Let the fusion begin. I will construct a pandas DataFrame from dictionary. So we will work in R towards Python dictionary.

Iris dataset is the best example for this transition. Small but very useful.

Presume, you have a data.frame in R. In this case Iris dataset (complete or just first 15 rows, for sake of brevity of this useless function):

```
iris <- data.frame(iris)
#iris <- data.frame(iris[1:15,])
```

Once we have a data.frame we want to generate Python dictionary that will hold schema and data for direct creation (or import) into your favourite Python environment. Function RtoPy does the needed transformation:

```
RtoPy <- function(df_input, filename_path) {

    # column names and number of rows
    Nn <- names(df_input)
    nr <- nrow(df_input)

#Python is Indentation  sensitive - leave these two lines without
indentation
py_df <- "import pandas as pd
d = {"

    for (x in 1:length(Nn)){
      var <- (Nn[x])
      #Column Names
      py_df <- paste0(py_df, "'",var,"':[", collapse=NULL)

      #Data Rows
      for (i in 1:nr) {
        val <- df_input[i,x]
        #Check for data type
        if (sapply(df_input[i,x], class) == "factor") {
          py_df <- paste0(py_df, "'",val,"'", ",", collapse=NULL)
          #for last value in a column
          if (i == nr){
            py_df <- paste0(py_df, "'",val,"'", "],","\n",
collapse=NULL)
          }
        } else {
          py_df <- paste0(py_df, val, ",", collapse=NULL)
          #for last value in a column
          if (i == nr){
            py_df <- paste0(py_df, val, "],","\n", collapse=NULL)
          }
        }

      }
      if (x == length(Nn)){
        py_df <- substr(py_df, 1, nchar(py_df)-2)
        py_df <- paste0(py_df, "}
df=pd.DataFrame(data=d)", collapse=NULL)
      }
    }

    ## Store to file
    sink(file = filename_path)
    cat(py_df)
    sink(file = NULL)
}
```

The input parameters are:
– data.frame in R that you want to have it scripted in python
– filename to store the schema and data

```
# Get the data from R data.frame to Python Pandas script
iris <- data.frame(iris)
RtoPy(iris, "/users/tomazkastrun/desktop/iris_py.py")
```

And the python code for creating this data.frame in pandas is:

```
'''python
import pandas as pd
d = {'Sepal.Length':[5.1,4.9,4.7,4.6,5,5.4,4.6,5,4.4,4.9,5.4,4.
8,4.8,4.3,5.8,5.8],
'Sepal.Width':[3.5,3,3.2,3.1,3.6,3.9,3.4,3.4,2.9,3.1,3.7,3.4,3,3,4,4],
'Petal.Length':[1.4,1.4,1.3,1.5,1.4,1.7,1.4,1.5,1.4,1.5,1.5,
1.6,1.4,1.1,1.2,1.2],
'Petal.Width':[0.2,0.2,0.2,0.2,0.2,0.4,0.3,0.2,0.2,0.1,0.2,
0.2,0.1,0.1,0.2,0.2],
'Species':['setosa','setosa','setosa','setosa','setosa','
setosa','setosa','setosa','setosa','setosa','setosa','
setosa','setosa','setosa','setosa','setosa']}
df=pd.DataFrame(data=d)
```

Since Python is indentation sensitive, storing the schema and data to file turned out to be safest way. And don’t ask why not use CSV to do the transformation from one to another language. 🙂