Name: Chan Ka Tsun
SID: 53563363

1. SEN & R (correlation coef.): -0.60 | RELV & R (correlation coef.): 0.50

   Higher sensitivity score ➔ Less incentive to provide correcting info (Strong)

   Higher relevant score ➔ More incentive to provide correcting info (Medium)

   Mean age of male group is larger than female group.

   Methodology and detailed finding are attached in Appendix I.

2. Positive significant factors (Pooling model): Intercept, itr, age, relv

   Negative significant factors (Pooling model): sex, sen

   The pooling model ignore the heterogeneity of users. Thus, first difference, fixed effect and random effects model should be applied.

   All of the models pointed out that time-variant data relv and sen have major influence on the incentive of providing correcting information. Age is not a significant factor and sex has less power in random effect model. This can be explained by user heterogeneity is under certain control in random effect model. However, when we conduct Panel Generalized Linear Model with random effect, age become significant again.

   Although it is failing to reject null hypothesis in Hausman test, we still choose fixed effect model. Random effect model cannot perfectly control for time-invariant user heterogeneity. More details are explained in Appendix I.

3. In the fixed effect model, each one-unit increase in sen on average, in a 0.0056 decrease in the providing correcting information scale. Also, each one-unit increase in relv on average, in a 0.004 increase in the providing correcting information scale.

   In the random model, male and users who have higher internet trust are more likely to provide correct information. Each one-unit increase in internet trust, on average, in a 0.0496 increase in the providing scale. Also, higher relevance of the item would increase the incentive to provide the correcting information, and user are less likely to provide the correct information if the sensitivity of the item is low. F test to see whether all the coefficients in the model are different than zero is passed with p value < 2.22e-16, the model is valid. (See appendix I)

   As the data type of dependent variable is dichotomous, logistic regression should be applied (See Appendix I). In the Panel Generalized Linear Model, age also have significant effect on providing correct information. Each one-unit increase in internet trust, on average, in a 0.066 increase in the providing scale.

Name: Chan Ka Tsun
SID: 53563363

4. In random effect and fixed effect model, interaction term of sensitivity and relevance is not significant. No moderation exists. On the other hand, interaction term is quite significant in the logistic regression model combined with random effect. Negative coefficient of interaction term indicate that the effect of sensitivity is weaker in magnitude for high relevance of an item, i.e. users are more likely to provide correct information if the item is relevant, even it is sensitive.

5. As we only have mean value of sen and relv, it is not sufficient to establish the relationship. Imputing missing value with mean cause reducing in variability and weakening covariance and correlation estimates in the data, due to we ignore relationship between variables. Thus, we need each user's score in sen and relv. In addition, the above model can only control for time invariant variable. The bias caused by time variant variable are not being well controlled by sen and relv. More time variant data such as "whether the item will be public or not", should be added into the model.

Name: Chan Ka Tsun
SID: 53563363

Appendix I

Question 1

Methodology

The merged table contains 203 rows and 96 columns. Each row represents one user. There are 6 attributes for basic information (ID, know, expr, itr, ip, sex), 30 columns for dummy variables indicating the correct info, 30 columns for average sensitive score of each item, and 30 columns for average relevant score of each item. For now, we are able to generate several summarize tables by “groupby” function, in order to find the relationships between D & Sen and D & Relv.

We have also merged the table in another style. The new table contains 6090 rows and 11 columns. Each user has 30 observations (rows) in the table. There are 7 attributes to indicate the user’s characteristics, 1 column for dummy variable, 2 columns for sensitivity score and relevance score, and 1 column to indicate the item. Item and ID attributes will be used together as a primary key to signify each observation. The table are in panel structure now.

Sample

```
      S1              R1
   count  mean  std count  mean  std
1
0      5  33.2  0.0     5  82.5  0.0
1    198  33.2  0.0   198  82.5  0.0
=========================================================================
      S2              R2
   count  mean  std count  mean  std
2
0     20  46.8  0.0    20  75.0  0.0
1    183  46.8  0.0   183  75.0  0.0
=========================================================================
```

In addition, 30 more variables will be created to denote the correcting rate of each item. For now, we have three relevant groups to run statistical test, i.e. correcting rate, average sensitivity score and average relevant score of each item. As both dependent and independent variables are continuous, normal correlation test is preferred.

Findings

After all, we investigated that sensitivity score and relevant score are negatively correlated and positivity correlated to the correcting rate respectively. Both of the magnitude of their relationship are strong. It accounts for higher sensitivity of the item would reduce the incentive to provide the correcting information, and user are more likely to provide the correct information if the relevance of the item is high. Another interesting thing is that H_0: Age between male and female group are no difference is rejected by with p-value 7.128e-08 < 0.05. Perhaps men in this country do not have much chance to get date.

```
        Welch Two Sample t-test

data:  as.numeric(rdf$age) by as.numeric(rdf$sex)
t = 30.889, df = 5665.2, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.637793 1.859767
sample estimates:
mean in group 1 mean in group 2 
       23.17431        21.42553 
```

Question 2 & 3

Methodology

As we only have average data in sensitivity and relevance, each user's attitude toward two aspects would be the same. We adopted table which merged in question I and with panel structure.

Findings

In the pooling model, we found that there's still a lot of room for improvement. Unobserved heterogeneity of users is not controlled under pooling model. Thus, we apply first difference, fixed effect and random effect models. Firstly, we would eliminate the first difference result as we would lose more degree of freedom. The second reason to abolish first difference model is to avoid the issue of heteroscedasticity.

In order to make a choice between fixed effect model and random effect model, we would conduct Hausman test. The null hypothesis that the individual effects are uncorrelated with the repressors cannot be rejected, with p-value 1. Theoretically, we should choose random effect model. However, random effect model cannot perfectly control for observed users' heterogeneity and we do want to focus on the main effect of sen and relv. As a result, we still choose fixed effect model.

By comparing with the pooling model we conducted initially, the result is slightly different. Age do not have significant influence on the incentive of providing correcting information.

Name: Chan Ka Tsun
SID: 53563363

Addition

The data structure of the dummy variable is binary type; logistic regression should be applied. Thus we may use Panel Generalized Linear Model for pooling and random effect. Since strong assumption of fixed effect and first difference model cannot be hold as discussed in J. Wooldridge's "Econometric analysis of cross-section and panel data", chapter 15, we simply apply "plm" function for these two models.

Both of the result of those model with binomial distribution are consistent, except variable age. Age is more significant in Panel Generalized Linear Model with random effect.

Fixed effect model should be chosen when we conduct the phTest between Panel Generalized Linear Model with random effect and fixed effect model.

Pooling Model

```
Pooling Model

Call:
plm(formula = DV ~ know + expr + itr + ip + sex + age + sen +
    relv, data = df, model = "pooling", index = c("ID", "item"))

Balanced Panel: n=203, T=30, N=6090

Residuals :
   Min. 1st Qu.  Median 3rd Qu.    Max.
 -0.979  -0.470   0.188   0.340   0.855

Coefficients :
               Estimate  Std. Error  t-value  Pr(>|t|)
(Intercept)  0.42100197  0.08245457   5.1059 3.393e-07 ***
know         0.02462112  0.01974781   1.2468    0.2125
expr        -0.01749713  0.02010071  -0.8705    0.3841
itr          0.04968868  0.00664096   7.4822 8.348e-14 ***
ip          -0.00689283  0.00554967  -1.2420    0.2143
sex         -0.06338253  0.01256647  -5.0438 4.697e-07 ***
age          0.01078111  0.00264667   4.0735 4.691e-05 ***
sen         -0.00563553  0.00031668 -17.7958 < 2.2e-16 ***
relv         0.00407129  0.00035108  11.5963 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Total Sum of Squares:    1393.9
Residual Sum of Squares: 1219
R-Squared:       0.1255
Adj. R-Squared: 0.12435
F-statistic: 109.087 on 8 and 6081 DF, p-value: < 2.22e-16
```

Name: Chan Ka Tsun
SID: 53563363

First Difference Model

```
> summary(df.fd)
Oneway (individual) effect First-Difference Model

Call:
plm(formula = DV ~ know + expr + itr + ip + sex + age + sen +
    relv, data = df, model = "fd", index = c("ID", "item"))

Balanced Panel: n=203, T=30, N=6090
Observations used in estimation: 5887

Residuals :
    Min.  1st Qu.   Median  3rd Qu.     Max.
-1.34000 -0.18200 -0.00411  0.11900  1.48000

Coefficients :
               Estimate  Std. Error  t-value Pr(>|t|)
(intercept) -0.00691357  0.00748715  -0.9234   0.3558
sen         -0.00497088  0.00037280 -13.3341   <2e-16 ***
relv         0.00481612  0.00039863  12.0818   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Total Sum of Squares:    2102.9
Residual Sum of Squares: 1938.7
R-Squared:       0.078103
Adj. R-Squared: 0.07779
F-statistic: 249.246 on 2 and 5884 DF, p-value: < 2.22e-16
```

Fixed Effect Model

```
> summary(df.fe)
Oneway (individual) effect Within Model

Call:
plm(formula = DV ~ know + expr + itr + ip + sex + age + sen +
    relv, data = df, model = "within", index = c("ID", "item"))

Balanced Panel: n=203, T=30, N=6090

Residuals :
   Min. 1st Qu.  Median 3rd Qu.    Max.
 -1.120  -0.373   0.105   0.317   1.080

Coefficients :
        Estimate  Std. Error t-value  Pr(>|t|)
sen  -0.00563553  0.00029832 -18.891 < 2.2e-16 ***
relv  0.00407129  0.00033073  12.310 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Total Sum of Squares:    1194.5
Residual Sum of Squares: 1046.9
R-Squared:       0.12357
Adj. R-Squared: 0.093187
F-statistic: 414.862 on 2 and 5885 DF, p-value: < 2.22e-16
```

Name: Chan Ka Tsun
SID: 53563363

Random Effect Model

```
Oneway (individual) effect Random Effect Model
   (Swamy-Arora's transformation)

Call:
plm(formula = DV ~ know + expr + itr + ip + sex + age + sen +
    relv, data = df, model = "random", index = c("ID", "item"))

Balanced Panel: n=203, T=30, N=6090

Effects:
                  var std.dev share
idiosyncratic 0.17807 0.42198 0.884
individual    0.02339 0.15294 0.116
theta:  0.5501

Residuals :
   Min. 1st Qu.  Median 3rd Qu.    Max.
 -0.982  -0.430   0.157   0.323   0.974

Coefficients :
               Estimate  Std. Error  t-value  Pr(>|t|)
(Intercept)  0.42100197  0.16292406   2.5840 0.0097882 **
know         0.02462112  0.04135038   0.5954 0.5515805
expr        -0.01749713  0.04208931  -0.4157 0.6776337
itr          0.04968868  0.01390565   3.5733 0.0003553 ***
ip          -0.00689283  0.01162057  -0.5932 0.5530981
sex         -0.06338253  0.02631320  -2.4088 0.0160358 *
age          0.01078111  0.00554193   1.9454 0.0517764 .
sen         -0.00563553  0.00029832 -18.8911 < 2.2e-16 ***
relv         0.00407129  0.00033073  12.3101 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Total Sum of Squares:    1234.8
Residual Sum of Squares: 1081.7
R-Squared:        0.12401
Adj. R-Squared: 0.12286
F-statistic: 107.607 on 8 and 6081 DF, p-value: < 2.22e-16
```

phTest

```
> phtest(df.fe,df.re)

          Hausman Test

data:  DV ~ know + expr + itr + ip + sex + age + sen + relv
chisq = 1.473e-21, df = 2, p-value = 1
alternative hypothesis: one model is inconsistent
```

Name: Chan Ka Tsun
SID: 53563363

Generalized Panel Linear Model (Pooling)

```
> summary(df.gpl)
--------------------------------------------
Maximum Likelihood estimation
Newton-Raphson maximisation, 4 iterations
Return code 1: gradient close to zero
Log-Likelihood: -3565.739
9  free parameters
Estimates:
             Estimate Std. error t value  Pr(> t)    
(Intercept) -0.461035   0.413328  -1.115    0.265    
know         0.123008   0.098089   1.254    0.210    
expr        -0.088929   0.099835  -0.891    0.373    
itr          0.248617   0.033435   7.436 1.04e-13 ***
ip          -0.036387   0.027852  -1.306    0.191    
sex         -0.312616   0.062865  -4.973 6.60e-07 ***
age          0.054325   0.013382   4.059 4.92e-05 ***
sen         -0.026556   0.001589 -16.718  < 2e-16 ***
relv         0.018933   0.001744  10.858  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
--------------------------------------------
```

Generalized Panel Linear Model (Random Effect)

```
> summary(df.gre)
--------------------------------------------
Maximum Likelihood estimation
Newton-Raphson maximisation, 3 iterations
Return code 2: successive function values within tolerance limit
Log-Likelihood: -3360.986
10  free parameters
Estimates:
             Estimate Std. error t value  Pr(> t)    
(Intercept) -0.624332   0.932266  -0.670 0.503054    
know         0.182522   0.234406   0.779 0.436182    
expr        -0.119753   0.238078  -0.503 0.614967    
itr          0.305619   0.079581   3.840 0.000123 ***
ip          -0.050565   0.065869  -0.768 0.442694    
sex         -0.384469   0.150408  -2.556 0.010583 *  
age          0.065591   0.031804   2.062 0.039176 *  
sen         -0.030295   0.001717 -17.640  < 2e-16 ***
relv         0.021710   0.001874  11.584  < 2e-16 ***
sigma        1.221662   0.083224  14.679  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
--------------------------------------------
```

Name: Chan Ka Tsun
SID: 53563363

Question 4

Findings

Random effect model with moderation term

```
Call:
plm(formula = DV ~ know + expr + itr + ip + sex + age + sen +
    relv + sen * relv, data = df, model = "random", index = c("ID",
    "item"))

Balanced Panel: n=203, T=30, N=6090

Effects:
                 var std.dev share
idiosyncratic 0.1781  0.4220 0.884
individual    0.0233  0.1527 0.116
theta:  0.5494

Residuals :
   Min. 1st Qu.  Median 3rd Qu.    Max.
 -0.985  -0.429   0.155   0.325   0.960

Coefficients :
               Estimate  Std. Error t-value  Pr(>|t|)
(Intercept)  3.7442e-01  1.6922e-01  2.2126 0.0269589 *
know         2.4621e-02  4.1290e-02  0.5963 0.5509976
expr        -1.7497e-02  4.2028e-02 -0.4163 0.6771880
itr          4.9689e-02  1.3885e-02  3.5785 0.0003483 ***
ip          -6.8928e-03  1.1604e-02 -0.5940 0.5525166
sex         -6.3383e-02  2.6275e-02 -2.4123 0.0158815 *
age          1.0781e-02  5.5338e-03  1.9482 0.0514345 .
sen         -4.8975e-03  7.9531e-04 -6.1579 7.841e-10 ***
relv         4.9274e-03  9.1693e-04  5.3738 7.996e-08 ***
sen:relv    -1.4283e-05  1.4268e-05 -1.0011 0.3168235
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Total Sum of Squares:    1235
Residual Sum of Squares: 1081.6
R-Squared:      0.12416
Adj. R-Squared: 0.12286
F-statistic: 95.7633 on 9 and 6080 DF, p-value: < 2.22e-16
```

Name: Chan Ka Tsun
SID: 53563363

Panel Generalized Linear Model (combined with random effect)

```
Maximum Likelihood estimation
Newton-Raphson maximisation, 3 iterations
Return code 2: successive function values within tolerance limit
Log-Likelihood: -3357.611
11  free parameters
Estimates:
              Estimate Std. error t value  Pr(> t)
(Intercept) -1.342e+00  9.713e-01  -1.382 0.167090
know         1.815e-01  2.342e-01   0.775 0.438315
expr        -1.190e-01  2.378e-01  -0.500 0.616924
itr          3.049e-01  7.950e-02   3.835 0.000125 ***
ip          -5.040e-02  6.583e-02  -0.766 0.443935
sex         -3.832e-01  1.502e-01  -2.551 0.010753 *
age          6.548e-02  3.177e-02   2.061 0.039300 *
sen         -1.915e-02  4.593e-03  -4.169 3.07e-05 ***
relv         3.520e-02  5.533e-03   6.362 2.00e-10 ***
sen:relv    -2.176e-04  8.373e-05  -2.599 0.009342 **
sigma        1.220e+00  8.303e-02  14.694  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
--------------------------------------------
```

Fixed effect model with moderation term

```
Oneway (individual) effect Within Model

Call:
plm(formula = DV ~ know + expr + itr + ip + sex + age + sen +
    relv + sen * relv, data = df, model = "within", index = c("ID",
    "item"))

Balanced Panel: n=203, T=30, N=6090

Residuals :
   Min. 1st Qu.  Median 3rd Qu.    Max.
 -1.120  -0.375   0.102   0.317   1.070

Coefficients :
            Estimate  Std. Error t-value  Pr(>|t|)
sen      -4.8975e-03  7.9529e-04 -6.1581 7.847e-10 ***
relv      4.9274e-03  9.1690e-04  5.3740 7.998e-08 ***
sen:relv -1.4283e-05  1.4267e-05 -1.0011    0.3168
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Total Sum of Squares:    1194.5
Residual Sum of Squares: 1046.7
R-Squared:        0.12372
Adj. R-Squared: 0.093187
F-statistic: 276.909 on 3 and 5884 DF, p-value: < 2.22e-16
```

Name: Chan Ka Tsun
SID: 53563363

Appendix II – Coding

Question 1 Preprocessing the data (merging table) Python

Table Version I

```
#import libraries
import pandas as pd

#import data
path = '/Users/Lwmformula/Downloads/Assignment2/{}.csv'
df1 = pd.read_csv(path.format('eum_assignment',index_col=False))
df2 = pd.read_csv(path.format('eum_sensrelv',index_col=False))

#Create merge list
#perform groupby by ID
tmp = df1.groupby(['ID'])

#tidy up the first table
IDlist = []
klist = []
elist = []
itrlist = []
iplist = []
slist = []
agelist = []
tmplist = []
agelist = []
#create a dictionary
wholedict = dict.fromkeys([i for i in range(1,31)])
for i in wholedict: wholedict.update({i:[]})
wholedict.update({'ID':[],'know':[],'expr':[],
                  'itr':[],'ip':[],'sex':[],'age':[]})
```

```
#Merging two tables
for i in range(1,31):
    df['S{}'.format(i)] = sdict[i]
    df['R{}'.format(i)] = rdict[i]
#Recorder the table (tidy up only)
reordered = ['ID','know','expr','itr','ip','sex','age']
for i in range(1,31): reordered.append(i)
for i in range(1,31): reordered.append('S{}'.format(i))
for i in range(1,31): reordered.append('R{}'.format(i))
df = df[reordered]
#Summary table to discover the relationship
for i in range(1,31):
    print df.groupby(i).agg(['count','mean','std'])[['S{}'.format(i),'R{}'.format(i)]]
    print '=========================================================================='
```

```
for i,j in tmp:
    wholedict['ID'].append(i)
    wholedict['know'].append(j['know'].tolist()[0])
    wholedict['expr'].append(j['expr'].tolist()[0])
    wholedict['itr'].append(j['itr'].tolist()[0])
    wholedict['ip'].append(j['ip'].tolist()[0])
    wholedict['sex'].append(j['sex'].tolist()[0])
    wholedict['age'].append(j['age'].tolist()[0])
    tmp2 = j['prov'].tolist()
    tmp3 = []
    for ii in tmp2:
        if ii == 1: tmp3.append(1)
        else: tmp3.append(0)
    tmplist.append(tmp3)
for i in wholedict:
    try:
        for ii in tmplist:
            wholedict[i].append(ii[i-1])
    except: continue

#dealing with another table
df = pd.DataFrame.from_dict(wholedict)
df2 = df2[0:30]
sdict = dict.fromkeys([i for i in range(1,31)])
rdict = dict.fromkeys([i for i in range(1,31)])
stmp = df2['sen'].tolist()
rtmp = df2['relv'].tolist()
for i in sdict:
    sdict.update({i:[stmp[i-1] for ii in range(len(wholedict['ID']))]})
for i in rdict:
    rdict.update({i:[rtmp[i-1] for ii in range(len(wholedict['ID']))]})
```

Name: Chan Ka Tsun
SID: 53563363

Correlation Test

```
import numpy as np

#Find the correlation between Correct info and S and R
avg = []
Sscore = np.array(df2['sen'])
Rscore = np.array(df2['relv'])
for i in range(1,31):
    avg.append(np.mean(df[i].tolist()))
avg = np.array(avg)

#Structuring the lists in python, and conducting analysis in R
#API (python & R)
import rpy2.robjects as ro

#passing the data to R
ro.globalenv['R_avg'] = ro.FloatVector(avg)
ro.globalenv['R_Sscore'] = ro.FloatVector(Sscore)
ro.globalenv['R_Rscore'] = ro.FloatVector(Rscore)
#Correlation test
print (ro.r('cor(R_Sscore,R_avg)'))
print (ro.r('cor(R_Rscore,R_avg)'))

print Sscore
print Rscore
print avg
```

API– Ttest

```
from rpy2.robjects import pandas2ri
import rpy2.robjects.packages as rpackages

pandas2ri.activate()

ro.globalenv['rdf'] = pandas2ri.py2ri(iterdf)
#print (ro.r("rdf"))
print (ro.r("t.test(as.numeric(rdf$age)~as.numeric(rdf$sex))"))
```

Table Version II

```
stmp = []
rtmp = []
for i in df2['sen']:
    for j in range(len(df)):
        stmp.append(i)
stmp = np.array(stmp)
for i in df2['relv']:
    for j in range(len(df)):
        rtmp.append(i)
stmp = np.array(stmp)
rtmp = np.array(rtmp)

provtmp = []
for i in df1['prov'].tolist():
    if i == 1: provtmp.append(1)
    else: provtmp.append(0)

iterdf = df1.drop(df1.columns[8],axis=1,inplace=False)
iterdf['DV'] = provtmp
iterdf['sen'] = stmp
iterdf['relv'] = rtmp
```

Output as csv

```
iterdf.set_index(['ID','item'])
iterdf.to_csv('/Users/Lwmformula/Downloads/Assignment2/iterdf.csv',index=False,header=True)
```

Name: Chan Ka Tsun
SID: 53563363

Question 2-5 R

```
##############################################
############ Assignment 2 ##################
##############################################

#read table
df <- read.table("~/Downloads/GodBlessManhattan/iterdf.csv",header=TRUE,sep=",")

#libraries
library(plm)
library(pglm)

#pooling model (1)
df.pl <- plm(DV~know+expr+itr+ip+sex+age+sen+relv,data=df,index=c("ID","item"),model="pooling")
summary(df.pl)

#first difference model (2)
df.fd <- plm(DV~know+expr+itr+ip+sex+age+sen+relv,data=df,index=c("ID","item"),model="fd")
summary(df.fd)

#fixed effect model (3)
df.fe <- plm(DV~know+expr+itr+ip+sex+age+sen+relv,data=df,index=c("ID","item"),model="within")
summary(df.fe)

#random effect model (4)
df.re <- plm(DV~know+expr+itr+ip+sex+age+sen+relv,data=df,index=c("ID","item"),model="random")
summary(df.re)

#Panel Generalized Linear Model (pooling) (5)
df.gpl <- pglm(DV~know+expr+itr+ip+sex+age+sen+relv,data=df,index=c("ID","item"),model="pooling",family="binomial")
summary(df.gpl)

#Panel Generalized Linear Model (random effect) (6)
df.gre <- pglm(DV~know+expr+itr+ip+sex+age+sen+relv,data=df,index=c("ID","item"),model="random",family="binomial")
summary(df.gre)
```

```
#two phtests
phtest(df.fe,df.re)
phtest(df.fe,df.gre)

#fixed effect model with moderation (7)
df.fem <- plm(DV~know+expr+itr+ip+sex+age+sen+relv+sen*relv,data=df,index=c("ID","item"),model="within")
summary(df.fem)

#random effect model with moderation (8)
df.rem <- plm(DV~know+expr+itr+ip+sex+age+sen+relv+sen*relv,data=df,index=c("ID","item"),model="random")
summary(df.rem)

#Panel Generalized Linear Model (random effect) with moderation (9)
df.grem <- pglm(DV~know+expr+itr+ip+sex+age+sen+relv+sen*relv,data=df,index=c("ID","item"),model="random",family="binomia
summary(df.rem)


Requirements:
Question I: python code, t-test (refered to the screencap)

Question II & III.
Table I: pooling model (1) + random effect model (4) + Panel Generalized Linear Model (pooling) (5)
+ Panel Generalized Linear Model (random) (6)
Table II: first difference model (2) + fixed effect model (3)

Question IV:
Table I: random effect model with moderation (8) + Panel Generalized Linear Model (random effect) with moderation (9)
Table II: fixed effect model with moderation (7)
```