EDA HR_Data

In [1]:
```
# Importing libraries

import pandas as pd
import numpy as np
import  matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
# Importing data

data_hr = pd.read_csv("/Users/prateek/Desktop/01.1 Coursework/04_Data Management/205_HW 5 Proj/aug_train.csv")
data_hr.head(10)
```

Out[2]:

	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university	education_level	major_discipline	experience	com
0	8949	city_103	0.920	Male	Has relevent experience	no_enrollment	Graduate	STEM	>20	
1	29725	city_40	0.776	Male	No relevent experience	no_enrollment	Graduate	STEM	15	
2	11561	city_21	0.624	NaN	No relevent experience	Full time course	Graduate	STEM	5	
3	33241	city_115	0.789	NaN	No relevent experience	NaN	Graduate	Business Degree	<1	
4	666	city_162	0.767	Male	Has relevent experience	no_enrollment	Masters	STEM	>20	
5	21651	city_176	0.764	NaN	Has relevent experience	Part time course	Graduate	STEM	11	
6	28806	city_160	0.920	Male	Has relevent experience	no_enrollment	High School	NaN	5	
7	402	city_46	0.762	Male	Has relevent experience	no_enrollment	Graduate	STEM	13	

	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university	education_level	major_discipline	experience	com
8	27107	city_103	0.920	Male	Has relevent experience	no_enrollment	Graduate	STEM	7	
9	699	city_103	0.920	NaN	Has relevent experience	no_enrollment	Graduate	STEM	17	

In [3]:
```
# Number of records and columns in data

data_hr.shape
```

Out[3]:
```
(19158, 14)
```

In [4]:
```
# Checking type of values present in data

for col in list(data_hr):
    print(col)
    print(data_hr[col].unique() )
```

```
enrollee_id
[ 8949 29725 11561 ... 24576  5756 23834]
city
['city_103' 'city_40' 'city_21' 'city_115' 'city_162' 'city_176'
 'city_160' 'city_46' 'city_61' 'city_114' 'city_13' 'city_159' 'city_102'
 'city_67' 'city_100' 'city_16' 'city_71' 'city_104' 'city_64' 'city_101'
 'city_83' 'city_105' 'city_73' 'city_75' 'city_41' 'city_11' 'city_93'
 'city_90' 'city_36' 'city_20' 'city_57' 'city_152' 'city_19' 'city_65'
 'city_74' 'city_173' 'city_136' 'city_98' 'city_97' 'city_50' 'city_138'
 'city_82' 'city_157' 'city_89' 'city_150' 'city_70' 'city_175' 'city_94'
 'city_28' 'city_59' 'city_165' 'city_145' 'city_142' 'city_26' 'city_12'
 'city_37' 'city_43' 'city_116' 'city_23' 'city_99' 'city_149' 'city_10'
 'city_45' 'city_80' 'city_128' 'city_158' 'city_123' 'city_7' 'city_72'
 'city_106' 'city_143' 'city_78' 'city_109' 'city_24' 'city_134' 'city_48'
 'city_144' 'city_91' 'city_146' 'city_133' 'city_126' 'city_118' 'city_9'
 'city_167' 'city_27' 'city_84' 'city_54' 'city_39' 'city_79' 'city_76'
 'city_77' 'city_81' 'city_131' 'city_44' 'city_117' 'city_155' 'city_33'
 'city_141' 'city_127' 'city_62' 'city_53' 'city_25' 'city_2' 'city_69'
 'city_120' 'city_111' 'city_30' 'city_1' 'city_140' 'city_179' 'city_55'
 'city_14' 'city_42' 'city_107' 'city_18' 'city_139' 'city_180' 'city_166'
```

```
 'city_121' 'city_129' 'city_8' 'city_31' 'city_171']
city_development_index
[0.92  0.776 0.624 0.789 0.767 0.764 0.762 0.913 0.926 0.827 0.843 0.804
 0.855 0.887 0.91  0.884 0.924 0.666 0.558 0.923 0.794 0.754 0.939 0.55
 0.865 0.698 0.893 0.796 0.866 0.682 0.802 0.579 0.878 0.897 0.949 0.925
 0.896 0.836 0.693 0.769 0.775 0.903 0.555 0.727 0.64  0.516 0.743 0.899
 0.915 0.689 0.895 0.89  0.847 0.527 0.766 0.738 0.647 0.795 0.74  0.701
 0.493 0.84  0.691 0.735 0.742 0.479 0.722 0.921 0.848 0.856 0.898 0.83
 0.73  0.68  0.725 0.556 0.448 0.763 0.745 0.645 0.788 0.78  0.512 0.739
 0.563 0.518 0.824 0.487 0.649 0.781 0.625 0.807 0.664]
gender
['Male' nan 'Female' 'Other']
relevent_experience
['Has relevent experience' 'No relevent experience']
enrolled_university
['no_enrollment' 'Full time course' nan 'Part time course']
education_level
['Graduate' 'Masters' 'High School' nan 'Phd' 'Primary School']
major_discipline
['STEM' 'Business Degree' nan 'Arts' 'Humanities' 'No Major' 'Other']
experience
['>20' '15' '5' '<1' '11' '13' '7' '17' '2' '16' '1' '4' '10' '14' '18'
 '19' '12' '3' '6' '9' '8' '20' nan]
company_size
[nan '50-99' '<10' '10000+' '5000-9999' '1000-4999' '10/49' '100-500'
 '500-999']
company_type
[nan 'Pvt Ltd' 'Funded Startup' 'Early Stage Startup' 'Other'
 'Public Sector' 'NGO']
last_new_job
['1' '>4' 'never' '4' '3' '2' nan]
training_hours
[ 36  47  83  52   8  24  18  46 123  32 108  23  26 106   7 132  68  50
  48  65  13  22 148  72  40 141  82 145 206 152  42  14 112  87  20  21
  92 102  43  45  19  90  25  15  98 142  28 228  29  12  17  35   4 136
  27  74  86  75 332 140 182 172  33  34 150 160   3   2 210 101  59 260
 131 109  70  51  60 164 290 133  76 156 120 100  39  55  49   6 125 326
 198  11  41 114 246  81  31  84 105  38 178 104 202  88 218  62  10  80
  77  37 162 190  30  16   5  54  44 110 262 107 134 103  96  57 240  94
 113  56  64 320   9 129  58 126 166  95  97 204 116 161 146 302  53 143
 124 214 288 306 322  67  61 130 220  78 314 226 280  91 234 163 151  85
 256 168 144  66 128  73 122 154  63 292 188  71 135 138 184  89 157 118
 111 192 127 216 139 196  99 167 276 121  69 155 316 242 304 284 278 310
 222 212 250 180 258 330 158 149 165  79 194 176 174 312 200 328 300 153
 232 336 308 147 298 224 254 248 236 170 264 119 117 334 324   1 238 266
```

```
 282 268 244 272 294 270 286]
target
[1. 0.]
```

In [5]:

```
# Checking for NA values in data

for col in data_hr:
    print(col)
    print(data_hr[col].isnull().sum().sum())
```

```
enrollee_id
0
city
0
city_development_index
0
gender
4508
relevent_experience
0
enrolled_university
386
education_level
460
major_discipline
2813
experience
65
company_size
5938
company_type
6140
last_new_job
423
training_hours
0
target
0
```

In [6]:

```
# Since few of the key columns have NA, filling them with value 'Unknown'
# 'Unknown' will be considered as a new category for further analysis
```

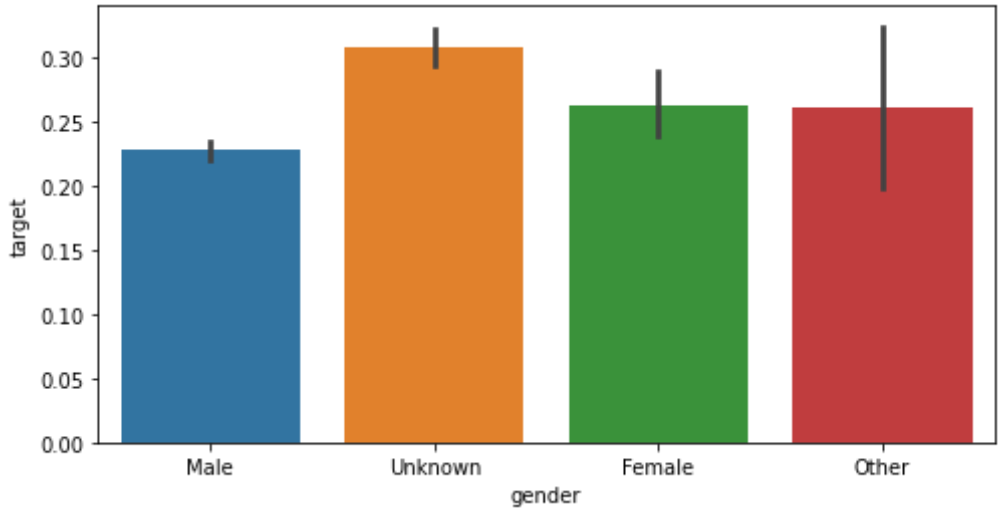
```
data_hr.fillna(value='Unknown', inplace=True )
data_hr.head(10)
```

Out[6]:

	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university	education_level	major_discipline	experience	co
0	8949	city_103	0.920	Male	Has relevent experience	no_enrollment	Graduate	STEM	>20	
1	29725	city_40	0.776	Male	No relevent experience	no_enrollment	Graduate	STEM	15	
2	11561	city_21	0.624	Unknown	No relevent experience	Full time course	Graduate	STEM	5	
3	33241	city_115	0.789	Unknown	No relevent experience	Unknown	Graduate	Business Degree	<1	
4	666	city_162	0.767	Male	Has relevent experience	no_enrollment	Masters	STEM	>20	
5	21651	city_176	0.764	Unknown	Has relevent experience	Part time course	Graduate	STEM	11	
6	28806	city_160	0.920	Male	Has relevent experience	no_enrollment	High School	Unknown	5	
7	402	city_46	0.762	Male	Has relevent experience	no_enrollment	Graduate	STEM	13	
8	27107	city_103	0.920	Male	Has relevent experience	no_enrollment	Graduate	STEM	7	
9	699	city_103	0.920	Unknown	Has relevent experience	no_enrollment	Graduate	STEM	17	

In [7]:

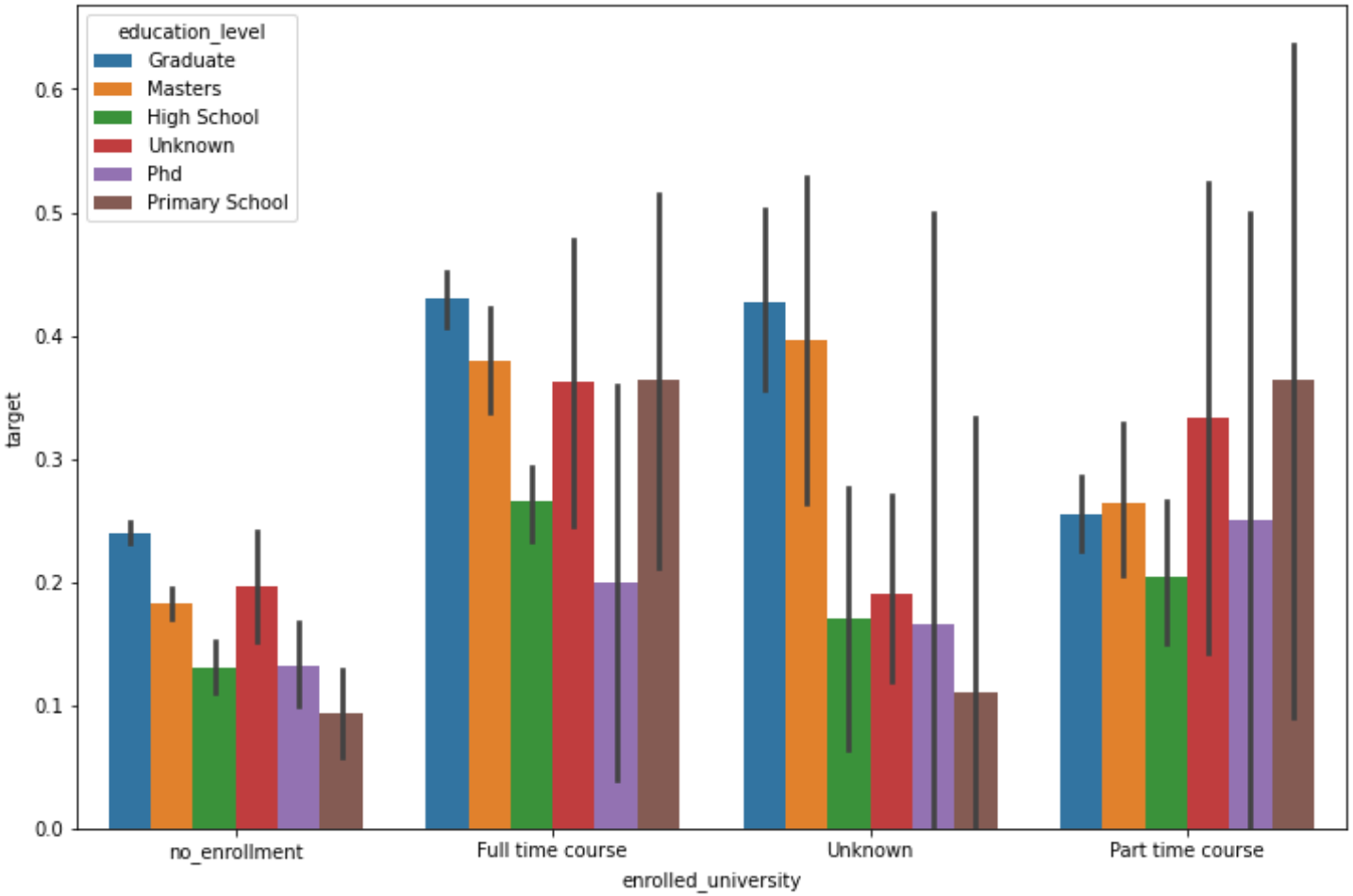
```
# Plotting gender wise interest for people 'Open for Job'

plt.figure(figsize=(8,4))
ax = sns.barplot(data=data_hr,x='gender', y='target')
```



As expected, there is a consistency across Gender for people open for Job

In [8]:
```
# Visualizing 'Education Type' & 'Education Level' for people open to job change

plt.figure(figsize=(12,8))
sns.barplot(data=data_hr ,x='enrolled_university', y='target', hue='education_level')
```

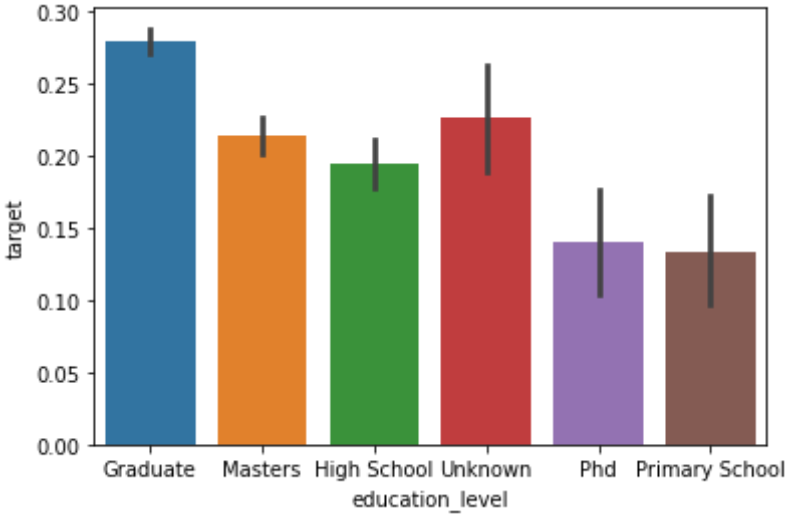
Out[8]:
```
<AxesSubplot:xlabel='enrolled_university', ylabel='target'>
```



The unkown category we created earlier from NA data is giving us valuable inputs. We can clearly see the population from unkown category is specially 'Grduates' & 'Masters' make a big chunk of population open for job

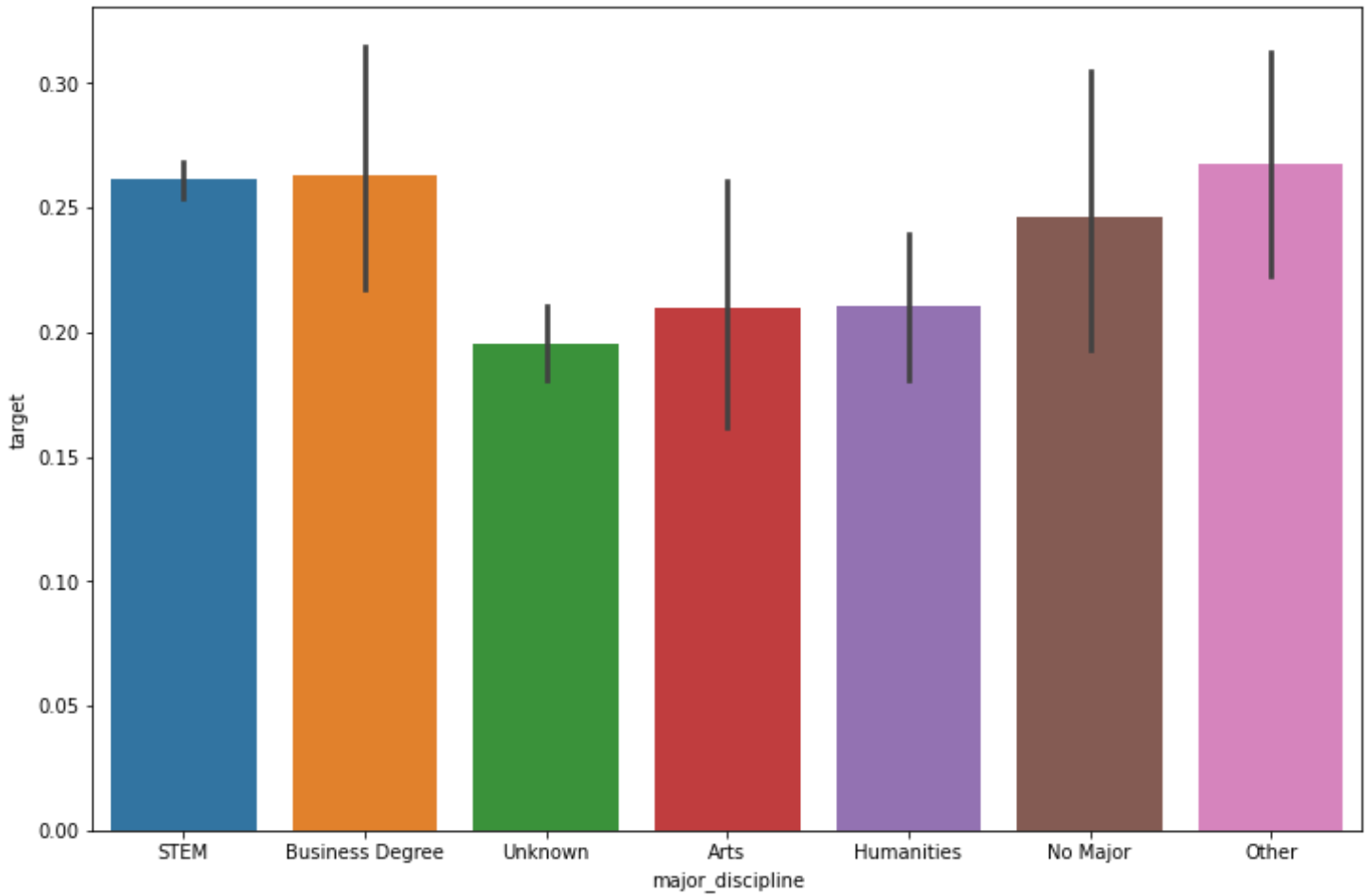
In [15]:
```
# PLotting education level of people open to job change

sns.barplot(data=data_hr ,x='education_level', y='target')
```

Out[15]:
```
<AxesSubplot:xlabel='education_level', ylabel='target'>
```



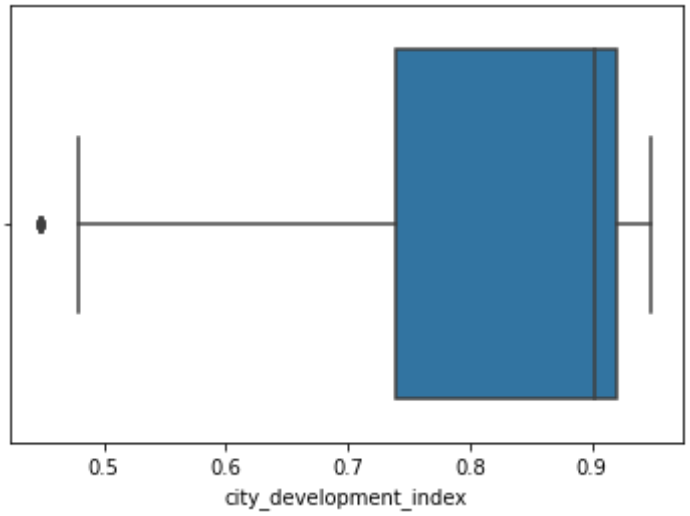
In [16]:
```
# Plotting Education Majors for people interested in Job change

plt.figure(figsize=(12,8))
sns.barplot(x='major_discipline', y='target', data=data_hr)
```

Out[16]: <AxesSubplot:xlabel='major_discipline', ylabel='target'>



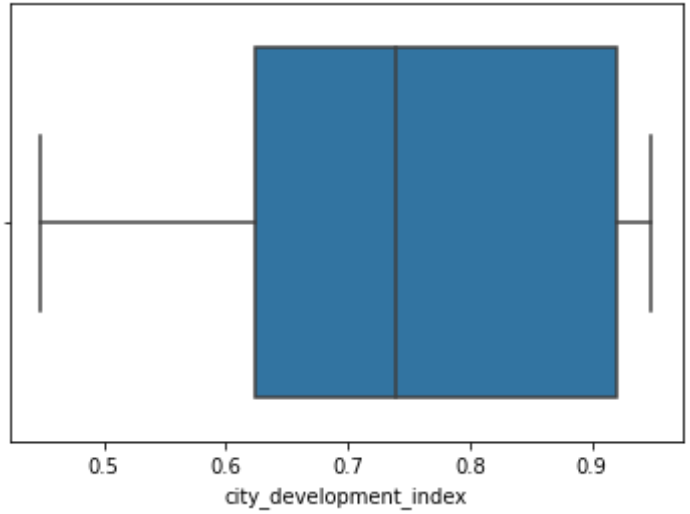
In [17]:
```
# Boxplot for 'City Development Index' for the entire population
sns.boxplot(data=data_hr, x='city_development_index')
```

Out[17]:
```
<AxesSubplot:xlabel='city_development_index'>
```



In [18]:
```
# 'City development index' for population 'Open to job'
target_data = data_hr[data_hr['target']==1]
sns.boxplot(data=target_data, x='city_development_index')
```

Out[18]: <AxesSubplot:xlabel='city_development_index'>

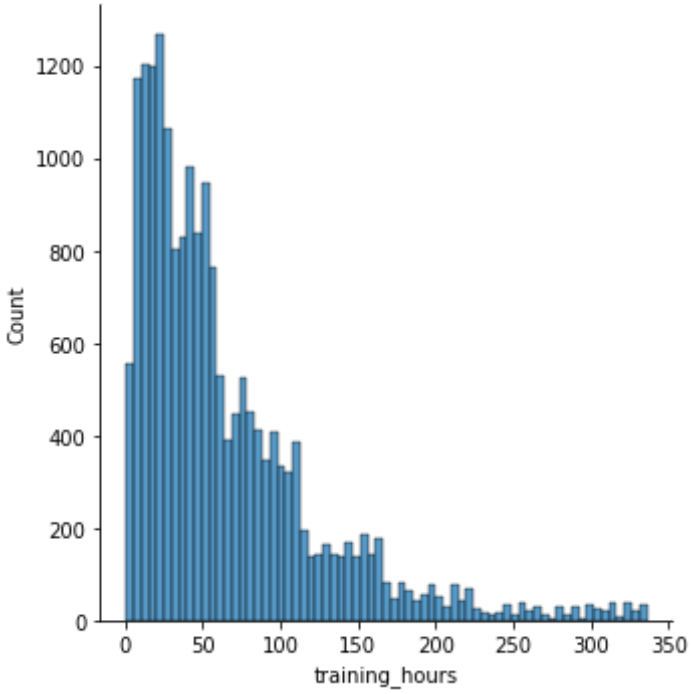


The median population open for Job Change is falls in the range [0.7 to 0.8] of 'city development index'

Compared to the median population with 0.9 'city development index'

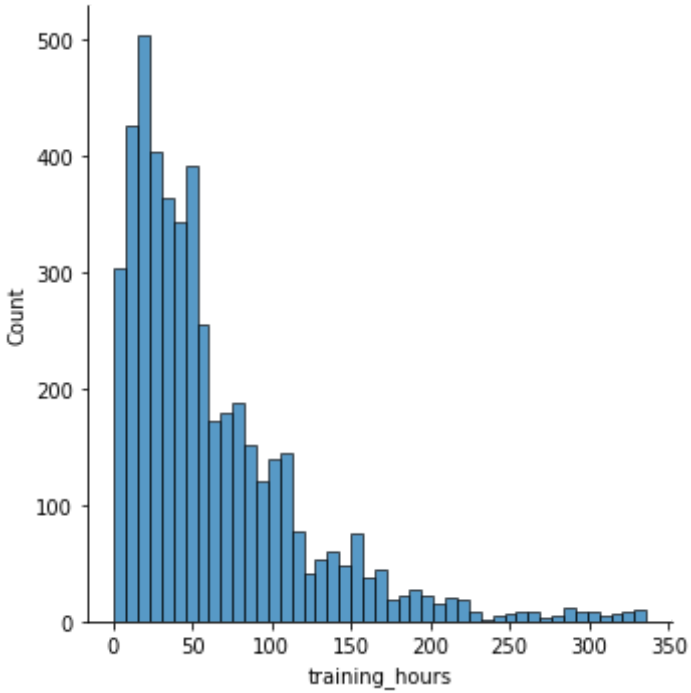
In [19]:
```
#Trainging hours for the entire population

sns.displot(data= data_hr['training_hours'])
```

Out[19]: <seaborn.axisgrid.FacetGrid at 0x123ca08b0>



In [20]:
```
# Reflecting the distribution plot for 'Training hours' for candidates 'open for Job'

sns.displot(data= target_data['training_hours'])
```

Out[20]: <seaborn.axisgrid.FacetGrid at 0x123ce5700>



As expected, traing hours for the 'entire population' seems to be a good representation for the training hours for the people 'Open to Job'