# Using miniCRAN in Azure ML

Azure Machine Learning Studio is a drag-and-drop tool to deploy data-driven solutions. It contains pre-built items including data preparation tools and Machine Learning algorithms. In addition, it allows to include R and Python custom scripts.

In order to build powerful R tools, you might want to use some packages from the CRAN repository. Azure ML already contains just a few packages, so you might need to include some others. There are 7000+ packages out of which you will need just a few. For this purpose, you can use the miniCRAN package which creates a local repository containing a selection of packages and their dependencies.

After having subscribed to AzureML, the first step is creating a miniCRAN local repository. You can find some instructions in this link

<http://blog.revolutionanalytics.com/2014/10/introducing-minicran.html>

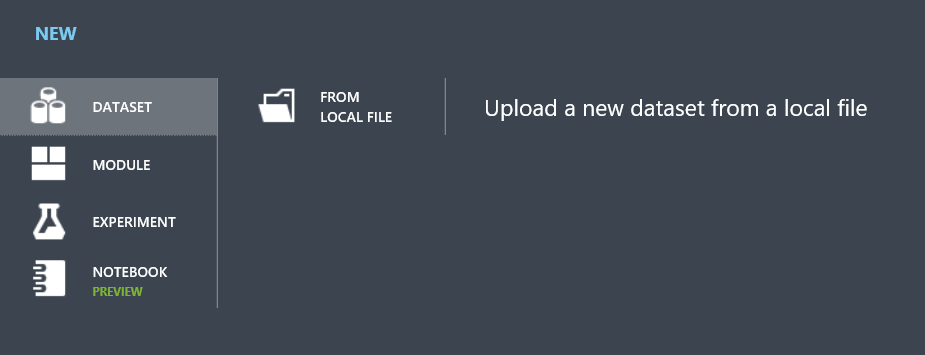
Azure ML is based on Windows, so in the function *makeRepo* you need to include the argument *type = "win.binary"*. In this demo, you will use the ggplot2 package, so it should be in the list.

After you create your own repository (called repoCRANwin, for instance), the package binary files are stored into the folder *repoCRANwin\bin\windows\contrib\3.1*.

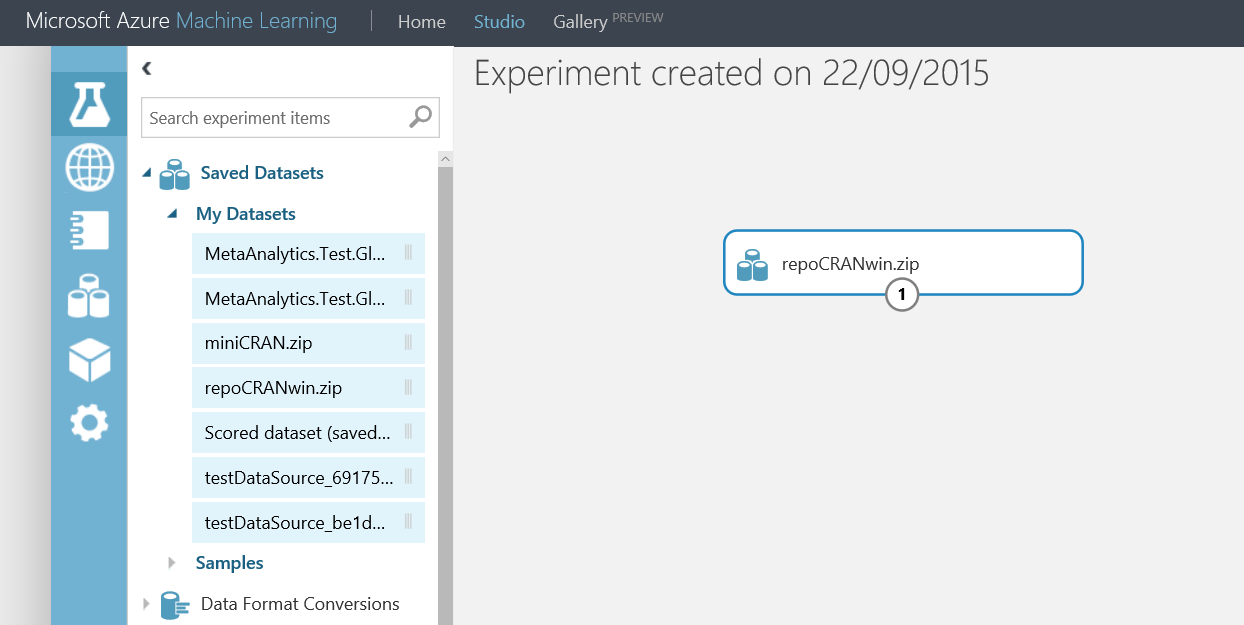
Now, you need to zip the main folder *repoCRANwin* and upload it to Azure ML. For this purpose, from the Azure ML menu, you need to select:

*New -> Dataset -> From local file*

After having clicked on *New* (on the bottom-left), you should see this

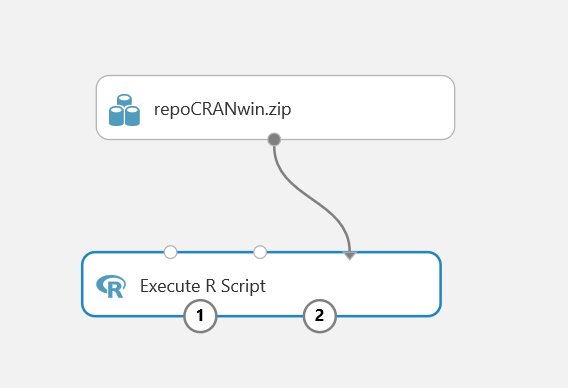


Now you need to create a new Azure ML experiment, open the *Saved Datasets -> My Dataset* tab, and drag and drop repoCRANwin.zip into the experiment.



Then, you include a custom R script from *R Language Modules -> Execute R script.*

In order to connect *repoCRANwin.zip* to the R script, you need to drag its output to the right-hand side input of *Execute R script*.



Opening the *Execute R script*, you can edit its R code. Your targets are

* Setting up the miniCRAN repository
* Extracting the list of available packages
* Testing a package, e.g. ggplot2

This is the R script to include:

*# setting-up the repository*

*uri\_repo <- "file:///C:/src/repoCRANwin/"*

*options(repos = uri\_repo)*

*# extracting the list of available packages*

*table\_packages <- data.frame(package = rownames(available.packages()))*

*# installing the ggplot2 package*

*install.packages("ggplot2")*

*library("ggplot2")*

*# building a sample ggplot2 chart*

*p <- qplot(iris$Species)*

*print(p)*

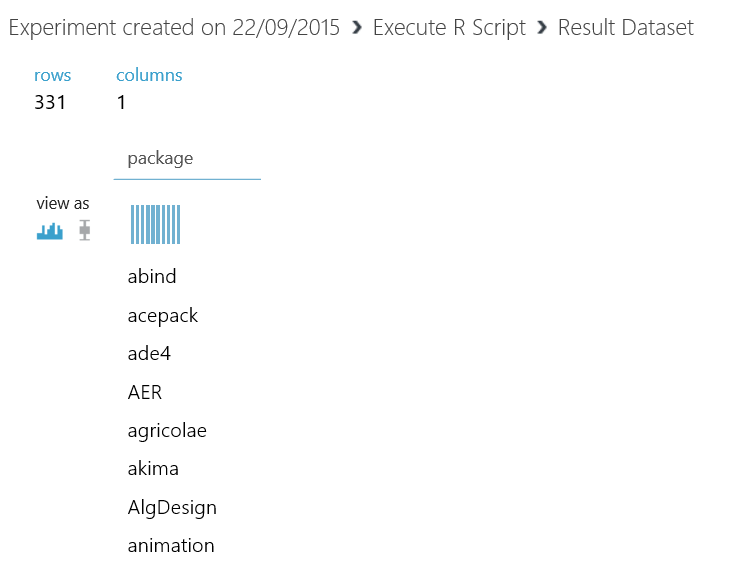
*# outputting the list of packages*

*maml.mapOutputPort("table\_packages")*

*Execute R script* has two outputs:

* the list of packages (on the bottom left-hand side)
* a sample ggplot2 chart (on the bottom right-hand side)

If you click on the left-hand side output and select “Visualize”, you’ll see this:



The “package” column contains the packages that can be installed and loaded.

If you click on the right-hand side output, you’ll see a sample ggplot2 chart. If this works, ggplot2 has been loaded and used properly, so we expect that most of the other packages will work.

Loading miniCRAN into an Azure ML R script allows you to access any package that you included. If you have a list of packages that you will use, you can just create a local miniCRAN archive and upload it. Then, you’ll just need to input miniCRAN to the related R scripts and include a few lines of R code to configure it into each script. A next step could be defining a miniCRAN repository for each topic. For instance, there might be one for data preparation, one for Machine Learning, and another for data visualization.