

Merge, Concatenate, join and more

```
In [1]: import pandas as pd # For data manipulation and analysis
import numpy as np # For numerical computations
import matplotlib.pyplot as plt # For creating static, animated, and interactive plots
import seaborn as sns # For statistical data visualization based on Matplotlib
import scipy # For scientific and technical computing (including optimization,
```

```
In [2]: # Creating realistic data for employees
data = {
    'Employee ID': np.arange(1001, 1011),
    'Employee Name': ['Satender Kumar', 'data 1', 'Jane Smith', 'Robert Brown',
    'Department': ['Data Analyst', 'IT', 'Finance', 'Marketing', 'Sales', 'Operations',
    'Age': [24, np.random.randint(25, 60), np.random.randint(25, 60), np.random.
    'Location': ['London, Canada', 'Toronto', 'London', 'Sydney', 'San Francisco',
    'Salary': np.random.randint(50000, 150000, size=10),
    'Years with Company': np.random.randint(1, 15, size=10),
    'Position': ['Data Analyst', 'Developer', 'Analyst', 'Designer', 'Consultant',
    'Performance Score': np.random.randint(1, 5, size=10),
    'Bonus': np.random.randint(1000, 10000, size=10),
    'Gender': ['Male', 'Male', 'Female', 'Male', 'Female', 'Male', 'Female', 'Male',
    'Marital Status': ['Single', 'Single', 'Married', 'Single', 'Single', 'Married',
    'Education': ['Bachelor', 'Master', 'PhD', 'Bachelor', 'Master', 'PhD', 'Bachelor',
    'Hire Date': pd.to_datetime(['2019-06-12', '2015-07-23', '2012-09-05', '2018-01-01',
    'Overtime Hours': np.random.randint(0, 20, size=10),
    'Sick Days Taken': np.random.randint(0, 10, size=10),
    'Vacation Days Taken': np.random.randint(5, 20, size=10),
    'Training Hours': np.random.randint(10, 50, size=10),
    'Certifications': ['Yes', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'Yes', 'No',
    'Supervisor': ['Anna Smith', 'Brian Adams', 'Clara Jones', 'Daniel Martin',
    }

# Creating the DataFrame
df = pd.DataFrame(data)
```

```
In [4]: # Display the DataFrame
print(df)
```

	Employee ID	Employee Name	Department	Age	Location	Salary	\
0	1001	Satender Kumar	Data Analyst	24	London, Canada	62989	
1	1002	data 1	IT	46	Toronto	96061	
2	1003	Jane Smith	Finance	48	London	60507	
3	1004	Robert Brown	Marketing	32	Sydney	71727	
4	1005	Emily Davis	Sales	40	San Francisco	81098	
5	1006	Michael Wilson	Operations	45	Paris	60924	
6	1007	Sarah Taylor	R&D	58	Berlin	116105	
7	1008	David Lee	Support	28	Tokyo	138939	
8	1009	Laura Johnson	Admin	38	Dubai	93249	
9	1010	James White	Legal	46	Singapore	141721	

	Years with Company	Position	Performance Score	Bonus	Gender	\
0	10	Data Analyst	4	6419	Male	
1	5	Developer	4	4746	Male	
2	1	Analyst	1	1447	Female	
3	9	Designer	3	7196	Male	
4	13	Consultant	3	2218	Female	
5	6	Engineer	2	7973	Male	
6	13	Scientist	4	7565	Female	
7	9	Support Agent	3	1800	Male	
8	7	Admin Assistant	4	1189	Female	
9	2	Lawyer	2	1654	Male	

	Marital Status	Education	Hire Date	Overtime Hours	Sick Days Taken	\
0	Single	Bachelor	2019-06-12	10	2	
1	Single	Master	2015-07-23	15	7	
2	Married	PhD	2012-09-05	0	5	
3	Single	Bachelor	2018-11-30	18	3	
4	Single	Master	2013-05-19	1	7	
5	Married	PhD	2019-02-14	0	1	
6	Married	Bachelor	2020-08-21	3	0	
7	Single	Master	2016-06-03	6	0	
8	Married	Bachelor	2014-01-28	15	1	
9	Single	Master	2017-03-15	3	7	

	Vacation Days Taken	Training Hours	Certifications	Supervisor
0	19	15	Yes	Anna Smith
1	14	31	No	Brian Adams
2	7	37	Yes	Clara Jones
3	11	12	No	Daniel Martin
4	17	30	Yes	Eva Rodriguez
5	16	24	No	Frank Bell
6	7	49	Yes	Grace Moore
7	6	10	Yes	Hannah Lewis
8	11	38	No	Ivan Scott
9	19	40	Yes	Jake Miller

```
In [6]: df.head()
```

Out[6]:

	Employee ID	Employee Name	Department	Age	Location	Salary	Years with Company	Position	Per
0	1001	Satender Kumar	Data Analyst	24	London, Canada	62989	10	Data Analyst	
1	1002	data 1	IT	46	Toronto	96061	5	Developer	
2	1003	Jane Smith	Finance	48	London	60507	1	Analyst	
3	1004	Robert Brown	Marketing	32	Sydney	71727	9	Designer	
4	1005	Emily Davis	Sales	40	San Francisco	81098	13	Consultant	

In [10]: df.describe()

Out[10]:

	Employee ID	Age	Salary	Years with Company	Performance Score	Bonus	I
count	10.00000	10.000000	10.000000	10.000000	10.000000	10.000000	
mean	1005.50000	40.500000	92332.000000	7.500000	3.000000	4220.700000	209:3
min	1001.00000	24.000000	60507.000000	1.000000	1.000000	1189.000000	200:C
25%	1003.25000	33.500000	65173.500000	5.250000	2.250000	1690.500000	206:C
50%	1005.50000	42.500000	87173.500000	8.000000	3.000000	3482.000000	2112:C
75%	1007.75000	46.000000	111094.000000	9.750000	4.000000	7001.750000	200:C
max	1010.00000	58.000000	141721.000000	13.000000	4.000000	7973.000000	200:C
std	3.02765	10.276727	30916.484542	4.116363	1.054093	2839.564366	

In [12]: df.isnull()

Out[12]:

	Employee ID	Employee Name	Department	Age	Location	Salary	Years with Company	Position	Perfomance
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
5	False	False	False	False	False	False	False	False	False
6	False	False	False	False	False	False	False	False	False
7	False	False	False	False	False	False	False	False	False
8	False	False	False	False	False	False	False	False	False
9	False	False	False	False	False	False	False	False	False

In [14]: df.notnull()

Out[14]:

	Employee ID	Employee Name	Department	Age	Location	Salary	Years with Company	Position	Perfomance
0	True	True	True	True	True	True	True	True	True
1	True	True	True	True	True	True	True	True	True
2	True	True	True	True	True	True	True	True	True
3	True	True	True	True	True	True	True	True	True
4	True	True	True	True	True	True	True	True	True
5	True	True	True	True	True	True	True	True	True
6	True	True	True	True	True	True	True	True	True
7	True	True	True	True	True	True	True	True	True
8	True	True	True	True	True	True	True	True	True
9	True	True	True	True	True	True	True	True	True

In [16]: df.count()

Out[16]: Employee ID 10
Employee Name 10
Department 10
Age 10
Location 10
Salary 10
Years with Company 10
Position 10
Performance Score 10
Bonus 10
Gender 10
Marital Status 10
Education 10
Hire Date 10
Overtime Hours 10
Sick Days Taken 10
Vacation Days Taken 10
Training Hours 10
Certifications 10
Supervisor 10
dtype: int64

In [19]: df.head(20
)

Out[19]:

	Employee ID	Employee Name	Department	Age	Location	Salary	Years with Company	Position	P
0	1001	Satender Kumar	Data Analyst	24	London, Canada	62989	10	Data Analyst	
1	1002	data 1	IT	46	Toronto	96061	5	Developer	
2	1003	Jane Smith	Finance	48	London	60507	1	Analyst	
3	1004	Robert Brown	Marketing	32	Sydney	71727	9	Designer	
4	1005	Emily Davis	Sales	40	San Francisco	81098	13	Consultant	
5	1006	Michael Wilson	Operations	45	Paris	60924	6	Engineer	
6	1007	Sarah Taylor	R&D	58	Berlin	116105	13	Scientist	
7	1008	David Lee	Support	28	Tokyo	138939	9	Support Agent	
8	1009	Laura Johnson	Admin	38	Dubai	93249	7	Admin Assistant	
9	1010	James White	Legal	46	Singapore	141721	2	Lawyer	

```

In [21]: # Creating realistic data for a second set of employees
data1 = {
    'Employee ID': np.arange(1011, 1021),
    'Employee Name': ['Satender Kumar', 'data 1', 'Chris Evans', 'Natalie Portma',
    'Department': ['Data Analyst', 'HR', 'IT', 'Marketing', 'Finance', 'Sales',
    'Age': [24, np.random.randint(25, 60), np.random.randint(25, 60), np.random.
    'Location': ['London, Canada', 'Los Angeles', 'New York', 'Chicago', 'Housto
    'Salary': np.random.randint(60000, 160000, size=10),
    'Years with Company': np.random.randint(1, 20, size=10),
    'Position': ['Data Analyst', 'HR Manager', 'IT Specialist', 'Marketing Coord
    'Performance Score': np.random.randint(1, 5, size=10),
    'Bonus': np.random.randint(2000, 12000, size=10),
    'Gender': ['Male', 'Male', 'Female', 'Female', 'Male', 'Female', 'Male', 'Fe
    'Marital Status': ['Single', 'Married', 'Single', 'Single', 'Married', 'Sing
    'Education': ['Master', 'Bachelor', 'Master', 'PhD', 'Bachelor', 'Master', '
    'Hire Date': pd.to_datetime(['2018-07-15', '2014-03-22', '2011-10-12', '2017
    'Overtime Hours': np.random.randint(0, 25, size=10),
    'Sick Days Taken': np.random.randint(0, 8, size=10),
    'Vacation Days Taken': np.random.randint(7, 22, size=10),
    'Training Hours': np.random.randint(15, 55, size=10),
    'Certifications': ['Yes', 'Yes', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'Yes
    'Supervisor': ['John Smith', 'Michael Johnson', 'Patricia Williams', 'Linda
}

# Creating the second DataFrame
df1 = pd.DataFrame(data1)

```

```

In [23]: # Display the DataFrame
print(df1)

```

	Employee ID	Employee Name	Department	Age	Location	Salary \
0	1011	Satender Kumar	Data Analyst	24	London, Canada	75719
1	1012	data 1	HR	40	Los Angeles	96413
2	1013	Chris Evans	IT	37	New York	74042
3	1014	Natalie Portman	Marketing	49	Chicago	108467
4	1015	Tom Holland	Finance	27	Houston	65822
5	1016	Emma Watson	Sales	56	Phoenix	114854
6	1017	Daniel Radcliffe	R&D	33	Philadelphia	117933
7	1018	Scarlett Johansson	Operations	41	San Antonio	87342
8	1019	Robert Downey Jr.	Legal	51	San Diego	72775
9	1020	Mark Ruffalo	Support	42	Dallas	142745

	Years with Company	Position	Performance Score	Bonus \
0	12	Data Analyst	2	3187
1	12	HR Manager	1	10993
2	14	IT Specialist	4	10834
3	11	Marketing Coordinator	4	2760
4	12	Financial Analyst	2	6263
5	5	Sales Manager	3	10287
6	17	Research Scientist	3	8290
7	8	Operations Manager	3	10047
8	6	Legal Advisor	2	6510
9	10	Support Specialist	1	5533

	Gender	Marital Status	Education	Hire Date	Overtime Hours \
0	Male	Single	Master	2018-07-15	17
1	Male	Married	Bachelor	2014-03-22	8
2	Female	Single	Master	2011-10-12	12
3	Female	Single	PhD	2017-04-17	10
4	Male	Married	Bachelor	2015-09-23	13
5	Female	Single	Master	2016-11-01	15
6	Male	Single	PhD	2019-05-11	8
7	Female	Married	Bachelor	2020-07-08	15
8	Male	Single	Master	2013-08-19	7
9	Male	Married	PhD	2012-01-09	2

	Sick Days Taken	Vacation Days Taken	Training Hours	Certifications \
0	5	12	21	Yes
1	3	18	41	Yes
2	4	8	22	No
3	7	20	42	Yes
4	6	11	35	No
5	0	20	35	Yes
6	1	15	48	No
7	7	9	50	Yes
8	7	20	35	Yes
9	4	17	15	No

	Supervisor
0	John Smith
1	Michael Johnson
2	Patricia Williams
3	Linda Brown
4	Barbara Jones
5	Elizabeth Garcia
6	Susan Martinez
7	Jessica Hernandez
8	Sarah Lopez
9	Karen Wilson

```
In [25]: df1.head()
```

Out[25]:

	Employee ID	Employee Name	Department	Age	Location	Salary	Years with Company	Position	P
0	1011	Satender Kumar	Data Analyst	24	London, Canada	75719	12	Data Analyst	
1	1012	data 1	HR	40	Los Angeles	96413	12	HR Manager	
2	1013	Chris Evans	IT	37	New York	74042	14	IT Specialist	
3	1014	Natalie Portman	Marketing	49	Chicago	108467	11	Marketing Coordinator	
4	1015	Tom Holland	Finance	27	Houston	65822	12	Financial Analyst	

```
In [27]: df.describe()
```

Out[27]:

	Employee ID	Age	Salary	Years with Company	Performance Score	Bonus	I
count	10.00000	10.000000	10.000000	10.000000	10.000000	10.000000	
mean	1005.50000	40.500000	92332.000000	7.500000	3.000000	4220.700000	209:30
min	1001.00000	24.000000	60507.000000	1.000000	1.000000	1189.000000	200:00
25%	1003.25000	33.500000	65173.500000	5.250000	2.250000	1690.500000	206:00
50%	1005.50000	42.500000	87173.500000	8.000000	3.000000	3482.000000	2112:00
75%	1007.75000	46.000000	111094.000000	9.750000	4.000000	7001.750000	200:00
max	1010.00000	58.000000	141721.000000	13.000000	4.000000	7973.000000	200:00
std	3.02765	10.276727	30916.484542	4.116363	1.054093	2839.564366	

```
In [28]: df1.notnull()
```


Out[28]:

	Employee ID	Employee Name	Department	Age	Location	Salary	Years with Company	Position	Perfo
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	

In [29]: df1.isnull()

Out[29]:

	Employee ID	Employee Name	Department	Age	Location	Salary	Years with Company	Position	Perfo
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	

In [30]: df1.count()

```
Out[30]: Employee ID      10
Employee Name      10
Department         10
Age                10
Location           10
Salary             10
Years with Company 10
Position           10
Performance Score  10
Bonus              10
Gender             10
Marital Status     10
Education          10
Hire Date          10
Overtime Hours     10
Sick Days Taken    10
Vacation Days Taken 10
Training Hours     10
Certifications     10
Supervisor         10
dtype: int64
```

```
In [ ]: df1.sum()
```

```
In [33]: print(df1)
```

	Employee ID	Employee Name	Department	Age	Location	Salary \
0	1011	Satender Kumar	Data Analyst	24	London, Canada	75719
1	1012	data 1	HR	40	Los Angeles	96413
2	1013	Chris Evans	IT	37	New York	74042
3	1014	Natalie Portman	Marketing	49	Chicago	108467
4	1015	Tom Holland	Finance	27	Houston	65822
5	1016	Emma Watson	Sales	56	Phoenix	114854
6	1017	Daniel Radcliffe	R&D	33	Philadelphia	117933
7	1018	Scarlett Johansson	Operations	41	San Antonio	87342
8	1019	Robert Downey Jr.	Legal	51	San Diego	72775
9	1020	Mark Ruffalo	Support	42	Dallas	142745

	Years with Company	Position	Performance Score	Bonus \
0	12	Data Analyst	2	3187
1	12	HR Manager	1	10993
2	14	IT Specialist	4	10834
3	11	Marketing Coordinator	4	2760
4	12	Financial Analyst	2	6263
5	5	Sales Manager	3	10287
6	17	Research Scientist	3	8290
7	8	Operations Manager	3	10047
8	6	Legal Advisor	2	6510
9	10	Support Specialist	1	5533

	Gender	Marital Status	Education	Hire Date	Overtime Hours \
0	Male	Single	Master	2018-07-15	17
1	Male	Married	Bachelor	2014-03-22	8
2	Female	Single	Master	2011-10-12	12
3	Female	Single	PhD	2017-04-17	10
4	Male	Married	Bachelor	2015-09-23	13
5	Female	Single	Master	2016-11-01	15
6	Male	Single	PhD	2019-05-11	8
7	Female	Married	Bachelor	2020-07-08	15
8	Male	Single	Master	2013-08-19	7
9	Male	Married	PhD	2012-01-09	2

	Sick Days Taken	Vacation Days Taken	Training Hours	Certifications \
0	5	12	21	Yes
1	3	18	41	Yes
2	4	8	22	No
3	7	20	42	Yes
4	6	11	35	No
5	0	20	35	Yes
6	1	15	48	No
7	7	9	50	Yes
8	7	20	35	Yes
9	4	17	15	No

	Supervisor
0	John Smith
1	Michael Johnson
2	Patricia Williams
3	Linda Brown
4	Barbara Jones
5	Elizabeth Garcia
6	Susan Martinez
7	Jessica Hernandez
8	Sarah Lopez
9	Karen Wilson

In [36]: `import pandas as pd`

```
# Assuming df and df1 are your DataFrames from the previous examples

# Merging the two DataFrames on the 'Employee ID' column
merged_df = pd.merge(df, df1, on='Employee ID', suffixes=('_df', '_df1'))

# Display the merged DataFrame
print(merged_df)
```

Empty DataFrame

Columns: [Employee ID, Employee Name_df, Department_df, Age_df, Location_df, Salary_df, Years with Company_df, Position_df, Performance Score_df, Bonus_df, Gender_df, Marital Status_df, Education_df, Hire Date_df, Overtime Hours_df, Sick Days Taken_df, Vacation Days Taken_df, Training Hours_df, Certifications_df, Supervisor_df, Employee Name_df1, Department_df1, Age_df1, Location_df1, Salary_df1, Years with Company_df1, Position_df1, Performance Score_df1, Bonus_df1, Gender_df1, Marital Status_df1, Education_df1, Hire Date_df1, Overtime Hours_df1, Sick Days Taken_df1, Vacation Days Taken_df1, Training Hours_df1, Certifications_df1, Supervisor_df1]

Index: []

[0 rows x 39 columns]

In [39]: `merged_df = pd.merge(df, df1, on='Employee ID', suffixes=('_df', '_df1'), how='c'`
`print(merged_df)`

	Employee ID	Employee Name_df	Department_df	Age_df	Location_df	\
0	1001	Satender Kumar	Data Analyst	24.0	London, Canada	
1	1002	data 1	IT	46.0	Toronto	
2	1003	Jane Smith	Finance	48.0	London	
3	1004	Robert Brown	Marketing	32.0	Sydney	
4	1005	Emily Davis	Sales	40.0	San Francisco	
5	1006	Michael Wilson	Operations	45.0	Paris	
6	1007	Sarah Taylor	R&D	58.0	Berlin	
7	1008	David Lee	Support	28.0	Tokyo	
8	1009	Laura Johnson	Admin	38.0	Dubai	
9	1010	James White	Legal	46.0	Singapore	
10	1011	NaN	NaN	NaN	NaN	
11	1012	NaN	NaN	NaN	NaN	
12	1013	NaN	NaN	NaN	NaN	
13	1014	NaN	NaN	NaN	NaN	
14	1015	NaN	NaN	NaN	NaN	
15	1016	NaN	NaN	NaN	NaN	
16	1017	NaN	NaN	NaN	NaN	
17	1018	NaN	NaN	NaN	NaN	
18	1019	NaN	NaN	NaN	NaN	
19	1020	NaN	NaN	NaN	NaN	

	Salary_df	Years with Company_df	Position_df	Performance Score_df	\
0	62989.0	10.0	Data Analyst	4.0	
1	96061.0	5.0	Developer	4.0	
2	60507.0	1.0	Analyst	1.0	
3	71727.0	9.0	Designer	3.0	
4	81098.0	13.0	Consultant	3.0	
5	60924.0	6.0	Engineer	2.0	
6	116105.0	13.0	Scientist	4.0	
7	138939.0	9.0	Support Agent	3.0	
8	93249.0	7.0	Admin Assistant	4.0	
9	141721.0	2.0	Lawyer	2.0	
10	NaN	NaN	NaN	NaN	
11	NaN	NaN	NaN	NaN	
12	NaN	NaN	NaN	NaN	
13	NaN	NaN	NaN	NaN	
14	NaN	NaN	NaN	NaN	
15	NaN	NaN	NaN	NaN	
16	NaN	NaN	NaN	NaN	
17	NaN	NaN	NaN	NaN	
18	NaN	NaN	NaN	NaN	
19	NaN	NaN	NaN	NaN	

	Bonus_df	...	Gender_df1	Marital Status_df1	Education_df1	Hire Date_df1	\
0	6419.0	...	NaN	NaN	NaN	NaT	
1	4746.0	...	NaN	NaN	NaN	NaT	
2	1447.0	...	NaN	NaN	NaN	NaT	
3	7196.0	...	NaN	NaN	NaN	NaT	
4	2218.0	...	NaN	NaN	NaN	NaT	
5	7973.0	...	NaN	NaN	NaN	NaT	
6	7565.0	...	NaN	NaN	NaN	NaT	
7	1800.0	...	NaN	NaN	NaN	NaT	
8	1189.0	...	NaN	NaN	NaN	NaT	
9	1654.0	...	NaN	NaN	NaN	NaT	
10	NaN	...	Male	Single	Master	2018-07-15	
11	NaN	...	Male	Married	Bachelor	2014-03-22	
12	NaN	...	Female	Single	Master	2011-10-12	
13	NaN	...	Female	Single	PhD	2017-04-17	
14	NaN	...	Male	Married	Bachelor	2015-09-23	

15	NaN	...	Female	Single	Master	2016-11-01
16	NaN	...	Male	Single	PhD	2019-05-11
17	NaN	...	Female	Married	Bachelor	2020-07-08
18	NaN	...	Male	Single	Master	2013-08-19
19	NaN	...	Male	Married	PhD	2012-01-09

	Overtime Hours_df1	Sick Days Taken_df1	Vacation Days Taken_df1	\
0	NaN	NaN	NaN	
1	NaN	NaN	NaN	
2	NaN	NaN	NaN	
3	NaN	NaN	NaN	
4	NaN	NaN	NaN	
5	NaN	NaN	NaN	
6	NaN	NaN	NaN	
7	NaN	NaN	NaN	
8	NaN	NaN	NaN	
9	NaN	NaN	NaN	
10	17.0	5.0	12.0	
11	8.0	3.0	18.0	
12	12.0	4.0	8.0	
13	10.0	7.0	20.0	
14	13.0	6.0	11.0	
15	15.0	0.0	20.0	
16	8.0	1.0	15.0	
17	15.0	7.0	9.0	
18	7.0	7.0	20.0	
19	2.0	4.0	17.0	

	Training Hours_df1	Certifications_df1	Supervisor_df1
0	NaN	NaN	NaN
1	NaN	NaN	NaN
2	NaN	NaN	NaN
3	NaN	NaN	NaN
4	NaN	NaN	NaN
5	NaN	NaN	NaN
6	NaN	NaN	NaN
7	NaN	NaN	NaN
8	NaN	NaN	NaN
9	NaN	NaN	NaN
10	21.0	Yes	John Smith
11	41.0	Yes	Michael Johnson
12	22.0	No	Patricia Williams
13	42.0	Yes	Linda Brown
14	35.0	No	Barbara Jones
15	35.0	Yes	Elizabeth Garcia
16	48.0	No	Susan Martinez
17	50.0	Yes	Jessica Hernandez
18	35.0	Yes	Sarah Lopez
19	15.0	No	Karen Wilson

[20 rows x 39 columns]

```
In [43]: print(df['Employee ID'])
print(df1['Employee ID'])
```

```

0    1001
1    1002
2    1003
3    1004
4    1005
5    1006
6    1007
7    1008
8    1009
9    1010
Name: Employee ID, dtype: int32
0    1011
1    1012
2    1013
3    1014
4    1015
5    1016
6    1017
7    1018
8    1019
9    1020
Name: Employee ID, dtype: int32

```

```
In [45]: df
```

Out[45]:

	Employee ID	Employee Name	Department	Age	Location	Salary	Years with Company	Position	P
0	1001	Satender Kumar	Data Analyst	24	London, Canada	62989	10	Data Analyst	
1	1002	data 1	IT	46	Toronto	96061	5	Developer	
2	1003	Jane Smith	Finance	48	London	60507	1	Analyst	
3	1004	Robert Brown	Marketing	32	Sydney	71727	9	Designer	
4	1005	Emily Davis	Sales	40	San Francisco	81098	13	Consultant	
5	1006	Michael Wilson	Operations	45	Paris	60924	6	Engineer	
6	1007	Sarah Taylor	R&D	58	Berlin	116105	13	Scientist	
7	1008	David Lee	Support	28	Tokyo	138939	9	Support Agent	
8	1009	Laura Johnson	Admin	38	Dubai	93249	7	Admin Assistant	
9	1010	James White	Legal	46	Singapore	141721	2	Lawyer	

```
In [46]: df1
```

Out[46]:

	Employee ID	Employee Name	Department	Age	Location	Salary	Years with Company	Position
0	1011	Satender Kumar	Data Analyst	24	London, Canada	75719	12	Data Analyst
1	1012	data 1	HR	40	Los Angeles	96413	12	HR Manager
2	1013	Chris Evans	IT	37	New York	74042	14	IT Specialist
3	1014	Natalie Portman	Marketing	49	Chicago	108467	11	Marketing Coordinator
4	1015	Tom Holland	Finance	27	Houston	65822	12	Financial Analyst
5	1016	Emma Watson	Sales	56	Phoenix	114854	5	Sales Manager
6	1017	Daniel Radcliffe	R&D	33	Philadelphia	117933	17	Research Scientist
7	1018	Scarlett Johansson	Operations	41	San Antonio	87342	8	Operations Manager
8	1019	Robert Downey Jr.	Legal	51	San Diego	72775	6	Legal Advisor
9	1020	Mark Ruffalo	Support	42	Dallas	142745	10	Support Specialist

In [50]: `print(pd.concat([df,df1]))`

	Employee ID	Employee Name	Department	Age	Location	Salary \
0	1001	Satender Kumar	Data Analyst	24	London, Canada	62989
1	1002	data 1	IT	46	Toronto	96061
2	1003	Jane Smith	Finance	48	London	60507
3	1004	Robert Brown	Marketing	32	Sydney	71727
4	1005	Emily Davis	Sales	40	San Francisco	81098
5	1006	Michael Wilson	Operations	45	Paris	60924
6	1007	Sarah Taylor	R&D	58	Berlin	116105
7	1008	David Lee	Support	28	Tokyo	138939
8	1009	Laura Johnson	Admin	38	Dubai	93249
9	1010	James White	Legal	46	Singapore	141721
0	1011	Satender Kumar	Data Analyst	24	London, Canada	75719
1	1012	data 1	HR	40	Los Angeles	96413
2	1013	Chris Evans	IT	37	New York	74042
3	1014	Natalie Portman	Marketing	49	Chicago	108467
4	1015	Tom Holland	Finance	27	Houston	65822
5	1016	Emma Watson	Sales	56	Phoenix	114854
6	1017	Daniel Radcliffe	R&D	33	Philadelphia	117933
7	1018	Scarlett Johansson	Operations	41	San Antonio	87342
8	1019	Robert Downey Jr.	Legal	51	San Diego	72775
9	1020	Mark Ruffalo	Support	42	Dallas	142745

	Years with Company	Position	Performance Score	Bonus \
0	10	Data Analyst	4	6419
1	5	Developer	4	4746
2	1	Analyst	1	1447
3	9	Designer	3	7196
4	13	Consultant	3	2218
5	6	Engineer	2	7973
6	13	Scientist	4	7565
7	9	Support Agent	3	1800
8	7	Admin Assistant	4	1189
9	2	Lawyer	2	1654
0	12	Data Analyst	2	3187
1	12	HR Manager	1	10993
2	14	IT Specialist	4	10834
3	11	Marketing Coordinator	4	2760
4	12	Financial Analyst	2	6263
5	5	Sales Manager	3	10287
6	17	Research Scientist	3	8290
7	8	Operations Manager	3	10047
8	6	Legal Advisor	2	6510
9	10	Support Specialist	1	5533

	Gender	Marital Status	Education	Hire Date	Overtime Hours \
0	Male	Single	Bachelor	2019-06-12	10
1	Male	Single	Master	2015-07-23	15
2	Female	Married	PhD	2012-09-05	0
3	Male	Single	Bachelor	2018-11-30	18
4	Female	Single	Master	2013-05-19	1
5	Male	Married	PhD	2019-02-14	0
6	Female	Married	Bachelor	2020-08-21	3
7	Male	Single	Master	2016-06-03	6
8	Female	Married	Bachelor	2014-01-28	15
9	Male	Single	Master	2017-03-15	3
0	Male	Single	Master	2018-07-15	17
1	Male	Married	Bachelor	2014-03-22	8
2	Female	Single	Master	2011-10-12	12
3	Female	Single	PhD	2017-04-17	10
4	Male	Married	Bachelor	2015-09-23	13

5	Female	Single	Master	2016-11-01	15
6	Male	Single	PhD	2019-05-11	8
7	Female	Married	Bachelor	2020-07-08	15
8	Male	Single	Master	2013-08-19	7
9	Male	Married	PhD	2012-01-09	2

	Sick Days Taken	Vacation Days Taken	Training Hours	Certifications	\
0	2	19	15	Yes	
1	7	14	31	No	
2	5	7	37	Yes	
3	3	11	12	No	
4	7	17	30	Yes	
5	1	16	24	No	
6	0	7	49	Yes	
7	0	6	10	Yes	
8	1	11	38	No	
9	7	19	40	Yes	
0	5	12	21	Yes	
1	3	18	41	Yes	
2	4	8	22	No	
3	7	20	42	Yes	
4	6	11	35	No	
5	0	20	35	Yes	
6	1	15	48	No	
7	7	9	50	Yes	
8	7	20	35	Yes	
9	4	17	15	No	

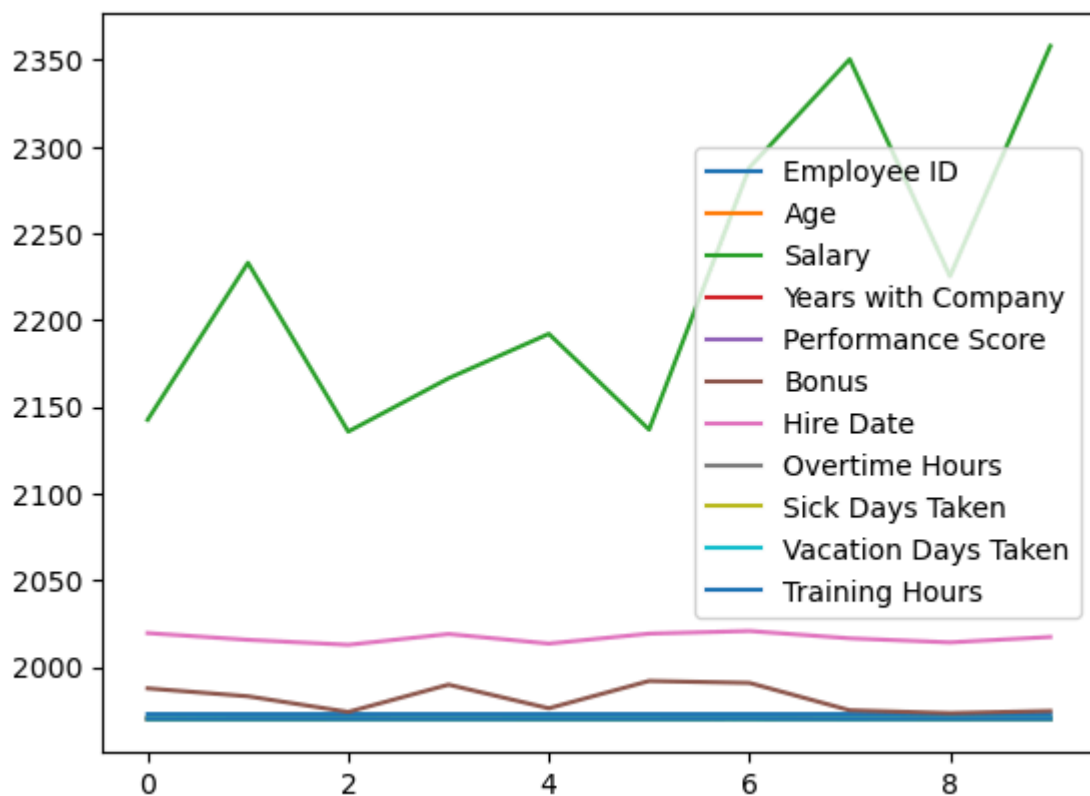
	Supervisor
0	Anna Smith
1	Brian Adams
2	Clara Jones
3	Daniel Martin
4	Eva Rodriguez
5	Frank Bell
6	Grace Moore
7	Hannah Lewis
8	Ivan Scott
9	Jake Miller
0	John Smith
1	Michael Johnson
2	Patricia Williams
3	Linda Brown
4	Barbara Jones
5	Elizabeth Garcia
6	Susan Martinez
7	Jessica Hernandez
8	Sarah Lopez
9	Karen Wilson

In [52]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 20 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Employee ID                          10 non-null     int32
1   Employee Name                        10 non-null     object
2   Department                           10 non-null     object
3   Age                                  10 non-null     int64
4   Location                             10 non-null     object
5   Salary                               10 non-null     int32
6   Years with Company                   10 non-null     int32
7   Position                             10 non-null     object
8   Performance Score                    10 non-null     int32
9   Bonus                               10 non-null     int32
10  Gender                               10 non-null     object
11  Marital Status                       10 non-null     object
12  Education                            10 non-null     object
13  Hire Date                            10 non-null     datetime64[ns]
14  Overtime Hours                       10 non-null     int32
15  Sick Days Taken                      10 non-null     int32
16  Vacation Days Taken                  10 non-null     int32
17  Training Hours                       10 non-null     int32
18  Certifications                       10 non-null     object
19  Supervisor                           10 non-null     object
dtypes: datetime64[ns](1), int32(9), int64(1), object(9)
memory usage: 1.3+ KB
```

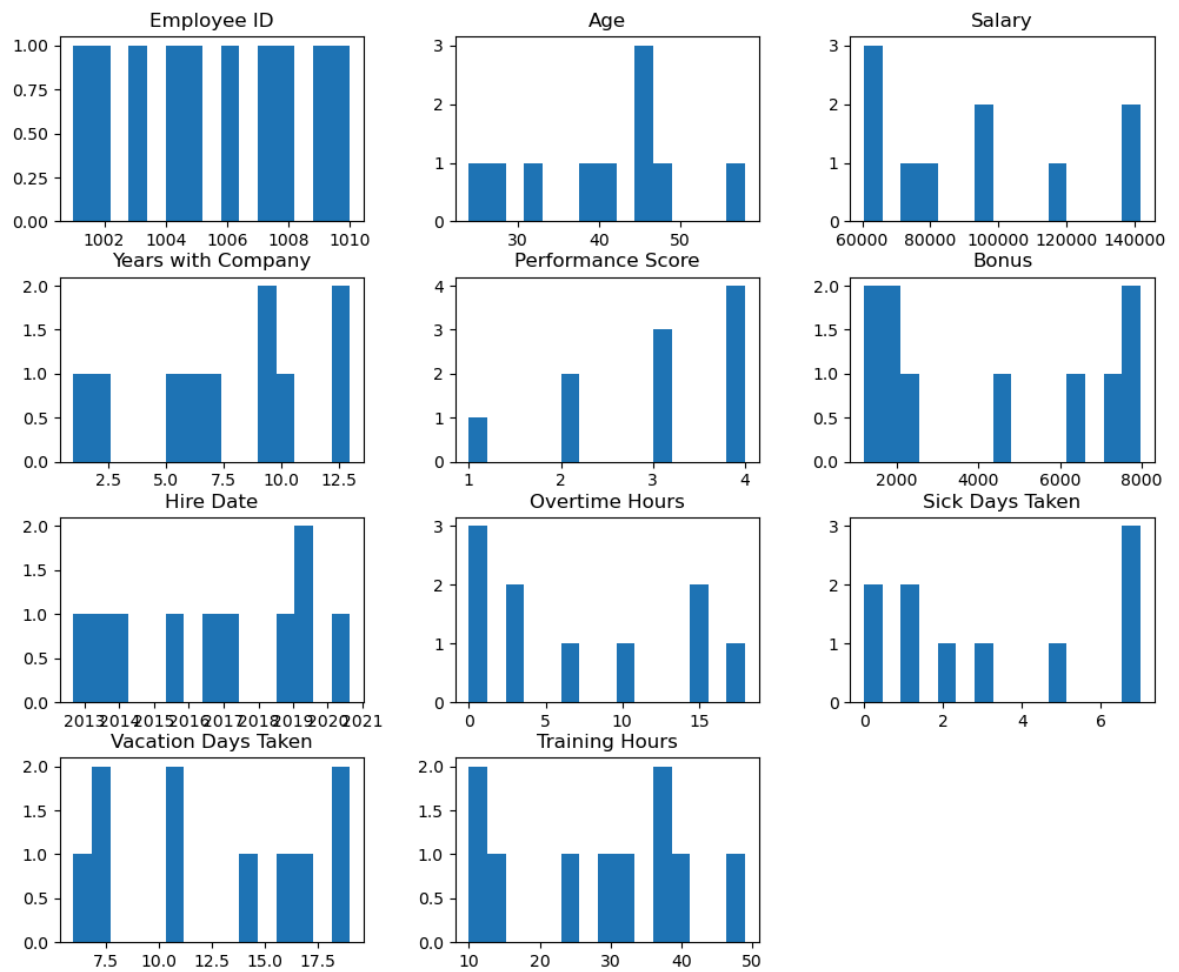
```
In [55]: df.plot()
```

```
Out[55]: <Axes: >
```



```
In [57]: df.hist(figsize=(12, 10), bins=15, grid=False)
```

```
# Display the plots
plt.show()
```



In []: