Analysis of Grades

This file will have a simple analysis of grades of students from different batches of AI training. Simple concepts of univariate analysis, central tendencies etc will be used to gather the insights.

Step 1: Preparing the environment

In this phase we will load the necessary packages and the obtain the dataset. We will view the dataset and the variables.

In [55]:
```
#getting the packages
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import scipy.stats as stats
import statsmodels.api as sm
from statsmodels.formula.api import ols
```

In [3]:
```
#getting the dataset
aibt = pd.read_csv('C:\\Users\\sujoydutta\\Desktop\\Data analysis\\Internships\\Innomatics\\Data science\\AI_ba

aibt .head()
```

Out[3]:

	Batch	User_ID	Score
0	AI_ELITE_7	uid_149	6 / 7
1	AI_ELITE_7	uid_148	6 / 7
2	AI_ELITE_7	uid_147	7 / 7
3	AI_ELITE_7	uid_146	7 / 7
4	AI_ELITE_7	uid_145	4 / 7

In [12]:
```
#getting the dataset info
aibt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 149 entries, 0 to 148
Data columns (total 3 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   Batch    149 non-null    object
 1   User_ID  149 non-null    object
 2   Score    149 non-null    object
dtypes: object(3)
memory usage: 3.6+ KB
```

Step 2: Data Pre processing

In this step, we are going to make the dataset fit for analysis. Here we are going to clean the variables and remove the unnecessary clutter. The variables would be put in the right format. Null values would be detected and will be dropped or replaced. Also this involves feature engineering where we will create right variable by combining new ones. The uselesss variables would be dropped from the dataset.

In [32]:
```
#seeing null values
aibt.isnull().sum().sort_values(ascending=False)
```

Out[32]:
```
Batch         0
User_ID       0
percentage    0
dtype: int64
```

In [8]:
```
# column name cleaning

aibt.columns = aibt.columns.str.replace(' ', '')
aibt.columns
```

Out[8]:
```
Index(['Batch', 'User_ID', 'Score'], dtype='object')
```

In [17]:
```
#splitting column
aibt[['Obtained marks', 'Max marks']] = aibt.Score.str.split("/", expand = True)
aibt
```

Out[17]:

	Batch	User_ID	Score	Obtained marks	Max marks
0	AI_ELITE_7	uid_149	6 / 7	6	7
1	AI_ELITE_7	uid_148	6 / 7	6	7
2	AI_ELITE_7	uid_147	7 / 7	7	7
3	AI_ELITE_7	uid_146	7 / 7	7	7
4	AI_ELITE_7	uid_145	4 / 7	4	7
...	...	...	...	...	...
144	AI_ELITE_4	uid_5	4 / 7	4	7
145	AI_ELITE_4	uid_4	4 / 7	4	7
146	AI_ELITE_4	uid_3	4 / 7	4	7
147	AI_ELITE_4	uid_2	3 / 7	3	7
148	AI_ELITE_4	uid_1	2 / 7	2	7

149 rows × 5 columns

In [18]:
```
#converting to the right format
aibt['Obtained marks'] = aibt['Obtained marks'].astype('float')
aibt['Max marks'] = aibt['Max marks'].astype('float')

aibt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 149 entries, 0 to 148
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Batch           149 non-null    object
 1   User_ID         149 non-null    object
 2   Score           149 non-null    object
 3   Obtained marks  149 non-null    float64
 4   Max marks       149 non-null    float64
dtypes: float64(2), object(3)
memory usage: 5.9+ KB
```

In [27]:
```
#feature engineering
aibt['percentage']=aibt['Obtained marks'] /aibt['Max marks'] *100
aibt['percentage']=aibt['percentage'].round(2)
aibt.head()
```

Out[27]:

	Batch	User_ID	Score	Obtained marks	Max marks	percentage
0	AI_ELITE_7	uid_149	6 / 7	6.0	7.0	85.71
1	AI_ELITE_7	uid_148	6 / 7	6.0	7.0	85.71
2	AI_ELITE_7	uid_147	7 / 7	7.0	7.0	100.00
3	AI_ELITE_7	uid_146	7 / 7	7.0	7.0	100.00
4	AI_ELITE_7	uid_145	4 / 7	4.0	7.0	57.14

In [30]:
```
#dropping unnecessary features
aibt=aibt.drop(['Score','Obtained marks','Max marks'], axis=1)
aibt.head()
```

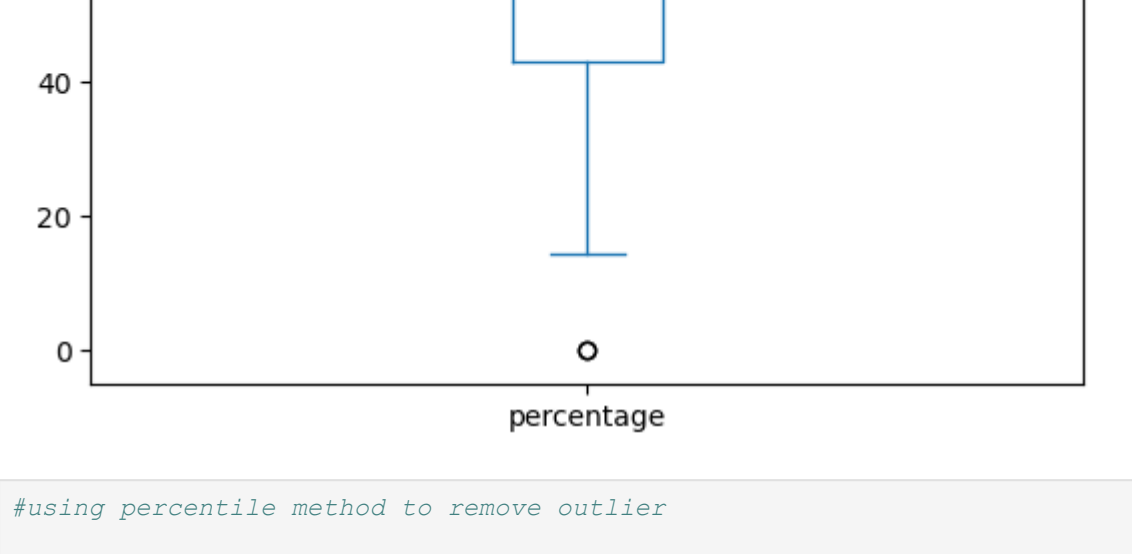
Out[30]:

	Batch	User_ID	percentage
0	AI_ELITE_7	uid_149	85.71
1	AI_ELITE_7	uid_148	85.71
2	AI_ELITE_7	uid_147	100.00
3	AI_ELITE_7	uid_146	100.00
4	AI_ELITE_7	uid_145	57.14

In [36]:
```
# Seeing extreme values


aibt['percentage'].plot(kind='box')
```

Out[36]:
```
<AxesSubplot: >
```



In [37]:
```
#using percentile method to remove outlier

aibt['percentage'].quantile([0.1, 0.25, 0.5, 0.70, 0.9, 0.95, 0.99])
```

Out[37]:
```
0.10     28.57
0.25     42.86
0.50     57.14
0.70     71.43
0.90    100.00
0.95    100.00
0.99    100.00
Name: percentage, dtype: float64
```

In [43]:
```
#dropping the outliers
aibt1= aibt[aibt['percentage']> 0].copy()
print("old length:",len(aibt))
print("new length:",len(aibt1))
print("outliers dropped:",len(aibt1)-len(aibt))
```

```
old length: 149
new length: 147
outliers dropped: -2
```

Step 3: Exploratory data Analysis

Here we are going to gather insights from the dataset which will help us understand what is happening and help us answer questions.

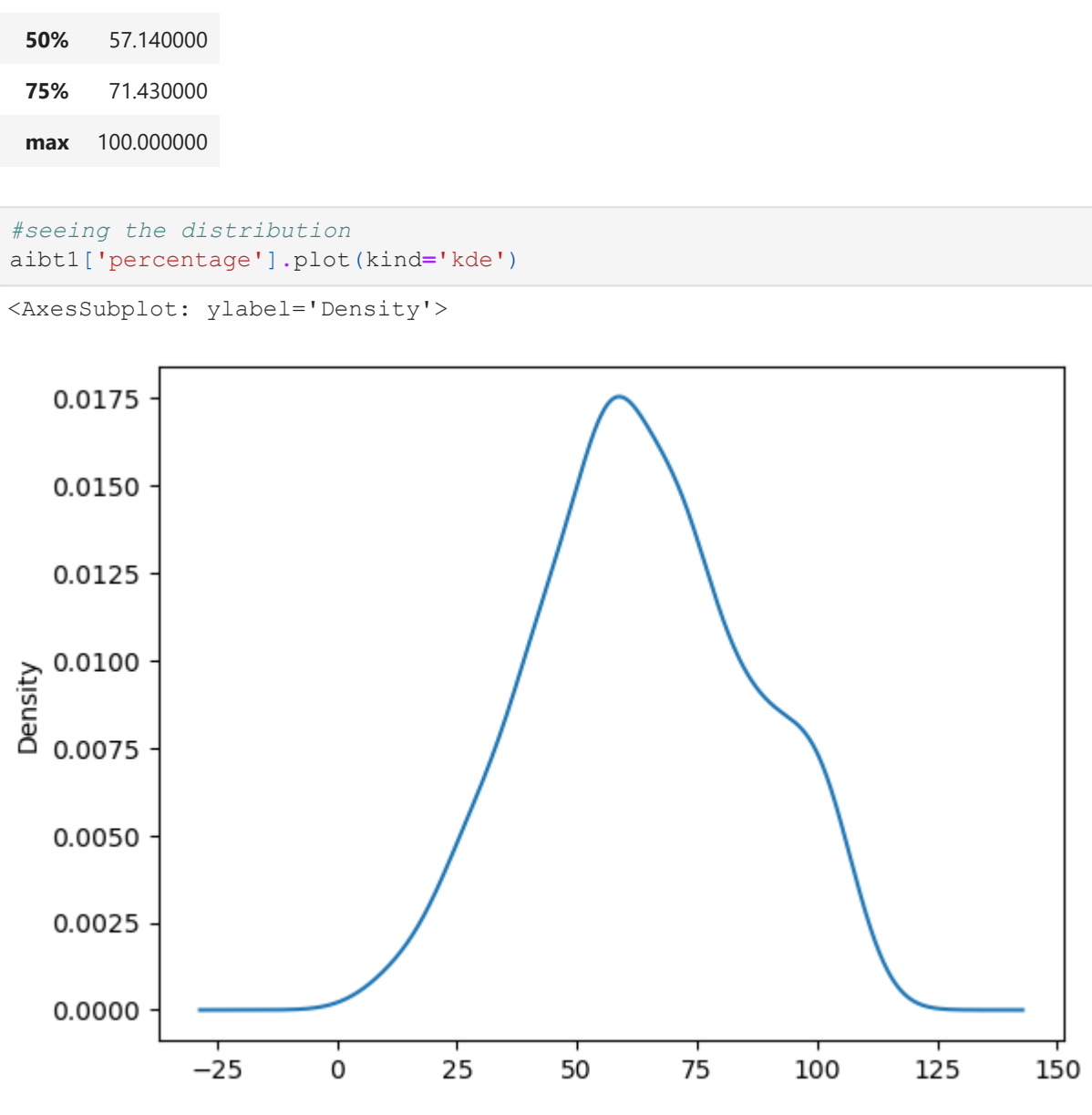
In [44]:
```
#summary statistics
aibt1.describe()
```

Out[44]:

	percentage
count	147.000000
mean	63.459116
std	21.679488
min	14.290000
25%	42.860000
50%	57.140000
75%	71.430000
max	100.000000

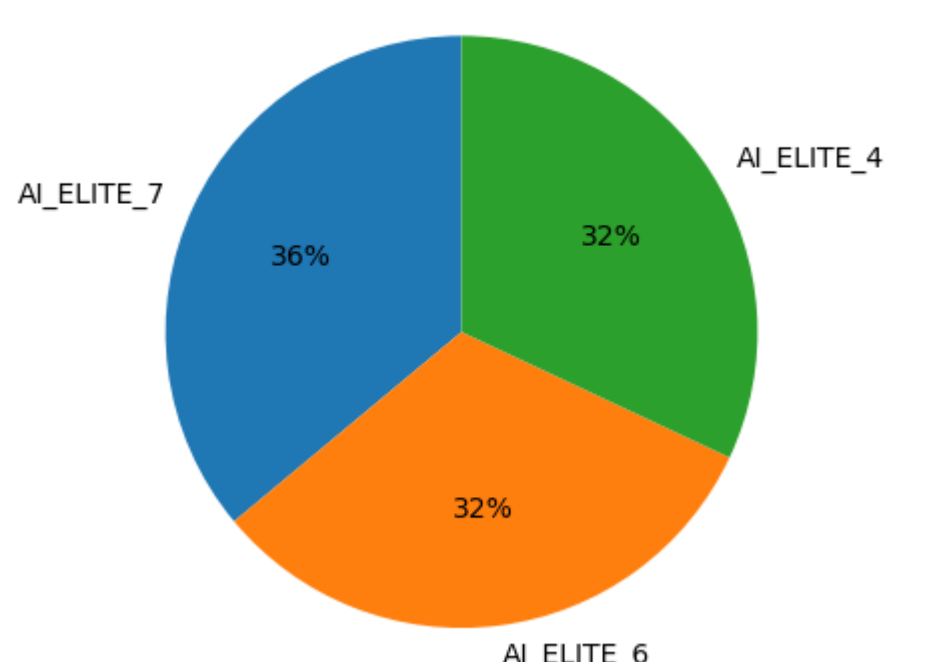
In [45]:
```
#seeing the distribution
aibt1['percentage'].plot(kind='kde')
```

Out[45]:
```
<AxesSubplot: ylabel='Density'>
```



In [105...
```
#Which batch has the most students?
Batch=aibt1['Batch'].unique()
Count=aibt1['Batch'].value_counts()


plt.pie(Count, labels = Batch, startangle = 90,autopct='%.0f%%')
plt.show()
```



In [83]:
```
#Seeing top fifteen students by percentage
toppers=aibt1[['percentage', 'Batch','User_ID']].sort_values('percentage', ascending=False).nlargest(15, 'perce
toppers
```

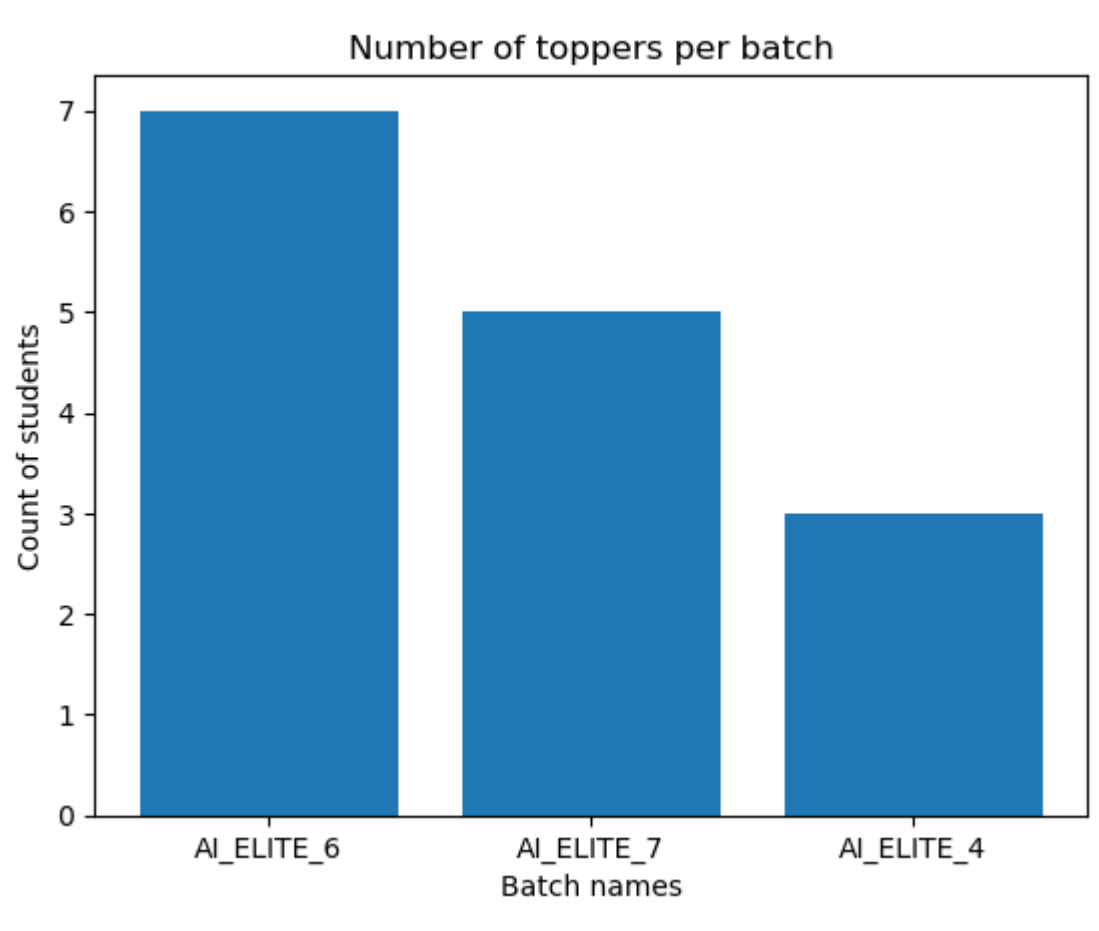
Out[83]:

	percentage	Batch	User_ID
74	100.0	AI_ELITE_6	uid_75
67	100.0	AI_ELITE_6	uid_82
16	100.0	AI_ELITE_7	uid_133
15	100.0	AI_ELITE_7	uid_134
24	100.0	AI_ELITE_7	uid_125
78	100.0	AI_ELITE_6	uid_71
19	100.0	AI_ELITE_7	uid_130
63	100.0	AI_ELITE_6	uid_86
98	100.0	AI_ELITE_6	uid_51
101	100.0	AI_ELITE_4	uid_48
105	100.0	AI_ELITE_4	uid_44
7	100.0	AI_ELITE_7	uid_142
35	100.0	AI_ELITE_7	uid_114
5	100.0	AI_ELITE_7	uid_144
139	100.0	AI_ELITE_4	uid_10

In [84]:
```
#Number of toppers per batch
Batch=toppers['Batch'].unique()
Count=toppers['Batch'].value_counts()

plt.bar(Batch, Count)
plt.xlabel("Batch names")
plt.ylabel("Count of students")
plt.title("Number of toppers per batch")
```

Out[84]:
```
Text(0.5, 1.0, 'Number of toppers per batch')
```



In [88]:
```
#Seeing bottom fifteen students by percentage
dunces=aibt1[['percentage', 'Batch','User_ID']].sort_values('percentage', ascending=False).nsmallest(15, 'perce
dunces
```

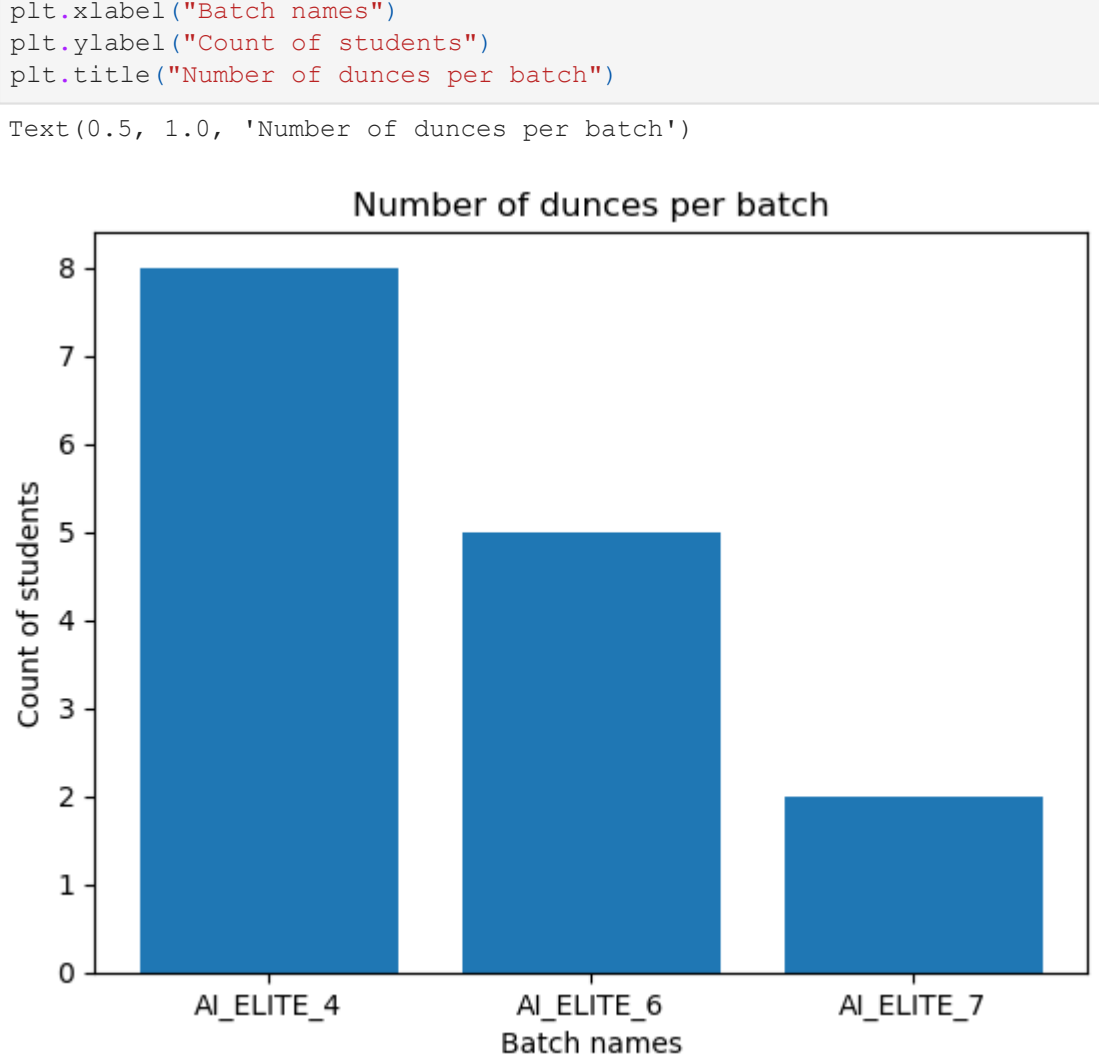
Out[88]:

	percentage	Batch	User_ID
129	14.29	AI_ELITE_4	uid_20
71	14.29	AI_ELITE_6	uid_78
97	14.29	AI_ELITE_6	uid_52
37	28.57	AI_ELITE_7	uid_112
92	28.57	AI_ELITE_6	uid_57
50	28.57	AI_ELITE_7	uid_99
93	28.57	AI_ELITE_6	uid_56
116	28.57	AI_ELITE_4	uid_33
114	28.57	AI_ELITE_4	uid_35
113	28.57	AI_ELITE_4	uid_36
111	28.57	AI_ELITE_4	uid_38
109	28.57	AI_ELITE_4	uid_40
108	28.57	AI_ELITE_4	uid_41
95	28.57	AI_ELITE_6	uid_54
148	28.57	AI_ELITE_4	uid_1

In [91]:
```
#Number of dunces per batch
Batch=dunces['Batch'].unique()
Count=dunces['Batch'].value_counts()

plt.bar(Batch, Count)
plt.xlabel("Batch names")
plt.ylabel("Count of students")
plt.title("Number of dunces per batch")
```

Out[91]:
```
Text(0.5, 1.0, 'Number of dunces per batch')
```



In [56]:
```
#checking if marks differs according to the batch of the student
mod = ols("percentage ~ Batch", data = aibt1).fit()
anov_table = sm.stats.anova_lm(mod)
anov_table
```

Out[56]:

	df	sum_sq	mean_sq	F	PR(>F)
Batch	2.0	7342.750157	3671.375078	8.627635	0.000289
Residual	144.0	61277.280228	425.536668	NaN	NaN

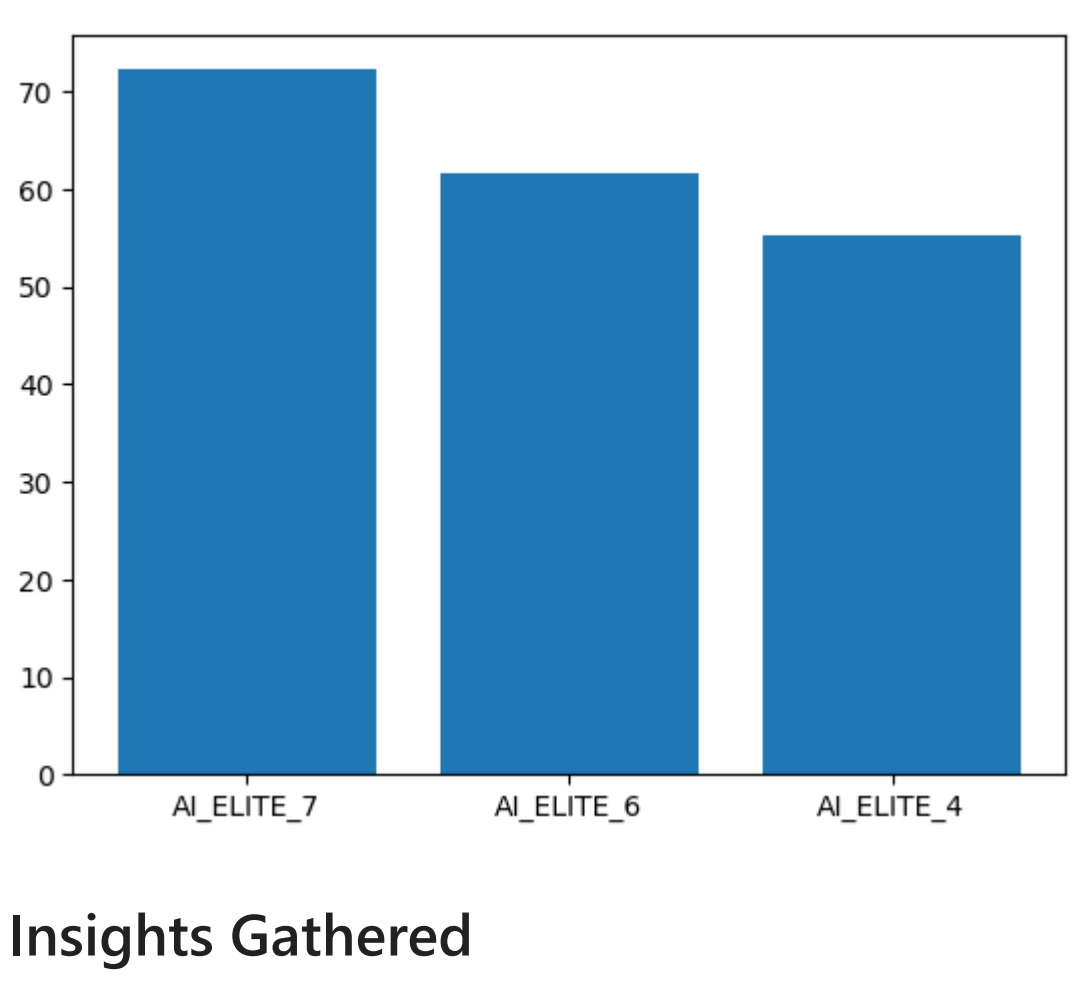
In [100...
```
#seeing average percentage by batch
avgmarks=aibt1.groupby(['Batch']).mean('percentage').reset_index().sort_values('percentage', ascending=False)
avgmarks
```

Out[100]:

	Batch	percentage
2	AI_ELITE_7	72.236226
1	AI_ELITE_6	61.702128
0	AI_ELITE_4	55.318511

In [104...
```
#plotting the average marks
plt.bar(avgmarks['Batch'], avgmarks['percentage'])
```

Out[104]:
```
<BarContainer object of 3 artists>
```



Insights Gathered

The following insights were gathered from the dataset:

1.There are 147 batches altogether in the three batches.

2.The average marks is 63% , highest is 100% and lowest is 14%.

3.The distribution of marks is normal.

4.There are almost same number of students in all batches but batch 7 has a bit more.

5.Batch 4 has the most bottom performers in all 3 batches.

6.Batch 6 has the most toppers in all 3 batches.

7.Batch 7 has the highest average in all 3 batches.