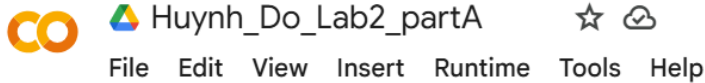


Huynh Do Lab#2 Part A:

Objective: This lab assignment exposes student to working with null values, which is data that is missing or is not known within a data set.



1. Import libraries

```
✓ [3] import pandas as pd
0s    import numpy as np
      from matplotlib import pyplot as plt
      import seaborn as sns # Optional for better visuals
      from google.colab import files
```

The code above imports several Python libraries commonly used in data analysis.

1. **import pandas as pd**
 - A library for data manipulation and analysis (e.g., working with tables, CSVs, etc.).
2. **import numpy as np :**
 - Support for numerical operations and arrays.
3. **from matplotlib import pyplot as plt**
 - Creating static, interactive, and animated plots.
4. **import seaborn as sns**
 - Imports **Seaborn** provides prettier and more informative plots.
5. **from google.colab import files**
 - Used in **Google Colab** to upload/download files from your local system within a Colab notebook.

2. Upload file NBA.csv

```
✓ [4] # Upload 'NBA.csv' file
10s   uploaded = files.upload()

      # Load the dataset
      df = pd.read_csv("NBA.csv")
      df.head()
```

The above screen shot is used to upload and load a CSV file named **NBA.csv** into a Pandas Data Frame in a Google Colab environment.

When upload is done:

Choose Files NBA.csv

- **NBA.csv**(text/csv) - 29636 bytes, last modified: 4/9/2025 - 100% done

Saving NBA.csv to NBA.csv

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	6-2	180	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99	SF	25	6-6	235	Marquette	6796117.0
2	John Holland	Boston Celtics	30	SG	27	6-5	205	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28	SG	22	6-5	185	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8	PF	29	6-10	231	NaN	5000000.0

3. Process Null Values

- **Step 1: Explore null values**

python

```
print("Null values in each column:\n")
print(df.isnull().sum())
```

- `df.isnull()` creates a DataFrame of the same shape as `df`, where each cell is True if it's null, otherwise False.
- `.sum()` adds up all the True values in each column (since True = 1), giving us the number of missing values per column.

- **Step 2: Fill with Null values**

```
# Step 2: Fill null values
# Example: Fill null numeric values with 0, and object types with "Unknown"
df_filled = df.fillna({
    col: 0 if df[col].dtype in ['int64', 'float64'] else "Unknown" for col in df.columns
})
```

The code above fills missing (NaN) values in a DataFrame (`df`) by assigning different fill values depending on each column's data type:

- If a column is numeric (int64 or float64), it fills nulls with 0
- Otherwise (usually text, i.e., object), it fills nulls with "Unknown"

In summary: `df_filled` is a copy of the original `df`, but with all NaN values replaced: Numbers → 0 | Strings → "Unknown"

df.fillna({...}) : Uses that dictionary to fill null values column by column, using the specified defaults.

- **Step 3:** check if all null values are gone

```
# Step 3: Check if all null values are gone
print("\nDataset info after filling nulls:\n")
df_filled.info()
```

This gives us a summary of the DataFrame after 've filled the missing values.

Specifically, it shows:

1. Number of entries (rows)
2. Column names
3. Non-null count for each column — want these to match the total row count if all nulls are gone
4. Data types of each column (e.g., int64, float64, object)
5. Memory usage

The below is result of calling **df_filled.info()** function

Null values in each column:

```
Name      0
Team      0
Number    0
Position  0
Age       0
Height    0
Weight    0
College   84
Salary    11
dtype: int64
```

Dataset info after filling nulls:

Dataset info after filling nulls:

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 457 entries, 0 to 456  
Data columns (total 9 columns):  
#   Column      Non-Null Count  Dtype  
---  -  
0   Name        457 non-null   object  
1   Team        457 non-null   object  
2   Number      457 non-null   int64  
3   Position    457 non-null   object  
4   Age         457 non-null   int64  
5   Height      457 non-null   object  
6   Weight      457 non-null   int64  
7   College     457 non-null   object  
8   Salary      457 non-null   float64  
dtypes: float64(1), int64(3), object(5)  
memory usage: 32.3+ KB
```
