Reticulate Example

Caden Hewlett

2024-03-24

Example of using Python and `reticulate`, as well as `py2tex` to convert the output to natural latex.

```
from sympy import symbols, simplify
from pytexit import py2tex

# Define symbols
a = symbols('a')

# Original equation's rearrangement
left_side_coefficient = 1 - (1/(6*a))
right_side = (3 - 2*a) / (6*a)

# Simplify the equation for p(1)
rho_1 = simplify(right_side / left_side_coefficient)
# sympy.latex(eval(rho_1))
pytex_obj = py2tex(str(rho_1))
```

Then, in the `R` cell, by setting `results = 'asis'`, we render the equation directly.

```
cat(py$pytex_obj)
```

$$\frac{3-2a}{6a-1}$$

```
from sympy import symbols, solve, lambdify, diff

b0, b1, b2, x = symbols('b_0 b_1 b_2 x')

model_eq = b0 + b1*x + b2*x**2

roots = solve(model_eq, x)

root_funcs = [lambdify((b0, b1, b2), root) for root in roots]

## x-value where y = 0
b0_val, b1_val, b2_val = (1, 0.0102, -4.321e-5)
root_values = [root_func(b0_val, b1_val, b2_val) for root_func in root_funcs]
print([root for root in root_values if root > 0])
```

```
## [310.57291756492936]
## x-value where d/dx (y) = 0
y_prime = diff(model_eq, x)
## set to zero
extrema = solve(y_prime, x)
## transform to function
zeroes = lambdify((b1, b2), extrema)
print(zeroes(b1_val, b2_val))
```

```
## [118.02823420504514]
```