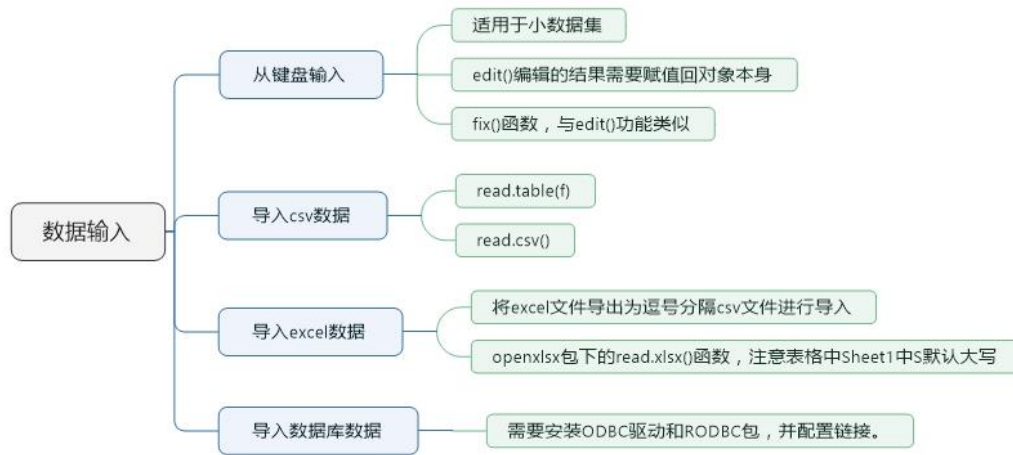


R 语言中数据的导入导出及控制语句

一、数据的导入导出



#从键盘输入

Console ~/myprojects/project/three/io/ ↗

```
> mydata<-data.frame(age=integer(3),gender=character(3),weight=numeric(3))
> mydata<-edit(mydata)
> fix(mydata)
> mydata
```

数据编辑器

	age	gender	weight	var4	var5	var6	var7
1	23	M	134				
2	24	F	110				
3	22	M	106				

> #csv 文件导入

```
> data1<-read.table("test.csv",head=TRUE,sep=",")
> data1
```

```
地区 A 企业 B 企业 C 企业
1 北京 6886 7861 9353
2 天津 3698 4344 5050
```

```
> data2<-read.csv("C:/Users/lmq/Documents/test.csv")
> head(data2)
```

```
地区 A 企业 B 企业 C 企业
1 北京 6886 7861 9353
2 天津 3698 4344 5050
```

> #excel 文件的导入

```
> data_EX<-read.xlsx(file.choose(),"Sheet1")
```

```
> data_EX
```

	地区	A 企业	B 企业	C 企业
1	北京	6886.310	7861.04	9353.32
2	天津	3697.620	4344.27	5050.40
3	河北	10096.110	11515.76	13709.50

```
> mydata<-data.frame(age=integer(3),gender=character(3),weight=numeric(3))
```

```
> mydata<-edit(mydata)
```

```
> names(data_EX)<-c("province","A","B","C")
```

```
> newdata<-transform(data_EX,sum=data_EX$A+data_EX$B+data_EX$C)
```

```
#将新数据导出
```

```
> write.table(newdata,file="newdata1.csv",sep=" ",row.names=FALSE)
```

```
> newdata<-read.csv("newdata1.csv",sep=" ",header = TRUE)
```

```
> newdata
```

	province	A	B	C	sum
1	北京	6886.310	7861.04	9353.32	24100.67
2	天津	3697.620	4344.27	5050.40	13092.29

```
#从 mysql 数据库导入
```

```
> myconn<-odbcConnect("mySQL_data",uid="root",pwd="root")
```

```
> sqlTables(myconn)#显示 myconn 中的所有表
```

	TABLE_CAT	TABLE_SCHEM	TABLE_NAME	TABLE_TYPE	REMARKS
1	test		cpi_table	TABLE	
2	test		people_table	TABLE	

```
> mytable<-sqlFetch(myconn,"cpi_table")
```

```
> head(mytable)
```

	id	province	month	cpi
1	1	安徽	1 月	104.7
2	2	安徽	2 月	102.9
3	3	安徽	3 月	103.3
4	4	安徽	4 月	102.8
5	5	安徽	5 月	102.5
6	6	安徽	6 月	101.4

```
#另外一种导入方式，可以直接写 SQL 语句
```

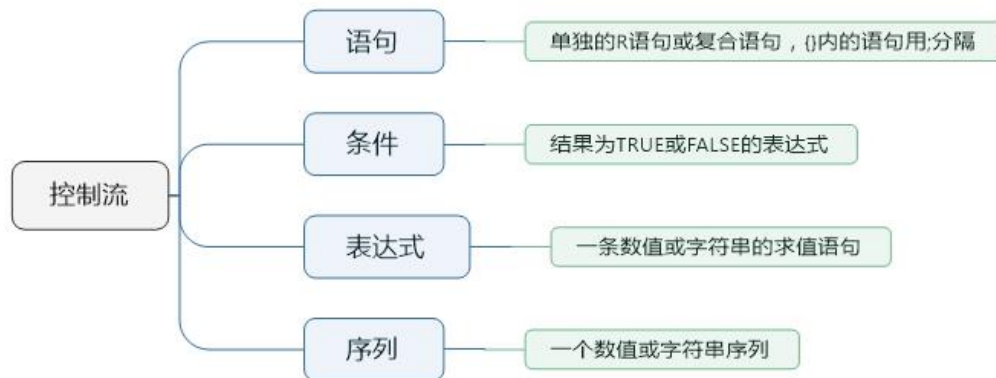
```
> people<-sqlQuery(myconn,"select * from people_table")
```

```
> head(people)
```

	ID	PROVINCE	YEAR	PEOPLE
1	1	全国	2005 年	130756
2	2	全国	2006 年	131448
3	3	全国	2007 年	132129
4	4	全国	2008 年	132802

5 5 全国 2009 年 133450
6 6 全国 2010 年 134091

二、控制流语句



- (1) 重复和循环: for 语句和 while 语句
- (2) 条件执行: if-else、ifelse、switch 结构
- (3) 自编函数: myfunction<- function (arg1,arg2,...) {statements return()}先定义函数，然后调用。

#for 循环

```
> for(i in 1:3)
+   print("Hello world")
[1] "Hello world"
[1] "Hello world"
[1] "Hello world"
```

```
> #while
> i<-3
> while (i>1)
+   {print("Hello");
+   i<-i-1}
[1] "Hello"
[1] "Hello"
```

If-else 结构

```
> if(class(1)=="integer")
+   print("Hello")
> else
```

```
Error: unexpected 'else' in "else"
```

```
> print(0)
```

```
[1] 0
```

报错的原因：else 要跟在上一行那个大括号}的后面，不能另起一行。R 因为不强制要求表达式末尾加分号，因此会以行尾作为标志来判断表达式是否结束。在 if/else 里，由于执行 if TRUE 的表达式体到大括号为止就结束了，并且是合法的表达式，那么 R 就会将 else 上方的部分作为完整的表达式单独执行；执行完成后再读入下一句 else，但是 else 自身当然不能作为一个表达式的开头，因此就会出现执行错误。以下代码便不再报错。

```
> score<-98
```

```
> if(score>90)
```

```
+ {print("excellent")}else
```

```
+ {print("normal")}
```

```
[1] "excellent"
```

```
#ifelse 语句
```

```
> x<-c(2,4,3,6,7)
```

```
> y<-c(3,2,4,4,8)
```

```
> ifelse(x>y,x,y)
```

```
[1] 3 4 4 6 8
```

```
#switch
```

```
> score<-c(91,83,67,98)
```

```
> for(i in score)
```

```
+ switch(i,
```

```
+       i>=60 & i<80 ,print("中"),
```

```
+       i>=80 & i<90 ,print("良"),
```

```
+       i>=90 & i<=100 ,print("优")
```

```
+ )
```

上面的语句没有输出结果，虽然没有报错，但是 R 语言中的 switch 不能这么用，具体为什么还不了解。以下是正确的用法：

```
> feelings <- c("sad", "afraid")
```

```
> for (i in feelings)
```

```
+ print(
```

```
+   switch(i,
```

```
+       happy = "I am glad you are happy",
```

```
+       afraid = "There is nothing to fear",
```

```
+       sad = "Cheer up",
```

```
+       angry = "Calm down now"
```

```
+ )
```

```
+ )  
[1] "Cheer up"  
[1] "There is nothing to fear"
```

#自编函数及调用

```
> mystats <- function(x, parametric=TRUE, print=FALSE) {  
+   if (parametric) {  
+     center <- mean(x); spread <- sd(x)  
+   } else {  
+     center <- median(x); spread <- mad(x)  
+   }  
+   if (print & parametric) {  
+     cat("Mean=", center, "\n", "SD=", spread, "\n")  
+   } else if (print & !parametric) {  
+     cat("Median=", center, "\n", "MAD=", spread, "\n")  
+   }  
+   result <- list(center=center, spread=spread)  
+   return(result)  
+ }  
> mystats(c(3,4,6,3,4))  
$center  
[1] 4  
  
$spread  
[1] 1.224745
```