In R, there are a couple ways to convert the column-oriented data frame to a row-oriented dictionary list or alike, e.g. a list of lists.

In the code snippet below, I would show each approach and how to extract keys and values from the dictionary. As shown in the benchmark, it appears that the generic R data structure is still the most efficient.

### LIST() FUNCTION IN BASE PACKAGE ###

x1 <- as.list(iris[1, ])

names(x1)

# [1] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width" "Species"

x1[["Sepal.Length"]]

# [1] 5.1

### ENVIRONMENT-BASED SOLUTION ###

envn\_dict <- function(x) {

e <- new.env(hash = TRUE)

for (name in names(x)) assign(name, x[, name], e)

return(e)

}

x2 <- envn\_dict(iris[1, ])

ls(x2)

# [1] "Petal.Length" "Petal.Width" "Sepal.Length" "Sepal.Width" "Species"

x2[["Sepal.Length"]]

# [1] 5.1

### COLLECTIONS PACKAGE ###

coll\_dict <- function(x) {

d <- collections::Dict$new()

for (name in names(x)) d$set(name, x[, name])

return(d)

}

x3 <- coll\_dict(iris[1, ])

x3$keys()

# [1] "Petal.Length" "Petal.Width" "Sepal.Length" "Sepal.Width" "Species"

x3$get("Sepal.Length")

# [1] 5.1

### HASH PACKAGE ###

hash\_dict <- function(x) {

d <- hash::hash()

for (name in names(x)) d[[name]] <- x[, name]

return(d)

}

x4 <- hash\_dict(iris[1, ])

hash::keys(x4)

# [1] "Petal.Length" "Petal.Width" "Sepal.Length" "Sepal.Width" "Species"

hash::values(x4, "Sepal.Length")

# Sepal.Length

# 5.1

### DATASTRUCTURES PACKAGE ###

data\_dict <- function(x) {

d <- datastructures::hashmap()

for (name in names(x)) d[name] <- x[, name]

return(d)

}

x5 <- data\_dict(iris[1, ])

datastructures::keys(x5)

# [1] "Species" "Sepal.Width" "Petal.Length" "Sepal.Length" "Petal.Width"

datastructures::get(x5, "Sepal.Length")

# [1] 5.1

### FROM PYTHON ###

py2r\_dict <- function(x) {

return(reticulate::py\_dict(names(x), x, TRUE))

}

x6 <- py2r\_dict(iris[1, ])

x6$keys()

# [1] "Petal.Length" "Sepal.Length" "Petal.Width" "Sepal.Width" "Species"

x6["Sepal.Length"]

# [1] 5.1

### CONVERT DATAFRAME TO DICTIONARY LIST ###

to\_list <- function(df, fn) {

l <- list()

for (i in seq(nrow(df))) l[[i]] <- fn(df[i, ])

return(l)

}

rbenchmark::benchmark(replications = 100, order = "elapsed", relative = "elapsed",

columns = c("test", "replications", "elapsed", "relative", "user.self", "sys.self"),

"BASE::LIST" = to\_list(iris, as.list),

"BASE::ENVIRONMENT" = to\_list(iris, envn\_dict),

"COLLECTIONS::DICT" = to\_list(iris, coll\_dict),

"HASH::HASH" = to\_list(iris, hash\_dict),

"DATASTRUCTURES::HASHMAP" = to\_list(iris, data\_dict),

"RETICULATE::PY\_DICT" = to\_list(iris, py2r\_dict)

)

# test replications elapsed relative user.self sys.self

#1 BASE::LIST 100 0.857 1.000 0.857 0.000

#2 BASE::ENVIRONMENT 100 1.607 1.875 1.607 0.000

#4 HASH::HASH 100 2.600 3.034 2.600 0.000

#3 COLLECTIONS::DICT 100 2.956 3.449 2.956 0.000

#5 DATASTRUCTURES::HASHMAP 100 16.070 18.751 16.071 0.000

#6 RETICULATE::PY\_DICT 100 18.030 21.039 18.023 0.008