

Program-1

P-1 Install R and R studio. Get yourself acquainted with GUI of various working windows of R studio.

#Name : Jyoti devi

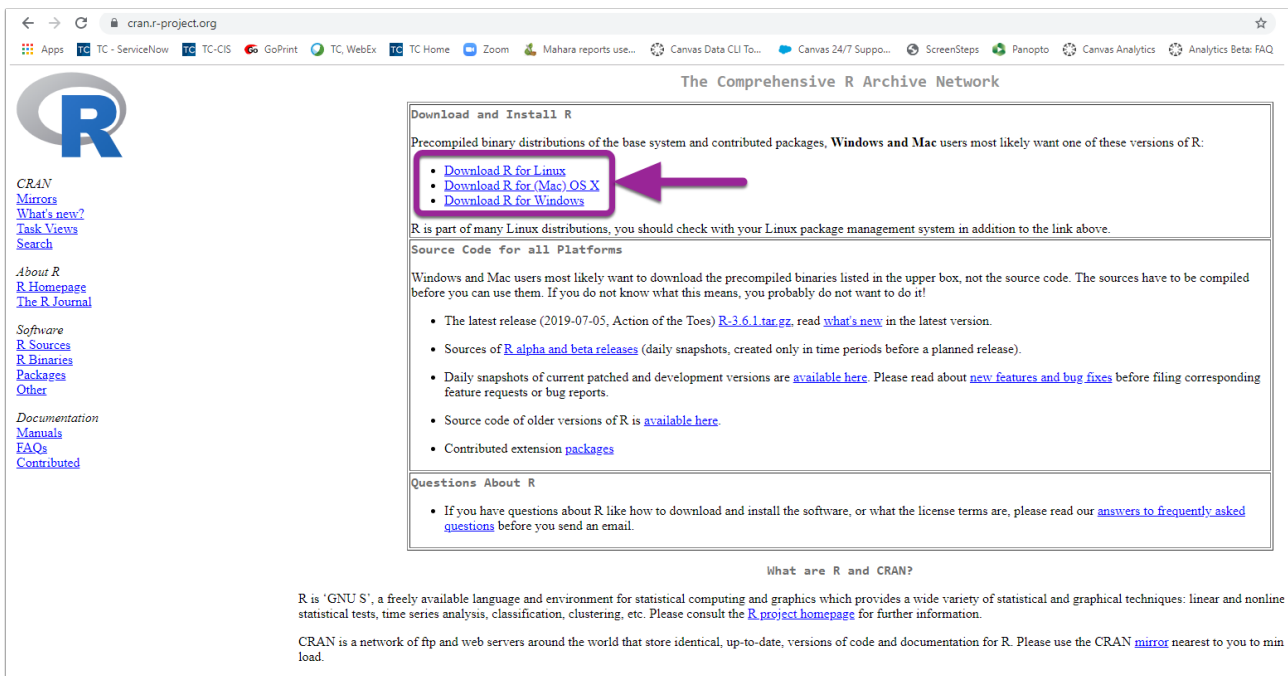
#Roll no. : 200010130051

#Class : B.tech(cse), G-3

Install R on Windows

1. To install R, go to cran.r-project.org

2. Depending on your operating system, click Download R for (your operating system).



The screenshot shows the CRAN website with the 'Download and Install R' section. A purple box highlights the download links for Linux, Mac OS X, and Windows, with a purple arrow pointing to the Windows link.

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 (2019-07-05, Action of the Toes) [R-3.6.1.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

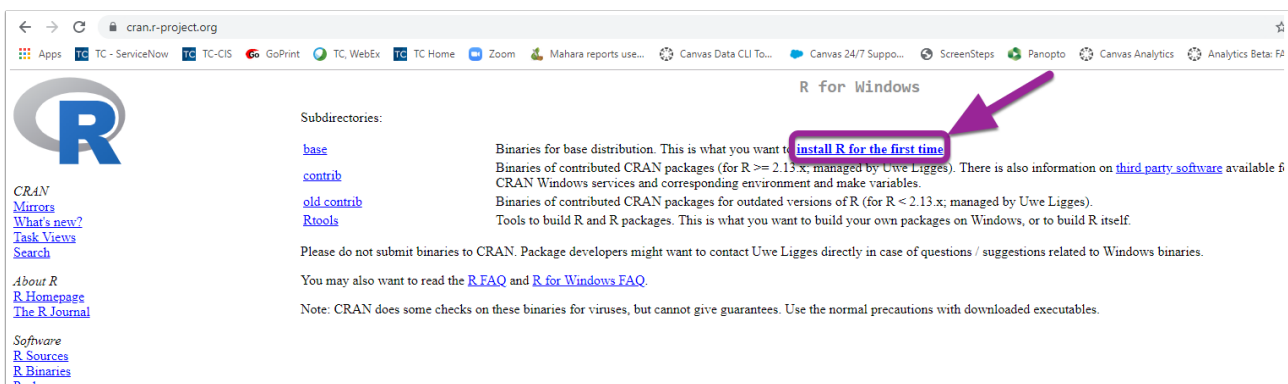
- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

What are R and CRAN?

R is 'GNU S', a freely available language and environment for statistical computing and graphics which provides a wide variety of statistical and graphical techniques: linear and nonlinear statistical tests, time series analysis, classification, clustering, etc. Please consult the [R project homepage](#) for further information.

CRAN is a network of ftp and web servers around the world that store identical, up-to-date, versions of code and documentation for R. Please use the CRAN [mirror](#) nearest to you to min load.

3. Click on install R for the first time.



The screenshot shows the CRAN website with the 'R for Windows' section. A purple box highlights the 'install R for the first time' link, with a purple arrow pointing to it.

R for Windows

Subdirectories:

- [base](#)
- [contrib](#)
- [old.contrib](#)
- [Rtools](#)

Binaries for base distribution. This is what you want to [install R for the first time](#).

Binaries of contributed CRAN packages (for R >= 2.13.x; managed by Uwe Ligges). There is also information on [third party software](#) available for CRAN Windows services and corresponding environment and make variables.

Binaries of contributed CRAN packages for outdated versions of R (for R < 2.13.x; managed by Uwe Ligges).

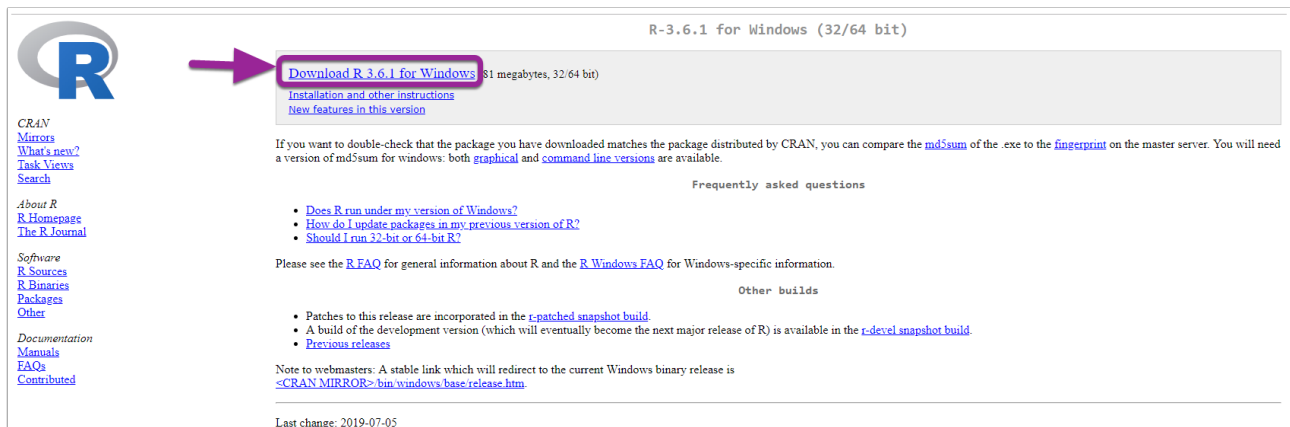
Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

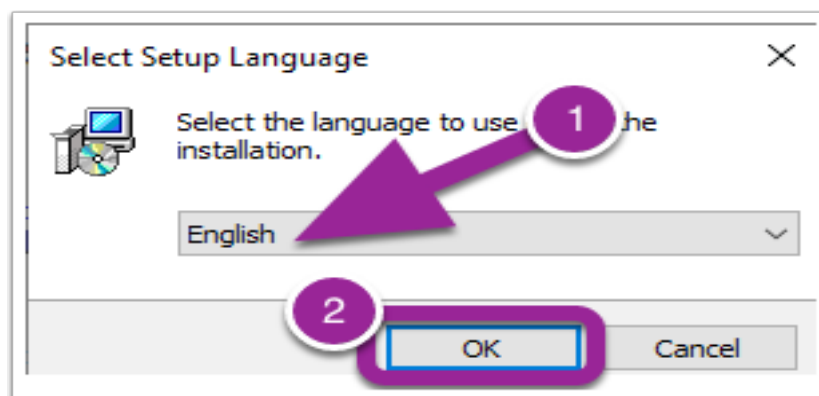
You may also want to read the [R FAQ](#) and [R for Windows FAQ](#).

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

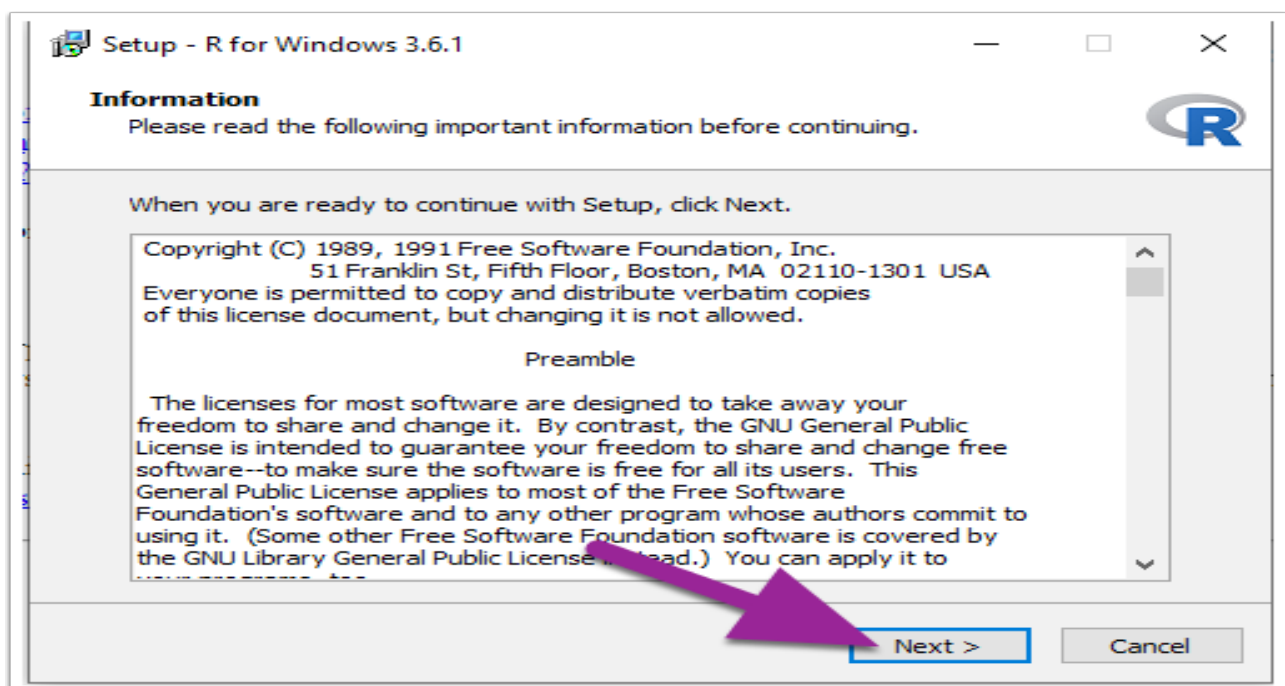
4. Click Download R for Windows. Open the downloaded file.



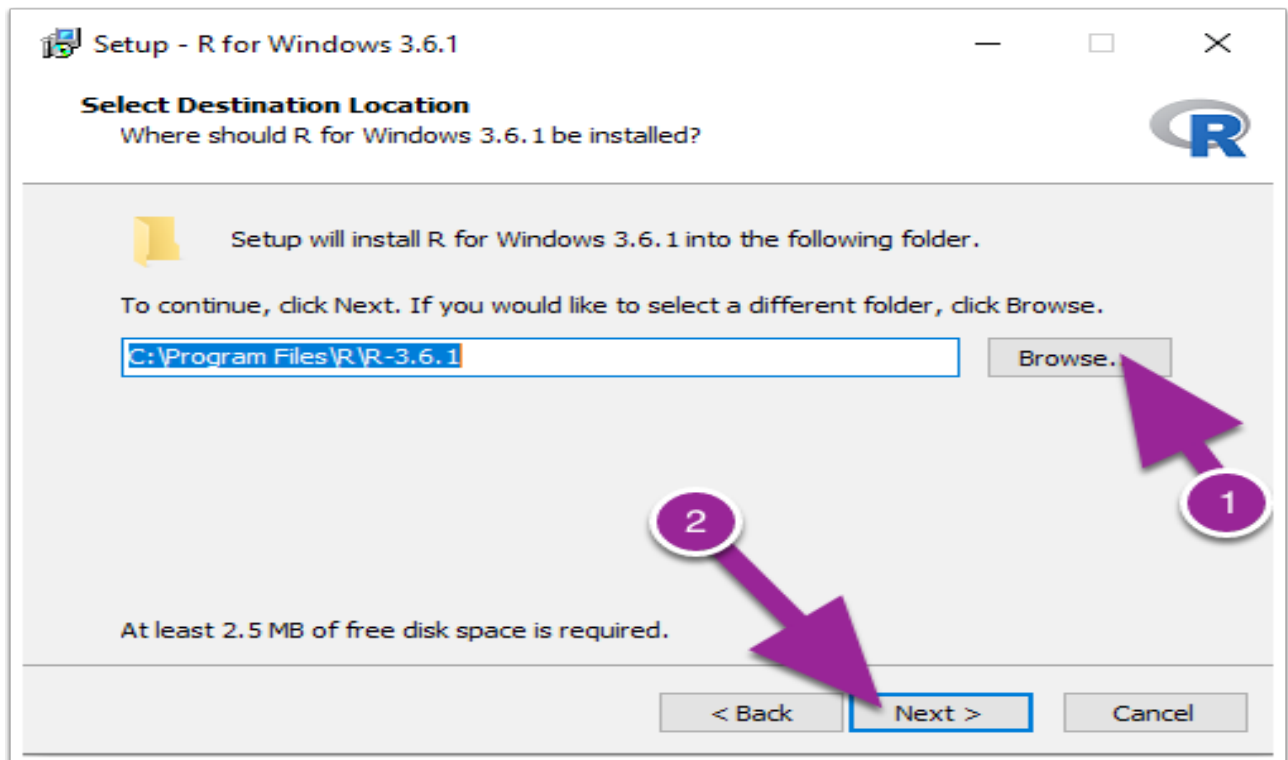
5. Select the language you would like to use during the installation. Then click OK.



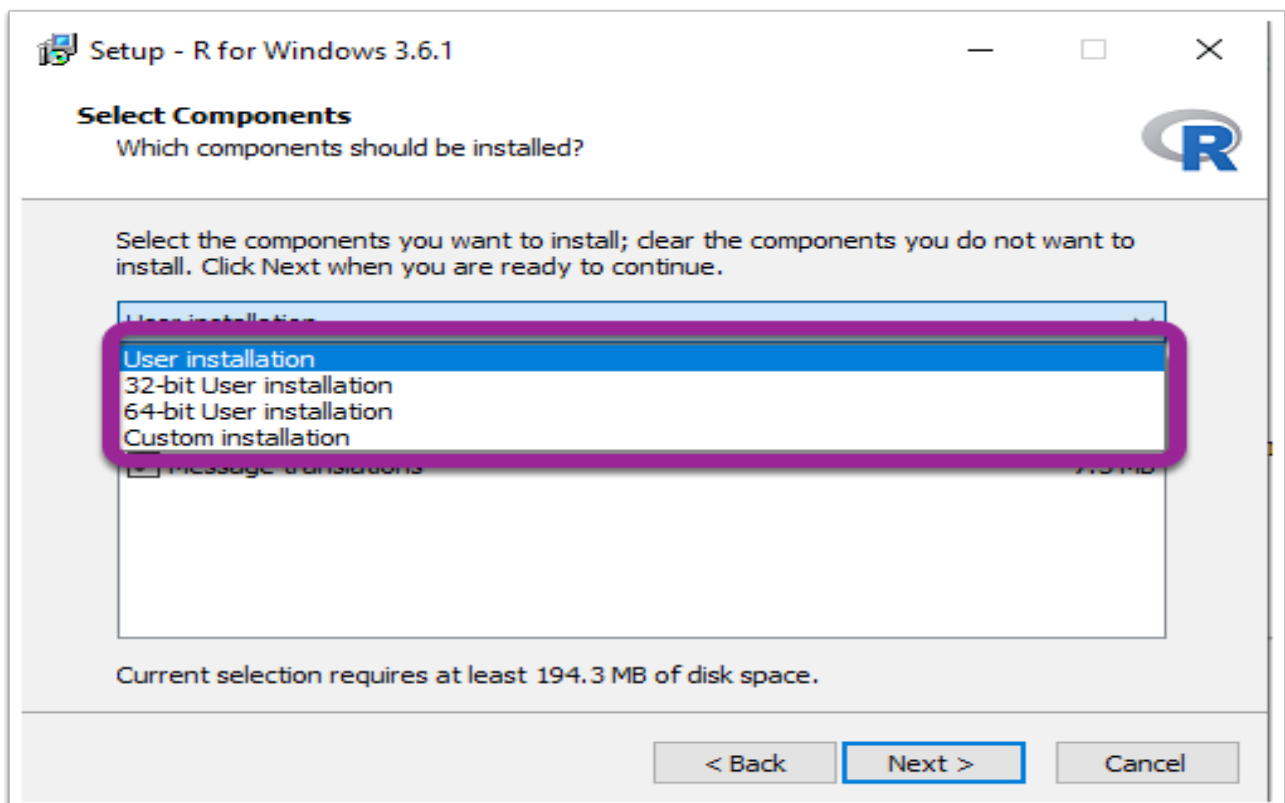
6. Click Next.



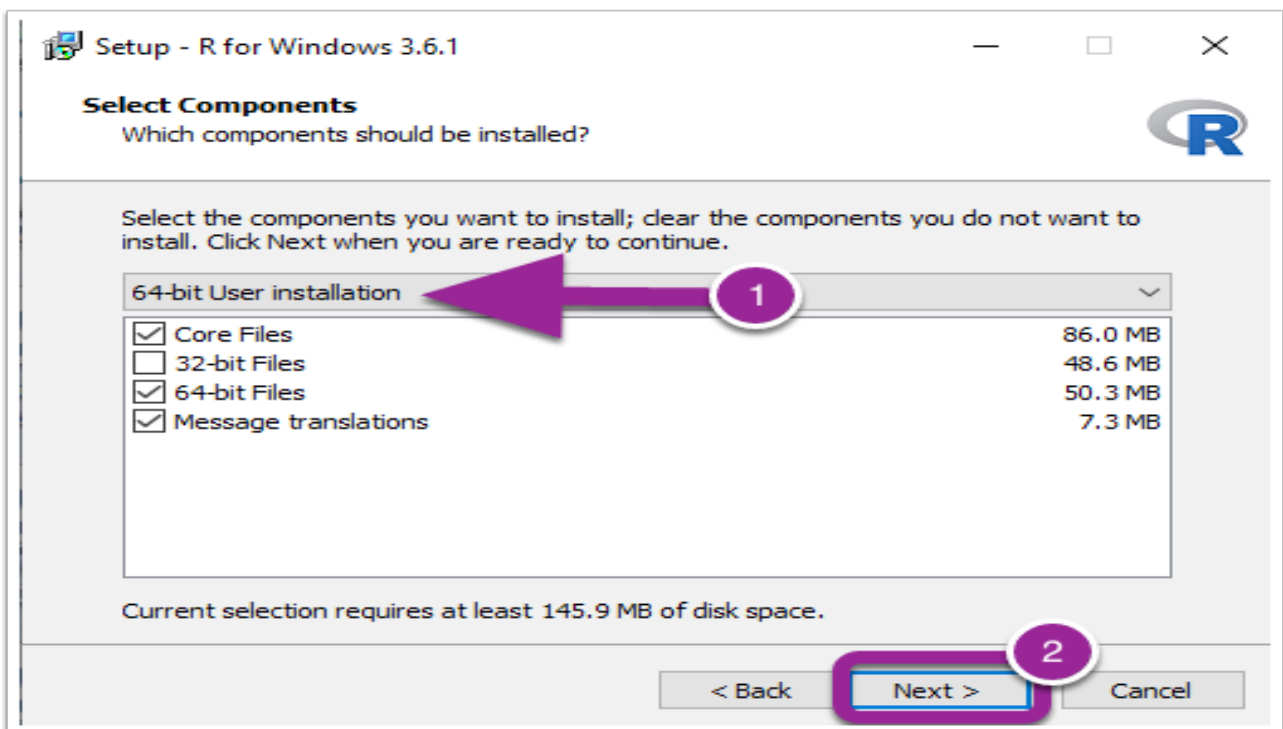
7. Select where you would like R to be installed. It will default to your Program Files on your C Drive. Click Next.



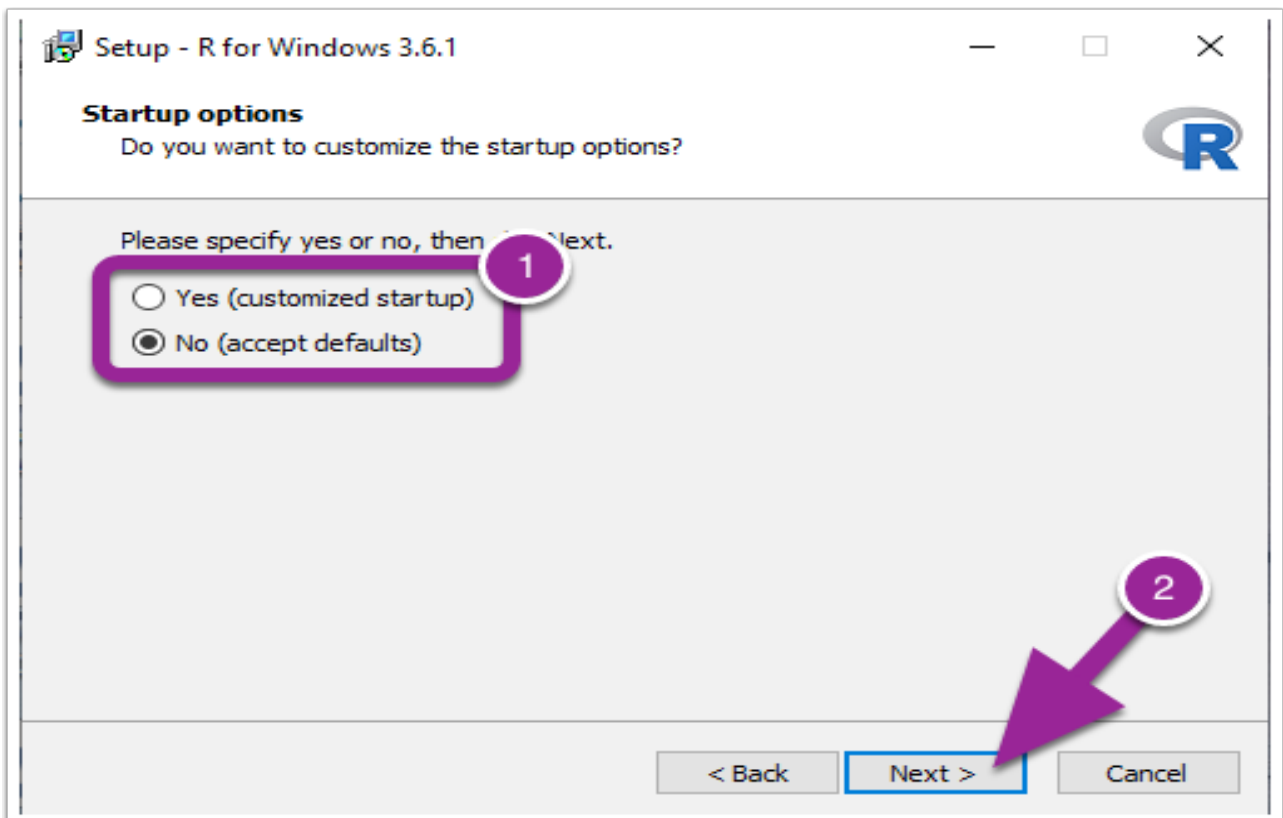
8. You can then choose which installation you would like.



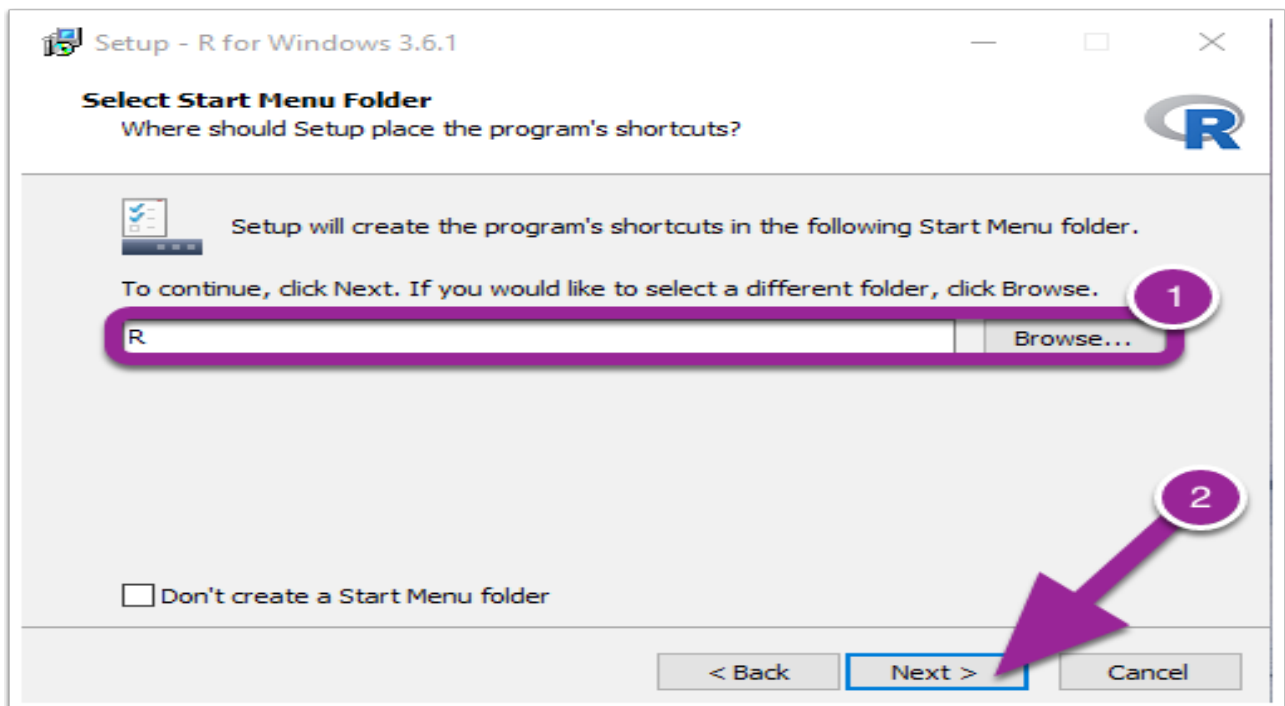
9. (Optional) If your computer is a 64-bit, you can choose the 64-bit User Installation. Then click Next.



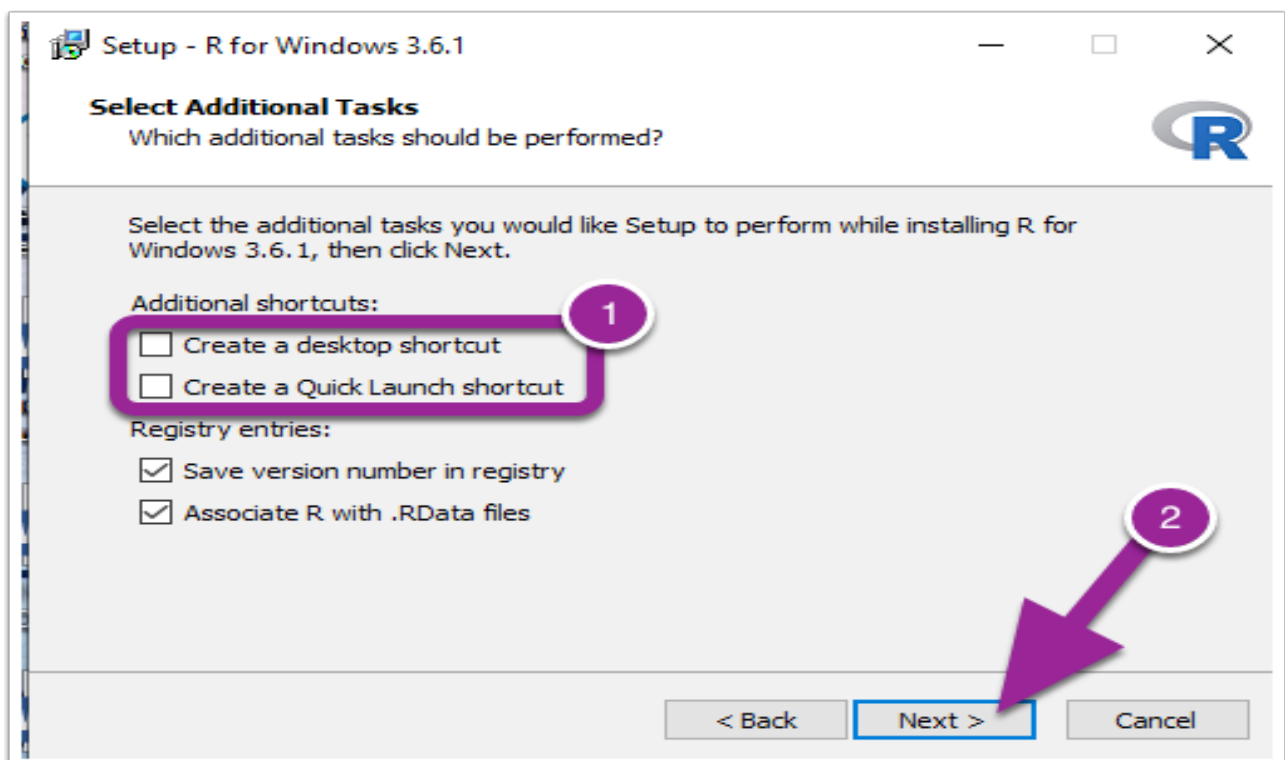
10. Then specify if you want to customized your startup or just use the defaults. Then click Next.



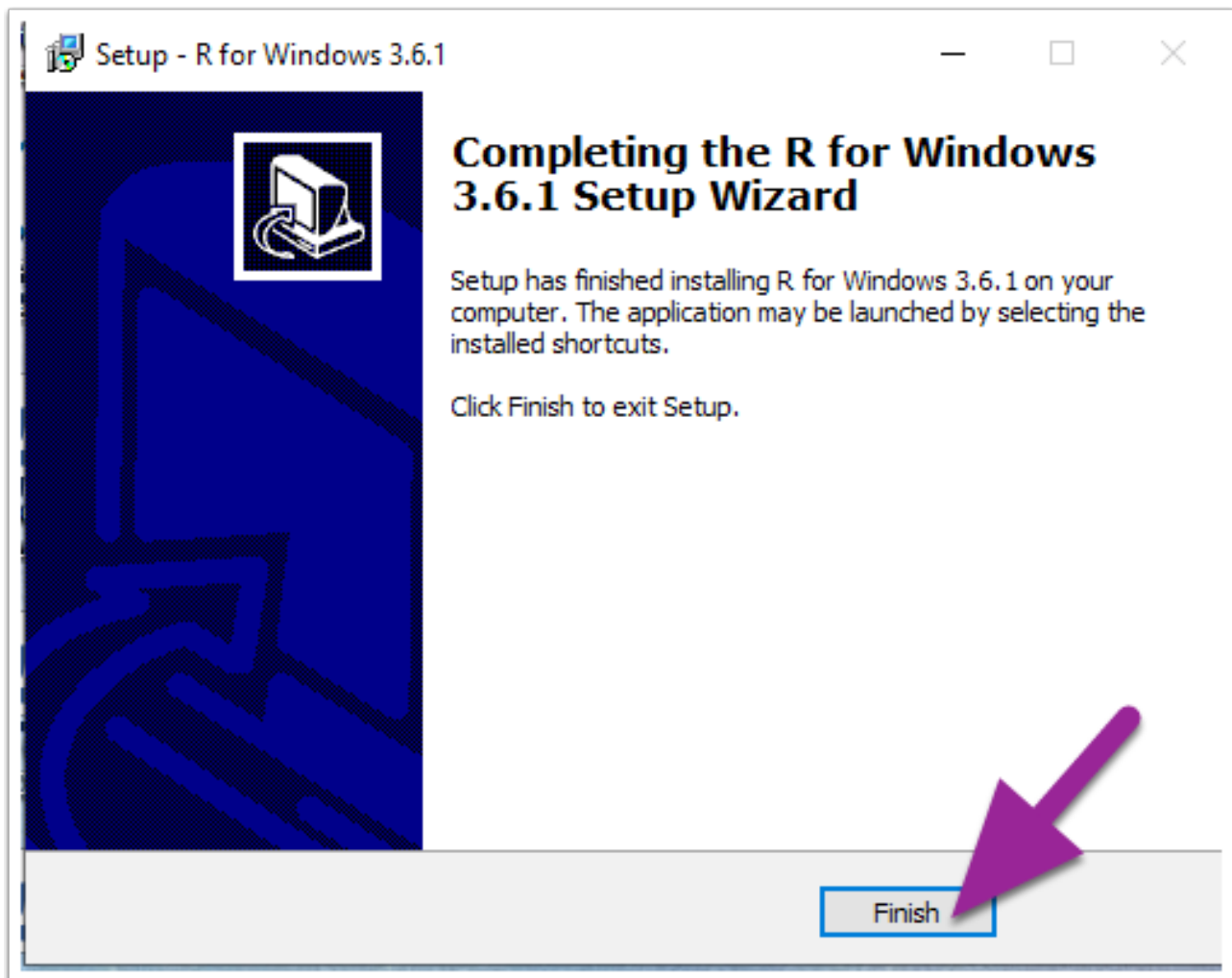
11. Then you can choose the folder that you want R to be saved within or the default if the R folder that was created. Once you have finished, click Next.



12. You can then select additional shortcuts if you would like. Click Next.

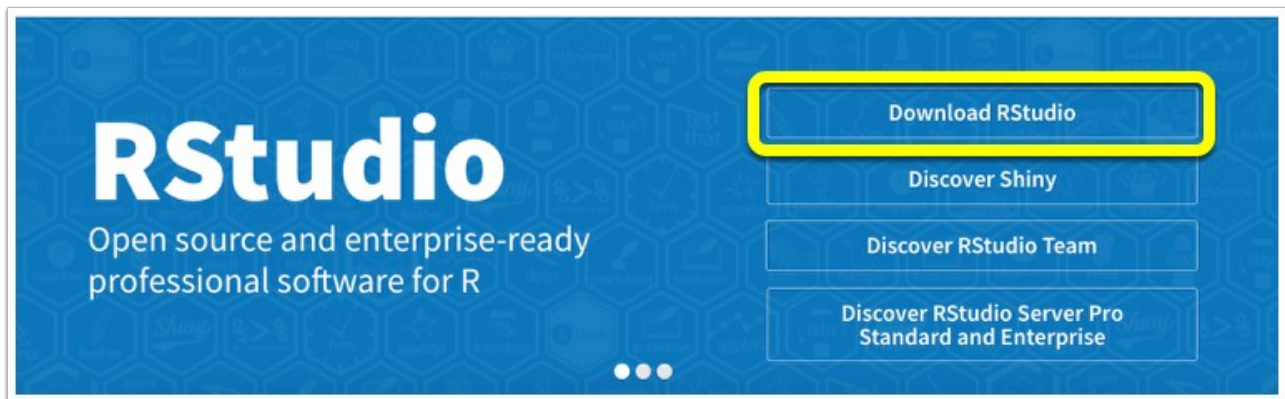


13. Click Finish

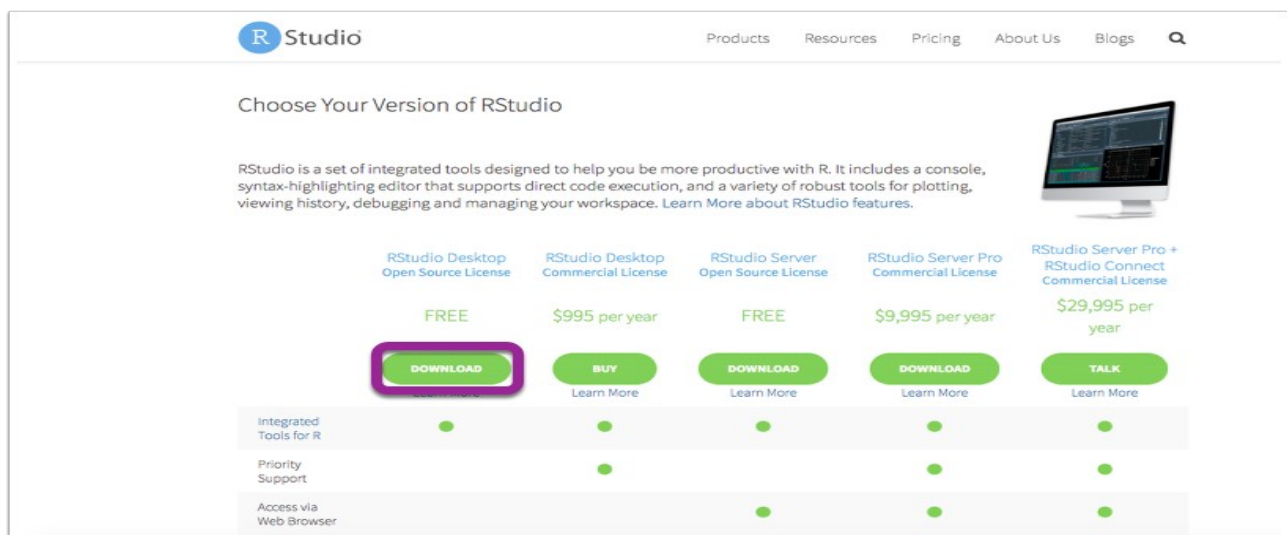


download RStudio. Go to www.rstudio.com

1. Click Download RStudio.



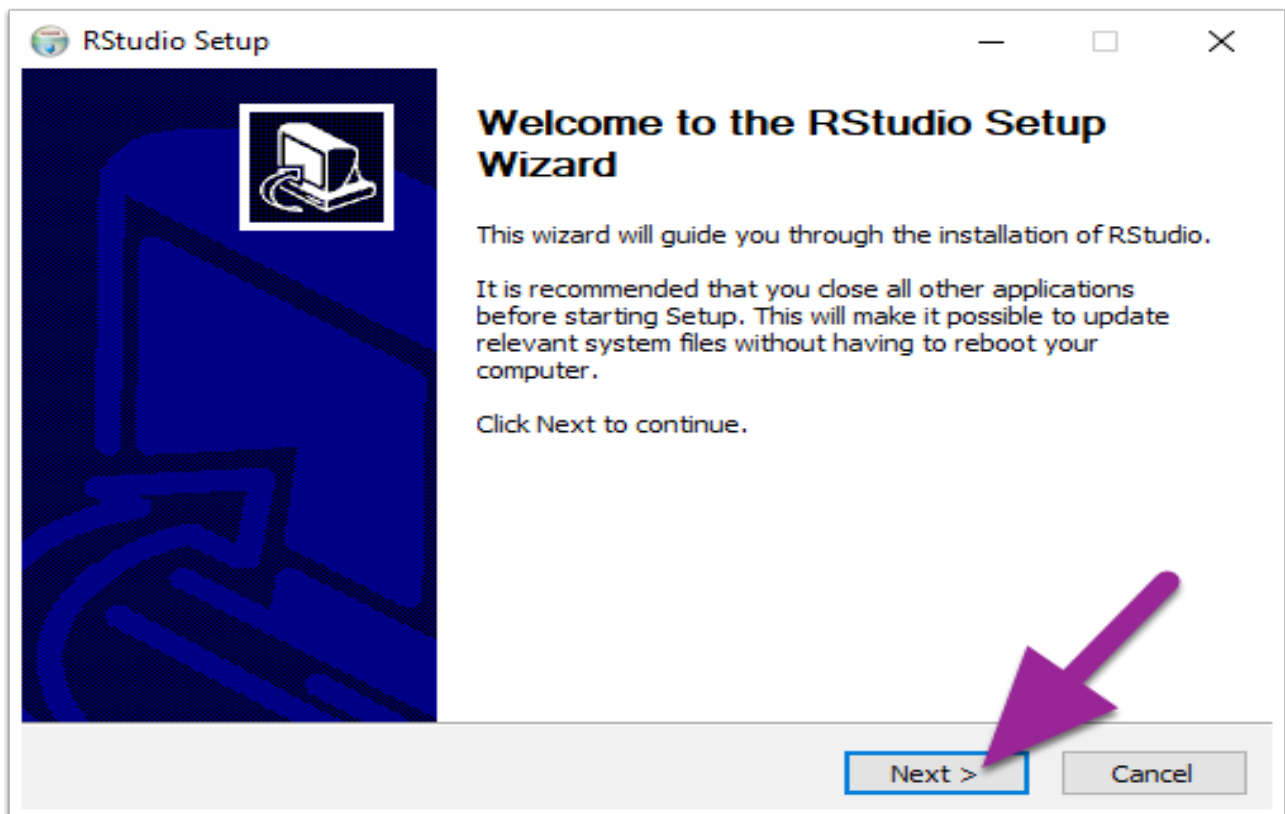
2. Click Download under RStudio Desktop- Open Source License.



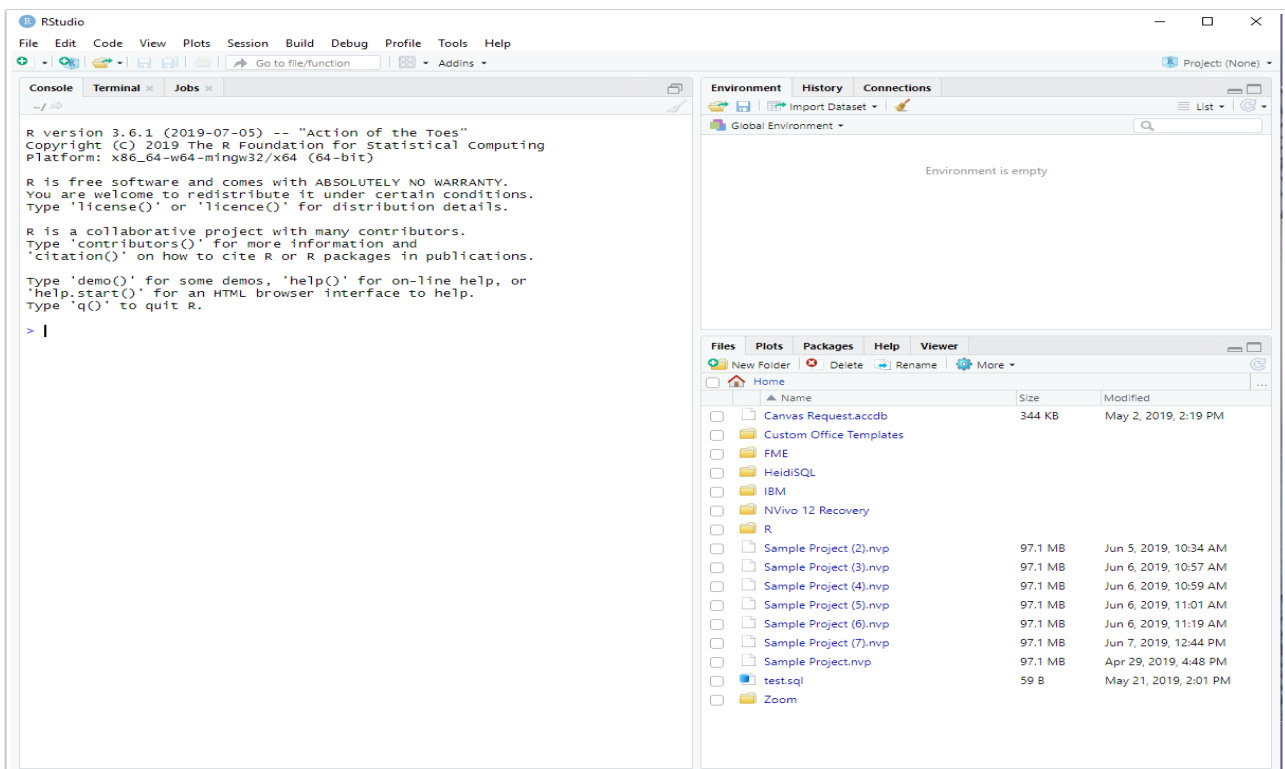
3. Click on the operating system that you are working with.

Installers for Supported Platforms			
Installers	Size	Date	MD5
RStudio 1.2.1335 - Windows 7+ (64-bit)	126.9 MB	2019-04-08	d0e2470f1f8ef4cd35a669aa323a2136
RStudio 1.2.1335 - Mac OS X 10.12+ (64-bit)	121.1 MB	2019-04-08	6c570b0e2144583f7c48c284ce299eef
RStudio 1.2.1335 - Ubuntu 14/Debian 8 (64-bit)	92.2 MB	2019-04-08	c1b07d0511469abfe582919b183eee83
RStudio 1.2.1335 - Ubuntu 16 (64-bit)	99.3 MB	2019-04-08	c142d69c210257fb10d18c045fff13c7
RStudio 1.2.1335 - Ubuntu 18 (64-bit)	100.4 MB	2019-04-08	71a8d1990c0d97939804b46cfb0aea75
RStudio 1.2.1335 - Fedora 19+/RedHat 7+ (64-bit)	114.1 MB	2019-04-08	296b6ef88969a91297fab6545f256a7a
RStudio 1.2.1335 - Debian 9+ (64-bit)	100.6 MB	2019-04-08	1e32d4d6f6e216f086a81ca82ef65a91
RStudio 1.2.1335 - OpenSUSE 15+ (64-bit)	101.6 MB	2019-04-08	2795a63c7efd8e2aa2dae86ba09a81e5
RStudio 1.2.1335 - SLES/OpenSUSE 12+ (64-bit)	94.4 MB	2019-04-08	c65424b06ef6737279d982db9eefcae1

4. The RStudio installation wizard will pop-up. Click Next and go through the installation steps.



5. Congratulations! You have now installed R and RStudio.



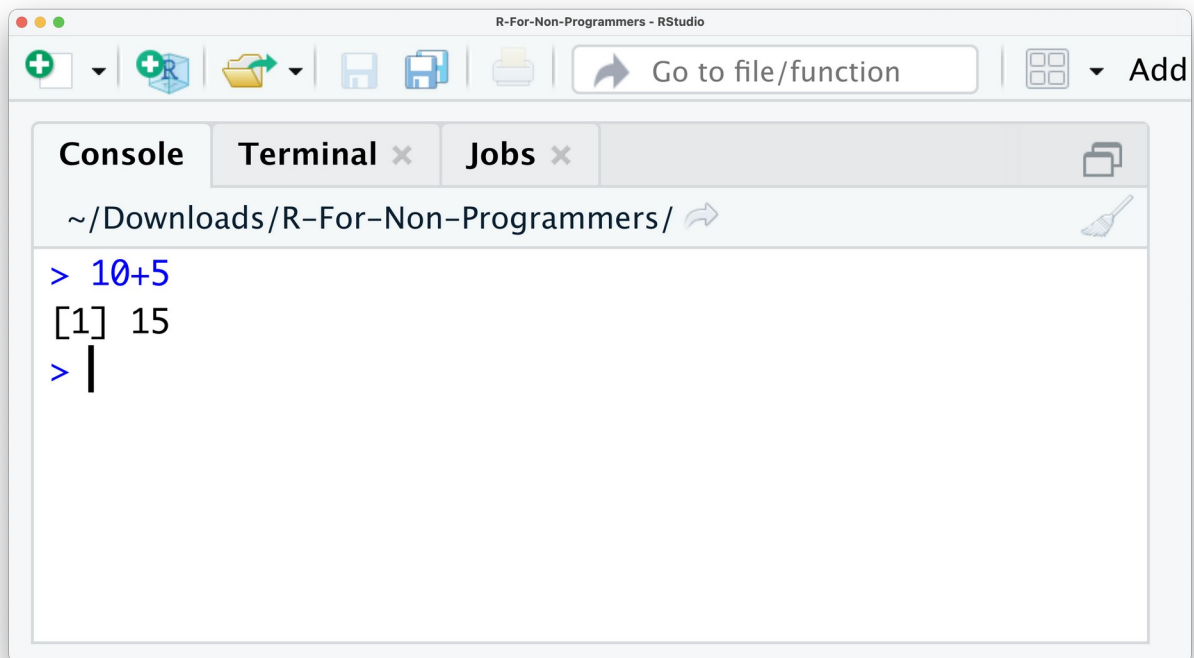
The RStudio Interface

1. The Console window

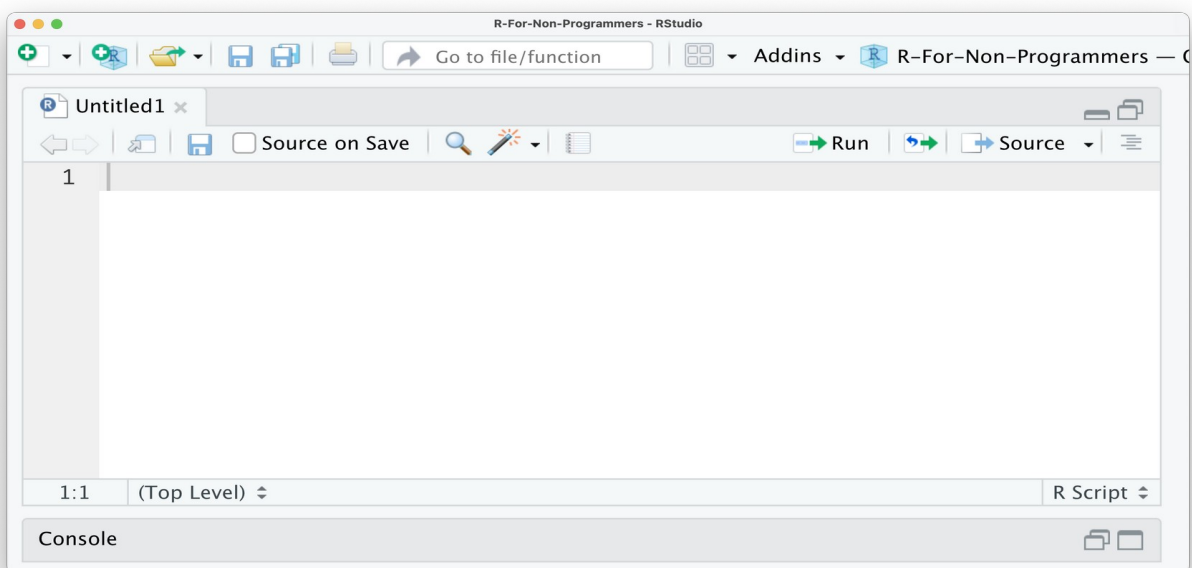
We type the below into the console

```
10 + 5
```

```
#> [1] 15
```

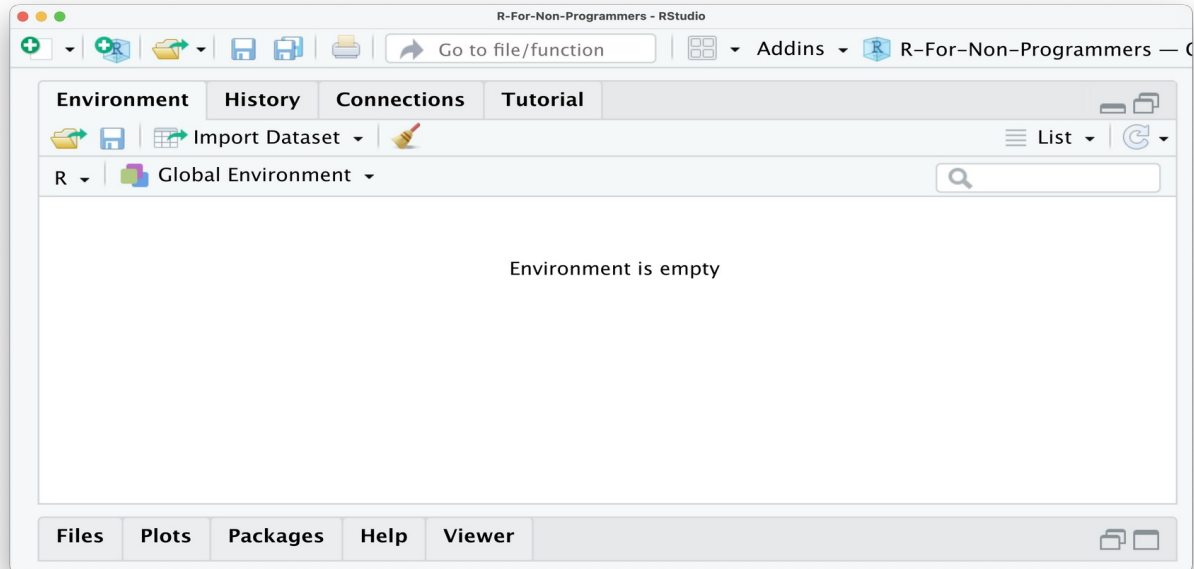


2. The Source window

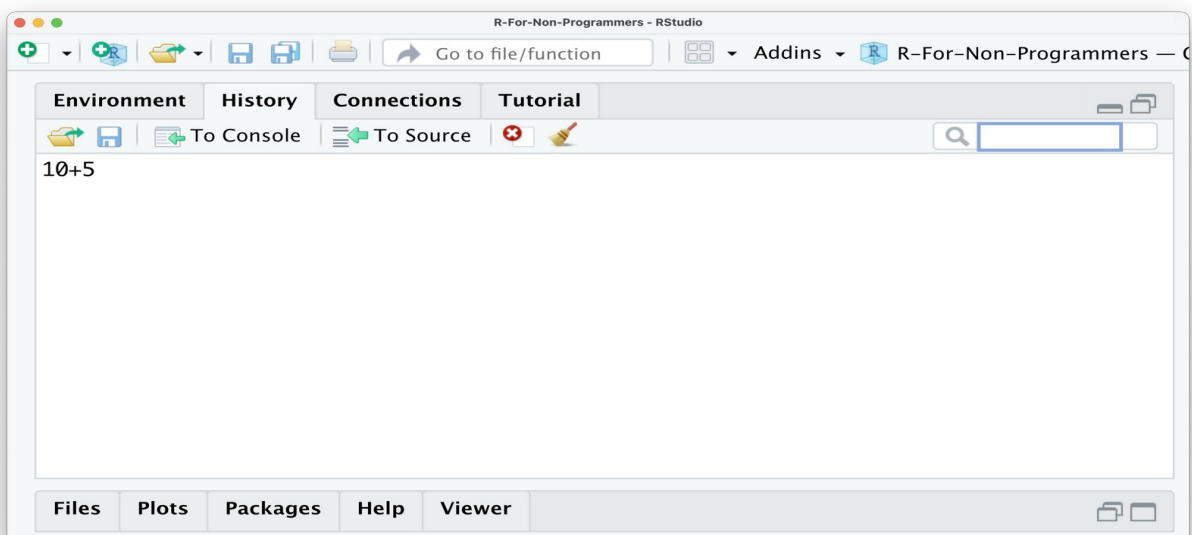


3. The Environment / History / Connections / Tutorial window

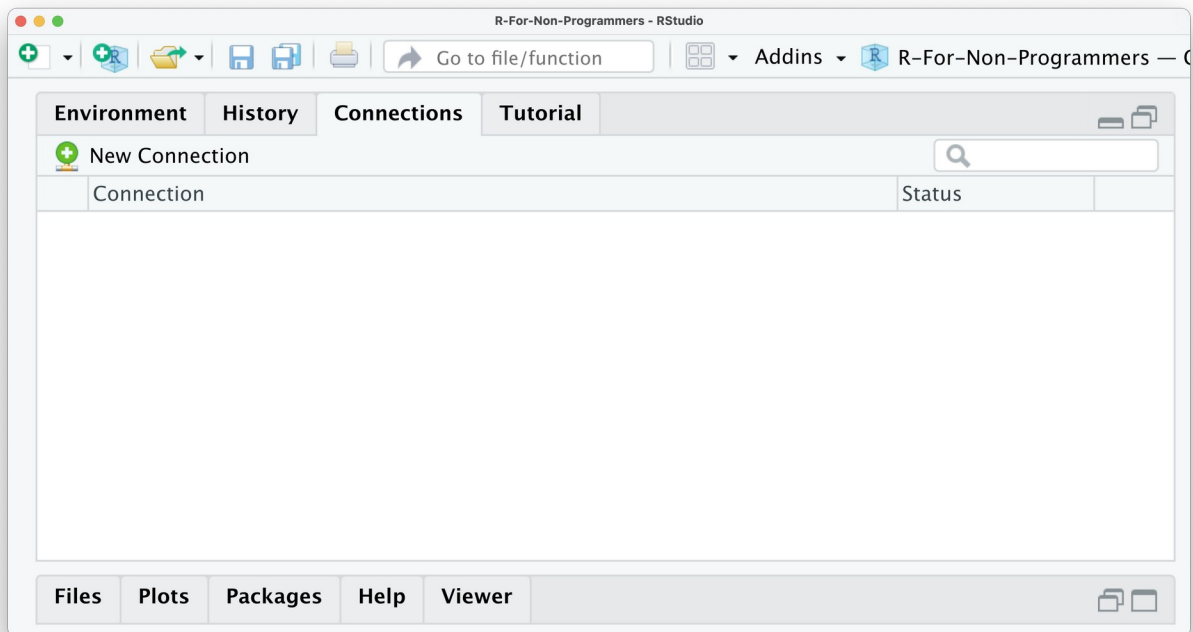
The window in the top right shows multiples panes. The first pane is called *Environment* and shows you objects which are available for computation



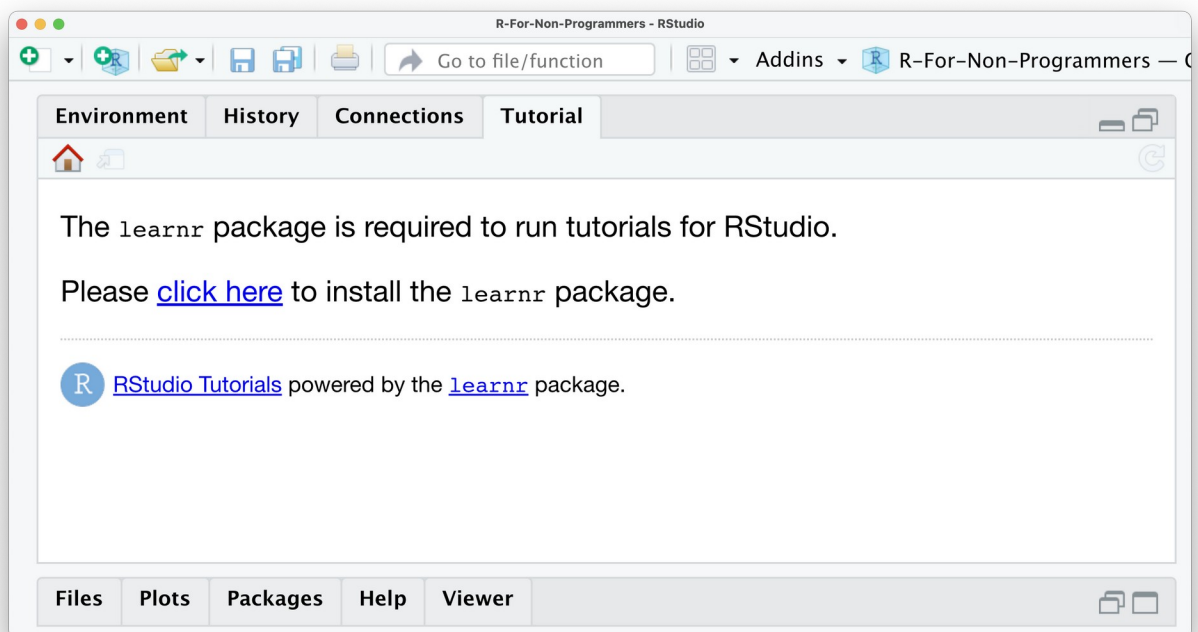
The **History** pane is very easy to understand. Whatever computation you run in the console will be stored. So you can go back and see what you coded and rerun that code.



The pane **Connections** allows you to tab into external databases directly.



The last pane is called **Tutorial**. Here you can find additional materials to learn **R** and Rstudio.



Program-2

P-2 Perform the following operations in R:

a) Create variable of different data type and print their class.

b) Perform type conversion.

c) Perform all the basic mathematical operations.

a).

#Name : Jyoti devi

#Roll no. : 200010130051

#Class : B.tech(cse), G-3

#Code:

#Numeric Datatype

x<-8.56

class(x)

#Integer Datatype

y<-24L

class(y)

#Logical Datatype

x<-FALSE

y<-TRUE

class(x)

class(y)

#Complex Datatype

z<-3+3i

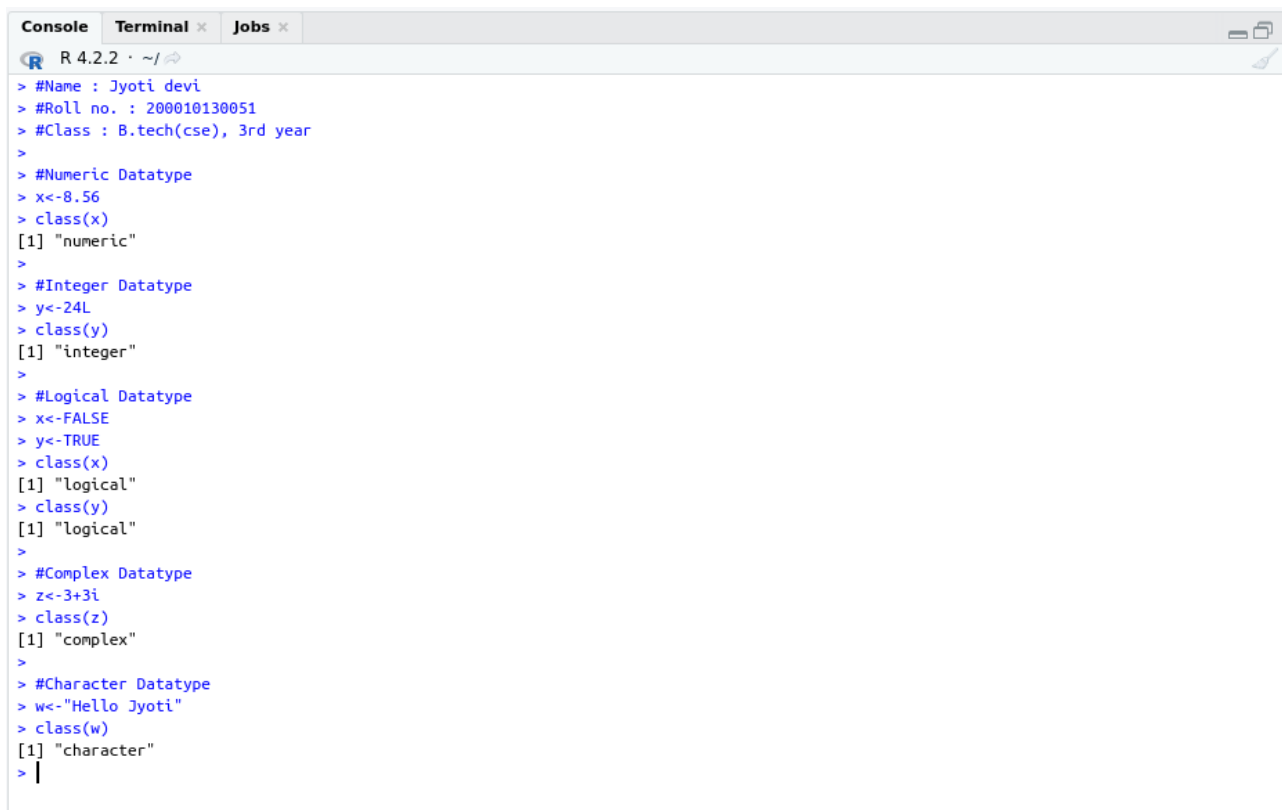
class(z)

#Character Datatype

w<- "Hello Jyoti"

class(w)

Output:



```
R 4.2.2 · ~/
> #Name : Jyoti devi
> #Roll no. : 200010130051
> #Class : B.tech(cse), 3rd year
>
> #Numeric Datatype
> x<-8.56
> class(x)
[1] "numeric"
>
> #Integer Datatype
> y<-24L
> class(y)
[1] "integer"
>
> #Logical Datatype
> x<-FALSE
> y<-TRUE
> class(x)
[1] "logical"
> class(y)
[1] "logical"
>
> #Complex Datatype
> z<-3+3i
> class(z)
[1] "complex"
>
> #Character Datatype
> w<-"Hello Jyoti"
> class(w)
[1] "character"
> |
```

b).

#Name : Jyoti devi

#Roll no. : 200010130051

#Class : B.tech(cse), G-3

#Code:

#1. Convert from numeric to integer

a<-11.3

class(a)

x<-as.integer(a)

class(x)

#2. Convert from integer to numeric

b<-46L

class(b)

y<-as.numeric(b)

class(y)

#3. Convert from numeric to complex

d<-11.34

w<-as.complex(d)

class(w)

#4. Convert from integer to complex

c<-11L

z<-as.complex(c)

class(z)

#5. Convert from complex to integer

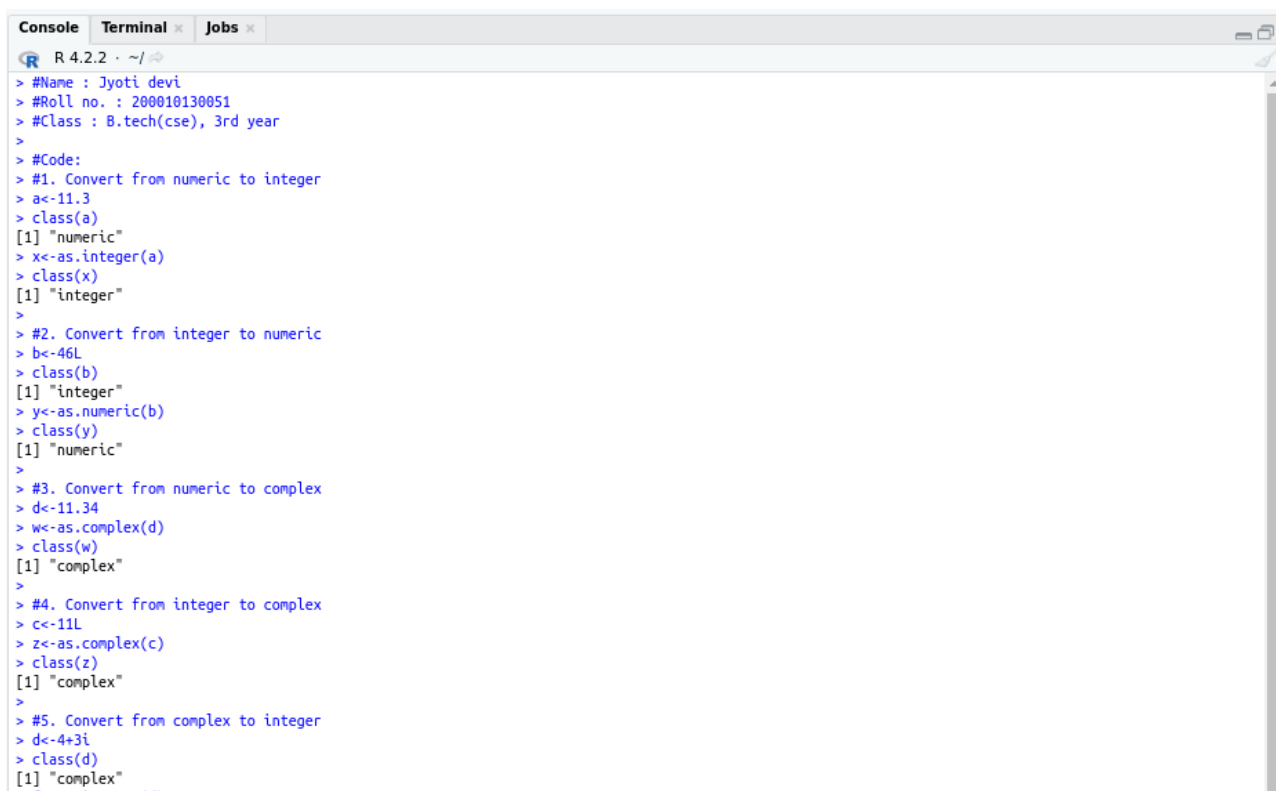
d<-4+3i

class(d)

f<-as.integer(d)

class(f)

Output:



```
R 4.2.2 · ~/
> #Name : Jyoti devi
> #Roll no. : 200010130051
> #Class : B.tech(cse), 3rd year
>
> #Code:
> #1. Convert from numeric to integer
> a<-11.3
> class(a)
[1] "numeric"
> x<-as.integer(a)
> class(x)
[1] "integer"
>
> #2. Convert from integer to numeric
> b<-46L
> class(b)
[1] "integer"
> y<-as.numeric(b)
> class(y)
[1] "numeric"
>
> #3. Convert from numeric to complex
> d<-11.34
> w<-as.complex(d)
> class(w)
[1] "complex"
>
> #4. Convert from integer to complex
> c<-11L
> z<-as.complex(c)
> class(z)
[1] "complex"
>
> #5. Convert from complex to integer
> d<-4+3i
> class(d)
[1] "complex"
```


c).

#Name : Jyoti devi

#Roll no. : 200010130051

#Class : B.tech(cse),G-3

#Code:

#min() returns the minimum value

a<-min(10,44,100,5,6)

a

#max() returns the maximum value

b<-max(.7010,20,50,80)

b

#sqrt() returns the square root

c<-sqrt(25)

c

#abs() returns the absolute value

d<-abs(-11)

d

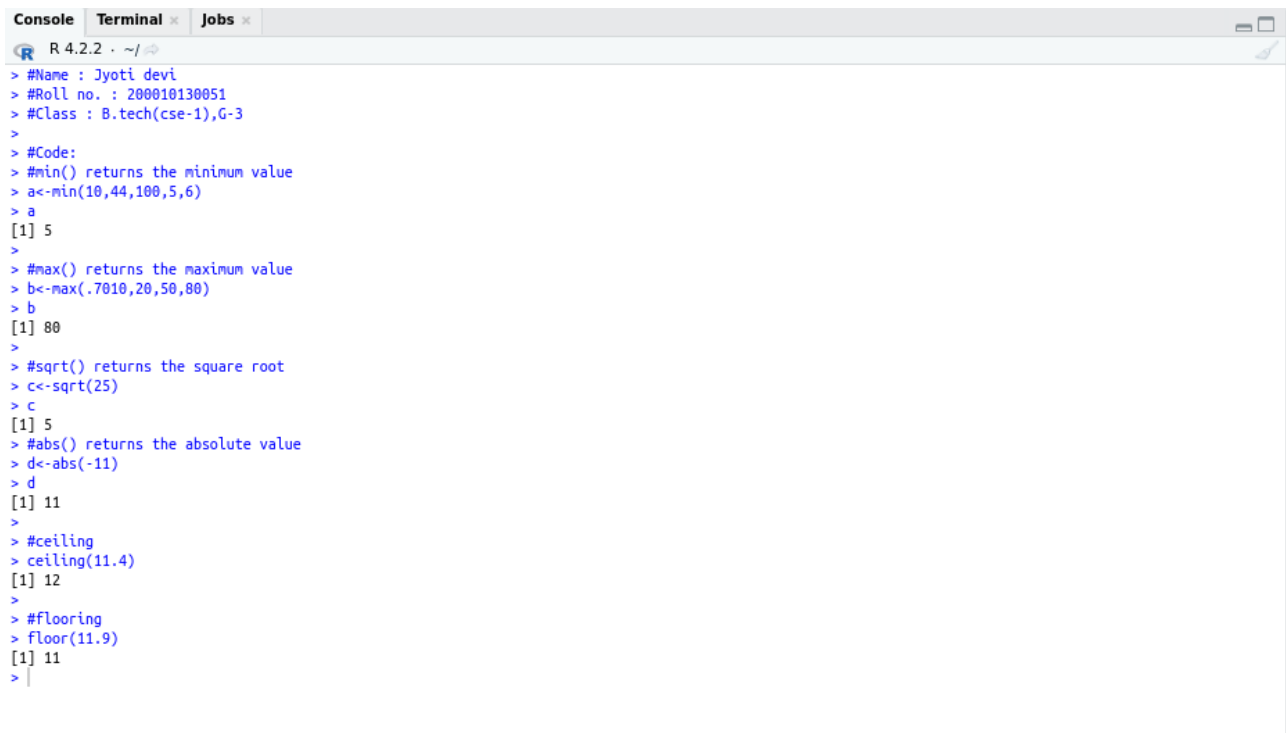
#ceiling

ceiling(11.4)

#flooring

floor(11.9)

Output:



```
Console | Terminal x | Jobs x
R 4.2.2 · ~/
> #Name : Jyoti devi
> #Roll no. : 200010130051
> #Class : B.tech(cse-1),G-3
>
> #Code:
> #min() returns the minimum value
> a<-min(10,44,100,5,6)
> a
[1] 5
>
> #max() returns the maximum value
> b<-max(.7010,20,50,80)
> b
[1] 80
>
> #sqrt() returns the square root
> c<-sqrt(25)
> c
[1] 5
> #abs() returns the absolute value
> d<-abs(-11)
> d
[1] 11
>
> #ceiling
> ceiling(11.4)
[1] 12
>
> #flooring
> floor(11.9)
[1] 11
> |
```

Program-3

P-3 Create two vectors and find their addition, subtraction and element wise multiplication. Concatenate the two vectors and find their sum and average.

#Name : Jyoti devi

#Roll no. : 200010130051

#Class : B.tech(cse),G-3

#Code:

#Creating two Vectors

vec1<-c(1,2,3,4,5)

vec1

vec2<-c(10,20,30,40,50)

vec2

#addition of vectors

vec3<-vec1+vec2

vec3

#subtraction of vectors

sub<-vec1-vec2

sub

#Element wise multiplication

mul<-vec1*vec2

mul

#concatenate two vectors

a<-c("hello","how","are","you")

b<-c("I","am","fine")

paste(a,b) #paste function used to combine two vectors

x<-c(1,2,3)

y<-c(4,5,6)

paste(x,y)

#sum and average of vectors

vec5<-c(11,22,33,44,55)

vec6<-c(11,12,13,14,15)

sum(vec5)

sum(vec6)

sum(vec5,vec6)

#average of vectors

mean(vec5)

avg<-sum(vec5)/length(vec5)

vec5

Output:

```
Console Terminal Jobs
R 4.2.2 ~ /
> #Name : Jyoti devi
> #Roll no. : 200010130051
> #Class : B.tech(cse), 3rd year
>
> #Code:
> #Creating two Vectors
> vec1<-c(1,2,3,4,5)
> vec1
[1] 1 2 3 4 5
> vec2<-c(10,20,30,40,50)
> vec2
[1] 10 20 30 40 50
>
> #Addition of vectors
> vec3<-vec1+vec2
> vec3
[1] 11 22 33 44 55
>
> #Subtraction of vectors
> sub<-vec1-vec2
> sub
[1] -9 -18 -27 -36 -45
>
> #Element wise multiplication
> mul<-vec1*vec2
> mul
[1] 10 40 90 160 250
>
> #Concatenate two vectors
> ac<-c("hello","how","are you")
> bc<-c("I","am","fine")
>
> paste(a,b) #paste function used to combine two vectors
[1] "hello I" "how an" "are you fine"
> x<-c(1,2,3)
> y<-c(4,5,6)
> paste(x,y)
[1] "1 4" "2 5" "3 6"
>
>
> #sum and average of vectors
> vec5<-c(11,22,33,44,55)
> vec6<-c(11,12,13,14,15)
> sum(vec5)
[1] 165
> sum(vec6)
[1] 65
> sum(vec5,vec6)
[1] 230
>
> #Average of vectors
> mean(vec5)
[1] 33
> avg<-sum(vec5)/length(vec5)
> vec5
[1] 11 22 33 44 55
> |
```