THE UNIVERSITY OF TEXAS AT DALLAS

NAVEEN JINDAL SCHOOL OF MANAGEMENT

COURSE NAME: MODELING FOR BUSINESS ANALYTICS

COURSE CODE: BUAN 6383/MIS 6386 001

GROUP NUMBER: 9

Project: 01

OBJECTIVE: Learn to build customized models

GROUP MEMBERS:

Anushree Sanagi Jagadeesh

Ruchika Gupta

Chetan Kulkarni

Poorvik Gambheer

Part I: Replicating Models from Class

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.

Python Code is available in the Python file. Below is a snapshot of the code.

In [268]:
```
poisson_pmf = lambda k,l: k*np.log(l)-l*(np.log(e))-np.log(factorial(k))
```

In [269]:
```
def LL(params,inputs):
    lambda0=params
    e=inputs['EXPOSURES']
    p=inputs['PEOPLE']
    sum=0
    for i in range(len(inputs)):
        sum+=poisson_pmf(e[i],lambda0)*p[i]
    return sum
```

In [270]:
```
def NLL(params, inputs):
    return(-(LL(params, inputs)))
```

In [271]:
```
final=minimize(NLL,
         args=inputs,
         x0=params,
         bounds=[(0.000001, None)],
        tol=1e-10,
        options={'ftol' : 1e-8},)
```

In [272]:
```
final
```

Out[272]:
```
      fun: array([929.04388273])
 hess_inv: <1x1 LbfgsInvHessProduct with dtype=float64>
      jac: array([-2.27373677e-05])
  message: 'CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH'
     nfev: 24
      nit: 10
     njev: 12
   status: 0
  success: True
        x: array([4.45599937])
```

The Value for Lambda is **4.455999368640959**

The Value of Maximum Log likelihood is **-929.0438827273031**

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.

Python Code is available in the Python file. Below is a snapshot of the code.

```
pmf_nbd=lambda a,n,k,t: (np.log(gamma(n+k))-(np.log(gamma(n))+np.log(factorial(k))))+(n*(np.log(a)-np.log(a+t)))+(k*(np.log(t
```

```
def NBDLL(params,inputs):
    e=inputs['EXPOSURES']
    p=inputs['PEOPLE']
    n=params[0]
    alpha=params[1]
    sum=0
    for i in range(len(inputs)):
        sum+=pmf_nbd(alpha,n,e[i],1)*p[i]
    return sum
```

```
def NLL(params, inputs):
    return(-(NBDLL(params, inputs)))
```

```
final=minimize(NLL,
         args=inputs,
         x0=params,
         bounds=[(0.000001, None),(0.000001, None)])
```

```
final
```

```
2]:       fun: 649.6888274836756
     hess_inv: <2x2 LbfgsInvHessProduct with dtype=float64>
          jac: array([ 6.82121023e-05, -3.41060513e-04])
      message: 'CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH'
         nfev: 42
          nit: 13
         njev: 14
       status: 0
      success: True
            x: array([0.96925927, 0.21751771])
```

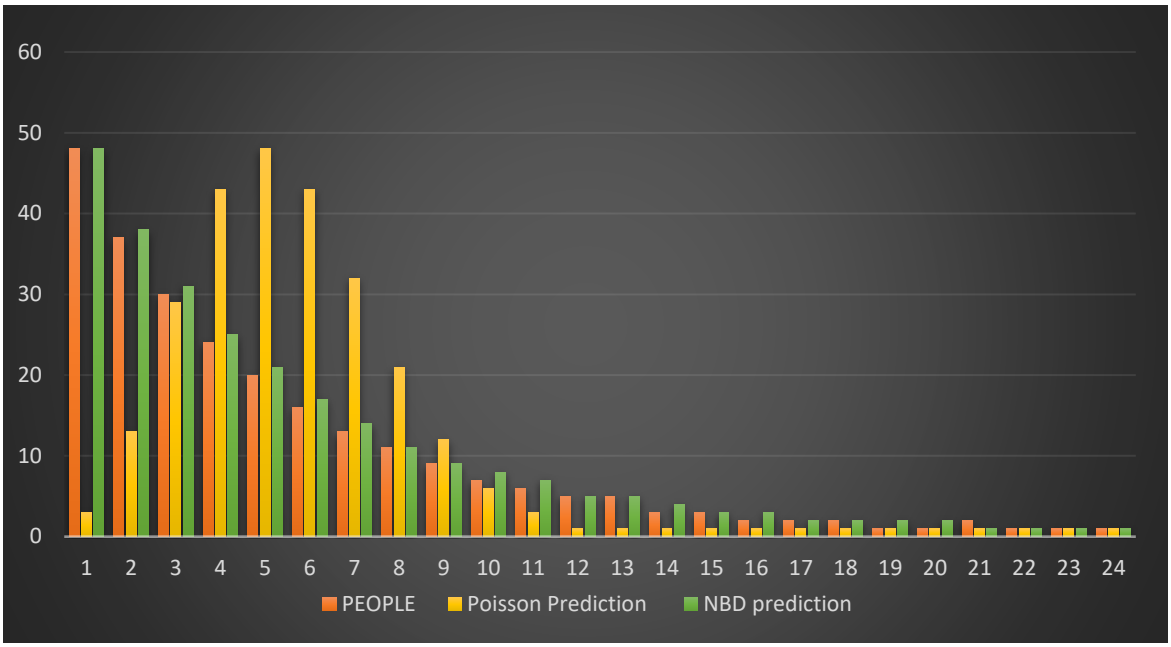
The Value for n is **0.96925927**

Alpha is **0.21751771**

The Value of Maximum Log likelihood is **- 649.6888274836756**

NBD stands better than Poisson because Log Likelihood Value is higher for NBD. The Poisson model makes an assumption that mean and the variance is same. NBD accommodates for over dispersion by not making any presumption about mean and variance.

Below are the predictions from both models, NBD predictions are closer to the actual.
Poisson is performing poorly.



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 parameter, and the maximum value of the log-likelihood.

Python Code is available in the Python file. Below is a snapshot of the code.

```
poisson_pmf = lambda k,l: k*np.log(l)-l*(np.log(e))-np.log(factorial(k))
```

```
params = np.array([0.01,0.01,0.1,0.01,0.01])
```

```
def LL(params,inputs):
    k=inputs['NumberofVisits'].tolist()
    kc1=inputs.drop(['NumberofVisits'],axis=1)
    a=kc1.to_numpy().tolist()
    beta=params
    lambda0=beta[4]
    beta=np.delete(beta, [4])
    lambdai=0
    sum2=0
    sum1=0

    for i in range(len(a)):
        sum1=0
        for j in range(len(a[i])):
            sum1=sum1+beta[j]*a[i][j]
        lambdai=lambda0*np.exp(sum1)
        sum2=sum2+poisson_pmf(k[i],lambdai)
    return sum2
```

```
def NegLL(params, inputs):
    return(-(LL(params, inputs)))
```

```
final=minimize(NegLL,
         args=inputs,
         x0=params,
         bounds=[(None, None), (None, None), (None, None), (None, None), (0.000001, None)])
```

```
print("The Value for lambda is ",final.x[4],"Beta parameters are ",final.x[0:4],".The Value of Maximum Log likelihood is ",-1
```

The Value for lambda is **0.043843803145908994**

Beta parameters are **[0.0937195 0.0042861 0.58831778 -0.03591484]**

The Value of Maximum Log likelihood is **-6291.496753218185**

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.

The Value for n is **0.1387428478752967**

Alpha is **8.1186808876206**

The Values for the beta parameters are **[0.07263612 -0.00942678 0.90191368 -0.02437216]**

The Value of Maximum Log likelihood is **-2888.966153766386**

In [4]:
```
pmf_nbd=lambda a,n,k,t: (mpmath.gamma(n+k)/(mpmath.gamma(n)*factorial(k)))*((a/(a+t))**n)*((t/(a+t))**k)
```

In [5]:
```
def NBDLL(params,inputs):
    k=inputs['NumberofVisits'].tolist()
    kc1=inputs.drop(['NumberofVisits'],axis=1)
    a=kc1.to_numpy().tolist()
    beta=params
    alpha=beta[5]
    n=beta[4]
    beta=np.delete(beta, [4])
    beta=np.delete(beta, [4])
    lambdai=0
    sum2=0
    sum1=0
    for i in range(len(a)):
        sum1=0
        for j in range(len(a[i])):
            sum1=sum1+beta[j]*a[i][j]
        t=math.exp(sum1)
        sum2=sum2+math.log(pmf_nbd(alpha,n,k[i],t))
    return sum2
```

In [6]:
```
def NegLL(params, inputs):
    return(-(NBDLL(params, inputs)))
```

In [7]:
```
final=minimize(NegLL,
         args=inputs,
         x0=params,
         bounds=[(None, None), (None, None), (None, None), (None, None), (0.000001, None),(0.000001, None)])
```

NBD Regression is better than the Poisson Regression because NBD has a higher log Likelihood.

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

For billboards example, NBD model performs better as it has a higher log likelihood. It is evident from the graphs that the predictions from NBD are closer to the actuals than poisons. As a management we would target customers who saw the billboards more than once but less than three times. They constitute majority of the future sales. We can use the Poisson model to estimate that in a month at least 16 people look at the billboards.

Nbd is better because it incorporates sum of probabilities. The estimation is accurate because, the probability is dependent on the previous probabilities. Hence, NBD gives better predictions

For the khakichinos.com NBD regression performs better with greater log likelihood.

Most Relevant parameters from Poisson Distribution is LnAge.
Even for NBD, LnAge stands out to be most relevant and second most relevant parameter is LnInc

As a manager we would target at older customer with higher income. As from the beta coefficients its evident that, with every unit increase in the age, the number of visitors increases by 0.9. That is every 2 unit increase in age will cause 1 increase in the number of visitors.

	Poisson Parameters	**NBD Parameters**
Maximum Likelihood	-6291.49675885234	-2888.96611657162
Lambda	0.043843803145908994	
Alpha		8.197058003322073

n		0.13875675766248502
LnInc	0.09383211	0.07347589
Sex	0.00427661	-0.00919222
LnAge	0.58845273	0.90195373
HHSize	-0.03590396	-0.02435738

Part 2: 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, and (ii) the number of people making the corresponding number of purchases) (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).

First few rows of books01.csv

	Qty_Purchased	Number_of_People
0	0	7639
1	1	753
2	2	362
3	3	175
4	4	126

Last few rows of books01.csv

	Qty_Purchased	Number_of_People
41	62	1
42	63	1
43	83	1
44	86	1
45	111	1

First few rows of books02.csv

	userid	education	region	hhsz	age	income	child	race	country	qty
1	6388054	2.0	4.0	1	6.0	5	0	1	0	0
2	6421559	5.0	4.0	4	5.0	6	0	1	0	0
3	6467806	NaN	2.0	2	6.0	3	0	1	0	0
4	6628110	4.0	4.0	5	4.0	7	1	1	0	0
5	6631403	5.0	3.0	1	10.0	3	0	1	1	0

Last few rows of books02.csv

	userid	education	region	hhsz	age	income	child	race	country	qty
1807	15695968	1.0	2.0	5	10.0	2	1	1	0	5
1808	15696910	NaN	3.0	4	8.0	4	1	1	0	2
1809	15698055	NaN	3.0	4	4.0	4	1	1	0	9
1810	15698341	NaN	4.0	6	8.0	6	1	1	0	2
1811	15698605	NaN	4.0	1	11.0	2	0	1	0	1

Full data set for Books 02.csv

```
#Final table contains 9451 rows and 10 columns
bk2
```

	userid	education	region	hhsz	age	income	child	race	country	qty
1	6388054	2.0	4.0	1	6.0	5	0	1	0	0
2	6421559	5.0	4.0	4	5.0	6	0	1	0	0
3	6467806	NaN	2.0	2	6.0	3	0	1	0	0
4	6628110	4.0	4.0	5	4.0	7	1	1	0	0
5	6631403	5.0	3.0	1	10.0	3	0	1	1	0
...	...	...	...	...	...	...	...	...	...	...
1807	15695968	1.0	2.0	5	10.0	2	1	1	0	5
1808	15696910	NaN	3.0	4	8.0	4	1	1	0	2
1809	15698055	NaN	3.0	4	4.0	4	1	1	0	9
1810	15698341	NaN	4.0	6	8.0	6	1	1	0	2
1811	15698605	NaN	4.0	1	11.0	2	0	1	0	1

9451 rows × 10 columns

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).

In [636]:
```
inputs=bk1.copy()
```

In [637]:
```
poisson_pmf = lambda k,l: k*np.log(l)-l*(np.log(e))-np.log(factorial(k))
```

In [638]:
```
params=0.1
```

In [639]:
```
inputs.columns=['Qty_Purchased','Number_of_People']
```

In [640]:
```
def LL(params,inputs):
    lambda0=params
    e=inputs['Qty_Purchased']
    p=inputs['Number_of_People']
    sum=0
    for i in range(len(inputs)):
        sum+=poisson_pmf(e[i],lambda0)*p[i]
    return sum
```

In [641]:
```
def NLL(params, inputs):
    return(-(LL(params, inputs)))
```

In [642]:
```
def p_callback(params):
    print(params)
```

In [643]:
```
final=minimize(NLL,
         args=inputs,
         x0=params,
         bounds=[(0.000001, None)],callback=p_callback)
```

```
[1.1]
[0.86518938]
[0.69368856]
[0.75703656]
[0.74911788]
[0.74848505]
[0.7484922]
```

In [644]:
```
print("The estimated value for lambda is ",final.x)
print("The value of Maximum Log likelihood is ",-1*final.fun)
```

estimated value for lambda	**0.7484922**
Maximum Log likelihood	**-18921.91842943**

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

In [665]:
```
bk2.drop(['userid'],axis=1,inplace=True)
```

In [666]:
```
bk2
```

Out[666]:

	education	region	hhsz	age	income	child	race	country	qty
1	2.0	4.0	1	6.0	5	0	1	0	0
2	5.0	4.0	4	5.0	6	0	1	0	0
3	NaN	2.0	2	6.0	3	0	1	0	0
4	4.0	4.0	5	4.0	7	1	1	0	0
5	5.0	3.0	1	10.0	3	0	1	1	0
...	...	...	...	...	...	...	...	...	...
1807	1.0	2.0	5	10.0	2	1	1	0	5
1808	NaN	3.0	4	8.0	4	1	1	0	2
1809	NaN	3.0	4	4.0	4	1	1	0	9
1810	NaN	4.0	6	8.0	6	1	1	0	2
1811	NaN	4.0	1	11.0	2	0	1	0	1

9451 rows × 9 columns

In [667]:
```
inputs=bk2.copy()
```

In [696]:
```
params = 1
```

In [706]:
```
def LL(params,inputs):
    k=inputs['qty'].tolist()
    lambda0=params
    sum1=0
    for i in range(len(k)):
        sum1=sum1+(poisson_pmf(k[i],lambda0))
    return sum1
```

In [707]:
```
def NLL(params, inputs):
    return(-1*(LL(params, inputs)))
```

In [708]:
```
inputs[:3]
```

Out[708]:

	education	region	hhsz	age	income	child	race	country	qty
1	2.0	4.0	1	6.0	5	0	1	0	0
2	5.0	4.0	4	5.0	6	0	1	0	0
3	NaN	2.0	2	6.0	3	0	1	0	0

In [709]:
```
LL(params,inputs)
```

Out[709]:
```
-19249.61978239895
```

```
In [711]: final=minimize(NLL,
                         args=inputs,
                         x0=params,
                         bounds=[(0.000001, None)],callback=p_callback)

[0.66011651]
[0.77819378]
[0.75199981]
[0.74835226]
[0.7484935]
[0.7484913]

In [712]: final

Out[712]:       fun: array([18921.91842944])
 hess_inv: <1x1 LbfgsInvHessProduct with dtype=float64>
      jac: array([-0.02073648])
  message: 'CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH'
     nfev: 16
      nit: 6
     njev: 8
   status: 0
  success: True
        x: array([0.7484913])

In [715]: print("The estimated value for lambda is ",final.x[0])
          print("The Value of Maximum Log likelihood is ",-1*final.fun)
          print("We get the same values as the previous question")

The estimated value for lambda is  0.7484913027336835
The Value of Maximum Log likelihood is  [-18921.91842944]
We get the same values as the previous question
```

estimated value for lambda	**0.7484913027336835**
Value of Maximum Log likelihood	**-18921.91842944**

We get the same values as the previous question

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).

```
In [740]: params = np.array([1,1])

In [741]: pmf_nbd=lambda a,n,k,t: (np.log(gamma(n+k))-(np.log(gamma(n))+np.log(factorial(k))))+(n*(np.log(a)-np.log(a+t)))+(k*(np.log(t)-np

In [742]: bk1

Out[742]:
```

	Qty_Purchased	Number_of_People
0	0	7639
1	1	753
2	2	362
3	3	175
4	4	126
5	5	82
6	6	74
7	7	30
8	8	48
9	9	31
10	10	20
11	11	12
12	12	12
13	13	11
14	14	3

15	15	6
16	16	8
17	17	4
18	18	1
19	19	5
20	20	3
21	21	6
22	23	3
23	24	1
24	27	3
25	28	5
26	29	2
27	30	3
28	31	1
29	32	1
30	33	1
31	34	3
32	35	1
33	37	2
34	39	2
35	43	2
36	46	1
37	47	1
37	47	1
38	50	1
39	56	1
40	58	1
41	62	1
42	63	1
43	83	1
44	86	1
45	111	1

In [743]:
```
inputs=bk1.copy()
```

In [744]:
```
def NBDLL(params,inputs):
    e=inputs['Qty_Purchased']
    p=inputs['Number_of_People']
    n=params[0]
    alpha=params[1]
    sum=0
    for i in range(len(inputs)):
        sum+=(pmf_nbd(alpha,n,e[i],1))*p[i]
    return sum
```

In [745]:
```
def NLL(params, inputs):
    return(-(NBDLL(params, inputs)))
```

In [746]:
```
def p_callback(params):
    print(params)
```

In [747]:
```
final=minimize(NLL,
         args=inputs,
         x0=params,
         bounds=[(0.000001, None),(0.000001, None)],callback=p_callback)
```

```
[0.9996994  1.35725779]
[0.14907399 0.40179165]
[0.0979495 0.2639978]
[0.08217807 0.22148967]
[0.08139086 0.21936793]
[0.08134365 0.2192407 ]
[0.08134119 0.21923405]
```

In [748]:
```
print("The Value for n is ",final.x[0]," and Alpha is ",final.x[1])
print("The Value of Maximum Log likelihood is ",-1*final.fun)
```

```
The Value for n is  0.08134118622615552  and Alpha is  0.21923404867443047
The Value of Maximum Log likelihood is  -8603.49884703562
```

n	**0.08134118622615552**
Alpha	**0.21923404867443047**
Maximum Log likelihood	**-8603.49884703562**

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-likeli hood are identical to those obtained with the NBD model developed using books01.csv. 2

In [721]:
```
inputs=bk2.copy()
```

In [722]:
```
params = np.array([1,1])
```

In [723]:
```
pmf_nbd=lambda a,n,k,t: (np.log(gamma(n+k))-(np.log(gamma(n))+np.log(factorial(k))))+(n*(np.log(a)-np.log(a+t)))+(k*(np.log(t)-np
```

In [735]:
```
def NBDLL(params,inputs):
    k=inputs['qty'].tolist()
    kc1=inputs.drop(['qty'],axis=1)
    beta=params
    alpha=beta[1]
    n=beta[0]
    sum1=0
    for i in range(len(k)):
        sum1=sum1+pmf_nbd(alpha,n,k[i],1)
    return sum1
```

In [736]:
```
def NegLL(params, inputs):
    return(-(NBDLL(params, inputs)))
```

In [737]:
```
final=minimize(NegLL,
        args=inputs,
        x0=params,
        bounds=[(0.000001, None),(0.000001, None)]
             ,callback=p_callback)
```

```
[0.99969942 1.35725546]
[0.14906563 0.401912  ]
[0.0979436  0.26407576]
[0.08216343 0.22152897]
[0.08137654 0.21940732]
[0.08132885 0.21927876]
[0.08132649 0.21927238]
```

In [738]:
```
final
```

Out[738]:
```
      fun: 8603.726982003966
 hess_inv: <2x2 LbfgsInvHessProduct with dtype=float64>
      jac: array([-7891.64424038,  2926.90274518])
  message: 'CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH'
     nfev: 63
      nit: 7
     njev: 21
   status: 0
  success: True
        x: array([0.08132649, 0.21927238])
```

In [749]:
```
print("The Value for n is ",final.x[0]," and Alpha is ",final.x[1])
print("The Value of Maximum Log likelihood is ",-1*final.fun)
print('The values are similar to the values obtained when using book1.csv')
```

```
The Value for n is  0.08134118622615552  and Alpha is  0.21923404867443047
The Value of Maximum Log likelihood is  -8603.49884703562
The values are similar to the values obtained when using book1.csv
```

n	**0.08134118622615552**
Alpha	**0.21923404867443047**
Maximum Log likelihood	**-8603.49884703562**

The values are similar to the values obtained when using book1.csv

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

reach for the NBD model is defined as 100*(1-P(x=0|n,alpha)), alpha value obtained after the MLE implementation is **0.2566150324283152** and n is **1.1846549361700596**

In [750]:
```
params = np.array([1,1])
inputs=bk1.copy()
inputs.columns=['Qty_Purchased','Number_of_People']
```

In [751]:
```
pmf_nbd=lambda a,n,k,t: (np.log(gamma(n+k))-(np.log(gamma(n))+np.log(factorial(k))))+(n*(np.log(a)-np.log(a+t)))+(k*(np.log(t)-np
```

In [752]:
```
def NBDLL(params,inputs):
    e=inputs['Qty_Purchased']
    p=inputs['Number_of_People']
    n=params[0]
    alpha=params[1]
    sum=0
    for i in range(len(inputs)):
        sum+=pmf_nbd(alpha,n,e[i],1)*p[i]
    return sum
```

In [753]:
```
def NLL(params, inputs):
    return(-(NBDLL(params, inputs)))

def p_callback(params):
    print(params)
```

```
In [754]: final=minimize(NLL,
                  args=inputs,
                  x0=params,
                  bounds=[(0.000001, None),(0.000001, None)],callback=p_callback)

          [0.9996994  1.35725779]
          [0.14907399 0.40179165]
          [0.0979495 0.2639978]
          [0.08217807 0.22148967]
          [0.08139086 0.21936793]
          [0.08134365 0.2192407 ]
          [0.08134119 0.21923405]

In [755]: print("The Value for n is ",final.x[0]," and Alpha is ",final.x[1])
          print("The Value of Maximum Log likelihood is ",-1*final.fun)

          The Value for n is  0.08134118622615552  and Alpha is  0.21923404867443047
          The Value of Maximum Log likelihood is  -8603.49884703562

In [757]: n=final.x[0]
          alpha=final.x[1]
          pmf_0=(alpha/(alpha+1))**n
          reach = 100*(1-pmf_0)
          print('reach is',reach)

          reach is 13.026639691492758

In [758]: reach

Out[758]: 13.026639691492758

In [760]: E_x= n/alpha

In [761]: print('E(x) is ',E_x )

          E(x) is  0.3710244221551086

In [762]: Average_Frequency=E_x/(1-pmf_0)

In [763]: grps= reach*Average_Frequency

In [764]: print('reach is ',reach)
          print('Average Frequency is' ,Average_Frequency)
          print('grps is ',grps)

          reach is  13.026639691492758
          Average Frequency is 2.8481974702763266
          grps is  37.10244221551086
```

reach	**13.026639691492758**
Average Frequency	**2.8481974702763266**
groups	**37.10244221551086**

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.

- **Below are the variables with the count of missing values in each column**

Variable	# Nulls
education	6914
region	11
hhsz	0
age	
income	0
child	0
race	0
country	0
qty	0

- **Below is the % of missing values compared to the total number of rows in the table.**

Variable	**#Nulls**	**% of Nulls**
education	6914	73%
region	11	0%
hhsz	0	0%
age	1	0%
income	0	0%
child	0	0%
race	0	0%
country	0	0%
qty	0	0%
Total Rows in data set	**9451**	**73%**

- **We can see that the % of nulls are 73 which is very high and we cannot delete these rows. So, we impute the mean for all missing values in the 'education' variable.**
- **The other two columns ('region' and 'age')have negligible number of rows having the missing values, so they are removed from the data set.**

Below is the code snippet for the changes made in the book 2.csv for treating Nulls.

In [569]:
```
bk2
```

Out[569]:

	education	region	hhsz	age	income	child	race	country	qty
1	2.0	4.0	1	6.0	5	0	1	0	0
2	5.0	4.0	4	5.0	6	0	1	0	0
3	NaN	2.0	2	6.0	3	0	1	0	0
4	4.0	4.0	5	4.0	7	1	1	0	0
5	5.0	3.0	1	10.0	3	0	1	1	0
...	...	...	...	...	...	...	...	...	...
1807	1.0	2.0	5	10.0	2	1	1	0	5
1808	NaN	3.0	4	8.0	4	1	1	0	2
1809	NaN	3.0	4	4.0	4	1	1	0	9
1810	NaN	4.0	6	8.0	6	1	1	0	2
1811	NaN	4.0	1	11.0	2	0	1	0	1

9451 rows × 9 columns

Treating missing values

In [570]:
```
#below are the number of missing values by each column in the books table.
print('Below is the list of columns and the number of nulls in each colum')
bk2.isna().sum()
```

```
Below is the list of columns and the number of nulls in each colum
```

Out[570]:
```
education    6914
region         11
hhsz            0
age             1
income          0
child           0
race            0
country         0
qty             0
dtype: int64
```

In [584]:
```
books_2 = pd.DataFrame(bk2, columns = ["education", "region", "hhsz","age","income","child","race","country","qty"])
books_2.head()
```

Out[584]:

	education	region	hhsz	age	income	child	race	country	qty
1	2.0	4.0	1	6.0	5	0	1	0	0
2	5.0	4.0	4	5.0	6	0	1	0	0
3	NaN	2.0	2	6.0	3	0	1	0	0
4	4.0	4.0	5	4.0	7	1	1	0	0
5	5.0	3.0	1	10.0	3	0	1	1	0

In [585]:
```
books_2.isna().sum()
```

Out[585]:
```
education    6914
region         11
hhsz            0
age             1
income          0
child           0
race            0
country         0
qty             0
dtype: int64
```

In [586]:
```
#below is the summary of the education variable
books_2.education.describe()
```

Out[586]:
```
count    2537.000000
mean        2.749310
std         1.430923
min         0.000000
25%         1.000000
50%         2.000000
75%         4.000000
max         5.000000
Name: education, dtype: float64
```

In [587]:
```
#imputing the mean of education in the education column instead of deleting it
#since the proportion of missing values to the total rows is high
books_3=books_2.copy()
books_3.education.fillna(books_3.education.mean(), inplace = True)
books_3.head()
```

Out[587]:

	education	region	hhsz	age	income	child	race	country	qty
1	2.00000	4.0	1	6.0	5	0	1	0	0
2	5.00000	4.0	4	5.0	6	0	1	0	0
3	2.74931	2.0	2	6.0	3	0	1	0	0
4	4.00000	4.0	5	4.0	7	1	1	0	0
5	5.00000	3.0	1	10.0	3	0	1	1	0

In [588]:
```
books_3.isna().sum()
```

Out[588]:
```
education     0
region       11
hhsz          0
age           1
income        0
child         0
race          0
country       0
qty           0
dtype: int64
```

In [589]:
```
#deleting the nulls rows which have nulls in the region column since we have only 46 rows
#which is negligible compared to the entire data set
books_3 = books_3.dropna(axis=0, subset=['region'])
```

In [590]:
```
books_3.isna().sum()
```

Out[590]:
```
education    0
region       0
hhsz         0
age          1
income       0
child        0
race         0
country      0
qty          0
dtype: int64
```

In [591]:
```
#deleting the nulls rows which have nulls in the age column since we have only 46 rows
#which is negligible compared to the entire data set
books_3 = books_3.dropna(axis=0, subset=['age'])
```

In [592]:
```
books_3.isna().sum()
```

Out[592]:
```
education    0
region       0
hhsz         0
age          0
income       0
child        0
race         0
country      0
qty          0
dtype: int64
```

In [593]:
```
books_3
```

Out[593]:

	education	region	hhsz	age	income	child	race	country	qty
1	2.00000	4.0	1	6.0	5	0	1	0	0
2	5.00000	4.0	4	5.0	6	0	1	0	0
3	2.74931	2.0	2	6.0	3	0	1	0	0
4	4.00000	4.0	5	4.0	7	1	1	0	0
5	5.00000	3.0	1	10.0	3	0	1	1	0
...	...	...	...	...	...	...	...	...	...
1807	1.00000	2.0	5	10.0	2	1	1	0	5
1808	2.74931	3.0	4	8.0	4	1	1	0	2
1809	2.74931	3.0	4	4.0	4	1	1	0	9
1810	2.74931	4.0	6	8.0	6	1	1	0	2
1811	2.74931	4.0	1	11.0	2	0	1	0	1

9439 rows × 9 columns

In [598]:
```
bk3=books_3
```

In [598]:
```
bk3=books_3
```

In [599]:
```
bk3
```

Out[599]:

	education	region	hhsz	age	income	child	race	country	qty
1	2.00000	4.0	1	6.0	5	0	1	0	0
2	5.00000	4.0	4	5.0	6	0	1	0	0
3	2.74931	2.0	2	6.0	3	0	1	0	0
4	4.00000	4.0	5	4.0	7	1	1	0	0
5	5.00000	3.0	1	10.0	3	0	1	1	0
...	...	...	...	...	...	...	...	...	...
1807	1.00000	2.0	5	10.0	2	1	1	0	5
1808	2.74931	3.0	4	8.0	4	1	1	0	2
1809	2.74931	3.0	4	4.0	4	1	1	0	9
1810	2.74931	4.0	6	8.0	6	1	1	0	2
1811	2.74931	4.0	1	11.0	2	0	1	0	1

9439 rows × 9 columns

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?

In [765]:
```
inputs=bk3.copy()
```

In [766]:
```
bk3
```

Out[766]:

	education	region	hhsz	age	income	child	race	country	qty
1	2.00000	4.0	1	6.0	5	0	1	0	0
2	5.00000	4.0	4	5.0	6	0	1	0	0
3	2.74931	2.0	2	6.0	3	0	1	0	0
4	4.00000	4.0	5	4.0	7	1	1	0	0
5	5.00000	3.0	1	10.0	3	0	1	1	0
...	...	...	...	...	...	...	...	...	...
1807	1.00000	2.0	5	10.0	2	1	1	0	5
1808	2.74931	3.0	4	8.0	4	1	1	0	2
1809	2.74931	3.0	4	4.0	4	1	1	0	9
1810	2.74931	4.0	6	8.0	6	1	1	0	2
1811	2.74931	4.0	1	11.0	2	0	1	0	1

9439 rows × 9 columns

In [767]:
```
params = np.array([1,1,1,1,1,1,1,1,1])
```

In [768]:
```
def LL(params,inputs):
    k=inputs['qty'].tolist()
    kc1=inputs.drop(['qty'],axis=1)
    a=kc1.to_numpy().tolist()
    beta=params
    lambda0=beta[8]
    beta=np.delete(beta, [8])
    lambdai=0
    sum2=0
    sum1=0
    for i in range(len(a)):
        sum1=0
        for j in range(len(a[i])):
            sum1=sum1+beta[j]*a[i][j]
        lambdai=lambda0*np.exp(sum1)
        sum2=sum2+poisson_pmf(k[i],lambdai)
    return sum2
```

In [769]:
```
def NLL(params, inputs):
    return(-1*(LL(params, inputs)))
```

```
In [770]: final=minimize(NLL,
                  args=inputs,
                  x0=params,
                  bounds=[(0.000001, None),(0.000001, None),(0.000001, None),(0.000001, None),(0.000001, None),(0.000001, None),(0.000001,
```

```
[0.66666667 0.66666667 0.66666667 0.66666667 0.66666667 0.66666667
 0.66666667 0.66666667 0.66666667]
[0.65314381 0.65321818 0.65338741 0.65294012 0.65282299 0.65287473
 0.65316911 0.65312421 0.65163125]
[0.62711426 0.62738249 0.62791008 0.62656962 0.62616393 0.62627598
 0.6272005  0.62705196 0.62213998]
[0.60600077 0.60647869 0.60732939 0.60524451 0.60453898 0.60465152
 0.60614687 0.60589678 0.59764191]
[0.58291482 0.58367984 0.5849187  0.58201455 0.58090744 0.58095622
 0.58313458 0.58275443 0.57020062]
[0.56049853 0.56160369 0.56325652 0.55956759 0.55799027 0.55789686
 0.56079472 0.56026805 0.54282981]
[0.53772315 0.53923933 0.54135332 0.53689691 0.53475304 0.53441458
 0.53809739 0.5373999  0.51419944]
[0.51498783 0.5169833  0.51960256 0.51443213 0.51162432 0.51091691
 0.51543435 0.51454227 0.48468518]
[0.4921426  0.49469203 0.49786969 0.49206084 0.48847576 0.48724393
 0.492648   0.49153351 0.45395065]
[0.46925605 0.47243571 0.47623088 0.46989456 0.46540608 0.4634589
```

```
In [771]: print("The Value for lambda is",final.x[8])
          print("The values for the beta parameters are",final.x[0:8])
          print("The Value of Maximum Log likelihood is ",-1*final.fun)
```

```
The Value for lambda is 0.553149828274452
The values for the beta parameters are [1.00000000e-06 1.00000000e-06 1.00000000e-06 2.58948978e-02
 1.60185434e-02 6.00522527e-02 1.00000000e-06 1.00000000e-06]
The Value of Maximum Log likelihood is  -18884.638464333206
```

lambda	**0.553149828274452**
Beta0 , education	**1.00000000e-06**
Beta1, region	**1.00000000e-06**
Beta2, hhsz	**1.00000000e-06**
Beta3, age	**2.58948978e-02**
Beta4, income	**1.60185434e-02**
Beta5,child	**6.00522527e-02**
Beta6, race	**1.00000000e-06**
Beta7, country	**1.00000000e-06**
Maximum Log likelihood	**-18884.638464333206**

- Targeted marketing campaigns can be used for children, age, income segment of customers.
- From the management view we would target children. The highest value for Beta5 is significant enough to infer that households having children, purchased more books.
- Also, high income users can be targeted by providing them discounts on the most purchased books. And users with high age can also be targeted.

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?

In [772]:
```
params = np.array([0.01,0.01,0.1,0.01,0.01,0.01,0.01,0.01,0.01,0.01])
```

In [773]:
```
pmf_nbd=lambda a,n,k,t: (np.log(gamma(n+k))-(np.log(gamma(n))+np.log(factorial(k))))+(n*(np.log(a)-np.log(a+t)))+(k*(np.log(t)-np
```

In [774]:
```
inputs=bk3.copy()
inputs
```

Out[774]:

	education	region	hhsz	age	income	child	race	country	qty
1	2.00000	4.0	1	6.0	5	0	1	0	0
2	5.00000	4.0	4	5.0	6	0	1	0	0
3	2.74931	2.0	2	6.0	3	0	1	0	0
4	4.00000	4.0	5	4.0	7	1	1	0	0
5	5.00000	3.0	1	10.0	3	0	1	1	0
...	...	...	...	...	...	...	...	...	...
1807	1.00000	2.0	5	10.0	2	1	1	0	5
1808	2.74931	3.0	4	8.0	4	1	1	0	2
1809	2.74931	3.0	4	4.0	4	1	1	0	9
1810	2.74931	4.0	6	8.0	6	1	1	0	2
1811	2.74931	4.0	1	11.0	2	0	1	0	1

9439 rows × 9 columns

In [775]:
```
def NBDLL(params,inputs):
    k=inputs['qty'].tolist()
    kc1=inputs.drop(['qty'],axis=1)
    a=kc1.to_numpy().tolist()
    beta=params
    alpha=beta[9]
    n=beta[8]
    beta=np.delete(beta, [9])
    beta=np.delete(beta, [8])
    lambdai=0
    sum2=0
    sum1=0
    for i in range(len(a)):
        sum1=0
        for j in range(len(a[i])):
            sum1=sum1+beta[j]*a[i][j]
        t=math.exp(sum1)
        sum2=sum2+pmf_nbd(alpha,n,k[i],t)
    return sum2
```

In [776]:
```
def NegLL(params, inputs):
    return(-(NBDLL(params, inputs)))
```

In [777]:
```
final=minimize(NegLL,
         args=inputs,
         x0=params,
         bounds=[(None, None), (None, None), (None, None), (None, None),(None, None), (None, None), (None, None), (None, None),(
```

```
In [778]: print("The Value for n is ",final.x[8]," and Alpha is ",final.x[9])
          print("The values for the beta parameters are",final.x[0:8])
          print("The Value of Maximum Log likelihood is ",-1*final.fun)

          The Value for n is  0.09835769568046158  and Alpha is  0.0785426592415001
          The values for the beta parameters are [-0.12936605 -0.09999939 -0.0030107   0.02986125  0.01733193  0.05690244
           -0.21812761 -0.06818342]
          The Value of Maximum Log likelihood is  -8355.151876290514
```

n	**0.09835769568046158**
Alpha	**0.0785426592415001**
Beta0 , education	**-0.12936605**
Beta1, region	**-0.09999939**
Beta2, hhsz	**-0.0030107**
Beta3, age	**0.02986125**
Beta4, income	**0.01733193**
Beta5, child	**0.05690244**
Beta6, race	**-0.21812761**
Beta7, country	**-0.06818342**
Maximum Log likelihood	**-8355.151876290514**

- Targeted marketing campaigns can be used for children, age, income segment of customers.
- From the management view we would target children. The highest value for Beta5 is significant enough to infer that households having children, purchased more books.
- Also, high income users can be targeted by providing them discounts on the most purchased books. And users with high age can also be targeted.

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?

Please refer to the Python file for the complete code

Below is a snippet of the code:

```
def NBDLL(params,inputs):
    k=inputs['qty'].tolist()
    kc1=inputs.drop(['qty'],axis=1)
    a=kc1.to_numpy().tolist()
    beta=params
    alpha=beta[9]
    n=beta[8]
    beta=np.delete(beta, [9])
    beta=np.delete(beta, [8])
    lambdai=0
    sum2=0
    sum1=0
    for i in range(len(a)):
        sum1=0
        for j in range(len(a[i])):
            sum1=sum1+beta[j]*a[i][j]
        t=math.exp(sum1)
        sum2=sum2+pmf_nbd(alpha,n,k[i],t)
    return sum2
```

```
def NegLL(params, inputs):
    return(-(NBDLL(params, inputs)))
```

```
final=minimize(NegLL,
         args=inputs,
         x0=params,
         bounds=[(None, None), (None, None), (None, None), (None, None),(None, None), (None, None), (None, None), (None, None
```

```
print("The Value for n is ",final.x[8]," and Alpha is ",final.x[9])
print("The values for the beta parameters are",final.x[0:8])
print("The Value of Maximum Log likelihood is ",-1*final.fun)
```

```
The Value for n is  0.09835769568046158  and Alpha is  0.0785426592415001
The values for the beta parameters are [-0.12936605 -0.09999939 -0.0030107   0.02986125  0.01733193  0.05690244
 -0.21812761 -0.06818342]
The Value of Maximum Log likelihood is  -8355.151876290514
```

- Below are the parameters, Max log-likelihood, and the beta from the NBD regression. The age, income and child variables are more relevant in deciding the quantity of books purchased. **The most relevant customer characteristics are age, income, and child variables**

N	0.09835769568046158
Alpha	0.0785426592415001
Maximum Log Likelihood	−8355.151876290514
Beta0 , education	−0.12936605
Beta1, region	−0.09999939
Beta2, hhsz	−0.0030107
Beta3, age	0.02986125
Beta4, income	0.01733193
Beta5, child	0.05690244
Beta6, race	−0.21812761
Beta7, country	−0.06818342

- Targeted marketing campaigns can be used for higher income, age, more children segment of customers. Bundle discounts can be used to club most sold books in both the age categories for higher sales.
- From the management view we would target older age group. The highest value for Beta5 is significant enough to infer that households having children, purchased more books.

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.

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.

Below are our findings from all the models, Log Likelihood, AIC, BIC and LL ratio.

Part	Q No.	Model Name	Dataset	Log Likelihood (LL)	K	n	AIC	BIC	LL Ratio
1	3	Poissons Regression	Khaki Chinos	-6291.49675	4	2728	12590.99	12596.74	
1	4	NBD Regression	Khaki Chinos	-2888.96615	4	2728	5785.932	5791.676	
2	2	Poisson Model	Books01	-18921.9184	1	46	37845.84	37845.5	
2	3	Poisson Model- w/o independent variables	Books02	-18921.9184	1	9451	37845.84	37847.81	
2	4	NBD Model	Books01	-8603.49885	1	46	17209	17208.66	
2	5	NBD Model- w/o independent variables	Books02	-8603.49885	1	9451	17209	17210.97	
2	8	Poisson Regression with Imputations & all Variables	Books02	-18884.6385	8	9439	37785.28	37801.08	74.56
2	9	NBD Regression with Imputations & all Variables	Books02	-8355.15188	8	9439	16726.3	16742.1	496.7

Below are our interpretations:

- While both models are based on the same dataset, NBD performs better with the higher log likelihood.
- Lower AIC and BIC values makes NBD a better fit for the dataset then Poisson.
- When we look at Poisson model for books01 and books02 – The log likelihood is same because there are no other independent variables. Typically, by adding more variables to the dataset the log likelihood is supposed to increase even if there are less significant variables that hamper the effective prediction of the model.
- The same behaviour is expected in NBD for Books01 and Books02. But the Log Likelihood is higher for NBD which makes it a better model. (Even without independent variables). Even with AIC, NBD is almost half of that of Poisson's
- Log Likelihood ratio is used to compare models with different number of columns. NBD nulls and NBD with all variables (and Imputations) gives a better ratio.

- **These results are consistent with part1 datasets and recommendations stands the same that NBD is a better fit. For all the datasets, NBD gives higher log likelihood.**
- We would choose model 9 as the best model so far. It considers other customers characteristics and shows that age, income, and children constitute buying patterns. NBD without independent variables is also a good model. The only difference between these two models is the number of independent variables. Having more variables will naturally increases the likelihood. And
- But it does not make informed decision by dependency on other variables.
- **The NBD model fits the data better since it includes the heterogeneity of customer demographics. Additionally, Poisson assumes that the mean of the distribution is very close to the variance which is only ideal. This might be causing the difference we see in the models.**

Learnings from the project-

- The biggest learning from the project is the comparative study btw NBD and Poisson.
- Mathematically we understood how NBD incorporates heterogeneity and performs well with all datasets.
- We also learnt that more number of variables added in the dataset, will increase the log likelihood.
- The NBD model without independent variables and NBD model with independent variables performs better compared to poisons models and, in that, NBD model with independent variables stands out with higher log Likelihood and lower AIC, BIC and higher log likelihood ratio.
- Nbd is better because it incorporates sum of probabilities. The estimation is accurate because, the probability is dependent on the previous probabilities. Hence, NBD gives better predictions
- As a management we would be interested in looking at the demographics for understanding the customer better. It can be used for targeted marketing. We have a clear understanding now that, older age group, many children and high-income household are effective drivers for purchase of quantity of books.
- We can set of discount campaigns and send flyers to targeted customers.