Reticulate and Sympy

2022-10-13

Disclaimer

Tests and experiments with `reticulate` and `sympy`

Basic Tests

From https://rstudio.github.io/reticulate/index.html, we check the basic functionality with the following code chunks

```
library(reticulate)
# use_python(Sys.which("python3"))
```

Simple usage for python

```
print("Hello Python!")
```

```
## Hello Python!
```

Installing pandas, to then be able to read data with pandas

```
# create virtual env
virtualenv_create("r-reticulate")
```

```
## virtualenv: r-reticulate
```

```
# install pandas
virtualenv_install("r-reticulate", "pandas")
```

```
## Using virtual environment ’r-reticulate’ ...
```

```
## + ’/Users/pvr/.virtualenvs/r-reticulate/bin/python’ -m pip install --upgrade --no-user ’pandas’
```

Use virtual env

```
use_virtualenv("r-reticulate")
```

Checking

```
py_available()
```

```
## [1] TRUE
```

Module pandas

```
py_module_available("pandas")
```

```
## [1] TRUE
```

Now use pandas to import data

```
import pandas
geno = pandas.read_csv("/Users/pvr/Data/Projects/Github/charlotte-ngs/lbgfs2022_gh-root/main/lbgfs2022/
print(geno.head())
```

```
##   ID;Genotype;Phenotype
## 0        46;2;-1.161803
## 1        47;4;-0.779888
## 2        48;2;+0.344549
## 3        49;4;-1.863295
## 4        50;2;+0.539282
```

Installation of ‘sympy’

The python package ‘sympy’ is installed with

```
virtualenv_install("r-reticulate", "sympy")
```

```
## Using virtual environment ’r-reticulate’ ...
```

```
## + ’/Users/pvr/.virtualenvs/r-reticulate/bin/python’ -m pip install --upgrade --no-user ’sympy’
```

Checking whether the module sympy was installed can be done with

```
py_module_available("sympy")
```

```
## [1] TRUE
```

Testing sympy

According to https://cerebralmastication.com/2020/07/27/using-python-sympy-in-a-python-code-chunk-in-rmarkdown/, the following test for sympy should work.

```
from sympy import *
x = symbols('x')
a = Integral(cos(x)*exp(x), x)
output = Eq(a, a.doit())
latex_output = latex(output)
print('$$' + latex_output + '$$\n')
```

$$\int e^{x} \cos{\left(x \right)}\, dx = \frac{e^{x} \sin{\left(x \right)}}{2} + \frac{e^{x} \cos{\left(x \right)}}{2}$$

An alternative way to show the output with an R-code chunk is

```
knitr::asis_output(sprintf("$$%s$$\n", py$latex_output))
```

$$\int e^{x} \cos{\left(x \right)}\, dx = \frac{e^{x} \sin{\left(x \right)}}{2} + \frac{e^{x} \cos{\left(x \right)}}{2}$$

Use sympy for Derivations in Quantitative Genetics

```
# import
from sympy import *
# symbol definition
a, d, p, q, mu, mu_res = symbols('a, d, p, q, mu, mu_res')
# allele frequencies
q = 1-p
# compute population mean
mu = a * p**2 + d * 2 * p * q + (-a) * q**2
mu_res = simplify(collect(expand(mu), [a, d]))
print('$$' + latex(mu) + '$$\n')
```

$$a p^{2} - a \left(1 - p\right)^{2} + 2 d p \left(1 - p\right)$$

```
print('$$' + latex(mu_res) + '$$\n')
```

$$a \left(2 p - 1\right) - 2 d p \left(p - 1\right)$$