


ACTIVITY 1

1. Write a Python program to select the 'name' and 'score' columns from the following DataFrame.

Sample DataFrame:


```
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'M',
                    'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
                    'attempts': [1, 3, 4, 3, 5, 3, 6, 1, 7, 1] }
```

```
import pandas as pd
import numpy as np
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Mat
df = pd.DataFrame(exam_data)
print(df)
```



| | name | score | attempts |
|---|-----------|-------|----------|
| 0 | Anastasia | 12.5 | 1 |
| 1 | Dima | 9.0 | 3 |
| 2 | Katherine | 16.5 | 4 |
| 3 | James | NaN | 3 |
| 4 | Emily | 9.0 | 5 |
| 5 | Michael | 20.0 | 3 |
| 6 | Matthew | 14.5 | 6 |
| 7 | Laura | NaN | 1 |
| 8 | Kevin | 8.0 | 7 |
| 9 | Jonas | 19.0 | 1 |

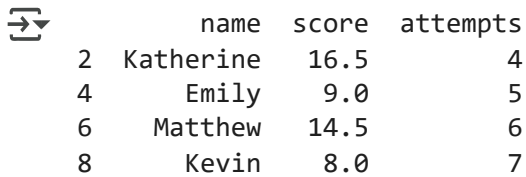
```
selected_columns = df[["name", "score"]]
print(selected_columns)
```



| | name | score |
|---|-----------|-------|
| 0 | Anastasia | 12.5 |
| 1 | Dima | 9.0 |
| 2 | Katherine | 16.5 |
| 3 | James | NaN |
| 4 | Emily | 9.0 |
| 5 | Michael | 20.0 |
| 6 | Matthew | 14.5 |
| 7 | Laura | NaN |
| 8 | Kevin | 8.0 |
| 9 | Jonas | 19.0 |

2. For the above dataframe, Write a program to select the data who's attempt is greater than 3.

```
attempts_greater = df[df['attempts'] > 3]
print(attempts_greater)
```



| | name | score | attempts |
|---|-----------|-------|----------|
| 2 | Katherine | 16.5 | 4 |
| 4 | Emily | 9.0 | 5 |
| 6 | Matthew | 14.5 | 6 |
| 8 | Kevin | 8.0 | 7 |

3. Write python code for indexing rows and columns based on the following conditions:

Assume we have the following dataframe:

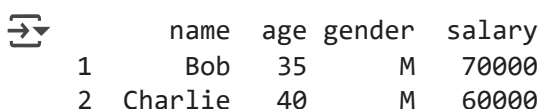
```
data = {'name': ['Alice', 'Bob', 'Charlie', 'Dave'],
        'age': [25, 35, 40, 28],
        'gender': ['F', 'M', 'M', 'M'],
        'salary': [50000, 70000, 60000, 80000]}

df = pd.DataFrame(data)
```

```
data = {'name': ['Alice', 'Bob', 'Charlie', 'Dave'], 'age': [25, 35, 40, 28], 'gender': ['F', 'M', 'M', 'M'], 'salary': [50000, 70000, 60000, 80000]}
df = pd.DataFrame(data)
```

- a. Select rows where age is greater than 30

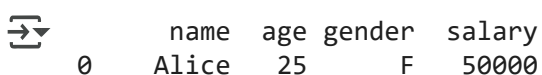
```
rows = df[df["age"]>30]
print(rows)
```



| | name | age | gender | salary |
|---|---------|-----|--------|--------|
| 1 | Bob | 35 | M | 70000 |
| 2 | Charlie | 40 | M | 60000 |

- b. Select rows where name contains 'e'

```
rows_e = df[df["name"].str.contains("e")]
print(rows_e)
```




| | name | age | gender | salary |
|---|-------|-----|--------|--------|
| 0 | Alice | 25 | F | 50000 |

| | | | | |
|---|---------|----|---|-------|
| 2 | Charlie | 40 | M | 60000 |
| 3 | Dave | 28 | M | 80000 |

c. Select rows where gender is 'M' and salary is greater than 65000


```
rows_condition = df[(df["salary"] > 65000) & (df["gender"] == "M")]
print(rows_condition)
```



| | name | age | gender | salary |
|---|------|-----|--------|--------|
| 1 | Bob | 35 | M | 70000 |
| 3 | Dave | 28 | M | 80000 |

d. Select columns 'name' and 'age'


```
slected_columns = df[["name","age"]]
print(slected_columns)
```



| | name | age |
|---|---------|-----|
| 0 | Alice | 25 |
| 1 | Bob | 35 |
| 2 | Charlie | 40 |
| 3 | Dave | 28 |

e. Select columns 'name' and 'salary' where age is less than or equal to 30

```
selected_columns = df[df["age"]<=30][["name","salary"]]
print(selected_columns)
```



| | name | salary |
|---|-------|--------|
| 0 | Alice | 50000 |
| 3 | Dave | 80000 |