



Cloud Computing Lab Project

BY: SHIVARAMAN R - RA1911042010008
NIKHIL UNNIKRIISHNAN - RA1911042010012
PAVAN NIKHIL YETURU - RA1911042010040



Data Representation in Python

Dataset instance is more precisely a table with rows and columns in which the columns are typed. Querying an example (a single row) will thus return a python dictionary with keys corresponding to column names, and values corresponding to the example's value for each column.

To explain it more clearly,

A data set is a collection of numbers or values that relate to a particular subject. For example, the test scores of each student in a particular class is a data set. The number of fish eaten by each dolphin at an aquarium is a data set.



OUR DATA SET

- Student list with their GPA's of their 1st and 4th year
- 3 columns

1st - Student Name

2nd - 1st year GPA

3rd - 4th year GPA

5 Rows

[illegible]




CODE:


```
#@title Default title text

import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
import warnings; warnings.filterwarnings(action='once')

large = 22; med = 16; small = 12
params = {'axes.titlesize': large,
          'legend.fontsize': med,
          'figure.figsize': (16, 10),
          'axes.labelsize': med,
```



```
'axes.titlesize': med,  
    'xtick.labelsize': med,  
    'ytick.labelsize': med,  
    'figure.titlesize': large}  
plt.rcParams.update(params)  
plt.style.use('seaborn-whitegrid')  
sns.set_style("white")  
%matplotlib inline
```



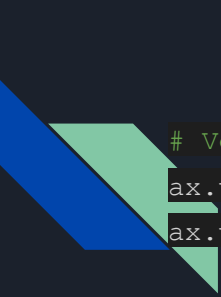
```
import matplotlib.lines as mlines

# Import Data
df =
pd.read_csv("https://raw.githubusercontent.com/tigershiva02/dataset/main/ct3clouddataset.csv")

left_label = [str(c) + ', ' + str(round(y)) for c, y in zip(df.Student, df['1st'])]
right_label = [str(c) + ', ' + str(round(y)) for c, y in zip(df.Student, df['4th'])]
klass = ['red' if (y1-y2) < 0 else 'green' for y1, y2 in zip(df['1st'], df['4th'])]

def newline(p1, p2, color='black'):
    ax = plt.gca()
    l = mlines.Line2D([p1[0],p2[0]], [p1[1],p2[1]], color='red' if p1[1]-p2[1] > 0 else
'green', marker='o', markersize=6)
    ax.add_line(l)
    return l

fig, ax = plt.subplots(1,1,figsize=(14,14), dpi= 80)
```



```
# Vertical Lines
```

```
ax.vlines(x=1, ymin=5, ymax=10, color='black', alpha=0.7, linewidth=1, linestyle='dotted')
```

```
ax.vlines(x=3, ymin=5, ymax=10, color='black', alpha=0.7, linewidth=1, linestyle='dotted')
```

```
# Points
```

```
ax.scatter(y=df['1st'], x=np.repeat(1, df.shape[0]), s=10, color='black', alpha=0.7)
```

```
ax.scatter(y=df['4th'], x=np.repeat(3, df.shape[0]), s=10, color='black', alpha=0.7)
```


```
# Line Segments and Annotation
```

```
for p1, p2, c in zip(df['1st'], df['4th'], df['Student']):
```

```
    newline([1,p1], [3,p2])
```

```
    ax.text(1-0.05, p1, c + ', ' + str(round(p1)), horizontalalignment='right',  
verticalalignment='center', fontdict={'size':14})
```

```
    ax.text(3+0.05, p2, c + ', ' + str(round(p2)), horizontalalignment='left',  
verticalalignment='center', fontdict={'size':14})
```

```
# 'Before' and 'After' Annotations
```

```
ax.text(1-0.05, 12, 'BEFORE', horizontalalignment='right', verticalalignment='center',  
fontdict={'size':18, 'weight':700})
```

```
ax.text(3+0.05, 12, 'AFTER', horizontalalignment='left', verticalalignment='center',  
fontdict={'size':18, 'weight':700})
```

```
# Decoration
```

```
ax.set_title("Slopechart: Comparing CGPA of students when they were in year 1 and 4",  
fontdict={'size':22})
```

```
ax.set(xlim=(0,4), ylim=(0,13), ylabel='CGPA')
```

```
ax.set_xticks([1,3])
```

```
ax.set_xticklabels(["1st Year", "4th Year"])
```

```
plt.yticks(np.arange(5, 11, 1), fontsize=15)
```

```
# Lighten borders
```

```
plt.gca().spines["top"].set_alpha(.0)
```

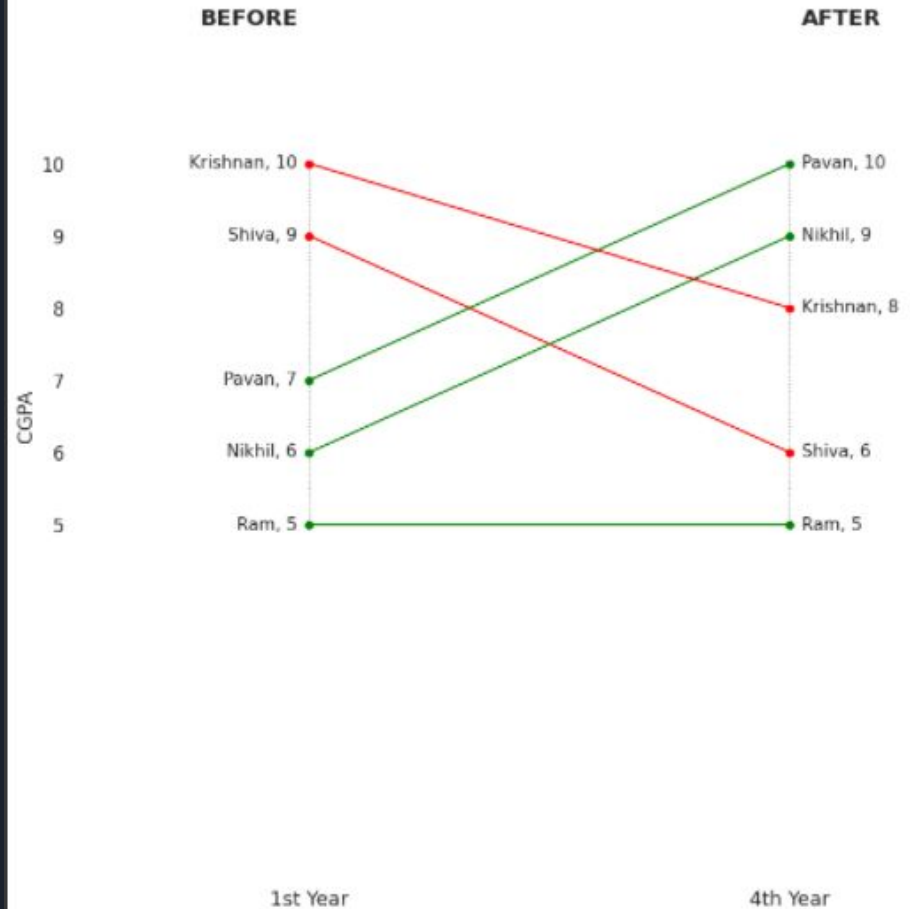
```
plt.gca().spines["bottom"].set_alpha(.0)
```

```
plt.gca().spines["right"].set_alpha(.0)
```

```
plt.gca().spines["left"].set_alpha(.0)
```

```
plt.show()
```

Slopechart: Comparing CGPA of students when they were in year 1 and 4



OUTPUT



THANK YOU