

IDC 409

Introduction to Data Science

Lecture 1

Google classroom code : **nxqogyej**

- R and python are two programming languages used for performing data analytics. <https://web.archive.org/web/20181014111802/http://ect.bell-labs.com/sl/S/>
- R is a language and environment for statistical computing and graphics.
- It was developed by Ross Ihaka and Robert Gentleman in the 1990s.
- It has been derived from two other languages S and Scheme.
- It is much similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues.
- R can be considered as a different implementation of S.
- There are some important differences, but much code written for S runs unaltered under R.
- Being open source, R picked up and is now used a lot worldwide as a statistical programming language.
- R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible.
- R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code.
- It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS.



What Gemini say

Your Goal...

To become a Data Analyst, Business Analyst, or work in Marketing Analytics.

To become a Machine Learning Engineer or AI Specialist.

To work in Academia, Scientific Research, or Bioinformatics.

To become a generalist "Full-Stack" Data Scientist.

Recommendation

Learn R (especially the Tidyverse and ggplot2).

Learn Python first.

Learn R first.

Start with Python, then learn R.

Why?

These roles are heavy on exploration, data cleaning, and visualization to communicate insights. R excels at this.

This career path is all about building, training, and deploying models. Python's ML ecosystem is dominant.

R is the standard language in these fields. You'll need it to collaborate and use existing tools.

Start with Python to build a strong, versatile foundation that opens the most doors. Then, add R to your toolkit to handle specialized statistical tasks and create top-tier visualizations. Being proficient in both makes you an extremely valuable candidate.



ChatGPT

What ChatGPT say

TL;DR: Should You Learn R?

Goal

Academic/research/statistics

Data visualization (especially static plots)

General-purpose data science, ML

Jobs in finance, healthcare research

Web apps, AI, engineering roles

Learn R?

Yes

Worth it

Stick to Python

Can be valuable

Focus on Python/JS



R related books <https://www.r-project.org/doc/bib/R-books.html>

R Manual <https://cran.r-project.org/manuals.html>

Manuals are always the best way to learn about the functions and usages available in any package.
It is always tedious to go through but always worthwhile.

R is a programming language and software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing.

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

- an effective data handling and storage facility,
- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either on-screen or on hardcopy, and
- a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

<https://www.r-project.org/>



<https://posit.co/download/rstudio-desktop/>

RStudio is an integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management.

RStudio is available in open source and commercial editions and runs on the desktop (Windows, Mac, and Linux) or in a browser connected to RStudio Server or RStudio Server Pro (Debian/Ubuntu, RedHat/CentOS, and SUSE Linux).

RStudio is a free and open-source integrated development environment (IDE) for R, a programming language for statistical computing and graphics.

JJ Allaire, creator of the programming language ColdFusion, founded RStudio. Hadley Wickham is the Chief Scientist at RStudio.

RStudio is available in two editions: RStudio Desktop, where the program is run locally as a regular desktop application; and RStudio Server, which allows accessing RStudio using a web browser while it is running on a remote Linux server.

Prepackaged distributions of RStudio Desktop are available for Windows, OS X, and Linux.

RStudio is written in the C++ programming language and uses the Qt framework for its graphical user interface.

```
vish@LAPTOP-5QP9DUIB:~$ sudo apt install r-base r-base-dev
vish@LAPTOP-5QP9DUIB:~$ R
R version 4.1.2 (2021-11-01) -- "Bird Hippie"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (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.
```

The above command may install older version of R.

In case you want to install recent, use the following

One can also install R from CRAN repository

```
# Add CRAN repository
sudo apt install -y --no-install-recommends software-properties-common dirmngr
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 51716619E084DAB9
sudo add-apt-repository "deb https://cloud.r-project.org/bin/linux/ubuntu $(lsb_release -cs)-cran40/"
# Update package index
sudo apt update# Install R base
sudo apt install -y r-base
```

vish@LAPTOP-5QP9DUIB:~ X +

vish@LAPTOP-5QP9DUIB:~\$ R

R version 4.1.2 (2021-11-01) -- "Bird Hippie"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (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.

[Previously saved workspace restored]

```
> x <-10
> y <-30
> z<-X^2+Y^2
Error: object 'X' not found
> z<-x^2+y^2
> z
[1] 1000
>
```

R is case sensitive

```
vish@LAPTOP-5QP9DUIB: ~      + | v
vish@LAPTOP-5QP9DUIB:~$ emacs add.R &
[2] 4850
[1] Done
vish@LAPTOP-5QP9DUIB:~$ Rscript add.R
[1] 1000
vish@LAPTOP-5QP9DUIB:~$
```

```
emacs@LAPTOP-5QP9DUIB
File Edit Options Buffers Tools Help
Save Undo
x<-10
y<-30
z<-x^2+y^2
z
-
```

<https://posit.co/products/open-source/rstudio/>

RStudio is an integrated development environment (IDE) for R and Python.

- ❖ It includes a console, syntax-highlighting editor that supports direct code execution, and tools for plotting, history, debugging, and workspace management.
- ❖ RStudio is available in open source and commercial editions and runs on the desktop (Windows, Mac, and Linux).



myVariables.R x

```

1 ## To see what are the variables in R
2 x <- 21 # float
3 class(x)
4 thetext <- "IDC409 Introduction to DS"
5 theList <- c(1,2,3,4,5,6,7,8,9,10)
6
7
3:1 (Top Level) R Script

```

SOURCE

First Window

Console Terminal x Background Jobs x

R 4.3.1 · ~/

R version 4.3.1 (2023-06-16 ucrt) -- "Beagle Scouts"
Copyright (C) 2023 The R Foundation for Statistical Computing
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.

> |
CONSOLE

Second Window

Environment History Connections Tutorial

Import Dataset 89 MiB

R Global Environment

Environment/History Environment is empty

Third Window

Files	Plots	Packages	Help	Viewer	Presentation
Install	Update				
		Name	Description		Version
		System Library	Files/Plots/Packages		<i>Fourth Window</i>
<input checked="" type="checkbox"/>	base	The R Base Package			4.3.1
<input type="checkbox"/>	boot	Bootstrap Functions (Originally by Angelo Canty for S)			1.3-28.1
<input type="checkbox"/>	class	Functions for Classification			7.3-22
<input type="checkbox"/>	cluster	"Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al.			2.1.4
<input type="checkbox"/>	codetools	Code Analysis Tools for R			0.2-19
<input type="checkbox"/>	compiler	The R Compiler Package			4.3.1
<input checked="" type="checkbox"/>	datasets	The R Datasets Package			4.3.1
<input type="checkbox"/>	foreign	Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ...			0.8-84
<input checked="" type="checkbox"/>	graphics	The R Graphics Package			4.3.1
<input checked="" type="checkbox"/>	grDevices	The R Graphics Devices and Support for Colours and Fonts			4.3.1
<input type="checkbox"/>	grid	The Grid Graphics Device			4.3.1

- **First Window**

- is the source window where the R code is entered.
- New R Script window is opened by clicking on the menu *File->New File ->R Script*.
- Source code is typed in the script window.
- Code has to be executed by using ‘Run’ in the same window or by pressing “*Control + Enter*” on the keyboard.

- **Second Window**

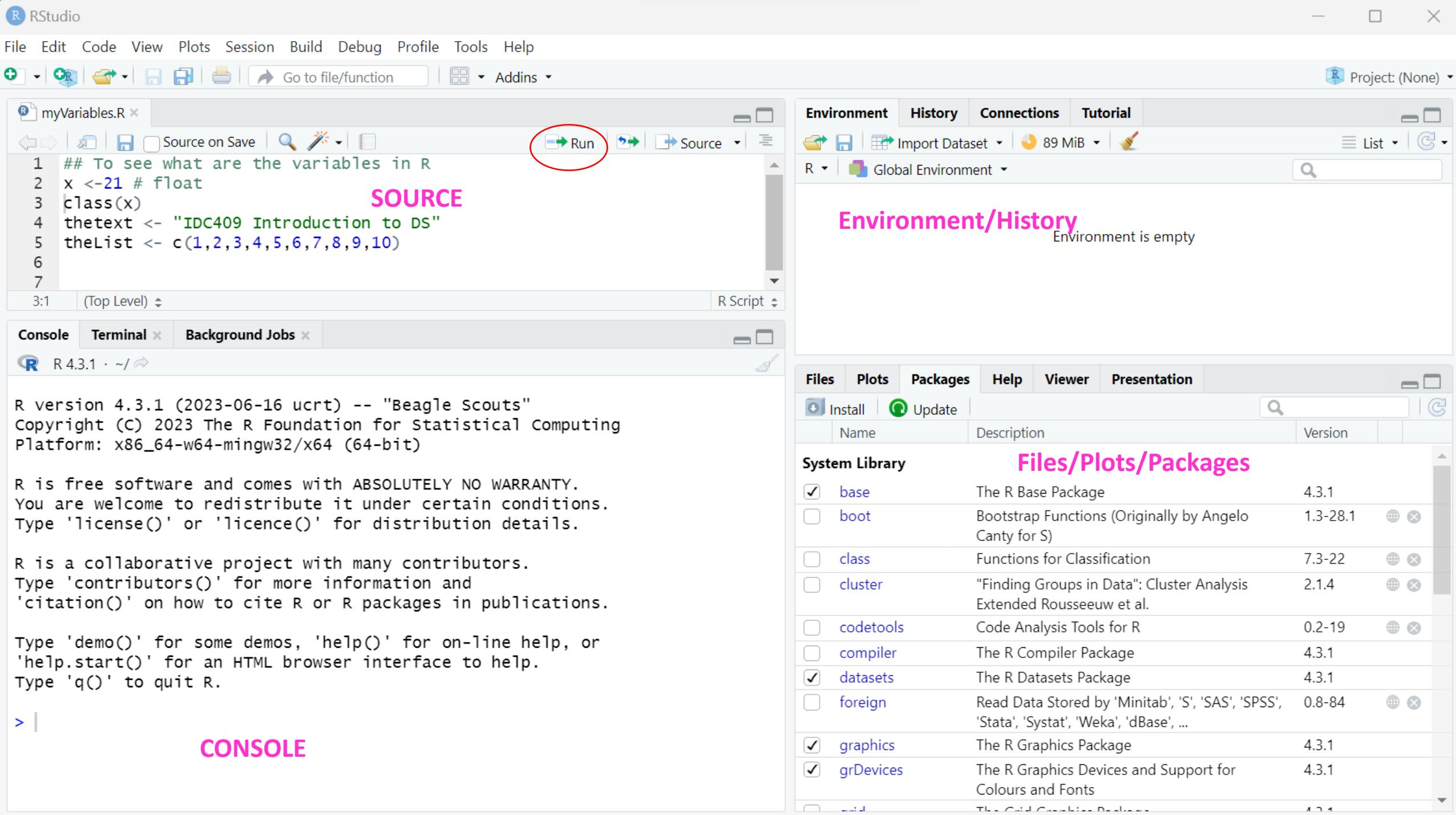
- is the console window.
- Codes typed in script window gets executed in the console window.
- Codes are executed in R prompt symbolized by ‘>’.

- **Third Window**

- is the environment/history window.
- Environment tab displays the names of all the variables or dataframes that are currently defined in the R session.

- **Fourth Window**

- shows the list of files and packages.
- File tab provides access to the file directory of the hard drive.
- One can navigate to the folder and set it as working directory by clicking ‘More’ and ‘Set as Working Directory’.



myVariables.R x

```
1 ## To see what are the variables in R
2 x <- 21 # float
3 class(x)
4 thetext <- "IDC409 Introduction to DS"
5 theList <- c(1,2,3,4,5,6,7,8,9,10)
6
7
```

6:1 (Top Level) R Script

Console Terminal x Background Jobs x

R 4.3.1 · ~/

```
R version 4.3.1 (2023-06-16 ucrt) -- "Beagle Scouts"
Copyright (C) 2023 The R Foundation for Statistical Computing
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.
```

```
> theList <- c(1,2,3,4,5,6,7,8,9,10)
>
```

Environment History Connections Tutorial

Import Dataset 45 MiB

R Global Environment

Values

theList	num [1:10] 1 2 3 4 5 6 7 8 9 10
---------	---------------------------------

Files Plots Packages Help Viewer Presentation

Install Update

Name	Description	Version
System Library		
<input checked="" type="checkbox"/> base	The R Base Package	4.3.1
<input type="checkbox"/> boot	Bootstrap Functions (Originally by Angelo Canty for S)	1.3-28.1
<input type="checkbox"/> class	Functions for Classification	7.3-22
<input type="checkbox"/> cluster	"Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al.	2.1.4
<input type="checkbox"/> codetools	Code Analysis Tools for R	0.2-19
<input type="checkbox"/> compiler	The R Compiler Package	4.3.1
<input checked="" type="checkbox"/> datasets	The R Datasets Package	4.3.1
<input type="checkbox"/> foreign	Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ...	0.8-84
<input checked="" type="checkbox"/> graphics	The R Graphics Package	4.3.1
<input checked="" type="checkbox"/> grDevices	The R Graphics Devices and Support for Colours and Fonts	4.3.1

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Project: (None)

myVariables.R

```
## To see what are the variables in R
x <- 21 # float
class(x)
thetext <- "IDC409 Introduction to DS"
theList <- c(1,2,3,4,5,6,7,8,9,10)
```

Source on Save Run Source

6:1 (Top Level) R Script

Console Terminal Background Jobs

R 4.3.1 · ~/

```
R version 4.3.1 (2023-06-16 ucrt) -- "Beagle Scouts"
Copyright (C) 2023 The R Foundation for Statistical Computing
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.
```

> theList <- c(1,2,3,4,5,6,7,8,9,10)
> theparticle <- list("apple","cat")
>

You can also write here and when you enter, variables goes into Global Environment

Environment History Connections Tutorial

Import Dataset 112 MiB

R Global Environment

Data theparticle List of 2
Values theList num [1:10] 1 2 3 4 5 6 7 8 9 10

Files Plots Packages Help Viewer Presentation

Install Update

Name	Description	Version
base	The R Base Package	4.3.1
boot	Bootstrap Functions (Originally by Angelo Canty for S)	1.3-28.1
class	Functions for Classification	7.3-22
cluster	"Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al.	2.1.4
codetools	Code Analysis Tools for R	0.2-19
compiler	The R Compiler Package	4.3.1
datasets	The R Datasets Package	4.3.1
foreign	Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ...	0.8-84
graphics	The R Graphics Package	4.3.1
grDevices	The R Graphics Devices and Support for Colours and Fonts	4.3.1

myVariables.R* ReadFile.R*

Source on Save Run Source

```
1 ## To see what are the variables in R
2 x <- 21 # float
3 class(x)
4 thetext <- "IDC409 Introduction to DS"
5 theList <- c(1,2,3,4,5,6,7,8,9,10)
6
7
8
```

Environment History Connections Tutorial

Import Dataset 120 MiB

R Global Environment

Data

theparticle List of 2

\$: chr "apple"
\$: chr "cat"

Values

theList num [1:10] 1 2 3 4 5 6 7 8 9 10

Files Plots Packages Help Viewer Presentation

6:1 (Top Level) R Script

Console Terminal Background Jobs

R 4.3.1 ~/

```
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.

>
> theList <- c(1,2,3,4,5,6,7,8,9,10)
> the particle <- list("apple", "cat")
Error: unexpected symbol in "the particle"
> theparticle <- list("apple", "cat")
>
```

RStudio

File Edit Code View Plots Session Build Debug

New File

New Project...

Open File... Ctrl+O

Open File in New Column...

Reopen with Encoding...

Recent Files

Open Project...

Open Project in New Session...

Recent Projects

Import Dataset

Save Ctrl+S

Save As...

Rename

Save with Encoding...

Save All Alt+Ctrl+S

Compile Report...

Print...

Close Ctrl+W

Close All Ctrl+Shift+W

R Script Ctrl+Shift+N

Quarto Document...

Quarto Presentation...

R Notebook

R Markdown...

Shiny Web App...

Plumber API...

C File

C++ File

Header File

Markdown File

HTML File

CSS File

JavaScript File

D3 Script

Python Script

Shell Script

SQL Script

Stan File

Text File

R Sweave

R HTML

R Documentation...

Project: (None)

Environment History Connections Tutorial

Import Dataset 89 MiB

R Global Environment

Environment is empty

Files Plots Packages Help Viewer Presentation

Install Update

Name	Description	Version
base	The R Base Package	4.3.1
boot	Bootstrap Functions (Originally by Angelo Carty for S)	1.3-28.1
class	Functions for Classification	7.3-22
cluster	"Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al.	2.1.4
codetools	Code Analysis Tools for R	0.2-19
compiler	The R Compiler Package	4.3.1
datasets	The R Datasets Package	4.3.1
foreign	Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ...	0.8-84
graphics	The R Graphics Package	4.3.1
grDevices	The R Graphics Devices and Support for Colours and Fonts	4.3.1
grid	The Grid Graphics Device	4.3.1

Comment Statements in R

- Comment statements are typed with # in the beginning.
- Any statement starting with # symbol will be ignored and will not executed.
- *# This is a comment #*

The screenshot shows the RStudio interface. In the top-left corner, there's a logo with a blue 'R' inside a circle. The main window has a title bar with tabs for 'myVariables.R' and 'Untitled1*'. Below the title bar is a toolbar with icons for file operations like Open, Save, and Print, along with a 'Go to file/function' search bar and an 'Addins' dropdown. The main workspace contains the following code:

```
1 # This is simply a comment statement #
```

In the bottom panel, there are three tabs: 'Console', 'Terminal', and 'Background Jobs'. The 'Console' tab is active, showing the R startup message and a command prompt:

```
R version 4.3.1 (2023-06-16 ucrt) -- "Beagle Scouts"  
Copyright (C) 2023 The R Foundation for Statistical Computing  
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.
```

Below the message, there are several blue command prompts:

```
> theList <- c(1,2,3,4,5,6,7,8,9,10)  
> theparticle <- list("apple","cat")  
>  
> # This is simply a comment statement #  
> |
```

Variables in R

- Variables in R are used to store
 - Integers
 - Complex numbers
 - Strings
 - Vectors
 - Matrices.
- Declaration of the variables is not need as R is a dynamically programmed language.
- Variable names can contain only alphabets, numbers, either dot or underscore.
- Variable names should start with an alphabet followed by alphanumeric characters.
- `x <- 2025`
- `course <- "IDC 409"`
- `newlists <- c(1,2,3,4)`
- `ClassRoom <- "LH1"`
- `Registered_Students <- 50`
- `var.r <- 1232`

Data Types in R

Numeric

Numeric data type stores decimal values to variables.
as.numeric() function converts the values into numeric data type

```
> x<-1234.56
> class(x)
[1] "numeric"
> y="345"
> class(y)
[1] "character"
> z<-as.numeric(y)
> class(z)
[1] "numeric"
> print(z)
[1] 345
>
```

Integer

Integers are the data types which assigns numbers without decimal digits.
Integer variables can also be created by assigning alphabet 'L' to the number
as.integer() function can be used to assign integers to variables.

```
> x<-1234
> class(x)
[1] "numeric"
> x<-1234L
> class(x)
[1] "integer"
> y=12441.4212
> y
[1] 12441.42
> z=as.integer(y)
> z
[1] 12441
> class(z)
[1] "integer"
```

Data Types in R

Complex

Complex data types contain two numeric values with an imaginary number ‘i’

```
> x<-10+5i  
> x  
[1] 10+5i  
> class(x)  
[1] "complex"  
>
```

Character

Character data type is used to represent strings.
Strings should be written within single or double quotes.
as.character() function to convert objects into character values.

```
> x<-"apple"  
> class(x)  
[1] "character"  
> a=12131.45  
> class(a)  
[1] "numeric"  
> z=as.character(a)  
> class(z)  
[1] "character"  
> z  
[1] "12131.45"
```

Logical

Logical data type stores two values, TRUE or FALSE.

```
> x<-"TRUE"  
> class(x)  
[1] "character"  
> X<-TRUE  
> class(X)  
[1] "logical"  
> Y<-FALSE  
> class(Y)  
[1] "logical"  
>
```

Operators in R

<- Assignment rightward

```
> x<-23  
> x  
[1] 23
```

<<- Assignment rightward

```
> x<<-452  
> x  
[1] 452
```

= Assignment

```
> zd=34  
> zd  
[1] 34
```

-> Assignment leftward

```
> 20->x  
> x  
[1] 20  
> 3141->>x  
> x  
[1] 3141
```

->> Assignment leftward

Assignment operators assign value to a variable

<- Regular Assignment

- ✓ Assigns a value **in the current environment**.
- ✓ This is the most commonly used assignment operator in R.

<<- Super Assignment

- ✓ Assigns a value **in the parent environment**.
- ✓ If the variable already exists in parent environment, updates it there
- ✓ If doesn't exist, create in global environment.

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

myVariables.R* ReadFile.R* Untitled1*

Run Source

```
1
2
3 y <- 1
4 print(y)
5
6 foo <- function() {
7   y <- 2      # local assignment
8   #y <<- 3    # super assignment to parent/global
9   print(y)    # prints local
10 }
11
12 foo()
13
14 print(y)
15
16
```

```
1
2
3 y <- 1
4 print(y)
5
6 foo <- function() {
7   y <- 2      # local assignment
8   y <<- 3    # super assignment to parent/global
9   print(y)    # prints local
10 }
11
12 foo()
13
14 print(y)
15
16
```

15:1 (Top Level) ▾

Console Terminal Background Jobs

R 4.3.1 . ~/

```
> source("~/active-rstudio-document", echo = TRUE)

> y <- 1

> print(y)
[1] 1

> foo <- function() {
+   y <- 2      # local assignment
+   #y <<- 3    # super assignment to parent/global
+   print(y)    # prints local
+   .... [TRUNCATED]

> foo()
[1] 2

> print(y)
[1] 1
>
```

8:3 foo() ▾

Console Terminal Background Jobs

R 4.3.1 . ~/

```
> source("~/active-rstudio-document", echo = TRUE)

> y <- 1

> print(y)
[1] 1

> foo <- function() {
+   y <- 2      # local assignment
+   y <<- 3    # super assignment to parent/global
+   print(y)    # prints local
+ }

> foo()
[1] 2

> print(y)
[1] 3
>
```

Arithmetic Operators in R

+	Addition
---	----------

-	Subtraction
---	-------------

*	Multiplication
---	----------------

/	Division
---	----------

^	Exponent
---	----------

%%	Modulus operator, Outputs remainder value
----	---

%/	Integer Division
----	------------------

```
> a<-400.34 ; b<-20
> a+b
[1] 420.34
> a-b
[1] 380.34
> a*b
[1] 8006.8
> a/b
[1] 20.017
> a^b
[1] 1.118355e+52
> a%%b
[1] 0.34
> a%/%b
[1] 20
```

Operators in R

< Less than

> Greater than

<= Less than or equal to

>= Greater than or equal to

== Equal to

!= Not equal to

```
> x<-31.3  
> y<-41.42  
> x<y  
[1] TRUE  
> x<=y  
[1] TRUE  
> x>y  
[1] FALSE  
> x>=y  
[1] FALSE  
> x==y  
[1] FALSE  
> x!=y  
[1] TRUE
```

Logical Operator	Description
!	Logical NOT
&	Element-wise logical AND
&&	Logical AND (Compares the first element in vector)
	Element-wise logical OR
	Logical OR (Compares the first element in vector)

The screenshot shows the RStudio interface with two files open: `myVariables.R*` and `ReadFile.R`. The `myVariables.R*` file contains the following R code:

```
1 a <- c(TRUE, FALSE, TRUE)
2 b <- c(TRUE, TRUE, FALSE)
3
4 d <- c(FALSE, FALSE, TRUE)
5
6 a[1] && d[1]
7
8 a&d
9
10 a&b
11 a[1] && b[1]
12
```

The R console window shows the execution of the code from `myVariables.R*`. The output is as follows:

```
11:8 (Top Level) ◊
Console Terminal × Background Jobs ×
R 4.3.1 . ~/ ◊
> source("~/active-rstudio-document", echo = TRUE)
> a <- c(TRUE, FALSE, TRUE)
> b <- c(TRUE, TRUE, FALSE)
> d <- c(FALSE, FALSE, TRUE)
> a[1] && d[1]
[1] FALSE
> a&d
[1] FALSE FALSE TRUE
> a&b
[1] TRUE FALSE FALSE
> a[1] && b[1]
[1] TRUE
>
```

The screenshot shows the RStudio interface with two files open: `myVariables.R*` and `ReadFile.R`. The `myVariables.R*` file contains the following R code:

```
1 a <- c(TRUE, FALSE, TRUE)
2 b <- c(TRUE, TRUE, FALSE)
3
4 d <- c(FALSE, FALSE, TRUE)
5
6 a[1] || d[1]
7
8 a|b
9
10 a|b
11 a[1] || b[1]
12
```

The R console window shows the execution of the code from `myVariables.R*`. The output is as follows:

```
8:3 (Top Level) ◊
Console Terminal × Background Jobs ×
R 4.3.1 . ~/ ◊
> source("~/active-rstudio-document", echo = TRUE)
> a <- c(TRUE, FALSE, TRUE)
> b <- c(TRUE, TRUE, FALSE)
> d <- c(FALSE, FALSE, TRUE)
> a[1] || d[1]
[1] TRUE
> a|d
[1] TRUE FALSE TRUE
> a|b
[1] TRUE TRUE TRUE
> a[1] || b[1]
[1] TRUE
>
```

Functions in R

Mathematical	Description
abs()	Absolute value
log(x,base=y)	Logarithm of x with base y; if base not specified, returns the natural logarithm
exp(x)	Exponential value
sqrt(x)	Square root of the data
factorial(x)	Factorial of data (x!)

Statistical	Description
mean(x)	Mean of x
median(x)	Median of x
var(x)	Variance of x
sd(x)	Standard deviation of x
quantile(x)	Divides the dataset into quartiles

General Purpose	Description
range(x)	Minimum and Maximum Values
sort(x)	Sort the data in ascending order
order(x)	Returns the order of the values in the dataset
length()	Length of the dataset
sign()	Sign of the data
tolower()	Converts alphabets to lowercase
toupper()	Converts alphabets to uppercase
ls()	Clears the console screen

User defined functions in R

One can define or write their own functions as per the need

```
Add_Function_Name <- function (variable1, variable2) {  
  Value <- variable1 + variable 2  
  return(Value)  
}
```

```
> var1<-109  
> var2<-341  
> myownfunction <- function(var1, var2){  
+ newvar<- var1* var2  
+ return(newvar)  
+ }  
> var3=myownfunction(var1,var2)  
> var3  
[1] 37169
```



A screenshot of the RStudio interface showing an R session. The session window displays the following code and its output:

```
R > R 4.3.1 > ~/  
> source("~/active-rstudio-document", echo = TRUE)  
  
> greet <- function(name = "Guest") {  
+   message <- paste("Namaste,", name)  
+   return(message)  
+ }  
  
> greet()  
[1] "Namaste, Guest"  
  
> greet("Captain vyom")  
[1] "Namaste, Captain vyom"
```

Vectors in R

Vector is a sequence of elements.

All the elements in the vector can be of same or different data type.

Atomic vector is one that has all elements of same data type.

Vector with elements of different data type is called a list.

The data type of the elements can be logical, integer, double, character or complex

Vector Creation	Description
c() function	Combines the elements and returns a vector in the form of a one-dimensional array.
':' operator	':' operator assigns numbers within a given range
seq() function	Assigns numbers in given range with an increment

```
> vec[1]; vec[2]; vec[3]; vec[4]; vec[5]
[1] 10
[1] 213
[1] 312
[1] 341
[1] 234
```

```
> vec<-c(10,213,312,341,234,234,123,434)
> vec
[1] 10 213 312 341 234 234 123 434
> vec2<-10:30
> vec2
[1] -10  -9  -8  -7  -6  -5  -4  -3  -2  -1   0   1   2   3   4   5   6   7   8
[20]    9   10   11   12   13   14   15   16   17   18   19   20   21   22   23   24   25   26   27
[39]   28   29   30
> vec3<-seq(-10,30, by=5)
> vec3
[1] -10  -5   0   5  10  15  20  25  30
```

```
> vec3[1:4]
[1] -10  -5   0   5
```

```
> vec3[1:4]
[1] -10 -5  0  5
> vec3[-1]
[1] -5  0  5 10 15 20 25 30
> vec3[-2]
[1] -10  0  5 10 15 20 25 30
> vec3[-3]
[1] -10 -5  5 10 15 20 25 30
> vec3[-4]
[1] -10 -5  0 10 15 20 25 30
> vec3[-5]
[1] -10 -5  0  5 15 20 25 30
> vec3[-7]
[1] -10 -5  0  5 10 15 25 30
```

```
> a<-c(2,4,6,8,10,12,14)
> b<-c(1,3,5,9,11,13)
> c= a+b
Warning message:
In a + b : longer object length is not a multiple of shorter object length
> b<-c(1,3,5,9,11,13,15)
> c= a+b
> c
[1]  3  7 11 17 21 25 29
> f<-c(10,20,30,40,50,60,70,80,90)
> f+13.3
[1]  23.3  33.3  43.3  53.3  63.3  73.3  83.3  93.3 103.3
> f
[1] 10 20 30 40 50 60 70 80 90
> g<-f+13.3
> g
[1]  23.3  33.3  43.3  53.3  63.3  73.3  83.3  93.3 103.3
```

```
> vec3
[1] -10 -5  0  5 10 15 20 25 30
> vec3[1]<-1989
> vec3
[1] 1989 -5  0  5 10 15 20 25 30
> vec3
[1] 1989 -5  0  5 10 15 20 25 30
> vec3[vec3>20]<-1999
> vec3
[1] 1999 -5  0  5 10 15 20 1999 1999
> vec3
[1] 1999 -5  0  5 10 15 20 1999 1999
> vec3<-vec3[7:9]
> vec3
[1] 20 1999 1999
```

```
> test<- c(1, 2, 3, 4, 5, c(6, 7, 8, 9))
> test
[1] 1 2 3 4 5 6 7 8 9
> class(test)
[1] "numeric"
```

```
> d<-NULL
> d
NULL
> class(d)
[1] "NULL"
```

The List

A **list** contains elements of different data types such as numbers, strings, characters or vectors within it.
The list is created using an inbuilt function called `list()`.

Creating a list

```
> thislist<-list("Apple", "Cat", "Rat", "Dog", c(192,123,1231,124,12), c(FALSE, TRUE, FALSE), 123.8, 123.1, NULL)
```

```
> thislist
[[1]]
[1] "Apple"

[[2]]
[1] "Cat"

[[3]]
[1] "Rat"

[[4]]
[1] "Dog"

[[5]]
[1] 192 123 1231 124 12

[[6]]
[1] FALSE TRUE FALSE

[[7]]
[1] 123.8

[[8]]
[1] 123.1

[[9]]
NULL
```

```
> thislist[0]; thislist[1]; thislist[2]; thislist[3]; thislist[4];
list()
[[1]]
[1] "Apple"

[[1]]
[1] "Cat"

[[1]]
[1] "Rat"

[[1]]
[1] "Dog"
```

```
> thislist[5]; thislist[6];thislist[7];thislist[8]; thislist[9]
[[1]]
[1] 192 123 1231 124 12

[[1]]
[1] FALSE TRUE FALSE

[[1]]
[1] 123.8

[[1]]
[1] 123.1

[[1]]
NULL
```

Data frames in R

- Dataframe is an array-like object represented in a tabular format.
- Each column in the data frame represents a variable and each row represents one set of values of the variable or record.
- The number of rows gives the number of observations in a data frame.
- The column names cannot be empty.
- Dataframe can be expanded by adding columns and rows.
- R also provides options for slicing and dicing the data frame by rows and columns

```
> students_df<-data.frame(  
+ id = c(0,1,2,3,4,5,6,7,8,9,10),  
+ name= c("A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K"),  
+ marks=c(40,41,42,43,44,45,46,47,48,49,50),  
+ class=c("1y", "1y", "1y", "1y", "1y", "1y", "1y", "1y", "1y", "1y"),  
+ package=c(100000, 100000, 100000, 100000, 100000, 100000, 100000, 100000, 100000, 100000)  
+ )  
> students_df  
   id name marks class package  
1   0    A    40    1y  1e+05  
2   1    B    41    1y  1e+05  
3   2    C    42    1y  1e+05  
4   3    D    43    1y  1e+05  
5   4    E    44    1y  1e+05  
6   5    F    45    1y  1e+05  
7   6    G    46    1y  1e+05  
8   7    H    47    1y  1e+05  
9   8    I    48    1y  1e+05  
10  9    J    49    1y  1e+05  
11 10    K    50    1y  1e+05
```

Accessing data from Dataframe

```
> students_df
  id name marks class package
1  0   A    40    1y  1e+05
2  1   B    41    1y  1e+05
3  2   C    42    1y  1e+05
4  3   D    43    1y  1e+05
5  4   E    44    1y  1e+05
6  5   F    45    1y  1e+05
7  6   G    46    1y  1e+05
8  7   H    47    1y  1e+05
9  8   I    48    1y  1e+05
10 9   J    49    1y  1e+05
11 10  K    50    1y  1e+05
```

```
> students_df$name
[1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K"
> students_df[["name"]]
[1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K"
> students_df[[2]]
[1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K"
[2] "B"
```

```

> names(students_df)[ names(students_df) == "marks" ] <- "grades"
> students_df
   id name grades class package
1   0   A     40    1y  1e+05
2   1   B     41    1y  1e+05
3   2   C     42    1y  1e+05
4   3   D     43    1y  1e+05
5   4   E     44    1y  1e+05
6   5   F     45    1y  1e+05
7   6   G     46    1y  1e+05
8   7   H     47    1y  1e+05
9   8   I     48    1y  1e+05
10  9   J     49    1y  1e+05
11 10   K     50    1y  1e+05

```

Modify the column name
“marks” to “grades”

Add new column age to the table

```

> #Adding a new column age
> students_df$age<-c(23,24,24,23,25,26,25,23,24,24,23)

```

	id	name	grades	class	package	age
1	0	A	40	1y	1e+05	23
2	1	B	41	1y	1e+05	24
3	2	C	42	1y	1e+05	24
4	3	D	43	1y	1e+05	23
5	4	E	44	1y	1e+05	25
6	5	F	45	1y	1e+05	26
7	6	G	46	1y	1e+05	25
8	7	H	47	1y	1e+05	23
9	8	I	48	1y	1e+05	24
10	9	J	49	1y	1e+05	24
11	10	K	50	1y	1e+05	23

Add new row to the table

```
> nstudents_df<- data.frame( id = c(11:12),name =c("X", "Y"), grades=c(51,52), class=c("1y", "1y"), package=c(150000,150000), age=c(23,25) )
```

```
> students_df
  id name grades class package age
1  0   A     40    1y  1e+05  23
2  1   B     41    1y  1e+05  24
3  2   C     42    1y  1e+05  24
4  3   D     43    1y  1e+05  23
5  4   E     44    1y  1e+05  25
6  5   F     45    1y  1e+05  26
7  6   G     46    1y  1e+05  25
8  7   H     47    1y  1e+05  23
9  8   I     48    1y  1e+05  24
10 9   J     49    1y  1e+05  24
11 10  K     50    1y  1e+05  23
```

```
> nstudents_df
  id name grades class package age
1 11   X     51    1y  150000  23
2 12   Y     52    1y  150000  25
```

```
> Newstudents_df<-rbind(students_df, nstudents_df)
> Newstudents_df
```

```
  id name grades class package age
1  0   A     40    1y  100000  23
2  1   B     41    1y  100000  24
3  2   C     42    1y  100000  24
4  3   D     43    1y  100000  23
5  4   E     44    1y  100000  25
6  5   F     45    1y  100000  26
7  6   G     46    1y  100000  25
8  7   H     47    1y  100000  23
9  8   I     48    1y  100000  24
10 9   J     49    1y  100000  24
11 10  K     50    1y  100000  23
12 11  X     51    1y  150000  23
13 12  Y     52    1y  150000  25
```

```
> # Extracting multiple columns from data frame  
> Newdf <- data.frame(Newstudents_df$name, Newstudents_df$grades, Newstudents_df$age)  
> Newdf
```

	Newstudents_df.name	Newstudents_df.grades	Newstudents_df.age
1	A	40	23
2	B	41	24
3	C	42	24
4	D	43	23
5	E	44	25
6	F	45	26
7	G	46	25
8	H	47	23
9	I	48	24
10	J	49	24
11	K	50	23
12	X	51	23
13	Y	52	25

```
> Newstudents_df<-rbind(students_df, nstudents_df)
> Newstudents_df
  id name grades class package age
1  0    A      40   1y  100000  23
2  1    B      41   1y  100000  24
3  2    C      42   1y  100000  24
4  3    D      43   1y  100000  23
5  4    E      44   1y  100000  25
6  5    F      45   1y  100000  26
7  6    G      46   1y  100000  25
8  7    H      47   1y  100000  23
9  8    I      48   1y  100000  24
10 9    J      49   1y  100000  24
11 10   K      50   1y  100000  23
12 11   X      51   1y  150000  23
13 12   Y      52   1y  150000  25
```

Select Rows

Rowdf <-Newstudents_df[4:9,]

```
> Rowdf
  id name grades class package age
4  3    D      43   1y  1e+05  23
5  4    E      44   1y  1e+05  25
6  5    F      45   1y  1e+05  26
7  6    G      46   1y  1e+05  25
8  7    H      47   1y  1e+05  23
9  8    I      48   1y  1e+05  24
```

Select column

```
> Columdf <- Newstudents_df[,2:3]
> Columdf
  name  grades
1   A     40
2   B     41
3   C     42
4   D     43
5   E     44
6   F     45
7   G     46
8   H     47
9   I     48
10  J     49
11  K     50
12  X     51
13  Y     52
```

Select only a particular row and/or column

```
> RCdf <- Newstudents_df[c(6,9), c(2,4) ]
> RCdf
  name  class
6   F    1y
9   I    1y
```

```
> Newstudents_df<-rbind(students_df, nstudents_df)
> Newstudents_df
  id name  grades class package age
1  0   A     40    1y 100000  23
2  1   B     41    1y 100000  24
3  2   C     42    1y 100000  24
4  3   D     43    1y 100000  23
5  4   E     44    1y 100000  25
6  5   F     45    1y 100000  26
7  6   G     46    1y 100000  25
8  7   H     47    1y 100000  23
9  8   I     48    1y 100000  24
10 9   J     49    1y 100000  24
11 10  K     50    1y 100000  23
12 11  X     51    1y 150000  23
13 12  Y     52    1y 150000  25
```

```
> Removedf = subset(Newstudents_df, select = -c(name, class))
> Removedf
  id grades package age
1  0     40  100000  23
2  1     41  100000  24
3  2     42  100000  24
4  3     43  100000  23
5  4     44  100000  25
6  5     45  100000  26
7  6     46  100000  25
8  7     47  100000  23
9  8     48  100000  24
10 9     49  100000  24
11 10    50  100000  23
12 11    51  150000  23
13 12    52  150000  25
```

```
> Newstudents_df<-rbind(students_df, nstudents_df)
> Newstudents_df
  id name grades class package age
1  0   A     40   1y  100000  23
2  1   B     41   1y  100000  24
3  2   C     42   1y  100000  24
4  3   D     43   1y  100000  23
5  4   E     44   1y  100000  25
6  5   F     45   1y  100000  26
7  6   G     46   1y  100000  25
8  7   H     47   1y  100000  23
9  8   I     48   1y  100000  24
10 9   J     49   1y  100000  24
11 10  K     50   1y  100000  23
12 11  X     51   1y  150000  23
13 12  Y     52   1y  150000  25
```

```
> Removedf = Newstudents_df[-c(1,2) ]
> Removedf
  grades class package age
1     40   1y  100000  23
2     41   1y  100000  24
3     42   1y  100000  24
4     43   1y  100000  23
5     44   1y  100000  25
6     45   1y  100000  26
7     46   1y  100000  25
8     47   1y  100000  23
9     48   1y  100000  24
10    49   1y  100000  24
11    50   1y  100000  23
12    51   1y  150000  23
13    52   1y  150000  25
```

```
> Removedf = Newstudents_df[ !( Newstudents_df$name=="A" & Newstudents_df$package==100000), ]  
> Removedf
```

	id	name	grades	class	package	age
2	1	B	41	1y	100000	24
3	2	C	42	1y	100000	24
4	3	D	43	1y	100000	23
5	4	E	44	1y	100000	25
6	5	F	45	1y	100000	26
7	6	G	46	1y	100000	25
8	7	H	47	1y	100000	23
9	8	I	48	1y	100000	24
10	9	J	49	1y	100000	24
11	10	K	50	1y	100000	23
12	11	X	51	1y	150000	23
13	12	Y	52	1y	150000	25

```
>  
> Newstudents_df<-rbind(students_df, nstudents_df)  
> Newstudents_df
```

	id	name	grades	class	package	age
1	0	A	40	1y	100000	23
2	1	B	41	1y	100000	24
3	2	C	42	1y	100000	24
4	3	D	43	1y	100000	23
5	4	E	44	1y	100000	25
6	5	F	45	1y	100000	26
7	6	G	46	1y	100000	25
8	7	H	47	1y	100000	23
9	8	I	48	1y	100000	24
10	9	J	49	1y	100000	24
11	10	K	50	1y	100000	23
12	11	X	51	1y	150000	23
13	12	Y	52	1y	150000	25

See the
comma here

```
> Removedf = Newstudents_df[ -c(4,9), ]  
> Removedf
```

	id	name	grades	class	package	age
1	0	A	40	1y	100000	23
2	1	B	41	1y	100000	24
3	2	C	42	1y	100000	24
5	4	E	44	1y	100000	25
6	5	F	45	1y	100000	26
7	6	G	46	1y	100000	25
8	7	H	47	1y	100000	23
10	9	J	49	1y	100000	24
11	10	K	50	1y	100000	23
12	11	X	51	1y	150000	23
13	12	Y	52	1y	150000	25

Ascending or Descending order

```
> Newstudents_df[order(Newstudents_df$age),]
```

	id	name	grades	class	package	age
1	0	A	40	1y	100000	23
4	3	D	43	1y	100000	23
8	7	H	47	1y	100000	23
11	10	K	50	1y	100000	23
12	11	X	51	1y	150000	23
2	1	B	41	1y	100000	24
3	2	C	42	1y	100000	24
9	8	I	48	1y	100000	24
10	9	J	49	1y	100000	24
5	4	E	44	1y	100000	25
7	6	G	46	1y	100000	25
13	12	Y	52	1y	150000	25
6	5	F	45	1y	100000	26

```
> Newstudents_df[order(-Newstudents_df$age),]
```

	id	name	grades	class	package	age
6	5	F	45	1y	100000	26
5	4	E	44	1y	100000	25
7	6	G	46	1y	100000	25
13	12	Y	52	1y	150000	25
2	1	B	41	1y	100000	24
3	2	C	42	1y	100000	24
9	8	I	48	1y	100000	24
10	9	J	49	1y	100000	24
1	0	A	40	1y	100000	23
4	3	D	43	1y	100000	23
8	7	H	47	1y	100000	23
11	10	K	50	1y	100000	23
12	11	X	51	1y	150000	23

```
1 # This is simply a comment statement #
2
3 mdata=read.csv("Rstudio/Lecture2025/customerDataset.csv")
4 dim(mdata)
5 str(mdata)
6 summary(mdata)
7
8
```

Run Source

Environment History Connections Tutorial

Import Dataset 110 MB

R Global Environment

mdata 5000 obs. of 14 variables

	phoneno	age	gender	zipcode	calls	sms	mms	charges	coverage	complaint	sim	phone	prepost	Airtel
\$	int	int	chr	int	int	int	int	int	int	int	chr	chr	chr	chr
	5974 4535 4016 8523 5052 8970 9042 5724 9238 4436 ...	1 1 1 2 2 2 3 2 3 ...	"Male" "Male" "Male" ...	91107 90089 94720 94112 91330 92121 91711 93943 90089 93023 ...	160 150 100 270 100 40 150 30 60 890 ...	25 45 39 35 35 37 53 50 35 34 ...	1 19 15 9 8 13 27 24 10 9 ...	490 340 110 1000 450 290 720 220 810 1800 ...	0 0 0 0 0 2 0 0 2 0 ...	4 3 1 1 4 4 2 1 3 1 ...	"Dual sim" "dual sim" "single sim" "single sim" ...	"Android" "Android" "Android" "Android" ...	"Prepaid" "Prepaid" "Prepaid" "Prepaid" ...	"No Airtel" "No Airtel" "No Airtel" "No Airtel" ...

Files Plots Packages Help Viewer Presentation

Zoom Export

7:1 (Top Level) R Script

Console Terminal Background Jobs

R 4.3.1 . ~/

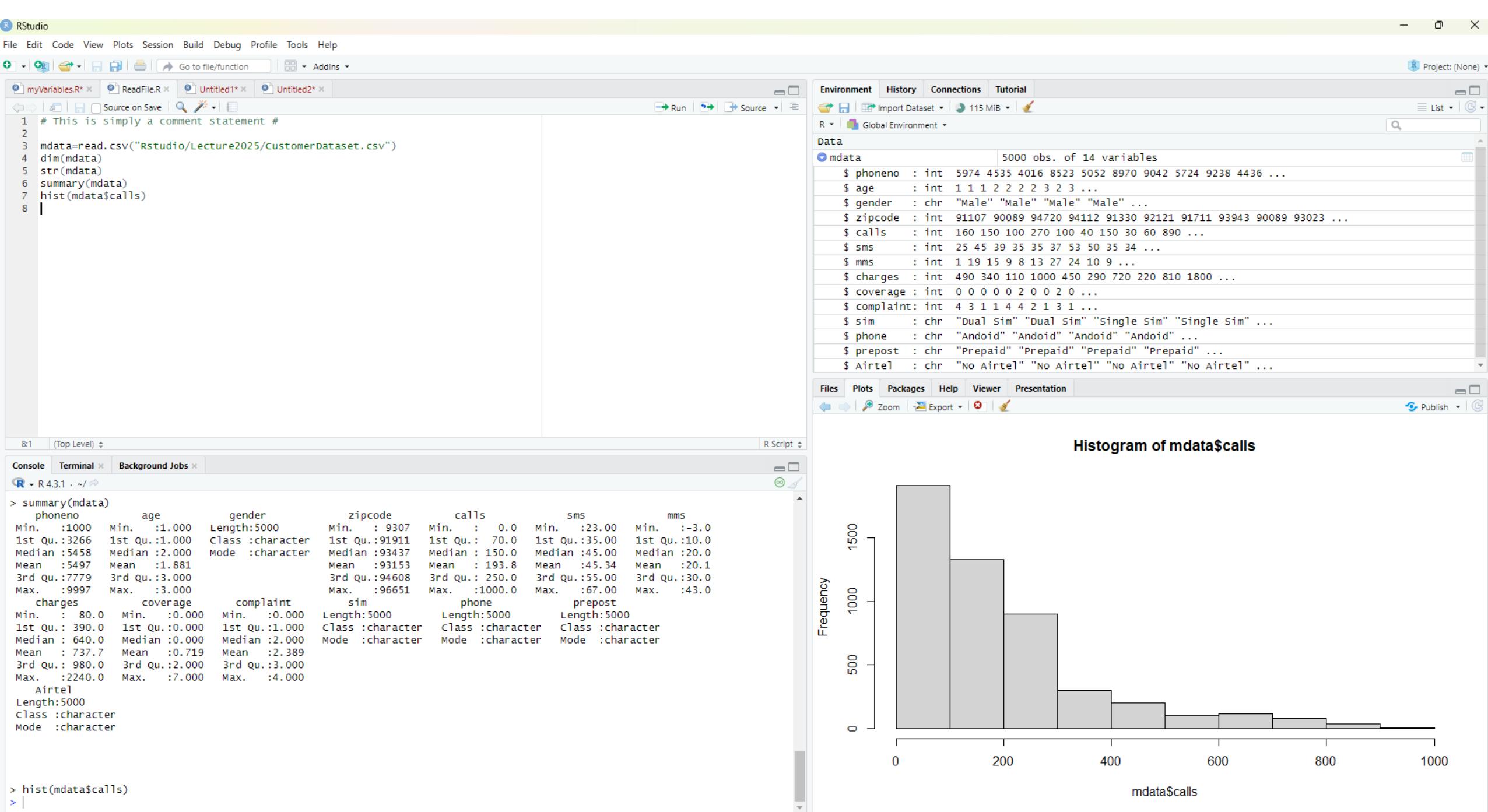
```
> source("~/Rstudio/Lecture2025/ReadFile.R", echo = TRUE)

> # This is simply a comment statement #
>
> mdata=read.csv("Rstudio/Lecture2025/customerDataset.csv")

> dim(mdata)
[1] 5000 14

> str(mdata)
'data.frame': 5000 obs. of 14 variables:
 $ phoneno : int 5974 4535 4016 8523 5052 8970 9042 5724 9238 4436 ...
 $ age      : int 1 1 1 2 2 2 3 2 3 ...
 $ gender   : chr "Male" "Male" "Male" ...
 $ zipcode  : int 91107 90089 94720 94112 91330 92121 91711 93943 90089 93023 ...
 $ calls    : int 160 150 100 270 100 40 150 30 60 890 ...
 $ sms      : int 25 45 39 35 35 37 53 50 35 34 ...
 $ mms      : int 1 19 15 9 8 13 27 24 10 9 ...
 $ charges  : int 490 340 110 1000 450 290 720 220 810 1800 ...
 $ coverage : int 0 0 0 0 0 2 0 0 2 0 ...
 $ complaint: int 4 3 1 1 4 4 2 1 3 1 ...
 $ sim      : chr "Dual sim" "dual sim" "single sim" "single sim" ...
 $ phone    : chr "Android" "Android" "Android" "Android" ...
 $ prepost  : chr "Prepaid" "Prepaid" "Prepaid" "Prepaid" ...
 $ Airtel   : chr "No Airtel" "No Airtel" "No Airtel" "No Airtel" ...
```

```
> summary(mdata)
      phoneno           age          gender        zipcode
  Min.   :1000   Min.   :1.000   Length:5000       Min.   : 9307
  1st Qu.:3266  1st Qu.:1.000   Class :character  1st Qu.:91911
  Median :5458   Median :2.000   Mode   :character  Median :93437
  Mean   :5497   Mean   :1.881
  3rd Qu.:7779  3rd Qu.:3.000
  Max.   :9997   Max.   :3.000
      calls            sms          mms        charges
  Min.   : 0.0   Min.   :23.00   Min.   :-3.0    Min.   : 80.0
  1st Qu.: 70.0  1st Qu.:35.00  1st Qu.:10.0   1st Qu.: 390.0
  Median :150.0  Median :45.00  Median :20.0    Median : 640.0
  Mean   :193.8  Mean   :45.34  Mean   :20.1    Mean   : 737.7
  3rd Qu.:250.0  3rd Qu.:55.00  3rd Qu.:30.0   3rd Qu.: 980.0
  Max.   :1000.0  Max.   :67.00  Max.   :43.0    Max.   :2240.0
      coverage         complaint        sim          phone
  Min.   :0.000   Min.   :0.000   Length:5000       Length:5000
  1st Qu.:0.000   1st Qu.:1.000   Class :character  Class :character
  Median :0.000   Median :2.000   Mode   :character  Mode   :character
  Mean   :0.719   Mean   :2.389
  3rd Qu.:2.000   3rd Qu.:3.000
  Max.   :7.000   Max.   :4.000
      prepost          Airtel
  Length:5000       Length:5000
  Class :character  Class :character
  Mode  :character  Mode  :character
```



File Edit Code View Plots Session Build Debug Profile Tools Help

myVariables.R* ReadFile.R* Untitled1* Untitled2*

Source on Save

```

1 # This is simply a comment statement #
2
3 mdata=read.csv("Rstudio/Lecture2025/CustomerDataset.csv")
4 dim(mdata)
5 str(mdata)
6 summary(mdata)
7 hist(mdata$calls)
8 hist(mdata$sms)
9

```

Run Source

Environment History Connections Tutorial

Import Dataset 111 MiB

Global Environment

Data

mdata 5000 obs. of 14 variables

	phoneno	age	gender	zipcode	calls	sms	mms
\$ phoneno	: int	5974 4535 4016 8523 5052 8970 9042 5724 9238 4436 ...					
\$ age	: int	1 1 1 2 2 2 3 2 3 ...					
\$ gender	: chr	"Male" "Male" "Male" ...					
\$ zipcode	: int	91107 90089 94720 94112 91330 92121 91711 93943 90089 93023 ...					
\$ calls	: int	160 150 100 270 100 40 150 30 60 890 ...					
\$ sms	: int	25 45 39 35 35 37 53 50 35 34 ...					
\$ mms	: int	1 19 15 9 8 13 27 24 10 9 ...					
\$ charges	: int	490 340 110 1000 450 290 720 220 810 1800 ...					
\$ coverage	: int	0 0 0 0 0 2 0 0 2 0 ...					
\$ complaint	: int	4 3 1 1 4 4 2 1 3 1 ...					
\$ sim	: chr	"Dual Sim" "Dual sim" "Single sim" "Single sim" ...					
\$ phone	: chr	"Android" "Android" "Android" "Android" ...					
\$ prepost	: chr	"Prepaid" "Prepaid" "Prepaid" "Prepaid" ...					
\$ Airtel	: chr	"No Airtel" "No Airtel" "No Airtel" "No Airtel" ...					

Files Plots Packages Help Viewer Presentation

Zoom Export

Publish

9:1 (Top Level)

Console Terminal Background Jobs

R Script

```

R 4.3.1 ~/~/
phoneno      age      gender      zipcode      calls      sms      mms
Min. :1000   Min. :1.000   Length:5000   Min. : 9307   Min. : 0.0   Min. :23.00   Min. : -3.0
1st Qu.:3266  1st Qu.:1.000   Class :character  1st Qu.:91911  1st Qu.: 70.0   1st Qu.:35.00   1st Qu.:10.0
Median :5458   Median :2.000   Mode  :character  Median :93437   Median :150.0   Median :45.00   Median :20.0
Mean  :5497   Mean  :1.881   Mean  :93153   Mean  :94608   Mean  :193.8   Mean  :45.34   Mean  :20.1
3rd Qu.:7779  3rd Qu.:3.000   3rd Qu.:94608  3rd Qu.:250.0  3rd Qu.:250.0   3rd Qu.:55.00   3rd Qu.:30.0
Max. :9997   Max. :3.000   Max. :96651   Max. :1000.0  Max. :1000.0   Max. :67.00   Max. :43.0

charges      coverage      complaint      sim      phone      prepost
Min. : 80.0   Min. :0.000   Min. :0.000   Length:5000   Length:5000   Length:5000
1st Qu.:390.0  1st Qu.:0.000  1st Qu.:1.000   Class :character  Class :character  Class :character
Median :640.0   Median :0.000  Median :2.000   Mode  :character  Mode  :character  Mode  :character
Mean  :737.7   Mean  :0.719   Mean  :2.389
3rd Qu.:980.0  3rd Qu.:2.000  3rd Qu.:3.000
Max. :2240.0  Max. :7.000   Max. :4.000

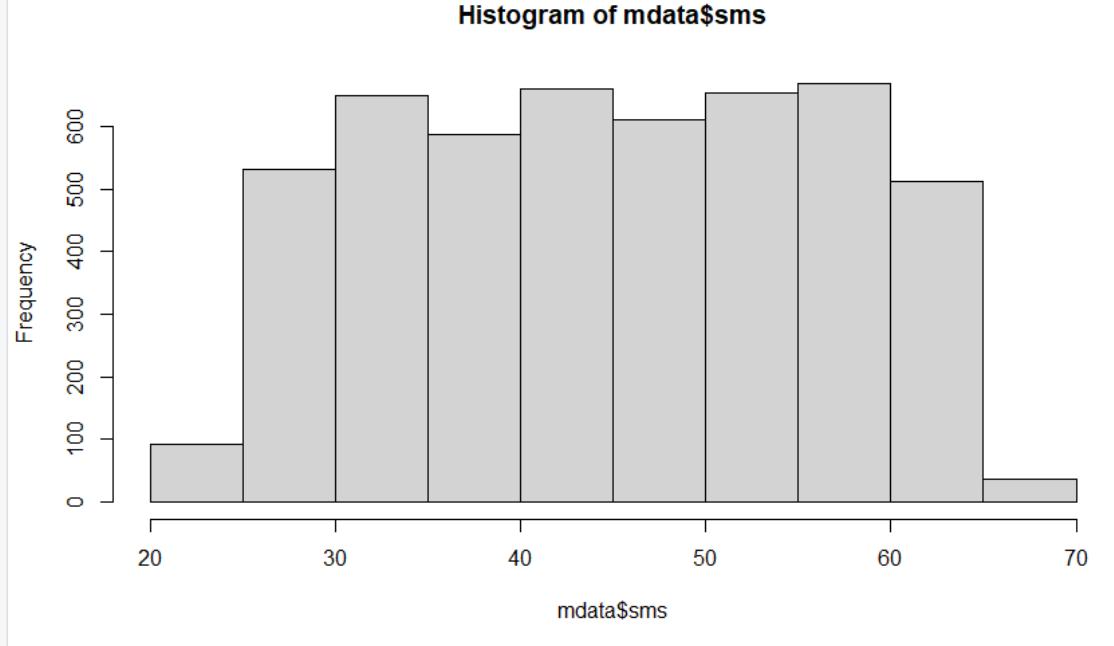
Airtel
Length:5000
Class :character
Mode  :character

```

> hist(mdata\$calls)

> hist(mdata\$sms)

>



```

> Complex_no <- 6 + 8i
> imaginary_part <- Im(Complex_no)
> Real_part <- Re(Complex_no)
> plot(Real_part, imaginary_part,
+       xlab = "Real ", ylab = "Imaginary",
+       main = "Complex Variable in Complex Plane")
> points(Real_part, imaginary_part, col = "red", pch = 16)

```

```

29 complex1 <- 5 + 7i
30 complex2 <- 10 - 9i
31
32 # Addition of complex numbers
33 result_addition <- complex1 + complex2
34 print(result_addition)
35
36 # Multiplication of complex numbers
37 result_multiplication <- complex1 * complex2
38 print(result_multiplication)
39
40
41

```

39:1 (Top Level) ▾

Console Terminal x Background Jobs x

R 4.3.1 · ~/

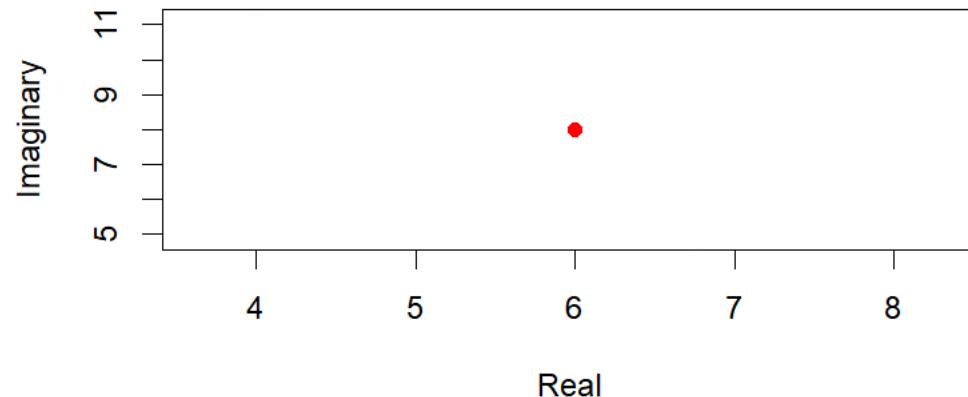
```

· points(Real_part, imaginary_part, col = "red", pch = 16)

·
·
· complex1 <- 5 + 7i
· complex2 <- 10 - 9i
· # Addition of complex numbers
· result_addition <- complex1 + complex2
· print(result_addition)
1] 15-2i
· # Multiplication of complex numbers
· result_multiplication <- complex1 * complex2
· print(result_multiplication)
1] 113+25i

```

Complex Variable in Complex Plane



```
> # Create two matrices
matrix1 <- matrix(c(1, 2, 3, 4), nrow = 2, ncol = 2)
matrix2 <- matrix(c(5, 6, 7, 8), nrow = 2, ncol = 2)

# Perform matrix multiplication
result <- matrix1 %*% matrix2
> matrix1
     [,1] [,2]
[1,]    1    3
[2,]    2    4
> matrix2
     [,1] [,2]
[1,]    5    7
[2,]    6    8
> result
     [,1] [,2]
[1,]   23   31
[2,]   34   46
> 
> matrix3 <- matrix(c(123,123,1244,1223,1243,1231,1235,6435,123,1243,1234,1223,1233,1234,123),nrow=5, ncol=3)
> matrix3
     [,1] [,2] [,3]
[1,] 123 1231 1234
[2,] 123 1235 1223
[3,] 1244 6435 1233
[4,] 1223 123 1234
[5,] 1243 1243 123
> inverse_matrix <-solve(matrix3)
Error in solve.default(matrix3) : 'a' (5 x 3) must be square
> matrix3 <- matrix(c(123,123,1244,1223,1243,1231,1235,6435,123,1243,1234,1223,1233,1234,123,1234),nrow=4, ncol=4)
> inverse_matrix <-solve(matrix3)
> print(inverse_matrix)
            [,1]          [,2]          [,3]          [,4]
[1,]  0.0010019421 -8.838746e-04  1.006047e-03 -2.175341e-04
[2,] -0.0002152302 -1.898778e-06 -2.126631e-04  2.381519e-04
[3,] -0.0008960716  8.928359e-04 -2.380556e-06  2.746857e-06
[4,]  0.0010174445  1.020227e-06  1.142655e-04 -2.186569e-04
```

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Install new package or library

Project: (None)

```
myVariables.R x ReadFile.R* x
Source on Save Run Source
41
42
43
44
45
46
47
48
49
50 New_mdata<-mdata[,4:9]
51 New_mdata_cor <- cor(New_mdata)
52 corrplot(New_mdata_cor)
53
54
```

53:1 (Top Level) ▾

Console Terminal Background Jobs

R 4.3.1 · ~/

```
158 94025 267 41 17 830 0
159 94720 150 32 6 790 0
160 94545 170 61 35 410 0
161 95819 650 29 0 1340 0
162 95053 280 61 35 800 0
163 90036 200 38 12 520 0
164 91125 260 28 4 700 0
165 95120 110 53 27 920 0
166 94706 150 27 1 430 0
```

```
[ reached 'max' / getOption("max.print") -- omitted 4834 rows ]
```

```
> New_mdata_cor <- cor(New_mdata)
```

```
> corrplot(New_mdata_cor)
```

```
Error in corrplot(New_mdata_cor) : could not find function "corrplot"
```

```
> New_mdata
```

Install Packages

Install from: [Configuring Repositories](#)
Repository (CRAN)

Packages (separate multiple with space or comma):
corrplot

Install to Library:
C:/Users/vishis/AppData/Local/R/win-library/4.3 [Default]

Install dependencies

Install Cancel

Environment	History	Connections	Tutorial
	Import Dataset	189 MiB	
R	Global Environment		List
\$ coverage : int	0 0 0 0 0 2 0 0 2 0 ...		
\$ complaint: int	4 3 1 1 4 4 2 1 3 1 ...		
\$ sim : chr	"Dual Sim" "Dual Sim" "Single Sim" "S...		
	chr "Android" "Android" "Android" "Android" ...		
	chr "Prepaid" "Prepaid" "Prepaid" "Prepaid" ...		
	chr "No Airtel" "No Airtel" "No Airtel" "No..."		
	5000 obs. of 6 variables		
	num [1:6, 1:6] 1 -0.00406 -0.02922 -0.028...		

Description	Version	
Code Analysis Tools for R	0.2-19	
The R Compiler Package	4.3.1	
The R Graphics Devices and Support for Colours and Fonts	4.3.1	
Mixed GAM Computation Vehicle with Automatic Smoothness Estimation	1.8-42	
Support for Parallel Computation in R	4.3.1	

K RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins Retrieving package installation context... Project: (None)

myVariables.R ReadFile.R*

Source on Save Run Source

41
42
43
44
45
46
47
48
49
50 New_mdata<-mdata[,4:9]
51 New_mdata_cor <- cor(New_mdata)
52 corplot(New_mdata_cor)

Environment History Connections Tutorial

Import Dataset 189 MiB

R Global Environment

\$ coverage : int 0 0 0 0 0 2 0 0 2 0 ...
\$ complaint: int 4 3 1 1 4 4 2 1 3 1 ...
\$ sim : chr "Dual Sim" "Dual Sim" "Single Sim" "Single Sim" ...
\$ phone : chr "Android" "Android" "Android" "Android" ...
\$ prepost : chr "Prepaid" "Prepaid" "Prepaid" "Prepaid" ...
\$ Airtel : chr "No Airtel" "No Airtel" "No Airtel" "No Airtel" ...

New_mdata 5000 obs. of 6 variables
New_mdata_cor num [1:6, 1:6] 1 -0.00406 -0.02922 -0.028...

Files Plots Packages Help Viewer Presentation

Install Update

co

The screenshot shows the RStudio interface. In the top-left, there are tabs for 'myVariables.R' and 'ReadFile.R*'. Below them is a toolbar with icons for file operations like 'Source on Save' and 'Run'. The main workspace contains R code from lines 41 to 54. The right side features the 'Global Environment' pane, which lists variables like 'coverage', 'complaint', 'sim', 'phone', 'prepost', and 'Airtel' along with their types and values. Below this is a table showing 'New_mdata' and 'New_mdata_cor'. At the bottom, there are tabs for 'Files', 'Plots', 'Packages', 'Help', 'Viewer', and 'Presentation', along with 'Install' and 'Update' buttons. A search bar at the bottom right contains the letter 'co'.

Useful place to learn R

<https://www.mayin.org/ajayshah/KB/R/html/p3.html>


```
mdata=read.csv("Rstudio/Lecture2022/CustomerDataset.csv")
dim(mdata)
str(mdata)
summary(mdata)
hist(mdata$calls)
hist(mdata$sms)
```