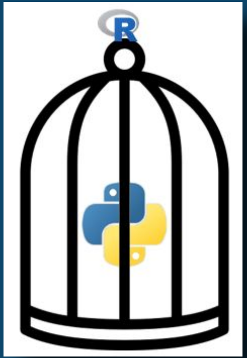
How to lock a Python in a cage?

Managing Python environment inside an R project



PyData meetup

Warsaw, 2018-02-06

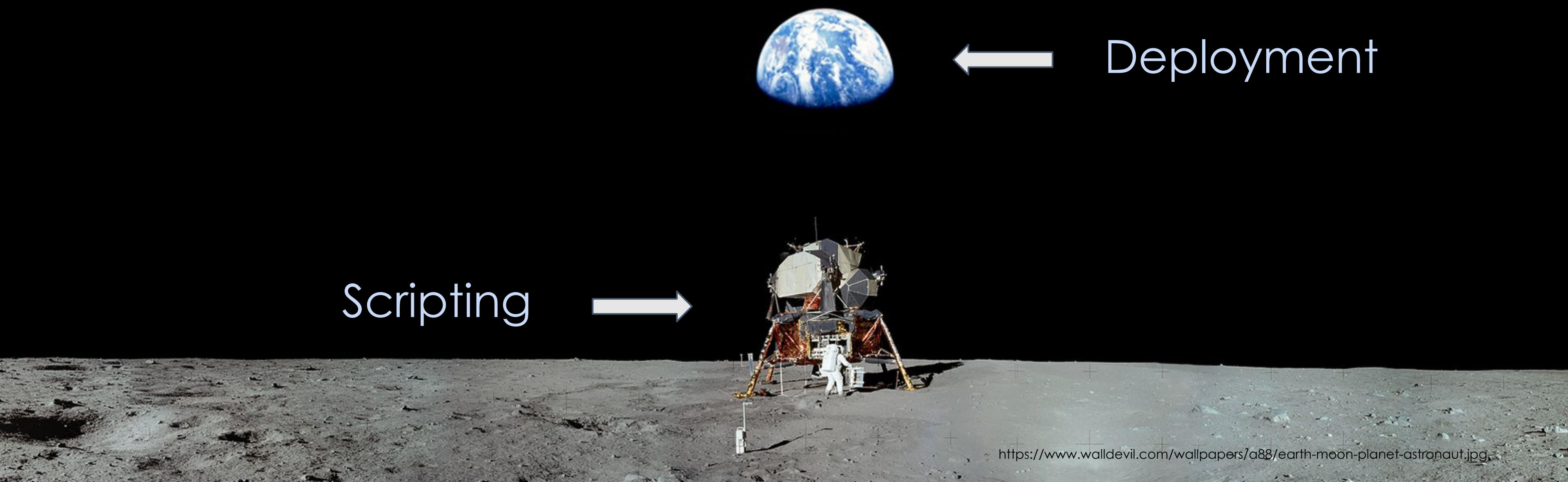
Piotr Chaberski, WLOG Solutions

A brief intro...

Imagine that you are developing a project using R and your big corporate customer, after weeks of processing requests to establish open-source analytical environment, finally managed to install R on their production machines. Now you realized, that it would be nice to use some Python library in your solution...

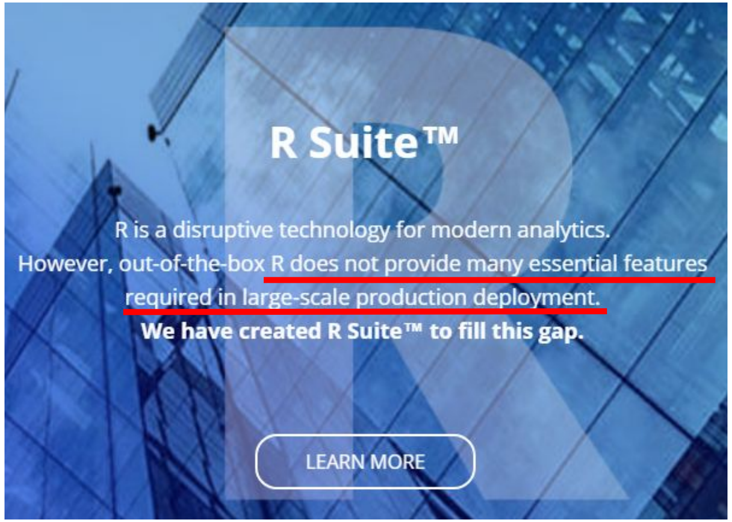
How would you tell the client to switch to Python for a while?

Both R and Python are great for writing scripts, but...



https://www.walldevil.com/wallpapers/a88/earth-moon-planet-astronaut.jpg

Why R Suite?



R Suite™

R is a disruptive technology for modern analytics.
However, out-of-the-box R does not provide many essential features required in large-scale production deployment.
We have created R Suite™ to fill this gap.

… so doesn't Python, to be clear

- develop your solution maintaining **full control over dependencies and their versions**
- build the solution for a **desired production environment**
- close your solution in **ready-to-deploy package** that doesn't require any installations or configuration

Why Python in R?

Why not only Python? Why not only R?

Your client has **R production environment**

You have well managed **R deployment procedure** (using R Suite)

You'd like to create a **Shiny app** on top of your solution

You just know/like R better

… but you still need some specific Python functionalities

So how to use Python when you have R on production?

… just hide it deep down in your R Suite project

Deep learning using Keras/Tensorflow in R

Use case with focus on deployment

Prerequisites

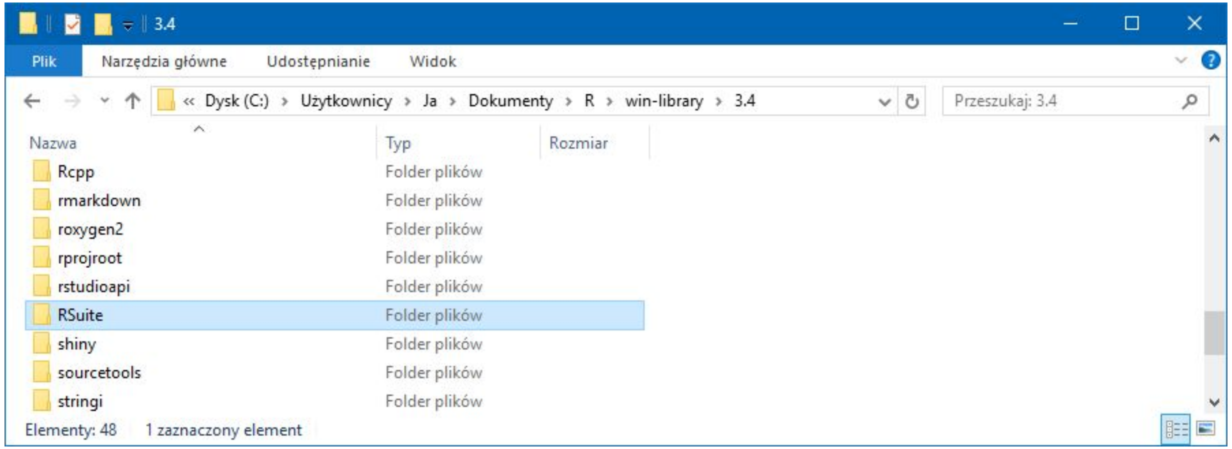
- R
- R Suite CLI
- RStudio (optional but recommended)
- Miniconda

Example created on Windows, but after prerequisites are installed for specific system, further steps are identical on Linux

Windows ubuntu debian redhat CentOS

1. Install R Suite R package

```
C:\Users\Ja>rsuite install
Detecting repositories ...
Will use repositories:
              CRAN = https://cloud.r-project.org/
             Other = http://wlog-rsuite.s3.amazonaws.com
Installing RSuite(v0.22-231) package ...
installing the source package 'RSuite'

All done.
```

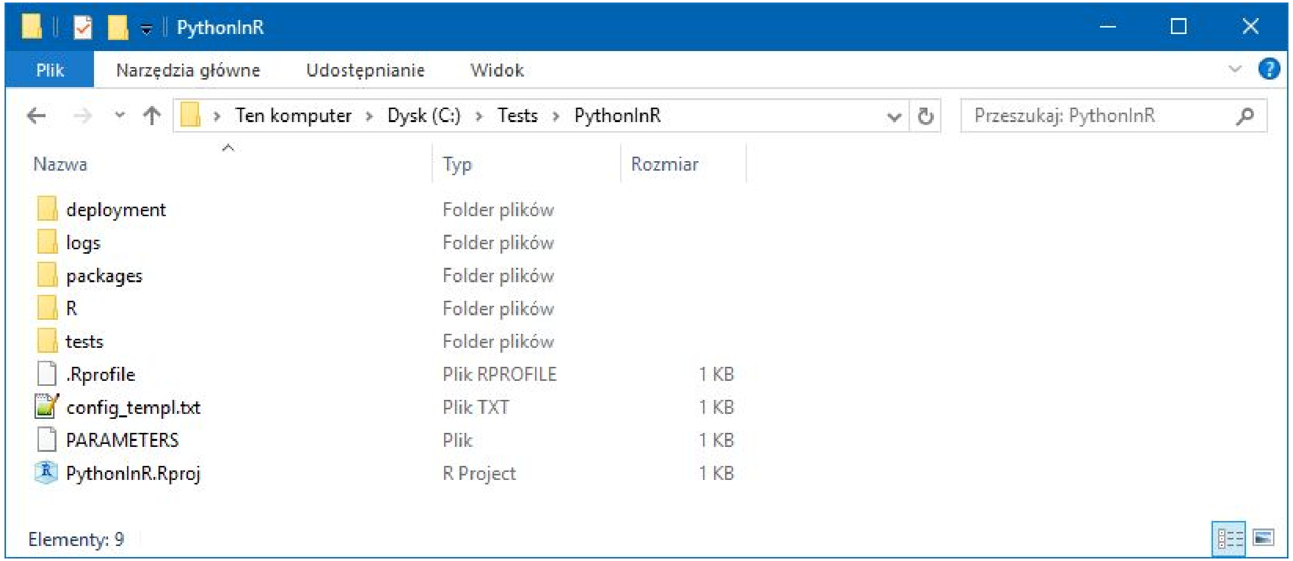

3.4
Plik
Narzędzia główne
Udostępnianie
Widok
« Dysk (C:) › Użytkownicy › Ja › Dokumenty › R › win-library › 3.4
Przeszukaj: 3.4

Nazwa	Typ	Rozmiar
Rcpp	Folder plików	
rmarkdown	Folder plików	
roxygen2	Folder plików	
rprojroot	Folder plików	
rstudioapi	Folder plików	
RSuite	Folder plików	
shiny	Folder plików	
sourcetools	Folder plików	
stringi	Folder plików	

Elementy: 48   1 zaznaczony element

2. Create R Suite project

```
C:\Tests>rsuite proj start -n PythonInR
2018-02-01 20:18:33 INFO:rsuite:Will create project PythonInR structure for RSuite v0.22.231.
2018-02-01 20:18:34 INFO:rsuite:Project PythonInR started.
```



Nazwa	Typ	Rozmiar
deployment	Folder plików	
logs	Folder plików	
packages	Folder plików	
R	Folder plików	
tests	Folder plików	
.Rprofile	Plik RPROFILE	1 KB
config_templ.txt	Plik TXT	1 KB
PARAMETERS	Plik	1 KB
PythonInR.Rproj	R Project	1 KB

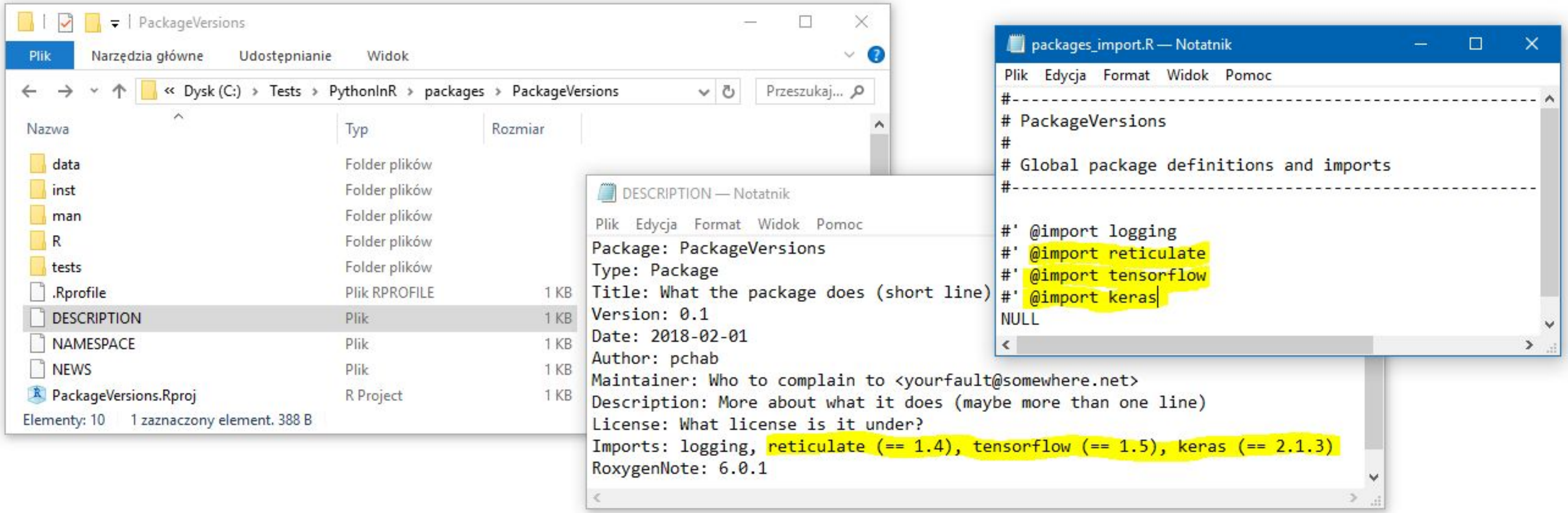
Elementy: 9

3. Create a dummy custom R package PackgageVersions

The solution is simple and there is no need to enclose custom functions in packages, however a dummy package will allow to specify external packages which might be useful when recreating dev environment. If not specified, the newest packages available in given set of repositories (PARAMETERS -> Repositories) will be installed as per library() calls in scripts.

```
C:\Tests\PythonInR>rsuite proj pkgadd -n PackageVersions
2018-02-01 20:49:51 INFO:rsuite:Package PackageVersions started in project PythonInR.
```

Edit DESCRIPTION file to contain all needed external packages. Also, list all imported packages in 'R/packages_import.R' file so they will be added to the PackageVersions' NAMESPACE.


PackageVersions
Plik
Narzędzia główne
Udostępnianie
Widok
« Dysk (C:) › Tests › PythonInR › packages › PackageVersions
Przeszukaj...

Nazwa	Typ	Rozmiar
data	Folder plików	
inst	Folder plików	
man	Folder plików	
R	Folder plików	
tests	Folder plików	
.Rprofile	Plik RPROFILE	1 KB
DESCRIPTION	Plik	1 KB
NAMESPACE	Plik	1 KB
NEWS	Plik	1 KB
PackageVersions.Rproj	R Project	1 KB

Elementy: 10 1 zaznaczony element. 388 B

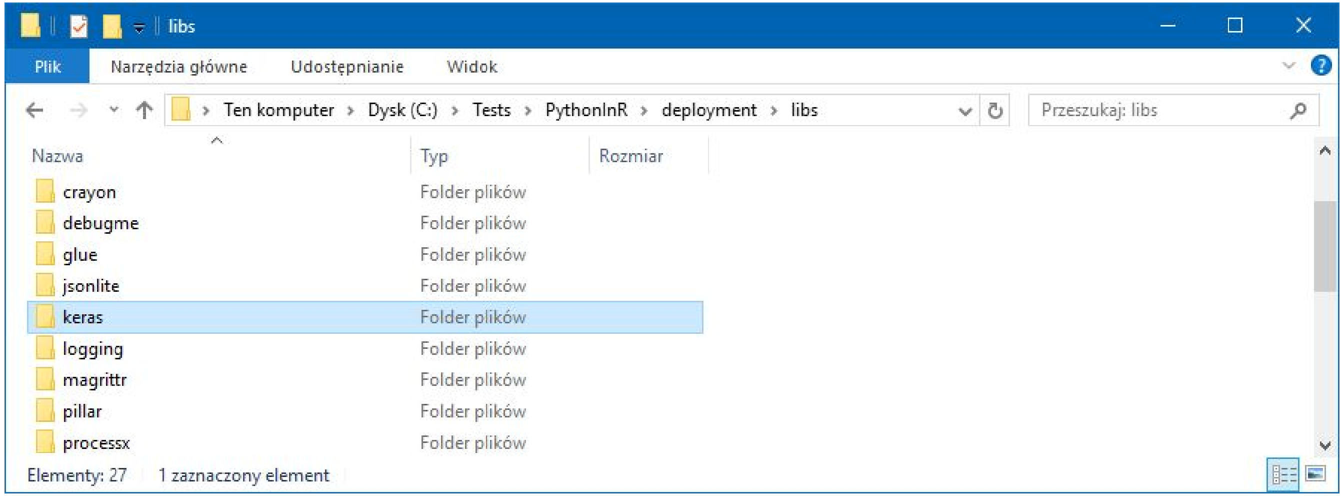
DESCRIPTION — Notatnik
Plik Edycja Format Widok Pomoc

```
Package: PackageVersions
Type: Package
Title: What the package does (short line)
Version: 0.1
Date: 2018-02-01
Author: pchab
Maintainer: Who to complain to <yourfault@somewhere.net>
Description: More about what it does (maybe more than one line)
License: What license is it under?
Imports: logging, reticulate (== 1.4), tensorflow (== 1.5), keras (== 2.1.3)
RoxygenNote: 6.0.1
```

```
#----------------------------------------------------------
# PackageVersions
#
# Global package definitions and imports
#----------------------------------------------------------

#' @import logging
#' @import reticulate
#' @import tensorflow
#' @import keras
NULL
```

4. Install external R packages to your local R Suite project

```
C:\Tests\PythonInR>rsuite proj depsinst
2018-02-01 21:08:10 INFO:rsuite:Detecting repositories (for R 3.4)...
2018-02-01 21:08:11 INFO:rsuite:Will look for dependencies in ...
2018-02-01 21:08:11 INFO:rsuite:.         MRAN#1 = https://mran.microsoft.com/snapshot/2018-02-01 (win.binary, sourc
2018-02-01 21:08:11 INFO:rsuite:Collecting project dependencies (for R 3.4)...
2018-02-01 21:08:11 INFO:rsuite:Resolving dependencies (for R 3.4)...
2018-02-01 21:08:14 INFO:rsuite:Detected 26 dependencies to install. Installing...
2018-02-01 21:14:02 INFO:rsuite:All dependencies successfully installed.
```

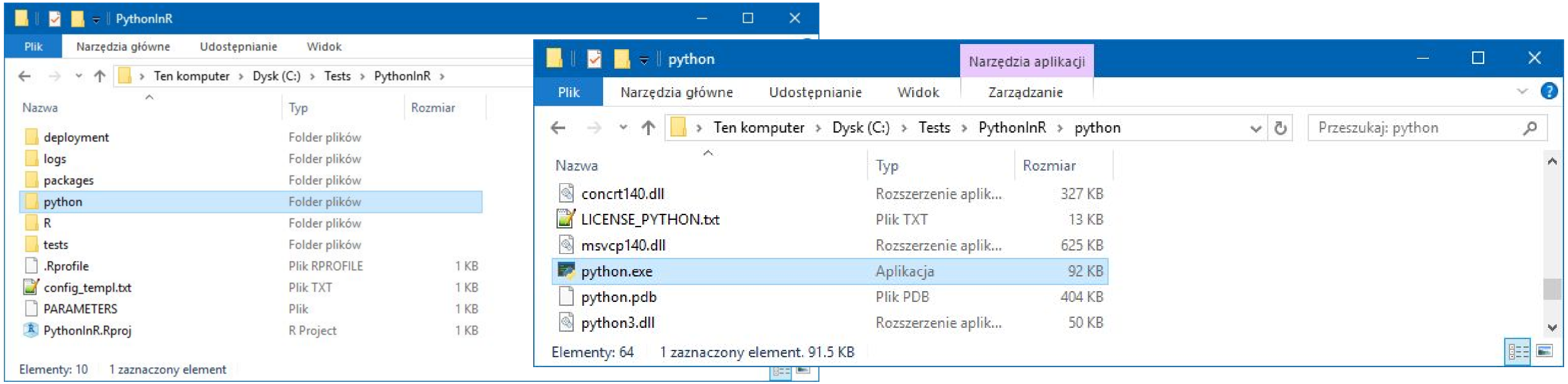

libs
Plik
Narzędzia główne
Udostępnianie
Widok
Ten komputer > Dysk (C:) > Tests > PythonInR > deployment > libs
Przeszukaj: libs

Nazwa	Typ	Rozmiar
crayon	Folder plików	
debugme	Folder plików	
glue	Folder plików	
jsonlite	Folder plików	
keras	Folder plików	
logging	Folder plików	
magrittr	Folder plików	
pillar	Folder plików	
processx	Folder plików	

Elementy: 27   1 zaznaczony element

5. Create local Python environment within R Suite project

```
C:\Tests\PythonInR>conda create -p python python=3.6.3
Fetching package metadata .............
Solving package specifications: .

Package plan for installation in environment C:\Tests\PythonInR\python:

The following NEW packages will be INSTALLED:

    certifi:        2018.1.18-py36_0
    pip:            9.0.1-py36h226ae91_4
    python:         3.6.3-h3b118a2_4
 ...
```



Nazwa	Typ	Rozmiar
deployment	Folder plików	
logs	Folder plików	
packages	Folder plików	
python	Folder plików	
R	Folder plików	
tests	Folder plików	
.Rprofile	Plik RPROFILE	1 KB
config_templ.txt	Plik TXT	1 KB
PARAMETERS	Plik	1 KB
PythonInR.Rproj	R Project	1 KB

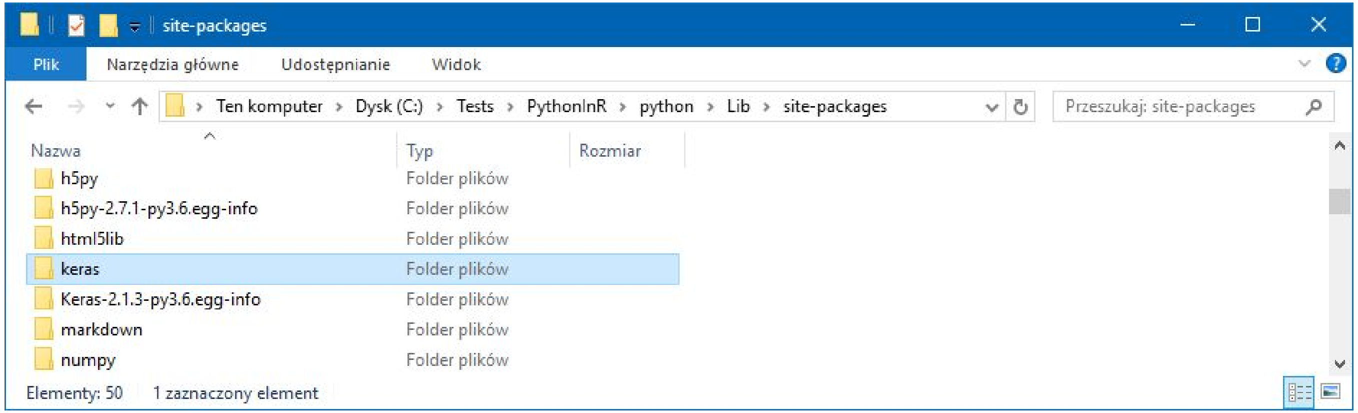
Elementy: 10 1 zaznaczony element

Nazwa	Typ	Rozmiar
concrt140.dll	Rozszerzenie aplik...	327 KB
LICENSE_PYTHON.txt	Plik TXT	13 KB
msvcp140.dll	Rozszerzenie aplik...	625 KB
python.exe	Aplikacja	92 KB
python.pdb	Plik PDB	404 KB
python3.dll	Rozszerzenie aplik...	50 KB

Elementy: 64 1 zaznaczony element. 91.5 KB

6. Install all needed Python packages with dependecies

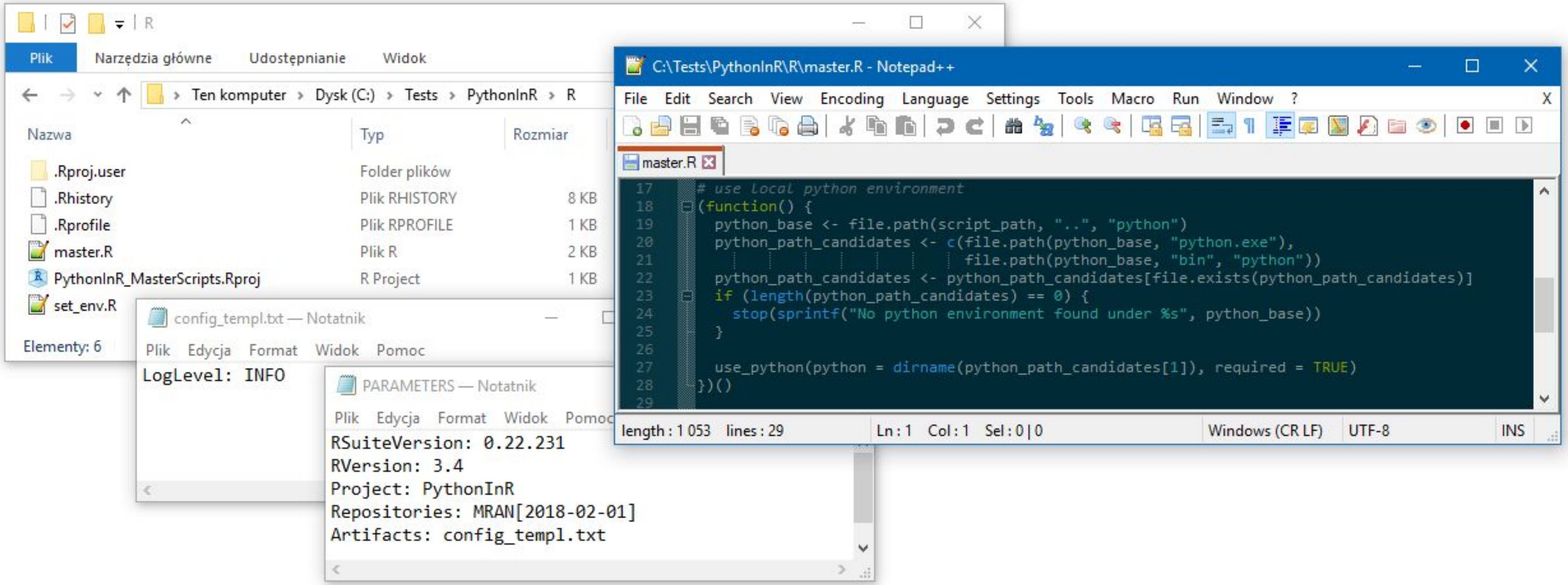
```
C:\Tests\PythonInR>conda install -n python tensorflow=1.2.1 keras=2.1.3 opencv=3.3.1 scikit-learn=0.19.1 pillow=5.0.0
Fetching package metadata .............
Solving package specifications: .

Package plan for installation in environment C:\Tests\PythonInR\python:

The following NEW packages will be INSTALLED:

    backports:          1.0-py36h81696a8_1
    backports.weakref: 1.0rc1-py36_0
...
    keras:              2.1.3-py36_0
...
```


site-packages
Plik
Narzędzia główne
Udostępnianie
Widok
Ten komputer › Dysk (C:) › Tests › PythonInR › python › Lib › site-packages
Przeszukaj: site-packages

```
Nazwa                          Typ             Rozmiar
h5py                           Folder plików
h5py-2.7.1-py3.6.egg-info      Folder plików
html5lib                       Folder plików
keras                          Folder plików
Keras-2.1.3-py3.6.egg-info     Folder plików
markdown                       Folder plików
numpy                          Folder plików
Elementy: 50    1 zaznaczony element
```

7. Develop the solution

As the R project and internal Python environment are configured, the project is ready to be devoloped. R master executable scripts can be modified, some custom R packages can be created and the solution can utilize installed R packages as well as the Python ones (using dedicated R interfaces such as 'reticulate').



R
Plik
Narzędzia główne
Udostępnianie
Widok
Ten komputer › Dysk (C:) › Tests › PythonInR › R
Elementy: 6
C:\Tests\PythonInR\R\master.R - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Window ?
master.R
config_templ.txt — Notatnik
PARAMETERS — Notatnik
Plik Edycja Format Widok Pomoc

Nazwa	Typ	Rozmiar
.Rproj.user	Folder plików	
.Rhistory	Plik RHISTORY	8 KB
.Rprofile	Plik RPROFILE	1 KB
master.R	Plik R	2 KB
PythonInR_MasterScripts.Rproj	R Project	1 KB
set_env.R		

```r
# use local python environment
(function() {
  python_base <- file.path(script_path, "..", "python")
  python_path_candidates <- c(file.path(python_base, "python.exe"),
                              file.path(python_base, "bin", "python"))
  python_path_candidates <- python_path_candidates[file.exists(python_path_candidates)]
  if (length(python_path_candidates) == 0) {
    stop(sprintf("No python environment found under %s", python_base))
  }

  use_python(python = dirname(python_path_candidates[1]), required = TRUE)
})()
```

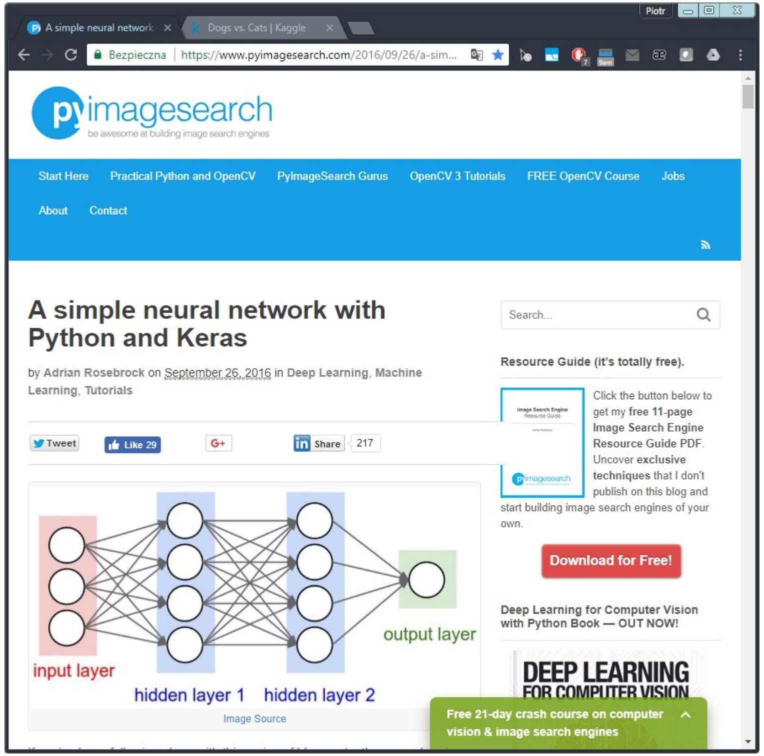
```
length : 1 053    lines : 29    Ln : 1    Col : 1    Sel : 0 | 0    Windows (CR LF)    UTF-8    INS
```

```
LogLevel: INFO
```

```
RSuiteVersion: 0.22.231
RVersion: 3.4
Project: PythonInR
Repositories: MRAN[2018-02-01]
Artifacts: config_templ.txt
```

Python code adapted from:

https://www.pyimagesearch.com/2016/09/26/a-simple-neural-network-with-python-and-keras



A simple neural network with Python and Keras

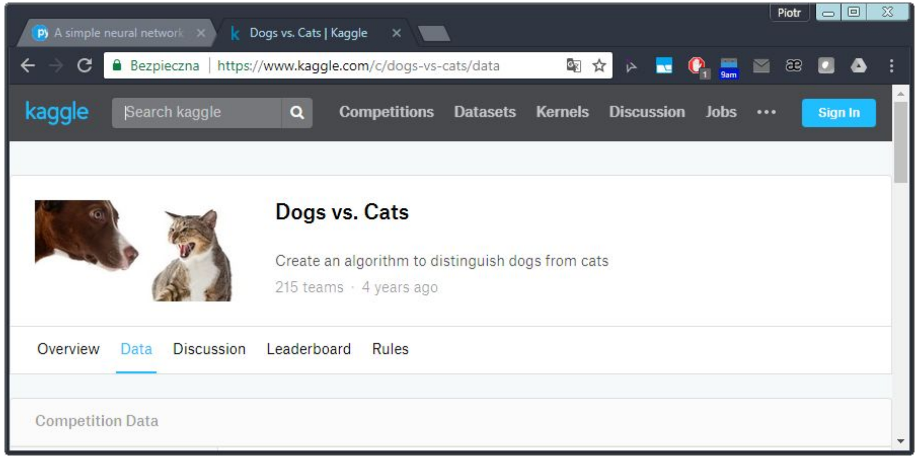
by Adrian Rosebrock on September 26, 2016 in Deep Learning, Machine Learning, Tutorials

Resource Guide (it's totally free).

Click the button below to get my free 11-page Image Search Engine Resource Guide PDF. Uncover exclusive techniques that I don't publish on this blog and start building image search engines of your own.

Deep Learning for Computer Vision with Python Book — OUT NOW!

Free 21-day crash course on computer vision & image search engines



Dogs vs. Cats

Create an algorithm to distinguish dogs from cats

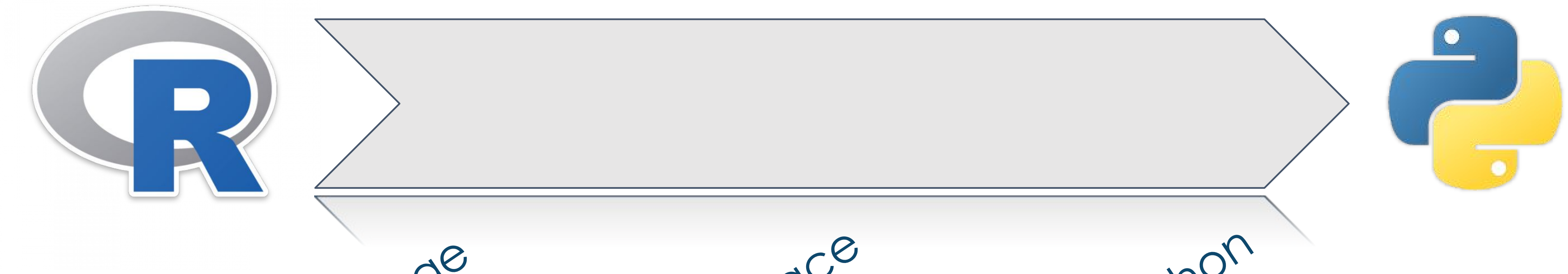
215 teams · 4 years ago

Competition Data

Dataset downloaded from:

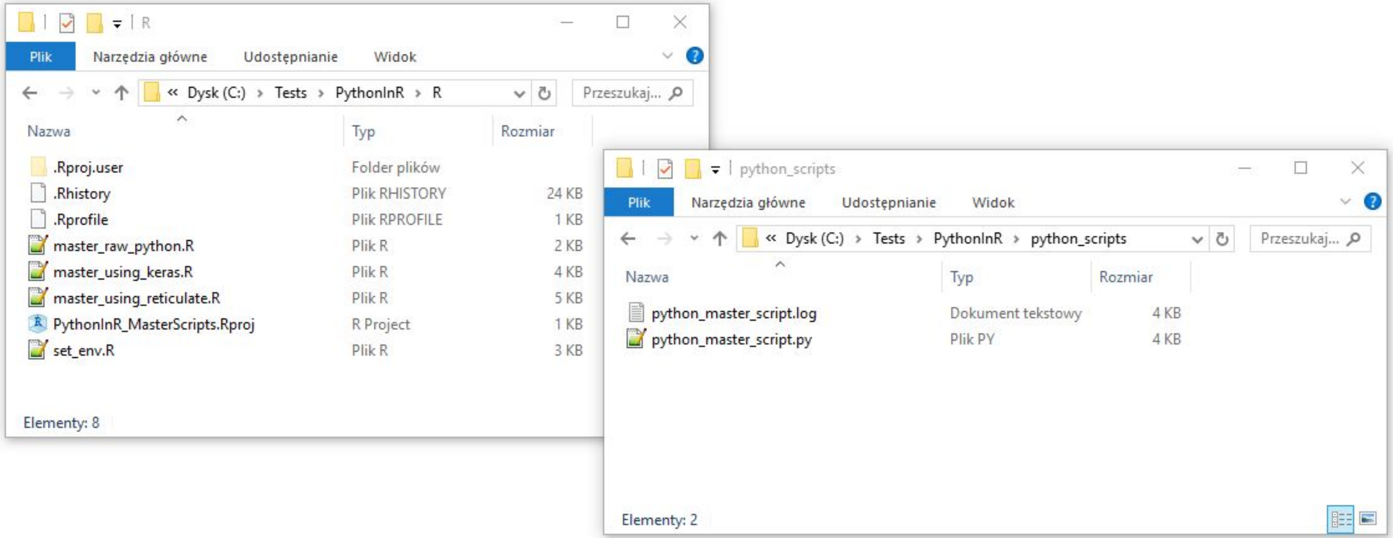
https://www.kaggle.com/c/dogs-vs-cats/data

3 ways to use Keras in R



1. keras R package
2. reticulate interface
3. running raw Python

3 ways to use Keras in R



Nazwa	Typ	Rozmiar
.Rproj.user	Folder plików	
.Rhistory	Plik RHISTORY	24 KB
.Rprofile	Plik RPROFILE	1 KB
master_raw_python.R	Plik R	2 KB
master_using_keras.R	Plik R	4 KB
master_using_reticulate.R	Plik R	5 KB
PythonInR_MasterScripts.Rproj	R Project	1 KB
set_env.R	Plik R	3 KB

Elementy: 8

Nazwa	Typ	Rozmiar
python_master_script.log	Dokument tekstowy	4 KB
python_master_script.py	Plik PY	4 KB

Elementy: 2

No matter how you use it, remember to use local Python!

```
use_python(python = dirname(...), required = TRUE)
```

1. *keras* R package

- most natural for R users
- wraps all Keras/Tensorflow features with R syntax
- provides some R Studio add-ins
- still uses *reticulate* and runs Python under the hood

```r
library(keras)

# keras model architecture
model <- keras_model_sequential() %>%
  layer_dense(units = 768, input_shape = 3072, kernel_initializer =
"uniform", activation = "relu") %>%
  layer_dense(units = 384, kernel_initializer = "uniform",
activation = "relu") %>%
  layer_dense(units = 2, activation = "softmax")

# keras model compilation
loginfo("Compiling model...")
model %>% compile(
  optimizer = optimizer_sgd(lr = 0.01),
  loss = "binary_crossentropy",
  metrics = "accuracy")

# keras model training and evaluation
loginfo("Model fitting and evaluation...")
model %>% fit(train_data, train_labels,
              epochs = 20,
              batch_size = 128,
              validation_data = list(valid_data, valid_labels))
loginfo("Model training complete.")
```

2. *reticulate* interface

- general interface for Python modules
- automatic conversions between Python and R data types (careful when passing integers, tuples, lists to Python functions)
- use “$” operator instead of Python’s “.”

```r
library(reticulate)

Sequential <- import("keras.models")$Sequential
Activation <- import("keras.layers")$Activation
SGD <- import("keras.optimizers")$SGD
Dense <- import("keras.layers")$Dense

# keras model architecture
model <- Sequential()
model$add(Dense(768L, input_dim = 3072L, init = "uniform",
activation = "relu"))
model$add(Dense(384L, init = "uniform", activation = "relu"))
model$add(Dense(2L))
model$add(Activation("softmax"))

# keras model compilation
loginfo("Compiling model...")
sgd <- SGD(lr = 0.01)
model$compile(loss = "binary_crossentropy",
              optimizer = sgd,
              metrics = list("accuracy"))

# keras model training and evaluation
loginfo("Starting model training...")
model$fit(train_data, train_labels,
          epochs = 20L,
          batch_size = 128L,
          validation_data = list(valid_data, valid_labels))
loginfo("Model training complete.")
```

3. running raw Python

- handy when you already have some code in Python
- hard to control Python script execution from R (separate loggers etc.)
- all objects from Python session available in R (converted or not)

```
library(reticulate)
```

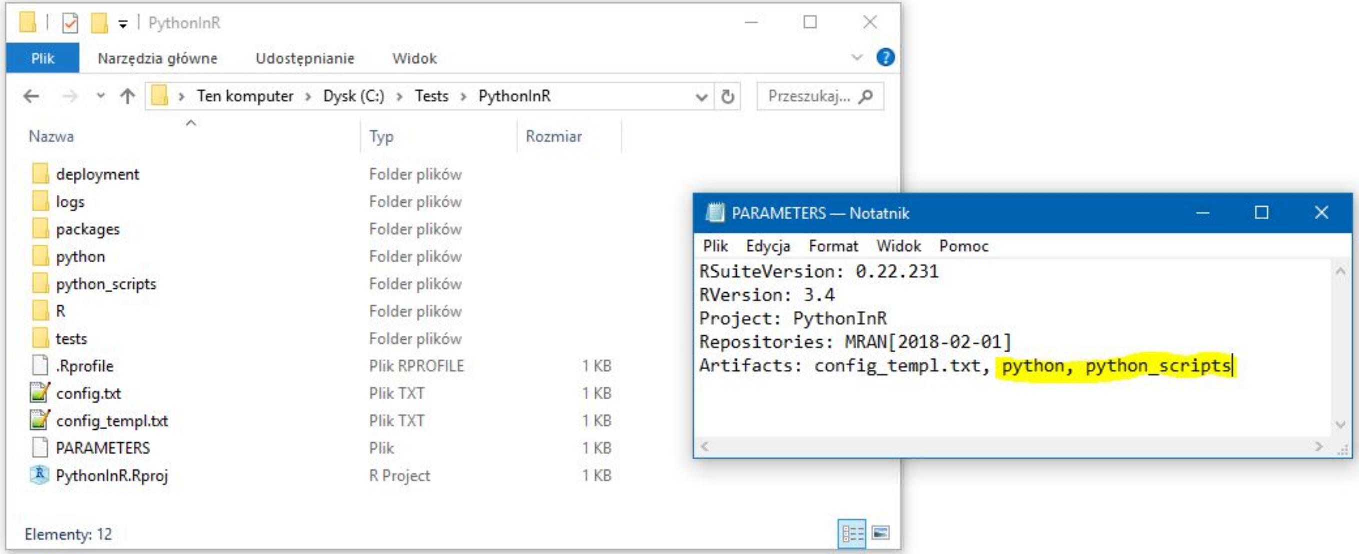
```
from keras.models import Sequential
from keras.layers import Activation
from keras.optimizers import SGD
from keras.layers import Dense

# keras model architecture
model = Sequential()
model.add(Dense(768, input_dim=3072, init="uniform",
activation="relu"))
model.add(Dense(384, init="uniform", activation="relu"))
model.add(Dense(2))
model.add(Activation("softmax"))

# keras model compilation
log.info("Compiling model...")
sgd = SGD(lr=0.01)
model.compile(loss="binary_crossentropy",
              optimizer=sgd,
              metrics=["accuracy"])

# keras model training and evaluation
log.info("Starting model training...")
model.fit(train_data, train_labels,
          epochs=20,
          batch_size=128,
          validation_data=[valid_data, valid_labels])
log.info("Model training complete.")
```

```
py_script_results <- py_run_file(python_script_fpath, convert = TRUE)
```

```
> py_script_results$model
<keras.models.Sequential>
```

8. Make internal Python env an artifact of R Suite project

After the development is done, 'python' folder containing local Python environment with all needed binary packages should be listed as an 'artifact' in PARAMETERS file. Also, if there are any Python scripts or anything else that should be additionally included in the deployment package, it needs to be marked as 'artifact'.



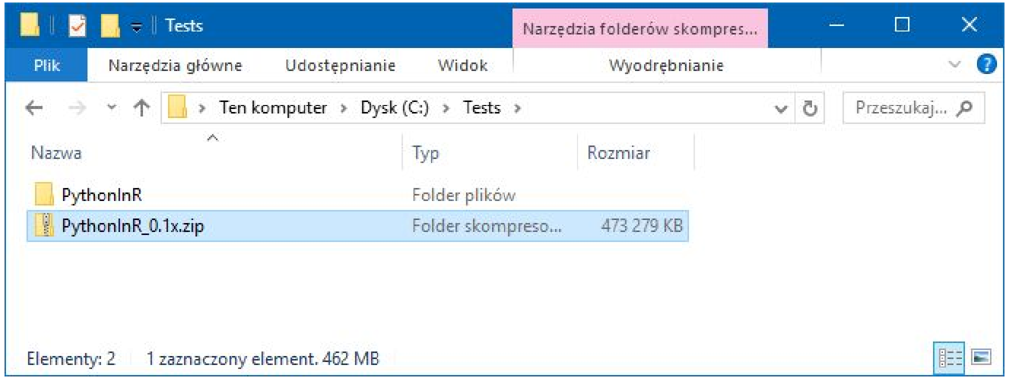
Nazwa	Typ	Rozmiar
deployment	Folder plików	
logs	Folder plików	
packages	Folder plików	
python	Folder plików	
python_scripts	Folder plików	
R	Folder plików	
tests	Folder plików	
.Rprofile	Plik RPROFILE	1 KB
config.txt	Plik TXT	1 KB
config_templ.txt	Plik TXT	1 KB
PARAMETERS	Plik	1 KB
PythonInR.Rproj	R Project	1 KB

```
RSuiteVersion: 0.22.231
RVersion: 3.4
Project: PythonInR
Repositories: MRAN[2018-02-01]
Artifacts: config_templ.txt, python, python_scripts
```

9. Create deployment package

```
C:\Tests\PythonInR>rsuite proj zip -p C:\Tests
2018-02-05 20:41:53 INFO:rsuite:Installing PackageVersions (for R 3.4) ...
2018-02-05 20:41:59 INFO:rsuite:Successfuly installed 1 packages
2018-02-05 20:41:59 INFO:rsuite:Preparing files for zipping...
2018-02-05 20:47:15 INFO:rsuite:... done. Creating zip file PythonInR_0.1x.zip ...
2018-02-05 20:51:41 INFO:rsuite:Zip file created: C:\Tests/PythonInR_0.1x.zip
```

If the R Suite project folder was under Git or SVN version control, created deployment package will be assigned with revision number.

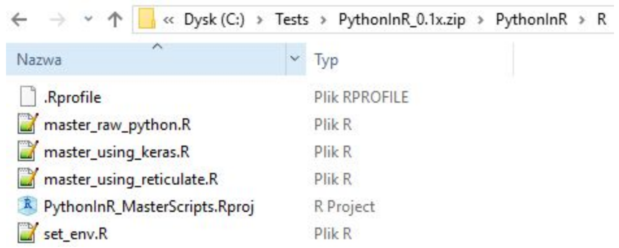


Nazwa	Typ	Rozmiar
PythonInR	Folder plików	
PythonInR_0.1x.zip	Folder skompreso...	473 279 KB

Elementy: 2   1 zaznaczony element. 462 MB

Deployment package contains...

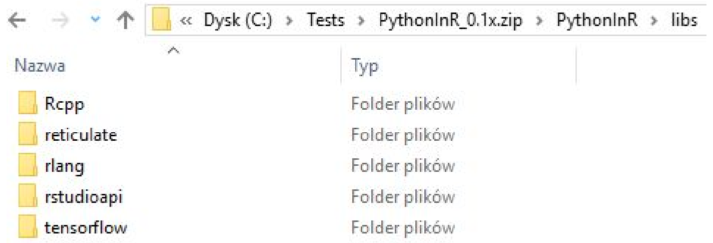
All R master scripts:



« Dysk (C:) › Tests › PythonInR_0.1x.zip › PythonInR › R

Nazwa	Typ
.Rprofile	Plik RPROFILE
master_raw_python.R	Plik R
master_using_keras.R	Plik R
master_using_reticulate.R	Plik R
PythonInR_MasterScripts.Rproj	R Project
set_env.R	Plik R

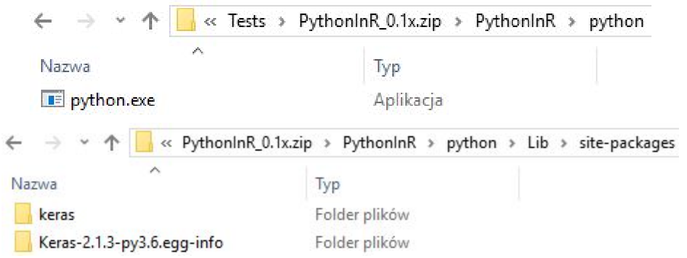
All R binary packages:



« Dysk (C:) › Tests › PythonInR_0.1x.zip › PythonInR › libs

Nazwa	Typ
Rcpp	Folder plików
reticulate	Folder plików
rlang	Folder plików
rstudioapi	Folder plików
tensorflow	Folder plików

Internal Python environment:



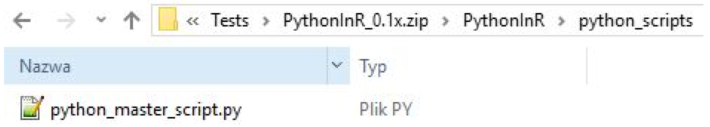
« Tests › PythonInR_0.1x.zip › PythonInR › python

Nazwa	Typ
python.exe	Aplikacja

« PythonInR_0.1x.zip › PythonInR › python › Lib › site-packages

Nazwa	Typ
keras	Folder plików
Keras-2.1.3-py3.6.egg-info	Folder plików

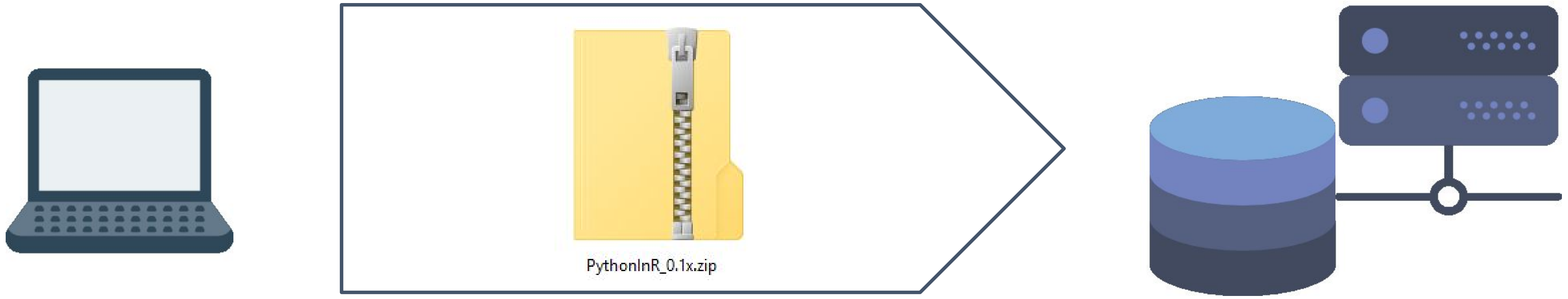
Additional Python scripts:



« Tests › PythonInR_0.1x.zip › PythonInR › python_scripts

Nazwa	Typ
python_master_script.py	Plik PY

Package deployment



Dev:

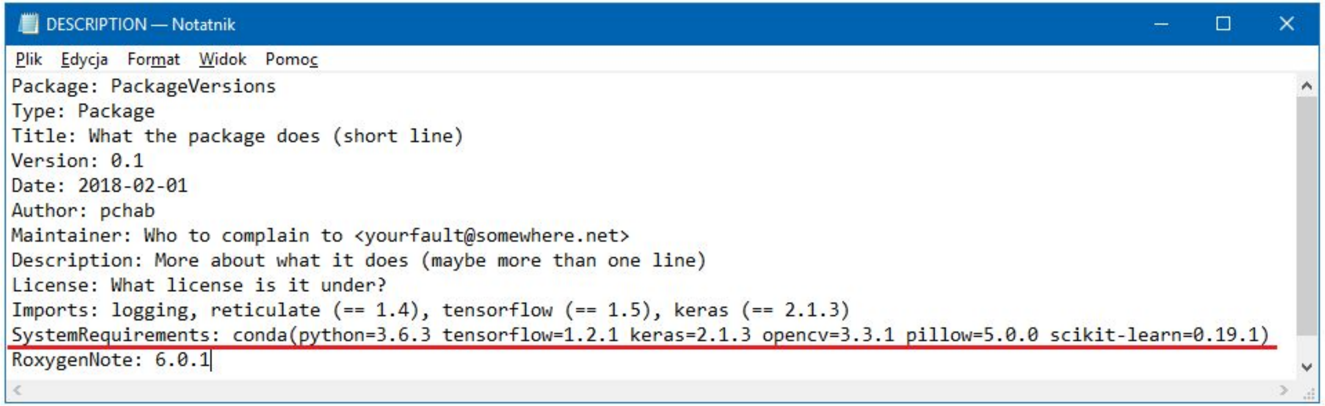
- binary consistent with Prod
- R
- R Suite
- Miniconda

Prod:

- binary consistent with Dev
- R

Soon in R Suite...

- add Python dependencies to R packages DESCRIPTION



```
Package: PackageVersions
Type: Package
Title: What the package does (short line)
Version: 0.1
Date: 2018-02-01
Author: pchab
Maintainer: Who to complain to <yourfault@somewhere.net>
Description: More about what it does (maybe more than one line)
License: What license is it under?
Imports: logging, reticulate (== 1.4), tensorflow (== 1.5), keras (== 2.1.3)
SystemRequirements: conda(python=3.6.3 tensorflow=1.2.1 keras=2.1.3 opencv=3.3.1 pillow=5.0.0 scikit-learn=0.19.1)
RoxygenNote: 6.0.1
```

- `rsuite sysreqs collect; rsuite sysreqs install`
- check if conda is installed, then create local Python env
- not only Python dependencies; also e.g. Linux system libraries

Thank You!

http://rsuite.io

https://github.com/WLOGSolutions/RSuite

Piotr Chaberski

pchaberski@wlogsolutions.com
info@wlogsolutions.com