Introductory Machine Learning: Assignment 6

Deadline:

Assignment 6 is due Thursday, November 30 at 11:59pm. Late work will not be accepted as per the course policies (see the Syllabus and Course policies on Canvas.

Directly sharing answers is not okay, but discussing problems with the course staff or with other students is encouraged.

You should start early so that you have time to get help if you're stuck. The drop-in office hours schedule can be found on Canvas. You can also post questions or start discussions on Ed Discussion. The problems are broken up into steps that should help you to make steady progress.

Submission:

Submit your assignment as a .pdf on Gradescope, and as a .ipynb on Canvas. You can access Gradescope through Canvas on the left-side of the class home page. The problems in each homework assignment are numbered. Note: When submitting on Gradescope, please select the correct pages of your pdf that correspond to each problem. This will allow graders to find your complete solution to each problem.

To produce the .pdf, please do the following in order to preserve the cell structure of the notebook:

1. Go to "File" at the top-left of your Jupyter Notebook
2. Under "Download as", select "HTML (.html)"
3. After the .html has downloaded, open it and then select "File" and "Print" (note you will not actually be printing)
4. From the print window, select the option to save as a .pdf

Topics

1. Bayesian inference
2. Topic models

The first two problems test some of the basics of Bayesian inference. The third problem has you building topic models and using them to fit some linear regressions. The fourth problem asks you to build topic models on the UN data.

Note: The assignment looks longer than it really is. We step you through most of the code that you need. But it's still on the long side. Although the assignment is due in three weeks, we encourage you to start early!

Problem 1: Let the good times roll (10 points)

Consider the scenario of rolling a 4-sided die with the numbers 1, 2, 3, and 4 on its faces. Suppose we roll this die many times and get a collection of n outcomes represented by $X_1, X_2, \ldots, X_n$.

Here each X_i is a random variable that independently follows a Multinomial(p_1, p_2, p_3, p_4) model (where $p_1 + p_2 + p_3 + p_4 = 1$).

This die may or may not be fair. If it were fair then $p_1 = p_2 = p_3 = p_4 = 0.25$, but since we are uncertain about these parameters we treat them as random and the problem requires Bayesian inference.

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Part (a)

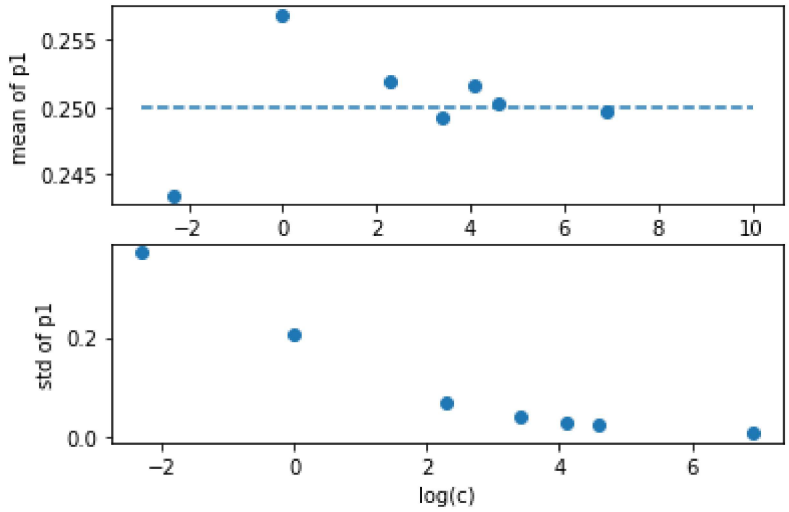
For (a) we will assume that (p_1, p_2, p_3, p_4) follows a Dirichlet$(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ distribution where $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ are unknown, positive-valued parameters. Suppose we have a prior belief that the four-sided die is close to being fair. This is represented by $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = c$ for some positive real number c.

For $c = 0.1, 1, 10, 30, 60, 100, 1000$ draw 1000 samples of (p_1, p_2, p_3, p_4) from a Dirichlet (c, c, c, c) distribution. For this sample, calculate the mean and standard deviation of p_1. Create a plot of $log(c)$ vs. the mean and another plot of $log(c)$ vs. the standard deviation. Describe in your own words what happens to these two quantities as c increases.

In [ ]:
```
p1_mean = []
p1_std = []
for c in [0.1, 1, 10, 30, 60, 100, 1000]:
    p1_mean.append(np.mean((np.random.dirichlet((c, c, c, c), 1000)), axis=0)[0])
    p1_std.append(np.std((np.random.dirichlet((c, c, c, c), 1000)), axis=0)[0])

plt.subplot(2, 1, 1)
plt.scatter(np.log([0.1, 1, 10, 30, 60, 100, 1000]), p1_mean)
plt.xlabel("log(c)")
plt.ylabel("mean of p1")
plt.hlines(0.25, -3, 10, linestyles="dashed")
plt.subplot(2, 1, 2)
plt.scatter(np.log([0.1, 1, 10, 30, 60, 100, 1000]), p1_std)
plt.xlabel("log(c)")
plt.ylabel("std of p1")
```

Out[ ]:
```
Text(0, 0.5, 'std of p1')
```



As c increases, the mean of p1 is gradually converging to 0.25, and the std of p1 is monotonically decreasing.

Part (b)

The following cell loads 10,000 rolls for the four-sided die. $[1,0,0,0]$ indicates that the die landed on face 1, $[0,1,0,0]$ indicates that the die landed on face 2, and so on. For $c = 0.1, 1, 10, 30, 60, 100, 1000$, use Dirichlet$(c,c,c,c)$ as the prior distribution for (p_1, p_2, p_3, p_4). *Using only the first* 100 *rolls of the die*, calculate the mean of the posterior distribution. What do you notice about the posterior mean as c increases?

Give code to compute the answer and plot the results. Also, give a markdown cell with a mathematical expression for the solution.

Hint: The mean of the Dirichlet$(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ is

$$\left(\frac{\alpha_1}{\alpha_1+\alpha_2+\alpha_3+\alpha_4}, \frac{\alpha_2}{\alpha_1+\alpha_2+\alpha_3+\alpha_4}, \frac{\alpha_3}{\alpha_1+\alpha_2+\alpha_3+\alpha_4}, \frac{\alpha_4}{\alpha_1+\alpha_2+\alpha_3+\alpha_4}\right)$$

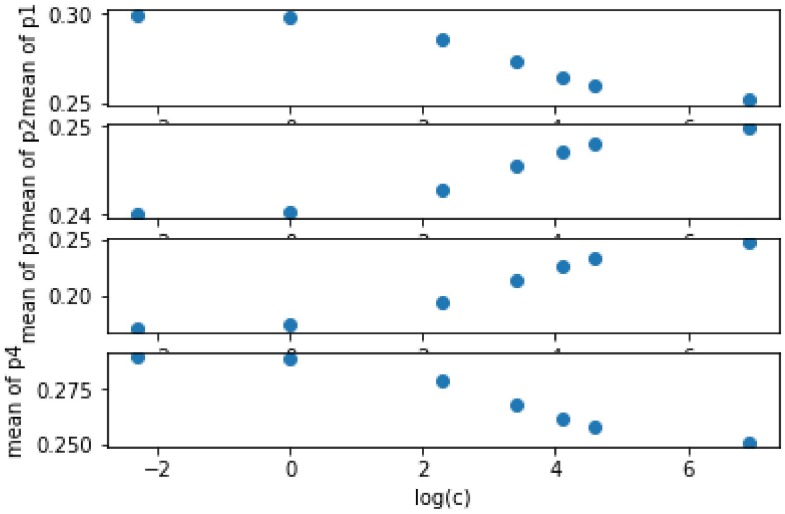
In [ ]:
```
X = pd.read_pickle('https://raw.githubusercontent.com/YData123/sds265-fa21/main/assignm
X
```

Out[ ]:
```
array([[0, 0, 0, 1],
       [0, 1, 0, 0],
       [1, 0, 0, 0],
       ...,
       [0, 0, 0, 1],
       [0, 1, 0, 0],
       [0, 1, 0, 0]], dtype=int64)
```

In [ ]:
```
post_dist = [[], [], [], []]
for c in [0.1, 1, 10, 30, 60, 100, 1000]:
    for i in range(4):
        post_dist[i].append(([c] * 4 + np.sum(X[0:100], axis=0))[i] / (100 + 4*c))

for i in range(4):
    plt.subplot(4, 1, i+1)
    plt.scatter(np.log([0.1, 1, 10, 30, 60, 100, 1000]), post_dist[i])
```

```
    plt.xlabel("log(c)")
    plt.ylabel("mean of p"+str(i+1))
```



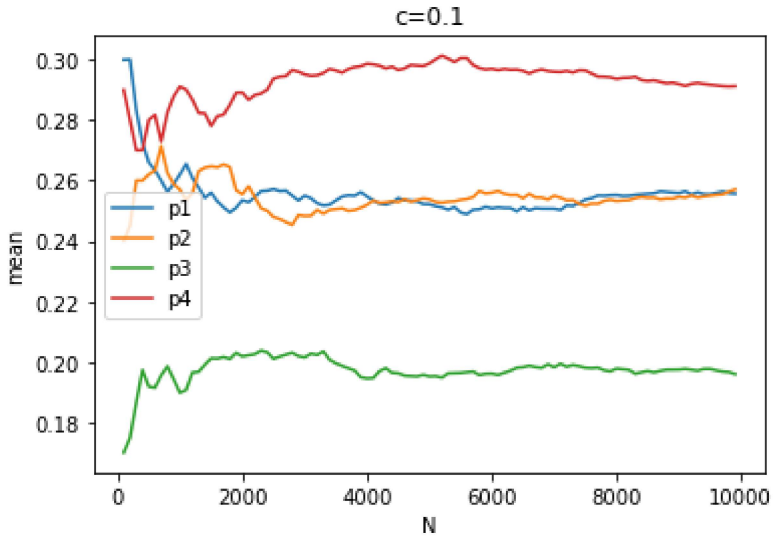
$E(p_i) = c + x_i/(4c + 100)$, for $i = 1, 2, 3, 4$. Hence, with c increasing, the mean of p_i converges to 0.25.

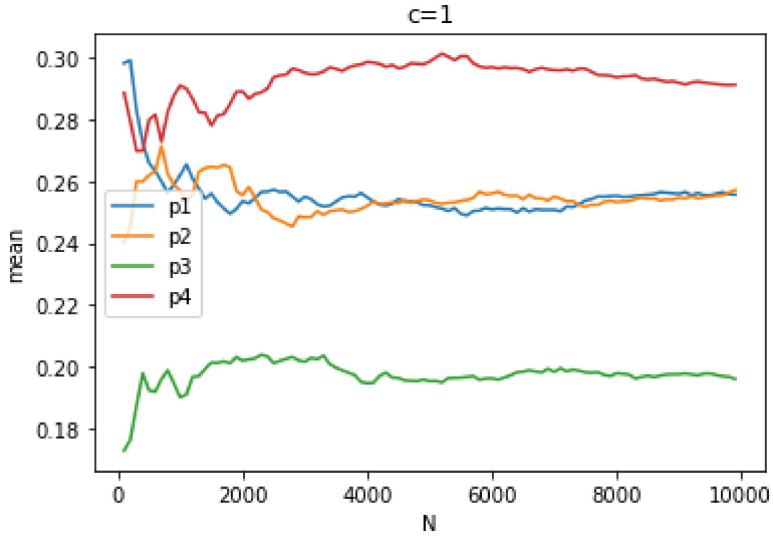
Part (c)

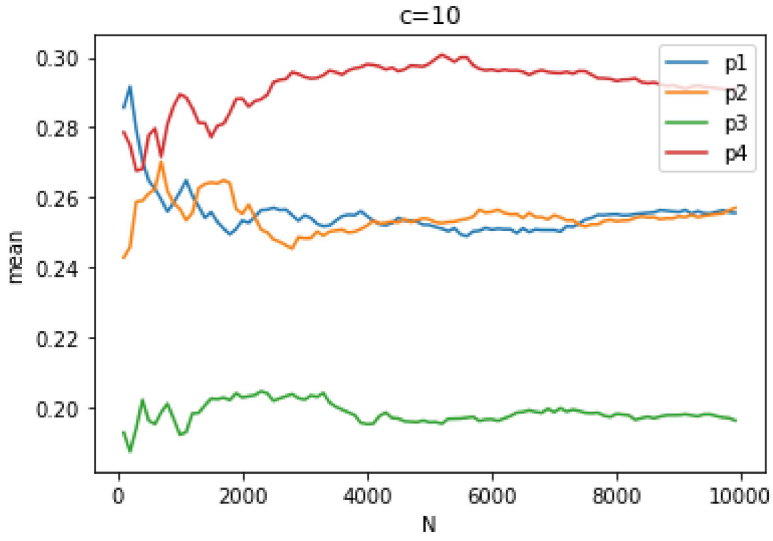
Now repeat the process in Part (b), but with sample sizes $N = 100, 200, 300, \ldots, 9900, 10000$. For each value of c, create a plot that shows the trend of the posterior mean for p_1 as a function of sample size N. Create a similar plot for p_2, p_3, and p_4. Explain what these plots illustrate about the choice of prior and the sample size. What do you estimate were the true parameters used to generate this dataset?

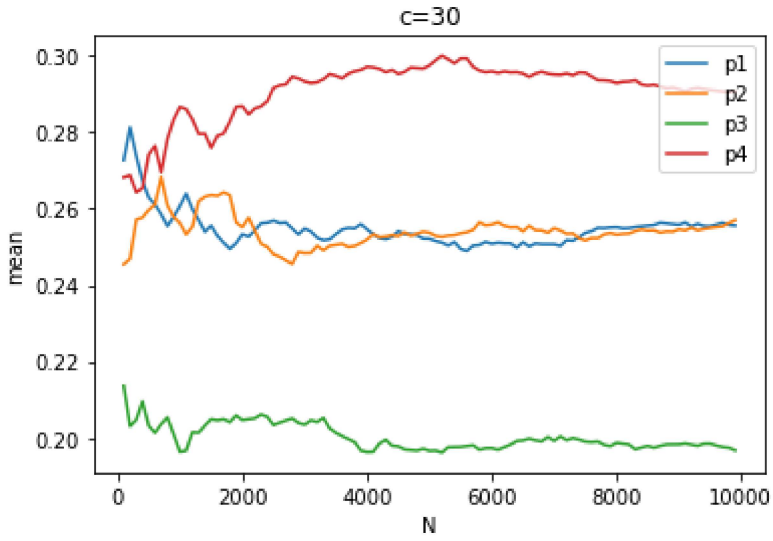
In [ ]:

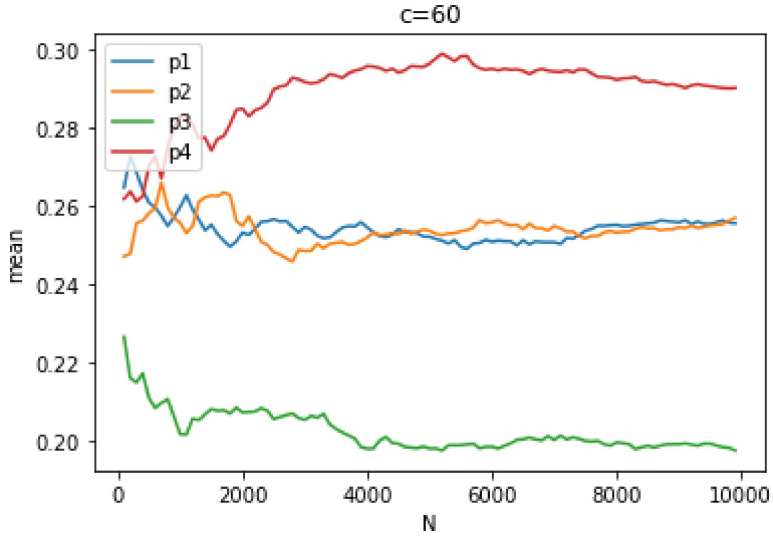
```
for c in [0.1, 1, 10, 30, 60, 100, 1000]:
    post_dist = [[], [], [], []]
    for N in range(100, 10000, 100):
        for i in range(4):
            post_dist[i].append(([c] * 4 + np.sum(X[0:N], axis=0))[i] / (N + 4*c))
    fig = plt.figure()
    plt.title("c=" + str(c))
    for i in range(4):
        plt.plot(range(100, 10000, 100), post_dist[i], label="p"+str(i+1))
        plt.legend()
        plt.xlabel("N")
        plt.ylabel("mean")
```

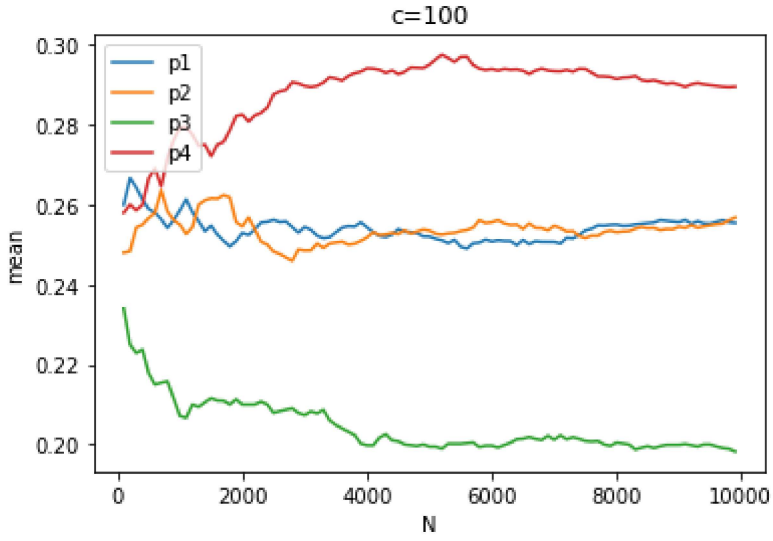

c=0.1
mean
N
p1
p2
p3
p4
0.30
0.28
0.26
0.24
0.22
0.20
0.18
0
2000
4000
6000
8000
10000

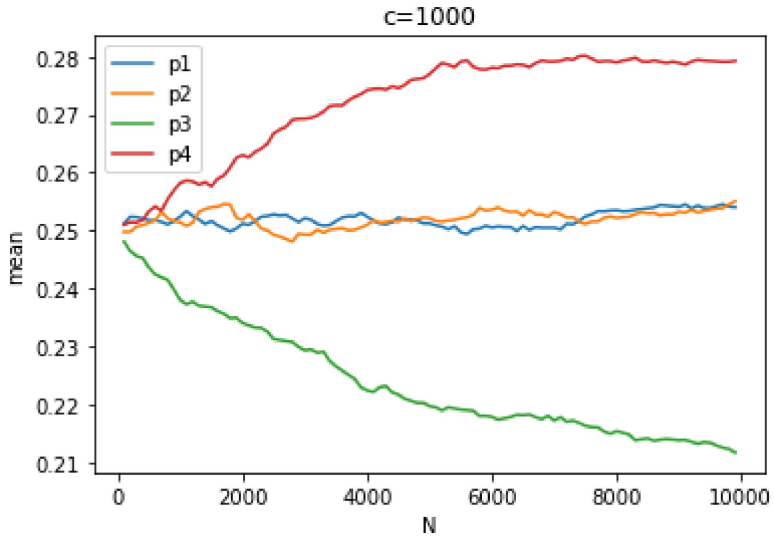

c=1
mean
N
p1
p2
p3
p4
0.30
0.28
0.26
0.24
0.22
0.20
0.18
0
2000
4000
6000
8000
10000


c=10
mean
N
p1
p2
p3
p4
0.30
0.28
0.26
0.24
0.22
0.20
0
2000
4000
6000
8000
10000


c=30
p1
p2
p3
p4
mean
N
0.30
0.28
0.26
0.24
0.22
0.20
0
2000
4000
6000
8000
10000


c=60
p1
p2
p3
p4
mean
N
0.30
0.28
0.26
0.24
0.22
0.20
0
2000
4000
6000
8000
10000


c=100
p1
p2
p3
p4
mean
N
0.30
0.28
0.26
0.24
0.22
0.20
0
2000
4000
6000
8000
10000



In [ ]:
```
X.sum(axis=0)
```

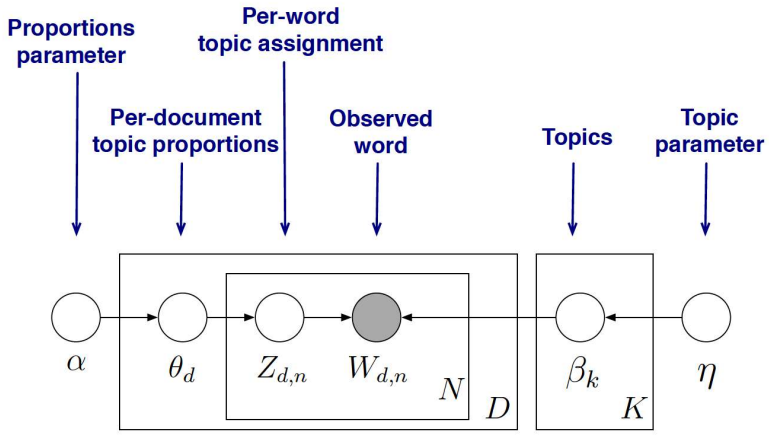
Out[ ]:
```
array([2555, 2574, 1953, 2918], dtype=int64)
```

The larger c is, with data sample size N increases, the influence of likelihood appears slower on the posterior estimate. Also, the larger c is, the converage curves are more smooth.

The real parameters are likely to be [0.25, 0.25, 0.2, 0.3].

Problem 2: Toy Story (12 points)

Gibbs sampling is one of the commonly used approach to approximate the inference for Latent Dirichlet Allocation model. In this problem, we will use the toy example from class.



Assume that there are 3 documents and 15 words in the corpus. We would like to build a topic model with 3 topics. The proportions parameter is α and the topic parameter is η. The table below shows an assignment of topics to words in the toy corpus at one stage of the Gibbs sampling algorithm.

w	z	w	z	w	z
meth	1	drug	1	inning	2
father	3	baseball	2	mother	3
divorce	3	hit	2	son	3
drug	1	inning	2	hit	2
illegal	1	steroids	1	baseball	2

Using only these assignment id Z for each word, the following problems ask you to calculate the posterior topic proportions for each document, and word probabilities for one word in each of the three topics. To answer these questions you only need to use the basic properties of the Dirichlet distribution as a prior for a multinomial, as presented in class (and in the notes on Bayesian inference).

Problem 2.1: Per-document topic proportions

Given the Z values in the table, what are the posterior distributions of θ_d for documents D_1, D_2 and D_3 from left to right. Assume the prior over θ is $\text{Dirichlet}(\alpha, \alpha, \alpha)$.

θ_1 ~ $\text{Dirichlet}(\alpha + 3, \alpha, \alpha + 2)$

θ_2 ~ $\text{Dirichlet}(\alpha + 2, \alpha + 3, \alpha)$

θ_3 ~ $\text{Dirichlet}(\alpha, \alpha + 3, \alpha + 2)$

Problem 2.2: Topics

Here are the 15 words in our corpus:

addiction, brother, baseball, catcher, daughter, divorce, drug, hit, inning, illegal, meth, mother, swing, son, steroids

What is the posterior mean for the probability $p(\text{addiction}|\text{topic } 1)$? Assume that the prior distribution over the topics is $\text{Dirichlet}(\eta, \ldots \eta)$.

$p(\text{addiction}|\text{topic } 1) = \eta/(15\eta + 5)$

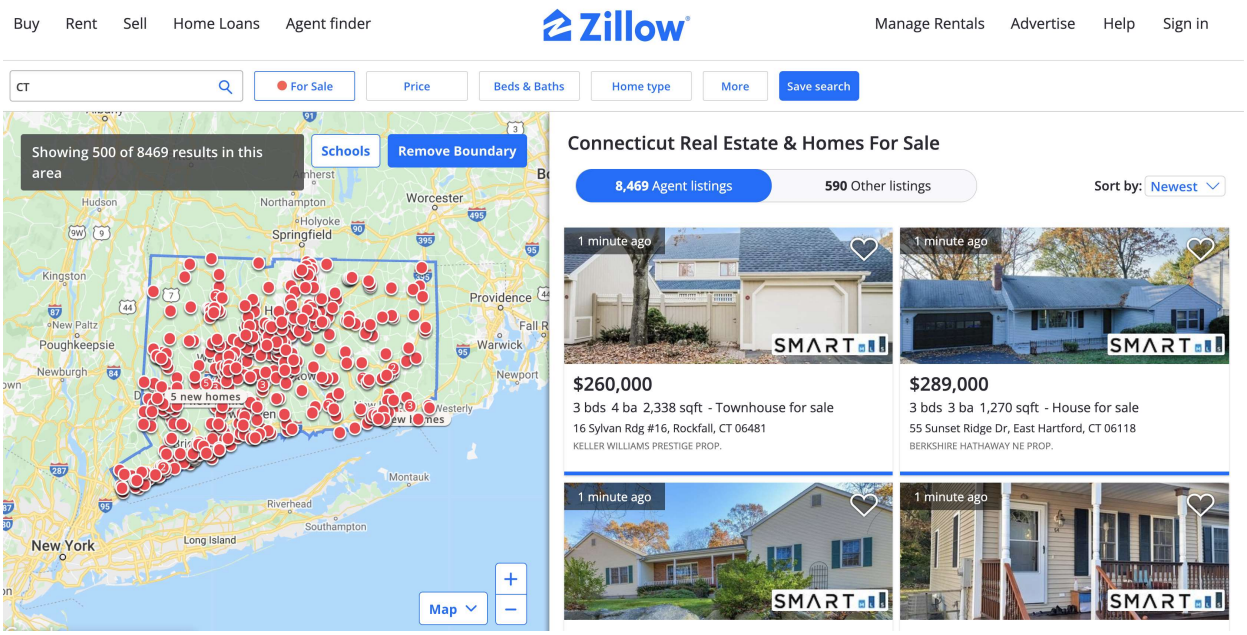
What is the posterior mean of the probability $p(\text{baseball}|\text{topic } 2)$?

$p(\text{baseball}|\text{topic } 2) = (2 + \eta)/(15\eta + 6)$

What is the posterior mean of the probability $p(\text{divorce}|\text{topic } 3)$?

$p(\text{divorce}|\text{topic } 3) = (1 + \eta)/(15\eta + 4)$

Problem 3: Read before you buy! (30 points)



Overview of the problem

Here we have a dataset of single family houses sold in Connecticut near the beginning of 2021, collected from Zillow. You will build linear models of the price for which each house sold, based on its characteristics given in the real estate listing. Such characteristics include internal square footage, the year it was built, the bedroom count, the bathroom count, and the area of the lot.

But there is also usually a lengthy description written by the real estate agent. Is there any additional information hidden in this description that would help improve the model of the price? This is the question we focus on in this problem.

Answering such a question is difficult because the description is written in natural language with thousands of different words. Here we use topic models as a dimension reduction technique. Specifically, instead of using thousands of possible words, and how many times they show up in each house description, we reduce the words to the topic proportions θ_d for each document, obtained by posterior inference. These proportions are combined with the other quantitative variables in a linear model with the logarithm of the house price as the response variable.

Important note: At first glance, this problem looks really long. But this is deceiving. After reading in the data, we have you make some plots of the log-transformed variables. After that, you just need to run the code that leads up to training a 10-topic topic model, and fitting a linear model using the resulting topic proportions. After this, you are asked to compare the results to those obtained with a 3-topic model. To do this, you can simply copy the code used for the 10-topic model. After that, the crux of the problem is to analyze, understand, and describe the results.

Acknowledgment: The data were scraped and the analysis was done by Parker Holzer, as he began his search for a new house for his family after beginning a job as a data scientist. Thanks Parker!

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

```
import re
import gensim
from collections import Counter
import statsmodels.formula.api as sm
import matplotlib.pyplot as plt
%matplotlib inline
```

Read in and clean up the data

In [ ]:
```
ct_homes = pd.read_csv('https://raw.githubusercontent.com/YData123/sds265-fa21/main/ass
ct_homes
```

Out[ ]:

	AREA	BED	BATH	BUILT	DESCRIPTION	LOTSIZE	PRICE
0	1629.0	2.0	2.0	1889.0	Welcome home! Charming & well kept, this 2 bed...	0.159986	224000.0
1	1278.0	3.0	2.0	1900.0	This adorable cape has a lot to offer. You st...	0.179981	225000.0
2	1264.0	3.0	2.0	1988.0	This 1264 sqft Colonial with its 3 bedrooms an...	0.089991	224900.0
3	2054.0	3.0	3.0	1960.0	The perfect oversized ranch awaits you at 7 No...	0.569994	370000.0
4	4198.0	5.0	3.0	1972.0	Beautiful Colonial-3020 sqft. living space and...	0.939989	489999.0
...	...	...	...	...	...	...	...
1921	848.0	3.0	2.0	1948.0	This home sets at the beginning of a Cul-de-Sa...	0.189990	429900.0
1922	2400.0	4.0	4.0	2021.0	New home to be built. Amazing unobstructed wat...	0.079981	800000.0
1923	6538.0	7.0	7.0	2002.0	Can you say water views galore? Wake up to the...	0.079981	2700000.0
1924	4480.0	5.0	5.0	1890.0	NEW YEAR! NEW FUTURE! Escape NY to Connect...	0.849998	2550000.0
1925	3000.0	4.0	3.0	2020.0	One of the nicest new construction homes avail...	0.119995	1275000.0

1926 rows × 7 columns

Transform the data

We add columns to `ct_homes` called `logAREA`, `logLOTSIZE`, and `logPRICE` that take the logarithms of the corresponding columns in the original data.

In [ ]:
```
ct_homes['logAREA'] = np.log(ct_homes['AREA'])
ct_homes['logLOTSIZE'] = np.log(ct_homes['LOTSIZE'])
ct_homes['logPRICE'] = np.log(ct_homes['PRICE'])
ct_homes
```

Out[ ]:

	AREA	BED	BATH	BUILT	DESCRIPTION	LOTSIZE	PRICE	logAREA	logLOTSIZE	logPRICE

	AREA	BED	BATH	BUILT	DESCRIPTION	LOTSIZE	PRICE	logAREA	logLOTSIZE	logPRICE
0	1629.0	2.0	2.0	1889.0	Welcome home! Charming & well kept, this 2 bed...	0.159986	224000.0	7.395722	-1.832669	12.319401
1	1278.0	3.0	2.0	1900.0	This adorable cape has a lot to offer. You st...	0.179981	225000.0	7.153052	-1.714902	12.323856
2	1264.0	3.0	2.0	1988.0	This 1264 sqft Colonial with its 3 bedrooms an...	0.089991	224900.0	7.142037	-2.408049	12.323411
3	2054.0	3.0	3.0	1960.0	The perfect oversized ranch awaits you at 7 No...	0.569994	370000.0	7.627544	-0.562129	12.821258
4	4198.0	5.0	3.0	1972.0	Beautiful Colonial-3020 sqft. living space and...	0.939989	489999.0	8.342364	-0.061887	13.102159
...	...	...	...	...	...	...	...	...	...	...
1921	848.0	3.0	2.0	1948.0	This home sets at the beginning of a Cul-de-Sa...	0.189990	429900.0	6.742881	-1.660781	12.971308
1922	2400.0	4.0	4.0	2021.0	New home to be built. Amazing unobstructed wat...	0.079981	800000.0	7.783224	-2.525960	13.592367
1923	6538.0	7.0	7.0	2002.0	Can you say water views galore? Wake up to the...	0.079981	2700000.0	8.785387	-2.525960	14.808762
1924	4480.0	5.0	5.0	1890.0	NEW YEAR! NEW FUTURE! Escape NY to Connect...	0.849998	2550000.0	8.407378	-0.162521	14.751604
1925	3000.0	4.0	3.0	2020.0	One of the nicest new construction homes avail...	0.119995	1275000.0	8.006368	-2.120304	14.058457

1926 rows × 10 columns

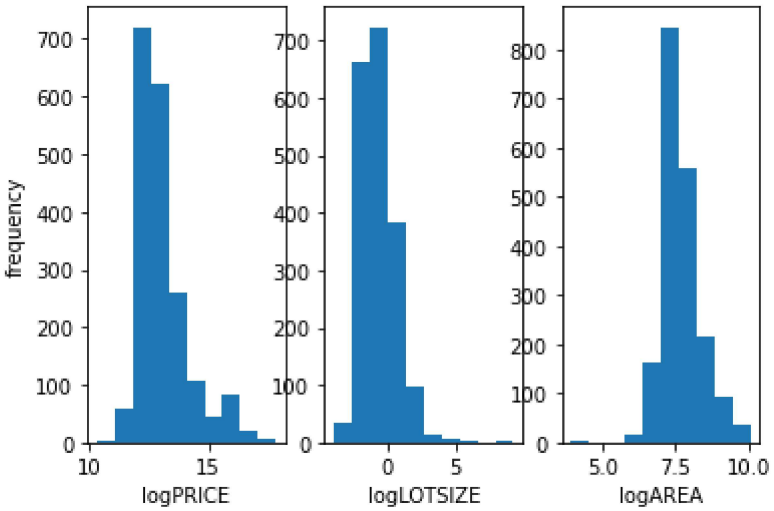
3.1 Plot the data

1. Show histograms of each of the log-transformed columns.

2. Our regression models will use these transformed values. Why might it be preferable to use the logarithms rather than the original data? Explain.

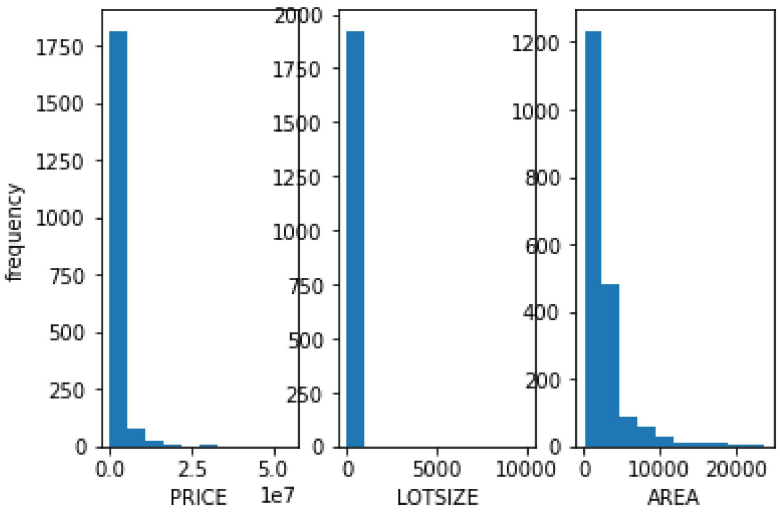
In [ ]:
```
plt.subplot(1, 3, 1)
plt.hist(ct_homes['logPRICE'])
plt.ylabel("frequency")
plt.xlabel("logPRICE")
plt.subplot(1, 3, 2)
plt.hist(ct_homes['logLOTSIZE'])
plt.xlabel("logLOTSIZE")
plt.subplot(1, 3, 3)
plt.hist(ct_homes['logAREA'])
plt.xlabel("logAREA")
```

Out[ ]:
```
Text(0.5, 0, 'logAREA')
```



In [ ]:
```
plt.subplot(1, 3, 1)
plt.hist(ct_homes['PRICE'])
plt.ylabel("frequency")
plt.xlabel("PRICE")
plt.subplot(1, 3, 2)
plt.hist(ct_homes['LOTSIZE'])
plt.xlabel("LOTSIZE")
plt.subplot(1, 3, 3)
plt.hist(ct_homes['AREA'])
plt.xlabel("AREA")
```

Out[ ]:
```
Text(0.5, 0, 'AREA')
```



As previous figures have shown, the price, lotsize, and area in the original data are long-tailed distributed, which brings difficulties to the regression model. After log-transform, the long-tailed distribution diappears, which makes it easier for regression prediction and classification.

Let's look at one of the descriptions as an example.

In [ ]:
```
example = 9
ct_homes["DESCRIPTION"][example]
```

Out[ ]:
```
"One of Ridgefield's most admired homes, custom built with absolutely no expense spared.
Stunning Stone and Clapboard New England Colonial with exceptional architectural details
throughout, beautifully sited on 1.48 park-like acres. The heart of this home features a
n updated gourmet Kitchen with Dacor double ovens, Viking cook top and Sub Zero Refriger
ator. Gorgeous Taj Mahal Quartz counters and island with Tumbled Limestone back splash.
Stylish Living Room with propane fireplace and spacious Dining Room with large Butler's
pantry also with quartz and tumbled marble. Handsome Cherry paneled Library and sun fill
ed Family Room with fireplace and built-in cabinets and desk. The Upper level features a
large Master Suite with custom marble bath and generous walk-in closet. Three Bedrooms a
re en suite; and two bedrooms share a lovely Jack n Jill Bathroom. Large, custom Laundry
Room and over sized Bonus Room round out the second floor. Other convenient features inc
lude front and rear staircases, mud room with built-ins, and large pantry and utility cl
osets. The finished Lower Level offers a second Family Room, Computer desk Stations, Gam
e Room and fabulous temperature controlled Wine Cellar. New automatic whole house Genera
tor and Rain Bird irrigation system are some of the special features. Close to all town
with shopping, great restaurants, amazing entertainment venues and schools. Excellent NY
C commute. This is one home not to be missed."
```

Helper functions

The following two functions will be used to clean up the text a bit and separate into tokens

In [ ]:
```
def cleanup_description(desc):
    if type(desc) == float:
        desc = ""
    words = [re.sub(r'[^a-z]', '', w) for w in desc.lower().split(' ')]
    return ' '.join(words)

def reduce_to_vocabulary(desc, vocab):
    return ' '.join([w for w in cleanup_description(desc).split(' ') if w in vocab])
```

In [ ]:
```
```

```
cleanup_description(ct_homes['DESCRIPTION'][example])
```

Out[ ]:
```
'one of ridgefields most admired homes custom built with absolutely no expense spared st
unning stone and clapboard new england colonial with exceptional architectural details t
hroughout beautifully sited on  parklike acres the heart of this home features an update
d gourmet kitchen with dacor double ovens viking cook top and sub zero refrigerator gorg
eous taj mahal quartz counters and island with tumbled limestone back splash stylish liv
ing room with propane fireplace and spacious dining room with large butlers pantry also
with quartz and tumbled marble handsome cherry paneled library and sun filled family roo
m with fireplace and builtin cabinets and desk the upper level features a large master s
uite with custom marble bath and generous walkin closet three bedrooms are en suite and
two bedrooms share a lovely jack n jill bathroom large custom laundry room and over size
d bonus room round out the second floor other convenient features include front and rear
staircases mud room with builtins and large pantry and utility closets the finished lowe
r level offers a second family room computer desk stations game room and fabulous temper
ature controlled wine cellar new automatic whole house generator and rain bird irrigatio
n system are some of the special features close to all town with shopping great restaura
nts amazing entertainment venues and schools excellent nyc commute this is one home not
to be missed'
```

Next we build a vocabulary of words

In [ ]:
```
vocab = Counter()
for dsc in ct_homes['DESCRIPTION']:
    vocab.update(cleanup_description(dsc).split(' '))
```

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

```
Number of unique tokens: 9738
```

Remove words that are either too common or too rare

In [ ]:
```
vocab = Counter(token for token in vocab.elements() if vocab[token] > 5)
stop_words = [item[0] for item in vocab.most_common(50)]
vocab = Counter(token for token in vocab.elements() if token not in stop_words)
print("Number of unique tokens: %d" % len(vocab))
```

```
Number of unique tokens: 2540
```

Build a mapping between unique words and integers

In [ ]:
```
desc = ct_homes['DESCRIPTION'][example]
print('Original description:\n----------------------')
print(desc)

print('\nCleaned up text:\n----------------')
print(cleanup_description(desc))

print('\nReduced to vocabulary:\n-----------------------')
print(reduce_to_vocabulary(desc, vocab))
```

```
Original description:
---------------------
One of Ridgefield's most admired homes, custom built with absolutely no expense spared.
Stunning Stone and Clapboard New England Colonial with exceptional architectural details
throughout, beautifully sited on 1.48 park-like acres. The heart of this home features a
n updated gourmet Kitchen with Dacor double ovens, Viking cook top and Sub Zero Refriger
ator. Gorgeous Taj Mahal Quartz counters and island with Tumbled Limestone back splash.
```

```
Stylish Living Room with propane fireplace and spacious Dining Room with large Butler's
pantry also with quartz and tumbled marble. Handsome Cherry paneled Library and sun fill
ed Family Room with fireplace and built-in cabinets and desk. The Upper level features a
large Master Suite with custom marble bath and generous walk-in closet. Three Bedrooms a
re en suite; and two bedrooms share a lovely Jack n Jill Bathroom. Large, custom Laundry
Room and over sized Bonus Room round out the second floor. Other convenient features inc
lude front and rear staircases, mud room with built-ins, and large pantry and utility cl
osets. The finished Lower Level offers a second Family Room, Computer desk Stations, Gam
e Room and fabulous temperature controlled Wine Cellar. New automatic whole house Genera
tor and Rain Bird irrigation system are some of the special features. Close to all town
with shopping, great restaurants, amazing entertainment venues and schools. Excellent NY
C commute. This is one home not to be missed.

Cleaned up text:
----------------
one of ridgefields most admired homes custom built with absolutely no expense spared stu
nning stone and clapboard new england colonial with exceptional architectural details th
roughout beautifully sited on  parklike acres the heart of this home features an updated
gourmet kitchen with dacor double ovens viking cook top and sub zero refrigerator gorgeo
us taj mahal quartz counters and island with tumbled limestone back splash stylish livin
g room with propane fireplace and spacious dining room with large butlers pantry also wi
th quartz and tumbled marble handsome cherry paneled library and sun filled family room
with fireplace and builtin cabinets and desk the upper level features a large master sui
te with custom marble bath and generous walkin closet three bedrooms are en suite and tw
o bedrooms share a lovely jack n jill bathroom large custom laundry room and over sized
bonus room round out the second floor other convenient features include front and rear s
taircases mud room with builtins and large pantry and utility closets the finished lower
level offers a second family room computer desk stations game room and fabulous temperat
ure controlled wine cellar new automatic whole house generator and rain bird irrigation
system are some of the special features close to all town with shopping great restaurant
s amazing entertainment venues and schools excellent nyc commute this is one home not to
be missed

Reduced to vocabulary:
----------------------
one most homes custom built absolutely no expense spared stunning stone clapboard englan
d colonial exceptional architectural details throughout beautifully sited parklike acres
heart updated gourmet dacor double ovens viking cook top sub refrigerator gorgeous quart
z counters island limestone back splash stylish propane spacious butlers pantry also qua
rtz marble handsome cherry paneled library sun filled builtin cabinets desk upper suite
custom marble generous walkin closet three are en suite two share lovely jack n jill bat
hroom custom laundry over sized bonus round out second other convenient include front re
ar staircases mud builtins pantry utility closets finished lower second desk stations ga
me fabulous controlled wine cellar automatic whole generator rain bird irrigation system
are some special close town shopping restaurants amazing entertainment schools excellent
nyc commute one not missed
```

Build a mapping between unique words and integers

In [ ]:

```
id2word = {idx: pair[0] for idx, pair in enumerate(vocab.items())}
word2id = {pair[0]: idx for idx, pair in enumerate(vocab.items())}

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

```
Number of tokens mapped: 2540
Identifier for 'nyc': 477
Word for identifier 477: nyc
```

Map to word id format

Now, use the format required to build a language model, mapping each word to its id,

In [ ]:
```
tokens = []
for dsc in ct_homes['DESCRIPTION']:
    clean = reduce_to_vocabulary(cleanup_description(dsc), vocab)
    toks = clean.split(' ')
    tokens.append(toks)
```

In [ ]:
```
corpus = []
for toks in tokens:
    tkn_count = Counter(toks)
    corpus.append([(word2id[item[0]], item[1]) for item in tkn_count.items()])

dsc = ct_homes['DESCRIPTION'][example]
clean = reduce_to_vocabulary(cleanup_description(dsc), vocab)
toks = clean.split(' ')
print("Abstract, tokenized:\n", toks, "\n")
print("Abstract, in corpus format:\n", corpus[10])
```

```
Abstract, tokenized:
 ['one', 'most', 'homes', 'custom', 'built', 'absolutely', 'no', 'expense', 'spared', 's
tunning', 'stone', 'clapboard', 'england', 'colonial', 'exceptional', 'architectural',
'details', 'throughout', 'beautifully', 'sited', 'parklike', 'acres', 'heart', 'update
d', 'gourmet', 'dacor', 'double', 'ovens', 'viking', 'cook', 'top', 'sub', 'refrigerato
r', 'gorgeous', 'quartz', 'counters', 'island', 'limestone', 'back', 'splash', 'stylis
h', 'propane', 'spacious', 'butlers', 'pantry', 'also', 'quartz', 'marble', 'handsome',
'cherry', 'paneled', 'library', 'sun', 'filled', 'builtin', 'cabinets', 'desk', 'upper',
'suite', 'custom', 'marble', 'generous', 'walkin', 'closet', 'three', 'are', 'en', 'suit
e', 'two', 'share', 'lovely', 'jack', 'n', 'jill', 'bathroom', 'custom', 'laundry', 'ove
r', 'sized', 'bonus', 'round', 'out', 'second', 'other', 'convenient', 'include', 'fron
t', 'rear', 'staircases', 'mud', 'builtins', 'pantry', 'utility', 'closets', 'finished',
'lower', 'second', 'desk', 'stations', 'game', 'fabulous', 'controlled', 'wine', 'cella
r', 'automatic', 'whole', 'generator', 'rain', 'bird', 'irrigation', 'system', 'are', 's
ome', 'special', 'close', 'town', 'shopping', 'restaurants', 'amazing', 'entertainment',
'schools', 'excellent', 'nyc', 'commute', 'one', 'not', 'missed']

Abstract, in corpus format:
 [(481, 1), (79, 1), (386, 2), (155, 1), (399, 1), (482, 1), (483, 1), (326, 1), (484,
1), (115, 1), (485, 1), (141, 2), (486, 1), (46, 1), (304, 1), (487, 1), (488, 1), (41,
2), (489, 1), (490, 1), (491, 1), (492, 1), (325, 1), (493, 1), (494, 1), (350, 1), (49
5, 1), (496, 1), (345, 2), (497, 1), (498, 1), (499, 1), (77, 1), (500, 2), (501, 2), (5
02, 1), (503, 1), (25, 1), (50, 1), (51, 1), (504, 1), (380, 1), (153, 1), (252, 1), (7
8, 1)]
```

Build a Topic Model with 10 topics

Note: Don't worry about the various settings used in the call to `LdaModel`. If you want to read up on these, just check out the documentation.

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

```
Wall time: 19.4 s
```

In [ ]:

```
num_topics = 10
num_words = 15
top_words = pd.DataFrame({'word rank': np.arange(1,num_words+1)})
for k in np.arange(num_topics):
    topic = tm.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['topic %d' % k] = words

top_words
```

Out[ ]:

	word rank	topic 0	topic 1	topic 2	topic 3	topic 4	topic 5	topic 6	topic
0	1	create	property	spacious	waterfront	north	will	beach	gues
1	2	architect	it	additional	ft	boasts	roof	water	so
2	3	premier	own	main	sq	provides	well	sound	pan
3	4	indoor	location	perfect	milford	state	have	views	ro
4	5	shaker	at	bathroom	unique	morning	newer	long	met
5	6	kayaks	can	two	built	fire	one	miles	s
6	7	acreage	opportunity	lower	construction	many	water	post	masterpie
7	8	restored	close	finished	if	garden	been	steps	ener
8	9	beginning	town	throughout	yale	work	by	directly	ar
9	10	ny	come	beautiful	barn	coffee	ranch	across	servi
10	11	coveted	train	car	yet	allows	out	sunsets	ca
11	12	landscape	shopping	first	build	original	windows	experience	accommoda
12	13	stained	only	walk	road	sits	basement	skylight	shelvi
13	14	gateway	minutes	storage	harbor	charm	owner	unobstructed	paradi
14	15	kayak	located	granite	few	warm	yard	consists	movi

In [ ]:

```
topic_dist = tm.get_document_topics(corpus[example])
topics = [pair[0] for pair in topic_dist]
probabilities = [pair[1] for pair in topic_dist]
topic_dist_table = pd.DataFrame()
topic_dist_table['Topic'] = topics
topic_dist_table['Probabilities'] = probabilities
topic_dist_table
```

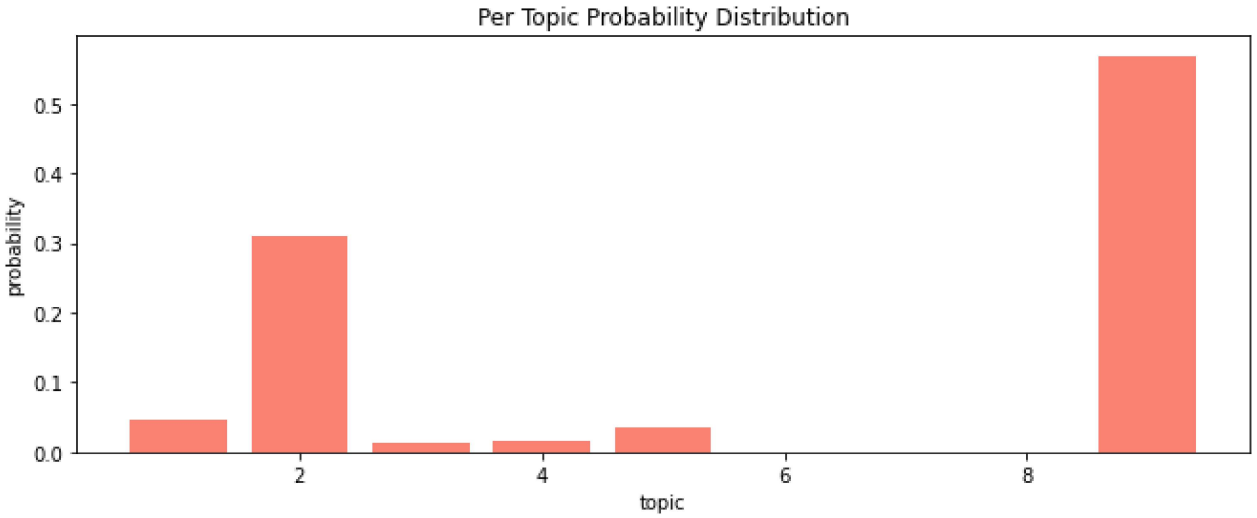
Out[ ]:

	Topic	Probabilities
0	1	0.044948

	Topic	Probabilities
1	2	0.308954
2	3	0.013631
3	4	0.014789
4	5	0.034338
5	9	0.568487

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

fig = plt.figure()
fig.set_size_inches(11,4)
plt.bar(topic_dist_table['Topic'], topic_dist_table['Probabilities'], align='center', a
plt.xlabel('topic')
plt.ylabel('probability')
plt.title('Per Topic Probability Distribution')
plt.show()
```



Include the topic proportions θ_d for each house

In [ ]:
```
num_topics = 10
theta = pd.DataFrame({"Theta0": np.zeros(ct_homes.shape[0])})
for t in np.arange(1,num_topics):
    theta["Theta"+str(t)] = np.zeros(ct_homes.shape[0])

for i in np.arange(ct_homes.shape[0]):
    for t in tm.get_document_topics(corpus[i]):
        theta.loc[i,"Theta"+str(t[0])] = t[1]
```

In [ ]:
```
num_topics = 10
theta = pd.DataFrame({"Theta0": np.zeros(ct_homes.shape[0])})
for t in np.arange(1,num_topics):
    theta["Theta"+str(t)] = np.zeros(ct_homes.shape[0])
```

```
for i in np.arange(ct_homes.shape[0]):
    for t in tm.get_document_topics(corpus[i]):
        theta.loc[i,"Theta"+str(t[0])] = t[1]
ct_topics = ct_homes.join(theta)
ct_topics
```

Out[ ]:

	AREA	BED	BATH	BUILT	DESCRIPTION	LOTSIZE	PRICE	logAREA	logLOTSIZE	logPRICE
0	1629.0	2.0	2.0	1889.0	Welcome home! Charming & well kept, this 2 bed...	0.159986	224000.0	7.395722	-1.832669	12.319401
1	1278.0	3.0	2.0	1900.0	This adorable cape has a lot to offer. You st...	0.179981	225000.0	7.153052	-1.714902	12.323856
2	1264.0	3.0	2.0	1988.0	This 1264 sqft Colonial with its 3 bedrooms an...	0.089991	224900.0	7.142037	-2.408049	12.323411
3	2054.0	3.0	3.0	1960.0	The perfect oversized ranch awaits you at 7 No...	0.569994	370000.0	7.627544	-0.562129	12.821258
4	4198.0	5.0	3.0	1972.0	Beautiful Colonial-3020 sqft. living space and...	0.939989	489999.0	8.342364	-0.061887	13.102159
...	...	...	...	...	...	...	...	...	...	...
1921	848.0	3.0	2.0	1948.0	This home sets at the beginning of a Cul-de-Sa...	0.189990	429900.0	6.742881	-1.660781	12.971308
1922	2400.0	4.0	4.0	2021.0	New home to be built. Amazing unobstructed wat...	0.079981	800000.0	7.783224	-2.525960	13.592367
1923	6538.0	7.0	7.0	2002.0	Can you say water views galore? Wake up to the...	0.079981	2700000.0	8.785387	-2.525960	14.808762
1924	4480.0	5.0	5.0	1890.0	NEW YEAR! NEW FUTURE! Escape NY to Connect...	0.849998	2550000.0	8.407378	-0.162521	14.751604
1925	3000.0	4.0	3.0	2020.0	One of the nicest new construction homes avail...	0.119995	1275000.0	8.006368	-2.120304	14.058457

1926 rows × 20 columns

Fit a linear model with the topic proportions included

We now fit a linear model with the topic proportions included. Note that the proportions satisfy $\theta_0 + \theta_1 + \cdots + \theta_9 = 1$. Therefore, we remove one of them, since it is redundant. If we don't do this the linear model will be harder to interpret!

In [ ]:
```
model = sm.ols("logPRICE ~ logAREA + logLOTSIZE + BED + BATH + BUILT + Theta0 + " +
               "Theta1 + Theta2 + Theta3 + Theta4 + Theta5 + Theta6 + Theta7 + Theta8",
model.summary()
```

Out[ ]:

OLS Regression Results

Dep. Variable:	logPRICE	**R-squared:**	0.849
Model:	OLS	**Adj. R-squared:**	0.848
Method:	Least Squares	**F-statistic:**	769.3
Date:	Tue, 30 Nov 2021	**Prob (F-statistic):**	0.00
Time:	15:36:56	**Log-Likelihood:**	-1036.1
No. Observations:	1926	**AIC:**	2102.
Df Residuals:	1911	**BIC:**	2186.
Df Model:	14		
Covariance Type:	nonrobust		

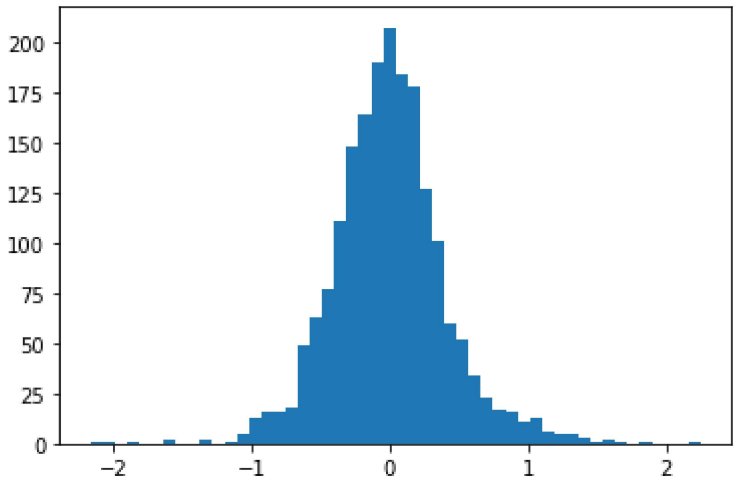
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	9.0960	0.529	17.190	0.000	8.058	10.134
logAREA	0.4800	0.033	14.570	0.000	0.415	0.545
logLOTSIZE	0.0496	0.009	5.318	0.000	0.031	0.068
BED	-0.0081	0.013	-0.644	0.519	-0.033	0.017
BATH	0.1461	0.010	14.585	0.000	0.126	0.166
BUILT	0.0008	0.000	3.598	0.000	0.000	0.001
Theta0	1.7750	0.631	2.814	0.005	0.538	3.012
Theta1	-2.4646	0.163	-15.093	0.000	-2.785	-2.144
Theta2	-2.2093	0.140	-15.758	0.000	-2.484	-1.934
Theta3	-1.0789	0.301	-3.579	0.000	-1.670	-0.488
Theta4	-1.9870	0.294	-6.763	0.000	-2.563	-1.411
Theta5	-2.3583	0.149	-15.820	0.000	-2.651	-2.066
Theta6	3.9929	0.465	8.578	0.000	3.080	4.906
Theta7	-1.0803	0.650	-1.662	0.097	-2.355	0.194

Theta8	0.6659	0.513	1.299	0.194	-0.340	1.671

Omnibus:	127.151	**Durbin-Watson:**	1.580
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	445.893
Skew:	0.247	**Prob(JB):**	1.50e-97
Kurtosis:	5.305	**Cond. No.**	1.39e+05

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.39e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

In [ ]:
```
plt.hist(model.resid, bins=50)
plt.show()
```



Model without the topics included

In [ ]:
```
model_without_topics = sm.ols("logPRICE ~ logAREA + logLOTSIZE + BED + BATH + BUILT", d
model_without_topics.summary()
```

Out[ ]:

OLS Regression Results

Dep. Variable:	logPRICE	**R-squared:**	0.787
Model:	OLS	**Adj. R-squared:**	0.786
Method:	Least Squares	**F-statistic:**	1417.
Date:	Tue, 30 Nov 2021	**Prob (F-statistic):**	0.00
Time:	15:36:57	**Log-Likelihood:**	-1370.4
No. Observations:	1926	**AIC:**	2753.
Df Residuals:	1920	**BIC:**	2786.
Df Model:	5		

Covariance Type: nonrobust

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	5.9172	0.528	11.203	0.000	4.881	6.953
logAREA	0.7285	0.035	20.723	0.000	0.660	0.797
logLOTSIZE	0.0673	0.010	6.481	0.000	0.047	0.088
BED	-0.0620	0.015	-4.214	0.000	-0.091	-0.033
BATH	0.2373	0.011	21.734	0.000	0.216	0.259
BUILT	0.0006	0.000	2.249	0.025	7.19e-05	0.001

Omnibus:	271.078	**Durbin-Watson:**	1.530
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	863.510
Skew:	0.705	**Prob(JB):**	3.10e-188
Kurtosis:	5.961	**Cond. No.**	9.18e+04

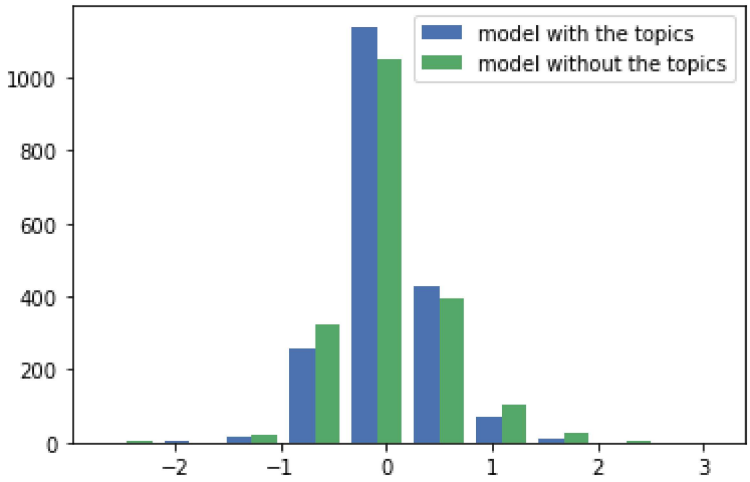
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.18e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

3.2 Plot the residuals

On a single plot, show a histogram of the residuals of the model without the topics, and the residuals of the model with the topics. Give a legend that shows which is which. Comment on the results.

In [ ]:

```
plt.style.use('seaborn-deep')
plt.hist([model.resid, model_without_topics.resid], label=["model with the topics", "mo
plt.legend(loc='upper right')
plt.show()
```



There are more samples with residuals less than 1/2 in the model with the topics than model without topics, and less samples with residuals greater than 1 in the model with topics than model without topics.

3.3 Quantify the improvement: R-squared

How do the two models compare in terms of R-squared? What do these numbers mean?

R-squared is how well the regression model fits the observed data. The model with topics has R-squared 0.849 vs. 0.787 in model without topics, which reveals that the model with topics fits and interprets the data well.

3.4 Quantify the improvement: MSE decrease

What is the percent decrease in the mean-squared-error of the model with the topics compared to the model that ignores the descriptions?

In [ ]:

```
print(np.mean(model.resid ** 2))
print(np.mean(model_without_topics.resid ** 2))
print("the percent decrease in the mean-squared-error of the model with the topics comp
```

```
0.17170058016149925
0.24296561156927043
the percent decrease in the mean-squared-error of the model with the topics compared to
the model that ignores the descriptions: 0.2933132427567975
```

3.5 Quantify the improvement: LOOCV

What is the percent decrease in the leave-one-out-cross-validation (LOOCV) error? Recall from class that the following formula can be used to calculate this:

A Slick Shortcut

Suppose the fitted values can be written $\widehat{Y} = HY$ where H is an $n \times n$ matrix.

This is the case for least squares and regularized regression

Then the leave-one-out-cross-validation error is

$$R_{LOOCV} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_{(-i)})^2$$
$$= \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i - \widehat{Y}_i}{1 - H_{ii}}\right)^2$$

where H_{ii} is the ith diagonal entry.

So, no need to fit n regressions!

29

The following line of code computes this for one of the models:

```
np.mean((model.resid/(1 - model.get_influence().hat_matrix_diag))**2)
```

In [ ]:
```
print(np.mean((model.resid/(1 - model.get_influence().hat_matrix_diag))**2))
print(np.mean((model_without_topics.resid/(1 - model_without_topics.get_influence().hat
print("percent decrease: "+ str((np.mean((model_without_topics.resid/(1 - model_without
```

```
0.1769764968718589
0.24733877014174002
percent decrease: 0.28447733135229586
```

3.6 Repeat for three topics

Now, repeat the above steps for a topic model that is trained using only three (3) topics. Specifically:

1. Train a model with three topics
2. Display the top words in each of the three topics
3. Augment the `ct_homes` data with the resulting topic proportions θ
4. Fit a linear model *using only the first two of the three* proportions
5. Plot a histogram of the residuals of the three linear models together
6. Comment on the improvement over the baseline in terms of R-squared, MSE, and LOOCV compared with the previous two models.

In [ ]:
```
%%time
tm2 = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                      id2word=id2word,
                                      num_topics=3,
                                      random_state=100,
                                      chunksize=100,
                                      passes=10,
                                      alpha='auto',
                                      per_word_topics=True)
```

```
Wall time: 19.5 s
```

In [ ]:

```
num_topics =3
num_words = 15
top_words = pd.DataFrame({'word rank': np.arange(1,num_words+1)})
for k in np.arange(num_topics):
    topic = tm2.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['topic %d' % k] = words

top_words
```

Out[ ]:

	word rank	topic 0	topic 1	topic 2
0	1	views	property	spacious
1	2	beach	it	additional
2	3	island	will	main
3	4	suite	can	appliances
4	5	custom	located	throughout
5	6	by	are	lower
6	7	sound	at	yard
7	8	pool	one	beautiful
8	9	waterfront	own	bathroom
9	10	stone	location	granite
10	11	water	lot	gas
11	12	long	make	closet
12	13	built	by	finished
13	14	designed	not	two
14	15	at	close	perfect

In [ ]:

```
num_topics = 3
theta = pd.DataFrame({"Theta0": np.zeros(ct_homes.shape[0])})
for t in np.arange(1,num_topics):
    theta["Theta"+str(t)] = np.zeros(ct_homes.shape[0])

for i in np.arange(ct_homes.shape[0]):
    for t in tm2.get_document_topics(corpus[i]):
        theta.loc[i,"Theta"+str(t[0])] = t[1]
ct_topics = ct_homes.join(theta)
ct_topics
```

Out[ ]:

	AREA	BED	BATH	BUILT	DESCRIPTION	LOTSIZE	PRICE	logAREA	logLOTSIZE	logPRICE

	AREA	BED	BATH	BUILT	DESCRIPTION	LOTSIZE	PRICE	logAREA	logLOTSIZE	logPRICE
0	1629.0	2.0	2.0	1889.0	Welcome home! Charming & well kept, this 2 bed...	0.159986	224000.0	7.395722	-1.832669	12.319401
1	1278.0	3.0	2.0	1900.0	This adorable cape has a lot to offer. You st...	0.179981	225000.0	7.153052	-1.714902	12.323856
2	1264.0	3.0	2.0	1988.0	This 1264 sqft Colonial with its 3 bedrooms an...	0.089991	224900.0	7.142037	-2.408049	12.323411
3	2054.0	3.0	3.0	1960.0	The perfect oversized ranch awaits you at 7 No...	0.569994	370000.0	7.627544	-0.562129	12.821258
4	4198.0	5.0	3.0	1972.0	Beautiful Colonial-3020 sqft. living space and...	0.939989	489999.0	8.342364	-0.061887	13.102159
...	...	...	...	...	...	...	...	...	...	...
1921	848.0	3.0	2.0	1948.0	This home sets at the beginning of a Cul-de-Sa...	0.189990	429900.0	6.742881	-1.660781	12.971308
1922	2400.0	4.0	4.0	2021.0	New home to be built. Amazing unobstructed wat...	0.079981	800000.0	7.783224	-2.525960	13.592367
1923	6538.0	7.0	7.0	2002.0	Can you say water views galore? Wake up to the...	0.079981	2700000.0	8.785387	-2.525960	14.808762
1924	4480.0	5.0	5.0	1890.0	NEW YEAR! NEW FUTURE! Escape NY to Connect...	0.849998	2550000.0	8.407378	-0.162521	14.751604
1925	3000.0	4.0	3.0	2020.0	One of the nicest new construction homes avail...	0.119995	1275000.0	8.006368	-2.120304	14.058457

1926 rows × 13 columns

In [ ]:
```
model3 = sm.ols("logPRICE ~ logAREA + logLOTSIZE + BED + BATH + BUILT + Theta0 + " +
                "Theta1", data=ct_topics).fit()
model3.summary()
```

Out[ ]:

OLS Regression Results

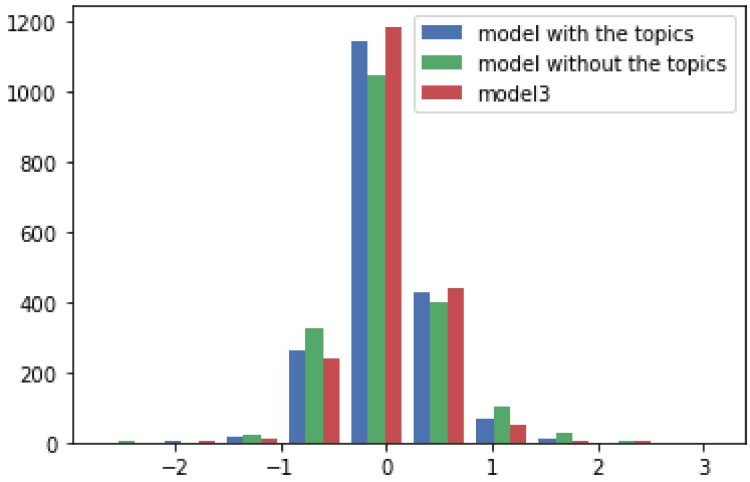
Dep. Variable:	logPRICE	**R-squared:**	0.865
Model:	OLS	**Adj. R-squared:**	0.865
Method:	Least Squares	**F-statistic:**	1759.
Date:	Tue, 30 Nov 2021	**Prob (F-statistic):**	0.00
Time:	15:37:24	**Log-Likelihood:**	-928.67
No. Observations:	1926	**AIC:**	1873.
Df Residuals:	1918	**BIC:**	1918.
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	6.6473	0.445	14.937	0.000	5.775	7.520
logAREA	0.4098	0.030	13.543	0.000	0.350	0.469
logLOTSIZE	0.0263	0.008	3.136	0.002	0.010	0.043
BED	0.0039	0.012	0.331	0.741	-0.019	0.027
BATH	0.1163	0.009	12.270	0.000	0.098	0.135
BUILT	0.0013	0.000	6.583	0.000	0.001	0.002
Theta0	2.0158	0.063	31.859	0.000	1.892	2.140
Theta1	-0.2029	0.038	-5.303	0.000	-0.278	-0.128

Omnibus:	143.835	**Durbin-Watson:**	1.570
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	536.967
Skew:	0.285	**Prob(JB):**	2.51e-117
Kurtosis:	5.523	**Cond. No.**	9.72e+04

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.72e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

In [ ]:
```
plt.style.use('seaborn-deep')
plt.hist([model.resid, model_without_topics.resid, model3.resid], label=["model with th
plt.legend(loc='upper right')
plt.show()
```



In [ ]:
```
print(np.mean(model3.resid ** 2))
print(np.mean(model.resid ** 2))
print(np.mean(model_without_topics.resid ** 2))
```

```
0.15358088584826596
0.17170058016149925
0.24296561156927043
```

In [ ]:
```
print(np.mean((model3.resid/(1 - model3.get_influence().hat_matrix_diag))**2))
print(np.mean((model.resid/(1 - model.get_influence().hat_matrix_diag))**2))
print(np.mean((model_without_topics.resid/(1 - model_without_topics.get_influence().hat
```

```
0.15628500829726097
0.1769764968718589
0.24733877014174002
```

For the last model, the R squared increases, and MSE and LOOCV error decrease, which indicates that the last model is better than the previous two.

3.7 Interpretation

Now, interpret the model. Use the coefficients of the linear model to help interpret the meaning of the topics. Comment on what this says about the effectiveness of the topic model for predicting the sale price of the house. Does it make intuitive sense? Why or why not?

In [ ]:
```
model3.summary()
```

Out[ ]:

OLS Regression Results

Dep. Variable:	logPRICE	**R-squared:**	0.865
Model:	OLS	**Adj. R-squared:**	0.865
Method:	Least Squares	**F-statistic:**	1759.
Date:	Tue, 30 Nov 2021	**Prob (F-statistic):**	0.00
Time:	15:37:25	**Log-Likelihood:**	-928.67
No. Observations:	1926	**AIC:**	1873.
Df Residuals:	1918	**BIC:**	1918.

Df Model:	7
Covariance Type:	nonrobust

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	6.6473	0.445	14.937	0.000	5.775	7.520
logAREA	0.4098	0.030	13.543	0.000	0.350	0.469
logLOTSIZE	0.0263	0.008	3.136	0.002	0.010	0.043
BED	0.0039	0.012	0.331	0.741	-0.019	0.027
BATH	0.1163	0.009	12.270	0.000	0.098	0.135
BUILT	0.0013	0.000	6.583	0.000	0.001	0.002
Theta0	2.0158	0.063	31.859	0.000	1.892	2.140
Theta1	-0.2029	0.038	-5.303	0.000	-0.278	-0.128

Omnibus:	143.835	**Durbin-Watson:**	1.570
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	536.967
Skew:	0.285	**Prob(JB):**	2.51e-117
Kurtosis:	5.523	**Cond. No.**	9.72e+04

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.72e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

Predicted log of PRICE = 0.4098*logAREA* + *0.0263*logLOTSIZE + 0.0039*BED* + *0.1163*BATH + 0.0013*BUILT* + *2.0158*Theta0(proportion of topic0) -0.2029 * Theta1(proportion of topic1).

In the model, the proportion of topic0 in the document is positively important to the predicted price and the proportion of topic1 is not so important. It makes sense because the words in topic0, like "beach""island""sound""pool""waterfront", are reasons which contribute to high sale price.

Problem 4: Topic models unite! (20 points)

In this problem we will continue working with topic models, but this time with a new dataset. Instead of real estate listings, we will create topic models over speeches from the United Nations, as described here. The dataset was obtained from Kaggle, an online community of data scientists.

In [ ]:
```
import numpy as np
import pandas as pd
import re
import gensim
from collections import Counter
import matplotlib.pyplot as plt
%matplotlib inline
```

The data are in a CSV format in `un-general-debates.csv`. To read it in you can use the function `pd.read_csv`.

In [ ]:
```
%%time
un_data = pd.read_csv('https://sds265.s3.amazonaws.com/un-general-debates.csv')
un_data
```

```
Wall time: 1min 25s
```

Out[ ]:

	session	year	country	text
0	44	1989	MDV	It is indeed a pleasure for me and the member...
1	44	1989	FIN	\nMay I begin by congratulating you. Sir, on ...
2	44	1989	NER	\nMr. President, it is a particular pleasure ...
3	44	1989	URY	\nDuring the debate at the fortieth session o...
4	44	1989	ZWE	I should like at the outset to express my del...
...	...	...	...	...
7502	56	2001	KAZ	This session\nthat is taking place under extr...
7503	56	2001	LBR	I am honoured to\nparticipate in this histori...
7504	56	2001	BDI	It\nis for me a signal honour to take the flo...
7505	56	2001	HUN	First, may I congratulate Mr. Han Seung-soo o...
7506	56	2001	KWT	On behalf of the State of Kuwait, it\ngives m...

7507 rows × 4 columns

Your task is to build a topic model of these UN speeches.

4.1 Clean and process the data, and fit a topic model.

You can simply copy over the code from Problem 3, and make appropriate modifications here. This time build a topic model with 15 topics. Here are some suggestions that may help.

- You can take just the first (say) 5,000 characters of each speech. Take more or less as you see fit.
- Construct a vocabulary that has no more than 5,000 tokens, by setting a sufficiently high count cutoff for the minimum number of times a word appears in the corpus.
- The above two items will help the topic model train in a reasonable amount of time and memory.

Here are the helper functions used above, just renamed:

In [ ]:
```
def cleanup_speech(desc):
    if type(desc) == float:
        desc = ""
    words = [re.sub(r'[^a-z]', '', w) for w in desc.lower().split(' ')]
    return ' '.join(words)
```

```
def reduce_to_vocabulary(desc, vocab):
    return ' '.join([w for w in cleanup_speech(desc).split(' ') if w in vocab])
```

Using this, for example, we have the first 5,000 characters of the 17th speech:

In [ ]:
```
example = 17
cleanup_speech(un_data['text'][example])[0:5000]
```

Out[ ]:
```
'please accept sir the congratulations of the delectation of the byelorussian soviet soc
ialist republic on your election as president of the fortyfourth session of the united n
ations general assembly we are most appreciative of the activities of the secretarygener
al and share the view he expressed in the report on the work of the organization thatthe
united nations needs to demonstrate its capacity to function as guardian of the worlds s
ecurity a p the united nations emerged from the second world war which was unleashed by
fascism and militarism which took advantage of the lack of unity among peaceloving force
s and of the negative response of the european nations to the soviet proposals for colle
ctive action against the aggressor our people took up arms to defend the goals and princ
iples of international relations which were later enshrined with our participation in th
e united nations charter the members of the antihitler coalition were fighting not only
for their national interests but also to bring freedom and independence to so many ensla
ved nations we welcome the statements made during the general debate about the inviolabi
lity of costwar borders in europefundamental changes have taken place in the world repre
sented in the united nations these changes are most profound and radical and signify the
end of the cold war and the dawning of an era of peace and mutual trust the renunciation
of propagandist polemics and the initiation of a quest for specific bilateral and multil
ateral action to resolve existing problems by political means drawing upon the prestige
and potential of the united nationsclearly the most important feature of these multi fac
eted features is the fact that they are not confined to narrow national interests of ind
ividual states they are global in scale and thus call for a multilateral approach not on
ly are these changes encouraging but they also call for a correct understanding of where
they are leading the world in conditions of freedom of choice and of pluralism of opinio
ns based on the new political thinking these changes also mean that we must further outl
ine joint measures to achieve demilitarisation democratization and the humanization of i
nternational contacts and to establish the primacy of law in relations among statesover
the past few years we have all had a difficult oath to tread gone are the days when peop
le saw everything in terms of black and whites when everyone believed that he alone was
right when socialism was made out to be the enemy rather than a partner in world affairs
when suspicion and mistrust repelled mutual tolerance and the natural desire of nations
to live in peace and friendship with one another when some would put forward proposals t
o strengthen peace and develop cooperation while others would reject them and without ev
en attempting to understand them when universal human values were not taken into account
without wishing to continue this review of an unhappy past for which there are still tho
se who remain nostalgic even today i would point out that in our detailed often heated a
nd at times disrespectful discussions we after all did succeed in restoring and enrichin
g in united nations resolutions and recommendations the meaning of such key notions as i
nternational peace and security disarmament economic development cooperation decolonizat
ion social justice human rights and so forthbut it is too early to rest on our laurels a
ll this should be embodied in mandatory international legal instruments in this context
the next decade which  will mark the end of the twentieth century and of the second mill
ennium will be decisive in terms of whether mankind succeeds in coping with new challeng
es and whether it will emerge victorious in the struggle for peace and wellbeing for eve
ryone on this planetwe are preoccupied with the problems of eliminating the threat of wa
r bringing about disarmament resolving regional conflicts eliminating the vestiges of co
lonialism ensuring development social progress observance of human rights and the preser
vation of an ecological balance it is being increasingly recognized that these problems
have a direct bearing on the level of security and the quality of life of the peoples of
the worldit would appear that today everyone understands that the use of military force
particularly nuclear force with all its devastating consequences has run its course it i
s also clear that there can be no just settlement of regional conflicts through the use
of military force the new level of this understanding has led to important conceptual br
eakthroughs that have made it possible to conclude and implement the first ever agreemen
t on the actual elimination of a portion of nuclear arsenals of soviet and united states
```

```
medium and shorterrange missiles it has also made possible the holding of substantive ta
lks on a  per cent reduction in united states and soviet strategic offensive weapons the
initiation of the vienna talks on reductions in armed forces and'
```

In [ ]:
```
vocab = Counter()
for spe in un_data['text']:
    vocab.update(cleanup_speech(spe)[0:5000].split(' '))
```

In [ ]:
```
print(f'UN speech original vocab length: {len(vocab)}')
vocab = Counter(token for token in vocab.elements() if vocab[token] > 120)
stop_words = [item[0] for item in vocab.most_common(100)]
vocab = Counter(token for token in vocab.elements() if token not in stop_words)
print("Number of unique tokens: %d" % len(vocab))
```

```
UN speech original vocab length: 158338
Number of unique tokens: 3458
```

In [ ]:
```
id2word = {idx: pair[0] for idx, pair in enumerate(vocab.items())}
word2id = {pair[0]: idx for idx, pair in enumerate(vocab.items())}

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

```
Number of tokens mapped: 3458
Identifier for 'honor': 3111
Word for identifier 3111: honor
```

In [ ]:
```
tokens = []
for spe in un_data['text']:
    clean = reduce_to_vocabulary(cleanup_speech(spe)[0:5000], vocab)
    toks = clean.split(' ')
    tokens.append(toks)
```

In [ ]:
```
corpus = []
for toks in tokens:
    tkn_count = Counter(toks)
    corpus.append([(word2id[item[0]], item[1]) for item in tkn_count.items()])

spe = un_data['text'][10][0:5000]
clean = reduce_to_vocabulary(cleanup_speech(spe), vocab)
toks = clean.split(' ')
print("Abstract, tokenized:\n", toks, "\n")
print("Abstract, in corpus format:\n", corpus[10])
```

```
Abstract, tokenized:
 ['developing', 'suriname', 'confronted', 'decade', 'prices', 'main', 'commodities', 'di
fficulties', 'markets', 'industrialized', 'obstacles', 'creating', 'solid', 'democrati
c', 'society', 'moreover', 'distant', 'interest', 'some', 'developed', 'sister', 'diffic
ult', 'process', 'changes', 'taking', 'place', 'provision', 'basic', 'needs', 'admit',
'however', 'while', 'experiences', 'had', 'negative', 'impact', 'steady', 'objectives',
'same', 'strengthened', 'determination', 'meet', 'challenges', 'face', 'redouble', 'achi
eve', 'goals', 'set', 'ourselves', 'ago', 'reason', 'once', 'again', 'come', 'york', 'pl
easure', 'share', 'experiences', 'part', 'emerging', 'understanding', 'within', 'indepen
dent', 'doing', 'however', 'wish', 'associate', 'myself', 'expressed', 'previous', 'spea
kers', 'congratulate', 'sir', 'election', 'fortyfourth', 'am', 'pleased', 'see', 'repres
```

```
entative', 'african', 'continent', 'ongoing', 'historical', 'presidency', 'africa', 'div
ersity', 'possibilities', 'play', 'decisive', 'role', 'shaping', 'emerging', 'relation
s', 'fitting', 'indeed', 'son', 'brother', 'nigeria', 'suriname', 'historical', 'ties',
'chosen', 'since', 'served', 'example', 'because', 'contribution', 'unity', 'africa', 'p
eaceful', 'coexistence', 'experience', 'diplomatic', 'skills', 'demonstrated', 'chairma
n', 'special', 'committee', 'against', 'apartheid', 'justify', 'sincere', 'under', 'lead
ership', 'come', 'successful', 'express', 'profound', 'appreciation', 'former', 'ministe
r', 'foreign', 'affairs', 'argentina', 'dante', 'caputo', 'able', 'manner', 'he', 'guide
d', 'affairs', 'fortythird', 'wish', 'him', 'well', 'future', 'endeavours', 'likewise',
'pay', 'tribute', 'secretarygeneral', 'javier', 'perez', 'de', 'cuellar', 'personal', 'c
ontributions', 'search', 'stability', 'tireless', 'strengthen', 'achieve', 'solutions',
'numerous', 'conflicts', 'threatening', 'well', 'dedication', 'cause', 'sustained', 'dev
eloping', 'well', 'known', 'deserves', 'suriname', 'feeling', 'pride', 'satisfaction',
'succeeded', 'wish', 'majority', 'suriname', 'actively', 'starting', 'internal', 'arme
d', 'conflict', 'fight', 'brother', 'took', 'up', 'arms', 'against', 'brother', 'had',
'going', 'convinced', 'combating', 'violence', 'violence', 'bring', 'prosperity', 'roa
d', 'dialogue', 'consensus', 'road', 'recently', 'led', 'conclusion', 'agreement', 'tur
n', 'set', 'off', 'genuine', 'process', 'brought', 'end', 'bloodshed', 'victims', 'wer
e', 'innocent', 'aware', 'just', 'begun', 'shall', 'certainly', 'obstacles', 'path', 'ar
med', 'reason', 'humanity', 'friendship', 'solidarity', 'qualities', 'noted', 'shall',
'overcome', 'today', 'important', 'step', 'forward', 'being', 'taken', 'process', 'proce
ss', 'dialogue', 'consensus', 'agreement', 'already', 'resulted', 'state', 'emergency',
'eastern', 'part', 'because', 'real', 'prospects', 'safe', 'speedy', 'return', 'thousand
s', 'refugees', 'created', 'through', 'contributions', 'agencies', 'office', 'high', 'co
mmissioner', 'refugees', 'indispensable', 'indeed', 'grateful', 'assistance', 'junctur
e', 'express', 'governments', 'appreciation', 'activities', 'high', 'commissioner', 'ref
ugees', 'regard', 'problems', 'refugees', 'displaced', 'persons', 'therefore', 'applau
d', 'guatemala', 'declaration', 'plan', 'concerted', 'action', 'were', 'adopted', 'firs
t', 'conference', 'central', 'american', 'refugees', 'important', 'steps', 'right', 'sur
iname', 'once', 'again', 'position', 'fulfil', 'out', 'internationally']

Abstract, in corpus format:
 [(443, 2), (1330, 5), (1331, 1), (302, 1), (179, 1), (317, 1), (1332, 1), (161, 1), (13
33, 1), (171, 1), (162, 2), (1334, 1), (1335, 1), (643, 1), (691, 1), (1306, 1), (1336,
1), (207, 1), (314, 1), (175, 1), (1337, 1), (694, 1), (117, 4), (529, 1), (530, 1), (53
1, 1), (1338, 1), (560, 1), (288, 1), (1339, 1), (251, 2), (172, 1), (1340, 2), (612,
2), (1341, 1), (1342, 1), (1343, 1), (1059, 1), (101, 1), (1344, 1), (304, 1), (296, 1),
(297, 1), (404, 1), (1345, 1), (349, 2), (1346, 1), (144, 2), (84, 1), (97, 1), (567,
2), (524, 3), (372, 3), (888, 2), (1347, 1), (1, 1), (735, 1), (310, 3), (1348, 2), (94
7, 1), (635, 1), (1349, 1), (292, 1), (489, 3), (1350, 1), (1351, 1), (318, 1), (34, 1),
(35, 1), (69, 1), (261, 1), (10, 1), (12, 1), (20, 1), (571, 1), (147, 1), (401, 1), (12
20, 1), (916, 1), (1352, 1), (1134, 2), (11, 1), (124, 2), (1353, 1), (1354, 1), (77,
1), (539, 1), (78, 1), (537, 1), (94, 1), (1355, 1), (0, 2), (1356, 1), (1357, 3), (464,
1), (1358, 1), (1359, 1), (646, 1), (245, 1), (1290, 1), (562, 2), (525, 1), (1003, 1),
(79, 1), (1270, 1), (19, 1), (1360, 1), (1361, 1), (371, 1), (1362, 1), (417, 1), (866,
1), (867, 2), (868, 1), (1363, 1), (8, 1), (23, 1), (26, 1), (422, 1), (272, 2), (528,
1), (37, 2), (875, 1), (578, 1), (579, 1), (516, 2), (768, 1), (41, 1), (42, 1), (24,
1), (44, 1), (45, 1), (1364, 1), (48, 1), (58, 1), (253, 3), (398, 1), (1365, 1), (1366,
1), (57, 1), (16, 1), (54, 1), (500, 1), (501, 1), (502, 1), (503, 1), (17, 1), (1367,
2), (1368, 1), (146, 1), (510, 1), (977, 1), (1171, 1), (1369, 1), (104, 1), (1370, 1),
(1371, 1), (55, 1), (967, 1), (1372, 1), (1373, 1), (1374, 1), (884, 1), (51, 1), (1047,
1), (1375, 1), (669, 1), (1376, 1), (628, 1), (1377, 2), (880, 1), (902, 1), (1378, 1),
(1202, 1), (239, 1), (1177, 1), (1379, 1), (1380, 1), (353, 2), (424, 1), (1126, 1), (13
81, 2), (1000, 2), (1382, 2), (194, 1), (779, 1), (630, 1), (550, 2), (1212, 1), (1383,
1), (698, 1), (1384, 1), (1225, 1), (1385, 1), (1386, 1), (863, 2), (1387, 1), (732, 1),
(824, 1), (1388, 1), (311, 2), (591, 1), (614, 1), (1389, 1), (1390, 1), (1061, 1), (18,
1), (989, 1), (413, 1), (82, 1), (85, 2), (1391, 1), (1023, 1), (606, 1), (105, 1), (35
7, 1), (447, 1), (287, 1), (1006, 1), (1032, 2), (181, 1), (1174, 1), (1392, 1), (1393,
1), (1394, 1), (1017, 1), (1395, 5), (327, 1), (218, 1), (1396, 1), (14, 1), (13, 2), (1
397, 2), (1398, 1), (1018, 1), (1007, 1), (1399, 1), (400, 1), (1400, 1), (126, 1), (13
4, 1), (1010, 1), (958, 1), (140, 1), (1401, 1), (1402, 1), (1403, 1), (129, 1), (840,
1), (598, 1), (421, 1), (328, 1), (242, 1), (1103, 1), (342, 1), (786, 1), (687, 1), (11
87, 1), (1404, 1), (494, 1), (1405, 1), (1406, 1), (1407, 1), (108, 1)]
```

Now, complete the processing of the data in order to build a 15-topic model, displaying the top

words in each topic.

In [ ]:
```
%%time
tm = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                     id2word=id2word,
                                     num_topics=15,
                                     random_state=100,
                                     chunksize=100,
                                     passes=10,
                                     alpha='auto',
                                     per_word_topics=True)
```

```
Wall time: 1min 30s
```

In [ ]:
```
num_topics = 15
num_words = 15
top_words = pd.DataFrame({'word rank': np.arange(1,num_words+1)})
for k in np.arange(num_topics):
    topic = tm.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['topic %d' % k] = words

top_words
```

Out[ ]:

	word rank	topic 0	topic 1	topic 2	topic 3	topic 4	topic 5	topic 6
0	1	nuclear	east	democracy	european	developing	ofthe	order
1	2	weapons	middle	democratic	europe	per	reform	social
2	3	arms	region	national	union	trade	unitednations	role
3	4	disarmament	arab	process	soviet	cent	theunited	through
4	5	treaty	conflict	burundi	relations	developed	theinternational	problems
5	6	nonproliferation	afghanistan	state	detente	poverty	member	common
6	7	military	iraq	elections	policy	growth	peacekeeping	system
7	8	race	israel	society	german	resources	conflict	important
8	9	destruction	resolutions	free	germany	economy	secretarygeneral	conflicts
9	10	threat	palestinian	social	foreign	financial	process	regional
10	11	use	lebanon	region	democratic	social	role	within
11	12	mass	solution	were	poland	mdgs	operations	effective
12	13	conference	resolution	through	socialist	poor	take	issues
13	14	testing	situation	institutions	military	debt	members	among
14	15	proliferation	islamic	first	forces	assistance	resolution	means

4.2 Label the Topics

Label each the 15 topics with a short (4 words or less) description.

In [ ]:
```
topic_des = ['nuclear weapons', 'middle east', 'democracy', 'europe', 'country developi
```

4.3 Table of Topics

Create a function `create_speech_table(data, speeches, corpus, lda_model)` which does the following:

- Goes through every speech and finds the most likely topic for that speech.
- Creates a DataFrame `speech_table` that has the following columns
  - `topic` : the topic number of the most likely topic for each abstract
  - `label` : the topic label of that topic number, which you assigned in part 1
  - `prob` : the probability of that topic number
  - `speech` : a string containing the first 200 characters of the speech
- Show the first 10 rows of the table, then return the table

In [ ]:
```
def create_speech_table(data, speeches, corpus, lda_model):

    topics, labels, probs, speech = [], [], [], []
    for i in range(len(corpus)):
        topic_dist = lda_model.get_document_topics(corpus[i])
#         topic = [pair[0] for pair in topic_dist]
        prob = [pair[1] for pair in topic_dist]
        probs.append(max(prob))
#         print()
        idx =prob.index(max(prob))
        topics.append(topic_dist[idx][0])
        labels.append(speeches[topic_dist[idx][0]])
        speech.append(data['text'][i][:200])
    topic_table = pd.DataFrame({'topic': topics, 'label': labels, 'prob': probs, 'speec
    print(topic_table.head(10))
    return topic_table


# print(len(speeches))
t_tb = create_speech_table(un_data, topic_des, corpus, tm)
```

```
   topic               label      prob  \
0     11        speech words  0.457240
1     11        speech words  0.297585
2     11        speech words  0.402353
3      2           democracy  0.261189
4      4  country developing  0.301511
5     10        common words  0.406983
6     11        speech words  0.355960
7      6         world order  0.428225
8     11        speech words  0.274192
9     11        speech words  0.342265

                                              speech
0  It is indeed a pleasure for me and the member...
1  \nMay I begin by congratulating you. Sir, on ...
2  \nMr. President, it is a particular pleasure ...
```

```
3  \nDuring the debate at the fortieth session o...
4  I should like at the outset to express my del...
5  Before you began to occupy that exalted seat,...
6  It gives me great pleasure to congratulate Am...
7  My task as head of the delegation of the Sovi...
8  \nPermit me to begin by warmly congratulating...
9  I should like to express my sincere congratul...
```

4.4 Analysis for selected speeches

Choose at least five speeches and discuss how the assignment of topics either does or does not make sense, according to your own understanding of the speeches and topics.

In [ ]:
```
def choose_speech(ord):
    print(t_tb.loc[ord]['label'])
    print(top_words['topic '+str(t_tb.loc[ord]['topic'])])
    print(un_data['text'][ord][1000:2000])
    print(t_tb.loc[ord]['prob'])
```

In [ ]:
```
choose_speech(3)
```

```
democracy
0        democracy
1       democratic
2         national
3          process
4          burundi
5            state
6        elections
7          society
8             free
9           social
10          region
11            were
12         through
13    institutions
14           first
Name: topic 2, dtype: object
who will be responsible for the destiny of the country for the next five years. This fac
t, commonplace in the political tradition of a democratic country, is particularly signi
ficant in out case, since it represents the culmination of a period of democratic recove
ry during which full respect for the Constitution has been restored? all individual righ
ts have been scrupulously observed, and national reconciliation has been achieved, inter
alia by a general amnesty ratified by popular mandate. Law and justice have been restore
d, and the country has been rescued from a declining economic situation through the vigo
rous revitalization of the external sector and the application of stabilizing policies w
hich have allowed reasonable product growth, a decrease in unemployment and an increase
in real earnings.
On the international front, Uruguay has honoured all its commitments, including those of
a financial nature which entailed - then and today - considerable sacrifices. It has act
ively contri
0.26118857
```

This assignment makes sense because words about democracy and elections take a large part of the speech contexts about Uruguay democracy.

In [ ]:
```
choose_speech(7)
```

```
world order
```

```
0         order
1        social
2          role
3       through
4      problems
5        common
6        system
7     important
8     conflicts
9      regional
10       within
11    effective
12       issues
13        among
14        means
Name: topic 6, dtype: object
eorge Bush of the United States, which, in our view, contained a number of very importan
t and interesting ideas.
As always, during these days of the General Assembly's regular session representatives o
f the world community have together been recreating a panorama of the past year in the l
ife of mankind. Its overriding idea is that of peace and security, its ideal composition
is harmony of universal human values and national interests.
In making our own national contribution to this grand agreement each of us, I am
sure, wants it to become a part of an organic and unitary whole. Unfortunately, in some
places the overall composition is still marred by cracks that impair its integrity. Over
the past 12 months we have seen a rather contradictory picture of the state of the worl
d.
Of course, the central concept remains the same and the theme, a product of mankind's th
ought and suffering, as we were appropriately reminded by the fiftieth anniversary of th
e outbreak of the Second World War, h
0.42822534
```

This assignment makes sense because words about world order and national interests take a large part of the speech contexts about the whole human and nations, and the probability is high.

In [ ]:
```
choose_speech(13)
```

```
speech words
0                   he
1     secretarygeneral
2             election
3                  him
4                 wish
5              express
6           delegation
7      congratulations
8              tribute
9                 hope
10                  me
11                take
12                last
13              behalf
14             towards
Name: topic 11, dtype: object
is year in a more positive international political environment conducive to strengthenin
g the credibility and relevance of the United Nations. The continuing detente between th
e two major Powers has helped sustain the trend towards the relaxation of global tension
s in many parts of the world. It has led to increased political co-operation between the
m, particularly in the important area of disarmament, and to their increased willingness
to help find solutions to regional conflicts.
These positive trends, which began only a few years ago, have strengthened the fabric of
international diplomacy. They have injected a new confidence into the diplomatic proces
s, breathed new hopes and inspired a revitalized faith in the United Nations as a vehicl
```

```
e and catalyst for positive global change. For the first time since the birth of the Uni
ted Nations we are presented with a unique opportunity to reshape the structure of inter
national relations to conform to the clear desire of mankind for peace, s
0.3242292
```

This assignment makes sense because this speech is a election congratulations with a lot of common words, and the probability is not high.

In [ ]:
```
choose_speech(232)
```

```
global issues
0            climate
1             change
2         millennium
3         challenges
4              goals
5             summit
6        sustainable
7         commitment
8            poverty
9             action
10    implementation
11            agenda
12           address
13          national
14              need
Name: topic 14, dtype: object
me for
all countries. The Millennium Declaration (resolution
55/2) set forth specific Goals that each country should
attain by 2015 in order to usher in a world in which
every individual can live in dignity. On the eve of that
deadline, it is appropriate to reflect on the progress
made since the Millennium Summit and to discuss
new prospects for shoring up the progress that has
been achieved in combating hunger, malnutrition and
disease.

The United Nations has the primary responsibility
for the maintenance of international peace and security.
However, those objectives can be sustainably achieved
only if abject poverty does not become a breeding
ground for all sorts of societal ills. That is why, when
we embark on a collective discussion of what should
happen post-2015, we must think above all about ways
to increase the economic and social development and
prosperity of nations and to prevent conflicts before
they even occur. That applies to all countries, but
particu
0.38415915
```

This assignment makes sense because this speech is about the Millennium Declaration which is the topword of this topic.

In [ ]:
```
choose_speech(1221)
```

```
speech words
0                   he
1     secretarygeneral
2             election
3                  him
4                 wish
```

```
5              express
6           delegation
7      congratulations
8              tribute
9                 hope
10                  me
11                take
12                last
13              behalf
14             towards
Name: topic 11, dtype: object
d a severe food crisis. Since
then, efforts have been made at the national level to
address the problem.
Let me take this opportunity, on behalf of the
Government and the people of the Kingdom of Lesotho
and, indeed, on my own behalf, to express our gratitude
for the rapid response of the United Nations and its
specialized agencies and programmes and that of the
donor community to the crisis facing my country.
HIV/AIDS, which has emerged as a major health
and development threat, continues to be a source of
grave concern in my country. Most of those who are
infected are between the ages of 15 and 45 and
constitute the potential and active workforce in
Lesotho.
Of equally great concern is the corrosive effect
that HIV/AIDS has on the family structure and on the
social fabric of our society. We now have a large
number of orphans and child-headed households. Those
who are sick not only lack adequate counselling and
medicine but also lack care and support.
My delegation therefore makes a sp
0.24886422
```

It is not apparent whether the assignment makes sense or not, because the largest topic probability is low, which means the topics of the speech are dispersed.