Fundamental Concepts in Data Insight:

Demo: Automating Insight

Fundamentals for a General Audience

What are algorithms?

An algorithm is *any* sequence of steps where the steps are drawn from a set of available operations.

```
define Algorithm A:
    require input NUMBER_OF_JUMPS

    repeat jump for NUMBER_OF_JUMPS
    turn
    sit
end
```

On digital computers algorithms are concerned with two types of operation:

- device-control
- calculation

How do you write an algorithm in python?

```
def algorithm_a(number_of_jumps):
    print( "JUMP " * number_of_jumps )
    print( "TURN" )
    print( "SIT" )
```

```
algorithm_a(3)
```

```
JUMP JUMP JUMP
TURN
SIT
```

```
algorithm_a(2)
```

```
JUMP JUMP
TURN
SIT
```

How do you write calculative algorithms?

```
def business_algorithm(input_data, business_decision):
    if business_decision:
        return 2_000 * input_data + 1
    else:
        return 3_000 * input_data + 1


x_age = 31
y_profit = business_algorithm(
            input_data = x_age,
            business_decision = True
        )

print(y_profit)
```

```
62001
```

How you access pre-defined algorithms?

```
import numpy as np # load numpy

x_age = np.random.normal(30, 5, 10_000).round() # simulate 10k ages
```

How do you combine business-specific and generic algorithms?

Below, `x_age` is generated using a library, `x_age = np.random.normal`, whereas we wrote `business_algorithm`,

```
y_profit = business_algorithm(x_age, False) # compute 10k predictio
ns
```

`y_profit` is a synthesis of generic library code and business-specific algorithms.

We can display both using another library,

```
import pandas as pd

pd.DataFrame({'Age': x_age, 'Profit': y_profit}).head()
```

	Age	Profit
0	22.0	66001.0
1	28.0	84001.0
2	35.0	105001.0
3	27.0	81001.0