Name = Saif Ul Mateen Emaill = ulmateen@gmail.com Whatsapp = +923032177002

This Notebook is About Practicing Different Functions of Pandas Libraries and Plotting Graphs With That Data

In [89]:
```
#Importing Libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [91]:
```
# Creating Simple Object/Cells
sc = pd.Series([1,2,3,4,5])
sc
```

Out[91]:
```
0    1
1    2
2    3
3    4
4    5
dtype: int64
```

In [92]:
```
# Creating Complete Table
sc2 = pd.DataFrame({
    "Student_Name" : pd.Categorical(["Haris" , "Anil", "Hamza", "Hashim", "Hasiq","Haris" , "Anil", "Hamza", "Hashim", "Hasiq"]),
    "Marks Out Of 1000" : pd.Series([950, 890, 940, 960, 860,950, 890, 940, 960, 860]),
    "%" : pd.Series([60, 70, 88, 49, 63,99, 50, 78, 89, 93]),
    "Comments" : pd.Categorical(["Good", "Excellent", "WellDone", "Great", "Great","Good", "Excellent", "WellDone", "Great", "Great"])
})

sc2
```

Out[92]:

	Student_Name	Marks Out Of 1000	%	Comments
0	Haris	950	60	Good
1	Anil	890	70	Excellent
2	Hamza	940	88	WellDone
3	Hashim	960	49	Great
4	Hasiq	860	63	Great
5	Haris	950	99	Good
6	Anil	890	50	Excellent
7	Hamza	940	78	WellDone
8	Hashim	960	89	Great
9	Hasiq	860	93	Great

In [93]:
```
# Checking Data Type
sc2.dtypes
```

Out[93]:
```
Student_Name          category
Marks Out Of 1000        int64
%                        int64
Comments              category
dtype: object
```

In [94]:
```
# Picking First N Values Of Data
sc2.head(2)
```

Out[94]:

	Student_Name	Marks Out Of 1000	%	Comments
0	Haris	950	60	Good
1	Anil	890	70	Excellent

In [95]:
```
# Picking Last N Values Of Data
sc2.tail(2)
```

Out[95]:

	Student_Name	Marks Out Of 1000	%	Comments
8	Hashim	960	89	Great
9	Hasiq	860	93	Great

In [96]:
```
sc2.index
```

Out[96]:
```
RangeIndex(start=0, stop=10, step=1)
```

In [97]:
```
#Converting Data to Array
    sc2.to_numpy()
```

Out[97]:
```
array([['Haris', 950, 60, 'Good'],
       ['Anil', 890, 70, 'Excellent'],
       ['Hamza', 940, 88, 'WellDone'],
       ['Hashim', 960, 49, 'Great'],
       ['Hasiq', 860, 63, 'Great'],
       ['Haris', 950, 99, 'Good'],
       ['Anil', 890, 50, 'Excellent'],
       ['Hamza', 940, 78, 'WellDone'],
       ['Hashim', 960, 89, 'Great'],
       ['Hasiq', 860, 93, 'Great']], dtype=object)
```

In [98]:
```
# Checking Mean, Median Etc..
sc2.describe()
```

Out[98]:

	Marks Out Of 1000	%
count	10.000000	10.000000
mean	920.000000	73.900000
std	40.55175	18.125795
min	860.000000	49.000000
25%	890.000000	60.750000
50%	940.000000	74.000000
75%	950.000000	88.750000
max	960.000000	99.000000

In [99]:
```
# Transposing Data
sc2.T
```

Out[99]:

	0	1	2	3	4	5	6	7	8	9
Student_Name	Haris	Anil	Hamza	Hashim	Hasiq	Haris	Anil	Hamza	Hashim	Hasiq
Marks Out Of 1000	950	890	940	960	860	950	890	940	960	860
%	60	70	88	49	63	99	50	78	89	93
Comments	Good	Excellent	WellDone	Great	Great	Good	Excellent	WellDone	Great	Great

In [100]:
```
# Sorting Data By Index
sc2.sort_index(axis=1, ascending=True)
```

Out[100]:

	%	Comments	Marks Out Of 1000	Student_Name
0	60	Good	950	Haris
1	70	Excellent	890	Anil
2	88	WellDone	940	Hamza
3	49	Great	960	Hashim
4	63	Great	860	Hasiq
5	99	Good	950	Haris
6	50	Excellent	890	Anil
7	78	WellDone	940	Hamza
8	89	Great	960	Hashim
9	93	Great	860	Hasiq

In [101]:
```
sc2.sort_index(axis=0, ascending=True)
```

Out[101]:

	Student_Name	Marks Out Of 1000	%	Comments
0	Haris	950	60	Good
1	Anil	890	70	Excellent
2	Hamza	940	88	WellDone
3	Hashim	960	49	Great
4	Hasiq	860	63	Great
5	Haris	950	99	Good
6	Anil	890	50	Excellent
7	Hamza	940	78	WellDone
8	Hashim	960	89	Great
9	Hasiq	860	93	Great

In [102]:
```
# Sorting Values By Values
sc2.sort_values(by="Marks Out Of 1000", ascending=True)
```

Out[102]:

	Student_Name	Marks Out Of 1000	%	Comments
4	Hasiq	860	63	Great
9	Hasiq	860	93	Great
1	Anil	890	70	Excellent
6	Anil	890	50	Excellent
2	Hamza	940	88	WellDone
7	Hamza	940	78	WellDone
0	Haris	950	60	Good
5	Haris	950	99	Good
3	Hashim	960	49	Great
8	Hashim	960	89	Great

In [103]:
```
# Selecting Different Values
sc2["%"]
```

Out[103]:
```
0    60
1    70
2    88
3    49
4    63
5    99
6    50
7    78
8    89
9    93
Name: %, dtype: int64
```

In [104]:
```
# Selecting Data Row Wise
sc2[4:5]
```

Out[104]:

	Student_Name	Marks Out Of 1000	%	Comments
4	Hasiq	860	63	Great
5	Haris	950	99	Good

In [105]:
```
# Selecting Data By Labels
sc2.loc[:, ["Student_Name"]]
```

Out[105]:

	Student_Name
0	Haris
1	Anil
2	Hamza
3	Hashim
4	Hasiq
5	Haris
6	Anil
7	Hamza
8	Hashim
9	Hasiq

In [106]:
```
sc2.loc[[0,2], ["%", "Comments"]]
```

Out[106]:

	%	Comments
0	60	Good
2	88	WellDone

In [107]:
```
sc2.at[2, "%"]
```

Out[107]:
```
88
```

In [108]:
```
sc2.iloc[3]
```

Out[108]:
```
Student_Name          Hashim
Marks Out Of 1000        960
%                         49
Comments                Great
Name: 3, dtype: object
```

Difference Between ***iLoc*** *and* ***LOC*** *is that* ***LOC*** *is Labels based function, You have to specify Rows and Columns Based on their Labels. On the other hand,* ***ILOC*** *is integer based function, You have to Specify Rows and Columns by their Integer Position*

In [109]:
```
# Specifying Both Rows and Columns
sc2.iloc[0:5, 0:1]
```

Out[109]:

	Student_Name
0	Haris
1	Anil
2	Hamza
3	Hashim
4	Hasiq

In [110]:
```
sc2[sc2["%"] > 80]
```

Out[110]:

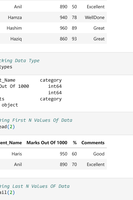
	Student_Name	Marks Out Of 1000	%	Comments
2	Hamza	940	88	WellDone
5	Haris	950	99	Good
8	Hashim	960	89	Great
9	Hasiq	860	93	Great

In [111]:
```
# Adding a New Column
sc2["Year of Passing"] = [2020, 2019, 2018, 2017, 2016, 2015, 2014, 2013, 2012, 2011]
```

Practicing Some Graphs On This Data

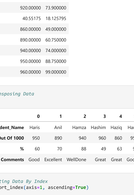
In [125]:
```
gp1 = sns.barplot(x="Student_Name", y="Marks Out Of 1000", data = sc2, palette="hls")
```

In [123]:
```
sns.set_theme(style="ticks", color_codes=True)
sns.set_style("darkgrid")
gp2 = sns.lineplot(x="Year of Passing", y="%", data=sc2)
```



In [126]:
```
gp3 = sns.barplot(x="Marks Out Of 1000", y="Student_Name", data = sc2, palette="hls")
```

In [129]:
```
gp4 = sns.boxplot(x="Year of Passing", y="%", data=sc2)
```



In [ ]:
```

```