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

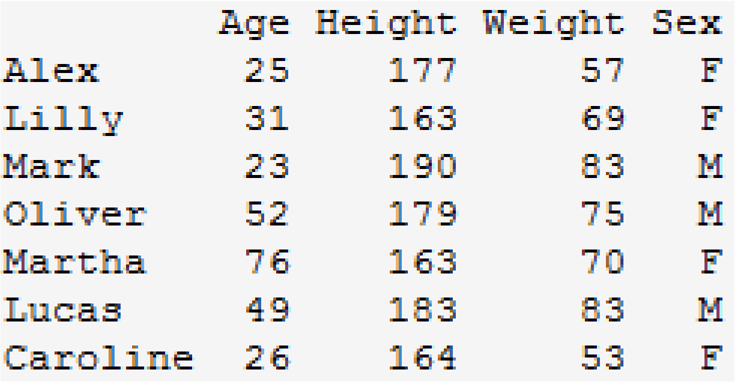
  - 常用相关函数
    - nrow, ncol, dim , str , head, tail
  - data.frame 和 tibble 的不同
  - 高级技巧：
    - with, within

- IO
  - 系统自带函数
  - readr 带的函数
  - 不同格式的读取
  - 从网络、压缩文件读取

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

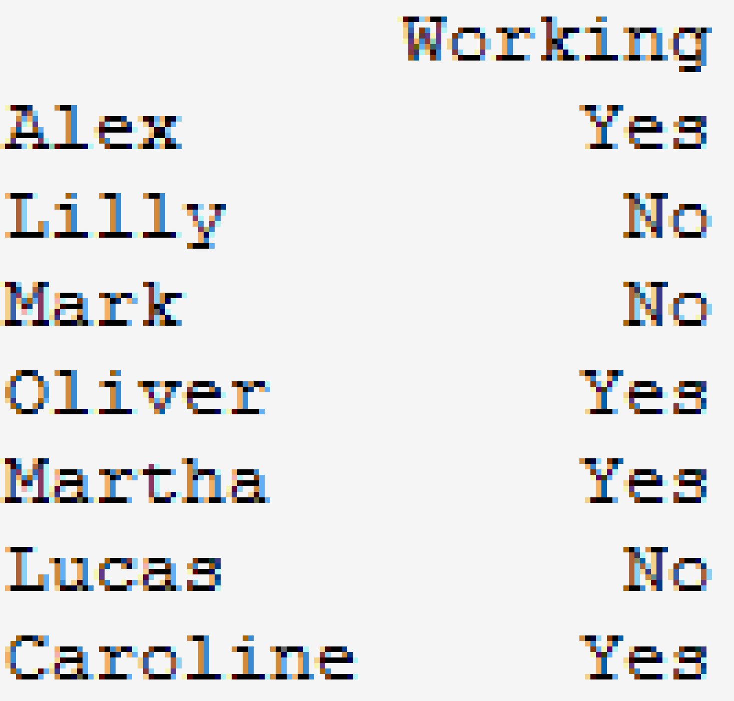


```
         Age Height Weight Sex
Alex      25    177     57   F
Lilly     31    163     69   F
Mark      23    190     83   M
Oliver    52    179     75   M
Martha    76    163     70   F
Lucas     49    183     83   M
Caroline  26    164     53   F
```

```
## 先生成前三列；
dat1 <-
data.frame( 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)
          )
row.names(dat1) <- c("Alex","Lilly","Mark","Oliver","Martha","Lucas","Caroline")
## 再插入第四列
Sex <- c("F","F","M","M","F","M","F")
dat1 <- cbind(dat1,Sex)
## 显示最终结果
dat1;
```

```
##          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
dat2 <-
data.frame(Working = c("Yes","No","No","Yes","Yes","No","Yes"))
row.names(dat2) <- c("Alex","Lilly","Mark","Oliver","Martha","Lucas","Caroline")
## 显示结果
dat2
```

```
##          Working
## Alex         Yes
## Lilly         No
## Mark          No
## Oliver       Yes
## Martha       Yes
## Lucas         No
## Caroline     Yes
## 显示 Working 列的性质
str(dat2)
```

```
## 'data.frame':    7 obs. of  1 variable:
##  $ Working: chr  "Yes" "No" "No" "Yes" ...
```

- 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
```

```
dat3 <- data.frame(state.center)
dat3
```

```
##            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；
m1 <- matrix( 1:250, nrow = 50, byrow = T)
rownames(m1) <- rownames(m1, do.NULL = F, prefix = "row_")
m1
```

```
##        [,1] [,2] [,3] [,4] [,5]
## row_1     1    2    3    4    5
## row_2     6    7    8    9   10
## row_3    11   12   13   14   15
## row_4    16   17   18   19   20
## row_5    21  22  23  24  25
## row_6    26  27  28  29  30
## row_7    31  32  33  34  35
## row_8    36  37  38  39  40
## row_9    41  42  43  44  45
## row_10   46  47  48  49  50
## row_11   51  52  53  54  55
## row_12   56  57  58  59  60
## row_13   61  62  63  64  65
## row_14   66  67  68  69  70
## row_15   71  72  73  74  75
## row_16   76  77  78  79  80
## row_17   81  82  83  84  85
## row_18   86  87  88  89  90
## row_19   91  92  93  94  95
## row_20   96  97  98  99 100
## row_21  101 102 103 104 105
## row_22  106 107 108 109 110
## row_23  111 112 113 114 115
## row_24  116 117 118 119 120
## row_25  121 122 123 124 125
## row_26  126 127 128 129 130
## row_27  131 132 133 134 135
## row_28  136 137 138 139 140
## row_29  141 142 143 144 145
## row_30  146 147 148 149 150
## row_31  151 152 153 154 155
## row_32  156 157 158 159 160
## row_33  161 162 163 164 165
## row_34  166 167 168 169 170
## row_35  171 172 173 174 175
## row_36  176 177 178 179 180
## row_37  181 182 183 184 185
## row_38  186  187  188  189  190
## row_39  191  192  193  194  195
## row_40  196  197  198  199  200
## row_41  201  202  203  204  205
## row_42  206  207  208  209  210
## row_43  211  212  213  214  215
## row_44  216  217  218  219  220
## row_45  221  222  223  224  225
## row_46  226  227  228  229  230
## row_47  231  232  233  234  235
## row_48  236  237  238  239  240
## row_49  241  242  243  244  245
## row_50  246  247  248  249  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；
str(VADeaths)
```

```
##  num [1:5, 1:4] 11.7 18.1 26.9 41 66 8.7 11.7 20.3 30.9 54.3 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:5] "50-54" "55-59" "60-64" "65-69" ...
##   ..$ : chr [1:4] "Rural Male" "Rural Female" "Urban Male" "Urban Female"
```

```
VADeaths <- data.frame(VADeaths)
Total = rowSums(VADeaths)
VADeaths <- cbind(VADeaths,Total)
VADeaths <- VADeaths[,c(5,1,2,3,4)]
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；
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
```

```
s1 <- swiss[c(1:3,10:13),c("Examination","Education","Infant.Mortality")]
s1["Sarine","Infant.Mortality"] <- NA
Mean <- apply(s1,1,mean,na.rm = TRUE)
s1 <- cbind(s1,Mean)
s1
```

```
##              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 14.50000
## 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)
ID <- data.frame(x,y)
ID
```

```
##        x     y
## 1   1.00 -20.0
## 2   2.68 -19.2
## 3   4.36 -18.4
## 4   6.04 -17.6
## 5   7.72 -16.8
## 6   9.40 -16.0
## 7  11.08 -15.2
## 8  12.76 -14.4
## 9  14.44 -13.6
## 10 16.12 -12.8
## 11 17.80 -12.0
## 12 19.48 -11.2
## 13 21.16 -10.4
## 14 22.84  -9.6
## 15 24.52  -8.8
## 16 26.20  -8.0
## 17 27.88  -7.2
## 18 29.56  -6.4
## 19 31.24  -5.6
## 20 32.92  -4.8
## 21 34.60  -4.0
## 22 36.28  -3.2
## 23 37.96  -2.4
## 24 39.64  -1.6
## 25 41.32  -0.8
## 26 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：将生成的向量作为已知向量元素的索引

```
## 代码写这里，并运行；
f1 <- seq(1,20,along.with = LETTERS)
f1
```

```
##  [1]  1.00  1.76  2.52  3.28  4.04  4.80  5.56  6.32  7.08  7.84  8.60  9.36
## [13] 10.12 10.88 11.64 12.40 13.16 13.92 14.68 15.44 16.20 16.96 17.72 18.48
## [25] 19.24 20.00
```

- **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；
df1 <- data.frame("Age" = c(14,12,15,10))
df2 <- data.frame("Sex" = c("F","M","M","F"),
                  "Code" = c("a","b","c","d"))
M <- head(cbind(df1,df2))
M
```

```
##   Age Sex Code
## 1  14   F    a
## 2  12   M    b
## 3  15   M    c
## 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
```

```
##   Age Sex
## 1  14   F
## 2  12   M
## 3  15   M
## 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);
```

答：2. 可行 3. 可行 4. 可行 5. 不可行，因为 10 不能被 3 整除

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 2021-10-16 07:28:56 2021-11-05     1 0.449 a
##  2 2021-10-16 14:22:21 2021-11-05     2 0.740 g
##  3 2021-10-16 05:29:32 2021-10-17     3 0.923 z
##  4 2021-10-15 23:06:59 2021-10-29     4 0.521 k
##  5 2021-10-16 21:51:57 2021-11-08     5 0.805 r
##  6 2021-10-16 10:26:27 2021-11-01     6 0.129 w
##  7 2021-10-15 23:41:23 2021-10-20     7 0.300 e
##  8 2021-10-16 12:26:57 2021-11-07     8 0.539 k
##  9 2021-10-16 01:57:21 2021-11-12     9 0.591 r
## 10 2021-10-16 14:31:22 2021-10-21    10 0.550 w
## # ... 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；
df <- tibble(
      x = runif(5),
      y = rnorm(5)
    )
df[,c("x")]
```

```
## # A tibble: 5 x 1
##       x
##   <dbl>
## 1 0.782
## 2 0.166
## 3 0.176
## 4 0.484
## 5 0.355
```

```
class(df[[2]])
```

```
## [1] "numeric"
```

- 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；
tb <- tibble(name = LETTERS[1:3],
             age = runif(3,min = 20, max = 50),
             salary = rnorm(3,mean = 5000,sd = 2000))
tb
```

```
## # A tibble: 3 x 3
##   name    age salary
##   <chr> <dbl>  <dbl>
## 1 A      46.7  4999.
## 2 B      33.2  6835.
## 3 C      45.2  6639.
```

- ** 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：因为 tibble 仅限于长度为 1 或等长的循环，不能同时多次循环

- **`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：attach 是将变量与对象名联系起来，便于直接获取数据 detach 是取消 attach 的操作

```
head(iris, n = 3);
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
```

```
attach(iris);
head(Petal.Length, n = 5);
```

```
## [1] 1.4 1.4 1.3 1.5 1.4
```

```
detach(iris);
```

- **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；
str(airquality)
```

```
## 'data.frame':    153 obs. of  6 variables:
##  $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
##  $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
##  $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
##  $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
##  $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
airquality <- tibble(airquality)
```

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

答：将向量转化为 tibble 形式

```
c(a = 1, b = 2)
```

```
## a b
## 1 2
```

```r
tibble::enframe(c(a = 1, b = 2))
```

```
## # A tibble: 2 x 2
##   name  value
##   <chr> <dbl>
## 1 a         1
## 2 b         2
```

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

答：tibble 是逐行输入的，因此可以直接引用已创建的变量来生成新变量，而 data.frame 不可以

```r
## 代码写这里，并运行；
tibble( x = 1:10,
        y = 2,
        z = x ^ 2 + y
        )
```

```
## # A tibble: 10 x 3
##        x     y     z
##    <int> <dbl> <dbl>
##  1     1     2     3
##  2     2     2     6
##  3     3     2    11
##  4     4     2    18
##  5     5     2    27
##  6     6     2    38
##  7     7     2    51
##  8     8     2    66
##  9     9     2    83
## 10    10     2   102
```

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；
read.table("data/Table0.txt");
```

```
##         V1 V2  V3 V4 V5
## 1     Alex 25 177 57  F
## 2    Lilly 31 163 69  F
## 3     Mark 23 190 83  M
## 4   Oliver 52 179 75  M
## 5   Martha 76 163 70  F
## 6    Lucas 49 183 83  M
## 7 Caroline 26 164 53  F
```

```
read.table("data/Table1.txt");
```

```
##         V1  V2     V3     V4  V5
## 1     Name Age Height Weight Sex
## 2     Alex  25    177     57   F
## 3    Lilly  31    163     69   F
## 4     Mark  23    190     83   M
## 5   Oliver  52    179     75   M
## 6   Martha  76    163     70   F
## 7    Lucas  49    183     83   M
## 8 Caroline  26    164     53   F
```

```
read.table("data/Table2.txt", fill = TRUE, header = TRUE);
```

```
##        Table X2.  Name.   Age. Height. Weight and Sex from X7 people
## 1       Name Age Height Weight     Sex     NA  NA  NA   NA NA     NA
## 2     /Alex/  25    177     57     /F/     NA  NA  NA   NA NA     NA
## 3    /Lilly/  31    163     69     /F/     NA  NA  NA   NA NA     NA
## 4     /Mark/  23    190     83     /M/     NA  NA  NA   NA NA     NA
## 5   /Oliver/  52    179     75     /M/     NA  NA  NA   NA NA     NA
## 6   /Martha/  76    163     70     /F/     NA  NA  NA   NA NA     NA
## 7    /Lucas/  49    183     83     /M/     NA  NA  NA   NA NA     NA
## 8 /Caroline/  26    164     53     /F/     NA  NA  NA   NA NA     NA
```

```
read.table("data/Table3.txt", fill = TRUE, header = TRUE);
```

```
##      Table X2.  Name.   Age. Height. Weight and Sex from X7 people
## 1     Name Age Height Weight     Sex     NA  NA  NA   NA NA     NA
## 2     Alex  25    177     57       F     NA  NA  NA   NA NA     NA
## 3    Lilly  31   <NA>     69       F     NA  NA  NA   NA NA     NA
## 4     Mark  --    190     83       M     NA  NA  NA   NA NA     NA
## 5   Oliver  52    179     75       M     NA  NA  NA   NA NA     NA
## 6   Martha  76      *     70       F     NA  NA  NA   NA NA     NA
## 7    Lucas  49    183     **       M     NA  NA  NA   NA NA     NA
## 8 Caroline  26    164     53       F     NA  NA  NA   NA NA     NA
```

```r
read.table("data/Table4.txt");
```

```
##         V1  V2     V3     V4  V5
## 1     Name Age Height Weight Sex
## 2     Alex  25   1,77     57   F
## 3    Lilly  31   <NA>     69   F
## 4     Mark  --   1,90     83   M
## 5   Oliver  52   1,79     75   M
## 6   Martha  76      *     70   F
## 7    Lucas  49   1,83     **   M
## 8 Caroline  26   1,64     53   F
```

```r
read.table("data/Table5.txt");
```

```
##                            V1
## 1 Name;Age;Height;Weight;Sex
## 2          Alex;25;1,77;57;F
## 3           Lilly;31;NA;69;F
## 4          Mark;--;1,90;83;M
## 5        Oliver;52;1,79;75;M
## 6            Martha;76;;70;F
## 7         Lucas;49;1,83;**;M
## 8      Caroline;26;1,64;53;F
```

```r
read.table("data/Table6.txt", fill = TRUE, header = TRUE);
```

```
##        Table X2.  Name.   Age. Height.     Weight and Sex from X7 people
## 1       Name Age Height Weight     Sex             NA  NA   NA NA     NA
## 2       Alex  25    177     57       F      @Boss  NA  NA   NA NA     NA
## 3      Lilly  31    163     69       F @Secretary  NA  NA   NA NA     NA
## 4       Mark  23    190     83       M             NA  NA   NA NA     NA
## 5     Oliver  52    179     75       M             NA  NA   NA NA     NA
## 6     Martha  76    163     70       F             NA  NA   NA NA     NA
## 7      Lucas  49    183     83       M             NA  NA   NA NA     NA
## 8   Caroline  26    164     53       F             NA  NA   NA NA     NA
## 9       Alex  25    177     57       F             NA  NA   NA NA     NA
## 10     Lilly  31    163     69       F             NA  NA   NA NA     NA
## 11      Mark  23    190     83       M             NA  NA   NA NA     NA
## 12    Oliver  52    179     75       M             NA  NA   NA NA     NA
## 13    Martha  76    163     70       F             NA  NA   NA NA     NA
## 14     Lucas  49    183     83       M             NA  NA   NA NA     NA
## 15  Caroline  26    164     53       F             NA  NA   NA NA     NA
## 16      Alex  25    177     57       F             NA  NA   NA NA     NA
## 17     Lilly  31    163     69       F             NA  NA   NA NA     NA
## 18      Mark  23    190     83       M             NA  NA   NA NA     NA
## 19    Oliver  52    179     75       M             NA  NA   NA NA     NA
## 20    Martha  76    163     70       F             NA  NA   NA NA     NA
## 21     Lucas  49    183     83       M             NA  NA   NA NA     NA
## 22  Caroline  26    164     53       F             NA  NA   NA NA     NA
## 23      Alex  25    177     57       F             NA  NA   NA NA     NA
## 24     Lilly  31    163     69       F             NA  NA   NA NA     NA
## 25      Mark  23    190     83       M             NA  NA   NA NA     NA
## 26    Oliver  52    179     75       M             NA  NA   NA NA     NA
## 27    Martha  76    163     70       F             NA  NA   NA NA     NA
## 28     Lucas  49    183     83       M             NA  NA   NA NA     NA
## 29  Caroline  26    164     53       F             NA  NA   NA NA     NA
## 30      Alex  25    177     57       F             NA  NA   NA NA     NA
## 31     Lilly  31    163     69       F             NA  NA   NA NA     NA
## 32      Mark  23    190     83       M             NA  NA   NA NA     NA
## 33    Oliver  52    179     75       M             NA  NA   NA NA     NA
## 34    Martha  76    163     70       F             NA  NA   NA NA     NA
## 35     Lucas  49    183     83       M             NA  NA   NA NA     NA
## 36  Caroline  26    164     53       F             NA  NA   NA NA     NA
## 37      Alex  25    177     57       F             NA  NA   NA NA     NA
## 38     Lilly  31    163     69       F             NA  NA   NA NA     NA
## 39      Mark  23    190     83       M             NA  NA   NA NA     NA
## 40    Oliver  52    179     75        M              NA  NA   NA NA      NA
## 41    Martha  76    163     70        F              NA  NA   NA NA      NA
## 42     Lucas  49    183     83        M              NA  NA   NA NA      NA
## 43  Caroline  26    164     53        F              NA  NA   NA NA      NA
## 44      Alex  25    177     57        F              NA  NA   NA NA      NA
## 45     Lilly  31    163     69        F              NA  NA   NA NA      NA
## 46      Mark  23    190     83        M              NA  NA   NA NA      NA
## 47    Oliver  52    179     75        M              NA  NA   NA NA      NA
## 48    Martha  76    163     70        F              NA  NA   NA NA      NA
## 49     Lucas  49    183     83        M              NA  NA   NA NA      NA
## 50  Caroline  26    164     53        F              NA  NA   NA NA      NA
## 51      Alex  25    177     57        F              NA  NA   NA NA      NA
## 52     Lilly  31    163     69        F              NA  NA   NA NA      NA
## 53      Mark  23    190     83        M              NA  NA   NA NA      NA
## 54    Oliver  52    179     75        M              NA  NA   NA NA      NA
## 55    Martha  76    163     70        F              NA  NA   NA NA      NA
## 56     Lucas  49    183     83        M              NA  NA   NA NA      NA
## 57  Caroline  26    164     53        F              NA  NA   NA NA      NA
## 58      Alex  25    177     57        F              NA  NA   NA NA      NA
## 59     Lilly  31    163     69        F              NA  NA   NA NA      NA
## 60      Mark  23    190     83        M              NA  NA   NA NA      NA
## 61    Oliver  52    179     75        M              NA  NA   NA NA      NA
## 62    Martha  76    163     70        F              NA  NA   NA NA      NA
## 63     Lucas  49    183     83        M              NA  NA   NA NA      NA
## 64  Caroline  26    164     53        F              NA  NA   NA NA      NA
## 65      Alex  25    177     57        F              NA  NA   NA NA      NA
## 66     Lilly  31    163     69        F              NA  NA   NA NA      NA
## 67      Mark  23    190     83        M              NA  NA   NA NA      NA
## 68    Oliver  52    179     75        M              NA  NA   NA NA      NA
## 69    Martha  76    163     70        F              NA  NA   NA NA      NA
## 70     Lucas  49    183     83        M              NA  NA   NA NA      NA
## 71  Caroline  26    164     53        F              NA  NA   NA NA      NA
## 72      Alex  25    177     57        F              NA  NA   NA NA      NA
## 73     Lilly  31    163     69       F             NA  NA   NA NA     NA
## 74      Mark  23    190     83       M             NA  NA   NA NA     NA
## 75    Oliver  52    179     75       M             NA  NA   NA NA     NA
## 76    Martha  76    163     70       F             NA  NA   NA NA     NA
## 77     Lucas  49    183     83       M             NA  NA   NA NA     NA
## 78  Caroline  26    164     53       F             NA  NA   NA NA     NA
## 79      Alex  25    177     57       F             NA  NA   NA NA     NA
## 80     Lilly  31    163     69       F             NA  NA   NA NA     NA
## 81      Mark  23    190     83       M             NA  NA   NA NA     NA
## 82    Oliver  52    179     75       M             NA  NA   NA NA     NA
## 83    Martha  76    163     70       F             NA  NA   NA NA     NA
## 84     Lucas  49    183     83       M             NA  NA   NA NA     NA
## 85  Caroline  26    164     53       F             NA  NA   NA NA     NA
## 86      Alex  25    177     57       F             NA  NA   NA NA     NA
## 87     Lilly  31    163     69       F             NA  NA   NA NA     NA
## 88      Mark  23    190     83       M             NA  NA   NA NA     NA
## 89    Oliver  52    179     75       M             NA  NA   NA NA     NA
## 90    Martha  76    163     70       F             NA  NA   NA NA     NA
## 91     Lucas  49    183     83       M             NA  NA   NA NA     NA
## 92  Caroline  26    164     53       F             NA  NA   NA NA     NA
## 93      Alex  25    177     57       F             NA  NA   NA NA     NA
## 94     Lilly  31    163     69       F             NA  NA   NA NA     NA
## 95      Mark  23    190     83       M             NA  NA   NA NA     NA
## 96    Oliver  52    179     75       M             NA  NA   NA NA     NA
## 97    Martha  76    163     70       F             NA  NA   NA NA     NA
## 98     Lucas  49    183     83       M             NA  NA   NA NA     NA
## 99  Caroline  26    164     53       F             NA  NA   NA NA     NA
## 100     Alex  25    177     57       F             NA  NA   NA NA     NA
## 101    Lilly  31    163     69       F             NA  NA   NA NA     NA
## 102     Mark  23    190     83       M             NA  NA   NA NA     NA
## 103   Oliver  52    179     75       M             NA  NA   NA NA     NA
## 104   Martha  76    163     70       F             NA  NA   NA NA     NA
## 105    Lucas  49    183     83       M             NA  NA   NA NA     NA
## 106 Caroline  26    164     53       F             NA  NA   NA NA     NA
```

```
read.csv("data/states1.csv");
```

```
##                  X Population Income Illiteracy Life.Exp Murder HS.Grad Frost
## 1          Alabama       3615   3624        2.1    69.05   15.1    41.3    20
## 2           Alaska        365   6315        1.5    69.31   11.3    66.7   152
## 3          Arizona       2212   4530        1.8    70.55    7.8    58.1    15
## 4         Arkansas       2110   3378        1.9    70.66   10.1    39.9    65
## 5       California      21198   5114        1.1    71.71   10.3    62.6    20
## 6         Colorado       2541   4884        0.7    72.06    6.8    63.9   166
## 7      Connecticut       3100   5348        1.1    72.48    3.1    56.0   139
## 8         Delaware        579   4809        0.9    70.06    6.2    54.6   103
## 9          Florida       8277   4815        1.3    70.66   10.7    52.6    11
## 10         Georgia       4931   4091        2.0    68.54   13.9    40.6    60
## 11          Hawaii        868   4963        1.9    73.60    6.2    61.9     0
## 12           Idaho        813   4119        0.6    71.87    5.3    59.5   126
## 13        Illinois      11197   5107        0.9    70.14   10.3    52.6   127
## 14         Indiana       5313   4458        0.7    70.88    7.1    52.9   122
## 15            Iowa       2861   4628        0.5    72.56    2.3    59.0   140
## 16          Kansas       2280   4669        0.6    72.58    4.5    59.9   114
## 17        Kentucky       3387   3712        1.6    70.10   10.6    38.5    95
## 18       Louisiana       3806   3545        2.8    68.76   13.2    42.2    12
## 19           Maine       1058   3694        0.7    70.39    2.7    54.7   161
## 20        Maryland       4122   5299        0.9    70.22    8.5    52.3   101
## 21   Massachusetts       5814   4755        1.1    71.83    3.3    58.5   103
## 22        Michigan       9111   4751        0.9    70.63   11.1    52.8   125
## 23       Minnesota       3921   4675        0.6    72.96    2.3    57.6   160
## 24     Mississippi       2341   3098        2.4    68.09   12.5    41.0    50
## 25        Missouri       4767   4254        0.8    70.69    9.3    48.8   108
## 26         Montana        746   4347        0.6    70.56    5.0    59.2   155
## 27        Nebraska       1544   4508        0.6    72.60    2.9    59.3   139
## 28          Nevada        590   5149        0.5    69.03   11.5    65.2   188
## 29   New Hampshire        812   4281        0.7    71.23    3.3    57.6   174
## 30     New Jersey       7333   5237        1.1    70.93    5.2    52.5   115
## 31     New Mexico       1144   3601        2.2    70.32    9.7    55.2   120
## 32       New York      18076   4903        1.4    70.55   10.9    52.7    82
## 33 North Carolina       5441   3875        1.8    69.21   11.1    38.5    80
## 34   North Dakota        637   5087        0.8    72.78    1.4    50.3   186
## 35           Ohio      10735   4561        0.8    70.82    7.4    53.2   124
## 36       Oklahoma       2715   3983        1.1    71.42    6.4    51.6    82
## 37         Oregon       2284   4660        0.6    72.13    4.2    60.0    44
## 38   Pennsylvania      11860   4449        1.0    70.43    6.1    50.2   126
## 39   Rhode Island        931   4558        1.3    71.90    2.4    46.4   127
## 40 South Carolina       2816   3635        2.3    67.96   11.6    37.8    65
## 41   South Dakota        681   4167        0.5    72.08    1.7    53.3   172
## 42      Tennessee       4173   3821        1.7    70.11   11.0    41.8    70
## 43          Texas      12237   4188        2.2    70.90   12.2    47.4    35
## 44           Utah       1203   4022        0.6    72.90    4.5    67.3   137
## 45        Vermont        472   3907        0.6    71.64    5.5    57.1   168
## 46       Virginia       4981   4701        1.4    70.08    9.5    47.8    85
## 47     Washington       3559   4864        0.6    71.72    4.3    63.5    32
## 48  West Virginia       1799   3617        1.4    69.48    6.7    41.6   100
## 49      Wisconsin       4589   4468        0.7    72.48    3.0    54.5   149
## 50        Wyoming        376   4566        0.6    70.29    6.9    62.9   173
##        Area
## 1   50708
## 2  566432
## 3  113417
## 4   51945
## 5  156361
## 6  103766
## 7    4862
## 8    1982
## 9   54090
## 10  58073
## 11   6425
## 12  82677
## 13  55748
## 14  36097
## 15  55941
## 16  81787
## 17  39650
## 18  44930
## 19  30920
## 20   9891
## 21   7826
## 22  56817
## 23  79289
## 24  47296
## 25  68995
## 26 145587
## 27  76483
## 28 109889
## 29   9027
## 30   7521
## 31 121412
## 32  47831
## 33  48798
## 34  69273
## 35  40975
## 36  68782
## 37  96184
## 38  44966
## 39   1049
## 40  30225
## 41  75955
## 42  41328
## 43 262134
## 44  82096
## 45   9267
## 46  39780
## 47  66570
## 48  24070
## 49  54464
## 50  97203
```

```
read.csv("data/states2.csv", header = FALSE);
```

```
##                                                                     V1    V2
## 1  ;Population;Income;Illiteracy;Life Exp;Murder;HS Grad;Frost;Area      
## 2                                              Alabama;3615;3624;2  1;69
## 3                                                Alaska;365;6315;1  5;69
## 4                                              Arizona;2212;4530;1  8;70
## 5                                             Arkansas;2110;3378;1  9;70
## 6                                          California;21198;5114;1  1;71
## 7                                             Colorado;2541;4884;0  7;72
## 8                                          Connecticut;3100;5348;1  1;72
## 9                                              Delaware;579;4809;0  9;70
## 10                                             Florida;8277;4815;1  3;70
## 11                                          Georgia;4931;4091;2;68 54;13
## 12                                               Hawaii;868;4963;1  9;73
## 13                                                Idaho;813;4119;0  6;71
## 14                                           Illinois;11197;5107;0  9;70
## 15                                             Indiana;5313;4458;0  7;70
## 16                                                Iowa;2861;4628;0  5;72
## 17                                              Kansas;2280;4669;0  6;72
## 18                                            Kentucky;3387;3712;1  6;70
## 19                                           Louisiana;3806;3545;2  8;68
## 20                                               Maine;1058;3694;0  7;70
## 21                                            Maryland;4122;5299;0  9;70
## 22                                       Massachusetts;5814;4755;1  1;71
## 23                                            Michigan;9111;4751;0  9;70
## 24                                           Minnesota;3921;4675;0  6;72
## 25                                          Mississippi;2341;3098;2  4;68
## 26                                             Missouri;4767;4254;0  8;70
## 27                                               Montana;746;4347;0  6;70
## 28                                             Nebraska;1544;4508;0  6;72
## 29                                                Nevada;590;5149;0  5;69
## 30                                         New Hampshire;812;4281;0  7;71
## 31                                           New Jersey;7333;5237;1  1;70
## 32                                           New Mexico;1144;3601;2  2;70
## 33                                            New York;18076;4903;1  4;70
## 34                                       North Carolina;5441;3875;1  8;69
## 35                                          North Dakota;637;5087;0  8;72
## 36                                                 Ohio;10735;4561;0  8;70
## 37                                             Oklahoma;2715;3983;1  1;71
## 38                                               Oregon;2284;4660;0  6;72
## 39                                     Pennsylvania;11860;4449;1;70  43;6
## 40                                          Rhode Island;931;4558;1  3;71
## 41                                       South Carolina;2816;3635;2  3;67
## 42                                          South Dakota;681;4167;0  5;72
## 43                                            Tennessee;4173;3821;1  7;70
## 44                                                Texas;12237;4188;2  2;70
## 45                                                  Utah;1203;4022;0  6;72
## 46                                                Vermont;472;3907;0  6;71
## 47                                             Virginia;4981;4701;1  4;70
## 48                                           Washington;3559;4864;0  6;71
## 49                                        West Virginia;1799;3617;1  4;69
## 50                                            Wisconsin;4589;4468;0  7;72
## 51                                                Wyoming;376;4566;0  6;70
##           V3             V4           V5
## 1
## 2     05;15           1;41   3;20;50708
## 3     31;11           3;66 7;152;566432
## 4      55;7           8;58  1;15;113417
## 5     66;10           1;39   9;65;51945
## 6     71;10           3;62  6;20;156361
## 7      06;6           8;63 9;166;103766
## 8      48;3  1;56;139;4862
## 9      06;6           2;54   6;103;1982
## 10    66;10           7;52   6;11;54090
## 11     9;40     6;60;58073
## 12      6;6           2;61     9;0;6425
## 13     87;5           3;59  5;126;82677
## 14    14;10           3;52  6;127;55748
## 15     88;7           1;52  9;122;36097
## 16     56;2 3;59;140;55941
## 17     58;4           5;59  9;114;81787
## 18     1;10           6;38   5;95;39650
## 19    76;13           2;42   2;12;44930
## 20     39;2           7;54  7;161;30920
## 21     22;8           5;52   3;101;9891
## 22     83;3           3;58   5;103;7826
## 23    63;11           1;52  8;125;56817
## 24     96;2           3;57  6;160;79289
## 25    09;12  5;41;50;47296
## 26     69;9           3;48  8;108;68995
## 27  56;5;59   2;155;145587
## 28      6;2           9;59  3;139;76483
## 29    03;11           5;65 2;188;109889
## 30     23;3           3;57   6;174;9027
## 31     93;5           2;52   5;115;7521
## 32     32;9           7;55 2;120;121412
## 33    55;10           9;52   7;82;47831
## 34    21;11           1;38   5;80;48798
## 35     78;1           4;50  3;186;69273
## 36     82;7           4;53  2;124;40975
## 37     42;6           4;51   6;82;68782
## 38     13;4  2;60;44;96184
## 39     1;50    2;126;44966
## 40      9;2           4;46   4;127;1049
## 41    96;11           6;37   8;65;30225
## 42     08;1           7;53  3;172;75955
## 43 11;11;41     8;70;41328
## 44     9;12           2;47  4;35;262134
## 45      9;4           5;67  3;137;82096
## 46     64;5           5;57   1;168;9267
## 47     08;9           5;47   8;85;39780
## 48     72;4           3;63   5;32;66570
## 49     48;6           7;41  6;100;24070
## 50  48;3;54    5;149;54464
## 51     29;6           9;62  9;173;97203
```

```
## 用 readr 包的函数读取，并显示读取的内容;
library(tidyverse);
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

```
library(readr);
read_table("data/Table0.txt");
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Alex = col_character(),
##   `25` = col_double(),
##   `177` = col_double(),
##   `57` = col_double(),
##   F = col_character()
## )
```

```
## # A tibble: 6 x 5
##   Alex      `25` `177`  `57` F
##   <chr>    <dbl> <dbl> <dbl> <chr>
## 1 Lilly       31   163    69 F
## 2 Mark        23   190    83 M
## 3 Oliver      52   179    75 M
## 4 Martha      76   163    70 F
## 5 Lucas       49   183    83 M
## 6 Caroline    26   164    53 F
```

```
read_table("data/Table1.txt");
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Name = col_character(),
##   Age = col_double(),
##   Height = col_double(),
##   Weight = col_double(),
##   Sex = col_character()
## )
## # A tibble: 7 x 5
##   Name       Age Height Weight Sex
##   <chr>    <dbl>  <dbl>  <dbl> <chr>
## 1 Alex        25    177     57 F
## 2 Lilly       31    163     69 F
## 3 Mark        23    190     83 M
## 4 Oliver      52    179     75 M
## 5 Martha      76    163     70 F
## 6 Lucas       49    183     83 M
## 7 Caroline    26    164     53 F
```

```
read_table("data/Table2.txt");
```

```
## Warning: Missing column names filled in: 'X12' [12]
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Table = col_character(),
##   `2:` = col_character(),
##   `Name,` = col_character(),
##   `Age,` = col_character(),
##   `Height,` = col_character(),
##   Weight = col_character(),
##   and = col_character(),
##   Sex = col_character(),
##   from = col_character(),
##   `7` = col_character(),
##   people = col_character(),
##   X12 = col_character()
## )
```

```
## Warning: 9 parsing failures.
## row col   expected    actual              file
##   1  -- 12 columns 1 columns 'data/Table2.txt'
##   2  -- 12 columns 5 columns 'data/Table2.txt'
##   3  -- 12 columns 5 columns 'data/Table2.txt'
##   4  -- 12 columns 5 columns 'data/Table2.txt'
##   5  -- 12 columns 5 columns 'data/Table2.txt'
## ... ... .......... ......... .................
## See problems(...) for more details.
```

```
## # A tibble: 9 x 12
##   Table      `2:`  `Name,` `Age,` `Height,` Weight and   Sex   from  `7`   people
##   <chr>      <chr> <chr>   <chr>  <chr>     <chr>  <chr> <chr> <chr> <chr> <chr>
## 1 <NA>       <NA>  <NA>    <NA>   <NA>      <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
## 2 Name       Age   Height  Weight Sex       <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
## 3 /Alex/     25    177     57     /F/       <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
## 4 /Lilly/    31    163     69     /F/       <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
## 5 /Mark/     23    190     83     /M/       <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
## 6 /Oliver/   52    179     75     /M/       <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
## 7 /Martha/   76    163     70     /F/       <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
## 8 /Lucas/    49    183     83     /M/       <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
## 9 /Caroline/ 26    164     53     /F/       <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
## # ... with 1 more variable: X12 <chr>
```

```r
read_table("data/Table3.txt");
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Table = col_character(),
##   `2:` = col_character(),
##   `Name,` = col_character(),
##   `Age,` = col_character(),
##   `Height,` = col_character(),
##   Weight = col_character(),
##   and = col_character(),
##   Sex = col_character(),
##   from = col_character(),
##   `7` = col_character(),
##   people = col_character()
## )
```

```
## Warning: 8 parsing failures.
## row col   expected    actual              file
##   1  -- 11 columns 5 columns 'data/Table3.txt'
##   2  -- 11 columns 5 columns 'data/Table3.txt'
##   3  -- 11 columns 5 columns 'data/Table3.txt'
##   4  -- 11 columns 5 columns 'data/Table3.txt'
##   5  -- 11 columns 5 columns 'data/Table3.txt'
## ... ... .......... ......... .................
## See problems(...) for more details.
```

```
## # A tibble: 8 x 11
##   Table    `2:`  `Name,` `Age,` `Height,` Weight and   Sex   from  `7`   people
##   <chr>    <chr> <chr>   <chr>  <chr>     <chr>  <chr> <chr> <chr> <chr> <chr>
## 1 Name     Age   Height  Weight Sex       <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
## 2 Alex     25    177     57     F         <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
## 3 Lilly    31    <NA>    69     F         <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
## 4 Mark     --    190     83     M         <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
## 5 Oliver   52    179     75     M         <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
## 6 Martha   76    *       70     F         <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
## 7 Lucas    49    183     **     M         <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
## 8 Caroline 26    164     53     F         <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
```

```
read_table("data/Table4.txt");
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Name = col_character(),
##   Age = col_character(),
##   Height = col_character(),
##   Weight = col_character(),
##   Sex = col_character()
## )
```

```
## # A tibble: 7 x 5
##   Name     Age   Height Weight Sex
##   <chr>    <chr> <chr>  <chr>  <chr>
## 1 Alex     25    1,77   57     F
## 2 Lilly    31    <NA>   69     F
## 3 Mark     --    1,90   83     M
## 4 Oliver   52    1,79   75     M
## 5 Martha   76    *      70     F
## 6 Lucas    49    1,83   **     M
## 7 Caroline 26    1,64   53     F
```

```
read_table("data/Table5.txt");
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   `Name;Age;Height;Weight;Sex` = col_character()
## )
```

```
## # A tibble: 7 x 1
##   `Name;Age;Height;Weight;Sex`
##   <chr>
## 1 Alex;25;1,77;57;F
## 2 Lilly;31;NA;69;F
## 3 Mark;--;1,90;83;M
## 4 Oliver;52;1,79;75;M
## 5 Martha;76;;70;F
## 6 Lucas;49;1,83;**;M
## 7 Caroline;26;1,64;53;F
```

```
read_table("data/Table6.txt");
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Table = col_character(),
##   `2:` = col_character(),
##   `Name,` = col_character(),
##   `Age,` = col_character(),
##   `Height,` = col_character(),
##   Weight = col_character(),
##   and = col_character(),
##   Sex = col_character(),
##   from = col_character(),
##   `7` = col_character(),
##   people = col_character()
## )
```

```
## Warning: 106 parsing failures.
## row col   expected    actual              file
##   1  -- 11 columns 5 columns 'data/Table6.txt'
##   2  -- 11 columns 6 columns 'data/Table6.txt'
##   3  -- 11 columns 6 columns 'data/Table6.txt'
##   4  -- 11 columns 5 columns 'data/Table6.txt'
##   5  -- 11 columns 5 columns 'data/Table6.txt'
## ... ... .......... ......... .................
## See problems(...) for more details.
```

```
## # A tibble: 106 x 11
##    Table    `2:`  `Name,` `Age,` `Height,` Weight and   Sex   from  `7`   people
##    <chr>    <chr> <chr>   <chr>  <chr>     <chr>  <chr> <chr> <chr> <chr> <chr>
##   1 Name     Age   Height  Weight Sex       <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
##  2 Alex     25    177     57      F         @Boss  <NA>  <NA>  <NA>  <NA>  <NA>
##  3 Lilly    31    163     69      F         @Secr~ <NA>  <NA>  <NA>  <NA>  <NA>
##  4 Mark     23    190     83      M         <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
##  5 Oliver   52    179     75      M         <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
##  6 Martha   76    163     70      F         <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
##  7 Lucas    49    183     83      M         <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
##  8 Caroline 26    164     53      F         <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
##  9 Alex     25    177     57      F         <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
## 10 Lilly    31    163     69      F         <NA>   <NA>  <NA>  <NA>  <NA>  <NA>
## # ... with 96 more rows
```

```
read_csv("data/states1.csv");
```

```
## New names:
## * `` -> ...1

## Rows: 50 Columns: 9

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ...1
## dbl (8): Population, Income, Illiteracy, Life Exp, Murder, HS Grad, Frost, Area

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## # A tibble: 50 x 9
##    ...1     Population Income Illiteracy `Life Exp` Murder `HS Grad` Frost   Area
##    <chr>        <dbl>  <dbl>      <dbl>      <dbl>  <dbl>     <dbl> <dbl>  <dbl>
##  1 Alabama       3615   3624        2.1       69.0   15.1      41.3    20  50708
##  2 Alaska         365   6315        1.5       69.3   11.3      66.7   152 566432
##  3 Arizona       2212   4530        1.8       70.6    7.8      58.1    15 113417
##  4 Arkans~       2110   3378        1.9       70.7   10.1      39.9    65  51945
##   5 Califo~     21198   5114        1.1       71.7   10.3      62.6    20 156361
##   6 Colora~      2541   4884        0.7       72.1    6.8      63.9   166 103766
##   7 Connec~      3100   5348        1.1       72.5    3.1      56     139   4862
##   8 Delawa~       579   4809        0.9       70.1    6.2      54.6   103   1982
##   9 Florida      8277   4815        1.3       70.7   10.7      52.6    11  54090
## 10 Georgia      4931   4091        2         68.5   13.9      40.6    60  58073
## # ... with 40 more rows
```

```
read_csv("data/states2.csv");
```

```
## Rows: 50 Columns: 1
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ;Population;Income;Illiteracy;Life Exp;Murder;HS Grad;Frost;Area
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## Warning: One or more parsing issues, see `problems()` for details
```

```
## # A tibble: 50 x 1
##    `;Population;Income;Illiteracy;Life Exp;Murder;HS Grad;Frost;Area`
##    <chr>
##  1 Alabama;3615;3624;2,1;69,05;15,1;41,3;20;50708
##  2 Alaska;365;6315;1,5;69,31;11,3;66,7;152;566432
##  3 Arizona;2212;4530;1,8;70,55;7,8;58,1;15;113417
##  4 Arkansas;2110;3378;1,9;70,66;10,1;39,9;65;51945
##  5 California;21198;5114;1,1;71,71;10,3;62,6;20;156361
##  6 Colorado;2541;4884;0,7;72,06;6,8;63,9;166;103766
##  7 Connecticut;3100;5348;1,1;72,48;3,1;56;139;4862
##  8 Delaware;579;4809;0,9;70,06;6,2;54,6;103;1982
##  9 Florida;8277;4815;1,3;70,66;10,7;52,6;11;54090
## 10 Georgia;4931;4091;2;68,54;13,9;40,6;60;58073
## # ... with 40 more rows
```