第三章 数据可视化

1. 直方图
2. 条形图/柱状图
3. 散点图
4. 饼状图
5. 箱型图

数据可视化：

数据可视化是指信息和数据的图形化表示。使用图形等可视化元素，数据可视化是查看和了解数据中趋势、异常值和模式的便利方式。在大数据领域，数据可视化工具和技术对于分析海量信息并制定数据驱动型决策而言至关重要。

1. 直方图

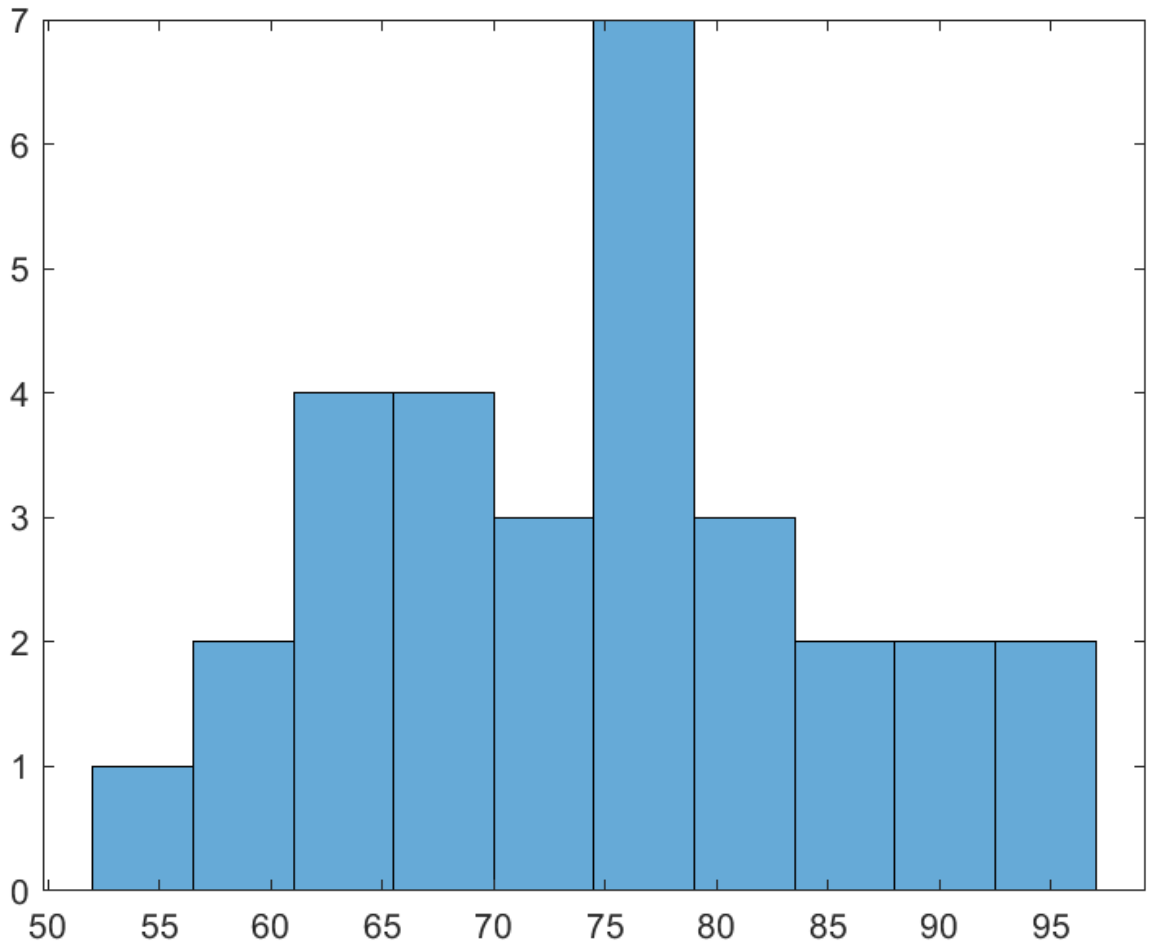
直方图(histogram)，又称质量分布图，是一种统计报告图，它是根据具体数据的分布情况，画成以组距为底边、以频数为高度的一系列连接起来的直方型矩形图，用以展示数据的分布情况，表示不同数据出现的频率。

例2-1.随机抽取30名大学生，得到某课程的考试分数数据如下：

59,77,97,60,88,63,64,65,75,67,67,69,73,70,66,76,76,54,77,85,78,78,79,61,83,93,86,91,71,80.

给出其成绩分布的直方图

1. 直方图



直方图

1. 直方图

Matlab实现，一般用横轴表示数据类型，纵轴表示分布情况。直方图的绘制通过histogram函数实现。

```
x=[59,77,97,60,88,63,64,65,75,67,67,69,73,70,66,
76,76,54,77,85,78,78,79,61,83,93,86,91,71,80];

nbins = 10;

h = histogram(x,nbins)
```

1. 直方图

Python实现，一般用横轴表示数据类型，纵轴表示分布情况。直方图的绘制通过pyplot中的hist()实现。

Pyplot.hist(x, bins = 10, color = None, range = None, rwidth = None, normed = False, orientation = u’vertical’, **kwargs)

1. 直方图

hist的主要参数如下。

x: 这个参数是arrays，指定bin（箱子）分布在x的位置。

bins: 这个参数指定bin（箱子）的个数，也就是总共有几条条状图。

normed:是否对y轴数据进行标准化（如果为True，则是在本区间的点在所有的点中所占的概率）。

1. 直方图

color:这个指定条状图（箱子）的颜色。

range:指定上下界，即最大值和最小值。

kwargs参数主要用于设置图形要素的属性。

1. 直方图

由大数定律知，f_i 作为总体 X 的观测值 $x_1, x_2, \cdots, x_n$ 中落入第 i 个小区间（t_{i-1}，t_i]内的频率应近似于(**Convergent in probability**) X 落入第 i 个小区间（t_{i-1}，t_i]的概率。

即应有（若 X 是连续型随机变量）

$$f_i \approx P\{t_{i-1} < X \le t_i\} = \int_{t_{i-1}}^{t_i} f(x)dx$$

其中 $f(x)$ 为 X 的密度函数，从而

$$f_i \approx f(\xi_i)\Delta t_i \quad , \quad f_i / \Delta t_i \approx f(\xi_i)$$

由此可见，频率直方图的上部轮廓线即是 X 的密度函数的良好近似。

2. 条形图/柱状图

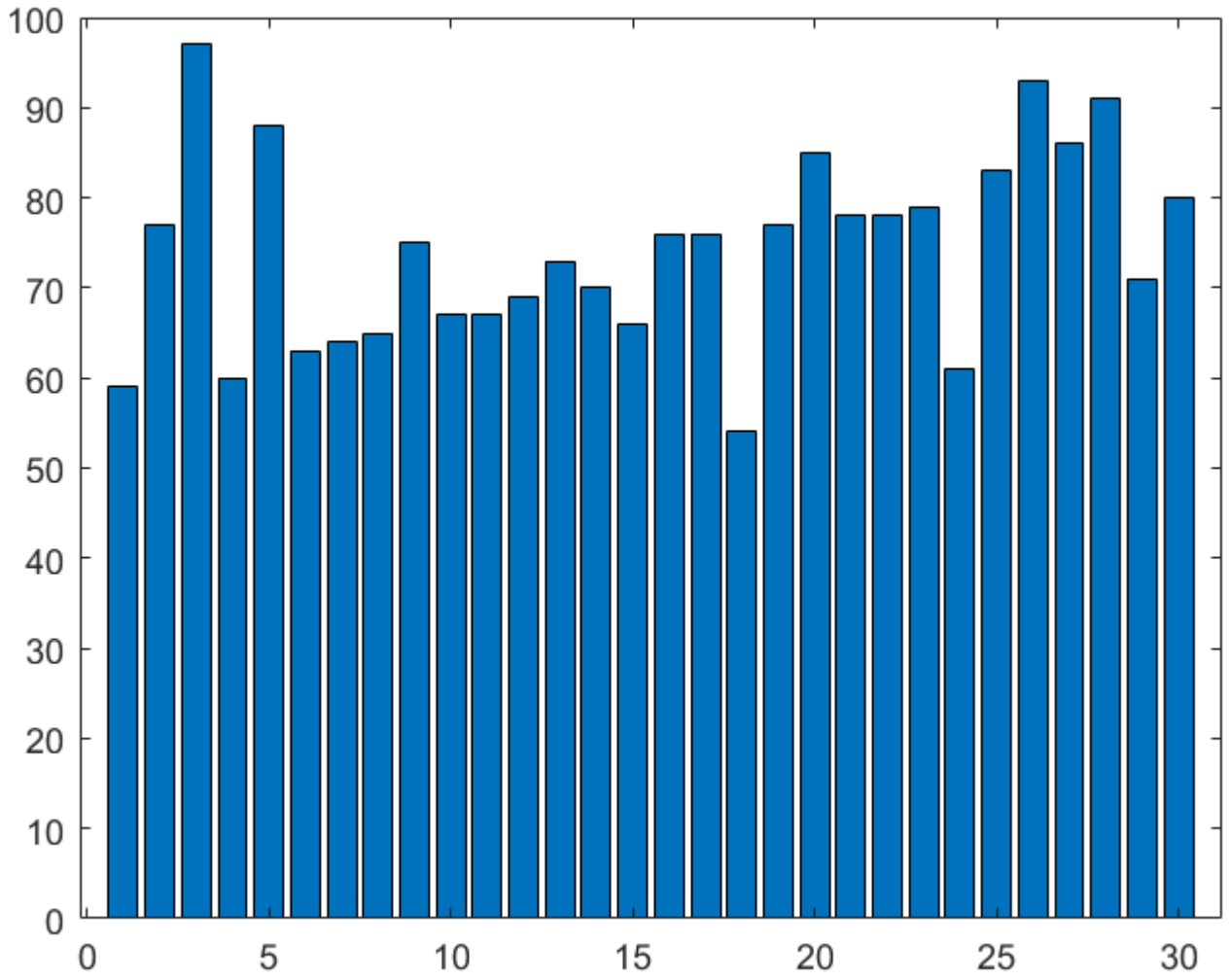
条形图是用一个单位长度表示一定的数量，根据数量的多少画成长短不同的直条，然后把这些直条按一定的顺序排列起来。

例2-1.随机抽取30名大学生，得到某课程的考试分数数据如下：

59,77,97,60,88,63,64,65,75,67,67,69,73,70,66,76,76,54,77,85,78,78,79,61,83,93,86,91,71,80.

给出30名学生成绩对比的条形图

2. 条形图/柱状图



条形图

1. 直方图

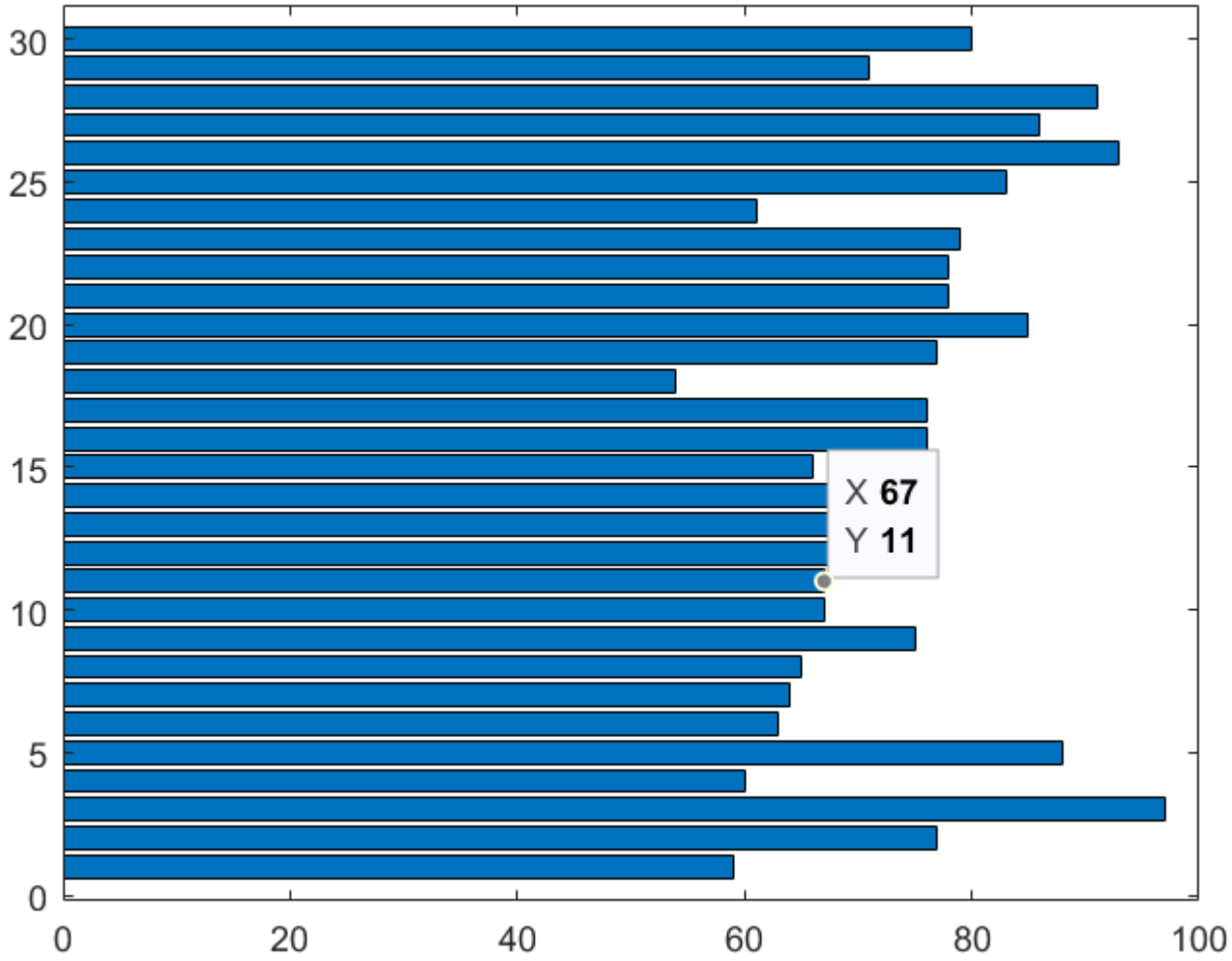
Matlab实现，一般用横轴表示数据类型，纵轴表示分布情况。直方图的绘制通过bar或barh函数实现。

```
x=[59,77,97,60,88,63,64,65,75,67,67,69,73,70,66,
76,76,54,77,85,78,78,79,61,83,93,86,91,71,80];

h = bar(x)

h = barh(x)
```

2. 条形图/柱状图



水平条形图

2. 条形图/柱状图

Python实现，条形图的绘制通过pyplot中的bar()或者是barh()来实现。

bar默认是绘制竖直方向的条形图。

barh就是绘制水平方向的条形图。

2. 条形图/柱状图

条形图的构造函数如下：

plt.bar(x, height, width, *, align = ‘center’ , **kwargs)

其中，x是包含所有条形的下表的列表。

height是包含所有条形的高度值的列表。

width表示每个条形的宽度。Align表示条形的对齐方式。

3. 散点图

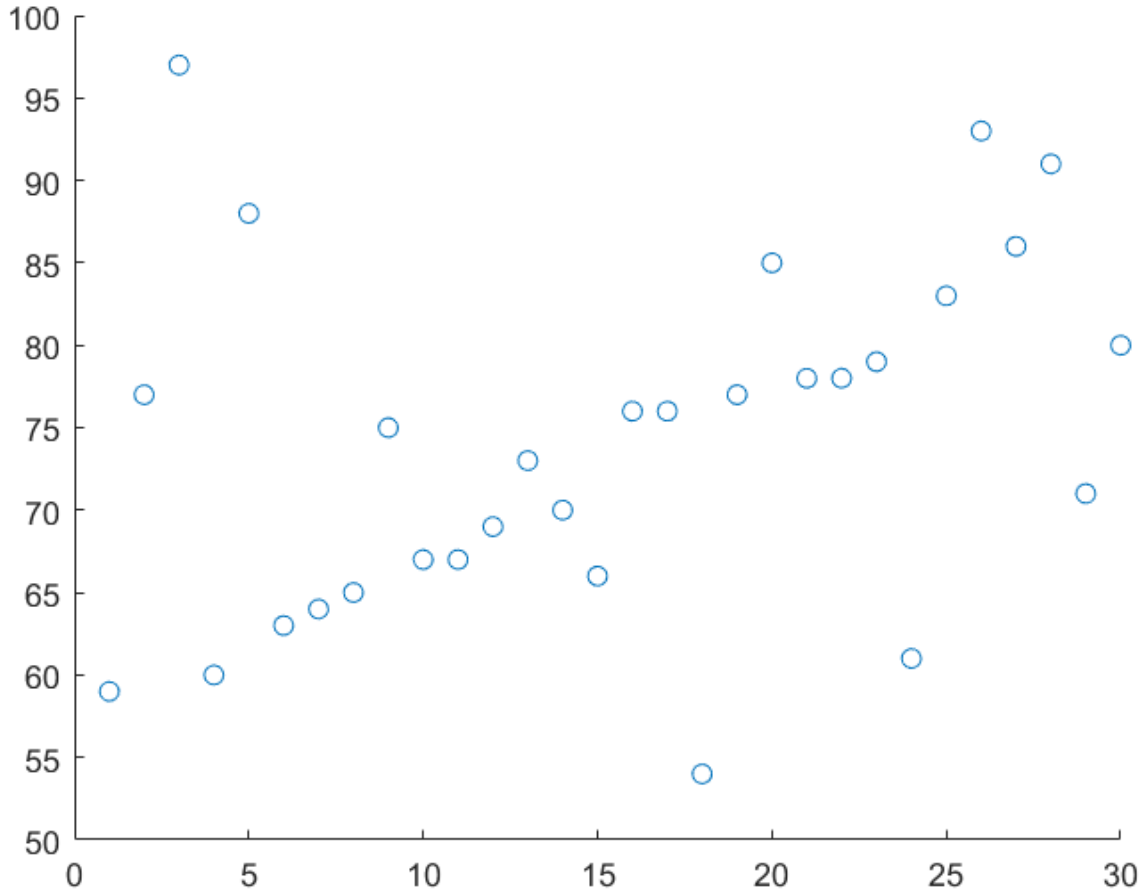
散点图(scatter diagram)，是一系列数据点在直角坐标系平面上的分布图。一般用两组数据构成多个坐标点，考查坐标点的分布，判断两变量之间是否存在某种关联或总结坐标点的分布模式。

例2-1.随机抽取30名大学生，得到某门课程的考试分数数据如下：

59,77,97,60,88,63,64,65,75,67,67,69,73,70,66,76,76,54,77,85,78,78,79,61,83,93,86,91,71,80

给出30名学生成绩的散点图

3. 散点图



散点图

3. 散点图

Matlab实现，一般用横轴表示数据x坐标类型，纵轴表示数据y坐标。散点图的绘制通过scatter函数实现。

```
num=1:30;

x=[59,77,97,60,88,63,64,65,75,67,67,69,73,70,66,
76,76,54,77,85,78,78,79,61,83,93,86,91,71,80];

h = scatter(num,x)
```

3. 散点图

Python实现，使用pyplot中的scatter()绘制散点图。

plt.scatter( )

4. 饼形图

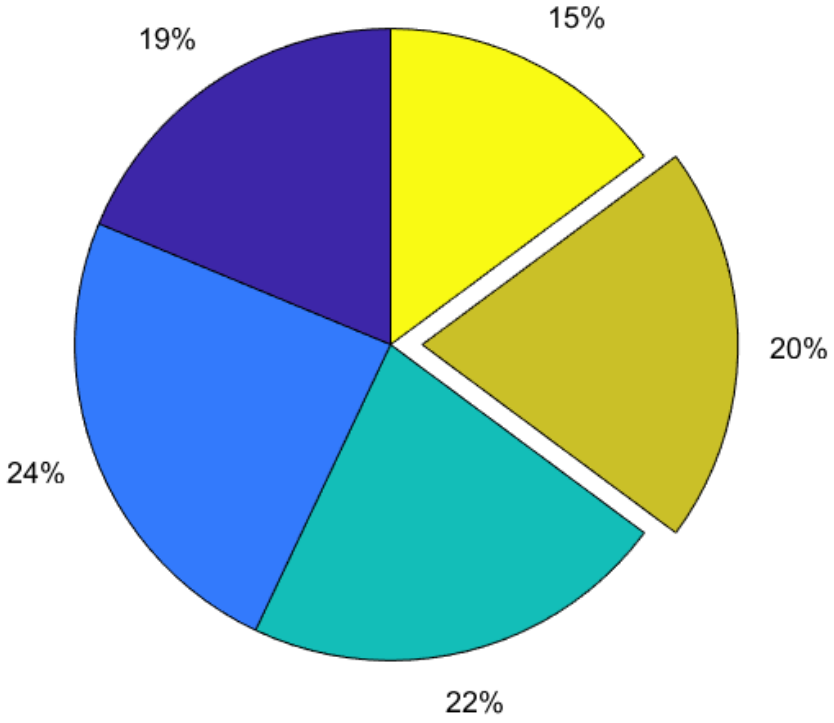
饼状图(sector graph，又名pie graph)显式一个数据系列中各项的大小与各项总和的比例。饼状图中的数据点显式为整个饼状图的百分比。

例3-1.已知某专业5各班级的人数分别为：

24、31、28，26、19

画出5个班级的人数的饼形图。

4. 饼形图



饼形图

4. 饼形图

Matlab实现，饼形图的绘制通过pie函数实现。

```
x=[24,31,28,26,19];

pie(x,[0 0 0 1 0])
```

4. 饼形图

Python实现，使用pyplot中的pie()绘制饼状图。

plt.pie(x, explode = None, labels = None, colors = None, autopct = None, shadow = False, startangle = None)

其中，x为数值列表，explode指定饼图中突出的分片，labels设置各个分片的标签， colors设置各个分片的颜色，autopct设置标签中的数字格式，shadow设置是否有阴影，startangle设置从哪个角度开始绘制圆饼。

5. 箱型图

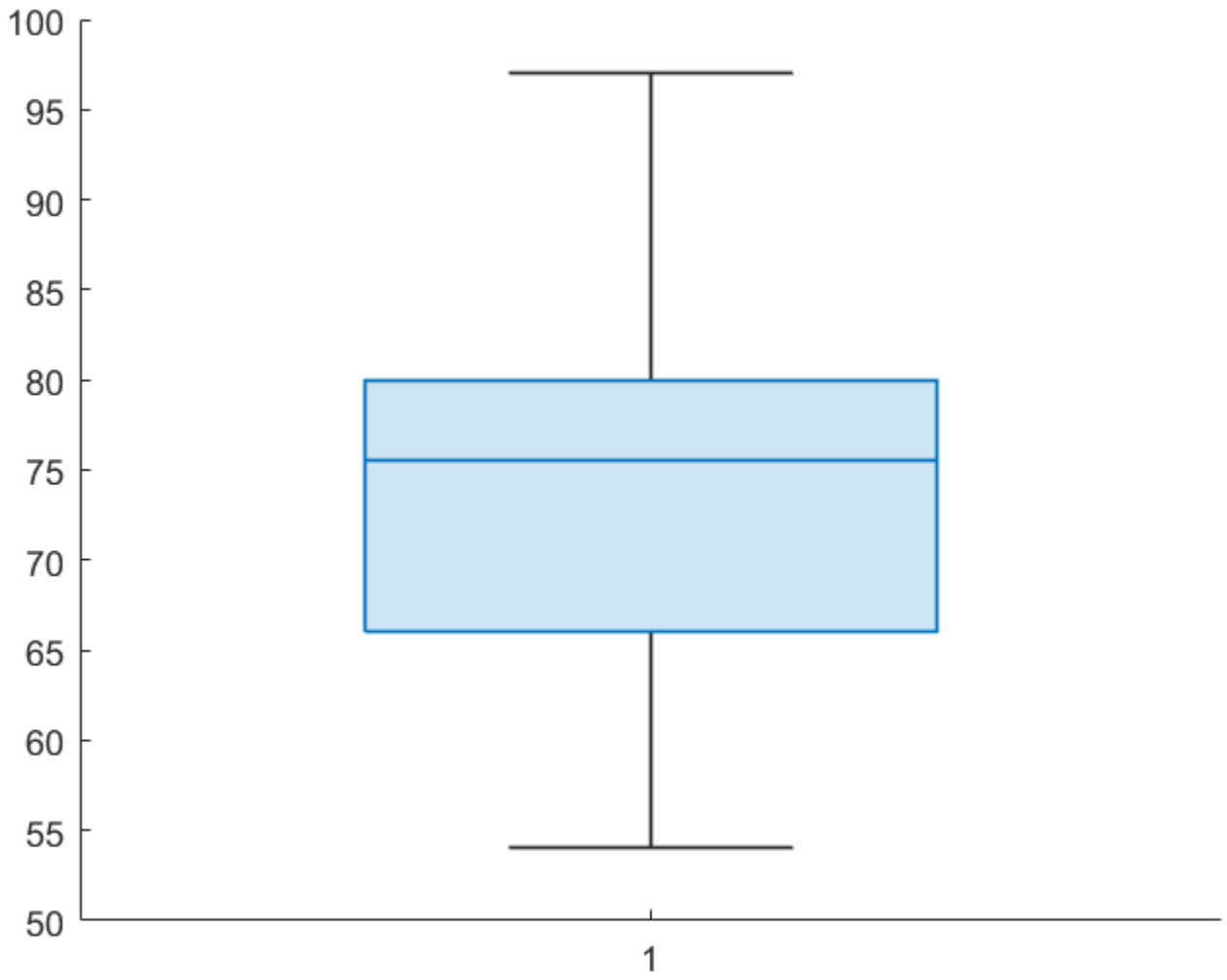
箱形图（Box-plot）又称为盒须图、盒式图或箱线图，是一种用作显示一组数据分散情况资料的统计图。箱线图的绘制方法是：先找出一组数据的上边缘、下边缘、中位数和两个四分位数；然后， 连接两个四分位数画出箱体；再将上边缘和下边缘与箱体相连接，中位数在箱体中间。

例2-1.随机抽取30名大学生，得到某门课程的考试分数数据如下：

59,77,97,60,88,63,64,65,75,67,67,69,73,70,66,76,76,54,77,85,78,78,79,61,83,93,86,91,71,80

给出30名学生成绩的箱型图

5. 箱型图



饼形图

5. 箱型图

Matlab实现，饼形图的绘制通过boxchart函数实现。

```
x=[59,77,97,60,88,63,64,65,75,67,67,69,73,70,66,
76,76,54,77,85,78,78,79,61,83,93,86,91,71,80];

h = boxchart(x)
```

5. 箱型图

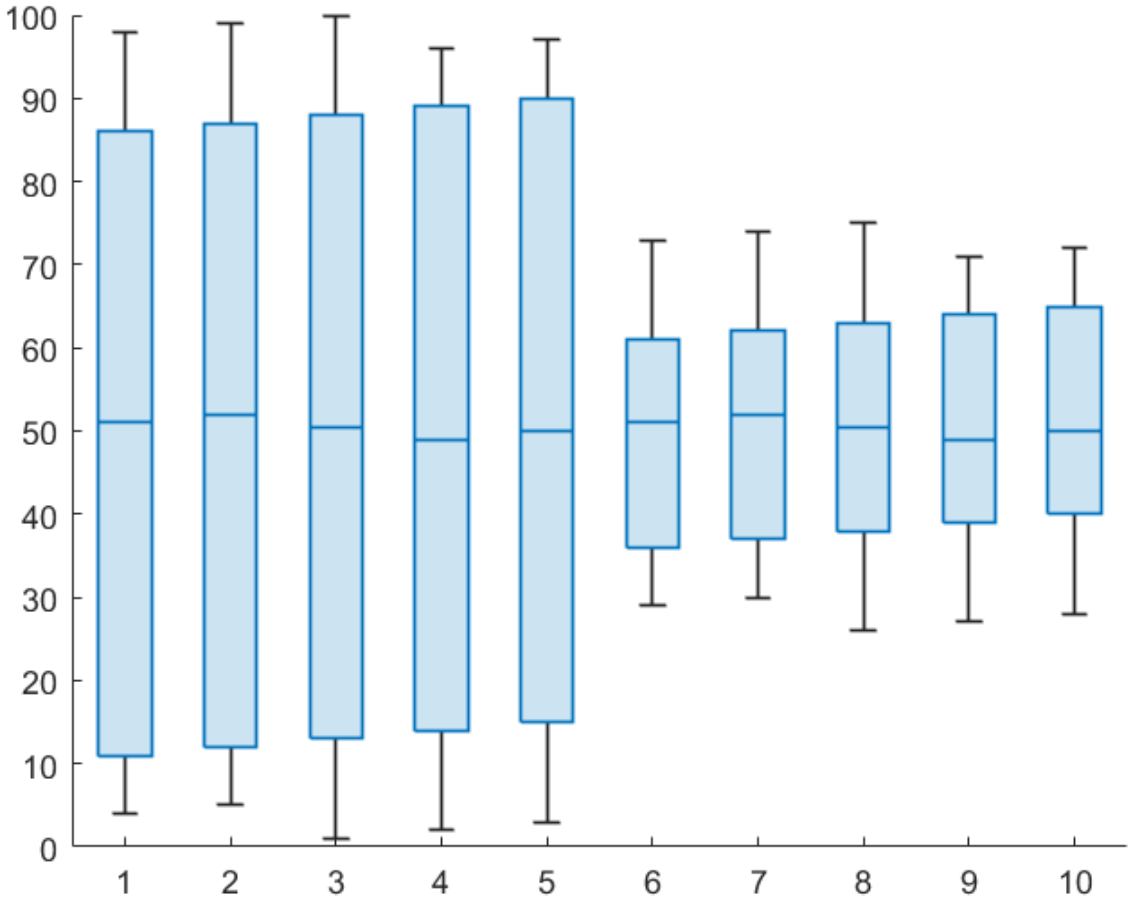
92	99	1	8	15	67	74	51	58	40
98	80	7	14	16	73	55	57	64	41
4	81	88	20	22	54	56	63	70	47
85	87	19	21	3	60	62	69	71	28
86	93	25	2	9	61	68	75	52	34
17	24	76	83	90	42	49	26	33	65
23	5	82	89	91	48	30	32	39	66
79	6	13	95	97	29	31	38	45	72
10	12	94	96	78	35	37	44	46	53
11	18	100	77	84	36	43	50	27	59

对于上述数据矩阵，画出各列的箱型图

5. 箱型图

```
x = [92  99  1  8  15  67  74  51  58  40
98  80  7  14  16  73  55  57  64  41
4  81  88  20  22  54  56  63  70  47
85  87  19  21  3  60  62  69  71  28
86  93  25  2  9  61  68  75  52  34
17  24  76  83  90  42  49  26  33  65
23  5  82  89  91  48  30  32  39  66
79  6  13  95  97  29  31  38  45  72
10  12  94  96  78  35  37  44  46  53
11  18  100  77  84  36  43  50  27  59];
h = boxchart(x)
```

5. 箱型图



5. 箱型图

Python实现，使用pyplot中的boxplot()绘制箱型图。

plt.boxplot(data, notch, position)

6.Python数据可视化环境准备

Matplotlib是python中最流行的2D会图库，它可以在各种平台上以各种硬拷贝格式交互式地生成具有出版品质的图形。对于高级用户，只需几行代码即可使用Matplotlib将数据生成条形图、饼图、散点图等常用图标。

在Matplotlib中使用最多的模块就是pyplot。Pyplot非常接近于MATLAB的绘图实现，而且大多数的命令与MATLAB极其类似。

6.Python数据可视化环境准备

当然，和MATLAB一样，它需要很多的数学运算，因此NumPy这个组件同样必不可少。

便于模块的使用，一般以以下两句开始：

```
import numpy as np
import matplotlib.pyplot as plt
```

这里导入了NumPy和matplotlib的pyplot两个模块，并分别使用np和plt作为二者的别名。