```
import random
import pandas as pd
```

```
def generate_covid_data(num_cases):
  return [
    {
      "fever": random.randint(95, 105), # fever: Temperature in Fahrenheit. Values range from 95 to 105.

      "cold": random.randint(0, 10), # cold: Severity of cold symptoms.
      # A value of 0 might represent no cold symptoms, while 10 could indicate severe cold.

      "shivering": random.randint(0, 5), # shivering: Frequency or intensity of shivering.
      # 0 might mean no shivering, while 10 could represent frequent or intense shivering.

      "weight_loss": random.randint(0, 10) # weight_loss: Amount of weight loss in kg.
    }
    for _ in range(num_cases)
  ]



```

```
def sort_data(data, parameter):
  return sorted(data, key=lambda x: x[parameter])
```

```
data = generate_covid_data(100)
df = pd.DataFrame(data)
```

```
df.head()
```

	fever	cold	shivering	weight_loss
0	97	10	0	4
1	103	10	1	8
2	102	3	2	3
3	102	3	1	6
4	103	7	5	8

Next steps: View recommended plots New interactive sheet

```
sorted_df = df.sort_values(by="fever")
print(sorted_df)
```

```
     fever  cold  shivering  weight_loss
28      95     8          3            1
61      95    10          4            9
95      95     9          0            0
60      95     2          3            4
89      95     6          1            7
..     ...   ...        ...          ...
90     105     0          4            5
76     105     8          3            3
31     105     6          5            2
10     105     2          3            0
62     105     2          0            4

[100 rows x 4 columns]
```

```
sorted_df = df.sort_values(by="weight_loss")
print(sorted_df)
```

```
     fever  cold  shivering  weight_loss
54     103     0          3            0
95      95     9          0            0
74     102     7          5            0
83     104     2          3            0
35      97     6          0            0
..     ...   ...        ...          ...
5       96     4          2           10
44      97     8          5           10
59     104     1          3           10
56     103     4          0           10
68     102     7          2           10

[100 rows x 4 columns]
```

```
a = input("enter a parameter to sort: ")
sorted_df = df.sort_values(by= a )
print(sorted_df)
```

```
enter a parameter to sort: cold
    fever  cold  shivering  weight_loss
66     96     0          4            8
65     97     0          4            3
54    103     0          3            0
87     97     0          5            3
90    105     0          4            5
..    ...   ...        ...          ...
27    104    10          2            4
61     95    10          4            9
39    100    10          4            4
84     98    10          5            5
0      97    10          0            4

[100 rows x 4 columns]
```

```
import numpy as np

def parse_equation(equation):
  return equation
```

```
def get_input_for_parameters(equation):

  params = [param for param in equation if param.isalpha()]  # Assuming parameters are single letters
  values = {param: float(input(f"Enter the value for {param}: ")) for param in params}
  return values
```

```
def evaluate_equation(equation, values):
  for param, value in values.items():
    equation = equation.replace(param, str(value))
  return eval(equation)
```

```
if __name__ == "__main__":
  equation = input("Enter the mathematical equation: ")
  values = get_input_for_parameters(equation)
  result = evaluate_equation(equation, values)
  print("Result:", result)
```

```
Enter the mathematical equation: a*2+b+c
Enter the value for a: 3
Enter the value for b: 4
Enter the value for c: 5
Result: 15.0
```

```

```