In [1]:
```python
import pandas as pd
```

In [2]:
```python
df = pd.read_csv('fe_binning.csv')
df.head()
```

Out[2]:

	country	happiness
0	Afghanistan	3.982855
1	Albania	4.606651
2	Argentina	6.697131
3	Armenia	4.348320
4	Australia	7.309061

In [3]:
```python
df.describe()
```

Out[3]:

	happiness
count	143.000000
mean	5.404037
std	1.116106
min	2.701591
25%	4.614304
50%	5.344383
75%	6.279204
max	7.603434

In [14]:
```python
binned = pd.cut(df['happiness'], bins=[2,4,6,10], labels=['L','M','H'])
df['happiness_band'] = binned
```

In [15]:
```
df.head()
```

Out[15]:

	country	happiness	happiness_band
0	Afghanistan	3.982855	L
1	Albania	4.606651	M
2	Argentina	6.697131	H
3	Armenia	4.348320	M
4	Australia	7.309061	H

In [16]:
```
df['happiness_band'].value_counts()
```

Out[16]:
```
M    80
H    44
L    19
Name: happiness_band, dtype: int64
```

In [17]:
```
mapping = pd.read_csv('country_region.csv')
```

In [19]:
```
mapping.head()
```

Out[19]:

	country	region
0	Afghanistan	South Asia
1	Albania	Europe & Central Asia
2	Algeria	Middle East & North Africa
3	American Samoa	East Asia & Pacific
4	Andorra	Europe & Central Asia

In [20]:
```
df.head()
```

Out[20]:

	country	happiness	happiness_band
0	Afghanistan	3.982855	L
1	Albania	4.606651	M
2	Argentina	6.697131	H
3	Armenia	4.348320	M
4	Australia	7.309061	H

In [22]:
```
df = pd.merge(df, mapping, on=['country', 'country'], how='left')
df.head()
```

Out[22]:

	country	happiness	happiness_band	region
0	Afghanistan	3.982855	L	South Asia
1	Albania	4.606651	M	Europe & Central Asia
2	Argentina	6.697131	H	Latin America & Caribbean
3	Armenia	4.348320	M	Europe & Central Asia
4	Australia	7.309061	H	East Asia & Pacific

In [23]:
```
df.isnull().mean()
```

Out[23]:
```
country           0.0
happiness         0.0
happiness_band    0.0
region            0.0
dtype: float64
```

In [24]:
```
df2 = pd.read_csv('fe_splitting.csv')
df2.head()
```

Out[24]:

	borough	property_type	timestamp_of_call
0	Kensington And chelsea	Purpose Built Flats/Maisonettes - 4 to 9 storeys	01/01/2017 16:48

	borough	property_type	timestamp_of_call
1	Camden	Purpose Built Flats/Maisonettes - 4 to 9 storeys	01/01/2017 22:20
2	Southwark	Purpose Built Flats/Maisonettes - 4 to 9 storeys	01/01/2017 09:51
3	Westminster	Purpose Built Flats/Maisonettes - 4 to 9 storeys	01/01/2017 00:28
4	Barking And dagenham	House - single occupancy	01/01/2017 13:33

In [26]:

```
df2.dtypes
```

Out[26]:

```
borough              object
property_type        object
timestamp_of_call    object
dtype: object
```

In [28]:

```
df2['timestamp_of_call'] = pd.to_datetime(df2['timestamp_of_call'])
df2.dtypes
```

Out[28]:

```
borough                      object
property_type                object
timestamp_of_call    datetime64[ns]
dtype: object
```

In [29]:

```
df2['day'] = df2['timestamp_of_call'].dt.day
df2['month'] = df2['timestamp_of_call'].dt.month
df2['year'] = df2['timestamp_of_call'].dt.year
df2['weekday'] = df2['timestamp_of_call'].dt.weekday
df2['hour'] = df2['timestamp_of_call'].dt.hour
df2.head()
```

Out[29]:

	borough	property_type	timestamp_of_call	day	month	year	weekday	hour
0	Kensington And chelsea	Purpose Built Flats/Maisonettes - 4 to 9 storeys	2017-01-01 16:48:00	1	1	2017	6	16

	borough	property_type	timestamp_of_call	day	month	year	weekday	hour
1	Camden	Purpose Built Flats/Maisonettes - 4 to 9 storeys	2017-01-01 22:20:00	1	1	2017	6	22
2	Southwark	Purpose Built Flats/Maisonettes - 4 to 9 storeys	2017-01-01 09:51:00	1	1	2017	6	9
3	Westminster	Purpose Built Flats/Maisonettes - 4 to 9 storeys	2017-01-01 00:28:00	1	1	2017	6	0
4	Barking And dagenham	House - single occupancy	2017-01-01 13:33:00	1	1	2017	6	13

In [34]:
```
df2.isnull().mean()
```

Out[34]:
```
borough              0.0
property_type        0.0
timestamp_of_call    0.0
day                  0.0
month                0.0
year                 0.0
weekday              0.0
hour                 0.0
dtype: float64
```

In [36]:
```
df2.head()
```

Out[36]:

	borough	property_type	timestamp_of_call	day	month	year	weekday	hour
0	Kensington And chelsea	Purpose Built Flats/Maisonettes - 4 to 9 storeys	2017-01-01 16:48:00	1	1	2017	6	16
1	Camden	Purpose Built Flats/Maisonettes - 4 to 9 storeys	2017-01-01 22:20:00	1	1	2017	6	22

	borough	property_type	timestamp_of_call	day	month	year	weekday	hour
2	Southwark	Purpose Built Flats/Maisonettes - 4 to 9 storeys	2017-01-01 09:51:00	1	1	2017	6	9
3	Westminster	Purpose Built Flats/Maisonettes - 4 to 9 storeys	2017-01-01 00:28:00	1	1	2017	6	0
4	Barking And dagenham	House - single occupancy	2017-01-01 13:33:00	1	1	2017	6	13

In [37]:
```
df2['property_type'].unique()
```

Out[37]:
```
array(['Purpose Built Flats/Maisonettes - 4 to 9 storeys',
       'House - single occupancy',
       'Converted Flat/Maisonette - Up to 2 storeys',
       'Purpose Built Flats/Maisonettes - Up to 3 storeys',
       'Purpose Built Flats/Maisonettes - 10 or more storeys',
       'Converted Flat/Maisonettes - 3 or more storeys',
       'Self contained Sheltered Housing',
       'Unlicensed House in Multiple Occupation - Up to 2 storeys',
       'House in Multiple Occupation - 3 or more storeys (not known if
licensed)',
       'Student Hall of Residence', 'Other Residential Home',
       'Unlicensed House in Multiple Occupation - 3 or more storeys',
       'Nursing/Care Home/Hospice', "Nurses'/Doctors' accommodation",
       'House in Multiple Occupation - Up to 2 storeys (not known if li
censed)',
       'Hotel/motel', "Children's Home",
       'Hostel (e.g. for homeless people)', 'Retirement/Old Persons Hom
e',
       'Licensed House in Multiple Occupation - Up to 2 storeys',
       'Bungalow - single occupancy', 'Other Dwelling',
       'Military/barracks', 'Houseboat (permanent dwelling)',
       'Licensed House in Multiple Occupation - 3 or more storeys',
       'Boarding House/B&B for homeless/asylum seekers',
       'Caravan/Mobile home (permanent dwelling)'], dtype=object)
```

In [38]:
```
df2[['property_type_type', 'property_type_size']] = df2['property_type'
```

```
].str.split('-', expand=True)
df2.head()
```

Out[38]:

	borough	property_type	timestamp_of_call	day	month	year	weekday	hour	property_t
0	Kensington And chelsea	Purpose Built Flats/Maisonettes - 4 to 9 storeys	2017-01-01 16:48:00	1	1	2017	6	16	Pur Flats/M
1	Camden	Purpose Built Flats/Maisonettes - 4 to 9 storeys	2017-01-01 22:20:00	1	1	2017	6	22	Pur Flats/M
2	Southwark	Purpose Built Flats/Maisonettes - 4 to 9 storeys	2017-01-01 09:51:00	1	1	2017	6	9	Pur Flats/M
3	Westminster	Purpose Built Flats/Maisonettes - 4 to 9 storeys	2017-01-01 00:28:00	1	1	2017	6	0	Pur Flats/M
4	Barking And dagenham	House - single occupancy	2017-01-01 13:33:00	1	1	2017	6	13	

In [39]:
```
df2.isnull().mean()
```

Out[39]:
```
borough                0.00000
property_type          0.00000
timestamp_of_call      0.00000
day                    0.00000
month                  0.00000
year                   0.00000
weekday                0.00000
hour                   0.00000
property_type_type     0.00000
property_type_size     0.10007
dtype: float64
```

In [40]:
```
df3 = pd.read_csv('fe_one_hot.csv')
df3.head()
```

Out[40]:

	country	happiness	region
0	Afghanistan	3.982855	South Asia
1	Albania	4.606651	Europe & Central Asia
2	Argentina	6.697131	Latin America & Caribbean
3	Armenia	4.348320	Europe & Central Asia
4	Australia	7.309061	East Asia & Pacific

In [41]:
```
region_one_hot = pd.get_dummies(df3.region)
region_one_hot.head()
```

Out[41]:

	East Asia & Pacific	Europe & Central Asia	Latin America & Caribbean	Middle East & North Africa	North America	South Asia	Sub-Saharan Africa
0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0
2	0	0	1	0	0	0	0
3	0	1	0	0	0	0	0
4	1	0	0	0	0	0	0

In [42]:
```
df3 = df3.join(region_one_hot).drop('region', axis=1)
df3.head()
```

Out[42]:

	country	happiness	East Asia & Pacific	Europe & Central Asia	Latin America & Caribbean	Middle East & North Africa	North America	South Asia	Sub-Saharan Africa
0	Afghanistan	3.982855	0	0	0	0	0	1	0
1	Albania	4.606651	0	1	0	0	0	0	0
2	Argentina	6.697131	0	0	1	0	0	0	0

	country	happiness	East Asia & Pacific	Europe & Central Asia	Latin America & Caribbean	Middle East & North Africa	North America	South Asia	Sub-Saharan Africa
3	Armenia	4.348320	0	1	0	0	0	0	0
4	Australia	7.309061	1	0	0	0	0	0	0

In [44]:
```
df4 = pd.read_csv('fe_calculated.csv')
df4.head()
```

Out[44]:

	country	gdp_usd	population
0	Afghanistan	1.936297e+10	37172386
1	Albania	1.505888e+10	2866376
2	Argentina	5.184750e+11	44494502
3	Armenia	1.243309e+10	2951776
4	Australia	1.432200e+12	24992369

In [47]:
```
df4['gdp_usd'] / df4['population']
```

Out[47]:
```
0        520.896603
1       5253.630064
2      11652.563276
3       4212.070943
4      57305.491928
           ...
125    17277.970110
126     1532.371639
127     2563.816070
128     1539.900158
129     2146.996385
Length: 130, dtype: float64
```

In [48]:
```
df4['per_capita'] = df4['gdp_usd'] / df4['population']
df4.head()
```

Out[48]:

	country	gdp_usd	population	per_capita
0	Afghanistan	1.936297e+10	37172386	520.896603
1	Albania	1.505888e+10	2866376	5253.630064
2	Argentina	5.184750e+11	44494502	11652.563276
3	Armenia	1.243309e+10	2951776	4212.070943
4	Australia	1.432200e+12	24992369	57305.491928

In [ ]: