

# Welcome to Colab!

## (New) Try the Gemini API

- [Generate a Gemini API key](#)
- [Talk to Gemini with the Speech-to-Text API](#)
- [Gemini API: Quickstart with Python](#)
- [Gemini API code sample](#)
- [Compare Gemini with ChatGPT](#)
- [More notebooks](#)

If you're already familiar with Colab, check out this video to learn about interactive tables, the executed code history view and the command palette.



Start coding or [generate](#) with AI.

## What is Colab?

Colab, or 'Colaboratory', allows you to write and execute Python in your browser, with

- Zero configuration required
- Access to GPUs free of charge
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch [Introduction to Colab](#) to find out more, or just get started below!

## ✓ Getting started

The document that you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable and prints the result:


```
seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
```

 86400

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut 'Command/Ctrl+Enter'. To edit the code, just click the cell and start editing.

Variables that you define in one cell can later be used in other cells:

```
seconds_in_a_week = 7 * seconds_in_a_day
seconds_in_a_week
```

 604800

Colab notebooks allow you to combine **executable code** and **rich text** in a single document, along with **images**, **HTML**, **LaTeX** and more. When you create your own Colab notebooks, they are stored in your Google Drive account. You can easily share your Colab notebooks with co-workers or friends, allowing them to comment on your notebooks or even edit them. To find out more, see [Overview of Colab](#). To create a new Colab notebook you can use the File menu above, or use the following link: [Create a new Colab notebook](#).

Colab notebooks are Jupyter notebooks that are hosted by Colab. To find out more about the Jupyter project, see [jupyter.org](#).

## ✓ Data science

With Colab you can harness the full power of popular Python libraries to analyse and visualise data. The code cell below uses **numpy** to generate some random data, and uses **matplotlib** to visualise it. To edit the code, just click the cell and start editing.

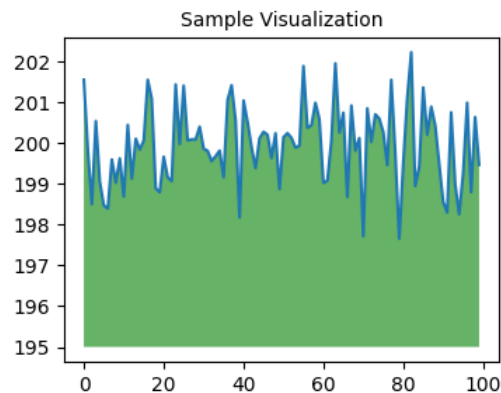
```
import numpy as np
import IPython.display as display
from matplotlib import pyplot as plt
import io
import base64
```

```
ys = 200 + np.random.randn(100)
x = [x for x in range(len(ys))]
```

```
fig = plt.figure(figsize=(4, 3), facecolor='w')
plt.plot(x, ys, '-')
plt.fill_between(x, ys, 195, where=(ys > 195), facecolor='g', alpha=0.6)
plt.title("Sample Visualization", fontsize=10)
```

```
data = io.BytesIO()
plt.savefig(data)
image = F"data:image/png;base64,{base64.b64encode(data.getvalue()).decode()}"
```

```
alt = "Sample Visualization"  
display.display(display.Markdown(F"!!! [{alt}]({image})"""))  
plt.close(fig)
```



You can import your own data into Colab notebooks from your Google Drive account, including from spreadsheets, as well as from GitHub and many other sources. To find out more about importing data, and how Colab can be used for data science, see the links below under [Working with data](#).

## ✓ Machine learning

With Colab you can import an image dataset, train an image classifier on it, and evaluate the model, all in just [a few lines of code](#). Colab notebooks execute code on Google's cloud servers, meaning you can leverage the power of Google hardware, including [GPUs and TPUs](#), regardless of the power of your machine. All you need is a browser.

Colab is used extensively in the machine learning community with applications including:

- Getting started with TensorFlow
- Developing and training neural networks
- Experimenting with TPUs
- Disseminating AI research
- Creating tutorials

To see sample Colab notebooks that demonstrate machine learning applications, see the [machine learning examples](#) below.

## ✓ More resources

Working with notebooks in Colab

- [Overview of Colaboratory](#)
- [Guide to markdown](#)
- [Importing libraries and installing dependencies](#)
- [Saving and loading notebooks in GitHub](#)
- [Interactive forms](#)
- [Interactive widgets](#)

## Working with data

- [Loading data: Drive, Sheets and Google Cloud Storage](#)
- [Charts: visualising data](#)
- [Getting started with BigQuery](#)

## Machine learning crash course

These are a few of the notebooks from Google's online machine learning course. See the [full course website](#) for more.

- [Intro to Pandas DataFrame](#)
- [Linear regression with tf.keras using synthetic data](#)

## Using accelerated hardware

- [TensorFlow with GPUs](#)
- [TensorFlow with TPUs](#)


## ✓ Featured examples



- [NeMo voice swap](#): Use Nvidia NeMo conversational AI toolkit to swap a voice in an audio fragment with a computer-generated one.
- [Retraining an Image Classifier](#): Build a Keras model on top of a pre-trained image classifier to distinguish flowers.
- [Text Classification](#): Classify IMDB film reviews as either *positive* or *negative*.
- [Style Transfer](#): Use deep learning to transfer style between images.
- [Multilingual Universal Sentence Encoder Q&A](#): Use a machine-learning model to answer questions from the SQuAD dataset.
- [Video Interpolation](#): Predict what happened in a video between the first and the last frame.

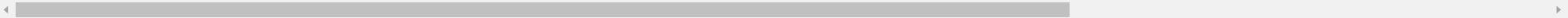
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
data=pd.read_excel('Universities.xlsx');
```

```
data.head()
```




	College Name	State	College_Type	Received_applications	Accepted_applications	Undergraduates	Graduation_rate	
0	Alaska Pacific University	AK	2	193	146	249	15	
1	University of Alaska Southeast	AK	1	146	117	492	39	
2	Alabama Agri. & Mech. Univ.	AL	1	2817	1920	3958	40	
3	Faulkner University	AL	2	345	320	1367	55	
4	University of Montevallo	AL	1	1351	892	2385	51	




Next steps: [Generate code with data](#) [View recommended plots](#) [New interactive sheet](#)

```
data.shape
```



```
(1193, 7)
```

```
data.dtypes
```



	0
College Name	object
State	object
College_Type	int64
Received_applications	int64
Accepted_applications	int64
Undergraduates	int64
Graduation_rate	int64

```
dtype: object
```



```
data.describe()
```



	College_Type	Received_applications	Accepted_applications	Undergraduates	Graduation_rate
<b>count</b>	1193.000000	1193.000000	1193.000000	1193.000000	1193.000000
<b>mean</b>	1.642917	2877.781224	1943.015088	3813.820620	60.633697
<b>std</b>	0.479341	3636.404064	2305.358434	4661.341213	18.799207
<b>min</b>	1.000000	35.000000	35.000000	59.000000	8.000000
<b>25%</b>	1.000000	740.000000	578.000000	988.000000	48.000000
<b>50%</b>	2.000000	1557.000000	1158.000000	1846.000000	60.000000
<b>75%</b>	2.000000	3500.000000	2446.000000	4772.000000	74.000000
<b>max</b>	2.000000	48094.000000	26330.000000	31643.000000	118.000000



```
print(data['College Name'].nunique())
```



1169

```
print(data['State'].nunique())
```



51

```
data['State'].unique()
```



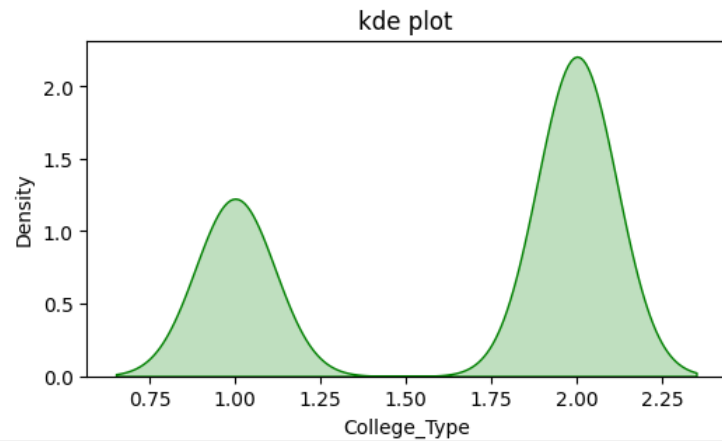
```
array(['AK', 'AL', 'AR', 'AZ', 'CA', 'CO', 'CT', 'DC', 'DE', 'FL', 'GA',
       'HI', 'IA', 'ID', 'IL', 'IN', 'KS', 'KY', 'LA', 'MA', 'MD', 'ME',
       'MI', 'MN', 'MO', 'MS', 'MT', 'NC', 'ND', 'NE', 'NH', 'NJ', 'NM',
       'NV', 'NY', 'OH', 'OK', 'OR', 'PA', 'RI', 'SC', 'SD', 'TN', 'TX',
       'UT', 'VA', 'VT', 'WA', 'WI', 'WV', 'WY'], dtype=object)
```

```
data['College Name'].unique()
```



```
array(['Alaska Pacific University', 'University of Alaska Southeast',
       'Alabama Agri. & Mech. Univ.', ...,
       'West Virginia Wesleyan College', 'Wheeling Jesuit College',
       'University of Wyoming'], dtype=object)
```

```
plt.figure(figsize=(20,18))
plt.subplot(5,3,1)
sns.kdeplot(data['College_Type'],color='green',fill =True)
plt.title('kde plot')
plt.show()
```



```
plt.figure(figsize=(20,18))
plt.subplot(5,3,1)
sns.kdeplot(data['College_Type'],color='blue',fill =True)
plt.title('kde plot')
plt.subplot(5,3,2)
sns.boxplot(data['College_Type'],color='green',orient='h')
plt.title('boxplot')
plt.subplot(5,3,3)
data['College_Type'].hist(grid=True,bins=10,color='orange',fill=True)
plt.title('Histogram')
```

```
plt.figure(figsize=(20,18))
plt.subplot(5,3,4)
sns.kdeplot(data['Received_applications'],color='blue',fill =True)
plt.title('kde plot')
plt.subplot(5,3,5)
sns.boxplot(data['Received_applications'],color='green',orient='h')
plt.title('boxplot')
plt.subplot(5,3,6)
data['Received_applications'].hist(grid=True,bins=10,color='orange',fill=True)
plt.title('Histogram')
```

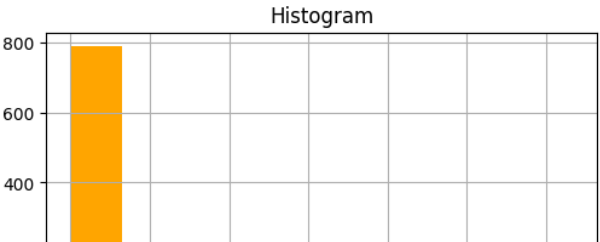
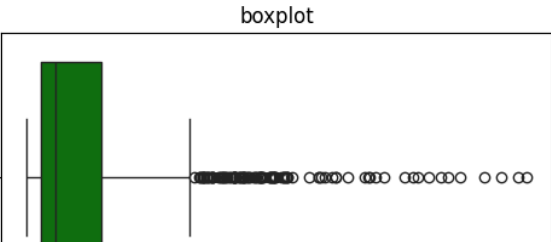
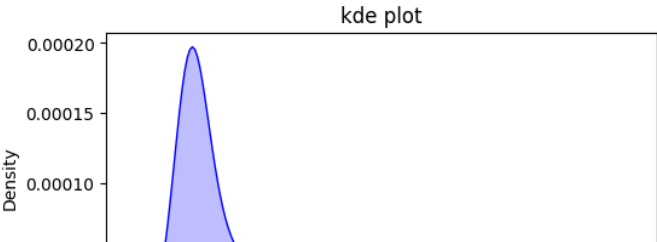
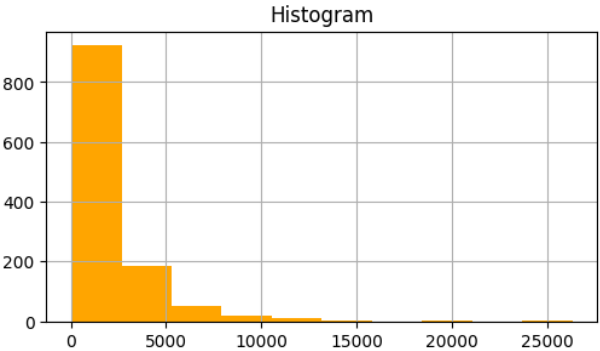
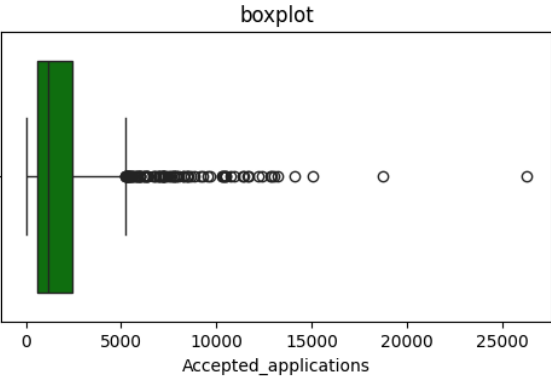
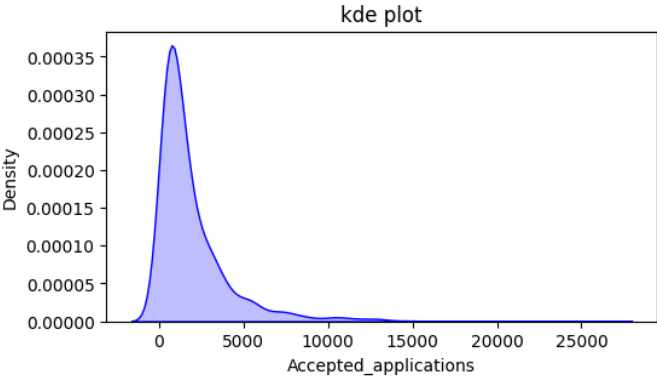
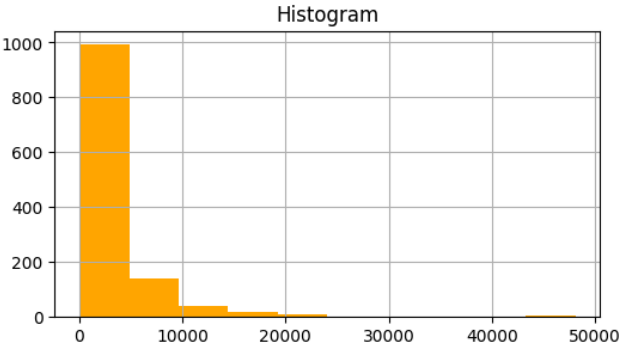
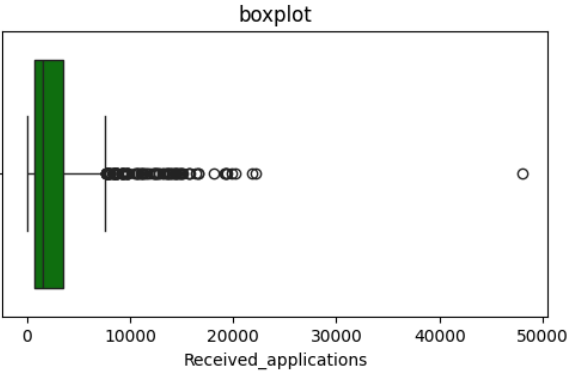
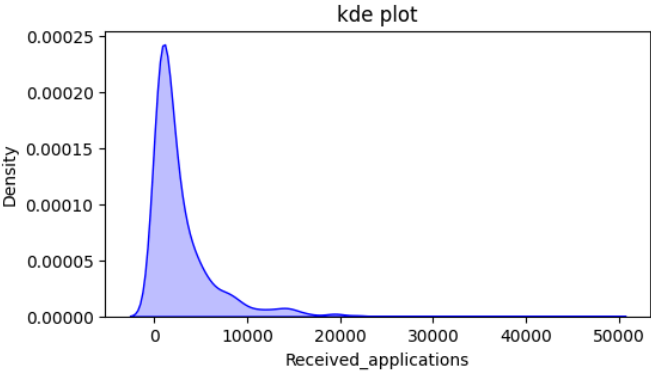
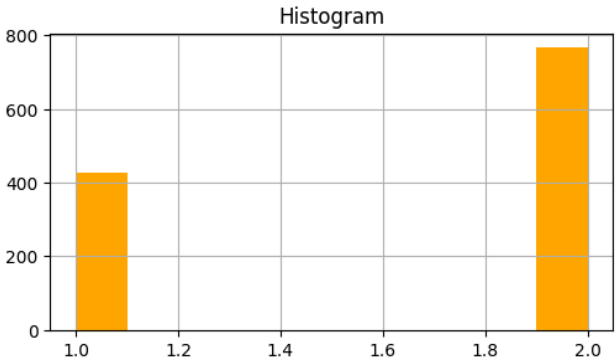
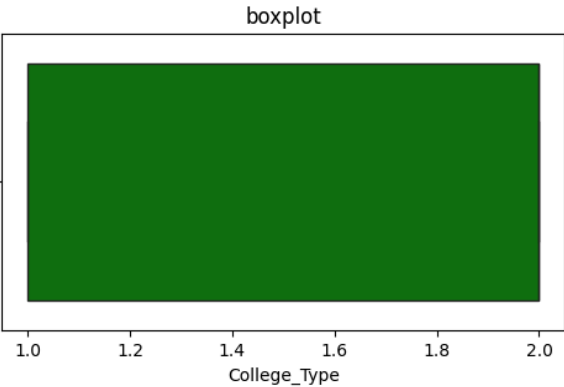
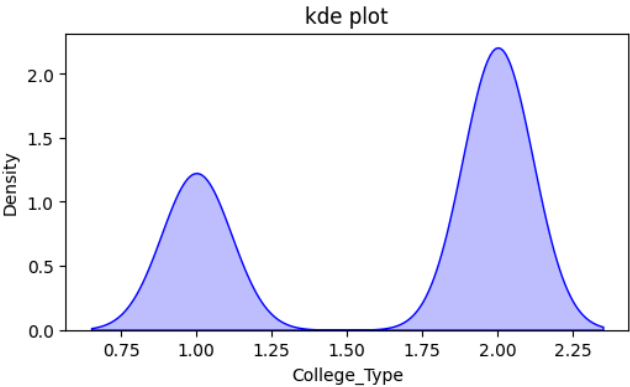
```
plt.figure(figsize=(20,18))
plt.subplot(5,3,7)
sns.kdeplot(data['Accepted_applications'],color='blue',fill =True)
plt.title('kde plot')
plt.subplot(5,3,8)
sns.boxplot(data['Accepted_applications'],color='green',orient='h')
plt.title('boxplot')
plt.subplot(5,3,9)
data['Accepted_applications'].hist(grid=True,bins=10,color='orange',fill=True)
plt.title('Histogram')
```

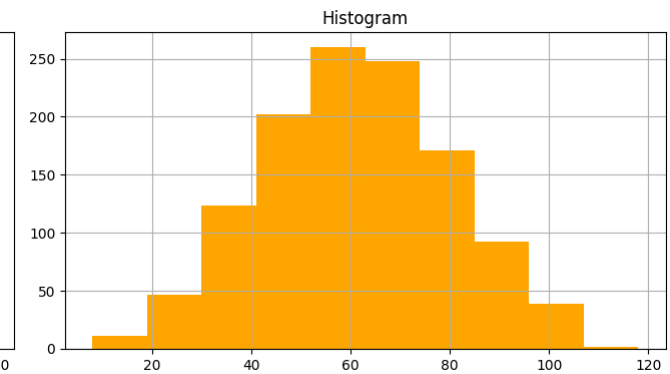
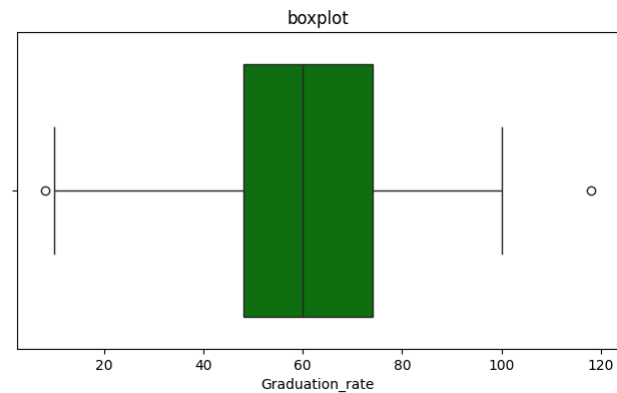
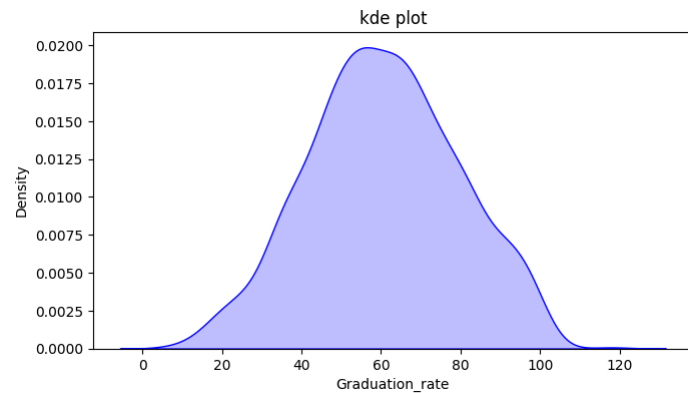
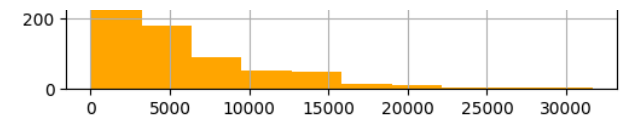
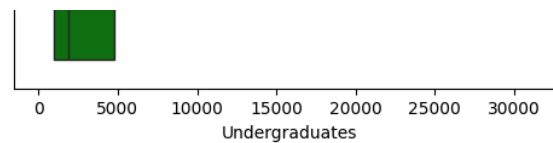
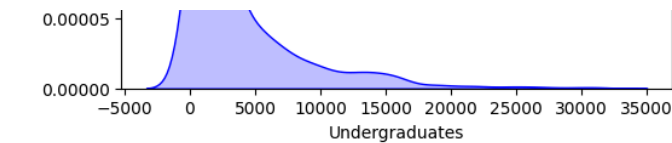
```
plt.figure(figsize=(20,18))
plt.subplot(5,3,10)
sns.kdeplot(data[ 'Undergraduates' ],color='blue',fill =True)
plt.title('kde plot')
plt.subplot(5,3,11)
sns.boxplot(data[ 'Undergraduates' ],color='green',orient='h')
plt.title('boxplot')
plt.subplot(5,3,12)
data[ 'Undergraduates' ].hist(grid=True,bins=10,color='orange',fill=True)
plt.title('Histogram')
```

```
plt.figure(figsize=(20,18))
plt.subplot(5,3,13)
sns.kdeplot(data[ 'Graduation_rate' ],color='blue',fill =True)
plt.title('kde plot')
plt.subplot(5,3,14)
sns.boxplot(data[ 'Graduation_rate' ],color='green',orient='h')
plt.title('boxplot')
plt.subplot(5,3,15)
data[ 'Graduation_rate' ].hist(grid=True,bins=10,color='orange',fill=True)
plt.title('Histogram')
```

```
plt.tight_layout()
plt.show()
```



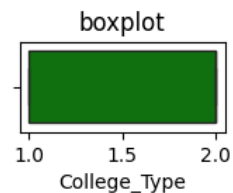




#

```
plt.subplot(5,3,2)
sns.boxplot(data['College_Type'],color='green',orient='h')
plt.title('boxplot')
```

Text(0.5, 1.0, 'boxplot')



#OBSERVATIONS :

```
#KDE_PLOT=
#1.KDE_PLOT OF FIRST HAVING TWO KDE PLOTS
#2,3,4 have same kde plots which are left skewed
#5 th have similar like normal distribution
```

```
# box-plot = box plot of college_type having Q1 and minimum are same and same for Q3
#box plot of SECOND having OUTLIERS MEANS MAX. VALUES ARE MORE
#box plot of THIRD AS SIMILAR AS SECOND
#Box plot of FOURTH having SO MANY OUTLIERS
#box plot of FIFTH HAVING VISCER AND ALSO HAVE SOME OUTLIERS ON LEFT HAND SIDE AND RIGHT SIDE
```

```
#HISTOGRAM=
```

```
#1.The histograms show a variety of distributions, including right-skewed, approximately normal, and binomial distributions for different variables.
```

```
data['College_Type']=np.where(data['College_Type']==2, 'Private', 'Public')
college=data['College_Type'].unique()
counts=data['College_Type'].value_counts()
plt.bar(college,counts,color='blue',edgecolor='green')
plt.title('countplot')
plt.ylabel('count')
plt.xlabel('College_Type')
plt.show()
```