

R Programming Tutorial

Zhe (Joe) Deng
Fall 2018 MIS 2502
Nov. 6th , 2018

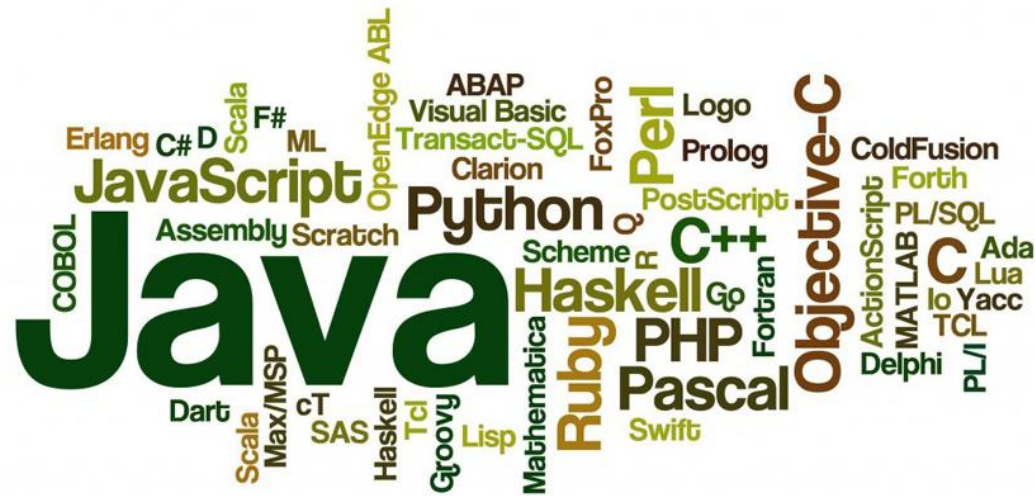
Reference:
<http://stanford.edu/~jgrimmer/RDataManagement.pdf>



Agenda

1. Basic Programming Concept and Coding Norm
 - Applied to C, Visual Basic, HTML, SQL, JAVA, C#, R, STATA, Python...
2. Basics of R
 - Data Type
 - Input

Part 1/2



Basic
Programming
Concept
and Coding Norm

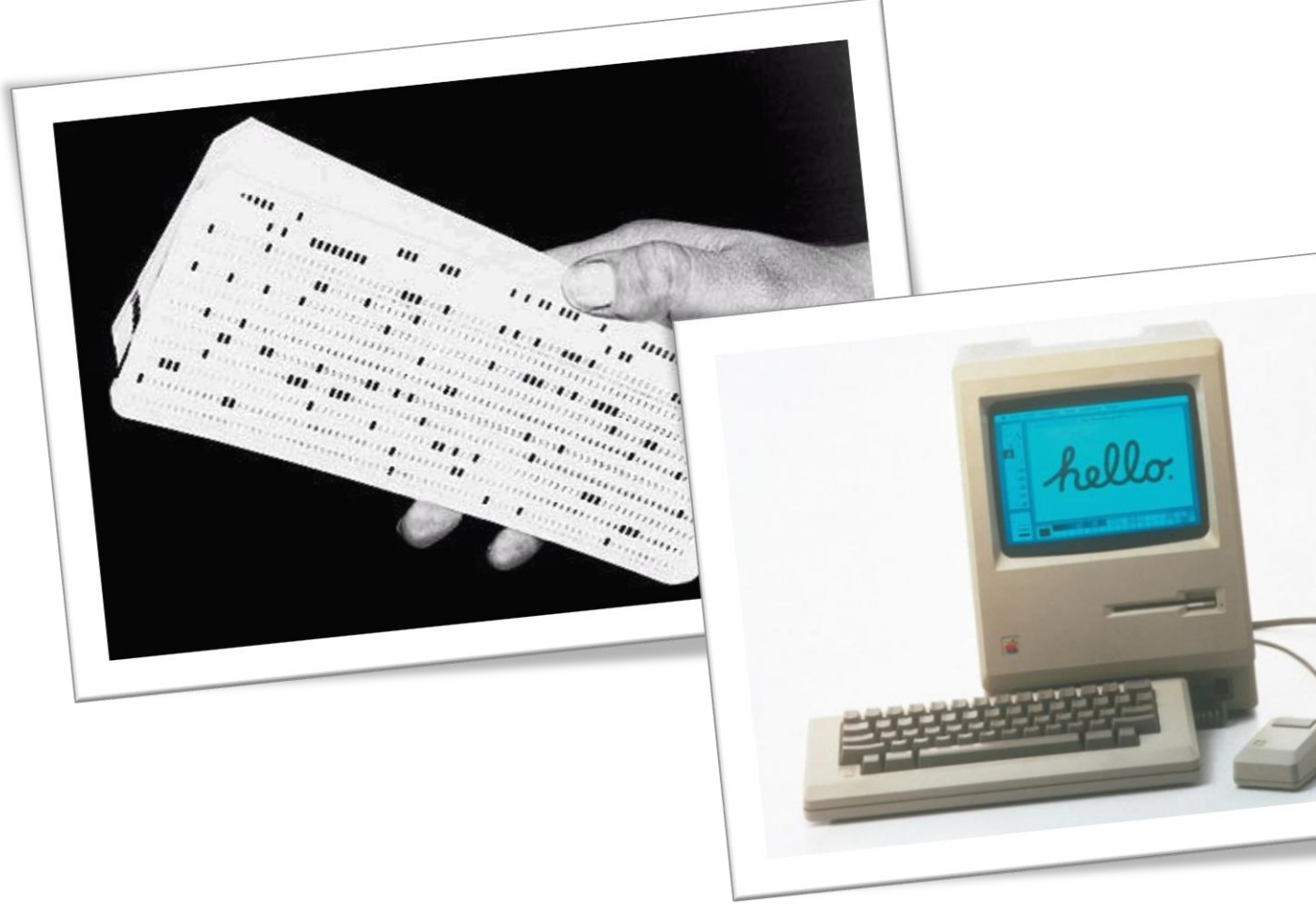
Basic Programming Concept & Coding Norm

- Programming Language Levels
- Execution Modes
- Variable & Value, Function & Argument
- Coding Norm/Convention

Basic Programming Concept & Coding Norm

- Programming Language Levels
- Execution Modes
- Variable & Value, Function & Argument
- Coding Norm/Convention

Basic Programming Concept



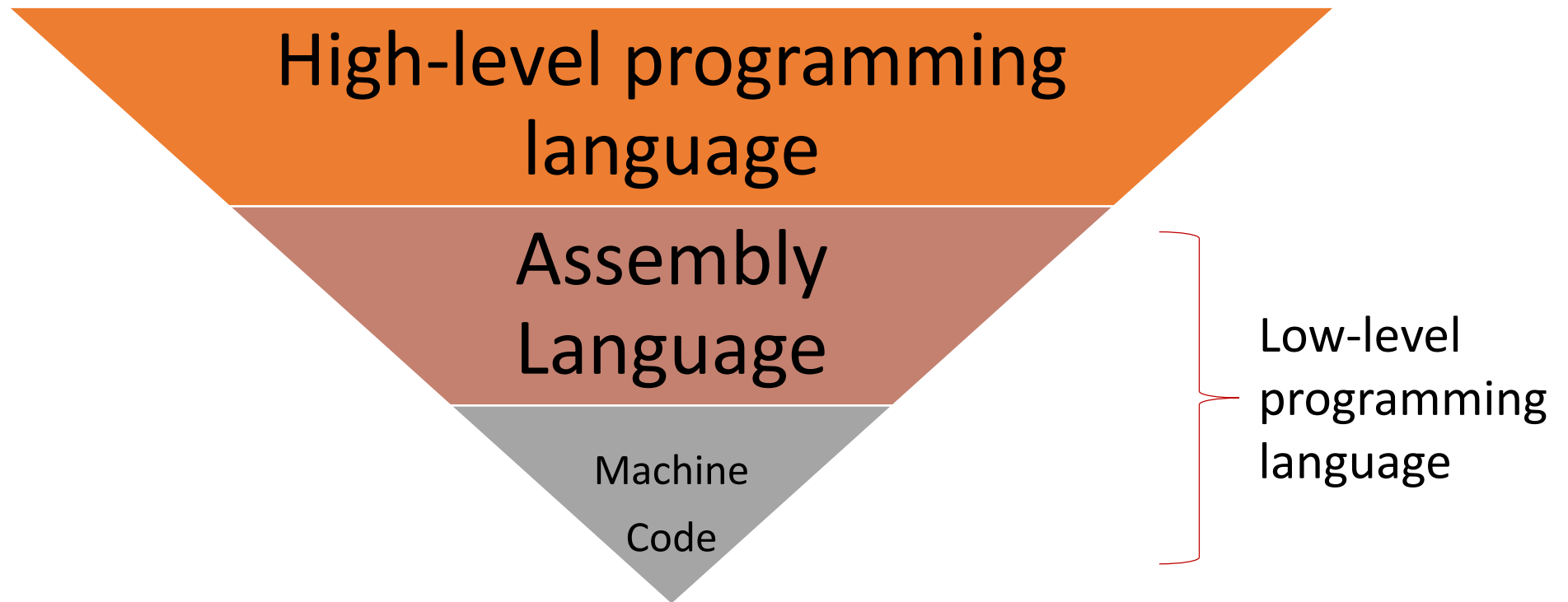
Owl's Nest

Temple's new High-Performance and Scientific Computing Cluster



Programming language levels

- Programming language levels



Programming language levels

- Low-level programming language

Machine Code

```
1 FOX 12:01a 23- 1
A 002000 C2 30 REP #$30
A 002002 18 CLC
A 002003 F8 SED
A 002004 A9 34 12 LDA #$1234
A 002007 69 21 43 ADC #$4321
A 00200A 8F 03 7F 01 STA $017F03
A 00200E D8 CLD
A 00200F E2 30 SEP #$30
A 002011 00 BRK
A 2012

r
PB PC NUmxDIZC .A .X .Y SP DP DB
; 00 E012 00110000 0000 0000 0002 CFFF 0000 00
g 2000

BREAK

PB PC NUmxDIZC .A .X .Y SP DP DB
; 00 2013 00110000 5555 0000 0002 CFFF 0000 00
m 7f03 7f03
>007f03 55 55 00 00 00 00 00 00 00 00 00 00 00 00 00:UU.....
```

Assembly Language

```
MONITOR FOR 6802 1.4          9-14-80 TSC ASSEMBLER PAGE 2

C000                                ORG    ROM+$0000 BEGIN MONITOR
C000 8E 00 70  START  LDS    #STACK

*****
* FUNCTION: INITA - Initialize ACIA
* INPUT: none
* OUTPUT: none
* CALLS: none
* DESTROYS: acc A

0013          RESETA EQU    %00010011
0011          CTLREG EQU    %00010001

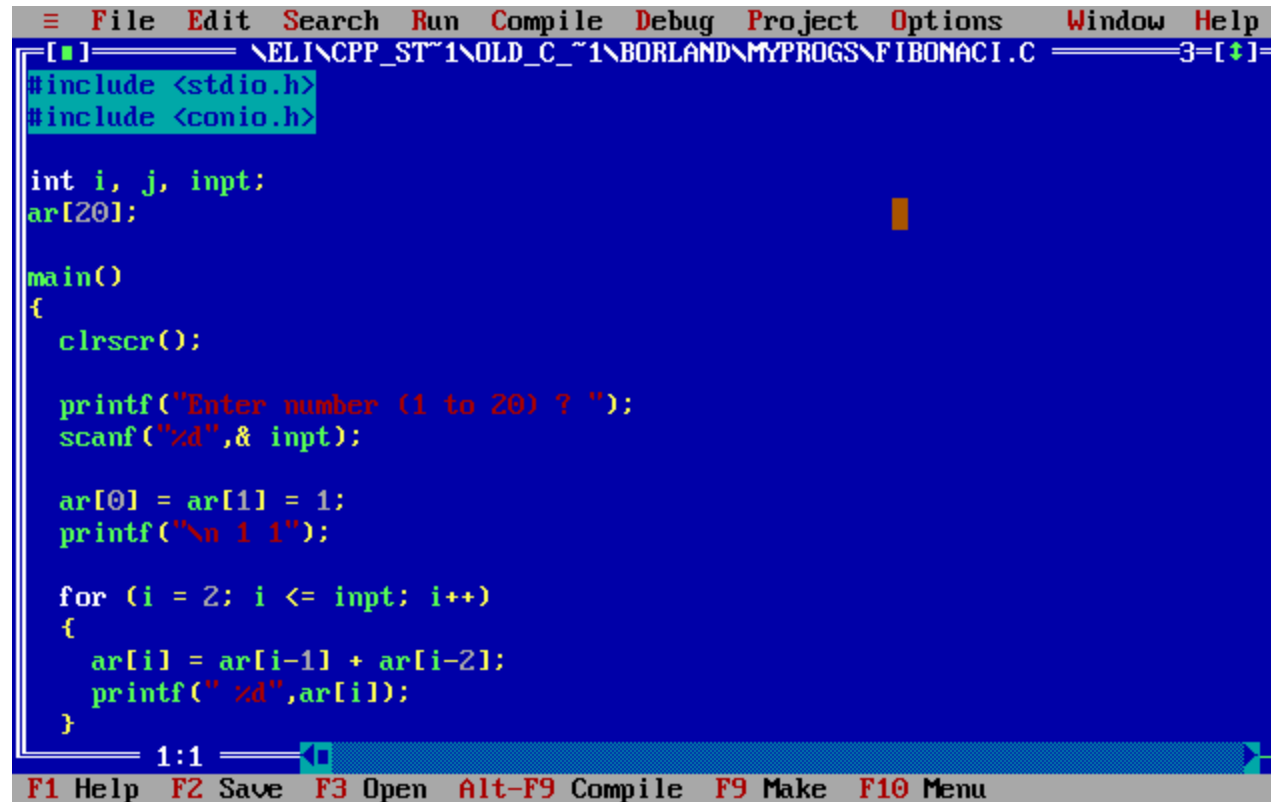
C003 86 13          INITA  LDA A  #RESETA  RESET ACIA
C005 B7 80 04          STA A  ACIA
C008 86 11          LDA A  #CTLREG  SET 8 BITS AND 2 STOP
C00A B7 80 04          STA A  ACIA

C00D 7E C0 F1          JMP    SIGNON  GO TO START OF MONITOR

*****
* FUNCTION: INCH - Input character
* INPUT: none
* OUTPUT: char in acc A
* DESTROYS: acc A
* CALLS: none
* DESCRIPTION: Gets 1 character from terminal
```


Programming language levels

- High-level programming language (e.g., C)



```
File Edit Search Run Compile Debug Project Options Window Help
\EL\CPP_ST~1\OLD_C_~1\BORLAND\MYPROGS\FIBONACI.C 3=[+]
```

```
#include <stdio.h>
#include <conio.h>

int i, j, inpt;
ar[20];

main()
{
    clrscr();

    printf("Enter number (1 to 20) ? ");
    scanf("%d",& inpt);

    ar[0] = ar[1] = 1;
    printf("\n 1 1");

    for (i = 2; i <= inpt; i++)
    {
        ar[i] = ar[i-1] + ar[i-2];
        printf(" %d",ar[i]);
    }
}
```

```
1:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
```

Basic Programming Concept & Coding Norm

- Programming Language Levels
- **Execution Modes**
- Variable & Value, Function & Argument
- Coding Norm/Convention

Execution Modes

- Three general modes of execution for modern high-level languages
 - Interpreted:
Source code => Executed directly (e.g., R)
 - Compiled:
Source code => Machine code/intermediate representation => Executed (e.g., Java, C#)
 - Source-to-Source Translated or Trans-compiled
Source code \approx compilers are already widely available in translation (e.g., C)
- Source code
 - A collection of code, possibly with comments, written using a human-readable programming language, usually as plain text. (e.g., R script)
- Testing and debugging

IDE

- An **integrated development environment (IDE)** is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a source code editor, build automation tools, and a [debugger](#).
- Some IDEs contain a [compiler](#), [interpreter](#), or both.
- E.g., R Studio, MS Visual Studio, Eclipse, NetBeans, PyCharm, Android Studio...

Basic Programming Concept & Coding Norm

- Programming Language Levels
- Execution Modes
- **Variable & Value, Function & Argument**
- Coding Norm/Convention

Variable & Value, Function & Argument

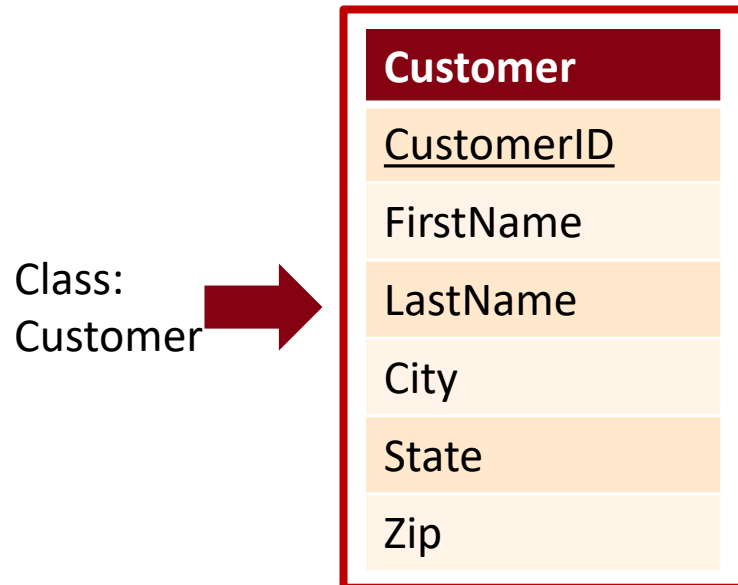
- Variable & Value

```
14 ##### The Basics: Variables #####
15
16 x=5      # <- and = do the same thing
17 y<-10
18 z=8
19 x+y-z
20 name<-"David"
21 rm(x)    # rm() removes the variable from memory
```

Read from the right to left as "Assign [value] 5 to [variable] x".
IDE first requires OS to allocate a segment of machine memory to store a empty variable template called "x", then requires OS to allocate another segment of memory to fill a copy of the template with a value, 5.

Variable & Value, Function & Argument

- OOP (Object Oriented Programming)
 - Class => Blank data template (data structure)
 - Object => A copy of the data template with filled data
 - Instantiation => The process of filling data into a copy of blank data template
 - SQL example:



Object: Lisa Cuddy

The diagram shows an object instance for 'Lisa Cuddy'. It is represented as a table with a red-bordered box around the row for Lisa Cuddy. The table has columns: CustomerID, FirstName, LastName, City, State, and Zip. The rows are: 1001, Greg, House, Princeton, NJ, 09120; 1002, Lisa, Cuddy, Plainsboro, NJ, 09123; 1003, James, Wilson, Pittsgrove, NJ, 09121; 1004, Eric, Foreman, Warminster, PA, 19111. A red arrow points from the text 'Object: Lisa Cuddy' to the table.

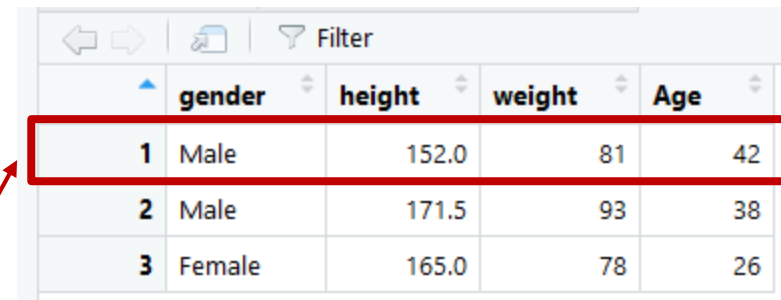
CustomerID	FirstName	LastName	City	State	Zip
1001	Greg	House	Princeton	NJ	09120
1002	Lisa	Cuddy	Plainsboro	NJ	09123
1003	James	Wilson	Pittsgrove	NJ	09121
1004	Eric	Foreman	Warminster	PA	19111

Variable & Value, Function & Argument

- OOP (Object Oriented Programming)
 - Class => Blank data template (data structure)
 - Object => A copy of the data template with filled data
 - Instantiation => The process of filling data into a copy of blank data template

• R example:

```
53 ##### data frames #####
54 BMI<-data.frame(
55   gender = c("Male","Male","Female"),
56   height = c(152,171.5,165),
57   weight = c(81,93,78),
58   Age = c(42,38,26)
59 )
```



	gender	height	weight	Age
1	Male	152.0	81	42
2	Male	171.5	93	38
3	Female	165.0	78	26

- Define a data frame type (Let's take it as a table for now) class called "BMI" with attributes – gender, height, weight, Age.
- Instantiate multiple objects (i.e., "Male, 152.0, 81, 42")
- c("Male","Male","Female")? A "sub-" class

Variable & Value, Function & Argument

- OOP (Object Oriented Programming)
 - A class can be a compound one to have multiple subclasses/attributes as shown on last slide.
 - How to get a subclass/attribute?
 - className.subclassName
 - className.attributeName
 - Remember how we get a table in a database? – SalesDB.customer

Variable & Value, Function & Argument

- Function & Argument

```
14 ##### The Basics: Variables #####
15
16 x=5      # <- and = do the same thing
17 y<-10
18 z=8
19 x+y-z
20 name<-"David"
21 rm(x)    # rm() removes the variable from memory
--
```

Function: `rm(Argument)`. `rm()` here is a build-in function. You can also define your own function. Some function are used to return a value, such as `AVG()` in SQL. The others are used to complete an operation, such as this. A function can take no argument, a single argument, or multiple arguments.

Variable & Value, Function & Argument

- Function & Argument

- Besides build-in functions, there are also plenty of third-part developed functions (packages), such as “psych”
- R also offers functions to install third-part packages
 - “psych” here is an argument required by function “install.packages”
 - Ignore the if(condition) {Statements} for now, we will discuss this in the second part of this tutorial.

```
74 if (!require("psych")) { install.packages("psych") }  
75 require("psych") } # In this class, we will use this statement to install and load a package
```

Basic Programming Concept & Coding Norm

- Programming Language Levels
- Execution Modes
- Variable & Value, Function & Argument
- **Coding Norm/Convention**

Coding Norm/Convention

- Block & Indentation
- Variable Naming
- Comment
- Online Resources

*To achieve source code
high reusability, maintainability, reliability, and complexity*

Coding Norm/Convention

- Block & Indentation

```
SELECT film.title, film.length
FROM moviedb.film, moviedb.`language`
WHERE film.language_id = `language`.language_id
AND `language`.`name`='English'
AND rating='G' AND film.length=(SELECT MIN(film.length)
FROM moviedb.film, moviedb.`language`
WHERE film.language_id = `language`.language_id
AND `language`.`name`='English'
AND rating='G');
```


Coding Norm/Convention

- Block & Indentation

```
SELECT film.title, film.length
FROM moviedb.film, moviedb.`language`
WHERE film.language_id = `language`.language_id
AND `language`.`name`='English'
AND rating='G'
AND film.length=(
    SELECT MIN(film.length)
    FROM moviedb.film, moviedb.`language`
    WHERE film.language_id = `language`.language_id
    AND `language`.`name`='English'
    AND rating='G'
);
```

← Operation: get information out of database

Sub-select Operation



← Conditions

Sub Select Conditions



Coding Norm/Convention

- Variable Naming
 - Meaning (e.g., sqrt, abs...)
 - Short
 - Conventions for different languages
 - [https://en.wikipedia.org/wiki/Naming_convention_\(programming\)](https://en.wikipedia.org/wiki/Naming_convention_(programming))

Coding Norm/Convention

- Comment

```
1 #This is a comment line
2
3 ##### The Basics: Calculations #####
4
5 12+23
6 15/2
7 2^4
8 exp(2)
9 sqrt(100)
10 log(10)
11 abs(-4)
12
```

```
1 # DESCRIPTIVES EXAMPLE
2 # Adapted from David Schuff, MIS Department, Fox School of Business, Temple University
3 # Really, the stats is all stuff you know already.
4 # This is just a demonstration of what you can do using R, and to get you used to the wacky syntax.
5 # In most cases, you'll just modify parameters in these scripts, but sometimes you'll need to change
6 # the statements themselves to get it to work with your own datasets.
7
8 # First, clear all previous stuff out of the workspace...
9 rm(list = ls())
```

Coding Norm/Convention

- Comment

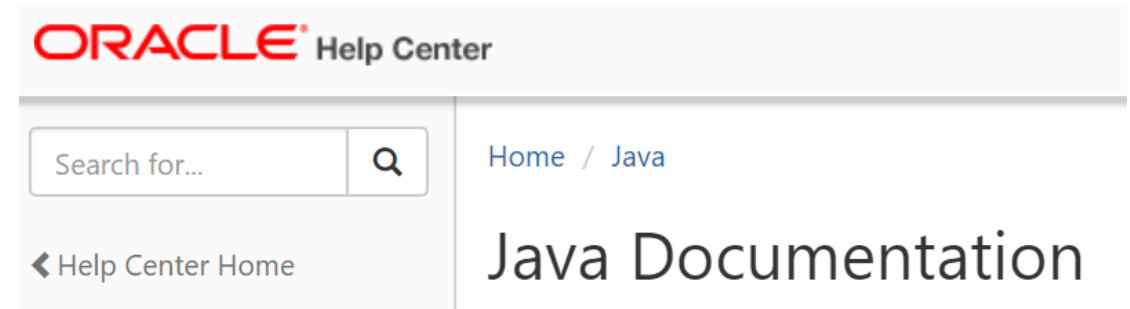
```
61 BMI[1,3]      # returns the cell value in the first row, second column
62 BMI[1,]      # returns everything in the first row
63 BMI[,3]      # returns everything in the third row
64 BMI[[2]]     # returns the second column as a vector
65 BMI$height   # returns the "height" column as a vector
66 mean(BMI$height) # returns the mean of "height" column
```

Coding Norm/Convention

- Online Resources

w3schools.com

 stackoverflow



R Tutorial

An R Introduction to Statistics

<http://www.r-tutor.com/>

Cutting corners to meet arbitrary management deadlines



Essential

Copying and Pasting from Stack Overflow

O'REILLY®

The Practical Developer
@ThePracticalDev

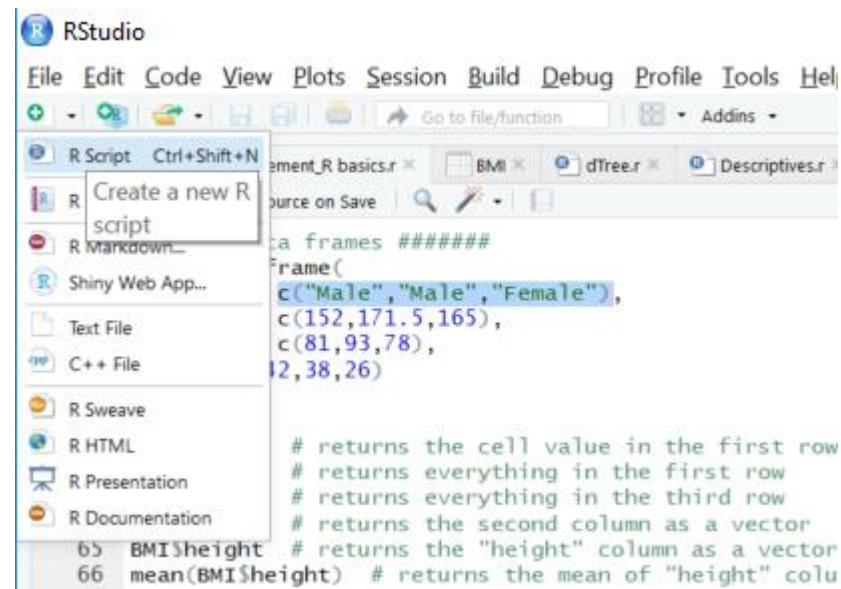
Part 2/2



Basics of R

Setting Up with R Studio

- For R and R Studio installation, please refer to [Advanced Data Analytics and R](#)
- Use source code editor in an IDE:



The screenshot shows the RStudio interface. The 'File' menu is open, displaying options such as 'Create a new R script', 'R Markdown...', 'Shiny Web App...', 'Text File', 'C++ File', 'R Sweave', 'R HTML', 'R Presentation', and 'R Documentation'. The background shows an R script editor with the following code:

```
ca frames #####  
frame(  
  c("Male", "Male", "Female"),  
  c(152, 171.5, 165),  
  c(81, 93, 78),  
  c(12, 38, 26)  
)  
  
# returns the cell value in the first row  
# returns everything in the first row  
# returns everything in the third row  
# returns the second column as a vector  
# returns the "height" column as a vector  
65 BMI$height  
66 mean(BMI$height) # returns the mean of "height" colu
```

R as calculator

```
> 5 + 4  
[1] 9  
> 8 * 2 - sqrt(9)  
[1] 13  
> log(4)/9^2  
[1] 0.01711|
```

Vectors

All objects consist of one or more vectors.

Vector: a combination of elements (i.e. numbers, words), usually created using `c()`, `seq()`, or `rep()`

```
53 ▾ ##### data frames #####  
54 BMI<-data.frame(  
55   gender = c("Male", "Male", "Female"),  
56   height = c(152, 171.5, 165),  
57   weight = c(81, 93, 78),  
58   Age = c(42, 38, 26)  
59 )
```

Vectors

```
> empty.vector <- c()
> empty.vector
NULL
> one.to.five <- c(1, 2, 3, 4, 5)
> one.to.five
[1] 1 2 3 4 5
> poli.sci <- c("theory", "amer.", "comp.", "ir")
> poli.sci
[1] "theory" "amer."  "comp."  "ir"
```

Data Types

- Numeric
- Character
- Logical

```
53 ▾ ##### data frames #####  
54 BMI<-data.frame(  
55   gender = c("Male","Male","Female"),  
56   height = c(152,171.5,165),  
57   weight = c(81,93,78),  
58   Age = c(42,38,26)  
59 )
```

Working Directory

- When loading or saving a dataset or object, R will look in the current working directory.
- If your working directory is not where the file is at, R will not find it, so make sure you change the working directory.

Working Directory

The screenshot displays the RStudio interface. The main editor window contains R code for creating a data frame, installing a package, and performing statistical operations. The Environment pane on the right shows the current data frame with its variables. The File Explorer pane at the bottom right shows the file system structure, with a red box highlighting the Desktop directory.

```
##### data frames #####
54 BMI<-data.frame(
55   gender = c("Male","Male","Female"),
56   height = c(152,171,5,165),
57   weight = c(81,93,78),
58   Age = c(42,38,26)
59 )
60 BMI
61 BMI[1,3] # returns the cell value in the first row, second column
62 BMI[1,] # returns everything in the first row
63 BMI[,3] # returns everything in the third row
64 BMI[[2]] # returns the second column as a vector
65 BMI$height # returns the "height" column as a vector
66 mean(BMI$height) # returns the mean of "height" column
67
68 ##### Packages #####
69 describe(scores) # In some cases, we have to install a package to use a function
70 install.packages("psych") # To install a package
71 require(psych) # To load a package; can also use library(psych)
72 describe(scores) # After loading the package, you can use functions included in the package
73
74 + if (!require("psych")) { install.packages("psych")
75   require("psych") } # In this class, we will use this statement to install and load a package
76
77 ##### Import data into R #####
78 # Hint: 1) Save the file to same folder where you saved the source code
79 # 2) Click the menu Session->Set Working Directory To Source File Location
80 teamData <- read.csv("2009baseballteamStats.csv")
81
82 summary(teamData) # Summary statistics
83 summary(teamData$runs) # Return summary statistics of "runs" column
84 describe(teamData$runs) # Descriptive statistics
85 describeBy(teamData$runs, teamData$league) # Descriptive statistics by League
86 t.test(teamData$runs~teamData$league) # t-test for differences in average runs by League
87
88
89 ##### Visualization in R #####
90 hist(teamData$battingAvg, # Draw histogram
91   xlab="Batting Average",
92   main="Batting Average Distribution")
93
##### data frames #####
537
```

Environment

Variable	Value
gender	Factor w/ 2 levels "Female","Male": 2 2 1
height	num 152 172 165
weight	num 81 93 78
Age	num 42 38 26

Files

Name	Size	Modified
08-2016 Mobile Distraction in Class.lnk	1.2 KB	Oct 30, 2016, 12:55 PM
5 Fall 2016.lnk	1.1 KB	Aug 30, 2016, 2:04 PM
CHITA 2016.pdf	392.1 KB	Oct 16, 2016, 2:13 PM
CIST 2016.pdf	668.2 KB	Oct 16, 2016, 2:14 PM
CCOE 2016.pdf	1.1 MB	Oct 16, 2016, 2:15 PM
Copy of midterm_2016_v3.xlsx	24 KB	Nov 2, 2016, 12:49 PM
episode_level_labels_7.27.docx	408.2 KB	Aug 2, 2016, 12:55 PM
Fall 2016 TA MIS 2502.lnk	1.2 KB	Oct 26, 2016, 9:48 AM
gaphi-tutorials-master		
Local Disk (D:).lnk	515 B	Apr 1, 2016, 8:02 AM
midterm_2016.xlsx	22.3 KB	Nov 2, 2016, 7:17 AM
Mobile Device Survey_Eln.docx	19.9 KB	Aug 28, 2016, 8:50 AM
Mobile Technology Policy_CIST2016.pptx	4.1 MB	Nov 6, 2016, 1:27 PM
Physician Network_Zhe Seegin-Kumar.lnk	1.1 KB	Apr 7, 2016, 5:06 AM
physician_pair1.jpg	1.3 KB	May 6, 2016, 8:33 PM
physician_pair3.jpg	1.1 KB	May 15, 2016, 3:29 PM
psycharm-community-2016.2.1.exe	187.2 MB	Aug 22, 2016, 9:25 PM
sketch.docx - Shortcut.lnk	988 B	Nov 2, 2016, 9:45 AM
Supplement_R_basics.r	2.9 KB	Nov 6, 2016, 2:34 PM
Survey Results.docx	55 KB	Aug 28, 2016, 8:50 AM
SW		
Table 9. Time Spent on Distraction within and between Experimental Conditions...	40.1 KB	Nov 2, 2016, 8:12 PM
WISE2016_Acceptance.pdf	268.1 KB	Oct 17, 2016, 8:13 AM
Zhe_Copy of midterm_2016.xlsx	24 KB	Nov 2, 2016, 12:49 PM
Descriptives.r	5.7 KB	Nov 6, 2016, 2:35 PM
RDataManagement.pdf	137.6 KB	Nov 7, 2016, 9:29 AM

Working Directory

- To change to the foo working directory, use `setwd("D:\MIS 2502")`
- To see the current working directory, type `getwd()`

Loading Datasets

Suppose you want to load the foo dataset.

If the dataset is in

- an existing R package, load the package and type `data(foo)`
- .RData format, type `load(foo)`
- .txt or other text formats, type `read.table("foo.txt")`
- .csv format, type `read.csv("foo.txt")`
- .dta (Stata) format, load the foreign library and type `read.dta("foo.dta")`
- Remember “function & argument” in the first part

To save objects into these formats, use the equivalent `write.table()`, `write.csv()`, etc. commands.



Assignment #5 Q&A





Questions, comments, feedback...

Zhe Deng

deng@temple.edu

Fox School of Business, Temple University