In [1]:
```
from datascience import *
import numpy as np
import re
import gensim

from collections import Counter

import logging
logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s'
, level=logging.ERROR)
logging.root.level = logging.CRITICAL

import warnings
warnings.filterwarnings("ignore",category=DeprecationWarning)

# direct plots to appear within the cell, and set their style
%matplotlib inline
import matplotlib.pyplot as plots
plots.style.use('fivethirtyeight')
```

In [2]:
```
filename = "https://s3.amazonaws.com/sds171/labs/lab07/ted_talks.csv"
data = Table.read_table(filename)

transcripts = data.column('transcript')
```

In [3]:
```
#Using regular expression to clean the data
transcripts = [re.sub('-', ' ', plot) for plot in transcripts]
transcripts = [re.sub('[^\w\s]', '', plot) for plot in transcripts]
transcripts = [re.sub('[A-Z]\w*', '', plot) for plot in transcripts]
transcripts = [re.sub('[ ]+', ' ', plot) for plot in transcripts]
```

In [4]:
```
def is_numeric(string):
    for char in string:
        if char.isdigit():
            return True
    return False

def has_poss_contr(string):
    for i in range(len(string) - 1):
        if string[i] == '\'' and string[i+1] == 's':
            return True
    return False

def empty_string(string):
    return string == ''

def remove_string(string):
    return is_numeric(string) | has_poss_contr(string) | empty_string(
string)
```

In [5]:
```
#Tokenize
plots_tok = []
for plot in transcripts:
    processed = plot.lower().strip().split(' ')
    plots_tok.append(processed)

#Removing numeric, posessives/contractions, and empty strings
temp = []
for plot in plots_tok:
    filtered = []
    for token in plot:
        if not remove_string(token):
            filtered.append(token)
    temp.append(filtered)
plots_tok = temp
```

In [6]:
```
import nltk
from nltk.stem import WordNetLemmatizer
nltk.download('wordnet')

#Lemmatizing the tokens
lemmatizer = WordNetLemmatizer()

temp = []
for plot in plots_tok:
    processed = []
    for token in plot:
        processed.append(lemmatizer.lemmatize(token, pos='v'))
    temp.append(processed)
plots_tok = temp
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     /Users/michaelchau/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

In [7]:
```
#Creating the Counter
vocab = Counter()
for plot in plots_tok:
    vocab.update(plot)

print("Number of unique tokens: %d" % len(vocab))
```

```
Number of unique tokens: 33616
```

In [8]:
```
#Keeping tokens that appear more than 20 times
tokens = []
for token in vocab.elements():
    if vocab[token] > 20:
        tokens.append(token)
vocab = Counter(tokens)

print("Number of unique tokens: %d" % len(vocab))
```

```
Number of unique tokens: 7101
```

In [9]:
```
#Removing rare and stop words
stop_words = []
for item in vocab.most_common(200):
    stop_word = item[0]
    stop_words.append(stop_word)
tokens = []
for token in vocab.elements():
    if token not in stop_words:
        tokens.append(token)
vocab = Counter(tokens)

print("Number of unique tokens: %d" % len(vocab))
```

```
Number of unique tokens: 6901
```

In [10]:
```
#Creating the identifier mappings word2id and id2word
items = vocab.items()
id2word = {}
word2id = {}
idx = 0
for word, count in vocab.items():
    id2word[idx] = word
    word2id[word] = idx
    idx += 1

print("Number of tokens mapped: %d" % len(id2word))
print("Identifier for 'photograph': %d" % word2id['photograph'])
print("Word for identifier %d: %s" % (word2id['photograph'], id2word[w
ord2id['photograph']]))
```

```
Number of tokens mapped: 6901
Identifier for 'photograph': 2252
Word for identifier 2252: photograph
```

In [11]:
```
#Filtering the tokens
temp = []
for plot in plots_tok:
    filtered = []
    for token in plot:
        if token in vocab:
            filtered.append(token)
    temp.append(filtered)
plots_tok = temp
```

In [12]:
```
#Creating the Corpus
sample = 30
corpus = []
for plot in plots_tok:
    plot_count = Counter(plot)
    corpus_doc = []
    for item in plot_count.items():
        pair = (word2id[item[0]], item[1])
        corpus_doc.append(pair)
    corpus.append(corpus_doc)

print("Plot, tokenized:\n", plots_tok[sample], "\n")
print("Plot, in corpus format:\n", corpus[sample])
```

```
Plot, tokenized:
 ['stuff', 'book', 'mine', 'hope', 'resonate', 'youve', 'already', '
connections', 'myself', 'case', 'miss', 'official', 'official', 'off
icial', 'industrial', 'societies', 'official', 'run', 'maximize', 'w
elfare', 'citizens', 'maximize', 'individual', 'freedom', 'reason',
'both', 'freedom', 'itself', 'valuable', 'worthwhile', 'essential',
'freedom', 'act', 'maximize', 'welfare', 'decide', 'behalf', 'maximi
ze', 'freedom', 'maximize', 'choice', 'choice', 'freedom', 'freedom'
, 'welfare', 'deeply', 'embed', 'water', 'supply', 'wouldnt', 'occur
', 'anyone', 'deeply', 'embed', 'examples', 'modern', 'progress', 'p
ossible', 'supermarket', 'such', 'word', 'salad', 'dress', 'salad',
'dress', 'supermarket', 'count', 'extra', 'virgin', 'olive', 'oil',
'buy', 'large', 'number', 'salad', 'dress', 'chance', 'none', 'store
', 'offer', 'suit', 'supermarket', 'consumer', 'electronics', 'store
', 'set', 'stereo', 'speakers', 'player', 'tape', 'player', 'single'
, 'consumer', 'electronics', 'store', 'stereo', 'systems', 'construc
t', 'six', 'half', 'million', 'stereo', 'systems', 'components', 'of
fer', 'store', 'admit', 'choice', 'domains', 'communications', 'boy'
, 'telephone', 'service', 'rent', 'phone', 'buy', 'consequence', 'ph
one', 'break', 'days', 'almost', 'unlimited', 'variety', 'phone', 'e
specially', 'cell', 'phone', 'cell', 'phone', 'future', 'favorite',
'middle', 'player', 'nose', 'hair', 'chance', 'havent', 'store', 'ye
t', 'rest', 'assure', 'soon', 'lead', 'walk', 'store', 'answer', 'an
swer', 'possible', 'buy', 'cell', 'phone', 'doesnt', 'too', 'aspects
', 'significant', 'buy', 'explosion', 'choice', 'true', 'care', 'lon
ger', 'case', 'doctor', 'doctor', 'doctor', 'doctor', 'benefit', 'ri
sk', 'benefit', 'risk', 'benefit', 'risk', 'benefit', 'risk', 'resul
t', 'patient', 'autonomy', 'sound', 'shift', 'burden', 'responsibili
ty', 'decision', 'somebody', 'namely', 'doctor', 'somebody', 'nothin
g', 'almost', 'certainly', 'sick', 'thus', 'best', 'shape', 'decisio
ns', 'namely', 'patient', 'enormous', 'market', 'prescription', 'dru
g', 'sense', 'since', 'buy', 'market', 'buy', 'answer', 'expect', 'd
octor', 'morning', 'dramatic', 'identity', 'matter', 'choice', 'slid
e', 'indicate', 'inherit', 'identity', 'invent', 're', 'invent', 'ou
rselves', 'often', 'wake', 'morning', 'decide', 'person', 'respect',
```

```
'marriage', 'family', 'default', 'assumption', 'almost', 'everyone',
'marry', 'soon', 'soon', 'real', 'choice', 'everything', 'grab', 'te
ach', 'wonderfully', 'intelligent', 'students', 'assign', 'less', 'l
ess', 'smart', 'less', 'themselves', 'marry', 'marry', 'marry', 'lat
er', 'career', 'consume', 'answer', 'whether', 'assign', 'grade', 'i
ndeed', 'answer', 'bless', 'technology', 'enable', 'minute', 'planet
', 'except', 'corner', 'anybody', 'incredible', 'freedom', 'choice',
'respect', 'decision', 'again', 'again', 'again', 'whether', 'should
nt', 'watch', 'play', 'soccer', 'cell', 'phone', 'hip', 'hip', 'lapt
op', 'presumably', 'lap', 'shut', 'minute', 'watch', 'mutilate', 'so
ccer', 'game', 'ourselves', 'answer', 'cell', 'phone', 'respond', 'e
mail', 'draft', 'letter', 'answer', 'certainly', 'experience', 'socc
er', 'game', 'wouldve', 'everywhere', 'small', 'material', 'lifestyl
e', 'matter', 'choice', 'write', 'stone', 'choices', 'everything', '
matter', 'choice', 'news', 'bad', 'news', 'answer', 'yes', 'whats',
'whats', 'bad', 'choice', 'effect', 'negative', 'effect', 'effect',
'produce', 'paralysis', 'rather', 'liberation', 'options', 'choose',
'difficult', 'choose', 'dramatic', 'example', 'study', 'investments'
, 'voluntary', 'retirement', 'plan', 'colleague', 'mine', 'access',
'investment', 'record', 'gigantic', 'mutual', 'fund', 'company', 'mi
llion', 'employees', 'mutual', 'fund', 'employer', 'offer', 'rate',
'participation', 'offer', 'fund', 'fewer', 'employees', 'participate
', 'offer', 'five', 'fund', 'choose', 'damn', 'hard', 'decide', 'fun
d', 'choose', 'youll', 'until', 'tomorrow', 'tomorrow', 'tomorrow',
'tomorrow', 'tomorrow', 'eat', 'dog', 'food', 'retire', 'enough', 'm
oney', 'away', 'decision', 'hard', 'pass', 'significant', 'match', '
money', 'employer', 'participate', 'pass', 'dollars', 'employer', 'h
appily', 'match', 'contribution', 'paralysis', 'consequence', 'too',
'choices', 'lastly', 'eternity', 'cheese', 'ranch', 'decision', 'ete
rnity', 'pick', 'wrong', 'mutual', 'fund', 'wrong', 'salad', 'dress'
, 'effect', 'second', 'effect', 'manage', 'overcome', 'paralysis', '
choice', 'less', 'satisfy', 'result', 'choice', 'fewer', 'options',
'choose', 'several', 'reason', 'salad', 'dress', 'choose', 'buy', 'p
erfect', 'salad', 'dress', 'easy', 'imagine', 'choice', 'imagine', '
alternative', 'induce', 'regret', 'decision', 'regret', 'satisfactio
n', 'decision', 'decision', 'options', 'easier', 'regret', 'anything
', 'disappoint', 'option', 'choose', 'economists', 'opportunity', 'c
ost', 'morning', 'value', 'depend', 'compare', 'alternatives', 'cons
ider', 'easy', 'imagine', 'attractive', 'feature', 'alternatives', '
reject', 'less', 'satisfy', 'alternative', 'youve', 'choose', 'examp
le', 'stop', 'available', 'park', 'space', 'street', 'suppose', 'cou
ple', 'expensive', 'real', 'estate', 'beach', 'themselves', 'damn',
'guy', 'neighborhood', 'away', 'park', 'front', 'spend', 'weeks', 'm
iss', 'opportunity', 'park', 'space', 'cost', 'satisfaction', 'choos
e', 'choose', 'terrific', 'options', 'consider', 'attractive', 'feat
ure', 'options', 'reflect', 'opportunity', 'cost', 'example', 'carto
on', 'moment', 'probably', 'slowly', 'whenever', 'choose', 'choose',
'may', 'attractive', 'feature', 'less', 'attractive', 'expectations'
, 'hit', 'replace', 'jeans', 'wear', 'jeans', 'almost', 'jeans', 'fl
avor', 'buy', 'fit', 'crap', 'incredibly', 'uncomfortable', 'wear',
```

```
'wash', 'enough', 'replace', 'jeans', 'wear', 'old', 'ones', 'pair',
'jeans', 'size', 'fit', 'easy', 'fit', 'relax', 'fit', 'button', 'fl
y', 'fly', 'acid', 'wash', 'distress', 'boot', 'cut', 'blah', 'blah'
, 'jaw', 'drop', 'recover', 'spend', 'hour', 'damn', 'jeans', 'walk'
, 'store', 'truth', 'best', 'fit', 'jeans', 'ever', 'choice', 'possi
ble', 'felt', 'worse', 'write', 'whole', 'book', 'explain', 'myself'
, 'reason', 'reason', 'felt', 'worse', 'options', 'available', 'expe
ctations', 'pair', 'jeans', 'low', 'particular', 'expectations', 'fl
avor', 'flavor', 'damn', 'perfect', 'wasnt', 'perfect', 'compare', '
expect', 'disappoint', 'comparison', 'expect', 'options', 'increase'
, 'expectations', 'options', 'produce', 'less', 'satisfaction', 'res
ult', 'result', 'market', 'wait', 'disappoint', 'wouldnt', 'truth',
'everything', 'worse', 'reason', 'everything', 'everything', 'worse'
, 'everything', 'worse', 'possible', 'experience', 'pleasant', 'surp
rise', 'industrialize', 'citizens', 'perfection', 'expectation', 'be
st', 'ever', 'hope', 'stuff', 'expect', 'surprise', 'expectations',
'expectations', 'roof', 'secret', 'happiness', 'secret', 'happiness'
, 'low', 'expectations', 'moment', 'marry', 'wife', 'shes', 'quite',
'wonderful', 'couldnt', 'settle', 'settle', 'isnt', 'always', 'such'
, 'bad', 'consequence', 'buy', 'bad', 'fit', 'pair', 'jeans', 'buy',
'whos', 'responsible', 'answer', 'clear', 'responsible', 'hundreds',
'style', 'jeans', 'available', 'buy', 'disappoint', 'whos', 'respons
ible', 'equally', 'clear', 'answer', 'hundred', 'kinds', 'jeans', 'd
isplay', 'excuse', 'failure', 'decisions', 'though', 'result', 'deci
sions', 'disappoint', 'blame', 'themselves', 'depression', 'explode'
, 'industrial', 'generation', 'believe', 'significant', 'significant
', 'explosion', 'depression', 'suicide', 'experience', 'disappoint',
'standards', 'high', 'explain', 'experience', 'themselves', 'fault',
'net', 'result', 'general', 'worse', 'remind', 'official', 'true', '
false', 'true', 'choice', 'none', 'doesnt', 'follow', 'choice', 'cho
ice', 'magical', 'amount', 'pretty', 'confident', 'since', 'pass', '
options', 'improve', 'welfare', 'policy', 'matter', 'almost', 'polic
y', 'matter', 'enable', 'choice', 'industrial', 'societies', 'materi
al', 'several', 'too', 'choice', 'too', 'stuff', 'peculiar', 'modern
', 'societies', 'frustrate', 'yesterday', 'expensive', 'difficult',
'install', 'child', 'seat', 'waste', 'money', 'expensive', 'complica
te', 'choices', 'simply', 'hurt', 'worse', 'enable', 'societies', 'c
hoices', 'shift', 'societies', 'too', 'options', 'improve', 'ours',
'improve', 'economists', 'improve', 'everyone', 'poor', 'excess', 'c
hoice', 'plague', 'conclude', 'anything', 'limit', 'suppose', 'read'
, 'cartoon', 'sophisticate', 'person', 'fish', 'nothing', 'possible'
, 'imagination', 'view', 'read', 'however', 'view', 'fish', 'truth',
'matter', 'shatter', 'everything', 'possible', 'freedom', 'paralysis
', 'shatter', 'everything', 'possible', 'decrease', 'satisfaction',
'increase', 'paralysis', 'decrease', 'satisfaction', 'almost', 'cert
ainly', 'too', 'limit', 'perhaps', 'fish', 'certainly', 'absence', '
recipe', 'misery', 'suspect', 'disaster']

Plot, in corpus format:
 [(989, 3), (364, 2), (433, 2), (429, 2), (3308, 1), (859, 2), (520,
```

1), (1647, 1), (515, 2), (1474, 2), (202, 2), (1898, 5), (2754, 3), (3409, 5), (6, 1), (3923, 5), (3924, 4), (1174, 2), (1520, 1), (2186, 8), (175, 5), (1088, 1), (2098, 1), (1333, 1), (3925, 1), (3057, 1), (2038, 1), (1075, 3), (2259, 1), (986, 20), (2302, 2), (2563, 2), (226, 1), (2707, 1), (1677, 2), (2709, 1), (1067, 1), (945, 1), (412, 2), (2711, 1), (936, 7), (1561, 3), (1566, 2), (163, 1), (3926, 6), (1838, 6), (969, 1), (2991, 1), (1953, 1), (3826, 1), (1559, 1), (612, 11), (1085, 1), (320, 1), (121, 2), (897, 2), (2298, 7), (787, 5), (2434, 1), (617, 2), (3927, 2), (1196, 1), (3928, 3), (3294, 1), (2403, 3), (3929, 1), (693, 1), (135, 2), (641, 1), (79, 1), (1037, 1), (1135, 2), (923, 1), (863, 1), (3930, 1), (3931, 1), (108, 1), (2841, 1), (519, 1), (479, 1), (341, 8), (274, 3), (352, 1), (50, 1), (1663, 6), (3932, 1), (13, 1), (960, 1), (992, 5), (18, 1), (2567, 1), (506, 1), (1945, 1), (1883, 1), (1391, 1), (52, 1), (922, 1), (3262, 1), (2519, 3), (97, 1), (408, 2), (933, 10), (710, 2), (189, 6), (3933, 1), (584, 4), (290, 2), (295, 3), (1048, 1), (738, 1), (401, 6), (1145, 4), (1151, 4), (137, 6), (3934, 2), (3935, 1), (314, 1), (306, 2), (1144, 1), (1997, 1), (622, 7), (28, 2), (3591, 2), (1228, 2), (2636, 4), (407, 1), (2982, 1), (1076, 3), (1659, 1), (670, 3), (2084, 1), (728, 3), (3936, 1), (1402, 1), (2021, 1), (283, 2), (1345, 4), (0, 3), (3541, 2), (2391, 2), (180, 6), (543, 1), (304, 1), (3937, 1), (383, 2), (679, 1), (471, 2), (25, 1), (508, 1), (65, 2), (2104, 2), (1137, 1), (484, 1), (3938, 1), (3939, 1), (976, 2), (1200, 5), (237, 2), (216, 8), (2736, 1), (197, 1), (321, 1), (935, 1), (1451, 1), (3756, 2), (621, 7), (1642, 1), (572, 4), (529, 1), (416, 1), (3336, 1), (650, 2), (1457, 1), (2970, 1), (1405, 1), (287, 1), (602, 3), (88, 2), (196, 1), (674, 1), (1496, 1), (704, 1), (804, 1), (661, 3), (223, 1), (406, 2), (20, 1), (2477, 3), (1322, 2), (1042, 1), (3940, 1), (3941, 1), (348, 1), (3525, 1), (300, 2), (1060, 1), (802, 1), (2782, 1), (530, 1), (232, 4), (3942, 1), (90, 1), (799, 1), (555, 2), (3943, 1), (249, 2), (1926, 1), (618, 4), (553, 2), (552, 4), (925, 1), (16, 2), (289, 5), (569, 1), (220, 2), (3944, 5), (145, 1), (2610, 1), (616, 10), (906, 12), (1552, 2), (964, 3), (1453, 1), (571, 1), (3945, 1), (3946, 1), (537, 1), (533, 1), (1572, 1), (1209, 1), (550, 1), (3947, 1), (2628, 3), (1308, 6), (132, 1), (961, 2), (3948, 3), (1162, 1), (719, 1), (3447, 2), (1427, 2), (51, 1), (3150, 4), (647, 2), (126, 1), (146, 1), (3174, 5), (480, 1), (1092, 1), (1109, 1), (43, 1), (775, 2), (37, 3), (3, 2), (848, 3), (2643, 2), (1267, 1), (3949, 1), (557, 1), (1568, 1), (1938, 2), (2388, 1), (3950, 1), (1723, 1), (110, 2), (15, 1), (1916, 1), (2069, 1), (3951, 2), (1423, 2), (2858, 3), (578, 3), (147, 3), (890, 2), (2256, 1), (3877, 3), (3912, 5), (627, 1), (127, 2), (3952, 6), (3594, 1), (3783, 2), (451, 3), (483, 3), (278, 1), (2187, 1), (1553, 2), (3953, 2), (96, 2), (3954, 4), (882, 3), (3404, 1), (161, 1), (384, 3), (1113, 3), (1366, 2), (1131, 1), (715, 2), (488, 1), (2024, 3), (3955, 1), (546, 1), (781, 1), (1161, 1), (1704, 1), (773, 2), (1872, 1), (3956, 1), (1230, 1), (1889, 2), (365, 2), (257, 1), (2732, 1), (3357, 1), (19, 1), (3957, 7), (475, 1), (682, 2), (998, 11), (795, 3), (3857, 3), (833, 6), (3093, 1), (1183, 1), (1903, 1), (1830, 2), (353, 1), (952, 1), (1460, 3), (872, 1), (3958, 1), (744, 1), (461, 2),

```
(3692, 1), (3959, 1), (463, 1), (695, 1), (3960, 2), (3961, 1), (931
, 1), (3962, 1), (1285, 1), (1008, 3), (82, 2), (1949, 2), (994, 7),
(4, 1), (1002, 2), (500, 2), (436, 1), (60, 1), (1499, 1), (1365, 2)
, (892, 1), (3963, 1), (1732, 2), (1489, 1), (3964, 1), (3965, 1), (
1291, 1), (1066, 2), (3280, 2), (336, 1), (61, 1), (606, 1), (372, 1
), (380, 1), (1751, 2), (182, 1), (398, 1), (720, 2), (418, 3), (181
6, 2), (807, 1), (1854, 1), (1543, 1), (3288, 1), (1646, 1), (1478,
1), (1426, 1), (3653, 1), (798, 1), (864, 1), (2723, 2), (2675, 1),
(2349, 1), (143, 1), (1658, 1), (586, 1), (225, 1), (3966, 1), (2381
, 1), (2323, 1), (1397, 1), (2293, 1), (1626, 1), (2932, 1), (805, 1
), (69, 1), (3967, 1), (886, 4), (1083, 2), (1601, 1), (1796, 1), (3
19, 1), (3968, 1), (152, 1), (1922, 1), (1099, 1), (3210, 1), (990,
1), (1752, 1), (1207, 1), (1177, 1), (2997, 1), (3969, 1), (213, 1),
(700, 2), (1735, 2), (3239, 1), (2840, 3), (446, 1), (268, 2), (1431
, 1), (3630, 2), (3970, 2), (2250, 1), (3971, 1), (3972, 1), (3915,
1), (3610, 1), (2439, 1)]
```

In [13]:
```
%%time
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                            id2word=id2word,
                                            num_topics=10,
                                            random_state=100,
                                            update_every=1,
                                            chunksize=100,
                                            passes=10,
                                            alpha='auto',
                                            per_word_topics=True)
```

```
CPU times: user 1min 5s, sys: 532 ms, total: 1min 6s
Wall time: 36 s
```

In [14]:
```
num_topics = 10
num_words = 15
top_words = Table().with_column('word rank', np.arange(1,num_words+1))
for k in np.arange(num_topics):
    topic = lda_model.get_topic_terms(k, num_words)
    words = [id2word[topic[i][0]] for i in np.arange(num_words)]
    probs = [topic[i][1] for i in np.arange(num_words)]
    top_words = top_words.with_column('topic %d' % k, words)

top_words.show()
```

word rank	topic 0	topic 1	topic 2	topic 3	topic 4	topic 5	topic 6	topic 7	
1	company	planet	social	play	cancer	guy	water	data	
2	money	species	political	sound	health	write	energy	technology	
3	countries	animals	group	game	body	word	plant	machine	
4	dollars	fish	power	music	cells	ever	food	computer	
5	country	water	believe	body	disease	spend	climate	information	ex
6	market	tree	against	video	drug	run	carbon	science	
7	city	ocean	between	hand	patients	bite	grow	example	
8	global	star	reason	listen	blood	person	air	model	
9	cities	planets	government	arm	medical	too	solar	image	
10	million	sea	may	head	doctor	read	gas	process	
11	business	refugees	society	dance	care	job	produce	robot	
12	job	forest	case	voice	baby	maybe	fuel	robots	
13	pay	humans	war	experience	study	everything	eat	pattern	
14	cost	ice	state	record	medicine	car	fly	number	
15	economic	land	example	eye	die	name	cloud	object	

In [15]:
```
sample = 13
topic_dist = lda_model.get_document_topics(corpus[sample], minimum_pro
bability = 0)
topics = [pair[0] for pair in topic_dist]
probabilities = [pair[1] for pair in topic_dist]
topic_dist_table = Table().with_columns('Topic', topics, 'Probabilitie
s', probabilities)
topic_dist_table.show(20)
t = np.argmax(probabilities)
print("Topic with highest probability: %d (%f)" % (t, probabilities[t]
))
```

Topic	Probabilities
0	0.00198656
1	0.00127246
2	0.0105815
3	0.205505
4	0.000858629
5	0.0503681
6	0.0108479
7	0.646343
8	0.0696848
9	0.00255206

```
Topic with highest probability: 7 (0.646343)
```

In [16]:
```
print(transcripts[sample][0:2500])
```

```
 really excited to be here today show you some stuff thats just read
y to come out of the lab literally and really glad that you guys are
going to be among the first to see it in person because really think
this is going to really change the way we interact with machines fro
m this point on this is a rear projected drafting table about 36 inc
hes wide and its equipped with a multi touch sensor touch sensors th
at you see like on a kiosk or interactive whiteboards can only regis
ter one point of contact at a time thing allows you to have multiple
points at the same time can use both my hands can use chording actio
ns can just go right up and use all 10 fingers if wanted to know lik
e that multi touch sensing isnt completely new like have been playin
g around with it in the 80s the approach built here is actually high
resolution low cost and probably most importantly very scalable the
technology you know isnt the most exciting thing here right now othe
r than probably its newfound accessibility really interesting here i
s what you can do with it and the kind of interfaces you can build o
n top of it lets see for instance we have a lava lamp application he
re you can see can use both of my hands to kind of squeeze and put t
he blobs together can inject heat into the system here or can pull i
t apart with two of my fingers completely intuitive theres no instru
ction manual interface just kind of disappears started out as a scre
ensaver app that one of the students in our lab made think its true
identity comes out here whats great about a multi touch sensor is th
at you know could be doing this with as many fingers here but of cou
rse multi touch also inherently means multi user could be interactin
g with another part of while play around with it here can imagine a
new kind of sculpting tool where kind of warming something up making
it malleable and then letting it cool down and solidifying in a cert
ain state should have something like this in their lobby show you a
little more of a concrete example here as this thing loads is a phot
ographers light box application can use both of my hands to interact
and move photos around whats even cooler is that if have two fingers
can actually grab a photo and then stretch it out like that really e
asily can pan zoom and rotate it effortlessly can do that grossly wi
th both of my hands or can do it just with two fingers on each of my
hands together grab the canvas can do the same thing stretch it out
can do it simultaneously holding this down a
```

In this example, Topic 7, which represents technology, has the highest probability with .646. Looking at the transcript of the talk, we see that this is in fact true. In the transcript of this sample, we see terms like "screensaver", "touch sensor", and "interactive".

In [17]:
```
sample = 7
topic_dist = lda_model.get_document_topics(corpus[sample], minimum_pro
bability = 0)
probabilities = [pair[1] for pair in topic_dist]
topics = [pair[0] for pair in topic_dist]
topic_dist_table = Table().with_columns('Topic', topics, 'Probabilitie
s', probabilities)
topic_dist_table.show(20)
t = np.argmax(probabilities)
print("Topic with highest probability: %d (%f)" % (t, probabilities[t]
))
```

Topic	Probabilities
0	0.172373
1	0.0142661
2	0.0366336
3	0.000343127
4	0.000314115
5	0.0978113
6	0.000276459
7	0.159664
8	0.514234
9	0.00408484

```
Topic with highest probability: 8 (0.514234)
```

In [18]:
```
print(transcripts[sample][0:2500])
```

```
 going to present three projects in rapid fire dont have much time t
o do it want to reinforce three ideas with that rapid fire presentat
ion first is what like to call a hyper rational process a process th
at takes rationality almost to an absurd level and it transcends all
the baggage that normally comes with what people would call sort of
a rational conclusion to something it concludes in something that yo
u see here that you actually wouldnt expect as being the result of r
ationality second the second is that this process does not have a si
gnature is no authorship are obsessed with authorship is something t
hat has editing and it has teams but in fact we no longer see within
this process the traditional master architect creating a sketch that
his minions carry out the third is that it challenges and this is in
the length of this very hard to support why connect all these things
but it challenges the high modernist notion of flexibility modernist
s said we will create sort of singular spaces that are generic almos
t anything can happen within them call it sort of shotgun flexibilit
y turn your head this way shoot and youre bound to kill something th
is is the promise of high modernism within a single space actually a
ny kind of activity can happen as were seeing operational costs are
starting to dwarf capital costs in terms of design parameters so wit
h this sort of idea what happens is whatever actually is in the buil
ding on opening day or whatever seems to be the most immediate need
starts to dwarf the possibility and sort of subsume it of anything e
lse could ever happen so were proposing a different kind of flexibil
ity something that we call compartmentalized flexibility the idea is
that you within that continuum identify a series of points and you d
esign specifically to them can be pushed off center a little bit but
in the end you actually still get as much of that original spectrum
as you originally had hoped high modernist flexibility that doesnt r
eally work going to talk about going to build up the in this way bef
ore your eyes in about five or six diagrams and truly mean this is t
he design process that youll see the library staff and the library b
oard we settled on two core positions is the first one and this is s
howing over the last 900 years the evolution of the book and other t
echnologies diagram was our sort of position piece about the book an
d our position was books are technology thats something people forge
t but its a form of technology that will have
```

In this sample, we observe that topic 8, which represents art, has the highest probability with .51. Looking at the transcript, we can see that our topic model is correct since there are terms like "modernists", "design", and "diagram".

In [19]:
```
sample = 31
topic_dist = lda_model.get_document_topics(corpus[sample], minimum_pro
bability = 0)
probabilities = [pair[1] for pair in topic_dist]
topics = [pair[0] for pair in topic_dist]
topic_dist_table = Table().with_columns('Topic', topics, 'Probabilitie
s', probabilities)
topic_dist_table.show(20)
t = np.argmax(probabilities)
print("Topic with highest probability: %d (%f)" % (t, probabilities[t]
))
```

Topic	Probabilities
0	0.00560299
1	0.0331117
2	0.00944252
3	0.00036272
4	0.641746
5	0.139144
6	0.0233119
7	0.098927
8	0.00981698
9	0.0385338

```
Topic with highest probability: 4 (0.641746)
```

In [20]:
```
print(transcripts[sample][0:2500])
```

```
 you really an honor and a privilege to be here spending my last day
as a teenager want to talk to you about the future but first going t
o tell you a bit about the past story starts way before was born gra
ndmother was on a train to the death camp she was going along the tr
acks and the tracks split somehow we dont really know exactly the wh
ole story but the train took the wrong track and went to a work camp
rather than the death camp grandmother survived and married my grand
father were living in and my mother was born when my mother was two
years old the revolution was raging and they decided to escape got o
n a boat and yet another divergence the boat was either going to or
to got on and didnt know where they were going and ended up in to ma
ke a long story short they came to grandmother was a chemist worked
at the in and at 44 she died of stomach cancer never met my grandmot
her but carry on her name her exact name and like to think carry on
her scientific passion too found this passion not far from here actu
ally when was nine years old family was on a road trip and we were i
n the had never been a reader when was young my dad had tried me wit
h the tried tried all that and just didnt like reading books my moth
er bought this book when we were at the called was all about the out
break of the virus something about it just kind of drew me towards i
t was this big sort of bumpy looking virus on the cover and just wan
ted to read it picked up that book and as we drove from the edge of
the to and to actually here where we are today in read that book and
from when was reading that book knew that wanted to have a life in m
edicine wanted to be like the explorers read about in the book who w
ent into the jungles of went into the research labs and just tried t
o figure out what this deadly virus was from that moment on read eve
ry medical book could get my hands on and just loved it so much was
a passive observer of the medical world wasnt until entered high sch
ool that thought now you know being a big high school kid can maybe
become an active part of this big medical world was 14 and emailed p
rofessors at the local university to see if maybe could go work in t
heir lab hardly anyone responded mean why would they respond to a 14
year old anyway got to go talk to one professor who accepted me into
the lab that time was really interested in neuroscience and wanted t
o do a research project in neurology specifically looking at the eff
ects of heavy metals on the developing nervous
```

The topic with the highest probability of .64 is topic 4, which represents medicine. Looking at the transcript of the talk, we observe that our topic model was able to correctly identify the topic at hand. Terms like "cancer", "medical", and "research" were all used in this TED talk.