Using reticulate to access CPTAC data in R

`reticulate` is an R package that allows you to create Python objects and convert them into R objects for use in your R environment.

Thus, even though `cptac` is a Python package, with the help of `reticulate` you can load and work with the datasets in R.

Step 1: Environment setup

First, you need to load the `reticulate` package, and then tell it which Python environment you want to use to access the `cptac` package.

```r
# Install reticulate if necessary
if (!require(reticulate)) install.packages("reticulate")

# Load the package
library(reticulate)

# Specify to use the environment where the cptac package is installed. Replace
# "dev" with the name of your environment. If you use an environment manager
# besides conda, you'll want this command: use_virtualenv("myenv")
use_condaenv("dev", required = TRUE)
```

Step 2: Load cptac and access data

Now we will load `cptac` and get the dataset we want.

```r
# Import the package. We pass convert = FALSE so that objects won't be
# converted from Python to R until we explicitly ask for it. This is necessary
# to properly prepare a multiindex dataframe for conversion to R (see below).
cptac <- import("cptac", convert = FALSE)

# Load the dataset
en <- cptac$Endometrial()
```

Load a dataframe and convert it to an R object

This dataframe just has a regular column index (not a multiindex), so we can directly convert it into an R object after loading it.

When `reticulate` converts a `pandas.DataFrame` into R, it converts it to the R `data.frame` type.

```r
# Load the table
prot_py <- en$get_proteomics()

# Convert into R
prot <- py_to_r(prot_py)

print(prot[1:10, 1:8])
```

```
##                A1BG     A2M  A2ML1  A4GALT    AAAS    AACS   AADAT   AAED1
## C3L-00006 -1.180 -0.8630 -0.802  0.2220  0.2560  0.6650  1.2800 -0.3390
## C3L-00008 -0.685 -1.0700 -0.684  0.9840  0.1350  0.3340  1.3000  0.1390
## C3L-00032 -0.528 -1.3200  0.435     NaN -0.2400  1.0400 -0.0213 -0.0479
## C3L-00090 -1.670 -1.1900 -0.443  0.2430 -0.0993  0.7570  0.7400 -0.9290
## C3L-00098 -0.374 -0.0206 -0.537  0.3110  0.3750  0.0131 -1.1000     NaN
## C3L-00136 -1.080 -0.7080 -0.126 -0.4260 -0.1140 -0.1110  0.8950  1.2600
## C3L-00137 -1.320 -0.7080 -0.808 -0.0709  0.1380  0.6560 -0.2800 -0.1280
## C3L-00139 -0.467  0.3700 -0.339     NaN  0.4340  0.0358 -0.1750  0.1810
## C3L-00143 -1.120 -1.3100  0.912  0.4180 -0.0768  0.8460 -0.1210     NaN
## C3L-00145 -0.716 -0.8850  2.820 -0.3430  0.1470  0.4450 -0.0565 -0.8380
```

Load a multiindex dataframe

If it's a dataframe with a column multiindex, you need to first flatten the index before converting it to an R object.

See tutorial 4 for more information on column multiindexes.

```
# Load the table
phos_py_multiindex <- en$get_phosphoproteomics()

# Load cptac.utils so we can access a helper function to convert the multiindex
# to a single level index.
utils <- import("cptac.utils", convert = FALSE)

# Convert the multiindex to a single level index. This works because we passed
# convert = FALSE when we imported the cptac package, so the dataframe is still
# a Python object at this point (it  wasn't automatically converted into an R
# object on loading) and can be passed to this Python function.
phos_py_single_index <- utils$reduce_multiindex(phos_py_multiindex, flatten = TRUE)

# Convert to R
phos <- py_to_r(phos_py_single_index)

print(phos[1:10, 1:6])
```

```
##           AAAS_S495 AAAS_S541 AAAS_Y485 AACS_S618 AAED1_S12 AAGAB_S310
## C3L-00006       NaN       NaN       NaN    -0.881   -1.8100        NaN
## C3L-00008       NaN       NaN       NaN       NaN    0.0840        NaN
## C3L-00032    -0.202       NaN       NaN       NaN   -1.8800        NaN
## C3L-00090    -0.002       NaN    -0.407       NaN       NaN        NaN
## C3L-00098     0.556   -0.0461       NaN       NaN    0.9410        NaN
## C3L-00136       NaN       NaN       NaN       NaN    0.0796        NaN
## C3L-00137     0.300       NaN       NaN    -0.371   -1.1400        NaN
## C3L-00139     0.490       NaN       NaN       NaN       NaN        NaN
## C3L-00143    -0.231    0.0108       NaN       NaN    0.2420        NaN
## C3L-00145     0.268       NaN       NaN       NaN   -0.1120        NaN
```