```
import os
import sys
from tempfile import NamedTemporaryFile
from urllib.request import urlopen
from urllib.parse import unquote, urlparse
from urllib.error import HTTPError
from zipfile import ZipFile
import tarfile
import shutil

CHUNK_SIZE = 40960
DATA_SOURCE_MAPPING = 'hr-analytics-job-change-of-data-scientists:https%3A%2F%2Fstorage.googleapis.com%2Fkaggle-data-sets%2F1019790%2F1719283%

KAGGLE_INPUT_PATH='/kaggle/input'
KAGGLE_WORKING_PATH='/kaggle/working'
KAGGLE_SYMLINK='kaggle'

!umount /kaggle/input/ 2> /dev/null
shutil.rmtree('/kaggle/input', ignore_errors=True)
os.makedirs(KAGGLE_INPUT_PATH, 0o777, exist_ok=True)
os.makedirs(KAGGLE_WORKING_PATH, 0o777, exist_ok=True)

try:
  os.symlink(KAGGLE_INPUT_PATH, os.path.join("..", 'input'), target_is_directory=True)
except FileExistsError:
  pass
try:
  os.symlink(KAGGLE_WORKING_PATH, os.path.join("..", 'working'), target_is_directory=True)
except FileExistsError:
  pass

for data_source_mapping in DATA_SOURCE_MAPPING.split(','):
    directory, download_url_encoded = data_source_mapping.split(':')
    download_url = unquote(download_url_encoded)
    filename = urlparse(download_url).path
    destination_path = os.path.join(KAGGLE_INPUT_PATH, directory)
    try:
        with urlopen(download_url) as fileres, NamedTemporaryFile() as tfile:
            total_length = fileres.headers['content-length']
            print(f'Downloading {directory}, {total_length} bytes compressed')
            dl = 0
            data = fileres.read(CHUNK_SIZE)
            while len(data) > 0:
                dl += len(data)
                tfile.write(data)
                done = int(50 * dl / int(total_length))
                sys.stdout.write(f"\r[{'=' * done}{' ' * (50-done)}] {dl} bytes downloaded")
                sys.stdout.flush()
                data = fileres.read(CHUNK_SIZE)
            if filename.endswith('.zip'):
              with ZipFile(tfile) as zfile:
                zfile.extractall(destination_path)
            else:
              with tarfile.open(tfile.name) as tarfile:
                tarfile.extractall(destination_path)
            print(f'\nDownloaded and uncompressed: {directory}')
    except HTTPError as e:
        print(f'Failed to load (likely expired) {download_url} to path {destination_path}')
        continue
    except OSError as e:
        print(f'Failed to load {download_url} to path {destination_path}')
        continue

print('Data source import complete.')
```

```
Downloading hr-analytics-job-change-of-data-scientists, 301600 bytes compressed
[==================================================] 301600 bytes downloaded
Downloaded and uncompressed: hr-analytics-job-change-of-data-scientists
Data source import complete.
```

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

```
import warnings
warnings.simplefilter('ignore')
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
/kaggle/input/hr-analytics-job-change-of-data-scientists/aug_test.csv
/kaggle/input/hr-analytics-job-change-of-data-scientists/sample_submission.csv
/kaggle/input/hr-analytics-job-change-of-data-scientists/aug_train.csv
```

```
df = pd.read_csv('/kaggle/input/hr-analytics-job-change-of-data-scientists/aug_train.csv')
df
```

	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university	education_level	major_discipline
0	8949	city_103	0.920	Male	Has relevent experience	no_enrollment	Graduate	STEM
1	29725	city_40	0.776	Male	No relevent experience	no_enrollment	Graduate	STEM
2	11561	city_21	0.624	NaN	No relevent experience	Full time course	Graduate	STEM
3	33241	city_115	0.789	NaN	No relevent experience	NaN	Graduate	Business Degree
4	666	city_162	0.767	Male	Has relevent experience	no_enrollment	Masters	STEM
...	...	...	...	...	...	...	...	...
19153	7386	city_173	0.878	Male	No relevent experience	no_enrollment	Graduate	Humanities
19154	31398	city_103	0.920	Male	Has relevent experience	no_enrollment	Graduate	STEM
19155	24576	city_103	0.920	Male	Has relevent experience	no_enrollment	Graduate	STEM
19156	5756	city_65	0.802	Male	Has relevent experience	no_enrollment	High School	NaN
19157	23834	city_67	0.855	NaN	No relevent experience	no_enrollment	Primary School	NaN

19158 rows × 14 columns

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19158 entries, 0 to 19157
Data columns (total 14 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   enrollee_id             19158 non-null  int64  
 1   city                    19158 non-null  object 
 2   city_development_index  19158 non-null  float64
 3   gender                  14650 non-null  object 
 4   relevent_experience     19158 non-null  object 
 5   enrolled_university     18772 non-null  object 
 6   education_level         18698 non-null  object 
 7   major_discipline        16345 non-null  object 
 8   experience              19093 non-null  object 
 9   company_size            13220 non-null  object 
 10  company_type            13018 non-null  object 
 11  last_new_job            18735 non-null  object 
 12  training_hours          19158 non-null  int64  
 13  target                  19158 non-null  float64
dtypes: float64(2), int64(2), object(10)
memory usage: 2.0+ MB
```

```
df.isnull().sum()
```

	0
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

dtype: int64

```
df.nunique()
```

	0
enrollee_id	19158
city	123
city_development_index	93
gender	3
relevent_experience	2
enrolled_university	3
education_level	5
major_discipline	6
experience	22
company_size	8
company_type	6
last_new_job	6
training_hours	241
target	2

dtype: int64

Checking for Duplicate Values

```
df.duplicated().sum()
```

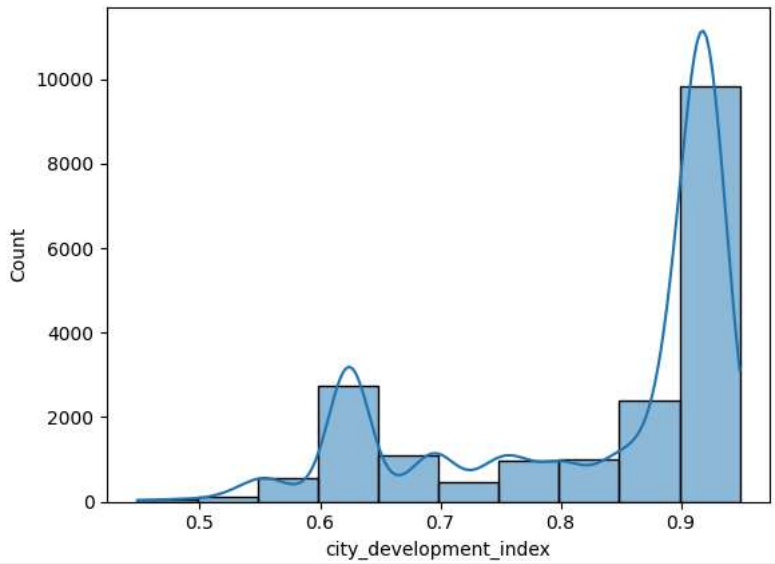
```
0
```

```
df['city_development_index'].describe()
```

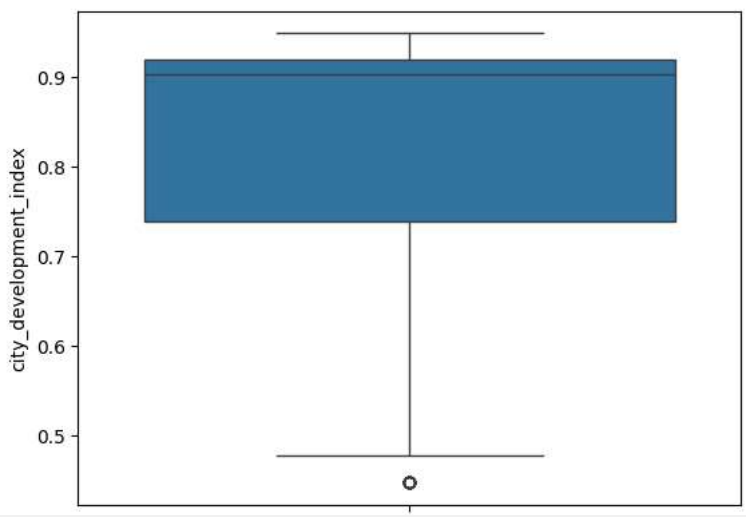
	city_development_index
count	19158.000000
mean	0.828848
std	0.123362
min	0.448000
25%	0.740000
50%	0.903000
75%	0.920000
max	0.949000

dtype: float64

```
sns.histplot(df['city_development_index'],bins=10,kde=True)
plt.show()
```




```
sns.boxplot(df['city_development_index'])
plt.show()
```

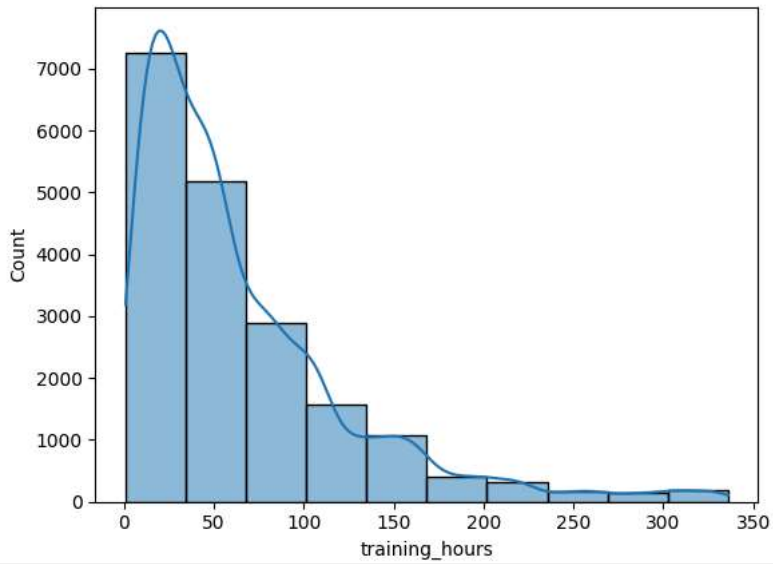



```
df['training_hours'].describe()
```

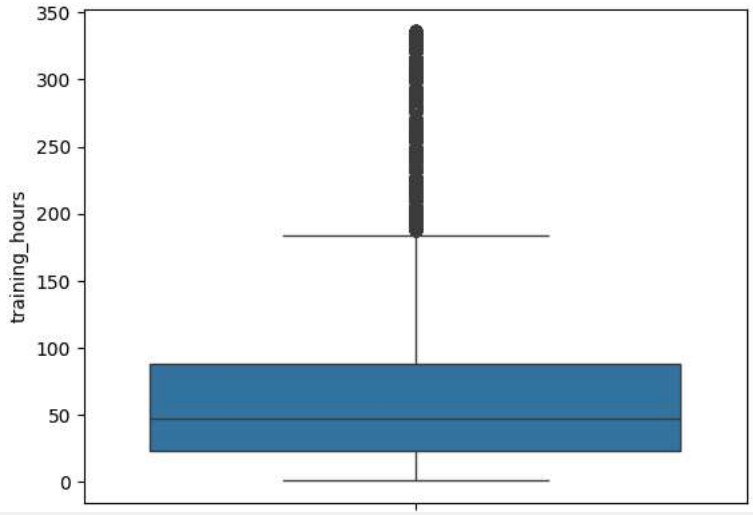
	training_hours
count	19158.000000
mean	65.366896
std	60.058462
min	1.000000
25%	23.000000
50%	47.000000
75%	88.000000
max	336.000000

dtype: float64

```
sns.histplot(df['training_hours'],bins=10,kde=True)
plt.show()
```




```
sns.boxplot(df['training_hours'])
plt.show()
```

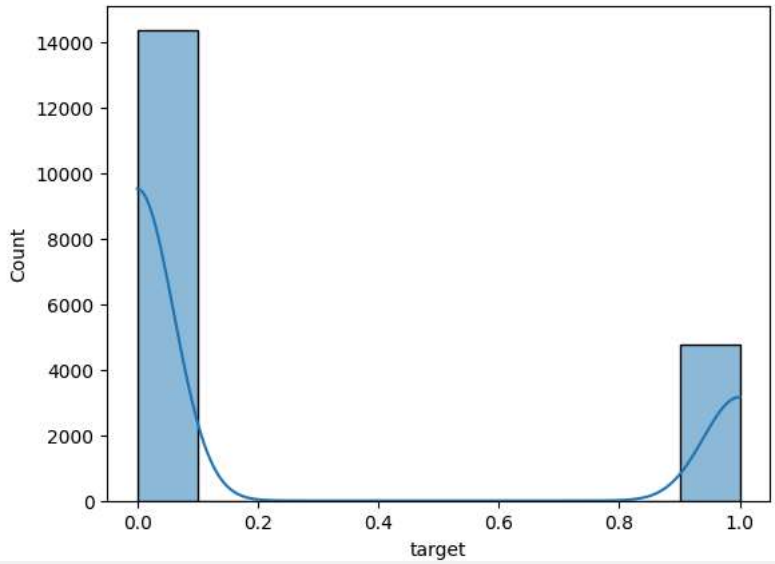



```
df['target'].describe()
```

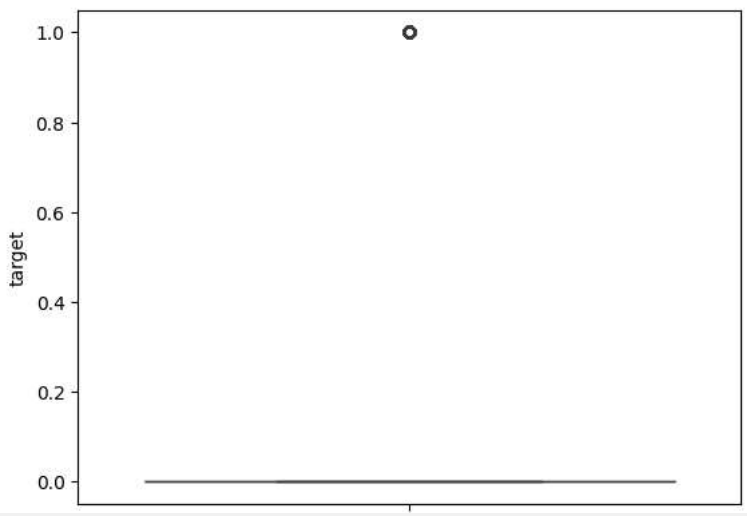
	target
count	19158.000000
mean	0.249348
std	0.432647
min	0.000000
25%	0.000000
50%	0.000000
75%	0.000000
max	1.000000

dtype: float64

```
sns.histplot(df['target'],bins=10,kde=True)
plt.show()
```




```
sns.boxplot(df['target'])
plt.show()
```




```
df.describe(include='object')
```

	city	gender	relevent_experience	enrolled_university	education_level	major_discipline	experience	company_size	company_t
count	19158	14650	19158	18772	18698	16345	19093	13220	13
unique	123	3	2	3	5	6	22	8	
top	city_103	Male	Has relevent experience	no_enrollment	Graduate	STEM	>20	50-99	Pv

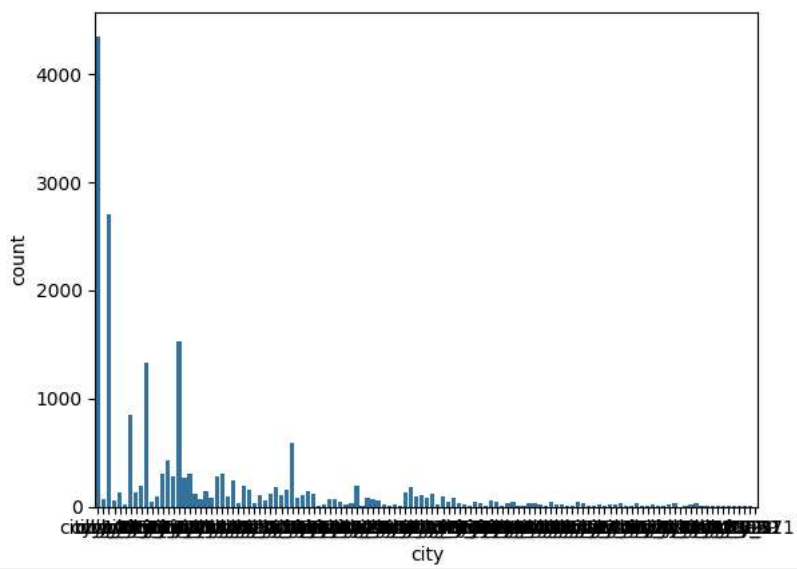
```
df['city'].value_counts()
```

city	count
city_103	4355
city_21	2702
city_16	1533
city_114	1336
city_160	845
...	...
city_129	3
city_111	3
city_121	3
city_140	1
city_171	1

123 rows × 1 columns

dtype: int64

```
sns.countplot(x='city', data=df)
plt.show()
```

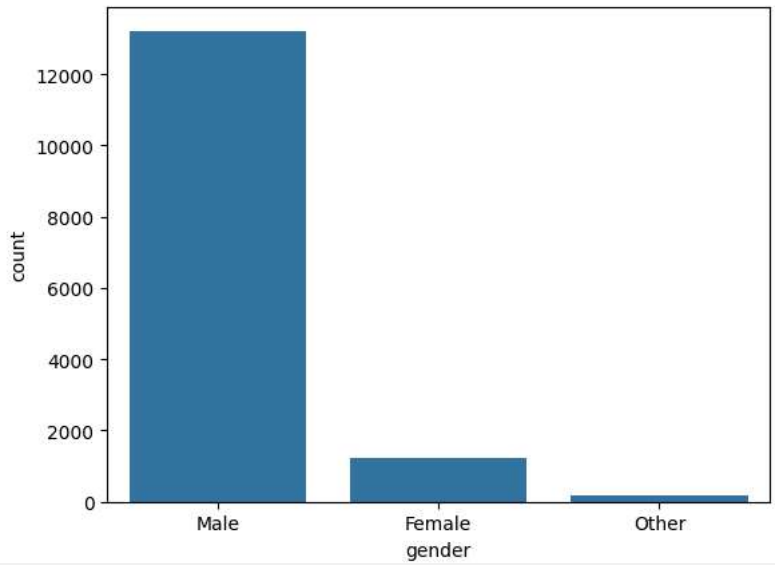



```
df['gender'].value_counts(dropna=False)
```

gender	count
Male	13221
NaN	4508
Female	1238
Other	191

dtype: int64

```
sns.countplot(x='gender',data=df)
plt.show()
```

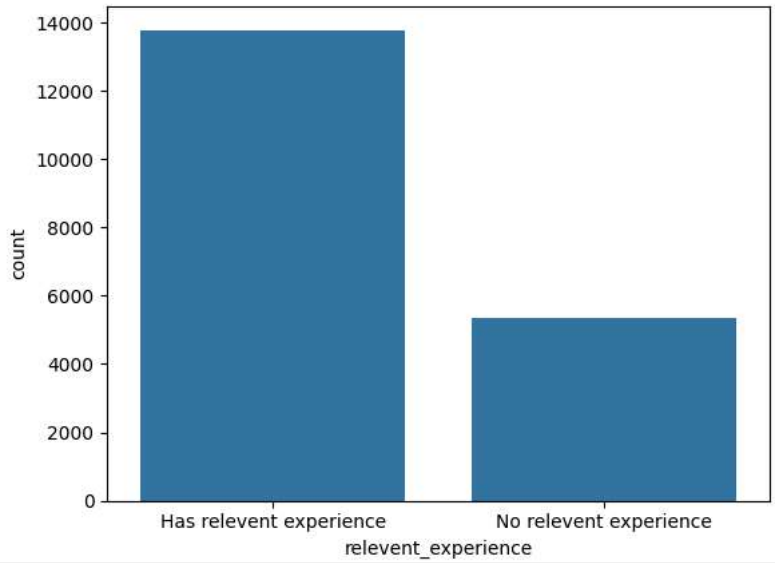



```
df['relevent_experience'].value_counts(dropna=False)
```

relevent_experience	count
Has relevent experience	13792
No relevent experience	5366

dtype: int64

```
sns.countplot(x='relevent_experience', data=df)
plt.show()
```

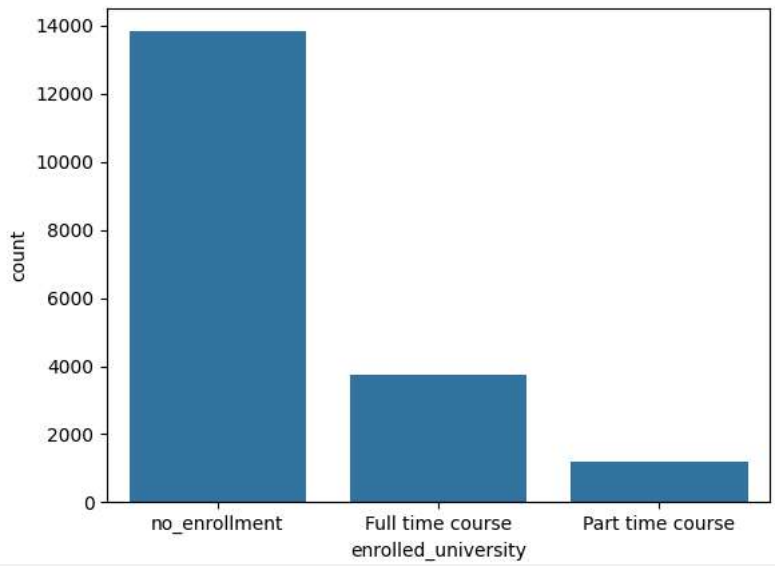



```
df['enrolled_university'].value_counts(dropna=False)
```

enrolled_university	count
no_enrollment	13817
Full time course	3757
Part time course	1198
NaN	386

dtype: int64

```
sns.countplot(x='enrolled_university', data=df)
plt.show()
```



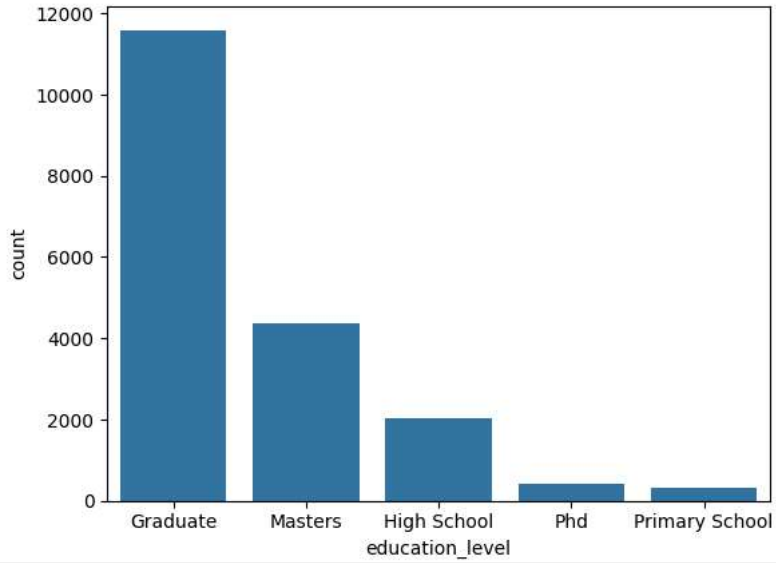
- Max employees have not enrolled in any university
- 386 employees have not specified

```
df['education_level'].value_counts(dropna=False)
```

education_level	count
Graduate	11598
Masters	4361
High School	2017
NaN	460
Phd	414
Primary School	308

dtype: int64

```
sns.countplot(x='education_level', data=df)
plt.show()
```



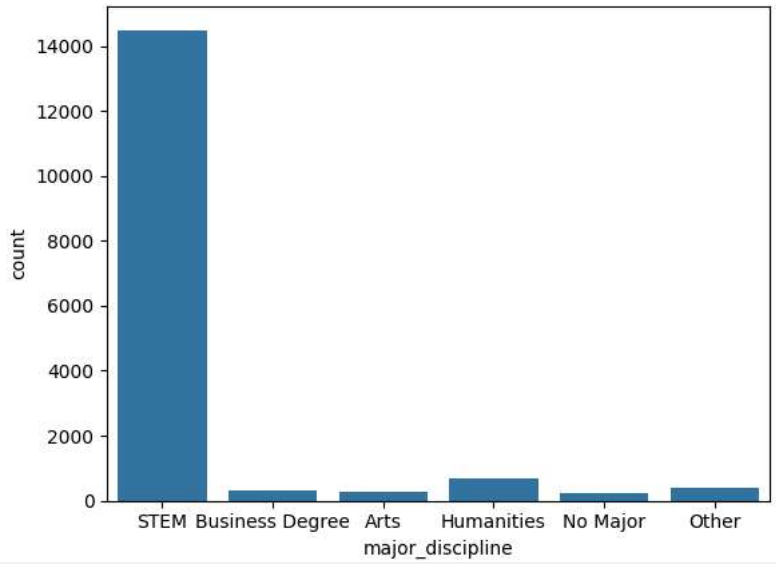
- Max employees have education level of Graduation
- 460 employees have not specified education level

```
df['major_discipline'].value_counts(dropna=False)
```

major_discipline	count
STEM	14492
NaN	2813
Humanities	669
Other	381
Business Degree	327
Arts	253
No Major	223

dtype: int64

```
sns.countplot(x='major_discipline', data=df)
plt.show()
```

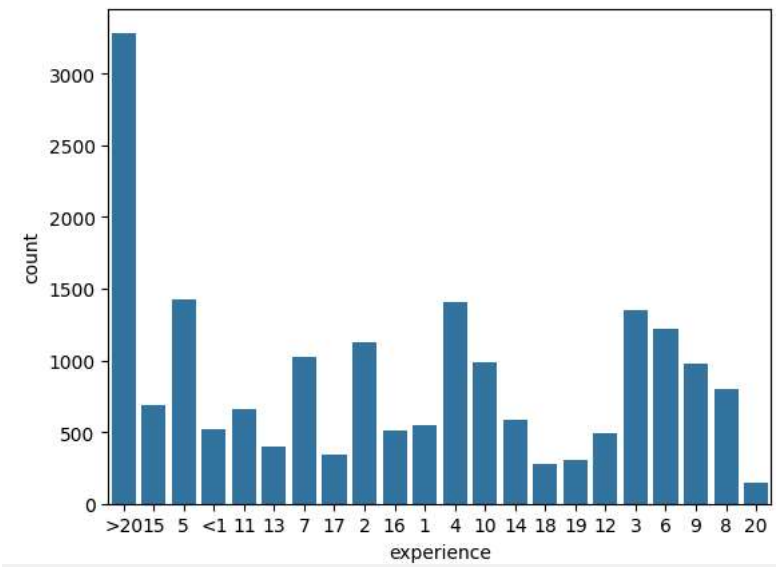



```
df['experience'].value_counts(dropna=False)
```

experience	count
>20	3286
5	1430
4	1403
3	1354
6	1216
2	1127
7	1028
10	985
9	980
8	802
15	686
11	664
14	586
1	549
<1	522
16	508
12	494
13	399
17	342
19	304
18	280
20	148
NaN	65

dtype: int64

```
sns.countplot(x='experience', data=df)
plt.show()
```

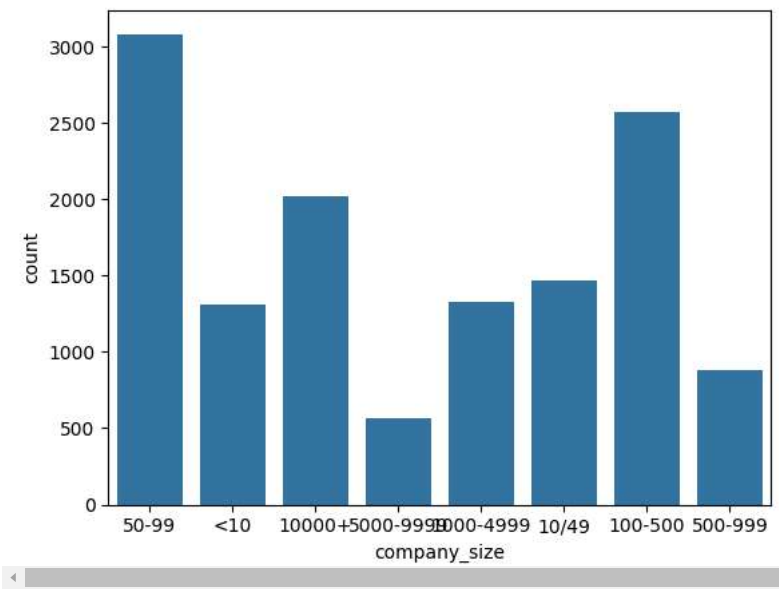



```
df['company_size'].value_counts(dropna=False)
```

company_size	count
NaN	5938
50-99	3083
100-500	2571
10000+	2019
10/49	1471
1000-4999	1328
<10	1308
500-999	877
5000-9999	563

dtype: int64

```
sns.countplot(x='company_size', data=df)
plt.show()
```



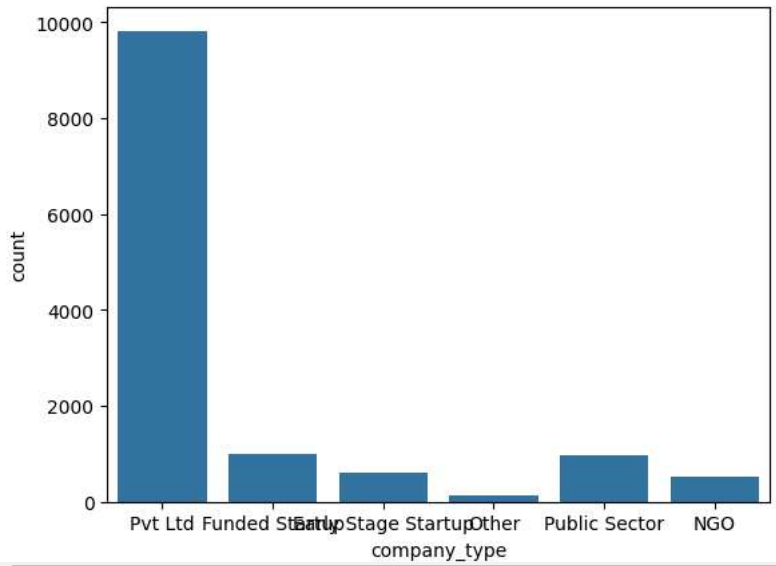
- Max employees have not specified their current company size

```
df['company_type'].value_counts(dropna=False)
```

company_type	count
Pvt Ltd	9817
NaN	6140
Funded Startup	1001
Public Sector	955
Early Stage Startup	603
NGO	521
Other	121

dtype: int64

```
sns.countplot(x='company_type', data=df)
plt.show()
```

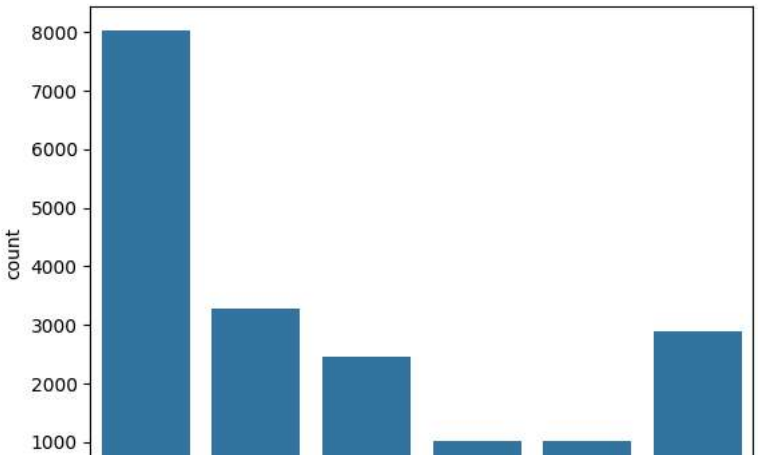



```
df['last_new_job'].value_counts(dropna=False)
```

last_new_job	count
1	8040
>4	3290
2	2900
never	2452
4	1029
3	1024
NaN	423

dtype: int64

```
sns.countplot(x='last_new_job', data=df)
plt.show()
```


8000
7000
6000
5000
4000
3000
2000
1000
count