Project 1

BUAN 6383 SEC 002

MODELING FOR BUSINESS ANALYTICS

Group 03

Amna Malik

Ayasha Anupam

Bindu Musham

Victor Angel Nava

Yen-Ting Liu

Part I: Replicating Models from Class 1.

1. The Poisson Model

Consider the example related to billboard exposures from class. The associated data is in the file billboard.csv. Write code to estimate the parameters of the Poisson model using maximum likelihood estimation (MLE). Report your code, the estimated parameters and the maximum value of the log-likelihood.

```
def LL(params, retained,k):
    prob = []
    ll = []
    lambda1 = params
    for i in range(len(bb)):
        prob.append((pow(lambda1,k[i])*math.exp(-lambda1))/math.factorial(k[i]))
        ll.append(retained[i]*np.log(float(prob[i])))
    return ll
```

```
#Converting

def NLL(params,retained, k):
    return(-np.sum(LL(params,retained, k)))
```

```
#Declaring the parameters

k = bb['EXPOSURES']
retained = bb['PEOPLE']
params = np.array((1))
```

```
#Declaring the minimizing function

soln = minimize(NLL,
                args = (retained, k),
                x0 = np.array((1)),
                bounds = [(0.000001, None)],
                tol = 1e-10,
                options = {'ftol':1e-8}
                )
```

```
#The value of the poisson parameters for the optimal solution

lambda1 = soln.x[0]

print(soln)
print('\n')
print('The optimal value of lambda :',lambda1)
print('The max value of log-likelihood:',-soln.fun)
```

```
      fun: 929.0438827273003
 hess_inv: <1x1 LbfgsInvHessProduct with dtype=float64>
      jac: array([-2.27373677e-05])
  message: 'CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH'
     nfev: 18
      nit: 8
     njev: 9
   status: 0
  success: True
        x: array([4.45599945])


The optimal value of lambda : 4.455999451041339
The max value of log-likelihood: -929.0438827273003
```

Optimal Values: lambda: 4.456, LL: -929.044

2. The NBD Model

Next, write code (for the same dataset) to estimate the parameters of the NBD model using MLE. Report your code, the estimated parameters and the maximum value of the log-likelihood. Evaluate the NBD model vis-`a-vis the Poisson model; explain which is better and why.

```
def LL_nbd(params_nbd,retained_nbd,k):
    prob_nbd = []
    ll_nbd = []
    alpha, n = params_nbd
    for i in range(len(bb)):
        if i==0:
            prob_nbd.append(pow((alpha/(alpha+1)),n))
        else:
            prob_nbd.append(prob_nbd[i-1]*(n+k[i]-1)/(k[i]*(alpha+1)))
        ll_nbd.append(retained_nbd[i]*np.log(float(prob_nbd[i])))
    return ll_nbd
```

```
#Converting

def NLL_nbd(params_nbd,retained_nbd,k):
    return(-np.sum(LL_nbd(params_nbd,retained_nbd,k)))
```

```
k = bb['EXPOSURES']
retained_nbd = bb['PEOPLE']
params_nbd = np.array((1,1))
```

```
soln_nbd = minimize(NLL_nbd,
                    args = (retained_nbd,k),
                    x0 = np.array((1,1)),
                    bounds = [(0.000001, None),(0.000001, None)],
                    tol = 1e-10,
                    options = {'ftol':1e-8}
                    )
```

```
print(soln_nbd)
print('\n')
print('The max value of Log-Likelihood:', -soln_nbd.fun)
print('The optimal value of alpha:',soln_nbd.x[0])
print('The optimal value of n:',soln_nbd.x[1])
```

```
      fun: 649.688827483666
 hess_inv: <2x2 LbfgsInvHessProduct with dtype=float64>
      jac: array([-2.16004992e-04,  4.54747349e-05])
  message: 'CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH'
     nfev: 42
      nit: 11
     njev: 14
   status: 0
  success: True
        x: array([0.21751777, 0.96925942])


The max value of Log-Likelihood: -649.688827483666
The optimal value of alpha: 0.21751777147403442
The optimal value of n: 0.9692594239387519
```

Optimal values: n=0.969249, alpha = 0.217515, LL: -649.689

NBD model performs better on this dataset as its log likelihood value is greater than from the log likelihood value of poisson model. By incorporating heterogeneity of population proves better for modeling this data and for predicting exposures to the billboard.

3. The Poisson Regression

Now consider the khakichinos.com example from class; The associated data is in the file khakichinos.csv. Estimate all relevant parameters for Poisson regression using MLE. Report your code, the estimated parameters and the maximum value of the log-likelihood.

```
def LL_poi_reg(params, inc, sex, age, size,k):
    lambda0, beta1, beta2, beta3, beta4 = params
    ll_poi_reg = []
    for i in range(len(kk)):
        lambdai = lambda0*(math.exp(np.dot([beta1,beta2,beta3,beta4],[inc[i],sex[i],age[i],size[i]])))
        ll_poi_reg.append((k[i]*np.log(lambdai)) - (lambdai) - (np.log(float(math.factorial(k[i])))))
    return ll_poi_reg
```

```
def NLL_poi_reg(params,inc,sex,age,size,k):
    return(-np.sum(LL_poi_reg(params,inc,sex,age,size,k)))
```

```
k = kk['NumberofVisits']
inc = kk['LnInc']
sex = kk['Sex']
age = kk['LnAge']
size = kk['HHSize']
params = np.array((1,1,1,1,1))
```

```
soln_poi_reg = minimize(NLL_poi_reg,
               args = (inc,sex,age,size,k),
               x0 = params,
               bounds = [(0.000001, None),(None, None),(None, None),(None, None),(None, None)],
               tol = 1e-10,
               options = {'ftol':1e-8}
               )
```

```
print(soln_poi_reg)
print('\n')

print('The max value of Log-Likelihood:', -soln_poi_reg.fun)
print('The optimal value of lambda0:', soln_poi_reg.x[0])
print('Coefficients:','\n', 'beta1(inc):',soln_poi_reg.x[1],'\n','beta2(sex):',soln_poi_reg.x[2],'\n','beta3(age):',soln_poi_reg.
```

```
      fun: 6291.496756514583
 hess_inv: <5x5 LbfgsInvHessProduct with dtype=float64>
      jac: array([-2.39324436, -1.19662218, -0.07539711, -0.38335201, -0.10186341])
  message: 'CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH'
     nfev: 660
      nit: 84
     njev: 110
   status: 0
  success: True
        x: array([ 0.04389665,  0.09375347,  0.00423921,  0.5883308 , -0.03585181])


The max value of Log-Likelihood: -6291.496756514583
The optimal value of lambda0: 0.0438966463653882
Coefficients:
 beta1(inc): 0.09375347199692735
 beta2(sex): 0.004239211073739389
 beta3(age): 0.5883308038590668
 beta4(size): -0.035851806194411036
```

Optimal Values: lambda=0.04387243, beta1=0.093846, beta2=0.004265, beta3=0.588256, beta4=-0.03591, LL:-6291.497

4. The NBD Regression

Consider the khakichinos.com example again. Estimate all relevant parameters for NBD Regression using MLE. Report your code, the estimated parameters and the maximum value of the log-likelihood. Evaluate the NBD regression vis-`a-vis the Poisson regression; explain which is better and why.

```
def LL_NBD_reg(params,inc,sex,age,size,k):
    alpha,n,beta1,beta2,beta3,beta4 = params
    prob_nbd_reg = []
    ll_nbd_reg = []
    for i in range(len(k)):
        exp_beta = np.exp(np.dot([beta1,beta2,beta3,beta4],[inc[i],sex[i],age[i],size[i]]))
        p1 = np.log((math.gamma(n+k[i]))) - np.log((math.gamma(n)*math.factorial(k[i])))
        p2 = n * np.log((alpha/(alpha+exp_beta)))
        p3 = k[i] * np.log((exp_beta/(alpha+exp_beta)))
        ll_nbd_reg.append(p1+p2+p3)
    return ll_nbd_reg
```

```
def NLL_NBD_reg(params,inc,sex,age,size,k):
    return(-np.sum(LL_NBD_reg(params,inc,sex,age,size,k)))
```

```
params = np.array((1,1,1,1,1,1))
```

```
soln_nbd_reg = minimize(NLL_NBD_reg,
                args = (inc,sex,age,size,k),
                x0 = params,
                bounds = [(0.000001, None),(0.000001, None),(None, None),(None, None),(None, None),(None, None)],
                tol = 1e-10,
                options = {'ftol':1e-8}
                )
```

```
print(soln_nbd_reg)
print('\n')

print('The max value of Log-Likelihood:', -soln_nbd_reg.fun)
print('The optimal value of alpha:', soln_nbd_reg.x[0])
print('The optimal value of n:', soln_nbd_reg.x[1])
print('Coefficients:','\n', 'beta1(inc):',soln_nbd_reg.x[2],'\n','beta2(sex):',soln_nbd_reg.x[3],'\n','beta3(age):',soln_nbd_reg.
```

```
      fun: 2888.966126998539
 hess_inv: <6x6 LbfgsInvHessProduct with dtype=float64>
      jac: array([-0.004502  ,  0.13342287,  0.35888661,  0.00281943,  0.13396857,
        0.06302798])
  message: 'CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH'
     nfev: 644
      nit: 82
     njev: 92
   status: 0
  success: True
        x: array([ 8.16593568,  0.13874458,  0.07304028, -0.00955906,  0.90230171,
       -0.02434341])


The max value of Log-Likelihood: -2888.966126998539
The optimal value of alpha: 8.165935679830705
The optimal value of n: 0.13874457682129718
Coefficients:
 beta1(inc): 0.07304027525846186
 beta2(sex): -0.009559058718189933
 beta3(age): 0.9023017102823873
 beta4(size): -0.024343405293263242
```

Optimal Values: n:0.138751, alpha:8.192937559, beta1:0.073384, beta2:-0.0093, beta3:0.902081, beta4:-0.02434, LL: -2888.966

Even here, like in the former models, NBD regression performs better on this dataset as its log likelihood value is greater than from the log likelihood value of poisson regression. As log likelihood is an indicator of a better fit of the model, we prefer NBD regression over Poisson regression for this dataset. By

accounting for the heterogenous population, it proves better for modeling this data and for predicting exposures to the billboard.

5. **For each of the models above, can you provide some managerial takeaways?**

Poisson Model vs NBD Model: The log-likelihood is -649.69 for the NBD model and -929.04 for Poisson Model. So based on log-likelihood value, NBD model is better than the Poisson model.

Poisson Regression and NBD Regression: The log-likelihood value of the NBD regression is -2888.97 and -6291.50 for poisson regression. So, for predicting the 'number of visits', NBD regression is more efficient model than Poisson regression. Based on the output of NBD regression, we can say that 'age' is one of the major factors for deciding the number of visits as the coefficient of age (0.90) is highest as compared to rest of the variables. Income is also one of the determining factors and out of all four independent variables, sex is the least important variable for determining the number of visits. Sex is having a negative effect on the number of visits which shows that the females (0) are more likely to visit rather than the males (1). The household size is having a negative effect on the number of visits, and this is justifiable as well because the people with large number of family size would have other responsibilities as well and they won't be able to visit frequently. So, a female customer whose age is higher, with higher income and having small household size is more likely to visit rather than a person with lower age, less income and large household size.

Part II: Analysis of New Data

1. **Read books.csv and generate two new datasets – (a) books01.csv, with the structure of the dataset used in the billboard exposures example (i.e., with only two columns – (i) the number purchases**

```
books01.head()
```

	No. of Books Purchased	No. of Customers
0	1	753
1	2	362
2	3	175
3	4	126
4	5	82

```
books01.tail()
```

	No. of Books Purchased	No. of Customers
40	86	1
41	31	1
42	63	1
43	111	1
44	35	1

(ii) the number of people making the corresponding number of purchases), and (b) books02.csv, with the structure of the dataset used in the khakichinos.com example, with a new column containing a count of the number of books purchased from barnesandnoble.com by each customer, while keeping the demographic variables (remember to drop date, product, and price). Print the first and last few records of both new datasets.

```
books02.head()
```

	domain	userid	education	region	hhsz	age	income	child	race	country	qty
8173	barnesandnoble.com	6365661	5.0	1.0	2	11.0	7	0	1	0	1
8174	barnesandnoble.com	6396922	2.0	2.0	2	8.0	4	0	1	0	1
8175	barnesandnoble.com	8999933	4.0	3.0	5	10.0	3	1	1	0	1
8176	barnesandnoble.com	9573834	0.0	8.0	4	20.0	10	2	2	0	2
8177	barnesandnoble.com	9576277	0.0	5.0	15	40.0	35	5	5	0	5

```
books02.tail()
```

	domain	userid	education	region	hhsz	age	income	child	race	country	qty
9980	barnesandnoble.com	15695968	5.0	10.0	25	50.0	10	5	5	0	5
9981	barnesandnoble.com	15696910	0.0	6.0	8	16.0	8	2	2	0	2
9982	barnesandnoble.com	15698055	0.0	24.0	32	32.0	32	8	8	0	9
9983	barnesandnoble.com	15698341	0.0	8.0	12	16.0	12	2	2	0	2
9984	barnesandnoble.com	15698605	0.0	4.0	1	11.0	2	0	1	0	1

2. Develop a Poisson model using books01.csv. Report your code, the estimated parameters and the maximum value of the log-likelihood (and any other information you believe is relevant).

```
def LL_b1(params_b1, retained_b1, k_b1):
    prob_b1 = []
    ll_b1 = []
    lambda1_b1 = params_b1
    for i in range(len(k_b1)):
        prob_b1.append((pow(lambda1_b1,k_b1[i])*float(Decimal(math.exp(-lambda1_b1))/math.factorial(k_b1[i]))))
        ll_b1.append(retained_b1[i]*np.log(float(prob_b1[i]), where=0<prob_b1[i]))
    return ll_b1
```

```
def NLL_b1(params_b1,retained_b1, k_b1):
    return(-np.sum(LL_b1(params_b1,retained_b1, k_b1)))
```

```
#Declaring the parameters

k_b1 = books01['No. of Books Purchased']
retained_b1 = books01['No. of Customers']
params_b1 = np.array((1))
```

```
#Declaring the minimizing function

soln_b1 = minimize(NLL_b1,
                   args = (retained_b1, k_b1),
                   x0 = np.array((1)),
                   bounds = [(0.000001, None)],
                   tol = 1e-10,
                   options = {'ftol':1e-8}
                   )
```

```
print(soln_b1)
print('\n')

print('The maximum value of the Log-Likelihood:',-soln_b1.fun)
print('The optimal value of lambda:', soln_b1.x[0])
```

```
      fun: 7237.86797890745
 hess_inv: <1x1 LbfgsInvHessProduct with dtype=float64>
      jac: array([0.])
  message: 'CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL'
     nfev: 18
      nit: 8
     njev: 9
   status: 0
  success: True
        x: array([3.90397381])


The maximum value of the Log-Likelihood: -7237.86797890745
The optimal value of lambda: 3.903973805485939
```

Optimal Values: LL:-7237.867979, lambda = 3.903973863

3. **Develop a Poisson model using books02.csv, i.e., by ignoring the independent variables available. Report your code and confirm that the estimated parameters and the maximum value of the log-likelihood are identical to those obtained with the Poisson model developed using books01.csv.**

```
def LL_poi_reg_b2(params,edu, region, hhsz, age, inc, child, race, country, k_b2):
    lambda0, b1, b2, b3, b4, b5, b6, b7, b8 = params
    prob_reg_b2 = []
    ll_poi_reg_b2 = []
    for i in range(len(k_b2)):
        lambdai = lambda0 * np.exp(0)
        prob_reg_b2.append(((pow(lambda0,k_b2[i])*np.exp(-lambda0))/math.factorial(k_b2[i])))
        ll_poi_reg_b2.append(np.log(prob_reg_b2[i]))
    return ll_poi_reg_b2
```

```
def NLL_poi_reg_b2(params_b2,edu, region, hhsz, age, inc, child, race, country,k_b2):
    return(-np.sum(LL_poi_reg_b2(params_b2,edu, region, hhsz, age, inc, child, race, country,k_b2)))
```

```
#Declaring the parameters

k_b2 = books02['qty'].tolist()
edu = books02['education']
inc =books02['income']
region = books02['region']
age = books02['age']
hhsz = books02['hhsz']
child = books02['child']
race = books02['race']
country = books02['country']

params_b2 = np.array((1,0,0,0,0,0,0,0,0))
```

```
#Declaring the minimizing function

soln_poi_reg_b2 = minimize(NLL_poi_reg_b2,
               args = (edu,region, hhsz, age, inc, child, race, country, k_b2),
               x0 = np.array((1,0,0,0,0,0,0,0,0)),
               bounds = [(0.000001, None),(None,None),(None,None),(None,None),(None,None),
                         (None,None),(None,None),(None,None),(None,None)],
               tol = 1e-10,
               options = {'ftol':1e-8}
               )
```

```
print(soln_poi_reg_b2)
print('\n')
print('The maximum value of the Log-Likelihood:',-soln_poi_reg_b2.fun)
print('The optimal value of lambda:', soln_poi_reg_b2.x[0])
```

```
      fun: 7237.86797890744
 hess_inv: <9x9 LbfgsInvHessProduct with dtype=float64>
      jac: array([-0.00027285,  0.        ,  0.        ,  0.        ,  0.        ,
        0.        ,  0.        ,  0.        ,  0.        ])
  message: 'CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH'
     nfev: 90
      nit: 8
     njev: 9
   status: 0
  success: True
        x: array([3.90397331, 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        ])


The maximum value of the Log-Likelihood: -7237.86797890744
The optimal value of lambda: 3.9039733088186863
```

Optimal Values: LL:-7237.867979, lambda = 3.903973863

4. **Develop an NBD model using books01.csv. Report your code, the estimated parameters and the maximum value of the log-likelihood (and any other information you believe is relevant).**

```
def LL_nbd_b1(params_nbd_b1,retained_nbd_b1,k_nbd_b1):
    prob_nbd_b1 = []
    ll_nbd_b1 = []
    alpha, n = params_nbd_b1
    p_X0 = pow((alpha/(alpha+1)),n)
    for i in range(len(k_nbd_b1)):
        if i==0:
            prob_nbd_b1.append(p_X0*(n+k_nbd_b1[i]-1)/(k_nbd_b1[i]*(alpha+1)))
        else:
            prob_nbd_b1.append(prob_nbd_b1[i-1]*(n+k_nbd_b1[i]-1)/(k_nbd_b1[i]*(alpha+1)))
        ll_nbd_b1.append(retained_nbd_b1[i]*np.log(float(prob_nbd_b1[i])))
    return ll_nbd_b1
```

```
#Converting

def NLL_nbd_b1(params_nbd_b1,retained_nbd_b1,k_nbd_b1):
    return(-np.sum(LL_nbd_b1(params_nbd_b1,retained_nbd_b1,k_nbd_b1)))
```

```
#Declaring the parameters

k_nbd_b1 = books01['No. of Books Purchased']
retained_nbd_b1 = books01['No. of Customers']
params_nbd_b1 = np.array((1))
```

```
soln_nbd_b1 = minimize(NLL_nbd_b1,
               args = (retained_nbd_b1,k_nbd_b1),
               x0 = np.array((1,1)),
               bounds = [(0.000001, None),(0.000001, None)],
               tol = 1e-10,
               options = {'ftol':1e-8}
               )
```

```
print(soln_nbd_b1)
print('\n')
print('The maximum value of Log-Likelihood:', -soln_nbd_b1.fun)
print('The optimal value of alpha:',soln_nbd_b1.x[0])
print('The optimal value of n:',soln_nbd_b1.x[1])
```

```
      fun: 4351.514797405789
 hess_inv: <2x2 LbfgsInvHessProduct with dtype=float64>
      jac: array([ 0.00381988, -0.0007276 ])
  message: 'CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH'
     nfev: 39
      nit: 10
     njev: 13
   status: 0
  success: True
        x: array([0.38333861, 1.39774627])


The maximum value of Log-Likelihood: -4351.514797405789
The optimal value of alpha: 0.3833386131316343
The optimal value of n: 1.39774627287412
```

Optimal Values: LL:-4336.61349, alpha:0.391193853, n:1.41745878578013

5. **Develop an NBD model using books02.csv (again, ignoring the variables available). Report your code and confirm that the estimated parameters and the maximum value of the log-likelihood are identical to those obtained with the NBD model developed using books01.csv.**

```
def LL_nbd_b2(params_nbd_b2,edu, region, hhsz, age, inc, child, race, country,k_nbd_b2):
    prob_nbd_b2 = []
    ll_nbd_b2 = []
    exp_beta = 1
    alpha, n, b1, b2, b3, b4, b5, b6, b7, b8 = params_nbd_b2
    for i in range(len(k_nbd_b2)):
        p1_b2 = ((math.gamma(n+k_nbd_b2[i])))/(math.gamma(n)*math.factorial(k_nbd_b2[i]))
        p2_b2 = pow((alpha/(alpha+exp_beta)),n)
        p3_b2 = pow((exp_beta/(alpha+exp_beta)),k_nbd_b2[i])
        prob_nbd_b2.append(p1_b2*p2_b2*p3_b2)
        ll_nbd_b2.append(np.log(float(prob_nbd_b2[i])))
    return ll_nbd_b2
```

```
def NLL_nbd_b2(params_nbd_b2,edu, region, hhsz, age, inc, child, race, country,k_nbd_b2):
    return(-np.sum(LL_nbd_b2(params_nbd_b2,edu, region, hhsz, age, inc, child, race, country,k_nbd_b2)))
```

```
#Declaring the parameters
k_nbd_b2 = books02['qty'].tolist()
edu = books02['education']
inc =books02['income']
region = books02['region']
age = books02['age']
hhsz = books02['hhsz']
child = books02['child']
race = books02['race']
country = books02['country']
params_nbd_b2 = np.array((1,1,0,0,0,0,0,0,0,0))
```

```
soln_nbd_b2 = minimize(NLL_nbd_b2,
               args = (edu,region, hhsz, age, inc, child, race, country, k_nbd_b2),
               x0 = np.array((1,1,0,0,0,0,0,0,0,0)),
               bounds = [(0.000001, None),(0.000001, None),(None,None),(None,None),(None,None),(None,None),
                         (None,None),(None,None),(None,None),(None,None)],
               tol = 1e-10,
               options = {'ftol':1e-8}
               )
```

```
print(soln_nbd_b2)
print('\n')
print('The maximum value of Log-Likelihood:', -soln_nbd_b2.fun)
print('The optimal value of alpha:',soln_nbd_b2.x[0])
print('The optimal value of n:',soln_nbd_b2.x[1])
```

```
      fun: 4483.172477495755
 hess_inv: <10x10 LbfgsInvHessProduct with dtype=float64>
      jac: array([-0.03201421,  0.02237357,  0.        ,  0.        ,  0.        ,
        0.        ,  0.        ,  0.        ,  0.        ,  0.        ])
  message: 'CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH'
     nfev: 165
      nit: 11
     njev: 15
   status: 0
  success: True
        x: array([0.30800428, 1.20244768, 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ])


The maximum value of Log-Likelihood: -4483.172477495755
The optimal value of alpha: 0.30800427865887614
The optimal value of n: 1.2024476769076407
```

Optimal Values: LL:-4483.172, alpha:0.307996, n:1.202408

6. **Calculate the values of (i) reach, (ii) average frequency, and (iii) gross ratings points (GRPs) based on the NBD model. Show your work.**

```
def effectiveness(alpha, n, t):
    p = pow((alpha/(alpha+t)),n)
    e = (n*t)/alpha
    reach = (100 * (1-p))
    avg_freq = (e/(1-p))
    grps = (100*e)
    print(' i) reach:',reach,'\n ii) average frequency:',avg_freq,'\n iii) grps:',grps)
```

```
effectiveness(soln_nbd_b2.x[0],soln_nbd_b2.x[1],1)
```

```
 i) reach: 82.42886430942178
 ii) average frequency: 4.73620096359915
 iii) grps: 390.399666570667
```

7. **Identify all independent variables with missing values. How many values are missing in each? Drop any variable with many missing values (specify how you are defining 'many'). If the number of missing values are very few (again, specify how you are defining 'few'), delete the rows involved. For the remaining variables (if any), replace the missing values with the means of the corresponding variables. Report your code.**

```
#Identifying Missing Values

mv = pd.DataFrame()
mv['No. of Missing Values'] = (books3.isnull().sum())
mv['Percentage of Missing Values'] = (books3.isnull().mean()*100)

mv
```

	No. of Missing Values	Percentage of Missing Values
userid	0	0.000000
education	4831	72.570227
region	4	0.060087
hhsz	0	0.000000
age	3	0.045065
income	0	0.000000
child	0	0.000000
race	0	0.000000
country	0	0.000000
domain	0	0.000000
date	0	0.000000
product	0	0.000000
qty	0	0.000000
price	0	0.000000

Variable 'education' has 4831 missing values which accounts for 72.57% of the total number of observations for that column which shows it has very high number of missing values. As it has very high number of missing values so we can drop this variable.

Variable 'region' and 'age' has only 4 and 3 missing values respectively which accounts for only 0.06% and 0.05% of the total number of observations of those columns and hence it is very less. As the number of missing values is very less so we will drop the rows having missing values.

8. **Incorporate the available customer characteristics and estimate all relevant parameters for Poisson regression using MLE. Report your code, the estimated parameters and the maximum value of the log-likelihood (and any other information you believe is relevant). What are the managerial takeaways — which customer characteristics seem to be important?**

```
def LL_poi_reg_b(params, region, hhsz, age, inc, child, race, country, k_b):
    lambda0, beta1, beta2, beta3, beta4, beta5, beta6, beta7 = params
    lambdai = []
    prob_poi_reg_b = []
    ll_poi_reg_b = []
    for i in range(len(k_b)):
        lambdai.append(lambda0*(np.exp(np.dot([beta1,beta2,beta3,beta4,beta5,beta6,beta7],[region[i],hhsz[i],age[i],inc[i],child
        prob_poi_reg_b.append(((pow(lambdai[i],k_b[i])*np.exp(-lambdai[i]))/math.factorial(k_b[i])))
        ll_poi_reg_b.append(np.log(float(prob_poi_reg_b[i]), where=0<prob_poi_reg_b[i]))
    return ll_poi_reg_b
```

```
def NLL_poi_reg_b(params,region, hhsz, age, inc, child, race, country, k_b):
    return(-np.sum(LL_poi_reg_b(params,region, hhsz, age, inc, child, race, country, k_b)))
```

```
k_b = books3['qty'].tolist()

region = books3['region']
hhsz = books3['hhsz']
age = books3['age']
inc =books3['income']
child = books3['child']
race = books3['race']
country = books3['country']

params = np.array([1,0,0,0,0,0,0,0])
```

```
soln_poi_reg_b = minimize(NLL_poi_reg_b,
               args = (region, hhsz, age, inc, child, race, country, k_b),
               x0 = params,
               bounds = [(0.000001, None),(None,None),(None,None),(None,None),(None,None),(None,None),(None,None),(None,None)],
               tol = 1e-10,
               options = {'ftol':1e-8}
               )
```

```
print(soln_poi_reg_b)
print('\n')

print('The maximum value of Log-Likelihood:', -soln_poi_reg_b.fun)
print('The optimal value of lambda0:', soln_poi_reg_b.x[0])
print('Coefficients:')
coeff = pd.DataFrame()

coeff['Variables'] = books3.columns
coeff.drop([1,9], inplace=True)
coeff['Coefficients'] = soln_poi_reg_b.x
coeff.drop([0],inplace=True)
coeff
```

```
      fun: 7194.88404473305
 hess_inv: <8x8 LbfgsInvHessProduct with dtype=float64>
      jac: array([ 0.00354703, -0.02410161,  0.05811671,  0.03901732,  0.05848051,
        0.0091859 ,  0.0077307 ,  0.00263753])
  message: 'CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH'
     nfev: 486
      nit: 45
     njev: 54
   status: 0
  success: True
        x: array([ 3.75347569, -0.00654586,  0.00864534,  0.01393335,  0.01750699,
        0.02437513, -0.13237971, -0.20346533])


The maximum value of Log-Likelihood: -7194.88404473305
The optimal value of lambda0: 3.7534756925645394
```

Variables along with their corresponding coefficient values:

	Variables	Coefficients
2	region	-0.006546
3	hhsz	0.008645
4	age	0.013933
5	income	0.017507
6	child	0.024375
7	race	-0.132380
8	country	-0.203465

Age - For a one unit change in age, the difference in the logs of expected counts is expected to increase by 0.013 times, given the other predictor variables in the model are held constant.

Income - For a one unit change in income, the difference in the logs of expected counts is expected to increase by 0.017 times, given the other predictor variables in the model are held constant.

Child – For a one unit change in variable child, the difference in the logs of expected counts is expected to increase by 0.02 times, given the other predictor variables in the model are held constant.

Race – For a one unit change in race, the difference in the logs of expected counts is expected to decrease by 0.13 times, given the other predictor variables in the model are held constant.

Country – If the variable country is 1(country labeled as 1), the difference in the logs of expected counts is expected to decrease by 0.2 times as compared to when the country labeled in the data is not chosen, given the other predictor variables in the model are held constant.

These variables have more of an impact on the count of the number of books purchased then the variables region and household size as they are very low in magnitude.

Hence managerially there should be more of a focus on country, race, children and income of their clientele. Having a household with children, belonging to a higher income group, belonging to a specific race and country will bring in more business for the store. For determining the number of books purchased, country seems to be the most important factor based on the output of the Poisson regression. Importance of variables:

country > race > child > income > age > hhsz > region

9. **Estimate all relevant parameters for NBD regression using MLE. Report your code, the estimated parameters and the maximum value of the log-likelihood (and any other information you believe is relevant). What are the managerial takeaways — which customer characteristics seem to be important?**

```
def LL_nbd_reg_b(params_nbd_b,edu, region, hhsz, age, inc, child, race, country,k_nbd_b):
    exp_betas = []
    prob_nbd_b = []
    ll_nbd_b = []
    alpha, n, beta1, beta2, beta3, beta4, beta5, beta6, beta7, beta8 = params_nbd_b
    for i in range(len(k_nbd_b)):
        exp_betas.append(np.exp(np.dot([beta1,beta2,beta3,beta4,beta5,beta6,beta7],[region[i],hhsz[i],age[i],inc[i],child[i],race
        p1_b = ((math.gamma(n+k_nbd_b[i])))/(math.gamma(n)*math.factorial(k_nbd_b[i]))
        p2_b = pow((alpha/(alpha+exp_betas[i])),n)
        p3_b = pow((exp_betas[i]/(alpha+exp_betas[i])),k_nbd_b[i])
        prob_nbd_b.append(p1_b*p2_b*p3_b)
        ll_nbd_b.append(np.log(float(prob_nbd_b[i])))
    return ll_nbd_b
```

```
#Converting

def NLL_nbd_reg_b(params_nbd_b,edu, region, hhsz, age, inc, child, race, country,k_nbd_b):
    return(-np.sum(LL_nbd_reg_b(params_nbd_b,edu, region, hhsz, age, inc, child, race, country,k_nbd_b)))
```

```
k_nbd_b = books3['qty'].tolist()

region = books3['region']
hhsz = books3['hhsz']
age = books3['age']
inc =books3['income']
child = books3['child']
race = books3['race']
country = books3['country']

params = np.array([1,1,0,0,0,0,0,0,0,0])
```

```
soln_nbd_reg_b = minimize(NLL_nbd_reg_b,
               args = (edu, region, hhsz, age, inc, child, race, country,k_nbd_b),
               x0 = params,
               bounds = [(0.000001, None),(0.000001, None),(None,None),(None,None),(None,None),(None,None),(None,None),(None,None
               tol = 1e-10,
               options = {'ftol':1e-8}
               )
```

```
print(soln_nbd_reg_b)
print('\n')

print('The maximum value of Log-Likelihood:', -soln_nbd_reg_b.fun)
print('The optimal value of alpha:', soln_nbd_reg_b.x[0])
print('The optimal value of n:', soln_nbd_reg_b.x[1])
print('Coefficients:')
coeff = pd.DataFrame()

coeff['Variables'] = books3.columns
coeff['Coefficients'] = soln_nbd_reg_b.x
coeff.drop([0,1,9],inplace=True)
coeff
```

```
      fun: 4468.082935073004
 hess_inv: <10x10 LbfgsInvHessProduct with dtype=float64>
      jac: array([ 0.05566108, -0.14888428,  0.07003109,  0.09986252, -0.2996785 ,
       -0.47893991, -0.09131327, -0.034197  , -0.06493792,  0.        ])
  message: 'CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH'
     nfev: 473
      nit: 38
     njev: 43
   status: 0
  success: True
        x: array([ 0.32596759,  1.21322207, -0.00909679,  0.01063129,  0.014275  ,
        0.01685809,  0.02097826, -0.12236849, -0.19786011,  0.        ])


The maximum value of Log-Likelihood: -4468.082935073004
The optimal value of alpha: 0.32596758831029493
The optimal value of n: 1.2132220687762507
```

Variables along with their corresponding coefficient values:

	Variables	Coefficients
2	region	-0.009097
3	hhsz	0.010631
4	age	0.014275
5	income	0.016858
6	child	0.020978
7	race	-0.122368
8	country	-0.197860

Age - For a one unit change in age, the difference in the logs of expected counts is expected to increase by 0.014 times, given the other predictor variables in the model are held constant.

Income - For a one unit change in income, the difference in the logs of expected counts is expected to increase by 0.016 times, given the other predictor variables in the model are held constant.

Child – For a one unit change in variable child, the difference in the logs of expected counts is expected to increase by 0.02 times, given the other predictor variables in the model are held constant.

Race – For a one unit change in race, the difference in the logs of expected counts is expected to decrease by 0.12 times, given the other predictor variables in the model are held constant.

Country – If the variable country is 1(country labeled as 1), the difference in the logs of expected counts is expected to decrease by 0.19 times as compared to when the country labeled in the data is not chosen, given the other predictor variables in the model are held constant.

These variables have more of an impact on the count of the number of books purchased then the variables region and household size as they are very low in magnitude, therefore showing almost no impact.

Hence managerially there should be more of a focus on country, race, children and income of their clientele. Having a household with children, belonging to a higher income group, belonging to a specific race and country will bring in more business for the store.

There are no major differences between the two models when looking at the coefficients. However, by considering the heterogeneity of the population, nbd is a better fit then poisson as shown by the log likelihood value. It shows that for our over-dispersed count data, that is when the conditional variance exceeds the conditional mean, nbd is a better choice here as with extra parameter, it is modeling the over dispersion. Importance of variables:

country > race > child > income > age > hhsz > region

10. **Evaluate all the models developed using the log-likelihood ratio, AIC, and BIC. What are your recommendations on which model to use based on each of these criteria? Are the recommendations consistent? Explain why you are recommending the model you have selected. Are there any significant differences among the results from the models? If so, what exactly are these differences? Discuss what you believe could be causing the differences.**

```
def modelSelection(k,n,ll,model):
    AIC = (2*k) - (2*ll)
    BIC = k*np.log(n) - (2*ll)
    print('The AIC of the ',model,' model is: ',AIC,'and the BIC is: ',BIC)
    return AIC, BIC
```

```
models = {'NBD_reg' : {'n':len(books3),'k':(len(books3.axes[1])-1),'ll':(-soln_nbd_reg_b.fun)},
          'Poisson_reg':{'n':len(books3),'k':(len(books3.axes[1])-2),'ll':(-soln_poi_reg_b.fun)}}


for k, i in models.items():
    modelSelection(i['k'],i['n'],i['ll'],k)
```

```
The AIC of the  NBD_reg  model is:  8954.165802485164 and the BIC is:  9003.670567853722
The AIC of the  Poisson_reg  model is:  14405.768109602257 and the BIC is:  14449.772345485419
```

NBD has performed better consistently; it has maintained higher log- likelihood value and has given lower AIC and BIC. There is not much difference between the coefficients between the 2 models. NBD is adjusting the variance independently from the mean better then poisson. It shows that for our over-dispersed count data, that is when the conditional variance exceeds the conditional mean, nbd is a better choice here as with extra parameter, it is modeling the over dispersion.

Briefly summarize what you learned from this project. This is an open-ended question, so please include anything you found worthwhile — relating to the modeling process, insights from the process and models, any managerial takeaways that were insightful to you, and so on.

In this project, we are comparing different models based on their Log-likelihood values and later AIC and BIC as well. We are working with Poisson Model, NBD Model, Poisson Regression and NBD Regression. Out of Poisson Model and NBD Model, NBD model outperforms the Poisson model and the similarly with Poisson and NBD Regression. The log-likelihood of both the NBD model and NBD regression is higher than the Poisson Model and Poisson Regression respectively.

In Part 1, our goal was to predict billboard exposures as best as possible. When we used the Poisson Model to determine billboard exposures, we assumed the exposure rate (lambda) was identical for all. Since lambda also corresponds to the average billboard exposures for the Poisson model, a lambda equal to 4.456 implies the billboard is not too effective at reaching the masses who drive by the billboard. However, since the Poisson model was not predicting actual billboard exposures accurately, we used the NBD model which considered a heterogeneous population.

We can take the "limited exposure to billboard" into account and perhaps consider alternative measures of advertising. Some examples of these could be internet ads that appear whenever a user searches for similar products, paper mailings, or even TV commercials (if feasible). At the same time, choosing a better location of the billboard, or in a heavier traffic area where cars are more likely to slow down and have the opportunity for the drivers to look around.

In Part 2, our objective was to estimate the number of books sold. And based on log-likelihood value, NBD regression outperforms the Poisson Regression. This means while estimating the number of purchased made by a customer, we should consider individual characteristic of the customer. In NBD Regression, we include the concept of heterogeneity and that helps in increasing the log-likelihood value as compared to the Poisson Model. And the end, the values of AIC and BIC also supports the above finding. The NBD Regression model has lower AIC and BIC as compared to the Poisson Regression Model which again shows that NBD Regression is the better model than Poisson Regression for estimating the number of books sold.

The 'country' and 'race' comes up as the most effective variables and having a negative effect on the number of books bought by the customers. 'hhsz', 'age', 'income' and 'child' are having a positive effect on the number of books bought. 'region' is the least important variables in determining the number of books bought.

Importance according to the NBD Regression model:

country > race > child > income > age > hhsz > region