UiT The Arctic University of Norway

R for beginners

20-02-2023

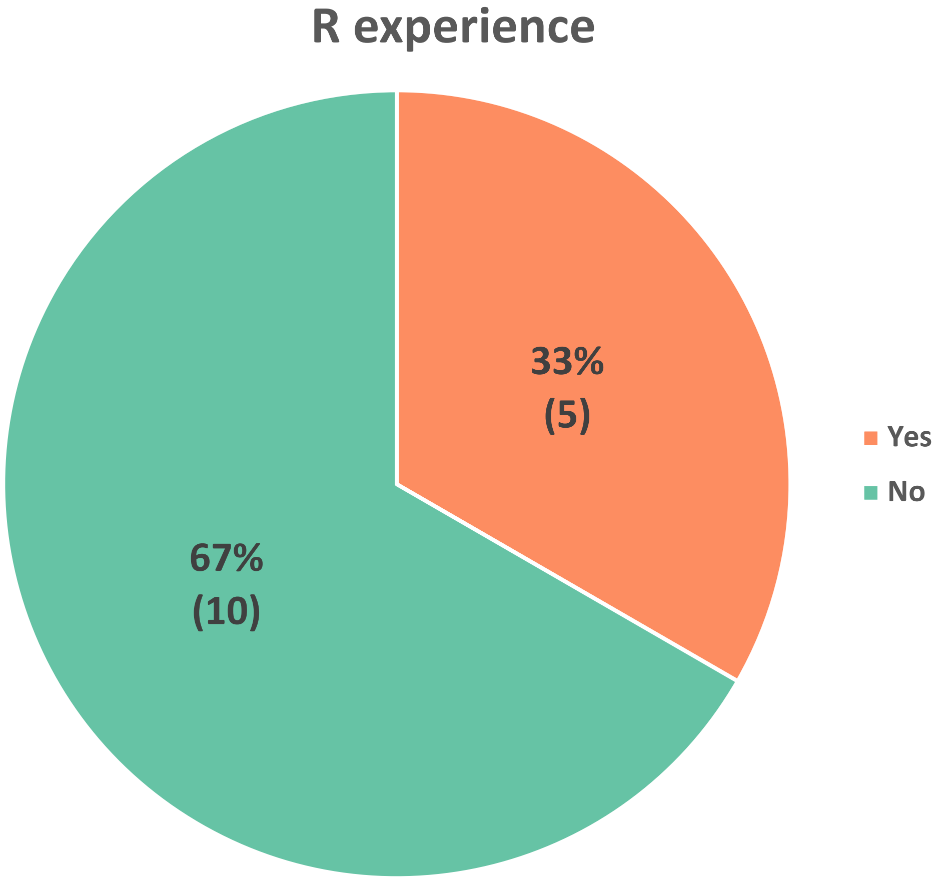
Ahmed Bargheet

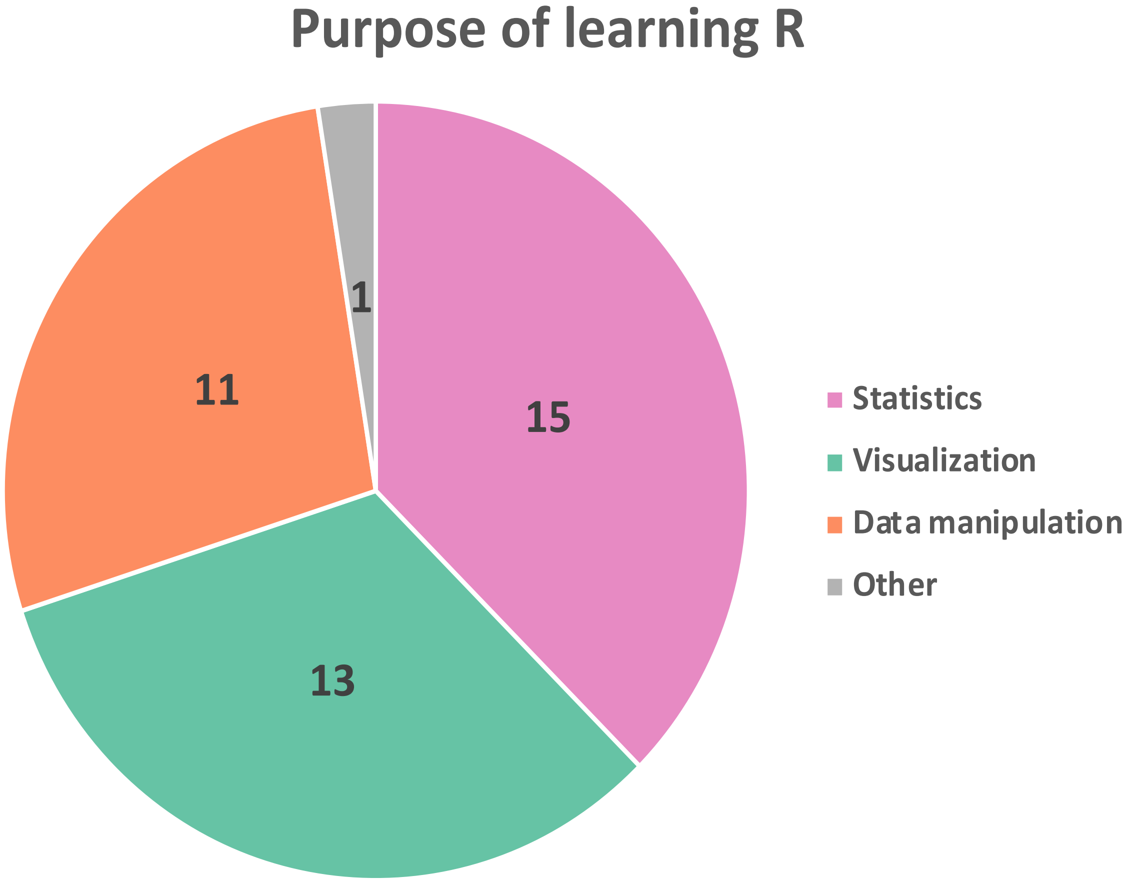
ahmed.bargheet@uit.no

Dorota Julia Buczek

dorota.j.buczek@uit.no

Before we start


R experience


Purpose of learning R

Before we start

- Please do not be shy!
- We are friends
- No stupid questions ☺
- Please be interactive!
- One of you will be chosen to help me or to answer questions during sessions

What is R

- Most widely used **statistics** programming language.

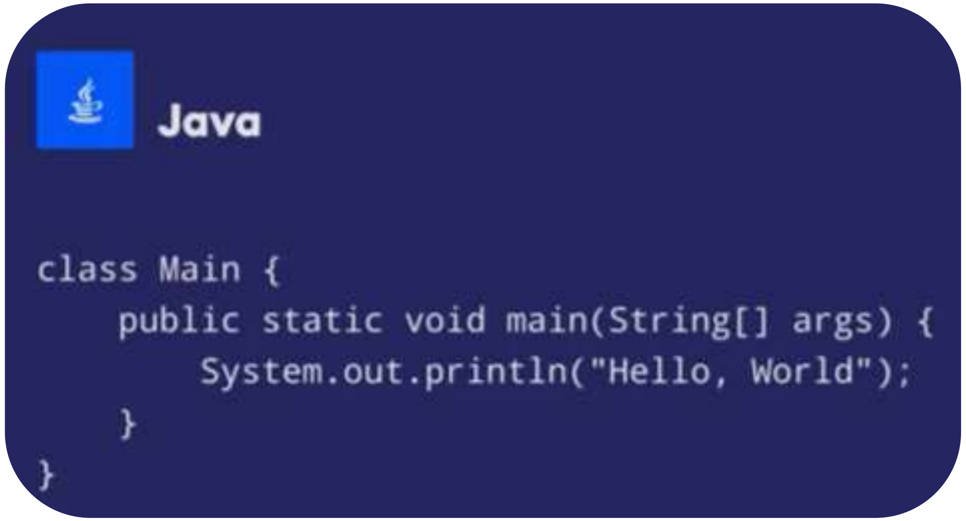
What is R

- Most widely used **statistics** programming language.
- It is the **#1 choice** of **data scientists** and **analysts**.

What is R

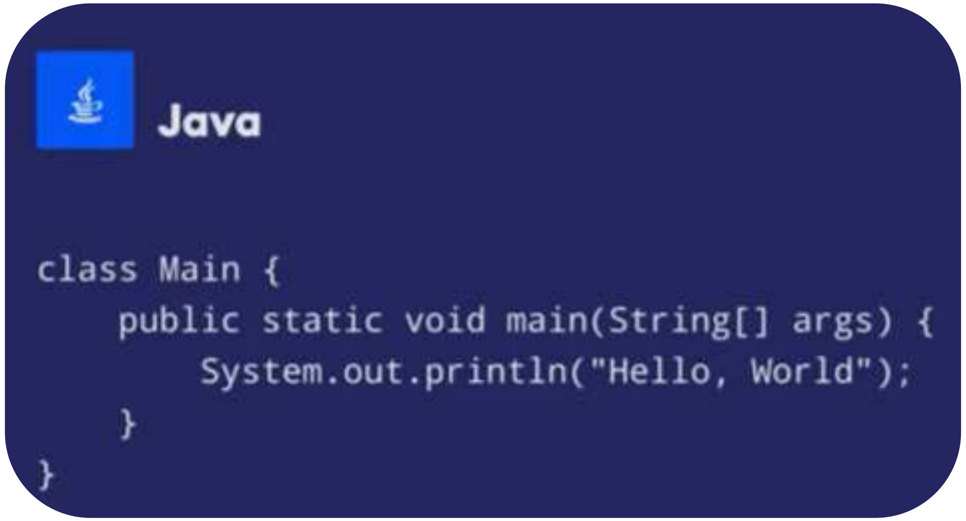
How to write "Hello world"

What is R



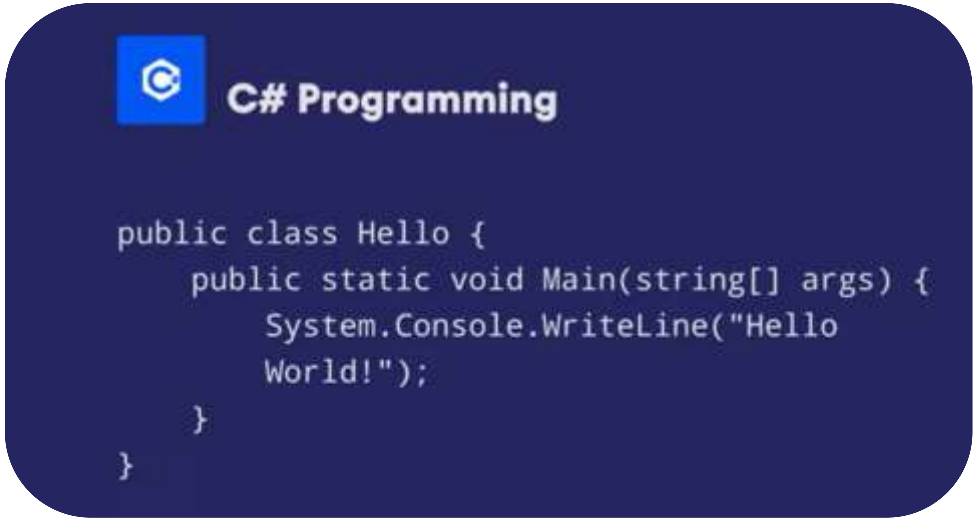
```
class Main {
    public static void main(String[] args) {
        System.out.println("Hello, World");
    }
}
```

What is R



Java

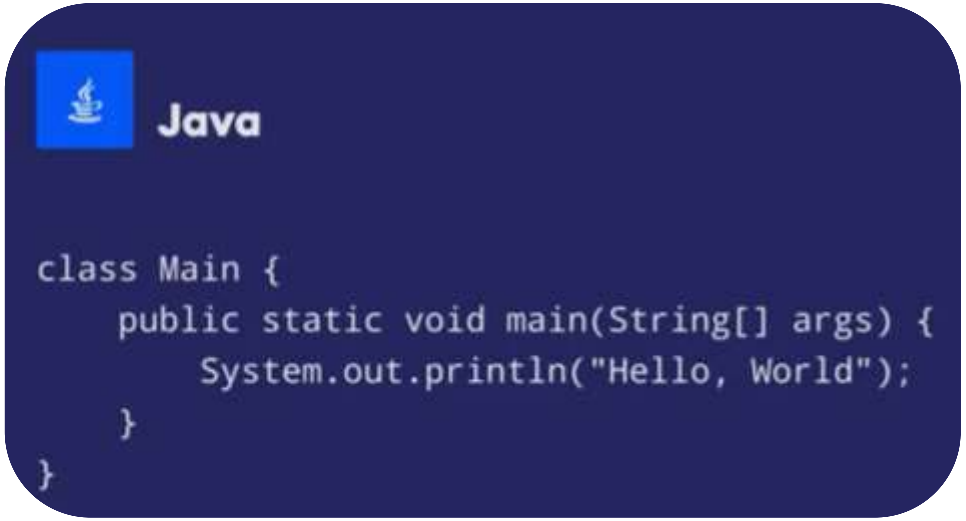
```
class Main {
    public static void main(String[] args) {
        System.out.println("Hello, World");
    }
}
```



C# Programming

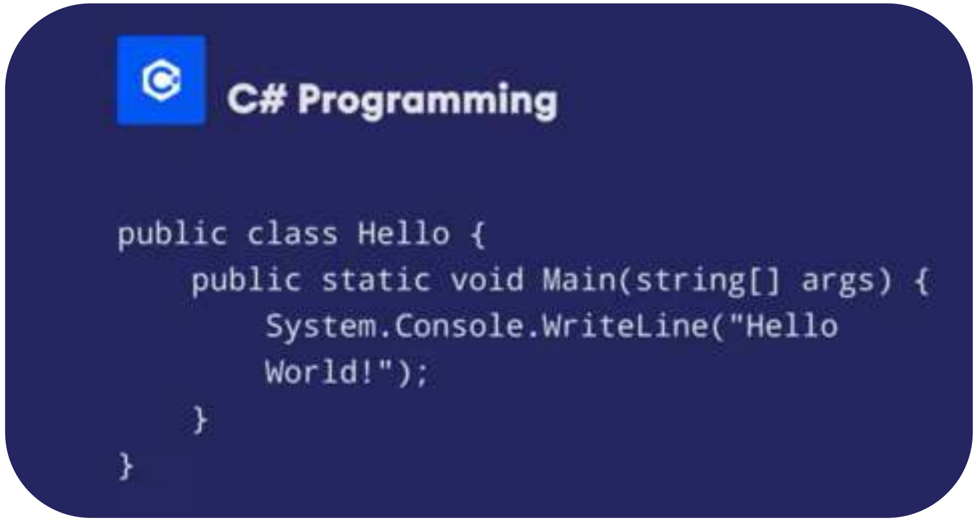
```
public class Hello {
    public static void Main(string[] args) {
        System.Console.WriteLine("Hello
        World!");
    }
}
```

What is R



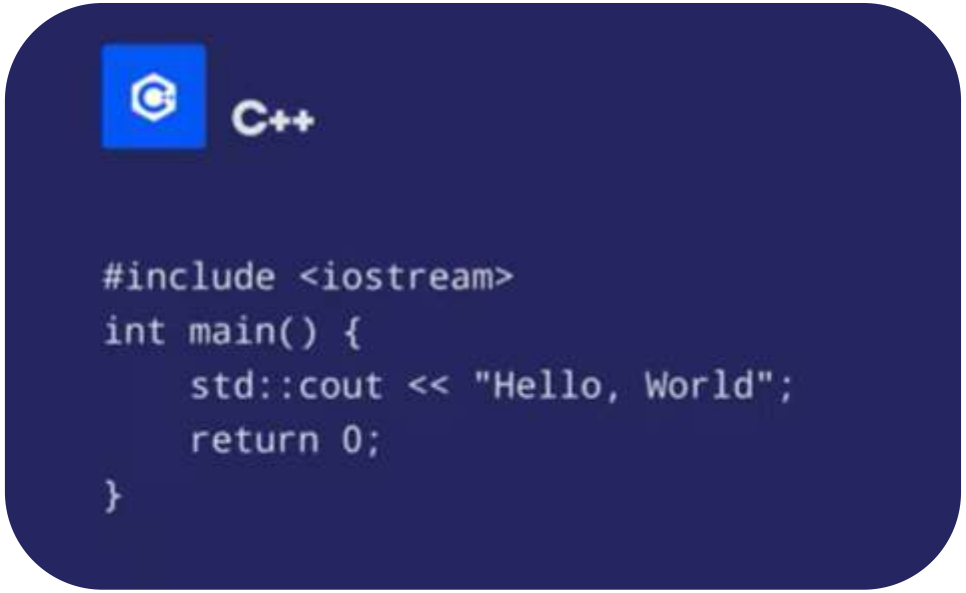
Java

```java
class Main {
    public static void main(String[] args) {
        System.out.println("Hello, World");
    }
}
```



C# Programming

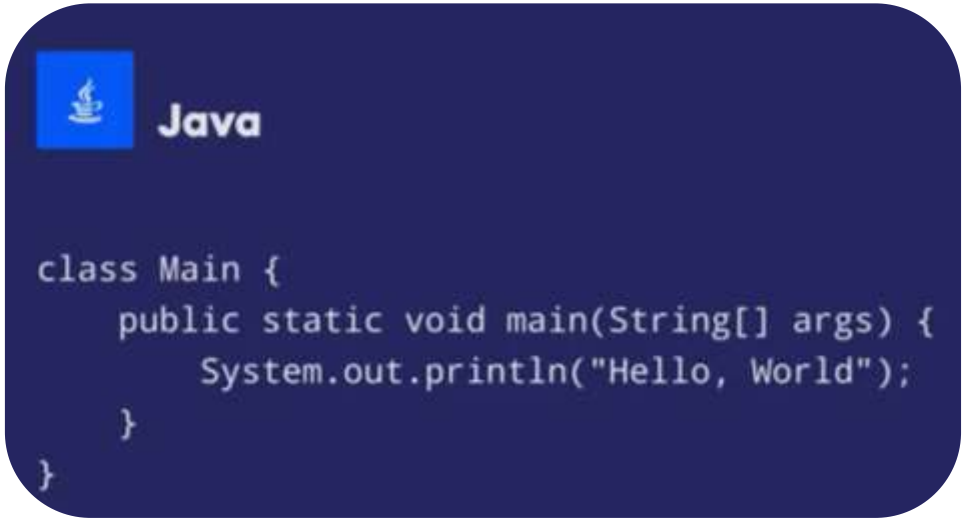
```csharp
public class Hello {
    public static void Main(string[] args) {
        System.Console.WriteLine("Hello
        World!");
    }
}
```



C++

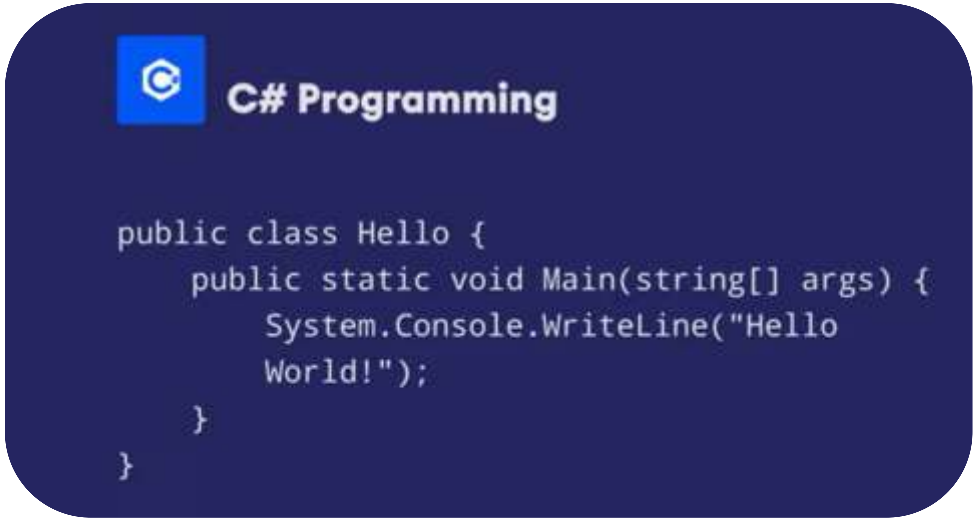
```cpp
#include <iostream>
int main() {
    std::cout << "Hello, World";
    return 0;
}
```

What is R



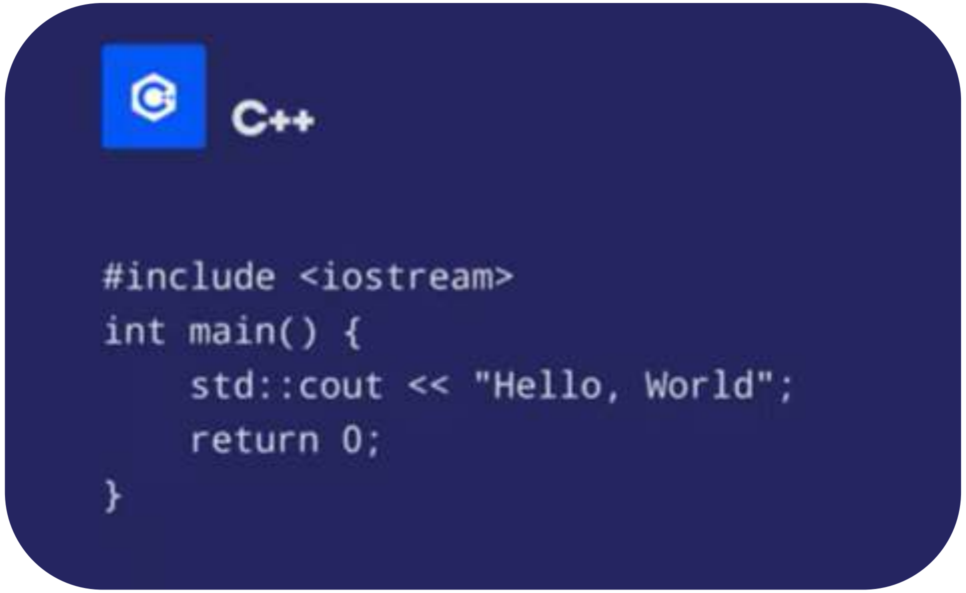
Java

```java
class Main {
    public static void main(String[] args) {
        System.out.println("Hello, World");
    }
}
```



C# Programming

```csharp
public class Hello {
    public static void Main(string[] args) {
        System.Console.WriteLine("Hello
        World!");
    }
}
```



C++

```cpp
#include <iostream>
int main() {
    std::cout << "Hello, World";
    return 0;
}
```

What is R

```
print "Hello world"
```

We will cover

How to obtain and install R

We will cover

How to obtain and install R

Basics of R

We will cover

How to obtain and install R

Basics of R

Import, manipulate, and export data

We will cover

How to obtain and install R

Basics of R

Import, manipulate, and export data

Perform data analysis

We will cover

How to obtain and install R

Basics of R

Import, manipulate, and export data

Perform data analysis

Visualize the results

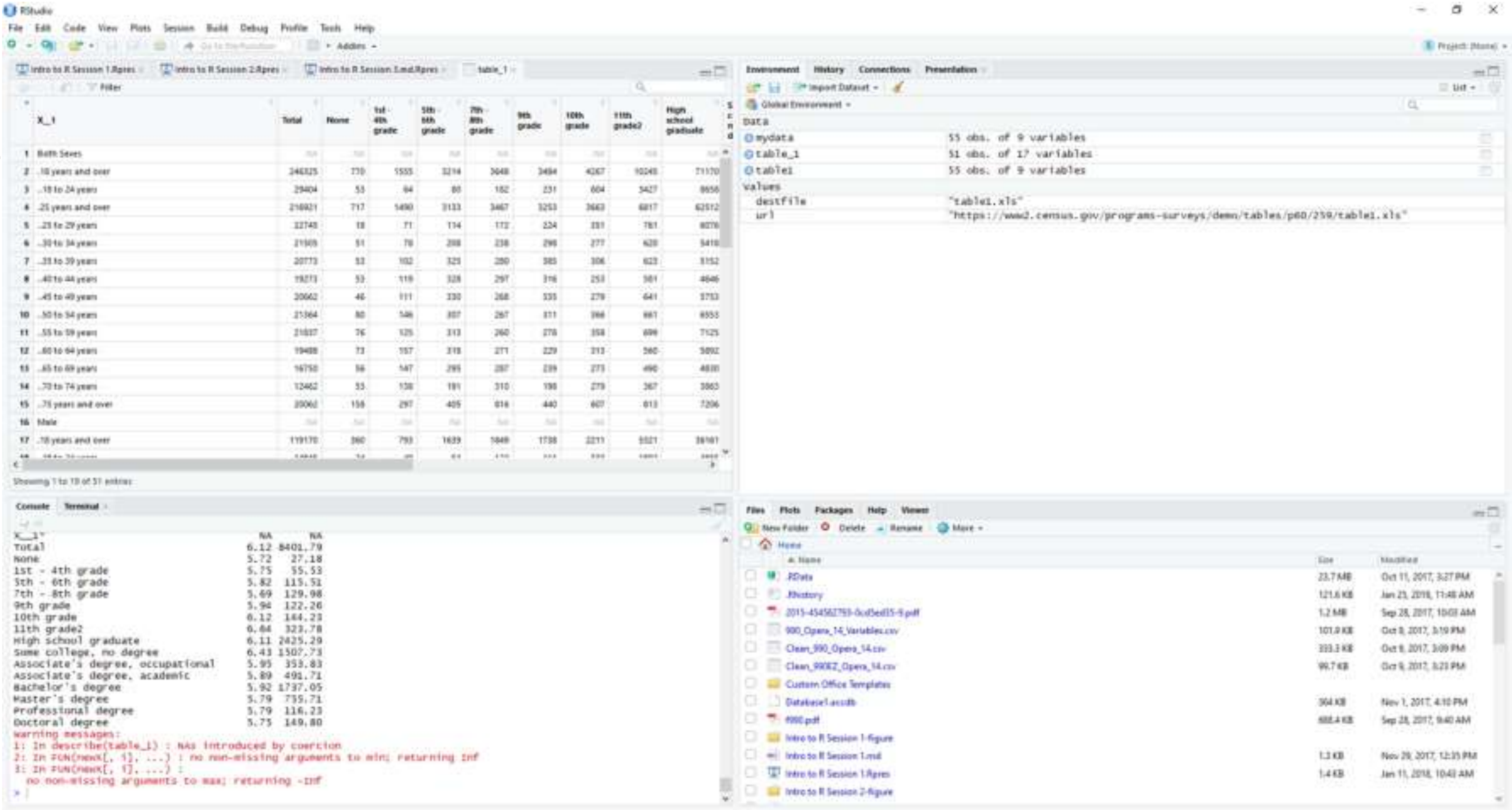
Download R

Ahmed Bargheet

Tour

Data/code

Sources


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Intro to R Session 1.Rpres
Intro to R Session 2.Rpres
Intro to R Session 3.md.Rpres
table_1
Filter
Environment History Connections Presentation
Import Dataset
List
Global Environment
Console Terminal
Files Plots Packages Help Viewer
New Folder Delete Rename More
Home

	X_1	Total	None	1st - 4th grade	5th - 6th grade	7th - 8th grade	9th grade	10th grade	11th grade2	High school graduate
1	Both Sexes	NA	NA	NA	NA	NA	NA	NA	NA	NA
2	.18 years and over	246325	770	1555	3214	3648	3494	4267	10248	71170
3	..18 to 24 years	29404	53	84	80	182	231	604	3427	8656
4	.25 years and over	216921	717	1490	3133	3467	3253	3663	6817	62512
5	..25 to 29 years	32748	18	71	114	172	224	351	781	8076
6	..30 to 34 years	21505	51	78	208	238	296	277	628	5418
7	..35 to 39 years	20773	53	102	325	280	385	306	623	5152
8	..40 to 44 years	19273	53	119	328	297	316	253	581	4646
9	..45 to 49 years	20062	46	111	330	268	335	279	641	5753
10	..50 to 54 years	21364	80	146	307	267	311	366	661	6553
11	..55 to 59 years	21037	76	125	313	260	278	358	699	7125
12	..60 to 64 years	19488	73	157	318	271	229	313	560	5892
13	..65 to 69 years	16758	56	147	295	287	239	273	490	4830
14	..70 to 74 years	12462	55	138	181	310	198	279	367	3863
15	..75 years and over	20062	158	297	405	814	440	607	813	7206
16	Male	NA	NA	NA	NA	NA	NA	NA	NA	NA
17	.18 years and over	119170	360	793	1639	1849	1738	2211	5521	38161

Showing 1 to 18 of 51 entries

```
Data
mydata                  55 obs. of 9 variables
table_1                 51 obs. of 17 variables
table1                  55 obs. of 9 variables
Values
  destfile              "table1.xls"
  url                   "https://www2.census.gov/programs-surveys/demo/tables/p60/259/table1.xls"
```

```
X__1*                                    NA      NA
Total                                  6.12 8401.79
None                                   5.72   27.18
1st - 4th grade                        5.75   55.53
5th - 6th grade                        5.82  115.51
7th - 8th grade                        5.69  129.98
9th grade                              5.94  122.26
10th grade                             6.12  144.23
11th grade2                            6.44  323.78
High school graduate                   6.11 2425.29
Some college, no degree                6.43 1507.73
Associate's degree, occupational       5.95  353.83
Associate's degree, academic           5.89  491.71
Bachelor's degree                      5.92 1737.05
Master's degree                        5.79  735.71
Professional degree                    5.79  116.23
Doctoral degree                        5.75  149.80
Warning messages:
1: In describe(table_1) : NAs introduced by coercion
2: In FUN(newX[, i], ...) : no non-missing arguments to min; returning Inf
3: In FUN(newX[, i], ...) :
  no non-missing arguments to max; returning -Inf
>
```

Name	Size	Modified
.RData	23.7 MB	Oct 11, 2017, 3:27 PM
.Rhistory	121.6 KB	Jan 23, 2018, 11:48 AM
2015-454562793-0cd5ed35-9.pdf	1.2 MB	Sep 28, 2017, 10:03 AM
990_Opera_14_Variables.csv	101.9 KB	Oct 8, 2017, 3:19 PM
Clean_990_Opera_14.csv	333.3 KB	Oct 8, 2017, 3:09 PM
Clean_990EZ_Opera_14.csv	99.7 KB	Oct 9, 2017, 3:23 PM
Custom Office Templates		
Database1.accdb	364 KB	Nov 1, 2017, 4:10 PM
f990.pdf	688.4 KB	Sep 28, 2017, 9:40 AM
Intro to R Session 1-figure		
Intro to R Session 1.md	1.3 KB	Nov 29, 2017, 12:35 PM
Intro to R Session 1.Rpres	1.4 KB	Jan 11, 2018, 10:43 AM
Intro to R Session 2-figure		

Console

Plots/Help

Basic R

Basic R

- Basic calculation

Process	Equation	Run
Addition	2+2	[Hit enter]
Subtraction	10-2	[Hit enter]
Multiplication	2*2	[Hit enter]
Division	10/2	[Hit enter]
log	log(100)	[Hit enter]
Square root	sqrt(4)	[Hit enter]
Exponentiation	2^2	[Hit enter]

Variable

Variable

Variables allow you to **store** and **manipulate** data.

Variable

Variables allow you to **store** and **manipulate** data.

Variables have a value and name

Variable

Variables allow you to **store** and **manipulate** data.

Variables have a value and name

For example

```
X <- 40
```

Variable

Variables allow you to **store** and **manipulate** data.

Variables have a value and name

For example

```
X <- 40
```

Another example

```
name <- " Write your name"
```

slido

What is the output of this code?

```
x = 9
y = 3
x = y
```

ⓘ Start presenting to display the poll results on this slide.

Data type

Data type

1. Numeric (a <- 2)

Data type

1. Numeric (a <- 2)

2. Character (a <- "Hello")

Data type

1. Numeric (a <- 2)

2. Character (a <- "Hello")

3. Boolean a <- 10   a < 10

Relational operator

1. > greater than

2. < less than

3. <= less than or equal to

4. >= greater than or equal to

5. == equal

6. != not equal

If a = 10. Choose the answer

a != 2

ⓘ Start presenting to display the poll results on this slide.

If a = 10. Choose the answer

$a < 6$ | 12

ⓘ Start presenting to display the poll results on this slide.

Data structure

Data structure

When working with data sets, we need to use **data structures** to **store** and **manipulate** data.

R provides several data structures that provide functions to **manipulate** and **manage** the data they store.

Data structure

Character

Data structure

Character

```
a <- "Hi"

b <- "there"
```

Data structure

Character

```
a <- "Hi"

b <- "there"
```

How to manipulate this?

Data structure

Character

```
a <- "Hi"

b <- "there"
```

How to manipulate this?

```
paste (a , b)
```

Data structure

Character

```
a <- "Hi"

b <- "there"
```

How to manipulate this?

```
paste (a, b, sep=???)
```

Data structure

Numeric

```
a <- 10

b <- 5
```

How to manipulate this?

```
a*12 + b/3
```

Data structure

Vector

A vector is a basic data structure in R.

It is a **sequence of elements** of the **same type**.

Data structure

Vector

A vector is a basic data structure in R.

It is a **sequence of elements** of the **same type**.

```
names <- c("Dorota", "Ahmed", "John")
```

Data structure

Vector

A vector is a basic data structure in R.

It is a **sequence of elements** of the **same type**.

```
names <- c("Dorota", "Ahmed", "John")

num <- c(1,2,3)
```

Data structure

Vector

Can we manipulate that?

```
names <- c("Dorota", "Ahmed", "John")
```

Data structure

Vector

Can we manipulate that?

```
names <- c("Dorota", "Ahmed", "John")
```

Extract only Ahmed!!

Data structure

Vector

Can we manipulate that?

```
names <- c("Dorota", "Ahmed", "John")
```

Extract only Ahmed!!

```
names[2]
```

slido

How to access and store John in a separate variable?

```
names = c("Dorota", "Ahmed", "John")
```

ⓘ Start presenting to display the poll results on this slide.

Data structure

Vector

We can also define **character** indices for the elements, in addition to the **numeric** indices.

Data structure

Vector

We can also define **character** indices for the elements, in addition to the **numeric** indices.

For example:

```
ages <- c("Dorota"=20, "Ahmed"=21, "John"=22)
```

Data structure

Vector

We can also define **character** indices for the elements, in addition to the **numeric** indices.

For example:

```
ages <- c("Dorota"=20, "Ahmed"=21, "John"=22)
```

```
ages[1]
```

Data structure

Vector

We can also define **character** indices for the elements, in addition to the **numeric** indices.

For example:

```
ages <- c("Dorota"=20, "Ahmed"=21, "John"=22)
```

```
ages[1]

ages[[1]]
```

Data structure

Vector

We can also use a **negative index** to **remove** an element in the vector.

For example:

```
ages <- c("Dorota"=20, "Ahmed"=21, "John"=22)
```

Data structure

Vector

We can also use a **negative index** to **remove** an element in the vector.

For example:

```
ages <- c("Dorota"=20, "Ahmed"=21, "John"=22)
```

```
ages[-3]
```

Data structure

Vector

We can also specify a **range** to select only a subset of the elements..

Data structure

Vector

We can also specify a range to select only a subset of the elements..

For example:

```
ages <- c("Dorota", "Ahmed", "John", "Mona", "Dave")
```

```
ages[2:4]
```

Data structure

Vector (Recap)

It is a **sequence of elements** of the **same type**.

Data structure

Vector (Recap)

It is a **sequence of elements** of the **same type**.

```
names <- c("Dorota", "Ahmed", "John")
num <- c(1,2,3)
```

Data structure

Vector (Recap)

It is a **sequence of elements** of the **same type**.

```
names <- c("Dorota", "Ahmed", "John")
num <- c(1,2,3)
```

We can also define **character** indices

```
ages <- c("Dorota"=20, "Ahmed"=21, "John"=22)
```

Data structure

Vector (Recap)

It is a **sequence of elements** of the **same type**.

```
names <- c("Dorota", "Ahmed", "John")
num <- c(1,2,3)
```

We can also define **character** indices

```
ages <- c("Dorota"=20, "Ahmed"=21, "John"=22)
```

Access to **element** in a vector

ages [3] OR ages [[3]]

Data structure

Vector (Recap)

It is a **sequence of elements** of the **same type**.

```
names <- c("Dorota", "Ahmed", "John")
num <- c(1,2,3)
```

We can also define **character** indices

```
ages <- c("Dorota"=20, "Ahmed"=21, "John"=22)
```

Access to **element** in a vector

```
ages [3] OR ages [[3]]
```

Remove an element in the vector.

```
ages[-2]
```

Data structure

Vector (Recap)

It is a **sequence of elements** of the **same type**.

```
names <- c("Dorota", "Ahmed", "John")
num <- c(1,2,3)
```

We can also define **character** indices

```
ages <- c("Dorota"=20, "Ahmed"=21, "John"=22)
```

Access to **element** in a vector

```
ages [3] OR ages [[3]]
```

Remove an element in the vector.

```
ages[-2]
```

We can define a range

```
ages [2:3]
```

slido

What is the output of this code?

```
n = c(8, 4, 2, 3, 5)
x = n[2:4]
x = x[-1]
x[1]
```

ⓘ Start presenting to display the poll results on this slide.

Data structure

Vector Arithmetic

Two vectors of the **same length** can be **added**, **subtracted**, **multiplied** or **divided** resulting in a new vector.

Data structure

Vector Arithmetic

Two vectors of the **same length** can be **added**, **subtracted**, **multiplied** or **divided** resulting in a new vector.

```
a <- c(2, 6, 1, 5)
b <- c(5, 3, 4, 8)

#addition
a+b

#subtraction
a-b

#multiplication
a*b

#division
a/b
```

Data structure

Vector Arithmetic

We can change an element in a vector

Data structure

Vector Arithmetic

We can change an element in a vector

```
a <- c(2, 6, 1, 5)
```

Data structure

Vector Arithmetic

We can change an element in a vector

```
a <- c(2, 6, 1, 5)
```

How to change 6 to 7?

Data structure

Vector Arithmetic

We can change an element in a vector

a <- c(2, 6, 1, 5)

How to change 6 to 7?

a[2] <- 7

Data structure

Vector Arithmetic

Calculate the **mean (average)** and **median (middle number)**

Data structure

Vector Arithmetic

Calculate the **mean (average)** and **median (middle number)**

```
a <- c(2, 6, 1, 5, 42)
mean(a)
```

Data structure

Vector Arithmetic

Calculate the **mean (average)** and **median (middle number)**

```
a <- c(2, 6, 1, 5, 42)
mean(a)
median(a)
```

Data structure

Lists

Lists are **similar to** vectors but can hold **different types of data**.

Data structure

Lists

Lists are **similar to** vectors but can hold **different types of data**.

For example,

```
a <- list("James", "Bob", c(2, 4, 8), 42)
```

slido

What is the type of the 2nd element of this list?

```
list("ABC", c(1, 2, 3), 42)
```

ⓘ Start presenting to display the poll results on this slide.

Data structure

Matrix

A matrix is a **two-dimensional data** set with **rows and columns**.

It is similar to a vector but has an additional dimension.

Data structure

Matrix

A matrix is a **two-dimensional data** set with **rows and columns**.

It is similar to a vector but has an additional dimension.

For example,

```
a <- matrix(c(1,2,3,4,5,6), nrow = 2, ncol = 3)
```

Data structure

Recap

Data structure

Recap

We have seen how to store data in **vectors**, **lists** and **matrices**.

Data structure

Recap

We have seen how to store data in **vectors**, **lists** and **matrices**.

Vectors store elements of the **same type** using **one dimension**.

Data structure

Recap

We have seen how to store data in **vectors**, **lists** and **matrices**.

Vectors store elements of the **same type** using **one dimension**.

Matrices are like vectors and have **2 dimensions**: rows and columns.

Data structure

Recap

We have seen how to store data in **vectors**, **lists** and **matrices**.

Vectors store elements of the **same type** using **one dimension**.

Matrices are like vectors and have **2 dimensions**: rows and columns.

Lists are similar to vectors and allow you to store **different types** of elements.

Data structure

Recap

We have seen how to store data in **vectors**, **lists** and **matrices**.

Vectors store elements of the **same type** using **one dimension**.

Matrices are like vectors and have **2 dimensions**: rows and columns.

Lists are similar to vectors and allow you to store **different types** of elements.

Most commonly our data comes in the form of a table and each column can be of different types

Data structure

Data frame

Data structure

Data frame

A **data frame** is a **table**, where each column has a name and can **contain any type of data**.

Each column must contain the same number of data items.

Data structure

Data frame

A **data frame** is a **table**, where each column has a name and can **contain any type of data**.

Each column must contain the same number of data items.

For example:

```
a <- data.frame("id" = 1:2, "name" = c("James", "Amy"),  "age" = c(42,18))
```

Data structure

Data frame operation

Data structure

Data frame operation

Mean to specific column

```
mean(a$age)
median(a$id)
```

Data structure

Data frame operation

Summary function

Data structure

Data frame operation

Summary function

```
a <- data.frame("id" = 1:2, "name" = c("James", "Amy"),  "age" = c(42,18))
summary(a)
```

Data structure

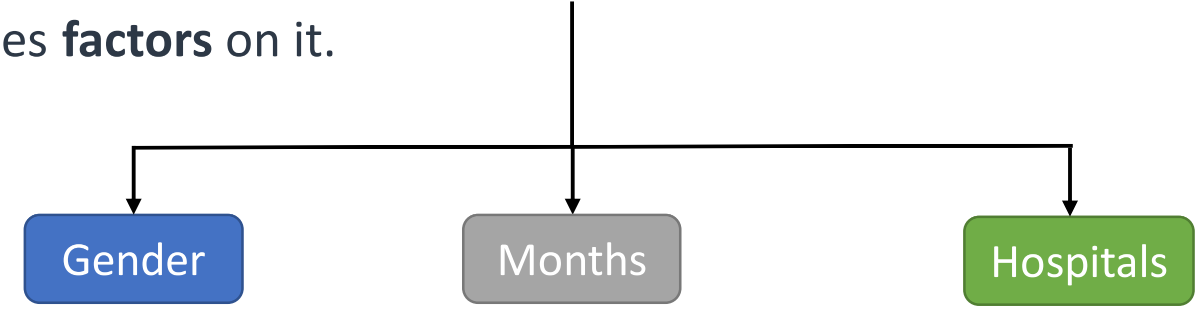
Factor

When a data frame has a **text column**, R treats that column as **categorical data** and creates **factors** on it.

Data structure

Factor

When a data frame has a **text column**, R treats that column as **categorical data** and creates **factors** on it.



Gender

Months

Hospitals

Ahmed Bargheet

Data structure

Factor

When a data frame has a **text column**, R treats that column as **categorical data** and creates **factors** on it.

For example.

```
gender <- factor(c("Male", "Female", "Male"))
```

Data structure

Factor

When a data frame has a **text column**, R treats that column as **categorical data** and creates **factors** on it.

For example.

```
gender <- factor(c("Male", "Female", "Male"))
```

How to know the class of gender and a$name data?
What is the difference between a$name and gender?
Can you change the levels in gender data?

Data structure

Factor

When a data frame has a **text column**, R treats that column as **categorical data** and creates **factors** on it.

For example.

```
gender <- factor(c("Male", "Female", "Male"))
```

How to know the class of gender and a$name data?
What is the difference between a$name and gender?
Can you change the levels in gender data?
Why factor(gender, levels = "Male","Female") will not work?

Data structure

Factor

When a data frame has a **text column**, R treats that column as **categorical data** and creates **factors** on it.

For example.

```
gender <- factor(c("Male", "Female", "Male"))
```

How to know the class of gender and a$name data?
What is the difference between a$name and gender?
Can you change the levels in gender data?
Why `factor(gender, levels = "Male","Female")` **will not work?**
Can you change a$name to a factor?

Which of the following data structures allows you to store elements of different types?

ⓘ Start presenting to display the poll results on this slide.

slido

Which of the following statements is true?

ⓘ Start presenting to display the poll results on this slide.

TUSEN TAKK!