In [1]:
```
import pandas as pd
import numpy as np
import math
import matplotlib.pyplot as plt
import matplotlib
import seaborn as sns
import scipy as sp
import statsmodels as sm
```

In [2]:
```
data_df=pd.read_csv('DATA - 3.csv')
```

In [3]:
```
data_df.head()
```

Out[3]:

	participantID	age	nativeLanguage	gender	education	city	country	responseID	sectio
0	12	28	URU_R	Fe	4	Montevideo	Uruguay	128	set_201
1	12	28	URU_R	Fe	4	Montevideo	Uruguay	129	set_201
2	12	28	URU_R	Fe	4	Montevideo	Uruguay	130	set_201
3	12	28	URU_R	Fe	4	Montevideo	Uruguay	131	set_201
4	12	28	URU_R	Fe	4	Montevideo	Uruguay	132	set_201

In [4]:
```
type(data_df)
```

Out[4]:
```
pandas.core.frame.DataFrame
```

In [5]:
```
data_df.dtypes
```

Out[5]:
```
participantID      int64
age                int64
nativeLanguage    object
gender            object
education          int64
city              object
country           object
responseID         int64
section           object
cue               object
R1                object
R2                object
R3                object
dtype: object
```

In [6]:
```
data_df.describe()
```

Out[6]:

	participantID	age	education	responseID
count	558503.000000	558503.000000	558503.000000	558503.000000
mean	21075.098390	37.796812	3.651834	280727.388893
std	12283.948985	15.118828	0.675921	161398.704512
min	12.000000	5.000000	1.000000	128.000000
25%	10513.000000	25.000000	3.000000	141213.500000
50%	20889.000000	35.000000	4.000000	280839.000000
75%	31387.000000	49.000000	4.000000	420464.500000
max	43297.000000	99.000000	5.000000	560428.000000

In [8]:
```
data_df.nunique()
```

Out[8]:
```
participantID    31029
age                 87
nativeLanguage       4
gender               3
education            5
city               716
country             64
responseID      558503
section              9
cue               9122
R1               60762
R2               55479
R3               50803
dtype: int64
```

In [7]:
```
# looking at unique values fo each categorical columns
```

In [9]:
```
data_df.nativeLanguage.unique()
```

Out[9]:
```
array(['URU_R', 'ARG_C', nan, 'ARG_R', 'ARG_N'], dtype=object)
```

In [10]:
```
data_df.gender.unique()
```

25%	20...	25.000000	3.000000	141213.500000
50%	29889.000000	35.000000	4.000000	280839.000000
75%	31387.000000	49.000000	4.000000	420464.500000
max	43297.000000	99.000000	5.000000	560428.000000

In [7]:
```
# looking at unique values fo each categorical columns
```

In [8]:
```
data_df.nunique()
```

Out[8]:
```
participantID      31023
age                   87
nativeLanguage         4
gender                 3
education              5
city                 716
country               64
responseID        558503
section                9
cue                 9122
R1                 60762
R2                 55479
R3                 50803
dtype: int64
```

In [9]:
```
data_df.nativeLanguage.unique()
```

Out[9]:
```
array(['URU_R', 'ARG_C', nan, 'ARG_R', 'ARG_N'], dtype=object)
```

In [10]:
```
data_df.gender.unique()
```

Out[10]:
```
array(['Fe', 'Ma', 'X'], dtype=object)
```

In [11]:
```
data_df.city.unique()
```

Out[11]:
```
array(['Montevideo', nan, 'Cachan', 'Paris', 'Mendoza', 'notfound',
       'Buenos Aires', 'Cambridge', 'Santiago', 'Bures-sur-yvette',
       'Moza', 'Mezel', 'La Paz', 'Federal', 'Lascano', 'Bridgewater',
       'Los Cerrillos', 'Pforzheim', 'Maipú', 'Tallahassee',
       'Las Piedras', 'Bayona', 'Minas', 'Tucumán', 'Drummoyne',
       'Tel Aviv', 'Huancavelica', 'Palermo', 'La Habana', 'Ames',
       'Mountain View', 'Moreno', 'Rochester', 'Vigo', 'Munro', 'Rosario',
       'Ituzaingó', 'New York', 'Dietmannsried', 'Berlin',
       'Pietermaritzburg', 'Ripollet', 'Córdoba', 'Opera', 'Partille',
       'Rio De Janeiro', 'Carrasco', 'Málaga', 'Aguas Corrientes',
       'Lanús', 'Montreal', 'São Paulo', 'Jujuy', 'Requena', 'Washington',
       'Salem', 'Gadsden', 'Bronx', 'Sevilla', 'Tel Mond',
       'Santa Coloma De Cervelló', 'Pilar', 'Montalieu', 'Paterna',
       'Philadelphia', 'Barcelona', 'Toronto', 'Niterói', 'Auckland',
       'East Rockaway', 'Valladolid', 'Madrid', 'Lima', 'Mexico',
       'Evanston', 'Chambly', 'Hospital', 'Ithaca', 'Belo Horizonte',
       'Velbert', 'Wellsford', 'Canelones', 'Herzliya', 'La Plata',
       'Valparaíso', 'Asunción', 'Bremen', 'Florida', 'Wilmington',
       'Kennesaw', 'Minneapolis', 'Yverdon', 'London', 'Noranda',
```

In [12]:
```
data_df.country.unique()
```

Out[12]:
```
array(['Uruguay', 'France', 'Argentina', 'notfound', 'United Kingdom',
       'Chile', 'Israel', 'United States', 'Germany', 'Spain',
       'Australia', 'Brazil', 'Peru', 'Cuba', 'South Africa', 'Italy',
       'Sweden', 'Canada', 'Mexico', 'Switzerland', 'Colombia', 'Austria',
       'New Zealand', 'Guatemala', 'Ireland', 'Paraguay', 'Panama',
       'Venezuela', 'Ecuador', 'Belgium', 'Costa Rica', 'España',
       'Brasil', 'República Federativa do Brasil', nan, 'Honduras',
       'Nueva Zelanda', 'México', 'Bolivia', 'Alemania', 'Singapore',
       'Portugal', 'Perú', 'Nicaragua', 'Danmark', 'Albania', '99',
       'Estados Unidos', 'Italia', 'Suiza', 'Netherlands', 'El Salvador',
       'Norway', 'Greece', 'Czechia', 'Poland', 'Taiwan', 'Finland',
       'Cambodia', 'Romania', 'Denmark', 'Iceland', 'Dominican Republic',
       'Hungary', 'Aruba'], dtype=object)
```

In [16]:
```
data_df.R2.unique()
```

Out[16]:
```
array(['cerveza', 'nocturno', 'derecha', ..., 'refutado', 'vivificante',
       'susceptible'], dtype=object)
```

In [17]:
```
data_df.R3.unique()
```

Out[17]:
```
array(['noche', 'baja', 'hermano', ..., 'Mirtha', 'funwbrero', 'Troya'],
      dtype=object)
```

In [13]:
```
data_df.section.unique()
```

Out[13]:
```
array(['set_2013', 'set_2014', 'set_2018', 'set_2019', 'set_2020',
       'set_2021', 'set_2022', 'set_2023', 'set_2024'], dtype=object)
```

In [18]:
```
# copying the dataframe so as to keep the original dataframe unchanged
```

In [14]:
```
data_df.cue.unique()
```

Out[14]:
```
array(['bar', 'tren', 'mano', ..., 'cargado', 'sacado', 'inocuo'],
      dtype=object)
```

In [19]:
```
df=data_df.copy()
```

In [20]:
```
df
```

In [15]:
```
data_df.R1.unique()
```

Out[15]:
```
array(['abierto', 'expreso', 'libre', ..., 'rebotado', 'anexiones',
       'asentimiento'], dtype=object)
```

Out[20]:

	participantID	age	nativeLanguage	gender	education	city	country	responseID	s
0	12	28	URU_R	Fe	4	Montevideo	Uruguay	128	s
1	12	28	URU_R	Fe	4	Montevideo	Uruguay	129	s
2	12	28	URU_R	Fe	4	Montevideo	Uruguay	130	s

In [16]:
```
data_df.R2.unique()
```

Out[16]:
```
array(['Nueva Zelanda', 'México', 'Bolivia', 'Alemania', 'Singapore',
       'Portugal', 'Perú', 'Nicaragua', 'Danmark', 'Albania', '99',
       'Estados Unidos', 'Italia', 'Suiza', 'Netherlands', 'El Salvador',
       'Norway', 'Greece', 'Czechia', 'Poland', 'Taiwan', 'Finland',
       'Cambodia', 'Romania', 'Denmark', 'Iceland', 'Dominican Republic',
       'Hungary', 'Aruba'], dtype=object)
```

In [17]:
```
data_df.R3.unique()
```

Out[17]:
```
array(['noche', 'baia', 'hermano', ..., 'Mirtha', 'funwbrero', 'Troya'],
      dtype=object)
```

In [13]:
```
data_df.section.unique()
```

Out[13]:
```
array(['set_2015', 'set_2014', 'set_2018', 'set_2019', 'set_2020',
       'set_2021', 'set_2022', 'set_2023', 'set_2024'], dtype=object)
```

In [18]:
```
# copying the dataframe so as to keep the original dataframe unchanged
```

In [14]:
```
data_df.cue.unique()
```

In [19]:
```
df=data_df.copy()
```

Out[14]:
```
array(['bar', 'tren', 'mano', ..., 'cargado', 'sacado', 'inocuo'],
      dtype=object)
```

In [20]:
```
df
```

Out[20]:

	participantID	age	nativeLanguage	gender	education	city	country	responseID	s
0	12	28	URU_R	Fe	4	Montevideo	Uruguay	128	
1	12	28	URU_R	Fe	4	Montevideo	Uruguay	129	
2	12	28	URU_R	Fe	4	Montevideo	Uruguay	130	
3	12	28	URU_R	Fe	4	Montevideo	Uruguay	131	
4	12	28	URU_R	Fe	4	Montevideo	Uruguay	132	
...	...	...	...	...	...	...	...	...	
558498	43296	33	ARG_R	Fe	3	Hurlingham	Argentina	560050	
558499	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560015	
558500	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560016	
558501	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560017	
558502	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560018	

558503 rows × 13 columns

In [16]:
```
data_df.R1.unique()
```

Out[16]:
```
array(['abierto', 'expreso', 'libre', ..., 'rebotado', 'anexionar',
       'asentimiento'], dtype=object)
```

In [21]:
```
# converting categorical data into numerical data
```

In [22]:
```
df.nativeLanguage.value_counts()
```

Out[22]:
```
URU_R    348890
ARG_R    177970
ARG_C      5922
ARG_N      3132
Name: nativeLanguage, dtype: int64
```

In [23]:
```
df['nativeLanguage_num']=pd.factorize(df.nativeLanguage)[0]
```

In [24]:
```
df
```

Out[24]:

	participantID	age	nativeLanguage	gender	education	city	country	responseID	s
0	12	28	URU_R	Fe	4	Montevideo	Uruguay	128	
1	12	28	URU_R	Fe	4	Montevideo	Uruguay	129	
2	12	28	URU_R	Fe	4	Montevideo	Uruguay	130	
3	12	28	URU_R	Fe	4	Montevideo	Uruguay	131	
4	12	28	URU_R	Fe	4	Montevideo	Uruguay	132	
...	...	...	...	...	...	...	...	...	
558498	43296	33	ARG_R	Fe	3	Hurlingham	Argentina	560050	
558499	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560015	
558500	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560016	
558501	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560017	
558502	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560018	

In [24]:
```
df
```

Out[24]:

	participantID	age	nativeLanguage	gender	education	city	country	responseID	s
0	12	28	URU_R	Fe	4	Montevideo	Uruguay	128	s
1	12	28	URU_R	Fe	4	Montevideo	Uruguay	129	s
2	12	28	URU_R	Fe	4	Montevideo	Uruguay	130	s
3	12	28	URU_R	Fe	4	Montevideo	Uruguay	131	s
4	12	28	URU_R	Fe	4	Montevideo	Uruguay	132	s
...	...	...	...	...	...	...	...	...	
558498	43296	33	ARG_R	Fe	3	Hurlingham	Argentina	560050	s
558499	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560015	s
558500	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560016	s
558501	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560017	s
558502	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560018	s

558503 rows × 14 columns

In [25]:
```
df['gender_num']=pd.factorize(df.gender)[0]
```

In [26]:
```
df
```

Out[26]:

	participantID	age	nativeLanguage	gender	education	city	country	responseID	s
0	12	28	URU_R	Fe	4	Montevideo	Uruguay	128	s
1	12	28	URU_R	Fe	4	Montevideo	Uruguay	129	s
2	12	28	URU_R	Fe	4	Montevideo	Uruguay	130	s
3	12	28	URU_R	Fe	4	Montevideo	Uruguay	131	s
4	12	28	URU_R	Fe	4	Montevideo	Uruguay	132	s
...	...	...	...	...	...	...	...	...	
558498	43296	33	ARG_R	Fe	3	Hurlingham	Argentina	560050	s
558499	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560015	s
558500	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560016	s
558501	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560017	s
558502	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560018	s

558503 rows × 15 columns

In [28]:
```
df
```
In [27]:
```
df['city_num']=pd.factorize(df.city)[0]
```

Out[28]:

	participantID	age	nativeLanguage	gender	education	city	country	responseID	s
0	12	28	URU_R	Fe	4	Montevideo	Uruguay	128	s
1	12	28	URU_R	Fe	4	Montevideo	Uruguay	129	s
2	12	28	URU_R	Fe	4	Montevideo	Uruguay	130	s
3	12	28	URU_R	Fe	4	Montevideo	Uruguay	131	s
4	12	28	URU_R	Fe	4	Montevideo	Uruguay	132	s
...	...	...	...	...	...	...	...	...	
558498	43296	33	ARG_R	Fe	3	Hurlingham	Argentina	560050	s
558499	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560015	s
558500	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560016	s
558501	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560017	s
558502	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560018	s

In [28]:
In [27]:
```
df
df['city_num']=pd.factorize(df.city)[0]
```

Out[28]:

	participantID	age	nativeLanguage	gender	education	city	country	responseID	
0	12	28	URU_R	Fe	4	Montevideo	Uruguay	128	s
1	12	28	URU_R	Fe	4	Montevideo	Uruguay	129	s
2	12	28	URU_R	Fe	4	Montevideo	Uruguay	130	s
3	12	28	URU_R	Fe	4	Montevideo	Uruguay	131	s
4	12	28	URU_R	Fe	4	Montevideo	Uruguay	132	s
...	...	...	...	...	...	...	...	...	
558498	43296	33	ARG_R	Fe	3	Hurlingham	Argentina	560050	s
558499	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560015	s
558500	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560016	s
558501	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560017	s
558502	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560018	s

558503 rows × 16 columns

In [29]:
```
df['country_num']=pd.factorize(df.country)[0]
```

In [30]:
```
df
```

Out[30]:

	participantID	age	nativeLanguage	gender	education	city	country	responseID	
0	12	28	URU_R	Fe	4	Montevideo	Uruguay	128	s
1	12	28	URU_R	Fe	4	Montevideo	Uruguay	129	s
2	12	28	URU_R	Fe	4	Montevideo	Uruguay	130	s
3	12	28	URU_R	Fe	4	Montevideo	Uruguay	131	s
4	12	28	URU_R	Fe	4	Montevideo	Uruguay	132	s
...	...	...	...	...	...	...	...	...	
558498	43296	33	ARG_R	Fe	3	Hurlingham	Argentina	560050	s
558499	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560015	s
558500	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560016	s
558501	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560017	s
558502	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560018	s

558503 rows × 17 columns

In [32]:
In [31]:
```
df
df['section_num']=pd.factorize(df.section)[0]
```

Out[32]:

	participantID	age	nativeLanguage	gender	education	city	country	responseID	
0	12	28	URU_R	Fe	4	Montevideo	Uruguay	128	s
1	12	28	URU_R	Fe	4	Montevideo	Uruguay	129	s
2	12	28	URU_R	Fe	4	Montevideo	Uruguay	130	s
3	12	28	URU_R	Fe	4	Montevideo	Uruguay	131	s
4	12	28	URU_R	Fe	4	Montevideo	Uruguay	132	s
...	...	...	...	...	...	...	...	...	
558498	43296	33	ARG_R	Fe	3	Hurlingham	Argentina	560050	s
558499	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560015	s
558500	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560016	s
558501	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560017	s
558502	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560018	s

In [32]: df

In [31]:
```
df['section_num']=pd.factorize(df.section)[0]
```

Out[32]:

	participantID	age	nativeLanguage	gender	education	city	country	responseID	s
0	12	28	URU_R	Fe	4	Montevideo	Uruguay	128	s
1	12	28	URU_R	Fe	4	Montevideo	Uruguay	129	s
2	12	28	URU_R	Fe	4	Montevideo	Uruguay	130	s
3	12	28	URU_R	Fe	4	Montevideo	Uruguay	131	s
4	12	28	URU_R	Fe	4	Montevideo	Uruguay	132	s
...	...	...	...	...	...	...	...	...	
558498	43296	33	ARG_R	Fe	3	Hurlingham	Argentina	560050	s
558499	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560015	s
558500	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560016	s
558501	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560017	s
558502	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560018	s

558503 rows × 18 columns

In [33]:
```
df['cue_num']=pd.factorize(df.cue)[0]
```

In [34]:
```
df
```

Out[34]:

	participantID	age	nativeLanguage	gender	education	city	country	responseID	s
0	12	28	URU_R	Fe	4	Montevideo	Uruguay	128	s
1	12	28	URU_R	Fe	4	Montevideo	Uruguay	129	s
2	12	28	URU_R	Fe	4	Montevideo	Uruguay	130	s
3	12	28	URU_R	Fe	4	Montevideo	Uruguay	131	s
4	12	28	URU_R	Fe	4	Montevideo	Uruguay	132	s
...	...	...	...	...	...	...	...	...	
558498	43296	33	ARG_R	Fe	3	Hurlingham	Argentina	560050	s
558499	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560015	s
558500	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560016	s
558501	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560017	s
558502	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560018	s

558503 rows × 19 columns

In [35]:
```
df['R1_num']=pd.factorize(df.R1)[0]
df['R2_num']=pd.factorize(df.R2)[0]
df['R3_num']=pd.factorize(df.R3)[0]
df
```

Out[35]:

	participantID	age	nativeLanguage	gender	education	city	country	responseID	s
0	12	28	URU_R	Fe	4	Montevideo	Uruguay	128	s
1	12	28	URU_R	Fe	4	Montevideo	Uruguay	129	s
2	12	28	URU_R	Fe	4	Montevideo	Uruguay	130	s
3	12	28	URU_R	Fe	4	Montevideo	Uruguay	131	s
4	12	28	URU_R	Fe	4	Montevideo	Uruguay	132	s
...	...	...	...	...	...	...	...	...	
558498	43296	33	ARG_R	Fe	3	Hurlingham	Argentina	560050	s
558499	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560015	s
558500	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560016	s
558501	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560017	s

In [35]:
```
df['R1_num']=pd.factorize(df.R1)[0]
df['R2_num']=pd.factorize(df.R2)[0]
df['R3_num']=pd.factorize(df.R3)[0]
df
```

Out[35]:

	participantID	age	nativeLanguage	gender	education	city	country	responseID	s
0	12	28	URU_R	Fe	4	Montevideo	Uruguay	128	s
1	12	28	URU_R	Fe	4	Montevideo	Uruguay	129	s
2	12	28	URU_R	Fe	4	Montevideo	Uruguay	130	s
3	12	28	URU_R	Fe	4	Montevideo	Uruguay	131	s
4	12	28	URU_R	Fe	4	Montevideo	Uruguay	132	s
...	...	...	...	...	...	...	...	...	
558498	43296	33	ARG_R	Fe	3	Hurlingham	Argentina	560050	s
558499	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560015	s
558500	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560016	s
558501	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560017	s
558502	43297	60	ARG_R	Fe	4	Buenos Aires	Argentina	560018	s

558503 rows × 22 columns

In [36]:
```
df.dtypes
```

Out[36]:
```
participantID          int64
age                    int64
nativeLanguage        object
gender                object
education              int64
city                  object
country               object
responseID             int64
section               object
cue                   object
R1                    object
R2                    object
R3                    object
nativeLanguage_num     int64
gender_num             int64
city_num               int64
country_num            int64
section_num            int64
cue_num                int64
R1_num                 int64
R2_num                 int64
R3_num                 int64
dtype: object
```

In [37]:
```
# taking only the numerical column for further task
```

In [39]:
```
df1
```

Out[38]:

In [38]:
```
df1=df.select_dtypes(include=['int64'])
```

	participantID	age	education	responseID	nativeLanguage_num	gender_num	city_num	co
0	12	28	4	128	0	0	0	
1	12	28	4	129	0	0	0	
2	12	28	4	130	0	0	0	
3	12	28	4	131	0	0	0	
4	12	28	4	132	0	0	0	
...	...	...	...	...	...	...	...	
558498	43296	33	3	560050	2	0	305	
558499	43297	60	4	560015	2	0	5	
558500	43297	60	4	560016	2	0	5	
558501	43297	60	4	560017	2	0	5	
558502	43297	60	4	560018	2	0	5	

558503 rows × 13 columns

In [37]:
```
# taking only the numerical column for further task
df1
```

In [39]:

Out[38]:

In [38]:
```
df1=df.drop(['participantID','age','education','responseID','nativeLanguage_num','gender_num','city_num','country_num','section_num','cue_num','R1_num','R2_num','R3_num','age'],axis=1)
```

	participantID	education	responseID	nativeLanguage_num	gender_num	city_num	co
0	12	4	128	0	0	0	
1	12	4	129	0	0	0	
2	12	4	130	0	0	0	
3	12	4	131	0	0	0	
4	12	4	132	0	0	0	
...	...	...	...	...	...	...	
558498	43296	3	560050	2	0	305	
558499	43297	4	560015	2	0	5	
558500	43297	4	560016	2	0	5	
558501	43297	4	560017	2	0	5	
558502	43297	4	560018	2	0	5	

558503 rows × 13 columns

In [40]:
```
# Using pop() method
df1['age'] = df1.pop('age')
```

In [41]:
```
df1
```

Out[41]:

	participantID	education	responseID	nativeLanguage_num	gender_num	city_num	country_
0	12	4	128	0	0	0	
1	12	4	129	0	0	0	
2	12	4	130	0	0	0	
3	12	4	131	0	0	0	
4	12	4	132	0	0	0	
...	...	...	...	...	...	...	
558498	43296	3	560050	2	0	305	
558499	43297	4	560015	2	0	5	
558500	43297	4	560016	2	0	5	
558501	43297	4	560017	2	0	5	
558502	43297	4	560018	2	0	5	

558503 rows × 13 columns

In [42]:
```
df1.nunique()
```

Out[42]:
```
participantID          31029
education                  5
responseID            558503
nativeLanguage_num         5
gender_num                 3
city_num                 717
country_num               65
section_num                9
cue_num                 9122
R1_num                 60763
R2_num                 55480
R3_num                 50804
age                       87
dtype: int64
```

In [44]:
```
import sklearn
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

In [45]:
```
# splitting data into training 75% and testing 25%
```

In [46]:
```
X=df1.iloc[:,:-1]
y=df1.iloc[:,-1]
X_train,X_test, y_train, y_test=train_test_split(X,y,test_size=0.25, random_state
```

In [47]:
```
X_train
```

In [43]:
```
# same as data_df
```

Out[47]:

	participantID	education	responseID	nativeLanguage_num	gender_num	city_num	country_
463314	35006	4	464938	2	0	670	
68490	5114	3	68708	0	0	0	
70470	5287	4	70688	0	0	0	
63487	4717	4	63705	0	1	0	
185111	14082	4	186645	0	0	0	

Splitting the data into training and testing

```
responseID             558565
nativeLanguage_num          5
gender_num                  3
city_num                  717
country_num                65
section_num                 9
cue_num                  9122
R1_num                  60763
R2_num                  55480
R3_num                  50804
age                        87
dtype: int64
```

In [43]:

```
# same as data_df
```

Splitting the data into training and testing

In [44]:

```
import sklearn
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

In [45]:

```
#splitting data into training 75% and testing 25%
```

In [46]:

```
X=df.iloc[:,:-1]
y=df.iloc[:,-1]
X_train,X_test, y_train, y_test=train_test_split(X,y,test_size=0.25, random_stat
```

In [47]:

```
X_train
```

Out[47]:

	participantID	education	responseID	nativeLanguage_num	gender_num	city_num	country_
463314	35006	4	464938	2	0	670	
66490	5114	3	66708	0	[illegible]	0	
70470	5287	4	70688	0	0	0	
63487	4717	4	63705	0	1	0	
185111	14082	4	186645	0	0	0	
...	...	...	...	...	...	...	
359783	26785	4	361407	0	0	-1	
152315	11465	3	153849	0	0	0	
117952	8913	3	119540	0	0	-1	
435829	32751	4	437399	0	0	0	
305711	22834	4	307281	0	0	0	

418877 rows × 12 columns

In [48]:

```
X_test
```

Out[48]:

	participantID	education	responseID	nativeLanguage_num	gender_num	city_num	country_
218195	16488	3	219783	0	0	-1	
509828	39049	3	511416	0	0	-1	
475113	36086	3	476701	2	0	5	
321135	23926	3	322759	0	0	0	
202457	15321	3	204045	0	0	0	
...	...	...	...	...	...	...	
271031	20276	4	272655	0	0	-1	
491343	37409	4	492931	2	0	-1	
31614	2273	4	31767	0	1	0	
86364	6602	4	86582	2	1	-1	
299781	22371	4	301387	2	1	177	

139626 rows × 12 columns

In [50]:

```
y_test
```

Out[50]:

```
218195    21
509828    47
475113    20
321135    [illegible]
202457    [illegible]
          ...
271031    [illegible]
491343    [illegible]
31614     36
86364     [illegible]
299781    [illegible]
Name: age, Length: 139626, dtype: int64
```

In [49]:

```
y_train
```

Out[49]:

```
463314    [illegible]
66490     [illegible]
70470     21
63487     [illegible]
185111    [illegible]
          ...
359783    [illegible]
152315    [illegible]
117952    21
435829    47
305711    22
Name: age, Length: 418877, dtype: int64
```

Training and evaluating the model's performance

In [51]:

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, mean_
```

In [50]:
```
y_test
```

Out[50]:
```
139626 rows × 12 columns
218195    21
500828    47
475113    20
[illegible]
70470     21
[illegible]
31614     36
[illegible]
Name: age, Length: 139626, dtype: int64
```

In [49]:
```
y_train
```

Out[49]:
```
[illegible]
435829    47
305711    22
Name: age, Length: 418877, dtype: int64
```

Training and evaluating the model's performance

In [51]:
```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, mean_s
```

In [52]:
```
# selecting linear regression model and training
```

In [53]:
```
ler=LinearRegression()
```

In [54]:
```
ler.fit(X_train,y_train)
```

Out[54]:
```
LinearRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [55]:
```
# making predictions
```

In [56]:
```
predictions=ler.predict(X_test)
```

In [57]:
```
predictions
```

Out[57]:
```
array([34.26249512, 43.41110539, 44.40754254, ..., 29.61130952,
       31.60389675, 37.3875704 ])
```

In [58]:
```
# Evaluating the model
```

In [59]:
```
# accuracy, precision, recall and RMSE
```

In [60]:
```
predictions=predictions.astype(int)
```

In [61]:
```
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
```

Fine tuning the model

In [62]:
```
from math import sqrt
```

In [64]:
```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import GridSearchCV
# Define the model
model = LinearRegression()

# Define the hyperparameters and their possible values
param_grid = {
    'fit_intercept': [True, False],
    'positive': [True, False]
}

# Perform GridSearch cross-validation to find the best hyperparameters
grid_search = GridSearchCV(estimator=model, param_grid=param_grid, cv=5)
grid_search.fit(X_train, y_train)

# Get the best parameters and model
best_params = grid_search.best_params_
best_model = grid_search.best_estimator_
```

In [63]:
```
accuracy = accuracy_score(y_test, predictions)
precision = precision_score(y_test, predictions, average='weighted',zero_divisio
recall = recall_score(y_test, predictions, average='weighted')
rmse = sqrt(mean_squared_error(y_test, predictions))

print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("Root Mean Squared Error (RMSE):", rmse)
```

```
Accuracy: 0.02287539569994127
Precision: 0.25586620818353886
Recall: 0.02287539569994127
Root Mean Squared Error (RMSE): 14.269335657174674
```

Fine tuning the model

In [62]:
```
from math import sqrt
```

In [64]:
In [63]:
```
from sklearn.linear_model import LinearRegression
accuracy = accuracy_score(y_test, predictions)
from sklearn.model_selection import GridSearchCV
precision = precision_score(y_test, predictions, average='weighted',zero_divisio
recall = recall_score(y_test, predictions, average='weighted')
# Define the model
rmse = sqrt(mean_squared_error(y_test, predictions))
model = LinearRegression()

print("Accuracy:", accuracy)
# Define the hyperparameters and their possible values
print("Precision:", precision)
param_grid = {
print("Recall:", recall)
    'fit_intercept': [True, False],
print("Root Mean Squared Error (RMSE):", rmse)
    'positive': [True, False]
}
```

```
Accuracy: 0.02287539569994127
# Perform Grid Search Cross-Validation to find the best hyperparameters
Precision: 0.25589662081833886
grid_search = GridSearchCV(estimator=model, param_grid=param_grid, cv=5)
Recall: 0.02287539569994127
grid_search.fit(X_train, y_train)
Root Mean Squared Error (RMSE): 14.269335657174674

# Get the best parameters and model
best_params = grid_search.best_params_
best_model = grid_search.best_estimator_
```

Applying statistical methods to draw conclusions and make predictions from data

In [65]:
```
# Hypothesis testing
```

In [66]:
```
df1
```

Out[66]:

	participantID	education	responseID	nativeLanguage_num	gender_num	city_num	country_
0	12	4	128	0	0	0	
1	12	4	129	0	0	0	
2	12	4	130	0	0	0	
3	12	4	131	0	0	0	
4	12	4	132	0	0	0	
...	...	...	...	...	...	...	
558498	43296	3	560050	2	0	305	
558499	43297	4	560015	2	0	5	
558500	43297	4	560016	2	0	5	
558501	43297	4	560017	2	0	5	
558502	43297	4	560018	2	0	5	

558503 rows × 13 columns

In [67]:
```
df1.sample(10000)
```

Out[67]:

	participantID	education	responseID	nativeLanguage_num	gender_num	city_num	country_
142624	10734	3	144212	0	0	0	
145144	10916	4	146732	0	0	-1	
69124	5175	3	69342	0	0	0	
15325	1111	5	15475	-1	1	-1	
444148	33427	3	445736	0	1	0	
...	...	...	...	...	...	...	
213075	16111	4	214645	0	0	0	
549395	42498	4	550965	0	0	0	
252119	18923	5	253725	2	0	82	
212803	16092	3	214391	0	0	0	
151009	11355	2	152597	-1	0	-1	

10000 rows × 13 columns

In [67]:
```
df1.sample(10000)
```

Out[67]:

	participantID	education	responseID	nativeLanguage_num	gender_num	city_num	country
142624	10734	3	144212	0	0	0	
145144	10916	4	146732	0	0	-1	
69124	5175	3	69342	0	0	0	
15325	1111	5	15475	-1	1	-1	
444148	33427	3	445736	0	1	0	
...	...	...	...	...	...	...	
213075	16111	4	214645	0	0	0	
549395	42498	4	550965	0	0	0	
252119	18923	5	253725	2	0	82	
212803	16092	3	214391	0	0	0	
151009	11355	2	152597	-1	0	-1	

10000 rows × 13 columns

In [68]:
```
import scipy.stats as stats
```

In [69]:
```
sample_mean=df1.age.sample(10000).mean()
sample_mean
```

Out[69]: 38.1231

In [74]:
```
sample_std=df1.age.sample(10000).std()
sample_std
```

Out[74]: 15.054943196221787

In [70]:
```
population_mean=df.age.mean()
population_mean
```

Out[70]: 37.796812192593414

In [75]:
```
population_std=df.age.std()
population_std
```

Out[75]: 15.118828395211377

In [76]:
```
import warnings
warnings.filterwarnings("ignore", category=RuntimeWarning)
```

In [83]:
```
z_score = (sample_mean - population_mean) / (population_std / np.sqrt(len(df1.ag
print("Z-score:", z_score)
```

Z-score: 2.1581553734013714

In [88]:
```
f_statistic, p_value = stats.f_oneway(df1.R1_num, df1.R2_num, df1.R3_num)
```

In [80]:
```
print("F-statistic:", f_statistic)
print("p-value:", p_value)
# For a two-tailed test
critical_z_left = stats.norm.ppf(alpha / 2)
critical_z_right = -critical_z_left  # Because it's symmetric

# For one-tailed test (e.g., testing for a mean less than)
critical_value = stats.norm.ppf(alpha)
```

F-statistic: 14471.828775582186
p-value: 0.0

In [81]:
```
if z_score < critical_z_left or z_score > critical_z_right:
    print("Reject the null hypothesis: The sample mean is significantly different
else:
    print("Fail to reject the null hypothesis: The sample mean is not significant
```

In [90]:
```
alpha = 0.05  # significance level
if p_value < alpha:
    print("Reject the null hypothesis: There are significant differences between
else:
    print("Fail to reject the null hypothesis: There are no significant differenc
```

Reject the null hypothesis: The sample mean is significantly different from the population mean.

In [84]:
```
# ANOVA test
```

Reject the null hypothesis: There are significant differences between the means.

In [79]:
```
f_statistic, p_value = stats.f_oneway(df1.R1_num, df1.R2_num, df1.R3_num)
```

In [80]:
```
print("F-statistic:", f_statistic)
print("p-value:", p_value)
```

```
F-statistic: 14471.828775582186
p-value: 0.0
```

In [88]:
```
alpha = 0.05  # significance level
if p_value < alpha:
    print("Reject the null hypothesis: There are significant differences between the means.")
else:
    print("Fail to reject the null hypothesis: There are no significant differences between the means.")
```

```
Reject the null hypothesis: There are significant differences between the means.
```

In [84]:
```
# ANOVA test
# For a two-tailed test
critical_z_left = stats.norm.ppf(alpha / 2)
critical_z_right = -critical_z_left  # Because it's symmetric

# For a one-tailed test (e.g., testing for a mean less than)
# critical_z = stats.norm.ppf(alpha)

if z_score < critical_z_left or z_score > critical_z_right:
    print("Reject the null hypothesis: The sample mean is significantly different from the population mean.")
else:
    print("Fail to reject the null hypothesis: The sample mean is not significantly different from the population mean.")
```

```
Reject the null hypothesis: The sample mean is significantly different from the population mean.
```

In [ ]: