task3

October 6, 2023

[138]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import linear_model
```

[139]:
```
df=pd.read_csv('DATA - 3.csv')
df.head(5)
```

[139]:

	participantID	age	nativeLanguage	gender	education	city	country	\
0	12	28	URU_R	Fe	4	Montevideo	Uruguay	
1	12	28	URU_R	Fe	4	Montevideo	Uruguay	
2	12	28	URU_R	Fe	4	Montevideo	Uruguay	
3	12	28	URU_R	Fe	4	Montevideo	Uruguay	
4	12	28	URU_R	Fe	4	Montevideo	Uruguay	

	responseID	section	cue	R1	R2	R3
0	128	set_2013	bar	abierto	cerveza	noche
1	129	set_2013	tren	expreso	nocturno	bala
2	130	set_2013	mano	libre	derecha	hermano
3	131	set_2013	sopa	fría	Mafalda	verde
4	132	set_2013	especie	ave	Darwin	extinción

[140]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 558503 entries, 0 to 558502
Data columns (total 13 columns):
 #   Column          Non-Null Count   Dtype 
---  ------          --------------   ----- 
 0   participantID   558503 non-null  int64 
 1   age             558503 non-null  int64 
 2   nativeLanguage  535914 non-null  object
 3   gender          558503 non-null  object
 4   education       558503 non-null  int64 
 5   city            406872 non-null  object
 6   country         555965 non-null  object
 7   responseID      558503 non-null  int64 
 8   section         558503 non-null  object
```

```
 9   cue             558503 non-null  object
 10  R1              558440 non-null  object
 11  R2              558445 non-null  object
 12  R3              558427 non-null  object
dtypes: int64(4), object(9)
memory usage: 55.4+ MB
```

[141]:
```
df.describe()
```

[141]:

	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%	20880.000000	35.000000	4.000000	280839.000000
75%	31387.000000	49.000000	4.000000	420464.500000
max	43297.000000	99.000000	5.000000	560428.000000

[142]:
```
df.isnull().sum()
```

[142]:
```
participantID          0
age                    0
nativeLanguage     22589
gender                 0
education              0
city              151631
country             2538
responseID             0
section                0
cue                    0
R1                    63
R2                    58
R3                    76
dtype: int64
```

[143]:
```
# handling null values
df['nativeLanguage']=df['nativeLanguage'].fillna("NA")
df['city']=df['city'].fillna("NA")
df['country']=df['country'].fillna("NA")
df['R1']=df['R1'].fillna("NA")
df['R2']=df['R2'].fillna("NA")
df['R3']=df['R3'].fillna("NA")
```

[144]:
```
df.isnull().sum()
```

[144]:
```
participantID     0
age               0
nativeLanguage    0
gender            0
education         0
city              0
country           0
responseID        0
section           0
cue               0
R1                0
R2                0
R3                0
dtype: int64
```

[145]:
```
x=df[['age','participantID']]
y=df[['education']]
```

[146]:
```
reg=linear_model.LinearRegression()
reg.fit(x,y)
```

[146]:
```
LinearRegression()
```

[147]:
```
reg.coef_
```

[147]:
```
array([[ 7.18341280e-03, -1.26228667e-05]])
```

[148]:
```
reg.intercept_
```

[148]:
```
array([3.64635162])
```

[149]:
```
reg.predict([[30,4]])
```

```
c:\Users\Lenovo\AppData\Local\Programs\Python\Python311\Lib\site-
packages\sklearn\base.py:464: UserWarning: X does not have valid feature names,
but LinearRegression was fitted with feature names
  warnings.warn(
```

[149]:
```
array([[3.86180351]])
```

[150]:
```
x
```

[150]:

	age	participantID
0	28	12
1	28	12
2	28	12
3	28	12

```
4         28              12
…      …             …
558498    33           43296
558499    60           43297
558500    60           43297
558501    60           43297
558502    60           43297

[558503 rows x 2 columns]
```

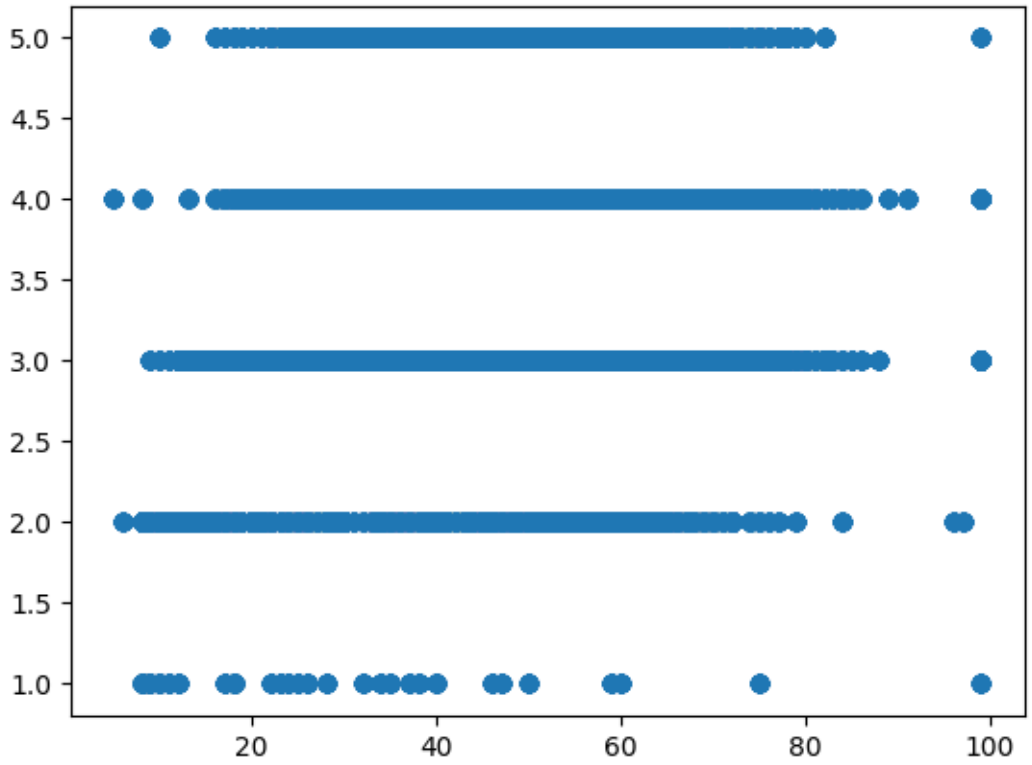
[151]:
```
y
```

[151]:
```
         education
0                4
1                4
2                4
3                4
4                4
…              …
558498           3
558499           4
558500           4
558501           4
558502           4

[558503 rows x 1 columns]
```

[152]:
```
plt.scatter(df['age'],df['education'])
```

[152]:
```
<matplotlib.collections.PathCollection at 0x2aac919dd90>
```



[153]:
```
from sklearn.model_selection import train_test_split
```

[154]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

[155]:
```
len(x_train)
```

[155]: 446802

[156]:
```
len(x_test)
```

[156]: 111701

[157]:
```
len(y_train)
```

[157]: 446802

[158]:
```
len(y_test)
```

[158]: 111701

[159]:
```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=1)
```

[160]:
```
knn.fit(x_train,y_train)    #training our dataset
```

```
c:\Users\Lenovo\AppData\Local\Programs\Python\Python311\Lib\site-
packages\sklearn\neighbors\_classification.py:228: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shape
of y to (n_samples,), for example using ravel().
  return self._fit(X, y)
```

[160]:
```
KNeighborsClassifier(n_neighbors=1)
```

[161]:
```
check=np.array([[13,1]])
check.shape
```

[161]:
```
(1, 2)
```

[162]:
```
test_predict=knn.predict(check)
print("His test section is: ",test_predict)      #predicting correct
```

```
His test section is:  [1]
```

```
c:\Users\Lenovo\AppData\Local\Programs\Python\Python311\Lib\site-
packages\sklearn\base.py:464: UserWarning: X does not have valid feature names,
but KNeighborsClassifier was fitted with feature names
  warnings.warn(
```

[163]:
```
y_pred=knn.predict(x_test)          #check prediction for test data
print("test set predictions: ",y_pred)
```

```
test set predictions:  [4 3 4 … 4 5 3]
```

[164]:
```
#checking accuracy of our model
knn.score(x_test,y_test)
```

[164]:
```
1.0
```

[165]:
```
import numpy as np
from scipy import stats

data1 = df[['age']]
data2 = df[['education']]

# Performing a two-sample t-test
t_statistic, p_value = stats.ttest_ind(data1, data2)

print("T-statistic:", t_statistic)
print("P-value:", p_value)

# Determine if the difference is statistically significant
```

```
alpha = 0.05  # desired significance level
if p_value < alpha:
    print("Reject the null hypothesis: There is a significant difference.")
else:
    print("Fail to reject the null hypothesis: There is no significant␣
 ↪difference.")
```

```
T-statistic: [1686.11765478]
P-value: [0.]
Reject the null hypothesis: There is a significant difference.
```

Chi-square test of 'participantID','age','education'

[166]:
```
from scipy.stats import chi2_contingency

# Creating a contingency table
observed = df[['participantID','age','education',]]

# Performing the chi-square test
chi2, p, dof, expected = chi2_contingency(observed)

print("Chi-square statistic:", chi2)
print("P-value:", p)
print("Degrees of freedom:", dof)
print("Expected frequencies table:")
print(expected)

# Determine if the difference is statistically significant
alpha = 0.05  # Our desired significance level
if p < alpha:
    print("Reject the null hypothesis: There is a significant relationship␣
 ↪between variables.")
else:
    print("Fail to reject the null hypothesis: There is no significant␣
 ↪relationship between variables.")
```

```
Chi-square statistic: 49896533.47729165
P-value: 0.0
Degrees of freedom: 1117004
Expected frequencies table:
[[4.39136345e+01 7.87562348e-02 7.60923064e-03]
 [4.39136345e+01 7.87562348e-02 7.60923064e-03]
 [4.39136345e+01 7.87562348e-02 7.60923064e-03]
 …
 [4.32758888e+04 7.76124795e+01 7.49872386e+00]
 [4.32758888e+04 7.76124795e+01 7.49872386e+00]
 [4.32758888e+04 7.76124795e+01 7.49872386e+00]]
Reject the null hypothesis: There is a significant relationship between
```

variables.

Performing ANOVA test with 'participantID','age','education' and responseID

[167]:
```
import scipy.stats as stats
group1 = df[['participantID']]
group2 = df[['age']]
group3 = df[['education']]
group4 = df[['responseID']]

#Using the f_oneway function from scipy.stats to perform the ANOVA test.
f_statistic, p_value = stats.f_oneway(group1, group2, group3,group4)

#print results
print("F-statistic:", f_statistic)
print("P-value:", p_value)

# making decision by Determining if there are statistically significant␣
 ↪differences among the groups by comparing the p-value to a chosen␣
 ↪significance level (alpha).

alpha = 0.05  # Set your desired significance level
if p_value < alpha:
    print("Reject the null hypothesis: There are significant differences among␣
 ↪the groups.")
else:
    print("Fail to reject the null hypothesis: There are no significant␣
 ↪differences among the groups.")
```

```
F-statistic: [1605124.68289494]
P-value: [0.]
Reject the null hypothesis: There are significant differences among the groups.
```