In [189]:

```
# Hypothesis Testing
```

Dataset Used : Coursera Course Dataset

URL : https://www.kaggle.com/datasets/siddharthm1698/coursera-course-dataset

Featured Engineered Data : https://www.kaggle.com/code/azminetoushikwasi/coursera-eda-prep-viz-fe-with-analytics-insights/notebook

Data Brief

Course dataset scrapped from Coursera website. This dataset contains mainly 6 columns and 890 course data. The detailed description:

1. **course_title : Contains the course title.**
2. **course_organization : It tells which organization is conducting the courses.**
3. **courseCertificatetype : It has details about what are the different certifications available in courses.**
4. **course_rating : It has the ratings associated with each course.**
5. **course_difficulty : It tells about how difficult or what is the level of the course.**
6. **coursestudentsenrolled : It has the number of students that are enrolled in the course.**

In [190]:

```
## Data import and Coorelation Matrix
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as sps

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

df=pd.read_csv("coursera_data_FEd.csv")
df=df.drop("Unnamed: 0",axis=1)
```

Sampling example

In [191]:

```
sample=df.sample(40,random_state=1)
sample.describe()
```

Out[191]:

	course_rating	course_students_enrolled_modified	course_difficulty_modified
count	40.000000	40.000000	40.000000
mean	4.675000	7.162798	0.325000
std	0.151488	8.651666	0.460629
min	4.200000	1.000000	0.000000
25%	4.600000	2.000000	0.000000
50%	4.700000	5.000000	0.000000
75%	4.800000	8.000000	0.500000

	course_rating	course_students_enrolled_modified	course_difficulty_modified
max	4.900000	38.000000	2.000000

Hypothesis Formulation

Hypothesis 01:

Null hypothesis: Equal or less than 50% enrolled courses are beginner level courses.

- **Test Method: z statstic**
- **Significance Level: 8%**

Hypothesis 02:

Null hypothesis: Coursera has a average course rating of more than 4.5.

Hypothesis 03:

Null hypothesis: University courses has more average rating by 0.2 from non-university courses.

Conducting a formal significance test for one of the hypotheses and discuss the results

Testing for Hypothesis 01:

Necessary Data

- **H_0: $\pi \leq 0.50$**
- **H_1: $\pi > 0.50$**
- **$\alpha = 0.08$**
- **Test Method: z statstic; $z = (p-\pi)/\sigma_p$, where $\sigma_p = sqrt(\pi(1-\pi)/n)$**

In [192]:

```
pi=0.5
sigma=0.08
```

Calculating P

In [193]:

```
sample_size=len(df)
sample_size
```

Out[193]:

```
891
```

so, total sample size = 891

In [194]:

```
# P, the value of sample statistic
positives= df[df['course_difficulty_modified']==0]['course_rating'].count()
positives
```

Out[194]:

```
487
```

number of courses with rating more than 4.5 = 745

In [195]:

```
P=positives/sample_size
P
```

Out[195]:

```
0.5465768799102132
```

Now, we will determine The value of σₚ, where σₚ=sqrt(π(1-π)/n)

In [196]:

```
import math

#defining meu_p function
def meu_p (pi,sample_size):
    temp=pi*(1-pi)/sample_size
    return math.sqrt(temp)
meu_p (pi,sample_size)
```

Out[196]:

```
0.016750630254320203
```

In [197]:

```
#defining z_statistic function

def z_stat(pi,p,sample_size):
    return (p-pi)/meu_p(pi,sample_size)
```

In [198]:

```
## Applying
z_stat(pi,P,sample_size)
```

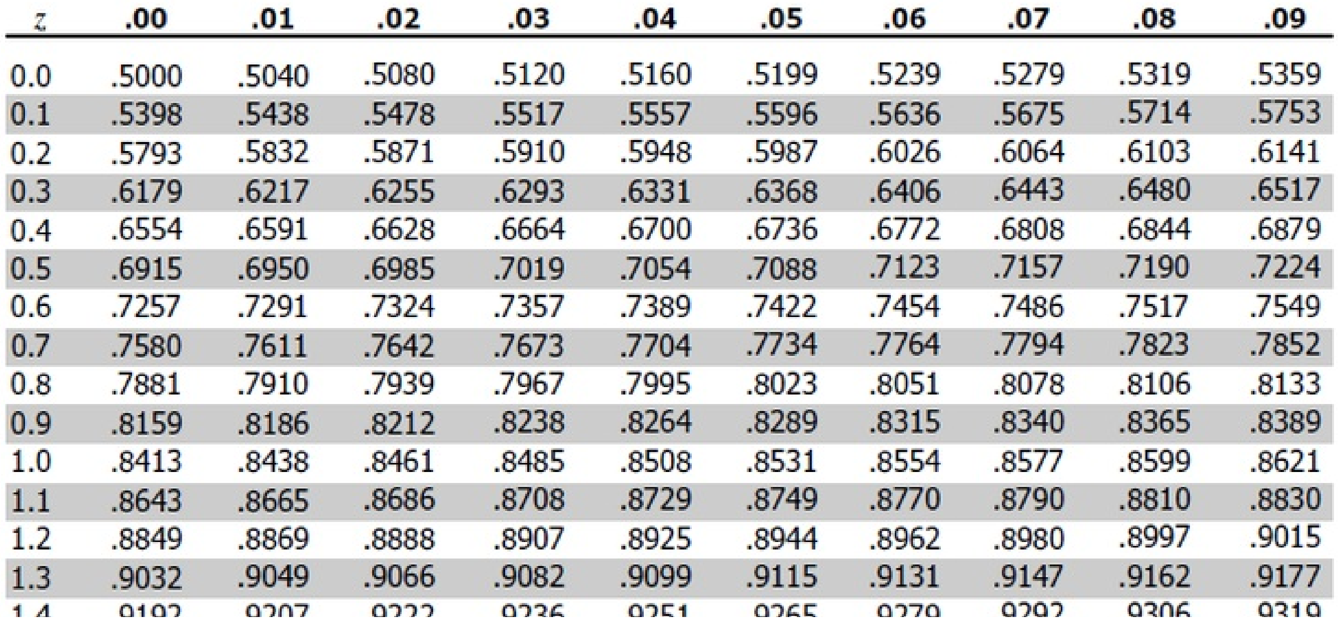
Out[198]:

```
2.7806046222171505
```

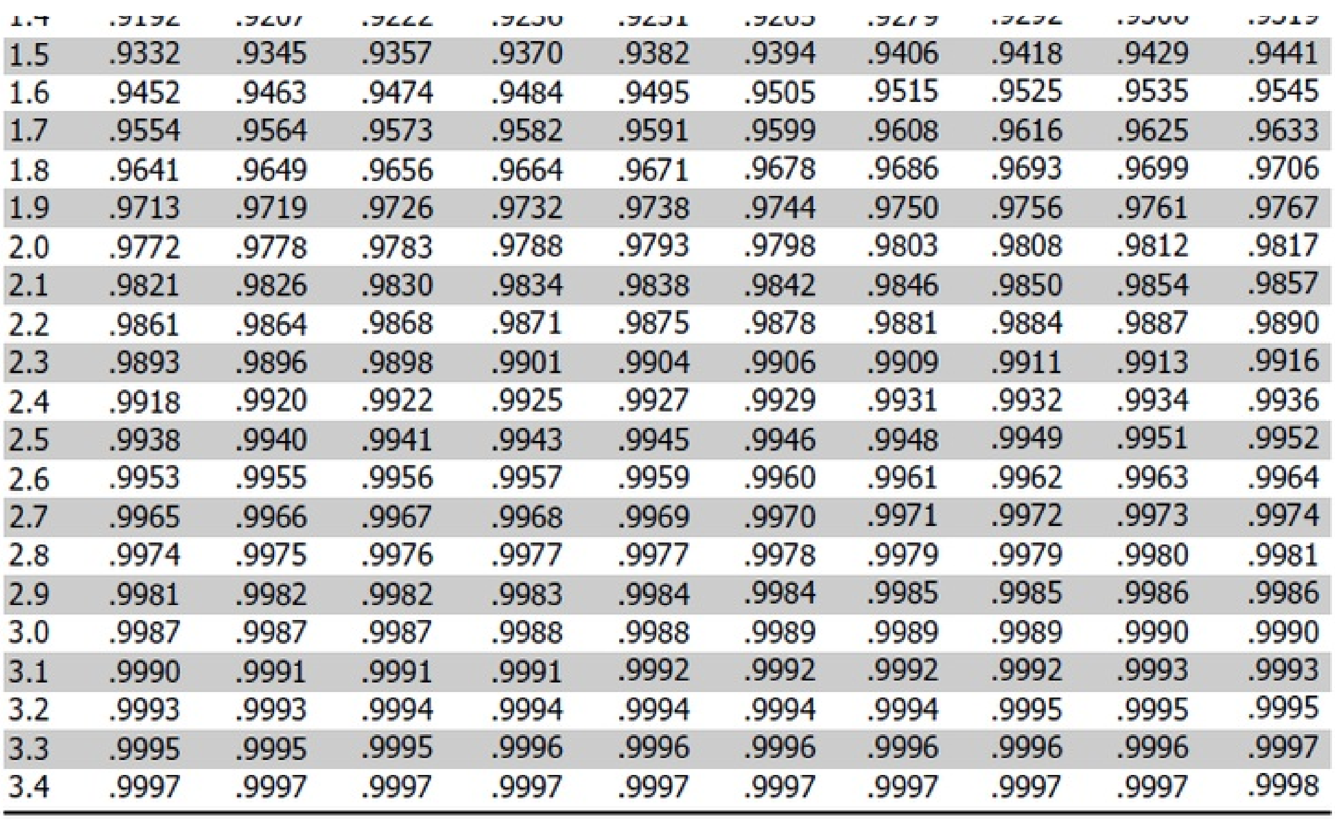
In [201]:

```
from IPython.display import Image
from IPython.core.display import HTML
Image(url= "http://www.z-table.com/uploads/2/1/7/9/21795380/8573955.png?759")
```

Out[201]:



z	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
0.0	.5000	.5040	.5080	.5120	.5160	.5199	.5239	.5279	.5319	.5359
0.1	.5398	.5438	.5478	.5517	.5557	.5596	.5636	.5675	.5714	.5753
0.2	.5793	.5832	.5871	.5910	.5948	.5987	.6026	.6064	.6103	.6141
0.3	.6179	.6217	.6255	.6293	.6331	.6368	.6406	.6443	.6480	.6517
0.4	.6554	.6591	.6628	.6664	.6700	.6736	.6772	.6808	.6844	.6879
0.5	.6915	.6950	.6985	.7019	.7054	.7088	.7123	.7157	.7190	.7224
0.6	.7257	.7291	.7324	.7357	.7389	.7422	.7454	.7486	.7517	.7549
0.7	.7580	.7611	.7642	.7673	.7704	.7734	.7764	.7794	.7823	.7852
0.8	.7881	.7910	.7939	.7967	.7995	.8023	.8051	.8078	.8106	.8133
0.9	.8159	.8186	.8212	.8238	.8264	.8289	.8315	.8340	.8365	.8389
1.0	.8413	.8438	.8461	.8485	.8508	.8531	.8554	.8577	.8599	.8621
1.1	.8643	.8665	.8686	.8708	.8729	.8749	.8770	.8790	.8810	.8830
1.2	.8849	.8869	.8888	.8907	.8925	.8944	.8962	.8980	.8997	.9015
1.3	.9032	.9049	.9066	.9082	.9099	.9115	.9131	.9147	.9162	.9177
1.4	.9192	.9207	.9222	.9236	.9251	.9265	.9279	.9292	.9306	.9319



1.4	.9192	.9207	.9222	.9236	.9251	.9265	.9279	.9292	.9306	.9319
1.5	.9332	.9345	.9357	.9370	.9382	.9394	.9406	.9418	.9429	.9441
1.6	.9452	.9463	.9474	.9484	.9495	.9505	.9515	.9525	.9535	.9545
1.7	.9554	.9564	.9573	.9582	.9591	.9599	.9608	.9616	.9625	.9633
1.8	.9641	.9649	.9656	.9664	.9671	.9678	.9686	.9693	.9699	.9706
1.9	.9713	.9719	.9726	.9732	.9738	.9744	.9750	.9756	.9761	.9767
2.0	.9772	.9778	.9783	.9788	.9793	.9798	.9803	.9808	.9812	.9817
2.1	.9821	.9826	.9830	.9834	.9838	.9842	.9846	.9850	.9854	.9857
2.2	.9861	.9864	.9868	.9871	.9875	.9878	.9881	.9884	.9887	.9890
2.3	.9893	.9896	.9898	.9901	.9904	.9906	.9909	.9911	.9913	.9916
2.4	.9918	.9920	.9922	.9925	.9927	.9929	.9931	.9932	.9934	.9936
2.5	.9938	.9940	.9941	.9943	.9945	.9946	.9948	.9949	.9951	.9952
2.6	.9953	.9955	.9956	.9957	.9959	.9960	.9961	.9962	.9963	.9964
2.7	.9965	.9966	.9967	.9968	.9969	.9970	.9971	.9972	.9973	.9974
2.8	.9974	.9975	.9976	.9977	.9977	.9978	.9979	.9979	.9980	.9981
2.9	.9981	.9982	.9982	.9983	.9984	.9984	.9985	.9985	.9986	.9986
3.0	.9987	.9987	.9987	.9988	.9988	.9989	.9989	.9989	.9990	.9990
3.1	.9990	.9991	.9991	.9991	.9992	.9992	.9992	.9992	.9993	.9993
3.2	.9993	.9993	.9994	.9994	.9994	.9994	.9994	.9995	.9995	.9995
3.3	.9995	.9995	.9995	.9996	.9996	.9996	.9996	.9996	.9996	.9997
3.4	.9997	.9997	.9997	.9997	.9997	.9997	.9997	.9997	.9997	.9998

Probability is approximately 0.998; But we wanted to calculate the probability to the right of z (because we are interested in obtaining the probability value that falls in the rejection region or critical region), i.e.

In [202]:

```
1-0.998
```

Out[202]:

```
0.0020000000000000018
```

Aplha is 0.05 So, the null hypothesis is rejected.

More than 50% students get enrolled in Beginner level courses.

Suggestions for next steps in analyzing this data

- **Testing other hypotheses.**
- **Analyze university based data.**
- **Try to group the courses to related subjects, based on subject name - keywords and see if any subject/field is performing better than others.**

The quality of this data set and a request for additional data if needed

- **Data quality is good, but data is not well distributed in various categories.**
- **The coirse-rating section is highly one-sided.**
- **Student enrollment number could be given in number, instead of string.**
- **Course length and these type info would have helped more.**

Data Request:

- **Require more data on some categories (advanced and so) to analyse far more better.**
- **More data means more accurate result. For a large platfrom like Cousera, we need more data and meta-data; like date-time of course launch, date of records and so on.**