talk03 练习与作业

目录

练习和作业说明

将相关代码填写入以 “‘{r} “‘ 标志的代码框中，运行并看到正确的结果；

完成后，用工具栏里的 “Knit” 按键生成 PDF 文档；

将生成的 PDF 改为：姓名-学号**-talk03** 作业**.pdf**，并提交到老师指定的平台/钉群。

talk03 内容回顾

- 二维表：`data.frame`, `tibble`
  - 声明
  - 操作
    - 增减行、列
    - 合并

练习与作业 1，data.frame

注：以下内容来自 https://www.r-exercises.com/。

- 生成下面的 data.frame 的前三列，之后再增加 Sex 这列

	Age	Height	Weight	Sex
Alex	25	177	57	F
Lilly	31	163	69	F
Mark	23	190	83	M
Oliver	52	179	75	M
Martha	76	163	70	F
Lucas	49	183	83	M
Caroline	26	164	53	F

```
## 先生成前三列；
row_names<-c('Alex','Lilly','Mark','Oliver','Martha','Lucas','Caroline')
Age<-c(25,31,23,52,76,49,26)
Height<-c(177,163,190,179,163,183,164)
```

```
Weight<-c(57,69,83,75,70,83,53)
df_1<-data.frame(Age=Age,Height=Height,Weight=Weight)
rownames(df_1)<-row_names
df_1
```

```
##           Age Height Weight
## Alex       25    177     57
## Lilly      31    163     69
## Mark       23    190     83
## Oliver     52    179     75
## Martha     76    163     70
## Lucas      49    183     83
## Caroline   26    164     53
```

```
## 再插入第四列
Sex<-c('F','F','M','M','F','M','F')
df_1<-cbind(df_1,Sex)
## 显示最终结果
df_1
```

```
##           Age Height Weight Sex
## Alex       25    177     57   F
## Lilly      31    163     69   F
## Mark       23    190     83   M
## Oliver     52    179     75   M
## Martha     76    163     70   F
## Lucas      49    183     83   M
## Caroline   26    164     53   F
```

- 生成以下 data.frame，确保 Working 这列的类型是 character，而不是 factor

	Working
Alex	Yes
Lilly	No
Mark	No
Oliver	Yes
Martha	Yes
Lucas	No
Caroline	Yes

```
## 生成 data.frame
row_names<-c('Alex','Lilly','Mark','Oliver','Martha','Lucas','Caroline')
Working<-c('Yes','No','No','Yes','Yes','No','Yes')
df_2<-data.frame(Working=Working)
row.names(df_2)=row_names
## 显示结果
df_2
```

```
##          Working
## Alex         Yes
## Lilly         No
## Mark          No
## Oliver       Yes
## Martha       Yes
## Lucas         No
## Caroline     Yes
```

```
## 显示 Working 列的性质
cat('Working 这一列的性质为: ',class(df_2$Working))
```

```
## Working这一列的性质为:  character
```

- 检查系统自带变量 state.center 的内容，将其转化为 data.frame

```
## 代码写这里，并运行;
state.center
```

```
## $x
##  [1]  -86.7509 -127.2500 -111.6250  -92.2992 -119.7730 -105.5130  -72.3573
##  [8]  -74.9841  -81.6850  -83.3736 -126.2500 -113.9300  -89.3776  -86.0808
## [15]  -93.3714  -98.1156  -84.7674  -92.2724  -68.9801  -76.6459  -71.5800
## [22]  -84.6870  -94.6043  -89.8065  -92.5137 -109.3200  -99.5898 -116.8510
## [29]  -71.3924  -74.2336 -105.9420  -75.1449  -78.4686 -100.0990  -82.5963
## [36]  -97.1239 -120.0680  -77.4500  -71.1244  -80.5056  -99.7238  -86.4560
## [43]  -98.7857 -111.3300  -72.5450  -78.2005 -119.7460  -80.6665  -89.9941
## [50] -107.2560
##
## $y
##  [1] 32.5901 49.2500 34.2192 34.7336 36.5341 38.6777 41.5928 38.6777 27.8744
## [10] 32.3329 31.7500 43.5648 40.0495 40.0495 41.9358 38.4204 37.3915 30.6181
## [19] 45.6226 39.2778 42.3645 43.1361 46.3943 32.6758 38.3347 46.8230 41.3356
## [28] 39.1063 43.3934 39.9637 34.4764 43.1361 35.4195 47.2517 40.2210 35.5053
## [37] 43.9078 40.9069 41.5928 33.6190 44.3365 35.6767 31.3897 39.1063 44.2508
## [46] 37.5630 47.4231 38.4204 44.5937 43.0504
```

```
str(state.center)
```

```
## List of 2
##  $ x: num [1:50] -86.8 -127.2 -111.6 -92.3 -119.8 ...
##  $ y: num [1:50] 32.6 49.2 34.2 34.7 36.5 ...
```

```
(as.data.frame(state.center))
```

```
##            x       y
## 1   -86.7509 32.5901
## 2  -127.2500 49.2500
## 3  -111.6250 34.2192
## 4   -92.2992 34.7336
## 5  -119.7730 36.5341
## 6  -105.5130 38.6777
## 7   -72.3573 41.5928
## 8   -74.9841 38.6777
## 9   -81.6850 27.8744
## 10  -83.3736 32.3329
## 11 -126.2500 31.7500
## 12 -113.9300 43.5648
## 13  -89.3776 40.0495
## 14  -86.0808 40.0495
## 15  -93.3714 41.9358
## 16  -98.1156 38.4204
## 17  -84.7674 37.3915
## 18  -92.2724 30.6181
## 19  -68.9801 45.6226
## 20  -76.6459 39.2778
## 21  -71.5800 42.3645
## 22  -84.6870 43.1361
## 23  -94.6043 46.3943
## 24  -89.8065 32.6758
## 25  -92.5137 38.3347
## 26 -109.3200 46.8230
## 27  -99.5898 41.3356
## 28 -116.8510 39.1063
## 29  -71.3924 43.3934
## 30  -74.2336 39.9637
```

```
## 31 -105.9420 34.4764
## 32  -75.1449 43.1361
## 33  -78.4686 35.4195
## 34 -100.0990 47.2517
## 35  -82.5963 40.2210
## 36  -97.1239 35.5053
## 37 -120.0680 43.9078
## 38  -77.4500 40.9069
## 39  -71.1244 41.5928
## 40  -80.5056 33.6190
## 41  -99.7238 44.3365
## 42  -86.4560 35.6767
## 43  -98.7857 31.3897
## 44 -111.3300 39.1063
## 45  -72.5450 44.2508
## 46  -78.2005 37.5630
## 47 -119.7460 47.4231
## 48  -80.6665 38.4204
## 49  -89.9941 44.5937
## 50 -107.2560 43.0504
```

- **生成一个 50 行 * 5 列的 matrix，将其行名改为：row_i 格式，其中 i 为当前的行号，比如 row_1, row_2 等**

```
## 代码写这里，并运行；
matrix_1<-matrix(1:250, 50, 5)
row_names=NULL
for (i in 1:50){
  row_names<-c(row_names,paste('row_',as.character(i),sep = ''))
}
row.names(matrix_1)<-row_names
matrix_1
```

```
##        [,1] [,2] [,3] [,4] [,5]
## row_1     1   51  101  151  201
## row_2     2   52  102  152  202
## row_3     3   53  103  153  203
## row_4     4   54  104  154  204
## row_5     5   55  105  155  205
## row_6     6   56  106  156  206
## row_7     7   57  107  157  207
## row_8     8   58  108  158  208
## row_9     9   59  109  159  209
## row_10   10   60  110  160  210
## row_11   11   61  111  161  211
## row_12   12   62  112  162  212
## row_13   13   63  113  163  213
## row_14   14   64  114  164  214
## row_15   15   65  115  165  215
## row_16   16   66  116  166  216
## row_17   17   67  117  167  217
## row_18   18   68  118  168  218
## row_19   19   69  119  169  219
## row_20   20   70  120  170  220
## row_21   21   71  121  171  221
## row_22   22   72  122  172  222
## row_23   23   73  123  173  223
## row_24   24   74  124  174  224
## row_25   25   75  125  175  225
## row_26   26   76  126  176  226
## row_27   27   77  127  177  227
## row_28   28   78  128  178  228
## row_29   29   79  129  179  229
## row_30   30   80  130  180  230
## row_31   31   81  131  181  231
## row_32   32   82  132  182  232
```

```
## row_33   33   83  133  183  233
## row_34   34   84  134  184  234
## row_35   35   85  135  185  235
## row_36   36   86  136  186  236
## row_37   37   87  137  187  237
## row_38   38   88  138  188  238
## row_39   39   89  139  189  239
## row_40   40   90  140  190  240
## row_41   41   91  141  191  241
## row_42   42   92  142  192  242
## row_43   43   93  143  193  243
## row_44   44   94  144  194  244
## row_45   45   95  145  195  245
## row_46   46   96  146  196  246
## row_47   47   97  147  197  247
## row_48   48   98  148  198  248
## row_49   49   99  149  199  249
## row_50   50  100  150  200  250
```

- **使用系统自带变量 VADeaths，做如下练习：**
- 检查 VADeaths 的类型，如果不是 data.frame，则转换之；
- 添加新的一列，取名 Total，其值每行的总合
- 调整列的顺序，将 Total 变为第一列。

```
## 代码写这里，并运行；
class(VADeaths)
```

```
## [1] "matrix" "array"
```

```
if(class(VADeaths)[1]!='data.frame')
{
  cat('VADeaths 的类型不是 data.frame, 现进行转换')
  new_VADeaths<-as.data.frame(VADeaths)
}else
{
  new_VADeaths<-VADeaths
}
```

```
## VADeaths 的类型不是data.frame, 现进行转换
```

```
new_VADeaths
```

```
##       Rural Male Rural Female Urban Male Urban Female
## 50-54       11.7          8.7       15.4          8.4
## 55-59       18.1         11.7       24.3         13.6
## 60-64       26.9         20.3       37.0         19.3
## 65-69       41.0         30.9       54.6         35.1
## 70-74       66.0         54.3       71.1         50.0
```

```
Total<-apply(new_VADeaths,1,sum)
cat('\n行和为: ',Total)
```

```
##
## 行和为:  44.2 67.7 103.5 161.6 241.4
```

```
new_VADeaths<-cbind(Total,new_VADeaths)
new_VADeaths
```

```
##       Total Rural Male Rural Female Urban Male Urban Female
## 50-54  44.2       11.7          8.7       15.4          8.4
## 55-59  67.7       18.1         11.7       24.3         13.6
## 60-64 103.5       26.9         20.3       37.0         19.3
```

```
## 65-69 161.6       41.0          30.9       54.6          35.1
## 70-74 241.4       66.0          54.3       71.1          50.0
```

- **用系统自带的 swiss 数据做练习：**
- 取子集，选取第 1, 2, 3, 10, 11, 12 and 13 行，第 Examination, Education 和 Infant.Mortality 列；
- 将 Sarine 行 Infant.Mortality 列的值改为 NA；
- 增加一列，命名为 Mean，其值为当前行的平均值；

```
## 代码写这里，并运行；
cat('原始 swiss 如下\n')
```

```
## 原始swiss如下
```

```
swiss
```

```
##              Fertility Agriculture Examination Education Catholic
## Courtelary        80.2        17.0          15        12     9.96
## Delemont          83.1        45.1           6         9    84.84
## Franches-Mnt      92.5        39.7           5         5    93.40
## Moutier           85.8        36.5          12         7    33.77
## Neuveville        76.9        43.5          17        15     5.16
## Porrentruy        76.1        35.3           9         7    90.57
## Broye             83.8        70.2          16         7    92.85
## Glane             92.4        67.8          14         8    97.16
## Gruyere           82.4        53.3          12         7    97.67
## Sarine            82.9        45.2          16        13    91.38
## Veveyse           87.1        64.5          14         6    98.61
## Aigle             64.1        62.0          21        12     8.52
## Aubonne           66.9        67.5          14         7     2.27
```

```
## Avenches          68.9        60.7          19          12      4.43
## Cossonay          61.7        69.3          22           5      2.82
## Echallens         68.3        72.6          18           2     24.20
## Grandson          71.7        34.0          17           8      3.30
## Lausanne          55.7        19.4          26          28     12.11
## La Vallee         54.3        15.2          31          20      2.15
## Lavaux            65.1        73.0          19           9      2.84
## Morges            65.5        59.8          22          10      5.23
## Moudon            65.0        55.1          14           3      4.52
## Nyone             56.6        50.9          22          12     15.14
## Orbe              57.4        54.1          20           6      4.20
## Oron              72.5        71.2          12           1      2.40
## Payerne           74.2        58.1          14           8      5.23
## Paysd'enhaut      72.0        63.5           6           3      2.56
## Rolle             60.5        60.8          16          10      7.72
## Vevey             58.3        26.8          25          19     18.46
## Yverdon           65.4        49.5          15           8      6.10
## Conthey           75.5        85.9           3           2     99.71
## Entremont         69.3        84.9           7           6     99.68
## Herens            77.3        89.7           5           2    100.00
## Martigwy          70.5        78.2          12           6     98.96
## Monthey           79.4        64.9           7           3     98.22
## St Maurice        65.0        75.9           9           9     99.06
## Sierre            92.2        84.6           3           3     99.46
## Sion              79.3        63.1          13          13     96.83
## Boudry            70.4        38.4          26          12      5.62
## La Chauxdfnd      65.7         7.7          29          11     13.79
## Le Locle          72.7        16.7          22          13     11.22
## Neuchatel         64.4        17.6          35          32     16.92
## Val de Ruz        77.6        37.6          15           7      4.97
## ValdeTravers      67.6        18.7          25           7      8.65
## V. De Geneve      35.0         1.2          37          53     42.34
## Rive Droite       44.7        46.6          16          29     50.43
```

```
## Rive Gauche       42.8        27.7          22        29    58.33
##              Infant.Mortality
## Courtelary               22.2
## Delemont                 22.2
## Franches-Mnt             20.2
## Moutier                  20.3
## Neuveville               20.6
## Porrentruy               26.6
## Broye                    23.6
## Glane                    24.9
## Gruyere                  21.0
## Sarine                   24.4
## Veveyse                  24.5
## Aigle                    16.5
## Aubonne                  19.1
## Avenches                 22.7
## Cossonay                 18.7
## Echallens                21.2
## Grandson                 20.0
## Lausanne                 20.2
## La Vallee                10.8
## Lavaux                   20.0
## Morges                   18.0
## Moudon                   22.4
## Nyone                    16.7
## Orbe                     15.3
## Oron                     21.0
## Payerne                  23.8
## Paysd'enhaut             18.0
## Rolle                    16.3
## Vevey                    20.9
## Yverdon                  22.5
## Conthey                  15.1
```

```
## Entremont                19.8
## Herens                   18.3
## Martigwy                 19.4
## Monthey                  20.2
## St Maurice               17.8
## Sierre                   16.3
## Sion                     18.1
## Boudry                   20.3
## La Chauxdfnd             20.5
## Le Locle                 18.9
## Neuchatel                23.0
## Val de Ruz               20.0
## ValdeTravers             19.5
## V. De Geneve             18.0
## Rive Droite              18.2
## Rive Gauche              19.3
```

```
tar_row<-c(1,2,3,10,11,12,13)
tar_col<-c('Examination','Education','Infant.Mortality')
cat('取完子集后\n')
```

```
## 取完子集后
```

```
new_swiss<-swiss[tar_row,tar_col]
new_swiss
```

```
##              Examination Education Infant.Mortality
## Courtelary            15        12             22.2
## Delemont               6         9             22.2
## Franches-Mnt           5         5             20.2
## Sarine                16        13             24.4
## Veveyse               14         6             24.5
## Aigle                 21        12             16.5
## Aubonne               14         7             19.1
```

```
cat('按题意修改后: \n')
```

```
## 按题意修改后:
```

```
new_swiss['Sarine','Infant.Mortality']<-NA
new_swiss
```

```
##              Examination Education Infant.Mortality
## Courtelary            15        12             22.2
## Delemont               6         9             22.2
## Franches-Mnt           5         5             20.2
## Sarine                16        13               NA
## Veveyse               14         6             24.5
## Aigle                 21        12             16.5
## Aubonne               14         7             19.1
```

```
cat('增加一列 Mean 后')
```

```
## 增加一列Mean后
```

```
Mean<-apply(new_swiss, 1, mean)
new_swiss<-cbind(new_swiss,Mean)
new_swiss
```

```
##              Examination Education Infant.Mortality     Mean
## Courtelary            15        12             22.2 16.40000
## Delemont               6         9             22.2 12.40000
## Franches-Mnt           5         5             20.2 10.06667
## Sarine                16        13               NA       NA
## Veveyse               14         6             24.5 14.83333
## Aigle                 21        12             16.5 16.50000
## Aubonne               14         7             19.1 13.36667
```

- 将下面三个变量合并生成一个 data.frame

```
Id <- LETTERS

x <- seq(1,43,along.with=Id)

y <- seq(-20,0,along.with=Id)
```

```
## 代码写这里，并运行；
Id <- LETTERS
x <- seq(1,43,along.with=Id)
y <- seq(-20,0,along.with=Id)
df_3<-data.frame(Id=Id,x=x,y=y)
df_3
```

```
##    Id     x     y
## 1   A  1.00 -20.0
## 2   B  2.68 -19.2
## 3   C  4.36 -18.4
## 4   D  6.04 -17.6
## 5   E  7.72 -16.8
## 6   F  9.40 -16.0
## 7   G 11.08 -15.2
## 8   H 12.76 -14.4
## 9   I 14.44 -13.6
## 10  J 16.12 -12.8
## 11  K 17.80 -12.0
## 12  L 19.48 -11.2
## 13  M 21.16 -10.4
## 14  N 22.84  -9.6
## 15  O 24.52  -8.8
## 16  P 26.20  -8.0
## 17  Q 27.88  -7.2
## 18  R 29.56  -6.4
## 19  S 31.24  -5.6
```

```
## 20  T 32.92  -4.8
## 21  U 34.60  -4.0
## 22  V 36.28  -3.2
## 23  W 37.96  -2.4
## 24  X 39.64  -1.6
## 25  Y 41.32  -0.8
## 26  Z 43.00   0.0
```

问： `seq` 函数中的 `along.with` 参数的意义是什么？请举例说明。

答：

```
## 代码写这里，并运行；
```

- **提供代码，合并以下两个 `data.frame`**

```
> df1 的内容
Id Age
1 14
2 12
3 15
4 10

>df2 的内容
Id Sex Code
1 F a
2 M b
3 M c
4 F d
```

合并之后的结果：

```
> M
```

```
Id Age Sex Code
1 14 F a
2 12 M b
3 15 M c
4 10 F d
```

```
## 代码写这里，并运行；
Id<-1:4
Age<-c(14,12,15,10)
Sex<-c('F','M','M','F')
Code<-c('a','b','c','d')
df1<-data.frame(Id=Id,Age=Age)
df1
```

```
##   Id Age
## 1  1  14
## 2  2  12
## 3  3  15
## 4  4  10
```

```
df2<-data.frame(Id,Sex,Code)
df2
```

```
##   Id Sex Code
## 1  1   F    a
## 2  2   M    b
## 3  3   M    c
## 4  4   F    d
```

```
M<-merge(df1,df2)
M
```

```
##   Id Age Sex Code
## 1  1  14   F    a
```

```
## 2  2  12   M    b
## 3  3  15   M    c
## 4  4  10   F    d
```

- 从上面的 data.frame 中删除 code 列

```
## 代码写这里，并运行;
M<-subset(M,select=-Code)
M
```

```
##   Id Age Sex
## 1  1  14   F
## 2  2  12   M
## 3  3  15   M
## 4  4  10   F
```

- 练习，回答代码中的问题

```
## 1. 生成一个10 行2 列的data.frame
df3 <- data.frame( data = 1:10, group = c("A","B") );
## 2. 增加一列，其长度是1，可以吗？
cbind(df3, newcol = 1);
## 3. 增加一列，其长度是10，可以吗？
cbind(df3, newcol = 1:10);
## 4. 增加一列，其长度是2，可以吗？
cbind(df3, newcol = 1:2);
## 5. 增加一列，其长度是3，可以吗？
cbind(df3, newcol = 1:3);
```

```
## 代码写这里，并运行;
df3 <- data.frame( data = 1:10, group = c("A","B") )
df3
```

```
##    data group
## 1     1     A
## 2     2     B
## 3     3     A
## 4     4     B
## 5     5     A
## 6     6     B
## 7     7     A
## 8     8     B
## 9     9     A
## 10   10     B
```

```
cbind(df3, newcol = 1);
```

```
##    data group newcol
## 1     1     A      1
## 2     2     B      1
## 3     3     A      1
## 4     4     B      1
## 5     5     A      1
## 6     6     B      1
## 7     7     A      1
## 8     8     B      1
## 9     9     A      1
## 10   10     B      1
```

```
cbind(df3, newcol = 1:10);
```

```
##    data group newcol
```

```
## 1     1     A      1
## 2     2     B      2
## 3     3     A      3
## 4     4     B      4
## 5     5     A      5
## 6     6     B      6
## 7     7     A      7
## 8     8     B      8
## 9     9     A      9
## 10   10     B     10
```

```
cbind(df3, newcol = 1:2);
```

```
##    data group newcol
## 1     1     A      1
## 2     2     B      2
## 3     3     A      1
## 4     4     B      2
## 5     5     A      1
## 6     6     B      2
## 7     7     A      1
## 8     8     B      2
## 9     9     A      1
## 10   10     B      2
```

```
# cbind(df3, newcol = 1:3);
```

答：通过运行代码，可以发现

2. 增加一列，其长度是 1；3. 增加一列，其长度是 10；4. 增加一列，其长度是 2 这三个是可以的，因为其长度正好是行数可以整除的数：10/1=10；10/10=1；10/2=5，可以完成并行计算或者说循环补齐

但 5. 增加一列，其长度是 3 这一条不行，出现报错

Error in data.frame(..., check.names = FALSE) : 参数值意味着不同的行数: 10, 3

因为其长度不是行数可以整除的数：10/3=3……1，无法正常完成循环计算、并行计算

练习与作业 2，tibble

- 运行以下代码，生成一个新的 tibble：

```
## 如果系统中没有 lubridate 包，则安装：
if (!require("lubridate")){
  chooseCRANmirror();
  install.packages("lubridate");
}
```

```
## 载入需要的程辑包：lubridate
```

```
##
## 载入程辑包：'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(lubridate);

if (!require("tibble")){
  chooseCRANmirror();
  install.packages("tibble");
}
```

```
## 载入需要的程辑包：tibble
```

```
library(tibble);

tibble(
  a = lubridate::now() + runif(1e3) * 86400,
  b = lubridate::today() + runif(1e3) * 30,
  c = 1:1e3,
  d = runif(1e3),
  e = sample(letters, 1e3, replace = TRUE)
)
```

```
## # A tibble: 1,000 x 5
##    a                   b              c      d e
##    <dttm>              <date>     <int>  <dbl> <chr>
##  1 2021-09-21 17:20:21 2021-09-26     1 0.795  i
##  2 2021-09-21 01:25:35 2021-09-25     2 0.311  h
##  3 2021-09-21 04:37:26 2021-10-12     3 0.118  c
##  4 2021-09-21 05:44:12 2021-10-05     4 0.841  h
##  5 2021-09-21 13:03:42 2021-09-25     5 0.720  c
##  6 2021-09-21 20:49:10 2021-09-24     6 0.0922 q
##  7 2021-09-21 05:59:12 2021-09-24     7 0.737  m
##  8 2021-09-21 14:38:04 2021-09-26     8 0.119  r
##  9 2021-09-21 00:06:46 2021-10-14     9 0.554  c
## 10 2021-09-21 17:52:17 2021-09-27    10 0.163  b
## # ... with 990 more rows
```

从中可以看出，tibble 支持一些细分数据类型，包括：

- <dttm>
- <date>

等；

- **生成一个如下的 tibble，完成以下任务：**

```
df <- tibble(
  x = runif(5),
  y = rnorm(5)
)
```

任务：

- 取一列，比如 x 这一列，得到一个 tibble；
- 取一列，比如 y 这一列，得到一个 vector；

```
## 代码写这里，并运行；
```

- **用 tibble 函数创建一个新的空表，并逐行增加一些随机的数据，共增加三行：**

```
## 代码写这里，并运行；
## 新 tibble, with defined columns ... 创建表头
tb <- tibble( name = character(), age = integer(), salary = double() );

## 增加三行随机数据；
```

- ** 请解释为什么下面第一行代码能够运行成功，但第二个不行？**

这个可以：

```
data.frame(a = 1:6, b = LETTERS[1:2]);
```

但下面这个不行：

```
tibble(a = 1:6, b = LETTERS[1:2]);
```

问：为什么？tibble 循环的规则是什么？

答：

- **attach 和 detach：**

问：这个两个函数的用途是什么？请用 iris 这个系统自带变量举例说明。

答：

- **使用内置变量 airquality：**
- 检查它是否是 tibble；
- 如果不是，转化为 tibble；

```
## 代码写这里，并运行；
```

- **问：tibble::enframe 函数的用途是什么？请举例说明：**

答：

- **简述 tibble 相比 data.frame 的优势？并用实例展示**

答：

```
## 代码写这里，并运行；
```

练习与作业 3：IO

- **提供代码，正确读取以下文件：**

注：数据在当前目录下的 `data/` 子目录里

- Table0.txt
- Table1.txt
- Table2.txt
- Table3.txt
- Table4.txt
- Table5.txt
- Table6.txt
- states1.csv
- states2.csv

注 2：每个文件读取需要提供两种方法，一种是利用系统自带函数，另一种是 `readr` 包的函数；

```
## 用系统自带函数，并显示读取的内容；

## 用 readr 包的函数读取，并显示读取的内容；
```