

```
import numpy as np
student_scores = np.array([
    [85, 78, 92, 88],
    [76, 85, 90, 80],
    [90, 88, 84, 92],
    [70, 75, 80, 85]
])
average_scores = np.mean(student_scores, axis=0)
subjects = ['Math', 'Science', 'English', 'History']
highest_avg_score_index = np.argmax(average_scores)
highest_avg_subject = subjects[highest_avg_score_index]
print("Average scores for each subject:", average_scores)
print("Subject with the highest average score:", highest_avg_subject)
```

➡ Average scores for each subject: [80.25 81.5 86.5 86.25]
Subject with the highest average score: English

```
import numpy as np
sales_data = np.array([
    [10, 15.5, 155.0],
    [20, 10.0, 200.0],
    [15, 20.0, 300.0]
])
average_price_per_unit = np.mean(sales_data[:, 1])
print("Average price of all products sold:", average_price_per_unit)
```

➡ Average price of all products sold: 15.166666666666666

```
import pandas as pd
data = {
    'bedrooms': [3, 4, 2, 3, 5],
    'square_footage': [1500, 2500, 900, 1800, 3200],
    'sale_price': [300000, 500000, 200000, 350000, 650000]
}

house_data = pd.DataFrame(data)
print(house_data)
average_bedrooms = house_data['bedrooms'].mean()
print("Average number of bedrooms:", average_bedrooms)
average_square_footage = house_data['square_footage'].mean()
print("Average square footage:", average_square_footage)
average_sale_price = house_data['sale_price'].mean()
print("Average sale price:", average_sale_price)
house_max_square_footage = house_data.loc[house_data['square_footage'].idxmax()]
print("House with maximum square footage:")
print(house_max_square_footage)
house_max_sale_price = house_data.loc[house_data['sale_price'].idxmax()]
print("House with highest sale price:")
print(house_max_sale_price)
```

➡

	bedrooms	square_footage	sale_price
0	3	1500	300000
1	4	2500	500000
2	2	900	200000
3	3	1800	350000
4	5	3200	650000

Average number of bedrooms: 3.4
Average square footage: 1980.0
Average sale price: 400000.0
House with maximum square footage:
bedrooms 5
square_footage 3200
sale_price 650000
Name: 4, dtype: int64
House with highest sale price:
bedrooms 5
square_footage 3200
sale_price 650000
Name: 4, dtype: int64

```
import numpy as np
sales_data = np.array([1000, 1500, 2000, 2500])
total_sales = np.sum(sales_data)
print("Total sales for the year:", total_sales)
percentage_increase = ((sales_data[3] - sales_data[0]) / sales_data[0]) * 100
print("Percentage increase in sales from Q1 to Q4:", percentage_increase)
```

→ Total sales for the year: 7000
Percentage increase in sales from Q1 to Q4: 150.0

```
import numpy as np
fuel_efficiency = np.array([25, 30, 22, 35, 28, 32])
average_fuel_efficiency = np.mean(fuel_efficiency)
print("Average fuel efficiency:", average_fuel_efficiency)
efficiency_model_1 = fuel_efficiency[1]
efficiency_model_2 = fuel_efficiency[4]
percentage_improvement = ((efficiency_model_2 - efficiency_model_1) / efficiency_model_1) * 100
print("Percentage improvement in fuel efficiency between model 2 and model 5:", percentage_improvement)
```

→ Average fuel efficiency: 28.666666666666668
Percentage improvement in fuel efficiency between model 2 and model 5: -6.666666666666667

```
import pandas as pd
from datetime import datetime
data = {
    'Employee ID': [101, 102, 103, 104, 105],
    'Department': ['HR', 'Finance', 'IT', 'HR', 'Finance'],
    'Salary': [60000, 80000, 90000, 75000, 85000],
    'Joining Date': ['2015-01-15', '2017-08-23', '2016-06-30', '2018-04-11', '2015-11-03']
}
```

```
employee_data = pd.DataFrame(data)
employee_data['Joining Date'] = pd.to_datetime(employee_data['Joining Date'])
print(employee_data)
highest_salaries = employee_data.groupby('Department')['Salary'].max()
lowest_salaries = employee_data.groupby('Department')['Salary'].min()
```

```
print("\nHighest salaries in each department:")
print(highest_salaries)
```

```
print("\nLowest salaries in each department:")
print(lowest_salaries)
current_date = pd.to_datetime('today')
employee_data['Tenure'] = (current_date - employee_data['Joining Date']).dt.days / 365.25
average_tenure = employee_data['Tenure'].mean()
```

```
print("\nAverage tenure of employees in the company:", average_tenure)
specific_date = pd.to_datetime('2017-01-01')
employees_before_date = employee_data[employee_data['Joining Date'] < specific_date]
```

```
print("\nEmployees who joined before January 1, 2017:")
print(employees_before_date)
```

→

	Employee ID	Department	Salary	Joining Date
0	101	HR	60000	2015-01-15
1	102	Finance	80000	2017-08-23
2	103	IT	90000	2016-06-30
3	104	HR	75000	2018-04-11
4	105	Finance	85000	2015-11-03

```
Highest salaries in each department:
Department
Finance    85000
HR         75000
IT         90000
Name: Salary, dtype: int64
```

```
Lowest salaries in each department:
Department
Finance    80000
HR         60000
IT         90000
Name: Salary, dtype: int64
```

```
Average tenure of employees in the company: 7.897604380561259
```

Employees who joined before January 1, 2017:

	Employee ID	Department	Salary	Joining Date	Tenure
0	101	HR	60000	2015-01-15	9.516769
2	103	IT	90000	2016-06-30	8.060233
4	105	Finance	85000	2015-11-03	8.717317