**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Roll No: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

***Dataset / File name: Employee.csv***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **RegNo** | **Name** | **Department** | **Salary** | **Age** | **Bloodgroup** | **Scale** |
| **01** | **Aslam** | **BBA** | **25000** | **50** | **B+** | **17** |
| **02** | **Ali** | **BBA** |  | **42** | **O+** | **18** |
| **03** | **Khan** | **BBA** | **75000** | **35** | **B-** | **N/A** |
| **04** | **Haider** | **CS** | **99000** | **46** | **A+** | **17** |
| **05** | **Alam** | **CS** | **65000** | **45** | **AB+** | **18** |
| **03** | **Khan** | **BBA** | **75000** | **35** | **B-** | **N/A** |
| **06** | **John** | **CS** |  |  | **B+** | **18** |

**Note: Scale is object**

data = pd.read\_csv("Book1.csv")

1. ***Select all employees of BBA department***

**data[data["Department"] == "BBA"]**

1. ***Select all employees of BBA department and age is more than 40***

**data[(data["Department"] == "BBA") & (data["Age"] > 40)]**

1. ***Select all employees of BBA department and age is more than 40 and bloodgroup is B+***

***data[(data["Department"] == "BBA") & (data["Age"] > 40) & (data["Bloodgroup"] == "B+")]***

1. ***Fill missing records of Salary with its median***

***data["Salary"].fillna(data["Salary"].median(), inplace=True)***

1. ***Drop only the last duplicate records***

**data.drop\_duplicates(inplace=True, keep="first")**

1. ***How to check if there is a relationship between Age and Salary***

***data.corr()***

1. ***Which department has the most employees***

***data["Department"].value\_counts()***

1. ***Select all records from CS department which have no missing values***

***CS = data[data['Department'] == "CS"]***

***CS = CS.dropna()***

1. ***Select record of highest paid employee from BBA department***

***bba = data[data['Department'] == "BBA"]***

***bba[bba['Salary'] == bba['Salary'].max()]***

1. ***Select all employees having scale more than 17***

***data["Scale"].fillna(0,inplace=True)***

***pd.to\_numeric(data['Scale'])***

***data[data["Scale"]>17]***

1. ***What happens when you use describe on int and str?***

***On int it gives different measures like mean, median, percentile etc***

***On string it gives count, unique, top, frequency***