

Installation guide for R and RStudio

Step 1 – Install R

1. Download the R installer from <https://cran.r-project.org/>



CRAN
Mirrors
What's new?
Task Views
Search

About R
R Homepage
The R Journal

Software
R Sources
R Binaries
Packages
Other

Documentation
Manuals
FAQs
Contributed

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2015-06-18, World-Famous Astronaut) [R-3.2.1 tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features](#) and [bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

What are R and CRAN?

Figure 1. Screenshot of <http://cran.csiro.au/>

2. Run the installer. Default settings are fine. If you do not have admin rights on your laptop, then ask you local IT support. In that case, it is important that you also ask them to give you full permissions to the R directories. Without this, you will not be able to install additional packages later

Step 2 – Install RStudio

1. Download RStudio: <https://www.rstudio.com/products/rstudio/download/>

RStudio Desktop 0.99.467 — Release Notes

RStudio requires R 2.11.1 (or higher). If you don't already have R, you can download it here.

Installers for Supported Platforms

Installers	Size	Date	MD5
RStudio 0.99.467 - Windows Vista/7/8	73.9 MB	2015-07-15	5c0bf6987adcfb6dd441326ecc67f6e8
RStudio 0.99.467 - Mac OS X 10.6+ (64-bit)	56.2 MB	2015-07-15	3116a0f3b9b3779b9531e9b08c394558
RStudio 0.99.467 - Ubuntu 12.04+/Debian 8+ (32-bit)	77.4 MB	2015-07-15	0ca919255495cc87112df12a1c0ff7e29
RStudio 0.99.467 - Ubuntu 12.04+/Debian 8+ (64-bit)	83.9 MB	2015-07-15	dd64fc165de55a0be229f2362cd776da
RStudio 0.99.467 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	76.8 MB	2015-07-15	1e152baf8bb6c5355a2ec0f6822abdf
RStudio 0.99.467 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	77.6 MB	2015-07-15	a82a27b113184e1790ec5bd3c36e2137

Zip/Tarballs

Zip/tar archives	Size	Date	MD5
RStudio 0.99.467 - Windows Vista/7/8	105.5 MB	2015-07-15	b88bbbc042aaa04b83a1786322981c5f
RStudio 0.99.467 - Ubuntu 12.04+/Debian 8+ (32-bit)	78.1 MB	2015-07-15	171de4b2e08edd33988ab1b75d571ae6
RStudio 0.99.467 - Ubuntu 12.04+/Debian 8+ (64-bit)	84.8 MB	2015-07-15	85ca4bc15c2e5e535594bcf586604324
RStudio 0.99.467 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	77.4 MB	2015-07-15	6abc51b45cdd1fa0c14a991b5336a8ad
RStudio 0.99.467 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	78.4 MB	2015-07-15	9d693e5e5ed1b2003e1b18691f31574b

Source Code

A tarball containing source code for RStudio v0.99.467 can be downloaded from [here](#)

Figure 2. Download RStudio on <https://www.rstudio.com/products/rstudio/download/>

2. Once the installation of R has completed successfully (and not before), run the RStudio installer.
3. If you do not have administrative rights on your laptop, step 2 may fail. Ask your IT Support or download a pre---built zip archive of RStudio which doesn't need installing. The link for this is towards the bottom of the download page, highlighted in Image 2.
 - a. Download the appropriate archive for your system (Windows/Linux only – the Mac version can be installed into your personal “Applications” folder without admin rights).
 - b. Double clicking on the zip archive should automatically unpack it on most Windows machines.

Step 3 – Check that R and RStudio are working

1. Open RStudio. It should open a window that looks similar to image 3 below.
2. In the left hand window, by the '>' sign, type '4+5' (without the quotes) and hit enter. An output line reading '[1] 9' should appear. This means that R and RStudio are working.
3. If this is not successful, contact us or your local IT support for further advice

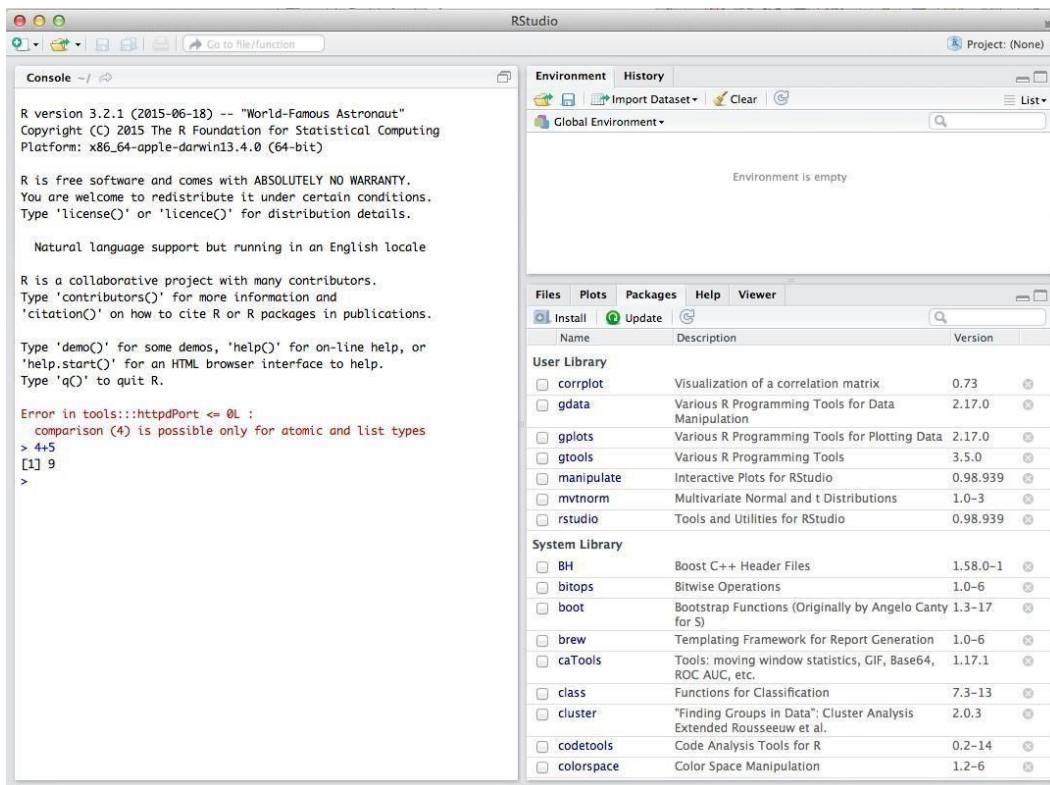


Figure 3. Running R with RStudio

Step 4 – Install R packages required for the workshop

1. Click on the tab ' Packages' then 'Install' as shown in Image 4. Or Tools ---> Install packages.
2. Install the following packages: mixOmics **version 6.1.0**, mvtnorm, RColorBrewer, corrrplot, igraph (see Image 4). For apple mac users, if you are unable to install the mixOmics imported library rgl, you will need to install the XQuartz software first <https://www.xquartz.org/>
3. Check that the packages are installed by typing 'library(mixOmics)' (without the quotes) in the prompt and press enter (see Image 5).
4. Then type 'sessionInfo()' and check that mixOmics version 6.1.0 has been installed (image 6).

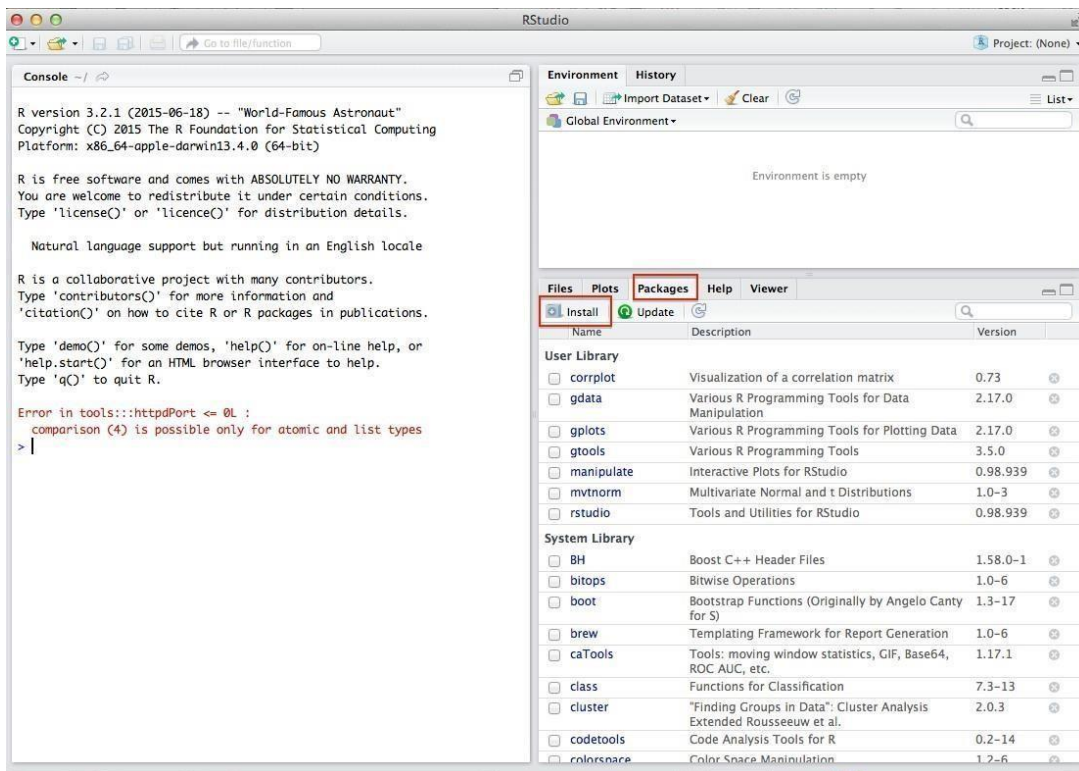


Figure 4. Click on Install to install R packages.

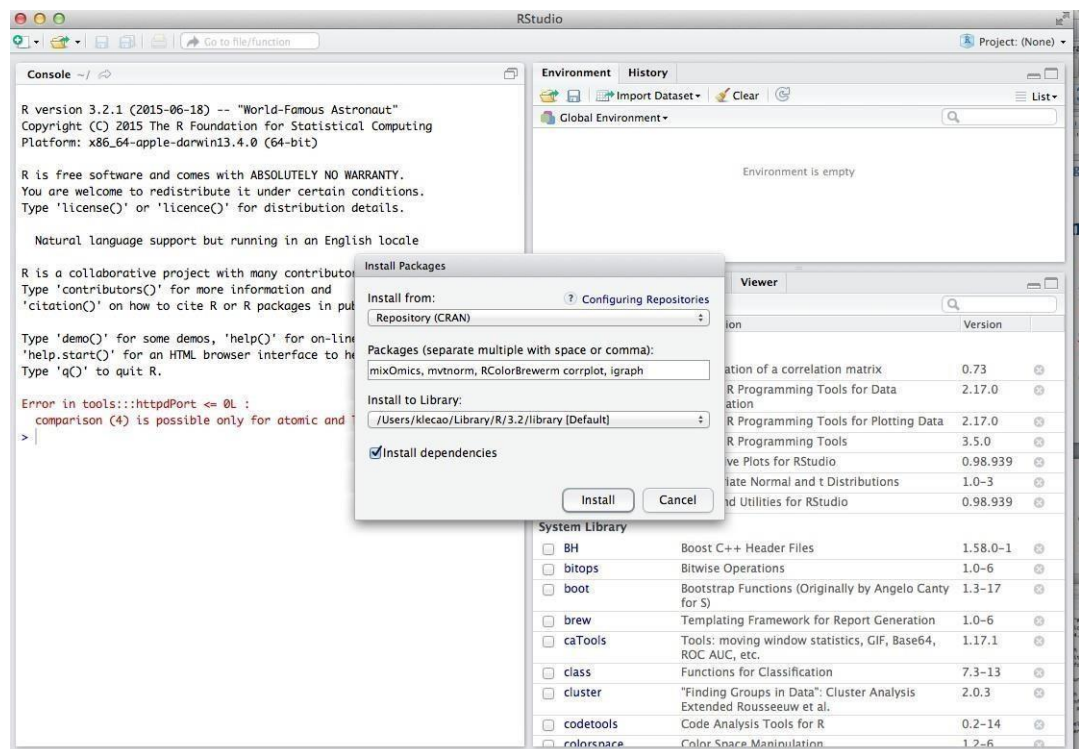


Figure 5. Specify the list of packages to be installed

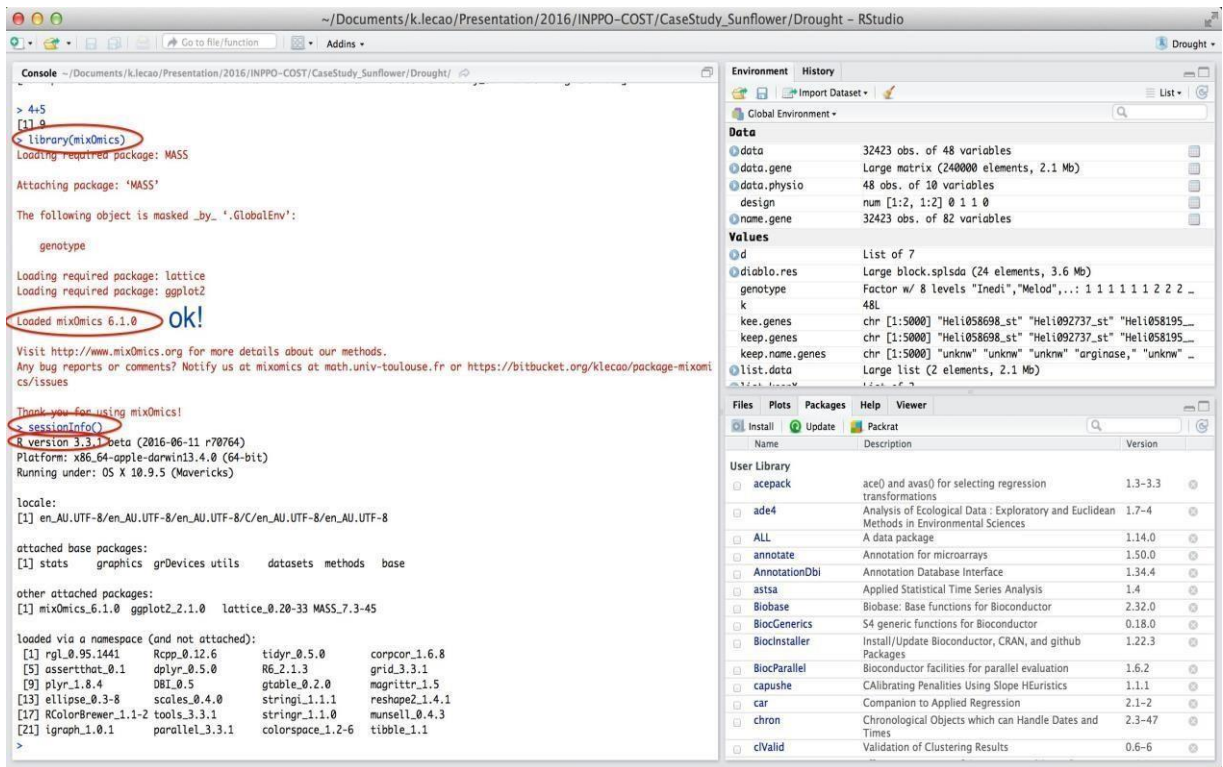


Figure 6. Check that the package mixOmics is installed and has the version 6.1.0.

Exp:7

ROLL NO: 210701268

EXP 7: Implement Linear and Logistic Regression

Aim: To implement linear and logistic regression in RStudio using R language.

PROCEDURE:

a)Linear regression

```
# Sample data heights <- c(150, 160, 165, 170, 175,
180, 185) weights
<- c(55, 60, 62, 68, 70, 75, 80)

# Create a data frame data <- data.frame(heights,
weights)

# Fit a linear regression model linear_model <- lm(weights
~ heights, data = data)

# Print the summary of the model
print(summary(linear_model))

# Plotting the data and regression line plot(data$heights,
data$weights,
  main = "Linear Regression: Weight vs. Height",
xlab = "Height (cm)", ylab = "Weight (kg)",
  pch = 19, col = "blue")

# Add regression line
abline(linear_model, col = "red", lwd = 2)
```



```

1 # Sample data
2 heights <- c(150, 160, 165, 170, 175, 180, 185)
3 weights <- c(55, 60, 62, 68, 70, 75, 80)
4 # Create a data frame
5 data <- data.frame(heights, weights)
6 # Fit a linear regression model
7 linear_model <- lm(weights ~ heights, data = data)
8 # Print the summary of the model
9 print(summary(linear_model))
10 # Plotting the data and regression line
11 plot(data$heights, data$weights,
12      main = "Linear Regression: Weight vs. Height",
13      xlab = "Height (cm)",
14      ylab = "Weight (kg)",
15      pch = 19, col = "blue")
16 # Add regression line
17 abline(linear_model, col = "red", lwd = 2)
18
19

```

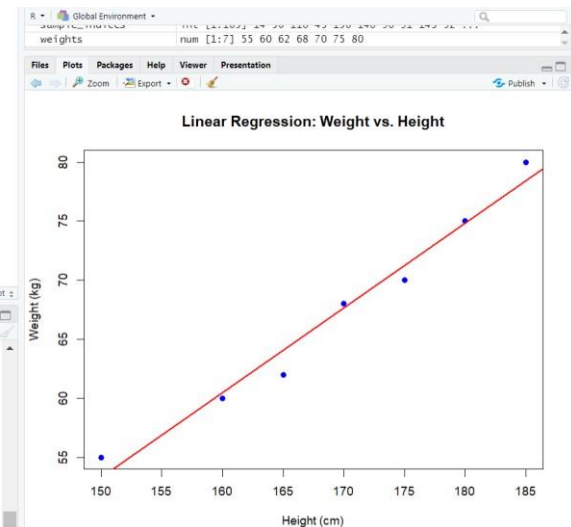
```

R 4.4.1 - ~/
Residuals:
    1      2      3      4      5      6      7
1.7049 -0.4754 -2.0656  0.3443 -1.2459  0.1639  1.5738

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -54.40984    8.74376  -6.223  0.00157 **
heights       0.71803    0.05154  13.932 3.42e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.521 on 5 degrees of freedom
Multiple R-squared:  0.9749,    Adjusted R-squared:  0.9699
F-statistic: 194.1 on 1 and 5 DF,  p-value: 3.424e-05
> |

```



b) Logistic regression

Load the dataset

```
data(mtcars)
```

```
# Convert 'am' to a factor (categorical variable) mtcars$am <- factor(mtcars$am,
levels = c(0, 1), labels = c("Automatic", "Manual"))
```

```
# Fit a logistic regression model logistic_model <- glm(am ~ mpg, data
= mtcars, family = binomial)
```

Print the summary of the model

```
print(summary(logistic_model))
```

```
# Predict probabilities for the logistic model predicted_probs
<- predict(logistic_model, type = "response")
```

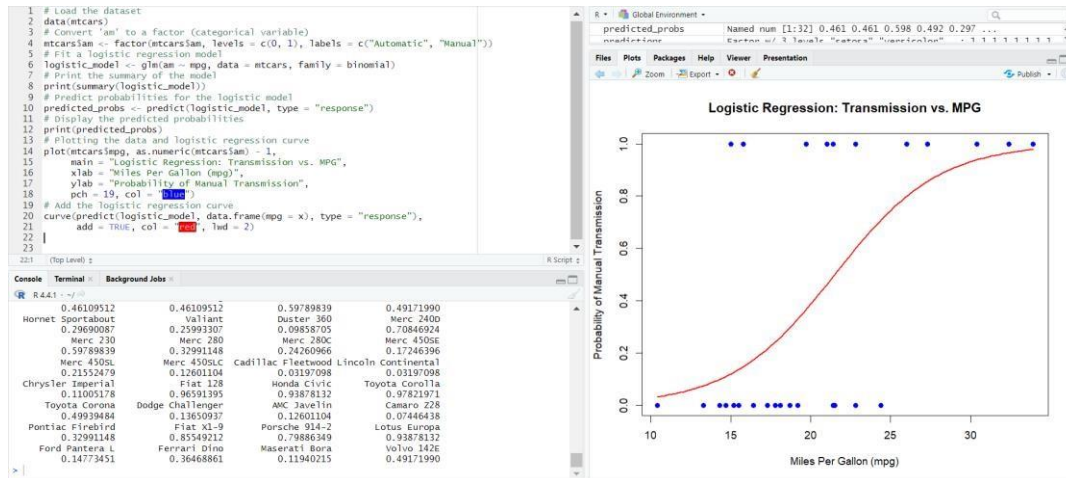
```
# Display the predicted probabilities print(predicted_probs)
```

```
# Plotting the data and logistic regression curve
plot(mtcars$mpg, as.numeric(mtcars$am) - 1, main =
"Logistic Regression: Transmission vs. MPG", xlab =
"Miles Per Gallon (mpg)", ylab = "Probability of
Manual Transmission", pch = 19, col = "blue")
```

Add the logistic regression curve

```
curve(predict(logistic_model, data.frame(mpg = x), type = "response"),
```

add = TRUE, col = "red", lwd = 2)



Result: Thus linear and logistic regression in RStudio is successfully implemented using R language.