Рубежный контроль 1

Максимова Екатерина ИУ5-23М

Вариант 10

In [1]:
```
# This Python 3 environment comes with many helpful analytics libraries insta
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will li

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

# You can write up to 20GB to the current directory (/kaggle/working/) that g
# You can also write temporary files to /kaggle/temp/, but they won't be save
```

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

In [13]:
```
data = pd.read_csv(
    '/kaggle/input/hr-analytics-job-change-of-data-scientists/aug_train.csv',
    sep=",")
```

In [4]:
```
data.head()
```

Out[4]:

	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university
0	8949	city_103	0.920	Male	Has relevent experience	no_enrollment
1	29725	city_40	0.776	Male	No relevent experience	no_enrollment
2	11561	city_21	0.624	NaN	No relevent experience	Full time course
3	33241	city_115	0.789	NaN	No relevent experience	NaN
4	666	city_162	0.767	Male	Has relevent experience	no_enrollment

In [7]:
```
data.shape
```

Out[7]:
```
(19158, 14)
```

Пропуски в данных

In [5]:
```
data.isna().sum()
```

Out[5]:
```
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
```

In [6]:
```
data.dtypes
```

Out[6]:
```
enrollee_id                int64
city                      object
city_development_index   float64
gender                    object
relevent_experience       object
enrolled_university       object
education_level           object
major_discipline          object
experience                object
company_size              object
company_type              object
last_new_job              object
training_hours             int64
target                   float64
dtype: object
```

Задача №10.

Для набора данных проведите устранение пропусков для одного (произвольного) категориального признака с использованием метода заполнения наиболее распространенным значением.

Заполним пропуски в признаке - gender

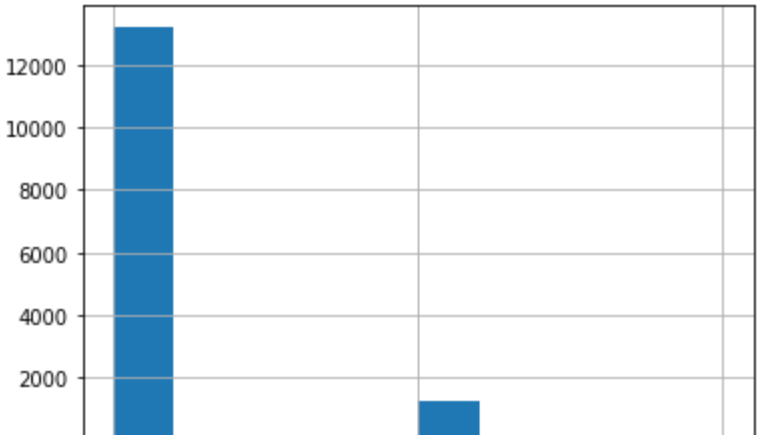
In [16]:
```
data['gender'].mode()
```

Out[16]:
```
0    Male
dtype: object
```

In [14]:
```
data['genderFull'] = data.gender.fillna(data['gender'].mode()[0])
```

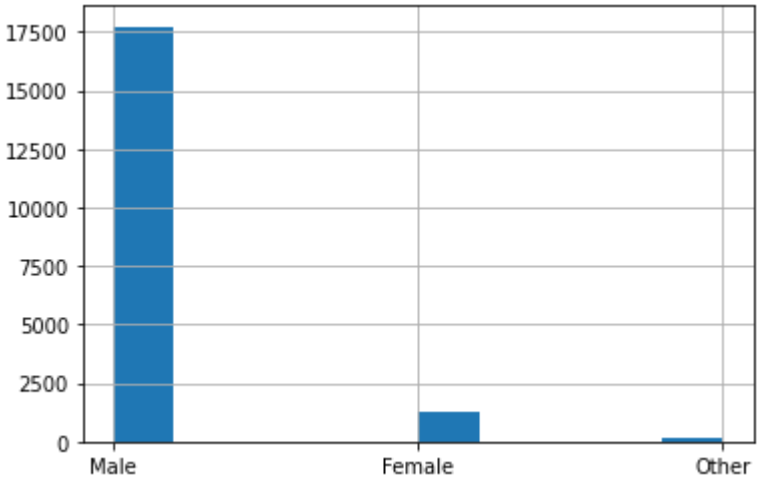
In [17]:
```
data.gender.hist()
```

Out[17]:
```
<AxesSubplot:>
```



In [18]:
```
data.genderFull.hist()
```

Out[18]: <AxesSubplot:>



Задача №30.

Для набора данных проведите удаление повторяющихся признаков.

В соответствии с описанием в данном датасете нет повторяющиеся признаков, поэтому мы можем их сгенерировать

А потом найдем повторяющиеся колонки с помощью функции get_duplicates

In [20]:
```
for col in data.columns:
    data[col+'1'] = data[col]
```

In [21]:
```
data.head()
```

Out[21]:

	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university
0	8949	city_103	0.920	Male	Has relevent experience	no_enrollment
1	29725	city_40	0.776	Male	No relevent experience	no_enrollment

	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university
2	11561	city_21	0.624	NaN	No relevent experience	Full time course
3	33241	city_115	0.789	NaN	No relevent experience	NaN
4	666	city_162	0.767	Male	Has relevent experience	no_enrollment

In [22]:
```
def get_duplicates(data):
    """
    Поиск дубликатов в колонках
    """
    pairs = {}
    dups = []
    # Перебор всех колонок (внешний)
    for i in range(data.shape[1]):
        # текущая колонка
        feat_outer = data.columns[i]
        # если текущая колонка не является дублем
        if feat_outer not in dups:
            # создаем запись в словаре, колонка является ключом
            pairs[feat_outer] = []
            # Перебор оставшихся колонок (внутренний)
            for feat_inner in data.columns[i + 1:]:
                # Если колонки идентичны
                if data[feat_outer].equals(data[feat_inner]):
                    # добавление в словарь и список дубликатов
                    pairs[feat_outer].append(feat_inner)
                    dups.append(feat_inner)
    return pairs
```

In [23]:
```
get_duplicates(data)
```

Out[23]:
```
{'enrollee_id': ['enrollee_id1'],
 'city': ['city1'],
 'city_development_index': ['city_development_index1'],
 'gender': ['gender1'],
 'relevent_experience': ['relevent_experience1'],
 'enrolled_university': ['enrolled_university1'],
 'education_level': ['education_level1'],
 'major_discipline': ['major_discipline1'],
 'experience': ['experience1'],
 'company_size': ['company_size1'],
 'company_type': ['company_type1'],
 'last_new_job': ['last_new_job1'],
 'training_hours': ['training_hours1'],
 'target': ['target1'],
 'genderFull': ['genderFull1']}
```

In [25]:
```
list(get_duplicates(data).values())[0]
```

Out[25]:
```
['enrollee_id1']
```

Удалим все дубликаты колонок

In [26]:
```
for i in list(get_duplicates(data).values()):
    for j in i:
        try:
            data.drop(columns=[j], inplace=True)
        except:
            print(j,'recently deleted')
```

В результате дубликаты колонок были удалены

In [27]:
```
data
```

Out[27]:

	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_unive
0	8949	city_103	0.920	Male	Has relevent experience	no_enrol
1	29725	city_40	0.776	Male	No relevent experience	no_enrol
2	11561	city_21	0.624	NaN	No relevent experience	Full time c
3	33241	city_115	0.789	NaN	No relevent experience	
4	666	city_162	0.767	Male	Has relevent experience	no_enrol
...	...	...	...	...	...	
19153	7386	city_173	0.878	Male	No relevent experience	no_enrol
19154	31398	city_103	0.920	Male	Has relevent experience	no_enrol
19155	24576	city_103	0.920	Male	Has relevent experience	no_enrol
19156	5756	city_65	0.802	Male	Has relevent experience	no_enrol
19157	23834	city_67	0.855	NaN	No relevent experience	no_enrol

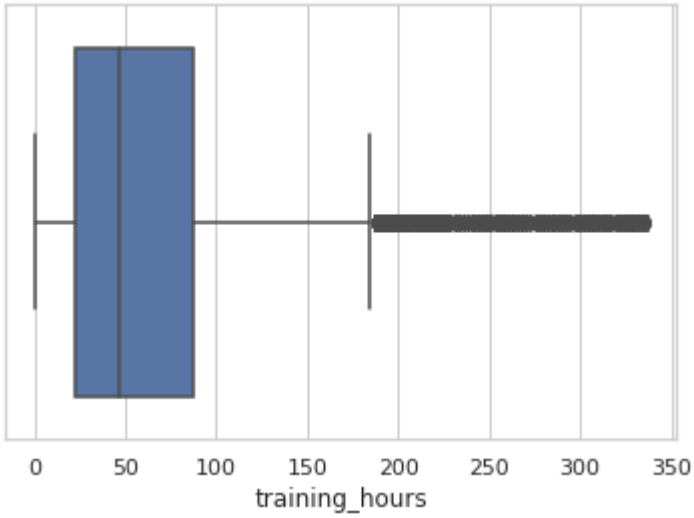
19158 rows × 15 columns

Доп задание

для произвольной колонки данных построить график "Ящик с усами (boxplot)"

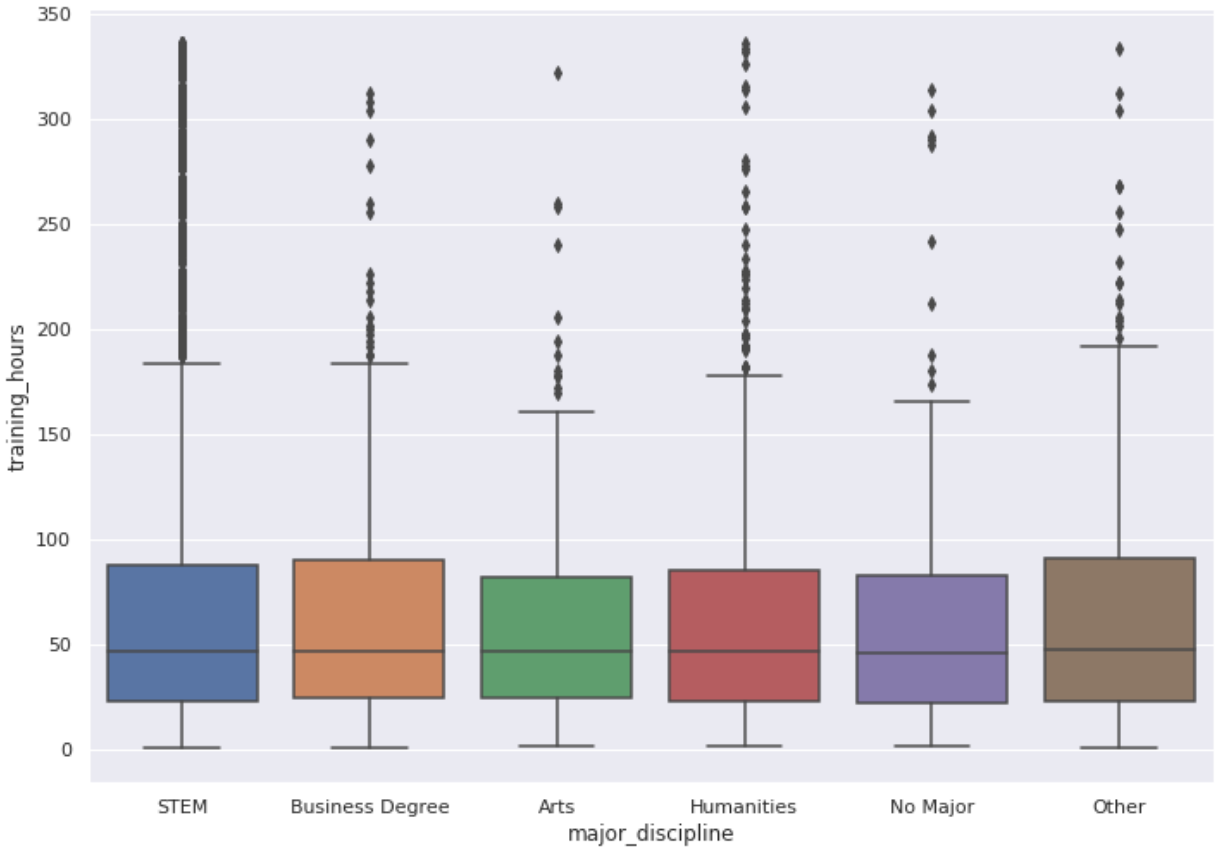
Построим boxplot для колонки training_hours - количество времени, потраченное на обучение

In [28]:
```
import seaborn as sns
sns.set_theme(style="whitegrid")
tips = sns.load_dataset("tips")
ax = sns.boxplot(x=data["training_hours"])
```



Распределение training_hours в зависимости от major_discipline - основного направления образования

In [32]:
```
sns.set(rc={'figure.figsize':(11.7,8.27)})
ax = sns.boxplot(x="major_discipline", y="training_hours", data=data)
```



Как видим, средние значения для всех направлений примерно равны - 50 часам

In [ ]: