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.

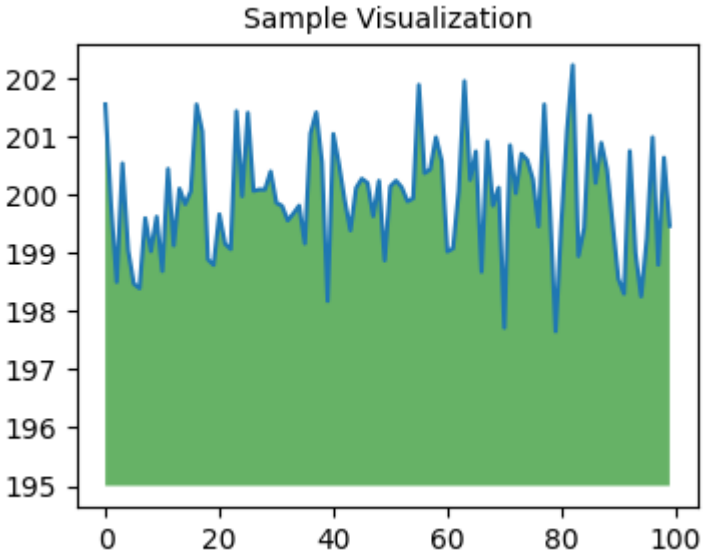
```
import numpy as np
import IPython.display as display
from matplotlib import pyplot as plt
import io
import base64

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"{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 pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder

df=pd.read_csv("student.csv")
df
df.isnull() ##false
series=pd.isnull(df["Math_Score"])
df[series]
df.notnull() #true
```

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placem
0	True	True	True	True	True	
1	True	True	True	True	True	
2	True	True	True	True	True	
3	True	True	True	True	True	
4	True	True	True	True	True	
5	True	True	True	True	True	
6	True	True	True	True	True	
7	True	True	True	True	True	
8	True	True	True	True	True	
9	True	True	True	True	True	
10	True	True	True	True	True	
11	True	True	True	True	True	
12	True	True	True	True	True	
13	True	True	True	True	True	
14	True	True	True	True	True	
15	True	True	True	True	True	
16	True	True	True	True	True	
17	True	True	True	True	True	
18	True	True	True	True	True	
19	True	True	True	True	True	
20	True	True	True	True	True	
21	True	True	True	True	True	
22	True	True	True	True	True	
23	True	True	True	True	True	
24	True	True	True	True	True	
25	True	True	True	True	True	
26	True	True	True	True	True	
27	True	True	True	True	True	
28	True	True	True	True	True	

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
```

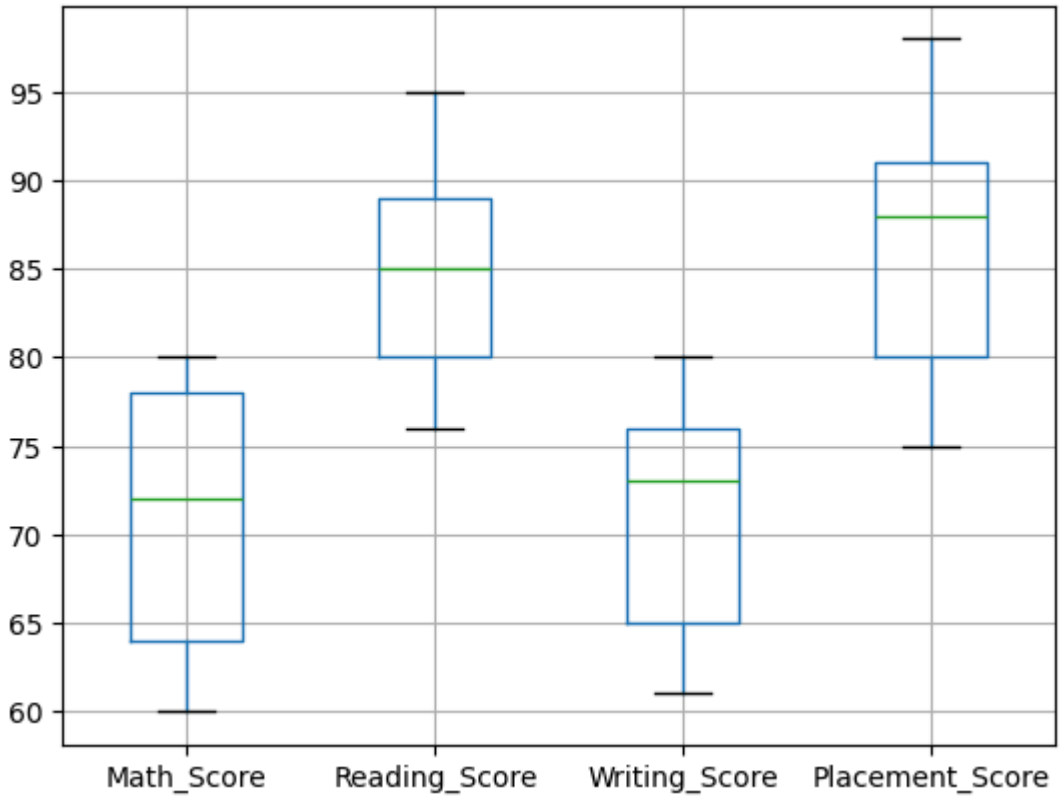
```
df=pd.read_csv("student.csv")
df
```

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placem
0	75	81	63	97	2021	
1	67	89	71	96	2020	
2	72	88	68	78	2019	
3	64	86	76	84	2020	
4	64	84	74	89	2018	
5	78	95	71	95	2019	
6	63	89	61	79	2021	
7	80	84	62	88	2019	
8	63	77	67	96	2020	
9	74	80	77	89	2019	
10	72	77	78	91	2019	
11	78	94	78	80	2021	
12	72	86	64	81	2018	
13	61	80	61	92	2021	
14	74	77	78	98	2019	
15	69	79	75	91	2020	
16	69	94	68	79	2018	
17	78	90	73	75	2019	
18	78	81	69	96	2019	
19	80	81	61	89	2020	
20	78	90	76	83	2021	
21	74	77	76	90	2020	
22	60	84	65	86	2020	
23	60	87	77	81	2020	
24	80	91	79	79	2021	
25	72	76	63	77	2020	
26	61	85	74	91	2019	
27	69	90	80	79	2019	
28	66	89	76	85	2020	

```
import pandas as pd
import numpy as np
```

```
df=pd.read_csv("student.csv")
df
col=['Math_Score','Reading_Score','Writing_Score','Placement_Score']
df.boxplot(col)
```

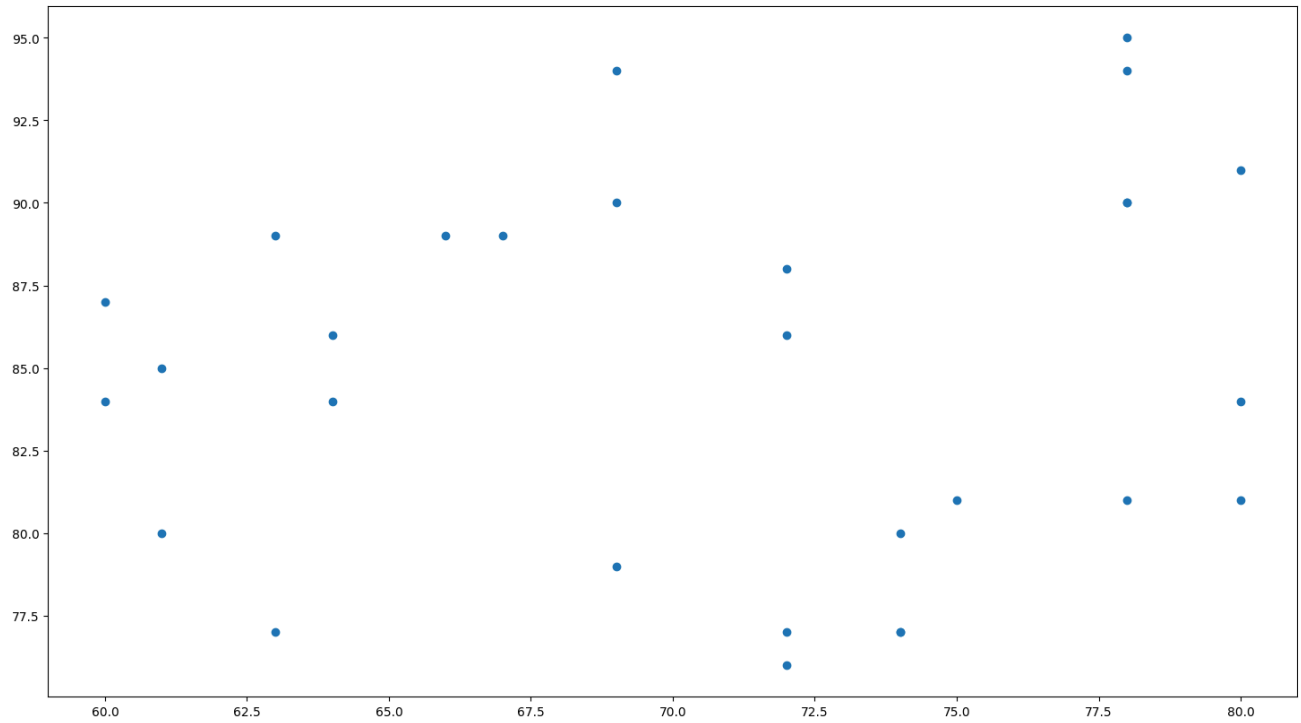
```
<Axes: >
```




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

df=pd.read_csv("student.csv")
df
fig,ax=plt.subplots(figsize=(18,10))
ax.scatter(df['Math_Score'],df['Reading_Score'])
plt.show()

print(np.where(df['Math_Score']<50))
print(np.where(df['Reading_Score']<50))
```




```
(array([], dtype=int64),)
(array([], dtype=int64),)
```

```
import pandas as pd
import numpy as np
from scipy import stats

df=pd.read_csv("student.csv")
df
z=np.abs(stats.zscore(df['Math_Score']))
print(z)
```

```
threshold=0.19
print(np.where(z>0.19))
```

```
0     0.650294
1     0.566385
2     0.194039
3     1.022640
4     1.022640
5     1.106549
6     1.174725
7     1.410718
8     1.174725
9     0.498209
10    0.194039
11    1.106549
12    0.194039
13    1.478894
14    0.498209
15    0.262215
16    0.262215
17    1.106549
18    1.106549
19    1.410718
20    1.106549
21    0.498209
22    1.630979
23    1.630979
24    1.410718
25    0.194039
26    1.478894
27    0.262215
28    0.718470
Name: Math_Score, dtype: float64
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28]),)
```

```
import pandas as pd

df = pd.read_csv("student.csv")


columns_to_bin = ['Math_Score', 'Reading_Score', 'Writing_Score', 'Placement_Score', 'Pla


df[columns_to_bin] = df[columns_to_bin].fillna(df[columns_to_bin].median())


for col in columns_to_bin:
    try:

        df[f'{col}_Binned'] = pd.qcut(df[col], q=4, labels=['Q1', 'Q2', 'Q3', 'Q4'], dupl
    except ValueError as e:

        print(f"Warning: qcut failed for {col}, Error: {e}. Skipping binning for this col
```

```
print("DataFrame with Binned Columns:")
print(df)
```

```
Warning: qcut failed for Placement_Offer_Count, Error: Bin labels must be one fewe
DataFrame with Binned Columns:
    Math_Score  Reading_Score  Writing_Score  Placement_Score  Club_Join_Date  \
0           75             81             63               97            2021
1           67             89             71               96            2020
2           72             88             68               78            2019
3           64             86             76               84            2020
4           64             84             74               89            2018
5           78             95             71               95            2019
6           63             89             61               79            2021
7           80             84             62               88            2019
8           63             77             67               96            2020
9           74             80             77               89            2019
10          72             77             78               91            2019
11          78             94             78               80            2021
12          72             86             64               81            2018
13          61             80             61               92            2021
14          74             77             78               98            2019
15          69             79             75               91            2020
16          69             94             68               79            2018
17          78             90             73               75            2019
18          78             81             69               96            2019
```