

Functions

Functions

$\log_2(2)$

Functions

`log2(2)`

Function Name



A diagram illustrating the components of a function call. A light blue rounded rectangle containing the text 'Function Name' has a vertical arrow pointing upwards from its top-right corner to the 'log2' part of the code 'log2(2)' shown above. The 'log2' is in light blue, the opening parenthesis '(' is yellow, the '2' inside is purple, and the closing parenthesis ')' is yellow.

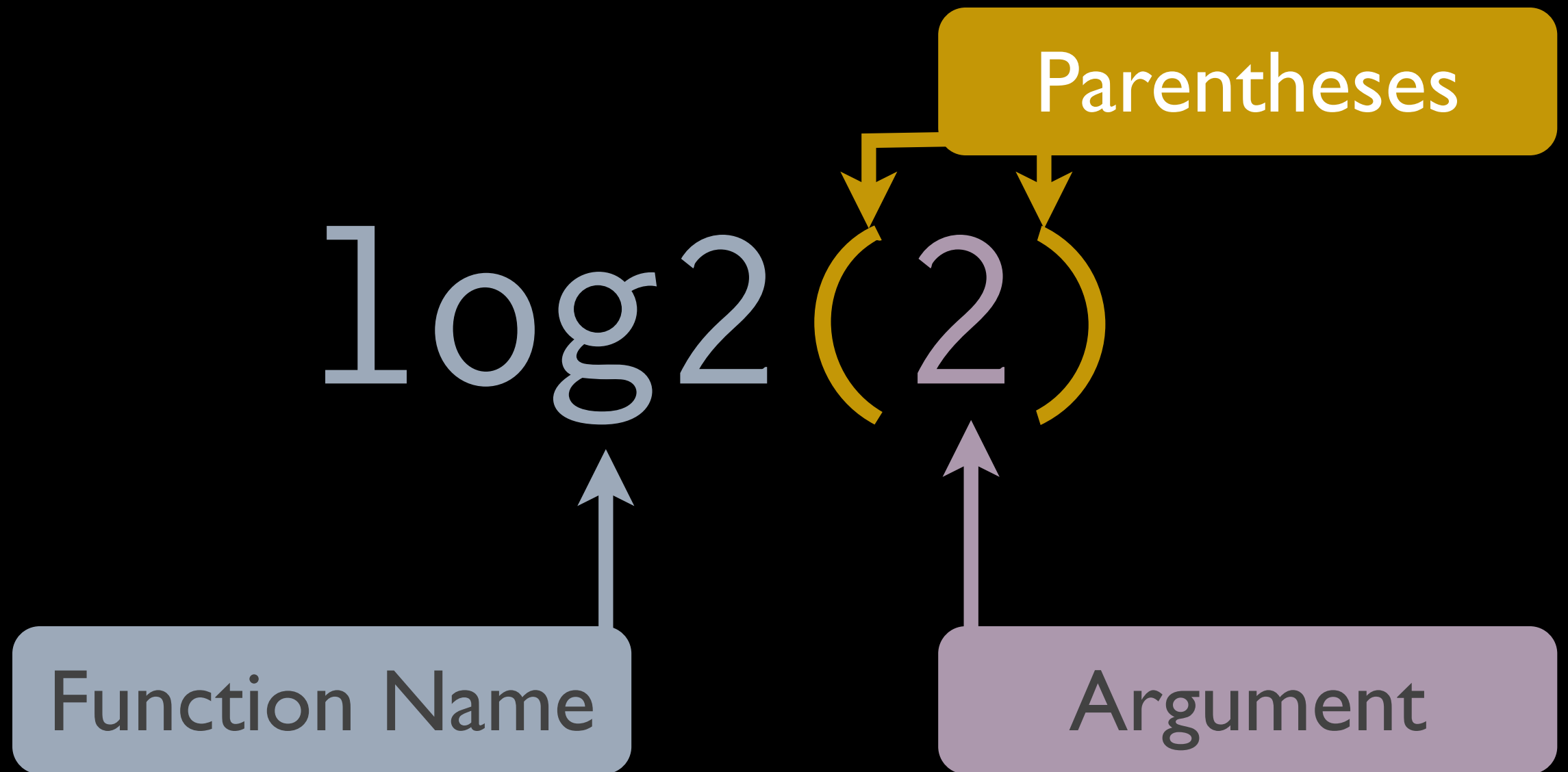
Functions

`log2(2)`

Function Name

Argument

Functions



Variables

Variables

num <- 2

Variables

num <- 2

Variable Name



Variables

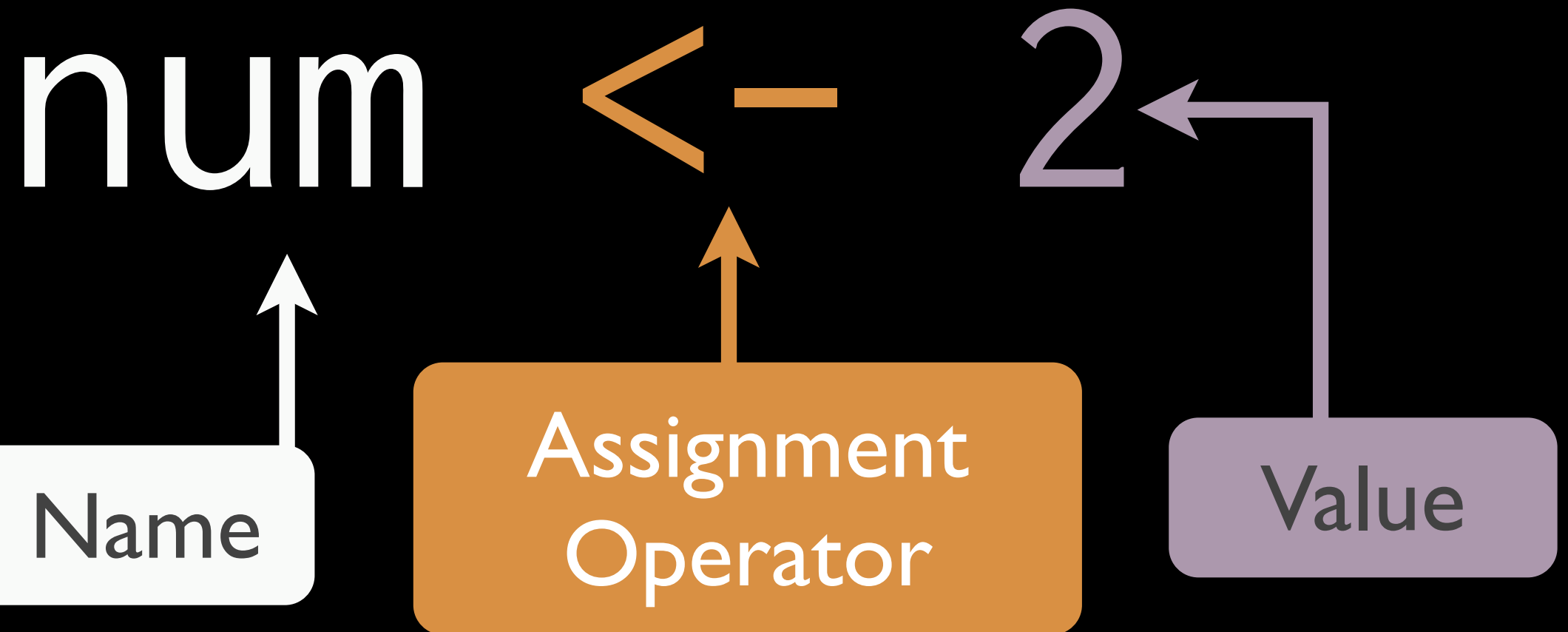
num <- 2

Variable Name



Assignment
Operator

Variables



Functions & Variables

Functions & Variables

```
num <- 2
```

Functions & Variables

```
num <- 2
```

```
log2(num)
```

Functions & Variables

Create
Variable



```
num <- 2
```

```
log2(num)
```

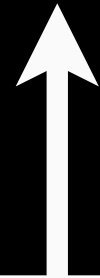
Functions & Variables

Create
Variable



```
num <- 2
```

```
log2(num)
```



Use As
Argument

Multiple Arguments

Multiple Arguments

```
df <- read.table("H:/input.txt", header = T, sep = "\t")
```

Multiple Arguments

```
df <- read.table("H:/input.txt", header = T, sep = "\t")
```



Variable Name

Multiple Arguments

Assignment



```
df <- read.table("H:/input.txt", header = T, sep = "\t")
```

Variable Name



Multiple Arguments

Assignment



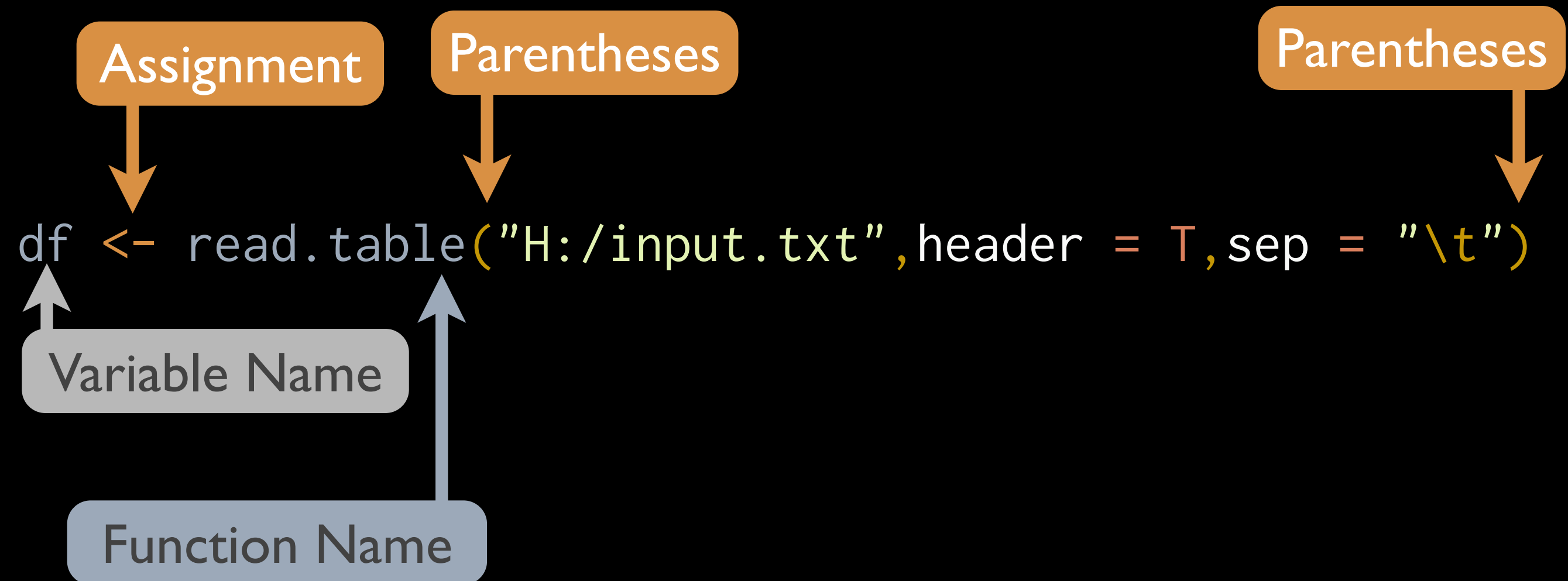
```
df <- read.table("H:/input.txt", header = T, sep = "\t")
```

Variable Name

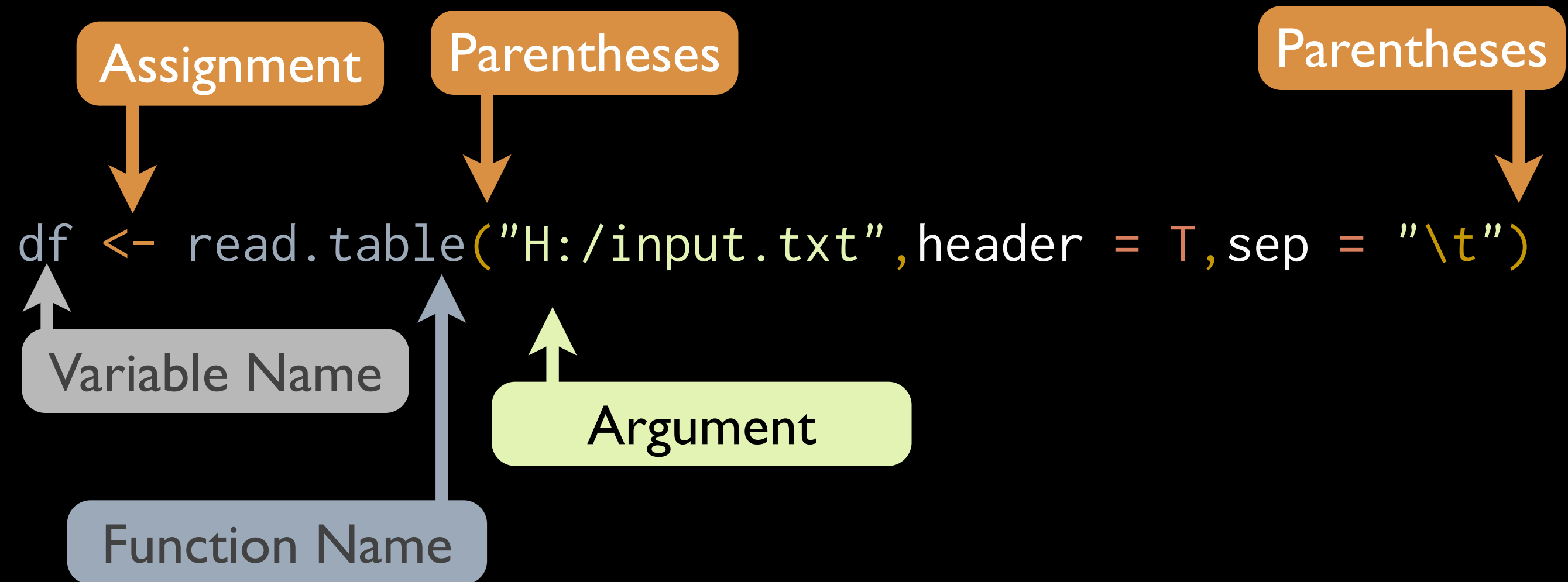


Function Name

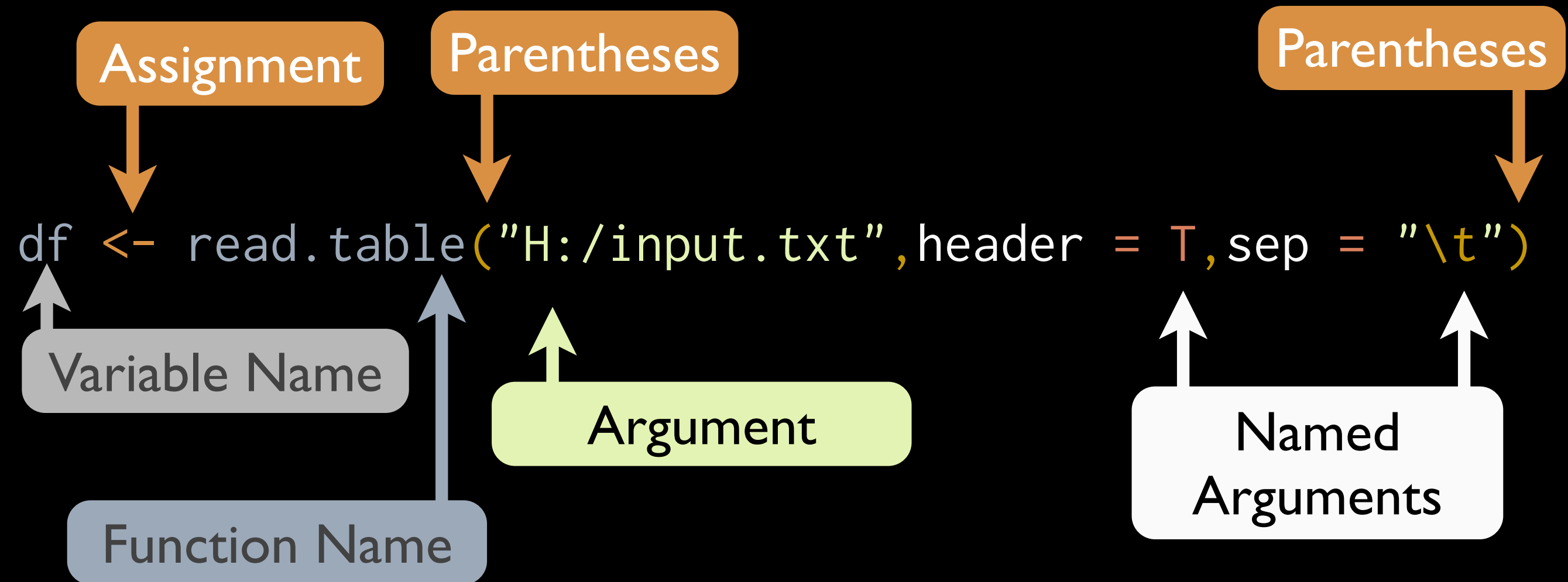
Multiple Arguments



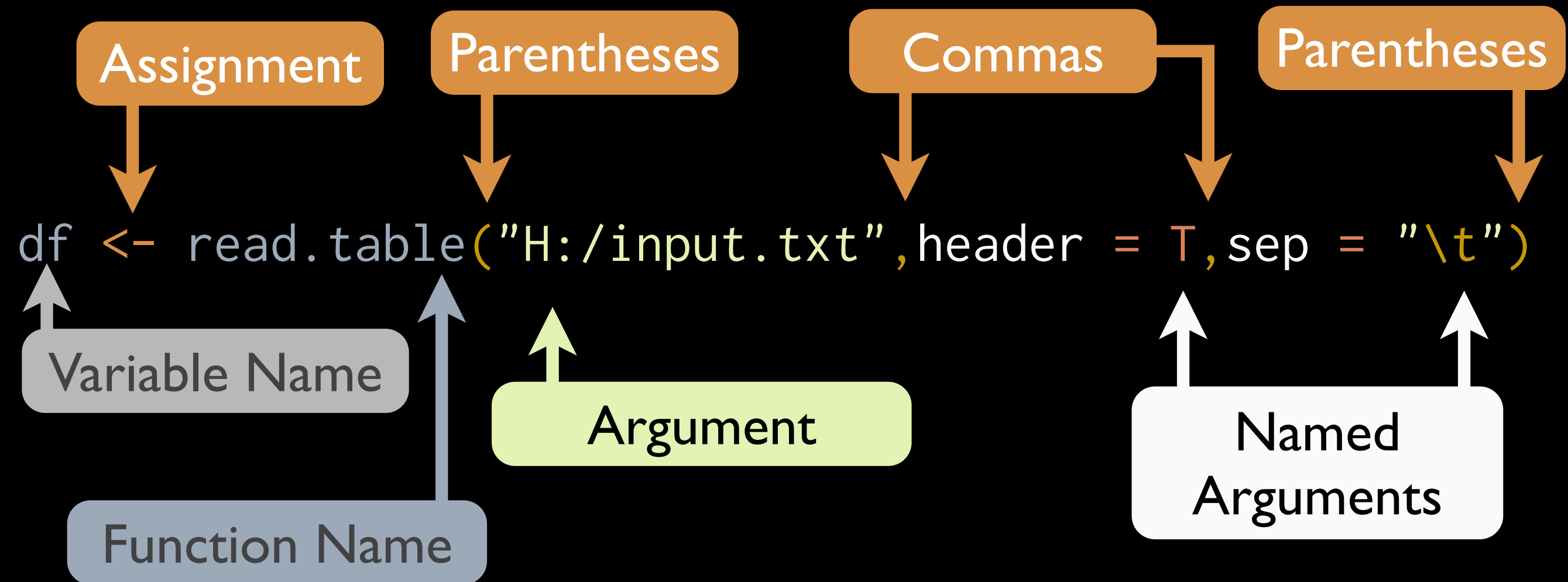
Multiple Arguments



Multiple Arguments



Multiple Arguments



Data Frames

Data Frames

```
data[1, 3]
```

Data Frames

data[1, 3]



Variable Name

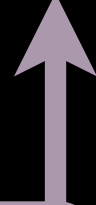
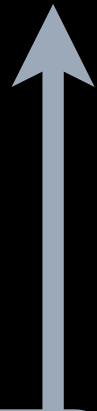
The diagram illustrates the components of the code snippet 'data[1, 3]'. A light blue rounded rectangle containing the text 'Variable Name' has a white arrow pointing upwards to the 'data' portion of the code. The code itself is displayed in a monospaced font, with 'data' in white, the opening bracket '[' in yellow, '1' in white, a comma in yellow, '3' in white, and the closing bracket ']' in yellow.

Data Frames

data[1, 3]

ROWS

Variable Name



Data Frames

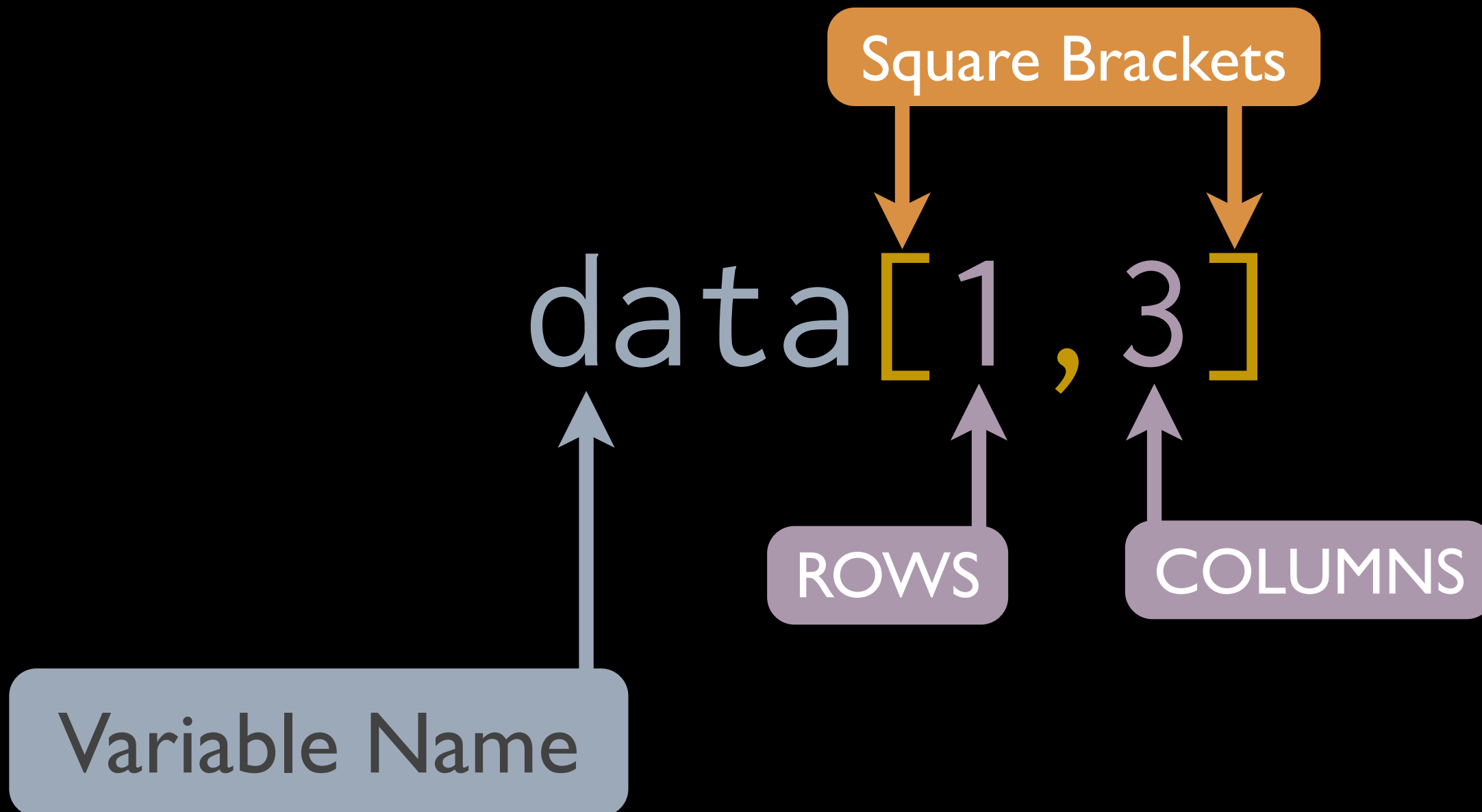
data[1, 3]

ROWS

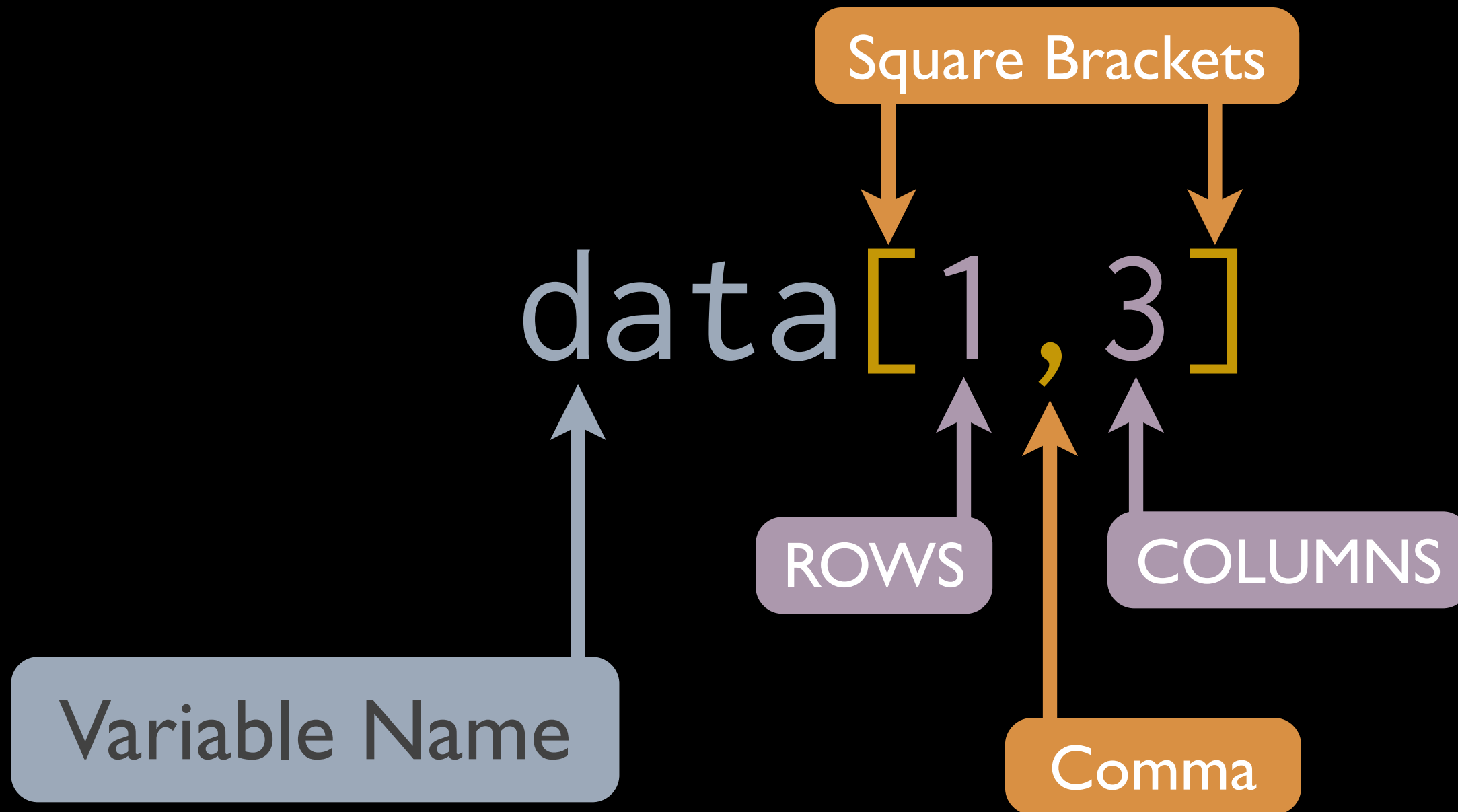
COLUMNS

Variable Name

Data Frames



Data Frames



Data Frames

Give me the value stored in
row 1, column 3 of 'data'

data[1, 3]

Square Brackets

ROWS

COLUMNS

Variable Name

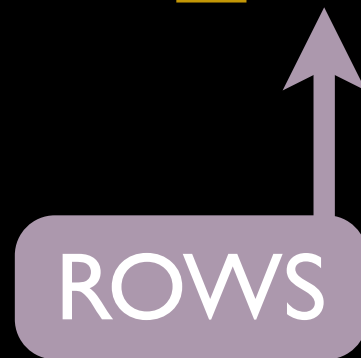
Comma

Data Frames

```
data[1,]
```

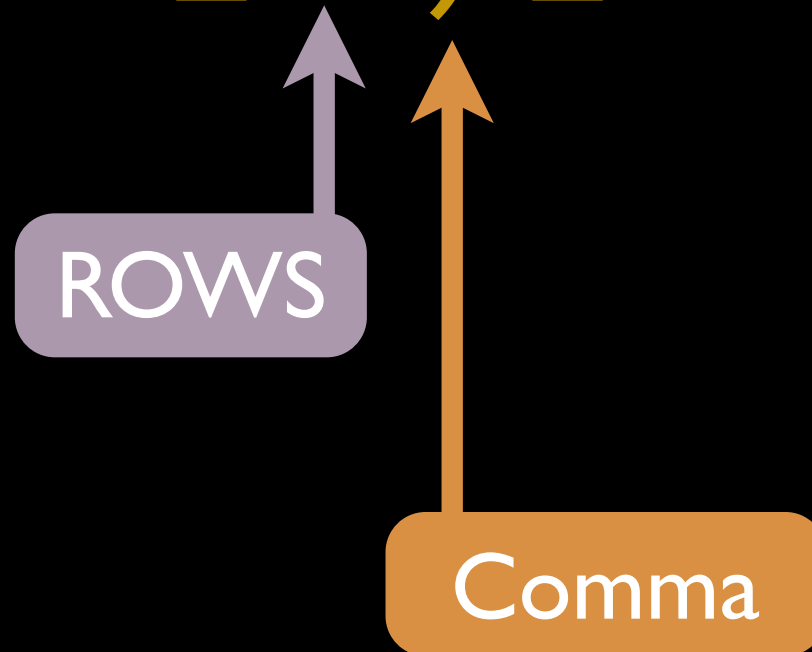
Data Frames

data[1,]



Data Frames

data[1,]



Data Frames

Give me the values
stored in row 1 of 'data'
(all columns)

data[1,]

ROWS

Comma

Data Frames

```
data[, 3]
```

Data Frames

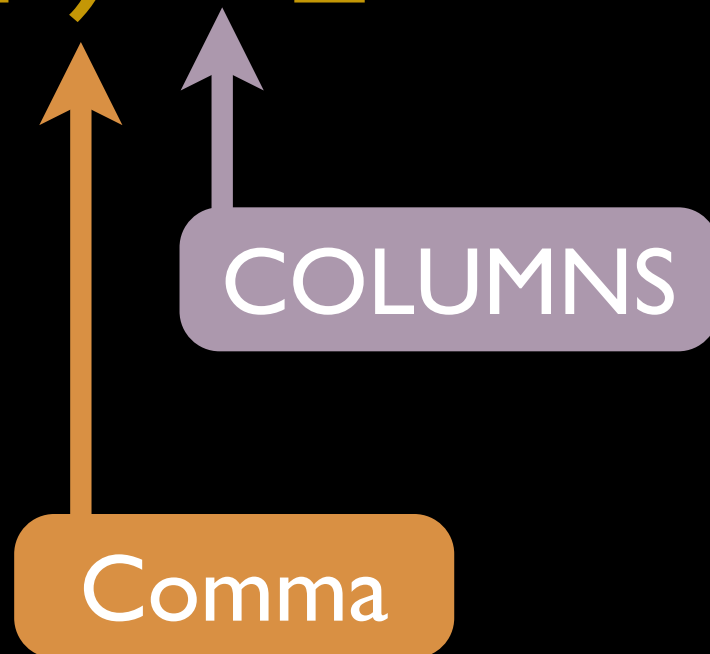
data[, 3]

COLUMNS

A diagram illustrating the indexing of a data frame. The text 'data[, 3]' is shown. The opening square bracket '[', the comma ',', and the closing square bracket ']' are highlighted in yellow. The number '3' is highlighted in light purple. A light purple arrow points upwards from a light purple rounded rectangle containing the word 'COLUMNS' to the number '3'.

Data Frames

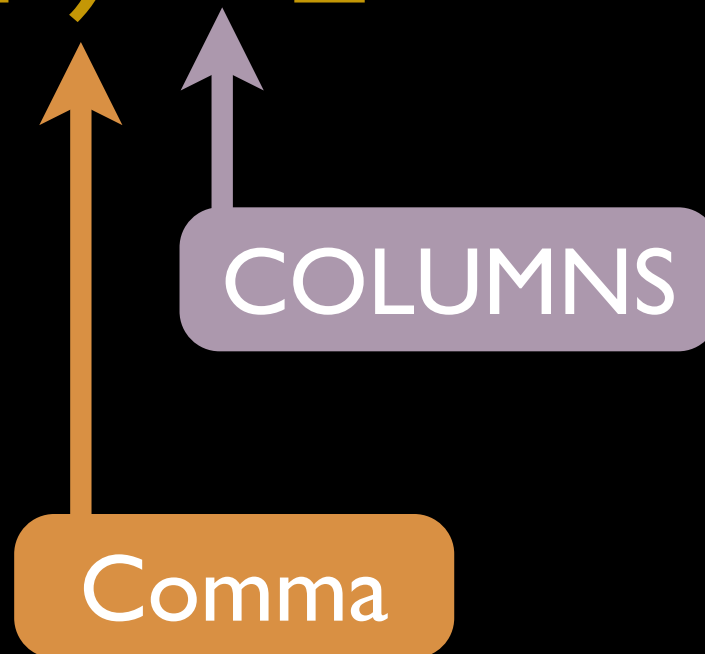
data[, 3]



Data Frames

Give me the values
stored in column 3 of 'data'
(all rows)

data[, 3]



Data Frames

```
data[1:3, 1:4]
```

Data Frames

```
data[1:3, 1:4]
```

ROWS

COLUMNS

Data Frames

Give me the values
stored in rows 1 - 3, columns
1 - 4 of 'data'

```
data[1:3, 1:4]
```

ROWS

COLUMNS

Data Frames

```
data[, "gene"]
```

Data Frames

```
data[, "gene"]
```

ROWS

COLUMNS

Data Frames

Give me the values
stored in the column named
“gene” of ‘data’
(all rows)

```
data[, "gene"]
```

ROWS

COLUMNS

Data Frames

data\$gene

Data Frames

data\$gene

Variable Name

A light blue rounded rectangle containing the text 'Variable Name' has a white arrow pointing upwards from its top-right corner to the 'data' portion of the 'data\$gene' expression above it.

Data Frames

data\$gene

Variable Name



dollar sign

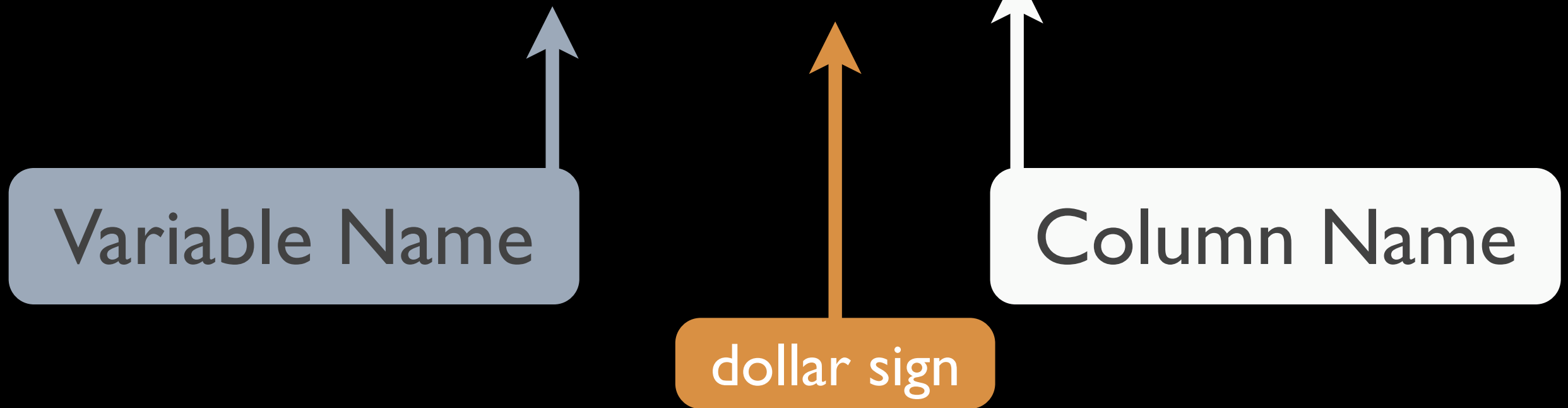
Data Frames

data\$gene

Variable Name

Column Name

dollar sign



Data Frames

Give me the values
stored in the column named
“gene” of ‘data’
(all rows)

data\$gene

Variable Name

Column Name

dollar sign

Index Vectors

Index Vectors

```
iv <- data$relationship == "overlapping"
```

Index Vectors

```
iv <- data$relationship == "overlapping"
```

Which rows have their 'relationship' column equal to the string "overlapping"

Index Vectors

```
iv <- data$relationship == "overlapping"
```



```
[FALSE,  
 FALSE,  
  TRUE,  
 FALSE,  
  TRUE,  
  ...]
```

Which rows have their 'relationship' column
equal to the string "overlapping"

Index Vectors

```
iv <- data$relationship == "overlapping"
```


Index Vectors

```
iv <- data$relationship == "overlapping"
```

```
sub <- data[iv,]
```

Index Vectors

```
iv <- data$relationship == "overlapping"
```

```
sub <- data[iv,]
```



ROWS

COLUMNS

Index Vectors

```
iv <- data$relationship == "overlapping"
```

```
sub <- data[iv,]
```

Give me the rows where the column 'relationship' is equal to the string 'overlapping' (all columns)

ROWS

COLUMNS

Index Vectors

```
iv <- data$relationship == "overlapping"  
sub <- data[iv,]
```

Index Vectors

```
iv <- data$relationship == "overlapping"  
sub <- data[iv,]  
  
----- Combined -----
```

Index Vectors

```
iv <- data$relationship == "overlapping"
```

```
sub <- data[iv,]
```

----- Combined -----

```
sub <- data[data$relationship == "overlapping",]
```


Index Vectors

```
iv <- data$relationship == "overlapping"
```

```
sub <- data[iv,]
```

----- Combined -----

```
sub <- data[data$relationship == "overlapping",]
```



Give me the rows where the
column 'relationship' is equal to the
string 'overlapping'
(all columns)


Index Vectors

```
iv <- data$relationship == "overlapping"
```

```
sub <- data[iv,]
```

----- Combined -----

```
sub <- data[data$relationship == "overlapping",]
```



Give me the rows where the
column 'relationship' is equal to the
string 'overlapping'
(all columns)

