Calling Functions

In this we are focuses on how functions are called and interpreted in an R session, covering topics such as scoping rules and argument matching.

Scoping Rules

Scoping rules in R determine how the language compartmentalizes objects and retrieves them in a given session.They also define situations where duplicate object names can co-exist.

Environments

1. Environments are compartments used by R to organize objects.They include the global environment (work space), package environments,and function environments.
2. The global environment is where objects created during an R session are stored.

```
foo <- 4+5
bar <- "stringtastic"
ls()
```

```
## [1] "bar" "foo"
```

3. Package environments store the functions and data sets associated with a particular package.

Package environments have visible functions available to users and invisible functions that provide internal support.They also manage imports, indicating functions or objects from other libraries needed for the package's functionali- ty.

```
ls("package:graphics")
```

```
##  [1] "abline"          "arrows"          "assocplot"       "axis"
##  [5] "Axis"            "axis.Date"       "axis.POSIXct"    "axTicks"
##  [9] "barplot"         "barplot.default" "box"             "boxplot"
## [13] "boxplot.default" "boxplot.matrix"  "bxp"             "cdplot"
## [17] "clip"            "close.screen"    "co.intervals"    "contour"
## [21] "contour.default" "coplot"          "curve"           "dotchart"
## [25] "erase.screen"    "filled.contour"  "fourfoldplot"    "frame"
## [29] "grconvertX"      "grconvertY"      "grid"            "hist"
## [33] "hist.default"    "identify"        "image"           "image.default"
## [37] "layout"          "layout.show"     "lcm"             "legend"
## [41] "lines"           "lines.default"   "locator"         "matlines"
## [45] "matplot"         "matpoints"       "mosaicplot"      "mtext"
```

```
## [49] "pairs"           "pairs.default"   "panel.smooth"    "par"
## [53] "persp"           "pie"             "plot"            "plot.default"
## [57] "plot.design"     "plot.function"   "plot.new"        "plot.window"
## [61] "plot.xy"         "points"          "points.default"  "polygon"
## [65] "polypath"        "rasterImage"     "rect"            "rug"
## [69] "screen"          "segments"        "smoothScatter"   "spineplot"
## [73] "split.screen"    "stars"           "stem"            "strheight"
## [77] "stripchart"      "strwidth"        "sunflowerplot"   "symbols"
## [81] "text"            "text.default"    "title"           "xinch"
## [85] "xspline"         "xyinch"          "yinch"
```

4. Function environments are created each time a function is called.

They store the function's arguments and any objects created within the function.

```
youthspeak <- matrix(data=c("OMG","LOL","WTF","YOLO"),nrow=2,ncol=2)
youthspeak
```

```
##      [,1]  [,2]
## [1,] "OMG" "WTF"
## [2,] "LOL" "YOLO"
```

Search Path

1. The search path is the order in which R searches for objects when a name is referenced.It starts with the global environment,followed by package environ- ments in the order they were loaded,and finally,the base package environment.

```
search()
```

```
## [1] ".GlobalEnv"        "package:stats"     "package:graphics"
## [4] "package:grDevices" "package:utils"     "package:datasets"
## [7] "package:methods"   "Autoloads"         "package:base"
```

Reserved and Protected Names

1. R has reserved and protected names that cannot be used as object names. These names are necessary to protect fundamental operations and data types.
2. Reserved names include keywords like if, else, for, while, function, TRUE, FALSE, Inf, NA, and NULL.

```
baz <- seq(from=0,to=3,length.out=5)
baz
```

```
## [1] 0.00 0.75 1.50 2.25 3.00
```

```
environment(seq)
```

```
## <environment: namespace:base>
```

Argument Matching

Argument matching is the process of assigning values to arguments when calling a function in R.There are four styles of argument matching:

1. **Exact Matching** occurs when the argument name is explicitly provided when calling the function.
2. **Partial Matching** happens when an abbreviated argument name is used, and R attempts to find a unique match.
3. **Positional Matching** relies on the order in which arguments are provided. R assigns values to arguments based on their position in the function definition.
4. **Mixed Matching** involves using multiple matching styles within a single function call.