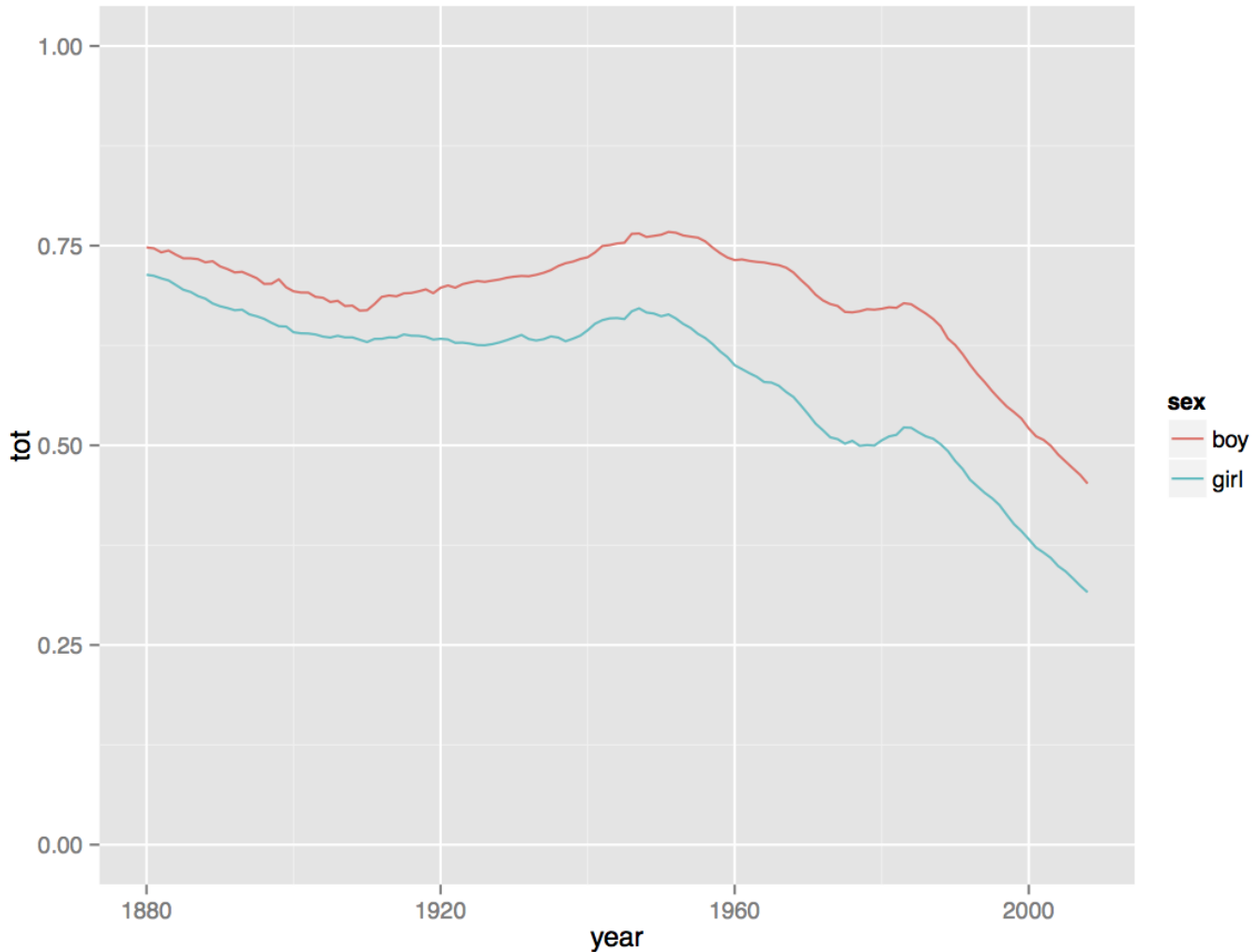


# Parametric Sweep with R and Python

This example uses a combination of R and Python to download data about baby names in the USA and calculate the percentage of babies per year with a name in the top 100 names of that year. The calculation is done in parallel with a parametric sweep job. The output looks something like this:



## Download Example Code

1. Open a Remote Desktop Connection to the head node. Log in with the username and password you set when you created the VM, **not** the domain user you created before installing HPC Pack. This will simplify the process later on.

## Install the Azure Command Line Interface

1. On the head node, go to <http://www.windowsazure.com/en-us/downloads/> and download installer packages for both the **Microsoft Azure PowerShell** and the **Cross-platform Command Line Interface**.

# Command line tools

## Windows downloads

### Windows Azure PowerShell

Last updated August 2013

[Get started tutorial »](#)

[Standalone installer »](#)

### Cross-platform Command Line Interface

Last updated August 2013

## Mac

### Command Line Interface

Last updated August 2013

[Get started tutorial »](#)

1. Run the Microsoft Azure PowerShell installer and install to the default location.
2. Run the Cross-platform Command Line Interface installer and install to the default location.
3. Open a PowerShell window and execute the commands below to configure the Microsoft Azure Command Line Interface:

```
cd Downloads  
azure account download
```

A browser window will open. Sign in when prompted and save the publishsettings file when the download begins. When the download completes, go back to the command line and import the publishsettings file:

```
azure account import <publishsettings file>
```

Now you can see your subscriptions:

```
azure account list
```

## Install and Package the Microsoft Azure Python SDK

1. Go to <http://www.windowsazure.com/en-us/downloads/> and select **Python** from the list on the left, then click **Windows** to download the Microsoft Azure Python SDK installer package.

Download and install language specific SDKs and tools for your platform of choice.

## .NET

[VS 2013 Install](#)  
[VS 2012 Install](#)  
[Client Libraries](#)  
[Documentation](#)  
[Previous Versions](#)

## Java

[Windows Install](#)  
[Mac Install](#)  
[Linux Install](#)  
[Documentation](#)

## Node.js

[Windows Install](#)  
[Mac Install](#)  
[Linux Install](#)  
[Documentation](#)

## PHP

[Windows Install](#)  
[Mac Install](#)  
[Linux Install](#)  
[Documentation](#)

## Python

[Windows Install](#)  
[Mac Install](#)  
[Linux Install](#)  
[Documentation](#)

## Ruby

[Windows Install](#)  
[Mac Install](#)  
[Linux Install](#)  
[Documentation](#)

## Mobile

[iOS Install](#)  
[Android Install](#)  
[Windows Store C# Install](#)  
[Windows Store JS Install](#)  
[Windows Phone 8 Install](#)  
[Documentation](#)

## Media

[iOS SDK Install](#)  
[Flash OSMF Install](#)  
[Windows 8 Install](#)  
[Silverlight Install](#)  
[.NET SDK Install](#)  
[Java SDK Install](#)  
[Documentation](#)

1. Run the Python SDK installer package and install to the default location. Python is now installed at **C:\Python27\python.exe** with pip at **C:\Python27\Scripts\pip.exe**.
2. We want to install Python on the compute nodes as well as the head node. Since we already have a fully-featured Python installation on the head node, we'll simply copy it to the compute nodes. Use [hpcpack create](#) to package the Python installation:

```
hpcpack create Python27.zip .\Python27
```

3. Use [hpcpack upload](#) to upload the package to the storage account associated with the AzureNode template you created earlier:

```
hpcpack upload Python27.zip /nodetemplate:"Default AzureNode Template" /relativePath:Python27
```

Be sure to use the **/relativePath** parameter. Otherwise Python will be placed on a path that involves a timestamp so it will be difficult to determine where the installation files are.

**NOTE:** Make sure you logged in as the right user.

If you get an "access denied" error when executing the above command, make sure you are logged in to the VM with the username and password you set when you created the VM and **not** any domain user. As a work-around, you can replace the /nodetemplate parameter with the /account and /key parameters. See the [hpcpack documentation](#) for more information.

## Install R on the Head Node and Package the R Installer

1. Log in to the VM with the username and password you set when you created the VM, **not** the domain user you created before installing HPC Pack. If you log in with a domain user, the following hpcpack commands will not work.
2. On the head node, download the [R base installer for Windows](#).
3. Run the installer on the head node and accept all default values.

4. We need to collect several files in order to automate the R installation on the compute nodes. On the cluster head node, make a folder named **RInstaller** in your home directory.
5. Copy the R installer (R-3.0.1-win.exe) to the RInstaller folder.
6. Copy **prep.r** from azuretraining-master\hpc-R-sweep to the RInstaller folder.
7. In the RInstaller folder, create an installation script named **install.bat** with these contents:

```
set root=%CCP_PACKAGE_ROOT%\RInstaller
%root%\R-3.0.1-win.exe /VERYSILENT
"D:\Program Files\R\R-3.0.1\bin\Rscript.exe" --no-restore --no-save %root%\prep.r
```

Note that the script calls R on the **D:** drive, not the C: drive.

1. Open a Command Prompt window and navigate to the folder containing RInstaller. For example, if you created the RInstaller folder in your home folder, navigate to your home folder. We'll be using the [hpcpack](#) command to distribute R to the cluster nodes.
2. Use [hpcpack create](#) to package the RInstaller folder for distribution to the cluster nodes:

```
hpcpack create RInstaller.zip RInstaller
```

3. Use [hpcpack upload](#) to upload the package to the storage account associated with the AzureNode template you created earlier:

```
hpcpack upload RInstaller.zip /nodetemplate:"Default AzureNode Template" /relativePath:RInstaller
```

Be sure to use the **/relativePath** parameter. Otherwise the R installer and install.bat will be placed on a path that involves a timestamp so it will be difficult to determine where the installation files are.

---

**NOTE:** Make sure you logged in as the right user.

If you get an "access denied" error when executing the above command, make sure you are logged in to the VM with the username and password you set when you created the VM and **not** any domain user. As a work-around, you can replace the /nodetemplate parameter with the /account and /key parameters. See the [hpcpack documentation](#) for more information.

---

1. Use [hpcpack list](#) to verify that the package is in your storage account:

```
hpcpack list /nodetemplate:"Default AzureNode Template"
```

## Create a Node Startup Script

1. We'll use a startup script to automatically install R when the Azure nodes boot. Create **startup.bat** with the following contents. If you already have a startup script, append these lines to your existing script:

```
xcopy /h/i/c/k/e/r/y %CCP_PACKAGE_ROOT%\Python27 C:\Python27
cd /D %CCP_PACKAGE_ROOT%\RInstaller
.\install.bat
```

When the nodes boot they will use the [hpcsync](#) command to automatically download and unpack the RInstaller.zip package. By default, hpcsync deploys files to a location on the Microsoft Azure nodes that is determined in part by the **%CCPPACKAGEROOT%** environment variable. This variable is set on Microsoft Azure nodes during the provisioning process. The extracted files are placed in a folder that is determined as follows: **%CCPPACKAGEROOT%\**. This is the expected location for SOA services, XLs, Excel workbooks, and startup scripts that are called from the node template. However, because we passed **/relativePath:RInstaller** to our hpcpack upload command, RInstaller.zip will be unpacked to **%CCPPACKAGEROOT%\RInstaller**. We used relativePath because we cannot easily determine the part of the default path.

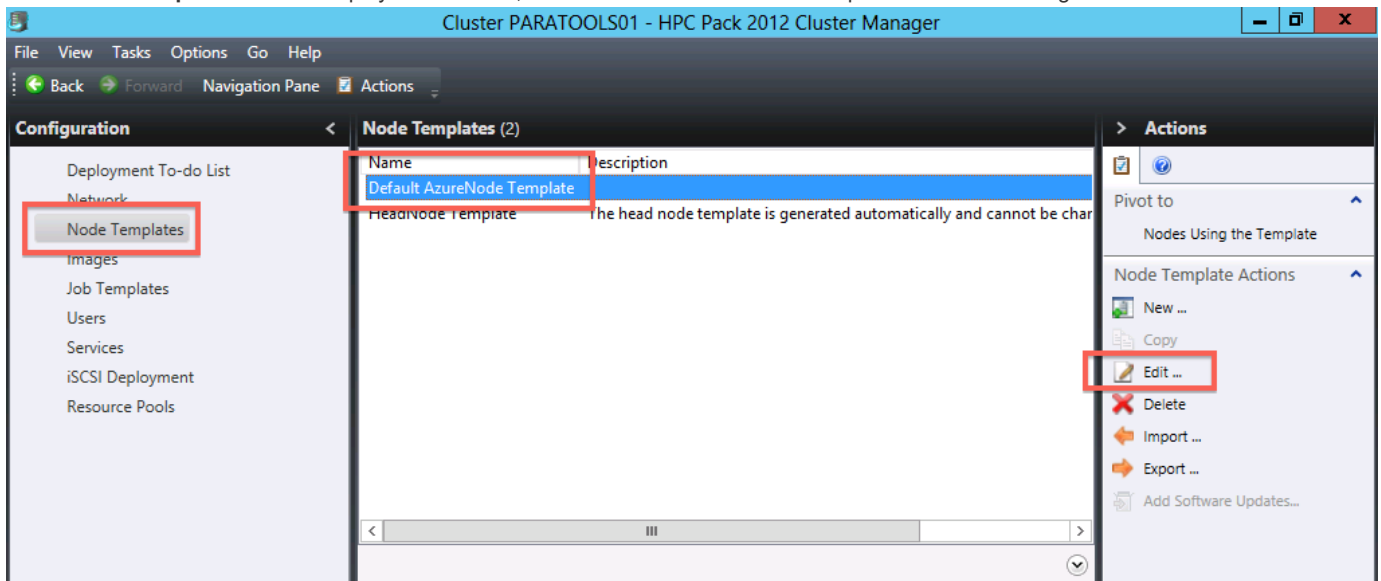
1. Open a command prompt, navigate to the folder containing startup.bat, and package the startup script:

```
hpcpack create startup.bat.zip startup.bat
```

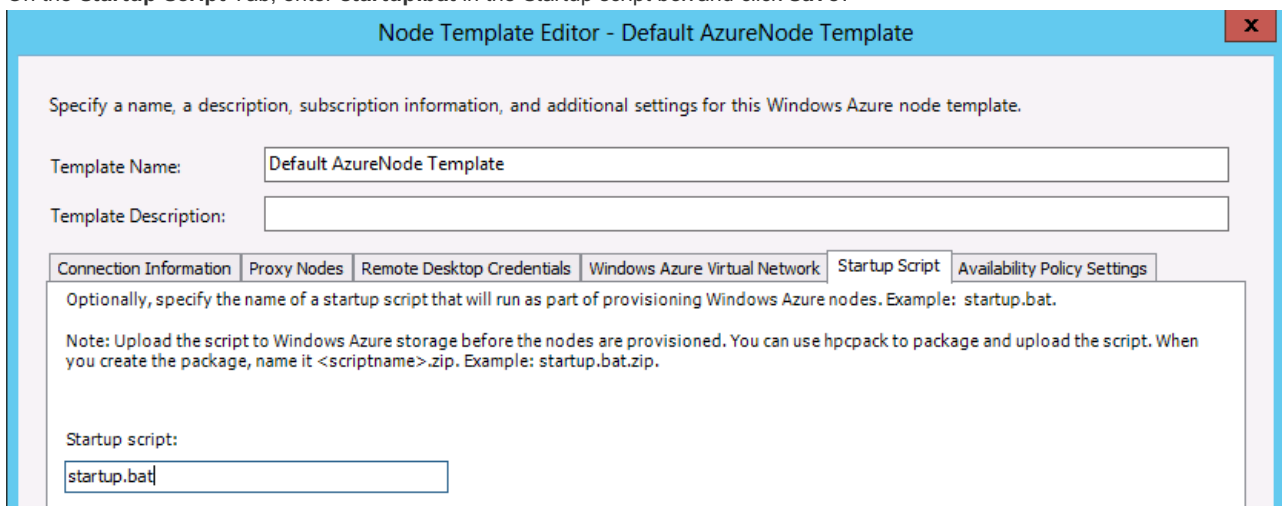
2. Upload the startup script to your storage account:

```
hpcpack upload startup.bat.zip /nodetemplate:"Default AzureNode Template"
```

3. Open the HPC Pack Cluster Manager. On the **Configuration** page, select **Node Templates** from the list on the left, select **Default AzureNode Template** from the display in the center, and click **Edit...** under Node Template Actions on the right.



4. On the **Startup Script** Tab, enter **startup.bat** in the Startup script box and click **Save**.



## Update or Add and Start Cluster Nodes

If you already have a group of Microsoft Azure Nodes online you can update them via **clusrun** and **hpcsync**: `clusrun /nodegroup:AzureNodes hpcsync clusrun /nodegroup:AzureNodes xcopy /h/i/c/k/e/r/y %CCPPACKAGEROOT%\Python27 C:\Python27 clusrun /nodegroup:AzureNodes %CCPPACKAGEROOT%\R\Installer\install.bat`

If you don't already have any nodes online you'll need to provision and start new compute nodes. The node startup script will automatically install R.

1. Open the Cluster Manager on the cluster head node.
2. On the Node Management page, select **Add Node** under Node Actions on the right.
3. Select **Add Microsoft Azure nodes** and click **Next**.
4. Verify that **Default AzureNode Template** is the selected template, enter **4** for the number of Microsoft Azure nodes, and select the **Medium** node size. Click **Next** and click **Finish**.

+
Add Node Wizard
✕

### Specify New Nodes

Select Deployment Method

Specify New Nodes

Summary

Select a Windows Azure node template, the size of the nodes, and the number of nodes that you want to add to the cluster. Then click Next.

Windows Azure node template: Default AzureNode Template

Number of Windows Azure nodes: 4

Size of Windows Azure nodes: Medium

i Windows Azure nodes deployed using a specific node template define a set of Windows Azure nodes. Windows Azure nodes in a set are managed as a set. You cannot start, stop, or delete individual Windows Azure nodes.

1. Select the first **AzureCN** node from the node list and click **Start** under Node Actions on the right.

Nodes (5)
> Actions

List
Heat Map
New Tab
+

Node Name	Node State	Node Health	Node Template
PARATOOLS01	Online	OK	HeadNode Tem
AzureCN-0032	Not-Deployed	Unapproved	Default AzureN
AzureCN-0031	Not-Deployed	Unapproved	Default AzureN
AzureCN-0030	Not-Deployed	Unapproved	Default AzureN
AzureCN-0029	Not-Deployed	Unapproved	Default AzureN

Pivot to

Jobs for the Selected Nodes

Failed Diagnostics for the Node:

Operations for the Nodes

Node Actions

Bring Online

Take Offline

Start

Stop

You will be notified that a set of nodes is being started. Verify that four nodes will be started and click **Start**.

1. The nodes are now provisioning (this will take a while). Once the provisioning is complete, they will be in the "Unapproved" state. To approve the nodes, select **all** the nodes and click **Bring Online** under Node Actions on the right.

Nodes (5)
> Actions

List
Heat Map
New Tab
+

Node Name	Node State	Node Health	Node Template
PARATOOLS01	Online	OK	HeadNode Tem
AzureCN-0032	Offline	OK	Default AzureN
AzureCN-0031	Offline	OK	Default AzureN
AzureCN-0030	Offline	OK	Default AzureN
AzureCN-0029	Offline	OK	Default AzureN

Pivot to

Jobs for the Selected Nodes

Failed Diagnostics for the Node:

Operations for the Nodes

Node Actions

Bring Online

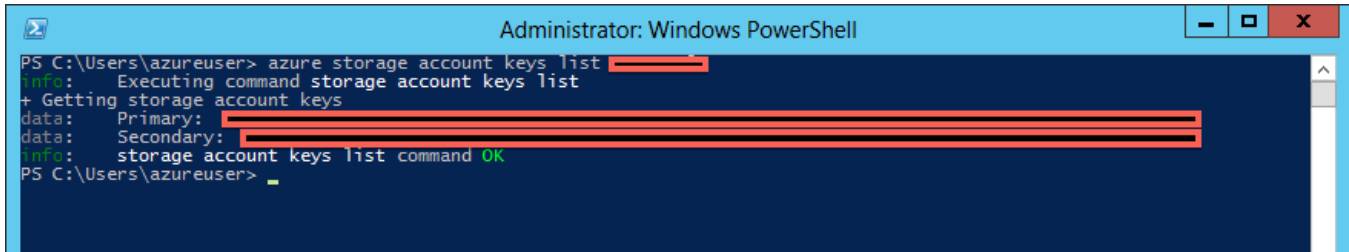
Take Offline

## Package Application Files

1. On the head node, make a folder named **bnames\_sweep** to contain the R script and supporting files.
2. Copy **downloadchunk.py**, **uploadchunk.py**, **top100.r**, and **runsweep.bat** from the training materials to the **bnames\_sweep** folder on the cluster head node.
3. We'll need your storage account key for the **runsweep.bat** script. Use the *azure command line interface* to get it. Replace **YOURACCOUNT** with your storage account name:

```
azure storage account keys list YOUR_ACCOUNT
```

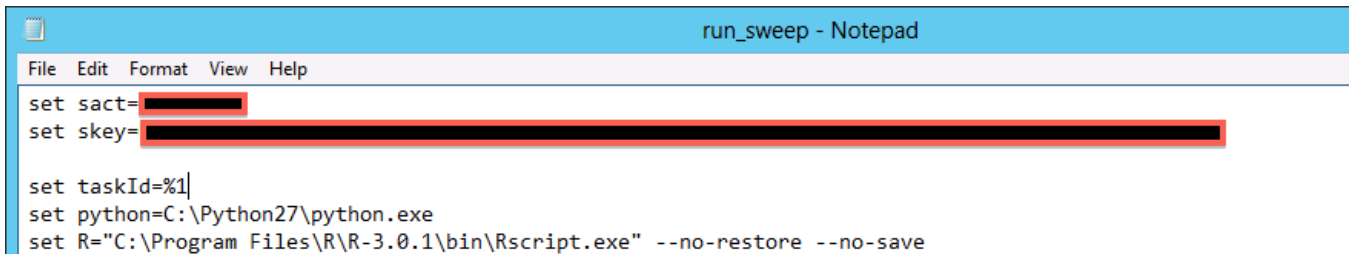
You should see output similar to this:



```
Administrator: Windows PowerShell
PS C:\Users\azureuser> azure storage account keys list [redacted]
info: Executing command storage account keys list
+ Getting storage account keys
data: Primary: [redacted]
data: Secondary: [redacted]
info: storage account keys list command OK
PS C:\Users\azureuser>
```

Highlight your storage key in the console and right-click to copy it to the clipboard.

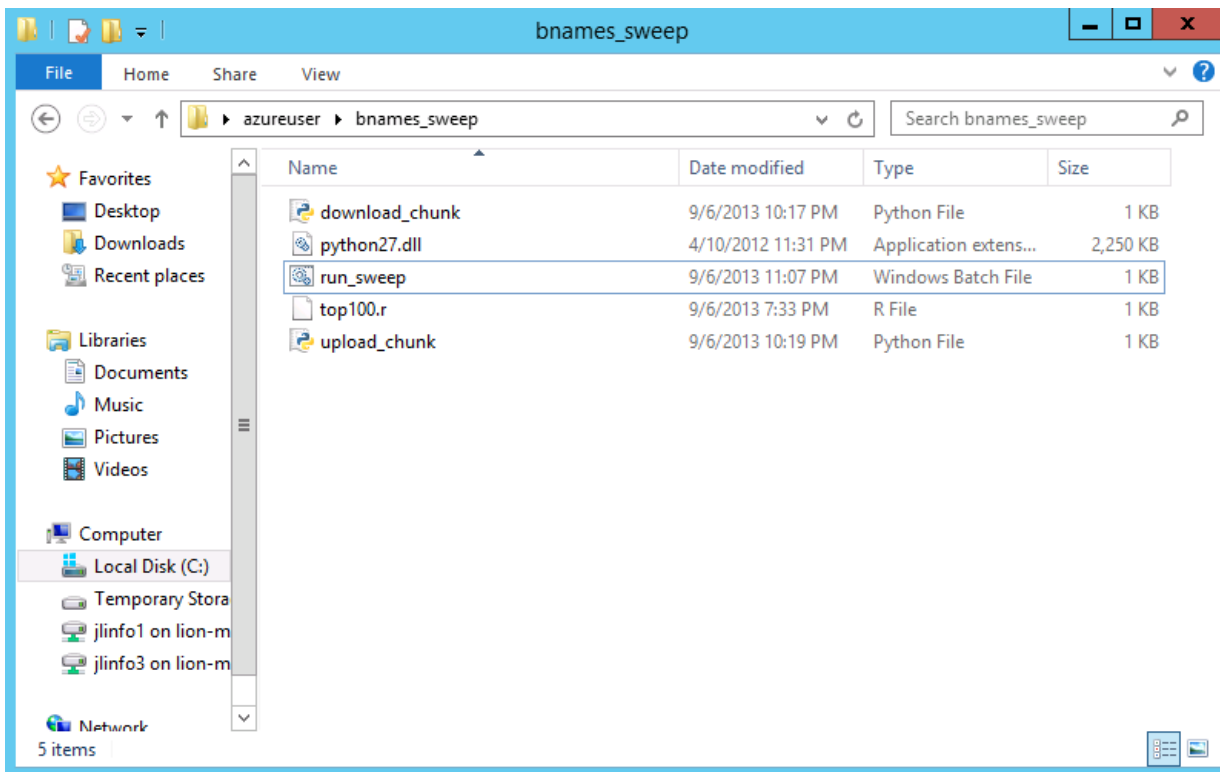
1. Edit **run\_sweep.bat** and set the **sact** and **skey** variables to your storage account name and storage account key. Save the file.



```
run_sweep - Notepad
File Edit Format View Help
set sact=[redacted]
set skey=[redacted]

set taskId=%1
set python=C:\Python27\python.exe
set R="C:\Program Files\R\R-3.0.1\bin\Rscript.exe" --no-restore --no-save
```

1. Copy **C:\Windows\SysWOW64\python27.dll** to the application folder. **bnames\_sweep** should look like this:



1. Open a command line window and navigate to the parent of the **bnames\_sweep** folder. Use **hpcpack** to package and upload the application:

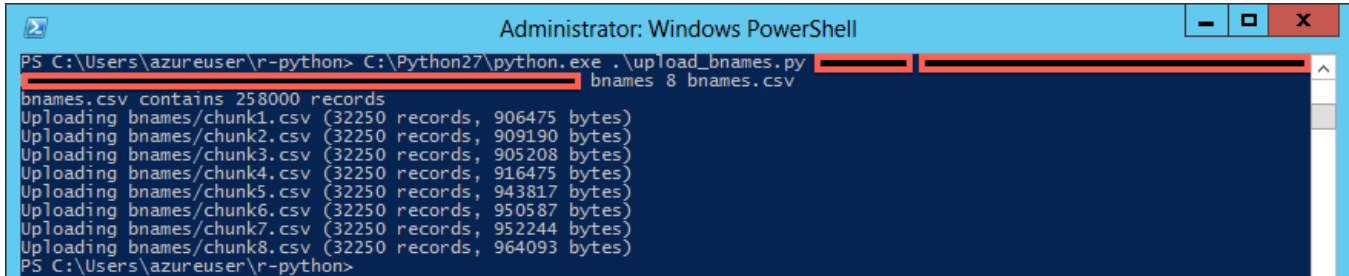
```
hpcpack create bnames_sweep.zip bnames_sweep
hpcpack upload bnames_sweep.zip /nodetemplate:"Default AzureNode Template" /relativePath:bnames_sweep
```

2. Execute **hpcsync** on all cluster nodes via the **clusrun** command. This will download the new package from your storage account to all cluster nodes:

```
clusrun /nodegroup:AzureNodes hpcsync
```

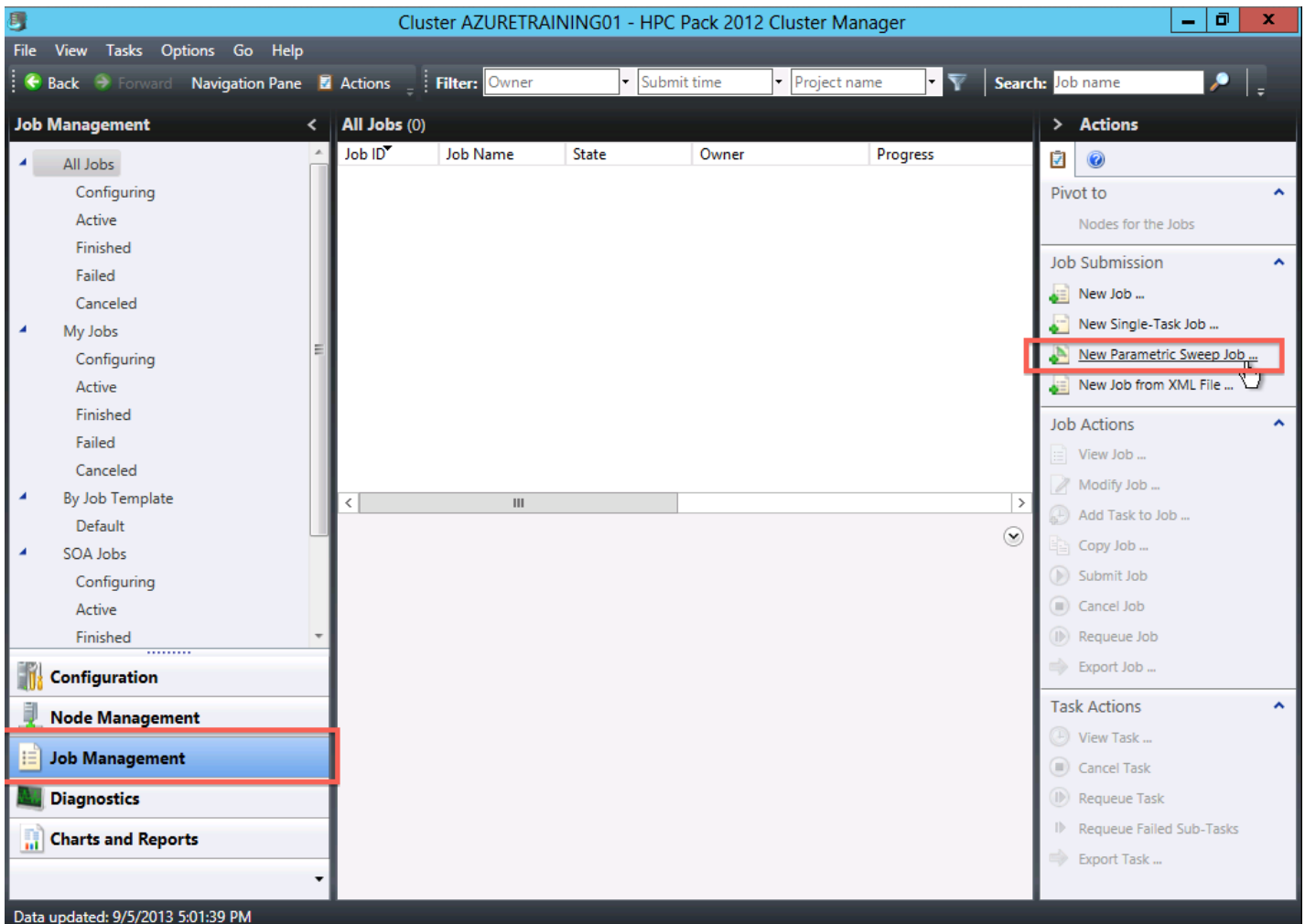
3. We need to upload our data to the storage account so the worker nodes can access it. Open a command prompt window and navigate to the training materials folder. Execute the **upload\_bnames.py** script as shown:

```
C:\Python27\python.exe upload_bnames.py YOUR_ACCOUNT YOUR_KEY bnames 8 bnames.csv
```

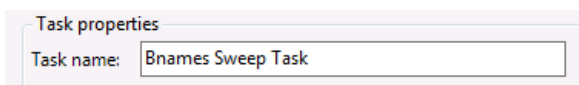


```
Administrator: Windows PowerShell
PS C:\Users\azureuser\r-python> C:\Python27\python.exe .\upload_bnames.py bnames 8 bnames.csv
bnames.csv contains 258000 records
Uploading bnames/chunk1.csv (32250 records, 906475 bytes)
Uploading bnames/chunk2.csv (32250 records, 909190 bytes)
Uploading bnames/chunk3.csv (32250 records, 905208 bytes)
Uploading bnames/chunk4.csv (32250 records, 916475 bytes)
Uploading bnames/chunk5.csv (32250 records, 943817 bytes)
Uploading bnames/chunk6.csv (32250 records, 950587 bytes)
Uploading bnames/chunk7.csv (32250 records, 952244 bytes)
Uploading bnames/chunk8.csv (32250 records, 964093 bytes)
PS C:\Users\azureuser\r-python>
```

1. Open the Cluster Manager on the head node. On the **Job Management** page, select **New Parametric Sweep Job** from the list on the right.



1. Enter **Bnames Sweep Task** as the task name.



The screenshot shows the 'Task properties' dialog box. The 'Task name:' field is filled with the text 'Bnames Sweep Task'.

1. Under Step 1, set the Start Value to **1** and the End Value to **8**. We will use eight parallel jobs since we have eight CPU cores in our cluster (four medium-sized nodes, two CPU cores per node).



Step 1: Select the start and end values for the sweep task:

Start value:    
 End value:

1. Leave the Increment Value as 1.

Step 2: Select the amount to increment the value at each step of the sweep task:

Increment value:

1. In the command line box, enter **run\_sweep.bat \***

The job scheduler replaces the asterisk with sequential integer values in the range specified in the Step 1 section, in this case, the numbers 1, 2, and so on up to 8. You can use this parameterization any way you like. In this example, we're simply using it to specify the taskID. The command line box should look like this:

Step 3: Enter the command line, working directory, and file locations for the sweep task.

Use an asterisk (\*) where the step values should be inserted.

Command line:

1. Set the working directory to **%CCP\_PACKAGE\_ROOT%\bnamesweep**. This folder was created automatically by hpcsync because we specified the /relativePath parameter when we executed our hpcpack upload command.

Working directory:

---

**IMPORTANT:** There is a 10GB limit on %CCPPACKAGEROOT%.

If your job is going to write a lot of data to files in the working directory then you'll need to use a different working directory and write a batch script to copy files from %CCPPACKAGEROOT% before running the job.

---

1. Leave the Standard input, Standard output, and Standard error fields empty.
2. Your "New Parametric Sweep Job" window should look like this with all fields completed. Click **Submit**.

**New Parametric Sweep Job**

**Job properties**  
 Select a job template to use for this job. The job template specifies a set of options to use when running a job.

Job template: Default

Send a notification when this job: ☐ Starts ☐ Completes

Send email notifications to:

**Task properties**

Task name: Bnames Sweep Task

Step 1: Select the start and end values for the sweep task:  
 Start value: 1 End value: 8

Step 2: Select the amount to increment the value at each step of the sweep task:  
 Increment value: 1

Step 3: Enter the command line, working directory, and file locations for the sweep task.  
 Use an asterisk (\*) where the step values should be inserted.

Command line: run\_sweep.bat \*

Working directory: %CCP\_PACKAGE\_ROOT%\bnames\_sweep Browse...

Standard input:  Browse...

Standard output:  Browse...

Standard error:  Browse...

Step 4: Preview your sweep task:

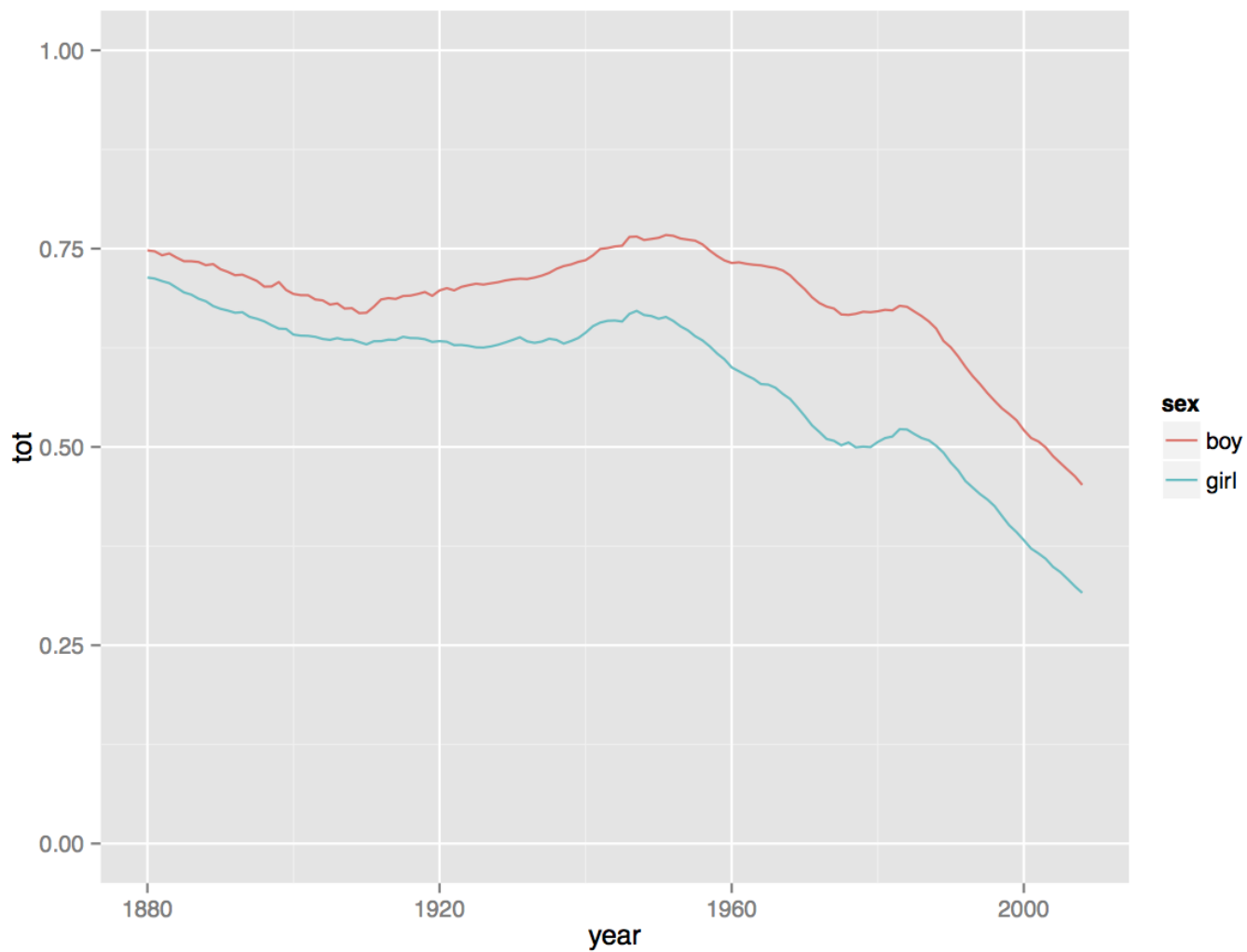
Command Line	Standard Output
run_sweep.bat 1	
run_sweep.bat 2	
...	
run_sweep.bat 8	

Submit Cancel

1. Once the job is finished, we'll need to postprocess the results to get our final answer. Edit **postprocess\_sweep.bat** and set the **sact** and **skey** variables to your storage account name and storage account key. Save the file.
2. Open a command prompt window and navigate to the training materials folder. Execute the **postprocess\_sweep.bat** script:

```
Administrator: Windows PowerShell
PS C:\Users\azureuser\r-python> .\postprocess_sweep.bat 8
C:\Users\azureuser\r-python>set sact=paratools
C:\Users\azureuser\r-python>set skey=KWrMGUp7jxwykPTk7wFLCv18DZfZSmjF+4YMMRRXXPgKFK6s/9negZHNdVD173bYA6t4mWoP9fXra+mHiB0yxQ==
C:\Users\azureuser\r-python>set nTasks=8
C:\Users\azureuser\r-python>set python=C:\Python27\python.exe
C:\Users\azureuser\r-python>set R="C:\Program Files\R\R-3.0.1\bin\Rscript.exe" --no-restore --no-save
C:\Users\azureuser\r-python>for /L %A IN (1 1 8) DO (C:\Python27\python.exe download_chunk.py paratools KWrMGUp7jxwykPTk7wFLCv18DZfZSmjF+4YMMRRXXPgKFK6s/9negZHNdVD173bYA6t4mWoP9fXra+mHiB0yxQ== bnames %A )
C:\Users\azureuser\r-python>(C:\Python27\python.exe download_chunk.py paratools KWrMGUp7jxwykPTk7wFLCv18DZfZSmjF+4YMMRRXXPgKFK6s/9negZHNdVD173bYA6t4mWoP9fXra+mHiB0yxQ== bnames 1 )
C:\Users\azureuser\r-python>(C:\Python27\python.exe download_chunk.py paratools KWrMGUp7jxwykPTk7wFLCv18DZfZSmjF+4YMMRRXXPgKFK6s/9negZHNdVD173bYA6t4mWoP9fXra+mHiB0yxQ== bnames 2 )
C:\Users\azureuser\r-python>(C:\Python27\python.exe download_chunk.py paratools KWrMGUp7jxwykPTk7wFLCv18DZfZSmjF+4YMMRRXXPgKFK6s/9negZHNdVD173bYA6t4mWoP9fXra+mHiB0yxQ== bnames 3 )
C:\Users\azureuser\r-python>(C:\Python27\python.exe download_chunk.py paratools KWrMGUp7jxwykPTk7wFLCv18DZfZSmjF+4YMMRRXXPgKFK6s/9negZHNdVD173bYA6t4mWoP9fXra+mHiB0yxQ== bnames 4 )
C:\Users\azureuser\r-python>(C:\Python27\python.exe download_chunk.py paratools KWrMGUp7jxwykPTk7wFLCv18DZfZSmjF+4YMMRRXXPgKFK6s/9negZHNdVD173bYA6t4mWoP9fXra+mHiB0yxQ== bnames 5 )
C:\Users\azureuser\r-python>(C:\Python27\python.exe download_chunk.py paratools KWrMGUp7jxwykPTk7wFLCv18DZfZSmjF+4YMMRRXXPgKFK6s/9negZHNdVD173bYA6t4mWoP9fXra+mHiB0yxQ== bnames 6 )
C:\Users\azureuser\r-python>(C:\Python27\python.exe download_chunk.py paratools KWrMGUp7jxwykPTk7wFLCv18DZfZSmjF+4YMMRRXXPgKFK6s/9negZHNdVD173bYA6t4mWoP9fXra+mHiB0yxQ== bnames 7 )
C:\Users\azureuser\r-python>(C:\Python27\python.exe download_chunk.py paratools KWrMGUp7jxwykPTk7wFLCv18DZfZSmjF+4YMMRRXXPgKFK6s/9negZHNdVD173bYA6t4mWoP9fXra+mHiB0yxQ== bnames 8 )
C:\Users\azureuser\r-python>"C:\Program Files\R\R-3.0.1\bin\Rscript.exe" --no-restore --no-save postprocess.r 8
Loading required package: methods
PS C:\Users\azureuser\r-python>
```

The final result is written to **bnames-top100.png**:



Copyright 2013 Microsoft Corporation. All rights reserved. Except where otherwise noted, these materials are licensed under the terms of the Apache License, Version 2.0. You may use it according to the license as is most appropriate for your project on a case-by-case basis. The terms of this license can be found in <http://www.apache.org/licenses/LICENSE-2.0>.