第 2 章

CHAPTER 2

MATLAB 软件

数以百万计的工程师和科学家都在使用 MATLAB 分析和设计改变着世界的系统和产品。基于矩阵的 MATLAB 语言是世界上表示计算数学最自然的方式。可以使用内置图形可视化数据和深入了解数据。

2.1 MATLAB 数据类型

MATLAB 是“matrix laboratory”的缩写形式。MATLAB 主要用于处理整个的矩阵和数组，而其他编程语言大多逐个处理数值。

2.1.1 矩阵

在默认情况下，MATLAB 中的所有变量都是双精度矩阵，不需要显式声明变量类型。矩阵可以是多维的，通过在圆括号中使用基于 1 的索引访问。可以使用单个索引，按列方式或每个维度一个索引的方式来寻址矩阵元素。要创建一个矩阵变量，只需为其分配值即可，创建 2 × 2 矩阵 a 的示例代码如下。

```
>> a=[1,4;5 7]
a =
     1     4
     5     7
>> a(1,1)
ans =
     1
>> a(4)
ans =
     7
```

可以简单地对没有特殊语法定义的矩阵进行加、减、乘、除等运算，矩阵必须符合所请求的线性代数运算的正确大小。使用单引号后缀 ' 表示转置，而矩阵幂运算使用运算符 ^。

```
>> a'
ans =
     1     5
     4     7
>> b=a'*a
```

```
b =
    26    39
    39    65
>> c=a^2
c =
    21    32
    40    69
>> d=b+c
d =
    47    71
    79   134
```

在默认情况下，每个变量都属于数值变量。可以使用 zeros、ones、eye、rand 等函数将矩阵初始化为给定数值，它们分别为 0 矩阵、1 矩阵、单位矩阵（对角线全为 1）、随机数矩阵。使用 isnumeric 函数来识别数值变量，各函数的相应说明如下。

（1）zeros：将矩阵初始化为 0 矩阵。

（2）ones：将矩阵初始化为 1 矩阵。

（3）eye：将矩阵初始化为单位矩阵。

（4）rand、randn：将矩阵初始化为随机数矩阵。

（5）isnumeric：判断是否为数值类型的矩阵。

（6）isscalar：判断是否为标量值（即 1 × 1 矩阵）。

（7）size：返回矩阵大小。

2.1.2　元胞数组

元胞数组是 MATLAB 独有的一种变量类型，它实际上是一个列表容器，可以在数组元素中存储任何类型的变量。就像矩阵一样，元胞数组可以是多维的，并且在许多代码运行环境中都非常有用。

过去建议用元胞数组处理文本和不同类型的表格数据，如电子表格中的数据。现在建议用 string 数组存储文本数据，用 table 存储表格数据。而对于异构数据，最适合用元胞数组，这种数据最适合在数组中按位置引用。

可使用 {} 运算符或 cell 函数创建元胞数组。当将数据放入元胞数组时，使用元胞数组构造运算符 {}。

```
>> C = {1,2,3;
    'text',rand(5,10,2),{11; 22; 33}}
C =
  2×3 cell 数组
    {[   1]}    {[          2]}    {[     3]}
    {'text'}    {5×10×2 double}    {3×1 cell}
```

与所有 MATLAB 数组一样，元胞数组也是矩阵，每一行中具有相同的元胞数。C 是一个 2 × 3 元胞数组。可以使用 {} 运算符创建一个空的 0 × 0 元胞数组。

```
>> C2={}
C2 =
空的 0×0 cell 数组
```

当要随时间推移或以循环方式向元胞数组添加值时，可先使用 cell 函数创建一个空数组，这种方法会为元胞数组的头部预分配内存，每个元胞包含一个空数组 []。

```
>> C3=cell(3,4)
C3 =
  3×4 cell 数组
    {0×0 double}    {0×0 double}    {0×0 double}    {0×0 double}
    {0×0 double}    {0×0 double}    {0×0 double}    {0×0 double}
    {0×0 double}    {0×0 double}    {0×0 double}    {0×0 double}
```

要对特定元胞数组进行读取或写入，可将索引括在花括号中。例如，用随机数据数组填充 C3。根据数组在元胞数组中的位置更改数组大小。

```
>> for row = 1:3
   for col = 1:4
      C3{row,col} = rand(row*10,col*10);
   end
end
C3
C3 =
  3×4 cell 数组
    {10×10 double}    {10×20 double}    {10×30 double}    {10×40 double}
    {20×10 double}    {20×20 double}    {20×30 double}    {20×40 double}
    {30×10 double}    {30×20 double}    {30×30 double}    {30×40 double}
```

元胞数组对于字符串集特别有用，许多 MATLAB 字符串搜索函数针对元胞数组（如 strcmp）进行了优化，使用 iseell 来判断变量是否为元胞数组，使用 deal 来操作结构数组和元胞数组的内容，常用函数说明如下。

（1）cell：初始化元胞数组。

（2）cellstr：从字符数组中生成元胞数组。

（3）iscell：判断是否为元胞数组。

（4）iscellstr：判断元胞数组中是否只包含字符串。

（5）celldisp：递归显示元胞数组的内容。

2.1.3 结构体

当有要按名称组织的数据时，可以使用结构体来存储这些数据。结构体将数据存储在名为字段的容器中，然后可以按指定的名称访问这些字段。使用圆点表示法创建、分配和访问结构体字段中的数据。如果存储在字段中的值是数组，则可以使用数组索引来访问数组的元素。当将多个结构体存储为一个结构体数组时，可以使用数组索引和圆点表示法来访问单个结构体及其字段。

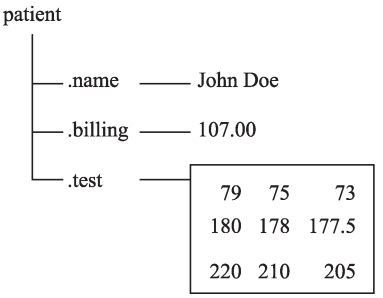

图 2-1 patient 的结构体

1. 创建标量结构体

创建一个名为 patient 的结构体，其中包含存储患者数据的字段。图 2-1 显示该结构体如何存储数据。像 patient 这样的结构体也被称为标量结构体，因为该变

量只存储一个结构体。

使用圆点表示法添加字段 name、billing 和 test，为每个字段分配数据。在实例中，语法 patient.name 创建结构体及其第一个字段。

```
>> patient.name = 'John Doe';
patient.billing = 107.00;
patient.test = [79 75 73; 180 178 177.5; 220 210 205]
patient =
包含以下字段的 struct:
       name: 'John Doe'
    billing: 107
       test: [3×3 double]
```

2. 访问字段中的值

创建字段后，可以继续使用圆点表示法来访问和更改它存储的值。例如，更改 billing 字段的值。

```
>> patient.billing = 502.00
patient =
包含以下字段的 struct:
       name: 'John Doe'
    billing: 502
       test: [3×3 double]
```

3. 对非标量结构体数组进行索引

结构体数组可以是非标量的。可以创建任意大小的结构体数组，只要数组中的每个结构体都有相同的字段即可。

例如，向 patients 添加第二个结构体，其中包含第二个患者的有关数据。此外，将原始值 107 赋给第一个结构体的 billing 字段。由于该数组现在有两个结构体，必须通过索引来访问第一个结构体，如 patient(1).billing = 107.00 所示。

```
>> patient(2).name = 'Ann Lane';
patient(2).billing = 28.50;
patient(2).test = [68 70 68; 118 118 119; 172 170 169];
patient(1).billing = 107.00
patient =
包含以下字段的 1×2 struct 数组：
    name
    billing
    test
```

因此，patient 是 1 × 2 结构体数组，其内容如图 2-2 所示。

数组中的每条患者记录都是 struct 类的结构体。由结构体构成的数组有时可称为结构体数组。与其他 MATLAB 数组类似，结构体数组可以具有任意维度，其具有下列属性。

（1）数组中的所有结构体都具有相同数目的字段。

（2）所有结构体都具有相同的字段名称。

（3）不同结构体中的同名字段可包含不同类型或大小的数据。

（4）如果向数组中添加新结构体而未指定其所有字段，则未指定的字段包含空数组。

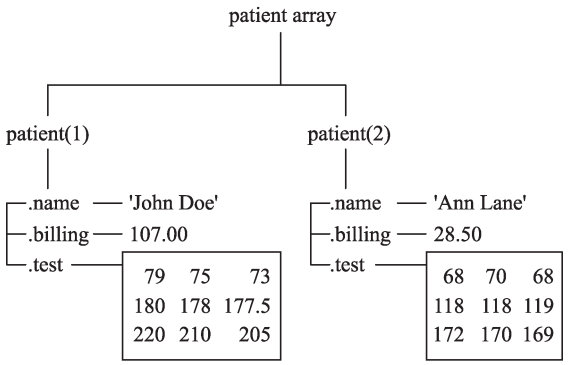

图 2-2　添加新内容后的 patient 结构体

```
>> patient(3).name = 'New Name';
patient(3)
ans =
包含以下字段的 struct:
       name: 'New Name'
    billing: []
       test: []
```

要对结构体数组进行索引，可使用数组索引。例如，patient(2) 返回第二个结构体：

```
>> patient(2)
ans =
包含以下字段的 struct:
       name: 'Ann Lane'
    billing: 28.5000
       test: [3×3 double]
```

要访问字段，可使用数组索引和圆点表示法。例如，返回第二个患者的 billing 字段的值。

```
>> patient(2).billing
ans =
28.5000
```

2.1.4　数据存储

1. 什么是数据存储

数据存储相当于一个存储库，用来存储具有相同结构和格式的数据。例如，数据存储中每个文件包含的数据必须具有相同的类型（如数字或文本）、以相同顺序显示并用相同的分隔符分隔。

在以下两种情况下，数据存储很有用。

（1）集合中的每个文件可能太大，无法放入内存中。数据存储允许从每个文件可放入内存的较小部分中读取和分析数据。

（2）集合中的文件具有任意名称。数据存储充当一个或多个文件夹中文件的存储库，这

些文件不需要具有序列名称。

可根据数据或应用程序的类型来创建数据存储。不同类型的数据存储包含与其支持的数据类型相关的属性，如表 2-1 所示。

表 2-1　数据存储类型

文件或数据的类型	数据存储类型
包含列向数据的文本文件，包括 CSV 文件	TabularTextDatastore
图像文件，包括 imread 支持的格式，如 JPEG 和 PNG	ImageDatastore
具有支持的 Excel 格式（如 .xlsx）的电子表格文件	SpreadsheetDatastore
作为 mapreduce 的输入或输出的键 - 值对组数据	KeyValueDataStore
包含列向数据的 Parquet 文件	ParquetDatastore
自定义文件格式。需要提供用于读取数据的函数	FileDatastore
用于存放 tall 数组的检查点的数据存储	TallDatatore

2. 创建和读取数据存储

使用 tabularTextDatastore 函数从实例文件 airlinesmall.csv 创建一个数据存储，其中包含有关各航空公司航班的出发和到达信息。结果是一个 TabularTextDatastore 对象。

```
>> s = tabularTextDatastore('airlinesmall.csv')
s =
  TabularTextDatastore - 属性：
                      Files: { ' ...\MATLAB\SupportPackages\R2023b\
examples\matlab\data\airlinesmall.csv'}
                    Folders: {                                'C:\
ProgramData\MATLAB\SupportPackages\R2023b\examples\matlab\data'
                             }
               FileEncoding: 'UTF-8'
   AlternateFileSystemRoots: {}
         VariableNamingRule: 'modify'
          ReadVariableNames: true
              VariableNames: {'Year', 'Month', 'DayofMonth' ... and 26
more}
             DatetimeLocale: en_US
```

文本格式属性：

```
             NumHeaderLines: 0
                  Delimiter: ','
               RowDelimiter: '\r\n'
             TreatAsMissing: ''
               MissingValue: NaN
```

高级文本格式属性：

```
            TextscanFormats: {'%f', '%f', '%f' ... and 26 more}
                   TextType: 'char'
          ExponentCharacters: 'eEdD'
               CommentStyle: ''
```

```
                  Whitespace: ' \b\t'
    MultipleDelimitersAsOne: false
```

控制 preview、read、readall 返回表的属性：

```
      SelectedVariableNames: {'Year', 'Month', 'DayofMonth' ... and 26
more}
            SelectedFormats: {'%f', '%f', '%f' ... and 26 more}
                   ReadSize: 20000 行
                 OutputType: 'table'
                   RowTimes: []
```

特定于写入的属性：

```
     SupportedOutputFormats: ["txt"   "csv"   "xlsx"   "xls"   "parquet"
"parq"]
        DefaultOutputFormat: "txt"
```

创建数据存储后，可以预览数据而无须将其全部加载到内存中。可以使用 SelectedVariableNames 属性指定相关变量（列），以预览或只读这些变量，效果如图 2-3 所示。

A	B	C	D	E	F	G	H	I	J	K	L	
						airlinesmall						
Year	Month	DayofMo...	DayOfWeek	DepTime	CRSDepTi...	ArrTime	CRSArrTime	UniqueCar...	FlightNum	TailNum	ActualElap...	CR
数值	数值	数值	数值	数值	数值	数值	数值	分类	数值	分类	数值	数
1987	10	23	5	2055	2035	2218	2157	PS	1589	NA	83	82
1987	10	23	5	1332	1320	1431	1418	PS	1655	NA	59	58
1987	10	22	4	629	630	746	742	PS	1702	NA	77	72
1987	10	28	3	1446	1343	1547	1448	PS	1729	NA	61	65
1987	10	8	4	928	930	1052	1049	PS	1763	NA	84	79
1987	10	10	6	859	900	1134	1123	PS	1800	NA	155	14
1987	10	20	2	1833	1830	1929	1926	PS	1831	NA	56	56
1987	10	15	4	1041	1040	1157	1155	PS	1864	NA	76	75
1987	10	15	4	1608	1553	1656	1640	PS	1907	NA	48	47
1987	10	21	3	949	940	1055	1052	PS	1939	NA	66	72
1987	10	22	4	1902	1847	2030	1951	PS	1973	NA	88	64
1987	10	16	5	1910	1838	2052	1955	TW	19	NA	162	13
1987	10	2	5	1130	1133	1237	1237	TW	59	NA	187	18
1987	10	30	5	1400	1400	1920	1934	TW	102	NA	200	21
1987	10	28	3	841	830	1233	1218	TW	136	NA	172	16

图 2-3　预览 airlinesmall.csv 数据

```
>> s.SelectedVariableNames = {'DepTime','DepDelay'};
preview(s)
ans =
  8×2 table
    DepTime    DepDelay
    _______    ________
      642          12
     1021           1
     2055          20
     1332          12
      629          -1
     1446          63
      928          -2
      859          -1
```

可以指定数据中表示缺失值的值。在 airlinesmall.csv 中，缺失值由 NA 表示。

```
>> ds.TreatAsMissing = 'NA';
```

如果相关变量的数据存储中的所有数据都放入内存，则可以使用 readall 函数读取。

```
>> T = readall(s);
```

否则，使用 read 函数读取放入内存的较小子集中的数据。在默认情况下，read 函数一次从 TabularTextDatastore 读取 20000 行。但是，可以通过为 ReadSize 属性分配一个新值来更改此值。

```
>> s.ReadSize = 15000;
```

在重新读取之前，使用 reset 函数将数据存储重置为初始状态。通过在 while 循环中调用 read 函数，可以对每个数据子集执行中间计算，然后在结束时汇总中间结果。以下代码计算 DepDelay 变量的最大值。

```
>> reset(s)
X = [];
while hasdata(s)
      T = read(s);
      X(end+1) = max(T.DepDelay);
end
maxDelay = max(X)
maxDelay =
        1438
```

如果每个单独文件中的数据都能放入内存，则可以指定每次调用 read 时应读取一个完整文件，而不是特定行数。

```
>> reset(s)
s.ReadSize = 'file';
X = [];
while hasdata(s)
      T = read(s);
      X(end+1) = max(T.DepDelay);
end
maxDelay = max(X);
```

除了读取数据存储中的数据子集，还可以使用 mapreduce 将 map 和 reduce 函数应用于数据存储，或使用 tall 函数创建一个 tall 数组。

2.1.5　tall 数组

tall 数组是 MATLAB R2016b 发行版中的新功能，它允许数组中拥有超出内存大小的更多的行，可以使用它来处理可能有数百万行的数据存储。tall 数组可以使用几乎任意 MATLAB 类型作为列变量，包括数值数据、元胞数组、字符串、时间和分类数据。MATLAB 文档提供了支持 tall 数组的函数列表。仅当使用 gather 函数显式请求在数组上的操作结果时才会对其进行求值。histogram 函数可以与 tall 数组一起使用，并将立即执行。

统计和机器学习工具箱、数据库工具箱、并行计算工具箱、分布式计算服务器和编译器都提供了额外的扩展来处理 tall 数组。

表 2-2 列出了创建和计算 tall 数组的相关函数。

表 2-2　创建和计算 tall 数组的相关函数

函　　数	说　　明
tall	创建 tall 数组
datastore	为大型数据集合创建数据存储
gather	执行排队的运算后，将 tall 数组收集到内存中
write	将 tall 数组写入本地和远程位置以设置检查点
mapreduce	为 mapreduce 或 tall 数组定义执行环境
tallrng	控制 tall 数组的随机数生成

调用 tall 函数创建 tall 数组的语法为：

```
t = tall(ds)                                    % 基于数据存储 ds 创建一个 tall 数组
```

如果 ds 是用于表格数据的数据存储（以使数据存储的 read 和 readall 方法返回表），则 t 是一个 tall 表，具体取决于数据存储配置为返回哪种类型。表格数据是以矩形方式排列且每一行具有相同条目数的数据。否则，t 为一个 tall 元胞数组。

t = tall(A)：将内存数组 A 转换为 tall 数组。t 与 class(A) 具有相同的基本数据类型。当需要快速创建一个 tall 数组时，如对算法进行调试或原型构建时，该语法非常有用。

在 MATLAB R2019b 及更高版本中，可以将内存数组转换为 tall 数组，对数组执行更高效的运算。在转换为 tall 数组后，MATLAB 可免于生成整个数组的临时副本，而是以较小的分块处理数据。这让 MATLAB 能够对数组执行各种运算，而不会耗尽内存。

【例 2-1】创建 tall 数组并进行计算。

```
% 为 airlinesmall.csv 数据集创建数据存储。将 'NA' 值视为缺失数据，以使它们被替换为 NaN 值。
% 此处选择要使用的变量的小型子集
>> varnames = {'ArrDelay', 'DepDelay', 'Origin', 'Dest'};
ds = tabularTextDatastore('airlinesmall.csv', 'TreatAsMissing', 'NA', ...
    'SelectedVariableNames', varnames);
>> % 使用 tall 函数为数据存储中的数据创建一个 tall 数组。由于 ds 中的数据是表格数据，因此
% 结果是一个 tall 表。如果数据不是表格数据，则 tall 函数将改为创建一个 tall 元胞数组
>> T = tall(ds)
Starting parallel pool (parpool) using the 'Processes' profile ...
Connected to parallel pool with 4 workers.
T =
  M×4 tall table
    ArrDelay    DepDelay    Origin      Dest
    ________    ________    _______    _______
        8           12       {'LAX'}    {'SJC'}
        8            1       {'SJC'}    {'BUR'}
       21           20       {'SAN'}    {'SMF'}
       13           12       {'BUR'}    {'SJC'}
        4           -1       {'SMF'}    {'LAX'}
       59           63       {'LAX'}    {'SJC'}
        3           -2       {'SAN'}    {'SFO'}
       11           -1       {'SEA'}    {'LAX'}
        :            :           :          :
```

```
        :            :            :            :
>>% 计算 tall 表的大小。由于计算 tall 数组的大小需要完全遍历数据，因此 MATLAB 不会立即计算
% 该值。而是与 tall 数组的大多数运算一样，结果是一个未计算的 tall 数组
>> s = size(T)
s =
  1×2 tall double 行向量
    ?     ?
```

使用 gather 函数可执行延迟计算并在内存中返回结果。size 返回的结果是一个非常小的 1×2 向量，适合放在内存中。

```
>> sz = gather(s)
正在使用 Parallel Pool 'rocesses' 计算 tall 表达式：
- 第 1 次遍历（共 1 次）：用时 9.5 秒
计算已完成，用时 12 秒
sz =
      123523           4
```

如果对未约简的 tall 数组使用 gather，则结果可能不适合放在内存中。如果不确定 gather 返回的结果是否适合放在内存中，可使用 gather(head(X)) 或 gather(tail(X))，只将一小部分计算结果放入内存中。

2.1.6　稀疏矩阵

稀疏矩阵中的大多数元素为 0，它属于一种特殊类别的矩阵，通常出现在大型优化问题中，并被许多工具箱使用。矩阵中的 0 被“挤压”出来，MATLAB 只存储非 0 元素及其索引数据，使得完整矩阵仍然可以重新创建。许多常规 MATLAB 函数（如 chol 或 diag）保留输入矩阵的稀疏性。

在 MATLAB 中，提供了相关函数用于创建稀疏矩阵，如表 2-3 所示。

表 2-3　创建稀疏矩阵的函数

函　　数	说　　明
spalloc	为稀疏矩阵分配空间
spdiags	提取非零对角线并创建稀疏带状对角矩阵
speye	稀疏单位矩阵
sprand	稀疏均匀分布随机矩阵
sprandn	稀疏正态分布随机矩阵
sprandsym	创建稀疏矩阵
sparse	从稀疏矩阵外部格式导入
spconvert	直接建立稀疏存储矩阵

MATLAB 从不会自动创建稀疏矩阵，相反，还必须确定矩阵中是否包含足够高百分比的零元素，以便利用稀疏方法。

1. 将满矩阵转换为稀疏矩阵

可以使用带有单个参数的 sparse 函数将满矩阵转换为稀疏矩阵，例如：

```
>> A = [ 0   0   0   5
      0   2   0   0
      1   3   0   0
      0   0   4   0];
S = sparse(A)
S =
   (3,1)         1
   (2,2)         2
   (3,2)         3
   (4,3)         4
   (1,4)         5
```

在输出结果中，列出了 S 的非零元素及其行索引和列索引。这些元素按列排序，反映了内部数据结构体。

扩展：如果矩阵阶数不太高，可以使用 full 函数将稀疏矩阵转换为满矩阵，例如，A = full(S)。

2. 直接创建稀疏矩阵

可以使用带有 5 个参数的 sparse 函数，基于一列非零元素来创建稀疏矩阵。

```
S = sparse(i,j,s,m,n)
```

i 和 j 分别是矩阵中非零元素的行索引和列索引的向量，s 是由对应的 (i,j) 对指定索引的非零值的向量，m 是生成的矩阵的行维度，n 是其列维度。

例 2-1 中的矩阵 S 可以直接通过以下代码生成。

```
>> S = sparse([3 2 3 4 1],[1 2 2 3 4],[1 2 3 4 5],4,4)
S =
   (3,1)         1
   (2,2)         2
   (3,2)         3
   (4,3)         4
   (1,4)         5
```

sparse 函数具有许多备用形式。上面代码使用的形式将矩阵中的最大非零元素数设置为 length(s)。如果需要，可以追加第六个参数用来指定更大的最大数，这样能在以后添加非零元素，而不必重新分配稀疏矩阵。

二阶差分算子的矩阵表示形式是一个很好的稀疏矩阵实例。实质上，生成稀疏矩阵的方式有多种，下面演示其中的一种。

```
>> n = 5;
D = sparse(1:n,1:n,-2*ones(1,n),n,n);
E = sparse(2:n,1:n-1,ones(1,n-1),n,n);
S = E+D+E'
S =
   (1,1)        -2
   (2,1)         1
   (1,2)         1
```

```
    (2,2)        -2
    (3,2)         1
    (2,3)         1
    (3,3)        -2
    (4,3)         1
    (3,4)         1
    (4,4)        -2
    (5,4)         1
    (4,5)         1
    (5,5)        -2
```

用 F = full(S) 可显示相应的满矩阵。

```
>> F = full(S)
F =
    -2     1     0     0     0
     1    -2     1     0     0
     0     1    -2     1     0
     0     0     1    -2     1
     0     0     0     1    -2
IdleTimeout has been reached.
Parallel pool using the 'Processes' profile is shutting down.
```

3. 基于稀疏矩阵的对角线元素创建稀疏矩阵

基于稀疏矩阵的对角线元素创建稀疏矩阵是一种常用操作，函数 spdiags 可以处理此任务。其语法格式为：

```
S = spdiags(B,d,m,n)              % 创建大小为 m × n 且元素在 p 对角线上的输出矩阵 S
```

（1）B 是大小为 min(m,n) × p 的矩阵，B 的列是用于填充 S 对角线的值。

（2）d 是长度为 p 的向量，其整数元素可以指定要填充的 S 对角线。

【例 2-2】利用 spdiags 函数创建稀疏矩阵。

```
% 考虑使用矩阵 B 和向量 d
>> B=[41 11 0;52 22 0;63 33 13;74 44 24];
>> d=[-3 0 2];
>> % 使用这些矩阵创建 7 × 4 稀疏矩阵 A
>> A = spdiags(B,d,7,4)
A =
    (1,1)        11
    (4,1)        41
    (2,2)        22
    (5,2)        52
    (1,3)        13
    (3,3)        33
    (6,3)        63
    (2,4)        24
    (4,4)        44
    (7,4)        74
>> % 在其满矩阵形式中，A 类似于
>> full(A)
```

```
ans =
    11     0    13     0
     0    22     0    24
     0     0    33     0
    41     0     0    44
     0    52     0     0
     0     0    63     0
     0     0     0    74
```

2.1.7 表与分类数组

1. 表数组

表（table）是 MATLAB R2013b 发行版本中引入的一个新数据结构，它允许将表格数据与元数据共同存储在一个工作区变量中。表格中的列可以被命名、分配单元和描述，并作为数据结构中的一个字段来访问，即 T. DataName。

要对表进行索引，可以使用圆括号返回子表，或者使用花括号提取内容，还可以使用名称访问变量和行。

在 MATLAB 中，可利用 table 函数根据现有的电子表格变量创建一个表。函数的语法格式为：

T = table(var1,...,varN)：根据输入变量 var1,...,varN 创建表，变量的大小和数据类型可以不同，但所有变量的行数必须相同。

如果输入是工作区变量，则 table 将输入名称指定为输出表中的变量名称。否则，table 将指定 'Var1',...,'VarN' 形式的变量名称，其中 N 是变量的数量。

T = table('Size',sz,'VariableTypes',varTypes)：创建一个表并为具有指定的数据类型的变量预分配空间。sz 是二元素数值数组，其中 sz(1) 指定行数，sz(2) 指定变量数。varTypes 指定变量的数据类型。

T = table(___,Name,Value)：使用一个或多个名称 - 值对组参数指定其他输入参数。例如，可以使用 'VariableNames' 名称 - 值对组指定变量名称。可将此语法与上述语法中的任何输入参数一起使用。

T = table：创建一个空的 0 × 0 表。

【例 2-3】在表中存储相关数据变量，并以矩阵形式访问所有表数据。

```
>> clear all;
>>% 创建包含患者数据的工作区变量。这些变量可以具有任何数据类型，但必须具有相同的行数
LastName = {'Sanchec';'Johnson';'chen';'Diaz';'Brown'};
Age = [38;43;37;38;49];
Smoker = logical([1;0;1;0;1]);
Height = [72;69;65;67;64];
Weight = [175;162;134;133;129];
BloodPressure = [125 93; 119 77; 125 83; 117 75; 121 80];
% 创建一个表 T，作为工作区变量的容器。table 函数使用工作区变量名称作为 T 中表变量的名称，一
% 个表变量可以有多个列。例如，T 中的 BloodPressure 变量是一个 5 × 2 数组
T = table(LastName,Age,Smoker,Height,Weight,BloodPressure)
T =
  5×6 table
```

```
      LastName       Age    Smoker    Height    Weight    BloodPressure
    ____________    ___    ______    ______    ______    _____________
    {'Sanchec'}     38     true        72        175       125     93
    {'Johnson'}     43     false       69        162       119     77
    {'chen'   }     37     true        65        134       125     83
    {'Diaz'   }     38     false       67        133       117     75
    {'Brown'  }     49     true        64        129       121     80
>> % 可以使用点索引来访问表变量。例如，使用 T.Height 中的值计算患者的平均身高
>> meanHeight = mean(T.Height)
meanHeight =
   67.4000
% 计算身体质量指数（Body Mass Index，BMI），并将其添加为新的表变量，还可以使用圆点在一
% 个步骤中添加和命名表变量
>> T.BMI = (T.Weight*0.453592)./(T.Height*0.0254).^2
T =
  5×7 table
  LastName       Age    Smoker    Height    Weight    BloodPressure    BMI
  _________      ___    _____     _____     ______    _____________    _____
  {'Sanchec'}     38     true        72       175       125     93     23.734
  {'Johnson'}     43     false       69       162       119     77     23.923
  {'chen'   }     37     true        65       134       125     83     22.299
  {'Diaz'   }     38     false       67       133       117     75     20.831
  {'Brown'  }     49     true        64       129       121     80     22.143
>>% 使用 BMI 计算的描述对表进行注释，可以使用通过 T.Properties 访问的元数据来对 T 及其变量
% 进行注释
>> T.Properties.Description = 'Patient data, including body mass index (BMI)
calculated using Height and Weight';
T.Properties
ans =
  TableProperties - 属性：
             Description: 'Patient data, including body mass index (BMI)
calculated using Height and Weight'
                UserData: []
          DimensionNames: {'Row'  'Variables'}
           VariableNames: {'LastName'  'Age'  'Smoker'  'Height'  'Weight'
'BloodPressure'  'BMI'}
    VariableDescriptions: {}
           VariableUnits: {}
      VariableContinuity: []
                 RowNames: {}
        CustomProperties: 未设置自定义属性
% 使用 DimensionNames 属性显示表的维度名称，第二个维度的默认名称是 Variables
>> T.Properties.DimensionNames
ans =
  1×2 cell 数组
    {'Row'}    {'Variables'}
%% 语法 T.Variables 以矩阵形式访问表数据。此语法等同于使用花括号语法 T{:,:} 访问所有内容。
% 如果表数据不能串联为一个矩阵，则会产生错误消息
>> T.Variables
错误使用　.
```

表无法串联表变量 'LastName' 和 'Age'，因为这两个变量的类型为 cell 和 double。

2. 分类数组

分类数组允许存储离散的非数值数据，并且经常在表中用于定义行组。例如，时间数据可以按照星期几来分组，地理数据可以按照州或县来组织。它们可以使用 unstack 在表中重新排列数据。

例如，语法 C = categorical({'R','G','B','B','G','B'}) 将创建一个分类数组，此数组包含 6 个属于类别 R、G 或 B 的元素。

分类数组可用来有效地存储并方便地处理非数值数据，同时还可为数据值赋予有意义的名称。这些分类可以采用自然排序，但并不要求一定如此。

【例 2-4】创建分类数组。

```
%% 基于字符串数组创建分类数组 %%
>> % 创建一个包含新英格兰各州名称的 1×11 字符串数组
state = ["MA","ME","CT","VT","ME","NH","VT","MA","NH","CT","RI"]
state =
  1×11 string 数组
    "MA"  "ME"  "CT"  "VT"  "ME"  "NH"  "VT"  "MA"  "NH"  "CT"  "RI"
>> % 将字符串数组 state 转换为无数学排序的分类数组
state = categorical(state)
state =
  1×11 categorical 数组
  MA    ME    CT    VT    ME    NH    VT    MA    NH    CT    RI
>> % 列出变量 state 中的离散类别
categories(state)
ans =
  6×1 cell 数组
    {'CT'}
    {'MA'}
    {'ME'}
    {'NH'}
    {'RI'}
    {'VT'}
>> %% 添加新元素和缺失的元素 %%
% 向原始字符串数组添加元素，其中一个元素是缺失字符串，显示为 <missing>
state = ["MA","ME","CT","VT","ME","NH","VT","MA","NH","CT","RI"];
state = [string(missing) state];
state(13) = "ME"
state =
  1×13 string 数组
<missing>  "MA"   "ME"    "CT"    "VT"    "ME"    "NH"    "VT"    "MA"
"NH"    "CT"    "RI"    "ME"
>> % 将字符串数组转换为 categorical 数组
state = categorical(state)
state =
  1×13 categorical 数组
<undefined>  MA  ME  CT  VT  ME  NH  VT  MA  NH  CT  RI  ME
%% 基于字符串数组创建有序分类数组 %%
% 创建一个包含 8 个对象的 1×8 字符串数组
```

```
>> AllSizes = ["medium","large","small","small","medium",...
            "large","medium","small"];  % 字符串数组 AllSizes 包含 "large"、
%"medium" 和 "small" 三个不同值
>> valueset = ["small","medium","large"];
sizeOrd = categorical(AllSizes,valueset,'Ordinal',true)
sizeOrd =                                % 分类数组 sizeOrd 中值的顺序保持不变
  1×8 categorical 数组
   medium    large    small    small   medium    large    medium    small
```

从结果可看出，分类数组 sizeOrd 中值的顺序保持不变。

```
>> % 列出分类变量 sizeOrd 中的离散类别
categories(sizeOrd)
ans =
  3×1 cell 数组
    {'small' }
    {'medium'}
    {'large' }
```

这些类别按指定的顺序列出以匹配数学排序 small < medium < large。

```
>> %% 基于 bin 创建有序分类数组
% 创建由 0 到 50 的 100 个随机数构成的向量
x = rand(100,1)*50;
% 使用 discretize 函数，通过对 x 的值进行分 bin，创建一个分类数组。将 0 到 15 的所有数归入第
% 一个 bin，15 到 35 的所有数归入第二个 bin，35 到 50 的所有数归入第三个 bin。每个 bin 包括左端点，
% 但不包括右端点
>> catnames = ["small","medium","large"];
binnedData = discretize(x,[0 15 35 50],'categorical',catnames);
>> % 使用 summary 函数输出每个类别中的元素数量
summary(binnedData)
     small       30
     medium      35
     large       35
```

2.1.8　大型 MAT 文件

使用 matfile 函数可直接从磁盘上的 MAT 文件访问 MATLAB 变量，而不必将全部变量都载入内存。当使用 matfile 创建新文件时，该函数将创建一个版本为 7.3 的 MAT 文件，后者还允许保存大小超过 2GB 的变量。

【例 2-5】使用 matfile 函数保存和加载。

解析：实例说明如何使用 matfile 函数在现有 MAT 文件中加载、修改和保存变量的一部分。

```
% 创建具有两个变量 A 和 B 的 7.3 版 MAT 文件
A = rand(5);
B = magic(10);
save example.mat A B -v7.3;
clear A B
```

基于 MAT 文件 example.mat 构造一个 MatFile 对象（exampleObject）。matfile 函数创建一个对应于 MAT 文件的 MatFile 对象，并包含 MatFile 对象的属性。默认情况下，matfile 仅允许从现有 MAT 文件加载。

```
>> exampleObject = matfile('example.mat');
```

要启用保存，需使用 Writable 参数调用 matfile。

```
>> exampleObject = matfile('example.mat','Writable',true);
```

构造该对象并在单独的步骤中设置 Properties.Writable。

```
>>exampleObject = matfile('example.mat');
exampleObject.Properties.Writable = true;
```

将 B 的第一行从 example.mat 加载到变量 firstRowB 中，并修改数据。当索引与 7.3 版的 MAT 文件关联的对象时，MATLAB 仅加载所指定的变量部分。

```
>> firstRowB = exampleObject.B(1,:);
firstRowB = 2 * firstRowB;
```

使用存储在 firstRowB 中的值更新 example.mat 中变量 B 的第一行中的值。

```
>> exampleObject.B(1,:) = firstRowB;
```

对于非常大的文件，最佳做法是应一次将尽可能多的数据读取和写入到内存中。否则，重复的文件访问会严重影响代码的性能。例如，假设文件包含许多行和列，并且加载一行就需要占用大部分可用内存。这种情况下不要一次更新一个元素，而应该更新一行。

```
>> [nrowsB,ncolsB] = size(exampleObject,'B');
for row = 1:nrowsB
  exampleObject.B(row,:) = row * exampleObject.B(row,:);
end
```

如果内存大小不是问题，则可以一次更新一个变量的完整内容。

```
>> exampleObject.B = 10 * exampleObject.B;
```

或者，通过使用 -append 选项调用 save 函数来更新变量。-append 选项要求 save 函数仅替换指定变量 B，并保留文件中的其他变量不变。此方法始终要求加载并保存整个变量。

```
>> load('example.mat','B');
B(1,:) = 2 * B(1,:);
save('example.mat','-append','B');
```

使用 matlab.io.MatFile 对象向文件中添加变量。

```
>> exampleObject.C = magic(8);
```

还可以通过使用 -append 选项调用 save 函数来添加变量。

```
>>C = magic(8);
save('example.mat','-append','C');
clear C
```

2.2　MATLAB 作图

强大的绘图功能是 MATLAB 的特点之一。MATLAB 可以给出数据的二维、三维乃至四维的图形表现，MATLAB 提供了两个层次的绘图操作：对图形句柄进行的低层图形命令与建立在低层绘图操作之上的高层绘图操作。高层绘图操作简单明了、方便高效，是用户最常用的绘图方法；而低层绘图操作和表现能力更强，为用户更加自主地绘制图形创造了条件。

2.2.1　二维线图

在 MATLAB 中，提供了许多相关函数用于绘制二维线图，其中 plot 是一个具有代表性的函数。根据输入数据对 plot 函数进行创建。

1. 向量和矩阵数据

当输入的数据是向量或矩阵时，plot 函数的语法格式如下。

plot(X,Y)：创建 Y 中数据对应 X 中值的二维线图。

① 要绘制由线段连接的一组坐标，需将 X 和 Y 指定为相同长度的向量。

② 要在同一组坐标区上绘制多组坐标，需将 X 或 Y 中的至少一个数据指定为矩阵。

plot(X,Y,LineSpec)：使用指定的线型、标记和颜色创建绘图。

plot(X1,Y1,...,Xn,Yn)：在同一组坐标轴上绘制多对 X 和 Y 坐标。此语法可替代将坐标指定为矩阵的形式。

plot(X1,Y1,LineSpec1,...,Xn,Yn,LineSpecn)：可为每个 X-Y 对组指定特定的线型、标记和颜色。可以对某些 X-Y 对组指定特定的线型，而对其他对组省略它。例如，plot(X1,Y1,"o",X2,Y2) 对第一个 X-Y 对组指定标记，但没有对第二个对组指定标记。

plot(Y)：绘制 Y 对一组隐式 X 坐标的图。

plot(Y,LineSpec)：使用隐式 X 坐标绘制 Y，并指定线型、标记和颜色。

【例 2-6】利用输入的向量或矩阵数据绘制线图。

```
>> clear all;
% 将 y 创建为 x 的正弦值，创建数据的线图
x = 0:pi/100:2*pi;
y = sin(x);
subplot(2,2,1);plot(x,y);
title(' 绘制一个线条 ')
x = linspace(-2*pi,2*pi);
y1 = sin(x);
y2 = cos(x);
subplot(2,2,2);
plot(x,y1,x,y2)
title(' 创建多个线条 ')
% 绘制三条正弦曲线，第一条正弦曲线使用绿色线条，不带标记
% 第二条正弦曲线使用蓝色虚线，带圆形标记。第三条正弦曲线只使用青蓝色星号标记
x = 0:pi/10:2*pi;
y1 = sin(x);
y2 = sin(x-0.25);
y3 = sin(x-0.5);
subplot(2,2,3);
```

```
plot(x,y1,'g',x,y2,'b--o',x,y3,'c*')
title('带标记的三条正弦曲线')
% 创建线图并使用 LineSpec 选项指定带正方形标记的绿色虚线
x = -pi:pi/10:pi;
y = tan(sin(x)) - sin(tan(x));
subplot(2,2,4);
plot(x,y,'--gs',...
    'LineWidth',2,...              % 使用 Name,Value 对组指定线宽、标记大小和标记颜色
    'MarkerSize',10,...
    'MarkerEdgeColor','b',...
    'MarkerFaceColor',[0.5,0.5,0.5])
title('指定线宽、标记大小和标记颜色')
```

运行程序，效果如图 2-4 所示。

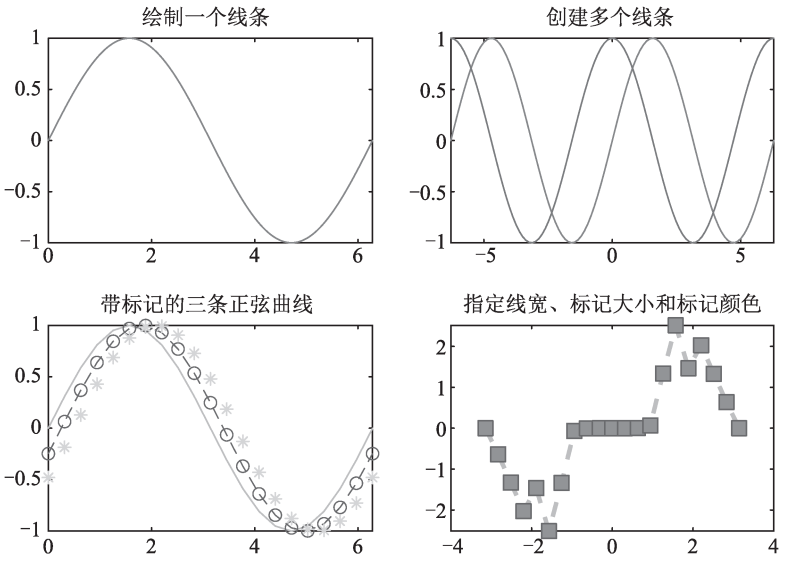

图 2-4　向量与矩阵输入的线图

2. 表数据

如果输入数据为表数据，则 plot 函数的调用格式如下。

plot(tbl,xvar,yvar)：绘制表 tbl 中的变量 xvar 和 yvar。要绘制一个数据集，需为 xvar 和 yvar 各指定一个变量。要绘制多个数据集，需为 xvar、yvar 或两者指定多个变量。如果两个参数都指定多个变量，它们的变量数目必须相同。

plot(tbl,yvar)：绘制表中的指定变量对表的行索引的图。如果该表是时间表，则绘制指定变量对时间表的行时间的图。

【例 2-7】以时间表形式读取文件 quarterlyFinances1999To2019.csv 中的数据。

```
% 将连续性时间之间的时间长度指定为一个日历季度，从 1999 年 1 月 1 日开始。将
%'VariableNamingRule' 设置为 preserve 以保留变量名称中的空白，并将
%'TrimNonNumeric' 设置为 true 以删除数据中数值前的 "$" 符号
>>tbl=readtimetable("quarterlyFinances1999To2019.csv","TimeStep",
calquarters(1),"StartTime", datetime(1999, 1, 1),...
```

```
    "VariableNamingRule", "preserve", "TrimNonNumeric", true);
summary(tbl)
plot(tbl,"Research and Development Expenses");
ylabel('研发费用 / 元 ')
xlabel('时间 ')
```

运行程序，输出如下，效果如图 2-5 所示。

```
RowTimes:
    Time: 80×1 datetime
        Values:
            Min         1999-01-01
            Median      2008-11-16
            Max         2018-10-01
Variables:
    Net Sales: 80×1 double
        Values:
            Min            35066
            Median    1.0407e+05
            Max       1.7684e+05
    Cost of Sales: 80×1 double
        Values:
            Min        18106
            Median     48624
            Max        77742
    Gross Margin: 80×1 double
        Values:
            Min        14563
            Median     56719
            Max        99097
    Research and Development Expenses: 80×1 double
        Values:
            Min        4904.9
            Median      24637
            Max         45234
    Administrative Expenses: 80×1 double
        Values:
            Min        1047.4
            Median     2015.3
            Max        2811.5
    Total Operating Expenses: 80×1 double
        Values:
            Min        5992.5
            Median      26518
            Max         48045
    Net Income: 80×1 double
        Values:
            Min        7634.3
            Median      28586
            Max         51051
    Total Shares: 80×1 double
```

```
        Values:
            Min           822
            Median     1820.5
            Max          2710
    Earnings per Share: 80×1 double
        Values:
            Min          6.52
            Median     15.515
            Max         24.62
```

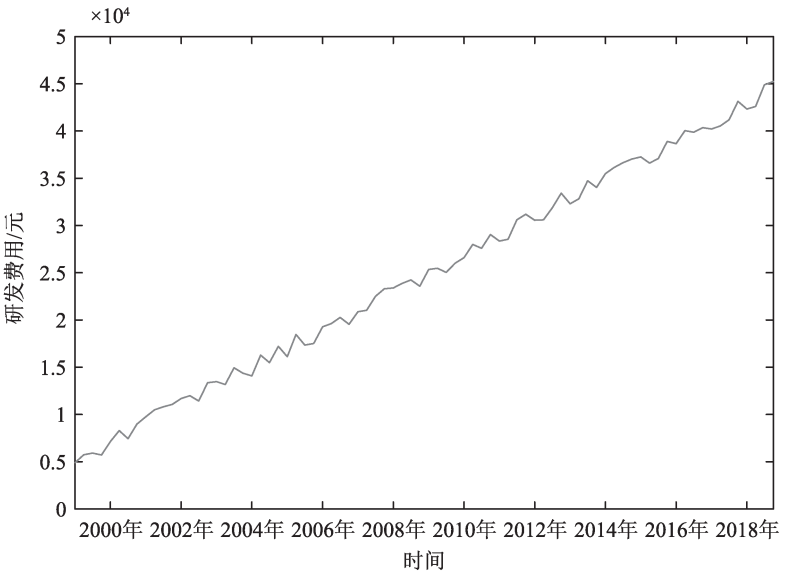

图 2-5　利用表数据绘制线图

3. 其他选项

输入除了向量、矩阵、表数据外，其他选项形式的语法格式如下。

plot(ax,___)：在目标坐标区上显示绘图。将坐标区指定为上述任一语法中的第一个参数。

plot(___,Name,Value)：使用一个或多个名称 - 值参数指定 Line 属性。这些属性应用于绘制的所有线条。需要在上述任一语法中的所有参数之后指定名称 - 值参数。

p = plot(___)：返回一个 Line 对象或 Line 对象数组。创建绘图后，使用 p 修改该绘图的属性。

【例 2-8】绘制以点 (4,3) 为中心、以 2 为半径的圆。

解析：使用 axis equal 可沿每个坐标方向使用相等的数据单位。

```
>> r = 2;
xc = 4;
yc = 3;

theta = linspace(0,2*pi);
x = r*cos(theta) + xc;
y = r*sin(theta) + yc;
plot(x,y)
axis equal
```

运行程序，效果如图 2-6 所示。

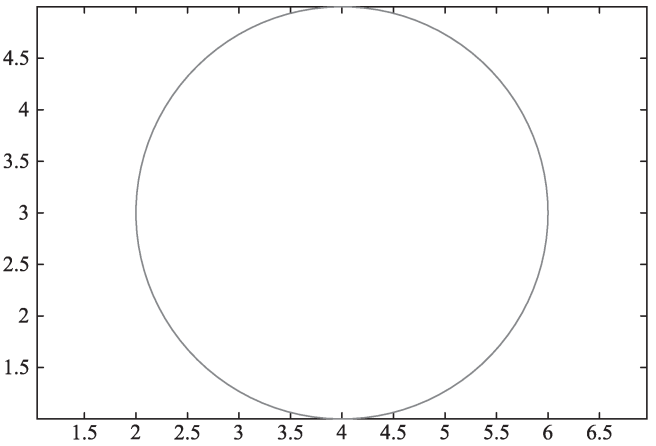

图 2-6　绘制圆

2.2.2　通用二维图形

除了 plot 函数外，MATLAB 还提供了一系列用于绘制通用二维图形的函数，下面直接通过一个实例来演示各函数的用法。

【例 2-9】绘制通用二维图形。

```
>> %bar 函数用来创建垂直条形图
x = -2.9:0.2:2.9;
y = exp(-x.*x);
subplot(2,3,1);bar(x,y);
title(' 条形图 ')
%stairs 函数用来创建阶梯图，它可以创建仅含 y 值的阶梯图，或同时包含 x 和 y 值的阶梯图
x = 0:0.25:10;
y = sin(x);
subplot(2,3,2);stairs(x,y)
title(' 阶梯图 ')

%errorbar 函数用来绘制 x 和 y 值的线图，并在每个观察点上叠加垂直误差条
x = -2:0.1:2;
y = erf(x);
eb = rand(size(x))/7;
subplot(2,3,3);errorbar(x,y,eb)
title(' 误差条 ')

%polarplot 函数用来绘制 theta 中的角度值（以弧度为单位）对 rho 中的半径值的极坐标图
theta = 0:0.01:2*pi;
rho = abs(sin(2*theta).*cos(2*theta));
subplot(2,3,4);polarplot(theta,rho)
title(' 极坐标图 ')

%stem 函数为每个通过竖线连接到一条公共基线的 x 和 y 值绘制一个标记
```

```
x = 0:0.1:4;
y = sin(x.^2).*exp(-x);
subplot(2,3,5);stem(x,y)
title('针状图')

%scatter 函数用来绘制 x 和 y 值的散点图
load patients Height Weight Systolic
subplot(2,3,6);scatter(Height,Weight)
xlabel('体重 / 千克')
ylabel('身高 / 厘米')
title('散点图')
```

运行程序，效果如图 2-7 所示。

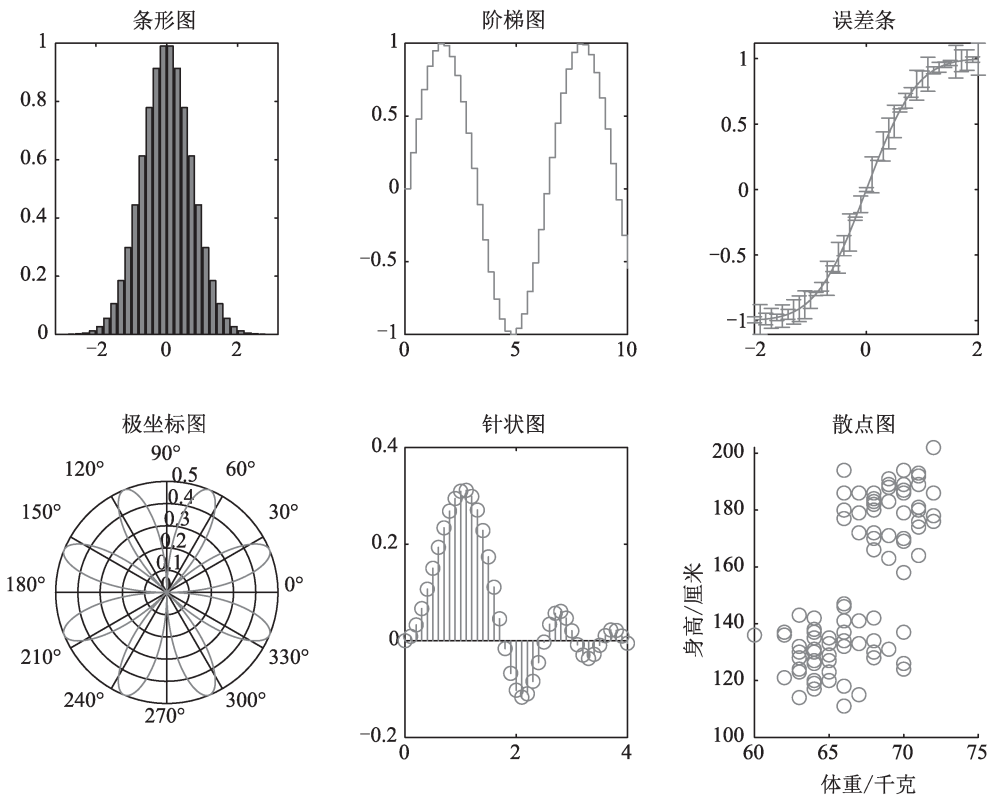

图 2-7　通用二维图形

2.2.3　三维点或线图

与二维线图一样，在 MATLAB 中，也提供了相关函数绘制三维点或线图，即 plot3 函数。其输入数据分以下几种形式。

1. 向量和矩阵数据

plot3(X,Y,Z)：绘制三维空间中的坐标。

① 要绘制由线段连接的一组坐标，需将 X、Y、Z 指定为相同长度的向量。

② 要在同一组坐标轴上绘制多组坐标，需将 X、Y 或 Z 中的至少一个数据指定为矩阵，其他指定为向量。

plot3(X,Y,Z,LineSpec)：使用指定的线型、标记和颜色创建绘图。

plot3(X1,Y1,Z1,...,Xn,Yn,Zn)：在同一组坐标轴上绘制多组坐标。使用此语法作为将多组坐标指定为矩阵的替代方法。

plot3(X1,Y1,Z1,LineSpec1,...,Xn,Yn,Zn,LineSpecn)：可为每个（X、Y、Z）三元组指定特定的线型、标记和颜色。还可以对某些三元组指定特定的线型，而对其他三元组省略它。例如，plot3(X1,Y1,Z1,'o',X2,Y2,Z2) 对第一个三元组指定标记，但没有对第二个三元组指定标记。

2. 表数据

plot3(tbl,xvar,yvar,zvar)：绘制表 tbl 中的变量 xvar、yvar 和 zvar。要绘制一个数据集，需为 xvar、yvar 和 zvar 各指定一个变量。要绘制多个数据集，需为其中至少一个参数指定多个变量。对于指定多个变量的参数，指定的变量数目必须相同。

3. 其他选项

plot3(ax,___)：在目标坐标区上显示绘图。将坐标区指定为上述任一语法中的第一个参数。

plot3(___,Name,Value)：使用一个或多个名称 - 值对组参数指定 Line 属性。在所有其他输入参数后指定属性。

p = plot3(___)：返回一个 Line 对象或 Line 对象数组。创建绘图后，使用 p 修改该绘图的属性。

【例 2-10】绘制三维点或线图。

```
>> % 指定线型。创建向量 t，然后使用 t 计算两组 x 和 y 值
t = 0:pi/20:10*pi;
xt1 = sin(t);
yt1 = cos(t);
% 绘制这两组值。第一组使用默认线条，第二组使用虚线
xt2 = sin(2*t);
yt2 = cos(2*t);
subplot(121);plot3(xt1,yt1,t,xt2,yt2,t,'--')
% 指定等间距刻度单位和轴标签
t = 0:pi/500:40*pi;
% 创建向量 xt、yt 和 zt
xt = (3 + cos(sqrt(32)*t)).*cos(t);
yt = sin(sqrt(32) * t);
zt = (3 + cos(sqrt(32)*t)).*sin(t);
% 绘制数据，使用 axis equal 命令沿每个轴等间距隔开刻度单位，并为每个轴指定标签
subplot(122);plot3(xt,yt,zt)
axis equal
xlabel('x(t)')
ylabel('y(t)')
zlabel('z(t)')
```

运行程序，效果如图 2-8 所示。

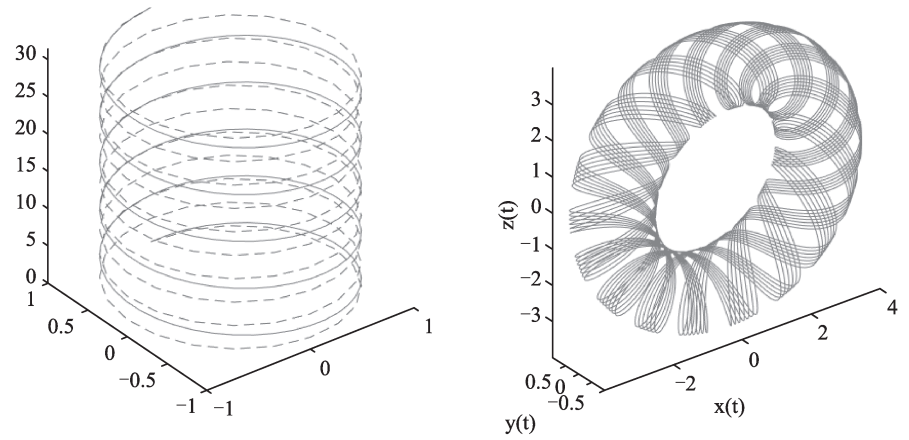

图 2-8　三维点或线图

2.2.4　通用三维图形

与二维绘图一样，在 MATLAB 中，也提供了一系列用于绘制通用三维图形的函数，下面也是直接通过一个实例来演示各函数的用法。

【例 2-11】绘制通用三维图形。

```
>> clear all;
[X,Y] = meshgrid(-8:.5:8);                % 设置 x 和 y 平面的网格，产生一个横纵坐标起始
                                          % 于 -8，终止于 8，且步距为 .5 的网格图形
R = sqrt(X.^2 + Y.^2) + eps;
Z = sin(R)./R;                            % 计算曲面的 z 矩阵
subplot(2,4,1);mesh(X,Y,Z)                % 画三维网格图；
title('三维网格图')
subplot(2,4,2);surf(X,Y,Z)                % 画三维曲面图
title('三维曲面图')
t=0:pi/20:2*pi;
% 标准三维曲面图形
[x,y,z]= cylinder(2+sin(t),30);
subplot(2,4,3);surf(x,y,z);
title('花瓶')
[x,y,z]=sphere;
subplot(2,4,4);surf(x,y,z);
title('球体')
subplot(2,4,5);bar3(magic(4))
title('三维柱状图')
y=2*sin(0:pi/10:2*pi);
subplot(2,4,6);stem3(y);
title('三维杆图')
subplot(2,4,7);pie3([2347,1827,2043,3025]);
title('三维饼图')
subplot(2,4,8);fill3(rand(3,5),rand(3,5),rand(3,5), 'y' )
title('三维填充形')
```

运行程序，效果如图 2-9 所示。

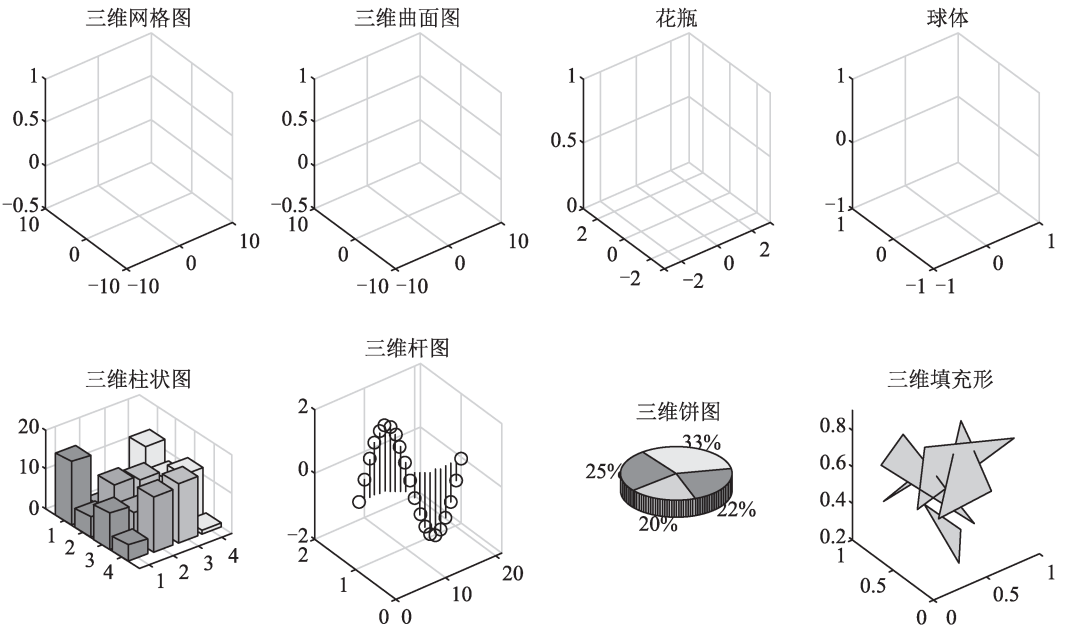

图 2-9　三维通用图