

**1.**

```
import pandas as pd
```

```
data = {
```

```
    'DEPARTMENT_ID': [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 260, 270],
```

```
    'DEPARTMENT_NAME': ['Administration', 'Marketing', 'Purchasing', 'Human Resources', 'Shipping', 'IT', 'Public Relations', 'Sales', 'Executive', 'Finance', 'Accounting', 'Treasury', 'Corporate Tax', 'Control And Credit', 'Shareholder Services', 'Benefits', 'Manufacturing', 'Construction', 'Contracting', 'Operations', 'IT Support', 'NOC', 'IT Helpdesk', 'Government Sales', 'Retail Sales', 'Recruiting', 'Payroll'],
```

```
    'MANAGER_ID': [200, 201, 114, 203, 121, 103, 204, 145, 100, 108, 205, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
```

```
    'LOCATION_ID': [1700, 1800, 1700, 2400, 1500, 1400, 2700, 2500, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700]
```

```
}
```

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

```
distinct_department_ids = df['DEPARTMENT_ID'].unique()
```

```
print(distinct_department_ids)
```

**2.**

```
import pandas as pd
```

```
data = {
```

```
    'EMPLOYEE_ID': [102, 101, 101, 201, 114, 122, 200, 176, 176, 200],
```

```
    'START_DATE': ['2001-01-13', '1997-09-21', '2001-10-28', '2004-02-17', '2006-03-24', '2007-01-01', '1995-09-17', '2006-03-24', '2007-01-01', '2002-07-01'],
```

```
    'END_DATE': ['2006-07-24', '2001-10-27', '2005-03-15', '2007-12-19', '2007-12-31', '2007-12-31', '2001-06-17', '2006-12-31', '2007-12-31', '2006-12-31'],
```

```
    'JOB_ID': ['IT_PROG', 'AC_ACCOUNT', 'AC_MGR', 'MK_REP', 'ST_CLERK', 'ST_CLERK', 'AD_ASST', 'SA_REP', 'SA_MAN', 'AC_ACCOUNT'],
```

```
    'DEPARTMENT_ID': [60, 110, 110, 20, 50, 50, 90, 80, 80, 90]
```

```
}
```

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

```
job_counts = df.groupby('EMPLOYEE_ID').size()
```

```
employees_with_multiple_jobs = job_counts[job_counts >= 2].index
```

```
print(employees_with_multiple_jobs.tolist())
```

**3.**

```
import pandas as pd
```

```
data = {
```

```
    'JOB_ID': ['AD_PRES', 'AD_VP', 'AD_ASST', 'FI_MGR', 'FI_ACCOUNT', 'AC_MGR', 'AC_ACCOUNT',  
              'SA_MAN', 'SA_REP', 'PU_MAN', 'PU_CLERK', 'ST_MAN', 'ST_CLERK', 'SH_CLERK',  
              'IT_PROG', 'MK_MAN', 'MK_REP', 'HR_REP', 'PR_REP'],
```

```
    'JOB_TITLE': ['President', 'Administration Vice President', 'Administration Assistant',  
                 'Finance Manager', 'Accountant', 'Accounting Manager', 'Public Accountant',  
                 'Sales Manager', 'Sales Representative', 'Purchasing Manager', 'Purchasing Clerk',  
                 'Stock Manager', 'Stock Clerk', 'Shipping Clerk', 'Programmer',  
                 'Marketing Manager', 'Marketing Representative', 'Human Resources Representative',  
                 'Public Relations Representative'],
```

```
    'MIN_SALARY': [20080, 15000, 3000, 8200, 4200, 8200, 4200, 10000, 6000, 8000, 2500, 5500,  
                  2008,
```

```
                  2500, 4000, 9000, 4000, 4000, 4500],
```

```
    'MAX_SALARY': [40000, 30000, 6000, 16000, 9000, 16000, 9000, 20080, 12008, 15000, 5500, 8500,  
                  5000, 5500, 10000, 15000, 9000, 9000, 10500]
```

```
}
```

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

```
df_sorted = df.sort_values(by='JOB_TITLE', ascending=False)
```

```
print(df_sorted)
```

**4.**

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
import yfinance as yf
```

```
ticker = 'GOOGL'
```

```
start_date = '2023-01-01'
```

```
end_date = '2023-12-31'
```

```
stock_data = yf.download(ticker, start=start_date, end=end_date)
```

```
print(stock_data.head())

plt.figure(figsize=(12, 6))

plt.plot(stock_data['Close'], label='GOOGL Close Price', color='b')

plt.title('Alphabet Inc. (GOOGL) Stock Prices')

plt.xlabel('Date')

plt.ylabel('Close Price (USD)')

plt.legend()

plt.grid(True)

plt.show()
```

## 5.

```
import pandas as pd

import matplotlib.pyplot as plt

import yfinance as yf

ticker = 'GOOGL'

start_date = '2023-01-01'

end_date = '2023-12-31'

stock_data = yf.download(ticker, start=start_date, end=end_date)

print(stock_data.head())

plt.figure(figsize=(12, 6))

plt.bar(stock_data.index, stock_data['Volume'], color='b')

plt.title('Alphabet Inc. (GOOGL) Trading Volume')

plt.xlabel('Date')

plt.ylabel('Volume')

plt.grid(True)

plt.show()
```

## 6.

```
import pandas as pd

import matplotlib.pyplot as plt
```

```

data = {
    'Date': ['01-04-2020', '02-04-2020', '03-04-2020', '06-04-2020', '07-04-2020', '08-04-2020',
            '09-04-2020', '13-04-2020', '14-04-2020', '15-04-2020', '16-04-2020', '17-04-2020',
            '20-04-2020', '21-04-2020', '22-04-2020', '23-04-2020', '24-04-2020', '27-04-2020',
            '28-04-2020', '29-04-2020', '30-04-2020', '01-05-2020'],
    'Open': [1122, 1098.26, 1119.015, 1138, 1122, 1206.5, 1224.05, 1209.18, 1242.09, 1245.61,
            1249.81, 1284.85, 1271, 1247, 1271, 1265.1, 1260, 1288, 1287.93, 1341.44, 1324.86, 1328.5],
    'High': [1129.69, 1126.86, 1135, 1194.66, 1138, 1219.07, 1232.57, 1220.09, 1263.93, 1280,
            1294.6, 1294.3, 1281.6, 1254.27, 1284.13, 1280.4, 1271, 1296.15, 1323.82, 1358.99,
            1355.12, 1352.07],
    'Low': [1097.45, 1096.4, 1079.81, 1130.94, 1107.38, 1188.16, 1198.735, 1187.598, 1236.93,
            1240.4,
            1240.62, 1271.23, 1267.45, 1209.71, 1261.7, 1249.45, 1259.2, 1278.55, 1289.04, 1325.34,
            1323.49, 1311],
    'Close': [1105.62, 1120.84, 1097.82, 1186.92, 1113.28, 1210.28, 1211.45, 1217.56, 1269.23,
            1262.47,
            1242.62, 1283.25, 1269.44, 1216.34, 1276.31, 1279.31, 1261.12, 1275.88, 1322.67, 1348.66,
            1348.66, 1320.61],
    'Adj Close': [1105.62, 1120.84, 1097.82, 1186.92, 1113.28, 1210.28, 1211.45, 1217.56, 1269.23,
            1262.47,
            1242.62, 1283.25, 1269.44, 1216.34, 1276.31, 1279.31, 1261.12, 1275.88, 1322.67,
            1348.66,
            1348.66, 1320.61],
    'Volume': [2343100, 1964900, 2314000, 2664700, 2337300, 1971500, 2157400, 1739800,
            2175400, 1671700,
            2151800, 1949000, 1695300, 2153000, 2134000, 1566200, 1542700, 1600600, 1404300,
            3794300,
            2665400, 2072500]
}

df = pd.DataFrame(data)
df['Date'] = pd.to_datetime(df['Date'], dayfirst=True)
start_date = '2020-04-01'
end_date = '2020-05-01'

```

```

filtered_data = df[(df['Date'] >= start_date) & (df['Date'] <= end_date)]

plt.figure(figsize=(10, 6))

plt.scatter(filtered_data['Volume'], filtered_data['Close'], alpha=0.5)

plt.title('Trading Volume vs. Stock Prices of Alphabet Inc.')

plt.xlabel('Trading Volume')

plt.ylabel('Stock Price (Close)')

plt.grid(True)

plt.show()

```

## 7.

```

import pandas as pd

data = {
    'Date': ['2024-07-01', '2024-07-01', '2024-07-02', '2024-07-02',
            '2024-07-03', '2024-07-03', '2024-07-04', '2024-07-04'],
    'Item': ['Item_A', 'Item_B', 'Item_A', 'Item_B',
            'Item_A', 'Item_B', 'Item_A', 'Item_B'],
    'Sales': [100, 200, 150, 300, 100, 250, 200, 100]
}

df = pd.DataFrame(data)

df['Date'] = pd.to_datetime(df['Date'])

pivot_table = df.pivot_table(values='Sales', index='Item', aggfunc=['max', 'min'])

pivot_table.columns = ['Max_Sales', 'Min_Sales']

print(pivot_table)

```

## 8.

```

import pandas as pd

data = {
    'TransactionID': [1, 2, 3, 4, 5, 6],

```

```

'Item': ['Item_A', 'Item_B', 'Item_A', 'Item_C', 'Item_B', 'Item_A'],
'UnitsSold': [10, 5, 7, 3, 6, 8],
'PricePerUnit': [2.5, 5.0, 2.5, 7.0, 5.0, 2.5]
}

df = pd.DataFrame(data)

pivot_table = pd.pivot_table(df, values='UnitsSold', index='Item', aggfunc='sum')

print(pivot_table)

```

## 9.

```

import pandas as pd

data = {

    'OrderDate': ['1-6-18', '1-23-18', '2-9-18', '2-26-18', '3-15-18', '4-1-18', '4-18-18', '5-5-18', '5-22-18',
'6-8-18',

                '6-25-18', '7-12-18', '7-29-18', '8-15-18', '9-1-18', '9-18-18', '10-5-18', '10-22-18'],

    'Region': ['East', 'Central', 'Central', 'Central', 'West', 'East', 'Central', 'Central', 'West', 'East',

              'Central', 'East', 'East', 'East', 'Central', 'East', 'Central', 'East'],

    'Manager': ['Martha', 'Hermann', 'Hermann', 'Timothy', 'Timothy', 'Martha', 'Martha', 'Hermann',
'Douglas', 'Martha',

               'Hermann', 'Martha', 'Douglas', 'Martha', 'Douglas', 'Martha', 'Hermann', 'Martha'],

    'SalesMan': ['Alexander', 'Shelli', 'Luis', 'David', 'Stephen', 'Alexander', 'Steven', 'Luis', 'Michael',
'Alexander',

               'Sigal', 'Diana', 'Karen', 'Alexander', 'John', 'Alexander', 'Sigal', 'Alexander'],

    'Item': ['Television', 'Home Theater', 'Television', 'Cell Phone', 'Television', 'Home Theater',
'Television', 'Television',

            'Television', 'Home Theater', 'Television', 'Home Theater', 'Home Theater', 'Television', 'Desk',
'Video Games',

            'Home Theater', 'Cell Phone'],

    'Units': [95, 50, 36, 27, 56, 60, 75, 90, 32, 60, 90, 29, 81, 35, 2, 16, 28, 64],

    'Unit_price': [1198.00, 500.00, 1198.00, 225.00, 1198.00, 500.00, 1198.00, 1198.00, 1198.00,
500.00, 1198.00, 500.00,

                 500.00, 1198.00, 125.00, 58.50, 500.00, 225.00],

    'Sale_amt': [113810.00, 25000.00, 43128.00, 6075.00, 67088.00, 30000.00, 89850.00, 107820.00,
38336.00, 30000.00,

```

```

        107820.00, 14500.00, 40500.00, 41930.00, 250.00, 936.00, 14000.00, 14400.00]
    }
    df = pd.DataFrame(data)
    pivot_table = pd.pivot_table(
        df,
        values='Sale_amt',
        index=['Region', 'Manager', 'SalesMan'],
        aggfunc='sum'
    )
    print(pivot_table)

```

## 10.

```

import pandas as pd
import numpy as np
np.random.seed(0)
df = pd.DataFrame(np.random.randn(10, 5), columns=list('ABCDE'))
def highlight_negatives(val):
    color = 'red' if val < 0 else 'black'
    return f'color: {color}'
styled_df = df.style.map(highlight_negatives)
styled_df
styled_df.to_html('styled_dataframe.html')
import webbrowser
webbrowser.open('styled_dataframe.html')

```

## 11.

```

import pandas as pd
import numpy as np
np.random.seed(0)

```

```

data = np.random.randn(10, 4)
df = pd.DataFrame(data, columns=['A', 'B', 'C', 'D'])
df.iloc[2, 0] = np.nan
df.iloc[5, 2] = np.nan
df.iloc[8, 3] = np.nan
def highlight_nan(value):
    if pd.isna(value):
        return 'background-color: yellow'
    else:
        return ""
styled_df = df.style.applymap(highlight_nan)
styled_df
styled_df.to_html('styled_dataframe.html')
import webbrowser
webbrowser.open('styled_dataframe.html')

```

## 12.

```

import pandas as pd
import numpy as np
np.random.seed(0)
df = pd.DataFrame(np.random.randn(10, 5), columns=['A', 'B', 'C', 'D', 'E'])
df.loc[:, 'C'] = np.nan
styled_df = df.style.applymap(lambda x: 'background-color: black; color: yellow')
styled_df
styled_df.to_html('styled_dataframe.html')
import webbrowser
webbrowser.open('styled_dataframe.html')

```

## 13.

```

import pandas as pd
import numpy as np
data = {

```



```

'ord_no': [70001.0, np.nan, 70002.0, 70004.0, np.nan, 70005.0, np.nan, 70010.0, 70003.0,
70012.0, np.nan, 70013.0],

'purch_amt': [150.50, 270.65, 65.26, 110.50, 948.50, 2400.60, 5760.00, 1983.43, 2480.40, 250.45,
75.29, 3045.60],

'ord_date': ['2012-10-05', '2012-09-10', np.nan, '2012-08-17', '2012-09-10', '2012-07-27', '2012-09-
10', '2012-10-10', '2012-10-10', '2012-06-27', '2012-08-17', '2012-04-25'],

'customer_id': [3002, 3001, 3001, 3003, 3002, 3001, 3001, 3004, 3003, 3002, 3001, 3001],

'salesman_id': [5002.0, 5003.0, 5001.0, np.nan, 5002.0, 5001.0, 5001.0, np.nan, 5003.0, 5002.0,
5003.0, np.nan]
}

df = pd.DataFrame(data)

missing_values = df.isna()

print(missing_values)

```

#### 14.

```

import pandas as pd

import numpy as np

data = {

'ord_no': [70001.0, np.nan, 70002.0, 70004.0, np.nan, 70005.0, '--', 70010.0, 70003.0, 70012.0,
np.nan, 70013.0],

'purch_amt': [150.5, 270.65, 65.26, 110.5, 948.5, 2400.6, 5760.0, '?', 12.43, 2480.4, 250.45,
3045.6],

'ord_date': ['?', '2012-09-10', np.nan, '2012-08-17', '2012-09-10', '2012-07-27', '2012-09-10', '2012-
10-10', '2012-10-10', '2012-06-27', '2012-08-17', '2012-04-25'],

'customer_id': [3002, 3001, 3001, 3003, 3002, 3001, 3001, 3004, '--', 3002, 3001, 3001],

'salesman_id': [5002.0, 5003.0, '?', 5001.0, np.nan, 5002.0, 5001.0, '?', 5003.0, 5002.0, 5003.0, '--]
}

df = pd.DataFrame(data)

df.replace(['?', '--'], np.nan, inplace=True)

missing_values = df.isna()

print(missing_values)

```

```
print("\nDataFrame after replacing missing values:")
print(df)
```

## 15

```
import pandas as pd
import numpy as np

data = {
    'ord_no': [np.nan, np.nan, 70002.0, np.nan, np.nan, 70005.0, np.nan, 70010.0, 70003.0, 70012.0,
np.nan, np.nan],
    'purch_amt': [np.nan, 270.65, 65.26, np.nan, 948.50, 2400.60, 5760.00, 1983.43, 2480.40, 250.45,
75.29, np.nan],
    'ord_date': [np.nan, '2012-09-10', np.nan, np.nan, '2012-09-10', '2012-07-27', '2012-09-10', '2012-
10-10', '2012-10-10', '2012-06-27', '2012-08-17', np.nan],
    'customer_id': [np.nan, 3001.0, 3001.0, np.nan, 3002.0, 3001.0, 3001.0, 3004.0, 3003.0, 3002.0,
3001.0, np.nan]
}

df = pd.DataFrame(data)

df_with_at_least_2_nans = df[df.isnull().sum(axis=1) >= 2]

print(df_with_at_least_2_nans)
```

## 16

```
import pandas as pd

df = pd.DataFrame({'school': ['V', 'V', 'VI', 'VI', 'V', 'VI'],
    'class': ['- ', 'V', 'VI', 'VI', 'V', 'VI'],
    'name': ['Alberto Franco', 'Gino Mcneill', 'Ryan Parkes',
    'Eesha Hinton', 'Gino Mcneill', 'David Parkes'],
    'date_Of_Birth': ['15/05/2002', '17/05/2002', '16/02/1999',
    '25/09/1998', '11/05/2002', '15/09/1997'],
    'age': [12, 12, 13, 13, 14, 12],
    'height': [173, 192, 186, 167, 151, 159],
    'weight': [35, 32, 33, 30, 31, 32]})

grouped_by_school_code = df.groupby('school')
```

```
print(type(grouped_by_school_code))
```

**17**

```
import pandas as pd

df = pd.DataFrame({
    'school': ['V', 'V', 'VI', 'VI', 'V', 'VI'],
    'age': [12, 12, 13, 13, 14, 12]
})

school_stats = df.groupby('school')['age'].describe()

print(school_stats)
```

**18**

```
import pandas as pd

data = {
    'school': ['s001', 's002', 's003', 's001', 's002', 's004'],
    'class': ['V', 'V', 'VI', 'VI', 'V', 'VI'],
    'name': ['Alberto Franco', 'Gino Mcneill', 'Ryan Parkes', 'Eesha Hinton', 'Gino Mcneill', 'David Parkes'],
    'date_Of_Birth': ['15/05/2002', '17/05/2002', '16/02/1999', '25/09/1998', '11/05/2002', '15/09/1997'],
    'age': [12, 12, 13, 13, 14, 12],
    'height': [173, 192, 186, 167, 151, 159],
    'weight': [35, 32, 33, 30, 31, 32],
    'address': ['street1', 'street2', 'street3', 'street1', 'street2', 'street4']
}

df = pd.DataFrame(data)

df['date_Of_Birth'] = pd.to_datetime(df['date_Of_Birth'], dayfirst=True)

grouped = df.groupby(['school', 'class'])

for name, group in grouped:
    print(name)
    print(group)
    print()
```

**19**

```
import pandas as pd

df = pd.DataFrame({
    'Year': [1986, 1986, 1985, 1986, 1987],
    'WHO region': ['Western Pacific', 'Americas', 'Africa', 'Americas', 'Americas'],
    'Country': ['Viet Nam', 'Uruguay', 'Cte d'Ivoire', 'Colombia', 'Saint Kitts and Nevis'],
    'Beverage Types': ['Wine', 'Other', 'Wine', 'Beer', 'Beer'],
    'Display Value': [0.00, 0.50, 1.62, 4.27, 1.98]})

print(df.shape)

print(df.columns.tolist())
```

**20**

```
import pandas as pd

df = pd.DataFrame({'name': ['Alice Smith', 'Bob Johnson', 'Charlie Brown',
                             'David Miller', 'Emily Williams', 'Frank Garcia']})

substring = "er"

indices = [i for i, name in df['name'].items() if substring in name]

print("Indices using list comprehension:", indices)

positions = df['name'].apply(lambda x: x.find(substring))

print("Positions using apply():")

print(positions)
```

**21**

```
import pandas as pd

df = pd.DataFrame({'name': ['Alice Smith', 'BoB JohnsoN', 'CHarlie Brown'],
                   'age': [25, 30, 28]})

column_name = 'name'

df[column_name] = df[column_name].str.swapcase()

print(df)
```

## 22

```
import matplotlib.pyplot as plt
x = [10, 20, 30, 40, 50]
y = [10, 20, 30, 40, 50]
plt.plot(x, y)
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("Line Graph")
plt.show()
```

## 23

```
import matplotlib.pyplot as plt
filename = "test.txt"
try:
    with open(filename, 'r') as file:
        data = [line.strip().split() for line in file]
        x = [float(item[0]) for item in data]
        y = [float(item[1]) for item in data]
        plt.plot(x, y)
        plt.xlabel("X-axis")
        plt.ylabel("Y-axis")
        plt.title("Line Graph")
        plt.show()
except FileNotFoundError:
    print(f"Error: File '{filename}' not found.")
except ValueError:
    print("Error: Invalid data format in the file.")
```

## 24

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

```

fdata = pd.DataFrame({
    'Date': ['10-03-16', '10-04-16', '10-05-16', '10-06-16', '10-07-16'],
    'Open': [774.25, 776.030029, 779.309998, 779, 779.659973],
    'High': [776.065002, 778.710022, 782.070007, 780.47998, 779.659973],
    'Low': [769.5, 772.890015, 775.650024, 775.539978, 770.75],
    'Close': [772.559998, 776.429993, 776.469971, 776.859985, 775.080017]
})
fdata['Date'] = pd.to_datetime(fdata['Date'])
fdata.set_index('Date', inplace=True)
fig, ax = plt.subplots(figsize=(10, 6))
ax.plot(fdata.index, fdata['Open'], label='Open')
ax.plot(fdata.index, fdata['High'], label='High')
ax.plot(fdata.index, fdata['Low'], label='Low')
ax.plot(fdata.index, fdata['Close'], label='Close')
ax.set_title('Alphabet Inc. Financial Data (Oct 3-7, 2016)')
ax.set_xlabel('Date')
ax.set_ylabel('Price ($)')
ax.legend()
plt.show()

```

## 25

```

import matplotlib.pyplot as plt
x = [0, 1, 2, 3, 4, 5]
y1 = [0, 1, 4, 9, 16, 25]
y2 = [0, 1, 8, 27, 64, 125]
plt.plot(x, y1, color='blue', linewidth=2, label='y = x^2')
plt.plot(x, y2, color='red', linewidth=4, label='y = x^3')
plt.title('Plot of two lines with legends, different widths, and colors')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.legend()

```

```
plt.show()
```

## 26

```
import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(0, 10, 100)
y1 = np.sin(x)
y2 = np.cos(x)
y3 = np.tan(x)
y4 = x**2
fig, axs = plt.subplots(2, 2, figsize=(10, 8))
axs[0, 0].plot(x, y1, color='blue', linewidth=2, label='sin(x)')
axs[0, 0].set_title('Plot of sin(x)')
axs[0, 0].legend()
axs[0, 1].plot(x, y2, color='green', linewidth=2, label='cos(x)')
axs[0, 1].set_title('Plot of cos(x)')
axs[0, 1].legend()
axs[1, 0].plot(x, y3, color='red', linewidth=2, label='tan(x)')
axs[1, 0].set_title('Plot of tan(x)')
axs[1, 0].set_ylim(-10, 10)
axs[1, 0].legend()
axs[1, 1].plot(x, y4, color='purple', linewidth=2, label='x^2')
axs[1, 1].set_title('Plot of x^2')
axs[1, 1].legend()
fig.suptitle('Multiple Plots Example')
plt.tight_layout()
plt.subplots_adjust(top=0.9)
plt.show()
```

## 27

```
import matplotlib.pyplot as plt
```

```

languages = ['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++']
popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]

plt.figure(figsize=(10, 6))

plt.bar(languages, popularity, color=['blue', 'green', 'red', 'purple', 'orange', 'cyan'])

plt.title('Popularity of Programming Languages')

plt.xlabel('Programming Languages')

plt.ylabel('Popularity (%)')

plt.show()

```

## 28

```

import matplotlib.pyplot as plt

languages = ['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++']
popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]

plt.figure(figsize=(10, 6))

plt.barh(languages, popularity, color=['blue', 'green', 'red', 'purple', 'orange', 'cyan'])

plt.title('Popularity of Programming Languages')

plt.xlabel('Popularity (%)')

plt.ylabel('Programming Languages')

plt.show()

```

## 29

```

import matplotlib.pyplot as plt

languages = ['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++']
popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]
colors = ['blue', 'green', 'red', 'purple', 'orange', 'cyan']

plt.figure(figsize=(10, 6))

plt.bar(languages, popularity, color=colors)

plt.title('Popularity of Programming Languages')

plt.xlabel('Programming Languages')

plt.ylabel('Popularity (%)')

plt.show()

```



**30**

```
import matplotlib.pyplot as plt
import numpy as np
groups = ['A', 'B', 'C', 'D', 'E']
means_men = [22, 30, 35, 35, 26]
means_women = [25, 32, 30, 35, 29]
n_groups = len(groups)
index = np.arange(n_groups)
bar_width = 0.35
fig, ax = plt.subplots(figsize=(10, 6))
bars1 = ax.bar(index, means_men, bar_width, label='Men', color='blue')
bars2 = ax.bar(index + bar_width, means_women, bar_width, label='Women', color='pink')
ax.set_xlabel('Group')
ax.set_ylabel('Scores')
ax.set_title('Scores by group and gender')
ax.set_xticks(index + bar_width / 2)
ax.set_xticklabels(groups)
ax.legend()
plt.show()
```

**31**

```
import numpy as np
import matplotlib.pyplot as plt
means_men = [22, 30, 35, 35, 26]
means_women = [25, 32, 30, 35, 29]
std_men = [4, 3, 4, 1, 5]
std_women = [3, 5, 2, 3, 3]
ind = np.arange(len(means_men))
width = 0.35
fig, ax = plt.subplots()
```

```

men_bars = ax.bar(ind, means_men, width, yerr=std_men, label='Men', capsize=5)

women_bars = ax.bar(ind, means_women, width, bottom=means_men, yerr=std_women,
label='Women', capsize=5)

ax.set_xlabel('Groups')

ax.set_ylabel('Scores')

ax.set_title('Scores by group and gender with error bars')

ax.set_xticks(ind)

ax.set_xticklabels(('G1', 'G2', 'G3', 'G4', 'G5'))

ax.legend()

plt.show()

```

### 32

```

import numpy as np

import matplotlib.pyplot as plt

np.random.seed(0)

x = np.random.randn(100)

y = np.random.randn(100)

plt.scatter(x, y, alpha=0.7, edgecolors='w', s=100)

plt.xlabel('X-axis')

plt.ylabel('Y-axis')

plt.title('Scatter Plot of Random Distributions')

plt.show()

```

### 33

```

import numpy as np

import matplotlib.pyplot as plt

np.random.seed(0)

x = np.random.randn(100)

y = np.random.randn(100)

plt.scatter(x, y, facecolors='none', edgecolors='b')

plt.xlabel('X-axis')

```

```
plt.ylabel('Y-axis')
plt.title('Scatter Plot of Random Distributions with Empty Circles')
plt.show()
```

### 34

```
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(0)
x = np.random.randn(50)
y = np.random.randn(50)
sizes = np.random.randint(10, 100, 50)
plt.scatter(x, y, s=sizes, alpha=0.7)
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Scatter Plot with Randomly Sized Balls')
plt.show()
```

### 35

```
import matplotlib.pyplot as plt
math_marks = [88, 92, 80, 89, 100, 80, 60, 100, 80, 34]
science_marks = [35, 79, 79, 48, 100, 88, 32, 45, 20, 30]
marks_range = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
plt.figure(figsize=(10, 6))
plt.scatter(marks_range, math_marks, color='red', label='Math marks')
plt.scatter(marks_range, science_marks, color='green', label='Science marks')
plt.xlabel('Marks Range')
plt.ylabel('Marks Scored')
plt.title('Scatter Plot')
plt.legend()
plt.xlim(0, 110)
plt.ylim(0, 110)
```

```
plt.grid(True)
plt.show()
```

### 36

```
import matplotlib.pyplot as plt
import numpy as np
np.random.seed(0)
weights1 = np.random.uniform(55, 65, 10)
heights1 = np.random.uniform(120, 180, 10)
weights2 = np.random.uniform(60, 70, 10)
heights2 = np.random.uniform(140, 200, 10)
weights3 = np.random.uniform(65, 75, 10)
heights3 = np.random.uniform(160, 220, 10)
plt.figure(figsize=(10, 6))
plt.scatter(weights1, heights1, color='blue', marker='*', label='Group 1')
plt.scatter(weights2, heights2, color='red', marker='*', label='Group 2')
plt.scatter(weights3, heights3, color='green', marker='*', label='Group 3')
plt.xlabel('Weight')
plt.ylabel('Height')
plt.title('Group wise Weight vs Height scatter plot')
plt.xlim(55, 75)
plt.ylim(100, 240)
plt.legend()
plt.grid(True)
plt.show()
```

### 37

```
import pandas as pd
data = {'X': [78, 85, 96, 80, 86],
        'Y': [84, 94, 89, 83, 86],
```

```
    'Z': [86, 97, 96, 72, 83]}

df = pd.DataFrame(data)

print(df)
```

### 38

```
import pandas as pd

import numpy as np

exam_data = {

    'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],

    'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],

    'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],

    'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']

}

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

df = pd.DataFrame(exam_data, index=labels)

print(df)
```

### 39

```
import pandas as pd

import numpy as np

exam_data = {

    'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],

    'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],

    'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],

    'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']

}

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

df = pd.DataFrame(exam_data, index=labels)

first_three_rows = df.head(3)

print("First three rows of the data frame:")
```

```
print(first_three_rows)
```

**40**

```
import pandas as pd
```

```
import numpy as np
```

```
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura',  
                      'Kevin', 'Jonas'],
```

```
              'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
```

```
              'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
```

```
              'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
```

```
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

```
df = pd.DataFrame(exam_data, index=labels)
```

```
result = df[['name', 'score']]
```

```
print(result)
```