set up

> x11 forwarding in PUTTY.

Try to install xming in your windows box

> http://www.straightrunning.com/XmingNotes/. Make sure to run xming

> before plotting

<http://n4.nabble.com/how-to-make-R-plot-under-Linux-tp1562060p1565113.html>

I suppose that the simplest solution (namely fitting "enough" memory on your present Windows machine) is deemed impossible. The fun begins...

Now, to work with R, you need only a \*text\* connection to your server.

It is enough to use function creating graphs ... as files that you can later display on your Windows machine. That's what ssh does : terminal emulation (plus the ability to copy files back and forth, via scp (which you \*will\* need), plus a way to create so-called tunnels and redirections... But that's a horse on an entirely different color (an elephant, actually:)).

If you want "real-time" graphics displayed by the R interpreter, you need, indeed, to use the "-X" switch to ssh. but that is \*not\* \*enough\* : your Windows machine \*must\* be fitted with software accepting commands emitted by the server's R interpreter and obeying them to actually create the image ; that is something called "an X server" (yes,

server : in this case, your windows machine offers a "displaying service", that your R interpreter uses for displaying your graph, thus becoming a client of your server).

Installing such a beast under Windows is (was ?) neither easy nor (usually) cheap.

There \*are\* free (in both meanings of this word) X server implementations for Windows (most notoriously Cygwin/X and Xming), but, as far as I know, none of them is "easy" to install for the uninitiated : to do this, you must understand what you are doing, which

implies (partially) mastering the X window system, which is ... complicated, to put it mildly. You'd better seek \*informed\* help around you on this one.

I am aware of a handful of commercial implementations claiming to be "easy to install", but canot emit any opinion of them : the price tag is enough to give me pause...

Another option (to be discussed with your server's manager) is to display on your Windows machine the image of a "virtual" X session started on the server. Such a solution, which has a couple of implementations (variants of VNC, variants of RDP) might be quite preferable if your network connection is slow/unreliable : X eats bandwidth like there's no tomorrow... I find VNC quite useful on the limited-bandwith connections that I use almost daily.

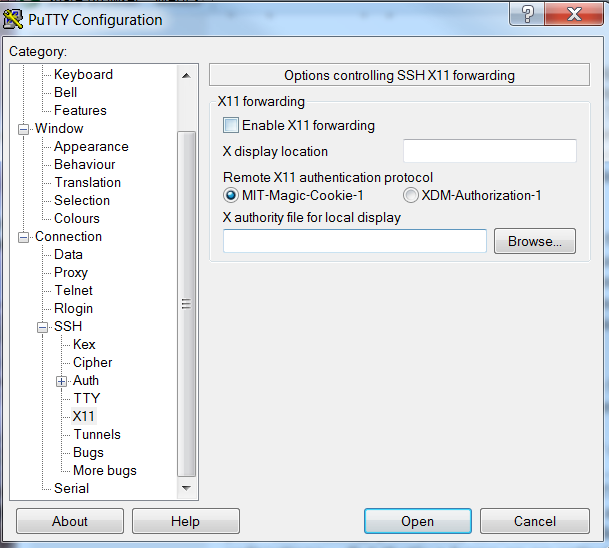
But, may well be that the \*simplest\* solution would be to install Linux on your own machine (dual boot for a first time...) : X is the native windowing interface on this system, and won't have a bit of trouble serving your R client : it comes \*already\* configured to do so.

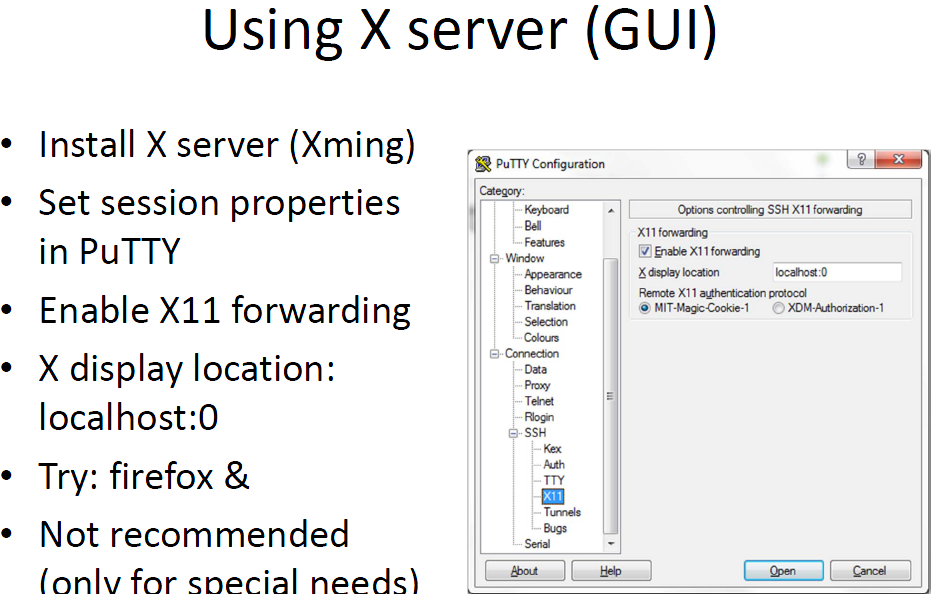
Nowadays, installing Ubuntu is probably \*simpler\* than setting Xming| Cygwin/X up properly. Again, seek local help to discuss this option.

The downside of this solution is that, while your machine runs Ubuntu, it won't run Windows, thus cutting you off the Windows software you already know, which might disrupt your workflow.

The \*upside\* of this solution is \*also\* that, while your machine runs Ubuntu, it won't run Windows, thus cutting you off the Windows software you already know and giving you a big incentive to try alternative, quite possibly better, solutions. That form of accelerated weaning might be more or less traumatic but might \*enormously\* boost your understanding of your tools. After all, Unix has been compared to the Tao for a reason... :-). If your major has anything to do with mathematics, science or engineering|technologies, it would be smart to at least consider this seriously.

In short, you should discuss your options (memory, X server, VNC/RDP, Linux) with your \*local\* friendly help. Again, the R help list is \*definitely\* \*NOT\* the right place for learning about the care and feeding of computers.





<http://stat.ethz.ch/R‐manual/Rpatched/library/utils/html/Rscript.html>

http://www.rackaid.com/resources/linuxscreen‐tutorial‐and‐how‐to/

Please try to setup environment variable DISPLAY on your remote machine.  
  
ex,  
  
csh:  
setenv DISPLAY localhostIPAddress:20.0  
  
or bash  
export DISPLAY=localhostIPAddress:20.0  
  
where localhostIPAddress is your IP address of your current working  
machine (local machine)  
  
Usually, I just use localhostIPAddress:0.0, you can ask your group  
member to get information, which \*.x you should use.

Try to install xming in your windows box  
[http://www.straightrunning.com/XmingNotes/.](http://www.straightrunning.com/XmingNotes/) Make sure to run xming

thank you all your guys. You all show a great deal of tolerance on my  
ignorance. I installed reflection X on windows and problems solved!

Yes, install R so that X11 is enabled, then do  
  
  
x11()  
hist( c(1:10) )  
  
  
and you should see your histogram.

I have connected to the Linux server via ssh -X using Putty and can  
successfully show "xclock" window in Xming.  
However, when i used R and run command "hist(c(1:10))", i cannot get the  
figure in the Xming window (nothing happened in the Xming window).  
The output of capabilities() is as following:

capabilities()

jpeg png tiff tcltk X11 aqua http/ftp sockets  
TRUE TRUE FALSE FALSE FALSE FALSE TRUE TRUE  
libxml fifo cledit iconv NLS profmem cairo  
TRUE TRUE TRUE TRUE TRUE FALSE TRUE  
  
What should i do to see the popup plot of R? should i need to ask the  
Administrator to enable X11?

What is x11 forward?

**X11** is the **X** Window System which allows you to run software on a UNIX/Linux server in a Windows-like way such that you can use your mouse to click around in it. The secure way to do this is to **forward** your **X11** packets through your ssh conection which automatically sets your DISPLAY environment variable for you.

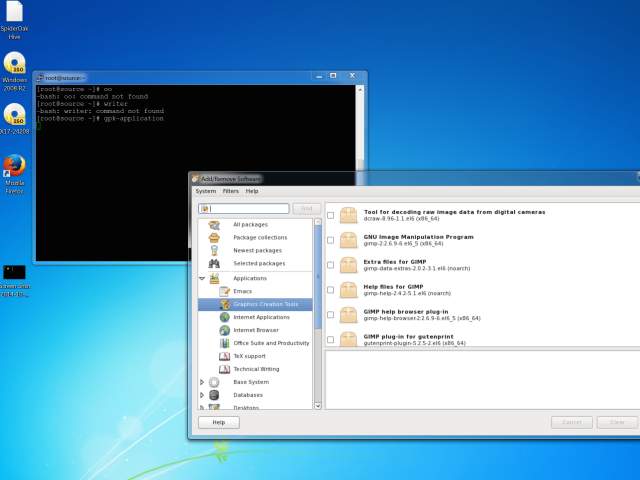
What is the X server in Linux?

An **X server** is a program in the **X** Window System that runs on local machines (i.e., the computers used directly by users) and handles all access to the graphics cards, display screens and input devices (typically a keyboard and mouse) on those

What is x11 for Linux?

**X11** is a network protocol designed for Unix and similar operating systems to enable remote graphical access to applications. The original X windowing system was announced in 1984 and developed at MIT. A machine running an X windowing system can launch a program on a remote computer.

X11 forwarding can be useful when a GUI is required, especially for system and configuration tools that don't have a CLI interface. Here's how to set up and use X11 Forwarding on Linux and Mac.



More experienced users know that you can connect to a Linux server quite easily using a variety of SSH (Secure Shell) clients. That's all well and good but what about if you need to use some tool that has a GUI?

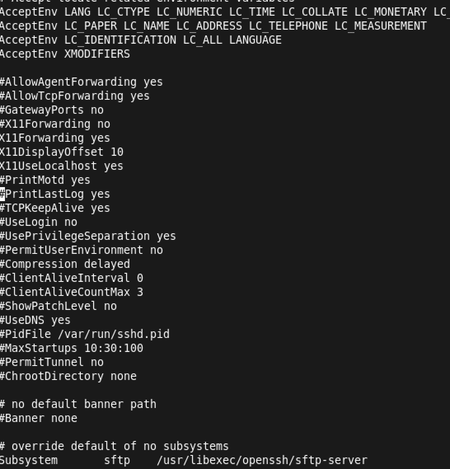
The answer is to use X11 forwarding. X11 forwarding is a mechanism that allows a user to start up remote applications but forward the application display to your local Windows machine.

### **X11 Forwarding on Linux**

There are a few minor changes needed on your Linux box to enable X11 forwarding. In this basic example I am using a standard CentOS 6.5 installation with GNOME installed. Other Linux distributions will have a very similar SSH daemon configuration. The SSH configuration file is located at **/etc/ssh/sshd\_config**. Locate and uncomment (or add) the following lines:

**X11Forwarding yes**

**X11UseForwarding yes**



CentOS ConfigurationThese items allow the traffic to be forwarded over to your computer from the host and also allow forwarding to be used. X11 forwarding can be configured on a per user basis. There are additional options that can be configured for X11 forwarding, but those go beyond the scope of this introductory article. The system can also be configured to allow multiple X11 forwarded users, but each user will have to use a separate port range.

Once the config file's modifications have been made to your Linux system, the SSH service will need to be restarted to effect the revised configuration. Restart the services by using the command:

            service sshd restart

All being well, the sshd service will be fully restarted. As a funny little test, install **"xeyes."** Use the command:

            yum -y install xeyes.

It is also prudent to check that the firewall is not blocking the forwarded requests. **Iptables**' configuration may need to be adjusted if they are. To turn off **Iptables** on a temporary basis, you can use the command:

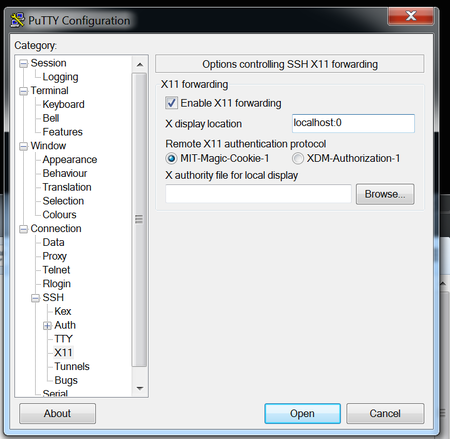
service iptables stop

On the Windows guest, the first requirement is to install **PuTTY**, which provides the mechanism to forward the display data between the host (the Linux server) and guest (your Windows machine). If you don't already have PuTTY, download the PuTTY application from the [PuTTY](http://www.chiark.greenend.org.uk/%7Esgtatham/putty/) website and install it. I'd suggest using the full download package.

There are many display managers you can use. In this example we are using **Xming**, which is free and downloadable from [sourceforge.net](http://sourceforge.net/projects/xming/). Download ***Xming*** to your Windows PC and run the setup. During installation, you might see a number of options; I'd suggest going with the defaults.

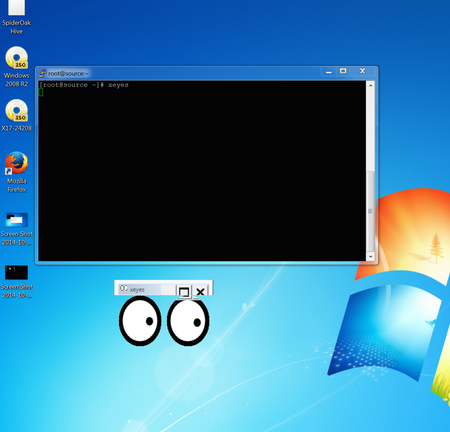
Once Xming is installed you are ready to try out X11 forwarding. You will need to allow the Windows firewall to be modified to allow **Xming** forwarding to work. This should automatically pop up and request the required access.

There are also two settings in PuTTY that need to be changed to make X11 forwarding work. Open PuTTY as normal, filling in the **IP address** or **DNS alias name**. Before clicking OK to connect to the server, go to the connection menu in PuTTY, select **SSH, X11** and tick the box **X11 forwarding.** The xdisplay location should also be set to **localhost:0**, which is basically the display you are working on currently.



PuTTY ConfigurationYou can now connect to the Linux server using PuTTY. At the command line, run the command: **xeyes**. Hopefully a silly pair of comedy eyes will pop up that follow your cursor around the screen. To keep the terminal available for additional commands after you launch the X11 forwarded application, use an **&** at the end of the command to prevent tying up

the terminal in question.



Xeyes example Now that you have seen it in action, it is perhaps time to share how it all works.

No matter which GUI you use on your Linux host, GNOME or KDE, they both have what is termed a xdisplay manager that underlies the GUI portion of the display. It is a network protocol that was designed from the outset to allow items to be forwarded to whichever destination is required.

### **X11 Forwarding on Mac**

X11 forwarding on Mac is just as easy. Download [Quartz](http://xquartz.macosforge.org/landing/) and install it as you would any other Mac package. Quartz is another instance of display management. To start using X11 forwarding to your Mac, use the -X switch. In other words, the ssh command would look like this:

**ssh -X root@mylinuxhost**

To test it just type in the command **xclock** or ***xeyes***, and, as if by magic, you get a basic clock or aforementioned eyes on your desktop.

<http://www.techotopia.com/index.php/Displaying_RHEL_6_Applications_Remotely_%28X11_Forwarding%29>

## Remotely Displaying an RHEL 6 Application

The first step in remotely displaying an application is to move to the system where the application is to be displayed. At this system, ssh into the remote system so that you have a command prompt. This can be achieved using the ssh command. When using the ssh command we need to use the -X flag to tell ssh that we plan to tunnel X traffic through the connection:

ssh -X user@hostname

In the above example username is the user name to use to log into the remote system and hostname is the hostname or IP address of the remote system. Enter your password at the login prompt. Once logged in, run the following command to see the DISPLAY setting:

echo $DISPLAY

The command should output something similar to the following:

localhost:10.0

To display an application simply run it from the command prompt. For example:

gedit

When run, the above command should run the gedit tool on the remote system, but display the output on the local system.

## Trusted X11 Forwarding

If the */etc/ssh/ssh\_config* file on the remote system contains the following line, then it is possible to use trusted X11 forwarding:

ForwardX11Trusted yes

Trusted X11 forwarding is slightly faster than untrusted forwarding since it does not engage the X11 security controls. The -Y flag is needed when using trusted X11 forwarding:

ssh -Y user@hostname

## Compressed X11 Forwarding

When using slower links the X11 data can be compressed using the -C flag:

ssh -X -C user@hostname

# Install Gnome GUI on CentOS 7 / RHEL 7

[Raj](http://www.itzgeek.com/author/admin) August 4, 2014 [37 Comments](http://www.itzgeek.com/how-tos/linux/centos-how-tos/install-gnome-gui-on-centos-7-rhel-7.html#disqus_thread) [centos 7](http://www.itzgeek.com/tag/centos-7), [desktop](http://www.itzgeek.com/tag/desktop), [gui](http://www.itzgeek.com/tag/gui), [rhel 7](http://www.itzgeek.com/tag/rhel-7), [runlevel](http://www.itzgeek.com/tag/runlevel)

Linux admin spends most of his time on working in terminal, there are some who like to work on GUI instead of terminal. By default, CentOS 7 installed as minimal server, user intervention is required to change the installation type. This guide will help you to install GUI on CentOS 7 on the top of the minimal server installation.

Before installing GUI, make a [Local yum repository](http://www.itzgeek.com/how-tos/linux/centos-how-tos/create-local-yum-repository-on-centos-7-rhel-7-using-dvd.html) in order to avoid downloading packages from internet.

**Step 1:** Install Gnome GUI by issuing the following command.

**CentOS 7**

|  |
| --- |
| # yum groupinstall "GNOME Desktop" "Graphical Administration Tools" |

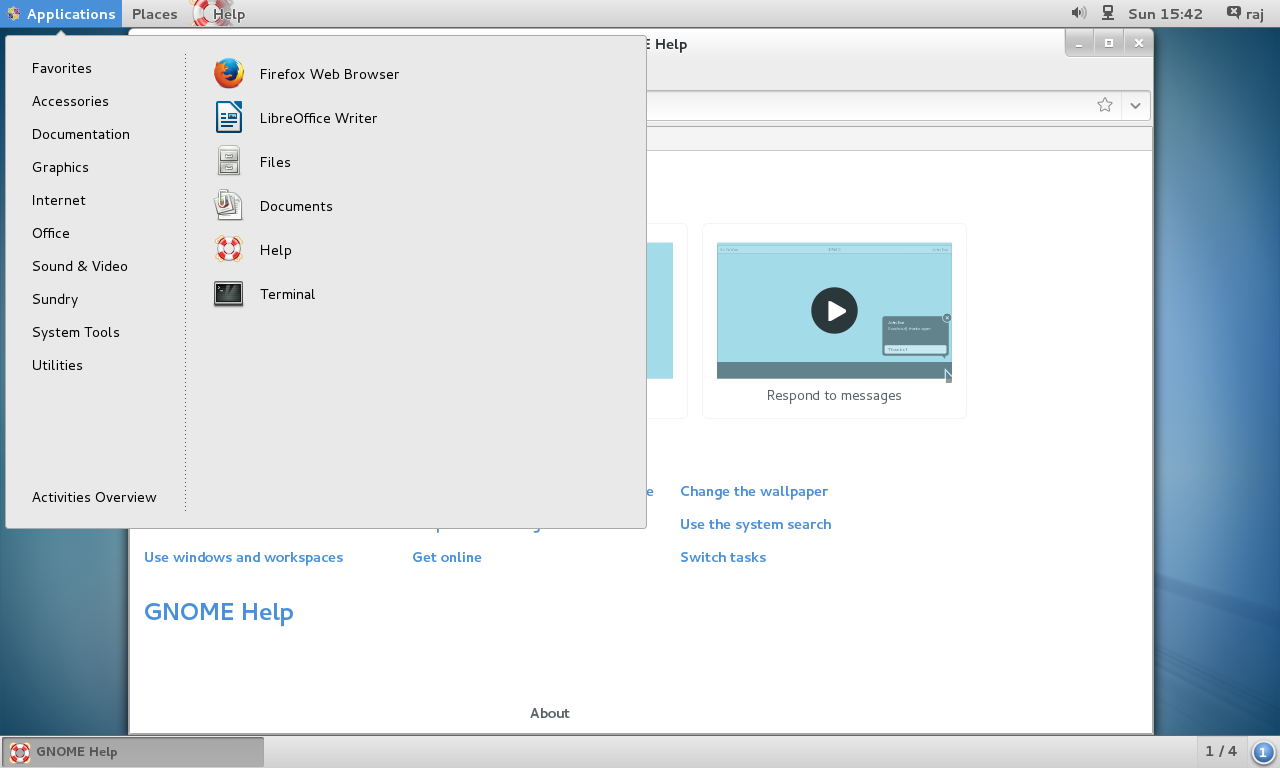
**RHEL 7**

|  |
| --- |
| # yum groupinstall "Server with GUI" |

**Step 2:** Enable GUI on system start up. In CentOS 7,  systemd uses ‘targets’ instead of runlevels; **/etc/inittab** file is no more used to change run levels. Issue the following command to enable the GUI on system start.

|  |
| --- |
| # ln -sf /lib/systemd/system/runlevel5.target /etc/systemd/system/default.target |

Once rebooted, you will get the desktop.

[](http://www.itzgeek.com/wp-content/uploads/2014/08/CentOS-7-GUI-Desktop.png)CentOS 7 – GUI Desktop

That’s All!!!, You have successfully installed GUI on [CentOS 7 / RHEL 7](http://www.itzgeek.com/tag/centos-7).

--

|  |
| --- |
| GNOME Desktop Environment  2014/08/10 |

|  |  |
| --- | --- |
|  | Desktop Environment is not neccessary for Server usage, though. But Sometimes installation or using an application requires Desktop Environment, then build Desktop Environment as follwos. |
| [1] | Install GNOME Desktop Environment on here. |

|  |
| --- |
| [root@dlp ~]# [yum](https://www.server-world.info/en/command/html/yum.html) -y groupinstall "X Window System"  [root@dlp ~]# [yum](https://www.server-world.info/en/command/html/yum.html) -y groupinstall "Desktop"  [root@dlp ~]# 0[yum](https://www.server-world.info/en/command/html/yum.html) -y groupinstall "General Purpose Desktop" |

|  |  |
| --- | --- |
| [2] | Input a command like below after finishing installation of new packages. |

|  |
| --- |
| [root@dlp ~]#  [startx](https://www.server-world.info/en/command/html/startx.html) |

|  |  |
| --- | --- |
| [3] | GNOME Desktop Environments starts like follows. |

|  |
| --- |
| https://www.server-world.info/en/CentOS_6/x/img/1.png |