Anamitra Bhattacharyya
Predict 420-DL, Section 55
Assignment 3 (May 8, 2016)

1) Script grEx3.log output ('Getting Data from the SSCC')

```
Script started on Wed 27 Apr 2016 08:18:56 PM CDT
]0;abv902@dornick:~/graded_exercise_3[?1034h[abv902@dornick graded_exercise_3]$ exit[2Pllpsql -h spspostgresql -U abv902 -d postgres[1Pnetid -d postgres[1@abv902 -d postgres
psql (8.4.20, server 9.4.1)
WARNING: psql version 8.4, server version 9.4.
         Some psql features might not work.
Type "help" for help.

[?1034hpostgres=> \qcopy (SELECT * FROM abv902work) TO 'abv902work.csv' WITH DELIMITER ',' NULL AS '\null' CSV HEADER
postgres=> [3PCREATE TEMP VIEW abv902work as SELECT * FROM pilot.item WHERE extract(month FROM trandate) = 1;
postgres=> \c xyz[K
psql (8.4.20, server 9.4.1)
WARNING: psql version 8.4, server version 9.4.
         Some psql features might not work.
You are now connected to database "xyz".
xyz=> CREATE TEMP VIEW netidwork as SELECT * FROM pilot.item[1P[1P[1P[1P[1P[1P[1P[1P[1P[1@a[1@b[1@v[1@9[1@0[1@2[1@i[1@t[1@e[1@m aas SELECT * FROM pilot.item;
CREATE VIEW
xyz=> \copy (SELECT * FROM netidwork) TO
'netidwork.csv'rk[1P[1P[1P[1P[1P[1P[1P[1P[1P[1@a[1@b[1@v[1@9[1@0[1@2[1@i[1@t[1@e[1@m[1P[1P[1P[1P[1P[1P[1P[1P[1P[1@a[1@b[1@v[1@9[1@0[1@2[1@i[1@t[1@e[1@m) TO 'abv902item.csv' WITH DELIMITER ','
NULL AS '\null' CSV HEADER
xyz=> \copy (SELECT * FROM abv902item) TO 'abv902item.csv' WITH DELIMITER ',' NULL AS '\null' CSV HEADER
xyz=> [42PCREATE TEMP VIEW abv902item as SELECT * FROM pilot.item;ot.item;[1P;[1P;[1P;[1P;c;u;s;t;o;m;e;r;[1P[1P[1P[1P[1@c[1@u[1@s[1@t[1@o[1@m[1@e[1@r as SELECT * FROM pilot.customer;
CREATE VIEW
xyz=> CREATE TEMP VIEW abv902customer as SELECT * FROM pilot.customer;
xyz=> \copy (SELECT * FROM abv902item) TO 'abv902item.csv' WITH DELIMITER ',' NULL AS '\null' CSV
HEADER[1P[1P[1P[1P[1@c[1@u[1@s[1@t[1@o[1@m[1@e[1@rm[1P[1P[1P[1P[1@c[1@u[1@s[1@t[1@o[1@m[1@e[1@r) TO 'abv902customer.csv' WITH DELIMITER ',' NULL AS '\null' CSV HEADER
xyz=> \copy (SELECT * FROM abv902customer) TO 'abv902customer.csv' WITH DELIMITER ',' NULL AS '\null' CSV HEADER
xyz=> [42PCREATE TEMP VIEW abv902customer as SELECT * FROM pilot.customer;[1P;[1P;[1P;[1P;[1P;[1P;[1P;[1P;m;a;i;l;[1P[1P[1P[1P[1P[1P[1P[1P[1@m[1@a[1@i[1@l as SELECT * FROM pilot.mail;
CREATE VIEW
xyz=> CREATE TEMP VIEW abv902mail as SELECT * FROM pilot.mail;
xyz=> \copy (SELECT * FROM abv902customer) TO 'abv902customer.csv' WITH DELIMITER ',' NULL AS '\null' CSV
HEADER[1P[1P[1P[1P[1P[1P[1P[1P[1@m[1@a[1@i[1@l[1P[1P[1P[1P[1P[1P[1P[1P[1@m[1@a[1@i[1@l) TO 'abv902mail.csv' WITH DELIMITER ',' NULL AS '\null' CSV HEADER
xyz=> \q
]0;abv902@dornick:~/graded_exercise_3[abv902@dornick graded_exercise_3]$ ls -la
total 83482
drwx------ 2 abv902 users      127 Apr 27 20:26 [0m[38;5;27m.[0m
drwx------ 9 abv902 users      469 Apr 27 20:07 [38;5;27m..[0m
-rw-r--r-- 1 abv902 users 57066474 Apr 27 20:25 abv902customer.csv
-rw-r--r-- 1 abv902 users  5373591 Apr 27 20:22 abv902item.csv
-rw-r--r-- 1 abv902 users  1295146 Apr 27 20:26 abv902mail.csv
-rw------- 1 abv902 users        0 Apr 27 20:18 grEx3.log
abv902@dornick graded_exercise_3]$ exit
exit

Script done on Wed 27 Apr 2016 08:32:10 PM CDT
```

2) Additional Items

After creating the three CSV files (*e.g.* customer, item and mail) and the grEx3 log file on the SPSS server on dornick, I used the Filezilla UI for SFTP to 'drag-and-drop' all the files onto a local directory on my computer, for further processing in Assignment 3.

3) Python Code for ‘Working with the Data’

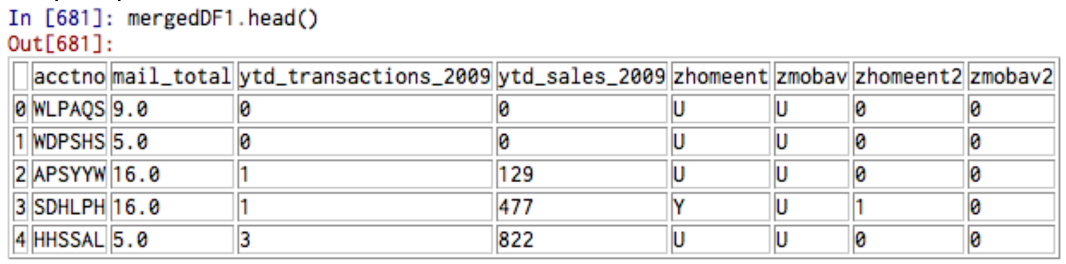
```
import pandas as pd #import panda
import numpy as np
#all files are in the current working directory
import cPickle as pickle

#1) Import each of the csv files you downloaded from the SSCC into a pandas DataFrame.
customerDF=pd.read_csv("abv902customer.csv", sep=',')
mailDF=pd.read_csv("abv902mail.csv", sep=',')
itemDF=pd.read_csv("abv902item.csv", sep=',')

#2) Verify that there are no duplicate customer records in the customer data
len(customerDF)#50,000 records
customerDF.duplicated().sum()# count duplicated records = 0 (no duplicates)

#3) Check the item and mail data to determine if there are any
#records in them for customers who are not in the customers data
len(mailDF); mailDF.duplicated().sum()#duplicates=166
len(itemDF); itemDF.duplicated().sum()#duplicates=6325

#There are no acct numbers in mail & item that are not in customer data (use sets)
mNotc=set(mailDF.acctno)-set(customerDF.acctno)
len(mNotc)# mNotc=0
iNotc=set(itemDF.acctno)-set(customerDF.acctno)
len(iNotc)# iNotc=0

#4)Create a sqlite database, and write the customer, item, and mail data into it as tables
import sqlite3
conn = sqlite3.connect(r"customerDB.db")#created customerDB database to hold tables

import sqlalchemy
from sqlalchemy import create_engine
eng=create_engine('sqlite:///customerDB.db')#specify db you want to use

#write customer, mail and item dataframes to new customerDB
customerDF.to_sql('customer', eng)#add customer table
mailDF.to_sql('mail', eng)#add mail table
itemDF.to_sql('item', eng)#add item table

#Look at metadata from an RDB using SQLAlchemy using the 'inspect' method:
from sqlalchemy import inspect
insp=inspect(eng)
insp.get_table_names()#three tables exist in DB (customer, mail, item)

#5)Create a target file for XYZ Company
#sum mail campaigns
df=mailDF
#Add a column 'mail_total' with customers mailed >= 5 times
```

```
df['mail_total'] = df.sum(axis=1)
df.head()#check that col added
test = df[df.mail_total >= 5]#test contains customers mailed >= 5 times
len(test)#so keep 11,535 customers that've been mailed >= 5 times
(len(mailDF) - len(test))#so discard 19,411 customer records
#keep acctno and total_mail columns, delete the rest of the columns
test.drop(test.columns[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]], axis=1, inplace=True)
#keep customerDF columns defined in specs in a new df called 'df2'
df2 = customerDF[['acctno','ytd_transactions_2009','ytd_sales_2009', 'zhomeent', 'zmobav']]
#Make additional column copies of zhomeent and zmobav
df2['zhomeent2']=df2['zhomeent']
df2['zmobav2']=df2['zmobav']

#Use numpy to convert 'Y' or not 'Y' in these cols to '1' or '0' in-place, respectively
df2['zhomeent2'] = np.where(df2['zhomeent2'] == 'Y', 1, 0)
df2['zmobav2'] = np.where(df2['zmobav2'] == 'Y', 1, 0)
df2.head()#check columns add to specification

#Merge test and df2 dataframes to show all records from both frames
mergedDF1=pd.merge(test, df2, on='acctno', how='outer')

#Then just delete the excess indices from customerDF in range 11535-50142 (end)
mergedDF1.drop(mergedDF1.index[11535::], inplace=True)
len(mergedDF1)#size = 11,535 i.e. customers mailed >= 5 times
mergedDF1.head()

#Create csv file of mergedDF1 (Deliverable)
mergedDF1.to_csv("xyz_directMail.csv")
```

#6)Pickle the direct mail list (mergedDF1)

```
import cPickle as pickle
pickle.dump(mergedDF1,open('xyz_directmail.p', 'wb'))
pickle.dump(mailDF,open('xyz_maildf.p', 'wb'))
pickle.dump(itemDF,open('xyz_itemdf.p', 'wb'))
pickle.dump(customerDF,open('xyz_customerdf.p', 'wb'))
```

Example output for deliverable:



```
In [681]: mergedDF1.head()
Out[681]:
```

	acctno	mail_total	ytd_transactions_2009	ytd_sales_2009	zhomeent	zmobav	zhomeent2	zmobav2
0	WLPAQS	9.0	0	0	U	U	0	0
1	WDPSHS	5.0	0	0	U	U	0	0
2	APSYYW	16.0	1	129	U	U	0	0
3	SDHLPH	16.0	1	477	Y	U	1	0
4	HHSSAL	5.0	3	822	U	U	0	0