Name	Ayush Y Khanapure
Roll No.	2301083
Subject	Python

Assignment on MongoDB, NumPy and Pandas

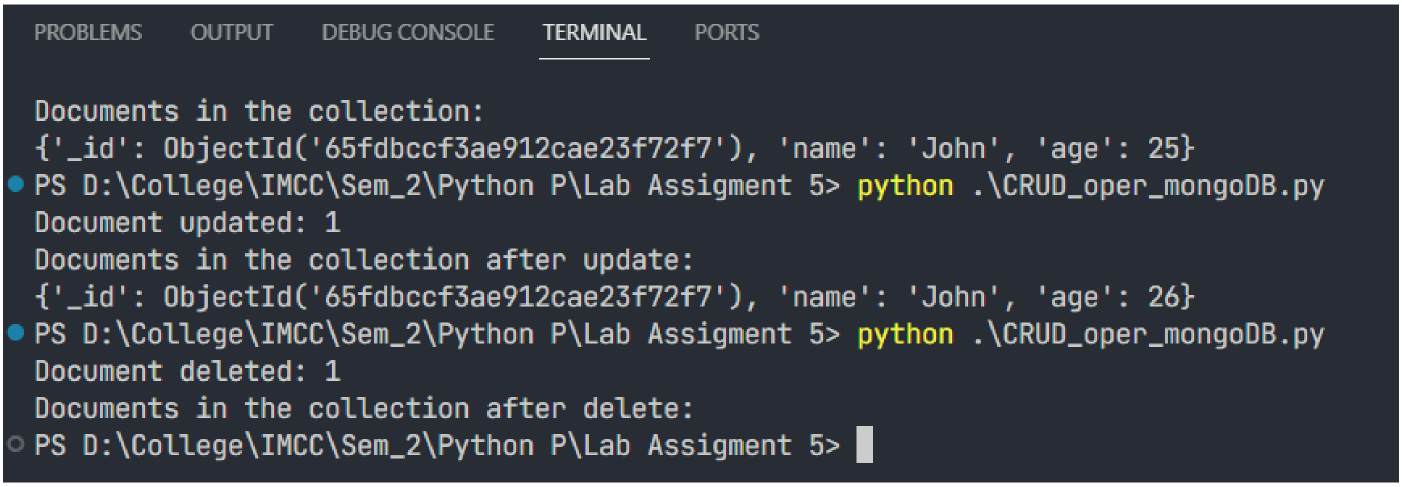
1. Write a program to perform CRUD operations using mongoDB shell and pycharm IDE, C-Create, R-Read, U-Update, D-Delete.

Ans –

```
from pymongo import MongoClient

client = MongoClient('localhost', 27017)
db = client['test_db']
collection = db['test_collection']

def create_document(data):
    result = collection.insert_one(data)
    print("Document created with id:", result.inserted_id)

def read_documents(query={}):
    documents = collection.find(query)
    for doc in documents:
        print(doc)

def update_document(query, new_data):
    result = collection.update_one(query, {"$set": new_data})
    print("Document updated:", result.modified_count)

def delete_document(query):
    result = collection.delete_one(query)
    print("Document deleted:", result.deleted_count)

if __name__ == "__main__":
    create_document({"name": "John", "age": 25})

    print("Documents in the collection:")
    read_documents()

    update_document({"name": "John"}, {"age": 26})

    print("Documents in the collection after update:")
    read_documents()

    delete_document({"name": "John"})

    print("Documents in the collection after delete:")
```

read_documents()

Output :



```
Documents in the collection:
{'_id': ObjectId('65fdbccf3ae912cae23f72f7'), 'name': 'John', 'age': 25}
PS D:\College\IMCC\Sem_2\Python P\Lab Assigment 5> python .\CRUD_oper_mongoDB.py
Document updated: 1
Documents in the collection after update:
{'_id': ObjectId('65fdbccf3ae912cae23f72f7'), 'name': 'John', 'age': 26}
PS D:\College\IMCC\Sem_2\Python P\Lab Assigment 5> python .\CRUD_oper_mongoDB.py
Document deleted: 1
Documents in the collection after delete:
PS D:\College\IMCC\Sem_2\Python P\Lab Assigment 5>
```

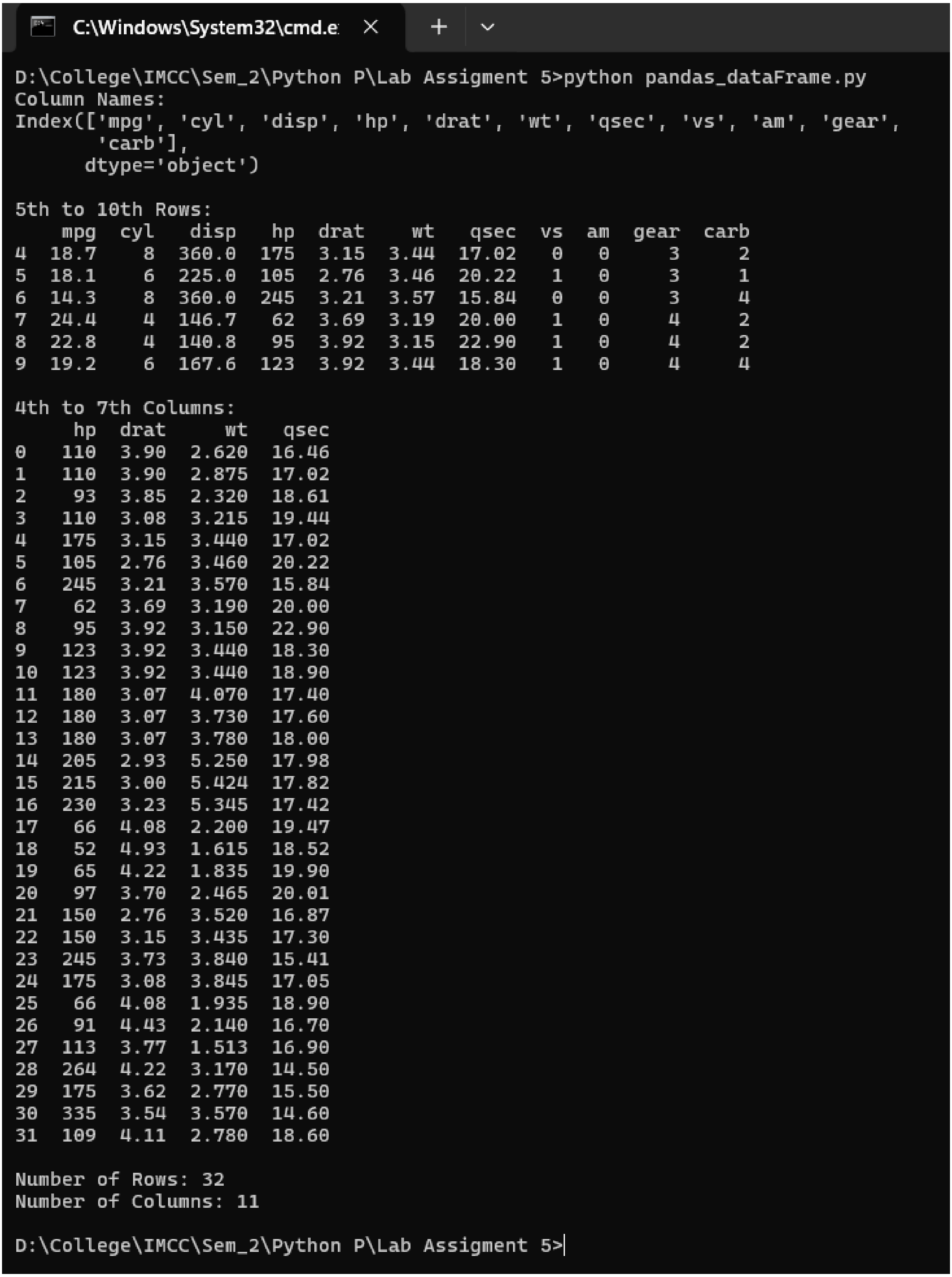
2. Create a pandas dataFrame using mtcars.csv CSV file and perform a following.

a) Display column names
b) Display 5th to 10th rows
c) Display 4th to 7th columns
d) Display no of rows and no of columns

Ans –

```
import pandas as pd

df = pd.read_csv("mtcars.csv")

print("Column Names:")
print(df.columns)

print("\n5th to 10th Rows:")
print(df.iloc[4:10])

print("\n4th to 7th Columns:")
print(df.iloc[:, 3:7])

num_rows, num_columns = df.shape
print("\nNumber of Rows:", num_rows)
print("Number of Columns:", num_columns)
```

Output :



```
D:\College\IMCC\Sem_2\Python P\Lab Assigment 5>python pandas_dataFrame.py
Column Names:
Index(['mpg', 'cyl', 'disp', 'hp', 'drat', 'wt', 'qsec', 'vs', 'am', 'gear',
       'carb'],
      dtype='object')

5th to 10th Rows:
    mpg  cyl   disp   hp  drat    wt   qsec  vs  am  gear  carb
4  18.7    8  360.0  175  3.15  3.44  17.02   0   0     3     2
5  18.1    6  225.0  105  2.76  3.46  20.22   1   0     3     1
6  14.3    8  360.0  245  3.21  3.57  15.84   0   0     3     4
7  24.4    4  146.7   62  3.69  3.19  20.00   1   0     4     2
8  22.8    4  140.8   95  3.92  3.15  22.90   1   0     4     2
9  19.2    6  167.6  123  3.92  3.44  18.30   1   0     4     4

4th to 7th Columns:
     hp  drat     wt   qsec
0   110  3.90  2.620  16.46
1   110  3.90  2.875  17.02
2    93  3.85  2.320  18.61
3   110  3.08  3.215  19.44
4   175  3.15  3.440  17.02
5   105  2.76  3.460  20.22
6   245  3.21  3.570  15.84
7    62  3.69  3.190  20.00
8    95  3.92  3.150  22.90
9   123  3.92  3.440  18.30
10  123  3.92  3.440  18.90
11  180  3.07  4.070  17.40
12  180  3.07  3.730  17.60
13  180  3.07  3.780  18.00
14  205  2.93  5.250  17.98
15  215  3.00  5.424  17.82
16  230  3.23  5.345  17.42
17   66  4.08  2.200  19.47
18   52  4.93  1.615  18.52
19   65  4.22  1.835  19.90
20   97  3.70  2.465  20.01
21  150  2.76  3.520  16.87
22  150  3.15  3.435  17.30
23  245  3.73  3.840  15.41
24  175  3.08  3.845  17.05
25   66  4.08  1.935  18.90
26   91  4.43  2.140  16.70
27  113  3.77  1.513  16.90
28  264  4.22  3.170  14.50
29  175  3.62  2.770  15.50
30  335  3.54  3.570  14.60
31  109  4.11  2.780  18.60

Number of Rows: 32
Number of Columns: 11

D:\College\IMCC\Sem_2\Python P\Lab Assigment 5>
```

3. Use the given file named cricket.csv and perform the following operations:

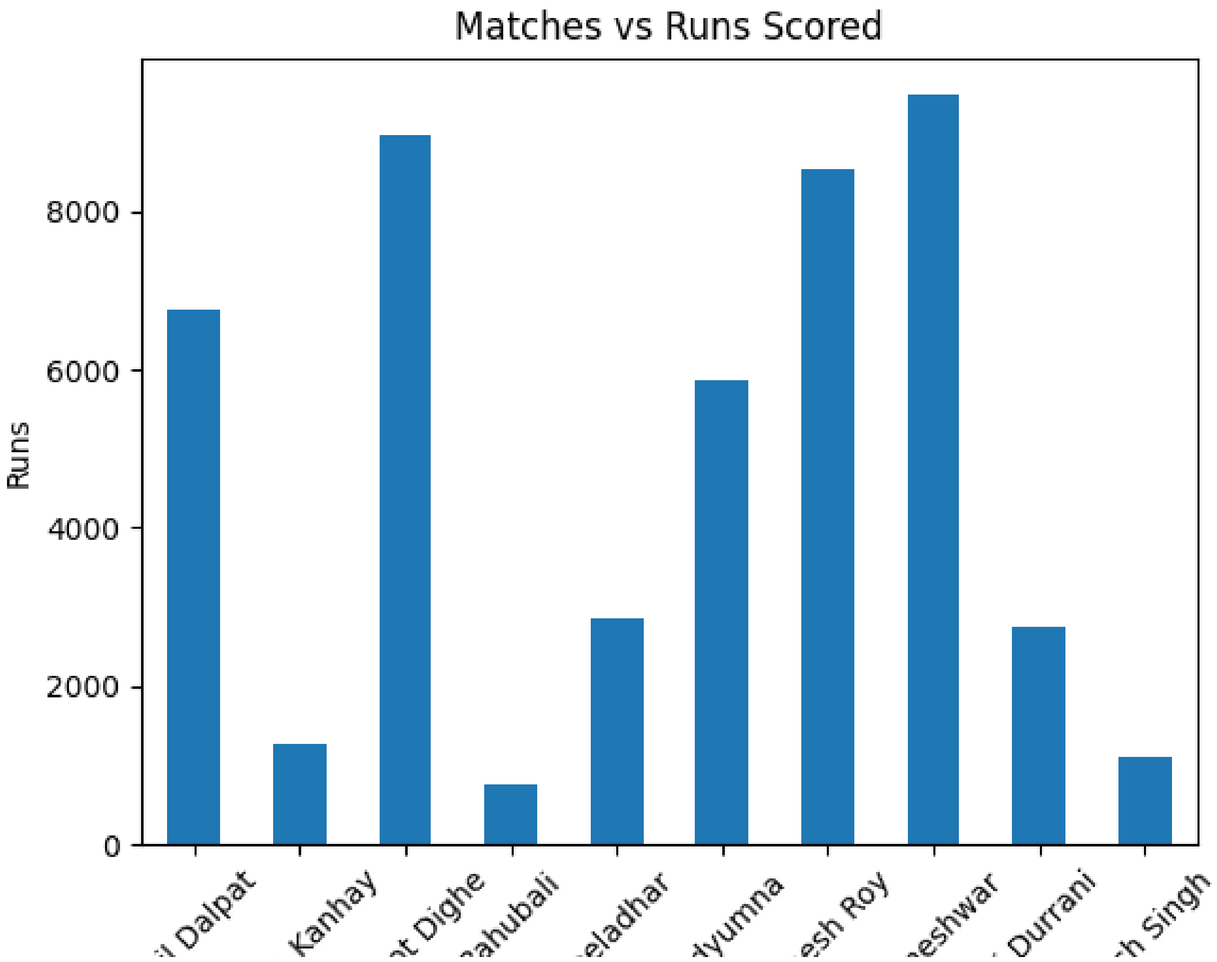
1. Read the file in DataFrame
2. List the name of cricketer and their respective runs
3. Find total wickets taken by them
4. Find average of catches taken
5. Find the name of wicketkeeper
6. Print the name of bowler who played highest number of matches
7. Find average of all the bowlers
8. Find name of the bowler with least bowling average
9. Draw the bar chart of matches against number of runs scored
10. Sort and print information about players by ascending order of runs
11. Print the names of players whose wickets are greater than matches

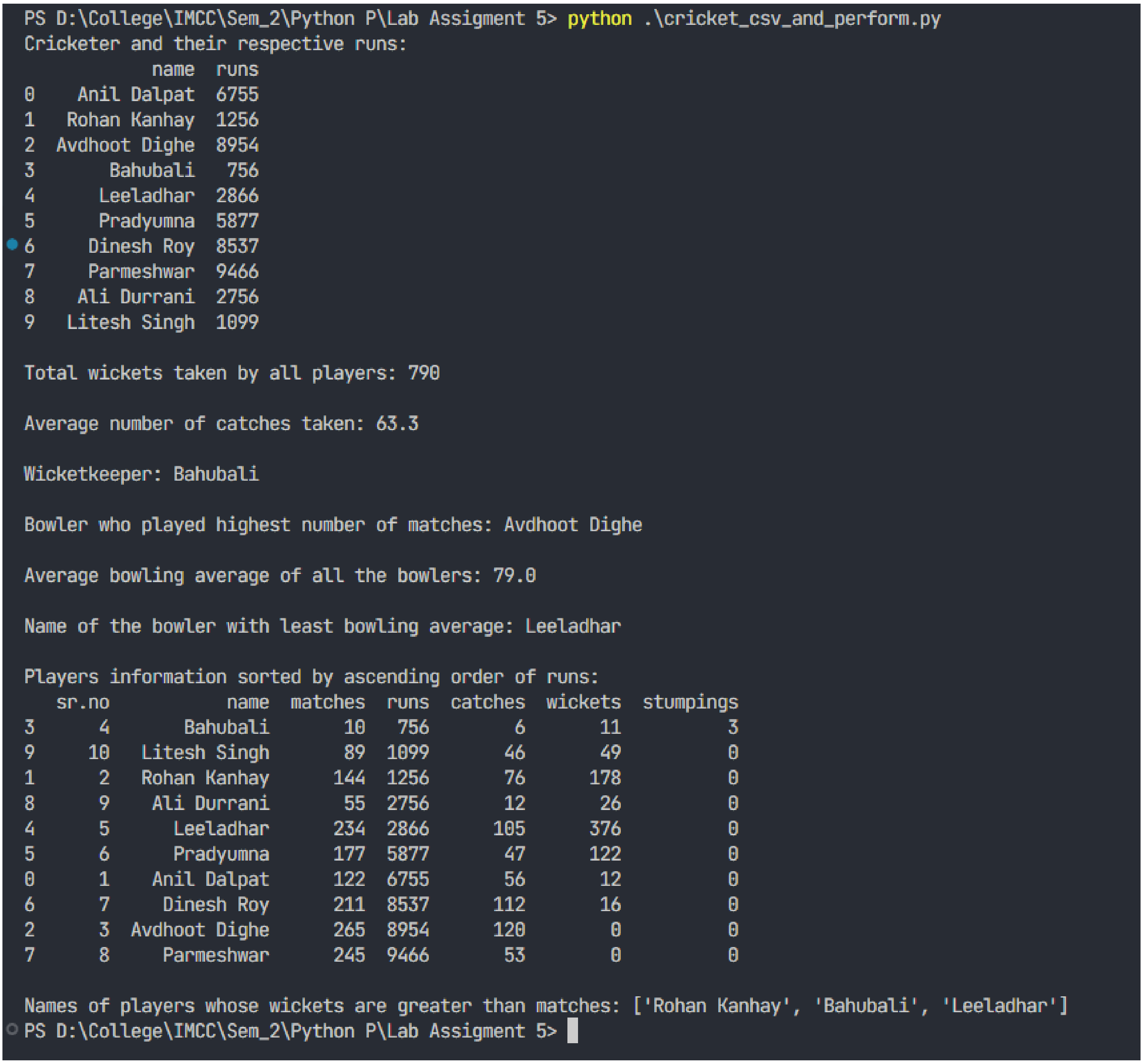
Ans –

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

df = pd.read_csv('cricket.csv')

print("Cricketer and their respective runs:")
print(df[['name', 'runs']])

total_wickets = df['wickets'].sum()
print("\nTotal wickets taken by all players:", total_wickets)

average_catches = df['catches'].mean()
print("\nAverage number of catches taken:", average_catches)

wicketkeeper = df.loc[df['stumpings'] > 0, 'name'].values
if len(wicketkeeper) > 0:
    print("\nWicketkeeper:", wicketkeeper[0])
else:
    print("\nWicketkeeper not found.")

highest_matches_bowler = df.loc[df['matches'].idxmax(), 'name']
print("\nBowler who played highest number of matches:", highest_matches_bowler)

bowlers_average = df['wickets'].astype(float).mean()
print("\nAverage bowling average of all the bowlers:", bowlers_average)

least_average_bowler = df.loc[df['wickets'] > 0, 'name'].iloc[df.loc[df['wickets'] > 0, 'wickets'].idxmin()]
print("\nName of the bowler with least bowling average:", least_average_bowler)

df.plot(kind='bar', x='name', y='runs', legend=None)
plt.title('Matches vs Runs Scored')
plt.xlabel('Name')
plt.ylabel('Runs')
plt.xticks(rotation=45)
plt.show()
```

```
sorted_df_by_runs = df.sort_values(by='runs')
print("\nPlayers information sorted by ascending order of runs:")
print(sorted_df_by_runs)

players_with_more_wickets_than_matches = df[df['wickets'] > df['matches']]['name'].tolist()
print("\nNames of players whose wickets are greater than matches:",
players_with_more_wickets_than_matches)
```

Output :





```
PS D:\College\IMCC\Sem_2\Python P\Lab Assigment 5> python .\cricket_csv_and_perform.py
Cricketer and their respective runs:
            name  runs
0    Anil Dalpat  6755
1   Rohan Kanhay  1256
2  Avdhoot Dighe  8954
3       Bahubali   756
4      Leeladhar  2866
5      Pradyumna  5877
6     Dinesh Roy  8537
7     Parmeshwar  9466
8    Ali Durrani  2756
9   Litesh Singh  1099

Total wickets taken by all players: 790

Average number of catches taken: 63.3

Wicketkeeper: Bahubali

Bowler who played highest number of matches: Avdhoot Dighe

Average bowling average of all the bowlers: 79.0

Name of the bowler with least bowling average: Leeladhar

Players information sorted by ascending order of runs:
   sr.no           name  matches  runs  catches  wickets  stumpings
3      4       Bahubali       10   756        6       11          3
9     10   Litesh Singh       89  1099       46       49          0
1      2   Rohan Kanhay      144  1256       76      178          0
8      9    Ali Durrani       55  2756       12       26          0
4      5      Leeladhar      234  2866      105      376          0
5      6      Pradyumna      177  5877       47      122          0
0      1    Anil Dalpat      122  6755       56       12          0
6      7     Dinesh Roy      211  8537      112       16          0
2      3  Avdhoot Dighe      265  8954      120        0          0
7      8     Parmeshwar      245  9466       53        0          0

Names of players whose wickets are greater than matches: ['Rohan Kanhay', 'Bahubali', 'Leeladhar']
PS D:\College\IMCC\Sem_2\Python P\Lab Assigment 5>
```