In [20]:

```
import numpy as np


if __name__ == '__main__':
    # There are 2000 words in the corpus
    alpha = np.full (2000, .1)
    # There are 100 topics
    beta = np.full (100, .1)

    # This gets us the probabilty of each word happening in each of the 100 topics
    wordsInTopic = np.random.dirichlet (alpha, 100)

    # Produced [doc, topic, word gives us the number of times that the given word was p
roduced by the given topic in the given doc
    produced = np.zeros ((50, 100, 2000))

# generate each doc
    for doc in range (0, 50):
        # Get the topic probabilities for this doc
        topicsInDoc = np.random.dirichlet (beta)

        # Assign each of the 2000 words in this doc to a topic
        wordsToTopic = np.random.multinomial (2000, topicsInDoc)

        # And generate each of the 2000 words
        for topic in range (0, 100):
            produced[doc, topic] = np.random.multinomial (wordsToTopic[topic], wordsInT
opic[topic])
```

In [21]:

```
# 1. Write a line of code that computes the number of words produced by topic 17 in doc
ument 18.
produced[18,17,:].sum()
```

Out[21]:

```
360.0
```

In [22]:

```
# 2. Write a line of code that computes the number of words produced by topic 17 thru 4
5 in document 18.
produced[18,17:45,:].sum()
```

Out[22]:

```
735.0
```

In [23]:

```
# 3. Write a line of code that computes the number of words in the entire corpus.
produced.sum()
```

Out[23]:

```
100000.0
```

In [24]:

```
# 4. Write a line of code that computes the number of words in the entire corpus produc
ed by topic 17 .
produced[:,17,:].sum()
```

Out[24]:

```
893.0
```

In [25]:

```
# 5. Write a line of code that computes the number of words in the entire corpus produc
ed by topic 17 or topic 23.
produced[:,np.array([17,23]),:].sum()
```

Out[25]:

```
3180.0
```

In [33]:

```
# 6. Write a line of code that computes the number of words in the entire corpus produc
ed by even numbered topics.
produced[:,np.arange(0,100,2),:].sum()
```

Out[33]:

```
51543.0
```

In [40]:

```
# 7. Write a line of code that computes the number of each word produced by topic 15.
produced[:,15,:].sum(0)
```

Out[40]:

```
array([0., 0., 3., ..., 0., 2., 3.])
```

In [54]:

```
# 8. Write a line of code that computes the topic responsible for the most instances of
each word in the corpus.
produced.sum(0).sum(1).argmax()
```

Out[54]:

```
23
```

In [56]:

```
# 9. Write a line of code that for each topic, computes the max number of occurrences
 (summed over all documents) of any word that it was responsible for.
# Method 1
produced[:,np.arange(0,100,1),produced.sum (0).argmax(1)].sum(0)
# Method 2
produced.sum(0).max(1)
```

Out[56]:

```
array([25., 63., 25., 29., 31., 43., 23., 36., 40., 49., 25., 17., 36.,
       23., 20., 16., 24., 19., 37., 22., 16., 35., 16., 48., 12., 15.,
       17., 19., 16., 22., 22., 27., 25., 28., 30., 14., 33., 24., 28.,
       23., 33., 17., 21., 54., 55., 33., 12., 18., 45., 27., 12., 16.,
       49., 10., 13.,  9., 20., 17., 27., 19., 28., 17., 19., 13., 30.,
       15., 38., 13., 20., 18., 16., 35., 25.,  6., 39.,  7., 26., 13.,
       17., 12., 19., 41., 13., 21., 33., 13., 21., 21., 14., 30., 34.,
       19., 24., 30., 18., 23., 23., 16., 19., 24.])
```