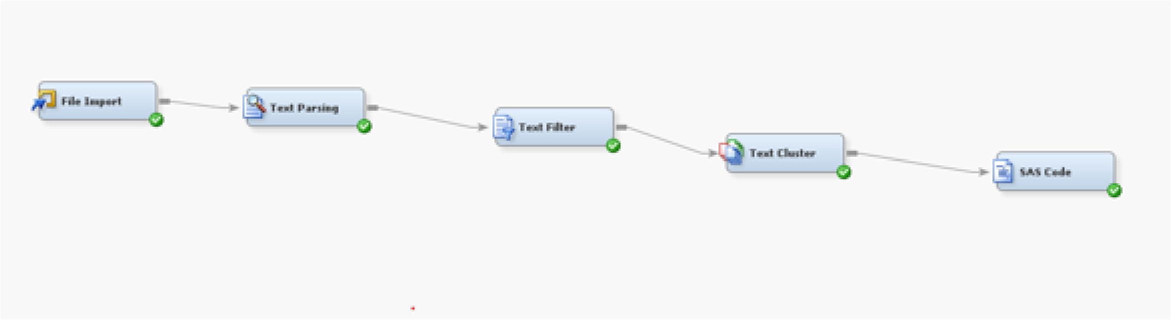
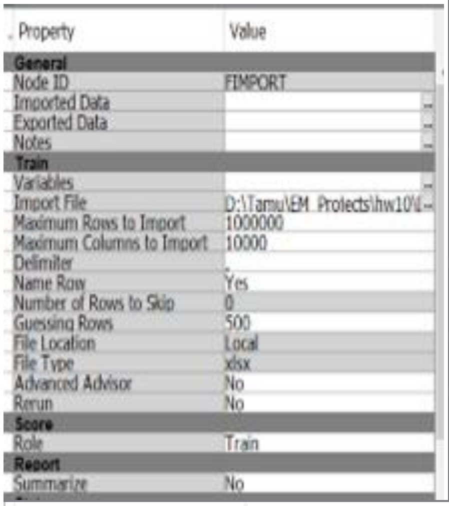
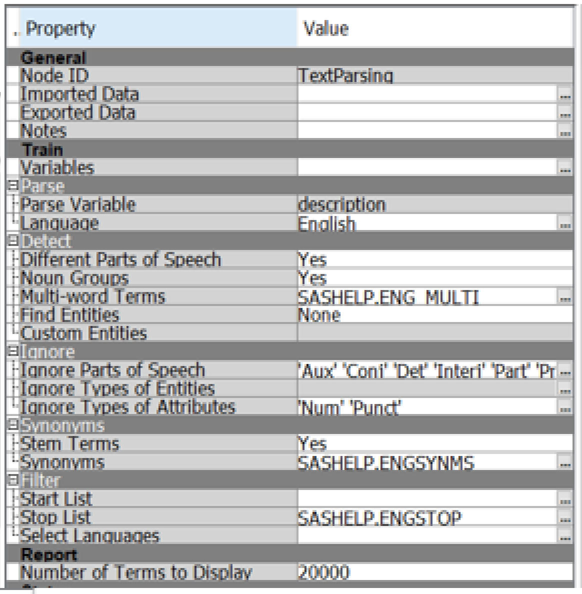
SAS Part

Report a screen shot of the diagram and the property windows for all nodes in the diagram.

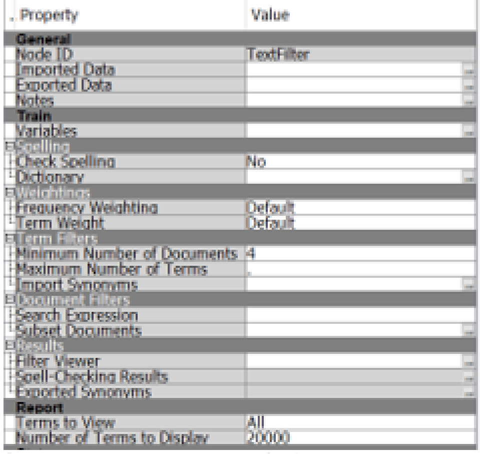




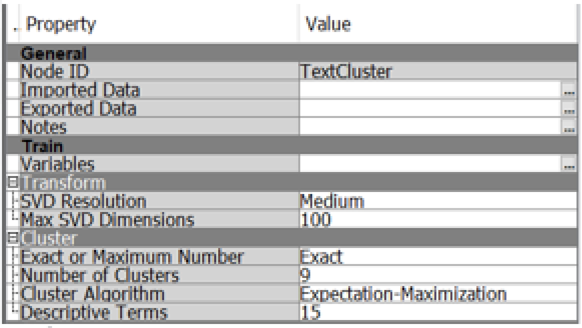
Property	Value
General	
Node ID	FIMPORT
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Import File	D:\Tamu\EM_Protects\hw10\
Maximum Rows to Import	1000000
Maximum Columns to Import	10000
Delimiter	,
Name Row	Yes
Number of Rows to Skip	0
Guessing Rows	500
File Location	Local
File Type	xlsx
Advanced Advisor	No
Rerun	No
Score	
Role	Train
Report	
Summarize	No



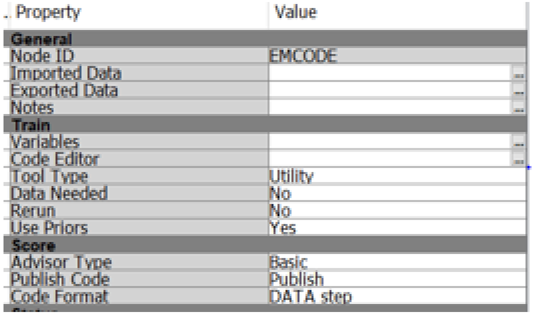
Property	Value
General	
Node ID	TextParsing
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Parse	
Parse Variable	description
Language	English
Detect	
Different Parts of Speech	Yes
Noun Groups	Yes
Multi-word Terms	SASHELP.ENG_MULTI
Find Entities	None
Custom Entities	
Ignore	
Ignore Parts of Speech	'Aux' 'Conj' 'Det' 'Interj' 'Part' 'Pr
Ignore Types of Entities	
Ignore Types of Attributes	'Num' 'Punct'
Synonyms	
Stem Terms	Yes
Synonyms	SASHELP.ENGSYNMS
Filter	
Start List	
Stop List	SASHELP.ENGSTOP
Select Languages	
Report	
Number of Terms to Display	20000



Property	Value
General	
Node ID	TextFilter
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Spelling	
Check Spelling	No
Dictionary	
Weightings	
Frequency Weighting	Default
Term Weight	Default
Term Filters	
Minimum Number of Documents	4
Maximum Number of Terms	.
Import Synonyms	
Document Filters	
Search Expression	
Subset Documents	
Results	
Filter Viewer	
Spell-Checking Results	
Exported Synonyms	
Report	
Terms to View	All
Number of Terms to Display	20000



Property	Value
General	
Node ID	TextCluster
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Transform	
SVD Resolution	Medium
Max SVD Dimensions	100
Cluster	
Exact or Maximum Number	Exact
Number of Clusters	9
Cluster Algorithm	Expectation-Maximization
Descriptive Terms	15

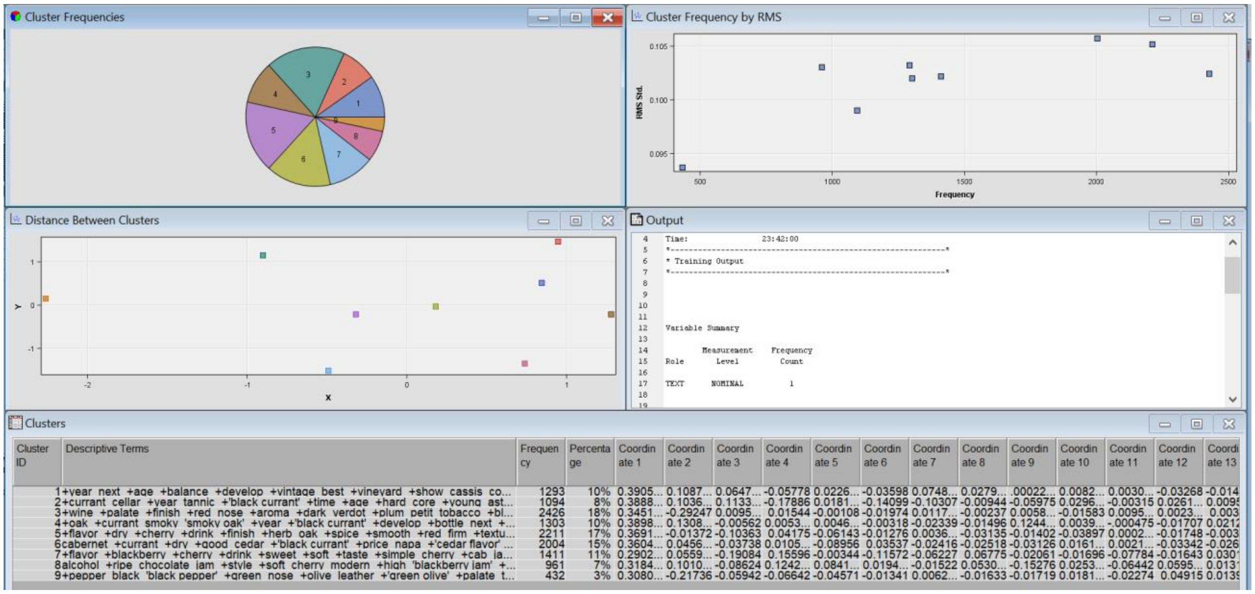


Property	Value
General	
Node ID	EMCODE
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Code Editor	
Tool Type	Utility
Data Needed	No
Rerun	No
Use Priors	Yes
Score	
Advisor Type	Basic
Publish Code	Publish
Code Format	DATA step

SAS code

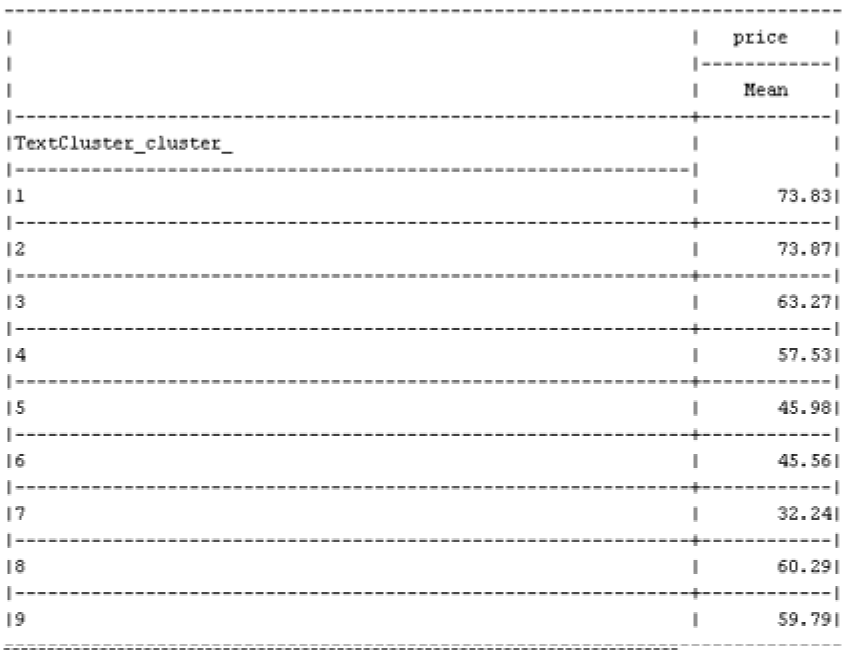
```
proc tabulate data=&em_import_data;
class TextCluster_cluster_;
var price;
var points;
table TextCluster_cluster_, price*mean;
table TextCluster_cluster_, points*mean;
run;
```

Table of average points and price for each topic group

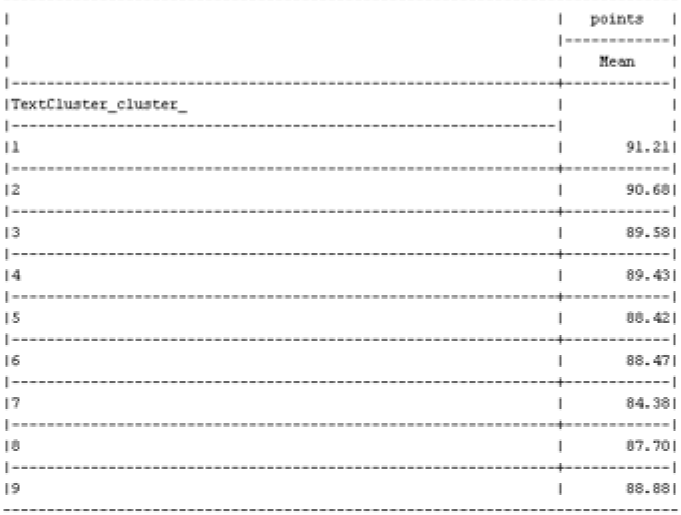


```
Time:                23:42:00
*------------------------------------------------------------*
* Training Output
*------------------------------------------------------------*




Variable Summary

          Measurement    Frequency
Role         Level         Count

TEXT       NOMINAL           1
```

Cluster ID	Descriptive Terms	Frequency	Percentage	Coordinate 1	Coordinate 2	Coordinate 3	Coordinate 4	Coordinate 5	Coordinate 6	Coordinate 7	Coordinate 8	Coordinate 9	Coordinate 10	Coordinate 11	Coordinate 12	Coordi ate 13
1	+year next +age +balance +develop +vintage best +vineyard +show cassis co...	1293	10%	0.3905...	0.1087...	0.0647...	-0.05778	0.0226...	-0.03598	0.0748...	0.0279...	.00022...	0.0082...	0.0030...	-0.03268	-0.014
2	+currant cellar +year tannic +'black currant' +time +age +hard core +young ast...	1094	8%	0.3888...	0.1036...	0.1133...	-0.17886	0.0181...	-0.14099	-0.10307	-0.00944	-0.05975	0.0296...	-0.00315	0.0261...	0.009
3	+wine +palate +finish +red nose +aroma +dark verdot +plum petit tobacco +bl...	2426	18%	0.3451...	-0.29247	0.0095...	0.01544	-0.00108	-0.01974	0.0117...	-0.00237	0.0058...	-0.01583	0.0095...	0.0023...	0.003
4	+oak +currant smoky 'smoky oak' +year +'black currant' +develop +bottle next +...	1303	10%	0.3898...	0.1308...	-0.00562	0.0053...	0.0046...	-0.00318	-0.02339	-0.01496	0.1244...	0.0039...	-.000475	-0.01707	0.021
5	+flavor +dry +cherry +drink +finish +herb oak +spice +smooth +red firm +textu...	2211	17%	0.3691...	-0.01372	-0.10363	0.04175	-0.06143	-0.01276	0.0036...	-0.03135	-0.01402	-0.03897	0.0002...	-0.01748	-0.003
6	cabernet +currant +dry +good cedar +'black currant' +price napa +'cedar flavor' ...	2004	15%	0.3604...	0.0456...	-0.03738	0.0105...	-0.08956	0.03537	-0.02416	-0.02518	-0.03126	0.0161...	0.0021...	-0.03342	-0.026
7	+flavor +blackberry +cherry +drink +sweet +soft +taste +simple cherry +cab ja...	1411	11%	0.2902...	0.0559...	-0.19084	0.15596	-0.00344	-0.11572	-0.06227	0.06775	-0.02061	-0.01696	-0.07784	-0.01643	0.030
8	alcohol +ripe chocolate jam +style +soft cherry modern +high 'blackberry jam' +...	961	7%	0.3184...	0.1010...	-0.08624	0.1242...	0.0841...	0.0194...	-0.01522	0.0530...	-0.15276	0.0253...	-0.06442	0.0595...	0.013
9	+pepper black 'black pepper' +green nose +olive leather +'green olive' +palate t...	432	3%	0.3080...	-0.21736	-0.05942	-0.06642	-0.04571	-0.01341	0.0062...	-0.01633	-0.01719	0.0181...	-0.02274	0.04915	0.013



```
-----------------------------------------------------------------------------
|                                                            |   price    |
|                                                            |------------|
|                                                            |    Mean    |
|------------------------------------------------------------+------------|
|TextCluster_cluster_                                        |            |
|------------------------------------------------------------|            |
|1                                                           |       73.83|
|------------------------------------------------------------+------------|
|2                                                           |       73.87|
|------------------------------------------------------------+------------|
|3                                                           |       63.27|
|------------------------------------------------------------+------------|
|4                                                           |       57.53|
|------------------------------------------------------------+------------|
|5                                                           |       45.98|
|------------------------------------------------------------+------------|
|6                                                           |       45.56|
|------------------------------------------------------------+------------|
|7                                                           |       32.24|
|------------------------------------------------------------+------------|
|8                                                           |       60.29|
|------------------------------------------------------------+------------|
|9                                                           |       59.79|
-----------------------------------------------------------------------------
```



```
-----------------------------------------------------------------------------
|                                                            |   points   |
|                                                            |------------|
|                                                            |    Mean    |
|------------------------------------------------------------+------------|
|TextCluster_cluster_                                        |            |
|------------------------------------------------------------|            |
|1                                                           |       91.21|
|------------------------------------------------------------+------------|
|2                                                           |       90.68|
|------------------------------------------------------------+------------|
|3                                                           |       89.58|
|------------------------------------------------------------+------------|
|4                                                           |       89.43|
|------------------------------------------------------------+------------|
|5                                                           |       88.42|
|------------------------------------------------------------+------------|
|6                                                           |       88.47|
|------------------------------------------------------------+------------|
|7                                                           |       84.38|
|------------------------------------------------------------+------------|
|8                                                           |       87.70|
|------------------------------------------------------------+------------|
|9                                                           |       88.88|
-----------------------------------------------------------------------------
```

Python Part

```
import pandas as pd
import numpy as np
import string
import nltk
from nltk import pos_tag
from nltk.tokenize import word_tokenize
from nltk.stem.snowball import SnowballStemmer
from nltk.stem import WordNetLemmatizer
from nltk.corpus import wordnet as wn
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.decomposition import LatentDirichletAllocation

nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
nltk.download('stopwords')
nltk.download('wordnet')

def fn_analyzer(s):
    #Synonym List
    synonyms = {'veh': 'vehicle',
          'car': 'vehicle',
          'chev':'cheverolet',
           'chevy':'cheverolet',
          'air bag': 'airbag',
          'seat belt':'seatbelt',
          "n't":'not',
          'to30':'to 30',
          'wont':'would not',
          'cant':'can not',
          'cannot':'can not',
          'couldnt':'could not',
          'shouldnt':'should not',
          'wouldnt':'would not' }

    s = s.lower()
    s = s.replace(',', '. ')

    tokens = word_tokenize(s)
    tokens = [word.replace(',','') for word in tokens ]
    tokens = [word for word in tokens if ('*' not in word) and \
              ("''" != word) and ("``" != word) and \
              (word!='description') and (word !='dtype') \
              and (word != 'object') and (word!="'s")]

    for i in range(len(tokens)):
        if tokens[i] in synonyms:
            tokens[i] = synonyms[tokens[i]]

    #Stop words removal
    punctuation = list(string.punctuation)+['..', '...']
    pronouns = ['i', 'he', 'she', 'it', 'him', 'they', 'we', 'us', 'them']
    stop = stopwords.words('english') + punctuation + pronouns
    filtered_terms = [word for word in tokens if (word not in stop) and \
                  (len(word)>1) and (not 
word.replace('.','',1).isnumeric()) \
                  and (not word.replace("'",'',2).isnumeric())]

    tag_words = pos_tag(filtered_terms, lang='eng')
    stemmer = SnowballStemmer("english")
    wn_tags = {'N':wn.NOUN, 'J':wn.ADJ, 'V':wn.VERB, 'R':wn.ADV}
    wnl = WordNetLemmatizer()
    stemmed_tokens = []
    for tag_token in tag_words:
        term = tag_token[0]
        pos  = tag_token[1]
        pos  = pos[0]
        try:
            pos   = wn_tags[pos]
            stemmed_tokens.append(wnl.lemmatize(term, pos=pos))
        except:
            stemmed_tokens.append(stemmer.stem(term))
    return stemmed_tokens

def fn_preprocessor(s):
    s = s.lower()
    s = s.replace(',', '. ')
    print("preprocessor")
    return(s)

def fn_tokenizer(s):
    print("Tokenizer")
    tokens = word_tokenize(s)
    tokens = [word.replace(',','') for word in tokens ]
    tokens = [word for word in tokens if word.find('*')!=True and \
              word != "''" and word !="``" and word!='description' \
              and word !='dtype']
    return tokens

pd.set_option('max_colwidth', 32575)

df = pd.read_excel("CaliforniaCabernet.xlsx")

num_docs     = len(df['description'])
num_samples  = num_docs
m_features = None
s_words    = 'english'
ngram = (1,2)

#Setup reviews in list 'discussions'
discussions = []
for i in range(num_samples):
    discussions.append(("%s" %df['description'].iloc[i]))

cv = CountVectorizer(max_df=0.95, min_df=2, max_features=m_features,\
                     analyzer=fn_analyzer, ngram_range=ngram)
tf = cv.fit_transform(discussions)
print("\nVectorizer Parameters\n", cv, "\n")


num_topics        = 9
```

```
max_iter        =  5
learning_offset = 20.
learning_method = 'online'

tf_idf = TfidfTransformer()
print("\nTF-IDF Parameters\n", tf_idf.get_params(),"\n")
tf_idf = tf_idf.fit_transform(tf)

#Constructing the TF/IDF matrix from the data
tfidf_vect = TfidfVectorizer(max_df=0.95, min_df=2,
max_features=m_features,analyzer=fn_analyzer, ngram_range=ngram)
tf_idf = tfidf_vect.fit_transform(discussions)
print("\nTF_IDF Vectorizer Parameters\n", tfidf_vect, "\n")

lda = LatentDirichletAllocation(n_components=num_topics,
max_iter=max_iter, learning_method=learning_method, \
                                learning_offset=learning_offset, \
                                random_state=12345)
lda.fit_transform(tf_idf)
print('{:.<22s}{:>6d}'.format("Number of Reviews", tf.shape[0]))
print('{:.<22s}{:>6d}'.format("Number of Terms", tf.shape[1]))
print("\nTopics Identified using LDA with TF_IDF")
tf_features = cv.get_feature_names()
max_words = 15
topic_description=[]
for topic_idx, topic in enumerate(lda.components):
        message = "Topic #%d: " % topic_idx
        message += " ".join([tf_features[i]
                  for i in topic.argsort()[:-max_words - 1:-1]])
                             topic_description.append(message[10:])
                             print(message)
                             print()

for i in range(len(topic_description)):
    topic_description[i]=topic_description[i].split(' ')

temp=lda.transform(tf_idf)
temp1=[]
for i in range(len(temp)):
    temp1.append(temp[i].argmax())
temp1=pd.DataFrame(temp1,columns=['Topic#'])
df=df.join(temp1)

table_1=df.pivot_table(['points','price'],index='Topic#')
table_1=table_1.join(pd.DataFrame(topic_description))
table_1=table_1.rename_axis({'points':'avg_points','price':'avg_price'},ax
is=1)
table_1.T

table_2=df.pivot_table('Review',index='Region',columns='Topic#',\
                      aggfunc='count',\
                      fill_value=0,margins=True)

def percent_convert(x):
    for index in x.index:
        for i in x.columns:
            x.loc[index,i]=round(x.loc[index,i]*100/x.loc[index,'All'],2)

    return x
percent_convert(table2)

print(table_1)
print(table_2)
```

Output

Table 1

```
Topic#         0         1     ...              7           8
avg_points    90.0806      84.5     ...          84.5          86
avg_price    64.7879    28.4286     ...             47      33.7778
0             wine     barely     ...      brightness      bouquet
1            flavor      wait     ...          weedy       effort
2            tannin     sweaty     ...       muscular        santa
3             black       bay     ...        breadth   light-bodied
4         blackberry   overpower     ...         recall      elevation
5           cabernet      weave     ...           farm          lurk
6           currant      chile     ...       opposite          loam
7               oak      front     ...           cake         slate
8              year     tongue     ...     black-fruit          ting
9              fruit     create     ...        relieve       notion
10           cherry      funky     ...       neighbor        excite
11              dry       drop     ...           lohr         gamy
12             rich  generosity     ...             j.       offset
13             show  acceptable     ...       six-plus  medium-weight
14             ripe  underbelly     ...         today     reduction
```

Table 2

```
Topic#                  0    1    2    3 ...    6    7    8   All
Region                                   ...
California Other     26.77  0.00  0.00  0.00  ...   0.27  0.00  0.40  100.0
Central Coast        50.70  0.17  0.28  0.00  ...   0.00  0.00  0.22  100.0
Central Valley       33.99  0.99  0.99  0.00  ...   0.00  0.00  0.00  100.0
Clear Lake            0.00  0.00  0.00  0.00  ...   0.00  0.00  0.00  100.0
High Valley           0.00  0.00  0.00  0.00  ...   0.00  0.00  0.00  100.0
Lake County          50.00  0.00  0.00  0.00  ...   0.00  0.00  0.00  100.0
Mendocino            60.00  0.00  3.33  0.00  ...   0.00  0.00  0.00  100.0
Mendocino County     62.07  0.00  0.00  0.00  ...   0.00  0.00  0.00  100.0
Mendocino Ridge          66.67  0.00  0.00  0.00  ...    0.00  0.00  0.00  100.0
Mendocino/Lake Counties  56.12  0.00  0.51  0.00  ...    0.00  0.00  0.00  100.0
Napa                     78.31  0.00  0.14  0.03  ...    0.04  0.05  0.03  100.0
Napa-Sonoma              70.24  0.00  0.00  0.00  ...    0.00  0.00  0.00  100.0
North Coast              36.07  1.09  0.00  0.00  ...    0.00  0.00  0.00  100.0
Red Hills Lake County    64.86  0.00  0.00  0.00  ...    0.00  0.00  0.00  100.0
Redwood Valley           66.67  0.00  0.00  0.00  ...    0.00  0.00  0.00  100.0
Sierra Foothills         48.41  0.00  0.00  0.00  ...    0.79  0.00  0.00  100.0
Sonoma                   65.22  0.31  0.00  0.00  ...    0.22  0.18  0.00  100.0
South Coast              42.31  0.00  0.00  0.00  ...    0.00  0.00  0.00  100.0
All                      67.07  0.11  0.14  0.02  ...    0.08  0.06  0.07  100.0
```