

Ex.No-2**PANDAS****AIM:**

To analyse and study the best performance point of Reciprocating pumps using Pandas.

PROCEDURE :**1. Dataset Creation:**

Create a hypothetical dataset containing information about actual discharge(m³/s), input power(W), and output power(W).

2. Correlation Analysis :

Calculate the correlation matrix to examine the relationships between actual Discharge, input power, and output power using pandas' 'corr()' function.

3. Efficiency calculation :

Calculate the efficiency for each input value using the given formula: Efficiency(%)
$$= \text{Output_power} / \text{Input_power} * 100$$

4. Head calculation:

Calculate the total head for each performance using the given formula : Head (m) =
$$\text{output_power} / \text{actual discharge} * \rho g$$

5. Best Efficiency Point (BEP) :

Identify the Best Efficiency Point of the reciprocating pump from the efficiency by selecting the highest index values using the pandas' 'nlargest()' function

PROGRAM:

```
import pandas as pd
data={
    'Actual Discharge':[40,50,60,70,80,90],
    'Input Power':[1,2,3,4,5,10],
    'Output Power':[70,30,90,100,140,170]
}

density=1000 gravity=9.81

a=pd.DataFrame(data)

a['Efficiency']=(a['Output Power']/a['Input Power'])*100

a['Head']=(a['Output Power']/a['Actual Discharge'])/(density*gravity)
corr_matrix=a.corr()

print(corr_matrix)

max_efficiency=corr_matrix['Efficiency'].nlargest(2).iloc[1]

print("\nParameter with the highest correlation with efficiency=",max_efficiency)
```

OUTPUT:

	Actual Discharge	Input Power	Output Power	Efficiency	\
Actual Discharge	1.000000	0.922018	0.901611	-0.614487	
Input Power	0.922018	1.000000	0.881684	-0.533271	
Output Power	0.901611	0.881684	1.000000	-0.227847	
Efficiency	-0.614487	-0.533271	-0.227847	1.000000	
Head	0.466245	0.489913	0.797480	0.391574	

	Head
Actual Discharge	0.466245
Input Power	0.489913
Output Power	0.797480
Efficiency	0.391574
Head	1.000000

Parameter with the highest correlation with efficiency= 0.3915744643953921

Result:

The programs were run successfully