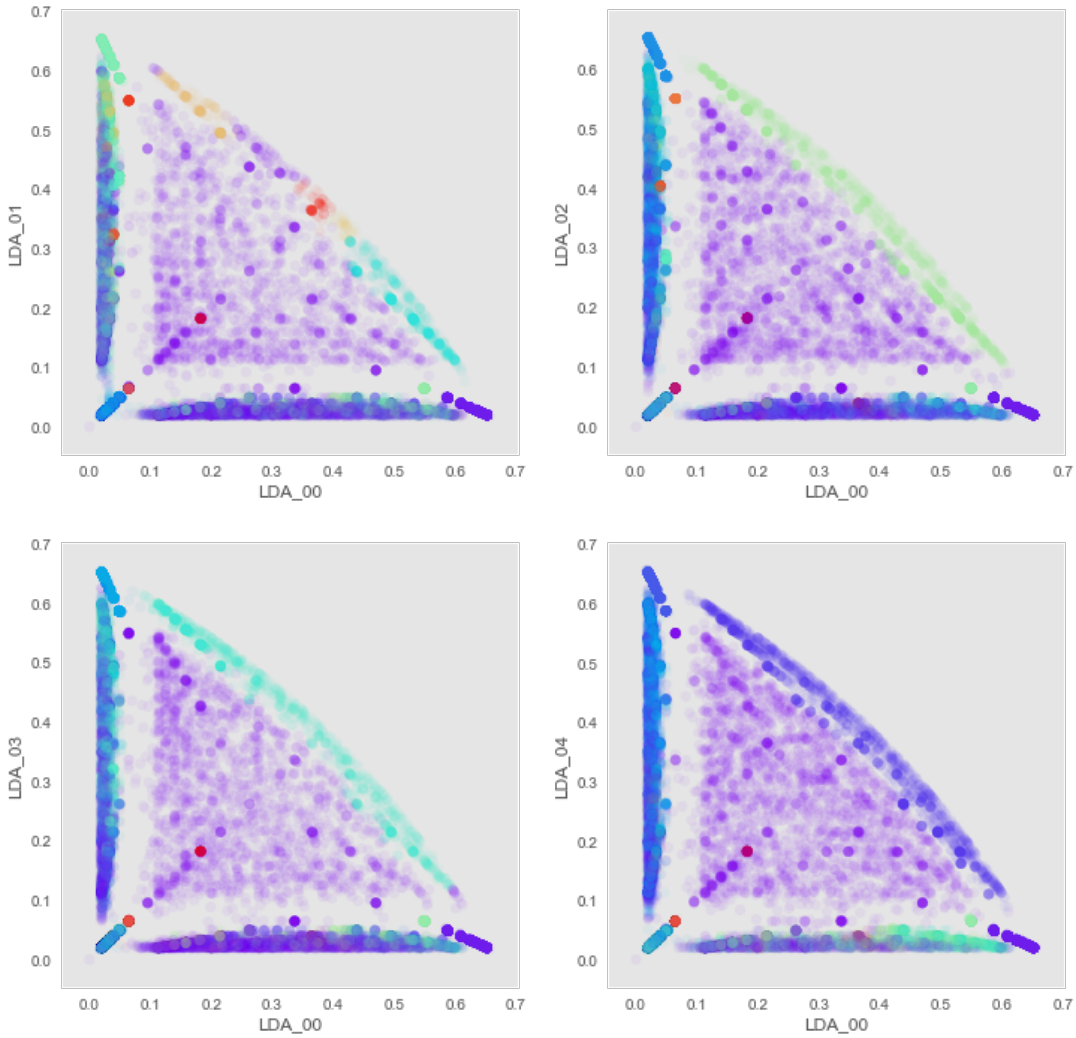

```
eps, min_pts, nclusters =  0.03 10 124
silhouette =  -0.210357563554
```



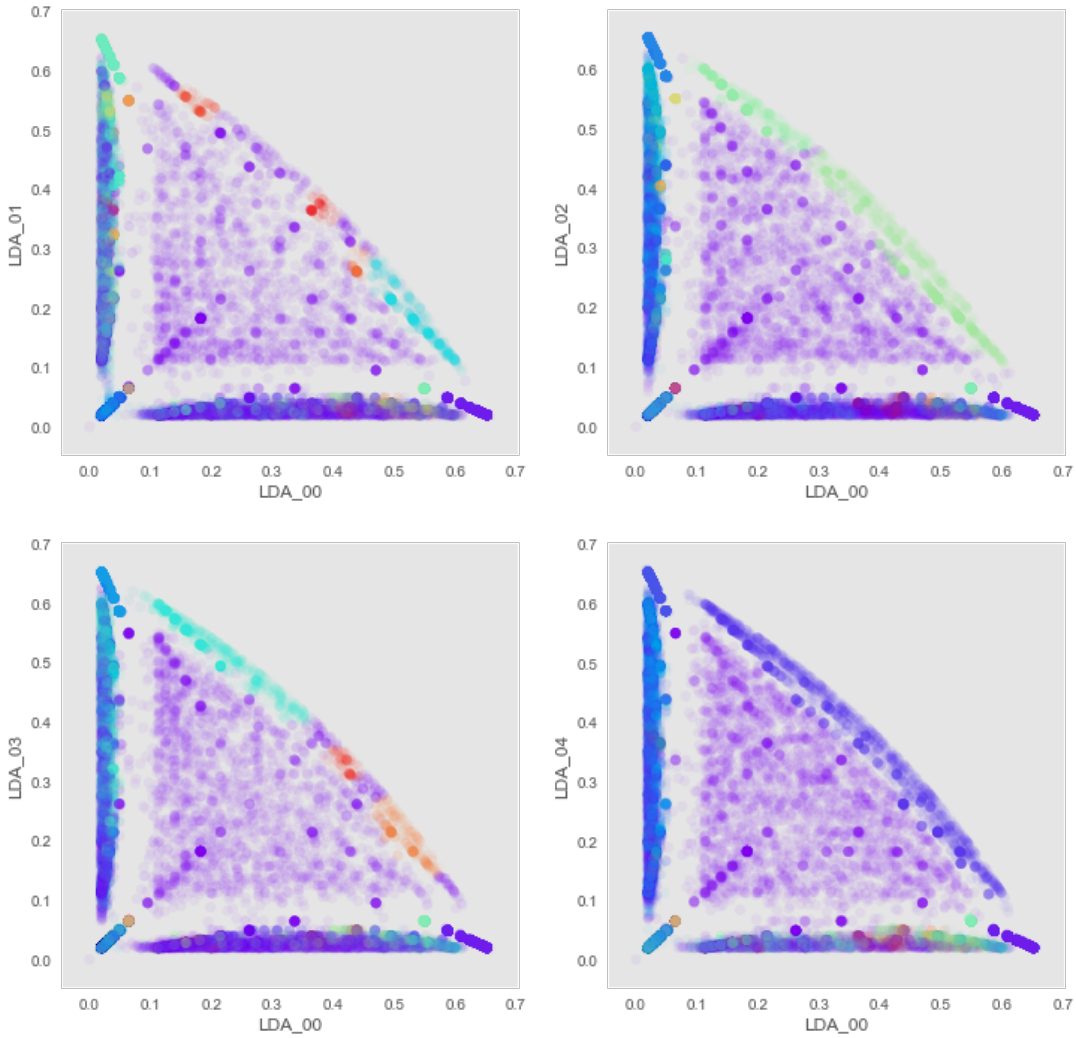

```
eps, min_pts, nclusters =  0.03 30 37
silhouette =  -0.0213734049662
```



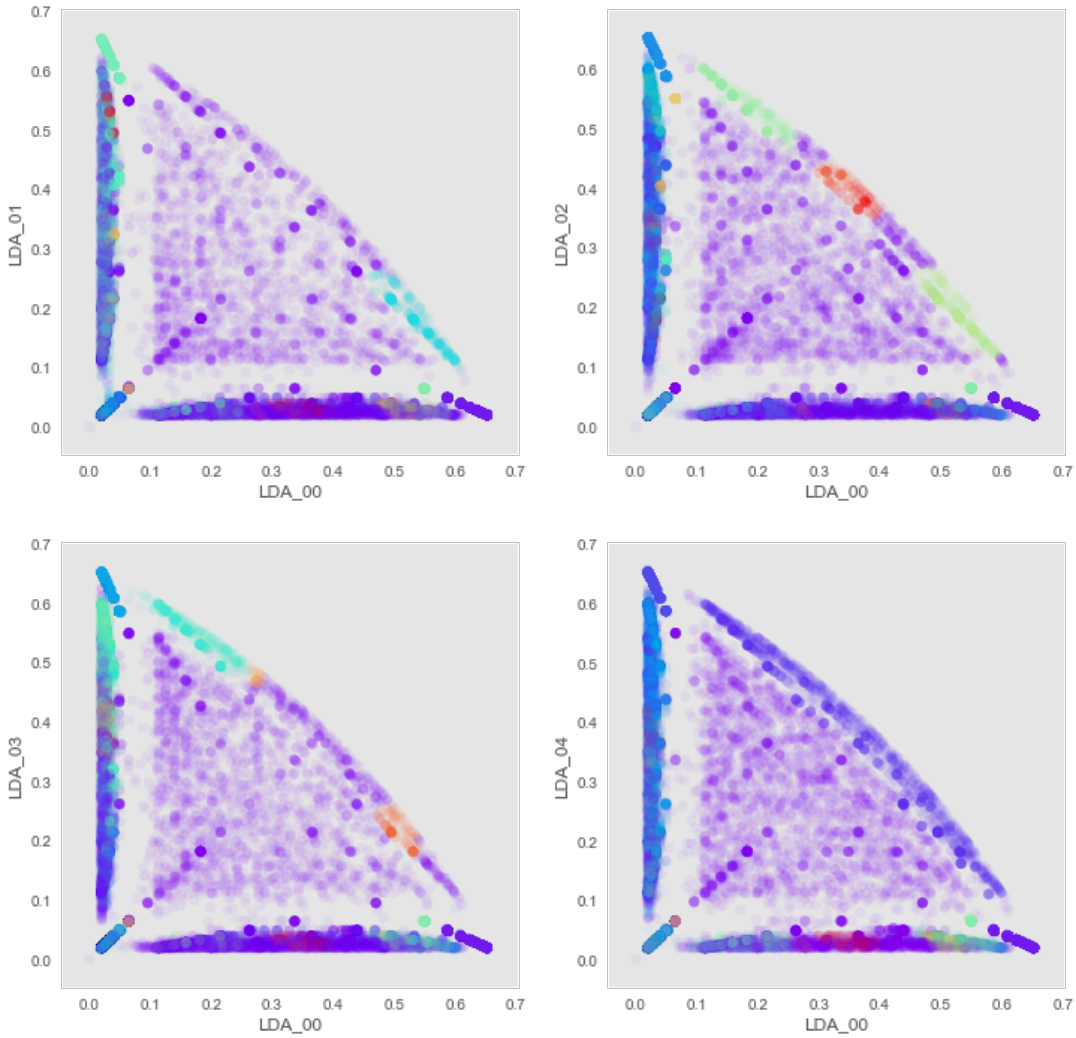

```
eps, min_pts, nclusters =  0.03 50 27
silhouette =  0.145458718953
```



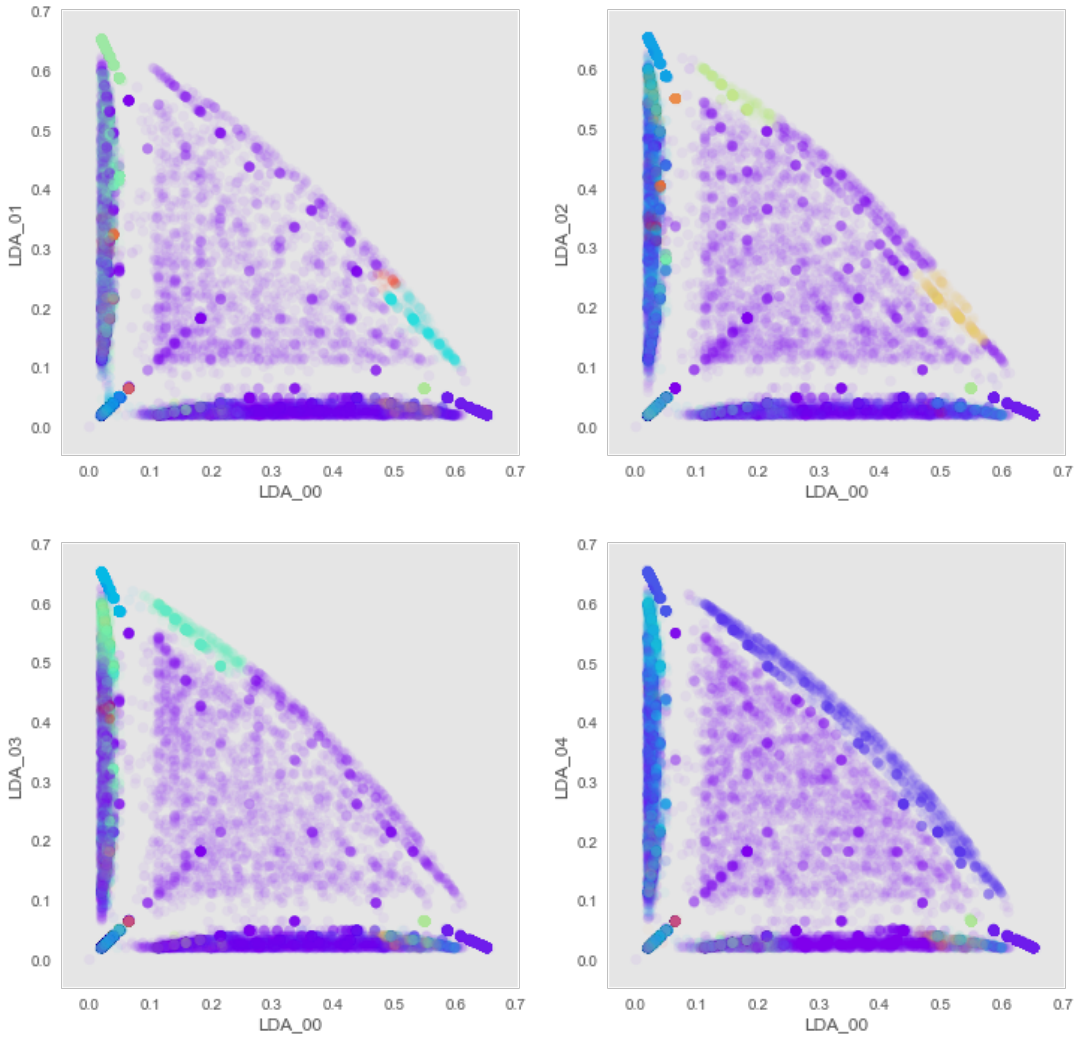

```
eps, min_pts, nclusters =  0.03 70 29
silhouette =  0.138460818714
```



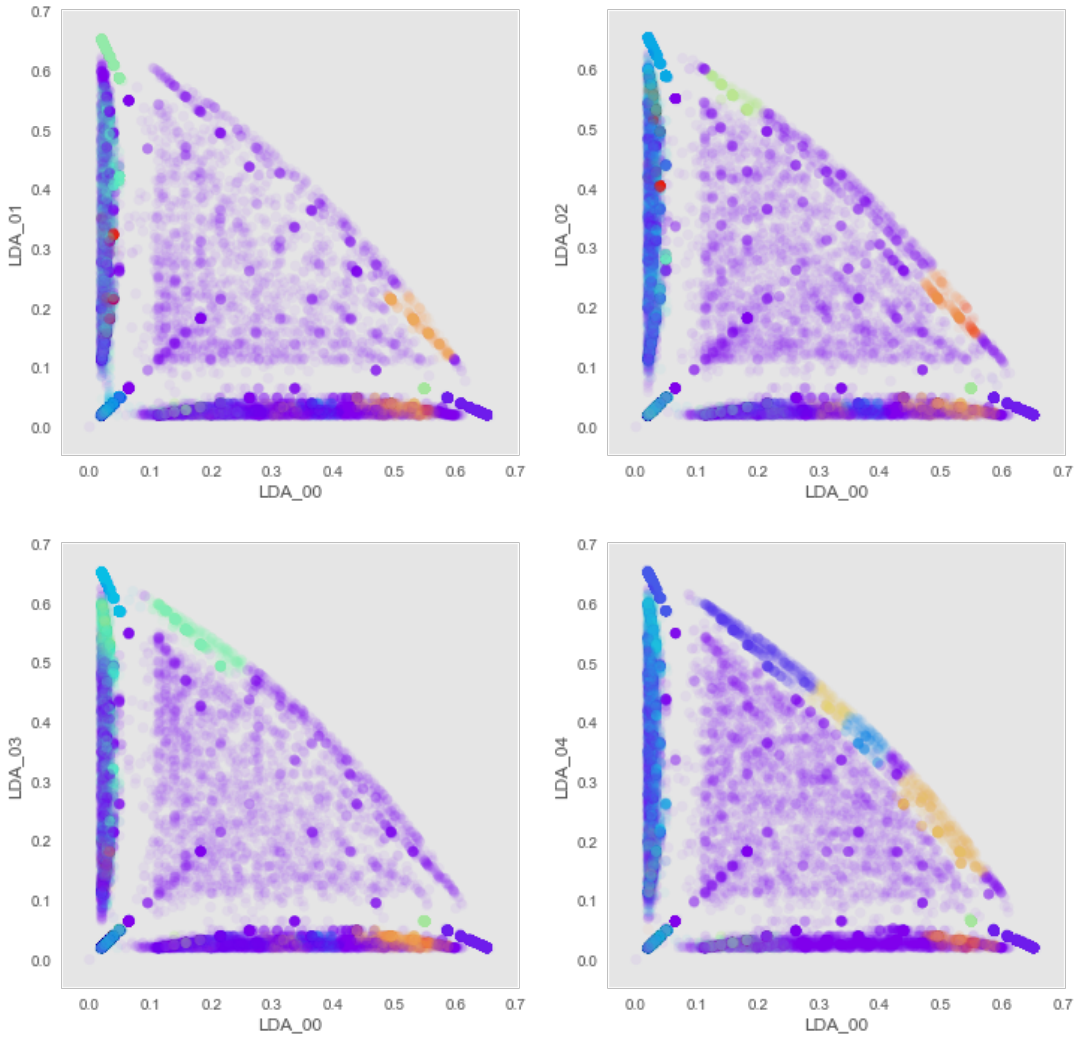

```
eps, min_pts, nclusters =  0.03 90 32
silhouette =  0.162128722293
```



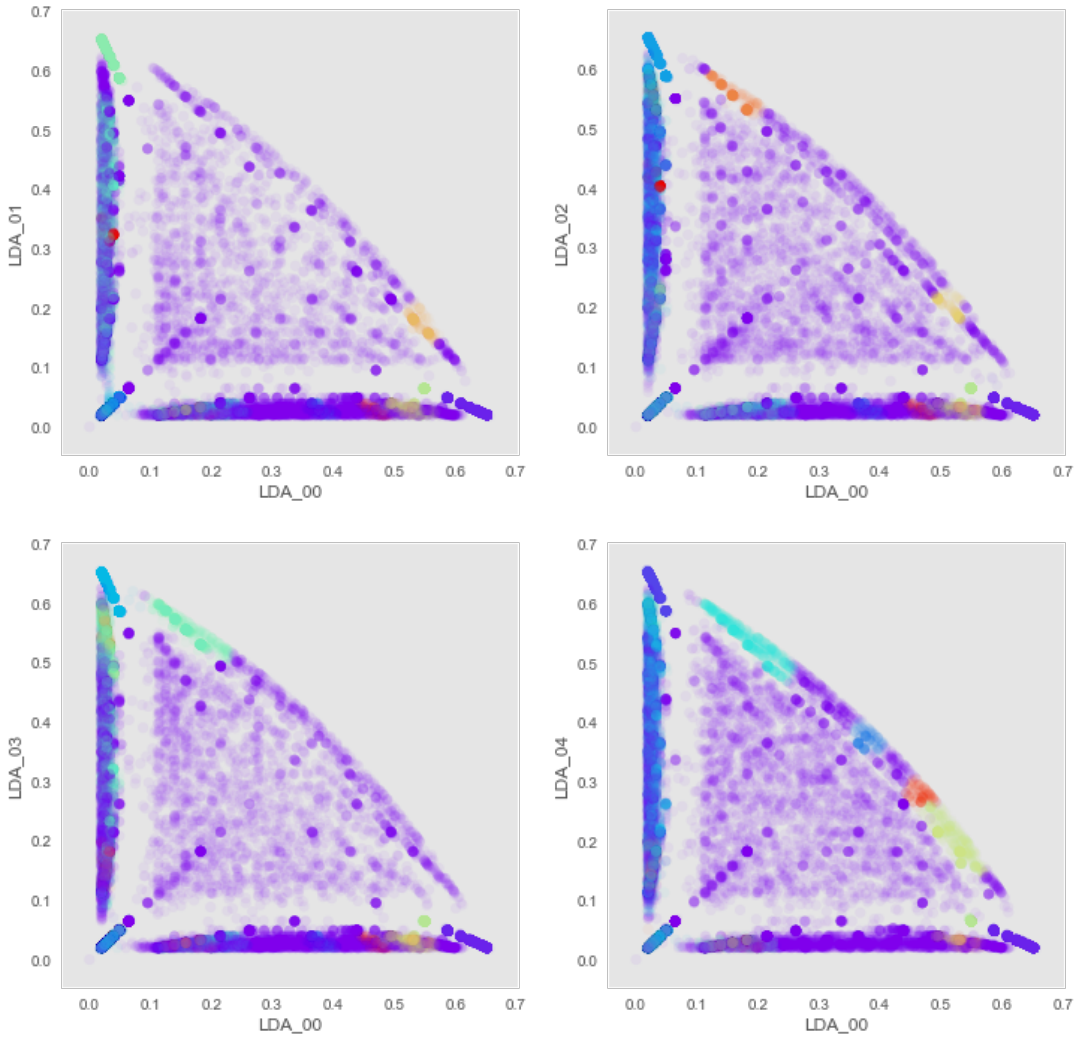

```
eps, min_pts, nclusters =  0.03 110 28
silhouette =  0.146650573747
```



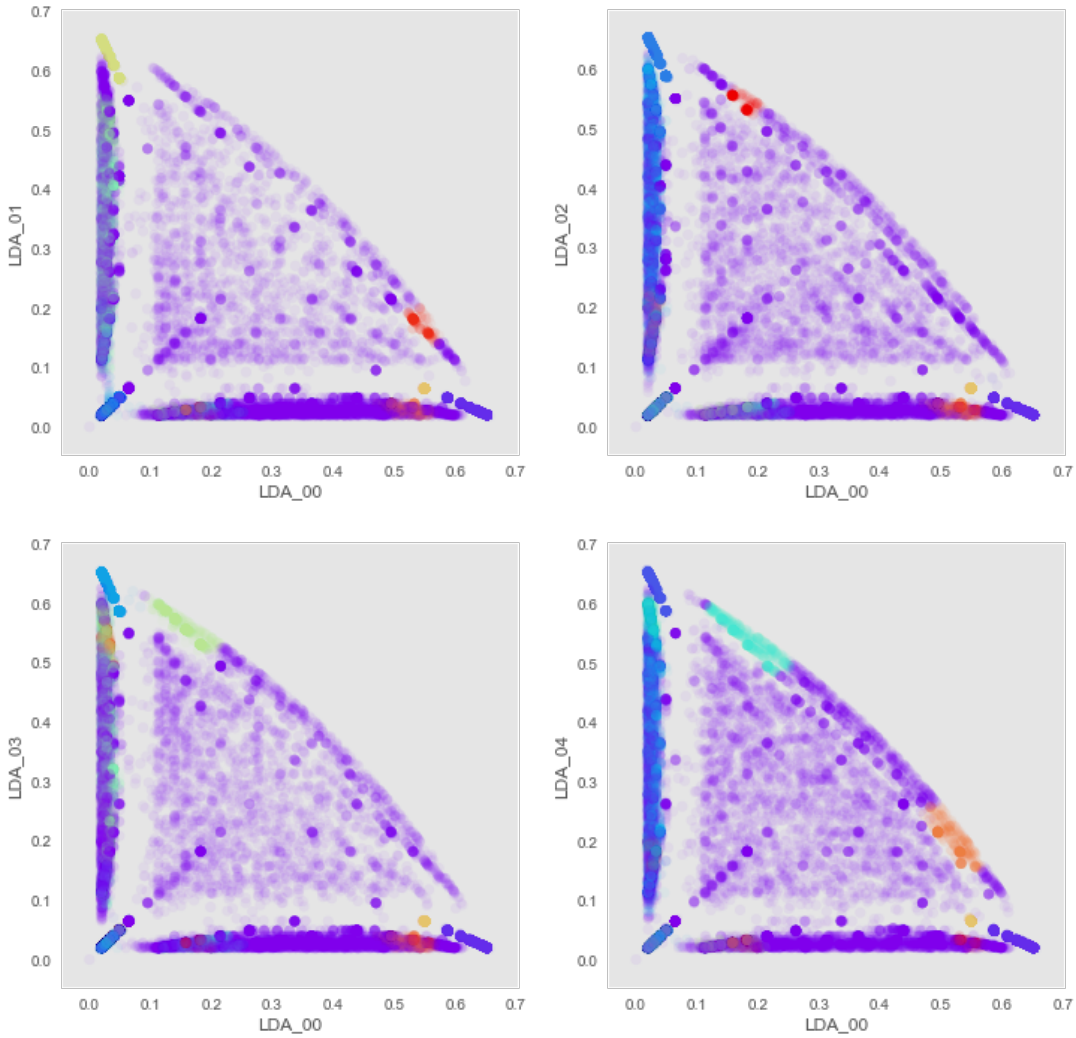

```
eps, min_pts, nclusters =  0.03 130 27
silhouette =  0.170528060286
```



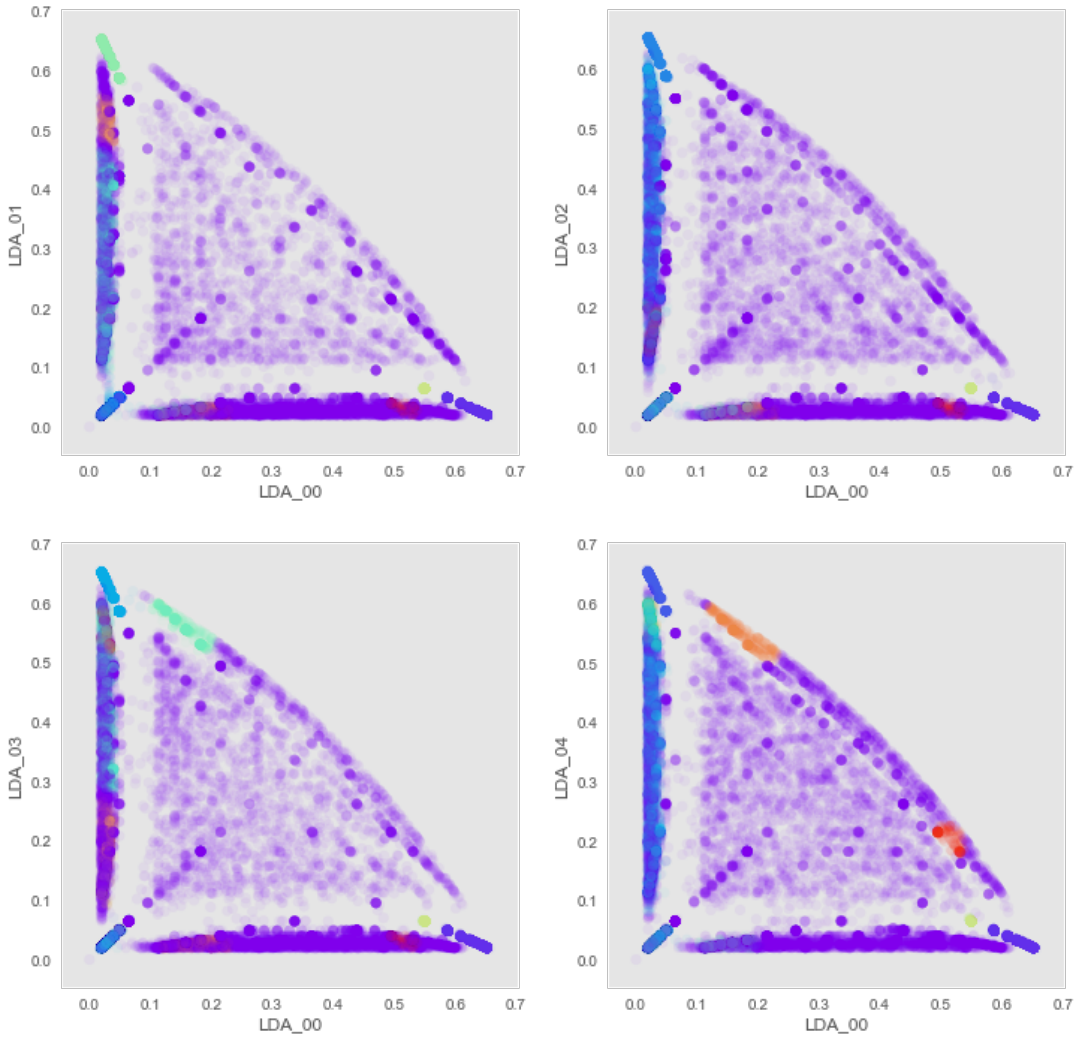

```
eps, min_pts, nclusters =  0.03 150 24
silhouette =  0.118947353013
```



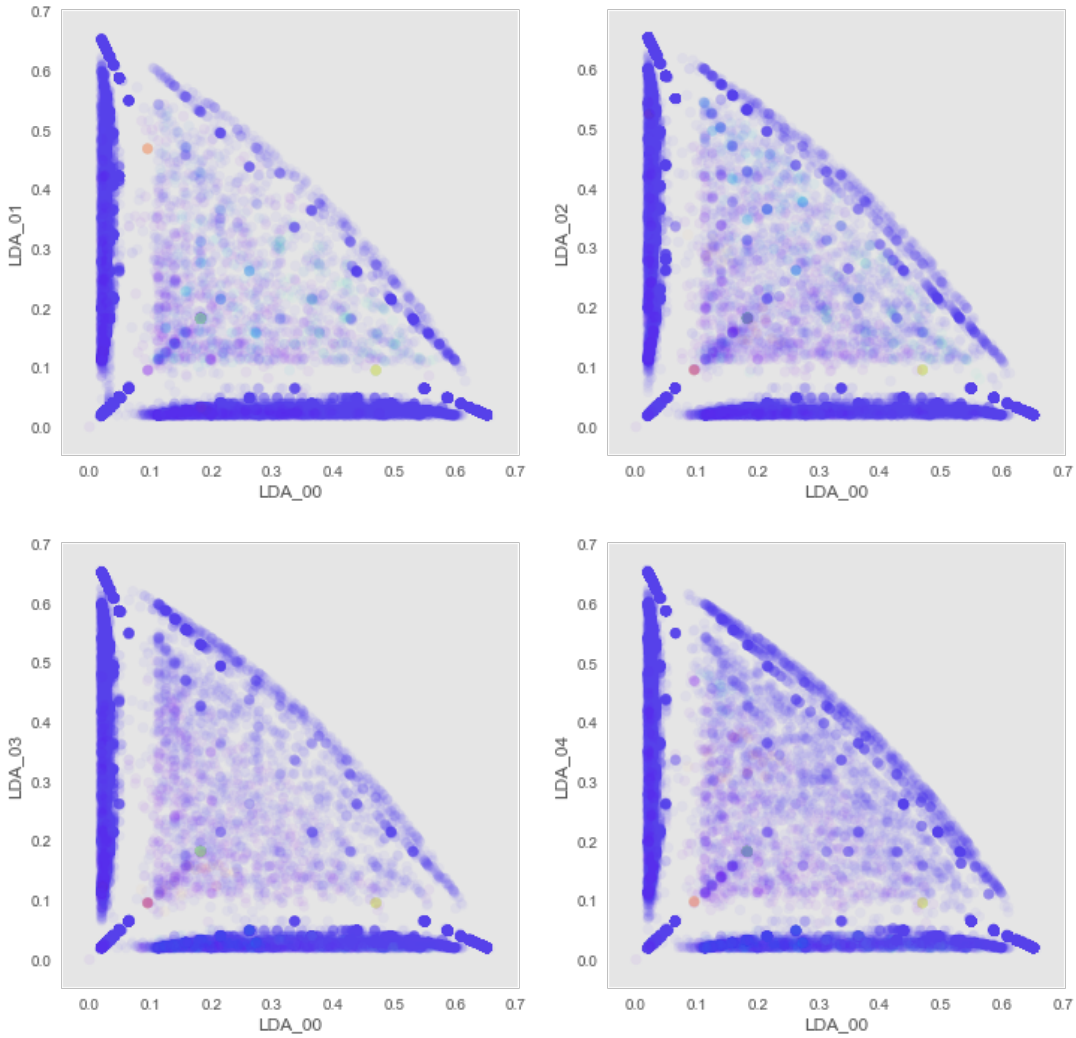

```
eps, min_pts, nclusters =  0.03 170 19
silhouette =  0.109054800548
```



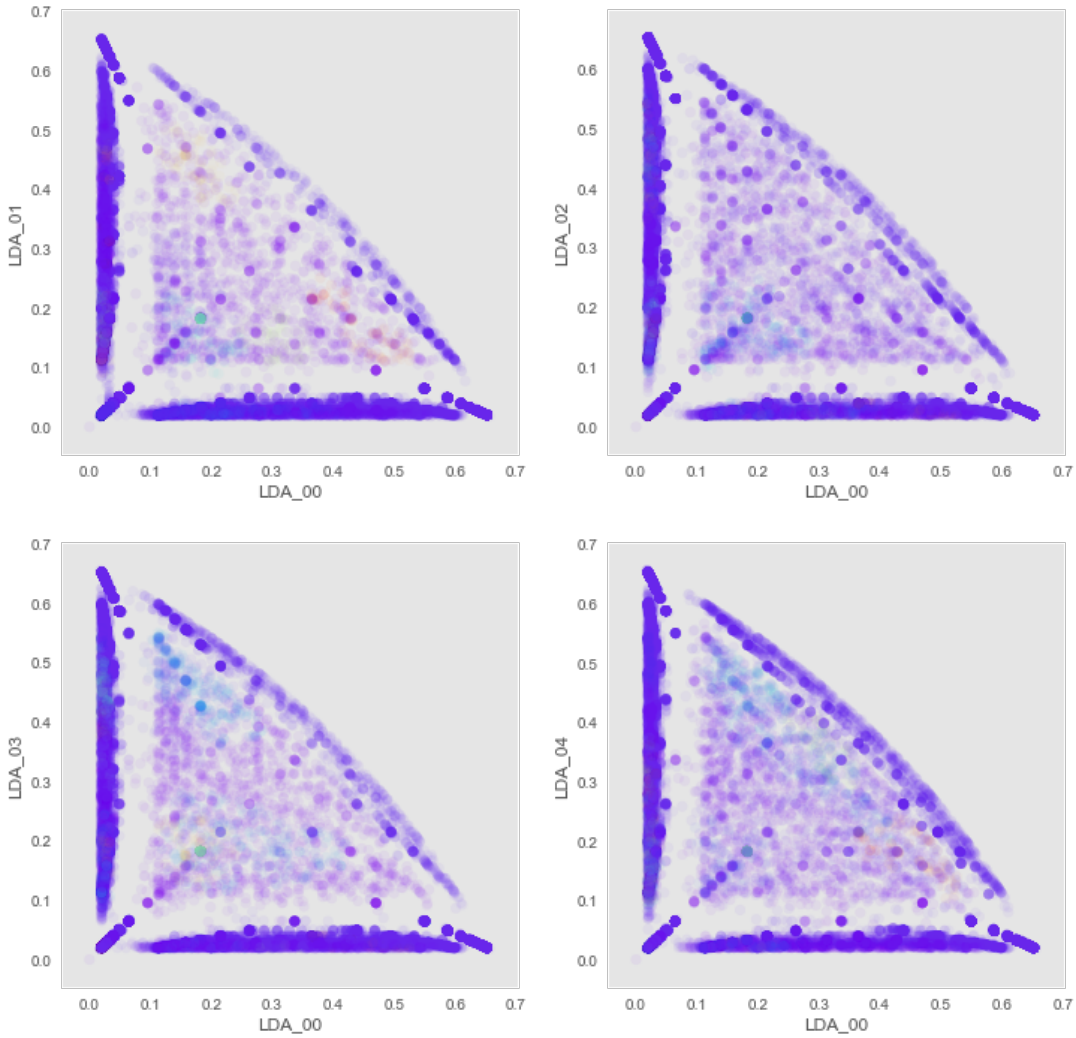

```
eps, min_pts, nclusters =  0.03 190 18
silhouette =  0.079039109307
```



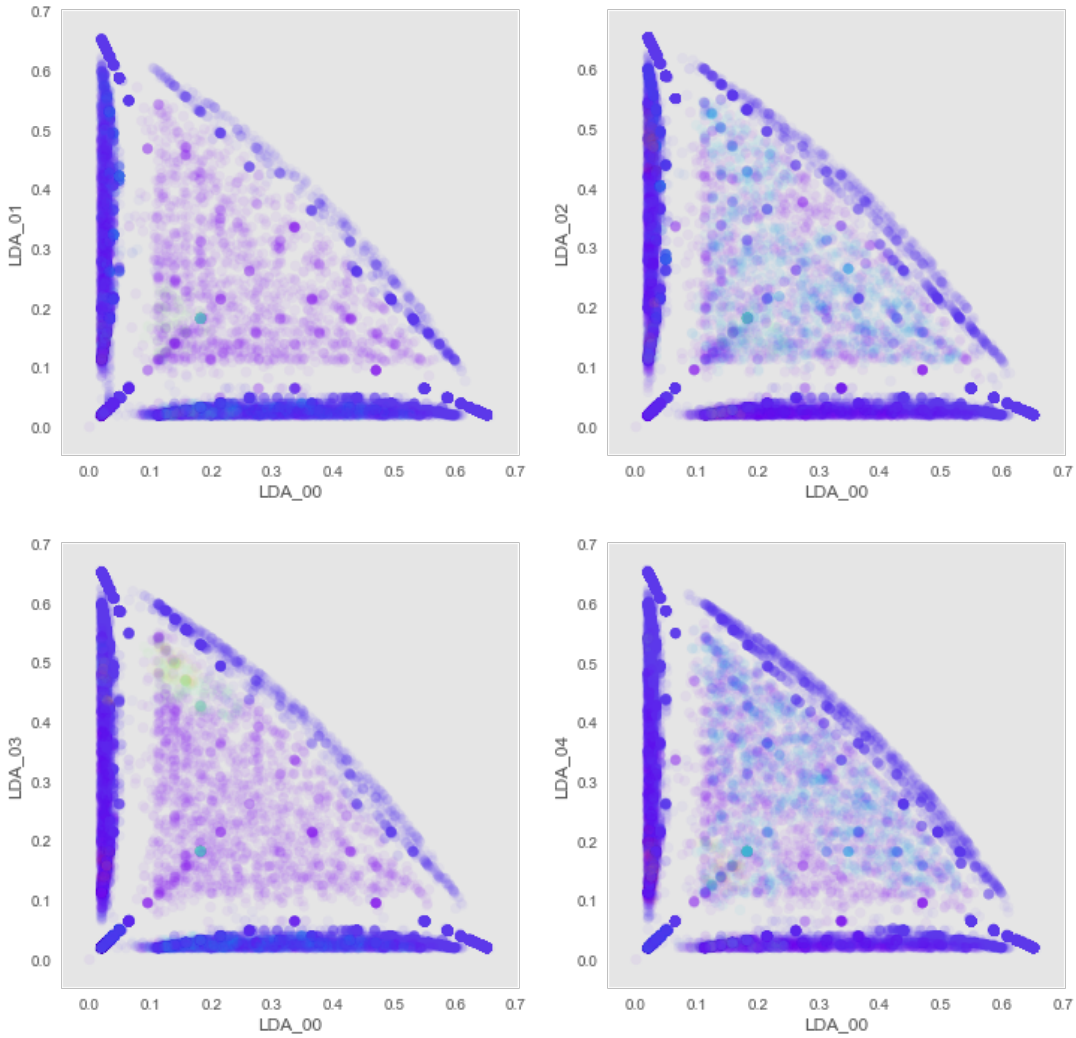

```
eps, min_pts, nclusters =  0.05 10 13
silhouette =  -0.554975391505
```



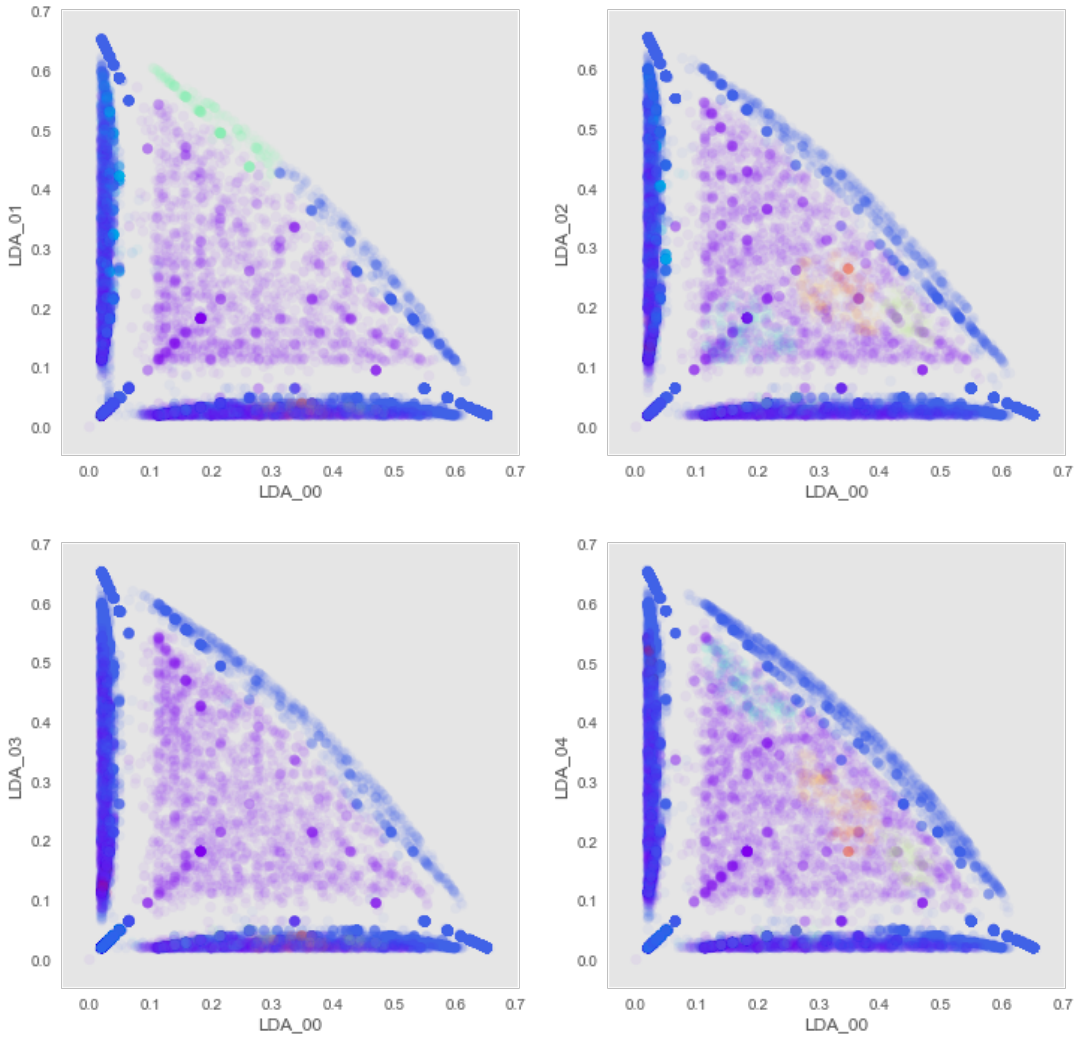

```
eps, min_pts, nclusters =  0.05 30 22
silhouette =  -0.583245695382
```



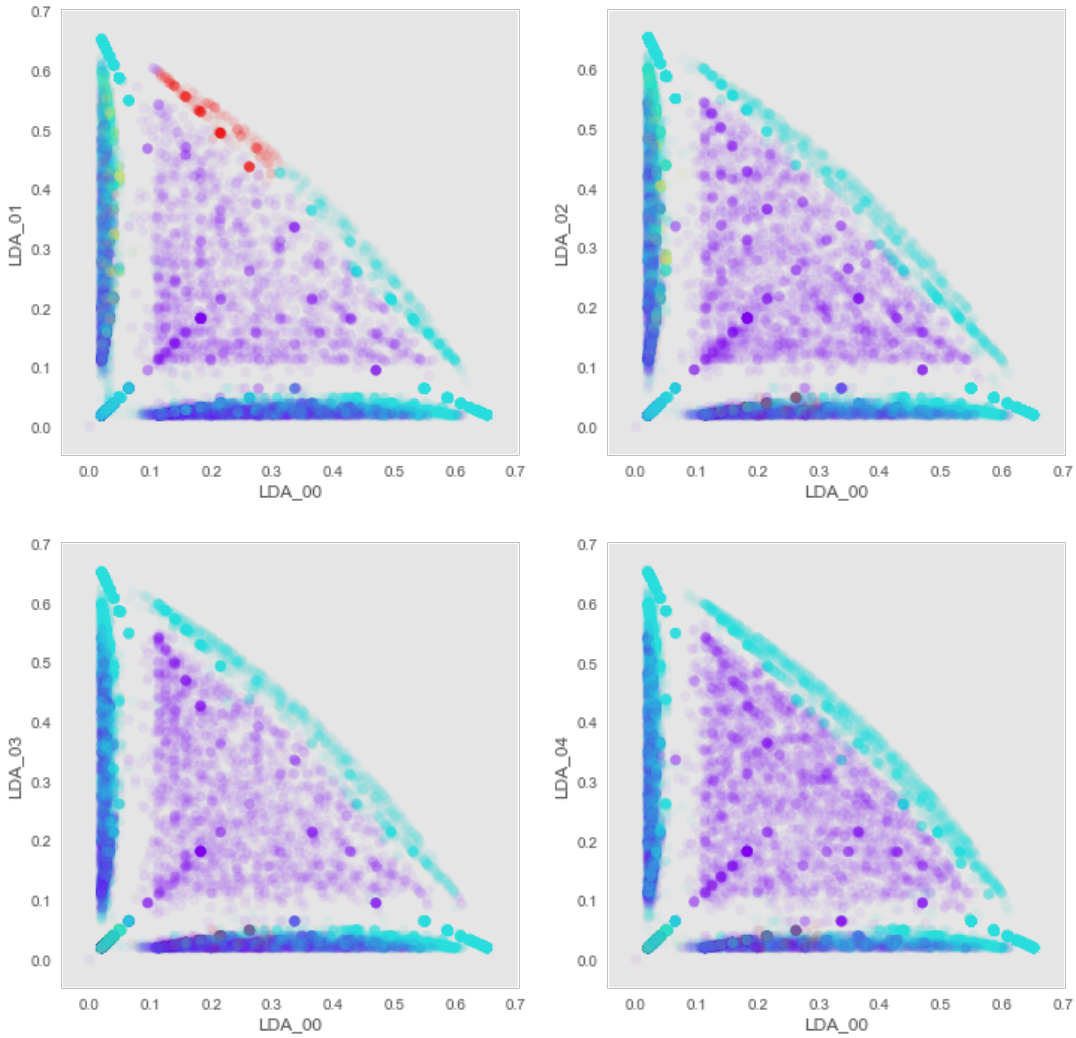

```
eps, min_pts, nclusters =  0.05 50 15
silhouette =  -0.508581617653
```



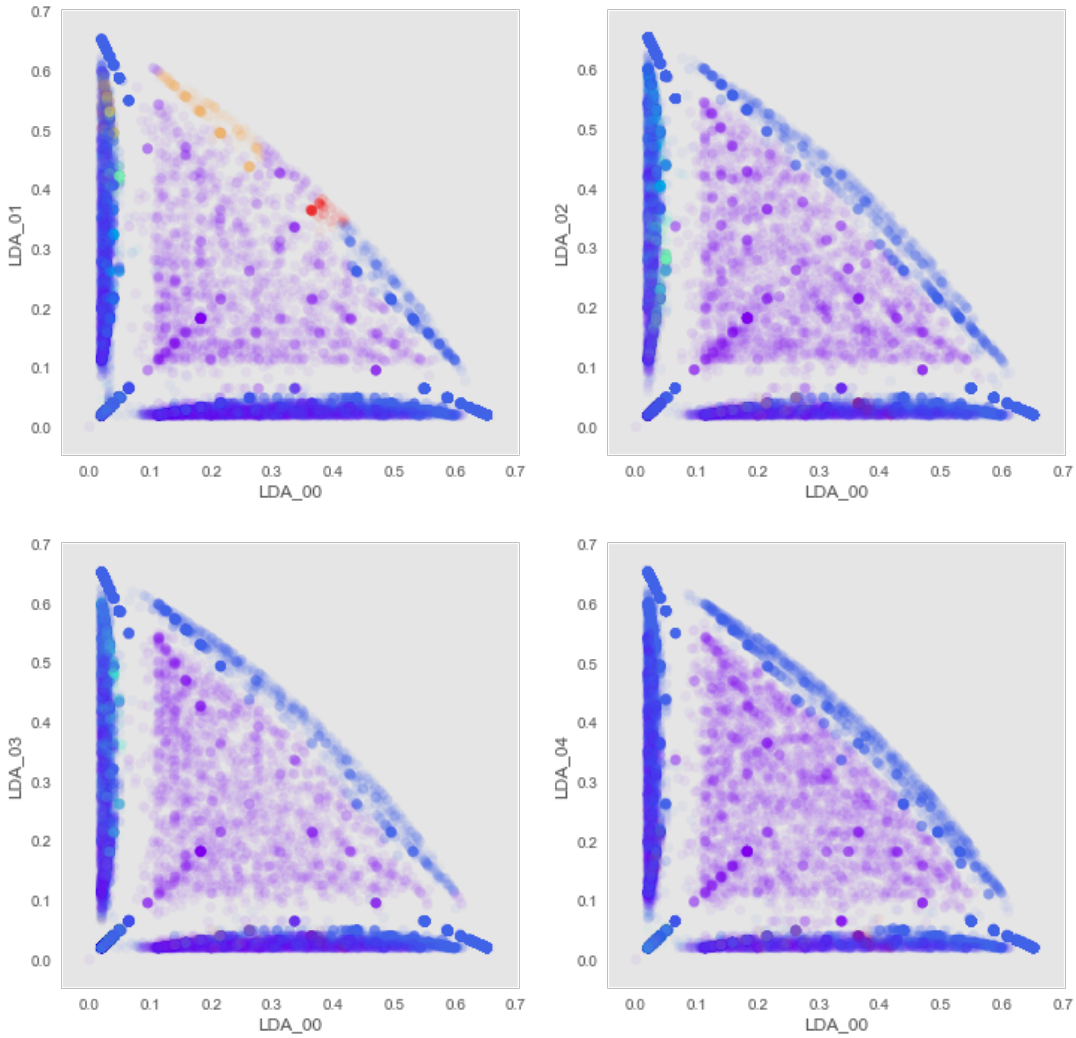

```
eps, min_pts, nclusters =  0.05 70 9
silhouette =  -0.384019284454
```



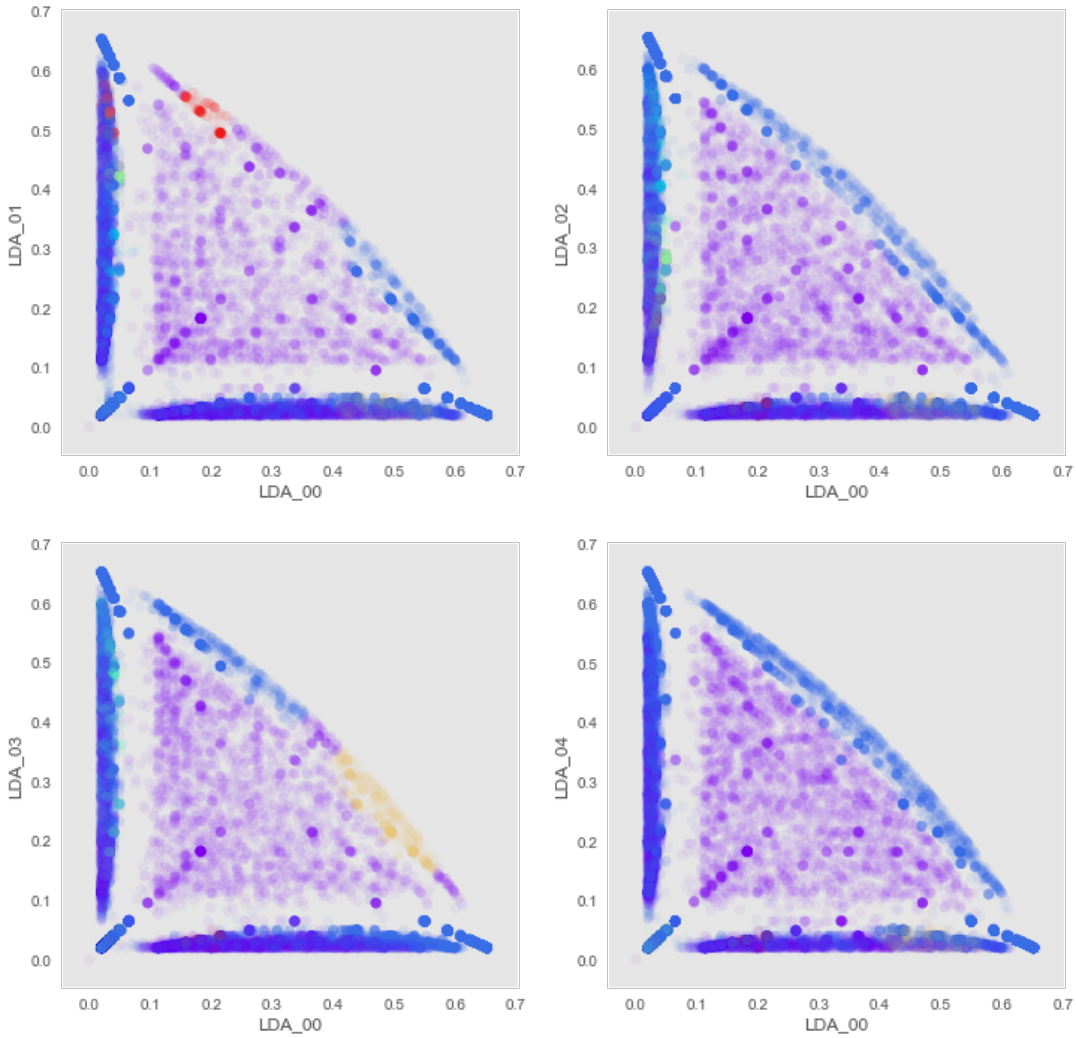

```
eps, min_pts, nclusters =  0.05 90 4
silhouette =  -0.143251757675
```



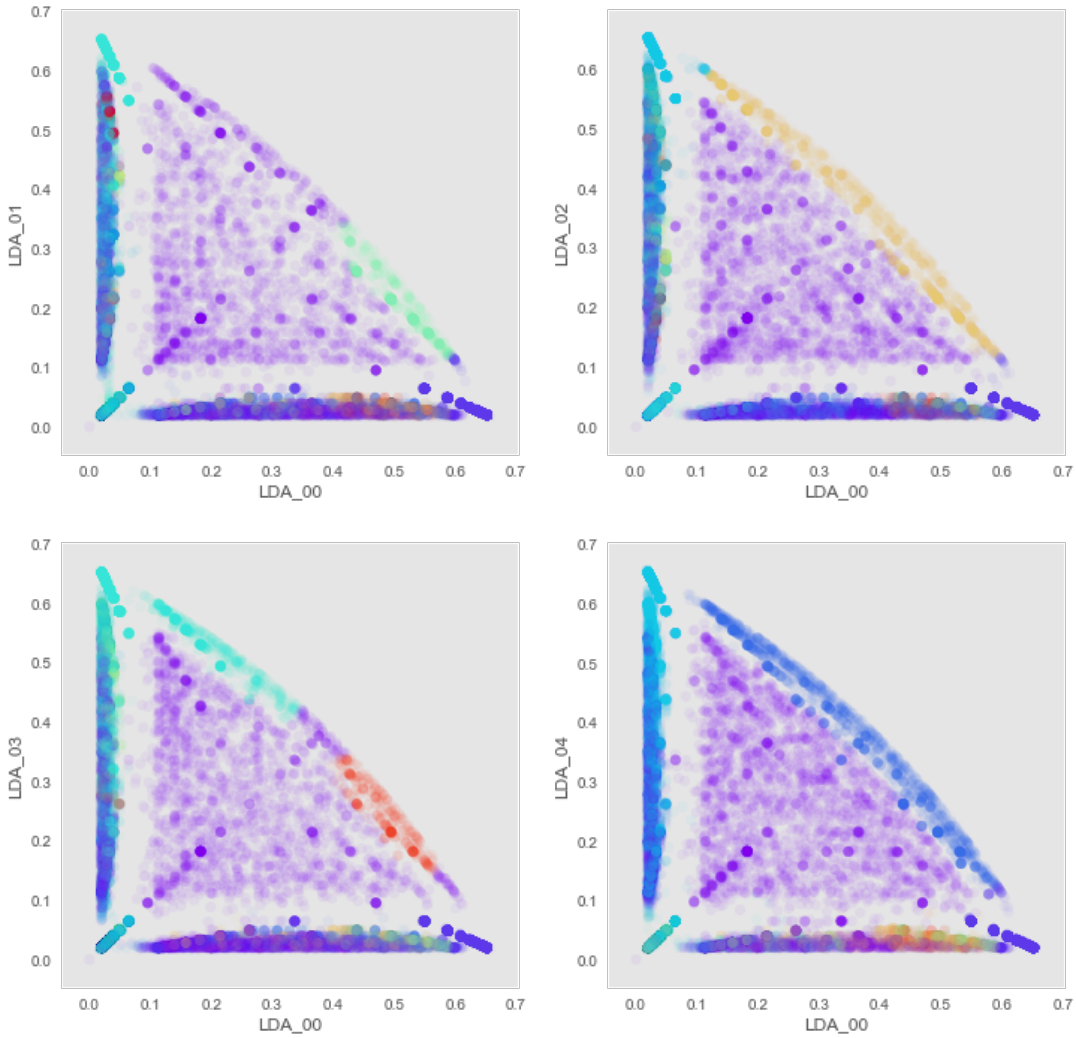

```
eps, min_pts, nclusters =  0.05 110 9
silhouette =  -0.266437896558
```



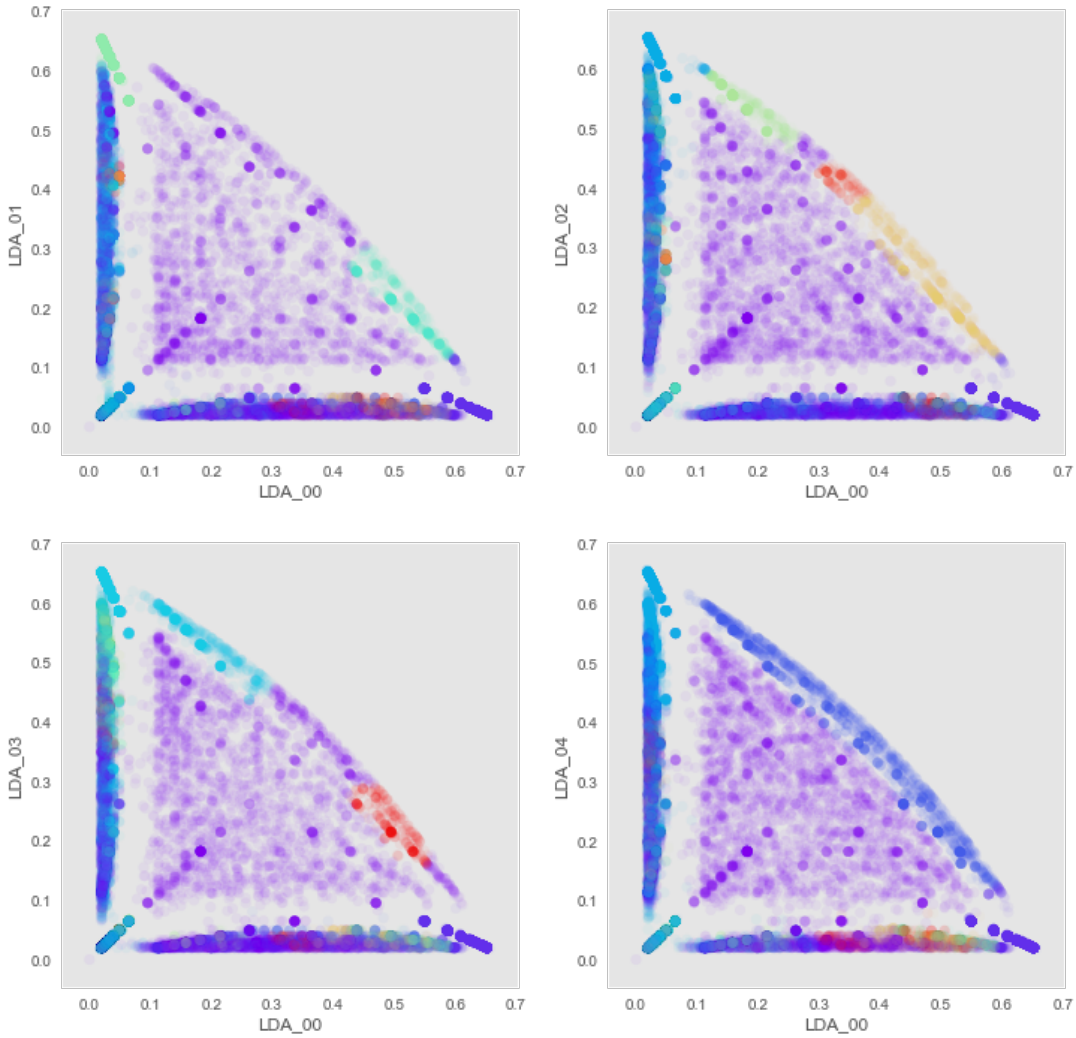

```
eps, min_pts, nclusters =  0.05 130 8
silhouette =  -0.291059815599
```



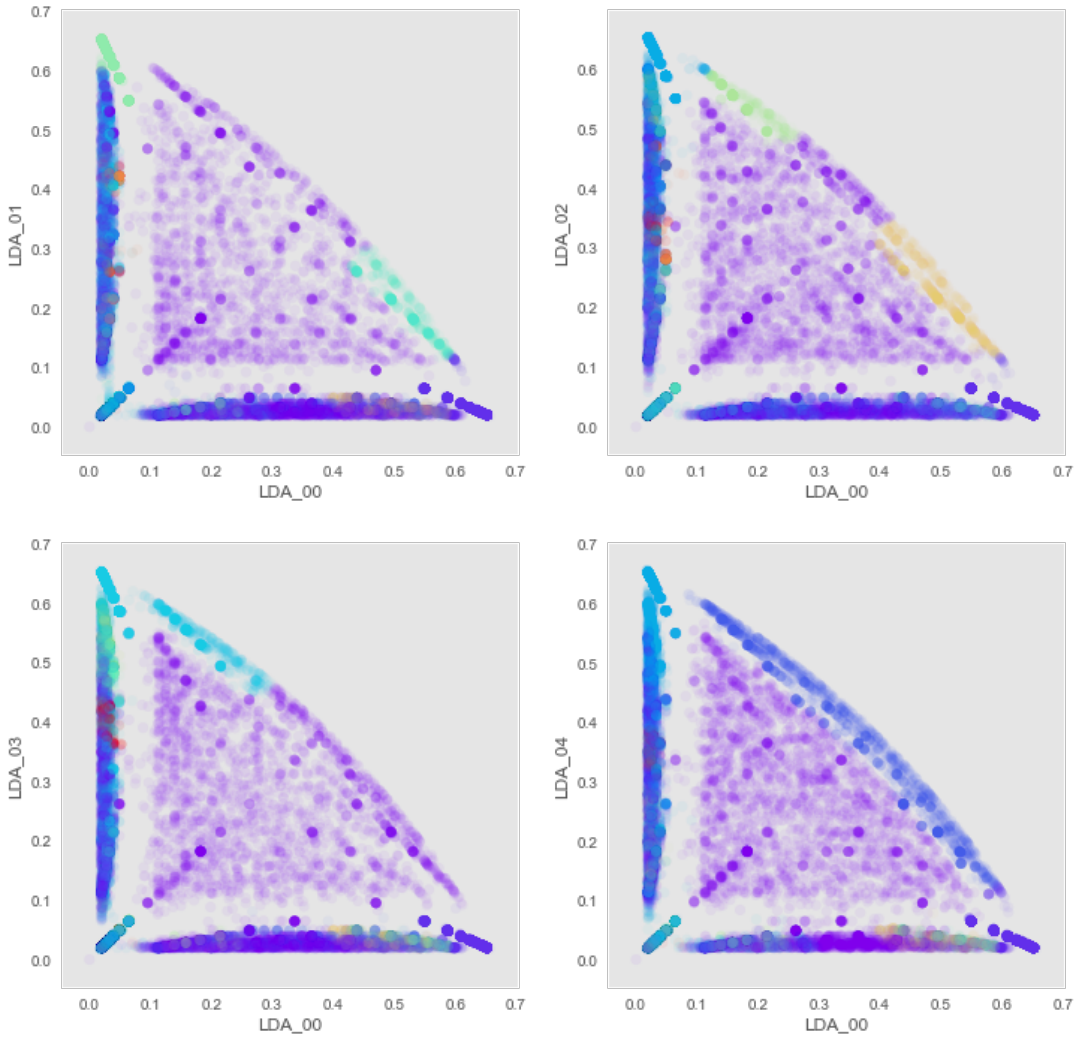

```
eps, min_pts, nclusters =  0.05 150 15
silhouette =  -0.0876455920594
```



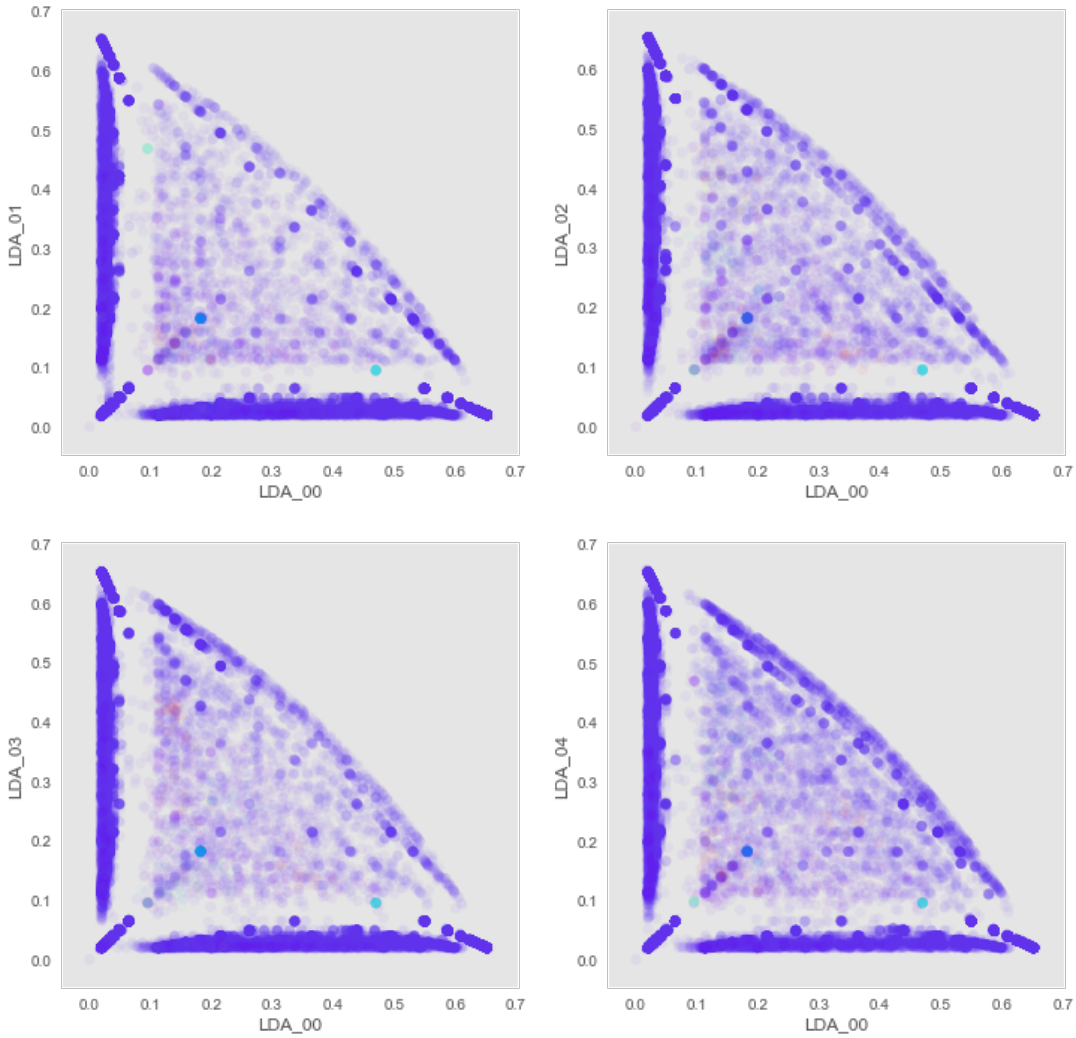

```
eps, min_pts, nclusters =  0.05 170 18
silhouette =  -0.00363225510685
```



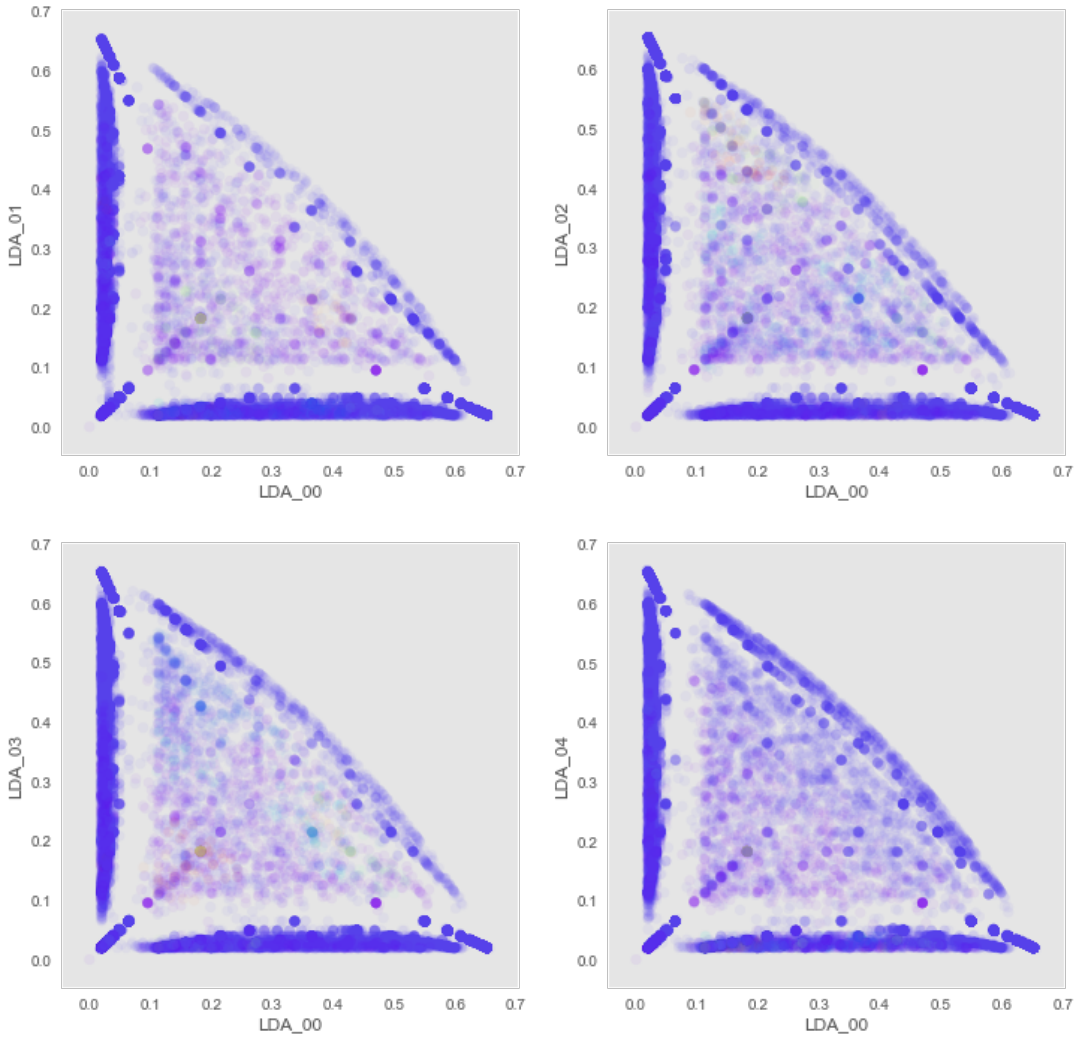

```
eps, min_pts, nclusters =  0.05 190 18
silhouette =  -0.0219842244597
```



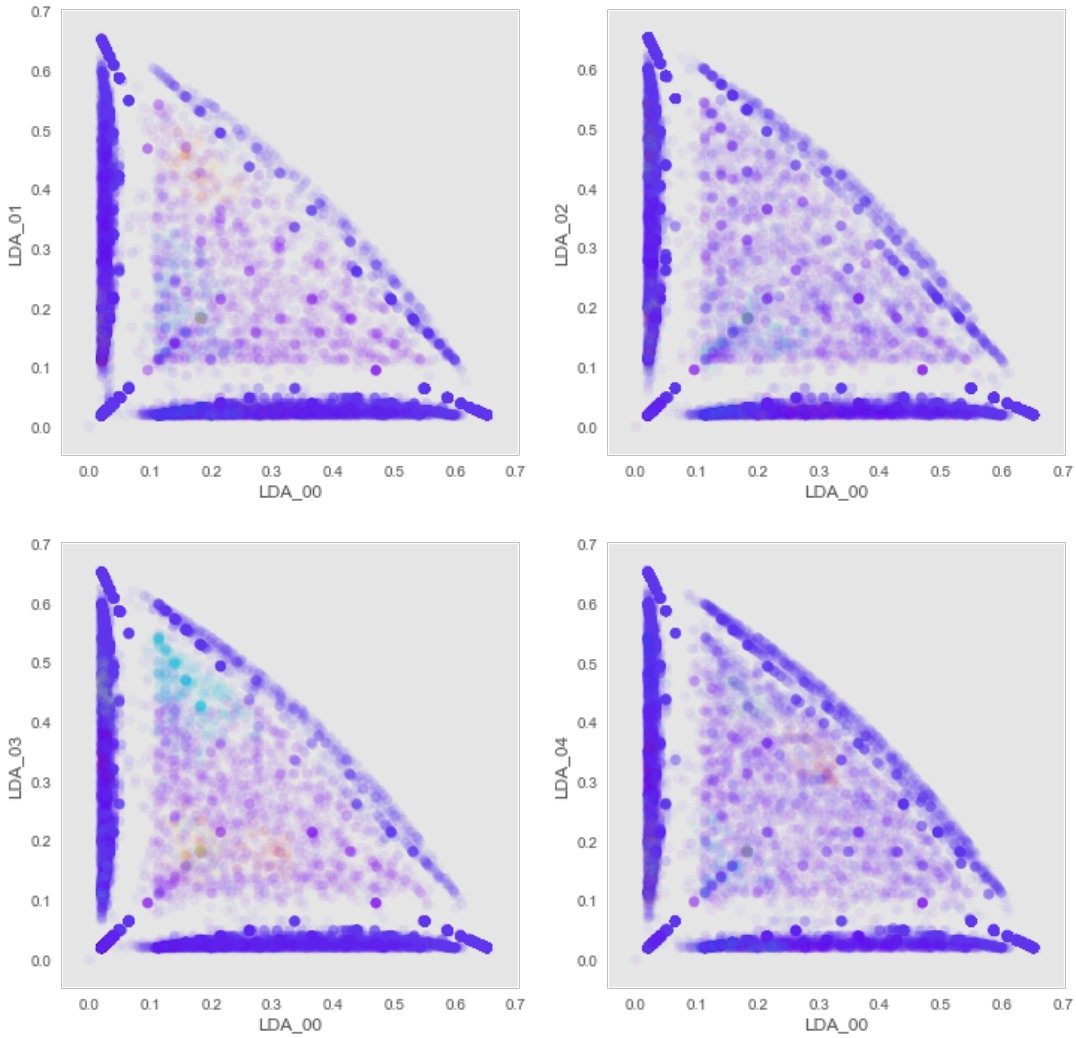

```
eps, min_pts, nclusters =  0.06 10 17
silhouette =  -0.625087159318
```



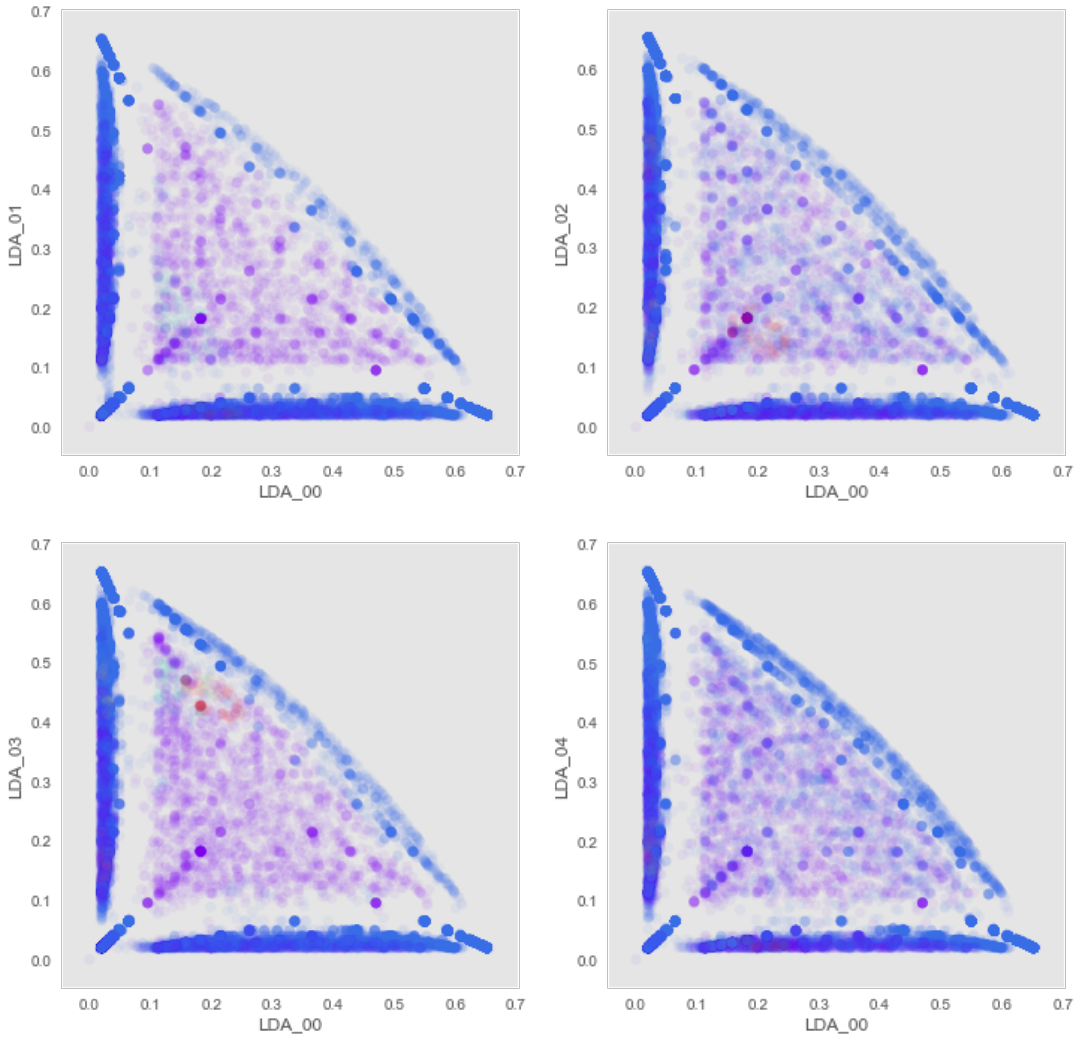

```
eps, min_pts, nclusters =  0.06 30 13
silhouette =  -0.509683854142
```



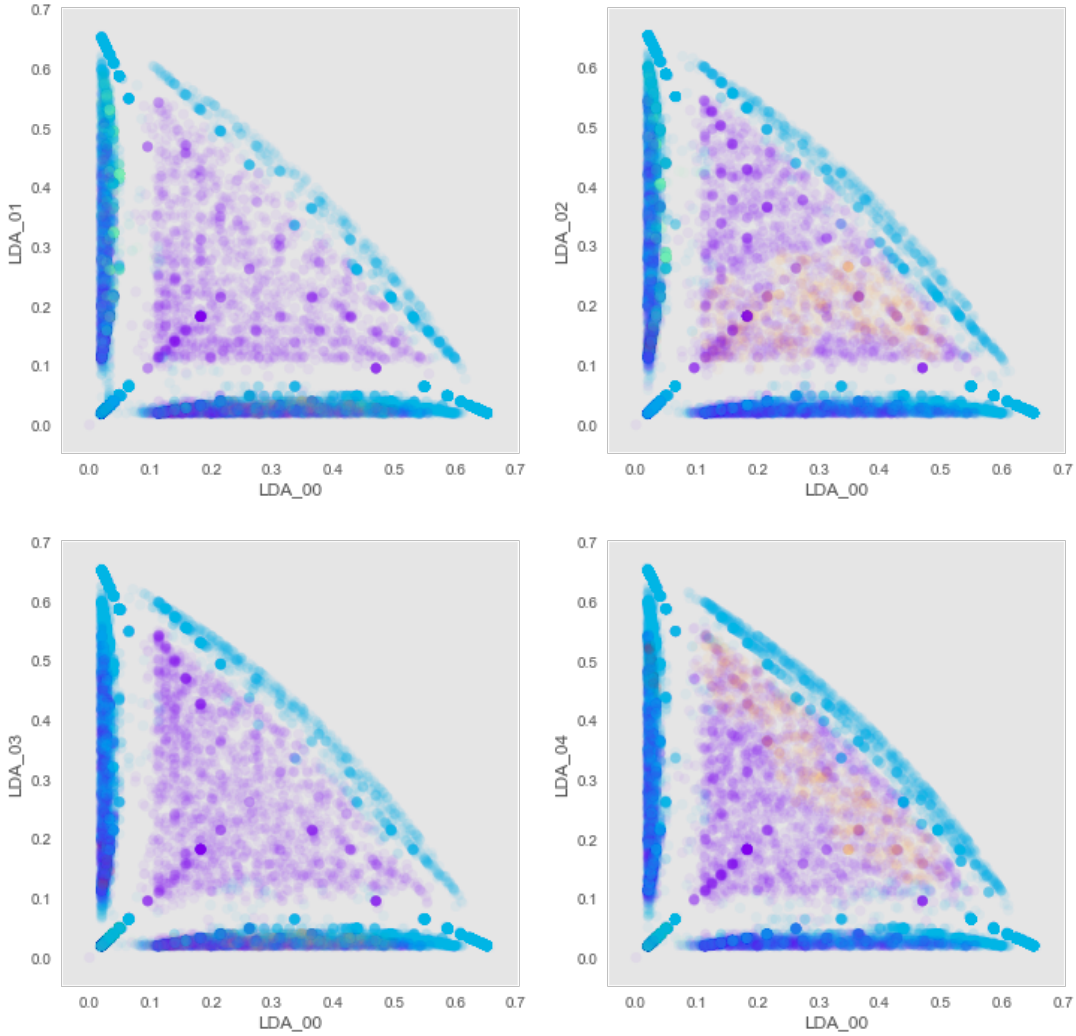

```
eps, min_pts, nclusters =  0.06 50 16
silhouette =  -0.547325246322
```



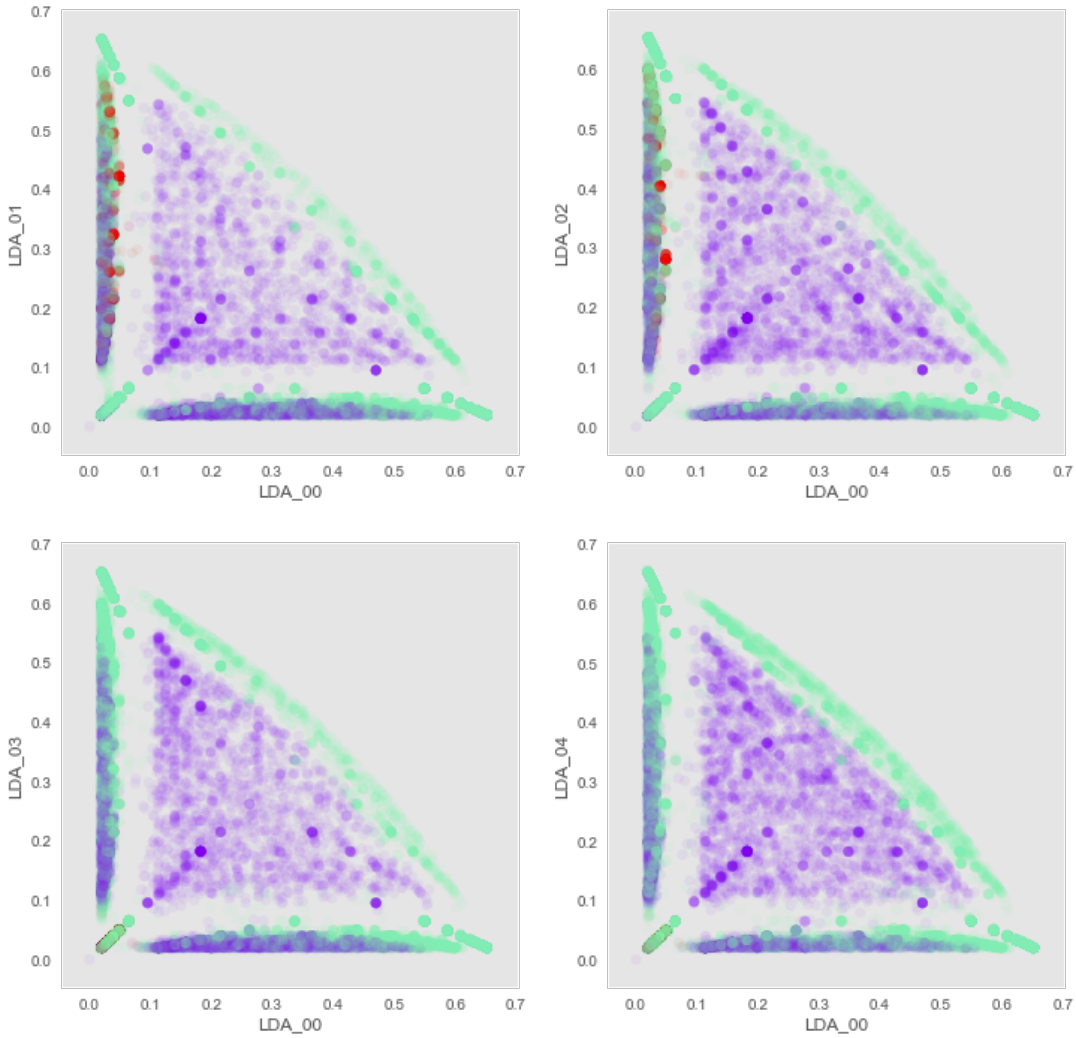

```
eps, min_pts, nclusters =  0.06 70 8
silhouette =  -0.458514547266
```



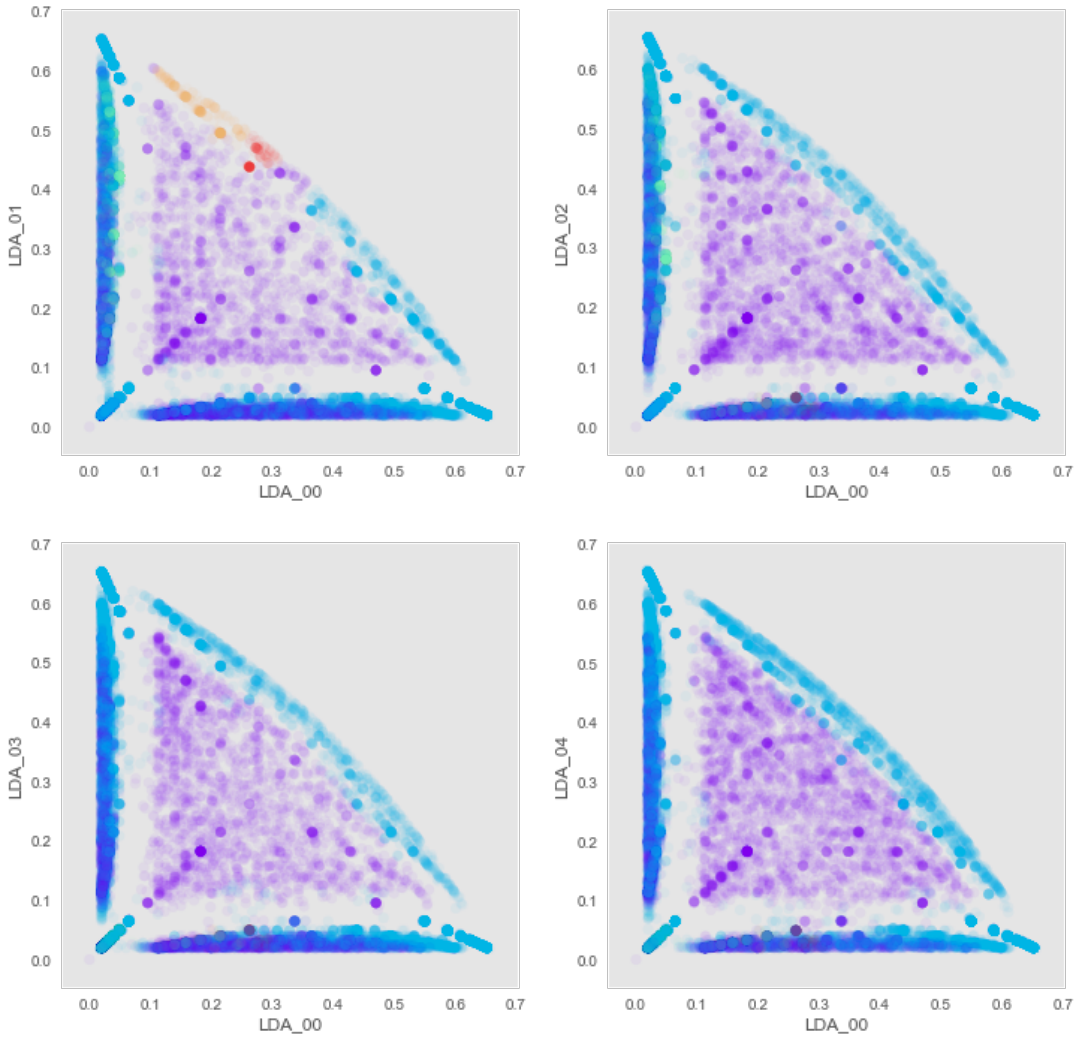

```
eps, min_pts, nclusters =  0.06 90 5
silhouette =  -0.294859343785
```



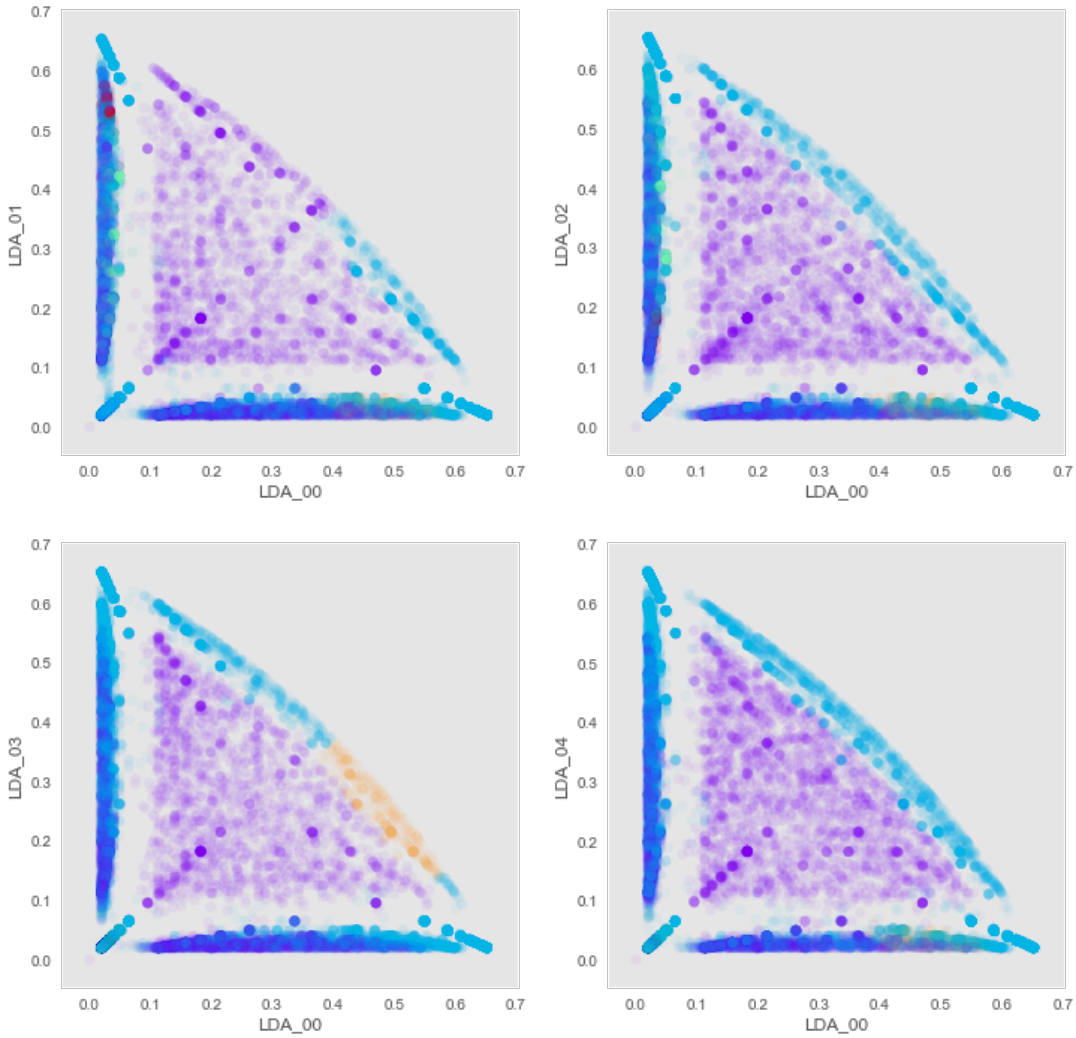

```
eps, min_pts, nclusters =  0.06 110 3
silhouette =  -0.115204301547
```



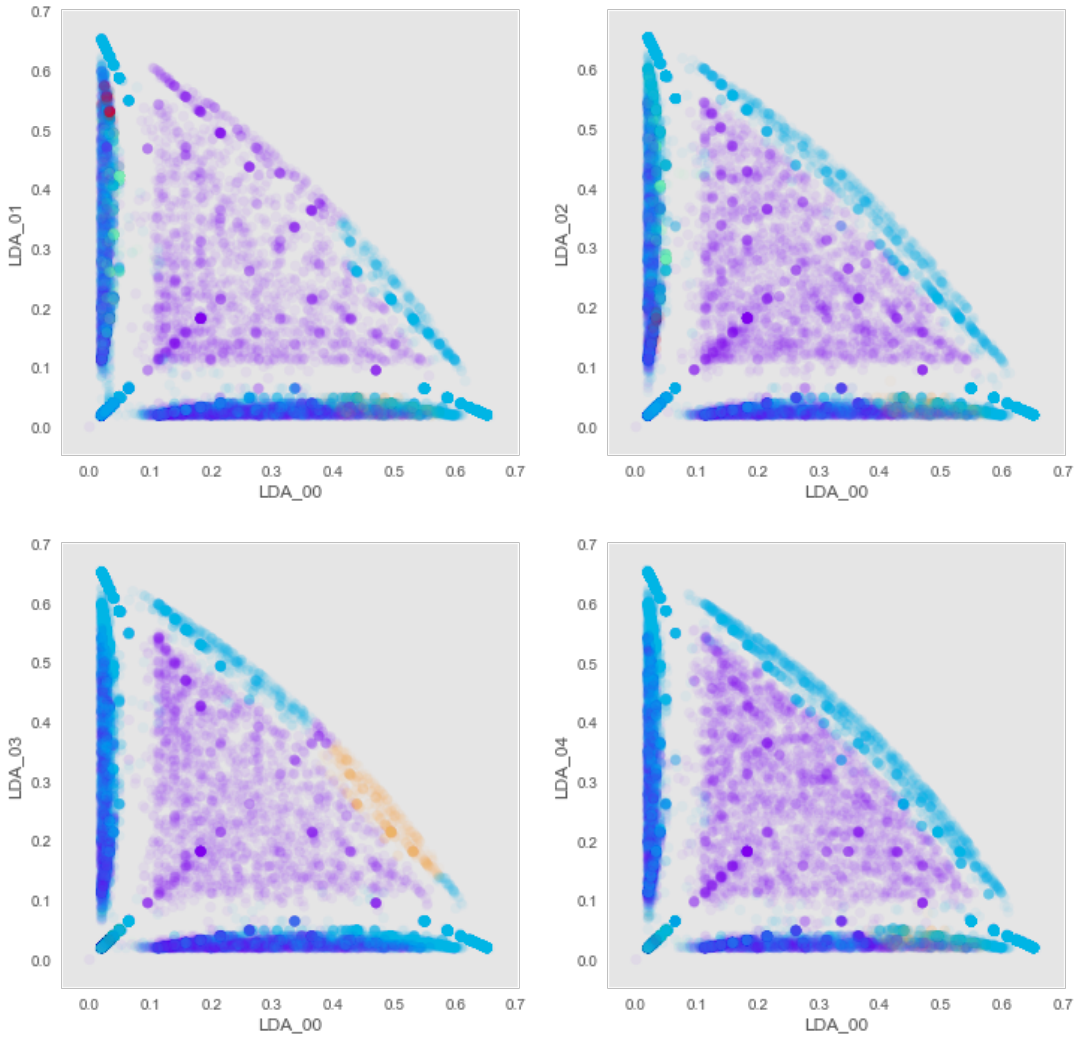

```
eps, min_pts, nclusters =  0.06 130 5
silhouette =  -0.162619911913
```



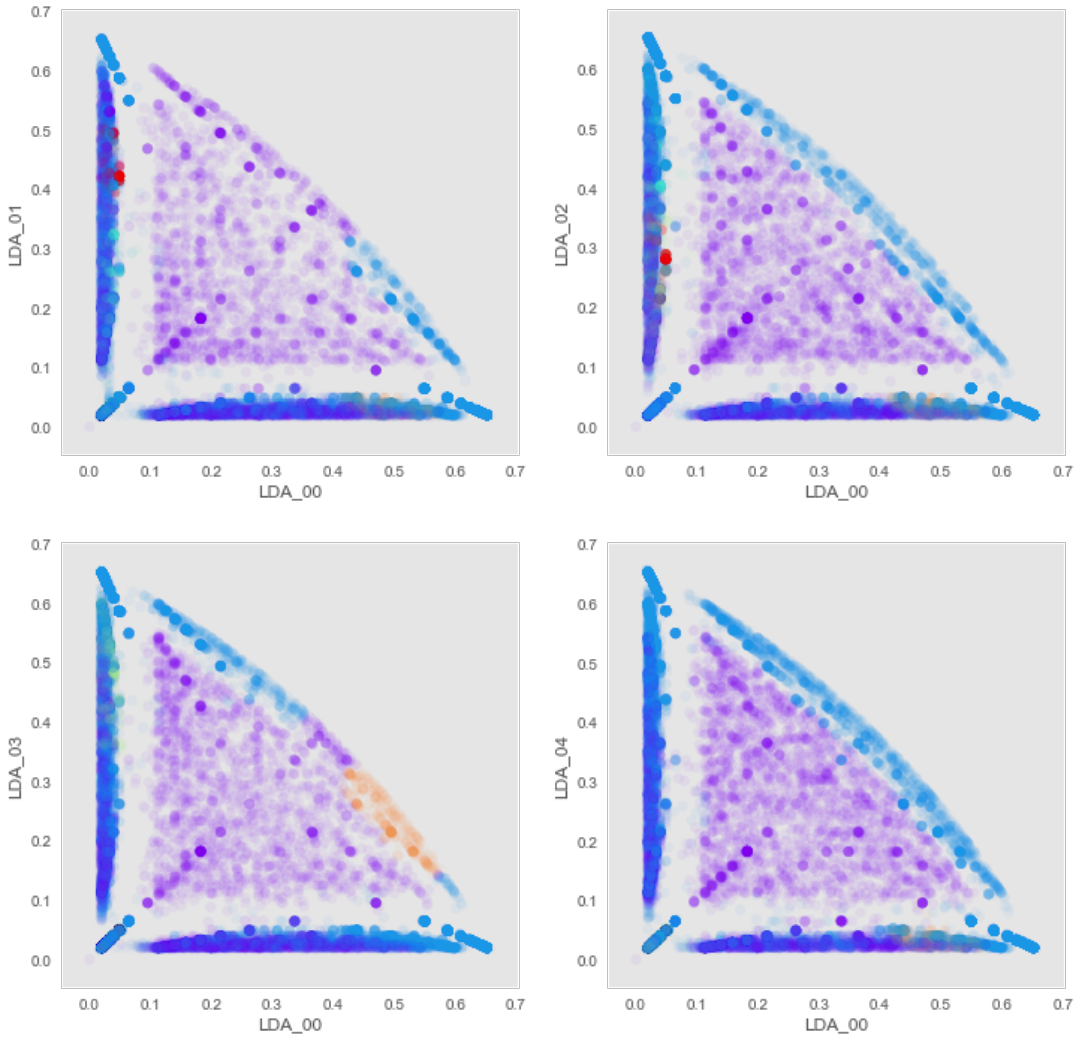

```
eps, min_pts, nclusters =  0.06 150 5
silhouette =  -0.236628413768
```



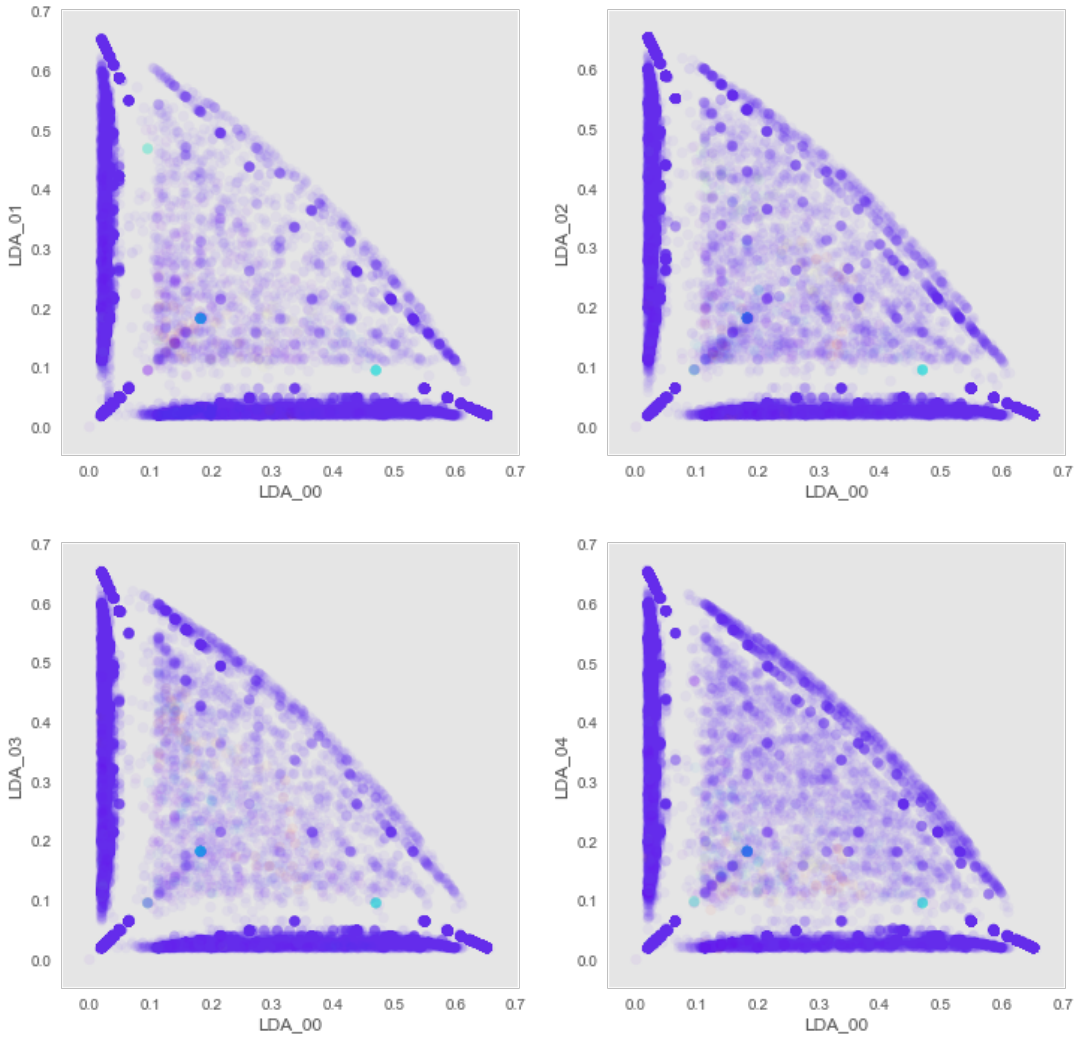

```
eps, min_pts, nclusters =  0.06 170 5
silhouette =  -0.240219582278
```



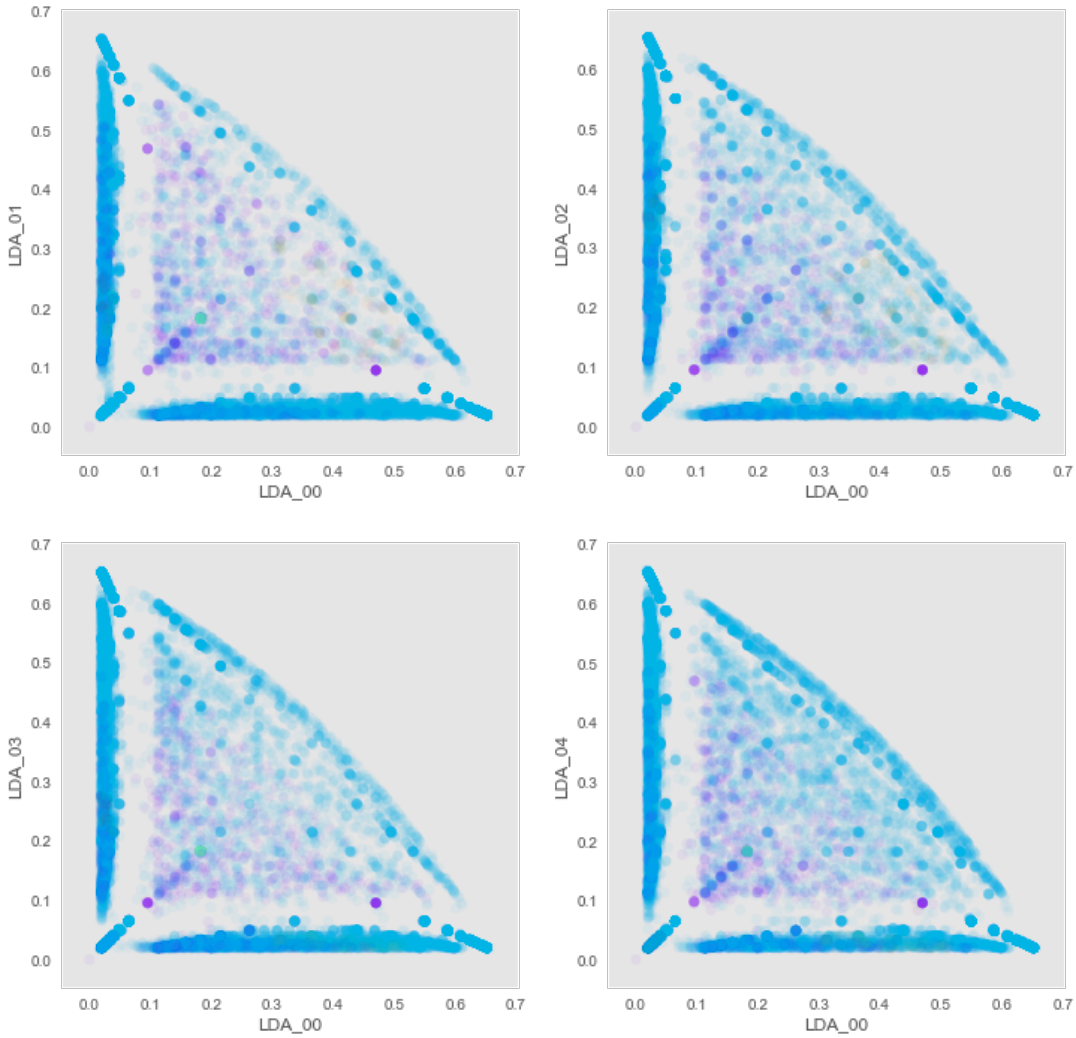

```
eps, min_pts, nclusters =  0.06 190 6
silhouette =  -0.328024893617
```



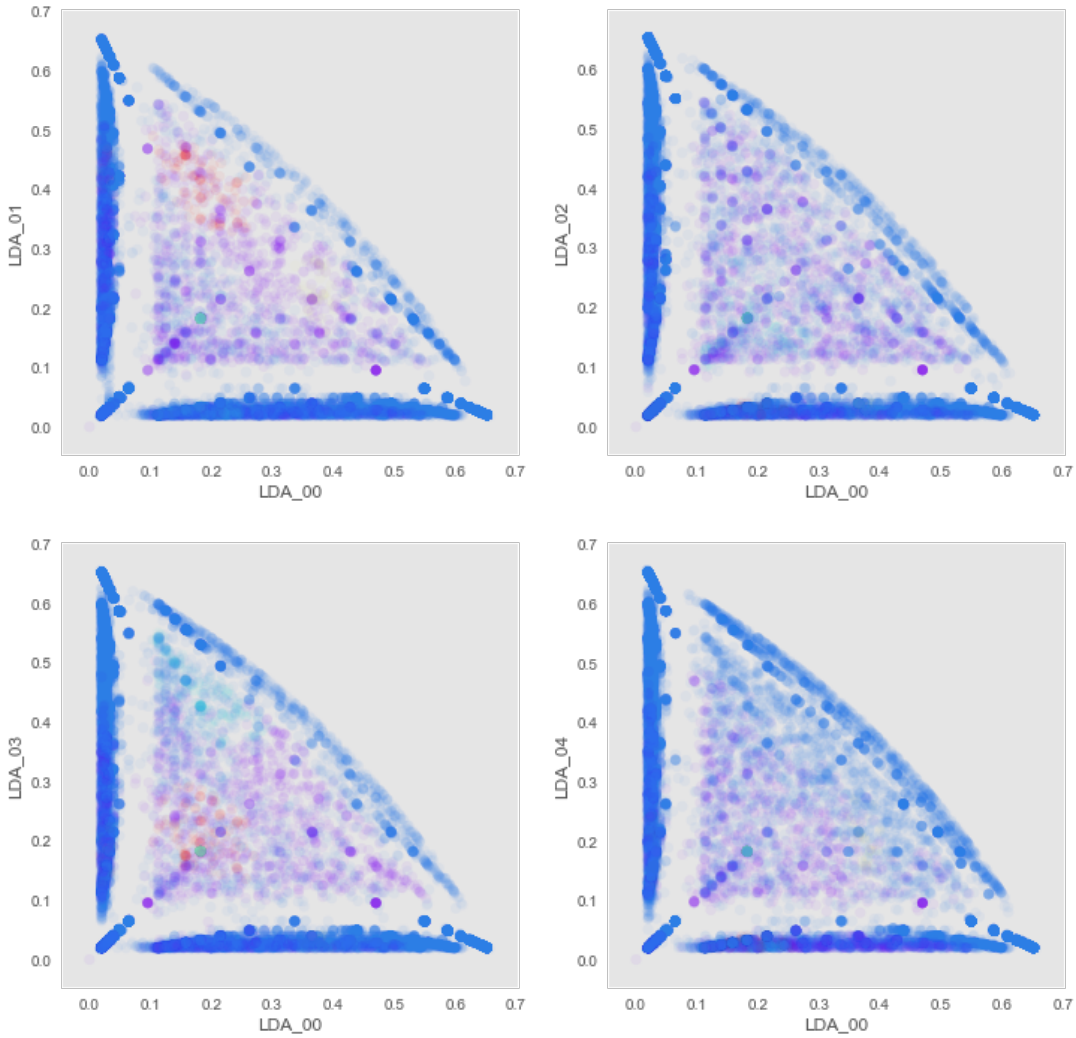

```
eps, min_pts, nclusters =  0.07 10 19
silhouette =  -0.593974977881
```



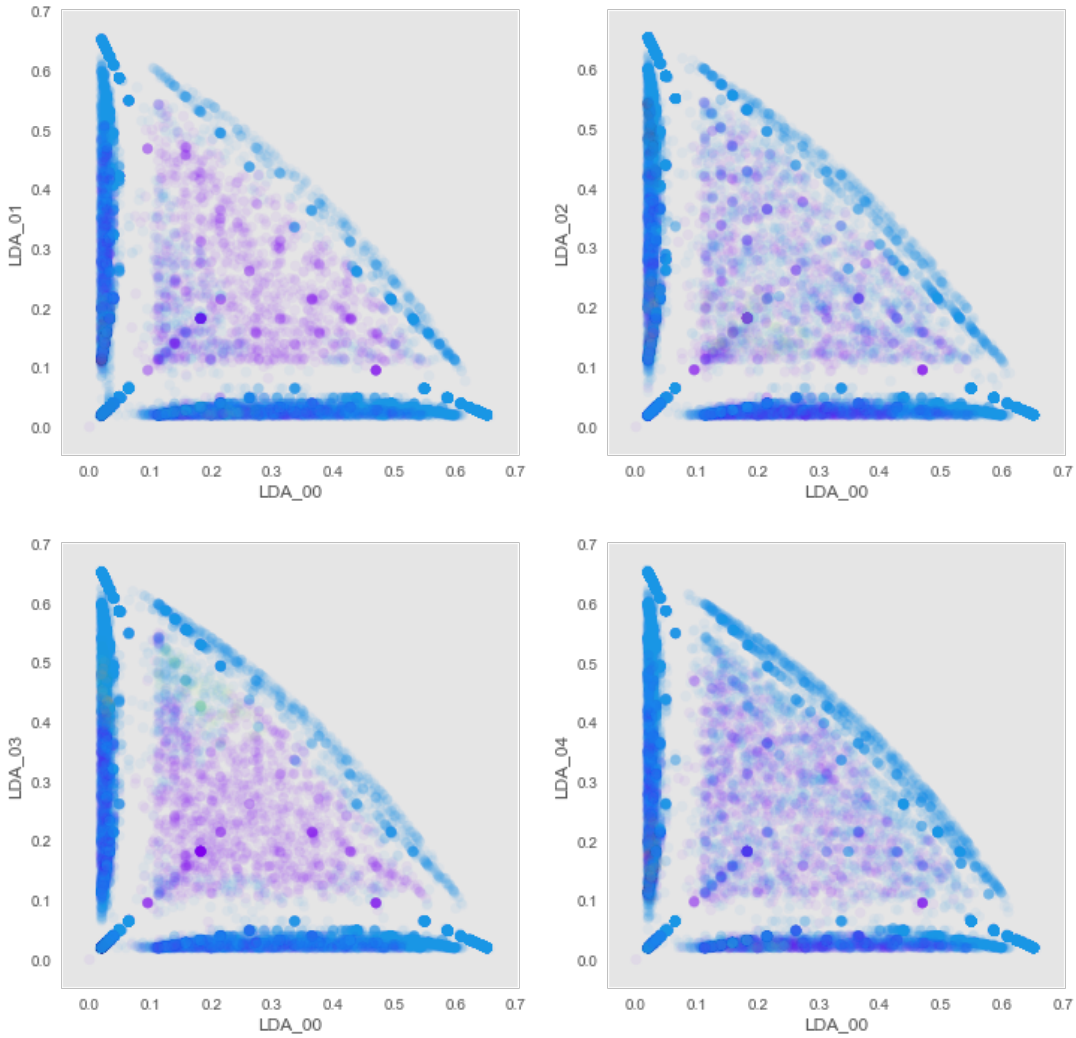

```
eps, min_pts, nclusters =  0.07 30 5
silhouette =  -0.331453669464
```



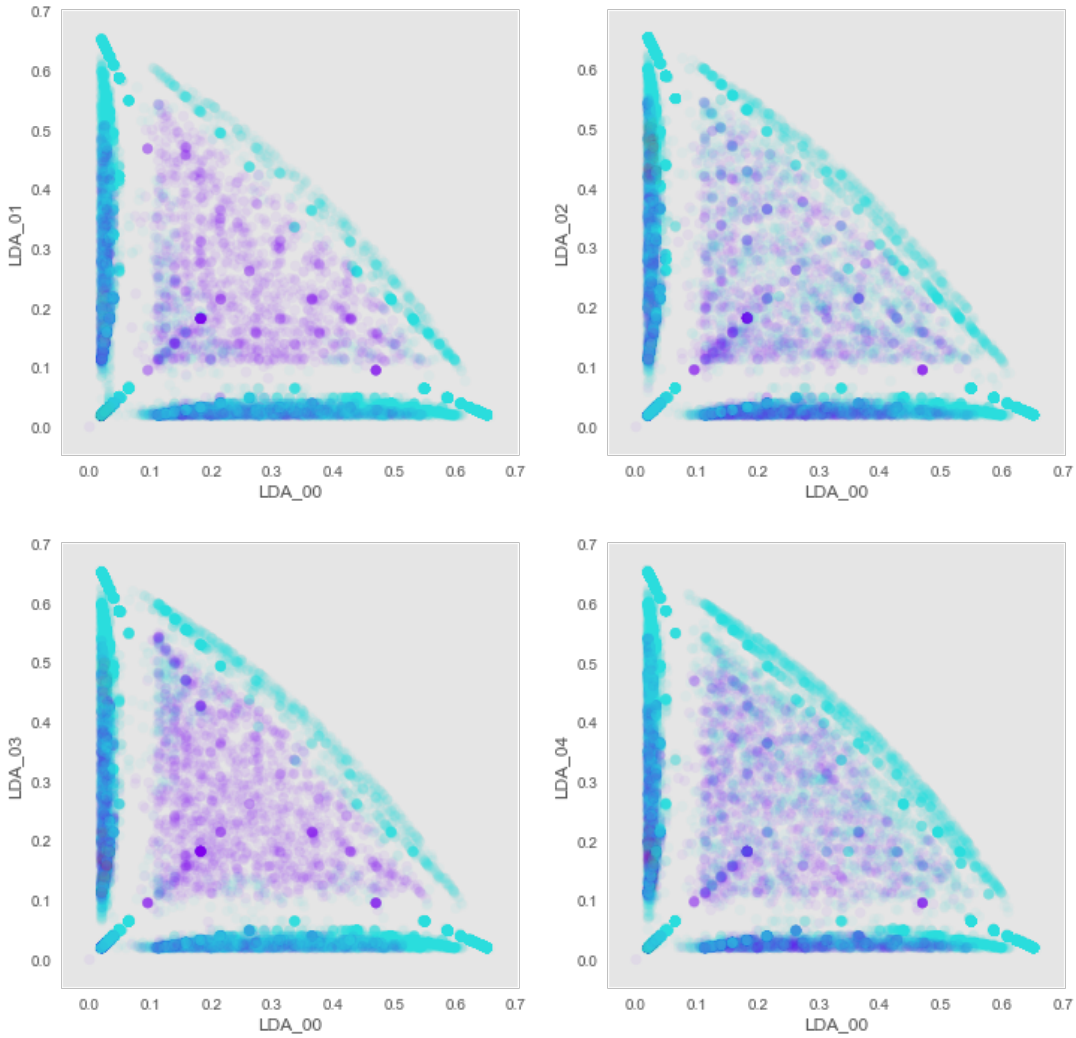

```
eps, min_pts, nclusters =  0.07 50 7
silhouette =  -0.391438906883
```



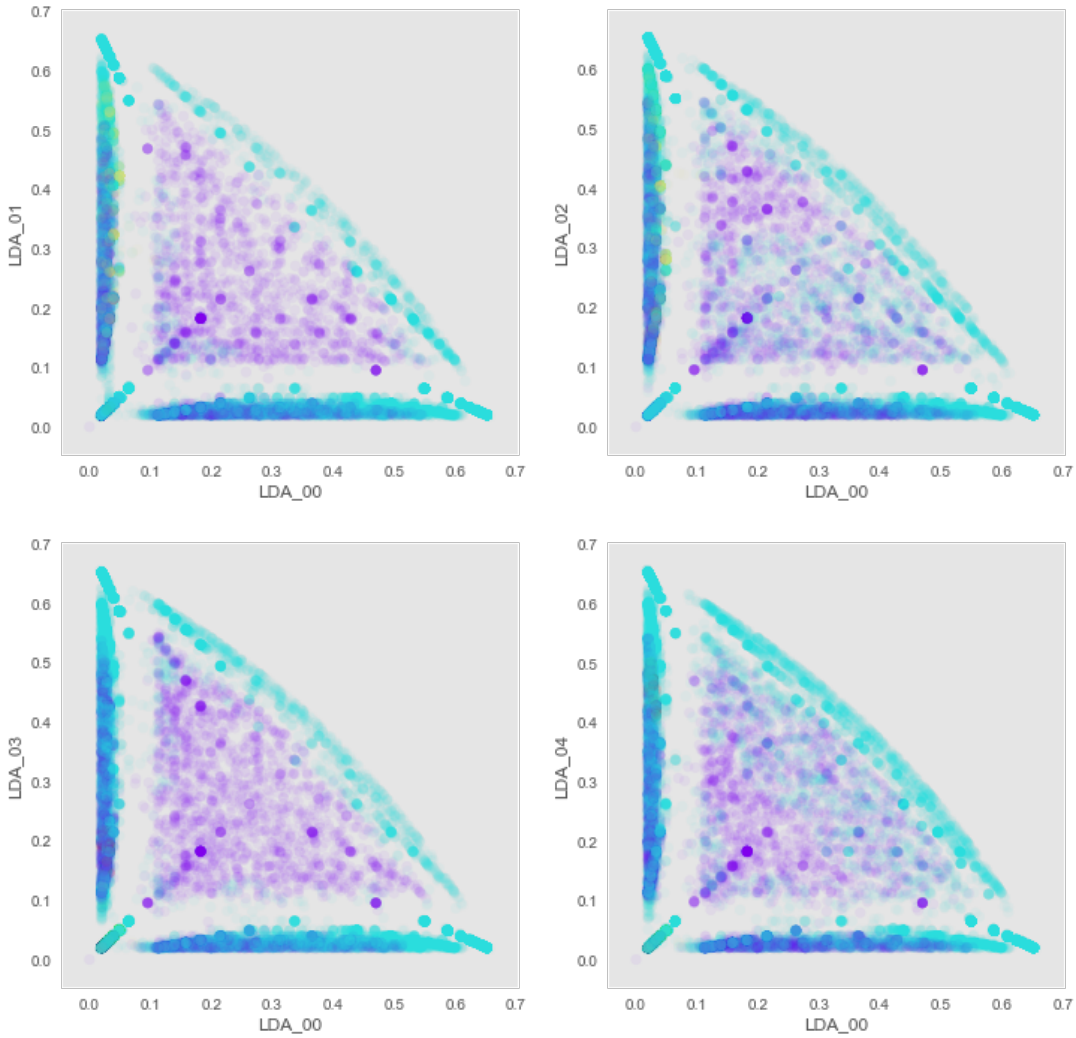

```
eps, min_pts, nclusters =  0.07 70 6
silhouette =  -0.357396694589
```



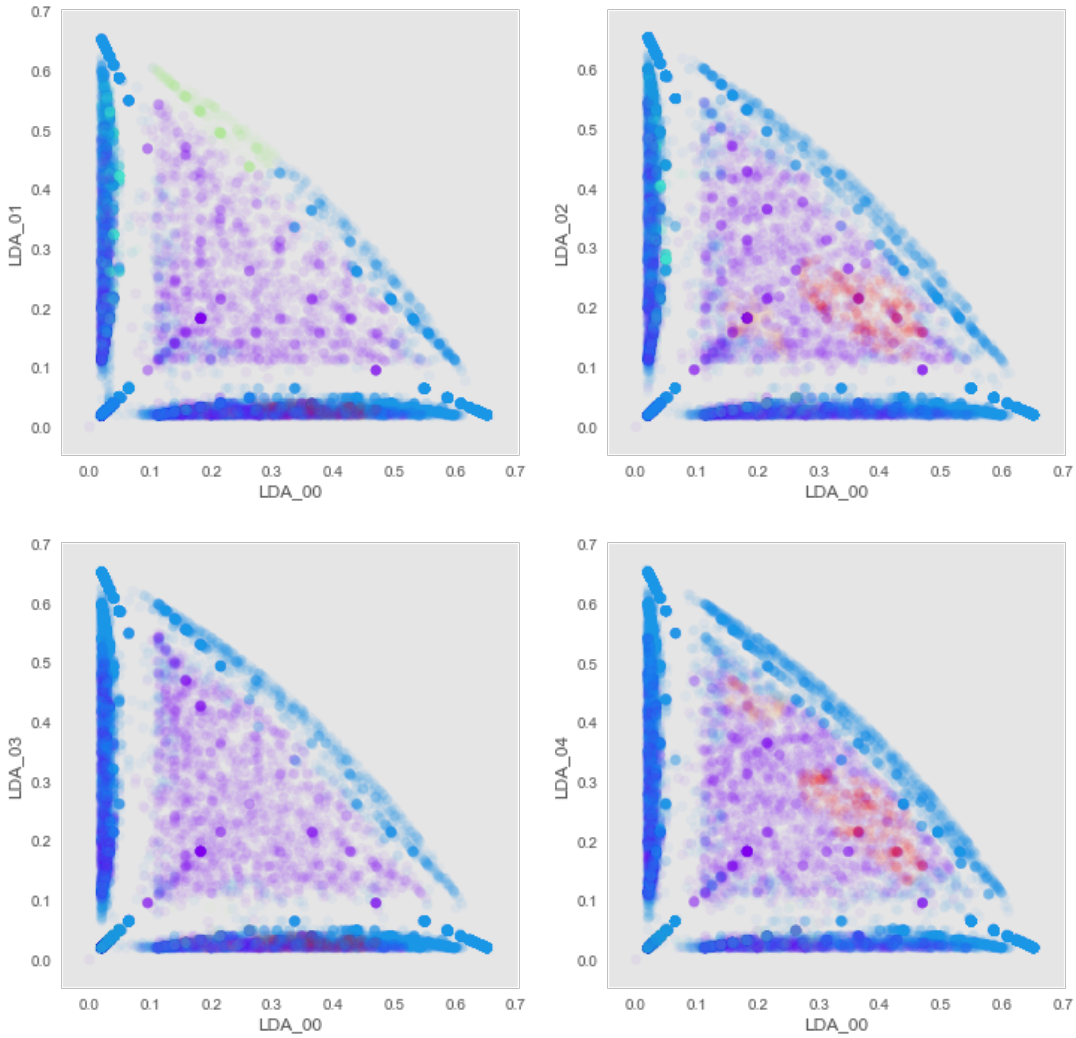

```
eps, min_pts, nclusters =  0.07 90 4
silhouette =  -0.327372947206
```



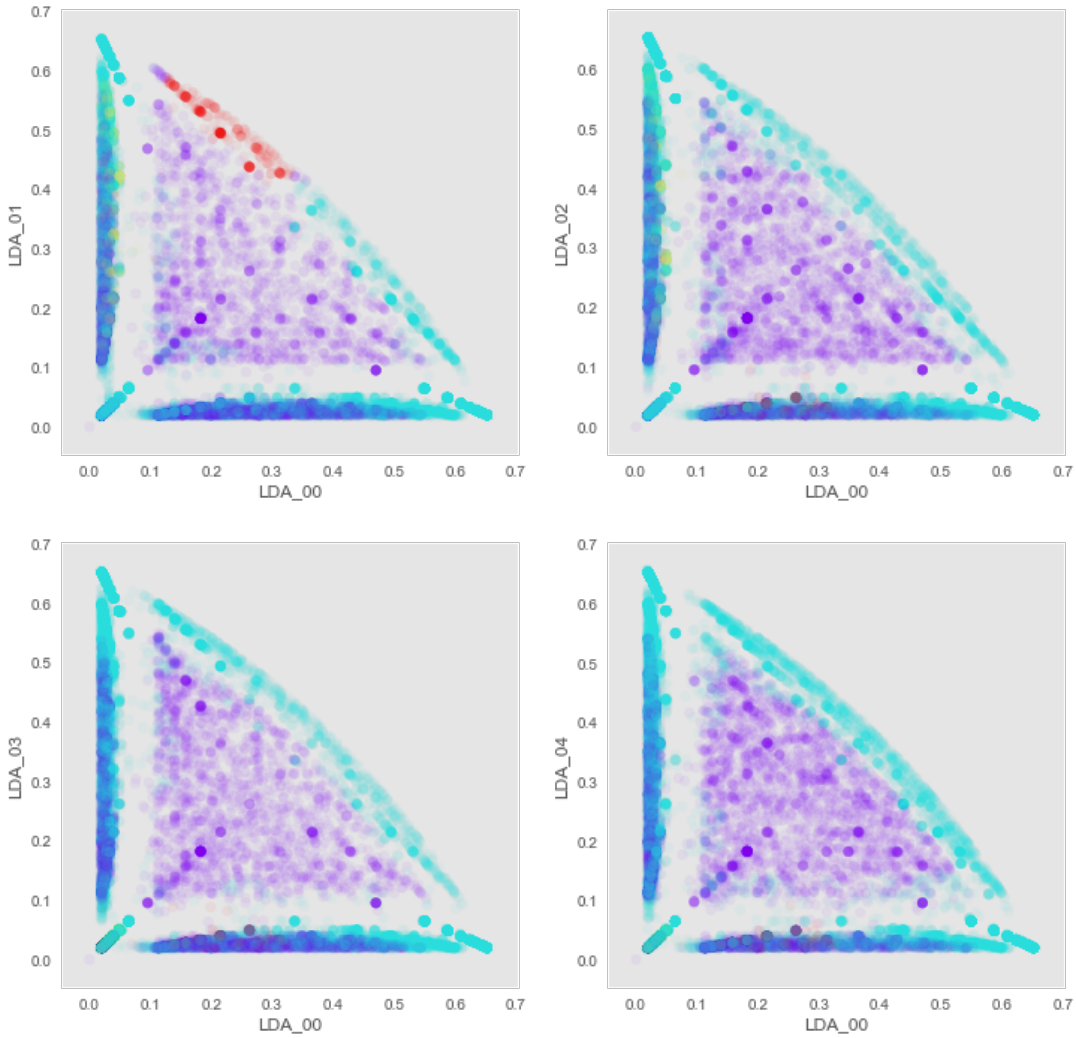

```
eps, min_pts, nclusters =  0.07 110 4
silhouette =  -0.249839903911
```



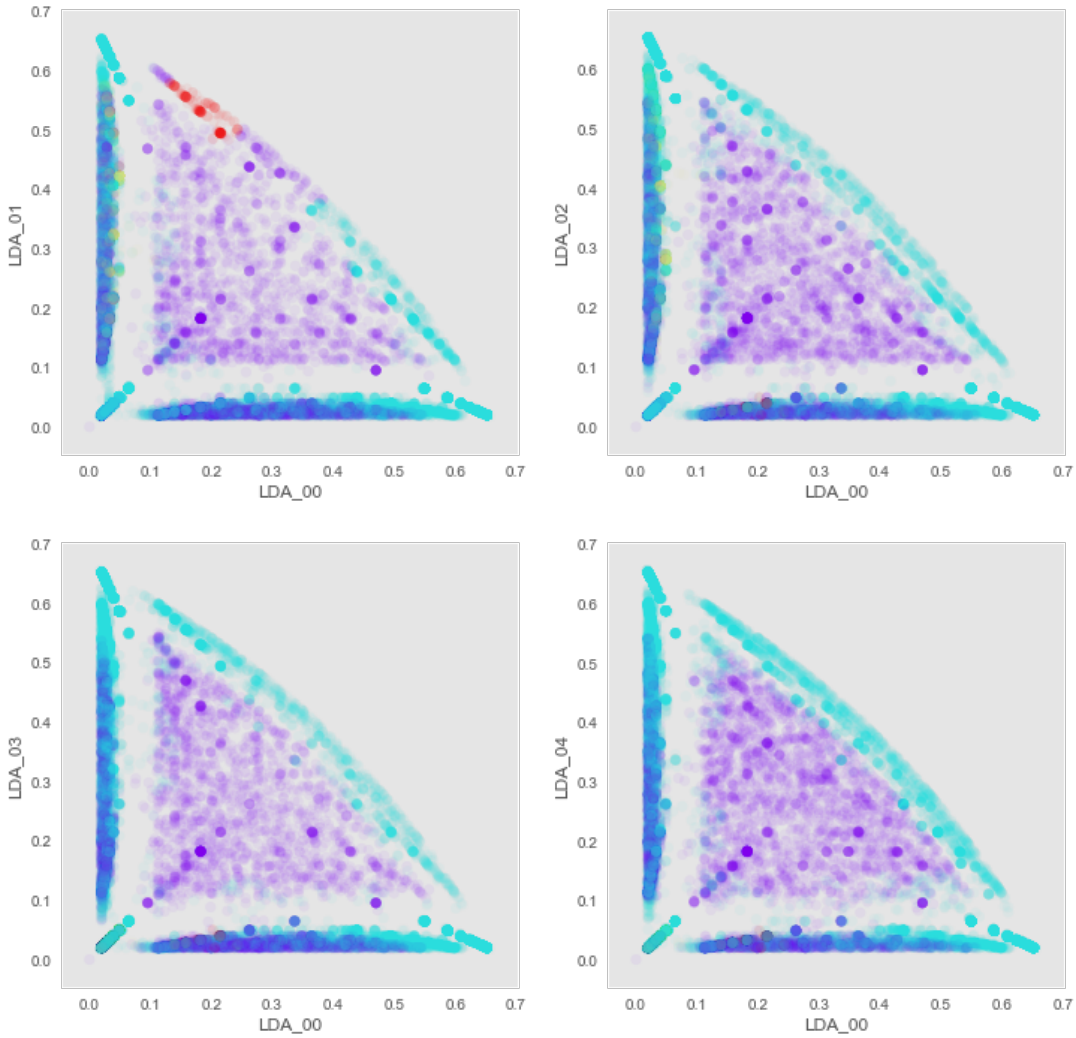

```
eps, min_pts, nclusters =  0.07 130 6
silhouette =  -0.340759262329
```



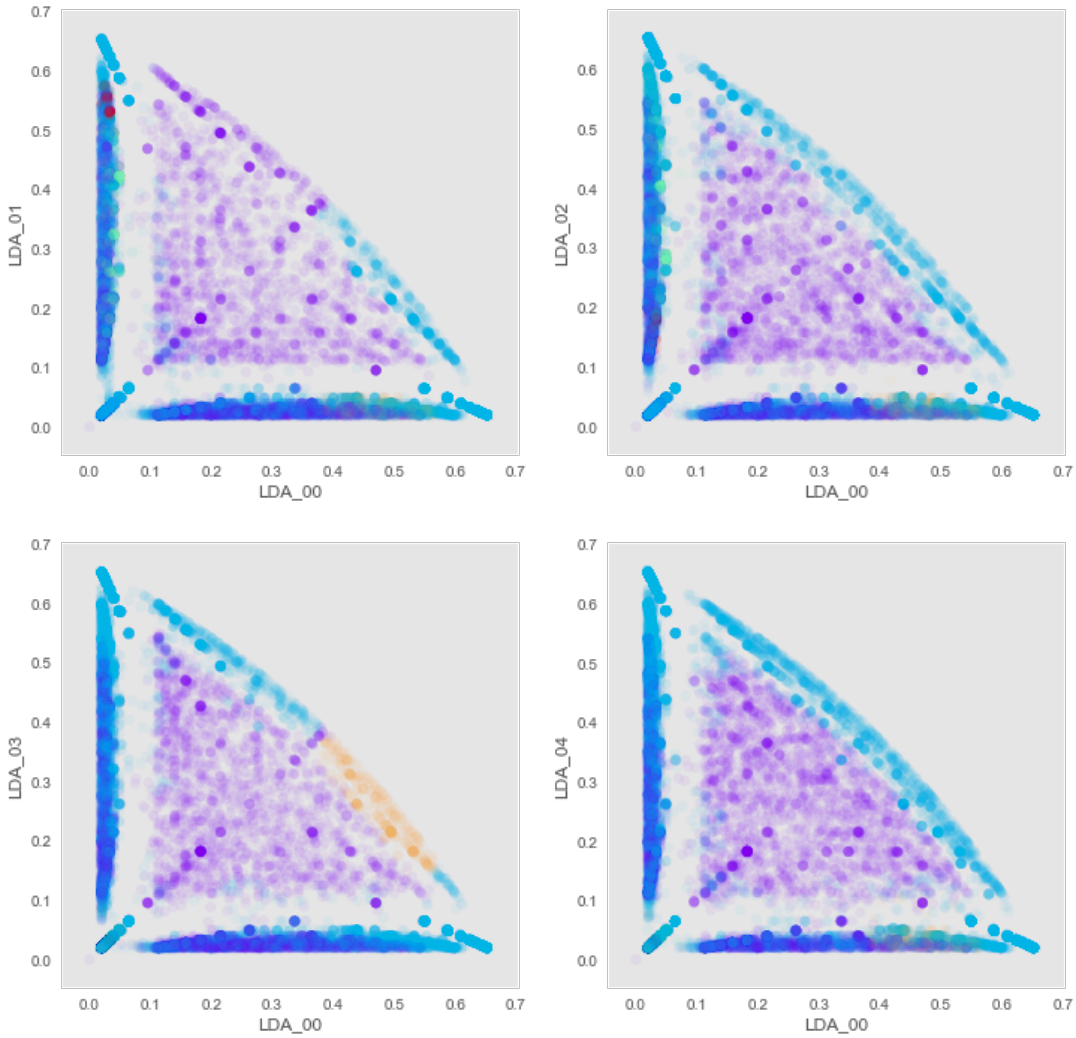

```
eps, min_pts, nclusters =  0.07 150 4
silhouette =  -0.148342782135
```




```
eps, min_pts, nclusters =  0.07 170 4
silhouette =  -0.152319241175
```




```
eps, min_pts, nclusters =  0.07 190 5
silhouette =  -0.242241225481
```




```
CPU times: user 6min 49s, sys: 10.8 s, total: 7min
Wall time: 6min 45s
Parser   : 124 ms
```

In [63]:
```
dbscan_tbl
```

Out[63]:

	model_name	n_clusters	epsilon	min_points	inertia	silhouette	process_time
1	DBScan - LDA features	153	0.02	10	0	-0.085596	5.4238
1	DBScan - LDA features	60	0.02	30	0	0.031906	4.1262
1	DBScan - LDA features	56	0.02	50	0	0.101427	3.9388
1	DBScan - LDA features	48	0.02	70	0	0.077119	3.9319
1	DBScan - LDA features	41	0.02	90	0	0.036830	3.7415
1	DBScan - LDA features	33	0.02	110	0	0.043183	3.6792
1	DBScan - LDA features	32	0.02	130	0	0.025074	3.6319
1	DBScan - LDA features	27	0.02	150	0	-0.012065	3.5670
1	DBScan - LDA features	19	0.02	170	0	-0.059899	3.4972
1	DBScan - LDA features	17	0.02	190	0	-0.071164	3.4778
1	DBScan - LDA features	124	0.03	10	0	-0.210358	5.2265
1	DBScan - LDA features	37	0.03	30	0	-0.021373	4.1868
1	DBScan - LDA features	27	0.03	50	0	0.145459	3.9858
1	DBScan - LDA features	29	0.03	70	0	0.138461	4.0710
1	DBScan - LDA features	32	0.03	90	0	0.162129	3.9581
1	DBScan - LDA features	28	0.03	110	0	0.146651	3.9046
1	DBScan - LDA features	27	0.03	130	0	0.170528	3.9858
1	DBScan - LDA features	24	0.03	150	0	0.118947	3.8420
1	DBScan - LDA features	19	0.03	170	0	0.109055	4.0999
1	DBScan - LDA features	18	0.03	190	0	0.079039	3.8014
1	DBScan - LDA features	13	0.05	10	0	-0.554975	4.0695

In [60]:
```
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ... - plot metrics across models for comparison
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

plt.figure(figsize=(16, 6));

# ... silhouette values

plt.subplot(131);
plt.scatter(dbscan_tbl['min_points'],
            dbscan_tbl['silhouette'],
            s = 40,
            linewidths = 1.0,
            marker = '^',
            edgecolors = 'black',
            alpha = 0.90);

plt.plot(dbscan_tbl['min_points'],
         dbscan_tbl['silhouette'])

plt.xlabel('min_points'), plt.ylabel('silhouette');
plt.grid();

# ... inertia values

plt.subplot(132);
plt.scatter(dbscan_tbl['min_points'],
            dbscan_tbl['n_clusters'],
            s = 40,
            linewidths = 1.0,
            marker = '^',
            edgecolors = 'black',
            alpha = 0.90);

plt.plot(dbscan_tbl['min_points'],
         dbscan_tbl['n_clusters'])

plt.xlabel('min_points'), plt.ylabel('n_clusters');
plt.grid();

# ... process time

plt.subplot(133);
plt.scatter(dbscan_tbl['min_points'],
            dbscan_tbl['process_time'],
            s = 40,
            linewidths = 1.0,
            marker = '^',
            edgecolors = 'black',
            alpha = 0.90);

#plt.plot(dbscan_tbl['n_clusters'],
#         dbscan_tbl['process_time'])

plt.xlabel('min_points'), plt.ylabel('process_time');
```

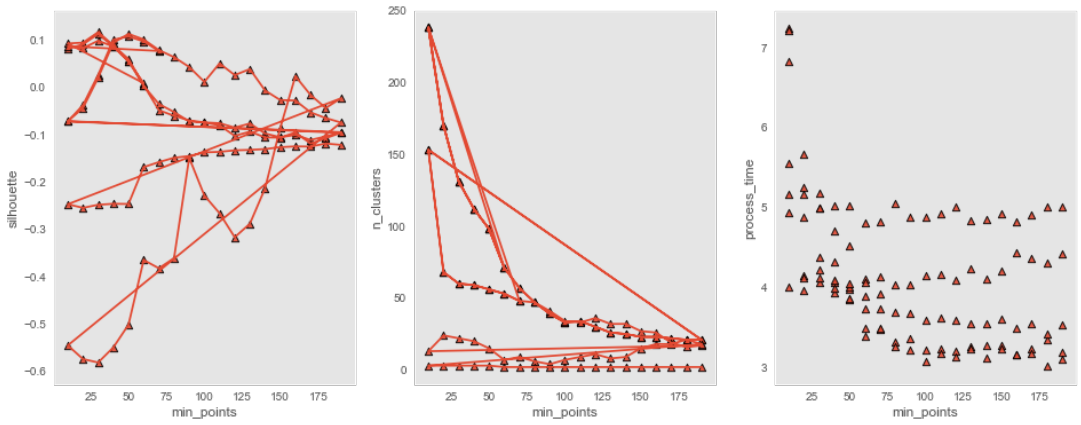


Table of Contents

Spectral Clustering

Spectral Clustering - 5 selected features

In [80]:
```
# set required variables for model comparison

spc_tbl = pd.DataFrame(columns = [
    'model_name',
    'n_clusters',
    'inertia',
    'silhouette',
    'process_time'])

i_index = []
i_index = 0

# preparation for cross validation and model comparison, each classifier is appended once mode
l is fit

models = []
```

In [81]:
```
from sklearn.cluster import SpectralClustering

# If a string, this may be one of
#  'nearest_neighbors', 'precomputed', 'rbf'
#  or one of the kernels supported by sklearn.metrics.pairwise_kernels

for n_clstr in range(2, 12):

    tic = time.clock()

    print ("n_clusters = ", n_clstr)

    X1 = df_cluster[['data_channel_n',
                     'ln_n_tokens_content',
                     'ln_num_hrefs',
                     'ln_num_imgs',
                     'ln_num_videos',]]

    X1 = X1.sample(frac = 0.1)

    spc = SpectralClustering(n_clusters = n_clstr,
                             affinity = 'nearest_neighbors')

    spc_labels = spc.fit_predict(X1)

    spc_silhouette = metrics.silhouette_score(X1,
                                              spc_labels,
                                              metric = 'euclidean',
                                              sample_size = 10000)
    print ("silhouette = ", spc_silhouette)

    toc =  time.clock()
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ... - save statistics for model comparison
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    exe_time = '{0:.4f}'.format(toc-tic)

    raw_data = {
    'model_name' : 'spc - LDA features',
    'n_clusters' : n_clstr,
    'inertia': 0,
    'silhouette': spc_silhouette,
    'process_time' : exe_time
    }

    df_tbl = pd.DataFrame(raw_data,
    columns = ['model_name', 'n_clusters', 'inertia', 'silhouette', 'process_time'],
    index = [i_index + 1])

    spc_tbl = spc_tbl.append(df_tbl)

# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ... - make some plots of clusters
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

```
n_clusters =  2

/home/mcdevitt/anaconda3/lib/python3.6/site-packages/sklearn/manifold/spectral_embedding_.p
y:234: UserWarning: Graph is not fully connected, spectral embedding may not work as expect
ed.
  warnings.warn("Graph is not fully connected, spectral embedding"

silhouette =  0.513948316463
```

Out[81]: `<matplotlib.figure.Figure at 0x7f4b925cd240>`

Out[81]: `<matplotlib.axes._subplots.AxesSubplot at 0x7f4b77124780>`

Out[81]: `<matplotlib.collections.PathCollection at 0x7f4b803ecd30>`

Out[81]:
```
(<matplotlib.text.Text at 0x7f4b77307dd8>,
 <matplotlib.text.Text at 0x7f4b91e4fac8>)
```

Out[81]: `<matplotlib.axes._subplots.AxesSubplot at 0x7f4b9014a9e8>`

Out[81]: `<matplotlib.collections.PathCollection at 0x7f4b777dc940>`

Out[81]:
```
(<matplotlib.text.Text at 0x7f4b91e93cf8>,
 <matplotlib.text.Text at 0x7f4b805a39b0>)
```

Out[81]: `<matplotlib.axes._subplots.AxesSubplot at 0x7f4b776da550>`

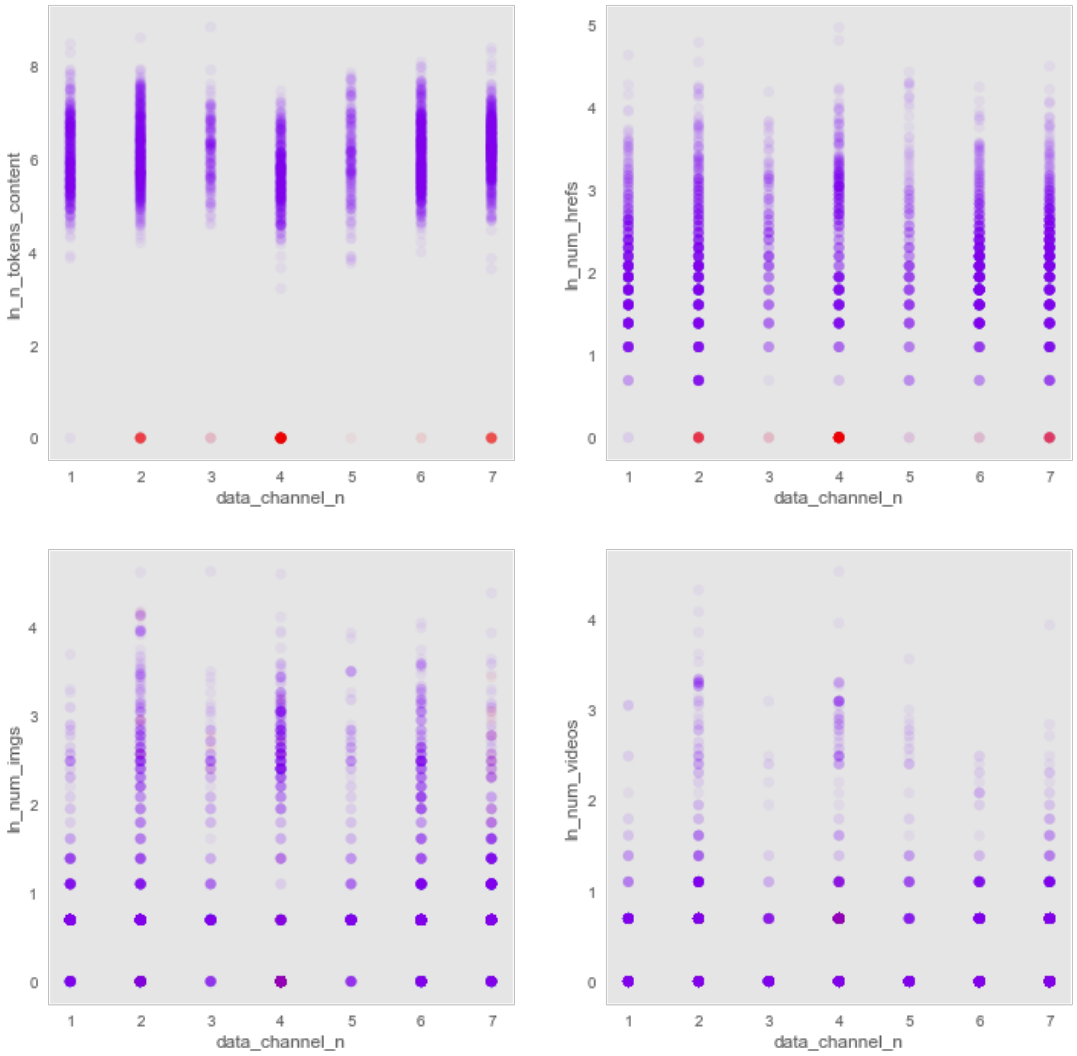
Out[81]: `<matplotlib.collections.PathCollection at 0x7f4b80166828>`

Out[81]:
```
(<matplotlib.text.Text at 0x7f4b77762dd8>,
 <matplotlib.text.Text at 0x7f4b91e2db38>)
```

Out[81]: `<matplotlib.axes._subplots.AxesSubplot at 0x7f4b80493c50>`

Out[81]: `<matplotlib.collections.PathCollection at 0x7f4b773e3470>`

Out[81]:
```
(<matplotlib.text.Text at 0x7f4b8029f828>,
 <matplotlib.text.Text at 0x7f4b9002df60>)
```




```
n_clusters =  3
silhouette =  0.469157506079
```

Out[81]: <matplotlib.figure.Figure at 0x7f4b925cd128>

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b77574ac8>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b91d32cc0>

Out[81]: (<matplotlib.text.Text at 0x7f4b92bbc518>,
 <matplotlib.text.Text at 0x7f4b9058f4a8>)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b776125c0>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b8016d1d0>

Out[81]: (<matplotlib.text.Text at 0x7f4b7743ff28>,
 <matplotlib.text.Text at 0x7f4b805575f8>)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b92a1c320>

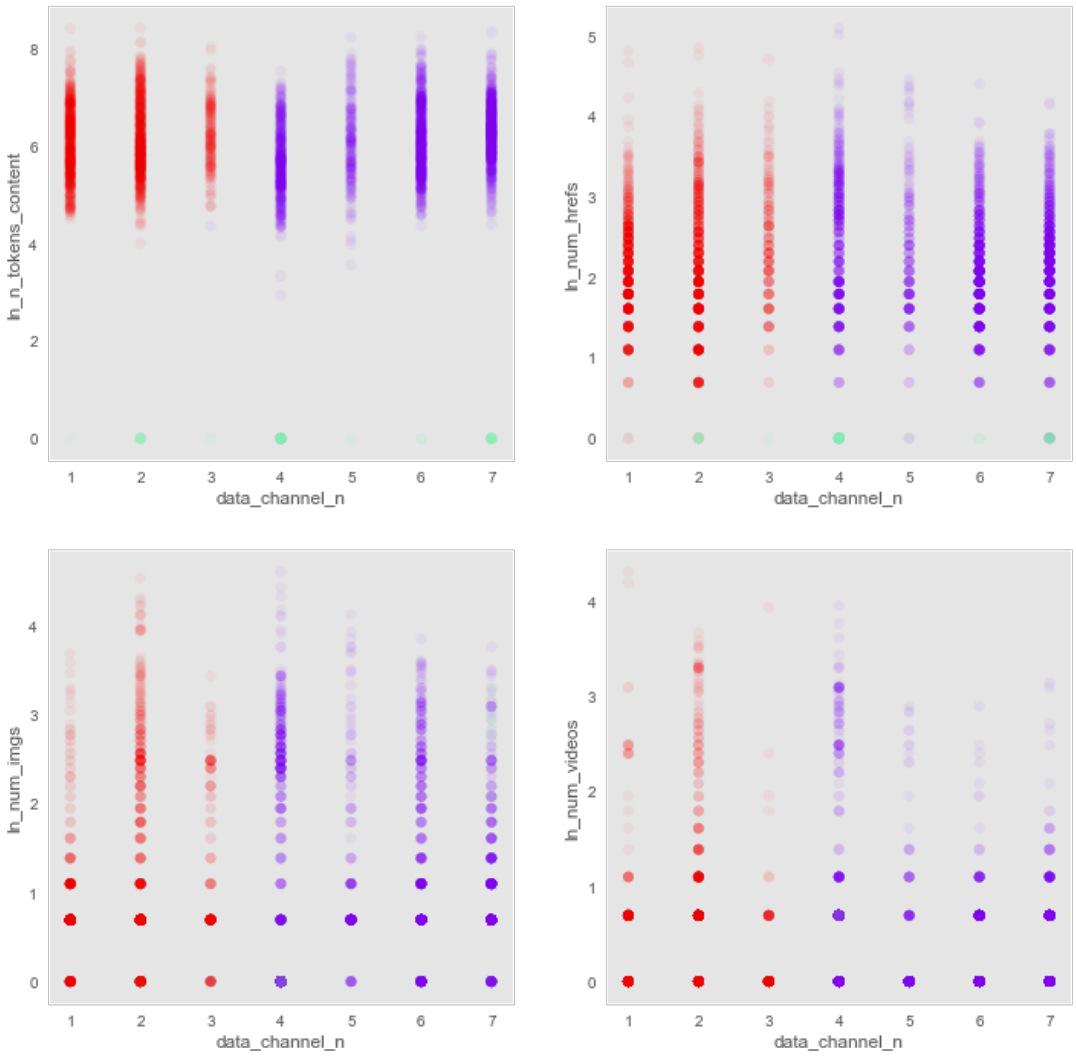
Out[81]: <matplotlib.collections.PathCollection at 0x7f4b7772b358>

Out[81]: (<matplotlib.text.Text at 0x7f4b774dec50>,
<matplotlib.text.Text at 0x7f4b900d4f98>)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b903016d8>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b91bf29e8>

Out[81]: (<matplotlib.text.Text at 0x7f4b91e67048>,
<matplotlib.text.Text at 0x7f4b91e67668>)




```
n_clusters =  4
silhouette =  0.277642447465
```

Out[81]: <matplotlib.figure.Figure at 0x7f4b905339e8>

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b7748a780>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b77771cc0>

Out[81]: (<matplotlib.text.Text at 0x7f4b777e4048>,
<matplotlib.text.Text at 0x7f4b91bd9400>)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b775d6518>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b92c60588>

Out[81]: (<matplotlib.text.Text at 0x7f4b775d88d0>,
<matplotlib.text.Text at 0x7f4b773c2b00>)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b92c60ba8>

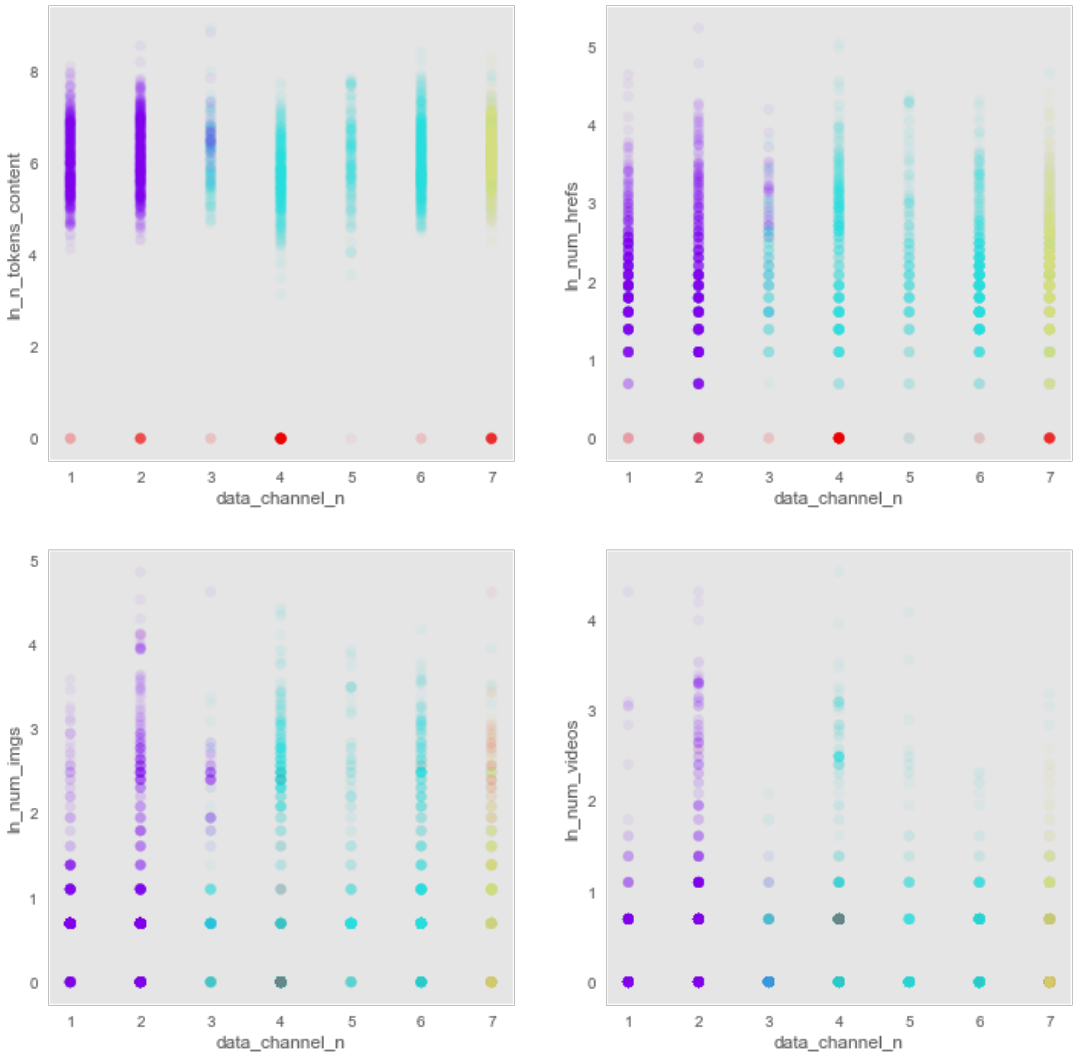
Out[81]: <matplotlib.collections.PathCollection at 0x7f4b92cc5a90>

Out[81]: (<matplotlib.text.Text at 0x7f4b91f520b8>,
<matplotlib.text.Text at 0x7f4b91f55438>)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b91f557f0>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b91b7b208>

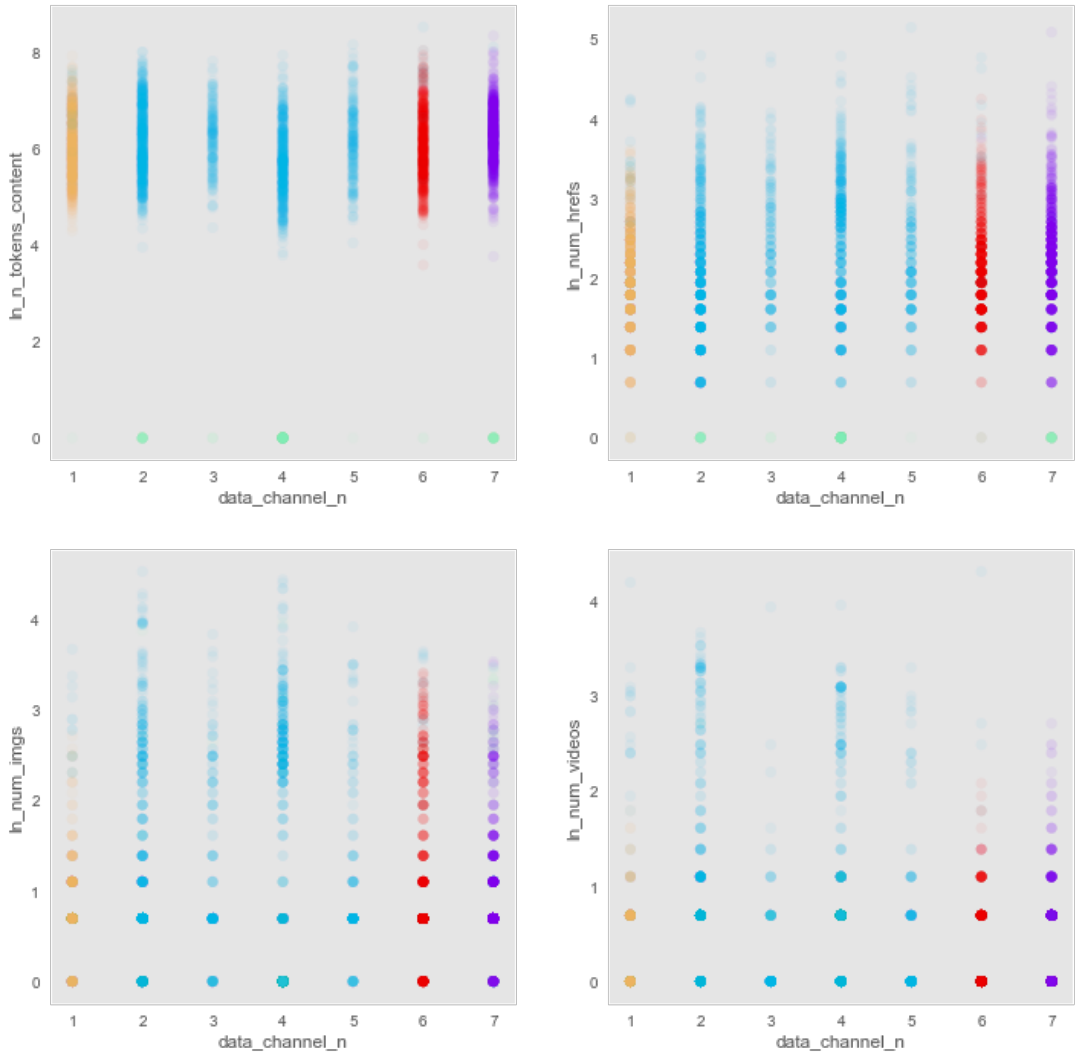
Out[81]: (<matplotlib.text.Text at 0x7f4b92cc5eb8>,
<matplotlib.text.Text at 0x7f4b92ce0748>)




```
n_clusters =  5
silhouette =  0.137926081974
```

Out[81]: <matplotlib.figure.Figure at 0x7f4b77325a58>

```
Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b804f32b0>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b91bfc2e8>

Out[81]: (<matplotlib.text.Text at 0x7f4b91ea0e10>,
          <matplotlib.text.Text at 0x7f4b91dc70f0>)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b91cb5fd0>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b91dffb00>

Out[81]: (<matplotlib.text.Text at 0x7f4b91ccae48>,
          <matplotlib.text.Text at 0x7f4b91cb31d0>)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b91add278>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b91ac6fd0>

Out[81]: (<matplotlib.text.Text at 0x7f4b91adf630>,
          <matplotlib.text.Text at 0x7f4b91aeb860>)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b91afdb00>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b92cccda0>

Out[81]: (<matplotlib.text.Text at 0x7f4b91c98588>,
          <matplotlib.text.Text at 0x7f4b91dc37f0>)
```




```
n_clusters =  6
silhouette =  0.166255692316
```

Out[81]: `<matplotlib.figure.Figure at 0x7f4b7718a4a8>`

Out[81]: `<matplotlib.axes._subplots.AxesSubplot at 0x7f4b91dc7048>`

Out[81]: `<matplotlib.collections.PathCollection at 0x7f4b7760e128>`

Out[81]:
```
(<matplotlib.text.Text at 0x7f4b91ba94a8>,
 <matplotlib.text.Text at 0x7f4b91bc7cf8>)
```

Out[81]: `<matplotlib.axes._subplots.AxesSubplot at 0x7f4b804172e8>`

Out[81]: `<matplotlib.collections.PathCollection at 0x7f4b900d4048>`

Out[81]:
```
(<matplotlib.text.Text at 0x7f4b775a4550>,
 <matplotlib.text.Text at 0x7f4b92cc6940>)
```

Out[81]: `<matplotlib.axes._subplots.AxesSubplot at 0x7f4b92c58d68>`

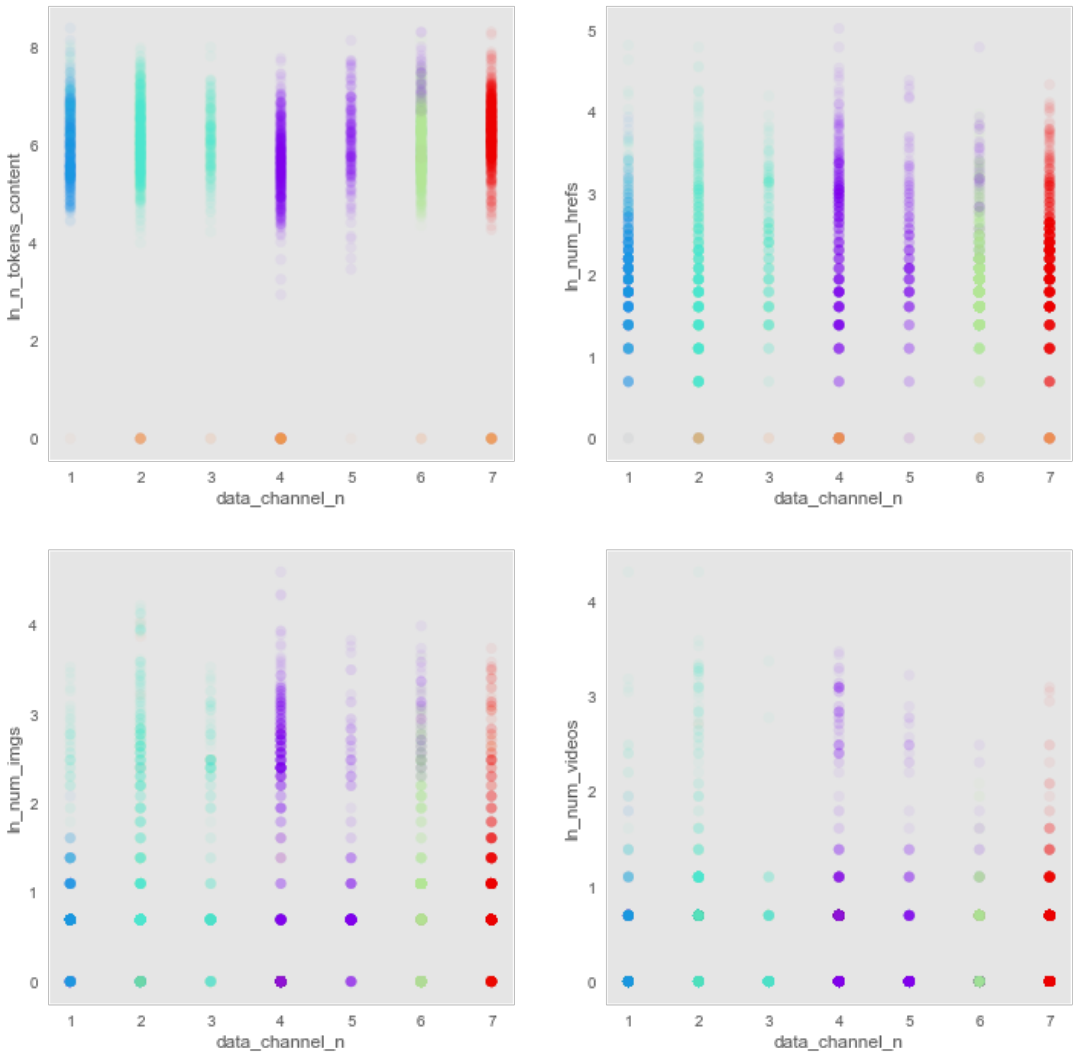
Out[81]: `<matplotlib.collections.PathCollection at 0x7f4b805a0278>`

Out[81]: (<matplotlib.text.Text at 0x7f4b9037a630>,
<matplotlib.text.Text at 0x7f4b91f5d198>)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b7731df98>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b92c47668>

Out[81]: (<matplotlib.text.Text at 0x7f4b8002ee80>,
<matplotlib.text.Text at 0x7f4b775b9470>)



n_clusters = 7
silhouette = 0.12099864669

Out[81]: <matplotlib.figure.Figure at 0x7f4b91ec15c0>

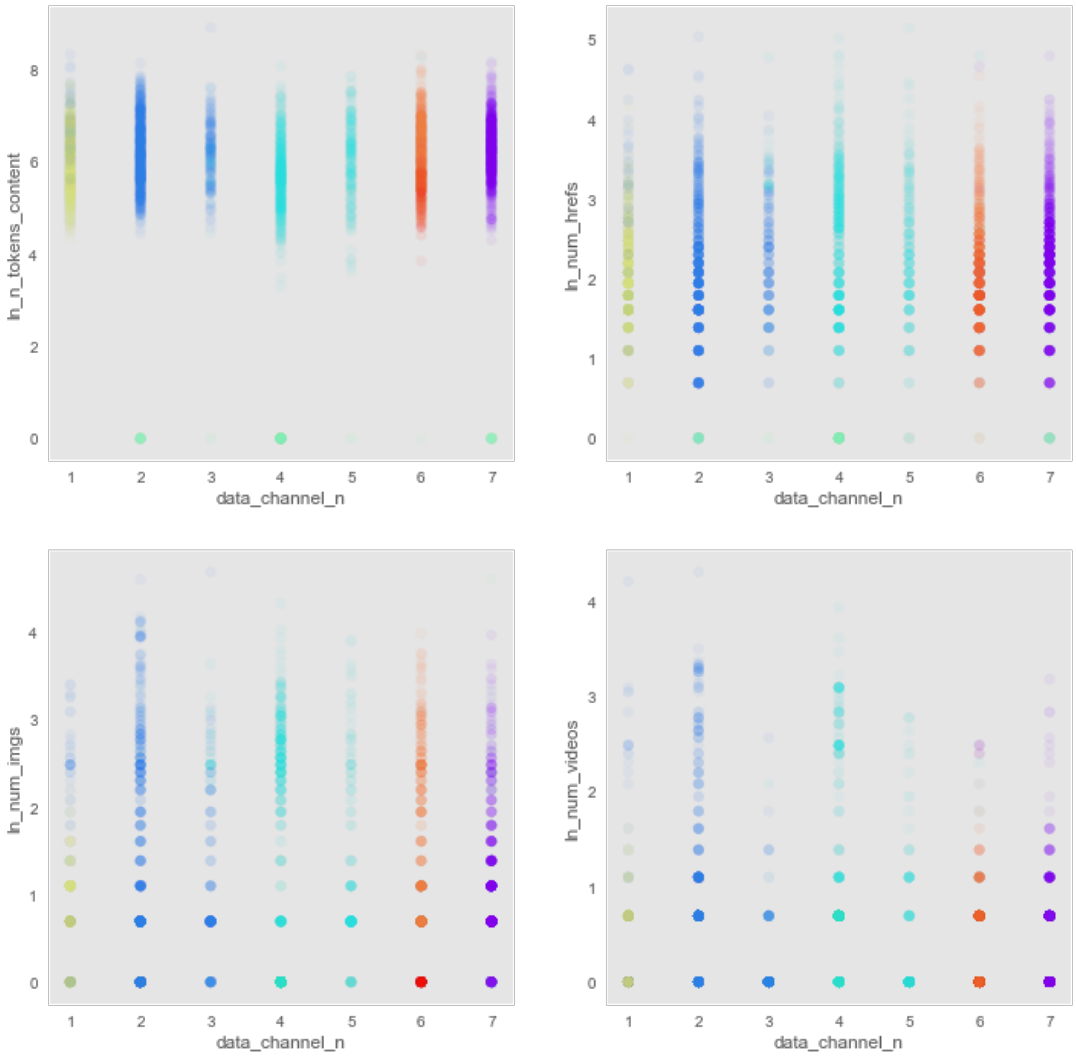
Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b91b94be0>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b7739b470>

Out[81]: (<matplotlib.text.Text at 0x7f4b91d71c88>,
<matplotlib.text.Text at 0x7f4b926079b0>)

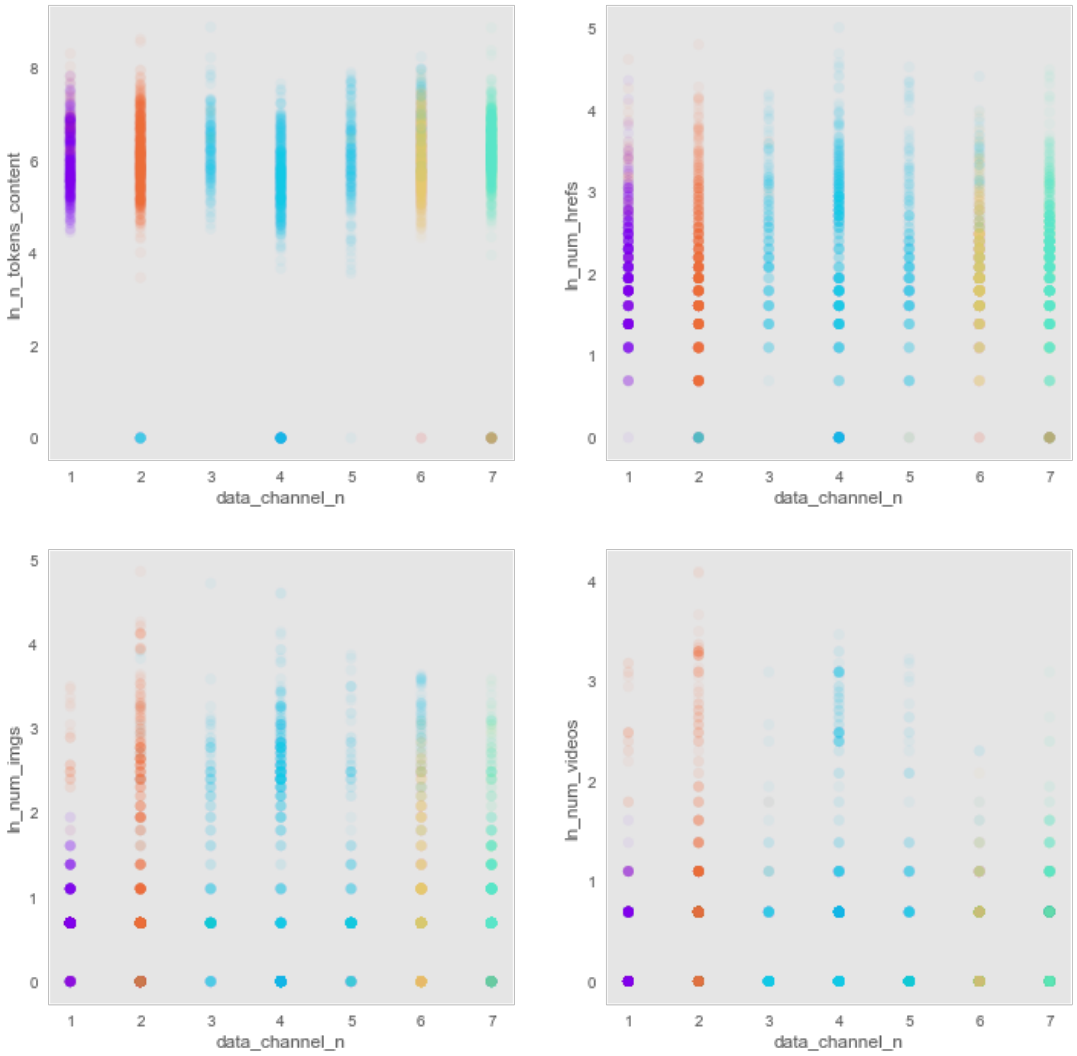
Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b802d3550>

```
Out[81]: <matplotlib.collections.PathCollection at 0x7f4b80238748>

Out[81]: (<matplotlib.text.Text at 0x7f4b777625c0>,
          <matplotlib.text.Text at 0x7f4b776f27f0>)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b8045e4e0>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b92c46dd8>

Out[81]: (<matplotlib.text.Text at 0x7f4b770f64e0>,
          <matplotlib.text.Text at 0x7f4b91e4f240>)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b7753d358>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b7742f748>

Out[81]: (<matplotlib.text.Text at 0x7f4b91e4f860>,
          <matplotlib.text.Text at 0x7f4b91b8cc88>)
```




```
n_clusters =  8
silhouette =  0.104695361366

Out[81]: <matplotlib.figure.Figure at 0x7f4b904df048>
```

```
Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b91ba9d68>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b91b76f98>

Out[81]: (<matplotlib.text.Text at 0x7f4b926aad68>,
          <matplotlib.text.Text at 0x7f4b90046fd0>)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b80027f28>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b8029db70>

Out[81]: (<matplotlib.text.Text at 0x7f4b80190b38>,
          <matplotlib.text.Text at 0x7f4b804f3630>)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b91e24320>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b804d8e80>

Out[81]: (<matplotlib.text.Text at 0x7f4b8040eeb8>,
          <matplotlib.text.Text at 0x7f4b91f25d30>)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b774b6080>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b92cc0ba8>

Out[81]: (<matplotlib.text.Text at 0x7f4b91f251d0>,
          <matplotlib.text.Text at 0x7f4b804d1908>)
```




```
n_clusters =  9
silhouette =  0.143887177055
```

Out[81]: <matplotlib.figure.Figure at 0x7f4b772e5358>

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b770e4278>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b77414d68>

Out[81]: (<matplotlib.text.Text at 0x7f4b804256d8>,
 <matplotlib.text.Text at 0x7f4b92c32a58>)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b9032b7b8>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b7743fb00>

Out[81]: (<matplotlib.text.Text at 0x7f4b77385cf8>,
 <matplotlib.text.Text at 0x7f4b771562b0>)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b80238c18>

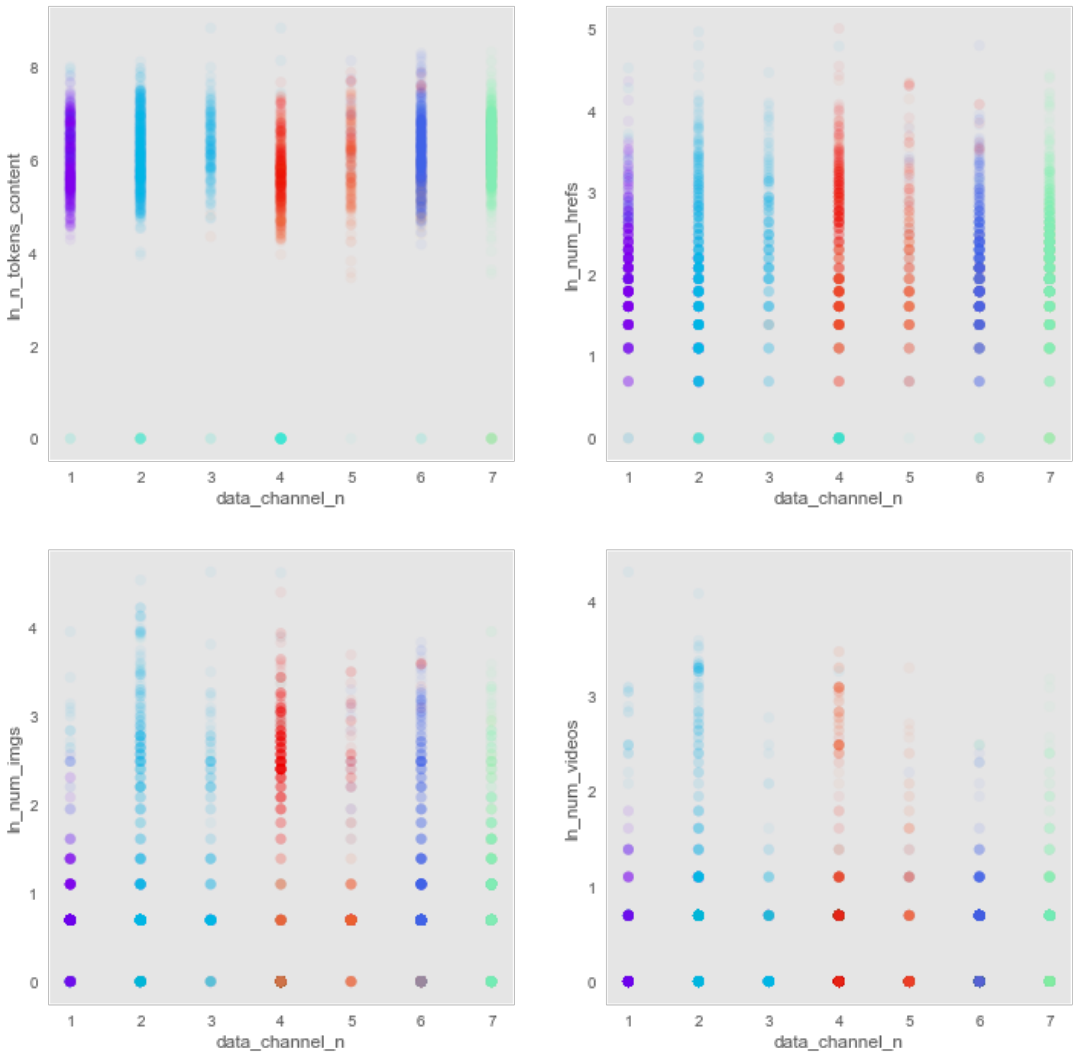
Out[81]: <matplotlib.collections.PathCollection at 0x7f4b8056dd68>

Out[81]: (<matplotlib.text.Text at 0x7f4b773f4278>,
<matplotlib.text.Text at 0x7f4b92bb82b0>)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b805a32b0>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b77311668>

Out[81]: (<matplotlib.text.Text at 0x7f4b92c50080>,
<matplotlib.text.Text at 0x7f4b92c507f0>)




```
n_clusters =  10
silhouette =  0.00688832942488
```

Out[81]: <matplotlib.figure.Figure at 0x7f4b775360b8>

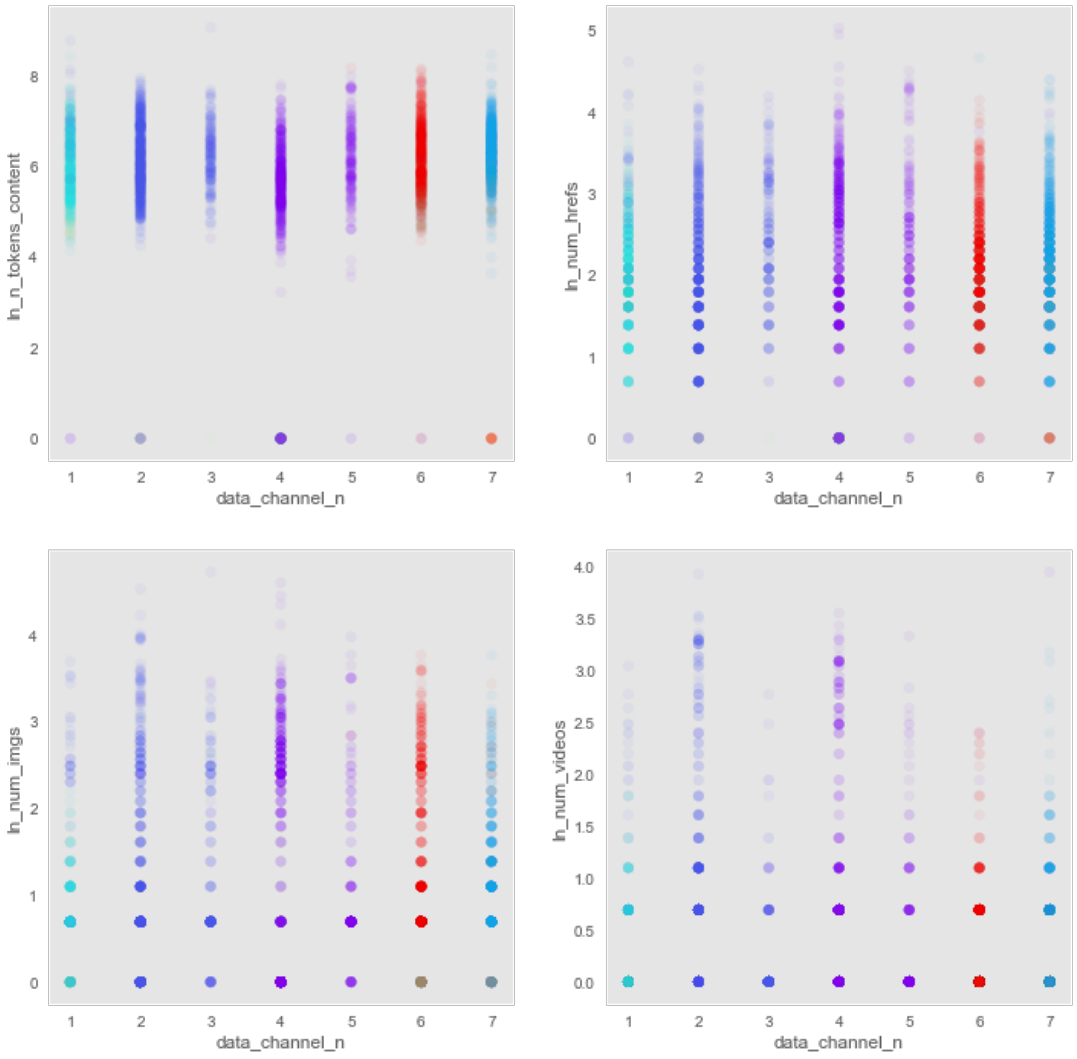
Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b905b5cc0>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b803a80b8>

Out[81]: (<matplotlib.text.Text at 0x7f4b802d4588>,
<matplotlib.text.Text at 0x7f4b904299e8>)

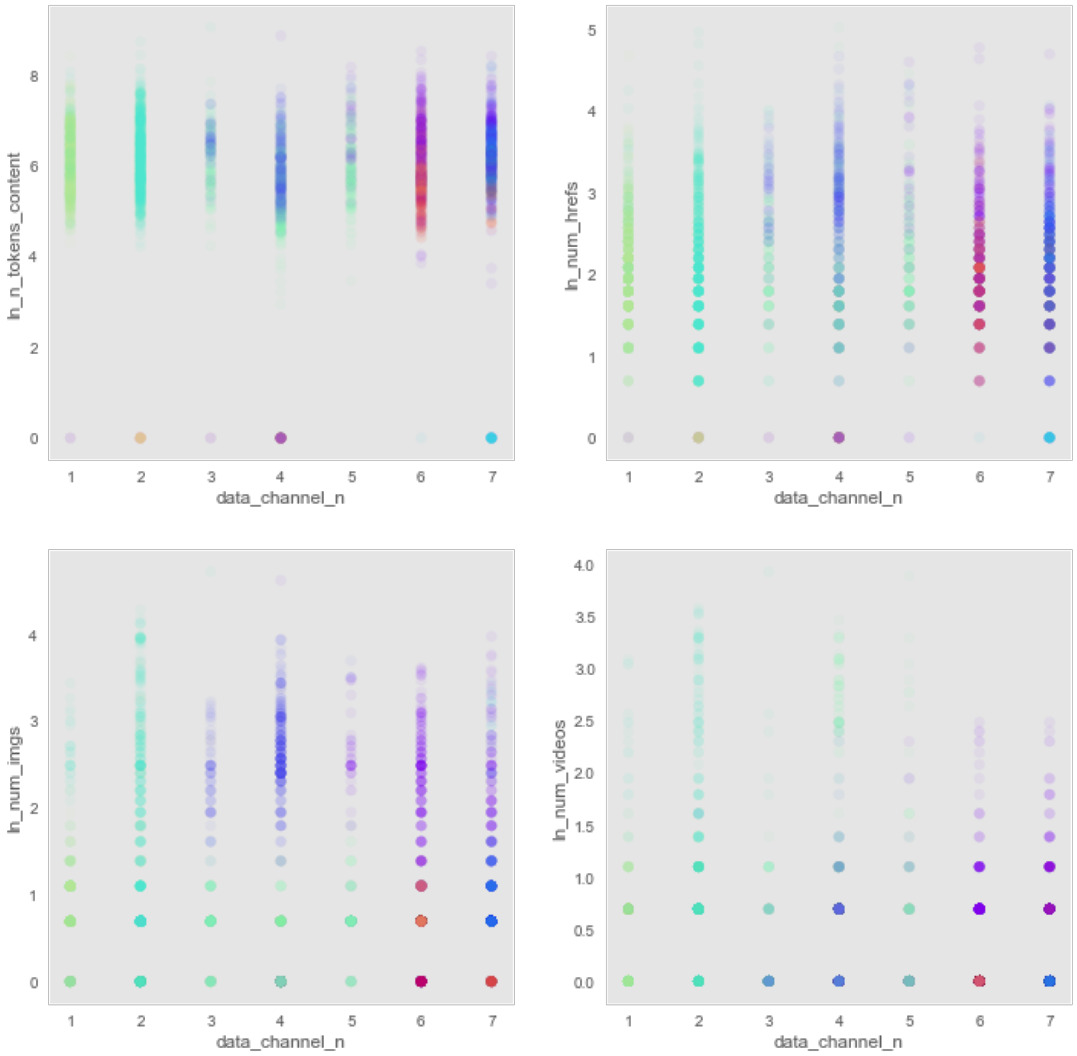
Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4bc945dac8>

```
Out[81]: <matplotlib.collections.PathCollection at 0x7f4b924ff9e8>

Out[81]: (<matplotlib.text.Text at 0x7f4b775b9ba8>,
          <matplotlib.text.Text at 0x7f4b7772b780>)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b92535c50>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b900a5710>

Out[81]: (<matplotlib.text.Text at 0x7f4b900a4940>,
          <matplotlib.text.Text at 0x7f4b805da748>)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b90046978>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b90432b70>

Out[81]: (<matplotlib.text.Text at 0x7f4b8029fcc0>,
          <matplotlib.text.Text at 0x7f4b91b9fcf8>)
```




```
n_clusters =  11
silhouette =  0.107343619097

Out[81]: <matplotlib.figure.Figure at 0x7f4b770badd8>
```

```
Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b770f6a58>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b92bbd1d0>

Out[81]: (<matplotlib.text.Text at 0x7f4b92c47cf8>,
          <matplotlib.text.Text at 0x7f4b92645080>)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b92bbda58>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b92cadac8>

Out[81]: (<matplotlib.text.Text at 0x7f4b92bbae10>,
          <matplotlib.text.Text at 0x7f4b91f7f080>)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b92ccf4a8>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b91eb9f60>

Out[81]: (<matplotlib.text.Text at 0x7f4b92cc35f8>,
          <matplotlib.text.Text at 0x7f4b91ee1828>)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4b91efee10>

Out[81]: <matplotlib.collections.PathCollection at 0x7f4b91bbbdd8>

Out[81]: (<matplotlib.text.Text at 0x7f4b91ed5470>,
          <matplotlib.text.Text at 0x7f4b91ed5cf8>)
```

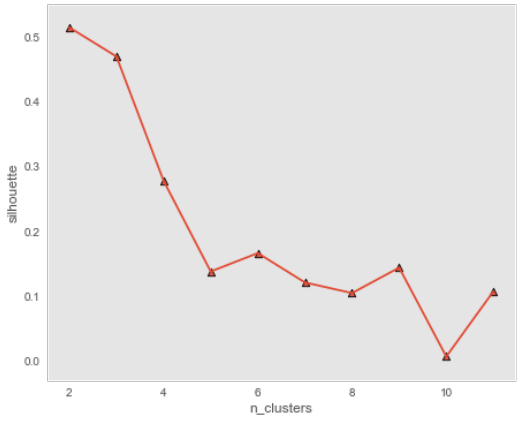


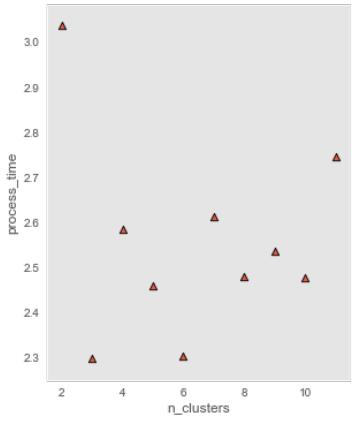
Out[81]:

	model_name	n_clusters	inertia	silhouette	process_time
1	spc - LDA features	2	0	0.513948	3.0362
1	spc - LDA features	3	0	0.469158	2.2964
1	spc - LDA features	4	0	0.277642	2.5837
1	spc - LDA features	5	0	0.137926	2.4575
1	spc - LDA features	6	0	0.166256	2.3021
1	spc - LDA features	7	0	0.120999	2.6126
1	spc - LDA features	8	0	0.104695	2.4773
1	spc - LDA features	9	0	0.143887	2.5353
1	spc - LDA features	10	0	0.006888	2.4749
1	spc - LDA features	11	0	0.107344	2.7446

In [82]:

```
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ... - plot metrics across models for comparison
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

plt.figure(figsize=(16, 6));

# ... silhouette values

plt.subplot(121);
plt.scatter(spc_tbl['n_clusters'],
            spc_tbl['silhouette'],
            s = 40,
            linewidths = 1.0,
            marker = '^',
            edgecolors = 'black',
            alpha = 0.90);

plt.plot(spc_tbl['n_clusters'],
         spc_tbl['silhouette'])

plt.xlabel('n_clusters'), plt.ylabel('silhouette');
plt.grid();


# ... process time

plt.subplot(133);
plt.scatter(spc_tbl['n_clusters'],
            spc_tbl['process_time'],
            s = 40,
            linewidths = 1.0,
            marker = '^',
            edgecolors = 'black',
            alpha = 0.90);

#plt.plot(spc_tbl['n_clusters'],
#         spc_tbl['process_time'])

plt.xlabel('n_clusters'), plt.ylabel('process_time');
plt.grid();


plt.show();
```





t-SNE

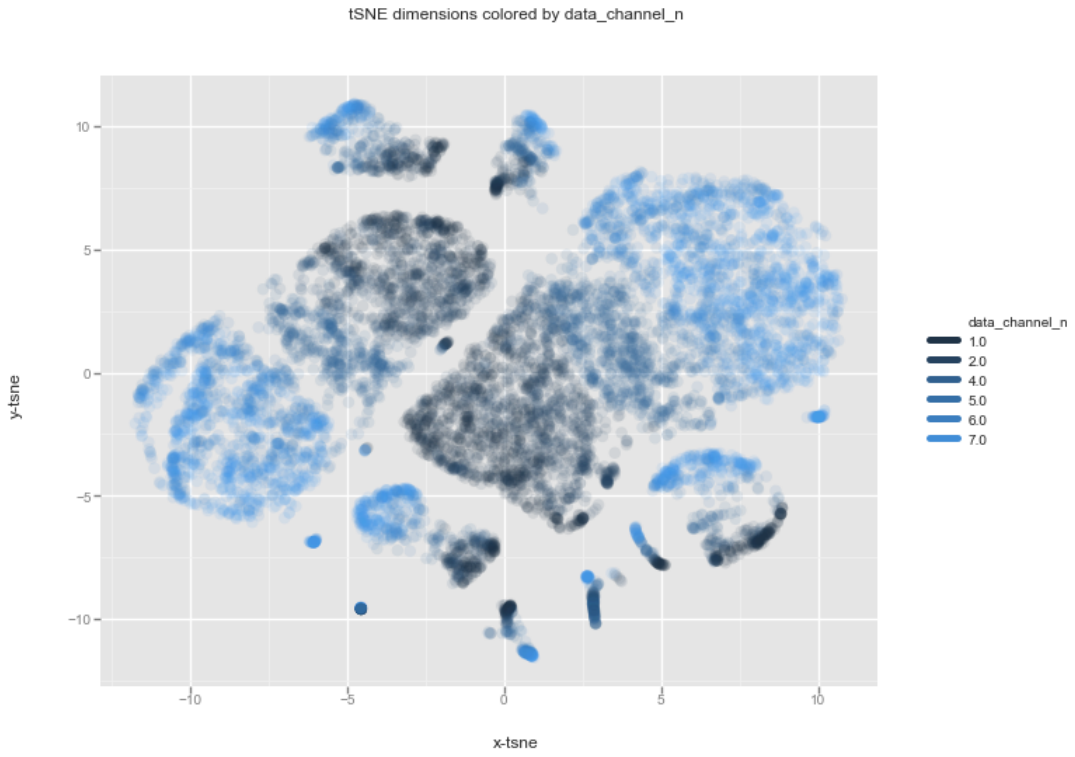
In [63]:
```
from sklearn.manifold import TSNE

X1 = df_cluster
X1 = X1.sample(frac = 0.25)

tic = time.clock()

tsne = TSNE(n_components = 2, verbose = 1, perplexity = 40, n_iter = 300)

tsne_results = tsne.fit_transform(X1)

toc = time.clock()
print (toc - tic)
```

```
[t-SNE] Computing 121 nearest neighbors...
[t-SNE] Indexed 9911 samples in 0.031s...
[t-SNE] Computed neighbors for 9911 samples in 8.457s...
[t-SNE] Computed conditional probabilities for sample 1000 / 9911
[t-SNE] Computed conditional probabilities for sample 2000 / 9911
[t-SNE] Computed conditional probabilities for sample 3000 / 9911
[t-SNE] Computed conditional probabilities for sample 4000 / 9911
[t-SNE] Computed conditional probabilities for sample 5000 / 9911
[t-SNE] Computed conditional probabilities for sample 6000 / 9911
[t-SNE] Computed conditional probabilities for sample 7000 / 9911
[t-SNE] Computed conditional probabilities for sample 8000 / 9911
[t-SNE] Computed conditional probabilities for sample 9000 / 9911
[t-SNE] Computed conditional probabilities for sample 9911 / 9911
[t-SNE] Mean sigma: 1.357765
[t-SNE] KL divergence after 250 iterations with early exaggeration: 75.686134
[t-SNE] Error after 300 iterations: 2.626567
154.91459400000002
```

In [64]:
```
tsne_results
```

Out[64]:
```
array([[ 2.31409001,  3.12258196],
       [-6.1454215 , -2.08470392],
       [ 3.25107622,  2.7788651 ],
       ...,
       [-1.28354645,  2.38217044],
       [ 1.27252066, -2.97045398],
       [ 1.62990141,  2.55411768]], dtype=float32)
```

In [65]:
```
from ggplot import *

df_tsne = X1.copy()
df_tsne['x-tsne'] = tsne_results[:,0]
df_tsne['y-tsne'] = tsne_results[:,1]

chart = ggplot( df_tsne,
        aes(x = 'x-tsne', y = 'y-tsne', color = 'data_channel_n') ) \
        + geom_point(size = 70, alpha = 0.1) \
        + ggtitle("tSNE dimensions colored by data_channel_n")
chart
```



Out[65]:
```
<ggplot: (-9223363271325156064)>
```

Table of Contents

end of file

In [ ]:

In [ ]: