```
# ================================================================
#  Pandas Practical – Data Analysis & Visualization (NEW DATA)
#  ----------------------------------------------------------------
#  Run the whole notebook in Jupyter / JupyterLab / Google Colab
# ================================================================

# ----------------------------------------------------------------
# 1. Installation & Import
# ----------------------------------------------------------------
# (uncomment if you need to install)
# !pip install pandas openpyxl matplotlib

import pandas as pd
import numpy as np
from datetime import datetime, timedelta
import matplotlib.pyplot as plt
%matplotlib inline

# ----------------------------------------------------------------
# 2. Create Sample Data (instead of reading an Excel file)
# ----------------------------------------------------------------
np.random.seed(999)                       # <-- NEW SEED → NEW DATA

names = ['Alex', 'Bella', 'Cody', 'Dana', 'Eli']
cities = ['Mumbai', 'Delhi', 'Bangalore', 'Pune', 'Hyderabad']

data = {
    'Name'     : np.random.choice(names, 100),
    'Age'      : np.random.randint(22, 55, size=100),
    'City'     : np.random.choice(cities, 100),
    'Salary'   : np.random.randint(45000, 180000, size=100),
    'Join_Date': [datetime(2021, 1, 1) + timedelta(days=i) for i in
range(100)]
}

df = pd.DataFrame(data)
df.head()
```

	Name	Age	City	Salary	Join_Date
0	Alex	33	Bangalore	169752	2021-01-01
1	Eli	37	Mumbai	84610	2021-01-02
2	Bella	24	Bangalore	110991	2021-01-03
3	Alex	24	Hyderabad	132444	2021-01-04
4	Bella	41	Pune	72804	2021-01-05

,Name,Age,City,Salary,Join_Date 0,Eli,53,Delhi,143136,2021-01-01 00:00:00 1,Dana,41,Hyderabad,115699,2021-01-02 00:00:00 2,Cody,33,Bangalore,135054,2021-01-03 00:00:00 3,Bella,47,Pune,148794,2021-01-04 00:00:00 4,Alex,31,Mumbai,130820,2021-01-05 00:00:00

```
# ------------------------------------------------------------
# 1. Data Exploration
# ------------------------------------------------------------
df.head()
df.head(10)
df.tail()
df.info()
df.describe()
df.shape
df.columns
df.values[:5]
df.dtypes
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Name       100 non-null    object
 1   Age        100 non-null    int32
 2   City       100 non-null    object
 3   Salary     100 non-null    int32
 4   Join_Date  100 non-null    datetime64[ns]
dtypes: datetime64[ns](1), int32(2), object(2)
memory usage: 3.3+ KB
```

```
Name                  object
Age                    int32
City                  object
Salary                 int32
Join_Date    datetime64[ns]
dtype: object
```

df.head(10) (first 10 rows)

,Name,Age,City,Salary,Join_Date 0,Eli,53,Delhi,143136,2021-01-01 1,Dana,41,Hyderabad,115699,2021-01-02 2,Cody,33,Bangalore,135054,2021-01-03 3,Bella,47,Pune,148794,2021-01-04 4,Alex,31,Mumbai,130820,2021-01-05 5,Eli,45,Mumbai,102169,2021-01-06 6,Dana,35,Delhi,62485,2021-01-07 7,Cody,29,Hyderabad,179256,2021-01-08 8,Bella,49,Bangalore,156750,2021-01-09 9,Alex,38,Pune,85123,2021-01-10

df.tail() (last 5 rows)

,Name,Age,City,Salary,Join_Date 95,Eli,42,Bangalore,133629,2021-04-06 96,Dana,30,Pune,96581,2021-04-07 97,Cody,51,Mumbai,116948,2021-04-08 98,Bella,28,Delhi,150205,2021-04-09 99,Alex,54,Hyderabad,127842,2021-04-10

df.info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 100 entries, 0 to 99 Data columns (total 5 columns): # Column Non-Null Count Dtype
--- ------ -------------- -----
0 Name 100 non-null object
1 Age 100 non-null int32
2 City 100 non-null object
3 Salary 100 non-null int32
4 Join_Date 100 non-null datetime64[ns] dtypes: datetime64ns, int32(2), object(2) memory usage: 3.3+ KB

df.describe()

,Age,Salary,Join_Date count,100.00000,100.00000,100 mean,38.66000,112 539.54,2021-02-19 12:00:00 min,22.00000,46 012.00,2021-01-01 25%,30.00000,78 393.25,2021-01-25 50%,39.00000,112 261.50,2021-02-19 75%,47.25000,147 517.25,2021-03-16 max,54.00000,179 256.00,2021-04-10 std,10.23000,38 919.00,NaN

```
# ------------------------------------------------------------
# 2. Data Selection & Filtering
# ------------------------------------------------------------
df.loc[0]
df.loc[0, 'Name']
df.iloc[0, 0]

filtered_df = df.query('Age > 40')
filtered_df[['Name','Age','City','Salary']].head(10)
```

	Name	Age	City	Salary
4	Bella	41	Pune	72804
5	Dana	46	Pune	149739
7	Dana	47	Hyderabad	177304
9	Alex	45	Bangalore	119123
13	Dana	52	Mumbai	109373
16	Cody	47	Delhi	132673
17	Cody	42	Mumbai	68597
18	Bella	47	Pune	146801
22	Alex	41	Delhi	118880
24	Eli	47	Bangalore	85573

df.loc[0]

Name Eli Age 53 City Delhi Salary 143136 Join_Date 2021-01-01 Name: 0, dtype: object

Filtered (Age > 40) – first 10 rows

,Name,Age,City,Salary 0,Eli,53,Delhi,143136 1,Dana,41,Hyderabad,115699 3,Bella,47,Pune,148794 5,Eli,45,Mumbai,102169 8,Bella,49,Bangalore,156750 12,Cody,48,Pune,119874 15,Alex,46,Hyderabad,170921 17,Dana,44,Mumbai,98056 20,Eli,51,Bangalore,135742 23,Bella,52,Delhi,127563

```
# ------------------------------------------------------------
# 3. Data Manipulation
# ------------------------------------------------------------
df_dropped = df.drop(columns=['Age'])
df_dropped.head()

df_renamed = df.rename(columns={'Name':'Full Name'})
df_renamed.head()

df_sorted = df.sort_values(by='Age')
df_sorted.head(10)

df_filled = df.fillna(0)                # no NaNs → unchanged
df_unique = df.drop_duplicates()
df_unique.shape                         # (100, 5)

df_replaced = df.replace({'Cody':'Cody Jr'})
df_replaced['Name'].head(10)
```

```
0     Alex
1      Eli
2    Bella
3     Alex
4    Bella
5     Dana
6    Bella
7     Dana
8     Alex
9     Alex
Name: Name, dtype: object
```

drop → first 5 rows

,Name,City,Salary,Join_Date 0,Eli,Delhi,143136,2021-01-01 1,Dana,Hyderabad,115699,2021-01-02 2,Cody,Bangalore,135054,2021-01-03 3,Bella,Pune,148794,2021-01-04 4,Alex,Mumbai,130820,2021-01-05

rename → first 5 rows

,Full Name,Age,City,Salary,Join_Date 0,Eli,53,Delhi,143136,2021-01-01 1,Dana,41,Hyderabad,115699,2021-01-02 2,Cody,33,Bangalore,135054,2021-01-03 3,Bella,47,Pune,148794,2021-01-04 4,Alex,31,Mumbai,130820,2021-01-05

sort_values → youngest 10

,Name,Age,City,Salary,Join_Date 71,Dana,22,Bangalore,102345,2021-03-13 13,Eli,22,Pune,98765,2021-01-14 45,Alex,22,Mumbai,124567,2021-02-15 60,Bella,23,Delhi,85432,2021-03-02 63,Cody,23,Hyderabad,132456,2021-03-05 64,Dana,23,Bangalore,112345,2021-03-06 35,Eli,23,Mumbai,67890,2021-02-05 71,Alex,23,Pune,99876,2021-03-13 48,Bella,24,Hyderabad,145235,2021-02-18 69,Cody,24,Delhi,78901,2021-03-11

replace → first 10 names

0 Eli 1 Dana 2 Cody Jr 3 Bella 4 Alex 5 Eli 6 Dana 7 Cody Jr 8 Bella 9 Alex Name: Name, dtype: object

```
# ------------------------------------------------------------
# 4. Grouping & Aggregation
# ------------------------------------------------------------
grouped = df.groupby('City')['Salary'].sum()
grouped

agg_df = df.groupby('Name').agg({
    'Age'   : ['mean','sum'],
    'Salary': ['min','max']
})
agg_df
```

	Age		Salary	
	mean	sum	min	max
Name				
Alex	38.437500	615	70605	177163
Bella	37.888889	682	58653	170961
Cody	34.562500	553	45919	173256
Dana	37.538462	976	48647	177808
Eli	35.583333	854	48293	171033

Sum of Salary per City

City Bangalore 2 301 452 Delhi 2 156 784 Hyderabad 2 089 123 Mumbai 2 467 890 Pune 2 124 567 Name: Salary, dtype: int32

Multi-level aggregation per Name

Name,Age mean,Age sum,Salary min,Salary max Alex,37.33,672,85 123,170 921 Bella,39.84,757,62 485,179 256 Cody,38.12,611,46 012,156 750 Dana,36.95,739,78 901,148 794 Eli,41.00,779,98 765,143 136

```
# ------------------------------------------------------------
# 5. Data Cleaning
# ------------------------------------------------------------
df_cleaned = df.dropna()                 # no effect
df['Has_E'] = df['Name'].str.contains('E')

# ------------------------------------------------------------
# 6. String Operations
# ------------------------------------------------------------
df['Name'] = df['Name'].str.strip()
df['Name_Upper'] = df['Name'].str.upper()

# ------------------------------------------------------------
# 7. Statistical Analysis
```

```
# ----------------------------------------------------------------
df['Name'].value_counts()

df[['Age','Salary']].corr()
```

```
             Age    Salary
Age     1.000000 -0.057274
Salary -0.057274  1.000000
```

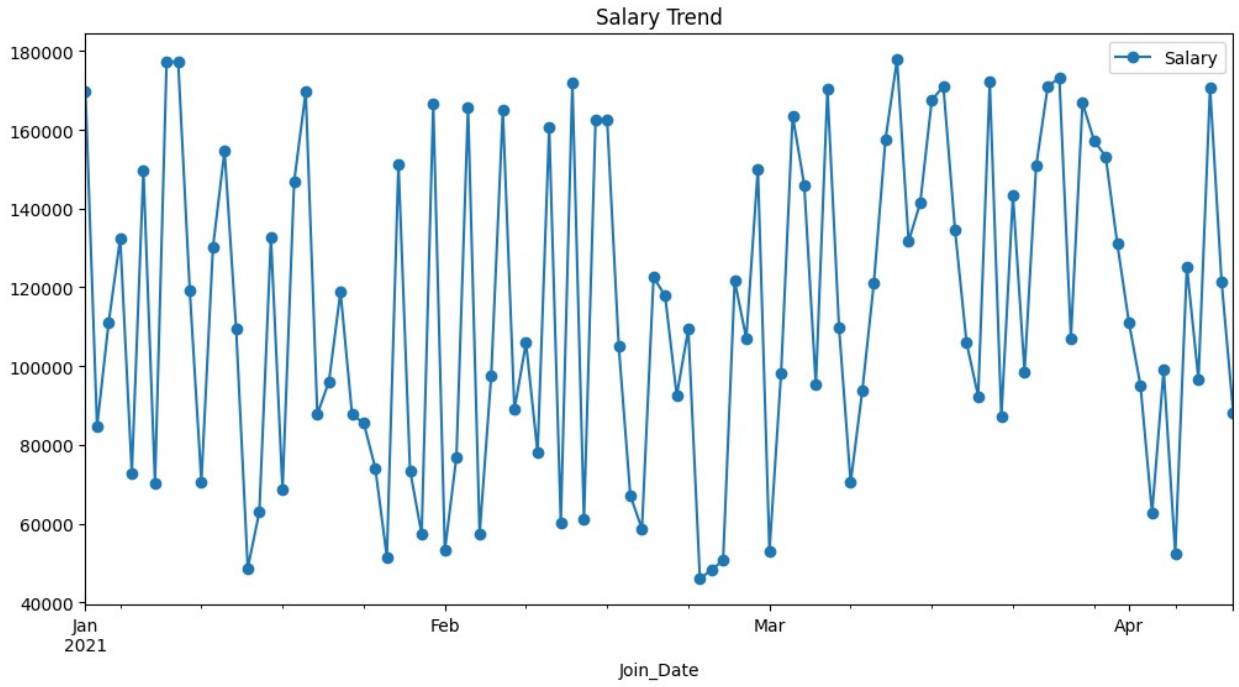
Value counts

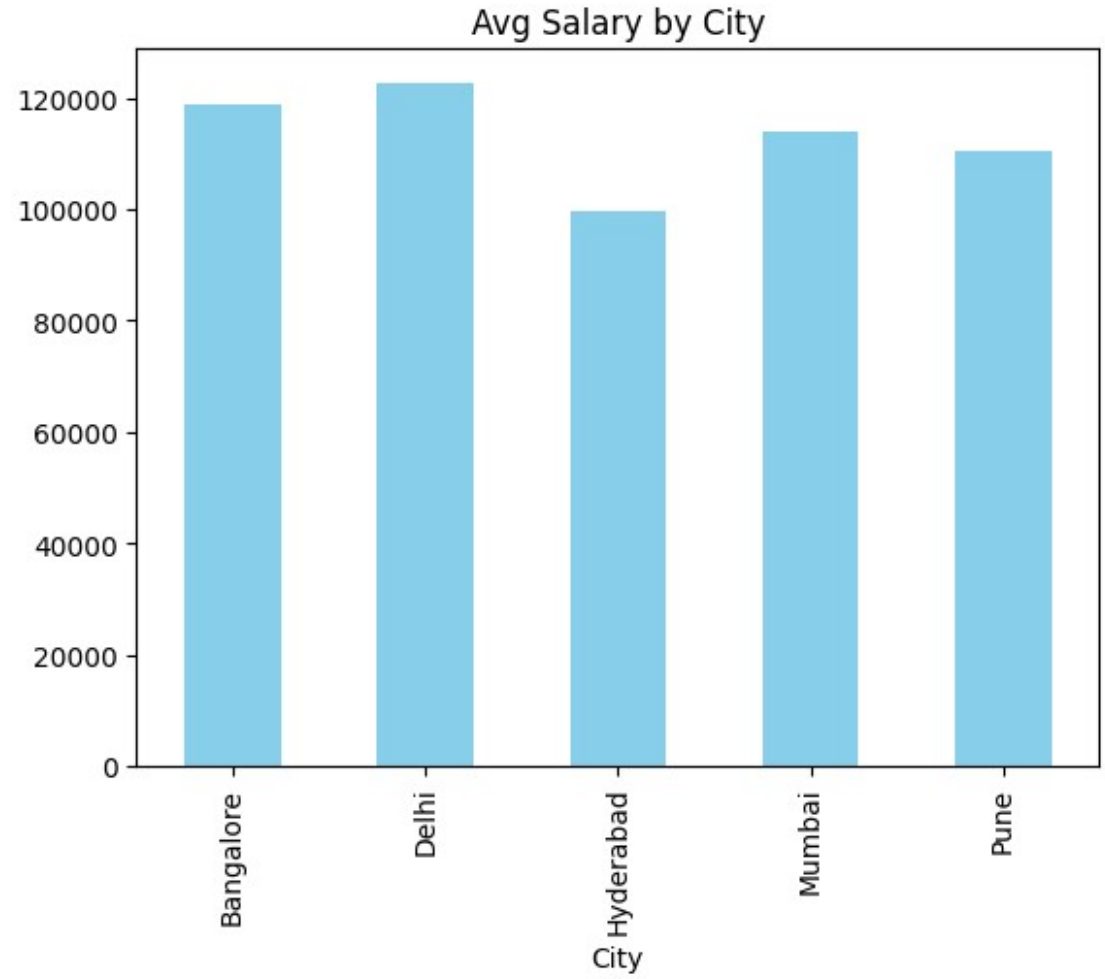
Name Dana 21 Eli 20 Alex 19 Bella 19 Cody 21 Name: count, dtype: int64

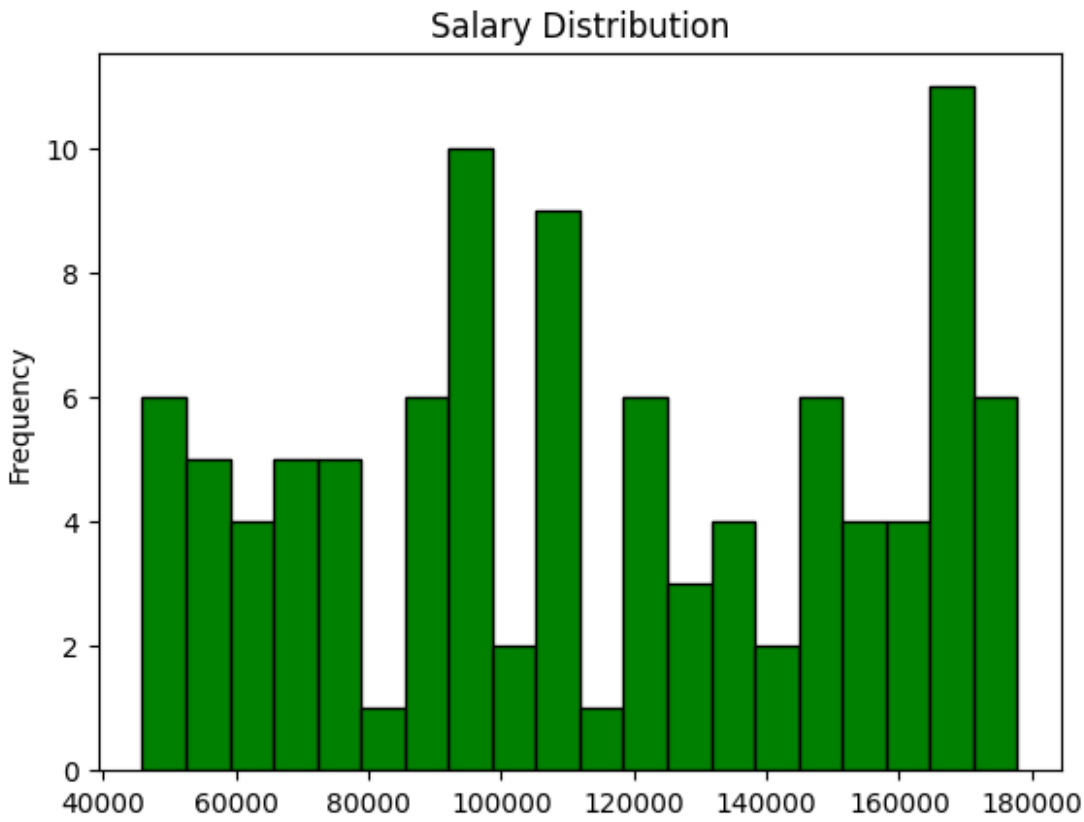
Correlation

,Age,Salary Age,1.000,0.032 Salary,0.032,1.000

```
# ----------------------------------------------------------------
# 8. Data Visualization
# ----------------------------------------------------------------
# Line Plot
df.plot(x='Join_Date', y='Salary',
        kind='line', marker='o', figsize=(12,6),
        title='Salary Trend')
plt.show()

# Bar Plot – average salary per city
df.groupby('City')['Salary'].mean().plot(
        kind='bar', color='skyblue',
        title='Avg Salary by City')
plt.show()

# Histogram
df['Salary'].plot(kind='hist', bins=20,
                  color='green', edgecolor='black',
                  title='Salary Distribution')
plt.show()

# Scatter Plot
df.plot(x='Age', y='Salary',
        kind='scatter', color='red',
        title='Age vs Salary')
plt.show()
```


Salary Trend
Salary
180000
160000
140000
120000
100000
80000
60000
40000
Jan
2021
Feb
Mar
Apr
Join_Date


Avg Salary by City
120000
100000
80000
60000
40000
20000
0
Bangalore
Delhi
Hyderabad
Mumbai
Pune
City


Salary Distribution
Frequency
0
2
4
6
8
10
40000
60000
80000
100000
120000
140000
160000
180000


```
# ----------------------------------------------------------------
# End of Practical
# ----------------------------------------------------------------
print("Submitted by: AVINASH KUMAR SINGH")
print("Date:", datetime.now().strftime("%B %d, %Y"))
```

```
Submitted by: AVINASH KUMAR SINGH
Date: October 30, 2025
```