

# Introduction to R

**AGR 5266C**  
**Field Plot Techniques**  
**(Sections 0877 and 1933)**

**Dev Paudel**

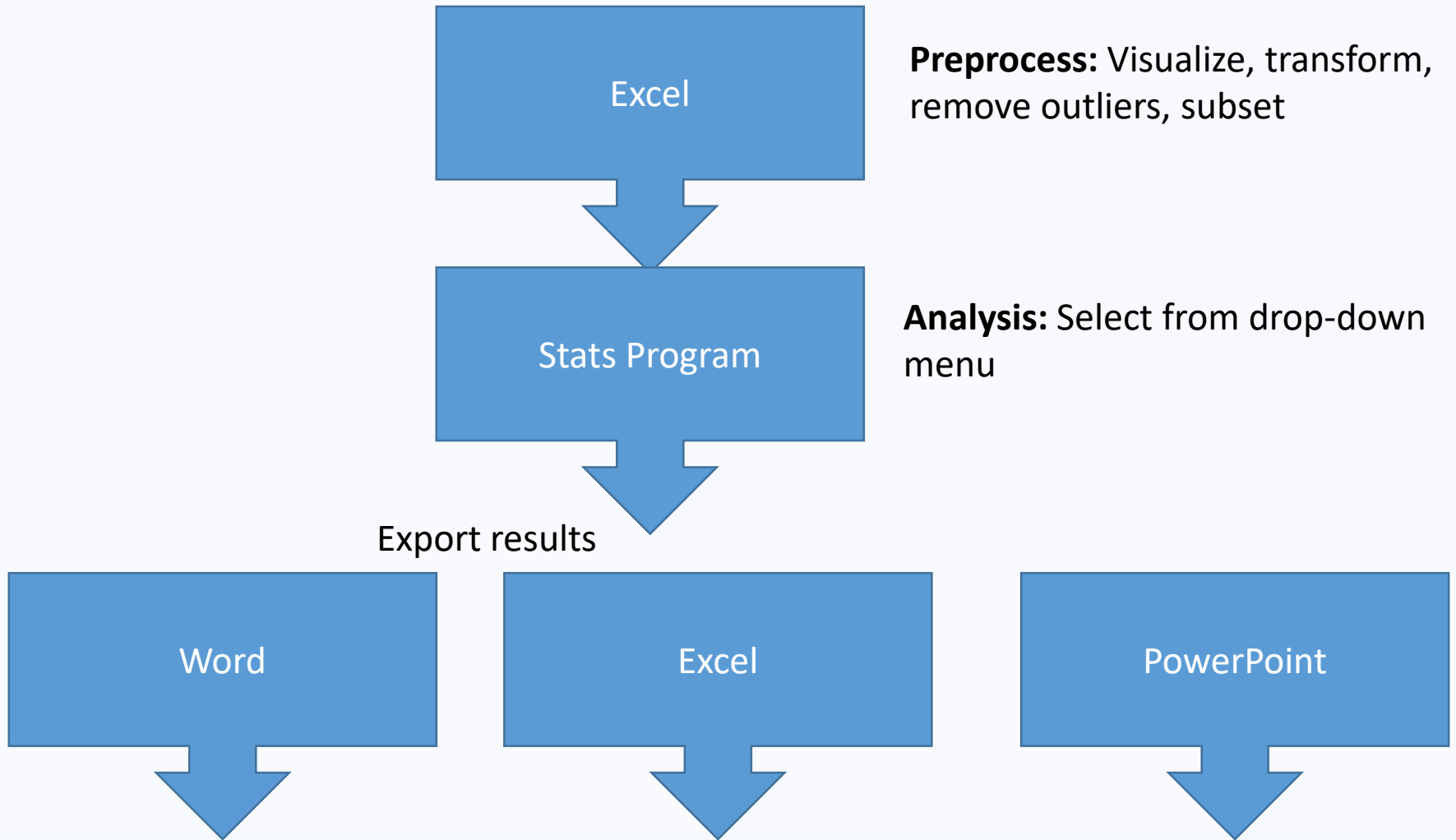


Studio<sup>®</sup>

# Computer programming for statistics

- Current practices
  - Statistics are often done in “canned” programs with dropdown menus eg SPSS, Excel.
  - Involve preprocessing in spreadsheets like Excel
  - Results are exported to make graphics and tables

# Multiple steps in analysis

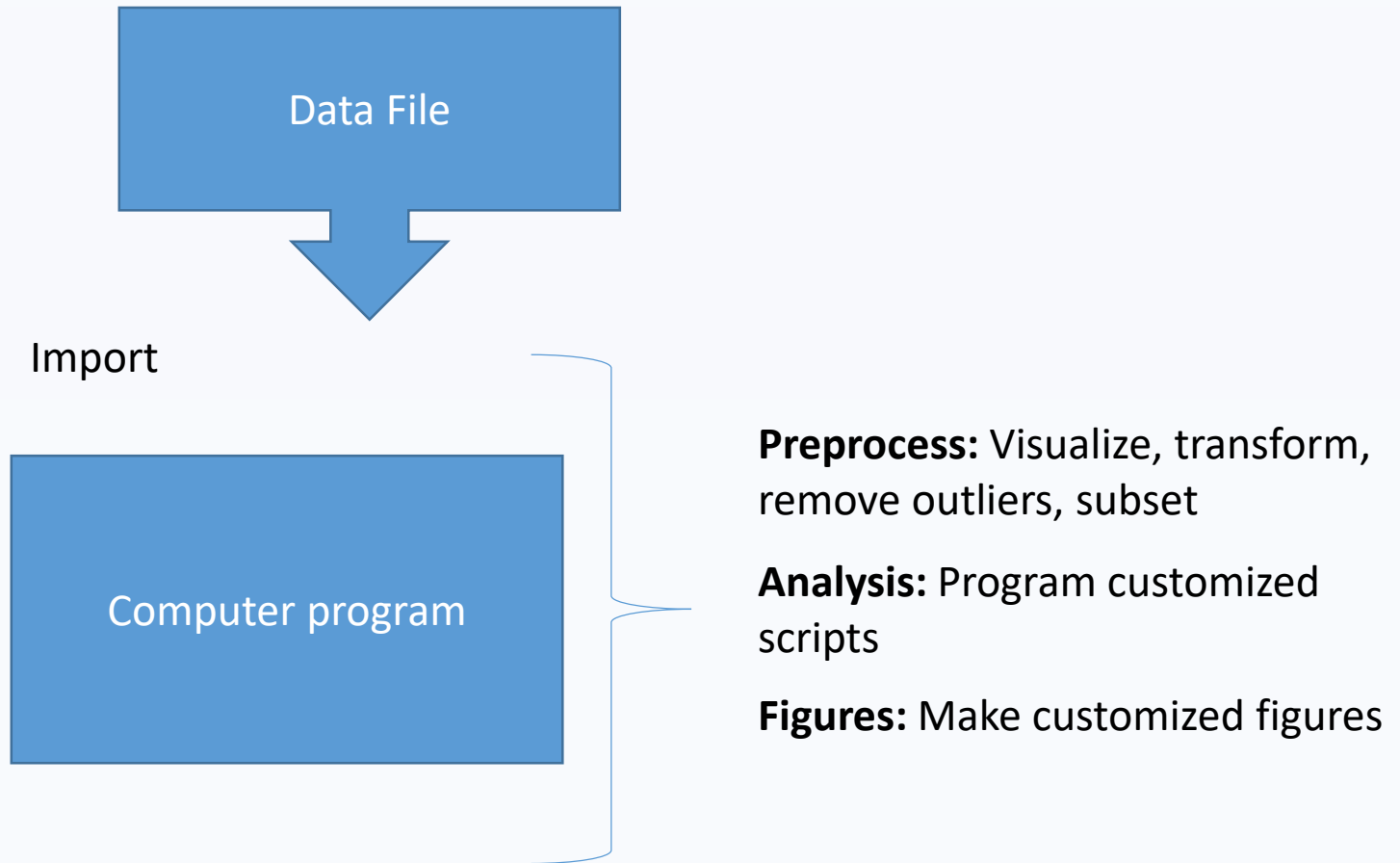


Figures: Make figures or alter figures

# How can programming help?

- Allows for flexible and customized analysis
  - In contrast to drop-down menus
- Allows preprocessing in the same software
  - Avoids use of multiple software
- Allows customizable graphs and tables directly

# How can programming help?



Streamline data analysis

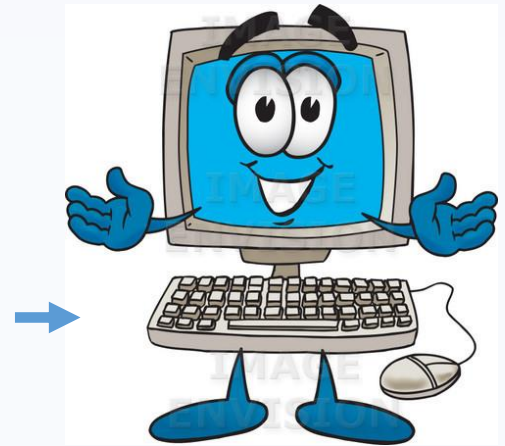
# How can programming help?

- Less room for error
- Easily documented analysis
- Allows reproducible science



Experimental design

→ Data collection



Analysis

# Reproducible research

- The goal of **reproducible research** is to tie specific instructions to data analysis and experimental data so that scholarship can be **recreated**, better **understood** and **verified**.

CRAN Task View: Reproducible Research

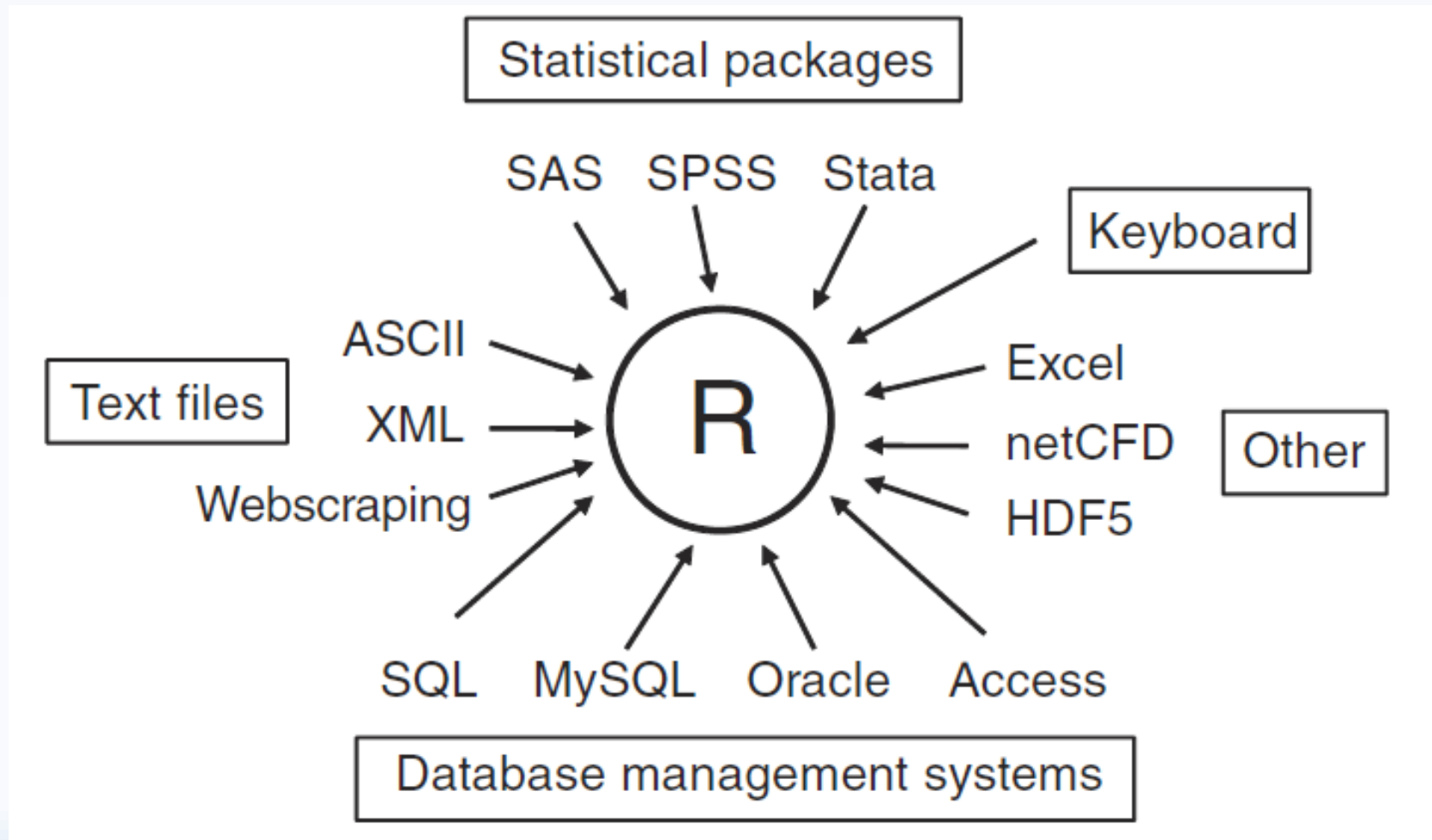
# What is R and why use R

- Powerful tool for statistics, graphics, and statistical programming
- R is free
- Contains advanced statistical platform
- Can easily import data from different software

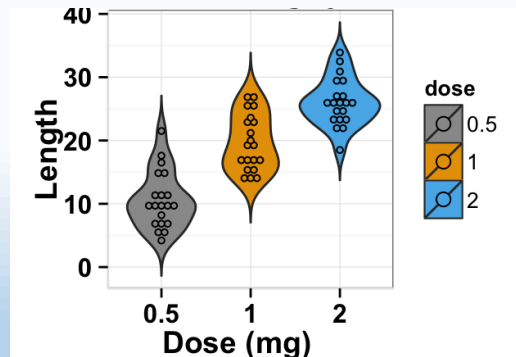
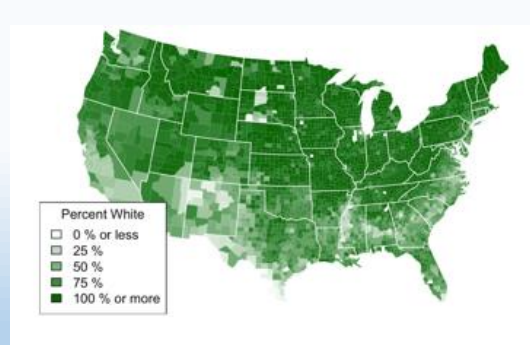
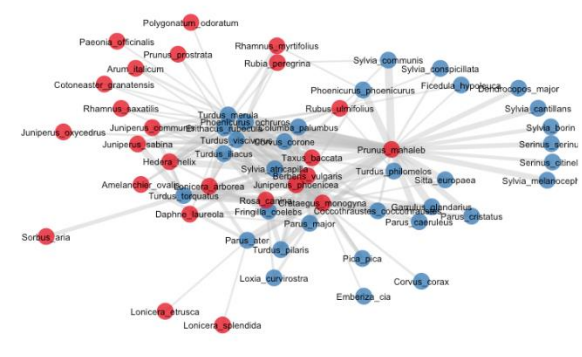
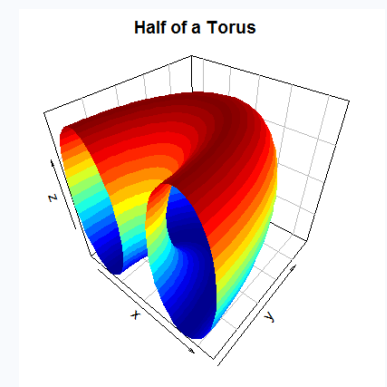
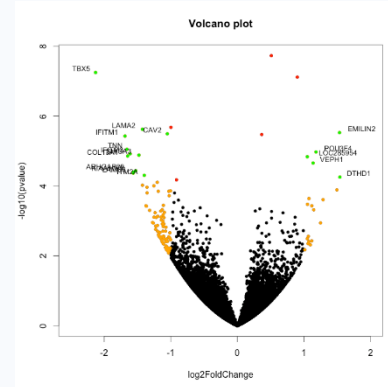
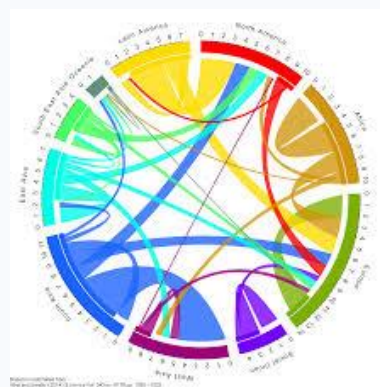
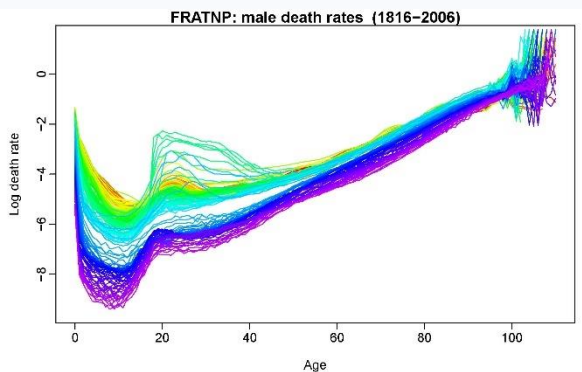
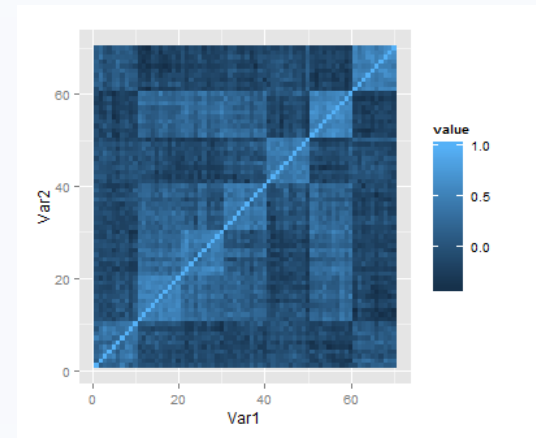
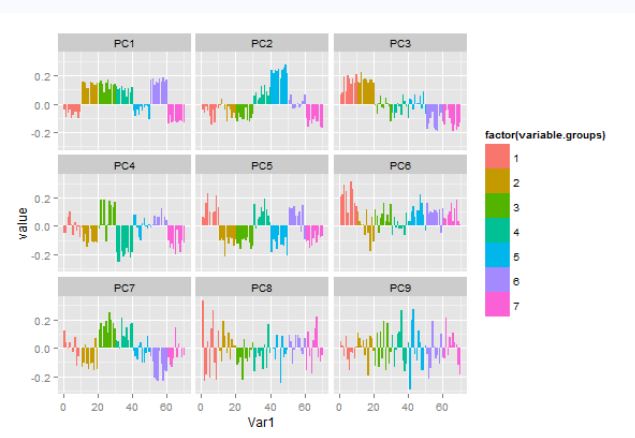
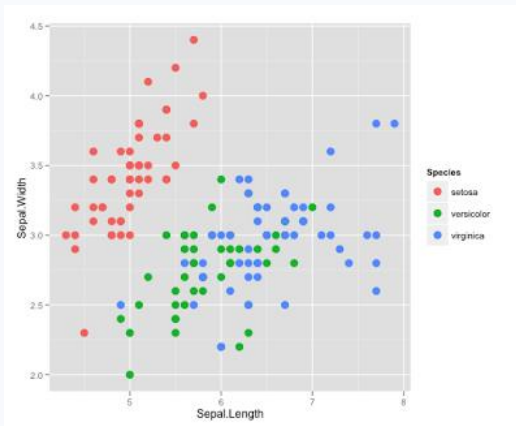




# What is R and why use R



# What is R and why use R



# Exploring R studio

RStudio

File Edit Code View Plots Session Build Debug Tools Help

Go to file/function

exomseq\_snpwisc.R \* subset.R\* \* Untitled1\* \* Untitled1\* \* Untitled2\* \* Untitled3\* \* Untitled4\* \*

Source on Save Run Source

```
1 x<-c(1:10)
2 y<-c(21:30)
3 plot(x,y)
4
```

4:1 (Top Level) R Script

Console C:/Users/Dev Paudel/Dropbox/ALS5932/module 3/

```
> x<-c(1:10)
> y<-c(21:30)
> plot(x,y)
>
```

Environment History

Import Dataset

Global Environment

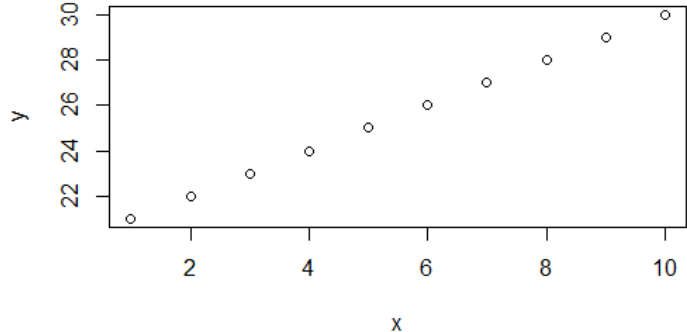
values

x	int [1:10]	1	2	3	4	5	6	7	8	9	10
y	int [1:10]	21	22	23	24	25	26	27	28	29	30

Files Plots Packages Help Viewer

Zoom Export

Publish



A scatter plot with 'x' on the horizontal axis and 'y' on the vertical axis. The x-axis ranges from 1 to 10 with major ticks every 2 units. The y-axis ranges from 21 to 30 with major ticks every 2 units. There are 10 data points, each represented by an open circle. The points follow a perfectly linear upward trend, starting at (1, 21) and ending at (10, 30).

x	y
1	21
2	22
3	23
4	24
5	25
6	26
7	27
8	28
9	29
10	30

# Exploring R studio

The image shows the RStudio interface with a red box highlighting the script editor and console. The script editor contains the following code:

```
1 x<-c(1:10)
2 y<-c(21:30)
3 plot(x,y)
4
```

The console shows the execution of the same code:

```
> x<-c(1:10)
> y<-c(21:30)
> plot(x,y)
>
```

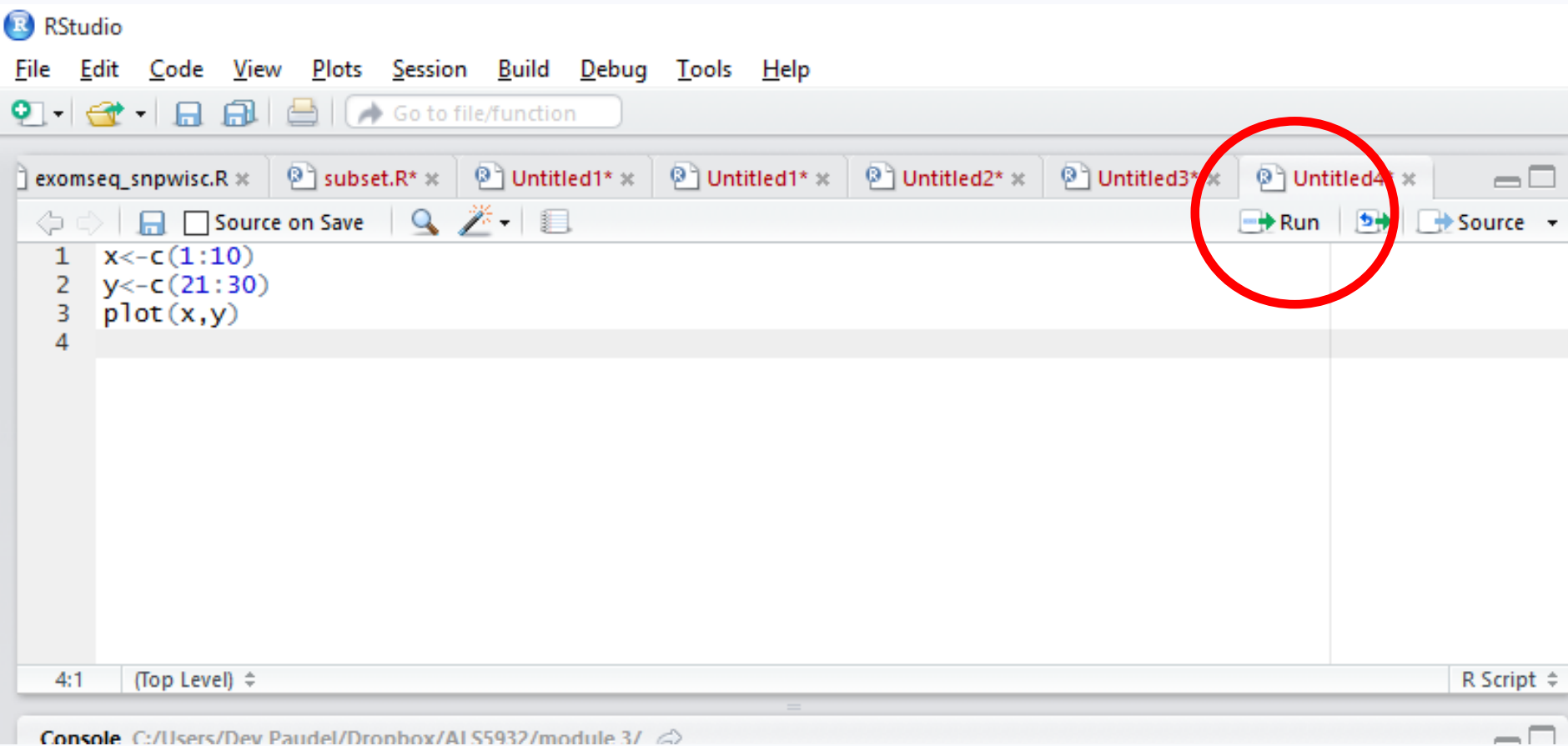
The Environment pane on the right shows the data objects created:

Global Environment											
x	int [1:10]	1	2	3	4	5	6	7	8	9	10
y	int [1:10]	21	22	23	24	25	26	27	28	29	30

The Plots pane shows a scatter plot of y versus x. The x-axis ranges from 1 to 10, and the y-axis ranges from 21 to 30. The data points are as follows:

x	y
1	21
2	22
3	23
4	24
5	25
6	26
7	27
8	28
9	29
10	30

# Exploring R studio



# Exploring R studio

The screenshot displays the RStudio environment with the following components:

- Source Editor:** Contains the following R code:

```
1 x<-c(1:10)
2 y<-c(21:30)
3 plot(x,y)
4
```
- Environment Pane:** Shows the global environment with the following values:

values	
x	int [1:10] 1 2 3 4 5 6 7 8 9 10
y	int [1:10] 21 22 23 24 25 26 27 28 29 30
- Console:** Displays the execution of the code:

```
> x<-c(1:10)
> y<-c(21:30)
> plot(x,y)
>
```
- Plots Pane:** Shows a scatter plot of y versus x. The x-axis ranges from 1 to 10, and the y-axis ranges from 21 to 30. The data points form a straight line with a positive slope.

# Exploring R studio

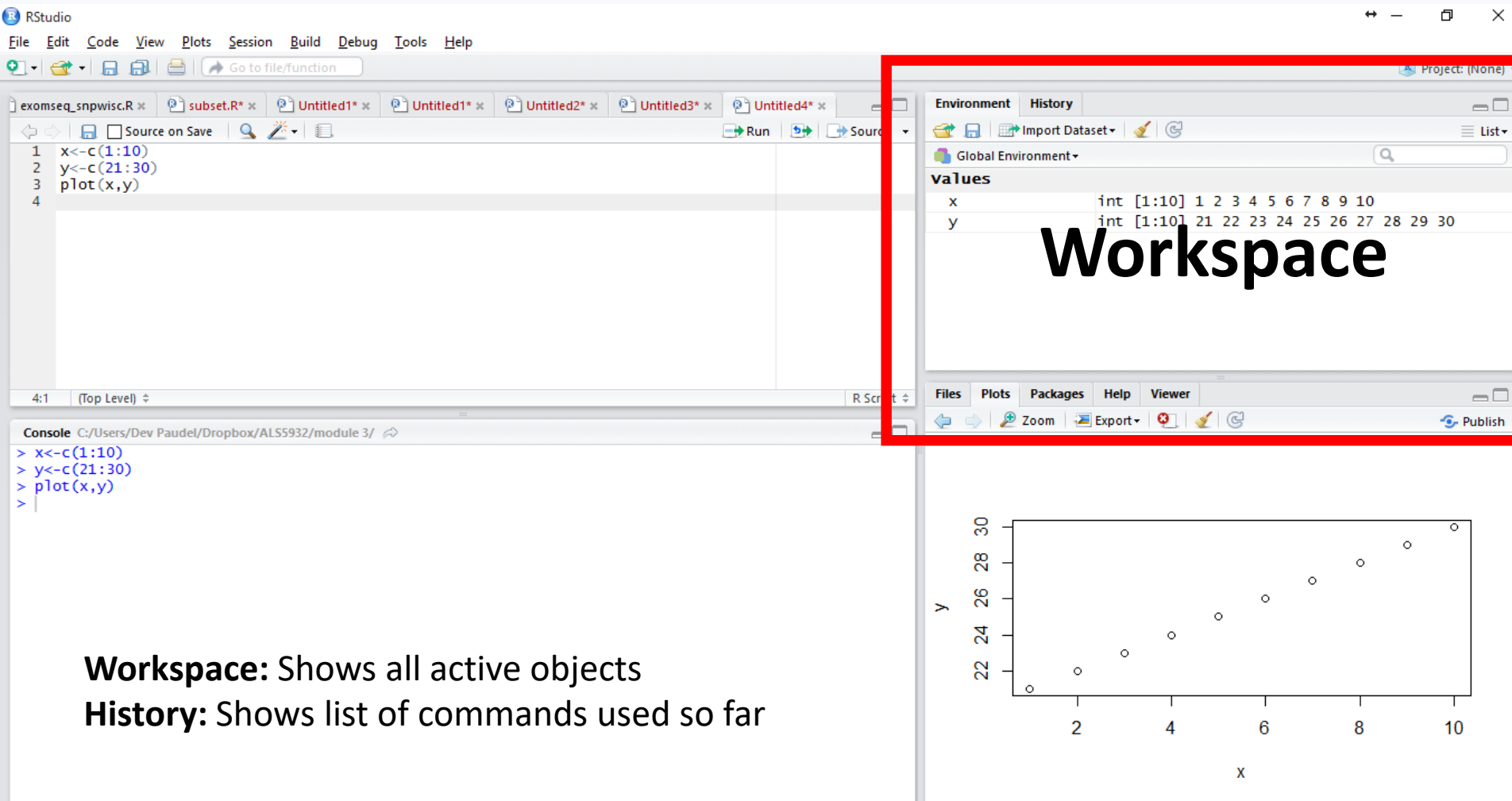


The screenshot shows the R Studio interface. At the top, there's a toolbar with a zoom level of 4:1 and a dropdown menu set to '(Top Level)'. To the right of the toolbar is a tab labeled 'R Script'. Below the toolbar is the console window, which has a title bar showing the file path 'C:/Users/Dev Paudel/Dropbox/ALS5932/module 3/'. The console contains the following R code and its output:

```
x<-c(1:10)
y<-c(21:30)
plot(x,y)
x
[1]  1  2  3  4  5  6  7  8  9 10
y
[1] 21 22 23 24 25 26 27 28 29 30
|
```

Console > You can type commands and see output

# Exploring R studio



The screenshot displays the RStudio environment. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Tools, and Help. The toolbar below contains icons for file operations and running code. The main editor window shows a script with the following code:

```
1 x<-c(1:10)
2 y<-c(21:30)
3 plot(x,y)
4
```

The Environment pane on the right, highlighted with a red border, shows the Global Environment with the following values:

values	
x	int [1:10] 1 2 3 4 5 6 7 8 9 10
y	int [1:10] 21 22 23 24 25 26 27 28 29 30

The word "Workspace" is overlaid in large black text on the Environment pane. Below the Environment pane, the Console shows the execution of the code:

```
> x<-c(1:10)
> y<-c(21:30)
> plot(x,y)
>
```

The bottom right pane displays a scatter plot of y versus x, showing a positive linear relationship. The x-axis ranges from 1 to 10, and the y-axis ranges from 21 to 30. The data points are represented by open circles.

**Workspace:** Shows all active objects  
**History:** Shows list of commands used so far



# Exploring R studio

The screenshot displays the RStudio environment with the following components:

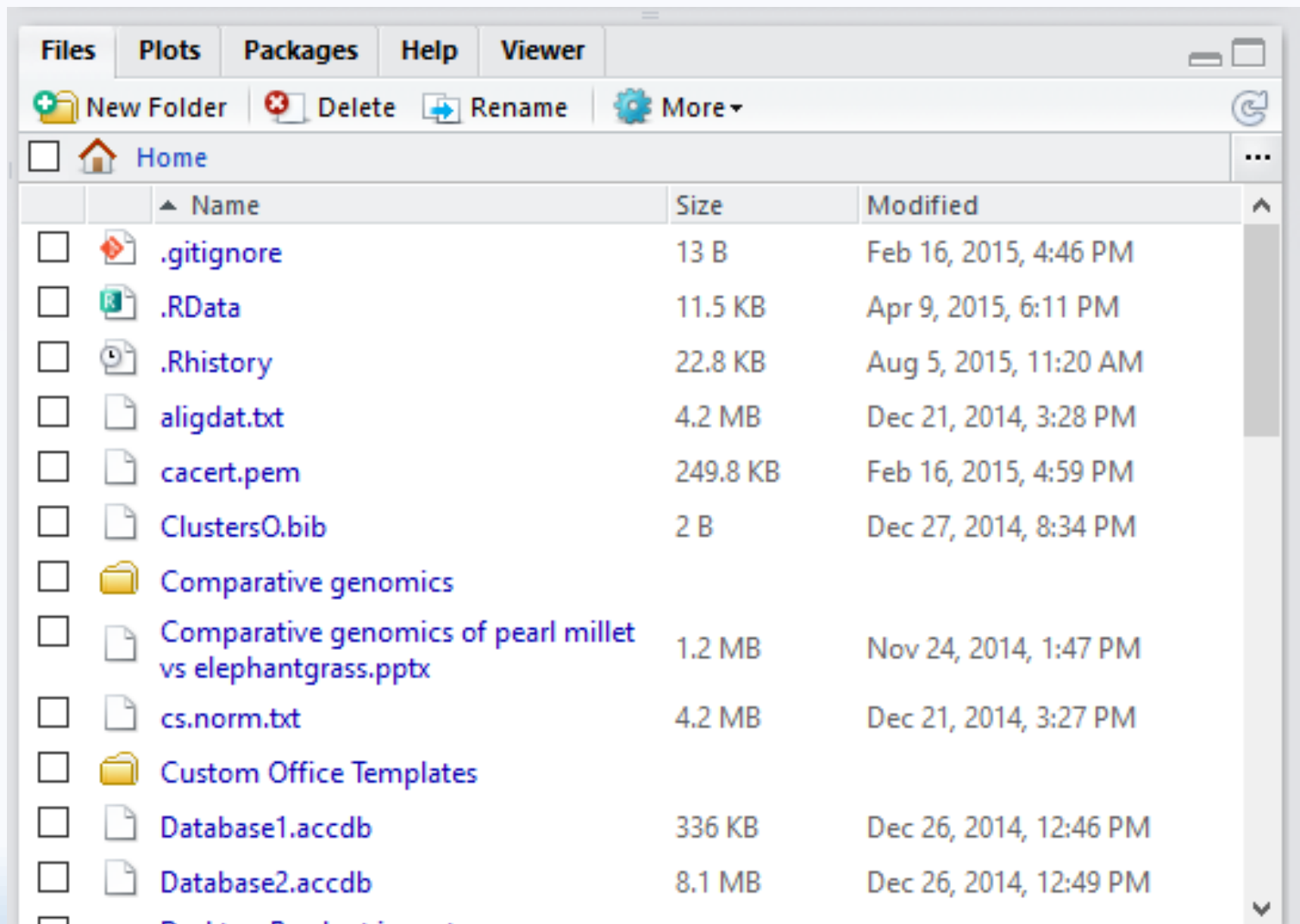
- Source Editor:** Contains an R script with the following code:

```
1 x<-c(1:10)
2 y<-c(21:30)
3 plot(x,y)
4
```
- Console:** Shows the execution of the script:

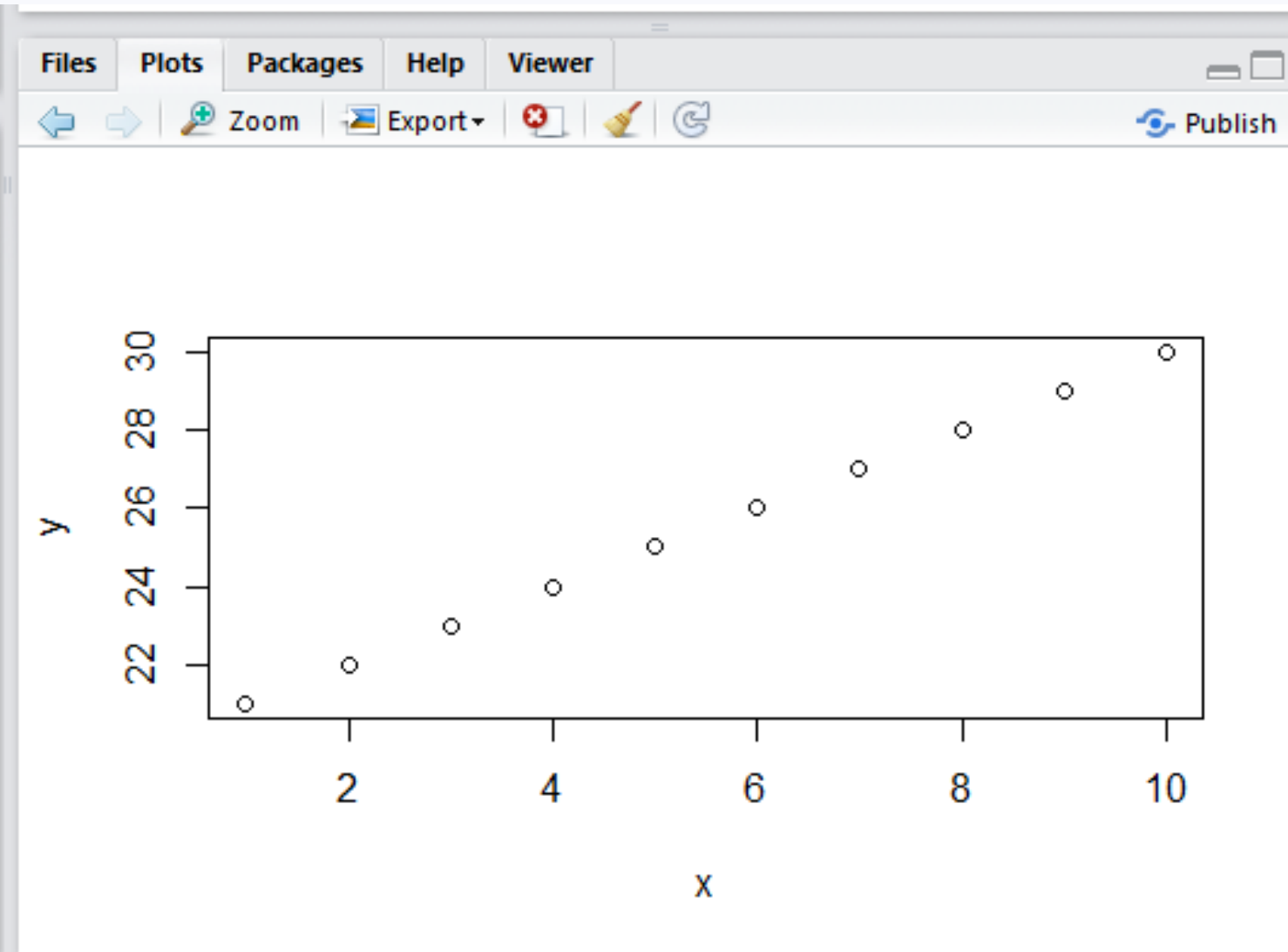
```
> x<-c(1:10)
> y<-c(21:30)
> plot(x,y)
>
```
- Environment Pane:** Displays the current environment with the following data:

values
x
int [1:10] 1 2 3 4 5 6 7 8 9 10
y
int [1:10] 21 22 23 24 25 26 27 28 29 30
- Plots Pane:** Displays a scatter plot of y versus x. The plot is highlighted with a red border. The x-axis ranges from 1 to 10, and the y-axis ranges from 21 to 30. The plot contains 10 data points, each represented by a small circle. Overlaid on the plot is the text "Files/plots/packages/help".

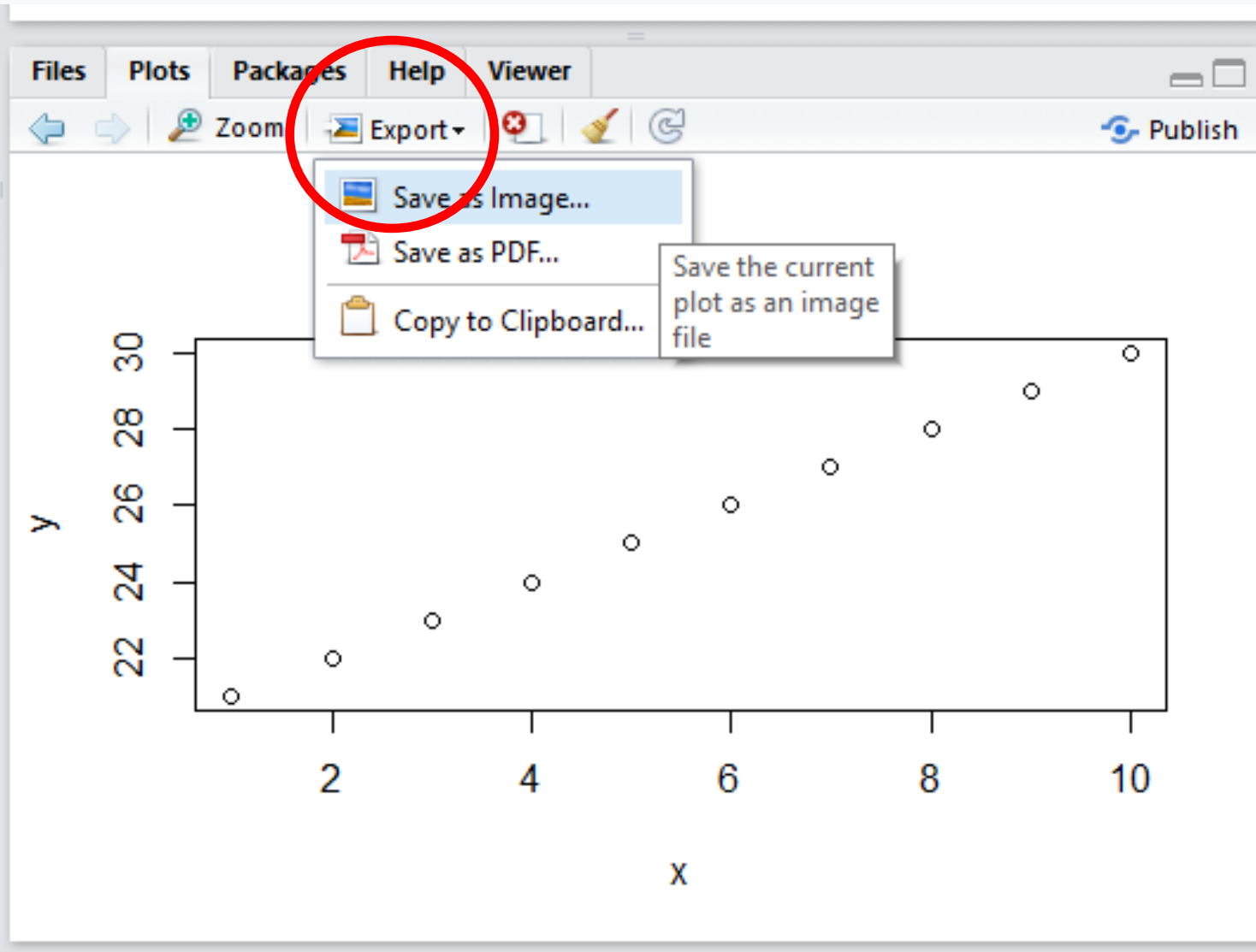
# Exploring R studio



# Exploring R studio



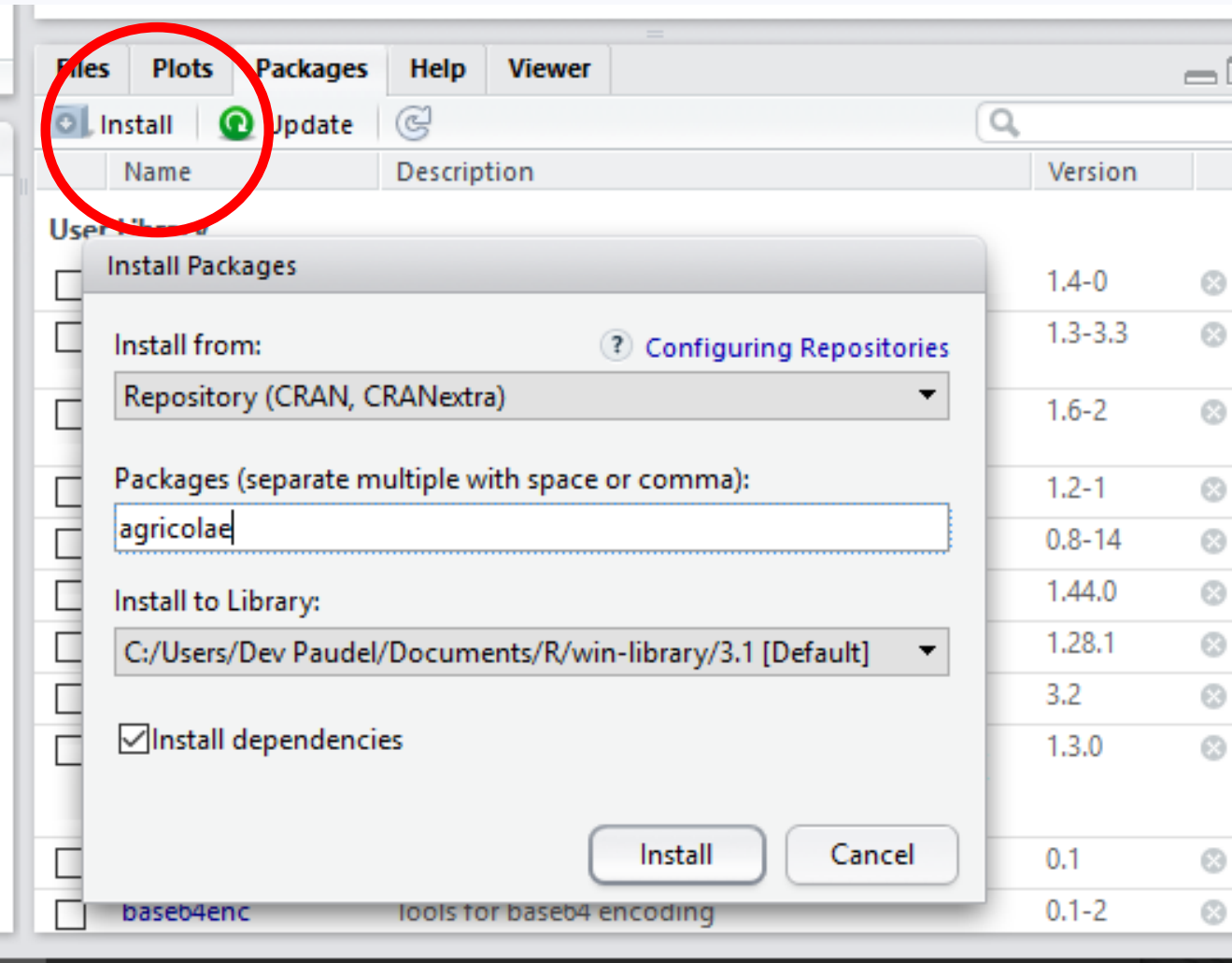
# Exploring R studio



# Exploring R studio

Files Plots Packages Help Viewer				
Install		Update		
Name		Description	Version	
User Library				
<input type="checkbox"/>	abind	Combine multi-dimensional arrays	1.4-0	ⓧ
<input type="checkbox"/>	acepack	ace() and avas() for selecting regression transformations	1.3-3.3	ⓧ
<input type="checkbox"/>	ade4	Analysis of Ecological Data : Exploratory and Euclidean methods in Environmental sciences	1.6-2	ⓧ
<input type="checkbox"/>	agricolae	Statistical Procedures for Agricultural Research	1.2-1	ⓧ
<input type="checkbox"/>	amap	Another Multidimensional Analysis Package	0.8-14	ⓧ
<input type="checkbox"/>	annotate	Annotation for microarrays	1.44.0	ⓧ
<input type="checkbox"/>	AnnotationDbi	Annotation Database Interface	1.28.1	ⓧ
<input type="checkbox"/>	ape	Analyses of Phylogenetics and Evolution	3.2	ⓧ
<input type="checkbox"/>	aplpack	Another Plot PACKage: stem.leaf, bagplot, faces, spin3R, plotsummary, plothulls, and some slider functions	1.3.0	ⓧ
<input type="checkbox"/>	assertthat	Easy pre and post assertions.	0.1	ⓧ

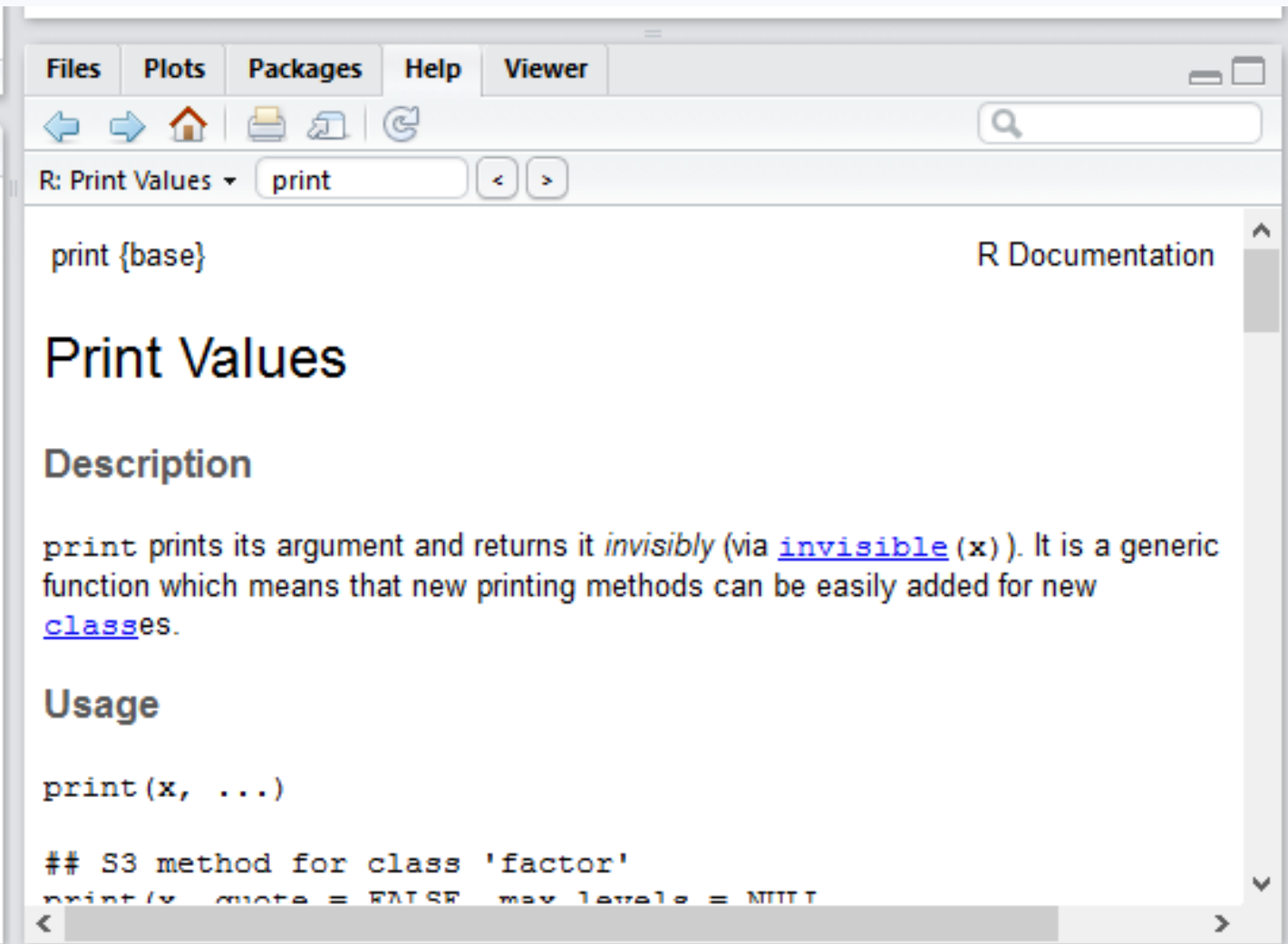
# Exploring R studio



`install.packages("agricolae")`

@devRpaudel

# Exploring R studio



?print  
??print

# Using R as a calculator





# Creating objects

## ## Naming Variables ##

**dice** = c(1,2,3,4,5,6) # c is for concatenation

**dice** <- c(1,2,3,4,5,6) # it is generally recommend using "<-" to assign  
the right parts to left object

# "=" usually reserve for assign value in the functions

- *Everything that exists is an object*
- *Everything that happens is a function call* -John Chambers

avg = mean (data)

avg <- mean (data)

Variable/ object

function

object

# Functions in R

```
dice <- c(1,2,3,4,5,6)
```

```
# Get average of dice
```

```
MEAN(dice) # R is case sensitive #
```

```
Mean(dice) # R is case sensitive #
```

```
mean (dice)
```

```
## Function within a function
```

```
log (x = sqrt(9))
```

```
log(3)
```

**Next: Data Structures**

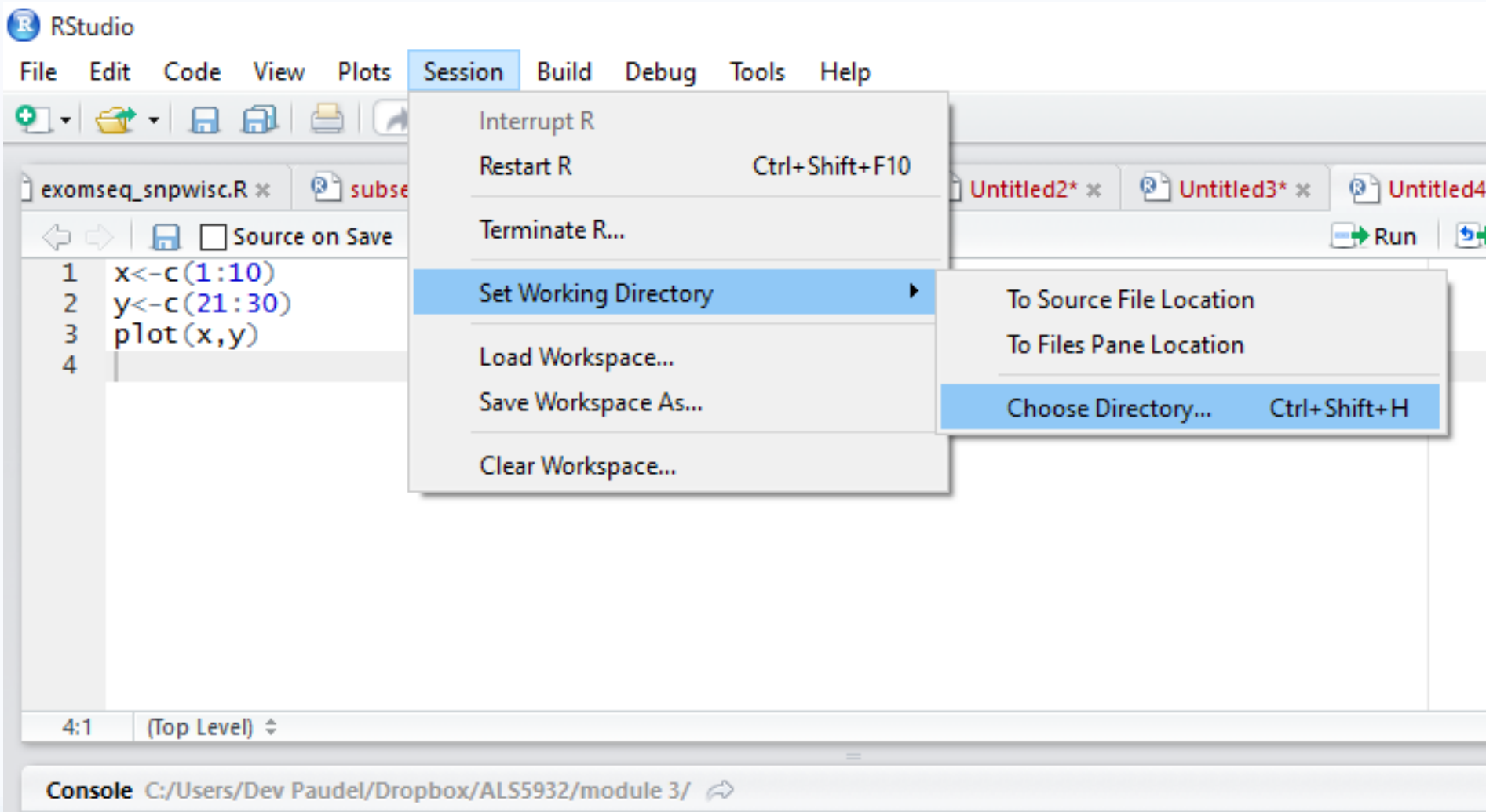


# Getting to know your data

- Setting working directory
- Reading data
- Exploring data



# Setting up working directory



# Importing data

**dir()**

~/Desktop/RLab

[https://raw.githubusercontent.com/dpaudel/cheatsheet/master/Wheat\\_91.txt](https://raw.githubusercontent.com/dpaudel/cheatsheet/master/Wheat_91.txt)

<https://raw.githubusercontent.com/dpaudel/cheatsheet/master/wheatc.csv>