Data Visualization with R and Google Play Store Data

Open Lab | Digital Summit 2019

www.miraclesoft.com

Data Visualization with R and Google Play Store Data

Introduction

This document contains a step-by-step process of analyzing Google Play Store Apps data with R and will teach you how to create graphs using R language.

This guide was prepared by **Miracle's Innovation Labs**.

Pre-Requisites

All attendees must have their workstation (with Internet) to participate in the lab. The following pre-requisites will help you to make the Hands-on Lab experience easier.

- Download and install R and R studio

Technology Involved

- R Programming

Lab Steps

So, let us get started with the Installation!

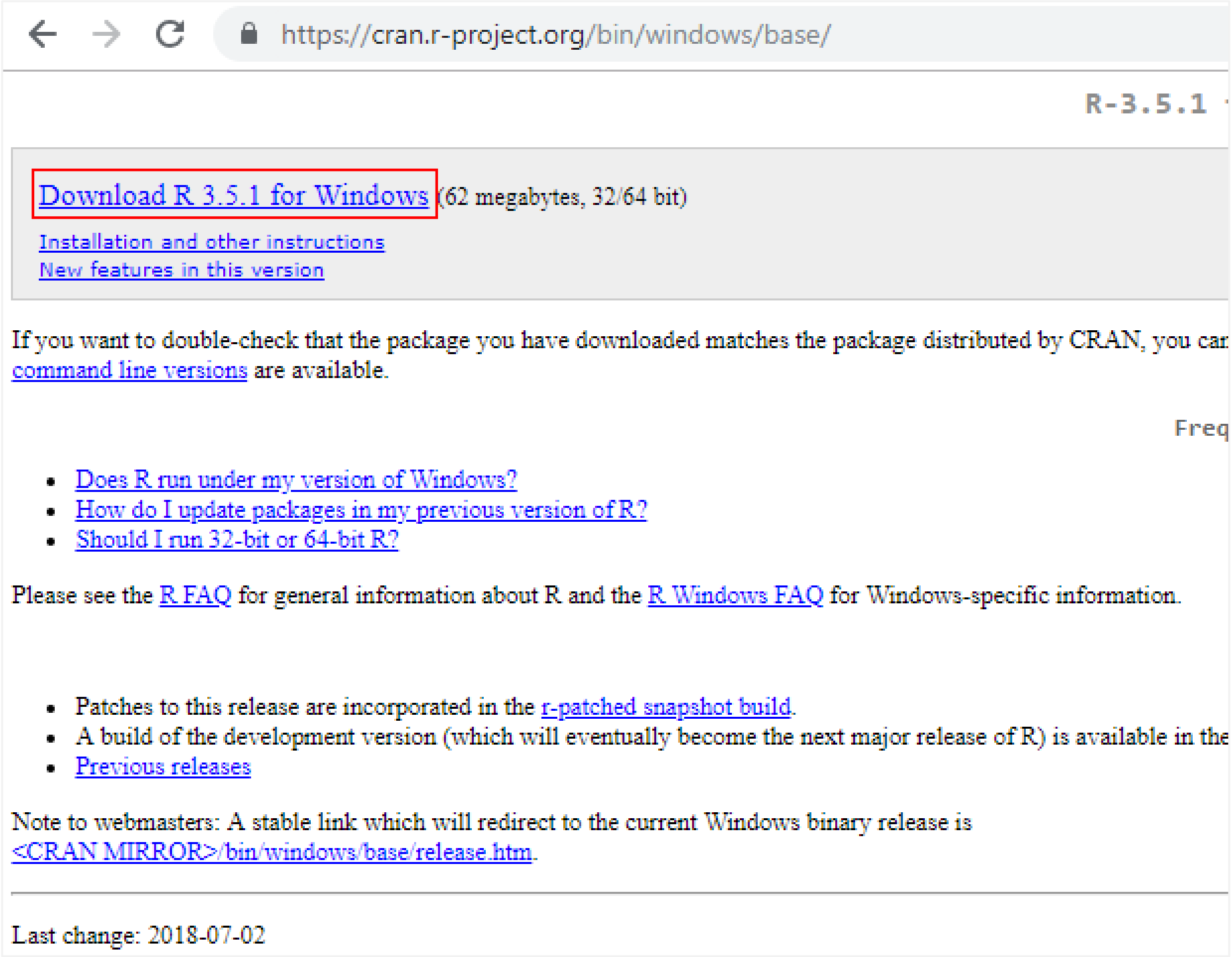
The following steps will outline how to install R and RStudio in your environment.

Step #1 | Installing R

To install R on your computer, follow these steps,

Use this link to download the latest R version,

https://cran.r-project.org/bin/windows/base/



https://cran.r-project.org/bin/windows/base/

R-3.5.1

Download R 3.5.1 for Windows (62 megabytes, 32/64 bit)

Installation and other instructions
New features in this version

If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can command line versions are available.

Freq

- Does R run under my version of Windows?
- How do I update packages in my previous version of R?
- Should I run 32-bit or 64-bit R?

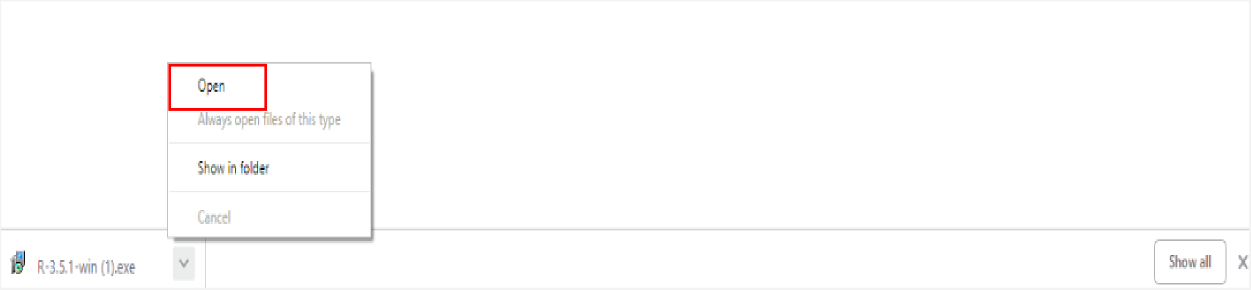
Please see the R FAQ for general information about R and the R Windows FAQ for Windows-specific information.

- Patches to this release are incorporated in the r-patched snapshot build.
- A build of the development version (which will eventually become the next major release of R) is available in the
- Previous releases

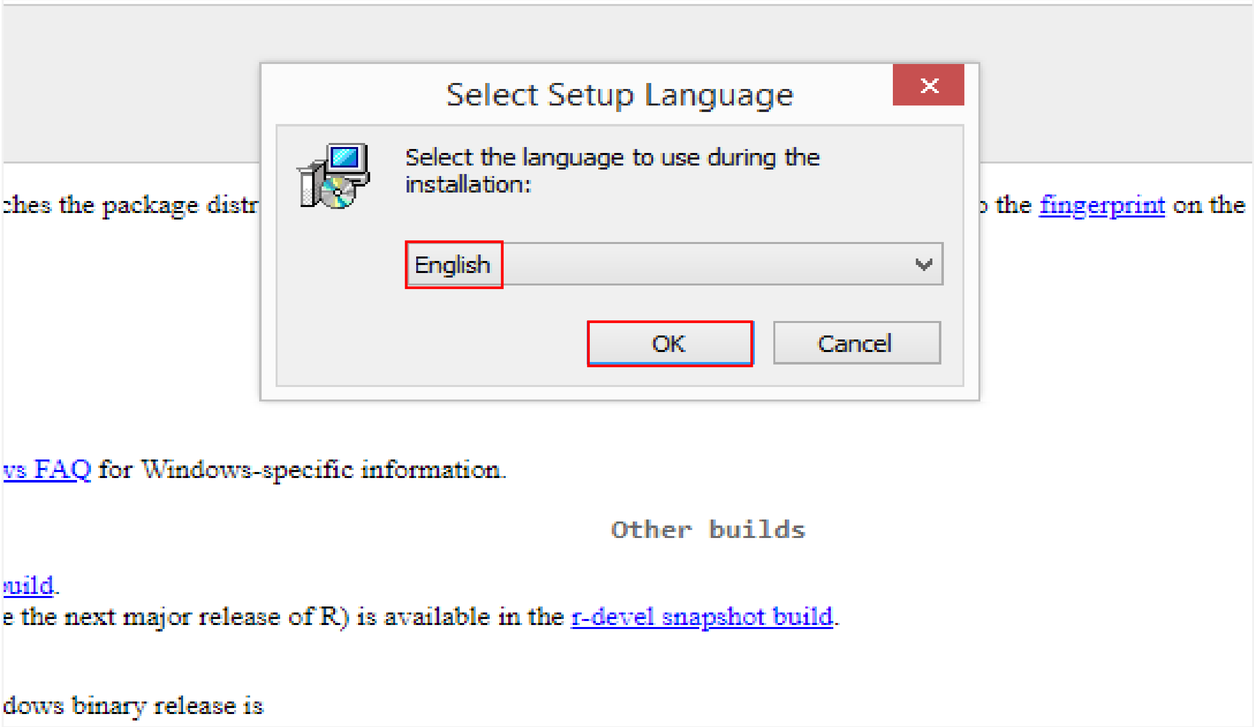
Note to webmasters: A stable link which will redirect to the current Windows binary release is <CRAN MIRROR>/bin/windows/base/release.htm.

Last change: 2018-07-02

Click on **Download R 3.5.1 for Windows.**



R is successfully downloaded on your machine, click on **Open** option to run the .exe file. You will be asked to choose a language to install, choose **English** and click on **Ok**.



Select Setup Language

Select the language to use during the installation:

English

ches the package distr … the fingerprint on the

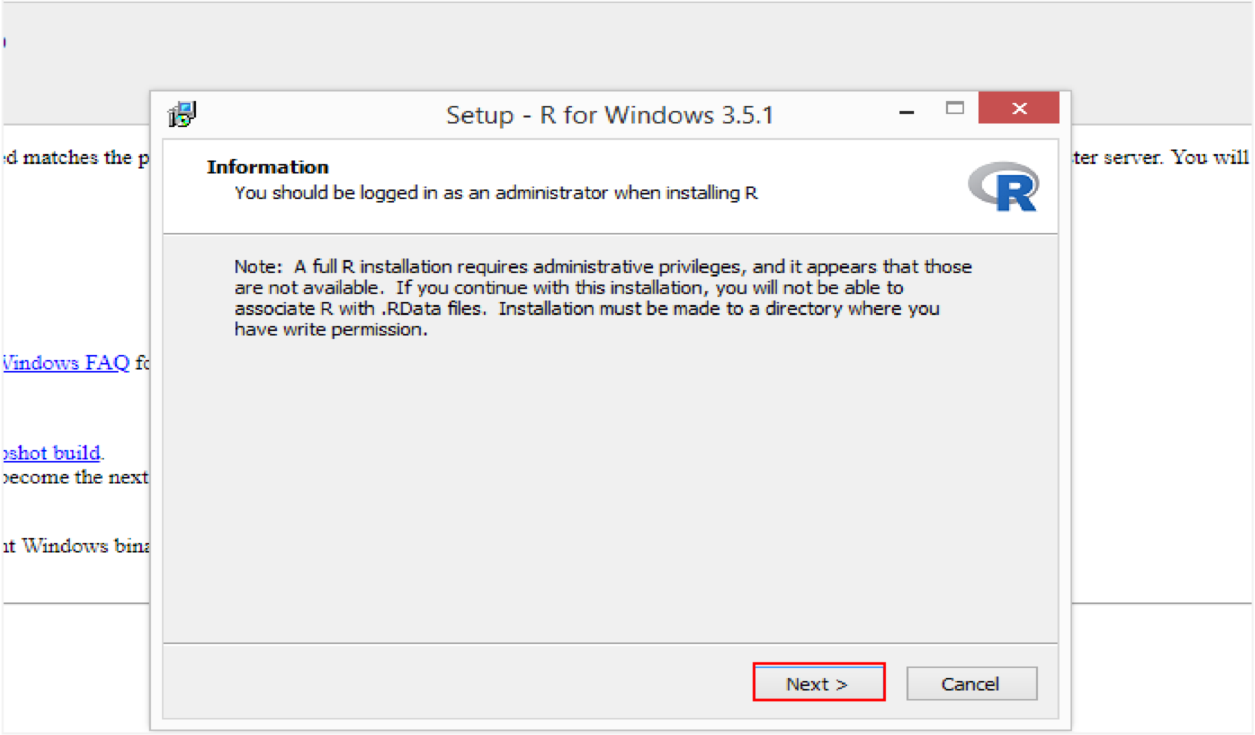
vs FAQ for Windows-specific information.

Other builds

uild.
e the next major release of R) is available in the r-devel snapshot build.

dows binary release is

The R Setup Wizard will appear in a window. Click on **Next** at the bottom of the R Setup wizard window.

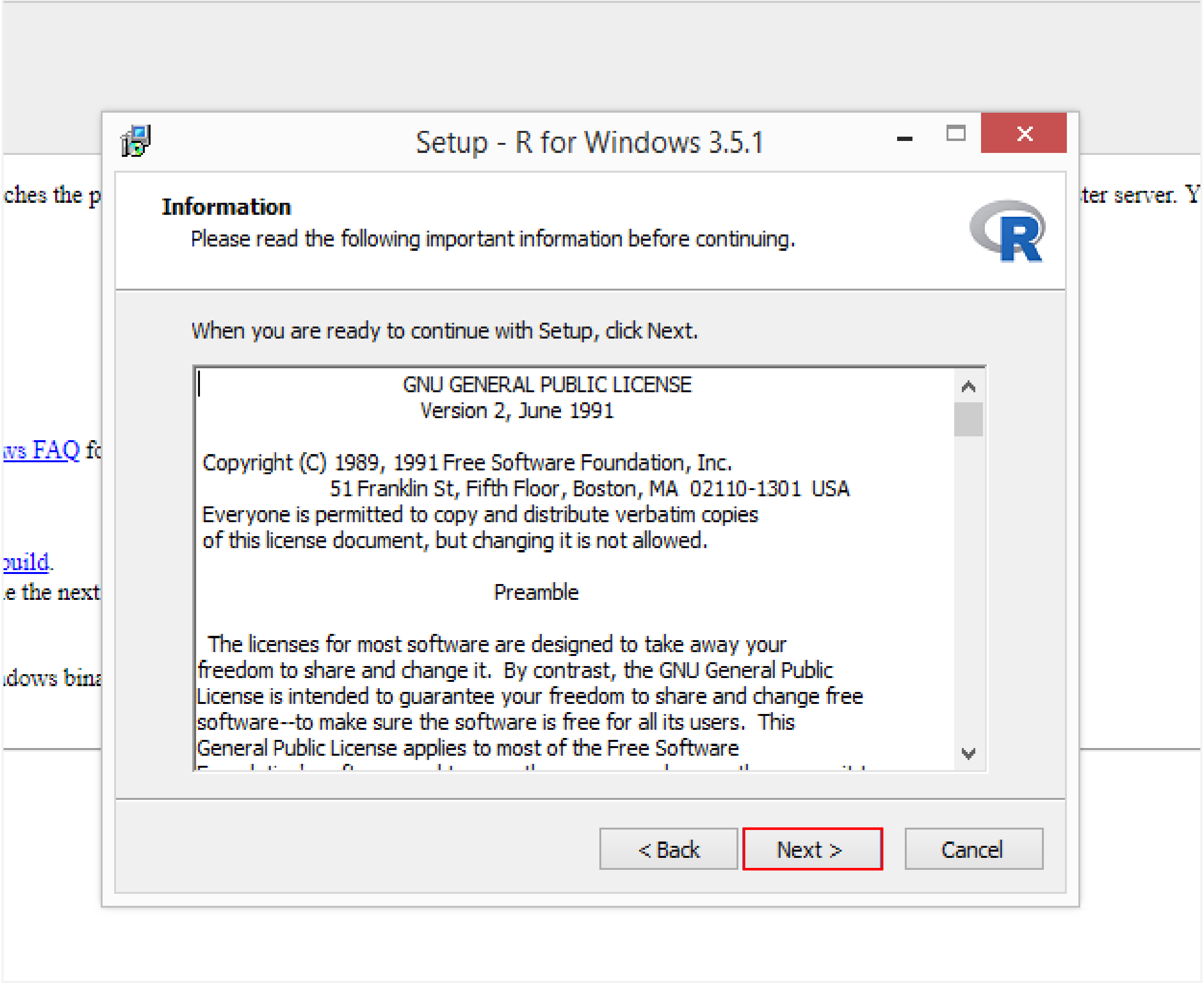


Setup - R for Windows 3.5.1

Information
You should be logged in as an administrator when installing R

Note: A full R installation requires administrative privileges, and it appears that those are not available. If you continue with this installation, you will not be able to associate R with .RData files. Installation must be made to a directory where you have write permission.

The next page provides the Information on R installation. Click on **Next**.



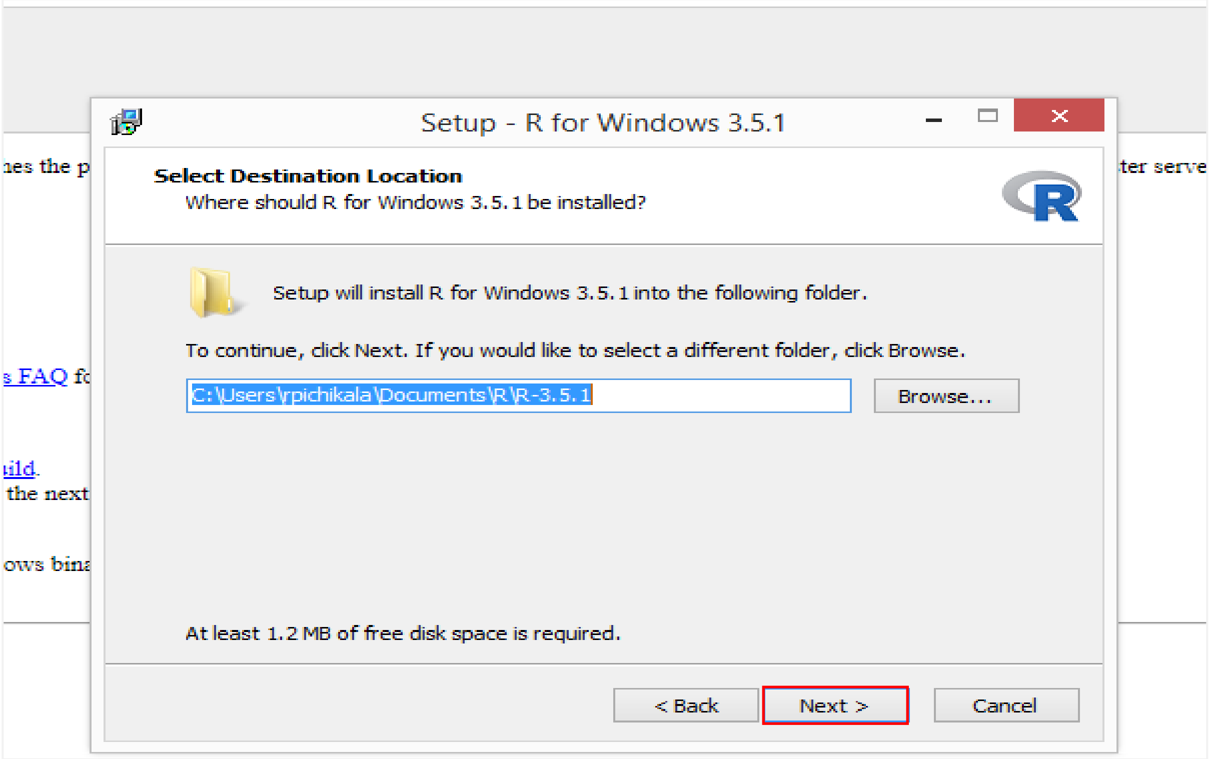
Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

```
                    GNU GENERAL PUBLIC LICENSE
                       Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.
                 51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.  This
General Public License applies to most of the Free Software
```

Select destination location at the top. By default, it will suggest installing R in "**C:\Program Files**" on your computer. Click on **Next** at the bottom of the R Setup wizard window.



Setup - R for Windows 3.5.1

Select Destination Location

Where should R for Windows 3.5.1 be installed?

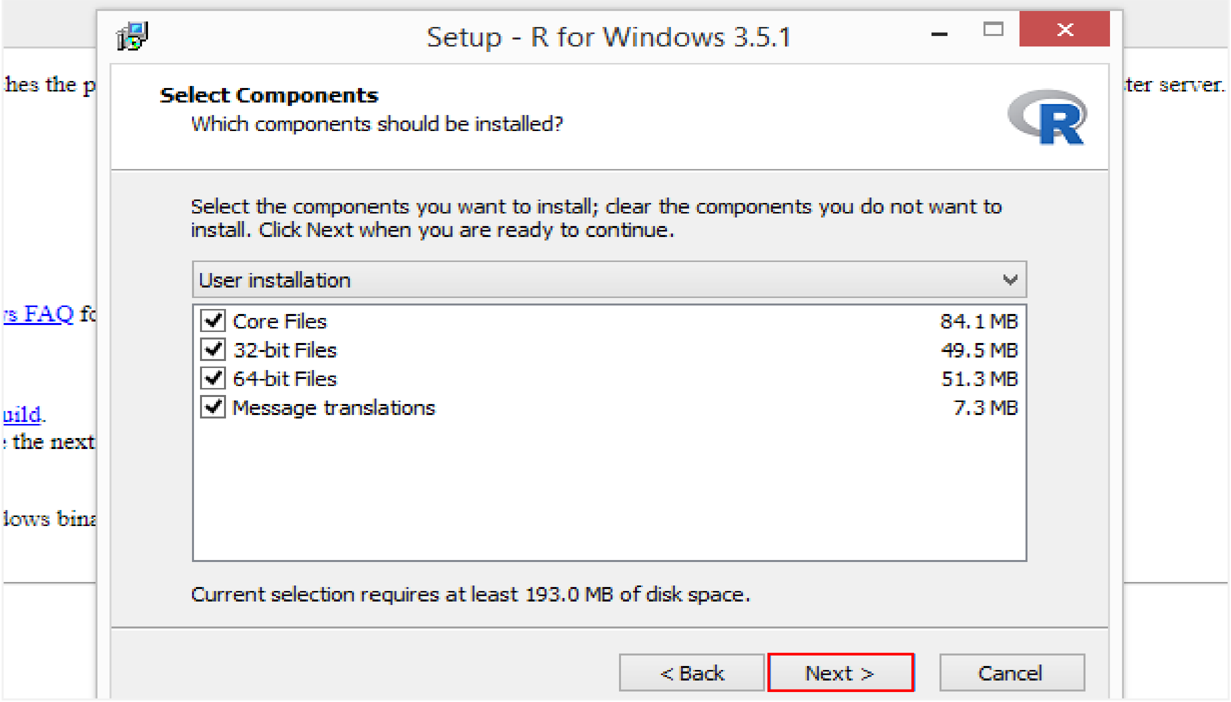
Setup will install R for Windows 3.5.1 into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Users\rpichikala\Documents\R\R-3.5.1

At least 1.2 MB of free disk space is required.

You will be requested to select components at the top. By default all the components will be selected and click on **Next**.



Setup - R for Windows 3.5.1

Select Components

Which components should be installed?

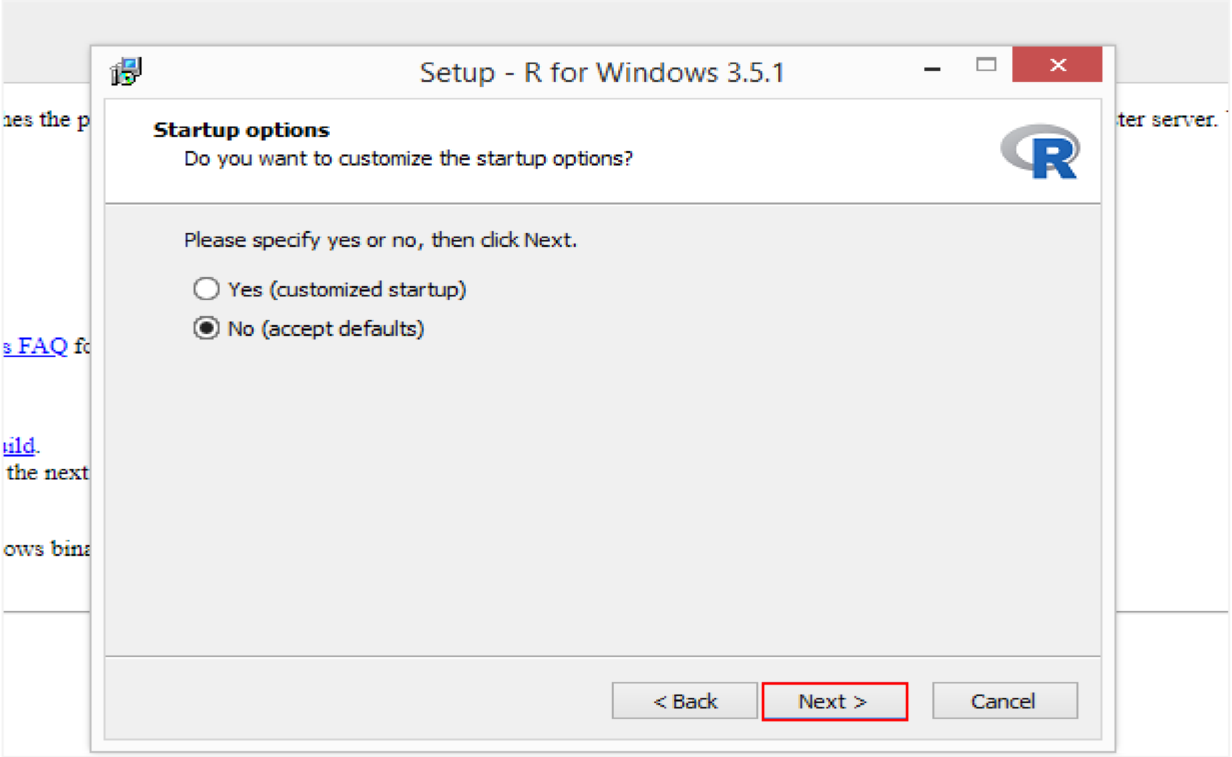
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

User installation

☑ Core Files	84.1 MB
☑ 32-bit Files	49.5 MB
☑ 64-bit Files	51.3 MB
☑ Message translations	7.3 MB

Current selection requires at least 193.0 MB of disk space.

The next step is about Startup options. Select **No** and Click on **Next**.



Startup options

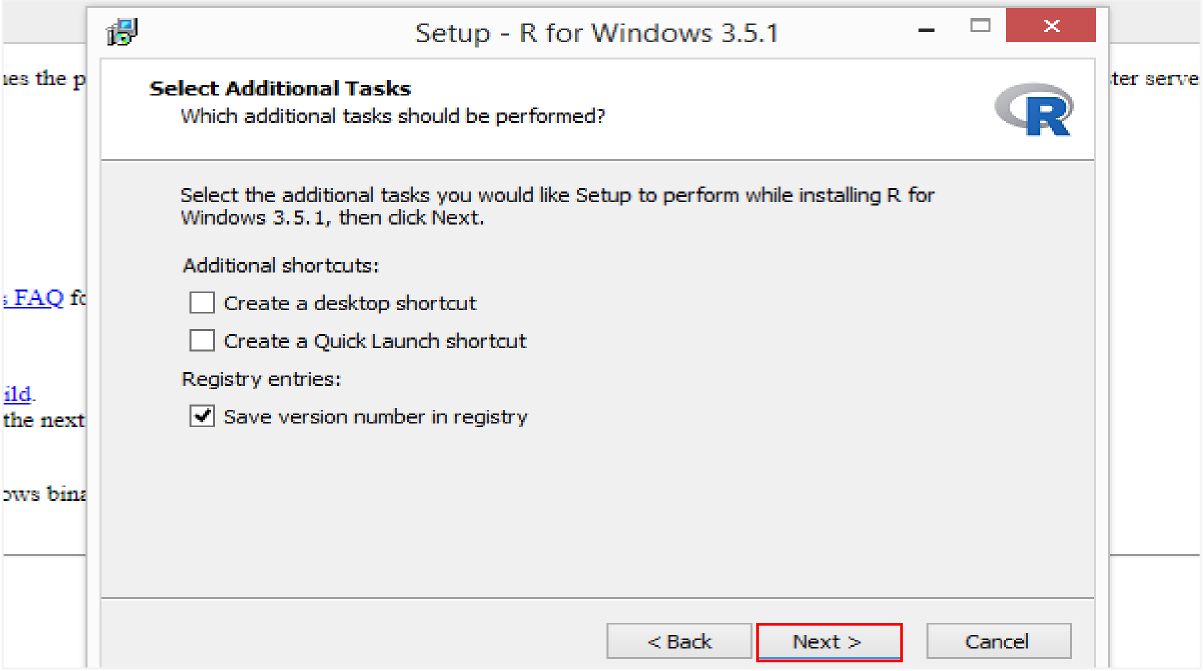
Do you want to customize the startup options?

Please specify yes or no, then click Next.

○ Yes (customized startup)

● No (accept defaults)

Finally, it asks you about additional tasks, click on **Next**.



Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing R for Windows 3.5.1, then click Next.

Additional shortcuts:

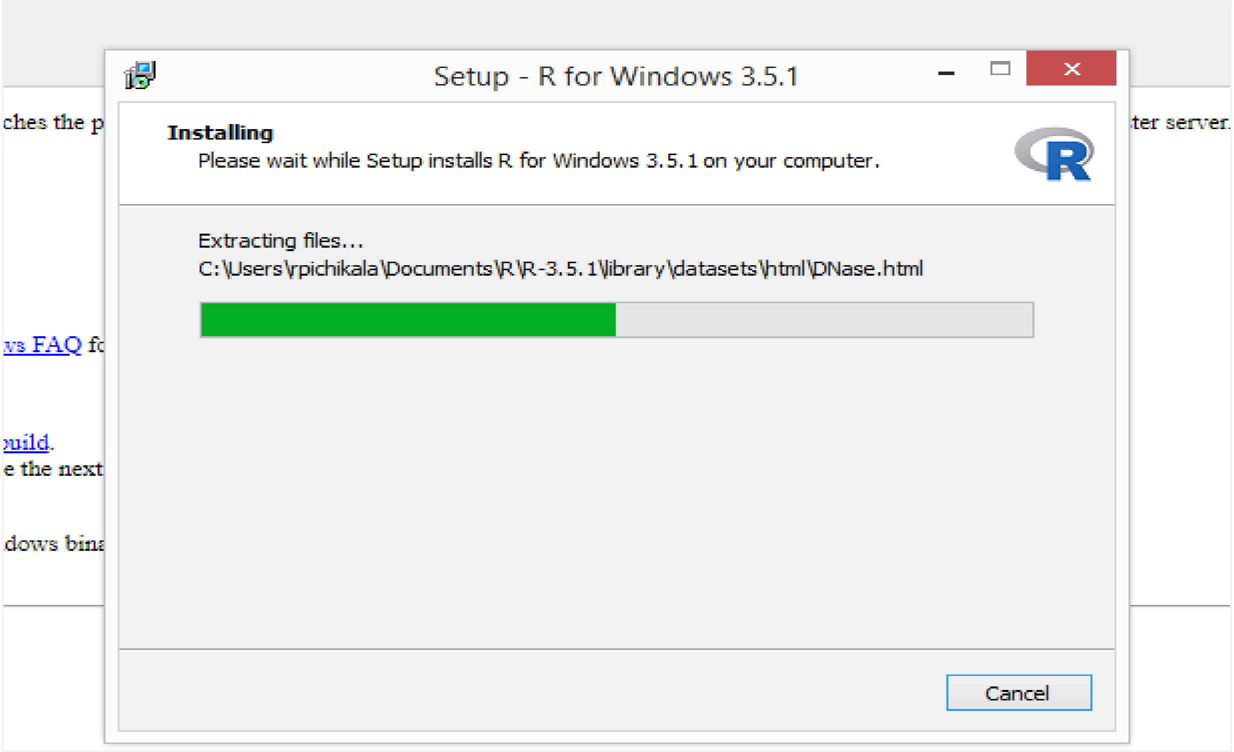
□ Create a desktop shortcut

□ Create a Quick Launch shortcut

Registry entries:

☑ Save version number in registry

R should now be installing. This will take about a minute or two to complete.



Setup - R for Windows 3.5.1

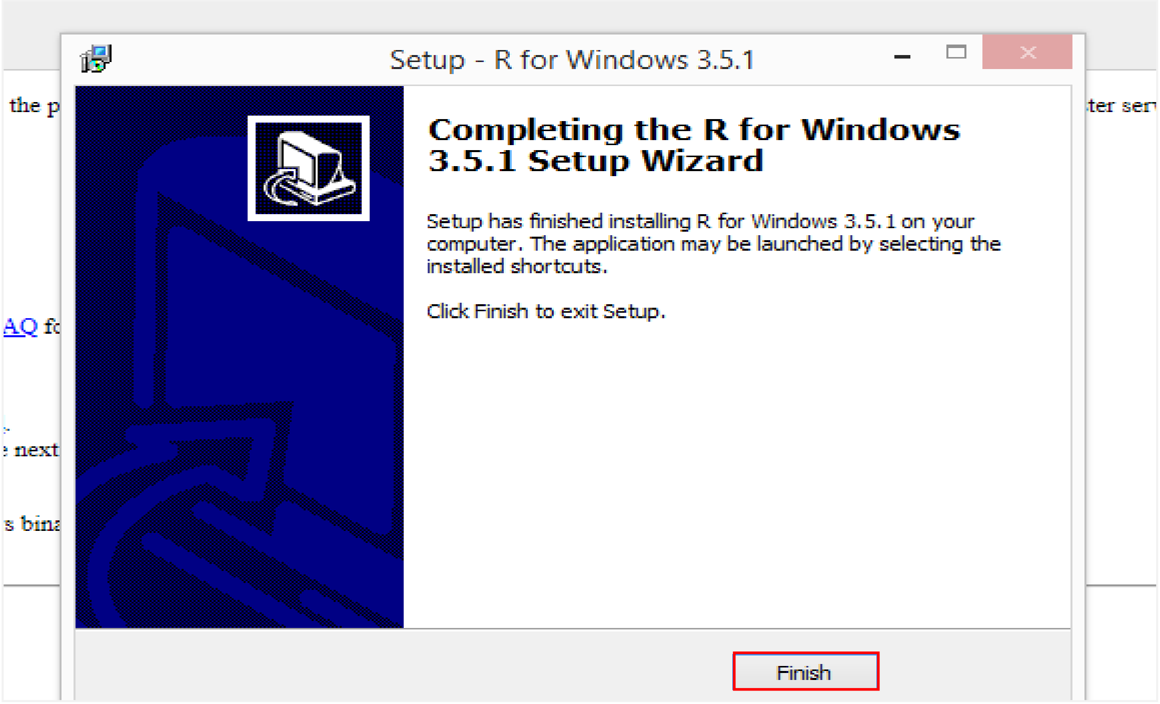
Installing

Please wait while Setup installs R for Windows 3.5.1 on your computer.

Extracting files...

C:\Users\rpichikala\Documents\R\R-3.5.1\library\datasets\html\DNase.html

Now, R for Windows Setup Wizard is completed, click on **Finish**.



Setup - R for Windows 3.5.1

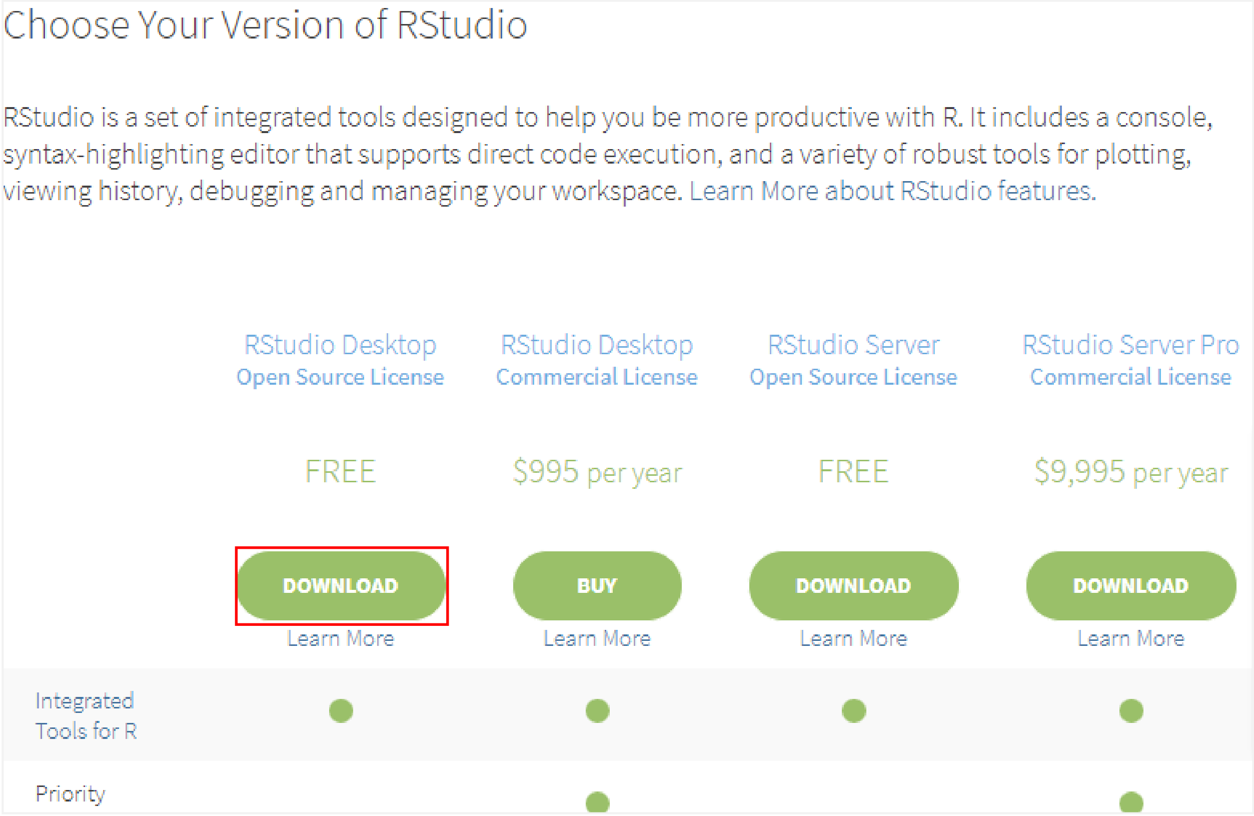
Completing the R for Windows 3.5.1 Setup Wizard

Setup has finished installing R for Windows 3.5.1 on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

Step #2 | Installing RStudio

Go to https://www.rstudio.com/products/rstudio/download/ and download the latest version of RStudio. Click on **Download** button which is under **Free** category in the page.

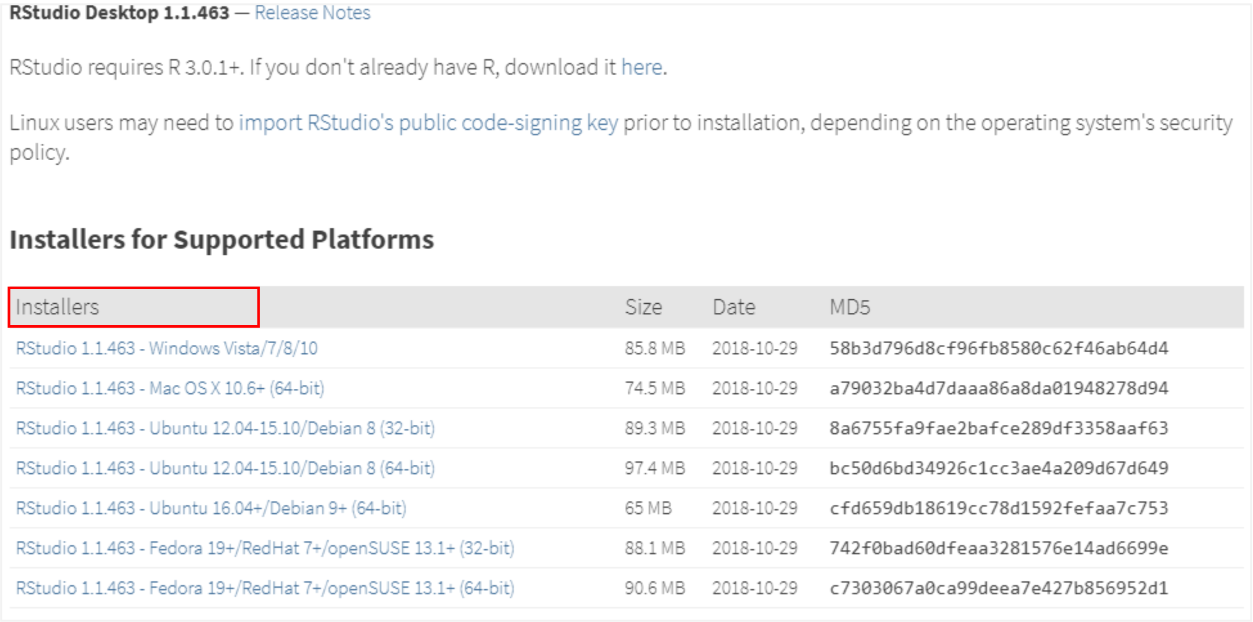


Choose Your Version of RStudio

RStudio is a set of integrated tools designed to help you be more productive with R. It includes a console, syntax-highlighting editor that supports direct code execution, and a variety of robust tools for plotting, viewing history, debugging and managing your workspace. Learn More about RStudio features.

	RStudio Desktop Open Source License	RStudio Desktop Commercial License	RStudio Server Open Source License	RStudio Server Pro Commercial License
	FREE	$995 per year	FREE	$9,995 per year
	DOWNLOAD Learn More	BUY Learn More	DOWNLOAD Learn More	DOWNLOAD Learn More
Integrated Tools for R	●	●	●	●
Priority		●		●

Select the link of the RStudio version which is appropriate to your system. If you are using Windows machine, please choose the first link and if you are using a Mac machine, please choose the second link.



RStudio Desktop 1.1.463 — Release Notes

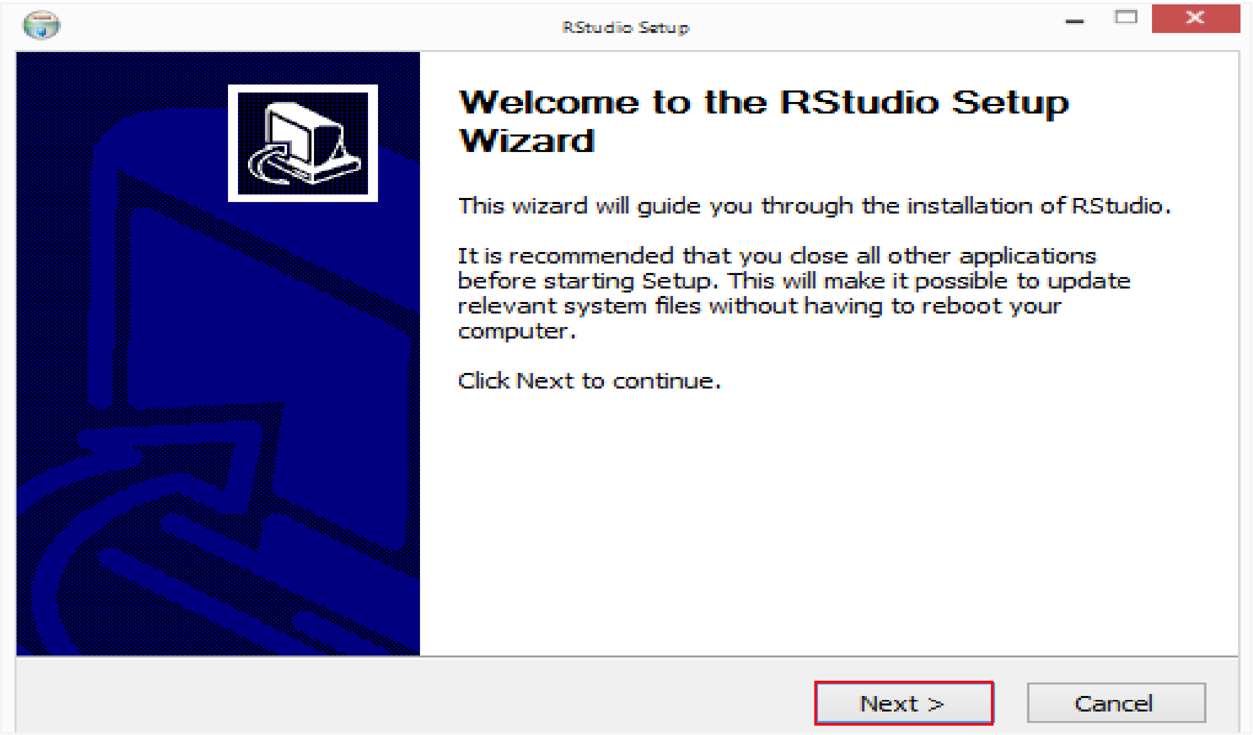
RStudio requires R 3.0.1+. If you don't already have R, download it here.

Linux users may need to import RStudio's public code-signing key prior to installation, depending on the operating system's security policy.

Installers for Supported Platforms

Installers	Size	Date	MD5
RStudio 1.1.463 - Windows Vista/7/8/10	85.8 MB	2018-10-29	58b3d796d8cf96fb8580c62f46ab64d4
RStudio 1.1.463 - Mac OS X 10.6+ (64-bit)	74.5 MB	2018-10-29	a79032ba4d7daaa86a8da01948278d94
RStudio 1.1.463 - Ubuntu 12.04-15.10/Debian 8 (32-bit)	89.3 MB	2018-10-29	8a6755fa9fae2bafce289df3358aaf63
RStudio 1.1.463 - Ubuntu 12.04-15.10/Debian 8 (64-bit)	97.4 MB	2018-10-29	bc50d6bd34926c1cc3ae4a209d67d649
RStudio 1.1.463 - Ubuntu 16.04+/Debian 9+ (64-bit)	65 MB	2018-10-29	cfd659db18619cc78d1592fefaa7c753
RStudio 1.1.463 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	88.1 MB	2018-10-29	742f0bad60dfeaa3281576e14ad6699e
RStudio 1.1.463 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	90.6 MB	2018-10-29	c7303067a0ca99deea7e427b856952d1

Click on **Next** to continue and the install wizard is opened.



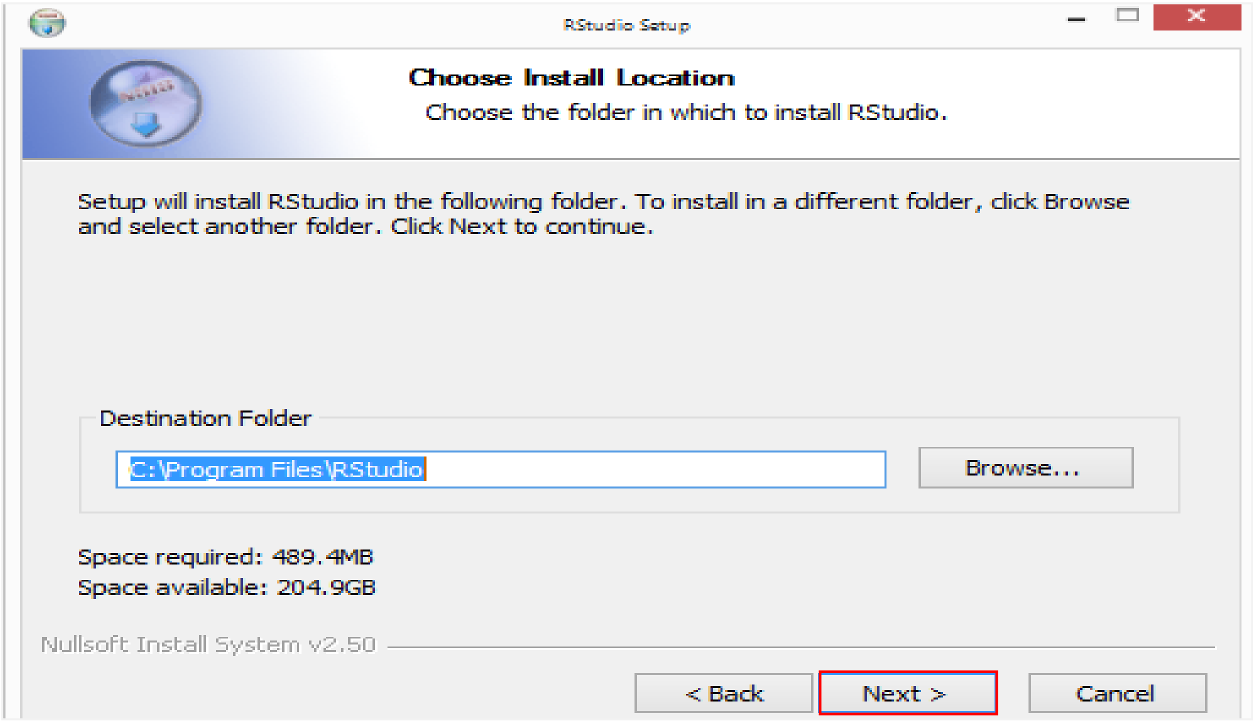
Welcome to the RStudio Setup Wizard

This wizard will guide you through the installation of RStudio.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Choose Install Location, by default it will suggest installing in "**C:\Program Files**" on your computer and click on **Next**.



RStudio Setup

Choose Install Location

Choose the folder in which to install RStudio.

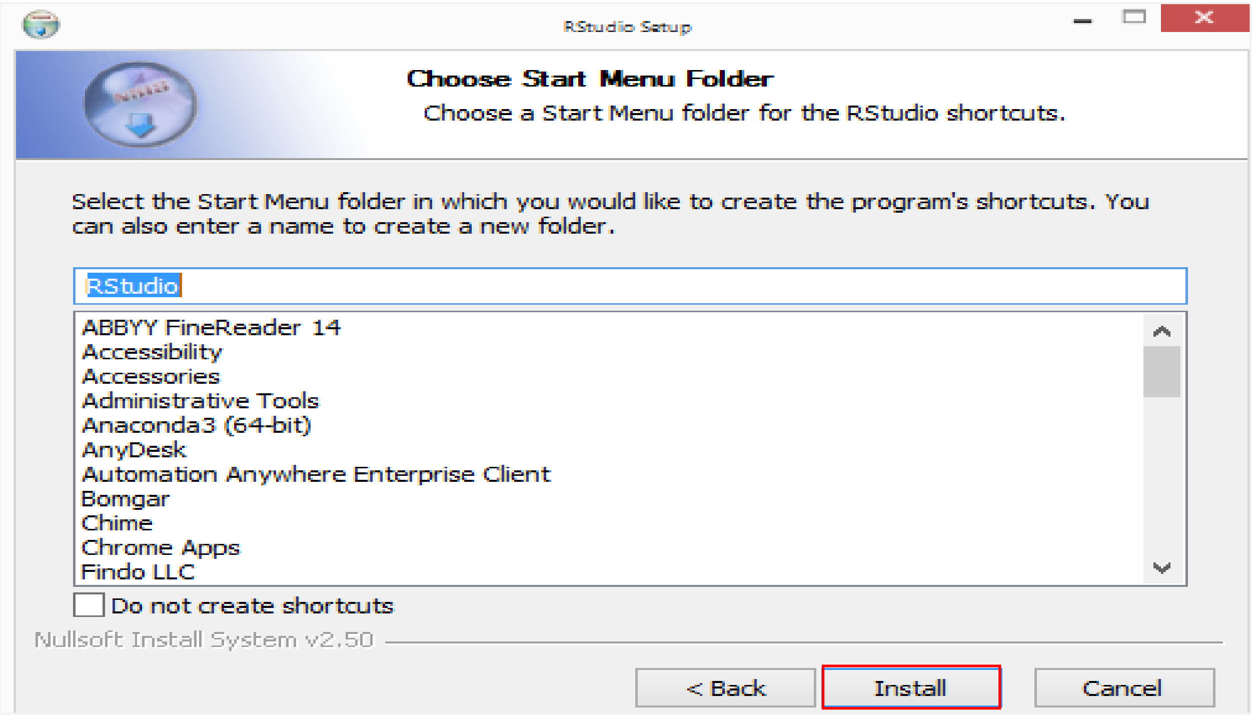
Setup will install RStudio in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder

C:\Program Files\RStudio

Space required: 489.4MB
Space available: 204.9GB

Click on **Install** to accept the default start menu folder and install RStudio.



RStudio Setup

Choose Start Menu Folder

Choose a Start Menu folder for the RStudio shortcuts.

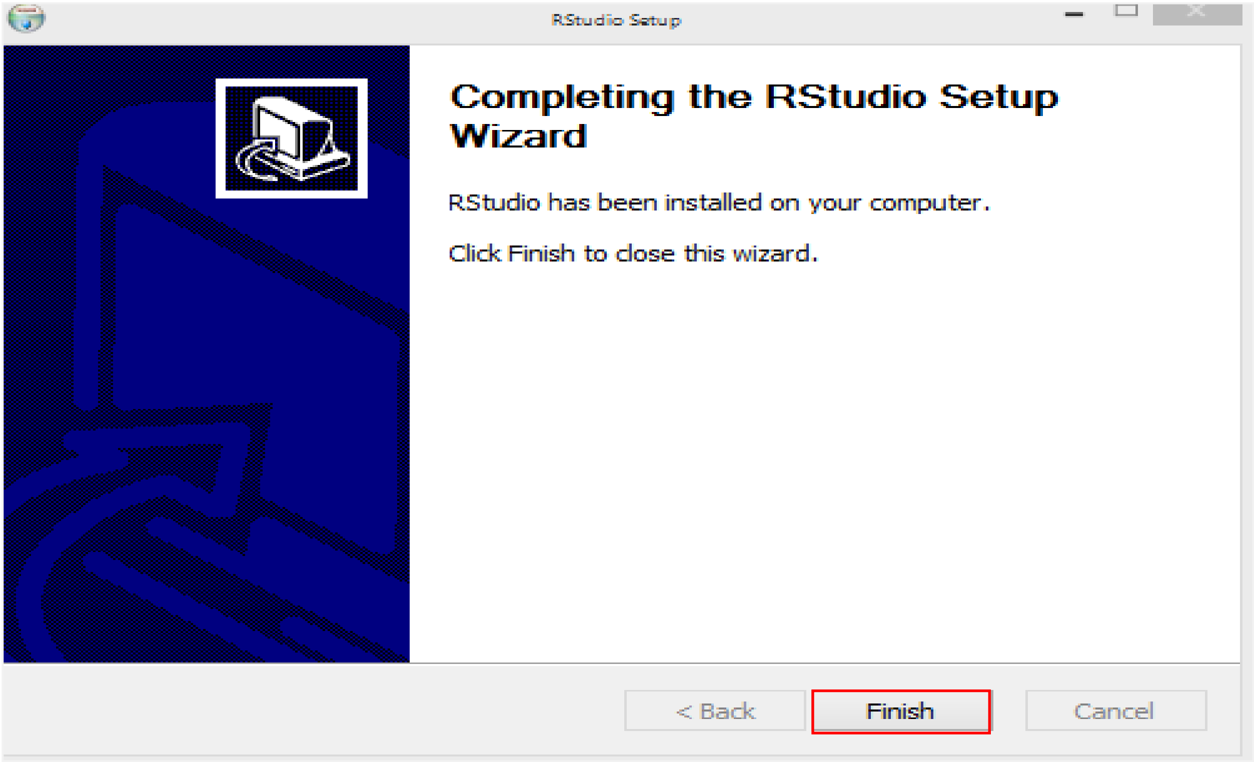
Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

RStudio

ABBYY FineReader 14
Accessibility
Accessories
Administrative Tools
Anaconda3 (64-bit)
AnyDesk
Automation Anywhere Enterprise Client
Bomgar
Chime
Chrome Apps
Findo LLC

□ Do not create shortcuts

Click on **Finish** to close the wizard.



Completing the RStudio Setup Wizard

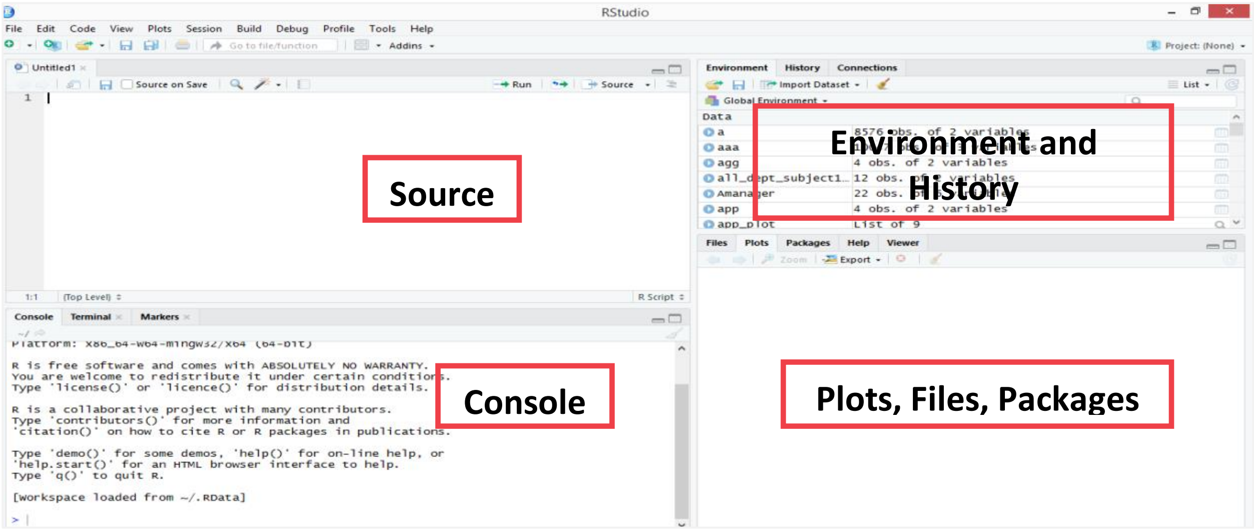
RStudio has been installed on your computer.

Click Finish to close this wizard.

Glance on RStudio

RStudio is the IDE to run R scripts. There are four main components in IDE - Source, Console, Environment and History and Plots/Files/Packages.

Source - we write the main program in the source area.
Console - The output and errors are displayed in console.



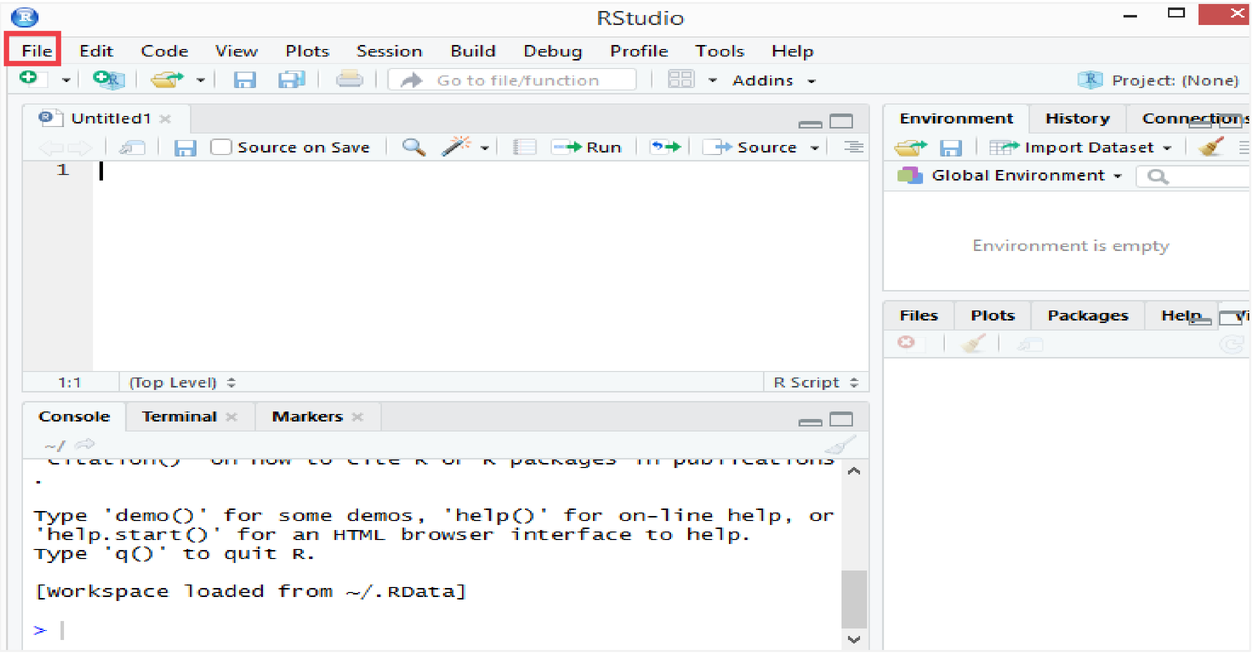
```
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

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.

[workspace loaded from ~/.RData]

> 
```

Environment and History - Data Frames and Lists are displayed in environment and provides interactive list of loaded R objects.

Plots, Files, Packages - Graphs are displayed in the plot section, we can see our installed packages under packages tab and also, we can search for any function with the help tab.

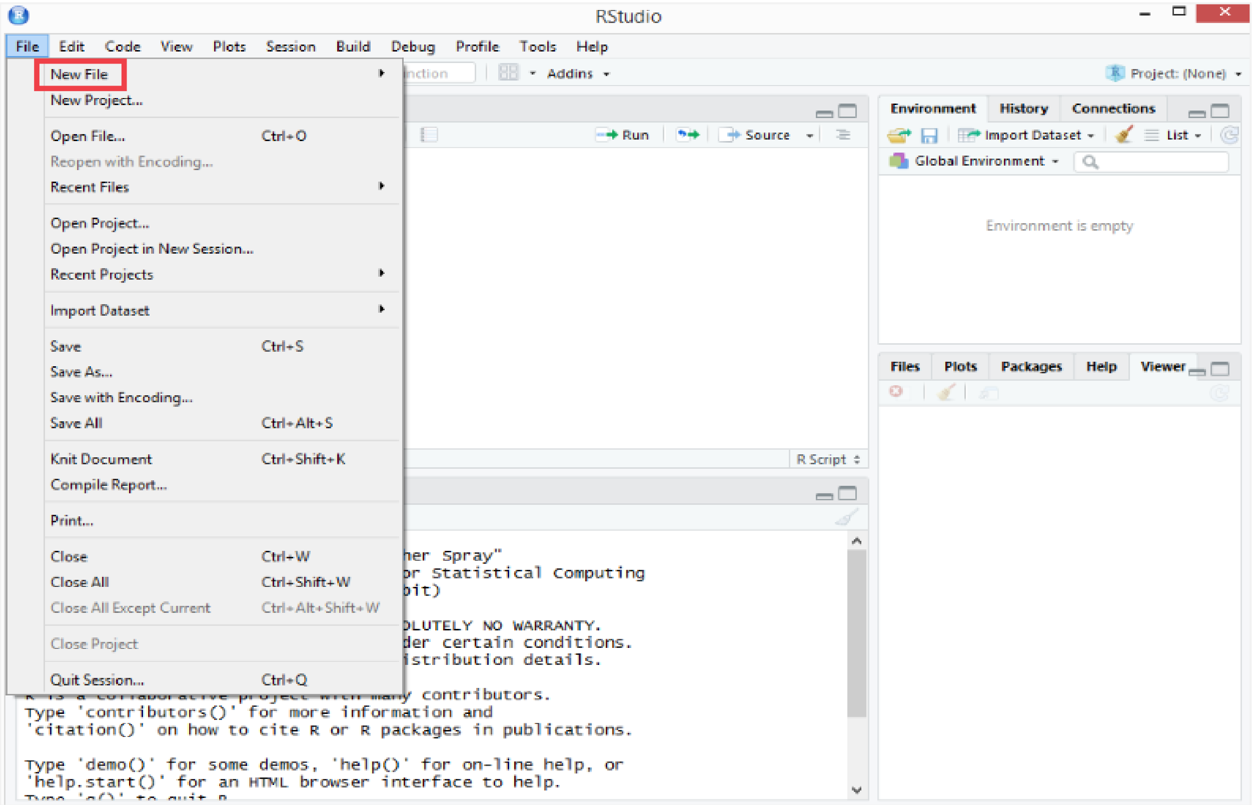
Step #3 | Create R Script

Open R Studio and go to **file** option which is at the top left corner of the RStudio.



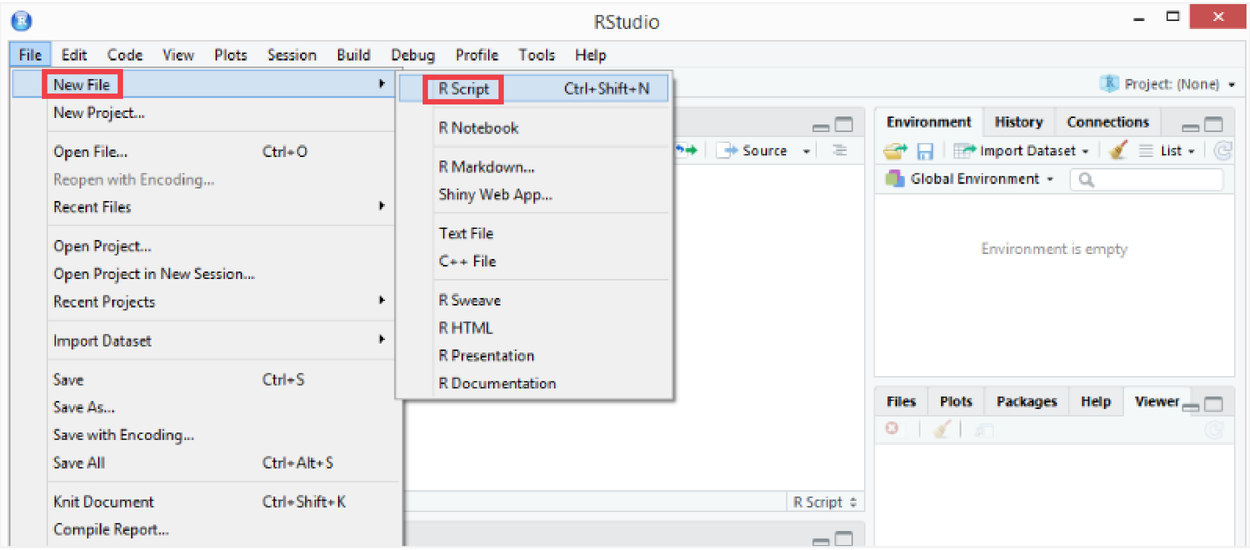
```
citation() on how to cite R or R packages in publications
.

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.

[Workspace loaded from ~/.RData]

>
```

Select **New File** to create R script.

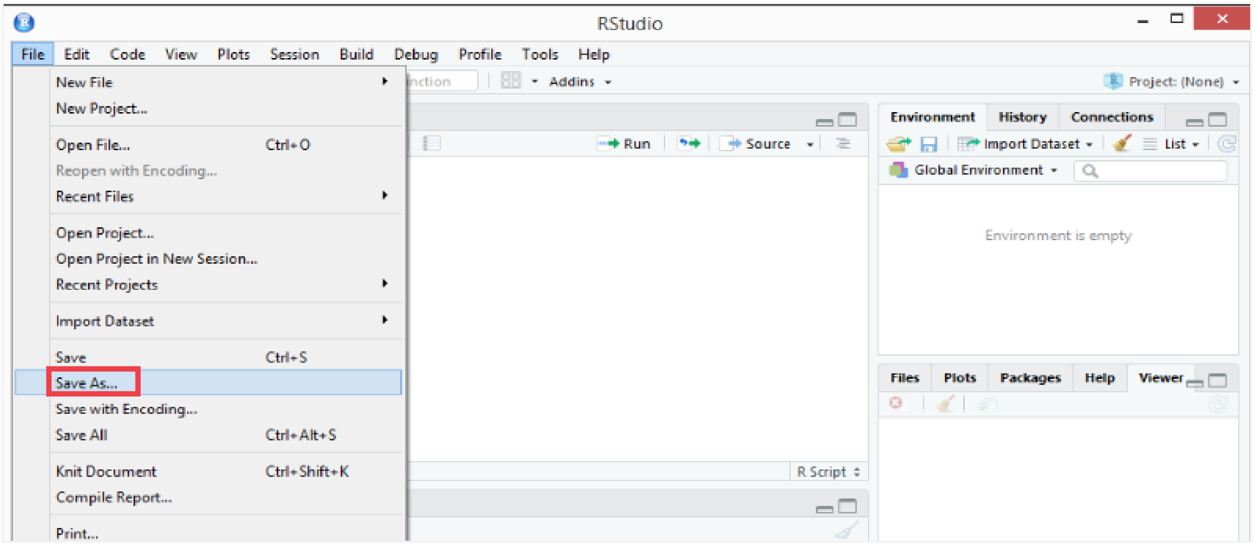


```
her Spray"
or Statistical Computing
bit)

OLUTELY NO WARRANTY.
der certain conditions.
istribution details.

contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
```

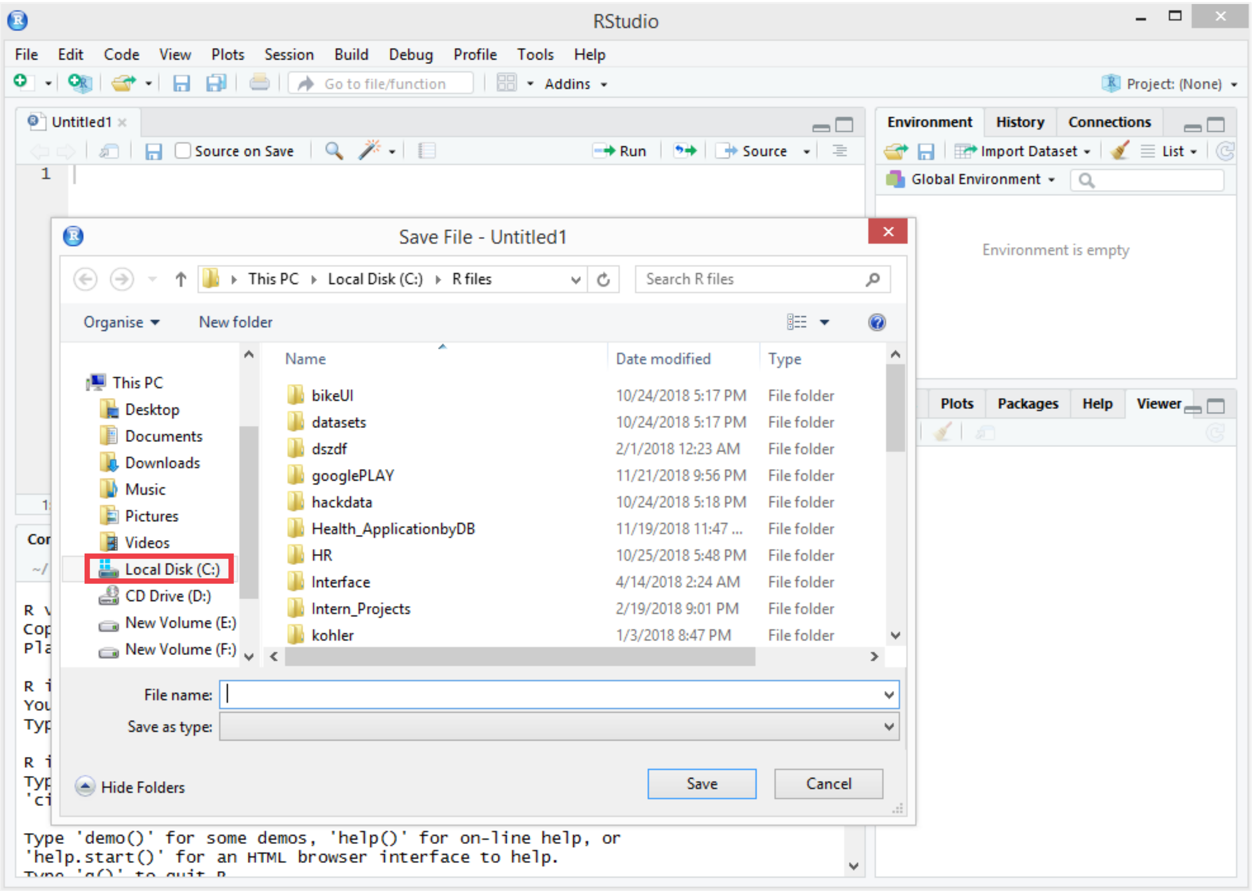
After selecting **New File** there, you will find multiple options. Select **R Script** which is at the top.



Now your R script will be created with the name **Untitled1.** To save R script, go to file and select **Save As** option.



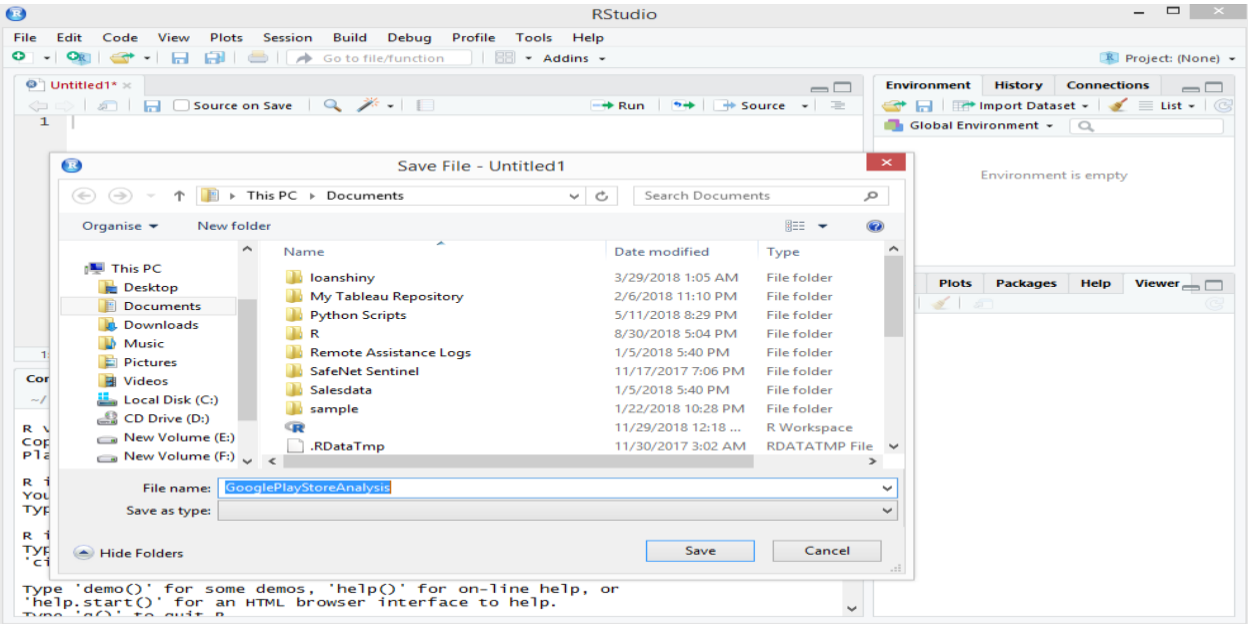
Choose the location of the directory in which you want to save.



Name	Date modified	Type
bikeUI	10/24/2018 5:17 PM	File folder
datasets	10/24/2018 5:17 PM	File folder
dszdf	2/1/2018 12:23 AM	File folder
googlePLAY	11/21/2018 9:56 PM	File folder
hackdata	10/24/2018 5:18 PM	File folder
Health_ApplicationbyDB	11/19/2018 11:47 ...	File folder
HR	10/25/2018 5:48 PM	File folder
Interface	4/14/2018 2:24 AM	File folder
Intern_Projects	2/19/2018 9:01 PM	File folder
kohler	1/3/2018 8:47 PM	File folder

```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
```

Enter the name you want to give to your R Script in **File name** field and click on **Save** option.



Name	Date modified	Type
loanshiny	3/29/2018 1:05 AM	File folder
My Tableau Repository	2/6/2018 11:10 PM	File folder
Python Scripts	5/11/2018 8:29 PM	File folder
R	8/30/2018 5:04 PM	File folder
Remote Assistance Logs	1/5/2018 5:40 PM	File folder
SafeNet Sentinel	11/17/2017 7:06 PM	File folder
Salesdata	1/5/2018 5:40 PM	File folder
sample	1/22/2018 10:28 PM	File folder
	11/29/2018 12:18 ...	R Workspace
.RDataTmp	11/30/2017 3:02 AM	RDATATMP File

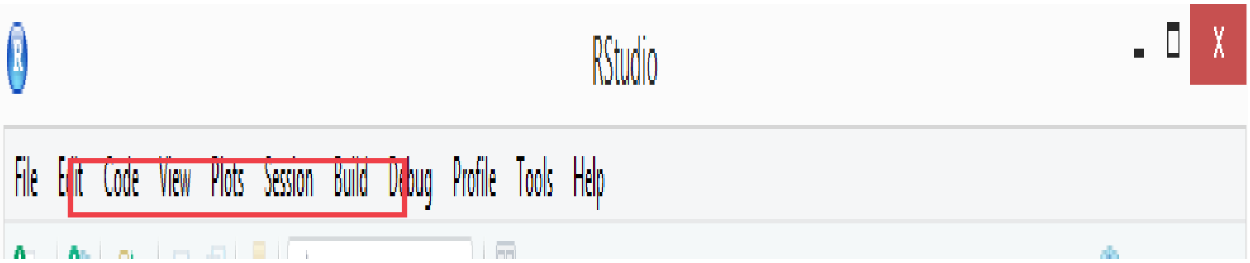
File name: GooglePlayStoreAnalysis

```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
```

Step #4 | Setting up your Directory

Set your data file location inside setwd function.

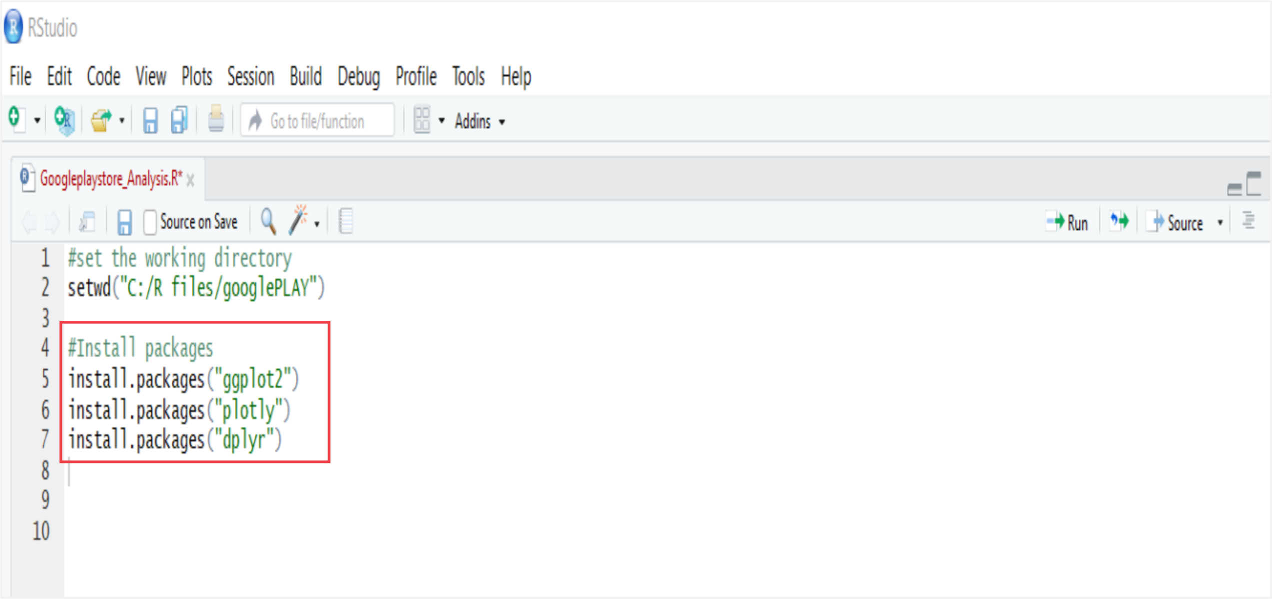
setwd(“C:/R files/googlePLAY”)


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help

Step #5 | Install Packages

Install the libraries which are necessary for our R application.

install.packages("ggplot2")
install.packages("plotly")
install.packages("dplyr")

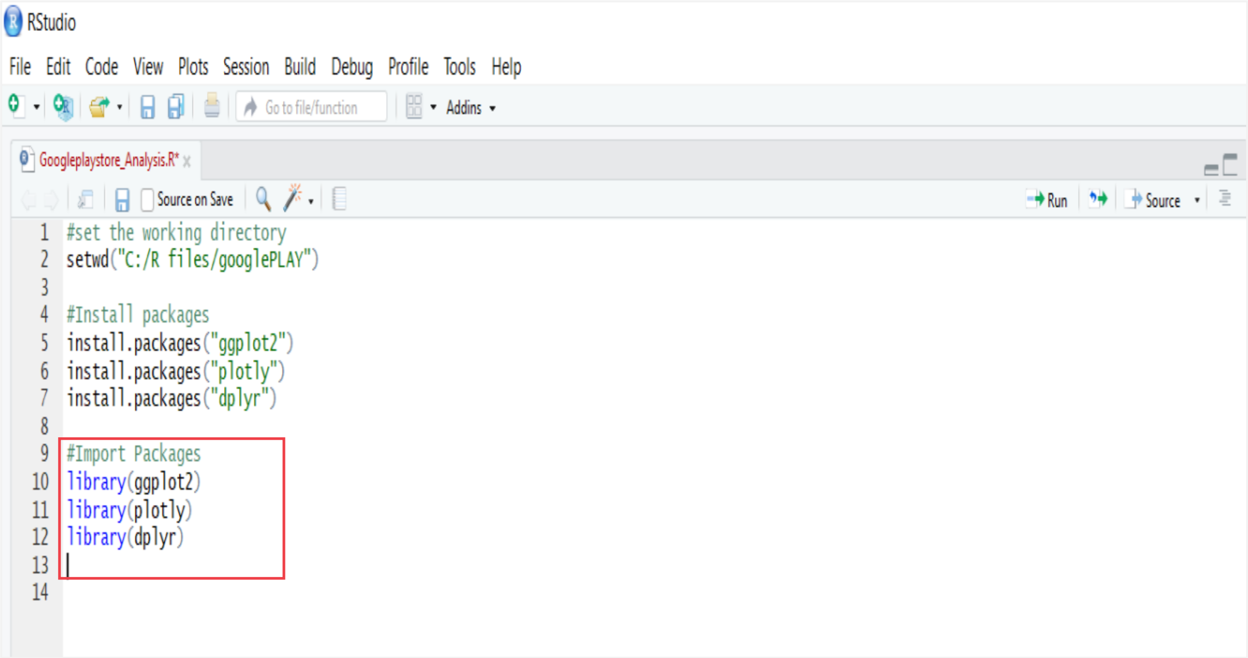

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Googleplaystore_Analysis.R*
Source on Save
Run
Source

```
#set the working directory
setwd("C:/R files/googlePLAY")

#Install packages
install.packages("ggplot2")
install.packages("plotly")
install.packages("dplyr")
```

Step #6 | Load Packages

Load the installed packages through libraries.

library(ggplot2)
library(plotly)
library(dplyr)

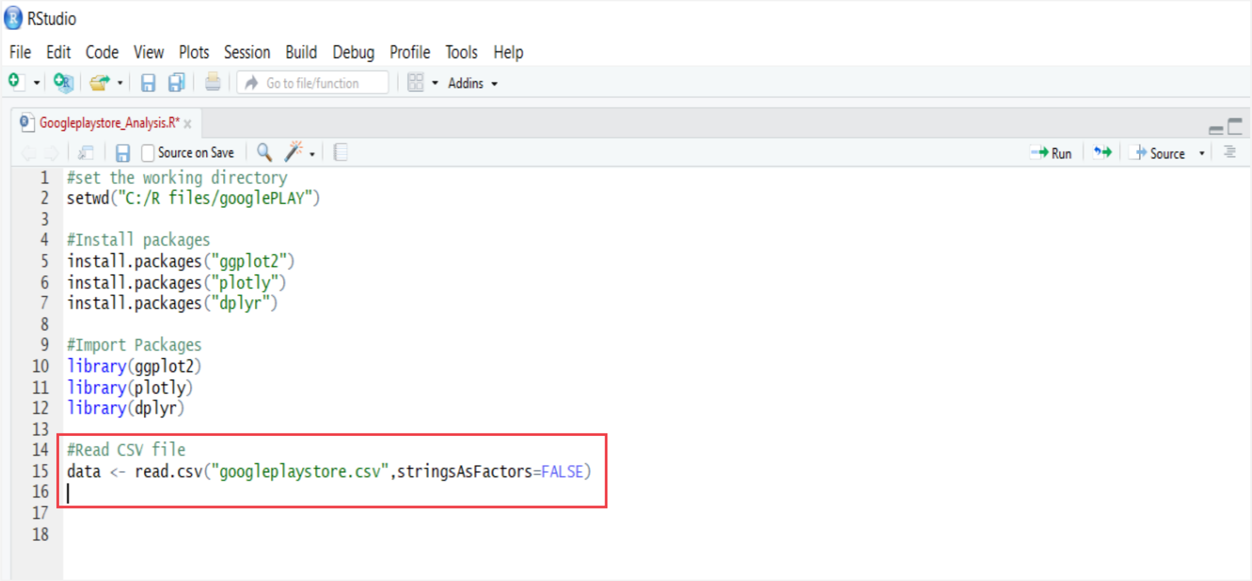

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Googleplaystore_Analysis.R*
Source on Save
Run
Source

```
#set the working directory
setwd("C:/R files/googlePLAY")

#Install packages
install.packages("ggplot2")
install.packages("plotly")
install.packages("dplyr")

#Import Packages
library(ggplot2)
library(plotly)
library(dplyr)
```

Step #7 | Import Data into R

Import the dataset to R environment and store it in a variable. Here, we load our data from the csv file into a variable called “data”.

data <- read.csv("googleplaystore.csv",stringsAsFactors=FALSE)

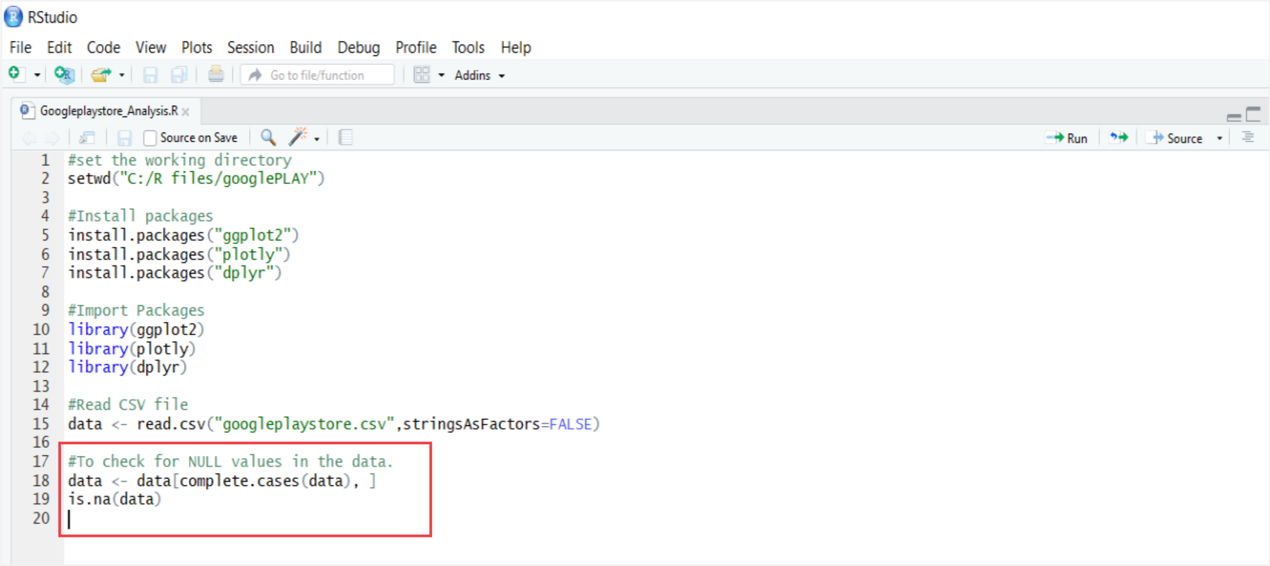

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Googleplaystore_Analysis.R*
Source on Save
Run
Source

```r
#set the working directory
setwd("C:/R files/googlePLAY")

#Install packages
install.packages("ggplot2")
install.packages("plotly")
install.packages("dplyr")

#Import Packages
library(ggplot2)
library(plotly)
library(dplyr)

#Read CSV file
data <- read.csv("googleplaystore.csv",stringsAsFactors=FALSE)
```

Step #8 | Data Cleansing Process

Check for null values in the data.

data <- data[complete.cases(data),]


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Googleplaystore_Analysis.R
Source on Save
Run
Source

```r
#set the working directory
setwd("C:/R files/googlePLAY")

#Install packages
install.packages("ggplot2")
install.packages("plotly")
install.packages("dplyr")

#Import Packages
library(ggplot2)
library(plotly)
library(dplyr)

#Read CSV file
data <- read.csv("googleplaystore.csv",stringsAsFactors=FALSE)

#To check for NULL values in the data.
data <- data[complete.cases(data), ]
is.na(data)
```

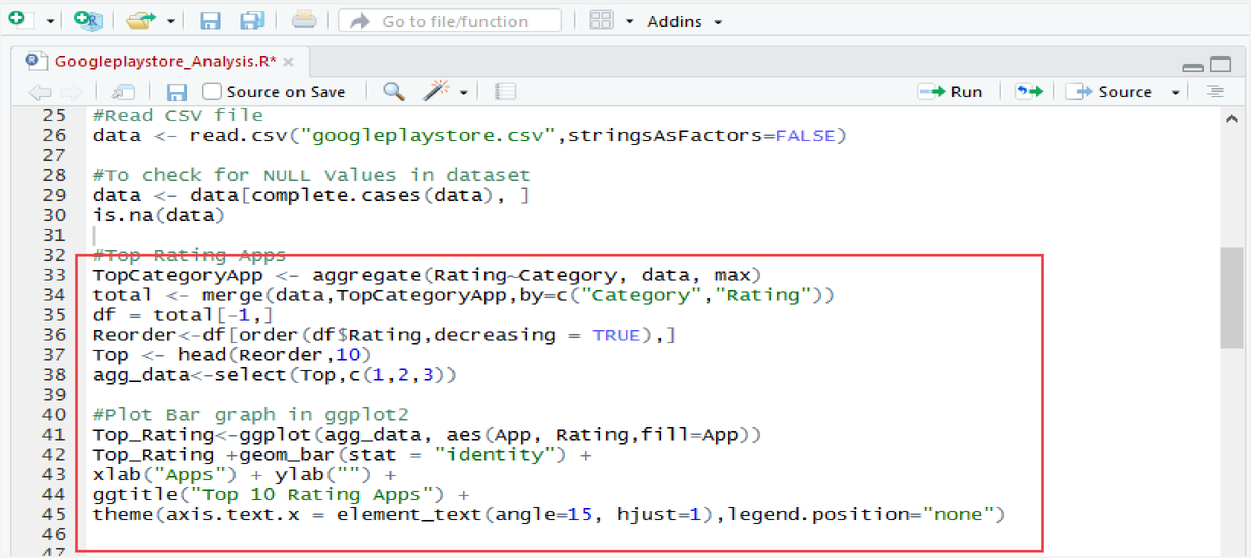
Step #9 | Find out Top Rating Apps

Data Preparation - Inorder to perform data analysis, we need to prepare the data according to our requirements.

```
TopCategoryApp <- aggregate(Rating~Category, data, max)
total <- merge(data,TopCategoryApp,by=c("Category","Rating"))
df = total[-1,]
Reorder<-df[order(df$Rating,decreasing = TRUE),]
Top <- head(Reorder,10)
agg_data<-select(Top,c(1,2,3))
```

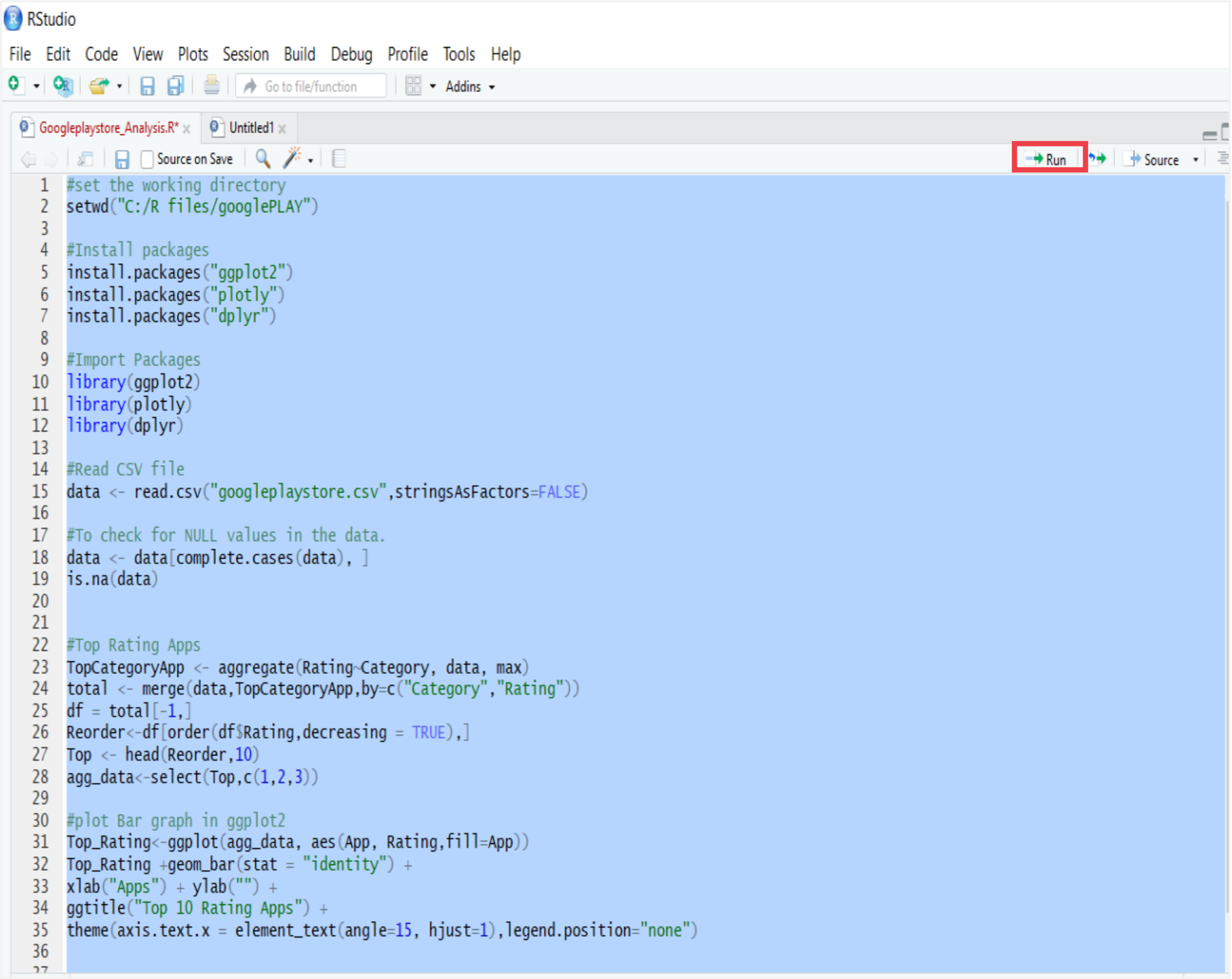
Visualize the resulted data in the form of graphs.

```
Top_Rating<-ggplot(agg_data, aes(App, Rating,fill=App))
Top_Rating +geom_bar(stat = "identity") +
xlab("Apps") + ylab("") +
ggtitle("Top 10 Rating Apps") +
theme(axis.text.x = element_text(angle=15, hjust=1),legend.position="none")
```



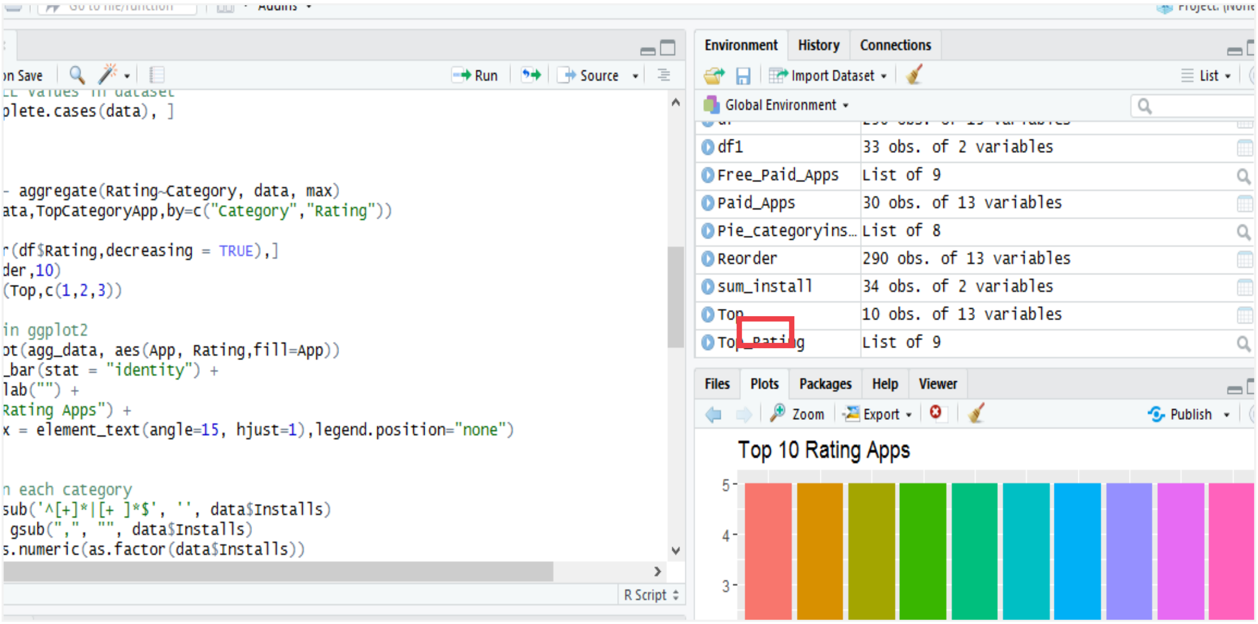
```
#Read CSV file
data <- read.csv("googleplaystore.csv",stringsAsFactors=FALSE)

#To check for NULL values in dataset
data <- data[complete.cases(data), ]
is.na(data)

#Top Rating Apps
TopCategoryApp <- aggregate(Rating~Category, data, max)
total <- merge(data,TopCategoryApp,by=c("Category","Rating"))
df = total[-1,]
Reorder<-df[order(df$Rating,decreasing = TRUE),]
Top <- head(Reorder,10)
agg_data<-select(Top,c(1,2,3))

#Plot Bar graph in ggplot2
Top_Rating<-ggplot(agg_data, aes(App, Rating,fill=App))
Top_Rating +geom_bar(stat = "identity") +
xlab("Apps") + ylab("") +
ggtitle("Top 10 Rating Apps") +
theme(axis.text.x = element_text(angle=15, hjust=1),legend.position="none")
```

To run commands in R script, select the commands you wish to execute and click on **Run** button.


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Googleplaystore_Analysis.R*
Untitled1
Source on Save
Run
Source

```r
#set the working directory
setwd("C:/R files/googlePLAY")

#Install packages
install.packages("ggplot2")
install.packages("plotly")
install.packages("dplyr")

#Import Packages
library(ggplot2)
library(plotly)
library(dplyr)

#Read CSV file
data <- read.csv("googleplaystore.csv",stringsAsFactors=FALSE)

#To check for NULL values in the data.
data <- data[complete.cases(data), ]
is.na(data)


#Top Rating Apps
TopCategoryApp <- aggregate(Rating~Category, data, max)
total <- merge(data,TopCategoryApp,by=c("Category","Rating"))
df = total[-1,]
Reorder<-df[order(df$Rating,decreasing = TRUE),]
Top <- head(Reorder,10)
agg_data<-select(Top,c(1,2,3))

#plot Bar graph in ggplot2
Top_Rating<-ggplot(agg_data, aes(App, Rating,fill=App))
Top_Rating +geom_bar(stat = "identity") +
xlab("Apps") + ylab("") +
ggtitle("Top 10 Rating Apps") +
theme(axis.text.x = element_text(angle=15, hjust=1),legend.position="none")
```

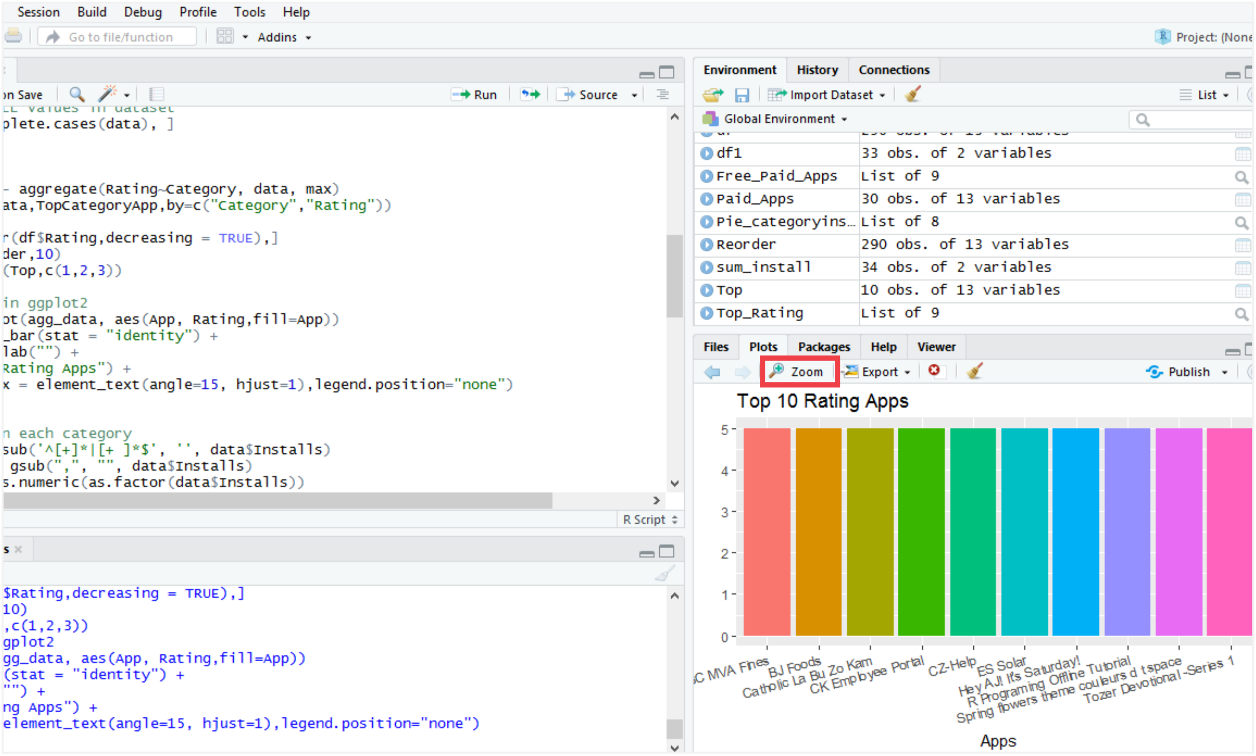
The graphs will be displayed under **Plots** tab which is at the bottom right corner of the R Studio.



```
LL values in dataset
plete.cases(data), ]


- aggregate(Rating~Category, data, max)
ata,TopCategoryApp,by=c("Category","Rating"))

r(df$Rating,decreasing = TRUE),]
der,10)
(Top,c(1,2,3))

in ggplot2
ot(agg_data, aes(App, Rating,fill=App))
_bar(stat = "identity") +
lab("") +
Rating Apps") +
x = element_text(angle=15, hjust=1),legend.position="none")


n each category
sub('^[+]*|[+ ]*$', '', data$Installs)
 gsub(",", "", data$Installs)
s.numeric(as.factor(data$Installs))
```

```
df1               33 obs. of 2 variables
Free_Paid_Apps    List of 9
Paid_Apps         30 obs. of 13 variables
Pie_categoryins…  List of 8
Reorder           290 obs. of 13 variables
sum_install       34 obs. of 2 variables
Top               10 obs. of 13 variables
Top_Rating        List of 9
```

You can select **Zoom** option as shown below to view the graph in a separate window.

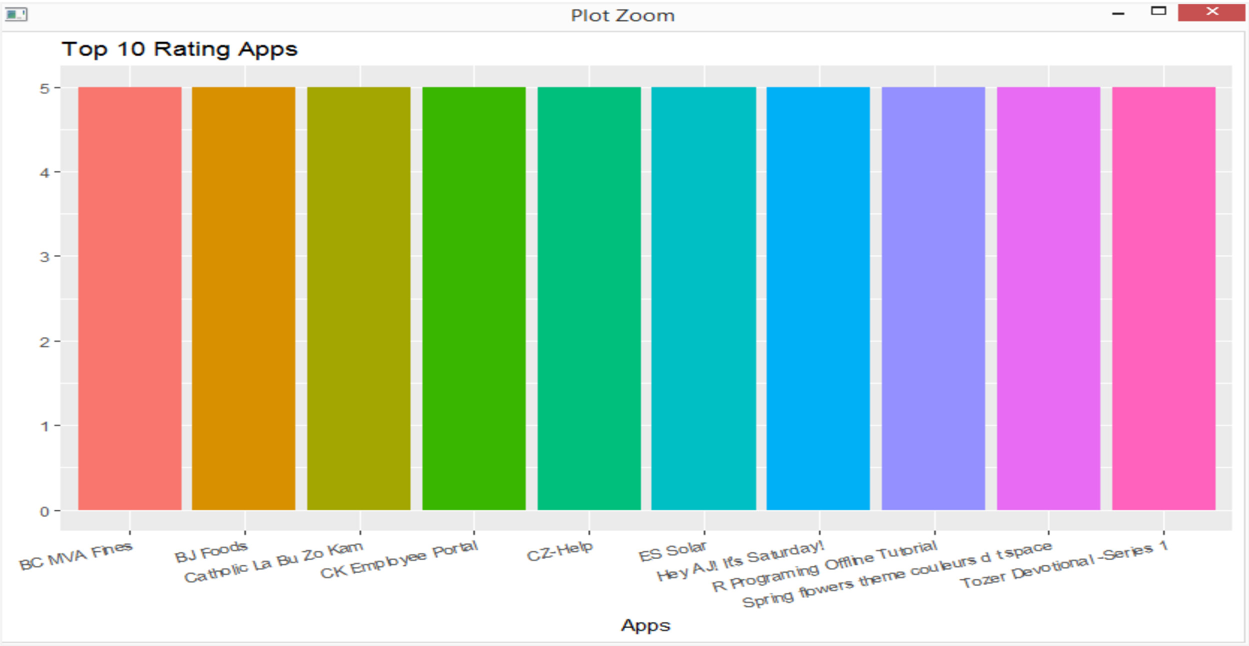


```
LL values in dataset
plete.cases(data), ]


- aggregate(Rating~Category, data, max)
ata,TopCategoryApp,by=c("Category","Rating"))

r(df$Rating,decreasing = TRUE),]
der,10)
(Top,c(1,2,3))

in ggplot2
ot(agg_data, aes(App, Rating,fill=App))
_bar(stat = "identity") +
lab("") +
Rating Apps") +
x = element_text(angle=15, hjust=1),legend.position="none")


n each category
sub('^[+]*|[+ ]*$', '', data$Installs)
 gsub(",", "", data$Installs)
s.numeric(as.factor(data$Installs))
```

```
df1               33 obs. of 2 variables
Free_Paid_Apps    List of 9
Paid_Apps         30 obs. of 13 variables
Pie_categoryins…  List of 8
Reorder           290 obs. of 13 variables
sum_install       34 obs. of 2 variables
Top               10 obs. of 13 variables
Top_Rating        List of 9
```

```
$Rating,decreasing = TRUE),]
10)
,c(1,2,3))
gplot2
gg_data, aes(App, Rating,fill=App))
(stat = "identity") +
"") +
ng Apps") +
element_text(angle=15, hjust=1),legend.position="none")
```

Below is the resulted ggplot graph for Top Rating Apps.

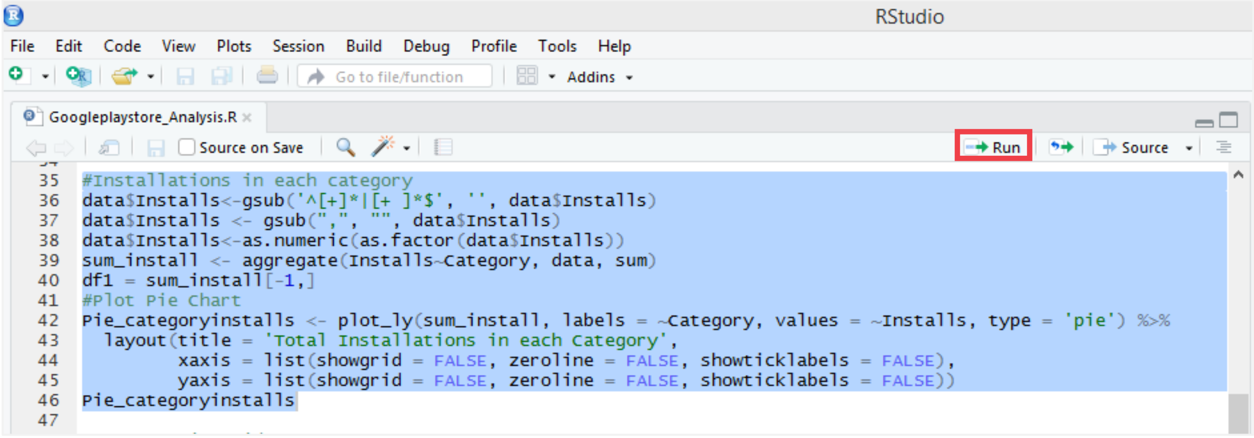


Step #10 | Find out Installations Count in each Category

Calculate number of installations in each category.

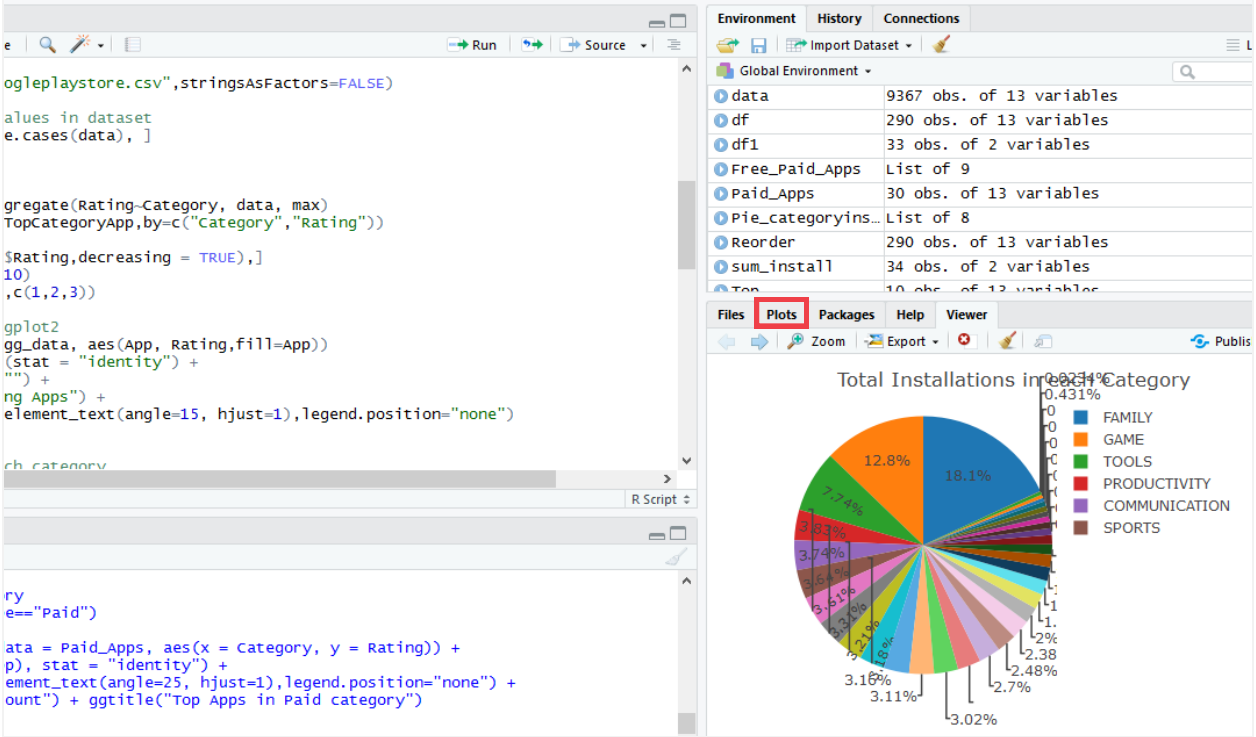
```
data$Installs<-gsub('^[+]*|[+ ]*$', '', data$Installs)
data$Installs <- gsub(",", "", data$Installs)
data$Installs<-as.numeric(as.factor(data$Installs))
sum_install <- aggregate(Installs~Category, data, sum)
df1 = sum_install[-1,]
#Plot Pie Chart
Pie_categoryinstalls <- plot_ly(sum_install, labels = ~Category, values = ~Installs,
type = 'pie') %>%
layout(title = 'Total Installations in each Category',
xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))
Pie_categoryinstalls
```

To run commands in R script, select the commands you wish to execute and click on **Run** button.


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Googleplaystore_Analysis.R
Source on Save
Run
Source

```
#Installations in each category
data$Installs<-gsub('^[+]*|[+ ]*$', '', data$Installs)
data$Installs <- gsub(",", "", data$Installs)
data$Installs<-as.numeric(as.factor(data$Installs))
sum_install <- aggregate(Installs~Category, data, sum)
df1 = sum_install[-1,]
#Plot Pie Chart
Pie_categoryinstalls <- plot_ly(sum_install, labels = ~Category, values = ~Installs, type = 'pie') %>%
  layout(title = 'Total Installations in each Category',
         xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
         yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))
Pie_categoryinstalls
```

The graphs will be displayed under **Plots** tab which is at the bottom right corner of the R Studio.

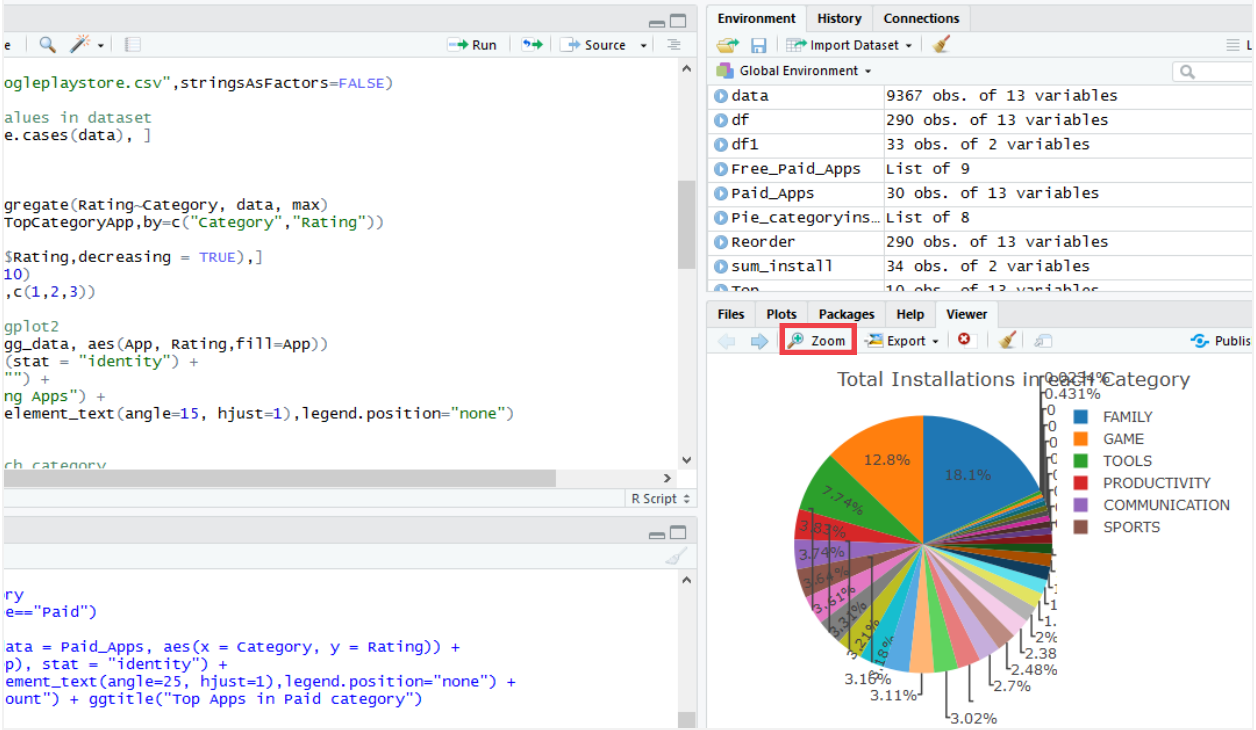


```
ogleplaystore.csv",stringsAsFactors=FALSE)

alues in dataset
e.cases(data), ]


gregate(Rating~Category, data, max)
TopCategoryApp,by=c("Category","Rating"))

$Rating,decreasing = TRUE),]
10)
,c(1,2,3))

gplot2
gg_data, aes(App, Rating,fill=App))
(stat = "identity") +
"") +
ng Apps") +
element_text(angle=15, hjust=1),legend.position="none")
```

```
ry
e=="Paid")

ata = Paid_Apps, aes(x = Category, y = Rating)) +
p), stat = "identity") +
ement_text(angle=25, hjust=1),legend.position="none") +
ount") + ggtitle("Top Apps in Paid category")
```

```
data                 9367 obs. of 13 variables
df                   290 obs. of 13 variables
df1                  33 obs. of 2 variables
Free_Paid_Apps       List of 9
Paid_Apps            30 obs. of 13 variables
Pie_categoryins…     List of 8
Reorder              290 obs. of 13 variables
sum_install          34 obs. of 2 variables
```

You can select **Zoom** option as shown below to view the graph in a separate window.

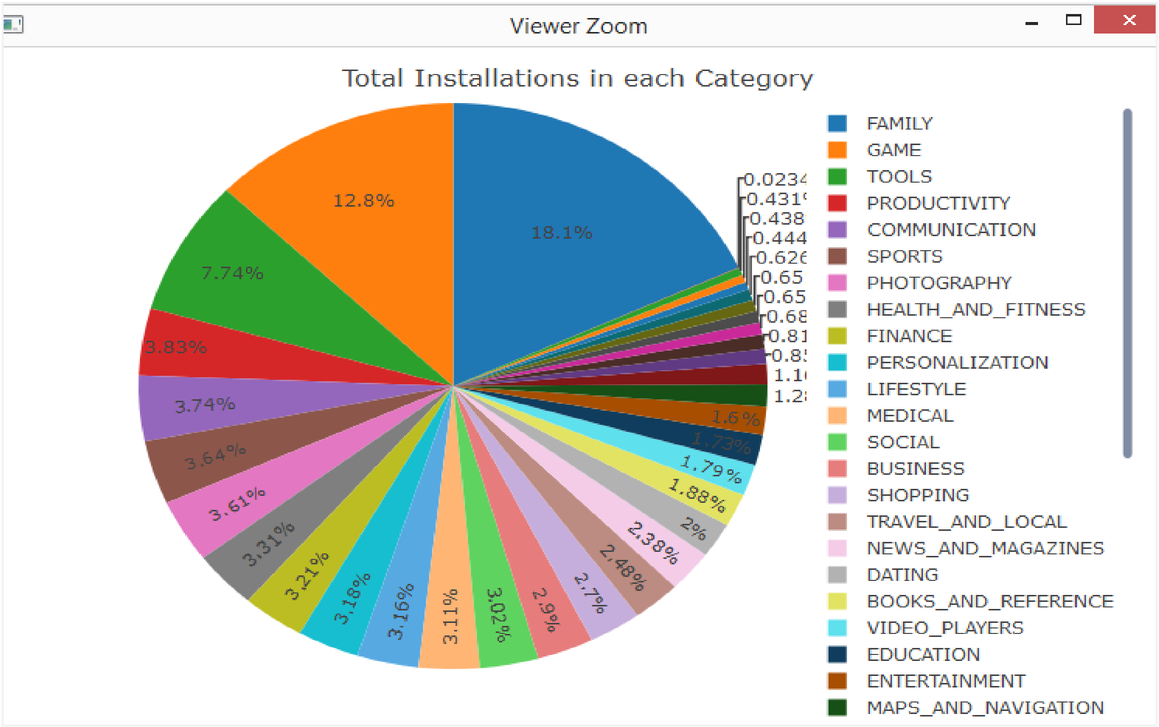


```
ogleplaystore.csv",stringsAsFactors=FALSE)

alues in dataset
e.cases(data), ]


gregate(Rating~Category, data, max)
TopCategoryApp,by=c("Category","Rating"))

$Rating,decreasing = TRUE),]
10)
,c(1,2,3))

gplot2
gg_data, aes(App, Rating,fill=App))
(stat = "identity") +
"") +
ng Apps") +
element_text(angle=15, hjust=1),legend.position="none")


ch category
```

```
ry
e=="Paid")

ata = Paid_Apps, aes(x = Category, y = Rating)) +
p), stat = "identity") +
ement_text(angle=25, hjust=1),legend.position="none") +
ount") + ggtitle("Top Apps in Paid category")
```

```
data               9367 obs. of 13 variables
df                 290 obs. of 13 variables
df1                33 obs. of 2 variables
Free_Paid_Apps     List of 9
Paid_Apps          30 obs. of 13 variables
Pie_categoryins…   List of 8
Reorder            290 obs. of 13 variables
sum_install        34 obs. of 2 variables
Top                10 obs. of 13 variables
```

Below is the Pie chart for Number of Installations in each Category.

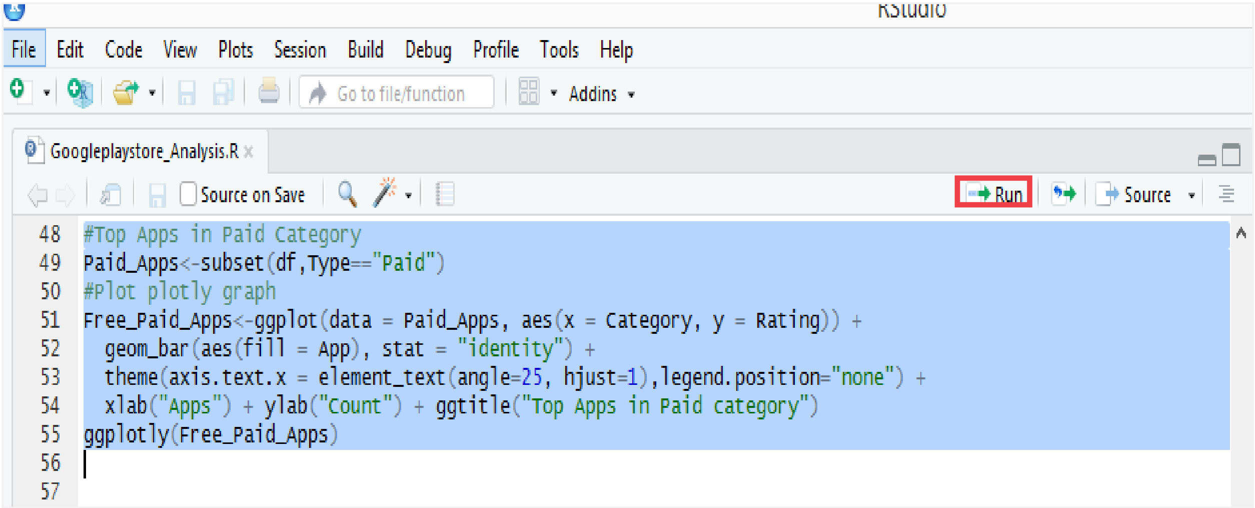


Step #11 | Find out Top Paid Apps

Analyze top paid apps.

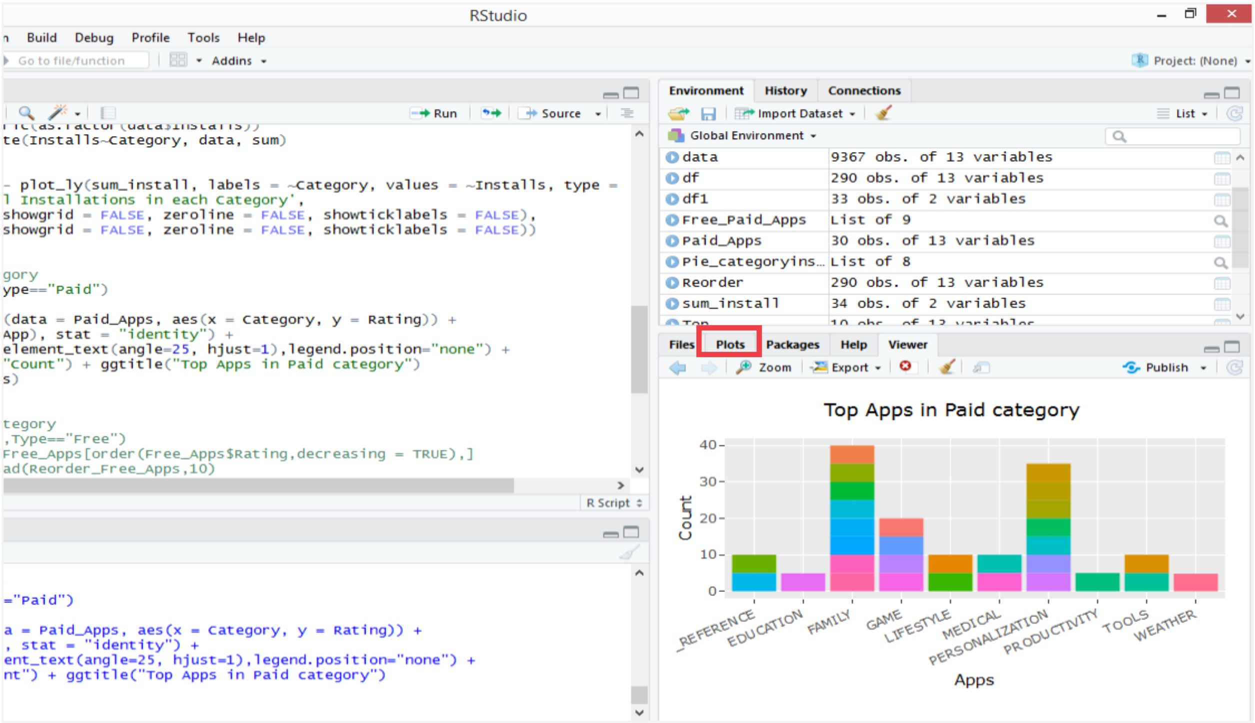
```
Paid_Apps<-subset(df,Type=="Paid")
TopPaid_Apps<-ggplot(data = Paid_Apps, aes(x = Category, y = Rating)) +
 geom_bar(aes(fill = App), stat = "identity") +
 theme(axis.text.x = element_text(angle=25, hjust=1),legend.position="none") +
 xlab("Apps") + ylab("Count") + ggtitle("Top Apps in Paid Category")
ggplotly(TopPaid_Apps)
```

To run commands in R script, select the commands you wish to execute and click on **run** button.



```
#Top Apps in Paid Category
Paid_Apps<-subset(df,Type=="Paid")
#Plot plotly graph
Free_Paid_Apps<-ggplot(data = Paid_Apps, aes(x = Category, y = Rating)) +
  geom_bar(aes(fill = App), stat = "identity") +
  theme(axis.text.x = element_text(angle=25, hjust=1),legend.position="none") +
  xlab("Apps") + ylab("Count") + ggtitle("Top Apps in Paid category")
ggplotly(Free_Paid_Apps)
```

The graphs will be displayed under **Plots** tab which is at the bottom right corner of the R Studio.

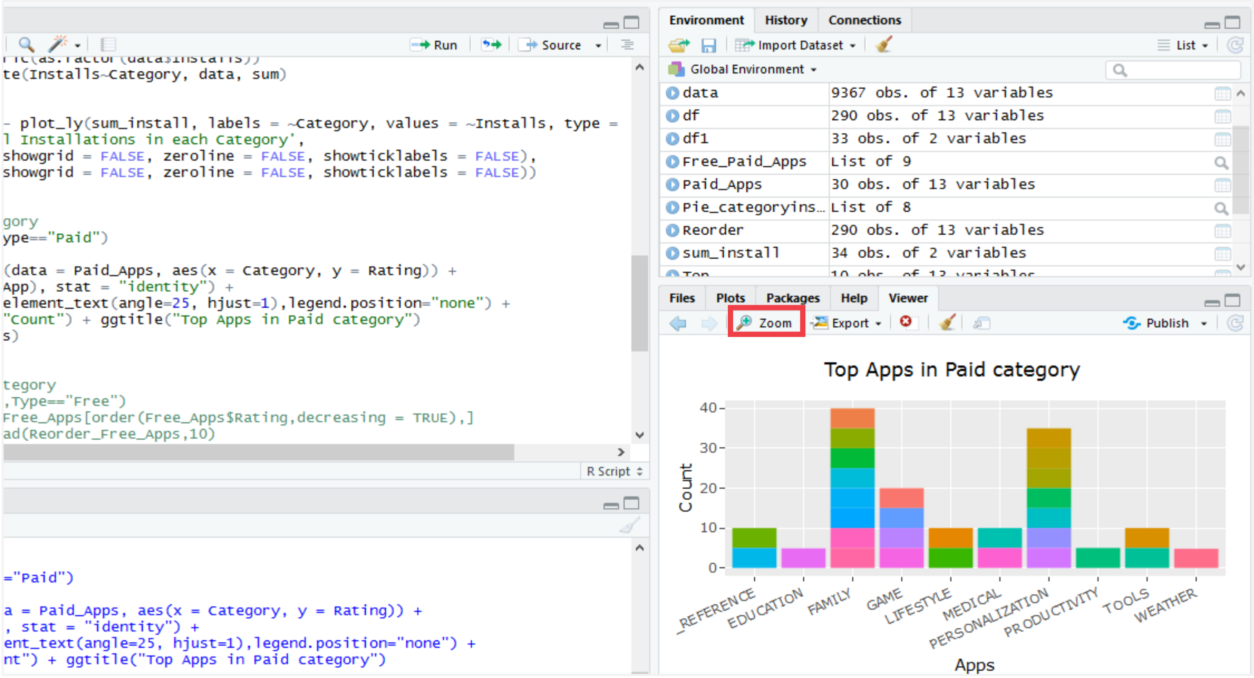


```
te(Installs~Category, data, sum)


- plot_ly(sum_install, labels = ~Category, values = ~Installs, type =
l Installations in each Category',
showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))


gory
ype=="Paid")

(data = Paid_Apps, aes(x = Category, y = Rating)) +
App), stat = "identity") +
element_text(angle=25, hjust=1),legend.position="none") +
"Count") + ggtitle("Top Apps in Paid category")
s)


tegory
,Type=="Free")
Free_Apps[order(Free_Apps$Rating,decreasing = TRUE),]
ad(Reorder_Free_Apps,10)
```

```
="Paid")

a = Paid_Apps, aes(x = Category, y = Rating)) +
, stat = "identity") +
ent_text(angle=25, hjust=1),legend.position="none") +
nt") + ggtitle("Top Apps in Paid category")
```

```
data              9367 obs. of 13 variables
df                290 obs. of 13 variables
df1               33 obs. of 2 variables
Free_Paid_Apps    List of 9
Paid_Apps         30 obs. of 13 variables
Pie_categoryins…  List of 8
Reorder           290 obs. of 13 variables
sum_install       34 obs. of 2 variables
Top               10 obs. of 13 variables
```

You can select **Zoom** option as shown below to view the graph in a separate window.

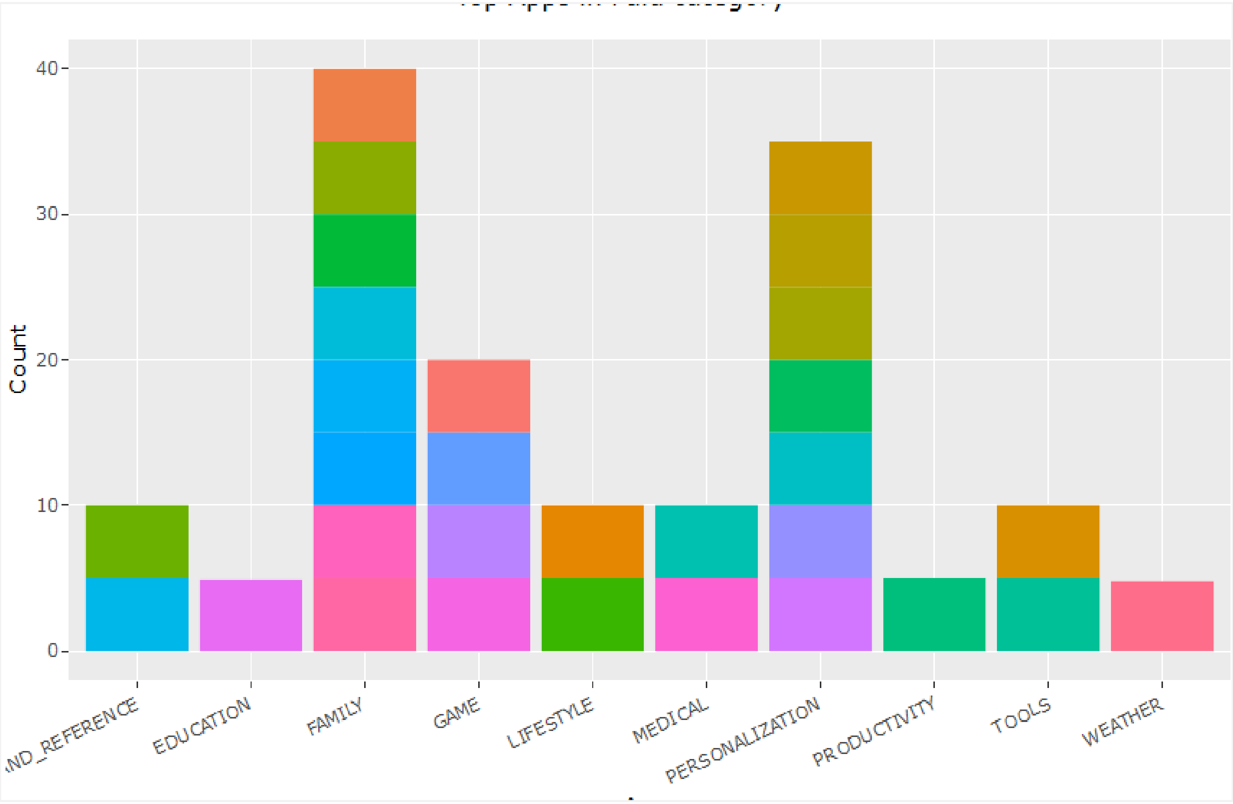


```
te(Installs~Category, data, sum)


- plot_ly(sum_install, labels = ~Category, values = ~Installs, type =
l Installations in each Category',
showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))


gory
ype=="Paid")

(data = Paid_Apps, aes(x = Category, y = Rating)) +
App), stat = "identity") +
element_text(angle=25, hjust=1),legend.position="none") +
"Count") + ggtitle("Top Apps in Paid category")
s)


tegory
,Type=="Free")
Free_Apps[order(Free_Apps$Rating,decreasing = TRUE),]
ad(Reorder_Free_Apps,10)
```

```
="Paid")

a = Paid_Apps, aes(x = Category, y = Rating)) +
, stat = "identity") +
ent_text(angle=25, hjust=1),legend.position="none") +
nt") + ggtitle("Top Apps in Paid category")
```

```
data              9367 obs. of 13 variables
df                290 obs. of 13 variables
df1               33 obs. of 2 variables
Free_Paid_Apps    List of 9
Paid_Apps         30 obs. of 13 variables
Pie_categoryins…  List of 8
Reorder           290 obs. of 13 variables
sum_install       34 obs. of 2 variables
Top               10 obs. of 13 variables
```

Below is the plotly graph for Top Apps in Paid Category.



Hurrah!! With this lab you were able to analyze and visualize the **Google Play Store Apps Data using R.**

For any questions regarding the lab please feel free to reach out to innoation@miraclesoft.com. **We hope you enjoyed analyzing data with us ☺**