1.1

这类规划问题，为了方便计算，我们采取变量替换

$$u=\frac{x+|x|}{2},v=\frac{x-|x|}{2} \tag{1.1}$$

然后就转变为了常规的线性规划问题，此时有

$$\begin{pmatrix}1 & -1 & -1 & 1\\ 1 & -1 & 1 & -3\\ 1 & -1 & -2 & 3\end{pmatrix}(u+v)=\begin{pmatrix}0\\ 1\\ -\frac{1}{2}\end{pmatrix} \tag{1.2}$$

此时题目的问题变为，在(1.2)的约束条件下，求解

$$\min\begin{pmatrix}1 & 2 & 3 & 4\end{pmatrix}(u-v) \tag{1.3}$$

代码如下：

```
clc,clear

c=[1:4]';

b=[0,1,-1/2]';

a=[1,-1,-1,1;1,-1,1,-3;1,-1,-2,3];

prob=optimproblem;

u=optimvar('u',4,'LowerBound',0);

v=optimvar('v',4,'LowerBound',0);

prob.Objective=sum(c'*(u+v));

prob.Constraints=a*(u-v)==b;

[sol,fval.flag,out] =solve(prob)

x=sol.u-sol.v
```

得出的结果如下：

```
fval =

  包含以下字段的 struct:

    flag: 1.2500
```

```
x =

    0.2500
         0
         0
   -0.2500
```

1.2

这是题基于实际问题的线性规划我们可以这样假设：

不妨设 x_1，x_2 为分别用 A_1,A_2 加工产品 I 的件数，x_3，x_4，x_5 为分别用 B_1,B_2，B_3 加工产品 I 的件数，x_6，x_7 为分别用 A_1,A_2 加工产品 II 的件数，由题意，x_6+x_7 为用 B_1 加工产品 II 的件数，此时我们题目便转化为了求

$$\begin{gathered}\max(1.25-0.25)(x_1+x_2)+(2-0.35)(x_6+x_7)+(2.8-0.5)x_8\\ -\frac{300}{6000}(5x_1+10x_6)-\frac{321}{10000}(7x_2+9x_7+12x_8)\\ -\frac{250}{4000}(6x_3+8x_6+8x_7)-\frac{783}{7000}(x_4+11x_8)-\frac{200}{4000}\times 7x_5\end{gathered} \tag{1.4}$$

使得

$$\begin{cases}5x_1+10x_6\le 6000\\ 7x_2+9x_7+12x_8\le 10000\\ 6x_3+8x_6+8x_7\le 4000\\ 4x_4+11x_8\le 7000\\ 7x_5\le 4000\\ x_1+x_2=x_3+x_4+x_5\\ x_1,x_2,\cdots,x_8\ge 0\end{cases} \tag{1.5}$$

代码如下：

```
clc, clear
c=[1-5*300/6000,1-321*7/10000,-6*250/4000,-4*783/7000,-200*7/4000,1.65-0.5-8*250/4000,1.65-321*9/10000-8*250/4000,2.3-321*12/10000-11*783/7000]';
b=[6000,10000,4000,7000,4000]';
a=[5,0,0,0,0,10,0,0;
```

```
    0,7,0,0,0,0,9,12;
    0,0,6,0,0,8,8,0;
    0,0,0,4,0,0,0,11;
    0,0,0,0,7,0,0,0];
prob = optimproblem('ObjectiveSense','max')
x = optimvar('x',8,'LowerBound',0);
prob.Objective = sum(c.*x);
prob.Constraints.con1 = a*x<=b;
prob.Constraints.con2 =
x(1)+x(2)-x(3)-x(4)-x(5)==0;
[sol,fval,flag,out]=solve(prob)
xx=sol.x
```

结果如下：

```
xx =

   1.0e+03 *

    1.2000
    0.2300
         0
    0.8586
    0.5714
         0

fval =

   1.1466e+03
```

1.3

容易求得

$$t=\frac{4r-40g-2}{rg} \tag{1.6}$$

由此我们可以求得 g 关于 t 的函数以及 r 关于 t 的函数

$$t=\frac{3-20g}{g}(0\le g\le 0.15) \tag{1.7}$$

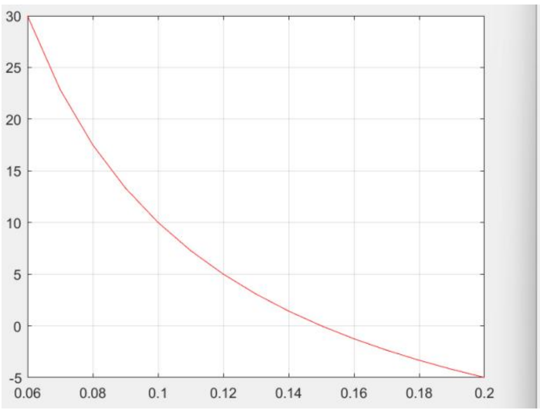
以及

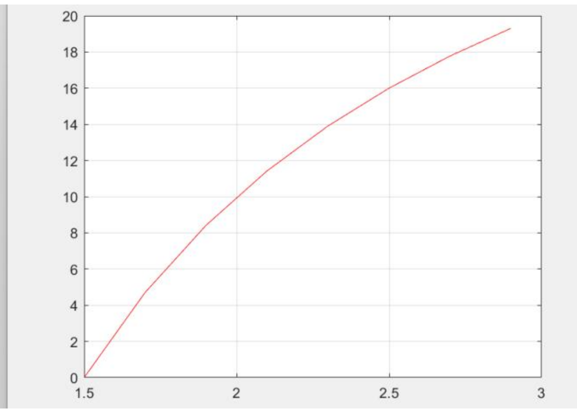
$$t=\frac{40r-60}{r}(r\ge 1.5) \tag{1.8}$$

此时我们可以画出图像，程序如下：

```
r=1.5:0.2:3;
g=0.06:0.01:0.2;
figure;
plot(r, (40*r-60)./r, 'r-');
grid on;
figure;
plot(g, (3-20*g)./g, 'r-');
grid on;
```

结果如下：





灵敏度分析：

我们如下定义 t 对 r 的灵敏度

$$S(t,r)=\frac{\Delta t}{\Delta r}=\frac{dt}{dr}\frac{r}{t} \tag{1.9}$$

此时我们带入(1.8)，即可得到 r=3 时，

$$S(t,r)=\frac{60}{40r-60}=\frac{1}{2} \tag{1.10}$$

即猪的体重 r 每天增加 1.5%，出售时间推迟 0.5%

同理，我们可以定义 t 对 s 的灵敏度

$$S(t,g)=\frac{\Delta t}{\Delta g}=\frac{dt}{dg}\frac{g}{t} \tag{1.11}$$

此时我们带入(1.9)，即可得到 g=0.1 时，

$$S(t,g)=-\frac{3}{3-20g}=-3 \tag{1.12}$$

即猪的价格 r 每天增加 1%，出售时间提前 3%

问题分析

该问题是部门在面对投资时经常遇到的问题，考虑到在限制时间内用 10 万元的投资得到最大的回报。

符号说明

x_{ij}:第 i 年(i=1,2,3,4,5)对分别对 A,B,C,D(j=1,2,3,4)四个项目的投资额

模型假设

假设部门每年将钱全部花出去，不留任何的钱

模型建立

在第一年，我们有如下投资

$$x_{11}+x_{14}=10 \tag{1.13}$$

在第二年的年初，我们有

$$x_{21}+x_{23}+x_{24}=1.06x_{14} \tag{1.14}$$

在第三年的年初，我们有

$$x_{31}+x_{32}+x_{34}=1.15x_{11}+1.06x_{24} \tag{1.15}$$

在第四年的年初，我们有

$$x_{41}+x_{44}=1.15x_{21}+1.06x_{34} \tag{1.16}$$

在第五年的年初，我们有

$$x_{54}=1.15x_{31}+1.06x_{44} \tag{1.17}$$

此时，我们的目标便转化为求解

$$\max\ y=1.15x_{41}+1.40x_{23}+1.25x_{32}+1.06x_{54} \tag{1.18}$$

于是乎，数学模型如下

$$\max\ y=1.15x_{41}+1.40x_{23}+1.25x_{32}+1.06x_{54}$$

$$s.t.\begin{cases} x_{11}+x_{14}=10 \\ x_{21}+x_{23}+x_{24}=1.06x_{14} \\ x_{31}+x_{32}+x_{34}=1.15x_{11}+1.06x_{24} \\ x_{41}+x_{44}=1.15x_{21}+1.06x_{34} \\ x_{54}=1.15x_{31}+1.06x_{44} \\ x_{32}\le 4, x_{23}\le 3 \\ x_{ij}\ge 0 \end{cases}$$

由于求解器的限制，我们将新元素重新排列成一个列向量

$$\begin{pmatrix} x_{11} \\ x_{14} \\ x_{21} \\ x_{23} \\ x_{24} \\ x_{31} \\ x_{32} \\ x_{34} \\ x_{41} \\ x_{44} \\ x_{54} \end{pmatrix} \tag{1.19}$$

代码如下：

```
clc,clear
prob=optimproblem('ObjectiveSense','max');
x = optimvar('x',11,'LowerBound',0);
prob.Objective =
1.15*x(9)+1.40*x(4)+1.25*x(7)+1.06*x(11);
prob.Constraints.con1 = x(1)+x(2)==10;
prob.Constraints.con2
=x(3)+x(4)+x(5)-1.06*x(2)==0;
prob.Constraints.con3 =
x(6)+x(7)+x(8)-1.15*x(1)-1.06*x(5)==0;
prob.Constraints.con4 =
x(9)+x(10)-1.15*x(3)-1.06*x(8)==0;
prob.Constraints.con5
=1.15*x(6)+1.06*x(10)-x(11)==0;
prob.Constraints.con6 =x(7)<=4;
prob.Constraints.con7=x(4)<= 3;
```

```
[sol,fval,flag,out]=solve(prob),sol.x;

xx=sol.x
```

结果如下：

```
fval =

   14.3750
```

3.4783
6.5217
3.9130
3
0
0
4
0
4.5000
0
0

此时

$$\begin{pmatrix} x_{11}=3.4783 \\ x_{14}=6.5217 \\ x_{21}=3.9130 \\ x_{23}=3 \\ x_{24}=0 \\ x_{31}=0 \\ x_{32}=4 \\ x_{34}=0 \\ x_{41}=4.5000 \\ x_{44}=0 \\ x_{54}=0 \end{pmatrix} \tag{1.20}$$

最大收益为 **14.3750** 万元

2.2

问题分析

该问题是典型的非线性转线性问题，我们需要将非线性转为线性

考虑到

$$y=x_1x_2 \tag{1.21}$$

于是题目的条件即可转化为

$$
\max\ x_1+y-x_3
$$

$$
s.t.\begin{cases}-2x_1+3x_2+x_3\leq 3\\ x_1+x_2-1\leq y\leq x_1\\ x_1+x_2-1\leq y\leq x_2\\ x_1,x_2,x_3,y=0\text{或}1\end{cases} \tag{1.22}
$$

2.2

问题分析

这是常见的 0-1 决策问题，需要我们在最小建校时能覆盖最大的区域

符号说明

x_i:对第 i(i=1,2,3,4,5,6)个区域的选取

模型建立

对小区 A_1 我们有

$$
x_1+x_2+x_3\geq 1 \tag{1.23}
$$

对小区 A_2，我们有

$$
x_2+x_4\geq 1 \tag{1.24}
$$

对小区 A_3，我们有

$$
x_3+x_5\geq 1 \tag{1.25}
$$

对小区 A_4，我们有

$$
x_4+x_6\geq 1 \tag{1.26}
$$

对小区 A_5，我们有

$$
x_2+x_5\geq 1 \tag{1.27}
$$

对小区 A_6，我们有

$$
x_5+x_6\geq 1 \tag{1.28}
$$

对小区 A_7，我们有

$$
x_1\geq 1
$$

对小区 A_8，我们有

$$
x_2+x_4+x_6\geq 1 \tag{1.29}
$$

于是，我们只要求解

$$
\min\sum_{i=1}^{6}x_i \tag{1.30}
$$

代码如下：

```
clc,clear
prob=optimproblem;
x=optimvar('x',6,'Type','integer','LowerBound',
0,'UpperBound',1);
prob.Objective = sum(x);
```

```
prob.Constraints.con1=x(1)+x(2)+x(3)>=1;
prob.Constraints.con2=x(2)+x(4)>=1;
prob.Constraints.con3=x(3)+x(5)>=1;
prob.Constraints.con4=x(4)+x(6)>=1;
prob.Constraints.con5=x(6)+x(5)>=1;
prob.Constraints.con6=x(1)>=1;
prob.Constraints.con7=x(4)+x(2)+x(6)>=1;
prob.Constraints.con8=x(2)+x(5)>=1;
[sol,fval,flag]=solve(prob),xx=sol.x
```

结果如下：

```
xx =

     1
     0
     0
     1
     1
     0
```

即此时只要对 B_1, B_4, B_5,建设即可

2.3

问题分析

这是常见的 0-1 决策问题，需要我们在限定设备时做出能提供最大收益的决策

符号说明

x_{ij}:对第 i(i=1,2,3,4)个企业对第 j(j=1,2,3,4)个工厂的选取

c_{ij}： x_{ij}:所对因的盈利

模型建立

每个企业至少有一个设备，我们有

$$\sum_{i=1}^{6} x_{ij} \geq 1 \tag{1.31}$$

每个设备都有一台，我们有

$$\sum_{j=1}^{4} x_{ij} = 1 \tag{1.32}$$

于是问题转化为

$$\text{求}\max\sum c_{ij}x_{ij}$$

$$s.t.\begin{cases}\sum_{i=1}^{6}x_{ij}\geq 1\\ \sum_{j=1}^{4}x_{ij}=1\\ x_{ij}=0\text{或}1\end{cases} \tag{1.33}$$

代码如下：

```
clc,clear
prob=optimproblem('ObjectiveSense','max');
b=[4,2,3,4;6,4,5,5;7,6,7,6;7,8,8,6;7,9,8,6;7,10,8,6]
x=optimvar('x',6,4,'Type','integer','LowerBound',0,'UpperBound',1);
prob.Objective = sum(b.*x,'all');
prob.Constraints.con1=sum(x,2)==1;
prob.Constraints.con2=sum(x,1)>=1;
[sol,fval,flag]=solve(prob),xx=sol.x
```

结果如下：

```
          xx =

               0     0     0     1
               1     0     0     0
               0     0     1     0
fval =         0     1     0     0
               0     1     0     0
    44         0     1     0     0
```

即最大收益为 44 千万

1.7

问题分析

这是常见的规划问题，让我们求解变量的最大值

模型建立

由题，我们有

$$\max\ v$$

$$s.t.\begin{cases}\sum_{i=1}^{100} a_{ij}x_i \geq v \\ \sum_{i=1}^{100} x_i = 1 \\ x_i \geq 0\end{cases} \tag{1.34}$$

代码如下：

```
clc,clear

prob=optimproblem('ObjectiveSense','max');

a=randi([0,10],100,150);

v=optimvar('v',1,'LowerBound',0)

x=optimvar('x',100,150,'LowerBound',0);

prob.Objective =v;

prob.Constraints.con1=sum(a.*x,1)>=v;

prob.Constraints.con2=sum(x,1)==1;

[sol,fval,flag]=solve(prob),xx=sol.x;
```

解得

```
fval =

    10
```

即 v 的最大值为 10

2.7

问题分析

这是常见的规划问题，需要我们求出获利最大时的选取

符号说明

x :对家电 I 的选取数
y：对家电 II 的选取数

模型建立

对于设备 A，我们有

$$5y \le 15 \tag{1.35}$$

对于设备 B 我们有

$$6x + 2y \le 24 \tag{1.36}$$

对于调试工序，我们有

$$x + y \le 5 \tag{1.37}$$

于是问题可转化为

$$\max\ 2x + y$$
$$s.t \begin{cases} 5y \le 15 \\ 6x + 2y \le 24 \\ x + y \le 5 \end{cases} \tag{1.38}$$

代码如下：

```
clc,clear
prob=optimproblem('ObjectiveSense','max');
v=optimvar('v',1,'LowerBound',0);
x=optimvar('x',1,'LowerBound',0);
prob.Objective =2*x+v;
prob.Constraints.con1=5*v<=15;
prob.Constraints.con2=6*x+2*v<=24;
prob.Constraints.con3=x+v<=5;
[sol,fval,flag]=solve(prob)
xx=sol.x,yy=sol.v
```

结果如下：

```
fval =

    8.5000
```

```
xx =

    3.5000


yy =

    1.5000
```

即对Ⅰ选 3.5 台，对Ⅱ选 1.5 台时，我们有最大值 8.5

3.2

问题分析

这是一题典型的同余方程组问题，我们只需求出最小解即可

符号假设

x：鸡蛋的数量

模型建立

由题，可构建如下方程

$$\min x$$

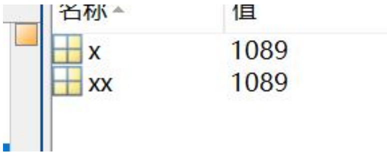
$$s.t\begin{cases} x mod(2)=1 \\ x mod(3)=0 \\ x mod(4)=1 \\ x mod(5)=4 \\ x mod(6)=3 \\ x mod(7)=4 \\ x mod(8)=1 \\ x mod(9)=0 \end{cases} \tag{1.39}$$

代码如下

```
clc,clear
x = 1;
while true
    if rem(x, 2) == 1
&&rem(x,4)==1&&rem(x,3)==0&& rem(x, 9) == 0&&
rem(x,5)==4&&rem(x,6)==3&&rem(x,7)==4&&rem(x,8)
==1;
        break;
```

```
    end
    x = x + 1;
end
xx = x;
```

结果如下



即最小值为 1089

3.7

问题分析

这是一道组合投资问题，需要我们求出在给定投资数下时的投资收益

符号说明

x_i：购买股票 i 的数量

$\sigma_i(i=1,2,3)$：A,B,C 相关收益的标准差

$\rho_{ij}(i=1,2,3,j=1,2,3)$：i 和 j 的相关系数

模型建立

由题目所给信息，我们首先能求出收益的协方差矩阵

$$R=\begin{pmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & \rho_{13}\sigma_1\sigma_3 \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 & \rho_{23}\sigma_2\sigma_3 \\ \rho_{13}\sigma_1\sigma_3 & \rho_{23}\sigma_2\sigma_3 & \sigma_2^2 \end{pmatrix}=\begin{pmatrix} 4 & 2.5 & -10 \\ 2.5 & 36 & -15 \\ -10 & -15 & 100 \end{pmatrix} \tag{1.40}$$

此时风险 **X** 就能表示为

$$X=x^T Rx \tag{1.41}$$

此时我们的收益 **Z** 可表示为

$$Z=5x_1+8x_2+10x_3 \tag{1.42}$$

则问题就转化为

$$\min x^T Rx$$
$$s.t\begin{cases} 5x_1+8x_2+10x_3\geq 100000 \\ 20x_1+25x_2+30x_3\leq 500000 \\ x_1,x_2,x_3\geq 0 \end{cases} \tag{1.43}$$

代码如下：

```
clc,clear
```

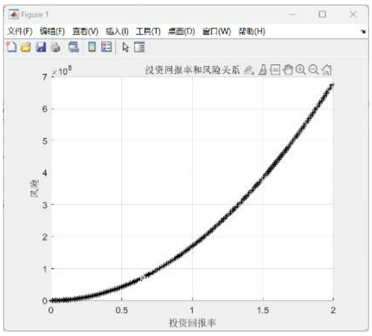
```
prob=optimproblem;
R=[4,2.5,-10;2.5,36,-15;-10,-15,100]
x=optimvar('x',3,'LowerBound',0);
prob.Objective =x'*R*x;
x0.x=rand(3,1);
for n=0:100
prob.Constraints.con1=5*x(1)+8*x(2)+10*x(3)>=0.01*n*500000;
prob.Constraints.con2=20*x(1)+25*x(2)+30*x(3)<=500000;
[sol,fval,flag,out]=solve(prob,x0)
y=fval;
plot(0.01*n,y);
end
```

结果如下

```
  1.0e+04 *

    1.3111
    0.1529
    0.2221
```

第一题：

A 投 13111，B 投 1529，C 投 2221

第二题：

从图像上显示，投资回报率越高，风险越高

3.9

问题如下

$$\max z = c^{\mathrm{T}}x + \frac{1}{2}x^{\mathrm{T}}Qx,$$

$$\text{s. t. } \begin{cases} -1 \leqslant x_1x_2 + x_3x_4 \leqslant 1, \\ -3 \leqslant x_1 + x_2 + x_3 + x_4 \leqslant 2, \\ x_1, x_2, x_3, x_4 \in \{-1, 1\}. \end{cases}$$

代码如下

```
clc,clear
prob=optimproblem('ObjectiveSense','max');
x=optimvar('x',4,'LowerBound',-1,'UpperBound',
1);
c=[6,8,4,2];
Q=[-1,2,0,0;2,-1,2,0;0,2,-1,2;0,0,2,-1];
prob.Objective = c*x+0.5*x'*Q*x;
prob.Constraints.con1=x(1)*x(2)+x(3)*x(4)>=-1;
prob.Constraints.con2=x(1)*x(2)+x(3)*x(4)<=1;
prob.Constraints.con3=x(1)+x(2)+x(3)+x(4)>=-3;
prob.Constraints.con3=x(1)+x(2)+x(3)+x(4)<=2;
```

```
x0.x=rand(4,1);
[sol,fval,flag,out]=solve(prob,x0),xx=sol.x;
```

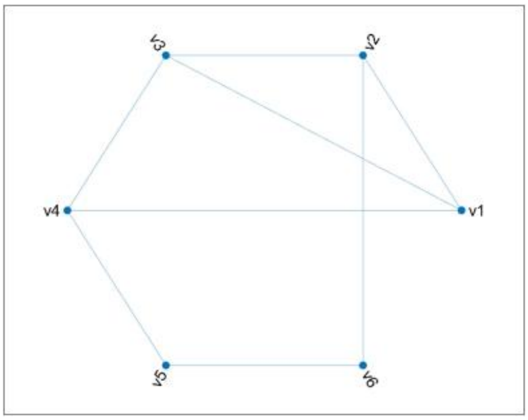
结果如下：

```
fval =

  16.3333
```

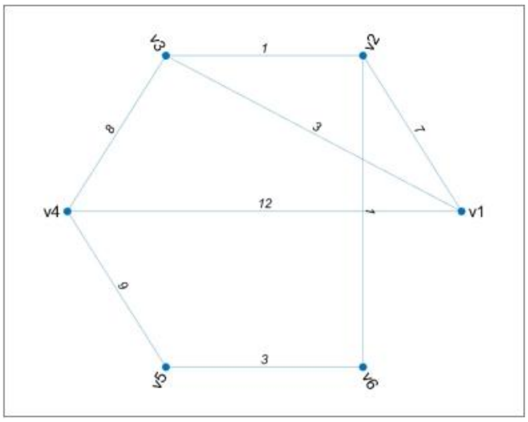
4.1

代码如下：

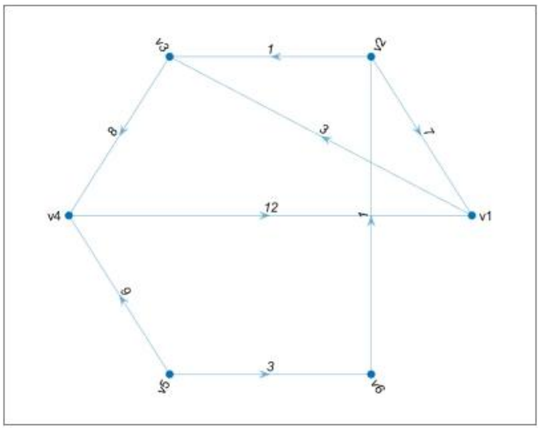
```
clc,clear,close all
a1=zeros(6);
a1(1,[2:4])=1;a1(2,[3,6])=1;a1(3,4)=1;a1(4,5)=1;a1(5,6)=1;
s=cellstr(strcat('v',int2str([1:6]')));
G1=graph(a1,s,'upper');
plot(G1,'Layout','circle')
```




```
a2=zeros(6);
a2(1,[2:4])=[7,3,12];a2(2,[3,6])=[1,1];
a2(3,4)=8;a2(4,5)=9;a2(5,6)=3;
s=cellstr(strcat('v',int2str([1:6]')));
G2=graph(a2,s,'upper');
plot(G2,'Layout','circle','EdgeLabel',G2.Edges.Weight)
```




```
a3=zeros(6);
a3(1,3)=3;a3(2,[1,3])=[7,1];a3(3,4)=8;
a3(4,1)=12;a3(5,[4,6])=[9,3];a3(6,2)=1;
s=cellstr(strcat('v',int2str([1:6]')));
G3=digraph(a3,s);
plot(G3,'EdgeLabel',G3.Edges.Weight,'Layout','circle')
```



4.4

问题分析：

这是一个最短路径的问题，可以使用 Dijkstra 标号算法求解。

符号说明：

分别用 p,d 表示最短路径和最短距离。

模型建立：

（1）首先从 v1 出发，v1 到 v1 的最短距离为 0，标记节点 1

（2）从 v1 出发，到 v2 的距离为 20，到 v5 的距离为 15，节点 2 和节点 5 更新，其前面点均为节点 1，标记节点 5

（3）从 v1 出发，经过 v5，到 v2 的距离为 40，到 v3 的距离为 33，到 v4 的距离为 50，到 v6 的距离为 30，节点 3、节点 4 和节点 6 更新，其前面点均为节点 5，标记节点 2

（4）从 v1 出发，经过 v2，到 v3 的距离为 40，到 v4 的距离为 80，到 v5 的距离为 45，不更新任何节点，标记节点 6

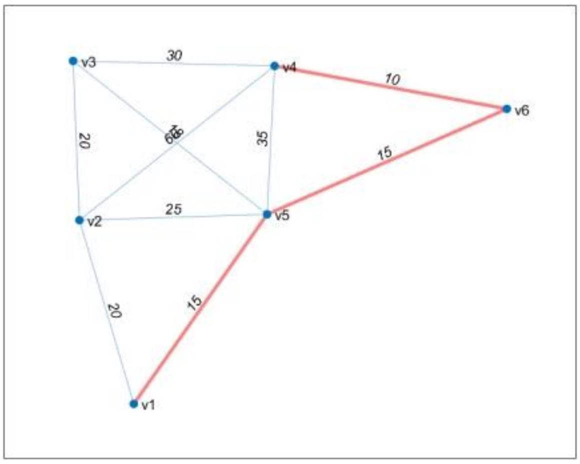
（5）从 v1 出发，经过 v6，到 v4 的距离为 40，节点 4 更新，其前面点为节点 6，标记节点 3

（6）从 v1 出发，经过 v3, 到 v4 的距离为 63，不更新任何节点，标记节点 4，结束。

求得从 v1 到 v4 的最短路径为 v1→v5→v6→v4，最短距离为 40。

代码：

```
clc,clear,close all
a=zeros(6);
a(1,[2,5])=[20,15];a(2,[3:5])=[20,60,25];
a(3,[4,5])=[30,18];a(4,[5,6])=[35,10];a(5,6)=15;
s=cellstr(strcat('v',int2str([1:6]')));
G=graph(a,s,'upper');
[p,d]=shortestpath(G,1,4)
h=plot(G,'EdgeLabel',G.Edges.Weight);
highlight(h,p,'EdgeColor','r','LineWidth',2)
disp('(d)赋权无向图');
```



运行结果：

```
p = 1×4
     1     5     6     4
d = 40
```

4.2

这题是一题典型的求最短路径问题，我们可以用到 Dijkstra 算法来求解

代码如下：

```
clc, clear;
L= {'A','B1',2;'A','B2',4 ;'B1','C1',3
'B1','C2',3;'B1','C3',1;'B2','C1',2
'B2','C2',3;'B2','C3',1;'C1','D',1
'C2','D',3 ;'C3','D',4};
G=digraph (L(:,1),L(:,2),cell2mat (L(:,3)));
plot (G), [p,d]=shortestpath(G,'A','D')
```

结果如下：

```
p =

  1×4 cell 数组

    {'A'}    {'B1'}    {'C1'}    {'D'}


d =

     6
```

4.7

这题也是一题典型的求最短路径问题，同样的，我们可以用到 Dijkstra 算法来求解

符号说明

s_j：到 j 村所需要的距离

c_j：j 村有的小学生人数

模型建立

我们首先可以得到对应的邻接矩阵

$$\boldsymbol{W}=\begin{bmatrix} 0 & 2 & 7 & \infty & \infty & \infty \\ 2 & 0 & 4 & 6 & 8 & \infty \\ 7 & 4 & 0 & 1 & 3 & \infty \\ \infty & 6 & 1 & 0 & 1 & 6 \\ \infty & 8 & 3 & 1 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix} \tag{1.1}$$

然后我们可以调用 Dijkstra 算法来求出所有顶点对的距离，最后问题便转化为求

$$\min\ s_j \tag{1.2}$$

代码如下：

```
clc,clear,w=zeros(6);
w(1,[2,3])=[2,7];w(2,[3:5])=[4,6,8];
w(3,[4,5])=[1,3];w(4,[5,6])=[1,6];
w(5,6)=3;
G=graph(w,'Upper');
d=distances(G)
md=max(d)
c=[50 40 60 20 70 90];
s=c*d
[ms,ind]=min(s)
```

结果如下：

结 果

```
d =

     0     2     6     7     8    11
     2     0     4     5     6     9
     6     4     0     1     2     5
     7     5     1     0     1     4
     8     6     2     1     0     3
    11     9     5     4     3     0


md =

    11     9     6     7     8    11


s =

        2130        1670        1070        1040        1050        1500


ms =

        1040


ind =
```

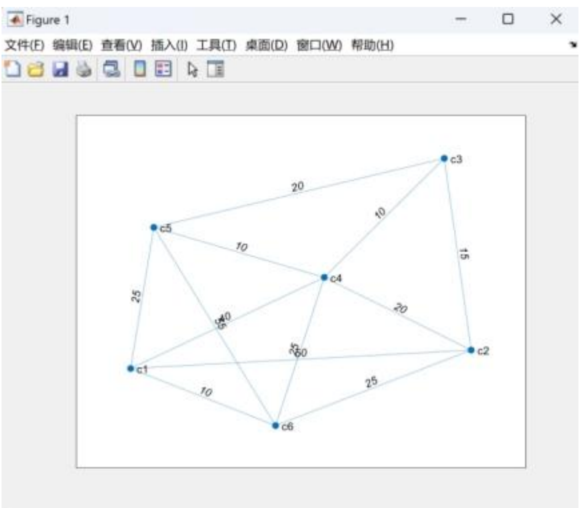
结果表明，选村庄四最好

补充题：

1. 问题分析

某公司在六个城市 $c_1, c_2, \dots, c_6$ 中有分公司,从 c_i 到 c_j,的直接航程票价记在下述矩阵的(i,j)位置上（∞表示无直接航路)。画出该矩阵对应的赋权图（顶点和边都要有标注），并帮助该公司设计一个简便的算法，能快速得到一张城市 c_1 到其它城市间的票价最便宜的路线图。

2. 赋权图



3. 从求解器确定答案：

从 c1 到其他城市的最短路径以及最短距离如下所示

```
p =

     1     6     2

d =

    35
```

1-2

```
p =

     1     5     4     3

d =

    45
```

1-3

```
p =

     1     5     4

d =

    35
```

1-4

```
p =

     1     5

d =

    25
```

1-5

```
p =

     1     6

d =

    10
```

1-6

4.从 Floyd 算法求解

```
price =

    35    45    35    25    10


path =

     6     5     5     1     1

fx >>
```

有图可知:

c_1-c_2:1-6-2 ,35

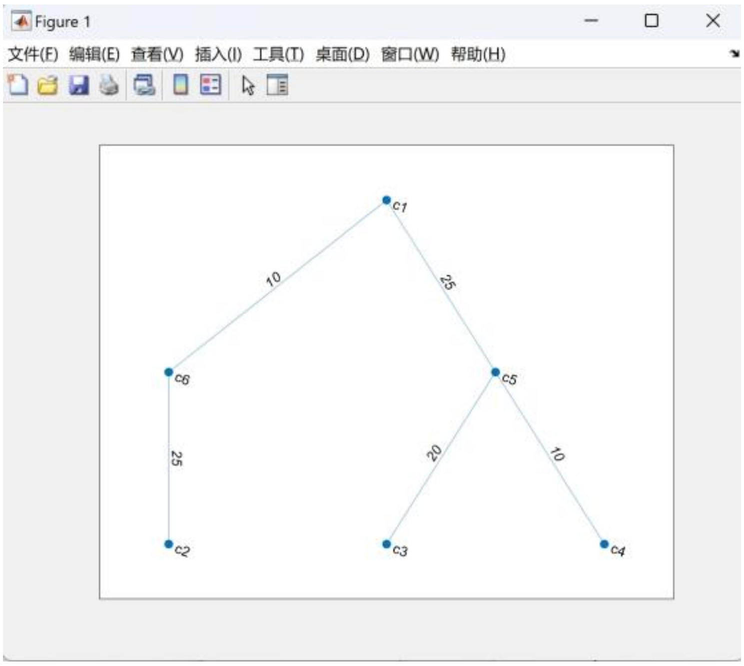
c_1-c_3:1-5-3 ,45

c_1-c_4:1-5-4 ,35

c_1-c_5:1-5　,25

c_1-c_6:1-6　,10

票价最便宜路线图



代码如下：

```
clc,clear,w = zeros(6);
w(1,[2,4,5,6]) = [50,40,25,10];w(2,[1,3,4,6]) = [50,15,20,25];
w(3,[2,4,5]) = [15,10,20];w(4,[1,2,3,5,6]) = [40,20,10,10,25];
w(5,[1,3,4,6]) = [25,20,10,55];w(6,[1,2,4,5]) = [10,25,25,55];
%构造完整的邻接矩阵
s = cellstr(strcat('c',int2str([1:6]')));%顶点字符串
G = graph(w,s,'upper');%利用邻接矩阵的上三角元素构造无向图
plot(G,'EdgeLabel',G.Edges.Weight);%画无向图
[price,path] = Floyd(w) %显示 c1 至其他地方票价最便宜价钱，路线矩阵
p = zeros(6);
p(1,5:6) = [25,10];p(5,3:4) = [20,10];p(6,2) = 25;
p = p+p';
G1 = graph(p,s,'upper');%利用邻接矩阵的上三角元素构造无向图
plot(G1,'EdgeLabel',G1.Edges.Weight);%画最短航线图

function [price,path] = Floyd(w)
%输出矩阵为两两顶点间最短距离矩阵，输入矩阵为待求邻接矩阵
n = length(w);
w(w == 0) = inf; %把零元素换成无穷大
w(1:n+1:end) = 0; %把对角线元素换成 0
path = -ones(n);%初始化 path 矩阵
for i=1:n
for j=1:n
if i~=j && w(i,j)~=inf
path(i,j)=i;
end
end
```

```
end %完成 path 矩阵未更新的建立
for k = 1:n
for i = 1:n
for j = 1:n
if w(i,k)+w(k,j) < w(i,j) %Floyd 算法核心，更新
w(i,j) = w(i,k)+w(k,j);
path(i,j) = k; %对 w 以及 path 矩阵更新
end
end
end
end
price = w(1,2:n);
path = path(1,2:n);%只考虑 c1 至其他城市，故只取部分
```

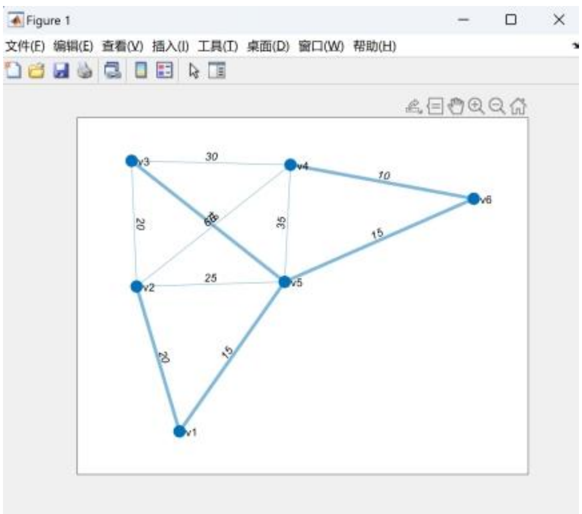
4.3

这个就只是画出赋权图，我们直接调用 prim 算法

代码如下：

```
clc,clear
a = zeros(6);
a(1,[2,5]) = [20,15];a(2,[3,4,5]) = [20,60,25];
a(3,[4,5]) = [30,18];a(4,[5,6]) = [35,10];
a(5,6) = 15; a = a+a'; %建立邻接矩阵
s = cellstr(strcat('v',int2str([1:6]')));
G = graph(a,s,'upper');%画出无向赋权图
p = plot(G,'EdgeLabel',G.Edges.Weight)
T = minspantree(G)%画出最小生成树
L = sum(T.Edges.Weight)%找出最小生成树的路径，并计算总和
highlight(p,T)%着重标记最小生成树
```

结果如下：



```
L =

    78
```

我们可知**最小生成树的长度为 78**

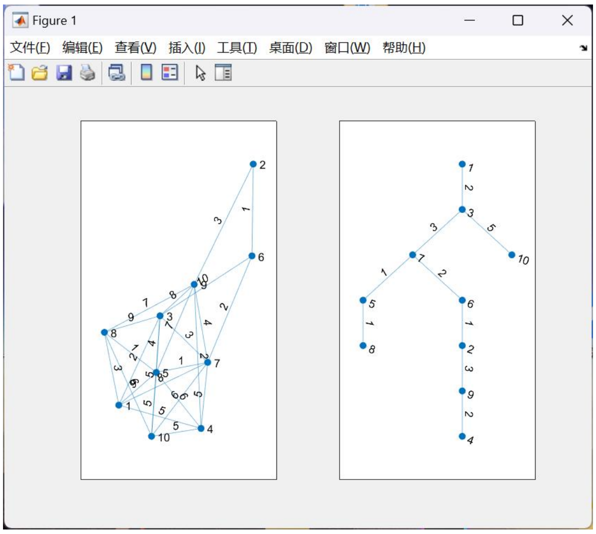
4.8

由于我们无法做到如题所示的 0.6 概率的随机，我们可以调用随机函数矩阵来表示随机的概念，

代码如下：

```
clc,clear
a = rand(10);%构造概率矩阵
a = triu(a,1);%我们取上三角元素
w = randi(10,10);%构造了权重矩阵
W = (a>=0.4).*w%生成无向赋权图邻接矩阵的上三角部分
W = W +W'%生成完全邻接矩阵
G = graph(W,'upper')
subplot(121),plot(G,'EdgeLabel',G.Edges.Weight)
T = minspantree(G)%使用 Prim 算法求得最小生成树
subplot(122),plot(T,'EdgeLabel',T.Edges.Weight)
[p,d] = shortestpath(G,1,10)%q 求得 1-10 的最短距离及最短路径;
d2 = distances(G)
```

结果如下：



```
p =

     1     3    10


d =

     7


d2 =

     0     8     2     5     4     7     5     3     7     7
     8     0     6     5     4     1     3     5     3     9
     2     6     0     7     4     5     3     5     7     5
     5     5     7     0     6     6     5     7     2     5
     4     4     4     6     0     3     1     1     5     5
     7     1     5     6     3     0     2     4     4     8
     5     3     3     5     1     2     0     2     4     6
     3     5     5     7     1     4     2     0     6     6
     7     3     7     2     5     4     4     6     0     7
     7     9     5     5     5     8     6     6     7     0
```

（1） 最小生成树如上图所示

（2） 路径为 1→3→10，最短路径长度为 7

（3） 每个点的最短距离如上

4.13

式中$N=|V|$表示点集Z中点的个数。

将该模型的第三个消除子回路的约束单独提出来：

$$\sum_{i,j\in S} x_{ij} \leqslant |S|-1, 2 \leqslant |S| \leqslant N-1, S \subset V$$

式中S为V的一个真子集，这个式子的含义是：对于一个V中的任意真子集S，S中连通的节点边数小于节点个数。

该问题可以转化为 0—1 整数规划类问题，具体问题可以转化为如下

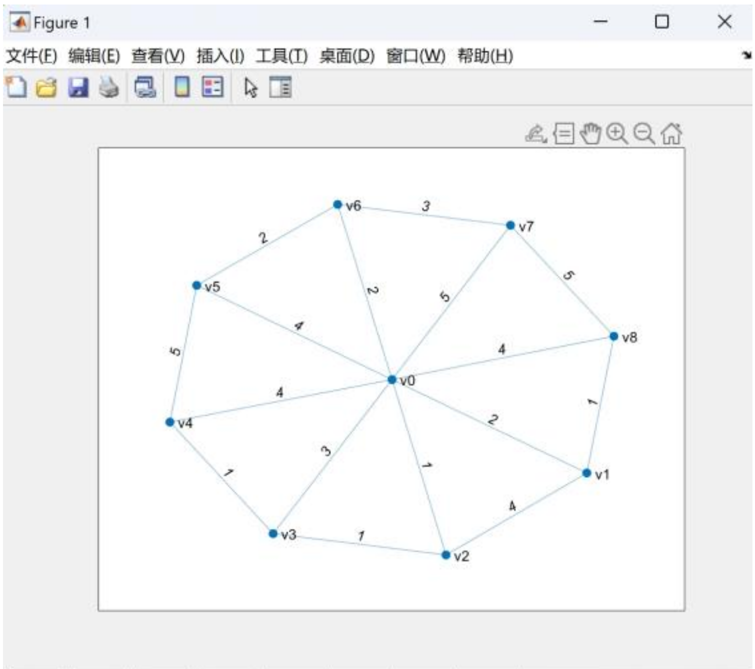
$$\text{求}\min\ z\sum\sum \omega_{ij}x_{ij}$$

$$s.t\begin{cases}\sum_{j=1}^{n} x_{ij} \geq 1\\ \sum_{i=1}^{n} x_{ij} = 1\\ \sum_{i,j\in S} x_{ij} \leq |S|-1, 2 \leq |S| \leq n-1, S \subset V\end{cases} \tag{1.1}$$

代码如下：

```
clc, clear, close all, n = 9;
nod =cellstr(strcat('v',int2str([0:n-1]')));%运用 cellstr 进行标号
G = graph(); G = addnode(G,nod); %定好无序图
ed ={ 'v0','v1',2;'v0','v2',1;'v0','v3',3;'v0','v4',4
'v0','v5',4;'v0','v6',2;'v0','v7',5;'v0','v8',4
'v1','v2',4;'v1','v8',1;'v2','v3',1;'v3','v4',1
'v4','v5',5;'v5','v6',2;'v6','v7',3;'v7','v8',5};
G = addedge(G,ed(:,1),ed(:,2),cell2mat(ed(:,3)));%无序图确认
```

```
p = plot(G,'EdgeLabel',G.Edges.Weight) %做出 9 个村庄道路及道路长度图
w = full(adjacency(G,'weighted')); %做出边权矩阵
w(w==0) = 1000000; %充分大的正实数，让所有不能直接到达的两个村庄改为足够大
prob = optimproblem;
x = optimvar('x',n,n,'Type','integer','LowerBound',0,'UpperBound',1);
prob.Objective = sum(sum(w.*x));
prob.Constraints.con1 = 1<=sum(x(1,:));%条件 1
prob.Constraints.con2 = sum(x(:,2:end))'==1; %条件 2
con3 = [];
for q = 2:n-1
a = zeros(q);
for m = 1:100 %100 次足够精度
b = randperm(n);%随机对 n 数进行排序
c = b(1:q); %相当于从 n 中随机抽取 p 个数
a = x(c,c);
con3 = [sum(sum(a)) <= q-1;con3];
end
end
prob.Constraints.con3 = con3;%条件 3
[sol,fval,flag,out] = solve(prob)
xx = sol.x
[i,j]=find(sol.x);
ind = [(i-1)'; (j-1)'] %输出树的顶点编号
```
结果如下：



```
ind =

     0     0     2     3     6     0     6     1
     1     2     3     4     5     6     7     8

>>
```

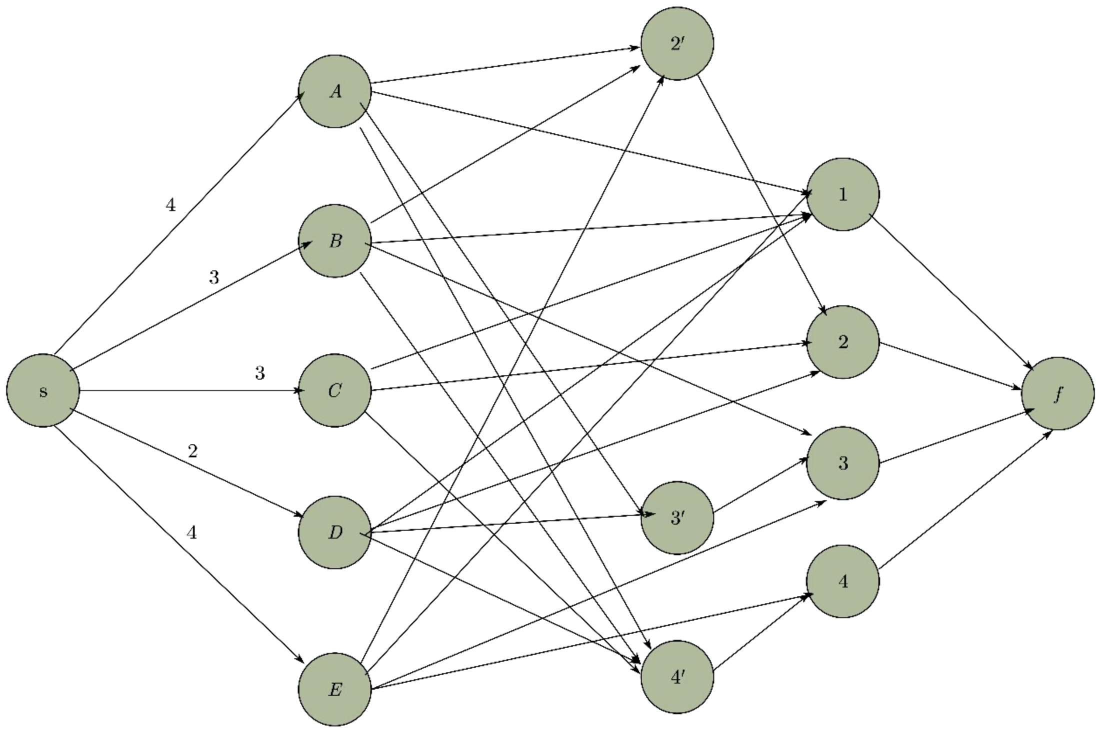
```
fval =

    13
```

即最小生成树的坐标和权重都可得到

4.10

这是个最大流问题。分别有五个专业作为多源，四个公司作为多汇。为此我们需要虚拟假设一个源点和一个汇点。将上述问题转为单对单模型。

由题意可知，转化原问题，可得如下模型：

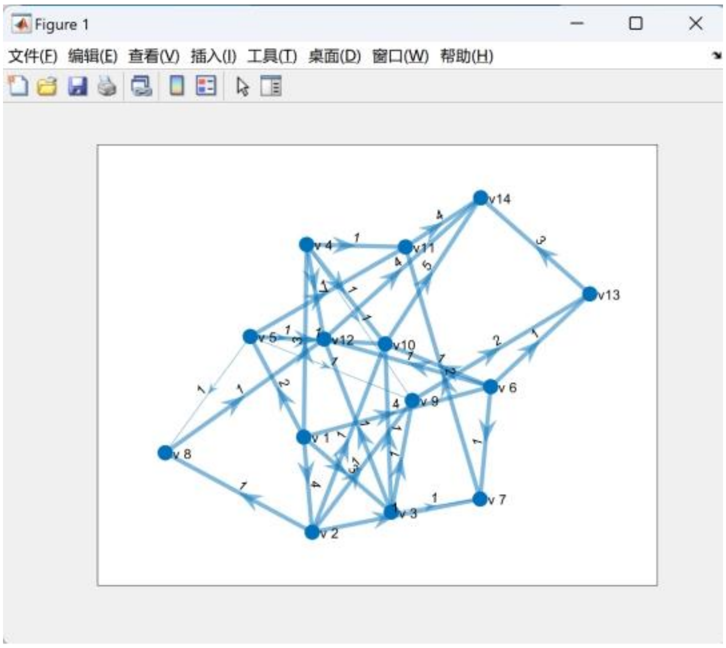


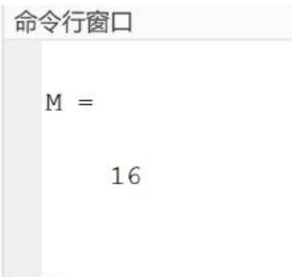
代码如下：

```
clc,clear
a = zeros(14);%总共有 5 个专业，4 个公司，1 个源点，1 个汇点，3 个中转点
a(1,[2:6]) = [4,3,3,2,4];a(2,[7:10]) = 1;
a(3,[7,9,10,12]) = 1;a(4,[9:12]) = 1;
a(5,[8:12]) = 1;a(6,[7,10,12,13]) = 1;
a(7,11) = 2;a(8,12) = 1;a(9,13) = 2;
a([10:13],14) = [5,4,4,3];%将权赋好
```

```
s = cellstr(strcat('v',int2str([1:14]')));%命名序号
G = digraph(a,s);%确定赋权图
[M,F] = maxflow(G,1,14) %使用默认 searchtrees 方法求最大流
p = plot(G,'Layout','force','EdgeLabel',G.Edges.Weight);
highlight(p,F)%显示最大流并画出最大流
```

结果如下：





所以最大流量是 16，正好与五个专业一共十六名毕业生想对应，所以所有公司都能招聘到各自需要的专业人才。

5.1

代码如下

```
clear,clc
x=linspace(0,10,1000);
gx=@(x)(3*x.^2+4*x+6).*sin(x)./(x.^2+8*x+6);
y=gx(x);%计算函数值
pp=csape(x,y)%求三次调样插值
gh=@(x)fnval(pp,x);
fplot(gh,[0,10]);
I1=integral(gx,0,10),
I2=integral(gh,0,10)
```

结果如下

I1 =

    2.2430

I2 =

    2.2430

5.3

代码如下

```
clear,clc
t=[700,720,740,760,780];
V=[0.0977,0.1218,0.1406,0.1551,0.1664];
v1=interp1(t,V,[750,770])%计算线性插值
pp=csape(t,V);
v2=fnval(pp,[750,770])%计算三次调样插值
plot(t,V,'* -')
hold on,fplot(@(t)fnval(pp,t),[700,780])
```

结果如下

v1 =

    0.1478   0.1608

v2 =

    0.1483   0.1611

5.4

代码如下

```
clear,clc
p=[36.9,46.7,63.7,77.8,84.0,87.5]';
T=[181,197,235,270,283,292]';
pb=mean(p);
Tb=mean(T);
ah=sum((T-Tb).*(p-pb))/sum((p-pb).^2) %求 a 的估计值
bh=Tb-ah*pb %求 b 的估计值
```

结果如下

ah =

2.2337

bh =

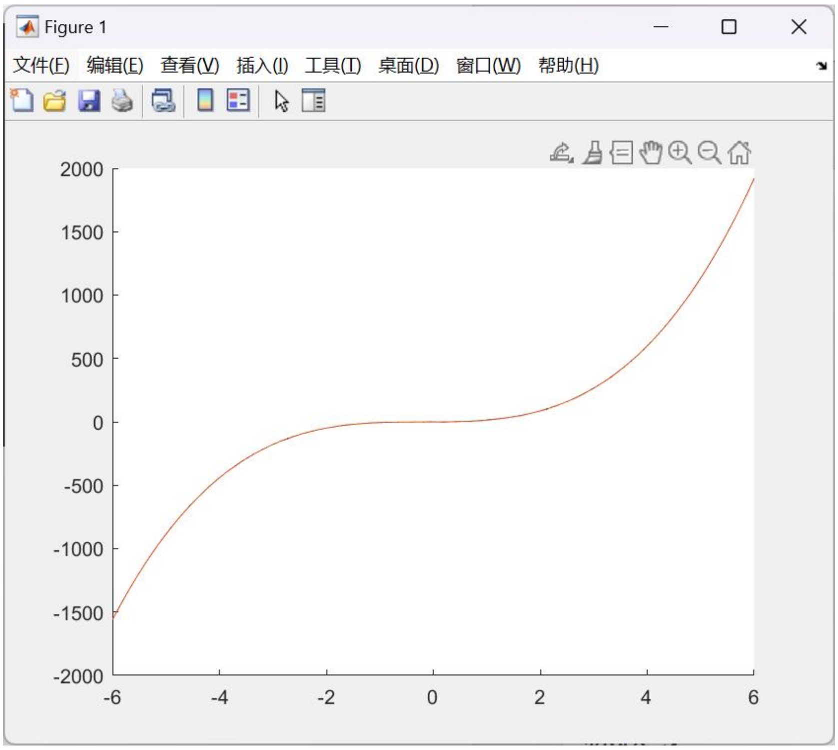
95.3524

5.5

代码如下

```
rng(2) %确保多次运行产生相同结果
x0=linspace(-6,6,100);
fx=@(x)8*x.^3+5*x.^2+2*x-1;
y0=fx(x0);
p=polyfit(x0,y0,3)
y1=polyval(p,x0) %生成多项式 p 在函数点 x0 的值
yh=y1+normrnd(0,1,size(y0)); %加入白噪声
p2=polyfit(x0,yh,3)
hold on
plot(x0,y1)
plot(x0,yh)
```

结果如下



微分方程补充题

问题分析

这是一个回归预测模型，需要我们采用经典的预测模型

符号说明

模型建立

$x(t)$：第t年的人口数量

x_m:环境人数最大容纳量

r：人口增长率

由题，我们可建立 logistic 人口模型

$$\begin{cases} \dfrac{dx}{dt} = r\left(1 - \dfrac{x}{x_m}\right)x \\ x(t_0) = x_0 \end{cases} \tag{1.1}$$

解方程，我们就可以得到

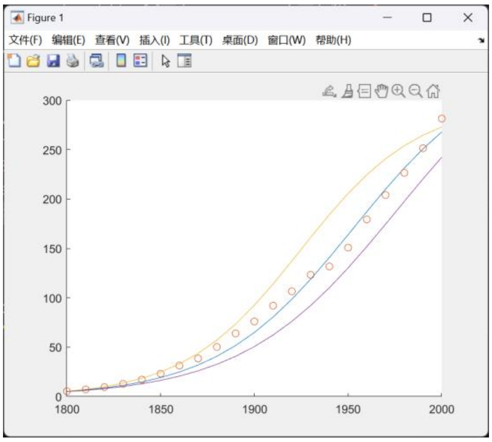
$$x(t) = \frac{x_m}{1 + \left(\frac{x_m}{x_0} - 1\right)e^{-r(t-t_0)}} \tag{1.2}$$

带入数据进行拟合即可得到系数

代码如下：

```
clear,clc
num=[5.3,7.2,9.6,12.9,17.1,23.2,31.4,38.6,50.2,63.9,76.0,92.0,106.5,123.2,131.7,150.7,179.3,204.0,226.5,251.4,281.4]';
num=num(~isnan(num));
t=[1800:10:2000]';
fn=@(r,xm,t)xm./(1+(xm/3.9-1)*exp(-r*(t-1790)));
ft=fittype(fn,'independent','t');
[f, st]=fit(t,num, ft, 'StartPoint',rand(1,2),...
    'Lower',[0,280],'Upper',[0.1,1000])  %由先验知识主观确定参数界限
xh=f(2010)  %求 2010 年的预测值
hold on
plot(t,f(t))
a=[ones(20,1), -num(1:end-1)];  %向前差分
b=diff(num)./num(1:end-1)/10;
cs=a\b; r1=cs(1), xm1=r1/cs(2)
xh2=fn(r1,xm1,2010)  %求 2010 年的预测值
plot(t,fn(r1,xm1,t))
a1=[ones(20,1), -num(2:end)];  %向后差分
b1=diff(num)./num(2:end)/10;
cs2=a1\b1; r2=cs2(1), xm2=r2/cs2(2)
xh3=fn(r2,xm2,2010)  %求 2010 年的预测值
plot(t,fn(r2,xm2,t))
```

结果如下：



xh =

282.5607

r1 =

0.0320

xm1 =

297.4815

xh2 =

279.2310

r2 =

0.0245

xm2 =

377.2795

xh3 =

  262.7204

6.1

这题我们可以直接解出方程的解，解如下

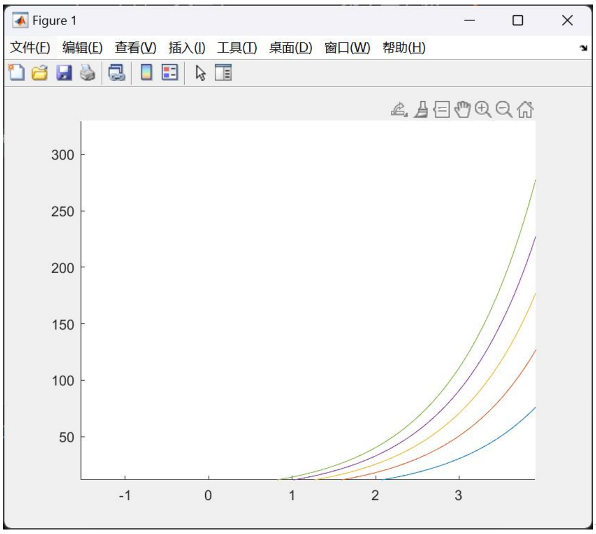
$$y = c_1 e^x - \frac{1}{2}(\sin x + \cos x) \tag{1.1}$$

带入方程即可

代码如下

```
clear,clc
x=-2:0.01:4;
hold on
for c=1.5:1:5.5
    plot(x,c.*exp(x)-0.5*(sin(x)+cos(x)))
end
```

结果如下



、

6.2

问题分析

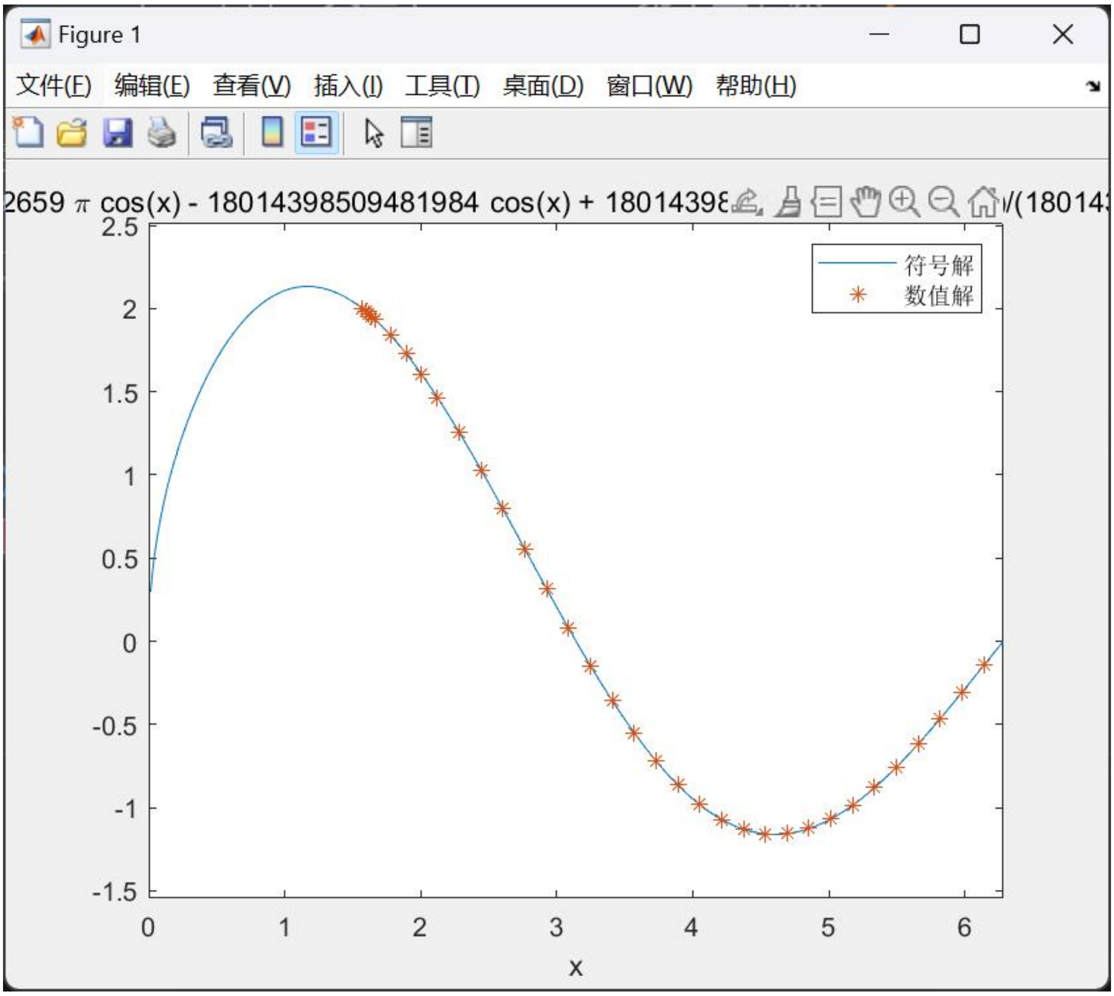
这题是贝塞尔方程，我们只需将其解出就可得到对应的解

代码如下

```
clc,clear
syms y(x)
Dy = diff(y);%按照差分的定义我们就可以得到导数
y=dsolve(x^2*diff(y,2)+x*diff(y)+(x^2-1/4)*y,y(pi/2)==2,Dy(pi/2)==-2/pi)
;
```

```
y=simplify(y)%化简所得到的符号解
pretty(y);
ezplot(y);
hold on
dy=@(x,y)[y(2);(1/(4*x^2)-1)*y(1)-y(2)/x];
[x,y]=ode45(dy,[pi/2,8],[2,-2/pi]);%用求解器求解微分方程
plot(x,y(:,1),'*')
legend('符号解','数值解')
```

结果如下：



6.3

问题分析

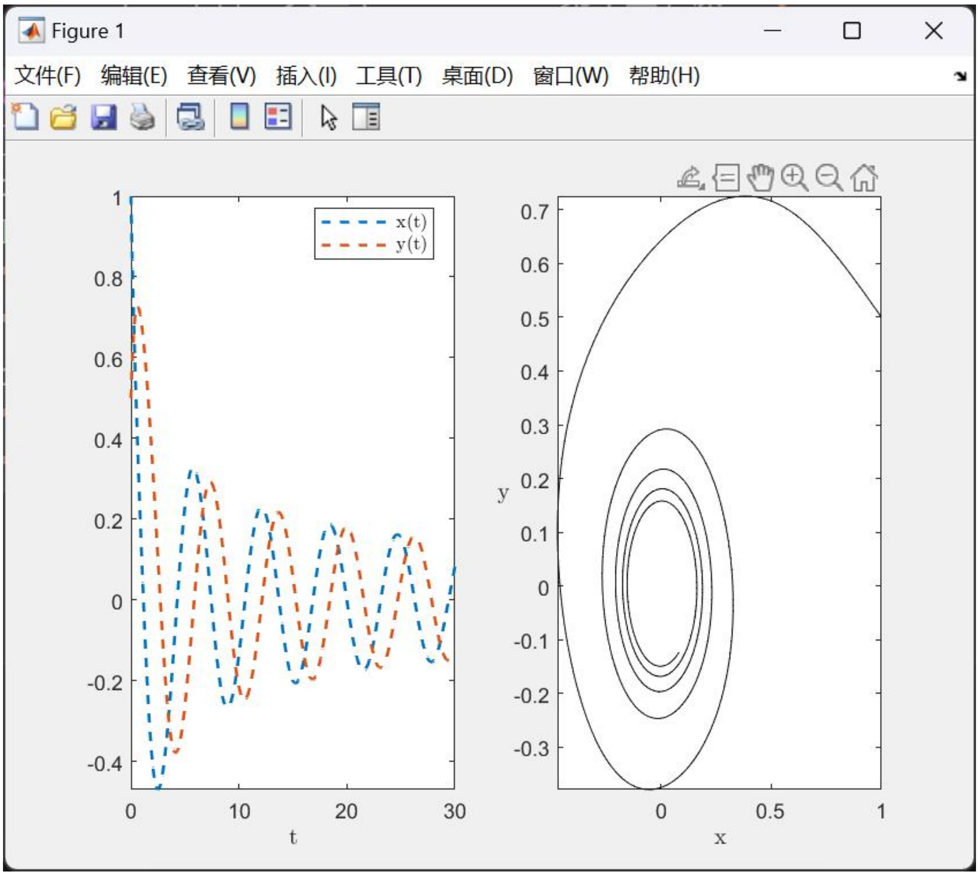
这题是解微分方程组，我们可以通过调用求解器来求解

代码如下

```
clc,clear
f=@(t,z)[-z(1)^3-z(2);z(1)-z(2)^3];
s=ode45(f,[0,30],[1;0.5])%使用求解器求解
subplot(121),fplot(@(t)deval(s,t,1),[0,30],'--','LineWidth',1.3)
hold on ,fplot(@(t)deval(s,t,2),[0,30],'--','LineWidth',1.3)
legend({'x(t)','y(t)'},'Location','best','Interpreter','latex')
xlabel('t','Interpreter','latex')
```

```
subplot(122),fplot(@(t)deval(s,t,1),@(t)deval(s,t,2),[0,30],'k')
xlabel('x','Interpreter','latex')
ylabel('y','Interpreter','latex','Rotation',0)
```

结果如下



6.8

问题分析

这是微分方程模型，需要我们建立微分方程对其求解

模型建立

由题，我们可以建立如下微分方程

$$\frac{ds}{dt}=pa(t)\left(1-\frac{s(t)}{M}\right)-\lambda s(t) \tag{1.2}$$

由题，对于广告选取如下

$$a(t)=\begin{cases}\frac{a}{\tau},0<t<\tau\\ 0,t>\tau\end{cases} \tag{1.3}$$

我们可以解得

$$s(t)=\begin{cases}\frac{\frac{pa}{r}}{\lambda+\frac{pa}{M\tau}}\left(1-e^{-\left(\lambda+\frac{pa}{M\tau}\right)t}\right)+s_0e^{-\left(\lambda+\frac{pa}{M\tau}\right)t},0\le t\le\tau\end{cases} \tag{1.4}$$

1. 贷款问题

问题重述

这是还款问题，需要我们从等额本金和等额本息的角度分别讨论所需还款的最优解

符号假设

s_n：第n个月所需还款的总金额

r:月利率

x:当月还款额度

Q:贷款总金额

模型建立

若采用等额本金，则·我们可建立需要还款的公式

$$s_n = (1+r)s_{n-1} - x \tag{1.1}$$

若采用等额本息，我们可建立下述的公式

$$Q = x * \frac{(1-(1+r)^n)}{-r} \tag{1.2}$$

结果如下

```
clear,clc
Q=300000;
r=0.051/12;
N=360;
x1=round((1+r)^N*Q*r/((1+r)^N-1),3)%等额本金
x2=round((-Q*r*(1+r)^N)/(1-(1+r)^N))%等额本息
S1=x1*N%等额本金还款总额
S2=x2*N%等额本息还款总额
```

x1 =

1.6288e+03

x2 =

1629

S1 =

5.8639e+05

S2 =

586440

就结果如上，建议别买房，还不起

2.植物基因分布

问题重述

这道题需要我们解决若干年后的分布问题

符号假设

a_n:第n代时AA的占比

b_n：第n代时Aa的占比

c_n:第n代时aa的占比

模型建立

对于任意一代基因，我们都有

$$a_n+b_n+c_n=1 \tag{1.3}$$

同时，我们有

$$\begin{cases} a_n=a_{n-1}+\dfrac{1}{2}b_{n-1} \\ b_n=\dfrac{1}{2}b_{n-1}+c_{n-1} \\ c_n=0 \end{cases} \tag{1.4}$$

可得到矩阵为

$$\begin{pmatrix} a_n \\ b_n \\ c_n \end{pmatrix}=\begin{pmatrix} 1 & \dfrac{1}{2} & 0 \\ 0 & \dfrac{1}{2} & 1 \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} a_{n-1} \\ b_{n-1} \\ c_{n-1} \end{pmatrix}=\begin{pmatrix} 1 & \dfrac{1}{2} & 0 \\ 0 & \dfrac{1}{2} & 1 \\ 0 & 0 & 0 \end{pmatrix}^n\begin{pmatrix} a_0 \\ b_0 \\ c_0 \end{pmatrix} \tag{1.5}$$

由特征值，该矩阵可相似为对角矩阵，我们可求得

$$\begin{pmatrix} a_n \\ b_n \\ c_n \end{pmatrix}=\begin{pmatrix} 1 & \dfrac{1}{2} & 0 \\ 0 & \dfrac{1}{2} & 1 \\ 0 & 0 & 0 \end{pmatrix}^n\begin{pmatrix} a_0 \\ b_0 \\ c_0 \end{pmatrix}=\begin{pmatrix} 1 & 1-\left(\dfrac{1}{2}\right)^n & 1-\left(\dfrac{1}{2}\right)^{n-1} \\ 0 & \left(\dfrac{1}{2}\right)^n & \left(\dfrac{1}{2}\right)^{n-1} \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} a_0 \\ b_0 \\ c_0 \end{pmatrix} \tag{1.6}$$

答案如下、

直接从结果看，结果分分布趋向于 AA

```
clear,clc
x0=[0.2,0.3,0.5]';
```

```
L=[1,1/2,0;0,1/2,1;0,0,0];
L=sym(L);%转变为符号矩阵
p=charpoly(L);%求特征多项式
r=roots(p);
[P,D]=eig(L);%求特征向量和特征值
xl=P*diag([0,0,1])*inv(P)*x0
```

xl =

1
0
0

3. 汽车租赁公司的运营

问题重述

这道题需要我们解决若干年后的分布问题

符号假设

a_n:第n代时A的数量

b_n：第n代时B的数量

c_n:第n代时C的数量

模型建立

由题，我们有

$$\begin{pmatrix} a_n \\ b_n \\ c_n \end{pmatrix} = \begin{pmatrix} 0.6 & 0.3 & 0.1 \\ 0.2 & 0.7 & 0.1 \\ 0.1 & 0.3 & 0.6 \end{pmatrix} \begin{pmatrix} a_{n-1} \\ b_{n-1} \\ c_{n-1} \end{pmatrix} \tag{1.7}$$

同时，我们有

$$\begin{pmatrix} a_n \\ b_n \\ c_n \end{pmatrix} = \begin{pmatrix} 0.6 & 0.3 & 0.1 \\ 0.2 & 0.7 & 0.1 \\ 0.1 & 0.3 & 0.6 \end{pmatrix}^n \begin{pmatrix} 200 \\ 200 \\ 200 \end{pmatrix} \tag{1.8}$$

代码如下

```
clear,clc
x0=[200,200,200]';
L=[0.6,0.3,0.1;0.2,0.7,0.1;0.1,0.3,0.6]
L=sym(L);%转变为符号矩阵
p=charpoly(L);%求特征多项式
r=roots(p);
[P,D]=eig(L)%求特征向量和特征值
xl=P*diag([1,0,0])*inv(P)*x0
```

结果如下：
L =

0.6000	0.3000	0.1000
0.2000	0.7000	0.1000
0.1000	0.3000	0.6000

P =

[1, -1/4,   1]
[1, -1/4, -1]
[1,   1,   1]

D =

[1,   0,   0]
[0, 1/2,   0]
[0,   0, 2/5]

xl =

200
200
200

7.1

一．问题分析：该题近似服从正态分布，置信水平为 0.9 时，显著性水平为 0.1，据此可求总体均值的置信水平为 0.9 的置信区间。

二．答案解释：利用 Matlab 软件求得，置信水平为 0.9 的置信区间是：[1065，1255]

三．代码与运行结果：

```
x0=[1050 1100 1120 1250 1280];
x0=x0(:);
pd=fitdist(x0,'Normal')%对数据进行正态拟合
ci=paramci(pd,'Alpha',0.1)%ci 的第一列是均值的置信区间
[mu,s,muci,sci]=normfit(x0,0.1)%另一种方法求置信区间
```

```
pd =

  NormalDistribution

  正态 分布
      mu =    1160   [1036.14, 1283.86]
   sigma = 99.7497   [59.7633, 286.636]


ci =

   1.0e+03 *

    1.0649    0.0648
    1.2551    0.2366


mu =

        1160


s =

   99.7497


muci =

   1.0e+03 *

    1.0649
    1.2551
```

```
sci =

   64.7680
  236.6417
```

7.2

一. 问题分析：本题要求在显著性水平α=0.05 下检验假设，将可能的区间分为五个两两不相交的小区间 $A_1, \ldots, A_5$，（分法见下代码），则有α=0.05，k=5，r=0，且计算得均值 x=15.078，样本标准差 s=0.4325，则可以得到结果。

二. 答案解释：利用 Matlab 软件求得：H=0，则接受原假设，故可以认为滚珠直径服从正态分布 N（15.078,0.4235^2）（α=0.05）

三. 代码与运行结果：

```
alpha=0.05
edges=[14:0.4:16];
x=[14.2:0.4:15.8]；%原始数据区间的边界和中心
mi=[3 8 19 12 8]%各组频数
pd=makedist('Normal','mu',15.078,'sigma',0.4325)%定义正态分布
[h,p,st]=chi2gof(x,'cdf',pd,'Edges',edges,'Frequency',mi)
k2=chi2inv(1-alpha,st.df)

alpha =

    0.0500


x =

   14.2000   14.6000   15.0000   15.4000   15.8000


mi =

     3     8    19    12     8


pd =

  NormalDistribution

  正态 分布
       mu = 15.078
```

```
    sigma = 0.4325


h =

     0


p =

    0.6599


st =

  包含以下字段的 struct:

    chi2stat: 1.5978
          df: 3
       edges: [14 14.8000 15.2000 15.6000 16]
           O: [11 19 12 8]
           E: [13.0093 17.5437 13.7606 5.6864]


k2 =

    7.8147
```

7.4

一．问题分析：本题同时考虑组内和组间的方差和，根据题目所给数据，可以计算得到 ANOVA 表（见下方代码与运行结果），并根据 p 值来判断这三组数据是否有明显差异。

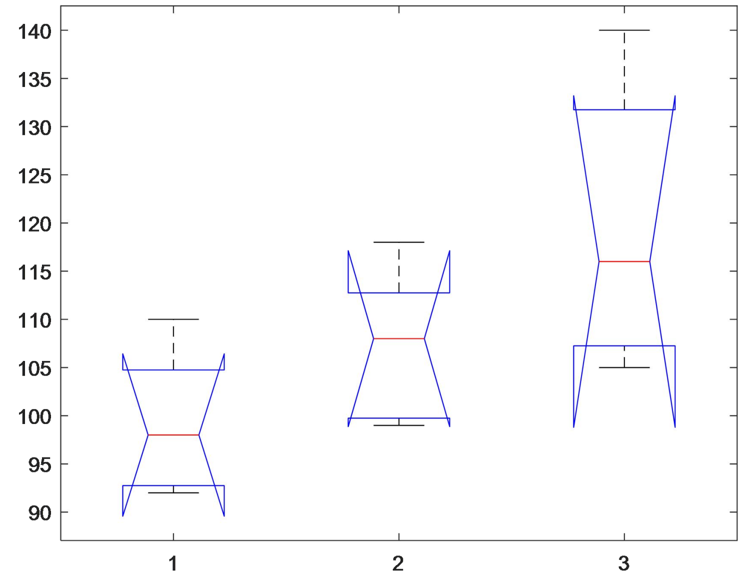
二．答案解释：利用 Matlab 软件求得：p 值为 0.0305＜0.05，因此可以得到结论：这三组五年保险理赔额有明显差异，画出的箱线图如下所示。

三．代码与运行结果：

```
a=[98,100,129;93,108,140;103,118,108;92,99,105;110,111,116]
[p,t,st]=anova1(a)
Fa=finv(0.95,t{2,3},t{3,3})
```

ANOVA 表

来源	SS	df	MS	F	p 值(F)
列	1056.53	2	528.267	4.73	0.0305
误差	1338.8	12	111.567		
合计	2395.33	14			



图：方差分析中的箱线图

a =

98	100	129
93	108	140
103	118	108
92	99	105
110	111	116

p =

0.0305

t =

4×6 cell 数组

{'来源'}	{'SS' }	{'df'}	{'MS' }	{'F' }	{'p 值(F)' }
{'列' }	{[1.0565e+03]}	{[2]}	{[528.2667]}	{[4.7350]}	{[0.0305]}
{'误差'}	{[1.3388e+03]}	{[12]}	{[111.5667]}	{0×0 double}	{0×0 double}
{'合计'}	{[2.3953e+03]}	{[14]}	{0×0 double}	{0×0 double}	{0×0 double}

st =

包含以下字段的 struct:

gnames: [3×1 char]
n: [5 5 5]
source: 'anova1'
means: [99.2000 107.2000 119.6000]
df: 12
s: 10.5625

Fa =

3.8853

>> [p,t,st]=anova(a)

7.5

一．问题分析：本题给出初始数据 x，y，并给出回归方程，则可利用最小二乘法求解回归方程系数。

二．答案解释：利用 Matlab 软件求得：a_1=0.6498，a_2=0.5901，a_3=0.0666，a_4=-0.0091，于是回归方程 $y=0.6498/x+0.5901+0.0666x-0.0091x^2$

三．代码与运行结果：

```
x=[1,2,4,5,7,8,9,10]';
y=[1.3,1,0.9,0.81,0.7,0.6,0.55,0.4]';%初始数据
A=[1,1,1,1,1,1,1,1]';
a=[1./x,A,x,x.^2];%构造系数矩阵
cs=a\y%算出系数
```

cs =

0.6498
0.5901
0.0666
-0.0091

7.6

一. 问题分析：题目已经给出 x，y 初始数据，且要求建立一元线性回归模型，可设 y=ax+b，通过最小二乘法即可求得系数。
二. 利用 Matlab 软件求得：a=0.9808，b=73.2410，则 y=73.2410+0.9808x。
三. 代码与运行结果：

```
x=[92,95,96,96.5,97,98,101,103.5,104,105,106,107,109]';
y=[163,165,167,168,171,170,172,174,176,176,177,177,181]';
A=[1,1,1,1,1,1,1,1,1,1,1,1,1]';
a=[x,A];
cs=a\y

cs =

    0.9808
   73.2410
```

7.8（1）
一. 问题分析：题目给出 y，x1，x2 的数据，并给出回归方程模型，通过最小二乘法可以求得系数。
二. 利用 Matlab 软件求得：b0=-67.3538,b1=106.9354，b2=0.0275，于是回归方程 y=-67.3538+106.9354x1+0.0275x2.（代码见底下附录）

7.8（2）
一. 问题分析：本题要求对 x1 和 x2 进行 T 检测，并求 R^2，则用 ttest 分别对 x1 和 x2 进行 T 检测，若 h=0 表示接受原假设，h=1 反之，另可通过计算分别求出残差平方和与总平方和，再利用公式 R^2=1-SSE/SST 可算出值。
二. 答案解释：利用 Matlab 软件求得：对 x1 和 x2 变量，在显著性水平α=0.05 的情况下，h 值都为 0，表示接受原假设，即 x1 和 x2 变量都是显著的，计算可得 SSE =1.0661e+04，SST = 6.4974e+05，R^2=1-SSE/SST=0.9836（代码见底下附录）

7.8（3）
一. 问题分析：已知 x1 与 x2，利用前面所得的回归函数，与 fitlm 函数可以得到预测值与预测置信区间。
二. 答案解释：当 x1=10,x2=9600 时，预测值为 1265.6154 元，显著性水平为 0.05 时，预测的置信区间是[1203.8319,1327.3988]

附录（代码与运行结果）：

```
x1=[4,5,4,7,5,10,7,6,9,8]';
x2=[3424,4086,4388,4808,5896,6604,6662,7018,8706,10478]';
y0=[450.5,613.9,501.5,781.5,611.1,1222.1,793.2,792.7,1121.0,1094.2]';
A=[1,1,1,1,1,1,1,1,1,1]';
a=[x1,x2,A];
```

```
cs=a\y0

[h1,p1,ci1,st1]=ttest(x1,mean(x1),'Alpha',0.05)
[h2,p2,ci2,st2]=ttest(x2,mean(x2),'Alpha',0.05)

n=length(y0)
y(:,1)=cs(1)*x1(:,1)+cs(2)*x2(:,1)+cs(3)*A
SSE=sum((y-y0).^2)
ay=sum(y0)/n
SST=sum((y0-ay).^2)
RR=1-SSE/SST

s1=10;s2=9600;
md=fitlm([x1,x2],y0)
[yh,yhint]=predict(md,[10,9600])


cs =

  106.9354
    0.0275
  -67.3538


h1 =

     0


p1 =

     1


ci1 =

    5.0204
    7.9796


st1 =

  包含以下字段的 struct:
```

```
    tstat: 0
       df: 9
       sd: 2.0683


h2 =

     0


p2 =

     1


ci2 =

   1.0e+03 *

    4.6413
    7.7727


st2 =

  包含以下字段的 struct:

    tstat: 0
       df: 9
       sd: 2.1887e+03


n =

    10


y =

   1.0e+03 *

    0.4544
    0.5795
    0.4809
```

```
    0.8132
    0.6292
    1.1833
    0.8641
    0.7670
    1.1341
    1.0759

SSE =

   1.0661e+04

ay =

  798.1700

SST =

   6.4974e+05

RR =

    0.9836

md =

线性回归模型:
    y ~ 1 + x1 + x2

估计系数:
                   Estimate       SE        tStat       pValue  
                   ________    ________    ________    ________

    (Intercept)     -67.354      44.301     -1.5204       0.17222
    x1               106.94      8.7484      12.223    5.6175e-06
    x2              0.02746    0.008267      3.3217      0.012736
```

```
观测值数目: 10，误差自由度: 7
均方根误差: 39
R 方: 0.984，调整 R 方 0.979
F 统计量(常量模型): 210，p 值 = 5.66e-07

yh =

   1.2656e+03


yhint =

   1.0e+03 *

    1.2038    1.3274
```