Making a function in Python

```
def function_name(arguments):
    expression
    return variable_name


def diff_two_nums(x, z):
    y = x - z
    return y
```

Py

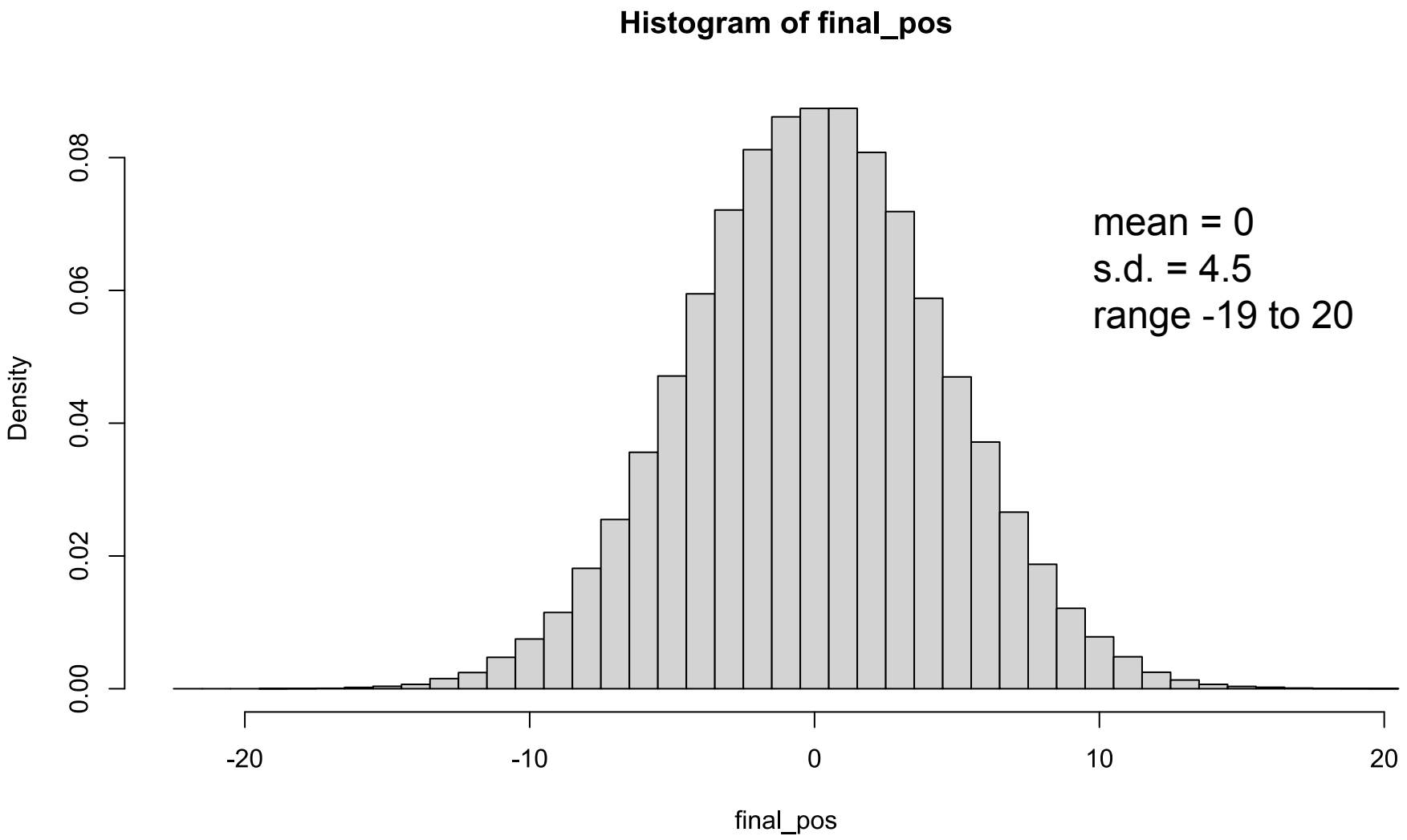
Vectorized programming

Demo in R

Data generating process

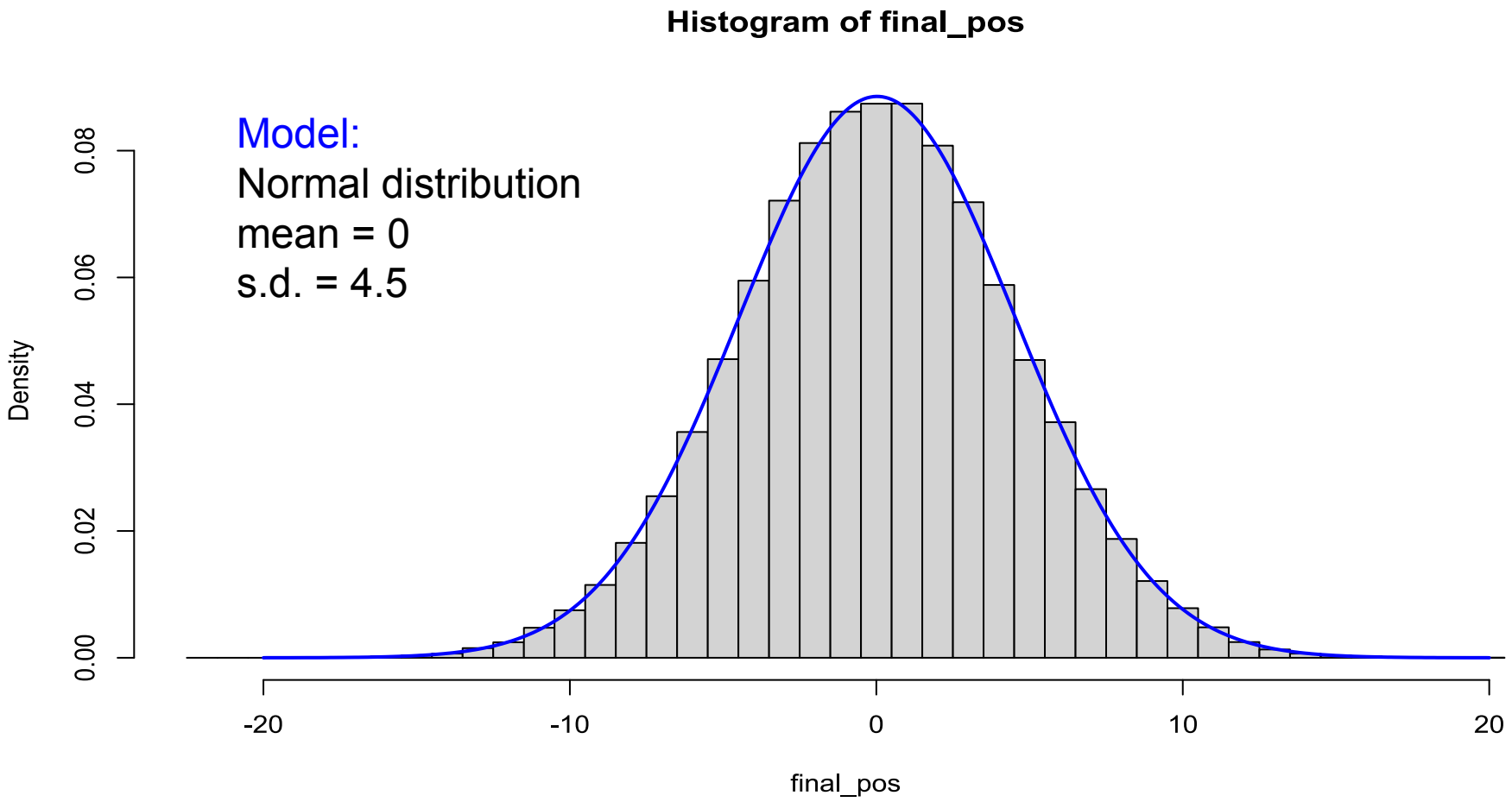
- We've seen biological models:
- Deterministic models (house finch)
- Stochastic DGP:
  - Movement, finding nut
  - Intrinsic stochastic process

Generated data



A new data generating model

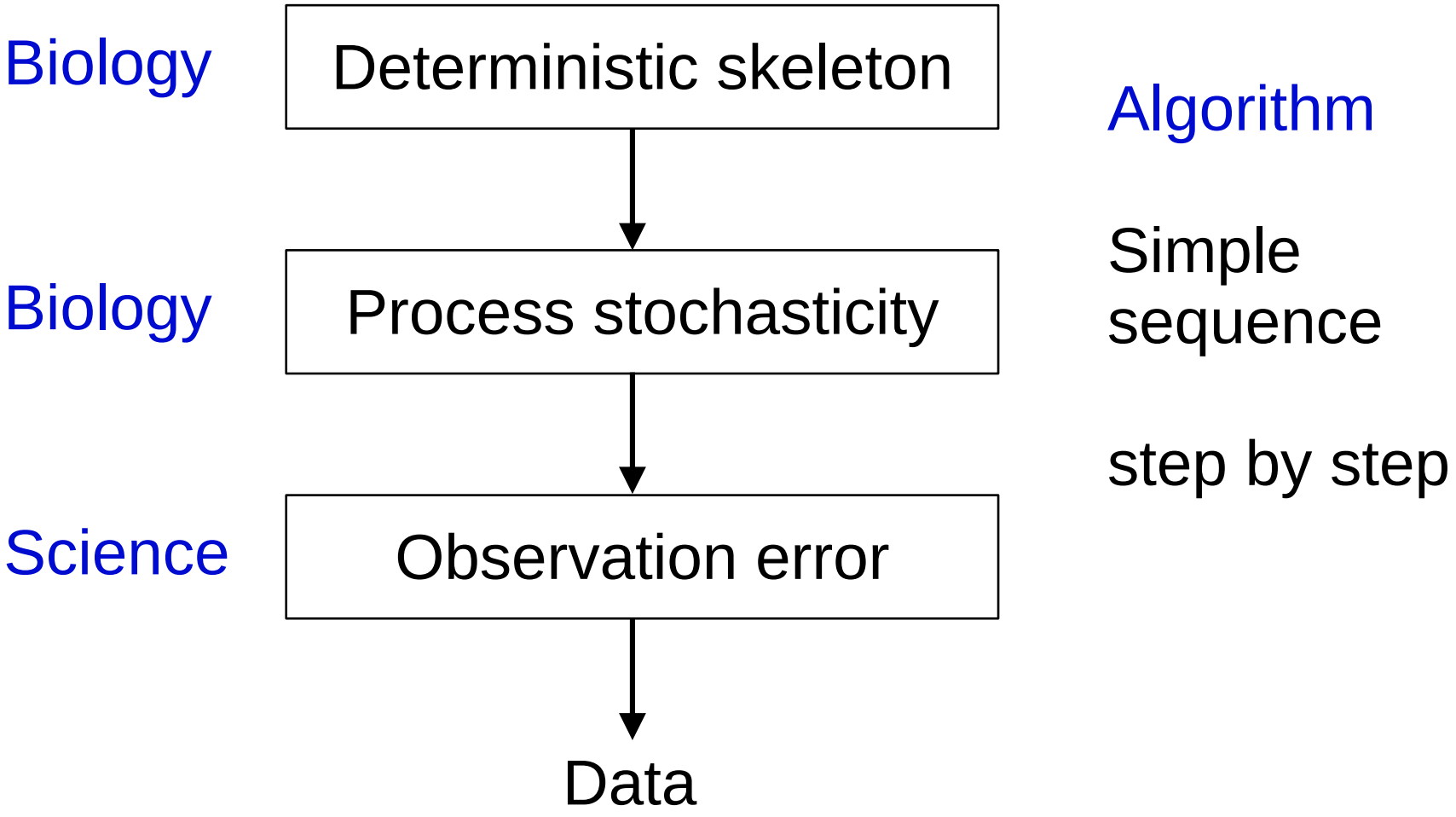
Phenomenological scale of abstraction



Data generating process

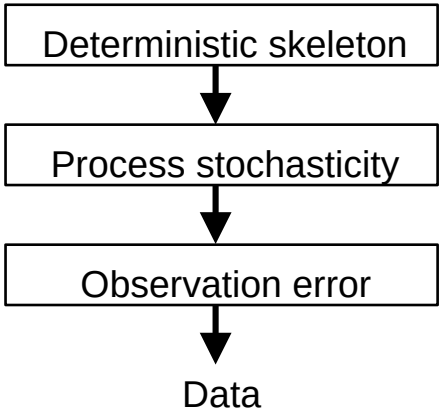
- We've seen biological models:
- Deterministic models (house finch)
- Biological stochastic DGP:
  - Movement, finding nut
  - Intrinsic, biological, stochastic process
- Next:
- Deterministic skeleton + stochasticity

DGP: Deterministic skeleton + stochasticity



DGP: Deterministic skeleton + stochasticity

- Skeleton can be biological model
  - e.g. house finch
  - birth & death processes
- Skeleton can be descriptive model
  - e.g. linear Normal model
  - relationships among variables
  - (emerge from biological processes)



Linear skeleton: function

```
function_name <- function(arguments) {
    expression
    return(object)
}
```

Exercise:
Make a function to calculate the linear deterministic skeleton given the model parameters and a vector of x data. In other words, turn the following into a function:

```
y <- b_0 + b_1 * x
```

Use vectorized operations

R

Linear skeleton: function

```
function_name <- function(arguments) {
    expression
    return(object)
}
```

Exercise:
Make a function to calculate the linear deterministic skeleton given the model parameters and a vector of x data. In other words, turn the following into a function:

```
y <- b_0 + b_1 * x
```

Solution:

```
lin_skel <- function(b_0, b_1, x) {
  y <- b_0 + b_1 * x
  return(y)
}
```

Use vectorized operations

R