Set Up Virtual Machine

Deepti Gupta

2020-03-18

## Install Software

* Most recent version of R: <https://cloud.r-project.org/>
* Most recent version of RStudio: <https://rstudio.com/products/rstudio/download/#download>

##r chunk - do not change these  
R.version

## \_   
## platform x86\_64-w64-mingw32   
## arch x86\_64   
## os mingw32   
## system x86\_64, mingw32   
## status   
## major 3   
## minor 6.1   
## year 2019   
## month 07   
## day 05   
## svn rev 76782   
## language R   
## version.string R version 3.6.1 (2019-07-05)  
## nickname Action of the Toes

#RStudio.Version() run this line but it won't knit with it "on"

## Set up your python

* Install the reticulate package (do not include this code).
* Load the reticulate library.

library(reticulate)

## Warning: package 'reticulate' was built under R version 3.6.2

## Install Miniconda

Try typing py\_config() below. You should get a prompt to install Miniconda. If not, use install\_miniconda().

py\_config()

## python: C:/Users/Deepti/AppData/Local/r-miniconda/envs/r-reticulate/python.exe  
## libpython: C:/Users/Deepti/AppData/Local/r-miniconda/envs/r-reticulate/python36.dll  
## pythonhome: C:/Users/Deepti/AppData/Local/r-miniconda/envs/r-reticulate  
## version: 3.6.10 |Anaconda, Inc.| (default, Jan 7 2020, 15:18:16) [MSC v.1916 64 bit (AMD64)]  
## Architecture: 64bit  
## numpy: C:/Users/Deepti/AppData/Local/r-miniconda/envs/r-reticulate/Lib/site-packages/numpy  
## numpy\_version: 1.18.1

## Show you’ve installed Python

Run py\_config() in the *R* chunk below.

py\_config()

## python: C:/Users/Deepti/AppData/Local/r-miniconda/envs/r-reticulate/python.exe  
## libpython: C:/Users/Deepti/AppData/Local/r-miniconda/envs/r-reticulate/python36.dll  
## pythonhome: C:/Users/Deepti/AppData/Local/r-miniconda/envs/r-reticulate  
## version: 3.6.10 |Anaconda, Inc.| (default, Jan 7 2020, 15:18:16) [MSC v.1916 64 bit (AMD64)]  
## Architecture: 64bit  
## numpy: C:/Users/Deepti/AppData/Local/r-miniconda/envs/r-reticulate/Lib/site-packages/numpy  
## numpy\_version: 1.18.1

## Let’s do some R

* In this chunk, we will load a dataset - use data(rock) to load it.
* Use the head() function to print out the first six rows of the dataset.

data(rock)  
  
head(rock, 6)

## area peri shape perm  
## 1 4990 2791.90 0.0903296 6.3  
## 2 7002 3892.60 0.1486220 6.3  
## 3 7558 3930.66 0.1833120 6.3  
## 4 7352 3869.32 0.1170630 6.3  
## 5 7943 3948.54 0.1224170 17.1  
## 6 7979 4010.15 0.1670450 17.1

## Let’s do some Python

* First, let’s install some packages.
* You will need numpy, nltk, spacy, scikit-learn and pandas for starters.
* You can check if you have them first by using py\_module\_available("PACKAGE").
* Note: scikit-learn is a special package, you can check if it’s avaliable by using sklearn but you install it with scikit-learn.
* If any of these return FALSE, then install them using `py\_i

py\_module\_available(“numpy”) py\_module\_available(“nltk”) py\_module\_available(“spacy”) py\_module\_available(“sklearn”) py\_module\_available(“pandas”) nstall(“PACKAGE”). If you receive an error saying it cannot be found on Minicode, usepy\_install(“PACKAGE”, pip = T)`.

#if they are all TRUE, leave this blank  
##r chunk

## Call a dataset in Python

* First, load the sklearn library, it has several sample datasets. You load python packages by using import PACKAGE. Note that you install and call this package different names (scikit-learn = sklearn).
* Next, import the datasets part of sklearn by doing from PACKAGE import FUNCTION. Therefore, you should use from sklearn import datasets.
* Then call the boston dataset by doing: dataset\_boston = datasets.load\_boston().
* To print out the first six rows, use the .head() function: df\_boston.head(), after converting the file with pandas (code included below).

from sklearn import datasets  
##convert to pandas  
import pandas as pd  
  
dataset\_boston = datasets.load\_boston(return\_X\_y = False)  
  
  
  
df\_boston = pd.DataFrame(data=dataset\_boston.data, columns=dataset\_boston.feature\_names)  
  
df\_boston.shape

## (506, 13)

QUESTION: Look in your environment window. What do you see?

Different packages are added now under what earlier was only Global Environments. Like package datasets has all the datasets in it.

## Print out Python information in R

* You can have the two environments interact. To print out information from Python in R: py$VARNAME.
* Normally, to print out R dataset columns, you do DATAFRAME$COLUMN. Try to print out the CRIM column from your df\_boston variable.

##r chunk  
py$sys$flags

## Tuple (13 items)

py$df\_boston$CRIM

## [1] 0.00632 0.02731 0.02729 0.03237 0.06905 0.02985 0.08829 0.14455  
## [9] 0.21124 0.17004 0.22489 0.11747 0.09378 0.62976 0.63796 0.62739  
## [17] 1.05393 0.78420 0.80271 0.72580 1.25179 0.85204 1.23247 0.98843  
## [25] 0.75026 0.84054 0.67191 0.95577 0.77299 1.00245 1.13081 1.35472  
## [33] 1.38799 1.15172 1.61282 0.06417 0.09744 0.08014 0.17505 0.02763  
## [41] 0.03359 0.12744 0.14150 0.15936 0.12269 0.17142 0.18836 0.22927  
## [49] 0.25387 0.21977 0.08873 0.04337 0.05360 0.04981 0.01360 0.01311  
## [57] 0.02055 0.01432 0.15445 0.10328 0.14932 0.17171 0.11027 0.12650  
## [65] 0.01951 0.03584 0.04379 0.05789 0.13554 0.12816 0.08826 0.15876  
## [73] 0.09164 0.19539 0.07896 0.09512 0.10153 0.08707 0.05646 0.08387  
## [81] 0.04113 0.04462 0.03659 0.03551 0.05059 0.05735 0.05188 0.07151  
## [89] 0.05660 0.05302 0.04684 0.03932 0.04203 0.02875 0.04294 0.12204  
## [97] 0.11504 0.12083 0.08187 0.06860 0.14866 0.11432 0.22876 0.21161  
## [105] 0.13960 0.13262 0.17120 0.13117 0.12802 0.26363 0.10793 0.10084  
## [113] 0.12329 0.22212 0.14231 0.17134 0.13158 0.15098 0.13058 0.14476  
## [121] 0.06899 0.07165 0.09299 0.15038 0.09849 0.16902 0.38735 0.25915  
## [129] 0.32543 0.88125 0.34006 1.19294 0.59005 0.32982 0.97617 0.55778  
## [137] 0.32264 0.35233 0.24980 0.54452 0.29090 1.62864 3.32105 4.09740  
## [145] 2.77974 2.37934 2.15505 2.36862 2.33099 2.73397 1.65660 1.49632  
## [153] 1.12658 2.14918 1.41385 3.53501 2.44668 1.22358 1.34284 1.42502  
## [161] 1.27346 1.46336 1.83377 1.51902 2.24236 2.92400 2.01019 1.80028  
## [169] 2.30040 2.44953 1.20742 2.31390 0.13914 0.09178 0.08447 0.06664  
## [177] 0.07022 0.05425 0.06642 0.05780 0.06588 0.06888 0.09103 0.10008  
## [185] 0.08308 0.06047 0.05602 0.07875 0.12579 0.08370 0.09068 0.06911  
## [193] 0.08664 0.02187 0.01439 0.01381 0.04011 0.04666 0.03768 0.03150  
## [201] 0.01778 0.03445 0.02177 0.03510 0.02009 0.13642 0.22969 0.25199  
## [209] 0.13587 0.43571 0.17446 0.37578 0.21719 0.14052 0.28955 0.19802  
## [217] 0.04560 0.07013 0.11069 0.11425 0.35809 0.40771 0.62356 0.61470  
## [225] 0.31533 0.52693 0.38214 0.41238 0.29819 0.44178 0.53700 0.46296  
## [233] 0.57529 0.33147 0.44791 0.33045 0.52058 0.51183 0.08244 0.09252  
## [241] 0.11329 0.10612 0.10290 0.12757 0.20608 0.19133 0.33983 0.19657  
## [249] 0.16439 0.19073 0.14030 0.21409 0.08221 0.36894 0.04819 0.03548  
## [257] 0.01538 0.61154 0.66351 0.65665 0.54011 0.53412 0.52014 0.82526  
## [265] 0.55007 0.76162 0.78570 0.57834 0.54050 0.09065 0.29916 0.16211  
## [273] 0.11460 0.22188 0.05644 0.09604 0.10469 0.06127 0.07978 0.21038  
## [281] 0.03578 0.03705 0.06129 0.01501 0.00906 0.01096 0.01965 0.03871  
## [289] 0.04590 0.04297 0.03502 0.07886 0.03615 0.08265 0.08199 0.12932  
## [297] 0.05372 0.14103 0.06466 0.05561 0.04417 0.03537 0.09266 0.10000  
## [305] 0.05515 0.05479 0.07503 0.04932 0.49298 0.34940 2.63548 0.79041  
## [313] 0.26169 0.26938 0.36920 0.25356 0.31827 0.24522 0.40202 0.47547  
## [321] 0.16760 0.18159 0.35114 0.28392 0.34109 0.19186 0.30347 0.24103  
## [329] 0.06617 0.06724 0.04544 0.05023 0.03466 0.05083 0.03738 0.03961  
## [337] 0.03427 0.03041 0.03306 0.05497 0.06151 0.01301 0.02498 0.02543  
## [345] 0.03049 0.03113 0.06162 0.01870 0.01501 0.02899 0.06211 0.07950  
## [353] 0.07244 0.01709 0.04301 0.10659 8.98296 3.84970 5.20177 4.26131  
## [361] 4.54192 3.83684 3.67822 4.22239 3.47428 4.55587 3.69695 13.52220  
## [369] 4.89822 5.66998 6.53876 9.23230 8.26725 11.10810 18.49820 19.60910  
## [377] 15.28800 9.82349 23.64820 17.86670 88.97620 15.87440 9.18702 7.99248  
## [385] 20.08490 16.81180 24.39380 22.59710 14.33370 8.15174 6.96215 5.29305  
## [393] 11.57790 8.64476 13.35980 8.71675 5.87205 7.67202 38.35180 9.91655  
## [401] 25.04610 14.23620 9.59571 24.80170 41.52920 67.92080 20.71620 11.95110  
## [409] 7.40389 14.43830 51.13580 14.05070 18.81100 28.65580 45.74610 18.08460  
## [417] 10.83420 25.94060 73.53410 11.81230 11.08740 7.02259 12.04820 7.05042  
## [425] 8.79212 15.86030 12.24720 37.66190 7.36711 9.33889 8.49213 10.06230  
## [433] 6.44405 5.58107 13.91340 11.16040 14.42080 15.17720 13.67810 9.39063  
## [441] 22.05110 9.72418 5.66637 9.96654 12.80230 10.67180 6.28807 9.92485  
## [449] 9.32909 7.52601 6.71772 5.44114 5.09017 8.24809 9.51363 4.75237  
## [457] 4.66883 8.20058 7.75223 6.80117 4.81213 3.69311 6.65492 5.82115  
## [465] 7.83932 3.16360 3.77498 4.42228 15.57570 13.07510 4.34879 4.03841  
## [473] 3.56868 4.64689 8.05579 6.39312 4.87141 15.02340 10.23300 14.33370  
## [481] 5.82401 5.70818 5.73116 2.81838 2.37857 3.67367 5.69175 4.83567  
## [489] 0.15086 0.18337 0.20746 0.10574 0.11132 0.17331 0.27957 0.17899  
## [497] 0.28960 0.26838 0.23912 0.17783 0.22438 0.06263 0.04527 0.06076  
## [505] 0.10959 0.04741

## Print our R in Python

* When using R in Python, instead of $, we use . like this: r.VARNAME.
* To print out a single column, you use DATAFRAME["COLUMNNAME"]. Try printing out the shape column in the rock dataset.

##python chunk  
  
  
head(rock, 6)

## area peri shape perm  
## 1 4990 2791.90 0.0903296 6.3  
## 2 7002 3892.60 0.1486220 6.3  
## 3 7558 3930.66 0.1833120 6.3  
## 4 7352 3869.32 0.1170630 6.3  
## 5 7943 3948.54 0.1224170 17.1  
## 6 7979 4010.15 0.1670450 17.1

rock["shape"]

## shape  
## 1 0.0903296  
## 2 0.1486220  
## 3 0.1833120  
## 4 0.1170630  
## 5 0.1224170  
## 6 0.1670450  
## 7 0.1896510  
## 8 0.1641270  
## 9 0.2036540  
## 10 0.1623940  
## 11 0.1509440  
## 12 0.1481410  
## 13 0.2285950  
## 14 0.2316230  
## 15 0.1725670  
## 16 0.1534810  
## 17 0.2043140  
## 18 0.2627270  
## 19 0.2000710  
## 20 0.1448100  
## 21 0.1138520  
## 22 0.2910290  
## 23 0.2400770  
## 24 0.1618650  
## 25 0.2808870  
## 26 0.1794550  
## 27 0.1918020  
## 28 0.1330830  
## 29 0.2252140  
## 30 0.3412730  
## 31 0.3116460  
## 32 0.2760160  
## 33 0.1976530  
## 34 0.3266350  
## 35 0.1541920  
## 36 0.2760160  
## 37 0.1769690  
## 38 0.4387120  
## 39 0.1635860  
## 40 0.2538320  
## 41 0.3286410  
## 42 0.2300810  
## 43 0.4641250  
## 44 0.4204770  
## 45 0.2007440  
## 46 0.2626510  
## 47 0.1824530  
## 48 0.2004470

## Get started with PyCharm!

* Great job! Here’s what you learned:
  + Installed Python!
  + You know how to install and load the libraries in both languages.
  + You know how to load built in datasets in both languages.
  + You know how to print out data from one language to another.
* Turn this document in for credit –> hit KNIT –> turn in the HTML file.
* Be sure to fill in your name at the top!
* Be sure to answer the embedded questions!