



دانشگاه اصفهان

دانشکده مهندسی کامپیوتر

گزارش پروژه

درس زبان‌های برنامه نویسی

## بررسی زبان R

سید محمدحسین هاشمی نصرآبادی  
بردیا جوادی

استاد : دکتر آرش شفیعی

پاییز ۱۴۰۳

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

# فهرست مطالب

۵	۱	مقدمه
۵	۱.۱	تاریخچه زبان R
۵	۲.۱	ویژگی‌های زبان R
۶	۳.۱	ابزارهای مرتبط با R
۷	۲	نحو و معنا شناسی
۷	۱.۲	نحو و معناشناسی
۷	۱.۱.۲	if - اجرای کد بر اساس شرط مشخص
۷	۲.۱.۲	else - بلوک جایگزین در صورت عدم تحقق شرط
۷	۳.۱.۲	repeat - اجرای حلقه بی‌نهایت تا زمان استفاده از break
۸	۴.۱.۲	while - اجرای حلقه تا زمانی که شرط برقرار باشد
۸	۵.۱.۲	function - تعریف توابع جدید
۸	۶.۱.۲	for - اجرای حلقه روی مجموعه‌ای از مقادیر
۸	۷.۱.۲	in - تعیین عضویت یا استفاده در حلقه
۹	۸.۱.۲	next - عبور از مرحله جاری و رفتن به مرحله بعدی حلقه
۹	۹.۱.۲	break - خروج از حلقه
۹	۱۰.۱.۲	TRUE و FALSE - مقادیر منطقی (Boolean) در R
۹	۱۱.۱.۲	NULL - نشان‌دهنده مقدار خالی یا بدون مقدار
۹	۱۲.۱.۲	NA - مقدار غیرموجود یا ناشناخته (Not Available)
۱۰	۱۳.۱.۲	Inf و -Inf - مقدار بی‌نهایت مثبت و منفی
۱۰	۱۴.۱.۲	NaN - عدد غیرقابل تعریف (Not a Number)
	۱۵.۱.۲	NA_real_، NA_integer_ و NA_character_
۱۰		- انواع خاص NA برای داده‌های مختلف
۱۱	۳	متغیرها و نوع‌های داده‌ای

## فهرست تصاویر

## فهرست کدها

۷	.....	if	۱۰.۲
۷	.....	else	۲.۲
۷	.....	repeat	۳.۲
۸	.....	while	۴.۲
۸	.....	function	۵.۲
۸	.....	for	۶.۲
۸	.....	in	۷.۲
۹	.....	next	۸.۲
۹	.....	break	۹.۲
۹	.....	TRUE, FALSE	۱۰.۲
۹	.....	NULL	۱۱.۲
۹	.....	NA	۱۲.۲
۱۰	.....	Inf, -Inf	۱۳.۲
۱۰	.....	Inf, -Inf	۱۴.۲
۱۰	.....	NA_integer_, NA_real_, NA_complex_, NA_character_	۱۵.۲

# فصل ۱

## مقدمه

### ۱.۱ تاریخچه زبان R

زبان برنامه‌نویسی R در ابتدا با هدف ساده‌سازی تحلیل‌های آماری و مدل‌سازی داده‌ها طراحی شد. این زبان که توسط راس ایهاکا و رابرت جنتلمن در دهه ۱۹۹۰ ابداع شد، به عنوان یک ابزار متن‌باز برای تحلیل داده‌های پیچیده توسعه یافت. در آن زمان، نیاز به زبانی که توانایی تحلیل آماری پیشرفته، مدل‌سازی ریاضی و تولید گراف‌های بصری را داشته باشد، به شدت احساس می‌شد. زبان R با الهام از زبان S طراحی شد و توانست مشکلاتی نظیر عدم انعطاف‌پذیری ابزارهای آماری موجود و محدودیت‌های گرافیکی آنها را برطرف کند. همچنین، متن‌باز بودن R موجب شد که جامعه‌ای پویا از کاربران و توسعه‌دهندگان حول آن شکل بگیرد، که این امر به گسترش سریع امکانات و کتابخانه‌های آن کمک کرد.

### ۲.۱ ویژگی‌های زبان R

R به دلیل قابلیت‌های منحصربه‌فرد خود در حوزه‌های مختلف کاربرد گسترده‌ای دارد. از جمله مهم‌ترین حوزه‌ها می‌توان به موارد زیر اشاره کرد:

- تحلیل داده‌ها و آمار پیشرفته: R به‌طور خاص برای تحلیل‌های آماری پیچیده و مدل‌سازی داده‌ها طراحی شده است.
- یادگیری ماشین و هوش مصنوعی: بسیاری از الگوریتم‌های یادگیری ماشین و ابزارهای هوش مصنوعی در R پیاده‌سازی شده‌اند.
- مصورسازی داده‌ها: قابلیت‌های گرافیکی پیشرفته R آن را به ابزاری مناسب برای ایجاد نمودارهای حرفه‌ای و گزارش‌های بصری تبدیل کرده است.
- زیست‌شناسی محاسباتی: R در تحلیل داده‌های زیستی، از جمله داده‌های ژنومی و پروتئومی، کاربرد فراوان دارد.
- مهندسی مالی و اقتصاد: بسیاری از مدل‌های مالی و پیش‌بینی‌های اقتصادی با استفاده از R پیاده‌سازی می‌شوند.

- تحقیقات علمی و دانشگاهی: R به عنوان یک ابزار تحقیقاتی در علوم اجتماعی، روانشناسی و بسیاری از رشته‌های علمی مورد استفاده قرار می‌گیرد. علاوه بر این، R به دلیل پشتیبانی گسترده از کتابخانه‌های تخصصی، امکان تحلیل‌های پیشرفته در حوزه‌های خاص را فراهم می‌کند.

### ۳.۱ ابزارهای مرتبط با R

زبان R اغلب با زبان Python مقایسه می‌شود، زیرا هر دو زبان ابزارهای اصلی تحلیل داده و یادگیری ماشین به شمار می‌روند. تفاوت‌های کلیدی R با زبان‌های مشابه عبارت‌اند از:

- تمرکز بر تحلیل آماری: R ابزارهایی پیشرفته برای تحلیل‌های آماری و آزمون‌های فرضیه ارائه می‌دهد، در حالی که Python در این زمینه به کتابخانه‌های شخص ثالث متکی است.
- مصورسازی داده‌ها: ابزارهایی مانند ggplot2 و lattice در R قابلیت‌های بصری‌سازی حرفه‌ای و پیچیده‌تری نسبت به کتابخانه‌های Python مانند Matplotlib و Seaborn دارند.
- یادگیری و استفاده: یادگیری R برای کسانی که با مفاهیم آماری آشنا هستند، سریع‌تر است، در حالی که Python به دلیل نحوه نگارش و انعطاف‌پذیری خود، مناسب‌تر برای پروژه‌های عمومی‌تر است.
- جامعه کاربری: جامعه کاربری R بیشتر شامل متخصصان آمار و دانشمندان داده است، در حالی که Python توسط طیف وسیع‌تری از توسعه‌دهندگان نرم‌افزار و محققان استفاده می‌شود. با وجود این تفاوت‌ها، R و Python اغلب به صورت مکمل استفاده می‌شوند و ترکیب این دو زبان در پروژه‌های داده‌محور بسیار رایج است.

## فصل ۲

# نحو و معنا شناسی

### ۱.۲ نحو و معناشناسی

کلمات کلیدی زبان R شامل موارد زیر است و نمی‌توان از آن‌ها به عنوان نام متغیر یا تابع استفاده کرد: NA, NULL, FALSE, TRUE, break, next, in, for, function, while, repeat, else, if, NA\_character\_ و NA\_complex\_, NA\_real\_, NA\_integer\_, NaN, Inf  
توضیح کلمات کلیدی:

#### ۱.۱.۲ if - اجرای کد بر اساس شرط مشخص

Listing 2.1: if

```
if (x > 0) {  
  print("is positive")  
}
```

#### ۲.۱.۲ else - بلوک جایگزین در صورت عدم تحقق شرط

Listing 2.2: else

```
if (x > 0) {  
  print("is positive")  
} else {  
  print("is negative or zero")  
}
```

#### ۳.۱.۲ repeat - اجرای حلقه بی‌نهایت تا زمان استفاده از break

Listing 2.3: repeat



```
i <- 1
repeat {
  print(i)
  if (i == 5) break
  i <- i + 1
}
```

۴.۱.۲ while - اجرای حلقه تا زمانی که شرط برقرار باشد

Listing 2.4: while

```
i <- 1
while (i <= 5) {
  print(i)
  i <- i + 1
}
```

۵.۱.۲ function - تعریف توابع جدید

Listing 2.5: function

```
my_function <- function(a, b) {
  return(a + b)
}
print(my_function(3, 4))
```

۶.۱.۲ for - اجرای حلقه روی مجموعه‌ای از مقادیر

Listing 2.6: for

```
for (i in 1:5) {
  print(i)
}
```

۷.۱.۲ in - تعیین عضویت یا استفاده در حلقه

Listing 2.7: in

```
x <- 5
print(x %in% c(3, 5, 7)) # output: TRUE
```

## ۸.۱.۲ next - عبور از مرحله جاری و رفتن به مرحله بعدی حلقه

Listing 2.8: next

```
for (i in 1:5) {  
  if (i == 3) next  
  print(i)  
}
```

## ۹.۱.۲ break - خروج از حلقه

Listing 2.9: break

```
for (i in 1:5) {  
  if (i == 3) break  
  print(i)  
}
```

## ۱۰.۱.۲ TRUE و FALSE - مقادیر منطقی (Boolean) در R

Listing 2.10: TRUE, FALSE

```
x <- TRUE  
y <- FALSE  
print(x & y) # output: FALSE
```

## ۱۱.۱.۲ NULL - نشان‌دهنده مقدار خالی یا بدون مقدار

Listing 2.11: NULL

```
x <- NULL  
print(is.null(x)) # output: TRUE
```

## ۱۲.۱.۲ NA - مقدار غیرموجود یا ناشناخته (Not Available)

Listing 2.12: NA

```
x <- c(1, NA, 3)  
print(is.na(x)) # output: FALSE TRUE FALSE
```

۱۳.۱.۲ Inf و -Inf - مقدار بی‌نهایت مثبت و منفی

Listing 2.13: Inf, -Inf

```
print(1 / 0) # output: Inf  
print(-1 / 0) # output: -Inf
```

۱۴.۱.۲ NaN - عدد غیرقابل تعریف (Not a Number)

Listing 2.14: NaN

```
print(0 / 0) # output: NaN
```

۱۵.۱.۲ NA\_integer\_, NA\_real\_, NA\_complex\_ و NA\_character\_ - انواع خاص NA برای داده‌های مختلف

Listing 2.15: NA\_integer\_, NA\_real\_, NA\_complex\_, NA\_character\_

```
x <- NA_integer_  
print(typeof(x)) # output: integer
```

## فصل ۳

### متغیرها و نوع های داده ای