Matching Harry Potter Spells to their Definitions.

In [1]:
```
#lauch jupyter from within the hpspells directory.
import hp_spells as hp
import numpy as np
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
sns.set(style="white", color_codes=True)
%matplotlib inline
```

In [2]:
```
model = hp.load_vectors("../../vectors/GoogleNews-vectors-negative300.bin",
True)
```

```
Loading:  ../../vectors/GoogleNews-vectors-negative300.bin
Loaded:  ../../vectors/GoogleNews-vectors-negative300.bin
```

Run the experiment

In [4]:
```
scores, syn_experiments, average, avg_cos_dists, iterationCount = hp.run_ex
periment(model, 10)
```

```
--------------- 0 ---------------
Num of spells that feature in definition:  27
Percentage:  30.0 %
Average Cosine-simalarity: 0.721283449825
Num of spells which are a synonyms:  7
--------------- 1 ---------------

/home/james/anaconda2/lib/python2.7/urllib.py:1299: UnicodeWarning: Unicode
equal comparison failed to convert both arguments to Unicode - interpreting
them as being unequal
  return ''.join(map(quoter, s))

Num of spells that feature in definition:  27
Percentage:  30.0 %
Average Cosine-simalarity: 0.733662894371
Num of spells which are a synonyms:  4
--------------- 2 ---------------
Num of spells that feature in definition:  34
Percentage:  37.7777777778 %
Average Cosine-simalarity: 0.715381949384
Num of spells which are a synonyms:  11
--------------- 3 ---------------
Num of spells that feature in definition:  28
Percentage:  31.1111111111 %
Average Cosine-simalarity: 0.701601432416
Num of spells which are a synonyms:  12
--------------- 4 ---------------
Num of spells that feature in definition:  30
Percentage:  33.3333333333 %
Average Cosine-simalarity: 0.6943964363
Num of spells which are a synonyms:  12
--------------- 5 ---------------
Num of spells that feature in definition:  20
Percentage:  22.2222222222 %
Average Cosine-simalarity: 0.697070274259
Num of spells which are a synonyms:  5
--------------- 6 ---------------
Num of spells that feature in definition:  38
Percentage:  42.2222222222 %
Average Cosine-simalarity: 0.696585639164
Num of spells which are a synonyms:  11
--------------- 7 ---------------
Num of spells that feature in definition:  36
Percentage:  40.0 %
Average Cosine-simalarity: 0.699209905192
Num of spells which are a synonyms:  8
--------------- 8 ---------------
Num of spells that feature in definition:  29
Percentage:  32.2222222222 %
Average Cosine-simalarity: 0.700530696648
Num of spells which are a synonyms:  9
--------------- 9 ---------------
Num of spells that feature in definition:  28
Percentage:  31.1111111111 %
Average Cosine-simalarity: 0.699924847048
Num of spells which are a synonyms:  8
```

Producing Graphs from the data

- list 1
- list2

In [6]:
```
results = pd.DataFrame({'scores': scores, 'similarity' : avg_cos_dists})
```

In [29]:
```
%store results
```

```
Stored 'results' (DataFrame)
```

In [2]:
```
%store -r #this is used to call the stored results.
```

```
no stored variable #this
no stored variable is used to call the stored results.

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-2-050411972447> in <module>()
      1 get_ipython().magic(u'store -r #this is used to call the stored res
ults.')
----> 2 print(results)

NameError: name 'results' is not defined
```

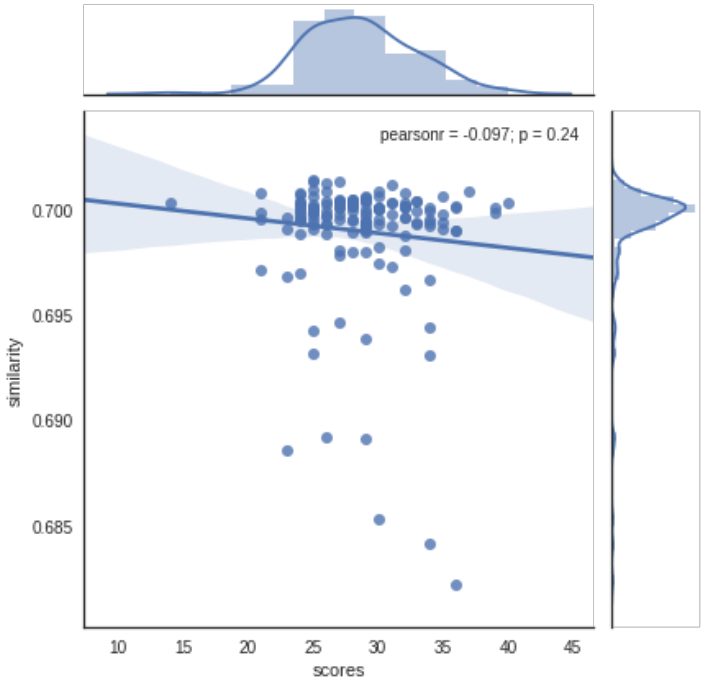
Violin plot of just the similarity

In [ ]:
```
g = sns.violinplot(x=results["similarity"])
```

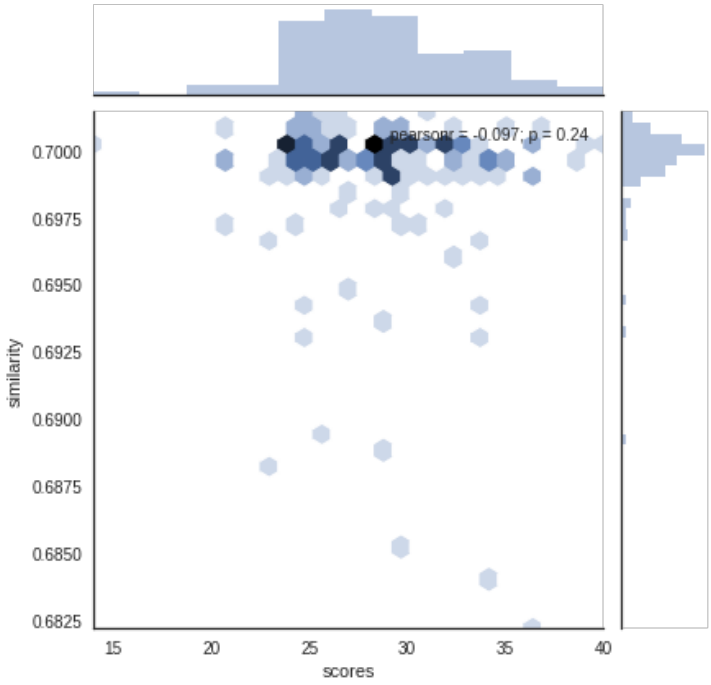
below is a joint plot u ysed this because....

Joint Plots

In [22]:
```
graph = sns.jointplot(x="scores", y="similarity", data=results, kind="reg")
```

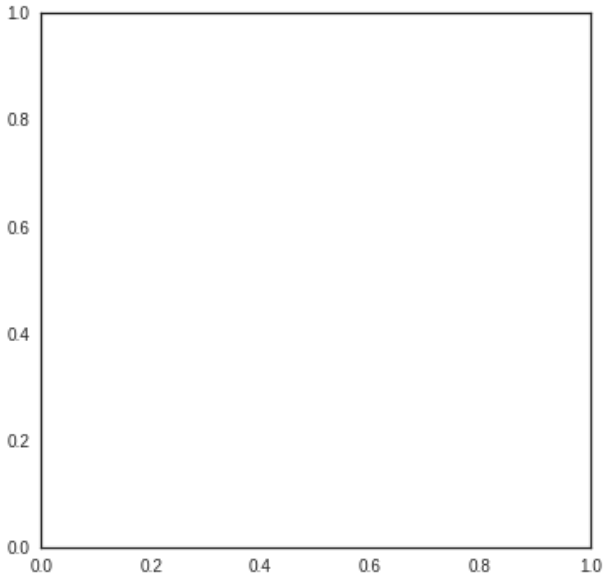


In [25]:
```
hex_graph = sns.jointplot(x="scores", y="similarity", data=results, kind="hex")
```



In [27]:
```
mean, cov = [0, 1], [(1, .5), (.5, 1)]
data = np.random.multivariate_normal(mean, cov, 200)
df = pd.DataFrame(data, columns=["x", "y"])

f, ax = plt.subplots(figsize=(6, 6))
cmap = sns.cubehelix_palette(as_cmap=True, dark=0, light=1, reverse=True)
sns.kdeplot(results.similarity, df.y, cmap=cmap, n_levels=60, shade=True);
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-27-15bd3771de79> in <module>()
      5 f, ax = plt.subplots(figsize=(6, 6))
      6 cmap = sns.cubehelix_palette(as_cmap=True, dark=0, light=1, reverse
=True)
----> 7 sns.kdeplot(results.similarity, df.y, cmap=cmap, n_levels=60, shade
=True);

/home/james/anaconda2/lib/python2.7/site-packages/seaborn/distributions.pyc
in kdeplot(data, data2, shade, vertical, kernel, bw, gridsize, cut, clip, l
egend, cumulative, shade_lowest, ax, **kwargs)
    598         ax = _bivariate_kdeplot(x, y, shade, shade_lowest,
    599                                 kernel, bw, gridsize, cut, clip, le
gend,
--> 600                                 ax, **kwargs)
    601     else:
    602         ax = _univariate_kdeplot(data, shade, vertical, kernel, bw,

/home/james/anaconda2/lib/python2.7/site-packages/seaborn/distributions.pyc
in _bivariate_kdeplot(x, y, filled, fill_lowest, kernel, bw, gridsize, cut,
clip, axlabel, ax, **kwargs)
    364     # Calculate the KDE
    365     if _has_statsmodels:
--> 366         xx, yy, z = _statsmodels_bivariate_kde(x, y, bw, gridsize,
cut, clip)
    367     else:
    368         xx, yy, z = _scipy_bivariate_kde(x, y, bw, gridsize, cut, c
lip)

/home/james/anaconda2/lib/python2.7/site-packages/seaborn/distributions.pyc
in _statsmodels_bivariate_kde(x, y, bw, gridsize, cut, clip)
    410         y = y.values
    411
--> 412     kde = smnp.KDEMultivariate([x, y], "cc", bw)
    413     x_support = _kde_support(x, kde.bw[0], gridsize, cut, clip[0])
    414     y_support = _kde_support(y, kde.bw[1], gridsize, cut, clip[1])

/home/james/anaconda2/lib/python2.7/site-packages/statsmodels/nonparametric
/kernel_density.pyc in __init__(self, data, var_type, bw, defaults)
    109         self.nobs, self.k_vars = np.shape(self.data)
    110         if self.nobs <= self.k_vars:
--> 111             raise ValueError("The number of observations must be la
rger " \
    112                              "than the number of variables.")
    113

ValueError: The number of observations must be larger than the number of va
riables.
```


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0