实验报告：

数据导入及基本数据处理

（2024-2025 学年第一学期）

学　　院　经济学院

课程名称　R 语言编程基础与金融数据分析

班　　级　22 金融 03 班

学　　号　24210819

姓　　名　周生瑞

任课教师　王皓

撰写日期　2024 年 10 月 13 日

代码地址：https://sr6688.github.io/

目录

实验内容 3
实验一 5
实验二 7
实验三 10
实验思考与经验总结 12
实验思考 12
经验总结 12

图表索引

图 1 读取当前系统日期的三个函数结果 5
图 2 读取系统时间函数结果的变量类型 5
图 3 将“日期”类型转换为指定格式字符串类型 6
图 4 设置工作空间目录 7
图 5 创建指定元素向量并进行元素查询 7
图 6 使用 seq()函数以及 rep()函数创建指定元素向量 7
图 7 创建重复元素因子序列 8
图 8 使用指定向量创建矩阵 8
图 9 使用指定向量创建数据框并更改列名 9
图 10 将数据框转换为 txt 文件并存入指定路径 9
图 11 读取指定路径 txt 文件为数据框并与已有数据框指定元素合并 10
图 12 将数据框合并转换为 csv 文件存入指定路径 11
图 13 问题导向 R 编程的数据存储与结构化处理 13

实验内容

- 实验目的
  - 了解 R 语言中数据类型的判别及转换函数，及其应用方法。
  - 了解 R 语言中对数据结构操作的函数，及其应用方法。
  - 了解 R 语言中读写数据文件的方法。
- 实验内容
  - 掌握读取日期时间值、数据类型的判别方法及转换函数。
  - 掌握不同数据结构的构建方式和转换函数。
  - 掌握数据文件读写的函数。
- 实验方法与步骤

1. 实验一：读取系统日期时间，进行变量类型转换，对转换前后的变量类型进行辨别对比。
   a) 使用读取系统当前日期时间的 3 个函数：Sys.Date()、Sys.time()、date()。
   b) 使用类型辨别函数 class()判断读取的 3 个不同的结果的类型。
   c) 将读取到的日期时间值转换为另外一种数据类型：将年月日格式的日期时间值转化为月日年格式的字符串。
   d) 判断转换后的结果的类型，判定是否转换成功。
2. 实验二：创建多种数据结构，并进行数据结构的转换、索引、扩展等编辑操作。
   a) 设置工作空间目录。
   b) 创建一个向量 x，内含元素为序列：5.1 4.9 4.7 4.6 5.0 3.5 3.0 3.2 3.1 3.6。
   c) 查询向量 x 中序号为 3，6，9 的元素，查询向量 x 中大于 4.0 且小于等于 5.0 的元素的位置。
   d) 创建一个向量 Petal.Length，内含等差序列：首位为 1.7，等差为 0.1，长度为 5。
   e) 创建一个向量 Petal.Width，内含重复序列：重复 0.2 五次。
   f) 创建一个向量为重复因子序列 Species：水平数为 3，各水平重复 2 次，序列长度为 5，三个水平为：setosa、versicolor、virginica。
   g) 创建一个 5 行 2 列的矩阵，元素为向量 x，按列填充。

h) 将矩阵写入数据框 data_iris，更改列名为：Sepal.Length、Sepal.Width。

i) 将向量 Petal.Length、Petal.Width、Species 按列合并至数据框 data_iris 中。

j) 将数据框 data_iris 保存为 txt 文件，保存到工作空间的 test 目录下。

3. 实验三：读取 txt 文件，进行编辑操作，在写入另外一个 csv 文件中。

a) 读取实验二保存在 test 目录下的 txt 文件 data_iris。

b) 将 R 的示例数据集 iris 中的第 6~10 行写入数据框 data_isir1 中。

c) 将数据框 data_iris 与 data_iris1 合并为数据框 data_iris2，并保存为 csv 文件在同目录下。

- 思考与实验总结
  - 不同的数据结构之间是如何转换的？
  - 如果读取的数据中出现乱码，如何处理？

实验一

- 使用读取系统当前日期时间的 3 个函数：Sys.Date()、Sys.time()、date()。

a=Sys.Date()#获取系统日期

b=Sys.time()#获取系统时间

c=date()#获取系统时间

a

b

c

```
[1] "2024-10-13"
> b
[1] "2024-10-13 14:30:51 CST"
> c
[1] "Sun Oct 13 14:30:52 2024"
```

图 1 读取当前系统日期的三个函数结果

- 使用类型辨别函数 class()判断读取的 3 个不同的结果的类型。

class(a)#判断 a 的数据类型

class(b)#判断 b 的数据类型

class(c)#判断 c 的数据类型

```
> class(a)#判断a的数据类型
[1] "Date"
> class(b)#判断b的数据类型
[1] "POSIXct" "POSIXt"
> class(c)#判断c的数据类型
[1] "character"
```

图 2 读取系统时间函数结果的变量类型

- 将读取到的日期时间值转换为另外一种的数据类型：将年月日格式的日期时间值转化为月日年格式的字符串。
- 判断转换后的结果的类型，判定是否转换成功。

d=format(a,format="%B %d %Y")#将年月日日期格式转化为月日年格式的字符串

d

class(d)#判断是否转换成功

```
> d=format(a,format="%B %d %Y")#将年月日日期格式转化为月日年格式的字符串
> d
[1] "October 13 2024"
> class(d)#判断是否转换成功
[1] "character"
```

图 3 将“日期”类型转换为指定格式字符串类型

实验二

- 设置工作空间目录。

getwd()#获取当前空间工作目录

setwd("/Users/apple/Desktop/test_r")#设置工作空间目录

```
> getwd()#获取当前空间工作目录
[1] "/Users/apple/Desktop"
> setwd("/Users/apple/Desktop/test_r")#设置工作空间目录
```

图 4 设置工作空间目录

- 创建一个向量 x，内含元素为序列：5.1 4.9 4.7 4.6 5.0 3.5 3.0 3.2 3.1 3.6。
- 查询向量 x 中序号为 3，6，9 的元素，查询向量 x 中大于 4.0 且小于等于 5.0 的元素的位置。

x=c(5.1,4.9,4.7,4.6,5.0,3.5,3.0,3.2,3.1,3.6)#建立向量 x

x[c(3,6,9)]#查询向量 x 中 3，6，9 号元素

which(x>4.0&x<=5.0)#查询向量 x 中大于 4 小于等于 5 元素的位置

```
> x=c(5.1,4.9,4.7,4.6,5.0,3.5,3.0,3.2,3.1,3.6)#建立向量x
> x[c(3,6,9)]#查询向量x中3，6，9号元素
[1] 4.7 3.5 3.1
> which(x>4.0&x<=5.0)#查询向量x中大于4小于等于5元素的位置
[1] 2 3 4 5
```

图 5 创建指定元素向量并进行元素查询

- 创建一个向量 Petal.Length，内含等差序列：首位为 1.7，等差为 0.1，长度为 5。
- 创建一个向量 Petal.Width，内含重复序列：重复 0.2 五次。

Petal.Length=seq(1.7,0.1,length.out=5)

#建立向量 Petal.Length：首项为 1.7，等差为 0.1 长度为 5 的等差数列

Petal.Width=rep(0.2,5)#建立向量 Petal.Width：重复 0.2 五次

Petal.Length	num [1:5] 1.7 1.3 0.9 0.5 0.1
Petal.Width	num [1:5] 0.2 0.2 0.2 0.2 0.2

图 6 使用 seq()函数以及 rep()函数创建指定元素向量

- 创建一个向量为重复因子的序列 Species：水平数为 3，各水平重复 2 次，序列长度为 5，三个水平为：setosa、versicolor、virginica。

Species=rep(c("setosa","versicolor","virginica"),2)#建立 Species 向量

factor(Species)#Species 向量因子化

Species=factor(Species)[1:5]#提取 Species 向量前五项

```
> Species=rep(c("setosa","versicolor","virginica"),2)#建立Species向量
> factor(Species)#Species向量因子化
[1] setosa     versicolor virginica  setosa
[5] versicolor virginica
Levels: setosa versicolor virginica
> Species=factor(Species)[1:5]#提取Species向量前五项
> Species
[1] setosa     versicolor virginica  setosa     versicolor
Levels: setosa versicolor virginica
```

图 7 创建重复元素因子序列

- 创建一个 5 行 2 列的矩阵，元素为向量 x，按列填充。

> M=matrix(x,nrow = 5)#使用向量 x 建立 M 矩阵

```
> M=matrix(x,nrow = 5)#建立M矩阵
> M
     [,1] [,2]
[1,]  5.1  3.5
[2,]  4.9  3.0
[3,]  4.7  3.2
[4,]  4.6  3.1
[5,]  5.0  3.6
```

图 8 使用指定向量创建矩阵

- 将矩阵写入数据框 data_iris，更改列名为：Sepal.Length、Sepal.Width。

data_iris=data.frame(M)#建立数据框 data_iris

colnames(data_iris)=c("Sepal.Length","Sepal.Width")#重新命名 data_iris 列名

```
> data_iris=data.frame(M)#建立数据框data_iris
> colnames(data_iris)=c("Sepal.Length","Sepal.Width")#重新命名data_iris列名
> data_iris
  Sepal.Length Sepal.Width
1          5.1         3.5
2          4.9         3.0
3          4.7         3.2
4          4.6         3.1
5          5.0         3.6
```

图 9 使用指定向量创建数据框并更改列名

- 将向量 Petal.Length、Petal.Width、Species 按列合并至数据框 data_iris 中。
- 将数据框 data_iris 保存为 txt 文件，保存到工作空间的 test 目录下。

```
data_iris=cbind(data_iris,Petal.Length,Petal.Width,Species)
#将三个列合并存入 data_iris 数据框中
write.table(data_iris,file = "/Users/apple/Desktop/test_r/test.txt")
#将 data_iris 以 test 名称，txt 文件形式存入工作目录中
```

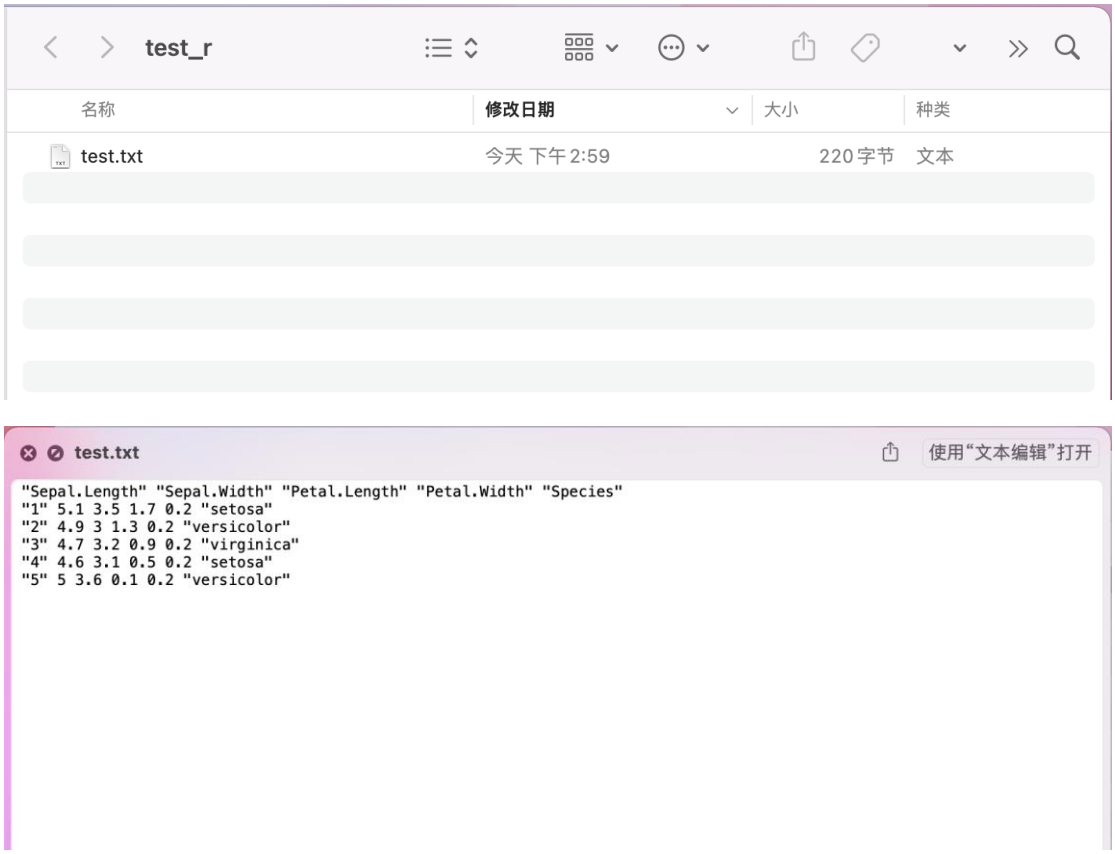

图 10 将数据框转换为 txt 文件并存入指定路径

实验三

- 读取实验二保存在 test 目录下的 txt 文件 data_iris。
- 将 R 的示例数据集 iris 中的第 6~10 行写入数据框 data_isir1 中。

```
read.table("/Users/apple/Desktop/test_r/test.txt")#读取 txt 文件 data_iris
data_iris1=data.frame(iris[6:10,])
#将 R 的示例数据集 iris 中的第 6~10 行写入数据框 data_isir1 中
```

```
> read.table("/Users/apple/Desktop/test_r/test.txt")#读取txt文件data_iris
   Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1           5.1         3.5          1.7         0.2     setosa
2           4.9         3.0          1.3         0.2 versicolor
3           4.7         3.2          0.9         0.2  virginica
4           4.6         3.1          0.5         0.2     setosa
5           5.0         3.6          0.1         0.2 versicolor
> data_iris1=data.frame(iris[6:10,])#将R的示例数据集iris中的第6~10行写入数据框data_isir1中
> data_iris1
   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
6           5.4         3.9          1.7         0.4  setosa
7           4.6         3.4          1.4         0.3  setosa
8           5.0         3.4          1.5         0.2  setosa
9           4.4         2.9          1.4         0.2  setosa
10          4.9         3.1          1.5         0.1  setosa
```

图 11 读取指定路径 txt 文件为数据框并与已有数据框指定元素合并

- 将数据框 data_iris 与 data_iris1 合并为数据框 data_iris2，并保存为 csv 文件在同目录下。

```
data_iris2=data.frame(rbind(data_iris1,data_iris))
#将数据框 data_iris 与 data_iris1 合并为数据框 data_iris2
write.csv(data_iris2,file = "/Users/apple/Desktop/test_r/test.csv")
#保存为 csv 文件在同目录下
```

```
> data_iris2=data.frame(rbind(data_iris1,data_iris))#将数据框data_iris与data_iris1合并为数据框data_iris2
> data_iris2
   Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
6           5.4         3.9          1.7         0.4     setosa
7           4.6         3.4          1.4         0.3     setosa
8           5.0         3.4          1.5         0.2     setosa
9           4.4         2.9          1.4         0.2     setosa
10          4.9         3.1          1.5         0.1     setosa
1           5.1         3.5          1.7         0.2     setosa
2           4.9         3.0          1.3         0.2 versicolor
3           4.7         3.2          0.9         0.2  virginica
4           4.6         3.1          0.5         0.2     setosa
5           5.0         3.6          0.1         0.2 versicolor
> write.csv(data_iris2,file = "/Users/apple/Desktop/test_r/test.csv")#保存为csv文件在同目录下
```

test.csv 使用 WPS Office 打开

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
1	5.1	3.5	1.7	0.2	setosa
2	4.9	3	1.3	0.2	versicolor
3	4.7	3.2	0.9	0.2	virginica
4	4.6	3.1	0.5	0.2	setosa
5	5	3.6	0.1	0.2	versicolor

图 12 将数据框合并转换为 csv 文件存入指定路径

实验思考与经验总结

实验思考

- 不同的数据结构之间是如何转换的？

在 R 语言中，数据结构之间的转换是非常灵活的。例如，我们可以通过 as.numeric()、as.character()、as.factor()、format()等函数将数据从一个类型转换为另一个类型。在实验中，我们学习了如何将日期时间值转换为字符串，以及如何将向量合并到数据框中。这些转换操作通常涉及到数据类型的改变，需要对数据结构有深入的理解。

- 如果读取的数据中出现乱码，如何处理？

如果在读取数据时出现乱码，通常是由于文件编码与 R 语言读取时使用的编码不匹配。处理乱码的方法通常包括：

i. 确认文件的编码格式，可以使用文本编辑器查看或转换文件编码。

ii. 在 R 语言中使用正确的编码参数读取文件，例如使用 read.table()函数时，可以指定 fileEncoding 参数。

iii. 如果数据是从网络或其他系统导入的，确保在传输过程中编码没有被错误地转换或损坏。

经验总结

通过本次实验，我们不仅学习了 R 语言中数据类型和结构的操作，还掌握了数据文件的读写方法。这些技能对于数据分析和处理非常重要，能够帮助我们更有效地处理和分析数据。通过实验二，我学习到了 R 语言中数据结构的多样性和灵活性。向量、矩阵和数据框是 R 中常用的数据结构，它们各有特点，适用于不同的数据处理场景。例如，向量是最基本的数据结构，适合存储单一类型的数据；矩阵适合于二维数据的存储和操作；而数据框则提供了更复杂的数据组织方式，适合存储具有不同类型数据的表格数据。掌握这些数据结构的创建、索引和转换方法，对于进行高效的数据处理至关重要。在实验三中，我实践了数据文件的读写操作。这不仅包括了如何将数据从文本文件中读取到 R

环境中，还包括了如何将处理后的数据保存回文件。这个过程让我认识到了数据文件格式的重要性，以及如何在不同的数据格式之间进行转换。例如，文本文件适合于简单的数据存储，而 CSV 文件则更适合于表格数据的交换和处理。（如图 13 所示）总之，通过本次实验，我不仅掌握了 R 语言中数据类型和结构的操作，还学会了如何高效地处理和分析数据。这些经验将为我未来的数据分析工作打下坚实的基础。

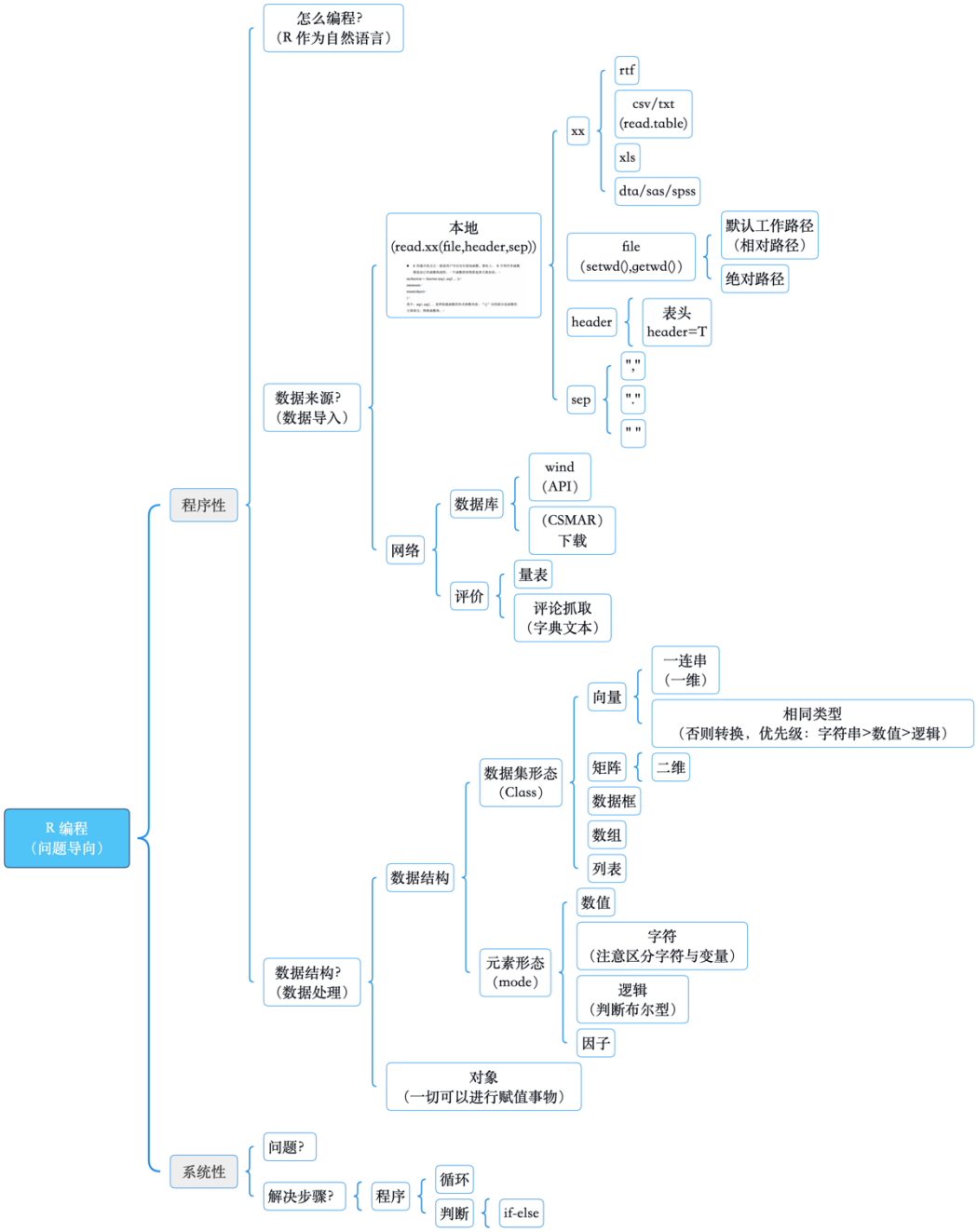

图 13 问题导向 R 编程的数据存储与结构化处理