**mport** pandas **as** pd

2

**import** seaborn **as** sns

3

​

**Import file**

In [2]:

1

udemy **=** pd.read\_csv("D:/python/PANDAS/uDEMY\_COURSES/udemy\_courses.csv",parse\_dates**=**['published\_timestamp'])

2

udemy.head(2)

Out[2]:

|  | **course\_id** | **course\_title** | **url** | **is\_paid** | **price** | **num\_subscribers** | **num\_reviews** | **num\_lectures** | **level** | **content\_duration** | **published\_timestamp** | **subject** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 1070968 | Ultimate Investment Banking Course | https://www.udemy.com/ultimate-investment-bank... | True | 200 | 2147 | 23 | 51 | All Levels | 1.5 | 2017-01-18 20:58:58+00:00 | Business Finance |
| **1** | 1113822 | Complete GST Course & Certification - Grow You... | https://www.udemy.com/goods-and-services-tax/ | True | 75 | 2792 | 923 | 274 | All Levels | 39.0 | 2017-03-09 16:34:20+00:00 | Business Finance |

**parse\_dates=['published\_timestamp'] change datetime column datatype**

In [3]:

1

udemy.dtypes

Out[3]:

course\_id int64

course\_title object

url object

is\_paid bool

price int64

num\_subscribers int64

num\_reviews int64

num\_lectures int64

level object

content\_duration float64

published\_timestamp datetime64[ns, UTC]

subject object

dtype: object

In [41]:

1

df1**=**udemy.copy()

2

df2**=**udemy.copy()

In [5]:

1

df1.head(1)

Out[5]:

|  | **course\_id** | **course\_title** | **url** | **is\_paid** | **price** | **num\_subscribers** | **num\_reviews** | **num\_lectures** | **level** | **content\_duration** | **published\_timestamp** | **subject** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 1070968 | Ultimate Investment Banking Course | https://www.udemy.com/ultimate-investment-bank... | True | 200 | 2147 | 23 | 51 | All Levels | 1.5 | 2017-01-18 20:58:58+00:00 | Business Finance |

In [6]:

1

df1.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 3678 entries, 0 to 3677

Data columns (total 12 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 course\_id 3678 non-null int64

1 course\_title 3678 non-null object

2 url 3678 non-null object

3 is\_paid 3678 non-null bool

4 price 3678 non-null int64

5 num\_subscribers 3678 non-null int64

6 num\_reviews 3678 non-null int64

7 num\_lectures 3678 non-null int64

8 level 3678 non-null object

9 content\_duration 3678 non-null float64

10 published\_timestamp 3678 non-null datetime64[ns, UTC]

11 subject 3678 non-null object

dtypes: bool(1), datetime64[ns, UTC](1), float64(1), int64(5), object(4)

memory usage: 319.8+ KB

In [7]:

1

df1.describe()

Out[7]:

|  | **course\_id** | **price** | **num\_subscribers** | **num\_reviews** | **num\_lectures** | **content\_duration** |
| --- | --- | --- | --- | --- | --- | --- |
| **count** | 3.678000e+03 | 3678.000000 | 3678.000000 | 3678.000000 | 3678.000000 | 3678.000000 |
| **mean** | 6.759720e+05 | 66.049483 | 3197.150625 | 156.259108 | 40.108755 | 4.094517 |
| **std** | 3.432732e+05 | 61.005755 | 9504.117010 | 935.452044 | 50.383346 | 6.053840 |
| **min** | 8.324000e+03 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **25%** | 4.076925e+05 | 20.000000 | 111.000000 | 4.000000 | 15.000000 | 1.000000 |
| **50%** | 6.879170e+05 | 45.000000 | 911.500000 | 18.000000 | 25.000000 | 2.000000 |
| **75%** | 9.613555e+05 | 95.000000 | 2546.000000 | 67.000000 | 45.750000 | 4.500000 |
| **max** | 1.282064e+06 | 200.000000 | 268923.000000 | 27445.000000 | 779.000000 | 78.500000 |

**Missing or null value findings**

In [8]:

1

df1.isnull().sum()

Out[8]:

course\_id 0

course\_title 0

url 0

is\_paid 0

price 0

num\_subscribers 0

num\_reviews 0

num\_lectures 0

level 0

content\_duration 0

published\_timestamp 0

subject 0

dtype: int64

In [9]:

1

null **=** df1.isnull().values.any()

2

print('Null value is prsent =', null)

Null value is prsent = False

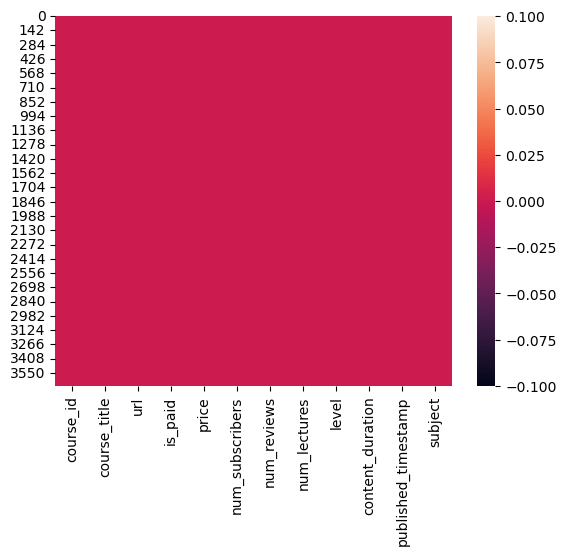
In [10]:

1

sns.heatmap(df1.isnull())

Out[10]:

<Axes: >



In [11]:

1

dup**=**df1.duplicated().any()

2

print('Duplicated values in dataset =', dup)

Duplicated values in dataset = True

In [12]:

1

*## Drop duplicated values*

2

df1**=** df1.drop\_duplicates()

In [13]:

1

dup**=**df1.duplicated().any()

2

print('Duplicated values in dataset =', dup)

Duplicated values in dataset = False

**Findout number of courses per subject**

In [14]:

1

df1.columns

Out[14]:

Index(['course\_id', 'course\_title', 'url', 'is\_paid', 'price',

'num\_subscribers', 'num\_reviews', 'num\_lectures', 'level',

'content\_duration', 'published\_timestamp', 'subject'],

dtype='object')

In [15]:

1

df1['subject'].value\_counts()

Out[15]:

Web Development 1199

Business Finance 1191

Musical Instruments 680

Graphic Design 602

Name: subject, dtype: int64

In [16]:

1

**import** matplotlib.pyplot **as** plt

In [17]:

1

​

2

sns.countplot(x**=**'subject', data**=**df1)

3

plt.xlabel('Subjects', fontsize **=** 12)

4

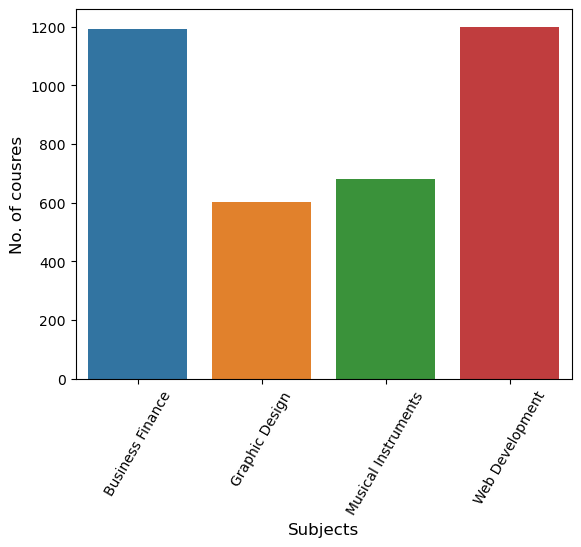
plt.ylabel('No. of cousres', fontsize **=** 12)

5

plt.xticks(rotation**=**60)

6

plt.show()



In [18]:

1

df1.columns

Out[18]:

Index(['course\_id', 'course\_title', 'url', 'is\_paid', 'price',

'num\_subscribers', 'num\_reviews', 'num\_lectures', 'level',

'content\_duration', 'published\_timestamp', 'subject'],

dtype='object')

In [19]:

1

df1['is\_paid'].value\_counts()

Out[19]:

True 3362

False 310

Name: is\_paid, dtype: int64

In [20]:

1

sns.countplot(x**=**'is\_paid', data**=**df1)

2

plt.xlabel('Couses', fontsize **=** 12)

3

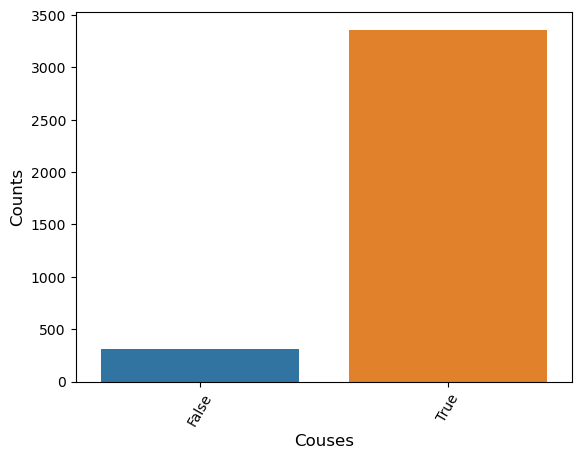
plt.ylabel('Counts', fontsize **=** 12)

4

plt.xticks(rotation**=**60)

5

plt.show()



**Which course has more lectures (Free or paid)**

In [21]:

1

df1.columns

2

​

Out[21]:

Index(['course\_id', 'course\_title', 'url', 'is\_paid', 'price',

'num\_subscribers', 'num\_reviews', 'num\_lectures', 'level',

'content\_duration', 'published\_timestamp', 'subject'],

dtype='object')

In [22]:

1

df1.groupby('is\_paid').mean()

C:\Users\Acer\AppData\Local\Temp\ipykernel\_8592\3404493598.py:1: FutureWarning: The default value of numeric\_only in DataFrameGroupBy.mean is deprecated. In a future version, numeric\_only will default to False. Either specify numeric\_only or select only columns which should be valid for the function.

df1.groupby('is\_paid').mean()

Out[22]:

|  | **course\_id** | **price** | **num\_subscribers** | **num\_reviews** | **num\_lectures** | **content\_duration** |
| --- | --- | --- | --- | --- | --- | --- |
| **is\_paid** |  |  |  |  |  |  |
| **False** | 691446.970968 | 0.000000 | 11533.951613 | 425.929032 | 21.416129 | 2.210753 |
| **True** | 674463.946163 | 72.198096 | 2421.270077 | 131.516359 | 41.866746 | 4.271584 |

**which course has a highest number od subcribers**

In [23]:

1

df1.groupby('is\_paid')['num\_subscribers'].max()

Out[23]:

is\_paid

False 268923

True 121584

Name: num\_subscribers, dtype: int64

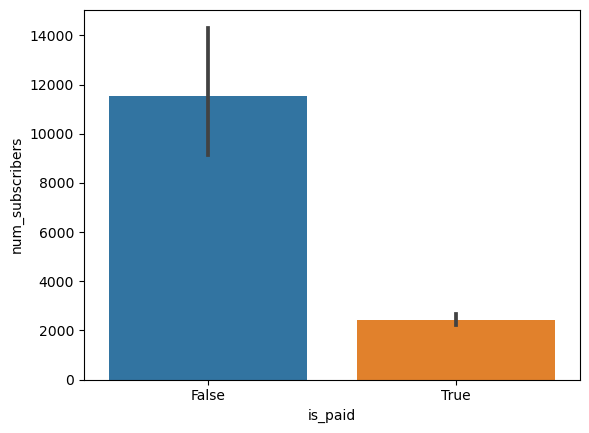
In [24]:

1

sns.barplot(x**=**'is\_paid', y**=**'num\_subscribers', data**=** df1)

Out[24]:

<Axes: xlabel='is\_paid', ylabel='num\_subscribers'>



**Which level has highest number of subcribers**

In [25]:

1

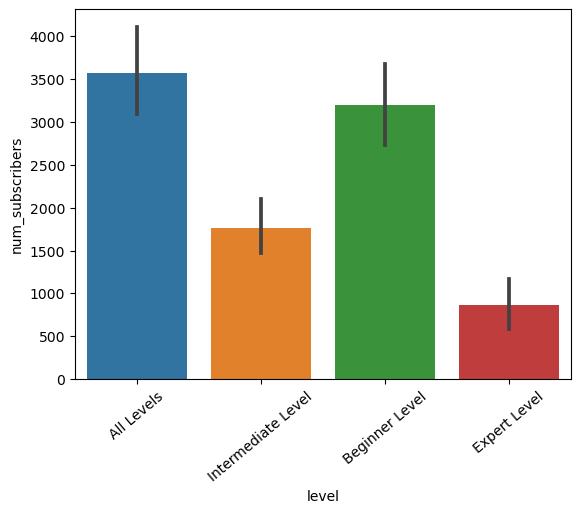
sns.barplot(x**=**'level', y**=**'num\_subscribers', data**=** df1)

2

plt.xticks(rotation**=**40)

3

plt.show()



**Find most propular course title**

In [26]:

1

df1.columns

Out[26]:

Index(['course\_id', 'course\_title', 'url', 'is\_paid', 'price',

'num\_subscribers', 'num\_reviews', 'num\_lectures', 'level',

'content\_duration', 'published\_timestamp', 'subject'],

dtype='object')

In [27]:

1

df1[df1['num\_subscribers'].max()**==**df1['num\_subscribers']]['course\_title']

Out[27]:

2827 Learn HTML5 Programming From Scratch

Name: course\_title, dtype: object

**Top 10 propular courses according to subcribers**

In [28]:

1

top\_10**=**df1.sort\_values(by**=**'num\_subscribers', ascending **=False**).head()

In [29]:

1

plt.figure(figsize**=**(8,6))

2

sns.barplot(x**=**'num\_subscribers', y**=**'course\_title', data**=** top\_10)

3

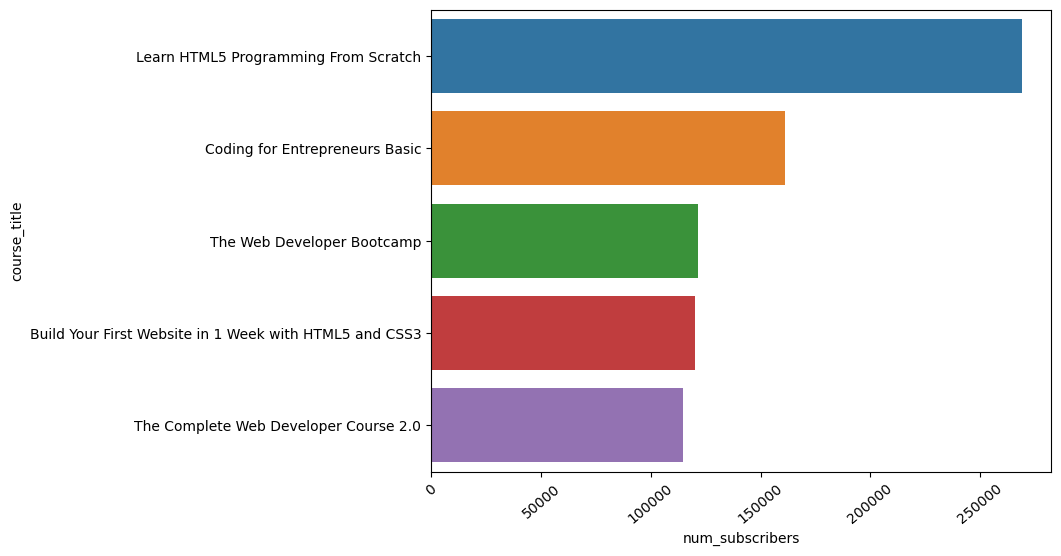
​

4

plt.xticks(rotation**=**40)

5

plt.show()



**Course has highest number of review**

In [30]:

1

df1.columns

Out[30]:

Index(['course\_id', 'course\_title', 'url', 'is\_paid', 'price',

'num\_subscribers', 'num\_reviews', 'num\_lectures', 'level',

'content\_duration', 'published\_timestamp', 'subject'],

dtype='object')

In [31]:

1

df1.sort\_values(by**=**'num\_reviews', ascending**=False**).head(1)['subject']

Out[31]:

3230 Web Development

Name: subject, dtype: object

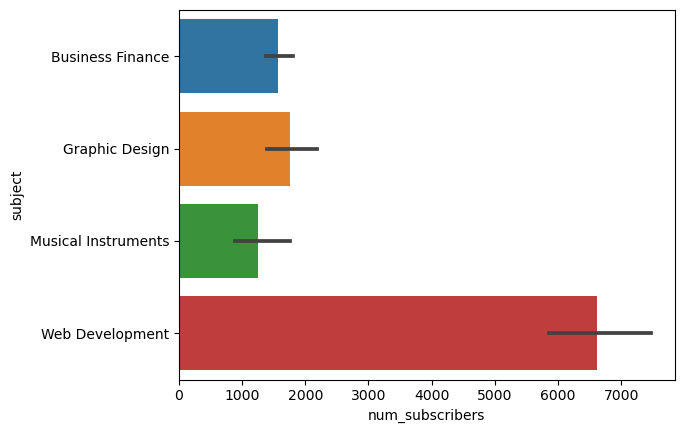
In [32]:

1

sns.barplot(x**=**'num\_subscribers', y**=**'subject', data**=** df1)

2

plt.show()



**Price impact on Number of reviews**

In [33]:

1

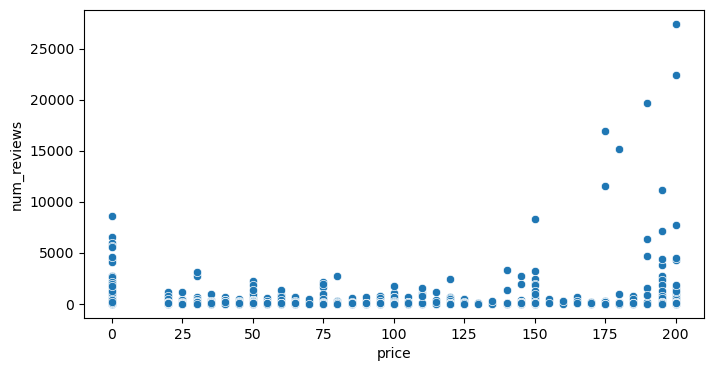
plt.figure(figsize**=**(8,4))

2

sns.scatterplot(x**=**'price', y**=** 'num\_reviews', data**=**df1)

3

plt.show()



**Total number courses realted to python**

In [34]:

1

len(df1[df1['course\_title'].str.contains('python', case **=** **False**)])

Out[34]:

29

**Top 10 python cousre as number of subcribers**

In [35]:

1

top\_10 **=**df1[df1['course\_title'].str.contains('python', case **=** **False**)][['num\_subscribers','course\_title']].\

2

sort\_values(by**=**'num\_subscribers',ascending**=False**).head(10)

3

top\_10

Out[35]:

|  | **num\_subscribers** | **course\_title** |
| --- | --- | --- |
| **2497** | 35267 | Web Programming with Python |
| **2570** | 23412 | Coding for Entrepreneurs: Learn Python, Django... |
| **2528** | 17714 | Learn Python and Django: Payment Processing |
| **3200** | 11832 | Python and Django Full Stack Web Developer Boo... |
| **3507** | 10917 | Fun and creative web engineering with Python a... |
| **2553** | 7827 | The Complete Ethical Hacking Course 2.0: Pytho... |
| **3138** | 7489 | Complete Python Web Course: Build 8 Python Web... |
| **3197** | 7407 | Try Django 1.9 | Build a Blog and Learn Python... |
| **2681** | 6153 | Python for Beginners: Python Programming Langu... |
| **3203** | 5151 | REST APIs with Flask and Python |

In [36]:

1

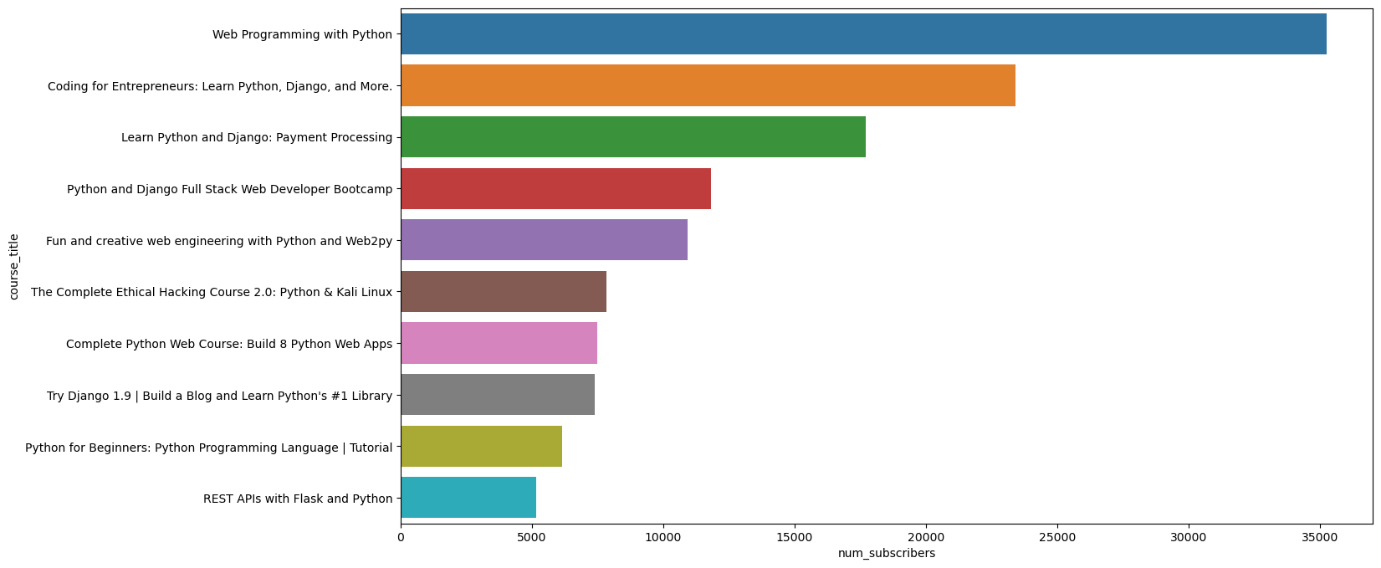
plt.figure(figsize**=**(15,8))

2

sns.barplot(x**=**'num\_subscribers',y**=**'course\_title', data **=** top\_10)

3

plt.show()



**In which year highest number of courses were posted**

In [42]:

1

df2['year']**=**df2['published\_timestamp'].dt.year

In [43]:

1

df2.head(2)

Out[43]:

|  | **course\_id** | **course\_title** | **url** | **is\_paid** | **price** | **num\_subscribers** | **num\_reviews** | **num\_lectures** | **level** | **content\_duration** | **published\_timestamp** | **subject** | **year** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 1070968 | Ultimate Investment Banking Course | https://www.udemy.com/ultimate-investment-bank... | True | 200 | 2147 | 23 | 51 | All Levels | 1.5 | 2017-01-18 20:58:58+00:00 | Business Finance | 2017 |
| **1** | 1113822 | Complete GST Course & Certification - Grow You... | https://www.udemy.com/goods-and-services-tax/ | True | 75 | 2792 | 923 | 274 | All Levels | 39.0 | 2017-03-09 16:34:20+00:00 | Business Finance | 2017 |

In [44]:

1

df2['year'].value\_counts()

Out[44]:

2016 1206

2015 1014

2017 715

2014 491

2013 202

2012 45

2011 5

Name: year, dtype: int64

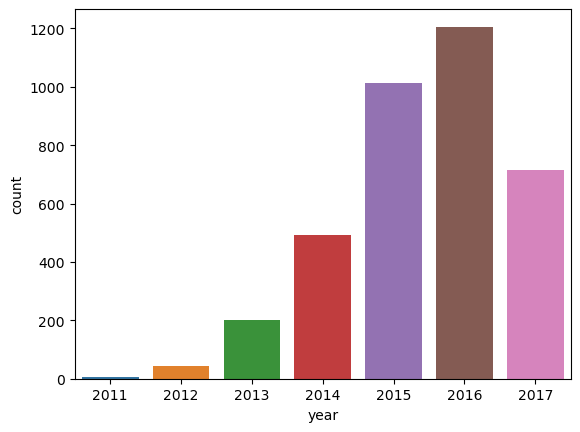
In [48]:

1

sns.countplot(x**=**'year', data**=**df2)

2

plt.show()



**Display category wise count of posted subjects [Year Wise]**

In [50]:

1

df2.columns

Out[50]:

Index(['course\_id', 'course\_title', 'url', 'is\_paid', 'price',

'num\_subscribers', 'num\_reviews', 'num\_lectures', 'level',

'content\_duration', 'published\_timestamp', 'subject', 'year'],

dtype='object')

In [53]:

1

df2.groupby('year')['subject'].value\_counts()

Out[53]:

year subject

2011 Web Development 5

2012 Web Development 19

Graphic Design 10

Musical Instruments 10

Business Finance 6

2013 Business Finance 84

Web Development 56

Musical Instruments 39

Graphic Design 23

2014 Business Finance 192

Musical Instruments 120

Web Development 113

Graphic Design 66

2015 Business Finance 339

Web Development 336

Musical Instruments 171

Graphic Design 168

2016 Web Development 448

Business Finance 349

Musical Instruments 228

Graphic Design 181

2017 Business Finance 225

Web Development 223

Graphic Design 155

Musical Instruments 112

Name: subject, dtype: int64