

## 2022 年中南大学数学建模竞赛

### 承 诺 书

我们完全清楚，在竞赛中必须合法合规地使用文献资料和软件工具，不能有任何侵犯知识产权的行为。否则我们将失去评奖资格，并可能受到严肃处理。

我们以中国大学生名誉和诚信郑重承诺，严格遵守竞赛章程和参赛规则，以保证竞赛的公正、公平性。如有违反竞赛章程和参赛规则的行为，我们将受到严肃处理。

我们授权中南大学数学建模竞赛指导教师团队，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

我们参赛选择的题号（从 A/B/C/D/E 中选择一项填写）：         A        

我们的报名参赛队号（7 位数字编号）：         2022261        

参赛队员（电子签名）： 1.         王岩琪        

2.         袁欣萍        

3.         郭禹含        

日期：   2022   年   7   月   3   日

（请勿改动此页内容和格式。以上内容请仔细核对，如填写错误，论文可能被取消评奖资格。）

# 满足一定力学性能的材料化学成分配方设计分析

## 摘 要

本文运用多元线性回归分析、最小二乘法、相关系数矩阵、SPSS 交互效应分析、二分决策树等模型进行了不同成分材料的力学性能差异的数学研究。

在第一问中，需要确定不同的化学成分对于各项力学性能产生的影响，因此针对该问题，我们首先使用 Python 中 Pandas 和 NumPy 创建相关系数矩阵，制作相关系数矩阵热图，对各成分与三项性能之间的相关程度进行可视化分析。随后利用 Spss 进行交互效应分析，并分析简单效应检测的 F 值与 sig 值，进一步确定元素成分与力学性能之间的相关关系。最终结合实验数据，制作三项力学性能的拟合曲线，从而直观得到不同特性随着不同化学成分占比的变化产生的变化。

在第二问中，需要在确定已知化学成分材料的条件下，求解未知的力学性能的均值与标准差，因此针对该问题我们利用 OLS (最小二乘法) 回归技术辅助多元线性回归 (Multiple Linear regression) 来解决。首先是利用 OLS 线性回归对各个熔炼号的材料进行屈服特性、抗拉特性、延伸率的均值和标准差的预测，确定各化学成分与力学性能的定量关系，其次是借助已经求解出来的定量关系，结合对应的材料与力学性能的多元线性回归方程的具体表达式，对指定熔炼号的化学成分进行分析计算。

在第三问中，在确定三种需求力学性质（屈服、抗拉、延伸率）后，首先根据“决策图”初步确定各元素三种力学性质的最优点成分范围，并通过“筛选图”检验实验数据合理性。在不发生冲突的情况下则可依据已选定的最优范围进行下一步实验；若发生冲突，则依据二分决策树进行决策，当所选取的基准在一定范围变化时，通过分析引起的相关力学因素变化的“权重图”，来确定二分迭代的方向，在减少试验次数即降低实验成本的同时最终不断缩小范围，从而确定符合性能要求的方案。

**关键字：** 相关系数矩阵 交互效应分析 线性回归模型 最小二乘法 二分决策

# 目录

满足一定力学性能的材料化学成分配方设计分析.....	I
摘 要.....	I
1 问题综述.....	1
1.1 问题背景.....	1
1.2 问题提出.....	1
1.3 资料条件.....	1
2 模型假设.....	1
2.1 模型基本假设.....	1
3 数据预处理.....	2
3.1 附件 1：化学成分及力学性能数据处理结果.....	2
3.1.1 指标选取.....	2
3.1.2 数据清洗.....	2
3.1.3 数据规约.....	2
3.1.4 分析截图.....	2
4 问题分析.....	2
4.1 问题一.....	2
4.2 问题二.....	3
4.3 问题三.....	3
5 模型建立及求解.....	3
5.1 问题一.....	3
5.1.1 模型分析及建立求解.....	3
5.2 问题二.....	6
5.2.1 模型分析及建立求解.....	6
5.2.2 模型原理解释.....	11
5.3 问题三.....	11
5.3.1 模型分析及建立求解.....	11
6 模型应用分析.....	15
6.1 效率分析.....	15
6.2 稳定性分析.....	15
7 模型评价与改进.....	17
7.1 模型评价.....	17
7.2 模型的改进.....	18

7.3 模型的推广 .....	18
参考文献 .....	18
附 录 .....	19
附录 A：化学成分及力学性能数据处理 .....	19
附录 B：代码 .....	19

# 1 问题综述

## 1.1 问题背景

信息、材料和能源在 20 世纪 70 年代被誉为当代文明的三大支柱，至 80 年代新材料、信息技术和生物技术又并列为新技术革命的三大重要标志。

材料的性能表征材料在给定的外界条件下所标示出来的行为，其使用性能包括力学性能、物理性能和化学性能。每一种金属材料能够得到最充分的利用、发挥出最大的价值，前提条件是金属材料的力学性能等各方面硬性条件和特点被使用者充分了解。材料的成分、组织及工艺密切影响着其性能，为使材料达成某一特定性能，需形成某种特定的组织，就需要确定加工处理工艺。即性能是目的，成分是基础，组织是形态，工艺是手段。力学性能指的是材料在外力（载荷）的作用下所显示出的特性，包括拉伸、压缩、弯曲等，针对某一特定材料生产过程中的相关成分数据及力学性能测试记录。

## 1.2 问题提出

材料的力学性能涉及多方面的问题，重点在于不同化学成分配方影响，往往主要考察屈服、抗拉、延伸率三个方面。需要从具体影响，定量关系，实验方案设计三个角度考虑，解决以下 3 个问题：

- (1) 问题 1: 某一特定材料，其不同化学成分如何对其各项力学性能产生影响，同时影响程度有多大；
- (2) 问题 2: 如何通过定量分析的相关模型建立，来确定材料中不同的化学成分与其各项力学性能的定量关系，并计算各熔炼号产出材料的力学性能均值和标准差；
- (3) 问题 3: 基于上面的各项模型建立与数据分析结果，为如何在保证最小成本的前提下，给出能尽快设计出符合性能要求材料的化学成分实验方案给出建议。

## 1.3 资料条件

附件提供了特定材料多次实验下相关数据记录和追求符合要求的最低成本的方案需求，各文件的详细说明如下：

- 化学成分及力学性能.xlsx

该文件提供了多次实验下记录的各熔炼号的不同化学成分比重及三项力学性能指标测度的数据，其中指标可能是模型的分析重点

# 2 模型假设

## 2.1 模型基本假设

- (1) 各化学成分对材料的力学性能有显著的影响，并呈密切的线性相关；

(2) 各化学成分之间应具有一定的互斥性，即不同化学成分之间的相关程度不高于化学成分与力学性能之间的相关程度；

(3) 化学成分具有完整的统计数据，其预测值容易确定。

### 3 数据预处理

#### 3.1 附件 1：化学成分及力学性能数据处理结果

##### 3.1.1 指标选取

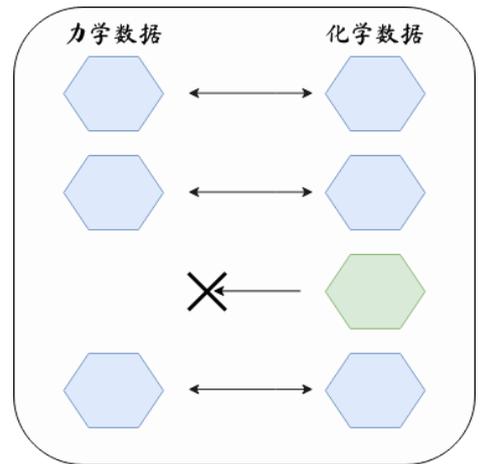
选取熔炼号所对应化学成分及力学性能均完备的数据

##### 3.1.2 数据清洗

通过移除合并等操作在解决不一致性的前提下来"清理"数据。使清洗后的数据能达到格式标准化，无异常数据、错误数据和重复数据。

##### 3.1.3 数据规约

通过算法提前计算好特定熔炼号所对应的化学成分以及力学性能的均值及标准差，既减小了数据的规模，同时基本保持了原数据的完整性，使结果与归约前结果相同或几乎相同。



##### 3.1.4 分析截图

```
data = pd.DataFrame({'熔炼号': ronglianhao, '屈服均值': phy_dict_qufu_mean_list, '抗拉均值': phy_dict_kangla_mean_list, '延伸率均值': phy_dict_yanshen_mean_list, '屈服标准差': phy_dict_qufu_std_list, '抗拉标准差': phy_dict_kangla_std_list, '延伸率标准差': phy_dict_yanshen_std_list, 'E1': np_E1, 'E2': np_E2, 'E3': np_E3, 'E4': np_E4, 'E5': np_E5, 'E6': np_E6})
```

熔炼号	屈服均值	抗拉均值	延伸率均值	屈服标准差	抗拉标准差	延伸率标准差	E1	E2	E3	E4	E5	E6	
0	90623	284.117647	304.411765	11.558824	4.042165	3.532352	0.565747	0.4965	1.0260	0.2155	0.1825	0.1070	0.0620
1	90622	280.833333	301.233333	11.733333	4.865411	4.192719	0.703957	0.4990	1.0325	0.2220	0.1840	0.1050	0.0625
2	90621	279.000000	300.812500	11.328125	4.183300	3.385978	0.786700	0.4935	1.0200	0.2290	0.1845	0.1020	0.0605
3	90620	283.088235	303.794118	11.558824	5.232071	4.350655	0.683468	0.4915	1.0165	0.2125	0.1800	0.1040	0.0620
4	90619	283.000000	304.400000	11.500000	3.346640	2.823709	0.645497	0.4945	1.0240	0.2175	0.1860	0.1155	0.0625
...	...	...	...	...	...	...	...	...	...	...	...	...	...
265	90020	277.718750	300.531250	11.203125	4.570519	3.699530	0.647825	0.5025	1.0255	0.2085	0.1905	0.1070	0.0630
266	90019	283.366667	305.166667	11.300000	4.408199	3.899430	0.556776	0.4920	1.0140	0.2085	0.1810	0.1090	0.0640
267	90018	276.605263	299.763158	11.710526	3.039559	2.905840	0.545696	0.4955	1.0115	0.2120	0.1770	0.1200	0.0645

## 4 问题分析

### 4.1 问题一

对于问题一，分析重点在于对已进行多次实验的某一特定材料，其在生产过程中的六种不同化学成分的数值及三项力学性能测试的记录。并依据数据制作拟合曲线图与相关系数矩阵热图，对已记录的数据进行直观表示，由此观察不同化学成分对各项力学性能的影响。

## 4.2 问题二

为确定已知化学成分材料的未知的力学性能均值与标准差，首先需要确定化学成分与力学性能的定量关系，其次借助已经确定的定量关系，计算得出指定熔炼号的材料的力学性能的数值，得出相关结果。

## 4.3 问题三

对于第三问，分析重点可分为两方面，一方面为符合性能要求材料的化学成分方案设计，需要充分考虑需求力学性质，并依据“决策图”确定不同成分的最优点范围；另一方面为最小成本确定，即需要尽可能的减少试验次数，一方面通过“筛选图”保证实验的成功率，一方面当数据发生冲突时，利用“决策二叉树”进行二分迭代实验，在保证精确度的前提下减次数、降成本。

# 5 模型建立及求解

## 5.1 问题一

### 5.1.1 模型分析及建立求解

材料力学性能是指材料在常温、静载作用下的宏观力学性能，是确定各种工程设计参数的主要依据，实验中针对化学性能的测度主要为屈服强度、抗拉强度与延伸率。材料中不同成分比重的变化将对不同的力学性能产生较大影响。

相关系数矩阵可以一次性同时显示多个变量之间的相关关系。

我们首先在 Python 中使用 Pandas 和 NumPy 创建相关系数矩阵，制作相关系数矩阵热图。由此可得到一个包含每个化学成分变量与其他力学性能变量之间的相关系数的图。图中的系数可以显示两两之间相关关系的强度及其方向。

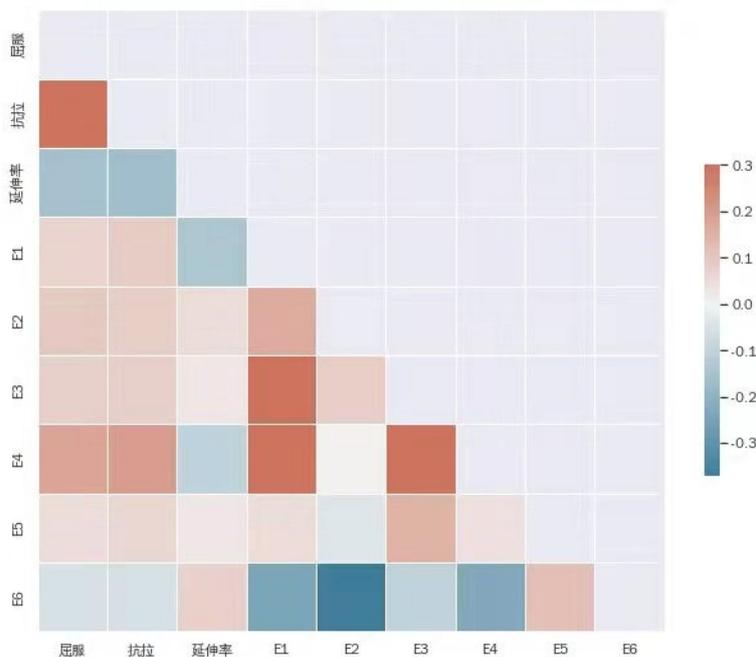
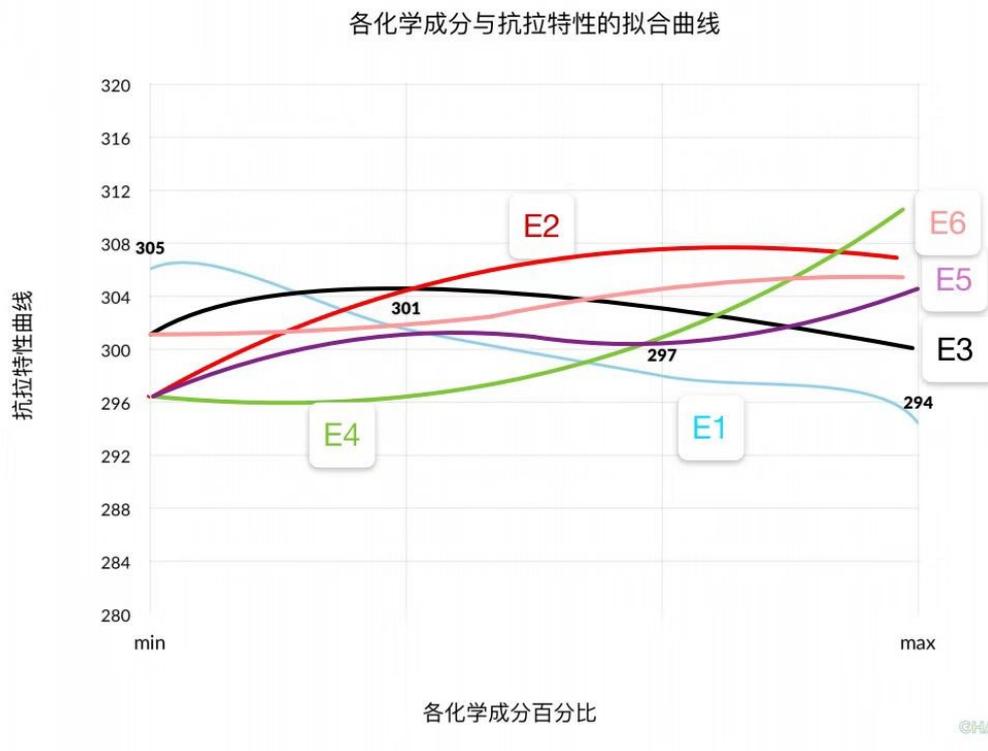


图 1 相关系数矩阵热图

根据热图中不同方块颜色对应的相关系数的大小，可以判断出变量之间相关性的强弱。例如 E4 与屈服特性、E1 与延伸率的相关关系较强，E3 与延伸率、E4 与延伸率的相关关系较弱。

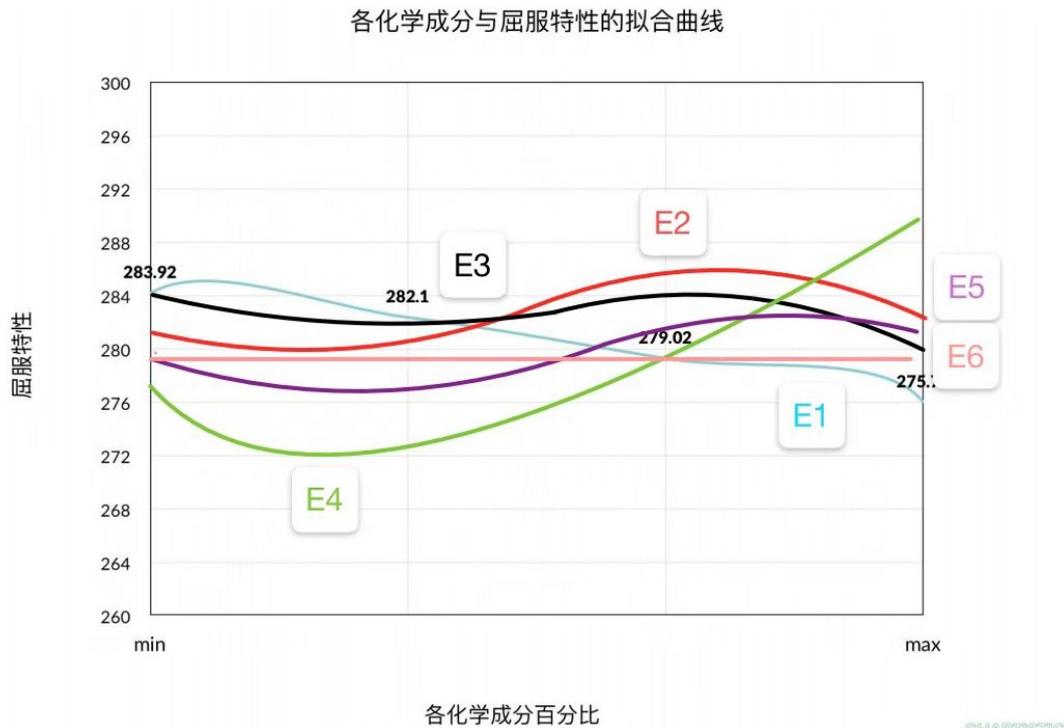
随后依据已知的表格中数据，利用 Spss 进行交互影响分析，利用“MANOVA”指令进行简单效应检验分析，得到各个元素与各力学特性的 F 值与 sig 值，进行比较分析，可得出与各力学性能紧密相关的要素。经验证，数据分析结构与相关系数矩阵热图基本吻合。进一步确定“元素是如何影响力学性能的”。

最后结合已知的数据进行回归方程拟合，并分析结果与上述分析基本契合。从而可分析六化学成分与三性能要素到底是如何相互作用的，最终绘制拟合曲线图并直观得出结果。



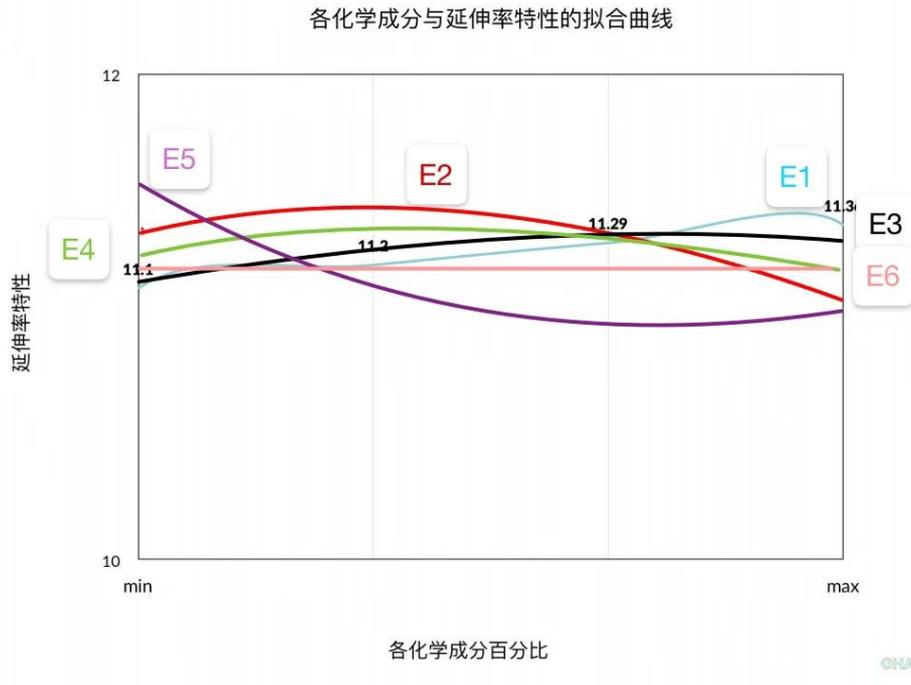
由图可知六化学成分占比变化时，对材料抗拉特性产生的影响。随着 E1 占比的提升，材料的抗拉特性在成分占比初始提升时（约为总量 10%）达到极致，但随后便持续降低；观察 E2 曲线可知，在其相对占比在 0-65%之间可知为，抗拉特性逐渐提升，并在约 65%处达到极致，随后略有降低；同时可知 E3 在占比为 0-30%之间逐渐攀升，并在约 30%处达到极致，随后抗拉特性便逐渐减弱；E4 整体曲线呈现上升趋势，在占比为 0-30%之间略有降低，随后逐渐增强，并在最终最大占比时达到最优；E5 整体略呈“S”形，变化较为明显，在 0-33%及 67%后缓慢提升，在中间区域缓慢下降；由图可知，E6 整体的影响及波动较小，抗拉特性变化较小，呈现缓慢上升趋势。

**//注 1：上述分析中所示百分比为相对占比 (min 对应所给数据中该元素最少含量, max 对应所给数据中该元素最多含量, 即对应元素数据中含量的最小值为 0%, 最大值为 100%)**



由图可知六化学成分占比变化时, 对材料屈服特性产生的影响。随着 E1 占比的提升, 材料的屈服特性在成分占比初始提升时 (约为总量 10%) 达到极致, 但随后便持续降低; 观察 E2 曲线可知, 在其总占比在 20%-75%之间可知为, 抗拉特性逐渐提升, 并在约 75%处达到极致, 而其他占比区间内均呈现降低态势; E3 整体曲线态势与 E2 重合, 不过相对波动较小、变化不明显; E4 整体曲线呈现先降后升趋势, 在占比为 0-25%之间降低, 随后逐渐增强, 并在最终最大占比时达到最优; E5 整体略呈“S”形, 变化较为明显, 在 0-33%及 67%后缓慢下降, 在中间区域缓慢上升; 由图可知, E6 整体的影响及波动小, 抗拉特性变化小, 呈现水平无变动趋势。

**//注 2：上述分析中所示百分比为相对占比 (min 对应所给数据中该元素最少含量, max 对应所给数据中该元素最多含量, 即对应元素数据中含量的最小值为 0%, 最大值为 100%)**



由图可知六化学成分占比变化时，对材料延伸率特性产生的影响。随着 E1 占比的提升，材料的屈服特性在成分占比提升时便逐渐攀升，但最后（90%处）便略有降低；观察 E2 曲线可知，以 40%为转折点，前期逐渐提升，后期持续降低；E3 整体曲线态势与 E1 重合，不过相对波动较小、变化不明显；E4 整体曲线态势与 E2 重合，不过相对波动较小、变化不明显，并在占比 35%处达到最优；E5 整体缓慢下降，在最后区域（80%后）缓慢上升；由图可知，E6 整体的影响及波动小，延伸率特性变化小，呈现水平无变动趋势。

**//注3：上述分析中所示百分比为相对占比（min 对应所给数据中该元素最少含量，max 对应所给数据中该元素最多含量，即对应元素数据中含量的最小值为 0%，最大值为 100%）**

## 5.2 问题二

### 5.2.1 模型分析及建立求解

为了量化地评估各化学成分对力学性能的影响程度，我们使用了多元线性回归的方法：将化学成分 E1,E2,E3,E4,E5,E6 分别记为 $x_1,x_2,x_3,x_4,x_5,x_6$ ,屈服特性的均值和标准差作为因变量分别记为 $y_1,y_2$ ,抗拉特性的均值和标准差记为 $y_3,y_4$ ,延伸率的均值和标准差记为 $y_5,y_6$ ,故可建立如下的数学模型：

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_6 x_6 + \xi \quad (i=1,2 \dots,6)$$

其中  $\beta_0, \dots, \beta_6$  是回归系数,  $\xi$  为随机误差项, 由于模型假设中认为各化学成分具有一定的互斥性, 故可认为  $\xi$  对实验中材料的力学性能无明显影响, 可在后续回归方程的求解中忽略。

在进行每一次多元线性回归之前, 先利用 Python(代码见附录)进行 OLS 回归对数据进行显著性检验。

显著性检验中利用 Durbin-Watson 检验, 因为自相关系数  $\rho$  的值介于 -1 和 1 之间, 所以  $0 \leq DW \leq 4$  并且  $DW=0 \Rightarrow \rho=1$ , 即存在正自相关性;  $DW=4 \Rightarrow \rho=-1$ , 即存在负自相关性,  $DW=2 \Rightarrow \rho=0$ , 即不存在(一阶)自相关性。因此, 当 DW 值显著的接近于 0 或 4 时, 则存在自相关性, 而接近于 2 时, 则不存在(一阶)自相关性。这样只要知道 DW 统计量的概率分布, 在给定的显著水平下, 根据临界值的位置就可以对原假设  $H_0$  进行检验。

同时利用到了 Jarque-Bera 检验, 其检验统计量定义为:

$$JB = \frac{n - k + 1}{6} * [S^2 + (0.25 * (C - 3)^2)]$$

JB 值总是正数, 如果统计量远大于零, 说明样本数据不具有正态分布特性, 可以使用多元回归方程估计。

在 Durbin-Watson 检验及 Jarque-Bera 检验均满足的条件下, 在 Python 中逐步建立上述多元回归的数学模型(代码见附录), 可以计算求得材料与力学性能均值与标准差的多元线性回归方程, 计算结果如下:

**(1)材料与屈服特性均值多元线性回归方程:**

	coef	std err	t	P> t	[0.025	0.975]
const	-0.7273	7.889	-0.092	0.927	-16.260	14.806
x1	-0.3171	8.382	-0.038	0.970	-16.822	16.188
x2	8.1385	5.774	1.410	0.160	-3.230	19.508
x3	-4.1692	8.264	-0.504	0.614	-20.442	12.104
x4	-18.5395	19.027	-0.974	0.331	-56.004	18.925
x5	-5.8194	17.024	-0.342	0.733	-39.340	27.701
x6	16.7100	30.795	0.543	0.588	-43.927	77.347
Omnibus:		201.569	Durbin-Watson:			1.642
Prob(Omnibus):		0.000	Jarque-Bera (JB):			3303.665
Skew:		2.815	Prob(JB):			0.00
Kurtosis:		19.185	Cond. No.			916.

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

$$y = -0.7273 - 0.3171x_1 + 8.1385x_2 - 4.1692x_3 - 18.5395x_4 - 5.8194x_5 + 16.7100x_6$$

验证 Durbin-Watson 检验及 Jarque-Bera 检验满足显著性条件, 可由 Python 得出 MLR 回归方程:

$$y_1 = 232.988 - 21.092x_1 + 24.716x_2 - 5.011x_3 + 226.012x_4 + 25.409x_5 - 139.418x_6$$

**(2)材料与屈服特性标准差多元线性回归方程：**

	coef	std err	t	P> t	[0.025	0.975]
const	234.8010	29.415	7.982	0.000	176.883	292.719
x1	-25.4511	31.256	-0.814	0.416	-86.996	36.093
x2	33.0086	21.530	1.533	0.126	-9.384	75.401
x3	-3.9719	30.816	-0.129	0.898	-64.650	56.706
x4	216.7327	70.946	3.055	0.002	77.038	356.428
x5	63.1518	63.478	0.995	0.321	-61.838	188.142
x6	23.3812	114.829	0.204	0.839	-202.720	249.482
Omnibus:		91.168	Durbin-Watson:			1.176
Prob(Omnibus):		0.000	Jarque-Bera (JB):			461.240
Skew:		-1.275	Prob(JB):			6.97e-101
Kurtosis:		8.873	Cond. No.			916.

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

$$y = 234.8010 - 25.4511x_1 + 33.0086x_2 - 3.9719x_3 + 216.7327x_4 + 63.1518x_5 + 23.3812x_6$$

验证 Durbin-Watson 检验及 Jarque-Bera 检验满足显著性条件，可由 Python 得出 MLR 回归方程：

$$y_2 = 5.063 + 4.611x_1 + 7.898x_2 - 15.912x_3 - 26.694x_4 - 5.158x_5 - 40.745x_6$$

**(3)材料与抗拉特性均值多元线性回归方程：**

	coef	std err	t	P> t	[0.025	0.975]
const	2.8216	9.671	0.292	0.771	-16.222	21.865
x1	-3.6748	10.277	-0.358	0.721	-23.910	16.561
x2	9.2533	7.079	1.307	0.192	-4.685	23.192
x3	-6.5848	10.132	-0.650	0.516	-26.535	13.366
x4	-21.5972	23.327	-0.926	0.355	-67.528	24.334
x5	-11.2795	20.871	-0.540	0.589	-52.375	29.816
x6	2.4244	37.755	0.064	0.949	-71.916	76.765
Omnibus:		238.636	Durbin-Watson:			1.608
Prob(Omnibus):		0.000	Jarque-Bera (JB):			6096.254
Skew:		3.424	Prob(JB):			0.00
Kurtosis:		25.249	Cond. No.			916.

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

$$y = 2.8216 - 3.6748x_1 + 9.2533x_2 - 6.5848x_3 - 21.5972x_4 - 11.2795x_5 + 2.4244x_6$$

验证 Durbin-Watson 检验及 Jarque-Bera 检验满足显著性条件，可由 Python 得出 MLR 回归方程：

$$y_3 = 265.837 - 8.333x_1 + 10.870x_2 - 4.294x_3 + 197.089x_4 + 30.815x_5 - 133.200x_6$$

**(4)材料与抗拉特性标准差多元线性回归方程：**

	coef	std err	t	P> t	[0.025	0.975]
const	203.3528	34.505	5.893	0.000	135.412	271.293
x1	-39.8312	36.665	-1.086	0.278	-112.025	32.363
x2	44.6137	25.255	1.767	0.078	-5.114	94.342
x3	1.2704	36.149	0.035	0.972	-69.907	72.448
x4	244.7439	83.223	2.941	0.004	80.876	408.612
x5	57.4122	74.462	0.771	0.441	-89.206	204.030
x6	27.2312	134.699	0.202	0.840	-237.995	292.457
Omnibus:		94.154	Durbin-Watson:			1.109
Prob(Omnibus):		0.000	Jarque-Bera (JB):			435.794
Skew:		-1.358	Prob(JB):			2.34e-95
Kurtosis:		8.600	Cond. No.			916.

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

$$y = 203.3528 - 39.8312x_1 + 44.6137x_2 + 1.2704x_3 + 244.7439x_4 + 57.4122x_5 + 27.2312x_6$$

验证 Durbin-Watson 检验及 Jarque-Bera 检验满足显著性条件，可由 Python 得出 MLR 回归方程：

$$y_4 = 1.530 + 5.522x_1 + 7.223x_2 - 13.924x_3 - 20.731x_4 - 0.081x_5 - 21.009x_6$$

(5)材料与延伸特性均值多元线性回归方程：

	coef	std err	t	P> t	[0.025	0.975]
const	9.9892	3.504	2.851	0.005	3.090	16.889
x1	-7.6164	3.724	-2.045	0.042	-14.948	-0.285
x2	3.8024	2.565	1.483	0.139	-1.248	8.853
x3	5.2722	3.671	1.436	0.152	-1.956	12.501
x4	-3.0590	8.452	-0.362	0.718	-19.701	13.583
x5	1.5053	7.562	0.199	0.842	-13.385	16.395
x6	15.4639	13.679	1.130	0.259	-11.471	42.399
Omnibus:		4.194	Durbin-Watson:			0.778
Prob(Omnibus):		0.123	Jarque-Bera (JB):			4.281
Skew:		-0.291	Prob(JB):			0.118
Kurtosis:		2.798	Cond. No.			916.

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

$$y = 9.9892 - 7.6164x_1 + 3.8024x_2 + 5.2722x_3 - 3.0590x_4 + 1.5053x_5 + 15.4639x_6$$

验证 Durbin-Watson 检验及 Jarque-Bera 检验满足显著性条件，可由 Python 得出 MLR 回归方程：

$$y_5 = 7.173 - 5.074x_1 + 5.762x_2 + 1.588x_3 - 8.096x_4 + 9.068x_5 + 22.527x_6$$

(6)材料与延伸特性标准差多元线性回归方程：

```

=====
              coef      std err          t      P>|t|      [0.025      0.975]
-----+-----
const          1.5023         1.257         1.195     0.233     -0.974         3.978
x1              0.1300         1.336         0.097     0.923     -2.501         2.761
x2             -0.3995         0.920        -0.434     0.665     -2.212         1.413
x3              0.3247         1.317         0.246     0.806     -2.269         2.919
x4             -0.3529         3.033        -0.116     0.907     -6.325         5.619
x5             -5.6869         2.714        -2.096     0.037    -11.030        -0.344
x6              1.8671         4.909         0.380     0.704     -7.798        11.532
=====
Omnibus:                223.455   Durbin-Watson:                1.792
Prob(Omnibus):           0.000   Jarque-Bera (JB):            6610.393
Skew:                    3.018   Prob(JB):                     0.00
Kurtosis:                26.477   Cond. No.                     916.
=====

```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

$$y = 1.5023 + 0.1300x_1 - 0.3995x_2 + 0.3247x_3 - 0.3529x_4 - 5.6869x_5 + 1.8671x_6$$

验证 Durbin-Watson 检验及 Jarque-Bera 检验满足显著性条件，可由 Python 得出 MLR 回归方程：

$$y_6 = 2.657 + 1.313x_1 - 1.717x_2 + 0.193x_3 - 0.867x_4 - 4.874x_5 - 4.013x_6$$

分别已知 OLS 回归方程和多元线性回归方程，首先只利用 OLS 线性回归对以下熔炼号的材料进行屈服特性，抗拉特性，延伸率的均值和标准差的预测，得出结果如下表：

熔炼号	均值			标准差		
	屈服	抗拉	延伸率	屈服	抗拉	延伸率
90624	281.0	302.1	11.7	4.04	3.92	0.68
90626	280.1	301.3	11.8	4.09	3.90	0.70
90627	281.1	302.1	11.8	4.18	4.00	0.67
90628	281.6	302.4	11.8	4.24	4.03	0.66
90629	280.6	301.7	11.8	4.13	3.95	0.68
90630	282.1	302.9	11.8	4.16	4.02	0.67
90631	281.2	302.1	11.8	4.13	3.90	0.67

表 1 指定熔炼号材料力学性能的均值和标准差(OLS)

基于上表 OLS 线性回归的结果，并结合对应的材料与力学性能的多元线性回归方程的具体表达式，对指定熔炼号的化学成分进行分析计算，可以对以下熔炼号的材料进行屈服特性，抗拉特性，延伸率的均值和标准差的预测，得出的结果如下表：

熔炼号	均值			标准差		
	屈服	抗拉	延伸率	屈服	抗拉	延伸率
90624	281.0	302.0	11.8	4.06	3.61	0.68
90626	280.5	301.6	11.7	4.08	3.56	0.71

90627	281.1	302.0	11.9	4.19	3.68	0.66
90628	281.5	302.4	11.9	4.28	3.75	0.65
90629	280.8	301.8	11.8	4.14	3.62	0.68
90630	282.1	302.9	11.8	4.23	3.74	0.66
90631	281.2	302.2	11.8	4.12	3.60	0.66

表 2 指定熔炼号材料力学性能的均值和标准差(MLR)

### 5.2.2 模型原理解释

针对该问题我们利用 OLS(最小二乘法)回归技术辅助多元线性回归(Multiple Linear regression)来解决。OLS(最小二乘法)作为一种回归手段和技术，主要用于线性回归的参数估计，它的思路很简单，就是求一些使得实际值和模型估值之差的平方和达到最小的值，将其作为参数估计值。就是说，通过最小化误差的平方和寻找数据的最佳函数匹配。利用最小二乘法可以简便地求得未知的数据，并使得这些求得的数据与实际数据之间误差的平方和为最小。

## 5.3 问题三

### 5.3.1 模型分析及建立求解

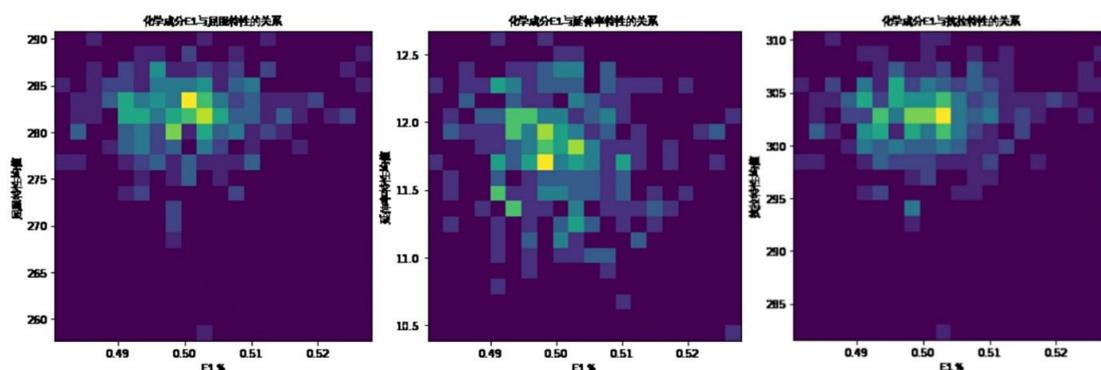
- 化学成分与力学特性的拟合关系——决策图

即通过题 1 中的关系拟合曲线初步确定元素范围。

- 实验数据拟合次数散点图——筛选图

利用散点图直观观测实验数据的分布情况，其反映了实验在各个成分比例区间的成功次数。（以黄色为中心的圆，越靠近圆心则越接近模拟真实实验，实验的成功率将大大提升）

制作二维直方图用以直观分析用所给实验数据分析出的拟合次数散点图，以 E1 为例可知分析如下：



针对 E1 成分与抗拉特性的实验数据统计图可知，E1 化学成分在 0.504% 左右时，实验数据拟合次数最多。因此可知在此区间范围内进行实验在满足性能要求的同时更加贴合实际，从而在此区间范围进行实验更加科学合理。

针对 E1 成分与屈服特性的二维直方图可知，在 E1 化学成分为 0.500% 时，实验数据拟合次数最多，即其在此区间范围内实验成功的可能性最大。

针对 E1 成分与延伸率的二维直方图可知，在 E1 化学成分为 0.497% 时，实验数据拟合次数最多，即其在此区间范围内实验成功的可能性最大。

#### ● 分析元素对相关力学性能的影响——权重图

以某一含量为基准，在此基准上元素含量进行上下 0.05% 的波动，观测对应力学性能的改变，绘制权重图来辅助二分迭代树确定二分进行的方向（权重图是通过分析在基准含量上下范围浮动 0.05% 的数据，并利用 Pandas 和 NumPy 库创建的相关系数矩阵，最终体现影响权重的图表）。

#### ● 发生冲突的决策方法——二分迭代树

首先利用决策图来分析满足相关性能要求下各元素成分比例，如果发生冲突，即两种特性无法兼得，则采用二分迭代树进行决策。首先选取冲突元素含量（较大与较小）的中间值作为基准，在此基准上绘制权重图，选取权重较大的一侧为二分方向（优先考虑影响较大的力学特性），之后不断重复迭代二分进行实验。在决策树已经初步缩小范围的基础上进一步缩小范围达到最优，且保证实验次数尽可能减少。当区间范围足够小或远超出筛选图的范围或临近两次实验效果接近时，则终止实验，此时为最终解。

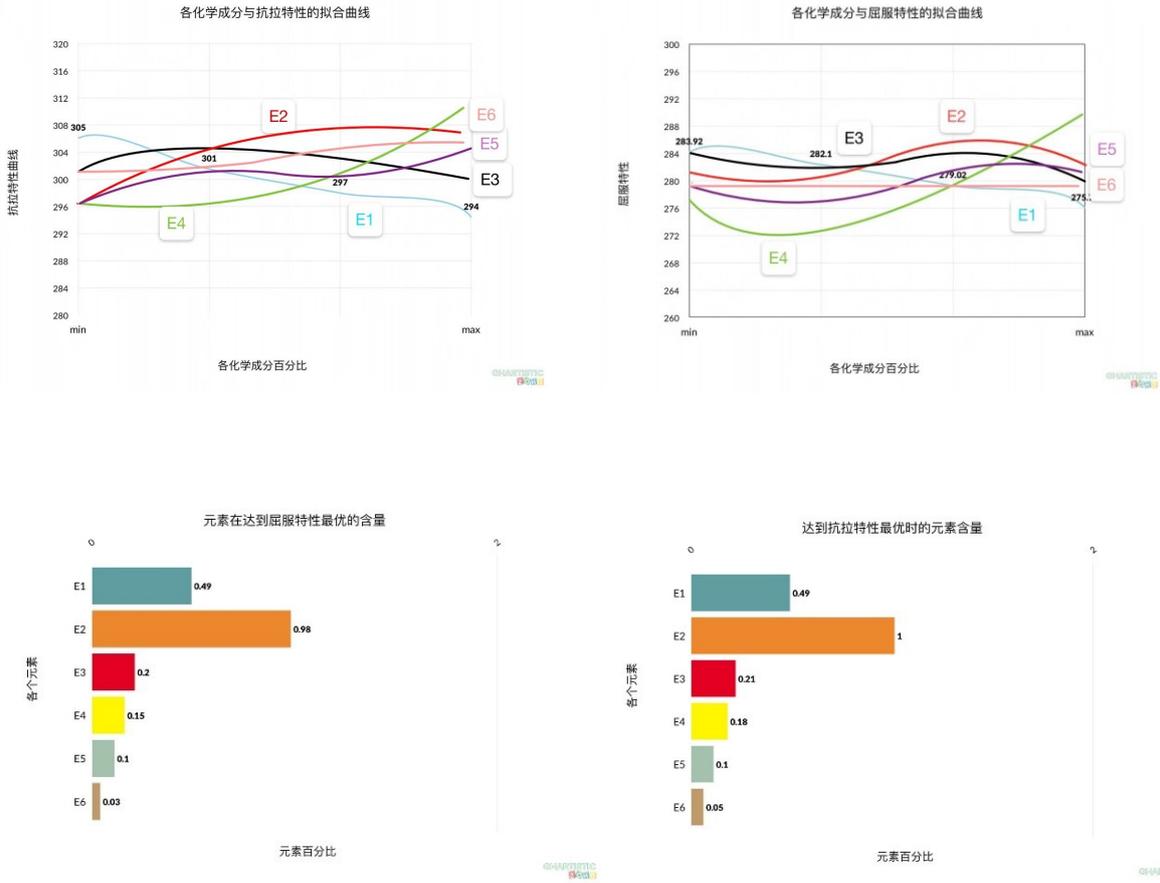
在确定三种需求力学性质（屈服、抗拉、延伸率）后，为设计符合性能要求的化学成分实验方案，第一步根据已知数据（化学成分与力学特性的拟合曲线）（下称“决策图”），初步确定各元素三种力学性质的最优点范围；第二步根据已知实验数据绘制的雷达散点图（下称“筛选图”），直观体现更加贴合实际实验的数据范围，在满足性能需求的基础上进一步保障实验的实际性与可操作性，从而进一步缩小实验范围。后续实验步骤根据具体情况具体分析，大体分为 2 种情况：冲突、不冲突。

随后参照同一化学成分在不同力学性能下的范围划定，如若需求的性能之间不发生冲突，则可直接依据决策图中的最优范围，结合筛选图确定实验区间范围，并进行下一步实验；如若利用决策图分析化学成分在不同性能的范围划定下发生冲突，则需要依据最小二分法，利用权重图分析不同性能影响的权重即观察实验结果受影响的相关浮动，不断向权重大的一侧迭代二分。二分的运用，目的是在保证结果精确度的同时，尽可能减少试验次数以利于降低实验成本，同时确定最终符合性能要求的方案。

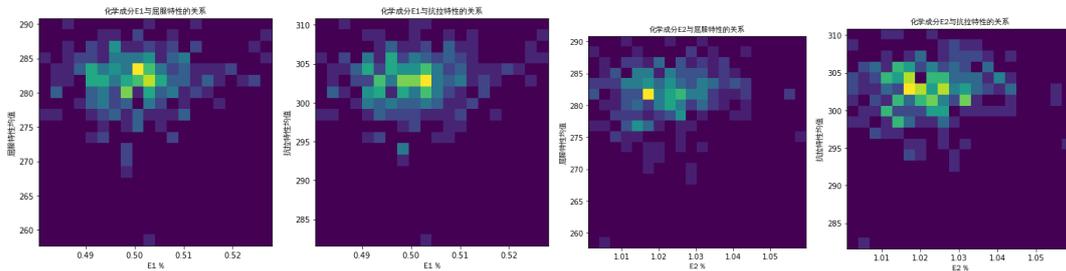
**以下举具体例子阐释决策应用：**

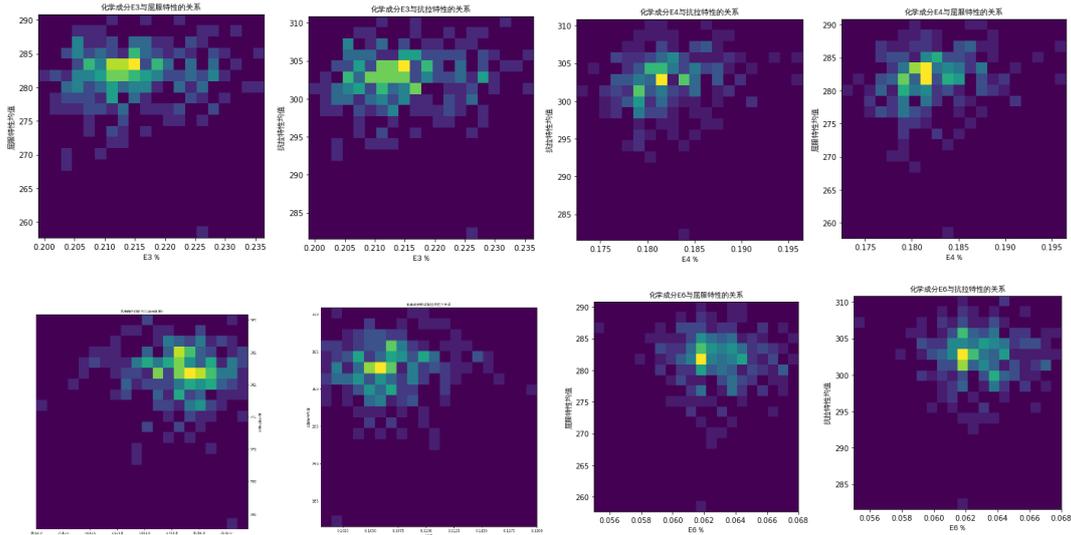
**例一：现需求一材料需同时满足抗拉与屈服较优（不冲突）**

首先根据决策图初步判断元素区间范围



此时发现各元素之间不发生冲突，则进行下一步的筛选：针对这一化学成分的不同特性需求进行范围筛定。

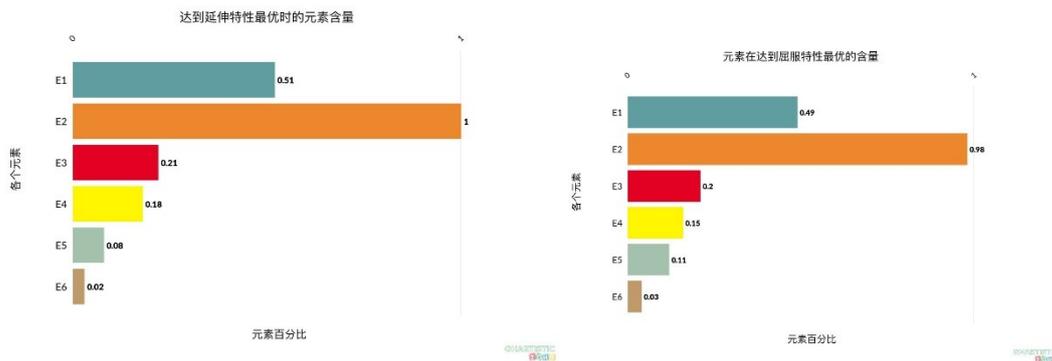




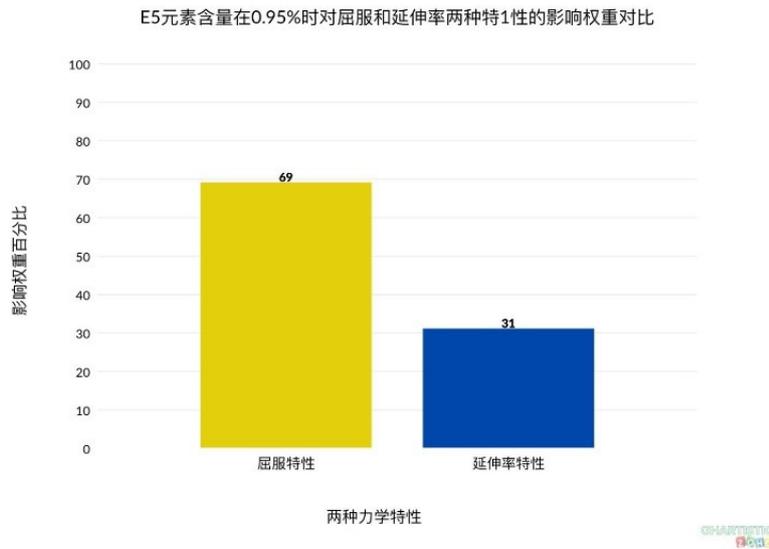
由筛选图可知，图中所选含量范围均在中心区域，则实验则围绕如决策图中的元素含量成分为中心开展实验。

**例二：现需求一材料需同时满足屈服与延展性较优（冲突）**

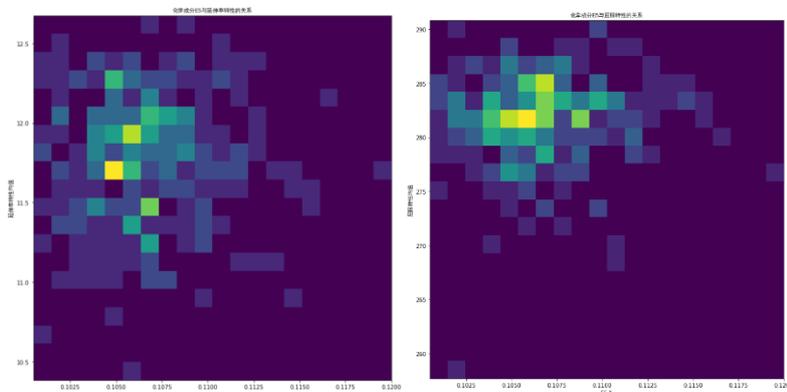
首先根据决策图初步判断元素区间范围



通过分析图标可知，E5 元素的含量发生了冲突（E5 含量与屈服特性成正比，与延展特性成反比）。此时需要运用二叉决策树进行进一步缩小实验范围，首先选取 0.08 与 1.10 的中间值 0.95%作为基准做权重图进行分析（权重图基于 0.95%左右的数据创建 Pandas 和 NumPy 相关系数矩阵构成），如下图



由权重图可知，在 0.95%的基准上，元素含量波动对屈服特性影响较大，所以确定了二分方向，即向 0.11%的方向继续迭代二分（优先考虑权重较大的力学性能），此时选取新的中间值 0.1005%进行实验，之后重复以上步骤做权重对比图。经测验，迭代 2 轮后，元素成分愈加趋于屈服特性筛选图黄色中心点(0.1053%左右)，同时数据不偏离延展性特性筛选图。则此时判定实验可以终止，继续进行性能提升不大。因此最终确定 E5 元素成分取第二轮实验结果 0.1053%为最终结果。



## 6 模型应用分析

### 6.1 效率分析

#### 效率指标

模型的效率考量应用的有效性程度及快速水平，好的模型必须经的起使用周期的检验。该模型采用二分法,实验效率为  $\log_2 n$ ，当实验基数越大时，效率提高优势越明显。

### 6.2 稳定性分析

#### 稳定性指标：误差分析

我们引入 2 种回归模型评价函数评估模型的预测准确性及泛化能力，为用平均绝对误差 (mean absolute error, 简称 MAE)、中值绝对误差 (median absolute error, 简称 MedAE) 评估模型的预测能力及稳定性。两种方法结合评估，评估实验值中存在误差异常值的影响。

运用 mae 进行计算分析可知在模型设置下的材料与三项力学性能的平均值，此数据说明拟合度较好

```

from sklearn.metrics import mean_absolute_error
# 材料与屈服特性均值 mae
y = np.array(dist2_qufu)
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=42)
model = LinearRegression().fit(X_train, y_train)
y_pred = model.predict(x)
mae = mean_absolute_error(y, y_pred)
print("材料与屈服特性均值 mae: " + str(mae))

材料与屈服特性均值 mae: 2.7263506275139773

# 材料与抗拉特性均值 mae
y = np.array(dist2_kangla)
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=42)
model = LinearRegression().fit(X_train, y_train)
y_pred = model.predict(x)
mae = mean_absolute_error(y, y_pred)
print("材料与抗拉特性均值 mae: " + str(mae))

材料与抗拉特性均值 mae: 2.322309498424725

# 材料与延伸率特性均值 mae
y = np.array(dist2_yanshen)
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=42)
model = LinearRegression().fit(X_train, y_train)
y_pred = model.predict(x)
mae = mean_absolute_error(y, y_pred)
print("材料与延伸率特性均值 mae: " + str(mae))

材料与延伸率特性均值 mae: 0.31346620270275244
    
```

运用 medae 进行计算分析可知在模型设置下的材料与三项力学性能的平均值，此数据说明拟合度较好

```

from sklearn.metrics import median_absolute_error
# 材料与屈服均值 MedAE
y = np.array(dist2_qufu)
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=42)
model = LinearRegression().fit(X_train, y_train)
y_pred = model.predict(x)
medae = median_absolute_error(y, y_pred)
print("材料与屈服特性均值 medae: " + str(medae))

材料与屈服特性均值 medae: 2.0298065954226274

# 材料与抗拉均值 MedAE
y = np.array(dist2_kangla)
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=42)
model = LinearRegression().fit(X_train, y_train)
y_pred = model.predict(x)
medae = median_absolute_error(y, y_pred)
print("材料与抗拉特性均值 medae: " + str(medae))

材料与抗拉特性均值 medae: 1.848467938080688

# 材料与延伸率均值 MedAE
y = np.array(dist2_yanshen)
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=42)
model = LinearRegression().fit(X_train, y_train)
y_pred = model.predict(x)
medae = median_absolute_error(y, y_pred)
print("材料与延伸率特性均值 medae: " + str(medae))

材料与延伸率特性均值 medae: 0.26360919435818626
    
```

运用 R 方计算分析，可知在模型设置下的材料与三项力学性能的平均值，此数据说明拟合度较好

```

import statsmodels.api as sm
x = sm.add_constant(np.array([dist1_E1, dist1_E2, dist1_E3, dist1_E4, dist1_E5, dist1_E6]).transpose())
# 材料与屈服特性均值回归方程
y = np.array(dist2_qufu)
model = sm.OLS(y, x)
results = model.fit()
print(results.summary())
    
```

OLS Regression Results			
Dep. Variable:	y	R-squared:	0.049
Model:	OLS	Adj. R-squared:	0.027
Method:	Least Squares	F-statistic:	2.247
Date:	Sun, 03 Jul 2022	Prob (F-statistic):	0.0393
Time:	14:20:56	Log-Likelihood:	-742.34
No. Observations:	270	AIC:	1499.
Df Residuals:	263	BIC:	1524.
Df Model:	6		
Covariance Type:	nonrobust		

```
# 材料与抗拉特性均值回归方程
y = np.array(dist2_kangla)
model = sm.OLS(y, x)
results = model.fit()
print(results.summary())
```

Dep. Variable:	y	R-squared:	0.052
Model:	OLS	Adj. R-squared:	0.031
Method:	Least Squares	F-statistic:	2.425
Date:	Sat, 02 Jul 2022	Prob (F-statistic):	0.0268
Time:	22:12:18	Log-Likelihood:	-699.25
No. Observations:	270	AIC:	1412.
Df Residuals:	263	BIC:	1438.
Df Model:	6		
Covariance Type:	nonrobust		

```
# 材料与延伸率特性均值回归方程
y = np.array(dist2_yanshen)
model = sm.OLS(y, x)
results = model.fit()
print(results.summary())
```

Dep. Variable:	y	R-squared:	0.039
Model:	OLS	Adj. R-squared:	0.017
Method:	Least Squares	F-statistic:	1.793
Date:	Sat, 02 Jul 2022	Prob (F-statistic):	0.101
Time:	22:12:18	Log-Likelihood:	-124.81
No. Observations:	270	AIC:	263.6
Df Residuals:	263	BIC:	288.8
Df Model:	6		
Covariance Type:	nonrobust		

## 7 模型评价与改进

### 7.1 模型评价

- (1) 问题一分别借助 Python 和 Spss 分别制作分析相关系数矩阵热图（描述各成分与三项性能之间的相关程度）及各种化学成分的交互影响关系，最终绘制出拟合曲线（描述影响变化），逻辑清晰、直观明了，但是在作图的过程中仍存在部分数据模糊不精确甚至缺失的情况。
- (2) 问题二利用 OLS(最小二乘法)回归技术辅助多元线性回归(Multiple Linear regression)方法，量化地评估各化学成分对力学性能的影响程度。在模型建立求解中获得的六个材料对应的不同的力学性能的回归方程中，对比同一变量的 OLS 回归方程和 MLR 回归方程，可以发现 MLR 的回归方程在 OLS 回归方程的基础上精度有所提高，与真实值误差更小，同时拟合度更高，故可以认为利用 OLS(最小二乘法)回归技术辅助多元线性回归这一数学模型以较高精度的方程解决了各化学成分与力学性能的定量关系。因此再利用由 OLS 回归的表格数据得出多元线性回归(MLR)对应的指定熔炼号材料力学性能的均值和标准差表格，其具有坚实的数学基础，模型有较高的精确度。但同时拟合的精度没有完全达到理想精度，仍需要控制更小的误差。
- (3) 问题三结合问题一和问题二，整体性强、架构清晰，且二分法计算简单、方法可靠、收敛性得到充分保证，模型整体构建较为良好，对工业生产改进、产品质量提升有

重要意义。但考虑到成本最小的要求，二分法仍存在速度慢、精确度相对较低的问题，此方面的探究仍有待深入。

## 7.2 模型的改进

- (1) 数据量过大、计算过于繁琐，需思考更为便捷的统计与计算方法。
- (2) 未能将所有的影响因素纳入计算，依然存在统计偏差，模型仍需改进，应尽可能提升精确度、减少偏差。
- (3) 分析第一问多因素影响权重时，可进一步学习  $R^2$  分解法进行更加全面的分析。

## 7.3 模型的推广

该模型可以推广到其他工艺生产方面，可以用  $x_i$  参数表示各种影响工艺品性能的各个互斥因素，从而解决有一定性能要求的工艺品成分问题。例如在解决混凝土配合比问题中，根据工程设计要求中同种混凝土不同强度等级在同一混凝土配合比设计中，选择不同的水灰比（或者说是灰水比）这就需要通过多元回归分析计算来验证确定所做的混凝土配合比设计。从而能够满足结构设计的强度抗渗、抗冻等等级要求，混凝土施工所要求的和易性，工程所处环境对混凝土耐久性的要求，同时又是最经济的配比。

## 参考文献

- [1] 《运筹学》教材编写组. 运筹学.第4版[M]. 清华大学出版社, 2012.
- [2] 俞树荣,张俊武,李建华.基于模糊聚类的材料力学性能确定方法[J].甘肃科学学报,2001(01):6-11.DOI:10.16468/j.cnki.issn1004-0366.2001.01.002.
- [3] 张良均. Python 数据分析与挖掘实战[M]. 机械工业出版社, 2016.
- [4] 冯国双. 白话统计[M]. 电子工业出版社, 2018.
- [5] 韩中庚. 数学建模方法及其应用-第2版[M]. 高等教育出版社, 2009.

## 附录

### 附录 A：化学成分及力学性能数据处理



附件1：化学成分  
及力学性能数据处

### 附录 B：代码

数学建模：满足一定力学性能的材料化学成分配方设计分析 CSU

Main.py

代码内容:

```
import pandas as pd
import numpy as np

def main():
    """
    数据清洗
    """

    with pd.ExcelFile("/home/bobmaster/Downloads/数学建模/附件1：化学成分及力学性能.xlsx") as origin_data:
        pd_chemicals_raw = pd.read_excel(origin_data, "化学成分", usecols=[0, 2, 3, 4, 5, 6, 7])
        pd_physics_raw = pd.read_excel(origin_data, "力学性能")

        pd_chemical = pd_chemicals_raw.iloc[1:, :]
        pd_physics = pd_physics_raw.dropna(how="any")
        # pd_chemical = pd_chemical.reindex(index = pd_chemical.index[::-1])
        pd_physics_ronglianhao = pd_physics.iloc[:, 0].astype("int64")
        pd_physics_qufu = pd_physics.iloc[:, 2]
        pd_physics_kangla = pd_physics.iloc[:, 3]
        pd_physics_yanshen = pd_physics.iloc[:, 4]

        # 提取相同熔炼号的力学性能数据
        comp_table = pd_physics.iloc[:, 0].duplicated(keep="last") # 比较表
        # phy_num = pd_physics.count() # 力学表数据量 11213
        phy_num = 11213
        # phy_ronglianhao = []
        phy_dict = {}
        phy_qufu = []
        phy_kangla = []
        phy_yanshen = []
        temp = 0
```

```
for i in range(phy_num):
    phy_qufu.append(pd_physics_qufu[i])
    phy_kangla.append(pd_physics_kangla[i])
    phy_yanshen.append(pd_physics_yanshen[i])
    if (comp_table[i] == False):
        # phy_ronglianhao[temp] = pd_physics_ronglianhao[i]
        phy_dict[pd_physics_ronglianhao[i]] = [phy_qufu, phy_kangla,
        phy_yanshen]
        temp += 1
        phy_qufu = []
        phy_kangla = []
        phy_yanshen = []

# 数据规约 - 力学性能数据均值和标准差
phy_dict_qufu_mean = {}
phy_dict_qufu_std = {}
phy_dict_kangla_mean = {}
phy_dict_kangla_std = {}
phy_dict_yanshen_mean = {}
phy_dict_yanshen_std = {}
phy_dict_qufu_mean_list = []
phy_dict_qufu_std_list = []
phy_dict_kangla_mean_list = []
phy_dict_kangla_std_list = []
phy_dict_yanshen_mean_list = []
phy_dict_yanshen_std_list = []

for key in phy_dict:
    np_physics_array_qufu = np.array(phy_dict[key][0])
    np_physics_array_kangla = np.array(phy_dict[key][1])
    np_physics_array_yanshen = np.array(phy_dict[key][2])
    phy_dict_qufu_mean[key] = np_physics_array_qufu.mean()
    phy_dict_qufu_std[key] = np_physics_array_qufu.std()
    phy_dict_kangla_mean[key] = np_physics_array_kangla.mean()
    phy_dict_kangla_std[key] = np_physics_array_kangla.std()
    phy_dict_yanshen_mean[key] = np_physics_array_yanshen.mean()
    phy_dict_yanshen_std[key] = np_physics_array_yanshen.std()

# 清洗化学成分
# 重建索引保证在同一熔炼号的情况下与力学指标数据匹配
pd_chem_ronglianhao = pd_chemical.iloc[:, 0].astype("int64")
pd_chem_ronglianhao = pd_chem_ronglianhao.drop_duplicates().reset_index().iloc[:, 1]
pd_chem_E1_data = pd_chemical.iloc[:, 1].reset_index().iloc[:, 1]
pd_chem_E2_data = pd_chemical.iloc[:, 2].reset_index().iloc[:, 1]
pd_chem_E3_data = pd_chemical.iloc[:, 3].reset_index().iloc[:, 1]
pd_chem_E4_data = pd_chemical.iloc[:, 4].reset_index().iloc[:, 1]
pd_chem_E5_data = pd_chemical.iloc[:, 5].reset_index().iloc[:, 1]
pd_chem_E6_data = pd_chemical.iloc[:, 6].reset_index().iloc[:, 1]
```

```
pd_chem_E1 = {}
pd_chem_E2 = {}
pd_chem_E3 = {}
pd_chem_E4 = {}
pd_chem_E5 = {}
pd_chem_E6 = {}

temp = 0

# 数据规约 - 化学成分
# 0-701 清洗后得到的范围
for i in range(702):
    if (i % 2 != 0 and temp != 321):
        pd_chem_E1[pd_chem_ronglianhao[temp]] = (pd_chem_E1_data[i - 1] +
pd_chem_E1_data[i]) / 2
        pd_chem_E2[pd_chem_ronglianhao[temp]] = (pd_chem_E2_data[i - 1] +
pd_chem_E2_data[i]) / 2
        pd_chem_E3[pd_chem_ronglianhao[temp]] = (pd_chem_E3_data[i - 1] +
pd_chem_E3_data[i]) / 2
        pd_chem_E4[pd_chem_ronglianhao[temp]] = (pd_chem_E4_data[i - 1] +
pd_chem_E4_data[i]) / 2
        pd_chem_E5[pd_chem_ronglianhao[temp]] = (pd_chem_E5_data[i - 1] +
pd_chem_E5_data[i]) / 2
        pd_chem_E6[pd_chem_ronglianhao[temp]] = (pd_chem_E6_data[i - 1] +
pd_chem_E6_data[i]) / 2
        temp += 1

# 整理出最终所需数据并保证化学成分与力学性能数据一致性
E1_list = []
E2_list = []
E3_list = []
E4_list = []
E5_list = []
E6_list = []
for key in pd_chem_E1:
    if key in phy_dict:
        E1_list.append(pd_chem_E1[key])
        E2_list.append(pd_chem_E2[key])
        E3_list.append(pd_chem_E3[key])
        E4_list.append(pd_chem_E4[key])
        E5_list.append(pd_chem_E5[key])
        E6_list.append(pd_chem_E6[key])

phy_dict_qufu_mean_list.append(phy_dict_qufu_mean[key])
phy_dict_qufu_std_list.append(phy_dict_qufu_std[key])
phy_dict_kangla_mean_list.append(phy_dict_kangla_mean[key])
phy_dict_kangla_std_list.append(phy_dict_kangla_std[key])
phy_dict_yanshen_mean_list.append(phy_dict_yanshen_mean[key])
phy_dict_yanshen_std_list.append(phy_dict_yanshen_std[key])
```

```
np_E1 = np.array(E1_list)
np_E2 = np.array(E2_list)
np_E3 = np.array(E3_list)
np_E4 = np.array(E4_list)
np_E5 = np.array(E5_list)
np_E6 = np.array(E6_list)

# 初始化二维直方图数据
# dist1 材料
dist1_E1 = np_E1
dist1_E2 = np_E2
dist1_E3 = np_E3
dist1_E4 = np_E4
dist1_E5 = np_E5
dist1_E6 = np_E6

# dist2 力学性能均值
dist2_qufu = np.array(phy_dict_qufu_mean_list)
dist2_kangla = np.array(phy_dict_kangla_mean_list)
dist2_yanshen = np.array(phy_dict_yanshen_mean_list)

# dist3 力学性能标准差
dist3_qufu = np.array(phy_dict_qufu_std_list)
dist3_kangla = np.array(phy_dict_kangla_std_list)
dist3_yanshen = np.array(phy_dict_yanshen_std_list)

# 绘制化学成分与力学特性关系的二维直方图

from hist2d import create_hist2d

create_hist2d(dist1_E1, dist2_qufu, title="化学成分 E1 与屈服特性的关系",
xlabel="E1 %", ylabel="屈服特性均值")
create_hist2d(dist1_E1, dist2_kangla, title="化学成分 E1 与抗拉特性的关系",
xlabel="E1 %", ylabel="抗拉特性均值")
create_hist2d(dist1_E1, dist2_yanshen, title="化学成分 E1 与延伸率特性的关系",
xlabel="E1 %", ylabel="延伸率特性均值")
create_hist2d(dist1_E2, dist2_qufu, title="化学成分 E2 与屈服特性的关系",
xlabel="E2 %", ylabel="屈服特性均值")
create_hist2d(dist1_E2, dist2_kangla, title="化学成分 E2 与抗拉特性的关系",
xlabel="E2 %", ylabel="抗拉特性均值")
create_hist2d(dist1_E2, dist2_yanshen, title="化学成分 E2 与延伸率特性的关系",
xlabel="E2 %", ylabel="延伸率特性均值")
create_hist2d(dist1_E3, dist2_qufu, title="化学成分 E3 与屈服特性的关系",
xlabel="E3 %", ylabel="屈服特性均值")
create_hist2d(dist1_E3, dist2_kangla, title="化学成分 E3 与抗拉特性的关系",
xlabel="E3 %", ylabel="抗拉特性均值")
```

```
    create_hist2d(dist1_E3, dist2_yanshen, title="化学成分 E3 与延伸率特性的关系",
", xlabel="E3 %", ylabel="延伸率特性均值")
    create_hist2d(dist1_E4, dist2_qufu, title="化学成分 E4 与屈服特性的关系",
xlabel="E4 %", ylabel="屈服特性均值")
    create_hist2d(dist1_E4, dist2_kangla, title="化学成分 E4 与抗拉特性的关系",
xlabel="E4 %", ylabel="抗拉特性均值")
    create_hist2d(dist1_E4, dist2_yanshen, title="化学成分 E4 与延伸率特性的关系",
", xlabel="E4 %", ylabel="延伸率特性均值")
    create_hist2d(dist1_E5, dist2_qufu, title="化学成分 E5 与屈服特性的关系",
xlabel="E5 %", ylabel="屈服特性均值", fig=(10, 10))
    create_hist2d(dist1_E5, dist2_kangla, title="化学成分 E5 与抗拉特性的关系",
xlabel="E5 %", ylabel="抗拉特性均值", fig=(10, 10))
    create_hist2d(dist1_E5, dist2_yanshen, title="化学成分 E5 与延伸率特性的关系",
", xlabel="E5 %", ylabel="延伸率特性均值", fig=(10, 10))
    create_hist2d(dist1_E6, dist2_qufu, title="化学成分 E6 与屈服特性的关系",
xlabel="E6 %", ylabel="屈服特性均值")
    create_hist2d(dist1_E6, dist2_kangla, title="化学成分 E6 与抗拉特性的关系",
xlabel="E6 %", ylabel="抗拉特性均值")
    create_hist2d(dist1_E6, dist2_yanshen, title="化学成分 E6 与延伸率特性的关系",
", xlabel="E6 %", ylabel="延伸率特性均值")

# 创建 OLS 回归模型
from linear_regression import OlsModel
from linear_regression import ols_calcutate_all

x = np.array([dist1_E1, dist1_E2, dist1_E3, dist1_E4, dist1_E5,
dist1_E6]).transpose()

# 材料与屈服特性均值回归模型
y = np.array(dist2_qufu)
qufu_mean_ols_model = OlsModel(x, y)

# 如需打印报告请删掉下一行的注释
# print(qufu_mean_ols_model.results.summary())

# 材料与抗拉特性均值回归模型
y = np.array(dist2_kangla)
kangla_mean_ols_model = OlsModel(x, y)

# 材料与延伸率特性均值回归模型
y = np.array(dist2_yanshen)
yanshen_mean_ols_model = OlsModel(x, y)

# 材料与屈服特性标准差回归模型
y = np.array(dist3_qufu)
qufu_std_ols_model = OlsModel(x, y)

# 材料与抗拉特性标准差回归模型
```

```

y = np.array(dist3_kangla)
kangla_std_ols_model = OlsModel(x, y)

# 材料与延伸率特性标准差回归模型
y = np.array(dist3_yanshen)
yanshen_std_ols_model = OlsModel(x, y)

# 给定熔炼号计算均值和标准差
# ronglianhao = 90624
# x1 = pd_chem_E1[ronglianhao]
# x2 = pd_chem_E2[ronglianhao]
# x3 = pd_chem_E3[ronglianhao]
# x4 = pd_chem_E4[ronglianhao]
# x5 = pd_chem_E5[ronglianhao]
# x6 = pd_chem_E6[ronglianhao]
# x = np.array([1, x1, x2, x3, x4, x5, x6])
# ols_calcutate_all(x, qufu_mean_ols_model, qufu_std_ols_model,
#                   kangla_mean_ols_model, kangla_std_ols_model,
#                   yanshen_mean_ols_model, yanshen_std_ols_model)

"""
屈服均值： [281.04367017]
抗拉均值： [302.12712467]
延伸率均值： [11.72968023]
屈服标准差： [4.04484533]
抗拉标准差： [3.60625011]
延伸率标准差： [0.68357895]
"""

from linear_regression import MlrModel
from linear_regression import mlr_calcutate_all
# 创建 MLR 多元线性回归模型
x = np.array([dist1_E1, dist1_E2, dist1_E3, dist1_E4, dist1_E5,
dist1_E6]).transpose()

# 材料与屈服特性均值回归模型
y = np.array(dist2_qufu)
qufu_mean_mlr_model = MlrModel(x, y)

# 回归系数
# qufu_mean_mlr_model.results.coef_
# 常数，回归方程截距
# qufu_mean_mlr_model.results.intercept_

# 材料与抗拉特性均值回归模型
y = np.array(dist2_kangla)
kangla_mean_mlr_model = MlrModel(x, y)

```

```
# 材料与延伸率特性均值回归模型
y = np.array(dist2_yanshen)
yanshen_mean_mlr_model = MlrModel(x, y)

# 材料与屈服特性标准差回归模型
y = np.array(dist3_qufu)
qufu_std_mlr_model = MlrModel(x, y)

# 材料与抗拉特性标准差回归模型
y = np.array(dist3_kangla)
kangla_std_mlr_model = MlrModel(x, y)

# 材料与延伸率特性标准差回归模型
y = np.array(dist3_yanshen)
yanshen_std_mlr_model = MlrModel(x, y)

# 给定熔炼号计算均值和标准差
# ronglianhao = 90624
# x1 = pd_chem_E1[ronglianhao]
# x2 = pd_chem_E2[ronglianhao]
# x3 = pd_chem_E3[ronglianhao]
# x4 = pd_chem_E4[ronglianhao]
# x5 = pd_chem_E5[ronglianhao]
# x6 = pd_chem_E6[ronglianhao]
# x = np.array([1, x1, x2, x3, x4, x5, x6]).reshape(-1,6)
# ols_calcutate_all(x, qufu_mean_ols_model, qufu_std_ols_model,
#                   kangla_mean_ols_model, kangla_std_ols_model,
#                   yanshen_mean_ols_model, yanshen_std_ols_model)

"""
屈服均值: [281.04919773]
抗拉均值: [302.13923671]
延伸率均值: [11.75333675]
屈服标准差: [4.06391763]
抗拉标准差: [3.6079243]
延伸率标准差: [0.68167218]
"""

from heatmap import create_heatmap
# 绘制热点图

dataset = pd.DataFrame(
    {'屈服': phy_dict_qufu_mean_list, '抗拉': phy_dict_kangla_mean_list, '
延伸率': phy_dict_yanshen_mean_list,
     'E1': np_E1, 'E2': np_E2, 'E3': np_E3, 'E4': np_E4, 'E5': np_E5,
     'E6': np_E6})
```

```
create_heatmap(dataset)

# from lineplot import create_lineplot
# 绘制折线图
# x = "E1" 横坐标, 从 E1-E6 选取
# y = "屈服" 纵坐标, 从屈服、抗拉、延伸率 选取
# create_lineplot(dataset, x, y)

if __name__ == '__main__':
    main()
```

## Lineplot.py

### 代码内容

```
import seaborn as sns

sns.set(font="WenQuanYi Zen Hei")

def create_lineplot(dataset, x, y):
    sns.lineplot(x="E1", y="屈服均值", data=dataset)
```

## linear\_regression

### 代码内容

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import statsmodels.api as sm

class OlsModel:
    def __init__(self, x, y):
        self.x = x
        self.y = y
        self.results = self.create_model()

    def create_model(self):
        x = sm.add_constant(self.x)
        model = sm.OLS(self.y, x)
        results = model.fit()
        return results

class MlrModel:
```

```
def __init__(self, x, y):
    self.x = x
    self.y = y
    self.results = self.create_model()

    def create_model(self):
        X_train, X_test, y_train, y_test = train_test_split(self.x, self.y,
test_size=0.25, random_state=42)
        model = LinearRegression().fit(X_train, y_train)
        return model

def ols_calcutate_all(x, qufu_mean_ols_model, qufu_std_ols_model,
                    kangla_mean_ols_model, kangla_std_ols_model,
                    yanshen_mean_ols_model, yanshen_std_ols_model):
    print("屈服均值: " + str(qufu_mean_ols_model.results.predict(x)) + "\n"
          "抗拉均值: " + str(kangla_mean_ols_model.results.predict(x)) + "\n"
          "延伸率均值: " + str(yanshen_mean_ols_model.results.predict(x)) +
"\n"
          "屈服标准差: " + str(qufu_std_ols_model.results.predict(x)) + "\n"
          "抗拉标准差: " + str(kangla_std_ols_model.results.predict(x)) + "\n"
          "延伸率标准差: " + str(yanshen_std_ols_model.results.predict(x)) +
"\n"
          )

def mlr_calcutate_all(x, qufu_mean_mlr_model, qufu_std_mlr_model,
                    kangla_mean_mlr_model, kangla_std_mlr_model,
                    yanshen_mean_mlr_model, yanshen_std_mlr_model):
    print("屈服均值: " + str(qufu_mean_mlr_model.results.predict(x)) + "\n"
          "抗拉均值: " + str(kangla_mean_mlr_model.results.predict(x)) + "\n"
          "延伸率均值: " + str(yanshen_mean_mlr_model.results.predict(x)) +
"\n"
          "屈服标准差: " + str(qufu_std_mlr_model.results.predict(x)) + "\n"
          "抗拉标准差: " + str(kangla_std_mlr_model.results.predict(x)) + "\n"
          "延伸率标准差: " + str(yanshen_std_mlr_model.results.predict(x)) +
"\n"
          )
```

## hist2d.py

### 代码内容:

```
import matplotlib.pyplot as plt
import matplotlib.font_manager as fm
```

```
myfont = fm.FontProperties(fname='/usr/lib/python3.10/site-packages/matplotlib/mpl-data/fonts/ttf/wqy-zenhei.ttc')

def create_hist2d(dist1, dist2, title, xlabel, ylabel, fig=(5, 5)):
    fig, ax = plt.subplots(figsize=fig, tight_layout=True)
    ax.set_title(title, fontproperties=myfont)
    ax.set_xlabel(xlabel, fontproperties=myfont)
    ax.set_ylabel(ylabel, fontproperties=myfont)
    hist2d = ax.hist2d(dist1, dist2, bins=20)
    plt.show()
```

## heatmap.py

### 代码内容

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

plt.rc('axes', unicode_minus=False)
sns.set_theme(style="white")
sns.set(font="WenQuanYi Zen Hei")

def create_heatmap(dataset):
    # 计算相关矩阵
    corr = dataset.corr()

    # 为上三角矩阵生成蒙版
    mask = np.triu(np.ones_like(corr, dtype=bool))

    # 设置图大小
    f, ax = plt.subplots(figsize=(11, 9))

    # 生成颜色图表
    cmap = sns.diverging_palette(230, 20, as_cmap=True)

    # 使用蒙版和正确的纵横比绘图
    sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.3, center=0,
                square=True, linewidths=.5, cbar_kws={"shrink": .5})

    plt.show()
```