Forward School

Program Code: J620-002-4:2020

Program Name: FRONT-END SOFTWARE DEVELOPMENT

Title : Case Study - Data Analysis of Student Performance

Name: Ooi Caaron

IC Number: 990701-07-5837

Date : 7/7/23

Introduction :

Conclusion : Still need to practice more and do revision

Guideline EDA link: https://medium.com/dataseries/an-eda-checklist-800beeaee555 (https://medium.com/dataseries/an-eda-checklist-800beeaee555)

Sample Exercise:

High Student students academic performance

I'll do the dataset in Excel

Randomizers in Excel (dont shoot me)

I like to "visualize my simulated data"

=RANDBETWEEN(0,100)

=CHOOSE(RANDBETWEEN(1,3),"B40","M40","T20")

What data is needed?

Describe the data

```
Student demography
Subjects taken
Trial exam results
attendance, contact Hours
Final results
Others? Sports activities
```

"Correlation is not causation"

Case Study Exercise

Plot the Student Results table

Some basic stats

Look for Average, Min, Max

Exploratory Data Analysis (EDA) Check list

- Domain knowledge
  - What is this dataset about?
- Check if the data is intuitive
- Find out how the data was generated
- Understand the process

- Select a smaller dataset
  - depending on the data size, If what to go big bang, make sure enough resources.
- Explore individual features
- Explore pairs and groups

- Clean up features
- Selecting features of interest
- Generating derived feature(s)
- Extract , Transform and Load (the whole dataset)
- Sampling the data (in ML)

In [13]:

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

1. Import Data from CSV

In [53]:

```
df = pd.read_csv('StudentResults.csv')
df
```

Out[53]:

	StudentID	Name	Term	IncomeGroup	NonsenseData	School	Tuisyen	Attendance
0	7	Psy	2	B40	xvxc	SK Bestari	No	60
1	8	Edward	2	M40	sf	SK Bestari	Yes	30
2	6	Mei Lin	2	M40	dsf	SK Bestari	Yes	78
3	9	Miyazawa	2	T20	df	SK Bestari	No	100
4	4	Letchumi	2	T20	xvxc	SK Bestari	No	80
5	3	Muthu	2	T20	sf	SK Bestari	Yes	58
6	5	Ah Chong	2	B40	dsf	SK Bestari	Yes	64
7	2	Siti	2	M40	df	SK Bestari	Yes	57
8	1	Ali	2	B40	xvxc	SK Bestari	No	100
9	10	Ah Beng	2	T20	sf	SK Bestari	No	100
10	7	Psy	1	B40	dsf	SK Bestari	No	60
11	8	Edward	1	M40	df	SK Bestari	Yes	30
12	6	Mei Lin	1	M40	xvxc	SK Bestari	Yes	78
13	9	Miyazawa	1	T20	sf	SK Bestari	No	100
14	4	Letchumi	1	T20	dsf	SK Bestari	No	80
15	3	Muthu	1	T20	df	SK Bestari	Yes	58
16	5	Ah Chong	1	B40	xvxc	SK Bestari	Yes	64
17	2	Siti	1	M40	sf	SK Bestari	Yes	57
18	1	Ali	1	B40	dsf	SK Bestari	No	100
19	10	Ah Beng	1	T20	df	SK Bestari	No	100

In [54]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 13 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   StudentID     20 non-null     int64  
 1   Name          20 non-null     object 
 2   Term          20 non-null     int64  
 3   IncomeGroup   20 non-null     object 
 4   NonsenseData  20 non-null     object 
 5   School        20 non-null     object 
 6   Tuisyen       20 non-null     object 
 7   Attendance    20 non-null     int64  
 8   BM            20 non-null     int64  
 9   BI            19 non-null     float64
 10  Maths         20 non-null     int64  
 11  Sejarah       20 non-null     int64  
 12  Total         20 non-null     int64  
dtypes: float64(1), int64(7), object(5)
memory usage: 2.2+ KB
```

In [55]:

```
df.isnull().sum()
```

Out[55]:

```
StudentID       0
Name            0
Term            0
IncomeGroup     0
NonsenseData    0
School          0
Tuisyen         0
Attendance      0
BM              0
BI              1
Maths           0
Sejarah         0
Total           0
dtype: int64
```

2. Data Cleaning - Remove Useless Data

In [56]:

```
df['BI'] = df['BI'].fillna(0).clip(lower=0)
df.sort_values('StudentID').set_index('StudentID')
```

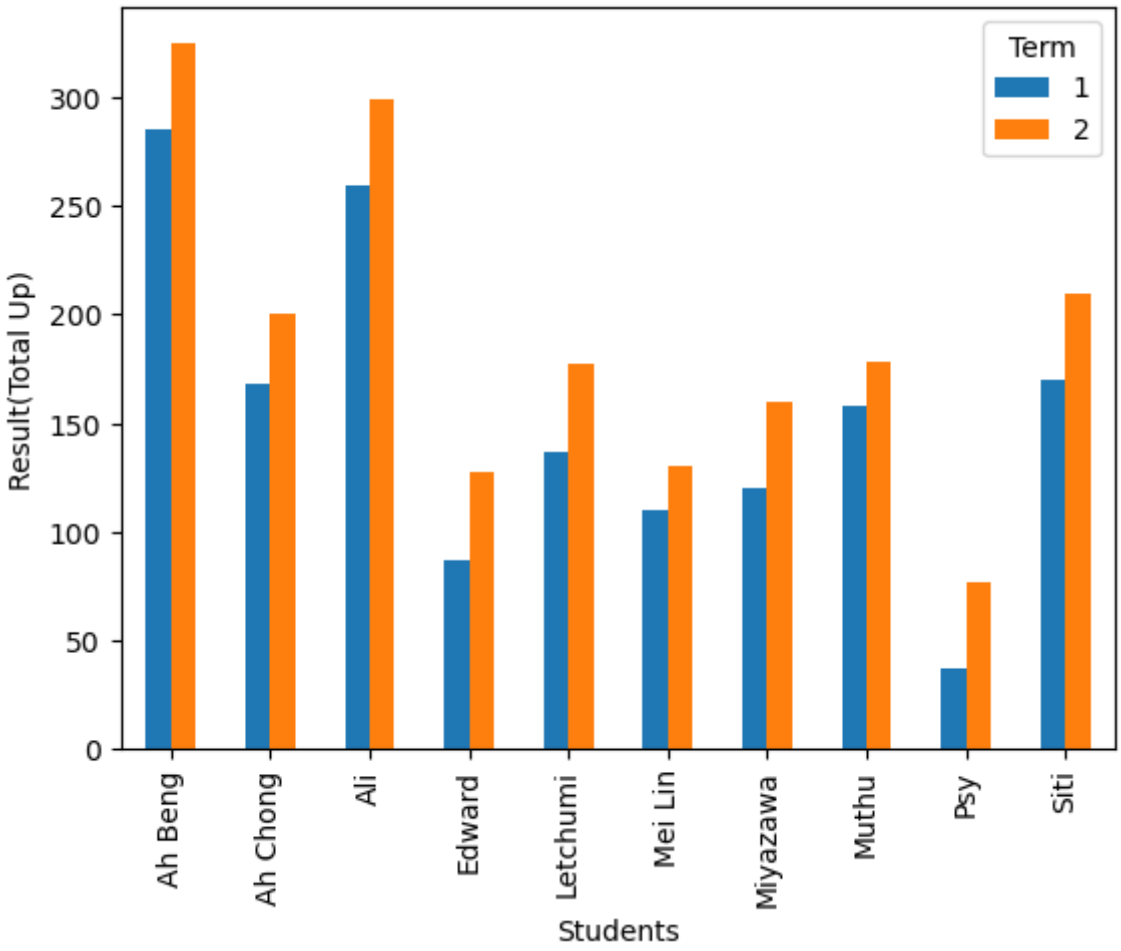
Out[56]:

StudentID	Name	Term	IncomeGroup	NonsenseData	School	Tuisyen	Attendance	BM
1	Ali	2	B40	xvxc	SK Bestari	No	100	16
1	Ali	1	B40	dsf	SK Bestari	No	100	6
2	Siti	2	M40	df	SK Bestari	Yes	57	35
2	Siti	1	M40	sf	SK Bestari	Yes	57	25
3	Muthu	2	T20	sf	SK Bestari	Yes	58	31
3	Muthu	1	T20	df	SK Bestari	Yes	58	21
4	Letchumi	2	T20	xvxc	SK Bestari	No	80	97
4	Letchumi	1	T20	dsf	SK Bestari	No	80	87
5	Ah Chong	2	B40	dsf	SK Bestari	Yes	64	16
5	Ah Chong	1	B40	xvxc	SK Bestari	Yes	64	6
6	Mei Lin	2	M40	dsf	SK Bestari	Yes	78	0
6	Mei Lin	1	M40	xvxc	SK Bestari	Yes	78	10
7	Psy	1	B40	dsf	SK Bestari	No	60	14
7	Psy	2	B40	xvxc	SK Bestari	No	60	24
8	Edward	2	M40	sf	SK Bestari	Yes	30	43
8	Edward	1	M40	df	SK Bestari	Yes	30	33
9	Miyazawa	1	T20	sf	SK Bestari	No	100	22
9	Miyazawa	2	T20	df	SK Bestari	No	100	32
10	Ah Beng	2	T20	sf	SK Bestari	No	100	43
10	Ah Beng	1	T20	df	SK Bestari	No	100	33

In [57]:

```
term = df.groupby(['Name', 'Term'])['Total'].sum().unstack()

term.plot.bar()
plt.xlabel('Students')
plt.ylabel('Result(Total Up)')
# plt.title('Country')
plt.show()
```



3. Basic Statistics of Table

In [59]:

```
df.describe()
```

Out[59]:

	StudentID	Term	Attendance	BM	BI	Maths	Sejarah	
count	20.000000	20.000000	20.000000	20.000000	20.000000	20.000000	20.000000	20.0
mean	5.500000	1.500000	72.700000	29.700000	55.500000	34.800000	51.200000	170.7
std	2.946898	0.512989	22.571757	24.525175	32.763105	30.365493	36.685864	75.5
min	1.000000	1.000000	30.000000	0.000000	0.000000	0.000000	-9.000000	37.0
25%	3.000000	1.000000	58.000000	15.500000	26.000000	11.500000	14.750000	125.2
50%	5.500000	1.500000	71.000000	24.500000	61.500000	26.000000	62.500000	164.0
75%	8.000000	2.000000	100.000000	33.500000	84.000000	39.500000	82.500000	202.5
max	10.000000	2.000000	100.000000	97.000000	100.000000	97.000000	97.000000	325.0

The top 3 and last 3 students each term

In [79]:

```
term1t = df[df['Term'] == 1].groupby(['StudentID','Name'])[['Total','Term']].sum().sort_
term1l = df[df['Term'] == 1].groupby(['StudentID','Name'])[['Total','Term']].sum().sort_
term2t = df[df['Term'] == 2].groupby(['StudentID','Name'])[['Total','Term']].sum().sort_
term2l = df[df['Term'] == 2].groupby(['StudentID','Name'])[['Total','Term']].sum().sort_
print("Top 3 students for 1st Term")
print(term1t)
print('')
print("Last 3 students for 1st Term")
print(term1l)
print('')
print("Top 3 students for 2nd Term")
print(term2t)
print('')
print("Last 3 students for 2nd Term")
print(term2t)
print('')
```

```
Top 3 students for 1st Term
                   Total  Term
StudentID Name
10        Ah Beng    285     1
1         Ali        259     1
2         Siti       170     1

Last 3 students for 1st Term
                   Total  Term
StudentID Name
6         Mei Lin    110     1
8         Edward      87     1
7         Psy         37     1

Top 3 students for 2nd Term
                   Total  Term
StudentID Name
10        Ah Beng    325     2
1         Ali        299     2
2         Siti       210     2

Last 3 students for 2nd Term
                   Total  Term
StudentID Name
10        Ah Beng    325     2
1         Ali        299     2
2         Siti       210     2
```

Average Scores for each term

In [81]:

```
avg = df.groupby('Term')[['BI', 'BM', 'Maths', 'Sejarah', 'Total']].mean()
avg
```

Out[81]:

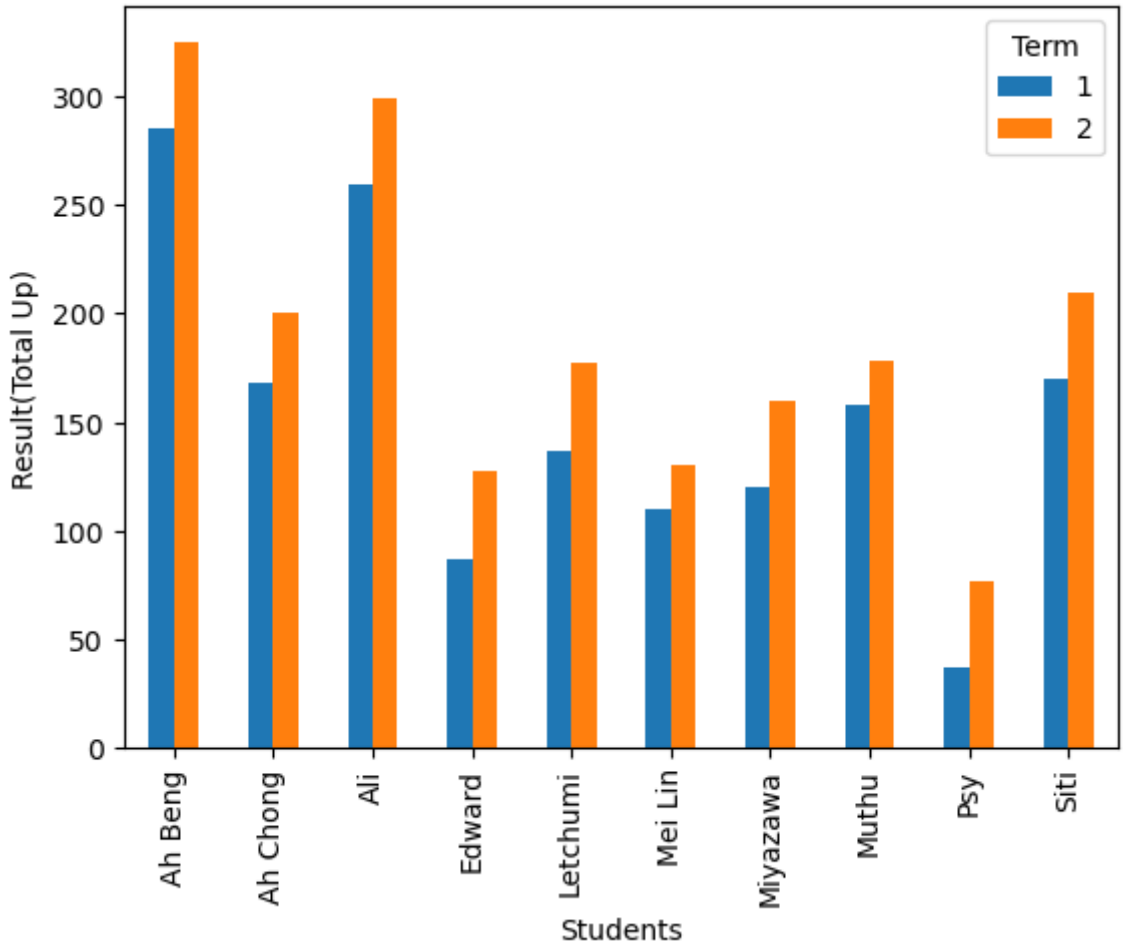
Term	BI	BM	Maths	Sejarah	Total
1	51.0	25.7	31.2	46.2	153.1
2	60.0	33.7	38.4	56.2	188.3

Max score for each subject

In [82]:

```
term = df.groupby(['Name', 'Term'])['Total'].sum().unstack()

term.plot.bar()
plt.xlabel('Students')
plt.ylabel('Result(Total Up)')
# plt.title('Country')
plt.show()
```



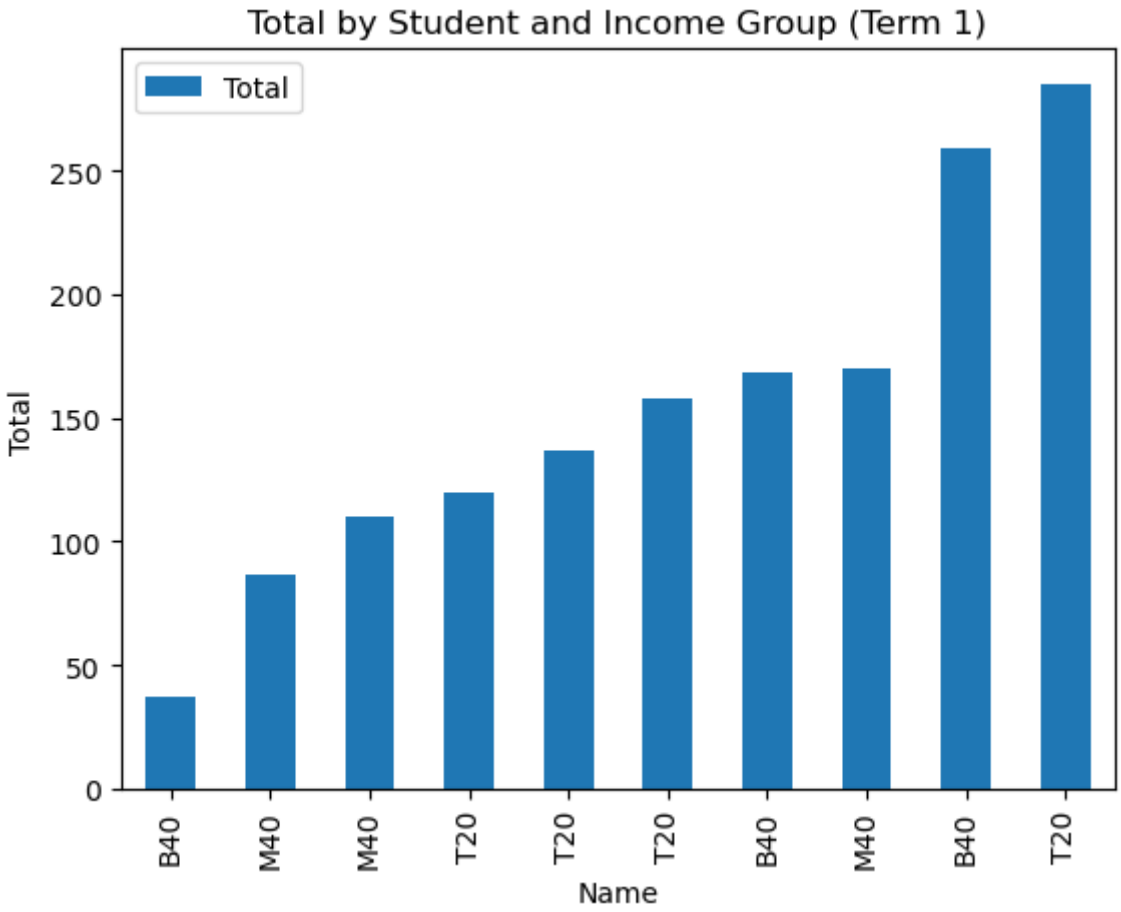
Type Markdown and LaTeX: $\alpha 2$

In [110]:

```
# term = df[df['Term'] == 1]
# plt.figure(figsize = (15,8))
# plt.bar(term['Name'], term['Total'])
# plt.ticklabel_format(useOffset = False, style = 'plain', axis = 'y')
# plt.xlabel('Region')
# plt.ylabel('Death Rate')
# plt.show()

term1 = df[df['Term'] == 1]
term1 = term1[['Total','IncomeGroup']].set_index('IncomeGroup')
term1.plot(kind='bar')

plt.xlabel('Name')
plt.ylabel('Total')
plt.title('Total by Student and Income Group (Term 1)')
plt.xticks(rotation=90)
plt.show()
```



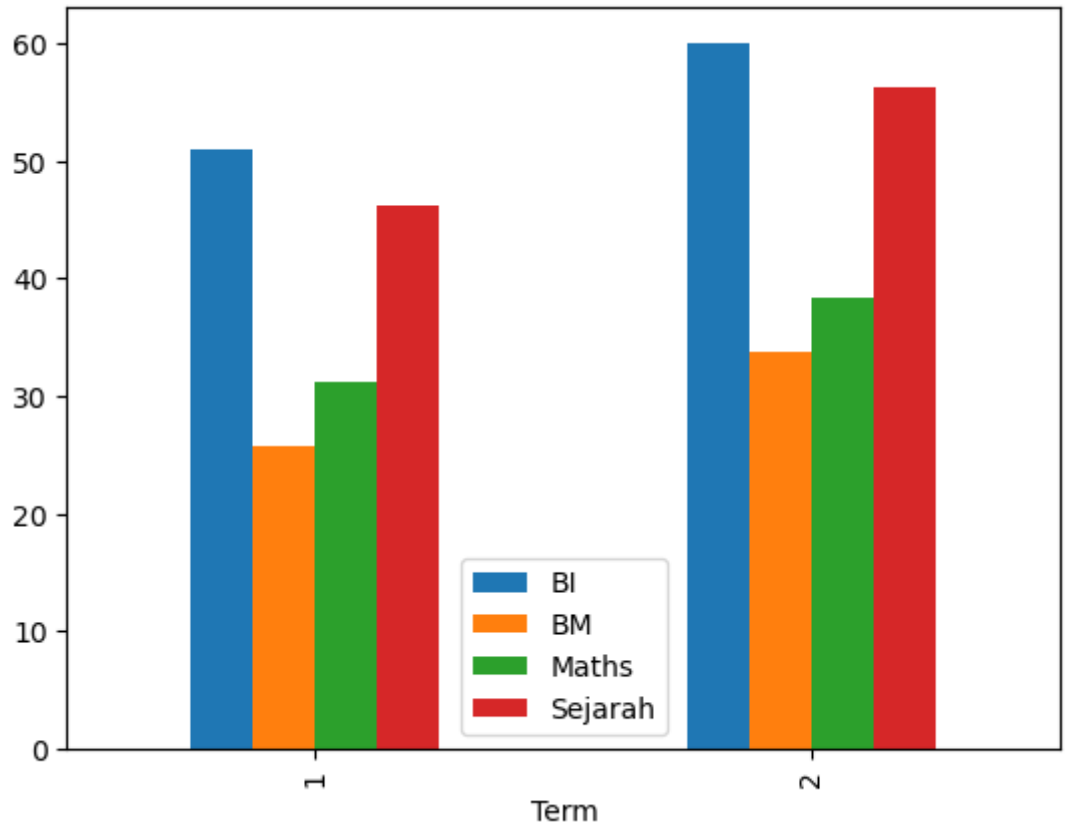
In [114]:

```
subs = df.groupby('Term')['BI','BM','Maths','Sejarah'].mean().reset_index().set_index('T
subs.plot(kind='bar')
```

```
C:\Users\User\AppData\Local\Temp\ipykernel_6620\4166321717.py:1: FutureWar
ning: Indexing with multiple keys (implicitly converted to a tuple of key
s) will be deprecated, use a list instead.
  subs = df.groupby('Term')['BI','BM','Maths','Sejarah'].mean().reset_inde
x().set_index('Term')
```

Out[114]:

```
<Axes: xlabel='Term'>
```



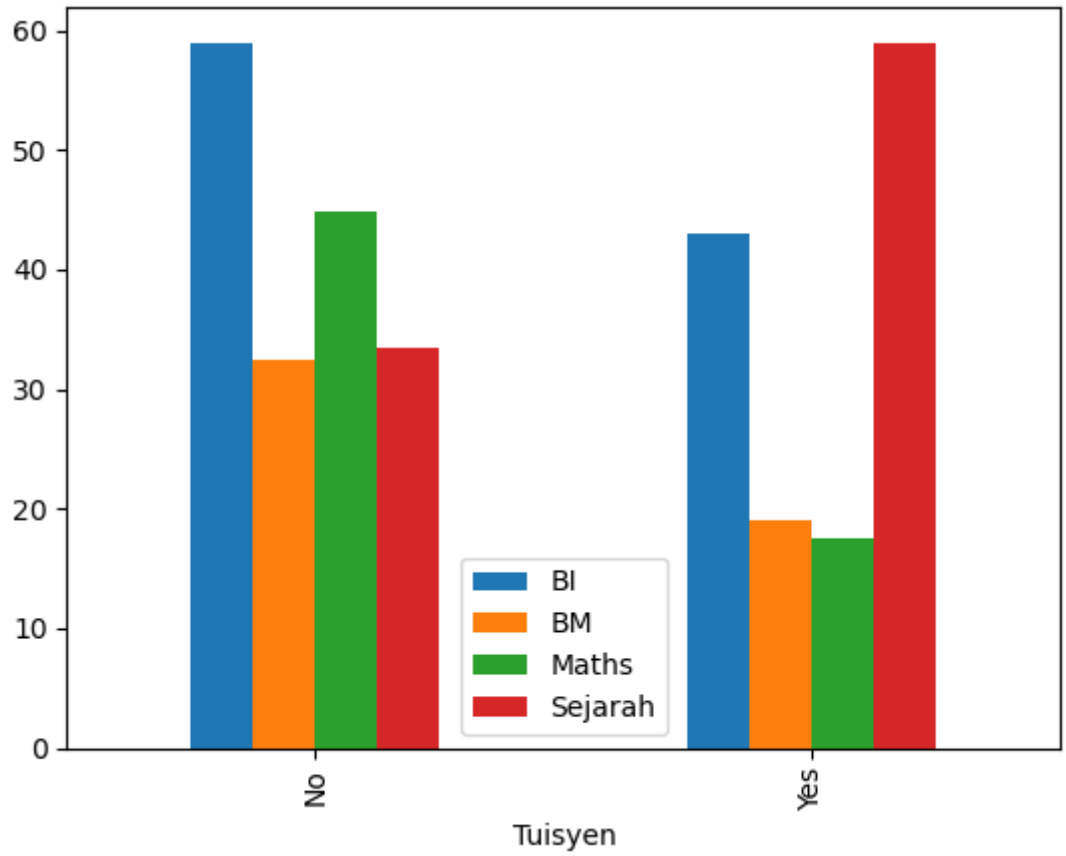
In [115]:

```
subs = df[df['Term']==1].groupby(['Tuisyen'])['BI','BM','Maths','Sejarah'].mean().reset_
subs.plot(kind='bar')
```

```
C:\Users\User\AppData\Local\Temp\ipykernel_6620\651614195.py:1: FutureWarn
ing: Indexing with multiple keys (implicitly converted to a tuple of keys)
will be deprecated, use a list instead.
  subs = df[df['Term']==1].groupby(['Tuisyen'])['BI','BM','Maths','Sejara
h'].mean().reset_index().set_index('Tuisyen')
```

Out[115]:

```
<Axes: xlabel='Tuisyen'>
```



4. Replace IncomeGroup and Tuisyen to Numerical Value

In [116]:

```
income_mapping = {'B40': 1, 'M40': 2, 'T20': 3}
tuisyen_mapping = {'Yes': 0, 'No': 1}

df['IncomeGroup'] = df['IncomeGroup'].map(income_mapping)
df['Tuisyen'] = df['Tuisyen'].map(tuisyen_mapping)
df
```

Out[116]:

	StudentID	Name	Term	IncomeGroup	NonsenseData	School	Tuisyen	Attendance
0	7	Psy	2	1	xvxc	SK Bestari	1	60
1	8	Edward	2	2	sf	SK Bestari	0	30
2	6	Mei Lin	2	2	dsf	SK Bestari	0	78
3	9	Miyazawa	2	3	df	SK Bestari	1	100
4	4	Letchumi	2	3	xvxc	SK Bestari	1	80
5	3	Muthu	2	3	sf	SK Bestari	0	58
6	5	Ah Chong	2	1	dsf	SK Bestari	0	64
7	2	Siti	2	2	df	SK Bestari	0	57
8	1	Ali	2	1	xvxc	SK Bestari	1	100
9	10	Ah Beng	2	3	sf	SK Bestari	1	100
10	7	Psy	1	1	dsf	SK Bestari	1	60
11	8	Edward	1	2	df	SK Bestari	0	30
12	6	Mei Lin	1	2	xvxc	SK Bestari	0	78
13	9	Miyazawa	1	3	sf	SK Bestari	1	100
14	4	Letchumi	1	3	dsf	SK Bestari	1	80
15	3	Muthu	1	3	df	SK Bestari	0	58
16	5	Ah Chong	1	1	xvxc	SK Bestari	0	64
17	2	Siti	1	2	sf	SK Bestari	0	57
18	1	Ali	1	1	dsf	SK Bestari	1	100
19	10	Ah Beng	1	3	df	SK Bestari	1	100

5. Check the correlation between income group, tuisyen and result

In [123]:

```
co = df[['IncomeGroup', 'Tuisyen','BM','BI','Maths','Sejarah', 'Total']].corr()
co
```

Out[123]:

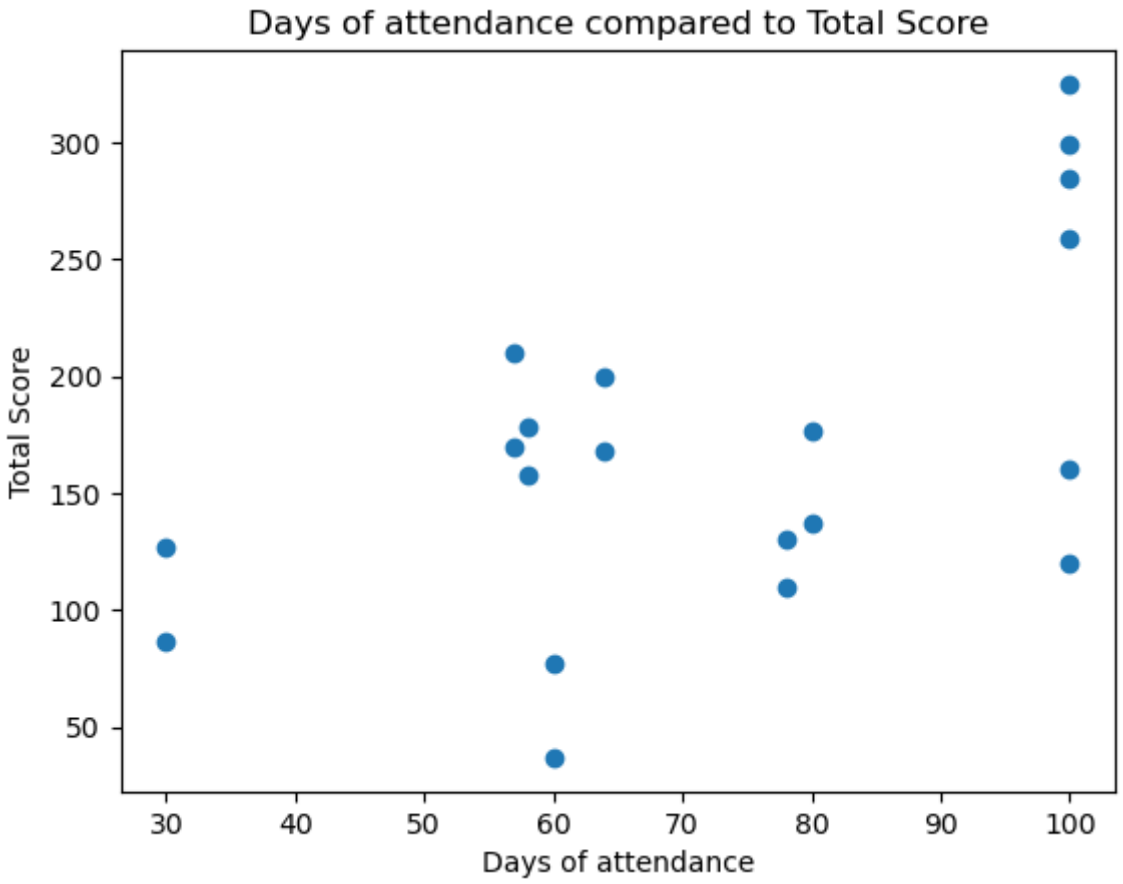
	IncomeGroup	Tuisyen	BM	BI	Maths	Sejarah	Total
IncomeGroup	1.000000	0.120386	0.565563	0.273317	-0.137483	-0.259916	0.129596
Tuisyen	0.120386	1.000000	0.322119	0.234863	0.506815	-0.357972	0.229420
BM	0.565563	0.322119	1.000000	0.028886	-0.219737	-0.427311	0.045778
BI	0.273317	0.234863	0.028886	1.000000	0.349795	0.449886	0.813588
Maths	-0.137483	0.506815	-0.219737	0.349795	1.000000	0.406545	0.681237
Sejarah	-0.259916	-0.357972	-0.427311	0.449886	0.406545	1.000000	0.713688
Total	0.129596	0.229420	0.045778	0.813588	0.681237	0.713688	1.000000

In [129]:

```
plt.scatter(df['Attendance'], df['Total'])
plt.ylabel('Total Score')
plt.xlabel('Days of attendance')
plt.title('Days of attendance compared to Total Score')
```

Out[129]:

```
Text(0.5, 1.0, 'Days of attendance compared to Total Score')
```



6. Conclusion

What is your finding?