Handout for Session 7 (with Solutions)

1. For loops and dictionaries

```
[1]: # Iterating through a dictionary
     d={'apple':5,'rice':4,'broccoli':8}
     for key in d:
         value=d[key]
         print(key,value)
```

```
apple 5
rice 4
broccoli 8
```

```
[2]: # Printing the dictionary in alphabetical order
     for key in sorted(d.keys()):
         print(key,d[key])
```

```
apple 5
broccoli 8
rice 4
```

```
[3]: # Building a dictionary iteratively
     l=['apple','rice','broccoli']
     d={}
     for item in l:
         d[item]=len(item)
     d
```

```
{'apple': 5, 'rice': 4, 'broccoli': 8}
```

Q1: Given the following dictionaries countaining word counts (`total` and `current`), use a `for` loop to iterate through the dictionary `current` and add the counts to the dictionary `total`. (If the word is not found in `total`, you have to first initialize the value in `total` to zero before adding.)

```
[4]: total={'happy':51,'cheap':30}
     current={'happy':2,'amazing':1,'price':2}
```

```
[5]: for key in current:
         if key not in total:
             total[key]=0
         total[key]+=current[key]
     total
```

```
{'happy': 53, 'cheap': 30, 'amazing': 1, 'price': 2}
```

2. Breaking Down Case 7a from Last Session (4 Step Method)

Step 1: Describe the task succintly and precisely

Obtain a list of unique domain names from a specified mail log, and print the list in alphabetical order.

Step 2: Decompose the task into components and describe how to do each in English

A. Traverse through the mail log and filter for lines starting with "From:"
B. Obtain the domain name from each line.
C. Maintain a list of unique domain names (using Q4 from last session).
D. Sort the list and print the elements.

Step 3: Translate each component into code and test them independently

```
[6]: # A. Traverse through the mail log and filter for lines starting with "From:"...

     file=open('mbox-short.txt','r')
     for line in file:
         line=line.strip()
         if line.startswith("From:"):
             print(line)
```

```
From: stephen.marquard@uct.ac.za
From: louis@media.berkeley.edu
From: zqian@umich.edu
From: rjlowe@iupui.edu
From: zqian@umich.edu
From: rjlowe@iupui.edu
From: cwen@iupui.edu
From: cwen@iupui.edu
From: gsilver@umich.edu
From: gsilver@umich.edu
From: zqian@umich.edu
From: gsilver@umich.edu
From: wagnermr@iupui.edu
From: zqian@umich.edu
From: antranig@caret.cam.ac.uk
From: gopal.ramasammycook@gmail.com
From: david.horwitz@uct.ac.za
From: david.horwitz@uct.ac.za
From: david.horwitz@uct.ac.za
From: david.horwitz@uct.ac.za
From: stephen.marquard@uct.ac.za
From: louis@media.berkeley.edu
From: louis@media.berkeley.edu
From: ray@media.berkeley.edu
From: cwen@iupui.edu
From: cwen@iupui.edu
From: cwen@iupui.edu
```

```
[7]: # B. Obtain the domain name from each line
     line='From: stephen.marquard@uct.ac.za'
     domain=line.split('@')[1]
     domain
```

```
'uct.ac.za'
```

```
[8]: # C. Maintain a list of unique domain names
     l=['berkeley.edu']
     domain='uct.ac.za'
     if domain not in l:
         l.append(domain)
     l
```

```
['berkeley.edu', 'uct.ac.za']
```

```
[9]: domain='uct.ac.za'
     if domain not in l:
         l.append(domain)
     l
```

```
['berkeley.edu', 'uct.ac.za']
```

```
[10]: # D. Sort the list and print the elements
      l=['c','a','b']
      l=sorted(l)
      for e in l:
          print(e)
```

```
a
b
c
```

Step 4: Combine Together and Test

i) Copy paste all the code together

```
# A. Traverse through the mail log and filter for lines starting with "From:"...
file=open('mbox-short.txt','r')
for line in file:
    line=line.strip()
    if line.startswith("From:"):
        print(line)

# B. Obtain the domain name from each line
line='From: stephen.marquard@uct.ac.za'
domain=line.split('@')[1]

# C. Maintain a list of unique domain names
l=['berkeley.edu']
domain='uct.ac.za'
if domain not in l:
    l.append(domain)

# D. Sort the list and print the elements
l=['c','a','b']
l=sorted(l)
for e in l:
    print(e)
```

ii) Review the logical relationship based on the English descriptions in Step 2: Component B and C are inside the loop of component A (except that the initialization of the list should come first). Component D should take place afterward.

iii) Combine the code appropriately and test.

```
[11]: l=[]  # C1. Initialize the list of unique elements

      # A. Traverse through the mail log and filter for lines starting with "From:"...
      file=open('mbox-short.txt','r')
      for line in file:
          line=line.strip()
          if line.startswith("From:"):
              domain=line.split('@')[1]  # B. Obtain the domain name from each line
              if domain not in l:        # C2. Maintain the list of unique elements
                  l.append(domain)

      # D. Sort and print
      for e in sorted(l):
          print(e)
```

```
caret.cam.ac.uk
gmail.com
iupui.edu
media.berkeley.edu
uct.ac.za
umich.edu
```

Q2: Apply the above 4 step method to solve case 7b) from last session.
Solution:
Describe: Obtain the count of each domain from the "From" lines of a mail log, and print the counts.
Decompose:
A: Traverse through and filter for the `"From"` lines (same as in 7a).
B: Obtain the domain name of each line (same as in 7a).
C: Maintain the count of each word using a dictionary (Q6 from last session).
D: Print the dictionary.
Translate:

```
[12]: # C: Maintain the count of each word
      count={'berkeley.edu':1}
      domain='uct.ac.za'
      if domain not in count:
          count[domain]=0
      count[domain]+=1
      count
```

```
{'berkeley.edu': 1, 'uct.ac.za': 1}
```

```
[13]: domain='uct.ac.za'
      if domain not in count:
          count[domain]=0
```

```
        count[domain]+=1
    count
```

```
{'berkeley.edu': 1, 'uct.ac.za': 2}
```

```
[14]: # D: Print the dictionary
      count={'berkeley.edu': 1, 'uct.ac.za': 2}
      for domain in sorted(count.keys()):
          print(domain,count[domain])
```

```
berkeley.edu 1
uct.ac.za 2
```

Combine:

```
[15]: filename='mbox-short.txt'
      file=open(filename,'r')
      count={}
      for line in file:
          line=line.rstrip()
          if line.startswith('From:'):
              domain=line.split('@')[1]
              if domain not in count:
                  count[domain]=0
              count[domain]+=1
      for domain in sorted(count.keys()):
          print(domain,count[domain])
```

```
caret.cam.ac.uk 1
gmail.com 1
iupui.edu 8
media.berkeley.edu 4
uct.ac.za 6
umich.edu 7
```

3. Pandas DataFrame Basics

```
[16]: dic1={'orange':6,'grape':5,'apple':5}
      dic2={'apple':'M','grape':'S','orange':'M'}
```

```
[17]: import pandas as pd
      df=pd.DataFrame({'Number of Letters':dic1,'Size':dic2})
      df
```

```
        Number of Letters Size
apple                   5    M
grape                   5    S
orange                  6    M
```

```
[18]: df.sort_values(by='Number of Letters',ascending=False)
```

```
        Number of Letters Size
orange                  6    M
apple                   5    M
grape                   5    S
```

```
[19]: df.sort_values(by=['Number of Letters','Size'],ascending=[False,True])
```

```
        Number of Letters Size
orange                  6    M
apple                   5    M
grape                   5    S
```

```
[20]: df
```

```
        Number of Letters Size
apple                   5    M
grape                   5    S
orange                  6    M
```

```
[21]: df['Rank']=[1,3,2]
```

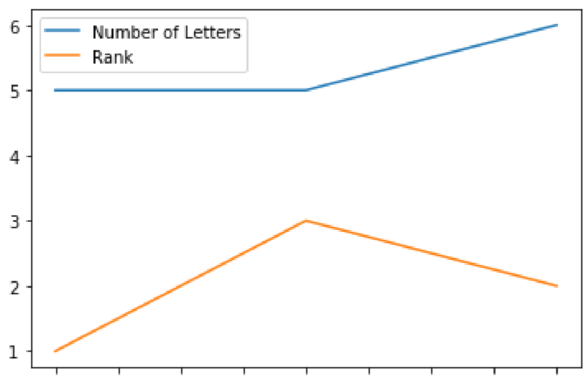
```
[22]: df.head(2)
```

```
       Number of Letters Size  Rank
apple                  5    M     1
grape                  5    S     3
```

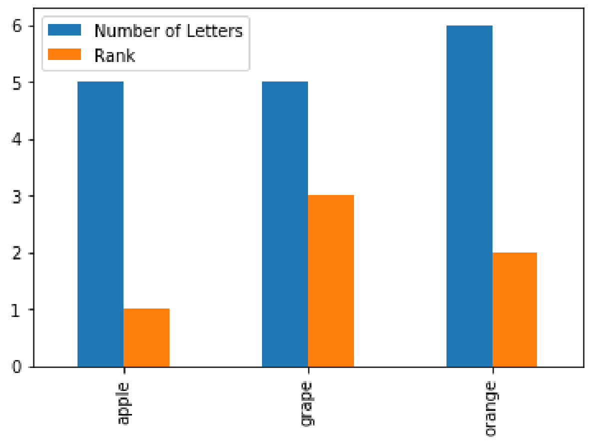
```
[23]: df.tail(1)
```

```
        Number of Letters Size  Rank
orange                  6    M     2
```

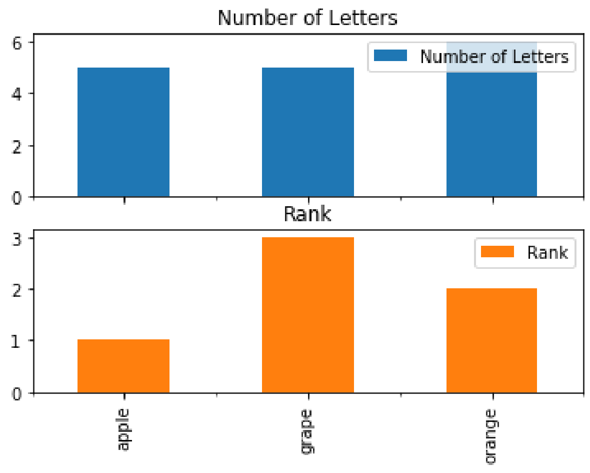
```
[35]: import matplotlib.pyplot as plt
      df.plot()
      plt.show()
```



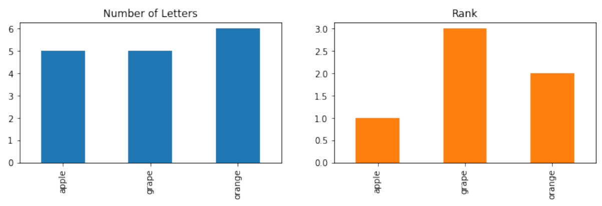
```
[25]: df.plot(kind='bar')
      plt.show()
```



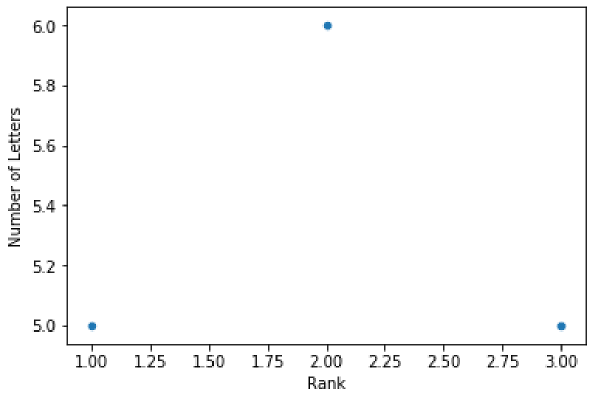
```
[26]: df.plot(kind='bar',subplots=True)
      plt.show()
```



```
[27]: df.plot(kind='bar',subplots=True,figsize=(12,3),legend=False,layout=(1,2))
      plt.show()
```



```
[28]: df.plot(x='Rank',y='Number of Letters',kind='scatter')
      plt.show()
```



```
[29]: df.to_csv('session7_output.csv')
```

```
[30]: pd.read_csv('session7_output.csv',index_col=0)
```

```
        Number of Letters Size  Rank
apple                   5    M     1
grape                   5    S     3
orange                  6    M     2
```

Q3-a: Create the following DataFrame and name it phones.

```
[31]: products=pd.DataFrame({'price':{'iPhone XR':749,'Samsung S9':619,'iPhone 8':599},\
                             'screen size':{'iPhone XR':6.1,'Samsung S9':5.8,'iPhone 8':4.7}}
      products
```

```
            price  screen size
Samsung S9    619          5.8
iPhone 8      599          4.7
iPhone XR     749          6.1
```

Q3-b: Sort the columns in descending order by screen size.

```
[32]: products.sort_values(by='screen size',ascending=False)
```

```
            price  screen size
iPhone XR     749          6.1
Samsung S9    619          5.8
iPhone 8      599          4.7
```

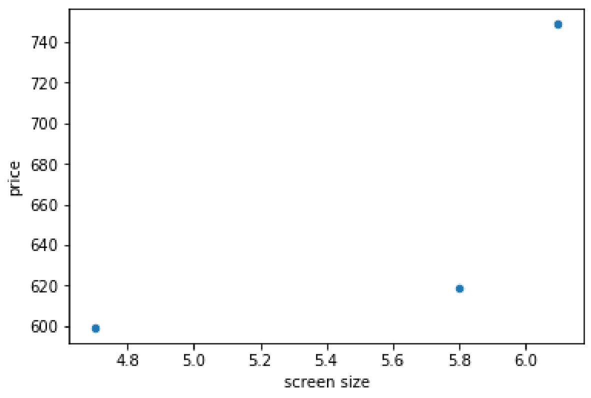
Q3-c: Obtain only the first two rows of the DataFrame (after sorting by screen size).

```
[33]: products.sort_values(by='screen size',ascending=False).head(2)
```

```
            price  screen size
iPhone XR     749          6.1
Samsung S9    619          5.8
```

Q3-d: Create a scatter plot where x axis is screen size and y axis is price.

```
[34]: import matplotlib.pyplot as plt
      products.plot(x='screen size',y='price',kind='scatter')
      plt.show()
```


price
740
720
700
680
660
640
620
600
4.8
5.0
5.2
5.4
5.6
5.8
6.0
screen size