

# Focus Boxes and Equations

This is a x- and y-centered **green focusbox** with 60% width and an unnumbered equation:

$$f(x) = x^2 + 2x + 1$$

We can add some vertical space between elements using the #v( ) command:

This is a **red focusbox** with a numbered equation and larger text.

$$\int_0^\infty e^{-x^2} dx = \frac{\sqrt{\pi}}{2} \quad (1)$$

# Focusbox configuration

The focusbox has several options:

- bg: - Background color (blue, red, green, cyan, magenta, yellow, gray, white)
- text-size: Font size (e.g., 0.8em, 1.2em)
- center\_x: Horizontal centering (true/false)
- center\_y: Vertical centering (true/false)
- width: Box width (e.g., 80%, 100%, auto)

We can also use the `#pause` marker to create animated subslides within a slide.

# Focusbox configuration

The focusbox has several options:

- `bg`: - Background color (blue, red, green, cyan, magenta, yellow, gray, white)
- `text-size`: Font size (e.g., 0.8em, 1.2em)
- `center_x`: Horizontal centering (true/false)
- `center_y`: Vertical centering (true/false)
- `width`: Box width (e.g., 80%, 100%, auto)

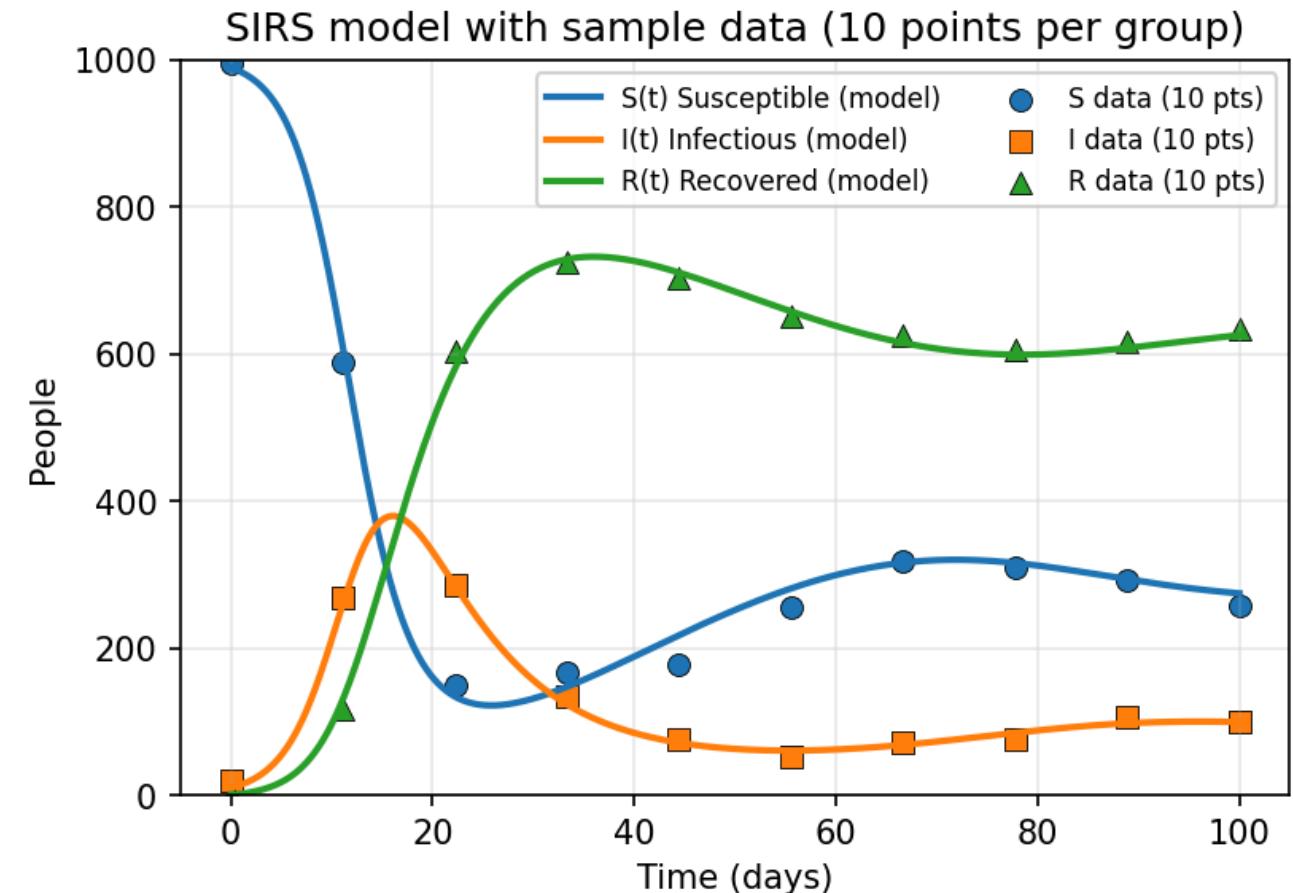
We can also use the `#pause` marker to create animated subslides within a slide.

The `#slide` function has these options:

- `headercolor`: Background color (blue, red, green, cyan, magenta, yellow, gray, white) - default: blue
- `title`: Slide title text - default: none
- `center_x`: Horizontal centering (true/false) - default: false
- `center_y`: Vertical centering (true/false) - default: true
- `slide-main-font`: Override main font for this slide only - default: none
- `slide-main-font-size`: Override main font size for this slide - default: none
- `slide-code-font`: Override code font for this slide - default: none
- `slide-code-font-size`: Override code font size for this slide - default: none
- `slide-equation-numbering`: Turn equation numbering on/off for this slide (auto/true/false) - default: auto
- `repeat`: Number of animation subslides (auto = auto-detect from `#pause` markers) - default: auto

# Data and Visualization

| Day | Susceptible S | Infectious I | Recovered R |
|-----|---------------|--------------|-------------|
| 0   | 990           | 10           | 0           |
| 2   | 950           | 35           | 15          |
| 4   | 880           | 75           | 45          |
| 6   | 780           | 120          | 100         |
| 8   | 650           | 155          | 195         |
| 10  | 520           | 175          | 305         |
| 12  | 400           | 180          | 420         |
| 14  | 300           | 165          | 535         |



## Python Code

We can embedd Python code for the simulation of an SIRS epidemiological model, using typst built-in code blocks:

```
import numpy as np

def sirs(N, beta, gamma, xi, I0, R0, days, dt):
    t = np.linspace(0, days, int(days/dt) + 1)
    S = np.zeros_like(t); I = np.zeros_like(t); R = np.zeros_like(t)
    S[0] = N - I0 - R0; I[0] = I0; R[0] = R0

    for k in range(len(t) - 1):
        dS = -beta*S[k]*I[k]/N + xi*R[k]
        dI = beta*S[k]*I[k]/N - gamma*I[k]
        dR = gamma*I[k] - xi*R[k]
        S[k+1] = S[k] + dt*dS
        I[k+1] = I[k] + dt*dI
        R[k+1] = R[k] + dt*dR

    return t, S, I, R
```