

Module - Working with Python

LearningSpoonsR

2018-05-20

1. Python 설치 및 사용법

1-1. Anaconda 설치

Anaconda를 설치하는 것이 가장 쉽고 대중적인 python을 설치 방법입니다.

1-2. Python 사용

- ❶ Anaconda console: 패키지 인스톨 등의 작업에 필요한 커맨드 창입니다.
- ❷ Spyder: Rstudio라고 생각하시면 됩니다.
- ❸ Jupyter Notebook: Rmarkdown이라고 생각하시면 됩니다.

2. feather 패키지를 이용한 Python과 R의 통신

2-1. Python에서 Data.Frame을 만들어 feather object로 저장합니다.

- 아래와 같이 rmarkdown의 code chunk에서 설정해주면 파이썬 명령어가 실행가능합니다?!
- {python, engine.path = "C:/Anaconda3/python.exe", highlight=TRUE}
- Python에서 Data.Frame을 만들어 feather object로 저장합니다.

```
import pandas as pd
import numpy as np
import feather
print("Hello World from Python")
```

```
## Hello World from Python
```

```
my_numpy_obj = np.random.rand(10)
my_pandas_obj = pd.DataFrame(my_numpy_obj)
feather.write_dataframe(my_pandas_obj, "my_python_feather")
```

- my_python_feather는 이제 파이썬의 pandas.DataFrame object를 저장하고 있습니다. pandas.DataFrame은 R의 data.frame에 대응되는 파이썬의 자료 구조이며, R에서 motivation을 받아 만들어졌다고 합니다.

2-2. R에서 my_python_feather를 import 합니다.

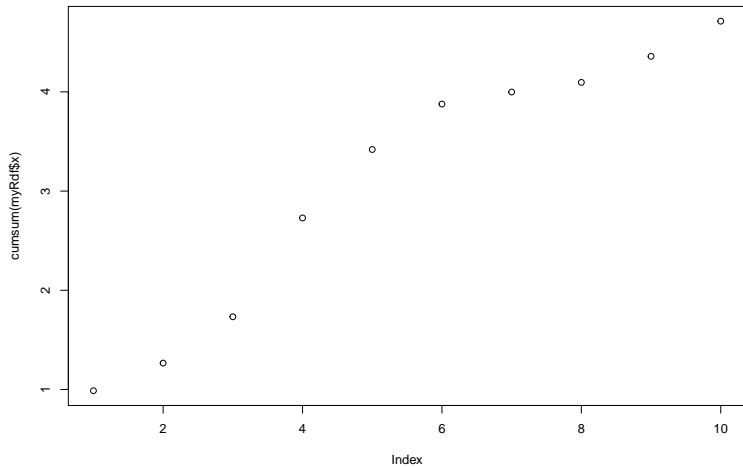
```
library(feather)
myRFeature <- read_feather("my_python_feather")
myRdf      <- data.frame(x = unlist(myRFeature))
class(myRdf)
```

```
## [1] "data.frame"
```

```
head(myRdf)
```

```
##           x
## 01 0.9879157
## 02 0.2781795
## 03 0.4668128
## 04 0.9962832
## 05 0.6897997
## 06 0.4581112
```

```
plot(cumsum(myRdf$x))
```



2-3. 결론

- Rmarkdown 환경에서도 파이썬 작업을 할 수 있습니다.
- Python에서만 작업이 편하거나 R에서 작업이 편한 경우에는 편한 곳에서 작업하면서 을 주고 받으며 작업할 수 있습니다.
- 그림을 그리거나 문서화 하는 경우는 R이 파워풀하므로, Python에서 데이터 프레임을 불러오는 예제로 만들었습니다. (반대의 경우도 가능합니다.)
- 예를 들어, Python에서 데이터를 전처리 한 이후에 R에서 불러와서 ggplot을 이용해서 그래프를 그리고 마크다운을 이용해서 docx, html, flexdashboard, shiny, pdf등을 렌더할 수 있습니다.
- R을 배우신 이후에 Python을 배우실 예정이시면 기초적인 부분을 먼저 공부하신 후에, “Python for R user”와 같은 검색어로 메뉴얼을 검색하시면 문법을 금방 배우실 수 있습니다.
- 언어의 경계를 없애는 일은 작업의 편의성을 높임과 함께, 함께 일하는 사람들과의 커뮤니케이션에서도 무척 유용합니다. 마치 회사에 한국어와 영어를 하는 사람이 있다면, 두 언어를 다 사용하는 사람의 가치가 높아지는 것과 같은 현상입니다.