NATIONAL ENGINEERING CENTER
University of the Philippines
Diliman, Quezon City

1.0 R for Business Intelligence and Analytics

Eugene Rex L. Jalao, Ph.D.

Associate Professor

Department Industrial Engineering and Operations Research

University of the Philippines Diliman

@thephdataminer

Module 6 of the Business Intelligence and Analytics Certification of UP NEC and the UP Center for Business Intelligence

UP NEC BI Modules

- Analyst Level
  1. Introduction to Business Intelligence
  2. Data Warehousing
  3. Data Mining
- Professional Level
  4. Time Series Analysis and Forecasting
  5. Optimization Analysis
  6. **Introduction to R Programming**

Outline for this Training

- Introduction to R and R Studio
- Data Types and Operators
  - Case Study on R Scripting
- Reading, Manipulating and Writing Data
  - Case Study on Dataset Analysis with ETL
- Basic R Programming
  - Case Study: Writing Functions
- Graphics and Plotting
- Deploying R and Dashboard Generation
  - Case Study: Deploying a Simple Dashboard
- Deploying R with C#
  - Case Study: A Simple Standalone GUI For R Apps

Outline for This Session

- Introductions
- **A Review of BA**
- What is R?
- Where to Get R and R Studio
- First Commands

A Review of BA

- Timely
- Accurate
- High-Value
- Actionable

DECISIONS

Via organizational (and sometimes external) data

eljalao@up.edu.ph

A Review of BA

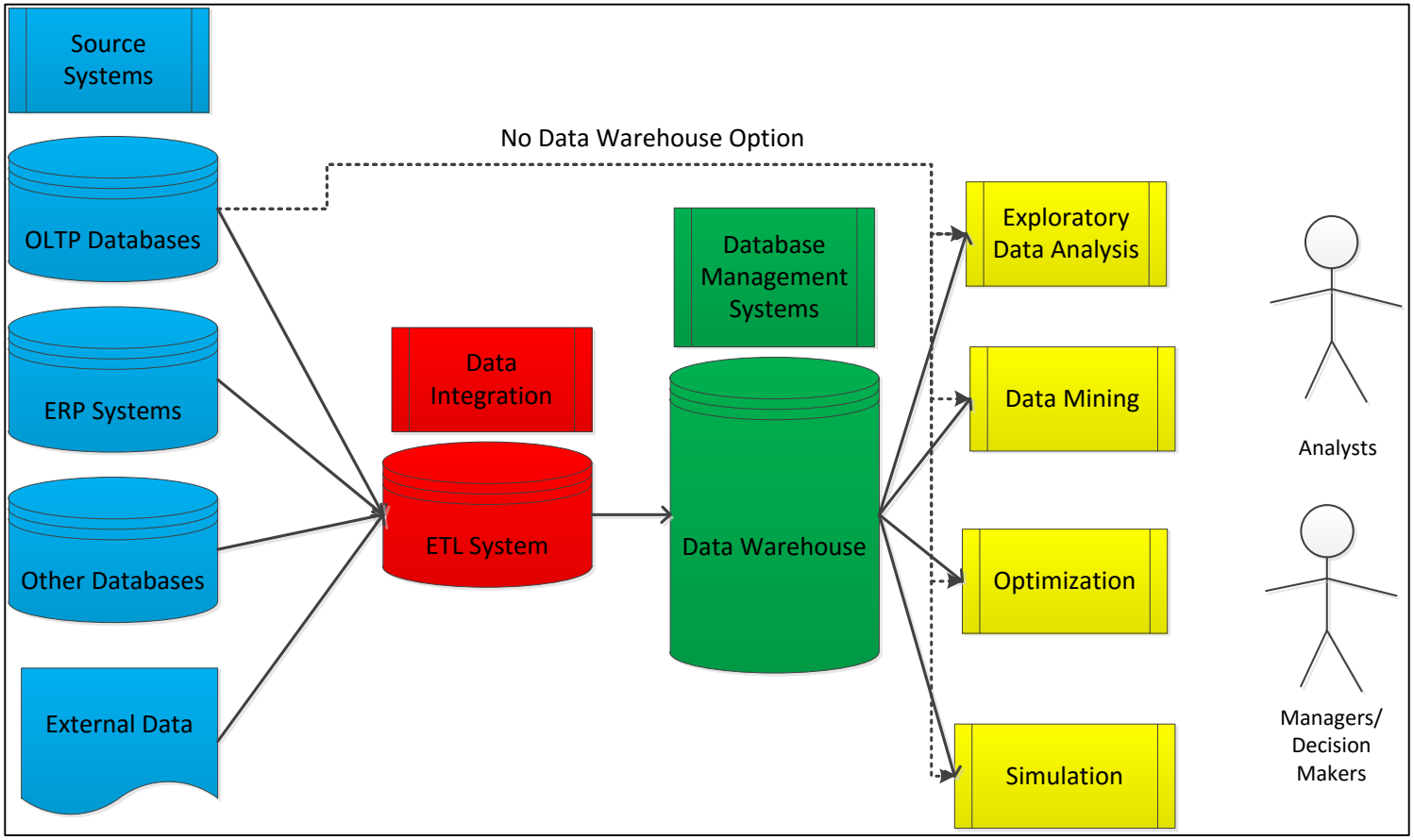


Figure 1.1: BA Framework

A Review of BA

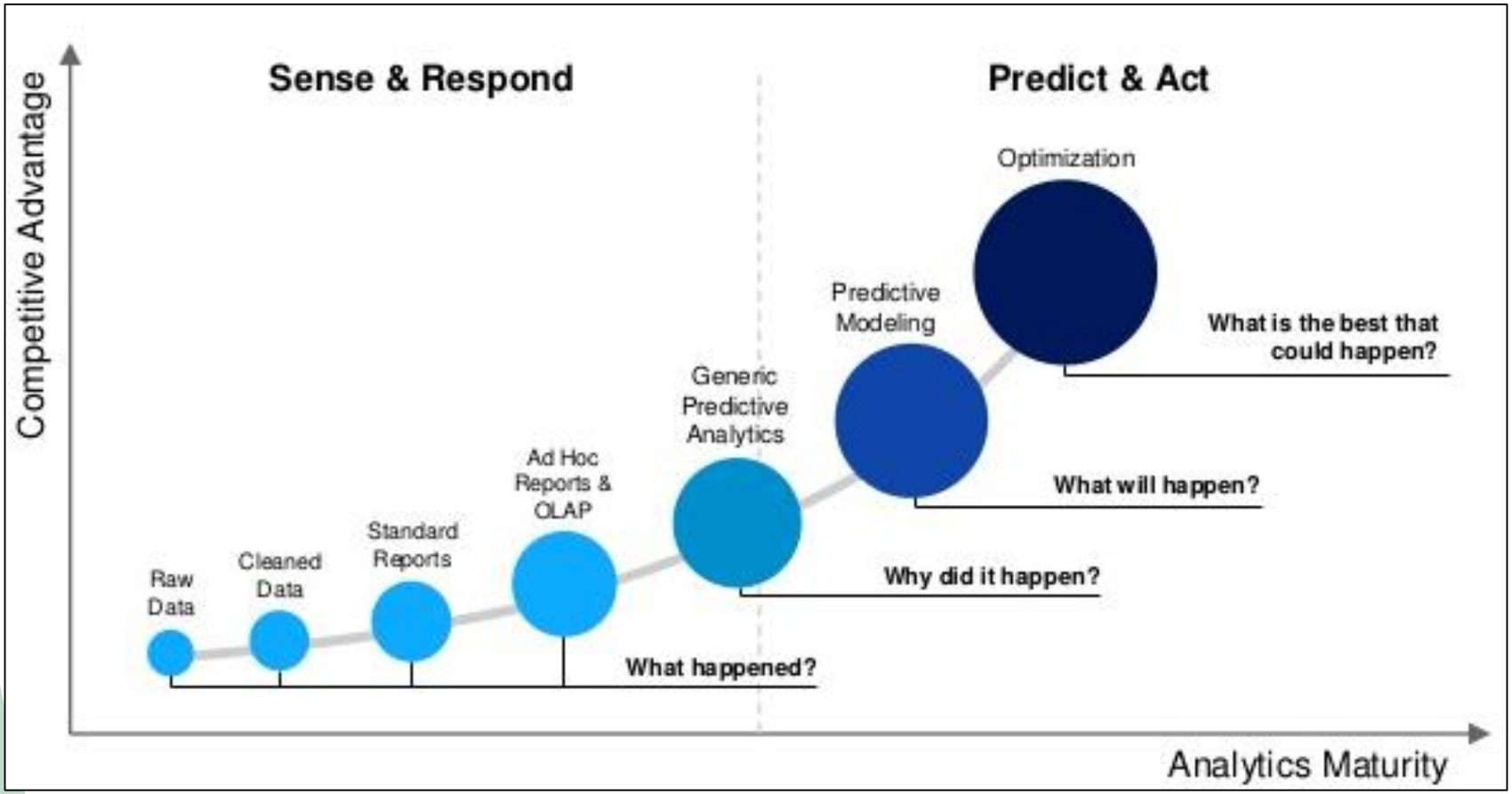


Figure 1.2: Types of BA According to Sophistication

A Review of BA

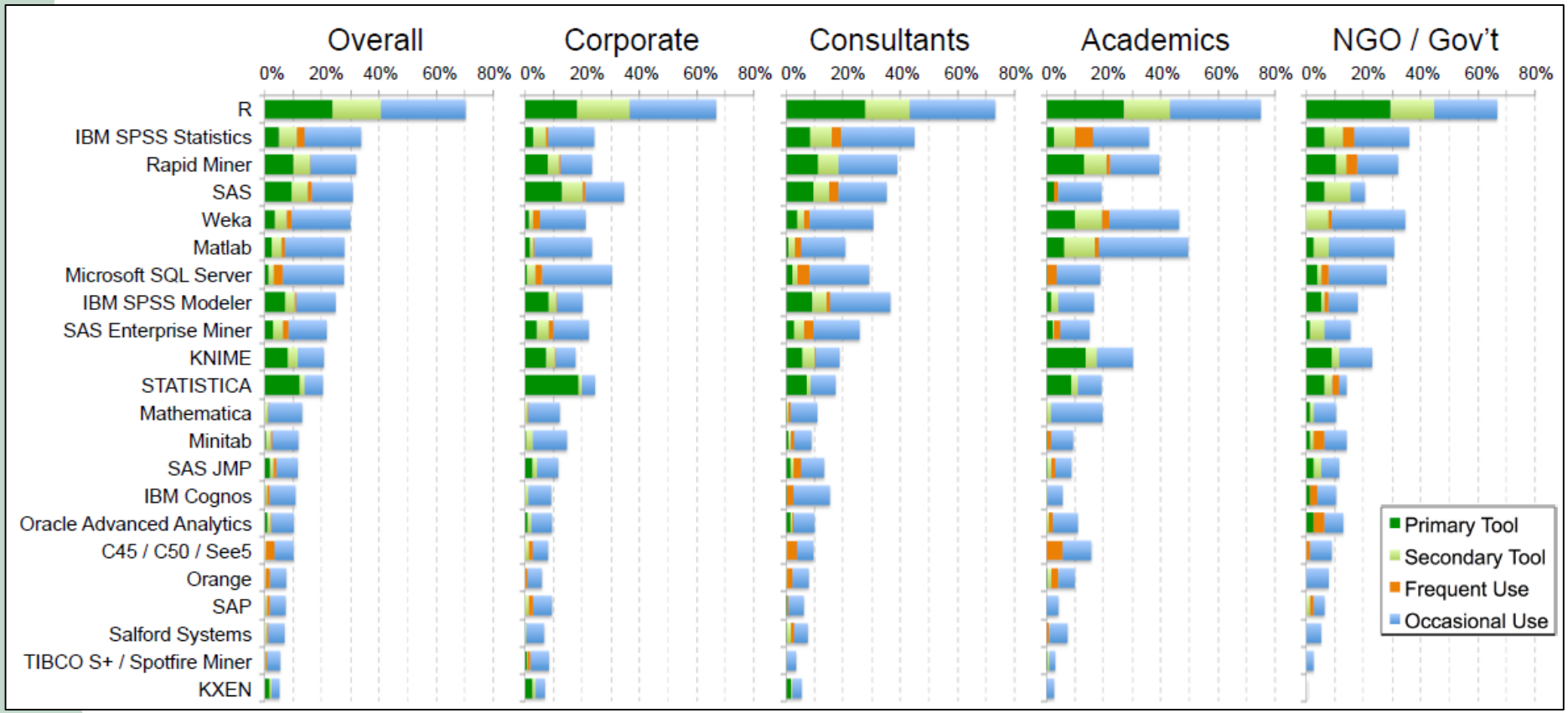


Figure 1.3: Tools of Business Analytics

http://www.rexeranalytics.com/Data-Miner-Survey-2013-Intro.html

A Review of BA

- KDNuggets Survey 2015: The top 10 tools **by share of users** were:
  - **R**, 46.9% share (38.5% in 2014)
  - **RapidMiner**, 31.5% (44.2% in 2014)
  - **SQL**, 30.9% (25.3% in 2014)
  - **Python**, 30.3% (19.5% in 2014)
  - **Excel**, 22.9% (25.8% in 2014)
  - **KNIME**, 20.0% (15.0% in 2014)
  - **Hadoop**, 18.4% (12.7% in 2014)
  - **Tableau**, 12.4% (9.1% in 2014)
  - **SAS**, 11.3 (10.9% in 2014)
  - **Spark**, 11.3% (2.6% in 2014)

http://www.kdnuggets.com/2015/05/poll-r-rapidminer-python-big-data-spark.html

Outline for This Session

- Introductions
- A Review of BI
- **What is R?**
- Where to Get R and R Studio
- First Commands

What is R?

Definition 1.1: R

- R is an integrated suite of software facilities for data manipulation, calculation and graphical display. Among other things it has:
  - 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 directly at the computer or on hardcopy
  - a well developed, simple and effective programming language (called 'S)

What is R?

- Many people use R as a statistics system.
- It is also an environment within which many classical and modern statistical techniques have been implemented.
- A few of these are built into the base R environment, but many are supplied as packages.
- There are about 25 packages supplied with R (called “standard” and “recommended” packages) and many more are available through the CRAN family of Internet sites

What is R?

- What does R Do?
  - Data handling and storage: numeric, textual
  - Matrix algebra
  - Hash tables and regular expressions
  - High-level data analytic and statistical functions
  - Programming language: loops, branching, subroutines
  - Graphics
  - Dashboards

What is R?

- What R doesn't do?
  - Is not a database, but connects to DBMSs
  - No spreadsheet view of data, but connects to Excel/MS Office
  - No professional / commercial support

What is R?

- Strengths
  - Free and Open Source
  - Strong User Community
  - Highly extensible, flexible
  - Implementation of high end statistical methods
  - Flexible graphics and intelligent defaults
- Weakness
  - Steep learning curve
  - Slow for very large datasets (>1Gb File)

What is R?

Definition 1.2: R Studio

- RStudio IDE is a powerful and productive 3rd Party user interface for R. It's free, open source, and works great on Windows, Mac, and Linux.

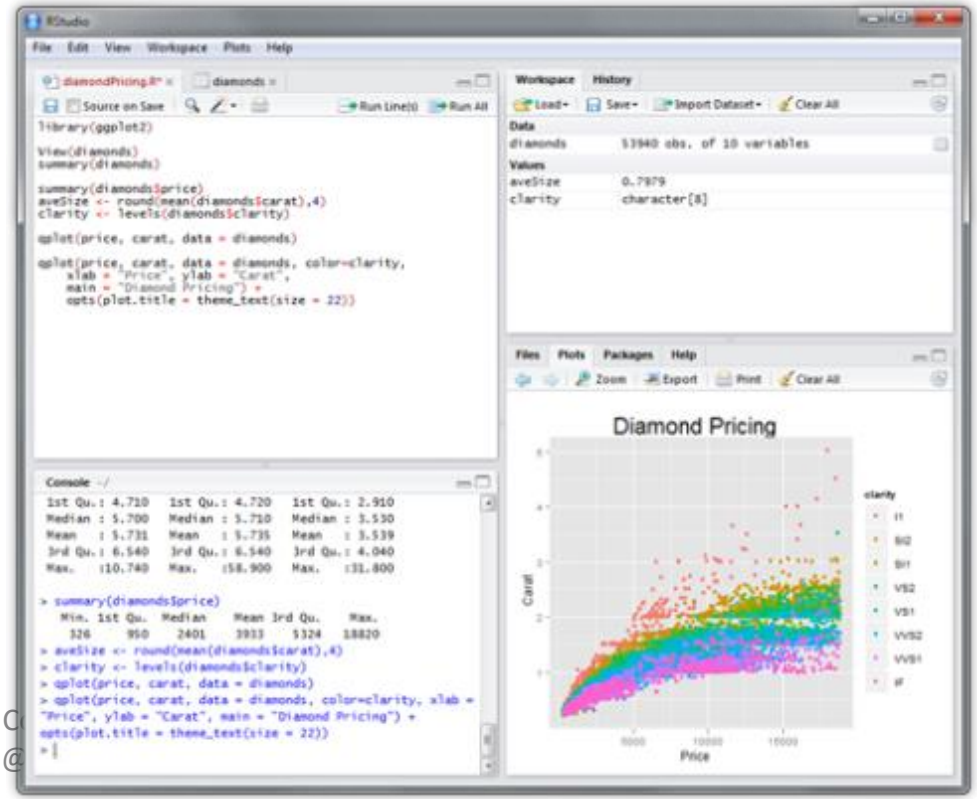


Figure 1.4: R Studio GUI

Outline for This Session

- Introductions
- A Review of BI
- What is R?
- **Where to Get R and R Studio**
- First Commands

Where to Get R and R Studio

- Where to Get R
  - Go to www.r-project.org
  - Downloads: CRAN
  - Set your Mirror: Anyone in the USA is fine.
  - Select Windows 95 or later.
  - Select base.
  - Select Latest R Version
  - Download Installer

Where to Get R and R Studio

- Where to Get R Studio
  - https://www.rstudio.com/products/rstudio/download/
  - RStudio 0.99.473 or latest version- Windows Vista/7/8, Mac OS X
  - Must install R **first** before R Studio

Outline for This Session

- Introductions
- A Review of BI
- What is R?
- Where to Get R and R Studio
- **First Commands**

First Commands

- R is an expression language with a very simple syntax.
- It is case sensitive, so "`A`" and "`a`" are different symbols and would refer to different variables.
- The set of symbols which can be used in R names depends on the operating system
- Normally all alphanumeric symbols are allowed plus '.' and '_', with the restriction that a name must start with '.' or a letter, and if it starts with '.' the second character must not be a digit.
- Names are effectively unlimited in length.

First Commands

- Click on **Start**
- Search for **R Studio** 
- Or
- Open R Studio from the Desktop 

- Click on File-> New File -> R Script

- You will be using this **R Script** for this exercise unless otherwise stated.
- You can always **save** the R Script using the Ctl+S Command

R Studio GUI

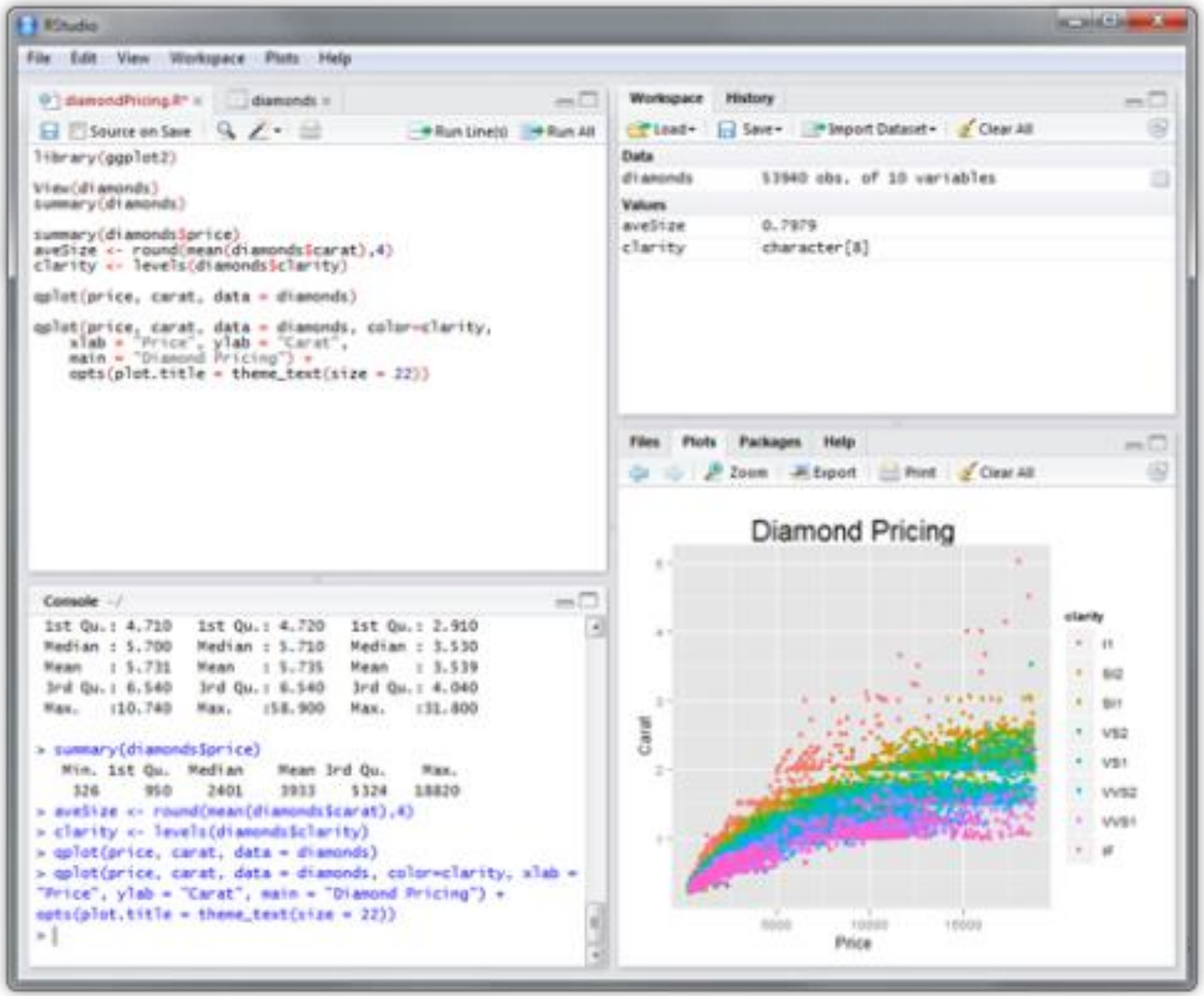


Figure 1.5: R Studio GUI

First Commands

- Panels:
  - Scripts
  - Environment/History
  - Console
  - Files/Plots/Packages/Help/Viewer

First Commands

Definition 1.3: Working Directory

- A Working Directory is where R **will look for files**, do manipulations and save results
- By Default, **My Documents** is the Default Working Directory when opening R and R Studio in Windows
- For this training, we shall create a separate sub-directory, called “work” to hold data files
- This will be the **working directory** whenever you use R for this training.

First Commands

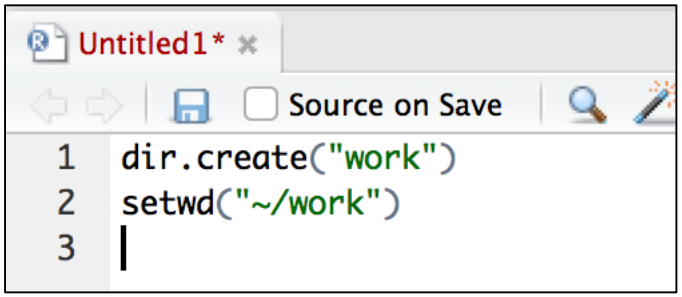
Example 1.1: Setting the Working Directory

- Create a folder named *work* in the My Documents folder and set it as a working directory
  - Type the **following code** in the new R Script Tab

➢ `dir.create("work")`

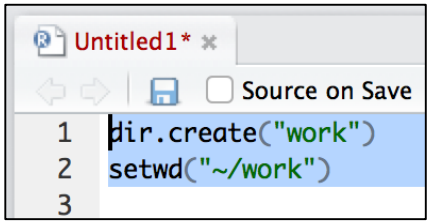
➢ `setwd("~/work")`

- R Studio should look like this:

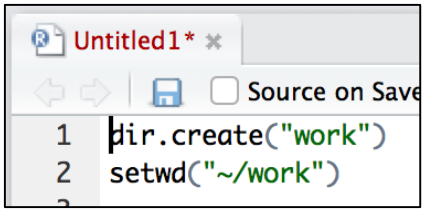


First Commands

- Two Ways of Executing Code
  - **Highlight and Run**
    - Highlight the two lines of code and click on Run 



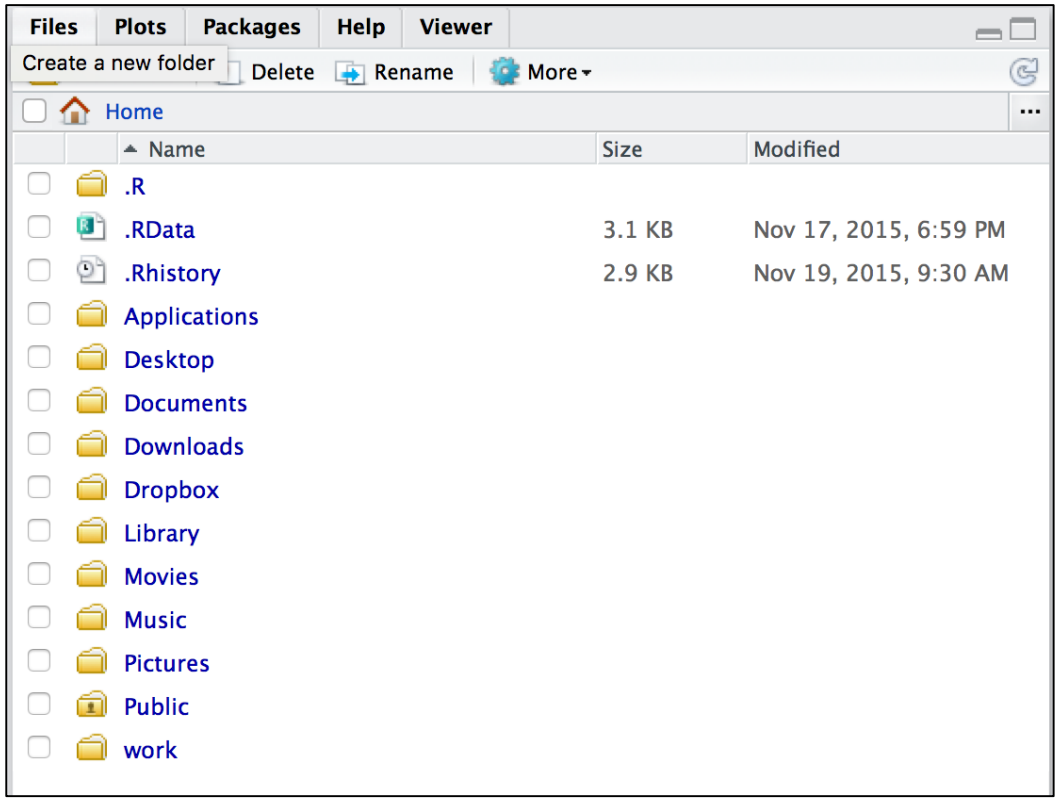
  - **Select Row and Run**
    - Click on the first row and then click on  one row at a time



    - Shortcut for  is Control + Enter

First Commands

- Changing Working Directory Using R Studio
  - In the Files Tab, Navigate to your desired Directory
  - Click on More -> Set as Working Directory



First Commands

- R has a **built-in help** facility to that helps the user know the command, its input requirements, what it does and what does it produce
- Invoked using the `help` or ? Command.

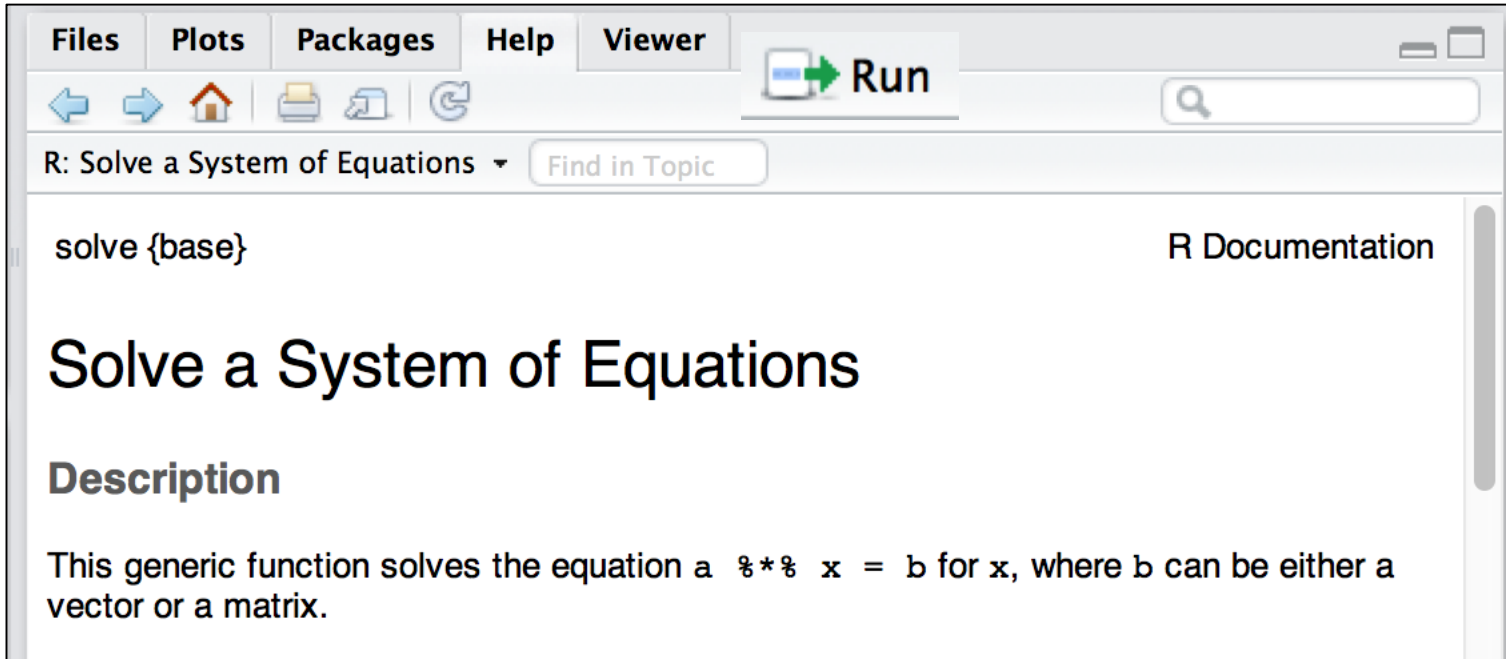
First Commands

Example 1.2: Help

- What does the `solve` function do?

➢ `help(solve)`

➢ `?solve`



First Commands

- Elementary commands consist of either **expressions or assignments**.
- If an expression is given as a command, it is evaluated, **printed** (unless specifically made invisible), and **the value is lost.**
- An assignment also evaluates an expression and passes the value to a variable but the result is **not automatically printed**.
- Commands are separated either by a semi-colon (‘;’), or by a newline.
- Elementary commands can be **grouped together** into one compound expression by braces (‘{’ and ‘}’).

First Commands

- Comments can be put almost anywhere, starting with a **hashmark** (‘#’), everything to the end of the line is a comment.
- If a command is not complete at the end of a line, R will give a different prompt, by default + on second and subsequent lines and continue to read input **until the command is syntactically complete.**

First Commands

Example 1.3: Elementary Commands

- How to write comments, assignments and expressions

```
#this is a comment not a hashtag
#expression
2+3
#assignment
x =2 +3
#print assignment
x
```

```
> #this is a comment not a hashtag
> #expression
> 2+3
[1] 5
> #assignment
> x =2 +3
> #print assignment
> x
[1] 5
```

E.R. L. Jalao, Copyright UP NEC,
eljalao@up.edu.ph

First Commands

Example 1.3 (Cont.): Elementary Commands

- How to group commands

```
#this is a group of commands
{
   y=2+3
   z=2+4
}
#this is an incomplete command
x = 1 +
2
```

```
> #this is a group of commands
> {
+ y=2+3
+ z=2+4
+ }
> #this is an incomplete command
> x = 1 +
+ 2
>
```


E.R. L. Jalao, Copyright UP NEC,
eljalao@up.edu.ph

First Commands

Definition 1.4: Packages

- Packages are prewritten functions that augment the basic functionalities of R
- Currently, the CRAN (Comprehensive R Archive Network) package repository features 6825 available packages.
- To install a package (Must be connected to the internet):
  - `install.packages("nameofpackage")`
- To load a package into R
  - `library("nameofpackage")`

First Commands

Example 1.4: Installing Packages

- Install the reshape2 package

➢ `install.packages("reshape2")`

➢ `library("reshape2")`

Outline for This Session

- Introductions
- A Review of BI
- What is R?
- Where to Get R and R Studio
- First Commands