In [25]:

```
import json

messages = [json.loads(line) for line in open('prep_enron_1000_emails_
in_json_file')]
```

In [26]:

```
messages[:2]
```

Out[26]:

```
[{u'_id': {u'$oid': u'4f16fdbdd1e2d323710589fc'},
  u'body': u'Attached is a draft of the Willamette Industries Master
Agreement as requested.\n\n \n\n -----Original Message-----\nFrom: \
tWilliams, Jason R (Credit)  \nSent:\tTuesday, November 06, 2001 7:2
0 PM\nTo:\tPerlingiere, Debra\nCc:\tFuller, Dave\nSubject:\tWillamet
te Industries\n\nDebra -\n\nPlease prepare a draft ENFOLIO per the a
ttached:\t\n\t\n << File: Willamette Master Firm 11062001.xls >> \n\
n\n\nDave -\n\nCan you please contact Debra with the name/email addr
ess of the contact at Willamette?\n\n\n\n\n\nThanks to both,\n\nJay'
,
  u'headers': {u'Date': u'Thu, 15 Nov 2001 14:01:53 -0800 (PST)',
   u'From': u'debra.perlingiere@enron.com',
   u'Message-ID': u'<5141907.1075861285797.JavaMail.evans@thyme>',
   u'Subject': u'RE: Willamette Industries',
   u'To': u'credit <.williams@enron.com>',
   u'X-From': u'Perlingiere, Debra </O=ENRON/OU=NA/CN=RECIPIENTS/CN=
DPERLIN>',
   u'X-To': u'Williams, Jason R (Credit) </O=ENRON/OU=NA/CN=RECIPIEN
TS/CN=Jwilli10>',
   u'X-bcc': u'',
   u'X-cc': u'Fuller, Dave </O=ENRON/OU=NA/CN=RECIPIENTS/CN=Dfuller>
'},
  u'mailbox': u'perlingiere-d',
  u'subFolder': u'sent_items'},
 {u'_id': {u'$oid': u'4f16fdbdd1e2d323710589fd'},
  u'body': u' \n  Thanks for our email.  At the present time, Ena is
not putting new GISBs in place.  However, we have a Enfolio Spot Agr
eement which will achieve the same purpose.  Please see attached.\n\
n   \n\n -----Original Message-----\nFrom: \tAnthony Targan <targana
@dteenergy.com>@ENRON  \nSent:\tThursday, November 15, 2001 8:35 AM\
nTo:\tdperlin@enron.com\nSubject:\tGISB contract with DTE Energy Tra
ding\n\nDebra,\n\nIt was a pleasure speaking with you yesterday rega
rding the GISB\ncontract.\nI look forward to receipt of your suggest
ed language for the netting and\ndispute resolution provisions, and
your decision on the acceptability of\nour Sections 10.3, 10.4, and
15 (Confidentiality).\n\nPlease do not hesitate to call if we need t
o discuss this further.\nUnless you instruct otherwise, we intend to
sign the Base Contract\nfirst, and then FedEx partially executed ori
ginals to you for signature.\n\nThanks,\n--Anthony\n\n - targana.vcf
<< File: targana.vcf >> ',
  u'headers': {u'Date': u'Thu, 15 Nov 2001 16:06:30 -0800 (PST)',
   u'From': u'debra.perlingiere@enron.com',
   u'Message-ID': u'<17383820.1075861285819.JavaMail.evans@thyme>',
   u'Subject': u'RE: GISB contract with DTE Energy Trading',
   u'To': u'targana@dteenergy.com',
   u'X-From': u'Perlingiere, Debra </O=ENRON/OU=NA/CN=RECIPIENTS/CN=
DPERLIN>',
   u'X-To': u"'Anthony Targan <targana@dteenergy.com>@ENRON'",
   u'X-bcc': u'',
   u'X-cc': u''},
  u'mailbox': u'perlingiere-d',
  u'subFolder': u'sent_items'}]
```

In [27]:
```
type(messages)
```

Out[27]:
```
list
```

In [28]:
```
type(messages[0])
```

Out[28]:
```
dict
```

In [29]:
```
type(messages[0]['headers'])
```

Out[29]:
```
dict
```

In [30]:
```
type(messages[0]['body'])
```

Out[30]:
```
unicode
```

In [31]:
```
list_of_emails_dict_data = []

for message in messages:
    tmp_my_record_flattened_parent_dict = message
    tmp_my_record_flattened_child_dict = message['headers']
    del tmp_my_record_flattened_parent_dict['headers']
    del tmp_my_record_flattened_parent_dict['_id']
    tmp_my_record_flattened_parent_dict.update(tmp_my_record_flattened
_child_dict)
    list_of_emails_dict_data.append(tmp_my_record_flattened_parent_dic
t.copy())
```

In [32]:
```
#Sanity test
list_of_emails_dict_data[:2]
```

Out[32]:
```
[{u'Date': u'Thu, 15 Nov 2001 14:01:53 -0800 (PST)',
  u'From': u'debra.perlingiere@enron.com',
  u'Message-ID': u'<5141907.1075861285797.JavaMail.evans@thyme>',
  u'Subject': u'RE: Willamette Industries',
  u'To': u'credit <.williams@enron.com>',
  u'X-From': u'Perlingiere, Debra </O=ENRON/OU=NA/CN=RECIPIENTS/CN=D
PERLIN>',
  u'X-To': u'Williams, Jason R (Credit) </O=ENRON/OU=NA/CN=RECIPIENT
S/CN=Jwilli10>',
  u'X-bcc': u'',
  u'X-cc': u'Fuller, Dave </O=ENRON/OU=NA/CN=RECIPIENTS/CN=Dfuller>'
,
  u'body': u'Attached is a draft of the Willamette Industries Master
Agreement as requested.\n\n \n\n -----Original Message-----\nFrom: \
tWilliams, Jason R (Credit)  \nSent:\tTuesday, November 06, 2001 7:2
0 PM\nTo:\tPerlingiere, Debra\nCc:\tFuller, Dave\nSubject:\tWillamet
te Industries\n\nDebra -\n\nPlease prepare a draft ENFOLIO per the a
ttached:\t\n\t\n << File: Willamette Master Firm 11062001.xls >> \n\
n\n\nDave -\n\nCan you please contact Debra with the name/email addr
ess of the contact at Willamette?\n\n\n\n\n\nThanks to both,\n\nJay'
,
  u'mailbox': u'perlingiere-d',
  u'subFolder': u'sent_items'},
 {u'Date': u'Thu, 15 Nov 2001 16:06:30 -0800 (PST)',
  u'From': u'debra.perlingiere@enron.com',
  u'Message-ID': u'<17383820.1075861285819.JavaMail.evans@thyme>',
  u'Subject': u'RE: GISB contract with DTE Energy Trading',
  u'To': u'targana@dteenergy.com',
  u'X-From': u'Perlingiere, Debra </O=ENRON/OU=NA/CN=RECIPIENTS/CN=D
PERLIN>',
  u'X-To': u"'Anthony Targan <targana@dteenergy.com>@ENRON'",
  u'X-bcc': u'',
  u'X-cc': u'',
  u'body': u' \n  Thanks for our email.  At the present time, Ena is
not putting new GISBs in place.  However, we have a Enfolio Spot Agr
eement which will achieve the same purpose.  Please see attached.\n\
n   \n\n -----Original Message-----\nFrom: \tAnthony Targan <targana
@dteenergy.com>@ENRON  \nSent:\tThursday, November 15, 2001 8:35 AM\
nTo:\tdperlin@enron.com\nSubject:\tGISB contract with DTE Energy Tra
ding\n\nDebra,\n\nIt was a pleasure speaking with you yesterday rega
rding the GISB\ncontract.\nI look forward to receipt of your suggest
ed language for the netting and\ndispute resolution provisions, and
your decision on the acceptability of\nour Sections 10.3, 10.4, and
15 (Confidentiality).\n\nPlease do not hesitate to call if we need t
o discuss this further.\nUnless you instruct otherwise, we intend to
sign the Base Contract\nfirst, and then FedEx partially executed ori
ginals to you for signature.\n\nThanks,\n--Anthony\n\n - targana.vcf
<< File: targana.vcf >> ',
  u'mailbox': u'perlingiere-d',
  u'subFolder': u'sent_items'}]
```

In [33]:

```
from pandas import DataFrame
enron_email_df = DataFrame(list_of_emails_dict_data)
```

In [34]:

```
# Sanity Test
enron_email_df.head()
```

Out[34]:

| | Date | From | Message-ID |
|---|---|---|---|
| **0** | Thu, 15 Nov 2001 14:01:53 -0800 (PST) | debra.perlingiere@enron.com | <5141907.1075861285797.JavaMail.evans@thym |
| **1** | Thu, 15 Nov 2001 16:06:30 -0800 (PST) | debra.perlingiere@enron.com | <17383820.1075861285819.JavaMail.evans@thy |
| **2** | Thu, 15 Nov 2001 16:07:18 -0800 (PST) | debra.perlingiere@enron.com | <11433168.1075861285841.JavaMail.evans@thy |
| **3** | Tue, 27 Nov 2001 08:51:53 -0800 (PST) | debra.perlingiere@enron.com | <24587012.1075861285863.JavaMail.evans@thy |
| **4** | Wed, 10 Oct 2001 12:51:21 -0700 (PDT) | debra.perlingiere@enron.com | <21707489.1075852243982.JavaMail.evans@thy |

In [35]:

```
# Loop through dataframe of messages and calculate the lengeth of every field in Bytes
# You could add any number of fields you are interested in; following calculed lengths for only 4 fields
# data will have a list of dictionaries; a dictionary for every message,
# the octionary will have the length of every field for every message

enron_email_df.fillna("", inplace=True)


data = []

for i in enron_email_df.index:
    message_data = {}
    message_data['From'] = len(enron_email_df.ix[i]['From'])
    message_data['To'] = len(enron_email_df.ix[i]['To'])
    message_data['Subject'] = len(enron_email_df.ix[i]['Subject'])
    message_data['body'] = len(enron_email_df.ix[i]['body'])
    data.append(message_data)
```

In [36]:

```
#Now create a data frame such that we can get summary statistics about the messages and the fields

enron_email_field_lengths_df = DataFrame(data)
```

In [37]:

```
enron_email_field_lengths_df.head()
```

Out[37]:

| | From | Subject | To | body |
|---|---|---|---|---|
| **0** | 27 | 25 | 28 | 495 |
| **1** | 27 | 41 | 21 | 936 |
| **2** | 27 | 0 | 27 | 157 |
| **3** | 27 | 26 | 17 | 2149 |
| **4** | 27 | 0 | 93 | 212 |

## Basic stat summary of all messages in the sample collection using describe

In [38]:

```
enron_email_field_lengths_df.describe()
```

Out[38]:

| | **From** | **Subject** | **To** | **body** |
|---|---|---|---|---|
| **count** | 1000.000000 | 1000.00000 | 1000.000000 | 1000.000000 |
| **mean** | 23.032000 | 33.37900 | 248.645000 | 2486.009000 |
| **std** | 3.383848 | 21.00263 | 948.659244 | 5873.329993 |
| **min** | 13.000000 | 0.00000 | 0.000000 | 2.000000 |
| **25%** | 20.000000 | 17.00000 | 22.000000 | 281.000000 |
| **50%** | 23.000000 | 31.00000 | 24.000000 | 778.000000 |
| **75%** | 25.000000 | 48.00000 | 82.000000 | 1967.000000 |
| **max** | 45.000000 | 132.00000 | 15292.000000 | 46100.000000 |

## What is total size(bytes) for every field of all messages in the sample collection?

In [39]:

```
enron_email_field_lengths_df.sum(axis=0)
```

Out[39]:

```
From          23032
Subject       33379
To           248645
body        2486009
dtype: int64
```

## What is the size (bytes) of every message in the set of first 10 messages in the sample collection (all fields included)?

```
In [40]: enron_email_field_lengths_df[:10].sum(axis=1)
```

```
Out[40]: 0     575
         1    1025
         2     211
         3    2219
         4     332
         5    1036
         6    1481
         7     902
         8     392
         9     345
         dtype: int64
```

## What is the minimum message size of ALL messages in the collection (all fields included)?

```
In [41]: enron_email_field_lengths_df.sum(axis=1).min(axis=0)
```

```
Out[41]: 43
```

## Deliverable #1: What is the mean message size of all messages in the collection (all fields included)?

```
In [42]: enron_email_field_lengths_df.sum(axis=1).mean(axis=0)
```

```
Out[42]: 2791.0650000000001
```

## Deliverable #2: What is the max message size of all messages in the collection (all fields included)?

```
In [43]: enron_email_field_lengths_df.sum(axis=1).max(axis=0)
```

```
Out[43]: 46295
```

## Deliverable #3: What is the median size for every field of all messages?

```
In [44]: enron_email_field_lengths_df.median(axis=0)

Out[44]: From        23.0
         Subject     31.0
         To          24.0
         body       778.0
         dtype: float64
```

## Deliverable #4: Do you think of any metric/stat that must be considered in the analysis? show data output for it.

```
In [45]: enron_email_field_lengths_df.std(axis=0)

Out[45]: From          3.383848
         Subject      21.002630
         To          948.659244
         body       5873.329993
         dtype: float64
```

```
In [ ]:
```