

# qaisar\_imtiaz\_data\_analysis

July 18, 2025

## 1 Mini Project

```
[2]: import pandas as pd
```

```
[4]: # 1. Using Dictionaries, create a random dataset related to employees
employees_data = {
    "ID" : [101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113,
    ↪114, 115, 116, 117, 118, 119, 120],
    "Name" : ["Asad", "Qaisar", "Javed", "Abid", "Naeem", "Adil", "Atif",
    ↪"Abrar", "shahid", "Detha", "Majid", "Sajid", "Rizwan", "Toqeer", "Junaid",
    ↪"Shakeel", "Rafaqat", "Tariq", "Danish", "Sami"],
    "Father_Name" : ["Islam", "Imtiaz", "Shoukat", "Sadiq", "Sadeeq", "Ameer",
    ↪"Hussain", "Hayat", "Pera", "Shoukat", "Sadiq", "Mehboob", "Mumtaz",
    ↪"Illahi", "Khair Illahi", "Hussain", "Basheer", "basir", "Ejaz", "Shakeel"],
    "Age" : [40, 34, 54, 34, 43, 23, 32, 40, 41, 42, 39, 36, 36, 45, 46, 38,
    ↪37, 36, 34, 32],
    "Department" : ["AA", "AA", "AA", "IT", "IT", "IT", "IT", "IT", "MD", "MD",
    ↪"MD", "MD", "MD", "MD", "MD", "SD", "SD", "HR", "HR", "HR"],
    "Salary" : [11000, 11000, 12000, 14000, 175000, 135000, 180000, 190000,
    ↪192000, 210000, 220000, 230000, 95000, 82000, 360000, 210000, 192000,
    ↪135000, 120000, 115000],
    "Education" : ["Bachelors", "Masters", "PhD", "Bachelors", "Masters",
    ↪"PhD", "Bachelors", "Masters", "PhD", "Bachelors", "Masters", "PhD",
    ↪"Bachelors", "Masters", "PhD", "Bachelors", "Masters", "PhD", "Bachelors",
    ↪"Masters"],
    "City" : ["Lahore", "Karachi", "Islamabad", "Multan", "Faisalabad",
    ↪"Lahore", "Karachi", "Islamabad", "Multan", "Faisalabad", "Lahore",
    ↪"Karachi", "Islamabad", "Multan", "Faisalabad", "Lahore", "Karachi",
    ↪"Islamabad", "Multan", "Faisalabad"],
    "Address" : ["Street 10", "Street 5", "Block A", "Sector 3", "Avenue 2",
    ↪"Street 10", "Street 5", "Block A", "Sector 3", "Avenue 2", "Street 10",
    ↪"Street 5", "Block A", "Sector 3", "Avenue 2", "Street 10", "Street 5",
    ↪"Block A", "Sector 3", "Avenue 2"],
}

# 2.Convert the dictionary into a DataFrame
```

```
emp_df = pd.DataFrame(employees_data)
```

```
#View DataFrame
```

```
emp_df
```

```
[4]:
```

	ID	Name	Father_Name	Age	Department	Salary	Education	City \
0	101	Asad	Islam	40	AA	11000	Bachelors	Lahore
1	102	Qaisar	Imtiaz	34	AA	11000	Masters	Karachi
2	103	Javed	Shoukat	54	AA	12000	PhD	Islamabad
3	104	Abid	Sadiq	34	IT	14000	Bachelors	Multan
4	105	Naeem	Sadeeq	43	IT	175000	Masters	Faisalabad
5	106	Adil	Ameer	23	IT	135000	PhD	Lahore
6	107	Atif	Hussain	32	IT	180000	Bachelors	Karachi
7	108	Abrar	Hayat	40	IT	190000	Masters	Islamabad
8	109	shahid	Pera	41	MD	192000	PhD	Multan
9	110	Detha	Shoukat	42	MD	210000	Bachelors	Faisalabad
10	111	Majid	Sadiq	39	MD	220000	Masters	Lahore
11	112	Sajid	Mehboob	36	MD	230000	PhD	Karachi
12	113	Rizwan	Mumtaz	36	MD	95000	Bachelors	Islamabad
13	114	Toqeer	Illahi	45	MD	82000	Masters	Multan
14	115	Junaid	Khair Illahi	46	MD	360000	PhD	Faisalabad
15	116	Shakeel	Hussain	38	SD	210000	Bachelors	Lahore
16	117	Rafaqat	Basheer	37	SD	192000	Masters	Karachi
17	118	Tariq	basir	36	HR	135000	PhD	Islamabad
18	119	Danish	Ejaz	34	HR	120000	Bachelors	Multan
19	120	Sami	Shakeel	32	HR	115000	Masters	Faisalabad

```

      Address
0  Street 10
1  Street 5
2  Block A
3  Sector 3
4  Avenue 2
5  Street 10
6  Street 5
7  Block A
8  Sector 3
9  Avenue 2
10 Street 10
11 Street 5
12 Block A
13 Sector 3
14 Avenue 2
15 Street 10
16 Street 5
17 Block A
18 Sector 3

```

## 1.1 3. Basic Operations

```
[131]: # 3. Basic Operations
print("Info: ", emp_df.info()) #info
print("\nDescribe: ", emp_df.describe()) # description
print("\nSize: ", emp_df.size) # size multiple rows and columns
print("\nShape: ", emp_df.shape) #Show number of rows and columns
print("\nShow All Columns names: ", emp_df.columns.tolist()) #Show name of all
    ↪ columns
print("Show first three rows\n", emp_df.head(3)) #Show first three rows
print("Show last three rows\n", emp_df.tail(3)) #Show last three rows
print("\nMissing Values (True means missing):\n", emp_df.isnull()) # if no null
    ↪ show false
print("\nNon-Missing Values (True means not missing):\n", emp_df.notnull()) #if
    ↪ no null show true
print("\nTotal Missing Values Per Column:\n", emp_df.isnull().sum()) #show
    ↪ total number of null

print("\nTotal number of Department:\n", emp_df['Department'].count()) #Total
    ↪ number of Department
print("\nTotal number of Department per category:\n", emp_df['Department'].
    ↪ value_counts()) #sTotal number of Department per category
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 20 entries, 0 to 19
```

```
Data columns (total 9 columns):
```

#	Column	Non-Null Count	Dtype
0	ID	20 non-null	int64
1	Name	20 non-null	object
2	Father_Name	20 non-null	object
3	Age	20 non-null	int64
4	Department	20 non-null	object
5	Salary	20 non-null	int64
6	Education	20 non-null	object
7	City	20 non-null	object
8	Address	20 non-null	object

```
dtypes: int64(3), object(6)
```

```
memory usage: 1.5+ KB
```

```
Info: None
```

Describe:	ID	Age	Salary
count	20.00000	20.00000	20.000000
mean	110.50000	38.10000	144450.000000
std	5.91608	6.43101	90333.577717

```

min    101.00000  23.00000  11000.000000
25%    105.75000  34.00000  91750.000000
50%    110.50000  37.50000  155000.000000
75%    115.25000  41.25000  196500.000000
max    120.00000  54.00000  360000.000000

```

Size: 180

Shape: (20, 9)

Show All Columns names: ['ID', 'Name', 'Father\_Name', 'Age', 'Department', 'Salary', 'Education', 'City', 'Address']

Show first three rows

	ID	Name	Father_Name	Age	Department	Salary	Education	City	\
0	101	Asad	Islam	40	AA	11000	Bachelors	Lahore	
1	102	Qaisar	Imtiaz	34	AA	11000	Masters	Karachi	
2	103	Javed	Shoukat	54	AA	12000	PhD	Islamabad	

Address

```

0 Street 10
1 Street 5
2 Block A

```

Show last three rows

	ID	Name	Father_Name	Age	Department	Salary	Education	City	\
17	118	Tariq	basir	36	HR	135000	PhD	Islamabad	
18	119	Danish	Ejaz	34	HR	120000	Bachelors	Multan	
19	120	Sami	Shakeel	32	HR	115000	Masters	Faisalabad	

Address

```

17 Block A
18 Sector 3
19 Avenue 2

```

Missing Values (True means missing):

	ID	Name	Father_Name	Age	Department	Salary	Education	City	\
0	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	
5	False	False	False	False	False	False	False	False	
6	False	False	False	False	False	False	False	False	
7	False	False	False	False	False	False	False	False	
8	False	False	False	False	False	False	False	False	
9	False	False	False	False	False	False	False	False	
10	False	False	False	False	False	False	False	False	
11	False	False	False	False	False	False	False	False	
12	False	False	False	False	False	False	False	False	

13	False	False	False	False	False	False	False	False	False
14	False	False	False	False	False	False	False	False	False
15	False	False	False	False	False	False	False	False	False
16	False	False	False	False	False	False	False	False	False
17	False	False	False	False	False	False	False	False	False
18	False	False	False	False	False	False	False	False	False
19	False	False	False	False	False	False	False	False	False

#### Address

0	False
1	False
2	False
3	False
4	False
5	False
6	False
7	False
8	False
9	False
10	False
11	False
12	False
13	False
14	False
15	False
16	False
17	False
18	False
19	False

#### Non-Missing Values (True means not missing):

	ID	Name	Father_Name	Age	Department	Salary	Education	City	\
0	True	True	True	True	True	True	True	True	
1	True	True	True	True	True	True	True	True	
2	True	True	True	True	True	True	True	True	
3	True	True	True	True	True	True	True	True	
4	True	True	True	True	True	True	True	True	
5	True	True	True	True	True	True	True	True	
6	True	True	True	True	True	True	True	True	
7	True	True	True	True	True	True	True	True	
8	True	True	True	True	True	True	True	True	
9	True	True	True	True	True	True	True	True	
10	True	True	True	True	True	True	True	True	
11	True	True	True	True	True	True	True	True	
12	True	True	True	True	True	True	True	True	
13	True	True	True	True	True	True	True	True	
14	True	True	True	True	True	True	True	True	
15	True	True	True	True	True	True	True	True	

16	True	True	True	True	True	True	True	True	True
17	True	True	True	True	True	True	True	True	True
18	True	True	True	True	True	True	True	True	True
19	True	True	True	True	True	True	True	True	True

Address	
0	True
1	True
2	True
3	True
4	True
5	True
6	True
7	True
8	True
9	True
10	True
11	True
12	True
13	True
14	True
15	True
16	True
17	True
18	True
19	True

Total Missing Values Per Column:

ID	0
Name	0
Father_Name	0
Age	0
Department	0
Salary	0
Education	0
City	0
Address	0

dtype: int64

Total number of Department:

20

Total number of Department per catergory:

Department	
MD	7
IT	5
AA	3
HR	3

```
SD      2
Name: count, dtype: int64
```

```
[30]: print("Info: ",emp_df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   ID              20 non-null    int64
 1   Name            20 non-null    object
 2   Father_Name     20 non-null    object
 3   Age             20 non-null    int64
 4   Department      20 non-null    object
 5   Salary          20 non-null    int64
 6   Education       20 non-null    object
 7   City            20 non-null    object
 8   Address         20 non-null    object
dtypes: int64(3), object(6)
memory usage: 1.5+ KB
Info: None
```

```
[46]: #Index
index_st = emp_df.set_index('ID') # ID set to default index
index_st
```

```
[46]:
```

	Name	Father_Name	Age	Department	Salary	Education	City \
ID							
101	Asad	Islam	40	AA	11000	Bachelors	Lahore
102	Qaisar	Imtiaz	34	AA	11000	Masters	Karachi
103	Javed	Shoukat	54	AA	12000	PhD	Islamabad
104	Abid	Sadiq	34	IT	14000	Bachelors	Multan
105	Naeem	Sadeeq	43	IT	175000	Masters	Faisalabad
106	Adil	Ameer	23	IT	135000	PhD	Lahore
107	Atif	Hussain	32	IT	180000	Bachelors	Karachi
108	Abrar	Hayat	40	IT	190000	Masters	Islamabad
109	shahid	Pera	41	MD	192000	PhD	Multan
110	Detha	Shoukat	42	MD	210000	Bachelors	Faisalabad
111	Majid	Sadiq	39	MD	220000	Masters	Lahore
112	Sajid	Mehboob	36	MD	230000	PhD	Karachi
113	Rizwan	Mumtaz	36	MD	95000	Bachelors	Islamabad
114	Toqeer	Illahi	45	MD	82000	Masters	Multan
115	Junaid	Khair Illahi	46	MD	360000	PhD	Faisalabad
116	Shakeel	Hussain	38	SD	210000	Bachelors	Lahore
117	Rafaqat	Basheer	37	SD	192000	Masters	Karachi
118	Tariq	basir	36	HR	135000	PhD	Islamabad

119	Danish	Ejaz	34	HR	120000	Bachelors	Multan
120	Sami	Shakeel	32	HR	115000	Masters	Faisalabad

Address

ID

101	Street 10
102	Street 5
103	Block A
104	Sector 3
105	Avenue 2
106	Street 10
107	Street 5
108	Block A
109	Sector 3
110	Avenue 2
111	Street 10
112	Street 5
113	Block A
114	Sector 3
115	Avenue 2
116	Street 10
117	Street 5
118	Block A
119	Sector 3
120	Avenue 2

## 1.2 4. Using Dictionaries, create 5 more records to the data making a total of 25 records.

[5]: # 4. Using Dictionaries, create 5 more records to the data making a total of 25 records.

```
new_data = {
    "ID" : [121, 122, 123, 124, 125],
    "Name" : ["Asad", "Qaisar", "Javed", "Abid", "Naeem"],
    "Father_Name" : ["Islam", "Imtiaz", "Shoukat", "Sadiq", "Sadeeq"],
    "Age" : [40, 34, 54, 34, 43],
    "Department" : ["AA", "AA", "AA", "IT", "IT"],
    "Salary" : [180000, 190000, 192000, 210000, 220000],
    "Education" : ["Bachelors", "Masters", "PhD", "Bachelors", "Masters"],
    "City" : ["Karachi", "Islamabad", "Multan", "Faisalabad", "Faisalabad"],
    "Address" : ["Block A", "Sector 3", "Avenue 2", "Street 10", "Street 5"],
}
```

# 2. Convert the dictionary into a DataFrame

```
new_df = pd.DataFrame(new_data)
```

#View DataFrame



```
new_df
```

```
[5]:
```

	ID	Name	Father_Name	Age	Department	Salary	Education	City \
0	121	Asad	Islam	40	AA	180000	Bachelors	Karachi
1	122	Qaisar	Imtiaz	34	AA	190000	Masters	Islamabad
2	123	Javed	Shoukat	54	AA	192000	PhD	Multan
3	124	Abid	Sadiq	34	IT	210000	Bachelors	Faisalabad
4	125	Naeem	Sadeeq	43	IT	220000	Masters	Faisalabad

```
Address
```

0	Block A
1	Sector 3
2	Avenue 2
3	Street 10
4	Street 5

```
[6]: #concat the new and curent dataset
updated_df = pd.concat([emp_df , new_df], ignore_index = True)
updated_df
```

```
[6]:
```

	ID	Name	Father_Name	Age	Department	Salary	Education	City \
0	101	Asad	Islam	40	AA	11000	Bachelors	Lahore
1	102	Qaisar	Imtiaz	34	AA	11000	Masters	Karachi
2	103	Javed	Shoukat	54	AA	12000	PhD	Islamabad
3	104	Abid	Sadiq	34	IT	14000	Bachelors	Multan
4	105	Naeem	Sadeeq	43	IT	175000	Masters	Faisalabad
5	106	Adil	Ameer	23	IT	135000	PhD	Lahore
6	107	Atif	Hussain	32	IT	180000	Bachelors	Karachi
7	108	Abrar	Hayat	40	IT	190000	Masters	Islamabad
8	109	shahid	Pera	41	MD	192000	PhD	Multan
9	110	Detha	Shoukat	42	MD	210000	Bachelors	Faisalabad
10	111	Majid	Sadiq	39	MD	220000	Masters	Lahore
11	112	Sajid	Mehboob	36	MD	230000	PhD	Karachi
12	113	Rizwan	Mumtaz	36	MD	95000	Bachelors	Islamabad
13	114	Toqeer	Illahi	45	MD	82000	Masters	Multan
14	115	Junaid	Khair Illahi	46	MD	360000	PhD	Faisalabad
15	116	Shakeel	Hussain	38	SD	210000	Bachelors	Lahore
16	117	Rafaqat	Basheer	37	SD	192000	Masters	Karachi
17	118	Tariq	basir	36	HR	135000	PhD	Islamabad
18	119	Danish	Ejaz	34	HR	120000	Bachelors	Multan
19	120	Sami	Shakeel	32	HR	115000	Masters	Faisalabad
20	121	Asad	Islam	40	AA	180000	Bachelors	Karachi
21	122	Qaisar	Imtiaz	34	AA	190000	Masters	Islamabad
22	123	Javed	Shoukat	54	AA	192000	PhD	Multan
23	124	Abid	Sadiq	34	IT	210000	Bachelors	Faisalabad
24	125	Naeem	Sadeeq	43	IT	220000	Masters	Faisalabad

	Address
0	Street 10
1	Street 5
2	Block A
3	Sector 3
4	Avenue 2
5	Street 10
6	Street 5
7	Block A
8	Sector 3
9	Avenue 2
10	Street 10
11	Street 5
12	Block A
13	Sector 3
14	Avenue 2
15	Street 10
16	Street 5
17	Block A
18	Sector 3
19	Avenue 2
20	Block A
21	Sector 3
22	Avenue 2
23	Street 10
24	Street 5

```
[7]: # Rename column names (e.g., change 'name' to 'employee_name' and 'id' to
      ↪ 'employee_id').
      update_column_df = updated_df.rename(columns = { 'ID' : 'Employee_id' , 'Name' :
      ↪ 'Employee_name'})
      update_column_df
```

```
[7]:
```

	Employee_id	Employee_name	Father_Name	Age	Department	Salary	\
0	101	Asad	Islam	40	AA	11000	
1	102	Qaisar	Imtiaz	34	AA	11000	
2	103	Javed	Shoukat	54	AA	12000	
3	104	Abid	Sadiq	34	IT	14000	
4	105	Naeem	Sadeeq	43	IT	175000	
5	106	Adil	Ameer	23	IT	135000	
6	107	Atif	Hussain	32	IT	180000	
7	108	Abrar	Hayat	40	IT	190000	
8	109	shahid	Pera	41	MD	192000	
9	110	Detha	Shoukat	42	MD	210000	
10	111	Majid	Sadiq	39	MD	220000	
11	112	Sajid	Mehboob	36	MD	230000	
12	113	Rizwan	Mumtaz	36	MD	95000	

13	114	Toqeer	Illahi	45	MD	82000
14	115	Junaid	Khair Illahi	46	MD	360000
15	116	Shakeel	Hussain	38	SD	210000
16	117	Rafaqat	Basheer	37	SD	192000
17	118	Tariq	basir	36	HR	135000
18	119	Danish	Ejaz	34	HR	120000
19	120	Sami	Shakeel	32	HR	115000
20	121	Asad	Islam	40	AA	180000
21	122	Qaisar	Imtiaz	34	AA	190000
22	123	Javed	Shoukat	54	AA	192000
23	124	Abid	Sadiq	34	IT	210000
24	125	Naeem	Sadeeq	43	IT	220000

	Education	City	Address
0	Bachelors	Lahore	Street 10
1	Masters	Karachi	Street 5
2	PhD	Islamabad	Block A
3	Bachelors	Multan	Sector 3
4	Masters	Faisalabad	Avenue 2
5	PhD	Lahore	Street 10
6	Bachelors	Karachi	Street 5
7	Masters	Islamabad	Block A
8	PhD	Multan	Sector 3
9	Bachelors	Faisalabad	Avenue 2
10	Masters	Lahore	Street 10
11	PhD	Karachi	Street 5
12	Bachelors	Islamabad	Block A
13	Masters	Multan	Sector 3
14	PhD	Faisalabad	Avenue 2
15	Bachelors	Lahore	Street 10
16	Masters	Karachi	Street 5
17	PhD	Islamabad	Block A
18	Bachelors	Multan	Sector 3
19	Masters	Faisalabad	Avenue 2
20	Bachelors	Karachi	Block A
21	Masters	Islamabad	Sector 3
22	PhD	Multan	Avenue 2
23	Bachelors	Faisalabad	Street 10
24	Masters	Faisalabad	Street 5

### 1.3 5. Filter the data using conditions by applying comparison and logical operators

```
[141]: # 5. Filter the data using conditions by applying comparison and logical
      ↪ operators

      #Comparison Operators Equal to
```

```
# show record of employee that have ID equal to 101
updated_df[updated_df['ID'] == 101]
```

```
[141]:      ID  Name  Father_Name  Age  Department  Salary  Education  City  Address
0   101   Asad      Islam    40          AA   11000  Bachelors  Lahore  Street 10
```

```
[142]: #Comparison Operators greater than
# show record of employee that have salary greater than to 90000
updated_df[updated_df['Salary'] > 230000]
```

```
[142]:      ID  Name  Father_Name  Age  Department  Salary  Education  City \
14   115  Junaid  Khair Illahi   46          MD  360000      PhD  Faisalabad

      Address
14  Avenue 2
```

```
[143]: #Comparison Operators greater than and equal to
# show record of employee that have salary greater than and equal to 230000
updated_df[updated_df['Salary'] >= 230000]
```

```
[143]:      ID  Name  Father_Name  Age  Department  Salary  Education  City \
11   112   Sajid    Mehboob   36          MD  230000      PhD   Karachi
14   115  Junaid  Khair Illahi   46          MD  360000      PhD  Faisalabad

      Address
11  Street 5
14  Avenue 2
```

```
[144]: #Comparison Operators less than
# show record of employee that have salary less than to 90000
updated_df[updated_df['Salary'] < 90000]
```

```
[144]:      ID  Name  Father_Name  Age  Department  Salary  Education  City \
0   101   Asad      Islam    40          AA   11000  Bachelors  Lahore
1   102  Qaisar    Imtiaz   34          AA   11000   Masters  Karachi
2   103   Javed   Shoukat   54          AA   12000      PhD  Islamabad
3   104   Abid    Sadiq    34          IT   14000  Bachelors  Multan
13  114  Toqeer    Illahi   45          MD   82000   Masters  Multan

      Address
0  Street 10
1  Street 5
2   Block A
3   Sector 3
13  Sector 3
```

```
[145]: #Comparison Operators less than and equal
# show record of employee that have salary less than and equal to 90000
updated_df[updated_df['Salary'] <= 90000]
```

```
[145]:
```

	ID	Name	Father_Name	Age	Department	Salary	Education	City	\
0	101	Asad	Islam	40	AA	11000	Bachelors	Lahore	
1	102	Qaisar	Imtiaz	34	AA	11000	Masters	Karachi	
2	103	Javed	Shoukat	54	AA	12000	PhD	Islamabad	
3	104	Abid	Sadiq	34	IT	14000	Bachelors	Multan	
13	114	Toqeer	Illahi	45	MD	82000	Masters	Multan	

```

Address
0    Street 10
1    Street 5
2    Block A
3    Sector 3
13   Sector 3

```

```
[146]: #Comparison Operators less than and equal
# show record without employee that have ID = 101
updated_df[updated_df['ID'] != 101]
```

```
[146]:
```

	ID	Name	Father_Name	Age	Department	Salary	Education	City	\
1	102	Qaisar	Imtiaz	34	AA	11000	Masters	Karachi	
2	103	Javed	Shoukat	54	AA	12000	PhD	Islamabad	
3	104	Abid	Sadiq	34	IT	14000	Bachelors	Multan	
4	105	Naeem	Sadeeq	43	IT	175000	Masters	Faisalabad	
5	106	Adil	Ameer	23	IT	135000	PhD	Lahore	
6	107	Atif	Hussain	32	IT	180000	Bachelors	Karachi	
7	108	Abrar	Hayat	40	IT	190000	Masters	Islamabad	
8	109	shahid	Pera	41	MD	192000	PhD	Multan	
9	110	Detha	Shoukat	42	MD	210000	Bachelors	Faisalabad	
10	111	Majid	Sadiq	39	MD	220000	Masters	Lahore	
11	112	Sajid	Mehboob	36	MD	230000	PhD	Karachi	
12	113	Rizwan	Mumtaz	36	MD	95000	Bachelors	Islamabad	
13	114	Toqeer	Illahi	45	MD	82000	Masters	Multan	
14	115	Junaid	Khair Illahi	46	MD	360000	PhD	Faisalabad	
15	116	Shakeel	Hussain	38	SD	210000	Bachelors	Lahore	
16	117	Rafaqat	Basheer	37	SD	192000	Masters	Karachi	
17	118	Tariq	basir	36	HR	135000	PhD	Islamabad	
18	119	Danish	Ejaz	34	HR	120000	Bachelors	Multan	
19	120	Sami	Shakeel	32	HR	115000	Masters	Faisalabad	
20	121	Asad	Islam	40	AA	180000	Bachelors	Karachi	
21	122	Qaisar	Imtiaz	34	AA	190000	Masters	Islamabad	
22	123	Javed	Shoukat	54	AA	192000	PhD	Multan	
23	124	Abid	Sadiq	34	IT	210000	Bachelors	Faisalabad	
24	125	Naeem	Sadeeq	43	IT	220000	Masters	Faisalabad	

	Address
1	Street 5
2	Block A
3	Sector 3
4	Avenue 2
5	Street 10
6	Street 5
7	Block A
8	Sector 3
9	Avenue 2
10	Street 10
11	Street 5
12	Block A
13	Sector 3
14	Avenue 2
15	Street 10
16	Street 5
17	Block A
18	Sector 3
19	Avenue 2
20	Block A
21	Sector 3
22	Avenue 2
23	Street 10
24	Street 5

```
[147]: # logical operators AND Operator
# Salary > 90000 or Age > 45 both condition sill be true
updated_df[(updated_df['Salary'] > 90000) & (updated_df['Age'] > 45)]
```

```
[147]:
```

	ID	Name	Father_Name	Age	Department	Salary	Education	City	\
14	115	Junaid	Khair Illahi	46	MD	360000	PhD	Faisalabad	
22	123	Javed	Shoukat	54	AA	192000	PhD	Multan	

	Address
14	Avenue 2
22	Avenue 2

```
[148]: # logical operators OR Oprator
# Salary > 300000 or Age > 45 any condition will be true
updated_df[(updated_df['Salary'] > 300000) | (updated_df['Age'] > 45)]
```

```
[148]:
```

	ID	Name	Father_Name	Age	Department	Salary	Education	City	\
2	103	Javed	Shoukat	54	AA	12000	PhD	Islamabad	
14	115	Junaid	Khair Illahi	46	MD	360000	PhD	Faisalabad	
22	123	Javed	Shoukat	54	AA	192000	PhD	Multan	

```

        Address
2    Block A
14  Avenue 2
22  Avenue 2

```

```

[149]: # logical operators NOT Operator
# Salary > 100000 and Age NOT > 25
updated_df[(updated_df['Salary'] > 100000) &~ (updated_df['Age'] > 25)]

```

```

[149]:      ID  Name  Father_Name  Age  Department  Salary  Education  City  Address
5   106  Adil      Ameer    23          IT   135000      PhD  Lahore  Street 10

```

```

[150]: # Filter data using loc and iloc
updated_df.loc[1:3] #access by label there will be show index range 1 to 3

```

```

[150]:      ID  Name  Father_Name  Age  Department  Salary  Education  City \
1   102  Qaisar      Imtiaz   34          AA   11000    Masters  Karachi
2   103   Javed      Shoukat  54          AA   12000      PhD  Islamabad
3   104   Abid      Sadiq   34          IT   14000  Bachelors   Multan

```

```

        Address
1  Street 5
2   Block A
3  Sector 3

```

```

[151]: updated_df.iloc[1:3] #access by index there will be show only 2 rows like index
      ↪- 1 and index will be 1 to 2

```

```

[151]:      ID  Name  Father_Name  Age  Department  Salary  Education  City \
1   102  Qaisar      Imtiaz   34          AA   11000    Masters  Karachi
2   103   Javed      Shoukat  54          AA   12000      PhD  Islamabad

```

```

        Address
1  Street 5
2   Block A

```

#### 1.4 6. Use and apply different methods to the data such as: adding a new column (bonus for each employee), applying grouping and aggregation functions.

```

[105]: # Added new column bonus
updated_df['Bonus'] = [5000, 6000, 5500, 6500, 4500, 5000, 6000, 5500, 6500,
      ↪4500, 5000, 6000, 5500, 6500, 4500, 5000, 6000, 5500, 6500, 4500, 5000,
      ↪6000, 5500, 6500, 4500]
updated_df

```

[105]:

	ID	Name	Father_Name	Age	Department	Salary	Education	City \
0	101	Asad	Islam	40	AA	11000	Bachelors	Lahore
1	102	Qaisar	Imtiaz	34	AA	11000	Masters	Karachi
2	103	Javed	Shoukat	54	AA	12000	PhD	Islamabad
3	104	Abid	Sadiq	34	IT	14000	Bachelors	Multan
4	105	Naeem	Sadeeq	43	IT	175000	Masters	Faisalabad
5	106	Adil	Ameer	23	IT	135000	PhD	Lahore
6	107	Atif	Hussain	32	IT	180000	Bachelors	Karachi
7	108	Abrar	Hayat	40	IT	190000	Masters	Islamabad
8	109	shahid	Pera	41	MD	192000	PhD	Multan
9	110	Detha	Shoukat	42	MD	210000	Bachelors	Faisalabad
10	111	Majid	Sadiq	39	MD	220000	Masters	Lahore
11	112	Sajid	Mehboob	36	MD	230000	PhD	Karachi
12	113	Rizwan	Mumtaz	36	MD	95000	Bachelors	Islamabad
13	114	Toqeer	Illahi	45	MD	82000	Masters	Multan
14	115	Junaid	Khair Illahi	46	MD	360000	PhD	Faisalabad
15	116	Shakeel	Hussain	38	SD	210000	Bachelors	Lahore
16	117	Rafaqat	Basheer	37	SD	192000	Masters	Karachi
17	118	Tariq	basir	36	HR	135000	PhD	Islamabad
18	119	Danish	Ejaz	34	HR	120000	Bachelors	Multan
19	120	Sami	Shakeel	32	HR	115000	Masters	Faisalabad
20	121	Asad	Islam	40	AA	180000	Bachelors	Karachi
21	122	Qaisar	Imtiaz	34	AA	190000	Masters	Islamabad
22	123	Javed	Shoukat	54	AA	192000	PhD	Multan
23	124	Abid	Sadiq	34	IT	210000	Bachelors	Faisalabad
24	125	Naeem	Sadeeq	43	IT	220000	Masters	Faisalabad

	Address	Bonus
0	Street 10	5000
1	Street 5	6000
2	Block A	5500
3	Sector 3	6500
4	Avenue 2	4500
5	Street 10	5000
6	Street 5	6000
7	Block A	5500
8	Sector 3	6500
9	Avenue 2	4500
10	Street 10	5000
11	Street 5	6000
12	Block A	5500
13	Sector 3	6500
14	Avenue 2	4500
15	Street 10	5000
16	Street 5	6000
17	Block A	5500
18	Sector 3	6500



19	Avenue 2	4500
20	Block A	5000
21	Sector 3	6000
22	Avenue 2	5500
23	Street 10	6500
24	Street 5	4500

```
[116]: # Applying grouping and aggregation functions
# maximum salary as per department
updated_df.groupby('Department')['Salary'].max()
```

```
[116]: Department
AA      192000
HR      135000
IT      220000
MD      360000
SD      210000
Name: Salary, dtype: int64
```

```
[115]: # Applying grouping and aggregation functions

# Aggregate multiple columns to show mean for salary, maximum of age and sum
↳ of bonus per department
updated_df.groupby('Department').agg({'Salary': 'mean', 'Age': 'max', 'Bonus':
↳ 'sum'})
```

```
[115]:
```

	Salary	Age	Bonus
Department			
AA	99333.333333	54	33000
HR	123333.333333	36	16500
IT	160571.428571	43	38500
MD	198428.571429	46	38500
SD	201000.000000	38	11000

1.5 7. Merge the existing 25 records with new data. Create a separate dataset with employee\_id, joining\_year, and month\_name for 25 employees and join both datasets using merge() or concat() methods.

```
[ ]:
```

```
[8]: #
emp_joined = {
    "Employee_id" : [101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111,
↳ 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125],
```

```

    "joining_year" : [2020, 2021, 2018, 2019, 2023, 2020, 2021, 2018, 2019,
↪2023, 2020, 2021, 2018, 2019, 2023, 2020, 2021, 2018, 2019, 2023, 2020,
↪2021, 2018, 2019, 2023],
    "month_name" : ["Jan", "Feb", "Mar", "Jan", "June", "Jan", "Feb",
↪"Mar", "Jan", "June", "Jan", "Feb", "Mar", "Jan", "June", "Jan", "Feb",
↪"Mar", "Jan", "June", "Jan", "Feb", "Mar", "Jan", "June"]
}

join_df = pd.DataFrame(emp_joined)
join_df

```

```

[8]:
   Employee_id  joining_year month_name
0           101           2020        Jan
1           102           2021        Feb
2           103           2018        Mar
3           104           2019        Jan
4           105           2023        June
5           106           2020        Jan
6           107           2021        Feb
7           108           2018        Mar
8           109           2019        Jan
9           110           2023        June
10          111           2020        Jan
11          112           2021        Feb
12          113           2018        Mar
13          114           2019        Jan
14          115           2023        June
15          116           2020        Jan
16          117           2021        Feb
17          118           2018        Mar
18          119           2019        Jan
19          120           2023        June
20          121           2020        Jan
21          122           2021        Feb
22          123           2018        Mar
23          124           2019        Jan
24          125           2023        June

```

```

[9]: # Merge the existing 25 records with new data
emp_all_df = pd.merge(update_column_df, join_df, on='Employee_id')
emp_all_df

```

```

[9]:
   Employee_id  Employee_name  Father_Name  Age  Department  Salary  \
0           101           Asad           Islam   40           AA   11000
1           102          Qaisar           Imtiaz   34           AA   11000
2           103          Javed           Shoukat   54           AA   12000
3           104          Abid           Sadiq    34           IT   14000

```

4	105	Naeem	Sadeeq	43	IT	175000
5	106	Adil	Ameer	23	IT	135000
6	107	Atif	Hussain	32	IT	180000
7	108	Abrar	Hayat	40	IT	190000
8	109	shahid	Pera	41	MD	192000
9	110	Detha	Shoukat	42	MD	210000
10	111	Majid	Sadiq	39	MD	220000
11	112	Sajid	Mehboob	36	MD	230000
12	113	Rizwan	Mumtaz	36	MD	95000
13	114	Toqeer	Illahi	45	MD	82000
14	115	Junaid	Khair Illahi	46	MD	360000
15	116	Shakeel	Hussain	38	SD	210000
16	117	Rafaqat	Basheer	37	SD	192000
17	118	Tariq	basir	36	HR	135000
18	119	Danish	Ejaz	34	HR	120000
19	120	Sami	Shakeel	32	HR	115000
20	121	Asad	Islam	40	AA	180000
21	122	Qaisar	Imtiaz	34	AA	190000
22	123	Javed	Shoukat	54	AA	192000
23	124	Abid	Sadiq	34	IT	210000
24	125	Naeem	Sadeeq	43	IT	220000

	Education	City	Address	joining_year	month_name
0	Bachelors	Lahore	Street 10	2020	Jan
1	Masters	Karachi	Street 5	2021	Feb
2	PhD	Islamabad	Block A	2018	Mar
3	Bachelors	Multan	Sector 3	2019	Jan
4	Masters	Faisalabad	Avenue 2	2023	June
5	PhD	Lahore	Street 10	2020	Jan
6	Bachelors	Karachi	Street 5	2021	Feb
7	Masters	Islamabad	Block A	2018	Mar
8	PhD	Multan	Sector 3	2019	Jan
9	Bachelors	Faisalabad	Avenue 2	2023	June
10	Masters	Lahore	Street 10	2020	Jan
11	PhD	Karachi	Street 5	2021	Feb
12	Bachelors	Islamabad	Block A	2018	Mar
13	Masters	Multan	Sector 3	2019	Jan
14	PhD	Faisalabad	Avenue 2	2023	June
15	Bachelors	Lahore	Street 10	2020	Jan
16	Masters	Karachi	Street 5	2021	Feb
17	PhD	Islamabad	Block A	2018	Mar
18	Bachelors	Multan	Sector 3	2019	Jan
19	Masters	Faisalabad	Avenue 2	2023	June
20	Bachelors	Karachi	Block A	2020	Jan
21	Masters	Islamabad	Sector 3	2021	Feb
22	PhD	Multan	Avenue 2	2018	Mar
23	Bachelors	Faisalabad	Street 10	2019	Jan

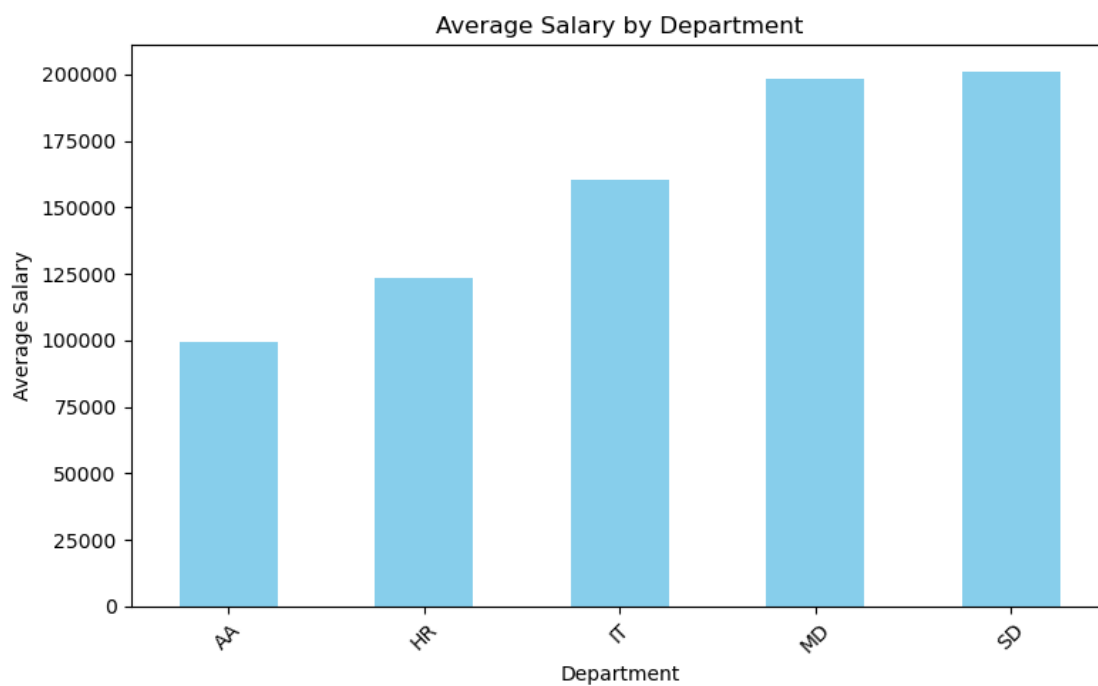
1.6 8. Finally, visualize some columns using bar chart, pie chart, line chart, histogram, and heatmap by using matplotlib or seaborn libraries.

```
[20]: import matplotlib.pyplot as plt
import seaborn as sns
```

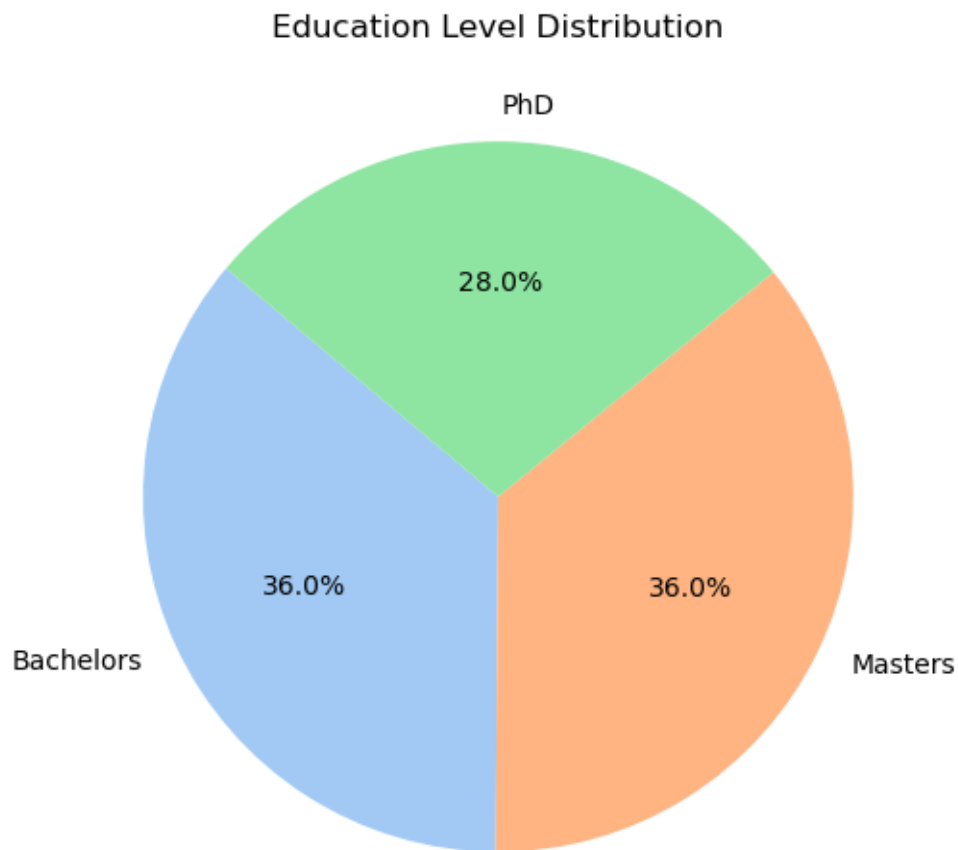
```
[12]: emp_all_df.columns
```

```
[12]: Index(['Employee_id', 'Employee_name', 'Father_Name', 'Age', 'Department',
        'Salary', 'Education', 'City', 'Address', 'joining_year', 'month_name'],
        dtype='object')
```

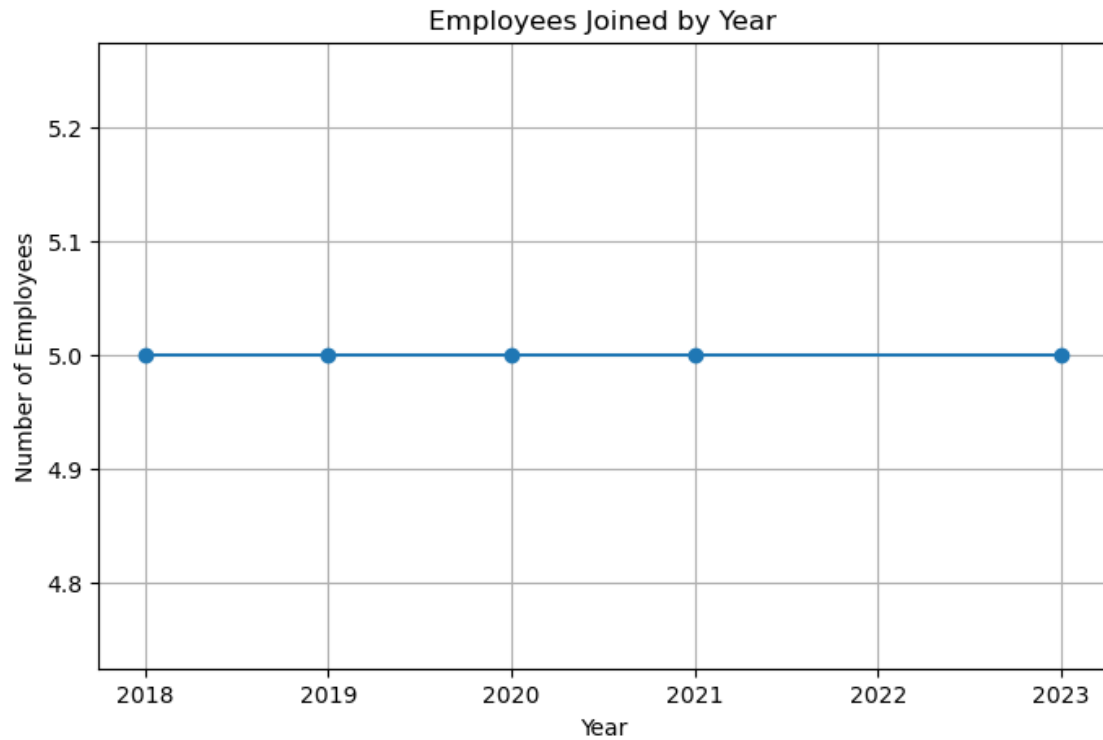
```
[18]: # Bar Chart Visualize the Average Salary by Department
plt.figure(figsize=(8, 5))
emp_all_df.groupby('Department')['Salary'].mean().plot(kind='bar',
    color='skyblue')
plt.title('Average Salary by Department')
plt.xlabel('Department')
plt.ylabel('Average Salary')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



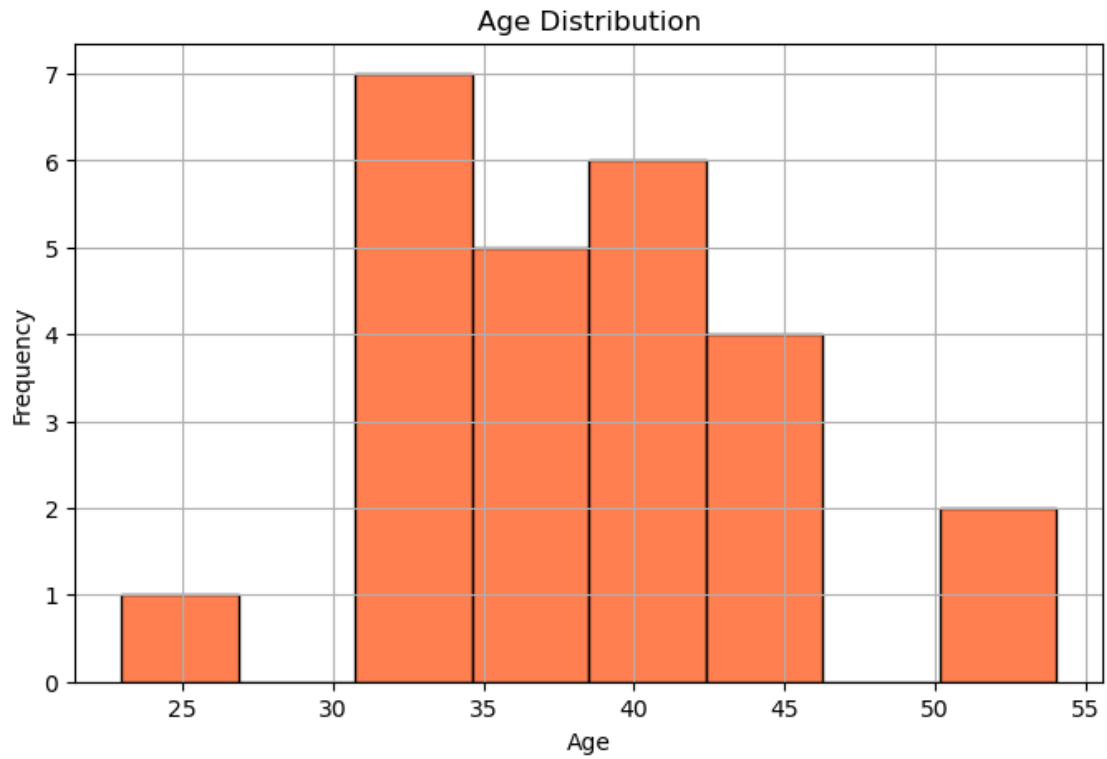
```
[23]: # Pie Chart - Visualize Distribution of Education Levels
edu_counts = emp_all_df['Education'].value_counts()
plt.figure(figsize=(6, 6))
plt.pie(edu_counts, labels=edu_counts.index, autopct='%1.1f%%', startangle=140,
        colors=sns.color_palette('pastel'))
plt.title('Education Level Distribution')
plt.show()
```



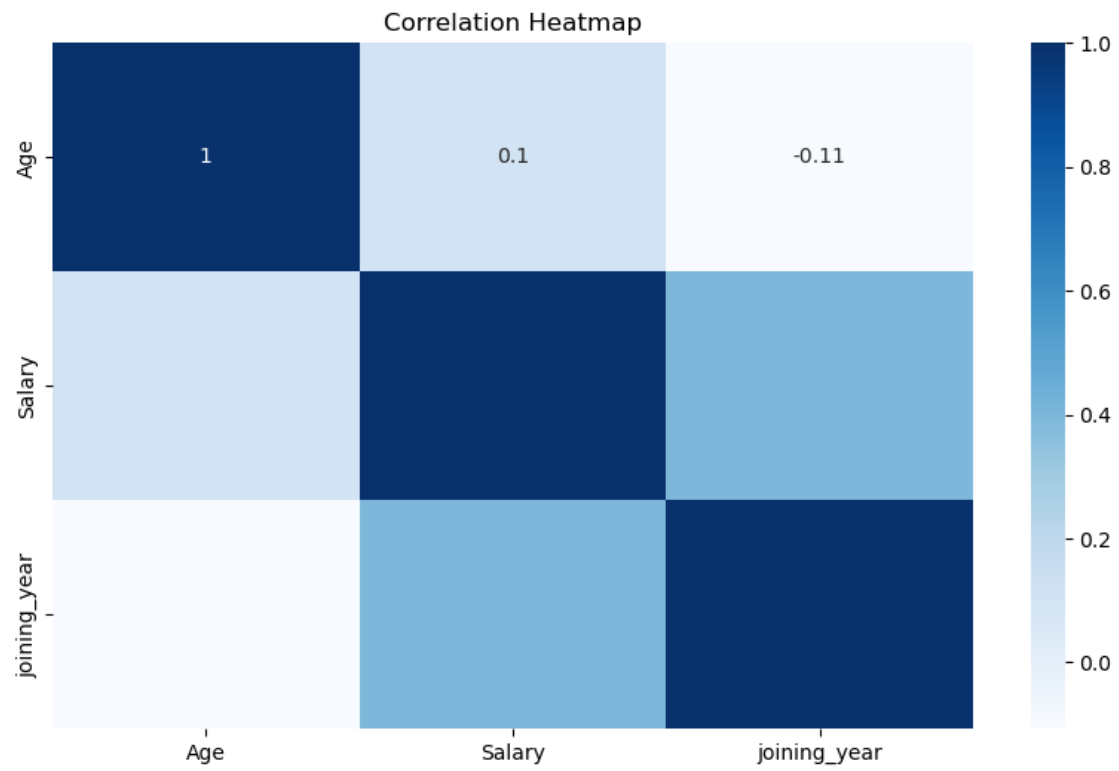
```
[26]: # Line Chart - Number of Employees Joined Each Year
plt.figure(figsize=(8, 5))
emp_all_df['joining_year'].value_counts().sort_index().plot(kind='line',
        marker='o')
plt.title('Employees Joined by Year')
plt.xlabel('Year')
plt.ylabel('Number of Employees')
plt.grid(True)
plt.show()
```



```
[28]: # Histogram - Age Distribution of Employees
plt.figure(figsize=(8, 5))
plt.hist(emp_all_df['Age'], bins=8, color='coral', edgecolor='black')
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```



```
[29]: # Heatmap - Correlation Matrix
plt.figure(figsize=(10, 6))
sns.heatmap(emp_all_df[['Age', 'Salary', 'joining_year']].corr(), annot=True,
            cmap='Blues')
plt.title('Correlation Heatmap')
plt.show()
```



[ ]: