Name: NG Hing Yu, Lam Wun Ming
SID: 1155049053, 1155049536

Answer sheet (Please see major findings and explanation from Appendix)

Q1. Average age (Mobile adopter): 35 | Average age (Mobile non-adopter): 38

There tends to be an age difference between the two groups. (See Appendix I Table I)

Q2. Female (Mobile adopter): 48.10% | Female (Mobile non-adopter): 58.45%

There tends to be a gender difference between the two groups. (See Appendix I Table II)

Q3. Mean order price (Mobile): 26654.7 KRW | Mean order price (Online): 33267.4 KRW

Online transactions generated by mobile non-adopter group should not be included, as we are interesting in the difference within mobile adopter group.

Our conjecture is that online transaction has a higher order price. Thus, a one-tailed Welch Two sample t-test is conducted (H_0: online = mobile vs H_1: online > mobile). H_0 is rejected with p = 2.2e-16 < 0.05 in favor of H_1. It suggests that order price of online transaction is higher than mobile transaction. (See Appendix I Table III, IV, Appendix II)

Q4. Confirmation rate (Mobile): 79.97% | Confirmation rate (Online): 85.16%

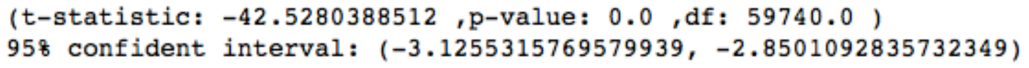
This is consistent with our conjecture that online confirmation rate is higher than mobile one. There may be technical errors during mobile purchase e.g. loss of internet signal, or distraction from other mobile notifications that lead to more incomplete transaction. (See Appendix I Table V)

Q5. Mobile transaction is more dependent on certification. H_0: Certification has no effect on channel choosing, is rejected with p-value 2.858e^08 <0.05. (See Appendix I Table VI)

Name: NG Hing Yu, Lam Wun Ming
SID: 1155049053, 1155049536

Appendix I

Table I



```
(t-statistic: -42.5280388512 ,p-value: 0.0 ,df: 59740.0 )
95% confident interval: (-3.1255315769579939, -2.8501092835732349)
```

Q1. A Welch two-sample t-test was performed with t(59740.0) = -42, p = 0.0

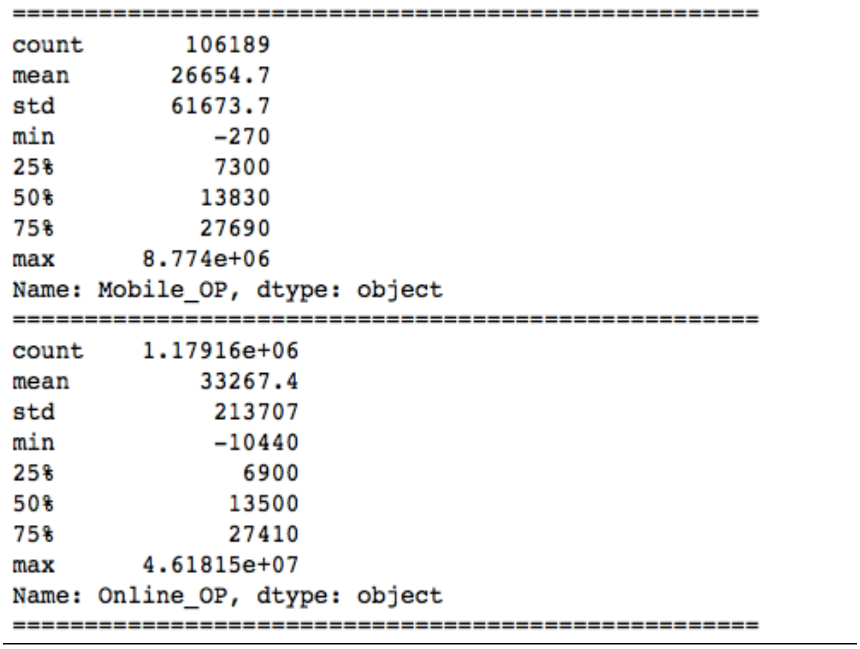
∴ (H_0: no mean difference between 2 groups) is rejected as p-value is significantly small and t-statistics significantly deviates from zero.

Table II: Contingency table

	Male = 1	Female = 0
Mobile non-adopter = 1	12388	17426
Mobile adopter = 0	15533	14395

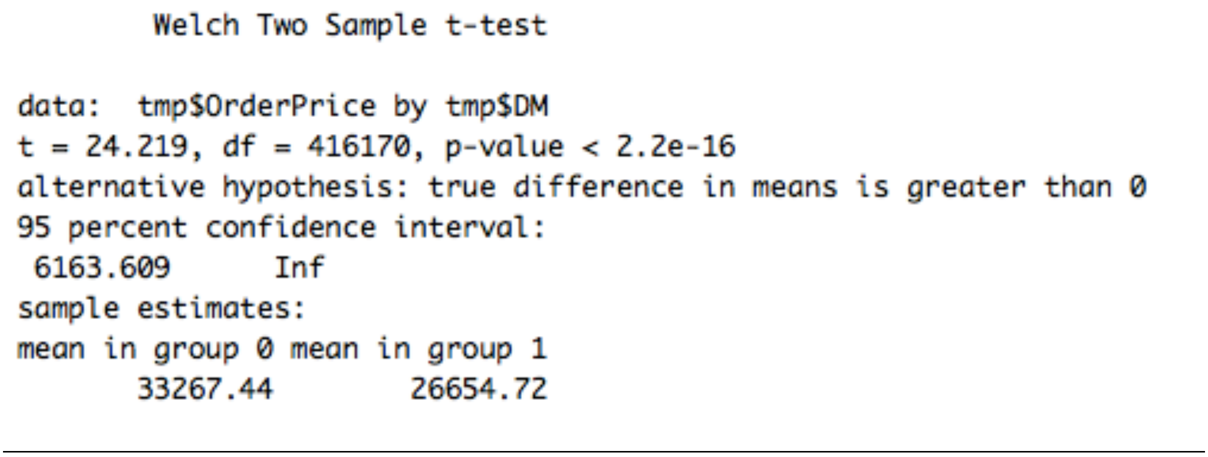
Q2. A chi-square test with Yates' continuity correction was performed to examine the gender difference. (H_0: no gender difference difference between 2 groups) is rejected, X^2 = 642, p = 1.041983e^(-141) < 0.05. Phi coefficient -0.1037 might indicate that male are more likely to adopt mobile channel.

Table III:



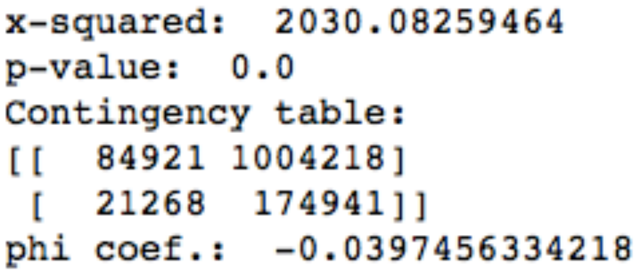
```
=====================================================
count       106189
mean       26654.7
std        61673.7
min           -270
25%           7300
50%          13830
75%          27690
max      8.774e+06
Name: Mobile_OP, dtype: object
=====================================================
count    1.17916e+06
mean         33267.4
std           213707
min           -10440
25%             6900
50%            13500
75%            27410
max      4.61815e+07
Name: Online_OP, dtype: object
=====================================================
```

Table IV



```
        Welch Two Sample t-test

data:  tmp$OrderPrice by tmp$DM
t = 24.219, df = 416170, p-value < 2.2e-16
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 6163.609      Inf
sample estimates:
mean in group 0 mean in group 1
       33267.44        26654.72
```

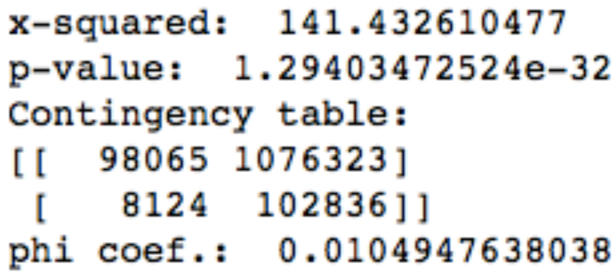
Table V:



```
x-squared:  2030.08259464
p-value:  0.0
Contingency table:
[[  84921 1004218]
 [  21268  174941]]
phi coef.:  -0.0397456334218
```

Q4. A Pearson’s Chi-squared test with Yates’ continuity correction is performed to investigate the correlation between confirmation rate and channel choosing. H_0: Channel choosing do not have impact on confirmation, is rejected with $X^2 = 2030.08$, $p = 0.0 < 0.05$. However, the small phi-coefficient supported that there is no relation between confirmation rate and channel choosing. It might be the case that users would cancel the order if they find the products are not suitable anymore. This is irrelevant to where the transaction happened.

Table VI:



```
x-squared:  141.432610477
p-value:  1.29403472524e-32
Contingency table:
[[  98065 1076323]
 [   8124  102836]]
phi coef.:  0.0104947638038
```

Before conducting Pearson's Chi-squared tests with Yates' continuity correction on certificate and channel choosing, data are arranged to be dichotomous. We will make another dummy variable to indicate the certification (i.e. 1 = one of the 'Okseller', 'Quickseller' or 'Bigseller' field shown 'Y'; 0 = otherwise). H_0: Certification has no effect on channel choosing, is rejected with p-value 1.294e^-32 <0.05. Phi-coefficient = 0.0105, which is not a significant result. Its positivity implicates that users who chose mobile channel are more likely to consider certification.

Appendix II

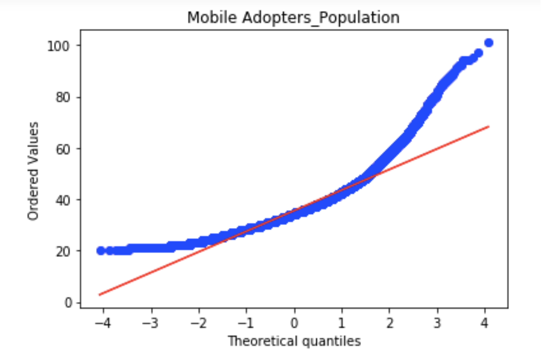
Question 1

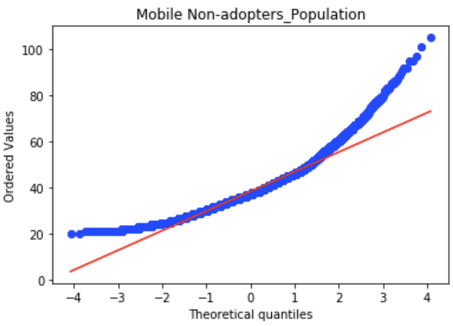
T-test assumptions:

1. Continuous dependent variable

2. Dependent variable is normally distributed

3. Two dependent observation group are independent

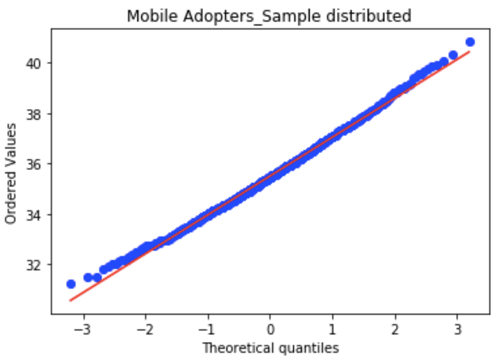


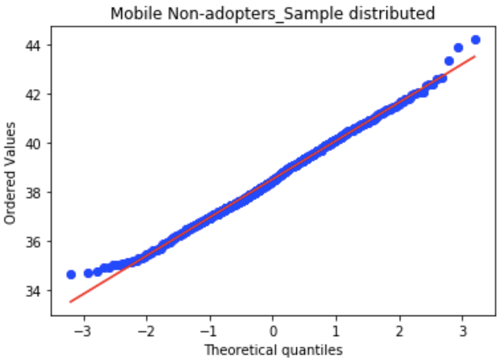


QQ plots above showed that distribution of age in both groups are positively skewed. The second assumption of t-test cannot hold. Thus, we must force it to follow normal distribution by applying Central Limit Theorem. We are going to randomly select certain amount of samples from the population with replacement.

Observations in each sample can be randomly chosen at 30, as long as each sample can reserve enough characteristics of the population. For the sample size, we can simply choose 2000, but we have to consider the effect size to reduce the level of type I error.

QQ plots- Sample

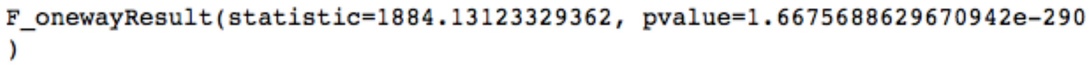




Approximately normally distributed now.

Beside normal distribution, we also have to consider problems of variance equality and paired test.

Since the adopters and non-adopters are independent, paired test is not suitable. After conducting the F-test, we reject the null hypothesis that they have same variance.



```
F_onewayResult(statistic=1884.13123329362, pvalue=1.6675688629670942e-290
)
```

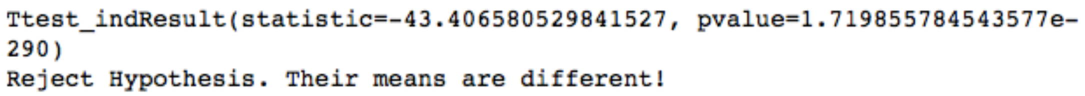
Test: Welch Two sample t-test

Observations in each sample: 30

Sample size: 2000 (1000 in each group)

Paired or unpaired: unpaired

Variance equality: unequal



```
Ttest_indResult(statistic=-43.406580529841527, pvalue=1.7198557845435577e-
290)
Reject Hypothesis. Their means are different!
```

The test statistics rejected the null hypothesis that they have significant difference in mean. The Cohen's d is 1.949537, indicating that their difference is significant.

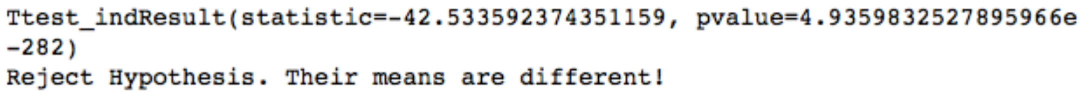
Test: Welch Two sample t-test

Observations in each sample: 30

Sample size: 2000 (1000 in each group)

Paired or unpaired: unpaired

Variance equality: Equal



```
Ttest_indResult(statistic=-42.533592374351159, pvalue=4.9359832527895966e
-282)
Reject Hypothesis. Their means are different!
```

Cohen's d: 1.908593

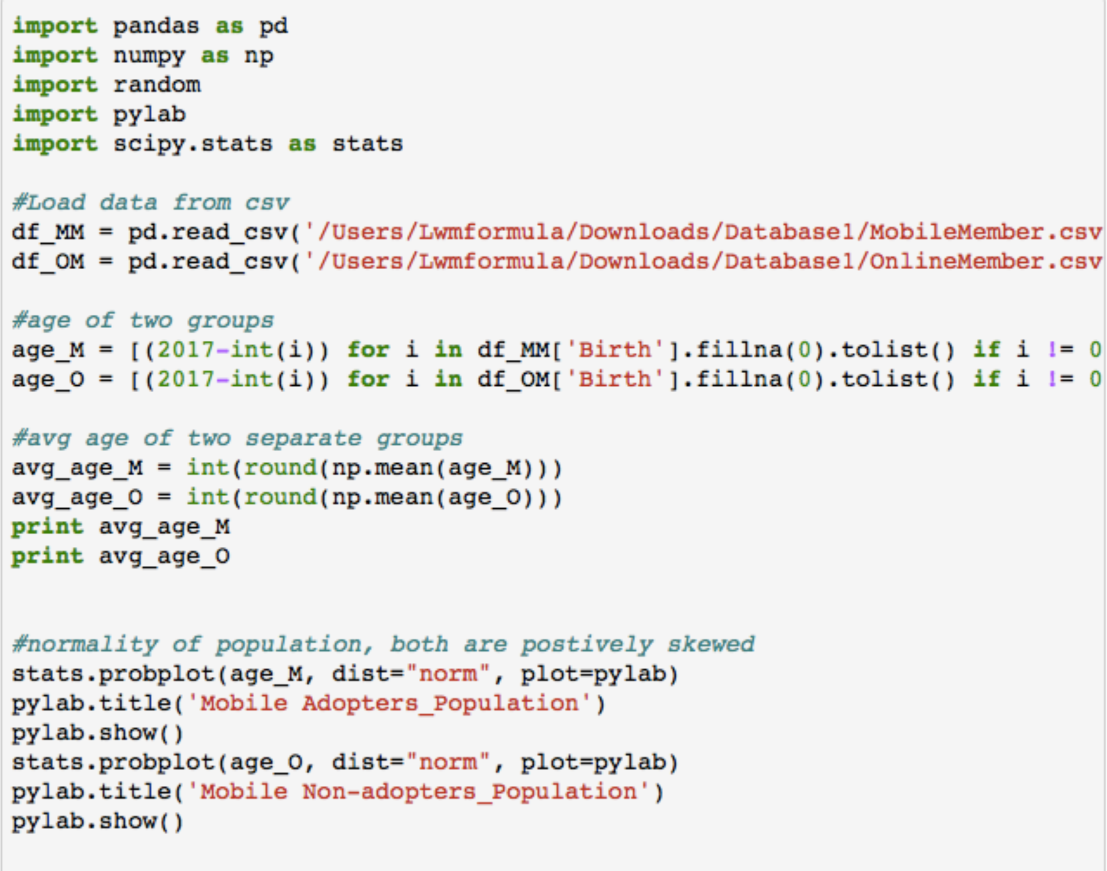
To avoid existing Type I error in F test. Another t test with equal variance assumption is conducted.

The conclusion is consistent.

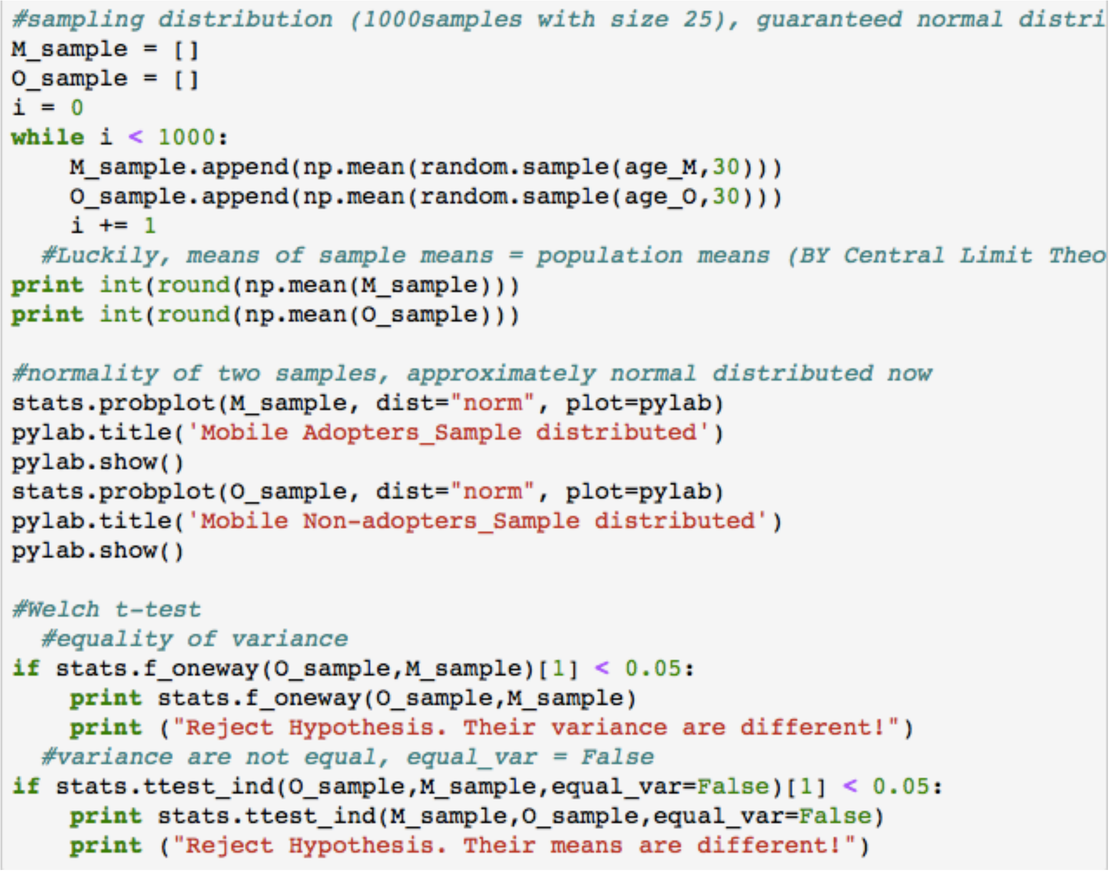
Mean age of adopter and non-adopter groups have significant difference.

Name: NG Hing Yu, Lam Wun Ming
SID: 1155049053, 1155049536

Coding



```
import pandas as pd
import numpy as np
import random
import pylab
import scipy.stats as stats

#Load data from csv
df_MM = pd.read_csv('/Users/Lwmformula/Downloads/Database1/MobileMember.csv
df_OM = pd.read_csv('/Users/Lwmformula/Downloads/Database1/OnlineMember.csv

#age of two groups
age_M = [(2017-int(i)) for i in df_MM['Birth'].fillna(0).tolist() if i != 0
age_O = [(2017-int(i)) for i in df_OM['Birth'].fillna(0).tolist() if i != 0

#avg age of two separate groups
avg_age_M = int(round(np.mean(age_M)))
avg_age_O = int(round(np.mean(age_O)))
print avg_age_M
print avg_age_O


#normality of population, both are postively skewed
stats.probplot(age_M, dist="norm", plot=pylab)
pylab.title('Mobile Adopters_Population')
pylab.show()
stats.probplot(age_O, dist="norm", plot=pylab)
pylab.title('Mobile Non-adopters_Population')
pylab.show()
```



```
#sampling distribution (1000samples with size 25), guaranteed normal distri
M_sample = []
O_sample = []
i = 0
while i < 1000:
    M_sample.append(np.mean(random.sample(age_M,30)))
    O_sample.append(np.mean(random.sample(age_O,30)))
    i += 1
  #Luckily, means of sample means = population means (BY Central Limit Theo
print int(round(np.mean(M_sample)))
print int(round(np.mean(O_sample)))

#normality of two samples, approximately normal distributed now
stats.probplot(M_sample, dist="norm", plot=pylab)
pylab.title('Mobile Adopters_Sample distributed')
pylab.show()
stats.probplot(O_sample, dist="norm", plot=pylab)
pylab.title('Mobile Non-adopters_Sample distributed')
pylab.show()

#Welch t-test
  #equality of variance
if stats.f_oneway(O_sample,M_sample)[1] < 0.05:
    print stats.f_oneway(O_sample,M_sample)
    print ("Reject Hypothesis. Their variance are different!")
  #variance are not equal, equal_var = False
if stats.ttest_ind(O_sample,M_sample,equal_var=False)[1] < 0.05:
    print stats.ttest_ind(M_sample,O_sample,equal_var=False)
    print ("Reject Hypothesis. Their means are different!")
```

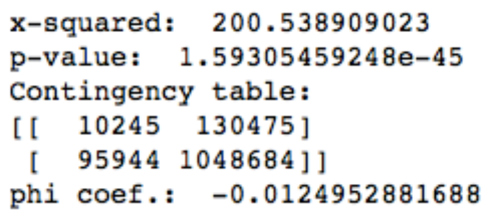
Question 3

By common sense, transactions which contained multiple items usually have larger amount. I made the conjecture that users who bought several things are more likely to choose online channel, because of the cost of searching. Cost of searching is associated with complexity of purchasing multiple items (i.e. purchase multiple items in one transaction is easier). To validate my conjecture, I conduct the Pearson's Chi-squared tests with Yates' continuity correction in mobile transaction and online transaction.

(Mobile transaction = 1, multiple = 1; Online transaction = 0, single = 0)

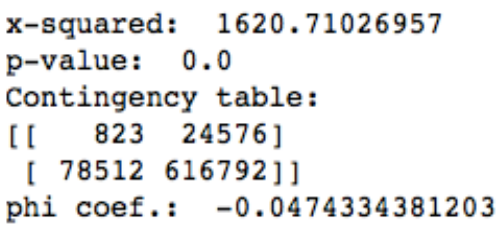
By Pearson's Chi-squared tests with Yates' continuity correction (Phi coefficient)

(Item)



```
x-squared:  200.538909023
p-value:  1.59305459248e-45
Contingency table:
[[  10245  130475]
 [  95944 1048684]]
phi coef.:  -0.0124952881688
```

(Category)



```
x-squared:  1620.71026957
p-value:  0.0
Contingency table:
[[   823  24576]
 [ 78512 616792]]
phi coef.:  -0.0474334381203
```

Indeed, the result in item test is not acceptable. Despite they are negatively correlated (i.e. users who purchased multiple items are less likely to purchase in mobile channel), but the coefficient was small and not significant enough. Therefore, it drives us to think about the important of category. Theoretically, cost of searching should be lower when users purchase multiple categories in one basket or transaction. I conducted another chi-squared test on category. There is a stronger negative relationship between mobile transaction and quantity of purchasing.

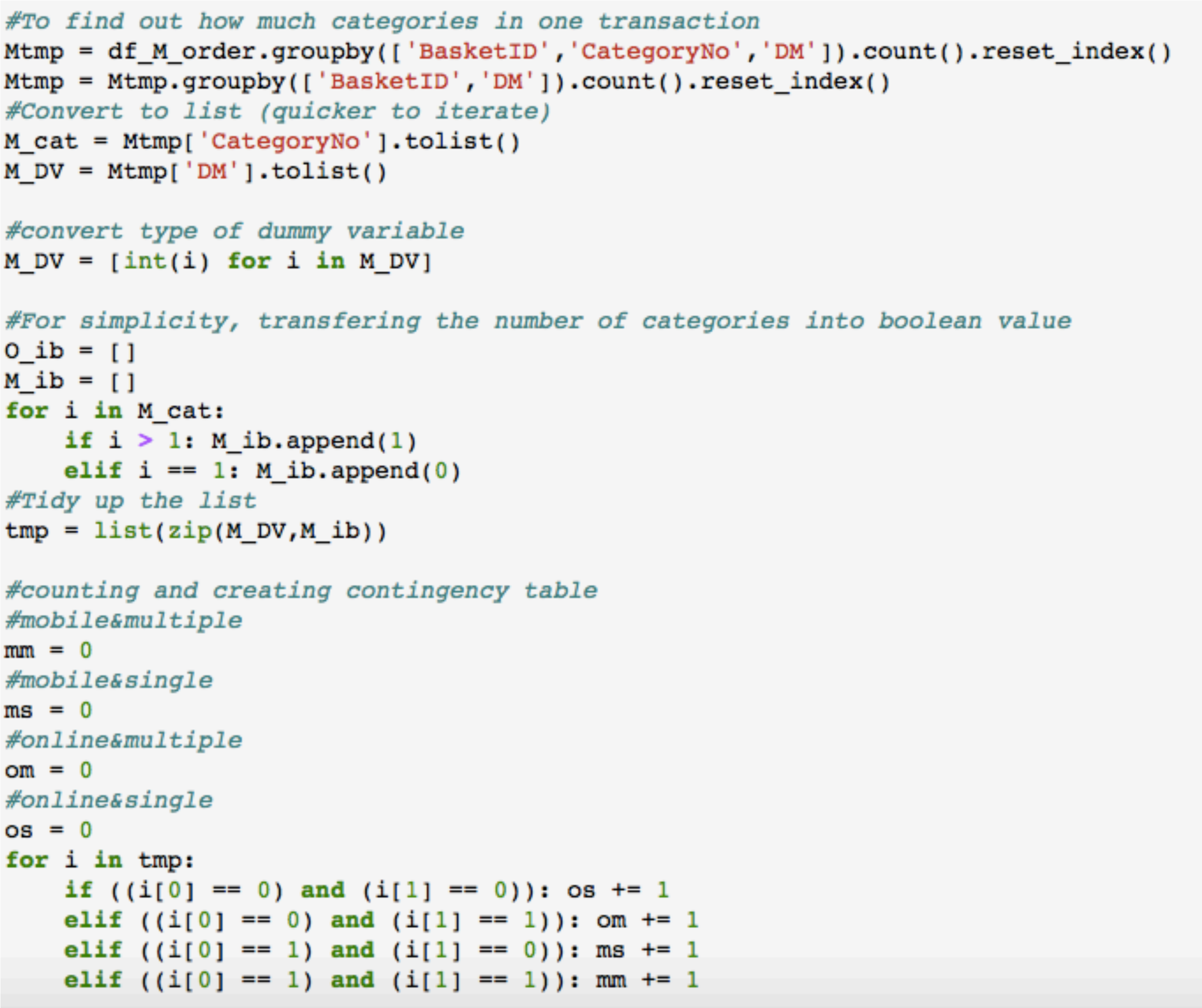
To conclude, quantity of purchasing affect searching cost, and searching cost has little effect on choosing channel. That makes the order price in online channel outperform mobile channel.

Beside, quantity of purchasing is not the only concern. Safety, frequency of browsing and quality of sellers can also be deterministic. For example, users who purchase a single item with large order price would consider more about safety and quality of sellers, which is not explained by the above result.

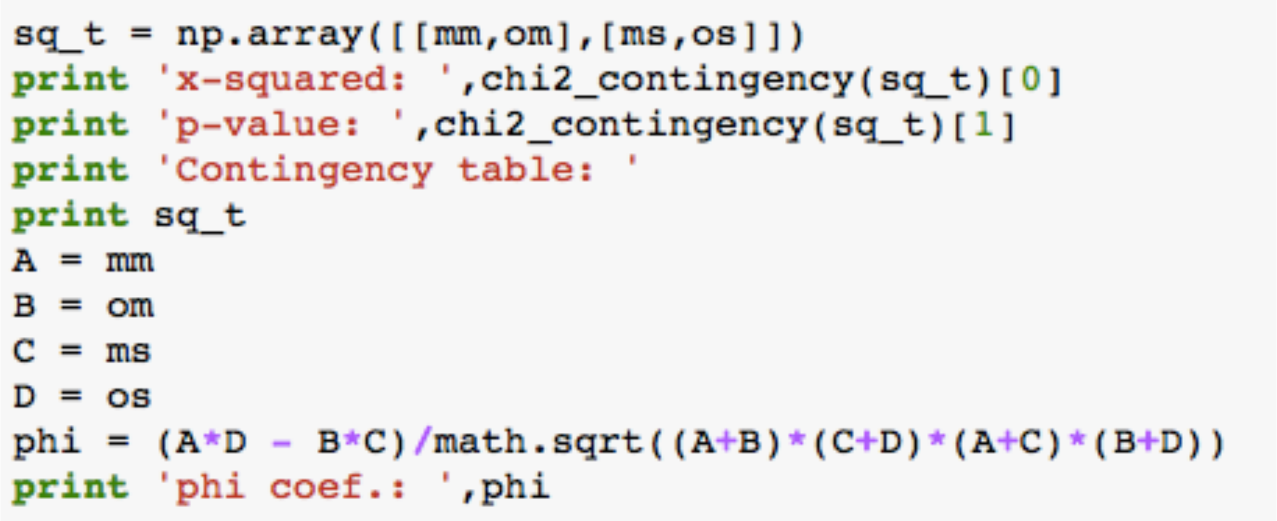
Name: NG Hing Yu, Lam Wun Ming
SID: 1155049053, 1155049536

Coding

(Chi-squared test)- Category

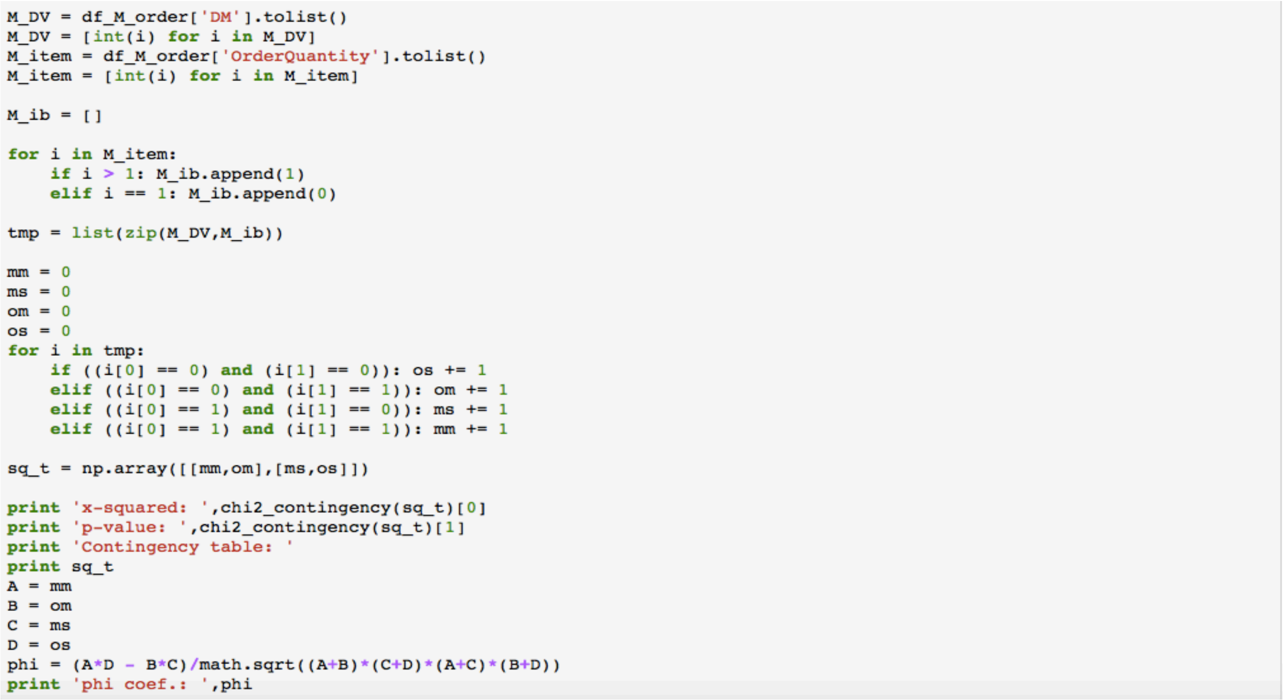


```
#To find out how much categories in one transaction
Mtmp = df_M_order.groupby(['BasketID','CategoryNo','DM']).count().reset_index()
Mtmp = Mtmp.groupby(['BasketID','DM']).count().reset_index()
#Convert to list (quicker to iterate)
M_cat = Mtmp['CategoryNo'].tolist()
M_DV = Mtmp['DM'].tolist()

#convert type of dummy variable
M_DV = [int(i) for i in M_DV]

#For simplicity, transfering the number of categories into boolean value
O_ib = []
M_ib = []
for i in M_cat:
    if i > 1: M_ib.append(1)
    elif i == 1: M_ib.append(0)
#Tidy up the list
tmp = list(zip(M_DV,M_ib))

#counting and creating contingency table
#mobile&multiple
mm = 0
#mobile&single
ms = 0
#online&multiple
om = 0
#online&single
os = 0
for i in tmp:
    if ((i[0] == 0) and (i[1] == 0)): os += 1
    elif ((i[0] == 0) and (i[1] == 1)): om += 1
    elif ((i[0] == 1) and (i[1] == 0)): ms += 1
    elif ((i[0] == 1) and (i[1] == 1)): mm += 1
```



```
sq_t = np.array([[mm,om],[ms,os]])
print 'x-squared: ',chi2_contingency(sq_t)[0]
print 'p-value: ',chi2_contingency(sq_t)[1]
print 'Contingency table: '
print sq_t
A = mm
B = om
C = ms
D = os
phi = (A*D - B*C)/math.sqrt((A+B)*(C+D)*(A+C)*(B+D))
print 'phi coef.: ',phi
```

(Chi-squared test)- Item



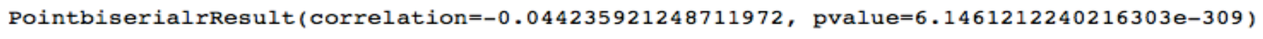
```
M_DV = df_M_order['DM'].tolist()
M_DV = [int(i) for i in M_DV]
M_item = df_M_order['OrderQuantity'].tolist()
M_item = [int(i) for i in M_item]

M_ib = []

for i in M_item:
    if i > 1: M_ib.append(1)
    elif i == 1: M_ib.append(0)

tmp = list(zip(M_DV,M_ib))

mm = 0
ms = 0
om = 0
os = 0
for i in tmp:
    if ((i[0] == 0) and (i[1] == 0)): os += 1
    elif ((i[0] == 0) and (i[1] == 1)): om += 1
    elif ((i[0] == 1) and (i[1] == 0)): ms += 1
    elif ((i[0] == 1) and (i[1] == 1)): mm += 1

sq_t = np.array([[mm,om],[ms,os]])

print 'x-squared: ',chi2_contingency(sq_t)[0]
print 'p-value: ',chi2_contingency(sq_t)[1]
print 'Contingency table: '
print sq_t
A = mm
B = om
C = ms
D = os
phi = (A*D - B*C)/math.sqrt((A+B)*(C+D)*(A+C)*(B+D))
print 'phi coef.: ',phi
```

Someone may argue that creating another dummy variable for purchasing quantity (i.e. 1,2,3,4,5,6,… ➔ 0 or 1) may distort the original data. The result becomes not reliable. Thus, we are trying to solve this problem by using Point Biserial R test, which is designed for finding the relation coefficient between dichotomous and continuous data.

Independent variable: Mobile or Online (dichotomous)

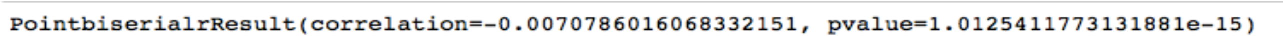
Dependent variable: Number of categories or items (continuous)

(Category)



```
PointbiserialrResult(correlation=-0.044235921248711972, pvalue=6.1461212240216303e-309)
```

(Item)

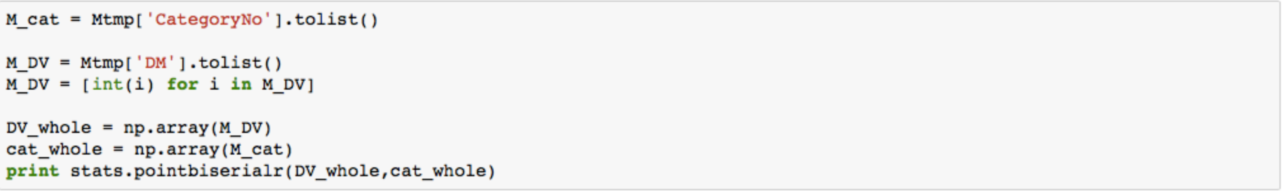


```
PointbiserialrResult(correlation=-0.0070786016068332151, pvalue=1.0125411773131881e-15)
```

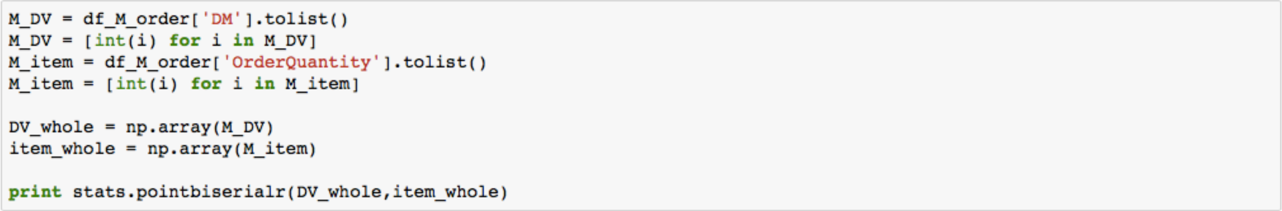
The test results are consistent with Pearson's Chi-squared tests with Yates' continuity correction.

Users prefer purchasing multiple categories of products via online transaction.

Coding



```
M_cat = Mtmp['CategoryNo'].tolist()

M_DV = Mtmp['DM'].tolist()
M_DV = [int(i) for i in M_DV]

DV_whole = np.array(M_DV)
cat_whole = np.array(M_cat)
print stats.pointbiserialr(DV_whole,cat_whole)
```



```
M_DV = df_M_order['DM'].tolist()
M_DV = [int(i) for i in M_DV]
M_item = df_M_order['OrderQuantity'].tolist()
M_item = [int(i) for i in M_item]

DV_whole = np.array(M_DV)
item_whole = np.array(M_item)

print stats.pointbiserialr(DV_whole,item_whole)
```

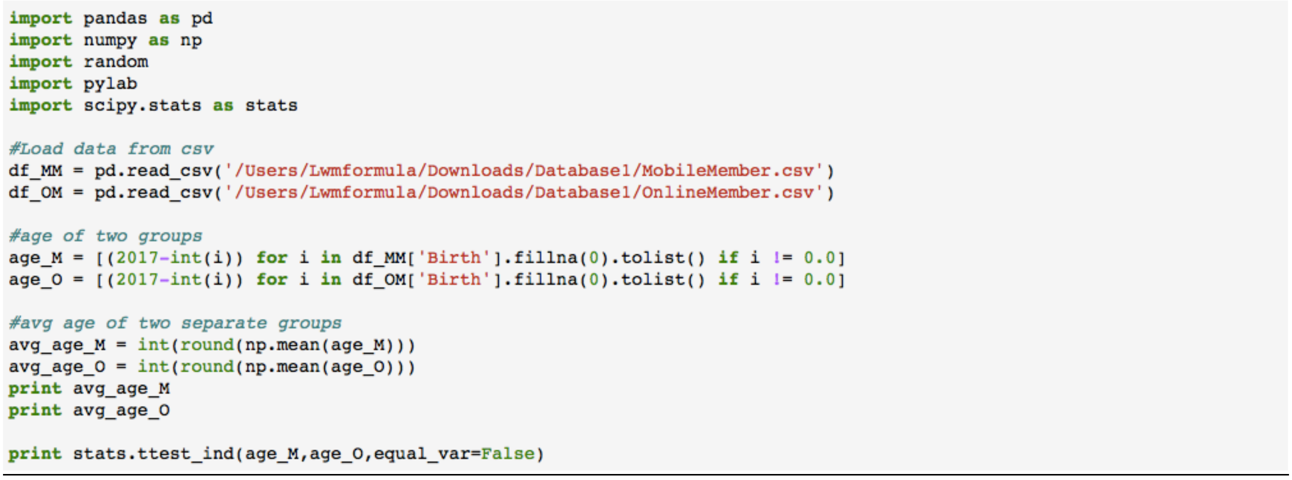
Question 4

The possible reason of higher confirmation rate in online channel is that users are risk averse. By comparing with online channel, m-commerce is relatively new. Users are not confident in the system of mobile channel. If they did not receive any confirmation announcement as usual after purchasing via mobile channel, they might choose to cancel out the purchasing in order to avoid loss or troublesome caused by technical error. To valid my conjecture, data of users' risk attitude are required, such as rate of choosing confirmation message.

Name: NG Hing Yu, Lam Wun Ming
SID: 1155049053, 1155049536

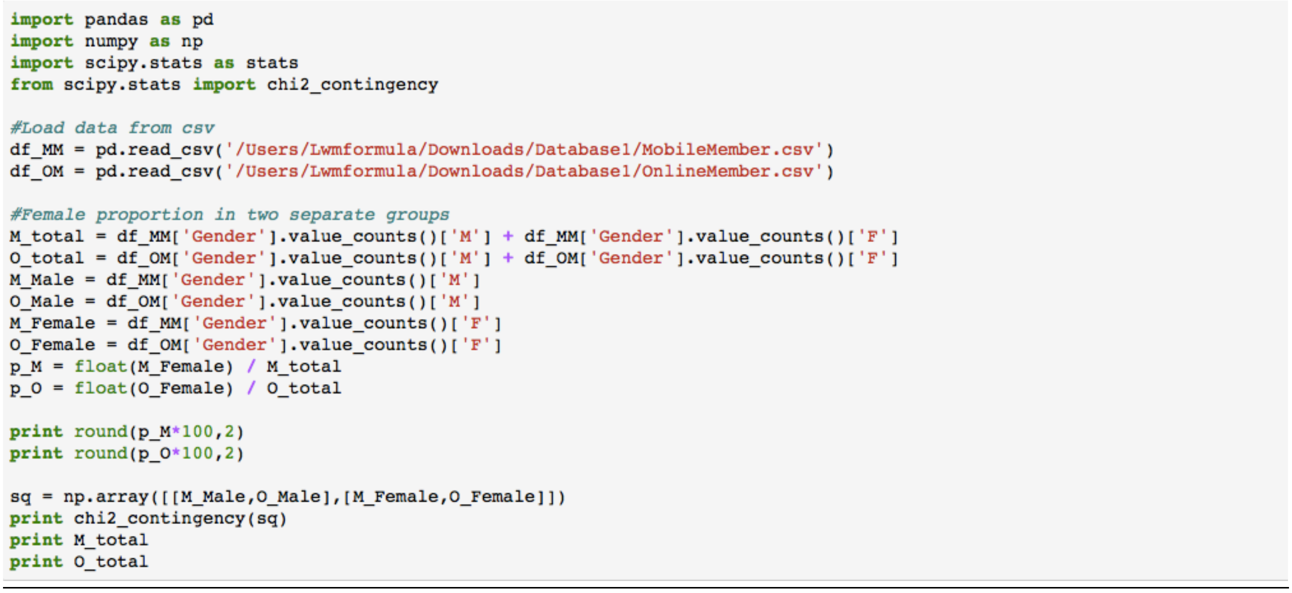
Appendix III

Question 1



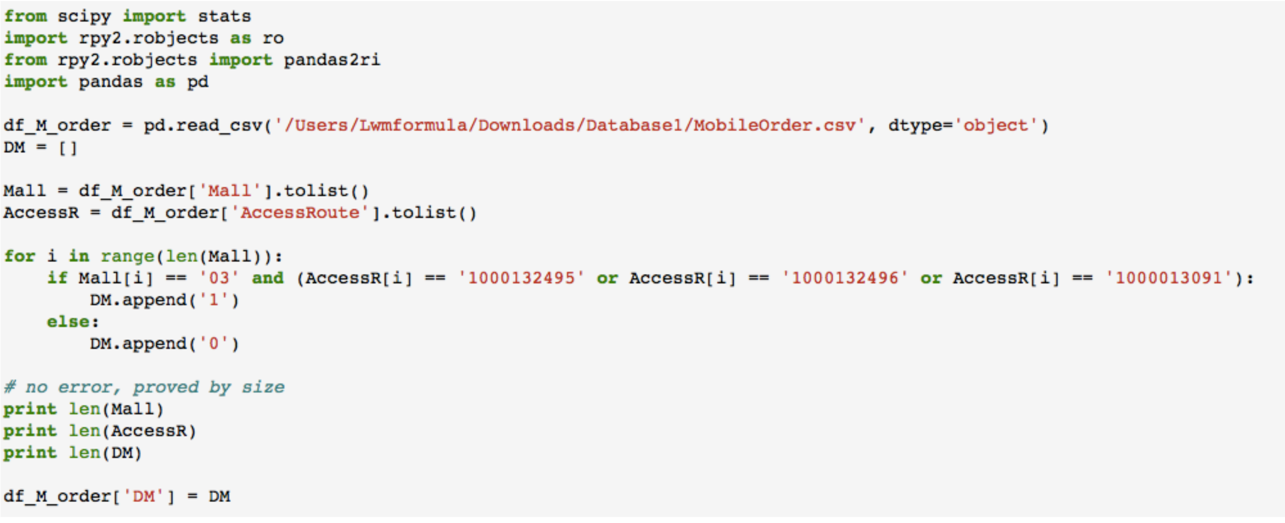
```
import pandas as pd
import numpy as np
import random
import pylab
import scipy.stats as stats

#Load data from csv
df_MM = pd.read_csv('/Users/Lwmformula/Downloads/Database1/MobileMember.csv')
df_OM = pd.read_csv('/Users/Lwmformula/Downloads/Database1/OnlineMember.csv')

#age of two groups
age_M = [(2017-int(i)) for i in df_MM['Birth'].fillna(0).tolist() if i != 0.0]
age_O = [(2017-int(i)) for i in df_OM['Birth'].fillna(0).tolist() if i != 0.0]

#avg age of two separate groups
avg_age_M = int(round(np.mean(age_M)))
avg_age_O = int(round(np.mean(age_O)))
print avg_age_M
print avg_age_O

print stats.ttest_ind(age_M,age_O,equal_var=False)
```

Question 2

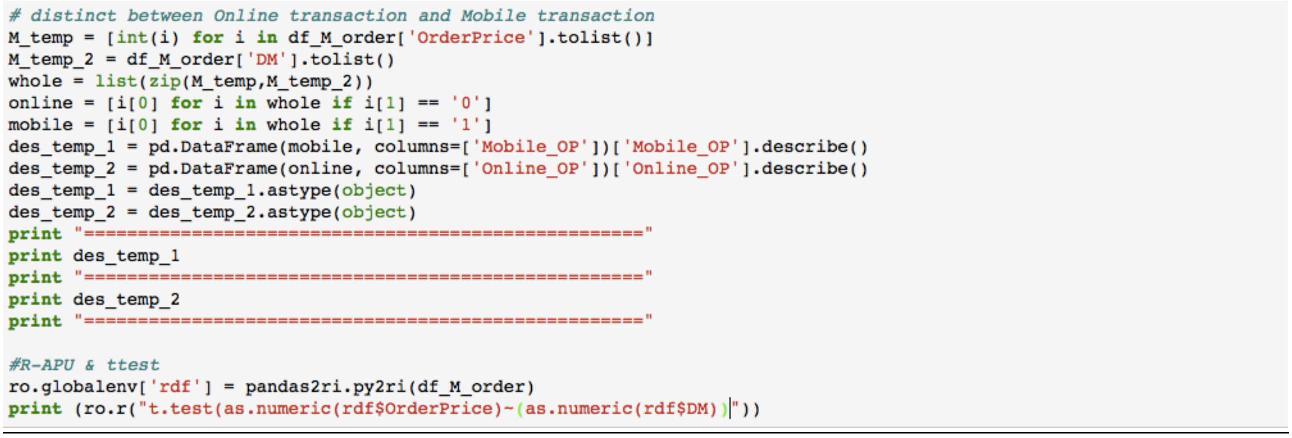


```
import pandas as pd
import numpy as np
import scipy.stats as stats
from scipy.stats import chi2_contingency

#Load data from csv
df_MM = pd.read_csv('/Users/Lwmformula/Downloads/Database1/MobileMember.csv')
df_OM = pd.read_csv('/Users/Lwmformula/Downloads/Database1/OnlineMember.csv')

#Female proportion in two separate groups
M_total = df_MM['Gender'].value_counts()['M'] + df_MM['Gender'].value_counts()['F']
O_total = df_OM['Gender'].value_counts()['M'] + df_OM['Gender'].value_counts()['F']
M_Male = df_MM['Gender'].value_counts()['M']
O_Male = df_OM['Gender'].value_counts()['M']
M_Female = df_MM['Gender'].value_counts()['F']
O_Female = df_OM['Gender'].value_counts()['F']
p_M = float(M_Female) / M_total
p_O = float(O_Female) / O_total

print round(p_M*100,2)
print round(p_O*100,2)

sq = np.array([[M_Male,O_Male],[M_Female,O_Female]])
print chi2_contingency(sq)
print M_total
print O_total
```

Question 3

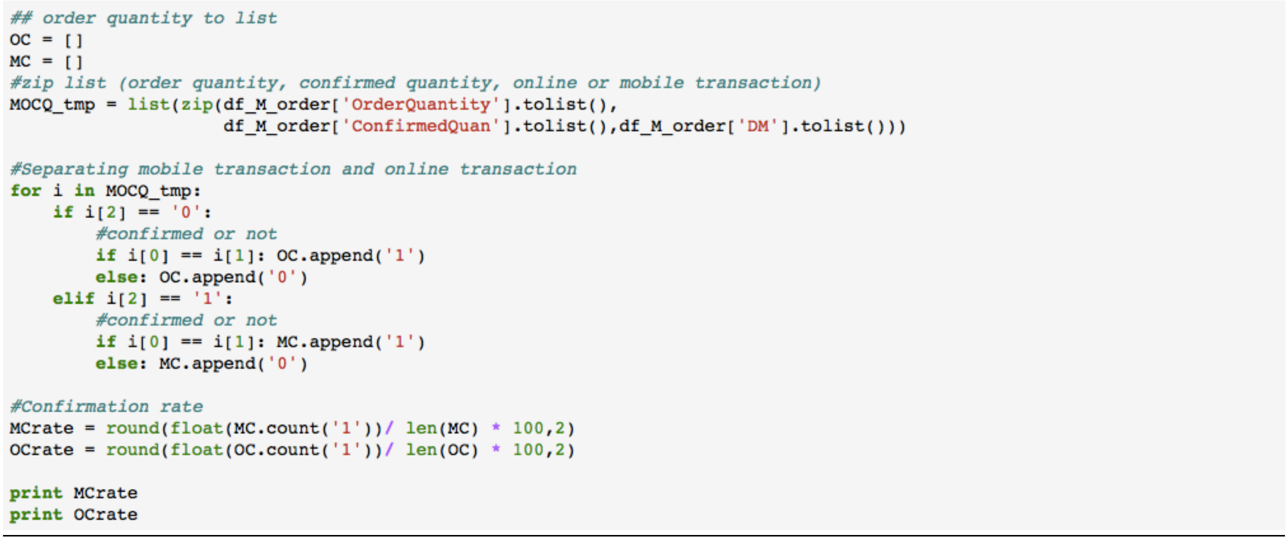


```
from scipy import stats
import rpy2.robjects as ro
from rpy2.robjects import pandas2ri
import pandas as pd

df_M_order = pd.read_csv('/Users/Lwmformula/Downloads/Database1/MobileOrder.csv', dtype='object')
DM = []

Mall = df_M_order['Mall'].tolist()
AccessR = df_M_order['AccessRoute'].tolist()

for i in range(len(Mall)):
    if Mall[i] == '03' and (AccessR[i] == '1000132495' or AccessR[i] == '1000132496' or AccessR[i] == '1000013091'):
        DM.append('1')
    else:
        DM.append('0')

# no error, proved by size
print len(Mall)
print len(AccessR)
print len(DM)

df_M_order['DM'] = DM
```



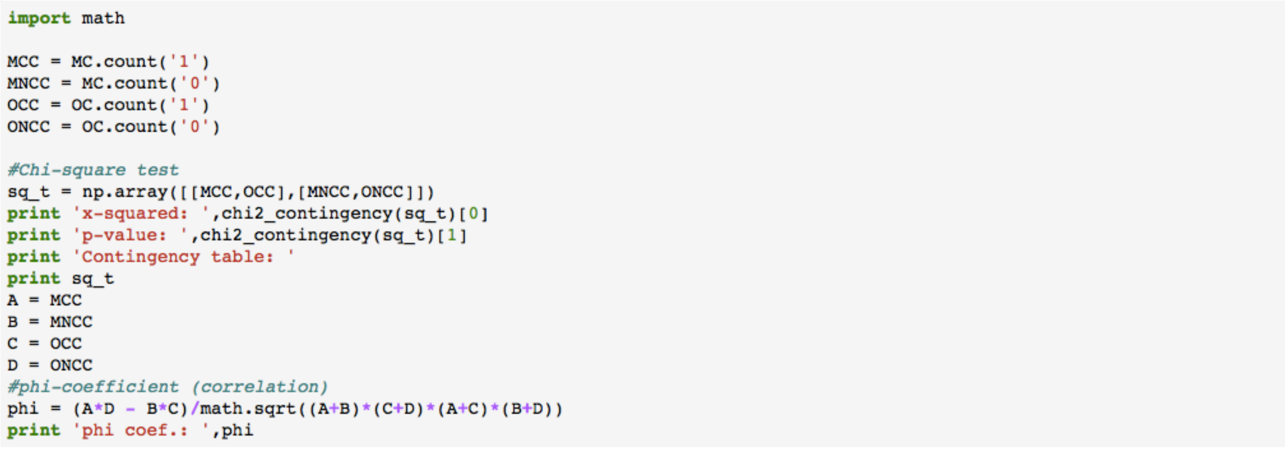
```
# distinct between Online transaction and Mobile transaction
M_temp = [int(i) for i in df_M_order['OrderPrice'].tolist()]
M_temp_2 = df_M_order['DM'].tolist()
whole = list(zip(M_temp,M_temp_2))
online = [i[0] for i in whole if i[1] == '0']
mobile = [i[0] for i in whole if i[1] == '1']
des_temp_1 = pd.DataFrame(mobile, columns=['Mobile_OP'])['Mobile_OP'].describe()
des_temp_2 = pd.DataFrame(online, columns=['Online_OP'])['Online_OP'].describe()
des_temp_1 = des_temp_1.astype(object)
des_temp_2 = des_temp_2.astype(object)
print "===================================================="
print des_temp_1
print "===================================================="
print des_temp_2
print "===================================================="

#R-APU & ttest
ro.globalenv['rdf'] = pandas2ri.py2ri(df_M_order)
print (ro.r("t.test(as.numeric(rdf$OrderPrice)~(as.numeric(rdf$DM))"))
```

Question 4



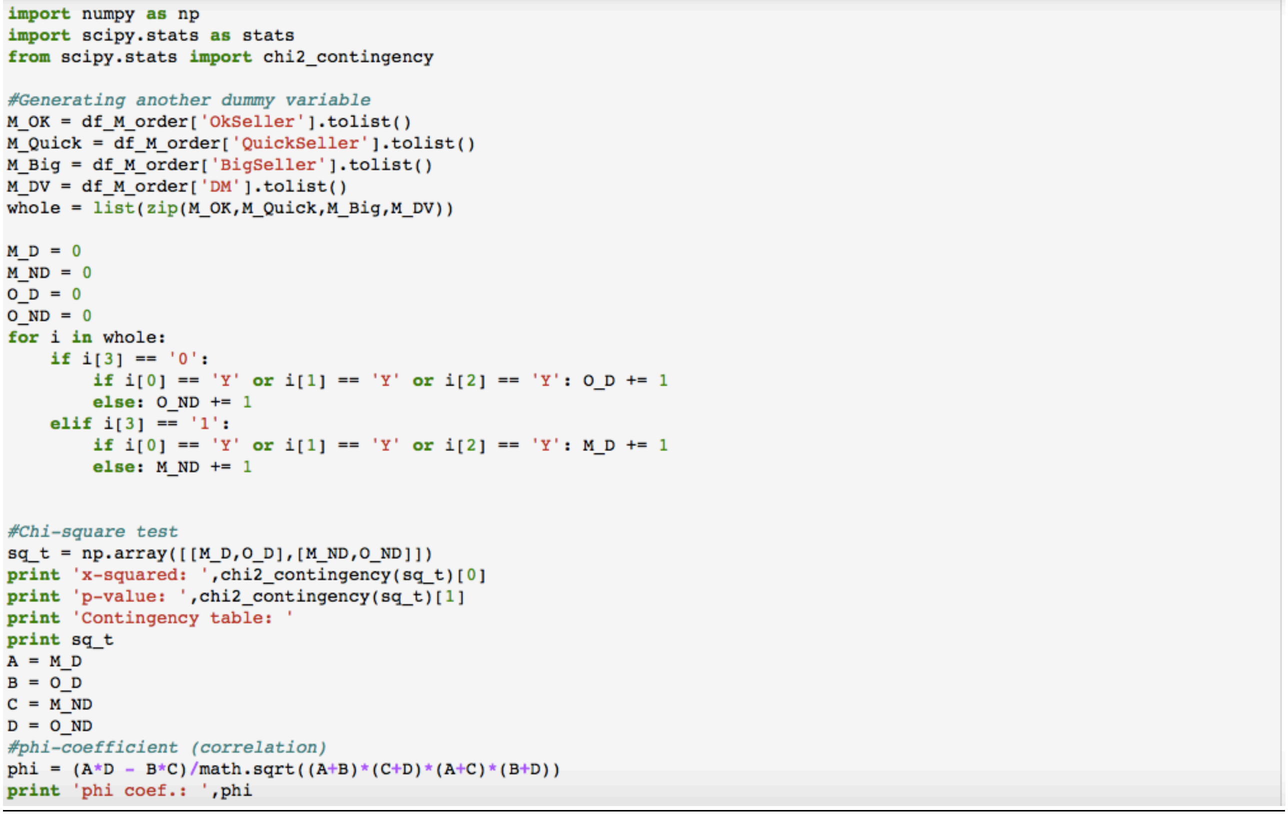
```
## order quantity to list
OC = []
MC = []
#zip list (order quantity, confirmed quantity, online or mobile transaction)
MOCQ_tmp = list(zip(df_M_order['OrderQuantity'].tolist(),
                    df_M_order['ConfirmedQuan'].tolist(),df_M_order['DM'].tolist()))

#Separating mobile transaction and online transaction
for i in MOCQ_tmp:
    if i[2] == '0':
        #confirmed or not
        if i[0] == i[1]: OC.append('1')
        else: OC.append('0')
    elif i[2] == '1':
        #confirmed or not
        if i[0] == i[1]: MC.append('1')
        else: MC.append('0')

#Confirmation rate
MCrate = round(float(MC.count('1'))/ len(MC) * 100,2)
OCrate = round(float(OC.count('1'))/ len(OC) * 100,2)

print MCrate
print OCrate
```

Name: NG Hing Yu, Lam Wun Ming
SID: 1155049053, 1155049536



```
import math

MCC = MC.count('1')
MNCC = MC.count('0')
OCC = OC.count('1')
ONCC = OC.count('0')

#Chi-square test
sq_t = np.array([[MCC,OCC],[MNCC,ONCC]])
print 'x-squared: ',chi2_contingency(sq_t)[0]
print 'p-value: ',chi2_contingency(sq_t)[1]
print 'Contingency table: '
print sq_t
A = MCC
B = MNCC
C = OCC
D = ONCC
#phi-coefficient (correlation)
phi = (A*D - B*C)/math.sqrt((A+B)*(C+D)*(A+C)*(B+D))
print 'phi coef.: ',phi
```

Question 5



```
import numpy as np
import scipy.stats as stats
from scipy.stats import chi2_contingency

#Generating another dummy variable
M_OK = df_M_order['OkSeller'].tolist()
M_Quick = df_M_order['QuickSeller'].tolist()
M_Big = df_M_order['BigSeller'].tolist()
M_DV = df_M_order['DM'].tolist()
whole = list(zip(M_OK,M_Quick,M_Big,M_DV))

M_D = 0
M_ND = 0
O_D = 0
O_ND = 0
for i in whole:
    if i[3] == '0':
        if i[0] == 'Y' or i[1] == 'Y' or i[2] == 'Y': O_D += 1
        else: O_ND += 1
    elif i[3] == '1':
        if i[0] == 'Y' or i[1] == 'Y' or i[2] == 'Y': M_D += 1
        else: M_ND += 1


#Chi-square test
sq_t = np.array([[M_D,O_D],[M_ND,O_ND]])
print 'x-squared: ',chi2_contingency(sq_t)[0]
print 'p-value: ',chi2_contingency(sq_t)[1]
print 'Contingency table: '
print sq_t
A = M_D
B = O_D
C = M_ND
D = O_ND
#phi-coefficient (correlation)
phi = (A*D - B*C)/math.sqrt((A+B)*(C+D)*(A+C)*(B+D))
print 'phi coef.: ',phi
```