sudo apt-get update

sudo apt-get remove hplip cups-filters cups hplip-data system-config-printer-udev

sudo apt-get install build-essential tix groff dc axel cups

sudo apt-get install cups-filters unp system-config-printer-gnome

sudo rm -rf /usr/share/hplip

cd /tmp

rm foo\*

axel http://foo2zjs.rkkda.com/foo2zjs.tar.gz

unp foo2zjs.tar.gz

cd foo2zjs/

make

./getweb 1020 # Get HP LaserJet 1020 firmware file

sudo make install

sudo make install-hotplug

# Unplug and re-plug the USB printer into the PC

# add new HP Laserjet 1020 printer via system-config-printer tool

# and choose to use foo2zjs foomatic printer driver :

system-config-printer

# check /var/log/syslog for any errors related to foo2zjs driver:

tail /var/log/syslog

Then power cycle both your PC and your HP printer Then retest printing using the newly installed foo2zjs driver This procedure should solve the hplip driver error “Device Communication Error, code 5012” Source: <https://answers.launchpad.net/hplip/+question/249391>

Similar instructions for installing printer drivers for the HP Laserjet P1005 can be found here:

<https://answers.launchpad.net/ubuntu/+source/hplip/+question/269457>

### Introduction

sudo apt-get update

sudo apt-get install default-jre

sudo apt-get install default-jdk

## Installing the Oracle JDK

sudo add-apt-repository ppa:webupd8team/java

sudo apt-get update

sudo apt-get install oracle-java8-installer

### Oracle JDK 9

sudo apt-get install oracle-java9-installer

## Managing Java

sudo update-alternatives --config java

You can now choose the number to use as a default. This can also be done for other Java commands, such as the compiler (javac), the documentation generator (javadoc), the JAR signing tool (jarsigner), and more. You can use the following command, filling in the command you want to customize.

sudo update-alternatives --config command

## Setting the JAVA\_HOME Environment Variable

sudo update-alternatives --config java

sudo nano /etc/environment

JAVA\_HOME="/usr/lib/jvm/java-8-oracle"

source /etc/environment

echo $JAVA\_HOME

$ python --version

$ sudo apt-get install python3

$ python3 --version

Python 3.6.2

Weka Installation Files

1. Change directory into your Weka installation directory. For example

cd /Applications/weka-3-8-0

|  |  |
| --- | --- |
| 1 | cd /Applications/weka-3-8-0 |

2. Start the Java virtual machine with the weka.jar file, For example:

java -jar weka.jar

|  |  |
| --- | --- |
| 1 | java -jar weka.jar |

Through **Terminal**

* Press Ctrl+Alt+T to open **Terminal**
* Then execute **sudo apt-get update**
* After that, **sudo apt-get install r-base**
* Open **Ubuntu Software Center**
* Search for **r-base**
* And click **Install**
* Then run R by executing **R** in the **Terminal**

|  |
| --- |
|  |
|  |

|  |
| --- |
| # Install R + RStudio on Ubuntu 16.04 |

|  |
| --- |
|  |

|  |
| --- |
| sudo apt-key adv –keyserver keyserver.ubuntu.com –recv-keys E084DAB9 |

|  |
| --- |
|  |

|  |
| --- |
| # Ubuntu 12.04: precise |

|  |
| --- |
| # Ubuntu 14.04: trusty |

|  |
| --- |
| # Ubuntu 16.04: xenial |

|  |
| --- |
| # Basic format of next line deb https://<my.favorite.cran.mirror>/bin/linux/ubuntu <enter your ubuntu version>/ |

|  |
| --- |
| sudo add-apt-repository 'deb https://ftp.ussg.iu.edu/CRAN/bin/linux/ubuntu xenial/' |

|  |
| --- |
| sudo apt-get update |

|  |
| --- |
| sudo apt-get install r-base |

|  |
| --- |
| sudo apt-get install r-base-dev |

|  |
| --- |
|  |

|  |
| --- |
| # Download and Install RStudio |

|  |
| --- |
| sudo apt-get install gdebi-core |

|  |
| --- |
| wget https://download1.rstudio.org/rstudio-1.0.44-amd64.deb |

|  |
| --- |
| sudo gdebi rstudio-1.0.44-amd64.deb |

rm rstudio-1.0.44-amd64.deb

## Step 1 — Installing R

R is a fast-moving project, and the latest stable version isn't always available from Ubuntu's repositories, so we'll start by adding the external repository maintained by CRAN:

Note: CRAN maintains the repositories within their network, but not all external repositories are reliable. Be sure to install only from trusted sources.

**sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys E298A3A825C0D65DFD57CBB651716619E084DAB9**

Output

E298A3A825C0D65DFD57CBB651716619E084DAB9

gpg: requesting key E084DAB9 from hkp server keyserver.ubuntu.com

gpg: key E084DAB9: public key "Michael Rutter <marutter@gmail.com>" imported

gpg: Total number processed: 1

gpg: imported: 1 (RSA: 1)

Once we have the trusted key in each server’s database, we can add the repository.

**sudo add-apt-repository 'deb [arch=amd64,i386] https://cran.rstudio.com/bin/linux/ubuntu xenial/'**

We'll need to run update after this in order to include package manifests from the new repository:

**sudo apt-get update**

Output

. . .

Get:6 https://cran.rstudio.com/bin/linux/ubuntu xenial/ InRelease [3,590 B]

Get:7 https://cran.rstudio.com/bin/linux/ubuntu xenial/ Packages [31.5 kB]

. . .

If the lines above appear in the output from the update command, we've successfully added the repository. We can be sure we won't accidentally install an older version.

Now we're ready to install R:

**sudo apt-get install r-base**

At the time of this writing, the latest stable version from CRAN is at 3.3.1, which is displayed when you start R. Since we’re planning to install the example package for everyone on the system, we’ll start R as the root user so that the libraries will available to all users automatically:

**sudo -i R**

Output

R version 3.3.1 (2016-06-21) -- "Bug in Your Hair"

. . .

Type 'demo()' for some demos, 'help()' for on-line help, or

'help.start()' for an HTML browser interface to help.

Type 'q()' to quit R.

>

This confirms that we’ve successfully installed R and entered its interactive shell.

## Step 2 — Installing R Packages from CRAN

Part of R's strength is the abundance of add-on packages. For demonstration purposes, we'll install txtplot, a library that outputs ASCII graphs, including scatterplot, line plot, density plot, acf and bar charts:

**install.packages('txtplot')**

As part of the installation process, you’ll be given a choice of mirrors to install from:

Output

--- Please select a CRAN mirror for use in this session ---

HTTPS CRAN mirror

1: 0-Cloud [https] 2: Algeria [https]

3: Australia (Melbourne) [https] 4: Australia (Perth) [https]

5: Austria [https] 6: Belgium (Ghent) [https]

7: Brazil (SP 1) [https] 8: Bulgaria [https]

9: Canada (MB) [https] 10: Chile [https]

11: China (Beijing 4) [https] 12: Colombia (Cali) [https]

13: Czech Republic [https] 14: Denmark [https]

15: France (Lyon 1) [https] 16: France (Lyon 2) [https]

17: France (Marseille) [https] 18: France (Paris 2) [https]

19: Germany (Falkenstein) [https] 20: Germany (Münster) [https]

21: Iceland [https] 22: Ireland [https]

23: Italy (Padua) [https] 24: Japan (Tokyo) [https]

25: Malaysia [https] 26: Mexico (Mexico City) [https]

27: New Zealand [https] 28: Norway [https]

29: Philippines [https] 30: Russia (Moscow) [https]

31: Serbia [https] 32: Spain (A Coruña) [https]

33: Spain (Madrid) [https] 34: Switzerland [https]

35: Taiwan (Chungli) [https] 36: Turkey (Denizli) [https]

37: UK (Bristol) [https] 38: UK (Cambridge) [https]

39: UK (London 1) [https] 40: USA (CA 1) [https]

41: USA (IA) [https] 42: USA (KS) [https]

43: USA (MI 1) [https] 44: USA (TN) [https]

45: USA (TX) [https] 46: USA (WA) [https]

47: (HTTP mirrors)

Selection: 1

We’ve entered 1 for 0-Cloud, which will connect us to the Content Delivery Network (CDN) provided by RStudio, in order to get the geographically closest option. This mirror will be set as the default for the remainder of the session. Once you exit R and re-enter, you’ll be prompted to choose a mirror again.

Note: Before the list of mirrors, the following output showed where the package was being installed.

Output

**Installing package into ‘/usr/local/lib/R/site-library’**

(as ‘lib’ is unspecified)

. . .

This site-wide path is available because we ran R as root and is the correct location to make the package available to all users.

When the installation is complete, we can load txtplot:

**library('txtplot')**

If there are no error messages, the library has successfully loaded. Let's see it in action now with an example which demonstrates a basic plotting function with axis labels. The example data, supplied by R's datasets package, contains the speed of cars and the distance required to stop based on data from the 1920s:

**txtplot(cars[,1], cars[,2], xlab = "speed", ylab = "distance")**

Output

+----+-----------+------------+-----------+-----------+--+

120 + \* +

| |

d 100 + \* +

i | \* \* |

s 80 + \* \* +

t | \* \* \* \* |

a 60 + \* \* \* \* \* +

n | \* \* \* \* \* |

c 40 + \* \* \* \* \* \* \* +

e | \* \* \* \* \* \* \* |

20 + \* \* \* \* \* +

| \* \* \* |

0 +----+-----------+------------+-----------+-----------+--+

5 10 15 20 25

speed

If you are interested to learn more about txtplot, use help(txtplot).

GCC

|  |
| --- |
| These commands are based on a askubuntu answer http://askubuntu.com/a/581497 |

|  |
| --- |
| To install gcc-6 (gcc-6.1.1), I had to do more stuff as shown below. |

|  |
| --- |
| USE THOSE COMMANDS AT YOUR OWN RISK. I SHALL NOT BE RESPONSIBLE FOR ANYTHING. |

|  |
| --- |
| ABSOLUTELY NO WARRANTY. |

|  |
| --- |
|  |

|  |
| --- |
| If you are still reading let's carry on with the code. |

|  |
| --- |
|  |

|  |
| --- |
| sudo apt-get update && \ |

|  |
| --- |
| sudo apt-get install build-essential software-properties-common -y && \ |

|  |
| --- |
| sudo add-apt-repository ppa:ubuntu-toolchain-r/test -y && \ |

|  |
| --- |
| sudo apt-get update && \ |

|  |
| --- |
| sudo apt-get install gcc-snapshot -y && \ |

|  |
| --- |
| sudo apt-get update && \ |

|  |
| --- |
| sudo apt-get install gcc-6 g++-6 -y && \ |

|  |
| --- |
| sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-6 60 --slave /usr/bin/g++ g++ /usr/bin/g++-6 && \ |

|  |
| --- |
| sudo apt-get install gcc-4.8 g++-4.8 -y && \ |

|  |
| --- |
| sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.8 60 --slave /usr/bin/g++ g++ /usr/bin/g++-4.8; |

|  |
| --- |
|  |

|  |
| --- |
| When completed, you must change to the gcc you want to work with by default. Type in your terminal: |

|  |
| --- |
| sudo update-alternatives --config gcc |

|  |
| --- |
|  |

|  |
| --- |
| To verify if it worked. Just type in your terminal |

|  |
| --- |
| gcc -v |

|  |
| --- |
|  |

|  |
| --- |
| If everything went fine you should see gcc 6.1.1 by the time I am writing this gist |

|  |
| --- |
|  |

|  |
| --- |
| Happy coding! |

sudo apt-get update && \

sudo apt-get install build-essential software-properties-common -y && \

sudo add-apt-repository ppa:ubuntu-toolchain-r/test -y && \

sudo apt-get update && \

sudo apt-get install gcc-snapshot -y && \

sudo apt-get update && \

sudo apt-get install gcc-6 g++-6 -y && \

sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-6 60 --slave /usr/bin/g++ g++ /usr/bin/g++-6 && \

I just installed Ubuntu 16.04 and want to install StarUML. I downloaded StarUML from <http://staruml.io/download/release/v2.0.2/StarUML-v2.0.2-64-bit.deb>, then in the terminal:

First download libgcrypt package: <https://launchpad.net/ubuntu/+archive/primary/+files/libgcrypt11_1.5.3-2ubuntu4.2_amd64.deb>

Then install it:

sudo dpkg -i libgcrypt11\_1.5.3-2ubuntu4.2\_amd64.deb

Finally, install StarUML:

sudo dpkg -i StarUML-v2.7.0-64-bit.deb

dpkg -i StarUML-v2.0.2-64-bit.deb

The StarUML package has a couple of dependencies:

Depends: gconf-service,

libasound2 (>= 1.0.23),

libatk1.0-0 (>= 1.12.4),

libc6 (>= 2.11),

libcairo2 (>= 1.6.0),

libcups2 (>= 1.4.0),

libdbus-1-3 (>= 1.2.14),

libexpat1 (>= 1.95.8),

libfontconfig1 (>= 2.8.0),

libfreetype6 (>= 2.3.9),

libgcc1 (>= 1:4.1.1),

libgconf-2-4 (>= 2.31.1),

libgcrypt11 (>= 1.4.5),

libgdk-pixbuf2.0-0 (>= 2.22.0),

libglib2.0-0 (>= 2.18.0),

libgtk2.0-0 (>= 2.24.0),

libnspr4 (>= 1.8.0.10),

libnss3 (>= 3.12.6),

libpango1.0-0 (>= 1.22.0),

libstdc++6 (>= 4.6),

libudev0 (>= 147) | libudev1 (>= 198),

libx11-6 (>= 2:1.4.99.1),

libxcomposite1 (>= 1:0.3-1),

libxdamage1 (>= 1:1.1),

libxext6, libxfixes3,

libxrandr2 (>= 2:1.2.0),

libxrender1,

ca-certificates,

libcurl3,

lsb-base (>= 3.2),

xdg-utils (>= 1.0.2),

wget

## Install StarUML

1. Download StarUML (x64)

PowerShell

wget http://staruml.io/download/release/v2.8.0/StarUML-v2.8.0-64-bit.deb

|  |  |
| --- | --- |
| 1 | wget http://staruml.io/download/release/v2.8.0/StarUML-v2.8.0-64-bit.deb |

There is also 32-bit version which you can get with the command below:

PowerShell

wget http://staruml.io/download/release/v2.8.0/StarUML-v2.8.0-32-bit.deb

|  |  |
| --- | --- |
| 1 | wget http://staruml.io/download/release/v2.8.0/StarUML-v2.8.0-32-bit.deb |

2. Run dpkg to install it

sudo dpkg -i StarUML-v2.0.2-64-bit.deb

|  |  |
| --- | --- |
| 1 | sudo dpkg -i StarUML-v2.0.2-64-bit.deb |

You shouldn’t have any problems with this installation if you choose the right version for your system.

StarUML is located under Programming in Gnome application menu

For more information about StarUML, visit the [official website](http://staruml.io/).

**Download/Install OS-Uninstaller in Ubuntu :**  
either add ‘ppa:yannubuntu/boot-repair’ to your Software Sources via the Software Centre or, for speeds-sake, add it using a new Terminal session:  
  
\* sudo add-apt-repository ppa:yannubuntu/boot-repair  
\* sudo apt-get update && sudo apt-get install -y os-uninstaller  
  
  
OS-Uninstaller can be installed & used from any Ubuntu session (normal session, or live-CD, or live-USB).  
  
PPA packages are available for current versions of Ubuntu.  
  
**Use OS-Uninstaller:**  
Launch it from System->Administration->OS-Uninstaller menu if you use Gnome, or search "os" in the dash if you use Unity. Then follow the menus...

bash /home/ml-22/Desktop/Visual\_Paradigm\_15.0/Visual\_Paradigm