Udemy Courses (Group 7)

Data Analysis & Visualisation Project (BHCS15A)

Submitted By:

Tanya (19562)
Shivani (19585)
Anushaka (19595)

Supervisor:

Mrs. Anamika Gupta

2021

Department of Computer Science
Shaheed Sukhdev College of Business Studies
University of Delhi

ACKNOWLEDGEMENT

I would like to acknowledge all those without whom this project would not have been successful. Firstly, I would wish to thank our teacher **Mrs. Anamika Gupta** who guided me throughout the project and gave her immense support. She made us understand how to successfully complete this project and without her, the project would not have been complete.

This project has been a source to learn and bring our theoretical knowledge to the real-life world. So, I would really acknowledge her help and guidance for this project.

I would also like to thank my friends who have always helped me whenever needed.

Once again, thanks to everyone for making this project successful.

DECLARATION

I hereby declare that this project report, submitted to Mrs. Anamika Gupta of Shaheed Sukhdev College of Business Studies, University of Delhi is a record of an official work done by me. The project is submitted as an assignment for the course Data Analysis & Visualisation under the degree of B.Sc. (H) Computer Science. The results embodied have not been submitted to any other University or Institute for the award of any degree or diploma.

Udemy Courses

Description

This dataset contains 3,682 records of courses from 4 subjects (Business Finance, Graphic Design, Musical Instruments and Web Design) in 12 columns taken from Udemy. Udemy is a massive online open course (MOOC) platform that offers both free and paid courses. Anybody can create a course, a business model by which allowed Udemy to have hundreds of thousands of courses. This version modifies column names, removes empty columns and aggregates everything into a single csv file for ease of use.

Importing necessary libraries

In [64]:

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

Dataset

In [3]:

```
df = pd.read_csv("udemy_courses.csv")
df.head()
```

Out[3]:

	course_id	course_title	url	is_paid	price	num_subscribers	num_reviews	nu
0	1070968	Ultimate Investment Banking Course	https://www.udemy.com/ultimate-investment-bank...	True	200	2147	23	
1	1113822	Complete GST Course & Certification - Grow You...	https://www.udemy.com/goods-and-services-tax/	True	75	2792	923	
2	1006314	Financial Modeling for Business Analysts and C...	https://www.udemy.com/financial-modeling-for-b...	True	45	2174	74	

Checking size of the data

In [4]:

```
df.shape
```

Out[4]:

```
(3678, 12)
```

Descriptive Statistics

In [5]:

```
df.describe()
```

Out[5]:

	course_id	price	num_subscribers	num_reviews	num_lectures	content_durat
count	3.678000e+03	3678.000000	3678.000000	3678.000000	3678.000000	3678.0000
mean	6.759720e+05	66.049483	3197.150625	156.259108	40.108755	4.0945
std	3.432732e+05	61.005755	9504.117010	935.452044	50.383346	6.0538
min	8.324000e+03	0.000000	0.000000	0.000000	0.000000	0.0000
25%	4.076925e+05	20.000000	111.000000	4.000000	15.000000	1.0000
50%	6.879170e+05	45.000000	911.500000	18.000000	25.000000	2.0000
75%	9.613555e+05	95.000000	2546.000000	67.000000	45.750000	4.5000
max	1.282064e+06	200.000000	268923.000000	27445.000000	779.000000	78.5000

Checking missing values

In [6]:

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

Out[6]:

```
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
```

There are no missing values in the data

Removing duplicates from dataset

In [7]:

```
dfd = df          #creating copy of dataset
dfd.drop_duplicates(subset = "course_id",keep = False, inplace = True)
dfd
```

Out[7]:

	course_id	course_title	url	is_paid	price	num_subscribers	num_reviews
0	1070968	Ultimate Investment Banking Course	https://www.udemy.com/ultimate-investment-bank...	True	200	2147	23
1	1113822	Complete GST Course & Certification - Grow You...	https://www.udemy.com/goods-and-services-tax/	True	75	2792	923
2	1006314	Financial Modeling for Business Analysts and C...	https://www.udemy.com/financial-modeling-for-b...	True	45	2174	74

Renaming the course name

In [8]:

```
result = dfd.replace(to_replace = ['Ultimate Investment Banking Course', 'Angular 4: From T
result          #row which are changed -  0 and 2698
```

Out[8]:

	course_id	course_title	url	is_paid	price	num_subscribers	num_reviews
0	1070968	Banking Course	https://www.udemy.com/ultimate-investment-bank...	True	200	2147	23
1	1113822	Complete GST Course & Certification - Grow You...	https://www.udemy.com/goods-and-services-tax/	True	75	2792	923
2	1006314	Financial Modeling for Business Analysts and C...	https://www.udemy.com/financial-modeling-for-b...	True	45	2174	74
		Beginner to Pro -	https://www.udemy.com/complete-				

Seperating numeric columns

In [9]:

```
df_num = df.drop(["course_id", "course_title", "url", "is_paid", "level", "published_timest
df_num.head()
```

Out[9]:

	price	num_subscribers	num_reviews	num_lectures	content_duration
0	200	2147	23	51	1.5
1	75	2792	923	274	39.0
2	45	2174	74	51	2.5
3	95	2451	11	36	3.0
4	200	1276	45	26	2.0

Find average, min, max values of all numeric columns

In [10]:

```
#average value of columns
df_num.mean()
```

Out[10]:

```
price                 66.156574
num_subscribers     3184.001637
num_reviews          156.484179
num_lectures          40.171849
content_duration       4.100700
dtype: float64
```

In [11]:

```
#minimum value of columns
df_num.min()
```

Out[11]:

```
price               0.0
num_subscribers     0.0
num_reviews         0.0
num_lectures        0.0
content_duration    0.0
dtype: float64
```

In [12]:

```
#maximum value of columns
df_num.max()
```

Out[12]:

```
price                  200.0
num_subscribers     268923.0
num_reviews          27445.0
num_lectures           779.0
content_duration        78.5
dtype: float64
```

Find average, min, max values of all rows

In [13]:

```
#average value of all rows
df_num.mean(axis = 1)
```

Out[13]:

```
0       484.5
1       820.6
2       469.3
3       519.2
4       309.8
        ...
3673    235.4
3674     75.9
3675    154.7
3676     84.4
3677    200.8
Length: 3666, dtype: float64
```

In [14]:

```
#minimum value of all rows
df_num.min(axis = 1)
```

Out[14]:

```
0        1.5
1       39.0
2        2.5
3        3.0
4        2.0
        ...
3673     2.0
3674     3.0
3675     3.5
3676     3.0
3677     2.0
Length: 3666, dtype: float64
```

```
In [15]:
```

```
#maximum value of all rows
df_num.max(axis = 1)
```

```
Out[15]:
```

```
0       2147.0
1       2792.0
2       2174.0
3       2451.0
4       1276.0
         ...
3673    1040.0
3674     306.0
3675     513.0
3676     300.0
3677     901.0
Length: 3666, dtype: float64
```

Find the unique values of each column

```
In [16]:
```

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

```
[1070968 1113822 1006314 ...  635248  905096  297602]
['Ultimate Investment Banking Course'
 'Complete GST Course & Certification - Grow Your CA Practice'
 'Financial Modeling for Business Analysts and Consultants' ...
 'Learn and Build using Polymer'
 'CSS Animations: Create Amazing Effects on Your Website'
 "Using MODX CMS to Build Websites: A Beginner's Guide"]
['https://www.udemy.com/ultimate-investment-banking-course/'
 'https://www.udemy.com/goods-and-services-tax/'
 'https://www.udemy.com/financial-modeling-for-business-analysts-and-consu
ltants/'
 ... 'https://www.udemy.com/learn-and-build-using-polymer/'
 'https://www.udemy.com/css-animations-create-amazing-effects-on-your-webs
ite/'
 'https://www.udemy.com/using-modx-cms-to-build-websites-a-beginners-guid
e/']
[ True False]
[200  75  45  95 150  65 195  30  20  50 175 140 115 190 125  60 145 105
 155 185 180 120  25 160  40   0 100  90  35  80  70  55 165 130  85 170
 110 135]
```

Qcut as a “Quantile-based discretization function.”

Qcut tries to divide up the underlying data into equal sized bins

In [10]:

```
#using lambda and map function

price_list = df["price"].tolist()
#print("Converting the price to list:")
#price_list

final_price_list = list(map(lambda x: x*2, price_list))       #doubling the price
#final_price_list

dfd['price'] = final_price_list      #changing the price from new price list into dataframe
dfd
```

Out[10]:

	course_id	course_title	url	is_paid	price	num_subscribers	num_reviews
0	1070968	Ultimate Investment Banking Course	https://www.udemy.com/ultimate-investment-bank...	True	400	2147	23
1	1113822	Complete GST Course & Certification - Grow You...	https://www.udemy.com/goods-and-services-tax/	True	150	2792	923
2	1006314	Financial Modeling for Business Analysts and C...	https://www.udemy.com/financial-modeling-for-b...	True	90	2174	74

In [11]:

```
#using lambda and map function

price_list = df["price"].tolist()
#print("Converting the price to list:")
#price_list

final_price_list = list(map(lambda x: x*2, price_list))       #doubling the price
#final_price_list

dfd['price'] = final_price_list      #changing the price from new price list into dataframe
dfd
```

Out[11]:

	course_id	course_title	url	is_paid	price	num_subscribers	num_reviews
0	1070968	Ultimate Investment Banking Course	https://www.udemy.com/ultimate-investment-bank...	True	800	2147	23
1	1113822	Complete GST Course & Certification - Grow You...	https://www.udemy.com/goods-and-services-tax/	True	300	2792	923
2	1006314	Financial Modeling for Business Analysts and C...	https://www.udemy.com/financial-modeling-for-b...	True	180	2174	74

In [12]:

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

Out[12]:

```
count    3666.000000
mean      264.626296
std       244.264488
min         0.000000
25%        80.000000
50%       180.000000
75%       380.000000
max       800.000000
Name: price, dtype: float64
```

In [13]:

```
pd.qcut(df['price'], q=4)         #q=4 means creating 4 bins of equal size which means
                                  # it is dividing the price into four equal intervals
```

Out[13]:

```
0       (380.0, 800.0]
1       (180.0, 380.0]
2        (80.0, 180.0]
3       (180.0, 380.0]
4       (380.0, 800.0]
             ...
3673    (380.0, 800.0]
3674     (80.0, 180.0]
3675     (80.0, 180.0]
3676    (180.0, 380.0]
3677     (80.0, 180.0]
Name: price, Length: 3666, dtype: category
Categories (4, interval[float64]): [(-0.001, 80.0] < (80.0, 180.0] < (180.0,
380.0] < (380.0, 800.0]]
```

In [14]:

```
pd.qcut(df['num_subscribers'], q=7)     #it is dividing the num_subscribers into 7 equal in
```

Out[14]:

```
0       (2146.0, 5092.429]
1       (2146.0, 5092.429]
2       (2146.0, 5092.429]
3       (2146.0, 5092.429]
4       (1200.286, 2146.0]
               ...
3673     (578.0, 1200.286]
3674      (168.143, 578.0]
3675      (168.143, 578.0]
3676      (168.143, 578.0]
3677     (578.0, 1200.286]
Name: num_subscribers, Length: 3666, dtype: category
Categories (7, interval[float64]): [(-0.001, 30.0] < (30.0, 168.143] < (168.
143, 578.0] < (578.0, 1200.286] < (1200.286, 2146.0] < (2146.0, 5092.429] <
(5092.429, 268923.0]]
```

Questions

- Tell the course id and course title of the courses which has intermediate level and price 200 or above

In [39]:

```
df_sub = df.loc[((df['level'] == 'Intermediate Level') & (df['price'] >= 200)), ['course_id
df_sub
```

Out[39]:

	course_id	course_title
4	1011058	How To Maximize Your Profits Trading Options
125	528784	Stock market Investing Encyclopedia: How to in...
147	1070886	Python Algo Trading: FX Trading with Oanda
274	867440	Bitcoin: el futuro del dinero, hoy
332	990440	My Forex Strategy that win consistently over a...
415	1208148	Coaching Course:Investment Analysis for your c...
750	971110	The Truths about (in)secure Retirement
796	412856	Stock Market Option Trading: How Sell Options ...
806	1023650	Financial Modeling for Professionals in 1 Day!
902	1051430	Intermediate Accounting 1: Easy. Fast. Simple!
1014	1156530	How to trade in the Forex market
1973	384928	101 Blues riffs - learn how the harmonica supe...
2023	206088	Guitar Chord Mastery!Turn Your Brain Into A Ch...
2512	670034	Advanced Javascript
2554	991290	Dynamic JavaScript Master Class AJAX JSON Simp...
2698	929130	Angular 4: From Theory to Practice & FREE E-Book
2756	1236746	WordPress Tips and Tricks
3026	908996	Parse Server: From Front End to Full Stack
3366	1194232	Learning Path: Akka: Building Applications and...
3446	592594	3D Programming with WebGL and Babylon.js for B...
3592	976854	Spring 4 Mastercourse: Covers Annotation & XML...

- Tell the course id, name and url of the course which has maximum number of lectures

In [40]:

```
max_lec = df['num_lectures'].max()
df_lec = df.loc[(df['num_lectures'] == max_lec), ['course_id', 'course_title', 'url']]
df_lec
```

Out[40]:

	course_id	course_title	url
2707	79154	Back to School Web Development and Programming...	https://www.udemy.com/back-to-school-web-devel...

- Find the number of cources of a particular subject

In [22]:

```
df[df["subject"] == "Business Finance"]['course_id'].count()
```

Out[22]:

```
1187
```

There are 1187 courses on the subject of Business Finance

- Find the number of cources of every subject

In [21]:

```
course_count = df.groupby(['subject'])['course_id'].count()
course_count
```

Out[21]:

```
subject
Business Finance       1187
Graphic Design          601
Musical Instruments     680
Web Development        1198
Name: course_id, dtype: int64
```

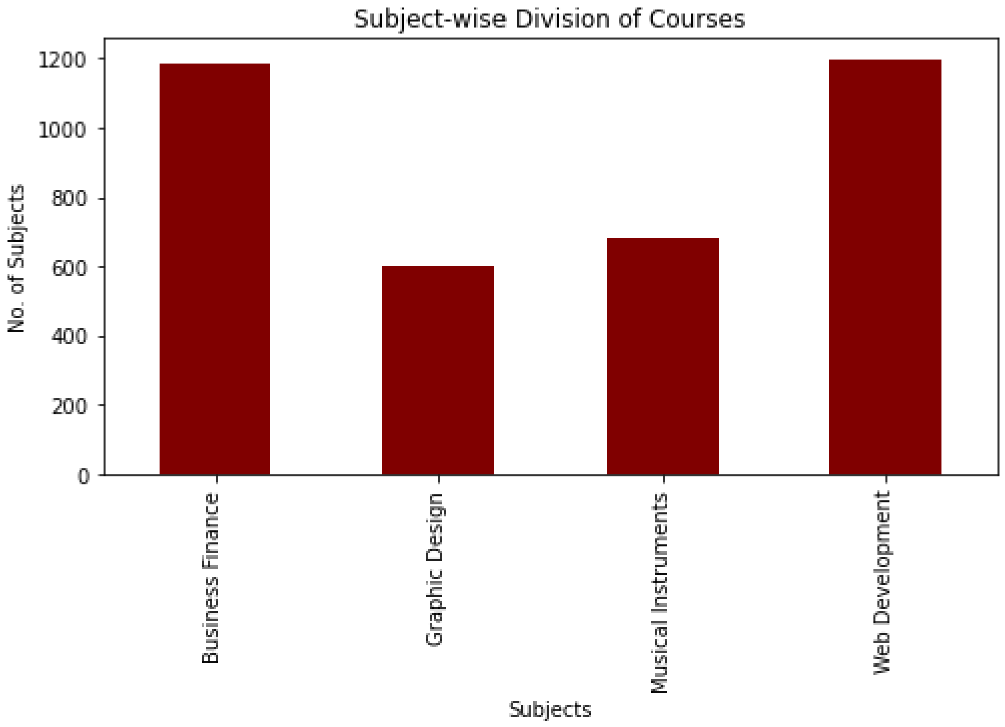
Plotting a histogram to show the number of courses which different subjects have -

In [47]:

```
course_count.plot.bar(figsize = (8, 4), color = 'maroon')
plt.xlabel('Subjects')
plt.ylabel('No. of Subjects')
plt.title('Subject-wise Division of Courses')
```

Out[47]:

```
Text(0.5, 1.0, 'Subject-wise Division of Courses')
```



- How many free courses are there for every subject

In [26]:

```
free_courses = df[df['price']==0].groupby(['subject'])['course_id'].count()
free_courses
```

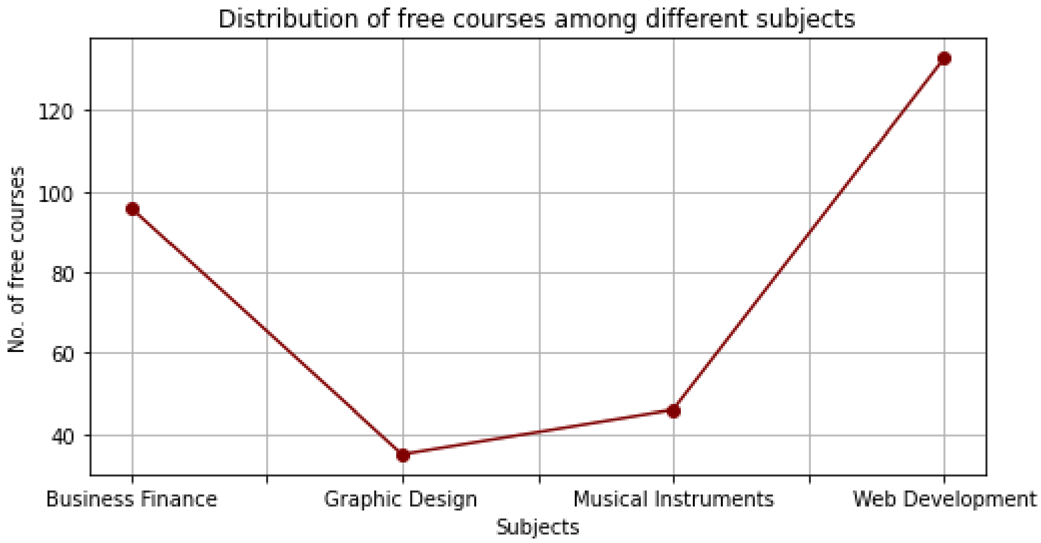
Out[26]:

```
subject
Business Finance        96
Graphic Design          35
Musical Instruments     46
Web Development        133
Name: course_id, dtype: int64
```

Plotting a graph to show the distribution of free courses among different subjects

In [29]:

```
free_courses.plot.line(figsize = (8, 4), marker = 'o', color = 'maroon')
plt.xlabel('Subjects')
plt.ylabel('No. of free courses')
plt.title('Distribution of free courses among different subjects')
plt.grid()
```



- Find the distribtuion of paid courses among different subjects

In [30]:

```
paid_courses = df[df['price']!=0].groupby(['subject'])['course_id'].count()
paid_courses
```

Out[30]:

```
subject
Business Finance       1091
Graphic Design          566
Musical Instruments     634
Web Development        1065
Name: course_id, dtype: int64
```

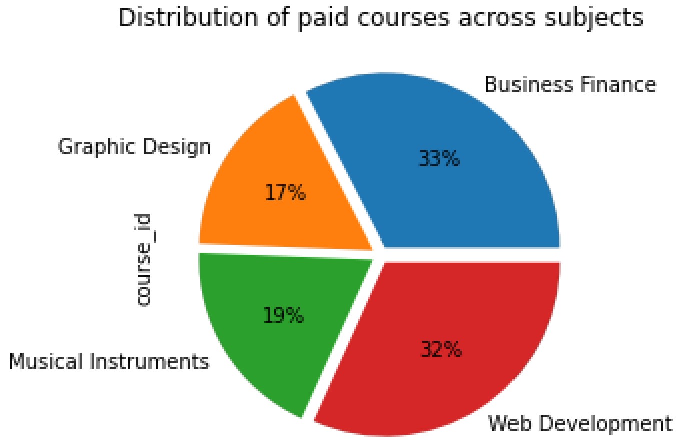
Plotting a graph to show distribution of paid courses among different subjects

In [38]:

```
paid_courses.plot.pie(figsize = (8, 4), autopct = '%0.f%%', explode = [0.05, 0.05, 0.05, 0.
plt.title('Distribution of paid courses across subjects')
```

Out[38]:

```
Text(0.5, 1.0, 'Distribution of paid courses across subjects')
```



- What are the most demanded subjects

In [43]:

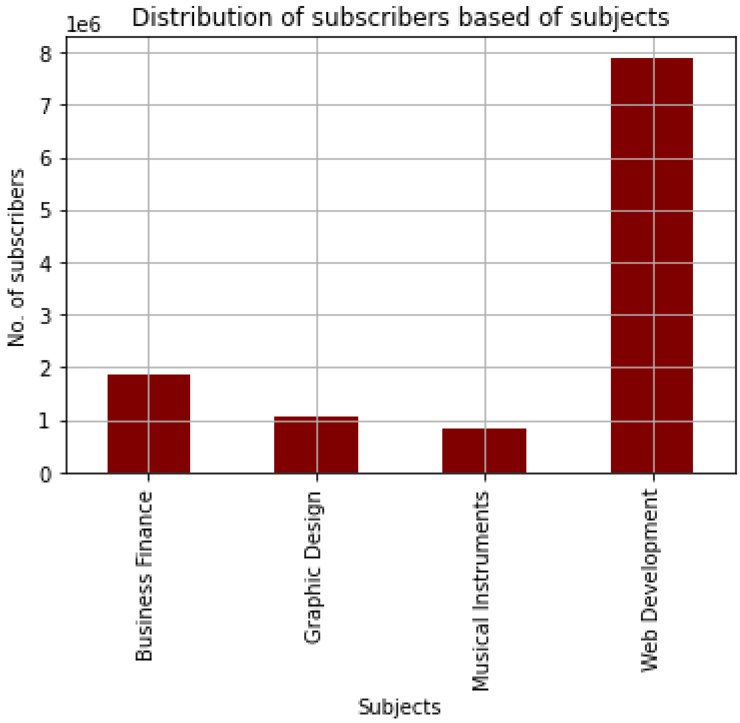
```
dem = df.groupby(['subject'])['num_subscribers'].sum()
dem
```

Out[43]:

```
subject
Business Finance       1868711
Graphic Design         1063148
Musical Instruments     846689
Web Development        7894002
Name: num_subscribers, dtype: int64
```

In [51]:

```
dem.plot.bar(color = 'maroon')
plt.xlabel('Subjects')
plt.ylabel('No. of subscribers')
plt.title('Distribution of subscribers based of subjects')
plt.grid()
```



- Year wise distribution of courses

In [48]:

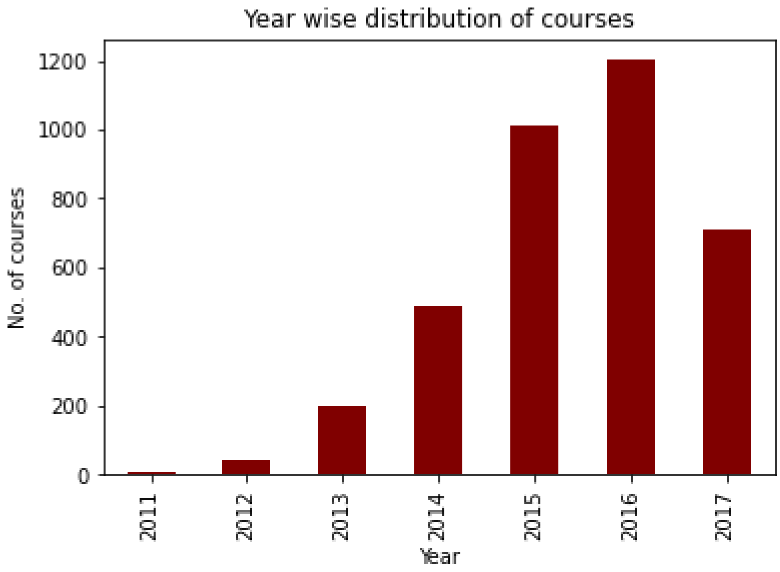
```
#creating year column
df['published_timestamp'] = pd.to_datetime(df['published_timestamp'])
df['year'] = df['published_timestamp'].dt.year
```

In [50]:

```
year_courses = df.groupby(['year'])['course_id'].count()
year_courses.plot.bar(color = 'maroon')
plt.xlabel('Year')
plt.ylabel('No. of courses')
plt.title('Year wise distribution of courses')
```

Out[50]:

```
Text(0.5, 1.0, 'Year wise distribution of courses')
```



- Distribution of subscribers across years

In [56]:

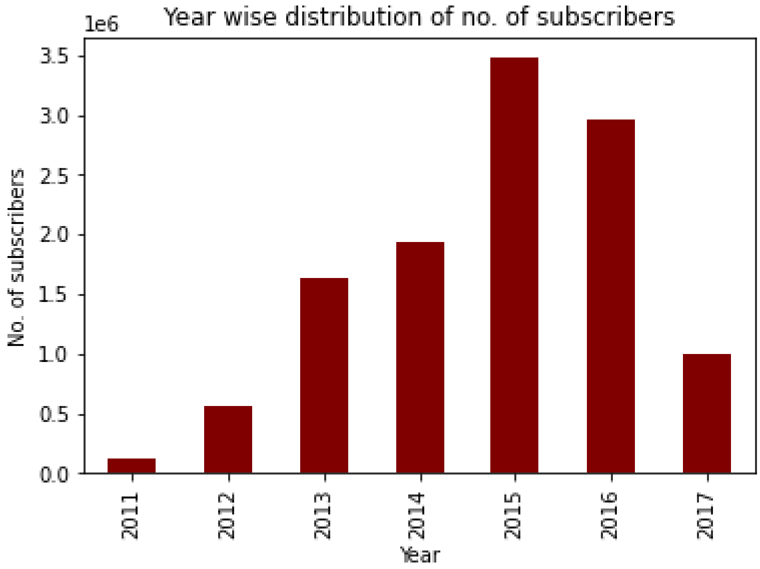
```
year_subs = df.groupby(['year'])['num_subscribers'].sum()
print(year_subs)

year_subs.plot.bar(color = 'maroon')
plt.xlabel('Year')
plt.ylabel('No. of subscribers')
plt.title('Year wise distribution of no. of subscribers')
```

```
year
2011     119028
2012     555339
2013    1636868
2014    1930406
2015    3475324
2016    2966644
2017     988941
Name: num_subscribers, dtype: int64
```

Out[56]:

```
Text(0.5, 1.0, 'Year wise distribution of no. of subscribers')
```



Distribution of subscribers across levels of courses

In [78]:

```
lev = df[df['level']!='All Levels'].groupby(['level'])['num_subscribers'].sum()
lev
```

Out[78]:

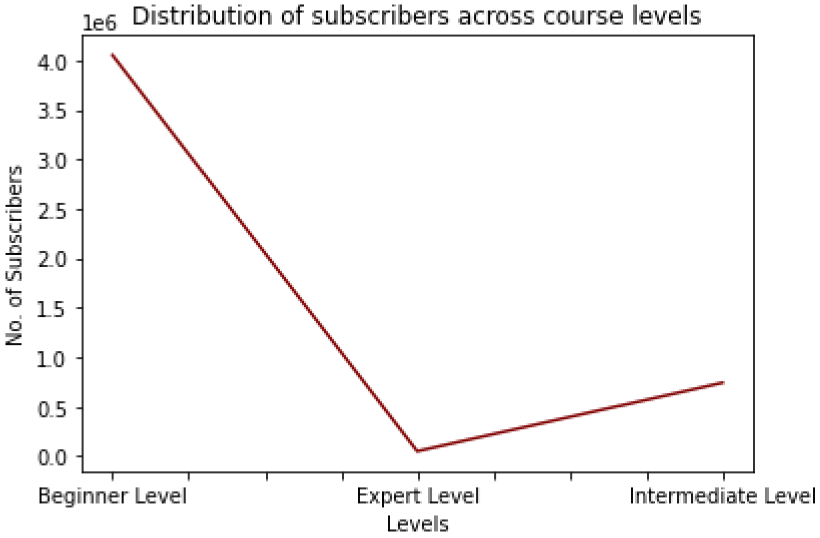
```
level
Beginner Level        4051843
Expert Level            50196
Intermediate Level     742005
Name: num_subscribers, dtype: int64
```

In [79]:

```
lev.plot.line(color = 'maroon')
plt.xlabel('Levels')
plt.ylabel('No. of Subscribers')
plt.title('Distribution of subscribers across course levels')
```

Out[79]:

```
Text(0.5, 1.0, 'Distribution of subscribers across course levels')
```



- Distribution of paid courses accross levels of courses

In [84]:

```
lev1 = df[df['level']!='All Levels'].groupby(['level'])['price'].count()
lev1
```

Out[84]:

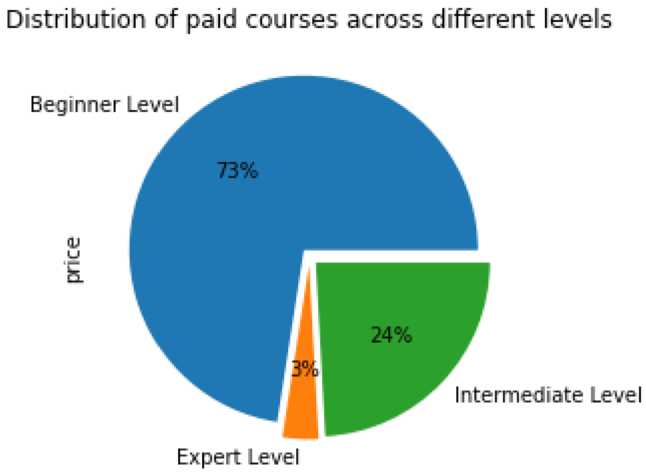
```
level
Beginner Level        1266
Expert Level            58
Intermediate Level     421
Name: price, dtype: int64
```

In [86]:

```
lev1.plot.pie(autopct = '%0.f%%', explode = [0.05, 0.05, 0.05])
plt.title('Distribution of paid courses across different levels')
```

Out[86]:

```
Text(0.5, 1.0, 'Distribution of paid courses across different levels')
```



- What is the relationship between price and number of lectures of courses

In [68]:

```
data = df[['price', 'num_lectures']].corr()
data
```

Out[68]:

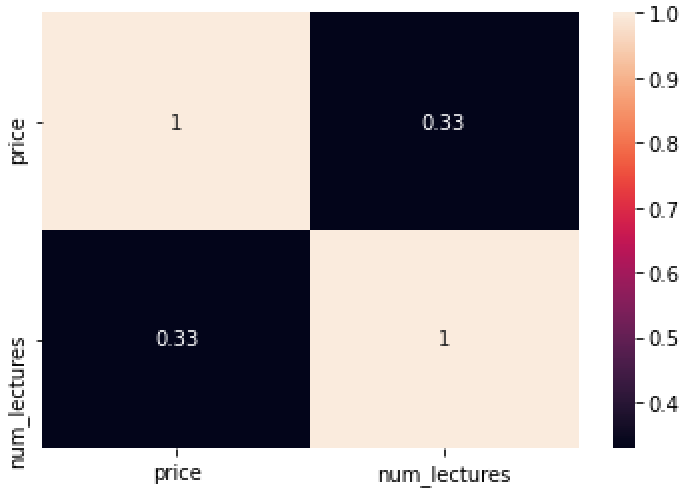
	price	num_lectures
price	1.000000	0.329727
num_lectures	0.329727	1.000000

In [66]:

```
sns.heatmap(data, annot = True)
```

Out[66]:

```
<AxesSubplot:>
```



The number of lectured do not depend completely on the price.

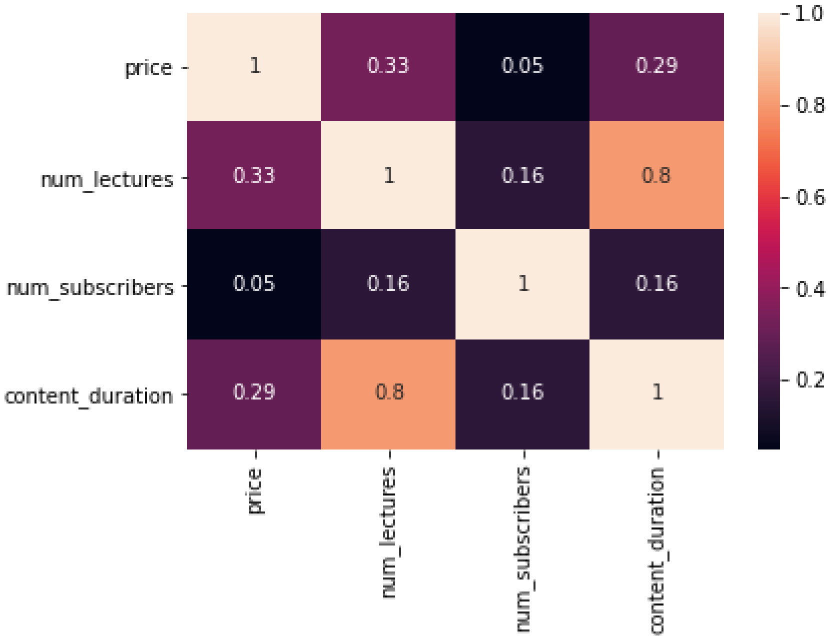
Relations between price, num of lecture, num of subscribers and content duration

In [77]:

```
subset = df[['price', 'num_lectures', 'num_subscribers', 'content_duration']].corr()
sns.heatmap(subset, annot = True)
```

Out[77]:

```
<AxesSubplot:>
```



Following can be observed from the heatmap -

- No. of subscribers have a very less dependency on the price of course
- No. of lectures and content duration are somewhat dependant on price
- No. of lectures don't make much difference on the no. of subscribers
- Content duration have a very little effect on no. of subscribers

Top 10 paid courses

In [87]:

```
top_paid = df[df['price']!=0][['course_title', 'subject', 'num_subscribers']].sort_values(b
top_paid
```

Out[87]:

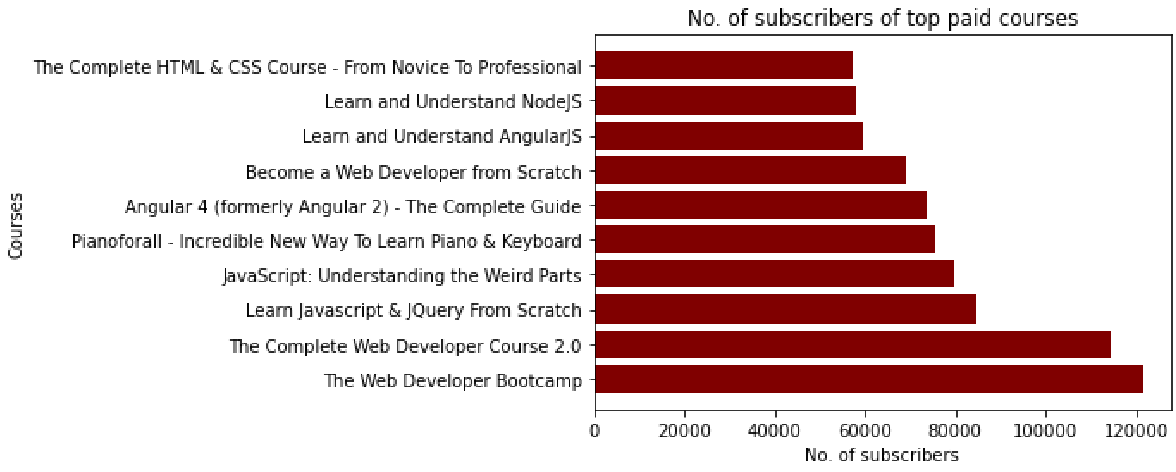
	course_title	subject	num_subscribers
3230	The Web Developer Bootcamp	Web Development	121584
3232	The Complete Web Developer Course 2.0	Web Development	114512
2619	Learn Javascript & JQuery From Scratch	Web Development	84897
3247	JavaScript: Understanding the Weird Parts	Web Development	79612
1979	Pianoforall - Incredible New Way To Learn Pian...	Musical Instruments	75499
3204	Angular 4 (formerly Angular 2) - The Complete ...	Web Development	73783
2701	Become a Web Developer from Scratch	Web Development	69186
3246	Learn and Understand AngularJS	Web Development	59361
3251	Learn and Understand NodeJS	Web Development	58208
2662	The Complete HTML & CSS Course - From Novice T...	Web Development	57422

In [92]:

```
plt.barh(top_paid['course_title'], top_paid['num_subscribers'], color = 'maroon')
plt.xlabel('No. of subscribers')
plt.ylabel('Courses')
plt.title('No. of subscribers of top paid courses')
```

Out[92]:

```
Text(0.5, 1.0, 'No. of subscribers of top paid courses')
```



- Top 10 free courses

In [93]:

```
top_free = df[df['price']==0][['course_title', 'subject', 'num_subscribers']].sort_values(b
top_free
```

Out[93]:

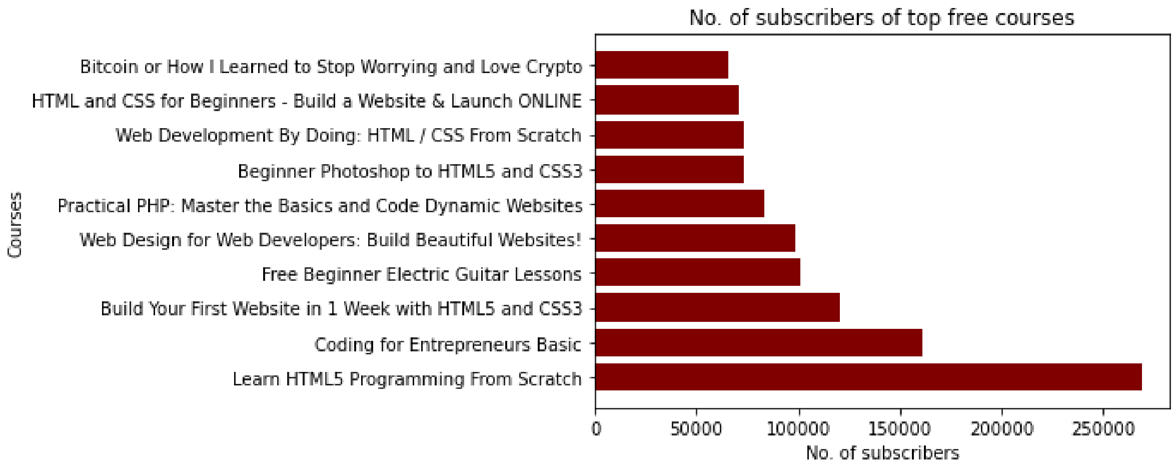
	course_title	subject	num_subscribers
2827	Learn HTML5 Programming From Scratch	Web Development	268923
3032	Coding for Entrepreneurs Basic	Web Development	161029
2783	Build Your First Website in 1 Week with HTML5 ...	Web Development	120291
1896	Free Beginner Electric Guitar Lessons	Musical Instruments	101154
2589	Web Design for Web Developers: Build Beautiful...	Web Development	98867
3289	Practical PHP: Master the Basics and Code Dyna...	Web Development	83737
3665	Beginner Photoshop to HTML5 and CSS3	Web Development	73110
2782	Web Development By Doing: HTML / CSS From Scratch	Web Development	72932
3325	HTML and CSS for Beginners - Build a Website &...	Web Development	70773
492	Bitcoin or How I Learned to Stop Worrying and ...	Business Finance	65576

In [94]:

```
plt.barh(top_free['course_title'], top_free['num_subscribers'], color = 'maroon')
plt.xlabel('No. of subscribers')
plt.ylabel('Courses')
plt.title('No. of subscribers of top free courses')
```

Out[94]:

```
Text(0.5, 1.0, 'No. of subscribers of top free courses')
```



In [95]:

```
top_reviewed = df[['course_title', 'subject', 'is_paid', 'num_reviews']].sort_values(by = '
top_reviewed
```

Out[95]:

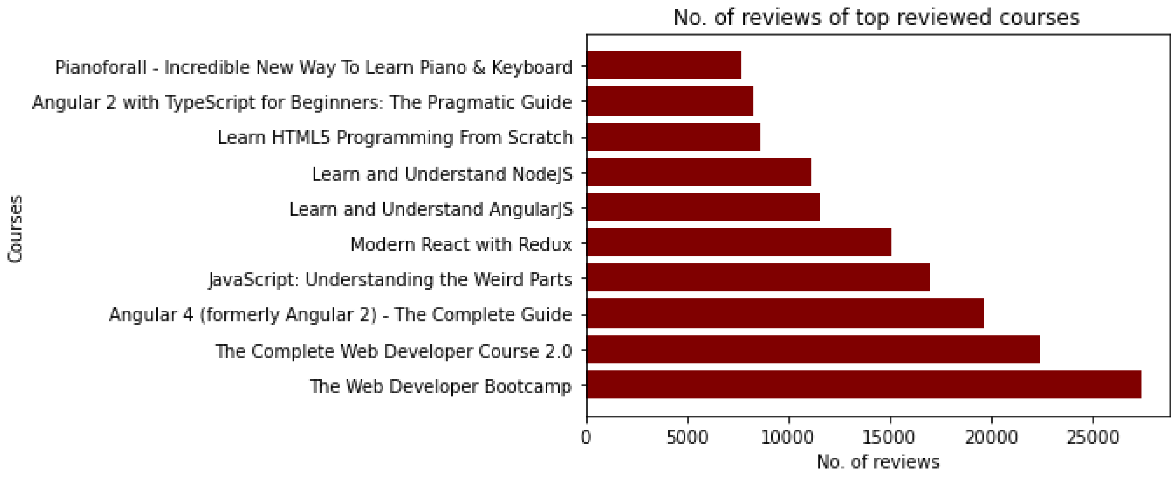
	course_title	subject	is_paid	num_reviews
3230	The Web Developer Bootcamp	Web Development	True	27445
3232	The Complete Web Developer Course 2.0	Web Development	True	22412
3204	Angular 4 (formerly Angular 2) - The Complete ...	Web Development	True	19649
3247	JavaScript: Understanding the Weird Parts	Web Development	True	16976
3254	Modern React with Redux	Web Development	True	15117
3246	Learn and Understand AngularJS	Web Development	True	11580
3251	Learn and Understand NodeJS	Web Development	True	11123
2827	Learn HTML5 Programming From Scratch	Web Development	False	8629
3228	Angular 2 with TypeScript for Beginners: The P...	Web Development	True	8341
1979	Pianoforall - Incredible New Way To Learn Pian...	Musical Instruments	True	7676

In [96]:

```
plt.barh(top_reviewed['course_title'], top_reviewed['num_reviews'], color = 'maroon')
plt.xlabel('No. of reviews')
plt.ylabel('Courses')
plt.title('No. of reviews of top reviewed courses')
```

Out[96]:

```
Text(0.5, 1.0, 'No. of reviews of top reviewed courses')
```



REFERENCES

1. Referred to the Book, “Python for Data Analysis”, 2nd Edition by Wes McKinney
2. Referred to Google for some basic queries
3. Referred to Python Documentation for some extra information relating to the project and Python Libraries