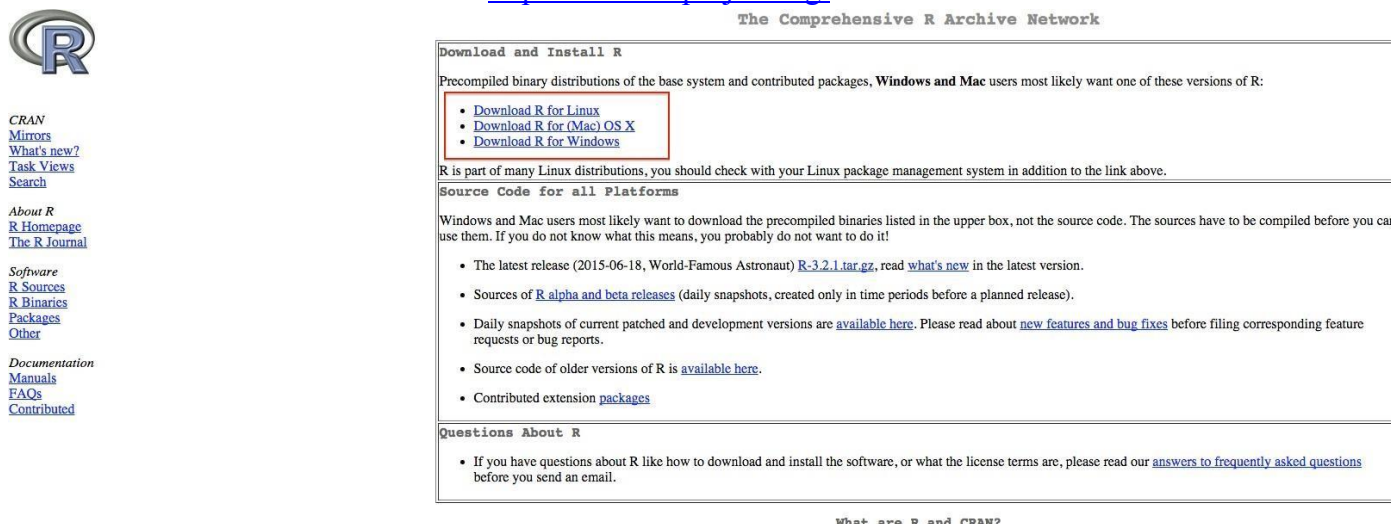


## Installation guide for R and RStudio

### Step 1 – Install R

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



**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**

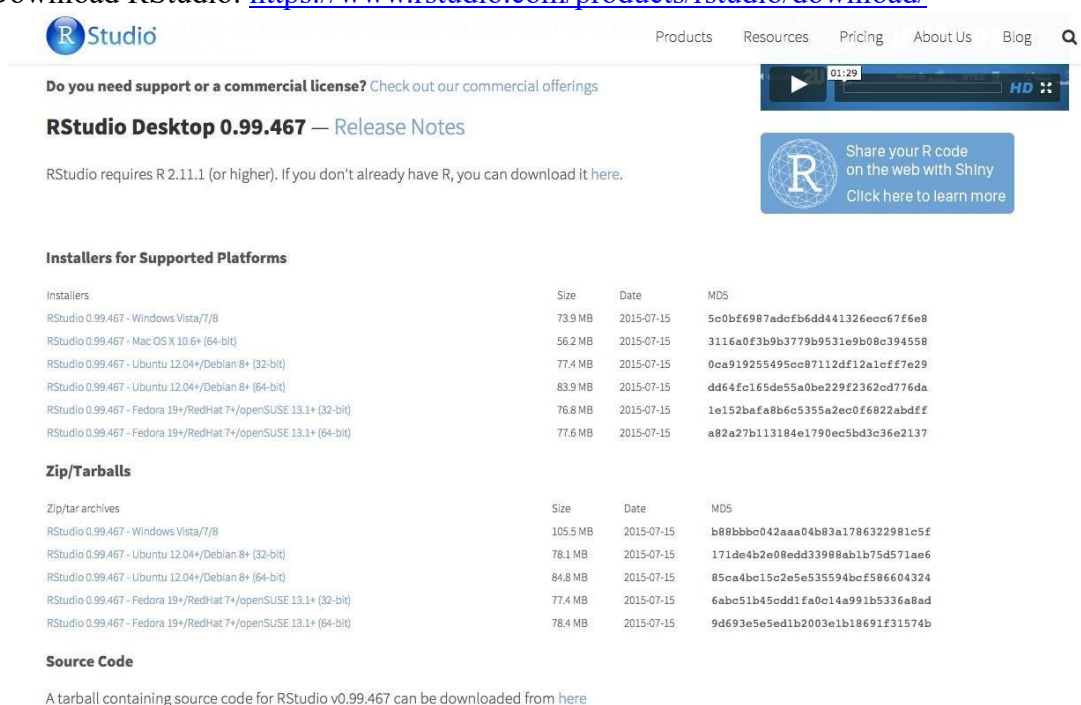
- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

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	5c0bf6987adcfb6dd441326ccc67f6e8
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	0ca91925495cc87112df12a1cfff7e29
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	1e152bafa8b6c5355a2ec0f6822abdff
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	171de4b2e0edd33988ab1b75d571ae6
RStudio 0.99.467 - Ubuntu 12.04+/Debian 8+ (64-bit)	84.8 MB	2015-07-15	85ca4bc15c2e5e535594bcf586660432a
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

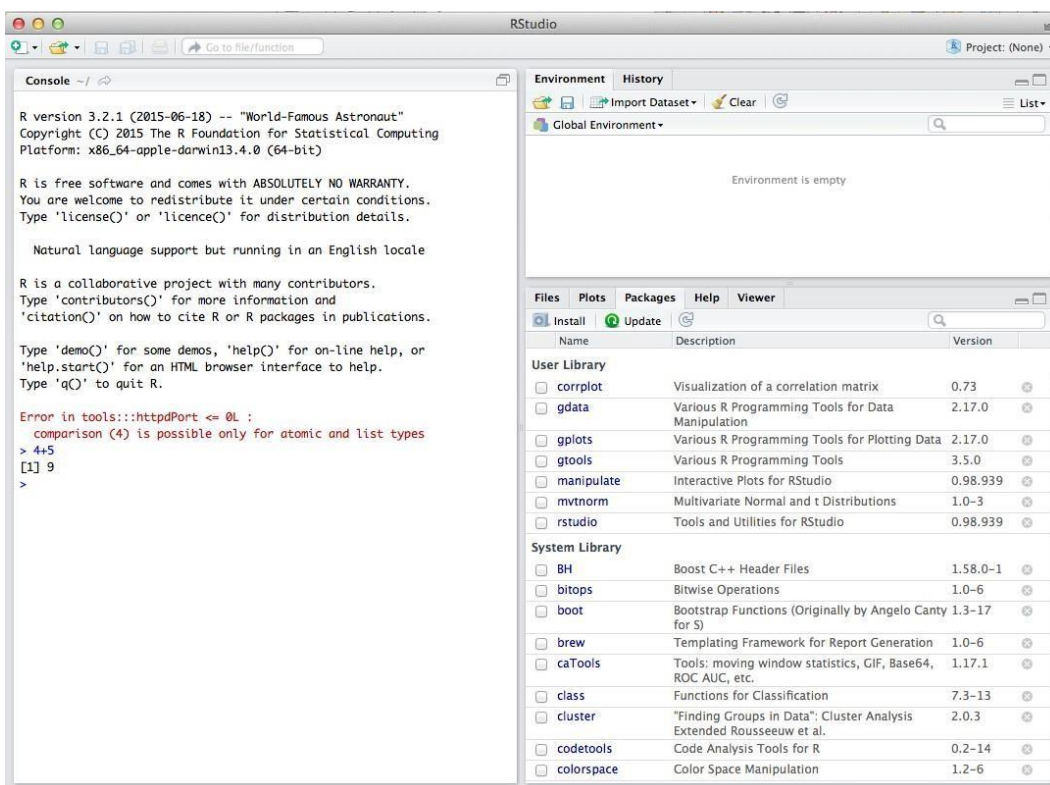


Figure 3. Running R with RStudio

### 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, corrplot, 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).

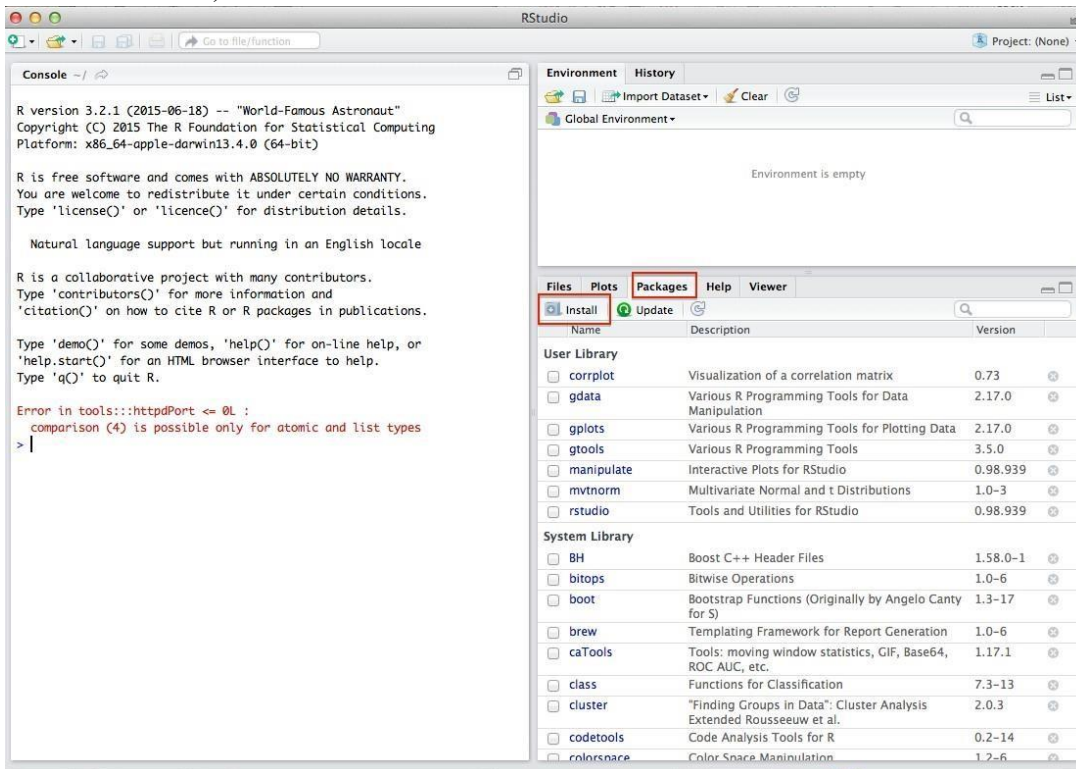


Figure 4. Click on Install to install R packages.

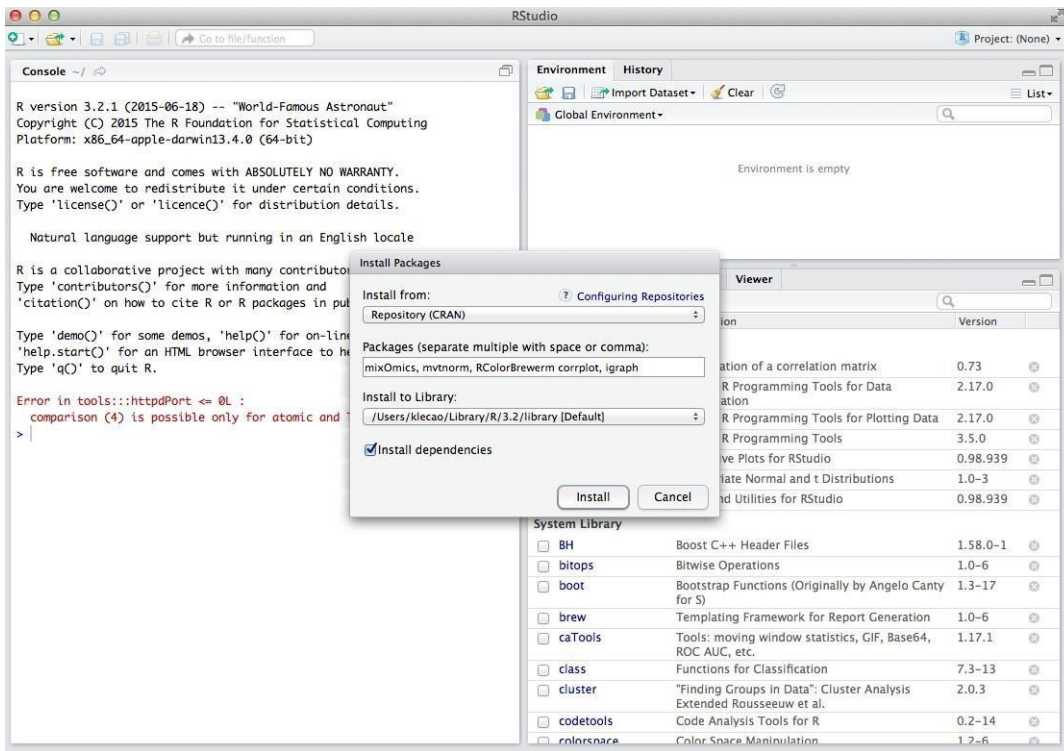


Figure 5. Specify the list of packages to be installed

The screenshot shows the RStudio interface with the following components:

- Console:**

```

> 4+5
[1] 9
> library(mixOmics)
Loading required package: MASS
Attaching package: 'MASS'
The following object is masked _by_ '.GlobalEnv':
  genotype

Loading required package: lattice
Loading required package: ggplot2
Loaded mixOmics 6.1.0 ok!

Visit http://www.mixOmics.org for more details about our methods.
Any bug reports or comments? Notify us at math.univ-toulouse.fr or https://bitbucket.org/klecao/package-mixomics/issues

Thank you for using mixOmics!
> sessionInfo()
R version 3.3.1 beta (2016-06-11 r70764)
Platform: x86_64-apple-darwin13.4.0 (64-bit)
Running under: OS X 10.9.5 (Mavericks)

locale:
 [1] en_AU.UTF-8/en_AU.UTF-8/en_AU.UTF-8/C/en_AU.UTF-8/en_AU.UTF-8

attached base packages:
[1] stats      graphics  grDevices  utils      datasets  methods   base

other attached packages:
[1] mixOmics_6.1.0  ggplot2_2.1.0  lattice_0.20-33 MASS_7.3-45

loaded via a namespace (and not attached):
 [1] rgl_0.95.1441  Rcpp_0.12.6    tidy_0.5.0     corpcor_1.6.8
 [5] assertthat_0.1 dplyr_0.5.0    R6_2.1.3       grid_3.3.1
 [9] plyr_1.8.4     DBI_0.5        gtable_0.2.0   magrittr_1.5
[13] ellipse_0.3-8  scales_0.4.0   stringi_1.1.1  reshape2_1.4.1
[17] RColorBrewer_1.1-2 tools_3.3.1    stringr_1.1.0  munsell_0.4.3
[21] igraph_1.0.1   parallel_3.3.1 colorspace_1.2-6 tibble_1.1

```
- Environment:**
  - Data:**
    - data: 32423 obs. of 48 variables
    - data.gene: Large matrix (240000 elements, 2.1 Mb)
    - data.physio: 48 obs. of 10 variables
    - design: num [1:2, 1:2] 0 1 1 0
    - name.gene: 32423 obs. of 82 variables
  - Values:**
    - d: List of 7
    - diablo.res: Large block.splsda (24 elements, 3.6 Mb)
    - genotype: Factor w/ 8 levels "Inedi","Melod",...: 1 1 1 1 1 2 2 2 ...
    - k: 48L
    - kee.genes: chr [1:5000] "Heli058698\_st" "Heli092737\_st" "Heli058195\_..."
    - keep.genes: chr [1:5000] "Heli058698\_st" "Heli092737\_st" "Heli058195\_..."
    - keep.name.genes: chr [1:5000] "unknw" "unknw" "unknw" "arginase," "unknw" ...
    - list.data: Large list (2 elements, 2.1 Mb)
- Packages:**

Name	Description	Version
acepack	ace() and avas() for selecting regression transformations	1.3-3.3
ade4	Analysis of Ecological Data: Exploratory and Euclidean Methods in Environmental Sciences	1.7-4
ALL	A data package	1.14.0
annotate	Annotation for microarrays	1.50.0
AnnotationDbi	Annotation Database Interface	1.34.4
astsa	Applied Statistical Time Series Analysis	1.4
Biobase	Biobase: Base functions for Bioconductor	2.32.0
BiocGenerics	S4 generic functions for Bioconductor	0.18.0
BiocInstaller	Install/Update Bioconductor, CRAN, and github Packages	1.22.3
BiocParallel	Bioconductor facilities for parallel evaluation	1.6.2
capushe	CAIibrating Penalties Using Slope HEuristics	1.1.1
car	Companion to Applied Regression	2.1-2
chron	Chronological Objects which can Handle Dates and Times	2.3-47
clValid	Validation of Clustering Results	0.6-6

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

**Exp.No: 7****IMPLEMENT LINEAR AND LOGISTIC REGRESSION****AIM:**

To write an R code to implement linear and logistic regression.

**PROCEDURE:**

1. Create sample data for heights and weights, fit a linear regression model, and plot the data with the regression line.
2. Use the sample data to create a data frame for the regression model.
3. Fit the linear regression model using the `lm()` function and display the summary.
4. Plot the data points and add the regression line using the `plot()` and `abline()` functions.
5. Load the `mtcars` dataset, convert the 'am' variable to a factor, fit a logistic regression model using the `glm()` function, and plot the probabilities.

**PROGRAM CODE:****a)Linear regression**

```
# Linear Regression
```

```
heights <- c(150, 160, 165, 170, 175, 180, 185)
```

```
weights <- c(55, 60, 62, 68, 70, 75, 80)
```

```
data <- data.frame(heights, weights)
```

```
linear_model <- lm(weights ~ heights, data = data)
```

```
print(summary(linear_model))
```

```
# Plotting Linear Regression
```

```
plot(data$heights, data$weights,
```

```
  main = "Linear Regression: Weight vs. Height",
```

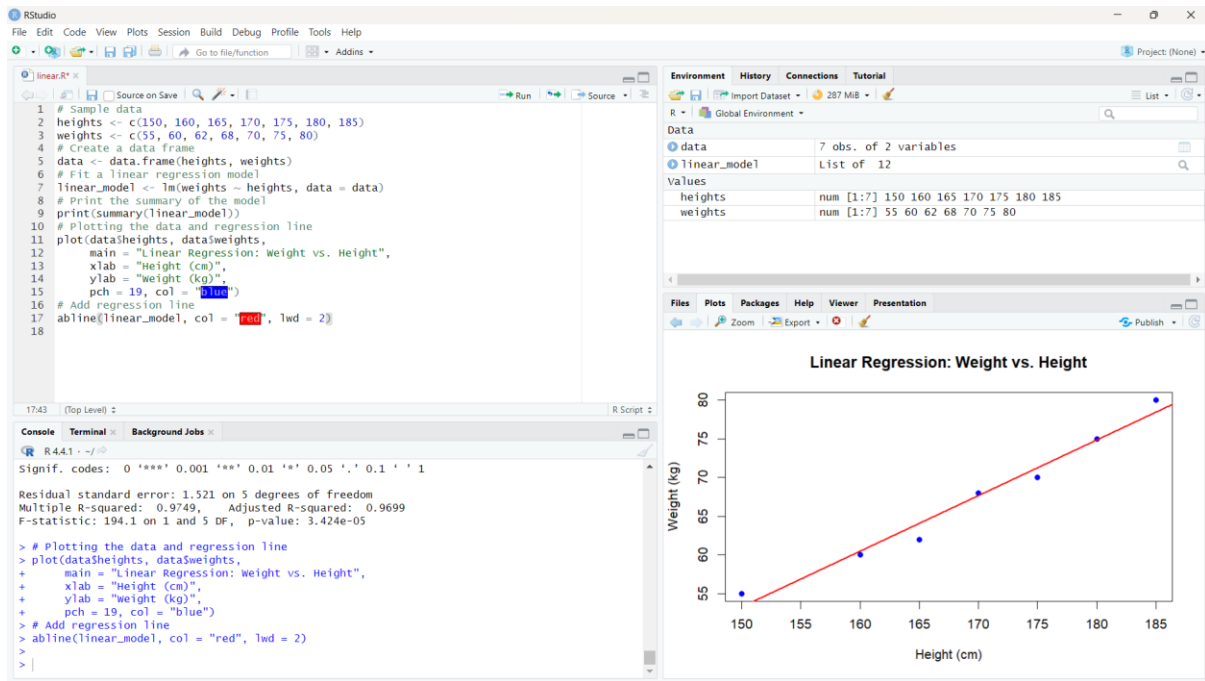
```
  xlab = "Height (cm)",
```

```
  ylab = "Weight (kg)",
```

```
  pch = 19, col = "blue")
```

```
abline(linear_model, col = "red", lwd = 2)
```



**OUTPUT:****b) Logistic regression**

# Logistic Regression

data(mtcars)

mtcars\$am &lt;- factor(mtcars\$am, levels = c(0, 1), labels = c("Automatic", "Manual"))

logistic\_model &lt;- glm(am ~ mpg, data = mtcars, family = binomial)

print(summary(logistic\_model))

# Plotting Logistic Regression

predicted\_probs &lt;- predict(logistic\_model, type = "response")

print(predicted\_probs)

plot(mtcars\$mpg, as.numeric(mtcars\$am) - 1,

main = "Logistic Regression: Transmission vs. MPG",

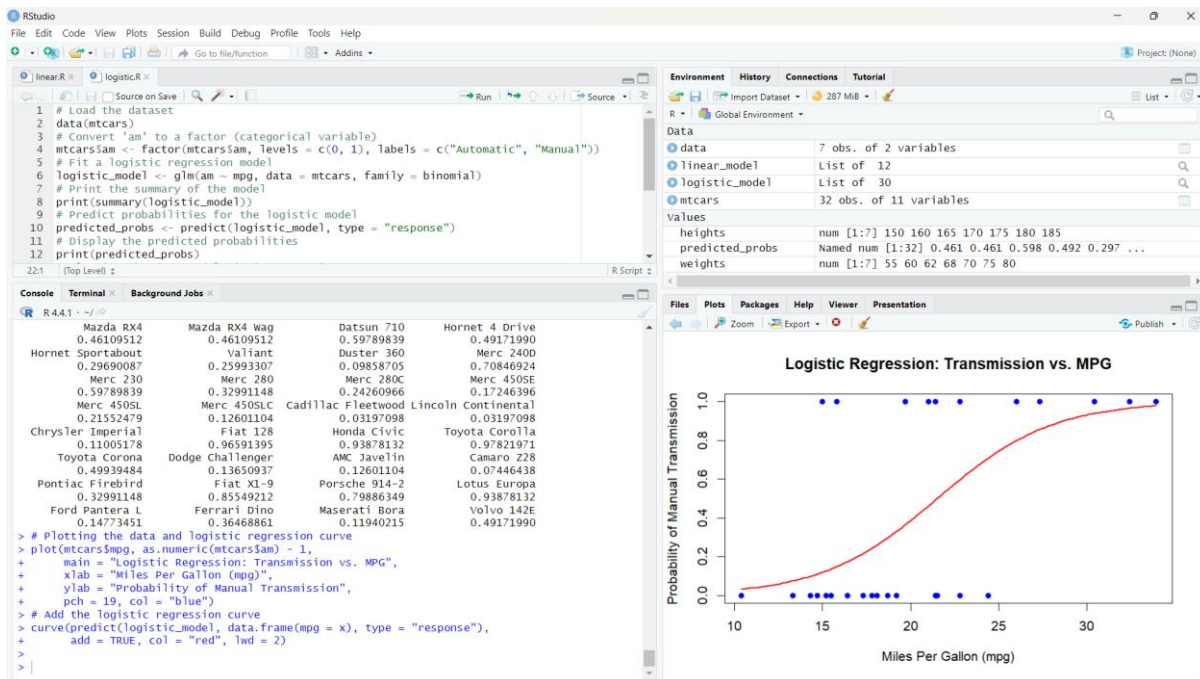
xlab = "Miles Per Gallon (mpg)",

ylab = "Probability of Manual Transmission",

pch = 19, col = "blue")

curve(predict(logistic\_model, data.frame(mpg = x), type = "response"),

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

**OUTPUT:****RESULT:**

Thus the R program to implement Linear and Logistic Regression has been executed and verified successfully.