Question 2

In [1]:
```
# ECE 561 - Jackson Hellmers

# Parts A, B and C
in_string = "this deluge of data calls for automated methods of data analysis which is
list_of_strings = in_string.split()
print(list_of_strings[::3])  #fix me to answer the question
print(list_of_strings[-16])  #fix me to answer the question
```

```
['this', 'data', 'automated', 'data', 'is', 'learning', 'particular', 'machine', 'a', 'm
ethods', 'automatically', 'in', 'then', 'uncovered', 'predict', 'or', 'other', 'decisio
n', 'uncertainty']
uncovered
```

In [2]:
```
# Part D
tuple_of_strings = tuple(list_of_strings)
print(tuple_of_strings)

# A tuple is a built-in data type that unlike a list is immutable. This means that if y
# list but do not want the elements to change you should use a tuple.
```

```
('this', 'deluge', 'of', 'data', 'calls', 'for', 'automated', 'methods', 'of', 'data',
'analysis', 'which', 'is', 'what', 'machine', 'learning', 'provides', 'in', 'particula
r', 'we', 'define', 'machine', 'learning', 'as', 'a', 'set', 'of', 'methods', 'that', 'c
an', 'automatically', 'detect', 'patterns', 'in', 'data', 'and', 'then', 'use', 'the',
'uncovered', 'patterns', 'to', 'predict', 'future', 'data', 'or', 'to', 'perform', 'othe
r', 'kinds', 'of', 'decision', 'making', 'under', 'uncertainty')
```

In [3]:
```
# Part E
set_of_strings = set(list_of_strings)
print("Number of items in set:",len(set_of_strings))
print("Set Length equal to List Length:",len(list_of_strings)==len(set_of_strings))

# Sets do not allow duplicates so repeated words are removed resulting in different len
# while a list is ordered.
```

```
Number of items in set: 43
Set Length equal to List Length: False
```

In [19]:
```
# Part F
d = {}  #create an empty dictionary
for i in list_of_strings:
    if i not in d:
        d[i]=0
    #d[i] =  list_of_strings.count(i)
    d[i] = d[i] + 1

print(d)
occur = max(d, key=d.get)
print("Most Frequent Word:",occur,"- occuring",d[occur],"times")
```

```
{'this': 1, 'deluge': 1, 'of': 4, 'data': 4, 'calls': 1, 'for': 1, 'automated': 1, 'meth
ods': 2, 'analysis': 1, 'which': 1, 'is': 1, 'what': 1, 'machine': 2, 'learning': 2, 'pr
ovides': 1, 'in': 2, 'particular': 1, 'we': 1, 'define': 1, 'as': 1, 'a': 1, 'set': 1,
'that': 1, 'can': 1, 'automatically': 1, 'detect': 1, 'patterns': 2, 'and': 1, 'then':
1, 'use': 1, 'the': 1, 'uncovered': 1, 'to': 2, 'predict': 1, 'future': 1, 'or': 1, 'per
form': 1, 'other': 1, 'kinds': 1, 'decision': 1, 'making': 1, 'under': 1, 'uncertainty':
1}
Most Frequent Word: of - occuring 4 times
```

In [5]:

```
# Part H
list_of_tuples = [(k,v) for k,v in d.items()]  #this line uses a list comprehension
print(list_of_tuples)
```

```
[('this', 1), ('deluge', 1), ('of', 4), ('data', 4), ('calls', 1), ('for', 1), ('automat
ed', 1), ('methods', 2), ('analysis', 1), ('which', 1), ('is', 1), ('what', 1), ('machin
e', 2), ('learning', 2), ('provides', 1), ('in', 2), ('particular', 1), ('we', 1), ('def
ine', 1), ('as', 1), ('a', 1), ('set', 1), ('that', 1), ('can', 1), ('automatically',
1), ('detect', 1), ('patterns', 2), ('and', 1), ('then', 1), ('use', 1), ('the', 1), ('u
ncovered', 1), ('to', 2), ('predict', 1), ('future', 1), ('or', 1), ('perform', 1), ('ot
her', 1), ('kinds', 1), ('decision', 1), ('making', 1), ('under', 1), ('uncertainty',
1)]
```

Question 3

In [22]:

```
# Part A, B and C
import numpy as np
import matplotlib.pyplot as plt
A = [[1,1,3],[4,4,4],[5,6,9]]
A = np.array(A)
A_inv = np.linalg.inv(A)

print(A_inv)

### Write code below to answer question
```

```
[[ 1.5    1.125 -1.   ]
 [-2.    -0.75   1.   ]
 [ 0.5   -0.125  0.   ]]
```

In [25]:

```
# Part D
rand_10 = np.random.rand(10)
print(rand_10)
```

```
[0.82662575 0.57225401 0.97269111 0.79290621 0.44398145 0.16446366
 0.6624094  0.07943787 0.7917924  0.34098034]
```

In [26]:

```
# Part E
n_vals = [1,2,3,4,5,6,7,8]
min_rand_vals = [min(np.random.rand(pow(10,n))) for n in n_vals]
print(min_rand_vals)
```

```
[0.12306193110957686, 0.011118579783409022, 0.0016560753011836127, 0.000139496000994054
9, 5.800378605202994e-07, 1.115730806189319e-06, 1.369673865436738e-07, 1.8974449678133e
-08]
```

In [30]:

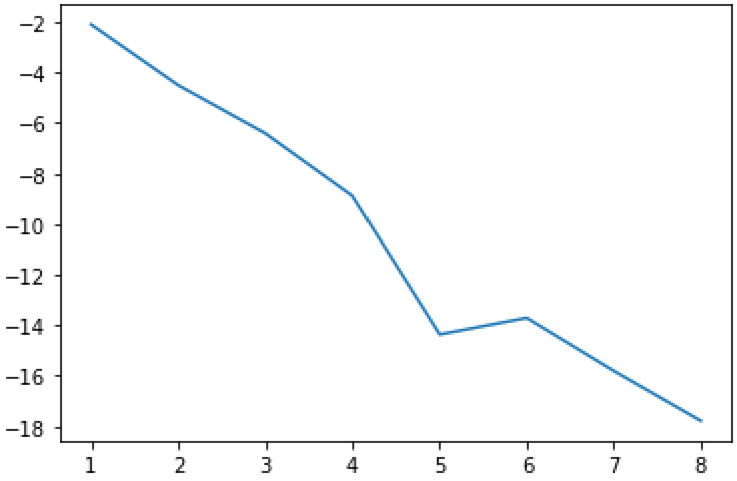
```
# Part E Continued
plt.plot(n_vals,np.log(min_rand_vals))

# I expected the graph to trend downward so that seems correct. I am unsure if the loga
# supposed to be linear or not but if n=10^5 is disregarded the trend appears very line

# The downward trend is expeted as the larger value of n allows for more random samples
# truly random than having more samples would give more possibilty for a small number t
```

Out[30]:

```
[<matplotlib.lines.Line2D at 0x2150002efa0>]
```



In [ ]: